



패턴

AWS 권장 가이드



AWS 권장 가이드: 패턴

Copyright © 2024 Amazon Web Services, Inc. and/or its affiliates. All rights reserved.

의 상표 및 브랜드 디자인은 외 제품 또는 서비스와 함께, 브랜드 이미지를 떨어뜨리거나 고객에게 혼동을 일으킬 수 있는 방식으로 사용할 수 없습니다. Amazon이 소유하지 않은 기타 모든 상표는 Amazon 계열사, 관련 업체 또는 Amazon의 지원 업체 여부에 상관없이 해당 소유자의 자산입니다.

Table of Contents

AWS 규범적 지침 패턴	1
분석	3
Microsoft SQL Server Analysis Services에서 Amazon Redshift 데이터를 분석	5
요약	5
사전 조건 및 제한 사항	5
아키텍처	6
도구	6
에픽	6
관련 리소스	8
.....	9
요약	9
사전 조건 및 제한 사항	9
아키텍처	10
도구	10
에픽	11
관련 리소스	16
AWS Glue에서 암호화 적용 자동화	17
요약	17
사전 조건 및 제한 사항	17
아키텍처	17
도구	18
모범 사례	19
에픽	20
관련 리소스	22
AWS Glue를 사용하여 Amazon S3에서 Amazon Redshift로의 ETL 파이프라인 빌드	23
요약	23
사전 조건 및 제한 사항	23
아키텍처	24
도구	24
에픽	25
관련 리소스	31
추가 정보	31
AWS 서비스를 사용하여 위험 가치(VaR) 계산	33
요약	33

사전 조건 및 제한 사항	34
아키텍처	34
도구	35
모범 사례	36
에픽	36
관련 리소스	39
정규화를 Amazon Redshift SQL로 변환	40
요약	40
사전 조건 및 제한 사항	40
아키텍처	40
도구	41
에픽	46
관련 리소스	46
RESET WHEN을 Amazon Redshift SQL로 변환	48
요약	48
사전 조건 및 제한 사항	48
아키텍처	48
도구	49
에픽	53
관련 리소스	53
.....	55
요약	55
사전 조건 및 제한 사항	55
아키텍처	56
도구	56
에픽	57
관련 리소스	60
첨부	60
Amazon S3에 Amazon EMR 로깅이 활성화되었는지 확인	61
요약	61
사전 조건 및 제한 사항	62
아키텍처	62
도구	63
에픽	63
관련 리소스	66
첨부	66

AWS Glue를 사용하여 테스트 데이터 생성	67
요약	67
사전 조건 및 제한 사항	67
아키텍처	68
도구	68
모범 사례	68
에픽	69
관련 리소스	78
추가 정보	78
Lambda 함수를 사용하여 Amazon EMR에서 Spark 작업 시작	83
요약	83
사전 조건 및 제한 사항	83
아키텍처	84
도구	84
에픽	85
관련 리소스	88
추가 정보	88
첨부	90
Apache Cassandra 워크로드를 Amazon Keyspaces로 마이그레이션하기	91
요약	91
사전 조건 및 제한 사항	91
아키텍처	92
도구	92
모범 사례	93
에픽	93
문제 해결	104
관련 리소스	105
추가 정보	105
Oracle Business Intelligence 12C를 AWS 클라우드로 마이그레이션	107
요약	107
사전 조건 및 제한 사항	107
아키텍처	108
도구	109
에픽	110
관련 리소스	120
추가 정보	120

를 사용하여 Kafka 클러스터를 Amazon MSK로 마이그레이션합니다. MirrorMaker	125
요약	125
사전 조건 및 제한 사항	125
아키텍처	126
도구	126
모범 사례	127
에픽	127
관련 리소스	130
추가 정보	131
ELK 스택을 AWS 클라우드로 마이그레이션	132
요약	132
사전 조건 및 제한 사항	133
아키텍처	134
도구	136
에픽	137
관련 리소스	143
추가 정보	145
Starburst를 사용하여 데이터를 AWS로 마이그레이션	146
요약	146
사전 조건 및 제한 사항	146
아키텍처	146
도구	148
에픽	149
관련 리소스	151
입력 파일 크기의 ETL 수집 최적화	153
요약	153
사전 조건 및 제한 사항	153
아키텍처	153
도구	154
에픽	154
관련 리소스	157
추가 정보	157
AWS Step Functions를 사용하여 ETL 파이프라인 오케스트레이션	158
요약	158
사전 조건 및 제한 사항	158
아키텍처	159

도구	160
에픽	161
문제 해결	168
관련 리소스	168
추가 정보	169
Amazon Redshift 기계 학습을 이용하여 기계 학습 분석 수행	170
요약	170
사전 조건 및 제한 사항	170
아키텍처	171
도구	172
에픽	173
관련 리소스	175
Athena를 사용하여 DynamoDB 테이블을 쿼리	177
요약	177
사전 조건 및 제한 사항	177
아키텍처	178
도구	178
에픽	179
관련 리소스	186
추가 정보	186
Amazon Redshift 쿼리 결과에 대한 언어별 정렬 설정	188
요약	188
사전 조건 및 제한 사항	188
아키텍처	189
도구	189
에픽	189
관련 리소스	194
추가 정보	194
크로스 리전 S3 버킷에서 이벤트 알림을 수신하도록 Lambda 함수 구독	198
요약	198
사전 조건 및 제한 사항	198
아키텍처	199
도구	199
에픽	200
관련 리소스	203
데이터 변환을 위한 세 가지 AWS Glue 작업 유형	204

요약	204
사전 조건 및 제한 사항	204
아키텍처	205
도구	205
에픽	206
관련 리소스	209
추가 정보	209
첨부	215
Athena를 사용하여 Amazon Redshift 감사 로그를 시각화하고 QuickSight	216
요약	216
사전 조건 및 제한 사항	216
아키텍처	217
도구	217
에픽	217
관련 리소스	221
첨부	221
Amazon을 사용하여 IAM 자격 증명 보고서를 시각화하십시오. QuickSight	222
요약	222
사전 조건 및 제한 사항	223
아키텍처	223
도구	224
에픽	225
추가 정보	230
패턴 더 보기	232
비즈니스 생산성	234
AWS에서 고가용성 PeopleSoft 아키텍처 설정	235
요약	235
사전 조건 및 제한 사항	235
아키텍처	236
도구	239
모범 사례	240
에픽	243
관련 리소스	260
더 많은 패턴	262
클라우드 네이티브	263
비디오 처리 파이프라인 구축	264

요약	264
사전 조건 및 제한 사항	264
아키텍처	265
도구	265
에픽	266
관련 리소스	272
추가 정보	272
첨부	273
AWS CLI를 사용하여 S3 버킷의 데이터를 다른 계정 및 리전으로 복사	274
요약	274
사전 조건 및 제한 사항	275
아키텍처	275
도구	275
모범 사례	275
에픽	275
문제 해결	287
관련 리소스	287
SAP RHEL 페이스메이커 클러스터를 모니터링하세요.	288
요약	288
사전 조건 및 제한 사항	288
아키텍처	289
도구	289
모범 사례	290
에픽	290
관련 리소스	304
첨부	304
S3 버킷을 CloudFormation 스택으로 성공적으로 가져오기	305
요약	305
사전 조건 및 제한 사항	305
아키텍처	305
에픽	306
관련 리소스	314
첨부	314
패턴 더 보기	315
컨테이너 및 마이크로서비스	317
Amazon ECS에서 컨테이너 애플리케이션에 액세스	319

요약	319
사전 조건 및 제한 사항	320
아키텍처	320
도구	321
에픽	322
관련 리소스	331
AWS Fargate 시작 유형을 사용하여 Amazon ECS의 컨테이너 애플리케이션에 액세스	334
요약	334
사전 조건 및 제한 사항	335
아키텍처	335
도구	336
에픽	337
관련 리소스	346
Amazon EKS에서 비공개로 컨테이너 애플리케이션에 액세스	348
요약	348
사전 조건 및 제한 사항	348
아키텍처	349
도구	349
에픽	350
관련 리소스	354
Amazon EKS의 App Mesh에서 mTLS 활성화	356
요약	356
사전 조건 및 제한 사항	356
아키텍처	357
도구	357
에픽	358
관련 리소스	361
추가 정보	361
Amazon RDS for PostgreSQL DB 인스턴스 백업 자동화	363
요약	363
사전 조건 및 제한 사항	364
아키텍처	364
도구	365
에픽	366
관련 리소스	371
추가 정보	372

노드 종료 핸들러 배포 자동화	375
요약	375
사전 조건 및 제한 사항	376
아키텍처	377
도구	377
모범 사례	379
에픽	379
문제 해결	386
관련 리소스	386
추가 정보	387
Amazon EKS에 Java 애플리케이션 자동 구축 및 배포	389
요약	389
사전 조건 및 제한 사항	389
아키텍처	390
도구	392
모범 사례	393
에픽	393
관련 리소스	409
추가 정보	409
Amazon EFS를 사용하여 EC2 인스턴스에 Amazon ECS 작업 정의 생성	410
요약	410
사전 조건 및 제한 사항	410
아키텍처	411
도구	412
에픽	412
관련 리소스	414
첨부	414
Fargate를 사용하여 Amazon ECS에 Java 마이크로서비스 배포	415
요약	415
사전 조건 및 제한 사항	415
아키텍처	415
도구	416
에픽	417
관련 리소스	420
Amazon ECR과 AWS Fargate를 사용하여 Amazon ECS에 Java 마이크로서비스 배포	421
요약	421

사전 조건 및 제한 사항	421
아키텍처	421
도구	422
에픽	423
관련 리소스	427
Amazon ECR 및 로드 밸런싱을 사용하여 Amazon ECS에 Java 마이크로서비스 배포	429
요약	429
사전 조건 및 제한 사항	430
아키텍처	430
도구	431
에픽	431
관련 리소스	432
Amazon EKS와 Helm을 사용하여 Kubernetes 패키지 배포	434
요약	434
사전 조건 및 제한 사항	434
아키텍처	435
도구	435
에픽	436
관련 리소스	443
첨부	443
컨테이너 이미지로 Lambda 함수 배포	444
요약	444
사전 조건 및 제한 사항	444
아키텍처	445
도구	445
모범 사례	446
에픽	446
문제 해결	450
관련 리소스	450
추가 정보	450
Amazon EKS에 Java 마이크로서비스를 배포하고 Application Load Balancer를 사용하여 노출하 기	453
요약	453
사전 조건 및 제한 사항	453
아키텍처	454
도구	454

에픽	455
관련 리소스	461
추가 정보	461
AWS Copilot을 사용하여 클러스터링된 애플리케이션을 Amazon ECS에 배포	465
요약	465
사전 조건 및 제한 사항	465
아키텍처	466
도구	467
에픽	468
관련 리소스	474
Amazon EKS에 gRPC 기반 애플리케이션 배포	476
요약	476
사전 조건 및 제한 사항	476
아키텍처	477
도구	477
에픽	478
관련 리소스	484
추가 정보	485
Amazon EKS 클러스터의 배포 및 디버깅	487
요약	487
사전 조건 및 제한 사항	487
아키텍처	488
도구	489
에픽	490
문제 해결	511
관련 리소스	511
추가 정보	511
Elastic Beanstalk를 사용하여 컨테이너 배포	515
요약	515
사전 조건 및 제한 사항	516
아키텍처	516
도구	517
에픽	517
관련 리소스	520
추가 정보	520
Lambda 및 Amazon VPC를 사용하여 정적 아웃바운드 IP 주소 생성	521

요약	521
사전 조건 및 제한 사항	521
아키텍처	521
도구	522
에픽	523
관련 리소스	532
Amazon EKS 워커 노드에 SSM 에이전트 설치	533
요약	533
사전 조건 및 제한 사항	533
아키텍처	534
도구	534
에픽	536
관련 리소스	537
다음을 사용하여 Amazon EKS 작업자 노드에 SSM CloudWatch 에이전트 및 에이전트를 설치합 니다. preBootstrapCommands	538
요약	538
사전 조건 및 제한 사항	538
아키텍처	539
도구	539
에픽	540
관련 리소스	541
추가 정보	542
생성된 도커 이미지 최적화	545
요약	545
사전 조건 및 제한 사항	545
아키텍처	545
도구	546
에픽	547
관련 리소스	553
첨부	554
Amazon EKS에서 호환 가능한 노드에 Kubernetes 포드 배치	555
요약	555
사전 조건 및 제한 사항	555
아키텍처	556
도구	557
에픽	558

문제 해결	568
관련 리소스	568
추가 정보	569
필터링된 Amazon ECR 컨테이너 이미지를 계정 또는 리전 전반적으로 복제	572
요약	572
사전 조건 및 제한 사항	572
아키텍처	573
도구	573
에픽	576
관련 리소스	587
추가 정보	587
첨부	588
컨테이너를 다시 시작하지 않고 보안 인증 교체	589
요약	589
사전 조건 및 제한 사항	590
아키텍처	590
도구	591
에픽	593
관련 리소스	593
첨부	594
아마존에서 Amazon ECS 작업 실행 WorkSpaces	595
요약	595
사전 조건 및 제한 사항	595
아키텍처	596
도구	596
에픽	597
관련 리소스	604
첨부	604
AWS에서 ASP.NET 웹 API Docker 컨테이너를 실행합니다.	605
요약	605
사전 조건 및 제한 사항	605
아키텍처	606
도구	606
에픽	608
관련 리소스	615
AWS Fargate를 사용하여 메시지 기반 워크로드 실행	616

요약	616
사전 조건 및 제한 사항	616
아키텍처	617
도구	617
에픽	618
관련 리소스	623
영구 데이터 스토리지로 상태 저장 워크로드를 실행합니다.	624
요약	624
사전 조건 및 제한 사항	625
아키텍처	625
도구	626
모범 사례	627
에픽	628
관련 리소스	644
추가 정보	645
패턴 더 보기	646
콘텐츠 전송	647
아마존 데이터 파이어호스를 사용하여 스폰링크에 AWS WAF 로그를 전송	648
요약	648
사전 조건 및 제한 사항	649
아키텍처	649
도구	650
에픽	651
관련 리소스	655
다음을 사용하여 VPC를 통해 S3 버킷의 정적 콘텐츠를 제공합니다. CloudFront	656
요약	656
사전 조건 및 제한 사항	656
아키텍처	657
도구	658
에픽	658
관련 리소스	662
추가 정보	662
패턴 더 보기	665
비용 관리	666
AWS Glue 작업에 대한 자세한 비용 및 사용 보고서 생성	667
요약	667

사전 조건 및 제한 사항	667
아키텍처	667
도구	668
에픽	668
Amazon EMR 클러스터에 대한 자세한 비용 및 사용 보고서를 생성	672
요약	672
사전 조건 및 제한 사항	672
아키텍처	672
도구	673
에픽	673
더 많은 패턴	676
데이터 레이크	677
AWS Data Exchange에서 Amazon S3로 자동으로 데이터 수집	678
요약	678
사전 조건 및 제한 사항	678
아키텍처	679
도구	679
에픽	680
관련 리소스	681
첨부	682
AWS DataOps 개발 키트를 사용하여 Google 애널리틱스 데이터를 처리하기 위한 데이터 파이 프라인을 구축하십시오.	683
요약	683
사전 조건 및 제한 사항	683
아키텍처	684
도구	685
에픽	686
문제 해결	687
관련 리소스	688
추가 정보	688
Athena를 사용하여 공유 AWS Glue 데이터 카탈로그에 대한 크로스 계정 액세스 구성	691
요약	691
사전 조건 및 제한 사항	691
아키텍처	692
도구	693
에픽	693

관련 리소스	704
추가 정보	705
.....	706
요약	706
사전 조건 및 제한 사항	706
아키텍처	707
도구	708
모범 사례	708
에픽	709
관련 리소스	713
추가 정보	713
AWS에서 서버리스 데이터 레이크를 배포하고 관리	714
요약	714
사전 조건 및 제한 사항	715
아키텍처	715
도구	716
에픽	717
관련 리소스	719
IoT 데이터를 Amazon S3로 직접 수집	720
요약	720
사전 조건 및 제한 사항	720
아키텍처	721
도구	722
모범 사례	722
에픽	723
문제 해결	730
관련 리소스	731
추가 정보	731
WanDisco 마이그레이션기를 사용하여 하둡 데이터를 Amazon S3로 마이그레이션합니다.	
LiveData	735
요약	735
사전 조건 및 제한 사항	735
아키텍처	736
에픽	737
관련 리소스	742
추가 정보	742

패턴 더 보기	744
데이터베이스 수	745
연결된 서버를 사용하여 온프레미스 SQL Server 데이터에 액세스합니다.	747
요약	747
사전 조건 및 제한 사항	747
아키텍처	747
도구	748
에픽	748
관련 리소스	751
추가 정보	751
AWS 기반 PeopleSoft 오라클에 HA를 추가합니다.	753
요약	753
사전 조건 및 제한 사항	754
아키텍처	754
도구	755
모범 사례	755
에픽	755
관련 리소스	774
추가 정보	774
SQL Server 데이터베이스를 AWS의 MongoDB Atlas로 마이그레이션하기 위한 쿼리 성능 평 가	777
요약	777
사전 조건 및 제한 사항	777
아키텍처	778
도구	779
모범 사례	779
에픽	779
관련 리소스	785
AWS 계정 전반에 걸쳐 Amazon RDS 인스턴스의 복제를 자동화	786
요약	786
사전 조건 및 제한 사항	786
아키텍처	787
도구	788
에픽	789
관련 리소스	797
추가 정보	797

SAP HANA 데이터베이스 자동으로 백업	800
요약	800
사전 조건 및 제한 사항	800
아키텍처	801
도구	802
에픽	803
관련 리소스	807
Amazon RDS에 대한 퍼블릭 액세스 차단	808
요약	808
사전 조건 및 제한 사항	808
아키텍처	809
도구	809
에픽	810
관련 리소스	813
추가 정보	813
Always On 가용성 그룹에서 읽기 전용 라우팅 구성	815
요약	815
사전 조건 및 제한 사항	815
아키텍처	816
도구	817
모범 사례	817
에픽	817
문제 해결	820
관련 리소스	820
추가 정보	821
pgAdmin에서 SSH 터널을 사용하여 연결	823
요약	823
사전 조건 및 제한 사항	823
아키텍처	824
도구	824
에픽	825
관련 리소스	826
JSON Oracle 쿼리를 PostgreSQL 데이터베이스 SQL로 변환	827
요약	827
사전 조건 및 제한 사항	827
아키텍처	828

도구	829
모범 사례	829
에픽	829
관련 리소스	833
추가 정보	833
계정 전반적으로 Amazon DynamoDB 테이블을 복사합니다.	857
요약	857
사전 조건 및 제한 사항	857
아키텍처	857
도구	858
에픽	858
관련 리소스	862
계정 전반적으로 Amazon DynamoDB 테이블을 복사합니다.	863
요약	863
사전 조건 및 제한 사항	863
아키텍처	864
도구	864
모범 사례	867
에픽	867
관련 리소스	873
추가 정보	873
첨부	873
Amazon RDS 및 Amazon Aurora에 대한 비용 및 사용 보고서 생성	874
요약	874
사전 조건 및 제한 사항	874
아키텍처	874
도구	875
에픽	876
관련 리소스	879
Aurora PostgreSQL을 사용하여 Oracle RAC 워크로드 에뮬레이션하기	880
요약	880
사전 조건 및 제한 사항	880
아키텍처	881
도구	881
에픽	882
관련 리소스	884

PostgreSQL DB 인스턴스에 대한 암호화된 연결 활성화하기	886
요약	886
사전 조건 및 제한 사항	886
아키텍처	886
도구	887
모범 사례	887
에픽	887
문제 해결	893
관련 리소스	893
기존 Amazon RDS for PostgreSQL DB 인스턴스 암호화하기	894
요약	894
사전 조건 및 제한 사항	894
아키텍처	895
도구	895
에픽	896
관련 리소스	899
추가 정보	899
시작 시 Amazon RDS 데이터베이스의 자동 태그 지정 적용	901
요약	901
사전 조건 및 제한 사항	901
아키텍처	902
도구	902
에픽	903
관련 리소스	905
첨부	905
DynamoDB 비용 추정	906
요약	906
사전 조건 및 제한 사항	906
도구	907
모범 사례	907
에픽	908
관련 리소스	912
추가 정보	912
첨부	915
Amazon DynamoDB 테이블의 스토리지 비용 추정	916
요약	916

사전 조건 및 제한 사항	916
도구	917
에픽	917
관련 리소스	918
추가 정보	919
첨부	919
AWR 보고서를 사용하여 Oracle 데이터베이스의 Amazon RDS 엔진 크기 추정	920
요약	920
사전 조건 및 제한 사항	920
아키텍처	921
도구	921
모범 사례	922
에픽	922
관련 리소스	951
Amazon RDS for SQL Server 테이블을 S3 버킷으로 내보내기	952
요약	952
사전 조건 및 제한 사항	953
아키텍처	953
도구	954
에픽	954
관련 리소스	961
추가 정보	961
동적 SQL 명령문에서 익명 블록 처리	963
요약	963
사전 조건 및 제한 사항	963
아키텍처	964
도구	964
에픽	965
관련 리소스	967
추가 정보	968
Aurora PostgreSQL-Compatible에서 오버로드된 Oracle 함수 처리	970
요약	970
사전 조건 및 제한 사항	970
도구	971
에픽	971
관련 리소스	976

DynamoDB 태깅 적용 지원	977
요약	977
사전 조건 및 제한 사항	977
아키텍처	978
도구	978
에픽	979
관련 리소스	981
첨부	982
크로스 리전 DR 구현	983
요약	983
사전 조건 및 제한 사항	983
아키텍처	984
도구	985
에픽	985
관련 리소스	996
추가 정보	997
100개 이상의 인수 Oracle 함수를 PostgreSQL로 마이그레이션	998
요약	998
사전 조건 및 제한 사항	998
아키텍처	999
도구	999
모범 사례	1000
에픽	1000
문제 해결	1002
관련 리소스	1002
추가 정보	1002
Amazon RDS for Oracle DB 인스턴스를 AMS 계정으로 마이그레이션	1004
요약	1004
사전 조건 및 제한 사항	1004
아키텍처	1005
도구	1006
에픽	1007
관련 리소스	1011
추가 정보	1011
Oracle OUT 바인드 변수를 PostgreSQL로 마이그레이션	1013
요약	1013

사전 조건 및 제한 사항	1014
아키텍처	1014
도구	1014
에픽	1015
관련 리소스	1016
추가 정보	1016
HSR을 사용하여 SAP HANA를 AWS로 마이그레이션	1021
요약	1021
사전 조건 및 제한 사항	1022
아키텍처	1023
도구	1024
모범 사례	1025
에픽	1025
관련 리소스	1031
추가 정보	1032
분산된 가용성 그룹을 사용하여 SQL Server를 AWS로 마이그레이션	1033
요약	1033
사전 조건 및 제한 사항	1033
아키텍처	1034
도구	1035
에픽	1035
관련 리소스	1043
AWS DMS를 사용하여 오라클 8i 또는 9i에서 오라클용 아마존 RDS로 마이그레이션할 수 있습니다. SharePlex	1044
요약	1044
사전 조건 및 제한 사항	1044
아키텍처	1045
도구	1046
에픽	1046
관련 리소스	1051
Amazon Aurora의 암호화 모니터링	1052
요약	1052
사전 조건 및 제한 사항	1052
아키텍처	1053
도구	1053
에픽	1054

관련 리소스	1056
첨부	1057
Amazon을 사용하여 GoldenGate 로그를 모니터링하십시오. CloudWatch	1058
요약	1058
사전 조건 및 제한 사항	1058
아키텍처	1059
도구	1059
에픽	1060
문제 해결	1068
관련 리소스	1069
Oracle Database EE를 Amazon RDS for Oracle SE2로 리플랫폼	1070
요약	1070
사전 조건 및 제한 사항	1070
아키텍처	1071
도구	1072
에픽	1073
관련 리소스	1079
Precisely Connect를 사용하여 메인프레임 데이터베이스를 AWS에 복제하기	1080
요약	1080
사전 조건 및 제한 사항	1080
아키텍처	1081
도구	1083
모범 사례	1084
에픽	1085
관련 리소스	1095
Amazon RDS 및 Aurora PostgreSQL 작업 예약하기	1096
요약	1096
사전 조건 및 제한 사항	1096
아키텍처	1097
도구	1097
에픽	1098
관련 리소스	1101
Db2 페더레이션 데이터베이스의 사용자 액세스 보호	1102
요약	1102
사전 조건 및 제한 사항	1102
아키텍처	1103

도구	1103
에픽	1103
관련 리소스	1109
추가 정보	1109
온프레미스 SMTP 서버를 사용하여 RDS for SQL Server에 대한 알림 전송하기	1111
요약	1111
사전 조건 및 제한 사항	1111
아키텍처	1112
도구	1112
에픽	1113
관련 리소스	1121
AWS 기반 IBM Db2에서 SAP용 DR 설정	1122
요약	1122
사전 조건 및 제한 사항	1122
아키텍처	1123
도구	1124
모범 사례	1124
에픽	1125
문제 해결	1141
관련 리소스	1142
추가 정보	1142
Amazon RDS Custom에서 Oracle E-Business Suite를 위한 HA/DR 아키텍처를 설정합니다. ...	1143
요약	1143
사전 조건 및 제한 사항	1143
아키텍처	1144
도구	1145
에픽	1145
관련 리소스	1149
RDS for MySQL와 Amazon EC2의 MySQL 간에 데이터 복제를 설정합니다.	1151
요약	1151
사전 조건 및 제한 사항	1151
아키텍처	1152
도구	1152
에픽	1153
관련 리소스	1156
Oracle PeopleSoft 애플리케이션의 역할 전환	1157

요약	1157
사전 조건 및 제한 사항	1157
아키텍처	1158
도구	1158
모범 사례	1159
에픽	1159
관련 리소스	1188
워크로드별 데이터베이스 마이그레이션 패턴	1189
IBM	1190
Microsoft	1191
해당 사항 없음	1193
오픈 소스	1194
Oracle	1196
SAP	1199
패턴 더 보기	1200
DevOps	1205
AWS 리소스 평가 자동화	1208
요약	1208
사전 조건 및 제한 사항	1208
아키텍처	1209
도구	1210
모범 사례	1211
에픽	1211
문제 해결	1219
관련 리소스	1219
추가 정보	1220
SAP 시스템 설치 자동화	1221
요약	1221
사전 조건 및 제한 사항	1221
아키텍처	1222
도구	1223
에픽	1224
관련 리소스	1231
AWS CDK를 사용하여 Service Catalog 포트폴리오 및 제품 배포 자동화	1232
요약	1232
사전 조건 및 제한 사항	1233

아키텍처	1233
도구	1234
모범 사례	1234
에픽	1235
관련 리소스	1246
추가 정보	1246
AWS에서 Amazon CodeCommit S3로의 백업 자동화	1249
요약	1249
사전 조건 및 제한 사항	1249
아키텍처	1250
도구	1250
에픽	1251
관련 리소스	1254
추가 정보	1254
AWS CodePipeline 및 AWS를 사용하여 스택 세트 배포를 자동화합니다. CodeBuild	1257
요약	1257
사전 조건 및 제한 사항	1257
아키텍처	1258
도구	1259
모범 사례	1260
에픽	1260
문제 해결	1277
관련 리소스	1278
추가 정보	1278
Systems Manager의 관리형 정책을 EC2 인스턴스 프로파일에 자동으로 연결	1285
요약	1285
사전 조건 및 제한 사항	1286
아키텍처	1287
도구	1288
에픽	1288
관련 리소스	1299
첨부	1299
마이크로서비스용 CI/CD 파이프라인 및 Amazon ECS 클러스터 자동으로 구축	1300
요약	1300
사전 조건 및 제한 사항	1300
아키텍처	1301

도구	1302
에픽	1303
관련 리소스	1310
추가 정보	1310
첨부	1311
마이크로서비스와 느슨하게 결합된 아키텍처 구축하기	1312
요약	1312
사전 조건 및 제한 사항	1312
아키텍처	1313
도구	1313
모범 사례	1314
에픽	1314
관련 리소스	1321
추가 정보	1321
도커 이미지를 빌드하여 Amazon ECR로 푸시하기	1323
요약	1323
사전 조건 및 제한 사항	1323
아키텍처	1324
도구	1324
모범 사례	1325
에픽	1325
문제 해결	1328
관련 리소스	1328
AWS 서비스를 사용하여 iOS 앱을 구축하고 테스트합니다.	1329
요약	1329
사전 조건 및 제한 사항	1329
아키텍처	1330
도구	1330
에픽	1331
관련 리소스	1333
규칙 팩을 사용하여 AWS CDK 애플리케이션 또는 CloudFormation 템플릿에서 모범 사례를 확 인하십시오.	1335
요약	1335
사전 조건 및 제한 사항	1335
도구	1336
에픽	1336

관련 리소스	1339
크로스 계정 Amazon DynamoDB 액세스 구성	1340
요약	1340
사전 조건 및 제한 사항	1340
아키텍처	1340
도구	1341
에픽	1341
관련 리소스	1353
추가 정보	1354
Amazon EKS의 애플리케이션에 대한 상호 TLS를 구성합니다.	1356
요약	1356
사전 조건 및 제한 사항	1356
아키텍처	1357
도구	1357
에픽	1357
관련 리소스	1366
Firelens를 사용하여 Amazon ECS용 사용자 지정 로그 구문 분석기를 생성	1367
요약	1367
사전 조건 및 제한 사항	1367
아키텍처	1368
도구	1368
에픽	1369
관련 리소스	1374
첨부	1375
HashiCorp 패커를 사용하여 CodePipeline 파이프라인 및 AMI 생성	1376
요약	1376
사전 조건 및 제한 사항	1376
아키텍처	1376
도구	1377
에픽	1378
관련 리소스	1381
첨부	1382
파이프라인을 생성하고 다음을 사용하여 온프레미스 EC2 인스턴스에 업데이트를 배포합니다.	
CodePipeline	1383
요약	1383
사전 조건 및 제한 사항	1383

아키텍처	1384
도구	1384
에픽	1385
관련 리소스	1390
첨부	1390
Java 및 Python 프로젝트를 위한 동적 CI 파이프라인 생성	1391
요약	1391
사전 조건 및 제한 사항	1391
아키텍처	1392
도구	1393
모범 사례	1394
에픽	1395
관련 리소스	1405
CloudWatch Synthetics 카나리아 배포하기	1406
요약	1406
사전 조건 및 제한 사항	1406
아키텍처	1407
도구	1407
에픽	1408
문제 해결	1410
관련 리소스	1410
추가 정보	1411
Amazon ECS에 Java 마이크로서비스를 위한 CI/CD 파이프라인 배포	1413
요약	1413
사전 조건 및 제한 사항	1413
아키텍처	1413
도구	1415
에픽	1416
관련 리소스	1420
여러 AWS 계정에 CI/CD 파이프라인 배포	1421
요약	1421
사전 조건 및 제한 사항	1422
아키텍처	1422
도구	1422
에픽	1423
관련 리소스	1425

AWS Network Firewall과 AWS Transit Gateway를 사용하여 방화벽 배포	1427
요약	1427
사전 조건 및 제한 사항	1427
아키텍처	1428
도구	1428
에픽	1429
관련 리소스	1439
.....	1440
요약	1440
사전 조건 및 제한 사항	1440
아키텍처	1441
도구	1441
에픽	1442
관련 리소스	1443
첨부	1443
EC2 인스턴스 프로파일을 사용하여 AWS Cloud9에서 Amazon EKS 클러스터의 배포	1444
요약	1444
사전 조건 및 제한 사항	1444
아키텍처	1445
도구	1445
에픽	1446
관련 리소스	1453
첨부	1454
여러 AWS 리전에 코드 배포	1455
요약	1455
사전 조건 및 제한 사항	1455
아키텍처	1456
도구	1456
에픽	1458
관련 리소스	1465
첨부	1465
AWS Backup 보고서를 CSV 파일로 내보내기	1466
요약	1466
사전 조건 및 제한 사항	1466
아키텍처	1467
도구	1468

모범 사례	1468
에픽	1469
관련 리소스	1473
Amazon EC2 인스턴스 태그를 CSV 파일로 내보내기	1475
요약	1475
사전 조건 및 제한 사항	1475
도구	1476
에픽	1476
관련 리소스	1480
AWS Config CloudFormation 관리형 규칙이 포함된 AWS 템플릿을 생성합니다.	1481
요약	1481
사전 조건 및 제한 사항	1481
에픽	1482
첨부	1487
SageMaker 노트북 인스턴스에 리포지토리에 대한 계정 간 액세스 권한을 부여하십시오.	
CodeCommit	1488
요약	1488
사전 조건 및 제한 사항	1488
아키텍처	1489
도구	1489
모범 사례	1490
에픽	1490
관련 리소스	1495
추가 정보	1495
GitHub 플로우 브랜칭 전략 구현	1497
요약	1497
사전 조건 및 제한 사항	1498
아키텍처	1498
도구	1499
모범 사례	1499
에픽	1500
문제 해결	1504
관련 리소스	1504
Gitflow 브랜칭 전략 구현	1506
요약	1506
사전 조건 및 제한 사항	1507

아키텍처	1507
도구	1508
모범 사례	1509
에픽	1509
문제 해결	1515
관련 리소스	1516
트렁크 브랜칭 전략 구현	1517
요약	1517
사전 조건 및 제한 사항	1518
아키텍처	1518
도구	1519
모범 사례	1519
에픽	1520
문제 해결	1521
관련 리소스	1521
모노리포지토리의 변경 사항 감지 후 다양한 CI/CD 파이프라인 시작	1523
요약	1523
사전 조건 및 제한 사항	1523
아키텍처	1524
도구	1525
모범 사례	1525
에픽	1526
문제 해결	1532
관련 리소스	1537
Bitbucket 리포지토리를 AWS Amplify와 통합	1538
요약	1538
사전 조건 및 제한 사항	1538
아키텍처	1538
도구	1539
에픽	1539
관련 리소스	1545
첨부	1545
Lambda를 사용하여 여러 AWS 계정에서 CodeBuild 프로젝트 시작	1546
요약	1546
사전 조건 및 제한 사항	1546
아키텍처	1547

도구	1547
모범 사례	1548
에픽	1549
문제 해결	1556
여러 계정 및 리전에 대한 마이크로서비스의 블루/그린 배포를 관리	1557
요약	1557
사전 조건 및 제한 사항	1558
아키텍처	1559
도구	1559
에픽	1560
문제 해결	1587
관련 리소스	1588
Amazon ECR 리포지토리에서 와일드카드 권한 모니터링	1589
요약	1589
사전 조건 및 제한 사항	1589
아키텍처	1590
도구	1591
에픽	1592
첨부	1593
AWS CodeCommit 이벤트에서 사용자 지정 작업 수행	1594
요약	1594
사전 조건 및 제한 사항	1594
아키텍처	1594
도구	1594
에픽	1595
관련 리소스	1597
Amazon CloudWatch 메트릭을 CSV 파일에 게시	1598
요약	1598
사전 조건 및 제한 사항	1598
도구	1599
에픽	1599
관련 리소스	1601
추가 정보	1602
첨부	1603
AWS Glue에서 Python ETL 작업에 대한 유닛 테스트 실행	1604
요약	1604

사전 조건 및 제한 사항	1604
아키텍처	1605
도구	1606
모범 사례	1607
에픽	1607
문제 해결	1613
관련 리소스	1614
추가 정보	1615
Amazon S3에서 Helm v3 차트 설정	1616
요약	1616
사전 조건 및 제한 사항	1616
아키텍처	1617
도구	1617
에픽	1618
관련 리소스	1623
다음을 사용하여 CI/CD 파이프라인을 설정합니다. CodePipeline	1624
흠	1624
사전 조건 및 제한 사항	1625
아키텍처	1625
도구	1626
모범 사례	1627
에픽	1627
문제 해결	1636
관련 리소스	1636
Amazon EKS에서 애플리케이션을 위한 end-to-end 암호화를 설정합니다.	1638
요약	1638
사전 조건 및 제한 사항	1639
아키텍처	1640
도구	1640
에픽	1641
관련 리소스	1649
Amazon EKS 멀티테넌트 애플리케이션 배포를 간소화합니다.	1650
요약	1650
사전 조건 및 제한 사항	1651
아키텍처	1651
도구	1652

모범 사례	1653
에픽	1653
문제 해결	1665
관련 리소스	1666
추가 정보	1666
여러 이메일 엔드포인트에서 SNS 주제를 구독하십시오.	1667
요약	1667
사전 조건 및 제한 사항	1667
아키텍처	1668
도구	1668
에픽	1669
관련 리소스	1671
첨부	1671
테스트 기반 개발에 Serverspec을 사용합니다	1672
요약	1672
사전 조건 및 제한 사항	1673
아키텍처	1673
도구	1674
에픽	1675
관련 리소스	1677
추가 정보	1677
첨부	1679
AWS의 타사 Git 리포지토리 사용 CodePipeline	1680
요약	1680
사전 조건 및 제한 사항	1680
아키텍처	1681
도구	1681
에픽	1683
관련 리소스	1687
AWS를 사용하여 테라폼 구성을 검증합니다. CodePipeline	1689
요약	1689
사전 조건 및 제한 사항	1689
아키텍처	1690
도구	1691
에픽	1692
문제 해결	1701

관련 리소스	1701
추가 정보	1701
패턴 더 보기	1704
최종 사용자 컴퓨팅	1707
AWS를 사용하여 AppStream 2.0 리소스 생성 CloudFormation	1708
요약	1708
사전 조건 및 제한 사항	1708
아키텍처	1709
도구	1709
에픽	1710
관련 리소스	1711
추가 정보	1712
더 많은 패턴	1714
고성능 컴퓨팅	1715
AWS용 Grafana 모니터링 대시보드 설정 ParallelCluster	1716
요약	1716
사전 조건 및 제한 사항	1716
아키텍처	1717
도구	1718
에픽	1719
문제 해결	1726
관련 리소스	1726
NICE DCV를 사용하여 Auto Scaling VDI 설정	1728
요약	1728
사전 조건 및 제한 사항	1728
아키텍처	1729
도구	1729
에픽	1730
문제 해결	1739
관련 리소스	1740
하이브리드 클라우드	1741
AWS의 VMware Cloud로의 데이터 센터 확장 구성	1742
요약	1742
사전 조건 및 제한 사항	1742
아키텍처	1743
도구	1744

에픽	1744
관련 리소스	1746
AWS의 VMware Cloud에서 VM을 프로비저닝하도록 vRealize Automation을 구성합니다.	1747
요약	1747
사전 조건 및 제한 사항	1747
아키텍처	1748
도구	1750
에픽	1750
관련 리소스	1756
AWS의 VMware Cloud를 사용하여 SDDC 배포	1757
요약	1757
사전 조건 및 제한 사항	1757
아키텍처	1758
도구	1759
에픽	1759
관련 리소스	1764
VMware vRealize 네트워크 인사이트를 VMware Cloud on AWS와 통합	1765
요약	1765
사전 조건 및 제한 사항	1765
아키텍처	1766
도구	1766
에픽	1767
관련 리소스	1769
HCX OSAM을 사용하여 VM을 AWS의 VMware Cloud로 마이그레이션	1770
요약	1770
사전 조건 및 제한 사항	1770
아키텍처	1771
도구	1772
에픽	1772
관련 리소스	1774
AWS 기반 VMware 클라우드에서 스폰링크로 로그 전송	1775
요약	1775
사전 조건 및 제한 사항	1775
아키텍처	1776
도구	1777
에픽	1777

관련 리소스	1780
Amazon ECS Anywhere에서 하이브리드 워크로드를 위한 CI/CD 파이프라인 설정	1781
요약	1781
사전 조건 및 제한 사항	1781
아키텍처	1782
도구	1784
모범 사례	1785
에픽	1785
문제 해결	1797
관련 리소스	1798
더 많은 패턴	1799
인프라	1800
Session Manager 및 Amazon EC2 인스턴스 연결을 사용하여 Bastion Host에 액세스	1801
요약	1801
사전 조건 및 제한 사항	1802
아키텍처	1803
도구	1804
모범 사례	1805
에픽	1806
문제 해결	1812
관련 리소스	1812
추가 정보	1813
AWS Managed Microsoft AD를 사용하여 DNS 확인 중앙 집중화	1815
요약	1815
사전 조건 및 제한 사항	1815
아키텍처	1816
도구	1817
에픽	1817
관련 리소스	1824
Observability Access Manager를 사용하여 모니터링 중앙 집중화	1826
요약	1826
사전 조건 및 제한 사항	1827
아키텍처	1827
도구	1828
모범 사례	1829
에픽	1829

관련 리소스	1837
시작 시 EC2 인스턴스에 필수 태그가 있는지 확인	1838
요약	1838
사전 조건 및 제한 사항	1838
아키텍처	1839
도구	1839
에픽	1840
관련 리소스	1842
첨부	1842
Session Manager를 사용하여 EC2 인스턴스에 연결	1843
요약	1843
사전 조건 및 제한 사항	1843
아키텍처	1844
도구	1844
모범 사례	1845
에픽	1845
문제 해결	1848
관련 리소스	1848
AWS를 지원하지 않는 AWS 지역에 파이프라인을 생성하십시오. CodePipeline	1849
요약	1849
사전 조건 및 제한 사항	1849
아키텍처	1849
도구	1850
에픽	1851
관련 리소스	1855
프라이빗 고정 IP로 Amazon EC2에 Cassandra 클러스터 배포	1856
요약	1856
사전 조건 및 제한 사항	1856
아키텍처	1857
에픽	1857
관련 리소스	1862
Transit Gateway Connect를 사용하여 VRF를 AWS로 확장	1863
요약	1863
사전 조건 및 제한 사항	1864
아키텍처	1864
도구	1867

에픽	1868
관련 리소스	1877
첨부	1878
AWS KMS 키의 상태 변경에 대한 Amazon SNS 알림 받기	1879
요약	1879
사전 조건 및 제한 사항	1879
아키텍처	1879
도구	1880
에픽	1881
관련 리소스	1884
추가 정보	1884
Micro Focus를 사용하여 메인프레임 환경을 현대화하기	1885
요약	1885
사전 조건 및 제한 사항	1887
아키텍처	1889
도구	1895
에픽	1896
관련 리소스	1900
비 워크로드 서브넷을 위한 다중 계정 VPC 설계에서 라우팅 가능한 IP 공간 보존	1902
요약	1902
사전 조건 및 제한 사항	1902
아키텍처	1902
도구	1903
모범 사례	1904
에픽	1905
관련 리소스	1906
추가 정보	1906
코드 리포지토리에서 Service Catalog의 Terraform 제품 프로비저닝	1907
요약	1907
사전 조건 및 제한 사항	1908
아키텍처	1908
도구	1908
모범 사례	1909
에픽	1909
관련 리소스	1921
추가 정보	1922

단일 이메일 주소로 여러 AWS 계정 등록	1925
요약	1925
사전 조건 및 제한 사항	1925
아키텍처	1926
도구	1927
에픽	1928
문제 해결	1935
관련 리소스	1937
추가 정보	1937
다중 계정 AWS 환경에서 하이브리드 네트워크를 위한 DNS 확인 설정	1939
요약	1939
사전 조건 및 제한 사항	1939
아키텍처	1940
도구	1940
에픽	1941
관련 리소스	1944
단일 계정 AWS 환경에서 하이브리드 네트워크를 위한 DNS 확인 설정	1945
요약	1945
사전 조건 및 제한 사항	1945
아키텍처	1946
도구	1946
에픽	1946
관련 리소스	1949
Amazon EC2에서 UiPath RPA 봇을 자동으로 설정합니다.	1950
요약	1950
사전 조건 및 제한 사항	1951
아키텍처	1951
도구	1952
모범 사례	1953
에픽	1953
문제 해결	1963
관련 리소스	1963
오라클 JD 에드워즈를 위한 재해 복구 설정 EnterpriseOne	1965
요약	1965
사전 조건 및 제한 사항	1966
아키텍처	1967

도구	1969
모범 사례	1969
에픽	1970
문제 해결	1987
관련 리소스	1988
SAP Pacemaker 클러스터를 ENSA1에서 ENSA2 클러스터로 업그레이드	1989
요약	1989
사전 조건 및 제한 사항	1990
아키텍처	1990
도구	1992
모범 사례	1992
에픽	1992
관련 리소스	2009
여러 계정의 VPC에서 일관된 가용 영역 사용	2010
요약	2010
사전 조건 및 제한 사항	2010
아키텍처	2011
도구	2012
에픽	2013
관련 리소스	2014
Account Factory에서 Terraform 코드를 로컬에서 검증합니다.	2015
요약	2015
사전 조건 및 제한 사항	2015
아키텍처	2016
도구	2016
에픽	2018
패턴 더 보기	2030
IoT	2033
IoT 환경에서 보안 이벤트에 대한 로깅 및 모니터링을 구성합니다.	2034
요약	2034
사전 조건 및 제한 사항	2035
아키텍처	2035
도구	2037
에픽	2038
관련 리소스	2042
AWS IoT SiteWise 메타데이터 속성 추출 및 쿼리	2043

요약	2043
사전 조건 및 제한 사항	2043
아키텍처	2044
도구	2044
에픽	2045
관련 리소스	2047
추가 정보	2048
.....	2050
요약	2050
사전 조건 및 제한 사항	2051
아키텍처	2051
도구	2052
모범 사례	2052
에픽	2053
문제 해결	2067
관련 리소스	2068
추가 정보	2069
더 많은 패턴	2071
기계 학습 및 AI	2072
Athena의 ML 예측을 위한 DynamoDB의 데이터 집계	2073
요약	2073
사전 조건 및 제한 사항	2073
아키텍처	2074
도구	2075
에픽	2076
관련 리소스	2085
여러 계정에서 AWS CodeCommit 리포지토리를 Amazon SageMaker 스튜디오와 연결	2086
요약	2086
사전 조건 및 제한 사항	2086
아키텍처	2087
도구	2087
에픽	2088
추가 정보	2093
Amazon Lookout for Vision 모델 교육을 자동화하세요	2096
요약	2096
사전 조건 및 제한 사항	2097

아키텍처	2097
도구	2098
모범 사례	2098
에픽	2099
관련 리소스	2101
PDF 파일에서 콘텐츠 자동 추출하기	2103
요약	2103
사전 조건 및 제한 사항	2103
아키텍처	2104
도구	2105
에픽	2105
관련 리소스	2109
첨부	2110
Azure를 사용하여 MLOps 워크플로를 구축하세요. SageMaker DevOps	2111
요약	2111
사전 조건 및 제한 사항	2111
아키텍처	2112
도구	2113
모범 사례	2114
에픽	2115
문제 해결	2122
관련 리소스	2122
Step Functions에서 모델 학습을 SageMaker 위해 Docker 컨테이너를 생성합니다.	2124
요약	2124
사전 조건 및 제한 사항	2124
아키텍처	2125
도구	2125
에픽	2126
관련 리소스	2136
단일 엔드포인트에 여러 파이프라인 모델 객체를 배포합니다. SageMaker	2138
요약	2138
사전 조건 및 제한 사항	2138
아키텍처	2139
도구	2139
에픽	2140
관련 리소스	2149

RAG 및 프롬프트를 사용하여 AI 채팅 기반 어시스턴트 개발 ReAct	2150
요약	2150
사전 조건 및 제한 사항	2151
아키텍처	2152
도구	2153
모범 사례	2154
에픽	2155
문제 해결	2161
관련 리소스	2161
추가 정보	2161
Amazon Bedrock을 사용하여 채팅 기반 어시스턴트를 개발하십시오.	2163
요약	2163
사전 조건 및 제한 사항	2164
아키텍처	2164
도구	2165
모범 사례	2167
에픽	2168
관련 리소스	2171
추가 정보	2172
Amazon Personalize를 사용하여 맞춤형 추천 생성	2174
요약	2174
사전 조건 및 제한 사항	2174
아키텍처	2175
도구	2176
에픽	2177
관련 리소스	2179
추가 정보	2179
사용자 지정 GPU 지원 ML 모델 교육 및 배포	2183
요약	2183
사전 조건 및 제한 사항	2183
아키텍처	2184
도구	2184
에픽	2184
관련 리소스	2199
추가 정보	2199

테라바이트 규모의 ML 데이터세트의 분산 기능 엔지니어링에는 SageMaker 프로세싱을 사용하 십시오.	2202
요약	2202
사전 조건 및 제한 사항	2202
아키텍처	2203
도구	2205
에픽	2206
관련 리소스	2216
첨부	2216
Flask와 Elastic Beanstalk를 사용하여 AI/ML 모델 결과를 시각화합니다.	2217
요약	2217
사전 조건 및 제한 사항	2217
아키텍처	2218
도구	2220
에픽	2220
관련 리소스	2229
추가 정보	2229
더 많은 패턴	2233
메인프레임	2234
메인프레임 데이터를 Amazon S3에 백업하고 보관하기	2235
요약	2235
사전 조건 및 제한 사항	2235
아키텍처	2236
도구	2237
에픽	2238
관련 리소스	2257
AWS 클라우드에서 메인프레임 파일 뷰어 구축	2259
요약	2259
사전 조건 및 제한 사항	2259
아키텍처	2260
도구	2261
에픽	2262
관련 리소스	2270
추가 정보	2270
현대화된 Blu Age 애플리케이션 컨테이너화	2272
요약	2272

사전 조건 및 제한 사항	2272
아키텍처	2273
도구	2274
모범 사례	2274
에픽	2275
관련 리소스	2280
EBCDIC 데이터를 AWS에서 ASCII로 변환	2282
요약	2282
사전 조건 및 제한 사항	2282
아키텍처	2283
도구	2284
에픽	2285
관련 리소스	2298
AWS Lambda를 사용하여 메인프레임 EBCDIC 파일을 ASCII 파일로 변환	2300
요약	2300
사전 조건 및 제한 사항	2300
아키텍처	2301
도구	2302
모범 사례	2303
에픽	2303
관련 리소스	2316
복잡한 레코드 레이아웃의 메인프레임 데이터 파일 변환	2317
요약	2317
사전 조건 및 제한 사항	2317
도구	2318
에픽	2318
관련 리소스	2332
컨테이너화된 앱을 위한 환경 배포	2333
요약	2333
사전 조건 및 제한 사항	2334
아키텍처	2334
도구	2337
모범 사례	2338
에픽	2338
관련 리소스	2342
Stonebranch 유니버설 컨트롤러를 AWS와 통합	2343

요약	2343
사전 조건 및 제한 사항	2344
아키텍처	2345
도구	2349
에픽	2350
관련 리소스	2373
추가 정보	2373
Precisely를 사용하여 VSAM 파일을 AWS Cloud로 마이그레이션하고 복제하기	2374
요약	2374
사전 조건 및 제한 사항	2374
아키텍처	2375
도구	2377
에픽	2378
관련 리소스	2387
추가 정보	2387
AWS의 메인프레임 출력 관리 현대화	2390
요약	2390
사전 조건 및 제한 사항	2391
아키텍처	2391
도구	2395
에픽	2397
관련 리소스	2429
추가 정보	2429
첨부	2431
AWS에서 메인프레임 배치 인쇄 워크로드를 현대화	2432
요약	2432
사전 조건 및 제한 사항	2432
아키텍처	2433
도구	2436
에픽	2437
관련 리소스	2455
추가 정보	2456
첨부	2457
AWS에서 메인프레임 온라인 인쇄 워크로드를 현대화	2458
요약	2458
사전 조건 및 제한 사항	2458

아키텍처	2459
도구	2462
에픽	2463
관련 리소스	2483
추가 정보	2484
첨부	2486
Transfer Family를 사용하여 메인프레임 파일을 Amazon S3로 이동	2487
요약	2487
사전 조건 및 제한 사항	2487
아키텍처	2488
도구	2489
모범 사례	2489
에픽	2490
관련 리소스	2498
Db2 z/OS 데이터를 AWS로 전송	2499
요약	2499
사전 조건 및 제한 사항	2500
아키텍처	2500
도구	2501
모범 사례	2503
에픽	2503
관련 리소스	2523
추가 정보	2523
더 많은 패턴	2525
관리 및 거버넌스	2526
Data Firehose 리소스가 암호화되지 않은 경우 알림	2527
요약	2527
사전 조건 및 제한 사항	2527
아키텍처	2528
도구	2528
에픽	2529
관련 리소스	2530
추가 정보	2531
첨부	2531
Windows 레지스트리 항목의 추가 또는 업데이트 자동화	2532
요약	2532

사전 조건 및 제한 사항	2532
아키텍처	2532
도구	2533
에픽	2534
관련 리소스	2535
첨부	2536
aws-nuke를 사용하여 AWS 리소스 자동 삭제	2537
요약	2537
사전 조건 및 제한 사항	2539
아키텍처	2540
도구	2541
에픽	2543
관련 리소스	2550
추가 정보	2551
Amazon RDS DB 인스턴스의 자동 중지 및 시작	2555
요약	2555
사전 조건 및 제한 사항	2555
아키텍처	2556
도구	2557
에픽	2557
관련 리소스	2566
Terraform을 사용하여 AWS Organizations에서 소프트웨어 패키지 배포를 중앙 집중화합니 다.	2567
요약	2567
사전 조건 및 제한 사항	2567
아키텍처	2568
도구	2569
모범 사례	2570
에픽	2570
문제 해결	2577
관련 리소스	2577
계정 간 VPC 흐름 로그 구성	2578
요약	2578
사전 조건 및 제한 사항	2578
아키텍처	2579
도구	2580

모범 사례	2580
에픽	2583
관련 리소스	2584
추가 정보	2584
로그에서.NET 애플리케이션에 대한 로깅을 구성합니다. CloudWatch	2587
요약	2587
사전 조건 및 제한 사항	2587
아키텍처	2588
도구	2588
모범 사례	2589
에픽	2589
문제 해결	2593
관련 리소스	2593
추가 정보	2594
AWS 계정 및 리전 전체에서 AWS Service Catalog 제품 복사	2595
요약	2595
사전 조건 및 제한 사항	2595
아키텍처	2596
도구	2596
에픽	2598
관련 리소스	2602
첨부	2602
다음을 사용하여 사용자 지정 지표에 대한 경보를 생성합니다. CloudWatch	2603
요약	2603
사전 조건 및 제한 사항	2603
아키텍처	2604
도구	2604
에픽	2605
관련 리소스	2608
첨부	2608
랜딩 존 설계를 문서화하세요	2609
요약	2609
사전 조건 및 제한 사항	2609
에픽	2610
관련 리소스	2611
첨부	2611

드리프트 감지 및 보고	2612
요약	2612
사전 조건 및 제한 사항	2612
아키텍처	2613
도구	2613
에픽	2614
관련 리소스	2616
추가 정보	2616
첨부	2616
AWS CDK를 사용하여 조직 전체에서 Amazon DevOps Guru를 지원할 수 있습니다.	2617
요약	2617
사전 조건 및 제한 사항	2618
아키텍처	2618
도구	2620
에픽	2620
관련 리소스	2641
부트스트랩 파이프라인을 사용하여 AFT를 구현하세요.	2642
요약	2642
사전 조건 및 제한 사항	2642
아키텍처	2643
도구	2646
모범 사례	2647
에픽	2647
문제 해결	2657
관련 리소스	2658
여러 AWS 계정 및 리전의 AWS Service Catalog 제품을 관리	2659
요약	2659
사전 조건 및 제한 사항	2660
아키텍처	2660
도구	2660
에픽	2661
관련 리소스	2664
추가 정보	2665
AWS Organizations의 AWS 계정을 AWS Control Tower로 마이그레이션	2666
요약	2666
사전 조건 및 제한 사항	2666

아키텍처	2667
도구	2667
에픽	2668
문제 해결	2675
관련 리소스	2676
AWS 계정 전반의 AMI 사용 모니터링	2677
요약	2677
사전 조건 및 제한 사항	2677
아키텍처	2678
도구	2679
모범 사례	2680
에픽	2681
문제 해결	2691
관련 리소스	2691
AWS Organizations의 프로그래밍 방식 계정 폐쇄에 대한 알림 설정	2693
요약	2693
사전 조건 및 제한 사항	2693
아키텍처	2694
도구	2695
에픽	2696
관련 리소스	2700
패턴 더 보기	2701
메시징 및 커뮤니케이션	2703
Amazon MQ에서 RabbitMQ 구성의 자동화	2704
요약	2704
사전 조건 및 제한 사항	2704
아키텍처	2705
도구	2705
에픽	2706
관련 리소스	2710
첨부	2710
Amazon Connect의 에이전트 워크스테이션의 통화 품질을 개선합니다.	2711
요약	2711
사전 조건 및 제한 사항	2712
아키텍처	2712
도구	2712

에픽	2713
관련 리소스	2723
패턴 더 보기	2724
마이그레이션	2725
마이그레이션 전략 식별 및 계획 자동화	2726
요약	2726
사전 조건 및 제한 사항	2727
아키텍처	2727
도구	2728
에픽	2728
관련 리소스	2732
AWS CloudFormation DMS용 AWS 템플릿 생성	2734
요약	2734
사전 조건 및 제한 사항	2734
아키텍처	2735
도구	2735
에픽	2736
관련 리소스	2737
자동화된 포트폴리오 검색으로 시작하기	2738
요약	2738
에픽	2738
관련 리소스	2743
추가 정보	2743
첨부	2744
온프레미스 Cloudera 워크로드를 AWS로 마이그레이션	2745
요약	2745
사전 조건 및 제한 사항	2749
아키텍처	2750
도구	2751
에픽	2752
관련 리소스	2759
SELinux를 비활성화하지 않고 Replication Agent를 자동으로 다시 시작	2761
요약	2761
사전 조건 및 제한 사항	2761
도구	2762
에픽	2763

관련 리소스	2767
리아키텍트	2768
VARCHAR2(1) 데이터 유형을 부울 데이터 유형으로 변환	2770
Aurora PostgreSQL 호환에서 사용자 및 역할 생성	2780
Aurora 글로벌 데이터베이스를 사용하여 Oracle DR 에뮬레이션하기	2794
Amazon RDS for Oracle에서 Amazon RDS for PostgreSQL로 점진적으로 마이그레이션 ...	2800
BLOB 파일을 Aurora PostgreSQL-Compatible에 로드	2807
SSL 모드에서 Amazon RDS for Oracle을 Amazon RDS for PostgreSQL로 마이그레이션 ..	2822
AWS SCT 및 AWS DMS를 사용하여 Amazon RDS for Oracle을 Amazon RDS for PostgreSQL로 마이그레이션	2846
Oracle SERIALLY_REUSABLE 프라그마 패키지를 AWS로 마이그레이션	2860
Oracle 외부 테이블을 Amazon Aurora로 마이그레이션	2867
Oracle 함수 기반 인덱스 마이그레이션	2892
Oracle 네이티브 함수를 PostgreSQL로 마이그레이션	2898
Db2 데이터베이스를 Amazon EC2에서 Aurora MySQL과 호환되는 Aurora로 마이그레이션	2906
Amazon EC2에서 Amazon DocumentDB로 SQL 서버 데이터베이스를 마이그레이션	2921
ThoughtSpot 팔콘 데이터베이스를 아마존 Redshift로 마이그레이션하기	2929
Amazon DynamoDB로 Oracle 데이터베이스 마이그레이션	2941
AWS DMS를 사용하여 Oracle 파티션형 테이블을 PostgreSQL로 마이그레이션하기	2946
Oracle에서 Amazon RDS for Oracle로 마이그레이션	2950
IBM Db2에서 Aurora PostgreSQL 호환 버전으로 마이그레이션	2958
Quest를 사용하여 Oracle 8i/9i에서 PostgreSQL용 Amazon RDS로 마이그레이션하십시오.	
SharePlex	2966
구체화된 뷰를 사용하여 Oracle 8i/9i에서 Amazon RDS for PostgreSQL로 마이그레이션 ...	2976
Amazon EC2 기반 오라클에서 Amazon RDS for MySQL로 마이그레이션하기	2987
Oracle에서 Amazon DocumentDB로 마이그레이션	2996
Oracle에서 Amazon RDS for MariaDB로 마이그레이션	3002
Oracle에서 Amazon RDS for MySQL로 마이그레이션	3011
Oracle에서 Amazon RDS for PostgreSQL로 마이그레이션	3016
오라클을 사용하여 PostgreSQL용 오라클에서 Amazon RDS로 마이그레이션하십시오.	
GoldenGate	3028
Oracle에서 Amazon Redshift로 마이그레이션	3036
Oracle에서 Aurora PostgreSQL-Compatible로 마이그레이션하기	3045
standby가 포함된 Oracle에서 Aurora PostgreSQL로 마이그레이션하기	3055
SAP ASE에서 Amazon RDS for SQL Server로 마이그레이션	3065

SQL Server에서 Amazon Redshift로 마이그레이션	3070
데이터 추출 에이전트를 사용하여 SQL Server에서 Amazon Redshift로 마이그레이션	3075
데이터 추출 에이전트를 사용하여 Teradata에서 Amazon Redshift로 마이그레이션	3080
데이터 추출 에이전트를 사용하여 Vertica에서 Amazon Redshift로 마이그레이션하기	3085
레거시 애플리케이션을 Oracle Pro*C에서 ECPG로 마이그레이션	3090
가상으로 생성된 열을 오라클에서 PostgreSQL로 마이그레이션	3108
Amazon Aurora에서 Oracle UTL_FILE 기능 설정	3115
.....	3130
리호스팅	3138
AWS로의 Microsoft 워크로드 마이그레이션을 가속화하세요	3139
사전 워크로드 수집 활동 자동화	3149
마이그레이션 중 방화벽 요청에 대한 승인 프로세스를 생성	3157
EC2 Windows 인스턴스를 AMS 계정으로 수집	3162
로그 전달을 사용하여 Db2를 Amazon EC2로 마이그레이션합니다.	3170
HADR을 사용하여 Db2를 Amazon EC2로 마이그레이션하세요.	3187
PowerCLI를 사용하여 HCX 자동화를 통해 VMware VM 마이그레이션	3221
F5 BIG-IP 워크로드를 F5 BIG-IP VE로 마이그레이션	3232
온프레미스 Go 애플리케이션을 AWS Elastic Beanstalk로 마이그레이션	3241
.....	3246
온프레미스 VM을 AWS로 마이그레이션	3254
AWS SFTP를 사용하여 Amazon S3로 데이터 마이그레이션	3265
오라클에서 AWS Elastic GlassFish Beanstalk로 마이그레이션	3270
Oracle에서 Amazon EC2로 마이그레이션	3276
Oracle Data Pump를 사용하여 Oracle에서 Amazon EC2로 마이그레이션	3283
SAP ASE에서 Amazon EC2로 마이그레이션	3291
SQL 서버에서 Amazon EC2로 마이그레이션	3297
온프레미스 MySQL을 Amazon EC2로 마이그레이션	3304
동종 SAP 마이그레이션 전환 시간 단축	3310
AWS의 온프레미스 워크로드 리호스팅: 마이그레이션 체크리스트	3318
SQL Server Always On FCI용 다중 AZ 인프라 설정	3332
BMC Discovery를 사용하여 마이그레이션 계획 데이터 추출	3351
재배치	3361
Amazon RDS for Oracle을 다른 AWS 리전 및 계정으로 마이그레이션	3362
VMware SDDC를 AWS의 VMware Cloud로 마이그레이션	3371
Amazon RDS DB 인스턴스를 다른 VPC 또는 계정으로 마이그레이션	3375
Amazon RDS for Oracle DB를 다른 VPC로 마이그레이션	3382

.....	3387
VMware HCX를 사용하여 워크로드를 AWS의 VMware Cloud로 마이그레이션합니다.	3401
Amazon RDS DB 인스턴스 간에 PostgreSQL 데이터베이스를 전송	3432
리플랫폼	3443
Oracle Database와 Aurora 간의 링크 구성	3445
Microsoft SQL Server 데이터베이스를 Amazon S3로 내보내기	3480
ML, 빌드, 교육 및 워크로드를 Amazon으로 마이그레이션 SageMaker	3486
AWS로 OpenText TeamSite 워크로드 마이그레이션	3492
PostgreSQL의 개별 행으로 Oracle CLOB 값을 마이그레이션	3512
Oracle Data Pump와 데이터베이스 링크로 Oracle 데이터베이스를 마이그레이션합니다. ...	3519
Oracle E-Business Suite를 Amazon RDS Custom으로 마이그레이션	3535
PeopleSoft 오라클을 Amazon RDS 커스텀으로 마이그레이션	3627
Oracle ROWID 기능을 PostgreSQL로 마이그레이션	3654
Oracle 오류 코드를 Amazon Aurora PostgreSQL-Compatible 데이터베이스로 마이그레이 션	3666
Redis 워크로드를 AWS의 Redis Enterprise Cloud로 마이그레이션	3672
Amazon EC2에서 SAP ASE를 Aurora PostgreSQL-Compatible로 마이그레이션하기	3697
ACM을 사용하여 Windows SSL 인증서를 Application Load Balancer로 마이그레이션	3706
메시지 대기열을 마이크로소프트 Azure에서 Amazon SQS로 마이그레이션	3715
오라클 JD 에드워즈 EnterpriseOne 데이터베이스를 AWS로 마이그레이션	3722
Oracle PeopleSoft 데이터베이스를 AWS로 마이그레이션	3749
온프레미스 MySQL 데이터베이스를 Amazon RDS for MySQL로 마이그레이션	3772
온프레미스 SQL Server 데이터베이스를 Amazon RDS for SQL Server로 마이그레이션	3780
Azure Blob에서 Amazon S3로 데이터 마이그레이션하기	3785
카우치베이스 서버에서 카우치베이스 카펠라로 마이그레이션	3795
IBM에서 Amazon WebSphere EC2의 아파치 톰캣으로 마이그레이션하기	3824
WebSphere Auto Scaling을 사용하여 IBM에서 Amazon EC2의 아파치 톰캣으로 마이그레이 션하세요	3831
Microsoft Azure 앱 서비스에서 AWS Elastic Beanstalk로 마이그레이션	3838
MongoDB에서 AWS의 MongoDB Atlas로 마이그레이션	3844
Amazon ECS를 통해 오라클에서 WebLogic ToMee로 마이그레이션	3853
Amazon EC2 기반 Oracle에서 Amazon RDS for Oracle로 마이그레이션	3862
Logstash를 사용하여 오라클에서 아마존 OpenSearch 서비스로 마이그레이션하기	3868
Oracle에서 Amazon RDS for Oracle로 마이그레이션	3875
Oracle Data Pump를 사용하여 Oracle에서 Amazon RDS로 마이그레이션	3887
Amazon EC2의 PostgreSQL에서 Amazon RDS for PostgreSQL로 마이그레이션합니다. ...	3897

PostgreSQL에서 Aurora PostgreSQL로 마이그레이션	3903
Windows의 SQL Server에서 Amazon EC2의 Linux로 마이그레이션	3913
연결된 서버를 사용하여 SQL Server에서 Amazon RDS for SQL Server로 마이그레이션 ...	3917
기본 백업 및 복원을 사용하여 SQL Server에서 Amazon RDS for SQL Server로 마이그레이 션	3922
SQL Server에서 Aurora MySQL로 마이그레이션	3927
온프레미스 MariaDB에서 Amazon RDS for MariaDB로 마이그레이션	3936
온프레미스 MySQL 데이터베이스를 Aurora MySQL로 마이그레이션	3941
Percona를 사용하여 온프레미스 MySQL에서 Aurora MySQL로 마이그레이션 XtraBackup	3946
App2Container를 사용하여 온프레미스 애플리케이션 마이그레이션	3960
AWS 대규모 마이그레이션에서 공유 파일 시스템 마이그레이션	3971
Oracle GoldenGate 플랫폼 파일 어댑터를 사용하여 Amazon RDS로 마이그레이션하기	3996
데이터베이스 마이그레이션을 지원하도록 Python 및 Perl 애플리케이션 변경	4002
워크로드별 마이그레이션 패턴	4034
IBM	4035
Microsoft	4036
해당 사항 없음	4037
오픈 소스	4038
Oracle	4039
SAP	4042
패턴 더 보기	4043
현대화	4045
CAST Imaging의 소프트웨어 아키텍처 분석 및 시각화	4046
요약	4046
사전 조건 및 제한 사항	4046
아키텍처	4047
도구	4047
에픽	4047
관련 리소스	4053
CAST Highlight를 사용하여 AWS로 마이그레이션하기 전에 애플리케이션 준비 상태 평가	4055
요약	4055
사전 조건 및 제한 사항	4055
아키텍처	4056
도구	4057
에픽	4057
관련 리소스	4073

만료된 DynamoDB 데이터를 Amazon S3에 자동 보관	4074
요약	4074
사전 조건 및 제한 사항	4075
아키텍처	4075
도구	4075
에픽	4076
관련 리소스	4086
추가 정보	4087
Micro Focus Enterprise Server PAC 빌드	4089
요약	4089
사전 조건 및 제한 사항	4089
아키텍처	4090
도구	4094
에픽	4095
관련 리소스	4098
추가 정보	4099
Amazon Service에서 멀티테넌트 서버리스 아키텍처 구축 OpenSearch	4107
요약	4107
사전 조건 및 제한 사항	4108
아키텍처	4108
도구	4109
에픽	4110
관련 리소스	4149
추가 정보	4150
첨부	4153
다중 스택 애플리케이션 배포	4154
요약	4154
사전 조건 및 제한 사항	4154
아키텍처	4155
도구	4156
에픽	4157
관련 리소스	4160
추가 정보	4161
첨부	4162
AWS SAM을 사용하여 중첩된 애플리케이션 배포	4163
요약	4163

사전 조건 및 제한 사항	4163
아키텍처	4164
도구	4165
에픽	4166
관련 리소스	4170
추가 정보	4170
AWS Lambda TVM을 사용하여 Amazon S3에 대한 SaaS 테넌트 격리를 구현합니다.	4171
요약	4171
사전 조건 및 제한 사항	4171
아키텍처	4172
도구	4172
에픽	4173
관련 리소스	4193
추가 정보	4193
첨부	4193
AWS Step Functions을 사용하여 서버리스 사가 패턴 구현	4194
요약	4194
사전 조건 및 제한 사항	4195
아키텍처	4195
도구	4196
에픽	4197
관련 리소스	4202
추가 정보	4203
Amazon ECS Anywhere를 사용한 온프레미스 컨테이너 애플리케이션 관리	4208
요약	4208
사전 조건 및 제한 사항	4208
아키텍처	4209
도구	4210
에픽	4210
관련 리소스	4217
AWS에서 ASP.NET Web Forms 애플리케이션 현대화	4218
요약	4218
사전 조건 및 제한 사항	4219
아키텍처	4219
도구	4220
에픽	4221

관련 리소스	4228
추가 정보	4229
AWS Fargate를 사용하여 이벤트 기반 워크로드 실행	4231
요약	4231
사전 조건 및 제한 사항	4231
아키텍처	4232
도구	4233
에픽	4234
관련 리소스	4237
추가 정보	4238
첨부	4239
SaaS 아키텍처의 테넌트 온보딩	4240
요약	4240
사전 조건 및 제한 사항	4240
아키텍처	4242
도구	4244
에픽	4246
관련 리소스	4261
추가 정보	4262
CQRS 및 이벤트 소싱 사용	4265
요약	4265
사전 조건 및 제한 사항	4265
아키텍처	4266
도구	4267
에픽	4268
관련 리소스	4280
추가 정보	4280
첨부	4287
패턴 더 보기	4288
네트워킹	4290
AWS Transit Gateway의 자동 피어링	4291
요약	4291
사전 조건 및 제한 사항	4291
아키텍처	4292
도구	4293
에픽	4293

관련 리소스	4297
첨부	4297
AWS Transit Gateway를 사용하여 네트워크 연결 중앙 집중화	4298
요약	4298
사전 조건 및 제한 사항	4298
아키텍처	4298
도구	4299
에픽	4299
관련 리소스	4303
애플리케이션 로드 밸런서를 사용하여 Oracle JD EnterpriseOne Edwards용 HTTPS 암호화를 구성합니다.	4304
요약	4304
사전 조건 및 제한 사항	4305
아키텍처	4305
도구	4305
모범 사례	4305
에픽	4306
문제 해결	4312
관련 리소스	4313
프라이빗 네트워크를 통해 Application Migration Service 데이터 및 컨트롤 플레인에 연결	4314
요약	4314
사전 조건 및 제한 사항	4314
아키텍처	4316
도구	4316
에픽	4317
관련 리소스	4325
추가 정보	4325
AWS 사용자 지정 리소스를 사용하여 Infoblox 객체 생성 CloudFormation	4326
요약	4326
사전 조건 및 제한 사항	4327
아키텍처	4328
도구	4329
에픽	4332
관련 리소스	4336
첨부	4337
Network Firewall에 대한 CloudWatch 알림 사용자 지정	4338

요약	4338
사전 조건 및 제한 사항	4338
아키텍처	4339
도구	4339
에픽	4340
관련 리소스	4355
추가 정보	4355
Route 53 프라이빗 호스팅 영역으로 DNS 레코드 대량 이전	4357
요약	4357
사전 조건 및 제한 사항	4357
아키텍처	4358
도구	4358
에픽	4359
관련 리소스	4365
F5에서 AWS의 Application Load Balancer로 마이그레이션할 때 HTTP 헤더를 수정	4366
요약	4366
사전 조건 및 제한 사항	4366
아키텍처	4367
도구	4367
에픽	4368
관련 리소스	4370
다중 VPC에서 AWS 서비스 엔드포인트에 비공개로 액세스	4372
요약	4372
사전 조건 및 제한 사항	4372
아키텍처	4373
도구	4374
에픽	4376
관련 리소스	4380
여러 AWS 계정에서 Network Access Analyzer 조사 결과 보고	4381
요약	4381
사전 조건 및 제한 사항	4382
아키텍처	4383
도구	4385
에픽	4387
문제 해결	4405
관련 리소스	4405

추가 정보	4405
자동으로 Transit Gateway Attachment 태그 지정	4407
요약	4407
사전 조건 및 제한 사항	4407
아키텍처	4408
도구	4409
에픽	4411
관련 리소스	4416
.....	4417
요약	4417
사전 조건 및 제한 사항	4417
아키텍처	4418
도구	4418
에픽	4419
관련 리소스	4422
첨부	4422
Splunk를 사용하여 AWS 네트워크 방화벽 로그 및 지표 보기	4423
요약	4423
사전 조건 및 제한 사항	4423
아키텍처	4424
도구	4424
에픽	4425
관련 리소스	4432
더 많은 패턴	4434
운영 체제	4435
AWS MGN을 사용하여 RHEL BYOL 인스턴스를 AWS LI 인스턴스로 마이그레이션합니다.	4436
요약	4436
사전 조건 및 제한 사항	4436
아키텍처	4437
도구	4437
에픽	4437
관련 리소스	4449
SQL Server를 AWS로 마이그레이션한 후 연결 오류 해결	4450
요약	4450
사전 조건 및 제한 사항	4450
도구	4451

에픽	4451
관련 리소스	4452
패턴 더 보기	4453
운영	4454
Python을 사용하여 자동으로 RFC 생성	4455
요약	4455
사전 조건 및 제한 사항	4455
아키텍처	4456
도구	4456
에픽	4457
관련 리소스	4460
첨부	4460
클라우드 운영을 위한 RACI 지표 생성	4461
요약	4461
에픽	4461
관련 리소스	4464
첨부	4464
암호화된 기본 EBS 볼륨을 사용하여 AWS Cloud9 IDE를 생성	4465
요약	4465
사전 조건 및 제한 사항	4465
아키텍처	4466
도구	4466
에픽	4466
관련 리소스	4468
추가 정보	4468
태그 기반 대시보드를 자동으로 CloudWatch 생성	4470
요약	4470
사전 조건 및 제한 사항	4470
아키텍처	4471
도구	4472
모범 사례	4472
에픽	4473
문제 해결	4477
관련 리소스	4478
추가 정보	4478
AWS Config를 사용하여 생성 날짜를 기준으로 AWS 리소스 검색	4479

요약	4479
사전 조건 및 제한 사항	4480
도구	4480
에픽	4481
추가 정보	4482
계정 또는 조직의 EBS 스냅샷 세부 정보 보기	4484
요약	4484
사전 조건 및 제한 사항	4484
아키텍처	4484
도구	4485
에픽	4485
관련 리소스	4487
추가 정보	4487
패턴 더 보기	4491
SaaS	4492
여러 SaaS 제품의 테넌트를 중앙에서 관리	4493
요약	4493
사전 조건 및 제한 사항	4493
아키텍처	4494
도구	4496
모범 사례	4497
에픽	4497
관련 리소스	4503
더 많은 패턴	4504
보안, 자격 증명, 규정 준수	4505
Amazon Cognito를 사용하여 ASP.NET에서 AWS 서비스에 액세스	4508
요약	4508
사전 조건 및 제한 사항	4508
아키텍처	4509
도구	4509
에픽	4510
문제 해결	4514
관련 리소스	4514
첨부	4514
AWS Directory Service를 사용하여 SQL 서버 인증하기	4515
요약	4515

사전 조건 및 제한 사항	4515
아키텍처	4516
도구	4516
에픽	4516
관련 리소스	4520
인시던트 응답 및 포렌식 자동화	4521
요약	4521
사전 조건 및 제한 사항	4522
아키텍처	4522
도구	4525
에픽	4526
관련 리소스	4529
추가 정보	4529
첨부	4530
Security Hub 표준 조사 결과에 대한 문제 해결 자동화	4531
요약	4531
사전 조건 및 제한 사항	4532
아키텍처	4532
도구	4533
모범 사례	4533
에픽	4534
관련 리소스	4536
첨부	4536
Amazon Inspector를 사용하여 교차 계정 워크로드에 대한 보안 스캔 자동화	4537
요약	4537
사전 조건 및 제한 사항	4537
아키텍처	4538
도구	4539
에픽	4540
관련 리소스	4543
첨부	4543
보안 모범 사례를 CloudTrail 사용하여 AWS를 자동으로 다시 활성화합니다.	4544
요약	4544
사전 조건 및 제한 사항	4544
아키텍처	4545
도구	4545

에픽	4546
관련 리소스	4550
첨부	4551
암호화되지 않은 Amazon RDS DB 인스턴스 및 클러스터를 자동으로 수정하기	4552
요약	4552
사전 조건 및 제한 사항	4552
아키텍처	4554
도구	4554
모범 사례	4555
에픽	4555
관련 리소스	4561
추가 정보	4562
IAM 사용자 액세스 키 자동 교체	4563
요약	4563
사전 조건 및 제한 사항	4564
아키텍처	4565
도구	4566
에픽	4568
관련 리소스	4577
AWS 계정에서 IAM 정책 및 역할을 자동으로 검증하고 배포	4578
요약	4578
사전 조건 및 제한 사항	4579
아키텍처	4579
도구	4580
에픽	4581
관련 리소스	4584
Security Hub와 Jira를 양방향으로 통합하기	4585
요약	4585
사전 조건 및 제한 사항	4586
아키텍처	4586
도구	4588
에픽	4588
관련 리소스	4597
추가 정보	4597
강화된 컨테이너 이미지를 위한 파이프라인을 구축하세요.	4599
요약	4599

사전 조건 및 제한 사항	4599
아키텍처	4600
도구	4603
에픽	4603
문제 해결	4611
관련 리소스	4611
Terraform을 사용하여 AWS Organizations의 IAM 액세스 키 관리를 중앙 집중화하십시오.	4612
요약	4612
사전 조건 및 제한 사항	4613
아키텍처	4613
도구	4615
모범 사례	4615
에픽	4616
문제 해결	4622
관련 리소스	4623
중앙 집중식 로깅 및 다중 계정 보안	4624
요약	4624
사전 조건 및 제한 사항	4625
아키텍처	4625
도구	4627
에픽	4628
관련 리소스	4634
첨부	4635
Amazon CloudFront 배포판에서 액세스 로깅, HTTPS 및 TLS 버전을 확인하십시오.	4636
요약	4636
사전 조건 및 제한 사항	4637
아키텍처	4637
도구	4638
에픽	4638
관련 리소스	4641
첨부	4641
IPv4 및 IPv6용 보안 그룹 수신 규칙에서 단일 호스트 네트워크 항목 확인	4642
요약	4642
사전 조건 및 제한 사항	4642
아키텍처	4643
도구	4643

에픽	4644
관련 리소스	4646
첨부	4647
Amazon Cognito 인증 흐름 선택	4648
요약	4648
사전 조건 및 제한 사항	4648
아키텍처	4649
도구	4653
에픽	4653
관련 리소스	4656
추가 정보	4656
가드를 사용하여 AWS Config 사용자 지정 규칙을 생성합니다.	4658
요약	4658
사전 조건 및 제한 사항	4659
아키텍처	4659
도구	4664
에픽	4664
문제 해결	4666
관련 리소스	4667
여러 AWS 계정의 Prowler 조사 결과에 대한 보고서 생성	4668
요약	4668
사전 조건 및 제한 사항	4669
아키텍처	4670
도구	4670
에픽	4672
문제 해결	4693
관련 리소스	4693
추가 정보	4694
AWS Config로 사용하지 않는 EBS 볼륨 삭제	4696
요약	4696
사전 조건 및 제한 사항	4696
아키텍처	4697
도구	4697
에픽	4698
문제 해결	4700
관련 리소스	4701

CDK를 사용하여 Control Tower 제어 배포	4702
요약	4702
사전 조건 및 제한 사항	4703
아키텍처	4704
도구	4704
모범 사례	4705
에픽	4706
관련 리소스	4712
추가 정보	4712
Terraform을 사용하여 AWS Control Tower 컨트롤 배포하기	4715
요약	4715
사전 조건 및 제한 사항	4716
아키텍처	4716
도구	4717
모범 사례	4718
에픽	4718
문제 해결	4723
관련 리소스	4724
추가 정보	4725
코드의 보안 문제를 탐지하는 파이프라인을 배포하세요.	4727
요약	4727
사전 조건 및 제한 사항	4727
아키텍처	4728
도구	4729
에픽	4729
문제 해결	4732
관련 리소스	4732
추가 정보	4732
Terraform을 사용하여 AWS WAF의 보안 자동화 솔루션 배포	4734
요약	4734
사전 조건 및 제한 사항	4734
아키텍처	4735
도구	4735
모범 사례	4736
에픽	4736
문제 해결	4739

관련 리소스	4739
추가 정보	4740
Step Functions를 사용하여 IAM Access Analyzer로 IAM 정책을 동적으로 생성하기	4741
요약	4741
사전 조건 및 제한 사항	4742
아키텍처	4742
도구	4743
에픽	4744
관련 리소스	4750
AWS Amplify 웹 애플리케이션에 대해 AWS WAF 활성화하기	4751
요약	4751
사전 조건 및 제한 사항	4751
아키텍처	4752
도구	4754
에픽	4755
관련 리소스	4760
템플릿 GuardDuty 사용 CloudFormation 활성화	4761
요약	4761
사전 조건 및 제한 사항	4761
아키텍처	4762
도구	4762
에픽	4763
관련 리소스	4764
추가 정보	4765
Amazon RDS for SQL Server에서 투명한 데이터 암호화 활성화하기	4769
요약	4769
사전 조건 및 제한 사항	4769
아키텍처	4770
도구	4770
에픽	4770
관련 리소스	4773
AWS CloudFormation 스택이 승인된 S3 버킷에서 시작되었는지 확인	4774
요약	4774
사전 조건 및 제한 사항	4774
아키텍처	4775
도구	4775

에픽	4776
관련 리소스	4776
추가 정보	4777
첨부	4777
AWS 로드 밸런서가 보안 리스너 프로토콜을 사용하는지 확인	4778
요약	4778
사전 조건 및 제한 사항	4779
아키텍처	4779
도구	4779
모범 사례	4780
에픽	4780
문제 해결	4783
관련 리소스	4783
첨부	4783
Amazon EMR 저장 데이터에 대한 암호화 확인	4784
요약	4784
사전 조건 및 제한 사항	4785
아키텍처	4785
도구	4786
에픽	4786
관련 리소스	4789
첨부 파일	4789
IAM 프로파일이 EC2 인스턴스와 연결되었는지 확인	4790
요약	4790
사전 조건 및 제한 사항	4790
아키텍처	4791
도구	4791
에픽	4792
관련 리소스	4794
첨부	4794
새 Amazon Redshift 클러스터가 암호화되었는지 확인합니다.	4795
요약	4795
사전 조건 및 제한 사항	4795
아키텍처	4796
도구	4796
에픽	4797

관련 리소스	4799
첨부	4799
IAM Identity Center ID 및 할당 보고서 내보내기	4800
요약	4800
사전 조건 및 제한 사항	4801
아키텍처	4802
도구	4802
에픽	4803
문제 해결	4804
관련 리소스	4805
추가 정보	4806
예정된 KMS 키 삭제를 방지하는 데 도움이 됩니다.	4808
요약	4808
사전 조건 및 제한 사항	4808
아키텍처	4809
도구	4810
에픽	4811
관련 리소스	4814
추가 정보	4814
첨부	4815
ABAC를 사용하여 퍼블릭 서브넷 보호	4816
요약	4816
사전 조건 및 제한 사항	4816
아키텍처	4817
도구	4818
에픽	4818
관련 리소스	4824
추가 정보	4825
AWS Organizations의 퍼블릭 S3 버킷 식별	4827
요약	4827
사전 조건 및 제한 사항	4827
아키텍처	4828
도구	4829
에픽	4829
문제 해결	4833
관련 리소스	4833

추가 정보	4834
Okta와 IAM Identity Center 통합	4835
요약	4835
사전 조건 및 제한 사항	4835
아키텍처	4835
도구	4836
에픽	4836
관련 리소스	4847
다음을 사용하여 IAM ID 센터 권한 세트를 관리합니다. CodePipeline	4848
요약	4848
사전 조건 및 제한 사항	4849
아키텍처	4849
도구	4851
모범 사례	4852
에픽	4852
문제 해결	4861
관련 리소스	4862
AWS Secrets Manager로 보안 인증 정보 관리	4863
요약	4863
사전 조건 및 제한 사항	4863
아키텍처	4864
도구	4864
에픽	4864
관련 리소스	4866
추가 정보	4866
시작 시 전송 중 암호화가 있는지 Amazon EMR 클러스터 모니터링	4869
요약	4869
사전 조건 및 제한 사항	4870
아키텍처	4870
도구	4871
에픽	4872
관련 리소스	4874
첨부	4874
Amazon ElastiCache 클러스터의 미사용 암호화 모니터링	4875
요약	4875
사전 조건 및 제한 사항	4876

아키텍처	4877
도구	4877
에픽	4878
관련 리소스	4880
첨부	4880
EC2 인스턴스 키 페어를 모니터링	4881
요약	4881
사전 조건 및 제한 사항	4881
아키텍처	4882
도구	4882
에픽	4883
관련 리소스	4886
첨부	4886
.....	4887
요약	4887
사전 조건 및 제한 사항	4887
아키텍처	4888
도구	4888
에픽	4889
관련 리소스	4891
첨부	4891
IAM 루트 사용자 활동 모니터링	4892
요약	4892
사전 조건 및 제한 사항	4892
아키텍처	4893
도구	4893
에픽	4895
관련 리소스	4899
추가 정보	4899
IAM 사용자 생성 시 알림	4900
요약	4900
사전 조건 및 제한 사항	4900
아키텍처	4901
도구	4901
에픽	4902
관련 리소스	4904

첨부	4904
Git 리포지토리에서 민감한 정보 검사하기	4905
요약	4905
사전 조건 및 제한 사항	4905
아키텍처	4905
도구	4906
모범 사례	4906
에픽	4906
관련 리소스	4910
AWS Network Firewall에서 Slack 채널로 알림 전송	4911
요약	4911
사전 조건 및 제한 사항	4911
아키텍처	4912
도구	4913
에픽	4914
관련 리소스	4918
추가 정보	4918
AWS 프라이빗 CA와 AWS RAM을 사용하여 프라이빗 인증서 관리를 간소화합니다.	4923
요약	4923
사전 조건 및 제한 사항	4924
아키텍처	4924
도구	4925
에픽	4926
관련 리소스	4932
추가 정보	4932
다중 계정 환경에서 모든 Security Hub 멤버 계정의 보안 표준 제어를 끕니다.	4933
요약	4933
사전 조건 및 제한 사항	4933
아키텍처	4934
도구	4935
에픽	4936
관련 리소스	4938
를 사용하여 IAM 자격 증명 센터의 AWS CLI 자격 증명을 업데이트합니다. PowerShell	4939
요약	4939
사전 조건 및 제한 사항	4939
아키텍처	4940

도구	4940
모범 사례	4941
에픽	4941
문제 해결	4943
관련 리소스	4943
추가 정보	4944
AWS Config를 사용하여 Amazon Redshift 모니터링	4946
요약	4946
사전 조건 및 제한 사항	4946
아키텍처	4947
도구	4947
에픽	4948
관련 리소스	4951
추가 정보	4952
Network Firewall을 사용하여 아웃바운드 네트워크 트래픽에서 DNS 도메인 이름 캡처	4953
요약	4953
사전 조건 및 제한 사항	4953
아키텍처	4954
도구	4955
에픽	4955
Terraform을 사용하여 자동으로 활성화하십시오. GuardDuty	4969
요약	4969
사전 조건 및 제한 사항	4970
아키텍처	4971
도구	4972
에픽	4973
관련 리소스	4982
추가 정보	4982
.....	4984
요약	4984
사전 조건 및 제한 사항	4984
아키텍처	4985
도구	4985
에픽	4986
관련 리소스	4988
첨부	4988

.....	4989
요약	4989
사전 조건 및 제한 사항	4989
아키텍처	4990
도구	4990
에픽	4991
관련 리소스	4993
첨부	4993
패턴 더 보기	4994
Serverless	4996
AWS Amplify를 사용하여 리액트 네이티브 앱을 구축하십시오.	4997
요약	4997
사전 조건 및 제한 사항	4997
아키텍처	4998
도구	4998
에픽	4999
관련 리소스	5014
Kinesis Data Streams 및 Amazon Data Firehose를 사용하여 Amazon S3에 DynamoDB 레코드 를 전송합니다.	5015
요약	5015
사전 조건 및 제한 사항	5016
아키텍처	5016
도구	5017
에픽	5017
관련 리소스	5020
API Gateway와 아마존 SQS 통합	5022
요약	5022
사전 조건 및 제한 사항	5022
아키텍처	5022
도구	5023
에픽	5023
관련 리소스	5035
Step Functions에서 Systems Manager 자동화 작업을 동기적으로 실행	5036
요약	5036
사전 조건 및 제한 사항	5037
아키텍처	5037

도구	5038
에픽	5038
관련 리소스	5042
추가 정보	5042
AWS Lambda를 사용하여 S3 객체의 병렬 읽기 실행	5047
요약	5047
사전 조건 및 제한 사항	5048
아키텍처	5048
도구	5049
모범 사례	5050
에픽	5050
문제 해결	5056
관련 리소스	5057
추가 정보	5057
Amazon S3 버킷에 대한 프라이빗 액세스를 설정합니다.	5059
요약	5059
사전 조건 및 제한 사항	5059
아키텍처	5060
도구	5061
모범 사례	5062
에픽	5062
문제 해결	5064
관련 리소스	5064
서버리스 접근 방식을 사용하여 AWS 서비스를 함께 연결	5066
요약	5066
사전 조건 및 제한 사항	5066
아키텍처	5067
도구	5067
에픽	5068
패턴 더 보기	5071
소프트웨어 개발 및 테스트	5073
DynamoDB를 위한 PynamoDB 모델 및 CRUD 함수를 자동으로 생성합니다.	5074
요약	5074
사전 조건 및 제한 사항	5074
아키텍처	5075
도구	5076

에픽	5077
관련 리소스	5080
추가 정보	5080
Green Boost를 사용한 웹 앱 개발 살펴보기	5081
요약	5081
사전 조건 및 제한 사항	5081
아키텍처	5082
도구	5083
모범 사례	5084
에픽	5085
문제 해결	5104
관련 리소스	5105
AWS를 사용하여 단위 테스트 실행 CodeBuild	5106
요약	5106
사전 조건 및 제한 사항	5106
아키텍처	5107
도구	5107
에픽	5107
관련 리소스	5111
추가 정보	5111
육각형 아키텍처로 Python 프로젝트 구조화	5115
요약	5115
사전 조건 및 제한 사항	5115
아키텍처	5116
도구	5117
모범 사례	5118
에픽	5119
관련 리소스	5139
패턴 더 보기	5141
스토리지 및 백업	5142
EC2 인스턴스가 AMS의 S3 버킷에 대한 쓰기 액세스 권한을 쓸 수 있도록 허용	5143
요약	5143
사전 조건 및 제한 사항	5143
아키텍처	5144
도구	5144
에픽	5144

관련 리소스	5147
Snowflake 데이터베이스로 데이터 스트림 수집 자동화	5148
요약	5148
사전 조건 및 제한 사항	5148
아키텍처	5148
도구	5149
에픽	5149
관련 리소스	5155
추가 정보	5155
EBS 볼륨을 자동으로 암호화하기	5159
요약	5159
사전 조건 및 제한 사항	5159
아키텍처	5160
도구	5160
에픽	5162
관련 리소스	5168
AWS의 Charon-SSP 에뮬레이터에서 Sun SPARC 서버 백업하기	5169
요약	5169
사전 조건 및 제한 사항	5170
도구	5173
에픽	5175
관련 리소스	5183
추가 정보	5184
첨부	5187
Veeam을 사용하여 Amazon S3에 데이터를 백업 및 보관	5188
요약	5188
사전 조건 및 제한 사항	5189
아키텍처	5190
도구	5191
모범 사례	5192
에픽	5193
관련 리소스	5205
추가 정보	5206
AWS 기반 VMware NetBackup 클라우드를 구성하도록 구성	5210
요약	5210
사전 조건 및 제한 사항	5211

아키텍처	5212
도구	5212
에픽	5213
관련 리소스	5216
Amazon PrivateLink S3용 DistCp AWS를 사용하여 하둡 데이터를 Amazon S3로 마이그레이션 합니다.	5217
요약	5217
사전 조건 및 제한 사항	5217
아키텍처	5218
도구	5218
에픽	5219
CloudEndure 온프레미스 재해 복구에 사용	5230
요약	5230
사전 조건 및 제한 사항	5231
아키텍처	5231
도구	5231
에픽	5232
관련 리소스	5243
패턴 더 보기	5244
웹 및 모바일 앱	5245
Amplify 웹 애플리케이션을 지속적으로 배포	5246
요약	5246
사전 조건 및 제한 사항	5247
아키텍처	5247
도구	5248
에픽	5248
관련 리소스	5252
Amplify를 사용한 React 앱 생성과 Amazon Cognito를 사용한 인증 추가	5253
요약	5253
사전 조건 및 제한 사항	5253
아키텍처	5253
도구	5254
에픽	5254
관련 리소스	5267
Amazon S3에 React 기반 SPA를 배포하고 CloudFront	5268
요약	5268

사전 조건 및 제한 사항	5268
아키텍처	5269
도구	5269
에픽	5270
추가 정보	5274
프라이빗 엔드포인트와 Application Load Balancer를 사용하여 내부 웹 사이트에 Amazon API Gateway API 배포	5275
요약	5275
사전 조건 및 제한 사항	5275
아키텍처	5276
도구	5277
에픽	5277
관련 리소스	5281
로컬 Angular QuickSight 애플리케이션에 Amazon 대시보드 내장	5282
요약	5282
사전 조건 및 제한 사항	5282
아키텍처	5283
도구	5283
에픽	5284
관련 리소스	5298
추가 정보	5299
패턴 더 보기	5300
.....	5302

AWS 규범적 지침 패턴

Amazon Web Services (AWS) 규범적 step-by-step 지침 패턴은 특정 클라우드 마이그레이션, 현대화 및 배포 시나리오를 구현하기 위한 지침, 아키텍처, 도구 및 코드를 제공합니다. 에서 AWS 주제 전문가가 검토하는 이러한 패턴은 마이그레이션을 계획 중이거나 진행 중인 빌더 및 실무 사용자를 대상으로 합니다. AWS 또한 이미 클라우드 서비스를 사용하고 있으며 클라우드 운영을 최적화하거나 AWS 현대화할 방법을 찾고 있는 사용자도 지원합니다.

프로젝트의 개념 증명, 계획 또는 구현 단계에 있는 관계없이 이러한 패턴을 사용하여 복잡성이 다양한 온프레미스 또는 클라우드 워크로드를 이전하고 클라우드 채택, 최적화 및 현대화 노력을 가속화할 수 있습니다. AWS 예를 들어 클라우드 마이그레이션 프로젝트의 경우:

- 계획 단계에서 AWS에 마이그레이션할 수 있는 다양한 옵션을 평가할 수 있습니다. 재배치, 리호스팅, 플랫폼포밍 또는 리아키텍트 여부에 따라 요구 사항에 맞는 올바른 패턴을 선택할 수 있습니다. 또한 마이그레이션에 사용할 수 있는 다양한 도구를 이해하고 라이선스 조달 계획을 시작하거나 공급업체와 초기 대화를 시작할 수 있습니다.
- 개념 증명 및 구현 단계에서는 패턴에 제공된 step-by-step 지침에 따라 워크로드를 마이그레이션할 수 있습니다. AWS 각 패턴에는 사전 요구 사항, 대상 참조 아키텍처, 도구, step-by-step 작업, 모범 사례, 문제 해결 및 코드와 같은 세부 정보가 포함됩니다.
- 이미 를 사용하고 있다면 클라우드 리소스 사용을 현대화 AWS 클라우드, 최적화, 확장 및 보호하는데 도움이 되는 패턴을 찾을 수 있습니다.

기술 도메인별 패턴 목록을 보려면 다음 링크 또는 [AWS 권장 가이드 홈페이지](#)의 필터링 및 검색 옵션을 사용하세요.

- [분석](#)
- [비즈니스 생산성](#)
- [클라우드 네이티브](#)
- [컨테이너 및 마이크로서비스](#)
- [콘텐츠 전송](#)
- [비용 관리](#)
- [데이터 레이크](#)
- [데이터베이스](#)
- [DevOps](#)

- [최종 사용자 컴퓨팅](#)
- [고성능 컴퓨팅](#)
- [하이브리드 클라우드](#)
- [인프라](#)
- [IoT](#)
- [기계 학습 및 AI](#)
- [메인프레임](#)
- [관리 및 거버넌스](#)
- [메시징 및 커뮤니케이션](#)
- [마이그레이션](#)
- [현대화](#)
- [네트워킹](#)
- [운영 체제](#)
- [운영](#)
- [SaaS](#)
- [보안, 자격 증명 및 규정 준수](#)
- [서버리스](#)
- [소프트웨어 개발 및 테스트](#)
- [스토리지 및 백업](#)
- [웹 및 모바일 앱](#)

가이드, 전략, 패턴을 포함한 모든 간행물을 보려면 [AWS 권장 가이드 홈페이지](#)를 참조하세요.

분석

주제

- [Microsoft SQL Server Analysis Services에서 Amazon Redshift 데이터를 분석](#)
- [Amazon Athena와 Amazon을 사용하여 중첩된 JSON 데이터를 분석하고 시각화합니다. QuickSight](#)
- [AWS CloudFormation 템플릿을 사용하여 AWS Glue에서 암호화 적용을 자동화합니다.](#)
- [AWS Glue를 사용하여 Amazon S3에서 Amazon Redshift로 데이터를 점차 늘려 로딩하기 위한 ETL 서비스 파이프라인 빌드](#)
- [AWS 서비스를 사용하여 위험 가치\(VaR\) 계산](#)
- [Teradata NORMALIZE 임시 기능을 Amazon Redshift SQL로 변환](#)
- [Teradata RESET WHEN 기능을 Amazon Redshift SQL로 변환](#)
- [시작 시 Amazon EMR 클러스터에 태그 지정 적용](#)
- [시작 시 Amazon S3에 Amazon EMR 로깅이 활성화되었는지 확인](#)
- [AWS Glue 작업과 Python을 사용하여 테스트 데이터 생성](#)
- [Lambda 함수를 사용하여 임시 EMR 클러스터에서 Spark 작업 시작](#)
- [AWS Glue를 사용하여 아파치 카산드라 워크로드를 Amazon Keyspaces로 마이그레이션하십시오.](#)
- [온프레미스 서버에서 Oracle Business Intelligence 12c를 AWS 클라우드로 마이그레이션.](#)
- [를 사용하여 온프레미스 Apache Kafka 클러스터를 Amazon MSK로 마이그레이션합니다. MirrorMaker](#)
- [ELK 스택을 AWS 기반 Elastic 클라우드로 마이그레이션](#)
- [Starburst를 사용하여 데이터를 AWS 클라우드로 마이그레이션하십시오.](#)
- [AWS에서 입력 파일 크기의 ETL 수집 최적화](#)
- [AWS Step Functions를 사용하여 검증, 변환 및 파티셔닝을 통해 ETL 파이프라인 오케스트레이션](#)
- [Amazon Redshift 기계 학습을 이용하여 고급 분석 수행](#)
- [Athena를 사용한 Amazon DynamoDB 테이블 액세스, 쿼리 및 조인](#)
- [스칼라 Python UDF를 사용하여 Amazon Redshift 쿼리 결과에 대한 언어별 정렬 설정](#)
- [Lambda 함수가 서로 다른 AWS 리전의 S3 버킷에서 이벤트 알림을 수신하도록 Lambda 함수 구독](#)
- [데이터를 Apache Parquet으로 변환하기 위한 세 가지 AWS Glue ETL 작업 유형](#)
- [Amazon Athena와 Amazon을 사용하여 Amazon Redshift 감사 로그를 시각화합니다. QuickSight](#)
- [Amazon을 사용하여 모든 AWS 계정에 대한 IAM 자격 증명 보고서를 시각화합니다. QuickSight](#)

- [패턴 더 보기](#)

Microsoft SQL Server Analysis Services에서 Amazon Redshift 데이터를 분석

작성자: Sunil Vora(AWS)

환경: PoC 또는 파일럿	소스: Amazon Redshift	대상: Microsoft SQL Server Analysis Services
R 유형: 해당 없음	워크로드: Microsoft	기술: 분석
AWS 서비스: Amazon Redshift		

요약

이 패턴은 데이터베이스 액세스를 위해 Intellisoft OLE DB Provider 또는 CData ADO.NET Provider를 사용하여 Microsoft SQL Server Analysis Services에 Amazon Redshift 데이터를 연결하고 분석하는 방법을 설명합니다.

Amazon Redshift는 클라우드에서 완전히 관리되는 페타바이트급 데이터 웨어하우스 서비스입니다. SQL Server Analysis Services는 Amazon Redshift와 같은 데이터 마트와 데이터 웨어하우스의 데이터를 분석하는 데 사용할 수 있는 온라인 분석 처리(OLAP) 도구입니다. SQL Server Analysis Services를 사용하면 데이터에서 OLAP 큐브를 생성하여 신속한 고급 데이터 분석을 수행할 수 있습니다.

사전 조건 및 제한 사항

가정

- 이 패턴은 Amazon Elastic Compute Cloud(Amazon EC2) 인스턴스에서 SQL Server Analysis Services와 Intellisoft OLE DB Provider 또는 CData ADO.NET Provider for Amazon Redshift를 설정하는 방법을 설명합니다. 또는 기업 데이터 센터의 호스트에 둘 다 설치할 수도 있습니다.

사전 조건

- 활성 상태의 AWS 계정.
- 보안 인증 정보가 있는 Amazon Redshift 클러스터

아키텍처

소스 기술 스택

- Amazon Redshift 클러스터

대상 기술 스택

- 대상: Microsoft SQL Server Analysis Services

소스 및 타겟 아키텍처

도구

- [Microsoft Visual Studio 2019\(Community Edition\)](#)
- [Amazon Redshift용 Intellisense OLE DB Provider\(평가판\)](#) 또는 [Amazon Redshift용 CData ADO.NET Provider\(평가판\)](#)

에픽

테이블 분석

작업	설명	필요한 기술
가져올 테이블과 데이터를 분석하세요.	가져올 Amazon Redshift 테이블과 그 크기를 파악하세요.	DBA

EC2 인스턴스 설정 및 도구 설치

작업	설명	필요한 기술
EC2 인스턴스를 설정합니다.	AWS 계정에서 프라이빗 또는 퍼블릭 서브넷에 EC2 인스턴스를 생성합니다.	시스템 관리자

작업	설명	필요한 기술
데이터베이스 액세스를 위한 도구를 설치합니다.	Amazon Redshift용 Intellisoft OLE DB Provider (또는 Amazon Redshift용 CData ADO.NET Provider)를 다운로드하고 설치합니다.	시스템 관리자
Visual Studio를 설치합니다.	Visual Studio 2019(Community Edition) 를 다운로드하여 설치합니다.	시스템 관리자
익스텐션을 설치합니다.	Visual Studio에 Microsoft Analysis Services Projects 확장 프로그램을 설치합니다.	시스템 관리자
프로젝트를 생성합니다.	Visual Studio에서 새 테이블 형식 모델 프로젝트를 생성하여 Amazon Redshift 데이터를 저장합니다. Visual Studio에서는 프로젝트를 만들 때 Analysis Services Tabular Project 옵션을 선택합니다.	DBA

데이터 소스 생성 및 테이블 가져오기

작업	설명	필요한 기술
Amazon Redshift 데이터 소스를 생성합니다.	Intellisoft OLE DB Provider for Amazon Redshift(또는 CData ADO.NET Provider for Amazon Redshift) 및 Amazon Redshift 보안 인증 정보를 사용하여 Amazon Redshift 데이터 소스를 생성합니다.	Amazon Redshift, DBA

작업	설명	필요한 기술
테이블을 가져옵니다.	Amazon Redshift에서 테이블과 뷰를 선택하여 SQL Server Analysis Services Project로 가져옵니다.	Amazon Redshift, DBA

마이그레이션 후 정리

작업	설명	필요한 기술
EC2 인스턴스를 삭제합니다.	이전에 시작한 EC2 인스턴스를 삭제합니다.	시스템 관리자

관련 리소스

- [Amazon Redshift](#)(AWS 설명서)
- [SQL Server Analysis Services 설치](#)(Microsoft 설명서)
- [테이블 형식 모델 디자이너](#)(Microsoft 설명서)
- [고급 분석을 위한 OLAP 큐브 개요](#)(Microsoft 설명서)
- [Microsoft Visual Studio 2019\(Community Edition\)](#)
- [Amazon Redshift용 Intellisense OLE DB Provider\(평가판\)](#)
- [Amazon Redshift용 CData ADO.NET Provider\(평가판\)](#)

Amazon Athena와 Amazon을 사용하여 중첩된 JSON 데이터를 분석하고 시각화합니다. QuickSight

작성자: 아늑싱 (AWS)

환경: PoC 또는 파일럿

기술: 분석, 데이터베이스

AWS 서비스: 아마존 아테나, 아마존 QuickSight

요약

이 패턴은 Amazon Athena를 사용하여 JSON 형식의 중첩된 데이터 구조를 표 형식 보기로 변환한 다음 Amazon에서 데이터를 시각화하는 방법을 설명합니다. QuickSight

운영 체제의 API 기반 데이터 피드에 JSON 형식의 데이터를 사용하여 데이터 제품을 생성할 수 있습니다. 또한 이 데이터는 고객과 제품과의 상호 작용을 더 잘 이해하는 데 도움이 되므로 사용자 경험을 맞춤화하고 결과를 예측할 수 있습니다.

사전 조건 및 제한 사항

사전 조건

- 액티브 AWS 계정
- 중첩된 데이터 구조를 나타내는 JSON 파일 (이 패턴은 샘플 파일을 제공함)

제한:

- JSON 기능은 Athena의 기존 SQL 지향 함수와 잘 통합됩니다. 하지만 ANSI SQL과 호환되지 않으며 JSON 파일은 각 레코드를 별도의 줄에 전달해야 합니다. Athena의 `ignore.malformed.json` 속성을 사용하여 잘못된 형식의 JSON 레코드를 null 문자로 변환해야 하는지 아니면 오류를 생성해야 하는지 지정해야 할 수 있습니다. 자세한 내용은 Athena [설명서의 JSON 데이터 읽기 모범 사례를](#) 참조하십시오.
- 이 패턴은 단순하고 적은 양의 JSON 형식 데이터만 고려합니다. 이러한 개념을 대규모로 사용하려면 데이터 파티셔닝을 적용하고 데이터를 더 큰 파일로 통합하는 것을 고려해 보세요.

아키텍처

다음 다이어그램은 이 패턴의 아키텍처와 워크플로를 보여줍니다. 중첩된 데이터 구조는 아마존 심플 스토리지 서비스 (Amazon S3) 에 JSON 형식으로 저장됩니다. Athena에서는 JSON 데이터가 Athena 데이터 구조에 매핑됩니다. 그런 다음 뷰를 만들어 데이터를 분석하고 에서 데이터 구조를 시각화합니다. QuickSight

도구

AWS 서비스

- [Amazon Simple Storage Service\(S3\)](#)는 원하는 양의 데이터를 저장, 보호 및 검색하는 데 도움이 되는 클라우드 기반 객체 스토리지 서비스입니다. 이 패턴은 Amazon S3를 사용하여 JSON 파일을 저장합니다.
- [Amazon Athena](#)는 표준 SQL을 사용하여 Amazon S3에 있는 데이터를 직접 분석할 수 있는 대화형 쿼리 서비스입니다. 이 패턴은 Athena를 사용하여 JSON 데이터를 쿼리하고 변환합니다. 에서 몇 가지 작업을 수행하면 Athena에게 Amazon S3의 데이터를 알려주고 표준 SQL을 사용하여 일회성 쿼리를 실행할 수 있습니다. AWS Management Console Athena는 서버리스이므로 인프라를 설정하거나 관리할 필요가 없으며 실행한 쿼리에 대한 비용만 지불하면 됩니다. Athena는 자동으로 크기를 조정하고 쿼리를 병렬로 실행하므로 대규모 데이터 세트와 복잡한 쿼리에서도 결과를 빠르게 얻을 수 있습니다.
- [QuickSightAmazon](#)은 단일 대시보드에서 데이터를 시각화, 분석 및 보고하는 데 도움이 되는 클라우드 규모의 비즈니스 인텔리전스 (BI) 서비스입니다. QuickSight 기계 학습 (ML) 통찰력을 포함하는 대화형 대시보드를 쉽게 만들고 게시할 수 있습니다. 모든 디바이스에서 이러한 대시보드에 액세스하여 애플리케이션, 포털 및 웹 사이트에 임베드할 수 있습니다.

예제 코드

다음 JSON 파일은 이 패턴에서 사용할 수 있는 중첩된 데이터 구조를 제공합니다.

```
{
  "symbol": "AAPL",
  "financials": [
    {
      "reportDate": "2017-03-31",
      "grossProfit": 20591000000,

```

```

    "costOfRevenue": 32305000000,
    "operatingRevenue": 52896000000,
    "totalRevenue": 52896000000,
    "operatingIncome": 14097000000,
    "netIncome": 11029000000,
    "researchAndDevelopment": 2776000000,
    "operatingExpense": 6494000000,
    "currentAssets": 101990000000,
    "totalAssets": 334532000000,
    "totalLiabilities": 200450000000,
    "currentCash": 15157000000,
    "currentDebt": 13991000000,
    "totalCash": 67101000000,
    "totalDebt": 98522000000,
    "shareholderEquity": 134082000000,
    "cashChange": -1214000000,
    "cashFlow": 12523000000,
    "operatingGainsLosses": null
  }
]
}

```

에픽

S3 버킷 설정

작업	설명	필요한 기술
S3 버킷을 생성합니다.	JSON 파일을 저장할 버킷을 생성하려면 에 로그인하고 Amazon S3 콘솔 을 연 다음 버킷 생성을 선택합니다. AWS Management Console 자세한 내용은 Amazon S3 설명서의 버킷 생성 을 참조하십시오.	시스템 관리자
중첩된 JSON 데이터를 추가합니다.	JSON 파일을 S3 버킷에 업로드합니다. 샘플 JSON 파일은 이전 섹션을 참조하십시오. 이에 관한 지침은 Amazon S3 설	시스템 관리자

작업	설명	필요한 기술
	명서의 객체 업로드 를 참조하세요.	

Athena의 데이터 분석

작업	설명	필요한 기술
JSON 데이터를 매핑하기 위한 테이블을 생성합니다.	<ol style="list-style-type: none"> Athena 콘솔을 엽니다. Athena 설명서의 지침에 따라 데이터베이스를 생성합니다. 데이터베이스 메뉴에서 생성한 데이터베이스를 선택합니다. 쿼리 편집기에 다음과 같은 CREATE TABLE 명령문을 입력합니다. <pre>CREATE EXTERNAL TABLE financials_json (symbol string, financials array< struct<re portdate: string, grossprof it: bigint, totalreve nue: bigint, totalcash : bigint, totaldebt : bigint, researcha nddevelopment: bigint>>)</pre>	개발자

작업	설명	필요한 기술
	<pre data-bbox="630 212 1029 428">ROW FORMAT SERDE 'org.openx.data.js onserde.JsonSerDe' LOCATION 's3://s3b ucket-for-athena/'</pre> <p data-bbox="630 464 964 594">여기서 JSON 파일이 포 함된 S3 버킷의 위치를 LOCATION 지정합니다.</p> <p data-bbox="591 617 1019 699">5. [Run] 을 선택하여 테이블을 생성합니다.</p> <p data-bbox="591 774 1013 905">테이블 생성에 대한 자세한 내 용은 Athena 설명서를 참조하 십시오.</p>	

작업	설명	필요한 기술
데이터 분석을 위한 뷰를 만드세요.	<ol style="list-style-type: none"> 1. Athena 콘솔을 엽니다. 2. Athena 설명서의 지침에 따라 데이터베이스를 생성합니다. 3. 데이터베이스 메뉴에서 생성한 데이터베이스를 선택합니다. 4. 쿼리 편집기에 다음과 같은 CREATE VIEW 명령문을 입력합니다. <div data-bbox="630 751 1027 1671" style="border: 1px solid #ccc; border-radius: 10px; padding: 10px; margin: 10px 0;"> <pre>CREATE OR REPLACE VIEW financial_json_view AS SELECT symbol, financials[1].reportdate one_report_date, -- indexes start with 1 financials[1].totalrevenue one_total_revenue, financials[1].reportdate another_report_date, financials[1].totalrevenue another_total_revenue FROM financials_json where symbol='AAPL' ORDER BY 1</pre> </div> 5. 실행(Run)을 선택하여 뷰를 생성합니다. 	개발자

작업	설명	필요한 기술
	뷰 만들기에 대한 자세한 내용은 Athena 설명서를 참조하십시오.	
데이터를 분석하고 검증하십시오.	<ol style="list-style-type: none"> Athena 콘솔을 엽니다. 쿼리 편집기에서 이전 단계에서 만든 뷰를 사용하여 쿼리를 실행합니다. JSON 파일을 기준으로 데이터를 검증하여 열 이름과 데이터 유형이 올바르게 매핑되었는지 확인합니다. 	개발자

에서 데이터를 시각화하십시오. QuickSight

작업	설명	필요한 기술
에서 Athena를 데이터 원본으로 설정합니다. QuickSight	<ol style="list-style-type: none"> QuickSight 콘솔을 엽니다. 데이터 세트, 새 데이터 세트를 선택합니다. Athena를 데이터 원본으로 선택합니다. 만든 뷰가 포함된 데이터베이스를 선택합니다. 데이터세트를 만들려는 뷰를 선택합니다. 데이터 세트 생성 완료 페이지에서 데이터 직접 쿼리를 선택합니다. [Visualize]를 선택합니다. 	시스템 관리자
에서 QuickSight 데이터를 시각화하세요.	<ol style="list-style-type: none"> 데이터세트를 시각화한 후 왼쪽 창에서 시각적 개체를 	데이터 분석가

작업	설명	필요한 기술
	<p>선택하고 데이터세트의 필드를 선택합니다. 자세한 내용은 설명서의 튜토리얼을 참조하세요. QuickSight</p> <ol style="list-style-type: none"> 2. 분석 변경 내용을 저장합니다. 3. 대시보드 게시를 선택하여 만든 시각적 개체를 게시합니다. 	

관련 리소스

- [Amazon Athena 설명서](#)
- [아마존 QuickSight 튜토리얼](#)
- [중첩된 JSON 사용하기 \(블로그 게시물\)](#)

AWS CloudFormation 템플릿을 사용하여 AWS Glue에서 암호화 적용을 자동화합니다.

작성자: 디오고 게데스(AWS)

코드 리포지토리: AWS Glue 암호화 적용	환경: 프로덕션	기술: 분석, 보안, ID, 규정 준수
워크로드: 기타 모든 워크로드	AWS 서비스: 아마존 EventBridge, AWS Glue, AWS KMS, AWS Lambda, AWS CloudFormation	

요약

이 패턴은 AWS CloudFormation 템플릿을 사용하여 AWS Glue에서 암호화 적용을 설정하고 자동화하는 방법을 보여줍니다. 템플릿은 암호화를 적용하는 데 필요한 모든 구성과 리소스를 생성합니다. 이러한 리소스에는 초기 구성, Amazon EventBridge 규칙에 의해 생성된 예방 제어, AWS Lambda 함수가 포함됩니다.

사전 조건 및 제한 사항

사전 조건

- 활성 상태의 AWS 계정.
- CloudFormation 템플릿 및 해당 리소스를 배포할 수 있는 권한

제한 사항

이 보안 제어는 리전별로 적용됩니다. AWS Glue에서 암호화 적용을 설정하려는 각 AWS 리전에 보안 제어를 배포해야 합니다.

아키텍처

대상 기술 스택

- 아마존 CloudWatch 로그 (AWS Lambda에서 제공)

- 아마존 EventBridge 룰
- AWS CloudFormation 스택
- AWS CloudTrail
- AWS Identity and Access Management(IAM) 관리형 역할 및 정책
- AWS Key Management Service (AWS KMS)
- AWS KMS 별칭
- AWS Lambda 함수
- AWS Systems Manager Parameter Store

대상 아키텍처

다음 다이어그램은 AWS Glue에서 암호화 적용을 자동화하는 방법을 보여줍니다.

이 다이어그램은 다음 워크플로를 보여줍니다.

1. [CloudFormation 템플릿](#)은 AWS Glue의 암호화 적용을 위한 초기 구성 및 탐지 제어를 비롯한 모든 리소스를 생성합니다.
2. EventBridge 규칙은 암호화 구성의 상태 변경을 감지합니다.
3. Lambda 함수는 Logs를 통한 평가 및 로깅을 위해 호출됩니다. CloudWatch 규정 미준수 탐지의 경우, Parameter Store는 AWS KMS 키의 Amazon 리소스 이름(ARN)으로 복구됩니다. 서비스는 암호화가 활성화된 상태에서 규정 준수 상태로 개선됩니다.

자동화 및 규모 조정

[AWS Organizations](#)를 사용하는 경우, [AWS CloudFormation StackSets Glue에서](#) 암호화 적용을 활성화하려는 여러 계정에 AWS를 사용하여 이 템플릿을 배포할 수 있습니다.

도구

- [Amazon](#)은 AWS 리소스의 지표와 AWS에서 실행하는 애플리케이션을 실시간으로 모니터링할 수 있도록 CloudWatch 도와줍니다.
- [EventBridgeAmazon](#)은 다양한 소스의 실시간 데이터와 애플리케이션을 연결하는 데 도움이 되는 서버리스 이벤트 버스 서비스입니다. Lambda 함수, API 대상을 사용하는 HTTP 간접 호출 엔드포인트 또는 다른 AWS 계정의 이벤트 버스를 예로 들 수 있습니다.

- [AWS는 AWS CloudFormation](#) 리소스를 설정하고, 빠르고 일관되게 프로비저닝하고, AWS 계정 및 지역 전반의 수명 주기 전반에 걸쳐 리소스를 관리할 수 있도록 지원합니다.
- [AWS는 AWS](#) 계정의 운영 및 위험 감사, 거버넌스 및 규정 준수를 지원하도록 CloudTrail 지원합니다.
- [AWS Glue](#)는 완전 관리형 추출, 전환, 적재(ETL) 서비스입니다. 이를 통해 데이터 스토어와 데이터 스트림 간에 데이터를 안정적으로 분류, 정리, 보강하고 이동할 수 있습니다.
- [AWS Key Management Service\(AWS KMS\)](#)를 사용하면 암호화 키를 생성하고 제어하여 데이터를 보호할 수 있습니다.
- [Lambda](#)는 서버를 프로비저닝하거나 관리할 필요 없이 코드를 실행할 수 있도록 도와주는 컴퓨팅 서비스입니다. 필요할 때만 코드를 실행하며 자동으로 확장이 가능하므로 사용한 컴퓨팅 시간만큼만 비용을 지불합니다.
- [AWS Systems Manager](#)는 AWS 클라우드에서 실행되는 애플리케이션과 인프라를 관리하는 데 도움이 됩니다. 애플리케이션 및 리소스 관리를 간소화하고, 운영 문제의 감지 및 해결 시간을 단축하며, AWS 리소스를 규모에 따라 안전하게 관리하는 데 도움이 됩니다.

코드

이 패턴의 코드는 GitHub [aws-custom-guardrail-event-driven](#) 리포지토리에서 사용할 수 있습니다.

모범 사례

AWS Glue는 [AWS Glue에 작업 작성 및 개발 엔드포인트를 사용하여 스크립트 개발](#)에 대한 저장된 데이터 암호화를 지원합니다.

다음 모범 사례를 고려하세요.

- AWS KMS 키를 사용하여 암호화된 저장 데이터를 기록하도록 ETL 작업 및 개발 엔드포인트를 구성합니다.
- AWS KMS를 통해 관리하는 키를 사용하여 [AWS Glue 데이터 카탈로그](#)에 저장된 메타데이터를 암호화합니다.
- AWS KMS 키를 사용하여 [크롤러](#) 및 ETL 작업에서 생성된 작업 북마크와 로그를 암호화합니다.

에픽

템플릿을 실행합니다. CloudFormation

작업	설명	필요한 기술
CloudFormation 템플릿을 배포합니다.	<p>GitHub 리포지토리에서 aws-custom-guardrail-event-driven.yaml 템플릿을 다운로드한 다음 템플릿을 배포합니다. CREATE_COMPLETE 상태는 템플릿이 성공적으로 배포되었음을 나타냅니다.</p> <p>참고: 템플릿에는 입력 파라미터가 필요하지 않습니다.</p>	클라우드 아키텍트

AWS Glue의 암호화 설정 확인

작업	설명	필요한 기술
AWS KMS 키 구성을 확인합니다.	<ol style="list-style-type: none"> 1. AWS Management Console에 로그인한 다음에 AWS Glue 콘솔을 엽니다. 2. 탐색 창에서 데이터 카탈로그의 카탈로그 설정을 선택합니다. 3. 메타데이터 암호화 및 연결 암호 암호화 설정에 플래그가 지정되고 KMSKeyGlue 을(를) 사용하도록 구성되어 있는지 확인합니다. 	클라우드 아키텍트

암호화 적용 테스트

작업	설명	필요한 기술
<p>에서 암호화 설정을 CloudFormation 식별하십시오.</p>	<ol style="list-style-type: none"> 1. AWS 관리 콘솔에 로그인한 다음 CloudFormation 콘솔 을 엽니다. 2. 탐색 창에서 스택을 선택한 후 해당 스택을 선택합니다. 3. 리소스 탭을 선택합니다. 4. 리소스 테이블에서 논리적 ID별 암호화 설정을 찾습니다. 	클라우드 아키텍트
<p>프로비저닝된 인프라를 미준수 상태로 전환합니다.</p>	<ol style="list-style-type: none"> 1. AWS Management Console 에 로그인한 다음에 AWS Glue 콘솔 을 엽니다. 2. 탐색 창에서 데이터 카탈로그의 카탈로그 설정을 선택합니다. 3. 메타데이터 암호화 확인란의 선택을 취소합니다. 4. 연결 암호 암호화 확인란의 선택을 취소합니다. 5. 저장을 선택합니다. 6. AWS Glue 콘솔을 새로 고칩니다. <p>가드레일은 확인란의 선택을 취소한 후 AWS Glue에서 규정 미준수 상태를 탐지한 다음 잘못된 암호화 구성을 자동으로 수정하여 규정 준수를 적용합니다. 이에 따라, 페이지를 새로</p>	클라우드 아키텍트

작업	설명	필요한 기술
	고친 후 암호화 확인란을 다시 선택해야 합니다.	

관련 리소스

- [AWS CloudFormation 콘솔에서 스택 생성](#) (AWS CloudFormation 설명서)
- [AWS를 사용하여 AWS API 호출에서 트리거되는 CloudWatch 이벤트 규칙 생성 CloudTrail](#) (아마존 CloudWatch 설명서)
- [AWS Glue에서 암호화 설정](#)(AWS Glue 설명서)

AWS Glue를 사용하여 Amazon S3에서 Amazon Redshift로 데이터를 점차 늘려 로딩하기 위한 ETL 서비스 파이프라인 빌드

제작자: Rohan Jamadagni(AWS)와 Arunabha Datta(AWS)

환경: 프로덕션

기술: 분석, 데이터 레이크, 스토리지 및 백업

AWS 서비스: Amazon Redshift, Amazon S3, AWS Glue, AWS Lambda

요약

이 패턴은 최적의 데이터 레이크 성능을 위해 Amazon Simple Storage Service(S3)를 구성한 다음, AWS Glue를 사용하여 추출, 변환, 적재(ETL) 작업을 수행하여 Amazon S3에서 Amazon Redshift로 점진적인 데이터 변경 사항을 로드하는 방법에 대한 지침을 제공합니다.

Amazon S3의 소스 파일은 쉼표로 구분된 값 (CSV), XML, JSON 파일 등 다양한 형식을 가질 수 있습니다. 이 패턴은 AWS Glue를 사용하여 소스 파일을 Apache Parquet과 같이 비용 최적화 및 성능이 최적화된 형식으로 변환하는 방법을 설명합니다. Amazon Athena와 Amazon Redshift Spectrum에서 직접 파켓 파일을 쿼리할 수 있습니다. 또한 Parquet 파일을 Amazon Redshift로 로드하여 집계하고 집계된 데이터를 소비자와 공유하거나 Amazon을 사용하여 데이터를 시각화할 수 있습니다. QuickSight

사전 조건 및 제한 사항

사전 조건

- 활성 상태의 AWS 계정
- 적절한 권한이 있고 CSV, XML 또는 JSON 파일을 포함하는 S3 소스 버킷.

가정

- CSV, XML 또는 JSON 소스 파일은 이미 Amazon S3에 로드되어 있으며, AWS Glue 및 Amazon Redshift가 구성된 계정에서 액세스할 수 있습니다.
- [Amazon Redshift 설명서](#)에 설명된 대로 파일 로드, 파일 분할, 압축 및 매니페스트 사용에 대한 모범 사례를 따릅니다.
- 소스 파일 구조는 변경되지 않습니다.

- 소스 시스템은 Amazon S3에 정의된 폴더 구조를 따라 Amazon S3에 데이터를 수집할 수 있습니다.
- Amazon Redshift 클러스터는 단일 가용 영역에 걸쳐 있습니다. (이 아키텍처는 AWS Lambda, AWS Glue 및 Amazon Athena가 서버리스이기 때문에 적절합니다.)고가용성을 위해 클러스터 스냅샷은 일정한 빈도로 촬영됩니다.

제한 사항

- 파일 형식은 [현재 AWS Glue에서 지원](#)하는 형식으로 제한됩니다.
- 실시간 다운스트림 보고는 지원되지 않습니다.

아키텍처

소스 기술 스택

- CSV, XML 또는 JSON 파일이 포함된 S3 버킷

대상 기술 스택

- S3 데이터 레이크(파티셔닝된 Parquet 파일 스토리지 포함)
- Amazon Redshift

대상 아키텍처

데이터 흐름

도구

- [Amazon S3](#) - Amazon Simple Storage Service(S3)는 확장성이 뛰어난 객체 스토리지 서비스입니다. Amazon S3는 웹 사이트, 모바일 애플리케이션, 백업, 데이터 레이크 등 다양한 스토리지 솔루션에 사용할 수 있습니다.
- [AWS Lambda](#) - AWS Lambda는 서버를 프로비저닝하거나 관리하지 않고도 코드를 실행할 수 있습니다. AWS Lambda는 이벤트 기반 서비스이므로 다른 AWS 서비스에서 자동으로 시작되도록 코드를 설정할 수 있습니다.

- [Amazon Redshift](#) - Amazon Redshift는 클라우드의 완전 관리형 페타바이트 규모 데이터 웨어하우스 서비스입니다. Amazon Redshift를 사용하면 표준 SQL을 사용하여 데이터 웨어하우스와 데이터 레이크 전반에서 페타바이트 규모의 정형 및 반정형 데이터를 쿼리할 수 있습니다.
- [AWS Glue](#) - AWS Glue는 고객이 분석할 데이터를 쉽게 준비하고 로드할 수 있는 완전관리형 ETL 서비스입니다. AWS Glue는 데이터를 검색하고 관련 메타데이터(예: 테이블 정의, 스키마 등)를 AWS Glue 데이터 카탈로그에 저장합니다. 카탈로그로 만들어지면 데이터를 즉시 검색하고 쿼리하고 ETL에 사용할 수 있습니다.
- [AWS Secrets Manager](#) — AWS Secrets Manager를 사용하면 애플리케이션 또는 서비스 액세스에 필요한 비밀을 보호하고 중앙에서 관리할 수 있습니다. 이 서비스는 데이터베이스 보안 인증 정보, API 키, 기타 보안 암호를 저장하므로 민감한 정보를 일반 텍스트 형식으로 하드코딩할 필요가 없습니다. Secrets Manager는 또한 보안 및 규정 준수 요구 사항을 충족하기 위해 키 교체를 제공합니다. Amazon Redshift, Amazon Relational Database Service(RDS), Amazon DocumentDB 및 Amazon DocumentDB에 대한 통합 기능이 내장되어 있습니다. Secrets Manager 콘솔, 명령줄 인터페이스 (CLI) 또는 Secrets Manager API 및 SDK를 사용하여 비밀을 저장하고 중앙에서 관리할 수 있습니다.
- [Amazon Athena](#) - Amazon Athena는 Amazon S3에 저장된 데이터를 간편하게 분석할 수 있는 대화형 쿼리 서비스입니다. Athena는 서버리스이며 AWS Glue와 통합되어 있으므로 AWS Glue를 사용하여 카탈로그에 등록된 데이터를 직접 쿼리할 수 있습니다. Athena는 탄력적으로 확장되어 대화형 쿼리 성능을 제공합니다.

에픽

S3 버킷 및 폴더 구조 생성

작업	설명	필요한 기술
소스 시스템의 데이터 구조 및 속성을 분석합니다.	Amazon S3 데이터 레이크에 포함되는 데이터 소스에 대해 이 작업을 수행합니다.	데이터 엔지니어
파티션 및 액세스 전략을 정의합니다.	이 전략은 데이터 캡처 빈도, 델타 처리 및 소비 요구 사항을 기반으로 해야 합니다. S3 버킷은 일반에 공개되지 않도록 하고 액세스는 특정 서비스 역할 기반 정책으로만 제어해야 합니다.	데이터 엔지니어

작업	설명	필요한 기술
<p>각 데이터 소스 유형에 대해 별도의 S3 버킷을 생성하고 처리된 (Parquet) 데이터에 대해 소스별로 별도의 S3 버킷을 생성합니다.</p>	<p>다. 자세한 내용은 Amazon S3 설명서를 참조하십시오.</p> <p>각 소스에 대해 별도의 버킷을 생성한 다음, 소스 시스템의 데이터 수집 빈도에 따라 폴더 구조를 생성합니다. 예: s3://source-system-name/date/hour . 처리된 (Parquet 형식으로 변환된) 파일의 경우 비슷한 구조를 생성합니다. 예: s3://source-processed-bucket/date/hour . S3 버킷 생성에 대한 자세한 내용은 Amazon S3 설명서를 참조하세요.</p>	<p>데이터 엔지니어</p>

대상 Amazon Redshift 데이터 웨어하우스 생성

작업	설명	필요한 기술
<p>적절한 파라미터 그룹과 유지관리 및 백업 전략을 사용하여 Amazon Redshift 클러스터를 시작합니다.</p>	<p>Amazon Redshift 클러스터를 생성할 때는 Secrets Manager 데이터베이스 암호를 관리자 보안 인증 정보로 사용하세요. Amazon Redshift 클러스터를 생성하고 크기를 조정하는 방법에 대한 자세한 내용은 Amazon Redshift 설명서 및 클라우드 데이터 웨어하우스 크기 조정 백서를 참조하세요.</p>	<p>데이터 엔지니어</p>
<p>IAM 서비스 역할을 생성하여 Amazon Redshift 클러스터에 연결합니다.</p>	<p>AWS Identity and Access Management(IAM) 서비스 역할은 Secrets Manager 및 소스</p>	<p>데이터 엔지니어</p>

작업	설명	필요한 기술
	S3버킷에 대한 액세스를 보장합니다. 자세한 내용은 권한 부여 및 역할 추가 에 대한 AWS 설명서를 참조하세요.	
데이터베이스 스키마를 생성합니다.	Amazon Redshift 테이블 설계 모범 사례 사용 사례에 따라 적절한 정렬 및 배포 키와 가능한 최상의 압축 인코딩을 선택합니다. 모범 사례는 AWS 설명서 를 참조하세요.	데이터 엔지니어
워크로드 관리 구성	요구 사항에 따라 워크로드 관리(WLM) 대기열, 짧은 쿼리 가속화(SQA) 또는 동시성 확장을 구성하세요. 자세한 내용은 Amazon Redshift 설명서의 워크로드 관리 구현 을 참고하세요.	데이터 엔지니어

Secrets Manager에서 보안 암호 생성

작업	설명	필요한 기술
새 암호를 생성하여 Amazon Redshift 로그인 보안 인증 정보를 Secrets Manager에 저장합니다.	이 암호는 개별 데이터베이스 서비스 사용자뿐 아니라 관리자 사용자의 보안 인증 정보를 저장합니다. 자세한 지침은 Secrets Manager 설명서 를 참조하세요. Amazon Redshift 클러스터를 암호 유형으로 선택합니다. 또한 암호 교체 페이지에서 교체를 활성화하세요. 그러면 Amazon Redshift 클러스터에 적절한 사용자가 생성되	데이터 엔지니어

작업	설명	필요한 기술
	고 정의된 간격에 따라 키 암호가 교체됩니다.	
IAM 정책을 생성하여 Secrets Manager 액세스를 제한합니다.	Secrets Manager에 대한 액세스를 Amazon Redshift 관리자 와 AWS Glue로만 제한합니다.	데이터 엔지니어

AWS Glue 구성

작업	설명	필요한 기술
AWS Glue 데이터 카탈로그에서 Amazon Redshift에 대한 연결을 추가합니다.	지침은 AWS Glue 설명서 를 참조하세요.	데이터 엔지니어
Secrets Manager, Amazon Redshift 및 S3 버킷에 액세스할 수 있도록 AWS Glue용 IAM 서비스 역할을 생성하고 연결합니다.	자세한 내용은 AWS Glue 설명서 를 참조하세요.	데이터 엔지니어
해당 소스의 AWS Glue 데이터 카탈로그를 정의합니다.	이 단계에는 AWS Glue 데이터 카탈로그에 데이터베이스와 필수 테이블을 생성하는 작업이 포함됩니다. 크롤러를 사용하여 AWS Glue 데이터베이스의 테이블을 카탈로그화하거나 Amazon Athena 외부 테이블로 정의할 수 있습니다. 또한 AWS Glue 데이터 카탈로그를 통해 Athena에 정의된 외부 테이블에 액세스할 수 있습니다. Athena에서 데이터 카탈로그 정의 를 하고 외부 테이블 생성 을 하는 방법에 대한 자세한	데이터 엔지니어

작업	설명	필요한 기술
	내용은 AWS 설명서를 참조하세요.	
AWS Glue 작업을 생성하여 소스 데이터를 처리합니다.	AWS Glue 작업은 Python 셀일 수도 있고 원본 PySpark 데이터 파일을 표준화, 중복 제거 및 정리하기 위한 것일 수도 있습니다. 성능을 최적화하고 전체 S3 원본 버킷을 쿼리하지 않으려면 AWS Glue 작업의 푸시다운 조건으로 S3 버킷을 날짜, 연도, 월, 일, 시간별로 분류하여 파티션을 나눕니다. 자세한 내용은 AWS Glue 설명서 를 참조하세요. 처리 및 변환된 데이터를 처리된 S3 버킷 파티션에 Parquet 형식으로 로드합니다. Athena에서 파켓 파일을 쿼리할 수 있습니다.	데이터 엔지니어
Amazon Redshift로 데이터를 로드하는 AWS Glue 작업을 생성합니다.	AWS Glue 작업은 Python 셀일 수도 PySpark 있고 데이터를 업로드한 후 전체 새로 고침하여 데이터를 로드하는 것일 수도 있습니다. 자세한 내용은 AWS Glue 설명서 및 추가 정보 단원을 참조하세요.	데이터 엔지니어

작업	설명	필요한 기술
<p>(선택 사항) 필요에 따라 트리거를 사용하여 AWS Glue 작업을 예약합니다.</p>	<p>중분 데이터 로드는 주로 AWS Lambda 함수가 AWS Glue 작업을 직접적으로 호출하도록 하는 Amazon S3 이벤트에 의해 구동됩니다. 이벤트 기반 일정 대신 시간 기반 일정을 요구하는 모든 데이터 로드에는 AWS Glue 트리거 기반 예약을 사용하세요.</p>	<p>데이터 엔지니어</p>

Lambda 함수 생성

작업	설명	필요한 기술
<p>AWS Lambda가 S3 버킷과 AWS Glue 작업에 액세스할 수 있도록 IAM 서비스 연결 역할을 생성하고 연결합니다.</p>	<p>Amazon S3 객체 및 버킷을 읽는 정책과 AWS Glue API에 액세스하여 AWS Glue 작업을 시작하는 정책을 사용하여 AWS Lambda용 IAM 서비스 연결 역할을 생성합니다. 자세한 정보는 지식 센터를 참조하세요.</p>	<p>데이터 엔지니어</p>
<p>정의된 Amazon S3 이벤트를 기반으로 AWS Glue 작업을 실행하는 Lambda 함수를 생성합니다.</p>	<p>Amazon S3 매니페스트 파일을 생성하여 Lambda 함수를 시작해야 합니다. Lambda 함수는 Amazon S3 폴더 위치(예: source_bucket/년/월/날짜/시간)를 AWS Glue 작업에 파라미터로 전달해야 합니다. AWS Glue 작업은 이 파라미터를 푸시다운 조건자로 사용하여 파일 액세스 및 작업 처리 성능을 최적화합니다. 자세한 내용은</p>	<p>데이터 엔지니어</p>

작업	설명	필요한 기술
	AWS Glue 설명서 를 참조하세요.	
Amazon S3 PUT 객체 이벤트를 생성하여 객체 생성을 감지하고 해당 Lambda 함수를 직접적으로 호출합니다.	Amazon S3 PUT 객체 이벤트는 매니페스트 파일 생성을 통해서만 시작해야 합니다. 매니페스트 파일은 Lambda 함수와 AWS Glue 작업 동시성을 제어하며, S3 소스 버킷의 특정 파티션에 도착하는 개별 파일을 처리하는 대신 로드를 일괄 처리합니다. 자세한 내용은 Lambda 설명서 를 참조하세요.	데이터 엔지니어

관련 리소스

- [Amazon S3 설명서](#)
- [AWS Glue 설명서](#)
- [Amazon Redshift 설명서](#)
- [AWS Lambda](#)
- [Amazon Athena](#)
- [AWS Secrets Manager](#)

추가 정보

완전히 새로고침하기 위한 세부 접근 방식

업서트: 비즈니스 사용 사례에 따라 기록의 집계가 필요한 데이터 세트를 위한 것입니다. 비즈니스에 필요한 사항에 따라 [새 데이터 업데이트 및 삽입](#) (Amazon Redshift 설명서)에 설명된 접근 방식 중 하나를 따르십시오.

전체 새로 고침: 기록 집계가 필요하지 않은 소규모 데이터 세트를 위한 것입니다. 다음 접근법 중 하나를 따르세요.

1. Amazon Redshift 테이블을 자릅니다.
2. 스테이징 영역에서 현재 파티션을 로드합니다.

또는:

1. 현재 파티션 데이터를 사용하여 임시 테이블을 생성합니다.
2. 대상 Amazon Redshift의 대상 테이블을 삭제합니다.
3. 임시 테이블의 이름을 대상 테이블로 변경합니다.

AWS 서비스를 사용하여 위험 가치(VaR) 계산

작성자: Sumon Samanta(AWS)

환경: PoC 또는 파일럿	기술: 분석, 서버리스	AWS 서비스: 아마존 키네시스 데이터 스트림, AWS Lambda, 아마존 SQS, 아마존 ElastiCache
----------------	--------------	---

요약

이 패턴은 AWS 서비스를 사용하여 위험 가치(VaR) 계산 시스템을 구현하는 방법을 설명합니다. 온프레미스 환경에서 대부분의 VaR 시스템은 대규모의 전용 인프라와 사내 또는 상용 그리드 스케줄링 소프트웨어를 사용하여 배치 프로세스를 실행합니다. 이 패턴은 AWS 클라우드에서 VaR 프로세스를 처리할 수 있는 간단하고 안정적이며 확장 가능한 아키텍처를 제공합니다. Amazon Kinesis Data Streams를 스트리밍 서비스로, Amazon Simple Queue Service (Amazon SQS)를 관리형 대기열 서비스로, Amazon을 캐싱 서비스로, ElastiCache AWS Lambda를 사용하여 주문을 처리하고 위험을 계산하는 서버리스 아키텍처를 구축합니다.

VaR은 트레이더와 위험 관리자가 포트폴리오의 잠재적 손실을 특정 신뢰 수준 이상으로 추정하는 데 사용하는 통계 지표입니다. 대부분의 VaR 시스템에는 많은 수의 수학적 및 통계적 계산을 실행하고 결과를 저장하는 작업이 포함됩니다. 이러한 계산에는 상당한 컴퓨팅 리소스가 필요하므로 VaR 배치 프로세스를 더 작은 컴퓨팅 작업 세트로 나누어야 합니다. 대규모 배치를 작은 작업으로 분할하는 것은 이러한 작업이 대부분 독립적이기 때문에 가능합니다. 즉, 한 작업의 계산이 다른 작업에 종속되지 않습니다.

VAR 아키텍처의 또 다른 중요한 요구 사항은 컴퓨팅 확장성입니다. 이 패턴은 컴퓨팅 부하에 따라 자동으로 규모를 늘리거나 줄이는 서버리스 아키텍처를 사용합니다. 배치 또는 온라인 컴퓨팅 수요를 예측하기 어렵기 때문에 서비스 수준에 관한 계약(SLA)에 따른 일정 내에 프로세스를 완료하려면 동적인 규모 조정이 필요합니다. 또한 비용 최적화된 아키텍처는 해당 리소스의 작업이 완료되는 즉시 각 컴퓨팅 리소스를 스케일 다운할 수 있어야 합니다.

AWS 서비스는 확장 가능한 컴퓨팅 및 스토리지 용량, 비용 최적화된 방식으로 처리하는 분석 서비스, 위험 관리 워크플로를 실행할 수 있는 다양한 유형의 스케줄러를 제공하므로 VaR 계산에 매우 적합합니다. 또한 AWS에서 사용한 컴퓨팅 및 스토리지 리소스에 대해서만 비용을 지불합니다.

사전 조건 및 제한 사항

사전 조건

- 활성 상태의 AWS 계정.
- 입력 파일(비즈니스 요구 사항에 따라 다름). 일반적인 사용 사례에는 다음과 같은 입력 파일이 포함됩니다.
 - 시장 데이터 파일(VaR 계산 엔진에 입력)
 - 거래 데이터 파일(거래 데이터가 스트림을 통해 제공되는 경우 제외).
 - 구성 데이터 파일(모델 및 기타 정적 구성 데이터)
 - 계산 엔진 모델 파일(정량적 라이브러리)
 - 시계열 데이터 파일(지난 5년간의 주가와 같은 과거 데이터용)
- 시장 데이터 또는 기타 입력이 스트림을 통해 들어오는 경우 Amazon Kinesis Data Streams가 설정되고 스트림에 쓰도록 Amazon Identity and Access Management(IAM) 권한이 구성됩니다.

이 패턴은 트레이딩 시스템에서 Kinesis 데이터 스트림으로 거래 데이터를 쓰는 아키텍처를 구축합니다. 스트리밍 서비스를 사용하는 대신, 작은 배치 파일에 거래 데이터를 저장하여 Amazon Simple Storage Service(S3) 버킷에 저장한 다음, 이벤트를 간접적으로 호출하여 데이터 처리를 시작할 수 있습니다.

제한 사항

- Kinesis 데이터 스트림 시퀀싱은 각 샤드에서 보장되므로 여러 샤드에 작성된 거래 주문은 쓰기 작업과 동일한 순서로 전달되지 않을 수 있습니다.
- AWS Lambda 런타임 제한은 현재 15분입니다. (자세한 내용은 [Lambda FAQ](#)를 참조하세요.)

아키텍처

대상 아키텍처

다음 아키텍처 다이어그램은 위험 평가 시스템의 AWS 서비스와 워크플로를 보여줍니다.

다이어그램은 다음을 보여 줍니다.

1. 주문 관리 시스템에서 거래가 유입됩니다.

2. 티켓 포지션 네팅 Lambda 함수는 주문을 처리하고 각 티커에 대한 통합 메시지를 Amazon SQS의 위험 대기열에 기록합니다.
3. 위험 계산 엔진 Lambda 함수는 Amazon SQS의 메시지를 처리하고, 위험 계산을 수행하고, Amazon의 위험 캐시에 있는 VaR 손익 (PnL) 정보를 업데이트합니다. ElastiCache
4. 읽기 ElastiCache 데이터 Lambda 함수는 데이터베이스 및 S3 ElastiCache 버킷에서 위험 결과를 검색하고 이를 데이터베이스와 S3 버킷에 저장합니다.

이러한 서비스와 단계에 대한 자세한 내용은 에픽 섹션을 참조하세요.

자동화 및 규모 조정

AWS 클라우드 개발 키트 (AWS CDK) 또는 AWS CloudFormation 템플릿을 사용하여 전체 아키텍처를 배포할 수 있습니다. 이 아키텍처는 일괄 처리와 당일(실시간) 처리를 모두 지원할 수 있습니다.

규모 조정은 아키텍처에 내장되어 있습니다. 더 많은 거래가 Kinesis 데이터 스트림에 기록되고 처리 대기 중인 경우 추가 Lambda 함수를 간접적으로 호출하여 해당 거래를 처리한 다음 처리가 완료된 후 규모를 축소할 수 있습니다. 여러 Amazon SQS 위험 계산 대기열을 통한 처리 옵션도 있습니다. 대기열 전체에 엄격한 순서 지정 또는 통합이 필요한 경우 처리를 병렬화할 수 없습니다. 하지만 end-of-the-day 일괄처리 또는 미니 인트라데이 배치의 경우 Lambda 함수가 병렬로 처리하여 최종 결과를 저장할 수 있습니다. ElastiCache

도구

AWS 서비스

- [Amazon Aurora MySQL-Compatible Edition](#)은 MySQL 배포의 설정, 운영, 규모 조정에 도움이 되는 완전 관리형의 MySQL 호환 관계형 데이터베이스 엔진입니다. 이 패턴은 MySQL을 예로 사용하지만 모든 RDBMS 시스템을 사용하여 데이터를 저장할 수 있습니다.
- [ElastiCacheAmazon](#)은 AWS 클라우드에서 분산된 인메모리 캐시 환경을 설정, 관리 및 확장할 수 있도록 지원합니다.
- [Amazon Kinesis Data Streams](#)를 사용하여 대규모 데이터 레코드 스트림을 실시간으로 수집하고 처리할 수 있습니다.
- [AWS Lambda](#)는 서버를 프로비저닝하거나 관리할 필요 없이 코드를 실행하는 데 도움이 되는 컴퓨팅 서비스입니다. 필요할 때만 코드를 실행하며 자동으로 확장이 가능하므로 사용한 컴퓨팅 시간만큼만 비용을 지불합니다.
- [Amazon Simple Queue Service\(Amazon SQS\)](#)는 내구력 있고 가용성이 뛰어난 보안 호스팅 대기열을 제공하며 이를 통해 분산 소프트웨어 시스템과 구성 요소를 통합 및 분리할 수 있습니다.

- [Amazon Simple Storage Service\(S3\)](#)는 원하는 양의 데이터를 저장, 보호 및 검색할 수 있는 클라우드 기반 객체 스토리지 서비스입니다.

code

이 패턴은 AWS 클라우드의 VaR 시스템을 위한 예제 아키텍처를 제공하고 VaR 계산에 Lambda 함수를 사용하는 방법을 설명합니다. Lambda 함수를 생성하려면 [Lambda 설명서](#)의 코드 예제를 참조하세요. 도움이 필요하면 [AWS Professional Services](#)에 문의하세요.

모범 사례

- 각 VaR 컴퓨팅 작업을 최대한 작고 가볍게 유지합니다. 각 컴퓨팅 작업에서 서로 다른 수의 거래를 실험하여 어떤 것이 컴퓨팅 시간과 비용에 가장 최적화되어 있는지 확인해 보세요.
- ElastiCacheAmazon에 재사용 가능한 객체를 저장합니다. Apache Arrow와 같은 프레임워크를 사용하여 직렬화 및 역직렬화를 줄이세요.
- Lambda의 시간 제한을 고려해 보세요. 컴퓨팅 작업이 15분을 초과할 것으로 생각되면 Lambda 시간 초과가 발생하지 않도록 작은 작업으로 나눕니다. 이것이 가능하지 않다면 AWS Fargate, Amazon Elastic Container Service(Amazon ECS), Amazon Elastic Kubernetes Service(Amazon EKS)의 컨테이너 오케스트레이션 솔루션을 고려해 볼 수 있습니다.

에픽

리스크 시스템으로의 거래 흐름

작업	설명	필요한 기술
거래 작성을 시작합니다.	신규, 정산 또는 부분 정산 거래가 주문 관리 시스템에서 리스크 스트림으로 기록됩니다. 이 패턴은 Amazon Kinesis를 관리형 스트리밍 서비스로 사용합니다. 거래 주문 티커의 해시는 여러 샤드에 거래 주문을 보내는 데 사용됩니다.	Amazon Kinesis

주문 처리를 위한 Lambda 함수 실행

작업	설명	필요한 기술
Lambda로 위험 처리를 시작합니다.	<p>새 주문에 대해 AWS Lambda 함수를 실행합니다. 보류 중인 거래 주문 수에 따라 Lambda가 자동으로 규모를 조정합니다. 각 Lambda 인스턴스에는 하나 이상의 주문이 있으며 Amazon에서 각 티커의 최신 포지션을 검색합니다. ElastiCache (다른 금융 파생상품의 CUSIP ID, Curve 이름 또는 인덱스 이름을 데이터를 저장하고 검색하는 키로 사용할 수 있습니다.) ElastiCache에서는 ElastiCache 전체 포지션 (수량) 과 키-값 쌍 < 티커, 넷 포지션 > (여기서 순 포지션은 스케일링 팩터임) 이 각 티커에 대해 한 번씩 업데이트됩니다.</p>	아마존 키네시스, AWS 람다, 아마존 ElastiCache

각 티커에 대한 메시지를 대기열에 기록

작업	설명	필요한 기술
통합 메시지를 위험 대기열에 기록합니다.	<p>메시지를 대기열에 기록합니다. 이 패턴은 Amazon SQS를 관리형 대기열 서비스로 사용합니다. 단일 Lambda 인스턴스는 언제든지 소량의 거래 주문을 받을 수 있지만 Amazon SQS에는 각 티커에 대해 단일 메시지만 기록합니다. 스케일</p>	Amazon SQS, AWS Lambda

작업	설명	필요한 기술
	링 팩터는 (기존 순 포지션 + 현재 포지션) / 기존 순 포지션과 같이 계산됩니다.	

리스크 엔진 간접 호출

작업	설명	필요한 기술
위험 계산을 시작합니다.	리스크 엔진 Lambda에 대한 Lambda 함수가 간접적으로 호출됩니다. 각 위치는 단일 Lambda 함수에 의해 처리됩니다. 하지만 최적화를 위해 각 Lambda 함수는 Amazon SQS의 여러 메시지를 처리할 수 있습니다.	Amazon SQS, AWS Lambda

캐시에서 위험 결과 검색

작업	설명	필요한 기술
위험 캐시를 검색 및 업데이트합니다.	Lambda는 에서 각 티커의 현재 순 포지션을 검색합니다. ElastiCache 또한 각 티커에 대한 VaR 손익 (PnL) 배열을 검색합니다. ElastiCache PnL 배열이 이미 존재하는 경우, Lambda 함수는 네팅 Lambda 함수가 작성한 Amazon SQS 메시지에서 가져온 스케일을 사용하여 배열과 VaR을 업데이트합니다. PnL 배열이 포함되지 않은 경우 시	아마존 SQS, AWS 람다, 아마존 ElastiCache

작업	설명	필요한 기술
	물레이션된 티커 ElasticCache 가격 시리즈 데이터를 사용하여 새 PnL과 VaR을 계산합니다.	

Elastic Cache에서 데이터를 업데이트하고 데이터베이스에 저장

작업	설명	필요한 기술
위험 결과를 저장합니다.	에서 ElastiCache VaR 및 PnL 번호가 업데이트된 후 5분마다 새로운 Lambda 함수가 호출됩니다. 이 함수는 저장된 모든 데이터를 Aurora MySQL 호환 데이터베이스와 S3 버킷에서 ElastiCache 읽고 저장합니다.	AWS 람다, 아마존 ElastiCache

관련 리소스

- [바젤 VaR 프레임워크](#)

Teradata NORMALIZE 임시 기능을 Amazon Redshift SQL로 변환

작성자: Po Hong(AWS)

소스: Teradata 데이터 웨어하우스	대상: Amazon Redshift	R 타입: 리아키텍트
환경: 프로덕션	기술: 분석, 데이터베이스, 마이그레이션	워크로드: 기타 모든 워크로드
AWS 서비스: Amazon Redshift		

요약

정규화는 ANSI SQL 표준의 테라데이터 확장입니다. SQL 테이블에 기간 데이터 유형의 열이 포함된 경우 정규화는 해당 열과 일치하거나 겹치는 값을 결합하여 여러 개별 기간 값을 통합하는 단일 기간을 형성합니다. 정규화를 사용하려면 SQL 선택 목록에서 하나 이상의 열이 Teradata의 임시 기간 데이터 유형이어야 합니다. 정규화에 대한 자세한 내용은 [Teradata 설명서](#)를 참조하세요.

Amazon Redshift는 정규화를 지원하지 않지만 Amazon Redshift의 네이티브 SQL 구문과 LAG 윈도우 함수를 사용하여 이 기능을 구현할 수 있습니다. 이 패턴은 가장 많이 사용되는 형식인 ON MEETS 또는 OVERLAPS 조건과 함께 테라데이터 정규화 확장 프로그램을 사용하는 데 중점을 둡니다. 테라데이터에서 이 기능이 작동하는 방식과 이 기능을 Amazon Redshift 네이티브 SQL 구문으로 변환하는 방법을 설명합니다.

사전 조건 및 제한 사항

사전 조건

- 테라데이터 SQL에 대한 기본 지식 및 경험
- Amazon Redshift에 대한 지식과 경험

아키텍처

소스 기술 스택

- Teradata 데이터 웨어하우스

대상 기술 스택

- Amazon Redshift

대상 아키텍처

Teradata 데이터베이스를 Amazon Redshift로 마이그레이션하기 위한 전체적인 아키텍처는 [AWS SCT 데이터 추출 에이전트를 사용하여 Teradata 데이터베이스를 Amazon Redshift로 마이그레이션](#) 패턴을 참조하세요. 마이그레이션은 테라데이타 정규화 구문을 Amazon Redshift SQL로 자동으로 변환하지 않습니다. 이 패턴의 지침에 따라 이 테라데이타 확장을 변환할 수 있습니다.

도구

코드

정규화의 개념과 기능을 설명하려면 Teradata에 있는 다음 테이블 정의를 살펴보세요.

```
CREATE TABLE systest.project
(
  emp_id      INTEGER,
  project_name VARCHAR(20),
  dept_id     INTEGER,
  duration    PERIOD(DATE)
);
```

다음 SQL 코드를 실행하여 샘플 데이터를 테이블에 삽입합니다.

```
BEGIN TRANSACTION;

INSERT INTO systest.project VALUES (10, 'First Phase', 1000, PERIOD(DATE '2010-01-10',
DATE '2010-03-20')));
INSERT INTO systest.project VALUES (10, 'First Phase', 2000, PERIOD(DATE '2010-03-20',
DATE '2010-07-15')));

INSERT INTO systest.project VALUES (10, 'Second Phase', 2000, PERIOD(DATE
'2010-06-15', DATE '2010-08-18')));
INSERT INTO systest.project VALUES (20, 'First Phase', 2000, PERIOD(DATE '2010-03-10',
DATE '2010-07-20')));
```

```
INSERT INTO systest.project VALUES (20, 'Second Phase', 1000, PERIOD(DATE
  '2020-05-10', DATE '2020-09-20') );

END TRANSACTION;
```

결과:

```
select * from systest.project order by 1,2,3;

*** Query completed. 4 rows found. 4 columns returned.
*** Total elapsed time was 1 second.
```

emp_id	project_name	dept_id	duration
10	First Phase	1000	('10/01/10', '10/03/20')
10	First Phase	2000	('10/03/20', '10/07/15')
10	Second Phase	2000	('10/06/15', '10/08/18')
20	First Phase	2000	('10/03/10', '10/07/20')
20	Second Phase	1000	('20/05/10', '20/09/20')

테라데이터 정규화 사용 사례

이제 선택 문에 테라데이터 정규화 SQL 절을 추가합니다.

```
SELECT NORMALIZE ON MEETS OR OVERLAPS emp_id, duration
FROM systest.project
ORDER BY 1,2;
```

이 정규화 작업은 단일 열(emp_id)에서 수행됩니다. emp_id=10의 경우 다음과 같이 기간에서 겹치는 세 개의 기간 값이 단일 기간 값으로 합쳐집니다.

emp_id	duration
10	('10/01/10', '10/08/18')
20	('10/03/10', '10/07/20')
20	('20/05/10', '20/09/20')

다음 선택 문은 project_name 및 dept_id.에 대해 정규화 작업을 수행합니다. 참고로 선택 목록에는 기간 열인 기간 하나만 포함되어 있습니다.

```
SELECT NORMALIZE project_name, dept_id, duration
```

```
FROM systest.project;
```

출력:

project_name	dept_id	duration
First Phase	1000	('10/01/10', '10/03/20')
Second Phase	1000	('20/05/10', '20/09/20')
First Phase	2000	('10/03/10', '10/07/20')
Second Phase	2000	('10/06/15', '10/08/18')

Amazon Redshift에 해당하는 SQL

Amazon Redshift는 현재 테이블의 기간 데이터 유형을 지원하지 않습니다. 대신 다음과 같이 테라데이타 기간 데이터 필드를 start_date, end_date의 두 부분으로 나누어야 합니다.

```
CREATE TABLE systest.project
(
  emp_id          INTEGER,
  project_name    VARCHAR(20),
  dept_id         INTEGER,
  start_date      DATE,
  end_date        DATE
);
```

다음 테이블에 샘플 데이터를 삽입합니다.

```
BEGIN TRANSACTION;

INSERT INTO systest.project VALUES (10, 'First Phase', 1000, DATE '2010-01-10', DATE
'2010-03-20' );
INSERT INTO systest.project VALUES (10, 'First Phase', 2000, DATE '2010-03-20', DATE
'2010-07-15');

INSERT INTO systest.project VALUES (10, 'Second Phase', 2000, DATE '2010-06-15', DATE
'2010-08-18' );
INSERT INTO systest.project VALUES (20, 'First Phase', 2000, DATE '2010-03-10', DATE
'2010-07-20' );

INSERT INTO systest.project VALUES (20, 'Second Phase', 1000, DATE '2020-05-10', DATE
'2020-09-20' );
```

```
END TRANSACTION;
```

출력:

emp_id	project_name	dept_id	start_date	end_date
10	First Phase	1000	2010-01-10	2010-03-20
10	First Phase	2000	2010-03-20	2010-07-15
10	Second Phase	2000	2010-06-15	2010-08-18
20	First Phase	2000	2010-03-10	2010-07-20
20	Second Phase	1000	2020-05-10	2020-09-20

(5 rows)

테라데이터의 정규화 절을 다시 작성하려면 Amazon Redshift의 [LAG 윈도우 함수](#)를 사용하면 됩니다. 이 함수는 파티션에서 현재 행 위(앞)의 지정 오프셋에 위치한 행의 값을 반환합니다.

LAG 함수를 사용하면 기간이 이전 기간과 일치하거나 겹치는지를 판단하여 새 기간을 시작하는 각 행을 식별할 수 있습니다(예인 경우 0, 아니요인 경우 1). 이 플래그를 누적하여 합산하면 Amazon Redshift에서 원하는 결과에 도달하기 위해 외부 그룹별 절에서 사용할 수 있는 그룹 식별자가 제공됩니다.

LAG()를 사용하는 Amazon Redshift SQL 문 샘플은 다음과 같습니다.

```
SELECT emp_id, start_date, end_date,
       (CASE WHEN start_date <= LAG(end_date) OVER (PARTITION BY emp_id ORDER BY
start_date, end_date) THEN 0 ELSE 1 END) AS GroupStartFlag
FROM systest.project
ORDER BY 1,2;
```

출력:

emp_id	start_date	end_date	groupstartflag
10	2010-01-10	2010-03-20	1
10	2010-03-20	2010-07-15	0
10	2010-06-15	2010-08-18	0
20	2010-03-10	2010-07-20	1
20	2020-05-10	2020-09-20	1

(5 rows)

다음 Amazon Redshift SQL 문은 emp_id 열에서만 정규화됩니다.

```

SELECT T2.emp_id, MIN(T2.start_date) as new_start_date, MAX(T2.end_date) as
  new_end_date
FROM
( SELECT T1.*, SUM(GroupStartFlag) OVER (PARTITION BY emp_id ORDER BY start_date ROWS
  UNBOUNDED PRECEDING) As GroupID
FROM ( SELECT emp_id, start_date, end_date,
          (CASE WHEN start_date <= LAG(end_date) OVER (PARTITION BY emp_id ORDER BY
            start_date, end_date) THEN 0 ELSE 1 END) AS GroupStartFlag
FROM systest.project ) T1
) T2
GROUP BY T2.emp_id, T2.GroupID
ORDER BY 1,2;

```

출력:

```

emp_id | new_start_date | new_end_date
-----+-----+-----
    10 | 2010-01-10    | 2010-08-18
    20 | 2010-03-10    | 2010-07-20
    20 | 2020-05-10    | 2020-09-20
(3 rows)

```

다음 Amazon Redshift SQL 문은 project_name 열과 dept_id 열 모두에서 정규화됩니다.

```

SELECT T2.project_name, T2.dept_id, MIN(T2.start_date) as new_start_date,
  MAX(T2.end_date) as new_end_date
FROM
( SELECT T1.*, SUM(GroupStartFlag) OVER (PARTITION BY project_name, dept_id ORDER BY
  start_date ROWS UNBOUNDED PRECEDING) As GroupID
FROM ( SELECT project_name, dept_id, start_date, end_date,
          (CASE WHEN start_date <= LAG(end_date) OVER (PARTITION BY project_name,
            dept_id ORDER BY start_date, end_date) THEN 0 ELSE 1 END) AS GroupStartFlag
FROM systest.project ) T1
) T2
GROUP BY T2.project_name, T2.dept_id, T2.GroupID
ORDER BY 1,2,3;

```

출력:

```

project_name | dept_id | new_start_date | new_end_date

```

```

-----+-----+-----+-----
First Phase | 1000 | 2010-01-10 | 2010-03-20
First Phase | 2000 | 2010-03-10 | 2010-07-20
Second Phase | 1000 | 2020-05-10 | 2020-09-20
Second Phase | 2000 | 2010-06-15 | 2010-08-18
(4 rows)
    
```

에픽

정규화를 Amazon Redshift SQL로 변환

작업	설명	필요한 기술
테라데이타 SQL 코드를 생성합니다.	필요에 따라 Normalize 문구를 사용합니다.	SQL Developer
코드를 Amazon Redshift SQL로 변환합니다.	코드를 변환하려면 이 패턴의 '도구' 섹션에 있는 지침을 따르세요.	SQL Developer
Amazon Redshift에서 코드를 실행합니다.	테이블을 생성하고, 테이블에 데이터를 로드하고, Amazon Redshift에서 코드를 실행합니다.	SQL Developer

관련 리소스

참조

- [테라데이타 정규화 임시 기능](#)(테라데이타 설명서)
- [LAG 윈도우 함수](#)(Amazon Redshift 설명서)
- [Amazon Redshift로 마이그레이션](#)(AWS 웹 사이트)
- [AWS SCT 데이터 추출 에이전트를 사용하여 테라데이타 데이터베이스를 Amazon Redshift로 마이그레이션](#)(AWS 권장 가이드)
- [Teradata RESET WHEN 기능을 Amazon Redshift SQL로 변환](#)(AWS 권장 가이드)

도구

- [AWS Schema Conversion Tool\(AWS SCT\)](#)

파트너

- [AWS 마이그레이션 컴피턴시 파트너](#)

Teradata RESET WHEN 기능을 Amazon Redshift SQL로 변환

작성자: Po Hong(AWS)

소스: Teradata 데이터 웨어하우스	대상: Amazon Redshift	R 타입: 리아키텍트
환경: 프로덕션	기술: 분석, 데이터베이스, 마이그레이션	워크로드: 기타 모든 워크로드
AWS 서비스: Amazon Redshift		

요약

RESET WHEN은 SQL 분석 창 함수에 사용되는 Teradata 기능입니다. ANSI SQL 표준의 확장입니다. RESET WHEN은 특정 조건에 따라 SQL 창 함수가 작동하는 파티션을 결정합니다. 조건이 TRUE로 평가되면 기존 창 파티션 내에 새로운 동적 하위 파티션이 생성됩니다. RESET WHEN에 대한 자세한 내용은 [Teradata 설명서](#)를 참조하세요.

Amazon Redshift는 SQL 창 함수에서 RESET WHEN을 지원하지 않습니다. 이 기능을 구현하려면 RESET WHEN을 Amazon Redshift의 네이티브 SQL 구문으로 변환하고 여러 개의 중첩 함수를 사용해야 합니다. 이 패턴은 Teradata RESET WHEN 기능을 사용하는 방법과 이를 Amazon Redshift SQL 구문으로 변환하는 방법을 보여 줍니다.

사전 조건 및 제한 사항

사전 조건

- Teradata 데이터 웨어하우스와 해당 SQL 구문에 대한 기본 지식
- Amazon Redshift와 해당 SQL 구문에 대한 충분한 이해

아키텍처

소스 기술 스택

- Teradata 데이터 웨어하우스

대상 기술 스택

- Amazon Redshift

아키텍처

Teradata 데이터베이스를 Amazon Redshift로 마이그레이션하기 위한 전체적인 아키텍처는 [AWS SCT 데이터 추출 에이전트를 사용하여 Teradata 데이터베이스를 Amazon Redshift로 마이그레이션](#) 패턴을 참조하세요. 마이그레이션은 Teradata RESET WHEN 구문을 Amazon Redshift SQL로 자동 변환하지 않습니다. 다음 섹션의 지침에 따라 이 Teradata 확장을 변환할 수 있습니다.

도구

코드

RESET WHEN의 개념을 설명하기 위해 Teradata의 다음 테이블 정의를 생각해 보세요.

```
create table systest.f_account_balance
( account_id integer NOT NULL,
  month_id integer,
  balance integer )
unique primary index (account_id, month_id);
```

다음 SQL 코드를 실행하여 샘플 데이터를 테이블에 삽입합니다.

```
BEGIN TRANSACTION;
Insert Into systest.f_account_balance values (1,1,60);
Insert Into systest.f_account_balance values (1,2,99);
Insert Into systest.f_account_balance values (1,3,94);
Insert Into systest.f_account_balance values (1,4,90);
Insert Into systest.f_account_balance values (1,5,80);
Insert Into systest.f_account_balance values (1,6,88);
Insert Into systest.f_account_balance values (1,7,90);
Insert Into systest.f_account_balance values (1,8,92);
Insert Into systest.f_account_balance values (1,9,10);
Insert Into systest.f_account_balance values (1,10,60);
Insert Into systest.f_account_balance values (1,11,80);
Insert Into systest.f_account_balance values (1,12,10);
END TRANSACTION;
```

샘플 테이블에는 다음과 같은 데이터가 있습니다.

account_id	month_id	balance
1	1	60
1	2	99
1	3	94
1	4	90
1	5	80
1	6	88
1	7	90
1	8	92
1	9	10
1	10	60
1	11	80
1	12	10

각 계좌에 대해 연속적인 월간 잔액 증가 순서를 분석한다고 가정합니다. 한 달 잔액이 이전 달의 잔액보다 작거나 같으면 카운터를 0으로 재설정하고 다시 시작해야 합니다.

Teradata RESET WHEN 사용 사례

이 데이터를 분석하기 위해 Teradata SQL은 다음과 같이 중첩 집계와 RESET WHEN 구문이 있는 창 함수를 사용합니다.

```
SELECT account_id, month_id, balance,
       ( ROW_NUMBER() OVER (PARTITION BY account_id ORDER BY month_id
        RESET WHEN balance <= SUM(balance) over (PARTITION BY account_id ORDER BY month_id ROWS
        BETWEEN 1 PRECEDING AND 1 PRECEDING) ) -1 ) as balance_increase
FROM systest.f_account_balance
```

```
ORDER BY 1,2;
```

출력:

account_id	month_id	balance	balance_increase
1	1	60	0
1	2	99	1
1	3	94	0
1	4	90	0
1	5	80	0
1	6	88	1
1	7	90	2
1	8	92	3
1	9	10	0
1	10	60	1
1	11	80	2
1	12	10	0

쿼리는 Teradata에서 다음과 같이 처리됩니다.

1. SUM(balance) 집계 함수는 특정 달에 특정 계좌의 모든 잔액 합계를 계산합니다.
2. 특정 달의(특정 계좌의) 잔액이 이전 달의 잔액보다 큰지 확인합니다.
3. 잔액이 증가하면 누적 카운트 값을 추적합니다. RESET WHEN 조건이 거짓으로 평가되면(즉, 잔액이 몇 달 연속 증가함) 카운트를 계속 늘립니다.
4. ROW_NUMBER() 정렬 분석 함수는 카운트 값을 계산합니다. 잔액이 이전 달의 잔액보다 작거나 같은 달에 도달하면 RESET WHEN 조건이 참으로 평가됩니다. 그렇다면 새 파티션을 시작하고 ROW_NUMBER()는 카운트를 1부터 다시 시작합니다. 이전 행의 값에 액세스하려면 ROWS BETWEEN 1 PRECEDING AND 1 PRECEDING을 사용합니다.

5. 카운트 값이 0부터 시작하도록 1을 뺍니다.

Amazon Redshift에 해당하는 SQL

Amazon Redshift는 SQL 분석 창 함수의 RESET WHEN 구문을 지원하지 않습니다. 동일한 결과를 생성하려면 다음과 같이 Amazon Redshift 네이티브 SQL 구문과 중첩된 하위 쿼리를 사용하여 Teradata SQL을 다시 작성해야 합니다.

```
SELECT account_id, month_id, balance,
       (ROW_NUMBER() OVER(PARTITION BY account_id, new_dynamic_part ORDER BY month_id) -1)
       as balance_increase
FROM
( SELECT account_id, month_id, balance, prev_balance,
  SUM(dynamic_part) OVER (PARTITION BY account_id ORDER BY month_id ROWS BETWEEN
  UNBOUNDED PRECEDING AND CURRENT ROW) As new_dynamic_part
FROM ( SELECT account_id, month_id, balance,
  SUM(balance) over (PARTITION BY account_id ORDER BY month_id ROWS BETWEEN 1 PRECEDING
  AND 1 PRECEDING) as prev_balance,
  (CASE When balance <= prev_balance Then 1 Else 0 END) as dynamic_part
FROM systest.f_account_balance ) A
) B
ORDER BY 1,2;
```

Amazon Redshift는 단일 SQL 문의 SELECT 절에서 중첩된 창 함수를 지원하지 않으므로 두 개의 중첩된 하위 쿼리를 사용해야 합니다.

- 내부 하위 쿼리(별칭 A)에서 동적 파티션 표시기(dynamic_part)가 생성되고 채워집니다. 한 달의 잔액이 이전 달의 잔액보다 작거나 같으면 dynamic_part는 1로 설정되고, 그렇지 않으면 0으로 설정됩니다.
- 다음 레이어(별칭 B)에서는 SUM 창 함수의 결과로 new_dynamic_part 속성이 생성됩니다.
- 마지막으로 new_dynamic_part를 새 파티션 속성(dynamic partition)으로 기존 파티션 속성(account_id)에 추가하고 Teradata에서와 동일한 ROW_NUMBER() 창 함수를 적용합니다(그리고 1을 뺍니다).

이러한 변경 후 Amazon Redshift SQL은 Teradata와 동일한 출력을 생성합니다.

에픽

RESET WHEN을 Amazon Redshift SQL로 변환

작업	설명	필요한 기술
Teradata 창 함수를 생성합니다.	필요에 따라 중첩 집계 및 RESET WHEN 구문을 사용합니다.	SQL Developer
코드를 Amazon Redshift SQL로 변환합니다.	코드를 변환하려면 이 패턴의 '도구' 섹션에 있는 지침을 따르세요.	SQL Developer
Amazon Redshift에서 코드를 실행합니다.	테이블을 생성하고, 테이블에 데이터를 로드하고, Amazon Redshift에서 코드를 실행합니다.	SQL Developer

관련 리소스

참조

- [RESET WHEN 구문](#)(Teradata 설명서)
- [RESET WHEN 설명](#)(Stack Overflow)
- [Amazon Redshift로 마이그레이션](#)(AWS 웹 사이트)
- [AWS SCT 데이터 추출 에이전트를 사용하여 테라데이터 데이터베이스를 Amazon Redshift로 마이그레이션](#)(AWS 권장 가이드)
- [Teradata NORMALIZE 임시 기능을 Amazon Redshift SQL로 변환](#)(AWS 권장 가이드)

도구

- [AWS Schema Conversion Tool\(AWS SCT\)](#)

파트너

- [AWS 마이그레이션 컴피턴시 파트너](#)

시작 시 Amazon EMR 클러스터에 태그 지정 적용

작성자: Priyanka Chaudhary(AWS)

환경: 프로덕션	기술: 분석, 보안, 보안 인증 정보, 규정 준수	AWS 서비스: 아마존 EMR, AWS Lambda, 아마존 이벤트 CloudWatch
----------	-----------------------------	--

요약

이 패턴은 Amazon EMR 클러스터가 생성될 때 해당 클러스터에 태그가 지정되도록 하는 보안 제어를 제공합니다.

Amazon EMR은 방대한 양의 데이터를 처리하고 분석하기 위한 Amazon Web Services(AWS) 서비스입니다. Amazon EMR은 사내 클러스터 컴퓨팅 실행에 대한 손쉬운 대안으로 확장 가능하고 구성이 적은 서비스를 제공합니다. 태그를 사용하여 용도, 소유자 또는 환경을 기준으로 하는 것과 같이 AWS 리소스를 다양한 방식으로 분류할 수 있습니다. 예를 들어, 각 클러스터에 사용자 정의 메타데이터를 할당하여 Amazon EMR 클러스터에 태그를 지정할 수 있습니다. 태그는 사용자가 정의하는 키와 값으로 구성됩니다. 조직 요구 사항에 맞는 일관된 태그 집합을 생성하는 것이 좋습니다. 또한 태그를 Amazon EMR 클러스터에 추가하면 태그가 클러스터와 연결된 각각의 활성 Amazon Elastic Compute Cloud(Amazon EC2) 인스턴스로 전파됩니다. 마찬가지로, 태그를 Amazon EMR 클러스터에서 제거할 경우 해당 태그가 각각의 연결된 활성 EC2 인스턴스에서도 제거됩니다.

탐지 제어는 API 호출을 모니터링하고 [RunJobFlow](#), [AddTagsRemoveTags](#), 및 [CreateTagsAPI](#)에 대한 Amazon CloudWatch Events 이벤트를 시작합니다. 이벤트는 Python 스크립트를 실행하는 AWS Lambda를 호출합니다. Python 함수는 이벤트의 JSON 입력으로부터 Amazon EMR 클러스터 ID를 가져오고 다음 검사를 수행합니다.

- Amazon EMR 클러스터가 지정한 태그 이름으로 구성되어 있는지 확인합니다.
- 그렇지 않은 경우 Amazon EMR 클러스터 이름, 위반 세부 정보, AWS 리전, AWS 계정, 이 알림의 출처인 Lambda의 Amazon 리소스 이름(ARN)과 같은 관련 정보를 포함하여 Amazon Simple Notification Service(SNS) 알림을 사용자에게 보내세요.

사전 조건 및 제한 사항

사전 조건

- 활성 상태의 AWS 계정.
- 제공된 Lambda 코드를 업로드하기 위한 Amazon Simple Storage Service(S3) 버킷입니다. 또는 에픽 섹션에 설명된 대로 이 용도로 S3 버킷을 생성할 수 있습니다.
- 위반 알림을 받을 활성 이메일 주소
- 확인하려는 필수 태그 목록입니다.

제한 사항

- 이 보안 제어는 리전별로 적용됩니다. 모니터링하려는 각 AWS 리전에 배포해야 합니다.

제품 버전

- Amazon EMR 릴리스 4.8.0 이상에만 사용됩니다.

아키텍처

워크플로 아키텍처

자동화 및 규모 조정

- [AWS Organizations](#)를 사용하는 경우, [AWS StackSets Cloudformation](#)을 사용하여 모니터링하려는 여러 계정에 이 템플릿을 배포할 수 있습니다.

도구

서비스

- [AWS CloudFormation](#) — AWS는 AWS CloudFormation 리소스를 모델링 및 설정하고, 빠르고 일관되게 프로비저닝하고, 수명 주기 전반에 걸쳐 관리할 수 있도록 지원합니다. 템플릿을 사용하여 리소스와 해당 종속성을 설명하고, 리소스를 개별적으로 관리하는 대신 스택으로 함께 시작하고 구성할 수 있습니다. 여러 AWS 계정 및 AWS 리전에서 스택을 관리하고 프로비저닝할 수 있습니다.
- [Amazon CloudWatch Events](#) - Amazon CloudWatch Events는 AWS 리소스의 변경 사항을 설명하는 시스템 이벤트의 스트림을 거의 실시간으로 제공합니다.
- [Amazon EMR](#) - Amazon EMR은 빅 데이터 프레임워크 실행을 간소화하고 방대한 양의 데이터를 효율적으로 처리하는 웹 서비스입니다.

- [AWS Lambda](#) - AWS Lambda는 서버를 프로비저닝하거나 관리하지 않고도 코드를 실행할 수 있도록 지원하는 컴퓨팅 서비스입니다. Lambda는 필요 시에만 코드를 실행하며, 일일 몇 개의 요청에서 초당 수천 개의 요청까지 자동으로 규모를 조정합니다.
- [Amazon S3](#)-Amazon Simple Storage Service(S3)는 객체 스토리지 서비스입니다. Amazon S3를 사용하면 인터넷을 통해 언제 어디서든 원하는 양의 데이터를 저장하고 검색할 수 있습니다.
- [Amazon SNS](#) - Amazon Simple Notification Service(SNS)는 웹 서버와 이메일 주소를 포함하여 게시자와 클라이언트 간에 메시지를 전달 또는 전송하는 것을 조정하고 관리합니다. 구독자는 구독하는 주제에 게시된 모든 메시지를 수신하며 주제에 대한 모든 구독자는 동일한 메시지를 수신합니다.

코드

이 패턴에는 다음과 같은 첨부 파일이 포함됩니다.

- EMRTagValidation.zip - 보안 제어를 위한 Lambda 코드입니다.
- EMRTagValidation.yml— 이벤트 및 Lambda 함수를 설정하는 CloudFormation 템플릿.

에픽

S3 버킷 선택

작업	설명	필요한 기술
S3 버킷을 정의합니다.	Amazon S3 콘솔 에서 Lambda 코드 .zip 파일을 호스팅할 S3 버킷을 선택하거나 생성합니다. 이 S3 버킷은 모니터링하려는 Amazon EMR 클러스터와 동일한 AWS 리전에 있어야 합니다. Amazon S3 버킷 이름은 전역 수준에서 고유하며 네임스페이스는 모든 AWS 계정이 공유합니다. S3 버킷 이름에는 선행 슬래시를 포함할 수 없습니다.	클라우드 아키텍트
Lambda 코드를 업로드합니다.	첨부 파일 섹션에 제공된 Lambda 코드 .zip 파일을 S3	클라우드 아키텍트

작업	설명	필요한 기술
	버킷에 업로드합니다.	

AWS CloudFormation 템플릿 배포

작업	설명	필요한 기술
AWS CloudFormation 템플릿을 실행합니다.	S3 버킷과 동일한 AWS 리전에서 AWS CloudFormation 콘솔 을 열고 템플릿을 배포합니다. AWS CloudFormation 템플릿 배포에 대한 자세한 내용은 CloudFormation 설명서의 AWS CloudFormation 콘솔에 스택 생성 을 참조하십시오.	클라우드 아키텍트
템플릿에서 파라미터를 작성합니다.	<p>템플릿을 시작하면 다음 정보를 입력하라는 메시지가 표시됩니다.</p> <ul style="list-style-type: none"> S3 버킷: 첫 번째 에픽에서 생성하거나 선택한 버킷을 지정합니다. 첨부된 Lambda 코드(.zip 파일)를 업로드한 위치입니다. S3 키: S3 버킷에 있는 Lambda .zip 파일의 위치를 지정합니다(예: filename.zip 또는 controls/filename.zip). 선행 슬래시를 포함하지 마세요. 알림 이메일: Amazon SNS 알림을 받으려는 활성 이메일 주소를 입력합니다. 	클라우드 아키텍트

작업	설명	필요한 기술
	<ul style="list-style-type: none"> • 태깅 키 이름: 확인하려는 태그를 쉼표로 구분된 목록(예: ApplicationID , Environment , Owner)으로 입력합니다. CloudWatch Events 이벤트는 클러스터에서 이러한 태그를 모니터링하고 태그를 찾을 수 없는 경우 알림을 보냅니다. • Lamba 로깅 수준: Lambda 함수의 로깅 수준 및 빈도를 지정합니다. 정보를 사용하여 진행 상황에 대한 자세한 정보 메시지를 기록하고, 배포를 계속할 수 있는 오류 이벤트의 경우 오류를 기록하고, 잠재적으로 유해한 상황에 대한 경고를 기록할 수 있습니다. 	

구독 확인

작업	설명	필요한 기술
<p>구독을 확인합니다.</p>	<p>CloudFormation 템플릿이 성공적으로 배포되면 입력한 이메일 주소로 구독 이메일이 전송됩니다. 위반 알림을 받기 시작하려면 이 이메일 구독을 확인해야 합니다.</p>	<p>클라우드 아키텍트</p>

관련 리소스

- [AWS Lambda 개발자 안내서](#)
- [Amazon EMR에서 클러스터에 태그 지정](#)

첨부

이 문서와 관련된 추가 콘텐츠에 액세스하려면 [attachment.zip](#) 파일의 압축을 풉니다.

시작 시 Amazon S3에 Amazon EMR 로깅이 활성화되었는지 확인

작성자: Priyanka Chaudhary(AWS)

환경: 프로덕션

기술: 보안, 자격 증명, 규정 준수; 서버리스; 분석

워크로드: 오픈 소스

AWS 서비스: 아마존 EMR, 아마존 S3, 아마존 SNS, 아마존 CloudWatch

요약

이 패턴은 Amazon Web Services(AWS)에서 실행되는 Amazon EMR 클러스터의 로깅 구성을 모니터링하는 보안 제어를 제공합니다.

Amazon EMR은 빅 데이터 처리 및 분석을 위한 AWS 도구입니다. Amazon EMR은 사내 클러스터 컴퓨팅 실행의 대안으로 확장 가능한 저구성 서비스를 제공합니다. Amazon EMR은 두 가지 유형의 EMR 클러스터를 제공합니다.

- **임시 Amazon EMR 클러스터:** 임시 Amazon EMR 클러스터는 처리가 완료되면 자동으로 종료되고 비용 발생을 중단합니다.
- **영구 Amazon EMR 클러스터:** 영구 Amazon EMR 클러스터는 데이터 처리 작업이 완료된 후에도 계속 실행됩니다.

Amazon EMR와 Hadoop은 모두 클러스터에서 상태를 보고하는 로그 파일을 생성합니다. 기본적으로 이러한 파일은 `/mnt/var/log/` 디렉터리의 프라이머리 노드에 기록됩니다. 클러스터를 시작할 때 클러스터를 구성하는 방법에 따라, Amazon Simple Storage Service(S3)에 로그를 저장하고 그래픽 디버깅 도구를 통해 볼 수도 있습니다. Amazon S3 로깅은 클러스터가 시작될 때만 지정할 수 있다는 점에 유의하십시오. 이 구성을 사용하면 5분마다 프라이머리 노드에서 Amazon S3 위치로 로그가 전송됩니다. 임시 클러스터의 경우, 처리가 완료되면 클러스터가 사라지고 이러한 로그 파일을 사용하여 실패한 작업을 디버깅할 수 있으므로 Amazon S3 로깅이 중요합니다.

이 패턴은 AWS CloudFormation 템플릿을 사용하여 API 호출을 모니터링하고 ""에서 Amazon CloudWatch Events를 시작하는 보안 제어를 RunJobFlow 배포합니다. 이 트리거는 Python 스크립트를 실행하는 AWS Lambda를 호출합니다. Lambda 함수는 이벤트 JSON 입력에서 EMR 클러스터 ID를

검색하고 Amazon S3 로그 URI도 확인합니다. Amazon S3 URI를 찾을 수 없는 경우, Lambda 함수는 EMR 클러스터 이름, 위반 세부 정보, AWS 리전, AWS 계정 및 알림의 출처가 되는 Lambda Amazon 리소스 이름(ARN)을 자세히 설명하는 Amazon Simple Notification Service(SNS) 알림을 보냅니다.

사전 조건 및 제한 사항

사전 조건

- 활성 상태의 AWS 계정
- Lambda 코드 .zip 파일을 위한 S3 버킷
- 위반 알림을 수신하려는 이메일 주소

제한 사항

- 이 보안 제어는 리전과 관련이 있으므로 모니터링하고자 하는 AWS 리전에 배포해야 합니다.

제품 버전

- Amazon EMR 릴리스 4.8.0 이상에만 사용됩니다.

아키텍처

대상 기술 스택

- 아마존 CloudWatch 이벤트 이벤트
- Amazon EMR
- Lambda 함수
- S3 버킷
- Amazon SNS

대상 아키텍처

자동화 및 규모 조정

- AWS Organizations를 사용하는 경우, [CloudFormation StackSetsAWS](#)를 사용하여 모니터링하려는 여러 계정에 이 템플릿을 배포할 수 있습니다.

도구

도구

- [AWS CloudFormation](#) — AWS는 인프라를 코드로 사용하여 AWS 리소스를 모델링하고 설정할 수 CloudFormation 있도록 지원합니다.
- [AWS Cloudwatch Events](#) — AWS CloudWatch Events는 AWS 리소스의 변경 사항을 설명하는 시스템 이벤트의 스트림을 거의 실시간으로 제공합니다.
- [Amazon EMR](#) - Amazon EMR은 빅 데이터 프레임워크 실행을 간소화하는 관리형 클러스터 플랫폼입니다.
- [AWS Lambda](#) - AWS Lambda는 서버를 프로비저닝하거나 관리하지 않고도 코드를 실행할 수 있습니다. Lambda는 필요 시에만 코드를 실행하며, 일일 몇 개의 요청에서 초당 수천 개의 요청까지 자동으로 규모를 조정합니다.
- [Amazon S3](#) - Amazon S3에서 언제든지 웹상 어디서나 원하는 양의 데이터를 저장하고 검색할 수 있는 웹 서비스 인터페이스입니다.
- [Amazon SNS](#) - Amazon SNS는 게시자와 클라이언트 간에 웹 서버와 이메일 주소를 포함한 메시지 전달 또는 전송을 조정하고 관리하는 웹 서비스입니다.

코드

- 프로젝트의 .zip 파일은 첨부 파일로 제공됩니다.

에픽

S3 버킷 정의

작업	설명	필요한 기술
S3 버킷을 정의합니다.	Lambda 코드 .zip 파일을 호스트하려면, 앞에 슬래시가 없는 고유한 이름을 가진 S3 버	클라우드 아키텍트

작업	설명	필요한 기술
	킷을 선택하거나 생성합니다. S3 버킷 이름은 전역 수준에서 고유하며, 네임스페이스는 모든 AWS 계정이 공유합니다. S3 버킷은 평가 중인 Amazon EMR 클러스터와 동일한 AWS 리전에 있어야 합니다.	

Lambda 코드를 S3 버킷에 업로드

작업	설명	필요한 기술
Lambda 코드를 S3 버킷에 업로드	“첨부 파일” 섹션에 나와 있는 Lambda 코드 .zip 파일을 S3 버킷에 업로드합니다. S3 버킷은 평가 중인 Amazon EMR 클러스터와 동일한 AWS 리전에 있어야 합니다.	클라우드 아키텍트

AWS CloudFormation 템플릿 배포

작업	설명	필요한 기술
AWS CloudFormation 템플릿을 배포합니다.	AWS CloudFormation 콘솔의 S3 버킷과 동일한 리전에 이 패턴의 첨부 파일로 제공된 AWS CloudFormation 템플릿을 배포합니다. 다음 에픽에서 파라미터에 대한 값을 입력합니다. AWS CloudFormation 템플릿 배포에 대한 자세한 내용은 “관련 리소스” 섹션을 참조하십시오.	클라우드 아키텍트

AWS CloudFormation 템플릿에서 파라미터 완성

작업	설명	필요한 기술
S3 버킷에 이름을 지정합니다.	첫 번째 에픽에서 생성한 S3 버킷의 이름을 입력합니다.	클라우드 아키텍트
Amazon S3 키를 입력합니다.	S3 버킷에 있는 Lambda 코드 .zip 파일의 위치를 앞에 슬래시 없이 입력합니다(예: <directory>/<file-name>.zip).	클라우드 아키텍트
이메일 주소를 입력합니다.	Amazon SNS 알림을 수신할 활성 이메일 주소를 입력합니다.	클라우드 아키텍트
로깅 수준을 정의합니다.	Lambda 함수의 로깅 수준 및 빈도를 정의합니다. '정보'는 애플리케이션 진행 상황에 대한 자세한 정보 메시지를 의미합니다. '오류'는 여전히 애플리케이션을 계속 실행할 수 있는 오류 이벤트를 의미합니다. '경고'는 잠재적으로 위험한 상황을 의미합니다.	클라우드 아키텍트

구독 확인

작업	설명	필요한 기술
구독을 확인합니다.	템플릿이 성공적으로 배포되면 입력한 이메일 주소로 구독 이메일 메시지가 전송됩니다. 위반 알림을 받으려면 이 이메일 구독을 확인해야 합니다.	클라우드 아키텍트

관련 리소스

[AWS Lambda](#)

[Amazon EMR 로깅](#)

[AWS CloudFormation 템플릿 배포](#)

첨부

이 문서와 관련된 추가 콘텐츠에 액세스하려면 [attachment.zip](#) 파일의 압축을 풉니다.

AWS Glue 작업과 Python을 사용하여 테스트 데이터 생성

작성자: Moinul Al-Mamun

환경: 프로덕션

기술: 분석, 클라우드 네이티브, 데이터 레이크, 소프트웨어 개발 및 테스트, 서버리스, 빅 데이터

AWS 서비스: AWS Glue, Amazon S3

요약

이 패턴은 Python으로 작성된 AWS Glue 작업을 생성하여 수백만 개의 샘플 파일을 동시에 빠르고 쉽게 생성하는 방법을 보여줍니다. 샘플 파일은 Amazon Simple Storage Service(S3) 버킷에 저장됩니다. AWS 클라우드에서 서비스를 테스트하거나 평가하려면 대량의 샘플 파일을 빠르게 생성할 수 있는 능력이 중요합니다. 예를 들어 Amazon S3 접두사에 있는 수백만 개의 작은 파일에 대한 데이터 분석을 수행하여 AWS Glue Studio 또는 AWS Glue DataBrew 작업의 성능을 테스트할 수 있습니다.

다른 AWS 서비스를 사용하여 샘플 데이터 세트를 생성할 수도 있지만 AWS Glue를 사용하는 것이 좋습니다. AWS Glue는 서버리스 데이터 처리 서비스이므로 인프라를 관리할 필요가 없습니다. 코드를 가져와서 AWS Glue 클러스터에서 실행하기만 하면 됩니다. 또한 AWS Glue는 작업 실행에 필요한 리소스를 프로비저닝, 구성 및 확장합니다. 사용하는 리소스에 대해서만 비용을 지불합니다.

사전 조건 및 제한 사항

사전 조건

- 활성 상태의 AWS 계정.
- AWS 계정과 작동하도록 [설치](#) 및 [구성](#)된 AWS Command Line Interface(AWS CLI)

제품 버전

- Python 3.9
- CLI version 2

제한 사항

트리거당 최대 AWS Glue 작업 수는 50개입니다. 자세한 내용은 [AWS Glue 엔드포인트 및 할당량을 참조](#)하세요.

아키텍처

다음 다이어그램은 출력(즉, 샘플 파일)을 S3 버킷에 쓰는 AWS Glue 작업을 중심으로 한 예제 아키텍처를 보여줍니다.

이 다이어그램은 다음 워크플로우를 포함합니다.

1. AWS CLI, AWS Management Console 또는 API를 사용하여 AWS Glue 작업을 시작합니다. AWS CLI 또는 API를 사용하면 호출된 작업의 병렬화를 자동화하고 샘플 파일 생성 런타임을 줄일 수 있습니다.
2. AWS Glue 작업은 파일 콘텐츠를 무작위로 생성하고, 콘텐츠를 CSV 형식으로 변환한 다음, 콘텐츠를 공통 접두사 아래의 Amazon S3 객체로 저장합니다. 각 파일은 1킬로바이트 미만입니다. AWS Glue 작업은 두 개의 사용자 정의 작업 파라미터인 START_RANGE 및 END_RANGE를 허용합니다. 이러한 파라미터를 사용하여 각 작업 실행에 의해 Amazon S3에서 생성되는 파일 이름과 파일 수를 설정할 수 있습니다. 이 작업의 여러 인스턴스를 병렬로 실행할 수 있습니다(예: 인스턴스 100개).

도구

- [Amazon Simple Storage Service\(S3\)](#)는 원하는 양의 데이터를 저장, 보호 및 검색하는 데 도움이 되는 클라우드 기반 객체 스토리지 서비스입니다.
- [AWS Command Line Interface \(AWS CLI\)](#)는 명령줄 셸에서 명령을 사용하여 AWS 서비스와 상호 작용할 수 있는 오픈 소스 도구입니다.
- [AWS Glue](#)는 완전 관리형 추출, 변환, 적재(ETL) 서비스입니다. 이를 통해 데이터 스토어와 데이터 스트림 간에 데이터를 안정적으로 분류, 정리, 보강하고 이동할 수 있습니다.
- [AWS Identity and Access Management\(IAM\)](#)는 사용자에 대한 인증 및 권한 부여를 제어함으로써 AWS 리소스에 대한 액세스를 안전하게 제어할 수 있습니다.

모범 사례

이 패턴을 구현할 때는 다음과 같은 AWS Glue 모범 사례를 고려하세요.

- 적절한 AWS Glue 작업자 유형을 사용하여 비용을 절감합니다. 작업자 유형의 다양한 속성을 이해한 다음 CPU 및 메모리 요구 사항에 따라 워크로드에 적합한 작업자 유형을 선택하는 것이 좋습니다. 이 패턴의 경우 DPU를 최소화하고 비용을 줄이려면 Python 셸 작업을 작업 유형으로 사용하는 것이 좋습니다. 자세한 내용은 AWS Glue 개발자 가이드의 [AWS Glue에서 작업 추가](#)를 참조하세요.
- 적절한 동시성 한도를 사용하여 작업을 확장합니다. 시간 요구 사항과 필요한 파일 수를 기준으로 AWS Glue 작업의 최대 동시 실행 시간을 정하는 것이 좋습니다.
- 처음에는 적은 수의 파일을 생성하기 시작합니다. AWS Glue 작업을 구축할 때 비용을 줄이고 시간을 절약하려면 적은 수의 파일(예: 1,000개)부터 시작합니다. 이렇게 하면 문제 해결이 더 쉬워질 수 있습니다. 적은 수의 파일을 생성하는 데 성공하면 더 많은 파일로 확장할 수 있습니다.
- 먼저 로컬에서 실행합니다. AWS Glue 작업을 구축할 때 비용을 줄이고 시간을 절약하려면 로컬에서 개발을 시작하고 코드를 테스트합니다. 셸과 통합 개발 환경(IDE)에서 AWS Glue 추출, 전환, 적재(ETL) 작업을 작성하는 데 도움이 되는 Docker 컨테이너를 설정하는 방법에 대한 지침은 AWS 빅데이터 블로그의 [컨테이너를 사용하여 로컬에서 AWS Glue ETL 작업 개발](#)을 참조하세요.

더 많은 AWS Glue 모범 사례는 AWS Glue 설명서의 [모범 사례](#)를 참조하세요.

에픽

대상 S3 버킷 및 IAM 역할 생성

작업	설명	필요한 기술
파일을 저장할 S3 버킷을 만듭니다.	<p>S3 버킷과 그 안에 접두사를 생성합니다.</p> <p>참고: 이 패턴은 데모용으로 <code>s3://{your-s3-bucket-name}/small-files/</code> 위치를 사용합니다.</p>	앱 개발자
IAM 역할 생성 및 구성.	<p>AWS Glue 작업에서 S3 버킷에 쓰는 데 사용할 수 있는 IAM 역할을 생성해야 합니다.</p> <p>1. IAM 역할(예: "AWSGlueServiceRole-smallfi</p>	앱 개발자

작업	설명	필요한 기술
	<p>les" 로 부름)을 생성합니다.</p> <p>2. 정책의 신뢰할 수 있는 엔티티로 AWS Glue를 선택합니다.</p> <p>3. AWS 관리형 정책을 "AWSGlueServiceRole" 라고 부르는 역할에 연결합니다.</p> <p>4. 다음 구성을 기반으로 "s3-small-file-access" 라고 부르는 인라인 정책 또는 고객 관리형 정책을 생성합니다. "{bucket}" 을 버킷의 이름으로 바꿉니다.</p> <pre data-bbox="630 1031 1029 1879"> { "Version": "2012-10-17", "Statement": [{ "Effect": "Allow", "Action": ["s3:GetObject", "s3:PutObject"], "Resource": ["arn:aws:s3:::{bucket}/small-files/input/*"] }] } </pre>	

작업	설명	필요한 기술
	<pre> }] } </pre> <p>5. 역할에 "s3-small-file-access" 정책을 연결합니다.</p>	

동시 실행을 처리하도록 AWS Glue 작업을 생성하고 구성합니다.

작업	설명	필요한 기술
<p>AWS Glue 작업을 생성합니다.</p>	<p>콘텐츠를 생성하여 S3 버킷에 저장하는 AWS Glue 작업을 생성해야 합니다.</p> <p>AWS Glue 작업을 생성한 후 다음 단계를 완료하여 작업을 구성합니다.</p> <ol style="list-style-type: none"> 1. AWS Management Console에 로그인하고 AWS Glue 콘솔을 엽니다. 2. 탐색 창의 데이터 통합 및 ETL에서 작업을 선택합니다. 3. 작업 생성 섹션에서 Python 쉘 스크립트 편집기를 선택합니다. 4. 옵션 섹션에서 보일러플레이트 코드로 새 스크립트 만들기를 선택한 다음 만들기를 선택합니다. 5. 세부 정보 보기를 선택합니다. 	<p>앱 개발자</p>

작업	설명	필요한 기술
	<ol style="list-style-type: none"> 6. 이름에 <code>create_small_files</code>를 입력합니다. 7. IAM 역할에서, 이전에 만든 IAM 역할을 선택합니다. 8. 이 작업 실행 섹션에서 직접 작성할 새 스크립트를 선택합니다. 9. 고급 속성을 선택합니다. 10. 최대 동시성에 대해서는 데모용으로 100을 입력합니다. 참고: 최대 동시성은 병렬로 실행할 수 있는 작업 인스턴스 수를 정의합니다. 11. 저장을 선택합니다. 	

작업	설명	필요한 기술
작업 코드를 업데이트합니다.	<ol style="list-style-type: none"> 1. AWS Glue 콘솔을 엽니다. 2. 탐색 창에서, 작업을 선택합니다. 3. 내 작업 섹션에서 이전에 생성한 작업을 선택합니다. 4. 스크립트 탭을 선택한 다음 아래의 코드에 따라 스크립트를 업데이트합니다. BUCKET_NAME , PREFIX, text_str 변수를 원하는 값으로 업데이트합니다. <pre data-bbox="630 806 1029 1852"> from awsglue.utils import getResolvedOptions import sys import boto3 from random import randrange # Two arguments args = getResolvedOptions(sys.argv , ['START_RANGE', 'END_RANGE']) START_RANGE = int(args['START_RANGE']) END_RANGE = int(args['END_RANGE']) BUCKET_NAME = '{BUCKET_NAME}' PREFIX = 'small-files/input/' s3 = boto3.res ource('s3') </pre>	앱 개발자

작업	설명	필요한 기술
	<pre> for x in range(STA RT_RANGE, END_RANGE): # generate file name file_name = f"input_{x}.txt" # generate text text_str = str(randrange(1000 00))+","+str(randr ange(100000))+", " + str(randrange(1000 0000)) + "," + str(randrange(1000 0)) # write in s3 s3.Object(BUCKE T_NAME, PREFIX + file_name).put(Bod y=text_str) </pre> <p>5. 저장을 선택합니다.</p>	

명령줄 또는 콘솔에서 AWS Glue 작업을 실행합니다.

작업	설명	필요한 기술
<p>명령줄에서 AWS Glue 작업을 실행합니다.</p>	<p>AWS CLI에서 AWS Glue 작업을 실행하려면 해당 값을 사용하여 다음 명령을 실행합니다.</p> <pre> cmd:~\$ aws glue start- job-run --job-name create_small_files --arguments '{"--STAR T_RANGE":"0","--EN D_RANGE":"1000000"}' </pre>	<p>앱 개발자</p>

작업	설명	필요한 기술
	<pre data-bbox="610 212 1011 506">cmd:~\$ aws glue start- job-run --job-name create_small_files --arguments '{"--STAR T_RANGE":"1000000" , "--END_RANGE":"20 00000"}'</pre> <p data-bbox="594 541 1024 863">참고: AWS Management Console에서 AWS Glue 작업을 실행하는 방법에 대한 지침은 이 패턴의 AWS Management Console에서 AWS Glue 작업 실행 스토리를 참조하세요.</p> <p data-bbox="594 909 1024 1136">팁: 위의 예와 같이 다른 파라미터로 한 번에 여러 실행을 실행하려면 AWS CLI를 사용하여 AWS Glue 작업을 실행하는 것이 좋습니다.</p> <p data-bbox="594 1182 1024 1451">특정 병렬화 요소를 사용하여 정의된 수의 파일을 생성하는데 필요한 모든 AWS CLI 명령을 생성하려면 해당 값을 사용하여 다음 bash 코드를 실행합니다.</p> <pre data-bbox="610 1507 1011 1854"># define parameters NUMBER_OF_FILES= 10000000; PARALLELIZATION=50; # initialize _SB=0; # generate commands</pre>	

작업	설명	필요한 기술
	<pre data-bbox="592 210 1031 892"> for i in \$(seq 1 \$PARALLELIZATION); do echo aws glue start-job-run -- job-name create_sm all_files --argumen ts ""'{"--START_RANG E":"'\${(((NUMBER_OF _FILES/PARALLELIZA TION) * (i-1) + _SB))}'", "--END_RAN GE":"'\${(((NUMBER_O F_FILES/PARALLELIZ ATION) * (i)))}'"}'"""; _SB=1; done </pre> <p data-bbox="592 934 1031 1018">위 스크립트를 사용하는 경우 다음 사항을 고려합니다.</p> <ul data-bbox="592 1060 1031 1795" style="list-style-type: none"> • 이 스크립트는 소규모 파일을 대규모로 호출 및 생성하는 것을 단순화합니다. • 해당 값을 사용하여 NUMBER_OF_FILES 및 PARALLELIZATION 을 업데이트합니다. • 위 스크립트는 실행해야 하는 명령 목록을 인쇄합니다. 해당 출력 명령을 복사한 다음 터미널에서 실행합니다. • 스크립트 내에서 명령을 직접 실행하려면 11번째 줄에서 echo 명령문을 제거합니다. 	

작업	설명	필요한 기술
	<p>참고: 위 스크립트의 출력 예를 보려면 이 패턴의 추가 정보 섹션에서 셸 스크립트 출력을 참조하세요.</p>	
<p>AWS Management Console에서 AWS Glue 작업을 실행합니다.</p>	<ol style="list-style-type: none"> 1. AWS Management Console에 로그인하고 AWS Glue 콘솔을 엽니다. 2. 탐색 창의 데이터 통합 및 ETL에서 작업을 선택합니다. 3. 내 작업 섹션에서 작업을 선택합니다. 4. 파라미터 (선택 사항) 섹션에서 파라미터를 업데이트합니다. 5. 작업을 선택한 후 작업 실행을 선택합니다. 6. 필요한 만큼 3에서 5단계를 반복합니다. 예를 들어, 천만 개의 파일을 만들려면 이 프로세스를 10번 반복합니다. 	<p>앱 개발자</p>

작업	설명	필요한 기술
AWS Glue 작업의 상태를 확인합니다.	<ol style="list-style-type: none"> AWS Glue 콘솔을 엽니다. 탐색 창에서, 작업을 선택합니다. 내 작업 섹션에서 이전에 생성한 작업(즉, <code>create_small_files</code>)을 선택합니다. 파일의 진행 상황 및 생성을 자세히 알아보려면 실행 ID, 실행 상태 및 기타 열을 검토합니다. 	앱 개발자

관련 리소스

참조

- [Registry of Open Data on AWS](#)
- [분석용 데이터 세트](#)
- [AWS 기반 오픈 데이터](#)
- [AWS Glue에 작업 추가](#)
- [AWS Glue 시작하기](#)

가이드 및 패턴

- [AWS Glue 모범 사례](#)
- [부하 테스트 애플리케이션](#)

추가 정보

벤치마킹 테스트

이 패턴은 벤치마킹 테스트의 일환으로 다양한 병렬화 파라미터를 사용하여 1천만 개의 파일을 생성하는 데 사용되었습니다. 다음 표는 테스트 결과를 보여줍니다.

병렬화	작업 실행으로 생성된 파일 수	작업 기간	Speed(속도)
10	1,000,000	6시간 40분	매우 느림
50	200,000	80분	보통
100	100,000건	40분	빠른

프로세스를 더 빠르게 진행하려면 작업 구성에서 더 많은 동시 실행을 구성할 수 있습니다. 요구 사항에 따라 작업 구성을 쉽게 조정할 수 있지만, AWS Glue 서비스 할당량 한도가 있다는 점을 기억합니다. 자세한 내용은 [AWS Glue 엔드포인트 및 할당량](#)을 참조하세요.

셸 스크립트 출력

다음 예제는 이 패턴의 명령줄에서 AWS Glue 작업 실행 스토리의 셸 스크립트 출력을 보여줍니다.

```
user@MUC-1234567890 MINGW64 ~
$ # define parameters
NUMBER_OF_FILES=10000000;
PARALLELIZATION=50;
# initialize
_SB=0;

# generate commands
for i in $(seq 1 $PARALLELIZATION);
do
    echo aws glue start-job-run --job-name create_small_files --arguments
    ""'{"--START_RANGE":"'${((NUMBER_OF_FILES/PARALLELIZATION) (i-1) + SB)}'","--
ENDRANGE":"'${((NUMBER_OF_FILES/PARALLELIZATION) (i))}'"}'""';
    _SB=1;
done

aws glue start-job-run --job-name create_small_files --arguments '{"--
START_RANGE":"0","--END_RANGE":"200000"}'
aws glue start-job-run --job-name create_small_files --arguments '{"--
START_RANGE":"200001","--END_RANGE":"400000"}'
aws glue start-job-run --job-name create_small_files --arguments '{"--
START_RANGE":"400001","--END_RANGE":"600000"}'
aws glue start-job-run --job-name create_small_files --arguments '{"--
START_RANGE":"600001","--END_RANGE":"800000"}'
```

```
aws glue start-job-run --job-name create_small_files --arguments '{"--START_RANGE":"800001","--END_RANGE":"1000000"}'
```

```
aws glue start-job-run --job-name create_small_files --arguments '{"--START_RANGE":"1000001","--END_RANGE":"1200000"}'
```

```
aws glue start-job-run --job-name create_small_files --arguments '{"--START_RANGE":"1200001","--END_RANGE":"1400000"}'
```

```
aws glue start-job-run --job-name create_small_files --arguments '{"--START_RANGE":"1400001","--END_RANGE":"1600000"}'
```

```
aws glue start-job-run --job-name create_small_files --arguments '{"--START_RANGE":"1600001","--END_RANGE":"1800000"}'
```

```
aws glue start-job-run --job-name create_small_files --arguments '{"--START_RANGE":"1800001","--END_RANGE":"2000000"}'
```

```
aws glue start-job-run --job-name create_small_files --arguments '{"--START_RANGE":"2000001","--END_RANGE":"2200000"}'
```

```
aws glue start-job-run --job-name create_small_files --arguments '{"--START_RANGE":"2200001","--END_RANGE":"2400000"}'
```

```
aws glue start-job-run --job-name create_small_files --arguments '{"--START_RANGE":"2400001","--END_RANGE":"2600000"}'
```

```
aws glue start-job-run --job-name create_small_files --arguments '{"--START_RANGE":"2600001","--END_RANGE":"2800000"}'
```

```
aws glue start-job-run --job-name create_small_files --arguments '{"--START_RANGE":"2800001","--END_RANGE":"3000000"}'
```

```
aws glue start-job-run --job-name create_small_files --arguments '{"--START_RANGE":"3000001","--END_RANGE":"3200000"}'
```

```
aws glue start-job-run --job-name create_small_files --arguments '{"--START_RANGE":"3200001","--END_RANGE":"3400000"}'
```

```
aws glue start-job-run --job-name create_small_files --arguments '{"--START_RANGE":"3400001","--END_RANGE":"3600000"}'
```

```
aws glue start-job-run --job-name create_small_files --arguments '{"--START_RANGE":"3600001","--END_RANGE":"3800000"}'
```

```
aws glue start-job-run --job-name create_small_files --arguments '{"--START_RANGE":"3800001","--END_RANGE":"4000000"}'
```

```
aws glue start-job-run --job-name create_small_files --arguments '{"--START_RANGE":"4000001","--END_RANGE":"4200000"}'
```

```
aws glue start-job-run --job-name create_small_files --arguments '{"--START_RANGE":"4200001","--END_RANGE":"4400000"}'
```

```
aws glue start-job-run --job-name create_small_files --arguments '{"--START_RANGE":"4400001","--END_RANGE":"4600000"}'
```

```
aws glue start-job-run --job-name create_small_files --arguments '{"--START_RANGE":"4600001","--END_RANGE":"4800000"}'
```

```
aws glue start-job-run --job-name create_small_files --arguments '{"--START_RANGE":"4800001","--END_RANGE":"5000000"}'
```

```
aws glue start-job-run --job-name create_small_files --arguments '{"--START_RANGE":"5000001","--END_RANGE":"5200000"}'
```

```
aws glue start-job-run --job-name create_small_files --arguments '{"--START_RANGE":"5200001","--END_RANGE":"5400000"}'
```

```
aws glue start-job-run --job-name create_small_files --arguments '{"--START_RANGE":"5400001","--END_RANGE":"5600000"}'
```

```
aws glue start-job-run --job-name create_small_files --arguments '{"--START_RANGE":"5600001","--END_RANGE":"5800000"}'
```

```
aws glue start-job-run --job-name create_small_files --arguments '{"--START_RANGE":"5800001","--END_RANGE":"6000000"}'
```

```
aws glue start-job-run --job-name create_small_files --arguments '{"--START_RANGE":"6000001","--END_RANGE":"6200000"}'
```

```
aws glue start-job-run --job-name create_small_files --arguments '{"--START_RANGE":"6200001","--END_RANGE":"6400000"}'
```

```
aws glue start-job-run --job-name create_small_files --arguments '{"--START_RANGE":"6400001","--END_RANGE":"6600000"}'
```

```
aws glue start-job-run --job-name create_small_files --arguments '{"--START_RANGE":"6600001","--END_RANGE":"6800000"}'
```

```
aws glue start-job-run --job-name create_small_files --arguments '{"--START_RANGE":"6800001","--END_RANGE":"7000000"}'
```

```
aws glue start-job-run --job-name create_small_files --arguments '{"--START_RANGE":"7000001","--END_RANGE":"7200000"}'
```

```
aws glue start-job-run --job-name create_small_files --arguments '{"--START_RANGE":"7200001","--END_RANGE":"7400000"}'
```

```
aws glue start-job-run --job-name create_small_files --arguments '{"--START_RANGE":"7400001","--END_RANGE":"7600000"}'
```

```
aws glue start-job-run --job-name create_small_files --arguments '{"--START_RANGE":"7600001","--END_RANGE":"7800000"}'
```

```
aws glue start-job-run --job-name create_small_files --arguments '{"--START_RANGE":"7800001","--END_RANGE":"8000000"}'
```

```
aws glue start-job-run --job-name create_small_files --arguments '{"--START_RANGE":"8000001","--END_RANGE":"8200000"}'
```

```
aws glue start-job-run --job-name create_small_files --arguments '{"--START_RANGE":"8200001","--END_RANGE":"8400000"}'
```

```
aws glue start-job-run --job-name create_small_files --arguments '{"--START_RANGE":"8400001","--END_RANGE":"8600000"}'
```

```
aws glue start-job-run --job-name create_small_files --arguments '{"--START_RANGE":"8600001","--END_RANGE":"8800000"}'
```

```
aws glue start-job-run --job-name create_small_files --arguments '{"--START_RANGE":"8800001","--END_RANGE":"9000000"}'
```

```
aws glue start-job-run --job-name create_small_files --arguments '{"--START_RANGE":"9000001","--END_RANGE":"9200000"}'
```

```
aws glue start-job-run --job-name create_small_files --arguments '{"--START_RANGE":"9200001","--END_RANGE":"9400000"}'
```

```
aws glue start-job-run --job-name create_small_files --arguments '{"--START_RANGE":"9400001","--END_RANGE":"9600000"}'
```

```
aws glue start-job-run --job-name create_small_files --arguments '{"--START_RANGE":"9600001","--END_RANGE":"9800000"}'  
aws glue start-job-run --job-name create_small_files --arguments '{"--START_RANGE":"9800001","--END_RANGE":"10000000"}'
```

```
user@MUC-1234567890 MINGW64 ~
```

FAQ

동시 실행 또는 병렬 작업은 몇 개나 사용해야 할까요?

동시 실행 및 병렬 작업 수는 필요한 시간 및 원하는 테스트 파일 수에 따라 달라집니다. 만들고 있는 파일의 크기를 확인하는 것이 좋습니다. 먼저, AWS Glue 작업에서 원하는 수의 파일을 생성하는 데 걸리는 시간을 확인합니다. 그런 다음, 적절한 수의 동시 실행을 사용하여 목표를 달성합니다. 예를 들어 100,000개의 파일이 실행을 완료하는 데 40분이 걸리지만 목표 시간이 30분이라고 가정하면 AWS Glue 작업의 동시성 설정을 늘려야 합니다.

이 패턴을 사용하여 어떤 유형의 콘텐츠를 생성할 수 있나요?

구분자가 다른 텍스트 파일(예: PIPE, JSON 또는 CS)과 같은 모든 유형의 콘텐츠를 만들 수 있습니다. 이 패턴은 Boto3를 사용하여 파일에 쓴 다음 파일을 S3 버킷에 저장합니다.

S3 버킷에 필요한 IAM 권한 수준은 어느 정도인가요?

S3 버킷에 있는 객체에 대한 Write 액세스를 허용하는 ID 기반 정책이 있어야 합니다. 자세한 내용은 Amazon S3 설명서의 [Amazon S3: S3 버킷에 있는 객체에 대한 읽기 및 쓰기 액세스 권한 허용](#)을 참조하세요.

Lambda 함수를 사용하여 임시 EMR 클러스터에서 Spark 작업 시작

작성자: Dhruvajyoti Mukherjee(AWS)

환경: 프로덕션

기술: 분석

워크로드: 오픈 소스

AWS 서비스: Amazon
DynamoDB, Identity and
Access Management, Lambda

요약

이 패턴은 Amazon EMR RunJobFlow API 작업을 사용하여 Lambda 함수에서 Spark 작업을 실행하는 임시 클러스터를 시작합니다. 임시 EMR 클러스터는 작업이 완료되거나 오류가 발생하는 즉시 종료되도록 설계되었습니다. 임시 클러스터는 계산 시간에만 실행되므로 비용을 절감할 수 있으며 클라우드 환경에서 확장성과 유연성을 제공합니다.

임시 EMR 클러스터는 Lambda 함수에서 Boto3 API와 Python 프로그래밍 언어를 사용하여 시작됩니다. Python으로 작성된 Lambda 함수는 필요할 때 클러스터를 시작하는 추가 유연성을 제공합니다.

샘플 배치 계산 및 출력을 시연하기 위해 이 패턴은 Lambda 함수의 EMR 클러스터에서 Spark 작업을 시작하고 가상 회사의 예제 판매 데이터에 대해 배치 계산을 실행합니다. Spark 작업의 출력은 Amazon Simple Storage Service(S3)에서 쉼표로 구분된 값(CSV) 파일입니다. 입력 데이터 파일, Spark.jar 파일, 코드 스니펫, 계산을 실행하기 위한 가상 사설 클라우드 (VPC) 및 AWS Identity 및 Access Management (IAM) 역할에 대한 AWS CloudFormation 템플릿이 첨부 파일로 제공됩니다.

사전 조건 및 제한 사항

사전 조건

- 활성 상태의 AWS 계정

제한 사항

- 코드에서는 한 번에 하나의 Spark 작업만 시작할 수 있습니다.

제품 버전

- Amazon EMR 6.0.0에서 테스트됨

아키텍처

대상 기술 스택

- Amazon EMR
- AWS Lambda
- Amazon S3
- Apache Spark

대상 아키텍처

자동화 및 규모 조정

Spark-EMR 배치 계산을 자동화하려면 다음 옵션 중 하나를 사용할 수 있습니다.

- 크론 스케줄에서 Lambda 함수를 시작할 수 있는 Amazon EventBridge 규칙을 구현하십시오. 자세한 내용은 [자습서: 를 사용하여 AWS Lambda 함수 일정 잡기를](#) 참조하십시오. EventBridge
- 파일 도착 시 Lambda 함수를 시작하도록 [Amazon S3 이벤트 알림](#)을 구성합니다.
- 이벤트 본문과 Lambda 환경 변수를 통해 입력 파라미터를 AWS Lambda 함수로 전달합니다.

도구

서비스

- [Amazon EMR](#) 은 AWS에서 빅 데이터 프레임워크 실행을 간소화하여 방대한 양의 데이터를 처리하고 분석하는 관리형 클러스터 플랫폼입니다.
- [Lambda](#)는 서버를 프로비저닝하거나 관리할 필요 없이 코드를 실행할 수 있도록 도와주는 컴퓨팅 서비스입니다. 필요할 때만 코드를 실행하며 자동으로 확장이 가능하므로 사용한 컴퓨팅 시간만큼만 비용을 지불합니다.
- [Amazon Simple Storage Service\(S3\)](#)는 원하는 양의 데이터를 저장, 보호 및 검색하는 데 도움이 되는 클라우드 기반 객체 스토리지 서비스입니다.

기타 도구

- [Apache Spark](#)는 대규모 데이터 처리를 위한 다중 언어 분석 엔진입니다.

에픽

Amazon EMR 및 Lambda IAM 역할과 VPC 생성

작업	설명	필요한 기술
IAM 역할과 VPC를 생성합니다.	이미 AWS Lambda 및 Amazon EMR IAM 역할과 VPC가 있는 경우 이 단계를 건너뛸 수 있습니다. 코드를 실행하려면 EMR 클러스터와 Lambda 함수 모두에 IAM 역할이 필요합니다. 또한 EMR 클러스터에는 NAT 게이트웨이와 함께 퍼블릭 서브넷이나 프라이빗 서브넷이 있는 VPC도 필요합니다. 모든 IAM 역할과 VPC를 자동으로 생성하려면 연결된 CloudFormation AWS 템플릿을 그대로 배포하거나 추가 정보 섹션에 지정된 대로 역할과 VPC를 수동으로 생성할 수 있습니다.	클라우드 아키텍트
AWS CloudFormation 템플릿 출력 키를 기록해 둡니다.	CloudFormation 템플릿이 성공적으로 배포되면 AWS CloudFormation 콘솔의 Outputs 탭으로 이동합니다. 다섯 가지 출력 키를 유념합니다. <ul style="list-style-type: none"> • S3Bucket • LambdaExecutionRole • ServiceRole • JobFlowRole 	클라우드 아키텍트

작업	설명	필요한 기술
	<ul style="list-style-type: none"> • Ec2SubnetId <p>Lambda 함수를 생성할 때 이러한 키의 값을 사용합니다.</p>	

Spark .jar 파일 업로드

작업	설명	필요한 기술
Spark .jar 파일을 업로드합니다.	Spark .jar 파일을 AWS CloudFormation 스택이 생성한 S3 버킷에 업로드합니다. 버킷 이름은 S3Bucket 출력 키와 동일합니다.	일반 AWS

Lambda 함수를 생성하여 EMR 클러스터를 시작합니다.

작업	설명	필요한 기술
Lambda 함수를 생성합니다.	Lambda 콘솔에서 실행 역할을 가진 Python 3.9+ Lambda 함수를 생성합니다. 실행 역할 정책은 Lambda가 EMR 클러스터를 시작할 수 있도록 허용해야 합니다. (첨부된 AWS CloudFormation 템플릿을 참조하십시오.)	데이터 엔지니어, 클라우드 엔지니어
코드를 복사하여 붙여넣습니다.	lambda_function.py 파일의 코드를 이 패턴의 추가 정보 섹션에 나와 있는 코드로 바꿉니다.	데이터 엔지니어, 클라우드 엔지니어

작업	설명	필요한 기술
코드의 파라미터를 변경합니다.	코드의 설명에 따라 파라미터 값을 AWS 계정에 맞게 변경합니다.	데이터 엔지니어, 클라우드 엔지니어
클러스터를 시작하는 함수를 실행합니다.	제공된 Spark .jar 파일을 사용하여 임시 EMR 클러스터 생성을 시작하는 함수를 실행합니다. 작업이 완료되면 Spark 작업과 템플릿이 자동으로 실행됩니다.	데이터 엔지니어, 클라우드 엔지니어
EMR 클러스터 상태를 확인합니다.	EMR 클러스터가 시작되면 Amazon EMR 콘솔의 클러스터 탭에 나타납니다. 클러스터를 시작하거나 작업을 실행하는 동안 발생하는 모든 오류를 적절히 확인할 수 있습니다.	데이터 엔지니어, 클라우드 엔지니어

샘플 데모 설정 및 실행

작업	설명	필요한 기술
Spark .jar 파일을 업로드합니다.	첨부 파일 섹션에서 Spark .jar 파일을 다운로드하여 S3 버킷에 업로드합니다.	데이터 엔지니어, 클라우드 엔지니어
입력 데이터를 업로드합니다.	첨부된 fake_sales_data.csv 파일을 S3 버킷에 업로드합니다.	데이터 엔지니어, 클라우드 엔지니어
Lambda 코드를 붙여넣고 파라미터를 변경합니다.	도구 섹션에서 코드를 복사하고 Lambda 함수에 코드를 붙여넣어 코드 lambda_function.py 파일을 바꿉니다.	데이터 엔지니어, 클라우드 엔지니어

작업	설명	필요한 기술
	다. 계정에 맞게 파라미터 값을 변경합니다.	
함수를 시작하고 출력을 확인합니다.	Lambda 함수는 제공된 Spark 작업으로 클러스터를 시작한 후 S3 버킷에 .csv 파일을 생성합니다.	데이터 엔지니어, 클라우드 엔지니어

관련 리소스

- [Building Spark](#)
- [Apache Spark 및 Amazon EMR](#)
- [Boto3 Docs run_job_flow 설명서](#)
- [Apache Spark 정보 및 설명서](#)

추가 정보

코드

```

"""
Copy paste the following code in your Lambda function. Make sure to change the
following key parameters for the API as per your account

-Name (Name of Spark cluster)
-LogUri (S3 bucket to store EMR logs)
-Ec2SubnetId (The subnet to launch the cluster into)
-JobFlowRole (Service role for EC2)
-ServiceRole (Service role for Amazon EMR)

The following parameters are additional parameters for the Spark job itself. Change the
bucket name and prefix for the Spark job (located at the bottom).

-s3://your-bucket-name/prefix/lambda-emr/SparkProfitCalc.jar (Spark jar file)
-s3://your-bucket-name/prefix/fake_sales_data.csv (Input data file in S3)
-s3://your-bucket-name/prefix/outputs/report_1/ (Output location in S3)
"""
import boto3

```

```

client = boto3.client('emr')

def lambda_handler(event, context):
    response = client.run_job_flow(
        Name='spark_job_cluster',
        LogUri='s3://your-bucket-name/prefix/logs',
        ReleaseLabel='emr-6.0.0',
        Instances={
            'MasterInstanceType': 'm5.xlarge',
            'SlaveInstanceType': 'm5.large',
            'InstanceCount': 1,
            'KeepJobFlowAliveWhenNoSteps': False,
            'TerminationProtected': False,
            'Ec2SubnetId': 'subnet-XXXXXXXXXXXXXXX'
        },
        Applications=[{'Name': 'Spark'}],
        Configurations=[
            {'Classification': 'spark-hive-site',
             'Properties': {
                 'hive.metastore.client.factory.class':
                 'com.amazonaws.glue.catalog.metastore.AWSGlueDataCatalogHiveClientFactory'
             }
        },
        VisibleToAllUsers=True,
        JobFlowRole='EMRLambda-EMREC2InstanceProfile-XXXXXXXXXX',
        ServiceRole='EMRLambda-EMRRole-XXXXXXXXXX',
        Steps=[
            {
                'Name': 'flow-log-analysis',
                'ActionOnFailure': 'TERMINATE_CLUSTER',
                'HadoopJarStep': {
                    'Jar': 'command-runner.jar',
                    'Args': [
                        'spark-submit',
                        '--deploy-mode', 'cluster',
                        '--executor-memory', '6G',
                        '--num-executors', '1',
                        '--executor-cores', '2',
                        '--class', 'com.aws.emr.ProfitCalc',
                        's3://your-bucket-name/prefix/lambda-emr/SparkProfitCalc.jar',
                        's3://your-bucket-name/prefix/fake_sales_data.csv',
                        's3://your-bucket-name/prefix/outputs/report_1/'
                    ]
                }
            }
        ]
    )

```

```

    ]
  }
}
]
)

```

IAM 역할 및 VPC 생성

Lambda 함수에서 EMR 클러스터를 시작하려면 VPC 및 IAM 역할이 필요합니다. 이 패턴의 첨부 파일 섹션에서 AWS CloudFormation 템플릿을 사용하여 VPC 및 IAM 역할을 설정하거나 다음 링크를 사용하여 수동으로 생성할 수 있습니다.

Lambda 및 Amazon EMR을 실행하려면 다음과 같은 IAM 역할이 필요합니다.

Lambda 실행 역할

Lambda 함수의 [실행 역할](#)은 AWS 서비스 및 리소스에 대한 액세스 권한을 부여합니다.

Amazon EMR의 서비스 역할

[Amazon EMR 역할](#)은 리소스를 프로비저닝하고 클러스터 내에서 실행 중인 Amazon Elastic Compute Cloud(Amazon EC2) 인스턴스의 맥락에서 수행되지 않는 서비스 수준 작업을 수행할 때 Amazon EMR에 대해 허용되는 작업을 정의합니다. 예를 들어, 서비스 역할은 클러스터를 시작할 때 EC2 인스턴스를 프로비저닝하는 데 사용됩니다.

EC2 인스턴스의 서비스 역할

[클러스터 EC2 인스턴스의 서비스 역할](#)(Amazon EMR에 대한 EC2 인스턴스 프로파일이라고도 함)은 인스턴스를 시작할 때 Amazon EMR 클러스터의 모든 EC2 인스턴스에 할당되는 특별한 유형의 서비스 역할입니다. Apache Hadoop의 상단에서 실행되는 애플리케이션 프로세스는 다른 AWS 서비스와 상호 작용하기 위한 권한에 대해 이 역할을 말합니다.

VPC 및 서브넷 생성

VPC 콘솔에서 [VPC](#)를 생성할 수 있습니다.

첨부

이 문서와 관련된 추가 콘텐츠에 액세스하려면 [attachment.zip](#) 파일의 압축을 풉니다.

AWS Glue를 사용하여 아파치 카산드라 워크로드를 Amazon Keyspaces로 마이그레이션하십시오.

작성자: Nikolai Kolesnikov (AWS), Karthiga Priya Chandran (AWS), Samir Patel (AWS)

환경: 프로덕션	소스: Cassandra	대상: Amazon Keyspaces
R 유형: 해당 없음	워크로드: 오픈 소스, 기타 모든 워크로드	기술: 분석, 마이그레이션, 서버리스, 빅 데이터
AWS 서비스: AWS Glue, 아마존 키스페이스, 아마존 S3, AWS CloudShell		

요약

이 패턴은 AWS Glue에서 CQLReplicator를 사용하여 기존 아파치 카산드라 워크로드를 Amazon Keyspaces (아파치 카산드라용) 로 마이그레이션하는 방법을 보여줍니다. AWS Glue의 CQLReplicator를 사용하면 워크로드를 마이그레이션하는 데 따른 복제 지연을 몇 분으로 줄일 수 있습니다. 또한, Amazon Simple Storage Service(Amazon S3) 버킷을 사용하여 [Apache Parquet](#) 파일, 구성 파일, 스크립트 등 마이그레이션에 필요한 데이터를 저장하는 방법도 알아봅니다. 이 패턴은 카산드라 워크로드가 가상 사설 클라우드 (VPC) 의 Amazon Elastic Compute Cloud (Amazon EC2) 인스턴스에 호스팅된다고 가정합니다.

사전 조건 및 제한 사항

사전 조건

- 소스 테이블이 있는 Cassandra 클러스터
- 워크로드 복제를 위한 Amazon Keyspace의 대상 테이블
- 점진적 데이터 변경이 포함된 중간 Parquet 파일을 저장할 S3 버킷
- 작업 구성 파일 및 스크립트를 저장할 S3 버킷

제한 사항

- AWS Glue의 CQLReplicator는 카산드라 워크로드를 위한 데이터 처리 장치 (DPU) 를 프로비저닝하는 데 시간이 걸립니다. Cassandra 클러스터와 Amazon Keyspaces의 대상 키스페이스 및 테이블 사이의 복제 지연은 단 몇 분 동안만 지속될 가능성이 높습니다.

아키텍처

소스 기술 스택

- Apache Cassandra
- DataStax 서버
- ScyllaDB

대상 기술 스택

- Amazon Keyspaces

마이그레이션 아키텍처

다음 다이어그램은 Cassandra 클러스터가 EC2 인스턴스에 호스팅되고 세 개의 가용 영역에 분산되어 있는 예제 아키텍처를 보여줍니다. Cassandra 노드는 개인 서브넷에서 호스팅됩니다.

이 다이어그램은 다음 워크플로를 보여줍니다.

1. 사용자 지정 서비스 역할은 Amazon Keyspaces 및 S3 버킷에 대한 액세스를 제공합니다.
2. AWS Glue 작업은 S3 버킷의 작업 구성 및 스크립트를 읽습니다.
3. AWS Glue 작업은 포트 9042를 통해 연결되어 Cassandra 클러스터에서 데이터를 읽습니다.
4. AWS Glue 작업은 포트 9142를 통해 연결하여 Amazon Keyspaces에 데이터를 씁니다.

도구

AWS 서비스 및 도구

- [AWS Command Line Interface\(AWS CLI\)](#)는 명령줄 셸에서 명령을 사용하여 AWS 서비스와 상호 작용할 수 있는 오픈 소스 도구입니다.

- [CloudShell](#)은 [AWS](#) 명령줄 인터페이스 (AWS CLI) 및 사전 설치된 다양한 개발 도구를 사용하여 AWS 서비스를 관리하는 데 사용할 수 있는 브라우저 기반 셸입니다.
- [AWS Glue](#)는 데이터 스토어와 데이터 스트림 간에 데이터를 안정적으로 분류, 정리, 보강, 이동할 수 있는 완전 관리형 ETL 서비스입니다.
- [Amazon Keyspaces\(Apache Cassandra용\)](#)는 AWS 클라우드에서 Cassandra 워크로드를 마이그레이션, 실행, 확장할 수 있도록 지원하는 관리형 데이터베이스 서비스입니다.

코드

[이 패턴의 코드는 CQLReplicator 리포지토리에서 사용할 수 있습니다. GitHub](#)

모범 사례

- 마이그레이션에 필요한 AWS Glue 리소스를 결정하려면 원본 Cassandra 테이블의 행 수를 추정하십시오. 예를 들어, 84GB 디스크를 사용하는 경우 0.25DPU (vCPU 2개, 메모리 4GB) 당 250,000개의 행을 예로 들 수 있습니다.
- CQLReplicator를 실행하기 전에 Amazon Keyspaces 테이블을 미리 워밍하십시오. 예를 들어 8개의 CQLReplicator 타일 (AWS Glue 작업) 은 초당 최대 22K WCU를 쓸 수 있으므로 타겟은 초당 최대 25~30K WCU까지 미리 워밍해야 합니다.
- AWS Glue 구성 요소 간의 통신을 활성화하려면 보안 그룹의 모든 TCP 포트에 대해 자체 참조 인바운드 규칙을 사용하십시오.
- 증분 트래픽 전략을 사용하여 시간 경과에 따라 마이그레이션 워크로드를 분산할 수 있습니다.

에픽

CQLReplicator를 배포하십시오.

작업	설명	필요한 기술
대상 키스페이스와 테이블을 생성합니다.	1. Amazon Keyspace에서 키스페이스와 테이블 을 생성합니다. 쓰기 용량에 대한 자세한 내용은 이 패턴의 추가 정보 섹	앱 소유자, AWS 관리자, DBA, 앱 개발자

작업	설명	필요한 기술
	<p>선에 있는 쓰기 단위 계산을 참조하십시오.</p> <p>Cassandra Query Language(CQL)를 사용하여 키스페이스를 생성할 수도 있습니다. 자세한 내용은 이 패턴의 추가 정보 섹션에서 CQL을 사용하여 키스페이스 만들기를 참조하십시오.</p> <p>참고: 테이블을 생성한 후, 불필요한 요금이 부과되지 않도록 테이블을 온디맨드 용량 모드로 테이블을 전환하는 것을 고려해 보십시오.</p> <p>2. 처리량 모드로 업데이트하려면 다음 스크립트를 실행합니다.</p> <pre>ALTER TABLE target_keyspace.target_table WITH CUSTOM_PROPERTIES = { 'capacity_mode': { 'throughput_mode': 'PAY_PER_REQUEST' } }</pre>	

작업	설명	필요한 기술
<p>Cassandra에 연결하도록 Cassandra 드라이버를 구성합니다.</p>	<p>다음 구성 스크립트를 사용합니다.</p> <pre data-bbox="597 346 1027 1339"> Datastax-java-driver { basic.request.consistency = "LOCAL_QUORUM" basic.contact-points = ["127.0.0.1:9042"] advanced.reconnect-on-init = true basic.load-balancing-policy { local-datascenter = "datacenter1" } advanced.auth-provider = { class = PlainTextAuthProvider username = "user-at-sample" password = "S@MPLE=PASSWORD=" } } </pre> <p>참고: 위 스크립트는 Spark Cassandra Connector를 사용합니다. 자세한 내용은 Cassandra의 참조 구성을 참조하십시오.</p>	<p>DBA</p>

작업	설명	필요한 기술
<p>Amazon Keyspaces에 연결하도록 Cassandra 드라이버를 구성합니다.</p>	<p>다음과 같은 구성 스트립트를 사용합니다.</p> <pre data-bbox="592 346 1031 1831"> datastax-java-driver { basic { load-balancing-policy { local-datacenter = us-west-2 } contact-points = ["cassandra.us-west-2.amazonaws.com:9142"] request { page-size = 2500 timeout = 360 seconds consistency = LOCAL_QUORUM } } advanced { control-connection { timeout = 360 seconds } session-leak.threshold = 6 connection { connect-timeout = 360 seconds init-query-timeout = 360 seconds warn-on-init-error = false } auth-provider = { class = software. aws.mcs.auth.SigV4 AuthProvider </pre>	<p>DBA</p>

작업	설명	필요한 기술
	<pre> aws-region = us- west-2 } ssl-engine-factory { class = DefaultSs lEngineFactory } } </pre> <p>참고: 위 스크립트는 Spark Cassandra Connector를 사용합니다. 자세한 내용은 Cassandra의 참조 구성을 참조하십시오.</p>	
<p>AWS Glue 작업에 대한 IAM 역할을 생성합니다.</p>	<p>AWS Glue를 신뢰할 수 있는 개체로 glue-cassandra-migration 사용하여 이름을 지정한 새 AWS 서비스 역할을 생성합니다.</p> <p>참고: 는 S3 버킷과 Amazon Keyspace에 대한 읽기 및 쓰기 액세스를 glue-cassandra-migration 제공해야 합니다. S3 버킷에는 .jar 파일, Amazon Keyspaces 및 Cassandra용 구성 파일, 중간 Parquet 파일이 들어 있습니다. 예를 들어, 여기에는,, 관리형 정책이 포함됩니다.</p> <pre> AWSGlueServiceRole AmazonS3FullAccess AmazonKeyspacesFullAccess </pre>	<p>AWS DevOps</p>

작업	설명	필요한 기술
<p>AWS에서 CQL 리플리케이터를 다운로드하십시오. CloudShell</p>	<p>다음 명령을 실행하여 홈 폴더에 프로젝트를 다운로드합니다.</p> <pre data-bbox="594 394 1029 951">git clone https://github.com/aws-samples/cql-replicator.git cd cql-replicator/glue # Only for AWS CloudShell, the bc package includes bc and dc. Bc is an arbitrary precision numeric processing arithmetic language sudo yum install bc -y</pre>	
<p>참조 구성 파일을 수정합니다.</p>	<p>CassandraConnector.conf Keyspaces Connector.conf 를 프로젝트 폴더의 ../glue/conf 디렉토리로 복사합니다.</p>	<p>AWS DevOps</p>

작업	설명	필요한 기술
<p>마이그레이션 프로세스 시작.</p>	<p>다음 명령은 CQLReplicator 환경을 초기화합니다. 초기화에는 .jar 아티팩트를 복사하고 AWS Glue 커넥터, S3 버킷, AWS Glue 작업, migration 키스페이스 및 테이블을 생성하는 작업이 포함됩니다.</p> <p>ledger</p> <pre data-bbox="594 632 1029 1388"> cd cql-replicator/glu e/bin ./cqlreplicator --state init --sg "sg-1","sg-2" \ --subnet "subnet-XXXXXXXXXXXX" \ --az us- west-2a --region us- west-2 \ --glue- iam-role glue-cass andra-migration \ -- landing-zone s3://cql- replicator-1234567 890-us-west-2 </pre> <p>이 스크립트에는 다음 파라미터가 포함되어 있습니다.</p> <ul style="list-style-type: none"> • --sg— AWS Glue에서 Cassandra 클러스터에 액세스할 수 있도록 허용하고 모든 트래픽에 대한 자체 참조 인바운드 규칙을 포함하는 보안 그룹 	<p>AWS DevOps</p>

작업	설명	필요한 기술
	<ul style="list-style-type: none"> • <code>--subnet</code>— Cassandra 클러스터가 속한 서브넷 • <code>--az</code>— 서브넷의 가용 영역 • <code>--region</code>— 카산드라 클러스터가 배포된 AWS 지역 • <code>--glue-iam-role</code> — 사용자를 대신하여 Amazon Keyspace와 Amazon S3를 호출할 때 AWS Glue가 맡을 수 있는 IAM 역할 권한 • <code>--landing zone</code>— S3 버킷 재사용을 위한 선택적 파라미터 (<code>--landing zone</code> 파라미터 값을 제공하지 않으면 <code>init</code> 프로세스에서 구성 파일, <code>.jar</code> 아티팩트 및 중간 파일을 저장할 새 버킷을 만들려고 시도합니다.) 	
<p>배포를 검증합니다.</p>	<p>이전 명령을 실행한 후 AWS 계정에는 다음이 포함되어야 합니다.</p> <ul style="list-style-type: none"> • CQLReplicator AWS Glue 작업 및 AWS Glue의 AWS Glue 커넥터 • 아티팩트를 저장하는 S3 버킷 • Amazon Keyspace의 대상 키스페이스 migration 및 ledger 테이블 	<p>AWS DevOps</p>

CQL 리플리케이터 실행

작업	설명	필요한 기술
<p>마이그레이션 프로세스를 시작합니다.</p>	<p>AWS Glue에서 CQLReplicator를 작동하려면 <code>--state run</code> 명령과 일련의 파라미터를 차례로 사용해야 합니다. 이러한 파라미터의 정확한 구성은 주로 고유한 마이그레이션 요구 사항에 따라 결정됩니다. 예를 들어 TTL (Time to Live) 값 및 업데이트를 복제하거나 1MB를 초과하는 객체를 Amazon S3로 오프로드하는 경우 이러한 설정이 달라질 수 있습니다.</p> <p>Cassandra 클러스터에서 Amazon Keyspaces로 워크로드를 복제하려면 다음 명령을 실행합니다.</p> <pre data-bbox="594 1150 1029 1843"> ./cqlreplicator --state run --tiles 8 \ -- landing-zone s3://cql-replicator-1234567 890-us-west-2 \ --region us-west-2 \ --src- keyspace source_keyspace \ --src- table source_table \ --trg- keyspace target_keyspace \ </pre>	<p>AWS DevOps</p>

작업	설명	필요한 기술
	<pre> -- writetime-column column_name \ --trg- table target_table -- inc-traffic </pre> <p>소스 키스페이스와 테이블은 Cassandra 클러스터에 있습니다. source_keyspace.source_table 대상 키스페이스와 테이블은 Amazon target_keyspace.target_table Keyspace에 있습니다. 파라미터는 많은 요청으로 인해 증가하는 트래픽이 Cassandra 클러스터 및 Amazon Keyspace에 과부하되는 것을 방지하는 --inc-traffic 데 도움이 됩니다.</p> <p>업데이트를 복제하려면 명령줄에 추가하십시오. --writetime-column regular_column_name 일반 열이 쓰기 타임스탬프의 소스로 사용될 것입니다.</p>	

마이그레이션 프로세스 모니터링

작업	설명	필요한 기술
<p>이전 마이그레이션 단계에서 마이그레이션된 Cassandra 행을 검증하십시오.</p>	<p>채우기 단계에서 복제된 행 수를 확인하려면 다음 명령을 실행합니다.</p>	<p>AWS DevOps</p>

작업	설명	필요한 기술
	<pre>./cqlreplicator --state stats \ -- landing-zone s3://cql- replicator-1234567 890-us-west-2 \ --src- keyspace source_ke yspace --src-table source_table --region us-west-2</pre>	

마이그레이션 프로세스 중지

작업	설명	필요한 기술
<p>cqlreplicator 명령 또는 AWS Glue 콘솔을 사용하십시오.</p>	<p>마이그레이션 프로세스를 정상적으로 중지하려면 다음 명령을 실행합니다.</p> <pre>./cqlreplicator --state request-stop --tiles 8 \ -- landing-zone s3://cql- replicator-1234567 890-us-west-2 \ --region us-west-2 \ --src- keyspace source_ke yspace --src-table source_table</pre>	<p>AWS DevOps</p>

작업	설명	필요한 기술
	마이그레이션 프로세스를 즉시 중지하려면 AWS Glue 콘솔을 사용하십시오.	

정리

작업	설명	필요한 기술
배포된 리소스를 삭제합니다.	<p>다음 명령은 AWS Glue 작업, 커넥터, S3 버킷 및 Keyspaces 테이블을 삭제합니다. ledger</p> <pre>./cqlreplicator --state cleanup --landing-zone s3://cql-replicator-1234567890-us-west-2</pre>	AWS DevOps

문제 해결

문제	Solution
AWS Glue 작업이 실패하고 메모리 부족 (OOM) 오류가 반환되었습니다.	<ol style="list-style-type: none"> 작업자 유형을 변경 (스케일 업) 하십시오. 예를 들어, G0.25X 로 G.1X 또는 G.1X 로 변경합니다 G.2X. 또는 CQLReplicator에서 AWS Glue 작업 (스케일 아웃) 당 DPU 수를 늘리십시오. 중단된 시점부터 마이그레이션 프로세스를 시작하십시오. 실패한 CQLReplicator 작업을 다시 시작하려면 동일한 매개 변수를 사용하여 명령을 다시 실행합니다. --state run

관련 리소스

- [AWS Glue를 사용한 CQL 리플리케이터 README.MD](#)
- [AWS Glue 설명서](#)
- [아마존 키스페이스 설명서](#)
- [아파치 카산드라](#)

추가 정보

마이그레이션 고려 사항

AWS Glue를 사용하여 Cassandra 워크로드를 Amazon Keyspace로 마이그레이션하는 동시에 마이그레이션 프로세스 중에 Cassandra 소스 데이터베이스가 완전히 기능하도록 유지할 수 있습니다. 복제가 완료되면 Cassandra 클러스터와 Amazon Keyspaces 간의 복제 지연 시간(몇 분 미만)을 최소화하면서 애플리케이션을 Amazon Keyspaces로 전환하도록 선택할 수 있습니다. 데이터 일관성을 유지하기 위해 유사한 파이프라인을 사용하여 Amazon Keyspaces에서 데이터를 Cassandra 클러스터로 다시 복제할 수도 있습니다.

쓰기 단위 계산

한 시간 동안 행 크기가 1KiB인 500,000,000을 쓰는 경우를 예로 들어 보겠습니다. 필요한 Amazon Keyspaces 쓰기 단위(WCU)의 총 수는 다음 계산을 기반으로 합니다.

$$\begin{aligned} & (\text{number of rows}/60 \text{ mins } 60\text{s}) \text{ 1 WCU per row} = (500,000,000/(60*60\text{s})) * 1 \text{ WCU} \\ & = 69,444 \text{ WCUs required} \end{aligned}$$

초당 69,444WCU는 1시간 요금이지만, 오버헤드에 대비하여 약간의 여유를 추가할 수 있습니다. 예를 들어, $69,444 * 1.10 = 76,388$ WCUs는 오버헤드가 10%입니다.

CQL을 사용하여 키스페이스 생성

CQL을 사용하여 키스페이스를 생성하려면 다음 명령을 실행합니다.

```
CREATE KEYSPACE target_keyspace WITH replication = {'class': 'SingleRegionStrategy'}
CREATE TABLE target_keyspace.target_table (
  userid uuid, level text, gameid int,
  description text, nickname text, zip text, email text, updatetime text, PRIMARY KEY
  (userid, level, gameid) ) WITH default_time_to_live = 0 AND CUSTOM_PROPERTIES =
  {'capacity_mode':{'throughput_mode':'PROVISIONED', 'write_capacity_units':76388,
  'read_capacity_units':3612 }} AND CLUSTERING ORDER BY (level ASC, gameid ASC)
```


온프레미스 서버에서 Oracle Business Intelligence 12c를 AWS 클라우드로 마이그레이션.

제작: 레인 (랜레이) 쇼운미 (AWS), 패트릭 황 (AWS)

환경: 프로덕션	소스: 온프레미스	대상: Amazon EC2, Amazon RDS, Amazon ALB, Amazon EFS
R 유형: 리플랫폼	워크로드: Oracle	기술: 분석, 데이터베이스
AWS 서비스: 아마존 EBS, 아마존 EC2, 아마존 EFS, CloudFormation AWS, 엘라스틱 로드 밸런싱 (ELB), AWS 인증서 관리자 (ACM)		

요약

이 패턴은 AWS를 사용하여 [Oracle 비즈니스 인텔리전스 엔터프라이즈 에디션 12c](#)를 온프레미스 서버에서 AWS 클라우드로 마이그레이션하는 방법을 보여줍니다. CloudFormation 또한 다른 AWS 서비스를 사용하여고가용성, 보안, 유연성 및 동적 규모 조정 기능을 제공하는 Oracle BI 12c 구성 요소를 구현하는 방법도 설명합니다.

Oracle BI 12c를 AWS 클라우드로 마이그레이션하는 것과 관련된 모범 사례 목록은 이 패턴의 추가 정보 섹션을 참조하세요.

참고: 기존 Oracle BI 12c 데이터를 클라우드로 전송하기 전에 여러 테스트 마이그레이션을 실행하는 것이 가장 좋습니다. 이러한 테스트를 통해 마이그레이션 접근 방식을 미세 조정하고, 잠재적 문제를 식별 및 수정하고, 가동 중지 시간 요구 사항을 보다 정확하게 예측할 수 있습니다.

사전 조건 및 제한 사항

사전 조건

- 활성 상태의 AWS 계정.

- [AWS Virtual Private Network\(AWS VPN\)](#) 서비스 또는 [AWS Direct Connect](#)를 통해 온프레미스 서버와 AWS 간의 네트워크 연결을 보호합니다.
- 오라클 운영 체제, 오라클 BI 12c, 오라클 데이터베이스, 오라클 WebLogic 서버, 오라클 HTTP 서버 용 소프트웨어 라이선스

제한 사항

스토리지 크기 제한에 대한 자세한 내용은 [Oracle용 Amazon Relational Database Service\(Amazon RDS\)](#) 설명서를 참조하세요.

제품 버전

- Oracle Business Intelligence Enterprise Edition 12c
- 오라클 WebLogic 서버 12c
- Oracle HTTP Server 12c
- Oracle Database 12c(또는 그 이상)
- Oracle Java SE 8

아키텍처

다음 다이어그램은 AWS 클라우드에서 Oracle BI 12c 구성 요소를 실행하기 위한 예제 아키텍처를 보여줍니다.

이 다이어그램은 다음 아키텍처를 보여줍니다.

1. Amazon Route 53은 도메인 이름 서비스(DNS) 구성을 제공합니다.
2. Elastic Load Balancing(ELB)은 네트워크 트래픽을 분산하여 여러 가용 영역에 걸쳐 Oracle BI 12c 구성 요소의 확장성과 가용성을 개선합니다.
3. Amazon Elastic Compute Cloud(Amazon EC2) Auto Scaling 그룹은 여러 가용 영역에 걸쳐 Oracle HTTP Server, Weblogic Admin 서버, 관리형 BI 서버를 호스팅합니다.
4. Oracle 데이터베이스용 Amazon Relational Database Service(Amazon RDS)는 여러 가용 영역에 걸쳐 BI Server 메타데이터를 저장합니다.
5. Amazon Elastic File System(Amazon EFS)은 공유 파일 스토리지의 모든 Oracle BI 12c 구성 요소에 탑재됩니다.

기술 스택

- Amazon Elastic Block Store(Amazon EBS)
- Amazon Elastic Compute Cloud(Amazon EC2)
- Amazon Elastic File System(Amazon EFS)
- Amazon RDS for Oracle
- AWS Certificate Manager (ACM)
- Elastic Load Balancing(ELB)
- Oracle BI 12c
- 오라클 WebLogic 서버 12c
- Oracle HTTP Server(OHS)

도구

- [AWS는 AWS CloudFormation](#) 리소스를 설정하고, 빠르고 일관되게 프로비저닝하고, 전체 AWS 계정 및 지역의 수명 주기 전반에 걸쳐 리소스를 관리할 수 있도록 지원합니다.
- [AWS Certificate Manager\(ACM\)](#)는 AWS 웹사이트와 애플리케이션을 보호하는 퍼블릭 및 프라이빗 SSL/TLS X.509 인증서와 키를 만들고, 저장하고, 갱신하는 데 도움을 줍니다.
- [AWS Database Migration Service\(AWS DMS\)](#)를 사용하면 데이터 스토어를 AWS 클라우드로 마이그레이션하거나 클라우드와 온프레미스 설정 간에 데이터 스토어를 마이그레이션할 수 있습니다.
- [Amazon Elastic Compute Cloud\(Amazon EC2\)](#)는 AWS 클라우드에서 규모를 조정할 수 있는 컴퓨팅 용량을 제공합니다. 필요한 만큼 가상 서버를 시작하고 빠르게 규모를 조정할 수 있습니다.
- [Amazon EC2 Auto Scaling](#)을 사용하면 애플리케이션 가용성을 유지하고 정의된 조건에 따라 Amazon EC2 인스턴스를 자동으로 추가하거나 제거할 수 있습니다.
- [Amazon Elastic File System\(Amazon EFS\)](#)은 AWS 클라우드에서 공유 파일 시스템을 생성하고 구성하는 데 도움이 됩니다.
- [Elastic Load Balancing\(ELB\)](#)은 들어오는 애플리케이션 또는 네트워크 트래픽을 여러 대상에 분산합니다. 예를 들어 하나 이상의 가용 영역에 있는 Amazon Elastic Compute Cloud(Amazon EC2) 인스턴스, 컨테이너, IP 주소 전반에 걸쳐 트래픽을 분산할 수 있습니다.
- [Amazon Relational Database Service\(Amazon RDS\)](#)는 AWS 클라우드에서 관계형 데이터베이스를 설정, 운영 및 조정하는 데 도움이 됩니다.
- [Amazon Simple Storage Service\(S3\)](#)는 원하는 양의 데이터를 저장, 보호 및 검색하는 데 도움이 되는 클라우드 기반 객체 스토리지 서비스입니다.

- [Amazon Virtual Private Cloud\(VPC\)](#)를 이용하면 사용자가 정의한 가상 네트워크로 AWS 리소스를 시작할 수 있습니다. 이 가상 네트워크는 사용자의 자체 데이터 센터에서 운영하는 기존 네트워크와 유사하며 AWS의 확장 가능한 인프라를 사용한다는 이점이 있습니다.
- [Oracle Data Pump](#)를 사용하면 한 데이터베이스에서 다른 데이터베이스로 데이터와 메타데이터를 빠른 속도로 이동할 수 있습니다.
- [Oracle Fusion Middleware](#)는 ID 관리, 협업 및 비즈니스 인텔리전스 보고를 위한 애플리케이션 개발 도구 및 통합 솔루션 제품군입니다.
- [GoldenGate](#) 오라클은 [Oracle Cloud Infrastructure](#)에서 데이터 복제 및 스트리밍 데이터 처리 솔루션을 설계, 실행, 조정 및 모니터링할 수 있도록 지원합니다.
- [Oracle WebLogic 스크립팅 툴 \(WLST\)](#) 은 클러스터를 수평적으로 확장할 수 있는 명령줄 인터페이스를 제공합니다. WebLogic

에픽

소스 환경 평가

작업	설명	필요한 기술
소프트웨어 인벤토리 정보를 수집합니다.	<p>다음은 포함하여 각 소스 기술 스택의 소프트웨어 구성 요소에 대한 버전 및 패치 수준을 식별합니다.</p> <ul style="list-style-type: none"> • Oracle 운영 체제 • Oracle Database • Oracle BI 12c • WebLogic 오라클 서버 • Oracle HTTP Server • Java 	마이그레이션 아키텍트, 솔루션 아키텍트, 애플리케이션 소유자, Oracle BI 관리자
컴퓨팅 및 스토리지 인벤토리 정보를 수집합니다.	<p>소스 환경에서 다음과 같은 현재 및 과거 사용률 지표를 검토합니다.</p> <ul style="list-style-type: none"> • CPU 사용량 	마이그레이션 아키텍트, 솔루션 아키텍트, 애플리케이션 소유자, Oracle BI 관리자, 시스템 관리자

작업	설명	필요한 기술
	<ul style="list-style-type: none"> 메모리 사용량 스토리지 사용량 <p>중요: 과거 사용량 급증을 고려해야 합니다.</p>	
<p>소스 환경의 아키텍처 및 요구 사항에 대한 정보를 수집합니다.</p>	<p>다음에 대한 지식을 포함하여 소스 환경의 아키텍처 및 요구 사항을 완전히 이해합니다.</p> <ul style="list-style-type: none"> 오라클 WebLogic 서버 도메인 구성 클러스터링 로드 밸런싱 연결 가용성 재해 복구 요구 사항 	<p>마이그레이션 아키텍트, 솔루션 아키텍트, 애플리케이션 소유자, Oracle BI 관리자</p>
<p>Java Database Connectivity(JDBC) 데이터 소스를 식별합니다.</p>	<p>소스 환경의 JDBC 데이터 소스 및 사용하는 각 데이터베이스 엔진에 대한 드라이버 정보를 수집합니다.</p>	<p>마이그레이션 아키텍트, 애플리케이션 소유자, Oracle BI 관리자, 데이터베이스 엔지니어 또는 관리자</p>
<p>환경별 설정에 대한 정보를 수집합니다.</p>	<p>다음을 포함하여 소스 환경과 관련된 설정 및 구성에 대한 정보를 수집합니다.</p> <ul style="list-style-type: none"> 사용자 지정 시작 및 종료 스크립트 Java 및 기타 환경 변수 인증서 	<p>마이그레이션 아키텍트, 솔루션 아키텍트, 애플리케이션 소유자, Oracle BI 관리자</p>

작업	설명	필요한 기술
다른 애플리케이션에 대한 종속성을 식별합니다.	<p>다른 애플리케이션과의 종속성을 유발하는 소스 환경에서의 통합에 대한 정보를 수집합니다.</p> <p>중요: 경량 디렉터리 액세스 프로토콜(LDAP) 통합 및 기타 네트워킹 요구 사항을 반드시 확인합니다.</p>	마이그레이션 아키텍트, 솔루션 아키텍트, 애플리케이션 소유자, Oracle BI 관리자

대상 환경 설계

작업	설명	필요한 기술
상위 수준 설계 문서를 작성합니다.	대상 아키텍처 설계 문서를 작성합니다. 소스 환경을 평가할 때 수집한 정보를 설계 문서에 반영해야 합니다.	솔루션스 아키텍트, 애플리케이션 아키텍트, 데이터베이스 엔지니어, 마이그레이션 아키텍트
설계 문서에 대한 승인을 얻습니다.	이해 관계자와 함께 설계 문서를 검토하고 필요한 승인을 받습니다.	애플리케이션 또는 서비스 소유자, 솔루션 아키텍트, 애플리케이션 아키텍트

인프라 배포

작업	설명	필요한 기술
에서 인프라 코드를 준비하십시오 CloudFormation.	<p>CloudFormation 템플릿을 생성하여 AWS 클라우드에서 Oracle BI 12c 인프라를 프로비저닝하십시오.</p> <p>자세한 내용은 AWS CloudFormation 사용 설명서의</p>	클라우드 인프라 아키텍트, 솔루션 아키텍트, 애플리케이션 아키텍트

작업	설명	필요한 기술
	<p>AWS CloudFormation 템플릿 사용을 참조하십시오.</p> <p>참고: 모든 리소스에 대해 하나의 큰 CloudFormation 템플릿을 생성하는 대신 각 Oracle BI 12c 계층에 대해 모듈식 템플릿을 생성하는 것이 가장 좋습니다. CloudFormation 모범 사례에 대한 자세한 내용은 AWS 블로그에서 CloudFormation AWS로 배포를 자동화하는 8가지 모범 사례를 참조하십시오.</p>	
<p>필수 소프트웨어를 다운로드합니다.</p>	<p>Oracle 웹사이트에서 필요한 버전 및 패치와 함께 다음 소프트웨어를 다운로드합니다.</p> <ul style="list-style-type: none"> • Java JDK8 • 오라클 서버 12c WebLogic • Oracle BI 12c 	<p>마이그레이션 아키텍트, 데이터베이스 엔지니어, 애플리케이션 아키텍트</p>
<p>설치 스크립트를 준비합니다.</p>	<p>자동 설치를 실행하는 소프트웨어 설치 스크립트를 생성합니다. 이 스크립트는 배포 자동화를 단순화합니다.</p> <p>자세한 내용은 Oracle Support 사이트에서 OBIEE 12c: 자동 설치 수행 방법을 참조하십시오. 설명서를 보려면 Oracle Support 계정이 필요합니다.</p>	<p>마이그레이션 아키텍트, 데이터베이스 엔지니어, 애플리케이션 아키텍트</p>

작업	설명	필요한 기술
<p>웹 및 애플리케이션 티어를 위한 Amazon EBS 기반 Linux AMI를 생성합니다.</p>	<ol style="list-style-type: none"> 1. 웹 및 애플리케이션 티어에 맞게 Amazon EC2 인스턴스를 배포 및 구성합니다. 인스턴스가 다음을 실행하기 위한 사전 요구 사항을 충족하는지 확인합니다. <ul style="list-style-type: none"> • Oracle 운영 체제 환경 설정 • Oracle 운영 체제 사용자 계정 설정 • Java 소프트웨어 설치 2. 인스턴스의 Amazon Machine Image(AMI)를 생성하고 나중에 사용할 수 있도록 사본을 저장합니다. 자세한 내용은 Linux 인스턴스용 Amazon EC2 사용 설명서의 Amazon EBS 기반 Linux AMI 생성을 참조하세요. 	<p>마이그레이션 아키텍트, 데이터베이스 엔지니어, 애플리케이션 아키텍트</p>
<p>를 사용하여 AWS 인프라를 시작하십시오 CloudFormation.</p>	<p>생성한 CloudFormation 템플릿을 사용하여 Oracle BI 12c 웹 및 애플리케이션 티어를 모듈로 배포하십시오.</p> <p>지침은 AWS 사용 CloudFormation 설명서의 AWS CloudFormation 시작하기를 참조하십시오.</p>	<p>클라우드 인프라 아키텍트, 솔루션 아키텍트, 애플리케이션 아키텍트</p>

새로운 설치로 Oracle BI 12c를 AWS로 마이그레이션하기

작업	설명	필요한 기술
필수 소프트웨어를 스테이징합니다.	Amazon EC2 인스턴스에 액세스할 수 있는 위치에 필수 소프트웨어를 스테이징합니다. 예를 들어, Amazon S3 또는 웹 및 애플리케이션 서버에 액세스할 수 있는 다른 Amazon EC2 인스턴스에 소프트웨어를 스테이징합니다.	마이그레이션 아키텍트, Oracle BI 아키텍트, 클라우드 인프라 아키텍트, 솔루션 아키텍트, 애플리케이션 아키텍트
Oracle BI 12c 설치를 위한 리포지토리 데이터베이스를 준비합니다.	Amazon RDS for Oracle 데이터베이스 인스턴스에 대해 Oracle 리포지토리 생성 유틸리티(RCU) 를 실행하여 Oracle BI 12c 스키마를 생성합니다.	클라우드 인프라 아키텍트, 솔루션 아키텍트, 애플리케이션 아키텍트, 마이그레이션 아키텍트, Oracle BI 아키텍트
Oracle Fusion Middleware 12c 및 Oracle BI 12c를 설치합니다.	<p>1. Amazon EC2 인스턴스 하나로 시작하여 Oracle Fusion Middleware 12c 인프라와 OBIEE 12c를 설치합니다. 자세한 내용은 Oracle Business Intelligence용 Oracle Fusion Middleware Enterprise 배포 가이드의 다음 섹션을 참조하세요.</p> <ul style="list-style-type: none"> • BIHOST1에서 인프라 설치 프로그램 시작 • 엔터프라이즈 배포 준비를 위한 Oracle Business Intelligence 설치 <p>참고: Amazon EFS를 사용하여 Oracle BI 12c 클러스</p>	마이그레이션 아키텍트, Oracle BI 아키텍트

작업	설명	필요한 기술
	<p>터 노드 간에 공유될 디렉터리를 호스팅할 수 있습니다.</p> <p>2. 설치에 필요한 모든 패치를 적용합니다.</p> <p>3. 인스턴스의 AMI를 생성하고 나중에 사용할 수 있도록 사본을 저장합니다.</p>	
<p>Oracle BI 12c에 맞게 오라클 WebLogic 서버 도메인을 구성하십시오.</p>	<p>Oracle BI 12c 도메인을 비클러스터형 배포로 구성합니다.</p> <p>자세한 내용은 Oracle Business Intelligence용 Oracle Fusion Middleware Enterprise 배포 가이드의 BI 도메인 구성을 참조하세요.</p>	<p>마이그레이션 아키텍트, Oracle BI 아키텍트</p>
<p>Oracle BI 12c에서 수평적 스케일 아웃을 수행합니다.</p>	<p>단일 노드를 원하는 노드 수까지 수평으로 스케일 아웃합니다.</p> <p>자세한 내용은 Oracle Business Intelligence용 Oracle Fusion Middleware Enterprise 배포 가이드의 Oracle Business Intelligence 스케일 아웃을 참조하세요.</p>	<p>마이그레이션 아키텍트, Oracle BI 아키텍트</p>

작업	설명	필요한 기술
<p>Oracle HTTP Server 12c를 설치합니다.</p>	<ol style="list-style-type: none"> 1. Oracle 웹 티어 Amazon EC2 인스턴스에 Oracle HTTP Server 12c를 설치합니다. 지침은 Oracle 액세스 관리 12c용 Oracle HTTP Server 설치 및 구성의 Oracle HTTP Server 12c 설치를 참조하세요. 2. 설치에 필요한 모든 패치를 적용합니다. 3. 인스턴스의 AMI를 생성하고 나중에 사용할 수 있도록 사본을 저장합니다. 	<p>마이그레이션 아키텍트, Oracle BI 아키텍트</p>
<p>SSL 종료를 위한 로드 밸런서를 구성합니다.</p>	<ol style="list-style-type: none"> 1. ACM에서 인증서를 가져오기 또는 생성합니다. 2. SSL 인증서를 ELB에 연결합니다. 	<p>클라우드 인프라 아키텍트, 마이그레이션 아키텍트</p>
<p>비즈니스 인텔리전스 메타데이터 아티팩트를 AWS로 마이그레이션합니다.</p>	<ol style="list-style-type: none"> 1. 온프레미스 Oracle BI 12c 설치에서 Oracle Business Intelligence 애플리케이션 아카이브(BAR) 파일을 내보냅니다. BAR 파일을 내보내려면 WebLogic 스크립팅 도구 (WLST) 를 사용하여 명령을 실행합니다. exportServiceInstance 2. 온프레미스 BAR 파일을 AWS Oracle BI 12c 설치로 가져옵니다. BAR 파일을 가져오려면 importServiceInstanceWLST 명령을 실행합니다. 	<p>마이그레이션 아키텍트, Oracle BI 아키텍트</p>

작업	설명	필요한 기술
<p>마이그레이션 후 작업을 수행합니다.</p>	<p>BAR 파일을 가져온 후 다음을 수행합니다.</p> <ul style="list-style-type: none"> • 추가 JDBC 데이터 소스를 구성합니다. • PostgreSQL 또는 Amazon Redshift와 같은 다른 데이터 소스용 드라이버를 설치합니다. • 오라클 LDAP, SSL, 싱글 사인온 (SSO) 및 보안 저장소를 구성합니다. WebLogic • AWS Identity and Access Management(IAM) 정책을 구성합니다. • 사용량 추적을 활성화합니다. • 다른 시스템과의 통합을 설정합니다. • 모든 사용자 지정 스크립트를 마이그레이션합니다. 	<p>마이그레이션 아키텍트, Oracle BI 아키텍트</p>

새 환경을 테스트합니다.

작업	설명	필요한 기술
<p>새로운 Oracle BI 12c 환경을 테스트합니다.</p>	<p>새로운 Oracle end-to-end BI 12c 환경에서 테스트를 수행하십시오. 자동화를 최대한 많이 사용합니다.</p> <p>테스트 활동의 예는 다음과 같습니다.</p>	<p>마이그레이션 아키텍트, 솔루션 아키텍트, 애플리케이션 소유자, Oracle BI 관리자</p>

작업	설명	필요한 기술
	<ul style="list-style-type: none"> 대시보드, 보고서 및 URL 검증 사용자 승인 테스트(UAT) 운영 승인 테스트(OAT) <p>참고: 필요에 따라 추가 테스트 및 검증을 수행합니다.</p>	

새 환경으로 전환

작업	설명	필요한 기술
온프레미스 Oracle BI 12c 환경으로의 트래픽을 끊습니다.	지정된 전환 기간에 온프레미스 Oracle BI 12c 환경으로 향하는 모든 트래픽을 중지합니다.	마이그레이션 아키텍트, 솔루션 아키텍트, 애플리케이션 소유자, Oracle BI 관리자
새 Oracle BI 12c 리포지토리 데이터베이스를 소스 데이터베이스와 재동기화합니다.	Amazon RDS Oracle BI 12c 리포지토리 데이터베이스를 온프레미스 데이터베이스와 재동기화합니다. 데이터베이스를 동기화하려면 Oracle Data Pump 새로 고침 또는 AWS DMS 변경 데이터 캡처(CDC) 를 사용할 수 있습니다.	Oracle BI 관리자, 데이터베이스 엔지니어 및 관리자
새로운 AWS 환경을 가리키도록 Oracle BI 12c URL을 전환합니다.	새로운 AWS 설치를 가리키도록 내부 DNS 서버의 Oracle BI 12c URL을 업데이트합니다.	마이그레이션 아키텍트, 솔루션 아키텍트, 애플리케이션 소유자, Oracle BI 관리자
새 환경을 모니터링합니다.	다음 도구 중 하나를 사용하여 새로운 Oracle BI 12c 환경을 모니터링합니다.	Oracle BI 관리자, 데이터베이스 엔지니어 및 관리자, 애플리케이션 관리자

작업	설명	필요한 기술
	<ul style="list-style-type: none"> • 아마존 CloudWatch • Amazon RDS 성능 개선 도우미 • Oracle Enterprise Manager 	
프로젝트를 승인 받습니다.	이해 관계자와 함께 테스트 결과를 검토하고 마이그레이션을 마무리하는 데 필요한 승인을 받습니다.	애플리케이션 소유자, 서비스 소유자, 클라우드 인프라 아키텍트, 마이그레이션 아키텍트, Oracle BI 아키텍트

관련 리소스

- [Oracle용 RDS에서 Oracle 리포지토리 생성 유틸리티 사용](#)(Amazon RDS 사용 설명서)
- [Amazon RDS 기반 Oracle](#)(Amazon RDS 사용 설명서)
- AWS 기반 [오라클 WebLogic 서버 12c \(AWS 백서\)](#)
- [고가용성을 위한 Oracle Business Intelligence 배포](#)(Oracle 도움말 센터)
- [Oracle Business Intelligence 애플리케이션 아카이브\(BAR\) 파일](#)(Oracle 도움말 센터)
- [환경 간에 OBI 12c를 마이그레이션하는 방법](#)(Oracle Support)

추가 정보

다음은 Oracle BI 12c를 AWS 클라우드로 마이그레이션하는 것과 관련된 모범 사례 목록입니다.

리포지토리 데이터베이스

Oracle BI 12c 데이터베이스 스키마를 Amazon RDS for Oracle 인스턴스에서 호스팅하는 것이 가장 좋습니다. 이 인스턴스 유형은 하드웨어 프로비저닝, 데이터베이스 설정, 패치 및 백업과 같은 관리 작업을 자동화하는 동시에 비용 효율적이고 크기 조정 가능한 용량을 제공합니다.

자세한 내용은 Amazon RDS 사용 설명서의 [Oracle용 RDS에서 Oracle 리포지토리 생성 유틸리티 사용](#)을 참조하세요.

웹 및 애플리케이션 티어

[메모리 최적화 Amazon EC2 인스턴스](#)는 대개 Oracle BI 12c 서버에 적합합니다. 어떤 인스턴스 유형을 선택하든 프로비저닝하는 인스턴스가 시스템의 메모리 사용 요구 사항을 충족하는지 확인합니다. 또한 Amazon EC2 인스턴스의 사용 가능한 메모리를 기반으로 [충분한 JVM \(WebLogic Java 가상 머신\) 힙 크기를 구성해야](#) 합니다.

로컬 스토리지

I/O는 Oracle BI 12c 애플리케이션의 전체 성능에서 중요한 역할을 합니다. Amazon Elastic Block Store(Amazon EBS)는 다양한 워크로드 패턴에 최적화된 다양한 스토리지 클래스를 제공합니다. 사용 사례에 맞는 Amazon EBS 볼륨 유형을 선택해야 합니다.

자세한 내용은 Amazon EBS 설명서의 [Amazon EBS 기능](#)을 참조하세요.

공유 스토리지

클러스터링된 Oracle BI 12c 도메인에는 다음 리소스에 대한 공유 스토리지가 필요합니다.

- 구성 파일
- Oracle BI 12c 싱글톤 데이터 디렉터리(SDD)
- Oracle 글로벌 캐시
- Oracle BI 스케줄러 스크립트
- 오라클 서버 바이너리 WebLogic

확장 가능한 완전 관리형 탄력적 네트워크 파일 시스템(NFS)을 제공하는 [Amazon EFS](#)를 사용하면 이러한 공유 스토리지 요구 사항을 충족할 수 있습니다.

공유 스토리지 성능 미세 조정

Amazon EFS에는 프로비저닝과 버스팅이라는 두 가지 [처리량 모드](#)가 있습니다. 또한 이 서비스에는 범용 및 최대 I/O라는 두 가지 [성능 모드](#)가 있습니다.

성능을 미세 조정하려면 먼저 범용 성능 모드와 프로비저닝 처리량 모드에서 워크로드를 테스트합니다. 이러한 테스트를 수행하면 해당 기본 모드가 원하는 서비스 수준을 충족하기에 충분한지 판단하는데 도움이 됩니다.

자세한 내용은 Amazon EFS 사용 설명서의 [Amazon EFS 성능](#)을 참조하세요.

가용성 및 재해 복구

가용 영역 장애 발생 시 리소스를 보호하기 위해서 여러 가용 영역에 걸쳐 Oracle BI 12c 구성 요소를 배포하는 것이 가장 좋습니다. 다음은 AWS 클라우드에서 호스팅되는 특정 Oracle BI 12c 리소스의 가용성 및 재해 복구 모범 사례 목록입니다.

- Oracle BI 12c 리포지토리 데이터베이스: 다중 AZ Amazon RDS 데이터베이스 인스턴스를 Oracle BI 12 리포지토리 데이터베이스에 배포합니다. 다중 AZ 배포에서 Amazon RDS는 자동으로 서로 다른 AZ에 동기식 예비 복제본을 프로비저닝하고 유지합니다. Oracle BI 12c 리포지토리 데이터베이스를 여러 가용 영역에 걸쳐 실행하면 계획된 시스템 유지 관리 중 가용성을 향상시킬 수 있으며, 인스턴스 및 가용 영역 중단으로부터 데이터베이스를 보호할 수 있습니다.
- Oracle BI 12c 관리 대상 서버: 내결함성을 확보하려면 여러 가용 영역에 걸쳐 구성된 Amazon EC2 Auto Scaling 그룹의 관리 대상 서버에 Oracle BI 12c 시스템 구성 요소를 배포하는 것이 가장 좋습니다. Auto Scaling은 [Amazon EC2 상태 확인](#)을 기반으로 결함이 있는 인스턴스를 대체합니다. 가용 영역에 장애가 발생하는 경우 Oracle HTTP Server는 계속해서 작동하는 가용 영역의 관리 대상 서버로 트래픽을 전달합니다. 그런 다음 Auto Scaling에서 호스트 수 요구 사항을 충족하기 위해 인스턴스를 시작합니다. 기존 세션이 정상적으로 작동하는 관리 대상 서버로 원활하게 장애 조치되도록 하려면 HTTP 세션 상태 복제를 활성화하는 것이 좋습니다.
- Oracle BI 12c 관리 서버: 관리 서버의 가용성을 높을 수준으로 유지하려면 여러 가용 영역에 걸쳐 있도록 구성된 Amazon EC2 Auto Scaling 그룹에서 관리 서버를 호스팅합니다. 그런 다음, 그룹의 최소 및 최대 크기를 1로 설정합니다. 가용 영역에 장애가 발생하는 경우 Amazon EC2 Auto Scaling은 대체 가용 영역에서 대체 관리 서버를 시작합니다. 동일한 가용 영역 내에서 장애가 발생한 기본 호스트를 복구하려면 [Amazon EC2 Auto Recovery](#)를 활성화합니다.
- 오라클 웹 티어 서버: 오라클 HTTP 서버를 오라클 WebLogic 서버 도메인과 연결하는 것이 가장 좋습니다.고가용성을 위해 Oracle HTTP Server를 여러 가용 영역을 포함하도록 구성된 Amazon EC2 Auto Scaling 그룹에 배포합니다. 그런 다음 서버를 탄력적 로드 밸런서(ELB) 뒤에 배치합니다. 호스트 장애에 대한 추가 보호를 제공하려면 Amazon EC2 Auto Recovery를 활성화합니다.

확장성

AWS 클라우드의 탄력성은 워크로드 요구 사항에 따라 애플리케이션을 수평 또는 수직으로 확장할 수 있도록 도와줍니다.

수직 크기 조정

애플리케이션을 수직으로 확장하려면 Oracle BI 12c 구성 요소를 실행하는 Amazon EC2 인스턴스의 크기와 유형을 변경합니다. 배포 초기에 인스턴스를 오버프로비저닝하여 불필요한 비용을 발생시킬 필요가 없습니다.

수평 크기 조정

Amazon EC2 Auto Scaling을 사용하면 워크로드 요구 사항에 따라 관리 대상 서버를 자동으로 추가 또는 제거하여 애플리케이션을 수평적으로 확장할 수 있습니다.

참고: Amazon EC2 Auto Scaling을 사용한 수평적 규모 조정을 구현하려면 스크립팅 기술과 철저한 테스트를 실행해야 합니다.

백업 및 복구

다음은 AWS 클라우드에서 호스팅되는 특정 Oracle BI 12c 리소스에 대한 백업 및 복구 모범 사례 목록입니다.

- Oracle Business Intelligence 메타데이터 리포지토리: Amazon RDS는 데이터베이스 인스턴스의 백업을 자동으로 생성하고 저장합니다. 이러한 백업은 지정된 기간 동안 보존됩니다. 데이터 보호 요구 사항에 따라 Amazon RDS 백업 기간 및 보존 설정을 구성해야 합니다. 자세한 내용은 [Amazon RDS 백업 및 복원](#)을 참조하세요.
- 관리 대상 서버, 관리 서버 및 웹 티어 서버: 데이터 보호 및 보존 요구 사항에 따라 [Amazon EBS 스냅샷](#)을 구성해야 합니다.
- 공유 스토리지: [AWS Backup](#)을 사용하여 Amazon EFS에 저장된 파일의 백업 및 복구를 관리할 수 있습니다. 또한 AWS Backup 서비스를 배포하여 Amazon EC2, Amazon EBS 및 Amazon RDS를 비롯한 다른 서비스의 백업 및 복구를 중앙에서 관리할 수 있습니다. 자세한 내용은 [AWS Backup이란?](#)을 AWS Backup 개발자 가이드에서.

보안 및 규정 준수

다음은 AWS 클라우드에서 Oracle BI 12c 애플리케이션을 보호하는 데 도움이 되는 보안 모범 사례 및 AWS 서비스 목록입니다.

- 저장 중 암호화: Amazon RDS, Amazon EFS 및 Amazon EBS는 모두 업계 표준 암호화 알고리즘을 지원합니다. [AWS Key Management Service\(AWS KMS\)](#)를 사용하여 암호화 키를 생성 및 관리하고, AWS 서비스 및 애플리케이션 전반에서 암호화 키 사용을 제어할 수 있습니다. 또한 Oracle BI 12c 리포지토리 데이터베이스를 호스팅하는 Amazon RDS for Oracle 데이터베이스 인스턴스에서 [Oracle 투명 데이터 암호화\(TDE\)](#)를 구성할 수 있습니다.
- 전송 중 암호화: SSL 또는 TLS 프로토콜을 활성화하여 Oracle BI 12c 설치의 다양한 계층 간에 전송되는 데이터를 보호하는 것이 가장 좋습니다. [AWS Certificate Manager\(ACM\)](#)를 사용하여 Oracle BI 12c 리소스를 위한 퍼블릭 및 프라이빗 SSL 및 TLS 인증서를 프로비저닝, 관리, 배포할 수 있습니다.
- 네트워크 보안: 사용 사례에 맞게 적절한 액세스 제어가 구성된 Amazon VPC에 Oracle BI 12c 리소스를 배포해야 합니다. 설치를 실행하는 Amazon EC2 인스턴스의 인바운드 및 아웃바운드 트래픽을

필터링하도록 보안 그룹을 구성합니다. 또한 정의된 규칙에 따라 트래픽을 허용하거나 거부하는 [네트워크 액세스 제어 목록\(NACL\)](#)을 구성해야 합니다.

- 모니터링 및 로깅: [AWS를 CloudTrail](#) 사용하여 Oracle BI 12c 리소스를 포함한 AWS 인프라에 대한 API 호출을 추적할 수 있습니다. 이 기능은 인프라 변경 사항을 추적하거나 보안 분석을 수행할 때 유용합니다. 또한 [CloudWatchAmazon](#)을 사용하여 Oracle BI 12c 애플리케이션의 성능 및 상태에 대한 실행 가능한 통찰력을 제공하는 운영 데이터를 볼 수 있습니다. 경보를 구성하고 해당 경보를 기반으로 자동화된 조치를 취할 수도 있습니다. Amazon RDS는 [확장 모니터링](#) 및 [성능 개선 도구 미](#)를 비롯한 추가 모니터링 도구를 제공합니다.

를 사용하여 온프레미스 Apache Kafka 클러스터를 Amazon MSK로 마이그레이션합니다. MirrorMaker

작성자: 한 장(AWS) 및 태너 프랫(AWS)

환경: PoC 또는 파일럿	소스: 온프레미스 또는 자체 관리형 Apache Kafka 클러스터	대상: Amazon Managed Streaming for Apache Kafka(Amazon MSK)
R 유형: 리플랫폼	워크로드: 오픈 소스, 기타 모든 워크로드	기술: 분석, 빅 데이터, 마이그레이션

AWS 서비스: Amazon MSK

요약

이 패턴은 온프레미스, 자체 관리형 또는 호스팅된 Apache Kafka 클러스터를 Amazon Managed Streaming for Apache Kafka(Amazon MSK)로 마이그레이션하기 위한 지침을 제공합니다. 또한 이 패턴을 사용하여 한 Amazon MSK 클러스터에서 다른 Amazon MSK 클러스터로 마이그레이션할 수 있습니다.

Apache Kafka에는 두 Kafka 클러스터 간에 데이터를 복제하는 MirrorMaker 기능이 포함되어 있습니다. MirrorMaker 소비자 그룹에 속하는 소비자 컬렉션으로 구성됩니다. 소비자가 소스 클러스터의 주제에서 데이터를 읽은 다음 이 데이터를 생산자에게 전달하고, 생산자는 대상 클러스터에 이 데이터를 씁니다.

Amazon MSK 설명서에는 MirrorMaker 버전 1.0을 사용하여 온프레미스 Kafka 클러스터를 Amazon MSK로 마이그레이션하는 프로세스에 대한 [개괄적인 개요](#)가 포함되어 있습니다. 이 패턴은 버전 2.0 사용에 대한 포괄적인 지침을 제공하여 이 정보를 보완합니다. step-by-step MirrorMaker

사전 조건 및 제한 사항

사전 조건

- 활성 상태의 AWS 계정
- 다음 중 하나인 Kafka 소스 클러스터입니다.
 - 온프레미스 데이터 센터 내

- 클라우드 내의 자체 관리형
- 파트너를 통해 호스팅됨

제한 사항

- MirrorMaker 버전 2.0을 사용하려면 소스 클러스터에서 Apache Kafka 버전 2.4.0 이상을 운영 중이어야 합니다. 이전 버전의 경우 [Amazon MSK 설명서의](#) 지침을 참조하여 MirrorMaker 버전 1.0을 사용하십시오.

제품 버전

- MirrorMaker 버전 2.0
- Apache Kafka 버전 2.4.0 이상. Amazon MSK가 지원하는 Apache Kafka 버전에 대한 자세한 내용은 [지원되는 Apache Kafka 버전](#)을 참조하세요.

아키텍처

소스 기술 스택

- 온프레미스 또는 자체 관리형 Kafka 클러스터

대상 기술 스택

- Amazon MSK 클러스터

대상 아키텍처

이 다이어그램은 다음 프로세스를 보여줍니다.

1. MirrorMaker 소스 Kafka 클러스터의 주제 및 소비자 그룹에서 데이터를 읽습니다.
2. MirrorMaker 데이터 및 소비자 정보를 대상 Amazon MSK 클러스터에 복제합니다.

도구

서비스

- [Amazon Elastic Compute Cloud\(Amazon EC2\)](#)는 AWS 클라우드에서 확장 가능한 컴퓨팅 용량을 제공합니다. 필요한 만큼 가상 서버를 시작하고 빠르게 스케일 업하거나 스케일 다운할 수 있습니다.
- [Amazon Managed Streaming for Apache Kafka\(Amazon MSK\)](#)는 Apache Kafka를 사용하여 스트리밍 데이터를 처리하는 애플리케이션의 구축 및 실행에 도움이 되는 완전 관리형 서비스입니다.
- [Amazon Virtual Private Cloud\(VPC\)](#)를 이용하면 사용자가 정의한 가상 네트워크로 AWS 리소스를 시작할 수 있습니다. 이 가상 네트워크는 사용자의 자체 데이터 센터에서 운영하는 기존 네트워크와 유사하며 AWS의 확장 가능한 인프라를 사용한다는 이점이 있습니다.

기타 도구

- [Apache Kafka](#)는 오픈 소스 이벤트 스트리밍 플랫폼입니다. 이 패턴에서는 Kafka의 [MirrorMaker](#)기능을 사용하여 클러스터 간 마이그레이션을 수행합니다.

모범 사례

소스 또는 대상 환경 MirrorMaker 모두에서 실행할 수 있지만 대상 클러스터와 최대한 가깝게 실행하는 것이 좋습니다. 자세한 내용은 Apache Kafka 설명서의 [모범 사례: 원격에서 소비, 로컬에 생산](#)을 참조하세요.

에픽

VPC를 생성하고 Amazon MSK 클러스터를 대상으로 지정

작업	설명	필요한 기술
VPC를 생성합니다.	<ol style="list-style-type: none"> 1. 대상 AWS 계정에서 VPC를 생성합니다. 자세한 내용은 VPC 생성을 참조하세요. 2. 새 VPC의 서로 다른 가용 영역에 있는 3개의 프라이빗 서브넷을 생성합니다. 자세한 내용은 서브넷 생성을 참조하세요. 서로 다른 가용 영역을 사용하여고가용성 및 내결함성을 제공할 수 있습니다. 	AWS 시스템 관리자, DevOps 엔지니어, 클라우드 관리자

작업	설명	필요한 기술
	<p>참고: 퍼블릭 인터넷 연결을 사용하여 Kafka 클러스터를 마이그레이션하는 경우, 퍼블릭 서브넷을 생성하고 Amazon MSK 클러스터에 대한 퍼블릭 액세스를 활성화하세요.</p>	
<p>Amazon MSK 클러스터를 생성합니다.</p>	<p>Amazon MSK 클러스터를 생성합니다. 지침은 AWS Management Console을 사용하여 클러스터 생성 또는 AWS CLI를 사용하여 클러스터 생성을 참조하세요. 이전에 생성한 VPC와 서브넷을 사용하도록 클러스터를 구성합니다.</p>	<p>AWS 시스템 관리자, DevOps 엔지니어, 클라우드 관리자</p>

설정 MirrorMaker

작업	설명	필요한 기술
<p>설치 MirrorMaker.</p>	<ol style="list-style-type: none"> 1. EC2 인스턴스를 시작합니다. 2. EC2 인스턴스에 연결합니다. 3. EC2 인스턴스에서 최신 Kafka 릴리스를 다운로드하고 압축을 풉니다. 지침은 빠른 시작 섹션(Kafka 설명서)을 참조하세요. <p>참고: 이 패턴에서는 MirrorMaker 2.0을 Amazon EC2</p>	<p>AWS 시스템 관리자, 클라우드 관리자, DevOps 엔지니어</p>

작업	설명	필요한 기술
	<p>MirrorMaker 인스턴스에 전용 클러스터로 설치합니다. 이 옵션은 개발 환경에 적합하며 이 패턴에서 사용되는 접근 방식입니다. MirrorMaker2.0의 다른 배포 옵션에 대한 자세한 내용은 이 패턴의 추가 정보 섹션을 참조하십시오.</p>	
<p>Kafka 클러스터 정보를 지정하세요.</p>	<p>Kafka 클라이언트 설치 bin 폴더에서 mm2.properties 파일을 생성하고 소스 Kafka 클러스터에 맞게 구성합니다. 지침은 전용 MirrorMaker 클러스터 실행 (Kafka 설명서) 을 참조하십시오.</p>	<p>AWS 시스템 관리자, 클라우드 관리자, DevOps 엔지니어</p>
<p>시작 MirrorMaker.</p>	<p>다음 명령을 입력하여 mm2.properties 파일을 MirrorMaker 시작하고 전달합니다.</p> <pre style="border: 1px solid #ccc; border-radius: 10px; padding: 10px; margin: 10px 0;">\$./bin/connect-mirror-maker.sh mm2.properties</pre>	<p>AWS 시스템 관리자, 클라우드 관리자, DevOps 엔지니어</p>
<p>진행 상태를 모니터링합니다.</p>	<p>각 주제에 대한 마지막 오프셋과 해당 주제의 현재 소비되는 오프셋 사이의 지연을 검사하여 진행 상황을 확인하십시오. MirrorMaker 지침은 Kafka 설명서의 지리적 복제 모니터링을 참조하세요.</p>	<p>AWS 시스템 관리자, 클라우드 관리자, DevOps 엔지니어</p>

전환

작업	설명	필요한 기술
소비자 애플리케이션을 중지합니다.	소스 클러스터의 데이터를 소비하는 모든 소비자 애플리케이션을 중지하십시오.	앱 개발자
소비자 애플리케이션을 시작합니다.	대상 클러스터를 가리키도록 애플리케이션 부트스트랩 구성을 변경합니다. 그런 다음 대상 클러스터에서 소비를 시작합니다.	앱 개발자
소스 클러스터에서 생산자를 중지합니다.	소비자 애플리케이션이 대상 클러스터에서 성공적으로 소비되면 소스 클러스터의 생산자를 중지합니다.	앱 개발자
대상 클러스터에서 생산자를 시작합니다.	생산자의 구성 부트스트랩 서버를 변경하고 대상 클러스터를 가리킵니다. MirrorMaker 생산자를 시작하기 전에 소스 클러스터의 모든 데이터 미러링이 완료될 때까지 기다리십시오.	앱 개발자
그만해 MirrorMaker.	생산자가 대상 클러스터로 이동한 후에는 MirrorMaker 중단하십시오.	AWS 시스템 관리자, 클라우드 관리자, DevOps 엔지니어

관련 리소스

AWS 리소스

- [MirrorMaker\(Amazon MSK 설명서\)](#) 를 사용하여 클러스터를 마이그레이션하는 중
- [Amazon MSK 마이그레이션 랩](#) (AWS 워크숍 스튜디오)

기타 리소스

- [MirrorMaker 2.0](#) (아파치 카프카 개선 제안)
- [지리적 복제: 클러스터 간 데이터 미러링](#) (Apache Kafka 설명서)

추가 정보

이 패턴은 MirrorMaker 2.0을 Amazon EC2에서 전용 MirrorMaker 클러스터로 실행합니다. 이 옵션은 개발 환경에 적합합니다. 이 패턴에서는 설명하지 않지만 Kafka Connect 클러스터에서 MirrorMaker 2.0을 실행할 수도 있습니다. 이 배포 옵션은 크기 조정 및 유지 관리를 개선하는 Kafka 에코시스템 내의 프레임워크를 사용합니다. 애플리케이션을 실행하기 위한 관련 구성을 사용하여 커넥터를 Kafka Connect 클러스터에 배포합니다. 커넥터는 개발 또는 테스트를 위해 독립 실행형 모드로 실행하거나 프로덕션용 분산 모드에서 실행할 수 있습니다. 자세한 내용은 [Connect MirrorMaker 클러스터에서 실행](#) (Apache Kafka 설명서) 을 참조하십시오. 다른 MirrorMaker 2.0 배포 옵션에 대한 자세한 내용은 안내: [MirrorMaker 2.0 실행 \(Kafka 설명서\)](#) 을 참조하십시오.

ELK 스택을 AWS 기반 Elastic 클라우드로 마이그레이션

작성자: Battulga Purevragchaa(AWS), uday reddy, Antony Prasad Thevaraj(AWS)

환경: 프로덕션	출처 : Elasticsearch	대상: Elastic 클라우드
R 유형: 리플랫폼	워크로드: 기타 모든 워크로드	기술: 분석, 보안, 자격 증명, 규정 준수

AWS 서비스: Amazon EC2,
Amazon EC2 Auto Scaling,
Elastic Load Balancing(ELB),
Amazon S3, Amazon Route
53

요약

[Elastic](#)은 수년 동안 서비스를 제공해 왔으며, 사용자와 고객은 일반적으로 온프레미스에서 Elastic을 직접 관리합니다. 관리형 [Elasticsearch 서비스](#)인 [Elastic Cloud](#)는 [엔터프라이즈 검색](#), [관찰성](#) 및 [보안](#)을 위한 Elastic 스택(ELK 스택)과 솔루션을 사용하는 방법을 제공합니다. 로그, 메트릭, APM(애플리케이션 성능 모니터링), SIEM(보안 정보 및 이벤트 관리)과 같은 앱을 통해 Elastic 솔루션에 액세스할 수 있습니다. 기계 학습, 인덱스 수명 주기 관리, Kibana 렌즈(드래그 앤 드롭 시각화용)와 같은 통합 기능을 사용할 수 있습니다.

자체 관리형 Elasticsearch에서 Elastic 클라우드로 전환하면 Elasticsearch 서비스가 다음 작업을 처리합니다.

- 기본 인프라 프로비저닝 및 관리
- Elasticsearch 클러스터 생성 및 관리
- 클러스터 확장 및 축소
- 업그레이드, 패치 적용, 스냅샷 촬영

이를 통해 다른 문제를 해결하는 데 더 많은 시간을 할애할 수 있습니다.

이 패턴은 온프레미스 Elasticsearch 7.13을 Amazon Web Services(AWS)의 Elastic 클라우드에서 Elasticsearch로 마이그레이션하는 방법을 정의합니다. 다른 버전에서는 이 패턴에 설명된 프로세스를 약간 수정해야 할 수 있습니다. 자세한 내용은 Elastic 담당자에게 문의하십시오.

사전 조건 및 제한 사항

사전 조건

- 스냅샷을 위해 [Amazon Simple Storage Service\(S3\)](#)에 액세스할 수 있는 활성 [AWS 계정](#)
- 스냅샷 데이터 파일을 Amazon S3로 복사하기 위한 안전하고 충분히 높은 대역폭의 [프라이빗 링크](#)
- [Amazon S3 전송 가속화](#)
- 데이터 모으기가 충분히 큰 로컬 데이터 스토어 또는 원격 스토리지(Amazon S3)에 정기적으로 보관되도록 하는 [Elastic 스냅샷 정책](#)

마이그레이션을 시작하기 전에 스냅샷의 크기와 함께 제공되는 인덱스의 [수명 주기 정책](#)이 온프레미스에 얼마나 큰지 이해해야 합니다. 자세한 내용은 [Elastic에 문의하십시오](#).

역할 및 기술

마이그레이션 프로세스에는 다음 표에 설명된 역할과 전문 지식도 필요합니다.

역할	전문성	책임
앱 지원	Elastic 클라우드 및 Elastic 온프레미스에 대한 지식	모든 Elastic 관련 작업
시스템 관리자 또는 DBA	온프레미스 Elastic 환경 및 해당 구성에 대한 심층 지식	스토리지를 프로비저닝하고 AWS Command Line Interface (AWS CLI)를 설치 및 사용하고 온프레미스에서 Elastic을 지원하는 모든 데이터 소스를 식별하는 기능
네트워크 관리자	온프레미스에서 AWS로의 네트워크 연결, 보안 및 성능에 대한 지식	연결 대역폭에 대한 이해를 바탕으로 온프레미스에서 Amazon S3로 네트워크 링크 설정

제한 사항

- Elastic 클라우드 기반 Elasticsearch는 [지원되는 AWS 리전](#)에서만 사용할 수 있습니다(2021년 9월).

제품 버전

- Elasticsearch 7.13

아키텍처

소스 기술 스택

온프레미스 Elasticsearch 7.13 이상

- 클러스터 스냅샷
- 인덱스 스냅샷
- [Beats](#) 구성

소스 기술 아키텍처

다음 다이어그램은 다양한 수집 방법, 노드 유형 및 Kibana를 사용하는 일반적인 온프레미스 아키텍처를 보여줍니다. 다양한 노드 유형은 Elasticsearch 클러스터, 인증 및 시각화 역할을 반영합니다.

1. Beats에서 Logstash로 수집
2. Beats에서 Apache Kafka 메시지 대기열로 수집
3. Filebeat에서 Logstash로 수집
4. Apache Kafka 메시지 대기열에서 Logstash로 수집
5. Logstash에서 Elasticsearch 클러스터로 수집
6. Elasticsearch 클러스터
7. 인증 및 알림 노드
8. Kibana 및 블럽 노드

대상 기술 스택

Elastic 클라우드는 클러스터 간 복제를 통해 여러 AWS 리전의 서비스형 소프트웨어(SaaS) 계정에 배포됩니다.

- 클러스터 스냅샷
- 인덱스 스냅샷
- Beats 구성
- Elastic 클라우드
- Network Load Balancer
- Amazon Route 53
- Amazon S3

대상 아키텍처

관리형 Elastic 클라우드 인프라는 다음과 같습니다.

- 여러 [가용 영역](#) 및 여러 AWS 리전에 존재하므로 가용성이 높습니다.
- Elastic 클라우드 [클러스터 간 복제 \(CCR\)](#)를 사용하여 데이터(인덱스 및 스냅샷)가 복제되므로 리전 내결함성이 있습니다.
- 아카이브, 스냅샷이 [Amazon S3](#)에 보관되기 때문
- [Network Load Balancers](#) 및 [Route 53](#)의 조합을 통한 네트워크 파티션 내성
- [Elastic APM](#), [Beats](#), [Logstash](#)에서 시작되지만 이에 국한되지 않는 데이터 모으기

상위 수준 마이그레이션 단계

Elastic은 온프레미스 Elastic 클러스터를 Elastic 클라우드로 마이그레이션하기 위한 자체 규범적 방법론을 개발했습니다. Elastic 방법론은 [Well-Architected Framework](#) 및 [AWS Migration Acceleration Program](#)(MAP)을 비롯한 AWS 마이그레이션 지침 및 모범 사례와 직접 연계되고 보완됩니다. 일반적으로 세 가지 AWS 마이그레이션 단계는 다음과 같습니다.

- 평가
- 동원
- 마이그레이션 및 현대화

Elastic은 유사한 마이그레이션 단계를 따르며 상호 보완적인 용어를 사용합니다.

- 시작
- 계획
- 구현
- 전송
- Close

Elastic은 Elastic 구현 방법론을 사용하여 프로젝트 결과 전송을 촉진합니다. 이는 Elastic, 컨설팅 팀, 고객 팀이 명확하게 협력하여 의도한 결과를 공동으로 제공할 수 있도록 설계된 것입니다.

Elastic 방법론은 구현 단계 내에서 기존의 워터폴 단계를 스크럼과 결합합니다. 기술 요구 사항 구성은 위험을 최소화하면서 협업 방식으로 반복적으로 제공됩니다.

도구

서비스

- [Amazon Route 53](#) - Amazon Route 53은 가용성과 확장성이 뛰어난 도메인 이름 시스템(DNS) 웹 서비스입니다. Route 53을 사용하여 세 가지 주요 기능, 즉 도메인 등록, DNS 라우팅, 상태 확인을 조합하여 실행할 수 있습니다.
- [Amazon S3](#)-Amazon Simple Storage Service(S3)는 객체 스토리지 서비스입니다. Amazon S3를 사용하면 인터넷을 통해 언제 어디서든 원하는 양의 데이터를 저장하고 검색할 수 있습니다. 이 패턴은 S3 버킷과 [Amazon S3 전송 가속화](#)를 사용합니다.
- [Elastic Load Balancing](#) - Elastic Load Balancing은 둘 이상의 가용 영역에서 EC2 인스턴스, 컨테이너, IP 주소 등 여러 대상에 걸쳐 수신되는 트래픽을 자동으로 분산합니다.

기타 도구

- [Beats](#) - Beats는 Logstash 또는 Elasticsearch의 데이터를 전송합니다.
- [Elastic 클라우드](#) - Elastic 클라우드는 Elasticsearch를 호스팅하기 위한 관리형 서비스입니다.
- [Elasticsearch](#) - Elasticsearch는 Elastic 스택을 사용하여 규모 조정 가능한 검색 및 분석을 위해 데이터를 중앙에 저장하는 검색 및 분석 엔진입니다. 또한 이 패턴은 스냅샷 생성과 클러스터 간 복제를 사용합니다.

- [Logstash](#) — Logstash는 여러 소스에서 데이터를 수집하고 변환한 다음 데이터 스토리지로 전송하는 서버 측 데이터 처리 파이프라인입니다.

에픽

마이그레이션 준비

작업	설명	필요한 기술
온프레미스 Elastic 솔루션을 실행하는 서버를 식별하십시오.	Elastic 마이그레이션이 지원되는지 확인합니다.	앱 소유자
온프레미스 서버 구성을 이해합니다.	온프레미스에서 워크로드를 성공적으로 구동하는 데 필요한 서버 구성을 이해하려면 현재 사용 중인 서버 하드웨어 설치 공간, 네트워크 구성 및 스토리지 특성을 찾아보십시오.	앱 지원
사용자 및 앱 계정 정보를 수집합니다.	온프레미스 Elastic 환경에서 사용되는 사용자 이름과 앱 이름을 식별하십시오.	시스템 관리자, 앱 지원
Beats 및 데이터 발송자 구성을 문서화하십시오.	구성을 문서화하려면 기존 데이터 소스와 싱크를 살펴보십시오. 자세한 내용은 Elastic 설명서 를 참조하십시오.	앱 지원
데이터의 속도와 양을 결정하십시오.	클러스터가 처리하는 데이터의 양에 대한 기준을 설정합니다.	시스템 관리자, 앱 지원
RPO 및 RTO 시나리오를 문서화하십시오.	운영 중단 및 서비스 수준에 관한 계약(SLA)의 측면에서 Recovery Point Objective (RPO) 및 Recovery Time Objective(RTO) 시나리오를 문서화합니다.	앱 소유자, 시스템 관리자, 앱 지원

작업	설명	필요한 기술
최적의 스냅샷 수명 주기 설정을 결정하십시오.	마이그레이션 도중 및 이후에 Elastic 스냅샷을 사용하여 데이터를 보호해야 하는 빈도를 정의하십시오.	앱 소유자, 시스템 관리자, 앱 지원
마이그레이션 후 성능 기대치를 정의하십시오.	현재 및 예상 화면 새로 고침, 쿼리 런타임, 사용자 인터페이스 동작에 대한 지표를 생성합니다.	시스템 관리자, 앱 지원
인터넷 액세스 전송, 대역폭 및 가용성 요구 사항을 문서화합니다.	스냅샷을 Amazon S3로 복사하기 위한 인터넷 연결의 속도, 지연 시간 및 복원력을 확인합니다.	네트워크 관리자
Elastic 온프레미스 런타임의 현재 비용을 문서화합니다.	AWS 대상 환경의 규모가 고성능이면서 비용 효율적으로 설계되었는지 확인합니다.	DBA, 시스템 관리자, 앱 지원
인증 및 권한 부여 요구 사항 파악	Elastic 스택 보안 기능은 경량 디렉터리 액세스 프로토콜(LDAP), Security Assertion Markup Language(SAML), OpenID Connect(OIDC)와 같은 내장 영역을 제공합니다.	DBA, 시스템 관리자, 앱 지원
지리적 위치에 기반한 특정 규제 요구 사항을 이해하십시오.	요구 사항 및 관련 국가 요구 사항에 따라 데이터를 내보내고 암호화해야 합니다.	DBA, 시스템 관리자, 앱 지원

마이그레이션 구현

작업	설명	필요한 기술
<p>Amazon S3에서 스테이징 역할을 준비합니다.</p>	<p>Amazon S3 에서 스냅샷을 수신하려면 새로 생성한 버킷에 대한 전체 액세스 권한을 가진 S3 버킷 생성 및 임시 AWS Identity and Access Management(IAM) 역할을 생성합니다. 자세한 내용은 IAM 사용자에게 권한을 위임하기 위한 역할 생성을 참조하십시오. AWS 보안 토큰 서비스를 사용하여 임시 보안 보안 인증을 요청합니다. 액세스 키 ID, 비밀 액세스 키 및 세션 토큰을 안전하게 유지합니다.</p> <p>버킷에서 Amazon S3 전송 가속화 활성화</p>	<p>AWS 관리자</p>
<p>AWS CLI와 Amazon S3 플러그인을 온프레미스에 설치합니다.</p>	<p>각 Elasticsearch 노드에서 다음 명령을 실행합니다.</p> <pre data-bbox="597 1276 1027 1438">sudo bin/elasticsearch-plugin install repository-s3</pre> <p>그런 다음 노드를 재부팅합니다.</p>	<p>AWS 관리자</p>
<p>Amazon S3 클라이언트 액세스를 구성합니다.</p>	<p>다음 명령을 실행하여 이전에 생성한 키를 추가합니다.</p> <pre data-bbox="597 1724 1027 1885">elasticsearch-keystore add s3.client.default.access_key</pre>	<p>AWS 관리자</p>

작업	설명	필요한 기술
	<pre>elasticsearch-keystore add s3.client.default.secret_key</pre> <pre>elasticsearch-keystore add s3.client.default.session_token</pre>	
<p>Elastic 데이터용 스냅샷 리포지토리를 등록하십시오.</p>	<p>Kibana 개발 도구를 사용하여 온프레미스 로컬 클러스터에 어느 원격 S3 버킷에 쓸 것인지 알립니다.</p>	<p>AWS 관리자</p>
<p>스냅샷 정책을 구성합니다.</p>	<p>스냅샷 수명 주기 관리를 구성하려면 Kibana Policies 탭에서 SLM 정책을 선택하고, 포함해야 하는 시간, 데이터 스트림 또는 인덱스 및 사용할 이름을 정의합니다.</p> <p>스냅샷을 자주 찍는 정책을 구성하십시오. 스냅샷은 증분식이며 스토리지를 효율적으로 사용합니다. 준비 상태 평가 결정과 일치시키십시오. 또한 정책을 통해 보존 정책을 지정하고 스냅샷이 더 이상 필요하지 않은 경우 스냅샷을 자동으로 삭제할 수 있습니다.</p>	<p>앱 지원</p>
<p>스냅샷이 작동하는지 확인합니다.</p>	<p>Kibana 개발 도구에서 다음 명령을 실행합니다.</p> <pre>GET _snapshot/<your_repo_name>/_all</pre>	<p>AWS 관리자, 앱 지원</p>

작업	설명	필요한 기술
Elastic 클라우드에 새 클러스터를 배포하십시오.	Elastic에 로그인 하고 준비 상태 평가의 비즈니스 결과에서 도출한 ‘관찰성, 검색 또는 보안’을 위한 클러스터를 선택하십시오.	AWS 관리자, 앱 지원
클러스터 키 스토어 액세스를 설정합니다.	새 클러스터는 스냅샷을 저장할 S3 버킷에 대한 액세스 권한이 필요합니다. Elasticsearch 서비스 콘솔에서 보안을 선택하고 이전에 생성한 액세스 및 비밀 IAM 키를 입력합니다.	AWS 관리자
Amazon S3 액세스할 Elastic 클라우드 호스트 클러스터를 구성합니다.	<p>Amazon S3에서 이전에 생성한 스냅샷 리포지토리에 대한 새 클러스터 액세스를 설정합니다. Kibana를 사용하여 다음을 수행합니다.</p> <ol style="list-style-type: none"> 1. 스택 관리, 스냅샷 설정, 을 선택합니다 RegisterRepo. 2. 별칭 필드에 리포지토리 이름을 입력합니다. 3. S3 클라이언트 이름의 경우 보조를 선택합니다. 4. 이전에 생성한 S3 버킷을 저장소에 추가합니다. 5. 스냅샷 압축을 선택합니다. 6. 암호화 설정의 경우 기본값을 유지합니다. 	AWS 관리자, 앱 지원

작업	설명	필요한 기술
새로운 Amazon S3 리포지토리를 확인합니다.	Elastic 클라우드 클러스터에 호스팅된 새 리포지토리에 액세스할 수 있는지 확인하십시오.	AWS 관리자
Elasticsearch 서비스 클러스터를 초기화하십시오.	Elasticsearch 서비스 콘솔에서 S3 스냅샷의 Elasticsearch 서비스 클러스터를 초기화합니다. 다음 명령을 POST로 실행합니다. <pre>*/_close?expand_wildcards=all</pre> <pre>/_snapshot/<your-repo-name>/<your-snapshot-name>/_restore</pre> <pre>*/_open?expand_wildcards=all</pre>	앱 지원

마이그레이션 완료

작업	설명	필요한 기술
스냅샷 복원이 성공했는지 확인하십시오.	Kibana 개발 도구를 사용해서 다음 명령을 실행합니다. <pre>GET _cat/indices</pre>	앱 지원
통합 서비스를 재배포하십시오.	Beats 및 Logstash의 엔드포인트를 새로운 Elasticsearch 서	앱 지원

작업	설명	필요한 기술
	비스 엔드포인트에 연결합니다.	

클러스터 환경 테스트 및 정리

작업	설명	필요한 기술
클러스터 환경을 검증하십시오.	온프레미스 Elastic 클러스터 환경을 AWS로 마이그레이션 한 후에는 해당 환경에 연결하여 자체 사용자 승인 테스트 (UAT) 도구를 사용하여 새 환경을 검증할 수 있습니다.	앱 지원
리소스를 정리합니다.	클러스터가 성공적으로 마이그레이션되었는지 확인한 후 마이그레이션에 사용된 S3 버킷과 IAM 역할을 제거합니다.	AWS 관리자

관련 리소스

Elastic 참조

- [Elastic 클라우드](#)
- [AWS 기반 관리형 Elasticsearch와 Kibana](#)
- [Elastic 엔터프라이즈 검색](#)
- [Elastic 통합](#)
- [Elastic 관찰성](#)
- [Elastic 보안](#)
- [Beats](#)
- [Elastic APM](#)
- [인덱스 수명 주기 관리로 마이그레이션](#)

- [Elastic 구독](#)
- [Elastic에 문의](#)

Elastic 블로그 포스트

- [자체 관리형 Elasticsearch에서 AWS 기반 Elastic 클라우드로 마이그레이션하는 방법](#)(블로그 게시물)
- [Elastic 클라우드로 마이그레이션](#)(블로그 게시물)

Elastic 설명서

- [튜토리얼: SLM을 통한 백업 자동화](#)
- [ILM: 인덱스 수명 주기 관리](#)
- [Logstash](#)
- [클러스터 간 복제\(CCR\)](#)
- [파이프라인 수집](#)
- [Elasticsearch API 요청 실행](#)
- [스냅샷 보존](#)

Elastic 비디오 및 웨비나

- [Elastic 클라우드 마이그레이션](#)
- [Elastic Cloud: 고객이 마이그레이션하는 이유](#)(웨비나)

AWS 참조

- [AWS Marketplace 기반 Elastic 클라우드](#)
- [AWS 명령줄 인터페이스](#)
- [AWS Direct Connect](#)
- [AWS Migration Acceleration Program](#)
- [Network Load Balancers](#)
- [리전 및 가용 영역](#)
- [Amazon Route 53](#)

- [Amazon Simple Storage Service\(S3\)](#)
- [Amazon S3 전송 가속화](#)
- [VPN 연결](#)
- [Well-Architected 프레임워크](#)

추가 정보

복잡한 워크로드를 마이그레이션할 계획이라면 [Elastic 컨설팅 서비스](#)에 문의하십시오. 구성 및 서비스와 관련된 기본적인 질문이 있는 경우 [Elastic 지원](#) 팀에 문의하십시오.

Starburst를 사용하여 데이터를 AWS 클라우드로 마이그레이션하십시오.

제작: 안토니 프라사드 테바라즈 (AWS), 손 밴 스타든 (Starburst), 수레쉬 베라고니 (AWS)

환경: 프로덕션

기술: 분석, 데이터 레이크, 데이터베이스

워크로드: 기타 모든 워크로드

AWS 서비스: Amazon EKS

요약

Starburst는 기존 데이터 소스를 단일 액세스 포인트로 통합하는 엔터프라이즈 쿼리 엔진을 제공하여 Amazon Web Services(AWS)로의 데이터 마이그레이션 여정을 가속화하도록 지원합니다. 마이그레이션 계획을 확정하기 전에 여러 데이터 소스에 대한 분석을 실행하여 귀중한 통찰력을 얻을 수 있습니다. business-as-usual 분석을 중단하지 않고 Starburst 엔진 또는 전용 ETL (추출, 변환 및 로드) 애플리케이션을 사용하여 데이터를 마이그레이션할 수 있습니다.

사전 조건 및 제한 사항

사전 조건

- 활성 상태의 AWS 계정.
- Virtual Private Cloud(VPC).
- Amazon Elastic Kubernetes Service(Amazon EKS): 클러스터
- Amazon Elastic Compute Cloud(Amazon EC2) 오토 스케일링
- 마이그레이션해야 하는 현재 시스템 워크로드 목록
- AWS에서 온프레미스 환경으로의 네트워크 연결

아키텍처

참조 아키텍처

다음 상위 아키텍처 다이어그램은 AWS 클라우드에서의 Starburst Enterprise의 일반적인 배포를 보여줍니다.

1. Starburst 엔터프라이즈 클러스터는 AWS 계정 내에서 실행됩니다.
2. 사용자는 경량 디렉터리 액세스 프로토콜(LDAP) 또는 공개 인증(OAuth)을 사용하여 인증하고 Starburst 클러스터와 직접 상호 작용합니다.
3. Starburst는 AWS Glue, Amazon Simple Storage Service(S3), Amazon Relational Database Service(RDS), Amazon Redshift와 같은 여러 AWS 데이터 소스에 연결할 수 있습니다. Starburst는 AWS 클라우드, 온프레미스 또는 기타 클라우드 환경의 데이터 소스 전반에 걸쳐 페더레이션된 쿼리 기능을 제공합니다.
4. 차트 Helm을 사용하여 Amazon EKS 클러스터에서 Starburst 엔터프라이즈를 시작할 수 있습니다.
5. Starburst 엔터프라이즈는 Amazon EC2 Auto Scaling 그룹과 Amazon EC2 스팟 인스턴스를 사용하여 인프라를 최적화합니다.
6. Starburst 엔터프라이즈는 기존 온프레미스 데이터 소스에 직접 연결하여 데이터를 실시간으로 읽습니다. 또한 이 환경에 Starburst Enterprise가 이미 배포되어 있는 경우, AWS 클라우드의 새 Starburst 클러스터를 이 기존 클러스터에 직접 연결할 수 있습니다.

다음 사항에 유의하십시오.

- Starburst는 데이터 가상화 플랫폼이 아닙니다. 분석을 위한 전체 데이터 메시 전략의 기초를 형성하는 SQL 기반 대량 병렬 처리(MPP) 쿼리 엔진입니다.
- 마이그레이션의 일환으로 Starburst를 배포하면 기존 온프레미스 인프라에 직접 연결됩니다.
- Starburst는 다양한 레거시 시스템과의 연결을 용이하게 하는 몇 가지 내장형 엔터프라이즈 및 오픈 소스 커넥터를 제공합니다. 커넥터 및 해당 기능의 전체 목록은 Starburst Enterprise 사용 설명서의 [커넥터를](#) 참조하십시오.
- Starburst는 온프레미스 데이터 소스에서 실시간으로 데이터를 쿼리할 수 있습니다. 이를 통해 데이터를 마이그레이션하는 동안 정기적인 비즈니스 운영이 중단되는 것을 방지할 수 있습니다.
- 기존 온프레미스 Starburst Enterprise 배포에서 마이그레이션하는 경우 특수 커넥터인 Starburst Stargate를 사용하여 AWS의 Starburst Enterprise 클러스터를 온프레미스 클러스터에 직접 연결할 수 있습니다. 이는 비즈니스 사용자와 데이터 분석가가 AWS 클라우드에서 온프레미스 환경으로 쿼리를 페더레이션할 때 추가적인 성능 이점을 제공합니다.

높은 수준의 프로세스 개요

Starburst를 사용하면 데이터를 마이그레이션하기 전에 모든 데이터에 대한 통찰력을 확보할 수 있으므로 Starburst를 사용하면 데이터 마이그레이션 프로젝트를 가속화할 수 있습니다. 다음 이미지는 Starburst를 사용하여 데이터를 마이그레이션하는 일반적인 프로세스를 보여줍니다.

역할

Starburst를 사용하여 마이그레이션을 완료하려면 일반적으로 다음과 같은 역할이 필요합니다.

- 클라우드 관리자 — Starburst Enterprise 애플리케이션을 실행하는 데 클라우드 리소스를 사용할 수 있도록 하는 일을 담당합니다.
- Starburst 관리자 — Starburst 애플리케이션 설치, 구성, 관리 및 지원을 담당합니다.
- 데이터 엔지니어— 담당:
 - 레거시 데이터를 클라우드로 마이그레이션
 - 분석을 지원하는 시맨틱 뷰 구축
- 솔루션 또는 시스템 소유자 — 전체 솔루션 구현을 담당합니다.

도구

서비스

- [Amazon EC2](#) – Amazon Elastic Compute Cloud(Amazon EC2)는 AWS 클라우드에서 확장 가능한 컴퓨팅 용량을 제공합니다.
- [Amazon EKS](#) – Amazon Elastic Kubernetes Service(Amazon EKS)는 자체 Kubernetes 컨트롤 플레인을 구축하거나 유지 관리할 필요 없이 AWS에서 Kubernetes를 실행하기 위한 관리형 서비스입니다. Kubernetes는 컨테이너화된 애플리케이션의 배포, 조정 및 관리 자동화를 위한 오픈 소스 시스템입니다.

기타 도구

- [Helm](#) – Kubernetes용 Helm 패키지 관리자는 Kubernetes 클러스터에서 애플리케이션을 설치하고 관리하는 데 도움이 됩니다.
- [Starburst 엔터프라이즈](#) — Starburst Enterprise는 분석을 위한 전체 데이터 메시 전략의 기초를 형성하는 SQL 기반 대량 병렬 처리(MPP) 쿼리 엔진입니다.

- [Starburst 스타게이트](#) — Starburst Stargate는 한 Starburst Enterprise 환경(예: 온프레미스 데이터 센터의 클러스터)에 있는 카탈로그 및 데이터 소스를 다른 Starburst Enterprise 환경(예: AWS 클라우드의 클러스터)에 있는 카탈로그 및 데이터 소스에 연결합니다.

에픽

데이터 평가

작업	설명	필요한 기술
데이터를 식별하고 우선순위를 정하세요.	이동하려는 데이터를 식별하세요. 대규모 온프레미스 레거시 시스템에는 마이그레이션하려는 핵심 데이터와 함께 이동하고 싶지 않거나 규정 준수상의 이유로 이동할 수 없는 데이터가 포함될 수 있습니다. 데이터 인벤토리로 시작하면 먼저 대상으로 삼아야 하는 데이터의 우선 순위를 정하는 데 도움이 됩니다. 자세한 내용은 자동화된 포트폴리오 검색 시작 을 참조하세요.	데이터 엔지니어, DBA
데이터를 탐색하고, 인벤토리를 작성하고, 백업하세요.	사용 사례에 맞는 데이터의 품질, 수량, 관련성을 검증하세요. 필요에 따라 데이터를 백업하거나 스냅샷을 만들고 데이터의 대상 환경을 확정합니다.	데이터 엔지니어, DBA

Starburst 엔터프라이즈 환경 설정

작업	설명	필요한 기술
AWS 클라우드에서 Starburst 엔터프라이즈를 구성합니다.	데이터를 카탈로그화하는 동안 관리형 Amazon EKS 클러스터	AWS 관리자, 앱 개발자

작업	설명	필요한 기술
	에 Starburst 엔터프라이즈를 설정하십시오. 자세한 내용은 Starburst 엔터프라이즈 참조 문서의 Kubernetes를 사용한 배포 를 참조하십시오. 이를 통해 데이터 마이그레이션이 진행되는 동안에도 business-as-usual 분석이 가능합니다.	
Starburst를 데이터 소스에 연결합니다.	데이터를 식별하고 Starburst 엔터프라이즈를 설정한 후 Starburst를 데이터 소스에 연결합니다. Starburst는 데이터 소스에서 SQL 쿼리로 직접 데이터를 읽습니다. 자세한 내용은 Starburst 엔터프라이즈 참조 문서 를 참조하세요.	AWS 관리자, 앱 개발자

데이터 마이그레이션

작업	설명	필요한 기술
ETL 파이프라인을 빌드하고 실행하세요.	데이터 마이그레이션 프로세스를 시작합니다. 이 활동은 business-as-usual 분석과 동시에 발생할 수 있습니다. 마이그레이션에는 타사 제품 또는 Starburst를 사용할 수 있습니다. Starburst는 다양한 소스에서 데이터를 읽고 쓸 수 있는 기능을 모두 갖추고 있습니다. 자세한 내용은 Starburst 엔터프라이즈 참조 문서 를 참조하세요.	데이터 엔지니어

작업	설명	필요한 기술
데이터를 검증합니다.	데이터를 마이그레이션한 후 데이터를 검증하여 필요한 모든 데이터가 이동되었고 손상되지 않았는지 확인합니다.	데이터 엔지니어, DevOps 엔지니어

컷오버 및 롤아웃

작업	설명	필요한 기술
데이터를 잘라내세요.	데이터 마이그레이션 및 검증이 완료된 후 데이터를 잘라낼 수 있습니다. 여기에는 Starburst의 데이터 연결 링크 변경이 포함됩니다. 온프레미스 소스를 가리키는 대신 새 클라우드 소스를 가리키고 시맨틱 뷰를 업데이트합니다. 자세한 내용은 Starburst Enterprise 참조 문서의 커넥터를 참조하십시오.	데이터 엔지니어, 전환 리드
사용자 대상으로 출시하세요.	데이터 소비자는 마이그레이션된 데이터 원본으로 작업을 시작합니다. 분석 최종 사용자에게는 이 프로세스가 보이지 않습니다.	전환 리드, 데이터 엔지니어

관련 리소스

AWS Marketplace

- [Starburst 갤러리](#)
- [Starburst 엔터프라이즈](#)
- [스타버스트 데이터 JumpStart](#)

- [Starburst 엔터프라이즈 \(그래비톤 포함\)](#)

Starburst 문서

- [Starburst 엔터프라이즈 사용 설명서](#)
- [Starburst 엔터프라이즈 참조 문서](#)

기타 AWS 설명서

- [자동 포트폴리오 검색 시작하기](#) (AWS 권장 가이드)
- [AWS 기반 Starburst를 통한 클라우드 인프라 비용 및 성능 최적화](#) (블로그 게시물)

AWS에서 입력 파일 크기의 ETL 수집 최적화

작성자: Apoorva Patrikar(AWS)

환경: PoC 또는 파일럿

기술: 분석, 데이터 레이크

워크로드: 오픈 소스

AWS 서비스: AWS Glue,
Amazon S3

요약

이 패턴은 데이터를 처리하기 전에 파일 크기를 최적화하여 AWS Glue의 빅 데이터 및 Apache Spark 워크로드에 대한 추출, 전환, 적재(ETL) 프로세스의 수집 단계를 최적화하는 방법을 보여줍니다. 이 패턴을 사용하여 작은 파일 문제를 예방하거나 해결할 수 있습니다. 즉, 작은 파일이 많으면 파일의 전체 크기 때문에 데이터 처리 속도가 느려지는 경우입니다. 예를 들어, 각각 수백 킬로바이트에 불과한 수백 개의 파일은 AWS Glue 작업의 데이터 처리 속도를 크게 저하시킬 수 있습니다. AWS Glue는 Amazon Simple Storage Service(S3)에서 내부 목록 기능을 수행해야 하고 YARN(또 다른 리소스 협상자)은 대량의 메타데이터를 저장해야 하기 때문입니다. 데이터 처리 속도를 높이려면, 그룹화를 사용하여 ETL 작업이 입력 파일 그룹을 단일 인 메모리 파티션으로 읽을 수 있도록 만들 수 있습니다. 파티션은 크기가 작은 파일들을 자동으로 그룹화합니다. 또는 사용자 지정 코드를 사용하여 기존 파일에 배치 로직을 추가할 수 있습니다.

사전 조건 및 제한 사항

사전 조건

- 활성 상태의 AWS 계정
- 하나 이상의 AWS Glue [작업](#)
- 하나 이상의 빅 데이터 또는 [Apache Spark](#) 워크로드
- [S3 버킷](#)

아키텍처

다음 패턴은 성능에 대한 가시성을 확보하기 위해 AWS Glue 작업에서 다양한 형식의 데이터를 처리한 다음 S3 버킷에 저장하는 방법을 보여줍니다.

이 다이어그램은 다음 워크플로를 보여줍니다.

1. AWS Glue 작업은 CSV, JSON 및 Parquet 형식의 작은 파일을 동적 프레임으로 변환합니다. 참고: 입력 파일의 크기는 AWS Glue 작업의 성능에 가장 큰 영향을 미칩니다.
2. AWS Glue 작업은 S3 버킷에서 내부 목록 기능을 수행합니다.

도구

- [AWS Glue](#)는 완전 관리형 ETL 서비스입니다. 이를 통해 데이터 스토어와 데이터 스트림 간에 데이터를 안정적으로 분류, 정리, 보강하고 이동할 수 있습니다.
- [Amazon Simple Storage Service\(S3\)](#)는 원하는 양의 데이터를 저장, 보호 및 검색하는 데 도움이 되는 클라우드 기반 객체 스토리지 서비스입니다.

에픽

그룹화를 사용하여 읽는 동안 ETL 수집을 최적화합니다.

작업	설명	필요한 기술
그룹 크기를 지정합니다.	50,000개 이상의 파일이 있는 경우 기본값으로 그룹화가 설정됩니다. 그러나 <code>connectionOptions</code> 파라미터에 그룹 크기를 지정하여 50,000개 미만의 파일에 대해 그룹화를 사용할 수도 있습니다. <code>connectionOptions</code> 파라미터는 <code>create_dynamic_frame.from_options</code> 방법에 있습니다.	데이터 엔지니어
그룹화 코드를 작성하십시오.	<code>create_dynamic_frame</code> 방법을 사용하여 동적 프레임을 만들 수 있습니다. 예:	데이터 엔지니어

작업	설명	필요한 기술
	<pre data-bbox="609 220 1015 1186"> S3bucket_node1 = glueContext.create _dynamic_frame.from m_options(format_options={"m ultiline": False}, connection_type="s 3", format="json", connection_options ={ "paths": ["s3:// bucket/prefix/file.j son"], "recurse": True, "groupFiles": 'inPartition', "groupSize": 1048576 }, transformation_ctx ="S3bucket_node1",) </pre> <p data-bbox="592 1239 1031 1606">참고: Amazon S3 파티션 그룹의 파일을 그룹화하는 데 <code>groupFiles</code> 를 사용합니다. 메모리에서 읽을 그룹의 목표 크기를 설정하는 데 <code>groupSize</code> 를 사용합니다. 바이트 단위로 <code>groupSize</code> 를 지정합니다(1,048,576 = 1MB).</p>	
워크플로에 코드를 추가합니다.	AWS Glue의 작업 워크플로 에 그룹화 코드를 추가합니다.	데이터 엔지니어

사용자 지정 로직을 사용하여 ETL 수집을 최적화합니다.

작업	설명	필요한 기술
언어 및 처리 플랫폼을 선택합니다.	사용 사례에 맞는 스크립팅 언어와 처리 플랫폼을 선택하십시오.	클라우드 아키텍트
코드를 작성합니다.	사용자 지정 로직을 작성하여 파일을 일괄 처리하십시오.	클라우드 아키텍트
워크플로에 코드를 추가합니다.	AWS Glue의 작업 워크플로 에 코드를 추가합니다. 이렇게 하면 작업이 실행될 때마다 사용자 지정 로직을 적용할 수 있습니다.	데이터 엔지니어

변환 후 데이터를 쓸 때 파티션을 다시 분할합니다.

작업	설명	필요한 기술
소비 패턴을 분석하세요.	작성한 데이터를 다운스트림 애플리케이션에서 어떻게 사용하는지 알아보십시오. 예를 들어 매일 데이터를 쿼리하고 지역별로만 데이터를 분할하거나 출력 파일이 매우 작은 경우 (예: 파일당 2.5KB) 사용에는 적합하지 않습니다.	DBA
쓰기 전에 데이터를 다시 파티셔닝하십시오.	처리 중 (처리 로직 기반) 및 처리 후 (사용량 기준) 의 조인 또는 쿼리를 기반으로 재분할합니다. 예를 들어 바이트 크기를 기준으로 재분할하거나 열을 기준으로 <code>.repartition(100000)</code> 재분할	데이터 엔지니어

작업	설명	필요한 기술
	합니다 (예:). <code>.repartition("column_name")</code>	

관련 리소스

- [입력 파일을 더 큰 그룹에서 읽기](#)
- [AWS Glue 모니터링](#)
- [아마존 CloudWatch 지표를 사용한 AWS Glue 모니터링](#)
- [작업 모니터링 및 디버깅](#)
- [AWS Glue에서 서버리스 ETL 시작하기](#)

추가 정보

파일 크기 결정

파일 크기가 너무 큰지 작은지 판단할 수 있는 간단한 방법은 없습니다. 파일 크기가 처리 성능에 미치는 영향은 클러스터 구성에 따라 달라집니다. 코어 Hadoop에서는 블록 크기를 최대한 활용하려면 128MB 또는 256MB의 파일을 사용하는 것이 좋습니다.

AWS Glue의 대부분의 텍스트 파일 워크로드의 경우, 5~10 DPU 클러스터에 대해 100MB에서 1GB 사이의 파일 크기를 사용하는 것이 좋습니다. 입력 파일의 최적 크기를 파악하려면, AWS Glue 작업의 사전 처리 섹션을 모니터링한 다음 해당 작업의 CPU 사용률과 메모리 사용률을 확인하십시오.

추가 고려 사항

초기 ETL 단계의 성능에 병목 현상이 발생하는 경우, 처리하기 전에 데이터 파일을 그룹화하거나 병합하는 것을 고려해 보십시오. 파일 생성 프로세스를 완벽하게 제어할 수 있다면, 원시 데이터를 AWS로 보내기 전에 소스 시스템 자체에서 데이터 포인트를 집계하는 것이 훨씬 더 효율적일 수 있습니다.

AWS Step Functions를 사용하여 검증, 변환 및 파티셔닝을 통해 ETL 파이프라인 오케스트레이션

작성자: Sandip Gangapadhyay(AWS)

코드 리포지토리 [aws-step-functions-etl](#): -파이프라인 패턴

환경: 프로덕션

기술: 분석, 빅데이터, 데이터 레이크, 서버리스 DevOps

AWS 서비스: AWS Step Functions, AWS Glue, AWS Lambda, AWS Step Functions

요약

이 패턴은 성능 및 비용 최적화를 위해 대규모 CSV 데이터 세트를 검증, 변환, 압축 및 분할하기 위한 서버리스 추출, 전환, 적재(ETL) 파이프라인을 구축하는 방법을 설명합니다. 파이프라인은 AWS Step Functions에 의해 오케스트레이션되며 오류 처리, 자동 재시도 및 사용자 알림 기능을 포함합니다.

CSV 파일이 Amazon Storage Service(S3) 버킷 소스 폴더에 업로드되면 ETL 파이프라인이 실행되기 시작합니다. 파이프라인은 소스 CSV 파일의 콘텐츠와 스키마를 검증하고, CSV 파일을 압축된 Apache Parquet 형식으로 변환하고, 데이터 세트를 연도, 월, 일별로 분할하고 분석 도구가 처리할 수 있도록 별도의 폴더에 저장합니다.

이 패턴을 자동화하는 코드는 [AWS Step GitHub Functions의 ETL 파이프라인 리포지토리에서](#) 사용할 수 있습니다.

사전 조건 및 제한 사항

사전 조건

- 활성 상태의 AWS 계정
- AWS 명령줄 인터페이스 (AWS CLI) 는 AWS 계정으로 설치 및 구성되므로, AWS 스택을 배포하여 AWS 리소스를 생성할 수 있습니다. CloudFormation AWS CLI 버전 2가 권장됩니다. 설치 지침은 AWS CLI 설명서에서 [AWS CLI 버전 2 설치, 업데이트 및 제거](#)를 참조하십시오. AWS CLI 구성 지침은 AWS CLI 설명서의 [구성 및 보안 인증 파일 설정](#)을 참조하십시오.

- Amazon S3 버킷.
- 올바른 스키마가 포함된 CSV 데이터 세트. (이 패턴에 포함된 [코드 리포지토리](#)는 사용할 수 있는 올바른 스키마와 데이터 유형이 포함된 샘플 CSV 파일을 제공합니다.)
- AWS Management Console에서 사용할 수 있도록 지원되는 웹 브라우저. ([지원되는 브라우저 목록](#)을 참조하십시오.)
- AWS Glue 콘솔 액세스.
- AWS Step Functions 콘솔 액세스.

제한 사항

- AWS Step Functions에서 기록 로그를 보관할 수 있는 최대 한도는 90일입니다. 자세한 내용은 AWS Step Functions 설명서의 [할당량 및 표준 워크플로에 대한 할당량](#)을 참조하십시오.

제품 버전

- AWS 람다용 파이썬 3.11
- AWS Glue 버전 2.0

아키텍처

다이어그램에 표시된 워크플로는 다음과 같은 상위 수준 단계로 구성되어 있습니다.

1. 사용자가 Amazon S3의 소스 폴더에 CSV 파일을 업로드합니다.
2. Amazon S3 알림 이벤트는 Step Functions 상태 머신을 시작하는 AWS Lambda 함수를 시작합니다.
3. Lambda function은 원시 CSV 파일의 스키마와 데이터 유형을 검증합니다.
4. 검증 결과에 따라:
 - a. 소스 파일의 검증에 성공하면 파일은 추가 처리를 위해 스테이지 폴더로 이동합니다.
 - b. 검증에 실패하는 경우 파일은 오류 폴더로 이동하고 Amazon Simple Notification Service(SNS)를 통해 오류 알림이 전송됩니다.
5. AWS Glue 크롤러는 Amazon S3의 스테이지 폴더에서 원시 파일의 스키마를 생성합니다.
6. AWS Glue 작업은 원시 파일을 Parquet 형식으로 변환, 압축 및 파티션합니다.
7. 또한, AWS Glue 작업은 파일을 Amazon S3의 변환 폴더로 이동합니다.

8. AWS Glue 크롤러는 변환된 파일에서 스키마를 생성합니다. 결과 스키마는 모든 분석 작업에서 사용할 수 있습니다. Amazon Athena를 사용하여 임시 쿼리를 실행할 수 있습니다.
9. 파이프라인이 오류 없이 완료되면 스키마 파일이 아카이브 폴더로 이동됩니다. 오류가 발생하는 경우 파일은 대신 오류 폴더로 이동됩니다.
- 10 Amazon SNS는 파이프라인 완료 상태를 기반으로 성공 또는 실패를 나타내는 알림을 보냅니다.

이 패턴에 사용되는 모든 AWS 리소스는 서버리스입니다. 관리할 서버가 없습니다.

도구

서비스

- [AWS Glue](#) - AWS Glue는 고객이 분석할 데이터를 쉽게 준비하고 로드할 수 있는 완전 관리형 ETL 서비스입니다.
- [AWS Step Functions](#) - AWS Step Functions는 AWS Lambda 함수와 기타 AWS 서비스를 결합할 수 있는 서버리스 오케스트레이션 서비스로, 비즈니스 크리티컬 애플리케이션을 구축합니다. AWS Step Functions 그래픽 콘솔을 통해 애플리케이션의 워크플로를 일련의 이벤트 기반 단계로 볼 수 있습니다.
- [Amazon S3](#) - Amazon Simple Storage Service(S3)는 업계 최고의 확장성, 데이터 가용성, 보안 및 성능을 제공하는 객체 스토리지 서비스입니다.
- [Amazon SNS](#) - Amazon Simple Notification Service(SNS)는 가용성이 높고 내구성이 뛰어나며 안전한 완전 관리형 게시/구독 메시징 서비스로, 마이크로서비스, 분산 시스템 및 서버리스 애플리케이션을 분리할 수 있습니다.
- [AWS Lambda](#) - AWS Lambda는 서버를 프로비저닝하거나 관리하지 않고도 코드를 실행할 수 있게 해주는 컴퓨팅 서비스입니다. AWS Lambda는 필요 시에만 코드를 실행하며, 일일 몇 개의 요청에서 초당 수천 개의 요청까지 자동으로 규모를 조정합니다.

code

이 패턴의 코드는 [AWS Step Functions의 ETL 파이프라인](#) 리포지토리에서 사용할 수 있습니다. GitHub 코드 리포지토리에는 다음 파일과 폴더가 포함되어 있습니다.

- `template.yml`— AWS Step Functions를 사용하여 ETL 파이프라인을 생성하기 위한 AWS CloudFormation 템플릿.
- `parameter.json` - 모든 파라미터와 파라미터 값을 포함합니다. 예픽 섹션에 설명된 대로 이 파일을 업데이트하여 파라미터 값을 변경합니다.

- myLayer/python 폴더 - 이 프로젝트에 필요한 AWS Lambda 계층을 생성하는 데 필요한 Python 패키지가 들어 있습니다.
- lambda 폴더 - 다음과 같은 Lambda 함수를 포함합니다.
 - move_file.py - 소스 데이터 세트를 아카이브, 변환 또는 오류 폴더로 이동합니다.
 - check_crawler.py - 실패 메시지를 보내기 전에 RETRYLIMIT 환경 변수로 구성된 횟수만큼 AWS Glue 크롤러의 상태를 확인합니다.
 - start_crawler.py - AWS Glue 크롤러를 시작합니다.
 - start_step_function.py - AWS Step Functions를 시작합니다.
 - start_codebuild.py— AWS CodeBuild 프로젝트를 시작합니다.
 - validation.py - 입력 원시 데이터 세트를 검증합니다.
 - s3object.py - S3 버킷 내에 필요한 디렉터리 구조를 생성합니다.
 - notification.py - 파이프라인 끝에 성공 또는 오류 알림을 보냅니다.

샘플 코드를 사용하려면 에픽 섹션의 지침을 따르십시오.

에픽

소스 파일 준비

작업	설명	필요한 기술
샘플 코드 리포지토리를 복제합니다.	<ol style="list-style-type: none"> 1. AWS Step Functions를 사용한 ETL 파이프라인 리포지토리를 엽니다. 2. 파일 목록 위의 기본 리포지토리 페이지에서 코드를 선택하고 HTTPS로 복제 아래에 나열된 URL을 복사합니다. 3. 작업 디렉터리를 샘플 파일을 저장하려는 위치로 변경합니다. 4. 터미널 또는 명령 프롬프트에서 명령을 입력합니다. 	개발자

작업	설명	필요한 기술
	<pre>git clone <repoURL></pre> <p><repoURL> 는 2단계에서 복사한 URL을 나타냅니다.</p>	

작업	설명	필요한 기술
<p>파라미터 값 업데이트</p>	<p>리포지토리의 로컬 사본에서 다음과 같이 parameter .json 파일을 편집하고 기본 파라미터 값을 업데이트합니다.</p> <ul style="list-style-type: none"> • pS3BucketName - 데이터 세트를 저장하기 위한 S3 버킷 이름. 템플릿을 사용하면 이 버킷이 생성됩니다. 버킷 이름은 전역적으로 고유해야 합니다. • pSourceFolder - 소스 CSV 파일을 업로드하는 데 사용될 S3 버킷 내 폴더 이름. • pStageFolder - 프로세스 중에 스테이징 영역으로 사용될 S3 버킷 내 폴더 이름. • pTransformFolder - 변환되고 분할된 데이터 세트를 저장하는 데 사용될 S3 버킷 내 폴더 이름. • pErrorFolder - 소스 CSV 파일을 검증할 수 없는 경우 해당 파일이 이동되는 S3 버킷 내 폴더. • pArchiveFolder - 소스 CSV 파일을 보관하는 데 사용할 S3 버킷 내부의 폴더 이름. • pEmailforNotification - 성공/오류 알림을 받 	<p>개발자</p>

작업	설명	필요한 기술
	<p>을 수 있는 유효한 이메일 주소.</p> <ul style="list-style-type: none"> • pPrefix- AWS Glue 크롤러 이름에 사용될 접두사 문자열. • pDatasetSchema - 소스 파일의 유효성을 검사할 데이터 세트 스키마. Cerberus Python 패키지는 소스 데이터 세트 검증에 사용됩니다. 자세한 내용은 Cerberus 웹사이트를 참조하십시오. 	

작업	설명	필요한 기술
<p>소스 코드를 S3 버킷에 업로드합니다.</p>	<p>ETL 파이프라인을 자동화하는 CloudFormation 템플릿을 배포하기 전에 CloudFormation 템플릿의 소스 파일을 패키징하여 S3 버킷에 업로드해야 합니다. 이렇게 하려면 사전 구성된 프로필로 다음 AWS CLI 명령을 실행하십시오.</p> <pre data-bbox="597 632 1027 989">aws cloudformation package --template- file template.yml --s3- bucket <bucket_name> --output-template- file packaged.template --profile <profile_ name></pre> <p>여기서 각 항목은 다음과 같습니다.</p> <ul data-bbox="597 1150 1027 1675" style="list-style-type: none"> • <bucket_name> 은 스택을 배포하고자 하는 AWS 리전에 있는 기존 S3 버킷의 이름입니다. 이 버킷은 템플릿의 소스 코드 패키지를 저장하는 데 사용됩니다. CloudFormation • <profile_name> 은 AWS CLI를 설정할 때 미리 구성한 유효한 AWS CLI 프로필입니다. 	<p>개발자</p>

스택 생성

작업	설명	필요한 기술
<p>CloudFormation 템플릿을 배포합니다.</p>	<p>CloudFormation 템플릿을 배포하려면 다음 AWS CLI 명령을 실행합니다.</p> <pre data-bbox="592 478 1027 919">aws cloudformation deploy --stack-name <stack_name> --templat e-file packaged. template --parameter- overrides file://pa rameter.json --capabil ities CAPABILITY_IAM --profile <profile_ name></pre> <p>여기서 각 항목은 다음과 같습니다.</p> <ul data-bbox="592 1081 1027 1365" style="list-style-type: none"> • <stack_name> CloudFormation 스택의 고유 식별자입니다. • <profile-name> 은 사전 구성된 AWS CLI 프로필입니다. 	<p>개발자</p>
<p>진행 상황을 확인합니다.</p>	<p>AWS CloudFormation 콘솔에서 스택 개발 진행 상황을 확인합니다. 상태가 CREATE_COMPLETE 이면 스택이 성공적으로 배포된 것입니다.</p>	<p>개발자</p>
<p>AWS Glue 데이터베이스 이름을 기록하십시오.</p>	<p>스택의 출력 탭에는 AWS Glue 데이터베이스의 이름이 표시됩니다. 키 이름이 GlueDBOutput 입니다.</p>	<p>개발자</p>

파이프라인 테스트

작업	설명	필요한 기술
ETL 파이프라인을 시작하십시오.	<ol style="list-style-type: none"> S3 버킷 내의 소스 폴더 (source 또는 parameter.json 파일에 설정한 폴더 이름)로 이동합니다. 샘플 CSV 파일을 이 폴더에 업로드합니다. (코드 리포지토리는 사용할 수 있는 Sample_Bank_Transaction_Raw_Dataset.csv 라고 부르는 샘플 파일을 제공합니다.) 파일을 업로드하면 Step Functions를 통해 ETL 파이프라인이 시작됩니다. Step Functions 콘솔에서 ETL 파이프라인 상태를 확인합니다. 	개발자
파티셔닝된 데이터 세트를 확인하십시오.	ETL 파이프라인이 완료되면 Amazon S3 변환 폴더(transform 또는 parameter.json 파일에 설정한 폴더 이름)에서 분할된 데이터 세트를 사용할 수 있는지 확인합니다.	개발자
파티셔닝된 AWS Glue 데이터베이스를 확인합니다.	<ol style="list-style-type: none"> AWS Glue 콘솔에서 스택에 의해 생성된 AWS Glue 데이터베이스를 선택합니다(이 데이터베이스는 이전 에픽에서 기록해 둔 데이터베이스입니다). 	개발자

작업	설명	필요한 기술
	2. AWS Glue 데이터 카탈로그에서 파티션을 나눈 테이블을 사용할 수 있는지 확인합니다.	
쿼리를 실행하십시오.	(선택 사항) Amazon Athena를 사용하여 파티셔닝되고 변환된 데이터베이스에서 임시 쿼리를 실행합니다. 지침은 AWS 설명서의 Amazon Athena를 사용한 SQL 쿼리 실행 을 참조하십시오.	데이터베이스 분석가

문제 해결

문제	Solution
AWS Glue 작업 및 크롤러에 대한 AWS ID 및 액세스 관리 (IAM) 권한	AWS Glue 작업 또는 크롤러를 추가로 사용자 지정하는 경우 AWS Glue 작업에서 사용하는 IAM 역할에 적절한 IAM 권한을 부여하거나 AWS Lake Formation에 데이터 권한을 제공해야 합니다. 자세한 내용은 AWS 설명서 를 참조하십시오.

관련 리소스

AWS 서비스 설명서

- [AWS Step Functions](#)
- [AWS Glue](#)
- [AWS Lambda](#)
- [Amazon S3](#)
- [Amazon SNS](#)

추가 정보

다음 다이어그램은 Step Functions Inspector 패널에서 가져온 성공적인 ETL 파이프라인을 위한 AWS Step Functions 워크플로를 보여줍니다.

다음 다이어그램은 Step Functions Inspector 패널에서 입력 검증 오류로 인해 실패한 ETL 파이프라인의 AWS Step Functions 워크플로를 보여줍니다.

Amazon Redshift 기계 학습을 이용하여 고급 분석 수행

작성자: Po Hong(AWS)

환경: PoC 또는 파일럿

기술: 분석, 기계 학습 및 AI

워크로드: 기타 모든 워크로드

AWS 서비스: 아마존 레드시프트, 아마존 SageMaker

요약

Amazon Web Services(AWS) 클라우드에서는 Amazon Redshift 기계 학습(Amazon Redshift ML)을 사용하여 Amazon Redshift 클러스터 또는 Amazon Simple Storage Service(S3)에서 저장된 데이터에 대한 기계 학습 분석을 수행할 수 있습니다. Amazon Redshift 기계 학습은 일반적으로 고급 분석에 사용되는 지도 학습을 지원합니다. Amazon Redshift ML 사용 사례에는 수익 예측, 신용 카드 사기 탐지, 고객 평생 가치(CLV) 또는 고객 이탈 예측 등이 포함됩니다.

Amazon Redshift 기계 학습을 사용하면 데이터베이스 사용자가 표준 SQL 명령으로 기계 학습 모델을 쉽게 생성, 훈련 및 배포할 수 있습니다. Amazon Redshift ML은 Amazon SageMaker Autopilot을 사용하여 제어력과 가시성을 유지하면서 데이터를 기반으로 분류 또는 회귀에 가장 적합한 ML 모델을 자동으로 학습 및 조정합니다.

Amazon Redshift, Amazon S3 및 Amazon 간의 모든 상호 SageMaker 작용은 추상화되어 자동화됩니다. ML 모델은 훈련 및 배포가 완료되면 Amazon Redshift에서 [사용자 정의 함수\(UDF\)](#)로 사용할 수 있게 되며, SQL 쿼리에 사용할 수 있습니다.

[이 패턴은 AWS 블로그의 Amazon Redshift ML을 사용하여 SQL을 사용하여 Amazon Redshift에서 ML 모델을 생성, 교육 및 배포하고 시작하기 리소스 센터의 SageMaker Amazon을 사용하여 ML 모델을 구축, 교육 및 배포하는 자습서를 보완합니다.](#)

사전 조건 및 제한 사항

사전 조건

- 활성 상태의 AWS 계정
- Amazon Redshift 테이블의 기존 데이터

기술

- 기계 학습, 훈련 및 예측을 포함한 Amazon Redshift ML에서 사용하는 용어 및 개념을 숙지해야 합니다. 이에 관한 자세한 내용은 Amazon Machine Learning(Amazon ML) 설명서의 [ML 모델 훈련](#)을 참조하세요.
- Amazon Redshift 사용자 설정, 액세스 관리 및 표준 SQL 구문에 대한 경험이 있어야 합니다. 이에 관한 자세한 내용은 Amazon Redshift 설명서의 [Amazon Redshift 시작하기](#)를 참조하세요.
- Amazon S3 및 AWS Identity and Access Management(IAM)에 관한 지식과 경험이 있어야 합니다.
- AWS Command Line Interface(AWS CLI)에서 명령을 실행한 경험이 있으면 유용하지만 필수 사항은 아닙니다.

제한 사항

- Amazon Redshift 클러스터와 S3 버킷이 동일한 AWS 리전에 있어야 합니다.
- 이 패턴의 접근 방식은 회귀, 바이너리 분류 및 멀티클래스 분류와 같은 지도 학습 모델만 지원합니다.

아키텍처

다음 단계에서는 Amazon Redshift ML을 사용하여 ML 모델을 구축, 교육 및 SageMaker 배포하는 방법을 설명합니다.

1. Amazon Redshift는 훈련 데이터를 S3 버킷으로 내보냅니다.
2. SageMaker 오토파일럿은 교육 데이터를 자동으로 전처리합니다.
3. CREATE MODEL 명령문이 호출되면 Amazon Redshift SageMaker ML은 교육에 사용합니다.
4. SageMaker 오토파일럿은 평가 지표를 최적화하는 ML 알고리즘과 최적의 하이퍼 파라미터를 검색하고 추천합니다.
5. Amazon Redshift ML은 출력 ML 모델을 Amazon Redshift 클러스터에 SQL 함수로 등록합니다.
6. 기계 학습 모델의 함수는 SQL 문에서 사용할 수 있습니다.

기술 스택

- Amazon Redshift

- SageMaker
- Amazon S3

도구

- [Amazon Redshift](#) – Amazon Redshift는 페타바이트 규모의 엔터프라이즈 레벨 완전 관리형 데이터 웨어하우징 서비스입니다.
- [Amazon Redshift ML](#) – Amazon Redshift 기계 학습(Amazon Redshift ML)은 모든 기술 수준의 분석가와 데이터 사이언티스트가 ML 기술을 쉽게 사용할 수 있도록 하는 강력한 클라우드 기반 서비스입니다.
- [Amazon S3](#) – Amazon Simple Storage Service(S3)는 인터넷에 대한 스토리지입니다.
- [Amazon SageMaker](#) — SageMaker 완전 관리형 ML 서비스입니다.
- [Amazon SageMaker 오토파일럿](#) — SageMaker 오토파일럿은 자동 기계 학습 (AutoML) 프로세스의 주요 작업을 자동화하는 기능 세트입니다.

코드

다음 코드를 사용하여 Amazon Redshift에서 지도 ML 모델을 생성할 수 있습니다.

```
“CREATE MODEL customer_churn_auto_model
FROM (SELECT state,
             account_length,
             area_code,
             total_charge/account_length AS average_daily_spend,
             cust_serv_calls/account_length AS average_daily_cases,
             churn
      FROM customer_activity
      WHERE record_date < '2020-01-01'
     )
TARGET churn
FUNCTION ml_fn_customer_churn_auto
IAM_ROLE 'arn:aws:iam::XXXXXXXXXXXX:role/Redshift-ML'
SETTINGS (
  S3_BUCKET 'your-bucket'
);”
```

참고: SELECT 상태는 Amazon Redshift 일반 테이블, Amazon Redshift Spectrum 외부 테이블 또는 둘 다를 참조할 수 있습니다.

에픽

훈련 및 테스트 데이터 세트 준비

작업	설명	필요한 기술
훈련 및 테스트 데이터 세트를 준비합니다.	<p>AWS 관리 콘솔에 로그인하고 Amazon SageMaker 콘솔을 엽니다. 기계 학습 모델 구축, 훈련 및 배포 자습서의 지침에 따라 레이블 열(지도 학습)이 있고 헤더가 없는.csv 또는 Apache Parquet 파일을 생성합니다.</p> <p>참고: 원시 데이터 세트를 혼합하여 모델 학습을 위한 훈련 세트(70%)와 모델 성능 평가를 위한 테스트 세트(30%)로 분할하는 것이 좋습니다.</p>	데이터 사이언티스트

기술 스택 준비 및 구성

작업	설명	필요한 기술
Amazon Redshift 클러스터를 생성하고 구성합니다.	<p>Amazon Redshift 콘솔에서 사용자의 요건에 따라 클러스터를 생성합니다. 이에 대한 자세한 내용은 Amazon Redshift 설명서의 클러스터 생성을 참조하세요.</p> <p>중요: Amazon Redshift 클러스터는 SQL_PREVIEW 유지 관</p>	DBA, 클라우드 아키텍트

작업	설명	필요한 기술
	<p>리 스택으로 생성해야 합니다. 트랙 미리 보기에 관한 자세한 내용은 Amazon Redshift 설명서의 클러스터 유지 관리 트랙 선택 섹션을 참조하세요.</p>	
<p>S3 버킷을 생성하여 훈련 데이터와 모델 아티팩트를 저장합니다.</p>	<p>Amazon S3 콘솔에서 훈련 및 테스트 데이터를 위한 S3 버킷을 생성합니다. S3 버킷 생성에 대한 자세한 내용은 AWS Quick Starts의 S3 버킷 생성을 참조하세요.</p> <p>중요: Amazon Redshift 클러스터와 S3 버킷이 같은 리전에 있는지 확인하세요.</p>	<p>DBA, 클라우드 아키텍트</p>
<p>IAM 정책을 생성하여 Amazon Redshift 클러스터에 연결합니다.</p>	<p>Amazon Redshift 클러스터가 Amazon S3에 SageMaker 액세스할 수 있도록 허용하는 IAM 정책을 생성합니다. 지침 및 단계는 Amazon Redshift 설명서에서 Amazon Redshift ML을 사용하기 위한 클러스터 설정을 참조하세요.</p>	<p>DBA, 클라우드 아키텍트</p>
<p>Amazon Redshift 사용자 및 그룹이 스키마와 테이블에 액세스할 수 있도록 허용합니다.</p>	<p>Amazon Redshift의 사용자 및 그룹이 내부 및 외부 스키마와 테이블에 액세스할 수 있도록 권한을 부여합니다. 단계 및 지침은 Amazon Redshift 설명서의 권한 및 소유권 관리를 참조하세요.</p>	<p>DBA</p>

Amazon Redshift에서 ML 모델 생성 및 훈련

작업	설명	필요한 기술
Amazon Redshift에서 ML 모델을 생성하고 훈련합니다.	Amazon Redshift ML에서 ML 모델을 생성하고 훈련합니다. 자세한 내용은 Amazon Redshift 설명서의 CREATE MODEL 명령문을 참조하세요.	개발자, 데이터 사이언티스트

Amazon Redshift에서 배치 추론 및 예측 수행

작업	설명	필요한 기술
생성된 ML 모델 함수를 사용하여 추론을 수행합니다.	생성된 ML 모델 함수를 사용하여 추론을 수행하는 방법에 대한 자세한 내용은 Amazon Redshift 설명서의 예측 을 참조하세요.	데이터 사이언티스트, 비즈니스 인텔리전스 사용자

관련 리소스

훈련 및 테스트 데이터 세트 준비

- [Amazon을 통한 기계 학습 모델 구축, 교육 및 배포 SageMaker](#)

기술 스택 준비 및 구성

- [Amazon Redshift 클러스터 생성](#)
- [Amazon Redshift 클러스터 유지 관리 트랙 선택](#)
- [S3 버킷 생성](#)
- [Amazon Redshift ML 사용을 위해 Amazon Redshift 클러스터 설정](#)
- [Amazon Redshift에서 권한 및 소유권 관리](#)

Amazon Redshift에서 ML 모델 생성 및 훈련

- [Amazon Redshift에서 CREATE MODEL 문 생성](#)

Amazon Redshift에서 배치 추론 및 예측 수행

- [Amazon Redshift에서 예측](#)

기타 리소스

- [Amazon Redshift ML 시작하기](#)
- [Amazon Redshift ML과 SQL을 사용하여 Amazon Redshift에서 ML 모델 생성, 훈련 및 배포](#)
- [Amazon Redshift 파트너](#)
- [AWS 기계 학습 컴퓨팅 파트너](#)

Athena를 사용한 Amazon DynamoDB 테이블 액세스, 쿼리 및 조인

작성자: 모이널 알 마문 (AWS)

환경: 프로덕션

기술: 분석, 데이터베이스, 서버 리스, 빅 데이터

AWS 서비스: Amazon Athena, Amazon DynamoDB, AWS Lambda, Amazon S3

요약

이 패턴은 Amazon Athena DynamoDB 커넥터를 사용하여 Amazon Athena와 Amazon DynamoDB 간의 연결을 설정하는 방법을 보여줍니다. 커넥터는 AWS Lambda 함수를 사용하여 DynamoDB의 데이터를 쿼리합니다. 연결을 설정하기 위해 코드를 작성할 필요가 없습니다. 연결이 설정되면 Athena [페더레이션 쿼리](#)를 사용해 Athena에서 SQL 명령을 실행하여 DynamoDB 테이블에 빠르게 액세스하고 분석할 수 있습니다. 또한 하나 이상의 DynamoDB 테이블을 서로 조인하거나 Amazon Redshift 또는 Amazon Aurora와 같은 다른 데이터 소스에 조인할 수 있습니다.

사전 조건 및 제한 사항

사전 조건

- DynamoDB 테이블, Athena 데이터 소스, Lambda, AWS Identity and Access Management(IAM) 역할을 관리할 권한이 있는 활성 AWS 계정
- Athena가 쿼리 결과를 저장할 수 있는 Amazon Simple Storage Service(S3) 버킷
- Athena DynamoDB 커넥터가 데이터를 단기간에 저장할 수 있는 S3 버킷
- [Athena 엔진 버전 2](#)를 지원하는 AWS 리전
- Athena 및 필수 S3 버킷에 액세스할 수 있는 IAM 권한
- [Amazon Athena DynamoDB 커넥터](#), 설치됨

제한 사항

DynamoDB 테이블을 쿼리하는 데는 비용이 듭니다. 테이블 크기가 몇 기가바이트(GB)를 초과하면 비용이 많이 들 수 있습니다. 전체 테이블 SCAN 작업을 수행하기 전에 비용을 고려하는 것이 좋습니다. 자세한 내용은 [Amazon DynamoDB 요금](#)을 참조하세요. 비용을 줄이고 높은 퍼포먼스를 달성하려면

쿼리에 항상 LIMIT를 사용하는 것이 좋습니다(예:SELECT * FROM table1 LIMIT 10). 또한 프로덕션 환경에서 JOIN 또는 GROUP BY 쿼리를 수행하기 전에 테이블 크기를 고려해야 합니다. 테이블이 너무 크면 [테이블을 Amazon S3로 마이그레이션](#)하는 등의 대체 옵션을 고려해 보세요.

아키텍처

다음 다이어그램은 사용자가 Athena의 DynamoDB 테이블에서 SQL 쿼리를 실행하는 방법을 보여줍니다.

이 다이어그램은 다음 워크플로를 보여줍니다.

1. DynamoDB 테이블을 쿼리하기 위해 사용자는 Athena에서 SQL 쿼리를 실행합니다.
2. Athena는 Lambda 함수를 시작합니다.
3. Lambda 함수는 DynamoDB 테이블의 요청된 데이터를 쿼리합니다.
4. DynamoDB는 요청된 데이터를 Lambda 함수에 반환합니다. 그런 다음 함수는 Athena를 통해 쿼리 결과를 사용자에게 전송합니다.
5. Lambda 함수는 S3 버킷에 데이터를 저장합니다.

기술 스택

- Amazon Athena
- Amazon DynamoDB
- Amazon S3
- AWS Lambda

도구

- [Amazon Athena](#)는 표준 SQL을 사용하여 Amazon S3에 있는 데이터를 직접 분석할 수 있는 대화형 쿼리 서비스입니다.
- [Amazon Athena DynamoDB 커넥터](#)는 Athena가 DynamoDB에 연결하고 SQL 쿼리를 사용하여 테이블에 액세스할 수 있도록 하는 AWS 도구입니다.
- [Amazon DynamoDB](#)는 빠르고 예측 가능하고 확장 가능한 성능을 제공하는 완전 관리형 NoSQL 데이터베이스 서비스입니다.

- [AWS Lambda](#)는 서버를 프로비저닝하거나 관리할 필요 없이 코드를 실행하는 데 도움이 되는 컴퓨팅 서비스입니다. 필요할 때만 코드를 실행하며 자동으로 확장이 가능하므로 사용한 컴퓨팅 시간만큼만 비용을 지불합니다.

에픽

샘플 DynamoDB 테이블 생성

작업	설명	필요한 기술
첫 번째 샘플 테이블을 생성합니다.	<ol style="list-style-type: none"> 1. AWS Management Console에 로그인하고 DynamoDB 콘솔을 엽니다. 2. 테이블 생성을 선택합니다. 3. 테이블 이름에 dydbtable1을 입력합니다. 4. 파티션 키에 PK1을 입력합니다. 5. 정렬 키에는 SK1을 입력합니다. 6. 테이블 설정 섹션에서 사용자 지정 설정을 선택합니다. 7. 테이블 클래스 섹션에서 DynamoDB 표준을 선택합니다. 8. 읽기/쓰기 용량 설정 섹션의 용량 모드에서 온디맨드를 선택합니다. 9. 저장된 암호화 섹션에서 Amazon DynamoDB 소유를 선택합니다. 10. 테이블 생성을 선택합니다. 	개발자
첫 번째 테이블에 샘플 데이터를 삽입합니다.	<ol style="list-style-type: none"> 1. DynamoDB 콘솔을 엽니다. 	개발자

작업	설명	필요한 기술
	<p>2. 탐색 창에서 테이블을 선택한 다음 이름 열에서 테이블을 선택합니다.</p> <p>3. 작업을 선택한 다음 아이템 생성을 선택합니다.</p> <p>4. JSON 보기를 선택합니다.</p> <p>5. 속성 편집기의 제목 표시줄에서 DynamoDB JSON 보기를 끕니다.</p> <p>6. 속성 편집기에서 다음 샘플 데이터를 하나씩 입력합니다.</p> <pre data-bbox="597 884 1024 1121"> { "PK1": "1234", "SK1": "info", "Salary": "5000" } </pre> <pre data-bbox="597 1157 1024 1394"> { "PK1": "1235", "SK1": "info", "Salary": "5200" } </pre>	

작업	설명	필요한 기술
<p>두 번째 샘플 테이블을 생성합니다.</p>	<ol style="list-style-type: none"> 1. DynamoDB 콘솔을 엽니다. 2. 테이블 생성을 선택합니다. 3. 테이블 이름에 dydbtable2을 입력합니다. 4. 파티션 키에 PK2를 입력합니다. 5. 정렬 키에는 SK2을 입력합니다. 6. 테이블 설정 섹션에서 사용자 지정 설정을 선택합니다. 7. 테이블 클래스 섹션에서 DynamoDB 표준을 선택합니다. 8. 읽기/쓰기 용량 설정 섹션의 용량 모드에서 온디맨드를 선택합니다. 9. 저장된 암호화 섹션에서 Amazon DynamoDB 소유를 선택합니다. 10.테이블 생성을 선택합니다. 	<p>개발자</p>

작업	설명	필요한 기술
두 번째 테이블에 샘플 데이터를 삽입합니다.	<ol style="list-style-type: none"> DynamoDB 콘솔을 엽니다. 탐색 창에서 테이블을 선택한 다음 이름 옆에서 테이블을 선택합니다. 작업을 선택한 다음 아이템 생성을 선택합니다. 속성 편집기의 제목 표시줄에서 DynamoDB JSON 보기를 끕니다. 속성 편집기에서 다음 샘플 데이터를 하나씩 입력합니다. <pre>{ "PK2": "1234", "SK2": "bonus", "Bonus": "500" }</pre> <pre>{ "PK2": "1235", "SK2": "bonus", "Bonus": "1000" }</pre>	개발자

Athena에서 DynamoDB용 데이터 소스를 생성

작업	설명	필요한 기술
데이터 소스 커넥터를 설정합니다.	DynamoDB용 데이터 소스를 생성한 다음 Lambda 함수를 생성하여 해당 데이터 소스에 연결합니다.	개발자

작업	설명	필요한 기술
	<ol style="list-style-type: none"> 1. AWS Management Console에 로그인하고 Athena 콘솔을 엽니다. 2. 탐색 창에서 데이터 소스를 선택한 다음 데이터 소스 생성을 선택합니다. 3. Amazon DynamoDB 데이터 소스를 선택한 후 다음을 선택합니다. 4. 데이터 소스 세부 정보 섹션의 데이터 소스 이름에 testDynamoDB를 입력합니다. 5. 연결 세부 정보 섹션에서 이미 배포된 Lambda 함수를 선택하거나, 이 패턴에 사용할 Lambda 함수가 없는 경우 Lambda 함수 생성을 선택합니다. 참고: Lambda 함수 생성에 대한 자세한 내용은 Lambda 개발자 안내서의 Lambda 시작하기를 참조하세요. 6. (선택 사항) Lambda 함수 생성을 선택한 경우 해당 스택을 배포하기 전에 Java 애플리케이션에 포함된 CloudFormation AWS 템플릿을 구성해야 합니다. 템플릿에는 ApplicationName, SpillBucket AthenaCatalogName, 및 기타 애플리케이션 설정이 포함됩니다. 참고: 이 Java 기반 애플리 	

작업	설명	필요한 기술
	<p>케이션을 배포한 후 스택은 Athena가 DynamoDB와 통신할 수 있도록 Lambda 함수를 생성합니다. 이렇게 하면 SQL 명령을 통해 테이블에 액세스할 수 있습니다.</p> <p>7. Lambda 함수를 배포합니다.</p> <p>8. 다음을 선택합니다.</p>	
<p>Lambda 함수가 S3 유출 버킷에 액세스할 수 있는지 확인합니다.</p>	<ol style="list-style-type: none"> 1. Lambda 콘솔을 엽니다. 2. 탐색 창에서 함수를 선택한 후, 앞서 생성한 함수를 선택합니다. 3. 구성 탭을 선택합니다. 4. 왼쪽 창에서 환경 변수를 선택한 다음 키 값이 <code>spill_bucket</code> 인지 확인합니다. 5. 왼쪽 창에서 권한을 선택한 다음 실행 역할 섹션에서 연결된 IAM 역할을 선택합니다. 참고: IAM 콘솔에서 Lambda 함수에 연결된 IAM 역할로 이동합니다. 6. <code>spill_bucket</code> 버킷에 대한 쓰기 권한이 있는지 확인하세요. <p>오류가 발생하는 경우 이 패턴의 추가 정보 섹션에서 지침을 참조하세요.</p>	개발자

Athena에서 DynamoDB 테이블에 액세스

작업	설명	필요한 기술
DynamoDB 테이블을 쿼리합니다.	<ol style="list-style-type: none"> 1. AWS Management Console에 로그인하고 Athena 콘솔을 엽니다. 2. 탐색 창에서 데이터 소스를 선택한 다음 데이터 소스 생성을 선택합니다. 3. 탐색 창에서 쿼리 편집기를 선택합니다. 4. 편집기 탭의 데이터 섹션에서 데이터 소스로 데이터 소스의 데이터 소스를 선택합니다. 5. 데이터베이스에서 데이터베이스를 선택합니다. 6. 쿼리 1에 <code>SELECT * FROM dydhtable1 t1;</code> 다음 쿼리를 입력합니다. 7. 실행을 선택한 다음 테이블에서 출력을 확인합니다. 8. 쿼리 2에 <code>SELECT * FROM dydhtable2 t2;</code> 다음 쿼리를 입력합니다. 9. 실행을 선택한 다음 테이블에서 출력을 확인합니다. 	개발자
두 DynamoDB 테이블을 조인합니다.	DynamoDB는 NoSQL 데이터 스토어이며 SQL 조인 작업을 지원하지 않습니다. 따라서 두 DynamoDB 테이블에서 조인 작업을 수행해야 합니다.	개발자

작업	설명	필요한 기술
	<ol style="list-style-type: none"> 1. 더하기 아이콘을 선택하여 새 쿼리를 생성합니다. 2. 쿼리 3에는 다음 쿼리를 입력합니다. <pre>SELECT pk1, salary, bonus FROM dydbtable1 t1 JOIN dydbtable2 t2 ON t1.pk1 = t2.pk2;</pre>	

관련 리소스

- [Amazon Athena DynamoDB 커넥터\(AWS 랩\)](#)
- [Amazon Athena의 새로운 페더레이션 쿼리로 모든 데이터 소스를 쿼리\(AWS 빅 데이터 블로그\)](#)
- [Athena 엔진 버전 참조\(Athena 사용자 가이드\)](#)
- [AWS Glue 및 Amazon Athena를 사용하여 Amazon DynamoDB 데이터 추출 및 분석을 간소화하세요\(AWS 데이터베이스 블로그\)](#)

추가 정보

Athena에서 {bucket_name}/folder_name/ 형식의 spill_bucket을(를) 사용하여 쿼리를 실행하면 다음과 같은 오류 메시지가 나타날 수 있습니다.

```
"GENERIC_USER_ERROR: Encountered an exception[java.lang.RuntimeException] from your
LambdaFunction[arn:aws:lambda:us-east-1:xxxxxx:function:testdynamodb] executed in
context[retrieving meta-data] with message[You do NOT own the spill bucket with the
name: s3://test-bucket-dynamodbconnector/athena_dynamodb_spill_data/]
This query ran against the "default" database, unless qualified by the query. Please
post the error message on our forum or contact customer support with Query Id:
[query-id]"
```

이 오류를 해결하려면 Lambda 함수의 환경 변수를 spill_bucket에서 {bucket_name_only}(으)로 업데이트한 다음 버킷 쓰기 액세스에 대한 다음 Lambda IAM 정책을 업데이트하세요.

```
{
  "Action": [
    "s3:GetObject",
    "s3:ListBucket",
    "s3:GetBucketLocation",
    "s3:GetObjectVersion",
    "s3:PutObject",
    "s3:PutObjectAcl",
    "s3:GetLifecycleConfiguration",
    "s3:PutLifecycleConfiguration",
    "s3:DeleteObject"
  ],
  "Resource": [
    "arn:aws:s3:::spill_bucket",
    "arn:aws:s3:::spill_bucket/*"
  ],
  "Effect": "Allow"
}
```

또는 이전에 만든 Athena 데이터 소스 커넥터를 제거하고 `spill_bucket`에 `{bucket_name}`만 사용하여 새로 생성하세요.

스칼라 Python UDF를 사용하여 Amazon Redshift 쿼리 결과에 대한 언어별 정렬 설정

작성자: 에단 스타크(AWS)

환경: 프로덕션

기술: 분석

AWS 서비스: Amazon Redshift

요약

이 패턴은 스칼라 Python UDF(사용자 정의 함수)를 사용하여 Amazon Redshift 쿼리 결과에 대해 대소문자를 구분하지 않는 언어 정렬을 설정하는 단계 및 샘플 코드를 제공합니다. Amazon Redshift는 바이너리 UTF-8 순서를 기준으로 결과를 반환하며 언어별 정렬을 지원하지 않으므로 스칼라 Python UDF를 사용해야 합니다. Python UDF는 Python 2.7 프로그램을 기반으로 하며 데이터 웨어하우스에서 실행되는 비SQL 처리 코드입니다. 단일 쿼리에서 SQL 문을 사용하여 Python UDF 코드를 실행할 수 있습니다. 자세한 내용은 AWS 빅 데이터 블로그 게시물 [Amazon Redshift의 Python UDF 소개](#)를 참조하세요.

이 패턴의 샘플 데이터는 데모용으로 튀르키예어 알파벳을 기반으로 합니다. 이 패턴의 스칼라 Python UDF는 Amazon Redshift의 기본 쿼리 결과가 튀르키예어 문자의 언어 순서를 따르도록 구축되었습니다. 자세한 내용은 이 패턴의 추가 정보 섹션에서 튀르키예어 예를 참조하세요. 다른 언어에 대해 이 패턴의 스칼라 Python UDF를 수정할 수 있습니다.

사전 조건 및 제한 사항

사전 조건

- 데이터베이스, 스키마 및 테이블을 포함하는 Amazon Redshift [클러스터](#)
- CREATE TABLE 및 CREATE FUNCTION 권한이 있는 Amazon Redshift [사용자](#)
- [Python 2.7](#) 이상

제한 사항

이 패턴의 쿼리에서 사용하는 언어 정렬은 대소문자를 구분하지 않습니다.

아키텍처

기술 스택

- Amazon Redshift
- Python UDF

도구

서비스

- [Amazon Redshift](#)는 AWS 클라우드의 관리형 페타바이트 규모 데이터 웨어하우스 서비스입니다. Amazon Redshift는 데이터 레이크와 통합되므로 데이터를 사용하여 비즈니스 및 고객에 대한 새로운 인사이트를 얻을 수 있습니다.

기타 도구

- [Python\(UDF\) 사용자 정의 함수](#)는 Python으로 작성한 다음 SQL 문에서 호출할 수 있는 함수입니다.

에픽

쿼리 결과를 언어순으로 정렬하는 코드 개발

작업	설명	필요한 기술
샘플 데이터용 테이블을 생성합니다.	<p>Amazon Redshift에서 테이블을 생성하고 샘플 데이터를 테이블에 삽입하려면 다음 SQL 문을 사용합니다.</p> <pre>CREATE TABLE my_table (first_name varchar(30)); INSERT INTO my_table (first_name) VALUES ('ali'),</pre>	데이터 엔지니어

작업	설명	필요한 기술
	<pre data-bbox="592 205 1031 703"> ('Ali'), ('ırmak'), ('IRMAK'), ('irem'), ('İREM'), ('oğuz'), ('OĞUZ'), ('ömer'), ('ÖMER'), ('sedat'), ('SEDAT'), ('şule'), </pre> <p data-bbox="592 735 1031 1071">참고: 샘플 데이터의 첫 번째 이름에는 튀르키예어 알파벳의 특수 문자가 포함됩니다. 이 예제의 튀르키예어 고려 사항에 대한 자세한 내용은 이 패턴의 추가 정보 섹션에서 튀르키예어 예를 참조하세요.</p>	

작업	설명	필요한 기술
<p>샘플 데이터의 기본 정렬을 확인합니다.</p>	<p>Amazon Redshift에서 샘플 데이터의 기본 정렬을 확인하려면 다음 쿼리를 실행합니다.</p> <pre data-bbox="597 394 1024 552">SELECT first_name FROM my_table ORDER BY first_name;</pre> <p>쿼리는 이전에 생성한 테이블에서 첫 번째 이름의 목록을 반환합니다.</p> <pre data-bbox="597 758 1024 1434">first_name ----- Ali IRMAK OĞUZ SEDAT ali irem oğuz sedat ÖMER ömer İREM ırmak ŞULE şule</pre> <p>기본 바이너리 UTF-8 순서는 튀르키예어 특수 문자의 언어 순서를 수용하지 않으므로 쿼리 결과의 순서가 올바르지 않습니다.</p>	<p>데이터 엔지니어</p>

작업	설명	필요한 기술
스칼라 Python UDF를 생성합니다.	<p>스칼라 Python UDF를 생성하려면 다음 SQL 코드를 사용합니다.</p> <pre> CREATE OR REPLACE FUNCTION collate_sort (value varchar) RETURNS varchar IMMUTABLE AS \$\$ def sort_str(val): import string dictionary = { 'I': 'ı', 'ı': 'h~', 'İ': 'i', 'Ş': 's~', 'ş': 's~', 'Ğ': 'g~', 'ğ': 'g~', 'Ü': 'u~', 'ü': 'u~', 'Ö': 'o~', 'ö': 'o~', 'Ç': 'c~', 'ç': 'c~' } for key, value in dictionary.items(): val = val.replace(key, value) return val.lower () </pre>	데이터 엔지니어

작업	설명	필요한 기술
	<pre> return sort_str(value) \$\$ LANGUAGE plpythonu; </pre>	
<p>샘플 데이터를 쿼리합니다.</p>	<p>Python UDF를 사용하여 샘플 데이터를 쿼리하려면 다음 SQL 쿼리를 실행합니다.</p> <pre> SELECT first_name FROM my_table ORDER BY collate_order(firs t_name); </pre> <p>이제 쿼리는 샘플 데이터를 튀르키예어 언어 순서로 반환합니다.</p> <pre> first_name ----- ali Ali ırmak IRMAK irem İREM oğuz OĞUZ ömer Ömer sedat SEDAT şule ŞULE </pre>	<p>데이터 엔지니어</p>

관련 리소스

- [ORDER BY 절](#)(Amazon Redshift 설명서)
- [스칼라 Python UDF 생성](#)(Amazon Redshift 설명서)

추가 정보

튀르키예어 예

Amazon Redshift는 언어별 정렬 순서가 아닌 바이너리 UTF-8 정렬 순서를 기준으로 쿼리 결과를 반환합니다. 즉, 튀르키예어 문자를 포함하는 Amazon Redshift 테이블을 쿼리하면 쿼리 결과는 튀르키예어의 언어 순서에 따라 정렬되지 않습니다. 튀르키예어에는 로마자에 나타나지 않는 6개의 특수 문자(ç, ı, ğ, ö, ş, ü)가 포함되어 있습니다. 다음 표와 같이 이러한 특수 문자는 바이너리 UTF-8 순서를 기준으로 정렬된 결과 세트의 끝에 배치됩니다.

바이너리 UTF-8 순서	튀르키예어 언어 순서
a	a
b	b
c	c
d	ç(*)
e	d
f	e
g	f
h	g
i	ğ(*)
j	h
k	ı(*)
l	i

m	j
n	k
o	l
p	m
r	n
s	o
t	ö(*)
u	p
v	r
y	s
z	ş(*)
ç(*)	t
ğ(*)	u
ı(*)	ü(*)
ö(*)	v
ş(*)	y
ü(*)	z

참고: 별표(*)는 튀르키예어의 특수 문자를 나타냅니다.

위 표에서 볼 수 있듯이 특수 문자 ç는 튀르키예어 언어 순서에서는 c와 d 사이에 있지만 바이너리 UTF-8 순서에서는 z 뒤에 나타납니다. 이 패턴의 스칼라 Python UDF는 다음 문자 대체 사전을 사용하여 튀르키예어 특수 문자를 해당하는 로마자 문자로 대체합니다.

튀르키예어 특수 문자

로마자 해당 문자

Ç	c~
ı	h~
ğ	g~
ö	o~
ş	s~
ü	u~

참고: 해당 튀르키예어 특수 문자를 대체하는 로마자 문자 끝에는 물결표(~) 문자가 추가됩니다.

스칼라 Python UDF 함수 수정

함수가 locate 파라미터를 받아들이고 여러 트랜잭션 사전을 지원하도록 이 패턴에서 스칼라 Python UDF 함수를 수정하려면 다음 SQL 코드를 사용합니다.

```
CREATE OR REPLACE FUNCTION collate_sort (value varchar, locale varchar)
RETURNS varchar
IMMUTABLE
AS
$$
def sort_str(val):
    import string
    # Turkish Dictionary
    if locale == 'tr-TR':
        dictionary = {
            'I': 'ı',
            'ı': 'h~',
            'İ': 'i',
            'Ş': 's~',
            'ş': 's~',
            'Ğ': 'g~',
            'ğ': 'g~',
            'Ü': 'u~',
            'ü': 'u~',
            'Ö': 'o~',
            'ö': 'o~',
            'Ç': 'c~',
            'ç': 'c~'
        }
```

```
    }
    # German Dictionary
    if locale == 'de-DE':
        dictionary = {
            ....
            ....
        }

    for key, value in dictionary.items():
        val = val.replace(key, value)

    return val.lower()

return sort_str(value)

$$ LANGUAGE plpythonu;
```

다음 예제 코드에서는 수정된 Python UDF를 쿼리하는 방법을 보여 줍니다.

```
SELECT first_name FROM my_table ORDER BY collate_order(first_name, 'tr-TR');
```

Lambda 함수가 서로 다른 AWS 리전의 S3 버킷에서 이벤트 알림을 수신하도록 Lambda 함수 구독

작성자: Suresh Konathala(AWS) 및 Arindom Sarkar(AWS)

환경: 프로덕션

기술: 분석

AWS 서비스: AWS Lambda, Amazon S3, Amazon SNS, Amazon SQS

요약

[Amazon Simple Storage Service\(S3\) 이벤트 알림](#)은 S3 버킷의 특정 이벤트(예: 객체 생성 이벤트, 객체 제거 이벤트 또는 객체 복원 이벤트)에 대한 알림을 게시합니다. AWS Lambda 함수를 사용하여 애플리케이션의 요구 사항에 따라 이러한 알림을 처리할 수 있습니다. 하지만 Lambda 함수는 다른 AWS 리전에서 호스팅되는 S3 버킷의 알림을 직접 구독할 수 없습니다.

이 패턴의 접근 방식은 각 리전의 Amazon Simple Notification Service(SNS) 주제를 사용하여 리전 간 S3 버킷의 Amazon S3 알림을 처리하는 [팬아웃 시나리오](#)를 배포합니다. 이러한 리전 SNS 주제는 Lambda 함수가 포함된 중앙 리전의 Amazon Simple Queue Service(Amazon SQS) 대기열로 Amazon S3 이벤트 알림을 보냅니다. Lambda 함수는 이 SQS 대기열을 구독하고 조직의 요구 사항에 따라 이벤트 알림을 처리합니다.

사전 조건 및 제한 사항

사전 조건

- 활성 상태의 AWS 계정.
- Amazon SQS 대기열 및 Lambda 함수를 호스팅하는 중앙 리전을 포함하여 여러 리전의 기존 S3 버킷.
- AWS Command Line Interface(AWS CLI), 설치 및 구성됨. 이에 관한 자세한 내용은 AWS CLI 설명서의 [AWS CLI 설치, 업데이트 및 제거](#)를 참조하십시오.
- Amazon SNS의 팬아웃 시나리오에 익숙해야 합니다. 이에 대한 자세한 내용은 Amazon SNS 설명서에서 [일반적인 Amazon SNS 시나리오](#)를 참조하십시오.

아키텍처

다음 다이어그램은 이 패턴 접근 방식의 아키텍처를 보여줍니다.

이 다이어그램은 다음 워크플로를 보여줍니다.

1. Amazon S3는 S3 버킷(예: 객체 생성, 제거된 객체 또는 복원된 객체)에 대한 이벤트 알림을 동일한 리전의 SNS 주제로 전송합니다.
2. SNS 주제는 중앙 리전의 SQS 대기열에 이벤트를 게시합니다.
3. SQS 대기열은 Lambda 함수의 이벤트 소스로 구성되며 Lambda 함수의 이벤트 메시지를 버퍼링합니다.
4. Lambda 함수는 SQS 대기열에서 메시지를 폴링하고 애플리케이션의 요구 사항에 따라 Amazon S3 이벤트 알림을 처리합니다.

기술 스택

- Lambda
- Amazon SNS
- Amazon SQS
- Amazon S3

도구

- [AWS CLI](#) – AWS Command Line Interface(AWS CLI)는 명령줄 셸에서 명령을 사용하여 AWS 서비스와 상호 작용할 수 있는 오픈 소스 도구입니다. 최소한의 구성으로 명령 프롬프트에서 브라우저 기반 AWS Management Console에서 제공되는 것과 동일한 기능을 구현하는 AWS CLI 명령을 실행할 수 있습니다.
- [AWS CloudFormation](#) — AWS는 AWS CloudFormation 리소스를 모델링 및 설정하고, 빠르고 일관되게 프로비저닝하고, 수명 주기 전반에 걸쳐 관리할 수 있도록 지원합니다. 템플릿을 사용하여 리소스와 해당 종속성을 설명하고, 리소스를 개별적으로 관리하는 대신 스택으로 함께 시작하고 구성할 수 있습니다. 여러 AWS 계정 및 AWS 리전에서 스택을 관리하고 프로비저닝할 수 있습니다.
- [AWS Lambda](#) – AWS Lambda는 서버를 프로비저닝하거나 관리하지 않고도 코드를 실행하도록 지원하는 컴퓨팅 서비스입니다. Lambda는 필요 시에만 코드를 실행하며, 일일 몇 개의 요청에서 초당

수천 개의 요청까지 자동으로 규모를 조정합니다. 사용한 컴퓨팅 시간만큼만 비용을 지불하고, 코드가 실행되지 않을 때는 요금이 부과되지 않습니다.

- [Amazon SNS](#) – Amazon Simple Notification Service(SNS)는 웹 서버와 이메일 주소를 포함하여 게시자와 클라이언트 간에 메시지를 전달 또는 전송하는 것을 조정하고 관리합니다. 구독자는 구독하는 주제에 게시된 모든 메시지를 수신하며 주제에 대한 모든 구독자는 동일한 메시지를 수신합니다.
- [Amazon SQS](#) – Amazon Simple Queue Service(Amazon SQS)는 내구력 있고 가용성이 뛰어난 보안 호스팅 대기열을 제공하며 이를 통해 분산 소프트웨어 시스템과 구성 요소를 통합 및 분리할 수 있습니다. Amazon SQS는 표준 대기열과 FIFO 대기열을 모두 지원합니다.

에픽

중앙 리전에서 SQS 대기열 및 Lambda 함수를 생성합니다.

작업	설명	필요한 기술
Lambda 트리거를 사용하여 SQS 대기열을 생성합니다.	<p>AWS Management Console에 로그인하고 AWS Lambda 설명서의 Amazon SQS에서 Lambda 사용하기 튜토리얼의 지침을 사용하여 중앙 리전에 다음 리소스를 생성하십시오.</p> <ul style="list-style-type: none"> • Lambda 실행 역할 • Amazon S3 이벤트를 처리하는 Lambda 함수 • SQS 대기열 <p>참고: Lambda 함수의 이벤트 소스로 SQS 대기열을 구성해야 합니다.</p>	AWS DevOps, 클라우드 아키텍트

SNS 주제를 생성하고 각 필수 리전의 S3 버킷에 대한 이벤트 알림을 설정합니다.

작업	설명	필요한 기술
Amazon S3 이벤트 알림을 수신하도록 SNS 주제를 생성합니다.	<p>Amazon S3 이벤트 알림을 수신하려는 리전에서 SNS 주제를 생성합니다. 이에 대한 자세한 내용은 Amazon SNS 설명서의 SNS 주제 생성을 참조하십시오.</p> <p>중요: SNS 주제의 Amazon 리소스 이름(ARN)을 기록해 두어야 합니다.</p>	AWS DevOps, 클라우드 아키텍트
SNS 주제를 SQS 대기열에 구독합니다.	<p>중앙 리전에서 호스팅하는 SQS 대기열에 SNS 주제를 구독합니다. 이에 대한 자세한 내용은 Amazon SNS 설명서의 Amazon SNS 주제에 구독 설정을 참조하십시오.</p>	AWS DevOps, 클라우드 아키텍트
SNS 주제의 액세스 정책을 업데이트합니다.	<ol style="list-style-type: none"> 1. Amazon SNS 콘솔을 열고 주제를 선택한 다음 이전에 생성한 SNS 주제를 선택합니다. 2. 편집을 선택한 다음 액세스 정책 - 선택 사항 섹션을 펼칩니다. 3. 다음 액세스 정책을 SNS 주제에 연결하여 Amazon S3에 대한 sns:publish 권한을 허용한 다음 저장을 선택합니다. <pre>{</pre>	AWS DevOps, 클라우드 아키텍트

작업	설명	필요한 기술
	<pre> "Version": "2012-10-17", "Statement": [{ "Sid": "0", "Effect": "Allow", "Principal": { "Service": "s3.amazonaws.com" }, "Action": "sns:Publish", "Resource": "arn:aws:sns:us-we st-2::s3Events-SNS Topic-us-west-2" }] } </pre>	
<p>리전의 각 S3 버킷에 대한 알림을 설정합니다.</p>	<p>리전의 각 S3 버킷에 대한 이벤트 알림을 설정합니다. 이에 대한 자세한 내용은 Amazon S3 설명서에서 Amazon S3 콘솔을 사용하여 이벤트 알림 활성화 및 구성을 참조하십시오.</p> <p>참고: 대상 섹션에서 SNS 주제를 선택하고 이전에 생성한 SNS 주제의 ARN을 지정합니다.</p>	<p>AWS DevOps, 클라우드 아키텍트</p>
<p>모든 필수 리전에 대해 이 에픽을 반복하십시오.</p>	<p>중요: 중앙 리전을 포함하여 Amazon S3 이벤트 알림을 수신하려는 각 리전에 대해 이 에픽의 작업을 반복합니다.</p>	<p>AWS DevOps, 클라우드 아키텍트</p>

관련 리소스

- [액세스 정책 구성](#)(Amazon SQS 설명서)
- [SQS 대기열을 이벤트 소스로 구성](#)(AWS Lambda 설명서)
- [Lambda 함수를 시작하도록 SQS 대기열 구성](#)(Amazon SQS 설명서)
- [AWS::Lambda::Function 리소스](#) (AWS CloudFormation 설명서)

데이터를 Apache Parquet으로 변환하기 위한 세 가지 AWS Glue ETL 작업 유형

작성자: Adnan Alvee(AWS), Karthikeyan Ramachandran 및 Nith Govindasivan(AWS)

환경: PoC 또는 파일럿

기술: 분석

워크로드: 기타 모든 워크로드

AWS 서비스: AWS Glue

요약

Amazon Web Services(AWS) 클라우드에서 AWS Glue는 완전관리형 추출, 전환, 적재(ETL) 서비스입니다. AWS Glue를 사용하면 경제적으로 데이터를 분류, 정리, 보강하고, 다양한 데이터 스토어와 데이터 스트림 간에 안정적으로 이동할 수 있습니다.

이 패턴은 AWS Glue에서 다양한 작업 유형을 제공하며 세 가지 스크립트를 사용하여 ETL 작업 작성을 보여줍니다.

AWS Glue를 사용하여 Python 셸 환경에서 ETL 작업을 작성할 수 있습니다. 관리형 Apache Spark 환경에서 Python (PySpark) 또는 Scala를 사용하여 일괄 처리 및 스트리밍 ETL 작업을 모두 생성할 수도 있습니다. ETL 작업 작성을 시작하기 위해 이 패턴은 Python, 셸 PySpark, Scala를 사용하는 일괄 ETL 작업에 중점을 둡니다. Python 셸 작업은 컴퓨팅 파워가 덜 필요한 워크로드를 위한 것입니다. 관리형 Apache Spark 환경은 높은 컴퓨팅 파워가 필요한 워크로드에 적합합니다.

Apache Parquet은 효율적인 압축 및 인코딩 체계를 지원하도록 구축되었습니다. 데이터를 컬럼 방식으로 저장하므로 분석 워크로드의 속도를 높일 수 있습니다. 데이터를 Parquet으로 변환하면 장기적으로 스토리지 공간, 비용 및 시간을 절감할 수 있습니다. Parquet에 대해 자세히 알아보려면 블로그 게시물 [Apache Parquet: 오픈 소스 컬럼형 데이터 형식으로 영웅이 되는 방법을 참조하십시오.](#)

사전 조건 및 제한 사항

사전 조건

- AWS 자격 증명 및 액세스 관리 (IAM) 역할 (역할이 없는 경우 추가 정보 섹션 참조)

아키텍처

대상 기술 스택

- Glue
- Amazon Simple Storage Service(S3)
- Apache Parquet

자동화 및 규모 조정

- [AWS Glue 워크플로](#)는 ETL 파이프라인의 완전 자동화를 지원합니다.
- 데이터 처리 단위(DPU)의 수 또는 작업자 유형을 변경하여 수평 및 수직으로 규모를 조정할 수 있습니다.

도구

AWS 서비스

- [Amazon Simple Storage Service\(S3\)](#)는 원하는 양의 데이터를 저장, 보호 및 검색하는 데 도움이 되는 클라우드 기반 객체 스토리지 서비스입니다.
- [AWS Glue](#)는 다양한 데이터 스토어와 데이터 스트림 간에 데이터를 분류, 정리, 보강하고 이동할 수 있는 완전관리형 ETL 서비스입니다.

기타 도구

- [Apache Parquet](#)는 스토리지 및 검색을 위해 설계된 오픈 소스 열 지향 데이터 파일 형식입니다.

구성

다음 설정을 사용하여 AWS Glue ETL의 컴퓨팅 파워를 구성합니다. 비용을 줄이려면 이 패턴으로 제공되는 워크로드를 실행할 때 최소 설정을 사용합니다.

- Python 셸 – 1DPU를 사용하여 16GB의 메모리를 활용하거나 0.0625DPU를 사용하여 1GB의 메모리를 활용할 수 있습니다. 이 패턴은 AWS Glue 콘솔의 기본값인 0.0625 DPU를 사용합니다.
- Python 또는 Scala for Spark – 콘솔에서 Spark 관련 작업 유형을 선택하면 AWS Glue는 기본적으로 10개의 작업자와 G.1X 작업자 유형을 사용합니다. 이 패턴은 허용된 최소 수인 두 개의 작업자를 사용하며, 표준 작업자 유형은 충분하고 비용 효율적입니다.

다음 표에는 Apache Spark 환경의 다양한 AWS Glue 작업자 유형이 나와 있습니다. Python 셸 작업은 Apache Spark 환경을 사용하여 Python을 실행하지 않으므로 테이블에 포함되지 않습니다.

	표준	G.1X	G.2X
vCPU	4	4	8
메모리	16 GB	16 GB	32GB
디스크 공간	50GB	64GB	128GB
작업자당 실행기	2	1	1

코드

IAM 역할 및 파라미터 구성을 포함하여 이 패턴에 사용되는 코드는 추가 정보 섹션을 참조하십시오.

에픽

데이터 업로드

작업	설명	필요한 기술
새 S3 버킷이나 기존 S3 버킷에 데이터를 업로드합니다.	계정에서 기존 S3 버킷을 생성하거나 사용합니다. 첨부 파일 섹션에서 sample_data.csv 파일을 업로드하고 S3 버킷과 접두사 위치를 기록해 둡니다.	일반 AWS

AWS Glue 작업 생성 및 실행

작업	설명	필요한 기술
AWS Glue 작업을 생성합니다.	AWS Glue 콘솔의 ETL 섹션에서 AWS Glue 작업을 추가합니다. 적절한 작업 유형, AWS Glue 버전, 해당 DPU/작업자	개발자, 클라우드 또는 데이터

작업	설명	필요한 기술
	<p>유형 및 작업자 수를 선택합니다. 자세한 내용은 구성 섹션을 참조하십시오.</p>	
<p>입력 및 출력 위치를 변경합니다.</p>	<p>AWS Glue 작업에 해당하는 코드를 복사하고 데이터 업로드 에픽에서 기록해 둔 입력 및 출력 위치를 변경합니다.</p>	<p>개발자, 클라우드 또는 데이터</p>

작업	설명	필요한 기술
<p>파라미터를 구성합니다.</p>	<p>추가 정보 섹션에 제공된 스니펫을 사용하여 ETL 작업의 파라미터를 설정할 수 있습니다. AWS Glue는 내부적으로 4개의 인수 이름을 사용합니다.</p> <ul style="list-style-type: none"> • --conf • --debug • --mode • --JOB_NAME <p>--JOB_NAME 파라미터는 AWS Glue 콘솔에 명시적으로 입력해야 합니다. 작업, 작업 편집, 보안 구성, 스크립트 라이브러리 및 작업 파라미터(선택 사항)를 선택합니다. --JOB_NAME 을 키로 입력하고 값을 제공합니다. AWS Command Line Interface(AWS CLI) 또는 AWS Glue API를 사용하여 이 파라미터를 설정할 수도 있습니다. 이 --JOB_NAME 파라미터는 Spark에서 사용되며 Python 셸 환경 작업에는 필요하지 않습니다.</p> <p>모든 파라미터 이름 앞에 --을 추가해야 합니다. 그렇지 않으면 코드가 작동하지 않습니다. 예를 들어 코드 스니펫의 경우 --input_loc 및 --output_loc 를 사용하여 위</p>	<p>개발자, 클라우드 또는 데이터</p>

작업	설명	필요한 기술
	치 파라미터를 간접적으로 호출해야 합니다.	
ETL 작업을 실행합니다.	작업을 실행하고 출력을 확인합니다. 소스 파일보다 공간이 얼마나 줄어들었는지 확인합니다.	개발자, 클라우드 또는 데이터

관련 리소스

참조

- [Apache Spark](#)
- [AWS Glue: 작동 방식](#)
- [AWS Glue 요금](#)

튜토리얼 및 동영상

- [AWS Glue란 무엇입니까?](#)

추가 정보

IAM 역할

AWS Glue 작업을 생성할 때 다음 코드 스니펫에 표시된 권한이 있는 기존 IAM 역할을 사용하거나 새 역할을 사용할 수 있습니다.

새 역할을 생성하려면 다음 YAML 코드를 사용합니다.

```
# (c) 2022 Amazon Web Services, Inc. or its affiliates. All Rights Reserved. This AWS
Content is provided subject to the terms of the AWS Customer
# Agreement available at https://aws.amazon.com/agreement/ or other written agreement
between Customer and Amazon Web Services, Inc.

AWSTemplateFormatVersion: "2010-09-09"

Description: This template will setup IAM role for AWS Glue service.
```

```

Resources:
  rGlueRole:
    Type: AWS::IAM::Role
    Properties:
      AssumeRolePolicyDocument:
        Version: "2012-10-17"
        Statement:
          - Effect: "Allow"
            Principal:
              Service:
                - "glue.amazonaws.com"
            Action:
              - "sts:AssumeRole"
      ManagedPolicyArns:
        - arn:aws:iam::aws:policy/service-role/AWSGlueServiceRole
      Policies:
        - PolicyName: !Sub "${AWS::StackName}-s3-limited-read-write-inline-policy"
          PolicyDocument:
            Version: "2012-10-17"
            Statement:
              - Effect: Allow
                Action:
                  - "s3:PutObject"
                  - "s3:GetObject"
                Resource: "arn:aws:s3:::*/*"
      Tags:
        - Key : "Name"
          Value : !Sub "${AWS::StackName}"

Outputs:
  oGlueRoleName:
    Description: AWS Glue IAM role
    Value:
      Ref: rGlueRole
    Export:
      Name: !Join [ ":", [ !Ref "AWS::StackName", rGlueRole ] ]

```

AWS Glue(Python 셸)

Python 코드는 Pandas와 PyArrow 라이브러리를 사용하여 데이터를 Parquet으로 변환합니다. Pandas 라이브러리는 이미 사용할 수 있습니다. PyArrow 라이브러리는 한 번만 실행되므로 패턴을 실행하면 다운로드됩니다. 쉘 파일을 사용하여 라이브러리로 PyArrow 변환하고 파일을 라이브러리 패키지로 제

공할 수 있습니다. 훔 파일 패키징에 대한 자세한 내용은 [자체 Python 라이브러리 제공](#)을 참조하십시오.

AWS Glue Python 셸 파라미터

```
from awsglue.utils import getResolvedOptions

args = getResolvedOptions(sys.argv, ["input_loc", "output_loc"])
```

AWS Glue Python 셸 코드

```
from io import BytesIO
import pandas as pd
import boto3
import os
import io
import site
from importlib import reload
from setuptools.command import easy_install
install_path = os.environ['GLUE_INSTALLATION']
easy_install.main( ["--install-dir", install_path, "pyarrow"] )
reload(site)
import pyarrow

input_loc = "bucket-name/prefix/sample_data.csv"
output_loc = "bucket-name/prefix/"

input_bucket = input_loc.split('/', 1)[0]
object_key = input_loc.split('/', 1)[1]

output_loc_bucket = output_loc.split('/', 1)[0]
output_loc_prefix = output_loc.split('/', 1)[1]

s3 = boto3.client('s3')
obj = s3.get_object(Bucket=input_bucket, Key=object_key)
df = pd.read_csv(io.BytesIO(obj['Body'].read()))

parquet_buffer = BytesIO()
```

```
s3_resource = boto3.resource('s3')
df.to_parquet(parquet_buffer, index=False)
s3_resource.Object(output_loc_bucket, output_loc_prefix + 'data' +
'.parquet').put(Body=parquet_buffer.getvalue())
```

Python을 사용한 AWS Glue Spark 작업

Python에서 AWS Glue Spark 작업 유형을 사용하려면 작업 유형으로 Spark를 선택합니다. 작업 스타트업 시간이 개선된 Spark 3.1, Python 3(Glue 버전 3.0)을 AWS Glue 버전으로 선택합니다.

AWS Glue Python 파라미터

```
from awsglue.utils import getResolvedOptions

args = getResolvedOptions(sys.argv, ["JOB_NAME", "input_loc", "output_loc"])
```

Python 코드를 사용한 AWS Glue Spark 작업

```
import sys
from pyspark.context import SparkContext
from awsglue.context import GlueContext
from awsglue.transforms import *
from awsglue.dynamicframe import DynamicFrame
from awsglue.utils import getResolvedOptions
from awsglue.job import Job

sc = SparkContext()
glueContext = GlueContext(sc)
spark = glueContext.spark_session
job = Job(glueContext)

input_loc = "bucket-name/prefix/sample_data.csv"
output_loc = "bucket-name/prefix/"

inputDyf = glueContext.create_dynamic_frame_from_options(\
    connection_type = "s3", \
    connection_options = {
        "paths": [input_loc]}, \
    format = "csv",
    format_options={
        "withHeader": True,
        "separator": ",",
```

```
})
```

```
outputDF = glueContext.write_dynamic_frame.from_options(\
  frame = inputDyf, \
  connection_type = "s3", \
  connection_options = {"path": output_loc \
    }, format = "parquet")
```

압축된 대용량 파일이 많은 경우(예: 각각 약 3MB인 파일 1,000개), 다음 코드와 같이 recurse 파라미터와 함께 compressionType 파라미터를 사용하여 접두사 내에서 사용 가능한 모든 파일을 읽습니다.

```
input_loc = "bucket-name/prefix/"
output_loc = "bucket-name/prefix/"

inputDyf = glueContext.create_dynamic_frame_from_options(
  connection_type = "s3",
  connection_options = {"paths": [input_loc],
    "compressionType": "gzip", "recurse" : "True",
    },
  format = "csv",
  format_options={"withHeader": True, "separator": ","}
)
```

압축된 작은 파일(예: 각각 약 133KB인 파일 1,000개) 이 많은 경우 groupFiles 파라미터를 compressionType 및 recurse 파라미터와 함께 사용하십시오. groupFiles 파라미터는 작은 파일을 여러 개의 큰 파일로 그룹화하고 groupSize 파라미터는 지정된 크기(예: 1MB)로의 그룹화를 제어합니다. 다음 코드 스니펫은 코드 내에서 이러한 파라미터를 사용하는 예를 제공합니다.

```
input_loc = "bucket-name/prefix/"
output_loc = "bucket-name/prefix/"

inputDyf = glueContext.create_dynamic_frame_from_options(
  connection_type = "s3",
  connection_options = {"paths": [input_loc],
    "compressionType": "gzip", "recurse" : "True",
    "groupFiles" : "inPartition",
    "groupSize" : "1048576",
    },
  format = "csv",
  format_options={"withHeader": True, "separator": ","}
```

)

워커 노드를 변경하지 않고도 AWS Glue 작업에서 여러 파일(크든 작든, 압축이 있든 없든)을 읽고 Parquet 형식으로 대상에 쓸 수 있습니다.

Scala를 사용한 AWS Glue Spark 작업

Scala에서 AWS Glue Spark 작업 유형을 사용하려면 작업 유형으로 Spark를 선택하고 Scala로 언어를 선택합니다. 작업 스타트업 시간이 개선된 Spark 3.1, Scala 2(Glue 버전 3.0)를 AWS Glue 버전으로 선택합니다. 스토리지 스페이스를 절약하기 위해 다음 AWS Glue with Scala 샘플도 이 applyMapping 기능을 사용하여 데이터 유형을 변환합니다.

AWS Glue Scala 파라미터

```
import com.amazonaws.services.glue.util.GlueArgParser val args =
  GlueArgParser.getResolvedOptions(sysArgs, Seq("JOB_NAME", "inputLoc",
    "outputLoc")).toArray)
```

Scala 코드를 사용한 AWS Glue Spark 작업

```
import com.amazonaws.services.glue.GlueContext
import com.amazonaws.services.glue.MappingSpec
import com.amazonaws.services.glue.DynamicFrame
import com.amazonaws.services.glue.errors.CallSite
import com.amazonaws.services.glue.util.GlueArgParser
import com.amazonaws.services.glue.util.Job
import com.amazonaws.services.glue.util.JsonOptions
import org.apache.spark.SparkContext
import scala.collection.JavaConverters._

object GlueScalaApp {
  def main(sysArgs: Array[String]) {

    @transient val spark: SparkContext = SparkContext.getOrCreate()
    val glueContext: GlueContext = new GlueContext(spark)

    val inputLoc = "s3://bucket-name/prefix/sample_data.csv"
    val outputLoc = "s3://bucket-name/prefix/"

    val readCSV = glueContext.getSource("csv", JsonOptions(Map("paths" ->
      Set(inputLoc))))).getDynamicFrame()
```

```
val applyMapping = readCSV.applyMapping(mappings = Seq(("_c0", "string", "date",
"string"), ("_c1", "string", "sales", "long"),
("_c2", "string", "profit", "double")), caseSensitive = false)

val formatPartition = applyMapping.toDF().coalesce(1)

val dynamicFrame = DynamicFrame(formatPartition, glueContext)

val dataSink = glueContext.getSinkWithFormat(
  connectionType = "s3",
  options = JsonOptions(Map("path" -> outputLoc )),
  transformationContext = "dataSink", format =
"parquet").writeDynamicFrame(dynamicFrame)
}
}
```

첨부

이 문서와 관련된 추가 콘텐츠에 액세스하려면 [attachment.zip](#) 파일의 압축을 풉니다.

Amazon Athena와 Amazon을 사용하여 Amazon Redshift 감사 로그를 시각화합니다. QuickSight

작성자: Sanket Sirsikar(AWS), Gopal Krishna Bhatia(AWS)

환경: PoC 또는 파일럿

기술: 분석, 빅데이터, 데이터 레이크

AWS 서비스: 아마존 아테나, 아마존 레드시프트, 아마존 S3, 아마존 QuickSight

요약

Amazon Web Services(AWS) 클라우드의 데이터베이스 운영에서 보안은 필수적인 부분입니다. 조직은 데이터베이스 사용자 활동 및 연결을 모니터링하여 잠재적인 보안 사고 및 위험을 탐지하도록 해야 합니다. 이 패턴은 데이터베이스 감사라고 알려진 프로세스인 보안 및 문제 해결 목적으로 데이터베이스를 모니터링하는 데 도움이 됩니다.

이 패턴은 Amazon Redshift 로그를 감사하는 데 도움이 되는 Amazon의 보고 대시보드에 대한 Amazon Athena 테이블 및 뷰 생성을 자동화하는 SQL 스크립트를 제공합니다. QuickSight 이렇게 하면 데이터베이스 활동 모니터링을 담당하는 사용자가 데이터 보안 기능에 편리하게 액세스할 수 있습니다.

사전 조건 및 제한 사항

사전 조건

- 활성 상태의 AWS 계정.
- 기존 Amazon Redshift 클러스터입니다. 이에 대한 자세한 내용은 Amazon Redshift 설명서의 [Amazon Redshift 클러스터 생성](#)을 참조하세요.
- 기존 Athena 워크그룹에 액세스할 수 있습니다. 자세한 내용은 Amazon Athena 설명서의 [작업 그룹 작동 방식](#)을 참조하세요.
- 필수 AWS ID 및 Access Management(IAM) 권한이 있는 기존 Amazon Simple Storage Service(S3) 소스 버킷입니다. 자세한 내용은 Amazon Redshift 설명서의 [데이터베이스 감사 로깅](#)에서 [Amazon Redshift 감사 로깅을 위한 버킷 권한](#)을 참조하세요.

아키텍처

기술 스택

- Athena
- Amazon Redshift
- Amazon S3
- QuickSight

도구

- [Amazon Athena](#) – Amazon Athena는 표준 SQL을 사용해 Amazon S3에 저장된 데이터를 간편하게 분석할 수 있는 대화식 쿼리 서비스입니다.
- [Amazon QuickSight](#) — QuickSight 확장 가능한 서버리스, 내장 가능한 기계 학습 기반 비즈니스 인텔리전스 (BI) 서비스입니다.
- [Amazon Redshift](#) – Amazon Redshift는 페타바이트 규모의 엔터프라이즈급 완전 관리형 데이터 웨어하우징 서비스입니다.
- [Amazon S3](#) – Amazon Simple Storage Service(S3)는 인터넷 스토리지 서비스입니다.

에픽

Amazon Redshift 클러스터 구성

작업	설명	필요한 기술
Amazon Redshift 클러스터에 대한 감사 로깅을 활성화합니다.	<ol style="list-style-type: none"> 1. AWS Management Console에 로그인하고 Amazon Redshift 콘솔을 열고 클러스터를 선택한 다음 로깅을 활성화하려는 클러스터를 선택합니다. 2. 속성 탭을 선택한 다음 Amazon Redshift 설명서 	DBA, 데이터 엔지니어

작업	설명	필요한 기술
	<p>의 콘솔을 사용하여 감사 구성의 지침에 따라 감사를 활성화합니다.</p>	

작업	설명	필요한 기술
<p>Amazon Redshift 클러스터 파라미터 그룹에서 로깅을 활성화합니다.</p>	<p>AWS Management Console, Amazon Redshift API 참조 또는 AWS Command Line Interface(AWS CLI)를 사용하여 연결 로그, 사용자 로그 및 사용자 활동 로그에 대한 감사를 동시에 활성화할 수 있습니다.</p> <p>사용자 활동 로그를 감사하려면 <code>enable_user_activity_logging</code> 데이터베이스 파라미터를 활성화해야 합니다. 연결된 파라미터를 제외하고 감사 로깅 기능만 활성화하면 데이터베이스 감사 로그가 사용자 작업 로그를 제외한 연결 및 사용자 로그 정보를 기록합니다. <code>enable_user_activity_logging</code> 파라미터는 기본적으로 활성화되지 않지만 <code>false</code>에서 <code>true</code>로 변경하여 활성화할 수 있습니다.</p> <p>중요: <code>user_activity_logging</code> 파라미터가 활성화된 새 클러스터 파라미터 그룹을 생성하여 Amazon Redshift 클러스터에 연결해야 합니다. 이에 대한 자세한 내용은 Amazon Redshift 설명서의 클러스터 수정을 참조하세요.</p>	<p>DBA, 데이터 엔지니어</p>

작업	설명	필요한 기술
	<p>이 작업에 대한 자세한 내용은 Amazon Redshift 설명서에서 Amazon Redshift 파라미터 그룹 및 콘솔을 사용한 감사 구성을 참조하세요.</p>	
<p>Amazon Redshift 클러스터 로깅에 대한 S3 버킷의 권한을 구성합니다.</p>	<p>로깅을 사용하면 Amazon Redshift가 로깅 정보를 수집한 후 S3 버킷에 저장된 로그 파일로 업로드합니다. 기존 S3 버킷을 사용하거나 새 버킷을 생성할 수 있습니다.</p> <p>중요: Amazon Redshift에 S3 버킷에 액세스하는 데 필요한 IAM 권한이 있는지 확인합니다. 이에 대한 자세한 내용은 Amazon Redshift 설명서의 데이터베이스 감사 로깅에서 Amazon Redshift 감사 로깅을 위한 버킷 권한을 참조하세요.</p>	<p>DBA, 데이터 엔지니어</p>

Athena 테이블 및 뷰 생성

작업	설명	필요한 기술
<p>Athena 테이블과 뷰를 생성하여 S3 버킷에서 Amazon Redshift 감사 로그 데이터를 쿼리할 수 있습니다.</p>	<p>Amazon Athena 콘솔을 열고 <code>AuditLogging.sql</code> SQL 스크립트(첨부)의 데이터 정의 언어(DDL) 쿼리를 사용하여 사용자 활동 로그, 사용자 로그 및 연결 로그에 대한 테이블과 뷰를 생성합니다.</p>	<p>데이터 엔지니어</p>

작업	설명	필요한 기술
	자세한 내용 및 지침은 Amazon Athena 워크숍의 테이블 생성 및 쿼리 실행 자습서를 참조하세요.	

대시보드에서 로그 모니터링을 설정합니다. QuickSight

작업	설명	필요한 기술
Athena를 데이터 소스로 사용하여 QuickSight 대시보드를 생성합니다.	Amazon QuickSight 콘솔을 열고 Amazon Athena 워크숍의 QuickSight Athena를 사용하여 시각화하기 자습서의 지침에 따라 QuickSight 대시보드를 생성합니다.	DBA, 데이터 엔지니어

관련 리소스

- [Athena에서 테이블 생성 및 쿼리 실행](#)
- [QuickSight Athena를 사용하여 시각화하기](#)

첨부

이 문서와 관련된 추가 콘텐츠에 액세스하려면 [attachment.zip](#) 파일의 압축을 풉니다.

Amazon을 사용하여 모든 AWS 계정에 대한 IAM 자격 증명 보고서를 시각화합니다. QuickSight

작성자: Parag Nagwekar (AWS) 및 Arun Chandapillai (AWS)

코드 리포지토리: [IAM 자격 증명 보고서를 조직 전체에 대한 가시성 확보](#)

환경: 프로덕션

기술: 분석, 자문, 관리 및 거버넌스, 보안, ID, 규정 준수

워크로드: 기타 모든 워크로드

AWS 서비스: 아마존 아테나, AWS, CloudFormation 아마존 EventBridge, AWS ID 및 액세스 관리, 아마존 QuickSight

요약

경고: IAM 사용자는 장기 자격 증명을 보유하므로 보안상 위험이 있습니다. 이 위험을 줄이려면 이러한 사용자에게 작업을 수행하는 데 필요한 권한만 제공하고 더 이상 필요하지 않을 경우 이러한 사용자를 제거하는 것이 좋습니다.

AWS Identity and Access Management(IAM) 보안 인증 보고서를 사용하면 조직의 보안, 감사, 규정 준수 요구 사항을 충족하는 데 도움이 될 수 있습니다. [보안 인증 보고서](#)는 AWS 계정의 모든 사용자 목록을 제공하며 암호, 액세스 키, 다중 인증(MFA) 디바이스 등 보안 인증의 상태를 보여줍니다. [AWS Organizations](#)에서 관리하는 여러 AWS 계정에 대한 보안 인증 보고서를 사용할 수 있습니다.

이 패턴에는 Amazon QuickSight 대시보드를 사용하여 조직의 모든 AWS 계정에 대한 IAM 자격 증명 보고서를 생성하고 공유하는 데 도움이 되는 단계와 코드가 포함되어 있습니다. 대시보드를 조직의 이해 관계자와 공유할 수 있습니다. 보고서는 조직이 다음과 같은 목표 비즈니스 성과를 달성하는 데 도움이 될 수 있습니다.

- IAM 사용자와 관련된 보안 사건을 식별합니다.
- IAM 사용자를 AWS Single Sign-On(SSO) 인증으로 실시간 마이그레이션하는 과정을 추적합니다.
- IAM 사용자가 액세스한 AWS 리전 추적

- 규정 준수 유지
- 다른 이해관계자와 정보 공유

사전 조건 및 제한 사항

사전 조건

- 활성 상태의 AWS 계정
- 멤버 계정이 있는 [조직](#)
- Organizations의 계정에 액세스할 수 있는 권한이 있는 [IAM 역할](#)
- [설치](#) 및 [구성](#)된 AWS Command Line Interface(AWS CLI) 버전 2
- [Amazon QuickSight 엔터프라이즈 에디션 구독](#)

아키텍처

기술 스택

- Amazon Athena
- 아마존 EventBridge
- 아마존 QuickSight
- Amazon Simple Storage Service(S3)
- Glue
- AWS Identity and Access Management(IAM)
- AWS Lambda
- AWS Organizations

대상 아키텍처

다음 다이어그램은 여러 AWS 계정에서 IAM 보안 인증 보고서 데이터를 캡처하는 워크플로를 설정하는 아키텍처를 보여줍니다.

1. EventBridge Lambda 함수를 매일 호출합니다.

2. Lambda 함수는 조직 전체의 모든 AWS 계정에서 IAM 역할을 맡습니다. 그런 다음 함수는 IAM 보안 인증 보고서를 생성하고 보고서 데이터를 중앙 집중식 S3 버킷에 저장합니다. S3 버킷에서 암호화를 활성화하고 퍼블릭 액세스를 비활성화해야 합니다.
3. AWS Glue 크롤러는 매일 S3 버킷을 크롤링하고 그에 따라 Athena 테이블을 업데이트합니다.
4. QuickSight 자격 증명 보고서에서 데이터를 가져와 분석하고 이해 관계자가 시각화하고 공유할 수 있는 대시보드를 구축합니다.

도구

서비스

- [Amazon Athena](#)는 표준 SQL을 사용해 Amazon S3에 저장된 데이터를 간편하게 분석할 수 있는 대화식 쿼리 서비스입니다.
- [EventBridgeAmazon](#)은 다양한 소스의 실시간 데이터와 애플리케이션을 연결하는 데 도움이 되는 서버리스 이벤트 버스 서비스입니다. Lambda 함수, API 대상을 사용하는 HTTP 간접 호출 엔드포인트 또는 다른 AWS 계정의 이벤트 버스를 예로 들 수 있습니다.
- [QuickSightAmazon](#)은 단일 대시보드에서 데이터를 시각화, 분석 및 보고하는 데 도움이 되는 클라우드 규모의 비즈니스 인텔리전스 (BI) 서비스입니다.
- [AWS Identity and Access Management\(IAM\)](#)를 사용하면 사용자에 대해 인증 및 권한 부여를 제어함으로써 AWS 리소스에 대한 액세스를 안전하게 관리할 수 있습니다.
- [AWS Lambda](#)는 서버를 프로비저닝하거나 관리할 필요 없이 코드를 실행하는 데 도움이 되는 컴퓨팅 서비스입니다. 필요할 때만 코드를 실행하며 자동으로 확장이 가능하므로 사용한 컴퓨팅 시간만큼만 비용을 지불합니다.

코드

이 패턴의 코드는 리포지토리에서 사용할 수 있습니다. GitHub [getiamcredsreport-allaccounts-org](#) 이 리포지토리의 코드를 사용하여 Organizations의 AWS 계정 전반에 걸쳐 IAM 보안 인증 보고서를 생성하고 중앙 위치에 저장할 수 있습니다.

에픽

인프라 설정

작업	설명	필요한 기술
Amazon QuickSight 엔터프라이즈 에디션을 설정하세요.	<ol style="list-style-type: none"> AWS 계정에서 Amazon QuickSight 엔터프라이즈 에디션을 활성화하십시오. 자세한 내용은 QuickSight 설명서의 Amazon QuickSight 내부 사용자 액세스 관리를 참조하십시오. 대시보드 권한을 부여하려면 사용자의 Amazon 리소스 이름 (ARN) 을 QuickSight 가져오십시오. 	AWS 관리자, AWS DevOps, 클라우드 관리자, 클라우드 아키텍트
QuickSight 아마존을 Amazon S3 및 Athena와 통합합니다.	AWS CloudFormation 스택을 QuickSight 배포하기 전에 Amazon S3와 Athena를 사용할 수 있는 권한을 부여해야 합니다.	AWS 관리자, AWS DevOps, 클라우드 관리자, 클라우드 아키텍트

인프라 배포

작업	설명	필요한 기술
GitHub 리포지토리를 복제합니다.	<ol style="list-style-type: none"> 다음 명령을 실행하여 GitHub getiamcredsreport-allaccounts-org 리포지토리를 로컬 시스템에 복제합니다. <code>git clone https://github.com/aws-samples/getia</code> 	AWS 관리자

작업	설명	필요한 기술
	mcredsreport-allac counts-org	

작업	설명	필요한 기술
인프라를 배포합니다.	<ol style="list-style-type: none"> 1. AWS Management Console에 로그인하고 CloudFormation 콘솔을 엽니다. 2. 탐색 창에서 스택 생성을 선택한 다음 새 리소스 사용 (표준)을 선택합니다. 3. 리소스 식별 페이지에서 다음을 선택합니다. 4. 템플릿 지정 페이지의 템플릿 소스에서 템플릿 파일 업로드를 선택합니다. 5. 파일 선택을 선택하고 복제된 GitHub 저장소에서 Cloudformation-createdcredrepo.yaml 파일을 선택한 후 다음을 선택합니다. 6. 파라미터에서 IAM 역할로 IAMRoleName 을 업데이트합니다. 이 역할을 조직의 모든 계정에서 Lambda가 맡게 하려는 IAM 역할이어야 합니다. 이 역할은 보안 인증 보고서를 생성합니다. 참고: 스택 생성의 이 단계에서 역할이 모든 계정에 존재할 필요는 없습니다. 7. 파라미터에서 Lambda가 모든 계정의 자격 증명을 저장할 수 있는 S3 버킷의 이름으로 S3BucketName 을 업데이트합니다. 	AWS 관리자

작업	설명	필요한 기술
	<p>8. 스택 이름에 스택 이름을 입력합니다.</p> <p>9. 제출을 선택합니다.</p> <p>10 Lambda 함수의 역할 이름을 기록합니다.</p>	
IAM 권한 정책을 생성합니다.	<p>다음 권한을 사용하여 조직의 모든 AWS 계정에 대한 IAM 정책을 생성합니다.</p> <pre data-bbox="594 667 1029 1381"> { "Version": "2012-10-17", "Statement": [{ "Effect": "Allow", "Action": ["iam:GenerateCredentialReport", "iam:GetCredentialReport"], "Resource": "*" }] } </pre>	AWS DevOps, 클라우드 관리자, 클라우드 아키텍트, 데이터 엔지니어

작업	설명	필요한 기술
<p>신뢰 정책이 있는 IAM 역할을 생성합니다.</p>	<ol style="list-style-type: none"> 1. AWS 계정에 대한 IAM 역할을 생성하고 이전 단계에서 생성한 권한 정책을 연결합니다. 2. IAM 역할에 다음 신뢰 정책을 역할에 연결합니다. <pre data-bbox="597 583 1026 1419"> { "Version": "2012-10-17", "Statement": [{ "Effect": "Allow", "Principal": { "AWS": ["arn:aws:iam::<MasterAccountID>:role/<LambdaRole>"] }, "Action": "sts:AssumeRole" }] } </pre> <p>중요: <code>arn:aws:iam::<MasterAccountID>:role/<LambdaRole></code> 을 이전에 언급한 Lambda 역할의 ARN으로 바꿉니다.</p> <p>참고: 조직은 일반적으로 자동화를 사용하여 AWS 계정</p>	<p>클라우드 관리자, 클라우드 아키텍트, AWS 관리자</p>

작업	설명	필요한 기술
	<p>의 IAM 역할을 생성합니다. 가능한 경우 이 자동화를 사용하는 것이 좋습니다. 또는 코드 리포지토리의 CreateRoleforOrg.py 스크립트를 사용할 수도 있습니다. 스크립트에는 모든 AWS 계정에서 IAM 정책 및 역할을 생성할 권한이 있는 기존 관리자 역할 또는 기타 IAM 역할이 필요합니다.</p>	
<p>데이터를 QuickSight 시각화하도록 Amazon을 구성합니다.</p>	<ol style="list-style-type: none"> 1. 자격 QuickSight증명으로 로그인합니다. 2. Athena를 사용(iamcredreportdb 데이터베이스 "cfn_iamcredreport" 및 테이블 사용)하여 데이터 세트를 생성한 다음 자동으로 데이터 세트를 새로 고칩니다. 3. 에서 분석을 QuickSight 생성하세요. 4. QuickSight 대시보드 만들기. 	<p>AWS DevOps, 클라우드 관리자, 클라우드 아키텍트, 데이터 엔지니어</p>

추가 정보

추가 고려 사항

다음을 고려하세요.

- 를 CloudFormation 사용하여 인프라를 배포한 후에는 Lambda와 AWS Glue가 일정에 따라 실행될 때까지 Amazon S3에서 생성되고 Athena에서 분석한 보고서를 받을 때까지 기다릴 수 있습니다. 또

는 Lambda를 수동으로 실행하여 Amazon S3에서 보고서를 가져온 다음 AWS Glue 크롤러를 실행하여 데이터에서 생성된 Athena 테이블을 가져올 수 있습니다.

- QuickSight 비즈니스 요구 사항에 따라 데이터를 분석하고 시각화할 수 있는 강력한 도구입니다. 에서 [QuickSight 매개 변수를](#) 사용하여 선택한 데이터 필드를 기반으로 위젯 데이터를 제어할 수 있습니다. 또한 QuickSight 분석을 통해 데이터셋에서 매개변수 (예: 계정, 날짜, 사용자 필드, partition_0, partition_1, 등) 를 생성하여 계정, 날짜, 사용자에 대한 매개변수에 대한 제어를 추가할 수 있습니다. user
- QuickSight [대시보드를 직접 구축하려면 AWS Workshop Studio 웹 사이트의 워크샵을 참조하십시오 QuickSight .](#)
- 샘플 QuickSight 대시보드를 보려면 GitHub [getiamcredsreport-allaccounts-org](#) 코드 리포지토리를 참조하십시오.

목표 비즈니스 성과

이 패턴을 사용하여 다음과 같은 목표 비즈니스 성과를 달성할 수 있습니다.

- IAM 사용자와 관련된 보안 사고 식별 - 단일 창을 사용하여 조직 내 모든 AWS 계정의 모든 사용자를 조사합니다. IAM 사용자가 가장 최근에 액세스한 개별 AWS 리전 및 사용한 서비스의 추세를 추적할 수 있습니다.
- IAM 사용자의 SSO 인증으로의 실시간 마이그레이션 추적 - SSO를 사용하면 사용자는 단일 보안 인증 정보로 한 번 로그인하고 여러 AWS 계정 및 애플리케이션에 액세스할 수 있습니다. IAM 사용자를 SSO로 마이그레이션하려는 경우 이 패턴을 통해 SSO로 전환하고 모든 AWS 계정의 모든 IAM 사용자 보안 인증 정보 사용(예: AWS Management Console 액세스 또는 액세스 키 사용)을 추적할 수 있습니다.
- IAM 사용자가 액세스하는 AWS 리전 추적 - 데이터 주권 및 비용 관리와 같은 다양한 목적으로 지역에 대한 IAM 사용자 액세스를 제어할 수 있습니다. 또한 모든 IAM 사용자의 리전 사용을 추적할 수 있습니다.
- 규정 준수 유지 - 최소 권한 원칙에 따라 특정 작업을 수행하는 데 필요한 특정 IAM 권한만 부여할 수 있습니다. 또한, AWS 서비스, AWS Management Console, 장기 보안 인증 정보 사용에 대한 액세스를 추적할 수 있습니다.
- 다른 이해 관계자와 정보 공유 - IAM 보안 인증 보고서 또는 AWS 계정에 대한 액세스 권한을 부여하지 않고도 다른 이해 관계자와 선별된 대시보드를 공유할 수 있습니다.

패턴 더 보기

- [AWS Data Exchange에서 Amazon S3로 자동으로 데이터 수집](#)
- [Amazon Textract를 사용하여 PDF 파일에서 콘텐츠 자동 추출하기](#)
- [AWS DataOps 개발 키트를 사용하여 Google 애널리틱스 데이터를 수집, 변환 및 분석할 수 있는 데이터 파이프라인을 구축하십시오.](#)
- [Amazon Athena를 사용하여 공유 AWS Glue 데이터 카탈로그에 대한 크로스 계정 액세스 구성](#)
- [AWS IoT Greengrass를 사용하여 IoT 데이터를 Amazon S3에 직접 비용 효율적으로 수집할 수 있습니다](#)
- [AWS Cost Explorer를 사용하여 Amazon EMR 클러스터에 대한 자세한 비용 및 사용 보고서를 생성](#)
- [Amazon RDS 및 Amazon Aurora에 대한 자세한 비용 및 사용 보고서 생성](#)
- [AWS Cost Explorer를 사용하여 AWS Glue 작업에 대한 자세한 비용 및 사용 보고서를 생성합니다.](#)
- [계정 간 데이터 공유 자동화](#)
- [AWS 클라우드에서 인프라를 코드로 사용하여 서버리스 데이터 레이크 배포와 관리](#)
- [로컬 Angular QuickSight 애플리케이션에 Amazon 대시보드 내장](#)
- [Amazon Redshift 클러스터를 생성할 때 암호화되었는지 확인합니다.](#)
- [시작 시 Amazon EMR 저장 데이터에 대한 암호화가 활성화되었는지 확인](#)
- [데이터 레이크의 AWS IoT SiteWise 메타데이터 속성 추출 및 쿼리](#)
- [SageMaker 노트북 인스턴스에 다른 AWS 계정의 CodeCommit 리포지토리에 대한 임시 액세스 권한 부여](#)
- [Amazon Data Firehose 리소스가 AWS KMS 키로 암호화되지 않은 경우 식별 및 알림](#)
- [자체 호스팅된 MongoDB 환경을 AWS 클라우드의 MongoDB Atlas로 마이그레이션](#)
- [Oracle 플랫 파일 어댑터를 사용하여 GoldenGate Oracle 데이터베이스를 Amazon RDS for Oracle로 마이그레이션합니다.](#)
- [AWS DMS 및 AWS SCT를 사용하여 Amazon Redshift로 Oracle 데이터베이스 마이그레이션](#)
- [Amazon S3용 DistCp PrivateLink AWS를 사용하여 온 프레미스 하둡 환경에서 Amazon S3로 데이터를 마이그레이션합니다.](#)
- [카우치베이스 서버에서 AWS의 카우치베이스 카펠라로 마이그레이션](#)
- [온프레미스 Cloudera 워크로드를 AWS의 Cloudera 데이터 플랫폼으로 마이그레이션](#)
- [시작 시 전송 중 암호화가 있는지 Amazon EMR 클러스터 모니터링](#)
- [AWS용 Grafana 모니터링 대시보드 설정 ParallelCluster](#)
- [새 Amazon Redshift 클러스터에 필수 SSL 엔드포인트가 있는지 확인](#)

- [새 Amazon Redshift 클러스터가 VPC에서 시작되는지 확인](#)
- [Flask와 Elastic Beanstalk를 사용하여 AI/ML 모델 결과를 시각화합니다.](#)

비즈니스 생산성

주제

- [AWS에서 고가용성 PeopleSoft 아키텍처 설정](#)
- [더 많은 패턴](#)

AWS에서 고가용성 PeopleSoft 아키텍처 설정

작성자: 라마나단 무랄리다르 (AWS)

환경: 프로덕션

기술: 비즈니스 생산성; 인프라;
웹 및 모바일 앱, 데이터베이스

워크로드: Oracle

AWS 서비스: Amazon EC2
Auto Scaling, Amazon EFS,
Elastic Load Balancing(ELB),
Amazon RDS

요약

PeopleSoft 워크로드를 AWS로 마이그레이션할 때 복원력은 중요한 목표입니다. 이를 통해 PeopleSoft 애플리케이션의 가용성이 항상 높고 장애로부터 신속하게 복구할 수 있습니다.

이 패턴은 네트워크, PeopleSoft 애플리케이션 및 데이터베이스 계층에서 고가용성 (HA) 을 보장하는 AWS 기반 애플리케이션의 아키텍처를 제공합니다. 오라클의 경우 [Amazon Relational Database Service\(RDS\)](#)를 사용하고 데이터베이스 티어에는 Amazon RDS for SQL Server 데이터베이스를 사용합니다. 이 아키텍처에는 [Amazon Route 53](#), [Amazon Elastic Compute Cloud\(Amazon EC2\)](#) 리눅스 인스턴스, [Amazon 엘라스틱 블록 스토리지 \(Amazon EBS\)](#), [Amazon Elastic File System\(Amazon EFS\)](#), [Application Load Balancer](#)와 같은 AWS 서비스도 포함되며 확장이 가능합니다.

[PeopleSoft오라클](#)은 인력 관리 및 기타 비즈니스 운영을 위한 도구 및 애플리케이션 제품군을 제공합니다.

사전 조건 및 제한 사항

사전 조건

- 활성 상태의 AWS 계정
- AWS에서 설정하는 데 필요한 라이선스가 있는 PeopleSoft 환경
- 다음 리소스로 AWS 계정에 설정된 Virtual Private Cloud(VPC)
 - 최소 두 개의 가용 영역

- 각 가용 영역에 1개의 퍼블릭 서브넷과 3개의 프라이빗 서브넷
- NAT 게이트웨이와 인터넷 게이트웨이
- 트래픽을 라우팅하기 위한 각 서브넷의 라우팅 테이블
- 조직의 표준에 따라 PeopleSoft 애플리케이션의 보안을 보장하는 데 도움이 되도록 정의된 네트워크 액세스 제어 목록 (네트워크 ACL) 및 보안 그룹

제한 사항

- 이 패턴은 고가용성 (HA) 솔루션을 제공합니다. 재해 복구 (DR) 시나리오는 지원하지 않습니다. 드문 경우이긴 하지만 HA 구현을 위한 전체 AWS 리전이 다운되는 경우 애플리케이션을 사용할 수 없게 됩니다.

제품 버전

- PeopleSoft PeopleTools 8.52 이상을 실행하는 애플리케이션

아키텍처

대상 아키텍처

PeopleSoft 프로덕션 애플리케이션의 다운타임 또는 중단은 애플리케이션의 가용성에 영향을 미치고 비즈니스에 심각한 지장을 초래합니다.

항상 가용성이 높도록 PeopleSoft 프로덕션 애플리케이션을 설계하는 것이 좋습니다. 단일 장애 지점을 제거하고, 신뢰할 수 있는 크로스오버 또는 장애 조치를 추가하고, 장애를 감지함으로써 이를 달성할 수 있습니다. 다음 다이어그램은 AWS 기반 HA 아키텍처를 PeopleSoft 보여줍니다.

이 아키텍처 배포에서는 Amazon RDS for Oracle을 데이터베이스로 사용하고 Red Hat 엔터프라이즈 리눅스 (RHEL) 에서 실행되는 EC2 인스턴스를 사용합니다. PeopleSoft Amazon RDS for SQL Server를 피플소프트 데이터베이스로 사용할 수도 있습니다.

이 아키텍처에는 다음 구성 요소가 포함됩니다.

- [Amazon Route 53](#)은 인터넷에서 PeopleSoft 애플리케이션으로 요청을 라우팅하기 위한 DNS (도메인 이름 서버) 로 사용됩니다.

- [AWS WAF](#)를 사용하면 가용성에 영향을 미치거나, 보안을 손상시키거나, 리소스를 과도하게 소비할 수 있는 일반적인 웹 익스플로잇과 봇으로부터 보호할 수 있습니다. [AWS Shield Advanced](#)(그림 없음)는 훨씬 더 광범위한 보호를 제공합니다.
- [Application Load Balancer](#)는 웹 서버를 대상으로 하는 고급 요청 라우팅을 통해 HTTP 및 HTTPS 트래픽의 부하를 분산합니다.
- 애플리케이션을 지원하는 웹 서버, 애플리케이션 서버, 프로세스 스케줄러 서버, Elasticsearch 서버는 여러 가용 영역에서 실행되며 Amazon [EC2](#) Auto Scaling을 사용합니다. PeopleSoft
- PeopleSoft 애플리케이션에서 사용하는 데이터베이스는 [Amazon RDS에서](#) 다중 AZ 구성으로 실행됩니다.
- PeopleSoft 애플리케이션에서 사용하는 파일 공유는 [Amazon EFS에서](#) 구성되며 인스턴스 간에 파일에 액세스하는 데 사용됩니다.
- [Amazon EC2 Auto Scaling](#)에서는 필요할 때 PeopleSoft 구성 요소를 신속하게 복제할 수 있도록 [Amazon 머신 이미지 \(AMI\)](#) 를 사용합니다.
- [NAT 게이트웨이](#)는 프라이빗 서브넷의 인스턴스를 VPC 외부의 서비스에 연결하고 외부 서비스가 해당 인스턴스와 연결을 시작할 수 없도록 합니다.
- [인터넷 게이트웨이](#)는 수평적 확장, 이중화,고가용성의 VPC 구성 요소로, VPC와 인터넷 간의 통신을 가능하게 합니다.
- 퍼블릭 서브넷의 배스천 호스트는 인터넷 또는 온프레미스 네트워크와 같은 외부 네트워크에서 프라이빗 서브넷의 서버에 대한 액세스를 제공합니다. 배스천 호스트는 프라이빗 서브넷의 서버에 대한 제어되고 안전한 액세스를 제공합니다.

아키텍처 세부 정보

PeopleSoft 데이터베이스는 다중 AZ 구성의 오라클용 Amazon RDS (또는 SQL Server용 Amazon RDS) 데이터베이스에 보관됩니다. [Amazon RDS 다중 AZ 기능](#)은 데이터베이스 업데이트를 두 개의 가용 영역에 복제하여 내구성과 가용성을 높입니다. Amazon RDS는 계획된 유지 관리 및 예기치 않은 중단에 대비하여 대기 데이터베이스로 자동 페일오버합니다.

PeopleSoft 웹 및 미들 티어는 EC2 인스턴스에 설치됩니다. 이러한 인스턴스는 여러 가용 영역에 분산되어 있으며 [오토 스케일링](#)으로 묶여 있습니다. 이렇게 하면 이러한 구성 요소의 가용성이 항상 높아집니다. 애플리케이션을 항상 사용할 수 있고 필요할 때 확장할 수 있도록 필요한 최소 인스턴스 수를 유지합니다.

OEM EC2 인스턴스에는 최신 EC2 인스턴스 유형을 사용하는 것이 좋습니다. 현재 세대 인스턴스 유형(예: [AWS Nitro System에 구축된 인스턴스](#))은 하드웨어 가상 머신(HVM)을 지원합니다. [향상된 네트워킹](#)을 활용하려면 HVM AMI가 필요하며 향상된 보안도 제공합니다. 각 오토 스케일링에 속하는 EC2

인스턴스는 인스턴스를 교체하거나 확장할 때 자체 AMI를 사용합니다. 애플리케이션에서 처리하려는 부하와 PeopleSoft 애플리케이션 및 릴리스에 대해 오라클에서 권장하는 최소값을 기준으로 EC2 인스턴스 유형을 선택하는 것이 좋습니다. PeopleSoft PeopleTools 하드웨어 및 소프트웨어 요구 사항에 대한 자세한 내용은 [오라클 지원 웹 사이트](#)를 참조하십시오.

PeopleSoft 웹 및 미들 티어는 Amazon EFS 마운트를 공유하여 보고서, 데이터 파일 및 (필요한 경우) PS_HOME 디렉토리를 공유합니다. Amazon EFS는 성능 및 비용상의 이유로 각 가용 영역에 탑재 대상을 포함하도록 구성되어 있습니다.

Application Load Balancer는 PeopleSoft 애플리케이션에 액세스하는 트래픽을 지원하고 다양한 가용 영역에 있는 웹 서버 간의 트래픽 부하를 분산하도록 프로비저닝됩니다. Application Load Balancer는 최소 2개의 가용 영역에서 HA를 제공하는 네트워크 장치입니다. 웹 서버는 부하 분산 구성을 사용하여 서로 다른 애플리케이션 서버로 트래픽을 분산합니다. 웹 서버와 애플리케이션 서버 간의 부하 분산을 통해 인스턴스 전체에 부하가 고르게 분산되고 인스턴스 과부하로 인한 병목 현상과 서비스 중단을 방지할 수 있습니다.

Amazon Route 53은 인터넷에서 애플리케이션 로드 밸런서로 트래픽을 라우팅하는 DNS 서비스로 사용됩니다. Route 53은 가용성과 확장성이 뛰어난 DNS 웹 서비스입니다.

HA 세부 정보

- **데이터베이스:** Amazon RDS의 다중 AZ 기능은 동기 복제를 통해 여러 가용 영역에 있는 두 데이터베이스를 운영합니다. 이렇게 하면 자동 장애 조치가 가능한 가용성이 높은 환경이 만들어집니다. Amazon RDS는 장애 조치 이벤트 감지 기능을 갖추고 있으며 이러한 이벤트가 발생하면 자동 장애 조치를 시작합니다. Amazon RDS API를 통해 수동 장애 조치를 시작할 수도 있습니다. 자세한 설명은 블로그 게시물 [Amazon RDS 언더더 후드: 다중 AZ](#)를 참조하십시오. 장애 조치가 원활하며 장애 조치 발생 시 애플리케이션이 자동으로 데이터베이스에 다시 연결됩니다. 하지만 장애 조치 중에 발생하는 모든 프로세스 스케줄러 작업은 오류를 생성하므로 다시 제출해야 합니다.
- **PeopleSoft 애플리케이션 서버:** 애플리케이션 서버는 여러 가용 영역에 분산되어 있으며 이를 위해 Auto Scaling 그룹이 정의되어 있습니다. 인스턴스에 장애가 발생하면 오토 스케일링은 즉시 해당 인스턴스를 애플리케이션 서버 템플릿의 AMI에서 복제된 정상 인스턴스로 교체합니다. 특히 충격 폴링이 활성화되어 애플리케이션 서버 인스턴스가 중단되면 세션이 자동으로 다른 애플리케이션 서버로 장애 조치되고 오토 스케일링이 자동으로 다른 인스턴스를 가동하고 애플리케이션 서버를 가동하여 Amazon EFS 마운트에 등록합니다. 새로 생성된 애플리케이션 서버는 웹 서버의 PSSTRSETUP.SH 스크립트를 사용하여 웹 서버에 자동으로 추가됩니다. 이렇게 하면 애플리케이션 서버의 가용성이 항상 높고 장애 발생 시 신속하게 복구할 수 있습니다.
- **프로세스 스케줄러:** 프로세스 스케줄러 서버는 여러 가용 영역에 분산되어 있으며 이를 위해 오토 스케일링이 정의되어 있습니다. 인스턴스에 장애가 발생하면 오토 스케일링은 즉시 프로세스 스케줄

러 서버 템플릿의 AMI에서 복제된 정상 인스턴스로 교체합니다. 특히, 프로세스 스케줄러 인스턴스가 다운되면 오토 스케일링은 자동으로 다른 인스턴스를 가동시키고 프로세스 스케줄러를 불러옵니다. 인스턴스에 장애가 발생했을 때 실행 중이었던 모든 작업을 다시 제출해야 합니다. 이렇게 하면 프로세스 스케줄러가 항상고가용성을 유지하고 장애로부터 빠르게 복구할 수 있습니다.

- **엘라스틱서치 서버:** Elasticsearch 서버에는 이들을 위한 오토 스케일링이 정의되어 있습니다. 인스턴스에 장애가 발생하면 오토 스케일링은 해당 인스턴스를 Elasticsearch 서버 템플릿의 AMI에서 복제된 정상 인스턴스로 즉시 교체합니다. 특히, Elasticsearch 인스턴스가 다운되면 요청을 처리하는 Application Load Balancer가 장애를 감지하고 해당 인스턴스로의 트래픽 전송을 중단합니다. 오토 스케일링은 자동으로 다른 인스턴스를 가동시키고 Elasticsearch 인스턴스를 불러옵니다. Elasticsearch 인스턴스가 백업되면 Application Load Balancer는 정상 상태를 감지하고 다시 요청을 보내기 시작합니다. 이렇게 하면 Elasticsearch 서버가 항상고가용성을 유지하고 장애로부터 빠르게 복구할 수 있습니다.
- **웹 서버:** 웹 서버에는 오토 스케일링이 정의되어 있습니다. 인스턴스에 장애가 발생하면 오토 스케일링은 해당 인스턴스를 웹 서버 템플릿의 AMI에서 복제된 정상 인스턴스로 즉시 교체합니다. 특히 웹 서버 인스턴스가 다운되면 요청을 처리하는 Application Load Balancer가 장애를 감지하고 해당 인스턴스에 대한 트래픽 전송을 중단합니다. 오토 스케일링은 자동으로 다른 인스턴스를 가동시키고 웹 서버 인스턴스를 불러옵니다. 웹 서버 인스턴스가 백업되면 Application Load Balancer는 정상 상태를 감지하고 요청을 다시 보내기 시작합니다. 이렇게 하면 웹 서버의 가용성이 항상 높아지고 장애로부터 신속하게 복구할 수 있습니다.

도구

서비스

- [Application Load Balancer](#)는 들어오는 애플리케이션 트래픽을 여러 가용 영역의 EC2 인스턴스 등 여러 대상에 분산합니다.
- [Amazon Elastic Block Store\(Amazon EBS\)](#)는 Amazon Elastic Compute Cloud(Amazon EC2) 인스턴스와 함께 사용할 수 있는 블록 스토리지 볼륨을 제공합니다.
- [Amazon Elastic Compute Cloud\(Amazon EC2\)](#)는 AWS 클라우드에서 확장 가능한 컴퓨팅 용량을 제공합니다. 필요한 만큼 가상 서버를 시작하고 빠르게 스케일 업하거나 스케일 다운할 수 있습니다.
- [Amazon Elastic File System\(Amazon EFS\)](#)은 AWS Cloud에서 공유 파일 시스템을 생성하고 구성하는 데 도움이 됩니다.
- [Amazon Relational Database Service\(RDS\)](#)는 AWS 클라우드에서 관계형 데이터베이스를 설정, 운영 및 조정하는 데 도움이 됩니다.
- [Amazon Route 53](#)는 가용성과 확장성이 뛰어난 DNS 웹 서비스입니다.

모범 사례

운영 모범 사례

- PeopleSoft AWS에서 실행하는 경우 Route 53을 사용하여 인터넷 및 로컬에서 트래픽을 라우팅합니다. 기본 DB 인스턴스를 사용할 수 없는 경우 [장애 조치 옵션](#)을 사용하여 트래픽을 재해 복구 (DR) 사이트로 다시 라우팅합니다.
- 항상 환경 앞에서 Application Load Balancer를 PeopleSoft 사용하십시오. 이렇게 하면 트래픽이 웹 서버로 안전하게 로드 밸런싱됩니다.
- Application Load Balancer 대상 그룹 설정에서 로드 밸런서가 생성한 [쿠키로 고정성이 켜져](#) 있는지 확인합니다.

참고: 외부 Single Sign-On(SSO)를 사용하는 경우 애플리케이션 기반 쿠키를 사용해야 할 수 있습니다. 이렇게 하면 웹 서버와 애플리케이션 서버 간에 연결이 일관되게 유지됩니다.

- PeopleSoft 프로덕션 애플리케이션의 경우 Application Load Balancer 유휴 제한 시간은 사용하는 웹 프로필에 설정된 것과 일치해야 합니다. 이렇게 하면 로드 밸런서 계층에서 사용자 세션이 만료되는 것을 방지할 수 있습니다.
- PeopleSoft 프로덕션 애플리케이션의 경우 애플리케이션 서버 [재활용 횟수를 메모리 누수를 최소화하는](#) 값으로 설정하십시오.
- 이 패턴에 설명된 대로 PeopleSoft 프로덕션 애플리케이션에 Amazon RDS 데이터베이스를 사용하는 경우 [고가용성을 위해 다중 AZ 형식으로](#) 실행하십시오.
- 데이터베이스가 PeopleSoft 프로덕션 애플리케이션용 EC2 인스턴스에서 실행되는 경우 [고가용성을 위해 대기 데이터베이스가 다른 가용 영역에서 실행되고](#) 있는지 확인하십시오.
- DR의 경우 Amazon RDS 데이터베이스 또는 EC2 인스턴스가 프로덕션 데이터베이스와는 별도의 AWS 리전에 구성된 예비 복제본이 있는지 확인하십시오. 이렇게 하면 해당 지역에 재해가 발생할 경우 애플리케이션을 다른 지역으로 전환할 수 있습니다.
- DR의 경우 [Amazon Elastic 재해 복구](#)를 사용하여 프로덕션 구성 요소와는 별도의 지역에 애플리케이션 수준 구성 요소를 설정할 수 있습니다. 이렇게 하면 해당 지역에 재해가 발생하는 경우 애플리케이션을 다른 지역으로 전환할 수 있습니다.
- Amazon EFS (중간 I/O 요구 사항의 경우) 또는 [Amazon FSx](#) (높은 I/O 요구 사항의 경우) 를 사용하여 보고서, 첨부 파일 및 데이터 파일을 PeopleSoft 저장합니다. 이를 통해 콘텐츠가 한 곳에 중앙 위치에 저장되고 인프라 내 어느 곳에서나 액세스할 수 있습니다.
- [Amazon CloudWatch](#) (기본 및 세부 정보) 을 사용하여 PeopleSoft 애플리케이션이 사용하는 AWS 클라우드 리소스를 거의 실시간으로 모니터링할 수 있습니다. 이렇게 하면 문제에 대해 즉시 알림을 받고 환경 가용성에 영향을 미치기 전에 문제를 신속하게 해결할 수 있습니다.

- Amazon RDS 데이터베이스를 데이터베이스로 사용하는 경우 [향상된 모니터링을 PeopleSoft](#) 사용하십시오. 이 기능을 사용하면 CPU, 메모리, 파일 시스템 I/O, 디스크 I/O를 비롯한 50개 이상의 지표에 액세스할 수 있습니다.
- [CloudTrailAWS](#)를 사용하면 PeopleSoft 애플리케이션이 사용하는 AWS 리소스의 API 호출을 모니터링할 수 있습니다. 이를 통해 보안 분석, 리소스 변경 추적 및 규정 준수 감사를 수행할 수 있습니다.

보안 모범 사례

- [SQL 인젝션 또는 크로스 사이트 스크립팅 \(XSS\) 과 같은 일반적인 악용으로부터 PeopleSoft 애플리케이션을 보호하려면 AWS WAF를 사용하십시오.](#) 맞춤형 탐지 및 완화 서비스를 위해 [AWS Shield Advanced](#)를 사용해 보십시오.
- 애플리케이션 보안을 위해 트래픽을 HTTP에서 HTTPS로 자동 리디렉션하는 규칙을 Application Load Balancer에 추가합니다. PeopleSoft
- Application Load Balancer에 대해 별도의 보안 그룹을 설정합니다. 이 보안 그룹은 HTTPS/HTTP 인바운드 트래픽만 허용하고 아웃바운드 트래픽은 허용하지 않아야 합니다. 이렇게 하면 의도된 트래픽만 허용되고 애플리케이션을 보호하는 데 도움이 됩니다.
- 애플리케이션 서버, 웹 서버 및 데이터베이스에는 프라이빗 서브넷을 사용하고 아웃바운드 인터넷 트래픽에는 [NAT 게이트웨이](#)를 사용합니다. 이렇게 하면 애플리케이션을 지원하는 서버에 공개적으로 접속할 수 없고 필요한 서버에만 퍼블릭 액세스를 제공할 수 있습니다.
- 서로 다른 VPC를 사용하여 프로덕션 환경과 PeopleSoft 비프로덕션 환경을 실행하십시오. [AWS Transit Gateway](#), [VPC 피어링](#), [네트워크 ACL](#) 및 [보안 그룹](#)을 사용하여 [VPC와](#) 필요한 경우 온프레미스 데이터 센터 간의 트래픽 흐름을 제어할 수 있습니다.
- 최소 권한 원칙을 따르세요. PeopleSoft 애플리케이션에서 사용하는 AWS 리소스에 대한 액세스 권한을 꼭 필요한 사용자에게만 부여하십시오. 작업을 수행하는 데 필요한 최소 권한만 부여합니다. 자세한 내용은 AWS Well-Architected Framework의 [보안 필터](#)를 참조하세요.
- 가능하면 [AWS Systems Manager](#)를 사용하여 PeopleSoft 애플리케이션이 사용하는 EC2 인스턴스에 액세스하십시오.

안정성 모범 사례

- Application Load Balancer를 사용하는 경우 활성화된 각 가용 영역에 대해 단일 대상을 등록하십시오. 따라서 로드 밸런서가 가장 효과적입니다.
- 각 PeopleSoft 프로덕션 환경에는 애플리케이션 액세스를 위한 URL 1개, 통합 브로커를 위한 URL 1개, 보고서 보기를 위한 URL 1개 등 3개의 고유한 URL을 사용하는 것이 좋습니다. 가능하면 각 URL

에는 자체 전용 웹 서버와 애플리케이션 서버가 있어야 합니다. 이 디자인은 각 URL마다 고유한 기능과 통제된 액세스를 제공하므로 PeopleSoft 애플리케이션의 보안을 강화하는 데 도움이 됩니다. 또한 기본 서비스에 장애가 발생할 경우 영향을 받을 수 있는 범위를 최소화합니다.

- PeopleSoft 애플리케이션의 [로드 밸런서 대상 그룹에 상태 확인](#)을 구성하는 것이 좋습니다. 상태 확인은 해당 서버를 실행하는 EC2 인스턴스 대신 웹 서버에서 수행해야 합니다. 이렇게 하면 웹 서버가 충돌하거나 웹 서버를 호스팅하는 EC2 인스턴스가 다운되는 경우 Application Load Balancer가 해당 정보를 정확하게 반영할 수 있습니다.
- PeopleSoft 프로덕션 애플리케이션의 경우 웹 서버를 3개 이상의 가용 영역에 분산하는 것이 좋습니다. 이렇게 하면 가용 영역 중 하나에 장애가 발생하더라도 PeopleSoft 애플리케이션의 가용성이 항상 높아집니다.
- PeopleSoft 프로덕션 애플리케이션의 경우 jolt pooling () 을 활성화하십시오. joltPooling=true 이렇게 하면 패치 적용 목적 또는 VM 장애로 인해 서버가 다운되는 경우 애플리케이션이 다른 애플리케이션 서버로 페일오버될 수 있습니다.
- PeopleSoft 프로덕션 애플리케이션의 경우 1로 설정합니다DynamicConfigReload . 이 설정은 PeopleTools 버전 8.52 이상에서 지원됩니다. 서버를 다시 시작하지 않고도 웹 서버에 새 애플리케이션 서버를 동적으로 추가합니다.
- PeopleTools 패치를 적용할 때 다운타임을 최소화하려면 웹 및 애플리케이션 서버의 Auto Scaling 그룹 시작 구성에 블루/그린 배포 방법을 사용하십시오. 자세한 내용은 [AWS 기반 배포 옵션 개요](#) 백서를 참조하세요.
- [AWS Backup을 사용하여 AWS에서 PeopleSoft 애플리케이션을 백업하십시오](#). AWS Backup은 대규모 데이터 보호를 간소화하는 비용 효율적인 완전관리형 정책 기반 서비스입니다.

성능 모범 사례

- 비즈니스에 환경 전체에 암호화된 트래픽이 필요한 경우가 아니라면 Application Load Balancer에서 SSL을 종료하여 PeopleSoft 환경 성능을 최적화하십시오.
- Amazon [Simple Notification Service \(Amazon SNS\)](#) 와 같은 AWS 서비스를 위한 [인터페이스 VPC 엔드포인트를 생성하여 트래픽이 항상 내부적으로 CloudWatch](#) 유지되도록 합니다. 이는 비용 효율적이며 애플리케이션을 안전하게 유지하는 데 도움이 됩니다.

비용 최적화 모범 사례

- PeopleSoft 환경에서 사용하는 모든 리소스에 태그를 지정하고 [비용 할당](#) 태그를 활성화하십시오. 이러한 태그를 통해 리소스 비용을 확인하고 관리할 수 있습니다.

- PeopleSoft 프로덕션 애플리케이션의 경우 웹 서버 및 애플리케이션 서버에 대한 Auto Scaling 그룹을 설정합니다. 이렇게 하면 애플리케이션을 지원하는 웹 및 애플리케이션 서버 수를 최소한으로 유지할 수 있습니다. [오토 스케일링 정책](#)을 사용하여 필요에 따라 서버를 확장하거나 축소할 수 있습니다.
- 비용이 지정한 예산 임계값을 초과할 경우 [청구 경고](#)를 사용하여 알림을 받을 수 있습니다.

지속가능성 모범 사례

- IaC ([코드형 인프라](#)) 를 사용하여 PeopleSoft 환경을 유지 관리하십시오. 이를 통해 일관된 환경을 구축하고 변경 관리를 유지할 수 있습니다.

에픽

PeopleSoft 데이터베이스를 Amazon RDS로 마이그레이션

작업	설명	필요한 기술
DB 서브넷 그룹을 생성합니다.	Amazon RDS 콘솔 의 탐색 창에서 [서브넷 그룹] 을 선택한 다음 여러 가용 영역에 서브넷이 있는 Amazon RDS DB 서브넷 그룹을 생성합니다. 이는 Amazon RDS 데이터베이스를 다중 AZ 구성으로 실행하는 데 필요합니다.	클라우드 관리자
Amazon RDS 데이터베이스를 생성합니다.	PeopleSoft HA 환경을 위해 선택한 AWS 지역의 가용 영역에 Amazon RDS 데이터베이스를 생성합니다. Amazon RDS 데이터베이스를 생성할 때는 다중 AZ 옵션 (대기 인스턴스 생성) 과 이전 단계에서 생성한 데이터베이스 서브넷 그룹을 선택해야 합니다. 자세한 내용	클라우드 관리자, Oracle 데이터베이스 관리자

작업	설명	필요한 기술
	은 Amazon RDS 설명서 를 참조하세요.	
PeopleSoft 데이터베이스를 Amazon RDS로 마이그레이션 하십시오.	AWS PeopleSoft 데이터베이스 마이그레이션 서비스 (AWS DMS) 를 사용하여 기존 데이터베이스를 Amazon RDS 데이터베이스로 마이그레이션합니다. 자세한 내용은 AWS DMS 설명서 및 AWS 블로그 게시물 AWS DMS를 사용하여 다운타임이 거의 없는 오라클 데이터베이스 마이그레이션 을 참조하세요.	클라우드 관리자, DBA PeopleSoft

Amazon EFS 파일 시스템 설정

작업	설명	필요한 기술
파일 시스템을 만듭니다.	Amazon EFS 콘솔 에서 각 가용 영역에 대한 파일 시스템과 탑재 대상을 생성합니다. 자세한 지침은 Amazon EFS 설명서 를 참조하십시오. 파일 시스템이 생성되면 해당 DNS 이름을 기록해 둡니다. 파일 시스템을 탑재할 때 이 정보를 사용합니다.	클라우드 관리자

PeopleSoft 애플리케이션 및 파일 시스템 설정

작업	설명	필요한 기술
EC2 인스턴스를 시작합니다.	<p>PeopleSoft 애플리케이션용 EC2 인스턴스를 시작합니다. 자세한 지침은 Amazon EC2 설명서를 참조하십시오.</p> <ul style="list-style-type: none"> 이름에 APP_TEMPLATE 를 입력합니다. OS 이미지의 경우 Red Hat을 선택하십시오. 인스턴스 유형에서 PeopleSoft 애플리케이션에 적합한 인스턴스 유형을 선택합니다. 자세한 내용은 아키텍처 섹션의 아키텍처 세부 정보를 참조하십시오. 	클라우드 관리자, PeopleSoft 관리자
인스턴스에 PeopleSoft 설치합니다.	<p>PeopleSoft 애플리케이션을 설치하고 생성한 EC2 인스턴스에 PeopleTools 설치합니다. 지침은 Oracle 설명서를 참조하십시오.</p>	클라우드 관리자, PeopleSoft 관리자
애플리케이션 서버를 생성합니다.	<p>AMI 템플릿용 애플리케이션 서버를 생성하고 Amazon RDS 데이터베이스에 제대로 연결되었는지 확인합니다.</p>	클라우드 관리자, PeopleSoft 관리자
Amazon EFS 파일 시스템을 마운트합니다.	<p>루트 사용자로 EC2 인스턴스에 로그인하고 다음 명령을 실행하여 Amazon EFS 파일 시스템을 PSFTMNT 서버의 호출 폴더에 마운트합니다.</p> <pre>sudo su -</pre>	클라우드 관리자, PeopleSoft 관리자

작업	설명	필요한 기술
	<pre>mkdir /psftmnt cat /etc/fstab</pre> <p>/etc/fstab 파일에 다음 줄을 추가합니다. 파일 시스템을 생성할 때 기록해 둔 DNS 이름을 사용하십시오.</p> <pre>fs-09e064308f11453 88.efs.us-east-1.amazonaws.com:/ / psftmnt nfs4 nfsvers=4 .1,rsize=1048576,write=1048576,hard, timeo=600,retrans= 2,noresvport,_netdev 0 0 mount -a</pre>	
권한을 확인합니다.	PeopleSoft 사용자가 PSFTMNT 폴더에 제대로 액세스할 수 있도록 폴더에 적절한 권한이 있는지 확인하세요.	클라우드 관리자, PeopleSoft 관리자
인스턴스를 더 생성합니다.	이 에픽의 이전 단계를 반복하여 프로세스 스케줄러, 웹 서버 및 Elasticsearch 서버용 템플릿 인스턴스를 생성합니다. 이들 인스턴스의 이름을 PRCS_TEMPLATE , WEB_TEMPLATE , SRCH_TEMPLATE (으)로 지정하세요. 웹 서버의 경우 joltPooling=true 및 DynamicConfigReload=1 을(를) 설정합니다.	클라우드 관리자, PeopleSoft 관리자

서버를 설정하기 위한 스크립트 생성

작업	설명	필요한 기술
<p>애플리케이션 서버를 설치하는 스크립트를 생성합니다.</p>	<p>Amazon EC2 APP_TEMPLATE 인스턴스에서 PeopleSoft 사용자로서 다음 스크립트를 생성합니다. 이름을 <code>appstart.sh</code> 지정하고 디렉터리에 배치합니다.</p> <p>PS_HOME 이 스크립트를 사용하여 애플리케이션 서버를 불러오고 Amazon EFS 마운트에 서버 이름을 기록합니다.</p> <pre data-bbox="592 835 1029 1234">#!/bin/ksh . /usr/homes/hcmdemo/.profile psadmin -c configure -d HCMDEMO psadmin -c parallelboot -d HCMDEMO touch /psftmnt/`echo \$HOSTNAME`</pre>	<p>PeopleSoft 관리자</p>
<p>스크립트를 생성하여 프로세스 스케줄러 서버를 설치합니다.</p>	<p>Amazon EC2 PRCS_TEMPLATE 인스턴스에서 PeopleSoft 사용자로서 다음 스크립트를 생성합니다. 이름을 <code>prcsstart.sh</code> 지정하고 디렉터리에 배치합니다.</p> <p>PS_HOME 이 스크립트를 사용하여 프로세스 스케줄러 서버를 불러올 수 있습니다.</p> <pre data-bbox="592 1728 1029 1854">#!/bin/ksh . /usr/homes/hcmdemo/.profile</pre>	<p>PeopleSoft 관리자</p>

작업	설명	필요한 기술
	<pre>/* The following line ensures that the process scheduler always has a unique name during replaceme nt or scaling activity. */ sed -i "s/*PrCs ServerName.*`\host name -I awk -F. '{print "PrCsServ erName=PSUNX"\$3\$4} `/" \$HOME/appserv/ prcs*/psprcs.cfg psadmin -p configure -d HCMDEMO psadmin -p start -d HCMDEMO</pre>	

작업	설명	필요한 기술
<p>스크립트를 생성하여 Elasticsearch 서버를 설치하세요.</p>	<p>Amazon EC2 SRCH_TEMP LATE 인스턴스에서 Elasticsearch 사용자로서 다음 스크립트를 생성합니다. 이름을 srchstart.sh 지정하고 디렉터리에 배치합니다. HOME</p> <pre data-bbox="597 537 1029 1136">#!/bin/ksh /* The following line ensures that the correct IP is indicated in the elasticsearch arch.yaml file. */ sed -i "s/. *network.host.*`hostname -I awk '{print \"host:\"\$0}'`/" \$ES_HOME_DIR/config/elasticsearch.yaml nohup \$ES_HOME_DIR/bin/elasticsearch &</pre>	<p>PeopleSoft 관리자</p>

작업	설명	필요한 기술
<p>웹 서버를 설치하는 스크립트를 생성합니다.</p>	<p>Amazon EC2 WEB_TEMPLATE 인스턴스에서 웹 서버 사용자로서 HOME 디렉터리에 다음 스크립트를 생성합니다.</p> <p>renip.sh: 이 스크립트는 AMI에서 복제할 때 웹 서버가 올바른 IP를 갖도록 합니다.</p> <pre data-bbox="597 619 1027 1371">#!/bin/ksh hn=`hostname` /* On the following line, change the IP with the hostname with the hostname of the web template. */ for text_file in `find * -type f -exec grep -l '<hostname-of-the- web-template>' {} \;` do sed -e 's/<hostname-of-the-web-template>/'\$hn'/g' \$text_file > temp mv -f temp \$text_file done</pre> <p>psstrsetup.sh : 이 스크립트는 웹 서버가 현재 실행 중인 올바른 애플리케이션 서버 IP를 사용하도록 합니다. 충격 포트의 각 애플리케이션 서버에 연결을 시도하여 구성 파일에 추가합니다.</p> <pre data-bbox="597 1770 1027 1858">#!/bin/ksh c2=""</pre>	<p>PeopleSoft 관리자</p>

작업	설명	필요한 기술
	<pre> for ctr in `ls -1 / psftmnt/*.internal` do c1=`echo \$ctr awk -F "/" '{print \$3}'` /* In the following lines, 9000 is the jolt port. Change it if necessary. */ if nc -z \$c1 9000 2> / dev/null; then if [[\$c2 = ""]]; then c2="psserver="`echo \$c1`:9000" else c2=`echo \$c2`,`echo \$c1`:9000" fi fi done </pre> <p>webstart.sh : 이 스크립트는 두 개의 이전 스크립트를 실행하고 웹 서버를 시작합니다.</p> <pre> #!/bin/ksh /* Change the path in the following if necessary. */ cd /usr/homes/hcmdemo ./renip.sh ./psstrsetup.sh webserv/peoplesoft/ bin/startPIA.sh </pre>	

작업	설명	필요한 기술
<p>크론탭 항목을 추가하세요.</p>	<p>Amazon EC2 WEB_TEMPLATE 인스턴스에서 웹 서버 사용자로서 crontab에 다음 줄을 추가합니다. 필요한 값을 반영하도록 시간과 경로를 변경하십시오. 이 항목을 사용하면 웹 서버의 configuration.properties 파일에 항상 올바른 애플리케이션 서버 항목이 포함될 수 있습니다.</p> <div style="border: 1px solid #ccc; border-radius: 10px; padding: 10px; margin-top: 10px;"> <pre>* * * * * /usr/homes/hcmdemo/psstrsetup.sh</pre> </div>	<p>PeopleSoft 관리자</p>

AMI 및 오토 스케일링 템플릿 생성

작업	설명	필요한 기술
<p>애플리케이션 서버 템플릿용 AMI를 생성합니다.</p>	<p>Amazon EC2 콘솔에서 Amazon APP_TEMPLATE EC2 인스턴스의 AMI 이미지를 생성합니다. AMI의 이름을 PSAPPSRV-SCG-VER1 (으)로 지정합니다. 자세한 지침은 Amazon EC2 설명서를 참조하십시오.</p>	<p>클라우드 관리자, PeopleSoft 관리자</p>
<p>다른 서버용 AMI를 생성합니다.</p>	<p>이전 단계를 반복하여 프로세스 스케줄러, Elasticsearch 서버 및 웹 서버를 위한 AMI를 생성합니다.</p>	<p>클라우드 관리자, PeopleSoft 관리자</p>

작업	설명	필요한 기술
<p>애플리케이션 서버 오토 스케일링 그룹에 대한 시작 템플릿을 만듭니다.</p>	<p>애플리케이션 서버 오토 스케일링 그룹에 대한 시작 템플릿을 만듭니다. 템플릿 이름을 지정합니다. PSAPPSRV_TEMPLATE. 템플릿에서 APP_TEMPLATE 인스턴스로 생성한 AMI를 선택합니다. 자세한 지침은 Amazon EC2 설명서를 참조하십시오.</p> <ul style="list-style-type: none"> 시작 템플릿에서 요구 사항에 따라 인스턴스 유형을 선택합니다. 고급 세부 정보 섹션의 사용자 데이터 필드에 다음 항목을 추가합니다. 경로와 사용자 정보가 정확한지 확인하세요. 이전 단계에서 appstart.sh 스크립트를 만들었습니다. <pre data-bbox="625 1201 1029 1402"> #! /bin/ksh su -c "/usr/homes/hcmdemo/appstart.sh" - hcmdemo </pre>	<p>클라우드 관리자, PeopleSoft 관리자</p>

작업	설명	필요한 기술
<p>프로세스 스케줄러 서버 오토 스케일링에 대한 시작 템플릿을 생성합니다.</p>	<p>이전 단계를 반복하여 프로세스 스케줄러 서버 오토 스케일링에 대한 시작 템플릿을 생성합니다. 템플릿의 이름을 PSPRCS_TEMPLATE (으)로 지정합니다. 템플릿에서 프로세스 스케줄러용으로 생성한 AMI를 선택합니다.</p> <ul style="list-style-type: none"> 고급 세부 정보 섹션의 사용자 데이터 필드에 다음 항목을 추가합니다. 경로와 사용자 정보가 정확한지 확인하세요. 이전 단계에서 prcsstart.sh 스크립트를 만들었습니다. <pre data-bbox="626 999 1029 1199"> #! /bin/ksh su -c "/usr/homes/hcmdemo/prcsstart.sh" - hcmdemo </pre>	<p>클라우드 관리자, PeopleSoft 관리자</p>

작업	설명	필요한 기술
<p>Elasticsearch 서버 오토 스케일링을 위한 시작 템플릿을 생성합니다.</p>	<p>이전 단계를 반복하여 Elasticsearch 서버 오토 스케일링에 대한 시작 템플릿을 생성합니다. 템플릿의 이름을 SRCH_TEMPLATE (으)로 지정합니다. 템플릿에서 검색 서버 용으로 생성한 AMI를 선택합니다.</p> <ul style="list-style-type: none"> 고급 세부 정보 섹션의 사용자 데이터 필드에 다음 항목을 추가합니다. 경로와 사용자 정보가 정확한지 확인하세요. 이전 단계에서 srchstart.sh 스크립트를 만들었습니다. <pre data-bbox="625 999 1029 1199"> #! /bin/ksh su -c "/usr/homes/es/essearch/srchstart.sh" - essearch </pre>	<p>클라우드 관리자, PeopleSoft 관리자</p>

작업	설명	필요한 기술
웹 서버 오토 스케일링에 대한 시작 템플릿을 만듭니다.	<p>이전 단계를 반복하여 웹 서버 오토 스케일링에 대한 시작 템플릿을 생성합니다. 템플릿의 이름을 WEB_TEMPLATE (으) 로 지정합니다. 템플릿에서 웹 서버용으로 생성한 AMI의 유형을 선택합니다.</p> <ul style="list-style-type: none"> 고급 세부 정보 섹션의 사용자 데이터 필드에 다음 항목을 추가합니다. 경로와 사용자 정보가 정확한지 확인하세요. 이전 단계에서 webstart.sh 스크립트를 만들었습니다. <pre>#!/bin/ksh su -c "/usr/homes/hcmdemo/webstart.sh" - hcmdemo</pre>	클라우드 관리자, PeopleSoft 관리자

오토 스케일링 생성

작업	설명	필요한 기술
Application 서버를 위한 오토 스케일링을 생성합니다.	Amazon EC2 콘솔에서 PSAPPSRV_TEMPLATE 템플릿을 사용하여 애플리케이션 서버용으로 호출되는 PSAPPSRV_ASG 오토 스케일링을 생성합니다. 자세한 지침은 Amazon EC2 설명서 를 참조하십시오.	클라우드 관리자, PeopleSoft 관리자

작업	설명	필요한 기술
	<ul style="list-style-type: none"> 인스턴스 시작 옵션 선택 페이지에서 올바른 VPC를 선택한 다음 여러 가용 영역에서 여러 서브넷을 선택합니다. 고급 옵션 구성 페이지에서 로드 밸런서를 선택하지 마세요. 그룹 크기 및 조정 정책 구성 페이지에서 시스템을 설계하려는 부하의 양과 조정 정책을 사용할지 여부에 따라 설정을 선택합니다. 원하는 용량과 최소 용량을 최소 2로 설정하여 언제든지 트래픽을 처리하는 데 사용할 수 있는 인스턴스가 하나 이상 있도록 하는 것이 좋습니다. Auto Scaling 정책에 대한 자세한 내용은 Amazon EC2 설명서를 참조하세요. 	
<p>다른 서버에 대한 오토 스케일링을 생성합니다.</p>	<p>이전 단계를 반복하여 프로세스 스케줄러, Elasticsearch 서버 및 웹 서버를 위한 오토 스케일링을 생성합니다.</p>	<p>클라우드 관리자, PeopleSoft 관리자</p>

대상 그룹 생성 및 구성

작업	설명	필요한 기술
<p>웹 서버의 대상 그룹 생성</p>	<p>Amazon EC2 콘솔에서 웹 서버의 대상 그룹을 생성합니다. 자세한 내용은 탄력성 로드 밸</p>	<p>클라우드 관리자</p>

작업	설명	필요한 기술
	<p>런싱 설명서를 참조하세요. 포트를 웹 서버가 수신하는 포트 로 설정합니다.</p>	
<p>상태 확인을 구성합니다.</p>	<p>상태 확인에 비즈니스 요구 사항을 반영하는 올바른 값이 있는지 확인하세요. 자세한 내용은 탄력성 로드 밸런싱 설명서를 참조하세요.</p>	<p>클라우드 관리자</p>
<p>Elasticsearch 서버를 위한 대상 그룹을 생성합니다.</p>	<p>이전 단계를 반복하여 Elasticsearch 서버를 PSFTSRCH 호출하는 대상 그룹을 생성하고 올바른 Elasticsearch 포트를 설정합니다.</p>	<p>클라우드 관리자</p>
<p>오토 스케일링에 대상 그룹을 추가합니다.</p>	<p>앞서 생성한 웹 서버 PSPIA_ASG 오토 스케일링을 엽니다. 로드 밸런싱 탭에서 편집을 선택한 다음 오토 스케일링에 PSFTWEB 대상 그룹을 추가합니다.</p> <p>Elasticsearch 오토 스케일링에 대해 이 단계를 PSSRCH_ASG 반복하여 이전에 PSFTSRCH 생성한 대상 그룹을 추가합니다.</p>	<p>클라우드 관리자</p>

작업	설명	필요한 기술
세션 고정성을 설정합니다.	<p>대상 그룹 PSFTWEB에서 속성 탭을 선택하고 편집 을 선택한 다음 세션 고정도를 설정합니다. 고정성 유형의 경우 로드 밸런서 생성 쿠키를 선택하고 기간을 1로 설정합니다. 자세한 내용은 탄력성 로드 밸런싱 설명서를 참조하세요.</p> <p>대상 그룹 PSFTSRCH에 대해 단계를 반복합니다.</p>	클라우드 관리자

Application Load Balancer 생성 및 구성

작업	설명	필요한 기술
웹 서버용 로드 밸런서를 만듭니다.	<p>웹 서버에 대한 트래픽을 로드 PSFTLB 밸런싱하기 위해 이름이 지정된 Application Load Balancer를 생성합니다. 자세한 내용은 탄력성 로드 밸런싱 설명서를 참조하세요.</p> <ul style="list-style-type: none"> 로드 밸런서 이름을 입력합니다. 계획에서 Internet-facing을 선택합니다. 네트워크 매핑 섹션에서 올바른 VPC와 서로 다른 가용 영역의 퍼블릭 서브넷을 두 개 이상 선택합니다. 리스너 및 라우팅 섹션에서 대상 그룹 PSFTWEB을(를) 선 	클라우드 관리자

작업	설명	필요한 기술
	<p>택하고 올바른 프로토콜과 포트 번호를 지정합니다.</p>	
<p>Elasticsearch 서버를 위한 로드 밸런서를 생성하세요.</p>	<p>Elasticsearch 서버로 향하는 트래픽을 로드 밸런싱하기 위해 이름이 지정된 PSFTSCH Application Load Balancer를 생성하십시오.</p> <ul style="list-style-type: none"> • 로드 밸런서 이름을 입력합니다. • 계획에서 내부 를 선택합니다. • 네트워크 매핑 섹션에서 올바른 VPC와 프라이빗 서브넷을 선택합니다. • 리스너 및 라우팅 섹션에서 대상 그룹 PSFTSRCH을(를) 선택하고 올바른 프로토콜과 포트 번호를 지정합니다. 	<p>클라우드 관리자</p>
<p>Route 53을 구성합니다.</p>	<p>Amazon Route 53 콘솔에서 PeopleSoft 애플리케이션을 서비스할 호스팅 영역에 레코드를 생성합니다. 지침은 Amazon Route 53 설명서를 참조하십시오. 이렇게 하면 모든 트래픽이 PSFTLB 로드 밸런서를 통과하게 됩니다.</p>	<p>클라우드 관리자</p>

관련 리소스

- [오라클 PeopleSoft 웹 사이트](#)
- [AWS 설명서](#)

더 많은 패턴

- [AWS Copilot을 사용하여 클러스터링된 애플리케이션을 Amazon ECS에 배포](#)
- [테라폼을 사용하여 CloudWatch Synthetics 카나리아를 배포하세요](#)

클라우드 네이티브

주제

- [Amazon Kinesis Video Streams 및 AWS Fargate를 사용하여 비디오 처리 파이프라인 구축하기](#)
- [AWS CLI를 사용하여 S3 버킷에서 다른 계정 및 리전으로 데이터 복사](#)
- [AWS 서비스를 사용하여 SAP RHEL 페이스메이커 클러스터를 모니터링합니다.](#)
- [S3 버킷을 AWS CloudFormation 스택으로 성공적으로 가져오기](#)
- [패턴 더 보기](#)

Amazon Kinesis Video Streams 및 AWS Fargate를 사용하여 비디오 처리 파이프라인 구축하기

작성자: Piotr Chotkowski(AWS)와 Pushparaju Thangavel(AWS)

환경: PoC 또는 파일럿

기술: 클라우드 네이티브, 소프트웨어 개발 및 테스트, 미디어 서비스

AWS 서비스: AWS Fargate, Amazon Kinesis, Amazon S3

요약

이 패턴은 [Amazon Kinesis Video Streams](#)와 [AWS Fargate](#)를 사용하여 비디오 스트림에서 프레임을 추출하고 추가 처리를 위해 [Amazon Simple Storage Service\(S3\)](#)에 저장하는 방법을 보여줍니다.

이 패턴은 Java Maven 프로젝트 형태의 샘플 애플리케이션을 제공합니다. 이 애플리케이션은 [AWS Cloud Development Kit\(AWS CDK\)](#)를 사용하여 AWS 인프라를 정의합니다. 프레임 처리 로직과 인프라 정의 모두 Java 프로그래밍 언어로 작성되었습니다. 이 샘플 애플리케이션을 기반으로 실시간 비디오 처리 파이프라인을 개발하거나 기계 학습 파이프라인의 비디오 전처리 단계를 구축할 수 있습니다.

사전 조건 및 제한 사항

사전 조건

- 활성 상태의 AWS 계정.
- Java SE Development Kit (JDK) 11가 설치됨
- [Apache Maven](#)이 설치됨
- [AWS Cloud Development Kit\(AWS CDK\)](#)가 설치됨
- [AWS Command Line Interface\(AWS CLI\)](#) 버전 2가 설치됨
- [도커](#)(AWS Fargate 작업 정의에서 사용할 도커 이미지를 구축하는 데 필요)가 설치됨

제한 사항

이 패턴은 개념 증명 또는 추가 개발을 위한 기반으로 사용됩니다. 프로덕션 배포에는 현재 형태로 사용할 수 없습니다.

제품 버전

- 이 패턴은 AWS CDK 버전 1.77.0에서 테스트했습니다([AWS CDK 버전 참조](#))
- JDK 11
- CLI version 2

아키텍처

대상 기술 스택

- Amazon Kinesis Video Streams
- AWS Fargate 태스크
- Amazon Simple Queue Service(Amazon SQS) 대기열
- Amazon S3 버킷

대상 아키텍처

사용자는 Kinesis 비디오 스트림을 생성하고, 비디오를 업로드하며, 입력 Kinesis 비디오 스트림 및 출력 S3 버킷에 대한 세부 정보가 포함된 JSON 메시지를 SQS 대기열로 전송합니다. 컨테이너에서 기본 애플리케이션을 실행하는 AWS Fargate는 SQS 대기열에서 메시지를 가져와 프레임 추출을 시작합니다. 각 프레임은 이미지 파일에 저장되고 대상 S3 버킷에 저장됩니다.

자동화 및 규모 조정

샘플 애플리케이션은 단일 AWS 리전 내에서 수평 및 수직으로 규모를 조정할 수 있습니다. SQS 대기열에서 읽는 배포된 AWS Fargate 작업 수를 늘리면 수평적 규모 조정이 가능합니다. 애플리케이션의 프레임 분할 및 이미지 게시 스레드 수를 늘리면 수직적 규모 조정이 가능합니다. 이러한 설정은 AWS CDK의 [QueueProcessingFargateService](#) 리소스 정의에서 애플리케이션에 환경 변수로 전달됩니다. AWS CDK 스택 배포의 특성상 추가 노력 없이 이 애플리케이션을 여러 AWS 리전 및 계정에 배포할 수 있습니다.

도구

도구

- [AWS CDK](#)는, Python, Java TypeScript JavaScript, C#.Net 등의 프로그래밍 언어를 사용하여 클라우드 인프라 및 리소스를 정의하기 위한 소프트웨어 개발 프레임워크입니다.

- [Amazon Kinesis Video Streams](#)는 라이브 비디오를 디바이스에서 AWS 클라우드로 스트리밍하거나 실시간 비디오 처리 또는 배치 중심 비디오 분석을 위한 애플리케이션을 빌드하는 데 사용할 수 있는 완전 관리형 AWS 서비스입니다.
- [AWS Fargate](#)는 컨테이너용 서버리스 컴퓨팅 엔진입니다. Fargate를 사용하면 서버를 프로비저닝하고 관리할 필요가 없으므로 애플리케이션 개발에만 집중할 수 있습니다.
- [Amazon S3](#)는 업계 최고의 확장성, 데이터 가용성, 보안 및 성능을 제공하는 객체 스토리지 서비스입니다.
- [Amazon SQS](#)는 마이크로서비스와 분산 시스템, 서버리스 애플리케이션을 분리하거나 확장하기 쉽게 해 주는 완전 관리형 메시지 대기열 서비스입니다.

코드

- 샘플 애플리케이션 프로젝트(`frame-splitter-code.zip`) 의.zip 파일이 첨부되어 있습니다.

에픽

인프라 배포

작업	설명	필요한 기술
Docker 대몬(daemon)을 시작합니다.	로컬 시스템에서 Docker 대몬(daemon)을 시작합니다. AWS CDK는 Docker를 사용하여 AWS Fargate 작업에 사용되는 이미지를 구축합니다. 다음 단계를 진행하기 전에 Docker를 실행해야 합니다.	개발자, 엔지니어 DevOps
프로젝트를 빌드합니다.	<code>frame-splitter-code</code> 샘플 애플리케이션(첨부)을 다운로드하고 해당 콘텐츠를 로컬 시스템의 폴더에 추출합니다. 인프라를 배포하려면 먼저 Java Maven 프로젝트를 빌드해야 합니다. 명령 프롬프트에서 프로젝트의 루트 디렉터리	개발자, DevOps 엔지니어

작업	설명	필요한 기술
	<p>로 이동하고 다음 명령을 실행하여 프로젝트를 빌드합니다.</p> <pre>mvn clean install</pre>	
<p>AWS CDK를 부트스트랩합니다.</p>	<p>(AWS CDK를 처음 사용하는 사용자만 해당) AWS CDK를 처음 사용하는 경우, 다음에서 AWS CLI 명령을 실행하여 환경을 부트스트랩해야 할 수 있습니다.</p> <pre>cdk bootstrap --profile "\$AWS_PROFILE_NAME"</pre> <p><code>\$AWS_PROFILE_NAME</code> 이 AWS 보안 인증 정보의 AWS 프로필 이름을 보유할 때. 또는 이 파라미터를 제거하여 기본 프로필을 사용할 수 있습니다. 자세한 내용은 AWS CDK 설명서를 참조하세요.</p>	<p>개발자, DevOps 엔지니어</p>

작업	설명	필요한 기술
AWS CDK 스택을 배포합니다.	<p>이 단계에서는 AWS 계정에서 필요한 인프라 리소스(SQS 대기열, S3 버킷, AWS Fargate 작업 정의)를 생성하고, AWS Fargate 작업에 필요한 Docker 이미지를 구축하고, 애플리케이션을 배포합니다. 명령 프롬프트에서 프로젝트의 루트 디렉터리로 이동하고 다음 명령을 실행합니다.</p> <pre data-bbox="597 730 1026 886">cdk deploy --profile "\$AWS_PROFILE_NAME" --all</pre> <p><code>\$AWS_PROFILE_NAME</code> 이 AWS 보안 인증 정보의 AWS 프로필 이름을 보유할 때. 또는 이 파라미터를 제거하여 기본 프로필을 사용할 수 있습니다. 배포를 확인합니다. CDK 배포 출력의 <code>QueueUrl</code> 및 버킷 값을 기록해 두십시오. 이 값은 이후 단계에서 필요합니다. AWS CDK는 자산을 생성하고, 이를 AWS 계정에 업로드하고, 모든 인프라 리소스를 생성합니다. AWS CloudFormation 콘솔에서 리소스 생성 프로세스를 관찰할 수 있습니다. 자세한 내용은 AWS CloudFormation 설명서 및 AWS CDK 설명서를 참조하십시오.</p>	개발자, 엔지니어 DevOps

작업	설명	필요한 기술
<p>비디오 스트림을 생성합니다.</p>	<p>이 단계에서는 비디오 처리를 위한 입력 스트림으로 사용할 Kinesis 비디오 스트림을 생성합니다. AWS CLI가 설치되고 구성되어 있는지 확인합니다. AWS CLI에서 다음을 실행합니다.</p> <pre data-bbox="592 583 1027 903">aws kinesisvideo --profile "\$AWS_PROFILE_NAME" create-stream --stream-name "\$STREAM_NAME" --data-retention-in-hours "24"</pre> <p>\$AWS_PROFILE_NAME 은(는) AWS 보안 인증 정보의 AWS 프로필 이름을 보유하고 있으며(또는 기본 프로필을 사용하려면 이 파라미터를 제거), \$STREAM_NAME 은(는) 임의의 유효한 스트림 이름일 때.</p> <p>또는 Kinesis 비디오 스트림 설명서의 단계에 따라 Kinesis 콘솔을 사용하여 비디오 스트림을 생성할 수도 있습니다. 생성된 스트림의 AWS 리소스 이름(ARN)을 적어 둡니다. 나중에 필요할 수 있습니다.</p>	<p>개발자, DevOps 엔지니어</p>

예제 실행

작업	설명	필요한 기술
<p>비디오를 스트림에 업로드합니다.</p>	<p>샘플 <code>frame-splitter-code</code> 애플리케이션의 프로젝트 폴더에서 <code>src/test/java/amazon/awscdk/examples/splitter</code> 폴더에 있는 <code>ProcessingTaskTest.java</code> 파일을 엽니다. <code>profileName</code> 및 <code>streamName</code> 변수를 이전 단계에서 사용한 값으로 대체합니다. 이전 단계에서 만든 Kinesis 비디오 스트림에 예시 비디오를 업로드하려면 다음을 실행합니다.</p> <div style="border: 1px solid #ccc; border-radius: 10px; padding: 10px; margin: 10px 0;"> <pre>amazon.awscdk.examples.splitter.ProcessingTaskTest#testExample test</pre> </div> <p>또는 Kinesis 비디오 스트림 설명서에 설명된 방법 중 하나를 사용하여 비디오를 업로드할 수 있습니다.</p>	<p>개발자, DevOps 엔지니어</p>
<p>비디오 프로세싱을 시작합니다.</p>	<p>Kinesis 비디오 스트림에 비디오를 업로드했으므로 이제 처리를 시작할 수 있습니다. 처리 로직을 시작하려면 AWS CDK가 배포 중에 생성한 세부 정보가 포함된 메시지를 SQS 대기열로 보내야 합니다. AWS CLI</p>	<p>개발자, 엔지니어 DevOps</p>

작업	설명	필요한 기술
	<p>를 사용하여 메시지를 보내려면 다음을 실행합니다.</p> <pre data-bbox="597 331 1026 569">aws sqs --profile "\$AWS_PROFILE_NAME" send-message --queue-u rl QUEUE_URL --message -body MESSAGE</pre> <p>여기서 \$AWS_PROF ILE_NAME 는 AWS 자격 증 명의 AWS 프로파일 이름을 보유 하고 있으며 (기본 프로파일을 사 용하려면 이 파라미터 제거), QUEUE_URL 는 AWS CDK 출 력의 QueueUrl값이고, 는 다음 형식의 JSON MESSAGE 문자열 입니다.</p> <pre data-bbox="597 1066 1026 1304">{ "streamARN": "STREAM_ARN", "bucket": "BUCKET_N AME", "s3Directory": "test-output" }</pre> <p>STREAM_ARN 은(는) 이전 단 계에서 생성한 비디오 스트 림의 ARN이고, BUCKET_NA ME 은(는) AWS CDK 출력의 버킷 값일 때.</p> <p>이 메시지를 보내면 비디오 처리가 시작됩니다. Amazon SQS 설명서에 설명된 대로 Amazon SQS 콘솔을 사용하여 메시지를 보낼 수도 있습니다.</p>	

작업	설명	필요한 기술
비디오 프레임 이미지 보기.	s3://BUCKET_NAME/test-output BUCKET_NAME 이(가) AWS CDK 출력의 버킷 값일 때, S3 출력 버킷에서 결과 이미지를 볼 수 있습니다.	개발자, DevOps 엔지니어

관련 리소스

- [AWS CDK 설명서](#)
- [AWS CDK API 레퍼런스](#)
- [AWS CDK 입문 워크숍](#)
- [Amazon Kinesis Video Streams 설명서](#)
- [예: 를 사용하여 비디오 스트림의 객체 식별 SageMaker](#)
- [예: Kinesis Video Streams 프래그먼트의 구문 분석 및 렌더링](#)
- [Amazon Kinesis Video Streams와 Amazon을 사용하여 대규모 라이브 비디오를 실시간으로 분석할 수 있습니다 \(AWS Machine Learning 블로그 게시물\). SageMaker](#)
- [AWS Fargate 시작하기](#)

추가 정보

IDE 선택

선호하는 Java IDE를 사용하여 이 프로젝트를 빌드하고 탐색하는 것이 좋습니다.

정리

이 예제를 모두 실행한 후에는 배포된 리소스를 모두 제거하여 추가 AWS 인프라 비용이 발생하지 않도록 하십시오.

인프라 및 비디오 스트림을 제거하려면 AWS CLI에서 다음 두 명령을 사용하십시오.

```
cdk destroy --profile "$AWS_PROFILE_NAME" --all
```

```
aws kinesisisvideo --profile "$AWS_PROFILE_NAME" delete-stream --stream-arn "$STREAM_ARN"
```

또는 AWS 콘솔을 사용하여 AWS CloudFormation 스택을 제거하고 Kinesis CloudFormation 콘솔을 사용하여 Kinesis 비디오 스트림을 제거하여 리소스를 수동으로 제거할 수 있습니다. 단, `cdk destroy`은(는) 출력 S3 버킷이나 Amazon Elastic Container Registry(Amazon ECR) 리포지토리(`aws-cdk/assets`)의 이미지를 제거하지 않습니다. 수동으로 제거해야 합니다.

첨부

이 문서와 관련된 추가 콘텐츠에 액세스하려면 [attachment.zip](#) 파일의 압축을 풉니다.

AWS CLI를 사용하여 S3 버킷에서 다른 계정 및 리전으로 데이터 복사

작성자: Appasaheb Bagali(AWS), Purushotham G K(AWS)

환경: 프로덕션

기술: 클라우드 네이티브, 보안, ID, 규정 준수, 스토리지 및 백업, 마이그레이션, 서버리스, 현대화

워크로드: 기타 모든 워크로드, Microsoft

AWS 서비스: 아마존 S3, AWS CLI, AWS 자격 증명 및 액세스 관리, AWS KMS, AWS 관리 콘솔

요약

이 패턴은 AWS 소스 계정의 Amazon Simple Storage Service(S3) 버킷에서 동일한 AWS 리전이나 다른 리전에 있는 다른 AWS 계정의 대상 S3 버킷으로 데이터를 마이그레이션하는 방법을 설명합니다.

소스 S3 버킷은 연결된 리소스 정책을 사용하여 AWS Identity 및 Access Management(IAM) 액세스를 허용합니다. 대상 계정의 사용자는 소스 버킷에 대한 권한이 있는 PutObject 및 GetObject의 역할을 맡아야 합니다. 마지막으로 copy 및 sync 명령을 실행하여 원본 S3 버킷에서 대상 S3 버킷으로 데이터를 전송합니다.

계정은 S3 버킷에 업로드한 객체를 소유합니다. 계정 및 리전 간에 객체를 복사하는 경우 대상 계정에 복사한 객체의 소유권을 부여합니다. [액세스 제어 목록\(ACL\)](#) 을 bucket-owner-full-control로 변경하여 객체의 소유권을 변경할 수 있습니다. 그러나 여러 객체에 대해 ACL을 관리하기가 어려울 수 있으므로 대상 계정에 프로그래밍 방식의 교차 계정 권한을 부여하는 것이 좋습니다.

경고: 이 시나리오에서는 IAM 사용자에게 프로그래밍 액세스 및 장기 자격 증명에 필요한 권한만 제공하고 더 이상 필요하지 않을 경우 이러한 사용자를 제거하는 것이 좋습니다. 필요한 경우 액세스 키를 업데이트할 수 있습니다. 자세한 내용은 IAM 사용 설명서의 [액세스 키 업데이트](#)를 참조하세요.

사전 조건 및 제한 사항

- 동일한 AWS 리전 또는 서로 다른 AWS 리전의 활성 AWS 계정 2개.
- 소스 계정의 기존 S3 버킷
- 소스 또는 대상 Amazon S3 버킷에 [기본 암호화](#)가 활성화되어 있는 경우 AWS Key Management Service(AWS KMS) 키 권한을 수정해야 합니다. 자세한 내용은 이 주제에 대한 [AWS re:Post 문서](#)를 참조하세요.
- 교차 계정 권한에 대해 잘 알고 있어야 합니다.

아키텍처

도구

- [Amazon Simple Storage Service\(S3\)](#)는 원하는 양의 데이터를 저장, 보호 및 검색하는 데 도움이 되는 클라우드 기반 객체 스토리지 서비스입니다.
- [AWS Command Line Interface\(AWS CLI\)](#)는 명령줄 셸에서 명령을 사용하여 AWS 서비스와 상호 작용할 수 있는 오픈 소스 도구입니다.
- [AWS Identity and Access Management\(IAM\)](#)은 사용자에게 대한 인증 및 권한 부여를 제어함으로써 AWS 리소스에 대한 액세스를 안전하게 관리할 수 있습니다.

모범 사례

- [IAM의 보안 모범 사례](#)(IAM 설명서)
- [최소 권한 적용](#)(IAM 설명서)

에픽

대상 AWS 계정에서 IAM 사용자 및 역할 생성

작업	설명	필요한 기술
IAM 사용자를 생성하고 액세스 키를 받습니다.	1. AWS Management Console에 로그인하고 프로그래밍	AWS DevOps

작업	설명	필요한 기술
	<p>방식으로 액세스할 수 있는 IAM 사용자를 생성합니다. 자세한 단계는 IAM 설명서의 IAM 사용자 생성을 참조하세요. 이 사용자에게 대한 정책은 추가할 필요가 없습니다.</p> <p>2. 이 사용자의 액세스 키와 비밀 키를 생성합니다. 지침은 AWS 설명서의 AWS 계정 및 액세스 키를 참조하세요.</p>	

작업	설명	필요한 기술
<p>자격 증명 기반 IAM 정책을 생성합니다.</p>	<p>다음 권한을 사용하여 S3MigrationPolicy 라는 이름이 지정된 IAM ID 기반 정책을 생성합니다. 자세한 내용은 IAM 설명서의 IAM 정책 생성을 참조하세요.</p> <pre data-bbox="597 537 1029 1862"> { "Version": "2012-10-17", "Statement": [{ "Effect": "Allow", "Action": ["s3:ListBucket", "s3:GetObject", "s3:GetObjectTagging", "s3:GetObjectVersion", "s3:GetObjectVersionTagging"], "Resource": ["arn:aws:s3:::awsexamplesourcebucket", "arn:aws:s3:::awsexamplesourcebucket/*"] }] } </pre>	<p>AWS DevOps</p>

작업	설명	필요한 기술
	<pre> "Effect": "Allow", "Action": ["s3:ListBucket", "s3:PutObject", "s3:PutObjectAcl", "s3:PutObjectTagging", "s3:GetObjectTagging", "s3:GetObjectVersion", "s3:GetObjectVersionTagging"], "Resource": ["arn:aws:s3:::awsexampledestinationbucket", "arn:aws:s3:::awsexampledestinationbucket/*"] }] } </pre> <p>참고: 사용 사례에 따라 원본 및 대상 버킷 이름을 수정합니다.</p>	

작업	설명	필요한 기술
	이 ID 기반 정책을 통해 이 역할을 수임하는 사용자는 원본 버킷과 대상 버킷에 액세스할 수 있습니다.	

작업	설명	필요한 기술
IAM 역할을 생성합니다.	<p>다음 신뢰 정책을 사용하여 S3MigrationRole 이라는 이름이 지정된 IAM 역할을 생성한 후 이전에 생성한 S3MigrationPolicy 를 연결합니다. 자세한 단계는 IAM 설명서의 역할을 생성하여 IAM 사용자에게 권한 위임을 참조 하세요.</p> <pre data-bbox="592 682 1031 1554"> { "Version": "2012-10-17", "Statement": [{ "Effect": "Allow", "Principal": { "AWS": "arn:aws:iam:<destination_account>: user/<user_name>" }, "Action": "sts:AssumeRole", "Condition": {} }] } </pre> <p>참고: 사용 사례에 따라 신뢰 정책에서 대상 IAM 역할 또는 사용자 이름의 Amazon 리소스 이름 (ARN) 을 수정하십시오.</p>	AWS DevOps

작업	설명	필요한 기술
	이 신뢰 정책은 새로 생성한 IAM 사용자가 S3MigrationRole 을 가정하도록 허용합니다.	

소스 계정에 S3 버킷 정책 생성 및 연결

작업	설명	필요한 기술
S3 버킷 정책을 생성하고 연결합니다.	<p>소스 계정의 AWS Management Console에 로그인하고 Amazon S3 콘솔을 엽니다. S3 버킷을 선택한 다음 권한을 선택합니다. 버킷 정책에서 편집을 선택한 다음, 다음 버킷 정책에 붙여 넣습니다. 저장을 선택합니다.</p> <pre> { "Version": "2012-10-17", "Statement": [{ "Sid": "DelegateS3Access", "Effect": "Allow", "Principal": {"AWS": "arn:aws:iam::<destination_account>:role/<RoleName>"}, "Action": ["s3:ListBucket", "s3:GetObject", </pre>	클라우드 관리자

작업	설명	필요한 기술
	<pre> "s3:GetObjectTagging", "s3:GetObjectVersion", "s3:GetObjectVersionTagging"], "Resource": ["arn:aws:s3:::awsexamplesourcebucket/*", "arn:aws:s3:::awsexamplesourcebucket"]] } </pre> <p>참고: 대상 계정의 AWS 계정 ID를 포함하고 요구 사항에 따라 버킷 정책 템플릿을 구성해야 합니다.</p> <p>이 리소스 기반 정책은 대상 역할 S3MigrationRole 이 소스 계정의 S3 객체에 액세스할 수 있도록 허용합니다.</p>	

대상 S3 버킷 구성

작업	설명	필요한 기술
대상 S3 버킷을 생성합니다.	대상 계정의 AWS Management Console에 로그인하고 Amazon S3 콘솔을 연 다음 버킷 생성을 선택합니다. 요구 사항에 따라 S3 버킷을 생성합니다. 자세한 내용은 Amazon S3 설명서의 버킷 생성 을 참조하세요.	클라우드 관리자

대상 S3 버킷으로 데이터 복사

작업	설명	필요한 기술
새로 생성한 사용자 보안 인증 정보로 AWS CLI를 구성합니다.	<ol style="list-style-type: none"> AWS CLI의 최신 릴리스를 설치합니다. 지침은 AWS CLI 설명서에서 최신 버전의 AWS CLI 설치 또는 업데이트를 참조하세요. 생성한 사용자의 AWS 액세스 키로 \$ aws configure 를 실행하고 CLI를 업데이트합니다. 자세한 내용은 AWS CLI 설명서의 구성 및 보안 인증 파일 설정을 참조하세요. 	AWS DevOps
S3 마이그레이션 역할을 수입합니다.	<ol style="list-style-type: none"> AWS CLI를 사용하여 S3MigrationRole 을 가정해 보세요. <div style="border: 1px solid #ccc; border-radius: 10px; padding: 5px; margin-top: 10px;"> <pre>aws sts assume-role \</pre> </div> 	AWS 관리자

작업	설명	필요한 기술
	<pre data-bbox="634 205 1027 506"> --role-arn "arn:aws:iam::<des tination_account>: role/S3MigrationRo le" \ --role-session- name AWSCLI-Session </pre> <p data-bbox="630 541 1027 1388">이 명령은 여러 정보를 출력합니다. 보안 인증 정보 블록 내에는 AccessKeyId , SecretAccessKey 및 SessionToken 이 필요합니다. 이 예제에서는 환경 변수 RoleAccessKeyID , RoleSecretKey , 및 RoleSessionToken 를 사용합니다. 참고로 만료 필드의 타임스탬프는 UTC 표준 시간대로 표시됩니다. 타임스탬프는 IAM 역할의 임시 보안 인증이 만료되는 시기를 나타냅니다. 임시 보안 인증이 만료되면 sts:AssumeRole API를 다시 직접적으로 호출해야 합니다.</p> <p data-bbox="591 1415 1016 1591">2. IAM 역할을 맡을 환경 변수 3개를 생성합니다. 이러한 환경 변수는 다음과 같은 출력으로 채워집니다.</p> <pre data-bbox="634 1633 1027 1799"> # Linux export AWS_ACCES S_KEY_ID=RoleAcces sKeyID </pre>	

작업	설명	필요한 기술
	<pre data-bbox="633 210 990 850"> export AWS_SECRET_ACCESS_KEY=RoleSecretKey export AWS_SESSION_TOKEN=RoleSessionToken # Windows set AWS_SECRET_ACCESS_KEY_ID=RoleSecretKeyID set AWS_SECRET_ACCESS_KEY=RoleSecretKey set AWS_SESSION_TOKEN=RoleSessionToken </pre> <p data-bbox="592 871 998 1008">3. 다음 명령을 실행하여 IAM 역할을 가정했는지 검증합니다.</p> <pre data-bbox="633 1050 990 1165"> aws sts get-caller-identity </pre> <p data-bbox="592 1228 998 1323">자세한 정보는 AWS 지식 센터를 참조하세요.</p>	

작업	설명	필요한 기술
<p>소스 S3 버킷의 데이터를 대상 S3 버킷으로 복사 및 동기화합니다.</p>	<p>S3MigrationRole 역할을 맡았으면 복사(cp) 또는 동기화(sync) 명령을 사용하여 데이터를 복사할 수 있습니다.</p> <p>복사(자세한 내용은 AWS CLI 명령 참조 참조):</p> <pre data-bbox="594 569 1027 1005">aws s3 cp s3:// DOC-EXAMPLE-BUCKET-SOURCE / \ s3:// DOC-EXAMPLE-BUCKET-TARGET / \ --recursive -- source-region SOURCE-REGION-NAME --region DESTINATION-REGION-NAME</pre> <p>동기화(자세한 내용은 AWS CLI 명령 참조 참조):</p> <pre data-bbox="594 1163 1027 1558">aws s3 sync s3:// DOC-EXAMPLE-BUCKET-SOURCE / \ s3:// DOC-EXAMPLE-BUCKET-TARGET / \ --source-region SOURCE-REGION-NAME --region DESTINATION-REGION-NAME</pre>	<p>클라우드 관리자</p>

문제 해결

문제	Solution
<p>ListObjects 작업을 호출할 때 다음과 같은 오류가 발생했습니다(AccessDenied). 액세스 거부</p>	<ul style="list-style-type: none"> • S3MigrationRole 역할을 맡았는지 확인하세요. • <code>aws sts get-caller-identity</code> 를 실행하여 사용된 역할을 확인합니다. 출력에 S3MigrationRole 에 대한 ARN이 표시되지 않는 경우 역할을 다시 수입하고 다시 시도하세요.

관련 리소스

- [S3 버킷 생성](#)(Amazon S3 설명서)
- [Amazon S3 버킷 정책 및 사용자 정책](#)(Amazon S3 설명서)
- [IAM 자격 증명\(사용자, 그룹 및 역할\)](#)(IAM 설명서)
- [cp 명령](#)(AWS CLI 설명서)
- [동기화 명령](#)(AWS CLI 설명서)

AWS 서비스를 사용하여 SAP RHEL 페이스메이커 클러스터를 모니터링합니다.

제작자: 하쉬 토리아 (AWS), 랜디 저먼 (AWS), 라벤드라 부어 (AWS)

환경: 프로덕션	기술: 클라우드 네이티브, 인프라, 운영 체제	워크로드: SAP
AWS 서비스: 아마존 CloudWatch, 아마존 SNS, 아마존 CloudWatch 로그		

요약

이 패턴은 Amazon 및 Amazon Simple Notification Service (Amazon SNS) 를 사용하여 SAP 애플리케이션 및 SAP HANA 데이터베이스 서비스용 RHEL (RHEL) Pacemaker 클러스터에 대한 알림을 모니터링하고 구성하는 단계를 설명합니다. CloudWatch

이 구성을 사용하면 SAP SCS 또는 ASCS, 대기열에 넣기 복제 서버 (ERS) 및 SAP HANA 클러스터 리소스가 “중지됨” 상태에 있을 때 CloudWatch 로그 스트림, 지표 필터 및 경보를 사용하여 모니터링할 수 있습니다. Amazon SNS는 인프라 또는 SAP Basis 팀에 중지된 클러스터 상태에 대한 이메일을 보냅니다.

AWS CloudFormation 스크립트나 AWS 서비스 콘솔을 사용하여 이 패턴에 맞는 AWS 리소스를 생성할 수 있습니다. 이 패턴은 콘솔을 사용하고 있다고 가정하며, Amazon CloudWatch SNS의 인프라 배포나 CloudFormation 스크립트를 제공하지는 않습니다. Pacemaker 명령은 클러스터 알림 구성을 설정하는 데 사용됩니다.

사전 조건 및 제한 사항

사전 조건

- 활성 상태의 AWS 계정.
- Amazon SNS는 이메일 또는 모바일 알림을 전송하도록 설정되었습니다.
- ABAP용 SAP ASCS/ERS 또는 자바용 SCS/ERS, 그리고 SAP HANA 데이터베이스 RHEL 페이스메이커 클러스터. 지침은 다음을 참조하세요.

- [SAP HANA 클러스터 설정](#)
- [SAP 넷위버 ABAP/자바 클러스터 설치](#)

제한 사항

- 이 솔루션은 현재 RHEL 버전 7.3 이상의 페이스메이커 기반 클러스터에서 작동합니다. SUSE 운영 체제에서는 테스트되지 않았습니다.

제품 버전

- RHEL 7.3 이상

아키텍처

대상 기술 스택

- RHEL 페이스메이커 알림 이벤트 기반 에이전트
- Amazon Elastic Compute Cloud(Amazon EC2)
- CloudWatch 알람
- CloudWatch 로그 그룹 및 메트릭 필터
- Amazon SNS

대상 아키텍처

다음 다이어그램은 이 솔루션의 구성 요소와 워크플로를 보여줍니다.

자동화 및 규모 조정

- 스크립트를 사용하여 AWS 리소스 생성을 자동화할 수 있습니다. CloudFormation 또한 추가 지표 필터를 사용하여 여러 클러스터를 확장하고 포괄할 수 있습니다.

도구

서비스

- [Amazon](#)은 실행 중인 AWS 리소스 및 애플리케이션의 지표를 AWS 실시간으로 모니터링할 수 있도록 CloudWatch 도와줍니다.
- [Amazon Simple Notification Service\(Amazon SNS\)](#)를 사용하면 웹 서버 및 이메일 주소를 포함하여 게시자와 클라이언트 간의 메시지 교환을 조정하고 관리할 수 있습니다.

도구

- CloudWatch 에이전트 (통합) 는 EC2 인스턴스에서 시스템 수준 지표, 로그 및 추적을 수집하고 애플리케이션에서 사용자 지정 지표를 검색하는 도구입니다.
- Pacemaker 알림 에이전트 (RHEL 7.3 이상용) 는 Pacemaker 클러스터에서 리소스가 중지되거나 다시 시작될 때와 같이 변경이 있을 때 작업을 시작하는 도구입니다.

모범 사례

- 에서 AWS SAP 워크로드를 사용하는 모범 사례는 AWS Well-Architected 프레임워크를 위한 [SAP Lens](#)를 참조하십시오.
- SAP HANA 클러스터에 대한 CloudWatch 모니터링을 설정하는 데 드는 비용을 생각해 보세요. 자세한 내용은 [CloudWatch 설명서](#)를 참조하십시오.
- Amazon SNS 알림에는 페이지 또는 티켓팅 메커니즘을 사용해 보십시오.
- PC, 페이스메이커 및 펜싱 에이전트용 RPM 패키지의 RHA (고가용성) 버전이 있는지 항상 확인하십시오. AWS

에픽

Amazon SNS 설정

작업	설명	필요한 기술
SNS 주제를 생성합니다.	<ol style="list-style-type: none"> 1. AWS Management Console에 로그인하고 https://console.aws.amazon.com/sns/v3/home에서 Amazon SNS 콘솔을 엽니다. 2. Amazon SNS 대시보드의 [일반 작업(Common 	AWS 관리자

작업	설명	필요한 기술
	<p>actions)]에서 [주제 생성 (Create Topic)]을 선택합니다.</p> <ol style="list-style-type: none"> 3. 새 주제 만들기 대화 상자의 유형에서 표준을 선택합니다. 4. 주제 이름에 주제 이름 (예:my-topic) 을 입력합니다. 5. 주제 생성을 선택합니다. <p>그러면 알림을 게시할 수 있는 리소스 정책이 포함된 SNS 주제가 생성됩니다.</p> <ol style="list-style-type: none"> 6. 주제 ARN (예:) 을 복사합니다. <code>arn:aws:sns:us-east-1:111122223333:my-topic</code> 이 ARN은 이후 단계에서 사용하게 됩니다. 	

작업	설명	필요한 기술
<p>SNS 주제에 대한 액세스 정책을 수정하십시오.</p>	<ol style="list-style-type: none"> 1. Amazon SNS 콘솔의 탐색창에서 [Topics] 를 선택한 다음 생성한 주제를 선택합니다. 2. 편집을 선택하고 액세스 정책 섹션으로 이동합니다. 3. 액세스 정책에 이 주제에 게시할 수 있는 서비스 주체 중 하나가 포함되어 CloudWatch 있는지 확인하십시오. 예: <pre data-bbox="630 747 1029 1581"> { "Sid": "Allow AWS CloudWatch to Publish to this SNS topic", "Effect": "Allow", "Principal": { "Service": ["cloudwat ch.amazonaws.com"] }, "Action": "SNS:Publish", "Resource": "arn:aws:sns:us-ea st-1:111122223333: my-topic" } </pre> 4. 변경 사항 저장을 선택합니다. 	<p>AWS 시스템 관리자</p>

작업	설명	필요한 기술
SNS 주제를 구독합니다.	<ol style="list-style-type: none"> 1. Amazon SNS 콘솔의 탐색 창에서 구독, 구독 생성을 선택합니다. 2. Topic ARN의 경우 첫 번째 작업에서 생성한 ARN을 붙여넣습니다. 3. 프로토콜에서 이메일을 선택합니다. 4. Endpoint의 경우 SAP Pacemaker 클러스터를 담당하고 알림을 받아야 하는 사람이나 팀의 이메일 주소를 입력합니다. 예를 들어 이 주소는 SAP Basis 또는 인프라 팀의 배포 목록에 대한 이메일 주소일 수 있습니다. 5. 구독 생성을 선택합니다. 6. 이메일 애플리케이션에서 AWS 알림에서 보낸 메시지를 연 다음, 구독을 확인합니다. <p>웹 브라우저에 Amazon SNS의 확인 응답이 표시됩니다.</p>	AWS 시스템 관리자

클러스터 설정을 확인합니다.

작업	설명	필요한 기술
클러스터 상태를 확인합니다.	pcs status 명령을 사용하여 리소스가 온라인 상태인지 확인합니다.	SAP Basis 관리자

페이스메이커 알림을 구성하십시오.

작업	설명	필요한 기술
<p>기본 클러스터 인스턴스에서 Pacemaker 알림 에이전트를 구성합니다.</p>	<p>기본 클러스터의 EC2 인스턴스에 로그인하고 다음 명령을 실행합니다.</p> <pre data-bbox="597 499 1024 1528">install --mode=0755 /usr/share/pacemaker/alerts/alert_file.sh.sample touch /var/lib/pacemaker/alert_file.sh touch /var/log/pcmk_alert_file.log chown hacluster:haclient /var/log/pcmk_alert_file.log chmod 600 /var/log/pcmk_alert_file.log pcs alert create id=alert_file description="Log events to a file." path=/var/lib/pacemaker/alert_file.sh pcs alert recipient add alert_file id=my-alert_logfile value=/var/log/pcmk_alert_file.log</pre>	<p>SAP Basis 관리자</p>
<p>보조 클러스터 인스턴스에서 Pacemaker 알림 에이전트를 구성합니다.</p>	<p>보조 클러스터의 보조 클러스터 EC2 인스턴스에 로그인하고 다음 명령을 실행합니다.</p> <pre data-bbox="597 1738 1024 1837">install --mode=0755 /usr/share/pacemaker</pre>	<p>SAP Basis 관리자</p>

작업	설명	필요한 기술
<p>RHEL 알림 리소스가 생성되었는지 확인합니다.</p>	<pre> er/alerts/alert_file.sh.sample touch /var/lib/pacemaker/alert_file.sh touch /var/log/pcm_alert_file.log chown hacluster:haclient /var/log/pcm_alert_file.log chmod 600 /var/log/pcm_alert_file.log </pre> <p>다음 명령을 사용하여 알림 리소스가 생성되었는지 확인합니다.</p> <pre> pcs alert </pre> <p>명령의 출력은 다음과 같습니다.</p> <pre> [root@xxxxxxx ~]# pcs alert Alerts: Alert: alert_file (path=/var/lib/pacemaker/alert_file.sh) Description: Log events to a file. Recipients: Recipient: my-alert_logfile (value=/var/log/pcm_alert_file.log) </pre>	<p>SAP Basis 관리자</p>

CloudWatch 에이전트를 구성합니다.

작업	설명	필요한 기술
<p>CloudWatch 에이전트를 설치합니다.</p>	<p>EC2 인스턴스에 CloudWatch 에이전트를 설치하는 방법은 여러 가지가 있습니다. 명령줄을 사용하려면:</p> <ol style="list-style-type: none"> CloudWatch 에이전트 패키지 다운로드: <div data-bbox="630 674 1029 993" style="border: 1px solid #ccc; border-radius: 10px; padding: 10px; margin: 10px 0;"> <pre>wget https://s3.<region>.amazonaws.com/amazoncloudwatch-agent-region/redhat/amd64/latest/amazon-cloudwatch-agent.rpm</pre> </div> <p>EC2 인스턴스의 위치는 AWS 리전 어디입니까 (예:us-west-2). <region></p> 선택 사항) 패키지 서명을 확인합니다. 지침은 CloudWatch 설명서에서 CloudWatch 에이전트 패키지의 서명 확인을 참조하십시오. 첫 번째 인스턴스에 패키지 설치: <div data-bbox="630 1646 1029 1793" style="border: 1px solid #ccc; border-radius: 10px; padding: 10px; margin: 10px 0;"> <pre>sudo rpm -U ./amazon-cloudwatch-agent.rpm</pre> </div> 	<p>AWS 시스템 관리자</p>

작업	설명	필요한 기술
	<p>4. 보조 인스턴스에서도 반복합니다.</p> <p>자세한 내용은 CloudWatch 설명서를 참조하십시오.</p>	
<p>EC2 인스턴스에 IAM 역할을 연결합니다.</p>	<p>CloudWatch 에이전트가 인스턴스에서 데이터를 전송할 수 있게 하려면 IAM CloudWatchAgentServerRole 역할을 각 인스턴스에 연결해야 합니다. 또는 기존 IAM 역할에 CloudWatch 에이전트에 대한 정책을 추가할 수 있습니다. 자세한 내용은 CloudWatch 설명서를 참조하십시오.</p>	<p>AWS 관리자</p>

작업	설명	필요한 기술
<p>기본 클러스터 CloudWatch 인스턴스에서 Pacemaker 경고 에이전트 로그 파일을 모니터링하도록 에이전트를 구성합니다.</p>	<ol style="list-style-type: none"> 다음 명령을 실행하여 기본 클러스터 인스턴스를 구성합니다. <div data-bbox="630 394 1029 596" style="border: 1px solid #ccc; border-radius: 10px; padding: 10px; margin: 10px 0;"> <pre>sudo /opt/aws/amazon-cloudwatch-agent/bin/amazon-cloudwatch-agent-config-wizard</pre> </div> Linux용 1을 선택한 다음 모니터링 전략에 맞는 옵션을 선택합니다. “로그 파일을 모니터링하시겠습니까?” 라는 질문에 [예]를 선택하고 pcs alert 명령에서 Pacemaker 로그 파일의 경로를 제공하십시오. 우리의 경우에는 그렇습니다. <pre>var/log/pcmk_alert_file.log</pre> 로그 그룹 및 로그 스트림의 이름을 입력합니다. 로그 스트림을 지정하지 않으면 AWS 인스턴스 ID가 기본값으로 사용됩니다. 보조 클러스터 인스턴스에 대해 1~4단계를 반복합니다. 	<p>AWS 관리자</p>

작업	설명	필요한 기술
기본 및 보조 클러스터 인스턴스에서 CloudWatch 에이전트를 시작합니다.	<p>에이전트를 시작하려면 기본 및 보조 클러스터의 EC2 인스턴스에서 다음 명령을 실행합니다.</p> <pre>sudo /opt/aws/amazon-cloudwatch-agent/bin/amazon-cloudwatch-agent-ctl -a fetch-config -m ec2 -s -c file:/opt/aws/amazon-cloudwatch-agent/bin/config.json</pre>	AWS 관리자

리소스 설정 CloudWatch

작업	설명	필요한 기술
CloudWatch 로그 그룹을 설정합니다.	<ol style="list-style-type: none"> https://console.aws.amazon.com/cloudwatch/ 에서 CloudWatch 콘솔을 엽니다. 탐색 창에서 로그 그룹, 로그 그룹 생성을 선택합니다. 로그 그룹 이름을 입력한 다음 로그 그룹 생성을 선택합니다. <p>CloudWatch 에이전트는 Pacemaker 경고 파일을 CloudWatch 로그 그룹에 로그 스트림으로 전송합니다.</p>	AWS 관리자

작업	설명	필요한 기술
<p>CloudWatch 지표 필터를 설정합니다.</p>	<p>지표 필터를 사용하면 CloudWatch 로그 스트림과 같은 <code>stop <cluster-resource-name></code> 패턴을 검색할 수 있습니다. 이 패턴이 식별되면 지표 필터는 사용자 지정 지표를 업데이트합니다.</p> <ol style="list-style-type: none"> 1. CloudWatch 콘솔의 탐색 창에서 로그 그룹을 선택합니다. 2. 이전 작업에서 만든 로그 그룹 이름을 선택합니다. 3. 작업, 지표 필터 생성을 선택합니다. 4. 필터 패턴에는 사용할 필터 패턴 (예: 이름이 지정된 <code>ABC_scs SAP SCS</code> 클러스터 리소스의 중지 이벤트와 일치하도록) 을 입력합니다. <code>stop ABC_scs</code> <p>자세한 내용은 CloudWatch 설명서의 필터 패턴 구문을 참조하십시오.</p> <ol style="list-style-type: none"> 5. (선택 사항) 필터 패턴을 테스트하려면 테스트 패턴에 패턴을 테스트하는 데 사용할 로그 이벤트를 하나 이상 입력합니다. 줄 바꿈은 로그 이벤트 메시지 상자에서 로그 이벤트를 구분하는 데 사용되므로 각 로그 이벤트는 	<p>AWS 관리자, SAP Basis 관리자</p>

작업	설명	필요한 기술
	<p>별도의 줄에 지정해야 합니다.</p> <p>6. 다음을 선택하고 필터의 이름을 입력합니다.</p> <p>7. 메트릭 세부 정보에서 메트릭 네임스페이스에 메트릭을 게시할 CloudWatch 네임스페이스의 이름 (예:) 을 입력합니다. <code>sapcluster_monitoring</code> 이 네임스페이스가 아직 없는 경우 새로 만들기를 선택합니다.</p> <p>8. 메트릭 이름에 새 메트릭의 이름을 입력합니다 (예: 여기서 <code><sid></code> 는 SAP 시스템 ID 이름). <code>sapcluster_<sid></code></p> <p>9. 지표 값에는 1을 입력합니다.</p> <p>또는 다음과 같은 토큰을 입력할 수 <code>\$size</code> 있습니다. 이렇게 하면 <code>size</code> 필드가 포함된 모든 로그 이벤트에 대해 <code>size</code> 필드의 수치 값만큼씩 지표가 증가합니다.</p> <p>10. 기본값에는 0을 입력합니다.</p> <p>11. 지표 필터 생성을 선택합니다.</p> <p>지표 필터가 4단계에서 패턴을 식별하면 CloudWatch 사용자</p>	

작업	설명	필요한 기술
	<p>지정 지표의 값을 <code>sapcluster_abc</code> 1로 업데이트합니다.</p> <p>CloudWatch 경보는 지표를 <code>SAP-Cluster-QA1-ABC_sapcluster_abc</code> 모니터링하고 지표 값이 1로 변경되면 SNS 알림을 보냅니다. 이는 클러스터 리소스가 중지되었으며 조치를 취해야 함을 나타냅니다.</p>	

작업	설명	필요한 기술
<p>SAP ASCS/SCS 및 ERS 지표에 대한 CloudWatch 메트릭 경보를 설정합니다.</p>	<p>단일 지표를 기반으로 경보를 만들려면:</p> <ol style="list-style-type: none"> 1. CloudWatch 콘솔의 탐색 창에서 Alarms, All alarms를 선택합니다. 2. 경보 생성을 선택하세요. 3. 지표 선택을 선택합니다. 4. 이전 작업에서 만든 사용자 지정 지표를 <code>sapcluster_monitoring</code> 검색합니다. 5. 이전 작업에서도 만든 SAP SCS의 메트릭 이름 (예:<code>sapcluster_<abc></code>)을 선택합니다. 6. 그래프로 표시된 지표 탭에서 다음을 설정합니다. <ul style="list-style-type: none"> • 통계에서 최대를 선택합니다. • 기간에서 1분을 선택합니다. • 임계값 유형에서 정적을 선택하고 임계값을 1보다 크거나 같은 값으로 설정합니다. sapcluster_<sid> 7. 다음을 선택합니다. 8. 알림에서 첫 번째 에픽에서 만든 SNS 주제를 선택합니다. 	<p>AWS 관리자</p>

작업	설명	필요한 기술
	<p>9. 이름 및 설명에 알람 이름과 간략한 설명을 입력하고 다음을 선택합니다.</p> <p>10.경보 생성을 선택합니다.</p>	
SAP HANA CloudWatch 지표에 대한 메트릭 경보를 설정합니다.	<p>이전 작업의 CloudWatch 메트릭 경보 설정 단계를 반복하여 다음과 같이 변경하십시오.</p> <ul style="list-style-type: none"> 5단계에서는 SAP HANA의 메트릭 이름 (예:sapcluster_db_<abc>)을 선택합니다. 6단계에서는 의 임계값을 0보다 큰 sapcluster_<sid> 값으로 설정합니다. 	AWS 관리자

관련 리소스

- [클러스터 이벤트에 대한 스크립트 트리거 \(RHEL 설명서\)](#)
- [마법사를 사용하여 CloudWatch 에이전트 구성 파일 생성 \(설명서\)](#) CloudWatch
- [서버에 CloudWatch 에이전트 설치 및 실행 \(CloudWatch 설명서\)](#)
- [정적 임계값을 기반으로 CloudWatch 경보 생성 \(CloudWatch 설명서\)](#)
- [고가용성 클러스터를 사용하여 AWS에서 SAP HANA를 수동으로 배포 \(AWS웹 사이트의 SAP 설명서\)](#)
- [SAP NetWeaver 가이드 \(AWS웹 사이트의 SAP 설명서\)](#)

첨부

이 문서와 관련된 추가 콘텐츠에 액세스하려면 [attachment.zip](#) 파일의 압축을 풉니다.

S3 버킷을 AWS CloudFormation 스택으로 성공적으로 가져오기

작성자: Ram Kandaswamy(AWS)

환경: 프로덕션

기술: 클라우드 네이티브, 스토리지 및 백업

AWS 서비스: 아마존 S3, AWS CloudFormation

요약

Amazon Simple Storage Service (Amazon S3) 버킷과 같은 Amazon Web Services (AWS) 리소스를 사용하고 코드형 인프라 (IaC) 접근 방식을 사용하려는 경우 리소스를 CloudFormation AWS로 가져와서 스택으로 관리할 수 있습니다.

이 패턴은 S3 버킷을 AWS CloudFormation 스택으로 성공적으로 가져오는 단계를 제공합니다. 이 패턴의 접근 방식을 사용하면 S3 버킷을 한 번의 작업으로 가져올 때 발생할 수 있는 오류를 피할 수 있습니다.

사전 조건 및 제한 사항

사전 조건

- 활성 상태의 AWS 계정.
- 기존 S3 버킷 및 S3 버킷 정책. 이에 대한 자세한 내용은 AWS 지식 센터의 [AWS Config 규칙 bucket-ssl-requests-only s3-를 준수하기 위해 어떤 S3 버킷 정책을 사용해야 하는지를](#) 참조하십시오.
- 기존 AWS Key Management Service(AWS KMS) 키 및 별칭 이에 대한 자세한 내용은 AWS KMS 설명서의 [별칭 작업](#)을 참조하십시오.
- 샘플 CloudFormation-template-S3-bucket AWS CloudFormation 템플릿 (첨부) 이 로컬 컴퓨터에 다운로드됩니다.

아키텍처

이 다이어그램은 다음 워크플로를 보여줍니다.

1. 사용자는 JSON 또는 YAML 형식의 AWS 템플릿을 생성합니다. CloudFormation

2. 템플릿은 S3 버킷을 가져오기 위한 AWS CloudFormation 스택을 생성합니다.
3. AWS CloudFormation 스택은 템플릿에서 지정한 S3 버킷을 관리합니다.

기술 스택

- AWS CloudFormation
- AWS Identity and Access Management(IAM)
- AWS KMS
- Amazon S3

도구

- [AWS CloudFormation](#) — AWS는 예측 가능하고 반복적으로 AWS 인프라 배포를 생성하고 프로비저닝할 수 있도록 CloudFormation 지원합니다.
- [AWS Identity and Access Management\(IAM\)](#) – IAM은 AWS 서비스에 대한 액세스를 안전하게 제어하는 웹 서비스입니다.
- [AWS KMS](#) – AWS Key Management Service(AWS KMS)는 클라우드에 맞게 규모를 조정할 수 있는 암호화 및 키 관리 서비스입니다.
- [Amazon S3](#) – Amazon Simple Storage Service(S3)는 인터넷 스토리지 서비스입니다.

에픽

CMK 기반 암호화가 적용된 S3 버킷을 AWS 스택으로 가져오기 CloudFormation

작업	설명	필요한 기술
템플릿을 생성하여 S3 버킷과 CMK를 가져옵니다.	로컬 컴퓨터에서 다음 샘플 템플릿을 사용하여 S3 버킷과 CMK를 가져올 템플릿을 생성합니다.	AWS DevOps

```
AWSTemplateFormatVersion: 2010-09-09

Parameters:
```

작업	설명	필요한 기술
	<pre> bucketName: Type: String Resources: S3Bucket: Type: 'AWS::S3: :Bucket' DeletionPolicy: Retain Properties: BucketName: !Ref bucketName BucketEncryption: ServerSid eEncryptionConfigu ration: - ServerSid eEncryptionByDefault: SSEAlgori thm: 'aws:kms' KMSMaster KeyID: !GetAtt - KMSS3Encryption - Arn KMSS3Encryption: </pre>	

작업	설명	필요한 기술
	<pre> Type: 'AWS::KMS ::Key' DeletionPolicy: Retain Properties: Enabled: true KeyPolicy: !Sub - { "Id": "key- consolepolicy-3", "Version": "2012-10-17", "Statemen t": [{ "Sid": "Enable IAM User Permissions", "Effect": "Allow", "Principal": { "AWS": ["arn:aws:iam:: \${AWS::AccountId}:roo t"] }, },], }, </pre>	

작업	설명	필요한 기술
	<pre> "Action": "kms:*", "Resource": "*" } }] } EnableKey Rotation: true </pre>	
<p>스택을 생성합니다.</p>	<ol style="list-style-type: none"> 1. AWS Management 콘솔에 로그인하고, AWS CloudFormation 콘솔을 열고, 스택 보기를 선택하고, 스택 생성을 선택한 다음, 기존 리소스 사용 (리소스 가져오기) 을 선택합니다. 2. 템플릿 파일 업로드를 선택한 다음 이전에 만든 템플릿 파일을 업로드합니다. 3. 스택 이름을 입력하고 요구 사항에 따라 나머지 옵션을 구성합니다. 4. 스택 생성을 선택하고 스택 상태가 IMPORT_COMPLETE 로 변경될 때까지 기다립니다. 	<p>AWS DevOps</p>

작업	설명	필요한 기술
KMS 키 별칭을 생성합니다.	<ol style="list-style-type: none"> 1. AWS CloudFormation 콘솔에서 [Stacks] 를 선택하고, 이전에 생성한 스택의 이름을 선택하고, [템플릿] 창을 선택한 다음 [Designer에서 보기] 를 선택합니다. 2. 템플릿의 Resource 섹션에 다음 스니펫을 추가한 다음 스택 생성을 선택하고 마법사를 완료합니다. <pre data-bbox="594 772 1029 1411"> KMS_S3EncryptionAlias: Type: 'AWS::KMS::Alias' DeletionPolicy: Retain Properties: AliasName: alias/S3BucketKey TargetKeyId: !Ref KMS_S3Encryption </pre> <p>이에 대한 자세한 내용은 AWS CloudFormation 설명서의 AWS CloudFormation 스택 업데이트를 참조하십시오.</p>	AWS DevOps

작업	설명	필요한 기술
<p>S3 버킷 정책을 포함하도록 스택을 업데이트합니다.</p>	<ol style="list-style-type: none"> 1. AWS CloudFormation 콘솔에서 [Stacks] 를 선택하고, 이전에 생성한 스택의 이름을 선택하고, [템플릿] 창을 선택한 다음 [Designer에서 보기] 를 선택합니다. 2. 템플릿의 Resource 섹션에 다음 스니펫을 추가한 다음 스택 생성을 선택하고 마법사를 완료합니다. <pre data-bbox="597 772 1024 1862"> S3BucketPolicy: Type: 'AWS::S3: :BucketPolicy' Properties: Bucket: !Ref S3Bucket PolicyDocument: ! Sub - { "Version": "2008-10- 17", "Id": "restricthttp", "Statement": [{ </pre>	<p>AWS DevOps</p>

작업	설명	필요한 기술
	<pre> "sid": "denyhttp", "effect": "Deny", "principal": { "aws": "*" }, "action": "s3:*", "resource": ["arn:aws:s3:::\${S3Bucket}" , "arn:aws:s3:::\${S3Bucket}/*"], "condition": { "bool": { "aws:SecureTransport": "false" } } } </pre>	

작업	설명	필요한 기술
	<div style="border: 1px solid #ccc; border-radius: 10px; padding: 10px; margin-bottom: 10px;"> <pre style="margin: 0;">] } </pre> </div> <p>참고: 이 S3 버킷 정책에는 안전하지 않은 API 직접 호출을 제한하는 거부 명령문이 있습니다.</p>	
<p>키 정책을 업데이트합니다.</p>	<ol style="list-style-type: none"> 1. AWS CloudFormation 콘솔에서 [Stacks] 를 선택하고, 이전에 생성한 스택의 이름을 선택하고, [템플릿] 창을 선택한 다음 [Designer에서 보기] 를 선택합니다. 2. 관리자가 CMK를 관리할 수 있도록 허용하는 키 정책을 포함하도록 템플릿의 KMS 리소스를 수정합니다. 3. 스택 생성을 선택하고 다음을 선택한 후 요구 사항에 따라 마법사를 완료합니다. <p>이에 대한 자세한 내용은 AWS KMS 설명서의 AWS KMS의 키 정책 사용 및 키 관리자의 CMK 관리 허용을 참조하십시오.</p>	<p>AWS 관리자</p>

작업	설명	필요한 기술
<p>리소스 수준 태그를 추가합니다.</p>	<ol style="list-style-type: none"> 1. AWS CloudFormation 콘솔에서 [Stacks] 를 선택하고, 이전에 생성한 스택의 이름을 선택하고, [템플릿] 창을 선택한 다음 [Designer에서 보기] 를 선택합니다. 2. 템플릿의 Amazon S3 리소스 Properties 섹션에 다음 스니펫을 추가한 다음 스택 생성을 선택하고 마법사를 완료합니다. <div data-bbox="597 821 1027 1100" style="border: 1px solid #ccc; border-radius: 10px; padding: 10px; margin-top: 10px;"> <p>Tags:</p> <ul style="list-style-type: none"> - Key: createdBy Value: Cloudformation </div>	<p>AWS DevOps</p>

관련 리소스

- [기존 리소스를 AWS CloudFormation 관리로 가져오기](#)
- [AWS re:Invent 2017: AWS CloudFormation 심층 분석 \(동영상\)](#)

첨부

이 문서와 관련된 추가 콘텐츠에 액세스하려면 [attachment.zip](#) 파일의 압축을 풉니다.

패턴 더 보기

- [Session Manager 및 Amazon EC2 인스턴스 연결을 사용한 Bastion Host 액세스](#)
- [한 AWS 계정의 AWS CodeCommit 리포지토리를 다른 계정의 SageMaker Studio와 연결](#)
- [AWS Systems Manager를 사용하여 Windows 레지스트리 항목의 추가 또는 업데이트 자동화](#)
- [이상 탐지를 위한 Amazon Lookout for Vision 교육 및 배포를 자동화합니다.](#)
- [aws-nuke를 사용하여 AWS 리소스 자동 삭제](#)
- [AWS를 사용하여 AppStream 2.0 리소스 생성을 자동화합니다. CloudFormation](#)
- [CI/CD 파이프라인을 사용하여 Amazon EKS에 Java 애플리케이션 자동 구축 및 배포](#)
- [Python을 사용하여 AWS에서 자동으로 RFC 생성](#)
- [AWS Systems Manager Maintenance Windows를 사용하여 Amazon RDS DB 인스턴스를 자동 중지 및 시작](#)
- [Amazon EC2 Auto Scaling 및 Systems Manager를 사용하여 Micro Focus Enterprise Server PAC 구축하기](#)
- [서버리스 접근 방식을 사용하여 AWS 서비스를 함께 연결](#)
- [시작 시 EC2 인스턴스에 필수 태그가 있는지 확인](#)
- [AWS에서 VMware 클라우드를 NetBackup 사용하도록 베리타스를 구성합니다.](#)
- [Session Manager를 사용하여 Amazon EC2 인스턴스에 연결](#)
- [Amazon CloudWatch 이상 탐지를 사용하여 사용자 지정 지표에 대한 경보 생성](#)
- [Amazon ECS 작업 정의를 생성하고 Amazon EFS를 사용하여 EC2 인스턴스에 파일 시스템을 마운트](#)
- [Java 및 Python 프로젝트를 위한 동적 CI 파이프라인을 자동으로 생성](#)
- [태그 기반 Amazon CloudWatch 대시보드를 자동으로 생성](#)
- [AWS Copilot을 사용하여 클러스터링된 애플리케이션을 Amazon ECS에 배포](#)
- [Amazon S3에 React 기반 단일 페이지 애플리케이션을 배포하고 CloudFront](#)
- [Amazon EKS 클러스터의 배포 및 디버깅](#)
- [AWS CDK 및 AWS를 사용하여 AWS 컨트롤 타워 컨트롤을 배포하고 관리합니다. CloudFormation](#)
- [Terraform을 사용하여 AWS Control Tower 컨트롤을 배포하고 관리합니다.](#)
- [Elastic Beanstalk를 사용하여 컨테이너 배포](#)
- [컨테이너 이미지로 Lambda 함수 배포](#)
- [시작 시 Amazon RDS 데이터베이스의 자동 태그 지정 적용](#)

- [온디맨드 용량에 대한 DynamoDB 테이블의 비용 추정](#)
- [Green Boost를 통한 풀스택 클라우드 네이티브 웹 애플리케이션 개발 살펴보기](#)
- [AWS DMS를 사용하여 Amazon RDS for SQL Server 테이블을 S3 버킷으로 내보내기](#)
- [Amazon Personalize를 사용하여 개인화되고 순위가 다시 매겨진 추천 생성](#)
- [AWS Glue 작업과 Python을 사용하여 테스트 데이터 생성](#)
- [AWS KMS 키의 키 상태가 변경될 때 Amazon SNS 알림 받기](#)
- [DynamoDB 태깅 적용 지원](#)
- [Amazon Data Firehose 리소스가 AWS KMS 키로 암호화되지 않은 경우 식별 및 알림](#)
- [AWS Step Functions을 사용하여 서버리스 사가 패턴 구현](#)
- [AWS CDK를 사용하여 여러 AWS 지역, 계정 및 DevOps OU에서 Amazon Guru를 활성화하여 운영 성능을 개선하십시오.](#)
- [EC2 Windows 인스턴스를 수집하여 AWS Managed Services 계정으로 마이그레이션합니다.](#)
- [여러 AWS 계정 및 AWS 리전의 AWS Service Catalog 제품을 관리](#)
- [AWS DMS를 사용하여 Microsoft SQL 서버 데이터베이스를 Amazon EC2에서 Amazon DocumentDB로 마이그레이션](#)
- [DNS 레코드를 Amazon Route 53 프라이빗 호스팅 영역으로 대량 마이그레이션합니다.](#)
- [AWS DMS를 사용하여 오라클 8i 또는 9i에서 오라클용 아마존 RDS로 마이그레이션할 수 있습니다. SharePlex](#)
- [Amazon ElastiCache 클러스터의 미사용 암호화 모니터링](#)
- [시작 시 전송 중 암호화가 있는지 Amazon EMR 클러스터 모니터링](#)
- [보안 그룹의 ElastiCache 클러스터 모니터링](#)
- [Precisely Connect를 사용하여 메인프레임 데이터베이스를 AWS에 복제하기](#)
- [다중 지역, 다중 계정 조직에서 AWS CloudFormation 드리프트 감지 설정](#)
- [AWS Lambda를 사용하여 육각형 아키텍처로 Python 프로젝트 구조화](#)
- [C# 및 AWS CDK를 사용한 사일로 모델을 위한 SaaS 아키텍처의 테넌트 온보딩](#)
- [를 사용하여 AWS IAM 자격 증명 센터의 AWS CLI 자격 증명을 업데이트합니다. PowerShell](#)
- [Terraform을 사용하여 조직에서 GuardDuty Amazon을 자동으로 활성화하십시오.](#)
- [Splunk를 사용하여 AWS 네트워크 방화벽 로그 및 지표 보기](#)

컨테이너 및 마이크로서비스

주제

- [PrivateLink AWS와 Network Load Balancer를 사용하여 Amazon ECS에서 비공개로 컨테이너 애플리케이션에 액세스할 수 있습니다.](#)
- [AWS Fargate, PrivateLink AWS 및 Network Load Balancer를 사용하여 Amazon ECS에서 비공개로 컨테이너 애플리케이션에 액세스할 수 있습니다.](#)
- [PrivateLink AWS와 Network Load Balancer를 사용하여 Amazon EKS에서 비공개로 컨테이너 애플리케이션에 액세스합니다.](#)
- [Amazon EKS의 AWS 프라이빗 CA를 사용하여 AWS App Mesh에서 MTL을 활성화합니다.](#)
- [AWS Batch를 사용하여 Amazon RDS for PostgreSQL DB 인스턴스 백업 자동화](#)
- [CI/CD 파이프라인을 사용하여 Amazon EKS에서 노드 종료 핸들러 배포 자동화](#)
- [CI/CD 파이프라인을 사용하여 Amazon EKS에 Java 애플리케이션 자동 구축 및 배포](#)
- [Amazon ECS 작업 정의를 생성하고 Amazon EFS를 사용하여 EC2 인스턴스에 파일 시스템을 마운트](#)
- [Fargate를 사용하여 Amazon ECS에 Java 마이크로서비스 배포](#)
- [Amazon ECR과 AWS Fargate를 사용하여 Amazon ECS에 Java 마이크로서비스 배포](#)
- [Amazon ECR 및 로드 밸런싱을 사용하여 Amazon ECS에 Java 마이크로서비스 배포](#)
- [Amazon EKS와 Amazon S3의 차트 Helm 리포지토리를 사용하여 Kubernetes 리소스 및 패키지 배포](#)
- [컨테이너 이미지로 Lambda 함수 배포](#)
- [Amazon EKS에 샘플 Java 마이크로서비스를 배포하고 Application Load Balancer를 사용하여 마이크로서비스 노출하기](#)
- [AWS Copilot을 사용하여 클러스터링된 애플리케이션을 Amazon ECS에 배포](#)
- [Amazon EKS 클러스터에 gRPC 기반 애플리케이션을 배포하고 Application Load Balancer를 사용하여 액세스하기](#)
- [Amazon EKS 클러스터의 배포 및 디버깅](#)
- [Elastic Beanstalk를 사용하여 컨테이너 배포](#)
- [Lambda 함수, Amazon VPC 및 서버리스 아키텍처를 사용하여 정적 아웃바운드 IP 주소 생성](#)
- [쿠버네티스를 사용하여 Amazon EKS 작업자 노드에 SSM 에이전트를 설치합니다. DaemonSet](#)

- [다음을 사용하여 Amazon EKS 작업자 노드에 SSM CloudWatch 에이전트 및 에이전트를 설치합니다. preBootstrapCommands](#)
- [AWS App2Container 생성 도커 이미지 최적화](#)
- [노드 어피니티, 테인트 및 톨러레이션을 사용하여 Amazon EKS에 Kubernetes 포드 배치](#)
- [필터링된 Amazon ECR 컨테이너 이미지를 계정 또는 리전 전반적으로 복제](#)
- [컨테이너를 다시 시작하지 않고 데이터베이스 보안 인증 교체](#)
- [Amazon ECS Anywhere를 WorkSpaces 사용하여 Amazon에서 Amazon ECS 작업을 실행합니다.](#)
- [Amazon EC2 리눅스 인스턴스에서 ASP.NET 코어 웹 API Docker 컨테이너를 실행합니다.](#)
- [AWS Fargate를 사용하여 메시지 기반 워크로드를 대규모로 실행](#)
- [AWS Fargate와 함께 Amazon EKS에서 Amazon EFS를 사용하여 영구 데이터 스토리지로 스테이트풀 워크로드를 실행하십시오.](#)
- [패턴 더 보기](#)

PrivateLink AWS와 Network Load Balancer를 사용하여 Amazon ECS에서 비공개로 컨테이너 애플리케이션에 액세스할 수 있습니다.

작성자: Kirankumar Chandrashekar(AWS)

환경: 프로덕션

기술: 컨테이너 및 마이크로서비스, 네트워킹, 보안, ID, 규정 준수, 웹 및 모바일 앱

워크로드: 기타 모든 워크로드

AWS 서비스: Amazon EC2, Amazon EC2 Auto Scaling, Amazon EC2 컨테이너 레지스트리, Amazon EFS, Amazon RDS, Amazon VPC, Amazon ECS, Elastic Load Balancing (ELB), AWS Lambda

요약

이 패턴은 Network Load Balancer 위의 Amazon Elastic Container Service (Amazon ECS) 에서 도커 컨테이너 애플리케이션을 비공개로 호스팅하고 AWS를 사용하여 애플리케이션에 액세스하는 방법을 설명합니다. PrivateLink 그런 다음 프라이빗 네트워크를 사용하여 Amazon Web Services(AWS) 클라우드의 서비스에 안전하게 액세스할 수 있습니다. Amazon Relational Database Service(RDS)는 고가용성(HA)을 통해 Amazon ECS에서 실행되는 애플리케이션용 관계형 데이터베이스를 호스팅합니다. Amazon Elastic File System(Amazon EFS)은 애플리케이션에 영구적인 스토리지가 필요한 경우 사용됩니다.

프런트 엔드에 Network Load Balancer가 있는 Docker 애플리케이션을 실행하는 Amazon ECS 서비스를 가상 사설 클라우드 (VPC) 엔드포인트와 연결하여 AWS를 통해 액세스할 수 있습니다. PrivateLink 그리고 나서 이 VPC 엔드포인트 서비스의 VPC 엔드포인트를 사용하여 다른 VPC와 공유할 수 있습니다.

Amazon EC2 Auto Scaling 그룹 대신 [AWS Fargate](#)도 사용할 수 있습니다. 자세한 내용은 [AWS Fargate, PrivateLink AWS 및 Network Load Balancer를 사용하여 Amazon ECS에서 비공개로 컨테이너 애플리케이션에 액세스하기](#)를 참조하십시오.

사전 조건 및 제한 사항

사전 조건

- 활성 상태의 AWS 계정.
- [AWS Command Line Interface\(AWS CLI\) 버전 2](#), Linux, macOS, 또는 Windows에 설치 및 구성됨
- [Docker](#), Linux, macOS, 또는 Windows에 설치 및 구성됨
- Docker에서 실행되는 애플리케이션

아키텍처

기술 스택

- 아마존 CloudWatch
- Amazon Elastic Compute Cloud(Amazon EC2)
- Amazon EC2 Auto Scaling
- Amazon Elastic Container Registry (Amazon ECR)
- Amazon ECS
- Amazon RDS
- Amazon Simple Storage Service(S3)
- AWS Lambda
- AWS PrivateLink
- AWS Secrets Manager
- Application Load Balancer
- Network Load Balancer
- VPC

자동화 및 규모 조정

- [CloudFormation](#)을 사용하여 [AWS](#)를 사용하여 [코드형 인프라](#)를 사용하여 이 패턴을 생성할 수 있습니다.

도구

- [Amazon EC2](#) – Amazon Elastic Compute Cloud(Amazon EC2)는 AWS 클라우드에서 확장 가능한 컴퓨팅 용량을 제공합니다.
- [Amazon EC2 Auto Scaling](#) – Amazon EC2 Auto Scaling을 사용하면 애플리케이션의 로드를 처리할 수 있는 정확한 수의 Amazon EC2 인스턴스를 유지할 수 있습니다.
- [Amazon ECS](#) – Amazon Elastic Container Service(Amazon ECS)는 클러스터에서 컨테이너를 손쉽게 실행, 중지 및 관리할 수 있게 하는 컨테이너 관리 서비스로서 확장성과 속도가 뛰어납니다.
- [Amazon ECR](#) – Amazon Elastic Container Registry(Amazon ECR)는 안전하고 확장 가능하고 신뢰할 수 있는 AWS 관리형 컨테이너 이미지 레지스트리 서비스입니다.
- [Amazon EFS](#) – Amazon Elastic File System(Amazon EFS)은 AWS 클라우드 서비스 및 온프레미스 리소스와 함께 사용할 수 있는 간단하고 확장 가능하며 완전 관리형 탄력적인 NFS 파일 시스템을 제공합니다.
- [AWS Lambda](#) - Lambda는 서버를 프로비저닝하거나 관리하지 않고도 코드를 실행할 수 있게 해주는 컴퓨팅 서비스입니다.
- [Amazon RDS](#) - Amazon Relational Database Service(RDS)는 AWS 클라우드의 관계형 데이터베이스를 더 쉽게 설치, 운영 및 확장할 수 있게 하는 웹 서비스입니다.
- [Amazon S3](#) – Amazon Simple Storage Service(S3)는 인터넷에 대한 스토리지입니다. 이 서비스는 개발자가 더 쉽게 웹 규모 컴퓨팅 작업을 수행할 수 있도록 설계되었습니다.
- [AWS Secrets Manager](#) – Secrets Manager는 코드의 암호를 포함해 하드 코딩된 보안 인증 정보를 Secrets Manager에서 프로그래밍 방식으로 보안 암호를 검색하도록 하는 API 직접 호출로 바꿀 수 있습니다.
- [Amazon VPC](#) – Amazon Virtual Private Cloud(VPC)를 이용하면 사용자가 정의한 가상 네트워크로 AWS 리소스를 시작할 수 있습니다.
- [Elastic Load Balancing](#) – Elastic Load Balancing은 여러 가용 영역에 있는 Amazon EC2 인스턴스, 컨테이너, IP 주소와 같은 여러 대상에 걸쳐 수신되는 애플리케이션 또는 네트워크 트래픽을 분산합니다.
- [Docker](#) - Docker를 사용하면 개발자가 모든 애플리케이션을 가볍고 휴대가 간편하며 자급자족할 수 있는 컨테이너로 포장, 배송 및 실행할 수 있습니다.

에픽

네트워킹 구성 요소 생성

작업	설명	필요한 기술
VPC를 생성합니다.	<ol style="list-style-type: none"> 1. AWS Management Console에 로그인한 후 Amazon VPC 콘솔을 엽니다. VPC 생성을 선택하고 VPC 등을 선택합니다. 2. VPC의 이름을 입력하고 적절한 CIDR 블록 범위를 선택합니다. 3. 가용 영역 2개, 퍼블릭 서브넷 2개, 프라이빗 서브넷 4개를 지정합니다. 프라이빗 서브넷 2개는 Amazon ECS 작업용이고, 프라이빗 서브넷 2개는 Amazon RDS 데이터베이스용입니다. 4. 각 가용 영역에 NAT 게이트웨이 하나를 지정합니다. 5. VPC 생성을 선택합니다. 	클라우드 관리자

로드 밸런서 생성

작업	설명	필요한 기술
Network Load Balancer를 생성합니다.	<ol style="list-style-type: none"> 1. Amazon EC2 콘솔을 열고 사용자의 VPC가 포함된 AWS 리전을 선택합니다. 2. 로드 밸런싱에서 로드 밸런서를 선택하고 로드 밸런서 생성을 선택합니다. 	클라우드 관리자

작업	설명	필요한 기술
	<ol style="list-style-type: none"> 3. Network Load Balancer에 대해 생성을 선택합니다. 4. 로드 밸런서 구성 페이지에서 Network Load Balancer 및 리스너를 구성합니다. 중요: Network Load Balancer의 체계를 내부로 선택해야 합니다. 5. 적용되는 보안 설정을 선택하고 보안 그룹과 대상 그룹을 구성하세요. 라우팅 구성 섹션에서 대상 유형으로 인스턴스 또는 IP를 선택합니다. 대상을 등록하지 않았는지 확인하세요. 6. 모든 설정을 구성했으면 다음: 검토를 선택한 다음 생성을 선택합니다. 	

작업	설명	필요한 기술
Application Load Balancer을 생성합니다.	<ol style="list-style-type: none"> 1. Amazon EC2 콘솔에서 사용자 VPC가 포함된 동일한 리전을 선택합니다. 2. 로드 밸런싱에서 로드 밸런서를 선택하고 로드 밸런서 생성을 선택합니다. 3. Application Load Balancer를 선택하고 생성을 선택합니다. 4. Application Load Balancer와 리스너를 구성하십시오. 중요: Application Load Balancer의 체계를 내부로 선택해야 합니다. 5. 적용되는 보안 설정을 선택하고 보안 그룹과 대상 그룹을 구성하세요. 라우팅 구성 섹션에서 대상 유형으로 인스턴스 또는 IP를 선택합니다. 대상을 등록하지 않았는지 확인하세요. 6. 모든 설정을 구성했으면 다음: 검토를 선택한 다음 생성을 선택합니다. 	클라우드 관리자

Amazon EFS 파일 시스템 생성

작업	설명	필요한 기술
Amazon EFS 파일 시스템을 생성합니다.	<ol style="list-style-type: none"> 1. Amazon EFS 콘솔을 열고 파일 시스템 생성을 선택합니다. 	클라우드 관리자

작업	설명	필요한 기술
	<ol style="list-style-type: none"> 파일 시스템 생성 대화 상자에서 파일 시스템 이름을 입력하고 VPC를 선택합니다. 생성을 선택하여 파일 시스템을 생성합니다. Amazon EFS 파일 시스템을 설정하고 구성합니다. 	
서브넷의 대상을 마운트합니다.	<ol style="list-style-type: none"> Amazon EFS 콘솔로 돌아가서 파일 시스템을 선택합니다. 파일 시스템 페이지에 사용자 계정의 Amazon EFS 파일 시스템이 표시됩니다. 생성한 파일 시스템을 선택하고 관리를 선택하여 가용 영역을 표시합니다. 마운트 대상을 추가하려면 마운트 대상 추가를 선택하고 생성한 4개의 프라이빗 서브넷을 추가합니다. 	클라우드 관리자
서브넷이 대상으로 마운트되었는지 확인합니다.	<ol style="list-style-type: none"> Amazon EFS 콘솔에서 파일 시스템을 선택합니다. 네트워크를 선택하여 기존 마운트 대상 목록을 표시합니다. 여기에 생성한 4개의 서브넷이 포함되어 있는지 확인하세요. 	클라우드 관리자

S3 버킷 생성

작업	설명	필요한 기술
S3 버킷을 생성합니다.	필요한 경우 Amazon S3 콘솔을 열고 S3 버킷을 생성하여 애플리케이션의 고정 자산을 저장합니다.	클라우드 관리자

Secrets Manager 보안 암호 생성

작업	설명	필요한 기술
Secrets Manager의 보안 암호를 암호화하기 위해 AWS KMS 키를 생성합니다.	AWS Key Management Service(AWS KMS) 콘솔을 열고 KMS 키를 생성합니다.	클라우드 관리자
Amazon RDS 암호를 보관하기 위한 Secrets Manager 보안 암호를 생성합니다.	<ol style="list-style-type: none"> 1. AWS Secrets Manager 콘솔을 열고 새 암호 저장을 선택합니다. 2. 생성한 KMS 키를 선택하고 새 보안 암호를 저장합니다. 	클라우드 관리자

Amazon RDS 인스턴스 생성

작업	설명	필요한 기술
DB 서브넷 그룹을 생성합니다.	<ol style="list-style-type: none"> 1. Amazon RDS 콘솔을 열고 서브넷 그룹을 선택합니다. 2. DB 서브넷 그룹 생성을 선택하고 DB 서브넷 그룹의 이름과 설명을 입력합니다. 3. 이전에 생성한 VPC를 선택한 다음, 가용 영역과 서브넷 	클라우드 관리자

작업	설명	필요한 기술
	을 선택합니다. 그런 다음 생성을 선택합니다.	
Amazon RDS 인스턴스를 생성합니다.	프라이빗 서브넷 내에서 Amazon RDS 인스턴스를 생성하고 구성합니다.고가용성(HA)을 위해 다중 AZ가 켜져 있는지 확인하세요.	클라우드 관리자
Amazon RDS 인스턴스로 데이터를 로드합니다.	애플리케이션에 필요한 관계형 데이터를 Amazon RDS 인스턴스로 로드합니다. 이 프로세스는 애플리케이션의 요구 사항과 데이터베이스 체계의 정의 및 설계 방식에 따라 달라집니다.	클라우드 관리자, DBA

Amazon ECS 구성 요소 생성

작업	설명	필요한 기술
ECS 클러스터를 생성합니다.	<ol style="list-style-type: none"> Amazon ECS 콘솔을 열고 클러스터를 선택합니다. 클러스터 생성을 선택하고 필요한 사양에 따라 ECS 클러스터를 설정합니다. 	클라우드 관리자
도커 이미지를 생성합니다.	관련 리소스 섹션의 지침에 따라 도커 이미지를 생성합니다.	클라우드 관리자
Amazon ECR 리포지토리를 생성합니다.	<ol style="list-style-type: none"> Amazon ECR 콘솔에서 리포지토리를 선택합니다. 리포지토리 생성을 선택하고 리포지토리의 고유한 이름을 입력합니다. 	클라우드 관리자, DevOps 엔지니어

작업	설명	필요한 기술
	<p>3. 필요한 경우 AWS KMS 암호화를 포함하여 사용자 사양에 따라 리포지토리를 구성합니다.</p>	
<p>Docker 클라이언트를 Amazon ECR 리포지토리를 인증합니다.</p>	<p>Amazon ECR 리포지토리의 Docker 클라이언트를 인증하려면 AWS CLI에서 <code>aws ecr get-login-password</code> 명령을 실행하세요.</p>	<p>클라우드 관리자</p>
<p>Amazon ECR 리포지토리에 도커 이미지를 푸시합니다.</p>	<ol style="list-style-type: none"> 1. 푸시하려는 도커 이미지를 식별하고 AWS CLI에서 <code>docker images</code> 명령을 실행합니다. 2. Amazon ECR 레지스트리, 리포지토리 및 선택적 이미지 태그 이름 조합을 사용하여 이미지에 태그를 지정합니다. 3. <code>docker push</code> 명령을 실행하여 도커 이미지를 푸시합니다. 4. 필요한 모든 이미지에 위 단계를 반복합니다. 	<p>클라우드 관리자</p>

작업	설명	필요한 기술
<p>Amazon ECS 작업 정의를 생성합니다.</p>	<p>Amazon ECS에서 Docker 컨테이너를 실행하려면 작업 정의가 필요합니다.</p> <ol style="list-style-type: none"> 1. Amazon ECS 콘솔로 돌아가서 작업 정의를 선택한 다음 새 작업 정의 생성을 선택합니다. 2. 호환성 선택 페이지에서 해당 작업이 사용해야 하는 시작 유형을 선택하고 다음 단계를 선택합니다. <p>작업 정의를 설정하는 데 도움이 필요하면 관련 리소스 섹션의 '작업 정의 생성'을 참조하세요. 중요: Amazon ECR에 푸시한 도커 이미지를 제공해야 합니다.</p>	<p>클라우드 관리자</p>
<p>Amazon ECS 서비스를 생성합니다.</p>	<p>이전에 생성한 ECS 클러스터를 사용하여 Amazon ECS 서비스를 생성합니다. Amazon EC2를 시작 유형으로 선택하고, 이전 단계에서 생성한 작업 정의와 Application Load Balancer의 대상 그룹을 선택해야 합니다.</p>	<p>클라우드 관리자</p>

Amazon EC2 Auto Scaling 그룹을 생성

작업	설명	필요한 기술
시작 구성을 생성합니다.	Amazon EC2 콘솔을 열고 시작 구성을 생성합니다. 사용자 데이터에 EC2 인스턴스가 원하는 ECS 클러스터에 조인하도록 허용하는 코드가 있는지 확인하세요. 필요한 코드의 예는 관련 리소스 섹션을 참조하세요.	클라우드 관리자
Amazon EC2 Auto Scaling 그룹을 생성합니다.	Amazon EC2 콘솔로 돌아가서 Auto Scaling에서 오토 스케일링을 선택합니다. Amazon EC2 Auto Scaling 그룹을 설정합니다. 프라이빗 서브넷을 선택하고 이전에 생성한 구성을 시작했는지 확인하세요.	클라우드 관리자

AWS 설정 PrivateLink

작업	설명	필요한 기술
AWS PrivateLink 엔드포인트를 설정합니다.	<ol style="list-style-type: none"> Amazon VPC 콘솔에서 AWS PrivateLink 엔드포인트를 생성합니다. 이 엔드포인트를 Network Load Balancer와 연결하면 Amazon ECS에 호스팅된 애플리케이션을 고객이 비공개로 사용할 수 있습니다. <p>자세한 내용은 관련 리소스 섹션을 참조하세요.</p>	클라우드 관리자

VPC 엔드포인트 생성

작업	설명	필요한 기술
VPC 엔드포인트를 생성합니다.	이전에 생성한 AWS 엔드포인트에 대한 VPC PrivateLink 엔드포인트를 생성합니다. VPC 엔드포인트 FQDN (정규화된 도메인 이름)은 AWS 엔드포인트 FQDN을 가리킵니다. PrivateLink 그러면 DNS 엔드포인트가 액세스할 수 있는 VPC 엔드포인트 서비스에 대한 탄력적 네트워크 인터페이스가 생성됩니다.	클라우드 관리자

Lambda 함수 생성

작업	설명	필요한 기술
Lambda 함수를 생성합니다.	AWS Lambda 콘솔에서 Lambda 함수를 생성하여 Application Load Balancer IP 주소를 Network Load Balancer의 대상으로 업데이트하세요. 이에 대한 자세한 내용은 관련 리소스 섹션의 'Application Load Balancer에 고정 IP 주소 사용' 블로그 게시물을 참조하세요.	앱 개발자

관련 리소스

로드 밸런서 생성:

- [Network Load Balancer 생성](#)

- [Application Load Balancer 생성](#)

Amazon EFS 파일 시스템 생성:

- [Amazon EFS 파일 시스템 생성](#)
- [Amazon EFS에서 마운트 대상 생성](#)

S3 버킷 생성:

- [S3 버킷 생성](#)

Secrets Manager 보안 암호 생성:

- [AWS KMS에서 키 생성](#)
- [AWS Secrets Manager에서 보안 암호 생성](#)

Amazon RDS 인스턴스 생성:

- [Amazon RDS DB 인스턴스 생성](#)

Amazon ECS 구성 요소 생성:

- [Amazon ECS 클러스터를 생성](#)
- [도커 이미지를 생성](#)
- [Amazon ECR 리포지토리를 생성](#)
- [Amazon ECR 리포지토리로 Docker 인증](#)
- [Amazon ECR 리포지토리에 이미지를 푸시](#)
- [Amazon ECS 작업 정의 생성](#)
- [Amazon ECS 서비스를 생성](#)

Amazon EC2 Auto Scaling 그룹을 생성:

- [시작 구성을 생성](#)
- [시작 구성을 사용하여 Auto Scaling 그룹 생성](#)

- [Amazon EC2 사용자 데이터를 사용하여 컨테이너 인스턴스 부트스트래핑](#)

AWS 설정 PrivateLink:

- [VPC 엔드포인트 서비스 \(AWS\) PrivateLink](#)

VPC 엔드포인트 생성:

- [인터페이스 VPC 엔드포인트 \(AWS\) PrivateLink](#)

Lambda 함수 생성:

- [Lambda 함수 생성](#)

기타 리소스:

- [Application Load Balancer에 고정 IP 주소 사용](#)
- [AWS를 통한 안전한 서비스 액세스 PrivateLink](#)

AWS Fargate, PrivateLink AWS 및 Network Load Balancer를 사용하여 Amazon ECS에서 비공개로 컨테이너 애플리케이션에 액세스할 수 있습니다.

작성자: Kirankumar Chandrashekar(AWS)

환경: 프로덕션

기술: 컨테이너 및 마이크로서비스, 네트워킹, 보안, ID, 규정 준수, 웹 및 모바일 앱

워크로드: 기타 모든 워크로드

AWS 서비스: Amazon EC2 Container Registry, Amazon ECS, Amazon EFS, Amazon RDS, Amazon VPC, Elastic Load Balancing(ELB), AWS Lambda

요약

이 패턴은 Network Load Balancer 뒤에서 AWS Fargate 시작 유형의 Amazon Elastic Container Service (Amazon ECS) 를 사용하여 Amazon Web Services (AWS) 클라우드에서 Docker 컨테이너 애플리케이션을 비공개로 호스팅하고 AWS를 사용하여 애플리케이션에 액세스하는 방법을 설명합니다. PrivateLink Amazon Relational Database Service(RDS)는 고가용성(HA)을 통해 Amazon ECS에서 실행되는 애플리케이션용 관계형 데이터베이스를 호스팅합니다. 애플리케이션에 영구적인 스토리지가 필요한 경우 Amazon Elastic File System(Amazon EFS)을 사용할 수 있습니다.

이 패턴은 Docker 애플리케이션을 실행하는 Amazon ECS 서비스에 [Fargate 시작 유형](#)을 사용하며 프런트 엔드에 Network Load Balancer가 있습니다. 그런 다음 AWS를 통해 액세스할 수 있도록 가상 사설 클라우드 (VPC) 엔드포인트와 연결할 수 있습니다. PrivateLink 그리고 나서 이 VPC 엔드포인트 서비스의 VPC 엔드포인트를 사용하여 다른 VPC와 공유할 수 있습니다.

Amazon ECS와 함께 Fargate를 사용하면 Amazon Elastic Compute Cloud(Amazon EC2) 인스턴스의 서버나 클러스터를 관리할 필요 없이 컨테이너를 실행할 수 있습니다. Fargate 대신 Amazon EC2 Auto Scaling 그룹도 사용할 수 있습니다. 자세한 내용은 [AWS PrivateLink 및 Network Load Balancer를 사용하여 Amazon ECS에서 비공개로 컨테이너 애플리케이션에 액세스하기](#)를 참조하십시오.

사전 조건 및 제한 사항

사전 조건

- 활성 상태의 AWS 계정.
- [AWS Command Line Interface\(AWS CLI\) 버전 2](#), Linux, macOS, 또는 Windows에 설치 및 구성됨
- [Docker](#), Linux, macOS, 또는 Windows에 설치 및 구성됨
- Docker에서 실행되는 애플리케이션

아키텍처

기술 스택

- 아마존 CloudWatch
- Amazon Elastic Container Registry (Amazon ECR)
- Amazon ECS
- Amazon EFS
- Amazon RDS
- Amazon Simple Storage Service(S3)
- AWS Fargate
- AWS Lambda
- AWS PrivateLink
- AWS Secrets Manager
- Application Load Balancer
- Network Load Balancer
- VPC

자동화 및 규모 조정

- [CloudFormationAWS](#)를 사용하여 [코드형 인프라](#)를 사용하여 이 패턴을 생성할 수 있습니다.

도구

- [Amazon ECS](#) – Amazon Elastic Container Service(Amazon ECS)는 클러스터에서 컨테이너를 손쉽게 실행, 중지 및 관리할 수 있게 하는 컨테이너 관리 서비스로서 확장성과 속도가 뛰어납니다.
- [Amazon ECR](#) – Amazon Elastic Container Registry(Amazon ECR)는 안전하고 확장 가능하고 신뢰할 수 있는 AWS 관리형 컨테이너 이미지 레지스트리 서비스입니다.
- [Amazon EFS](#) – Amazon Elastic File System(Amazon EFS)은 AWS 클라우드 서비스 및 온프레미스 리소스와 함께 사용할 수 있는 간단하고 확장 가능하며 완전 관리형 탄력적인 NFS 파일 시스템을 제공합니다.
- [AWS Fargate](#) – AWS Fargate는 Amazon EC2 인스턴스의 서버나 클러스터를 관리할 필요 없이 컨테이너를 실행하기 위해 Amazon ECS에 사용할 수 있는 기술입니다.
- [AWS Lambda](#) - Lambda는 서버를 프로비저닝하거나 관리하지 않고도 코드를 실행할 수 있게 해주는 컴퓨팅 서비스입니다.
- [Amazon RDS](#) - Amazon Relational Database Service(RDS)는 AWS 클라우드의 관계형 데이터베이스를 더 쉽게 설치, 운영 및 확장할 수 있게 하는 웹 서비스입니다.
- [Amazon S3](#) – Amazon Simple Storage Service(S3)는 인터넷에 대한 스토리지입니다. 이 서비스는 개발자가 더 쉽게 웹 규모 컴퓨팅 작업을 수행할 수 있도록 설계되었습니다.
- [AWS Secrets Manager](#) – Secrets Manager는 코드의 암호를 포함해 하드 코딩된 보안 인증 정보를 Secrets Manager에서 프로그래밍 방식으로 보안 암호를 검색하도록 하는 API 직접 호출로 바꿀 수 있습니다.
- [Amazon VPC](#) – Amazon Virtual Private Cloud(VPC)를 이용하면 사용자가 정의한 가상 네트워크로 AWS 리소스를 시작할 수 있습니다.
- [Elastic Load Balancing](#) – Elastic Load Balancing(ELB)은 여러 가용 영역에 있는 EC2 인스턴스, 컨테이너, IP 주소와 같은 여러 대상에 걸쳐 수신되는 애플리케이션 또는 네트워크 트래픽을 분산합니다.
- [Docker](#) - Docker를 사용하면 개발자가 모든 애플리케이션을 가볍고 휴대가 간편하며 자급자족할 수 있는 컨테이너로 포장, 배송 및 실행할 수 있습니다.

에픽

네트워킹 구성 요소 생성

작업	설명	필요한 기술
VPC를 생성합니다.	<ol style="list-style-type: none"> 1. AWS Management Console에 로그인하고 Amazon VPC 콘솔을 엽니다. VPC 생성을 선택하고 VPC 등을 선택합니다. 2. VPC의 이름을 입력하고 적절한 CIDR 블록 범위를 선택합니다. 3. 가용 영역 2개, 퍼블릭 서브넷 2개, 프라이빗 서브넷 4개를 지정합니다. 프라이빗 서브넷 2개는 Amazon ECS 작업용이고, 프라이빗 서브넷 2개는 Amazon RDS 데이터베이스용입니다. 4. 각 가용 영역에 NAT 게이트웨이 하나를 지정합니다. 5. VPC 생성을 선택합니다. 	클라우드 관리자

로드 밸런서 생성

작업	설명	필요한 기술
Network Load Balancer를 생성합니다.	<ol style="list-style-type: none"> 1. Amazon EC2 콘솔을 열고 사용자의 VPC가 포함된 AWS 리전을 선택합니다. 2. 로드 밸런싱에서 로드 밸런서를 선택하고 로드 밸런서 생성을 선택합니다. 	클라우드 관리자

작업	설명	필요한 기술
	<ol style="list-style-type: none"> 3. Network Load Balancer에 대해 생성을 선택합니다. 4. 로드 밸런서 구성 페이지에서 Network Load Balancer 및 리스너를 구성합니다. 중요: Network Load Balancer의 체계를 내부로 선택해야 합니다. 5. 적용되는 보안 설정을 선택하고 보안 그룹과 대상 그룹을 구성하세요. 라우팅 구성 섹션에서 대상 유형으로 IP를 선택합니다. 대상을 등록하지 않았는지 확인하세요. 6. 모든 설정을 구성했으면 다음: 검토를 선택한 다음 생성을 선택합니다. <p>이 이야기와 다른 이야기에 대한 도움이 필요하면 관련 리소스 섹션을 참조하십시오.</p>	

작업	설명	필요한 기술
Application Load Balancer을 생성합니다.	<ol style="list-style-type: none"> 1. Amazon EC2 콘솔에서 사용자 VPC가 포함된 동일한 리전을 선택합니다. 2. 로드 밸런싱에서 로드 밸런서를 선택하고 로드 밸런서 생성을 선택합니다. 3. Application Load Balancer를 선택하고 생성을 선택합니다. 4. Application Load Balancer 및 그에 대한 리스너를 구성합니다. 중요: Application Load Balancer의 체계를 내부로 선택해야 합니다. 5. 적용되는 보안 설정을 선택하고 보안 그룹과 대상 그룹을 구성하세요. 라우팅 구성 섹션에서 대상 유형으로 IP를 선택합니다. 대상을 등록하지 않았는지 확인하세요. 6. 모든 설정을 구성했으면 다음: 검토를 선택한 다음 생성을 선택합니다. 	클라우드 관리자

Amazon EFS 파일 시스템 생성

작업	설명	필요한 기술
Amazon EFS 파일 시스템을 생성합니다.	<ol style="list-style-type: none"> 1. Amazon EFS 콘솔을 열고 파일 시스템 생성을 선택합니다. 	클라우드 관리자

작업	설명	필요한 기술
	<ol style="list-style-type: none"> 2. 파일 시스템 생성 대화 상자에서 파일 시스템 이름을 입력하고 VPC를 선택합니다. 3. 생성을 선택하여 파일 시스템을 생성합니다. 4. Amazon EFS 파일 시스템을 설정하고 구성합니다. 	
서브넷의 대상을 마운트합니다.	<ol style="list-style-type: none"> 1. Amazon EFS 콘솔로 돌아가서 파일 시스템을 선택합니다. 파일 시스템 페이지에 사용자 계정의 Amazon EFS 파일 시스템이 표시됩니다. 2. 생성한 파일 시스템을 선택하고 관리를 선택하여 가용 영역을 표시합니다. 3. 마운트 대상을 추가하려면 마운트 대상 추가를 선택하고 생성한 4개의 프라이빗 서브넷을 추가합니다. 	클라우드 관리자
서브넷이 대상으로 마운트되었는지 확인합니다.	<ol style="list-style-type: none"> 1. Amazon EFS 콘솔에서 파일 시스템을 선택합니다. 2. 네트워크를 선택하여 기존 마운트 대상 목록을 표시합니다. 여기에 생성한 4개의 서브넷이 포함되어 있는지 확인하세요. 	클라우드 관리자

S3 버킷 생성

작업	설명	필요한 기술
S3 버킷을 생성합니다.	필요한 경우 Amazon S3 콘솔을 열고 S3 버킷을 생성하여 애플리케이션의 고정 자산을 저장합니다.	클라우드 관리자

Secrets Manager 보안 암호 생성

작업	설명	필요한 기술
Secrets Manager의 보안 암호를 암호화하기 위해 AWS KMS 키를 생성합니다.	AWS Key Management Service(AWS KMS) 콘솔을 열고 KMS 키를 생성합니다.	클라우드 관리자
Amazon RDS 암호를 보관하기 위한 Secrets Manager 보안 암호를 생성합니다.	<ol style="list-style-type: none"> 1. AWS Secrets Manager 콘솔을 열고 새 암호 저장을 선택합니다. 2. 생성한 KMS 키를 선택하고 새 보안 암호를 저장합니다. 	클라우드 관리자

Amazon RDS 인스턴스 생성

작업	설명	필요한 기술
DB 서브넷 그룹을 생성합니다.	<ol style="list-style-type: none"> 1. Amazon RDS 콘솔을 열고 서브넷 그룹을 선택합니다. 2. DB 서브넷 그룹 생성을 선택하고 DB 서브넷 그룹의 이름과 설명을 입력합니다. 3. 이전에 생성한 VPC를 선택한 다음, 가용 영역과 서브넷 	클라우드 관리자

작업	설명	필요한 기술
	을 선택합니다. 그런 다음 생성을 선택합니다.	
Amazon RDS 인스턴스를 생성합니다.	프라이빗 서브넷 내에서 Amazon RDS 인스턴스를 생성하고 구성합니다.고가용성(HA)을 위해 다중 AZ가 켜져 있는지 확인하세요.	클라우드 관리자
Amazon RDS 인스턴스로 데이터를 로드합니다.	애플리케이션에 필요한 관계형 데이터를 Amazon RDS 인스턴스로 로드합니다. 이 프로세스는 애플리케이션의 요구 사항과 데이터베이스 체계의 정의 및 설계 방식에 따라 달라집니다.	DBA

Amazon ECS 구성 요소 생성

작업	설명	필요한 기술
ECS 클러스터를 생성합니다.	<ol style="list-style-type: none"> Amazon ECS 콘솔을 열고 클러스터를 선택합니다. 클러스터 생성을 선택하고 필요한 사양에 따라 ECS 클러스터를 설정합니다. 	클라우드 관리자
도커 이미지를 생성합니다.	관련 리소스 섹션의 지침에 따라 도커 이미지를 생성합니다.	클라우드 관리자
Amazon ECR 리포지토리를 생성합니다.	<ol style="list-style-type: none"> Amazon ECR 콘솔을 열고 리포지토리를 선택합니다. 리포지토리 생성을 선택하고 리포지토리의 고유한 이름을 입력합니다. 	클라우드 관리자, DevOps 엔지니어

작업	설명	필요한 기술
	<p>3. 필요한 경우 AWS KMS 암호화를 포함하여 사용자 사양에 따라 리포지토리를 구성합니다.</p>	
<p>Amazon ECR 리포지토리에 도커 이미지를 푸시합니다.</p>	<ol style="list-style-type: none"> 1. 푸시하려는 도커 이미지를 식별하고 AWS CLI에서 <code>docker images</code> 명령을 실행합니다. 2. Amazon ECR 레지스트리, 리포지토리 및 선택적 이미지 태그 이름 조합을 사용하여 이미지에 태그를 지정합니다. 3. <code>docker push</code> 명령을 실행하여 도커 이미지를 푸시합니다. 4. 필요한 모든 이미지에 위 단계를 반복합니다. 	<p>클라우드 관리자</p>

작업	설명	필요한 기술
<p>Amazon ECS 작업 정의를 생성합니다.</p>	<p>Amazon ECS에서 Docker 컨테이너를 실행하려면 작업 정의가 필요합니다.</p> <ol style="list-style-type: none"> 1. Amazon ECS 콘솔로 돌아가서 작업 정의를 선택한 다음 새 작업 정의 생성을 선택합니다. 2. 호환성 선택 페이지에서 해당 작업이 사용해야 하는 시작 유형을 선택하고 다음 단계를 선택합니다. <p>작업 정의를 설정하는 데 도움이 필요하면 관련 리소스 섹션의 '작업 정의 생성'을 참조하세요. 중요: Amazon ECR에 푸시한 도커 이미지를 제공하는지 확인합니다.</p>	<p>클라우드 관리자</p>
<p>ECS 서비스를 생성하고 Fargate를 시작 유형으로 선택합니다.</p>	<ol style="list-style-type: none"> 1. 이전에 생성한 ECS 클러스터를 사용하여 Amazon ECS 서비스를 생성합니다. Fargate를 시작 유형으로 선택했는지 확인하세요. 2. 이전 단계에서 생성한 작업 정의를 선택하고 Application Load Balancer의 대상 그룹을 선택합니다. 	<p>클라우드 관리자</p>

AWS 설정 PrivateLink

작업	설명	필요한 기술
AWS PrivateLink 엔드포인트를 설정합니다.	<ol style="list-style-type: none"> 1. Amazon VPC 콘솔을 열고 AWS PrivateLink 엔드포인트를 생성합니다. 2. 이 엔드포인트를 Network Load Balancer와 연결하면 Amazon ECS에 호스팅된 애플리케이션을 고객이 비공개로 사용할 수 있습니다. <p>자세한 내용은 관련 리소스 섹션을 참조하세요.</p>	클라우드 관리자

VPC 엔드포인트 생성

작업	설명	필요한 기술
VPC 엔드포인트를 생성합니다.	<p>이전에 생성한 AWS 엔드포인트에 대한 VPC PrivateLink 엔드포인트를 생성합니다. VPC 엔드포인트 FQDN (정규화된 도메인 이름)은 AWS 엔드포인트 FQDN을 가리킵니다. PrivateLink 그러면 도메인 이름 서비스 엔드포인트가 액세스할 수 있는 VPC 엔드포인트 서비스에 대한 탄력적인 네트워크 인터페이스가 생성됩니다.</p>	클라우드 관리자

Lambda 함수 생성

작업	설명	필요한 기술
Lambda 함수를 생성합니다.	Lambda 콘솔을 열고 Application Load Balancer IP 주소를 Network Load Balancer의 대상으로 업데이트하는 Lambda 함수를 생성합니다. 이에 대한 자세한 내용은 관련 리소스 섹션의 'Application Load Balancer에 고정 IP 주소 사용' 블로그 게시물을 참조하세요.	앱 개발자

관련 리소스

로드 밸런서 생성:

- [Network Load Balancer 생성](#)
- [Application Load Balancer 생성](#)

Amazon EFS 파일 시스템 생성:

- [Amazon EFS 파일 시스템 생성](#)
- [Amazon EFS에서 마운트 대상 생성](#)

S3 버킷 생성:

- [S3 버킷 생성](#)

Secrets Manager 보안 암호 생성:

- [AWS KMS에서 키 생성](#)
- [AWS Secrets Manager에서 보안 암호 생성](#)

Amazon RDS 인스턴스 생성:

- [Amazon RDS DB 인스턴스 생성](#)

Amazon ECS 구성 요소 생성:

- [Amazon ECS 클러스터를 생성](#)
- [도커 이미지를 생성](#)
- [Amazon ECR 리포지토리를 생성](#)
- [Amazon ECR 리포지토리로 Docker 인증](#)
- [Amazon ECR 리포지토리에 이미지를 푸시](#)
- [Amazon ECS 작업 정의 생성](#)
- [Amazon ECS 서비스를 생성](#)

AWS 설정 PrivateLink:

- [VPC 엔드포인트 서비스 \(AWS\) PrivateLink](#)

VPC 엔드포인트 생성:

- [인터페이스 VPC 엔드포인트 \(AWS\) PrivateLink](#)

Lambda 함수 생성:

- [Lambda 함수 생성](#)

기타 리소스:

- [Application Load Balancer에 고정 IP 주소 사용](#)
- [AWS를 통한 안전한 서비스 액세스 PrivateLink](#)

PrivateLink AWS와 Network Load Balancer를 사용하여 Amazon EKS에서 비공개로 컨테이너 애플리케이션에 액세스합니다.

작성자: Kirankumar Chandrashekar(AWS)

환경: 프로덕션

기술: 컨테이너 및 마이크로서
비스, 현대화, 보안, ID, 규정
DevOps 준수

워크로드: 기타 모든 워크로드

AWS 서비스: Amazon EKS,
Amazon VPC

요약

이 패턴은 Network Load Balancer 뒤에서 Amazon Elastic Kubernetes Service (Amazon EKS) 에서 도커 컨테이너 애플리케이션을 비공개로 호스팅하고 AWS를 사용하여 애플리케이션에 액세스하는 방법을 설명합니다. PrivateLink 그런 다음 프라이빗 네트워크를 사용하여 Amazon Web Services(AWS) 클라우드의 서비스에 안전하게 액세스할 수 있습니다.

프런트 엔드에 Network Load Balancer가 있는 Docker 애플리케이션을 실행하는 Amazon EKS 클러스터를 가상 사설 클라우드 (VPC) 엔드포인트와 연결하여 AWS를 통해 액세스할 수 있습니다. PrivateLink 그리고 나서 이 VPC 엔드포인트 서비스의 VPC 엔드포인트를 사용하여 다른 VPC와 공유할 수 있습니다.

이 패턴에서 설명하는 설정은 VPC와 AWS 계정 간에 애플리케이션 액세스를 공유하는 안전한 방법입니다. 소비자와 공급자 계정 간의 연결은 글로벌 AWS 백본에 있고 퍼블릭 인터넷을 통과하지 않기 때문에 특별한 연결 또는 라우팅 구성이 필요하지 않습니다.

사전 조건 및 제한 사항

사전 조건

- [Docker](#), Linux, macOS 또는 Windows에 설치 및 구성됨.
- Docker에서 실행되는 애플리케이션.
- 활성 상태의 AWS 계정
- [AWS Command Line Interface\(AWS CLI\) 버전 2](#), Linux, macOS 또는 Windows에 설치 및 구성됨.

- 태그가 지정된 프라이빗 서브넷이 있고 애플리케이션을 호스팅하도록 구성된 기존 Amazon EKS 클러스터. 자세한 내용은 Amazon EKS 설명서의 [서브넷 태깅](#)을 참조하십시오.
- Kubectl, Amazon EKS 클러스터의 리소스에 액세스하도록 설치 및 구성됨. 자세한 내용은 Amazon EKS 설명서의 [kubectl 설치](#)를 참조하세요.

아키텍처

기술 스택

- Amazon EKS
- AWS PrivateLink
- Network Load Balancer

자동화 및 규모 조정

- 쿠버네티스 매니페스트는 Git 기반 리포지토리 (예: AWS) 에서 추적 및 관리하고 AWS의 지속적 통합 및 지속적 전달 (CI/CD CodeCommit) 을 사용하여 배포할 수 있습니다. CodePipeline
- CloudFormation AWS를 사용하여 코드형 인프라 (IaC) 를 사용하여 이 패턴을 생성할 수 있습니다.

도구

- [AWS CLI](#) – AWS Command Line Interface(AWS CLI)는 명령줄 셸에서 명령을 사용하여 AWS 서비스와 상호 작용할 수 있는 오픈 소스 도구입니다.
- [Elastic Load Balancing](#) - Elastic Load Balancing은 하나 이상의 가용 영역에서 Amazon Elastic Compute Cloud(Amazon EC2) 인스턴스, 컨테이너, IP 주소 등 여러 대상에 걸쳐 수신되는 애플리케이션 또는 네트워크 트래픽을 분산합니다.
- [Amazon EKS](#) – Amazon Elastic Kubernetes Service(Amazon EKS)는 Kubernetes 컨트롤 플레인 또는 노드를 설치, 작동 및 유지 관리할 필요 없이 AWS에서 Kubernetes를 실행하는 데 사용할 수 있는 관리형 서비스입니다.
- [Amazon VPC](#) – Amazon Virtual Private Cloud(VPC)를 이용하면 사용자가 정의한 가상 네트워크로 AWS 리소스를 시작할 수 있습니다.

- [Kubectl](#) – Kubectl은 Kubernetes 클러스터에 대해 명령을 실행하기 위한 명령줄 유틸리티입니다.

에픽

Kubernetes 배포 및 서비스 매니페스트 파일 배포

작업	설명	필요한 기술
<p>Kubernetes 배포 매니페스트 파일을 생성합니다.</p>	<p>필요에 따라 다음 샘플 파일을 수정하여 배포 매니페스트 파일을 생성하세요.</p> <pre data-bbox="594 709 1026 1663"> apiVersion: apps/v1 kind: Deployment metadata: name: sample-app spec: replicas: 3 selector: matchLabels: app: nginx template: metadata: labels: app: nginx spec: containers: - name: nginx image: public.ecr.aws/z9d2n7e1/nginx:1.19.5 ports: - name: http container Port: 80 </pre> <p>참고: 이것은 NGINX 도커 이미지를 사용하여 배포되는 NGINX 샘플 구성 파일입니다. 자세한 내용은 Docker 설명서</p>	<p>DevOps 엔지니어</p>

작업	설명	필요한 기술
	<p>의 공식 NGINX 도커 이미지를 사용하는 방법을 참조하세요.</p>	
Kubernetes 배포 매니페스트 파일을 배포합니다.	<p>다음 명령을 실행하여 Amazon EKS 클러스터에 배포 매니페스트 파일을 적용합니다.</p> <pre>kubectl apply -f <your_deployment_f ile_name></pre>	DevOps 엔지니어

작업	설명	필요한 기술
<p>Kubernetes 서비스 매니페스트 파일을 생성합니다.</p>	<p>필요에 따라 다음 샘플 파일을 수정하여 서비스 매니페스트 파일을 생성하세요.</p> <pre data-bbox="594 394 1026 1230"> apiVersion: v1 kind: Service metadata: name: sample-service annotations: service.beta.kubernetes.io/aws-load-balancer-type: nlb service.beta.kubernetes.io/aws-load-balancer-internal: "true" spec: ports: - port: 80 targetPort: 80 protocol: TCP type: LoadBalancer selector: app: nginx </pre> <p>중요: 내부 Network Load Balancer를 정의할 때 다음 annotations 을(를) 포함했는지 확인하세요.</p> <pre data-bbox="594 1482 1026 1801"> service.beta.kubernetes.io/aws-load-balancer-type: nlb service.beta.kubernetes.io/aws-load-balancer-internal: "true" </pre>	<p>DevOps 엔지니어</p>

작업	설명	필요한 기술
Kubernetes 서비스 매니페스트 파일을 배포합니다.	<p>다음 명령을 실행하여 Amazon EKS 클러스터에 서비스 매니페스트 파일을 적용합니다.</p> <pre>kubectl apply -f <your_service_file_name></pre>	DevOps 엔지니어

엔드포인트 생성

작업	설명	필요한 기술
Network Load Balancer의 이름을 기록해 둡니다.	<p>다음 명령을 실행하여 Network Load Balancer 이름을 검색합니다.</p> <pre>kubectl get svc sample-service -o wide</pre> <p>AWS PrivateLink 엔드포인트를 생성하는 데 필요한 네트워크 로드 밸런서의 이름을 기록해 둡니다.</p>	DevOps 엔지니어
AWS PrivateLink 엔드포인트를 생성합니다.	<p>AWS 관리 콘솔에 로그인하고 Amazon VPC 콘솔을 연 다음 AWS PrivateLink 엔드포인트를 생성합니다. 이 엔드포인트를 Network Load Balancer와 연결하면 고객이 애플리케이션을 비공개로 사용할 수 있습니다. 자세한 내용은 Amazon VPC 설명서의 VPC 엔드포인트</p>	클라우드 관리자

작업	설명	필요한 기술
	<p>트 서비스 PrivateLink (AWS)를 참조하십시오.</p> <p>중요: 소비자 계정에 애플리케이션 액세스가 필요한 경우, 소비자 계정의 AWS 계정 ID를 AWS PrivateLink 엔드포인트 구성의 허용된 보안 주체 목록에 추가해야 합니다. 자세한 내용은 Amazon VPC 설명서의 엔드포인트 서비스 권한 추가 및 권한 제거를 참조하세요.</p>	
<p>VPC 엔드포인트를 생성합니다.</p>	<p>Amazon VPC 콘솔에서 엔드포인트 서비스를 선택하고 엔드포인트 서비스 생성을 선택합니다. AWS PrivateLink 엔드포인트에 대한 VPC 엔드포인트를 생성합니다.</p> <p>VPC 엔드포인트의 FQDN (정규화된 도메인 이름)은 AWS 엔드포인트의 FQDN을 가리킵니다. PrivateLink 그러면 DNS 엔드포인트가 액세스할 수 있는 VPC 엔드포인트 서비스에 대한 탄력적 네트워크 인터페이스가 생성됩니다.</p>	<p>클라우드 관리자</p>

관련 리소스

- [공식 NGINX 도커 이미지 사용](#)
- [Amazon EKS의 네트워크 로드 밸런싱](#)
- [VPC 엔드포인트 서비스 \(AWS\) 생성 PrivateLink](#)

- [엔드포인트 서비스에 대한 권한 추가 및 제거](#)

Amazon EKS의 AWS 프라이빗 CA를 사용하여 AWS App Mesh에서 MTL을 활성화합니다.

작성자: Omar Kahil(AWS), Emmanuel Saliu(AWS), Muhammad Shahzad(AWS)

환경: PoC 또는 파일럿	기술: 컨테이너 및 마이크로서비스	AWS 서비스: AWS App Mesh, Amazon EKS, AWS Certificate Manager(ACM)
----------------	--------------------	---

요약

이 패턴은 AWS App Mesh에서 AWS 사설 인증 기관(AWS 사설 CA)의 인증서를 사용하여 Amazon Web Services(AWS)에서 상호 전송 계층 보안(MTL)을 구현하는 방법을 보여줍니다. 모두를 위한 보안 프로덕션 ID 프레임워크(SPIFE)를 통해 Envoy 비밀 검색 서비스(SDS) API를 사용합니다. SPIFE는 세분화되고 동적인 워크로드 ID 관리를 제공하는 광범위한 커뮤니티 지원이 포함된 클라우드 네이티브 컴퓨팅 재단(CNCF) 오픈 소스 프로젝트입니다. SPIFE 표준을 구현하려면 SPIRE SPIFE 런타임 환경을 사용하세요.

App Mesh에서 mTLS를 사용하면 TLS를 통한 보안 계층이 추가되고 메시의 서비스가 연결을 설정하는 클라이언트를 확인할 수 있기 때문에 양방향 피어 인증을 제공합니다. 클라이언트-서버 관계에 있는 클라이언트도 세션 협상 프로세스 중에 X.509 인증서를 제공합니다. 서버는 이 인증서를 사용하여 클라이언트를 식별하고 인증합니다. 이를 통해 신뢰할 수 있는 인증 기관(CA) 서 발급한 인증서와 유효한 인증서인지 확인할 수 있습니다.

사전 조건 및 제한 사항

사전 조건

- 자체 관리형 또는 관리형 노드 그룹이 있는 Amazon Elastic Kubernetes Service(Amazon EKS) 클러스터
- SDS가 활성화된 상태로 클러스터에 배포된 App Mesh 컨트롤러
- AWS Certificate Manager(ACM)의 프라이빗 인증서로, AWS 프라이빗 CA에서 발행합니다.

제한 사항

- SPIRE 에이전트는 쿠버네티스로 실행해야 하므로 SPIRE를 AWS Fargate에 설치할 수 없습니다.
DaemonSet

제품 버전

- AWS App Mesh Controller 차트 1.3.0 이상

아키텍처

다음 다이어그램은 VPC에서 App Mesh를 사용하는 EKS 클러스터를 보여줍니다. 한 워커 노드의 SPIRE 서버는 다른 워커 노드의 SPIRE 에이전트 및 AWS 사설 CA와 통신합니다. Envoy는 SPIRE 에이전트 워커 노드 간의 mTLS 통신에 사용됩니다.

다이어그램은 다음 단계들을 보여줍니다.

1. 인증서가 발급됩니다.
2. 인증서 서명 및 인증서를 요청합니다.

도구

서비스

- [AWS Private CA](#) – AWS Private Certificate Authority(AWS Private CA)를 사용하면 온프레미스 CA를 운영하는 데 드는 투자 및 유지 관리 비용 없이 루트 및 하위 CA를 비롯한 사설 CA 계층을 생성할 수 있습니다.
- [AWS App Mesh](#) – AWS App Mesh는 서비스를 손쉽게 모니터링하고 제어할 수 있는 서비스 메시입니다. App Mesh는 서비스 통신 방식을 표준화하여 애플리케이션의 모든 서비스에 대해 일관된 가시성과 네트워크 트래픽 제어를 제공합니다.
- [Amazon EKS](#) – Amazon Elastic Kubernetes Service(Amazon EKS)는 Kubernetes 컨트롤 플레인 또는 노드를 설치, 작동 및 유지 관리할 필요 없이 AWS에서 Kubernetes를 실행하는 데 사용할 수 있는 관리형 서비스입니다.

기타 도구

- [Helm](#) – Kubernetes용 Helm 패키지 관리자는 Kubernetes 클러스터에서 애플리케이션을 설치하고 관리하는 데 도움이 됩니다. 이 패턴은 Helm을 사용하여 AWS App Mesh Controller를 배포합니다.
- [AWS App Mesh 컨트롤러 차트](#) - 이 패턴은 AWS App Mesh 컨트롤러 차트를 사용하여 Amazon EKS에서 AWS App Mesh를 활성화합니다.

에픽

환경 설정

작업	설명	필요한 기술
Amazon EKS에서 App Mesh를 설정합니다.	리포지토리 에 제공된 기본 배포 단계를 따르세요.	DevOps 엔지니어
SPIRE를 설치합니다.	spire_setup.yaml 을 사용하여 EKS 클러스터에 SPIRE를 설치합니다.	DevOps 엔지니어
AWS 프라이빗 CA 인증서를 설치합니다.	AWS 설명서 의 지침에 따라 프라이빗 루트 CA용 인증서를 생성하고 설치합니다.	DevOps 엔지니어
클러스터 노드 인스턴스 역할에 권한을 부여합니다.	클러스터 노드 인스턴스 역할에 정책을 연결하려면 추가 정보 섹션에 있는 코드를 사용하세요.	DevOps 엔지니어
AWS 프라이빗 CA용 SPIRE 플러그인을 추가합니다.	SPIRE 서버 구성에 플러그인을 추가하려면 추가 정보 섹션에 있는 코드를 사용하세요. <code>certificate_authority_arn</code> Amazon 리소스 이름(ARN)을 프라이빗 CA ARN으로 바꿉니다. 사용되는 서명 알고리즘은 사실 CA의 서명 알고리즘과 동일해야 합니다.	DevOps 엔지니어

작업	설명	필요한 기술
	<p>your_region 을 AWS 리전으로 바꿉니다.</p> <p>플러그인에 대한 자세한 내용은 서버 플러그인: UpstreamAuthority "aws_pca"를 참조하십시오.</p>	
<p>bundle.cert를 업데이트합니다.</p>	<p>SPIRE 서버를 생성하면 spire-bundle.yaml 파일이 생성됩니다. spire-bundle.yaml 파일의 bundle.crt 값을 사설 CA에서 공개 인증서로 변경합니다.</p>	<p>DevOps 엔지니어</p>

워크로드 배포 및 등록

작업	설명	필요한 기술
<p>SPIRE에 노드 및 워크로드 항목을 등록합니다.</p>	<p>SPIRE 서버에 노드 및 워크로드(서비스)를 등록하려면 리포지토리의 코드를 사용하세요.</p>	<p>DevOps 엔지니어</p>
<p>mTL이 활성화된 상태로 App Mesh에서 메시를 생성합니다.</p>	<p>App Mesh에서 마이크로서비스 애플리케이션의 모든 구성요소(예: 가상 서비스, 가상 라우터, 가상 노드)를 사용하여 새 메시를 생성합니다.</p>	<p>DevOps 엔지니어</p>
<p>등록된 항목을 검사합니다.</p>	<p>다음 명령을 실행하여 노드와 워크로드에 대해 등록된 항목을 검사할 수 있습니다.</p> <pre data-bbox="591 1787 1029 1883">kubect1 exec -n spire spire-server-0 -- /</pre>	<p>DevOps 엔지니어</p>

작업	설명	필요한 기술
	<pre>opt/spire/bin/spire-server entry show</pre> <p>그러면 SPIRE Agent의 항목이 표시됩니다.</p>	

mTLS 트래픽 검증

작업	설명	필요한 기술
<p>mTLS 트래픽을 검증합니다.</p>	<ol style="list-style-type: none"> 프론트엔드 서비스에서 백엔드 서비스에 HTTP 헤더를 보내고 SPIRE에 등록된 서비스로 응답이 성공했는지 확인합니다. 상호 TLS 인증의 경우 다음 명령을 실행하여 <code>ssl.handshake</code> 통계를 검사할 수 있습니다. <pre>kubectl exec -it \$POD -n \$NAMESPACE -c envoy -- curl http:// localhost:9901/stats grep ssl.handshake</pre> <p>이전 명령을 실행한 후 리스너 <code>ssl.handshake</code> 개수를 확인할 수 있습니다. 리스너 수는 다음 예와 비슷합니다.</p> <pre>listener.0.0.0.0_1 5000.ssl.handshake: 2</pre>	<p>DevOps 엔지니어</p>

작업	설명	필요한 기술
<p>인증서가 AWS 사설 CA에서 발급되고 있는지 검증합니다.</p>	<p>SPIRE 서버의 로그를 보면 업스트림 사설 CA에서 플러그인이 올바르게 구성되었고 인증서가 발급되었는지 확인할 수 있습니다. 다음 명령을 실행합니다.</p> <pre data-bbox="597 537 1027 657">kubect1 logs spire-server-0 -n spire</pre> <p>그런 다음 생성된 로그를 확인합니다. 이 코드는 서버의 이름이 spire-server-0 이라고 지정되고 스파이어 네임스페이스에서 호스팅된다고 가정합니다. 플러그인이 성공적으로 로드되고 업스트림 사설 CA에 연결되는 것을 확인할 수 있을 것입니다.</p>	<p>DevOps 엔지니어</p>

관련 리소스

- [Amazon EKS의 AWS App Mesh에서 SPIFFE/SPIRE를 통해 MTL 사용](#)
- [다중 계정 Amazon EKS 환경에서 SPIFFE/SPIRE를 사용하여 AWS App Mesh에서 MTL을 활성화합니다](#)
- [이 패턴에 사용된 연습](#)
- [서버 플러그인: UpstreamAuthority “aws_pca”](#)
- [Quickstart for Kubernetes](#)

추가 정보

클러스터 노드 인스턴스 역할에 관한 연결

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "ACMPCASigning",
      "Effect": "Allow",
      "Action": [
        "acm-pca:DescribeCertificateAuthority",
        "acm-pca:IssueCertificate",
        "acm-pca:GetCertificate",
        "acm:ExportCertificate"
      ],
      "Resource": "*"
    }
  ]
}
AWS Managed Policy: "AWSAppMeshEnvoyAccess"
```

ACM용 SPIRE 플러그인 추가

Add the SPIRE plugin for ACM

Change `certificate_authority_arn` to your PCA ARN. The signing algorithm used must be the same as the signing algorithm on the PCA. Change `your_region` to the appropriate AWS Region.

```
UpstreamAuthority "aws_pca" {
  plugin_data {
    region = "your_region"
    certificate_authority_arn = "arn:aws:acm-pca:...."
    signing_algorithm = "your_signing_algorithm"
  }
}
```

AWS Batch를 사용하여 Amazon RDS for PostgreSQL DB 인스턴스 백업 자동화

작성자: Kirankumar Chandrashekar(AWS)

환경: PoC 또는 파일럿

기술: 컨테이너 및 마이크로서비스, 데이터베이스, DevOps

워크로드: 기타 모든 워크로드

AWS 서비스: 아마존 RDS, AWS Batch, 아마존, AWS Lambda CloudWatch, 아마존 S3

요약

PostgreSQL 데이터베이스를 백업하는 것은 중요한 작업이며, 보통 기본적으로 COPY 명령을 사용하여 PostgreSQL 데이터베이스의 스키마 및 데이터 덤프를 생성하는 [pg_dump 유틸리티](#)를 사용하여 완료할 수 있습니다. 하지만 여러 PostgreSQL 데이터베이스를 정기적으로 백업해야 하는 경우 이 프로세스가 반복될 수 있습니다. PostgreSQL 데이터베이스가 클라우드에서 호스팅되는 경우, PostgreSQL용 Amazon Relational Database Service(Amazon RDS)에서 제공하는 [자동 백업](#) 기능도 활용할 수 있습니다. 이 패턴은 pg_dump 유틸리티를 사용하여 Amazon RDS for PostgreSQL DB 인스턴스 정기 백업을 자동화하는 방법을 설명합니다.

참고: 이 지침에서는 Amazon RDS를 사용하고 있다고 가정합니다. 하지만 Amazon RDS 외부에서 호스팅되는 PostgreSQL 데이터베이스에도 이 접근 방식을 사용할 수 있습니다. 백업을 수행하려면 AWS Lambda 함수가 데이터베이스에 액세스할 수 있어야 합니다.

시간 기반 Amazon CloudWatch Events 이벤트는 Amazon RDS에서 PostgreSQL DB [인스턴스의 메타 데이터에 적용된 특정 백업 태그](#)를 검색하는 Lambda 함수를 시작합니다. PostgreSQL DB 인스턴스에 BKP:AutomatedDbdump = Active 태그와 기타 필수 백업 태그가 있는 경우, Lambda 함수는 각 데이터베이스 백업에 대한 개별 작업을 AWS Batch에 제출합니다.

AWS Batch는 이러한 작업을 처리하고 백업 데이터를 Amazon Simple Storage Service(Amazon S3) 버킷에 업로드합니다. 이 패턴은 Dockerfile과 entrypoint.sh 파일을 사용하여 AWS Batch 작업에서 백업을 만드는 데 사용되는 Docker 컨테이너 이미지를 빌드합니다. 백업 프로세스가 완료되면 AWS Batch는 Amazon DynamoDB의 인벤토리 테이블에 백업 세부 정보를 기록합니다. 추가 보호 수단으

로, CloudWatch 이벤트 이벤트는 AWS Batch에서 작업이 실패할 경우 Amazon Simple Notification Service (Amazon SNS) 알림을 시작합니다.

사전 조건 및 제한 사항

사전 조건

- 활성 상태의 AWS 계정.
- 기존의 관리형 또는 비관리형 컴퓨팅 환경입니다. 자세한 내용은 AWS Batch 설명서의 [관리형 및 비관리형 컴퓨팅 환경](#) 참조하세요.
- [AWS 명령줄 인터페이스\(CLI\) 버전 2 도커 이미지](#), 설치 및 구성됨.
- 기존 Amazon RDS for PostgreSQL DB 인스턴스입니다.
- 기존 S3 버킷입니다.
- [Docker](#), Linux, macOS 또는 Windows에 설치 및 구성됨
- Lambda에서의 코딩에 익숙합니다.

아키텍처

기술 스택

- 아마존 CloudWatch 이벤트
- Amazon DynamoDB
- Amazon Elastic Container Registry (Amazon ECR)
- Amazon RDS
- Amazon SNS
- Amazon S3
- AWS Batch
- AWS Key Management Service (AWS KMS)
- AWS Lambda
- AWS Secrets Manager
- Docker

도구

- [Amazon CloudWatch CloudWatch Events](#) — Events는 AWS 리소스의 변경 사항을 설명하는 시스템 이벤트의 스트림을 거의 실시간으로 제공합니다.
- [Amazon DynamoDB](#) - DynamoDB는 완전 관리형 NoSQL 데이터베이스 서비스로, 원활한 확장성과 함께 빠르고 예측 가능한 성능을 제공합니다.
- [Amazon ECR](#) - Amazon Elastic Container Registry(Amazon ECR)는 안전하고 확장 가능하고 신뢰할 수 있는 관리형 AWS 컨테이너 이미지 레지스트리 서비스입니다.
- [Amazon RDS](#) - Amazon Relational Database Service(RDS)는 AWS 클라우드의 관계형 데이터베이스를 더 쉽게 설치, 운영 및 확장할 수 있게 하는 웹 서비스입니다.
- [Amazon SNS](#) – Amazon Simple Notification Service(Amazon SNS)는 게시자에서 구독자로 메시지를 전송하는 관리형 서비스입니다.
- [Amazon S3](#) – Amazon Simple Storage Service(S3)는 인터넷에 대한 스토리지입니다.
- [AWS Batch](#) - AWS Batch는 AWS 클라우드에서 배치 컴퓨팅 워크로드를 실행할 수 있도록 도와줍니다.
- [AWS KMS](#) - AWS Key Management Service(AWS KMS)는 데이터 암호화에 사용하는 암호화 키를 쉽게 생성하고 제어할 수 있게 해주는 관리형 서비스입니다.
- [AWS Lambda](#) - Lambda는 서버를 프로비저닝하거나 관리하지 않고도 코드를 실행할 수 있게 해주는 컴퓨팅 서비스입니다.
- [AWS Secrets Manager](#) - Secrets Manager는 코드의 암호를 포함해 하드코딩된 보안 인증 정보를 Secrets Manager에서 프로그래밍 방식으로 보안 암호를 검색하도록 하는 API 호출로 바꿀 수 있습니다.
- [Docker](#) - Docker를 사용하면 개발자가 모든 애플리케이션을 가볍고 휴대가 가능하며 자급자족할 수 있는 컨테이너로 쉽게 포장, 배송 및 실행할 수 있습니다.

Amazon RDS의 PostgreSQL DB 인스턴스에는 [메타데이터에 태그가 적용](#)되어 있어야 합니다.

Lambda 함수는 태그를 검색하여 백업해야 하는 DB 인스턴스를 식별하며, 일반적으로 다음과 같은 태그가 사용됩니다.

태그

설명

kp:AutomatedDBDump = Active

Amazon RDS DB 인스턴스를 백업 대상으로 식별합니다.

<p>bkp: = AutomatedBackupSecret <secret_name ></p>	<p>Amazon RDS 로그인 보안 인증 정보가 포함된 Secrets Manager 암호를 식별합니다.</p>
<p>bkp:AutomatedDBDumpS3Bucket = <s3_bucket_name></p>	<p>백업을 전송할 S3 버킷을 식별합니다.</p>
<p>BKP: 오토메이티드 DB DumpFrequency</p> <p>BKP: 자동 DB DumpTime</p>	<p>데이터베이스를 백업해야 하는 빈도와 시간을 확인합니다.</p>
<p>bkp:pgdumpcommand = <pgdump_command></p>	<p>백업을 수행해야 하는 데이터베이스를 식별합니다.</p>

에픽

DynamoDB에 인벤토리 테이블 생성

작업	설명	필요한 기술
DynamoDB에서 테이블을 생성합니다.	AWS Management Console에 로그인한 다음 Amazon DynamoDB 콘솔을 열고 테이블을 생성합니다. 이 이야기와 다른 이야기에 대한 도움이 필요하면 관련 리소스 섹션을 참조하십시오.	클라우드 관리자, 데이터베이스 관리자
테이블이 생성되었는지 확인합니다.	aws dynamodb describe-table --table-name <table-name> grep TableStatus 명령을 실행합니다. 테이블이 존재하면 명령이 "TableStatus": "ACTIVE", 결과를 반환합니다.	클라우드 관리자, 데이터베이스 관리자

AWS Batch에서 작업 실패 이벤트에 대한 SNS 주제 생성

작업	설명	필요한 기술
SNS 주제를 생성합니다.	Amazon SNS 콘솔을 열고 주제를 선택한 다음 JobFailedAlert 이름을 사용하여 SNS 주제를 생성합니다. 주제에 대해 활성 이메일 주소를 구독하고, 이메일 수신함을 확인하여 AWS 알림의 SNS 구독 이메일을 확인합니다.	클라우드 관리자
AWS Batch에 대한 작업 실패 이벤트 규칙을 생성합니다.	Amazon CloudWatch 콘솔을 열고 [Events] 를 선택한 다음 [규칙 생성] 을 선택합니다. 고급 옵션 보기를 선택하고 편집을 선택합니다. 대상이 처리할 이벤트를 선택하는 패턴 빌드의 경우, 기존 텍스트를 추가 정보 섹션의 “Failed job event” 코드로 대체합니다. 이 코드는 AWS Batch에 CloudWatch 이벤트가 있을 때 시작되는 Failed 이벤트 규칙을 정의합니다.	클라우드 관리자
이벤트 규칙 대상을 추가합니다.	대상에서 대상 추가를 선택하고 JobFailedAlert SNS 주제를 선택합니다. 나머지 세부 정보를 구성하고 Cloudwatch 이벤트를 규칙을 생성합니다.	클라우드 관리자

도커 이미지를 빌드하고 Amazon ECR 리포지토리에 푸시하기

작업	설명	필요한 기술
Amazon ECR 리포지토리를 생성합니다.	Amazon ECR 콘솔을 열고 리포지토리를 만들 AWS 리전을 선택합니다. 리포지토리를 선택하고 리포지토리 생성을 선택합니다. 요구 사항에 따라 리포지토리를 구성합니다.	클라우드 관리자
Dockerfile을 씁니다.	Docker에 로그인하고 추가 정보 섹션의 "Sample Dockerfile" 및 "Sample entrypoint.sh file"을 사용하여 Dockerfile을 빌드합니다.	DevOps 엔지니어
도커 이미지를 생성하여 Amazon ECR 리포지토리로 푸시합니다.	Dockerfile을 도커 이미지로 빌드하고 Amazon ECR 리포지토리로 푸시합니다. 이 사례에 대한 도움이 필요하면 관련 리소스 섹션을 참조하세요.	DevOps 엔지니어

AWS Batch 구성 요소 생성

작업	설명	필요한 기술
AWS Batch 작업 정의를 생성합니다.	AWS Batch 콘솔을 열고 Amazon ECR 리포지토리의 URI(Uniform Resource Identifier)를 속성 Image(으)로 포함하는 작업 정의를 생성합니다.	클라우드 관리자
AWS Batch 작업 대기열을 구성합니다.	AWS Batch 콘솔에서 작업 대기열을 선택한 다음 대기열 생성을 선택합니다. AWS Batch	클라우드 관리자

작업	설명	필요한 기술
	가 컴퓨팅 환경 내의 리소스에서 작업을 실행할 때까지 작업을 저장할 작업 대기열을 생성합니다. 중요: DynamoDB 인벤토리 테이블에 백업 세부 정보를 기록하는 로직을 AWS Batch에 작성해야 합니다.	

Lambda 함수 생성 및 예약

작업	설명	필요한 기술
Lambda 함수를 생성하여 태그를 검색합니다.	PostgreSQL DB 인스턴스에서 태그를 검색하고 백업 후 정보를 식별하는 Lambda 함수를 생성합니다. Lambda 함수가 <code>bkp:AutomatedDBDump = Active</code> 태그 및 기타 모든 필수 태그를 식별할 수 있는지 확인하세요. 중요: 또한 Lambda 함수가 AWS Batch 작업 대기열에도 작업을 추가할 수 있어야 합니다.	DevOps 엔지니어
시간 기반 CloudWatch 이벤트 이벤트 생성	Amazon CloudWatch 콘솔을 열고 cron 표현식을 사용하여 Lambda 함수를 정기적으로 실행하는 이벤트 CloudWatch 이벤트를 생성합니다. 중요: 예정된 이벤트는 모두 UTC 시간대를 사용합니다.	클라우드 관리자

백업 자동화 테스트

작업	설명	필요한 기술
Amazon KMS 키를 생성합니다.	Amazon KMS 콘솔을 열고 AWS Secrets Manager에 저장된 Amazon RDS 보안 인증을 암호화하는 데 사용할 수 있는 KMS 키를 생성합니다.	클라우드 관리자
AWS Secrets Manager 보안 암호를 생성합니다.	AWS Secrets Manager 콘솔을 열고 Amazon RDS for PostgreSQL 데이터베이스 보안 인증을 암호로 저장합니다.	클라우드 관리자
PostgreSQL DB 인스턴스에 필요한 태그를 추가합니다.	Amazon RDS 콘솔을 열고 자동으로 백업하려는 PostgreSQL DB 인스턴스에 태그를 추가합니다. 도구 섹션의 테이블에 있는 태그를 사용할 수 있습니다. 동일한 Amazon RDS 인스턴스 내에 있는 여러 PostgreSQL 데이터베이스의 백업이 필요한 경우, <code>bkp:pgdumpcommand</code> 태그에 대한 값으로 <code>-d test:-d test1</code> 을(를) 사용합니다. 중요: <code>test</code> 및 <code>test1</code> 은(는) 데이터베이스 이름입니다. 콜론(:) 뒤에 공백이 없는지 확인하세요.	클라우드 관리자
백업 자동화를 확인합니다.	백업 자동화를 확인하려면 Lambda 함수를 호출하거나 백업 일정이 시작될 때까지 기다릴 수 있습니다. 백업 프로세스가 완료되면 DynamoDB 인벤토리 테이블에 PostgreSQL DB	클라우드 관리자

작업	설명	필요한 기술
	인스턴스에 대한 유효한 백업 항목이 있는지 확인합니다. 두 값이 일치하면 백업 자동화 프로세스가 성공한 것입니다.	

관련 리소스

DynamoDB에 인벤토리 테이블 생성

- [Amazon DynamoDB 테이블 생성](#)

AWS Batch에서 작업 실패 이벤트에 대한 SNS 주제 생성

- [Amazon SNS 주제 생성](#)
- [AWS Batch에서 작업 실패 이벤트에 대한 SNS 알림 전송](#)

도커 이미지를 빌드하고 Amazon ECR 리포지토리에 푸시하기

- [Amazon ECR 리포지토리 생성](#)
- [Dockerfile을 작성하고, 도커 이미지를 생성하여 Amazon ECR로 푸시](#)

AWS Batch 구성 요소 생성

- [AWS Batch 작업 정의 생성](#)
- [컴퓨팅 환경 및 AWS Batch 작업 대기열 구성](#)
- [AWS Batch에서 작업 대기열을 생성](#)

Lambda 함수 생성

- [Lambda 함수 생성 및 코드 작성](#)
- [DynamoDB와 함께 Lambda 사용](#)

이벤트 이벤트 생성 CloudWatch

- [시간 기반 CloudWatch 이벤트 이벤트 생성](#)
- [Cloudwatch Events에서 cron 표현식 사용](#)

백업 자동화 테스트

- [Amazon KMS 키 생성](#)
- [Secrets Manager 보안 암호 생성](#)
- [Amazon RDS 인스턴스에 태그 추가](#)

추가 정보

작업 실패 이벤트:

```
{
  "detail-type": [
    "Batch Job State Change"
  ],
  "source": [
    "aws.batch"
  ],
  "detail": {
    "status": [
      "FAILED"
    ]
  }
}
```

샘플 Dockerfile:

```
FROM alpine:latest
RUN apk --update add py-pip postgresql-client jq bash && \
pip install awscli && \
rm -rf /var/cache/apk/*
ADD entrypoint.sh /usr/bin/
RUN chmod +x /usr/bin/entrypoint.sh
ENTRYPOINT ["entrypoint.sh"]
```

샘플 entrypoint.sh file:

```
#!/bin/bash
set -e
DATETIME=`date +"%Y-%m-%d_%H_%M"`
FILENAME=RDS_PostGres_dump_${RDS_INSTANCE_NAME}
FILE=${FILENAME}_${DATETIME}

aws configure --profile new-profile set role_arn arn:aws:iam:${TargetAccountId}:role/
${TargetAccountRoleName}
aws configure --profile new-profile set credential_source EcsContainer

echo "Central Account access provider IAM role is: "
aws sts get-caller-identity

echo "Target Customer Account access provider IAM role is: "
aws sts get-caller-identity --profile new-profile

securestring=$(aws secretsmanager get-secret-value --secret-id $SECRETID --output json
--query 'SecretString' --region=$REGION --profile new-profile)

if [[ ${securestring} ]]; then
    echo "successfully accessed secrets manager and got the credentials"
    export PGPASSWORD=$(echo $securestring | jq --raw-output | jq -r '.DB_PASSWORD')
    PGSQL_USER=$(echo $securestring | jq --raw-output | jq -r '.DB_USERNAME')
    echo "Executing pg_dump for the PostGres endpoint ${PGSQL_HOST}"
    # pg_dump -h $PGSQL_HOST -U $PGSQL_USER -n dms_sample | gzip -9 -c | aws s3 cp -
--region=$REGION --profile new-profile s3://$BUCKET/$FILE
    # in="-n public:-n private"
    IFS=':' list=($EXECUTE_COMMAND);
    for command in "${list[@]}";
    do
        echo $command;
        pg_dump -h $PGSQL_HOST -U $PGSQL_USER $command | gzip -9 -c | aws s3 cp - --
region=$REGION --profile new-profile s3://$BUCKET/$FILE-${command}.sql.gz"
```

```
    echo $?;
    if [[ $? -ne 0 ]]; then
        echo "Error occurred in database backup process. Exiting now....."
        exit 1
    else
        echo "Postgresql dump was successfully taken for the RDS endpoint
        ${PGSQL_HOST} and is uploaded to the following S3 location s3://${BUCKET}/${FILE}-
        ${command}.sql.gz"
        #write the details into the inventory table in central account
        echo "Writing to DynamoDB inventory table"
        aws dynamodb put-item --table-name ${RDS_POSTGRES_DUMP_INVENTORY_TABLE} --
region=$REGION --item '{ "accountId": { "S": ""${TargetAccountId}"" }, "dumpFileUrl":
{"S": ""s3://${BUCKET}/${FILE}-${command}.sql.gz"" }, "DumpAvailableTime": {"S":
""`date +%Y-%m-%d::%H::%M::%S` UTC""}}'
        echo $?
        if [[ $? -ne 0 ]]; then
            echo "Error occurred while putting item to DynamoDb Inventory Table.
Exiting now....."
            exit 1
        else
            echo "Successfully written to DynamoDb Inventory Table
${RDS_POSTGRES_DUMP_INVENTORY_TABLE}"
            fi
        fi
    done;
else
    echo "Something went wrong {${?}}"
    exit 1
fi

exec "$@"
```

CI/CD 파이프라인을 사용하여 Amazon EKS에서 노드 종료 핸들러 배포 자동화

작성자: 샌딕 강가파디아(AWS), 존 바르가스(AWS), 프라티딕 싱(AWS), 샌딕 가완데 (AWS) 및 비요마 사흐데바(AWS)

코드 리포지토리: [NTH를 EKS에 배포](#)

환경: 프로덕션

기술: 컨테이너 및 마이크로서비스, DevOps

AWS 서비스: AWS CodePipeline, 아마존 EKS, AWS CodeBuild

요약

Amazon Web Services(AWS) 클라우드에서 오픈 소스 프로젝트인 [AWS Node Termination Handler](#)를 사용하여 Kubernetes 내에서 Amazon Elastic Compute Cloud(Amazon EC2) 인스턴스 종료를 적절하게 처리할 수 있습니다. AWS Node Termination Handler는 EC2 인스턴스를 사용할 수 없게 만들 수 있는 이벤트에 Kubernetes 컨트롤 플레인이 적절하게 응답하도록 도와줍니다. 이러한 이벤트에는 다음이 포함됩니다.

- [EC2 인스턴스 유지 관리 일정](#)
- [Amazon EC2 스팟 인스턴스 중단](#)
- [Auto Scaling 그룹 스케일 인](#)
- 가용 영역 전반에 걸친 [Auto Scaling 그룹 리밸런싱](#)
- API 또는 AWS Management Console을 통한 EC2 인스턴스 종료

이벤트가 처리되지 않으면 애플리케이션 코드가 적절하게 중지되지 않을 수 있습니다. 또한 전체 가용성을 복구하는 데 시간이 더 오래 걸리거나 다운되는 노드에 실수로 작업을 예약할 수도 있습니다. aws-node-termination-handler (NTH)는 인스턴스 메타데이터 서비스(IMDS)또는 대기열 프로세서라는 두 가지 모드로 작동할 수 있습니다. 두 모드에 대한 자세한 내용은 [Readme 파일](#)을 참조하세요.

이 패턴은 지속적 통합 및 지속적 전달(CI/CD) 파이프라인을 통해 대기열 프로세서를 사용하여 NTH를 자동으로 배포합니다.

참고: [EKS 관리형 노드 그룹](#) 사용하는 경우에는 aws-node-termination-handler이(가) 필요하지 않습니다.

사전 조건 및 제한 사항

사전 조건

- 활성 상태의 AWS 계정.
- AWS Management Console에서 사용할 수 있도록 지원되는 웹 브라우저. [지원되는 브라우저 목록](#)을 참조하세요.
- AWS Cloud Development Kit(AWS CDK)가 [설치됨](#).
- kubectl, Kubernetes 명령줄 도구가 [설치됨](#).
- eksctl, Amazon Elastic Kubernetes Service(Amazon EKS)를 위한 AWS Command Line Interface(AWS CLI)가 [설치됨](#).
- 버전 1.20 이상으로 실행 중인 EKS 클러스터
- EKS 클러스터에 연결된 자체 관리형 노드 그룹입니다. 자체 관리형 노드 그룹이 있는 Amazon EKS 클러스터를 생성하려면 다음 명령을 실행합니다.

```
eksctl create cluster --managed=false --region <region> --name <cluster_name>
```

eksctl에 관한 자세한 내용은 [eksctl 설명서](#)를 참조하세요.

- 클러스터에 대한 AWS Identity and Access Management(IAM) OpenID Connect(OIDC) 공급자입니다. 자세한 내용은 [클러스터에 대한 IAM OIDC 공급자 생성](#)을 참조하세요.

제한 사항

- Amazon EKS 서비스를 지원하는 AWS 리전을 사용해야 합니다.

제품 버전

- Kubernetes 버전 1.20 이상
- eksctl 버전 0.107.0 이상
- AWS CDK 버전 2.27.0 이상

아키텍처

대상 기술 스택

- Virtual Private Cloud(VPC)
- EKS 클러스터
- Amazon Simple Queue Service(Amazon SQS)
- IAM
- Kubernetes

대상 아키텍처

다음 다이어그램은 노드 종료 시작되는 시점의 end-to-end 단계를 개괄적으로 보여줍니다.

다이어그램에 표시된 워크플로우는 다음과 같은 개괄적인 단계로 구성되어 있습니다.

1. 자동 조정 EC2 인스턴스 종료 이벤트는 SQS 대기열로 전송됩니다.
2. NTH 포드는 SQS 대기열의 새 메시지를 모니터링합니다.
3. NTH 포드는 새 메시지를 수신하고 다음을 수행합니다.
 - 새 포드가 노드에서 실행되지 않도록 노드를 차단합니다.
 - 노드를 빼서 기존 포드를 비우도록 합니다.
 - 노드를 종료할 수 있도록 Auto Scaling 그룹에 수명 주기 후크 신호를 전송합니다.

자동화 및 규모 조정

- 코드는 AWS CDK에서 관리하고 배포하며, AWS CloudFormation 중첩 스택을 기반으로 합니다.
- [Amazon EKS 컨트롤 플레인](#)은 여러 가용 영역에 걸쳐 실행되어고가용성을 보장합니다.
- [자동 조정](#)을 위해 Amazon EKS는 Kubernetes [Cluster Autoscaler](#) 및 [Karpenter](#)를 지원합니다.

도구

서비스

- [AWS Cloud Development Kit\(AWS CDK\)](#)는 AWS 클라우드 인프라를 코드로 정의하고 프로비저닝하는 데 도움이 되는 소프트웨어 개발 프레임워크입니다.
- [CodeBuildAWS](#)는 소스 코드를 컴파일하고, 단위 테스트를 실행하고, 배포할 준비가 된 아티팩트를 생성하는 데 도움이 되는 완전관리형 빌드 서비스입니다.
- [CodeCommitAWS](#)는 자체 소스 제어 시스템을 관리할 필요 없이 Git 리포지토리를 비공개로 저장하고 관리할 수 있는 버전 제어 서비스입니다.
- [AWS](#)는 소프트웨어 릴리스의 여러 단계를 신속하게 모델링 및 구성하고 소프트웨어 변경 사항을 지속적으로 릴리스하는 CodePipeline 데 필요한 단계를 자동화하도록 지원합니다.
- [Amazon Elastic Kubernetes Service \(Amazon EKS\)](#)는 자체 Kubernetes 컨트롤 플레인이나 노드를 설치하거나 유지 관리할 필요 없이 AWS에서 Kubernetes를 실행할 수 있도록 도와줍니다.
- [Amazon EC2 Auto Scaling](#)을 사용하면 애플리케이션 가용성을 유지하고 정의된 조건에 따라 Amazon EC2 인스턴스를 자동으로 추가하거나 제거할 수 있습니다.
- [Amazon Simple Queue Service\(Amazon SQS\)](#)는 내구력 있고 가용성이 뛰어난 보안 호스팅 대기열을 제공하며 이를 통해 분산 소프트웨어 시스템과 구성 요소를 통합 및 분리할 수 있습니다.

기타 도구

- [kubectl](#)은 Kubernetes 클러스터에 대해 명령을 실행하기 위한 Kubernetes 명령줄 도구입니다. kubectl을 사용하여 애플리케이션을 배포하고, 클러스터 리소스를 검사 및 관리하고, 로그를 볼 수 있습니다.

code

이 패턴의 코드는 GitHub .com의 [deploy-nth-to-eks](#) 리포지토리에서 확인할 수 있습니다. 코드 리포지토리에는 다음 파일 및 폴더가 포함되어 있습니다.

- `nth folder`— 노드 종료 핸들러용 AWS CloudFormation 템플릿을 스캔하고 배포하기 위한 Helm 차트, 값 파일 및 스크립트.
- `config/config.json` - 애플리케이션의 구성 파라미터 파일입니다. 이 파일에는 CDK를 배포하는 데 필요한 모든 파라미터가 포함되어 있습니다.
- `cdk` - AWS CDK 소스 코드입니다.
- `setup.sh` - 필수 CI/CD 파이프라인 및 기타 필수 리소스를 생성하기 위해 AWS CDK 애플리케이션을 배포하는 데 사용되는 스크립트입니다.
- `uninstall.sh` - 리소스를 정리하는 데 사용되는 스크립트입니다.

샘플 코드를 사용하려면 에픽 섹션의 지침을 따르십시오.

모범 사례

AWS 노드 종료 핸들러를 자동화하는 모범 사례는 다음을 참조하세요.

- [EKS 모범 사례 가이드](#)
- [노드 종료 핸들러 - 구성](#)

에픽

환경을 설정합니다.

작업	설명	필요한 기술
리포지토리를 복제합니다.	<p>SSH(보안 셸)를 사용하여 리포지토리를 복제하려면 다음 명령을 실행합니다.</p> <pre>git clone git@github.com:aws-samples/deploy-nth-to-eks.git</pre> <p>HTTPS를 사용하여 리포지토리를 복제하려면 다음 명령을 실행합니다.</p> <pre>git clone https://github.com/aws-samples/deploy-nth-to-eks.git</pre> <p>리포지토리를 복제하면 <code>deploy-nth-to-eks</code> (이)라는 이름의 폴더가 생성됩니다.</p> <p>해당 디렉터리로 변경합니다.</p>	앱 개발자, AWS DevOps, DevOps 엔지니어

작업	설명	필요한 기술
	<pre>cd deploy-nth-to-eks</pre>	
kubeconfig 파일을 설정합니다.	<p>터미널에 AWS 보안 인증을 설정하고 클러스터 역할을 맡을 권한이 있는지 확인합니다. 다음 예제 코드를 사용할 수 있습니다.</p> <pre>aws eks update-kubeconfig --name <Cluster_Name> --region <region>--role-arn <Role_ARN></pre>	AWS DevOps, DevOps 엔지니어, 앱 개발자

CI/CD 파이프라인 배포

작업	설명	필요한 기술
파라미터를 설정합니다.	<p>config/config.json 파일에서 다음과 같은 필수 파라미터를 설정합니다.</p> <ul style="list-style-type: none"> • pipelineName : AWS CDK에서 생성할 CI/CD 파이프라인의 이름(예제: deploy-nth-to-eks-pipeline)입니다. CodePipeline AWS는 이 이름을 가진 파이프라인을 생성합니다. • repositoryName : 생성할 AWS CodeCommit 리포지토리 (예:deploy-nth-to-eks-repo). AWS CDK는 	앱 개발자, AWS DevOps, DevOps 엔지니어

작업	설명	필요한 기술
	<p>이 리포지토리를 생성하고 이를 CI/CD 파이프라인의 소스로 설정합니다.</p> <p>참고: 이 솔루션은 이 CodeCommit 리포지토리와 브랜치를 생성합니다 (다음 브랜치 파라미터에 제공됨).</p> <ul style="list-style-type: none"> • <code>branch</code>: 리포지토리의 브랜치 이름(예제: <code>main</code>)입니다. 이 브랜치를 커밋하면 CI/CD 파이프라인이 시작됩니다. • <code>cfn_scan_script</code> : AWS CloudFormation 템플릿에서 NTH (<code>scan.sh</code>) 를 스캔하는 데 사용할 스크립트의 경로입니다. 이 스크립트는 AWS CodeCommit 리포지토리의 일부가 될 <code>nth</code> 폴더에 있습니다. • <code>cfn_deploy_script</code>: NTH용 AWS CloudFormation 템플릿을 배포하는 데 사용할 스크립트 경로 (<code>installApp.sh</code>). • <code>stackName</code> : 배포할 CloudFormation 스택의 이름. • <code>eksClusterName</code> : 기존 EKS 클러스터 이름입니다. • <code>eksClusterRole</code> : 모든 Kubernetes API 호출을 위해 EKS 클러스터에 액세스하는 데 사용되는 IAM 역할 	

작업	설명	필요한 기술
	<p>입니다(예제: clusteradmin). 일반적으로 aws-auth ConfigMap 에 이 역할이 추가됩니다.</p> <ul style="list-style-type: none"> • create_cluster_role : eksClusterRole IAM 역할을 생성하려면 yes를 입력합니다. eksClusterRole 파라미터에 기존 클러스터 역할을 제공하려면 no를 입력합니다. • create_iam_oidc_provider : 클러스터에 대한 IAM OIDC 공급자를 생성하려면 yes를 입력합니다. IAM OIDC 공급자가 이미 존재하면 no를 입력합니다. 자세한 내용은 클러스터에 대한 IAM OIDC 공급자 생성을 참조하세요. • AsgGroupName : EKS 클러스터에 속하는 Auto Scaling 그룹 이름을 심포로 구분한 목록입니다(예제: ASG_Group_1, ASG_Group_2). • region: 클러스터가 위치한 AWS 리전의 이름(예제: us-east-2)입니다. • install_cdk : AWS CDK가 현재 머신에 설치되어 있지 않은 경우 yes를 입력합니다. cdk --version 명령 	

작업	설명	필요한 기술
	<p>령을 실행하여 설치된 AWS CDK 버전이 2.27.0 이상인지 확인합니다. 이 경우 no를 입력합니다.</p> <p>yes를 입력하면 setup.sh 스크립트가 sudo npm install -g cdk@2.27.0 명령을 실행하여 AWS CDK를 시스템에 설치합니다. 스크립트에는 sudo 권한이 필요하므로 메시지가 표시되면 계정 암호를 제공하세요.</p>	
<p>CI/CD 파이프라인을 생성하여 NTH를 배포합니다.</p>	<p>setup.sh 스크립트를 실행합니다.</p> <div data-bbox="594 1016 1027 1098" style="border: 1px solid #ccc; border-radius: 10px; padding: 5px; margin: 10px 0;"> <pre>./setup.sh</pre> </div> <p>스크립트는 파일의 사용자 입력 파라미터를 기반으로 예제 코드, 파이프라인, CodeBuild 프로젝트가 포함된 CodeCommit 리포지토리를 생성하는 AWS CDK 애플리케이션을 배포합니다. config/config.json</p> <p>이 스크립트는 sudo 명령으로 npm 패키지를 설치할 때 암호를 묻습니다.</p>	<p>앱 개발자, AWS DevOps, DevOps 엔지니어</p>

작업	설명	필요한 기술
<p>CI/CD 파이프라인을 검토합니다.</p>	<p>AWS Management Console을 열고 스택에 생성된 다음 리소스를 검토하세요.</p> <ul style="list-style-type: none"> • CodeCommit 폴더 내용이 포함된 nth 리포지토리 • CloudFormation 템플릿의 취약점을 스캔하는 AWS CodeBuild 프로젝트 cfn-scan. • CodeBuild 이 프로젝트는 Nth-Deploy AWS CloudFormation 템플릿과 해당 NTH Helm 차트를 AWS 파이프라인을 통해 배포합니다. CodePipeline • NTH를 배포하기 위한 CodePipeline 파이프라인. <p>파이프라인이 성공적으로 실행되면 EKS 클러스터에 Helm 릴리스 aws-node-termination-handler (이)가 설치됩니다. 또한 클러스터의 kube-system 네임스페이스에서 aws-node-termination-handler 이름의 포드가 실행 중입니다.</p>	<p>앱 개발자, AWS DevOps, DevOps 엔지니어</p>

NTH 배포 테스트

작업	설명	필요한 기술
Auto Scaling 그룹 스케일 인 이벤트를 시뮬레이션합니다.	<p>자동 조정 스케일 인 이벤트를 시뮬레이션하려면 다음을 수행합니다.</p> <ol style="list-style-type: none"> 1. AWS console EC2 콘솔을 열고 Auto Scaling 그룹을 선택합니다. 2. config/config.json 에서 제공한 것과 이름이 같은 Auto Scaling 그룹을 선택하고 편집을 선택합니다. 3. 원하는 용량과 최소 용량을 1씩 줄입니다. 4. 업데이트를 선택합니다. 	
로그를 검토합니다.	<p>스케일 인 이벤트 중에 NTH Pod는 해당 워커 노드(스케일 인 이벤트의 일부로 종료될 EC2 인스턴스)를 차단하고 드레이닝합니다. 로그를 확인하려면 추가 정보 섹션의 코드를 사용합니다.</p>	<p>앱 개발자, AWS DevOps, DevOps 엔지니어</p>

정리

작업	설명	필요한 기술
모든 AWS 리소스를 정리합니다.	<p>이 패턴으로 생성된 리소스를 정리하려면 다음 명령을 실행합니다.</p> <pre>./uninstall.sh</pre>	<p>DevOps 엔지니어</p>

작업	설명	필요한 기술
	그러면 CloudFormation 스택이 삭제되어 이 패턴으로 생성된 모든 리소스가 정리됩니다.	

문제 해결

문제	Solution
npm 레지스트리가 올바르게 설정되지 않았습니다.	<p>이 솔루션을 설치하는 동안 스크립트는 npm install을 설치하여 필수 패키지를 모두 다운로드 합니다. 설치 중에 “모듈을 찾을 수 없습니다”라는 메시지가 표시되면 npm 레지스트리가 올바르게 설정되지 않은 것일 수 있습니다. 현재 레지스트리 설정을 확인하려면 다음 명령을 실행합니다.</p> <pre>npm config get registry</pre> <p>https://registry.npmjs.org/ (으)로 레지스트리를 설정하려면 다음 명령을 실행합니다.</p> <pre>npm config set registry https://registry.npmjs.org</pre>
SQS 메시지 전송을 지연합니다.	<p>문제 해결의 일환으로, NTH 포드로 SQS 메시지 전송을 지연하려는 경우 SQS 전송 지연 파라미터를 조정할 수 있습니다. 자세한 내용은 Amazon SQS 지연 대기열을 참조하세요.</p>

관련 리소스

- [AWS 노드 종료 핸들러 소스 코드](#)

- [EC2 워크숍](#)
- [AWS CodePipeline](#)
- [Amazon Elastic Kubernetes Service\(Amazon EKS\)](#)
- [AWS Cloud Development Kit](#)
- [AWS CloudFormation](#)

추가 정보

1. NTH 포드 이름을 찾습니다.

```
kubectl get pods -n kube-system |grep aws-node-termination-handler
aws-node-termination-handler-65445555-kbqc7 1/1 Running 0 26m
kubectl get pods -n kube-system |grep aws-node-termination-handler
aws-node-termination-handler-65445555-kbqc7 1/1 Running 0 26m
```

2. 로그를 확인합니다. 예제 로그는 다음과 같습니다. Auto Scaling 그룹 수명 주기 후크 완료 신호를 보내기 전에 노드가 차단되고 드레이닝되었음을 보여줍니다.

```
kubectl -n kube-system logs aws-node-termination-handler-65445555-kbqc7
022/07/17 20:20:43 INF Adding new event to the event store
  event={"AutoScalingGroupName":"eksctl-my-cluster-target-nodegroup-
ng-10d99c89-NodeGroup-ZME36IGAP701","Description":"ASG Lifecycle Termination
event received. Instance will be interrupted at 2022-07-17 20:20:42.702
+0000 UTC \n","EndTime":"0001-01-01T00:00:00Z","EventID":"asg-lifecycle-
term-33383831316538382d353564362d343332362d613931352d383430666165636334333564","InProgress":fal
east-2.compute.internal","NodeProcessed":false,"Pods":null,"ProviderID":"aws:///us-
east-2c/i-0409f2a9d3085b80e","StartTime":"2022-07-17T20:20:42.702Z","State":""}
2022/07/17 20:20:44 INF Requesting instance drain event-id=asg-lifecycle-
term-33383831316538382d353564362d343332362d613931352d383430666165636334333564
instance-id=i-0409f2a9d3085b80e kind=SQS_TERMINATE node-name=ip-192-168-75-60.us-
east-2.compute.internal provider-id=aws:///us-east-2c/i-0409f2a9d3085b80e
2022/07/17 20:20:44 INF Pods on node node_name=ip-192-168-75-60.us-
east-2.compute.internal pod_names=["aws-node-qchsw","aws-node-termination-
handler-65445555-kbqc7","kube-proxy-mz5x5"]
2022/07/17 20:20:44 INF Draining the node
2022/07/17 20:20:44 ??? WARNING: ignoring DaemonSet-managed Pods: kube-system/aws-node-
qchsw, kube-system/kube-proxy-mz5x5
2022/07/17 20:20:44 INF Node successfully cordoned and drained
node_name=ip-192-168-75-60.us-east-2.compute.internal reason="ASG Lifecycle
```

```
Termination event received. Instance will be interrupted at 2022-07-17 20:20:42.702
+0000 UTC \n"
2022/07/17 20:20:44 INF Completed ASG Lifecycle Hook (NTH-K8S-TERM-HOOK) for instance
i-0409f2a9d3085b80e
```

CI/CD 파이프라인을 사용하여 Amazon EKS에 Java 애플리케이션 자동 구축 및 배포

작성자: MAHESH RAGHUNANDANAN(AWS), James Radtke(AWS), Jomcy Pappachen(AWS)

코드 리포지토리: aws-cicd-java-eks	환경: 프로덕션	기술: 컨테이너 및 마이크로 서비스, 클라우드 네이티브, DevOps 현대화
워크로드: 기타 모든 워크로드	AWS 서비스: AWS CloudFormation, AWS, AWS CodeCommit CodePipeline, 아마존 EC2 컨테이너 레지스트리, 아마존 EKS	

요약

이 패턴은 권장 DevSecOps 사례에 따라 자바 애플리케이션을 자동으로 빌드하고 Amazon Web Services (AWS) 클라우드의 Amazon Elastic Kubernetes Service (Amazon EKS) 클러스터에 배포하는 지속적 통합 및 지속적 전송 (CI/CD) 파이프라인을 생성하는 방법을 설명합니다. 이 패턴은 Spring Boot Java 프레임워크로 개발되고 Apache Maven을 사용하는 인사말 애플리케이션을 사용합니다.

이 패턴의 접근 방식을 사용하여 Java 애플리케이션용 코드를 빌드하고, 애플리케이션 아티팩트를 도커 이미지로 패키징하고, 이미지를 보안 스캔하고, 이미지를 Amazon EKS의 워크로드 컨테이너로 업로드할 수 있습니다. 이 패턴의 접근 방식은 긴밀하게 연결된 모놀리식 아키텍처에서 마이크로서비스 아키텍처로 마이그레이션하려는 경우에 유용합니다. 또한 이 접근 방식은 Java 애플리케이션의 전체 라이프사이클을 모니터링하고 관리하는 데 도움이 되므로 자동화 수준을 높이고 오류나 버그를 방지하는 데 도움이 됩니다.

사전 조건 및 제한 사항

사전 조건

- 활성 상태의 AWS 계정.
- AWS Command Line Interface(AWS CLI) 버전 2, 설치 및 구성됨. 이에 대한 자세한 내용은 AWS CLI 설명서에서 [AWS CLI 버전 2 설치, 업데이트 및 제거](#)를 참조하십시오.

- AWS CLI 버전 2는 Amazon EKS 클러스터를 생성하는 것과 동일한 IAM 역할로 구성해야 합니다. 해당 역할만 다른 IAM 역할을 aws-auth ConfigMap에 추가할 권한이 있기 때문입니다. AWS CLI를 구성하는 데 필요한 정보 및 단계는 AWS CLI [설명서의 구성 기본 사항을](#) 참조하십시오.
- AWS 자격 증명 및 액세스 관리 (IAM) 역할 및 권한 (CloudFormationAWS에 대한 전체 액세스 권한 포함) 이에 대한 자세한 내용은 AWS CloudFormation 설명서의 [IAM을 통한 액세스 제어를](#) 참조하십시오.
- EKS 클러스터에 있는 워커 노드의 IAM 역할 이름 및 IAM 역할 Amazon 리소스 이름(ARN)에 대한 세부 정보가 포함된 기존 Amazon EKS 클러스터.
- Amazon EKS 클러스터에 설치 및 구성된 Kubernetes Cluster Autoscaler. 자세한 내용은 Amazon EKS 설명서에서 [클러스터 자동 규모 조정](#)을 참조하십시오.
- GitHub 리포지토리의 코드에 대한 액세스.

중요 정보

AWS Security Hub는 코드에 있는 AWS CloudFormation 템플릿의 일부로 활성화됩니다. 기본적으로 Security Hub가 활성화되면 30일 무료 평가판이 제공되며, 그 이후에는 이 AWS 서비스와 관련된 비용이 발생합니다. 요금에 대한 자세한 내용은 [AWS Security Hub 요금](#)을 참조하십시오.

제품 버전

- Helm 버전 3.4.2 이상
- Apache Maven 버전 3.6.3 이상
- BridgeCrew Checkov 버전 2.2 이상
- Aqua Security Trivy 버전 0.37 이상

아키텍처

기술 스택

- AWS CodeBuild
- AWS CodeCommit
- 아마존 CodeGuru
- AWS CodePipeline
- Amazon Elastic 컨테이너 레지스트리

- Amazon Elastic Kubernetes 서비스
- 아마존 EventBridge
- AWS Security Hub
- Amazon Simple Notification Service(SNS)

대상 아키텍처

이 다이어그램은 다음 워크플로를 보여줍니다.

1. 개발자는 CodeCommit 리포지토리의 기본 브랜치에서 Java 애플리케이션 코드를 업데이트하여 풀 리퀘스트 (PR) 를 생성합니다.
2. PR이 제출되는 즉시 Amazon CodeGuru Reviewer는 코드를 자동으로 검토하고 Java 모범 사례를 기반으로 코드를 분석하고 개발자에게 권장 사항을 제공합니다.
3. PR이 기본 브랜치에 병합된 후 Amazon EventBridge 이벤트가 생성됩니다.
4. EventBridge 이벤트가 시작되고 CodePipeline 파이프라인이 시작됩니다.
5. CodePipeline CodeSecurity 스캔 단계 (지속적 보안) 를 실행합니다.
6. CodeBuild Checkov를 사용하여 Dockerfile 및 Kubernetes 배포 Helm 파일을 스캔하고 증분 코드 변경을 기반으로 애플리케이션 소스 코드를 스캔하는 보안 스캔 프로세스를 시작합니다. 응용 프로그램 소스 코드 스캔은 [CodeGuru 리뷰어 명령줄 인터페이스 \(CLI\) 래퍼](#)를 통해 수행됩니다.
7. 보안 스캔 단계가 성공하면 빌드 단계(지속적 통합)가 시작됩니다.
8. 빌드 단계에서 아티팩트를 CodeBuild 빌드하고, 아티팩트를 Docker 이미지에 패키징하고, Aqua Security Trivy를 사용하여 이미지의 보안 취약성을 스캔하고, 이미지를 Amazon ECR에 저장합니다.
9. 8단계에서 탐지된 취약성은 개발자 또는 엔지니어가 추가 분석을 위해 Security Hub에 업로드됩니다. Security Hub는 취약성 해결을 위한 개요 및 권장 사항을 제공합니다.
10. CodePipeline 파이프라인 내의 다양한 단계에 대한 이메일 알림이 Amazon SNS를 통해 전송됩니다.
11. 지속적 통합 단계가 완료되면 배포 단계 (지속적 전송) 로 CodePipeline 들어갑니다.
12. 도커 이미지는 차트 Helm을 사용하여 Amazon EKS에 컨테이너 워크로드(포드)로 배포됩니다.
13. 애플리케이션 포드는 Amazon CodeGuru Profiler Agent로 구성되어 있습니다. Amazon Profiler Agent는 애플리케이션의 프로파일링 데이터 (CPU, 힙 사용량, 지연 시간) 를 Amazon CodeGuru Profiler로 전송하여 개발자가 애플리케이션의 동작을 이해하는 데 도움이 됩니다.

도구

서비스

- [AWS는 AWS CloudFormation](#) 리소스를 설정하고, 빠르고 일관되게 프로비저닝하고, AWS 계정 및 지역 전반의 수명 주기 전반에 걸쳐 리소스를 관리할 수 있도록 지원합니다.
- [CodeBuildAWS](#)는 소스 코드를 컴파일하고, 단위 테스트를 실행하고, 배포할 준비가 된 아티팩트를 생성하는 데 도움이 되는 완전관리형 빌드 서비스입니다.
- [CodeCommitAWS](#)는 자체 소스 제어 시스템을 관리할 필요 없이 Git 리포지토리를 비공개로 저장하고 관리할 수 있는 버전 제어 서비스입니다.
- [Amazon CodeGuru Profiler](#)는 라이브 애플리케이션에서 런타임 성능 데이터를 수집하고 애플리케이션 성능을 미세 조정하는 데 도움이 되는 권장 사항을 제공합니다.
- [Amazon CodeGuru Reviewer](#)는 프로그램 분석 및 기계 학습을 사용하여 개발자가 찾기 어려운 잠재적 결함을 탐지하고 Java 및 Python 코드 개선을 위한 제안을 제공합니다.
- [AWS](#)는 소프트웨어 릴리스의 여러 단계를 신속하게 모델링 및 구성하고 소프트웨어 변경 사항을 지속적으로 릴리스하는 CodePipeline 데 필요한 단계를 자동화하도록 지원합니다.
- [Amazon Elastic Container Registry\(Amazon ECR\)](#)는 안전하고 확장성이 있고 신뢰할 수 있는 관리형 컨테이너 이미지 레지스트리 서비스입니다.
- [Amazon Elastic Kubernetes Service \(Amazon EKS\)](#)는 자체 Kubernetes 컨트롤 플레인 또는 노드를 설치하거나 유지 관리할 필요 없이 Kubernetes를 실행하는 데 도움이 됩니다.
- [EventBridgeAmazon](#)은 다양한 소스의 실시간 데이터와 애플리케이션을 연결하는 데 도움이 되는 서버리스 이벤트 버스 서비스입니다. AWS Lambda 함수, API 대상을 사용하는 HTTP 간접 호출 엔드포인트 또는 다른 AWS 계정의 이벤트 버스를 예로 들 수 있습니다.
- [AWS Identity and Access Management\(IAM\)](#)는 누구에게 인증 및 사용 권한이 있는지 제어하여 AWS 리소스에 대한 액세스를 안전하게 관리할 수 있도록 도와줍니다.
- [AWS Security Hub](#)는 AWS의 보안 상태에 대한 포괄적인 보기를 제공합니다. 이를 통해 AWS 환경에서 보안 업계 표준 및 모범 사례를 준수하는지 확인할 수도 있습니다.
- [Amazon Simple Notification Service\(Amazon SNS\)](#)를 사용하면 웹 서버 및 이메일 주소를 포함하여 게시자와 클라이언트 간의 메시지 교환을 조정하고 관리할 수 있습니다.
- [Amazon Simple Storage Service\(S3\)](#)는 원하는 양의 데이터를 저장, 보호 및 검색하는 데 도움이 되는 클라우드 기반 객체 스토리지 서비스입니다.

기타 서비스

- [Helm](#)은 Kubernetes용 오픈소스 패키지 관리자입니다.
- [Apache Maven](#)은 소프트웨어 프로젝트 관리 및 이해 도구입니다.
- [BridgeCrew Checkov](#)는 IaC (Infrastructure as Code) 파일을 스캔하여 보안 또는 규정 준수 문제로 이어질 수 있는 구성 오류를 찾아내는 정적 코드 분석 도구입니다.
- [Aqua Security Trivy](#)는 구성 문제 외에도 컨테이너 이미지, 파일 시스템 및 Git 리포지토리의 취약성에 대한 포괄적인 스캐너입니다.

code

이 패턴의 코드는 리포지토리에서 사용할 수 있습니다. GitHub [aws-codepipeline-devsecops-amazoneks](#)

모범 사례

- 이 솔루션의 모든 단계에서 IAM 엔터티에 대한 최소 권한 원칙이 준수되었습니다. 추가 AWS 서비스 또는 타사 도구를 사용하여 솔루션을 확장하려는 경우 최소 권한 원칙을 따르는 것이 좋습니다.
- Java 애플리케이션이 여러 개 있는 경우 각 애플리케이션에 대해 별도의 CI/CD 파이프라인을 생성하는 것이 좋습니다.
- 모놀리식 애플리케이션을 사용하는 경우 애플리케이션을 최대한 마이크로서비스로 분할하는 것이 좋습니다. 마이크로서비스는 더 유연하고 애플리케이션을 컨테이너로 쉽게 배포할 수 있으며 애플리케이션의 전체 빌드 및 배포에 대한 가시성을 높여줍니다.

에픽

환경 설정

작업	설명	필요한 기술
GitHub 리포지토리를 복제합니다.	리포지토리를 복제하려면 다음 명령을 실행합니다. <pre>git clone https://github.com/aws-samples/aws-codepipeline-devsecops-amazoneks</pre>	앱 개발자, DevOps 엔지니어

작업	설명	필요한 기술
<p>S3 버킷을 생성하고 코드를 업로드합니다.</p>	<ol style="list-style-type: none"> 1. AWS Management Console에 로그인하고 Amazon S3 콘솔을 연 다음 이 솔루션을 배포하려는 AWS 리전에 S3 버킷을 생성합니다. 자세한 내용은 Amazon S3 설명서의 버킷 생성을 참조하십시오. 2. S3 버킷에서 이름이 code인 폴더를 생성합니다. 3. 리포지토리를 복제한 위치로 이동하십시오. 확장명이.zip(cicdstack.zip)인 전체 코드의 압축 버전을 만들고.zip 파일의 유효성을 검사하려면 다음 명령을 순서대로 실행합니다. <p>참고: python 명령이 실패하고 Python을 찾을 수 없다고 표시되면 대신 python3을 사용하십시오.</p> <pre>cd aws-codepipeline-d evsecops-amazoneks python -m zipfile -c cicdstack.zip * python -m zipfile -t cicdstack.zip</pre> 4. 이전에 S3 버킷에서 생성한 코드 폴더에 cicdstack.zip 파일을 업로드합니다. 	<p>AWS DevOps, DevOps 엔지니어, 클라우드 관리자 DevOps</p>

작업	설명	필요한 기술
<p>AWS CloudFormation 스택을 생성합니다.</p>	<ol style="list-style-type: none"> 1. AWS CloudFormation 콘솔을 열고 [스택 생성] 을 선택합니다. 2. 템플릿 지정에서 템플릿 파일 업로드를 선택하고, <code>cf_templates/codecommit_ecr.yaml</code> 파일을 업로드한 후 다음을 선택합니다. 3. 스택 세부 정보 지정에서 스택 이름을 입력하고 다음 입력 파라미터 값을 제공합니다. <ul style="list-style-type: none"> • CodeCommitRepositoryBranchName: 코드가 위치할 브랜치 이름 (기본값은 main) • CodeCommitRepositoryName: 생성할 CodeCommit 리포지토리의 이름. • CodeCommitRepositoryS3Bucket: 코드 폴더를 생성한 S3 버킷의 이름 • CodeCommitRepositoryS3: BucketObjKey code/cicdstack.zip • ECR RepositoryName: 생성될 아마존 ECR 리포지토리의 이름 	<p>AWS DevOps, DevOps</p>

작업	설명	필요한 기술
	<ol style="list-style-type: none"> 4. 다음을 선택하고 스택 구성 옵션의 기본 설정을 사용한 후 다음을 선택합니다. 5. 검토 섹션에서 템플릿과 스택 세부 정보를 확인한 다음 스택 생성을 선택합니다. 그러면 CodeCommit 및 Amazon ECR 리포지토리를 포함한 스택이 생성됩니다. 6. Java CI/CD 파이프라인 설정에 필요한 CodeCommit 및 Amazon ECR 리포지토리의 이름을 기록해 둡니다. 	
CloudFormation 스택 배포를 검증합니다.	<ol style="list-style-type: none"> 1. CloudFormation 콘솔의 스택에서 배포한 CloudFormation 스택의 상태를 확인합니다. 스택의 상태는 생성 완료여야 합니다. 2. 또한 콘솔에서 Amazon ECR이 프로비저닝되고 준비되었는지 CodeCommit 확인합니다. 	DevOps 엔지니어
S3 버킷을 삭제합니다.	이전에 생성한 S3 버킷을 비우고 삭제합니다. 자세한 내용은 Amazon S3 설명서의 버킷 삭제 를 참조하십시오.	AWS DevOps, DevOps

차트 Helm 구성

작업	설명	필요한 기술
<p>Java 애플리케이션의 차트 Helm을 구성하십시오.</p>	<p>1. GitHub 리포지토리를 복제한 위치에서 해당 <code>helm_charts/aws-proserve-java-greeting</code> 폴더로 이동합니다. 이 폴더의 <code>values.dev.yaml</code> 파일에는 Amazon EKS에 컨테이너 배포를 위해 수정할 수 있는 Kubernetes 리소스 구성에 대한 정보가 포함되어 있습니다. AWS 계정 ID, AWS 리전, Amazon ECR 리포지토리 이름을 제공하여 Docker 리포지토리 파라미터를 업데이트하십시오.</p> <pre data-bbox="630 1121 1029 1398"> image: repository: <account-id>.dkr.ecr.<region>.amazonaws.com/<app-ecr-repo-name> </pre> <p>2. Java 포드의 서비스 유형은 <code>LoadBalancer</code> 로 설정되어 있습니다.</p> <pre data-bbox="630 1583 1029 1873"> service: type: LoadBalancer port: 80 targetPort: 8080 path: /hello initialDelaySeconds: 60 </pre>	<p>DevOps</p>

작업	설명	필요한 기술
	<pre data-bbox="630 205 1029 268">periodSeconds: 30</pre> <p data-bbox="630 302 1016 533">다른 서비스(예:NodePort)를 사용하려면 파라미터를 변경할 수 있습니다. 자세한 내용은 Kubernetes 설명서를 참조하십시오.</p> <p data-bbox="591 554 1026 785">3. autoscaling 파라미터를 <code>enabled: true</code>로 변경하여 Kubernetes Horizontal Pod Autoscaler를 활성화할 수 있습니다.</p> <pre data-bbox="630 814 1029 1213">autoscaling: enabled: true minReplicas: 1 maxReplicas: 100 targetCPUUtilizationPercentage: 80 # targetMemoryUtilizationPercentage: 80</pre> <p data-bbox="591 1276 1016 1562">개발, 프로덕션, UAT 또는 QA 환경이 있는 <code>values.<ENV>.yaml</code> <ENV> 파일의 값을 변경하여 Kubernetes 워크로드에 대해 다양한 기능을 활성화할 수 있습니다.</p>	

작업	설명	필요한 기술
<p>차트 Helm에 구문 오류가 있는지 검증하십시오.</p>	<ol style="list-style-type: none"> 1. 터미널에서 다음 명령을 실행하여 Helm v3가 로컬 워크스테이션에 설치되어 있는지 확인합니다. <div style="border: 1px solid #ccc; border-radius: 10px; padding: 5px; margin: 10px 0; text-align: center;"> <code>helm --version</code> </div> <p>헬름 v3가 설치되지 않은 경우 설치하십시오.</p> 2. 터미널에서 차트 Helm 디렉터리 (helm_charts/ aws-proserve-java-greeting) 로 이동하여 다음 명령을 실행합니다. <div style="border: 1px solid #ccc; border-radius: 10px; padding: 5px; margin: 10px 0; text-align: center;"> <code>helm lint . -f values.dev.yaml</code> </div> <p>그러면 차트 Helm에 구문 오류가 있는지 확인할 수 있습니다.</p> 	<p>DevOps 엔지니어</p>

Java CI/CD 파이프라인 설정

작업	설명	필요한 기술
<p>CI/CD 파이프라인을 생성합니다.</p>	<ol style="list-style-type: none"> 1. AWS CloudFormation 콘솔을 열고 [스택 생성] 을 선택합니다. 2. 템플릿 지정에서 템플릿 파일 업로드를 선택하고, cf_templates/build _deployment.yaml 템 	<p>AWS DevOps</p>

작업	설명	필요한 기술
	<p>플릿을 업로드한 후 다음을 선택합니다.</p> <p>3. 스택 세부 정보 지정에서 스택 이름을 지정하고 다음 입력 파라미터 값을 제공합니다.</p> <ul style="list-style-type: none"> • CodeBranchName: 코드가 있는 CodeCommit 리포지토리의 브랜치 이름 • EKSClusterName: EKS 클러스터 이름 (ID 아님) EKSCluster • EKS CodeBuild AppName: 앱 이름 헬름 차트 () aws-proserve-java-greeting • EKS WorkerNodeRole ARN: 아마존 EKS 워커 노드 IAM 역할의 ARN • EKS WorkerNodeRoleName: Amazon EKS 작업자 노드에 할당된 IAM 역할의 이름 • EcrDockerRepository: 코드의 Docker 이미지가 저장되는 Amazon ECR 리포지토리의 이름 • EmailRecipient: 빌드 알림을 전송해야 하는 이메일 주소 • EnvType: 환경 (예: 개발, 테스트 또는 프로덕션) 	

작업	설명	필요한 기술
	<ul style="list-style-type: none"> • SourceRepoName: 코드 가 있는 CodeCommit 리 포지토리의 이름 <ol style="list-style-type: none"> 4. 다음을 선택합니다. 스택 구 성 옵션의 기본 설정을 사용 한 후 다음을 선택합니다. 5. [Review] 섹션에서 AWS CloudFormation 템플릿과 스택 세부 정보를 확인한 후 [Next] 를 선택합니다. 6. 스택 생성을 선택합니다. 7. CloudFormation 스택 배포 중에 파라미터에 제공한 이 메일 주소의 소유자는 SNS 주제를 구독하라는 메시지 를 받게 됩니다. Amazon SNS를 구독하려면 소유자 가 메시지의 링크를 선택해 야 합니다. 8. 스택이 생성되면 스택의 출 력 탭을 열고 EksCodeBu ildkubernetesRoleARN 출 력 키의 ARN 값을 기록합 니다. 이 IAM ARN 값은 나 중에 Amazon EKS 클러스 터에서 워크로드를 배포할 권한을 IAM 역할에 제공할 CodeBuild 때 필요합니다. 	

Security Hub와 Aqua Security 간의 통합 활성화

작업	설명	필요한 기술
Aqua Security 통합을 켜십시오.	<p>이 단계는 Trivy에서 보고한 도커 이미지 취약성 조사 결과를 Security Hub에 업로드하는 데 필요합니다. AWS는 Security Hub 통합을 CloudFormation 지원하지 않으므로 이 프로세스는 수동으로 수행해야 합니다.</p> <ol style="list-style-type: none"> 1. AWS Security Hub 콘솔을 열고 통합으로 이동합니다. 2. Aqua Security를 검색하고 Aqua Security: Aqua Security를 선택합니다. 3. 결과 수락을 선택합니다. 	AWS 관리자, DevOps 엔지니어

헬름 또는 kubectl 명령을 CodeBuild 실행하도록 구성

작업	설명	필요한 기술
Amazon EKS 클러스터에서 헬름 또는 kubectl 명령을 실행할 수 있습니다. CodeBuild	EKS 클러스터에서 Helm 또는 <code>kubectl</code> 명령을 사용하도록 인증하려면 IAM 역할을 에 추가해야 합니다. CodeBuild <code>aws-auth ConfigMap</code> 이 경우 CodeBuild 서비스가 EKS 클러스터에 액세스하고 클러스터에 워크로드를 배포할 수 있도록 생성된 IAM 역할인 IAM 역할의 ARN을 추가합니다. EksCodeBuildkuberol	DevOps

작업	설명	필요한 기술
	<p>1eARN 이는 일회성 작업입니다.</p> <p>중요: 배포 승인 단계가 시작되기 전에 다음 절차를 완료해야 합니다. CodePipeline</p> <ol style="list-style-type: none"> 1. Amazon Linux 또는 MacOS 환경에서 cf_templates/kube_aws_auth_configmap_patch.sh 셸 스크립트를 엽니다. 2. 다음 명령을 실행하여 Amazon EKS 클러스터를 인증합니다. <pre data-bbox="630 928 1029 1129">aws eks --region <aws-region> update-kubeconfig --name <eks-cluster-name></pre> 3. 이전에 기록한 ARN 값 EksCodeBuildkubernetesRoleARN 으로 <rolearn-eks-codebuild-kubect1> 을 대체하여 다음 명령을 사용하여 셸 스크립트를 실행합니다. <pre data-bbox="630 1503 1029 1730">bash cf_templates/kube_aws_auth_configmap_patch.sh <rolearn-eks-codebuild-kubect1></pre> 	

작업	설명	필요한 기술
	aws_auth ConfigMap 가 구성되고 액세스 권한이 부여됩니다.	

CI/CD 파이프라인을 검증합니다.

작업	설명	필요한 기술
CI/CD 파이프라인이 자동으로 시작되는지 확인합니다.	<p>1. Checkov가 Dockerfile 또는 Helm 차트에서 취약점을 탐지하면 파이프라인의 CodeSecurity 스캔 단계는 일반적으로 실패합니다. 하지만 이 예제의 목적은 CI/CD 파이프라인을 통해 취약성을 수정하는 대신 잠재적인 보안 취약성을 식별하는 프로세스 (일반적으로 프로세스) 를 수립하는 것입니다. DevSecOps buildspec /buildspec_secscan .yaml 파일에서 checkov 명령은 --soft-fail 플래그를 사용하여 파이프라인 장애를 방지합니다.</p> <pre data-bbox="630 1472 1029 1848"> - echo -e "\n Running Dockerfile Scan" - checkov -f code/app/Dockerfile --framework dockerfile --soft-fail --summary- position bottom </pre>	DevOps

작업	설명	필요한 기술
	<pre data-bbox="630 205 1026 1024"> - echo -e "\n Running Scan of Helm Chart files" - cp -pv helm_charts/\$EKS_C ODEBUILD_APP_NAME/ values.dev.yaml helm_charts/\$EKS_C ODEBUILD_APP_NAME/ values.yaml - checkov -d helm_charts/\$EKS_C ODEBUILD_APP_NAME --framework helm -- soft-fail --summary- position bottom - rm -rfv helm_charts/\$EKS_C ODEBUILD_APP_NAME/ values.yaml </pre> <p data-bbox="630 1056 1026 1528"> Dockerfile 및 Helm 차트에 대한 취약성이 보고되었을 때 파이프라인이 실패하도록 하려면 checkov 명령에서 --soft-fail 옵션을 제거해야 합니다. 그러면 개발자 또는 엔지니어가 취약성을 수정하고 소스 코드 저장소에 변경 내용을 커밋할 수 있습니다. CodeCommit </p> <p data-bbox="589 1545 1026 1869"> 2. CodeSecurity 스캔과 마찬가지로 빌드 단계에서는 애플리케이션을 Amazon ECR로 푸시하기 전에 Aqua Security Trivy를 사용하여 심각하고 심각한 Docker 이미지 취약성을 식별합니다. </p>	

작업	설명	필요한 기술
	<p>이 예시에서는 도커 이미지 취약성에 대한 파이프라인 장애를 만들지 않았습니다. <code>buildspec/buildspec.yml</code> 파일에서 <code>trivy</code> 명령에는 값이 <code>0</code> 포함된 플래그 <code>--exit-code</code> 가 포함되어 있기 때문에 HIGH 또는 CRITICAL 도커 이미지 취약성이 보고되어도 파이프라인이 실패하지 않습니다.</p> <pre data-bbox="630 810 1029 1604"> - AWS_REGION= \$AWS_DEFAULT_REGION AWS_ACCOUNT_ID=\$AWS_ACCOUNT_ID trivy - d image --no-progress --ignore-unfixed -- exit-code 0 --severit y HIGH,CRITICAL -- format template -- template "@securit yhub/asff.tpl" -o securityhub/report .asff \$AWS_ACCO UNT_ID.dkr.ecr.\$AW S_DEFAULT_REGION.a mazonaws.com/\$IMAG E_REPO_NAME:\$CODEB UILD_RESOLVED_SOUR CE_VERSION </pre> <p>HIGH, CRITICAL 취약성이 보고될 때 파이프라인이 실패하도록 하려면 <code>--exit-code</code> 의 값을 1로 변경하십시오.</p>	

작업	설명	필요한 기술
	<p>그러면 개발자 또는 엔지니어가 취약성을 수정하고 소스 코드 리포지토리에 변경 내용을 커밋할 수 있습니다. CodeCommit</p> <p>3. qua Security Trivy에서 보고한 도커 이미지 취약점은 Security Hub에 업로드됩니다. AWS Security Hub 콘솔에서 조사 결과로 이동합니다. 검색 결과를 레코드 상태 = 활성, 제품 = Aqua Security로 필터링합니다. 여기에는 Security Hub의 도커 이미지 취약성이 나열됩니다. Security Hub에 취약점이 나타나는 데 15분에서 1시간이 걸릴 수 있습니다.</p> <p>를 사용하여 CodePipeline 파이프라인을 시작하는 방법에 대한 자세한 내용은 AWS CodePipeline 설명서에서 파이프라인 시작 CodePipeline, 수동으로 파이프라인 시작 및 일정에 따라 파이프라인 시작을 참조하십시오.</p>	

작업	설명	필요한 기술
<p>배포를 승인하십시오.</p>	<ol style="list-style-type: none"> 1. 빌드 단계가 완료되면 배포 승인 게이트가 있습니다. 검토자 또는 릴리스 관리자는 빌드를 검사하고 모든 요구 사항이 충족되면 승인해야 합니다. 이는 애플리케이션 배포를 위해 지속적 딜리버리를 사용하는 팀에 권장되는 접근 방식입니다. 2. 승인 후 파이프라인은 배포 단계를 시작합니다. 3. 배포 단계가 성공하면 이 단계의 CodeBuild 로그에 애플리케이션의 URL이 제공됩니다. URL을 사용하여 애플리케이션의 준비 상태를 검증할 수 있습니다. 	<p>DevOps</p>
<p>애플리케이션 프로파일링 검증.</p>	<p>배포가 완료되고 애플리케이션 포드가 Amazon EKS에 배포되면 애플리케이션에 구성된 Amazon CodeGuru Profiler 에이전트가 애플리케이션의 프로파일링 데이터 (CPU, 힙 요약, 지연 시간, 병목 현상) 를 Amazon Profiler로 전송하려고 시도합니다. CodeGuru</p> <p>Amazon CodeGuru Profiler는 애플리케이션을 처음 배포할 때 프로파일링 데이터를 시각화하는 데 약 15분이 걸립니다.</p>	<p>AWS DevOps</p>

관련 리소스

- [AWS CodePipeline 설명서](#)
- [AWS에서 Trivy로 이미지 스캔하기 CodePipeline](#) (블로그 게시물)
- [Amazon CodeGuru 프로파일러를 사용한 Java 애플리케이션 개선](#) (블로그 게시물)
- [AWS 보안 조사 결과 형식\(ASFF\) 구문](#)
- [아마존 EventBridge 이벤트 패턴](#)
- [Helm 업그레이드](#)

추가 정보

CodeGuru 프로파일러를 기능 측면에서 AWS X-Ray 서비스와 혼동해서는 안 됩니다. CodeGuru 프로파일러는 병목 현상이나 보안 문제를 일으킬 수 있는 가장 비용이 많이 드는 코드 라인을 식별하고 잠재적 위험이 되기 전에 수정하는 데 선호됩니다. AWS X-Ray 서비스는 애플리케이션 성능 모니터링을 위한 서비스입니다.

이 패턴에서 이벤트 규칙은 기본 이벤트 버스와 연결됩니다. 필요한 경우 패턴을 확장하여 사용자 지정 이벤트 버스를 사용할 수 있습니다.

이 패턴은 CodeGuru Reviewer를 애플리케이션 코드에 대한 정적 애플리케이션 보안 테스트 (SAST) 도구로 사용합니다. 이 파이프라인을 Checkmarx와 같은 다른 도구에도 사용할 수 SonarQube 있습니다. 의 스캔 지침 대신 이러한 도구 중 하나에 해당하는 스캔 설정 지침을 추가할 수 있습니다.

```
buildspec/buildspec_secscan.yaml CodeGuru
```

Amazon ECS 작업 정의를 생성하고 Amazon EFS를 사용하여 EC2 인스턴스에 파일 시스템을 마운트

작성자: Durga Prasad Cheepuri(AWS)

환경: PoC 또는 파일럿

기술: 컨테이너 및 마이크로서
비스, 클라우드 네이티브, 관리
및 거버넌스, 스토리지 및 백업,
웹 및 모바일 앱

AWS 서비스: Amazon ECS,
Amazon EFS

요약

이 패턴은 Amazon Elastic File System(Amazon EFS)을 사용하여 해당 EC2 인스턴스에 파일 시스템을 마운트하면서 Amazon Web Services(AWS) 클라우드의 Amazon Elastic Compute Cloud(Amazon EC2) 인스턴스에서 실행되는 Amazon Elastic Container Service(Amazon ECS) 작업 정의를 생성하는 코드 샘플 및 단계를 제공합니다. Amazon EFS를 사용하는 Amazon ECS 작업은 작업 정의에서 지정된 파일 시스템을 자동으로 마운트하고 AWS 리전의 모든 가용 영역에 있는 작업 컨테이너에서 이러한 파일 시스템을 사용할 수 있도록 합니다.

영구 스토리지 및 공유 스토리지 요구 사항을 충족하기 위해 Amazon ECS와 Amazon EFS를 함께 사용할 수 있습니다. 예를 들어,고가용성을 위해 서로 다른 가용 영역에서 실행 중인 활성 및 대기 ECS 컨테이너 페어가 있는 애플리케이션의 영구 사용자 데이터와 애플리케이션 데이터를 저장하는 데 Amazon EFS를 사용할 수 있습니다. 또한 Amazon EFS를 사용하여 ECS 컨테이너와 분산 작업 워크로드에서 병렬로 액세스할 수 있는 공유 데이터를 저장할 수 있습니다.

Amazon EFS를 Amazon ECS와 함께 사용하려면 작업 정의에 하나 이상의 볼륨 정의를 추가할 수 있습니다. 볼륨 정의에는 Amazon EFS 파일 시스템 ID, 액세스 포인트 ID, 그리고 전송 중 AWS Identity and Access Management(IAM) 인증 또는 전송 계층 보안(TLS) 암호화를 위한 구성이 포함됩니다. 작업 정의 내에서 컨테이너 정의를 사용하여 컨테이너가 실행될 때 마운트되는 작업 정의 볼륨을 지정할 수 있습니다. Amazon EFS 파일 시스템을 사용하는 작업이 실행되면 Amazon ECS는 파일 시스템이 마운트되어 액세스가 필요한 컨테이너에서 사용할 수 있도록 합니다.

사전 조건 및 제한 사항

사전 조건

- 활성 상태의 AWS 계정.
- 가상 프라이빗 네트워크(VPN) 엔드포인트 또는 라우터가 있는 Virtual Private Cloud(VPC)
- (권장)Amazon EFS 액세스 포인트 및 IAM 인증 기능과의 호환성을 위한 [Amazon ECS 컨테이너 에이전트 1.38.0 이상](#)(자세한 내용은 AWS 블로그의 [Amazon EFS 새 소식 - IAM 인증 및 액세스 포인트](#) 포스트를 참조하세요.)

제한 사항

- 1.35.0 이전의 Amazon ECS 컨테이너 에이전트 버전은 EC2 시작 유형을 사용하는 작업의 Amazon EFS 파일 시스템을 지원하지 않습니다.

아키텍처

다음 다이어그램은 Amazon ECS를 사용하여 작업 정의를 생성하고 ECS 컨테이너의 EC2 인스턴스에 Amazon EFS 파일 시스템을 마운트하는 애플리케이션의 예를 보여줍니다.

이 다이어그램은 다음 워크플로를 보여줍니다.

1. Amazon EFS 파일 시스템을 생성합니다.
2. 컨테이너로 작업 정의를 생성합니다.
3. Amazon EFS 파일 시스템을 마운트하도록 컨테이너 인스턴스를 구성합니다. 작업 정의는 볼륨 마운트를 참조하여, 컨테이너 인스턴스가 Amazon EFS 파일 시스템을 사용할 수 있습니다. ECS 작업은 해당 작업이 생성된 컨테이너 인스턴스와 상관없이 동일한 Amazon EFS 파일 시스템에 액세스할 수 있습니다.
4. 작업 정의의 인스턴스 3개를 사용하여 Amazon ECS 서비스를 생성합니다.

기술 스택

- Amazon EC2
- Amazon ECS
- Amazon EFS

도구

- [Amazon EC2](#) – Amazon Elastic Compute Cloud(Amazon EC2)는 AWS 클라우드에서 확장 가능한 컴퓨팅 용량을 제공합니다. Amazon EC2를 사용하여 필요에 따라 많거나 적은 수의 가상 서버를 시작하고 스케일 아웃 또는 스케일 인할 수 있습니다.
- [Amazon ECS](#) – Amazon Elastic Container Service(Amazon ECS)는 클러스터에서 컨테이너를 실행, 중지 및 관리하기 위한 컨테이너 관리 서비스로서 확장성과 속도가 뛰어납니다. AWS Fargate에서 관리하는 서버리스 인프라에서 작업 및 서비스를 실행할 수 있습니다. 또는 인프라에 대한 더 세부적인 제어를 위해 관리하는 EC2 인스턴스의 클러스터에서 작업과 서비스를 실행할 수 있습니다.
- [Amazon EFS](#) – Amazon Elastic File System(Amazon EFS)은 AWS 클라우드 서비스 및 온프레미스 리소스와 함께 사용할 수 있는 간단하고 확장 가능하며 완전 관리형 탄력적인 NFS 파일 시스템을 제공합니다.
- [AWS CLI](#) – AWS Command Line Interface(AWS CLI)는 명령줄 셸에서 명령을 사용하여 AWS 서비스와 상호 작용할 수 있는 오픈 소스 도구입니다. 최소한의 구성으로 명령 프롬프트에서 브라우저 기반 AWS Management Console에서 제공되는 것과 동일한 기능을 구현하는 AWS CLI 명령을 실행할 수 있습니다.

에픽

Amazon EFS 파일 시스템 생성

작업	설명	필요한 기술
AWS Management Console을 사용하여 Amazon EFS 파일 시스템을 생성합니다.	<ol style="list-style-type: none"> 1. Amazon EFS 파일 시스템을 생성하고 컨테이너가 포함된 VPC를 선택합니다. 참고: 다른 VPC를 사용하는 경우 VPC 피어링 연결을 설정하세요. 2. 파일 시스템 ID를 기록합니다. 	AWS DevOps

Amazon EFS 파일 시스템 또는 AWS CLI를 사용하여 Amazon ECS 작업 정의를 생성

작업	설명	필요한 기술
<p>Amazon EFS 파일 시스템을 사용하여 작업 정의를 생성합니다.</p>	<p>새로운 Amazon ECS 콘솔 또는 다음과 같은 구성의 클래식 Amazon ECS 콘솔을 사용하여 작업 정의를 생성합니다.</p> <ul style="list-style-type: none"> • 새 콘솔을 사용하는 경우 애플리케이션용 Amazon EC2 인스턴스를 선택하세요. 클래식 콘솔을 사용하는 경우 시작 유형으로 EC2를 선택하세요. • 볼륨을 추가합니다. 볼륨 이름을 입력하고 볼륨 유형으로 EFS를 선택한 다음 앞서 기록해 둔 파일 시스템 ID를 선택합니다. 루트 디렉터리의 경우 Amazon ECS 컨테이너 호스트에서 호스팅하려는 Amazon EFS 파일 시스템 경로를 선택합니다. 	AWS DevOps
<p>AWS CLI를 사용하여 작업 정의를 생성합니다.</p>	<ol style="list-style-type: none"> 1. 작업 정의의 입력 파라미터 플레이스홀더가 있는 JSON 템플릿을 생성하려면 다음 명령을 실행합니다. <div data-bbox="630 1480 1027 1675" style="border: 1px solid #ccc; border-radius: 10px; padding: 10px; margin: 10px 0;"> <pre>aws ecs register-task-definition --generate-cli-skeleton</pre> </div> 2. JSON 템플릿을 사용하여 작업 정의를 생성하려면 다음 명령을 실행합니다. 	AWS DevOps

작업	설명	필요한 기술
	<pre data-bbox="634 212 1027 447">aws ecs register-task-definition --cli-input-json file://<path_to_your_json_file></pre> <p data-bbox="591 464 1027 972">3. 첨부된 task_definition_parameters.json 파일을 기반으로 JSON 템플릿에 입력 파라미터를 입력합니다. 참고: 입력 파라미터에 대한 자세한 내용은 작업 정의 파라미터 (Amazon ECS 설명서) 및 register-task-definition(AWS CLI 명령 참조) 을 참조하십시오.</p>	

관련 리소스

- [Amazon ECS 작업 정의](#)
- [Amazon EFS 볼륨](#)

첨부

이 문서와 관련된 추가 콘텐츠에 액세스하려면 [attachment.zip](#) 파일의 압축을 풉니다.

Fargate를 사용하여 Amazon ECS에 Java 마이크로서비스 배포

작성자: Vijay Thompson 및 Sandeep Bondugula

환경: PoC 또는 파일럿	소스: 컨테이너	대상: Amazon ECS
R 유형: 해당 없음	기술: 컨테이너 및 마이크로서비스, 웹 및 모바일 앱	AWS 서비스: Amazon ECS

요약

이 패턴은 Fargate를 사용하여 Amazon Elastic Container Service(Amazon ECS)에 컨테이너식 Java 마이크로서비스를 배포하기 위한 지침을 제공합니다. 이 패턴은 컨테이너 관리에 Amazon Elastic Container Registry(Amazon ECR)를 사용하지 않으며, 그 대신에 Docker 허브에서 Docker 이미지를 가져옵니다.

사전 조건 및 제한 사항

사전 조건

- Docker 허브의 기존 Java 마이크로서비스 애플리케이션
- 퍼블릭 Docker 리포지토리
- 활성 상태의 AWS 계정
- Amazon ECS 및 Fargate를 포함한 서비스에 대한 지식
- Docker, Java 및 Spring Boot 프레임워크
- Amazon Relational Database Service(Amazon RDS) 가동 및 실행 중(선택 사항)
- 애플리케이션에 Amazon RDS(선택 사항)가 필요한 경우 Virtual Private Cloud(VPC)

아키텍처

소스 기술 스택

- Java 마이크로서비스(예: Spring Boot에 구현됨) 및 Docker에 배포됨

소스 아키텍처

대상 기술 스택

- Fargate를 사용하여 각 마이크로서비스를 호스팅하는 Amazon ECS 클러스터
- Amazon ECS 클러스터 및 관련 보안 그룹을 호스팅하는 VPC 네트워크
- Fargate를 사용하여 컨테이너를 구동하는 각 마이크로서비스에 대한 클러스터/태스크 정의

대상 아키텍처

도구

도구

- [Amazon ECS](#)를 사용하면 자체 컨테이너 오케스트레이션 소프트웨어를 설치 및 운영하거나, 가상 시스템 클러스터를 관리 및 확장하거나, 가상 시스템에서 컨테이너를 예약할 필요가 없습니다.
- [AWS Fargate](#)를 사용하면 서버 또는 Amazon Elastic Compute Cloud(Amazon EC2) 인스턴스를 관리할 필요 없이 컨테이너를 실행할 수 있습니다. Amazon Elastic Container Service(Amazon ECS)와 함께 사용합니다.
- [Docker](#)는 애플리케이션을 신속하게 구축, 테스트 및 배포하기 위한 소프트웨어 플랫폼입니다. Docker는 소프트웨어를 컨테이너라는 표준화된 단위로 패키징합니다. 컨테이너에는 라이브러리, 시스템 도구, 코드 및 런타임을 포함하여 소프트웨어 실행에 필요한 모든 것이 들어 있습니다.

Docker 코드

다음 Dockerfile은 사용되는 Java Development Kit(JDK) 버전, Java Archive(JAR) 파일이 있는 위치, 노출되는 포트 번호, 애플리케이션의 진입점을 지정합니다.

```
FROM openjdk:11
ADD target/Spring-docker.jar Spring-docker.jar
EXPOSE 8080
ENTRYPOINT ["java", "-jar", "Spring-docker.jar"]
```

에픽

새 태스크 정의 생성

작업	설명	필요한 기술
작업 정의를 생성합니다.	Amazon ECS에서 Docker 컨테이너를 실행하려면 태스크 정의가 필요합니다. https://console.aws.amazon.com/ecs/ 에서 Amazon ECS 콘솔을 열고 태스크 정의를 선택한 다음 새 태스크 정의를 생성합니다. 자세한 내용은 Amazon ECS 설명서 를 참조하세요.	시스템 관리자, 앱 개발자
시작 유형을 선택합니다.	Fargate를 시작 유형으로 선택합니다.	시스템 관리자, 앱 개발자
태스크를 구성합니다.	태스크 이름을 정의하고 적절한 양의 태스크 메모리와 CPU로 애플리케이션을 구성합니다.	시스템 관리자, 앱 개발자
컨테이너를 정의합니다.	컨테이너 이름을 지정합니다. 이미지의 경우 Docker 사이트 이름, 리포지토리 이름 및 Docker 이미지(docker.io/sample-repo/sample-application:sample-tag-name)의 태그 이름을 입력합니다. 애플리케이션의 메모리 제한을 설정하고 허용된 포트에 포트 매핑(8080, 80)을 설정합니다.	시스템 관리자, 앱 개발자
태스크를 생성합니다.	태스크 및 컨테이너 구성이 준비되면 태스크를 생성합니다.	시스템 관리자, 앱 개발자

작업	설명	필요한 기술
	세부적인 지침은 관련 리소스 섹션의 링크를 참조하세요.	

클러스터 구성

작업	설명	필요한 기술
클러스터를 생성하고 구성합니다.	네트워킹 전용을 클러스터 유형으로 선택하고 이름을 구성한 다음 클러스터를 생성하거나 이용할 수 있다면 기존 클러스터를 사용합니다. 자세한 내용은 Amazon ECS 설명서 를 참조하세요.	시스템 관리자, 앱 개발자

태스크 구성

작업	설명	필요한 기술
작업 생성.	클러스터 내에서 새 태스크 실행을 선택합니다.	시스템 관리자, 앱 개발자
시작 유형을 선택합니다.	Fargate를 시작 유형으로 선택합니다.	시스템 관리자, 앱 개발자
태스크 정의, 수정, 플랫폼 버전을 선택합니다.	실행하려는 태스크, 태스크 정의의 수정, 플랫폼 버전을 선택합니다.	시스템 관리자, 앱 개발자
클러스터를 선택합니다.	태스크를 실행하려는 해당 클러스터를 선택합니다.	시스템 관리자, 앱 개발자
태스크 수를 지정합니다.	실행해야 하는 태스크 수를 구성합니다. 두 개 이상의 태스크로 시작하는 경우 태스크 간	시스템 관리자, 앱 개발자

작업	설명	필요한 기술
	에 트래픽을 분산하기 위한 로드 밸런서가 필요합니다.	
태스크 그룹을 지정합니다.	(선택 사항) 관련된 태스크 집합을 태스크 그룹으로 식별할 수 있도록 태스크 그룹 이름을 지정합니다.	시스템 관리자, 앱 개발자
클러스터 VPC, 서브넷 및 보안 그룹 구성합니다.	애플리케이션을 배포하고자 하는 클러스터 VPC와 서브넷을 구성합니다. 인바운드 및 아웃바운드 연결에 대한 액세스를 제공할 수 있도록 보안 그룹(HTTP, HTTPS, 포트 8080)을 생성하거나 업데이트합니다.	시스템 관리자, 앱 개발자
퍼블릭 IP 설정을 구성합니다.	Fargate 태스크에 퍼블릭 IP 주소를 사용할지 여부에 따라 퍼블릭 IP를 활성화하거나 비활성화합니다. 활성화가 기본값이며 권장 옵션입니다.	시스템 관리자, 앱 개발자
설정 검토 및 태스크 생성	설정을 검토한 다음, 태스크 실행을 선택합니다.	시스템 관리자, 앱 개발자

전환

작업	설명	필요한 기술
애플리케이션 URL을 복사합니다.	태스크 상태가 실행 중으로 업데이트되면 태스크를 선택합니다. 네트워킹 섹션에서 퍼블릭 IP를 복사합니다.	시스템 관리자, 앱 개발자

작업	설명	필요한 기술
애플리케이션을 테스트합니다.	브라우저에서 애플리케이션을 테스트할 수 있도록 퍼블릭 IP를 입력합니다.	시스템 관리자, 앱 개발자

관련 리소스

- [Amazon ECS를 위한 Docker 기본 사항](#)(Amazon ECS 설명서)
- [Fargate의 Amazon ECS](#)(Amazon ECS 설명서)
- [태스크 정의 생성](#)(Amazon ECS 설명서)
- [클러스터 생성](#)(Amazon ECS 설명서)
- [기본 서비스 파라미터 구성](#)(Amazon ECS 설명서)
- [네트워크 구성](#)(Amazon ECS 설명서)
- [Amazon ECS에 Java 마이크로서비스 배포](#)(블로그 게시물)

Amazon ECR과 AWS Fargate를 사용하여 Amazon ECS에 Java 마이크로서비스 배포

작성자: Vijay Thompson 및 Sandeep Bondugula

환경: PoC 또는 파일럿	소스: 컨테이너	대상: Amazon ECS
R 유형: 해당 없음	기술: 컨테이너 및 마이크로서비스, 웹 및 모바일 앱	AWS 서비스: Amazon ECS

요약

이 패턴은 Amazon Elastic Container Service(Amazon ECS)에서 Java 마이크로서비스를 컨테이너화된 애플리케이션으로 배포하는 단계를 안내합니다. 또한 패턴은 Amazon Elastic Container Registry(Amazon ECR)를 사용하여 컨테이너를 관리하고 AWS Fargate를 사용하여 컨테이너를 실행합니다.

사전 조건 및 제한 사항

사전 조건

- Docker에서 온프레미스로 실행되는 기존 Java 마이크로서비스 애플리케이션
- 활성 상태의 AWS 계정
- Amazon ECR, Amazon ECS, AWS Fargate 및 AWS Command Line Interface(AWS CLI)에 익숙해야 함
- Java 및 Docker 소프트웨어에 대한 지식

제품 버전

- AWS CLI 버전 1.7 이상

아키텍처

소스 기술 스택

- Java 마이크로서비스(예: Spring Boot를 사용하여 개발) 및 온프레미스로 배포
- Docker

소스 아키텍처

대상 기술 스택

- Amazon ECR
- Amazon ECS
- AWS Fargate

대상 아키텍처

도구

도구

- [Amazon Elastic Container Registry\(Amazon ECR\)](#)는 개발자가 Docker 컨테이너 이미지를 간편하게 저장, 관리 및 배포할 수 있게 해주는 완전 관리형 레지스트리입니다. Amazon ECR은 Amazon ECS와 통합되어 워크플로를 단순화합니다. development-to-production Amazon ECR은 가용성과 확장성이 뛰어난 아키텍처에서 이미지를 호스팅하므로 애플리케이션을 위한 컨테이너를 안정적으로 배포할 수 있습니다. AWS Identity and Access Management(IAM)와 통합하면 각 리포지토리를 리소스 수준에서 제어할 수 있습니다.
- [Amazon Elastic Container Service \(Amazon ECS\)](#)는 Docker 컨테이너를 지원하고 AWS에서 컨테이너화된 애플리케이션을 쉽게 실행하고 확장할 수 있는 확장성이 뛰어난 고성능 컨테이너 오케스트레이션 서비스입니다. Amazon ECS를 사용하면 자체 컨테이너 오케스트레이션 소프트웨어를 설치 및 운영하거나, 가상 시스템 클러스터를 관리 및 확장하거나, 가상 시스템에서 컨테이너를 예약할 필요가 없습니다.
- [AWS Fargate](#)는 서버나 클러스터를 관리할 필요 없이 컨테이너를 실행할 수 있는 Amazon ECS용 컴퓨팅 엔진입니다. AWS Fargate를 사용하면 더 이상 컨테이너를 실행하기 위해 가상 머신의 클러스터를 프로비저닝, 구성 또는 확장할 필요가 없습니다. 따라서 서버 유형을 선택하거나, 클러스터를 조정할 시점을 결정하거나, 클러스터 패킹을 최적화할 필요가 없습니다.
- [Docker는 컨테이너라는](#) 패키지로 애플리케이션을 구축, 테스트 및 제공할 수 있는 플랫폼입니다.

코드

다음은 사용되는 Java 개발 키트 (JDK) 버전, JAR 아카이브 (JAR) 파일이 있는 위치, 노출되는 포트 번호, 애플리케이션의 진입점을 DockerFile 지정합니다.

```
FROM openjdk:8
ADD target/Spring-docker.jar Spring-docker.jar
EXPOSE 8080
ENTRYPOINT ["java", "-jar", "Spring-docker.jar"]
```

에픽

Amazon ECR 리포지토리를 생성합니다.

작업	설명	필요한 기술
리포지토리를 생성합니다.	AWS Management Console에 로그인하고 https://console.aws.amazon.com/ecr/repositories 에서 Amazon ECR 콘솔을 엽니다. 프라이빗 리포지토리를 생성합니다. 자세한 내용은 Amazon ECR 설명서의 프라이빗 리포지토리 생성 을 참조하십시오.	개발자, 시스템 관리자
프로젝트를 업로드합니다.	리포지토리를 열고 푸시 명령 보기를 선택합니다. 표시된 단계에 따라 프로젝트를 업로드합니다. (이 단계는 AWS CLI 버전 1.7 이상을 사용하는 경우에만 작동합니다.) 업로드가 완료되면 리포지토리에 있는 빌드의 URL을 복사합니다. Amazon ECS에서 컨테이너를 생성할 때 이 URL을 사용하게 됩니다.	개발자, 시스템 관리자

컨테이너 생성 및 실행

작업	설명	필요한 기술
작업 정의를 생성합니다.	Amazon ECS에서 Docker 컨테이너를 실행하려면 작업 정의가 필요합니다. https://console.aws.amazon.com/ecs/ 에서 Amazon ECS 콘솔을 열고 작업 정의를 선택한 다음 새 작업 정의를 생성합니다. 자세한 내용은 Amazon ECS 설명서의 작업 정의 생성 을 참조하십시오.	개발자, 시스템 관리자
시작 유형을 선택합니다.	Fargate를 시작 유형으로 선택하십시오.	개발자, 시스템 관리자
작업 구성	작업 이름을 정의하고 적절한 양의 작업 메모리와 CPU로 애플리케이션을 구성합니다.	개발자, 시스템 관리자
컨테이너를 정의합니다.	이름, Amazon ECR 리포지토리의 URL, 메모리 제한, 포트 매핑을 제공하여 컨테이너를 추가합니다. 포트 8080 및 80은 포트 매핑용으로 구성됩니다. 애플리케이션 요구 사항에 따라 나머지 설정을 구성하십시오.	개발자, 시스템 관리자
작업을 생성합니다.	태스크 및 컨테이너 구성이 준비되면 태스크를 생성합니다. 자세한 지침은 관련 리소스 섹션의 링크를 참조하십시오.	개발자, 시스템 관리자

Amazon ECS 클러스터를 생성하고 서비스를 구성합니다.

작업	설명	필요한 기술
클러스터를 생성 또는 선택합니다.	Amazon ECS 클러스터는 작업 또는 서비스의 논리적 그룹을 제공합니다. 기존 클러스터를 사용하거나 새 클러스터를 생성할 수 있습니다. 새 클러스터를 생성하려는 경우 요구 사항에 따라 클러스터 유형을 선택합니다. 이 예에서는 네트워킹 클러스터를 선택했습니다. 클러스터 이름을 제공하고 Fargate 작업에 사용할 새 Virtual Private Cloud(VPC)를 생성할지 여부를 선택합니다.	개발자, 시스템 관리자
서비스를 생성합니다.	클러스터 내에서 서비스 생성을 선택합니다.	개발자, 시스템 관리자
시작 유형을 선택합니다.	Fargate를 시작 유형으로 선택하십시오.	개발자, 시스템 관리자
작업 정의, 수정, 플랫폼 버전을 선택합니다.	실행하려는 작업을 선택하고 이어서 작업 정의의 수정 및 플랫폼 버전을 선택합니다.	개발자, 시스템 관리자
클러스터를 선택합니다.	드롭다운 목록에서 서비스를 생성할 클러스터를 선택합니다.	개발자, 시스템 관리자
서비스 이름을 입력합니다.	생성 중인 서비스의 고유 이름을 입력합니다.	개발자, 시스템 관리자
작업 수를 지정합니다.	서비스 시작 시 실행해야 하는 작업 수를 구성합니다. 두 개 이상의 작업으로 시작하는 경우	개발자, 시스템 관리자

작업	설명	필요한 기술
	작업 밸런싱을 위해 로드 밸런서가 필요합니다. 구성할 수 있는 최소 작업 수는 1개입니다.	
최소 및 최대 정상 비율을 설정합니다.	애플리케이션의 최소 및 최대 정상 백분율을 구성하거나 제공된 기본 옵션을 그대로 사용하십시오.	개발자, 시스템 관리자
배포 설정을 구성합니다.	요구 사항에 따라 배포 유형을 선택합니다. 롤링 업데이트 또는 블루/그린 배포를 선택할 수 있습니다.	개발자, 시스템 관리자
클러스터 VPC, 서브넷 및 보안 그룹을 구성합니다.	클러스터 VPC, 애플리케이션을 배포할 서브넷, 인바운드/아웃바운드 연결에 대한 액세스를 제공하는 보안 그룹(HTTP, HTTPS, 포트 8080)을 구성합니다.	개발자, 시스템 관리자
퍼블릭 IP 설정을 구성합니다.	Fargate 작업에 퍼블릭 IP 주소를 사용할지 여부에 따라 퍼블릭 IP를 활성화하거나 비활성화합니다.	개발자, 시스템 관리자
부하 분산을 구성합니다.	둘 이상의 작업으로 서비스를 시작하는 경우 로드 밸런서를 구성하십시오. 서비스를 시작하기 전에 로드 밸런서와 대상 그룹을 만들어야 합니다.	개발자, 시스템 관리자

작업	설명	필요한 기술
자동 크기 조정을 구성합니다.	Amazon ECS Service Auto Scaling을 사용하여 요구 사항에 따라 원하는 작업 수를 늘리거나 줄일 수 있도록 서비스를 구성합니다.	개발자, 시스템 관리자
설정을 검토한 다음 생성을 선택합니다.	서비스 설정을 검토한 후 서비스 생성을 선택합니다.	개발자, 시스템 관리자

전환

작업	설명	필요한 기술
애플리케이션을 테스트합니다.	작업이 배포될 때 생성되는 퍼블릭 DNS를 사용하여 애플리케이션을 테스트하십시오. 애플리케이션에 로드 밸런서가 있는 경우 이를 사용하여 애플리케이션을 테스트한 다음 전환하십시오.	개발자, 시스템 관리자

관련 리소스

- [Amazon ECS를 위한 Docker 기본 사항](#)(Amazon ECS 설명서)
- [AWS Fargate의 Amazon ECS](#)(Amazon ECS 설명서)
- [프라이빗 리포지토리 생성](#) (Amazon ECR 설명서)
- [작업 정의 생성](#)(Amazon ECS 설명서)
- [컨테이너 정의](#)(Amazon ECS 설명서)
- [클러스터 생성](#)(Amazon ECS 설명서)
- [기본 서비스 파라미터 구성](#)(Amazon ECS 설명서)
- [네트워크 구성](#)(Amazon ECS 설명서)
- [로드 밸런서를 사용하도록 서비스를 구성](#)(Amazon ECS 설명서)

- [Auto Scaling을 사용하도록 서비스를 구성](#)(Amazon ECS 설명서)

Amazon ECR 및 로드 밸런싱을 사용하여 Amazon ECS에 Java 마이크로서비스 배포

작성자: Durga Prasad Cheepuri(AWS)

R 유형: 해당 없음	소스: Java	대상: Amazon ECS
작성자: AWS	환경: PoC 또는 파일럿	기술: 웹 및 모바일 앱, 컨테이너 및 마이크로서비스
AWS 서비스: Amazon ECS		

요약

이 패턴은 애플리케이션을 더 쉽게 확장하고 더 빠르게 개발할 수 있도록 Amazon Elastic Container Service(Amazon ECS)에 컨테이너화된 Java 마이크로서비스 아키텍처를 배포하는 단계를 설명합니다. 이는 혁신을 가능하게 하고 새로운 기능을 가속화하는 time-to-market 데 도움이 됩니다.

또한 이 패턴은 Amazon Elastic Container Registry (Amazon ECR) 를 사용하여 Docker 기반 컨테이너를 저장 및 관리하고, Python 스크립트가 포함된 CloudFormation AWS 템플릿을 사용하여 인프라 설정을 자동화합니다. 이 패턴은 AWS Compute 블로그에 게시된 [Amazon Elastic 컨테이너 서비스에 Java 마이크로서비스를 배포하기](#) 게시한 게시물을 기반으로 합니다.

마이크로서비스는 소프트웨어 개발에 대한 아키텍처 및 조직적 접근 방식을 제공합니다. 여기서 소프트웨어는 잘 정의된 애플리케이션 프로그래밍 인터페이스(API)를 통해 통신하는 작고 독립적인 서비스로 구성됩니다. 독립된 소규모 팀이 이러한 서비스를 소유합니다.

Amazon ECS는 확장성이 뛰어난 고성능 컨테이너 오케스트레이션 서비스입니다. Docker 컨테이너를 지원하므로 AWS에서 컨테이너식 애플리케이션을 빠르게 실행하고 규모를 조정할 수 있습니다. Amazon ECS를 사용하면 더 이상 컨테이너 오케스트레이션 소프트웨어를 설치 및 운영하거나, 가상 머신(VM) 클러스터를 관리 및 규모 조정하거나, 해당 VM에서 컨테이너를 예약할 필요가 없습니다.

간단한 API 호출을 통해 Docker 지원 애플리케이션을 시작 및 중지하고, 요청의 전체 상태를 쿼리하고, AWS Identity and Access Management (IAM) 역할, 보안 그룹, 로드 밸런서, Amazon Events, AWS 템플릿 CloudWatch CloudFormation , AWS 로그와 같은 다양한 기본 기능에 액세스할 수 있습니다. CloudTrail

사전 조건 및 제한 사항

사전 조건

- 활성 상태의 AWS 계정
- Java 마이크로서비스 소스 코드, Java 개발 키트 버전 1.7 이상 포함
- 계정 내 사용자의 액세스 키 및 비밀 액세스 키
- AWS Command Line Interface(AWS CLI)
- Java, Python용 AWS 소프트웨어 개발 키트 (SDK) (Boto3), Docker 소프트웨어
- 이전 기술 사용에 대한 지식
- Amazon ECS, AWS CloudFormation, Elastic Load Balancing과 같은 AWS 서비스에 대한 지식

아키텍처

소스 기술 스택

- Java로 구현되고 온프레미스 환경의 Apache Tomcat에 배포된 마이크로서비스

대상 기술 스택

- 클라이언트 요청을 검사하는 Application Load Balancer. 로드 밸런서는 라우팅 규칙에 따라 상태와 일치하는 대상 그룹의 인스턴스 및 포트로 요청을 전달합니다.
- 각 마이크로서비스의 대상 그룹. 대상 그룹은 해당 서비스에서 사용 가능한 컨테이너 인스턴스를 등록하는 데 사용됩니다. 각 대상 그룹에는 경로가 있으므로 특정 마이크로서비스에 대한 경로를 직접 호출하면 올바른 대상 그룹에 매핑됩니다. 이를 통해 하나의 Application Load Balancer를 사용하여 경로에서 액세스하는 모든 마이크로서비스를 제공할 수 있습니다. 예를 들어 `https://owner/ *`는 소유자 마이크로서비스를 매핑하여 전달합니다.
- 각 마이크로서비스의 컨테이너를 호스팅하는 Amazon ECS 클러스터입니다.
- Amazon ECS 클러스터 및 관련 보안 그룹을 호스팅하기 위한 Amazon Virtual Private Cloud(VPC) 네트워크입니다.
- Amazon Elastic Container Registry (Amazon ECR) 리포지토리.
- Amazon ECS 클러스터의 인스턴스에서 컨테이너를 가동시키는 각 마이크로서비스의 서비스 또는 작업 정의.

대상 아키텍처

도구

- [Amazon ESC](#) - Amazon ESC에서는 간단한 API 직접 호출을 사용하여 컨테이너 기반 애플리케이션을 시작 및 중지할 수 있고 중앙 집중식 서비스를 사용하여 클러스터 상태를 확인할 수 있으며 다수의 친숙한 Amazon Elastic Compute Cloud(Amazon EC2) 기능에 액세스할 수 있습니다.
- [Amazon ECR](#) - Amazon Elastic Container Registry(Amazon ECR)는 개발자가 Docker 컨테이너 이미지를 쉽게 저장, 관리, 배포할 수 있는 완전 관리형 레지스트리입니다. Amazon ECR은 Amazon ECS와 통합되어 워크플로를 단순화합니다. development-to-production Amazon ECR은 가용성과 확장성이 뛰어난 아키텍처에서 이미지를 호스팅하므로 애플리케이션을 위한 컨테이너를 안정적으로 배포할 수 있습니다. AWS Identity and Access Management(IAM)와 통합하면 각 리포지토리를 리소스 수준에서 제어할 수 있습니다.

에픽

AWS CloudFormation 템플릿을 생성하여 Java 마이크로서비스를 호스팅할 Amazon ECS 클러스터를 설정합니다.

작업	설명	필요한 기술
Amazon EC2 Linux 인스턴스를 프로비저닝하고, Docker를 설치하고, 각 마이크로서비스에 대한 Docker 파일을 생성합니다.		Ops
Amazon ECR에 도커 이미지를 설정합니다.	(선택 사항) 푸시할 이미지의 Dockerfile이 있는 경우 이미지를 빌드하고 새 리포지토리로 태그를 지정합니다. 각 마이크로서비스에 대해 동일한 작업을 수행하십시오. 새로 태그가 지정된 이미지를 리포지토리에 푸시합니다.	Ops

작업	설명	필요한 기술
AWS CloudFormation 템플릿을 생성합니다.	AWS CloudFormation 템플릿을 생성하여 가상 사설 클라우드 (VPC), Amazon ECS 클러스터 및 Amazon RDS (Amazon RDS) 를 프로비저닝합니다.	Ops

AWS 서비스 프로비저닝

작업	설명	필요한 기술
이전에 만든 CloudFormation 템플릿을 사용하여 AWS 인프라를 생성합니다.	https://github.com/aws-labs/amazon-ecs-java-microservices-PetClinic/blob/master/2_ECS_Java_Spring_Microservices/setup.py 의 Python 스크립트를 사용하여 이전에 만든 AWS 템플릿을 호출하십시오. CloudFormation 이 템플릿은 대상 환경에 필요한 AWS 인프라를 생성합니다.	Ops
Amazon ECR 리포지토리, 작업, 서비스, Application Load Balancer, 대상 그룹을 생성합니다.	Python 스크립트는 AWS CloudFormation 템플릿의 출력을 읽고 BOTO3 API 호출을 사용하여 Amazon ECR 리포지토리, 작업, 서비스, 애플리케이션 로드 밸런서 및 대상 그룹을 생성합니다.	Ops

관련 리소스

- [Amazon Elastic Container Service에 Java Microservices 배포](#) (AWS Compute 블로그 게시물)

- [Python 스크립트](#)
- [Amazon ECS 설명서](#)
- [Amazon ECS용 도커 기본 사항](#)
- [AWS SDK for Python](#)
- [Amazon VPC 설명서](#)
- [Amazon ECR 설명서](#)

Amazon EKS와 Amazon S3의 차트 Helm 리포지토리를 사용하여 Kubernetes 리소스 및 패키지 배포

작성자: Sagar Panigrahi(AWS)

환경: PoC 또는 파일럿

기술: 컨테이너 및 마이크로서
비스, DevOps

AWS 서비스: Amazon EKS

요약

이 패턴은 복잡성과 관계없이 Kubernetes 애플리케이션을 효율적으로 관리하는 데 도움이 됩니다. 이 패턴은 Helm을 기존의 지속적 통합 및 지속적 전달(CI/CD) 파이프라인에 통합하여 Kubernetes 클러스터에 애플리케이션을 배포합니다. Helm은 Kubernetes 애플리케이션을 관리하는데 도움이 되는 Kubernetes 패키지 관리자입니다. 차트 Helm은 복잡한 Kubernetes 애플리케이션을 정의, 설치 및 업그레이드하는 데 도움이 됩니다. 차트를 버전화하여 Helm 리포지토리에 저장할 수 있어 정전 시 평균 복원 시간(MTTR)을 개선할 수 있습니다.

이 패턴은 Kubernetes 클러스터에 대해 Amazon Elastic Kubernetes Service(Amazon EKS)를 사용합니다. Amazon Simple Storage Service(S3)를 차트 Helm 리포지토리로 사용하므로, 조직 전체의 개발자가 차트를 중앙에서 관리하고 액세스할 수 있습니다.

사전 조건 및 제한 사항

사전 조건

- Virtual Private Cloud(VPC)가 있는 활성 Amazon Web Services(AWS) 계정
- Amazon EKS 클러스터
- Amazon EKS 클러스터 내에 설정되고 워크로드를 처리할 준비가 된 작업자 노드
- 클라이언트 머신의 대상 클러스터에 대해 Amazon EKS kubeconfig 파일을 구성하기 위한 Kubectl
- S3 버킷을 생성하기 위한 AWS Identity and Access Management(IAM) 액세스
- 클라이언트 머신에서 Amazon S3에 대한 IAM(프로그래밍 방식 또는 역할) 액세스
- 소스 코드 관리 및 CI/CD 파이프라인

제한 사항

- 현재 사용자 지정 리소스 정의(CRD)의 업그레이드, 삭제 또는 관리는 지원되지 않습니다.
- CRD를 참조하는 리소스를 사용하는 경우 CRD를 별도로(차트 외부에) 설치해야 합니다.

제품 버전

- Helm v3.6.3

아키텍처

대상 기술 스택

- Amazon EKS
- Amazon VPC
- Amazon S3
- 소스 코드 관리
- Helm
- Kubectl

대상 아키텍처

자동화 및 규모 조정

- AWS를 사용하여 인프라 생성을 자동화할 CloudFormation 수 있습니다. 자세한 내용은 Amazon EKS CloudFormation 설명서에서 [AWS를 사용하여 Amazon EKS 리소스 생성을](#) 참조하십시오.
- Helm을 기존 CI/CD 자동화 도구에 통합하여 차트 Helm의 패키징 및 버전 관리를 자동화할 예정입니다(이 패턴의 범위를 벗어남).
- GitVersion 또는 Jenkins 빌드 번호를 사용하여 차트 버전 관리를 자동화할 수 있습니다.

도구

도구

- [Amazon EKS](#) – Amazon Elastic Kubernetes Service(Amazon EKS)는 자체 Kubernetes 컨트롤 플레인 구축하거나 유지 관리할 필요 없이 AWS에서 Kubernetes를 실행하기 위한 관리형 서비스입니다.

다. Kubernetes는 컨테이너화된 애플리케이션의 배포, 조정 및 관리 자동화를 위한 오픈 소스 시스템입니다.

- [Helm](#) - Helm은 Kubernetes 클러스터에 애플리케이션을 설치하고 관리하는 데 도움이 되는 Kubernetes용 패키지 관리자입니다.
- [Amazon S3](#) – Amazon Simple Storage Service(S3)는 인터넷에 대한 스토리지입니다. Amazon S3를 사용하면 인터넷을 통해 언제 어디서든 원하는 양의 데이터를 저장하고 검색할 수 있습니다.
- [Kubectl](#) - Kubectl은 Kubernetes 클러스터에 대해 명령을 실행하기 위한 명령줄 유틸리티입니다.

코드

예제 코드가 첨부되어 있습니다.

에픽

Helm 구성 및 초기화

작업	설명	필요한 기술
Helm 클라이언트를 설치합니다.	로컬 시스템에 Helm 클라이언트를 다운로드하고 설치하려면 다음 명령을 사용하십시오. <pre>sudo curl https://raw.githubusercontent.com/helm/helm/master/scripts/get-helm-3 bash</pre>	DevOps 엔지니어
Helm 설치를 검사합니다.	Helm이 Amazon EKS 클러스터 내의 Kubernetes API 서버와 통신할 수 있는지 확인하려면 <code>helm version</code> 을 실행하십시오.	DevOps 엔지니어

Amazon EKS 클러스터에 차트 Helm 생성 및 설치

작업	설명	필요한 기술
NGINX용 차트 Helm을 만드십시오.	클라이언트 머신에 이름이 my-nginx로 지정된 차트 Helm을 만들려면 <code>helm create my-nginx</code> 를 실행하십시오.	DevOps 엔지니어
차트의 구조를 검토하십시오.	차트의 구조를 검토하려면 <code>tree my-nginx/</code> 명령을 실행합니다.	DevOps 엔지니어
차트에서 서비스 계정 생성을 비활성화합니다.	<code>values.yaml</code> 의 <code>serviceAccount</code> 섹션 아래에서 <code>create</code> 키를 <code>false</code> 로 설정합니다. 이 패턴에 대한 서비스 계정을 만들 필요가 없으므로 이 기능은 꺼져 있습니다.	DevOps 엔지니어
수정된 차트에 구문 오류가 있는지 검증(린트)합니다.	대상 클러스터에 차트를 설치하기 전에 차트에 구문 오류가 있는지 확인하려면 <code>helm lint my-nginx/</code> 를 실행하십시오.	DevOps 엔지니어
차트를 설치하여 Kubernetes 리소스를 배포합니다.	다음 명령을 사용하여 차트 Helm 설치 관리자를 실행합니다. <pre>helm install --name my-nginx-release --debug my-nginx/ --namespace helm-space</pre> 선택적 <code>debug</code> 플래그는 설치 중에 모든 디버그 메시지를 출력합니다. <code>namespace</code> 플래	DevOps 엔지니어

작업	설명	필요한 기술
	그는 이 차트의 리소스 부분이 생성될 네임스페이스를 지정합니다.	
Amazon EKS 클러스터에서 리소스를 검토하십시오.	<p>helm-space 네임스페이스에서 차트 Helm의 일부로 생성된 리소스를 검토하려면 다음 명령을 사용하십시오.</p> <pre>kubectl get all -n helm-space</pre>	DevOps 엔지니어

이전 버전의 Kubernetes 애플리케이션으로 롤백

작업	설명	필요한 기술
릴리스를 수정하고 업그레이드하십시오.	<p>values.yaml 의 차트를 수정하려면 replicaCount 값을 2로 변경합니다. 그런 다음 다음 명령을 실행하여 이미 설치된 릴리스를 업그레이드합니다.</p> <pre>helm upgrade my-nginx-release my-nginx/ --namespace helm-space</pre>	DevOps 엔지니어
Helm 릴리스 기록을 검토하십시오.	<p>Helm을 사용하여 설치한 특정 릴리스의 모든 수정 버전을 나열하려면 다음 명령을 실행하십시오.</p> <pre>helm history my-nginx-release</pre>	DevOps 엔지니어

작업	설명	필요한 기술
특정 개정에 대한 세부 정보를 검토합니다.	작동 버전으로 전환하거나 롤백하기 전에 그리고 수정 버전을 설치하기 전에 추가 검증 단계를 수행하려면 다음 명령을 사용하여 각 수정 버전에 전달된 값을 확인하십시오. <pre>helm get --revision=2 my-nginx-release</pre>	DevOps 엔지니어
이전 버전으로 롤백합니다.	이전 버전으로 롤백하려면 다음 명령을 사용합니다. <pre>helm rollback my-nginx-release 1</pre> <p>이 예제는 개정 번호 1로 롤백하는 것입니다.</p>	DevOps 엔지니어

S3 버킷을 Helm 리포지토리로 초기화합니다.

작업	설명	필요한 기술
차트 Helm을 위한 S3 버킷을 생성합니다.	고유한 S3 버킷을 생성합니다. 버킷에서 이름이 charts인 폴더를 생성합니다. 이 패턴의 예시는 <code>s3://my-helm-charts/charts</code> 를 대상 차트 리포지토리로 사용합니다.	클라우드 관리자
Amazon S3용 Helm 플러그인을 설치합니다.	Helm-s3 플러그인을 클라이언트 머신에 설치하려면 다음 명령을 사용합니다.	DevOps 엔지니어

작업	설명	필요한 기술
	<pre>helm plugin install https://github.com/ hypnoglows/helm-s3.git --version 0.10.0</pre> <p>참고: Helm V3 지원은 플러그인 버전 0.9.0 이상에서 사용할 수 있습니다.</p>	
<p>Amazon S3 Helm 리포지토리를 초기화합니다.</p>	<p>대상 폴더를 Helm 리포지토리로 초기화하려면 다음 명령을 사용합니다.</p> <pre>helm s3 init s3://my-helm-charts/charts</pre> <p>이 명령은 대상에 <code>index.yaml</code> 파일을 생성하여 해당 위치에 저장된 모든 차트 정보를 추적합니다.</p>	<p>DevOps 엔지니어</p>
<p>Amazon S3 리포지토리를 Helm에 추가합니다.</p>	<p>클라이언트 머신에서 리포지토리를 추가하려면 다음 명령을 사용합니다.</p> <pre>helm repo add my-helm-charts s3://my-helm-charts/charts</pre> <p>이 명령은 Helm 클라이언트 컴퓨터의 대상 리포지토리에 별칭을 추가합니다.</p>	<p>DevOps 엔지니어</p>
<p>리포지토리 목록을 검토하십시오.</p>	<p>Helm 클라이언트 컴퓨터의 리포지토리 목록을 보려면 <code>helm repo list</code>를 실행하십시오.</p>	<p>DevOps 엔지니어</p>

Amazon S3 Helm 리포지토리에 차트를 패키징하고 저장합니다.

작업	설명	필요한 기술
차트를 패키징합니다.	<p>생성한 my-nginx 차트를 패키징하려면 <code>helm package ./my-nginx/</code> 를 실행하십시오. 이 명령은 my-nginx 차트 폴더의 모든 내용을 아카이브 파일로 패키징하며, 아카이브 파일은 <code>Chart.yaml</code> 파일에 언급된 버전 번호를 사용하여 이름이 지정됩니다.</p>	DevOps 엔지니어
Amazon S3 Helm 리포지토리에 패키지를 저장합니다.	<p>Amazon S3의 Helm 리포지토리에 패키지를 업로드하려면 올바른 <code>.tgz</code> 파일 이름을 사용하여 다음 명령을 실행합니다.</p> <pre>helm s3 push ./my-nginx-0.1.0.tgz my-helm-charts</pre>	DevOps 엔지니어
차트 Helm을 검색하십시오.	<p>차트를 로컬에서 표시하고 Amazon S3의 Helm 리포지토리에서 모두 표시되는지 확인하려면 다음 명령을 실행합니다.</p> <pre>helm search repo my-nginx</pre>	DevOps 엔지니어

차트 수정, 버전 지정, 패키징

작업	설명	필요한 기술
차트를 수정하고 패키징합니다.	<p>values.yaml 에서 replicaCount 값을 1로 설정합니다. 그런 다음 helm package ./my-nginx/ 를 실행하여 차트를 패키징하며, 이번에는 Chart.yaml 의 버전을 0.1.1로 변경합니다.</p> <p>버전 관리는 CI/CD 파이프라인의 Jenkins 빌드 번호와 같은 GitVersion 도구를 사용한 자동화를 통해 업데이트되는 것이 이상적입니다. 버전 번호 자동화는 이 패턴의 범위를 벗어납니다.</p>	DevOps 엔지니어
Amazon S3의 Helm 리포지토리에 새 버전을 푸시합니다.	<p>0.1.1 버전의 새 패키지를 Amazon S3의 my-helm-charts Helm 리포지토리로 푸시하려면 다음 명령을 실행합니다.</p> <pre>helm s3 push ./my-nginx-0.1.1.tgz my-helm-charts</pre>	DevOps 엔지니어

Amazon S3 Helm 리포지토리에서 차트 검색 및 설치

작업	설명	필요한 기술
my-nginx 차트의 모든 버전을 검색하십시오.	사용 가능한 차트 버전을 모두 보려면 --versions 플래그	DevOps 엔지니어

작업	설명	필요한 기술
	<p>를 사용하여 다음 명령을 실행합니다.</p> <pre>helm search repo my-nginx --versions</pre> <p>플래그가 없는 경우 Helm은 기본적으로 가장 최근에 업로드된 차트 버전을 표시합니다.</p>	
<p>Amazon S3 Helm 리포지토리에서 차트를 설치하십시오.</p>	<p>이전 작업의 검색 결과에는 my-nginx 차트의 여러 버전이 표시됩니다. Amazon S3 Helm 리포지토리에서 새 버전(0.1.1)을 설치하려면 다음 명령을 사용합니다.</p> <pre>helm upgrade my-nginx-release my-helm-charts/my-nginx --version 0.1.1 --namespace helm-space</pre>	<p>DevOps 엔지니어</p>

관련 리소스

- [Helm 설명서](#)
- [helm-s3 플러그인\(MIT 라이선스\)](#)
- [Helm 클라이언트 바이너리](#)
- [Amazon EKS 설명서](#)

첨부

이 문서와 관련된 추가 콘텐츠에 액세스하려면 [attachment.zip](#) 파일의 압축을 풉니다.

컨테이너 이미지로 Lambda 함수 배포

작성자: Ram Kandaswamy(AWS)

환경: 프로덕션

기술: 컨테이너 및 마이크로서비스, 클라우드 네이티브, 소프트웨어 개발 및 테스트, 서버리스

워크로드: 기타 모든 워크로드

AWS 서비스: Amazon EC2 컨테이너 레지스트리, AWS Lambda

요약

AWS Lambda는 컨테이너 이미지를 배포 모델로 지원합니다. 이 패턴은 컨테이너 이미지를 통해 Lambda 함수를 배포하는 방법을 보여줍니다.

Lambda는 서버를 프로비저닝하거나 관리하지 않고도 사실상 모든 유형의 애플리케이션이나 백엔드 서비스에 대한 코드를 실행할 수 있는 서버리스 이벤트 기반 컴퓨팅 서비스입니다. Lambda 함수에 대한 컨테이너 이미지 지원을 통해 애플리케이션 아티팩트에 최대 10GB의 스토리지와 친숙한 컨테이너 이미지 개발 도구를 사용할 수 있는 이점을 얻을 수 있습니다.

이 패턴의 예제에서는 Python을 기본 프로그래밍 언어로 사용하지만 Java, Node.js, Go와 같은 다른 언어를 사용할 수 있습니다. 패턴은 AWS를 CodeCommit 소스로 사용하지만 GitHub, 비트버킷 또는 아마존 심플 스토리지 서비스 (Amazon S3) 를 사용할 수도 있습니다.

사전 조건 및 제한 사항

사전 조건

- Amazon Elastic Container Registry(Amazon ECR) 활성화됨
- 애플리케이션 코드
- 런타임 인터페이스 클라이언트와 최신 버전의 Python이 포함된 Docker 이미지

제한 사항

- 지원되는 최대 이미지 크기는 10GB입니다.
- Lambda 기반 컨테이너 배포의 최대 런타임은 15분입니다.

아키텍처

대상 기술 스택

- Python 프로그래밍 언어
- AWS CodeBuild
- AWS CodeCommit
- 도커 이미지
- Amazon ECR
- AWS Identity and Access Management(IAM)
- AWS Lambda
- 아마존 CloudWatch 로그

대상 아키텍처

1. 를 사용하여 리포지토리를 생성하고 애플리케이션 코드를 CodeCommit 커밋합니다.
2. 소스 공급자로 사용되는 변경 사항이 CodeCommit 적용되면 CodeBuild 프로젝트가 시작됩니다.
3. CodeBuild 프로젝트는 Docker 이미지를 생성하고 Amazon ECR에 이미지를 게시합니다.
4. Amazon ECR에서 이미지를 사용하여 Lambda 함수를 생성합니다.

자동화 및 규모 조정

이 패턴은 AWS CloudFormation, AWS 클라우드 개발 키트 (AWS CDK) 또는 SDK의 API 작업을 사용하여 자동화할 수 있습니다. Lambda는 요청 수에 따라 자동으로 규모를 조정할 수 있으며, 동시성 파라미터를 사용하여 Lambda를 조정할 수 있습니다. 자세한 내용은 [Lambda 설명서](#)를 참조하세요.

도구

서비스

- [AWS CloudFormation Designer](#)는 템플릿을 보고 편집하는 데 도움이 되는 통합 JSON 및 YAML CloudFormation 편집기를 제공합니다.
- [CodeBuildAWS](#)는 소스 코드를 컴파일하고, 단위 테스트를 실행하고, 배포할 준비가 된 아티팩트를 생성하는 데 도움이 되는 완전관리형 빌드 서비스입니다.
- [CodeCommitAWS](#)는 자체 소스 제어 시스템을 관리할 필요 없이 Git 리포지토리를 비공개로 저장하고 관리할 수 있는 버전 제어 서비스입니다.
- [CodeStarAWS](#)는 [AWS에서](#) 소프트웨어 개발 프로젝트를 만들고, 관리하고, 작업할 수 있는 클라우드 기반 서비스입니다. 이 패턴의 경우 AWS CodeStar 또는 다른 개발 환경을 사용할 수 있습니다.
- [Amazon Elastic Container Registry\(Amazon ECR\)](#)는 안전하고 확장 가능하고 신뢰할 수 있는 관리형 컨테이너 이미지 레지스트리 서비스입니다.
- [AWS Lambda](#)는 서버를 프로비저닝하거나 관리할 필요 없이 코드를 실행하는 데 도움이 되는 컴퓨팅 서비스입니다. 필요할 때만 코드를 실행하며 자동으로 확장이 가능하므로 사용한 컴퓨팅 시간만큼만 비용을 지불합니다.

기타 도구

- [Docker](#)는 운영 체제 수준의 가상화를 사용하여 컨테이너에 소프트웨어를 제공하는 서비스형 플랫폼(PaaS) 제품 세트입니다.

모범 사례

- 함수를 최대한 효율적이고 작게 만들어 불필요한 파일이 로드되지 않도록 하십시오.
- Docker 파일 목록에서 정적 계층은 위쪽에 배치하고 자주 변경되는 계층은 아래쪽에 배치하십시오. 이렇게 하면 캐싱이 개선되어 성능이 향상됩니다.
- 이미지 업데이트 및 패치 적용은 이미지 소유자가 담당합니다. 운영 프로세스에 해당 업데이트 커밋을 추가합니다. 자세한 내용은 [AWS Lambda 설명서](#)를 참조하세요.

에픽

에서 프로젝트를 생성하십시오. CodeBuild

작업	설명	필요한 기술
CodeCommit 리포지토리 만들기.	Dockerfile, 파일 및 애플리케이션 소스 코드를 포함할	개발자

작업	설명	필요한 기술
	<p>CodeCommit 리포지토리를 생성합니다. buildspec.yaml 자세한 내용은 AWS CodeCommit 설명서를 참조하십시오.</p>	
CodeBuild 프로젝트 생성.	<p>CodeBuild 콘솔에서 CodeCommit 리포지토리와 buildspec.yaml 파일을 사용하는 새 프로젝트를 생성합니다. CodeBuild 프로젝트를 사용하여 이미지를 생성합니다.</p> <p>권한 모드가 활성화되었는지 확인합니다. Docker 이미지를 빌드하려면 이것이 필요합니다. 그렇지 않으면 이미지가 성공적으로 빌드되지 않습니다.</p> <p>프로젝트 이름 및 설명에 대한 값을 입력합니다. 소스 제공자의 경우 선택하세요 CodeCommit. 자세한 내용은 AWS 설명서를 참조하십시오.</p>	개발자

작업	설명	필요한 기술
<p>Dockerfile을 편집합니다.</p>	<p>Dockerfile은 애플리케이션을 개발하는 최상위 디렉터리에 있어야 합니다. Python 코드는 <code>src</code> 폴더에 있어야 합니다.</p> <p>이미지를 생성할 때는 Lambda 지원 공식 이미지를 사용하십시오. 그렇지 않으면 부트스트랩 오류가 발생하여 패키징 프로세스가 더 어려워집니다.</p> <p>자세한 내용은 추가 정보 섹션을 참조하세요.</p>	<p>개발자</p>
<p>Amazon ECR에 리포지토리를 생성합니다.</p>	<p>Amazon ECR에 컨테이너 리포지토리를 생성합니다. 다음 예제 명령에서 생성된 리포지토리의 이름은 <code>cf-demo</code>입니다. 리포지토리는 <code>buildspec.yaml</code> 파일에서 재사용됩니다.</p> <pre data-bbox="597 1209 1026 1327">aws ecr create-repository --cf-demo</pre>	<p>AWS 관리자, 개발자</p>

작업	설명	필요한 기술
이미지를 Amazon ECR로 푸시합니다.	를 CodeBuild 사용하여 이미지 빌드 프로세스를 수행할 수 있습니다. CodeBuild Amazon ECR과 상호 작용하고 S3를 사용할 수 있는 권한이 필요합니다. 프로세스의 일부로 Docker 이미지가 빌드되어 Amazon ECR 레지스트리로 푸시됩니다. 템플릿과 코드에 대한 자세한 내용은 추가 정보 섹션을 참조하세요.	개발자
이미지가 리포지토리에 있는지 확인하십시오.	이미지가 리포지토리에 있는지 확인하려면 Amazon ECR 콘솔에서 리포지토리를 선택합니다. Amazon ECR 설정에서 해당 기능이 켜져 있는 경우 취약성 스캔 보고서 결과와 함께 이미지가 태그 및 함께 나열되어야 합니다. 자세한 내용은 AWS 설명서 를 참조하십시오.	개발자

Lambda 함수를 생성하고 이미지를 배포합니다.

작업	설명	필요한 기술
Lambda 함수를 생성합니다.	Lambda 콘솔에서 함수 생성을 선택한 다음 컨테이너 이미지를 선택합니다. Amazon ECR 리포지토리에 있는 이미지의 URI와 함수 이름을 입력한 다음 함수 생성을 선택합니다. 자세한 내용은 AWS Lambda 설명서 를 참조하세요.	앱 개발자

작업	설명	필요한 기술
Lambda 함수를 테스트합니다.	함수를 간접 호출하고 테스트하려면 테스트를 선택합니다. 자세한 내용은 AWS Lambda 설명서 를 참조하세요.	앱 개발자

문제 해결

문제	Solution
빌드가 성공하지 못했습니다.	<ol style="list-style-type: none"> 프로젝트에 권한 모드가 켜져 있는지 확인하십시오. CodeBuild Docker 관련 명령에 필요한 권한이 있는지 확인합니다. 명령에 sudo를 추가해 보고 있습니다. 연결된 IAM 역할에 Amazon ECR, Amazon S3 및 CloudWatch 로그와 상호 작용하기 위한 적절한 작업이 포함된 정책이 있는지 확인하십시오. CodeBuild

관련 리소스

- [Lambda용 기본 이미지](#)
- [도커 샘플은 다음과 같습니다. CodeBuild](#)
- [임시 보안 인증 전달](#)

추가 정보

Dockerfile 편집

다음 코드는 Dockerfile에서 편집하는 명령을 보여줍니다.

```
FROM public.ecr.aws/lambda/python:3.11
```

```
# Copy function code
COPY app.py ${LAMBDA_TASK_ROOT}
COPY requirements.txt ${LAMBDA_TASK_ROOT}

# install dependencies
RUN pip3 install --user -r requirements.txt

# Set the CMD to your handler (could also be done as a parameter override outside of
  the Dockerfile)
CMD [ "app.lambda_handler" ]
```

FROM 명령 값은 공개 Amazon ECR 이미지 리포지토리의 Lambda 함수를 사용하는 Python 3.11 기본 이미지에 해당합니다.

이 COPY app.py \${LAMBDA_TASK_ROOT} 명령은 Lambda 함수가 사용할 작업 루트 디렉터리에 코드를 복사합니다. 이 명령은 환경 변수를 사용하므로 실제 경로에 대해 걱정할 필요가 없습니다. 실행할 함수는 CMD ["app.lambda_handler"] 명령에 인수로 전달됩니다.

이 COPY requirements.txt 명령은 코드에 필요한 종속성을 캡처합니다.

이 RUN pip install --user -r requirements.txt 명령은 로컬 사용자 디렉터리에 종속성을 설치합니다.

이미지를 빌드하려면 다음 명령을 실행합니다.

```
docker build -t <image name> .
```

Amazon ECR에 이미지를 추가합니다.

다음 코드에서 aws_account_id를 계정 번호로 바꾸고, 다른 리전을 사용하는 경우 us-east-1을 바꾸십시오. buildspec파일은 CodeBuild 빌드 번호를 사용하여 이미지 버전을 태그 값으로 고유하게 식별합니다. 이를 요구 사항에 맞게 변경할 수 있습니다.

빌드스펙 사용자 지정 코드

```
phases:
  install:
    runtime-versions:
      python: 3.11
  pre_build:
    commands:
      - python3 --version
```

```
- pip3 install --upgrade pip
- pip3 install --upgrade awscli
- sudo docker info
build:
  commands:
    - echo Build started on `date`
    - echo Building the Docker image...
    - ls
    - cd app
    - docker build -t cf-demo:$CODEBUILD_BUILD_NUMBER .
    - docker container ls
post_build:
  commands:
    - echo Build completed on `date`
    - echo Pushing the Docker image...
    - aws ecr get-login-password --region us-east-1 | docker login --username AWS --
password-stdin aws_account_id.dkr.ecr.us-east-1.amazonaws.com
    - docker tag cf-demo:$CODEBUILD_BUILD_NUMBER aws_account_id.dkr.ecr.us-
east-1.amazonaws.com/cf-demo:$CODEBUILD_BUILD_NUMBER
    - docker push aws_account_id.dkr.ecr.us-east-1.amazonaws.com/cf-demo:
$CODEBUILD_BUILD_NUMBER
```

Amazon EKS에 샘플 Java 마이크로서비스를 배포하고 Application Load Balancer를 사용하여 마이크로서비스 노출하기

작성자: 비제이 톰슨(AWS) 및 아카마하데비 하이어매스(AWS)

환경: PoC 또는 파일럿

기술: 컨테이너 및 마이크로서비스

워크로드: 오픈 소스

AWS 서비스: Amazon EC2
Container Registry, Amazon EKS, Amazon ECR

요약

이 패턴은 eksctl 명령줄 유틸리티와 Amazon Elastic Container Registry(Amazon ECR)를 사용하여 샘플 Java 마이크로서비스를 Amazon Elastic Kubernetes Service(Amazon EKS)에 컨테이너화된 애플리케이션으로 배포하는 방법을 설명합니다. Application Load Balancer을 사용하여 애플리케이션 트래픽을 로드 밸런싱할 수 있습니다.

사전 조건 및 제한 사항

사전 조건

- 활성 상태의 AWS 계정
- macOS, Linux 또는 Windows에 설치 및 구성된 AWS Command Line Interface(AWS CLI) 버전 1.7 이상
- 실행 중인 [Docker 데몬\(daemon\)](#)
- MacOS, Linux 또는 Windows에 설치 및 구성된 eksctl 명령줄 유틸리티(자세한 내용은 Amazon EKS 설명서의 [Amazon EKS - eksctl 시작하기](#) 참조)
- MacOS, Linux 또는 Windows에 설치 및 구성된 kubectl 명령줄 유틸리티(자세한 내용은 Amazon EKS 설명서의 [kubectl 설치 또는 업데이트](#) 참조)

제한 사항

- 이 패턴은 Application Load Balancer의 SSL 인증서 설치에는 적용되지 않습니다.

아키텍처

대상 기술 스택

- Amazon ECR
- Amazon EKS
- Elastic Load Balancing

대상 아키텍처

다음 다이어그램은 Amazon EKS에서 Java 마이크로서비스를 컨테이너화하기 위한 아키텍처를 보여줍니다.

도구

- [Amazon Elastic Container Registry\(Amazon ECR\)](#)는 안전하고 확장 가능하고 신뢰할 수 있는 관리형 컨테이너 이미지 레지스트리 서비스입니다.
- [Amazon Elastic Kubernetes Service \(Amazon EKS\)](#)는 자체 Kubernetes 컨트롤 플레인이나 노드를 설치하거나 유지 관리할 필요 없이 AWS에서 Kubernetes를 실행할 수 있도록 도와줍니다.
- [AWS Command Line Interface\(AWS CLI\)](#)는 명령줄 셸에서 명령을 사용하여 AWS 서비스와 상호 작용할 수 있는 오픈 소스 도구입니다.
- [Elastic Load Balancing](#)은 하나 이상의 가용 영역에서 Amazon Elastic Compute Cloud(Amazon EC2) 인스턴스, 컨테이너, IP 주소 등 여러 대상에 걸쳐 수신되는 트래픽을 자동으로 분산합니다.
- [eksctl](#)은 Amazon EKS에서 클러스터를 생성하는 데 도움이 됩니다.
- [kubecti](#)를 사용하면 쿠버네티스 클러스터에 대해 명령을 실행할 수 있습니다.
- [Docker](#)는 컨테이너라는 패키지로 애플리케이션을 빌드, 테스트 및 제공할 수 있도록 도와줍니다.

에픽

eksctl을 사용하여 Amazon EKS 클러스터 생성

작업	설명	필요한 기술
Amazon EKS 클러스터를 생성합니다.	<p>두 개의 t2.small Amazon EC2 인스턴스를 노드로 사용하는 Amazon EKS 클러스터를 생성하려면 다음 명령을 실행합니다.</p> <pre>eksctl create cluster --name <your-cluster-name> --version <version-number> --nodes=1 --node-type=t2.small</pre> <p>참고: 이 프로세스는 15분에서 20분 정도 걸릴 수 있습니다. 클러스터가 생성되고 나면 적절한 Kubernetes 구성이 kubeconfig 파일에 추가됩니다. 이후 단계에서 kubectl 와(과) 함께 kubeconfig 파일을 사용하여 애플리케이션을 배포할 수 있습니다.</p>	개발자, 시스템 관리자
Amazon EKS 클러스터를 확인합니다.	클러스터가 생성되었고 클러스터에 연결할 수 있는지 확인하려면 kubectl get nodes 명령을 실행합니다.	개발자, 시스템 관리자

Amazon ECR 리포지토리를 생성하고 도커 이미지를 푸시합니다.

작업	설명	필요한 기술
Amazon ECR 리포지토리를 생성합니다.	Amazon ECR 설명서의 프라이빗 리포지토리 생성 의 지침을 따르세요.	개발자, 시스템 관리자
POM XML 파일을 생성합니다.	이 패턴의 추가 정보 섹션에 있는 예제 POM 파일 코드를 기반으로 파일 pom.xml을(를) 생성합니다.	개발자, 시스템 관리자
소스 파일을 생성합니다.	<p>다음 예제를 기반으로 src/main/java/eksExample 경로에서 HelloWorld.java 라는 소스 파일을 생성합니다.</p> <pre>package eksExample; import static spark.Spark.get; public class HelloWorld { public static void main(String[] args) { get("/", (req, res) -> { return "Hello World!"; }); } }</pre> <p>다음과 같은 디렉터리 구조를 사용하도록 합니다.</p> <pre>### Dockerfile ### deployment.yaml</pre>	

작업	설명	필요한 기술
	<pre> ### ingress.yaml ### pom.xml ### service.yaml ### src ### main ### java ### eksExample ### HelloWorld.java </pre>	
Dockerfile을 생성합니다.	이 패턴의 추가 정보 섹션에 있는 예제 Dockerfile 코드를 기반으로 Dockerfile 을(를) 생성합니다.	개발자, 시스템 관리자

작업	설명	필요한 기술
<p>도커 이미지를 빌드하고 푸시합니다.</p>	<p>Dockerfile 이(가) 이미지를 빌드하고 태그를 지정하고 Amazon ECR로 푸시하도록 하려는 디렉터리에서 다음 명령을 실행합니다.</p> <pre data-bbox="592 489 1027 1360"> aws ecr get-login --password --region <region> docker login --username <username > --password-stdin <account_number>.d kr.ecr.<region>.am azonaws.com docker buildx build -- platform linux/amd64 -t hello-world-java:v 1 . docker tag hello-wor ld-java:v1 <account_ number>.dkr.ecr.<r egion>.amazonaws.com/ <repository_name>:v1 docker push <account_ number>.dkr.ecr.<r egion>.amazonaws.com/ <repository_name>:v1 </pre> <p>참고: 이전 명령에서 AWS 리전, 계정 번호, 리포지토리 세부 정보를 수정합니다. 나중에 사용할 수 있도록 이미지 URL을 기록하세요.</p> <p>중요: M1 칩이 장착된 macOS 시스템에서는 AMD64 플랫폼에서 실행되는 Amazon EKS와 호환되는 이미지를 구축하는</p>	

작업	설명	필요한 기술
	데 문제가 있습니다. 이 문제를 해결하려면 docker buildx 를 사용하여 Amazon EKS에서 작동하는 도커 이미지 빌드하세요.	

Java 마이크로서비스 배포

작업	설명	필요한 기술
배포 파일을 생성합니다.	이 패턴의 추가 정보 섹션에 있는 예제 배포 파일 코드를 기반으로 <code>deployment.yaml</code> 라는 YAML 파일을 생성합니다. 참고: 이전에 복사한 이미지 URL을 Amazon ECR 리포지토리의 이미지 파일의 경로로 사용합니다.	개발자, 시스템 관리자
Amazon EKS 클러스터에 Java 마이크로서비스를 배포합니다.	Amazon EKS 클러스터에 배포를 생성하려면 <code>kubectl apply -f deployment.yaml</code> 명령을 실행합니다.	개발자, 시스템 관리자
포드의 상태를 확인합니다.	<ol style="list-style-type: none"> 포드 상태를 확인하려면 <code>kubectl get pods</code> 명령을 실행합니다. 상태가 Ready로 변경될 때까지 기다립니다. 	개발자, 시스템 관리자
서비스를 생성합니다.	<ol style="list-style-type: none"> 이 패턴의 추가 정보 섹션에 있는 예제 서비스 파일 코드를 기반으로 <code>service.yaml</code> 라는 파일을 생성합니다. 	개발자, 시스템 관리자

작업	설명	필요한 기술
AWS Load Balancer Controller 추가 기능을 설치합니다.	<p>2. <code>kubectl apply -f service.yaml</code> 명령을 실행합니다.</p> <p>Amazon EKS 설명서의 AWS AWS Load Balancer Controller 추가 기능 지침을 따르세요.</p> <p>참고: Kubernetes 서비스를 위한 Application Load Balancer 또는 Network Load Balancer를 생성하려면 추가 기능을 설치해야 합니다.</p>	개발자, 시스템 관리자
인그레스 리소스를 생성합니다.	이 패턴의 추가 정보 섹션에 있는 예제 인그레스 리소스 파일 코드를 기반으로 <code>ingress.yaml</code> 라는 YAML 파일을 생성합니다.	개발자, 시스템 관리자
Application Load Balancer을 생성합니다.	인그레스 리소스를 배포하고 Application Load Balancer를 생성하려면 <code>kubectl apply -f ingress.yaml</code> 명령을 실행합니다.	개발자, 시스템 관리자

애플리케이션 테스트

작업	설명	필요한 기술
애플리케이션을 테스트하고 확인합니다.	1. ADDRESS 필드에서 로드 밸런서의 DNS 이름을 가져오려면 <code>kubectl get ingress.networking.k8s.io/java-micro</code>	개발자, 시스템 관리자

작업	설명	필요한 기술
	<p><code>service-ingress</code> 명령을 실행합니다.</p> <p>2. Amazon EKS 노드와 동일한 VPC에 있는 EC2 인스턴스에서 <code>curl -v <DNS address from previous command></code> 명령을 실행합니다.</p>	

관련 리소스

- [프라이빗 리포지토리 생성](#) (Amazon ECR 설명서)
- [도커 이미지 푸시](#) (Amazon ECR 설명서)
- [인그레스 컨트롤러](#) (Amazon EKS Workshop)
- [Docker buildx](#) (Docker Docs)

추가 정보

예제 POM 파일

```
<?xml version="1.0" encoding="UTF-8"?>
<project xmlns="http://maven.apache.org/POM/4.0.0" xmlns:xsi="http://www.w3.org/2001/
XMLSchema-instance"
  xsi:schemaLocation="http://maven.apache.org/POM/4.0.0 http://maven.apache.org/xsd/
maven-4.0.0.xsd">
  <modelVersion>4.0.0</modelVersion>

  <groupId>helloWorld</groupId>
  <artifactId>helloWorld</artifactId>
  <version>1.0-SNAPSHOT</version>

  <dependencies>
    <dependency>
```

```

    <groupId>com.sparkjava</groupId><artifactId>spark-core</
artifactId><version>2.0.0</version>
  </dependency>
</dependencies>
<build>
  <plugins>
    <plugin>
      <groupId>org.apache.maven.plugins</groupId><artifactId>maven-jar-plugin</
artifactId><version>2.4</version>
      <configuration><finalName>eksExample</finalName><archive><manifest>
        <addClasspath>true</addClasspath><mainClass>eksExample.HelloWorld</
mainClass><classpathPrefix>dependency-jars</classpathPrefix>
        </manifest></archive>
      </configuration>
    </plugin>
    <plugin>
      <groupId>org.apache.maven.plugins</groupId><artifactId>maven-compiler-plugin</
artifactId><version>3.1</version>
      <configuration><source>1.8</source><target>1.8</target></configuration>
    </plugin>
    <plugin>
      <groupId>org.apache.maven.plugins</groupId><artifactId>maven-assembly-plugin</
artifactId>
      <executions>
        <execution>
          <goals><goal>attached</goal></goals><phase>package</phase>
          <configuration>
            <finalName>eksExample</finalName>
            <descriptorRefs><descriptorRef>jar-with-dependencies</descriptorRef></
descriptorRefs>
            <archive><manifest><mainClass>eksExample.HelloWorld</mainClass></
manifest></archive>
          </configuration>
        </execution>
      </executions>
    </plugin>
  </plugins>
</build>
</project>

```

예제 Dockerfile

```
FROM bellsoft/liberica-openjdk-alpine-musl:17
```

```

RUN apk add maven
WORKDIR /code

# Prepare by downloading dependencies
ADD pom.xml /code/pom.xml
RUN ["mvn", "dependency:resolve"]
RUN ["mvn", "verify"]

# Adding source, compile and package into a fat jar
ADD src /code/src
RUN ["mvn", "package"]

EXPOSE 4567
CMD ["java", "-jar", "target/eksExample-jar-with-dependencies.jar"]

```

예제 배포 파일

```

apiVersion: apps/v1
kind: Deployment
metadata:
  name: microservice-deployment
spec:
  replicas: 2
  selector:
    matchLabels:
      app.kubernetes.io/name: java-microservice
  template:
    metadata:
      labels:
        app.kubernetes.io/name: java-microservice
    spec:
      containers:
        - name: java-microservice-container
          image: .dkr.ecr.amazonaws.com/:
          ports:
            - containerPort: 4567

```

예제 서비스 파일

```

apiVersion: v1
kind: Service
metadata:

```

```
name: "service-java-microservice"
spec:
  ports:
    - port: 80
      targetPort: 4567
      protocol: TCP
  type: NodePort
  selector:
    app.kubernetes.io/name: java-microservice
```

예제 인그레스 리소스 파일

```
apiVersion: networking.k8s.io/v1
kind: Ingress
metadata:
  name: "java-microservice-ingress"
  annotations:
    kubernetes.io/ingress.class: alb
    alb.ingress.kubernetes.io/load-balancer-name: apg2
    alb.ingress.kubernetes.io/target-type: ip
  labels:
    app: java-microservice
spec:
  rules:
    - http:
        paths:
          - path: /
            pathType: Prefix
            backend:
              service:
                name: "service-java-microservice"
                port:
                  number: 80
```

AWS Copilot을 사용하여 클러스터링된 애플리케이션을 Amazon ECS에 배포

작성자: 장 밥티스트 길로이스(AWS), 매튜 조지(AWS) 및 토마스 스콧(AWS)

코드 리포지토리: [클러스터링된 샘플 애플리케이션 데모](#)

환경: 프로덕션

기술: 컨테이너 및 마이크로서비스, 비즈니스 생산성, 클라우드 네이티브, 소프트웨어 개발 및 테스트

AWS 서비스: Amazon ECS, AWS Fargate, Amazon ECR

요약

이 패턴은 AWS Copilot이 배포 작업을 간소화하는 방법을 보여주기 위해 Amazon Web Services(AWS) Management Console을 사용하는 방법과 AWS Copilot을 사용하는 두 가지 방법으로 Amazon Elastic Container Service(Amazon ECS) 클러스터에 컨테이너를 배포하는 방법을 보여줍니다.

Amazon ECS는 클러스터에서 컨테이너를 손쉽게 실행, 중지 및 관리할 수 있게 해주는 컨테이너 관리 서비스로서 확장성과 속도가 뛰어납니다. 컨테이너는 서비스 내에서 개별 작업이나 여러 작업을 실행하는 데 사용하는 작업 정의에 정의됩니다. AWS Fargate에서 관리하는 서버리스 인프라에서 작업 및 서비스를 실행할 수 있습니다. 또는 인프라에 대한 더 세부적인 제어를 위해 사용자가 관리하는 Amazon Elastic Compute Cloud(Amazon EC2) 인스턴스의 클러스터에서 작업과 서비스를 실행할 수 있습니다.

AWS Copilot 명령줄 인터페이스(CLI) 명령은 로컬 개발 환경에서 Amazon ECS에 기반한 프로덕션 지원 컨테이너화된 애플리케이션의 구축, 릴리스 및 운영을 간소화합니다. AWS Copilot CLI는 인프라를 코드로 사용하는 것부터 사용자를 대신하여 프로비저닝되는 지속적 통합 및 지속적 전달(CI/CD) 파이프라인 생성에 이르기까지 최신 애플리케이션 모범 사례를 지원하는 개발자 워크플로우에 부합합니다. AWS Management Console 대신에 일상적인 개발 및 테스트 주기의 일부로 AWS Copilot CLI를 사용할 수 있습니다.

사전 조건 및 제한 사항

사전 조건

- 활성 상태의 AWS 계정
- AWS 계정을 사용하도록 AWS Command Line Interface(AWS CLI)가 로컬에 설치 및 구성됨(AWS CLI 설명서의 [설치 지침](#) 및 [구성 지침](#) 참조)
- AWS Copilot 로컬에 설치됨(Amazon ECS 설명서의 [설치 지침](#) 참조)
- 로컬 시스템에 Docker가 설치됨([Docker 설명서](#) 참조)

제한 사항

- Docker는 무료 요금제에서 IP 주소당 6시간당 컨테이너 이미지 100개의 풀 한도를 적용합니다.

아키텍처

대상 기술 스택

- Virtual Private Cloud(VPC), 퍼블릭 및 프라이빗 서브넷, 보안 그룹으로 설정된 AWS 환경
- Amazon ECS 클러스터
- Amazon ECS 서비스 및 작업 정의
- Amazon Elastic Container Registry (Amazon ECR)
- Amazon DynamoDB
- Application Load Balancer
- AWS Fargate
- Amazon Identity and Access Management(IAM)
- 아마존 CloudWatch
- AWS CloudTrail

대상 아키텍처

이 패턴에 대해 샘플 애플리케이션을 배포하면 여러 작업이 생성되어 별도의 가용 영역에 배포됩니다. 각 작업은 Amazon DynamoDB에 데이터를 저장합니다. 작업을 위해 웹 페이지에 액세스하면 다른 모든 작업의 데이터를 볼 수 있습니다.

도구

서비스

- [Amazon ECR](#) – Amazon Elastic Container Registry(Amazon ECR)는 안전하고 확장 가능하고 신뢰할 수 있는 AWS 관리형 컨테이너 이미지 레지스트리 서비스입니다. Amazon ECR은 IAM을 사용하여 리소스 기반 권한을 가진 프라이빗 리포지토리를 지원합니다.
- [Amazon ECS](#) - Amazon Elastic Container Service(Amazon ECS)는 클러스터에서 컨테이너를 손쉽게 실행, 중지 및 관리하기 위한 컨테이너 관리 서비스로서 확장성과 속도가 뛰어납니다. AWS Fargate에서 관리하는 서버리스 인프라에서 작업 및 서비스를 실행할 수 있습니다. 또는 인프라에 대한 더 세부적인 제어를 위해 사용자가 관리하는 Amazon Elastic Compute Cloud(Amazon EC2) 인스턴스의 클러스터에서 작업과 서비스를 실행할 수 있습니다.
- [AWS Copilot](#) - AWS Copilot은 레지스트리로 푸시, 작업 정의 생성, 클러스터 생성을 포함하여 AWS에서 컨테이너화된 애플리케이션을 시작하고 관리하는 데 도움이 되는 명령줄 인터페이스를 제공합니다.
- [AWS Fargate](#) — AWS Fargate는 서버를 관리하지 않고 애플리케이션 구축에만 집중할 수 있게 해주는 서버리스 pay-as-you-go 컴퓨팅 엔진입니다. AWS Fargate는 Amazon ECS 및 Amazon Elastic Kubernetes Service(Amazon EKS) 모두와 호환됩니다. Fargate 시작 유형 또는 Fargate 용량 공급자를 사용하여 Amazon Amazon ECS 태스크와 서비스를 실행할 때는 애플리케이션을 컨테이너에 패키징하고, CPU 및 메모리 요구 사항을 지정한 다음, 네트워킹 및 IAM 정책을 정의하고, 애플리케이션을 시작합니다. 각 Fargate 작업에는 자체 격리 경계가 있으며 다른 작업과 기본 커널, CPU 리소스, 메모리 리소스 또는 탄력적 네트워크 인터페이스를 공유하지 않습니다.
- [Amazon DynamoDB](#) - Amazon DynamoDB는 완전 관리형 NoSQL 데이터베이스 서비스로, 원활한 확장성과 함께 빠르고 예측 가능한 성능을 제공합니다.
- [Elastic Load Balancing \(ELB\)](#) – Elastic Load Balancing은 하나 이상의 가용 영역에서 EC2 인스턴스, 컨테이너, IP 주소 등 여러 대상에 걸쳐 수신되는 트래픽을 자동으로 분산합니다. 등록된 대상의 상태를 모니터링하면서 상태가 양호한 대상으로만 트래픽을 라우팅합니다. Elastic Load Balancing은 수신 트래픽이 시간이 지남에 따라 변경됨에 따라 로드 밸런서를 확장합니다. 대다수의 워크로드에 맞게 자동으로 조정할 수 있습니다.

도구

- [Docker 명령줄 인터페이스](#)
- [AWS Command Line Interface\(AWS CLI\)](#)
- [AWS Copilot 명령줄 인터페이스](#)

code

이 패턴에 사용된 샘플 애플리케이션의 코드는 [클러스터 샘플](#) 애플리케이션 리포지토리의 에서 GitHub 사용할 수 있습니다. 샘플 파일을 사용하려면 다음 섹션의 지침을 따르십시오.

에픽

애플리케이션 스택 배포 - 옵션 1 (AWS Management Console)

작업	설명	필요한 기술
GitHub 리포지토리를 복제합니다.	다음 명령을 사용하여 샘플 코드 리포지토리를 복제합니다. <pre data-bbox="594 772 1029 1052">git clone https://github.com/aws-samples/cluster-sample-app cluster-sample-app && cd cluster-sample-app</pre>	앱 개발자, AWS DevOps
Amazon ECR 리포지토리를 생성합니다.	<ol style="list-style-type: none"> 1. AWS Management Console에 로그인하고 https://console.aws.amazon.com/ecr/repositories에서 Amazon ECR 콘솔을 엽니다. 2. 리포지토리 생성을 선택합니다. 3. 리포지토리 이름을 입력하십시오 cluster-sample-app. 4. 기타 모든 설정은 기본값을 유지합니다. 5. 리포지토리 생성을 선택합니다. 	앱 개발자, AWS DevOps

작업	설명	필요한 기술
	자세한 내용은 Amazon ECR 설명서의 프라이빗 리포지토리 생성 을 참조하세요.	

작업	설명	필요한 기술
<p>도커 이미지를 빌드하고 태그를 지정하여 Amazon ECR 리포지토리에 푸시합니다.</p>	<ol style="list-style-type: none"> 1. 방금 생성한 리포지토리를 선택하고 푸시 명령 보기를 선택합니다. 2. 표시된 명령을 복사하고 로컬에서 실행하여 도커 이미지를 빌드하고 태그를 지정하고 푸시합니다. 이러한 명령은 다음과 비슷합니다. <p>Docker 클라이언트를 레지스트리에 인증하려면:</p> <pre>aws ecr get-login -password --region <YOUR_AWS_REGION> docker login --username e AWS --password-stdin <YOUR_AWS_ACCOUNT> .dkr.ecr.<YOUR_AWS _REGION>.amazonaws .com</pre> <p>도커 이미지 빌드하려면:</p> <pre>docker build -t cluster- sample-app .</pre> <p>도커 이미지에 태그를 지정하려면:</p> <pre>docker tag cluster- sample-app:latest <YOUR_AWS_ACCOUNT> .dkr.ecr.<YOUR_AWS _REGION>.amazonaws</pre>	<p>앱 개발자, AWS DevOps</p>

작업	설명	필요한 기술
	<pre>.com/cluster-sample-app:latest</pre> <p>리포지토리에 도커 이미지를 푸시하려면:</p> <pre>docker push <YOUR_AWS_ACCOUNT>.dkr.ecr.<YOUR_AWS_REGION>.amazonaws.com/cluster-sample-app:latest</pre>	

작업	설명	필요한 기술
<p>애플리케이션 스택을 배포합니다.</p>	<ol style="list-style-type: none"> 1. https://console.aws.amazon.com/cloudformation/ 에서 AWS CloudFormation 콘솔을 엽니다. 2. 스택 생성을 선택합니다. 3. 템플릿 준비에서 템플릿 준비 완료를 선택합니다. 4. 템플릿 지정 섹션에서 템플릿 파일 업로드를 선택합니다. 5. GitHub 리포지토리에 복제한 로컬 파일을 <code>cluster-sample-app-stack.yml</code> CloudFormation 템플릿으로 선택한 후 다음을 선택합니다. 6. 스택에 대해 이름을 입력한 후 다음을 선택합니다. 7. 기본 옵션을 유지한 다음에 다음을 선택합니다. 8. 모든 옵션을 검토하고 IAM 리소스 생성을 확인한 다음 스택 생성을 선택합니다. 9. 애플리케이션 스택이 배포되면 출력 탭을 선택하고 URL을 복사한 다음 브라우저에서 열어 애플리케이션에 액세스합니다. <p>CloudFormation 템플릿 배포에 대한 자세한 내용은 AWS</p>	<p>AWS DevOps, 앱 개발자</p>

작업	설명	필요한 기술
	CloudFormation 설명서의 스택 생성 을 참조하십시오.	

애플리케이션 스택 배포 - 옵션 2 (AWS Copilot CLI)

작업	설명	필요한 기술
GitHub 리포지토리를 복제합니다.	<p>다음 명령을 사용하여 샘플 코드 리포지토리를 복제합니다.</p> <pre>git clone https://github.com/aws-samples/cluster-sample-app cluster-sample-app && cd cluster-sample-app</pre>	앱 개발자, AWS DevOps
AWS Copilot CLI를 사용하여 컨테이너 이미지를 AWS에 배포합니다.	<p>프로젝트의 루트 디렉터리에서 다음 명령을 사용하여 애플리케이션을 한 번에 배포합니다.</p> <pre>copilot init --app cluster-sample-app --name demo --type "Load Balanced Web Service" --dockerfile ./Dockerfile --port 8080 --deploy</pre> <p>그러면 출력으로 제공된 DNS 이름을 사용하여 애플리케이션에 액세스할 수 있어야 합니다.</p>	앱 개발자, AWS DevOps

생성된 리소스 삭제

작업	설명	필요한 기술
AWS Management Console을 통해 생성된 리소스를 삭제합니다.	<p>옵션 1(AWS Management Console)을 사용하여 애플리케이션 스택을 배포한 경우, 생성한 리소스를 삭제할 준비가 되면 다음 단계를 따릅니다.</p> <ol style="list-style-type: none"> 1. https://console.aws.amazon.com/cloudformation/ 에서 CloudFormation 콘솔을 엽니다. 2. 생성한 스택을 선택한 다음 삭제를 선택합니다. 3. Amazon ECR 콘솔(https://console.aws.amazon.com/ecr/repositories)을 엽니다. 4. 생성한 리포지토리를 선택한 다음 삭제를 선택합니다. 	앱 개발자, AWS DevOps
AWS Copilot에서 생성한 리소스를 삭제합니다.	<p>옵션 2(AWS Copilot CLI)를 사용하여 애플리케이션 스택을 배포한 경우, 생성한 리소스를 삭제할 준비가 되면 프로젝트의 루트 디렉터리에서 다음 명령을 실행합니다.</p> <pre>copilot app delete</pre>	앱 개발자, AWS DevOps

관련 리소스

- [AWS CLI의 최신 버전 설치 또는 업데이트](#) (AWS CLI 설명서)
- [AWS Copilot 명령줄 인터페이스 사용](#) (Amazon ECS 설명서)

- [AWS Fargate의 Amazon ECS](#) (Amazon ECR 설명서)
- [Amazon ECS 설명서](#)
- [Amazon ECR 설명서](#)
- [아마존 CloudFormation 설명서](#)
- [Docker Desktop](#) (Docker 설명서)

Amazon EKS 클러스터에 gRPC 기반 애플리케이션을 배포하고 Application Load Balancer를 사용하여 액세스하기

키란쿠마르 찬드라세카르 (AWS) 와 후이 응우옌 (AWS) 이 제작했습니다.

[grpc-traffic-on-alb코드 리포지토리: -to-eks](#)

환경: PoC 또는 파일럿

기술: 컨테이너 및 마이크로서비스, 콘텐츠 전송, 웹 및 모바일 앱

워크로드: 기타 모든 워크로드

AWS 서비스: Amazon EKS, Elastic Load Balancing(ELB)

요약

이 패턴은 Amazon Elastic Kubernetes Service(Amazon EKS) 클러스터에서 gRPC 기반 애플리케이션을 호스팅하고 Application Load Balancer를 통해 안전하게 액세스하는 방법을 설명합니다.

[gRPC](#)는 모든 환경에서 실행할 수 있는 오픈 소스 원격 프로시저 호출(RPC) 프레임워크입니다. 마이크로서비스 통합 및 클라이언트 서버 통신에 사용할 수 있습니다. gRPC에 대한 자세한 내용은 AWS 블로그 게시물 [end-to-end HTTP/2 및 gRPC에 대한 애플리케이션 로드 밸런서 지원을 참조하십시오](#).

이 패턴은 Amazon EKS의 쿠버네티스 포드에서 실행되는 gRPC 기반 애플리케이션을 호스팅하는 방법을 보여줍니다. gRPC 클라이언트는 SSL/TLS로 암호화된 연결을 사용하여 HTTP/2 프로토콜을 통해 애플리케이션 로드 밸런서에 연결합니다. Application Load Balancer는 Amazon EKS 포드에서 실행되는 gRPC 애플리케이션으로 트래픽을 전달합니다. [Kubernetes Horizontal Pod Autoscaler](#)를 사용하여 gRPC 포드의 수를 트래픽에 따라 자동으로 조정할 수 있습니다. Application Load Balancer의 대상 그룹은 Amazon EKS 노드에서 상태 확인을 수행하고 대상이 정상인지 평가한 다음 트래픽을 정상 노드에만 전달합니다.

사전 조건 및 제한 사항

사전 조건

- 활성 상태의 AWS 계정.
- [Docker](#), Linux, macOS 또는 Windows에 설치 및 구성됨

- [AWS Command Line Interface\(AWS CLI\) 버전 2](#), Linux, macOS 또는 Windows에 설치 및 구성됨.
- [eksctl](#), 리눅스, 맥OS 또는 윈도우에 설치 및 구성
- kubectl, Amazon EKS 클러스터의 리소스에 액세스하도록 설치 및 구성됨. 자세한 내용은 Amazon EKS 설명서의 [kubectl 설치 또는 업데이트를](#) 참조하십시오.
- [gRPCURL](#), 설치 및 구성됨
- 신규 또는 기존 Amazon EKS 클러스터. 자세한 내용은 [Amazon EKS 시작하기](#)를 참조하십시오.
- 컴퓨터 터미널이 Amazon EKS 클러스터에 액세스하도록 구성되어 있습니다. 자세한 내용은 Amazon EKS 설명서에서 [클러스터와 통신하도록 컴퓨터 구성](#)을 참조하십시오.
- [AWS Load Balancer Controller](#), Amazon EKS 클러스터에서 프로비저닝됨.
- 유효한 SSL 또는 SSL/TLS 인증서가 있는 기존 DNS 호스트 이름. AWS Certificate Manager(ACM)를 사용하거나 기존 인증서를 ACM에 업로드하여 도메인에 대한 인증서를 받을 수 있습니다. 이 두 가지 옵션에 대한 자세한 내용은 ACM 설명서의 [공인 인증서 요청](#) 및 [AWS Certificate Manager로 인증서 가져오기](#)를 참조하세요.

아키텍처

다음 다이어그램은 이 패턴으로 구현된 아키텍처를 보여줍니다.

다음 다이어그램은 Application Load Balancer로 오프로드되는 gRPC 클라이언트에서 SSL/TLS 트래픽을 수신하는 워크플로우를 보여줍니다. 트래픽은 Virtual Private Cloud(VPC)에서 오기 때문에 gRPC 서버에 일반 텍스트로 전달됩니다.

도구

서비스

- [AWS Command Line Interface\(AWS CLI\)](#)는 명령줄 셸에서 명령을 사용하여 AWS 서비스와 상호 작용할 수 있는 오픈 소스 도구입니다.
- [Elastic Load Balancing\(ELB\)](#)은 들어오는 애플리케이션 또는 네트워크 트래픽을 여러 대상에 분산합니다. 예를 들어 하나 이상의 가용 영역에 있는 Amazon Elastic Compute Cloud(Amazon EC2) 인스턴스, 컨테이너, IP 주소 전반에 걸쳐 트래픽을 분산할 수 있습니다.

- [Amazon Elastic Container Registry\(Amazon ECR\)](#)는 안전하고 확장 가능하고 신뢰할 수 있는 관리형 컨테이너 이미지 레지스트리 서비스입니다.
- [Amazon Elastic Kubernetes Service \(Amazon EKS\)](#)는 자체 Kubernetes 컨트롤 플레인 또는 노드를 설치하거나 유지 관리할 필요 없이 Kubernetes를 실행하는 데 도움이 됩니다.

도구

- [eksctl](#)은 Amazon EKS에서 클러스터를 생성하기 위한 간단한 CLI 도구입니다.
- [kubecti](#)은 Kubernetes 클러스터에 대해 명령을 실행하기 위한 명령줄 유틸리티입니다.
- [AWS Load Balancer Controller](#)는 Kubernetes 클러스터의 AWS Elastic Load Balancer를 관리하는 데 유용합니다.
- [gRPCurl](#)은 gRPC 서비스와 상호 작용하는 데 도움이 되는 명령줄 도구입니다.

코드 리포지토리

[이 패턴의 코드는 -to-eks 리포지토리에서 사용할 수 있습니다. GitHub grpc-traffic-on-alb](#)

에픽

gRPC 서버의 도커 이미지를 빌드하여 Amazon ECR로 푸시합니다.

작업	설명	필요한 기술
Amazon ECR 리포지토리를 생성합니다.	<p>AWS 관리 콘솔에 로그인하고 Amazon ECR 콘솔을 연 다음 Amazon ECR 리포지토리를 생성합니다. 자세한 내용은 Amazon ECR 설명서의 리포지토리 생성을 참조하십시오. Amazon ECR 리포지토리의 URL을 기록하세요.</p> <p>다음 명령을 실행하여 AWS CLI를 사용하여 Amazon ECR 리포지토리를 생성할 수도 있습니다.</p>	클라우드 관리자

작업	설명	필요한 기술
	<pre>aws ecr create-repository --repository-name helloworld-grpc</pre>	
<p>Docker 이미지를 구축합니다.</p>	<ol style="list-style-type: none"> 1. GitHub grpc-traffic-on-alb-to-eks 리포지토리를 복제하십시오. <pre>git clone https://github.com/aws-samples/grpc-traffic-on-alb-to-eks.git</pre> <ol style="list-style-type: none"> 2. 리포지토리의 루트 디렉터리에서 Dockerfile이 존재하는지 확인한 후 다음 명령을 실행하여 Docker 이미지를 빌드합니다. <pre>docker build -t <amazon_ecr_repository_url>:<Tag> .</pre> <p>중요: 이전에 생성한 Amazon ECR 리포지토리의 <amazon_ecr_repository_url> URL로 바꿔야 합니다.</p>	<p>DevOps 엔지니어</p>

작업	설명	필요한 기술
<p>도커 이미지를 Amazon ECR로 푸시합니다.</p>	<p>1. 다음 명령을 실행하여 Amazon ECR 리포지토리에 로그인합니다.</p> <pre data-bbox="630 394 1029 793">aws ecr get-login -password --region us-east-1 --no-cli- auto-prompt docker login --username AWS --password-stdin <your_aws_account_ id>.dkr.ecr.us-eas t-1.amazonaws.com</pre> <p>2. 다음 명령을 실행하여 도커 이미지를 Amazon ECR 리포지토리에 푸시합니다.</p> <pre data-bbox="630 974 1029 1213">docker push <your_aws _account_id>.dkr.e cr.us-east-1.amazo naws.com/helloworl d-grpc:1.0</pre> <p>중요: <your_aws_account_id> 을(를) AWS 계정 ID로 변경하세요.</p>	<p>DevOps 엔지니어</p>

Kubernetes 매니페스트를 Amazon EKS 클러스터에 배포합니다.

작업	설명	필요한 기술
<p>Kubernetes 매니페스트 파일의 값을 수정합니다.</p>	<p>1. 요구 사항에 따라 grpc-sample.yaml 리포지토리의 Kubernetes 폴더에 있는 Kubernetes 매니페스트 파</p>	<p>DevOps 엔지니어</p>

작업	설명	필요한 기술
	<p>일을 수정하십시오. 인그레스 리소스의 주석과 호스트 이름을 수정해야 합니다. 샘플 인그레스 리소스는 추가 정보 섹션을 참조하세요. 인그레스 주석에 대한 자세한 내용은 Kubernetes 설명서의 인그레스 주석을 참조하세요.</p> <p>2. Kubernetes 배포 리소스에서 배포 리소스 image을 (를) 도커 이미지를 푸시한 Amazon ECR 리포지토리의 URI(Uniform Resource Identifier)로 변경합니다. 샘플 배포 리소스는 추가 정보 섹션을 참조하세요.</p>	
<p>Kubernetes 매니페스트 파일을 배포합니다.</p>	<p>다음 kubectl 명령을 실행하여 Amazon EKS 클러스터에 grpc-sample.yaml 파일을 배포합니다.</p> <pre data-bbox="597 1283 1026 1440">kubectl apply -f ./kubernetes/grpc-sample.yaml</pre>	<p>DevOps 엔지니어</p>

Application Load Balancer의 FQDN에 대한 DNS 레코드 생성

작업	설명	필요한 기술
<p>Application Load Balancer에 대한 FQDN을 기록합니다.</p>	<p>1. 다음 kubectl 명령을 실행하여 Application Load Balancer를 관리하는</p>	<p>DevOps 엔지니어</p>

작업	설명	필요한 기술
	<p>Kubernetes 인그레스 리소스를 설명합니다.</p> <pre data-bbox="630 331 1029 449">kubect1 get ingress -n grpcserver</pre> <p>샘플 출력은 추가 정보 섹션에 제공됩니다. 출력의 HOSTS 필드에는 SSL 인증서 생성에 사용한 DNS 호스트 이름이 표시됩니다.</p> <ol style="list-style-type: none"> 출력 Address 필드에서 애플리케이션 로드 밸런서의 FQDN (정규화된 도메인 이름) 을 기록합니다. 애플리케이션 로드 밸런서의 FQDN을 가리키는 DNS 레코드를 생성합니다. DNS 공급자가 Amazon Route 53 인 경우 Application Load Balancer의 FQDN을 가리키는 별칭 레코드를 생성할 수 있습니다. 이 옵션에 대한 자세한 내용은 Route 53 설명서의 별칭 레코드와 비별칭 레코드 간의 선택을 참조하십시오. 	

솔루션 테스트

작업	설명	필요한 기술
gRPC 서버를 테스트합니다.	<p>다음 명령을 실행하여 gRPCurl 을 사용하여 엔드포인트를 테스트합니다.</p> <pre data-bbox="594 491 1029 770"> grpcurl grpc.example.com:443 list grpc.reflection.v1alpha.ServerReflection helloworld.helloworld </pre> <p>참고: DNS 이름으로 <code>grpc.example.com</code> 을(를) 변경합니다.</p>	DevOps 엔지니어
gRPC 클라이언트를 사용하여 gRPC 서버를 테스트합니다.	<p><code>helloworld_client_ssl.py</code> 샘플 gRPC 클라이언트에서 의 호스트 이름을 gRPC <code>grpc.example.com</code> 서버에 사용되는 호스트 이름으로 바꿉니다.</p> <p>다음 코드 샘플은 클라이언트 요청에 대한 gRPC 서버의 응답을 보여줍니다.</p> <pre data-bbox="594 1472 1029 1885"> python ./app/helloworld_client_ssl.py message: "Hello to gRPC server from Client" message: "Thanks for talking to gRPC server!! Welcome to hello world. Received message is \"Hello to </pre>	DevOps 엔지니어

작업	설명	필요한 기술
	<pre>gRPC server from Client \"" received: true</pre> <p>이는 클라이언트가 서버와 통신할 수 있고 연결에 성공했음을 보여줍니다.</p>	

정리

작업	설명	필요한 기술
DNS 레코드를 제거합니다.	이전에 만든 애플리케이션 로드 밸런서의 FQDN을 가리키는 DNS 레코드를 제거합니다.	클라우드 관리자
로드 밸런서를 제거합니다.	Amazon EC2 콘솔에서 Load Balancers를 선택한 다음 Kubernetes 컨트롤러가 수신 리소스용으로 생성한 로드 밸런서를 제거합니다.	클라우드 관리자
Amazon EKS 클러스터를 삭제합니다.	다음을 사용하여 eksctl Amazon EKS 클러스터를 삭제합니다.	AWS DevOps
	<pre>eksctl delete cluster - f ./eks.yaml</pre>	

관련 리소스

- [Amazon EKS의 네트워크 로드 밸런싱](#)
- [Application Load Balancer 대상 그룹](#)

추가 정보

샘플 인그레스 리소스:

```

---
apiVersion: networking.k8s.io/v1
kind: Ingress
metadata:
  annotations:
    alb.ingress.kubernetes.io/healthcheck-protocol: HTTP
    alb.ingress.kubernetes.io/ssl-redirect: "443"
    alb.ingress.kubernetes.io/backend-protocol-version: "GRPC"
    alb.ingress.kubernetes.io/listen-ports: '[{"HTTP": 80}, {"HTTPS":443}]'
    alb.ingress.kubernetes.io/scheme: internet-facing
    alb.ingress.kubernetes.io/target-type: ip
    alb.ingress.kubernetes.io/certificate-arn: arn:aws:acm:<AWS-
Region>:<AccountId>:certificate/<certificate_ID>
    alb.ingress.kubernetes.io/healthcheck-protocol: HTTP
  labels:
    app: grpcserver
    environment: dev
    name: grpcserver
    namespace: grpcserver
spec:
  ingressClassName: alb
  rules:
  - host: grpc.example.com # <----- replace this as per your host name for which the
SSL certtificate is available in ACM
    http:
      paths:
      - backend:
          service:
            name: grpcserver
            port:
              number: 9000
        path: /
        pathType: Prefix

```

샘플 배포 리소스:

```

apiVersion: apps/v1
kind: Deployment
metadata:

```

```

name: grpcserver
namespace: grpcserver
spec:
  selector:
    matchLabels:
      app: grpcserver
  replicas: 1
  template:
    metadata:
      labels:
        app: grpcserver
    spec:
      containers:
        - name: grpc-demo
          image: <your_aws_account_id>.dkr.ecr.us-east-1.amazonaws.com/helloworld-
grpc:1.0 #<----- Change to the URI that the Docker image is pushed to
          imagePullPolicy: Always
          ports:
            - name: grpc-api
              containerPort: 9000
          env:
            - name: POD_IP
              valueFrom:
                fieldRef:
                  fieldPath: status.podIP
          restartPolicy: Always

```

샘플 출력:

NAME	CLASS	HOSTS	Address
PORTS	AGE		
grpcserver	<none>	<DNS-HostName>	<ELB-address>
80	27d		

Amazon EKS 클러스터의 배포 및 디버깅

작성자: Svenja Raether 및 Mathew George

환경: PoC 또는 파일럿

기술: 컨테이너 및 마이크로서비스, 인프라, 현대화, 서버리스, 클라우드 네이티브

워크로드: 기타 모든 워크로드

AWS 서비스: Amazon EKS, Fargate

요약

컨테이너는 클라우드 네이티브 애플리케이션 개발의 필수적인 부분이 되고 있습니다. Kubernetes는 컨테이너의 관리 및 오케스트레이션을 위한 효율적인 방법을 제공합니다. [Amazon Elastic Kubernetes Service\(Amazon EKS\)](#)는 Amazon Web Services(AWS)에서 Kubernetes 클러스터를 빌드, 보안 유지, 운영, 유지 관리하기 위한 인증된 완전 관리형 [Kubernetes](#) 준수 서비스입니다. Fargate에서 포드를 실행하여 적정 규모의 온디맨드 컴퓨팅 용량을 제공하는 것을 지원합니다.

개발자와 관리자는 컨테이너식 워크로드를 실행할 때 디버깅 옵션을 알아야 합니다. 이 패턴은 [Fargate](#)를 사용하여 Amazon EKS에 컨테이너를 배포하고 디버깅하는 과정을 안내합니다. 여기에는 Amazon EKS 워크로드의 생성, 배포, 액세스, 디버깅, 정리가 포함됩니다.

사전 조건 및 제한 사항

사전 조건

- 활성 상태의 [AWS 계정](#)
- Amazon EKS, IAM 역할, 서비스 연결 역할을 생성하고 이와 상호 작용할 수 있는 충분한 권한이 구성된 [Identity and Access Management\(IAM\)](#) 역할
- 로컬 시스템에 설치된 [Command Line Interface \(CLI\)](#)
- [eksctl](#)
- [kubectl](#)
- [Helm](#)

제한 사항

- 이 패턴은 개발자에게 개발 환경에 유용한 디버깅 방법을 제공합니다. 프로덕션 환경에 대한 모범 사례는 설명하지는 않습니다.
- Windows를 실행하는 경우 운영 체제별 명령을 사용하여 환경 변수를 설정합니다.

사용된 제품 버전

- [CLI version 2](#)
- 사용하고 있는 Amazon EKS 제어 플레인의 한 가지 마이너 버전 차이 이내에 있는 [kubect1 버전](#)
- [eksctl](#) 최신 버전
- [Helm v3](#)

아키텍처

기술 스택

- Application Load Balancer
- Amazon EKS
- AWS Fargate

대상 아키텍처

다이어그램에 표시된 모든 리소스는 로컬 시스템에서 실행된 eksctl 및 kubect1 명령을 사용하여 프로비저닝됩니다. 프라이빗 클러스터는 프라이빗 VPC 내부에 있는 인스턴스에서 실행해야 합니다.

대상 아키텍처는 Fargate 시작 유형을 사용하여 EKS 클러스터로 구성됩니다. 이를 통해 서버 유형을 지정할 필요 없이 적정 규모의 온디맨드 컴퓨팅 용량을 제공합니다. EKS 클러스터에는 클러스터 노드와 워크로드를 관리하는 데 사용되는 제어 플레인이 있습니다. 포드는 여러 가용 영역에 걸친 프라이빗 VPC 서브넷에 프로비저닝됩니다. Amazon ECR 퍼블릭 갤러리를 참조하여 NGINX 웹 서버 이미지를 검색해서 클러스터의 포드에 배포합니다.

다이어그램은 kubect1 명령을 사용하여 Amazon EKS 제어 플레인에 액세스하는 방법과 Application Load Balancer를 사용하여 애플리케이션에 액세스하는 방법을 보여줍니다.

1. 클라우드 외부에 있는 로컬 시스템은 Amazon EKS 관리형 VPC 내부에 있는 Kubernetes 제어 플레인에 명령을 전송합니다.
2. Amazon EKS는 Fargate 프로파일의 선택기를 기반으로 포드를 스케줄링합니다.
3. 로컬 시스템이 브라우저에서 Application Load Balancer URL을 엽니다.
4. Application Load Balancer는 여러 가용 영역에 걸친 프라이빗 서브넷에 배포된 Fargate 클러스터 노드의 Kubernetes 포드 간 트래픽을 나눕니다.

도구

서비스

- [Amazon Elastic Container Registry\(Amazon ECR\)](#)는 안전하고 확장성이 있고 신뢰할 수 있는 관리형 컨테이너 이미지 레지스트리 서비스입니다.
- [Amazon Elastic Kubernetes Service \(Amazon EKS\)](#)는 자체 Kubernetes 컨트롤 플레인 또는 노드를 설치하거나 유지 관리할 필요 없이 Kubernetes를 실행하는 데 도움이 됩니다. 또한 이 패턴은 eksctl 명령줄 도구를 사용하여 Amazon EKS의 Kubernetes 클러스터와 함께 작동합니다.
- [AWS Fargate](#)를 사용하면 서버 또는 Amazon Elastic Compute Cloud(Amazon EC2) 인스턴스를 관리할 필요 없이 컨테이너를 실행할 수 있습니다. Amazon Elastic Container Service(Amazon ECS)와 함께 사용합니다.
- [Elastic Load Balancing\(ELB\)](#)은 들어오는 애플리케이션 또는 네트워크 트래픽을 여러 대상에 분산합니다. 예를 들어 하나 이상의 가용 영역에 있는 Amazon Elastic Compute Cloud(Amazon EC2) 인스턴스, 컨테이너, IP 주소 전반에 걸쳐 트래픽을 분산할 수 있습니다. 이 패턴은 [Kubernetes 인그레스](#)가 프로비저닝될 때 [AWS Load Balancer Controller](#) 제어 구성 요소를 사용하여 Application Load Balancer를 생성합니다. Application Load Balancer를 사용하여 여러 대상 사이로 수신 트래픽을 분산합니다.

기타 도구

- [Helm](#)은 Kubernetes용 오픈소스 패키지 관리자입니다. 이 패턴에서, Helm은 Load Balancer Controller를 설치하는 데 사용됩니다.
- [Kubernetes](#)는 컨테이너화된 애플리케이션의 배포, 규모 조정 및 관리 자동화를 위한 오픈 소스 시스템입니다.
- [NGINX](#)는 고성능 웹 및 리버스 프록시 서버입니다.

에픽

EKS 클러스터 생성

작업	설명	필요한 기술
<p>파일을 생성합니다.</p>	<p>추가 정보 섹션에 코드를 사용하여 다음 파일을 생성합니다.</p> <ul style="list-style-type: none"> • clusterconfig-fargate.yaml • nginx-deployment.yaml • nginx-service.yaml • nginx-ingress.yaml • index.html 	<p>앱 개발자, AWS 관리자, AWS DevOps</p>
<p>환경 변수를 설정합니다.</p>	<p>참고: 이전에 완료하지 않은 작업 때문에 명령이 실패하는 경우 몇 초 정도 기다린 다음 명령을 다시 실행합니다.</p> <p>이 패턴은 파일 clusterconfig-fargate.yaml 에 정의된 AWS 리전 및 클러스터 이름을 사용합니다. 추가 명령에서 참조할 수 있도록 환경 변수와 동일한 값을 설정합니다.</p> <pre>export AWS_REGION="us-east-1" export CLUSTER_NAME="my-fargate"</pre>	<p>앱 개발자, AWS DevOps, AWS 시스템 관리자</p>
<p>EKS 클러스터를 생성합니다.</p>	<p>clusterconfig-fargate.yaml 파일의 사양을 사용하는 EKS 클러스터를 생성</p>	<p>앱 개발자, AWS DevOps, AWS 관리자</p>

작업	설명	필요한 기술
	<p>하려면 다음 명령을 실행합니다.</p> <pre data-bbox="594 331 1027 489">eksctl create cluster -f clusterconfig-fargate.yaml</pre> <p>이 파일에는 ClusterConfig (이)가 포함되어 있으며, 이는 my-fargate-cluster 이름이 지정된 us-east-1 Region의 새 EKS 클러스터와 하나의 기본 Fargate 프로파일을 프로비저닝합니다(fp-default).</p> <p>기본 Fargate 프로파일은 두 개의 선택기로 구성됩니다 (default 및 kube-system).</p>	

작업	설명	필요한 기술
<p>생성된 클러스터를 확인합니다.</p>	<p>생성된 클러스터를 확인하려면 다음 명령을 실행합니다.</p> <pre>eksctl get cluster --output yaml</pre> <p>출력은 다음과 같아야 합니다.</p> <pre>- Name: my-fargate Owned: "True" Region: us-east-1</pre> <p>CLUSTER_NAME (을)를 사용하여 생성된 Fargate 프로파일을 확인합니다.</p> <pre>eksctl get fargateprofile --cluster \$CLUSTER_NAME --output yaml</pre> <p>이 명령은 리소스에 대한 정보를 표시합니다. 이 정보를 사용하여 생성된 클러스터를 검증할 수 있습니다. 출력은 다음과 같아야 합니다.</p> <pre>- name: fp-default podExecutionRoleARN: arn:aws:iam::<YOUR-ACCOUNT-ID>:role/eksctl-my-fargate-cluster-FargatePodExecutionRole-xxx selectors: - namespace: default</pre>	<p>앱 개발자, AWS DevOps, AWS 시스템 관리자</p>

작업	설명	필요한 기술
	<ul style="list-style-type: none"> - namespace: kube-system status: ACTIVE subnets: - subnet-aaa - subnet-bbb - subnet-ccc 	

컨테이너 배포

작업	설명	필요한 기술
<p>NGINX 웹 서버를 배포합니다.</p>	<p>클러스터에 NGINX 웹 서버 배포를 적용하려면 다음 명령을 실행합니다.</p> <pre>kubectl apply -f ./nginx-deployment.yaml</pre> <p>출력은 다음과 같아야 합니다.</p> <pre>deployment.apps/nginx-deployment created</pre> <p>배포에는 Amazon ECR 퍼블릭 갤러리에서 가져온 NGINX 이미지의 복제본 3개가 포함됩니다. 이미지는 기본 네임스페이스에 배포되고 실행 중인 포드의 포트 80에 노출됩니다.</p>	<p>앱 개발자, AWS DevOps, AWS 시스템 관리자</p>
<p>배포 및 포드를 확인합니다.</p>	<p>(선택 사항) 배포를 확인합니다. 다음 명령을 사용하여 배포 상태를 쿼리할 수 있습니다.</p>	<p>앱 개발자, AWS DevOps, AWS 관리자</p>

작업	설명	필요한 기술
	<pre>kubectl get deployment</pre> <p>출력은 다음과 같아야합니다.</p> <pre>NAME READY UP-TO-DATE AVAILABLE AGE nginx-deployment 3/3 3 3 7m14s</pre> <p>포드는 Kubernetes에서 배포 가능한 객체로, 하나 이상의 컨테이너를 포함합니다. 모든 포드를 목록화하려면, 다음 명령을 실행합니다.</p> <pre>kubectl get pods</pre> <p>출력은 다음과 같아야합니다.</p> <pre>NAME STATUS READY RESTARTS AGE nginx-deployment-xxxx-aaa 1/1 Running 0 94s nginx-deployment-xxxx-bbb 1/1 Running 0 94s nginx-deployment-xxxx-ccc 1/1 Running 0 94s</pre>	

작업	설명	필요한 기술
<p>배포의 규모를 조정합니다.</p>	<p>deployment.yaml 에서 지정된 복제본 3개에서 복제본 4개로 배포를 조정하려면 다음 명령을 사용합니다.</p> <pre data-bbox="592 441 1031 598">kubectl scale deployment nginx-deployment --replicas 4</pre> <p>출력은 다음과 같아야합니다.</p> <pre data-bbox="592 714 1031 829">deployment.apps/nginx-deployment scaled</pre>	<p>앱 개발자, AWS DevOps, AWS 시스템 관리자</p>

Load Balancer Controller 배포

작업	설명	필요한 기술
<p>환경 변수를 설정합니다.</p>	<p>클러스터의 VPC에 대한 정보를 검색할 수 있는 클러스터의 CloudFormation 스택을 설명하십시오.</p> <pre data-bbox="592 1333 1031 1690">aws cloudformation describe-stacks --stack-name eksctl-\$CLUSTER_NAME-cluster --query "Stacks[0].Outputs[?OutputKey==`\`VPC\`].OutputValue"</pre> <p>출력은 다음과 같아야합니다.</p> <pre data-bbox="592 1795 1031 1858">[</pre>	<p>앱 개발자, AWS DevOps, AWS 시스템 관리자</p>

작업	설명	필요한 기술
	<pre data-bbox="594 205 1027 348">"vpc-<YOUR-VPC-ID> "</pre> <p data-bbox="594 380 1027 464">VPC ID를 복사하여 환경 변수로 내보냅니다.</p> <pre data-bbox="594 499 1027 625">export VPC_ID="vpc- <YOUR-VPC-ID>"</pre>	
<p data-bbox="110 657 526 741">클러스터 서비스 계정에 대한 IAM을 구성합니다.</p>	<p data-bbox="594 657 1027 842">이전 에픽의 <code>AWS_REGION</code> 및 <code>CLUSTER_NAME</code> (을)를 사용하여 클러스터용 IAM Open ID Connect 공급자를 생성합니다.</p> <pre data-bbox="594 877 1027 1157">eksctl utils associate- iam-oidc-provider \ --region \$AWS_REGION \ --cluster \$CLUSTER_ NAME \ --approve</pre>	<p data-bbox="1068 657 1430 741">앱 개발자, AWS DevOps, AWS 시스템 관리자</p>

작업	설명	필요한 기술
<p>IAM 정책을 다운로드하고 생성합니다.</p>	<p>사용자 대신 API를 호출할 수 있는 로드 밸런서 컨트롤러의 IAM 정책을 다운로드합니다.</p> <pre data-bbox="594 394 1026 751">curl -o iam-policy.json https://raw.githubusercontent.com/kubernets-sigs/aws-load-balancer-controller/main/docs/install/iam_policy.json</pre> <p>CLI를 사용하여 AWS 계정에서 정책을 생성합니다.</p> <pre data-bbox="594 909 1026 1224">aws iam create-policy \ --policy-name AWSLoadBalancerControllerIAMPolicy \ --policy-document file://iam-policy.json</pre> <p>다음과 같이 출력되어야 합니다.</p> <pre data-bbox="594 1381 1026 1875">{ "Policy": { "PolicyName": "AWSLoadBalancerControllerIAMPolicy", "PolicyId": "<YOUR_POLICY_ID>", "Arn": "arn:aws:iam:<YOUR-ACCOUNT-ID>:policy/AWSLoadBalancerControllerIAMPolicy",</pre>	<p>앱 개발자, AWS DevOps, AWS 시스템 관리자</p>

작업	설명	필요한 기술
	<pre data-bbox="609 210 1015 861"> "Path": "/", "DefaultVersionId": "v1", "AttachmentCount": 0, "PermissionsBoundaryUsageCount": 0, "IsAttachable": true, "CreateDate": "<YOUR-DATE>", "UpdateDate": "<YOUR-DATE>" } } </pre> <p data-bbox="592 892 1031 1029">정책의 Amazon 리소스 이름 (ARN)를 \$POLICY_ARN 로서 저장합니다.</p> <pre data-bbox="609 1060 1015 1344"> export POLICY_ARN="arn:aws:iam::<YOUR-ACCOUNT-ID>:policy/AWSLoadBalancerControllerIAMPolicy" </pre>	

작업	설명	필요한 기술
IAM 서비스 계정을 생성합니다.	<p>aws-load-balancer-controller 이라는 IAM 서비스 계정을 kube-system 네임스페이스에 생성합니다. 이전에 구성한 CLUSTER_NAME , AWS_REGION , POLICY_ARN (을)를 사용합니다.</p> <pre data-bbox="597 636 1027 1234"> eksctl create iamserviceaccount \ --cluster=\$CLUSTER_NAME \ --region=\$AWS_REGION \ --attach-policy-arn=\$POLICY_ARN \ --namespace=kube-system \ --name=aws-load-balancer-controller \ --override-existing-serviceaccounts \ --approve </pre> <p>생성을 검증합니다.</p> <pre data-bbox="597 1346 1027 1738"> eksctl get iamserviceaccount \ --cluster \$CLUSTER_NAME \ --name aws-load-balancer-controller \ --namespace kube-system \ --output yaml </pre> <p>출력은 다음과 같아야합니다.</p>	앱 개발자, AWS DevOps, AWS 시스템 관리자

작업	설명	필요한 기술
	<pre> - metadata: name: aws-load-balancer-controller namespace: kube-system status: roleARN: arn:aws:iam::<YOUR-ACCOUNT-ID>:role/eksctl-my-fargate-addon-iam-serviceaccount-kubernetes-Role1-<YOUR-ROLE-ID> wellKnownPolicies: autoScaler: false awsLoadBalancerController: false certManager: false ebsCSIDriver: false efsCSIDriver: false externalDNS: false imageBuilder: false </pre>	

작업	설명	필요한 기술
<p>로드 밸런서 컨트롤러를 설치합니다.</p>	<p>Helm 리포지토리를 업데이트합니다.</p> <pre data-bbox="594 348 1029 428">helm repo update</pre> <p>Helm 리포지토리에 Amazon EKS 차트 리포지토리를 추가합니다.</p> <pre data-bbox="594 634 1029 793">helm repo add eks https://aws.github .io/eks-charts</pre> <p>배경에서 Load Balancer Controller eks-chart가 사용하는 Kubernetes 사용자 지정 리소스 정의(CRD)를 적용합니다.</p> <pre data-bbox="594 1045 1029 1323">kubectl apply -k "github.com/aws/ek s-charts/stable/aw s-load-balancer-co ntroller//crds?ref =master"</pre> <p>출력은 다음과 같아야합니다.</p> <pre data-bbox="594 1432 1029 1862">customresourcedefi nition.apiextensio ns.k8s.io/ingressc lassparams.elbv2.k 8s.aws created customresourcedefin ition.apiextension s.k8s.io/targetgro upbindings.elbv2.k 8s.aws created</pre>	<p>앱 개발자, AWS DevOps, AWS 시스템 관리자</p>

작업	설명	필요한 기술
	<p>이전에 설정한 환경 변수를 사용하여 Helm 차트를 설치합니다.</p> <pre data-bbox="597 380 1029 1014">helm install aws-load-balancer-controller eks/aws-load-balancer-controller \ --set clusterName=\$CLUSTER_NAME \ --set serviceAccount.create=false \ --set region=\$AWS_REGION \ --set vpcId=\$VPC_ID \ --set serviceAccount.name=aws-load-balancer-controller \ -n kube-system</pre> <p>출력은 다음과 같아야 합니다.</p> <pre data-bbox="597 1125 1029 1602">NAME: aws-load-balancer-controller LAST DEPLOYED: <YOUR-DATE> NAMESPACE: kube-system STATUS: deployed REVISION: 1 TEST SUITE: None NOTES: AWS Load Balancer controller installed!</pre>	

작업	설명	필요한 기술
<p>NGINX 서비스를 생성합니다.</p>	<p>nginx-service.yaml 파일을 사용하여 NGINX 포드를 노출할 서비스를 생성합니다.</p> <pre>kubectl apply -f nginx-service.yaml</pre> <p>출력은 다음과 같아야합니다.</p> <pre>service/nginx-service created</pre>	<p>앱 개발자, AWS DevOps, AWS 시스템 관리자</p>
<p>Kubernetes 인그레스 리소스를 생성합니다.</p>	<p>nginx-ingress.yaml 파일을 사용하여 Kubernetes NGINX 인그레스를 노출할 서비스를 생성합니다.</p> <pre>kubectl apply -f nginx-ingress.yaml</pre> <p>출력은 다음과 같아야합니다.</p> <pre>ingress.networking .k8s.io/nginx-ingress created</pre>	<p>앱 개발자, AWS DevOps, AWS 시스템 관리자</p>

작업	설명	필요한 기술
로드 밸런서 URL을 가져옵니다.	<p>인그레스 정보를 검색하려면 다음 명령을 사용합니다.</p> <pre>kubectl get ingress nginx-ingress</pre> <p>출력은 다음과 같아야 합니다.</p> <pre>NAME CLASS HOSTS ADDRESS PORTS AGE nginx-ingress <none> * k8s-defau lt-nginxing-xxx.us -east-1.elb.amazon aws.com 80 80s</pre> <p>출력에서 ADDRESS(예: k8s-default-nginxing-xxx.us-east-1.elb.amazonaws.com)(을)를 복사하여 브라우저에 붙여 넣으면 index.html 파일에 액세스할 수 있습니다.</p>	앱 개발자, AWS DevOps, AWS 시스템 관리자

실행 중인 컨테이너의 디버깅

작업	설명	필요한 기술
포드를 선택합니다.	<p>모든 포드를 나열하고 원하는 포드의 이름을 복사합니다.</p> <pre>kubectl get pods</pre>	앱 개발자, AWS DevOps, AWS 시스템 관리자

작업	설명	필요한 기술
	<p>출력은 다음과 같아야 합니다.</p> <pre> NAME READY STATUS RESTARTS AGE nginx-deployment- xxxx-aaa 1/1 Running 0 55m nginx-deployment- xxxx-bbb 1/1 Running 0 55m nginx-deployment- xxxx-ccc 1/1 Running 0 55m nginx-deployment- xxxx-ddd 1/1 Running 0 42m </pre> <p>이 명령은 기존 포드와 추가 정보를 나열합니다.</p> <p>특정 포드에 관심이 있는 경우 <code>POD_NAME</code> 변수에 대해 사용자가 흥미있어 하는 포드의 이름을 입력하거나 환경 변수로 설정합니다. 그렇지 않다면, 이 파라미터를 생략하여 모든 리소스를 조회합니다.</p> <pre> export POD_NAME="nginx- deployment-<YOUR-POD- NAME>" </pre>	

작업	설명	필요한 기술
로그에 액세스합니다.	<p>디버깅하려는 포드에서 로그를 가져옵니다.</p> <pre>kubectl logs \$POD_NAME</pre>	앱 개발자, AWS 시스템 관리자, AWS DevOps
NGINX 포트를 전달합니다.	<p>포트 전달을 사용하여, NGINX 웹 서버에 액세스하기 위한 포드의 포트를 로컬 시스템의 포트에 매핑합니다.</p> <pre>kubectl port-forward deployment/nginx-deployment 8080:80</pre> <p>브라우저에서 다음 URL을 엽니다.</p> <pre>http://localhost:8080</pre> <p>port-forward 명령을 사용하면 로드 밸런서를 통해 index.html 파일을 공개적으로 사용할 수 있게 하지 않고도 그 파일에 액세스할 수 있습니다. 이는 디버깅하는 동안 실행 중인 애플리케이션에 액세스하는 데 유용합니다. 키보드 명령 Ctrl+C를 눌러 포트 전달을 중지할 수 있습니다.</p>	앱 개발자, AWS DevOps, AWS 시스템 관리자

작업	설명	필요한 기술
<p>포드 내에서 명령을 실행합니다.</p>	<p>현재 <code>index.html</code> 파일을 보려면 다음 명령을 사용합니다.</p> <pre>kubectl exec \$POD_NAME -- cat /usr/share/nginx/html/index.html</pre> <p><code>exec</code> 명령을 사용하여 포드에서 직접 모든 명령을 내릴 수 있습니다. 이는 실행 중인 애플리케이션을 디버깅하는 데 유용합니다.</p>	<p>앱 개발자, AWS DevOps, AWS 시스템 관리자</p>
<p>파일을 포드에 복사합니다.</p>	<p>이 포드에서 기본 <code>index.html</code> 파일을 제거합니다.</p> <pre>kubectl exec \$POD_NAME -- rm /usr/share/nginx/html/index.html</pre> <p>사용자 지정된 로컬 파일 <code>index.html</code> (을)를 포드에 업로드합니다.</p> <pre>kubectl cp index.html \$POD_NAME:/usr/share/nginx/html/</pre> <p><code>cp</code> 명령을 사용하여 모든 포드에 파일을 직접 변경하거나 추가할 수 있습니다.</p>	<p>앱 개발자, AWS DevOps, AWS 시스템 관리자</p>

작업	설명	필요한 기술
포트 전달을 사용하여 변경 내용을 표시합니다.	<p>포트 전달을 사용하여 이 포드에 대해 변경한 내용을 검증합니다.</p> <pre>kubectl port-forward pod/\$POD_NAME 8080:80</pre> <p>사용자의 브라우저에서 다음 URL을 엽니다.</p> <pre>http://localhost:8080</pre> <p>index.html 파일에 적용되는 변경 내용이 브라우저에 표시되어야 합니다.</p>	앱 개발자, AWS DevOps, AWS 시스템 관리자

리소스 삭제하기

작업	설명	필요한 기술
로드 밸런서를 삭제합니다.	<p>인그레스를 삭제합니다.</p> <pre>kubectl delete ingress/nginx-ingress</pre> <p>출력은 다음과 같아야 합니다.</p> <pre>ingress.networking.k8s.io "nginx-ingress" deleted</pre> <p>서비스를 삭제합니다.</p> <pre>kubectl delete service/nginx-service</pre>	앱 개발자, AWS DevOps, AWS 시스템 관리자

작업	설명	필요한 기술
	<p>출력은 다음과 같아야합니다.</p> <pre>service "nginx-service" deleted</pre> <p>로드 밸런서 컨트롤러를 삭제합니다.</p> <pre>helm delete aws-load-balancer-controller -n kube-system</pre> <p>출력은 다음과 같아야합니다.</p> <pre>release "aws-load-balancer-controller" uninstalled</pre> <p>서비스 계정을 삭제합니다.</p> <pre>eksctl delete iamserviceaccount --cluster \$CLUSTER_NAME --namespace kube-system --name aws-load-balancer-controller</pre>	

작업	설명	필요한 기술
<p>배포를 삭제합니다.</p>	<p>배포 리소스를 삭제하려면 다음 명령을 사용합니다.</p> <pre>kubectl delete deploy/nginx-deployment</pre> <p>출력은 다음과 같아야 합니다.</p> <pre>deployment.apps "nginx-deployment" deleted</pre>	<p>앱 개발자, AWS DevOps, AWS 시스템 관리자</p>
<p>클러스터를 삭제합니다.</p>	<p>다음 명령을 사용하여 EKS 클러스터를 삭제합니다. 여기서 <code>my-fargate</code> (은)는 클러스터 이름입니다.</p> <pre>eksctl delete cluster --name \$CLUSTER_NAME</pre> <p>이 명령은 모든 관련 리소스를 포함한 전체 클러스터를 삭제합니다.</p>	<p>앱 개발자, AWS DevOps, AWS 시스템 관리자</p>
<p>IAM 정책을 삭제합니다.</p>	<p>CLI를 사용하여 이전에 생성한 정책을 삭제합니다.</p> <pre>aws iam delete-policy --policy-arn \$POLICY_ARN</pre>	<p>앱 개발자, AWS 관리자, AWS DevOps</p>

문제 해결

문제	Solution
<p>클러스터대상 가용 영역에 클러스터를 지원하는 데 필요한 용량이 충분하지 않다는 클러스터 생성 시 오류 메시지가 나타납니다. 다음과 비슷한 메시지가 나타나야 합니다.</p> <div data-bbox="115 558 792 919" style="border: 1px solid #ccc; border-radius: 10px; padding: 10px; margin-top: 10px;"> <pre>Cannot create cluster 'my-fargate' because us-east-1e, the targeted availability zone, does not currently have sufficient capacity to support the cluster. Retry and choose from these availability zones: us-east-1a, us-east-1b, us-east-1c, us-east-1d, us-east-1f</pre> </div>	<p>오류 메시지에서 권장 가용 영역을 사용하여 클러스터를 다시 생성합니다. <code>clusterconfig-fargate.yaml</code> 파일의 마지막 줄에서 가용 영역 목록을 지정합니다(예: <code>availabilityZones: ["us-east-1a", "us-east-1b", "us-east-1c"]</code>).</p>

관련 리소스

- [Amazon EKS 설명서](#)
- [Amazon EKS에서의 애플리케이션 로드 밸런싱](#)
- [EKS 모범 사례 가이드](#)
- [로드 밸런서 컨트롤러 설명서](#)
- [eksctl 설명서](#)
- [Amazon ECR 퍼블릭 갤러리 NGINX 이미지](#)
- [Helm 설명서](#)
- [실행 중인 포드의 디버깅](#)(Kubernetes 설명서)
- [Amazon EKS 워크샵](#)
- [EKS 클러스터 생성 오류](#)

추가 정보

clusterconfig-fargate.yaml

```
apiVersion: eksctl.io/v1alpha5
kind: ClusterConfig

metadata:
  name: my-fargate
  region: us-east-1

fargateProfiles:
  - name: fp-default
    selectors:
      - namespace: default
      - namespace: kube-system
```

nginx-deployment.yaml

```
apiVersion: apps/v1
kind: Deployment
metadata:
  name: "nginx-deployment"
  namespace: "default"
spec:
  replicas: 3
  selector:
    matchLabels:
      app: "nginx"
  template:
    metadata:
      labels:
        app: "nginx"
    spec:
      containers:
        - name: nginx
          image: public.ecr.aws/nginx/nginx:latest
          ports:
            - containerPort: 80
```

nginx-service.yaml

```
apiVersion: v1
kind: Service
metadata:
  annotations:
```

```

  alb.ingress.kubernetes.io/target-type: ip
  name: "nginx-service"
  namespace: "default"
spec:
  ports:
  - port: 80
    targetPort: 80
    protocol: TCP
  type: NodePort
  selector:
    app: "nginx"

```

nginx-ingress.yaml

```

apiVersion: networking.k8s.io/v1
kind: Ingress
metadata:
  namespace: "default"
  name: "nginx-ingress"
  annotations:
    kubernetes.io/ingress.class: alb
    alb.ingress.kubernetes.io/scheme: internet-facing
spec:
  rules:
  - http:
      paths:
      - path: /
        pathType: Prefix
        backend:
          service:
            name: "nginx-service"
            port:
              number: 80

```

index.html

```

<!DOCTYPE html>
<html>

<body>
  <h1>Welcome to your customized nginx!</h1>
  <p>You modified the file on this running pod</p>
</body>

```

```
</html>
```

Elastic Beanstalk를 사용하여 컨테이너 배포

작성자: Thomas Scott(AWS) 및 Jean-Baptiste Guillois(AWS)

코드 리포지토리: [클러스터 샘플 앱](#)

환경: 프로덕션

기술: 컨테이너 및 마이크로서비스, 클라우드 네이티브, 현대화

AWS 서비스: AWS Elastic Beanstalk

요약

Amazon Web Services(AWS) 클라우드에서 AWS Elastic Beanstalk는 Docker를 사용 가능한 플랫폼으로 지원하므로, 생성된 환경에서 컨테이너를 실행할 수 있습니다. 이 패턴은 Elastic Beanstalk 서비스를 사용하여 컨테이너를 배포하는 방법을 보여줍니다. 이 패턴을 배포할 때는 Docker 플랫폼 기반 웹 서버 환경을 사용합니다.

Elastic Beanstalk를 사용하여 웹 애플리케이션과 서비스를 배포하고 확장하려는 경우, 코드를 업로드하면 배포가 자동으로 처리됩니다. 용량 프로비저닝, 로드 밸런싱, 자동 조정 및 애플리케이션 상태 모니터링도 포함됩니다. Elastic Beanstalk를 사용하면 사용자 대신 Elastic Beanstalk에서 생성하는 AWS 리소스를 완전히 제어할 수 있습니다. Elastic Beanstalk에 대한 추가 비용은 없습니다. 애플리케이션을 저장하고 실행하는 데 사용된 AWS 리소스에 대한 비용만 지불하면 됩니다.

이 패턴에는 [AWS Elastic Beanstalk Command Line Interface\(EB CLI\)](#) 및 AWS Management Console을 사용하여 배포하기 위한 지침이 포함되어 있습니다.

사용 사례

Elastic Beanstalk의 사용 사례는 다음과 같습니다.

- 프로토타입 환경을 배포하여 프론트엔드 애플리케이션을 시연하십시오. (이 패턴은 Dockerfile을 예로 사용합니다.)
- API를 배포하여 지정된 도메인에 대한 API 요청을 처리하십시오.
- Docker-Compose(이 패턴의 실제 예제에서는 `docker-compose.yml`이 사용되지 않음)를 사용하여 오케스트레이션 솔루션을 배포합니다.

사전 조건 및 제한 사항

사전 조건

- AWS 계정
- AWS EB CLI가 로컬에 설치됨
- Docker가 로컬 머신에 설치됨

제한 사항

- 무료 요금제에서는 IP 주소당 6시간당 100회의 풀로 Docker 풀 제한이 있습니다.

아키텍처

대상 기술 스택

- Amazon Elastic Compute Cloud(Amazon EC2) 인스턴스
- 보안 그룹
- Application Load Balancer
- Auto Scaling 그룹

대상 아키텍처

자동화 및 규모 조정

AWS Elastic Beanstalk는 요청의 수에 따라 자동으로 규모가 조정됩니다. 환경에 대해 생성된 AWS 리소스에는 Application Load Balancer, Auto Scaling 그룹, 하나 이상의 Amazon EC2 인스턴스가 포함됩니다.

로드 밸런서는 Auto Scaling 그룹의 일부인 Amazon EC2 인스턴스의 앞에 위치합니다. Amazon EC2 Auto Scaling은 추가 Amazon EC2 인스턴스를 자동으로 시작하여 애플리케이션의 증가하는 로드를 처리합니다. 애플리케이션의 로드가 감소하면 Amazon EC2 Auto Scaling은 인스턴스를 중지하지만 최소한의 개의 인스턴스는 실행 상태로 유지합니다.

자동 크기 조정 트리거

Elastic Beanstalk 환경의 Auto Scaling 그룹은 CloudWatch 두 개의 아마존 경보를 사용하여 조정 작업을 시작합니다. 기본 트리거는 각 인스턴스의 평균 아웃바운드 네트워크 트래픽이 5분 이상 6MB보다 높거나 2MB보다 낮은 경우 규모 조정을 트리거합니다. Amazon EC2 Auto Scaling을 효과적으로 사용하려면 애플리케이션, 인스턴스 유형 및 서비스 요구 사항에 적절한 트리거를 구성합니다. 지연 시간, 디스크 I/O, CPU 사용률 및 요청 수 등 여러 통계를 기준으로 규모를 조정할 수 있습니다. 자세한 정보는 [Auto Scaling 트리거](#)를 참조하십시오.

도구

서비스

- [AWS Command Line Interface\(AWS CLI\)](#)는 명령줄 셸에서 명령을 사용하여 AWS 서비스와 상호 작용할 수 있는 오픈 소스 도구입니다.
- [AWS EB Command Line Interface\(EB CLI\)](#)는 Elastic Beanstalk 환경을 생성, 구성 및 관리하는 데 사용할 수 있는 명령줄 클라이언트입니다.
- [Elastic Load Balancing\(ELB\)](#)은 들어오는 애플리케이션 또는 네트워크 트래픽을 여러 대상에 분산합니다. 예를 들어 하나 이상의 가용 영역에 있는 Amazon Elastic Compute Cloud(Amazon EC2) 인스턴스, 컨테이너, IP 주소 전반에 걸쳐 트래픽을 분산할 수 있습니다.

기타 서비스

- [Docker](#)는 소프트웨어를 라이브러리, 시스템 도구, 코드 및 런타임을 포함하는 컨테이너라는 표준화된 유닛으로 패키징합니다.

코드

[이 패턴의 코드는 클러스터 샘플 애플리케이션 리포지토리에서 사용할 수 있습니다. GitHub](#)

에픽

Dockerfile을 사용하여 구축

작업	설명	필요한 기술
원격 리포지토리를 복제합니다.	<ul style="list-style-type: none"> 리포지토리를 복제하려면 <code>git clone https://github.com/aws-samples/cluster-sample</code> 	앱 개발자, AWS 관리자, AWS DevOps

작업	설명	필요한 기술
	<p>-app.git 명령을 실행합니다.< /p></p>	
<p>Elastic Beanstalk Docker 프로젝트를 초기화합니다.</p>	<ol style="list-style-type: none"> 루트에서 aws.json이라는 파일을 생성하십시오. aws.json 파일에 다음 코드를 추가합니다. <pre data-bbox="630 562 1029 1276"> { "AWSEBDoc kerrunVersion":"1", "Image":{ "Name":"c luster-sample-app" }, "Ports":[{ "ContainerPort":80 , "HostPort":8080 }] } </pre> 프로젝트 루트에서 eb init -p docker 명령을 실행합니다. 	<p>앱 개발자, AWS 관리자, AWS DevOps</p>
<p>프로젝트를 로컬에서 테스트하십시오.</p>	<ol style="list-style-type: none"> 프로젝트 루트에서 eb local run 명령을 실행합니다. http://localhost 로 이동하여 애플리케이션을 테스트합니다. 	<p>앱 개발자, AWS 관리자, AWS DevOps</p>

EB CLI를 사용하여 배포

작업	설명	필요한 기술
배포 명령 실행	1. 프로젝트 루트에서 <code>eb create docker-sample-cluster-app</code> 명령을 실행합니다.	앱 개발자, AWS 관리자, AWS DevOps
배포된 버전에 액세스하십시오.	배포 명령이 완료되면 <code>eb open</code> 명령어를 사용하여 프로젝트에 액세스합니다.	앱 개발자, AWS 관리자, AWS DevOps

콘솔을 사용하여 배포

작업	설명	필요한 기술
브라우저를 사용하여 애플리케이션을 배포합니다.	<ol style="list-style-type: none"> 콘솔을 엽니다. Elastic Beanstalk 콘솔로 이동합니다. 애플리케이션 생성을 선택합니다. 애플리케이션 이름에 <code>Cluster-Sample-App</code>을 입력합니다. Docker를 플랫폼으로 선택합니다. 코드 업로드를 선택합니다. 로컬 <code>.zip</code> 파일(복제된 프로젝트의 루트에 있음) 또는 Amazon Simple Storage Service(S3) 공개 URL을 선택합니다. 	앱 개발자, AWS 관리자, AWS DevOps

작업	설명	필요한 기술
배포된 버전에 액세스하십시오.	배포 후 배포된 애플리케이션에 액세스하고 제공된 URL을 선택합니다.	앱 개발자, AWS 관리자, AWS DevOps

관련 리소스

- [웹 서버 환경](#)
- [macOS에 EB CLI 설치](#)
- [수동으로 EB CLI 설치](#)

추가 정보

Elastic Beanstalk 사용의 이점

- 자동 인프라 프로비저닝
- 기본 플랫폼의 자동 관리
- 애플리케이션 지원을 위한 자동 패치 및 업데이트
- 애플리케이션 자동 조정
- 노드 수를 사용자 지정하는 기능
- 필요한 경우 인프라 구성 요소에 액세스하는 기능
- 다른 컨테이너 배포 솔루션 대비 쉬운 배포

Lambda 함수, Amazon VPC 및 서버리스 아키텍처를 사용하여 정적 아웃바운드 IP 주소 생성

작성자: Thomas Scott(AWS)

환경: 프로덕션

기술: 컨테이너 및 마이크로서비스, 소프트웨어 개발 및 테스트

AWS 서비스: AWS Lambda

요약

이 패턴은 서버리스 아키텍처로 Amazon Web Services(AWS) 클라우드에서 고정 아웃바운드 IP 주소를 생성하는 방법을 설명합니다. 조직에서 Secure File Transfer Protocol(SFTP)을 사용하여 별도의 사업체에 파일을 보내려는 경우 이 접근 방식을 활용할 수 있습니다. 즉, 비즈니스 엔터티는 방화벽을 통해 파일을 허용하는 IP 주소에 액세스할 수 있어야 합니다.

이 패턴의 접근 방식은 [엘라스틱 IP 주소를 아웃바운드 IP](#) 주소로 사용하는 AWS Lambda 함수를 생성하는 데 도움이 됩니다. 이 패턴의 단계를 따르면 고정 IP 주소를 사용하는 인터넷 게이트웨이를 통해 아웃바운드 트래픽을 라우팅하는 Lambda 함수와 Virtual Private Cloud(VPC)를 생성할 수 있습니다. 고정 IP 주소를 사용하려면 Lambda 함수를 VPC와 서브넷에 연결합니다.

사전 조건 및 제한 사항

사전 조건

- 활성 상태의 AWS 계정.
- Lambda 함수를 생성 및 배포하고 VPC 및 서브넷을 생성할 수 있는 AWS Identity and Access(IAM) 권한. 이에 대한 자세한 내용은 AWS Lambda 설명서의 [실행 역할 및 사용자 권한](#)을 참조하십시오.
- IaC (코드형 인프라) 를 사용하여 이 패턴의 접근 방식을 구현하려면 AWS Cloud9와 같은 통합 개발 환경 (IDE) 이 필요합니다. 이에 대한 자세한 내용은 AWS Cloud9 설명서의 [AWS Cloud9란 무엇인가요?](#)를 참조하세요

아키텍처

다음 다이어그램은 이 패턴의 서버리스 아키텍처입니다.

이 다이어그램은 다음 워크플로를 보여줍니다.

1. 아웃바운드 트래픽은 NAT gateway 1를 Public subnet 1에 남깁니다.
2. 아웃바운드 트래픽은 NAT gateway 2를 Public subnet 2에 남깁니다.
3. Lambda 함수는 Private subnet 1 또는 Private subnet 2에서 실행할 수 있습니다.
4. Private subnet 1 및 Private subnet 2는 퍼블릭 서브넷의 NAT 게이트웨이로 트래픽을 라우팅합니다.
5. NAT 게이트웨이는 퍼블릭 서브넷에서 인터넷 게이트웨이로 아웃바운드 트래픽을 전송합니다.
6. 아웃바운드 데이터는 인터넷 게이트웨이에서 외부 서버로 전송됩니다.

기술 스택

- Lambda
- Amazon Virtual Private Cloud(VPC)

자동화 및 규모 조정

서로 다른 가용 영역에서 두 개의 퍼블릭 서브넷과 두 개의 프라이빗 서브넷을 사용하여 고가용성(HA)을 보장할 수 있습니다. 하나의 가용 영역을 사용할 수 없게 되더라도 패턴 솔루션은 계속 작동합니다.

도구

- [AWS Lambda](#) - AWS Lambda는 서버를 프로비저닝하거나 관리하지 않고도 코드를 실행할 수 있게 해주는 컴퓨팅 서비스입니다. Lambda는 필요 시에만 코드를 실행하며, 일일 몇 개의 요청에서 초당 수천 개의 요청까지 자동으로 규모를 조정합니다. 사용한 컴퓨팅 시간만큼만 비용을 지불하고, 코드가 실행되지 않을 때는 요금이 부과되지 않습니다.
- [Amazon VPC](#) - Amazon Virtual Private Cloud(VPC)를 사용하면 정의한 가상 네트워크에서 AWS 리소스를 시작할 수 있도록 클라우드의 논리적으로 격리된 섹션을 프로비저닝할 수 있습니다. 이 가상 네트워크는 AWS의 확장 가능한 인프라를 사용한다는 이점과 함께 고객의 자체 데이터 센터에서 운영하는 기존 네트워크와 매우 유사합니다.

에픽

새 VPC 생성

작업	설명	필요한 기술
새 VPC를 생성합니다.	<p>AWS Management Console에 로그인하고 Amazon VPC 콘솔을 연 다음 IPv4 CIDR 범위로 10.0.0.0/25 를 가진 Lambda VPC 이름의 VPC를 생성합니다.</p> <p>VPC 생성에 대한 자세한 내용은 Amazon VPC 설명서의 Amazon VPC 시작하기를 참조하십시오.</p>	AWS 관리자

퍼블릭 서브넷 2개 생성

작업	설명	필요한 기술
첫 번째 퍼블릭 서브넷을 생성합니다.	<ol style="list-style-type: none"> 1. Amazon VPC 콘솔에서 서브넷을 선택한 다음 서브넷 생성을 선택합니다. 2. 이름 태그에 public-one 을 입력합니다. 3. VPC에 Lambda VPC를 선택합니다. 4. 가용 영역을 선택하고 기록하십시오. 5. IPv4 CIDR 블록의 경우 10.0.0.0/28 을 입력하고 서브넷 생성을 선택합니다. 	AWS 관리자

작업	설명	필요한 기술
두 번째 퍼블릭 서브넷을 생성합니다.	<ol style="list-style-type: none"> 1. Amazon VPC 콘솔에서 서브넷을 선택한 다음 서브넷 생성을 선택합니다. 2. 이름 태그에 public-two 을 입력합니다. 3. VPC에 Lambda VPC를 선택합니다. 4. 가용 영역을 선택하고 기록하십시오. 중요: public-one 서브넷이 포함된 가용 영역을 사용할 수 없습니다. 5. IPv4 CIDR 블록의 경우 10.0.0.16/28 를 입력하고 서브넷 생성을 선택합니다. 	AWS 관리자

프라이빗 서브넷 2개 생성

작업	설명	필요한 기술
첫 번째 프라이빗 서브넷을 생성합니다.	<ol style="list-style-type: none"> 1. Amazon VPC 콘솔에서 서브넷을 선택한 다음 서브넷 생성을 선택합니다. 2. 이름 태그에 private-one 을 입력합니다. 3. VPC에 Lambda VPC를 선택합니다. 4. 이전에 생성한 public-one 서브넷이 포함된 가용 영역을 선택합니다. 5. IPv4 CIDR 블록의 경우 10.0.0.32/28 을 입력하 	AWS 관리자

작업	설명	필요한 기술
	고 서브넷 생성을 선택합니다.	
두 번째 프라이빗 서브넷을 생성합니다.	<ol style="list-style-type: none"> 1. Amazon VPC 콘솔에서 서브넷을 선택한 다음 서브넷 생성을 선택합니다. 2. 이름 태그에 <code>private-two</code> 을 입력합니다. 3. VPC에 Lambda VPC를 선택합니다. 4. 이전에 생성한 <code>public-two</code> 서브넷이 포함된 동일한 가용 영역을 선택합니다. 5. IPv4 CIDR 블록의 경우 <code>10.0.0.64/28</code> 을 입력하고 서브넷 생성을 선택합니다. 	AWS 관리자

NAT 게이트웨이에 대한 Elastic IP 주소 2개 생성

작업	설명	필요한 기술
첫 번째 Elastic IP 주소를 생성합니다.	<ol style="list-style-type: none"> 1. Amazon VPC 콘솔에서 Elastic IP를 선택한 다음 새 주소 할당을 선택합니다. 2. 할당을 선택하고 새로 만든 Elastic IP 주소의 할당 ID를 기록합니다. <p>참고: 이 Elastic IP 주소는 첫 번째 NAT 게이트웨이에 사용 됩니다.</p>	AWS 관리자

작업	설명	필요한 기술
두 번째 Elastic IP 주소를 생성합니다.	<ol style="list-style-type: none"> 1. Amazon VPC 콘솔에서 Elastic IP를 선택한 다음 새 주소 할당을 선택합니다. 2. 할당을 선택하고 이 두 번째 Elastic IP 주소의 할당 ID를 기록합니다. <p>참고: 이 Elastic IP 주소는 두 번째 NAT 게이트웨이에 사용 됩니다.</p>	AWS 관리자

인터넷 게이트웨이 생성

작업	설명	필요한 기술
인터넷 게이트웨이를 만듭니다.	<ol style="list-style-type: none"> 1. Amazon VPC 콘솔에서 인터넷 게이트웨이를 선택한 후 인터넷 게이트웨이 생성을 선택합니다. 2. 이름으로 <code>Lambda internet gateway</code>를 입력한 다음 인터넷 게이트웨이 생성을 선택합니다. 인터넷 게이트웨이 ID를 기록했는지 확인하십시오. 	AWS 관리자
인터넷 게이트웨이를 VPC에 연결합니다.	방금 생성한 인터넷 게이트웨이를 선택한 후 Actions, Attach to VPC(작업, VPC에 연결)을 선택합니다.	AWS 관리자

NAT 게이트웨이 2개 생성

작업	설명	필요한 기술
첫 번째 NAT 게이트웨이를 생성합니다.	<ol style="list-style-type: none"> 1. Amazon VPC 콘솔에서 NAT 게이트웨이를 선택한 다음 NAT 게이트웨이 생성을 선택합니다. 2. NAT 게이트웨이의 이름으로 nat-one을 입력합니다. 3. NAT 게이트웨이를 생성할 서브넷으로 public-one 을 선택합니다. 4. 연결 유형으로는 퍼블릭을 선택합니다. 5. Elastic IP 할당 ID에서 앞서 만든 Elastic IP 주소를 선택하고 이 주소를 NAT 게이트웨이와 연결합니다. 6. NAT 게이트웨이 생성을 선택합니다. 	AWS 관리자
두 번째 NAT 게이트웨이를 생성합니다.	<ol style="list-style-type: none"> 1. Amazon VPC 콘솔에서 NAT 게이트웨이를 선택한 다음 NAT 게이트웨이 생성을 선택합니다. 2. NAT 게이트웨이의 이름으로 nat-two를 입력합니다. 3. NAT 게이트웨이를 생성할 서브넷으로 public-two 를 선택합니다. 4. 연결 유형으로는 퍼블릭을 선택합니다. 5. Elastic IP 할당 ID에서 앞서 만든 두 번째 Elastic IP 주소 	AWS 관리자

작업	설명	필요한 기술
	<p>를 NAT 게이트웨이와 연결합니다.</p> <p>6. NAT 게이트웨이 생성을 선택합니다.</p>	

퍼블릭 및 프라이빗 서브넷을 위한 라우팅 테이블을 생성합니다.

작업	설명	필요한 기술
퍼블릭 1 서브넷의 라우팅 테이블을 생성합니다.	<ol style="list-style-type: none"> 1. Amazon VPC 콘솔에서 라우팅 테이블을 선택한 다음 라우팅 테이블 생성을 선택합니다. 2. 라우팅 테이블 이름으로 <code>public-one-subnet</code> 을 입력한 다음 라우팅 테이블 생성을 선택합니다. 3. <code>public-one-subnet</code> 라우팅 테이블을 선택하고, 라우팅 편집을 선택한 다음 라우팅 추가를 선택합니다. 4. 대상란에서 <code>0.0.0.0</code>을 지정한 다음 대상 목록에서 인터넷 게이트웨이 ID를 선택합니다. 5. 서브넷 연결 탭에서 서브넷 연결 편집을 선택하고, <code>10.0.0.0/28</code> CIDR 범위가 있는 <code>public-one</code> 서브넷을 선택한 다음 연결 저장을 선택합니다. 6. 변경 사항 저장(Save Changes)을 선택합니다. 	AWS 관리자

작업	설명	필요한 기술
퍼블릭 2 서브넷의 라우팅 테이블 생성	<ol style="list-style-type: none"> 1. Amazon VPC 콘솔에서 라우팅 테이블을 선택한 다음 라우팅 테이블 생성을 선택합니다. 2. 라우팅 테이블 이름으로 <code>public-two-subnet</code> 을 입력한 다음 라우팅 테이블 생성을 선택합니다. 3. <code>public-two-subnet</code> 라우팅 테이블을 선택하고, 라우팅 편집을 선택한 다음 라우팅 추가를 선택합니다. 4. 대상란에서 <code>0.0.0.0</code>을 지정한 다음 대상 목록에서 인터넷 게이트웨이 ID를 선택합니다. 5. 서브넷 연결 탭에서 서브넷 연결 편집을 선택하고, CIDR <code>10.0.0.16/28</code> 범위가 있는 <code>public-two</code> 서브넷을 선택한 다음 연결 저장을 선택합니다. 6. 변경 사항 저장(Save Changes)을 선택합니다. 	AWS 관리자

작업	설명	필요한 기술
<p>프라이빗 1 서브넷의 라우팅 테이블을 생성합니다.</p>	<ol style="list-style-type: none"> 1. Amazon VPC 콘솔에서 라우팅 테이블을 선택한 다음 라우팅 테이블 생성을 선택합니다. 2. 라우팅 테이블 이름으로 <code>private-one-subnet</code> 을 입력한 다음 라우팅 테이블 생성을 선택합니다. 3. <code>private-one-subnet</code> 라우팅 테이블을 선택하고, 라우팅 편집을 선택한 다음 라우팅 추가를 선택합니다. 4. 대상란에서 <code>0.0.0.0</code>을 지정한 다음 대상 목록의 <code>public-one</code> 서브넷에서 NAT 게이트웨이를 선택합니다. 5. 서브넷 연결 탭에서 서브넷 연결 편집을 선택하고, <code>10.0.0.32/28</code> CIDR 범위가 있는 <code>private-one</code> 서브넷을 선택한 다음 연결 저장을 선택합니다. 6. 변경 사항 저장(Save Changes)을 선택합니다. 	<p>AWS 관리자</p>

작업	설명	필요한 기술
프라이빗 2 서브넷의 라우팅 테이블을 생성합니다.	<ol style="list-style-type: none"> 1. Amazon VPC 콘솔에서 라우팅 테이블을 선택한 다음 라우팅 테이블 생성을 선택합니다. 2. 라우팅 테이블 이름으로 <code>private-two-subnet</code> 을 입력한 다음 라우팅 테이블 생성을 선택합니다. 3. <code>private-two-subnet</code> 라우팅 테이블을 선택하고, 라우팅 편집을 선택한 다음 라우팅 추가를 선택합니다. 4. 대상란에서 <code>0.0.0.0</code>을 지정한 다음 대상 목록의 <code>public-two</code> 서브넷에서 NAT 게이트웨이를 선택합니다. 5. 서브넷 연결 탭에서 서브넷 연결 편집을 선택하고, <code>10.0.0.64/28</code> CIDR 범위가 있는 <code>private-two</code> 서브넷을 선택한 다음 연결 저장을 선택합니다. 6. 변경 사항 저장(Save Changes)을 선택합니다. 	AWS 관리자

Lambda 함수를 생성하고, VPC에 추가하며, 솔루션을 테스트합니다.

작업	설명	필요한 기술
새 Lambda 함수를 생성합니다.	<ol style="list-style-type: none"> 1. AWS Lambda 콘솔을 열고 함수 생성을 선택합니다. 	AWS 관리자

작업	설명	필요한 기술
	<ol style="list-style-type: none"> 기본 정보에서 함수 이름에 <code>Lambda test</code>를 입력한 다음 런타임에서 원하는 언어를 선택합니다. 함수 생성을 선택합니다. 	
Lambda 함수를 VPC에 추가합니다.	<ol style="list-style-type: none"> AWS Lambda 콘솔에서 함수를 선택한 다음 앞서 생성한 함수를 선택합니다. 구성을 선택한 다음 VPC를 선택합니다. 편집을 선택한 다음 Lambda VPC 및 두 개의 프라이빗 서브넷을 모두 선택합니다. 테스트용 기본 보안 그룹을 선택한 다음 저장을 선택합니다. 	AWS 관리자
외부 서비스를 직접적으로 호출하기 위한 코드를 작성합니다.	<ol style="list-style-type: none"> 선택한 프로그래밍 언어로 IP 주소를 반환하는 외부 서비스를 직접적으로 호출하는 코드를 작성하십시오. 반환된 IP 주소가 Elastic IP 주소 중 하나와 일치하는지 확인하십시오. 	AWS 관리자

관련 리소스

- [VPC에서 리소스에 액세스하도록 Lambda 함수 구성](#)

쿠버네티스를 사용하여 Amazon EKS 작업자 노드에 SSM 에이전트를 설치합니다. DaemonSet

작성자: Mahendra Siddappa(AWS)

환경: PoC 또는 파일럿

기술: 컨테이너 및 마이크로서비스, 인프라 DevOps

AWS 서비스: Amazon EKS, AWS Systems Manager

요약

참고, 2021년 9월: 최신 Amazon EKS 최적화 AMI는 SSM 에이전트를 자동으로 설치합니다. 자세한 내용은 2021년 6월 AMI [릴리스 정보](#)를 참조하세요..

Amazon Elastic Kubernetes Service(Amazon EKS)에서는 보안 지침으로 인해 워커 노드에 Secure Shell(SSH) 키 페어가 연결되어 있지 않습니다. 이 패턴은 수동으로 설치하거나 노드의 Amazon 머신 이미지 (AMI) 를 교체하는 대신 Kubernetes DaemonSet 리소스 유형을 사용하여 모든 작업자 노드에 AWS Systems Manager Agent (SSM Agent) 를 설치하는 방법을 보여줍니다. DaemonSet 작업자 노드의 크론 작업을 사용하여 SSM 에이전트 설치 일정을 잡습니다. 이 패턴을 사용하여 워커 노드에 다른 패키지를 설치할 수도 있습니다.

클러스터에서 문제를 해결할 때 필요에 따라 SSM 에이전트를 설치하면 SSH 키 페어 없이 워커 노드와 SSH 세션을 설정하거나, 로그를 수집하거나, 인스턴스 구성을 살펴볼 수 있습니다.

사전 조건 및 제한 사항

사전 조건

- Amazon Elastic Compute Cloud(Amazon EC2) 워커 노드가 있는 기존 Amazon EKS 클러스터입니다.
- 컨테이너 인스턴스는 SSM 서비스와 통신하는 데 필요한 권한을 가져야 합니다. AWS Identity 및 Access Management (IAM) 관리 역할 AmazonSSM은 EC2 인스턴스에서 SSM 에이전트를 실행하는 데 필요한 권한을 ManagedInstanceCore 제공합니다. 자세한 내용은 [AWS Systems Manager 설치 명서](#)를 참조하세요.

제한 사항

- 이 패턴은 Fargate 플랫폼에서 지원되지 DaemonSets 않으므로 AWS Fargate에는 적용할 수 없습니다.
- 이 패턴은 Linux 기반 워커 노드에만 적용됩니다.
- DaemonSet 포드는 권한 모드에서 실행됩니다. Amazon EKS 클러스터에 권한 모드에서 포드를 차단하는 웹후크가 있는 경우 SSM 에이전트는 설치되지 않습니다.

아키텍처

다음 사항은 이 패턴에 대한 아키텍처를 나타낸 다이어그램입니다.

도구

도구

- [kubecti](#)은 Amazon EKS 클러스터와 상호 작용하는 데 사용하는 명령줄 유틸리티입니다. 이 패턴은 Amazon EKS DaemonSet 클러스터에 a를 kubect1 배포하는 데 사용되며, 이 클러스터는 모든 작업자 노드에 SSM 에이전트를 설치합니다.
- [Amazon EKS](#)를 사용하면 자체 Kubernetes 컨트롤 플레인이나 노드를 설치, 운영 및 유지 관리할 필요 없이 AWS에서 Kubernetes를 쉽게 실행할 수 있습니다. Kubernetes는 컨테이너화된 애플리케이션의 배포, 조정 및 관리 자동화를 위한 오픈 소스 시스템입니다.
- [AWS Systems Manager Session Manager](#)를 사용하면 대화형, 원클릭, 브라우저 기반 셸 또는 AWS Command Line Interface(AWS CLI)를 통해 EC2 인스턴스, 온프레미스 인스턴스 및 가상 머신(VM)을 관리할 수 있습니다.

코드

다음 코드를 사용하여 Amazon EKS 클러스터에 SSM 에이전트를 설치할 DaemonSet 구성 파일을 생성합니다. [에픽](#) 섹션의 지침을 따르세요.

```
cat << EOF > ssm_daemonset.yaml
apiVersion: apps/v1
kind: DaemonSet
metadata:
  labels:
    k8s-app: ssm-installer
  name: ssm-installer
```

```
namespace: kube-system
spec:
  selector:
    matchLabels:
      k8s-app: ssm-installer
  template:
    metadata:
      labels:
        k8s-app: ssm-installer
    spec:
      containers:
        - name: sleeper
          image: busybox
          command: ['sh', '-c', 'echo I keep things running! && sleep 3600']
      initContainers:
        - image: amazonlinux
          imagePullPolicy: Always
          name: ssm
          command: ["/bin/bash"]
          args: ["-c", "echo '* * * * * root yum install -y https://s3.amazonaws.com/ec2-downloads-windows/SSMAgent/latest/linux_amd64/amazon-ssm-agent.rpm & rm -rf /etc/cron.d/ssmstart' > /etc/cron.d/ssmstart"]
          securityContext:
            allowPrivilegeEscalation: true
          volumeMounts:
            - mountPath: /etc/cron.d
              name: cronfile
          terminationMessagePath: /dev/termination-log
          terminationMessagePolicy: File
      volumes:
        - name: cronfile
          hostPath:
            path: /etc/cron.d
            type: Directory
      dnsPolicy: ClusterFirst
      restartPolicy: Always
      schedulerName: default-scheduler
      terminationGracePeriodSeconds: 30
EOF
```

에픽

kubectl 설정

작업	설명	필요한 기술
EKS 클러스터에 액세스하려면 kubectl을 설치하고 구성합니다.	Amazon EKS 클러스터에 액세스하도록 아직 kubectl을 설치 및 구성하지 않은 경우, Amazon EKS 설명서의 kubectl 설치 를 참조하세요.	DevOps

배포하십시오. DaemonSet

작업	설명	필요한 기술
DaemonSet 구성 파일을 생성합니다.	<p>이 패턴의 앞부분에 있는 코드 섹션의 코드를 사용하여 Amazon EKS 클러스터에 배포할 이라는 DaemonSet <code>ssm_daemonset.yaml</code> 구성 파일을 생성합니다.</p> <p>에서 시작한 DaemonSet 포드에는 기본 컨테이너와 컨테이너가 있습니다init. 기본 컨테이너에는 <code>sleep</code> 명령이 있습니다. <code>init</code> 컨테이너에는 <code>/etc/cron.d/</code> 경로에 SSM 에이전트를 설치하기 위한 <code>cron</code> 작업 파일을 생성하는 <code>command</code> 섹션이 포함되어 있습니다. 크론 작업은 한 번만 실행되며, 크론 작업이 생성된 파일은 작업이 완료된 후 자동으로 삭제됩니다.</p>	DevOps

작업	설명	필요한 기술
	<p>초기화 컨테이너가 완료되면 기본 컨테이너는 60분 동안 기다린 후 종료됩니다. 60분 후 새 포드가 시작됩니다. 이 포드는 SSM 에이전트가 없는 경우 SSM 에이전트를 설치하거나 SSM 에이전트를 최신 버전으로 업데이트합니다.</p> <p>필요한 경우 포드를 하루에 한 번 다시 시작하거나 더 자주 실행하도록 sleep 명령을 수정할 수 있습니다.</p>	
<p>Amazon EKS DaemonSet 클러스터에 배포합니다.</p>	<p>이전 단계에서 생성한 DaemonSet 구성 파일을 Amazon EKS 클러스터에 배포하려면 다음 명령을 사용합니다.</p> <pre data-bbox="597 1115 1026 1230">kubectl apply -f ssm_daemonset.yaml</pre> <p>이 명령은 워커 노드에서 포드를 DaemonSet 실행하여 SSM 에이전트를 설치하는 a를 생성합니다.</p>	<p>DevOps</p>

관련 리소스

- [kubectl 설치](#)(Amazon EKS 설명서)
- [세션 관리자 설정](#)(AWS Systems Manager 설명서)

다음을 사용하여 Amazon EKS 작업자 노드에 SSM CloudWatch 에이전트 및 에이전트를 설치합니다. preBootstrapCommands

작성자: Akkamahadevi Hiremath(AWS)

환경: 프로덕션

기술: 컨테이너 및 마이크로서비스, 인프라, 운영

AWS 서비스: 아마존 EKS, AWS Systems Manager, 아마존 CloudWatch

요약

이 패턴은 Amazon EKS 클러스터를 생성하는 동안 아마존 웹 서비스 (AWS) 클라우드의 아마존 CloudWatch Elastic Kubernetes Service (Amazon EKS) 워커 노드를 설치하기 위한 코드 샘플과 단계를 제공합니다. eksctl [구성](#) 파일 스키마의 preBootstrapCommands 속성을 사용하여 SSM 에이전트와 CloudWatch 에이전트를 설치할 수 있습니다 (Weaveworks 설명서). 그러면 Amazon Elastic Compute Cloud(Amazon EC2) 키 쌍을 사용하지 않고도 SSM 에이전트를 사용하여 워커 노드에 연결할 수 있습니다. 또한 CloudWatch 에이전트를 사용하여 Amazon EKS 작업자 노드의 메모리 및 디스크 사용률을 모니터링할 수 있습니다.

사전 조건 및 제한 사항

사전 조건

- 활성 상태의 AWS 계정
- macOS, Linux 또는 Windows에 설치 및 구성된 [eksctl 명령줄 유틸리티](#)
- macOS, Linux 또는 Windows에 설치 및 구성된 [kubectl 명령줄 유틸리티](#)

제한 사항

- 장기 실행 스크립트를 preBootstrapCommands 속성에 추가하지 않는 것이 좋습니다. 이렇게 하면 스케일링 작업 중에 노드가 Amazon EKS 클러스터에 가입하는 것이 지연되기 때문입니다. 대신 [사용자 지정 Amazon Machine Image\(AMI\)](#)를 생성하는 것이 좋습니다.
- 이 패턴은 Amazon EC2 Linux 인스턴스에만 적용됩니다.

아키텍처

기술 스택

- 아마존 CloudWatch
- Amazon Elastic Kubernetes Service(Amazon EKS)
- AWS Systems Manager Parameter Store

대상 아키텍처

다음 다이어그램은 preBootstrapCommands를 사용하여 설치한 SSM 에이전트를 사용함으로써 Amazon EKS 워커 노드에 연결하는 사용자의 예를 보여줍니다.

이 다이어그램은 다음 워크플로를 보여줍니다.

1. 사용자는 preBootstrapCommands 속성이 포함된 eksctl 구성 파일을 사용하여 Amazon EKS 클러스터를 생성합니다. 그러면 SSM 에이전트와 에이전트가 설치됩니다. CloudWatch
2. 조정 활동으로 인해 나중에 클러스터에 합류하는 모든 새 인스턴스는 사전 설치된 SSM 에이전트 및 에이전트를 사용하여 생성됩니다. CloudWatch
3. 사용자는 SSM 에이전트를 사용하여 Amazon EC2에 연결한 다음 에이전트를 사용하여 메모리 및 디스크 사용률을 모니터링합니다. CloudWatch

도구

- [Amazon](#)은 AWS 리소스의 지표와 AWS에서 실행하는 애플리케이션을 실시간으로 모니터링할 수 있도록 CloudWatch 도와줍니다.
- [Amazon Elastic Kubernetes Service\(Amazon EKS\)](#)는 자체 Kubernetes 컨트롤 플레인 또는 노드를 설치하거나 유지 관리할 필요 없이 AWS의 Kubernetes를 실행하는 데 도움이 됩니다.
- [AWS Systems Manager Parameter Store](#)는 구성 데이터 관리 및 암호 관리를 위한 안전한 계층적 스토리지를 제공합니다.
- [AWS Systems Manager Session Manager](#)를 사용하면 대화형, 원클릭, 브라우저 기반 셸 또는 AWS Command Line Interface(AWS CLI)를 통해 EC2 인스턴스, 온프레미스 인스턴스 및 가상 머신을 관리할 수 있습니다.

- [eksctl](#)은 Amazon EKS에서 Kubernetes 클러스터를 생성하고 관리하기 위한 명령줄 유틸리티입니다.
- [kubectl](#)는 클러스터 API 서버와 통신하기 위gks 명령줄 유틸리티입니다.

에픽

Amazon EKS 클러스터 생성

작업	설명	필요한 기술
<p>CloudWatch 에이전트 구성 파일을 저장합니다.</p>	<p>Amazon EKS 클러스터를 생성하려는 AWS 지역의 AWS Systems Manager Parameter Store에 CloudWatch 에이전트 구성 파일을 저장합니다. 이렇게 하려면 AWS Systems Manager Parameter Store에서 파라미터를 생성하고 파라미터의 이름(예:AmazonCloudwatch-linux)을 기록해 두세요.</p> <p>자세한 내용은 이 패턴의 추가 정보 섹션에 있는 예제 CloudWatch 에이전트 구성 파일 코드를 참조하십시오.</p>	<p>DevOps 엔지니어</p>
<p>eksctl 구성 파일 및 클러스터를 생성합니다.</p>	<ol style="list-style-type: none"> 1. CloudWatch 에이전트 및 SSM 에이전트 설치 단계가 포함된 eksctl 구성 파일을 생성합니다. 자세한 내용은 이 패턴의 추가 정보 섹션에 있는 예제 eksctl 구성 파일 코드를 참조하세요. 2. <code>eksctl create cluster -f cluster.y</code> 	<p>AWS DevOps</p>

작업	설명	필요한 기술
	am1 명령을 실행하여 클러스터를 생성합니다.	

SSM 에이전트와 CloudWatch 에이전트가 작동하는지 확인

작업	설명	필요한 기술
SSM 에이전트를 테스트합니다.	AWS Systems Manager 설명서에서 세션 시작 에 설명된 방법 중 하나를 사용하여 SSH를 통해 Amazon EKS 클러스터 노드에 연결합니다.	AWS DevOps
CloudWatch 에이전트를 테스트해 보세요.	CloudWatch 콘솔을 사용하여 CloudWatch 에이전트를 검증하세요. <ol style="list-style-type: none"> 1. AWS Management Console에 로그인하고 CloudWatch 콘솔을 엽니다. 2. 탐색 창에서 지표를 확장한 다음 모든 지표를 선택합니다. 3. 찾아보기 탭의 검색 상자에 메모리 및 디스크 지표를 보려면 CWAgent 지표를 입력한 다음 선택하세요. 	AWS DevOps

관련 리소스

- [서버에 CloudWatch 에이전트 설치 및 실행](#) (Amazon CloudWatch 설명서)
- [Systems Manager 파라미터 생성\(콘솔\)](#)(AWS Systems Manager 설명서)
- [CloudWatch 에이전트 구성 파일 생성](#) (Amazon CloudWatch 설명서)

- [세션 시작\(AWS CLI\)](#)(AWS Systems Manager 설명서)
- [세션 시작 Amazon EC2 콘솔](#)(AWS Systems Manager 설명서)

추가 정보

예제 CloudWatch 에이전트 구성 파일

다음 예제에서 CloudWatch 에이전트는 Amazon Linux 인스턴스의 디스크 및 메모리 사용률을 모니터링하도록 구성되어 있습니다.

```
{
  "agent": {
    "metrics_collection_interval": 60,
    "run_as_user": "cwagent"
  },
  "metrics": {
    "append_dimensions": {
      "AutoScalingGroupName": "${aws:AutoScalingGroupName}",
      "ImageId": "${aws:ImageId}",
      "InstanceId": "${aws:InstanceId}",
      "InstanceType": "${aws:InstanceType}"
    },
    "metrics_collected": {
      "disk": {
        "measurement": [
          "used_percent"
        ],
        "metrics_collection_interval": 60,
        "resources": [
          "*"
        ]
      },
      "mem": {
        "measurement": [
          "mem_used_percent"
        ],
        "metrics_collection_interval": 60
      }
    }
  }
}
```

예제 eksctl 구성 파일

```

apiVersion: eksctl.io/v1alpha5
kind: ClusterConfig
metadata:
  name: test
  region: us-east-2
  version: "1.24"
managedNodeGroups:
  - name: test
    minSize: 2
    maxSize: 4
    desiredCapacity: 2
    volumeSize: 20
    instanceType: t3.medium
    preBootstrapCommands:
      - sudo yum install amazon-ssm-agent -y
      - sudo systemctl enable amazon-ssm-agent
      - sudo systemctl start amazon-ssm-agent
      - sudo yum install amazon-cloudwatch-agent -y
      - sudo /opt/aws/amazon-cloudwatch-agent/bin/amazon-cloudwatch-agent-ctl -a fetch-
config -m ec2 -s -c ssm:AmazonCloudwatch-linux
    iam:
      attachPolicyARNs:
        - arn:aws:iam::aws:policy/AmazonEKSEWorkerNodePolicy
        - arn:aws:iam::aws:policy/AmazonEKS_CNI_Policy
        - arn:aws:iam::aws:policy/AmazonEC2ContainerRegistryReadOnly
        - arn:aws:iam::aws:policy/CloudWatchAgentServerPolicy
        - arn:aws:iam::aws:policy/AmazonSSMManagedInstanceCore

```

추가 코드 세부 정보

- `preBootstrapCommands` 속성의 마지막 줄에서 `AmazonCloudwatch-linux`는 AWS Systems Manager Parameter Store에서 생성된 파라미터의 이름입니다. Amazon EKS 클러스터를 생성한 동일한 AWS 리전의 파라미터 스토어에 `AmazonCloudwatch-linux`를 포함해야 합니다. 파일 경로를 지정할 수도 있지만 더 쉽게 자동화하고 재사용할 수 있도록 Systems Manager를 사용하는 것이 좋습니다.
- `eksctl` 구성 파일에서 `preBootstrapCommands`를 사용하는 경우 AWS Management Console에 두 개의 시작 템플릿이 표시됩니다. 첫 번째 시작 템플릿에는 `preBootstrapCommands`에 지정된 명령이 포함됩니다. 두 번째 템플릿에는 `preBootstrapCommands`에 지정된 명령과 기본 Amazon

EKS 사용자 데이터가 포함됩니다. 이 데이터는 노드가 클러스터에 가입하도록 하는 데 필요합니다. 노드 그룹의 Auto Scaling 그룹은 이 사용자 데이터를 사용하여 새 인스턴스를 스핀업합니다.

- eksctl 구성 파일의 iam 속성을 사용하는 경우 첨부된 AWS Identity 및 Access Management (IAM) 정책에 필요한 추가 정책과 함께 기본 Amazon EKS 정책을 나열해야 합니다. eksctl 구성 파일 및 클러스터 생성 단계의 코드 AmazonSSMManagedInstanceCore 스니펫에는 에이전트와 SSM CloudWatch 에이전트가 예상대로 작동하도록 하기 위한 추가 정책이 추가되었습니다. CloudWatchAgentServerPolicy AmazonEKSWorkerNodePolicy, AmazonEKS_CNI_Policy, AmazonEC2ContainerRegistryReadOnly 정책은 Amazon EKS 클러스터가 올바르게 작동하는 데 필요한 필수 정책입니다.

AWS App2Container 생성 도커 이미지 최적화

작성자: Varun Sharma(AWS)

환경: PoC 또는 파일럿

기술: 컨테이너 및 마이크로서
비스, 현대화, DevOps

AWS 서비스: Amazon ECS

요약

AWS App2Container는 코드를 변경할 필요 없이 온프레미스 또는 가상 머신에서 실행 중인 기존 애플리케이션을 컨테이너로 변환하는 데 도움이 되는 명령줄 도구입니다.

애플리케이션 유형에 따라 App2Container는 보수적인 접근 방식을 사용하여 종속성을 식별합니다. 프로세스 모드에서는 애플리케이션 서버의 모든 비시스템 파일이 컨테이너 이미지에 포함됩니다. 이 경우 상당히 큰 이미지가 생성될 수 있습니다.

이 패턴은 App2Container에서 생성된 컨테이너 이미지를 최적화하는 방법을 제공합니다. 이는 프로세스 모드에서 App2Container가 발견한 모든 Java 애플리케이션에 적용할 수 있습니다. 패턴에 정의된 워크플로는 애플리케이션 서버에서 실행되도록 설계되었습니다.

사전 조건 및 제한 사항

사전 조건

- 활성 상태의 AWS 계정
- Linux 서버에서 애플리케이션 서버를 실행하는 Java 애플리케이션
- Linux 서버에 [App2Container 설치 및 설정](#), 모든 사전 조건 충족

아키텍처

소스 기술 스택

- Linux 서버에서 실행되는 Java 애플리케이션

대상 기술 스택

- App2Container 생성 도커 이미지

대상 아키텍처 흐름

1. 애플리케이션 서버에서 실행 중인 애플리케이션을 찾고 애플리케이션을 분석합니다.
2. 애플리케이션을 컨테이너화합니다.
3. 도커 이미지의 크기를 평가합니다. 이미지가 너무 크면 4단계를 진행합니다.
4. 셸 스크립트(첨부됨)를 사용하여 대용량 파일을 식별하십시오.
5. analysis.json 파일의 appExcludedFiles 및 appSpecificFiles 목록을 업데이트하십시오.

도구

도구

- [AWS App2Container](#) - AWS App2Container(A2C)는 온프레미스 데이터 센터 또는 가상 머신에서 실행되는 애플리케이션을 Amazon Elastic Container Service(Amazon ECS) 또는 Amazon Elastic Kubernetes Service(Amazon EKS)에서 관리하는 컨테이너에서 실행되도록 리프트 앤드 시프트하는 데 도움이 되는 명령줄 도구입니다.

code

optimizeImage.sh 셸 스크립트와 예제 analysis.json 파일이 첨부되어 있습니다.

이 optimizeImage.sh 파일은 App2Container 생성 파일 ContainerFiles.tar의 내용을 검토하기 위한 유틸리티 스크립트입니다. 검토에서는 크기가 커서 제외할 수 있는 파일 또는 하위 디렉터리를 식별합니다. 이 스크립트는 다음 tar 명령의 래퍼입니다.

```
tar -Ptvf <path>|tr -s ' '|cut -d ' ' -f3,6| awk '$2 ~/<filetype>$/'| awk '$2 ~/^<toplevel>/'| cut -f1-<depth> -d '/'|awk '{ if ($1>= <size>) arr[$2]+=$1 } END { for (key in arr) { if(<verbose>) printf("%-50s\t%-50s\n", key, arr[key]) else printf("%s, \n", key) } } '|sort -k2 -nr
```

tar 명령에서 스크립트는 다음 값을 사용합니다.

path	ContainerFiles.tar 의 경로
filetype	일치시킬 파일 유형

<code>toplevel</code>	일치시킬 최상위 디렉터리
<code>depth</code>	절대 경로의 깊이
<code>size</code>	각 파일의 크기

스크립트는 다음 작업을 수행합니다.

1. 파일을 추출하지 않고 나열하는 데 `tar -Ptvf`를 사용합니다.
2. 최상위 디렉터리부터 시작하여 파일 유형별로 파일을 필터링합니다.
3. 깊이에 따라 절대 경로를 색인으로 생성합니다.
4. 색인 및 저장소를 기준으로 하위 디렉터리의 전체 크기를 제공합니다.
5. 하위 디렉터리의 크기를 인쇄합니다.

`tar` 명령에서 값을 수동으로 바꿀 수도 있습니다.

에픽

애플리케이션 검색, 분석 및 컨테이너화

작업	설명	필요한 기술
온프레미스 Java 애플리케이션을 찾아봅니다.	애플리케이션 서버에서 실행 중인 모든 애플리케이션을 검색하려면 다음 명령을 실행합니다. <pre>sudo app2container inventory</pre>	AWS DevOps
검색된 애플리케이션을 분석합니다.	인벤토리 단계에서 얻은 <code>application-id</code> 를 사용하여 각 애플리케이션을 원격으로 분석하려면, 다음 명령을 실행합니다.	AWS DevOps

작업	설명	필요한 기술
<p>분석된 애플리케이션을 컨테이너화합니다.</p>	<pre data-bbox="597 212 1027 369">sudo app2container analyze --application- id <java-app-id></pre> <p data-bbox="597 405 1027 489">애플리케이션을 컨테이너화하려면 다음 명령을 실행합니다.</p> <pre data-bbox="597 527 1027 726">sudo app2container containerize --applica tion-id <application- id></pre> <p data-bbox="597 762 1027 888">이 명령은 작업 공간 위치에 tar 번들과 함께 도커 이미지를 생성합니다.</p> <p data-bbox="597 940 1027 1024">도커 이미지가 너무 크면 다음 단계로 진행합니다.</p>	<p>AWS DevOps</p>

appExcludedFiles appSpecificFiles App2Container에서 추출한 tar 파일을 식별하고 해당 파일에서 추출합니다.

작업	설명	필요한 기술
<p>아티팩트 tar 파일 크기를 식별하십시오.</p>	<p data-bbox="597 1367 1027 1734">{workspace}/{java-app-id}/Artifacts 에서 ContainerFiles.tar 파일을 식별하며, 여기서 workspace 는 App2Container 작업 영역이고 java-app-id 는 애플리케이션 ID 입니다.</p> <pre data-bbox="597 1772 1027 1866">./optimizeImage.sh -p / {workspace}/{java-app-</pre>	<p>AWS DevOps</p>

작업	설명	필요한 기술
	<pre data-bbox="594 212 1027 344">id}/Artifacts/ContainerFiles.tar -d 0 -t / -v</pre> <p data-bbox="594 384 1027 464">이것은 최적화 후 tar 파일의 총 크기입니다.</p>	
<p data-bbox="115 510 548 590">/ 디렉터리 아래의 하위 디렉터리와 해당 크기를 나열합니다.</p>	<p data-bbox="594 510 1027 688">/ 최상위 디렉터리에 있는 주요 하위 디렉터리의 크기를 확인하려면 다음 명령을 실행합니다.</p> <pre data-bbox="594 747 1027 1497">./optimizeImage.sh -p /{workspace}/{java-app-id}/Artifacts/ContainerFiles.tar -d 1 -t / -s 1000000 -v /var 554144711 /usr 2097300819 /tmp 18579660 /root 43645397 /opt 222320534 /home 65212518 /etc 11357677</pre>	<p data-bbox="1068 510 1268 548">AWS DevOps</p>

작업	설명	필요한 기술
<p>/ 디렉터리 아래에 있는 대규모 하위 디렉터리를 식별하십시오.</p>	<p>이전 명령에 나열된 각 주요 하위 디렉터리에 대해 해당 하위 디렉터리의 크기를 식별하십시오. -d를 사용하여 깊이를 높이고 -t를 사용하여 최상위 디렉터리를 나타냅니다.</p> <p>예를 들어, /var를 최상위 디렉터리로 사용하십시오. /var 아래에서 모든 대형 하위 디렉터리와 그 크기를 식별하십시오.</p> <pre data-bbox="592 808 1031 1050">./optimizeImage.sh -p / {workspace}/{java-app- id}/Artifacts/Containe rFiles.tar -d 2 -t / var -s 1000000 -v</pre> <p>이전 단계에 나열된 각 하위 디렉터리(예를 들어 /usr, /tmp, /opt 및 /home)에 대해 이 프로세스를 반복합니다.</p>	<p>AWS DevOps</p>

작업	설명	필요한 기술
<p>/ 디렉터리 아래의 각 하위 디렉터리에 있는 대형 폴더를 분석합니다.</p>	<p>이전 단계에 나열된 각 하위 디렉터리에 대해 애플리케이션을 실행하는 데 필요한 폴더를 식별하십시오.</p> <p>예를 들어, 이전 단계의 하위 디렉터리를 사용하여 /var 디렉터리의 모든 하위 디렉터리와 해당 크기를 나열합니다. 애플리케이션에 필요한 모든 하위 디렉터리를 식별하십시오.</p> <pre data-bbox="594 758 1027 1039"> /var/tmp 237285851 /var/lib 24489984 /var/cache 237285851 </pre> <p>애플리케이션에 필요하지 않은 하위 디렉터리를 제외하려면 <code>analysis.json</code> 파일에서 해당 하위 디렉터리를 <code>containerParameters</code> 아래 <code>appExcludedFiles</code> 섹션에 추가합니다.</p> <p>예제 <code>analysis.json</code> 파일이 첨부되어 있습니다.</p>	<p>AWS DevOps</p>

작업	설명	필요한 기술
appExcludes 목록에서 필요한 파일을 식별하십시오..	<p>appExcludes 목록에 추가된 각 하위 디렉터리에 대해 애플리케이션에 필요한 해당 하위 디렉터리의 모든 파일을 식별하십시오. analysis.json 파일에서, containerParameters 아래 appSpecificFiles 섹션의 특정 파일 또는 하위 디렉터리를 추가합니다.</p> <p>예를 들어 /usr/lib 디렉터리가 제외 목록에 추가되었지만 /usr/lib/jvm 이 애플리케이션에 필요한 경우 /usr/lib/jvm 을 appSpecificFiles 섹션에 추가하십시오.</p>	AWS DevOps

애플리케이션을 다시 추출하고 컨테이너화합니다.

작업	설명	필요한 기술
분석된 애플리케이션을 컨테이너화합니다.	<p>애플리케이션을 컨테이너화하려면 다음 명령을 실행합니다.</p> <pre>sudo app2container containerize --application-id <application-id></pre> <p>이 명령은 작업 공간 위치에 tar 번들과 함께 도커 이미지를 생성합니다.</p>	AWS DevOps

작업	설명	필요한 기술
<p>아티팩트 tar 파일 크기를 식별하십시오.</p>	<p>{workspace}/{java-app-id}/Artifacts 에서 ContainerFiles.tar 파일을 식별하며, 여기서 workspace 는 App2Container 작업 영역이고 java-app-id 는 애플리케이션 ID 입니다.</p> <pre data-bbox="594 636 1024 873"> ./optimizeImage.sh -p / {workspace}/{java-app-id}/Artifacts/ContainerFiles.tar -d 0 -t / -v </pre> <p>이것은 최적화 후 tar 파일의 총 크기입니다.</p>	<p>AWS DevOps</p>
<p>도커 이미지를 실행합니다.</p>	<p>이미지가 오류 없이 시작되는지 확인하려면 다음 명령을 사용하여 로컬에서도커 이미지를 실행합니다.</p> <p>컨테이너의 imageId를 식별하려면 docker images grep java-app-id 를 사용하십시오.</p> <p>컨테이너를 실행하려면 docker run -d <image id>를 사용하십시오.</p>	<p>AWS DevOps</p>

관련 리소스

- [App2Container란 무엇입니까?](#)

- [AWS App2Container - Java 및 .NET 애플리케이션을 위한 새로운 컨테이너화 도구](#)(블로그 게시물)

첨부

이 문서와 관련된 추가 콘텐츠에 액세스하려면 [attachment.zip](#) 파일의 압축을 풉니다.

노드 어피니티, 테인트 및 톨러레이션을 사용하여 Amazon EKS에 Kubernetes 포드 배치

작성: 히테시 패릭(AWS), 라구 바미디마리(AWS)

환경: PoC 또는 파일럿

기술: 컨테이너 및 마이크로서
비스

워크로드: 오픈 소스

AWS 서비스: Amazon EKS

요약

이 패턴은 Kubernetes 노드 어피니티, 노드 테인트 및 포드 톨러레이션을 사용하여 Amazon Web Services(AWS) 클라우드의 Amazon Elastic Kubernetes Service(Amazon EKS) 클러스터에 있는 특정 워커 노드에 애플리케이션 포드를 의도적으로 예약하는 것을 보여줍니다.

테인트는 노드가 포드 세트를 거부할 수 있도록 하는 노드 속성입니다. 톨러레이션은 Kubernetes 스케줄러가 일치하는 테인트가 있는 노드에서 포드를 예약할 수 있게 해주는 포드 속성입니다.

하지만 톨러레이션만으로는 스케줄러가 테인트가 없는 워커 노드에 포드를 배치하는 것을 방지할 수 없습니다. 예를 들어, 톨러레이션이 있는 컴퓨팅 집약적 포드는 의도치 않게 테인트가 없는 범용 노드에 예약될 수 있습니다. 이 경우, 포드의 노드 어피니티 속성은 스케줄러에게 노드 어피니티에 지정된 노드 선택 기준을 충족하는 노드에 포드를 배치하도록 지시합니다.

테인트, 톨러레이션, 노드 어피니티를 함께 사용하면 스케줄러가 포드에 지정된 노드 어피니티 노드 선택 기준과 일치하는 테인트가 있는 노드와 노드 레이블에 일관되게 포드를 예약하도록 지시합니다.

이 패턴은 Kubernetes 배포 매니페스트 파일 예제와 EKS 클러스터를 생성하고 애플리케이션을 배포하고 포드 배치를 검증하는 단계를 제공합니다.

사전 조건 및 제한 사항

사전 조건

- AWS 계정에서 리소스를 생성하도록 구성된 보안 인증 정보가 있는 AWS 계정
- AWS Command Line Interface(AWS CLI)
- eksctl

- kubectl
- [Docker](#)가 설치되고(사용 중인 운영 체제용), 엔진이 시작됨(Docker 라이선스 요구 사항에 대한 자세한 내용은 [Docker 사이트](#) 참조)
- [Java](#) 버전 11 이상
- 선호하는 통합 개발 환경(IDE)에서 실행되는 Java 마이크로서비스(예: [AWS Cloud9](#), [IntelliJ IDEA Community Edition](#) 또는 [Eclipse](#))(Java 마이크로서비스가 없는 경우, 마이크로 서비스를 생성하는데 도움이 필요하다면 [Amazon EKS에 샘플 Java 마이크로서비스 배포](#) 패턴 및 [Spring을 사용한 마이크로서비스](#) 참조)

제한 사항

- 이 패턴은 Java 코드를 제공하지 않으며 사용자가 이미 Java에 익숙하다고 가정합니다. 기본 Java 마이크로서비스를 생성하려면 [Amazon EKS에 샘플 Java 마이크로서비스 배포](#)를 참조하세요.
- 이 문서의 단계는 비용이 발생할 수 있는 AWS 리소스를 생성합니다. 패턴을 구현하고 검증하기 위한 단계를 완료한 후에는 AWS 리소스를 정리해야 합니다.

아키텍처

대상 기술 스택

- Amazon EKS
- Java
- Docker
- Amazon Elastic Container Registry (Amazon ECR)

대상 아키텍처

솔루션 아키텍처 다이어그램은 두 개의 포드(배포 1 및 배포 2)와 각각 두 개의 노드가 있는 두 개의 노드 그룹(ng1 및 ng2)이 있는 Amazon EKS를 보여줍니다. 포드 및 노드에는 다음과 같은 속성이 있습니다.

	배포 1 포드	배포 2 포드	노드 그룹 1(ng1)	노드 그룹 2 (ng2)
톨러레이션	키: 분류_워크로드, 값: true, 효과: NoSchedule	None		

	키: 머신러닝_워크로드, 값: true, 효과: NoSchedule		
노드 어피니티	키: alpha.eksctl.io/nodegroup-name = ng1;	None	nodeGroups.name = ng1
테인트		키: 분류_워크로드, 값: true, 효과: NoSchedule	None
		키: 머신러닝_워크로드, 값: true, 효과: NoSchedule	

1. 배포 1 포드에는 톨러레이션과 노드 어피니티가 정의되어 있으며, 이는 Kubernetes 스케줄러가 배포 포드를 노드 그룹 1(ng1) 노드에 배치하도록 지시합니다.
2. 노드 그룹 2 (ng2)에는 배포 1의 노드 어피니티 노드 선택기 표현식과 일치하는 노드 레이블이 없으므로, 포드는 ng2 노드에 스케줄링되지 않습니다.
3. 배포 2 포드에는 배포 매니페스트에 정의된 톨러레이션이나 노드 어피니티가 없습니다. 스케줄러는 노드의 테인트 때문에 노드 그룹 1에 배포 2 포드를 스케줄링하는 것을 거부합니다.
4. 노드에 테인트가 없기 때문에 배포 2 포드는 대신 노드 그룹 2에 배치됩니다.

이 패턴은 테인트와 톨러레이션을 노드 어피니티와 함께 사용하면 특정 워커 노드 세트에 포드를 배치하는 것을 제어할 수 있음을 보여줍니다.

도구

서비스

- [AWS Command Line Interface\(AWS CLI\)](#)는 명령줄 쉘에서 명령을 사용하여 AWS 서비스와 상호 작용할 수 있는 오픈 소스 도구입니다.

- [Amazon Elastic Container Registry\(Amazon ECR\)](#)는 안전하고 확장 가능하고 신뢰할 수 있는 관리형 컨테이너 이미지 레지스트리 서비스입니다.
- [Amazon Elastic Kubernetes Service \(Amazon EKS\)](#)는 자체 Kubernetes 컨트롤 플레인이나 노드를 설치하거나 유지 관리할 필요 없이 AWS에서 Kubernetes를 실행할 수 있도록 도와줍니다.
- [eksctl](#)은 kubectl과 동일한 AWS이며 EKS를 생성하는 데 도움이 됩니다.

기타 도구

- [Docker](#)는 운영 체제 수준의 가상화를 사용하여 컨테이너에 소프트웨어를 제공하는 서비스형 플랫폼 (PaaS) 제품 세트입니다.
- [kubectl](#)는 Kubernetes 클러스터에 대해 명령의 실행을 돕는 명령줄 인터페이스입니다.

에픽

클러스터 생성

작업	설명	필요한 기술
cluster.yaml 파일을 생성합니다.	<p>다음 코드를 이용해 cluster.yaml (이)라는 파일을 생성합니다.</p> <pre> apiVersion: eksctl.io/v1alpha5 kind: ClusterConfig metadata: name: eks-taint-demo region: us-west-1 # Unmanaged nodegroups with and without taints. nodeGroups: - name: ng1 instanceType: m5.xlarge minSize: 2 maxSize: 3 </pre>	<p>앱 소유자, AWS DevOps, 클라우드 관리자, DevOps 엔지니어</p>

작업	설명	필요한 기술
	<pre> taints: - key: classified_workload value: "true" effect: NoSchedule - key: machine_learning_workload value: "true" effect: NoSchedule - name: ng2 instanceType: m5.xlarge minSize: 2 maxSize: 3 </pre>	
<p>eksctl을 사용하여 클러스터를 생성합니다.</p>	<p>cluster.yaml 파일을 실행하여 EKS 클러스터를 생성합니다. 클러스터를 생성하는 데 몇 분 정도 걸릴 수 있습니다.</p> <pre> eksctl create cluster -f cluster.yaml </pre>	<p>AWS DevOps, AWS 시스템 관리자, 앱 개발자</p>

이미지를 생성하여 Amazon ECR에 업로드

작업	설명	필요한 기술
<p>Amazon ECR 프라이빗 리포지토리를 생성합니다.</p>	<p>Amazon ECR 리포지토리를 생성하려면 프라이빗 리포지토리 생성을 참조하세요. 리포지토리의 URI를 기록합니다.</p>	<p>AWS DevOps, DevOps 엔지니어, 앱 개발자</p>
<p>Dockerfile을 생성합니다.</p>	<p>패턴을 테스트하는 데 사용할 기존 Docker 컨테이너 이미지</p>	<p>AWS DevOps, DevOps 엔지니어</p>

작업	설명	필요한 기술
	<p>가 있는 경우 이 단계를 건너뛸 수 있습니다.</p> <p>Dockerfile을 생성하려면 다음 스니펫을 참조로 사용합니다. 오류가 발생하는 경우 문제 해결 섹션을 참조하세요.</p> <pre>FROM adoptopenjdk/openjdk11:jdk-11.0.14.1_1-alpine RUN apk add maven WORKDIR /code # Prepare by downloading dependencies ADD pom.xml /code/pom.xml RUN ["mvn", "dependency:resolve"] RUN ["mvn", "verify"] # Adding source, compile and package into a fat jar ADD src /code/src RUN ["mvn", "package"] EXPOSE 4567 CMD ["java", "-jar", "target/eksExample-jar-with-dependencies.jar"]</pre>	

작업	설명	필요한 기술
pom.xml 및 소스 파일을 생성하고 도커 이미지를 빌드하고 푸시합니다.	<p>pom.xml 파일 및 Java 소스 파일을 생성하려면 Amazon EKS에 샘플 Java 마이크로서비스 배포 패턴을 참조하세요.</p> <p>해당 패턴의 지침을 사용하여 도커 이미지를 빌드하고 푸시합니다.</p>	AWS DevOps, DevOps 엔지니어, 앱 개발자

Amazon EKS에 배포

작업	설명	필요한 기술
deployment.yaml 파일을 생성합니다.	<p>deployment.yaml 파일을 생성하려면 추가 정보 섹션의 코드를 사용합니다.</p> <p>코드에서 노드 어피니티의 키는 노드 그룹을 생성할 때 생성하는 모든 레이블입니다. 이 패턴은 eksctl이 생성한 기본 레이블을 사용합니다. 레이블 사용자 지정에 대한 자세한 내용은 Kubernetes 설명서의 노드에 포드 할당을 참조하세요.</p> <p>노드 어피니티 키의 값은 cluster.yaml 에서 생성한 노드 그룹의 이름입니다.</p> <p>테인트의 키와 값을 가져오려면 다음 명령을 실행합니다.</p>	AWS DevOps, DevOps 엔지니어, 앱 개발자

작업	설명	필요한 기술
	<pre>kubectl get nodes -o json jq '.items[] .spec.taints'</pre> <p>이미지는 이전 단계에서 생성한 Amazon ECR 리포지토리의 URI 입니다.</p>	
<p>파일을 배포합니다.</p>	<p>Amazon EKS에 배포하려면 다음 명령을 실행합니다.</p> <pre>kubectl apply -f deployment.yaml</pre>	<p>앱 개발자, DevOps 엔지니어, AWS DevOps</p>

작업	설명	필요한 기술
<p>배포를 확인합니다.</p>	<p>1. 포드가 READY 상태인지 확인하려면 다음 명령을 실행합니다.</p> <pre data-bbox="630 394 1029 512">kubect1 get pods -o wide</pre> <p>포드가 준비되면 출력은 다음처럼 표시되며, STATUS이 (가) 실행 중으로 표시됩니다.</p> <pre data-bbox="630 768 1029 1323"> NAME READY STATUS RESTARTS AGE IP NODE NOMINATED NODE READINESS GATES <pod_name> 1/1 Running 0 12d 192.168.1 8.50 ip-192-16 8-20-110.us-west-1 .compute.internal <none> <none> </pre> <p>포드 이름과 노드 이름을 기록합니다. 다음 단계를 건너뛸 수 있습니다.</p> <p>2. (선택 사항) 포드에 대한 추가 세부 정보를 확인하고 포드의 톨러레이션을 확인하려면 다음 명령을 실행합니다.</p>	<p>앱 개발자, DevOps 엔지니어, AWS DevOps</p>

작업	설명	필요한 기술
	<pre data-bbox="630 212 1029 327">kubect1 describe pod <pod_name></pre> <p data-bbox="630 363 1019 447">출력의 예는 추가 정보 섹션에 있습니다.</p> <p data-bbox="591 470 1008 600">3. 노드의 포드 배치가 올바른지 확인하려면 다음 명령을 실행합니다.</p> <pre data-bbox="630 636 1029 795">kubect1 describe node <node name> grep -A 1 "Taints"</pre> <p data-bbox="630 831 1019 1108">노드의 테인트가 톨러레이션과 일치하고 노드의 레이블이 deployment.yaml 에 정의된 노드 어피니티와 일치하는지 확인합니다.</p> <p data-bbox="630 1152 1027 1472">톨러레이션과 노드 어피니티가 있는 포드는 테인트와 노드 어피니티 레이블이 일치하는 노드에 배치해야 합니다. 이전 명령은 노드의 테인트를 제공합니다. 다음은 예시 출력입니다.</p> <pre data-bbox="630 1514 1029 1850">kubect1 describe node ip-192-168-29-181. us-west-1.compute. internal grep -A 1 "Taints" Taints: classified_workload=true:NoSchedule</pre>	

작업	설명	필요한 기술
	<pre>machine_learning_w orkload=true:NoSch edule</pre> <p>또한 다음 명령을 실행하여 포드가 배치된 노드에 노드 어피니티 노드 레이블과 일치하는 레이블이 있는지 확인합니다.</p> <pre>kubect1 get node <node name> --show-labels</pre> <p>4. 애플리케이션이 의도한 대로 작동하는지 확인하려면 다음 명령을 실행하여 포드 로그를 확인합니다.</p> <pre>kubect1 logs -f <name- of-the-pod></pre>	

작업	설명	필요한 기술
<p>톨러레이션 및 노드 어피니티 없이 두 번째 배포 .yaml 파일을 생성합니다.</p>	<p>이 추가 단계는 배포 매니페스트 파일에 노드 어피니티 또는 톨러레이션이 지정되지 않은 경우 결과 포드가 테인트가 있는 노드에 스케줄링되지 않는지 확인하기 위한 것입니다. (테인트가 없는 노드에 스케줄링해야 합니다.) 다음 코드를 사용하여 <code>deploy_no_taint.yaml</code> (이)라는 새 배포 파일을 생성합니다.</p> <pre data-bbox="597 779 1024 1822"> apiVersion: apps/v1 kind: Deployment metadata: name: microservice-deployment-non-tainted spec: replicas: 1 selector: matchLabels: app.kubernetes.io/name: java-microservice-no-taint template: metadata: labels: app.kubernetes.io/name: java-microservice-no-taint spec: containers: - name: java-microservice-container-2 image: <account_number>.dkr.ecr<region>.amazonaws.com/... </pre>	<p>앱 개발자, AWS DevOps, DevOps 엔지니어</p>

작업	설명	필요한 기술
	<pre>zonaws.com/<repository_name>:latest ports: - container Port: 4567</pre>	
<p>두 번째 배포 .yaml 파일 배포 및 포드 배치 검증</p>	<p>1. 다음 명령을 실행합니다.</p> <pre>kubectl apply -f deploy_no_taint.yaml</pre> <p>2. 배포가 성공하면 이전에 실행한 것과 동일한 명령을 실행하여 테인트가 없는 노드 그룹의 포드 배치를 확인합니다.</p> <pre>kubectl describe node <node_name> grep "Taints"</pre> <p>출력은 다음과 같아야 합니다.</p> <pre>Taints: <none></pre> <p>이것으로 테스트가 완료됩니다.</p>	<p>앱 개발자, AWS DevOps, DevOps 엔지니어</p>

리소스 정리

작업	설명	필요한 기술
리소스를 정리합니다.	<p>실행 중인 리소스에 대한 AWS 요금이 발생하지 않도록 하려면 다음 명령을 사용합니다.</p> <pre>eksctl delete cluster --name <Name of the cluster> --region <region-code></pre>	AWS DevOps, 앱 개발자

문제 해결

문제	Solution
<p>시스템에서 arm64 아키텍처를 사용하는 경우 (특히 M1 Mac에서 실행하는 경우) 이러한 명령 중 일부가 실행되지 않을 수 있습니다. 다음 줄에 오류가 있을 수 있습니다.</p> <pre>FROM adoptopenjdk/openjdk11:jdk-11.0.14.1_1-alpine</pre>	<p>Dockerfile을 실행할 때 오류가 발생하면 해당 FROM 줄을 다음 줄로 바꿉니다.</p> <pre>FROM bellsoft/liberica-openjdk-alpine-musl:17</pre>

관련 리소스

- [Amazon EKS에 샘플 Java 마이크로서비스 배포](#)
- [Amazon ECR 프라이빗 리포지토리 생성](#)
- [노드에 포드 할당](#) (Kubernetes 설명서)
- [테인트 및 톨러레이션](#) (Kubernetes 설명서)
- [Amazon EKS](#)
- [Amazon ECR](#)
- [AWS CLI](#)

- [Docker](#)
- [IntelliJ IDEA CE](#)
- [Eclipse](#)

추가 정보

deployment.yaml

```
apiVersion: apps/v1
kind: Deployment
metadata:
  name: microservice-deployment
spec:
  replicas: 1
  selector:
    matchLabels:
      app.kubernetes.io/name: java-microservice
  template:
    metadata:
      labels:
        app.kubernetes.io/name: java-microservice
    spec:
      affinity:
        nodeAffinity:
          requiredDuringSchedulingIgnoredDuringExecution:
            nodeSelectorTerms:
              - matchExpressions:
                  - key: alpha.eksctl.io/nodegroup-name
                    operator: In
                    values:
                      - <node-group-name-from-cluster.yaml>
      tolerations: #only this pod has toleration and is viable to go to ng with taint
        - key: "<Taint key>" #classified_workload in our case
          operator: Equal
          value: "<Taint value>" #true
          effect: "NoSchedule"
        - key: "<Taint key>" #machine_learning_workload in our case
          operator: Equal
          value: "<Taint value>" #true
          effect: "NoSchedule"
    containers:
      - name: java-microservice-container
```

```

    image: <account_number>.dkr.ecr<region>.amazonaws.com/
    <repository_name>:latest
    ports:
      - containerPort: 4567

```

포드 예제 출력 설명

```

Name:          microservice-deployment-in-tainted-nodes-5684cc495b-vpcfx
Namespace:     default
Priority:      0
Node:          ip-192-168-29-181.us-west-1.compute.internal/192.168.29.181
Start Time:    Wed, 14 Sep 2022 11:06:47 -0400
Labels:        app.kubernetes.io/name=java-microservice-taint
                pod-template-hash=5684cc495b
Annotations:   kubernetes.io/psp: eks.privileged
Status:        Running
IP:            192.168.13.44
IPs:
  IP:          192.168.13.44
Controlled By: ReplicaSet/microservice-deployment-in-tainted-nodes-5684cc495b
Containers:
  java-microservice-container-1:
    Container ID:
      docker://5c158df8cc160de8f57f62f3ee16b12725a87510a809d90a1fb9e5d873c320a4
    Image:          934188034500.dkr.ecr.us-east-1.amazonaws.com/java-eks-apg
    Image ID:       docker-pullable://934188034500.dkr.ecr.us-east-1.amazonaws.com/
                    java-eks-apg@sha256:d223924aca8315aab20d54eddf3443929eba511b6433017474d01b63a4114835
    Port:           4567/TCP
    Host Port:      0/TCP
    State:          Running
      Started:      Wed, 14 Sep 2022 11:07:02 -0400
    Ready:          True
    Restart Count:  0
    Environment:    <none>
    Mounts:
      /var/run/secrets/kubernetes.io/serviceaccount from kube-api-access-ddvww (ro)
Conditions:
  Type           Status
  Initialized     True
  Ready           True
  ContainersReady True
  PodScheduled   True
Volumes:

```

```
kube-api-access-ddvww:
  Type: Projected (a volume that contains injected data from
multiple sources)
  TokenExpirationSeconds: 3607
  ConfigMapName: kube-root-ca.crt
  ConfigMapOptional: <nil>
  DownwardAPI: true
QoS Class: BestEffort
Node-Selectors: <none>
Tolerations: classified_workload=true:NoSchedule
              machine_learning_workload=true:NoSchedule
              node.kubernetes.io/not-ready:NoExecute op=Exists for 300s
              node.kubernetes.io/unreachable:NoExecute op=Exists for
300s
Events: <none>
```

필터링된 Amazon ECR 컨테이너 이미지를 계정 또는 리전 전반적으로 복제

작성자: Abdal Garuba

환경: 프로덕션

기술: 컨테이너 및 마이크로서비스, DevOps

AWS 서비스: 아마존 EC2 컨테이너 레지스트리, 아마존, AWS CodeBuild, CloudWatch AWS ID 및 액세스 관리, AWS CLI

요약

Amazon Elastic Container Registry(Amazon ECR)는 [교차 리전](#) 및 [교차 계정 복제](#) 기능을 사용하여 기본적으로 Amazon Web Services(AWS) 리전 및 계정 전반적으로 이미지 리포지토리에 있는 모든 컨테이너 이미지를 복제할 수 있습니다. (자세한 내용은 [Amazon ECR의 교차 리전 복제가 도입된 AWS 블로그](#) 게시물을 참조하십시오.) 하지만 리전 또는 계정 전반적으로 복사되는 이미지를 어떤 기준으로도 필터링할 수 있는 방법은 없습니다.

이 패턴은 이미지 태그 패턴을 기반으로 AWS 계정 및 리전 전반적으로 Amazon ECR에 저장된 컨테이너 이미지를 복제하는 방법을 설명합니다. 이 패턴은 Amazon CloudWatch Events를 사용하여 사전 정의된 사용자 지정 태그가 있는 이미지의 푸시 이벤트를 수신합니다. 푸시 이벤트는 AWS CodeBuild 프로젝트를 시작하고 이미지 세부 정보를 AWS 프로젝트에 전달합니다. CodeBuild 프로젝트는 제공된 세부 정보에 따라 원본 Amazon ECR 레지스트리에서 대상 레지스트리로 이미지를 복사합니다.

이 패턴은 계정 전반적으로 특정 태그가 있는 이미지를 복사합니다. 예를 들어 이 패턴을 사용하여 프로덕션 준비가 완료된 안전한 이미지만 프로덕션 계정에 복사할 수 있습니다. 개발 계정에서는 이미지를 철저히 테스트한 후 안전한 이미지에 사전 정의된 태그를 추가하고, 이 패턴의 절차를 사용하여 표시된 이미지를 프로덕션 계정에 복사할 수 있습니다.

사전 조건 및 제한 사항

사전 조건

- 소스 및 대상 Amazon ECR 레지스트리를 위한 활성 계정
- 이 패턴에 사용된 도구에 대한 관리 권한
- 테스트를 위해 로컬 시스템에 설치된 [Docker](#)

- Amazon ECR에 인증하기 위한 [Command Line Interface\(CLI\)](#)

제한 사항

- 이 패턴은 한 리전에서만 소스 레지스트리의 푸시 이벤트를 감시합니다. 이 패턴을 다른 리전에 배포하여 해당 리전의 레지스트리를 감시할 수 있습니다.
- 이 패턴에서는 Amazon CloudWatch Events 규칙 하나가 단일 이미지 태그 패턴을 수신합니다. 여러 패턴을 확인하려는 경우 추가 이미지 태그 패턴을 수신 대기하도록 이벤트를 추가할 수 있습니다.

아키텍처

대상 아키텍처

자동화 및 규모 조정

이 패턴은 코드형 인프라(IaC) 스크립트로 자동화하고 대규모로 배포할 수 있습니다. AWS CloudFormation 템플릿을 사용하여 이 패턴을 배포하려면 첨부 파일을 다운로드하고 [추가 정보](#) 섹션의 지침을 따르십시오.

사용자 지정 이벤트 패턴이 다른 여러 Amazon Events CloudWatch 이벤트를 동일한 AWS CodeBuild 프로젝트로 지정하여 여러 이미지 태그 패턴을 복제할 수 있지만, 여러 패턴을 지원하려면 다음과 같이 `buildspec.yaml` 파일 (첨부 파일 및 [도구](#) 섹션에 포함되어 있음) 의 보조 검증을 업데이트해야 합니다.

```
...
if [[ ${IMAGE_TAG} != release-* ]]; then
...

```

도구

Amazon 서비스

- [IAM](#) — Identity and Access Management(IAM)를 사용하여 서비스와 리소스에 대한 액세스를 안전하게 관리할 수 있습니다. 이 패턴에서는 컨테이너 이미지를 대상 레지스트리로 푸시할 때 AWS가 CodeBuild 맡게 되는 교차 계정 IAM 역할을 생성해야 합니다.
- [Amazon ECR](#) — Amazon Elastic Container Registry(Amazon ECR)는 컨테이너 이미지와 아티팩트를 어디서나 쉽게 저장, 관리, 공유, 배포할 수 있게 해주는 완전 관리형 컨테이너 레지스트리입니다.

소스 레지스트리로의 이미지 푸시 작업은 Amazon CloudWatch Events에서 픽업한 이벤트 버스로 시스템 이벤트 세부 정보를 전송합니다.

- [AWS CodeBuild](#) — CodeBuild AWS는 소스 코드 컴파일, 테스트 실행, 배포 준비가 완료된 아티팩트 생성 등의 작업을 수행할 수 있는 컴퓨팅 파워를 제공하는 완전관리형 지속적 통합 서비스입니다. 이 패턴은 CodeBuild AWS를 사용하여 원본 Amazon ECR 레지스트리에서 대상 레지스트리로 복사 작업을 수행합니다.
- [CloudWatch 이벤트](#) — Amazon CloudWatch Events는 AWS 리소스의 변경 사항을 설명하는 시스템 이벤트 스트림을 제공합니다. 이 패턴은 규칙을 사용하여, Amazon ECR 푸시 작업을 특정 이미지 태그 패턴과 일치시킵니다.

도구

- [Docker CLI](#) — Docker는 컨테이너를 더 쉽게 생성하고 관리할 수 있게 해주는 도구입니다. 컨테이너는 컨테이너 런타임을 지원하는 아무 플랫폼에서라도 쉽게 배포할 수 있는 하나의 장치 또는 패키지에 애플리케이션과 모든 종속성을 패키징합니다.

코드

다음과 같은 두 가지 방법으로 이 패턴을 구현할 수 있습니다.

- 자동 설정: 첨부 파일에 제공된 두 개의 AWS CloudFormation 템플릿을 배포합니다. 지침은 [추가 정보](#) 섹션을 참조하세요.
- 수동 설정: [에픽](#) 섹션의 절차를 따릅니다.

샘플 buildspec.yaml

이 패턴으로 제공된 CloudFormation 템플릿을 사용하는 경우 buildspec.yaml 파일이 CodeBuild 리소스에 포함됩니다.

```
version: 0.2
env:
  shell: bash
phases:
  install:
    commands:
      - export CURRENT_ACCOUNT=$(echo ${CODEBUILD_BUILD_ARN} | cut -d':' -f5)
      - export CURRENT_ECR_REGISTRY=${CURRENT_ACCOUNT}.dkr.ecr.
        ${AWS_REGION}.amazonaws.com
```

```

- export DESTINATION_ECR_REGISTRY=${DESTINATION_ACCOUNT}.dkr.ecr.
${DESTINATION_REGION}.amazonaws.com
pre_build:
  on-failure: ABORT
  commands:
    - echo "Validating Image Tag ${IMAGE_TAG}"
    - |
      if [[ ${IMAGE_TAG} != release-* ]]; then
        aws codebuild stop-build --id ${CODEBUILD_BUILD_ID}
        sleep 60
        exit 1
      fi
    - aws ecr get-login-password --region ${AWS_REGION} | docker login -u AWS --
password-stdin ${CURRENT_ECR_REGISTRY}
    - docker pull ${CURRENT_ECR_REGISTRY}/${REPO_NAME}:${IMAGE_TAG}
  build:
    commands:
      - echo "Assume cross-account role"
      - CREDENTIALS=$(aws sts assume-role --role-arn ${CROSS_ACCOUNT_ROLE_ARN} --
role-session-name Rolesession)
      - export AWS_DEFAULT_REGION=${DESTINATION_REGION}
      - export AWS_ACCESS_KEY_ID=$(echo ${CREDENTIALS} | jq -r
'.Credentials.AccessKeyId')
      - export AWS_SECRET_ACCESS_KEY=$(echo ${CREDENTIALS} | jq -r
'.Credentials.SecretAccessKey')
      - export AWS_SESSION_TOKEN=$(echo ${CREDENTIALS} | jq -r
'.Credentials.SessionToken')
      - echo "Logging into cross-account registry"
      - aws ecr get-login-password --region ${DESTINATION_REGION} | docker login -u
AWS --password-stdin ${DESTINATION_ECR_REGISTRY}
      - echo "Check if Destination Repository exists, else create"
      - |
        aws ecr describe-repositories --repository-names ${REPO_NAME} --region
${DESTINATION_REGION} \
        || aws ecr create-repository --repository-name ${REPO_NAME} --region
${DESTINATION_REGION}
      - echo "retag image and push to destination"
      - docker tag ${CURRENT_ECR_REGISTRY}/${REPO_NAME}:${IMAGE_TAG}
${DESTINATION_ECR_REGISTRY}/${REPO_NAME}:${IMAGE_TAG}
      - docker push ${DESTINATION_ECR_REGISTRY}/${REPO_NAME}:${IMAGE_TAG}

```

에픽

IAM 역할 생성

작업	설명	필요한 기술
<p>CloudWatch 이벤트 역할을 생성합니다.</p>	<p>원본 AWS 계정에서 Amazon CloudWatch Events가 맡을 IAM 역할을 생성합니다. 역할에는 AWS CodeBuild 프로젝트를 시작할 수 있는 권한이 있어야 합니다.</p> <p>CLI를 사용하여 역할을 생성하려면 IAM 설명서의 지침을 따릅니다.</p> <p>예제 신뢰 정책(trustpolicy.json):</p> <pre data-bbox="594 1035 1029 1551"> { "Version": "2012-10-17", "Statement": { "Effect": "Allow", "Principal": {"Service": "events.amazonaws.com"}, "Action": "sts:AssumeRole" } } </pre> <p>예제 권한 정책(permissionpolicy.json):</p> <pre data-bbox="594 1709 1029 1879"> { "Version": "2012-10-17", "Statement": { </pre>	<p>AWS 관리자, AWS DevOps, AWS 시스템 관리자, 클라우드 관리자, 클라우드 아키텍트, DevOps 엔지니어</p>

작업	설명	필요한 기술
	<pre data-bbox="597 205 1026 560">"Effect": "Allow", "Action": "codebuild:StartBuild", "Resource": "<CodeBuild Project ARN>" } }</pre>	

작업	설명	필요한 기술
<p>CodeBuild 역할 생성.</p>	<p>IAM 설명서의 지침에 따라 AWS가 CodeBuild 맡을 IAM 역할을 생성합니다. 이 역할에는 다음과 같은 권한도 있어야 합니다.</p> <ul style="list-style-type: none"> • 대상 교차 계정 역할을 떠맡을 수 있는 권한 • 로그 그룹 및 로그 스트림을 생성하고 로그 이벤트를 게시할 수 있는 권한 • 역할에 ContainerRegistryReadOnlyAmazonEC2 관리형 정책을 추가하여 모든 Amazon ECR 리포지토리에 대한 읽기 전용 권한 부여 • 중지할 수 있는 권한 CodeBuild <p>예제 신뢰 정책(trustpolicy.json):</p> <pre> { "Version": "2012-10-17", "Statement": [{ "Effect": "Allow", "Principal": { "Service": "codebuild.amazonaws.com" }, "Action": "sts:AssumeRole" }] } </pre>	<p>AWS 관리자, AWS DevOps, AWS 시스템 관리자, 클라우드 관리자, 클라우드 아키텍트, DevOps 엔지니어</p>

작업	설명	필요한 기술
	<pre>] } 예제 권한 정책(permissionpolicy.json): { "Version": "2012-10-17", "Statement": [{ "Action": ["codebuild:StartBuild", "codebuild:StopBuild", "codebuild:Get*", "codebuild:List*", "codebuild:BatchGet*"], "Resource": "*", "Effect": "Allow" }, { "Action": ["logs:CreateLogGroup", "logs:CreateLogStream", "logs:PutLogEvents"], </pre>	

작업	설명	필요한 기술
	<pre data-bbox="609 210 1015 976"> "Resource": "*", "Effect": "Allow" }, { "Action": "sts:AssumeRole", "Resource": "<ARN of destination role>", "Effect": "Allow", "Sid": "AssumeCrossAccountArn" }] } } </pre> <p data-bbox="592 1018 1031 1197">다음과 같이 관리형 정책 AmazonEC2ContainerRegistryReadOnly (을)를 CLI 명령에 연결합니다.</p> <pre data-bbox="609 1249 1015 1585"> ~\$ aws iam attach-role-policy \ --policy-arn arn:aws:iam::aws:policy/AmazonEC2ContainerRegistryReadOnly \ --role-name <name of CodeBuild Role> </pre>	

작업	설명	필요한 기술
<p>교차 계정 역할을 생성합니다.</p>	<p>대상 AWS 계정에서 원본 계정이 위임할 AWS CodeBuild 역할에 대한 IAM 역할을 생성합니다. 교차 계정 역할은 컨테이너 이미지가 새 리포지토리를 생성할 수 있게 하고, 컨테이너 이미지를 Amazon ECR에 업로드해야 합니다.</p> <p>CLI를 사용하여 IAM 역할을 생성하려면 IAM 설명서의 지침을 따릅니다.</p> <p>이전 단계의 AWS CodeBuild 프로젝트를 허용하려면 다음 신뢰 정책을 사용하십시오.</p> <pre data-bbox="592 982 1027 1539"> { "Version": "2012-10-17", "Statement": { "Effect": "Allow", "Principal": { "AWS": "<ARN of source codebuild role>" }, "Action": "sts:AssumeRole" } } </pre> <p>이전 단계의 AWS CodeBuild 프로젝트가 대상 레지스트리에 이미지를 저장하도록 허용하려면 다음 권한 정책을 사용하십시오.</p>	<p>AWS 관리자, AWS DevOps, 클라우드 관리자, 클라우드 아키텍트, DevOps 엔지니어, AWS 시스템 관리자</p>

작업	설명	필요한 기술
	<pre> { "Version": "2012-10-17", "Statement": [{ "Action": ["ecr:GetDownloadUr lForLayer", "ecr:BatchCheckLay erAvailability", "ecr:PutImage", "ecr:InitiateLayer Upload", "ecr:UploadLayerPa rt", "ecr:CompleteLayer Upload", "ecr:GetRepository Policy", "ecr:DescribeRepos itories", "ecr:GetAuthorizat ionToken", "ecr:CreateReposit ory"], "Resource": "*", "Effect": "Allow" } </pre>	

작업	설명	필요한 기술
	<pre>] } </pre>	

CodeBuild 프로젝트 생성

작업	설명	필요한 기술
<p>CodeBuild 프로젝트 생성.</p>	<p>AWS CodeBuild 설명서의 지침에 따라 소스 계정에서 AWS CodeBuild 프로젝트를 생성합니다. 프로젝트는 소스 레지스트리와 동일한 리전에 있어야 합니다.</p> <p>다음과 같이 프로젝트를 구성합니다.</p> <ul style="list-style-type: none"> • 환경 유형: LINUX CONTAINER • 서비스 역할: CodeBuild Role • 권한이 있는 모드: true • 환경 이미지: aws/codebuild/standard:x.x (이용 가능한 최신 이미지 사용) • 환경 변수: <ul style="list-style-type: none"> • CROSS_ACCOUNT_ROLE_ARN : 교차 계정 역할의 Amazon 리소스 이름 (ARN) • DESTINATION_REGION : 교차 계정 리전의 이름 	<p>AWS 관리자, AWS DevOps, AWS 시스템 관리자, 클라우드 관리자, 클라우드 아키텍트, DevOps 엔지니어</p>

작업	설명	필요한 기술
	<ul style="list-style-type: none"> • DESTINATION_ACCOUNT : 대상 계정의 번호 • 빌드 사양: 도구 섹션에 나열된 buildspec.yaml 파일을 사용합니다. 	

이벤트 생성

작업	설명	필요한 기술
이벤트 규칙을 생성합니다.	<p>패턴은 콘텐츠 필터링 기능을 사용하므로 Amazon을 사용하여 이벤트를 생성해야 EventBridge 합니다. EventBridge 설명서의 지침에 따라 몇 가지 수정을 거쳐 이벤트와 대상을 생성합니다.</p> <ul style="list-style-type: none"> • 패턴 정의에 대하여 이벤트 패턴을 선택한 다음 사용자 정의 패턴을 선택합니다. • 다음 사용자 지정 이벤트 패턴 샘플 코드를 제공되는 텍스트 상자에 복사합니다. <pre> { "source": ["aws.ecr"], "detail-type": ["ECR Image Action"], "detail": { "action-type": ["PUSH"], "result": ["SUCCESS"], </pre>	AWS 관리자, AWS DevOps, AWS 시스템 관리자, 클라우드 관리자, 클라우드 아키텍트, DevOps 엔지니어

작업	설명	필요한 기술
	<pre data-bbox="625 205 1031 430"> "image-tag": [{ "prefix": "release-"}] } } </pre> <ul data-bbox="592 441 1023 924" style="list-style-type: none"> • Select targets에서 AWS CodeBuild 프로젝트를 선택하고 이전 에픽에서 생성한 AWS CodeBuild 프로젝트의 ARN을 붙여넣습니다. • Configure input(입력 구성)에서 입력 변환기를 선택합니다. <ul data-bbox="625 840 998 924" style="list-style-type: none"> • 입력 경로 텍스트 상자에서 다음을 붙여넣습니다. <pre data-bbox="657 955 1031 1197"> {"IMAGE_TAG":"\$.detail.image-tag", "REPO_NAME":"\$.detail.repository-name" } </pre> <ul data-bbox="592 1207 1023 1344" style="list-style-type: none"> • 입력 템플릿 텍스트 상자에서 다음을 붙여넣습니다. <pre data-bbox="657 1375 1031 1732"> {"environmentVariablesOverride": [{"name": "IMAGE_TAG", "value":<IMAGE_TAG >}, {"name": "REPO_NAME", "value":<REPO_NAME>}]} </pre> <ul data-bbox="592 1753 1023 1837" style="list-style-type: none"> • 기존 역할 사용을 선택하고 IAM 역할 생성 에픽에서 이 	

작업	설명	필요한 기술
	전에 생성한 CloudWatch 이벤트 역할의 이름을 선택합니다.	

Validate

작업	설명	필요한 기술
Amazon ECR을 사용하여 인증합니다.	Amazon ECR 설명서 의 절차에 따라 소스 및 대상 레지스트리를 모두 인증합니다.	AWS 관리자, AWS DevOps, AWS 시스템 관리자, 클라우드 관리자, DevOps 엔지니어, 클라우드 아키텍트
이미지 복제를 테스트합니다.	소스 계정에서, release-로 접두사가 붙은 이미지 태그를 사용하여 컨테이너 이미지를 새 Amazon ECR 소스 리포지토리 또는 기존 Amazon ECR 소스 리포지토리에 푸시합니다. 이미지를 푸시하려면 Amazon ECR 설명서 의 절차를 따릅니다. CodeBuild 콘솔에서 CodeBuild 프로젝트 진행 상황을 모니터링할 수 있습니다. CodeBuild 프로젝트가 성공적으로 완료되면 대상 AWS 계정에 로그인하고 Amazon ECR 콘솔을 열고 대상 Amazon ECR 레지스트리에 이미지가 있는지 확인합니다.	AWS 관리자, AWS DevOps, AWS 시스템 관리자, 클라우드 관리자, 클라우드 아키텍트, DevOps 엔지니어
이미지 제외를 테스트합니다.	소스 계정에서, 사용자 지정 접두사가 없는 이미지 태그를 사	AWS 관리자, AWS DevOps, AWS 시스템 관리자, 클라우

작업	설명	필요한 기술
	<p>용하여 컨테이너 이미지를 신규 또는 기존 Amazon ECR 소스 리포지토리에 푸시합니다.</p> <p>CodeBuild 프로젝트가 시작되지 않았는지, 대상 레지스트리에 컨테이너 이미지가 표시되지 않는지 확인합니다.</p>	드 관리자, 클라우드 아키텍트, DevOps 엔지니어

관련 리소스

- [시작하기 CodeBuild](#)
- [아마존 시작하기 EventBridge](#)
- [Amazon EventBridge 이벤트 패턴의 콘텐츠 기반 필터링](#)
- [IAM 역할을 사용하여 계정 간 액세스 권한 위임](#)
- [프라이빗 이미지 복제](#)

추가 정보

이 패턴에 맞게 리소스를 자동으로 배포하려면 다음 절차를 따릅니다.

1. 첨부 파일을 다운로드하고 두 개의 CloudFormation 템플릿 (및) 을 추출하십시오. `part-1-copy-tagged-images.yaml` `part-2-destination-account-role.yaml`
2. [AWS CloudFormation 콘솔에](#) 로그인하여 원본 Amazon ECR 레지스트리와 동일한 AWS 계정 및 `part-1-copy-tagged-images.yaml` 지역에 배포하십시오. 필요에 따라 파라미터를 업데이트합니다. 템플릿은 다음 리소스를 배포합니다.
 - 아마존 CloudWatch 이벤트 IAM 역할
 - AWS CodeBuild 프로젝트 IAM 역할
 - AWS CodeBuild 프로젝트
 - AWS CloudWatch 이벤트 규칙
3. 출력 탭에서 `SourceRoleName`의 값을 기록해 둡니다. 다음 단계에서 이 값이 필요합니다.

4. Amazon ECR 컨테이너 이미지를 복사하려는 AWS 계정에 두 번째 CloudFormation 템플릿인 을 배포합니다. `part-2-destination-account-role.yaml` 필요에 따라 파라미터를 업데이트합니다. `SourceRoleName` 파라미터에 단계 3의 값을 지정합니다. 이 템플릿은 교차 계정 IAM 역할을 배포합니다.
5. [에픽](#) 섹션의 마지막 단계에 설명된 대로 이미지 복제 및 제외를 검증합니다.

첨부

이 문서와 관련된 추가 콘텐츠에 액세스하려면 [attachment.zip](#) 파일의 압축을 풉니다.

컨테이너를 다시 시작하지 않고 데이터베이스 보안 인증 교체

작성자: Josh Joy

환경: 프로덕션

기술: 컨테이너 및 마이크로서비스, 데이터베이스, 인프라, 보안, ID, DevOps 규정 준수, 관리 및 거버넌스

AWS 서비스: Amazon ECS, Amazon Aurora, Fargate, Secrets Manager, Amazon VPC

요약

Amazon Web Services(AWS) 클라우드에서는 Secrets Manager를 사용하여 수명 주기 내내 데이터베이스 보안 인증을 교체, 관리, 검색할 수 있습니다. 사용자와 애플리케이션은 Secrets Manager API를 호출하여 보안 암호를 검색하여, 일반 텍스트로 된 민감한 정보를 코딩할 필요를 없었습니다.

마이크로서비스 워크로드에 컨테이너를 사용하는 경우 AWS Secrets Manager에 보안 인증을 안전하게 저장할 수 있습니다. 구성과 코드를 분리하기 위해 일반적으로 이러한 보안 인증이 컨테이너에 삽입됩니다. 하지만 보안 인증을 주기적으로 자동으로 교체하는 것이 중요합니다. 해지 후 보안 인증을 새로 고칠 수 있는 기능을 지원하는 것도 중요합니다. 동시에 애플리케이션에는 다운스트림 가용성에 미치는 잠재적 영향을 줄이면서 보안 인증을 교체할 수 있는 기능이 필요합니다.

이 패턴은 컨테이너를 다시 시작할 필요 없이 컨테이너 내에서 AWS Secrets Manager로 보안 처리된 보안 암호를 교체하는 방법을 설명합니다. 또한 이 패턴은 Secrets Manager [클라이언트 측 캐싱 구성 요소](#)를 사용하여 Secrets Manager에 대한 보안 인증 조회 횟수를 줄입니다. 클라이언트 측 캐싱 구성 요소를 사용하여 애플리케이션 내에서 보안 인증을 새로 고치는 경우 교체된 보안 인증을 가져오기 위해 컨테이너를 다시 시작할 필요가 없습니다.

이러한 접근은 Amazon Elastic Kubernetes Service(Amazon EKS) 및 Amazon Elastic Container Service(Amazon ECS)에 효과적입니다.

[두 가지 시나리오가 다루어집니다.](#) 단일 사용자 시나리오에서는 만료된 보안 인증을 탐지하여 보안 암호 교체 시 데이터베이스 보안 인증을 새로 고칩니다. 보안 인증 캐시에 보안 암호를 새로 고치라는 명령이 내려지면 애플리케이션이 데이터베이스 연결을 다시 설정합니다. 클라이언트 측 캐싱 구성 요소는 애플리케이션 내에서 보안 인증을 캐시하므로 보안 인증을 조회할 때마다 Secrets Manager에 접속하지 않아도 됩니다. 컨테이너를 다시 시작하여 보안 인증을 강제로 새로 고칠 필요 없이 애플리케이션 내에서 보안 인증이 교체됩니다.

두 번째 시나리오에서는 두 사용자를 번갈아 가며 보안 암호를 교체합니다. 활성 사용자가 두 명이면 한 사용자의 보안 인증이 항상 활성 상태이므로 가동 중지 가능성이 줄어듭니다. 두 명의 사용자 보안 인증 교체는 보안 인증 업데이트의 전파 지연이 약간 발생할 수 있는 클러스터를 포함한 대규모 배포의 경우 유용합니다.

사전 조건 및 제한 사항

사전 조건

- 활성 상태의 AWS 계정.
- Amazon EKS 또는 Amazon ECS의 컨테이너에서 실행되는 애플리케이션입니다.
- 보안 인증은 [교체가 활성화된 상태로](#) Secrets Manager에 저장됩니다.
- 두 번째 보안 인증 집합은 두 명의 사용자 솔루션을 배포하는 경우 Secrets Manager에 저장됩니다. [코드 예제는 GitHub repo aws-secrets-manager-rotation-lambdas 저장소에서 찾을 수 있습니다.](#)
- Amazon Aurora 데이터베이스.

제한 사항

- 이 예제는 Python 애플리케이션을 대상으로 합니다. Java 애플리케이션의 경우 Secrets Manager용으로 [Java 클라이언트 측 캐싱 구성 요소](#) 또는 [JDBC 클라이언트 측 캐싱 라이브러리](#)를 사용할 수 있습니다.

아키텍처

대상 아키텍처

시나리오 1 — 단일 사용자의 보안 인증 교체

첫 번째 시나리오에서는 Secrets Manager가 단일 데이터베이스 보안 인증을 주기적으로 교체합니다. 애플리케이션 컨테이너는 Fargate에서 실행됩니다. 첫 번째 데이터베이스 연결이 설정되면 애플리케이션 컨테이너는 Aurora의 데이터베이스 보안 인증을 가져옵니다. 그러면 Secrets Manager 캐싱 구성 요소가 향후 연결 설정을 위해 보안 인증을 캐시합니다. 교체 기간이 경과하면 보안 인증이 만료되고 데이터베이스에서 인증 오류가 반환됩니다. 그런 다음 애플리케이션은 교체된 보안 인증을 가져와 캐시를 무효화하고 Secrets Manager 클라이언트 측 캐싱 구성 요소를 통해 보안 인증 캐시를 업데이트합니다.

이 시나리오에서는 보안 인증을 교체하고 오래된 연결에서 오래된 보안 인증을 사용하는 동안 중단이 최소화될 수 있습니다. 이 문제는 두 명의 사용자 시나리오를 사용하여 해결할 수 있습니다.

시나리오 2 — 단일 사용자의 보안 인증 교체

두 번째 시나리오에서는 Secrets Manager가 두 개의 데이터베이스 사용자 보안 인증(Alice와 Bob의 보안 인증)을 주기적으로 교체합니다. 애플리케이션 컨테이너는 Fargate 클러스터에서 실행됩니다. 첫 번째 데이터베이스 연결이 설정되면, 애플리케이션 컨테이너는 첫 번째 사용자(Alice)를 위한 Aurora의 데이터베이스 보안 인증을 가져옵니다. 그러면 Secrets Manager 캐싱 구성 요소가 향후 연결 설정을 위해 보안 인증을 캐시합니다.

두 명의 사용자와 보안 인증이 있지만 Secrets Manager는 활성 보안 인증을 하나만 관리합니다. 이 경우 캐싱 구성 요소는 주기적으로 만료되어 최신 보안 인증을 가져옵니다. Secrets Manager 교체 기간이 캐시 제한 시간보다 더 길면 캐싱 구성 요소가 두 번째 사용자(Bob)의 교체된 보안 인증을 선택합니다. 예를 들어 캐시 만료가 분 단위로 측정되고 순환 기간이 일 단위로 측정되는 경우 캐싱 구성 요소는 정기적인 캐시 새로 고침의 일환으로 새 보안 인증을 가져옵니다. 이렇게 하면 각 사용자의 보안 인증이 한 번의 Secrets Manager 교체 동안 활성화되므로 가동 중지 시간이 최소화됩니다.

자동화 및 규모 조정

[CloudFormationAWS](#)를 사용하여 [인프라를 코드로](#) 사용하여 이 패턴을 배포할 수 있습니다. 그러면 애플리케이션 컨테이너를 빌드 및 생성하고, Fargate 작업을 생성하며, 컨테이너를 Fargate에 배포하고, Aurora를 사용하여 Secrets Manager를 설정 및 구성합니다. step-by-step 배포 지침은 [readme](#) 파일을 참조하십시오.

도구

도구

- [AWS Secrets Manager](#)는 코드의 암호를 포함해 하드 코딩된 보안 인증을 Secrets Manager에서 프로그래밍 방식으로 보안 암호를 검색하도록 하는 API 직접 호출로 바꿀 수 있습니다. Secrets Manager는 일정에 따라 자동으로 보안 암호를 교체할 수 있으므로 장기 보안 암호를 단기 보안 암호로 대체하여 보안 침해 위험을 줄일 수 있습니다.
- [Docker](#)를 사용하면 개발자가 모든 애플리케이션을 가볍고 휴대가 간편하며 자급자족할 수 있는 컨테이너로 포장, 배송 및 실행할 수 있습니다.

code

Python 코드 예제

이 패턴은 Secrets Manager의 Python 클라이언트 측 캐싱 구성 요소를 사용하여, 데이터베이스 연결을 설정하는 동안 인증 보안 인증을 검색합니다. 클라이언트 측 캐싱 구성 요소를 사용하면 매번 Secrets Manager에 접속하지 않아도 됩니다.

이제 교체 기간이 경과하면 캐시된 보안 인증이 만료되고, 데이터베이스에 연결하면 인증 오류가 발생합니다. MySQL의 경우 인증 오류 코드는 1045입니다. 이 예제에서는 MySQL용 Amazon Aurora를 사용하지만 PostgreSQL과 같은 다른 엔진을 사용할 수도 있습니다. 인증 오류가 발생하면 데이터베이스 연결 예외 처리 코드가 오류를 캐시합니다. 그런 다음 Secrets Manager 클라이언트 측 캐싱 구성 요소에 암호를 새로 고친 다음 다시 인증하고 데이터베이스 연결을 재설정하도록 알립니다. PostgreSQL이나 다른 엔진을 사용하는 경우 해당하는 인증 오류 코드를 찾아봐야 합니다.

이제 컨테이너 애플리케이션은 컨테이너를 다시 시작하지 않고도 교체된 암호로 데이터베이스 암호를 업데이트할 수 있습니다.

데이터베이스 연결을 처리하는 애플리케이션 코드에 다음 코드를 넣습니다. 이 예제에서는 Django를 사용하며, 연결용으로 데이터베이스 래퍼를 사용하여 데이터베이스 백엔드를 [하위 클래스](#)로 만듭니다. 다른 프로그래밍 언어나 데이터베이스 연결 라이브러리를 사용하는 경우 데이터베이스 연결 라이브러리를 참조하여 데이터베이스 연결 검색을 하위 클래스로 만드는 방법을 검토합니다.

```
def get_new_connection(self, conn_params):
    try:
        logger.info("get connection")
        databascredentials.get_conn_params_from_secrets_manager(conn_params)
        conn =super(DatabaseWrapper,self).get_new_connection(conn_params)
        return conn
    except MySQLdb.OperationalError as e:
        error_code=e.args[0]
        if error_code!=1045:
            raise e

        logger.info("Authentication error. Going to refresh secret and try again.")
        databascredentials.refresh_now()
        databascredentials.get_conn_params_from_secrets_manager(conn_params)
        conn=super(DatabaseWrapper,self).get_new_connection(conn_params)
        logger.info("Successfully refreshed secret and established new database
connection.")
        return conn
```

CloudFormation AWS와 Python 코드

- <https://github.com/aws-samples/aws-secrets-manager-credential-rotation-without-container-restart>

에픽

보안 인증 교체 중에도 애플리케이션 가용성 유지

작업	설명	필요한 기술
캐싱 구성 요소를 설치합니다.	Python용 Secrets Manager 클라이언트 측 캐싱 구성 요소를 다운로드하여 설치합니다. 다운로드 링크는 관련 리소스 섹션을 참조하십시오.	개발자
작동 가능한 자격증명을 캐싱합니다.	Secrets Manager 클라이언트 측 캐싱 구성 요소를 사용하여 작동 가능한 보안 인증을 로컬로 캐싱합니다.	개발자
데이터베이스 연결에서 승인되지 않은 오류가 발생한 경우 보안 인증을 새로 고치도록 애플리케이션 코드를 업데이트합니다.	Secrets Manager를 사용하여 데이터베이스 보안 인증을 가져오고 새로 고치도록 애플리케이션 코드를 업데이트합니다. 승인되지 않은 오류 코드를 처리하는 로직을 추가한 다음 새로 교체된 보안 인증을 가져옵니다. 예제 Python 코드 섹션을 참조하십시오.	개발자

관련 리소스

Secrets Manager 보안 암호 생성

- [AWS KMS에서 키 생성](#)
- [AWS Secrets Manager를 사용한 보안 암호 생성 및 관리](#)

Amazon Aurora 클러스터 생성

- [Amazon RDS DB 인스턴스 생성](#)

Amazon ECS 구성 요소 생성

- [클래식 콘솔을 사용하여 클러스터 생성](#)
- [Docker 이미지 생성](#)
- [프라이빗 리포지토리 생성](#)
- [Amazon ECR 프라이빗 레지스트리](#)
- [Docker 이미지 푸시하기](#)
- [Amazon ECS 태스크 정의](#)
- [클래식 콘솔에서 Amazon ECS 서비스 생성](#)

Download and install the Secrets Manager 클라이언트 측 캐싱 구성 요소 다운로드 및 설치

- [Python 캐싱 클라이언트](#)

첨부

이 문서와 관련된 추가 콘텐츠에 액세스하려면 [attachment.zip](#) 파일의 압축을 풉니다.

Amazon ECS Anywhere를 WorkSpaces 사용하여 Amazon에서 Amazon ECS 작업을 실행합니다.

작성자: Akash Kumar(AWS)

환경: 프로덕션

기술: 컨테이너 및 마이크로서비스, 현대화

워크로드: 기타 모든 워크로드

AWS 서비스: 아마존 ECS, 아마존 WorkSpaces, AWS 디렉터리 서비스

요약

Amazon Elastic Container Service(Amazon ECS) Anywhere는 Amazon Web Services(AWS) 관리 인프라 및 고객 관리형 인프라를 비롯한 모든 환경에서 Amazon ECS 작업 배포를 지원합니다. 클라우드에서 실행되고 항상 최신 상태로 유지되는 완전한 AWS 관리형 컨트롤 플레인을 사용하면서 이 작업을 수행할 수 있습니다.

기업에서는 컨테이너 기반 애플리케이션 개발에 WorkSpaces Amazon을 자주 사용합니다. 이를 위해서는 ECS 작업을 테스트하고 실행하기 위해 Amazon ECS 클러스터와 함께 Amazon Elastic Compute Cloud(Amazon EC2) 또는 AWS Fargate가 필요했습니다. 이제 Amazon ECS Anywhere를 사용하여 WorkSpaces Amazon을 ECS 클러스터에 외부 인스턴스로 직접 추가하고 작업을 직접 실행할 수 있습니다. 이렇게 하면 Amazon WorkSpaces 로컬에서 ECS 클러스터를 사용하여 컨테이너를 테스트할 수 있으므로 개발 시간이 단축됩니다. 또한 컨테이너 애플리케이션을 테스트하기 위해 EC2 또는 Fargate 인스턴스를 사용하는 비용을 절감할 수 있습니다.

이 패턴은 Amazon WorkSpaces ECS Anywhere를 사용하여 Amazon에 ECS 작업을 배포하는 방법을 보여줍니다. ECS 클러스터를 설정하고 AWS Directory Service Simple AD를 사용하여 를 시작합니다. WorkSpaces 그런 다음 예제 ECS 태스크는 에서 NGINX를 시작합니다. WorkSpaces

사전 조건 및 제한 사항

- 활성 상태의 AWS 계정.
- AWS Command Line Interface(AWS CLI)
- [머신에 구성된](#) AWS 보안 인증

아키텍처

대상 기술 스택

- Virtual Private Cloud(VPC)
- Amazon ECS 클러스터
- 아마존 WorkSpaces
- AWS Directory Service와 Simple AD

대상 아키텍처

아키텍처에는 다음 서비스와 리소스가 포함되어 있습니다.

- 사용자 지정 VPC에 퍼블릭 및 프라이빗 서브넷이 있는 ECS 클러스터
- 사용자에게 Amazon에 대한 액세스를 제공하는 VPC의 Simple AD WorkSpaces
- Amazon은 WorkSpaces Simple AD를 사용하여 VPC에 프로비저닝했습니다.
- 아마존을 관리형 인스턴스로 추가하기 위해 AWS Systems WorkSpaces Manager가 활성화되었습니다.
- Amazon ECS와 AWS Systems Manager 에이전트 (SSM 에이전트) 를 사용하여 Amazon은 Systems Manager와 ECS 클러스터에 WorkSpaces 추가했습니다.
- ECS 클러스터에서 실행할 ECS 작업의 예 WorkSpaces

도구

- [AWS Directory Service Simple Active Directory\(Simple AD\)](#)는 Samba 4 Active Directory 호환 서버를 기반으로 하는 독립형 관리형 디렉터리입니다. Simple AD는 사용자를 관리하고 Amazon EC2 인스턴스에 안전하게 연결하는 기능을 포함하여 AWS Managed Microsoft AD에서 제공하는 일부 기능을 제공합니다.
- [Amazon Elastic Container Service\(Amazon ECS\)](#)는 클러스터에서 컨테이너를 실행, 중지 및 관리하는 데 도움이 되는 빠르고 확장 가능한 컨테이너 관리 서비스입니다.
- [AWS Identity and Access Management\(IAM\)](#)는 누구에게 인증 및 사용 권한이 있는지 제어하여 AWS 리소스에 대한 액세스를 안전하게 관리할 수 있도록 도와줍니다.

- [AWS Systems Manager](#)는 AWS 클라우드에서 실행되는 애플리케이션과 인프라를 관리하는 데 도움이 됩니다. 애플리케이션 및 리소스 관리를 간소화하고, 운영 문제의 감지 및 해결 시간을 단축하며, AWS 리소스를 규모에 따라 안전하게 관리하는 데 도움이 됩니다.
- [WorkSpacesAmazon](#)은 사용자를 위해 가상 클라우드 기반 Microsoft Windows 또는 Amazon Linux 데스크톱 (일명) 을 프로비저닝할 수 있도록 지원합니다. WorkSpaces WorkSpaces 하드웨어를 조달 및 배포하거나 복잡한 소프트웨어를 설치할 필요가 없습니다.

에픽

ECS 클러스터 설정

작업	설명	필요한 기술
ECS 클러스터를 생성하고 구성합니다.	<p>ECS 클러스터를 생성하려면 다음 단계를 포함하여 AWS 설명서의 지침을 따르십시오.</p> <ul style="list-style-type: none"> • 클러스터 호환성 선택에서 네트워킹 전용을 선택합니다. 그러면 WorkSpace Amazon을 ECS 클러스터의 외부 인스턴스로 지원할 수 있습니다. • 새 VPC 생성을 선택합니다. 	클라우드 아키텍트

아마존 런칭 WorkSpaces

작업	설명	필요한 기술
Simple AD를 설정하고 Amazon을 시작합니다 WorkSpaces.	<p>새로 생성한 VPC를 위한 Simple AD 디렉터리를 프로비저닝하고 Amazon을 WorkSpaces 시작하려면 AWS 설명서의 지침을 따르십시오.</p>	클라우드 아키텍트

하이브리드 환경을 위한 AWS Systems Manager 설정

작업	설명	필요한 기술
<p>첨부된 스크립트를 다운로드하십시오.</p>	<p>로컬 머신에서 첨부 섹션에 있는 <code>ssm-trust-policy.json</code> 및 <code>ssm-activation.json</code> 파일을 다운로드합니다.</p>	<p>클라우드 아키텍트</p>
<p>IAM 역할을 추가합니다.</p>	<p>비즈니스 요구 사항에 따라 환경 변수를 추가합니다.</p> <pre data-bbox="597 726 1027 1163"> export AWS_DEFAULT_REGION=\${AWS_REGION_ID} export ROLE_NAME=\${ECS_TASK_ROLE} export CLUSTER_NAME=\${ECS_CLUSTER_NAME} export SERVICE_NAME=\${ECS_CLUSTER_SERVICE_NAME} </pre> <p>다음 명령을 실행합니다.</p> <pre data-bbox="597 1272 1027 1507"> aws iam create-role --role-name \$ROLE_NAME --assume-role-policy-document file://ssm-trust-policy.json </pre>	<p>클라우드 아키텍트</p>
<p>AmazonSSM ManagedInstanceCore 정책을 IAM 역할에 추가합니다.</p>	<p>다음 명령을 실행합니다.</p> <pre data-bbox="597 1623 1027 1795"> aws iam attach-role-policy --role-name \$ROLE_NAME --policy-arn arn:aws:iam::aws:p </pre>	<p>클라우드 아키텍트</p>

작업	설명	필요한 기술
	<pre>olicy/AmazonSSMManagedInstanceCore</pre>	
Amazon EC2 EC2 Container Servicefor 역할 정책을 IAM 역할에 추가합니다.	<p>다음 명령을 실행합니다.</p> <pre>aws iam attach-role-policy --role-name \$ROLE_NAME --policy-arn arn:aws:iam::aws:policy/service-role/AmazonEC2ContainerServiceforEC2Role</pre>	클라우드 아키텍트
IAM 역할을 확인합니다.	<p>IAM 역할을 확인하려면 다음 명령을 실행합니다.</p> <pre>aws iam list-attached-role-policies --role-name \$ROLE_NAME</pre>	클라우드 아키텍트
Systems Manager를 활성화합니다.	<p>다음 명령을 실행합니다.</p> <pre>aws ssm create-activation --iam-role \$ROLE_NAME tee ssm-activation.json</pre>	클라우드 아키텍트

ECS WorkSpaces 클러스터에 추가

작업	설명	필요한 기술
귀하의 컴퓨터에 연결하십시오 WorkSpaces.	WorkSpaces에 연결하고 설정하려면 AWS 설명서 의 지침을 따르십시오.	앱 개발자

작업	설명	필요한 기술
ecs-anywhere 설치 스크립트를 다운로드하십시오.	<p>명령 프롬프트에서 다음 명령을 실행합니다.</p> <pre>curl -o "ecs-anywhere-install.sh" "https://amazon-ecs-agent-packages-preview.s3.us-east-1.amazonaws.com/ecs-anywhere-install.sh" && sudo chmod +x ecs-anywhere-install.sh</pre>	앱 개발자
셸 스크립트의 무결성을 확인합니다.	<p>(선택 사항) 다음 명령을 실행합니다.</p> <pre>curl -o "ecs-anywhere-install.sh.sha256" "https://amazon-ecs-agent-packages-preview.s3.us-east-1.amazonaws.com/ecs-anywhere-install.sh.sha256" && sha256sum -c ecs-anywhere-install.sh.sha256</pre>	앱 개발자
Amazon Linux에 EPEL 리포지토리를 추가합니다.	Enterprise Linux용 추가 패키지(EPEL) 리포지토리를 추가하려면 <code>sudo amazon-linux-extras install epel -y</code> 명령을 실행합니다.	앱 개발자

작업	설명	필요한 기술
Amazon ECS Anywhere를 설치합니다.	<p>다음 명령을 사용하여 설치 스크립트를 실행합니다.</p> <pre>sudo ./ecs-anywhere-install.sh --cluster \$CLUSTER_NAME --activation-id \$ACTIVATION_ID --activation-code \$ACTIVATION_CODE --region \$AWS_REGION</pre>	
ECS 클러스터에서 인스턴스 정보를 확인합니다.	<p>Systems Manager 및 ECS 클러스터 인스턴스 정보를 확인하고 클러스터에 WorkSpaces 추가된 인스턴스 정보를 확인하려면 로컬 시스템에서 다음 명령을 실행합니다.</p> <pre>aws ssm describe-instance-information" && "aws ecs list-container-instances --cluster \$CLUSTER_NAME</pre>	앱 개발자

에 대한 ECS 작업을 추가합니다. WorkSpaces

작업	설명	필요한 기술
작업 실행 IAM 역할을 생성하십시오.	<p>첨부 섹션에서 <code>task-execution-assume-role.json</code> 및 <code>external-task-definition.json</code> 를 다운로드합니다.</p>	클라우드 아키텍트

작업	설명	필요한 기술
	<p>로컬 머신에서 다음 명령을 실행합니다.</p> <pre>aws iam --region \$AWS_DEFAULT_REGION create-role -- role-name \$ECS_TASK _EXECUTION_ROLE -- assume-role-policy- document file://ta sk-execution-assume- role.json</pre>	
<p>실행 역할에 정책을 추가하십시오.</p>	<p>다음 명령을 실행합니다.</p> <pre>aws iam --region \$AWS_DEFAULT_REGION attach-role-policy --role-name \$ECS_TASK _EXECUTION_ROLE -- policy-arn arn:aws:i am::aws:policy/ser vice-role/AmazonEC STaskExecutionRole Policy</pre>	<p>클라우드 아키텍트</p>
<p>작업 역할을 생성하십시오.</p>	<p>다음 명령을 실행합니다.</p> <pre>aws iam --region \$AWS_DEFAULT_REGION create-role -- role-name \$ECS_TASK _EXECUTION_ROLE -- assume-role-policy- document file://ta sk-execution-assume- role.json</pre>	<p>클라우드 아키텍트</p>

작업	설명	필요한 기술
클러스터에 작업 정의를 등록하십시오.	<p>로컬 머신에서 다음 명령을 실행합니다.</p> <pre>aws ecs register-task-definition --cli-input-json file://external-task-definition.json</pre>	클라우드 아키텍트
작업을 실행하십시오.	<p>로컬 머신에서 다음 명령을 실행합니다.</p> <pre>aws ecs run-task --cluster \$CLUSTER_NAME --launch-type EXTERNAL --task-definition nginx</pre>	클라우드 아키텍트
작업 실행 상태를 확인합니다.	<p>작업 ID를 가져오려면 다음 명령을 실행합니다.</p> <pre>export TEST_TASKID=\$(aws ecs list-tasks --cluster \$CLUSTER_NAME jq -r '.taskArns[0]')</pre> <p>작업 ID로 다음 명령을 실행합니다.</p> <pre>aws ecs describe-tasks --cluster \$CLUSTER_NAME --tasks \${TEST_TASKID}</pre>	클라우드 아키텍트

작업	설명	필요한 기술
에서 작업을 확인합니다. Workspace	NGINX가 에서 실행되고 있는지 확인하려면 명령을 실행합니다. <code>Workspace curl http://localhost:8080</code>	앱 개발자

관련 리소스

- [ECS 클러스터](#)
- [하이브리드 환경 설정](#)
- [아마존 WorkSpaces](#)
- [Simple AD](#)

첨부

이 문서와 관련된 추가 콘텐츠에 액세스하려면 [attachment.zip](#) 파일의 압축을 풉니다.

Amazon EC2 리눅스 인스턴스에서 ASP.NET 코어 웹 API Docker 컨테이너를 실행합니다.

작성자: Vijai Anand Ramalingam(AWS) 및 Sreelaxmi Pai(AWS)

환경: PoC 또는 파일럿

기술: 컨테이너 및 마이크로서비스, 소프트웨어 개발 및 테스트, 웹 및 모바일 앱

워크로드: Microsoft

AWS 서비스: Amazon EC2, Elastic Load Balancing (ELB)

요약

이 패턴은 Amazon Web Services(AWS) 클라우드에서 애플리케이션을 컨테이너화하려는 사용자를 위한 것입니다. 클라우드에서 앱을 컨테이너화하기 시작하면 일반적으로 컨테이너 오케스트레이션 플랫폼이 설정되지 않습니다. 이 패턴을 사용하면 정교한 컨테이너 오케스트레이션 인프라 없이도 AWS에 인프라를 빠르게 설정하여 컨테이너식 애플리케이션을 테스트할 수 있습니다.

현대화 여정의 첫 번째 단계는 애플리케이션을 혁신하는 것입니다. 레거시 .NET Framework 애플리케이션인 경우 먼저 런타임을 ASP.NET Core로 변경해야 합니다. 뒤이어 다음과 같이 하십시오.

- Docker 컨테이너 이미지 생성
- 해당 이미지에서 Docker 컨테이너를 실행합니다.
- Amazon Elastic Container Service(Amazon ECS) 또는 Amazon Elastic Kubernetes Service(Amazon EKS)와 같은 컨테이너 오케스트레이션 플랫폼에 배포하기 전에 애플리케이션의 유효성을 검사하십시오.

이 패턴은 Amazon Elastic Compute Cloud(Amazon EC2) Linux 인스턴스에서 최신 애플리케이션 개발의 빌드, 실행 및 검증 측면을 다룹니다.

사전 조건 및 제한 사항

사전 조건

- [활성 Amazon Web Services\(AWS\) 계정](#)
- 이 패턴에 대한 [AWS 리소스를 생성하기에 충분한 액세스 권한이 있는 AWS Identity and Access Management\(IAM\) 역할](#)
- [비주얼 스튜디오 커뮤니티 2022](#) 이상 다운로드 및 설치
- ASP.NET Core로 현대화된 .NET 프레임워크 프로젝트
- 리포지토리 GitHub

제품 버전

- Visual Studio Community 2022 이상

아키텍처

대상 아키텍처

이 패턴은 [AWS CloudFormation 템플릿](#)을 사용하여 다음 다이어그램에 표시된 고가용성 아키텍처를 생성합니다. Amazon EC2 Linux 인스턴스는 프라이빗 서브넷에서 시작됩니다. AWS Systems Manager 세션 관리자는 프라이빗 Amazon EC2 Linux 인스턴스에 액세스하고 Docker 컨테이너에서 실행되는 API를 테스트하는 데 사용됩니다.

1. 세션 관리자를 통해 Linux 인스턴스에 액세스할 수 있습니다.

도구

서비스

- [AWS 명령줄 인터페이스](#) - AWS Command Line Interface(AWS CLI)는 명령줄 셸에서 명령을 통해 AWS 서비스와 상호 작용할 수 있는 오픈 소스 도구입니다. 최소한의 구성으로 브라우저 기반 AWS Management Console에서 제공하는 것과 동일한 기능을 구현하는 AWS CLI 명령을 실행할 수 있습니다.
- [AWS Management Console](#) - AWS Management Console은 AWS 서비스 관리를 위한 다양한 리소스 콘솔의 모음을 구성하는 웹 애플리케이션입니다. 처음 로그인하면 콘솔 홈 페이지가 나타납니다. 홈 페이지는 각 서비스 콘솔에 대한 액세스와 관련 작업을 수행하는 데 필요한 정보에 액세스할 수 있는 단일 위치를 제공합니다.

- [AWS Systems Manager 세션 관리자](#) - 세션 관리자는 완전 관리형 AWS Systems Manager 기능입니다. Session Manager를 사용하면 Amazon Elastic Compute Cloud(Amazon EC2) 인스턴스를 관리할 수 있습니다. 세션 관리자는 인바운드 포트를 열거나 배스천 호스트를 유지 관리하거나 SSH 키를 관리할 필요 없이 안전하고 감사 가능한 노드 관리를 제공합니다.

기타 도구

- [비주얼 스튜디오 2022](#) — 비주얼 스튜디오 2022는 통합 개발 환경 (IDE) 입니다.
- [Docker](#) — Docker는 운영 체제 수준의 가상화를 사용하여 컨테이너에 소프트웨어를 제공하는 서비스형 플랫폼 (PaaS) 제품 세트입니다.

코드

```
FROM mcr.microsoft.com/dotnet/aspnet:5.0 AS base
WORKDIR /app
EXPOSE 80
EXPOSE 443

FROM mcr.microsoft.com/dotnet/sdk:5.0 AS build
WORKDIR /src
COPY ["DemoNetCoreWebAPI/DemoNetCoreWebAPI.csproj", "DemoNetCoreWebAPI/"]
RUN dotnet restore "DemoNetCoreWebAPI/DemoNetCoreWebAPI.csproj"
COPY . .
WORKDIR "/src/DemoNetCoreWebAPI"
RUN dotnet build "DemoNetCoreWebAPI.csproj" -c Release -o /app/build

FROM build AS publish
RUN dotnet publish "DemoNetCoreWebAPI.csproj" -c Release -o /app/publish

FROM base AS final
WORKDIR /app
COPY --from=publish /app/publish .
ENTRYPOINT ["dotnet", "DemoNetCoreWebAPI.dll"]
```

에픽

ASP.NET Core 웹 API 개발

작업	설명	필요한 기술
<p>Visual Studio를 사용하여 예제 ASP.NET Core 웹 API를 생성합니다.</p>	<p>예제 ASP.NET Core 웹 API를 생성하려면 다음을 수행하십시오.</p> <ol style="list-style-type: none"> 1. 비주얼 스튜디오 2022를 엽니다. 2. 새 프로젝트 생성을 선택합니다. 3. ASP.NET 코어 웹 API 프로젝트 템플릿을 선택하고 다음을 선택합니다. 4. 프로젝트 이름으로 DemoNetCoreWebAPI를 입력하고 Next를 선택합니다. 5. 생성을 선택하세요. 6. 프로젝트를 로컬에서 실행하려면 F5를 누릅니다. 7. 기본 WeatherForecastAPI 엔드포인트가 Swagger를 사용하여 결과를 반환하는지 확인합니다. 8. 명령 프롬프트를 열고.csproj 프로젝트 폴더로 이동한 후 다음 명령을 실행하여 새 웹 API를 리포지토리로 푸시합니다. GitHub <div style="border: 1px solid #ccc; border-radius: 10px; padding: 10px; margin-top: 10px;"> <pre>git add --all git commit -m "Initial Version"</pre> </div>	<p>앱 개발자</p>

작업	설명	필요한 기술
	<pre>git push</pre>	
<p>Dockerfile을 생성합니다.</p>	<p>Docker파일을 생성하려면 다음 중 하나를 수행합니다.</p> <ul style="list-style-type: none"> • 코드 섹션의 샘플 Dockerfile을 사용하여 Dockerfile을 수동으로 생성합니다. 요구 사항에 따라 적절한 .NET 기본 이미지를 선택합니다. .NET 및 ASP.NET Core 관련 이미지에 대한 자세한 내용은 Docker 허브를 참조하십시오. • 비주얼 스튜디오와 Docker 데스크톱을 사용하여 Docker파일을 생성합니다. 솔루션 탐색기에서 프로젝트를 마우스 오른쪽 버튼으로 클릭하고 추가 -> Docker Support를 선택합니다. 대상 OS의 경우 Linux를 선택합니다. 새 Dockerfile이 솔루션 파일(.sln)과 동일한 경로에 있는지 확인하십시오. <p>GitHub 리포지토리에 변경 내용을 푸시하려면 다음 명령을 실행합니다.</p> <pre>git add --all git commit -m "Dockerfile added" git push</pre>	<p>앱 개발자</p>

Amazon EC2 Linux 인스턴스 설정

작업	설명	필요한 기술
인프라를 설정합니다.	<p>AWS CloudFormation 템플릿을 실행하여 다음과 같은 인프라를 생성합니다.</p> <ul style="list-style-type: none"> • AWS VPC 빠른 시작을 사용하는 Virtual Private Cloud(VPC)로, 두 개의 가용 영역에 걸쳐 두 개의 퍼블릭 서브넷과 두 개의 프라이빗 서브넷이 있습니다. • AWS Systems Manager를 활성화하는 데 필요한 IAM 역할. • 프라이빗 서브넷 중 하나에 최신 SSM 에이전트가 포함된 Amazon Linux 2 데모 인스턴스가 있습니다. 이 인스턴스는 인터넷에서 직접 연결되지는 않지만 배스천 호스트 없이도 AWS Systems Manager Session Manager를 사용하여 안전하게 액세스할 수 있습니다. <p>배스천 호스트 없이 세션 관리자를 사용하여 프라이빗 Amazon EC2 인스턴스에 액세스하는 방법에 대해 자세히 알아보려면 배스천 없는 세상을 향하여 블로그 게시물을 참조하십시오.</p>	앱 개발자, AWS 관리자, AWS DevOps

작업	설명	필요한 기술
<p>Amazon EC2 Linux 인스턴스에 로그인합니다.</p>	<p>프라이빗 서브넷에서 Amazon EC2 Linux 인스턴스로 연결하려면 다음을 수행하십시오.</p> <ol style="list-style-type: none"> 1. Amazon EC2 콘솔을 엽니다. 2. 탐색 창에서 인스턴스를 선택합니다. 3. Amazon Linux 2 데모 인스턴스를 선택하고 Connect를 선택합니다. 4. 세션 관리자를 선택합니다. 5. 연결을 선택하여 새 터미널 창을 엽니다. 6. 다음 명령을 실행합니다. <div data-bbox="630 997 1027 1079" style="border: 1px solid #ccc; border-radius: 10px; padding: 5px; text-align: center; margin-top: 10px;"> <code>sudo su</code> </div>	<p>앱 개발자</p>

작업	설명	필요한 기술
Docker를 설치 및 실행합니다.	<p>Amazon EC2 Linux 인스턴스에 Docker를 설치하고 시작하려면 다음과 같이 하십시오.</p> <ol style="list-style-type: none">1. Docker를 설치하려면 다음 명령을 실행합니다. <pre>yum install -y docker</pre>2. Docker 서비스를 시작하려면 다음 명령을 실행합니다. <pre>service docker start</pre>3. Docker 설치를 확인하려면 다음 명령을 실행합니다. <pre>docker info</pre>	앱 개발자, AWS 관리자, AWS DevOps

작업	설명	필요한 기술
<p>Git을 설치하고 리포지토리를 복제합니다.</p>	<p>Amazon EC2 Linux 인스턴스에 Git을 설치하고 리포지토리를 GitHub 복제하려면 다음과 같이 하십시오.</p> <ol style="list-style-type: none"> 1. Git을 설치하려면 다음 명령을 실행합니다. <pre data-bbox="630 569 1027 646">yum install git -y</pre> 2. 리포지토리를 복제하려면 다음 명령을 실행합니다. <pre data-bbox="630 785 1027 940">git clone https://github.com/<username>/<repo-name>.git</pre> 3. Dockerfile로 이동하려면 다음 명령을 실행합니다. <pre data-bbox="630 1079 1027 1199">cd <repo-name>/DemoNetCoreWebAPI/</pre> 	<p>앱 개발자, AWS 관리자, AWS DevOps</p>

작업	설명	필요한 기술
Docker 컨테이너를 빌드하고 실행합니다.	<p>도커 이미지를 빌드하고 Amazon EC2 Linux 인스턴스 내에서 컨테이너를 실행하려면 다음과 같이 하십시오.</p> <ol style="list-style-type: none"> 도커 이미지를 생성하려면 다음 명령을 실행합니다. <pre>docker build -t aspnetcorewebapiimage -f Dockerfile .</pre> <ol style="list-style-type: none"> 모든 도커 이미지를 보려면 다음 명령을 실행합니다. <pre>docker images</pre> <ol style="list-style-type: none"> 컨테이너를 생성하고 실행하려면 다음 명령을 실행합니다. <pre>docker run -d -p 80:80 --name aspnetcorewebapicontainer aspnetcorewebapiimage</pre>	<p>앱 개발자, AWS 관리자, AWS DevOps</p>

웹 API 테스트

작업	설명	필요한 기술
curl 명령을 사용하여 웹 API를 테스트합니다.	<p>웹 API를 테스트하려면 다음 명령을 실행합니다.</p> <pre>curl -X GET "http://localhost/WeatherForecast"</pre>	<p>앱 개발자</p>

작업	설명	필요한 기술
	<pre>recast" -H "accept: text/plain"</pre> <p>API 응답을 확인합니다.</p> <p>참고: 로컬에서 실행하는 경우 Swagger에서 각 엔드포인트에 대한 curl 명령을 가져올 수 있습니다.</p>	

리소스 정리

작업	설명	필요한 기술
모든 리소스를 삭제합니다.	스택을 삭제하여 모든 리소스를 제거합니다. 이렇게 하면 사용하지 않는 서비스에 대해 요금이 부과되지 않습니다.	AWS 관리자, AWS DevOps

관련 리소스

- [PuTTY를 사용하여 Windows에서 Linux 인스턴스에 연결](#)
- [ASP.NET Core를 사용하여 웹 API를 생성하십시오](#)
- [보루 없는 세상을 향하여](#)

AWS Fargate를 사용하여 메시지 기반 워크로드를 대규모로 실행

작성자: Stan Zubarev(AWS)

환경: PoC 또는 파일럿

기술: 컨테이너 및 마이크로서비스, 메시징 및 통신, 데이터베이스

AWS 서비스: AWS Fargate, Amazon SQS, Amazon DynamoDB

요약

이 패턴은 컨테이너와 AWS Fargate를 사용하여 AWS 클라우드에서 대규모로 메시지 기반 워크로드를 실행하는 방법을 보여줍니다.

애플리케이션이 처리하는 데이터의 양이 함수 기반 서버리스 컴퓨팅 서비스의 한계를 초과할 때 컨테이너를 사용하여 데이터를 처리하는 것이 유용할 수 있습니다. 예를 들어 애플리케이션에 AWS Lambda가 제공하는 것보다 더 많은 컴퓨팅 파워나 처리 시간이 필요한 경우 Fargate를 사용하면 성능을 개선할 수 있습니다.

다음 예제 설정에서는 [AWS Cloud Development Kit \(AWS CDK\) TypeScript](#) 를 사용하여 AWS 클라우드에서 다음 리소스를 구성하고 배포합니다.

- Fargate 서비스
- Amazon Simple Queue Service(Amazon SQS) 대기열
- Amazon DynamoDB 테이블
- 아마존 CloudWatch 대시보드

Fargate 서비스는 Amazon SQS 대기열에서 메시지를 수신하여 처리한 다음 Amazon DynamoDB 테이블에 저장합니다. 대시보드를 사용하여 Fargate에서 처리되는 Amazon SQS 메시지 수와 생성한 DynamoDB 항목 수를 모니터링할 수 있습니다. CloudWatch

참고: 이 패턴의 예제 코드를 사용하여 이벤트 기반 서버리스 아키텍처에서 더 복잡한 데이터 처리 워크로드를 구축할 수도 있습니다. 자세한 내용은 [AWS Fargate를 사용하여 대규모로 이벤트 기반 및 예약된 워크로드 실행](#)을 참조하십시오.

사전 조건 및 제한 사항

사전 조건

- 활성 상태의 AWS 계정
- [AWS Command Line Interface\(AWS CLI\)](#) 최신 버전 로컬 머신에 설치 및 구성
- [Git](#) 로컬 머신에 설치 및 구성
- [AWS CDK](#) 로컬 머신에 설치 및 구성
- [이동](#) 로컬 머신에 설치 및 구성
- [Docker](#) 로컬 머신에 설치 및 구성

아키텍처

대상 기술 스택

- Amazon SQS
- AWS Fargate
- Amazon DynamoDB

대상 아키텍처

다음 다이어그램은 Fargate를 사용하여 AWS 클라우드에서 대규모로 메시지 기반 워크로드를 실행하는 예제 워크플로를 보여줍니다.

이 다이어그램은 다음 워크플로를 보여줍니다.

1. Fargate 서비스는 Amazon [SQS 긴 폴링](#)을 사용하여 Amazon SQS 대기열의 메시지를 받습니다.
2. 그러면 Fargate 서비스가 Amazon SQS 메시지를 처리하여 DynamoDB 테이블에 저장합니다.

자동화 및 규모 조정

Fargate 작업 수를 자동으로 조정하려면 Amazon Elastic Container Service (Amazon ECS) 서비스 Auto Scaling을 구성할 수 있습니다. 애플리케이션의 Amazon SQS 대기열에 표시되는 메시지 수를 기준으로 조정 정책을 구성하는 것이 가장 좋습니다.

자세한 정보는 Amazon EC2 Auto Scaling 사용 설명서의 [Amazon SQS 기반 조정](#)을 참조하십시오.

도구

서비스

- [AWS Fargate](#)를 사용하면 서버 또는 Amazon Elastic Compute Cloud(Amazon EC2) 인스턴스를 관리할 필요 없이 컨테이너를 실행할 수 있습니다. Amazon Elastic Container Service(Amazon ECS)와 함께 사용합니다.
- [Amazon Simple Queue Service\(Amazon SQS\)](#)는 분산 소프트웨어 시스템과 구성 요소를 통합하고 분리하는 데 도움이 되는 안전하고 내구성이 뛰어나며 가용성이 높은 호스팅 대기열을 제공합니다.
- [Amazon DynamoDB](#)는 빠르고 예측 가능하고 확장 가능한 성능을 제공하는 완전 관리형 NoSQL 데이터베이스 서비스입니다.
- [Amazon](#)은 AWS 리소스의 지표와 AWS에서 실행하는 애플리케이션을 실시간으로 모니터링할 수 있도록 CloudWatch 도와줍니다.

code

이 패턴의 코드는 GitHub [sqs-fargate-ddb-cdk-go](#) 리포지토리에서 사용할 수 있습니다.

에픽

AWS CDK를 사용하여 리소스 생성 및 배포

작업	설명	필요한 기술
GitHub 리포지토리를 복제합니다.	다음 명령을 실행하여 GitHub sqs-fargate-ddb-cdk-go 리포지토리를 로컬 시스템에 복제합니다. git clone https://github.com/aws-samples/sqs-fargate-ddb-cdk-go.git	앱 개발자
AWS CLI가 올바른 AWS 계정으로 구성되어 있고 AWS CDK에 필요한 권한이 있는지 확인하십시오.	AWS CLI 구성 설정이 올바른지 확인하려면 다음 Amazon Simple Storage Service(S3) ls 명령을 실행합니다. aws s3 ls	앱 개발자

작업	설명	필요한 기술
	<p>또한 이 절차를 수행하려면 AWS CDK에 AWS 계정 내에서 인프라를 프로비저닝할 수 있는 권한이 있어야 합니다. 필요한 권한을 부여하려면 AWS CLI에서 이름이 지정된 AWS 프로필을 생성하고 이를 AWS_PROFILE 환경 변수로 내보내야 합니다.</p> <p>참고: 이전에 AWS 계정에서 AWS CDK를 사용한 적이 없다면 먼저 필요한 AWS CDK 리소스를 프로비저닝해야 합니다. 자세한 내용은 AWS CDK v2 개발자 안내서의 부트스트래핑을 참조하십시오.</p>	

작업	설명	필요한 기술
AWS CDK 스택을 AWS 계정에 배포하십시오.	<ol style="list-style-type: none"> 다음 AWS CLI 명령을 실행하여 컨테이너 이미지를 생성합니다. <code>docker build -t go-fargate .</code> 다음 명령을 실행하여 AWS CDK 디렉터리를 엽니다. <code>cd cdk</code> 다음 명령을 실행하여 필요한 npm 모듈을 설치합니다. <code>npm i</code> 다음 명령을 실행하여 AWS CDK 패턴을 AWS 계정에 배포합니다. <code>cdk deploy --profile \${AWS_PROFILE}</code> 	앱 개발자

설정 테스트

작업	설명	필요한 기술
Amazon SQS 대기열로 테스트 메시지를 전송합니다.	<p>지침은 Amazon SQS 개발자 안내서의 대기열(콘솔)로 메시지 전송을 참조하십시오.</p> <p>Amazon SQS 메시지 예제 테스트</p> <pre>{ "message": "hello, Fargate"</pre>	앱 개발자

작업	설명	필요한 기술
	<pre>} </pre>	
<p>Fargate 서비스의 로그에 테스트 메시지가 나타나는지 확인합니다. CloudWatch</p>	<p>Amazon ECS 개발자 안내서의 CloudWatch 로그 보기의 지침을 따르십시오. go-service-clusterECS 클러스터의 go-fargate-service로그 그룹에 대한 로그를 검토해야 합니다.</p>	<p>앱 개발자</p>
<p>테스트 메시지가 DynamoDB 테이블에 나타나는지 확인합니다.</p>	<ol style="list-style-type: none"> 1. DynamoDB 콘솔을 엽니다. 2. 왼쪽 탐색 창에서 테이블을 선택합니다. 그런 다음 목록에서 다음 sqs-fargate-ddb-table표를 선택합니다. 3. 테이블 항목 탐색을 선택합니다. 4. 반품된 품목 목록에 테스트 메시지가 나타나는지 확인합니다. 	<p>앱 개발자</p>
<p>Fargate 서비스가 Logs로 CloudWatch 메시지를 보내고 있는지 확인하십시오.</p>	<ol style="list-style-type: none"> 1. CloudWatch 콘솔을 엽니다. 2. 왼쪽 탐색 창에서 대시보드를 선택합니다. 3. 사용자 지정 대시보드 목록에서 이름이 지정된 대시보드를 선택합니다. go-service-dashboard 4. 테스트 메시지가 로그에 나타나는지 확인합니다. <p>참고: AWS CDK는 AWS 계정에 CloudWatch 대시보드를 자동으로 생성합니다.</p>	<p>앱 개발자</p>

정리

작업	설명	필요한 기술
AWS CDK 스택을 삭제합니다.	<ol style="list-style-type: none"> 다음 명령을 실행하여 AWS CLI의 AWS CDK 디렉터리를 엽니다. <code>cd cdk</code> 다음 명령을 실행하여 AWS CDK 스택을 삭제합니다. <code>cdk destroy --profile \${AWS_PROFILE}</code> 	앱 개발자
AWS CDK 스택이 삭제되었는지 확인합니다.	<p>스택이 삭제되었는지 확인하려면 다음 명령을 실행합니다.</p> <pre>aws cloudformation list-stacks --query \"StackSummaries[?contains(StackName, 'SqsFargate')].StackStatus\" \\ --profile \${AWS_PROFILE}</pre> <p>명령 출력에 반환되는 StackStatus 값은 스택이 삭제된 경우 DELETE_COMPLETE 입니다.</p> <p>자세한 내용은 AWS CloudFormation 사용 설명서의 스택 설명 및 나열을 참조하십시오.</p>	앱 개발자

관련 리소스

- [AWS CLI 구성](#)(AWS CLI 사용 설명서 버전 2)
- [API 레퍼런스](#) (AWS CDK API 레퍼런스)
- [AWS SDK for Go v2](#) (Go 설명서)

AWS Fargate와 함께 Amazon EKS에서 Amazon EFS를 사용하여 영구 데이터 스토리지로 스테이트풀 워크로드를 실행하십시오.

작성자: 리카르도 모라이스 (AWS), 로드리고 베르사 (AWS), 루시오 페레이라 (AWS)

코드 리포지토리: Fargate 및 Amazon EFS를 사용하는 Amazon EKS	환경: PoC 또는 파일럿	기술: 컨테이너 및 마이크로서비스, 스토리지 및 백업
워크로드: 오픈 소스	AWS 서비스: Amazon EFS, Amazon EKS, AWS Fargate	

요약

이 패턴은 AWS Fargate를 사용하여 컴퓨팅 리소스를 프로비저닝함으로써 Amazon Elastic Kubernetes Service (Amazon EKS) 에서 실행되는 컨테이너의 스토리지 디바이스로서 Amazon Elastic File System (Amazon EFS) 을 활성화하기 위한 지침을 제공합니다.

이 패턴에 설명된 설정은 보안 모범 사례를 따르며 기본적으로 저장 시 보안과 전송 중 보안을 제공합니다. Amazon EFS 파일 AWS Key Management Service(AWS KMS) 키를 사용하지만, KMS 키를 생성하는 프로세스를 디스패치하는 키 별칭을 지정할 수도 있습니다.

이 패턴의 단계에 따라 proof-of-concept (PoC) 애플리케이션을 위한 네임스페이스와 Fargate 프로필을 생성하고, Kubernetes 클러스터를 Amazon EFS와 통합하는 데 사용되는 Amazon EFS 컨테이너 스토리지 인터페이스 (CSI) 드라이버를 설치하고, 스토리지 클래스를 구성하고, PoC 애플리케이션을 배포할 수 있습니다. 이러한 단계를 통해 여러 Kubernetes 워크로드 간에 공유되는 Amazon EFS 파일 시스템이 Fargate를 통해 실행됩니다. 패턴에는 이러한 단계를 자동화하는 스크립트가 함께 제공됩니다.

컨테이너식 애플리케이션에서 데이터를 지속하고 조정 작업 중 데이터 손실을 방지하려는 경우 이 패턴을 사용할 수 있습니다. 예:

- DevOps 도구 — 일반적인 시나리오는 지속적 통합 및 지속적 전달 (CI/CD) 전략을 개발하는 것입니다. 이 경우 Amazon EFS를 공유 파일 시스템으로 사용하여 CI/CD 도구의 여러 인스턴스 간에 구성을 저장하거나 CI/CD 도구의 여러 인스턴스 간에 파이프라인 단계를 위한 캐시(예: Apache Maven 리포지토리)를 저장할 수 있습니다.
- 웹 서버 - 일반적인 시나리오는 Apache를 HTTP 웹 서버로 사용하는 것입니다. Amazon EFS를 공유 파일 시스템으로 사용하여 웹 서버의 여러 인스턴스 간에 공유되는 정적 파일을 저장할 수 있습니다.

이 예제 시나리오에서는 정적 파일을 도커 이미지로 통합하는 대신 수정 사항이 파일 시스템에 직접 적용됩니다.

사전 조건 및 제한 사항

사전 조건

- 활성 상태의 AWS 계정
- 쿠버네티스 버전 1.17 이상을 사용하는 기존 Amazon EKS 클러스터 (버전 1.27까지 테스트)
- StorageClass Kubernetes를 바인딩하고 파일 시스템을 동적으로 프로비저닝하기 위한 기존 Amazon EFS 파일 시스템
- 클러스터 관리 권한
- 원하는 Amazon EKS 클러스터를 가리키도록 구성된 컨텍스트

제한 사항

- Fargate와 함께 Amazon EKS를 사용할 때는 몇 가지 제한 사항을 고려해야 합니다. 예를 들어, 권한 있는 컨테이너와 같은 DaemonSets 일부 Kubernetes 구조의 사용은 지원되지 않습니다. Fargate 제한에 대한 자세한 내용은 Amazon EKS 설명서에서 AWS [Fargate 고려](#) 사항을 참조하십시오.
- 이 패턴과 함께 제공된 코드는 Linux 또는 macOS를 실행하는 워크스테이션을 지원합니다.

제품 버전

- AWS Command Line Interface(AWS CLI) 버전 2 이상
- 아마존 EFS CSI 드라이버 버전 1.0 이상 (버전 2.4.8까지 테스트 완료)
- eksctl 버전 0.24.0 이상 (버전 0.158.0까지 테스트됨)
- jq 버전 1.6 이상
- kubectl 버전 1.17 이상 (버전 1.27까지 테스트됨)
- 쿠버네티스 버전 1.17 이상 (버전 1.27까지 테스트)

아키텍처

대상 아키텍처는 다음과 같은 인프라로 구성됩니다.

- Virtual Private Cloud(VPC).
- 두 가용 영역
- 인터넷 액세스를 제공하는 NAT 게이트웨이가 있는 퍼블릭 서브넷
- Amazon EKS 클러스터와 Amazon EFS 탑재 대상 (탑재 지점이라고도 함) 이 있는 프라이빗 서브넷
- VPC 레벨에서의 아마존 EFS

Amazon EKS 클러스터의 환경 인프라는 다음과 같습니다.

- 네임스페이스 수준에서 쿠버네티스 구조를 수용하는 AWS Fargate 프로파일
- 다음을 포함하는 쿠버네티스 네임스페이스
 - 가용 영역 전체에 분산된 두 개의 애플리케이션 포드
 - 클러스터 수준에서 영구 볼륨 (PV) 에 바인딩된 영구 볼륨 클레임 (PVC) 1개
- 네임스페이스의 PVC에 바인딩되고 클러스터 외부의 프라이빗 서브넷에 있는 Amazon EFS 탑재 대상을 가리키는 클러스터 전체 PV

도구

서비스

- [AWS 명령줄 인터페이스 \(AWS CLI\)](#) 는 명령줄에서 AWS 서비스와 상호 작용하는 데 사용할 수 있는 오픈 소스 도구입니다.
- [Amazon Elastic File System\(Amazon EFS\)](#) 은 AWS Cloud에서 공유 파일 시스템을 생성하고 구성하는 데 도움이 됩니다. 이 패턴에서는 Amazon EKS와 함께 사용할 수 있는 단순하고 확장 가능한 완전 관리형 공유 파일 시스템을 제공합니다.
- [Amazon Elastic Kubernetes Service \(Amazon EKS\)](#) 를 사용하면 자체 클러스터를 설치하거나 운영할 필요 없이 AWS에서 쿠버네티스를 실행할 수 있습니다.
- [AWS Fargate](#) 는 아마존 EKS를 위한 서버리스 컴퓨팅 엔진입니다. Kubernetes 애플리케이션을 위한 컴퓨팅 리소스를 생성하고 관리합니다.
- [AWS Key Management Service\(AWS KMS\)](#) 를 사용하면 암호화 키를 생성하고 제어하여 데이터를 보호할 수 있습니다.

기타 도구

- [Docker](#)는 운영 체제 수준의 가상화를 사용하여 컨테이너에 소프트웨어를 제공하는 서비스형 플랫폼 (PaaS) 제품 세트입니다.
- [eksctl](#)은 Amazon EKS에서 Kubernetes 클러스터를 생성하고 관리하기 위한 명령줄 유틸리티입니다.
- [kubecti](#)는 Kubernetes 클러스터에 대해 명령의 실행을 돕는 명령줄 인터페이스입니다.
- [jq](#)는 JSON을 파싱하기 위한 명령줄 도구입니다.

코드

이 패턴의 코드는 [AWS Fargate 리포지토리를 사용하는 Amazon EKS의 Amazon EFS를 사용한 GitHub 지속성 구성에서](#) 제공됩니다. 스크립트는 이 패턴의 Epic 섹션의 순서에 따라 epic01 폴더별로 [에픽별로](#) 구성됩니다. epic06

모범 사례

대상 아키텍처는 다음과 같은 서비스와 구성 요소를 포함하며 [AWS Well-Architected Framework 모범 사례](#)를 따릅니다.

- Amazon EFS는 단순하고 확장 가능하며 완벽하게 관리되는 탄력적 NFS 파일 시스템을 제공합니다. 이는 선택한 Amazon EKS 클러스터의 프라이빗 서브넷에 배포되는 포드에서 실행되는 PoC 애플리케이션의 모든 복제본 중에서 공유 파일 시스템으로 사용됩니다.
- 각 프라이빗 서브넷의 Amazon EFS 탑재 대상. 이는 클러스터의 Virtual Private Cloud(VPC) 내 가용 영역별 이중화를 제공합니다.
- Kubernetes 워크로드를 실행하는 Amazon EKS. [사전 조건 섹션에 설명된 대로 이 패턴을 사용하기 전에 Amazon EKS 클러스터를 프로비저닝해야 합니다.](#)
- AWS KMS는 Amazon EFS 파일 시스템에 저장된 콘텐츠에 대해 저장 시 암호화를 제공합니다.
- Fargate는 컨테이너의 컴퓨팅 리소스를 관리하므로 인프라 부담 대신 비즈니스 요구 사항에 집중할 수 있습니다. Fargate 프로필은 모든 프라이빗 서브넷에 대해 생성됩니다. 클러스터의 Virtual Private Cloud(VPC) 내 가용 영역별 이중화를 제공합니다.
- Kubernetes Pod: 애플리케이션의 다양한 인스턴스에서 콘텐츠를 공유, 소비 및 작성할 수 있는지 검증하기 위한 것입니다.

에픽

Amazon EKS 클러스터 프로비저닝 (선택 사항)

작업	설명	필요한 기술
Amazon EKS 클러스터를 생성합니다.	<p>클러스터를 이미 배포한 경우 다음 에픽으로 건너뛰십시오. 기존 AWS 계정에서 Amazon EKS 클러스터를 생성합니다. GitHub Repo 디렉터리에서 패턴 중 하나를 사용하여 Terraform 또는 eksctl을 사용하여 Amazon EKS 클러스터를 배포합니다. 자세한 내용은 Amazon EKS 설명서의 Amazon EKS 클러스터 생성을 참조하십시오. 참고: Terraform 패턴에는 Fargate 프로필을 Amazon EKS 클러스터에 연결하고, Amazon EFS 파일 시스템을 생성하고, Amazon EKS 클러스터에 Amazon EFS CSI 드라이버를 배포하는 방법을 보여주는 예도 있습니다.</p>	AWS 관리자, 테라폼 또는 eksctl 관리자, 쿠버네티스 관리자
익스포트 환경 변수.	<p>env.sh 스크립트를 실행합니다. 이는 다음 단계에서 필요한 정보를 제공합니다.</p> <pre data-bbox="592 1543 1031 1879">source ./scripts/env.sh Inform the AWS Account ID: <13-digit-account-id> Inform your AWS Region: <aws-Region-code> Inform your Amazon EKS Cluster Name:</pre>	AWS 시스템 관리자

작업	설명	필요한 기술
	<pre data-bbox="597 212 1024 468"><amazon-eks-cluster-name> Inform the Amazon EFS Creation Token: <self-generated-uuid></pre> <p data-bbox="597 499 1024 678">아직 확인하지 않은 경우 다음 CLI 명령을 사용하여 위에서 요청한 모든 정보를 가져올 수 있습니다.</p> <pre data-bbox="597 720 1024 919"># ACCOUNT ID aws sts get-caller-identity --query "Account" --output text</pre> <pre data-bbox="597 951 1024 1066"># REGION CODE aws configure get region</pre> <pre data-bbox="597 1098 1024 1297"># CLUSTER EKS NAME aws eks list-clusters --query "clusters" --output text</pre> <pre data-bbox="597 1329 1024 1444"># GENERATE EFS TOKEN uuidgen</pre>	

Kubernetes 네임스페이스 및 연결된 Fargate 프로파일 생성

작업	설명	필요한 기술
<p>애플리케이션 워크로드를 위한 Kubernetes 네임스페이스 및 Fargate 프로파일을 생성합니다.</p>	<p>Amazon EFS와 상호 작용하는 애플리케이션 워크로드를 수신하기 위한 네임스페이스</p>	<p>권한이 부여된 Kubernetes 사용자</p>

작업	설명	필요한 기술
	<p>를 생성합니다. create-k8s-ns-and-linked-fargate-profile.sh 스크립트 실행. 사용자 지정 네임스페이스 이름을 사용하거나 제공된 기본 네임스페이스를 사용할 수 있습니다. poc-efs-eks-fargate</p> <p>사용자 지정 애플리케이션 네임스페이스 이름 사용:</p> <pre data-bbox="594 743 1029 1104"> export \$APP_NAME SPACE=<CUSTOM_NAME> ./scripts/epic01/create-k8s-ns-and-linked-fargate-profile.sh \ -c "\$CLUSTER_NAME" -n "\$APP_NAMESPACE" </pre> <p>사용자 지정 애플리케이션 네임스페이스 이름이 없는 경우:</p> <pre data-bbox="594 1262 1029 1499"> ./scripts/epic01/create-k8s-ns-and-linked-fargate-profile.sh \ -c "\$CLUSTER_NAME" </pre> <p>\$CLUSTER_NAME 는 Amazon EKS 클러스터의 이름입니다. -n <NAMESPACE> 파라미터는 선택사항이며, 이를 알리지 않으면 생성된 기본 네임스페이스 이름이 제공됩니다.</p>	

Amazon EFS 파일 시스템 생성

작업	설명	필요한 기술
고유한 토큰을 생성합니다.	Amazon EFS에서는 멱등성 작업을 보장하기 위해 생성 토큰이 필요합니다 (동일한 생성 토큰으로 작업을 직접 호출해도 효과가 없음). 이 요구 사항을 충족하려면 사용 가능한 기술을 통해 고유한 토큰을 생성해야 합니다. 예를 들어, 범용 고유 식별자 (UUID) 를 생성하여 생성 토큰으로 사용할 수 있습니다.	AWS 시스템 관리자
Amazon EFS 파일 시스템을 생성합니다.	<p>애플리케이션 워크로드에서 읽고 쓰는 데이터 파일을 수신하기 위한 파일 시스템을 생성합니다. 암호화되거나 암호화되지 않은 파일 시스템을 생성할 수 있습니다. (이 패턴의 코드는 암호화된 시스템을 생성하여 기본적으로 저장 시 암호화를 활성화하는 것이 가장 좋습니다.) 고유한 대칭 AWS KMS 키를 사용하여 파일 시스템을 암호화할 수 있습니다. 사용자 지정 키가 지정되지 않은 경우 AWS 관리형 키가 사용됩니다.</p> <p>Amazon EFS용 고유 토큰을 생성한 후 create-efs.sh 스크립트를 사용하여 암호화되거나 암호화되지 않은 Amazon EFS 파일 시스템을 생성합니다.</p>	AWS 시스템 관리자

작업	설명	필요한 기술
	<p>KMS 키를 사용하지 않고 저장 시 암호화를 사용하는 경우:</p> <pre data-bbox="597 331 1026 604">./scripts/epic02/c reate-efs.sh \ -c "\$CLUSTER_NAME" \ -t "\$EFS_CRE ATION_TOKEN"</pre> <p>여기서 \$CLUSTER_NAME 는 Amazon EKS 클러스터의 이름 이고 \$EFS_CREATION_TOKE N 는 파일 시스템의 고유한 생 성 토큰입니다.</p> <p>유휴 시 암호화, KMS 키 사용:</p> <pre data-bbox="597 993 1026 1308">./scripts/epic02/c reate-efs.sh \ -c "\$CLUSTER_NAME" \ -t "\$EFS_CRE ATION_TOKEN" \ -k "\$KMS_KEY_ALIAS"</pre> <p>여기서 \$CLUSTER_NAME 는 Amazon EKS 클러스터 의 \$EFS_CREATION_TOKE N 이름이고, 는 파일 시스템 의 고유한 생성 토큰이며, 는 \$KMS_KEY_ALIAS KMS 키 의 별칭입니다.</p> <p>암호화를 사용하지 않는 경우:</p> <pre data-bbox="597 1791 1026 1877">./scripts/epic02/c reate-efs.sh -d \</pre>	

작업	설명	필요한 기술
	<pre data-bbox="597 205 1019 384">-c "\$CLUSTER_NAME" \ -t "\$EFS_CREATION_TOKEN"</pre> <p data-bbox="597 422 1019 695">\$CLUSTER_NAME 는 Amazon EKS 클러스터의 \$EFS_CREATION_TOKEN 이름이고, 는 파일 시스템의 고유한 생성 토큰이며, 저장 시 암호화를 -d 비활성화합니다.</p>	
보안 그룹을 생성합니다.	Amazon EKS 클러스터가 Amazon EFS 파일 시스템에 액세스할 수 있도록 보안 그룹을 생성합니다.	AWS 시스템 관리자
보안 그룹의 인바운드 규칙을 업데이트합니다.	<p data-bbox="597 961 1019 1140">다음 설정에 대한 수신 트래픽을 허용하도록 보안 그룹의 인바운드 규칙을 업데이트합니다.</p> <ul data-bbox="597 1188 1019 1423" style="list-style-type: none"> • TCP 프로토콜 — 포트 2049 • 소스 — 쿠버네티스 클러스터를 포함하는 VPC의 프라이빗 서브넷에 대한 CIDR 블록 범위 	AWS 시스템 관리자
각 프라이빗 서브넷에 탑재 대상을 추가합니다.	Kubernetes 클러스터의 각 프라이빗 서브넷에 대해 파일 시스템과 보안 그룹에 대한 마운트 대상을 생성합니다.	AWS 시스템 관리자

Kubernetes 클러스터에 Amazon EFS 구성 요소 설치

작업	설명	필요한 기술
Amazon EFS CSI 드라이버를 배포합니다.	<p>Amazon EFS CSI 드라이버를 클러스터에 배포합니다. 드라이버는 애플리케이션에서 생성한 영구 볼륨 클레임에 따라 스토리지를 프로비저닝합니다. <code>create-k8s-efs-csi-sc.sh</code> 스크립트를 실행하여 Amazon EFS CSI 드라이버와 스토리지 클래스를 클러스터에 배포합니다.</p> <pre>./scripts/epic03/create-k8s-efs-csi-sc.sh</pre> <p>이 스크립트는 <code>kubectl</code> 유틸리티를 사용하므로 컨텍스트가 구성되어 있고 원하는 Amazon EKS 클러스터를 가리키는지 확인하십시오.</p>	권한이 부여된 Kubernetes 사용자
스토리지 클래스를 배포합니다.	Amazon EFS 프로비저닝 도구 (efs.csi.aws.com) 용 클러스터에 스토리지 클래스를 배포합니다.	권한이 부여된 Kubernetes 사용자

Kubernetes 클러스터에 PoC 애플리케이션 설치

작업	설명	필요한 기술
퍼시스턴트 볼륨을 배포하십시오.	영구 볼륨을 배포하고 생성된 스토리지 클래스 및 Amazon EFS 파일 시스템의 ID에 연	권한이 부여된 Kubernetes 사용자

작업	설명	필요한 기술
	<p>결합합니다. 애플리케이션은 영구 볼륨을 사용하여 콘텐츠를 읽고 씁니다. 저장소 필드에 영구 볼륨의 크기를 원하는 대로 지정할 수 있습니다. Kubernetes는 이 필드가 필요하지만, Amazon EFS는 탄력적인 파일 시스템이기 때문에 파일 시스템 용량을 강제하지 않습니다. 암호화를 사용하거나 사용하지 않고 영구 볼륨을 배포할 수 있습니다. (Amazon EFS CSI 드라이버는 기본적으로 암호화를 활성화하는 것이 모범 사례입니다.) <code>deploy-poc-app.sh</code> 스크립트를 실행하여 영구 볼륨, 영구 볼륨 클레임 및 두 워크로드를 배포합니다.</p> <p>전송 중 암호화:</p> <pre>./scripts/epic04/deploy-poc-app.sh \ -t "\$EFS_CREATION_TOKEN"</pre> <p>파일 시스템의 고유한 생성 <code>\$EFS_CREATION_TOKEN</code> 토큰은 어디에 있습니까?</p> <p>전송 중 암호화 사용 안 함:</p> <pre>./scripts/epic04/deploy-poc-app.sh -d \ -t "\$EFS_CREATION_TOKEN"</pre>	

작업	설명	필요한 기술
	\$EFS_CREATION_TOKEN 는 파일 시스템의 고유한 생성 토큰이며 전송 중 암호화를 -d 비활성화합니다.	
애플리케이션에서 요청한 영구 볼륨 클레임을 배포하십시오.	애플리케이션에서 요청한 영구 볼륨 클레임을 배포하고 스토리지 클래스에 연결합니다. 이전에 생성한 퍼시스턴트 볼륨과 동일한 액세스 모드를 사용하십시오. 스토리지 필드에 영구 볼륨 클레임의 크기를 지정할 수 있습니다. Kubernetes 는 이 필드가 필요하지만, Amazon EFS는 탄력적인 파일 시스템이기 때문에 파일 시스템 용량을 강제하지 않습니다.	권한이 부여된 Kubernetes 사용자
워크로드 배포 1.	애플리케이션의 워크로드 1을 나타내는 포드를 배포합니다. 이 워크로드는 파일에 /data/out1.txt 콘텐츠를 씁니다.	권한이 부여된 Kubernetes 사용자
워크로드 배포 2.	애플리케이션의 워크로드 2을 나타내는 포드를 배포합니다. 이 워크로드는 파일에 콘텐츠를 씁니다/data/out2.txt .	권한이 부여된 Kubernetes 사용자

파일 시스템 지속성, 내구성 및 공유 가능성 검증

작업	설명	필요한 기술
의 상태를 확인합니다PersistentVolume .	다음 명령을 입력하여 의 상태를 확인합니다PersistentVolume .	권한이 부여된 Kubernetes 사용자

작업	설명	필요한 기술
	<pre>kubectl get pv</pre> <p>예제 출력은 추가 정보 섹션을 참조하십시오.</p>	
<p>의 상태를 확인하십시오 PersistentVolumeClaim .</p>	<p>다음 명령을 입력하여 의 상태를 확인합니다 PersistentVolumeClaim .</p> <pre>kubectl -n poc-efs-eks-fargate get pvc</pre> <p>예제 출력은 추가 정보 섹션을 참조하십시오.</p>	<p>권한이 부여된 Kubernetes 사용자</p>
<p>워크로드 1가 파일 시스템에 쓸 수 있는지 확인합니다.</p>	<p>다음 명령을 입력하여 워크로드 1이 쓰기 중인 워크로드를 /data/out1.txt 검증합니다.</p> <pre>kubectl exec -ti poc-app1 -n poc-efs-eks-fargate -- tail -f /data/out1.txt</pre> <p>결과는 다음과 비슷합니다.</p> <pre>... Thu Sep 3 15:25:07 UTC 2023 - PoC APP 1 Thu Sep 3 15:25:12 UTC 2023 - PoC APP 1 Thu Sep 3 15:25:17 UTC 2023 - PoC APP 1 ...</pre>	<p>권한이 부여된 Kubernetes 사용자</p>

작업	설명	필요한 기술
<p>워크로드 2가 파일 시스템에 쓸 수 있는지 확인합니다.</p>	<p>다음 명령을 입력하여 워크로드 2가 쓰기 중인지 확인합니다/data/out2.txt .</p> <pre data-bbox="597 394 1026 592">kubect1 -n \$APP_NAME SPACE exec -ti poc-app2 -- tail -f /data/out 2.txt</pre> <p>결과는 다음과 비슷합니다.</p> <pre data-bbox="597 705 1026 1062">... Thu Sep 3 15:26:48 UTC 2023 - PoC APP 2 Thu Sep 3 15:26:53 UTC 2023 - PoC APP 2 Thu Sep 3 15:26:58 UTC 2023 - PoC APP 2 ...</pre>	<p>권한이 부여된 Kubernetes 사용자</p>

작업	설명	필요한 기술
<p>워크로드 1이 워크로드 2에서 작성한 파일을 읽을 수 있는지 확인합니다.</p>	<p>다음 명령을 입력하여 워크로드 1이 워크로드 2에서 작성한 /data/out2.txt 파일을 읽을 수 있는지 확인합니다.</p> <pre data-bbox="592 441 1027 640">kubect1 exec -ti poc-app1 -n poc-efs-eks-fargate -- tail -n 3 /data/out2.txt</pre> <p>결과는 다음과 비슷합니다.</p> <pre data-bbox="592 745 1027 1102">... Thu Sep 3 15:26:48 UTC 2023 - PoC APP 2 Thu Sep 3 15:26:53 UTC 2023 - PoC APP 2 Thu Sep 3 15:26:58 UTC 2023 - PoC APP 2 ...</pre>	<p>권한이 부여된 Kubernetes 사용자</p>

작업	설명	필요한 기술
<p>워크로드 2이 워크로드 1에서 작성한 파일을 읽을 수 있는지 확인합니다.</p>	<p>다음 명령을 입력하여 워크로드 2가 워크로드 1에서 작성한 /data/out1.txt 파일을 읽을 수 있는지 확인합니다.</p> <pre data-bbox="597 443 1026 640">kubect1 -n \$APP_NAME SPACE exec -ti poc-app2 -- tail -n 3 /data/out 1.txt</pre> <p>결과는 다음과 비슷합니다.</p> <pre data-bbox="597 751 1026 1108">... Thu Sep 3 15:29:22 UTC 2023 - PoC APP 1 Thu Sep 3 15:29:27 UTC 2023 - PoC APP 1 Thu Sep 3 15:29:32 UTC 2023 - PoC APP 1 ...</pre>	<p>권한이 부여된 Kubernetes 사용자</p>

작업	설명	필요한 기술
<p>애플리케이션 구성 요소를 제거한 후 파일이 유지되는지 확인하십시오.</p>	<p>그런 다음 스크립트를 사용하여 애플리케이션 구성 요소 (영구 볼륨, 영구 볼륨 할당, 포드)를 /data/out1.txt 제거하고 파일이 파일 시스템에 /data/out2.txt 보존되어 있는지 확인합니다. 다음 명령을 사용하여 validate-efs-content.sh 스크립트를 실행합니다.</p> <pre data-bbox="592 730 1027 968"> ./scripts/epic05/validate-efs-content.sh \ -t "\$EFS_CREATION_TOKEN" </pre> <p>파일 시스템의 고유한 생성 \$EFS_CREATION_TOKEN 토큰은 어디에 있습니까?</p> <p>결과는 다음과 비슷합니다.</p> <pre data-bbox="592 1255 1027 1822"> pod/poc-app-validation created Waiting for pod get Running state... Waiting for pod get Running state... Waiting for pod get Running state... Results from execution of 'find /data' on validation process pod: /data /data/out2.txt </pre>	<p>권한이 부여된 Kubernetes 사용자, 시스템 관리자</p>

작업	설명	필요한 기술
	/data/out1.txt	

운영 모니터링

작업	설명	필요한 기술
애플리케이션 로그를 모니터링합니다.	2일 차 작업의 일환으로 모니터링을 CloudWatch 위해 애플리케이션 로그를 Amazon으로 전송하십시오.	AWS 시스템 관리자, 권한이 부여된 쿠버네티스 사용자
Container Insights로 Amazon EKS 및 Kubernetes 컨테이너를 모니터링합니다.	2일 차 작업의 일환으로 Amazon Container Insights를 사용하여 Amazon EKS 및 Kubernetes 시스템을 모니터링하십시오. CloudWatch 이 도구는 다양한 수준 및 차원에서 컨테이너식 애플리케이션의 지표를 수집하고 종합하며 요약합니다. 자세한 내용은 관련 리소스 섹션을 참조하십시오.	AWS 시스템 관리자, 권한이 부여된 쿠버네티스 사용자
로 Amazon EFS를 CloudWatch 모니터링하십시오.	2일 차 작업의 일환으로 Amazon CloudWatch EFS의 원시 데이터를 수집하여 읽기 쉬운 실시간에 가까운 지표로 처리하는 Amazon을 사용하여 파일 시스템을 모니터링합니다. 자세한 내용은 관련 리소스 섹션을 참조하십시오.	AWS 시스템 관리자

리소스 정리

작업	설명	필요한 기술
<p>생성된 모든 리소스를 패턴에 대해 정리합니다.</p>	<p>이 패턴을 완료한 후에는 모든 리소스를 정리하여 AWS 요금이 발생하지 않도록 하십시오. PoC 애플리케이션 사용을 마친 후 <code>clean-up-resources.sh</code> 스크립트를 실행하여 모든 리소스를 제거합니다. 다음 옵션 중 하나를 완료하십시오.</p> <p>유휴 시 암호화, KMS 키 사용:</p> <pre data-bbox="592 865 1027 1224"> ./scripts/epic06/clean-up-resources.sh \ -c "\$CLUSTER_NAME" \ -t "\$EFS_CREATION_TOKEN" \ -k "\$KMS_KEY_ALIAS" </pre> <p>여기서 <code>\$CLUSTER_NAME</code> 는 Amazon EKS 클러스터의 이름, <code>\$EFS_CREATION_TOKEN</code> 는 파일 시스템의 생성 토큰, 는 KMS 키의 <code>\$KMS_KEY_ALIAS</code> 별칭입니다.</p> <p>유휴 상태의 암호화를 사용하지 않는 경우:</p> <pre data-bbox="592 1701 1027 1833"> ./scripts/epic06/clean-up-resources.sh \ </pre>	<p>권한이 부여된 Kubernetes 사용자, 시스템 관리자</p>

작업	설명	필요한 기술
	<pre data-bbox="592 210 1031 388">-c "\$CLUSTER_NAME" \ -t "\$EFS_CREATION_TOKEN"</pre> <p data-bbox="592 420 990 651">여기서 \$CLUSTER_NAME 는 Amazon EKS 클러스터의 이름과 파일 시스템의 생성 \$EFS_CREATION_TOKEN 토큰입니다.</p>	

관련 리소스

참조

- [아마존 EKS용 AWS Fargate는 이제 아마존 EFS를 지원합니다](#) (발표)
- [AWS Fargate에서 Amazon EKS를 사용할 때 애플리케이션 로그를 캡처하는 방법](#) (블로그 게시물)
- [컨테이너 인사이트 사용](#) (Amazon CloudWatch 설명서)
- [아마존 EKS 및 쿠버네티스에서 컨테이너 인사이트 설정](#) (아마존 설명서) CloudWatch
- [Amazon EKS 및 쿠버네티스 컨테이너 인사이트 지표](#) (Amazon 설명서) CloudWatch
- [아마존을 통한 아마존 EFS 모니터링 CloudWatch](#) (아마존 EFS 설명서)

GitHub 자습서 및 예제

- [정적 프로비저닝](#)
- [전송 중 데이터 암호화](#)
- [여러 포드에서 파일 시스템에 액세스](#)
- [에서 Amazon EFS 사용 StatefulSets](#)
- [서브패스 마운팅](#)
- [Amazon EFS 액세스 포인트 사용](#)
- [테라폼용 Amazon EKS 블루프린트](#)

필수 도구

- [AWS CLI 버전 2 설치](#)
- [eksctl 설치](#)
- [kubectl 설치](#)
- [jq 설치](#)

추가 정보

다음은 명령의 예제 출력입니다. `kubectl get pv`

NAME	CAPACITY	ACCESS MODES	RECLAIM POLICY	STATUS	CLAIM
	STORAGECLASS	REASON	AGE		
poc-app-pv	1Mi	RWX	Retain	Bound	poc-efs-eks-fargate/
poc-app-pvc	efs-sc		3m56s		

다음은 `kubectl -n poc-efs-eks-fargate get pvc` 명령의 예제 출력입니다.

NAME	STATUS	VOLUME	CAPACITY	ACCESS MODES	STORAGECLASS	AGE
poc-app-pvc	Bound	poc-app-pv	1Mi	RWX	efs-sc	4m34s

패턴 더 보기

- [CAST Highlight를 사용하여 AWS 클라우드로 마이그레이션하기 위한 애플리케이션 준비 상태 평가](#)
- [AWS CDK를 사용하여 마이크로서비스용 CI/CD 파이프라인 및 Amazon ECS 클러스터 자동으로 구축](#)
- [GitHub 액션과 테라폼을 사용하여 Docker 이미지를 빌드하고 Amazon ECR로 푸시합니다.](#)
- [Blu Age로 현대화된 메인프레임 워크로드 컨테이너화](#)
- [Firelens 로그 라우터를 사용하여 Amazon ECS용 사용자 지정 로그 구문 분석기를 생성](#)
- [Amazon ECS에 Java 마이크로서비스를 위한 CI/CD 파이프라인 배포](#)
- [EC2 인스턴스 프로파일을 사용하여 AWS Cloud9에서 Amazon EKS 클러스터의 배포](#)
- [Terraform을 사용하여 컨테이너화된 Blu Age 애플리케이션을 위한 환경 배포](#)
- [Amazon의 추론 파이프라인을 사용하여 단일 엔드포인트의 ML 모델에 전처리 로직을 배포하십시오. SageMaker](#)
- [AWS 코드 서비스 및 AWS KMS 다중 리전 키를 사용하여 여러 계정 및 리전에 대한 마이크로서비스의 블루/그린 배포를 관리](#)
- [AWS CDK로 Amazon ECS Anywhere를 설정하여 온프레미스 컨테이너 애플리케이션을 관리](#)
- [오라클에서 AWS Elastic GlassFish Beanstalk로 마이그레이션](#)
- [오라클에서 Amazon WebLogic ECS의 아파치 톰캣 \(TomEE\) 으로 마이그레이션](#)
- [AWS에서 ASP.NET Web Forms 애플리케이션 현대화](#)
- [AWS 및 AWS Config를 사용하여 Amazon ECR 리포지토리에서 와일드카드 권한을 모니터링합니다. CloudFormation](#)
- [AWS CDK를 사용하여 Amazon ECS Anywhere에서 하이브리드 워크로드를 위한 CI/CD 파이프라인을 설정하고 GitLab](#)
- [Amazon S3에서 Helm v3 차트 리포지토리 설정](#)
- [인증서 관리자 및 Let's Encrypt를 사용하여 Amazon EKS의 애플리케이션에 대한 end-to-end 암호화를 설정합니다.](#)
- [Flux를 사용하여 Amazon EKS 멀티테넌트 애플리케이션 배포를 간소화합니다.](#)
- [AWS Lambda를 사용하여 육각형 아키텍처로 Python 프로젝트 구조화](#)
- [Amazon에서 사용자 지정 GPU 지원 ML 모델을 트레이닝하고 배포하기 SageMaker](#)

콘텐츠 전송

주제

- [AWS Firewall Manager와 Amazon Data Firehose를 사용하여 AWS WAF 로그를 스프링크에 전송](#)
- [Amazon을 사용하여 VPC를 통해 Amazon S3 버킷의 정적 콘텐츠를 제공합니다. CloudFront](#)
- [패턴 더 보기](#)

AWS Firewall Manager와 Amazon Data Firehose를 사용하여 AWS WAF 로그를 스플링크에 전송

작성자: Michael Friedenthal (AWS), Aman Kaur Gandhi (AWS), and JJ Johnson (AWS)

환경: PoC 또는 파일럿

기술: 콘텐츠 전송, 보안, 자격 증명, 규정 준수

워크로드: 기타 모든 워크로드

AWS 서비스: AWS Firewall Manager, Amazon Kinesis Data Firehose, AWS WAF

요약

과거에는 데이터를 Splunk로 이동하는 두 가지 방법, 즉 푸시 아키텍처 또는 풀 아키텍처가 있었습니다. 풀 아키텍처는 재시도를 통해 전송 데이터를 보장하지만 Splunk에서 데이터를 폴링하는 전용 리소스가 필요합니다. 일반적으로 풀 아키텍처는 폴링으로 인해 실시간이 아닙니다. 푸시 아키텍처는 일반적으로 지연 시간이 짧고 확장성이 뛰어나며 운영 복잡성과 비용을 줄여줍니다. 하지만 전송이 보장되는 것은 아니며 일반적으로 에이전트가 필요합니다.

Splunk와 Amazon Data Firehose의 통합은 HTTP 이벤트 수집기 (HEC) 를 통해 Splunk에 실시간 스트리밍 데이터를 제공합니다. 이 통합은 푸시 아키텍처와 풀 아키텍처의 장점을 모두 제공합니다. 즉, 재시도를 통한 데이터 전송을 보장하고 실시간에 가깝고 지연 시간이 짧고 복잡성이 낮습니다. HEC는 HTTP 또는 HTTPS를 통해 빠르고 효율적으로 데이터를 Splunk로 직접 전송합니다. HEC는 토큰 기반이므로 애플리케이션이나 지원 파일에 보안 인증 정보를 하드코딩할 필요가 없습니다.

AWS Firewall Manager 정책에서는 모든 계정의 모든 AWS WAF 웹 ACL 트래픽에 대한 로깅을 구성한 다음 Firehose 전송 스트림을 사용하여 모니터링, 시각화 및 분석을 위해 해당 로그 데이터를 Splunk로 전송할 수 있습니다. 이 솔루션에는 다음과 같은 이점이 있습니다.

- 모든 계정의 AWS WAF 웹 ACL 트래픽에 대한 중앙 관리 및 로깅
- 단일 AWS 계정과의 Splunk 통합
- 확장성
- 거의 실시간으로 로그 데이터 전송

- 서버리스 솔루션을 사용하여 비용을 최적화하므로 사용하지 않은 리소스에 대해 비용을 지불할 필요가 없습니다.

사전 조건 및 제한 사항

사전 조건

- AWS Organizations의 조직에 속해 있는 활성 AWS 계정입니다.
- Firehose로 로깅을 활성화하려면 다음 권한이 있어야 합니다.
 - iam:CreateServiceLinkedRole
 - firehose:ListDeliveryStreams
 - wafv2:PutLoggingConfiguration
- AWS WAF와 해당 웹 ACL을 구성해야 합니다. 지침은 [AWS WAF 시작하기](#)를 참고하십시오.
- AWS Firewall Manager를 설정해야 합니다. 지침은 [AWS Firewall Manager 사전 조건](#)을 참고하십시오.
- AWS WAF에 대한 Firewall Manager 보안 정책을 구성해야 합니다. 지침은 [AWS Firewall Manager AWS WAF 정책 시작하기](#)를 참고하십시오.
- Splunk는 Firehose에서 연결할 수 있는 퍼블릭 HTTP 엔드포인트로 설정해야 합니다.

제한 사항

- AWS 계정은 AWS Organizations의 단일 조직에서 관리되어야 합니다.
- 웹 ACL은 전송 스트림과 동일한 리전에 있어야 합니다. CloudFrontAmazon의 로그를 캡처하는 경우 미국 동부 (버지니아 북부) 지역에서 Firehose 전송 스트림을 생성하십시오. us-east-1
- Firehose용 Splunk 애드온은 유료 Splunk Cloud 배포, 분산 Splunk Enterprise 배포 및 단일 인스턴스 Splunk Enterprise 배포에 사용할 수 있습니다. 이 애드온은 무료 평가판 Splunk Cloud 배포에는 지원되지 않습니다.

아키텍처

대상 기술 스택

- Firewall Manager
- Firehose

- Amazon S3
- AWS WAF
- Splunk

대상 아키텍처

다음 이미지는 Firewall Manager를 사용하여 모든 AWS WAF 데이터를 중앙에서 기록하고 Kinesis Data Firehose를 통해 Splunk로 전송하는 방법을 보여줍니다.

1. AWS WAF 웹 ACL은 Firewall Manager에 방화벽 로그 데이터를 전송합니다.
2. 방화벽 관리자가 Firehose에 로그 데이터를 전송합니다.
3. Firehose 전송 스트림은 로그 데이터를 Splunk와 S3 버킷으로 전달합니다. S3 버킷은 Firehose 전송 스트림에 오류가 발생하는 경우 백업 역할을 합니다.

자동화 및 규모 조정

이 솔루션은 조직 내 모든 AWS WAF 웹 ACL을 확장하고 수용하도록 설계되었습니다. 모든 웹 ACL이 동일한 Firehose 인스턴스를 사용하도록 구성할 수 있습니다. 하지만 Firehose 인스턴스를 여러 개 설정하여 사용하려는 경우에는 그렇게 할 수 있습니다.

도구

서비스

- [AWS Firewall Manager](#)는 AWS Organizations의 계정과 애플리케이션 전체에 대 방화벽 규칙을 중앙에서 구성 및 관리할 수 있는 보안 관리 서비스입니다.
- [Amazon Data Firehose](#)를 사용하면 Splunk와 같은 지원되는 타사 서비스 공급자가 소유한 다른 AWS 서비스, 사용자 지정 HTTP 엔드포인트 및 HTTP 엔드포인트에 실시간 [스트리밍 데이터를](#) 전송할 수 있습니다.
- [Amazon Simple Storage Service\(S3\)](#)는 원하는 양의 데이터를 저장, 보호 및 검색하는 데 도움이 되는 클라우드 기반 객체 스토리지 서비스입니다.
- [AWS WAF](#)는 보호되는 웹 애플리케이션 리소스로 전달되는 HTTP 및 HTTPS 요청을 모니터링할 수 있게 해주는 웹 애플리케이션 방화벽입니다.

기타 도구

- [Splunk](#)는 로그 데이터를 모니터링, 시각화 및 분석하는 데 도움이 됩니다.

에픽

Splunk 구성

작업	설명	필요한 기술
AWS용 Splunk 앱을 설치합니다.	<ol style="list-style-type: none"> 1. Splunk 헤비 포워더에 로그인합니다. 기본 URL은 <code>http://<IP address>:8000</code> 입니다. 2. 왼쪽 탐색창에서 앱 옆에 있는 기어 버튼을 선택합니다. 3. 더 많은 앱 찾아보기를 선택합니다. 4. aws 검색합니다. 5. AWS용 Splunk 앱에 대해 설치를 선택합니다. 6. Splunk.com 로그인 보안 인증 정보를 입력하고 이용약관에 동의한 다음 로그인 및 설치를 선택합니다. 7. 완료를 선택합니다. 	보안 관리자, Splunk 관리자
AWS WAF용 애드온을 설치합니다.	이전 지침을 반복하여 Splunk용 AWS 웹 애플리케이션 방화벽 추가 기능을 설치합니다.	보안 관리자, Splunk 관리자
Firehose용 스플링크 애드온을 설치하고 구성합니다.	<ol style="list-style-type: none"> 1. Firehose용 스플링크 애드온을 설치하고 구성합니다. 설치 및 구성 과정에서 Splunk 플랫폼에 필요한 경우 HTTP 이벤트 수집기를 설정하고 인덱서에 로그 데이터를 전송할 인프라를 준비합니다. 	보안 관리자, Splunk 관리자

작업	설명	필요한 기술
	<p>Splunk 배포에 해당하는 지침을 참고합니다.</p> <ul style="list-style-type: none"> • Splunk Cloud 배포 (Splunk 설명서) • Distributed Splunk Enterprise 배포 (Splunk 설명서) • 단일 인스턴스 Splunk Enterprise 배포 (Splunk 설명서) <p>중요: Splunk 추가 기능을 설치하고 구성한 후에는 이 절차를 중단하십시오. Splunk 플랫폼으로 데이터를 전송하도록 Firehose를 구성하는 지침은 계속 진행하지 마세요.</p> <p>2. HTTP 이벤트 컬렉터 토큰과 HTTP 엔드포인트를 기록해 두십시오. 나중에 전송 스트림을 구성할 때 이 값이 필요합니다.</p>	

Firehose 전송 스트림 생성

작업	설명	필요한 기술
<p>Firehose에 Splunk 대상에 대한 액세스 권한을 부여합니다.</p>	<p>Firehose가 Splunk 대상에 액세스하도록 허용하는 액세스 정책을 구성하고 로그 데이터를 S3 버킷에 백업합니다. 자세한 내용은 Firehose에 Splunk</p>	<p>보안 관리자</p>

작업	설명	필요한 기술
	대상에 대한 액세스 권한 부여 를 참조하십시오.	

작업	설명	필요한 기술
<p>Firehose 전송 스트림을 생성하세요.</p>	<p>AWS WAF용 웹 ACL을 관리하는 동일한 계정에서 Firehose에서 전송 스트림을 생성하십시오. 전송 스트림을 생성할 때 IAM 역할을 보유하고 있어야 합니다. Firehose는 IAM 역할을 가정하고 지정된 S3 버킷에 대한 액세스 권한을 얻습니다. 지침은 전송 스트림 생성을 참고하십시오. 유념할 사항:</p> <ul style="list-style-type: none"> • 전송 스트림 이름은 aws-waf-logs- 로 시작해야 합니다. • 소스에는 Direct PUT을 선택합니다. • S3 백업 모드인 경우 모든 이벤트 백업을 선택한 다음 기존 버킷을 선택하거나 새 버킷을 생성합니다. • 목적지의 경우 Firehose 설명서에서 목적지로 Splunk 선택에 나와 있는 지침을 따르세요. Splunk 엔드포인트 및 엔드포인트 유형의 값에 대한 자세한 내용은 Splunk 설명서의 Amazon Data Firehose 구성을 참조하십시오. <p>HTTP 이벤트 컬렉터에서 구성된 각 토큰에 대해 이 프로세스를 반복합니다.</p>	<p>보안 관리자</p>

작업	설명	필요한 기술
전송 스트림을 테스트합니다.	전송 스트림을 테스트하여 제대로 구성되었는지 확인합니다. 지침은 Firehose 설명서에서 <u>Splunk를 대상으로 사용한 테스트를 참조하십시오.</u>	보안 관리자

데이터를 기록하기 위해 Firewall Manager 구성

작업	설명	필요한 기술
Firewall Manager 정책을 구성합니다.	로깅을 활성화하고 로그를 올바른 Firehose 전송 스트림으로 전달하도록 Firewall Manager 정책을 구성해야 합니다. 자세한 정보와 지침은 AWS WAF 정책의 로깅 구성을 참고하십시오.	보안 관리자

관련 리소스

AWS 리소스

- [웹 ACL 트래픽 로깅](#) (AWS WAF 설명서)
- [AWS WAF 정책을 위한 로깅 구성](#) (AWS WAF 설명서)
- [자습서: Amazon Data Firehose를 사용하여 Splunk에 VPC 흐름 로그 전송](#) (Firehose 설명서)
- [Amazon Data Firehose를 사용하여 VPC 흐름 로그를 Splunk로 푸시하려면 어떻게 해야 하나요?](#) (AWS 지식 센터)
- [Amazon Data Firehose를 사용하여 Splunk로 데이터 통합 강화](#) (AWS 블로그 게시물)

Splunk 설명서

- [Amazon Data Firehose용 스플링크 애드온](#)

Amazon을 사용하여 VPC를 통해 Amazon S3 버킷의 정적 콘텐츠를 제공합니다. CloudFront

작성자: Angel Emmanuel Hernandez Cebrian

환경: PoC 또는 파일럿

기술: 콘텐츠 전송, 네트워킹, 보안, ID, 규정 준수, 서버리스, 웹 및 모바일 앱

AWS 서비스: 아마존 CloudFront, 엘라스틱 로드 밸런싱 (ELB), AWS Lambda

요약

Amazon Web Services (AWS) 에서 호스팅되는 정적 콘텐츠를 제공하는 경우 Amazon Simple Storage Service (S3) 버킷을 오리진으로 사용하고 CloudFront Amazon을 사용하여 콘텐츠를 배포하는 것이 좋습니다. 이 솔루션에는 두 가지 주요 이점이 있습니다. 하나는 엣지 로케이션에서 정적 콘텐츠를 캐싱할 수 있다는 편리함과 CloudFront 배포를 위한 [웹 액세스 제어 목록](#) (웹 ACL) 을 정의할 수 있어 구성 및 관리 오버헤드를 최소화하면서 콘텐츠에 대한 요청을 보호할 수 있다는 점입니다.

하지만 권장되는 표준 접근 방식에는 일반적인 아키텍처 제한이 있습니다. 일부 환경에서 사용자는 가상 사설 클라우드(VPC)에 배포된 가상 방화벽 어플라이언스가 정적 콘텐츠를 포함한 모든 콘텐츠를 검사하기를 원합니다. 표준 접근 방식은 검사를 위해 VPC를 통해 트래픽을 라우팅하지 않습니다. 이 패턴은 대체 아키텍처 솔루션을 제공합니다. 여전히 CloudFront 배포를 사용하여 S3 버킷의 정적 콘텐츠를 제공하지만 Application Load Balancer를 사용하여 VPC를 통해 트래픽이 라우팅됩니다. 그러면 AWS Lambda 함수가 S3 버킷에서 콘텐츠를 검색하고 반환합니다.

사전 조건 및 제한 사항

사전 조건

- 활성 상태의 AWS 계정
- S3 버킷에 호스팅된 정적 웹 사이트 콘텐츠.

제한 사항

- 이 패턴의 리소스는 단일 AWS 리전에 있어야 하지만 다른 AWS 계정에서 프로비저닝할 수 있습니다.

- 제한은 Lambda 함수가 수신하고 전송할 수 있는 최대 요청 및 응답 크기에 각각 적용됩니다. 자세한 내용은 [대상으로서의 Lambda 함수의 제한](#)(Elastic Load Balancing 설명서)을 참고하십시오.
- 이 접근 방식을 사용할 때는 성능, 확장성, 보안 및 비용 효율성 간에 적절한 균형을 찾는 것이 중요합니다. Lambda의 높은 확장성에도 불구하고 동시 Lambda 호출 수가 최대 할당량을 초과하는 경우 일부 요청에는 병목 현상이 발생합니다. 자세한 내용은 Lambda 할당량(Lambda 설명서)를 참고하십시오. Lambda를 사용할 때는 요금도 고려해야 합니다. Lambda 호출을 최소화하려면 배포용 캐시를 올바르게 정의해야 합니다. CloudFront 자세한 내용은 [캐싱 및 가용성 최적화 \(설명서\)](#)를 참조하십시오. CloudFront

아키텍처

대상 기술 스택

- CloudFront
- Amazon Virtual Private Cloud(VPC)
- Application Load Balancer
- Lambda
- Amazon S3

대상 아키텍처

다음 이미지는 VPC를 통해 S3 버킷의 정적 콘텐츠를 제공하는 CloudFront 데 사용해야 할 때 권장되는 아키텍처를 보여줍니다.

1. 클라이언트는 S3 버킷의 특정 웹 사이트 파일을 가져오기 위해 CloudFront 배포 URL을 요청합니다.
2. CloudFront 요청을 AWS WAF로 보냅니다. AWS WAF는 배포에 적용된 웹 ACL을 사용하여 요청을 필터링합니다. CloudFront 요청이 유효한 것으로 확인되면 플로우가 계속됩니다. 요청이 유효하지 않은 것으로 확인되면 클라이언트는 403 오류를 수신합니다.
3. CloudFront 내부 캐시를 확인합니다. 수신 요청과 일치하는 유효한 키가 있는 경우 관련 값이 클라이언트에 응답으로 다시 전송됩니다. 그렇지 않은 경우에는 플로우가 계속됩니다.
4. CloudFront 요청을 지정된 애플리케이션 로드 밸런서의 URL로 전달합니다.
5. Application Load Balancer에는 Lambda 함수를 기반으로 하는 대상 그룹과 연결된 리스너가 있습니다. 애플리케이션 로드 밸런서는 Lambda 함수를 호출합니다.
6. Lambda 함수는 S3 버킷에 연결하여 GetObject 작업을 수행하고 콘텐츠를 응답으로 반환합니다.

자동화 및 규모 조정

이 접근 방식을 사용하여 정적 콘텐츠 배포를 자동화하려면 CI/CD 파이프라인을 생성하여 웹사이트를 호스팅하는 Amazon S3 버킷을 업데이트하십시오.

Lambda 함수는 서비스의 할당량 및 한도 내에서 동시 요청을 처리하도록 자동으로 확장됩니다. 자세한 내용은 [Lambda 함수 조정](#) 및 [Lambda 할당량](#)(Lambda 설명서)을 참조하세요. Application Load Balancer와 같은 다른 AWS 서비스 CloudFront 및 기능의 경우 AWS는 이러한 서비스 및 기능을 자동으로 조정합니다.

도구

- [Amazon](#)은 전 세계 데이터 센터 네트워크를 통해 웹 콘텐츠를 전송함으로써 웹 콘텐츠 배포 CloudFront 속도를 높여 지연 시간을 줄이고 성능을 개선합니다.
- [Elastic Load Balancing\(ELB\)](#)은 들어오는 애플리케이션 또는 네트워크 트래픽을 여러 대상에 분산합니다. 이 패턴에서는 Elastic Load Balancing을 통해 프로비저닝된 [Application Load Balancer](#)를 사용하여 트래픽을 Lambda 함수로 전달합니다.
- [AWS Lambda](#)는 서버를 프로비저닝하거나 관리할 필요 없이 코드를 실행하는 데 도움이 되는 컴퓨팅 서비스입니다. 필요할 때만 코드를 실행하며 자동으로 확장이 가능하므로 사용한 컴퓨팅 시간만큼만 비용을 지불합니다.
- [Amazon Simple Storage Service\(S3\)](#)는 원하는 양의 데이터를 저장, 보호 및 검색하는 데 도움이 되는 클라우드 기반 객체 스토리지 서비스입니다.
- [Amazon Virtual Private Cloud\(VPC\)](#)를 이용하면 사용자가 정의한 가상 네트워크로 AWS 리소스를 시작할 수 있습니다. 이 가상 네트워크는 사용자의 자체 데이터 센터에서 운영하는 기존 네트워크와 유사하며 AWS의 확장 가능한 인프라를 사용한다는 이점이 있습니다.

에픽

VPC를 통해 Amazon S3의 정적 콘텐츠를 제공하는 CloudFront 데 사용합니다.

작업	설명	필요한 기술
VPC를 생성합니다.	이 패턴으로 배포된 리소스(예: Application Load Balancer 및 Lambda 함수)를 호스팅하기 위한 VPC를 생성합니다. 지침	클라우드 아키텍트

작업	설명	필요한 기술
	은 VPC 생성 (Amazon VPC 설명서)을 참고하십시오.	
AWS WAF 웹 ACL을 생성합니다.	AWS WAF 웹 ACL을 생성합니다. 이 패턴 후반부에서는 이 웹 ACL을 배포에 CloudFront 적용합니다. 지침은 웹 ACL 생성 (AWS WAF 설명서)을 참고하십시오.	클라우드 아키텍트
Lambda 함수를 생성합니다.	S3 버킷에 호스팅된 정적 콘텐츠를 웹사이트로 제공하는 Lambda 함수를 생성합니다. 이 패턴의 추가 정보 섹션에 제공되는 코드를 사용합니다. 대상 S3 버킷을 식별하도록 코드를 사용자 지정합니다.	일반 AWS
Lambda 함수를 업로드합니다.	다음 명령을 입력하여 Lambda 함수 코드를 Lambda의 .zip 파일 아카이브에 업로드합니다. <pre data-bbox="597 1192 1026 1470">aws lambda update-function-code \ --function-name \ --zip-file fileb://lambda-alb-s3-website.zip</pre>	일반 AWS

작업	설명	필요한 기술
Application Load Balancer을 생성합니다.	Lambda 함수를 가리키는 인터넷 연결 애플리케이션 로드 밸런서를 생성합니다. 지침은 Lambda 함수의 대상 그룹 생성 (Elastic Load Balancing 설명서)을 참고하십시오. 가용성이 높은 구성을 하기 위해 Application Load Balancer를 생성하고 이를 다른 가용 영역의 개인 서브넷에 연결합니다.	클라우드 아키텍트

작업	설명	필요한 기술
CloudFront 배포판 만들기.	<p>생성한 Application Load Balancer를 가리키는 CloudFront 배포를 생성합니다.</p> <ol style="list-style-type: none"> 1. AWS 관리 콘솔에 로그인하고 https://console.aws.amazon.com/cloudfront/v3/home 에서 CloudFront 콘솔을 엽니다. 2. [Create Distribution]을 선택합니다. 3. Create Distribution Wizard(배포 만들기 마법사)의 첫 번째 페이지에서 Web(웹) 섹션의 Get Started(시작하기)를 선택합니다. 4. 배포에 대해 설정을 지정합니다. 자세한 내용은 배포를 생성하거나 업데이트할 때 지정하는 값을 참조하세요. 유념할 사항: <ol style="list-style-type: none"> a. Application Load Balancer를 오리진으로 설정합니다. b. 배포 설정에서 AWS WAF를 통해 적용하려는 기존 웹 ACL을 선택합니다. 자세한 내용은 AWS WAF 웹 ACL을 참조하세요. 5. 변경 내용을 저장합니다. 	클라우드 아키텍트

작업	설명	필요한 기술
	6. 배포를 CloudFront 생성한 후에는 배포의 상태 열 값이 Deployed에서 InProgressDeployed로 변경됩니다. 배포를 활성화하도록 선택한 경우 상태가 배포 완료로 전환되면 요청을 처리할 수 있습니다.	

관련 리소스

AWS 설명서

- [캐싱 및 가용성 최적화 \(문서\)](#) CloudFront
- [대상으로서의 Lambda 함수](#) (Elastic Load Balancing 설명서)
- [Lambda 할당량](#) (Lambda 설명서)

AWS 서비스 웹사이트

- [Application Load Balancer](#)
- [Lambda](#)
- [CloudFront](#)
- [Amazon S3](#)
- [AWS WAF](#)
- [Amazon VPC](#)

추가 정보

코드

다음 예제 Lambda 함수는 Node.js로 작성됩니다. 이 Lambda 함수는 웹사이트 리소스가 포함된 S3 버킷에 GetObject 작업을 수행하는 웹 서버 역할을 합니다.

```
/**
```

```
* This is an AWS Lambda function created for demonstration purposes.  
  
* It retrieves static assets from a defined Amazon S3 bucket.  
  
* To make the content available through a URL, use an Application Load Balancer with a  
Lambda integration.  
*  
* Set the S3_BUCKET environment variable in the Lambda function definition.  
*/
```

```
var AWS = require('aws-sdk');  
  
exports.handler = function(event, context, callback) {  
  
    var bucket = process.env.S3_BUCKET;  
    var key = event.path.replace('/', '');  
  
    if (key == '') {  
        key = 'index.html';  
    }  
  
    // Fetch from S3  
    var s3 = new AWS.S3();  
    return s3.getObject({Bucket: bucket, Key: key},  
        function(err, data) {  
  
            if (err) {  
                return err;  
            }  
  
            var isBase64Encoded = false;  
            var encoding = 'utf8';  
  
            if (data.ContentType.indexOf('image/') > -1) {  
                isBase64Encoded = true;  
                encoding = 'base64'  
            }  
  
            var resp = {  
                statusCode: 200,  
                headers: {  
                    'Content-Type': data.ContentType,  
                },  
            },
```

```
        body: new Buffer(data.Body).toString(encoding),
        isBase64Encoded: isBase64Encoded
    };

    callback(null, resp);
}
);
};
```

패턴 더 보기

- [Amazon CloudFront 배포판에서 액세스 로깅, HTTPS 및 TLS 버전을 확인하십시오.](#)
- [Amazon EKS 클러스터에 gRPC 기반 애플리케이션을 배포하고 Application Load Balancer를 사용하여 액세스하기](#)
- [Terraform을 사용하여 AWS WAF의 보안 자동화 솔루션 배포](#)
- [ABAC\(속성 기반 액세스 제어\)를 사용하여 퍼블릭 서브넷 보호](#)
- [Splunk를 사용하여 AWS 네트워크 방화벽 로그 및 지표 보기](#)

비용 관리

주제

- [AWS Cost Explorer를 사용하여 AWS Glue 작업에 대한 자세한 비용 및 사용 보고서를 생성합니다.](#)
- [AWS Cost Explorer를 사용하여 Amazon EMR 클러스터에 대한 자세한 비용 및 사용 보고서를 생성](#)
- [더 많은 패턴](#)

AWS Cost Explorer를 사용하여 AWS Glue 작업에 대한 자세한 비용 및 사용 보고서를 생성합니다.

작성자: Parijat Bhide(AWS), Aromal Raj Jayarajan(AWS)

환경: 프로덕션

기술: 비용 관리, 분석

AWS 서비스: AWS 과금 정보 및 비용 관리, AWS Glue

요약

이 패턴은 [사용자 정의 비용 할당 태그](#)를 구성하여 AWS Glue 데이터 통합 작업의 사용 비용을 추적하는 방법을 보여줍니다. 이 태그를 사용하여 AWS Cost Explorer에서 여러 차원의 작업에 대한 자세한 비용 및 사용 보고서를 생성할 수 있습니다. 예를 들어 팀, 프로젝트 또는 비용 센터 단위의 사용 비용을 추적할 수 있습니다.

사전 조건 및 제한 사항

사전 조건

- 활성 상태의 AWS 계정
- 사용자 정의 태그가 활성화된 하나 이상의 [AWS Glue 작업](#)

아키텍처

대상 기술 스택

- Glue
- AWS Cost Explorer

다음 다이어그램은 태그를 적용하여 AWS Glue 작업의 사용 비용을 추적하는 방법을 보여줍니다.

이 다이어그램은 다음 워크플로를 보여줍니다.

1. 데이터 엔지니어 또는 AWS 관리자는 AWS Glue 작업에 대한 사용자 정의 비용 할당 태그를 생성합니다.
2. AWS 관리자가 태그를 활성화합니다.
3. 태그는 AWS Cost Explorer에 메타데이터를 보고합니다.

도구

- [AWS Glue](#)는 완전 관리형 추출, 전환, 적재(ETL) 서비스입니다. 이를 통해 데이터 스토어와 데이터 스트림 간에 데이터를 안정적으로 분류, 정리, 보강하고 이동할 수 있습니다.
- [AWS Cost Explorer](#)는 AWS 비용과 사용량을 보고 분석하는 데 도움이 됩니다.

에픽

AWS Glue 작업을 위한 태그 생성 및 활성화

작업	설명	필요한 기술
AWS Glue 작업에 사용자 정의 비용 할당 태그를 생성합니다.	<p>기존 AWS Glue 작업에 태그를 추가하는 방법</p> <ol style="list-style-type: none"> 1. AWS Management Console에 로그인하고 AWS Management Console을 엽니다. 2. 왼쪽 탐색 창의 ETL에서 작업을 선택합니다. 3. 내 작업 섹션에서 태그를 지정할 작업의 이름을 선택합니다. 4. [작업 세부 정보(Job details)] 탭을 선택합니다. 그런 다음 고급 속성 섹션을 펼칩니다. 5. 태그에서 새 태그 추가를 선택합니다. 	데이터 엔지니어

작업	설명	필요한 기술
	<p>6. 이름에 태그 이름을 입력합니다.</p> <p>7. (선택 사항) 값에 키와 연결할 값을 입력합니다.</p> <p>8. (선택 사항) 작업에 대해 생성하려는 각 태그에 대해 5~7단계를 반복합니다.</p> <p>9. 저장을 선택합니다.</p> <p>새 AWS Glue 작업에 태그를 추가하는 방법</p> <p>1. 사용 사례 요구 사항에 따라 새 AWS Glue 작업을 생성합니다. 지침은 AWS Glue 개발자 안내서의 AWS Glue 콘솔에서 작업 사용을 참조하세요.</p> <p>2. 작업 세부 정보 설정을 구성할 때는 이 작업의 기존 AWS Glue 작업에 태그를 추가하려면 섹션의 4~9단계를 따르세요.</p> <p>참고: 자세한 내용은 AWS Glue 개발자 안내서의 AWS Glue의 AWS 태그를 참조하세요.</p>	
<p>사용자 정의 비용 할당 태그를 활성화합니다.</p>	<p>AWS 결제 사용 설명서의 사용자 정의 비용 할당 태그 활성화의 지침을 따르세요.</p>	<p>AWS 관리자</p>

AWS Glue 작업에 대한 비용 및 사용 보고서 생성

작업	설명	필요한 기술
<p>AWS Cost Explorer에서 태그 필터를 사용하여 AWS Glue 작업에 대한 비용 및 사용 보고서를 생성합니다.</p>	<ol style="list-style-type: none"> 1. AWS Management Console에 로그인하고 AWS 비용 관리 콘솔을 엽니다. 2. 왼쪽 탐색 창에서 보고서를 선택합니다. 3. 새 보고서 생성을 선택합니다. 4. 보고서 유형 선택에서 비용 및 사용량(권장)을 선택합니다. 보고서 생성을 선택합니다. 5. 필터에서 서비스를 선택합니다. 서비스 드롭다운이 나타납니다. 6. Glue 옆의 확인란을 선택합니다. 그런 다음 필터 적용을 선택합니다. 7. 필터의 경우 태그를 선택합니다. 태그 드롭다운이 나타납니다. 8. 팀을 선택하세요. 그런 다음 태그를 할당한 팀 옆의 확인란에 체크합니다. 태그를 할당하지 않은 팀은 모두 제외하세요. 그런 다음 필터 적용을 선택합니다. 9. 차트 상단에서 태그를 선택합니다. 그런 다음 보고서를 생성하려는 AWS Glue 작업의 태그를 선택합니다. 	<p>일반 AWS, AWS 관리자</p>

작업	설명	필요한 기술
	<p>10.차트 상단에서 최근 3개월 드롭다운을 선택하고 보고서에서 다루고자 하는 기간을 선택합니다. 그런 다음 월별 드롭다운을 선택하고 보고서의 라인 항목을 기간에 따라 집계할 방법을 선택합니다.</p> <p>11.다른 이름으로 저장을 선택합니다. 그런 다음 보고서 제목을 입력합니다.</p> <p>12.보고서 저장을 선택합니다.</p> <p>자세한 내용은 AWS 비용 관리 사용 설명서의 Cost Explorer를 사용한 데이터 탐색을 참조하세요.</p>	

AWS Cost Explorer를 사용하여 Amazon EMR 클러스터에 대한 자세한 비용 및 사용 보고서를 생성

작성자: Parijat Bhide(AWS), Aromal Raj Jayarajan(AWS)

환경: 프로덕션

기술: 비용 관리, 분석, 빅 데이터

AWS 서비스: AWS 과금 정보 및 비용 관리, Amazon EMR

요약

이 패턴은 [사용자 정의 비용 할당 태그](#)를 구성하여 Amazon EMR 클러스터의 사용 비용을 추적하는 방법을 보여줍니다. 이 태그를 사용하여 AWS Cost Explorer에서 여러 차원의 클러스터에 대한 자세한 비용 및 사용 보고서를 생성할 수 있습니다. 예를 들어 팀, 프로젝트 또는 비용 센터 단위의 사용 비용을 추적할 수 있습니다.

사전 조건 및 제한 사항

사전 조건

- 활성 상태의 AWS 계정.
- 사용자 정의 태그가 활성화된 하나 이상의 [EMR 클러스터](#)

아키텍처

대상 기술 스택

- Amazon EMR
- AWS Cost Explorer

대상 아키텍처

다음 다이어그램은 태그를 적용하여 특정 Amazon EMR 클러스터의 사용 비용을 추적하는 방법을 보여줍니다.

이 다이어그램은 다음 워크플로를 보여줍니다.

1. 데이터 엔지니어 또는 AWS 관리자는 Amazon EMR 클러스터에 대한 사용자 정의 비용 할당 태그를 생성합니다.
2. AWS 관리자가 태그를 활성화합니다.
3. 태그는 AWS Cost Explorer에 메타데이터를 보고합니다.

도구

도구

- [Amazon EMR](#)은 AWS에서 빅 데이터 프레임워크 실행을 간소화하여 방대한 양의 데이터를 처리하고 분석하는 관리형 클러스터 플랫폼입니다.
- [AWS Cost Explorer](#)는 AWS 비용 및 사용량을 보고 분석할 수 있도록 지원합니다.

에픽

Amazon EMR 클러스터용 태그 생성 및 활성화

작업	설명	필요한 기술
Amazon EMR 클러스터의 사용자 정의 비용 할당 태그를 생성합니다.	<p>기존 Amazon EMR 클러스터에 태그를 추가하려면</p> <p>Amazon EMR 관리 가이드의 기존 클러스터에 태그 추가 지침을 따르세요.</p> <p>새 Amazon EMR 클러스터에 태그를 추가하려면</p> <p>Amazon EMR 관리 가이드의 새 클러스터에 태그 추가 지침을 따르세요.</p> <p>Amazon EMR 클러스터를 설정하는 방법에 대한 자세한 내</p>	데이터 엔지니어

작업	설명	필요한 기술
	용은 Amazon EMR 관리 안내서의 클러스터 계획 및 구성 을 참조하세요.	
사용자 정의 비용 할당 태그를 활성화합니다.	AWS 결제 사용 설명서의 사용자 정의 비용 할당 태그 활성화 의 지침을 따르세요.	AWS 관리자

Amazon EMR 클러스터에 대한 비용 및 사용 보고서 생성

작업	설명	필요한 기술
AWS Cost Explorer에서 태그 필터를 사용하여 Amazon EMR 클러스터에 대한 비용 및 사용 보고서를 생성합니다.	<ol style="list-style-type: none"> 1. AWS Management Console에 로그인한 후 AWS 비용 관리 콘솔을 엽니다. 2. 왼쪽 탐색 창에서 보고서를 선택합니다. 3. 새 보고서 생성을 선택합니다. 4. 보고서 유형 선택에서 비용 및 사용량(권장)을 선택합니다. 보고서 생성을 선택합니다. 5. 필터에서 서비스를 선택합니다. 서비스 드롭다운이 나타납니다. 6. EMR (Elastic MapReduce) 및 EC2 인스턴스 (엘라스틱 컴퓨트 클라우드 — 컴퓨팅) 옆의 확인란을 선택합니다. 그런 다음 필터 적용을 선택합니다. 	일반 AWS, AWS 관리자

작업	설명	필요한 기술
	<p>7. 필터의 경우 태그를 선택합니다. 태그 드롭다운이 나타납니다.</p> <p>8. 팀을 선택하세요. 그런 다음 태그를 할당한 팀 옆의 확인란에 체크합니다. 태그를 할당하지 않은 팀은 모두 제외하세요. 그런 다음 필터 적용을 선택합니다.</p> <p>9. 차트 상단에서 태그를 선택합니다. 그런 다음 보고서를 생성하려는 Amazon EMR 클러스터의 태그를 선택합니다.</p> <p>10. 차트 상단에서 최근 3개월 드롭다운을 선택하고 보고서에서 다루고자 하는 기간을 선택합니다. 그런 다음 월별 드롭다운을 선택하고 보고서의 라인 항목을 기간에 따라 집계할 방법을 선택합니다.</p> <p>11. 다른 이름으로 저장을 선택합니다. 그런 다음 보고서 제목을 입력합니다.</p> <p>12. 보고서 저장을 선택합니다.</p> <p>자세한 내용은 AWS 비용 관리 사용 설명서의 Cost Explorer를 사용한 데이터 탐색을 참조하세요.</p>	

더 많은 패턴

- [aws-nuke를 사용하여 AWS 리소스 자동 삭제](#)
- [AWS를 사용하여 AppStream 2.0 리소스 생성을 자동화합니다. CloudFormation](#)
- [DynamoDB TTL을 사용하여 Amazon S3에 항목 자동으로 보관](#)
- [AWS Systems Manager Maintenance Windows를 사용하여 Amazon RDS DB 인스턴스를 자동 중지 및 시작](#)
- [Amazon RDS 및 Amazon Aurora에 대한 자세한 비용 및 사용 보고서 생성](#)
- [AWS Config 및 AWS Systems Manager로 사용하지 않는 Amazon Elastic Block Store\(Amazon EBS\) 볼륨 삭제](#)
- [Amazon DynamoDB 테이블의 스토리지 비용 추정](#)
- [온디맨드 용량에 대한 DynamoDB 테이블의 비용 추정](#)

데이터 레이크

주제

- [AWS Data Exchange에서 Amazon S3로 자동으로 데이터 수집](#)
- [AWS DataOps 개발 키트를 사용하여 Google 애널리틱스 데이터를 수집, 변환 및 분석할 수 있는 데이터 파이프라인을 구축하십시오.](#)
- [Amazon Athena를 사용하여 공유 AWS Glue 데이터 카탈로그에 대한 크로스 계정 액세스 구성](#)
- [계정 간 데이터 공유 자동화](#)
- [AWS 클라우드에서 인프라를 코드로 사용하여 서버리스 데이터 레이크 배포와 관리](#)
- [AWS IoT Greengrass를 사용하여 IoT 데이터를 Amazon S3에 직접 비용 효율적으로 수집할 수 있습니다](#)
- [WanDisco 마이그레이션기를 사용하여 하둡 데이터를 Amazon S3로 마이그레이션합니다. LiveData](#)
- [패턴 더 보기](#)

AWS Data Exchange에서 Amazon S3로 자동으로 데이터 수집

작성자: 아드난 알비(AWS)

기술: 분석, 데이터 레이크

환경: 프로덕션

AWS 서비스: 아마존 S3,
아마존 CloudWatch, AWS
Lambda, 아마존 SNS

요약

이 패턴은 AWS 데이터 거래소의 데이터를 Amazon Simple Storage Service (Amazon S3) 의 데이터 레이크로 자동 수집할 수 있는 AWS CloudFormation 템플릿을 제공합니다.

AWS Data Exchange는 AWS 클라우드에서 파일 기반 데이터 세트를 안전하고 쉽게 교환할 수 있는 서비스입니다. AWS Data Exchange 데이터 세트는 구독 기반입니다. 구독자는 공급자가 새 데이터를 게시하면 데이터 세트 수정본에도 액세스할 수 있습니다.

AWS CloudFormation 템플릿은 아마존 CloudWatch 이벤트 이벤트와 AWS Lambda 함수를 생성합니다. 이벤트는 구독한 데이터 세트의 모든 업데이트를 감시합니다. 업데이트가 있는 경우 지정한 S3 버킷으로 데이터를 복사하는 Lambda 함수를 CloudWatch 시작합니다. 데이터가 성공적으로 복사되면, Lambda는 Amazon Simple Notification Service(Amazon SNS) 알림을 전송합니다.

사전 조건 및 제한 사항

사전 조건

- 활성 상태의 AWS 계정
- AWS Data Exchange의 데이터 세트 구독

제한 사항

- AWS CloudFormation 템플릿은 AWS Data Exchange의 각 구독 데이터 세트에 대해 별도로 배포해야 합니다.

아키텍처

대상 기술 스택

- AWS Lambda
- Amazon S3
- AWS Data Exchange
- 아마존 CloudWatch
- Amazon SNS

대상 아키텍처

자동화 및 규모 조정

데이터 레이크에 수집하려는 데이터 세트에 대해 AWS CloudFormation 템플릿을 여러 번 사용할 수 있습니다.

도구

- [AWS Data Exchange](#) – AWS 고객이 AWS 클라우드에서 파일 기반 데이터 세트를 안전하고 쉽게 교환할 수 있는 서비스입니다. 구독자는 자격을 갖춘 데이터 공급자의 수백 가지 제품을 찾아 구독할 수 있습니다. 그런 다음 데이터 세트를 빠르게 다운로드하거나 Amazon S3에 복사하여 다양한 AWS 분석 및 기계 학습 서비스에서 사용할 수 있습니다. AWS 계정이 있는 사람은 누구나 AWS Data Exchange 구독자가 될 수 있습니다.
- [AWS Lambda](#) – 서버를 프로비저닝하거나 관리하지 않고도 코드를 실행할 수 있게 해주는 컴퓨팅 서비스입니다. AWS Lambda는 필요 시에만 코드를 실행하며, 일일 몇 개의 요청에서 초당 수천 개의 요청까지 자동으로 규모를 조정합니다. 사용한 컴퓨팅 시간에 대해서만 요금을 지불하며 코드가 실행되지 않을 때는 요금이 부과되지 않습니다. AWS Lambda에서는 사실상 모든 유형의 애플리케이션이나 백엔드 서비스에 대한 코드를 별도의 관리 없이 실행할 수 있습니다. AWS Lambda는 고가용성 컴퓨팅 인프라에서 코드를 실행하고 서버와 운영 체제 유지 관리, 용량 프로비저닝 및 자동 규모 조정, 코드 모니터링 및 로깅 등 모든 컴퓨팅 리소스 관리를 수행합니다.
- [Amazon S3](#) – 인터넷 스토리지입니다. Amazon S3를 사용하면 인터넷을 통해 언제 어디서든 원하는 양의 데이터를 저장하고 검색할 수 있습니다.
- [Amazon CloudWatch Events](#) — AWS 리소스의 변경 사항을 설명하는 시스템 이벤트의 스트림을 거의 실시간으로 제공합니다. 빠르게 설정할 수 있는 간단한 규칙을 사용하여 이벤트를 매칭하고 하나

이상의 대상 함수 또는 스트림으로 라우팅할 수 있습니다. CloudWatch 이벤트는 운영상의 변화가 발생하는 즉시 이를 인지합니다. 또한 환경에 응답하기 위한 메시지를 전송하고 함수를 활성화하고 변경을 수행하고 상태 정보를 기록하는 등 이러한 운영 변경 사항에 응답하고 필요에 따라 교정 조치를 취합니다. 또한 CloudWatch 이벤트를 사용하여 cron 또는 rate 표현식을 사용하여 특정 시간에 자동으로 시작되는 자동화된 작업을 예약할 수 있습니다.

- **Amazon SNS** – 애플리케이션, 최종 사용자 및 디바이스가 클라우드에서 알림을 즉시 전송 및 수신할 수 있게 해주는 웹 서비스입니다. Amazon SNS는 처리량이 높은 푸시 기반 메시징을 위한 주제(통신 채널)를 제공합니다. many-to-many 게시자는 Amazon SNS 주제를 사용하여 Amazon Simple Queue Service (Amazon SQS) 대기열, AWS Lambda 함수, HTTP/S 웹후크 등 병렬 처리를 위해 많은 구독자에게 메시지를 배포할 수 있습니다. Amazon SNS를 사용하여 모바일 푸시, SMS 및 이메일을 사용하여 최종 사용자에게 알림을 전송할 수도 있습니다.

에픽

데이터 세트 구독

작업	설명	필요한 기술
데이터 세트를 구독합니다.	AWS Data Exchange 콘솔에서 데이터 세트를 구독합니다. 지침은 '관련 리소스' 섹션의 링크를 참조하세요.	일반 AWS
데이터 세트 속성을 기록하세요.	데이터 세트의 AWS 리전, ID 및 수정 ID를 기록하세요. 다음 단계에서 AWS CloudFormation 템플릿에 이 정보가 필요합니다.	일반 AWS

AWS CloudFormation 템플릿 배포

작업	설명	필요한 기술
S3 버킷 및 폴더를 생성합니다.	Amazon S3에 이미 데이터 레이크가 있는 경우, AWS Data Exchange에서 수집할 데이터	일반 AWS

작업	설명	필요한 기술
	<p>를 저장할 폴더를 생성하세요. 테스트 목적으로 템플릿을 배포하는 경우, 새 S3 버킷을 생성하고 다음 단계를 위해 버킷 이름과 폴더 접두사를 기록하세요.</p>	
<p>AWS CloudFormation 템플릿을 배포합니다.</p>	<p>이 패턴에 첨부하여 제공된 AWS CloudFormation 템플릿을 배포하십시오. AWS 계정, 데이터 세트, S3 버킷 설정에 대응하도록 다음의 파라미터 구성: 데이터세트 AWS 리전, 데이터 세트 ID, 수정 ID, S3 버킷 이름(예: DOC-EXAMPLE-BUCKET), 폴더 접두사(예: myfolder/), SNS 알림 이메일. 데이터 세트 이름 파라미터를 원하는 이름으로 설정할 수 있습니다. 템플릿을 배포하면 Lambda 함수가 실행되어 데이터 세트에서 사용 가능한 첫 번째 데이터 세트를 자동으로 수집합니다. 그런 다음 새 데이터가 데이터 세트에 도착하면 후속 수집이 자동으로 수행됩니다.</p>	<p>일반 AWS</p>

관련 리소스

- [AWS Data Exchange의 데이터 제품 구독](#)(AWS Data Exchange 설명서)

첨부

이 문서와 관련된 추가 콘텐츠에 액세스하려면 [attachment.zip](#) 파일의 압축을 풉니다.

AWS DataOps 개발 키트를 사용하여 Google 애널리틱스 데이터를 수집, 변환 및 분석할 수 있는 데이터 파이프라인을 구축하십시오.

작성자: Anton Kukushkin (AWS) 및 Rudy Puig (AWS)

<p>코드 리포지토리: AWS DDK 예제 — 아마존 AppFlow, 아마존 아테나, AWS DataOps 개발 키트를 사용한 구글 애널리틱스 데이터 분석</p>	<p>환경: PoC 또는 파일럿</p>	<p>기술: 데이터 레이크, 분석, 인프라 DevOps</p>
<p>워크로드: 오픈 소스</p>	<p>AWS 서비스: 아마존 AppFlow, 아마존 아테나, AWS CDK, AWS Lambda, 아마존 S3</p>	

요약

이 패턴은 AWS DataOps 개발 키트 (DDK) 및 기타 AWS 서비스를 사용하여 Google 애널리틱스 데이터를 수집, 변환 및 분석하는 데이터 파이프라인을 구축하는 방법을 설명합니다. AWS DDK는 AWS에서 데이터 워크플로우와 최신 데이터 아키텍처를 구축하는 데 도움이 되는 오픈 소스 개발 프레임워크입니다. AWS DDK의 주요 목표 중 하나는 파이프라인 조정, 인프라 구축, 인프라 구축 등 일반적으로 노동 집약적인 데이터 파이프라인 작업에 소요되는 시간과 노력을 절감하는 것입니다. DevOps 이러한 노동 집약적인 작업을 AWS DDK로 오프로드하여 코드 작성 및 기타 고부가가치 활동에 집중할 수 있습니다.

사전 조건 및 제한 사항

사전 조건

- 활성 상태의 AWS 계정
- 구글 애널리틱스용 Amazon AppFlow 커넥터, [구성](#)
- [Python](#) 및 [pip](#)(Python의 패키지 관리자)
- Git, 설치 및 [구성됨](#)
- AWS Command Line Interface(AWS CLI), [설치](#) 및 [구성됨](#)
- AWS Cloud Development Kit(AWS CDK), [설치됨](#)

제품 버전

- Python 3.7 이상
- pip 9.0.3 이상

아키텍처

기술 스택

- 아마존 AppFlow
- Amazon Athena
- 아마존 CloudWatch
- 아마존 EventBridge
- Amazon Simple Storage Service(S3)
- Amazon Simple Queue Service(Amazon SQS)
- AWS DataOps 개발 키트 (DDK)
- AWS Lambda

대상 아키텍처

다음 다이어그램은 Google Analytics 데이터를 수집, 변환, 분석하는 이벤트 기반 프로세스를 보여줍니다.

이 다이어그램은 다음 워크플로를 보여줍니다.

1. 아마존 CloudWatch 예약 이벤트 규칙은 Amazon을 호출합니다. AppFlow
2. Amazon은 AppFlow 구글 애널리틱스 데이터를 S3 버킷으로 수집합니다.
3. S3 버킷에서 데이터를 수집한 후 이벤트 알림이 EventBridge 생성되고 CloudWatch Events 규칙에 의해 캡처된 다음 Amazon SQS 대기열에 추가됩니다.
4. Lambda 함수는 Amazon SQS 대기열에서 이벤트를 사용하고, 각 S3 객체를 읽고, 객체를 Apache Parquet 형식으로 변환하고, 변환된 객체를 S3 버킷에 쓴 다음, AWS Glue Data Catalog 테이블 정의를 생성 또는 업데이트합니다.
5. Athena 쿼리는 테이블에 대해 실행됩니다.

도구

도구

- [AppFlowAmazon](#)은 SaaS (서비스형 소프트웨어) 애플리케이션 간에 데이터를 안전하게 교환할 수 있는 완전 관리형 통합 서비스입니다.
- [Amazon Athena](#)는 표준 SQL을 사용하여 Amazon S3에 있는 데이터를 직접 분석할 수 있는 대화형 쿼리 서비스입니다.
- [Amazon](#)은 AWS 리소스의 지표와 AWS에서 실행하는 애플리케이션을 실시간으로 모니터링할 수 있도록 CloudWatch 도와줍니다.
- [EventBridgeAmazon](#)은 다양한 소스의 실시간 데이터와 애플리케이션을 연결하는 데 도움이 되는 서버리스 이벤트 버스 서비스입니다. AWS Lambda 함수, API 대상을 사용하는 HTTP 간접 호출 엔드 포인트 또는 다른 AWS 계정의 이벤트 버스를 예로 들 수 있습니다.
- [Amazon Simple Storage Service\(S3\)](#)는 원하는 양의 데이터를 저장, 보호 및 검색하는 데 도움이 되는 클라우드 기반 객체 스토리지 서비스입니다.
- [Amazon Simple Queue Service\(Amazon SQS\)](#)는 내구력 있고 가용성이 뛰어난 보안 호스팅 대기열을 제공하며 이를 통해 분산 소프트웨어 시스템과 구성 요소를 통합 및 분리할 수 있습니다.
- [AWS Lambda](#)는 서버를 프로비저닝하거나 관리할 필요 없이 코드를 실행하는 데 도움이 되는 컴퓨팅 서비스입니다. 필요할 때만 코드를 실행하며 자동으로 확장이 가능하므로 사용한 컴퓨팅 시간만큼만 비용을 지불합니다.
- [AWS Cloud Development Kit \(CDK\)](#) 는 코드로 클라우드 인프라를 정의하고 AWS를 통해 프로비저닝하기 위한 프레임워크입니다. CloudFormation
- [AWS DataOps Development Kit \(DDK\)](#) 는 AWS에서 데이터 워크플로와 최신 데이터 아키텍처를 구축하는 데 도움이 되는 오픈 소스 개발 프레임워크입니다.

코드

이 패턴의 코드는 GitHub [AWS DataOps 개발 키트 \(DDK\)](#) 및 [Amazon AppFlow, Amazon Athena 및 DataOps AWS 개발 키트 리포지토리를 사용한 Google 애널리틱스 데이터 분석에서](#) 사용할 수 있습니다.

에픽

환경 준비

작업	설명	필요한 기술
소스 코드를 복제합니다.	다음 명령을 실행하여 소스 코드를 빌드합니다. <pre>git clone https://github.com/aws-samples/aws-ddk-examples.git</pre>	DevOps 엔지니어
가상 환경을 생성합니다.	소스 코드 디렉터리로 이동한 후, 다음 명령을 실행하여 가상 환경을 생성합니다. <pre>cd google-analytics-data-using-appflow/python && python3 -m venv .venv</pre>	DevOps 엔지니어
종속성을 설치합니다.	가상 환경을 활성화하고 종속성을 설치하려면 다음 명령을 실행하세요. <pre>source .venv/bin/activate && pip install -r requirements.txt</pre>	DevOps 엔지니어

데이터 파이프라인을 사용하는 애플리케이션을 배포합니다.

작업	설명	필요한 기술
환경을 부트스트랩합니다.	1. AWS CLI가 사용자의 AWS 계정에 대해 유효한 보안 인	DevOps 엔지니어

작업	설명	필요한 기술
	<p>중 정보로 설정되었는지 확인하십시오. 자세한 내용은 AWS CLI 설명서에서 네임드 프로파일 사용을 참조하십시오.</p> <p>2. <code>cdk bootstrap --profile [AWS_PROFILE]</code> 명령을 실행합니다.</p>	
데이터를 배포합니다.	데이터 파이프라인을 배포하려면 <code>cdk deploy --profile [AWS_PROFILE]</code> 명령을 실행합니다.	DevOps 엔지니어

배포 테스트

작업	설명	필요한 기술
스택 상태를 확인합니다.	<ol style="list-style-type: none"> AWS CloudFormation 콘솔을 엽니다. 스택 페이지에서 스택 <code>DdkAppflowAthenaStack</code> 상태가 <code>CREATE_COMPLETE</code> 인지 확인합니다. 	DevOps 엔지니어

문제 해결

문제	Solution
AWS::AppFlow::Flow 리소스 생성 중에 배포 실패 및 Connector Profile with name ga-connection does not exist 오류가 나타납니다.	Google 애널리틱스용 Amazon AppFlow 커넥터를 만들고 이름을 지정했는지 확인하십시오 <code>ga-connection</code> .

문제	Solution
	지침은 Amazon AppFlow 설명서의 Google 애널리틱스를 참조하십시오.

관련 리소스

- [AWS DataOps 개발 키트 \(DDK\) \(GitHub\)](#)
- [AWS DDK 예제 \(\) GitHub](#)

추가 정보

AWS DDK 데이터 파이프라인은 하나 이상의 단계로 구성됩니다. 다음 코드 예제에서는 AppFlowIngestionStage를 사용하여 Google Analytics에서 데이터를 수집하고, SqsToLambdaStage를 사용하여 데이터 변환을 처리하고, AthenaSQLStage를 사용하여 Athena 쿼리를 실행합니다.

먼저 다음과 같은 코드 예제에서 볼 수 있듯이 데이터 변환 및 수집 단계가 생성됩니다.

```

appflow_stage = AppFlowIngestionStage(
    self,
    id="appflow-stage",
    flow_name=flow.flow_name,
)
sqs_lambda_stage = SqsToLambdaStage(
    self,
    id="lambda-stage",
    lambda_function_props={
        "code": Code.from_asset("./ddk_app/lambda_handlers"),
        "handler": "handler.lambda_handler",
        "layers": [
            LayerVersion.from_layer_version_arn(
                self,
                id="layer",
                layer_version_arn=f"arn:aws:lambda:
{self.region}:336392948345:layer:AWSDataWrangler-Python39:1",
            )
        ],
        "runtime": Runtime.PYTHON_3_9,
    },
)

```

```

)
# Grant lambda function S3 read & write permissions
bucket.grant_read_write(sqs_lambda_stage.function)
# Grant Glue database & table permissions
sqs_lambda_stage.function.add_to_role_policy(
    self._get_glue_db_iam_policy(database_name=database.database_name)
)
athena_stage = AthenaSQLStage(
    self,
    id="athena-sql",
    query_string=[
        (
            "SELECT year, month, day, device, count(user_count) as cnt "
            f"FROM {database.database_name}.ga_sample "
            "GROUP BY year, month, day, device "
            "ORDER BY cnt DESC "
            "LIMIT 10; "
        )
    ],
    output_location=Location(
        bucket_name=bucket.bucket_name, object_key="query-results/"
    ),
    additional_role_policy_statements=[
        self._get_glue_db_iam_policy(database_name=database.database_name)
    ],
)

```

다음으로, 다음 코드 예제에서 볼 수 있듯이 이 DataPipeline 구문은 EventBridge 규칙을 사용하여 스테이지를 “연결”하는 데 사용됩니다.

```

(
    DataPipeline(self, id="ingestion-pipeline")
        .add_stage(
            stage=appflow_stage,
            override_rule=Rule(
                self,
                "schedule-rule",
                schedule=Schedule.rate(Duration.hours(1)),
                targets=appflow_stage.targets,
            ),
        )
        .add_stage(
            stage=sqs_lambda_stage,

```

```
        # By default, AppFlowIngestionStage stage emits an event after the flow
run finishes successfully
        # Override rule below changes that behavior to call the the stage when
data lands in the bucket instead
        override_rule=Rule(
            self,
            "s3-object-created-rule",
            event_pattern=EventPattern(
                source=["aws.s3"],
                detail={
                    "bucket": {"name": [bucket.bucket_name]},
                    "object": {"key": [{"prefix": "ga-data"}]},
                },
                detail_type=["Object Created"],
            ),
            targets=sqs_lambda_stage.targets,
        ),
    )
    .add_stage(stage=athena_stage)
)
```

더 많은 코드 예제는 [Amazon AppFlow, Amazon Athena 및 AWS DataOps 개발 키트 리포지토리를 GitHub 사용한 Google 애널리틱스 데이터 분석을 참조하십시오.](#)

Amazon Athena를 사용하여 공유 AWS Glue 데이터 카탈로그에 대한 크로스 계정 액세스 구성

작성자: Denis Avdonin(AWS)

환경: 프로덕션

기술: 데이터 레이크, 분석, 빅 데이터

워크로드: 기타 모든 워크로드

AWS 서비스: Amazon Athena, AWS Glue

요약

이 패턴은 AWS Glue 데이터 카탈로그를 사용하여 Amazon Simple Storage Service (Amazon S3) 버킷에 저장된 데이터 세트의 계정 간 공유를 구성하기 위한 step-by-step 지침을 제공합니다. 여기에는 AWS Identity 및 Access Management (IAM) 정책 샘플이 포함됩니다. S3 버킷에 데이터 세트를 저장할 수 있습니다. 메타데이터는 AWS Glue 크롤러에 의해 수집되어 AWS Glue 데이터 카탈로그에 저장됩니다. S3 버킷과 AWS Glue 데이터 카탈로그는 데이터 계정이라고 하는 AWS 계정에 있습니다. 소비자 계정이라고 하는 다른 AWS 계정의 IAM 보안 주체에 대한 액세스 권한을 제공할 수 있습니다. 사용자는 Amazon Athena 서버리스 쿼리 엔진을 사용하여 소비자 계정의 데이터를 쿼리할 수 있습니다.

사전 조건 및 제한 사항

사전 조건

- 활성 상태의 [AWS 계정](#) 두 개
- AWS 계정 중 하나에 있는 [S3 버킷](#)
- [Athena 엔진 버전 2](#)
- AWS 명령줄 인터페이스 (AWS CLI)₂ 설치 및 구성 (또는 [AWS CloudShell](#) CLI 명령 실행용 AWS)

제품 버전

이 패턴은 [Athena 엔진 버전 2](#)와 [Athena 엔진 버전 3](#)에서만 작동합니다. Athena 엔진 버전 3로 업그레이드하는 것이 좋습니다. Athena 엔진 버전 1에서 Athena 엔진 버전 3로 업그레이드할 수 없는 경우

AWS 빅 데이터 블로그의 [Cross-account AWS Glue Data Catalog access with Amazon Athena](#)에 있는 접근 방식을 따르세요.

아키텍처

대상 기술 스택

- Amazon Athena
- Amazon Simple Storage Service(S3)
- Glue
- AWS Identity and Access Management(IAM)
- AWS Key Management Service(AWS KMS)

다음 다이어그램은 IAM 권한을 사용하여 AWS Glue 데이터 카탈로그를 통해 한 AWS 계정(데이터 계정)의 S3 버킷에 있는 데이터를 다른 AWS 계정(소비자 계정)과 공유하는 아키텍처를 보여줍니다.

이 다이어그램은 다음 워크플로를 보여줍니다.

1. 데이터 계정의 S3 버킷 정책은 소비자 계정의 IAM 역할과 데이터 계정의 AWS Glue 크롤러 서비스 역할에 권한을 부여합니다.
2. 데이터 계정의 AWS KMS 키 정책은 소비자 계정의 IAM 역할과 데이터 계정의 AWS Glue 크롤러 서비스 역할에 권한을 부여합니다.
3. 데이터 계정의 AWS Glue 크롤러는 S3 버킷에 저장된 데이터의 스키마를 검색합니다.
4. 데이터 계정의 AWS Glue 데이터 카탈로그 리소스 정책은 소비자 계정의 IAM 역할에 대한 액세스 권한을 부여합니다.
5. 사용자는 AWS CLI 명령을 사용하여 소비자 계정에서 명명된 카탈로그 참조를 생성합니다.
6. IAM 정책은 소비자 계정의 IAM 역할에 데이터 계정의 리소스에 대한 액세스 권한을 부여합니다. IAM 역할의 신뢰 정책을 통해 소비자 계정의 사용자는 IAM 역할을 수입할 수 있습니다.
7. 소비자 계정의 사용자는 IAM 역할을 수입하고 SQL 쿼리를 사용하여 데이터 카탈로그의 객체에 액세스합니다.
8. Athena 서버리스 엔진은 SQL 쿼리를 실행합니다.

참고: [IAM 모범 사례에서는 IAM 역할에 권한을 부여하고 ID 페더레이션을 사용하는 것이 좋습니다.](#)

도구

- [Amazon Athena](#)는 표준 SQL을 사용하여 Amazon S3에 있는 데이터를 직접 분석할 수 있는 대화형 쿼리 서비스입니다.
- [Amazon Simple Storage Service\(S3\)](#)는 원하는 양의 데이터를 저장, 보호 및 검색할 수 있는 클라우드 기반 객체 스토리지 서비스입니다.
- [AWS Glue](#)는 완전 관리형 추출, 전환, 적재(ETL) 서비스입니다. 이를 통해 데이터 스토어와 데이터 스트림 간에 데이터를 안정적으로 분류, 정리, 보강하고 이동할 수 있습니다.
- [AWS Identity and Access Management\(IAM\)](#)를 이용하면 사용자에 대해 인증 및 권한 부여를 제어함으로써 AWS 리소스에 대한 액세스를 안전하게 관리할 수 있습니다.
- [AWS Key Management Service\(AWS KMS\)](#)를 이용하면 암호화 키를 생성하고 제어하여 데이터를 보호할 수 있습니다.

에픽

데이터 계정에서 권한 설정

작업	설명	필요한 기술
S3 버킷에 있는 데이터에 대한 액세스 권한을 부여합니다.	<p>다음 템플릿을 기반으로 S3 버킷 정책을 생성하고 데이터가 저장된 버킷에 정책을 할당합니다.</p> <pre> { "Version": "2012-10-17", "Statement": [{ "Effect": "Allow", "Principal": { "AWS": ["arn:aws:iam::<consumer account id>:role/<role name>", </pre>	클라우드 관리자

작업	설명	필요한 기술
	<pre> "arn:aws:iam::<data account id>:role/ service-role/AWSGl ueServiceRole-data- bucket-crawler"] }, "Action": "s3:GetObject", "Resource": "arn:aws:s3:::data- bucket/*" }, { "Effect": "Allow", "Principa l": { "AWS": ["arn:aws:iam::<consumer account id>:role/ <role name>", "arn:aws:iam::<data account id>:role/ service-role/AWSGl ueServiceRole-data- bucket-crawler"] }, "Action": "s3:ListBucket", "Resource": "arn:aws:s3:::data- bucket" }] } </pre>	

작업	설명	필요한 기술
	<p>버킷 정책은 소비자 계정의 IAM 역할과 데이터 계정의 AWS Glue 크롤러 서비스에 권한을 부여합니다.</p>	
<p>(필요한 경우) 데이터 암호화 키에 대한 액세스 권한을 부여합니다.</p>	<p>S3 버킷이 AWS KMS 키로 암호화된 경우 소비자 계정의 IAM 역할과 데이터 계정의 AWS Glue 크롤러 서비스에 키에 대한 kms:Decrypt 권한을 부여합니다.</p> <p>다음 문으로 키 정책을 업데이트합니다.</p> <pre data-bbox="597 873 1026 1789"> { "Effect": "Allow", "Principal": { "AWS": ["arn:aws:iam::<consumer account id>:role/<role name>", "arn:aws:iam::<data account id>:role/service-role/AWSGlueServiceRole-data-bucket-crawler"] }, "Action": "kms:Decrypt", "Resource": "arn:aws:kms:<region>:<data account id>:key/<key id>" }</pre>	<p>클라우드 관리자</p>

작업	설명	필요한 기술
<p>크롤러에게 데이터에 대한 액세스 권한을 부여합니다.</p>	<p>다음 IAM 정책을 크롤러의 서비스 역할에 연결합니다.</p> <pre data-bbox="597 346 1029 1339"> { "Version": "2012-10-17", "Statement": [{ "Effect": "Allow", "Action": "s3:GetObject", "Resource": "arn:aws:s3:::data- bucket/*" }, { "Effect": "Allow", "Action": "s3:ListBucket", "Resource": "arn:aws:s3:::data- bucket" }] } </pre>	<p>클라우드 관리자</p>

작업	설명	필요한 기술
<p>(필요한 경우) 크롤러에게 데이터 암호화 키에 대한 액세스 권한을 부여합니다.</p>	<p>S3 버킷이 AWS KMS 키로 암호화된 경우 다음 정책을 연결하여 크롤러의 서비스 역할에 키에 대한 kms:Decrypt 권한을 부여합니다.</p> <pre data-bbox="594 491 1026 886"> { "Effect": "Allow", "Action": "kms:Decrypt", "Resource": "arn:aws:kms:<region>:<data account id>:key/<key id>" } </pre>	<p>클라우드 관리자</p>

작업	설명	필요한 기술
<p>소비자 계정의 IAM 역할에 데이터 카탈로그에 대한 크롤러 액세스 권한을 부여합니다.</p>	<ol style="list-style-type: none"> 1. AWS Management Console에 로그인하여 AWS Glue 콘솔을 엽니다. 2. 탐색 창의 데이터 카탈로그에서 설정을 선택합니다. 3. 권한 섹션에 다음 문을 추가한 다음 저장을 선택합니다. <pre data-bbox="592 640 1031 1795"> { "Version" : "2012-10-17", "Statement" : [{ "Effect" : "Allow", "Principal" : { "AWS" : ["arn:aws:iam::<consumer account id>:role/<role name>", "arn:aws:iam::<data account id>:role/service-role/AWSGlueServiceRole-data-bucket-crawler"] }, "Action" : "glue:*", "Resource" " : ["arn:aws:glue:<reg </pre>	<p>클라우드 관리자</p>

작업	설명	필요한 기술
	<pre data-bbox="609 210 1013 781"> ion>:<data account id:>:catalog", "arn:aws:glue:<reg ion>:<data account id:>:database/*", "arn:aws:glue:<reg ion>:<data account id:>:table/*"] }] } </pre> <p data-bbox="592 819 1023 1281">이 정책은 데이터 계정의 모든 데이터베이스와 테이블에서 모든 AWS Glue 작업을 허용합니다. 소비자 보안 주체에게 필요한 권한만 부여하도록 정책을 사용자 지정할 수 있습니다. 예를 들어 데이터베이스의 특정 테이블이나 뷰에 대한 읽기 전용 액세스를 제공할 수 있습니다.</p>	

소비자 계정에서 데이터 액세스

작업	설명	필요한 기술
<p>데이터 카탈로그에 대한 명명된 참조를 생성합니다.</p>	<p>명명된 데이터 카탈로그 참조를 생성하려면 CloudShell 또는 로컬에 설치된 AWS CLI를 사용하여 다음 명령을 실행합니다.</p>	<p>클라우드 관리자</p>

작업	설명	필요한 기술
	<pre>aws athena create-da ta-catalog --name <shared catalog name> --type GLUE --paramet ers catalog-id=<data account id></pre>	

작업	설명	필요한 기술
<p>소비자 계정의 IAM 역할에 데이터 액세스 권한을 부여합니다.</p>	<p>소비자 계정의 IAM 역할에 다음 정책을 연결하여 역할에 데이터에 대한 크로스 계정 액세스 권한을 부여합니다.</p> <pre data-bbox="597 443 1029 1841"> { "Version": "2012-10-17", "Statement": [{ "Effect": "Allow", "Action": "s3:GetObject", "Resource": "arn:aws:s3:::data-bucket/*" }, { "Effect": "Allow", "Action": "s3:ListBucket", "Resource": "arn:aws:s3:::data-bucket" }, { "Effect": "Allow", "Action": "glue:*", "Resource": ["arn:aws:glue:<region>:<data account id>:catalog", "arn:aws:glue:<region>:"] }] } </pre>	<p>클라우드 관리자</p>

작업	설명	필요한 기술
	<pre data-bbox="609 210 1015 619"> ion>:<data account id:>:database/*", "arn:aws:glue:<reg ion>:<data account id>:table/*"] }] } </pre> <p data-bbox="592 661 1031 840">다음으로 다음 템플릿을 사용하여 신뢰 정책에서 IAM 역할을 수락할 수 있는 사용자를 지정합니다.</p> <pre data-bbox="609 882 1015 1669"> { "Version": "2012-10-17", "Statement": [{ "Effect": "Allow", "Principa 1": { "AWS": "arn:aws:iam::<con sumer account id>:user/ <IAM user>" }, "Action": "sts:AssumeRole" }] } </pre> <p data-bbox="592 1711 1031 1837">마지막으로 사용자가 속한 사용자 그룹에 동일한 정책을 연결하여 IAM 역할을 수임할 수</p>	

작업	설명	필요한 기술
<p>(필요한 경우) 소비자 계정의 IAM 역할에 데이터 암호화 키에 대한 액세스 권한을 부여합니다.</p>	<p>S3 버킷이 AWS KMS 키로 암호화된 경우 다음 정책을 연결하여 소비자 계정의 IAM 역할에 키에 대한 kms:Decrypt 권한을 부여합니다.</p> <pre data-bbox="592 604 1026 1003"> { "Effect": "Allow", "Action": "kms:Decrypt", "Resource": "arn:aws:kms:<region>:<data account id>:key/<key id>" }</pre>	<p>클라우드 관리자</p>
<p>소비자 계정의 IAM 역할로 전환하여 데이터에 액세스합니다.</p>	<p>데이터 소비자는 IAM 역할로 전환하여 데이터 계정의 데이터에 액세스합니다.</p>	<p>데이터 소비자</p>

작업	설명	필요한 기술
<p>데이터에 액세스합니다.</p>	<p>Athena를 사용하여 데이터를 쿼리합니다. 예를 들어 Athena 쿼리 편집기를 열고 다음 쿼리를 실행합니다.</p> <pre data-bbox="594 443 1027 642">SELECT * FROM <shared catalog name>.<database name>.<table name></pre> <p>명명된 카탈로그 참조를 사용하는 대신 Amazon 리소스 이름(ARN)으로 카탈로그를 참조할 수도 있습니다.</p> <p>참고: 쿼리 또는 뷰에서 동적 카탈로그 참조를 사용하는 경우 참조를 이스케이프된 큰따옴표(“)로 묶어야 합니다. 예:</p> <pre data-bbox="594 1115 1027 1434">SELECT * FROM \"glue:arn:aws:glue:<region>:<data account id>:catalog\".<database name>.<table name></pre> <p>자세한 내용은 Amazon Athena 사용 설명서의 AWS Glue 데이터 카탈로그에 대한 크로스 계정 액세스를 참조하세요.</p>	<p>데이터 소비자</p>

관련 리소스

- [AWS Glue 데이터 카탈로그에 대한 크로스 계정 액세스](#)(Athena 설명서)

- [\(AWS CLI\) \(create-data-catalogAWS CLI 명령 레퍼런스\)](#)
- [Cross-account AWS Glue Data Catalog access with Amazon Athena](#)(AWS 빅 데이터 블로그)
- [IAM의 보안 모범 사례](#)(IAM 설명서)

추가 정보

크로스 계정 공유 대안으로 Lake Formation 사용

AWS Lake Formation을 사용하여 계정 간에 AWS Glue 카탈로그 객체에 대한 액세스 권한을 공유할 수 있습니다. Lake Formation은 열 및 행 수준으로 세분화된 액세스 제어, 태그 기반 액세스 제어, ACID 트랜잭션에 대한 관리 테이블 및 기타 기능을 제공합니다. Lake Formation은 Athena와 잘 통합되어 있지만 이 패턴의 IAM 전용 접근 방식에 비해 추가 구성이 필요합니다. 전체 솔루션 아키텍처의 더 넓은 맥락에서 Lake Formation 또는 IAM 전용 액세스 제어를 사용할지 결정하는 것이 좋습니다. 고려 사항에는 관련된 다른 서비스와 이러한 서비스가 두 접근 방식과 통합되는 방식이 포함됩니다.

계정 간 데이터 공유 자동화

작성자: 이삼 하비비 (AWS), 루이스 아워케이드 (AWS), 마달레나 칼보 (AWS)

환경: PoC 또는 파일럿

기술: 데이터 레이크, 애널리틱스

워크로드: 기타 모든 워크로드

AWS 서비스: AWS Glue, AWS Lake Formation, AWS RAM, 아마존 아테나

요약

조직 내에 독립 사업부 (BU) 가 여러 개 있다는 것은 데이터 레이크 액세스 권한에 대한 엄격한 제어를 최우선 과제로 삼고 각 BU는 자체 데이터에만 액세스해야 한다는 것을 의미합니다. 그러나 분석 목적으로 BU의 워크로드에 다른 BU가 관심을 가질 수 있기 때문에 세분화된 권한 제어와 함께 BU 간 데이터 공유 주제에 대한 관심이 높아질 수 있습니다.

이 앱에서는 BU가 해당 데이터를 호스팅하는 AWS 계정 (S3에서 Glue로 크롤링된 데이터베이스) 에 매핑되어 있어 BU 간 데이터 공유가 AWS 계정 간 데이터 공유에 문제가 된다고 가정합니다. Lake Formation을 사용하는 외부 AWS 계정의 보안 주체와 Glue 데이터베이스의 특정 테이블을 공유하는 자동화된 방법을 제공할 예정입니다. 이 자동화를 통해 데이터 소유자는 외부 BU에 정의된 테이블에서 분석 쿼리 (예: Athena 사용) 를 실행할 수 있는 권한을 부여할 수 있습니다.

이 자동화된 솔루션을 사용하여 다음과 같은 일반적인 사용 사례를 충족할 수 있습니다.

인사 데이터 팀은 원본 AWS 계정에서 호스팅되며, 이 계정에서 급여 표를 데이터 분석팀의 대상 AWS 계정과 공유하면 Athena를 사용하여 추가 쿼리를 받게 됩니다.

사전 조건 및 제한 사항

사전 조건

이 배포에는 다음이 필요합니다.

- 이 코드에 패키징된 AWS 리소스를 배포할 수 있는 충분한 권한을 가진 두 개의 AWS 계정 (소스 계정 및 대상 계정)

- aws-cdk: 전 세계적으로 설치됨 (npm install -g aws-cdk)
- git 클라이언트
- 테이블이 포함된 크롤링된 Glue 데이터베이스가 하나 이상 있어야 합니다.
- 에픽 섹션에는 수동 Lake Formation 구성이 거의 전시되어 있지 않습니다.

제한 사항

- 이 솔루션을 사용하려면 AWS 소스 계정에서 이미 크롤링된 Glue 데이터베이스가 필요합니다.
- 이 솔루션은 부여된 권한을 취소하는 자동화된 방법을 아직 제공하지 않습니다. 소스 계정의 데이터를 대상 계정과 공유한 후에는 Lake Formation 콘솔에서 액세스 권한을 수동으로 취소해야 합니다.

아키텍처

솔루션 개요

이 CDK 코드는 아래 다이어그램에 요약된 아키텍처를 배포합니다.

여기에는 특히 다음이 포함됩니다.

소스 계정 스택:

- DynamoDb 표: 이 표에는 사용자가 업로드하는 공유 권한 정의가 들어 있습니다. DynamoDb 스트림이 활성화되었으며 테이블에 추가된 각 공유 권한 항목에 대해 램다를 트리거합니다.
- 램다 함수: 테이블에 대한 지정된 권한을 외부 보안 주체에 부여합니다.

대상 계정 스택:

- Resource Access Manager (RAM): Lake Formation으로부터 초대를 받았습니다. 공유 데이터에 대한 액세스 권한을 부여받으려면 초대를 수락해야 합니다.
- Amazon SQS: 소스 계정으로부터 공유 절차가 시작되었음을 알리는 메시지를 수신합니다.
- EventBridge 규칙: 이 규칙은 RAM 초대가 수락되면 트리거됩니다.
- Lambda 함수 2개: 하나는 RAM 초대를 자동으로 수락하는 SQS 대기열에 의해 트리거되고 다른 하나는 로컬 공유 데이터베이스를 생성하고 공유 리소스에 대한 리소스 링크를 생성하는 규칙에 의해

EventBridge 트리거되는 함수입니다. 이러한 리소스 링크는 Athena를 통해 추가로 쿼리할 수 있습니다.

프로세스는 다음 단계로 요약할 수 있습니다.

- 1- 사용자가 소스 계정의 DynamoDB 테이블에 공유 정의 항목을 업로드합니다.
- 2- DynamoDb 스트림은 레이크 포메이션을 사용하여 공유 정의 항목에 지정된 데이터베이스의 테이블을 대상 계정과 공유하는 소스 계정 람다를 트리거합니다. 이 공유는 대상 계정에 RAM 초대를 자동으로 보냅니다.
- 3- 또한 소스 계정 Lambda는 대상 계정의 SQS 대기열에 공유 절차의 시작을 알리는 메시지를 보냅니다.
- 4- 대상 계정에서 SQS 대기열은 수신된 RAM 초대를 수락하는 Lambda를 트리거합니다.
- 5- 초대를 수락하면 EventBridge 규칙에 따라 로컬 데이터베이스와 공유 테이블을 포함할 리소스 링크를 생성하는 람다가 트리거됩니다. 또한 이 람다는 대상 보안 주체에게 공유 데이터에 대한 권한을 부여합니다.
- 6- 교장은 Athena를 사용하여 데이터를 쿼리할 수 있습니다.

도구

코드 리포지토리

[이 패턴의 코드는 Gitlab에서 사용할 수 있습니다.](#)

모범 사례

- 앞서 언급한 것처럼 계정 내에 이미 Glue 크롤링된 데이터베이스가 있어야 합니다.
- 데이터베이스 이름과 테이블 이름은 Glue 크롤링 데이터베이스의 이름과 일치해야 합니다.
- DynamoDB에 삽입할 공유 입력 항목은 다음과 같아야 합니다.

에픽

리포지토리를 복제하고 배포를 구성합니다.

작업	설명	필요한 기술
리포지토리 복제	<p>컴퓨터의 gitlab 리포지토리를 복제하세요.</p> <pre>git clone git@ssh.g itlab.aws.dev:ihab ibi/cross-account- data-sharing.git cd cross-account-data -sharing</pre>	일반 AWS
배포를 구성하세요.	<p>지역, 사용 중인 소스/대상 계정, 대상 보안 주체 arn에 대한 정보가 포함된 resources.py 파일을 편집합니다.</p> <pre>AWS_REGION = 'eu-west- 1' AWS_SOURCE_ACCOUNT_ID = '111111111111' AWS_TARGET_ACCOUNT_ID = '222222222222' TARGET_PRINCIPAL_ARN = 'arn:aws:iam::2222 22222222:role/admin'</pre>	일반 AWS

AWS 계정을 부트스트랩하고 코드를 배포하십시오.

작업	설명	필요한 기술
소스 AWS 계정 부트스트랩	아직 배포하지 않았다면 이 CDK 애플리케이션을 배포하기	일반 AWS

작업	설명	필요한 기술
	<p>전에 AWS 환경을 부트스트랩해야 합니다.</p> <p>원본 AWS 계정의 AWS 자격 증명으로 아래 명령을 실행합니다.</p> <pre>cdk bootstrap aws://<source-account-id>/<aws-region></pre>	
소스 CDK 스택 배포	<p>이제 원본 AWS 계정이 부트스트랩되고 배포를 구성했으므로 다음 명령을 사용하여 CDK 애플리케이션을 배포할 수 있습니다.</p> <p>(/디렉터리에 cross-account-data-sharing 있는지 확인하십시오.)</p> <pre>cdk deploy SourceAccountStack</pre>	일반 AWS
대상 AWS 계정 부트스트랩	<p>아직 배포하지 않았다면 이 CDK 애플리케이션을 배포하기 전에 AWS 환경을 부트스트랩해야 합니다.</p> <p>대상 AWS 계정의 AWS 자격 증명으로 아래 명령을 실행합니다.</p> <pre>cdk bootstrap aws://<target-account-id>/<aws-region></pre>	일반 AWS

작업	설명	필요한 기술
대상 CDK 스택 배포	<p>이제 대상 AWS 계정이 부트스트랩되고 배포를 구성했으므로 다음 명령을 사용하여 CDK 애플리케이션을 배포할 수 있습니다.</p> <p>(/디렉터리에 cross-account-data-sharing 있는지 확인하십시오.)</p> <pre>cdk deploy TargetAccountStack</pre>	일반 AWS

소스 계정에 Lake Formation 설정

작업	설명	필요한 기술
소스 계정에 Lake Formation 설정	<ul style="list-style-type: none"> 소스 계정에서 Lake Formation 콘솔에 로그인하고 등록 및 수집 → 데이터 레이크 위치로 이동합니다. 데이터의 S3 위치를 등록하십시오. 권한 → 데이터 레이크 권한으로 이동합니다. 모든 IAM 권한을 취소합니다. AllowedGroup 	

교차 계정 공유를 테스트해 보세요.

작업	설명	필요한 기술
<p>소스 계정에서 대상 계정까지 테이블 공유</p>	<ul style="list-style-type: none"> 소스 계정의 콘솔에 DynamoDb 로그인하고 “permissions_table” 테이블을 찾아 이 스키마에 따라 항목을 삽입하십시오. AWS CLI를 사용할 수도 있습니다. <pre data-bbox="625 642 1029 1199"> { "share_id": "1", "table_name": "sample_data", "database_name": "database-ohio", "permissions": "DESCRIBE,SELECT", "source_acc_id": "111111111111", "target_acc_id": "222222222222" } </pre> <p>항목이 테이블에 삽입되면 전체 프로세스가 트리거되고 대상 계정에서 몇 초 안에 테이블을 쿼리할 수 있습니다.</p> <ul style="list-style-type: none"> 참고로 가능한 권한은 설명, 선택 권한입니다. 쉘표로 구분해야 합니다. 	<p>일반 AWS</p>
<p>대상 계정의 테이블을 쿼리하세요.</p>	<ul style="list-style-type: none"> 대상 계정의 콘솔에 로그인하면 Lake Formation이 이미 공유 테이블을 인식하고 있 	

작업	설명	필요한 기술
	으며 Athena를 사용하여 쿼리할 수 있습니다.	

관련 리소스

[Gitlab의 코드](#)

추가 정보

주요 사용 서비스 설명서:

[아마존 DynamoDb](#)

[AWS Lambda](#)

[AWS Lake Formation](#)

[AWS Glue](#)

[AWS Resource Access Manager](#)

[Amazon SQS](#)

AWS 클라우드에서 인프라를 코드로 사용하여 서버리스 데이터 레이크 배포와 관리

작성자: Kirankumar Chandrashekar(AWS) 및 Abdel Jaidi

환경: 프로덕션

기술: 데이터 레이크, 분석, 서버리스, DevOps

워크로드: 기타 모든 워크로드

AWS 서비스: 아마존 S3, 아마존 SQS, AWS, AWS, AWS Glue, 아마존 CloudWatch, AWS Lambda, AWS Step Functions, 아마존 DynamoDB, CloudFormation

요약

이 패턴은 [서버리스 컴퓨팅](#) 및 [코드형 인프라\(IaC\)](#)를 사용하여 Amazon Web Services(AWS) 클라우드에서 데이터 레이크를 구현하고 관리하는 방법을 설명합니다. 이 패턴은 AWS에서 개발한 [서버리스 데이터 레이크 프레임워크\(SDLF\)](#) 워크숍을 기반으로 합니다.

SDLF는 AWS 클라우드의 엔터프라이즈 데이터 레이크 전송을 가속화하고 프로덕션에 더 빠르게 배포하는 데 도움이 되는 재사용 가능한 리소스 컬렉션입니다. 모범 사례에 따라 데이터 레이크의 기본 구조를 구현하는 데 사용됩니다.

SDLF는 AWS, AWS, CodePipeline AWS 등의 AWS 서비스를 사용하여 코드 및 인프라 배포 전반에 걸쳐 지속적 통합/지속적 배포 (CI/CD) 프로세스를 구현합니다. CodeBuild CodeCommit

이 패턴은 여러 서버리스 서비스를 사용하여 데이터 레이크 관리를 단순화합니다. 여기에는 스토리지용 아마존 심플 스토리지 서비스 (Amazon S3) 와 아마존 다이내모DB, 컴퓨팅용 AWS Lambda와 AWS Glue, 오케스트레이션을 위한 아마존 이벤트, 아마존 심플 큐 서비스 (Amazon SQS), CloudWatch AWS Step Functions가 포함됩니다.

CloudFormation AWS와 AWS 코드 서비스는 IaC 계층 역할을 하여 손쉬운 운영 및 관리와 함께 재현 가능하고 빠른 배포를 제공합니다.

사전 조건 및 제한 사항

사전 조건

- 활성 상태의 AWS 계정.
- 설치 및 구성된 [Command Line Interface \(CLI\)](#).
- Git 클라이언트 설치 및 구성.
- [SDLF 워크샵](#)이 웹 브라우저 창에서 열리면 바로 사용할 수 있습니다.

아키텍처

아키텍처 다이어그램은 다음 단계를 포함하는 이벤트 중심 프로세스를 보여줍니다.

1. 파일이 원시 데이터 S3 버킷에 추가된 후에 Amazon S3 이벤트 알림이 SQS 대기열에 배치됩니다. 각 알림은 S3 버킷 이름, 객체 키 또는 타임스탬프와 같은 메타데이터를 포함하는 JSON 파일로서 전송됩니다.
2. 이 알림은 메타데이터를 기반으로 올바른 추출, 전환, 적재(ETL) 프로세스로 이벤트를 라우팅하는 Lambda 함수에서 사용됩니다. Lambda 함수는 Amazon DynamoDB 테이블에 저장된 상황별 구성을 사용할 수도 있습니다. 이 단계를 통해 데이터 레이크의 여러 애플리케이션을 분리하고 확장할 수 있습니다.
3. 이벤트는 ETL 프로세스의 첫 번째 Lambda 함수로 라우팅되며, 이 함수는 원시 데이터 영역의 데이터를 데이터 레이크의 스테이징 영역으로 변환하고 이동합니다. 첫 번째 단계는 종합 카탈로그를 업데이트하는 것입니다. 이는 데이터 레이크의 모든 파일 메타데이터를 포함하는 DynamoDB 테이블입니다. 이 테이블의 각 행에는 Amazon S3에 저장된 단일 객체에 대한 운영 메타데이터가 들어 있습니다. Lambda 함수가 동기식으로 호출되어 S3 객체에서 계산에 따른 리소스 비용이 높은 작업인 광변환(예: 파일을 한 형식에서 다른 형식으로 변환하는 작업)을 수행합니다. 스테이징 S3 버킷에 새 객체가 추가되었으므로 종합 카탈로그가 업데이트되고 메시지가 ETL에서 다음 단계를 기다리면서 SQS 대기열로 전송됩니다.
4. CloudWatch 이벤트 규칙은 5분마다 Lambda 함수를 트리거합니다. 이 함수는 메시지가 이전 ETL 단계에서 SQS 대기열로 전송되었는지 확인합니다. 메시지가 전송된 경우 Lambda 함수는 ETL 프로세스의 [Step Functions](#)에서 두 번째 함수를 시작합니다.

5. 그런 다음 배치 파일에 대규모 변환이 적용됩니다. 이러한 대규모 변환은 AWS Glue 작업에 대한 동기 호출, AWS Fargate 작업, Amazon EMR 단계 또는 Amazon 노트북과 같이 컴퓨팅 비용이 많이 드는 작업입니다. SageMaker 테이블 메타데이터는 AWS Glue 카탈로그를 업데이트하는 AWS Glue 크롤러를 사용하여 출력 파일에서 추출됩니다. 파일 메타데이터는 DynamoDB의 종합 카탈로그 테이블에도 추가됩니다. 마지막으로 [Deequ](#)를 활용하는 데이터 품질 단계도 실행됩니다.

기술 스택

- 아마존 CloudWatch 이벤트
- AWS CloudFormation
- AWS CodePipeline
- AWS CodeBuild
- AWS CodeCommit
- Amazon DynamoDB
- Glue
- Lambda
- Amazon S3
- Amazon SQS
- AWS Step Functions

도구

- [Amazon CloudWatch CloudWatch Events](#) — Events는 AWS 리소스의 변경 사항을 설명하는 시스템 이벤트의 스트림을 거의 실시간으로 제공합니다.
- [AWS CloudFormation](#) — 예측 가능하고 반복적으로 AWS 인프라 배포를 생성하고 프로비저닝할 수 있도록 CloudFormation 지원합니다.
- [AWS CodeBuild](#) — 소스 코드를 컴파일하고, 단위 테스트를 실행하고, 배포 준비가 완료된 아티팩트를 생성하는 완전관리형 빌드 CodeBuild 서비스입니다.
- [AWS CodeCommit — CodeCommit AWS에서](#) 호스팅하는 버전 제어 서비스로, 자산 (예: 소스 코드 및 바이너리 파일) 을 비공개로 저장하고 관리하는 데 사용할 수 있습니다.
- [AWS CodePipeline](#) — 소프트웨어 변경 사항을 지속적으로 릴리스하는 데 필요한 단계를 모델링, 시각화 및 자동화하는 데 사용할 수 있는 지속적 전송 CodePipeline 서비스입니다.

- [Amazon DynamoDB](#) - DynamoDB는 완전 관리형 NoSQL 데이터베이스 서비스로, 원활한 확장성과 함께 빠르고 예측 가능한 성능을 제공합니다.
- [Glue](#) - Glue는 고객이 분석할 데이터를 쉽게 준비하고 로드할 수 있는 완전관리형 ETL 서비스입니다.
- [Lambda](#) - Lambda는 서버를 프로비저닝하거나 관리하지 않고도 코드를 실행할 수 있습니다. Lambda는 필요 시에만 코드를 실행하며, 일일 몇 개의 요청에서 초당 수천 개의 요청까지 자동으로 규모를 조정합니다.
- [Amazon S3](#) - Amazon Simple Storage Service(S3)는 확장성이 뛰어난 객체 스토리지 서비스입니다. Amazon S3는 웹 사이트, 모바일 애플리케이션, 백업, 데이터 레이크 등 다양한 스토리지 솔루션에 사용할 수 있습니다.
- [Step Functions](#) — Step Functions는 Lambda 함수 및 여러 AWS 서비스를 비즈니스 크리티컬 애플리케이션으로 쉽게 시퀀싱할 수 있게 해주는 서버리스 함수 오케스트레이터입니다.
- [Amazon SQS](#) - Amazon Simple Queue Service(Amazon SQS)는 마이크로서비스, 분산 시스템 및 서버리스 애플리케이션을 분리하고 확장하는 메시지 대기열 서비스입니다.
- [Deequ](#) — Deequ는 대규모 데이터 세트에 대한 데이터 품질 지표를 계산하고, 데이터 품질 제약 조건을 정의 및 확인하며, 데이터 배포의 변경 사항을 지속적으로 파악할 수 있도록 지원하는 도구입니다.

code

SDLF의 소스 코드와 리소스는 [AWS Labs GitHub 리포지토리에서](#) 제공됩니다.

에픽

IaC를 프로비저닝하기 위한 CI/CD 파이프라인을 설정

작업	설명	필요한 기술
CI/CD 파이프라인을 설정하여 데이터 레이크의 IaC를 관리합니다.	AWS Management Console에 로그인하고 SDLF 워크숍의 초기 설정 섹션에 나와 있는 절차를 따릅니다. 그러면 데이터 레이크에 IaC를 프로비저닝하고 관리하는 CodeCommit 리포지토리, CodeBuild 환경,	DevOps 엔지니어

작업	설명	필요한 기술
	CodePipeline 파이프라인과 같은 초기 CI/CD 리소스가 생성됩니다.	

IaC의 버전 제어

작업	설명	필요한 기술
로컬 시스템의 CodeCommit 리포지토리를 복제합니다.	<p>SDLF 워크숍의 기반 배포 섹션에 나와 있는 절차를 따릅니다. 이렇게 하면 IaC를 호스팅하는 Git 리포지토리를 로컬 환경에 복제할 수 있습니다.</p> <p>자세한 내용은 설명서에서 CodeCommit 리포지토리에 연결을 참조하십시오.</p> <p>CodeCommit</p>	DevOps 엔지니어
CloudFormation 템플릿을 수정하세요.	<p>로컬 워크스테이션과 코드 편집기를 사용하여 사용 사례 또는 요구 사항에 따라 CloudFormation 템플릿을 수정하십시오. 로컬로 복제된 Git 리포지토리에 커밋합니다.</p> <p>자세한 내용은 AWS CloudFormation 설명서의 AWS CloudFormation 템플릿 사용을 참조하십시오.</p>	DevOps 엔지니어
변경 내용을 CodeCommit 저장소에 푸시합니다.	이제 인프라 코드가 버전 제어 하에 있으며 코드 베이스의 수정이 추적됩니다. 변경 내용을 CodeCommit 리포지토리에 푸시하면 인프라에 변경 내용이	DevOps 엔지니어

작업	설명	필요한 기술
	<p>CodePipeline 자동으로 적용되어 전달됩니다 CodeBuild.</p> <p>중요: 에서 CodeBuild AWS SAM CLI를 사용하는 경우 <code>sam package</code> 및 <code>sam deploy</code> 명령을 실행하십시오. CLI를 사용하는 경우 <code>aws cloudformation package</code> 및 <code>aws cloudformation deploy</code> 명령을 실행합니다.</p>	

관련 리소스

CI/CD 파이프라인을 설정하여 IaC를 프로비저닝

- [SDLF 워크숍 — 초기 설정](#)

IaC 버전 제어

- [SDLF 워크숍 — 기반 배포](#)
- [CodeCommit 리포지토리에 연결](#)
- [AWS CloudFormation 템플릿 사용](#)

기타 리소스

- [서버리스 데이터 분석 파이프라인 참조 아키텍처](#)
- [SDLF 설명서](#)

AWS IoT Greengrass를 사용하여 IoT 데이터를 Amazon S3에 직접 비용 효율적으로 수집할 수 있습니다

작성자: Sebastian Viviani(AWS), Rizwan Syed(AWS)

환경: PoC 또는 파일럿	기술: 데이터 레이크, 분석애널 리틱스, IoT	워크로드: 오픈 소스
AWS 서비스: AWS IoT Greengrass, Amazon S3, Amazon Athena		

요약

이 패턴은 AWS IoT Greengrass 버전 2 디바이스를 사용하여 Amazon Simple Storage Service(S3) 버킷으로 직접 사물 인터넷(IoT) 데이터를 비용 효율적으로 수집하는 방법을 보여줍니다. 기기는 IoT 데이터를 읽고 영구 리포지토리(즉, 로컬 디스크 또는 볼륨)에 데이터를 저장하는 사용자 지정 구성 요소를 실행합니다. 그런 다음 디바이스는 IoT 데이터를 Apache Parquet 파일로 압축하고 주기적으로 데이터를 S3 버킷에 업로드합니다.

수집하는 IoT 데이터의 양과 속도는 엣지 하드웨어 기능과 네트워크 대역폭에 의해서만 제한됩니다. Amazon Athena를 사용하면 수집된 데이터를 비용 효율적으로 분석할 수 있습니다. Athena는 [Amazon Managed Grafana](#)를 사용하여 압축된 Apache Parquet 파일 및 데이터 시각화를 지원합니다.

사전 조건 및 제한 사항

사전 조건

- 활성 상태의 AWS 계정
- [AWS IoT Greengrass 버전 2](#)에서 실행되고 센서로부터 데이터를 수집하는 [엣지 게이트웨이](#) (데이터 소스 및 데이터 수집 프로세스는 이 패턴의 범위를 벗어나지만 거의 모든 유형의 센서 데이터를 사용할 수 있습니다. 이 패턴은 데이터를 로컬에 게시하는 센서 또는 게이트웨이가 있는 로컬 [MQTT](#) 브로커를 사용합니다.)
- AWS IoT Greengrass [구성 요소](#), [역할](#) 및 [SDK 종속성](#)
- S3 버킷에 데이터를 업로드하기 위한 [스트림 관리자 구성 요소](#)
- [API를 실행하기 위한 자바용 AWS SDK](#), 또는 JavaScript 파이썬용 [AWS SDK \(Boto3\)](#)

제한 사항

- 이 패턴의 데이터는 S3 버킷에 실시간으로 업로드되지 않습니다. 지연 기간이 있으며 지연 기간을 구성할 수 있습니다. 데이터는 에지 디바이스에서 일시적으로 버퍼링된 후 기간이 만료되면 업로드됩니다.
- SDK는 Java, Node.js 및 Python에서만 이용 가능합니다.

아키텍처

대상 기술 스택

- Amazon S3
- AWS IoT Greengrass
- MQTT 브로커
- 스트림 매니저 컴포넌트

대상 아키텍처

다음 다이어그램은 IoT 센서 데이터를 수집하고 해당 데이터를 S3 버킷에 저장하도록 설계된 아키텍처를 보여줍니다.

이 다이어그램은 다음 워크플로를 보여줍니다.

1. 여러 센서(예: 온도 및 밸브) 업데이트가 로컬 MQTT 브로커에 게시됩니다.
2. 이러한 센서를 구독하는 Parquet 파일 압축기는 주제를 업데이트하고 업데이트를 수신합니다.
3. Parquet 파일 압축기는 업데이트를 로컬에 저장합니다.
4. 시간이 경과하면 저장된 파일이 Parquet 파일로 압축되고 스트림 관리자로 전달되어 지정된 S3 버킷에 업로드됩니다.
5. 스트림 관리자는 Parquet 파일을 S3 버킷에 업로드합니다.

참고: 스트림 관리자(StreamManager)는 관리되는 구성 요소입니다. Amazon S3로 데이터를 내보내는 방법에 대한 예는 AWS IoT Greengrass 설명서의 [스트림 관리자](#)를 참조하세요. 로컬 MQTT 브로커를 구성 요소로 사용하거나 [Eclipse Mosquitto](#)와 같은 다른 브로커로 사용할 수 있습니다.

도구

AWS 도구

- [Amazon Athena](#)는 표준 SQL을 사용하여 Amazon S3에 있는 데이터를 직접 분석할 수 있는 대화형 쿼리 서비스입니다.
- [Amazon Simple Storage Service\(S3\)](#)는 원하는 양의 데이터를 저장, 보호 및 검색할 수 있는 클라우드 기반 객체 스토리지 서비스입니다.
- [AWS IoT Greengrass](#)는 디바이스에서 IoT 애플리케이션을 구축, 배포 및 관리하는 데 도움이 되는 오픈 소스 IoT 엣지 런타임 및 클라우드 서비스입니다.

기타 도구

- [Apache Parquet](#)는 스토리지 및 검색을 위해 설계된 오픈 소스 열 지향 데이터 파일 형식입니다.
- [MQTT](#)(메시지 큐잉 텔레메트리 전송)는 제약이 있는 디바이스용으로 설계된 경량 메시징 프로토콜입니다.

모범 사례

업로드된 데이터에 적합한 파티션 형식 사용

S3 버킷의 루트 접두사 이름(예: "myAwesomeDataSet/" 또는 "dataFromSource")에 대한 특정 요구 사항은 없지만 데이터 세트의 용도를 쉽게 이해할 수 있도록 의미 있는 파티션과 접두사를 사용하는 것이 좋습니다.

또한 쿼리가 데이터 세트에서 최적으로 실행되도록 Amazon S3에서 올바른 파티셔닝을 사용하는 것이 좋습니다. 다음 예제에서는 각 Athena 쿼리에서 스캔되는 데이터의 양이 최적화되도록 데이터를 HIVE 형식으로 분할합니다. 이렇게 하면 성능을 개선하고 비용을 절감할 수 있습니다.

```
s3://<ingestionBucket>/<rootPrefix>/year=YY/month=MM/day=DD/  
HHMM_<suffix>.parquet
```

에픽

환경을 설정합니다.

작업	설명	필요한 기술
S3 버킷을 생성합니다.	<ol style="list-style-type: none"> 1. S3 버킷을 생성하거나 기존 버킷을 사용합니다. 2. IoT 데이터를 수집하려는 S3 버킷에 대해 의미 있는 접두사를 생성합니다(예: <code>s3://<bucket>/<prefix></code>). 3. 나중에 사용할 수 있도록 접두사를 기록합니다 	앱 개발자
S3 버킷에 IAM 권한을 부여합니다.	<p>이전에 생성한 S3 버킷 및 접두사에 대한 쓰기 액세스 권한을 사용자에게 부여하려면 AWS IoT Greengrass 역할에 다음 IAM 정책을 추가하세요.</p> <pre data-bbox="597 1178 1027 1869"> { "Version": "2012-10-17", "Statement": [{ "Sid": "S3DataUpload", "Effect": "Allow", "Action": ["s3:List*", "s3:Put*"], "Resource": [</pre>	앱 개발자

작업	설명	필요한 기술
	<pre data-bbox="609 241 1015 661"> "arn:aws:s3:::<ingestionBucket>", "arn:aws:s3:::<ingestionBucket>/<prefix>/*"] }] } } </pre> <p data-bbox="592 693 1015 871">자세한 내용은 Aurora 설명서의 Amazon S3 리소스에 액세스할 수 있는 IAM 정책 생성을 참조하세요.</p> <p data-bbox="592 913 1015 1102">다음으로 올바른 AWS 보안 주체로 쓰기 액세스를 허용하도록 S3 버킷의 리소스 정책(필요한 경우)을 업데이트하세요.</p>	

AWS IoT Greengrass 구성 요소 구축 및 배포

작업	설명	필요한 기술
<p data-bbox="105 1375 560 1459">구성 요소의 레시피를 업데이트합니다.</p>	<p data-bbox="592 1375 1031 1512">다음 예제를 기반으로 배포를 생성할 때 구성 요소 구성을 업데이트합니다.</p> <pre data-bbox="609 1554 1015 1879"> { "region": "<region>", "parquet_period": <period>, "s3_bucket": "<s3Bucket>", "s3_key_prefix": "<s3prefix>" } </pre>	<p data-bbox="1063 1375 1209 1417">앱 개발자</p>

작업	설명	필요한 기술
	<pre>} AWS 리전, 주기적 간격의 <region>, S3 버킷이 있는 <period>, 접두사가 있는 <s3Bucket> 로 바꿉니다 <s3prefix> .</pre>	

작업	설명	필요한 기술
구성 요소를 생성합니다.	<p>다음 중 하나를 수행합니다.</p> <ul style="list-style-type: none"> • 구성 요소를 생성합니다. • CI/CD 파이프라인(있는 경우)에 구성 요소를 추가합니다. 아티팩트 리포지토리의 아티팩트를 AWS IoT Greengrass 아티팩트 버킷으로 복사해야 합니다. 그런 다음 AWS IoT Greengrass 구성 요소를 생성하거나 업데이트합니다. • MQTT 브로커를 구성 요소로 추가하거나 나중에 수동으로 추가하세요. 참고: 이 결정은 브로커와 함께 사용할 수 있는 인증 체계에 영향을 미칩니다. 브로커를 수동으로 추가하면 브로커를 AWS IoT Greengrass에서 분리하고 브로커의 지원되는 인증 체계를 사용할 수 있습니다. AWS에서 제공하는 브로커 구성 요소에는 사전 정의된 인증 체계가 있습니다. 자세한 내용은 MQTT 3.1.1 브로커(Moquette) 및 MQTT 5 브로커(EMQX)를 참조하세요. 	앱 개발자

작업	설명	필요한 기술
<p>MQTT 클라이언트를 업데이트합니다.</p>	<p>구성 요소가 브로커에 로컬로 연결되므로 샘플 코드는 인증을 사용하지 않습니다. 시나리오가 다를 경우 필요에 따라 MQTT 클라이언트 섹션을 업데이트하세요. 또한 다음 사항을 수행하세요.</p> <ol style="list-style-type: none"> 1. 구독에서 MQTT 주제를 업데이트합니다. 2. 각 소스의 메시지가 다를 수 있으므로 필요에 따라 MQTT 메시지 파서를 업데이트합니다. 	<p>앱 개발자</p>

AWS IoT Greengrass 버전 2 코어 디바이스에 구성 요소 추가

작업	설명	필요한 기술
<p>코어 디바이스 배포를 업데이트합니다.</p>	<p>AWS IoT Greengrass 버전 2 코어 디바이스의 배포가 이미 있는 경우 배포를 수정하세요. 배포가 존재하지 않는 경우 새 배포를 생성하세요.</p> <p>구성 요소에 올바른 이름을 지정하려면 다음을 기반으로 새 구성 요소(필요한 경우)에 대한 로그 관리자 구성을 업데이트하세요.</p> <pre data-bbox="592 1705 1031 1839"> { "logsUploaderConfiguration": { </pre>	<p>앱 개발자</p>

작업	설명	필요한 기술
	<pre> "systemLogsConfiguration": { ... }, "componentLogsConfigurationMap": { "<com.iot .ingest.parquet>": { "minimumLogLevel": "INFO", "diskSpaceLimit": "20", "diskSpaceLimitUnit": "MB", "deleteLogFileAfterCloudUpload": "false" } ... } }, "periodicUploadIntervalSec": "300" } </pre> <p>마지막으로 AWS IoT Greengrass 코어 디바이스에 대한 배포 개정을 완료하세요.</p>	

S3 버킷으로의 데이터 모으기를 검증합니다.

작업	설명	필요한 기술
AWS IoT Greengrass 볼륨에 대한 로그를 확인합니다.	<p>다음 사항을 확인합니다.</p> <ul style="list-style-type: none"> MQTT 클라이언트가 로컬 MQTT 브로커에 성공적으로 연결되었습니다. 	앱 개발자

작업	설명	필요한 기술
	<ul style="list-style-type: none"> • MQTT 클라이언트가 올바른 주제를 구독하고 있습니다. • MQTT 주제에 대한 센서 업데이트 메시지가 브로커에 전달됩니다. • 패키지 압축은 매 주기마다 발생합니다. 	
S3 버킷을 선택합니다.	<p>데이터가 S3 버킷에 업로드되고 있는지 검증합니다. 매 기간마다 업로드되는 파일을 확인할 수 있습니다.</p> <p>다음 섹션에서 데이터를 쿼리하여 데이터가 S3 버킷에 업로드되었는지 확인할 수도 있습니다.</p>	앱 개발자

Athena에서 쿼리 설정

작업	설명	필요한 기술
데이터베이스 및 테이블을 생성합니다.	<ol style="list-style-type: none"> 1. AWS Glue 데이터베이스를 생성합니다(필요한 경우). 2. AWS Glue에서 수동으로 실행하거나 AWS Glue에서 크롤러를 실행하여 테이블을 생성합니다. 	앱 개발자
Athena에게 데이터에 대한 액세스 권한을 부여합니다.	<ol style="list-style-type: none"> 1. Athena가 S3 버킷에 액세스할 수 있도록 권한을 업데이트합니다. 자세한 내용은 Athena 설명서에서 AWS Glue 데이터 카탈로그의 데이터베이스와 테이블에 대 	앱 개발자

작업	설명	필요한 기술
	<p>한 세분화된 액세스를 참조하세요.</p> <p>2. 데이터베이스의 테이블을 쿼리합니다.</p>	

문제 해결

문제	Solution
MQTT 클라이언트 연결 실패	<ul style="list-style-type: none"> MQTT 브로커의 권한을 검증합니다. AWS의 MQTT 브로커가 있는 경우 MQTT 3.1.1 브로커(Moquette) 및 MQTT 5 브로커(EMQX)를 참조하세요. MQTT 클라이언트에서 보안 인증 정보를 검증합니다. AWS의 MQTT 브로커가 있는 경우 MQTT 3.1.1 브로커(Moquette) 및 MQTT 5 브로커(EMQX)를 참조하세요.
MQTT 클라이언트 구독 실패	MQTT 브로커의 권한을 검증합니다. AWS의 MQTT 브로커가 있는 경우 MQTT 3.1.1 브로커(Moquette) 및 MQTT 5 브로커(EMQX) 를 참조하세요.
패케이 파일은 생성되지 않습니다	<ul style="list-style-type: none"> MQTT 주제가 올바른지 검증합니다. 센서의 MQTT 메시지 형식이 올바른지 검증합니다.
객체가 S3 버킷에 업로드되지 않았습니다.	<ul style="list-style-type: none"> 인터넷 연결 및 엔드포인트 연결이 있는지 검증합니다. S3 버킷의 리소스 정책이 올바른지 검증합니다. AWS IoT Greengrass 버전 2 코어 디바이스 역할에 대한 권한을 검증합니다.

관련 리소스

- [DataFrame](#)(판다스 설명서)
- [Apache Parquet 설명서](#)(Parquet 설명서)
- [AWS IoT Greengrass 구성 요소 개발](#)(AWS IoT Greengrass 개발자 가이드, 버전 2)
- [AWS IoT Greengrass 구성 요소를 디바이스에 배포](#)(AWS IoT Greengrass 개발자 가이드, 버전 2)
- [로컬 IoT 디바이스와 상호 작용](#)(AWS IoT Greengrass 개발자 가이드, 버전 2)
- [MQTT 3.1.1 브로커\(Moquette\)](#)(AWS IoT Greengrass 개발자 가이드, 버전 2)
- [MQTT 5 브로커\(EMQX\)](#)(AWS IoT Greengrass 개발자 가이드, 버전 2)

추가 정보

비용 분석

다음 비용 분석 시나리오는 이 패턴에서 다루는 데이터 모으기 접근 방식이 AWS 클라우드의 데이터 모으기 비용에 어떤 영향을 미칠 수 있는지를 보여줍니다. 이 시나리오의 요금 예시는 발행 시점의 가격을 기준으로 합니다. 요금은 변경될 수 있습니다. 또한 비용은 AWS 리전, AWS service quotas 및 클라우드 환경과 관련된 기타 요인에 따라 달라질 수 있습니다.

입력 신호 세트

이 분석에서는 다음과 같은 입력 신호 세트를 기반으로 IoT 수집 비용을 사용 가능한 다른 대안과 비교합니다.

신호 수	Frequency(주파수)	신호당 데이터
125	25Hz	8 bytes

이 시나리오에서 시스템은 125개의 신호를 수신합니다. 각 신호는 8바이트이며 40밀리초(25Hz)마다 발생합니다. 이러한 신호는 개별적으로 제공되거나 공통 페이로드에 그룹화될 수 있습니다. 필요에 따라 이러한 신호를 분리하고 패킹할 수 있습니다. 지연 시간도 확인할 수 있습니다. 지연 시간은 데이터 수신, 누적 및 수집 기간으로 구성됩니다.

비교를 위해 이 시나리오의 수집 작업은 us-east-1 AWS 리전을 기반으로 합니다. 비용 비교는 AWS 서비스에만 적용됩니다. 하드웨어 또는 연결과 같은 기타 비용은 분석에 포함되지 않습니다.

비용 비교

다음 표는 각 섭취 방법에 대한 월별 비용을 미국 달러(USD)로 보여줍니다.

방법	월별 비용
AWS SiteWise IoT*	331.77 USD
데이터 처리 팩이 포함된 AWS IoT SiteWise Edge (모든 데이터를 엣지에 유지)	200 USD
원시 데이터 액세스를 위한 AWS IoT Core 및 Amazon S3 규칙	84.54 USD
엣지에서의 패키지 파일 압축 및 Amazon S3에 업로드	0.5 USD

*Service Quotas를 준수하려면 데이터를 다운샘플링해야 합니다. 즉, 이 방법을 사용하면 데이터가 약간 손실될 수 있습니다.

대체 방법

이 섹션에서는 다음과 같은 대체 방법에 대한 해당 비용을 보여줍니다.

- AWS IoT SiteWise — 각 신호는 개별 메시지로 업로드해야 합니다. 따라서 월별 총 메시지 수는 $125 \times 25 \times 3600 \times 24 \times 30$, 즉 월별 81억 개의 메시지입니다. 하지만 AWS IoT는 속성당 초당 10개의 데이터 포인트만 처리할 SiteWise 수 있습니다. 데이터를 10Hz로 다운샘플링한다고 가정하면 월별 메시지 수는 $125 \times 10 \times 3600 \times 24 \times 30$, 즉 32억 4천만 개로 줄어듭니다. 측정값을 10개씩 그룹으로 묶는 퍼블리셔 구성 요소를 사용하면(메시지 백만 개당 1 USD) 월별 비용은 324 USD입니다. 각 메시지가 8바이트(1Kb/125)라고 가정하면 25.92Gb의 데이터 스토리지가 됩니다. 이로 인해 매월 7.77 USD의 월별 비용이 추가됩니다. 첫 달의 총 비용은 331.77 USD이며 매달 7.77 USD씩 증가합니다.
- SiteWise 엣지에서 완전히 처리된 모든 모델 및 신호 (즉, 클라우드 통합 없음) 를 포함하는 데이터 처리 팩이 포함된 AWS IoT Edge — 데이터 처리 팩을 대안으로 사용하여 비용을 절감하고 엣지에서 계산되는 모든 모델을 구성할 수 있습니다. 이는 실제 계산이 수행되지 않더라도 스토리지 및 시각화에만 사용할 수 있습니다. 이 경우 엣지 게이트웨이에 강력한 하드웨어를 사용해야 합니다. 매월 200 USD의 고정 비용이 부과됩니다.
- MQTT를 통한 AWS IoT Core로의 직접 수집 및 Amazon S3에 원시 데이터를 저장하는 IoT 규칙 - 모든 신호가 공통 페이로드에 게시된다고 가정하면 AWS IoT Core에 게시되는 총 메시지 수는

25×3600×24×30, 즉 매월 6,480만 개입니다. 메시지 백만 개당 1 USD로 계산하면 월별 비용은 64.8 USD입니다. 규칙 활성화 횟수 백만 회당 0.15 USD이고 메시지당 규칙 1개를 사용할 경우 월별 비용은 19.44 USD입니다. Amazon S3의 스토리지 GB당 0.023 USD의 비용이 발생하므로 매월 1.5 USD가 추가됩니다(새 데이터를 반영하기 위해 매월 증가). 첫 달의 총 비용은 84.54 USD이며 매달 1.5 USD씩 증가합니다.

- Parquet 파일의 엣지에서 데이터를 압축하여 Amazon S3에 업로드(제안된 방법) - 압축률은 데이터 유형에 따라 다릅니다. MQTT에 대해 동일한 산업 데이터를 테스트한 결과, 한 달 동안의 총 출력 데이터는 1.2Gb입니다. 이 비용은 월 0.03 USD입니다. 다른 벤치마크에서 설명한 압축률(무작위 데이터 사용)은 66% 정도입니다(최악의 시나리오에 가까움). 총 데이터는 21Gb이고 비용은 월 0.5 USD입니다.

파케이 파일 생성기

다음 코드 예제는 Python으로 작성된 Parquet 파일 생성기의 구조를 보여줍니다. 코드 예제는 설명을 위한 용도로만 사용되며 사용자 환경에 붙여넣으면 작동하지 않습니다.

```
import queue
import paho.mqtt.client as mqtt
import pandas as pd

#queue for decoupling the MQTT thread
messageQueue = queue.Queue()
client = mqtt.Client()
streammanager = StreamManagerClient()

def feederListener(topic, message):
    payload = {
        "topic" : topic,
        "payload" : message,
    }
    messageQueue.put_nowait(payload)

def on_connect(client_instance, userdata, flags, rc):
    client.subscribe("#", qos=0)

def on_message(client, userdata, message):
    feederListener(topic=str(message.topic),
        message=str(message.payload.decode("utf-8")))

filename = "tempfile.parquet"
```

```
streamname = "mystream"
destination_bucket= "mybucket"
keyname="mykey"
period= 60

client.on_connect = on_connect
client.on_message = on_message
streammanager.create_message_stream(
    MessageStreamDefinition(name=streamname,
        strategy_on_full=StrategyOnFull.OverwriteOldestData)
    )

while True:
    try:
        message = messageQueue.get(timeout=myArgs.mqtt_timeout)
    except (queue.Empty):
        logger.warning("MQTT message reception timed out")

    currentTimeStamp = getCurrentTime()
    if currentTimeStamp >= nextUploadTimestamp:
        df = pd.DataFrame.from_dict(accumulator)
        df.to_parquet(filename)
        s3_export_task_definition = S3ExportTaskDefinition(input_url=filename,
            bucket=destination_bucket, key=key_name)
        streammanager.append_message(streamname,
            Util.validate_and_serialize_to_json_bytes(s3_export_task_definition))
        accumulator = {}
        nextUploadTimestamp += period
    else:
        accumulator.append(message)
```

WanDisco 마이그레이션기를 사용하여 하둡 데이터를 Amazon S3로 마이그레이션합니다. LiveData

작성자: Tony Velcich

소스: 온프레미스 Hadoop 클러스터	대상: Amazon S3	R 유형: 리호스팅
환경: 프로덕션	기술: 데이터 레이크, 빅 데이터, 하이브리드 클라우드, 마이그레이션	워크로드: 기타 모든 워크로드
AWS 서비스: Amazon S3		

요약

이 패턴은 Hadoop 분산 파일 시스템(HDFS)에서 Amazon Simple Storage Service(S3)로 Apache Hadoop 데이터를 마이그레이션합니다. WanDisco Migrator를 사용하여 데이터 LiveData 마이그레이션 프로세스를 자동화합니다.

사전 조건 및 제한 사항

사전 조건

- 마이그레이션기가 LiveData 설치될 하둡 클러스터 에지 노드. 노드는 다음 요구 사항을 충족해야 합니다.
 - 최소 사양: CPU 4개, RAM 16GB, 스토리지 100GB.
 - 최소 2Gbps의 네트워크.
 - 엣지 노드에서 포트 8081에 액세스하여 WanDisco UI에 액세스할 수 있습니다.
 - Java 1.8 64비트
 - 엣지 노드에 Hadoop 클라이언트 라이브러리가 설치되었습니다.
 - [HDFS 슈퍼유저](#)로 인증할 수 있습니다(예: "hdfs").
 - Hadoop 클러스터에서 Kerberos를 사용하도록 설정한 경우 HDFS 슈퍼유저에 적합한 보안 주체가 포함된 유효한 키탭을 엣지 노드에서 사용할 수 있어야 합니다.
 - 지원되는 운영 체제의 전체 목록은 [릴리스 정보](#)를 참조하십시오.

- S3 버킷에 액세스할 수 있는 활성 AWS 계정.
- 온프레미스 Hadoop 클러스터 (특히 엣지 노드) 와 AWS 사이에 설정된 AWS Direct Connect 링크.

제품 버전

- LiveData 마이그레이션자 1.8.6
- WANdisco UI(OneUI) 5.8.0

아키텍처

소스 기술 스택

- 온프레미스 Hadoop 클러스터

대상 기술 스택

- Amazon S3

아키텍처

다음 다이어그램은 LiveData Migrator 솔루션 아키텍처를 보여줍니다.

워크플로는 온프레미스 HDFS에서 Amazon S3로 데이터를 마이그레이션하기 위한 네 가지 기본 구성 요소로 구성되어 있습니다.

- [LiveData 마이그레이션자](#) — HDFS에서 Amazon S3로의 데이터 마이그레이션을 자동화하고 하둡 클러스터의 엣지 노드에 상주합니다.
- [HDFS](#) – 애플리케이션 데이터에 대한 높은 처리량 액세스를 제공하는 분산 파일 시스템입니다.
- [Amazon S3](#) – 업계 최고의 확장성, 데이터 가용성, 보안 및 성능을 제공하는 객체 스토리지 서비스입니다.
- [AWS Direct Connect](#) – 온프레미스 데이터 센터에서 AWS로 전용 네트워크 연결을 설정하는 서비스입니다.

자동화 및 규모 조정

일반적으로 경로 또는 디렉토리별로 소스 파일 시스템에서 특정 콘텐츠를 선택할 수 있도록 여러 마이그레이션을 생성합니다. 또한 여러 마이그레이션 리소스를 정의하여 여러 독립 파일 시스템으로 데이터를 동시에 마이그레이션할 수 있습니다.

에픽

AWS 계정에서 Amazon S3 스토리지 구성

작업	설명	필요한 기술
계정에 로그인.	AWS Management Console에 로그인하고 https://console.aws.amazon.com/s3/ 에서 Amazon S3 콘솔을 엽니다.	AWS 환경
S3 버킷을 생성합니다.	대상 스토리지로 사용할 기존 S3 버킷이 아직 없는 경우, Amazon S3 콘솔에서 “버킷 생성” 옵션을 선택하고 퍼블릭 액세스 차단을 위한 버킷 이름, AWS 리전 및 버킷 설정을 지정합니다. AWS와 WanDisco는 S3 버킷에 대한 블록 퍼블릭 액세스 옵션을 활성화하고 조직의 요구 사항에 맞게 버킷 액세스 및 사용자 권한 정책을 설정할 것을 권장합니다. AWS 예제는 https://docs.aws.amazon.com/AmazonS3/latest/dev/example-walkthroughs-managing-access-example1.html 에서 제공됩니다.	AWS 환경

LiveData 마이그레이터 설치

작업	설명	필요한 기술
마이그레이션자 LiveData 설치 프로그램을 다운로드하세요.	LiveData Migrator 설치 프로그램을 다운로드하여 하둡 에지 노드에 업로드합니다. https://www2.wandisco.com/ldm-trial 에서 Migrator 무료 평가판을 다운로드할 수 있습니다. LiveData . AWS Marketplace 의 https://aws.amazon.com/marketplace/pp/B07B8SZND9 에서도 LiveData 마이그레이션자에 액세스할 수 있습니다.	Hadoop 관리자, 애플리케이션 소유자
LiveData 마이그레이션기를 설치하십시오.	다운로드한 설치 프로그램을 사용하여 Hadoop 클러스터의 에지 노드에 HDFS 슈퍼유저로 LiveData Migrator를 설치합니다. 설치 명령에 대한 내용은 “추가 정보” 섹션을 참조하십시오.	Hadoop 관리자, 애플리케이션 소유자
마이그레이션자 및 기타 서비스의 상태를 확인하십시오. LiveData	“추가 정보” 섹션에 제공된 명령을 사용하여 LiveData 마이그레이션자, Hive 마이그레이션자 및 WanDisco UI의 상태를 확인합니다.	Hadoop 관리자, 애플리케이션 소유자

WanDisco UI를 통해 스토리지 구성

작업	설명	필요한 기술
마이그레이션자 계정을 등록하세요. LiveData	포트 8081(Hadoop 엣지 노드)에서 웹 브라우저를 통해 WanDisco UI에 로그인하	애플리케이션 소유자

작업	설명	필요한 기술
	<p>고 등록 세부 정보를 제공합니다. 예를 들어 myldmhost.example.com이라는 호스트에서 LiveData 마이그레이션기를 실행하는 경우 URL은 다음과 같습니다. http://myldmhost.example.com:8081</p>	
<p>소스 HDFS 스토리지를 구성합니다.</p>	<p>소스 HDFS 스토리지에 필요한 구성 세부 정보를 제공하십시오. 여기에는 “fs.DefaultFS” 값과 사용자 정의 스토리지 이름이 포함됩니다. Kerberos가 활성화된 경우 마이그레이션자가 사용할 기본 및 키탭 위치를 제공하십시오. LiveData 클러스터에서 NameNode HA가 활성화된 경우 에지 노드의 core-site.xml 및 hdfs-site.xml 파일 경로를 제공하십시오.</p>	<p>Hadoop 관리자, 애플리케이션 소유자</p>

작업	설명	필요한 기술
대상 Amazon S3 스토리지를 구성합니다.	대상 스토리지를 S3a 유형으로 추가합니다. 사용자 정의 스토리지 이름과 S3 버킷 이름을 제공합니다. 자격 증명 공급자 옵션에 “org.apache.hadoop.fs.s3a.simple”을 입력하고 S3 AWSCredentialsProvider 버킷에 대한 AWS 액세스 및 비밀번호 키를 제공합니다. 추가 S3a 속성도 필요합니다. 자세한 내용은 마이그레이션자 설명서 https://docs.wandisco.com/docs/command-reference/#3a 의 “S3a 속성” 섹션을 참조하십시오. LiveData live-data-migrator filesystem-add-s	AWS, 애플리케이션 소유자

마이그레이션 준비

작업	설명	필요한 기술
제외 항목 추가를 추가합니다 (필요한 경우).	마이그레이션에서 특정 데이터 세트를 제외하려면 소스 HDFS 스토리지에 대한 제외를 추가합니다. 이러한 제외는 파일 크기, 파일 이름(정규식 패턴 기반), 수정 날짜를 기반으로 할 수 있습니다.	Hadoop 관리자, 애플리케이션 소유자

마이그레이션 생성 및 시작

작업	설명	필요한 기술
마이그레이션을 생성하고 구성합니다.	WanDisco UI의 대시보드에서 마이그레이션을 생성합니다. 소스(HDFS)와 대상(S3 버킷)을 선택합니다. 이전 단계에서 정의한 새 제외를 추가합니다. “덮어쓰기” 또는 “크기가 일치하면 건너뛰기” 옵션을 선택합니다. 모든 필드가 완성되면 마이그레이션을 생성합니다.	Hadoop 관리자, 애플리케이션 소유자
마이그레이션을 시작합니다.	대시보드에서 생성한 마이그레이션을 선택합니다. 마이그레이션을 시작하려면 클릭합니다. 마이그레이션을 생성할 때 자동 시작 옵션을 선택하여 마이그레이션을 자동으로 시작할 수도 있습니다.	애플리케이션 소유자

대역폭 관리(선택 사항)

작업	설명	필요한 기술
소스와 대상 간의 네트워크 대역폭 제한을 설정합니다.	대시보드의 스토리지 목록에서 소스 스토리지를 선택하고 그룹화 목록에서 “대역폭 관리”를 선택합니다. 무제한 옵션의 선택을 취소하고 최대 대역폭 제한 및 단위를 정의합니다. “적용”을 선택합니다.	애플리케이션 소유자, 네트워킹

마이그레이션 모니터링 및 관리

작업	설명	필요한 기술
WanDisco UI를 사용하여 마이그레이션 정보를 볼 수 있습니다.	WanDisco UI를 사용하여 라이선스, 대역폭, 스토리지 및 마이그레이션 정보를 볼 수 있습니다. UI는 또한 알림 시스템을 제공하므로 오류, 경고 또는 사용과 관련된 중요한 이정표에 대한 알림을 받을 수 있습니다.	Hadoop 관리자, 애플리케이션 소유자
마이그레이션을 중지, 재개 및 삭제합니다.	중지된 상태로 전환하여 마이그레이션할 때 대상으로의 콘텐츠 전송을 중지할 수 있습니다. 중지된 마이그레이션을 재개할 수 있습니다. 중지된 상태의 마이그레이션도 삭제할 수 있습니다.	Hadoop 관리자, 애플리케이션 소유자

관련 리소스

- [LiveData 마이그레이션자 설명서](#)
- [LiveData AWS 마켓플레이스의 마이그레이션자](#)
- [WANdisco 지원 커뮤니티](#)
- [완디스코 LiveData 마이그레이터 데모 \(동영상\)](#)

추가 정보

LiveData 마이그레이터 설치

설치 프로그램이 작업 디렉토리 내에 있다고 가정하면 다음 명령을 사용하여 LiveData Migrator를 설치할 수 있습니다.

```
su - hdfs
chmod +x livedata-migrator.sh && sudo ./livedata-migrator.sh
```

설치 후 LiveData Migrator 및 기타 서비스의 상태 확인

다음 명령을 사용하여 LiveData 마이그레이션자, Hive 마이그레이션자 및 WanDisco UI의 상태를 확인합니다.

```
service livedata-migrator status
service hivemigrator status
service livedata-ui status
```

패턴 더 보기

- [AWS Glue를 사용하여 Amazon S3에서 Amazon Redshift로 데이터를 점차 늘려 로딩하기 위한 ETL 서비스 파이프라인 빌드](#)
- [Kinesis Data Streams를 사용하고 AWS CDK를 사용하는 Amazon Data Firehose를 사용하여 Amazon S3에 DynamoDB 레코드를 제공합니다.](#)
- [Amazon Redshift 클러스터를 생성할 때 암호화되었는지 확인합니다.](#)
- [AWS Glue 작업과 Python을 사용하여 테스트 데이터 생성](#)
- [Starburst를 사용하여 데이터를 AWS 클라우드로 마이그레이션하십시오.](#)
- [AWS에서 입력 파일 크기의 ETL 수집 최적화](#)
- [AWS Step Functions를 사용하여 검증, 변환 및 파티셔닝을 통해 ETL 파이프라인 오케스트레이션](#)
- [대규모 Db2 z/OS 데이터를 CSV 파일로 Amazon S3에 전송](#)
- [새 Amazon Redshift 클러스터에 필수 SSL 엔드포인트가 있는지 확인](#)
- [Amazon Athena와 Amazon을 사용하여 Amazon Redshift 감사 로그를 시각화합니다. QuickSight](#)

데이터베이스 수

주제

- [연결된 서버를 사용하여 Amazon EC2의 Microsoft SQL Server에서 온프레미스 Microsoft SQL Server 테이블에 액세스합니다.](#)
- [읽기 전용 복제본을 사용하여 Amazon RDS 커스텀 기반 PeopleSoft 오라클에 HA를 추가합니다.](#)
- [SQL Server 데이터베이스를 AWS의 MongoDB Atlas로 마이그레이션하기 위한 쿼리 성능 평가](#)
- [AWS 계정 전반에 걸쳐 Amazon RDS 인스턴스의 복제를 자동화](#)
- [Systems Manager를 사용하여 SAP HANA 데이터베이스를 자동으로 백업하고 EventBridge](#)
- [Cloud Custodian을 사용하여 Amazon RDS에 대한 공개 액세스 차단하기](#)
- [AWS 기반 SQL Server의 Always On 가용성 그룹에서 읽기 전용 라우팅 구성](#)
- [pgAdmin에서 SSH 터널을 사용하여 연결](#)
- [JSON Oracle 쿼리를 PostgreSQL 데이터베이스 SQL로 변환](#)
- [AWS Backup을 사용하여 계정 전반적으로 Amazon DynamoDB 테이블의 복사](#)
- [사용자 지정 구현을 사용하여 계정 전반적으로 Amazon DynamoDB 테이블을 복사합니다.](#)
- [Amazon RDS 및 Amazon Aurora에 대한 자세한 비용 및 사용 보고서 생성](#)
- [Aurora PostgreSQL의 사용자 지정 엔드포인트를 사용하여 Oracle RAC 워크로드 에뮬레이션하기](#)
- [Amazon RDS에서 PostgreSQL DB 인스턴스에 대한 암호화된 연결 활성화하기](#)
- [기존 Amazon RDS for PostgreSQL DB 인스턴스 암호화하기](#)
- [시작 시 Amazon RDS 데이터베이스의 자동 태그 지정 적용](#)
- [온디맨드 용량에 대한 DynamoDB 테이블의 비용 추정](#)
- [Amazon DynamoDB 테이블의 스토리지 비용 추정](#)
- [AWR 보고서를 사용하여 오라클 데이터베이스의 Amazon RDS 엔진 크기를 추정합니다.](#)
- [AWS DMS를 사용하여 Amazon RDS for SQL Server 테이블을 S3 버킷으로 내보내기](#)
- [Aurora PostgreSQL의 동적 SQL 명령문에서 익명 블록 처리](#)
- [Aurora PostgreSQL-Compatible에서 오버로드된 Oracle 함수 처리](#)
- [DynamoDB 태깅 적용 지원](#)
- [AWS DMS와 Amazon Aurora를 사용하여 지역 간 재해 복구 구현](#)
- [100개 이상의 인수가 있는 Oracle 함수 및 프로시저를 PostgreSQL로 마이그레이션](#)
- [Amazon RDS for Oracle DB 인스턴스를 AMS를 사용하는 다른 계정으로 마이그레이션](#)

- [Oracle OUT 바인드 변수를 PostgreSQL 데이터베이스로 마이그레이션](#)
- [동일한 호스트 이름을 가진 SAP HSR을 사용하여 SAP HANA를 AWS로 마이그레이션](#)
- [분산된 가용성 그룹을 사용하여 SQL Server를 AWS로 마이그레이션](#)
- [AWS DMS를 사용하여 오라클 8i 또는 9i에서 오라클용 아마존 RDS로 마이그레이션할 수 있습니다. SharePlex](#)
- [암호화를 사용하지 않는 인스턴스가 있는지 Amazon Aurora를 모니터링](#)
- [Amazon을 사용하여 오라클 GoldenGate 로그를 모니터링하십시오. CloudWatch](#)
- [Amazon RDS for Oracle에서 Oracle Database Enterprise Edition을 Standard Edition 2로 리플랫폼](#)
- [Precisely Connect를 사용하여 메인프레임 데이터베이스를 AWS에 복제하기](#)
- [Lambda와 Secrets Manager를 사용하여 Amazon RDS for PostgreSQL 및 Aurora PostgreSQL 작업 예약하기](#)
- [신뢰할 수 있는 컨텍스트를 사용하여 AWS에서 Db2 페더레이션 데이터베이스의 사용자 액세스 보호 및 간소화](#)
- [온프레미스 SMTP 서버 및 Database Mail을 사용하여 Amazon RDS for SQL Server 데이터베이스 인스턴스에 대한 알림 전송하기](#)
- [AWS 기반 IBM Db2에서 SAP를 위한 재해 복구 설정](#)
- [활성 대기 데이터베이스를 사용하여 Amazon RDS Custom에서 Oracle E-Business Suite를 위한 HA/DR 아키텍처를 설정합니다.](#)
- [GTID를 사용하여 Amazon RDS for MySQL와 Amazon EC2의 MySQL 간에 데이터 복제를 설정합니다.](#)
- [Amazon RDS 기반 오라클 PeopleSoft 애플리케이션의 역할 전환 \(오라클용 사용자 지정\)](#)
- [워크로드별 데이터베이스 마이그레이션 패턴](#)
- [패턴 더 보기](#)

연결된 서버를 사용하여 Amazon EC2의 Microsoft SQL Server에서 온프레미스 Microsoft SQL Server 테이블에 액세스합니다.

작성자: Tirumala Dasari(AWS) 및 Eduardo Valentim(AWS)

환경: PoC 또는 파일럿

기술: 데이터베이스

워크로드: Microsoft

요약

이 패턴은 연결된 서버를 사용하여 Amazon Elastic Compute Cloud(Amazon EC2) Windows 또는 Linux 인스턴스에서 실행되거나 호스팅되는 Microsoft SQL Server 데이터베이스에서 Microsoft Windows에서 실행되는 온프레미스 Microsoft SQL Server 데이터베이스 테이블에 액세스하는 방법을 설명합니다.

사전 조건 및 제한 사항

사전 조건

- 활성 상태의 AWS 계정.
- Amazon Linux AMI에서 실행되는 Microsoft SQL 서버가 탑재된 Amazon EC2 (Amazon Machine Image)
- 온프레미스 Microsoft SQL 서버(Windows)와 Windows 또는 Linux EC2 인스턴스 간의 AWS Direct Connect

제품 버전

- SQL Server 2016 이상

아키텍처

소스 기술 스택

- Windows에서 실행되는 온프레미스 Microsoft SQL Server 데이터베이스
- Windows AMI 또는 Linux AMI에서 실행되는 Microsoft SQL 서버가 탑재된 Amazon EC2

대상 기술 스택

- Amazon Linux AMI에서 실행되는 Microsoft SQL Server를 사용하는 Amazon EC2
- Windows AMI에서 실행되는 Microsoft SQL Server를 사용하는 Amazon EC2

소스 및 대상 데이터베이스 아키텍처

도구

- [Microsoft SQL Server Management Studio\(SSMS\)](#)는 SQL Server 인프라를 관리하기 위한 통합 환경입니다. SQL Server와 상호 작용하는 다양한 스크립트 편집기와 함께 사용자 인터페이스와 도구 그룹을 제공합니다.

에픽

Windows SQL 서버에서 SQL 서버용 인증 모드를 Windows로 변경

작업	설명	필요한 기술
SSMS를 통해 Windows SQL 서버에 연결합니다.		DBA
SQL Server 인스턴스의 컨텍스트 메뉴(마우스 오른쪽 클릭)에서 인증 모드를 Windows로 변경합니다.		DBA

Windows MSSQL 서비스를 다시 시작합니다.

작업	설명	필요한 기술
SQL 서비스를 다시 시작합니다.	1. SSMS 객체 탐색기에서 SQL Server 인스턴스를 선택합니다.	DBA

작업	설명	필요한 기술
	2. 마우스 오른쪽 클릭으로 컨텍스트 메뉴를 엽니다. 3. 다시 시작을 선택합니다.	

Windows SQL Server에서 새 로그인을 생성하고 액세스할 데이터베이스를 선택합니다.

작업	설명	필요한 기술
보안 탭에서 로그인에 대한 컨텍스트 메뉴(마우스 오른쪽 클릭)를 열고 새 로그인을 선택합니다.		DBA
일반 탭에서 SQL Server 인증을 선택하고 사용자 이름을 입력하고 암호를 입력한 다음 암호를 확인하고 다음 로그인 시 암호 변경 옵션을 선택 취소합니다.		DBA
서버 역할 탭에서 퍼블릭을 선택합니다.		DBA
사용자 매핑 탭에서 액세스하려는 데이터베이스와 스키마를 선택한 다음 데이터베이스를 강조 표시하여 데이터베이스 역할을 선택합니다.	퍼블릭 및 db_datareader를 선택하여 데이터베이스 테이블의 데이터에 액세스합니다.	DBA
확인을 선택하여 사용자를 생성합니다.		DBA

Linux SQL 서버 호스트 파일에 Windows SQL 서버 IP 추가

작업	설명	필요한 기술
터미널 창을 통해 Linux SQL 서버 박스에 연결합니다.		DBA
/etc/hosts 파일을 열고 SQL 서버가 설치된 Windows 시스템의 IP 주소를 추가합니다.		DBA
호스트 파일을 저장합니다.		DBA

Linux SQL 서버에 연결된 서버 생성

작업	설명	필요한 기술
저장된 절차인 master.sys.sp_addlinkedserver 및 master.dbo.sp_addlinkedsrvl을 사용하여 연결된 서버를 생성합니다.	이러한 저장 프로시저 사용에 대한 자세한 내용은 추가 정보 섹션을 참조하세요.	DBA, 개발자

SSMS에서 생성된 연결 서버 및 데이터베이스 확인

작업	설명	필요한 기술
SSMS의 Linux SQL Server에서 연결된 서버로 이동하여 새로 고칩니다.		DBA
왼쪽 창에서 생성된 연결 서버 및 카탈로그를 확장합니다.	선택한 SQL Server 데이터베이스가 테이블 및 뷰와 함께 표시됩니다.	DBA

Windows SQL Server 데이터베이스 테이블에 액세스할 수 있는지 확인합니다.

작업	설명	필요한 기술
SSMS 쿼리 창에서 "select top 3 * from [sqlin].dms_sample_win.dbo.mlb_data"라는 쿼리를 실행합니다.	참고로 FROM 절은 computer.database.schema.table이라는 네 부분으로 구성된 구문을 사용합니다(예: SELECT name "SQL2 databases" FROM [sqlin].master.sys.databases). 이 예제에서는 호스트 파일에 SQL2의 별칭을 만들었으므로 대괄호 사이에 실제 NetBIOS 이름을 입력할 필요가 없습니다. 실제 NetBIOS 이름을 사용하는 경우 AWS는 Win-xxxx와 같은 NetBIOS 이름을 기본값으로 사용하며 SQL Server에서는 대시가 있는 이름에는 대괄호를 사용해야 한다는 점에 유의합니다.	DBA, 개발자

관련 리소스

- [Linux 기반 SQL 서버의 릴리스 노트](#)

추가 정보

저장 프로시저를 사용하여 연결된 서버 만들기

SSMS는 Linux SQL Server용 연결된 서버 생성을 지원하지 않으므로 다음 저장 프로시저를 사용하여 서버를 만들어야 합니다.

```
EXEC master.sys.sp_addlinkedserver @server= N'SQLLIN' , @srvproduct= N'SQL Server'
```

```
EXEC master.dbo.sp_addlinkedserver  
@rmtsrvname=N'SQLLIN',@useself=N'False',@locallogin=NULL,@rmtuser=N'username',@rmtpassword='Te
```

참고 1: 이전에 Windows SQL Server에서 만든 로그인 보안 인증 정보를 저장 프로시저 `master.dbo.sp_addlinkedserver`에 입력합니다.

참고 2: `@server` 이름 `SQLLIN` 및 호스트 파일 항목 이름은 `172.12.12.4 SQLLIN`이 같아야 합니다.

다음 시나리오에 대해 이 프로세스를 사용하여 연결된 Server를 만들 수 있습니다.

- 연결된 서버를 통해 Linux SQL Server에서 Windows SQL Server로 연결(이 패턴에 지정된 대로)
- 연결된 서버를 통해 Windows SQL Server에서 Linux SQL Server로
- 연결된 서버를 통해 Linux SQL 서버를 다른 Linux SQL 서버로

읽기 전용 복제본을 사용하여 Amazon RDS 커스텀 기반 PeopleSoft 오라클에 HA를 추가합니다.

작성자: sampath kathirvel(AWS)

환경: 프로덕션

기술: 데이터베이스, 인프라

워크로드: Oracle

AWS 서비스: Amazon RDS

요약

Amazon Web Services (AWS) 에서 [Oracle PeopleSoft](#) 엔터프라이즈 리소스 계획 (ERP) 솔루션을 실행하려면 기본 운영 체제 및 [데이터베이스 환경에 액세스해야 하는 기존, 사용자 지정 및 패키지 애플리케이션을 지원하는 Amazon RDS \(Amazon RDS\)](#) 또는 Amazon RDS Custom을 사용할 수 있습니다. 마이그레이션을 계획할 때 고려해야 할 주요 요소는 AWS 권장 가이드의 [Oracle 데이터베이스 마이그레이션 전략](#)을 참조하세요.

이 글을 쓰는 시점에서 Amazon RDS for Oracle은 [다중 AZ](#) 옵션을 지원하지 않습니다. 이 옵션은 [Amazon RDS for Oracle](#)에서 스토리지 복제를 사용하는 HA 솔루션으로 사용할 수 있습니다. 대신 이 패턴은 기본 데이터베이스의 물리적 사본을 생성하고 유지 관리하는 대기 데이터베이스를 사용하여 HA를 달성합니다. 이 패턴은 Oracle Data Guard를 사용하여 읽기 전용 복제본을 설정하여 Amazon RDS Custom에서 HA를 사용하는 PeopleSoft 애플리케이션 데이터베이스를 실행하는 단계를 중점적으로 다룹니다.

또한 이 패턴은 읽기 전용 복제본을 읽기 전용 모드로 변경합니다. 읽기 전용 복제본을 읽기 전용 모드로 설정하면 다음과 같은 추가 이점이 있습니다.

- 기본 데이터베이스에서 읽기 전용 워크로드를 오프로드합니다.
- Oracle Active Data Guard 기능을 사용하여 대기 데이터베이스에서 정상 블록을 검색하여 손상된 블록을 자동으로 복구할 수 있도록 합니다
- 원거리 동기화 기능을 사용하면 장거리 다시 실행 로그 전송과 관련된 성능 오버헤드 없이 원격 대기 데이터베이스를 동기화된 상태로 유지할 수 있습니다.

읽기 전용 모드에서 복제본을 사용하려면 Oracle Active Data Guard 옵션이 필요합니다. [Oracle Active Data Guard](#) 옵션은 Oracle Database Enterprise Edition의 별도 라이선스 기능이므로 추가 비용이 듭니다.

사전 조건 및 제한 사항

사전 조건

- Amazon RDS 커스텀의 기존 PeopleSoft 애플리케이션. 애플리케이션이 없는 경우 [Oracle을 Amazon RDS 커스텀으로 PeopleSoft 마이그레이션하는 패턴](#)을 참조하십시오.
- 단일 PeopleSoft 애플리케이션 티어. 그러나 이 패턴을 여러 애플리케이션 계층에서 작동하도록 조정할 수 있습니다.
- Amazon RDS Custom은 최소 8GB의 스왑 공간으로 구성되었습니다.
- 읽기 전용 모드로 전환하고 보고 작업을 예비 복제본으로 오프로드하는 데 사용하기 위한 Oracle Active Data Guard 데이터베이스 라이선스입니다. 자세한 내용은 [Oracle Technology 상용 가격 목록](#)을 참조하세요.

제한 사항

- [RDS Custom for Oracle](#)에 대한 일반 제한 및 지원되지 않는 구성
- [Amazon RDS Custom for Oracle 읽기 전용 복제본](#)에 대한 제한 사항

제품 버전

- Amazon RDS Custom에서 지원하는 Oracle Database 버전에 대해서는 [Oracle용 RDS Custom](#)을 참조하세요.
- Amazon RDS Custom에서 지원하는 Oracle Database 인스턴스 클래스에 대해서는 [Oracle용 RDS Custom에 대한 DB 인스턴스 클래스 지원](#)을 참조하세요.

아키텍처

대상 기술 스택

- Amazon RDS Custom for Oracle
- AWS Secrets Manager
- Oracle Active Data Guard
- 오라클 PeopleSoft 애플리케이션

대상 아키텍처

다음 다이어그램에서는 Amazon RDS Custom DB 인스턴스와 Amazon RDS 사용자 지정 읽기 전용 복제본을 보여줍니다. 읽기 전용 복제본은 Oracle Active Data Guard를 사용하여 다른 가용 영역으로 복제합니다. 또한 읽기 전용 복제본을 사용하여 기본 데이터베이스의 읽기 트래픽을 오프로드하고 보고 목적으로 사용할 수 있습니다.

PeopleSoft AWS에서 Oracle을 사용하는 대표적인 아키텍처는 [AWS에서 고가용성 PeopleSoft 아키텍처 설정을](#) 참조하십시오.

도구

서비스

- [Amazon RDS Custom for Oracle](#)은 기본 운영 체제 및 데이터베이스 환경에 액세스해야 하는 레거시, 커스텀 및 패키지 애플리케이션을 위한 관리형 데이터베이스 서비스입니다.
- [AWS Secrets Manager](#)를 사용하면 코드에 하드코딩된 보안 인증 정보(암호 등)를 Secrets Manager에 대한 API 직접 호출을 통해 바꿔서 프로그래밍 방식으로 보안 암호를 검색할 수 있습니다. 이 패턴에서는 Secrets Manager에서 암호 이름 do-not-delete-rds-custom-+<<RDS Resource ID>>+-dg를 사용하여 RDS_DATAGUARD에 대한 데이터베이스 사용자 암호를 검색합니다.

기타 도구

- [Oracle Data Guard](#)를 사용하면 대기 데이터베이스를 생성, 유지, 관리 및 모니터링할 수 있습니다.

모범 사례

데이터 손실 제로(RPO=0) 목표를 달성하려면 성능 향상을 위한 다시 실행 전송 SYNC+NOAFFIRM 설정과 함께 MaxAvailability Data Guard 보호 모드를 사용하세요. 데이터베이스 보호 모드 선택에 대한 자세한 내용은 추가 정보 섹션을 참조하세요.

에픽

읽기 전용 복제본 생성

작업	설명	필요한 기술
읽기 전용 복제본을 생성합니다.	Amazon RDS 사용자 지정 DB 인스턴스의 읽기 전용 복제본	DBA

작업	설명	필요한 기술
	<p>을 생성하려면 Amazon RDS 설명서의 지침에 따라 생성한 Amazon RDS Custom DB 인스턴스 (사전 조건 섹션 참조)를 소스 데이터베이스로 사용합니다.</p> <p>기본적으로 Amazon RDS Custom 읽기 전용 복제본은 물리적 예비 복제본으로 생성되며 마운트된 상태입니다. 이는 Oracle Active Data Guard 라이선스 규정 준수를 보장하기 위한 것입니다.</p> <p>이 패턴에는 멀티테넌트 컨테이너 데이터베이스(CDB) 또는 비 CDB 인스턴스를 설정하기 위한 코드가 포함됩니다.</p>	

오라클 데이터 가드 보호 모드를 다음으로 변경하십시오. MaxAvailability

작업	설명	필요한 기술
<p>기본 데이터베이스의 Data Guard 브로커 구성에 액세스합니다.</p>	<p>이 예제에서 Amazon RDS Custom 읽기 전용 복제본은 비 CDB 인스턴스용 RDS_CUSTOM_ORCL_D 및 CDB 인스턴스용 RDS_CUSTOM_RDSCDB_B 입니다. 비 CDB용 데이터베이스는 orcl_a(기본) 및 orcl_d(대기)입니다. CDB의 데이터베이스 이름은 rdscdb_a (기본) 및 rdscdb_b(대기)입니다.</p>	<p>DBA</p>

작업	설명	필요한 기술
	<p>RDS 사용자 지정 읽기 전용 복제본에 직접 또는 기본 데이터베이스를 통해 연결할 수 있습니다. \$ORACLE_HOME/network/admin 디렉터리에 있는 tnsnames.ora 파일에서 데이터베이스의 넷 서비스 이름을 찾을 수 있습니다. RDS Custom for Oracle은 기본 데이터베이스 및 읽기 전용 복제본에 대해 이러한 항목을 자동으로 채웁니다.</p> <p>RDS_DATAGUARD 사용자의 암호는 보안 암호 이름인 do-not-delete-rds-custom-+<<RDS Resource ID>>+-dg와 함께 AWS Secrets Manager에 저장됩니다. Secrets Manager에서 검색한 SSH(보안 셸) 키를 사용하여 RDS 사용자 지정 인스턴스에 연결하는 방법에 대한 자세한 내용은 SSH를 사용하여 RDS 사용자 지정 DB 인스턴스에 연결을 참조하세요.</p> <p>Data Guard 명령줄(dgmgrl)을 통해 Oracle Data Guard 브로커 구성에 액세스하려면 다음 코드를 사용하세요.</p> <p>비 CDB</p>	

작업	설명	필요한 기술
	<pre> \$ dgmgrl RDS_DATAG UARD@RDS_CUSTOM_OR CL_D DGMGRL for Linux: Release 19.0.0.0.0 - Production on Fri Sep 30 22:44:49 2022 Version 19.10.0.0.0 Copyright (c) 1982, 2019, Oracle and/or its affiliates. All rights reserved. Welcome to DGMGRL, type "help" for informati on. Password: Connected to "ORCL_D" Connected as SYSDBG. DGMGRL> DGMGRL> show database orcl_d Database - orcl_d Role: PHYSICAL STANDBY Intended State: APPLY- ON Transport Lag: 0 seconds (computed 0 seconds ago) Apply Lag: 0 seconds (computed 0 seconds ago) Average Apply Rate: 11.00 KByte/s Instance(s): ORCL SUCCESS DGMGRL> </pre>	
	CDB	

작업	설명	필요한 기술
	<pre> -bash-4.2\$ dgmgrl C##RDS_DATAGUARD@R DS_CUSTOM_RDSCDB_B DGMGRL for Linux: Release 19.0.0.0.0 - Production on Wed Jan 11 20:24:11 2023 Version 19.16.0.0.0 Copyright (c) 1982, 2019, Oracle and/or its affiliates. All rights reserved. Welcome to DGMGRL, type "help" for informati on. Password: Connected to "RDSCDB_B " Connected as SYSDBG. DGMGRL> DGMGRL> show database rdscdb_b Database - rdscdb_b Role: PHYSICAL STANDBY Intended State: APPLY-ON Transport Lag: 0 seconds (computed 1 second ago) Apply Lag: 0 seconds (computed 1 second ago) Average Apply Rate: 2.00 KByte/s Real Time Query: OFF Instance(s): RDSCDB Database Status: SUCCESS </pre>	

작업	설명	필요한 기술
	DGMGRL>	

작업	설명	필요한 기술
<p>프라이머리 노드에서 DGMGRL에 연결하여 로그 전송 설정을 변경합니다.</p>	<p>다시 실행 전송 설정 SYNC +NOAFFIRM 에 해당하는 로그 전송 모드를 FastSync로 변경합니다. 역할 전환 후 설정이 유효한지 확인하려면 기본 데이터베이스와 대기 데이터베이스 모두에 대해 설정을 변경하세요.</p> <p>비 CDB</p> <pre>DGMGRL> DGMGRL> edit database orcl_d set property logxptmode=fastsync; Property "logxptmode" updated DGMGRL> show database orcl_d LogXptMode; LogXptMode = 'fastsync ' DGMGRL> edit database orcl_a set property logxptmode=fastsync; Property "logxptmode" updated DGMGRL> show database orcl_a logxptmode; LogXptMode = 'fastsync ' DGMGRL></pre> <p>CDB</p> <pre>DGMGRL> edit database rdscdb_b set property logxptmode=fastsyn c;DGMGRL> edit database</pre>	<p>DBA</p>

작업	설명	필요한 기술
	<pre> rdscdb_b set property logxptmode=fastsync; Property "logxptmode" updated DGMGRL> show database rdscdb_b LogXptMode; LogXptMode = 'fastsync' DGMGRL> edit database rdscdb_a set property logxptmode=fastsync; Property "logxptmode" updated DGMGRL> show database rdscdb_a logxptmode; LogXptMode = 'fastsync' DGMGRL> </pre>	

작업	설명	필요한 기술
<p>보호 모드를 로 변경합니다 MaxAvailability.</p>	<p>프라이머리 노드에서 DGMGRL로 연결하여 MaxAvailability 에 대한 보호 모드를 변경합니다.</p> <p>비 CDB</p> <pre data-bbox="592 520 1027 1396"> DGMGRL> edit configura tion set protectio n mode as maxavaila bility; Succeeded. DGMGRL> show configura tion; Configuration - rds_dg Protection Mode: MaxAvailability Members: orcl_a - Primary database orcl_d - Physical standby database Fast-Start Failover: Disabled Configuration Status: SUCCESS (status updated 38 seconds ago) DGMGRL> </pre> <p>CDB</p> <pre data-bbox="592 1507 1027 1837"> DGMGRL> show configura tion Configuration - rds_dg Protection Mode: MaxAvailability Members: rdscdb_a - Primary database </pre>	<p>DBA</p>

작업	설명	필요한 기술
	<pre> rdscdb_b - Physical standby database Fast-Start Failover: Disabled Configuration Status: SUCCESS (status updated 57 seconds ago) DGMGRL> </pre>	

복제본 상태를 마운트에서 읽기 전용으로 변경하고 재실행 적용을 활성화합니다.

작업	설명	필요한 기술
<p>대기 데이터베이스에 대한 재 실행 중지를 적용합니다.</p>	<p>읽기 전용 복제본은 기본적으로 MOUNT 모드에서 생성됩니다. 읽기 전용 모드로 열려면 먼저 기본 또는 대기 노드에서 DGMGRL에 연결하여 재실행 적용을 해제해야 합니다.</p> <p>비 CDB</p> <pre> DGMGRL> show database orcl_dDGMGRL> show database orcl_d Database - orcl_d Role: PHYSICAL STANDBY Intended State: APPLY- ON Transport Lag: 0 seconds (computed 1 second ago) Apply Lag: 0 seconds (computed 1 second ago) Average Apply Rate: 11.00 KByte/s Real Time Query: OFF Instance(s): </pre>	<p>DBA</p>

작업	설명	필요한 기술
	<pre> ORCL Database Status: SUCCESS DGMGRL> edit database orcl_d set state=app ly-off; Succeeded. DGMGRL> show database orcl_d Database - orcl_d Role: PHYSICAL STANDBY Intended State: APPLY- OFF Transport Lag: 0 seconds (computed 1 second ago) Apply Lag: 42 seconds (computed 1 second ago) Average Apply Rate: (unknown) Real Time Query: OFF Instance(s): ORCL Database Status: SUCCESS DGMGRL> CDB DGMGRL> show configura tionDGMGRL> show configuration Configuration - rds_dg Protection Mode: MaxAvailability Members: rdscdb_a - Primary database rdscdb_b - Physical standby database </pre>	

작업	설명	필요한 기술
	<pre> Fast-Start Failover: Disabled Configuration Status: SUCCESS (status updated 57 seconds ago) DGMGRL> show database rdscdb_b; Database - rdscdb_b Role: PHYSICAL STANDBY Intended State: APPLY-ON Transport Lag: 0 seconds (computed 1 second ago) Apply Lag: 0 seconds (computed 1 second ago) Average Apply Rate: 2.00 KByte/s Real Time Query: OFF Instance(s): RDSCDB Database Status: SUCCESS DGMGRL> edit database rdscdb_b set state=app ly-off; Succeeded. DGMGRL> show database rdscdb_b; Database - rdscdb_b Role: PHYSICAL STANDBY Intended State: APPLY-OFF Transport Lag: 0 seconds (computed 1 second ago) </pre>	

작업	설명	필요한 기술
	<pre>Apply Lag: 0 seconds (computed 1 second ago) Average Apply Rate: (unknown) Real Time Query: OFF Instance(s): RDSCDB Database Status: SUCCESS</pre>	

작업	설명	필요한 기술
읽기 전용 복제본 인스턴스를 읽기 전용 모드로 엽니다.	<p>TNS 항목을 사용하여 대기 데이터베이스에 접속하고 기본 또는 대기 노드에서 접속하여 대기 데이터베이스를 읽기 전용 모드로 엽니다.</p> <p>비 CDB</p> <pre> \$ sqlplus RDS_DATAGUARD@RDS_CUSTOM_ORCL_D as sysdg -bash-4.2\$ sqlplus RDS_DATAGUARD@RDS_CUSTOM_ORCL_D as sysdg SQL*Plus: Release 19.0.0.0.0 - Production on Fri Sep 30 23:00:14 2022 Version 19.10.0.0.0 Copyright (c) 1982, 2020, Oracle. All rights reserved. Enter password: Last Successful login time: Fri Sep 30 2022 22:48:27 +00:00 Connected to: Oracle Database 19c Enterprise Edition Release 19.0.0.0.0 - Production Version 19.10.0.0.0 SQL> select open_mode from v\$database; OPEN_MODE ----- MOUNTED SQL> alter database open read only; Database altered. </pre>	DBA

작업	설명	필요한 기술
	<pre> SQL> select open_mode from v\$database; OPEN_MODE ----- READ ONLY SQL> CDB -bash-4.2\$ sqlplus C##RDS_DATAGUARD@R DS_CUSTOM_RDSCDB_B as sysdg SQL*Plus: Release 19.0.0.0.0 - Productio n on Wed Jan 11 21:14:07 2023 Version 19.16.0.0.0 Copyright (c) 1982, 2022, Oracle. All rights reserved. Enter password: Last Successful login time: Wed Jan 11 2023 21:12:05 +00:00 Connected to: Oracle Database 19c Enterprise Edition Release 19.0.0.0.0 - Production Version 19.16.0.0.0 SQL> select name,open _mode from v\$database; NAME OPEN_MODE ----- RDSCDB MOUNTED SQL> alter database open read only; Database altered. </pre>	

작업	설명	필요한 기술
	<pre>SQL> select name,open _mode from v\$database; NAME OPEN_MODE ----- RDSCDB READ ONLY SQL></pre>	

작업	설명	필요한 기술
<p>읽기 전용 복제본 인스턴스에서 다시 실행 적용을 활성화합니다.</p>	<p>기본 또는 대기 노드의 DGMGRL을 사용하여 읽기 전용 복제본 인스턴스에서 다시 실행 적용을 활성화합니다.</p> <p>비 CDB</p> <pre data-bbox="592 520 1031 1768"> \$ dgmgrl RDS_DATAG UARD@RDS_CUSTOM_OR CL_D DGMGRL for Linux: Release 19.0.0.0.0 - Production on Fri Sep 30 23:02:16 2022 Version 19.10.0.0.0 Copyright (c) 1982, 2019, Oracle and/or its affiliates. All rights reserved. Welcome to DGMGRL, type "help" for informati on. Password: Connected to "ORCL_D" Connected as SYSDBG. DGMGRL> edit database orcl_d set state=apply-on; DGMGRL> edit database orcl_d set state=app ly-on; Succeeded. DGMGRL> show database orcl_d Database - orcl_d Role: PHYSICAL STANDBY Intended State: APPLY- ON </pre>	<p>DBA</p>

작업	설명	필요한 기술
	<pre> Transport Lag: 0 seconds (computed 0 seconds ago) Apply Lag: 0 seconds (computed 0 seconds ago) Average Apply Rate: 496.00 KByte/s Real Time Query: ON Instance(s): ORCL Database Status: SUCCESS DGMGRL> </pre> <p>CDB</p> <pre> -bash-4.2\$ dgmgrl C##RDS_DATAGUARD@R DS_CUSTOM_RDSCDB_B -bash-4.2\$ dgmgrl C##RDS_DATAGUARD@R DS_CUSTOM_RDSCDB_B DGMGRL for Linux: Release 19.0.0.0.0 - Production on Wed Jan 11 21:21:11 2023 Version 19.16.0.0.0 Copyright (c) 1982, 2019, Oracle and/or its affiliates. All rights reserved. Welcome to DGMGRL, type "help" for informati on. Password: Connected to "RDSCDB_B " Connected as SYSDBG. </pre>	

작업	설명	필요한 기술
	<pre> DGMGRL> edit database rdscdb_b set state=app ly-on; Succeeded. DGMGRL> show database rdscdb_b Database - rdscdb_b Role: PHYSICAL STANDBY Intended State: APPLY-ON Transport Lag: 0 seconds (computed 0 seconds ago) Apply Lag: 0 seconds (computed 0 seconds ago) Average Apply Rate: 35.00 KByte/s Real Time Query: ON Instance(s): RDSCDB Database Status: SUCCESS DGMGRL> show database rdscdb_b Database - rdscdb_b Role: PHYSICAL STANDBY Intended State: APPLY-ON Transport Lag: 0 seconds (computed 1 second ago) Apply Lag: 0 seconds (computed 1 second ago) Average Apply Rate: 16.00 KByte/s Real Time Query: ON Instance(s): RDSCDB </pre>	

작업	설명	필요한 기술
	<pre>Database Status: SUCCESS DGMGRL></pre>	

관련 리소스

- [Amazon RDS를 오라클 PeopleSoft 데이터베이스로 구성](#) (AWS 백서)
- [Oracle Data Guard Broker 가이드](#)(Oracle 참조 문서)
- [Oracle Data Guard Concepts 및 Administration](#)(Oracle 참조 문서)

추가 정보

데이터베이스 보호 모드 선택

Oracle Data Guard는 가용성, 보호 및 성능 요구 사항에 따라 Data Guard 환경을 구성할 수 있는 세 가지 보호 모드를 제공합니다. 다음 표에는 이러한 세 가지 모드가 요약되어 있습니다.

보호 모드	전송 설정 다시 실행	설명
최대 성능	ASYNC	<p>기본 데이터베이스에서 발생하는 트랜잭션의 경우 다시 실행 데이터가 비동기적으로 전송되어 대기 데이터베이스 다시 실행 로그에 기록됩니다. 따라서 성능에 미치는 영향은 최소화됩니다.</p> <p>MaxPerformance 은 비동기 로그 전달로 인해 RPO=0을 제공할 수 없습니다.</p>
최대한의 보호	SYNC+AFFIRM	<p>기본 데이터베이스의 트랜잭션의 경우 트랜잭션이 승인되기 전에 다시 실행 데이터가 디스크의 대기 데이터베이스 다시</p>

실행 로그에 동기적으로 전송되고 기록됩니다. 대기 데이터베이스를 사용할 수 없게 되면 기본 데이터베이스가 스스로 종료되어 트랜잭션이 보호됩니다.

최대 가용성

SYNC+AFFIRM

대기 데이터베이스로부터 승인을 받지 못하는 경우를 제외하면 이 모드는 MaxProtection 모드와 비슷합니다. 이 경우 다시 실행 스트림을 동기화된 대기 데이터베이스에 다시 쓸 수 있을 때까지 기본 데이터베이스 가용성을 보존하기 위해 MaxPerformance 모드에 있는 것처럼 작동합니다.

SYNC+NOAFFIRM

기본 데이터베이스의 트랜잭션의 경우 다시 실행이 대기 데이터베이스에 동기적으로 전송되고 기본 데이터베이스는 다시 실행이 대기 디스크에 기록된 것이 아니라 대기 데이터베이스에서 다시 실행이 수신되었다는 확인만 기다립니다. FastSync이라고도 하는 이 모드는 여러 개의 동시 오류가 발생하는 특수한 경우 데이터 손실에 노출될 수 있는 위험을 감수하면서 성능상의 이점을 제공할 수 있습니다.

RDS Custom for Oracle의 읽기 전용 복제본은 Oracle Data Guard의 기본 보호 모드이기도 한 최대 성능 보호 모드를 사용하여 생성됩니다. 최대 성능 모드는 기본 데이터베이스의 성능 영향을 최소화하므로 초 단위로 측정되는 Recovery Point Objective(RPO) 요구 사항을 충족할 수 있습니다.

데이터 손실 제로(RPO=0) 목표를 달성하기 위해 Oracle Data Guard 보호 모드를

MaxAvailability로 사용자 지정하고 다시 실행 전송을 SYNC+NOAFFIRM으로 설정하여 더 나은 성능을 얻을 수 있습니다. 기본 데이터베이스의 커밋은 해당 다시 실행 벡터가 대기 데이터베이스로 성공적으로 전송된 후에만 확인되므로 커밋에 민감한 워크로드의 경우 기본 인스턴스와 복제본 간의 네트워크 지연 시간이 매우 중요할 수 있습니다. 읽기 전용 복제본을 MaxAvailability 모드에서 실행하도록 사용자 지정했을 때 성능에 미치는 영향을 평가하려면 워크로드에 대한 부하 테스트를 수행하는 것이 좋습니다.

기본 데이터베이스와 동일한 가용 영역에 읽기 전용 복제본을 배포하면 다른 가용 영역에 읽기 전용 복제본을 배포하는 것보다 네트워크 지연 시간이 줄어듭니다. 그러나 동일한 가용 영역에 기본 복제본과 읽기 전용 복제본을 배포하는 것은 HA 요구 사항을 충족하지 못할 수 있습니다. 가용 영역을 사용할 수 없는 경우가 발생할 경우 기본 인스턴스와 읽기 전용 복제본 인스턴스가 모두 영향을 받기 때문입니다.

SQL Server 데이터베이스를 AWS의 MongoDB Atlas로 마이그레이션하기 위한 쿼리 성능 평가

제작: 바툴가 푸레브라그차 (AWS), 크리슈나쿠마르 PeerIslands 사티아나라야나 (미국), 바부 스리니 바산 (MongoDB)

환경: PoC 또는 파일럿	소스: Microsoft SQL Server	대상: MongoDB Atlas 또는 MongoDB Enterprise Advanced
R 유형: 리플랫폼	워크로드: Microsoft	기술: 데이터베이스, 마이그레이션

요약

이 패턴은 실제와 가까운 데이터를 사용하여 MongoDB를 로드하고 프로덕션 시나리오에 최대한 가까운 MongoDB 쿼리 성능을 평가하기 위한 지침을 제공합니다. 이 평가는 관계형 데이터베이스에서 MongoDB로의 마이그레이션을 계획하는 데 도움이 되는 정보를 제공합니다. 이 [PeerIslands 패턴은 테스트 데이터 생성기와 성능 분석기를 사용하여](#) 쿼리 성능을 테스트합니다.

이 패턴은 Microsoft SQL Server를 MongoDB로 마이그레이션할 때 특히 유용합니다. 스키마 변환을 수행하고 현재 SQL Server 인스턴스에서 MongoDB로 데이터를 로드하는 작업은 매우 복잡할 수 있기 때문입니다. 대신 실제 마이그레이션을 시작하기 전에 실제와 가까운 데이터를 MongoDB로 로드하고, MongoDB 성능을 이해하고, 스키마 디자인을 미세 조정할 수 있습니다.

사전 조건 및 제한 사항

사전 조건

- 활성 상태의 AWS 계정
- [MongoDB Atlas](#)에 대한 지식
- 대상 MongoDB 스키마
- 일반적인 쿼리 패턴

제한 사항

- 데이터 로드 시간과 성능은 MongoDB 클러스터 인스턴스 크기에 따라 제한됩니다. 실제 성능을 이해하려면 프로덕션 용도로 권장되는 인스턴스를 선택하는 것이 좋습니다.
- PeerIslands 테스트 데이터 생성기 및 성능 분석기는 현재 온라인 데이터 로드 및 쿼리만 지원합니다. 오프라인 일괄 처리(예: Spark 커넥터를 사용하여 MongoDB로 데이터 로드)는 아직 지원되지 않습니다.
- PeerIslands 테스트 데이터 생성기 및 성능 분석기는 컬렉션 내의 필드 관계를 지원합니다. 컬렉션 간의 관계는 지원하지 않습니다.

제품 에디션

- 이 패턴은 [MongoDB Atlas](#) 및 [MongoDB Enterprise Advanced](#)를 모두 지원합니다.

아키텍처

대상 기술 스택

- MongoDB Atlas 또는 MongoDB Enterprise Advanced

아키텍처

PeerIslands 테스트 데이터 생성기 및 성능 분석기는 Java와 Angular를 사용하여 구축되었으며, 생성된 데이터를 Amazon Elastic Block Store (Amazon EBS)에 저장합니다. 이 도구는 테스트 데이터 생성 및 성능 테스트라는 두 가지 워크플로우로 구성되어 있습니다.

- 테스트 데이터를 생성할 때는 생성해야 하는 데이터 모델의 JSON 표현인 템플릿을 생성합니다. 템플릿을 생성한 후에는 로드 생성 구성에 정의된 대로 대상 컬렉션에서 데이터를 생성할 수 있습니다.
- 성능 테스트에서 프로필을 생성합니다. 프로필은 생성, 읽기, 업데이트, 삭제(CRUD) 작업, 집계 파이프라인, 각 작업의 가중치, 각 단계의 기간을 구성할 수 있는 다단계 테스트 시나리오입니다. 프로필을 생성한 후 구성을 기반으로 대상 데이터베이스에서 성능 테스트를 실행할 수 있습니다.

PeerIslands 테스트 데이터 생성기 및 성능 분석기는 Amazon EBS에 데이터를 저장하므로 사용자는 피어링, 허용 목록, 프라이빗 엔드포인트를 포함하여 MongoDB가 지원하는 모든 연결 메커니즘을 사용하여 Amazon EBS를 MongoDB에 연결할 수 있습니다. 기본적으로 이 도구에는 운영 구성 요소가 포함되어 있지 않지만 필요한 경우 프로메테우스용 Amazon Managed Service, Amazon Managed Grafana, Amazon 및 AWS Secrets Manager를 사용하여 구성할 수 있습니다. CloudWatch

도구

- [PeerIslands 테스트 데이터 생성기와 성능 분석기](#)에는 두 가지 구성 요소가 포함되어 있습니다. Test Data Generator 구성 요소를 사용하면 MongoDB 스키마를 기반으로 고도로 고객별로 특화된 실제 데이터를 생성할 수 있습니다. 이 도구는 풍부한 데이터 라이브러리를 갖춘 완전한 UI 기반이며 MongoDB에서 수십억 개의 레코드를 빠르게 생성하는 데 사용할 수 있습니다. 이 도구는 MongoDB 스키마의 필드 간 관계를 구현하는 기능도 제공합니다. Performance Analyzer 구성 요소를 사용하면 고도로 고객별 쿼리와 집계를 생성하고 MongoDB에서 현실적인 성능 테스트를 수행할 수 있습니다. Performance Analyzer를 사용하여 다양한 로드 프로파일과 특정 사용 사례에 대한 파라미터화된 쿼리로 MongoDB 성능을 테스트할 수 있습니다.

모범 사례

다음 리소스를 참조하세요.

- [MongoDB 스키마 설계 모범 사례](#) (MongoDB 개발자 웹사이트)
- [AWS에 MongoDB Atlas를 배포하는 모범 사례](#) (MongoDB 웹사이트)
- [AWS를 사용하여 MongoDB Atlas 데이터 플레인에 애플리케이션을 안전하게 연결 \(PrivateLinkAWS 블로그 게시물\)](#)
- [MongoDB 성능 모범 사례 가이드](#) (MongoDB 웹사이트)

에픽

소스 데이터 이해

작업	설명	필요한 기술
현재 SQL Server 소스의 데이터베이스 발자국을 이해합니다.	현재 SQL Server 발자국을 이해합니다. 데이터베이스의 INFORMATION 스키마에 대해 쿼리를 실행하여 이 작업을 수행할 수 있습니다. 테이블 수와 각 테이블의 크기를 결정합니다. 각 테이블과 관련된 인덱스를 분석합니다. SQL 분석에 대한 자세한 내용은 웹 사이트	DBA

작업	설명	필요한 기술
	<p>의 블로그 게시물 SQL2Mongo : 데이터 마이그레이션 여정을 참조하십시오. PeerIslands</p>	
<p>소스 스키마를 이해합니다.</p>	<p>테이블 스키마와 데이터의 비즈니스 표현(예: 우편번호, 이름, 통화)을 결정합니다. 기존 엔터티 관계(ER) 다이어그램을 사용하거나 기존 데이터베이스에서 ER 다이어그램을 생성합니다. 자세한 내용은 웹 사이트의 블로그 게시물 SQL2Mongo : 데이터 마이그레이션 여정을 참조하십시오. PeerIslands</p>	<p>DBA</p>
<p>쿼리 패턴을 이해합니다.</p>	<p>사용하는 상위 10개 SQL 쿼리를 문서화합니다. 데이터베이스에서 사용할 수 있는 performance_schema.events_statements_summary_by_digest 테이블을 사용하여 상위 쿼리를 이해할 수 있습니다. 자세한 내용은 웹 사이트의 블로그 게시물 SQL2Mongo: 데이터 마이그레이션 여정을 참조하십시오. PeerIslands</p>	<p>DBA</p>

작업	설명	필요한 기술
SLA 약정을 이해합니다.	데이터베이스 운영에 대한 대상 서비스 수준에 관한 계약 (SLA)을 문서화합니다. 일반적인 측정값에는 쿼리 지연 시간 및 초당 쿼리가 포함됩니다. 측정값과 그 목표는 일반적으로 비기능적 요구 사항(NFR) 문서에서 확인할 수 있습니다.	DBA

MongoDB 스키마 정의

작업	설명	필요한 기술
대상 스키마를 정의합니다.	대상 MongoDB 스키마에 대한 다양한 옵션을 정의합니다. 자세한 내용은 MongoDB Atlas 설명서의 스키마 를 참조하세요. 테이블 관계를 기반으로 모범 사례와 디자인 패턴을 고려하세요. 자세한 내용은 MongoDB 설명서의 데이터 모델 예제 및 패턴 을 참조하세요.	MongoDB 엔지니어
대상 쿼리 패턴을 정의합니다.	MongoDB 쿼리 및 집계 파이프라인을 정의합니다. 이러한 쿼리는 SQL Server 워크로드에 대해 캡처한 상위 쿼리와 동일합니다. MongoDB 집계 파이프라인을 구성하는 방법을 이해하려면 MongoDB 설명서 를 참조하세요.	MongoDB 엔지니어
MongoDB 인스턴스 유형을 정의합니다.	테스트에 사용할 인스턴스의 크기를 결정합니다. 지침은	MongoDB 엔지니어

작업	설명	필요한 기술
	MongoDB 설명서 를 참조하세요.	

대상 데이터베이스를 준비합니다.

작업	설명	필요한 기술
MongoDB Atlas 클러스터를 설정합니다.	AWS에서 MongoDB 클러스터를 설정하려면 MongoDB 설명서 의 지침을 따르세요.	MongoDB 엔지니어
대상 데이터베이스에서 사용자를 생성합니다.	MongoDB 설명서 의 지침에 따라 액세스 및 네트워크 보안을 위해 MongoDB Atlas 클러스터를 구성합니다.	MongoDB 엔지니어
AWS에서 적절한 역할을 생성하고 Atlas에 대한 역할 기반 액세스 제어를 구성합니다.	필요한 경우 MongoDB 설명서 의 지침에 따라 추가 사용자를 설정합니다. AWS 역할을 통해 인증 및 권한 을 구성합니다.	MongoDB 엔지니어
MongoDB Atlas 액세스를 위한 Compass를 설정합니다.	쉽게 탐색하고 액세스할 수 있도록 MongoDB Compass GUI 유틸리티 를 설정합니다.	MongoDB 엔지니어

Test Data Generator를 사용하여 베이스 로드를 설정합니다.

작업	설명	필요한 기술
Test Data Generator를 설치합니다.	PeerIsland 테스트 데이터 생성기 를 사용자 환경에 설치하세요.	MongoDB 엔지니어

작업	설명	필요한 기술
적절한 데이터를 생성하도록 Test Data Generator기를 구성합니다.	데이터 라이브러리를 사용하여 MongoDB 스키마의 각 필드에 대한 특정 데이터를 생성하여 템플릿을 생성합니다. 자세한 내용은 MongoDB Data Generaotor & Performance Analyzer 동영상을 참조하세요.	MongoDB 엔지니어
Test Data Generator를 수평적으로 확장하여 필요한 로드를 생성합니다.	생성한 템플릿을 사용하여 필요한 병렬 처리를 구성하여 대상 컬렉션에 대한 로드 생성을 시작합니다. 필요한 데이터를 생성하기 위한 기간과 규모를 결정합니다.	MongoDB 엔지니어
MongoDB Atlas에서 로드를 검증합니다.	MongoDB Atlas에 로드된 데이터를 확인합니다.	MongoDB 엔지니어
MongoDB에서 필요한 인덱스를 생성합니다.	쿼리 패턴을 기반으로 필요에 따라 인덱스를 정의합니다. 모범 사례는 MongoDB 설명서 를 참조하세요.	MongoDB 엔지니어

성능 테스트 수행

작업	설명	필요한 기술
Performance Analyzer에서 로드 프로필을 설정합니다.	Performance Analyzer에서 특정 쿼리와 해당 가중치, 테스트 실행 기간 및 단계를 구성하여 성능 테스트 프로필을 생성합니다. 자세한 내용은 MongoDB Data Generaotor & Performan	MongoDB 엔지니어

작업	설명	필요한 기술
	ce Analyzer 동영상을 참조하세요.	
성능 테스트를 실행합니다.	생성한 성능 테스트 프로필을 사용하여 필요한 병렬 처리를 구성하여 대상 컬렉션에 대한 테스트를 시작합니다. 성능 테스트 도구를 수평적으로 확장하여 MongoDB Atlas에 대해 쿼리를 실행할 수 있습니다.	MongoDB 엔지니어
테스트 결과를 기록합니다.	쿼리에 대한 P95, P99 지연 시간을 기록합니다.	MongoDB 엔지니어
스키마와 쿼리 패턴을 조정합니다.	인덱스 및 쿼리 패턴을 수정하여 성능 문제를 해결합니다.	MongoDB 엔지니어

프로젝트 닫기

작업	설명	필요한 기술
임시 AWS 리소스를 종료합니다.	Test Data Generator and Performance Analyzer에 사용한 모든 임시 리소스를 삭제합니다.	AWS 관리자
성능 테스트 결과를 업데이트합니다.	MongoDB 쿼리 성능을 이해하고 SLA와 비교합니다. 필요한 경우 MongoDB 스키마를 미세 조정하고 프로세스를 다시 실행합니다.	MongoDB 엔지니어
프로젝트를 마칩니다.	프로젝트를 마무리하고 피드백을 제공하세요.	MongoDB 엔지니어

관련 리소스

- GitHub 리포지토리: [S3toAtlas](#)
- 스키마: [MongoDB 스키마 설계](#)
- 집계 파이프라인: [MongoDB 집계 파이프라인](#)
- MongoDB Atlas 크기 조정: [크기 조정 티어 선택](#)
- 동영상: [MongoDB Data Generator](#) 및 Performance Analyzer
- 참조: [MongoDB 설명서](#)
- 자습서: [MongoDB 개발자 가이드](#), [MongoDB Jumpstart](#)
- AWS Marketplace: [AWS Marketplace의 MongoDB Atlas](#)
- AWS 파트너 솔루션: [MongoDB Atlas on AWS Reference Deployment](#)

추가 리소스:

- [SQL 분석](#)
- [MongoDB 개발자 커뮤니티 포럼](#)
- [MongoDB 성능 튜닝 질문](#)
- [Atlas 및 Redshift를 사용한 운영 분석](#)
- [MongoDB Atlas 및 AWS Elastic Beanstalk를 사용한 애플리케이션 현대화](#)

AWS 계정 전반에 걸쳐 Amazon RDS 인스턴스의 복제를 자동화

작성자: Parag Nagwekar (AWS) 및 Arun Chandapillai (AWS)

환경: 프로덕션

기술: 데이터베이스, 서버리스,
인프라 DevOps

워크로드: 기타 모든 워크로드

AWS 서비스: Lambda,
Amazon RDS, SDK for
Python(Boto3), Step Functions
, Amazon SNS

요약

이 패턴은 Step Functions와 Lambda를 사용하여 여러 AWS 계정 전반적으로 Amazon Relational Database Service(RDS) DB 인스턴스를 복제, 추적, 롤백하는 프로세스를 자동화하는 방법을 보여줍니다. 이 자동화를 사용하면 조직의 규모와 상관없이 성능 영향이나 운영 오버헤드 없이 RDS DB 인스턴스의 대규모 복제를 수행할 수 있습니다. 또한 이 패턴을 사용하여 조직이 데이터를 여러 AWS 계정 및 AWS 리전 전반적으로 복제하고 중복하도록 요구하는 필수 데이터 거버넌스 전략 또는 규정 준수 요구 사항을 준수하도록 지원할 수 있습니다. Amazon RDS 데이터의 대규모 교차 계정 복제는 비용과 시간이 많이 소요될 수 있는 비효율적이고 오류가 발생하기 쉬운 수동 프로세스입니다. 하지만 이 패턴의 자동화는 교차 계정 복제를 안전하고 효과적이며 효율적으로 수행하는 데 도움이 될 수 있습니다.

사전 조건 및 제한 사항

사전 조건

- 두 개의 AWS 계정
- 소스 AWS 계정에서 실행되는 RDS DB 인스턴스
- 대상 AWS 계정의 RDS DB 인스턴스에 대한 서브넷 그룹
- 소스 AWS 계정에서 생성되어 대상 계정과 공유되는 Key Management Service(KMS) 키(정책 세부 정보에 대한 자세한 내용은 이 패턴의 추가 정보 섹션 참조)
- 대상 계정의 데이터베이스를 암호화하기 위한 대상 AWS 계정의 KMS 키

제품 버전

- Python 3.9(Lambda 사용)
- PostgreSQL 11.3, 13.x, and 14.x

아키텍처

기술 스택

- Amazon Relational Database Service(Amazon RDS)
- Amazon Simple Notification Service(Amazon SNS)
- AWS Key Management Service (AWS KMS)
- AWS Lambda
- AWS Secrets Manager
- AWS Step Functions

대상 아키텍처

다음 다이어그램은 Step Functions를 사용하여 소스 계정(계정 A)에서 대상 계정(계정 B)으로 RDS DB 인스턴스의 예약된 온디맨드 복제를 오케스트레이션하는 아키텍처를 보여줍니다.

소스 계정(다이어그램의 계정 A)에서 Step Functions 상태 시스템은 다음을 수행합니다.

1. 계정 A의 RDS DB 인스턴스를 통해 스냅샷을 생성합니다.
2. 계정 A에서 AWS KMS 키를 사용하여 스냅샷을 복사하고 암호화합니다. 전송 시 암호화를 보장하기 위해 스냅샷은 DB 인스턴스의 암호화 여부에 관계없이 암호화됩니다.
3. 계정 B에게 스냅샷에 대한 액세스 권한을 부여하여 계정 B와 DB 스냅샷을 공유합니다.
4. 알림을 SNS 주제로 푸시하면 SNS 주제가 계정 B에서 Lambda 함수를 호출합니다.

대상 계정(다이어그램의 계정 B)에서 Lambda 함수는 Step Functions 상태 시스템을 실행하여 다음을 오케스트레이션합니다.

1. 계정 A의 KMS 키를 사용하여 데이터를 먼저 복호화한 다음 계정 B의 KMS 키를 사용하여 데이터를 암호화하는 동안 계정 A에서 계정 B로 공유 스냅샷을 복사합니다.
2. Secrets Manager에서 보안 암호를 읽어 현재 DB 인스턴스의 이름을 캡처합니다.

3. 스냅샷에서 새 이름과 Amazon RDS용 기본 KMS 키를 사용하여 DB 인스턴스를 복원합니다.
4. 새 데이터베이스의 엔드포인트를 읽고 Secrets Manager에서 보안 암호를 새 데이터베이스 엔드포인트로 업데이트한 다음, 나중에 삭제할 수 있도록 이전 DB 인스턴스에 태그를 지정합니다.
5. 데이터베이스의 최신 N 인스턴스를 유지하고 나머지 인스턴스는 모두 삭제합니다.

도구

도구

- [Amazon Relational Database Service\(Amazon RDS\)](#)는 AWS 클라우드에서 관계형 데이터베이스를 설정, 운영 및 조정하는 데 도움이 됩니다.
- [Amazon Simple Notification Service\(Amazon SNS\)](#)를 사용하면 웹 서버 및 이메일 주소를 포함하여 게시자와 클라이언트 간의 메시지 교환을 조정하고 관리할 수 있습니다.
- [AWS는 AWS CloudFormation](#) 리소스를 설정하고, 빠르고 일관되게 프로비저닝하고, AWS 계정 및 지역 전반의 수명 주기 전반에 걸쳐 리소스를 관리할 수 있도록 지원합니다.
- [AWS Key Management Service\(AWS KMS\)](#)를 사용하면 암호화 키를 생성하고 제어하여 데이터를 보호할 수 있습니다.
- [Lambda](#)는 서버를 프로비저닝하거나 관리할 필요 없이 코드를 실행할 수 있도록 도와주는 컴퓨팅 서비스입니다. 필요할 때만 코드를 실행하며 자동으로 확장이 가능하므로 사용한 컴퓨팅 시간만큼만 비용을 지불합니다.
- [AWS SDK for Python\(Boto3\)](#)는 Python 애플리케이션, 라이브러리 또는 스크립트를 AWS 서비스와 통합하는 데 도움이 되는 소프트웨어 개발 키트입니다.
- [AWS Secrets Manager](#)를 사용하면 암호를 포함하여 코드에 하드코딩된 보안 인증을 Secrets Manager에 대한 API 호출로 대체하여 프로그래밍 방식으로 암호를 검색할 수 있습니다.
- [AWS Step Functions](#)는 Lambda 함수와 기타 AWS 서비스를 결합할 수 있는 서버리스 오케스트레이션 서비스로서 비즈니스 크리티컬 애플리케이션을 구축합니다.

코드

이 패턴의 코드는 GitHub [계정 간 RDS 복제 리포지토리에서](#) 사용할 수 있습니다.

에픽

AWS 계정 전반에 걸쳐 RDS DB 인스턴스의 복제를 단 한번의 클릭으로 자동화합니다.

작업	설명	필요한 기술
소스 계정에 CloudFormation 스택을 배포합니다.	<ol style="list-style-type: none"> 1. 소스 계정 (계정 A) 의 AWS Management 콘솔에 로그인 하고 CloudFormation 콘솔 을 엽니다. 2. 탐색 창에서 스택을 선택합니다. 3. 스택 생성을 선택한 다음 기존 리소스 사용(리소스 가져오기))를 선택합니다. 4. 리소스 식별 페이지에서 다음을 선택합니다. 5. 템플릿 지정 페이지에서 템플릿 업로드를 선택합니다. 6. [파일 선택] 을 선택하고 GitHub Crossaccount RDS 복제 리포지토리에서 Cloudformation-SourceAccountRDS.yaml 파일을 선택한 후 [다음] 을 선택합니다. 7. 스택 이름에 스택에 대한 이름을 입력합니다. 8. Parameters(파라미터) 섹션에서 스택 템플릿에서 정의된 파라미터를 다음과 같이 지정합니다. <ul style="list-style-type: none"> • 예 DestinationAccountNumber대상 RDS DB 인 	클라우드 관리자, 클라우드 아키텍트

작업	설명	필요한 기술
	<p>스턴스의 계정 번호를 입력합니다.</p> <ul style="list-style-type: none"> 의 경우 KeyName, AWS KMS 키를 입력합니다. 의 ScheduleExpression 경우 cron 표현식을 입력합니다 (기본값은 매일 오전 12시). SourceDBIdentifier에 소스 데이터의 설명을 입력합니다. SourceDB의 SnapshotName 경우 스냅샷 이름을 입력하거나 기본값을 그대로 사용합니다. <p>9. 다음을 선택합니다.</p> <p>10스택 옵션 구성 페이지에서 기본 옵션을 유지하고 다음 (Next)을 선택합니다.</p> <p>11스택 구성을 검토하고 제출을 선택합니다.</p> <p>12스택의 리소스 탭을 선택한 다음 SNS 주제의 Amazon 리소스 이름(ARN)을 기록해 둡니다.</p>	

작업	설명	필요한 기술
<p>대상 계정에 CloudFormation 스택을 배포합니다.</p>	<ol style="list-style-type: none"> 1. 대상 계정 (계정 B) 의 AWS Management 콘솔에 로그인하고 CloudFormation 콘솔 을 엽니다. 2. 탐색 창에서 스택을 선택합니다. 3. 스택 생성을 선택한 다음 기존 리소스 사용(리소스 가져오기))를 선택합니다. 4. 리소스 식별 페이지에서 다음을 선택합니다. 5. 템플릿 지정 페이지에서 템플릿 업로드를 선택합니다. 6. 파일을 선택하고 GitHub 크로스 어카운트 RDS 복제 리포지토리에서 Cloudformation-DestinationAccountRDS.yaml 파일을 선택한 후 다음을 선택합니다. 7. 스택 이름에 스택에 대한 이름을 입력합니다. 8. Parameters(파라미터) 섹션에서 스택 템플릿에서 정의된 파라미터를 다음과 같이 지정합니다. <ul style="list-style-type: none"> • DatabaseName에 데이터베이스 이름을 입력합니다. • Engine의 경우 소스 데이터베이스와 일치하는 데이터베이스 엔진 유형을 입력합니다. 	<p>클라우드 아키텍트, DevOps 엔지니어, 클라우드 관리자</p>

작업	설명	필요한 기술
	<ul style="list-style-type: none"> • InstanceClassDB의 경우 선호 데이터베이스 인스턴스 유형을 입력하거나 기본값을 그대로 사용합니다. • Subnetgroups의 경우 기존 VPC 서브넷 그룹을 입력합니다. 서브넷 그룹을 생성하는 방법에 대한 지침은 Amazon RDS 사용 설명서의 2단계: DB 서브넷 그룹 생성을 참조하십시오. • 의 경우 SecretName, 경로 및 암호 이름을 입력하거나 기본값을 그대로 사용합니다. • SGID의 경우 목적지 클러스터의 보안 그룹 ID를 입력합니다. • KMSKey의 경우 목적지 계정에 KMS 키의 ARN을 입력합니다. • 에 롤백을 위해 NoOfOlderInstances보관하려는 RDS DB 인스턴스의 이전 복사본 수를 입력합니다. <p>9. 다음을 선택합니다.</p> <p>10스택 옵션 구성 페이지에서 기본 옵션을 유지하고 다음 (Next)을 선택합니다.</p>	

작업	설명	필요한 기술
	<p>11스택 구성을 검토하고 제출을 선택합니다.</p> <p>12스택의 리소스 탭을 선택한 다음 InvokeStepFunction 의 물리적 ID 및 ARN을 기록해 둡니다.</p>	
<p>목적지 계정에서 RDS DB 인스턴스가 생성되었는지 확인합니다.</p>	<ol style="list-style-type: none"> 1. AWS Management Console 에 로그인하고 Amazon RDS 콘솔을 엽니다. 2. 탐색 창에서 데이터베이스를 선택한 다음 새 RDS DB 인스턴스가 새 클러스터 아래에 나타나는지 확인합니다. 	<p>클라우드 관리자, 클라우드 아키텍트, DevOps 엔지니어</p>

작업	설명	필요한 기술
<p>SNS 주제에 대한 Lambda 함수를 구독합니다.</p>	<p>다음의 Command Line Interface(CLI) 명령을 실행하여 소스 계정(계정 A)의 SNS 주제에 목적지 계정(계정 B)의 Lambda 함수를 구독해야 합니다.</p> <p>계정 A에서 다음 명령을 실행합니다.</p> <pre data-bbox="597 667 1026 1142">aws sns add-permission \ --label lambda-access \ --aws-account-id \ <DestinationAccount> \ --topic-arn <Arn of \ SNSTopic > \ --action-name Subscribe \ ListSubscriptionsByTopic</pre> <p>계정 B에서 다음 명령을 실행합니다.</p> <pre data-bbox="597 1297 1026 1850">aws lambda add-permission \ --function-name <Name \ of InvokeStepFunction \ > \ --source-arn <Arn of \ SNSTopic > \ --statement-id \ function-with-sns \ --action lambda:InvokeFunction \ --principal sns.amazonaws.com</pre>	<p>클라우드 관리자, 클라우드 아키텍트, DBA</p>

작업	설명	필요한 기술
	<p>계정 B에서 다음 명령을 실행합니다.</p> <pre data-bbox="597 331 1024 646">aws sns subscribe \ --protocol "lambda" \ --topic-arn <Arn of SNSTopic> \ --notification-e ndpoint <Arn of InvokeStepFunction></pre>	

작업	설명	필요한 기술
<p>소스 계정의 RDS DB 인스턴스를 목적지 계정과 동기화합니다.</p>	<p>소스 계정에서 Step Functions 상태 기계를 시작하여 온디맨드 데이터베이스 복제를 시작합니다.</p> <ol style="list-style-type: none"> 1. Step Functions 콘솔을 엽니다. 2. 탐색 창에서 상태 머신을 선택합니다. 3. 상태 머신을 선택합니다. 4. 실행 탭에서 함수를 선택한 다음 실행 시작을 선택하여 워크플로를 시작합니다. <p>참고: 일정에 따라 복제를 자동으로 실행할 수 있도록 스케줄러가 마련되어 있지만 스케줄러는 기본적으로 꺼져 있습니다. 대상 계정의 CloudFormation 스택에 있는 리소스 탭에서 스케줄러에 대한 Amazon CloudWatch 규칙의 이름을 찾을 수 있습니다. 이벤트를 수정하는 방법에 대한 지침은 사용 CloudWatch 설명서의 CloudWatch 이벤트 규칙 삭제 또는 비활성화를 참조하십시오.</p>	<p>클라우드 설계자, DevOps 엔지니어, 클라우드 관리자</p>

작업	설명	필요한 기술
필요한 경우 데이터베이스를 이전 복사본 중 하나로 롤백합니다.	<ol style="list-style-type: none"> 1. Secrets Manager 콘솔을 엽니다. 2. 암호 목록에서 이전에 CloudFormation 템플릿을 사용하여 만든 암호를 선택합니다. 애플리케이션은 보안 암호를 사용하여 대상 클러스터의 데이터베이스에 액세스합니다. 3. 세부 정보 페이지에서 보안 암호 값을 업데이트 하려면, 보안 암호 값 섹션에서 보안 암호 값 검색을 선택한 후 편집을 선택합니다. 4. 데이터베이스 엔드포인트의 세부 정보를 입력합니다. 	클라우드 관리자, DBA, 엔지니어 DevOps

관련 리소스

- [교차 리전 읽기 복제본](#)(Amazon RDS 사용 설명서)
- [블루/그린 배포](#)(Amazon RDS 사용 설명서)

추가 정보

다음 예제 정책을 사용하여 AWS 계정 전반에 걸쳐 KMS 키를 공유할 수 있습니다.

```
{
  "Version": "2012-10-17",
  "Id": "cross-account-rds-kms-key",
  "Statement": [
    {
      "Sid": "Enable user permissions",
      "Effect": "Allow",
      "Principal": {
```

```

        "AWS": "arn:aws:iam::<SourceAccount>:root"
    },
    "Action": "kms:*",
    "Resource": "*"
},
{
    "Sid": "Allow administration of the key",
    "Effect": "Allow",
    "Principal": {
        "AWS": "arn:aws:iam::<DestinationAccount>:root"
    },
    "Action": [
        "kms:Create*",
        "kms:Describe*",
        "kms:Enable*",
        "kms:List*",
        "kms:Put*",
        "kms:Update*",
        "kms:Revoke*",
        "kms:Disable*",
        "kms:Get*",
        "kms>Delete*",
        "kms:ScheduleKeyDeletion",
        "kms:CancelKeyDeletion"
    ],
    "Resource": "*"
},
{
    "Sid": "Allow use of the key",
    "Effect": "Allow",
    "Principal": {
        "AWS": [
            "arn:aws:iam::<DestinationAccount>:root",
            "arn:aws:iam::<SourceAccount>:root"
        ]
    },
    "Action": [
        "kms:Encrypt",
        "kms:Decrypt",
        "kms:ReEncrypt*",
        "kms:GenerateDataKey*",
        "kms:DescribeKey",
        "kms:CreateGrant"
    ],

```

```
    "Resource": "*"
  }
]
}
```

Systems Manager를 사용하여 SAP HANA 데이터베이스를 자동으로 백업하고 EventBridge

작성자: Ambarish Satarkar(AWS) 및 Gaurav Rath(AWS)

코드 리포지토리: HDB_BACKUP_SSM_Document	환경: 프로덕션	기술: 데이터베이스, 스토리지 및 백업
워크로드: SAP	AWS 서비스: 아마존 EC2, 아마존, 아마존 EventBridge S3, AWS Systems Manager	

요약

이 패턴은 AWS Systems Manager, 아마존 EventBridge, 아마존 심플 스토리지 서비스 (Amazon S3) 및 SAP HANA용 AWS Backint Agent를 사용하여 SAP HANA 데이터베이스 백업을 자동화하는 방법을 설명합니다.

이 패턴은 BACKUP DATA 명령을 사용하는 셸 스크립트 기반 접근 방식을 제공하므로 수많은 시스템에서 각 운영 체제(OS) 인스턴스에 대한 스크립트와 작업 구성을 유지 관리할 필요가 없습니다.

참고: 2023년 4월부터 AWS Backup은 Amazon Elastic Compute Cloud(Amazon EC2)에서 SAP HANA 데이터베이스를 지원한다고 발표했습니다. 자세한 내용은 [Amazon EC2 인스턴스 백업의 SAP HANA 데이터베이스](#)를 참조하십시오.

조직의 필요에 따라 AWS Backup 서비스를 사용하여 SAP HANA 데이터베이스를 자동으로 백업하거나 이 패턴을 사용할 수 있습니다.

사전 조건 및 제한 사항

사전 조건

- Systems Manager에 대해 구성된 관리형 Amazon Elastic Compute Cloud(Amazon EC2) 인스턴스에서 실행 중인 상태로 릴리스가 지원되는 기존 SAP HANA 인스턴스

- Systems Manager Agent(SSM Agent) 2.3.274.0 이상 설치
- 퍼블릭 액세스가 활성화되지 않은 S3 버킷
- SYSTEM로 명명된 hdbuserstore 키
- 일정에 따라 Automation 런북을 실행하기 위한 AWS Identity and Access Management(IAM) 역할
- AmazonSSMManagedInstanceCore 및 ssm:StartAutomationExecution 정책은 Systems Manager 자동화 서비스 역할에 연결됩니다.

제한 사항

- SAP HANA용 AWS Backint Agent는 중복 제거를 지원하지 않습니다.
- SAP HANA용 AWS Backint Agent는 데이터 압축을 지원하지 않습니다.

제품 버전

AWS Backint Agent 는 다음 운영 체제에서 지원됩니다.

- SUSE Linux Enterprise Server
- SUSE Linux Enterprise Server for SAP
- Red Hat Enterprise Linux for SAP

AWS Backint Agent는 다음 데이터베이스를 지원합니다.

- SAP HANA 1.0 SP12(단일 노드 및 다중 노드)
- SAP HANA 2.0 이상(단일 노드 및 다중 노드)

아키텍처

대상 기술 스택

- AWS Backint Agent
- Amazon S3
- AWS Systems Manager
- 아마존 EventBridge
- SAP HANA

대상 아키텍처

다음 다이어그램은 AWS Backint Agent, S3 버킷 및 Systems EventBridge Manager를 설치하고 명령 문서를 사용하여 정기 백업을 예약하는 설치 스크립트를 보여줍니다.

자동화 및 규모 조정

- Systems Manager Automation 런북을 사용하여 여러 AWS Backint Agent를 설치할 수 있습니다.
- Systems Manager 런북을 실행할 때마다 대상 선택에 따라 n개의 SAP HANA 인스턴스로 규모를 조정할 수 있습니다.
- EventBridge SAP HANA 백업을 자동화할 수 있습니다.

도구

- [SAP HANA용 AWS Backint Agent](#)는 기존 워크플로와 통합되어 구성 파일에 지정된 S3 버킷에 SAP HANA 데이터베이스를 백업하는 독립 실행형 애플리케이션입니다. AWS Backint Agent는 SAP HANA 데이터베이스의 전체, 증분 및 차등 백업을 지원합니다. 백업 및 카탈로그가 SAP HANA 데이터베이스에서 AWS Backint Agent로 전송되는 SAP HANA 데이터베이스 서버에서 실행됩니다.
- [EventBridgeAmazon](#)은 애플리케이션을 다양한 소스의 데이터와 연결하는 데 사용할 수 있는 서버리스 이벤트 버스 서비스입니다. EventBridge 애플리케이션, 서비스형 소프트웨어 (SaaS) 애플리케이션 및 AWS 서비스의 실시간 데이터 스트림을 AWS Lambda 함수, API 대상을 사용하는 HTTP 호출 엔드포인트 또는 다른 계정의 이벤트 버스와 같은 대상으로 전달합니다.
- [Amazon Simple Storage Service\(S3\)](#)는 객체 스토리지 서비스입니다. Amazon S3를 사용하면 인터넷을 통해 언제 어디서든 원하는 양의 데이터를 저장하고 검색할 수 있습니다.
- [AWS Systems Manager](#)는 AWS 인프라를 확인하고 제어할 수 있도록 지원합니다. Systems Manager 콘솔을 사용하여 여러 AWS 서비스의 운영 데이터를 보고 AWS 리소스에서 운영 작업을 자동화할 수 있습니다.

코드

이 패턴의 코드는 리포지토리에서 사용할 수 있습니다. [aws-backint-automated-backup](#) GitHub

에픽

hdbuserstore 키 시스템 생성

작업	설명	필요한 기술
<p>hdbuserstore 키를 생성하십시오.</p>	<ol style="list-style-type: none"> 1. <code>/usr/sap/<SID>/HDB <Inst No>/exe</code> 로 이동합니다. 2. SAP HANA 데이터베이스 인스턴스 번호로 XX를 사용하여 다음 명령을 실행합니다. <div style="border: 1px solid #ccc; border-radius: 10px; padding: 10px; margin: 10px 0;"> <pre>hdbuserstore -i set SYSTEM <hostname >:3XX13@SYSTEMDB SYSTEM</pre> </div> <p>예를 들어, 인스턴스 번호 00가 있는 SAP HANA 호스트 saphanadb 의 경우 다음 명령을 실행합니다.</p> <div style="border: 1px solid #ccc; border-radius: 10px; padding: 10px; margin: 10px 0;"> <pre>hdbuserstore -i set SYSTEM saphanadb :30013@SYSTEMDB SYSTEM</pre> </div>	<p>AWS 관리자, SAP HANA 관리자</p>

AWS Backint Agent 설치

작업	설명	필요한 기술
<p>AWS Backint Agent 설치.</p>	<p>AWS Backint Agent 설명서에서 SAP HANA용 AWS Backint</p>	<p>AWS 관리자, SAP HANA 관리자</p>

작업	설명	필요한 기술
	Agent 설치 및 구성 의 지침을 따르십시오.	

Systems Manager 명령 문서 생성

작업	설명	필요한 기술
Systems Manager 명령 문서를 생성하십시오.	<ol style="list-style-type: none"> 1. AWS Management Console에 로그인한 후 AWS Systems Manager 콘솔을 엽니다. 2. 문서를 선택하고 내 소유를 선택합니다. 3. SAP HANA 데이터베이스와 동일한 AWS 리전에 있는지 확인하십시오. 4. 문서 생성, 명령 또는 세션을 선택하여 문서를 생성합니다. 5. 공백 없이 고유하고 설명이 포함된 이름을 사용합니다 (예: SAP HANA-backup). 6. 문서 유형이 명령 문서로 설정되어 있는지 확인하십시오. 7. 콘텐츠 헤더 아래에 몇 가지 샘플 코드가 있습니다. JSON 코드 유형을 선택하고 코드를 GitHub 리포지토리에 있는 HDB_Backup_SSM_Document.json 파일의 코드로 바꿔야 합니다. 	AWS 관리자, SAP HANA 관리자

작업	설명	필요한 기술
	8. 문서 생성을 선택합니다. 9. 내 소유 섹션에서 문서를 확인하십시오.	

정기적인 빈도로 백업 일정 잡기

작업	설명	필요한 기술
Amazon을 사용하여 정기 백업을 EventBridge 예약하십시오.	<ol style="list-style-type: none"> 1. Amazon EventBridge 콘솔을 열고 [규칙] 을 선택한 다음 [규칙 생성] 을 선택합니다. 2. 규칙 세부 정보 정의 화면에서 규칙의 고유한 이름과 설명을 입력하고 기본 이벤트 버스를 사용합니다. 3. 규칙 유형에서 일정을 선택하고 다음을 선택합니다. 4. 그런 다음 일정 정의 화면에서 필요한 빈도에 따라 적절한 일정 패턴과 cron 또는 rate 표현식을 선택합니다. 5. 대상 선택 화면에서 대상 유형으로 AWS 서비스를 선택합니다. 대상 선택에서 Systems Manager Run Command를 선택합니다. 6. 이전에 생성한 문서를 선택합니다. 7. 대상 키 및 대상 값에서 인스턴스 ID를 입력합니다. 태그 이름과 태그 값을 사용하여 	AWS 관리자, SAP HANA 관리자

작업	설명	필요한 기술
	<p>여러 인스턴스를 추가할 수 있습니다.</p> <p>8. 자동화 매개변수 구성에서 증분 또는 차등 백업을 위한 상수를 선택합니다. 전체 백업을 원하면 매개 변수 없음을 선택합니다.</p> <p>9. 새 역할을 생성할지 또는 기존 역할을 사용할지 선택합니다. 기존 역할을 사용하는 경우, 대상을 간접 호출하는데 필요한 정책이 있는지 확인합니다.</p> <p>10. 기본 추가 설정을 유지하고 Next를 선택합니다.</p> <p>11. 태그 구성 화면은 선택 사항입니다. 다음을 선택합니다.</p> <p>12. 검토 및 생성 화면에서 규칙 설정을 검토하고 생성을 선택합니다. 규칙이 성공적으로 생성되어야 합니다.</p> <p>S3 버킷 경로에서 백업 성공을 확인할 수 있습니다.</p> <pre>s3: /<your_bucket_name>/<target folder>/<SID>/usr/sap/<SID>/SYS/global/hdb/backint/DB_<SID>/</pre> <p>SAP HANA 백업 카탈로그에서 백업을 확인할 수도 있습니다.</p>	

관련 리소스

- [SAP HANA용 AWS Backint Agent](#)
- [SAP HANA용 AWS Backint Agent 설치 및 구성](#)

Cloud Custodian을 사용하여 Amazon RDS에 대한 공개 액세스 차단하기

작성자: abhay kumar (AWS) 및 Dwarika Patra (AWS)

환경: 프로덕션

기술: 데이터베이스, 보안, 자격 증명, 규정 준수

워크로드: 기타 모든 워크로드, 오픈 소스

AWS 서비스: Amazon RDS

요약

많은 조직이 여러 클라우드 공급업체에서 워크로드와 서비스를 실행합니다. 이러한 하이브리드 클라우드 환경에서 클라우드 인프라에는 개별 클라우드 공급자가 제공하는 보안 외에도 엄격한 클라우드 거버넌스가 필요합니다. Amazon Relational Database Service(Amazon RDS)와 같은 클라우드 데이터베이스는 액세스 및 권한 취약성을 모니터링해야 하는 중요한 서비스입니다. 보안 그룹을 구성하여 Amazon RDS 데이터베이스에 대한 액세스를 제한할 수 있지만, 두 번째 보호 계층을 추가하여 공개 액세스와 같은 작업을 금지할 수 있습니다. 공개 액세스가 차단되도록 하면 일반 데이터 보호 규정(GDPR), 건강보험 이전과 책임에 관한 법(HIPAA), 미국 국립표준기술연구소(NIST), 지불 카드 보안 표준(PCI DSS) 준수에 도움이 됩니다.

Cloud Custodian은 Amazon Web Services (AWS) 리소스에 대한 액세스 제한을 적용하는 데 사용할 수 있는 오픈 소스 규칙 엔진입니다. Cloud Custodian을 사용하면 정의된 보안 및 규정 준수 표준에 따라 환경을 검증하는 규칙을 설정할 수 있습니다. Cloud Custodian을 사용하면 보안 정책, 태그 정책, 미사용 리소스의 쓰레기 수집 및 비용 관리를 준수하도록 지원하여 클라우드 환경을 관리할 수 있습니다. Cloud Custodian을 사용하면 단일 인터페이스를 사용하여 하이브리드 클라우드 환경에서 거버넌스를 구현할 수 있습니다. 예를 들어 Cloud Custodian 인터페이스를 사용하여 AWS 및 Microsoft Azure와 상호 작용하여 AWS Config, AWS 보안 그룹 및 Azure 정책과 같은 메커니즘을 사용하는 수고를 줄일 수 있습니다.

이 패턴은 AWS 기반 Cloud Custodian을 사용하여 Amazon RDS 인스턴스에 대한 공개 액세스 제한을 적용하기 위한 지침을 제공합니다.

사전 조건 및 제한 사항

사전 조건

- 활성 상태의 AWS 계정.
- [키 쌍](#)
- AWS Lambda가 설치되었습니다.

아키텍처

대상 기술 스택

- Amazon RDS
- AWS CloudTrail
- AWS Lambda
- Cloud Custodian

대상 아키텍처

다음 다이어그램은 클라우드 관리자가 Lambda에 정책을 배포하고, AWS가 이벤트를 CloudTrail 시작하고, Amazon RDS에서 Lambda 함수를 `CreateDBInstance` `false`로 설정하는 것을 보여줍니다. `PubliclyAccessible`

도구

서비스

- [AWS는 AWS](#) 계정의 거버넌스, 규정 준수 및 운영 위험을 감사할 수 있도록 CloudTrail 지원합니다.
- [AWS Command Line Interface\(AWS CLI\)](#)는 명령줄 셸에서 명령을 사용하여 AWS 서비스와 상호 작용할 수 있는 오픈 소스 도구입니다.
- [AWS Identity and Access Management\(IAM\)](#)를 사용하면 사용자에 대해 인증 및 권한 부여를 제어함으로써 AWS 리소스에 대한 액세스를 안전하게 관리할 수 있습니다.
- [AWS Lambda](#)는 서버를 프로비저닝하거나 관리할 필요 없이 코드를 실행하는 데 도움이 되는 컴퓨팅 서비스입니다. 필요할 때만 코드를 실행하며 자동으로 확장이 가능하므로 사용한 컴퓨팅 시간만큼만 비용을 지불합니다.
- [Amazon Relational Database Service\(RDS\)](#)는 AWS 클라우드에서 관계형 데이터베이스를 설정, 운영 및 조정하는 데 도움이 됩니다.

기타 도구

- [Cloud Custodian](#)은 많은 조직이 공개 클라우드 계정을 관리하는 데 사용하는 도구와 스크립트를 하나의 오픈소스 도구로 통합합니다. 정책 정의 및 시행에 스테이트리스 규칙 엔진을 사용하고 클라우드 인프라에 대한 지표, 구조화된 결과, 세부 보고 기능을 제공합니다. 서버리스 런타임과 긴밀하게 통합되어 운영 오버헤드를 낮추면서 실시간 수정 및 대응을 제공합니다.

에픽

AWS CLI 설정

작업	설명	필요한 기술
AWS CLI를 설치합니다.	AWS CLI를 설치하려면 AWS 설명서 의 지침을 따르세요.	AWS 관리자
AWS 보안 인증 설정	<p>AWS Region 및 사용하려는 출력 형식을 포함하여 AWS CLI가 AWS와 상호 작용하는 데 사용하는 설정을 구성합니다.</p> <pre> \$>aws configure AWS Access Key ID [None]: <your_access_key_id> AWS Secret Access Key [None]: <your_secret_access_key> Default region name [None]: Default output format [None]: </pre> <p>자세한 내용은 AWS 설명서를 참조하십시오.</p>	AWS 관리자
IAM 역할을 생성합니다.	Lambda 실행 역할을 사용하여 IAM 역할을 생성하려면 다음 명령을 실행합니다.	AWS DevOps

작업	설명	필요한 기술
	<pre>aws iam create-role -- role-name lambda-ex -- assume-role-policy- document '{"Version": "2012-10-17", "Stat ement": [{ "Effect": "Allow", "Principal": {"Service": "lambda.a mazonaws.com"}, "Action": "sts:Assu meRole"}]}'</pre>	

Cloud Custodian 설정

작업	설명	필요한 기술
<p>Cloud Custodian을 설치합니다.</p>	<p>운영 체제 및 환경에 맞게 Cloud Custodian을 설치하려면 Cloud Custodian 설명서의 지침을 따르십시오.</p>	<p>DevOps 엔지니어</p>
<p>Cloud Custodian 스키마를 확인합니다.</p>	<p>정책을 실행할 수 있는 Amazon RDS 리소스의 전체 목록을 보려면 다음 명령을 사용하십시오.</p> <pre>custodian schema aws.rds</pre>	<p>DevOps 엔지니어</p>
<p>Cloud Custodian 정책을 생성합니다.</p>	<p>YAML 확장을 사용하여 추가 정보 섹션의 Cloud Custodian 정책 파일에 있는 코드를 저장합니다.</p>	<p>DevOps 엔지니어</p>
<p>공개적으로 액세스할 수 있는 플래그를 변경하기 위한 Cloud Custodian 작업을 정의합니다.</p>	<ol style="list-style-type: none"> 관리자 코드(예: /Users/abcd/custodian/ 	<p>DevOps 엔지니어</p>

작업	설명	필요한 기술
	<p>lib/python3.9/site-packages/c7n/resources/rds.py)를 찾습니다.</p> <p>2. 추가 정보 섹션의 c7n 리소스 rds.py 파일에 있는 코드를 사용하여 rds.py에서 RDSSetPublicAvailability 클래스를 찾고 이 클래스를 수정합니다.</p>	
모의 실행을 수행합니다.	<p>(선택 사항) 리소스에 대한 작업을 실행하지 않고 정책으로 식별되는 리소스를 확인하려면 다음 명령을 사용합니다.</p> <pre>custodian run -dryrun <policy_name>.yaml -s <output_directory></pre>	DevOps 엔지니어

정책 배포하기

작업	설명	필요한 기술
Lambda를 사용하여 정책을 배포합니다.	<p>정책을 실행할 Lambda 함수를 생성하려면 다음 명령을 사용하십시오.</p> <pre>custodian run -s policy.yaml</pre> <p>그러면 AWS CloudTrail CreateDBInstance 이벤트가 이 정책을 시작합니다.</p>	DevOps 엔지니어

작업	설명	필요한 기술
	그 결과, AWS Lambda는 기존에 맞는 인스턴스에 대해 공개적으로 액세스할 수 있는 플래그를 거짓으로 설정합니다.	

관련 리소스

- [AWS Lambda](#)
- [Amazon RDS](#)
- [Cloud Custodian](#)

추가 정보

Cloud Custodian 정책 YAML 파일

```

policies:
  - name: "block-public-access"
    resource: rds
    description: |
      This Enforcement blocks public access for RDS instances.
    mode:
      type: cloudtrail
    events:
      - event: CreateDBInstance # Create RDS instance cloudtrail event
        source: rds.amazonaws.com
        ids: requestParameters.dbInstanceIdentifier
    role: arn:aws:iam::1234567890:role/Custodian-compliance-role
    filters:
      - type: event
        key: 'detail.requestParameters.publiclyAccessible'
        value: true
    actions:
      - type: set-public-access
        state: false

```

c7n 리소스 rds.py 파일

```
@actions.register('set-public-access')
class RDSSetPublicAvailability(BaseAction):

    schema = type_schema(
        "set-public-access",
        state={'type': 'boolean'})
    permissions = ('rds:ModifyDBInstance',)

    def set_accessibility(self, r):
        client = local_session(self.manager.session_factory).client('rds')
        waiter = client.get_waiter('db_instance_available')
        waiter.wait(DBInstanceIdentifier=r['DBInstanceIdentifier'])
        client.modify_db_instance(
            DBInstanceIdentifier=r['DBInstanceIdentifier'],
            PubliclyAccessible=self.data.get('state', False))

    def process(self, rds):
        with self.executor_factory(max_workers=2) as w:
            futures = {w.submit(self.set_accessibility, r): r for r in rds}
            for f in as_completed(futures):
                if f.exception():
                    self.log.error(
                        "Exception setting public access on %s \n %s",
                        futures[f]['DBInstanceIdentifier'], f.exception())

        return rds
```

Security Hub 통합

Cloud Custodian을 [AWS Security Hub](#)와 통합하여 보안 조사사 결과를 전송하고 수정 조치를 시도할 수 있습니다. 자세한 내용은 [AWS Security Hub와의 클라우드 관리 통합 발표](#)를 참고하십시오.

AWS 기반 SQL Server의 Always On 가용성 그룹에서 읽기 전용 라우팅 구성

작성자: Subhani Shaik(AWS)

환경: PoC 또는 파일럿

기술: 데이터베이스, 인프라

워크로드: Microsoft

AWS 서비스: AWS Managed
Microsoft AD; Amazon EC2

요약

이 패턴은 읽기 전용 워크로드를 기본 복제본에서 보조 복제본으로 넘겨서 SQL Server Always On에서 예비 보조 복제본을 사용하는 방법을 다룹니다.

데이터베이스 미러링에는 one-to-one 매핑이 있습니다. 보조 데이터베이스를 직접 읽을 수 없으므로 스냅샷을 만들어야 합니다. Always On 가용성 그룹 기능은 Microsoft SQL Server 2012에 도입되었습니다. 이후 버전에서는 읽기 전용 라우팅을 비롯한 주요 기능이 도입되었습니다. Always On 가용성 그룹에서는 복제본 모드를 읽기 전용으로 변경하여 보조 복제본에서 직접 데이터를 읽을 수 있습니다.

Always On 가용성 그룹 솔루션은 고가용성(HA), 재해 복구(DR) 및 데이터베이스 미러링의 대안을 지원합니다. Always On 가용성 그룹은 데이터베이스 수준에서 작동하며 사용자 데이터베이스 집합의 가용성을 극대화합니다.

SQL Server는 읽기 전용 라우팅 메커니즘을 사용하여 들어오는 읽기 전용 연결을 보조 읽기 전용 복제본으로 리디렉션합니다. 이렇게 하려면 연결 문자열에 다음 파라미터와 값을 추가해야 합니다.

- ApplicationIntent=ReadOnly
- Initial Catalog=<database name>

사전 조건 및 제한 사항

사전 조건

- Virtual Private Cloud(VPC), 두 개의 가용 영역, 프라이빗 서브넷 및 보안 그룹을 갖춘 활성 AWS 계정

- [Windows Server Failover Clustering\(WSFC\)](#)이 인스턴스 수준에 구성되었으며 프라이머리 노드 (WSFCNODE1)와 tagechtalk.com이라고 이름이 지정된 Microsoft Active Directory 디렉터리용 AWS Directory Service의 일부인 세컨더리 노드(WSFCNODE2) 사이의 SQL Server 수준에 Always On 가용성 그룹이 구성된 [SQL Server 2019 Enterprise Edition Amazon Machine Image](#)가 탑재된 Amazon Elastic Compute Cloud(AmazonEC2) 시스템 두 개
- 보조 복제본에 read-only를 허용하도록 구성된 하나 이상의 노드
- Always On 가용성 그룹에 대하여 이름이 SQLAG1인 리스너
- 두 노드에서 동일한 서비스 계정으로 실행되는 SQL Server 데이터베이스 엔진
- SQL Server Management Studio(SSMS)
- 이름이 test인 테스트 데이터베이스

제품 버전

- SQL 2014 이상

아키텍처

대상 기술 스택

- Amazon EC2
- AWS 관리형 Microsoft AD
- Amazon FSx

대상 아키텍처

다음 다이어그램은 Always On 가용성 그룹(AG) 리스너가 연결에 ApplicationIntent 파라미터가 포함된 쿼리를 적절한 보조 노드로 리디렉션하는 방법을 보여줍니다.

1. Always On 가용성 그룹 리스너로 요청이 전송됩니다.
2. 연결 문자열에 ApplicationIntent 파라미터가 없으면 해당 요청이 기본 인스턴스에 전송됩니다.
3. 연결 문자열에 ApplicationIntent=ReadOnly가 포함되어 있는 경우 요청은 읽기 전용 라우팅 구성을 사용하는 보조 인스턴스, 즉 Always On 가용성 그룹이 있는 WSFC로 전송됩니다.

도구

서비스

- [Microsoft Active Directory용 AWS Directory Service](#)를 사용하면 디렉터리 인식 워크로드와 AWS 리소스가 AWS 클라우드에서 Microsoft Active Directory를 사용할 수 있습니다.
- [Amazon Elastic Compute Cloud\(Amazon EC2\)](#)는 AWS 클라우드에서 규모를 조정할 수 있는 컴퓨팅 용량을 제공합니다. 필요한 만큼 가상 서버를 시작하고 빠르게 스케일 업하거나 스케일 다운할 수 있습니다.
- [Amazon FSx](#)는 업계 표준 연결 프로토콜을 지원하고 AWS 리전 전반에 걸쳐고가용성 및 복제를 제공하는 파일 시스템을 제공합니다.

기타 서비스

- SQL Management Studio(SSMS)는 SQL Server 인스턴스를 연결, 관리 및 집행하기 위한 도구입니다.
- sqlcmd는 명령줄 유틸리티입니다.

모범 사례

Always On 가용성 그룹에 대한 자세한 내용은 [SQL Server 설명서](#)를 참조하세요.

에픽

읽기 전용 라우팅 설정

작업	설명	필요한 기술
읽기 전용에 복제본을 업데이트합니다.	기본 복제본과 보조 복제본을 모두 읽기 전용으로 업데이트하려면 SSMS에서 기본 복제본에 연결하고 추가 정보 섹션에서 1단계 코드를 실행합니다.	DBA
라우팅 URL을 생성합니다.	두 복제본의 라우팅 URL을 생성하려면 추가 정보 섹션에서 2	DBA

작업	설명	필요한 기술
	단계 코드를 실행합니다. 이 코드에서 tagechtalk.com 은 AWS Managed Microsoft AD 디렉터리의 이름입니다.	
라우팅 목록을 생성합니다.	두 복제본의 라우팅 목록을 생성하려면 추가 정보 섹션에서 3 단계 코드를 실행합니다.	DBA
라우팅 목록 검증합니다.	SQL Server Management Studio에서 기본 인스턴스에 연결하고 추가 정보 섹션의 4단계 코드를 실행하여 라우팅 목록을 검증합니다.	DBA

읽기 전용 라우팅 테스트

작업	설명	필요한 기술
ApplicationIntent 파라미터를 사용하여 연결합니다.	<ol style="list-style-type: none"> SSMS에서 Always On 가용성 그룹 리스너 이름을 ApplicationIntent=ReadOnly;Initial Catalog=test 로 연결합니다. 보조 복제본과 연결이 설정됩니다. 이를 테스트하려면 다음 명령을 실행하여 연결된 Server 이름을 표시합니다. <pre>SELECT SERVERPROPERTY('ComputenamePhysicalNetBios')</pre>	DBA

작업	설명	필요한 기술
	<p>출력에는 현재 보조 복제본 이름(WSFNODE2)이 표시됩니다.</p>	
<p>장애 조치를 수행합니다.</p>	<ol style="list-style-type: none"> 1. SSMS에서 Always On 가용성 그룹 리스너 이름에 연결합니다. 2. 기본 데이터베이스와 보조 데이터베이스가 데이터 손실 없이 동기화되어 있는지 확인합니다. 3. 현재 기본 복제본이 보조 복제본이 되고 보조 복제본이 기본 복제본이 되도록 장애 조치를 수행합니다. 4. SSMS에서 Always On 가용성 그룹 리스너 이름을 ApplicationIntent=ReadOnly;InitialCatalog=test 로 연결합니다. 5. 보조 복제본과 연결이 설정됩니다. 이를 테스트하려면 다음 명령을 실행하여 연결된 서버 이름을 표시합니다. <div data-bbox="630 1486 1029 1646" style="border: 1px solid #ccc; border-radius: 10px; padding: 10px; margin: 10px 0;"> <pre>SELECT SERVERPROPERTY('ComputerNamePhysicalNetBios')</pre> </div> <p>현재 보조 복제본 이름(WSFNODE1)이 표시됩니다.</p>	<p>DBA</p>

sqlcmd 명령줄 유틸리티를 사용하여 연결

작업	설명	필요한 기술
sqlcmd를 사용하여 연결합니다.	<p>sqlcmd에서 연결하려면 명령 프롬프트의 추가 정보 섹션에서 5단계 코드를 실행합니다. 연결된 후 다음 명령을 실행하여 연결된 서버 이름을 표시합니다.</p> <pre>SELECT SERVERPROPERTY('ComputerNamePhysicalNetBios') .</pre> <p>출력에는 현재 보조 복제본 이름(WSFENODE1)이 표시됩니다.</p>	DBA

문제 해결

문제	Solution
리스너 생성이 실패하면 'WSFC 클러스터가 네트워크 이름 리소스를 온라인 상태로 가져올 수 없습니다'라는 메시지가 표시됩니다.	자세한 내용은 Microsoft 블로그 게시물 'WSFC 클러스터가 네트워크 이름 리소스를 온라인 상태로 가져올 수 없습니다' 라는 메시지로 리스너 생성 실패를 참조하세요.
기타 리스너 문제나 네트워크 액세스 문제를 비롯한 잠재적 문제.	Microsoft 설명서의 Always On 가용성 그룹 구성 문제 해결(SQL Server) 을 참조하세요.

관련 리소스

- [Always On 가용성 그룹에 대한 읽기 전용 라우팅을 구성합니다.](#)
- [Always On 가용성 그룹 구성 문제 해결\(SQL Server\)](#)

추가 정보

단계 1. 복제본을 읽기 전용으로 업데이트

```
ALTER AVAILABILITY GROUP [SQLAG1] MODIFY REPLICA ON N'WSFCNODE1' WITH (SECONDARY_ROLE
(ALLOW_CONNECTIONS = READ_ONLY))
GO
ALTER AVAILABILITY GROUP [SQLAG1] MODIFY REPLICA ON N'WSFCNODE2' WITH (SECONDARY_ROLE
(ALLOW_CONNECTIONS = READ_ONLY))
GO
```

단계 2. 라우팅 URL 생성

```
ALTER AVAILABILITY GROUP [SQLAG1] MODIFY REPLICA ON N'WSFCNODE1' WITH (SECONDARY_ROLE
(READ_ONLY_ROUTING_URL = N'TCP://WSFCNode1.tagechtalk.com:1433'))
GO
ALTER AVAILABILITY GROUP [SQLAG1] MODIFY REPLICA ON N'WSFCNODE2' WITH (SECONDARY_ROLE
(READ_ONLY_ROUTING_URL = N'TCP://WSFCNode2.tagechtalk.com:1433'))
GO
```

단계 3. 라우팅 목록 생성

```
ALTER AVAILABILITY GROUP [SQLAG1] MODIFY REPLICA ON N'WSFCNODE1' WITH
(PRIMARY_ROLE(READ_ONLY_ROUTING_LIST=('WSFCNODE2', 'WSFCNODE1')));
GO
ALTER AVAILABILITY GROUP [SQLAG1] MODIFY REPLICA ON N'WSFCNODE2' WITH (PRIMARY_ROLE
(READ_ONLY_ROUTING_LIST=('WSFCNODE1', 'WSFCNODE2')));
GO
```

단계 4. 라우팅 목록 검증

```
SELECT AGSrc.replica_server_name AS PrimaryReplica, AGRepl.replica_server_name AS
ReadOnlyReplica, AGRepl.read_only_routing_url AS RoutingURL , AGRL.routing_priority
AS RoutingPriority FROM sys.availability_read_only_routing_lists AGRL INNER JOIN
sys.availability_replicas AGSrc ON AGRL.replica_id = AGSrc.replica_id INNER JOIN
sys.availability_replicas AGRepl ON AGRL.read_only_replica_id = AGRepl.replica_id
INNER JOIN sys.availability_groups AV ON AV.group_id = AGSrc.group_id ORDER BY
PrimaryReplica
```

단계 5. SQL 명령 유틸리티

```
sqlcmd -S SQLAG1,1433 -E -d test -K ReadOnly
```

pgAdmin에서 SSH 터널을 사용하여 연결

작성자: Jeevan Shetty(AWS) 및 Bhanu Ganesh Gudivada(AWS)

환경: 프로덕션

기술: 데이터베이스, 보안, 자격 증명, 규정 준수

워크로드: 오픈 소스

AWS 서비스: Amazon RDS, Amazon Aurora

요약

보안상의 이유로 데이터베이스는 항상 프라이빗 서브넷에 배치하는 것이 좋습니다. Amazon Web Services(AWS) 클라우드의 퍼블릭 서브넷에 있는 Amazon Elastic Compute Cloud(Amazon EC2) Bastion Host를 통해 연결하여 데이터베이스에 대한 쿼리를 실행할 수 있습니다. 이를 위해서는 개발자나 데이터베이스 관리자가 일반적으로 사용하는 pGAdmin 또는 DBeaver와 같은 소프트웨어를 Amazon EC2 호스트에 설치해야 합니다.

Linux 서버에서 PgAdmin을 실행하고 웹 브라우저를 통해 액세스하려면 추가 종속성 설치, 권한 설정 및 구성이 필요합니다.

대체 솔루션으로는 개발자 또는 데이터베이스 관리자가 PgAdmin을 사용하여 로컬 시스템에서 SSH 터널을 활성화하여 PostgreSQL 데이터베이스에 연결할 수 있습니다. 이 접근 방식에서 PgAdmin은 데이터베이스에 연결하기 전에 퍼블릭 서브넷의 Amazon EC2 호스트를 중간 호스트로 사용합니다. 아키텍처 섹션의 다이어그램은 설정을 보여줍니다.

참고: PostgreSQL 데이터베이스에 연결된 보안 그룹이 Amazon EC2 호스트의 포트 5432를 통한 연결을 허용하는지 확인하십시오.

사전 조건 및 제한 사항

사전 조건

- 기존 AWS 계정
- 퍼블릭 서브넷 하나와 인스턴스의 프라이빗 서브넷을 사용하는 Virtual Private Cloud(VPC)
- 보안 그룹이 연결된 EC2 인스턴스

- 보안 그룹이 연결된 Amazon Aurora PostgreSQL-Compatible Edition 데이터베이스
- 터널 설정을 위한 Secure Shell(SSH) 키 페어

제품 버전

- PgAdmin 버전 6.2+
- Amazon Aurora PostgreSQL-Compatible Edition 버전 12.7+

아키텍처

대상 기술 스택

- Amazon EC2
- Amazon Aurora PostgreSQL-Compatible

대상 아키텍처

다음 다이어그램은 PgAdmin을 SSH 터널과 함께 사용하여 인터넷 게이트웨이를 통해 데이터베이스에 연결되는 EC2 인스턴스에 연결하는 것을 보여줍니다.

도구

서비스

- [Amazon Aurora PostgreSQL-Compatible Edition](#)은 PostgreSQL 배포를 설정, 운영 및 규모를 조정할 수 있는 완전관리형의 ACID 준수 관계형 데이터베이스 엔진입니다.
- [Amazon Elastic Compute Cloud\(Amazon EC2\)](#)는 AWS 클라우드에서 확장 가능한 컴퓨팅 용량을 제공합니다. 필요한 만큼 가상 서버를 시작하고 빠르게 스케일 업하거나 스케일 다운할 수 있습니다.

기타 서비스

- [pgAdmin](#)은 PostgreSQL을 위한 오픈 소스 관리 도구입니다. 데이터베이스 객체를 생성, 유지 관리 및 사용하는 데 도움이 되는 그래픽 인터페이스를 제공합니다.

에픽

연결 생성

작업	설명	필요한 기술
서버를 생성합니다.	PgAdmin에서 생성을 선택한 다음 서버를 선택합니다. 서버 대화 상자를 사용하여 서버를 등록하고, 연결을 구성하며, SSH 터널링을 통해 연결하도록 PgAdmin을 설정하는 방법에 대한 추가 도움말은 관련 리소스 섹션의 링크를 참조하십시오.	DBA
서버의 이름을 입력합니다.	일반 탭에서 이름을 입력합니다.	DBA
데이터베이스 세부 정보를 입력합니다.	연결 탭에서 다음 값을 입력합니다. <ul style="list-style-type: none"> 호스트 이름/주소 포트 데이터베이스 유지 관리 사용자 이름 암호 	DBA
Amazon EC2 서버 세부 정보를 입력합니다.	SSH 터널 탭에서 퍼블릭 서브넷에 있는 Amazon EC2 인스턴스의 세부 정보를 제공합니다. <ul style="list-style-type: none"> SSH 터널링 사용을 예로 설정하여 pgAdmin이 지정된 서버에 연결할 때 SSH 터널을 사용하도록 지정합니다. 	DBA

작업	설명	필요한 기술
	<ul style="list-style-type: none"> • 터널 호스트 필드에 SSH 호스트의 이름 또는 IP 주소 (예: 10.x.x.x) 를 지정합니다. • 터널 포트 필드에 SSH 호스트의 포트 (예: 22) 를 지정합니다. • 사용자 이름 필드에 SSH 호스트에 대한 로그인 권한이 있는 사용자 이름(예: ec2-user)을 지정합니다. • PgAdmin이 연결할 때 개인 키 파일을 사용하도록 인증 유형을 ID 파일로 지정합니다. • ID 파일 필드에 개인 정보 보호 강화 메일(PEM) 파일의 위치를 포함하십시오. .pem 파일은 Amazon EC2 키 페어입니다. 	
<p>저장하고 연결하십시오.</p>	<p>저장을 선택하여 설정을 완료하고 SSH 터널을 사용하여 Aurora PostgreSQL-Compatible 데이터베이스에 연결합니다.</p>	<p>DBA</p>

관련 리소스

- [서버 다이얼로그](#)
- [서버에 연결](#)

JSON Oracle 쿼리를 PostgreSQL 데이터베이스 SQL로 변환

작성자: Pinesh Singal(AWS), Lokesh Gurram(AWS)

환경: PoC 또는 파일럿	소스: 데이터베이스: 관계형	대상: Amazon RDS PostgreSQL
R 타입: 리아키텍트	워크로드: Oracle	기술 : 데이터베이스, 마이그레이션
AWS 서비스: Amazon Aurora, Amazon RDS		

요약

온프레미스에서 Amazon Web Services(AWS) 클라우드로 이전하기 위한 이 마이그레이션 프로세스는 AWS Schema Conversion Tool(AWS SCT)를 사용하여 Oracle 데이터베이스의 코드를 PostgreSQL 데이터베이스로 변환합니다. 대부분의 코드는 AWS SCT에 의해 자동으로 변환됩니다. 그러나 JSON 관련 Oracle 쿼리는 자동으로 변환되지 않습니다.

Oracle 12.2 버전부터 Oracle 데이터베이스는 JSON 기반 데이터를 행 기반 데이터로 변환하는 데 도움이 되는 다양한 JSON 함수를 지원합니다. 그러나 AWS SCT는 JSON 기반 데이터를 PostgreSQL에서 지원하는 언어로 자동 변환하지 않습니다.

이 마이그레이션 패턴은 주로 JSON_OBJECT, JSON_ARRAYAGG 및 JSON_TABLE와 같은 함수를 사용하는 JSON 관련 Oracle 쿼리를 Oracle 데이터베이스에서 PostgreSQL 데이터베이스로 수동으로 변환하는 데 중점을 둡니다.

사전 조건 및 제한 사항

사전 조건

- 활성 상태의 AWS 계정
- 온프레미스 Oracle 데이터베이스 인스턴스(작동 및 실행 중)
- Amazon Relational Database Service(RDS) for PostgreSQL 또는 Amazon Aurora PostgreSQL-Compatible Edition 데이터베이스 인스턴스(가동 및 실행 중)

제한 사항

- JSON 관련 쿼리에는 고정된 KEY 및 VALUE 형식이 필요합니다. 해당 형식을 사용하지 않으면 잘못된 결과가 반환됩니다.
- JSON 구조를 변경하여 결과 섹션에 새 KEY 및 VALUE 쌍이 추가되는 경우 SQL 쿼리에서 해당 프로시저 또는 함수를 변경해야 합니다.
- 일부 JSON 관련 함수는 이전 버전의 Oracle 및 PostgreSQL에서 지원되지만 기능이 더 적습니다.

제품 버전

- Oracle 데이터베이스 버전 12.2 이상
- Amazon RDS for PostgreSQL 또는 Aurora PostgreSQL 호환 버전 9.5 이상
- AWS SCT 최신 버전(버전 1.0.664를 사용하여 테스트됨)

아키텍처

소스 기술 스택

- 버전 19c의 Oracle 데이터베이스 인스턴스

대상 기술 스택

- 버전 13의 Amazon RDS for PostgreSQL 또는 Aurora PostgreSQL-Compatible 데이터베이스 인스턴스

대상 아키텍처

1. AWS SCT를 JSON 함수 코드와 함께 사용하여 소스 코드를 Oracle에서 PostgreSQL로 변환합니다.
2. 변환 과정에서 PostgreSQL이 지원하는 마이그레이션된 .sql 파일이 생성됩니다.
3. 변환되지 않은 Oracle JSON 함수 코드를 PostgreSQL JSON 함수 코드로 수동으로 변환합니다.
4. 대상 Aurora PostgreSQL 호환 DB 인스턴스에서 .sql 파일을 실행합니다.

도구

서비스

- [Amazon Aurora](#)는 클라우드용으로 구축되었으며 MySQL 및 PostgreSQL과 호환되는 완전 관리형 관계형 데이터베이스 엔진입니다.
- [Amazon Relational Database Service\(Amazon RDS\) for PostgreSQL](#)은 AWS 클라우드에서 PostgreSQL 관계형 데이터베이스를 설정, 운영 및 조정하는 데 도움이 됩니다.
- [AWS Schema Conversion Tool\(AWS SCT\)](#)은 소스 데이터베이스 스키마와 대부분의 사용자 지정 코드를 대상 데이터베이스와 호환되는 형식으로 자동 변환하여 이기종 데이터베이스 마이그레이션을 지원합니다.

기타 서비스

- [Oracle SQL Developer](#)는 기존 배포와 클라우드 기반 배포 모두에서 Oracle 데이터베이스의 개발 및 관리를 간소화하는 통합 개발 환경입니다.
- PGAdmin 또는 DBeaver. [pgAdmin](#)은 PostgreSQL을 위한 오픈 소스 관리 도구입니다. 데이터베이스 객체를 생성, 유지 관리 및 사용하는 데 도움이 되는 그래픽 인터페이스를 제공합니다. [DBeaver](#)는 유니버설 데이터베이스 도구입니다.

모범 사례

Oracle 쿼리는 JSON_TABLE 함수를 사용할 때 CAST 유형을 기본값으로 사용합니다. 두 번 큰 문자(>>)를 사용하여 PostgreSQL의 CAST도 사용하는 것이 가장 좋습니다.

자세한 내용은 추가 정보 섹션의 Postgres_SQL_Read_JSON을 참조하세요.

에픽

Oracle 및 PostgreSQL 데이터베이스에서 JSON 데이터를 생성합니다.

작업	설명	필요한 기술
JSON 데이터를 Oracle 데이터베이스에 저장합니다.	Oracle 데이터베이스에 테이블을 생성하고 CLOB 열에 JSON 데이터를 저장합니다. 추가 정보 섹션에 있는 Oracle_Ta	마이그레이션 엔지니어

작업	설명	필요한 기술
	ble_Creation_Insert_Script를 사용합니다.	
JSON 데이터를 PostgreSQL 데이터베이스에 저장합니다.	PostgreSQL 데이터베이스에서 테이블을 생성하고 TEXT 열에 JSON 데이터를 저장합니다. 추가 정보 섹션에 있는 Postgres_Table_Creation_Insert_Script를 사용합니다.	마이그레이션 엔지니어

JSON을 ROW 형식으로 변환

작업	설명	필요한 기술
Oracle 데이터베이스의 JSON 데이터를 변환합니다.	Oracle SQL 쿼리를 작성하여 JSON 데이터를 ROW 형식으로 읽어 들입니다. 자세한 내용 및 예제 구문은 추가 정보 섹션의 Oracle_SQL_Read_JSON을 참조하세요.	마이그레이션 엔지니어
PostgreSQL 데이터베이스의 JSON 데이터를 변환합니다.	PostgreSQL 쿼리를 작성하여 JSON 데이터를 ROW 형식으로 읽어 들입니다. 자세한 내용 및 예제 구문은 추가 정보 섹션의 Postgres_SQL_Read_JSON을 참조하세요.	마이그레이션 엔지니어

SQL 쿼리를 사용하여 JSON 데이터를 수동으로 변환하고 출력을 JSON 형식으로 보고합니다.

작업	설명	필요한 기술
Oracle SQL 쿼리에 대한 집계 및 검증을 수행합니다.	JSON 데이터를 수동으로 변환하려면 Oracle SQL 쿼리	마이그레이션 엔지니어

작업	설명	필요한 기술
	<p>에서 조인, 집계 및 검증을 수행하고 출력을 JSON 형식으로 보고하세요. 추가 정보 섹션의 Oracle_SQL_JSON_Aggregation_Join 아래에 있는 코드를 사용합니다.</p> <ol style="list-style-type: none"> 1. JOIN - JSON 형식의 데이터가 입력 파라미터로 쿼리에 전달됩니다. 이 정적 데이터와 Oracle DB 테이블 <code>aws_test_table</code> 의 JSON 데이터 사이에 내부 조인이 이루어집니다. 2. 검증을 통한 집계 - JSON 데이터에는 <code>accountNumber</code> , <code>parentAccountNumber</code> , <code>businessUnitId</code> , <code>positionId</code> 와 같은 값을 포함하는 KEY 및 VALUE 파라미터가 있으며, 이러한 값은 SUM 및 COUNT 집계에서 사용됩니다. 3. JSON 형식 - 조인 및 집계 후에는 <code>JSON_OBJECT</code> 및 <code>JSON_ARRAYAGG</code> 를 사용하여 데이터를 JSON 형식으로 보고합니다. 	

작업	설명	필요한 기술
<p>Postgres SQL 쿼리에 대한 집계 및 검증을 수행합니다.</p>	<p>JSON 데이터를 수동으로 변환하려면 PostgreSQL 쿼리에서 조인, 집계 및 검증을 수행하고 출력을 JSON 형식으로 보고하세요. 추가 정보 섹션의 Postgres_SQL_JSON_Aggregation_Join 아래에 있는 코드를 사용합니다.</p> <ol style="list-style-type: none"> 1. JOIN - JSON 형식의 데이터 (tab1)가 입력 파라미터로 WITH 조항 쿼리에 전달됩니다. 이 정적 데이터와 tab 테이블에 있는 JSON 데이터 사이에 JOIN이 이루어집니다. aws_test_pg_table 테이블에 JSON 데이터가 들어 있는 WITH 절을 사용하여 JOIN을 만들 수도 있습니다. 2. 집계 - JSON 데이터에는 accountNumber , parentAccountNumber , businessUnitId , positionId 와 같은 값을 포함하는 KEY 및 VALUE 파라미터가 있으며, 이러한 값은 및 SUM 및 COUNT 집계 사용됩니다. 3. JSON 형식 - 조인 및 집계 후에는 JSON_BUILT_OBJECT 및 JSON_AGG를 사용하여 데이터를 JSON 형식으로 보고합니다. 	<p>마이그레이션 엔지니어</p>

오라클 프로시저를 JSON 쿼리가 포함된 PostgreSQL 함수로 변환

작업	설명	필요한 기술
Oracle 프로시저의 JSON 쿼리를 행으로 변환합니다.	예제 Oracle 프로시저의 경우 이전 Oracle 쿼리와 추가 정보 섹션의 Oracle_procedure_with_JSON_Query에 있는 코드를 사용하세요.	마이그레이션 엔지니어
JSON 쿼리가 있는 PostgreSQL 함수를 행 기반 데이터로 변환합니다.	예제 PostgreSQL 함수의 경우 추가 정보 섹션의 PostgreSQL 쿼리와 Postgres_function_with_JSON_Query에 있는 코드를 사용하세요.	마이그레이션 엔지니어

관련 리소스

- [Oracle JSON 함수](#)
- [PostgreSQL JSON 함수](#)
- [Oracle JSON 함수 예제](#)
- [PostgreSQL JSON 함수 예제](#)
- [AWS Schema Conversion Tool](#)

추가 정보

Oracle 데이터베이스의 JSON 코드를 PostgreSQL 데이터베이스로 변환하려면 다음 스크립트를 순서대로 사용하세요.

1: Oracle_Table_Creation_Insert_Script

```
create table aws_test_table(id number,created_on date default sysdate,modified_on
date,json_doc clob);

REM INSERTING into EXPORT_TABLE
SET DEFINE OFF;
Insert into aws_test_table (ID,CREATED_ON,MODIFIED_ON,json_doc)
```

```

values (1,to_date('02-AUG-2022 12:30:14','DD-MON-YYYY HH24:MI:SS'),to_date('02-AUG-2022
12:30:14','DD-MON-YYYY HH24:MI:SS'),TO_CLOB(q'[{
  "metadata" : {
    "upperLastNameFirstName" : "ABC XYZ",
    "upperEmailAddress" : "abc@gmail.com",
    "profileType" : "P"
  },
  "data" : {
    "onlineContactId" : "032323323",
    "displayName" : "Abc, Xyz",
    "firstName" : "Xyz",
    "lastName" : "Abc",
    "emailAddress" : "abc@gmail.com",
    "productRegistrationStatus" : "Not registered",
    "positionId" : "0100",
    "arrayPattern" : " -'",
    "a]')
|| TO_CLOB(q'[ccount" : {
  "companyId" : "SMGE",
  "businessUnitId" : 7,
  "accountNumber" : 42000,
  "parentAccountNumber" : 32000,
  "firstName" : "john",
  "lastName" : "doe",
  "street1" : "ret0dertcaShr ",
  "city" : "new york",
  "postalcode" : "XY ABC",
  "country" : "United States"
}],
"products" : [
  {
    "appUserGuid" : "i0acc4450000001823fbad478e2eab8a0",
    "id" : "0000000046",
  }
]')
|| TO_CLOB(q'[      "name" : "ProView",
  "domain" : "EREADER",
  "registrationStatus" : false,
  "status" : "11"
  }
]
}
}]')));

```

```

Insert into aws_test_table (ID,CREATED_ON,MODIFIED_ON,json_doc) values (2,to_date('02-
AUG-2022 12:30:14','DD-MON-YYYY HH24:MI:SS'),to_date('02-AUG-2022 12:30:14','DD-MON-
YYYY HH24:MI:SS'),TO_CLOB(q'[{
  "metadata" : {
    "upperLastNameFirstName" : "PQR XYZ",
    "upperEmailAddress" : "pqr@gmail.com",
    "profileType" : "P"
  },
  "data" : {
    "onlineContactId" : "54534343",
    "displayName" : "Xyz, pqr",
    "firstName" : "pqr",
    "lastName" : "Xyz",
    "emailAddress" : "pqr@gmail.com",
    "productRegistrationStatus" : "Not registered",
    "positionId" : "0090",
    "arrayPattern" : " -'",
    "account" : {
      "companyId" : "CARS",
      "busin]')
|| TO_CLOB(q'[essUnitId" : 6,
  "accountNumber" : 42001,
  "parentAccountNumber" : 32001,
  "firstName" : "terry",
  "lastName" : "whitlock",
  "street1" : "U0 123",
  "city" : "TOTORON",
  "region" : "NO",
  "postalcode" : "LKM 111",
  "country" : "Canada"
},
"products" : [
  {
    "appUserGuid" : "ia744d7790000016899f8cf3f417d6df6",
    "id" : "0000000014",
    "name" : "ProView eLooseleaf",
  ]')
|| TO_CLOB(q'[ "domain" : "EREADER",
  "registrationStatus" : false,
  "status" : "11"
}
]
}
}]')));

```

```
commit;
```

2. Postgres_Table_Creation_Insert_Script

```
create table aws_test_pg_table(id int,created_on date ,modified_on date,json_doc text);
insert into aws_test_pg_table(id,created_on,modified_on,json_doc)
values(1,now(),now(),'{
  "metadata" : {
    "upperLastNameFirstName" : "ABC XYZ",
    "upperEmailAddress" : "abc@gmail.com",
    "profileType" : "P"
  },
  "data" : {
    "onlineContactId" : "032323323",
    "displayName" : "Abc, Xyz",
    "firstName" : "Xyz",
    "lastName" : "Abc",
    "emailAddress" : "abc@gmail.com",
    "productRegistrationStatus" : "Not registered",
    "positionId" : "0100",
    "arrayPattern" : " -",
    "account" : {
      "companyId" : "SMGE",
      "businessUnitId" : 7,
      "accountNumber" : 42000,
      "parentAccountNumber" : 32000,
      "firstName" : "john",
      "lastName" : "doe",
      "street1" : "ret0dertcaShr ",
      "city" : "new york",
      "postalcode" : "XY ABC",
      "country" : "United States"
    },
    "products" : [
      {
        "appUserGuid" : "i0acc4450000001823fbad478e2eab8a0",
        "id" : "0000000046",
        "name" : "ProView",
        "domain" : "EREADER",
        "registrationStatus" : false,
        "status" : "11"
      }
    ]
  }
}
```

```
    ]
  }
}');

insert into aws_test_pg_table(id,created_on,modified_on,json_doc)
values(2,now(),now(),'{
  "metadata" : {
    "upperLastNameFirstName" : "PQR XYZ",
    "upperEmailAddress" : "pqr@gmail.com",
    "profileType" : "P"
  },
  "data" : {
    "onlineContactId" : "54534343",
    "displayName" : "Xyz, pqr",
    "firstName" : "pqr",
    "lastName" : "Xyz",
    "emailAddress" : "a*b**@h**.k**",
    "productRegistrationStatus" : "Not registered",
    "positionId" : "0090",
    "arrayPattern" : " -",
    "account" : {
      "companyId" : "CARS",
      "businessUnitId" : 6,
      "accountNumber" : 42001,
      "parentAccountNumber" : 32001,
      "firstName" : "terry",
      "lastName" : "whitlock",
      "street1" : "U0 123",
      "city" : "TOTORON",
      "region" : "NO",
      "postalcode" : "LKM 111",
      "country" : "Canada"
    },
    "products" : [
      {
        "appUserGuid" : "ia744d7790000016899f8cf3f417d6df6",
        "id" : "0000000014",
        "name" : "ProView eLooseleaf",
        "domain" : "EREADER",
        "registrationStatus" : false,
        "status" : "11"
      }
    ]
  }
}');
```

```

}
}');

```

3. Oracle_SQL_Read_JSON

다음 코드 블록은 Oracle JSON 데이터를 행 형식으로 변환하는 방법을 보여줍니다.

쿼리 및 구문 예제

```

SELECT  JSON_OBJECT(
  'accountCounts' VALUE JSON_ARRAYAGG(
    JSON_OBJECT(
      'businessUnitId' VALUE business_unit_id,
      'parentAccountNumber' VALUE parent_account_number,
      'accountNumber' VALUE account_number,
      'totalOnlineContactsCount' VALUE online_contacts_count,
      'countByPosition' VALUE
        JSON_OBJECT(
          'taxProfessionalCount' VALUE tax_count,
          'attorneyCount' VALUE attorney_count,
          'nonAttorneyCount' VALUE non_attorney_count,
          'clerkCount' VALUE clerk_count
        ) ) ) ) FROM
  (SELECT  tab_data.business_unit_id,
    tab_data.parent_account_number,
    tab_data.account_number,
    SUM(1) online_contacts_count,
    SUM(CASE WHEN tab_data.position_id = '0095' THEN 1 ELSE 0 END) tax_count,
    SUM(CASE  WHEN tab_data.position_id = '0100' THEN 1 ELSE 0 END)
attorney_count,
    SUM(CASE  WHEN tab_data.position_id = '0090' THEN 1 ELSE 0 END)
non_attorney_count,
    SUM(CASE  WHEN tab_data.position_id = '0050' THEN 1 ELSE 0 END)
clerk_count
  FROM aws_test_table scco,JSON_TABLE ( json_doc, '$' ERROR ON ERROR
  COLUMNS (
    parent_account_number NUMBER PATH
    '$.data.account.parentAccountNumber',
    account_number NUMBER PATH '$.data.account.accountNumber',
    business_unit_id NUMBER PATH '$.data.account.businessUnitId',
    position_id VARCHAR2 ( 4 ) PATH '$.data.positionId'
  ) AS tab_data
  INNER JOIN JSON_TABLE ( '{
"accounts": [{

```

```

    "accountNumber": 42000,
    "parentAccountNumber": 32000,
    "businessUnitId": 7
  }, {
    "accountNumber": 42001,
    "parentAccountNumber": 32001,
    "businessUnitId": 6
  }
]
}', '$.accounts[*]' ERROR ON ERROR
COLUMNS (
parent_account_number PATH '$.parentAccountNumber',
account_number PATH '$.accountNumber',
business_unit_id PATH '$.businessUnitId')
) static_data
ON ( static_data.parent_account_number = tab_data.parent_account_number
AND static_data.account_number = tab_data.account_number
AND static_data.business_unit_id = tab_data.business_unit_id )
GROUP BY
tab_data.business_unit_id,
tab_data.parent_account_number,
tab_data.account_number );

```

JSON 문서는 데이터를 컬렉션으로 저장합니다. 각 컬렉션에는 KEY 및 VALUE 쌍이 있을 수 있습니다. 모든 VALUE는 KEY 중첩과 VALUE 쌍을 가질 수 있습니다. 다음 표는 JSON 문서의 VALUE에서 특정 내용을 읽는 방법에 대한 정보를 제공합니다.

키	값을 가져오는 데 사용할 계층 또는 경로	값
profileType	metadata -> profileType	"P"
positionId	data -> positionId	"0100"
accountNumber	data -> 계정 -> accountNumber	42000

이전 표에서는 KEY profileType이 metadata KEY의 VALUE입니다. KEY positionId는 data KEY의 VALUE입니다. KEY accountNumber는 account KEY의 VALUE이고, account KEY는 data KEY의 VALUE입니다.

예제 JSON 문서

```
{
  "metadata" : {
    "upperLastNameFirstName" : "ABC XYZ",
    "upperEmailAddress" : "abc@gmail.com",
    "profileType" : "P"
  },
  "data" : {
    "onlineContactId" : "032323323",
    "displayName" : "Abc, Xyz",
    "firstName" : "Xyz",
    "lastName" : "Abc",
    "emailAddress" : "abc@gmail.com",
    "productRegistrationStatus" : "Not registered",
    "positionId" : "0100",
    "arrayPattern" : " -",
    "account" : {
      "companyId" : "SMGE",
      "businessUnitId" : 7,
      "accountNumber" : 42000,
      "parentAccountNumber" : 32000,
      "firstName" : "john",
      "lastName" : "doe",
      "street1" : "ret0dertcaShr ",
      "city" : "new york",
      "postalcode" : "XY ABC",
      "country" : "United States"
    },
    "products" : [
      {
        "appUserGuid" : "i0acc4450000001823fbad478e2eab8a0",
        "id" : "0000000046",
        "name" : "ProView",
        "domain" : "EREADER",
        "registrationStatus" : false,
        "status" : "11"
      }
    ]
  }
}
```

JSON 문서에서 선택한 필드를 가져오는 데 사용되는 SQL 쿼리

```
select parent_account_number,account_number,business_unit_id,position_id from
aws_test_table aws,JSON_TABLE ( json_doc, '$' ERROR ON ERROR
COLUMNS (
parent_account_number NUMBER PATH '$.data.account.parentAccountNumber',
account_number NUMBER PATH '$.data.account.accountNumber',
business_unit_id NUMBER PATH '$.data.account.businessUnitId',
position_id VARCHAR2 ( 4 ) PATH '$.data.positionId'
)) as sc
```

이전 쿼리에서는 JSON_TABLE은 JSON 데이터를 행 형식으로 변환하는 Oracle의 기본 제공 함수입니다. JSON_TABLE 함수에는 JSON 형식의 파라미터가 예상됩니다.

COLUMNS의 모든 항목에는 PATH가 미리 정의되어 있으며 주어진 KEY에 적합한 VALUE가 행 형식으로 반환됩니다.

이전 쿼리의 결과

PARENT_AC COUNT_NUMBER	ACCOUNT_NUMBER	BUSINESS_UNIT_ID	POSITION_ID
32000	42000	7	0100
32001	42001	6	0090

4. Postgres_SQL_Read_JSON

쿼리 및 구문 예제

```
select *
from (
select (json_doc::json->'data'->'account'->'parentAccountNumber')::INTEGER as
parentAccountNumber,
(json_doc::json->'data'->'account'->'accountNumber')::INTEGER as accountNumber,
(json_doc::json->'data'->'account'->'businessUnitId')::INTEGER as businessUnitId,
(json_doc::json->'data'->'positionId')::VARCHAR as positionId
from aws_test_pg_table) d ;
```

Oracle에서 PATH는 특정 KEY 및 VALUE를 식별하는 데 사용됩니다. 그러나 PostgreSQL은 JSON에서 KEY 및 VALUE를 위한 HIERARCHY 모델을 사용합니다. Oracle_SQL_Read_JSON에 언급된 것과 동일한 JSON 데이터가 다음 예제에 사용됩니다.

유형이 CAST인 SQL 쿼리는 허용되지 않습니다

(CAST를 강제로 입력하면 구문 오류가 발생하여 쿼리가 실패합니다.)

```
select *
from (
select (json_doc::json->'data'->'account'->'parentAccountNumber') as
parentAccountNumber,
(json_doc::json->'data'->'account'->'accountNumber')as accountNumber,
(json_doc::json->'data'->'account'->'businessUnitId') as businessUnitId,
(json_doc::json->'data'->'positionId')as positionId
from aws_test_pg_table) d ;
```

하나 이상의 연산자(>)를 사용하면 해당 KEY에 대해 VALUE이 정의된 값이 반환됩니다. 예를 들어, KEY: positionId 및 VALUE: "0100"입니다.

하나 이상의 연산자(>)를 사용할 때는 유형 CAST를 사용할 수 없습니다.

유형이 CAST인 SQL 쿼리는 허용됩니다

```
select *
from (
select (json_doc::json->'data'->'account'->>'parentAccountNumber')::INTEGER as
parentAccountNumber,
(json_doc::json->'data'->'account'->>'accountNumber')::INTEGER as accountNumber,
(json_doc::json->'data'->'account'->>'businessUnitId')::INTEGER as businessUnitId,
(json_doc::json->'data'->>'positionId')::varchar as positionId
from aws_test_pg_table) d ;
```

CAST유형을 사용하려면 double greater-than 연산자를 사용해야 합니다. 하나 이상의 큰 연산자를 사용하는 경우 쿼리는 정의된 VALUE(예: KEY는 positionId 및 VALUE는 "0100")를 반환합니다. 두 배 큰 연산자(>>)를 사용하면 KEY (예: KEY: positionId 및 VALUE: 0100, 큰따옴표 제외)에 대해 정의된 실제 값이 반환됩니다.

이전 케이스에서 parentAccountNumber은 INT에 대한 CAST 타입, accountNumber은 INT에 대한 CAST 타입, businessUnitId는 INT에 대한 CAST 타입, positionId는 VARCHAR에 대한 CAST 타입입니다.

다음 표에는 단일 초과 연산자(>)와 두 배 큰 연산자(>>)의 역할을 설명하는 쿼리 결과가 나와 있습니다.

첫 번째 테이블 테이블의 쿼리는 단일 대보다 큰 연산자(>)를 사용합니다. 각 열은 JSON 유형이며 다른 데이터 유형으로 변환할 수 없습니다.

parentAccountNumber	accountNumber	businessUnitId	positionId
2003565430	2003564830	7	"0100"
2005284042	2005284042	6	"0090"
2000272719	2000272719	1	"0100"

두 번째 테이블에서 쿼리는 두 배 큰 연산자(>>)를 사용합니다. 각 열은 열 값을 기반으로 하는 CAST 유형을 지원합니다. 예를 들어 이 경우에는 INTEGER입니다.

parentAccountNumber	accountNumber	businessUnitId	positionId
2003565430	2003564830	7	0100
2005284042	2005284042	6	0090
2000272719	2000272719	1	0100

5. Oracle_SQL_JSON_Aggregation_Join

쿼리 예제

```
SELECT
  JSON_OBJECT(
    'accountCounts' VALUE JSON_ARRAYAGG(
      JSON_OBJECT(
        'businessUnitId' VALUE business_unit_id,
        'parentAccountNumber' VALUE parent_account_number,
        'accountNumber' VALUE account_number,
        'totalOnlineContactsCount' VALUE online_contacts_count,
        'countByPosition' VALUE
          JSON_OBJECT(
            'taxProfessionalCount' VALUE tax_count,
```

```

        'attorneyCount' VALUE attorney_count,
        'nonAttorneyCount' VALUE non_attorney_count,
        'clerkCount' VALUE clerk_count
    ) ) ) )
FROM
    (SELECT
        tab_data.business_unit_id,
        tab_data.parent_account_number,
        tab_data.account_number,
        SUM(1) online_contacts_count,
        SUM(CASE WHEN tab_data.position_id = '0095' THEN 1 ELSE 0 END) tax_count,
        SUM(CASE WHEN tab_data.position_id = '0100' THEN 1 ELSE 0 END)
attorney_count,
        SUM(CASE WHEN tab_data.position_id = '0090' THEN 1 ELSE 0 END)
non_attorney_count,
        SUM(CASE WHEN tab_data.position_id = '0050' THEN 1 ELSE 0 END)
clerk_count
    FROM aws_test_table scco,JSON_TABLE ( json_doc, '$' ERROR ON ERROR
    COLUMNS (
        parent_account_number NUMBER PATH
        '$.data.account.parentAccountNumber',
        account_number NUMBER PATH '$.data.account.accountNumber',
        business_unit_id NUMBER PATH '$.data.account.businessUnitId',
        position_id VARCHAR2 ( 4 ) PATH '$.data.positionId'
    ) AS tab_data
    INNER JOIN JSON_TABLE ( '{
"accounts": [{
    "accountNumber": 42000,
    "parentAccountNumber": 32000,
    "businessUnitId": 7
}, {
    "accountNumber": 42001,
    "parentAccountNumber": 32001,
    "businessUnitId": 6
}]
}', '$.accounts[*]' ERROR ON ERROR
    COLUMNS (
        parent_account_number PATH '$.parentAccountNumber',
        account_number PATH '$.accountNumber',
        business_unit_id PATH '$.businessUnitId')
    ) static_data
    ON ( static_data.parent_account_number = tab_data.parent_account_number
        AND static_data.account_number = tab_data.account_number
        AND static_data.business_unit_id = tab_data.business_unit_id )

```

```

GROUP BY
    tab_data.business_unit_id,
    tab_data.parent_account_number,
    tab_data.account_number
);

```

행 수준 데이터를 JSON 형식으로 변환하기 위해 Oracle에는 JSON_OBJECT, JSON_ARRAY, JSON_OBJECTAGG, JSON_ARRAYAGG와 같은 빌트인 함수가 있습니다.

- JSON_OBJECT는 KEY 및 VALUE라는 두 개의 파라미터를 허용합니다. KEY 파라미터는 하드코딩되거나 정적이어야 합니다. VALUE 파라미터는 테이블 출력에서 파생됩니다.
- JSON_ARRAYAGG은 JSON_OBJECT를 파라미터로 받아들입니다. 이렇게 하면 JSON_OBJECT 요소 세트를 목록으로 그룹화하는 데 도움이 됩니다. 예를 들어 여러 레코드(데이터 세트에 있는 여러 개의 KEY 및 VALUE 페어)가 있는 JSON_OBJECT 요소가 있는 경우 JSON_ARRAYAGG는 데이터 세트를 추가하고 목록을 생성합니다. 데이터 구조 언어에 따르면 LIST는 요소 그룹입니다. 이 컨텍스트에서는 LIST는 JSON_OBJECT 요소 그룹입니다.

다음 예에서는 JSON_OBJECT 요소 하나를 보여 줍니다.

```

{
  "taxProfessionalCount": 0,
  "attorneyCount": 0,
  "nonAttorneyCount": 1,
  "clerkCount": 0
}

```

다음 예제에서는 대괄호([])로 표시된 LIST가 있는 두 JSON_OBJECT 요소를 보여줍니다.

```

[
  {
    "taxProfessionalCount": 0,
    "attorneyCount": 0,
    "nonAttorneyCount": 1,
    "clerkCount": 0
  },
  {
    "taxProfessionalCount": 2,
    "attorneyCount": 1,
    "nonAttorneyCount": 3,
  }
]

```

```

        "clerkCount":4
    }
]

```

예제 SQL 쿼리

```

SELECT
  JSON_OBJECT(
    'accountCounts' VALUE JSON_ARRAYAGG(
      JSON_OBJECT(
        'businessUnitId' VALUE business_unit_id,
        'parentAccountNumber' VALUE parent_account_number,
        'accountNumber' VALUE account_number,
        'totalOnlineContactsCount' VALUE online_contacts_count,
        'countByPosition' VALUE
          JSON_OBJECT(
            'taxProfessionalCount' VALUE tax_count,
            'attorneyCount' VALUE attorney_count,
            'nonAttorneyCount' VALUE non_attorney_count,
            'clerkCount' VALUE clerk_count
          )
        )
      )
    )
FROM
  (SELECT
    tab_data.business_unit_id,
    tab_data.parent_account_number,
    tab_data.account_number,
    SUM(1) online_contacts_count,
    SUM(CASE WHEN tab_data.position_id = '0095' THEN 1 ELSE 0 END
    ) tax_count,
    SUM(CASE WHEN tab_data.position_id = '0100' THEN 1 ELSE
0 END
    ) attorney_count,
    SUM(CASE WHEN tab_data.position_id = '0090' THEN 1 ELSE
0 END
    ) non_attorney_count,
    SUM(CASE WHEN tab_data.position_id = '0050' THEN 1 ELSE
0 END

```

```

    )      clerk_count

FROM
  aws_test_table scco, JSON_TABLE ( json_doc, '$' ERROR ON ERROR
  COLUMNS (
    parent_account_number NUMBER PATH '$.data.account.parentAccountNumber',
    account_number NUMBER PATH '$.data.account.accountNumber',
    business_unit_id NUMBER PATH '$.data.account.businessUnitId',
    position_id VARCHAR2 ( 4 ) PATH '$.data.positionId'      )
  ) AS tab_data
  INNER JOIN JSON_TABLE ( '{
"accounts": [{
  "accountNumber": 42000,
  "parentAccountNumber": 32000,
  "businessUnitId": 7
}, {
  "accountNumber": 42001,
  "parentAccountNumber": 32001,
  "businessUnitId": 6
}]
}', '$.accounts[*]' ERROR ON ERROR
  COLUMNS (
    parent_account_number PATH '$.parentAccountNumber',
    account_number PATH '$.accountNumber',
    business_unit_id PATH '$.businessUnitId')
  ) static_data ON ( static_data.parent_account_number =
tab_data.parent_account_number
                    AND static_data.account_number = tab_data.account_number

                    AND static_data.business_unit_id =
tab_data.business_unit_id )
  GROUP BY
    tab_data.business_unit_id,
    tab_data.parent_account_number,
    tab_data.account_number
);

```

이전 SQL 쿼리의 출력 예제

```

{
  "accountCounts": [
    {
      "businessUnitId": 6,

```

```

    "parentAccountNumber": 32001,
    "accountNumber": 42001,
    "totalOnlineContactsCount": 1,
    "countByPosition": {
      "taxProfessionalCount": 0,
      "attorneyCount": 0,
      "nonAttorneyCount": 1,
      "clerkCount": 0
    }
  },
  {
    "businessUnitId": 7,
    "parentAccountNumber": 32000,
    "accountNumber": 42000,
    "totalOnlineContactsCount": 1,
    "countByPosition": {
      "taxProfessionalCount": 0,
      "attorneyCount": 1,
      "nonAttorneyCount": 0,
      "clerkCount": 0
    }
  }
]
}

```

6. Postgres_SQL_JSON_Aggregation_Join

PostgreSQL 빌트인 함수 `JSON_BUILD_OBJECT` 및 `JSON_AGG`는 ROW 수준 데이터를 JSON 형식으로 변환합니다. PostgreSQL `JSON_BUILD_OBJECT` 및 `JSON_AGG`는 Oracle `JSON_OBJECT` 및 `JSON_ARRAYAGG`와 동일합니다.

쿼리 예제

```

select
JSON_BUILD_OBJECT ('accountCounts',
  JSON_AGG(
    JSON_BUILD_OBJECT ('businessUnitId',businessUnitId
    , 'parentAccountNumber',parentAccountNumber
    , 'accountNumber',accountNumber
    , 'totalOnlineContactsCount',online_contacts_count,
    'countByPosition',
      JSON_BUILD_OBJECT (
        'taxProfessionalCount',tax_professional_count

```

```

        , 'attorneyCount', attorney_count
        , 'nonAttorneyCount', non_attorney_count
        , 'clerkCount', clerk_count
    )
)
)
)
from (
with tab as (select * from (
select (json_doc::json->'data'->'account'->'parentAccountNumber')::INTEGER as
parentAccountNumber,
(json_doc::json->'data'->'account'->'accountNumber')::INTEGER as accountNumber,
(json_doc::json->'data'->'account'->'businessUnitId')::INTEGER as businessUnitId,
(json_doc::json->'data'->'positionId')::varchar as positionId
from aws_test_pg_table) a ) ,
tab1 as ( select
(json_array_elements(b.jc -> 'accounts') ->> 'accountNumber')::integer accountNumber,
(json_array_elements(b.jc -> 'accounts') ->> 'businessUnitId')::integer
businessUnitId,
(json_array_elements(b.jc -> 'accounts') ->> 'parentAccountNumber')::integer
parentAccountNumber
from (
select '{
    "accounts": [{
        "accountNumber": 42001,
        "parentAccountNumber": 32001,
        "businessUnitId": 6
    }, {
        "accountNumber": 42000,
        "parentAccountNumber": 32000,
        "businessUnitId": 7
    }]
}'::json as jc) b)
select
tab.businessUnitId::text,
tab.parentAccountNumber::text,
tab.accountNumber::text,
SUM(1) online_contacts_count,
SUM(CASE WHEN tab.positionId::text = '0095' THEN 1 ELSE 0 END)
tax_professional_count,
SUM(CASE WHEN tab.positionId::text = '0100' THEN 1 ELSE 0 END) attorney_count,
SUM(CASE WHEN tab.positionId::text = '0090' THEN 1 ELSE 0 END)
non_attorney_count,

```

```
SUM(CASE WHEN tab.positionId::text = '0050' THEN      1 ELSE      0 END)
  clerk_count
from tab1,tab
where tab.parentAccountNumber::INTEGER=tab1.parentAccountNumber::INTEGER
and tab.accountNumber::INTEGER=tab1.accountNumber::INTEGER
and tab.businessUnitId::INTEGER=tab1.businessUnitId::INTEGER
GROUP BY      tab.businessUnitId::text,
              tab.parentAccountNumber::text,
              tab.accountNumber::text) a;
```

이전 쿼리의 예제 출력

Oracle과 PostgreSQL의 출력은 정확히 동일합니다.

```
{
  "accountCounts": [
    {
      "businessUnitId": 6,
      "parentAccountNumber": 32001,
      "accountNumber": 42001,
      "totalOnlineContactsCount": 1,
      "countByPosition": {
        "taxProfessionalCount": 0,
        "attorneyCount": 0,
        "nonAttorneyCount": 1,
        "clerkCount": 0
      }
    },
    {
      "businessUnitId": 7,
      "parentAccountNumber": 32000,
      "accountNumber": 42000,
      "totalOnlineContactsCount": 1,
      "countByPosition": {
        "taxProfessionalCount": 0,
        "attorneyCount": 1,
        "nonAttorneyCount": 0,
        "clerkCount": 0
      }
    }
  ]
}
```

7.Oracle_procedure_with_JSON_Query

이 코드는 Oracle 절차를 JSON SQL 쿼리가 있는 PostgreSQL 함수로 변환합니다. 쿼리가 JSON을 행으로, 그 반대로 변환하는 방법을 보여줍니다.

```
CREATE OR REPLACE PROCEDURE p_json_test(p_in_accounts_json IN varchar2,
  p_out_accunts_json  OUT varchar2)
IS
BEGIN
/*
p_in_accounts_json paramter should have following format:
    {
      "accounts": [{
        "accountNumber": 42000,
        "parentAccountNumber": 32000,
        "businessUnitId": 7
      }, {
        "accountNumber": 42001,
        "parentAccountNumber": 32001,
        "businessUnitId": 6
      }]
    }
*/
SELECT
  JSON_OBJECT(
    'accountCounts' VALUE JSON_ARRAYAGG(
      JSON_OBJECT(
        'businessUnitId' VALUE business_unit_id,
        'parentAccountNumber' VALUE parent_account_number,
        'accountNumber' VALUE account_number,
        'totalOnlineContactsCount' VALUE online_contacts_count,
        'countByPosition' VALUE
          JSON_OBJECT(
            'taxProfessionalCount' VALUE tax_count,
            'attorneyCount' VALUE attorney_count,
            'nonAttorneyCount' VALUE non_attorney_count,
            'clerkCount' VALUE clerk_count
          ) ) ) )
  into p_out_accunts_json
FROM
  (SELECT
    tab_data.business_unit_id,
    tab_data.parent_account_number,
    tab_data.account_number,
```

```

SUM(1) online_contacts_count,
SUM(CASE WHEN tab_data.position_id = '0095' THEN 1 ELSE 0 END) tax_count,
SUM(CASE WHEN tab_data.position_id = '0100' THEN 1 ELSE 0 END)
attorney_count,
SUM(CASE WHEN tab_data.position_id = '0090' THEN 1 ELSE 0 END)
non_attorney_count,
SUM(CASE WHEN tab_data.position_id = '0050' THEN 1 ELSE 0 END)
clerk_count
FROM aws_test_table scco,JSON_TABLE ( json_doc, '$' ERROR ON ERROR
COLUMNS (
parent_account_number NUMBER PATH '$.data.account.parentAccountNumber',
account_number NUMBER PATH '$.data.account.accountNumber',
business_unit_id NUMBER PATH '$.data.account.businessUnitId',
position_id VARCHAR2 ( 4 ) PATH '$.data.positionId' )
) AS tab_data
INNER JOIN JSON_TABLE ( p_in_accounts_json, '$.accounts[*]' ERROR ON ERROR

COLUMNS (
parent_account_number PATH '$.parentAccountNumber',
account_number PATH '$.accountNumber',
business_unit_id PATH '$.businessUnitId')
) static_data
ON ( static_data.parent_account_number = tab_data.parent_account_number
AND static_data.account_number = tab_data.account_number
AND static_data.business_unit_id = tab_data.business_unit_id )
GROUP BY
tab_data.business_unit_id,
tab_data.parent_account_number,
tab_data.account_number
);
EXCEPTION
WHEN OTHERS THEN
raise_application_error(-20001,'Error while running the JSON query');
END;
/

```

프로시저 실행

다음 코드 블록은 프로시저에 대한 예제 JSON 입력을 사용하여 이전에 만든 Oracle 프로시저를 실행하는 방법을 설명합니다. 또한 이 프로시저의 결과 또는 출력도 제공합니다.

```

set serveroutput on;
declare

```

```

v_out varchar2(30000);
v_in varchar2(30000):= '{
    "accounts": [{
        "accountNumber": 42000,
        "parentAccountNumber": 32000,
        "businessUnitId": 7
    }, {
        "accountNumber": 42001,
        "parentAccountNumber": 32001,
        "businessUnitId": 6
    }]
}';
begin
    p_json_test(v_in,v_out);
    dbms_output.put_line(v_out);
end;
/

```

프로시저 출력

```

{
  "accountCounts": [
    {
      "businessUnitId": 6,
      "parentAccountNumber": 32001,
      "accountNumber": 42001,
      "totalOnlineContactsCount": 1,
      "countByPosition": {
        "taxProfessionalCount": 0,
        "attorneyCount": 0,
        "nonAttorneyCount": 1,
        "clerkCount": 0
      }
    },
    {
      "businessUnitId": 7,
      "parentAccountNumber": 32000,
      "accountNumber": 42000,
      "totalOnlineContactsCount": 1,
      "countByPosition": {
        "taxProfessionalCount": 0,
        "attorneyCount": 1,
        "nonAttorneyCount": 0,

```

```

        "clerkCount": 0
    }
}
]
}

```

8.Postgres_function_with_JSON_Query

함수 예제

```

CREATE OR REPLACE FUNCTION f_pg_json_test(p_in_accounts_json text)
RETURNS text
LANGUAGE plpgsql
AS
$$
DECLARE
    v_out_accunts_json text;
BEGIN
SELECT
JSON_BUILD_OBJECT ('accountCounts',
    JSON_AGG(
        JSON_BUILD_OBJECT ('businessUnitId',businessUnitId
        , 'parentAccountNumber',parentAccountNumber
        , 'accountNumber',accountNumber
        , 'totalOnlineContactsCount',online_contacts_count,
        'countByPosition',
        JSON_BUILD_OBJECT (
            'taxProfessionalCount',tax_professional_count
            , 'attorneyCount',attorney_count
            , 'nonAttorneyCount',non_attorney_count
            , 'clerkCount',clerk_count
        )))
INTO v_out_accunts_json
FROM (
WITH tab AS (SELECT * FROM (
SELECT (json_doc::json->'data'->'account'->'parentAccountNumber')::INTEGER AS
parentAccountNumber,
(json_doc::json->'data'->'account'->'accountNumber')::INTEGER AS accountNumber,
(json_doc::json->'data'->'account'->'businessUnitId')::INTEGER AS businessUnitId,
(json_doc::json->'data'->'positionId')::varchar AS positionId
FROM aws_test_pg_table) a ) ,
tab1 AS ( SELECT
(json_array_elements(b.jc -> 'accounts') ->> 'accountNumber')::integer accountNumber,
(json_array_elements(b.jc -> 'accounts') ->> 'businessUnitId')::integer businessUnitId,

```

```

(json_array_elements(b.jc -> 'accounts') ->> 'parentAccountNumber')::integer
parentAccountNumber
FROM (
SELECT p_in_accounts_json::json AS jc) b)
SELECT
tab.businessUnitId::text,
tab.parentAccountNumber::text,
tab.accountNumber::text,
SUM(1) online_contacts_count,
SUM(CASE WHEN tab.positionId::text = '0095' THEN 1 ELSE 0 END)
tax_professional_count,
SUM(CASE WHEN tab.positionId::text = '0100' THEN 1 ELSE 0 END) attorney_count,
SUM(CASE WHEN tab.positionId::text = '0090' THEN 1 ELSE 0 END)
non_attorney_count,
SUM(CASE WHEN tab.positionId::text = '0050' THEN 1 ELSE 0 END)
clerk_count
FROM tab1,tab
WHERE tab.parentAccountNumber::INTEGER=tab1.parentAccountNumber::INTEGER
AND tab.accountNumber::INTEGER=tab1.accountNumber::INTEGER
AND tab.businessUnitId::INTEGER=tab1.businessUnitId::INTEGER
GROUP BY tab.businessUnitId::text,
tab.parentAccountNumber::text,
tab.accountNumber::text) a;
RETURN v_out_accunts_json;
END;
$$;

```

함수 실행

```

select f_pg_json_test('{
  "accounts": [{
    "accountNumber": 42001,
    "parentAccountNumber": 32001,
    "businessUnitId": 6
  }, {
    "accountNumber": 42000,
    "parentAccountNumber": 32000,
    "businessUnitId": 7
  }]
}') ;

```

함수 출력

다음 출력은 Oracle 프로시저 출력과 유사합니다. 차이점은 이 출력이 텍스트 형식이라는 점입니다.

```
{
  "accountCounts": [
    {
      "businessUnitId": "6",
      "parentAccountNumber": "32001",
      "accountNumber": "42001",
      "totalOnlineContactsCount": 1,
      "countByPosition": {
        "taxProfessionalCount": 0,
        "attorneyCount": 0,
        "nonAttorneyCount": 1,
        "clerkCount": 0
      }
    },
    {
      "businessUnitId": "7",
      "parentAccountNumber": "32000",
      "accountNumber": "42000",
      "totalOnlineContactsCount": 1,
      "countByPosition": {
        "taxProfessionalCount": 0,
        "attorneyCount": 1,
        "nonAttorneyCount": 0,
        "clerkCount": 0
      }
    }
  ]
}
```

AWS Backup을 사용하여 계정 전반적으로 Amazon DynamoDB 테이블의 복사

작성자: Ramkumar Ramanujam(AWS)

환경: PoC 또는 파일럿

기술: 데이터베이스, 마이그레이션

AWS 서비스: Amazon DynamoDB, Backup

요약

Amazon Web Services(AWS)의 Amazon DynamoDB를 사용할 때 일반적인 사용 사례는 개발, 테스트 또는 스테이징 환경에서 DynamoDB 테이블을 복사하거나 프로덕션 환경에 있는 테이블 데이터와 동기화하는 것입니다. 표준 관행에 따라 각 환경은 서로 다른 계정을 사용합니다.

Backup은 DynamoDB, Amazon Simple Storage Service(S3) 및 기타 서비스에 대한 데이터의 교차 리전 및 교차 계정 백업과 복원을 지원합니다. 이 패턴은 AWS Backup 교차 계정 백업 및 복원을 사용하여 AWS 계정 간에 DynamoDB 테이블을 복사하는 절차를 제공합니다.

사전 조건 및 제한 사항

사전 조건

- 동일한 AWS Organizations 조직에 속한 두 개의 활성 계정
- 두 계정의 DynamoDB 테이블
- 백업 볼트를 생성하고 사용할 수 있는 Identity and Access Management(IAM) 권한

제한 사항

- 소스 및 대상 계정은 동일한 AWS Organizations 조직에 속해야 합니다.

아키텍처

대상 기술 스택

- AWS Backup
- Amazon DynamoDB

대상 아키텍처

1. 소스 계정의 AWS Backup 백업 볼트에 DynamoDB 테이블 백업을 생성합니다.
2. 백업을 대상 계정의 백업 볼트에 복사합니다.
3. 대상 계정 백업 저장소의 백업을 사용하여 대상 계정의 DynamoDb 테이블을 복원합니다.

자동화 및 규모 조정

AWS Backup을 사용하여 백업이 특정 간격으로 실행되도록 예약할 수 있습니다.

도구

- [AWS Backup](#) — Backup은 서비스, 클라우드, 온프레미스 전반적으로 데이터 보호를 중앙 집중화하고 자동화하는 완전 관리형 서비스입니다. 이 서비스를 사용하여 백업 정책을 구성하고 리소스에 대한 활동을 한 곳에서 구성할 수 있습니다. 이를 통해 이전에 수행했던 백업 작업을 자동화하고 통합할 수 있으며 service-by-service, 사용자 지정 스크립트와 수동 프로세스를 만들 필요가 없습니다.
- [Amazon DynamoDB](#) – Amazon DynamoDB는 완전관리형 NoSQL 데이터베이스 서비스로서 원활한 확장성과 함께 빠르고 예측 가능한 성능을 제공합니다.

에픽

소스 및 대상 계정에서 AWS Backup 기능을 활성화합니다.

작업	설명	필요한 기술
DynamoDB 및 교차 계정 백업을 위한 고급 기능을 활성화합니다.	소스 계정과 대상 계정 모두에서 다음을 수행합니다. <ol style="list-style-type: none"> 1. AWS Management Console에서 AWS Backup 콘솔을 엽니다. 2. 설정을 선택합니다. 3. Amazon DynamoDB 백업의 고급 기능에서, 고급 기능이 	AWS DevOps, 마이그레이션 엔지니어

작업	설명	필요한 기술
	<p>활성화되어 있는지 확인하거나 활성화를 선택합니다.</p> <p>4. 교차 계정 관리에서, 교차 계정 백업에 대하여 활성화를 선택합니다.</p>	

소스 및 대상 계정에 백업 볼트 생성

작업	설명	필요한 기술
백업 볼트를 생성합니다.	<p>소스 계정과 대상 계정 모두에서 다음을 수행합니다.</p> <ol style="list-style-type: none"> 1. AWS Backup 콘솔에서 백업 볼트를 선택합니다. 2. 백업 저장소 생성을 선택합니다. 3. 백업 볼트의 Amazon 리소스 이름(ARN)을 복사하여 저장합니다. <p>소스 계정과 대상 계정 간에 DynamoDB 테이블 백업을 복사할 때는 소스 및 대상 백업 볼트의 ARN이 모두 필요합니다.</p>	AWS DevOps, 마이그레이션 엔지니어

백업 볼트를 사용하여 백업 및 복원 수행

작업	설명	필요한 기술
소스 계정에서 DynamoDB 테이블 백업을 생성합니다.	소스 계정의 DynamoDB 테이블에 대한 백업을 생성하려면 다음을 수행합니다.	AWS DevOps, DBA, 마이그레이션 엔지니어

작업	설명	필요한 기술
	<ol style="list-style-type: none"> 1. AWS Backup 대시보드 페이지에서 온디맨드 백업 생성을 선택합니다. 2. 설정 섹션에서, 리소스 유형에 대하여 DynamoDB를 선택한 다음 테이블 이름을 선택합니다. 3. 백업 볼트 드롭다운 목록에서, 소스 계정에서 생성한 백업 볼트를 선택합니다. 4. 원하는 보존 기간을 선택합니다. 5. 온디맨드 백업 생성을 선택합니다. <p>새 백업 작업이 생성됩니다.</p> <p>백업 작업의 상태를 모니터링하려면 AWS Backup 작업 페이지에서 백업 작업 탭을 선택합니다. 이 탭에는 모든 활성, 진행 중, 완료 백업 작업이 나열됩니다.</p>	

작업	설명	필요한 기술
<p>소스 계정에서 대상 계정으로 백업을 복사합니다.</p>	<p>백업 작업이 완료되면 소스 계정의 백업 볼트에서 대상 계정의 백업 볼트로 DynamoDB 테이블 백업을 복사합니다.</p> <p>백업 볼트를 복사하려면 소스 계정에서 다음을 수행합니다.</p> <ol style="list-style-type: none"> 1. AWS Backup 콘솔에서 백업 볼트를 선택합니다. 2. 백업에서 DynamoDB 테이블 백업을 선택합니다. 3. [Actions], [Copy]를 선택합니다. 4. 대상 계정의 리전을 입력합니다. 5. 외부 볼트 ARN의 경우 대상 계정에서 생성한 백업 볼트의 ARN을 입력합니다. 6. 소스 계정에서 대상 계정으로 백업을 복사하려면 대상 계정 백업 볼트에서 다른 계정을 통한 액세스를 활성화합니다. 	<p>AWS DevOps, 마이그레이션 엔지니어, DBA</p>

작업	설명	필요한 기술
대상 계정의 백업을 복원합니다.	<p>대상 계정에서 다음을 수행합니다.</p> <ol style="list-style-type: none"> 1. AWS Backup 콘솔에서 백업 볼트를 선택합니다. 2. 백업에서 소스 계정을 통해 복사한 백업을 선택합니다. 3. 실행에서 복원을 선택합니다. 4. 복원하고자 하는 대상 DynamoDB 테이블의 이름을 입력합니다. 	AWS DevOps, DBA, 마이그레이션 엔지니어

관련 리소스

- [DynamoDB와 함께 AWS Backup 사용](#)
- [계정 전반적으로 백업 사본 생성](#)
- [AWS Backup 요금 책정](#)

사용자 지정 구현을 사용하여 계정 전반적으로 Amazon DynamoDB 테이블을 복사합니다.

작성자: Ramkumar Ramanujam(AWS)

환경: 프로덕션	소스: Amazon DynamoDB	대상: Amazon DynamoDB
R 유형: 해당 없음	워크로드: 기타 모든 워크로드	기술: 데이터베이스
AWS 서비스: Amazon DynamoDB		

요약

Amazon Web Services(AWS)의 Amazon DynamoDB를 사용할 때 일반적인 사용 사례는 개발, 테스트 또는 스테이징 환경에서 DynamoDB 테이블을 복사하거나 프로덕션 환경에 있는 테이블 데이터와 동기화하는 것입니다. 표준 관행에 따라 각 환경은 서로 다른 계정을 사용합니다.

DynamoDB는 이제 AWS Backup을 사용하여 교차 계정 백업을 지원합니다. AWS Backup을 사용할 때의 관련 스토리지 비용에 대한 자세한 내용은 [Backup 요금 책정](#)을 참조하십시오. AWS Backup을 사용하여 계정 전반적으로 복사를 하는 경우 소스 및 대상 계정이 Organizations 조직의 일부여야 합니다. Data Pipeline 또는 Glue와 같은 서비스를 사용하여 교차 계정 백업 및 복원을 위한 다른 솔루션도 있습니다. 그러나 이러한 솔루션을 사용하면 배포 및 유지 관리해야 할 서비스가 더 많기 때문에 애플리케이션 설치 공간이 늘어납니다.

Amazon DynamoDB Streams를 사용하여 소스 계정의 테이블 변경 내용을 캡처할 수도 있습니다. 그런 다음 Lambda 함수를 시작하고 대상 계정의 대상 테이블에서 해당 내용을 변경할 수 있습니다. 하지만 이 솔루션은 소스 테이블과 대상 테이블을 항상 동기화된 상태로 유지해야 하는 사용 사례에 적용됩니다. 데이터가 자주 업데이트되는 개발, 테스트 및 스테이징 환경에는 적용하지 못할 수 있습니다.

이 패턴은 한 계정에서 다른 계정으로 Amazon DynamoDB 테이블을 복사하기 위해 사용자 지정 솔루션을 구현하는 절차를 제공합니다. 이 패턴은 C#, Java, Python과 같은 일반적인 프로그래밍 언어를 사용하여 구현할 수 있습니다. [SDK](#)에 의해 지원되는 언어를 사용하는 것이 좋습니다.

사전 조건 및 제한 사항

사전 조건

- 두 개의 활성 상태 계정
- 두 계정 모두의 DynamoDB 테이블
- Identity and Access Management(IAM) 역할과 정책에 대한 지식
- C#, Java 또는 Python과 같은 모든 일반적인 프로그래밍 언어를 사용하여 Amazon DynamoDB 테이블에 액세스하는 방법에 대한 지식

제한 사항

이 패턴은 크기가 약 2GB 이하인 DynamoDB 테이블에 적용됩니다. 연결 또는 세션 중단, 조절, 장애 및 재시도를 처리하는 추가 로직이 있으면 더 큰 테이블에 이를 사용할 수 있습니다.

소스 테이블에서 항목을 읽는 DynamoDB 스캔 작업은 한 번의 호출로 최대 1 MB의 데이터만 가져올 수 있습니다. 크기가 2 GB를 초과하는 대형 테이블의 경우 이 제한으로 인해 전체 테이블 복사를 수행하는 데 걸리는 총 시간이 늘어날 수 있습니다.

아키텍처

자동화 및 규모 조정

이 패턴은 크기가 약 2GB 이하인 DynamoDB 테이블에 적용됩니다.

대형 테이블에 이 패턴을 적용하려면 다음 문제를 해결합니다.

- 테이블 복사 작업 중에는 서로 다른 보안 토큰을 사용하여 두 개의 활성 세션을 유지합니다. 테이블 복사 작업이 토큰 만료 시간보다 더 오래 걸리는 경우 보안 토큰을 새로 고치는 로직을 마련해야 합니다.
- 읽기 용량 단위(RCU) 및 쓰기 용량 단위(WCU)가 충분히 프로비저닝되지 않은 경우 소스 또는 대상 테이블에 대한 읽기 또는 쓰기가 제한될 수 있습니다. 이러한 예외를 파악하고 처리해야 합니다.
- 기타 실패나 예외를 처리하고, 복사 작업이 실패한 부분부터 다시 시도하거나 계속할 수 있는 재시도 메커니즘을 마련합니다.

도구

도구

- [Amazon DynamoDB](#) – Amazon DynamoDB는 완전관리형 NoSQL 데이터베이스 서비스로서 원활한 확장성과 함께 빠르고 예측 가능한 성능을 제공합니다.

- 필요한 추가 도구는 구현을 위해 선택하는 프로그래밍 언어에 따라 달라집니다. 예를 들어 C#을 사용하는 경우 Microsoft Visual Studio와 다음 NuGet 패키지가 필요합니다.
 - AWSSDK
 - AWSSDK.DynamoDBv2

code

다음 Python 코드 조각은 Boto3 라이브러리를 사용하여 DynamoDB 테이블을 삭제하고 다시 생성합니다.

장기 자격 증명이므로 IAM 사용자의 AWS_ACCESS_KEY_ID 및 AWS_SECRET_ACCESS_KEY(을)를 사용해서는 안 됩니다. 서비스에 프로그래밍 방식으로 액세스할 때는 이를 자제해야 합니다. 임시 보안 자격 증명에 관한 자세한 내용은 모범 사례 섹션을 확인하십시오.

다음 코드 조각에 사용되는 AWS_ACCESS_KEY_ID, AWS_SECRET_ACCESS_KEY, TEMPORARY_SESSION_TOKEN(은)는 Security Token Service(STS)에서 가져온 임시 자격 증명입니다.

```
import boto3
import sys
import json

#args = input-parameters = GLOBAL_SEC_INDEXES_JSON_COLLECTION,
    ATTRIBUTES_JSON_COLLECTION, TARGET_DYNAMODB_NAME, TARGET_REGION, ...

#Input param: GLOBAL_SEC_INDEXES_JSON_COLLECTION
#[{"IndexName":"Test-index", "KeySchema":[{"AttributeName":"AppId", "KeyType":"HASH"},
{"AttributeName":"AppType", "KeyType":"RANGE"}], "Projection":
{"ProjectionType":"INCLUDE", "NonKeyAttributes":["PK", "SK", "OwnerName", "AppVersion"]}]

#Input param: ATTRIBUTES_JSON_COLLECTION
#[{"AttributeName":"PK", "AttributeType":"S"},
{"AttributeName":"SK", "AttributeType":"S"},
{"AttributeName":"AppId", "AttributeType":"S"},
{"AttributeName":"AppType", "AttributeType":"N"}]

region = args['TARGET_REGION']
target_ddb_name = args['TARGET_DYNAMODB_NAME']

global_secondary_indexes = json.loads(args['GLOBAL_SEC_INDEXES_JSON_COLLECTION'])
attribute_definitions = json.loads(args['ATTRIBUTES_JSON_COLLECTION'])
```

```
# Drop and create target DynamoDB table
dynamodb_client = boto3.Session(
    aws_access_key_id=args['AWS_ACCESS_KEY_ID'],
    aws_secret_access_key=args['AWS_SECRET_ACCESS_KEY'],
    aws_session_token=args['TEMPORARY_SESSION_TOKEN'],
).client('dynamodb')

# Delete table
print('Deleting table: ' + target_ddb_name + ' ...')

try:
    dynamodb_client.delete_table(Table_name=target_ddb_name)

    #Wait for table deletion to complete
    waiter = dynamodb_client.get_waiter('table_not_exists')
    waiter.wait(Table_name=target_ddb_name)
    print('Table deleted.')
except dynamodb_client.exceptions.ResourceNotFoundException:
    print('Table already deleted / does not exist.')
    pass

print('Creating table: ' + target_ddb_name + ' ...')

table = dynamodb_client.create_table(
    Table_name=target_ddb_name,
    KeySchema=[
        {
            'AttributeName': 'PK',
            'KeyType': 'HASH' # Partition key
        },
        {
            'AttributeName': 'SK',
            'KeyType': 'RANGE' # Sort key
        }
    ],
    AttributeDefinitions=attribute_definitions,
    GlobalSecondaryIndexes=global_secondary_indexes,
    BillingMode='PAY_PER_REQUEST'
)

waiter = dynamodb_client.get_waiter('table_exists')
waiter.wait(Table_name=target_ddb_name)
```

```
print('Table created.')
```

모범 사례

임시 보안 인증

보안 모범 사례로서 프로그래밍 방식으로 AWS 서비스에 액세스하는 동안에는 IAM 사용자의 AND를 사용하지 않는 것이 좋습니다. 이는 장기 자격 증명이기 때문입니다. AWS_ACCESS_KEY_ID, AWS_SECRET_ACCESS_KEY 프로그래밍 방식으로 서비스에 액세스하려면 항상 임시 자격 증명을 사용합니다.

예를 들어 개발자는 개발 중에 애플리케이션에서 IAM 사용자의 AWS_ACCESS_KEY_ID 및 AWS_SECRET_ACCESS_KEY(을)를 하드코딩하지만, 코드 리포지토리에 변경 내용을 푸시하기 전에는 하드코딩된 값을 제거하지 못합니다. 이러한 노출된 자격 증명은 의도하지 않았거나 악의적인 사용자가 사용할 수 있으며, 이는 심각한 영향을 미칠 수 있습니다(특히 노출된 자격 증명에 관리자 권한이 있는 경우 더욱 그렇습니다). 이러한 노출된 자격 증명은 IAM 콘솔 또는 Command Line Interface(CLI)를 사용하여 즉시 비활성화하거나 삭제해야 합니다.

프로그래밍 방식으로 서비스에 액세스하려면 항상 임시 자격 증명을 사용합니다. 임시 자격 증명은 지정된 시간(15분~36시간) 동안만 유효합니다. 임시 자격 증명의 최대 허용 기간은 역할 설정 및 역할 함께 묶기와 같은 요인에 따라 달라집니다. STS에 대한 자세한 내용은 [설명서](#)를 참조하세요.

에픽

DynamoDB 테이블 설정

작업	설명	필요한 기술
DynamoDB 테이블을 생성합니다.	<p>소스 및 대상 AWS 계정 모두에서 인덱스가 포함된 DynamoDB 테이블을 생성합니다.</p> <p>용량 프로비저닝을 온디맨드 모드로 설정하면 DynamoDB가 워크로드에 따라 읽기/쓰기 용량을 동적으로 확장할 수 있습니다.</p>	앱 개발자, DBA, 마이그레이션 엔지니어

작업	설명	필요한 기술
	또는 4,000개의 RCU와 4,000개의 WCU가 있는 프로비저닝된 용량을 사용할 수 있습니다.	
소스 테이블을 채웁니다.	소스 계정의 DynamoDB 테이블을 테스트 데이터로 채웁니다. 최소 50 MB 이상의 테스트 데이터가 있으면 테이블 복사 중에 사용된 최대 RCU와 평균 RCU를 확인할 수 있습니다. 그런 다음 필요에 따라 용량 프로비저닝을 변경할 수 있습니다.	앱 개발자, DBA, 마이그레이션 엔지니어

DynamoDB 테이블에 액세스하기 위한 자격 증명을 설정합니다.

작업	설명	필요한 기술
소스 및 대상 DynamoDB 테이블에 액세스하기 위한 IAM 역할을 생성합니다.	<p>소스 계정의 DynamoDB 테이블에 액세스(읽기) 할 권한이 있는 IAM 역할을 소스 계정에 생성합니다.</p> <p>소스 계정을 이 역할의 신뢰할 수 있는 개체로 추가합니다.</p> <p>대상 계정의 DynamoDB 테이블에 액세스(생성, 읽기, 업데이트, 삭제)할 권한이 있는 IAM 역할과 함께 대상 계정에 생성합니다.</p> <p>대상 계정을 이 역할의 신뢰할 수 있는 개체로 추가합니다.</p>	앱 개발자, AWS DevOps

계정 간에 테이블 데이터 복사

작업	설명	필요한 기술
IAM 역할을 위한 임시 자격 증명을 가져옵니다.	<p>소스 계정에서 생성한 IAM 역할을 위한 임시 자격 증명을 가져옵니다.</p> <p>대상 계정에서 생성한 IAM 역할을 위한 임시 자격 증명을 가져옵니다.</p> <p>IAM 역할을 위한 임시 자격 증명을 가져오는 한 가지 방법은 CLI에서 STS를 사용하는 것입니다.</p> <pre data-bbox="594 898 1027 1213">aws sts assume-role --role-arn arn:aws:iam::<account-id>:role/<role-name> -- role-session-name <session-name> -- profile <profile-name></account-id></pre> <p>적절한 프로필(소스 또는 대상 계정에 해당)을 사용합니다.</p> <p>임시 보안 자격 증명에 관한 자세한 내용은 다음을 참조하십시오.</p> <ul data-bbox="594 1556 1027 1745" style="list-style-type: none"> • Security Token Service API Reference • CLI 액세스를 위한 IAM 역할 자격 증명 가져오기 	앱 개발자, 마이그레이션 엔지니어

작업	설명	필요한 기술
<p>소스 및 대상 DynamoDB 액세스를 위한 DynamoDB 클라이언트를 초기화합니다.</p>	<p>소스 및 대상 DynamoDB 테이블에 대해, SDK에서 제공하는 DynamoDB 클라이언트를 초기화합니다.</p> <ul style="list-style-type: none"> • 소스 DynamoDB 클라이언트의 경우 소스 계정에서 가져온 임시 자격 증명을 사용합니다. • 대상 DynamoDB 클라이언트의 경우 대상 계정에서 가져온 임시 자격 증명을 사용합니다. <p>IAM 임시 자격 증명을 사용하여 요청하는 방법에 대한 자세한 내용은 설명서를 참조하십시오.</p>	<p>앱 개발자</p>

작업	설명	필요한 기술
<p>대상 테이블을 삭제하고 다시 생성합니다.</p>	<p>대상 계정 DynamoDB 클라이언트를 사용하여 대상 계정에서 대상 DynamoDB 테이블(인덱스 포함)을 삭제하고 다시 생성합니다.</p> <p>DynamoDB 테이블에서 모든 레코드를 삭제하는 작업은 프로비저닝된 WCU를 사용하기 때문에 비용이 많이 드는 작업입니다. 테이블을 삭제하고 다시 생성하면 이러한 추가 비용을 피할 수 있습니다.</p> <p>테이블을 생성한 후 테이블에 인덱스를 추가할 수 있지만, 이 경우 2~5분 정도 더 걸립니다. 인덱스 컬렉션을 <code>createTable</code> 호출에 전달하여 테이블을 생성하는 동안 인덱스를 생성하는 것이 더 효율적입니다.</p>	<p>앱 개발자</p>

작업	설명	필요한 기술
테이블 복사를 수행합니다.	<p>모든 데이터가 복사될 때까지 다음 절차를 반복합니다.</p> <ul style="list-style-type: none"> • 소스 DynamoDB 클라이언트를 사용하여 소스 계정의 테이블을 스캔합니다. 각 DynamoDB 스캔은 테이블에서 1 MB의 데이터만 검색하므로 모든 항목 또는 레코드를 읽을 때까지 이 작업을 반복해야 합니다. • 스캔한 각 항목 세트의 경우 DynamoDB용 SDK의 <code>BatchWriteItem</code> 호출을 사용하여 대상 DynamoDB 클라이언트와 함께 대상 계정의 테이블에 항목을 기록합니다. 따라서 DynamoDB에 대한 <code>PutItem</code> 요청 수가 줄어듭니다. • <code>BatchWriteItem</code> (은)는 쓰기 또는 올리기가 25개로 제한되거나 최대 16MB로 제한됩니다. <code>BatchWriteItem</code> 호출 전에 스캔한 항목을 25개까지 누적하는 로직을 추가해야 합니다. <code>BatchWriteItem</code> (은)는 성공적으로 복사하지 못한 항목의 목록을 반환합니다. 이 목록을 사용하여, 성공하지 못한 항목만으로 다른 <code>BatchWriteItem</code> 호출을 	앱 개발자

작업	설명	필요한 기술
	<p>수행하기 위한 재시도 로직을 추가합니다.</p> <p>자세한 내용은 첨부 파일 섹션에서 C#의 참조 구현(테이블 삭제, 생성, 채우기에 사용)을 참조하십시오. 예제 테이블 구성 JavaScript 객체 표기법(JSON) 파일도 첨부되어 있습니다.</p>	

관련 리소스

- [Amazon DynamoDB 설명서](#)
- [계정에서 IAM 사용자 생성](#)
- [SDK](#)
- [리소스에서 임시 자격 증명 사용](#)

추가 정보

이 패턴은 200,000개의 항목(평균 항목 크기 5 KB, 테이블 크기 250 MB) 이 있는 DynamoDB 테이블을 복사하기 위해 C#을 사용하여 구현되었습니다. 대상 DynamoDB 테이블은 RCU 4,000개와 WCU 4,000개의 프로비저닝된 용량으로 설정되었습니다.

테이블 삭제 및 재생성을 포함한 전체 테이블 복사 작업(소스 계정에서 대상 계정으로 복사)에는 5분이 걸렸습니다. 사용된 총 용량 단위: 30,000개의 RCU 및 약 400,000개의 WCU.

DynamoDB 용량 모드에 대한 자세한 내용은 AWS 설명서에서 [읽기/쓰기 용량 모드](#)를 참조하십시오.

첨부

이 문서와 관련된 추가 콘텐츠에 액세스하려면 [attachment.zip](#) 파일의 압축을 풉니다.

Amazon RDS 및 Amazon Aurora에 대한 자세한 비용 및 사용 보고서 생성

작성자: Lakshmanan Lakshmanan(AWS), Sudarshan Narasimhan

환경: 프로덕션

기술: 데이터베이스, 비용 관리, 분석

AWS 서비스: Amazon Athena, Amazon Aurora, Amazon RDS, AWS 과금 정보 및 비용 관리

요약

이 패턴은 [사용자 정의 비용 할당 태그](#)를 구성하여 Amazon Relational Database Service(RDS) 또는 Amazon Aurora 클러스터가 사용하는 비용을 추적하는 방법을 보여줍니다. 이 태그를 사용하여 AWS Cost Explorer에서 여러 차원의 클러스터에 대한 자세한 비용 및 사용 보고서를 생성할 수 있습니다. 예를 들어 팀, 프로젝트 또는 비용 센터 수준에서 사용 비용을 추적한 다음 Amazon Athena에서 이 데이터를 분석할 수 있습니다.

사전 조건 및 제한 사항

사전 조건

- 활성 상태의 AWS 계정.
- 하나 이상의 [Amazon RDS](#) 또는 [Amazon Aurora](#) 인스턴스

제한 사항

태그 지정 제한은 [AWS Billing 사용 설명서](#)를 참조하세요.

아키텍처

대상 기술 스택

- Amazon RDS 또는 Amazon Aurora
- AWS 비용 및 사용 보고서
- AWS Cost Explorer

- Amazon Athena

워크플로 및 아키텍처

태깅 및 분석 워크플로는 다음 단계로 구성됩니다.

1. 데이터 엔지니어, 데이터베이스 관리자 또는 AWS 관리자는 Amazon RDS 또는 Aurora 클러스터에 대한 사용자 정의 비용 할당 태그를 생성합니다.
2. AWS 관리자가 태그를 활성화합니다.
3. 태그는 AWS Cost Explorer에 메타데이터를 보고합니다.
4. 데이터 엔지니어, 데이터베이스 관리자 또는 AWS 관리자가 [월별 비용 할당 보고서](#)를 생성합니다.
5. 데이터 엔지니어, 데이터베이스 관리자 또는 AWS 관리자는 Amazon Athena를 사용하여 월별 비용 할당 보고서를 분석합니다.

다음 다이어그램은 Amazon RDS 또는 Aurora 인스턴스의 사용 비용을 추적하기 위해 태그를 적용하는 방법을 보여줍니다.

다음 아키텍처 다이어그램은 비용 할당 보고서가 분석을 위해 Amazon Athena와 통합되는 방법을 보여줍니다.

월별 비용 할당 보고서는 지정한 Amazon S3 버킷에 저장됩니다. 에픽 섹션에 설명된 대로 AWS CloudFormation 템플릿으로 Athena를 설정하면 템플릿은 AWS Glue 크롤러, AWS Glue 데이터베이스, 아마존 단순 알림 시스템 (Amazon SNS) 이벤트, AWS Lambda 함수, Lambda 함수에 대한 AWS ID 및 액세스 관리 (IAM) 역할 등 몇 가지 추가 리소스를 제공합니다. 새 비용 데이터 파일이 S3 버킷에 도착하면 이벤트 알림을 사용하여 이러한 파일을 Lambda 함수로 전달하여 처리합니다. Lambda 함수는 AWS Glue 데이터 카탈로그에서 테이블을 생성하거나 업데이트하는 AWS Glue 크롤러 작업을 시작합니다. 그런 다음 이 테이블을 사용하여 Athena에서 데이터를 쿼리합니다.

도구

- [Amazon Athena](#)는 표준 SQL을 사용해 Amazon S3에 저장된 데이터를 간편하게 분석할 수 있는 대화식 쿼리 서비스입니다.

- [Amazon Aurora](#)는 MySQL 및 PostgreSQL과 호환되는 완전 관리형 관계형 데이터베이스 엔진입니다.
- [Amazon Relational Database Service\(RDS\)](#)는 AWS 클라우드에서 관계형 데이터베이스를 설정, 운영 및 조정하는 데 도움이 됩니다.
- [CloudFormationAWS](#)는 [AWS](#) 및 타사 리소스를 쉽게 모델링, 프로비저닝 및 관리할 수 있는 코드형 인프라 (IaC) 서비스입니다.
- [AWS Cost Explorer](#)는 AWS 비용 및 사용량을 보고 분석할 수 있도록 지원합니다.

에픽

Amazon RDS 또는 Aurora 클러스터용 태그를 생성하고 활성화

작업	설명	필요한 기술
Amazon RDS 또는 Aurora 클러스터에 대한 사용자 정의 비용 할당 태그를 생성합니다.	<p>새로 생성했거나 기존에 사용하던 Amazon RDS 또는 Aurora 클러스터에 태그를 추가하려면 Amazon Aurora 사용 설명서의 태그 추가, 나열 및 제거 지침을 따르세요.</p> <p>참고: Amazon Aurora 클러스터를 설정하는 방법에 대한 자세한 내용은 Amazon Aurora 사용 설명서의 MySQL 및 PostgreSQL에 대한 지침을 참조하세요.</p>	AWS 관리자, 데이터 엔지니어, DBA
사용자 정의 비용 할당 태그를 활성화합니다.	AWS 결제 사용 설명서의 사용자 정의 비용 할당 태그 활성화 의 지침을 따르세요.	AWS 관리자

비용 및 사용 보고서 생성

작업	설명	필요한 기술
클러스터에 대한 비용 및 사용 보고서를 만들고 구성합니다.	<ol style="list-style-type: none"> 1. AWS Management Console에 로그인하고 AWS 빌링 콘솔을 엽니다. 2. 탐색 창에서 비용 및 사용 보고서를 선택합니다. 3. 보고서 생성을 선택합니다. 4. 보고서 이름을 제공하고 다른 옵션에 대한 기본 설정을 유지한 후 다음을 선택합니다. 5. 구성을 선택하고 기존 S3 버킷의 세부 정보를 제공합니다. 이 화면에서 새 S3 버킷을 생성하도록 선택할 수도 있습니다. 다음을 선택합니다. 6. 버킷에 적용할 기본 정책을 확인하고 확인란에 체크한 다음 저장을 선택합니다. 7. 보고서 경로 접두사에 보고서 이름 앞에 추가할 접두사를 지정합니다. 8. 시간 세분화를 위해서는 보고서에 사용할 데이터를 수집하려는 빈도에 따라 시간별, 일별 또는 월별을 선택합니다. 9. 보고서 버전 관리의 경우 새 버전의 보고서를 별도로 만들지 아니면 기존 보고서를 	앱 소유자, AWS 관리자, DBA, 일반 AWS, 데이터 엔지니어

작업	설명	필요한 기술
	<p>각 버전으로 덮어쓸지 선택합니다.</p> <p>10보고서 데이터 통합 활성화에서 Amazon Athena를 선택합니다. 압축 유형이 Parquet으로 설정되어 있는지 확인하세요.</p> <p>11다음을 선택합니다.</p> <p>12보고서 설정을 검토한 후 검토 후 완료를 선택합니다.</p> <p>데이터는 24시간 후에 사용할 수 있습니다.</p>	

비용 및 사용 보고서 데이터 분석

작업	설명	필요한 기술
<p>비용 및 사용 보고서 데이터를 분석하세요.</p>	<ol style="list-style-type: none"> Athena를 설정하고 사용하여 보고서 데이터를 분석하세요. 지침은 AWS 비용 및 사용 보고서 사용 설명서의 Amazon Athena를 사용하여 비용 및 사용 보고서 쿼리를 참조하세요. Athena에서 제공하는 AWS CloudFormation 템플릿을 사용하는 것이 좋습니다. Athena 쿼리를 실행합니다. 예를 들어 다음 SQL 쿼리를 사용하여 데이터 새로 고침 상태를 확인할 수 있습니다. 	<p>앱 소유자, AWS 관리자, DBA, 일반 AWS, 데이터 엔지니어</p>

작업	설명	필요한 기술
	<pre data-bbox="592 220 1027 367">select status from cost_and_usage_data_status</pre> <p data-bbox="592 405 1027 535">자세한 내용은 사용 설명서의 비용 및 사용 보고서 생성을 참조하세요.</p> <p data-bbox="592 577 1027 756">참고: SQL 쿼리를 실행할 때 드롭다운 목록에서 올바른 데이터베이스를 선택했는지 확인합니다.</p>	

관련 리소스

참조

- [AWS CloudFormation 템플릿을 사용하여 Athena를 설정하는 방법 \(권장\)](#)
- [Athena를 수동으로 설정](#)
- [Amazon Athena 쿼리 실행](#)
- [보고서 데이터를 다른 리소스로 로드](#)

자습서 및 동영상

- [Amazon Athena를 사용한 비용 및 사용 보고서 분석 \(동영상\)](#) YouTube

Aurora PostgreSQL의 사용자 지정 엔드포인트를 사용하여 Oracle RAC 워크로드 에뮬레이션하기

작성자: HariKrishna 보가다 (AWS)

환경: PoC 또는 파일럿	소스: 데이터베이스: 관계형	대상: Aurora PostgreSQL
R 타입: 리플랫폼	워크로드: Oracle	기술: 데이터베이스, 마이그레이션
AWS 서비스: 아마존 오로라, 아마존 CloudWatch		

요약

이 패턴은 단일 클러스터 내의 인스턴스 간에 워크로드를 분산하는 사용자 지정 엔드포인트가 있는 Amazon Aurora PostgreSQL-Compatible Edition을 사용하여 Oracle Real Application Clusters(Oracle RAC) 워크로드에서 서비스를 에뮬레이션하는 방법을 설명합니다. 이 패턴은 Amazon Aurora 데이터베이스의 [사용자 지정 엔드포인트](#)를 생성하는 방법을 보여줍니다. 사용자 지정 엔드포인트를 사용하면 Aurora 클러스터의 다양한 DB 인스턴스 세트에 워크로드를 분산하고 로드 밸런싱할 수 있습니다.

Oracle RAC 환경에서는 [서비스](#)가 하나 이상의 인스턴스에 걸쳐 있을 수 있으며 트랜잭션 성능을 기반으로 워크로드 밸런싱을 촉진할 수 있습니다. 서비스 기능에는 end-to-end 무인 복구, 워크로드별 순차적 변경, 완전한 위치 투명성 등이 포함됩니다. 이 패턴을 사용하여 이러한 기능 중 일부를 에뮬레이션할 수 있습니다. 예를 들어 보고 애플리케이션의 연결을 라우팅하는 기능을 에뮬레이션할 수 있습니다.

사전 조건 및 제한 사항

사전 조건

- 활성 상태의 AWS 계정.
- [PostgreSQL JDBC 드라이버](#)
- [Aurora PostgreSQL-Compatible 데이터베이스](#)
- Aurora PostgreSQL-Compatible 데이터베이스로 마이그레이션된 Oracle RAC 데이터베이스

제한 사항

- 사용자 지정 엔드포인트에 적용되는 제한 사항은 Amazon RDS 설명서의 [사용자 지정 엔드포인트에 대해 속성 지정하기](#)를 참고하십시오.

아키텍처

소스 기술 스택

- 3-노드 Oracle RAC 데이터베이스

대상 기술 스택

- 읽기 전용 복제본이 두 개 있는 Aurora PostgreSQL-Compatible 데이터베이스

소스 아키텍처

다음 다이어그램은 3-노드 Oracle RAC 데이터베이스의 아키텍처를 보여줍니다.

대상 아키텍처

다음 다이어그램은 읽기 전용 복제본 2개가 있는 Aurora PostgreSQL-Compatible 데이터베이스의 아키텍처를 보여줍니다. 서로 다른 세 가지 애플리케이션/서비스가 사용자 지정 엔드포인트를 사용하고 있는데, 이 엔드포인트는 서로 다른 애플리케이션 사용자에게 서비스를 제공하고 기본 복제본과 읽기 전용 복제본 간에 트래픽과 부하를 리디렉션합니다.

도구

- [Amazon Aurora PostgreSQL-Compatible Edition](#)은 PostgreSQL 배포를 설정, 운영 및 확장할 수 있는 완전 관리형 ACID 호환 관계형 데이터베이스 엔진입니다.
- [Amazon](#)은 AWS 리소스의 지표와 AWS에서 실행하는 애플리케이션을 실시간으로 모니터링할 수 있도록 CloudWatch 도와줍니다.
- [Amazon Relational Database Service\(Amazon RDS\) for PostgreSQL](#)은 AWS 클라우드에서 PostgreSQL 관계형 데이터베이스를 설정, 운영 및 조정하는 데 도움이 됩니다.
- [AWS Command Line Interface\(AWS CLI\)](#)는 명령줄 쉘에서 명령을 사용하여 AWS 서비스와 상호 작용할 수 있는 오픈 소스 도구입니다.

에픽

Aurora PostgreSQL-Compatible 클러스터 생성

작업	설명	필요한 기술
클러스터를 생성합니다.	클러스터를 생성하려면 Amazon RDS 설명서의 DB 클러스터 생성 및 Aurora PostgreSQL DB 클러스터의 데이터베이스에 연결 을 참조하세요.	AWS 관리자
워크로드에 대한 사용자 지정 파라미터 그룹을 생성합니다.	파라미터 그룹을 만들려면 Amazon RDS 설명서의 DB 클러스터 파라미터 그룹 생성하기 를 참조하세요.	AWS 관리자
이벤트 알림 및 경보를 생성합니다.	이벤트 알림 및 Amazon CloudWatch 경보를 사용하여 클러스터의 상태가 변경될 때 이를 알리고 사전 정의된 임계값에 도달하면 지표를 캡처할 수 있습니다. CloudWatch 경보를 생성하려면 설명서의 정적 임계값을 기반으로 CloudWatch 경보 생성 을 참조하십시오. CloudWatch 이벤트 알림을 만들려면 CloudWatch 설명서에서 CloudWatch 이벤트를 트리거하는 이벤트 규칙 만들기를 참조하십시오.	AWS 관리자

복제본을 Aurora PostgreSQL-Compatible DB 클러스터에 추가

작업	설명	필요한 기술
클러스터에 읽기 복제본을 추가합니다.	<ol style="list-style-type: none"> 읽기 복제본을 생성합니다. DB 클러스터가 있는 동일한 가용 영역에 읽기 전용 복제본을 추가합니다. 참고: 장애 조치 노드에 대한 요구 사항을 충족해야 하는 경우 다른 가용 영역을 사용할 수 있습니다. 	AWS 관리자
읽기 전용 복제본 엔드포인트를 기록합니다.	나중에 사용자 지정 엔드포인트를 만들 때 사용할 수 있도록 읽기 전용 복제본 엔드포인트를 문서화합니다.	AWS 관리자

사용자 지정 엔드포인트 생성

작업	설명	필요한 기술
사용자 지정 엔드포인트 이름을 입력합니다.	필요한 각 엔드포인트에 대해 워크로드 또는 애플리케이션과 관련된 고유한 엔드포인트 이름을 생성합니다.	AWS 관리자
엔드포인트 구성원을 추가합니다.	읽기 전용 복제본 엔드포인트를 사용자 지정 그룹에 추가합니다. 자세한 내용은 Amazon RDS 설명서의 사용자 지정 엔드포인트 편집하기 를 참고하십시오.	AWS 관리자
(선택 사항) 클러스터에 향후 인스턴스를 추가합니다.	사용자 지정 그룹에 더 많은 복제본이나 엔드포인트를 추가하려면 Amazon RDS 설명서의	AWS 관리자

작업	설명	필요한 기술
	DB 클러스터에 Aurora 복제본 추가하기 를 참고하십시오.	
엔드포인트를 생성합니다.	엔드포인트를 생성하려면 Amazon RDS 설명서의 사용자 지정 엔드포인트 생성하기 를 참고하십시오.	AWS 관리자

사용자 지정 엔드포인트를 사용하여 애플리케이션 연결을 테스트합니다.

작업	설명	필요한 기술
사용자 지정 엔드포인트 세부 정보를 워크로드를 가리키는 애플리케이션과 공유합니다.	테스트하려는 보고 애플리케이션의 데이터베이스 연결 세부 정보에 사용자 지정 엔드포인트 세부 정보를 추가합니다.	AWS 관리자
사용자 지정 엔드포인트를 사용하여 워크로드를 연결합니다.	보고 애플리케이션에서 사용자 지정 엔드포인트 세부 정보를 확인합니다.	AWS 관리자
데이터베이스에서 연결 세부 정보를 확인합니다.	<ol style="list-style-type: none"> 1. 애플리케이션의 사용자 이름과 연결 수를 테스트합니다. 2. 워크로드 전반의 부하 분산을 확인하여 다양한 사용자 지정 엔드포인트(기본 및 읽기 전용 복제본)에 연결이 분산되어 있는지 확인하십시오. 	AWS 관리자

관련 리소스

- [Aurora 엔드포인트 유형](#)

- [사용자 지정 엔드포인트의 멤버십 규칙](#)
- [E 사용자 nd-to-end 지정 엔드포인트에 대한 AWS CLI 예제](#)
- [Oracle RAC의 대안으로서의 Amazon Aurora](#)
- [Oracle에서 PostgreSQL로 마이그레이션할 때의 문제점과 이를 극복하는 방법](#)

Amazon RDS에서 PostgreSQL DB 인스턴스에 대한 암호화된 연결 활성화하기

작성자: Rohit Kapoor (AWS)

환경: PoC 또는 파일럿

기술: 네트워킹; 보안, 자격 증명, 규정 준수

워크로드: 오픈 소스

AWS 서비스: Amazon RDS, Amazon Aurora

요약

Amazon Relational Database Service(Amazon Relational Database Service)는 PostgreSQL DB 인스턴스에 대한 SSL 암호화를 지원합니다. SSL을 사용하여 애플리케이션과 Amazon RDS for PostgreSQL DB 인스턴스 사이의 PostgreSQL 연결을 암호화할 수 있습니다. 기본적으로 Amazon RDS for PostgreSQL는 SSL/TLS를 사용하며 모든 클라이언트가 SSL/TLS 암호화를 사용하여 연결하기를 기대합니다. Amazon RDS for PostgreSQL은 TLS 버전 1.1 및 1.2를 지원합니다.

이 패턴은 Amazon RDS for PostgreSQL DB instance 인스턴스의 암호화된 연결을 활성화하는 방법을 설명합니다. 동일한 프로세스를 사용하여 Amazon Aurora PostgreSQL-Compatible Edition에서 암호화된 연결을 활성화할 수 있습니다.

사전 조건 및 제한 사항

- 활성 상태의 AWS 계정
- [Amazon RDS for PostgreSQL DB 인스턴스](#)
- [SSL 번들](#)

아키텍처

도구

- [pgAdmin](#)은 PostgreSQL을 위한 오픈 소스 관리 및 개발 플랫폼입니다. Linux, Unix, macOS, Windows에서 pgAdmin을 사용하여 PostgreSQL 10 이상에서 데이터베이스 객체를 관리할 수 있습니다.
- [PostgreSQL 편집기](#)는 쿼리를 생성, 개발 및 실행하고 요구 사항에 따라 코드를 편집하는 데 도움이 되는 보다 사용자 친화적인 인터페이스를 제공합니다.

모범 사례

- 비보안 데이터베이스 연결을 모니터링합니다.
- 데이터베이스 액세스 권한을 감사합니다.
- 백업 및 스냅샷이 저장 시 암호화되는지 확인하도록 하십시오.
- 데이터베이스 액세스를 모니터링합니다.
- 무제한 액세스 그룹은 피하도록 하십시오.
- [Amazon](#)을 통해 알림을 향상시키십시오 GuardDuty.
- 정책 준수를 정기적으로 모니터링합니다.

에픽

신뢰할 수 있는 인증서를 다운로드하여 트러스트 스토어로 가져옵니다.

작업	설명	필요한 기술
컴퓨터에 신뢰할 수 있는 인증서를 로드합니다.	<p>컴퓨터의 신뢰할 수 있는 루트 인증 기관 스토어에 인증서를 추가하려면 다음 단계를 따르십시오. (이 지침에서는 Window Server를 예로 사용합니다.)</p> <ol style="list-style-type: none"> 1. Windows Server에서 시작, 실행을 선택한 다음 mmc를 입력합니다. 	DevOps 엔지니어, 마이그레이션 엔지니어, DBA

작업	설명	필요한 기술
	<ol style="list-style-type: none"> 2. 콘솔에서 파일, 스냅인 추가/제거를 선택합니다. 3. 사용 가능한 스냅인에서 인증서를 선택한 다음 추가를 선택합니다. 4. 이 스냅인이 항상 인증서를 관리에서 컴퓨터 계정, 다음을 선택합니다. 5. 로컬 컴퓨터를 선택하고 마침을 선택합니다. 6. 콘솔에 추가할 스냅인이 더 이상 없는 경우 확인을 선택합니다. 7. 콘솔 트리에서 인증서를 두 번 클릭합니다. 8. 신뢰할 수 있는 루트 인증 기관을 마우스 오른쪽 버튼으로 클릭합니다. 9. 다운로드한 인증서를 가져오려면 모든 작업, 가져오기를 선택합니다. 10. 인증서 가져오기 마법사의 단계를 따릅니다. 	

SSL 연결 강제 설정

작업	설명	필요한 기술
<p>매개변수 그룹을 생성하고 <code>rds.force_ssl</code> 매개변수를 설정합니다.</p>	<p>PostgreSQL DB 인스턴스에 사용자 지정 매개변수 그룹이 있는 경우 매개변수 그룹을 편집하고 <code>rds.force_ssl</code> 을 1로 변경합니다.</p>	<p>DevOps 엔지니어, 마이그레이션 엔지니어, DBA</p>

작업	설명	필요한 기술
	<p>DB 인스턴스에서 <code>rds.force_ssl</code> 이 활성화되지 않은 기본 매개변수 그룹을 사용하는 경우 새 매개변수 그룹을 생성합니다. Amazon RDS API를 사용하거나 다음 지침에 따라 수동으로 새 매개변수 그룹을 수정할 수 있습니다.</p> <p>새 파라미터 그룹을 생성하려면 다음을 수행하세요.</p> <ol style="list-style-type: none"> 1. AWS 관리 콘솔에 로그인하고 DB 인스턴스를 호스팅하는 AWS 리전의 Amazon RDS 콘솔을 엽니다. 2. 탐색 창에서 파라미터 그룹을 선택합니다. 3. 매개변수 그룹 생성을 선택하고 다음 값을 설정합니다. <ul style="list-style-type: none"> • 파라미터 그룹 패밀리에서 <code>postgres14</code>를 선택합니다. • 그룹 이름에 <code>pgsql-<database_instance>-ssl</code>을 입력합니다. • 설명에 추가하려는 매개변수 그룹에 대한 자유 형식 설명을 입력합니다. • 생성을 선택하세요. 4. 생성한 파라미터 그룹을 선택합니다. 5. 파라미터 그룹 작업에서 편집을 선택합니다. 	

작업	설명	필요한 기술
	<p>6. <code>rds.force_ssl</code>을 찾아 설정을 1로 변경합니다.</p> <p>참고: 이 매개변수를 변경하기 전에 클라이언트측 테스트를 수행하십시오.</p> <p>7. 변경 사항 저장을 선택합니다.</p> <p>파라미터 그룹을 PostgreSQL DB 인스턴스에 연결하려면 다음을 수행하세요.</p> <ol style="list-style-type: none"> 1. Amazon RDS 콘솔의 탐색창에서 데이터베이스를 선택한 다음, PostgreSQL DB 인스턴스를 선택합니다. 2. 수정을 선택합니다. 3. 추가 구성에서 새 매개변수 그룹을 선택한 다음 계속을 선택합니다. 4. 수정 사항 예약에서 즉시 적용을 선택합니다. 5. Modify DB instance(DB 인스턴스 수정)를 선택합니다. <p>자세한 내용은 Amazon RDS 설명서를 참조하세요.</p>	

작업	설명	필요한 기술
SSL 연결을 강제 설정합니다.	Amazon RDS for PostgreSQL DB 인스턴스에 연결합니다. SSL을 사용하지 않는 연결 시도는 오류 메시지와 함께 거부됩니다. 자세한 내용은 Amazon RDS 설명서 를 참조하세요.	DevOps 엔지니어, 마이그레이션 엔지니어, DBA

SSL 확장 설치

작업	설명	필요한 기술
SSL 확장을 설치합니다.	<ol style="list-style-type: none"> psql 또는 pgAdmin 연결을 DBA로 시작합니다. ssl_is_used() 함수를 호출하여 SSL이 사용 중인지 확인합니다. <div style="border: 1px solid #ccc; border-radius: 10px; padding: 10px; margin: 10px 0;"> <pre>select ssl_is_used();</pre> </div> <p>이 함수는 연결이 SSL을 사용할 경우 t를 반환하고, 그렇지 않으면 f를 반환합니다.</p> SSL 확장을 설치합니다. <div style="border: 1px solid #ccc; border-radius: 10px; padding: 10px; margin: 10px 0;"> <pre>create extension sslinfo; show ssl; select ssl_cipher();</pre> </div> <p>자세한 내용은 Amazon RDS 설명서를 참조하세요.</p> 	DevOps 엔지니어, 마이그레이션 엔지니어, DBA

SSL에 대한 PostgreSQL 클라이언트를 구성합니다.

작업	설명	필요한 기술
<p>SSL을 사용하도록 클라이언트를 구성합니다.</p>	<p>SSL을 사용하면 TLS 프로토콜을 사용하는 암호화된 연결을 지원하는 PostgreSQL 서버를 시작할 수 있습니다. 서버는 동일한 TCP 포트에서 표준 연결과 SSL 연결을 모두 수신하고 연결 클라이언트와 SSL 사용 여부를 협상합니다. 기본적으로, 이는 클라이언트 옵션입니다.</p> <p>psql 클라이언트를 사용 중인 경우,</p> <ol style="list-style-type: none"> 1. Amazon RDS 인증서가 로컬 컴퓨터에 로드되었는지 확인하십시오. 2. 다음을 추가하여 SSL 클라이언트 연결을 시작합니다. <pre data-bbox="634 1241 1029 1598"> psql postgres -h SOMEHOST.amazonaws .com -p 8192 -U someuser sslmode=v erify-full sslrootce rt=rds-ssl-ca-cert .pem select ssl_cipher(); </pre> <p>기타 PostgreSQL 클라이언트의 경우,</p> <ul style="list-style-type: none"> • 해당 애플리케이션 공개 키 매개변수를 수정하십시오. 	<p>DevOps 엔지니어, 마이그레이션 엔지니어, DBA</p>

작업	설명	필요한 기술
	<p>이는 GUI 도구의 연결 페이지에서 옵션, 연결 문자열의 일부 또는 속성으로 사용할 수 있습니다.</p> <p>이러한 클라이언트에 대해 다음 페이지를 검토하십시오.</p> <ul style="list-style-type: none"> • pgAdmin 설명서 • JDBC 설명서 	

문제 해결

문제	Solution
SSL 인증서를 다운로드할 수 없습니다.	웹사이트 연결을 확인하고, 로컬 컴퓨터에 인증서를 다시 다운로드해 보십시오.

관련 리소스

- [Amazon RDS for PostgreSQL 설명서](#)
- [PostgreSQL DB 인스턴스와 함께 SSL 사용](#) (Amazon RDS 설명서)
- [SSL을 사용한 안전한 TCP/IP 연결](#) (PostgreSQL 설명서)
- [SSL 사용](#) (JDBC 설명서)

기존 Amazon RDS for PostgreSQL DB 인스턴스 암호화하기

피유시 고얏(AWS), 쇼바나 라구(AWS), 야세르 라자 (AWS)가 제작했습니다.

환경: 프로덕션

기술: 데이터베이스, 보안, 자격 증명, 규정 준수

AWS 서비스: Amazon RDS, AWS KMS, AWS DMS

요약

이 패턴은 Amazon Web Services(AWS) 클라우드에서 가동 중지 시간을 최소화하면서 PostgreSQL 용 Amazon Relational Database Service(RDS) 단원을 암호화하는 방법을 설명합니다. 이 프로세스는 Amazon RDS for MySQL DB 인스턴스에서 사용할 수 있습니다.

Amazon RDS DB 인스턴스를 생성할 때 암호화를 사용하도록 설정할 수 있지만, 생성된 후에는 사용할 수 없습니다. 하지만 DB 인스턴스의 스냅샷을 만든 다음 해당 스냅샷의 암호화된 사본을 만들어 암호화되지 않은 DB 인스턴스에 암호화를 추가할 수 있습니다. 그런 다음 암호화된 스냅샷에서 DB 인스턴스를 복원하여 원본 DB 인스턴스의 암호화된 사본을 얻을 수 있습니다. 이 작업 중에 프로젝트가 다운타임 (최소한 쓰기 트랜잭션의 경우) 을 허용하는 경우 이 작업만 수행하면 됩니다. 암호화된 새 DB 인스턴스 사본을 사용할 수 있게 되면 애플리케이션이 새 데이터베이스를 가리키도록 할 수 있습니다. 그러나 프로젝트에서 이러한 활동으로 인한 심각한 가동 중지 시간이 허용되지 않는 경우 가동 중지 시간을 최소화하는 데 도움이 되는 대체 접근 방식이 필요합니다. 이 패턴은 AWS Database Migration Service(AWS DMS)를 사용하여 데이터를 마이그레이션하고 지속적으로 복제하므로 가동 중지 시간을 최소화하면서 암호화된 새 데이터베이스로 전환할 수 있습니다.

Amazon RDS 암호화 DB 인스턴스는 업계 표준인 AES-256 암호화 알고리즘을 사용하여 Amazon RDS DB 인스턴스를 호스팅하는 서버의 데이터를 암호화합니다. 데이터가 암호화되면 Amazon RDS 는 성능에 미치는 영향을 최소화하면서 데이터의 액세스 인증 및 암호 해독을 투명하게 처리합니다. 암호화를 사용하도록 데이터베이스 클라이언트 애플리케이션을 수정하지 않아도 됩니다.

사전 조건 및 제한 사항

사전 조건

- 활성 상태의 AWS 계정.
- 암호화 해제된 Amazon RDS for PostgreSQL DB 인스턴스

- AWS DMS 작업을 사용한 경험(생성, 수정 또는 중지) (AWS DMS 설명서의 [AWS DMS 작업](#) 사용 참조)
- 데이터베이스 암호화를 위한 AWS Key Management Service(AWS KMS)에 대한 지식이 있어야 함 ([AWS KMS 설명서](#) 참조)

제한 사항

- Amazon RDS DB 인스턴스에 대한 암호화는 암호화를 생성할 때에만 활성화할 수 있으며 DB 인스턴스가 생성된 후에는 불가능합니다.
- [로그되지 않은 테이블](#)의 데이터는 스냅샷을 사용하여 복원되지 않습니다. 자세한 내용은 [PostgreSQL로 작업하기 위한 모범 사례](#)를 참조하십시오.
- 암호화되지 않은 DB 인스턴스의 암호화된 읽기 전용 복제본이나 암호화된 DB 인스턴스의 암호화되지 않은 읽기 전용 복제본은 보유할 수 없습니다.
- 암호화되지 않은 백업 또는 스냅샷을 암호화된 DB 인스턴스로 복원할 수 없습니다.
- AWS DMS는 시퀀스를 자동으로 전송하지 않으므로 이를 처리하려면 추가 단계가 필요합니다.

자세한 내용은 Amazon RDS 설명서의 [Amazon RDS 암호화된 DB 인스턴스의 제한](#)을 참조하십시오.

아키텍처

소스 아키텍처

- 암호화되지 않은 RDS DB 인스턴스

대상 아키텍처

- 암호화된 RDS DB 인스턴스
 - 대상 RDS DB 인스턴스는 원본 RDS DB 인스턴스의 DB 스냅샷 복사본을 복원하여 생성됩니다.
 - AWS KMS 키는 스냅샷을 복원하는 동안 암호화하는 데 사용됩니다.
 - AWS DMS 복제 작업은 데이터를 마이그레이션하는 데 사용됩니다.

도구

암호화를 활성화하는 데 사용되는 도구:

- 암호화를 위한 AWS KMS 키 - 암호화된 DB 인스턴스를 생성할 때 고객 관리형 키 또는 Amazon RDS용 AWS 관리형 키를 선택하여 DB 인스턴스를 암호화할 수 있습니다. 고객 관리형 키에 대한 키 식별자를 지정하지 않으면 Amazon RDS는 새 DB 인스턴스에 대해 AWS 관리형 키를 사용합니다. Amazon RDS는 AWS 계정에 대한 AWS 관리형 키를 생성합니다. AWS 계정에는 각 AWS 리전마다 Amazon RDS에 대해 서로 다른 AWS 관리형 키가 있습니다. Amazon RDS 암호화에 KMS 키를 사용하는 방법에 대한 자세한 내용은 [Amazon RDS 리소스 암호화](#)를 참조하십시오.

지속적인 복제에 사용되는 도구:

- AWS Database Migration Service(AWS DMS)를 사용하여 원본 DB에서 대상 DB로 변경 내용을 복제할 수 있습니다. 다운타임을 최소화하려면 원본과 대상 DB를 동기화된 상태로 유지하는 것이 중요합니다. AWS DMS 설정 및 작업 생성에 대한 자세한 내용은 [AWS DMS 설명서](#)를 참조하십시오.

에픽

소스 DB 인스턴스의 스냅샷을 만들고 암호화합니다.

작업	설명	필요한 기술
원본 PostgreSQL DB 인스턴스의 세부 정보를 확인합니다.	Amazon RDS 콘솔에서 원본 PostgreSQL DB 인스턴스를 선택합니다. 구성 탭에서 인스턴스에 대해 암호화가 활성화되어 있는지 확인합니다. 화면 그림은 추가 정보 섹션을 참조하십시오.	DBA
DB 스냅샷을 만듭니다.	암호화하려는 인스턴스의 DB 스냅샷을 생성합니다. 스냅샷을 만드는 데 걸리는 시간은 데이터베이스의 크기에 따라 다릅니다. 지침은 Amazon RDS 설명서의 DB 스냅샷 생성 을 참조하십시오.	DBA
스냅샷을 암호화합니다.	Amazon RDS 콘솔 탐색 창에서 [스냅샷] 을 선택하고 생성한	DBA

작업	설명	필요한 기술
	<p>DB 스냅샷을 선택합니다. 작업에서 스냅샷 복사를 선택합니다. 대상 AWS 리전과 DB 스냅샷 사본의 이름을 해당 필드에 입력합니다. 암호화 활성화 확인란을 선택합니다. [Master Key]에 대해 DB 스냅샷 사본을 암호화할 때 사용할 KMS 키 식별자를 지정합니다. [Copy Snapshot]을 선택합니다. 자세한 내용은 Amazon RDS 문서에서 스냅샷 복사를 참조하세요.</p>	

대상 DB 인스턴스 준비

작업	설명	필요한 기술
<p>DB 스냅샷을 복원합니다.</p>	<p>Amazon RDS 콘솔에서 스냅샷을 선택합니다. 생성한 암호화된 스냅샷을 선택합니다. 작업에서 스냅샷 복원을 선택합니다. DB 인스턴스 식별자의 경우 새 DB 인스턴스의 고유 이름을 입력합니다. 인스턴스 세부 정보를 검토한 다음 DB 인스턴스 복원을 선택합니다. 스냅샷에서 암호화된 새 DB 인스턴스가 생성됩니다. 자세한 내용은 Amazon RDS 설명서에서 DB 스냅샷에서 복원을 참조하세요.</p>	<p>DBA</p>
<p>AWS DMS를 사용하여 데이터를 마이그레이션합니다.</p>	<p>AWS DMS 콘솔에서 AWS DMS 작업을 생성합니다. 마이</p>	<p>DBA</p>

작업	설명	필요한 기술
	<p>그레이션 유형에서 기존 데이터 마이그레이션 및 진행 중인 변경 사항 복제를 선택합니다. 작업 설정에서 대상 테이블 준비 모드에 대해 잘라내기를 선택합니다. 자세한 내용은 AWS DMS 문서에서 작업 만들기를 참조하세요.</p>	
<p>데이터 유효성 검사를 사용 설정합니다.</p>	<p>작업 설정에서 검증 활성화를 선택합니다. 이렇게 하면 소스 데이터와 대상 데이터를 비교하여 데이터가 정확하게 마이그레이션되었는지 확인할 수 있습니다.</p>	DBA
<p>대상 DB 인스턴스에서 제약 조건을 비활성화합니다.</p>	<p>대상 DB 인스턴스에서 트리거 및 외래 키 제약 조건을 비활성화한 다음 AWS DMS 작업을 시작합니다. 트리거 비활성화 및 외래 키 제약 조건에 대한 자세한 내용은 AWS DMS 설명서를 참조하십시오.</p>	DBA
<p>데이터 확인</p>	<p>전체 로드가 완료되면 대상 DB 인스턴스의 데이터가 원본 데이터와 일치하는지 확인합니다. 자세한 내용은 AWS DMS 설명서에서 AWS DMS 데이터 유효성 검사를 참조하세요.</p>	DBA

대상 DB 인스턴스로 전환

작업	설명	필요한 기술
소스 DB 인스턴스에서 쓰기 작업을 중지합니다.	애플리케이션 다운타임이 시작될 수 있도록 원본 DB 인스턴스에서 쓰기 작업을 중지하십시오. AWS DMS가 파이프라인의 데이터 복제를 완료했는지 확인하십시오. 대상 DB 인스턴스에서 트리거와 외래 키를 활성화합니다.	DBA
데이터베이스 시퀀스 업데이트	원본 데이터베이스에 시퀀스 번호가 포함되어 있는 경우 대상 데이터베이스의 시퀀스를 확인하고 업데이트하십시오.	DBA
애플리케이션 엔드포인트를 구성합니다.	애플리케이션 연결을 구성하여 Amazon RDS DB 인스턴스 엔드포인트를 사용합니다. 이제 DB 인스턴스가 암호화됩니다.	DBA, 애플리케이션 소유자

관련 리소스

- [AWS DMS 태스크 생성](#)
- [Amazon을 사용한 복제 작업 모니터링 CloudWatch](#)
- [AWS DMS 작업 모니터링](#)
- [Amazon RDS 암호화 키 업데이트](#)

추가 정보

원본 PostgreSQL DB 인스턴스의 암호화를 확인합니다.

이 패턴에 대한 추가 참고 사항:

- `rds.logical_replication` 파라미터를 1로 설정하여 PostgreSQL에서 복제를 활성화합니다.

중요 참고 사항: 복제 슬롯은 파일이 외부에서 사용될 때까지 WAL (Write Ahead Log) 파일을 보관합니다. 예를 들어, `pg_recvlogical` 등 추출, 전환, 적재(ETL) 작업이나 AWS DMS에 의해 사용 됩니다. `rds.logical_replication` 파라미터 값을 1로 설정하면 AWS DMS는 `wal_level`, `max_wal_senders`, `max_replication_slots`, 및 `max_connections` 파라미터를 설정합니다. 논리적 복제 슬롯이 있지만 복제 슬롯에 보관된 WAL 파일을 사용할 소비자가 없는 경우, 트랜잭션 로그 디스크 사용량은 증가하고 사용 가능한 스토리지 공간은 계속 감소할 수 있습니다. 이 문제를 해결 하기 위한 자세한 [내용 및 단계는 Amazon RDS for PostgreSQL에서 “디바이스에 남은 공간 없음” 또는 “DiskFull” 오류의 원인을 식별하려면 어떻게 해야 합니까?](#) 문서를 참조하십시오. AWS Support 지식 센터에서

- DB 스냅샷을 생성한 후 원본 DB 인스턴스에 적용한 스키마 변경 사항은 대상 DB 인스턴스에 적용 되지 않습니다.
- 암호화된 DB 인스턴스를 만든 후에는 해당 DB 인스턴스에서 사용하는 KMS 키를 변경할 수 없습니다. 암호화된 DB 인스턴스를 생성하기 전에 KMS 키 요구 사항을 확인해야 합니다.
- AWS DMS 작업을 실행하기 전에 대상 DB 인스턴스에서 트리거와 외래 키를 비활성화해야 합니다. 작업이 완료되면 이러한 기능을 다시 활성화할 수 있습니다.

시작 시 Amazon RDS 데이터베이스의 자동 태그 지정 적용

작성자: Susanne Kangnoh(AWS)

환경: 프로덕션

기술: 데이터베이스, 클라우드 네이티브, 보안, 자격 증명, 규정 준수

AWS 서비스: 아마존 RDS, 아마존 SNS, AWS CloudTrail, 아마존 CloudWatch

요약

Amazon Relational Database Service(RDS)는 Amazon Web Services(AWS) 클라우드에서 관계형 데이터베이스를 더 쉽게 설치, 운영 및 확장할 수 있는 웹 서비스입니다. 업계 표준 관계형 데이터베이스를 위한 비용 효율적이고 크기 조정이 가능한 용량을 제공하며 일반적인 데이터베이스 관리 작업을 관리합니다.

태그를 사용하여 다양한 방식으로 AWS 리소스를 분류할 수 있습니다. 계정에 리소스가 많을 때 태그에 따라 특정 리소스를 빠르게 식별하려는 경우 관계형 데이터베이스 태깅이 유용합니다. Amazon RDS 태그를 사용하여 RDS DB 인스턴스에 사용자 정의 메타데이터를 추가할 수 있습니다. 태그는 사용자 정의 키와 값으로 구성됩니다. 조직의 요구 사항에 맞는 일관된 태그 집합을 생성하는 것이 좋습니다.

이 패턴은 RDS DB 인스턴스를 모니터링하고 태그를 지정하는 데 도움이 되는 AWS CloudFormation 템플릿을 제공합니다. 템플릿은 AWS CloudTrail CreateDBInstance CloudWatch 이벤트를 감시하는 Amazon Events 이벤트를 생성합니다. (Amazon RDS에 대한 API 호출을 이벤트로 CloudTrail 캡처합니다.) 이 이벤트가 감지되면 사용자가 정의한 태그 키와 값을 자동으로 적용하는 AWS Lambda 함수를 호출합니다. 템플릿은 Amazon Simple Notification Service(SNS)를 사용하여 인스턴스에 태그가 지정되었다는 알림도 보냅니다.

사전 조건 및 제한 사항

사전 조건

- 활성 상태의 AWS 계정.
- Lambda 코드를 업로드하기 위한 Amazon Simple Storage Service(S3) 버킷
- 태깅 알림을 받으려는 이메일 주소입니다.

제한 사항

- 이 솔루션은 CloudTrail CreateDB 인스턴스 이벤트를 지원합니다. 다른 이벤트에 대한 알림은 생성되지 않습니다.

아키텍처

워크플로 아키텍처

자동화 및 규모 조정

- 여러 AWS 지역 및 계정에 대해 AWS CloudFormation 템플릿을 여러 번 사용할 수 있습니다. 템플릿은 각 리전 또는 계정에서 한 번만 실행하면 됩니다.

도구

서비스

- [AWS CloudTrail](#) — CloudTrail AWS는 AWS 계정의 거버넌스, 규정 준수, 운영 및 위험 감사를 지원하는 AWS 서비스입니다. 사용자, 역할 또는 AWS 서비스가 수행한 작업은 이 이벤트로 기록됩니다. CloudTrail.
- [Amazon CloudWatch Events](#) — Amazon CloudWatch Events는 AWS 리소스의 변경 사항을 설명하는 시스템 이벤트의 스트림을 거의 실시간으로 제공합니다. CloudWatch 이벤트는 운영 변경이 발생하는 즉시 이를 인지하고 필요에 따라 수정 조치를 취하여 환경에 대응하고, 기능을 활성화하고, 변경하고, 상태 정보를 캡처합니다.
- [AWS Lambda](#) - AWS Lambda는 서버를 프로비저닝하거나 관리할 필요 없이 코드 실행을 지원하는 컴퓨팅 서비스입니다. Lambda는 필요 시에만 코드를 실행하며, 일일 몇 개의 요청에서 초당 수천 개의 요청까지 자동으로 규모를 조정합니다. 사용한 컴퓨팅 시간만큼만 비용을 지불하고, 코드가 실행되지 않을 때는 요금이 부과되지 않습니다.
- [Amazon S3](#)-Amazon Simple Storage Service(S3)는 웹 사이트, 모바일 애플리케이션, 백업, 데이터 레이크 등 다양한 스토리지 솔루션에 사용할 수 있는 확장성이 뛰어난 객체 스토리지 서비스입니다.
- [Amazon SNS](#) – Amazon Simple Notification Service(SNS)는 애플리케이션, 최종 사용자 및 디바이스가 클라우드에서 즉시 알림을 전송하고 수신할 수 있게 해 주는 웹 서비스입니다.

코드

이 패턴에는 두 개의 파일이 포함된 첨부 파일이 포함됩니다.

- `index.zip`은(는) 이 패턴의 Lambda 코드가 포함된 압축 파일입니다.
- `rds.yaml` Lambda 코드를 배포하는 CloudFormation 템플릿입니다.

이러한 파일을 사용하는 방법에 대한 자세한 내용은 에픽 섹션을 참조하세요.

에픽

Lambda 코드 배포

작업	설명	필요한 기술
코드를 S3 버킷에 업로드합니다.	새 S3 버킷을 생성하거나 기존 S3 버킷을 사용하여 첨부 <code>index.zip</code> 파일(Lambda 코드)을 업로드합니다. 이 버킷은 모니터링하려는 리소스(RDS DB 인스턴스)와 동일한 AWS 리전에 있어야 합니다.	클라우드 아키텍트
템플릿을 배포합니다. CloudFormation	S3 버킷과 동일한 AWS 리전에서 CloudFormation 콘솔을 열고 첨부 파일에 제공된 <code>rds.yaml</code> 파일을 배포합니다. 다음 에픽에서 템플릿 파라미터에 대한 값을 입력합니다.	클라우드 아키텍트

CloudFormation 템플릿의 파라미터를 완성하세요.

작업	설명	필요한 기술
S3 버킷 이름을 제공합니다.	첫 번째 에픽에서 생성하거나 선택한 S3 버킷의 이름을 입력합니다. 이 S3 버킷에는 Lambda 코드용 <code>zip</code> 파일이 들어 있으며 모니터링하려는 템	클라우드 아키텍트

작업	설명	필요한 기술
	플릿 및 RDS DB 인스턴스와 동일한 CloudFormation AWS 리전에 있어야 합니다.	
S3 키를 입력합니다.	S3 버킷에 있는 Lambda 코드 .zip 파일의 위치를 앞에 슬래시 없이 입력합니다(예: index.zip 또는 controls/index.zip).	클라우드 아키텍트
이메일 주소를 입력합니다.	위반 알림을 받을 사용 중인 이메일 주소를 입력합니다.	클라우드 아키텍트
로깅 수준을 지정합니다.	로깅 수준 및 상세 정보를 지정합니다. Info는 애플리케이션 진행 상황에 대한 자세한 정보 메시지를 지정하며 디버깅용으로만 사용해야 합니다. Error는 애플리케이션을 계속 실행할 수 있는 오류 이벤트를 지정합니다. Warning은 잠재적으로 유해한 상황을 지정합니다.	클라우드 아키텍트
RDS DB 인스턴스의 태그 키와 값을 입력합니다.	RDS 인스턴스에 자동으로 적용할 필수 태그 키 및 값을 입력합니다. 자세한 내용은 AWS 설명서의 Amazon RDS 리소스에 태그 지정 을 참조하세요.	클라우드 아키텍트

구독 확인

작업	설명	필요한 기술
이메일 구독을 확인합니다.	CloudFormation 템플릿이 성공적으로 배포되면 입력한 이메일 주소로 구독 이메일 메시지가 전송됩니다. 인스턴스에 태그가 지정되었을 때 알림을 받으려면 이 이메일 구독을 확인해야 합니다.	클라우드 아키텍트

관련 리소스

- [버킷 생성](#)(Amazon S3 설명서)
- [Amazon RDS 리소스에 태그 지정](#)(Amazon Aurora 설명서)
- [객체 업로드](#)(Amazon S3 설명서)
- [AWS를 사용하여 AWS API 호출에서 트리거되는 CloudWatch 이벤트 규칙 생성 CloudTrail](#) (아마존 CloudWatch 설명서)

첨부

이 문서와 관련된 추가 콘텐츠에 액세스하려면 [attachment.zip](#) 파일의 압축을 풉니다.

온디맨드 용량에 대한 DynamoDB 테이블의 비용 추정

작성자: Moinul Al-Mamun

환경: 프로덕션

기술: 데이터베이스, 클라우드
네이티브, 서버리스, 비용 관리

AWS 서비스: Amazon
DynamoDB

요약

[Amazon DynamoDB](#)는 페타바이트 규모에서도 10밀리 초 미만의 지연 시간을 지원하는 NoSQL 트랜잭션 데이터베이스입니다. 이 Amazon Web Services(AWS) 서버리스 제품은 일관된 성능과 확장성으로 인기를 얻고 있습니다. 기본 인프라의 프로비저닝이 필요하지 않습니다. 단일 테이블은 페타바이트까지 확장 가능합니다.

온디맨드 용량 모드를 이용하면 애플리케이션이 테이블에서 수행하는 데이터 읽기 및 쓰기에 대해 요청당 요금을 지불합니다. AWS 요금은 한 달 동안 누적된 읽기 요청 단위(RRU) 및 쓰기 요청 단위(WRU)를 기준으로 합니다. DynamoDB는 한 달 내내 지속적으로 테이블 크기를 모니터링하여 스토리지 요금을 결정합니다. point-in-time-recovery (PITR) 를 통한 연속 백업을 지원합니다. DynamoDB는 한 달 내내 지속적으로 PITR 지원 테이블 크기를 모니터링하여 백업 요금을 결정합니다.

프로젝트의 DynamoDB 비용을 추정할 때 제품 수명 주기의 여러 단계에서 소비되는 RRU, WRU 및 스토리지의 양을 계산하는 것이 중요합니다. 대략적인 비용 산정을 위해 [AWS 요금 계산기](#)를 사용할 수 있지만 테이블에 필요한 RRU, WRU 및 스토리지 요구 사항의 대략적인 수를 제공해야 합니다. 프로젝트 초기에는 이러한 수치를 예측하기 어려울 수 있습니다. AWS 요금 계산기는 데이터 증가율이나 항목 크기를 고려하지 않으며, 기본 테이블과 글로벌 보조 인덱스(GSI)의 읽기 및 쓰기 수를 별도로 고려하지 않습니다. AWS 요금 계산기를 사용하려면 WRU, RRU 및 스토리지 크기에 대한 모든 측면을 추정하여 예상 비용을 산출해야 합니다.

이 패턴은 온디맨드 용량 모드에서 쓰기, 읽기, 스토리지, 백업 및 복구 비용과 같은 기본 DynamoDB 비용 요소를 추정하는 메커니즘과 재사용 가능한 Microsoft Excel 템플릿을 제공합니다. AWS 요금 계산기보다 더 세분화되어 있으며 기본 테이블과 GSI 요구 사항을 독립적으로 고려합니다. 또한 월별 품목 데이터 증가율을 고려하여 3년간의 비용을 예측합니다.

사전 조건 및 제한 사항

사전 조건

- DynamoDB 및 DynamoDB 데이터 모델 설계에 대한 기본 지식

- DynamoDB 요금, WRU, RRU, 스토리지, 백업 및 복구에 대한 기본 지식(자세한 내용은 [온디맨드 용량 요금](#) 참조)
- DynamoDB의 데이터, 데이터 모델, 항목 크기에 대한 지식
- DynamoDB GSI에 대한 지식

제한 사항

- 템플릿은 대략적인 계산을 제공하지만 모든 구성에 적합하지는 않습니다. 더 정확한 추정치를 얻으려면 기본 표와 GSI에 있는 각 항목의 개별 크기를 측정해야 합니다.
- 더 정확한 추정을 위해서는 월별 각 항목에 대한 예상 쓰기(삽입, 업데이트, 삭제) 및 읽기 횟수를 고려해야 합니다.
- 이 패턴은 고정된 데이터 증가 가정을 기반으로 향후 몇 년간 쓰기, 읽기, 스토리지, 백업 및 복구 비용만 추정하는 것을 지원합니다.

도구

서비스

- [Amazon DynamoDB](#)는 빠르고 예측 가능하고 확장 가능한 성능을 제공하는 완전 관리형 NoSQL 데이터베이스 서비스입니다.

기타 도구

- [AWS 요금 계산기](#)는 AWS 사용 사례에 대한 추정치를 생성하는 데 사용할 수 있는 웹 기반 계획 도구입니다.

모범 사례

비용을 낮게 유지하려면 다음과 같은 DynamoDB 설계 모범 사례를 고려해 보세요.

- [파티션 키 설계](#)-카디널리티가 높은 파티션 키를 사용하여 로드를 균등하게 분산합니다.
- [인접 목록 디자인 패턴](#) - 이 디자인 패턴을 관리 one-to-many 및 many-to-many 관계에 사용합니다.
- [스파스 인덱스](#) - GSI에 스파스 인덱스를 사용합니다. GSI를 생성할 때는 파티션 키와 정렬 키(선택 사항)를 지정해야 합니다. 해당 GSI 파티션 키를 포함하는 기본 테이블의 항목만 스파스 인덱스에 나타납니다. 이는 GSI를 더 작게 유지하는 데 도움이 됩니다.

- [인덱스 오버로딩](#) - 다양한 유형의 항목을 인덱싱하는 데 동일한 GSI를 사용합니다.
- [GSI 쓰기 샤드](#) - 효율적이고 빠른 쿼리를 위해 파티션 전체에 데이터를 현명하게 샤드합니다.
- [대용량 항목](#) - 테이블 내에 메타데이터만 저장하고, Amazon S3에 blob을 저장하고, DynamoDB에 참조를 유지합니다. 대용량 항목을 여러 항목으로 나누고 정렬 키를 사용하여 효율적으로 인덱싱합니다.

추가 설계 모범 사례는 Amazon DynamoDB [개발자 안내서](#)를 참조하세요.

에픽

DynamoDB 데이터 모델에서 항목 정보 추출

작업	설명	필요한 기술
항목의 크기를 확인합니다.	<ol style="list-style-type: none"> 1. 테이블에 저장할 항목 유형의 수를 확인합니다. 2. 각 항목의 크기를 킬로바이트 단위로 계산하려면 각 속성의 키와 값 크기를 더합니다. 3. 기본 테이블과 각 GSI의 항목 크기를 계산합니다. 	데이터 엔지니어
쓰기 비용을 추정합니다.	<p>온디맨드 용량 모드에서 쓰기 비용을 추정하려면 먼저 한 달에 소비될 WRU 수를 측정해야 합니다. 이를 위해서는 다음 요소를 고려해야 합니다.</p> <ul style="list-style-type: none"> • 한 달 간 각 항목에 대한 생성, 업데이트 및 삭제 작업 수. • 사용 가능한 GSI 수. 각 지수를 개별적으로 고려합니다. <ul style="list-style-type: none"> • 인덱스 항목의 평균 크기 • 인덱스의 동기화 횟수 	데이터 엔지니어

작업	설명	필요한 기술
	<ul style="list-style-type: none"> 매달 테이블에 새로 추가되는 항목(예: 구성 요소 또는 제품)은 몇 개입니까? 추가되는 항목의 수는 매달 다를 수 있지만 비즈니스 사례를 기반으로 평균 증가율을 가정할 수 있습니다. <p>자세한 내용은 추가 정보 섹션을 참조하세요.</p>	
읽기 비용을 추정합니다.	<p>온디맨드 모드에서 읽기 비용을 추정하려면 먼저 한 달에 소비될 RRU 수를 측정해야 합니다. 이를 위해서는 다음 요소를 고려해야 합니다.</p> <ul style="list-style-type: none"> 사용 가능한 GSI 수. 각 지수를 개별적으로 고려합니다. <ul style="list-style-type: none"> 인덱스 항목의 평균 크기 제품당 월별 평균 읽기 횟수. DynamoDB 테이블에서 사용 가능한 총 항목(구성 요소 또는 제품)의 수입니다. 	데이터 엔지니어, 앱 개발자

작업	설명	필요한 기술
<p>스토리지 크기 및 비용을 추정합니다.</p>	<p>먼저 표의 항목 크기를 기준으로 월 평균 스토리지 요구량을 추정합니다. 그런 다음 스토리지 크기에 AWS 리전의 GB당 스토리지 가격을 곱하여 스토리지 비용을 계산합니다.</p> <p>쓰기 비용 추정을 위한 데이터를 이미 입력한 경우 스토리지 크기를 계산할 때 다시 입력할 필요가 없습니다. 그렇지 않으면 스토리지 크기를 추정하기 위해 다음 요소를 고려해야 합니다.</p> <ul style="list-style-type: none"> • 테이블 디자인에 따른 모듈 (제품)의 데이터 항목 수. • 평균 항목 크기(킬로바이트). • 사용 가능한 GSI 수. 각 지수를 개별적으로 고려합니다. <ul style="list-style-type: none"> • 인덱스 항목의 평균 크기 • 매달 테이블에 새로 추가되는 제품은 몇 개입니까? 추가되는 제품의 수는 매달 다를 수 있지만 비즈니스 사례를 기반으로 평균 증가율을 가정할 수 있습니다. 이 예시에서는 매월 평균 1천만 개 새 제품을 사용합니다. 	<p>데이터 엔지니어</p>

Excel 템플릿에 항목 및 객체 정보 입력

작업	설명	필요한 기술
<p>첨부 파일 섹션에서 Excel 템플릿을 다운로드하여 사용 사례 표에 맞게 조정합니다.</p>	<ol style="list-style-type: none"> Excel 템플릿을 다운로드합니다. 테이블 디자인에 따라 비즈니스 모듈과 GSI를 조정합니다. 	<p>데이터 엔지니어</p>
<p>Excel 템플릿에 정보를 입력합니다.</p>	<ol style="list-style-type: none"> 시트에서 항목 정보를 업데이트합니다. 주황색 셀의 데이터만 업데이트합니다. 객체 수 조정: 매월 테이블에 추가할 수 있는 개수는 얼마인가요? AWS 리전의 백만 개당 WRU 및 RRU 가격을 업데이트합니다. AWS 리전의 월별 GB당 스토리지 및 백업 가격을 업데이트합니다. AWS 리전의 GB당 복구 가격을 업데이트합니다. <p>템플릿에는 정보, 메타데이터, 관계라는 세 가지 항목 또는 엔티티가 있습니다. 두 개의 GSI가 있습니다. 사용 사례에 따라 더 많은 항목이 필요한 경우 새 행을 생성합니다. GSI가 더 필요한 경우 기존 GSI 블록을 복사한 다음 붙여넣어 필요한 만큼 GSI 블록을 생성합니다. 그</p>	<p>데이터 엔지니어</p>

작업	설명	필요한 기술
	런 다음 합계 및 총계 열 계산을 조정합니다.	

관련 리소스

참조

- [온디맨드 용량에 대한 Amazon DynamoDB 요금](#)
- [DynamoDB용 AWS 요금 계산기](#)
- [DynamoDB를 사용한 설계 및 아키텍처 설계 모범 사례](#)
- [DynamoDB 시작하기](#)

가이드 및 패턴

- [Amazon DynamoDB를 사용한 데이터 모델링](#)
- [Amazon DynamoDB 테이블의 스토리지 비용 추정](#)

추가 정보

작성 비용 계산 예제

DynamoDB 데이터 모델 설계에서는 한 제품에 대해 세 개의 항목이 표시되며 평균 항목 크기는 4KB입니다. DynamoDB 기본 테이블에 새 제품을 추가하면 항목 수 * (항목 크기/1KB 쓰기 단위) = 3 * (4/1) = 12WRU를 소비합니다. 이 예제에서 1KB를 쓰는 경우 제품은 1WRU를 소비합니다.

읽기 비용 계산 예제

RRU 추정치를 구하려면 매달 각 항목을 읽는 횟수의 평균을 고려합니다. 예를 들어 정보 항목은 한 달에 평균 10회, 메타데이터 항목은 2회, 관계 항목은 5회 읽습니다. 예제 템플릿에서 모든 구성 요소의 총 RRU는 매월 생성되는 새 구성 요소 수 * 월별 구성 요소당 RRU = 1천만 * 17 RRU = 매월 1억 7천만 RRU입니다.

매달 새로운 항목(구성 요소 또는 제품)이 추가되며 총 제품 수는 시간이 지남에 따라 늘어납니다. 따라서 RRU 요구 사항도 시간이 지남에 따라 증가합니다.

- 첫 달 RRU 소비량은 1억 7천만입니다.
- 두 번째 달의 RRU 소비량은 $2 * 1억 7천만 = 3억 4천만$ 이 됩니다.
- 세 번째 달의 RRU 소비량은 $3 * 1억 7천만 = 5억 1천만$ 이 됩니다.

다음 그래프는 월별 RRU 소비량과 비용 예측을 보여줍니다.

그래프 내의 가격은 설명을 돕기 위한 것입니다. 사용 사례에 대한 정확한 예측을 생성하려면 AWS 요금 페이지를 확인하고 Excel 시트에서 해당 가격을 사용합니다.

스토리지, 백업 및 복구 비용 계산 예제

DynamoDB 스토리지, 백업, 복원은 모두 서로 연결되어 있습니다. 백업은 스토리지와 직접 연결되고 복구는 백업 크기와 직접 연결됩니다. 테이블 크기가 커지면 그에 따라 해당 스토리지, 백업 및 복구 비용도 증가합니다.

스토리지 크기 및 비용

스토리지 비용은 데이터 증가율에 따라 시간이 지나면서 증가합니다. 예를 들어 기본 테이블과 GSI에 있는 구성 요소 또는 제품의 평균 크기가 11KB이고 데이터베이스 테이블에 매달 1천만 개의 새 제품이 추가된다고 가정해 보겠습니다. 이 경우 DynamoDB 테이블 크기가 매월 $(11KB * 1천만) / 1024 / 1024 = 105GB$ 씩 늘어납니다. 첫 번째 달에는 테이블 스토리지 크기가 105GB이고, 두 번째 달에는 $105 + 105 = 210GB$ 가 되는 식입니다.

- 첫 달의 스토리지 비용은 AWS 리전의 GB당 105GB * 스토리지 요금이 부과됩니다.
- 두 번째 달의 스토리지 비용은 해당 리전의 GB당 210GB * 스토리지 요금이 부과됩니다.
- 세 번째 달의 스토리지 비용은 해당 리전의 GB당 315GB * 스토리지 요금이 부과됩니다.

향후 3년간의 스토리지 크기 및 비용은 스토리지 크기 및 예측 섹션을 참조하세요.

백업 비용

백업 비용은 데이터 증가율에 따라 시간이 지나면서 증가합니다. point-in-time-recovery (PITR) 를 사용하여 연속 백업을 켜면 연속 백업 요금은 월별 평균 스토리지 GB를 기준으로 부과됩니다. 월별 평균 백업 크기는 테이블 스토리지 크기와 같지만 실제 크기는 약간 다를 수 있습니다. 매달 새 제품이 추가되므로 시간이 지남에 따라 전체 스토리지 크기와 백업 크기도 커집니다. 예를 들어 첫 달의 평균 백업 크기가 105GB에서 두 번째 달에 210GB로 증가할 수 있습니다.

- 첫 달의 백업 비용은 AWS 리전의 GB당 월 105GB * 연속 백업 요금입니다.
- 두 번째 달의 백업 비용은 해당 리전의 GB당 월 210GB * 연속 백업 요금입니다.
- 세 번째 달의 백업 비용은 해당 리전의 GB당 월 315GB * 연속 백업 요금입니다.
- 이런 식으로 계속 진행됩니다.

백업 비용은 스토리지 크기 및 비용 예측 섹션의 그래프에 포함됩니다.

복구 비용

PITR을 활성화한 상태에서 연속 백업을 수행하는 경우 복구 작업 요금은 복원 크기에 따라 달라집니다. 복원할 때마다 기가바이트의 복원된 데이터를 기준으로 비용을 지불합니다. 테이블 크기가 크고 한 달에 여러 번 복원을 수행하면 비용이 많이 듭니다.

복원 비용을 추정하기 위해 이 예에서는 매월 말에 PITR 복구를 1회 수행한다고 가정합니다. 이 예에서는 월별 평균 백업 크기를 해당 월의 복원 데이터 크기로 사용합니다. 첫 달의 평균 백업 크기는 105GB 이고, 월말 복구의 경우 복원 데이터 크기는 105GB입니다. 두 번째 달에는 210GB가 되는 식입니다.

복구 비용은 데이터 증가율에 따라 시간이 지나면서 증가합니다.

- 첫 달의 복구 비용은 AWS 리전의 GB당 105GB * 복원 요금이 부과됩니다.
- 두 번째 달의 복구 비용은 해당 리전의 GB당 210GB * 복원 요금이 부과됩니다.
- 세 번째 달의 복구 비용은 해당 리전의 GB당 315GB * 복원 요금이 부과됩니다.

자세한 내용은 Excel 템플릿의 스토리지, 백업 및 복구 탭과 다음 섹션의 그래프를 참조하세요.

스토리지 크기 및 비용 예측

템플릿에서 실제 청구 가능한 스토리지 크기는 표준 테이블 클래스의 월 25GB 프리 티어를 빼서 계산합니다. 이 시트에는 월별 값으로 구분된 예측 그래프가 포함되어 있습니다.

다음 예제 차트는 향후 36개월의 월별 스토리지 크기(GB), 청구 가능한 스토리지 비용, 온디맨드 백업 비용, 복구 비용을 예측합니다. 모든 비용은 USD로 표시됩니다. 그래프를 보면 스토리지, 백업 및 복구 비용이 스토리지 크기 증가에 비례하여 증가한다는 것을 알 수 있습니다.

그래프에 사용된 가격은 설명을 돕기 위한 것입니다. 사용 사례에 대한 정확한 가격을 생성하려면 AWS 요금 페이지를 확인하고 Excel 템플릿에서 해당 가격을 사용합니다.

첨부

이 문서와 관련된 추가 콘텐츠에 액세스하려면 [attachment.zip](#) 파일의 압축을 풉니다.

Amazon DynamoDB 테이블의 스토리지 비용 추정

작성자: Moinul Al-Mamun

환경: PoC 또는 파일럿

기술: 데이터베이스, 빅 데이터,
비용 관리, 스토리지 및 백업

AWS 서비스: Amazon
DynamoDB

요약

[Amazon DynamoDB](#)는 페타바이트 규모에서도 10밀리 초 미만의 지연 시간을 지원하는 NoSQL 트랜잭션 데이터베이스입니다. 이 Amazon Web Services(AWS) 서버리스 제품은 일관된 성능과 확장성으로 인기를 얻고 있습니다. 스토리지 프로비저닝이 필요하지 않습니다. 단일 테이블은 페타바이트까지 확장 가능합니다.

DynamoDB는 한 달 내내 지속적으로 테이블 크기를 모니터링하여 스토리지 요금을 결정합니다. 그러면 AWS가 기가바이트 스토리지의 평균 크기에 대해 청구합니다. 시간이 지남에 따라 테이블 크기가 커질수록 스토리지 비용도 더 커집니다. 스토리지 비용을 계산하려면 [AWS 요금 계산기](#)를 사용할 수 있지만 글로벌 보조 인덱스(GSI)를 포함하여 테이블의 대략적인 크기를 제공해야 하는데, 이는 프로젝트 초기에 예측하기가 매우 어렵습니다. 또한 AWS 요금 계산기는 데이터 증가율을 고려하지 않습니다.

이 패턴은 DynamoDB 스토리지 크기 및 비용을 계산하는 메커니즘과 재사용 가능한 Microsoft Excel 템플릿을 제공합니다. 기본 테이블과 GSI의 스토리지 요구 사항을 개별적으로 고려합니다. 개별 항목의 크기와 시간에 따른 데이터 증가율을 고려하여 스토리지 크기를 계산합니다.

추정치를 구하려면 템플릿에 다음과 같은 두 가지 정보를 삽입합니다.

- 기본 테이블 및 GSI의 개별 항목 킬로바이트 크기
- 한 달에 평균적으로 테이블에 추가할 수 있는 새 객체 또는 제품 수(예를 들어 1,000만 개)

템플릿은 다음 예와 같이 향후 3년 간의 스토리지 및 비용 예측 그래프를 생성합니다.

사전 조건 및 제한 사항

사전 조건

- DynamoDB, DynamoDB 스토리지 및 요금에 대한 기본 지식
- DynamoDB의 데이터, 데이터 모델, 항목 크기에 대한 지식
- DynamoDB 글로벌 보조 인덱스(GSI)에 대한 지식

제한 사항

- 템플릿은 대략적인 계산을 제공하지만 모든 구성에 적합하지는 않습니다. 더 정확한 추정치를 얻으려면 기본 표와 GSI에 있는 각 항목의 개별 크기를 측정해야 합니다.
- 이 패턴은 고정된 데이터 증가 가정을 기반으로 향후 몇 년간 스토리지 크기와 비용만 추정하는 것을 지원합니다.

도구

서비스

- [Amazon DynamoDB](#)는 빠르고 예측 가능하고 확장 가능한 성능을 제공하는 완전 관리형 NoSQL 데이터베이스 서비스입니다.

기타 도구

- [AWS 요금 계산기](#)는 AWS 사용 사례에 대한 추정치를 생성하는 데 사용할 수 있는 웹 기반 계획 도구입니다.

에픽

DynamoDB 데이터 모델에서 항목 정보 추출

작업	설명	필요한 기술
항목의 크기를 확인합니다.	<ol style="list-style-type: none"> 1. 테이블에 저장할 항목 유형의 수를 확인합니다. 2. 각 항목의 크기를 킬로바이트 단위로 계산하려면 각 속성의 키와 값 크기를 더합니다. 	데이터 엔지니어

작업	설명	필요한 기술
	3. 기본 테이블과 각 GSI의 항목 크기를 계산합니다.	
한 달 동안 추가된 객체 수를 구합니다.	한 달 동안 DynamoDB 테이블에 추가될 구성 요소 또는 객체 수를 평균적으로 추정합니다.	데이터 엔지니어

Excel 템플릿에 항목 및 객체 정보 입력

작업	설명	필요한 기술
첨부된 문서에서 Excel 시트를 다운로드하여 사용 사례 표에 맞게 조정합니다.	<ol style="list-style-type: none"> Excel 템플릿을 다운로드합니다. 테이블 디자인에 따라 비즈니스 모듈과 GSI를 조정합니다. 	데이터 엔지니어
Excel 템플릿에 정보를 입력합니다.	<ol style="list-style-type: none"> 시트에서 항목 정보를 업데이트합니다. 객체 수 조정: 매월 테이블에 추가할 수 있는 개수는 얼마인가요? AWS 리전의 월별 GB당 스토리지 요금을 업데이트합니다. 	데이터 엔지니어

관련 리소스

- [Amazon DynamoDB 온디맨드 요금](#)
- [DynamoDB용 AWS 요금 계산기](#)

추가 정보

첨부된 템플릿은 표준 스토리지 테이블 클래스의 스토리지 크기 및 비용만 예측한다는 점에 유의하세요. 저장 비용 예측을 기반으로 개별 항목의 크기와 제품 또는 객체의 증가율을 고려하면 다음을 추정할 수 있습니다.

- 데이터 내보내기 비용
- 백업 및 복구 비용
- 데이터 스토리지 요구 사항

Amazon DynamoDB 데이터 스토리지 비용

DynamoDB는 지속적으로 테이블 크기를 모니터링하여 스토리지 요금을 결정합니다. DynamoDB는 데이터의 원시 바이트 크기와 활성화한 기능에 따라 항목당 스토리지 오버헤드를 추가하여 청구 가능한 데이터의 크기를 측정합니다. 자세한 내용은 [DynamoDB 개발자 안내서](#)를 참조하세요.

데이터 스토리지 요금은 테이블 클래스에 따라 다릅니다. DynamoDB 표준 테이블 클래스를 사용하는 경우 매월 처음 저장되는 25GB는 무료입니다. 여러 AWS 리전의 표준 테이블 클래스 및 Standard-Infrequent Access 테이블 클래스의 스토리지 비용에 대한 자세한 내용은 [온디맨드 용량 요금](#)을 참조하세요.

첨부

이 문서와 관련된 추가 콘텐츠에 액세스하려면 [attachment.zip](#) 파일의 압축을 풉니다.

AWR 보고서를 사용하여 오라클 데이터베이스의 Amazon RDS 엔진 크기를 추정합니다.

작성자: Abhishek Verma(AWS), Eduardo Valentim(AWS)

환경: 프로덕션	소스: Oracle 데이터베이스	대상: Amazon RDS 또는 Amazon Aurora
R 유형: 리아키텍트	워크로드: Oracle	기술: 데이터베이스, 마이그레이션
AWS 서비스: Amazon RDS, Amazon Aurora		

요약

Oracle 데이터베이스를 Amazon RDS (Amazon RDS) 또는 Amazon Aurora로 마이그레이션할 때는 대상 데이터베이스의 CPU, 메모리 및 디스크 I/O를 계산하는 것이 핵심 요구 사항입니다. Oracle AWR (자동 워크로드 리포지토리) 보고서를 분석하여 대상 데이터베이스의 필요한 용량을 추정할 수 있습니다. 이 패턴은 AWR 보고서를 사용하여 이러한 값을 추정하는 방법을 설명합니다.

원본 오라클 데이터베이스는 온프레미스이거나 Amazon Elastic Compute Cloud (Amazon EC2) 인스턴스에 호스팅될 수 있으며, 또는 오라클용 Amazon RDS DB 인스턴스일 수도 있습니다. 대상 데이터베이스는 모든 Amazon RDS 또는 Aurora 데이터베이스가 될 수 있습니다.

참고: 대상 데이터베이스 엔진이 Oracle인 경우 예상 용량이 더 정확합니다. 다른 Amazon RDS 데이터베이스의 경우 데이터베이스 아키텍처의 차이로 인해 엔진 크기가 달라질 수 있습니다.

Oracle 데이터베이스를 마이그레이션하기 전에 성능 테스트를 실행하는 것이 좋습니다.

사전 조건 및 제한 사항

사전 조건

- AWR 보고서를 다운로드하려면 오라클 데이터베이스 엔터프라이즈 에디션 라이선스와 오라클 진단 팩 라이선스가 필요합니다.

제품 버전

- 버전 11g (버전 11.2.0.3.v1 이상) 및 최대 12.2 및 18c, 19c에 대한 모든 오라클 데이터베이스 에디션
- 이 패턴은 오라클 엔지니어드 시스템 또는 오라클 클라우드 인프라 (OCI) 에는 적용되지 않습니다.

아키텍처

소스 기술 스택

다음 중 하나입니다.

- 온프레미스 Oracle 데이터베이스
- EC2 인스턴스의 Oracle 데이터베이스
- Amazon RDS for Oracle DB 인스턴스

대상 기술 스택

- 모든 아마존 RDS 또는 아마존 Aurora 데이터베이스

대상 아키텍처

전체 마이그레이션 프로세스에 대한 자세한 내용은 AWS DMS 및 AWS SCT를 [사용하여 Oracle 데이터베이스를 Aurora PostgreSQL로 마이그레이션하는](#) 패턴을 참조하십시오.

자동화 및 규모 조정

마이그레이션할 Oracle 데이터베이스가 여러 개 있고 추가 성능 지표를 사용하려는 경우, [Oracle 성능 지표를 기반으로 규모에 맞게 Amazon RDS 인스턴스 크기를 적절하게](#) 조정하는 블로그 게시물에 설명된 단계에 따라 프로세스를 자동화할 수 있습니다.

도구

- [Oracle 자동 워크로드 리포지토리 \(AWR\)](#) 는 Oracle 데이터베이스에 구축되는 리포지토리입니다. 시스템 활동 및 워크로드 데이터를 주기적으로 수집 및 저장한 다음 ADDM (자동 데이터베이스 진단 모니터) 에서 이를 분석합니다. AWR은 시스템 성능 데이터의 스냅샷을 주기적으로 (기본적으로 60 분마다) 만들고 정보를 저장합니다 (기본적으로 최대 8일). AWR 보기와 보고서를 사용하여 이 데이터를 분석할 수 있습니다.

모범 사례

- 대상 데이터베이스의 리소스 요구 사항을 계산하려면 단일 AWR 보고서, 다중 AWR 보고서 또는 동적 AWR 보기를 사용할 수 있습니다. 최대 부하 기간에 여러 AWR 보고서를 사용하여 이러한 최대 부하를 처리하는 데 필요한 리소스를 추정하는 것이 좋습니다. 또한 동적 보기는 리소스 요구 사항을 더 정확하게 계산하는 데 도움이 되는 더 많은 데이터 포인트를 제공합니다.
- 마이그레이션하려는 데이터베이스에 대해서만 IOPS를 추정해야 하며 디스크를 사용하는 다른 데이터베이스 및 프로세스에 대해서는 IOPS를 추정하지 않아야 합니다.
- 데이터베이스에서 사용 중인 I/O의 양을 계산하려면 AWR 보고서의 Load Profile 섹션에 있는 정보를 사용하지 마십시오. 사용 가능한 경우 I/O 프로필 섹션을 대신 사용하거나, 인스턴스 활동 통계 섹션으로 건너뛰고 물리적 읽기 및 쓰기 작업의 총 값을 살펴보세요.
- CPU 사용률을 추정할 때는 운영 체제 (OS) 통계 대신 데이터베이스 측정치 방법을 사용하는 것이 좋습니다. 데이터베이스에서만 사용하는 CPU를 기반으로 하기 때문입니다. (OS 통계에는 다른 프로세스의 CPU 사용량도 포함됩니다.) 또한 마이그레이션 후 성능을 개선하려면 ADDM 보고서에서 CPU 관련 권장 사항을 확인해야 합니다.
- 적절한 인스턴스 유형을 결정할 때는 특정 인스턴스 크기에 대한 I/O 처리량 제한, 즉 Amazon Elastic Block Store (Amazon Elastic Block Store) 처리량 및 네트워크 처리량을 고려하십시오.
- 마이그레이션하기 전에 성능 테스트를 실행하여 엔진 크기를 확인하십시오.

에픽

AWR 보고서 생성

작업	설명	필요한 기술
AWR 보고서를 활성화합니다.	보고서를 활성화하려면 Oracle 설명서의 지침을 따르십시오.	DBA
보존 기간을 확인하세요.	AWR 보고서의 보존 기간을 확인하려면 다음 쿼리를 사용하십시오. <pre>SQL> SELECT snap_interval,retention FROM dba_hist_wr_control;</pre>	DBA

작업	설명	필요한 기술
스냅샷을 생성하십시오.	<p>AWR 스냅샷 간격이 최대 워크로드의 스파이크를 캡처할 만큼 세분화되지 않은 경우 AWR 보고서를 수동으로 생성할 수 있습니다. 수동 AWR 스냅샷을 생성하려면 다음 쿼리를 사용하십시오.</p> <pre>SQL> EXEC dbms_workload_repository.create_snapshot;</pre>	DBA
최근 스냅샷을 확인하세요.	<p>최근 AWR 스냅샷을 확인하려면 다음 쿼리를 사용하십시오.</p> <pre>SQL> SELECT snap_id, to_char(begin_interval_time, 'dd/MON/yy hh24:mi') Begin_Interval, to_char(end_interval_time, 'dd/MON/yy hh24:mi') End_Interval FROM dba_hist_snapshot ORDER BY 1;</pre>	DBA

디스크 I/O 요구 사항 추정

작업	설명	필요한 기술
방법을 선택합니다.	<p>IOPS는 스토리지 디바이스의 초당 입력 및 출력 작업에 대한 표준 측정값이며, 읽기 및 쓰기 작업을 모두 포함합니다.</p>	DBA

작업	설명	필요한 기술
	<p>온프레미스 데이터베이스를 AWS로 마이그레이션하는 경우 데이터베이스에서 사용하는 최대 디스크 I/O를 결정해야 합니다. 다음 방법을 사용하여 대상 데이터베이스의 디스크 I/O를 추정할 수 있습니다.</p> <ul style="list-style-type: none"> • AWR 보고서의 로드 프로파일 섹션 • AWR 보고서의 인스턴스 활동 통계 섹션 (Oracle Database 12c 이상의 경우 이 섹션 사용) • AWR 보고서의 I/O 프로파일 섹션 (12c 이전의 Oracle 데이터베이스 버전인 경우 이 섹션 사용) • AWR 보기 <p>다음 단계는 이 네 가지 방법을 설명합니다.</p>	

작업	설명	필요한 기술																				
<p>옵션 1: 로드 프로파일 사용.</p>	<p>다음 표는 AWR 보고서의 로드 프로파일 섹션의 예를 보여줍니다.</p> <p>중요: 더 정확한 정보를 얻으려면 로드 프로파일 대신 옵션 2 (I/O 프로파일) 또는 옵션 3 (인스턴스 작업 통계) 을 사용하는 것이 좋습니다.</p> <table border="1" data-bbox="591 684 1024 1795"> <thead> <tr> <th></th> <th>초당</th> <th>거래당</th> <th>경영진당</th> <th>통화당</th> </tr> </thead> <tbody> <tr> <td>DB 시간 (초):</td> <td>26.6</td> <td>0.2</td> <td>0.00</td> <td>0.02</td> </tr> <tr> <td>데이터베이스 CPU</td> <td>18.0</td> <td>0.1</td> <td>0.00</td> <td>0.01</td> </tr> <tr> <td>백그라운드 CPU</td> <td>0.2</td> <td>0.0</td> <td>0.00</td> <td>0.00</td> </tr> </tbody> </table>		초당	거래당	경영진당	통화당	DB 시간 (초):	26.6	0.2	0.00	0.02	데이터베이스 CPU	18.0	0.1	0.00	0.01	백그라운드 CPU	0.2	0.0	0.00	0.00	<p>DBA</p>
	초당	거래당	경영진당	통화당																		
DB 시간 (초):	26.6	0.2	0.00	0.02																		
데이터베이스 CPU	18.0	0.1	0.00	0.01																		
백그라운드 CPU	0.2	0.0	0.00	0.00																		

작업	설명	필요한 기술
	<p>다시 실행 크기 (바이트): 2,451 17,099.9</p> <p>논리적 읽기 (블록): 3,375 23,455.5</p> <p>블록 변경 사항: 21,615 150,945.1</p> <p>물리적 읽기 (블록): 13,515 94,415.4</p> <p>물리적 3,461 24.1</p>	

작업	설명	필요한 기술
	<p>쓰기 (블록):</p> <p>입출력 요청 읽기: 3,584 24.9</p> <p>입출력 요청 작성: 574.1 4.0</p> <p>읽기 입출력 (MB): 106.1 0.7</p> <p>쓰기 입출력 (메가바이트): 27.1 0.2</p>	

작업	설명	필요한 기술
	<p>이 트):</p> <p>메 0.0 0.0</p> <p>신</p> <p>저</p> <p>스</p> <p>캔</p> <p>행:</p> <p>세</p> <p>션</p> <p>논</p> <p>리</p> <p>적</p> <p>읽</p> <p>기</p> <p>IM:</p> <p>사 1,24 8.7</p> <p>용</p> <p>자</p> <p>통</p> <p>화:</p> <p>파 4,62 32.2</p> <p>싱</p> <p>(SQL</p> <p>하 8.9 0.1</p> <p>드</p> <p>파</p> <p>싱</p> <p>(SQL</p> <p>SQL 824 5.7</p> <p>작</p>	

작업	설명	필요한 기술
	<p>업 영 역 (MB)</p> <p>로 1.7 0.0 그 온:</p> <p>실 136,1 950.4 행 횃 수 (SQL</p> <p>를 22.9 0.2 백:</p> <p>트 143.1 랜 잭 션:</p> <p>이 정보를 기반으로 IOPS와 처리량을 다음과 같이 계산할 수 있습니다.</p> <p>IOPS = 읽기 I/O 요청: + 쓰기 I/O 요청 = 3,586.8 + 574.7 = 4134.5</p> <p>처리량 = 물리적 읽기 (블록) + 물리적 쓰기 (블록) = 13,575.1 + 3,467.3 = 17,042.4</p>	

작업	설명	필요한 기술
	<p>Oracle의 블록 크기는 8KB이므로 총 처리량을 다음과 같이 계산할 수 있습니다.</p> <p>총 처리량 (MB) 은 $17042.4 * 8 * 1024 / 1024 / 1024 = 133.2\text{MB}$ 입니다.</p> <p>경고: 로드 프로필을 사용하여 인스턴스 크기를 추정하지 마십시오. 인스턴스 활동 통계 또는 I/O 프로필만큼 정확하지는 않습니다.</p>	

작업	설명	필요한 기술																
<p>옵션 2: 인스턴스 활동 통계 사용.</p>	<p>12c 이전의 Oracle 데이터베이스 버전을 사용하는 경우 AWR 보고서의 인스턴스 활동 통계 섹션을 사용하여 IOPS 및 처리량을 추정할 수 있습니다. 다음 표에서는 이 섹션의 예제를 보여줍니다.</p> <table border="1" data-bbox="592 619 1031 1858"> <thead> <tr> <th>통계</th> <th>합계</th> <th>초당</th> <th>트랜스당</th> </tr> </thead> <tbody> <tr> <td>물리적 읽기, 총 I/O 요청 수</td> <td>2,547,317</td> <td>3,610</td> <td>25.11</td> </tr> <tr> <td>물리적 읽기 총 바이트 수</td> <td>80,776,612</td> <td>114,482</td> <td>796,1498</td> </tr> <tr> <td>물리적 쓰기 총 IO 요청 수</td> <td>534,1908</td> <td>757.11</td> <td>5.27</td> </tr> </tbody> </table>	통계	합계	초당	트랜스당	물리적 읽기, 총 I/O 요청 수	2,547,317	3,610	25.11	물리적 읽기 총 바이트 수	80,776,612	114,482	796,1498	물리적 쓰기 총 IO 요청 수	534,1908	757.11	5.27	<p>DBA</p>
통계	합계	초당	트랜스당															
물리적 읽기, 총 I/O 요청 수	2,547,317	3,610	25.11															
물리적 읽기 총 바이트 수	80,776,612	114,482	796,1498															
물리적 쓰기 총 IO 요청 수	534,1908	757.11	5.27															

작업	설명	필요한 기술
	<p>물리 25,517 36,165 251,508 적 8,849, 1.84 8 쓰기 총 바이 트 수</p> <p>이 정보를 기반으로 총 IOPS와 처리량을 다음과 같이 계산할 수 있습니다.</p> <p>총 IOPS = 3,610.28 + 757.11 = 4367</p> <p>총 Mbps = 114,482,426.26 + 36,165,631.84 = 150648058.1 / 1024 / 1024 = 143Mbps</p>	

작업	설명	필요한 기술																				
<p>옵션 3: I/O 프로필을 사용합니다.</p>	<p>Oracle Database 12c의 AWR 보고서에는 모든 정보를 단일 테이블에 표시하고 데이터베이스 성능에 대한 보다 정확한 데이터를 제공하는 I/O 프로필 섹션이 포함되어 있습니다. 다음 표에서는 이 섹션의 예제를 보여줍니다.</p> <table border="1" data-bbox="591 653 1024 1745"> <thead> <tr> <th></th> <th>초당 읽기</th> <th>초당 읽기</th> <th>초당 쓰기 +쓰기</th> </tr> </thead> <tbody> <tr> <td>총 요청 수:</td> <td>4,367.</td> <td>3,610.</td> <td>757.1</td> </tr> <tr> <td>데이터베이스 요청:</td> <td>4,161.</td> <td>3,586.</td> <td>574.7</td> </tr> <tr> <td>최적화된 요청:</td> <td>0.0</td> <td>0.0</td> <td>0.0</td> </tr> <tr> <td>다시 실행 요청:</td> <td>179.3</td> <td>2.8</td> <td>176.6</td> </tr> </tbody> </table>		초당 읽기	초당 읽기	초당 쓰기 +쓰기	총 요청 수:	4,367.	3,610.	757.1	데이터베이스 요청:	4,161.	3,586.	574.7	최적화된 요청:	0.0	0.0	0.0	다시 실행 요청:	179.3	2.8	176.6	<p>DBA</p>
	초당 읽기	초당 읽기	초당 쓰기 +쓰기																			
총 요청 수:	4,367.	3,610.	757.1																			
데이터베이스 요청:	4,161.	3,586.	574.7																			
최적화된 요청:	0.0	0.0	0.0																			
다시 실행 요청:	179.3	2.8	176.6																			

작업	설명	필요한 기술
	합계 143.7 109.2 34.5 (MB):	
	데이 터베 이스 (MB):	
	최적화된 합계 (MB):	
	다시 실행 (MB):	
	데이 터베 이스 (블록):	
	버퍼 캐시를 통해 (블록):	
	다이렉트 (블록):	

작업	설명	필요한 기술
	<p>이 표는 처리량 및 총 IOPS에 대한 다음 값을 제공합니다.</p> <p>처리량 = 143MBPS(합계로 표시된 다섯 번째 행, 두 번째 열부터)</p> <p>IOPS = 4,367.4(총 요청 수로 표시된 첫 번째 행, 두 번째 열부터)</p>	
<p>옵션 4: AWR 보기를 사용합니다.</p>	<p>AWR 보기를 사용하여 동일한 IOPS 및 처리량 정보를 볼 수 있습니다. 이 정보를 확인하려면 다음 쿼리를 사용합니다.</p> <pre data-bbox="594 905 1029 1537"> break on report compute sum of Value on report select METRIC_NAME, avg(AVERAGE) as "Value" from dba_hist_ sysmetric_summary where METRIC_NAME in ('Physical Read Total IO Requests Per Sec', 'Physical Write Total IO Requests Per Sec') group by metric_name; </pre>	<p>DBA</p>

CPU 요구 사항 추정

작업	설명	필요한 기술
<p>방법을 선택합니다.</p>	<p>다음 세 가지 방법으로 대상 데이터베이스에 필요한 CPU를 추정할 수 있습니다.</p> <ul style="list-style-type: none"> • 프로세서의 실제 사용 가능한 코어 사용 • OS 통계를 기반으로 활용된 코어 사용 • 데이터베이스 통계를 기반으로 활용된 코어 사용 <p>코어를 사용하는 경우 OS 통계 대신 데이터베이스 지표 방법을 사용하는 것이 좋습니다. 마이그레이션하려는 데이터베이스에서만 사용하는 CPU를 기반으로 하기 때문입니다. (OS 통계에는 다른 프로세스의 CPU 사용량도 포함됩니다.) 또한 마이그레이션 후 성능을 개선하려면 ADDM 보고서에서 CPU 관련 권장 사항을 확인해야 합니다.</p> <p>CPU 생성량을 기반으로 요구 사항을 추정할 수도 있습니다. 여러 세대의 CPU를 사용하는 경우 최적의 워크로드 성능을 위한 vCPU 수 설명 백서의 지침에 따라 대상 데이터베이스의 필요한 CPU를 추정할 수 있습니다.</p>	<p>DBA</p>

작업	설명	필요한 기술
<p>옵션 1: 사용 가능한 코어를 기준으로 요구 사항을 추정합니다.</p>	<p>AWR 보고서:</p> <ul style="list-style-type: none"> • CPU는 논리적 및 가상 CPU를 의미합니다. • 코어는 물리적 CPU 칩셋의 프로세서 수입입니다. • 소켓은 칩을 보드에 연결하는 물리적 장치입니다. 멀티 코어 프로세서에는 여러 개의 CPU 코어가 포함된 소켓이 있습니다. <p>다음 두 가지 방법으로 사용 가능한 코어를 추정할 수 있습니다.</p> <ul style="list-style-type: none"> • OS 명령 사용 • AWR 보고서 사용 <p>OS 명령을 사용하여 사용 가능한 코어를 추정하려면</p> <p>다음 명령을 사용하여 프로세서의 코어 수를 계산합니다.</p> <pre style="border: 1px solid #ccc; border-radius: 10px; padding: 10px; margin-top: 10px;">\$ cat /proc/cpuinfo grep "cpu cores" uniq cpu cores : 4 cat /proc/cpuinfo egrep "core id physical id" tr -d "\n" sed s/physical/\nphysical/g grep -v ^\$ sort uniq wc -l</pre>	<p>DBA</p>

작업	설명	필요한 기술
	<p>다음 명령을 사용하여 프로세서의 소켓 수를 계산합니다.</p> <pre data-bbox="594 331 1027 527"> grep "physical id" / proc/cpuinfo sort -u physical id : 0 physical id : 1 </pre> <p>참고: nmon, sar와 같은 OS 명령을 사용하여 CPU 사용률을 추출하는 것은 권장하지 않습니다. 이러한 계산에는 다른 프로세스의 CPU 사용률이 포함되며 데이터베이스에서 사용하는 실제 CPU가 반영되지 않을 수 있기 때문입니다.</p> <p>AWR 보고서를 사용하여 사용 가능한 코어를 추정하려면</p> <p>또한 AWR 보고서의 첫 번째 섹션에서 CPU 사용률을 도출할 수 있습니다. 다음은 보고서에서 발췌한 내용입니다.</p> <pre data-bbox="594 1346 1027 1818"> C DB 인 인 시 릴 RA C Id 스 스 작 리 톱 턴 턴 시 스 스 스 간 번 호 X <DE XX> 1 05- 12.1 야 Sep 0 니 23:(요 </pre>	

작업	설명	필요한 기술
	<div data-bbox="592 210 1031 745" style="border: 1px solid #ccc; padding: 5px;"> <pre> 호 플 CPU 코 소 메 스 랫 코어 소 커 모 트 폼 리 이 (GB) 름 <ho Linu 80 80 2 441.7 e> x86 64 비 트 </pre> </div> <p data-bbox="592 808 1031 1228">이 예제에서 CPU 개수는 80이며, 이는 논리적(가상) CPU임을 나타냅니다. 또한 이 구성에는 소켓 2개, 각 소켓에 물리적 프로세서 1개(물리적 프로세서 총 2개)가 있고 각 물리적 프로세서 또는 소켓에는 40개의 코어가 있다는 것도 알 수 있습니다.</p>	

작업	설명	필요한 기술																																				
<p>옵션 2: OS 통계를 사용하여 CPU 사용률을 추정합니다.</p>	<p>OS CPU 사용량 통계는 OS에서 직접 확인(sar 또는 다른 호스트 OS 유틸리티 사용)하거나 AWR 보고서의 운영 체제 통계 섹션에서 IDLE/(IDLE+BUSY) 값을 검토하여 확인할 수 있습니다. v\$osstat에서 직접 CPU 사용 시간(초)을 확인할 수 있습니다. AWR 및 Statspack 보고서는 운영 체제 통계 섹션에서도 이 데이터를 표시합니다.</p> <p>동일한 상자에 여러 데이터베이스가 있는 경우 BUSY_TIME의 v\$osstat 값은 모두 동일합니다.</p> <table border="1" data-bbox="592 1029 1031 1764"> <thead> <tr> <th>통계</th> <th>값</th> <th>종료 값</th> </tr> </thead> <tbody> <tr> <td>FREE_M</td> <td>6,810,67</td> <td>12,280,79</td> </tr> <tr> <td>RY_BYT</td> <td>,248</td> <td>9,232</td> </tr> <tr> <td>INACTIV</td> <td>175,627,</td> <td>160,380,6</td> </tr> <tr> <td>MEMOR</td> <td>33,632</td> <td>53,568</td> </tr> <tr> <td>TES</td> <td></td> <td></td> </tr> <tr> <td>SWAP_F</td> <td>17,145,6</td> <td>17,145,87</td> </tr> <tr> <td>_BYTES</td> <td>4,336</td> <td>2,384</td> </tr> <tr> <td>BUSY_T</td> <td>1,305,56</td> <td></td> </tr> <tr> <td></td> <td>,937</td> <td></td> </tr> <tr> <td>IDLE_TIM</td> <td>4,312,71</td> <td></td> </tr> <tr> <td></td> <td>,839</td> <td></td> </tr> </tbody> </table>	통계	값	종료 값	FREE_M	6,810,67	12,280,79	RY_BYT	,248	9,232	INACTIV	175,627,	160,380,6	MEMOR	33,632	53,568	TES			SWAP_F	17,145,6	17,145,87	_BYTES	4,336	2,384	BUSY_T	1,305,56			,937		IDLE_TIM	4,312,71			,839		<p>DBA</p>
통계	값	종료 값																																				
FREE_M	6,810,67	12,280,79																																				
RY_BYT	,248	9,232																																				
INACTIV	175,627,	160,380,6																																				
MEMOR	33,632	53,568																																				
TES																																						
SWAP_F	17,145,6	17,145,87																																				
_BYTES	4,336	2,384																																				
BUSY_T	1,305,56																																					
	,937																																					
IDLE_TIM	4,312,71																																					
	,839																																					

작업	설명	필요한 기술
	IOWAIT_ 53,417,1 ME 4	
	NICE_TII 29,815	
	SYS_TIM 148,567, 70	
	USER_T 1,146,91 ,783	
	LOAD 25 29	
	VM_IN_E 593,920 ES	
	VM_OUT 327,680 TES	
	PHYSIC, 474,362, MEMOR 17,152 TES	
	NUM_CF 80	
	NUM_CF 80 ORES	
	NUM_CF 2 OCKETS	
	GLOBAL 4,194,30 CEIVE_< E_MAX	
	GLOBAL 2,097,15 ND_SIZE AX	

작업	설명	필요한 기술
	<p>TCP_RE 87,380 VE_SIZE EFAULT</p> <p>TCP_RE 6,291,45 VE_SIZE AX</p> <p>TCP_RE 4,096 VE_SIZE IN</p> <p>TCP_SE 16,384 SIZE_DE ULT</p> <p>TCP_SE 4,194,30 SIZE_M/</p> <p>TCP_SE 4,096 SIZE_MI</p> <p>시스템에 다른 주요 CPU 소비자가 없는 경우 다음 공식을 사용하여 CPU 사용률을 계산하세요.</p> <p>사용률 = 바쁜 시간 / 총 시간</p> <p>바쁜 시간 =요구 사항 = v \$osstat.BUSY_TIME</p> <p>C = 총 시간(Busy + Idle)</p> <p>C = 용량 = v\$ostat.B USY_TIME + v\$ostat.I DLE_TIME</p>	

작업	설명	필요한 기술
	$\text{사용률} = \text{BUSY_TIME} / (\text{BUSY_TIME} + \text{IDLE_TIME})$ $= -1,305,569,937 / (1,305,569,937 + 4,312,718,839)$ $= 23\% \text{ 사용}$	

작업	설명	필요한 기술																				
<p>옵션 3: 데이터베이스 메트릭을 사용하여 CPU 사용률을 추정합니다.</p>	<p>시스템에서 여러 데이터베이스가 실행 중인 경우 보고서 시작 부분에 나타나는 데이터베이스 지표를 사용할 수 있습니다.</p> <table border="1" data-bbox="591 464 1024 1434"> <thead> <tr> <th>스냅 ID</th> <th>스냅 시간</th> <th>세션</th> <th>커서/세션</th> </tr> </thead> <tbody> <tr> <td>시작 스냅:</td> <td>1846 28-Sep-09:00</td> <td>1226</td> <td>35.8</td> </tr> <tr> <td>종료 스냅:</td> <td>1854 06-Oct-13:00</td> <td>1876</td> <td>41.1</td> </tr> <tr> <td>경과:</td> <td>11,700(분)</td> <td></td> <td></td> </tr> <tr> <td>DB 시간:</td> <td>312,000(분)</td> <td></td> <td></td> </tr> </tbody> </table> <p>CPU 사용률 지표는 다음 공식을 사용하세요.</p> <p>데이터베이스 CPU 사용량(사용 가능한 CPU 성능의 비율(%)) = CPU 시간 / NUM_CPUS / 경과 시간</p>	스냅 ID	스냅 시간	세션	커서/세션	시작 스냅:	1846 28-Sep-09:00	1226	35.8	종료 스냅:	1854 06-Oct-13:00	1876	41.1	경과:	11,700(분)			DB 시간:	312,000(분)			<p>DBA</p>
스냅 ID	스냅 시간	세션	커서/세션																			
시작 스냅:	1846 28-Sep-09:00	1226	35.8																			
종료 스냅:	1854 06-Oct-13:00	1876	41.1																			
경과:	11,700(분)																					
DB 시간:	312,000(분)																					

작업	설명	필요한 기술
	<p>여기서 CPU 사용량은 CPU 시간으로 설명되며 CPU를 기다리는 시간이 아니라 CPU에 소요된 시간을 나타냅니다. 이 계산 결과는 다음과 같습니다.</p> $= 312,625.40 / 11,759.64/80$ <p>= CPU의 33% 사용 중</p> $\text{코어 수}(33\%) * 80 = 26.4\text{코어}$ <p>총 코어 수 = 26.4 * (120%) = 31.68코어</p> <p>이 두 값 중 더 큰 값을 사용하여 Amazon RDS 또는 Aurora DB 인스턴스의 CPU 사용률을 계산할 수 있습니다.</p> <p>참고: IBM AIX에서는 계산된 사용률이 운영 체제 또는 데이터베이스의 값과 일치하지 않습니다. 이러한 값은 다른 운영 체제에서는 일치합니다.</p>	

메모리 요구 사항 추정

작업	설명	필요한 기술
<p>메모리 통계를 사용하여 메모리 요구 사항을 추정합니다.</p>	<p>AWR 보고서를 사용하여 원본 데이터베이스의 메모리를 계산하고 대상 데이터베이스와 일치시킬 수 있습니다. 또한 기존 데이터베이스의 성능을 확인하고 메모리 요구 사항을 줄여</p>	<p>DBA</p>

작업	설명	필요한 기술
	<p>비용을 절감하거나 요구 사항을 높여 성능을 향상시켜야 합니다. 이를 위해서는 AWR 응답 시간과 애플리케이션의 서비스 수준에 관한 계약(SLA)에 대한 상세한 분석이 필요합니다. Oracle 시스템 글로벌 영역(SGA)와 프로그램 글로벌 영역(PGA) 사용량의 합계를 Oracle에 대한 예상 메모리 사용률로 사용합니다. 대상 메모리의 크기 요구 사항을 결정하려면 OS에 20%를 추가합니다. Oracle RAC의 경우 공통 블록에 저장되므로 모든 RAC 노드의 예상 메모리 사용률 합계를 사용하고 총 메모리를 줄이세요.</p> <p>1. 인스턴스 효율성 백분율 표에서 지표를 확인하세요. 표에서는 다음 용어를 사용합니다.</p> <ul style="list-style-type: none"> • 버퍼 검색% 결과는 물리적 I/O를 수행하지 않고 버퍼 캐시에서 특정 블록을 발견한 횟수의 백분율입니다. 성능을 높이려면 100%를 목표로 하세요. • 버퍼 노웨이트 %는 100%에 가까워야 합니다. • 래치 검색 결과 %는 100%에 가까워야 합니다. 	

작업	설명	필요한 기술												
	<ul style="list-style-type: none"> % 비구문분석 CPU는 비구문분석 작업에 소요된 CPU 시간의 백분율입니다. 이 값은 100%에 가까워야 합니다. <p>인스턴스 효율성 백분율(목표 100%)</p> <table border="1" data-bbox="617 630 1039 1848"> <tr> <td>버퍼노웨이사이트 %:</td> <td>99.99</td> <td>다시 실행% NoWait</td> <td>100.00</td> </tr> <tr> <td>버퍼검색결과 %:</td> <td>99.84</td> <td>인메모리정렬 %:</td> <td>100.00</td> </tr> <tr> <td>라이브리검색결과 %:</td> <td>748.7</td> <td>소프트구문분석 %:</td> <td>99.81</td> </tr> </table>	버퍼노웨이사이트 %:	99.99	다시 실행% NoWait	100.00	버퍼검색결과 %:	99.84	인메모리정렬 %:	100.00	라이브리검색결과 %:	748.7	소프트구문분석 %:	99.81	
버퍼노웨이사이트 %:	99.99	다시 실행% NoWait	100.00											
버퍼검색결과 %:	99.84	인메모리정렬 %:	100.00											
라이브리검색결과 %:	748.7	소프트구문분석 %:	99.81											

작업	설명	필요한 기술
	구문 분석을 위한 실행 %: 96.61 래치 검색 결과 %: 100.00	
	구문 분석 CPU에서 구문 분석 경과 %: 72.73 % 비구문 분석 CPU: 99.21	
	플래시 캐시 검색 결과: 0.00	

작업	설명	필요한 기술												
	<p>과 %:</p> <p>이 예시에서는 모든 지표가 괜찮아 보이므로 기존 데이 터베이스의 SGA 및 PGA를 용량 계획 요구 사항으로 사 용할 수 있습니다.</p> <p>2. 메모리 통계 섹션을 확인하 고 SGA나 PGA를 계산하세 요.</p> <table border="1" data-bbox="617 819 1039 1575"> <thead> <tr> <th></th> <th>시작</th> <th>종료</th> </tr> </thead> <tbody> <tr> <td>호스 트 메 모리 (MB):</td> <td>452,387</td> <td>452,387.3</td> </tr> <tr> <td>SGA 사 용량 (MB):</td> <td>220,544</td> <td>220,544.0</td> </tr> <tr> <td>PGA 사 용량 (MB):</td> <td>36,874.9</td> <td>45,270.0</td> </tr> </tbody> </table> <p>사용 중인 총 인스턴스 메모 리 = SGA + PGA = 220GB + 45GB = 265GB</p> <p>버퍼를 20% 추가하세요.</p>		시작	종료	호스 트 메 모리 (MB):	452,387	452,387.3	SGA 사 용량 (MB):	220,544	220,544.0	PGA 사 용량 (MB):	36,874.9	45,270.0	
	시작	종료												
호스 트 메 모리 (MB):	452,387	452,387.3												
SGA 사 용량 (MB):	220,544	220,544.0												
PGA 사 용량 (MB):	36,874.9	45,270.0												

작업	설명	필요한 기술
	<p>총 인스턴스 메모리 = 1.2 * 265GB = 318GB</p> <p>SGA와 PGA가 호스트 메모리의 70%를 차지하므로 총 메모리 요구 사항은 다음과 같습니다.</p> <p>총 호스트 메모리 = 318/0.7 = 464GB</p> <p>참고: Amazon RDS for Oracle로 마이그레이션하는 경우 PGA와 SGA는 사전 정의된 공식을 기반으로 사전 계산됩니다. 사전 계산된 값이 추정치와 비슷한지 확인하세요.</p>	

대상 데이터베이스의 DB 인스턴스 유형 결정

작업	설명	필요한 기술
<p>디스크 I/O, CPU 및 메모리 추정치를 기반으로 DB 인스턴스 유형을 결정합니다.</p>	<p>이전 단계의 추정치를 바탕으로 대상 Amazon RDS 또는 Aurora 데이터베이스의 용량은 다음과 같아야 합니다.</p> <ul style="list-style-type: none"> • 68코어 CPU • 143MBPS 처리량 • 4367 IOPS의 디스크 I/O • 464GB 메모리 <p>대상 Amazon RDS 또는 Aurora 데이터베이스에서 이러한 값을 32코어 용량, 512GB</p>	<p>DBA</p>

작업	설명	필요한 기술
	<p>RAM, 13,600Mbps의 처리량을 갖춘 db.r5.16xlarge 인스턴스 유형에 매핑할 수 있습니다. 자세한 내용은 Oracle 성능 지표를 기반으로 한 규모에 맞는 적절한 크기의 Amazon RDS 인스턴스 AWS 블로그 게시물을 참조하세요.</p>	

관련 리소스

- [Aurora DB 인스턴스 클래스](#)(Amazon Aurora 설명서)
- [Amazon RDS DB 인스턴스 스토리지](#)(Amazon RDS 설명서)
- [AWS 마이너 도구](#) (GitHub 리포지토리)

AWS DMS를 사용하여 Amazon RDS for SQL Server 테이블을 S3 버킷으로 내보내기

작성자: Subhani Shaik(AWS)

환경: PoC 또는 파일럿	소스: RDS	대상: S3
R 유형: 해당 없음	워크로드: Microsoft	기술: 데이터베이스, 클라우드 네이티브

AWS 서비스: AWS DMS, Amazon RDS, Amazon S3, AWS Secrets Manager, AWS Identity and Access Management

요약

Amazon Relational Database Service(Amazon RDS) for SQL Server는 Amazon Web Services(AWS) 클라우드의 다른 DB 엔진 연결 서버에 데이터를 로드하는 것을 지원하지 않습니다. 대신 AWS Database Migration Service(AWS DMS)를 사용하여 Amazon RDS for SQL Server 테이블을 다른 DB 엔진에서 데이터를 사용할 수 있는 Amazon Simple Storage Service(S3) 버킷으로 내보낼 수 있습니다.

AWS DMS를 통해 데이터베이스를 AWS로 빠르고 안전하게 마이그레이션할 수 있습니다. 소스 데이터베이스는 마이그레이션 중에도 완전히 작동하여 해당 데이터베이스에 사용하는 애플리케이션의 가동 중지 시간을 최소화합니다. AWS DMS는 광범위하게 사용되는 상용 및 오픈 소스 데이터베이스 간에 데이터를 마이그레이션할 수 있습니다.

이 패턴은 AWS DMS 엔드포인트를 구성할 때 AWS Secrets Manager를 사용합니다. Secrets Manager는 애플리케이션, 서비스, IT 리소스에 액세스하는 데 필요한 보안 암호를 지키도록 도와줍니다. 이 서비스를 사용하면 수명 주기 동안 데이터베이스 보안 인증 정보, API 키 및 기타 보안 암호를 손쉽게 교체, 관리 및 검색할 수 있습니다. 사용자와 애플리케이션은 Secrets Manager를 직접적으로 호출하여 보안 암호를 검색하므로 민감한 정보를 하드코딩할 필요가 없습니다. Secrets Manager는 Amazon RDS, Amazon Redshift 및 Amazon DocumentDB에 대한 통합 기능이 내장된 보안 로테이션을 제공합니다. 또한 이 서비스는 API 키 및 OAuth 토큰을 비롯한 다른 유형의 보안 암호로 확장할 수

있습니다. Secrets Manager를 사용하면 세분화된 권한을 사용하여 보안 암호에 대한 액세스를 제어하고 AWS 클라우드, 타사 서비스 및 온프레미스의 리소스에 대해 중앙에서 시크릿 로테이션을 감사할 수 있습니다.

사전 조건 및 제한 사항

사전 조건

- 활성 상태의 AWS 계정.
- S3 버킷
- Virtual Private Cloud(VPC)
- DB 서브넷
- Amazon RDS for SQL Server
- Amazon RDS 인스턴스를 사용하여 S3 버킷에 대한 액세스(객체 나열, 가져오기 및 추가)가 있는 AWS Identity and Access Management(IAM) 역할.
- RDS 인스턴스 보안 인증 정보를 저장할 Secrets Manager.

아키텍처

기술 스택

- Amazon RDS for SQL Server
- DMS
- Amazon S3
- AWS Secrets Manager

대상 아키텍처

다음 다이어그램은 AWS DMS를 사용하여 Amazon RDS 인스턴스에서 S3 버킷으로 데이터를 가져오는 아키텍처를 보여줍니다.

1. 소스 엔드포인트를 통해 소스 Amazon RDS 인스턴스에 연결하는 AWS DMS 마이그레이션 작업
2. 소스 Amazon RDS 인스턴스에서 데이터 복사
3. 대상 엔드포인트를 통해 대상 S3 버킷에 연결하는 AWS DMS 마이그레이션 작업

4. 복사된 데이터를 쉼표로 구분된 값(CSV) 형식으로 S3 버킷에 내보내기

도구

서비스

- [AWS Database Migration Service\(AWS DMS\)](#)를 사용하면 데이터 스토어를 AWS 클라우드로 마이그레이션하거나 클라우드와 온프레미스 설정 간에 데이터 스토어를 마이그레이션할 수 있습니다.
- [AWS Identity and Access Management\(IAM\)](#)는 사용자에 대한 인증 및 권한 부여를 제어함으로써 AWS 리소스에 대한 액세스를 안전하게 제어할 수 있습니다.
- [Amazon Relational Database Service\(Amazon RDS\)](#)는 AWS 클라우드에서 관계형 데이터베이스를 설정, 운영 및 조정하는 데 도움이 됩니다.
- [Amazon Simple Storage Service\(S3\)](#)는 원하는 양의 데이터를 저장, 보호 및 검색하는 데 도움이 되는 클라우드 기반 객체 스토리지 서비스입니다.
- [AWS Secrets Manager](#)를 사용하면 암호를 포함하여 코드에 하드코딩된 보안 인증을 Secrets Manager에 대한 API 호출로 대체하여 프로그래밍 방식으로 암호를 검색할 수 있습니다.

기타 서비스

- [Microsoft SQL Server Management Studio\(SSMS\)](#)는 SQL 서버 구성 요소에 대한 액세스, 구성 및 관리를 포함하여 SQL Server를 관리하기 위한 도구입니다.

에픽

Amazon RDS for SQL Server 인스턴스 구성

작업	설명	필요한 기술
Amazon RDS for SQL Server 인스턴스를 생성합니다.	1. AWS Management Console을 열고 RDS를 선택한 다음 표준 생성 옵션을 사용하여 SQL Server Express Edition, SQL Server Standard Edition 또는 SQL Server Enterprise Edition과 같은 필수 에디션	DBA, 엔지니어 DevOps

작업	설명	필요한 기술
	<p>으로 Amazon RDS 인스턴스를 생성합니다. 버전의 경우 2016 이상을 선택합니다.</p> <p>2. 템플릿에서 개발 및 테스트를 선택합니다.</p>	
인스턴스의 보안 인증 정보를 설정합니다.	<p>1. 인스턴스의 이름을 입력합니다.</p> <p>2. Amazon RDS 인스턴스의 사용자 이름과 암호를 입력합니다.</p>	DBA, 엔지니어 DevOps

작업	설명	필요한 기술
<p>인스턴스 클래스, 스토리지, Auto Scaling, 가용성을 구성합니다.</p>	<ol style="list-style-type: none"> 1. 목록에서 DB 인스턴스 클래스(표준, 메모리 최적화, 버스트 가능 클래스)를 선택합니다. 이 DB 인스턴스에 계획된 워크로드에 필요한 컴퓨팅, 네트워크 및 메모리 용량을 할당하는 DB 인스턴스 유형을 선택합니다. 자세한 내용은 AWS 설명서를 참조하세요. 2. 목록에서 스토리지 유형(범용 SSD, 프로비저닝된 IOPS SSD 또는 마그네틱)을 선택합니다. 필요에 따라 기본 스토리지 크기를 할당합니다. 3. 용량 계획에 따라 Amazon RDS 스토리지를 늘리려면 스토리지 자동 조정 활성화를 선택합니다. 4. AWS DMS는 복제 인스턴스를 사용한 다중 AZ 배포를 지원합니다. 가용 영역, 내부 하드웨어 또는 네트워크가 중단되는 경우, AWS DMS는 대기 인스턴스를 생성하고 대기 복제본에 자동 장애 조치를 통해 고가용성(HA)을 제공합니다. 가져오기의 크기에 따라 적절한 옵션을 선택합니다. 	<p>DBA, 엔지니어 DevOps</p>

작업	설명	필요한 기술
<p>VPC, 서브넷 그룹, 퍼블릭 액세스, 보안 그룹을 지정합니다.</p>	<p>Amazon RDS 인스턴스를 생성하는 데 필요한 VPC, DB 서브넷 그룹 및 VPC 보안 그룹을 선택합니다. 모범 사례를 따릅니다. 예를 들면 다음과 같습니다.</p> <ul style="list-style-type: none"> • RDS DB 인스턴스에 대한 퍼블릭 액세스를 활성화하지 마세요. • 보안 그룹에서는 CIDR 0.0.0.0/0을 사용하지 마세요. • 필요한 IP 주소 및 포트 세부 정보만 사용하여 RDS 인스턴스에 액세스합니다. 	<p>DBA, 엔지니어 DevOps</p>
<p>모니터링, 백업 및 유지 관리를 구성합니다.</p>	<ol style="list-style-type: none"> 1. 원하는 백업 옵션을 지정합니다. 기본적으로 자동 백업은 보존 기간이 7일인 상태로 활성화됩니다. 2. 적절한 자동 마이너 버전 업그레이드 및 유지 관리 기간 설정을 선택하여 Amazon RDS의 데이터베이스에 보류 중인 수정 사항 또는 유지 관리를 적용합니다. 3. 데이터베이스 생성을 선택합니다. 	<p>DBA, 엔지니어 DevOps</p>

데이터베이스 및 예제 데이터 설정

작업	설명	필요한 기술
테이블을 만들고 예제 데이터를 로드합니다.	새 데이터베이스에서 테이블을 생성합니다. 추가 정보 섹션의 예제 코드를 사용하여 데이터를 테이블에 로드합니다.	DBA, 엔지니어 DevOps

보안 인증 설정

작업	설명	필요한 기술
보안 암호를 생성합니다.	<ol style="list-style-type: none"> 콘솔에서 Secrets Manager를 열고 새 보안 암호 저장을 선택합니다. Amazon RDS for SQL Server 데이터베이스에 대한 사용자 이름과 암호를 입력합니다. <p>이 보안 암호는 AWS DMS 소스 엔드포인트에 사용됩니다.</p>	DBA, 엔지니어 DevOps

데이터베이스와 S3 버킷 간 액세스 설정

작업	설명	필요한 기술
Amazon RDS에 액세스할 수 있도록 IAM 역할을 생성합니다.	<ol style="list-style-type: none"> 콘솔에서 IAM을 선택하고 S3 버킷에 대한 읽기/쓰기 액세스 권한을 Amazon RDS에 부여하는 IAM 역할을 생성합니다. 기능에서 S3 통합을 선택합니다. 	DBA, 엔지니어 DevOps

S3 버킷을 생성합니다.

작업	설명	필요한 기술
S3 버킷을 생성합니다.	Amazon RDS for SQL Server의 데이터를 저장하려면 콘솔에서 S3를 선택한 다음 버킷 생성을 선택합니다. S3 버킷에 공개적으로 액세스할 수 없도록 합니다.	DBA, 엔지니어 DevOps

AWS DMS와 S3 버킷 간 액세스를 설정합니다.

작업	설명	필요한 기술
AWS DMS가 Amazon S3에 액세스를 하기 위한 IAM 역할을 생성합니다.	AWS DMS가 S3 버킷의 객체를 나열하고, 가져오고, 넣을 수 있는 IAM 역할을 생성합니다.	DBA, 엔지니어 DevOps

AWS DMS 구성

작업	설명	필요한 기술
AWS DMS 소스 엔드포인트를 생성합니다.	<ol style="list-style-type: none"> 콘솔에서 Database Migration Service를 선택하고 엔드포인트를 선택합니다. RDS DB 인스턴스 선택 확인란을 선택하여 소스 엔드포인트를 생성합니다. 소스 엔진의 경우 Microsoft SQL Server를 선택합니다. 엔드포인트 데이터베이스에 액세스에서 AWS Secrets Manager를 선택하고 이전에 생성한 보안 암호 및 IAM 역 	DBA, 엔지니어 DevOps

작업	설명	필요한 기술
	<p>할과 데이터베이스 이름을 입력합니다.</p> <p>4. 소스 엔드포인트를 테스트합니다.</p>	
AWS DMS 대상 엔드포인트를 생성합니다.	<p>Amazon S3를 대상 엔진으로 선택하여 대상 엔드포인트를 생성합니다.</p> <p>이전에 생성한 IAM 역할의 S3 버킷 이름과 폴더 이름을 제공합니다.</p>	DBA, 엔지니어 DevOps
AWS DMS 복제 인스턴스를 생성합니다.	동일한 VPC 및 서브넷 그룹에서 AWS DMS 복제 인스턴스를 생성합니다. 인스턴스 클래스에 대한 자세한 내용은 AWS 설명서 를 참조하세요.	DBA, 엔지니어 DevOps
AWS DMS 마이그레이션 작업을 생성합니다.	Amazon RDS for SQL Server에서 S3 버킷으로 데이터를 내보내려면 데이터베이스 마이그레이션 작업을 생성합니다. 마이그레이션 유형에서 기존 데이터 마이그레이션을 선택합니다. 생성한 AWS DMS 엔드포인트와 복제 인스턴스를 선택합니다.	DBA, 엔지니어 DevOps

S3 버킷으로 데이터 내보내기

작업	설명	필요한 기술
데이터베이스 마이그레이션 작업을 실행합니다.	SQL Server 테이블 데이터를 내보내려면 데이터베이스 마	DBA, 엔지니어 DevOps

작업	설명	필요한 기술
	이그레이션 작업을 시작합니다. 이 작업은 Amazon RDS for SQL Server에서 S3 버킷으로 데이터를 CSV 형식으로 내보냅니다.	

리소스 정리

작업	설명	필요한 기술
리소스를 삭제합니다.	<p>추가 비용이 발생하지 않도록 콘솔을 사용하여 다음 순서대로 리소스를 삭제합니다.</p> <ol style="list-style-type: none"> 1. 마이그레이션 작업 2. 복제 인스턴스 3. 엔드포인트 4. S3 버킷 5. 데이터베이스 인스턴스 	DBA, 엔지니어 DevOps

관련 리소스

- [DMS](#)
- [Amazon S3](#)
- [Amazon RDS for SQL Server](#)
- [Amazon S3 통합](#)

추가 정보

데이터베이스와 테이블을 만들고 예제 데이터를 로드하려면 다음 코드를 사용합니다.

```
--Step1: Database creation in RDS SQL Server
CREATE DATABASE [Test_DB]
```

```
ON PRIMARY
( NAME = N'Test_DB', FILENAME = N'D:\rdsdbdata\DATA\Test_DB.mdf' , SIZE = 5120KB ,
  FILEGROWTH = 10%)
LOG ON
( NAME = N'Test_DB_log', FILENAME = N'D:\rdsdbdata\DATA\Test_DB_log.ldf' , SIZE =
  1024KB , FILEGROWTH = 10%)
GO

--Step2: Create Table
USE Test_DB
GO
Create Table Test_Table(ID int, Company Varchar(30), Location Varchar(20))

--Step3: Load sample data.
USE Test_DB
GO
Insert into Test_Table values(1,'AnyCompany','India')
Insert into Test_Table values(2,'AnyCompany','USA')
Insert into Test_Table values(3,'AnyCompany','UK')
Insert into Test_Table values(4,'AnyCompany','Hyderabad')
Insert into Test_Table values(5,'AnyCompany','Banglore')
```

Aurora PostgreSQL의 동적 SQL 명령문에서 익명 블록 처리

작성자: Anuradha Chintha (AWS)

환경: PoC 또는 파일럿	소스: 관계형 데이터베이스	대상: PostgreSQL
R 유형: 리아키텍트	워크로드: Oracle, 오픈 소스	기술: 데이터베이스, 마이그레이션

AWS 서비스: Amazon Aurora, Amazon RDS

요약

이 패턴은 동적 SQL 명령문에서 익명 블록을 처리할 때 발생하는 오류를 방지하는 방법을 보여줍니다. AWS Schema Conversion Tool을 사용하여 Oracle 데이터베이스를 Aurora PostgreSQL 호환 버전 데이터베이스로 변환할 때 오류 메시지가 나타납니다. 오류를 방지하려면 OUT 바인드 변수의 값을 알아야 하지만 SQL 명령문을 실행하기 전까지 OUT 바인드 변수의 값을 알 수 없습니다. 이 오류는 AWS Schema Conversion Tool(AWS SCT)이 동적 SQL 명령문 내의 로직을 이해하지 못하기 때문에 발생합니다. AWS SCT는 PL/SQL 코드(즉, 함수, 프로시저, 패키지)의 동적 SQL 명령문을 변환할 수 없습니다.

사전 조건 및 제한 사항

사전 조건

- 활성 AWS 계정
- [Aurora PostgreSQL 데이터베이스\(DB\) 인스턴스](#)
- [Oracle DB 인스턴스용 Amazon Relational Database Service\(Amazon RDS\)](#)
- [PostgreSQL 인터랙티브 터미널 \(psql\)](#)
- [SQL *Plus](#)
- 대상 데이터베이스의 AWS_ORACLE_EXT 스키마([AWS SCT 확장 팩](#)의 일부)
- 최신 버전의 [AWS Schema Conversion Tool\(AWS SCT\)](#) 및 필수 드라이버

아키텍처

소스 기술 스택

- 온프레미스 Oracle Database 버전 10g 이상

대상 기술 스택

- Amazon Aurora PostgreSQL
- Amazon RDS for PostgreSQL
- AWS Schema Conversion Tool(AWS SCT)

마이그레이션 아키텍처

다음 다이어그램은 AWS SCT 및 Oracle OUT 바인드 변수를 사용하여 애플리케이션 코드에서 내장된 SQL 명령문을 스캔하고 코드를 Aurora 데이터베이스에서 사용할 수 있는 호환 가능한 형식으로 변환하는 방법을 보여줍니다.

이 다이어그램은 다음 워크플로우를 보여줍니다.

1. Aurora PostgreSQL을 대상 데이터베이스로 사용하여 소스 데이터베이스에 대한 AWS SCT 보고서를 생성합니다.
2. 동적 SQL 코드 블록(AWS SCT가 오류를 야기한 블록)에서 익명 블록을 식별합니다.
3. 코드 블록을 수동으로 변환하고 대상 데이터베이스에 코드를 배포합니다.

도구

서비스

- [Amazon Aurora PostgreSQL 호환 버전](#)은 PostgreSQL 배포를 설정, 운영 및 확장할 수 있고 ACID를 준수하는 완전관리형 관계형 데이터베이스 엔진입니다.
- [Oracle용 Amazon Relational Database Service\(Amazon RDS\)](#)는 AWS 클라우드에서 Oracle 관계형 데이터베이스를 설정, 운영 및 조정하는 데 도움이 됩니다.

- [AWS Schema Conversion Tool \(AWS SCT\)](#)은 소스 데이터베이스 스키마와 대부분의 데이터베이스 코드 객체를 대상 데이터베이스와 호환되는 형식으로 자동 변환하여 이기종 데이터베이스 마이그레이션을 예측 가능하게 합니다.

기타 도구

- [pgAdmin](#)을 사용하면 데이터베이스 서버에 연결하고 상호 작용할 수 있습니다.
- [Oracle SQL Developer](#)는 Oracle Database에서 데이터베이스를 개발하고 관리하는 데 사용할 수 있는 통합 개발 환경입니다. 이 패턴에는 [SQL *Plus](#) 또는 Oracle SQL Developer를 사용할 수 있습니다.

에픽

Oracle 소스 데이터베이스 구성

작업	설명	필요한 기술
Amazon RDS 또는 Amazon EC2에서 Oracle 인스턴스를 생성합니다.	<p>Amazon RDS에서 Oracle DB 인스턴스를 만들려면 Amazon RDS 설명서의 Oracle DB 인스턴스 생성 및 Oracle DB 인스턴스의 데이터베이스에 연결을 참조하세요.</p> <p>Amazon Elastic Compute Cloud (Amazon EC2)에서 Oracle DB 인스턴스를 생성하려면 AWS Prescriptive Guidance 설명서의 Oracle용 Amazon EC2를 참조하세요.</p>	DBA
마이그레이션을 위한 데이터베이스 스키마와 객체를 생성합니다.	Amazon Cloud Directory를 사용하여 데이터베이스 스키마를 생성할 수 있습니다. 자세한 내용은 Cloud Directory 설명서의 스키마 생성 을 참조하세요.	DBA

작업	설명	필요한 기술
인바운드 및 아웃바운드 보안 그룹을 구성합니다.	보안 그룹을 생성하고 구성하려면 Amazon RDS 설명서의 보안 그룹을 통한 액세스 제어 를 참조하세요.	DBA
데이터베이스가 실행 중인지 확인합니다.	데이터베이스 상태를 확인하려면 Amazon RDS 설명서의 Amazon RDS 이벤트 보기 를 참조하세요.	DBA

대상 Aurora PostgreSQL 데이터베이스 구성

작업	설명	필요한 기술
Amazon RDS에서 Aurora PostgreSQL 인스턴스를 생성합니다.	Aurora PostgreSQL 인스턴스를 생성하려면 Amazon RDS 설명서의 DB 클러스터 생성 및 Aurora PostgreSQL DB 클러스터의 데이터베이스에 연결 을 참조하세요.	DBA
인바운드 및 아웃바운드 보안 그룹을 구성합니다.	보안 그룹을 생성 및 구성하려면 Aurora 설명서의 보안 그룹을 생성하여 VPC 내의 DB 클러스터에 대한 액세스 제공 을 참조하세요.	DBA
Aurora PostgreSQL 데이터베이스가 실행 중인지 확인합니다.	데이터베이스 상태를 확인하려면 Aurora 설명서의 Amazon RDS 이벤트 보기 를 참조하세요.	DBA

AWS SCT 설정

작업	설명	필요한 기술
AWS SCT를 소스 데이터베이스에 연결합니다.	AWS SCT를 소스 데이터베이스에 연결하려면 AWS SCT 설명서의 PostgreSQL에 소스로 연결하기 를 참조하세요.	DBA
AWS SCT를 대상 데이터베이스에 연결합니다.	AWS SCT를 대상 데이터베이스에 연결하려면 AWS Schema Conversion Tool 사용 설명서의 AWS Schema Conversion Tool이란 무엇인가? 를 참조하세요.	DBA
AWS SCT에서 데이터베이스 스키마를 변환하고 자동으로 변환된 코드를 SQL 파일로 저장합니다.	AWS SCT로 변환된 파일을 저장하려면 AWS Schema Conversion Tool 사용 설명서의 변환된 스키마를 AWS SCT에 저장 및 적용 을 참조하세요.	DBA

코드 마이그레이션

작업	설명	필요한 기술
수동 변환을 위한 SQL 파일을 가져옵니다.	AWS SCT로 변환된 파일에서 수동 변환이 필요한 SQL 파일을 가져옵니다.	DBA
스크립트를 업데이트합니다.	SQL 파일을 수동으로 업데이트합니다.	DBA

관련 리소스

- [Amazon RDS](#)

- [Amazon Aurora 기능](#)

추가 정보

다음 예제 코드는 Oracle 소스 데이터베이스를 구성하는 방법을 보여줍니다.

```
CREATE or replace PROCEDURE calc_stats_new1 (  
  a NUMBER,  
  b NUMBER,  
  result out NUMBER)  
IS  
BEGIN  
  result:=a+b;  
END;  
/
```

```
set serveroutput on ;  
  
DECLARE  
  a NUMBER := 4;  
  b NUMBER := 7;  
  plsql_block VARCHAR2(100);  
  output number;  
BEGIN  
  plsql_block := 'BEGIN calc_stats_new1(:a, :b,:output); END;';  
  EXECUTE IMMEDIATE plsql_block USING a, b,out output;  
  DBMS_OUTPUT.PUT_LINE('output:' || output);  
  
END;
```

다음 예제 코드는 대상 Aurora PostgreSQL 데이터베이스를 구성하는 방법을 보여줍니다.

```
w integer,  
x integer)  
RETURNS integer  
AS  
$BODY$  
DECLARE  
begin  
return w + x ;  
end;
```

```

$BODY$
LANGUAGE plpgsql;

CREATE OR REPLACE FUNCTION test_pg.init()
RETURNS void
AS
$BODY$
BEGIN
if aws_oracle_ext.is_package_initialized
    ('test_pg' ) then
    return;
end if;
perform aws_oracle_ext.set_package_initialized
    ('test_pg' );

PERFORM aws_oracle_ext.set_package_variable('test_pg', 'v_output', NULL::INTEGER);
PERFORM aws_oracle_ext.set_package_variable('test_pg', 'v_status', NULL::text);
END;
$BODY$
LANGUAGE plpgsql;

DO $$
declare
v_sql text;
v_output_loc int;
a integer :=1;
b integer :=2;
BEGIN
perform test_pg.init();
--raise notice 'v_sql %',v_sql;
execute 'do $$ declare v_output_l int; begin select * from test_pg.calc_stats_new1('||
a||','||b||') into v_output_l;
PERFORM aws_oracle_ext.set_package_variable('test_pg', 'v_output', v_output_l) ;
end; $$' ;
v_output_loc := aws_oracle_ext.get_package_variable('test_pg', 'v_output');
raise notice 'v_output_loc %',v_output_loc;
END ;
$$

```

Aurora PostgreSQL-Compatible에서 오버로드된 Oracle 함수 처리

작성자: Sumana Yanamandra(AWS)

환경: PoC 또는 파일럿	소스: Oracle Database	대상: Aurora PostgreSQL-Compatible
R 유형: 리플랫폼	워크로드: Oracle	기술: 데이터베이스, 마이그레이션
AWS 서비스: Amazon Aurora		

요약

온프레미스 Oracle Database에서 Amazon Aurora PostgreSQL-Compatible Edition으로 마이그레이션하는 코드에는 오버로드된 함수가 포함될 수 있습니다. 이러한 함수는 정의 족, 함수 이름이 같고 입력(IN) 파라미터의 수와 데이터 유형이 동일하지만 데이터 유형이나 출력(OUT) 파라미터의 수는 다를 수 있습니다.

이러한 파라미터가 일치하지 않으면 PostgreSQL에서 문제가 발생할 수 있는데, 이는 실행할 함수를 결정하기가 어렵기 때문입니다. 이 패턴은 데이터베이스 코드를 Aurora PostgreSQL-Compatible로 마이그레이션할 때 오버로드된 함수를 처리하는 방법을 보여줍니다.

사전 조건 및 제한 사항

사전 조건

- Oracle Database 인스턴스를 소스 데이터베이스로 사용
- Aurora PostgreSQL-Compatible DB 인스턴스를 대상 데이터베이스로 사용(Aurora 설명서의 [지침 참조](#))

제품 버전

- Oracle Database 9i 이상
- Oracle SQL Developer 버전 18.4.0.376
- pgAdmin 4 클라이언트

- Aurora PostgreSQL-Compatible 버전 11 이상(Aurora 설명서의 [Amazon Aurora PostgreSQL 버전 확인 참조](#))

도구

서비스

- [Amazon Aurora PostgreSQL-Compatible Edition](#)은 PostgreSQL 배포를 설정, 운영 및 확장할 수 있는 완전 관리형 ACID 준수 관계형 데이터베이스 엔진입니다.

기타 도구

- [Oracle SQL Developer](#)는 기존 배포 및 클라우드 배포 모두에서 오라클 데이터베이스의 SQL 작업을 위한 무료 통합 개발 환경입니다.
- [pgAdmin](#)은 PostgreSQL을 위한 오픈 소스 관리 도구입니다. 데이터베이스 객체를 생성, 유지 관리 및 사용하는 데 도움이 되는 그래픽 인터페이스를 제공합니다.

에픽

단순 함수 생성

작업	설명	필요한 기술
PostgreSQL에서 입력 파라미터 하나와 출력 파라미터 하나를 갖는 함수를 생성합니다.	다음 예제는 Aurora PostgreSQL-Compatible에서 <code>test_overloading</code> 이름이 지정된 함수를 보여줍니다. 이 함수에는 두 개의 파라미터가 있습니다. 하나는 입력 텍스트 파라미터이고 다른 하나는 출력 텍스트 파라미터입니다. <pre>CREATE OR REPLACE FUNCTION public.te st_overloading(str1 text, OUT str2 text)</pre>	데이터 엔지니어, Aurora PostgreSQL-Compatible

작업	설명	필요한 기술
	<pre> LANGUAGE 'plpgsql' COST 100 VOLATILE AS \$BODY\$ DECLARE BEGIN str2 := 'Success'; RETURN ; EXCEPTION WHEN others THEN RETURN ; END; \$BODY\$; </pre>	
<p>PostgreSQL에서 함수를 실행합니다.</p>	<p>이전 단계에서 생성한 함수를 실행합니다.</p> <pre> select public.te st_overloading('Test'); </pre> <p>다음 출력이 표시됩니다.</p> <pre> Success </pre>	<p>데이터 엔지니어, Aurora PostgreSQL-Compatible</p>

함수 오버로드

작업	설명	필요한 기술
<p>동일한 함수 이름을 사용하여 PostgreSQL에서 오버로드된 함수를 생성할 수 있습니다.</p>	<p>Aurora PostgreSQL과 호환되는 오버로드된 함수를 생성하여 이전 함수와 동일한 함수 이름을 사용하십시오. 다음 예제 도 <code>test_overloading</code> 이름이 지정되었지만 입력 텍스트</p>	<p>데이터 엔지니어, Aurora PostgreSQL-Compatible</p>

작업	설명	필요한 기술
	<p>트 파라미터 하나, 출력 텍스트 파라미터 하나, 출력 정수 파라미터 하나 등 세 개의 파라미터가 있습니다.</p> <pre data-bbox="597 426 1027 1539"> CREATE OR REPLACE FUNCTION public.te st_overloading(str1 text, OUT str2 text, OUT num1 integer) LANGUAGE 'plpgsql' COST 100 VOLATILE AS \$BODY\$ DECLARE str3 text; BEGIN str2 := 'Success'; num1 := 100; RETURN ; EXCEPTION WHEN others THEN RETURN ; END; \$BODY\$; </pre>	

작업	설명	필요한 기술
<p>PostgreSQL에서 함수를 실행합니다.</p>	<p>이 함수를 실행하면 실행이 실패하고 다음 오류 메시지가 표시됩니다.</p> <pre data-bbox="597 394 1024 667"> ERROR: cannot change return type of existing function HINT: Use DROP FUNCTION test_over loading(text) first. </pre> <p>이는 Aurora PostgreSQL-Compatible이 함수 오버로딩을 직접 지원하지 않기 때문에 발생합니다. 입력 파라미터는 동일하지만 두 번째 버전의 함수에서는 출력 파라미터 수가 다르기 때문에 실행할 함수를 식별할 수 없습니다.</p>	<p>데이터 엔지니어, Aurora PostgreSQL-Compatible</p>

해결 방법 적용

작업	설명	필요한 기술
<p>첫 번째 출력 파라미터에 INOUT을 추가합니다.</p>	<p>이 문제를 해결하려면 첫 번째 출력 파라미터를 INOUT으로 표현하여 함수 코드를 수정하십시오.</p> <pre data-bbox="597 1591 1024 1841"> CREATE OR REPLACE FUNCTION public.te st_overloading(str1 text, INOUT str2 text, </pre>	<p>데이터 엔지니어, Aurora PostgreSQL-Compatible</p>

작업	설명	필요한 기술
	<pre> OUT num1 integer) LANGUAGE 'plpgsql' COST 100 VOLATILE AS \$BODY\$ DECLARE str3 text; BEGIN str2 := 'Success'; num1 := 100; RETURN ; EXCEPTION WHEN others THEN RETURN ; END; \$BODY\$; </pre>	
<p>수정된 함수를 실행합니다.</p>	<p>다음 쿼리를 사용하여 업데이트한 함수를 실행합니다. 오류를 방지하기 위해 이 파라미터를 INOUT으로 선언했으므로 이 함수의 두 번째 인수로 null 값을 전달합니다.</p> <pre> select public.te st_overloading('Test', null); </pre> <p>이제 함수가 성공적으로 생성되었습니다.</p> <pre> Success, 100 </pre>	<p>데이터 엔지니어, Aurora PostgreSQL-Compatible</p>

작업	설명	필요한 기술
결과 검증	오버로드된 함수가 포함된 코드가 성공적으로 변환되었는지 확인합니다.	데이터 엔지니어, Aurora PostgreSQL-Compatible

관련 리소스

- [Amazon Aurora PostgreSQL 작업](#)(Aurora 설명서)
- [Oracle의 함수 오버로딩](#)(Oracle 설명서)
- [PostgreSQL의 함수 오버로딩](#)(PostgreSQL 설명서)

DynamoDB 태깅 적용 지원

작성자: Mansi Suratwala(AWS)

환경: 프로덕션

기술: 클라우드 네이티브, 보안, 자격 증명, 규정 준수, 데이터베이스

워크로드: 기타 모든 워크로드

AWS 서비스: 아마존 CloudWatch, 아마존 DynamoDB, AWS Lambda, 아마존 SNS

요약

이 패턴은 미리 정의된 Amazon DynamoDB 태그가 Amazon Web Services(AWS) 클라우드의 DynamoDB 리소스에서 누락되거나 제거될 때 자동 알림을 설정합니다.

DynamoDB는 완전관리형 NoSQL 데이터베이스 서비스로서 확장성과 함께 빠르고 예측 가능한 성능을 제공합니다. DynamoDB를 사용하면 분산 데이터베이스를 운영하고 규모를 조정하는 데 따르는 관리 부담을 덜 수 있습니다. DynamoDB 는 하드웨어 프로비저닝, 설정 및 구성, 복제, 소프트웨어 패치 또는 클러스터 규모 조정에 대해 걱정할 필요가 없게 합니다.

이 패턴은 Amazon 이벤트 CloudWatch 이벤트와 AWS Lambda 함수를 생성하는 AWS CloudFormation 템플릿을 사용합니다. 이벤트는 AWS를 사용하여 새 DynamoDB 태깅 정보 또는 기존 DynamoDB 태깅 정보를 감시합니다. CloudTrail 사전 정의된 태그가 누락되거나 제거되면 Lambda 함수가 CloudWatch 트리거되어 위반 사실을 알리는 Amazon Simple Service (Amazon SNS) 알림이 전송됩니다.

사전 조건 및 제한 사항

사전 조건

- 활성 상태의 AWS 계정.
- Lambda 함수를 실행하는 Python 스크립트가 포함된 Lambda .zip 파일용 Amazon Simple Storage Service(S3) 버킷

제한 사항

- 솔루션은 또는 이벤트가 발생할 때만 작동합니다. TagResource UntagResource CloudTrail 다른 이벤트에 대한 알림은 생성되지 않습니다.

아키텍처

대상 기술 스택

- Amazon DynamoDB
- AWS CloudTrail
- 아마존 CloudWatch
- AWS Lambda
- Amazon S3
- Amazon SNS

대상 아키텍처

자동화 및 규모 조정

여러 AWS 지역 및 계정에 대해 AWS CloudFormation 템플릿을 여러 번 사용할 수 있습니다. 템플릿은 각 리전 또는 계정에서 한 번만 실행하면 됩니다.

도구

도구

- [Amazon DynamoDB](#) – DynamoDB는 완전관리형 NoSQL 데이터베이스 서비스로서 확장성과 함께 빠르고 예측 가능한 성능을 제공합니다.
- [AWS CloudTrail](#) — CloudTrail AWS 계정의 거버넌스, 규정 준수, 운영 및 위험 감사를 지원하는 AWS 서비스입니다. 사용자, 역할 또는 AWS 서비스가 수행한 작업은 에서 이벤트로 기록됩니다 CloudTrail.
- [Amazon CloudWatch Events](#) — Amazon CloudWatch Events는 AWS 리소스의 변경 사항을 설명하는 시스템 이벤트의 스트림을 거의 실시간으로 제공합니다.

- [AWS Lambda](#) - Lambda는 서버를 프로비저닝하거나 관리할 필요 없이 코드를 실행할 수 있게 해주는 컴퓨팅 서비스입니다. Lambda는 필요 시에만 코드를 실행하며, 일일 몇 개의 요청에서 초당 수천 개의 요청까지 자동으로 규모를 조정합니다.
- [Amazon S3](#)-Amazon Simple Storage Service(S3)는 웹 사이트, 모바일 애플리케이션, 백업, 데이터 레이크 등 다양한 스토리지 솔루션에 사용할 수 있는 확장성이 뛰어난 객체 스토리지 서비스입니다.
- [Amazon SNS](#) – Amazon Simple Notification Service(SNS)는 애플리케이션, 최종 사용자 및 디바이스가 클라우드에서 즉시 알림을 전송하고 수신할 수 있게 해 주는 웹 서비스입니다.

코드

- 프로젝트의 .zip 파일은 첨부 파일로 제공됩니다.

에픽

S3 버킷 정의

작업	설명	필요한 기술
S3 버킷을 정의합니다.	Amazon S3 콘솔에서 선행 스텝을 포함하지 않는 고유한 이름을 가진 S3 버킷을 선택하거나 생성합니다. 이 S3 버킷은 Lambda 코드 .zip 파일을 호스팅합니다. S3 버킷은 모니터링 중인 DynamoDB 리소스와 동일한 AWS 리전에 있어야 합니다.	클라우드 아키텍트

Lambda 코드를 S3 버킷에 업로드

작업	설명	필요한 기술
Lambda 코드를 S3 버킷에 업로드	첨부 파일 섹션에 제공된 Lambda 코드 .zip 파일을 S3 버킷에 업로드합니다. S3 버킷	클라우드 아키텍트

작업	설명	필요한 기술
	은 모니터링되는 DynamoDB 리소스와 동일한 리전에 있어야 합니다.	

AWS CloudFormation 템플릿 배포

작업	설명	필요한 기술
AWS CloudFormation 템플릿을 배포합니다.	AWS CloudFormation 콘솔에서 첨부 파일 섹션에 제공된 AWS CloudFormation 템플릿을 배포합니다. 다음 에픽에서 파라미터에 대한 값을 입력합니다.	클라우드 아키텍트

AWS CloudFormation 템플릿에서 파라미터 완성

작업	설명	필요한 기술
S3 버킷에 이름을 지정합니다.	첫 번째 에픽에서 생성하거나 선택한 S3 버킷의 이름을 입력합니다.	클라우드 아키텍트
Amazon S3 키를 입력합니다.	S3 버킷의 Lambda 코드 .zip 파일 위치를 선행 슬래시 없이 입력합니다(예: <folder>/<file-name>.zip).	클라우드 아키텍트
이메일 주소 입력	Amazon SNS 알림을 수신할 활성 이메일 주소를 입력합니다.	클라우드 아키텍트
로깅 수준을 정의합니다.	Lambda 함수의 로깅 수준 및 빈도를 정의합니다. Info는 애플리케이션 진행 상황에 대한	클라우드 아키텍트

작업	설명	필요한 기술
	<p>자세한 정보 메시지를 지정합니다. Error는 애플리케이션을 계속 실행할 수 있게 해주는 오류 이벤트를 지정합니다. Warning은 잠재적으로 유해한 상황을 지정합니다.</p>	
<p>필요한 DynamoDB 태그 키를 입력합니다.</p>	<p>태그는 쉼표로 구분하고 태그 사이에 스페이스가 없어야 합니다 (예: ApplicationId, CreatedBy, Environment, Organization). CloudWatch Events 이벤트는 이러한 태그를 검색하고 찾을 수 없는 경우 알림을 보냅니다.</p>	<p>클라우드 아키텍트</p>

구독을 확인합니다.

작업	설명	필요한 기술
<p>구독을 확인합니다.</p>	<p>템플릿이 성공적으로 배포되면 입력한 이메일 주소로 구독 이메일이 전송됩니다. 위반 알림을 받으려면 이 이메일 구독을 확인해야 합니다.</p>	<p>클라우드 아키텍트</p>

관련 리소스

- [S3 버킷 생성](#)
- [S3 버킷에 파일 업로드](#)
- [DynamoDB에서 리소스 태그 지정](#)
- [AWS를 사용하여 AWS API 호출에서 트리거되는 CloudWatch 이벤트 규칙 생성 CloudTrail](#)

첨부

이 문서와 관련된 추가 콘텐츠에 액세스하려면 [attachment.zip](#) 파일의 압축을 풉니다.

AWS DMS와 Amazon Aurora를 사용하여 지역 간 재해 복구 구현

작성자: Mark Hudson(AWS)

환경: 프로덕션

기술: 데이터베이스

AWS 서비스: AWS DMS,
Amazon RDS, Amazon Aurora

요약

자연재해나 사람에 의한 재해는 언제든지 발생할 수 있으며, 이는 특정 Amazon Web Services(AWS) 리전에서 실행되는 서비스 및 워크로드의 가용성에 영향을 미칠 수 있습니다. 위험을 완화하려면 AWS 서비스에 내장된 지역 간 기능을 통합하는 재해 복구(DR) 계획을 개발해야 합니다. 본질적으로 교차 리전 기능을 제공하지 않는 AWS 서비스의 경우, DR 플랜은 AWS 리전 전반의 장애 조치를 처리할 수 있는 솔루션도 제공해야 합니다.

이 패턴은 단일 지역에 있는 두 개의 Amazon Aurora MySQL Compatible Edition 데이터베이스 클러스터를 포함하는 재해 복구 설정을 안내합니다. DR 요구 사항을 충족하기 위해 데이터베이스 클러스터는 여러 AWS 리전에 걸친 단일 데이터베이스와 함께 Amazon Aurora Global Database 기능을 사용하도록 구성됩니다. AWS Database Migration Service(AWS DMS) 작업은 로컬 리전의 클러스터 간에 데이터를 복제합니다. 하지만 AWS DMS는 현재 지역 간 작업 장애 조치를 지원하지 않습니다. 이 패턴에는 이러한 제한을 해결하고 두 리전에서 AWS DMS를 독립적으로 구성하는 데 필요한 단계가 포함됩니다.

사전 조건 및 제한 사항

사전 조건

- [Amazon Aurora Global Database](#)를 지원하는 일부 기본 및 보조 AWS 리전.
- 기본 리전의 단일 계정에 두 개의 독립적인 Amazon Aurora MySQL Compatible Edition 데이터베이스 클러스터를 사용합니다.
- 데이터베이스 인스턴스 클래스 db.r5 이상(권장).
- 기존 데이터베이스 클러스터 간에 지속적인 복제를 수행하는 기본 리전의 AWS DMS 작업입니다.
- 데이터베이스 인스턴스 생성 요구 사항을 충족할 수 있는 DR 리전 리소스가 마련되어 있습니다. 자세한 내용은 [VPC에서 DB 인스턴스로 작업을 참조하세요](#).

제한 사항

- Amazon Aurora Global Database 제한 사항의 전체 목록은 [Amazon Aurora 글로벌 데이터베이스의 제한 사항](#)을 참조하세요.

제품 버전

- Amazon Aurora MySQL-Compatible Edition 5.7 또는 8.0. 자세한 내용은 [Amazon Aurora 버전](#)을 참조하세요.

아키텍처

대상 기술 스택

- Amazon Aurora MySQL-Compatible Edition 글로벌 데이터베이스 클러스터
- DMS

대상 아키텍처

다음 다이어그램은 두 AWS 지역의 글로벌 데이터베이스를 보여줍니다. 하나는 기본 및 보조 데이터베이스와 AWS DMS 복제가 있고 다른 하나는 보조 기본 및 보조 데이터베이스가 있습니다.

자동화 및 규모 조정

AWS를 CloudFormation 사용하여 가상 사설 클라우드 (VPC), 서브넷, 파라미터 그룹과 같은 필수 인프라를 보조 리전에 생성할 수 있습니다. 또한 AWS를 CloudFormation 사용하여 DR 지역에 보조 클러스터를 생성하고 글로벌 데이터베이스에 추가할 수 있습니다. CloudFormation 템플릿을 사용하여 기본 리전에 데이터베이스 클러스터를 생성한 경우 추가 템플릿으로 업데이트하거나 보강하여 글로벌 데이터베이스 리소스를 생성할 수 있습니다. 자세한 내용은 [두 개의 DB 인스턴스가 있는 Amazon Aurora DB 클러스터 생성](#) 및 [Aurora MySQL용 글로벌 데이터베이스 클러스터 생성](#)을 참조하세요.

마지막으로, 장애 조치 및 장애 복구 이벤트가 발생한 CloudFormation 후를 사용하여 기본 및 보조 지역에서 AWS DMS 작업을 생성할 수 있습니다. 자세한 내용은 [AWS::DMS::ReplicationTask](#)를 참조하십시오.

[AWS::DMS::ReplicationTask](#)

도구

- [Amazon Aurora](#) - Amazon Aurora는 MySQL 및 PostgreSQL과 호환되는 완전 관리형 관계형 데이터베이스 엔진입니다. 이 패턴은 Amazon Aurora MySQL Compatible Edition을 사용합니다.
- [Amazon Aurora Global Database](#) - Amazon Aurora Global Database는 전 세계에 분산된 애플리케이션을 위해 설계되었습니다. Amazon Aurora Global Database의 여러 AWS 리전에 분산될 수 있습니다. 데이터베이스 성능에 영향을 주지 않고 데이터를 복제합니다. 또한 각 지역에서 지연 시간을 줄이면서 로컬 읽기를 빠르게 수행할 수 있으며, 지역 전체의 정전 발생 시 재해 복구 기능을 제공합니다.
- [AWS DMS](#) - AWS Database Migration Service(AWS DMS)는 일회성 마이그레이션 또는 지속적인 복제를 제공합니다. 지속적인 복제 작업을 통해 원본 데이터베이스와 대상 데이터베이스를 동기화할 수 있습니다. 설정이 완료되면 진행 중인 복제 작업은 최소한의 지연 시간으로 소스 변경 사항을 대상에 지속적으로 적용합니다. 데이터 검증 및 변환과 같은 모든 AWS DMS 기능을 모든 복제 작업에 사용할 수 있습니다.

에픽

기본 리전의 기존 데이터베이스 클러스터를 준비합니다.

작업	설명	필요한 기술
데이터베이스 클러스터 파라미터 그룹을 수정합니다.	<p>기존 데이터베이스 클러스터 파라미터 그룹에서 <code>binlog_format</code> 파라미터를 열 값으로 설정하여 행 수준 바이너리 로깅을 활성화합니다.</p> <p>AWS DMS는 지속적인 복제 또는 변경 데이터 캡처(CDC)를 수행할 때 MySQL 호환 데이터베이스에 대한 행 수준 바이너리 로깅을 요구합니다. 자세한 내용은 AWS 관리형 MySQL 호환 데이터베이스를 AWS DMS의 원본으로 사용을 참조하세요.</p>	AWS 관리자

작업	설명	필요한 기술
<p>데이터베이스 바이너리 로그 보존 기간을 업데이트합니다.</p>	<p>최종 사용자 디바이스에 설치된 MySQL 클라이언트 또는 Amazon Elastic Compute Cloud(Amazon EC2) 인스턴스를 사용하여 기본 데이터베이스 클러스터의 작성자 노드에서 Amazon Relational Database Service(RDS)에서 제공하는 다음과 같은 저장 프로시저를 실행합니다. 여기에서 XX란 로그를 보관하는 데 걸리는 시간입니다.</p> <pre data-bbox="597 825 1024 982">call mysql.rds_set_configuration('binlog retention hours', XX)</pre> <p>다음 명령을 실행하면 설정을 확인할 수 있습니다.</p> <pre data-bbox="597 1140 1024 1262">call mysql.rds_show_configuration;</pre> <p>AWS에서 관리하는 MySQL 호환 데이터베이스는 가능한 한 빨리 바이너리 로그를 제거합니다. 따라서 보존 기간은 AWS DMS 작업이 실행되기 전에 로그가 삭제되지 않도록 충분히 길어야 합니다. 일반적으로 24 시간이면 충분하지만 DR 리전에서 AWS DMS 작업을 설정하는 데 필요한 시간을 기준으로 값을 정해야 합니다.</p>	<p>DBA</p>

기본 리전의 기존 AWS DMS 작업 업데이트

작업	설명	필요한 기술
<p>AWS DMS 태스크 ARN을 기록합니다</p>	<p>Amazon 리소스 이름(ARN)을 사용하여 나중에 사용할 수 있도록 AWS DMS 작업 이름을 얻습니다. AWS DMS 작업 ARN을 검색하려면 콘솔에서 작업을 보거나 다음 명령을 실행하세요.</p> <pre data-bbox="594 688 1027 810">aws dms describe-replication-tasks</pre> <p>ARN의 모습은 다음과 같습니다.</p> <pre data-bbox="594 968 1027 1205">arn:aws:dms:us-east-1:<accountid>:task:ANGHFFMPM246XOZVEUHCNSOVF7MQCLTOZUIRAMY</pre> <p>마지막 콜론 뒤의 문자는 이후 단계에서 사용된 작업 이름에 해당합니다.</p>	<p>AWS 관리자</p>
<p>체크포인트를 기록하도록 기존 AWS DMS 작업을 수정합니다.</p>	<p>AWS DMS는 복제 엔진이 변경 스트림의 복구 지점을 알 수 있도록 정보가 포함된 체크포인트를 생성합니다. 체크포인트 정보를 기록하려면 콘솔에서 다음 단계를 수행하세요.</p> <ol style="list-style-type: none"> 1. AWS DMS 작업을 중지합니다. 	<p>AWS 관리자</p>

작업	설명	필요한 기술
<p>체크포인트 정보를 검증합니다.</p>	<p>2. 작업의 JSON 편집기를 사용하여 TaskRecoveryTableEnabled 파라미터를 true로 설정합니다.</p> <p>3. AWS DMS 작업을 시작합니다.</p> <p>클러스터의 작성자 엔드포인트에 연결된 MySQL 클라이언트를 사용하여 리포터 데이터베이스 클러스터에서 새 메타데이터 테이블을 쿼리하여 해당 테이블이 존재하고 복제 상태 정보가 포함되어 있는지 확인합니다. 다음 명령을 실행합니다.</p> <pre data-bbox="594 993 1027 1150">select * from awsdms_control.aws_dms_txn_state;</pre> <p>ARN의 작업 이름은 이 표의 Task_Name 열에 있어야 합니다.</p>	<p>DBA</p>

두 Amazon Aurora 클러스터를 모두 DR 지역으로 확장

작업	설명	필요한 기술
<p>DR 리전에 기본 인프라를 생성합니다.</p>	<p>Amazon Aurora 클러스터를 생성하고 액세스하는 데 필요한 기본 구성 요소를 생성합니다.</p> <ul data-bbox="594 1787 1027 1881" style="list-style-type: none"> • Virtual Private Cloud(VPC) • 서브넷 	<p>AWS 관리자</p>

작업	설명	필요한 기술
	<ul style="list-style-type: none"> • 보안 그룹 • 네트워크 액세스 제어 목록 • 서브넷 그룹 • DB 파라미터 그룹 • DB 클러스터 파라미터 그룹 <p>두 파라미터 그룹의 구성이 기본 지역의 구성과 일치하는지 확인합니다.</p>	
<p>두 Amazon Aurora 클러스터 모두에 DR 지역을 추가합니다.</p>	<p>기본 및 리포터 Amazon Aurora 클러스터에 보조 리전 (DR 리전)을 추가합니다. 자세한 내용은 Amazon Aurora Global Database에 AWS 리전 추가를 참조하세요.</p>	<p>AWS 관리자</p>

장애 조치 수행

작업	설명	필요한 기술
<p>AWS DMS 작업을 중지합니다.</p>	<p>장애 조치가 발생한 후에는 기본 리전의 AWS DMS 작업이 제대로 작동하지 않으므로 오류를 방지하려면 작업을 중지해야 합니다.</p>	<p>AWS 관리자</p>
<p>관리형 장애 조치를 수행하세요.</p>	<p>기본 데이터베이스 클러스터를 DR 지역으로 관리되는 장애 조치를 수행하세요. 자세한 내용은 Amazon Aurora Global Database에 대한 관리형 계획 장애 조치 수행을 참조하세요. 기본 데이터베이스 클러스터에</p>	<p>AWS 관리자, DBA</p>

작업	설명	필요한 기술
	서 장애 조치가 완료된 후 리포터 데이터베이스 클러스터에서 동일한 작업을 수행합니다.	
데이터를 기본 데이터베이스로 로드합니다.	DR 데이터베이스 클러스터에 있는 기본 데이터베이스의 작성자 노드에 테스트 데이터를 삽입합니다. 이 데이터는 복제가 제대로 작동하는지 검증하는 데 사용됩니다.	DBA
AWS DMS 복제 인스턴스를 생성합니다.	DR 리전에서 AWS DMS 복제 인스턴스를 생성하려면 복제 인스턴스 생성 을 참조하세요.	AWS 관리자, DBA
AWS DMS 소스 및 대상 엔드포인트를 생성합니다.	DR 리전에서 AWS DMS 소스 및 대상 엔드포인트를 생성하려면 소스 및 대상 엔드포인트 생성 을 참조하세요. 원본은 기본 데이터베이스 클러스터의 작성자 인스턴스를 가리켜야 합니다. 대상은 보고자 데이터베이스 클러스터의 작성자 인스턴스를 가리켜야 합니다.	AWS 관리자, DBA

작업	설명	필요한 기술
<p>복제 체크포인트를 확보합니다.</p>	<p>복제 체크포인트를 얻으려면 MySQL 클라이언트를 사용하여 DR 지역의 리포터 데이터베이스 클러스터에 있는 작성자 노드에 대해 다음을 실행하여 메타데이터 테이블을 쿼리하세요.</p> <pre data-bbox="597 583 1026 743">select * from awsdms_control.awsdms_txn_state;</pre> <p>표에서 두 번째 에픽에서 획득한 기본 리전에 있는 AWS DMS 작업의 ARN에 해당하는 task_name 값을 찾으세요.</p>	<p>DBA</p>

작업	설명	필요한 기술
AWS DMS 작업을 생성합니다.	<p>콘솔을 사용하여 DR 지역에서 AWS DMS 작업을 생성합니다. 작업에서 데이터 변경 사항만 복제하는 마이그레이션 방법을 지정하세요. 자세한 내용은 작업 생성을 참조하세요.</p> <ol style="list-style-type: none"> 작업 설정에서 마법사를 사용하여 다음 사항을 지정합니다. <ul style="list-style-type: none"> 소스 트랜잭션을 위한 CDC 시작 모드 - 사용자 지정 CDC 시작 모드를 활성화합니다. 소스 트랜잭션을 위한 사용자 지정 CDC 시작점 - 복구 체크포인트를 지정합니다. 복구 체크포인트 상자에 awsdms_txn_state 테이블의 데이터베이스 쿼리를 통해 이전에 가져온 복제 체크포인트 값을 입력합니다. 작업 설정 섹션에서 JSON 편집기를 선택하고 TaskRecoveryTableEnabled 파라미터를 true로 설정합니다. <p>AWS DMS 작업 마이그레이션 작업 시작 설정을 생성 시 자동으로 설정합니다.</p>	AWS 관리자, DBA

작업	설명	필요한 기술
AWS DMS 태스크 ARN을 기록합니다	ARN을 사용하여 나중에 사용할 수 있도록 AWS DMS 작업 이름을 확보합니다. AWS DMS 작업 ARN을 검색하려면 다음 명령을 실행하세요. <pre data-bbox="597 489 1027 611">aws dms describe-replication-tasks</pre>	AWS 관리자, DBA
복제된 데이터를 검증합니다.	DR 리전의 보고서 데이터베이스 클러스터를 쿼리하여 기본 데이터베이스 클러스터에 로드한 테스트 데이터가 복제되었는지 확인합니다.	DBA

파일백 수행

작업	설명	필요한 기술
AWS DMS 작업을 중지합니다.	파일백 발생 후에는 DR 리전의 AWS DMS 작업이 제대로 작동하지 않으므로 오류를 방지하려면 작업을 중지해야 합니다.	AWS 관리자
관리형 파일백을 수행하세요.	기본 데이터베이스 클러스터를 기본 리전으로 파일백합니다. 자세한 내용은 Amazon Aurora Global Database에 대한 관리형 계획 장애 조치 수행 을 참조하세요. 기본 데이터베이스 클러스터의 파일백이 완료된 후 리포터 데이터베이스 클러스터에서 동일한 작업을 수행합니다.	AWS 관리자, DBA

작업	설명	필요한 기술
<p>복제 체크포인트를 확보합니다.</p>	<p>복제 체크포인트를 얻으려면 MySQL 클라이언트를 사용하여 DR 지역의 리포터 데이터베이스 클러스터에 있는 작성자 노드에 대해 다음을 실행하여 메타데이터 테이블을 쿼리하세요.</p> <pre data-bbox="594 583 1029 743">select * from awsdms_control.awsdms_txn_state;</pre> <p>표에서 네 번째 에픽에서 획득한 DR 리전에 있는 AWS DMS 작업의 ARN에 해당하는 <code>task_name</code> 값을 찾으세요.</p>	<p>DBA</p>
<p>AWS DMS 소스 및 대상 엔드포인트를 업데이트합니다.</p>	<p>데이터베이스 클러스터가 페일백된 후에는 기본 지역의 클러스터를 확인하여 어느 노드가 작성기 인스턴스인지 확인합니다. 그런 다음 기본 리전의 기존 AWS DMS 소스 및 대상 엔드포인트가 작성자 인스턴스를 가리키는지 확인합니다. 그렇지 않은 경우 작성자 인스턴스 도메인 이름 시스템(DNS) 이름으로 엔드포인트를 업데이트하세요.</p>	<p>AWS 관리자</p>

작업	설명	필요한 기술
AWS DMS 작업을 생성합니다.	<p>콘솔을 사용하여 기본 리전에 AWS DMS 작업을 생성합니다. 작업에서 데이터 변경 사항만 복제하는 마이그레이션 방법을 지정하세요. 자세한 내용은 작업 생성을 참조하세요.</p> <ol style="list-style-type: none"> 작업 설정에서 마법사를 사용하여 다음 사항을 지정합니다. <ul style="list-style-type: none"> 소스 트랜잭션을 위한 CDC 시작 모드 - 사용자 지정 CDC 시작 모드를 활성화합니다. 소스 트랜잭션을 위한 사용자 지정 CDC 시작점 - 복구 체크포인트를 지정합니다. 복구 체크포인트 상자에 <code>awsdms_txn_state</code> 테이블의 데이터베이스 쿼리를 통해 이전에 가져온 복제 체크포인트 값을 입력합니다. 또한 작업 설정 섹션에서 JSON 편집기를 선택하고 <code>TaskRecoveryTableEnabled</code> 파라미터를 <code>true</code>로 설정합니다. 마지막으로 AWS DMS 작업 마이그레이션 작업 시작 설정을 생성 시 자동으로 설정합니다. 	AWS 관리자, DBA

작업	설명	필요한 기술
<p>AWS DMS 태스크 Amazon 리소스 이름(ARN)을 기록합니다.</p>	<p>ARN을 사용하여 나중에 사용할 수 있도록 AWS DMS 작업 이름을 확보합니다. AWS DMS 작업 ARN을 검색하려면 다음 명령을 실행하세요.</p> <pre data-bbox="597 491 1026 604">aws dms describe-replication-tasks</pre> <p>다른 관리형 장애 조치를 수행할 때 또는 DR 시나리오 중에 작업 이름이 필요합니다.</p>	<p>AWS 관리자, DBA</p>
<p>AWS DMS 작업을 삭제합니다.</p>	<p>기본 리전에서는 기존의 AWS DMS 작업(현재 중지됨)을 삭제하고 보조 리전에서는 기존 AWS DMS 작업(현재 중지됨)을 삭제합니다.</p>	<p>AWS 관리자</p>

관련 리소스

- [Amazon Aurora DB 클러스터 구성](#)
- [Amazon Aurora Global Database 사용](#)
- [Amazon Aurora MySQL을 사용한 작업](#)
- [AWS DMS 복제 인스턴스를 사용한 작업](#)
- [AWS DMS 엔드포인트를 사용한 작업](#)
- [AWS DMS 태스크를 사용한 작업](#)
- [AWS란 CloudFormation 무엇입니까?](#)

추가 정보

Amazon Aurora Global Database는 1초의 Recovery Time Objective(RTO)와 1분 미만의 Recovery Point Objective(RPO)를 제공하여 기존 복제 솔루션보다 낮고 DR 시나리오에 이상적이기 때문에 이 예제에서 DR에 사용됩니다.

Amazon Aurora Global Database는 다음 사항을 비롯한 많은 다른 이점을 제공합니다.

- 로컬 지연 시간을 이용한 글로벌 읽기 - 전 세계 소비자는 로컬 지연 시간을 두고 로컬 지역의 정보에 액세스할 수 있습니다.
- 확장 가능한 보조 Amazon Aurora DB 클러스터 - 보조 클러스터는 독립적으로 확장하여 최대 16개의 읽기 전용 복제본을 추가할 수 있습니다.
- 기본 DB클러스터에서 보조 Amazon Aurora DB 클러스터로의 빠른 복제 - 복제는 기본 클러스터의 성능에 거의 영향을 미치지 않습니다. 이는 스토리지 계층에서 발생하며 일반적인 지역 간 복제 지연 시간은 1초 미만입니다.

이 패턴은 복제에도 AWS DMS를 사용합니다. Amazon Aurora 데이터베이스는 읽기 전용 복제본을 생성하는 기능을 제공하므로 복제 프로세스와 DR 설정을 간소화할 수 있습니다. 그러나 데이터 변환이 필요하거나 대상 데이터베이스에 원본 데이터베이스에 없는 추가 인덱스가 필요한 경우에는 AWS DMS를 복제하는 데 주로 사용됩니다.

100개 이상의 인수가 있는 Oracle 함수 및 프로시저를 PostgreSQL로 마이그레이션

작성자: Srinivas Potlachervoo(AWS)

환경: PoC 또는 파일럿	소스: Oracle	대상: PostgreSQL
R 유형: 리플렛포밍	워크로드: 오픈 소스, Oracle	기술: 데이터베이스, 마이그레이션

AWS 서비스: Amazon RDS,
Amazon Aurora

요약

이 패턴은 100개 이상의 인수가 있는 Oracle Database 함수 및 프로시저를 PostgreSQL로 마이그레이션하는 방법을 보여줍니다. 예를 들어 이 패턴을 사용하여 Oracle 함수 및 프로시저를 다음 PostgreSQL 호환 AWS 데이터베이스 서비스 중 하나로 마이그레이션할 수 있습니다.

- PostgreSQL용 Amazon Relational Database Service(Amazon RDS)
- Amazon Aurora PostgreSQL 호환 에디션

PostgreSQL은 100개 이상의 인수가 있는 함수 또는 프로시저를 지원하지 않습니다. 해결 방법으로 소스 함수의 인수와 일치하는 형식 필드가 있는 새로운 데이터 유형을 정의할 수 있습니다. 그런 다음 사용자 지정 데이터 유형을 인수로 사용하는 PL/pgSQL 함수를 만들고 실행할 수 있습니다.

사전 조건 및 제한 사항

사전 조건

- 활성 상태의 AWS 계정
- [Amazon RDS Oracle 데이터베이스\(DB\) 인스턴스](#)
- [PostgreSQL용 Amazon RDS DB 인스턴스](#) 또는 [Aurora PostgreSQL-Compatible DB 인스턴스](#)

제품 버전

- Amazon RDS Oracle DB 인스턴스 버전 10.2 이상
- Amazon RDS PostgreSQL DB 인스턴스 버전 9.4 이상 또는 Aurora PostgreSQL-Compatible DB 인스턴스 버전 9.4 이상
- Oracle SQL Developer 버전 18 이상
- pgAdmin 버전 4 이상

아키텍처

소스 기술 스택

- Amazon RDS Oracle DB 인스턴스 버전 10.2 이상

대상 기술 스택

- Amazon RDS PostgreSQL DB 인스턴스 버전 9.4 이상 또는 Aurora PostgreSQL-Compatible DB 인스턴스 버전 9.4 이상

도구

서비스

- [PostgreSQL용 Amazon Relational Database Service \(Amazon RDS\)](#)는 AWS Cloud에서 PostgreSQL 관계형 데이터베이스를 설정, 운영 및 조정하는 데 도움이 됩니다.
- [Amazon Aurora PostgreSQL-Compatible Edition](#)은 PostgreSQL 배포를 설정, 운영 및 규모를 조정할 수 있는 완전관리형 ACID 준수 관계형 데이터베이스 엔진입니다.

기타 서비스

- [Oracle SQL Developer](#)는 기존 배포와 클라우드 기반 배포 모두에서 Oracle 데이터베이스의 개발 및 관리를 간소화하는 통합 개발 환경입니다.
- [pgAdmin](#)은 PostgreSQL을 위한 오픈 소스 관리 도구입니다. 데이터베이스 객체를 생성, 유지 관리 및 사용하는 데 도움이 되는 그래픽 인터페이스를 제공합니다.

모범 사례

생성하는 데이터 유형이 소스 Oracle 함수 또는 프로시저에 포함된 유형 필드와 일치하는지 확인합니다.

에픽

100개 이상의 인수가 있는 Oracle 함수 또는 프로시저 실행

작업	설명	필요한 기술
100개 이상의 인수가 있는 기존 Oracle/PLSQL 함수 또는 프로시저를 생성 또는 식별합니다.	<p>100개 이상의 인수가 있는 Oracle/PLSQL 함수 또는 프로시저를 생성합니다.</p> <p>-또는-</p> <p>100개 이상의 인수가 있는 기존 Oracle/PLSQL 함수 또는 프로시저를 식별합니다.</p> <p>자세한 내용은 Oracle Database 설명서의 섹션 14.7 CREATE FUNCTION Statement 및 14.11 CREATE PROCEDURE Statement를 참조하세요.</p>	Oracle/PLSQL 전문 지식
Oracle/PLSQL 함수 또는 프로시저를 컴파일합니다.	<p>Oracle/PLSQL 함수 또는 프로시저를 컴파일합니다.</p> <p>자세한 내용은 Oracle Database 설명서의 함수 컴파일을 참조하세요.</p>	Oracle/PLSQL 전문 지식
Oracle/PLSQL 함수를 실행합니다.	Oracle/PLSQL 함수 또는 프로시저를 실행합니다. 그런 다음 출력을 저장합니다.	Oracle/PLSQL 전문 지식

소스 함수 또는 프로시저의 인수와 일치하는 새로운 데이터 유형 정의

작업	설명	필요한 기술
PostgreSQL에서 새로운 데이터 유형을 정의합니다.	PostgreSQL에서 소스 Oracle 함수 또는 프로시저의 인수에 나타나는 것과 동일한 필드를 모두 포함하는 새로운 데이터 유형을 정의합니다. 자세한 내용은 PostgreSQL 설명서의 CREATE USER 를 참조하세요.	PostgreSQL PL/pgSQL 지식

새로운 TYPE 인수를 포함하는 PostgreSQL 함수 생성

작업	설명	필요한 기술
새로운 데이터 유형을 포함하는 PostgreSQL 함수를 생성합니다.	새로운 TYPE 인수를 포함하는 PostgreSQL 함수를 생성합니다. 예제 함수를 검토하려면 이 패턴의 추가 정보 섹션을 참조하세요.	PostgreSQL PL/pgSQL 지식
PostgreSQL 함수를 컴파일합니다.	PostgreSQL에서 함수를 컴파일합니다. 새로운 데이터 유형 필드가 소스 함수나 프로시저의 인수와 일치하면 함수가 성공적으로 컴파일됩니다.	PostgreSQL PL/pgSQL 지식
PostgreSQL 함수를 실행합니다.	PostgreSQL 함수를 실행합니다.	PostgreSQL PL/pgSQL 지식

문제 해결

문제	Solution
<p>함수는 다음과 같은 오류를 반환합니다.</p> <p>ERROR: "<statement>" 근처의 구문 오류</p>	<p>함수의 명령문 모두가 세미콜론 (;)으로 끝나는지 확인합니다.</p>
<p>함수는 다음과 같은 오류를 반환합니다.</p> <p>ERROR: "<variable>"은 알려진 변수가 아님</p>	<p>함수 본문에 사용된 변수가 함수 DECLARE 섹션 내에 나열되어 있는지 확인합니다.</p>

관련 리소스

- [Amazon Aurora PostgreSQL을 사용한 작업](#) (Aurora용 Amazon Aurora 사용 설명서)
- [CREATE TYPE](#) (PostgreSQL 설명서)

추가 정보

TYPE 인수를 포함하는 PostgreSQL 함수의 예

```
CREATE OR REPLACE FUNCTION test_proc_new
(
    IN p_rec type_test_proc_args
)
RETURNS void
AS
$BODY$
BEGIN

    /*
    *****
    The body would contain code to process the input values.
    For our testing, we will display couple of values.
    *****
    */
    RAISE NOTICE USING MESSAGE = CONCAT_WS(' ', 'p_acct_id: ', p_rec.p_acct_id);
    RAISE NOTICE USING MESSAGE = CONCAT_WS(' ', 'p_ord_id: ', p_rec.p_ord_id);
    RAISE NOTICE USING MESSAGE = CONCAT_WS(' ', 'p_ord_date: ', p_rec.p_ord_date);
```

```
END;  
$BODY$  
LANGUAGE plpgsql  
COST 100;
```

Amazon RDS for Oracle DB 인스턴스를 AMS를 사용하는 다른 계정으로 마이그레이션

작성자: Pinesh Singal(AWS)

환경: PoC 또는 파일럿	소스: 데이터베이스: 관계형	대상: AWS Managed Services의 Amazon RDS for Oracle
R 유형: 리호스팅	워크로드: Oracle	기술: 데이터베이스, 마이그레이션, 스토리지 및 백업
AWS 서비스: Amazon RDS, AWS Managed Services		

요약

이 패턴은 Oracle DB 인스턴스용 Amazon Relational Database Service(RDS)를 한 AWS 계정에서 다른 AWS 계정으로 마이그레이션하는 방법을 보여줍니다. 이 패턴은 원본 AWS 계정은 AWS Managed Services(AMS)를 사용하지 않지만 대상 계정은 AMS를 사용하는 시나리오에 적용됩니다. AWS Management Console을 사용하여 데이터베이스 작업을 수행하는 대신 AMS의 [변경 요청\(RFC\)](#)을 사용하여 마이그레이션을 완료할 수 있습니다. 이 접근 방식은 트랜잭션 수가 많은 테라바이트급 Oracle 소스 데이터베이스의 가동 중지 시간을 최소화합니다. 예를 들어 400~900GB 데이터베이스의 가동 중지 시간은 약 2~3시간 동안 지속될 수 있습니다. 데이터베이스 마이그레이션 시간은 Amazon RDS for Oracle DB 인스턴스의 크기에 정비례합니다.

중요: 이 패턴을 사용하려면 소스 계정에서 Amazon RDS for Oracle DB 인스턴스의 데이터베이스 스냅샷을 만들고, AMS를 사용하는 대상 계정에 이 스냅샷을 복사한 다음 RFC를 통해 해당 스냅샷에서 새 DB 인스턴스를 만들어야 합니다.

사전 조건 및 제한 사항

사전 조건

- 소스 계정으로 활성 AWS 계정
- 대상 계정에 AMS를 사용하는 활성 AWS 계정

- 실행 중인 Amazon RDS for Oracle DB 인스턴스

제한 사항

- 소스 계정의 DB 인스턴스와 동일한 속성 또는 구성이 AMS의 새 대상 DB 인스턴스로 복사됩니다.
- 이 마이그레이션 접근 방식에 사용되는 RFC 방법은 Amazon RDS for Oracle을 지원하는 기능이 제한되어 있습니다. CloudFormation AWS 템플릿을 사용하여 데이터베이스 마이그레이션을 수행하면 Amazon RDS for Oracle의 전체 기능에 액세스할 수 있습니다.
- 예정된 가동 중지 시간 동안 마이그레이션을 완료해야 하기 때문에 몇 시간 동안 애플리케이션 중단이 발생할 수 있습니다. 가동 중지 시간에는 소스 계정의 DB 인스턴스를 중지한 다음 대상 계정의 새 DB 인스턴스를 라이브로 전환하세요.
- 이 마이그레이션 접근 방식은 동일한 AWS 계정 내의 한 AWS 리전에서 다른 리전으로 DB 인스턴스를 마이그레이션하는 경우에는 적용되지 않습니다.

제품 버전

- Amazon RDS for Oracle의 Oracle Database Edition 2(SE2) 12.1.0.2.v2 인스턴스 및 이후 버전
- Amazon RDS for Oracle 11g는 더 이상 지원되지 않습니다(자세한 내용은 Amazon RDS 설명서의 [Amazon RDS for Oracle](#) 참조).

아키텍처

소스 기술 스택

- Amazon RDS for Oracle의 Oracle Database SE2 12.1.0.2.v2 인스턴스
- Amazon RDS 서브넷 그룹
- Amazon RDS 옵션 그룹(필요한 경우)
- Amazon RDS 파라미터 그룹(필요한 경우)
- Amazon Virtual Private Cloud(VPC) 보안 그룹
- AWS 관리형 키 및 고객 관리형 키를 사용하는 AWS Key Management Service(AWS KMS)
- AWS Identity and Access Management(IAM) 역할(필요한 경우)

대상 기술 스택

- Amazon RDS for Oracle의 Oracle Database SE2 12.1.0.2.v2 인스턴스

- Amazon RDS 서브넷 그룹
- Amazon RDS 옵션 그룹(필요한 경우)
- Amazon RDS 파라미터 그룹(필요한 경우)
- Amazon VPC 보안 그룹
- AWS Managed Services(AMS)
- AWS 관리형 키 및 고객 관리형 키를 사용하는 AWS KMS
- IAM 역할(필요한 경우)

소스 및 타겟 마이그레이션 아키텍처

다음 다이어그램은 한 AWS 계정의 Amazon RDS for Oracle DB 인스턴스를 AMS를 사용하는 다른 AWS 계정의 Amazon RDS for Oracle DB 인스턴스로 마이그레이션하는 것을 보여줍니다.

이 다이어그램은 다음 워크플로를 보여줍니다.

1. 소스 계정에서 Amazon RDS for Oracle DB 인스턴스의 데이터베이스 스냅샷을 생성합니다.
2. 스냅샷을 대상 계정의 AMS로 복사합니다.
3. 대상 계정의 스냅샷에서 Amazon RDS for Oracle DB 인스턴스를 새로 생성합니다.

자동화 및 규모 조정

AMS에서 CloudFormation 템플릿을 사용하고 [RFC를 생성하여](#) 마이그레이션을 자동화하고 확장할 수 있습니다. CloudFormation 스냅샷에서 Oracle용 Amazon RDS DB 인스턴스를 생성할 때 DB 인스턴스를 구성하고 복원하는 기능을 포함하여 Oracle용 Amazon RDS의 모든 기능을 사용할 수 있습니다.

도구

- [Amazon Relational Database Service \(Amazon RDS\) for Oracle](#)을 사용하면 AWS 클라우드에서 관계형 데이터베이스를 설정, 운영 및 확장할 수 있습니다.
- [AWS Key Management Service\(AWS KMS\)](#)를 사용하면 암호화 키를 생성하고 제어하여 데이터를 보호할 수 있습니다.
- [AWS Managed Services\(AMS\)](#)는 AWS 인프라를 보다 효율적이고 안전하게 운영하는 데 도움이 됩니다.

에픽

대상 계정의 전환을 준비

작업	설명	필요한 기술
사용자 지정 KMS 키를 생성합니다.	<ol style="list-style-type: none"> KMS 키 생성이라는 자동 RFC를 생성하여 대상 계정에서 사용자 지정 KMS 키를 생성합니다. 사용자 지정 KMS 키를 소스 계정과 공유하세요. 참고: Amazon RDS용 기본 AWS 관리형 키(aws/rds)를 사용하는 Amazon RDS for Oracle DB 인스턴스는 공유할 수 없습니다. 대신 KMS 키에서 DB 인스턴스를 다시 암호화하여 DB 인스턴스를 공유하세요. 	AWS, AMS
보안 그룹을 생성합니다.	<p>보안 그룹 생성이라는 자동 RFC를 생성하여 대상 계정에서 VPC용 보안 그룹을 생성합니다.</p> <p>다음을 지정해야 합니다.</p> <ul style="list-style-type: none"> 새로운 보안 그룹 이름 TCP 및 UDP 수신 및 송신 규칙 표준 태그 	AWS, AMS
(선택 사항) Amazon RDS 리소스를 검토합니다.	Amazon RDS for Oracle DB 인스턴스를 생성할 때 다음 리소스가 생성됩니다.	AWS

작업	설명	필요한 기술
	<ul style="list-style-type: none"> Amazon RDS 서브넷 그룹 (서브넷 ID 기반) Amazon RDS 옵션 그룹(소스 DB 인스턴스의 스냅샷 기반) Amazon RDS 파라미터 그룹(DB 인스턴스의 스냅샷 기반) <p>DB 인스턴스를 만들 때 생성된 Amazon RDS 리소스를 검토하려면 Oracle DB 인스턴스에 연결하여 Amazon RDS 콘솔에서 서브넷 그룹, 옵션 그룹 및 파라미터 그룹을 찾을 수 있습니다.</p>	

소스 계정 전환

작업	설명	필요한 기술
애플리케이션을 중지합니다.	애플리케이션 및 해당 종속 서비스를 중지하세요. 소스 계정의 데이터베이스로 향하는 모든 트래픽을 중지해야 합니다.	앱 소유자
수동 스냅샷을 생성합니다.	소스 계정에서 Amazon RDS for Oracle DB 인스턴스의 DB 스냅샷을 수동으로 생성 합니다.	AWS
DB 인스턴스를 중지합니다.	Amazon RDS for Oracle DB 인스턴스를 중지 합니다.	AWS

작업	설명	필요한 기술
스냅샷을 복사합니다.	동일한 소스 계정에 DB 스냅샷을 복사 한 다음 대상 계정에서 공유한 사용자 지정 KMS 키를 사용하여 복사한 DB 스냅샷 파일을 다시 암호화합니다.	AWS
스냅샷을 공유합니다.	새 스냅샷(사용자 지정 KMS 키로 복사)을 대상 계정과 공유 합니다.	AWS

타겟 계정 전환

작업	설명	필요한 기술
스냅샷을 복사합니다.	<p>RDS 스냅샷 복사라는 자동 RFC를 생성하여 DB 스냅샷을 동일한 대상 계정에 복사하고, 재암호화를 위해 생성된 기본 AWS 관리형 KMS 키를 사용합니다.</p> <p>이는 대상 계정을 새 스냅샷의 소유자로 만들고 필요한 경우 스냅샷에서 생성된 Amazon RDS for Oracle DB 인스턴스를 옵션 그룹과 연결할 수 있도록 하는 데 필요합니다.</p>	AWS, AMS
DB 인스턴스의 스냅샷을 생성합니다.	<p>스냅샷에서 DB 생성이라는 자동 RFC를 생성하여 스냅샷에서 Amazon RDS for Oracle DB 인스턴스를 생성합니다.</p> <p>다음을 지정해야 합니다.</p>	AWS, AMS

작업	설명	필요한 기술
	<ul style="list-style-type: none"> • 이전 단계에서 생성한 새 스냅샷 ID • VPC ID • 서브넷 ID • RDS 인스턴스 ID • 표준 태그 	
<p>인스턴스를 보안 그룹에 연결하고 구성을 업데이트합니다.</p>	<ol style="list-style-type: none"> 1. 기타 업데이트라는 수동 RFC를 생성하여 이전에 생성한 VPC 보안 그룹으로 이전에 생성한 Amazon RDS for Oracle DB 인스턴스를 연결합니다. 2. Amazon RDS for Oracle DB 인스턴스 구성을 추가로 변경합니다. 	<p>AWS, AMS</p>

작업	설명	필요한 기술
DB 인스턴스를 테스트합니다.	<p>동일한 보안 그룹에 호스팅된 인스턴스 또는 애플리케이션 서버에 로그인하고 텔넷을 사용하여 1521 포트에 연결하여 새 Amazon RDS for Oracle DB 인스턴스 엔드포인트 연결을 테스트합니다. 자세한 내용은 Amazon RDS 설명서의 Amazon RDS DB 인스턴스에 연결을 참조하세요.</p> <p>참고: 기본 사용자 로그인 보안 인증 정보를 사용할 수 있는 경우 아무 SQL 클라이언트 (예: Oracle SQL Developer)에 로그인하여 Amazon RDS for Oracle DB 인스턴스를 테스트할 수 있습니다.</p>	AWS, DBA

관련 리소스

- [AWS Managed Services](#)(AWS 설명서)
- [RFC 작동 방식](#)(AWS Managed Services 설명서)
- [암호화된 스냅샷 공유](#)(Amazon RDS 사용 설명서)
- [암호화된 Amazon RDS DB 스냅샷을 다른 계정과 공유하려면?](#) (AWS 지식 센터)
- [Amazon Relational Database Service\(RDS\)란 무엇인가요?](#) (Amazon RDS 사용 설명서)
- [Amazon RDS for Oracle](#)(Amazon RDS 사용 설명서)
- [AMS 콘솔 사용](#)(AWS Managed Services 설명서)

추가 정보

마이그레이션 롤백

마이그레이션을 롤백하려는 경우 다음 단계를 따르세요.

1. 대상 계정에서 수동 RFC(Update Other)를 생성하여 대상 계정에서 생성된 데이터베이스 스택을 삭제합니다.
2. 소스 계정의 Amazon RDS for Oracle DB 인스턴스를 가리키도록 애플리케이션 구성을 업데이트하세요.
3. 소스 계정에서 Amazon RDS for Oracle DB 인스턴스를 시작합니다.

Oracle OUT 바인드 변수를 PostgreSQL 데이터베이스로 마이그레이션

작성자: Bikash Chandra Rout(AWS) 및 Vinay Paladi(AWS)

환경: PoC 또는 파일럿	소스: 관계형 데이터베이스	대상 : RDS/Aurora PostgreSQL
R 유형: 리플랫폼	워크로드: Oracle	기술: 데이터베이스, 마이그레이션
AWS 서비스: Amazon Aurora, Amazon RDS, AWS SCT		

요약

이 패턴은 Oracle Database OUT 바인드 변수를 다음 PostgreSQL 호환 AWS 데이터베이스 서비스 중 하나로 마이그레이션하는 방법을 보여줍니다.

- Amazon Relational Database Service (Amazon RDS) for PostgreSQL
- Amazon Aurora PostgreSQL 호환 에디션

PostgreSQL은 OUT 바인드 변수를 지원하지 않습니다. Python 명령문에서 동일한 기능을 사용하려면 GET 및 SET 패키지 변수를 대신 사용하는 사용자 지정 PL/pgSQL 함수를 만들 수 있습니다. 이 패턴에서 제공되는 예제 래퍼 함수 스크립트는 이러한 변수를 적용하기 위해 [AWS Schema Conversion Tool\(AWS SCT\) 확장팩](#)을 사용합니다.

참고: Oracle EXECUTE IMMEDIATE 명령문이 최대 하나의 행을 반환할 수 있는 SELECT 명령문인 경우 다음을 수행하는 것이 가장 좋습니다.

- INTO 절에 OUT 바인드 변수(정의) 넣기
- USING 절에 IN 바인드 변수 넣기

자세한 내용은 Oracle 설명서의 [EXECUTE IMMEDIATE 명령문](#)을 참조하십시오.

사전 조건 및 제한 사항

사전 조건

- 활성 상태의 AWS 계정.
- 온프레미스 데이터 센터의 Oracle Database 10g (or(또는 그 이상) 소스 데이터베이스
- [PostgreSQL용 Amazon RDS DB 인스턴스](#) 또는 [Aurora PostgreSQL-Compatible DB 인스턴스](#)

아키텍처

소스 기술 스택

- 온프레미스 Oracle Database 10g(또는 그 이상) 데이터베이스

대상 기술 스택

- Amazon RDS for PostgreSQL DB 인스턴스 또는 Aurora PostgreSQL Compatible DB 인스턴스

대상 아키텍처

다음 다이어그램은 Oracle Database OUT 바인드 변수를 PostgreSQL 호환 AWS 데이터베이스로 마이그레이션하기 위한 예제 워크플로를 보여줍니다.

이 다이어그램은 다음 워크플로를 보여줍니다.

1. AWS SCT는 소스 데이터베이스 스키마와 대부분의 사용자 지정 코드를 대상 PostgreSQL 호환 AWS 데이터베이스와 호환되는 형식으로 변환합니다.
2. 자동으로 변환할 수 없는 모든 데이터베이스 객체에는 PL/pgSQL 함수에 의해 플래그가 지정됩니다. 그런 다음 플래그가 지정된 객체를 수동으로 변환하여 마이그레이션을 완료합니다.

도구

- [Amazon Aurora PostgreSQL-Compatible Edition](#)은 PostgreSQL 배포를 설정, 운영 및 규모를 조정할 수 있는 완전관리형의 ACID 준수 관계형 데이터베이스 엔진입니다.

- [Amazon Relational Database Service\(RDS\) for PostgreSQL](#)는 AWS Cloud에서 관계형 데이터베이스를 설정, 운영 및 규모를 조정하는 데 도움이 됩니다.
- [AWS Schema Conversion Tool\(AWS SCT\)](#)은 소스 데이터베이스 스키마와 대부분의 사용자 지정 코드를 대상 데이터베이스와 호환되는 형식으로 자동 변환하여 이기종 데이터베이스 마이그레이션을 지원합니다.
- [pgAdmin](#)은 PostgreSQL을 위한 오픈 소스 관리 도구입니다. 데이터베이스 객체를 생성, 유지 관리 및 사용하는 데 도움이 되는 그래픽 인터페이스를 제공합니다.

에픽

사용자 지정 PL/PgSQL 함수 및 AWS SCT를 사용하여 Oracle OUT 바인드 변수를 마이그레이션합니다.

작업	설명	필요한 기술
PostgreSQL과 호환되는 AWS 데이터베이스에 연결합니다.	DB 인스턴스를 생성한 후에는 표준 SQL 클라이언트 애플리케이션을 사용하여 DB 클러스터의 데이터베이스에 연결할 수 있습니다. 예를 들어 pgAdmin 을 사용하여 DB 인스턴스에 연결할 수 있습니다. 자세한 내용은 다음 중 하나를 참조하십시오. <ul style="list-style-type: none"> • Amazon RDS 사용 설명서의 Amazon RDS DB 인스턴스에 연결 • Amazon Aurora 사용 설명서의 Amazon Aurora DB 클러스터에 연결 	마이그레이션 엔지니어
이 패턴의 예제 래퍼 함수 스크립트를 대상 데이터베이스의 기본 스키마에 추가합니다.	이 패턴의 추가 정보 섹션에서 예제 PL/pgSQL 래퍼 함수 스크립트를 복사하십시오. 그런 다음 대상 데이터베이스의 기	마이그레이션 엔지니어

작업	설명	필요한 기술
<p>(선택 사항) Test_pg 스키마를 포함하도록 대상 데이터베이스의 기본 스키마에서 검색 경로를 업데이트합니다.</p>	<p>본 스키마에 함수를 추가합니다.</p> <p>자세한 내용은 PostgreSQL 설명서에서 CREATE FUNCTION을 참조하십시오.</p> <p>성능을 향상시키기 위해 PostgreSQL search_path 변수를 업데이트하여 Test_pg 스키마 이름을 포함하도록 할 수 있습니다. 검색 경로에 스키마 이름을 포함하면 PL/pgSQL 함수를 직접적으로 호출할 때마다 이름을 지정하지 않아도 됩니다.</p> <p>자세한 내용은 PostgreSQL 설명서의 섹션 5.9.3 스키마 검색 경로를 참조하십시오.</p>	<p>마이그레이션 엔지니어</p>

관련 리소스

- [AWS Schema Conversion Tool](#)
- [OUT 바인드 변수](#)(Oracle 설명서)
- [바인드 변수를 사용하여 SQL 쿼리 성능 향상](#) (Oracle 블로그)

추가 정보

PL/PgSQL 함수 예제

```

/* Oracle */

CREATE or replace PROCEDURE test_pg.calc_stats_new1 (
                                a NUMBER,

```

```
        b NUMBER,  
        result out NUMBER  
    )  
  
IS  
BEGIN  
result:=a+b;  
END;  
/  
/* Testing */  
set serveroutput on  
DECLARE  
    a NUMBER := 4;  
    b NUMBER := 7;  
    plsql_block VARCHAR2(100);  
    output number;  
BEGIN  
    plsql_block := 'BEGIN test_pg.calc_stats_new1(:a, :b,:output); END;';  
    EXECUTE IMMEDIATE plsql_block USING a, b,out output; -- calc_stats(a, a, b, a)  
    DBMS_OUTPUT.PUT_LINE('output:'||output);  
END;  
  
output:11  
  
PL/SQL procedure successfully completed.  
  
--Postgres--  
  
/* Example : 1 */  
CREATE OR REPLACE FUNCTION test_pg.calc_stats_new1(  
    w integer,  
    x integer  
    )  
  
RETURNS integer  
AS  
$BODY$  
begin  
    return w + x ;  
end;  
$BODY$  
LANGUAGE plpgsql;
```

```

CREATE OR REPLACE FUNCTION aws_oracle_ext.set_package_variable(
    package_name name,
    variable_name name,
    variable_value
    anyelement
)
    RETURNS void
    LANGUAGE 'plpgsql'

    COST 100
    VOLATILE
AS $BODY$
begin
    perform set_config
        ( format( '%s.%s',package_name, variable_name )
        , variable_value::text
        , false );
end;
$BODY$;

CREATE OR REPLACE FUNCTION aws_oracle_ext.get_package_variable_record(
    package_name
    name,
    record_name name
)
    RETURNS text
    LANGUAGE 'plpgsql'
    COST 100
    VOLATILE
AS $BODY$
begin
    execute 'select ' || package_name || '$Init()';

    return aws_oracle_ext.get_package_variable
        (
            package_name := package_name
            , variable_name := record_name || '$REC' );
end;
$BODY$;

--init()--
CREATE OR REPLACE FUNCTION test_pg.init()
    RETURNS void
AS

```

```

$BODY$
BEGIN
if aws_oracle_ext.is_package_initialized('test_pg' ) then
    return;
end if;
perform aws_oracle_ext.set_package_initialized
    ('test_pg' );
PERFORM aws_oracle_ext.set_package_variable('test_pg', 'v_output', NULL::INTEGER);
PERFORM aws_oracle_ext.set_package_variable('test_pg', 'v_status', NULL::text);
END;
$BODY$
LANGUAGE plpgsql;

/* callable for 1st Example */

DO $$
declare
v_sql text;
v_output_loc int;
a integer :=1;
b integer :=2;
BEGIN
perform test_pg.init();
--raise notice 'v_sql %',v_sql;
execute 'do $$ declare v_output_l int; begin select * from test_pg.calc_stats_new1('||
a||','||b||') into v_output_l;
PERFORM aws_oracle_ext.set_package_variable('test_pg', 'v_output', v_output_l) ;
end; $$' ;
v_output_loc := aws_oracle_ext.get_package_variable('test_pg', 'v_output');
raise notice 'v_output_loc %',v_output_loc;
END ;
$$

/*In above Postgres example we have set the value of v_output using v_output_l in the
dynamic anonymous block to mimic the
behaviour of oracle out-bind variable .*/

--Postgres Example : 2 --
CREATE OR REPLACE FUNCTION test_pg.calc_stats_new2(
w integer,
x integer,
inout status text,
out result integer)
AS

```

```
$BODY$
DECLARE
begin
result := w + x ;
status := 'ok';
end;
$BODY$
LANGUAGE plpgsql;

/* callable for 2nd Example */
DO $$
declare
v_sql text;
v_output_loc int;
v_staus text:= 'no';
a integer :=1;
b integer :=2;
BEGIN
perform test_pg.init();
execute 'do $a$ declare v_output_l int; v_status_l text; begin select * from
test_pg.calc_stats_new2('||a||','||b||','||v_staus||') into v_status_l,v_output_l;
PERFORM aws_oracle_ext.set_package_variable('test_pg', 'v_output', v_output_l) ;
PERFORM aws_oracle_ext.set_package_variable('test_pg', 'v_status', v_status_l) ;
end; $a$' ;
v_output_loc := aws_oracle_ext.get_package_variable('test_pg', 'v_output');
v_staus := aws_oracle_ext.get_package_variable('test_pg', 'v_status');
raise notice 'v_output_loc %',v_output_loc;
raise notice 'v_staus %',v_staus;
END ;
$$
```

동일한 호스트 이름을 가진 SAP HSR을 사용하여 SAP HANA를 AWS로 마이그레이션

작성자: Pradeep Puliampatta(AWS)

환경: 프로덕션	소스: SAP HANA DB 온프레미스	대상: AWS 기반 SAP HANA DB
R 타입: 리호스팅	워크로드: SAP	기술 : 데이터베이스, 마이그레이션
AWS 서비스: AWS Client VPN, AWS Direct Connect, Amazon EBS		

요약

백업 및 복원, 내보내기 및 가져오기, SAP HANA 시스템 복제(HSR)를 비롯한 다양한 옵션을 사용하여 Amazon Web Services(AWS)로 SAP HANA를 마이그레이션할 수 있습니다. 특정 옵션의 선택은 소스 및 대상 SAP HANA 데이터베이스 간의 네트워크 연결, 소스 데이터베이스의 크기, 다운타임 고려 사항 및 기타 요인에 따라 달라집니다.

SAP HANA 워크로드를 AWS로 마이그레이션하기 위한 SAP HSR 옵션은 소스 시스템과 대상 시스템 사이에 안정적인 네트워크가 있고 SAP HSR의 네트워크 처리량 요구 사항에 대해 SAP에서 규정한 대로 전체 데이터베이스(SAP HANA DB 복제 스냅샷)를 1일 이내에 완전히 복제할 수 있을 때 잘 작동합니다. 이 접근 방식의 다운타임 요구 사항은 대상 AWS 환경에서 테이크오버, SAP HANA DB 백업 및 마이그레이션 후 작업을 수행하는 것으로 제한됩니다.

SAP HSR은 기본 또는 소스 시스템과 보조 또는 대상 시스템 간의 복제 트래픽에 대해 서로 다른 호스트 이름(다른 IP 주소에 매핑된 호스트 이름)을 사용할 수 있도록 지원합니다. `global.ini`의 `[system_replication_hostname_resolution]` 섹션에서 특정 호스트 이름 세트를 정의하여 이 작업을 수행할 수 있습니다. 이 섹션에서는 기본 및 보조 사이트의 모든 호스트를 각 호스트에 정의해야 합니다. 자세한 구성 단계는 [SAP 설명서](#)를 참조하세요.

이 설정에서 얻을 수 있는 한 가지 중요한 점은 기본 시스템의 호스트 이름이 보조 시스템의 호스트 이름과 달라야 한다는 것입니다. 그렇지 않으면 다음과 같은 오류가 발생할 수 있습니다.

- "each site must have a unique set of logical hostnames"
- "remoteHost does not match with any host of the source site. All hosts of source and target site must be able to resolve all hostnames of both sites correctly"

그러나 대상 AWS 환경에서 동일한 SAP HANA DB 호스트 이름을 사용하면 마이그레이션 후 단계 수를 줄일 수 있습니다.

이 패턴은 SAP HSR 옵션을 사용할 때 원본 및 대상 환경에서 동일한 호스트 이름을 사용하기 위한 해결 방법을 제공합니다. 이 패턴을 사용하면 SAP HANA 호스트 이름 바꾸기 옵션을 사용할 수 있습니다. 대상 SAP HANA DB에 임시 호스트 이름을 할당하여 SAP HSR의 호스트 이름 고유성을 용이하게 합니다. 대상 SAP HANA 환경의 테이크오버 마일스톤이 마이그레이션을 완료한 후 대상 시스템 호스트 이름을 소스 시스템의 호스트 이름으로 되돌릴 수 있습니다.

사전 조건 및 제한 사항

사전 조건

- 활성 상태의 AWS 계정
- 가상 프라이빗 네트워크(VPN) 엔드포인트 또는 라우터가 있는 Virtual Private Cloud(VPC)입니다.
- 소스에서 타겟으로 파일을 전송하도록 구성된 AWS Client VPN 또는 AWS Direct Connect입니다.
- 소스 및 대상 환경 모두의 SAP HANA 데이터베이스. 대상 SAP HANA DB 패치 수준은 동일한 SAP HANA 플랫폼 에디션 내에서 소스 SAP HANA DB 패치 수준과 같거나 그 이상이어야 합니다. 예를 들어 HANA 1.0과 HANA 2.0 시스템 간에는 복제를 설정할 수 없습니다. 자세한 내용은 SAP 노트: 1999880 - FAQ: SAP HANA 시스템 복제의 질문 15를 참조하세요.
- 대상 환경의 SAP 애플리케이션 서버.
- 대상 환경의 Amazon Elastic Block Store(Amazon EBS) 볼륨.

제한 사항

다음 SAP 문서 목록은 SAP HANA 동적 계층화 및 스케일 아웃 마이그레이션과 관련된 제약 조건을 포함하여 이 해결 방법과 관련된 알려진 문제를 다룹니다.

- 2956397 - SAP HANA 데이터베이스 시스템의 이름을 바꾸지 못했습니다
- 2222694 - HANA 시스템의 이름을 바꾸려고 할 때 다음과 같은 오류가 나타납니다. "원본 sidadm 사용자가 소스 파일을 소유하지 않았습니다(uid = xxxx)"

- 2607227 - hdblocm: register_rename_system: SAP HANA 인스턴스 이름 변경 실패
- 2630562 - HANA 호스트 이름 변경에 실패하여 HANA가 시작되지 않음
- 2935639 - sr_register가 global.ini 섹션의 system_replication_hostname_resolution에 지정된 호스트 이름을 사용하지 않습니다
- 2710211 - 오류: 소스 시스템과 타겟 시스템의 논리적 호스트 이름이 겹칩니다
- 2693441 - 오류로 인해 SAP HANA 시스템의 이름을 바꾸지 못했습니다
- 2519672 - HANA 기본 및 보조 시스템의 PKI, SSFS 데이터 및 키가 다르거나 확인할 수 없습니다
- 2457129 - 동적 계층화가 랜드스케이프의 일부인 경우 SAP HANA 시스템 호스트 이름 변경이 허용되지 않습니다
- 2473002 - HANA 시스템 복제를 사용하여 스케일 아웃 시스템을 마이그레이션합니다(스케일 아웃 SAP HANA 시스템에 이 호스트 이름 변경 접근 방식을 사용할 때는 SAP에서 제공하는 제한이 없습니다. 하지만 각 개별 호스트에서 이 절차를 반복해야 합니다. 이 접근 방식에는 다른 스케일 아웃 마이그레이션 제한 사항도 적용됩니다.)

제품 버전

- 이 솔루션은 SAP HANA DB 플랫폼 에디션 1.0 및 2.0에 적용됩니다.

아키텍처

소스 설정

SAP HANA 데이터베이스는 소스 환경에 설치됩니다. 모든 SAP 애플리케이션 서버 연결 및 DB 인터페이스는 클라이언트 연결에 동일한 호스트 이름을 사용합니다. 다음 다이어그램은 예제 소스 호스트인 hdbhost와 해당 IP 주소를 보여줍니다.

타겟 설정

AWS 클라우드 대상 환경은 동일한 호스트 이름을 사용하여 SAP HANA 데이터베이스를 실행합니다. AWS의 대상 환경에는 다음 사항이 포함됩니다.

- SAP HANA 데이터베이스
- SAP 애플리케이션 서버
- EBS 볼륨

중간 구성

다음 다이어그램에서는 원본 및 대상의 호스트 이름이 고유하도록 AWS 대상 환경의 호스트 이름이 일시적으로 temp-host로 변경되었습니다. 대상 환경의 테이크오버 마일스톤이 마이그레이션을 완료하면 대상 시스템 가상 호스트 이름이 원래 이름인 hdbhost를 사용하여 변경됩니다.

중간 구성에는 다음 옵션 중 하나가 포함됩니다.

- Client VPN 엔드포인트가 있는 AWS Client VPN
- 라우터에 연결하는 AWS Direct Connect

AWS 대상 환경의 SAP 애플리케이션 서버는 복제 설정 전 또는 인수 후에 설치할 수 있습니다. 그러나 복제 설정 전에 애플리케이션 서버를 설치하면 설치 중 가동 중지 시간을 줄이고 고가용성을 구성하고 백업하는 데 도움이 될 수 있습니다.

도구

서비스

- [AWS Client VPN](#) – AWS Client VPN은 AWS 리소스 및 온프레미스 네트워크의 AWS 리소스에 안전하게 액세스할 수 있도록 하는 관리형 클라이언트 기반 VPN 서비스입니다.
- [AWS Direct Connect](#) - AWS Direct Connect는 표준 이더넷 광섬유 케이블을 통해 내부 네트워크를 AWS Direct Connect 위치에 연결합니다. 이 연결을 통해 네트워크 경로의 인터넷 서비스 공급자를 우회하여 퍼블릭 AWS 서비스에 직접 가상 인터페이스를 생성할 수 있습니다.
- [Amazon EBS](#) – Amazon Elastic Block Store(Amazon EBS)는 EC2 인스턴스에 사용할 블록 스토리지 볼륨을 제공합니다. EBS 볼륨은 형식이 지정되지 않은 원시 블록 디바이스처럼 동작합니다. 이러한 볼륨을 인스턴스에 디바이스로 마운트할 수 있습니다.

기타 도구

- [SAP 애플리케이션 서버](#) - SAP 애플리케이션 서버는 프로그래머에게 비즈니스 로직을 표현할 수 있는 방법을 제공합니다. SAP 애플리케이션 서버는 비즈니스 로직을 기반으로 데이터 처리를 수행합니다. 실제 데이터는 별도의 구성 요소인 데이터베이스에 저장됩니다.

- [SAP HANA 콕핏](#) 및 [SAP HANA 스튜디오](#) - SAP HANA 콕핏과 SAP HANA 스튜디오 모두 SAP HANA 데이터베이스에 대한 관리 인터페이스를 제공합니다. SAP HANA 스튜디오에서 SAP HANA 관리 콘솔은 SAP HANA 데이터베이스 관리와 관련된 콘텐츠를 제공하는 시스템 뷰입니다.
- [SAP HANA 시스템 복제](#) - SAP HANA 시스템 복제(SAP HSR)는 SAP HANA 데이터베이스를 복제하기 위해 SAP에서 제공하는 표준 절차입니다. SAP HSR에 필요한 실행 파일은 SAP HANA 서버 커널 자체의 일부입니다.

모범 사례

< 작성자: 참고 사항 삭제: 사용자가 이 패턴을 보다 효과적으로 구현하는 데 도움이 될 수 있는 지침 및 권장 사항 목록을 제공하십시오. >

에픽

원본 및 대상 환경 준비

작업	설명	필요한 기술
SAP HANA 데이터베이스를 설치하고 구성합니다.	소스 및 대상 환경에서 AWS의 SAP HANA 모범 사례에 따라 SAP HANA DB를 설치하고 구성해야 합니다. 자세한 내용은 AWS의 SAP HANA 를 참조하세요.	SAP Basis 관리
IP 주소를 매핑합니다.	대상 환경에서 임시 호스트 이름이 내부 IP 주소에 할당되었는지 확인합니다. 1. EC2, 인스턴스, 작업, 네트워킹, IP 주소 관리, 새 IP 주소 할당으로 이동하여 AWS Management Console의 Amazon Elastic Compute Cloud(Amazon EC2) 인스턴스에 보조 IPv4 주소를 할당합니다.	AWS 관리

작업	설명	필요한 기술
	<p>2. 운영 체제에서 루트 사용자로 EC2 네트워크 어댑터 (NIC)에 동일한 주소를 할당하려면 <code>ip addr add <IP>/32 dev eth0</code> 명령을 실행하고 <IP>를 1단계의 IP 주소로 대체하세요.</p>	
<p>대상 호스트 이름을 확인합니다.</p>	<p>보조 SAP HANA DB에서 /etc/hosts 파일의 관련 호스트 이름을 업데이트하여 SAP HANA 복제 네트워크의 호스트 이름(hdbhost 및 temp-host)이 모두 확인되었는지 확인합니다.</p>	<p>Linux 관리</p>
<p>원본 및 대상 SAP HANA 데이터베이스를 백업합니다.</p>	<p>SAP HANA 스튜디오 또는 SAP HANA 콘솔을 사용하여 SAP HANA 데이터베이스에서 백업을 수행할 수 있습니다.</p>	<p>SAP Basis 관리</p>
<p>교환 시스템 PKI 인증서.</p>	<p>(SAP HANA 2.0 이상에만 적용) 기본 데이터베이스와 보조 데이터베이스 간 파일 시스템(SSFS) 저장소의 시스템 공개 키 인프라(PKI) 보안 저장소에 있는 인증서를 교환합니다. 자세한 내용은 SAP 참고 2369981 - SAP HANA 시스템 복제를 사용한 인증에 필요한 구성 단계를 참조하세요.</p>	<p>SAP Basis 관리</p>

대상 SAP HANA DB의 이름을 변경합니다.

작업	설명	필요한 기술
<p>대상 클라이언트 연결을 중지합니다.</p>	<p>대상 환경에서 SAP 애플리케이션 서버 및 기타 클라이언트 연결을 종료합니다.</p>	<p>SAP Basis 관리</p>
<p>대상 SAP HANA DB의 이름을 임시 호스트 이름으로 변경합니다.</p>	<ol style="list-style-type: none"> 루트 사용자인 경우 레지던트 hdb1cm을 사용하여 대상 SAP HANA DB 호스트 이름을 임시 호스트 이름으로 변경합니다. <div data-bbox="630 772 1029 926" style="border: 1px solid #ccc; border-radius: 10px; padding: 10px; margin: 10px 0;"> <pre>root \$> cd /hana/shared/<SID/hdb1cm root \$> ./hdb1cm</pre> </div> <ol style="list-style-type: none"> 9 rename_system Rename the SAP HANA Database System 옵션을 선택합니다. 새 이름 (temp-host)을 입력합니다. 필요에 따라 다른 옵션을 검증할 수 있습니다. 그러나 호스트 이름 변경을 SID 변경과 혼동하지 않도록 주의하세요(SAP 참고 2598814 - hdb1cm: SID 이름 바꾸기 실패). <p>hdb1cm에서 SAP HANA DB 중지 및 시작을 제어합니다.</p>	<p>SAP Basis 관리</p>
<p>복제 네트워크를 할당합니다.</p>	<p>소스 시스템의 global.in i 파일에서 [system_r</p>	<p>SAP Basis 관리</p>

작업	설명	필요한 기술
	<p>eplication_hostname_resolution] 헤더 아래에 소스 및 타겟 복제 네트워크 세부 정보를 제공합니다. 그런 다음 대상 시스템의 global.ini 파일에 항목을 복사합니다.</p>	
<p>기본에서 복제를 활성화합니다.</p>	<p>원본 SAP HANA DB에서 복제를 활성화하려면 다음 명령을 실행하세요.</p> <pre data-bbox="597 747 1026 863">hdbnsutil -sr_enable --name=siteA</pre>	<p>SAP Basis 관리</p>
<p>대상 SAP HANA DB를 보조 시스템으로 등록합니다.</p>	<p>대상 SAP HANA DB를 SAP HSR 소스의 보조 시스템으로 등록하려면 비동기 복제를 선택합니다.</p> <pre data-bbox="597 1121 1026 1556">(sid)adm \$> HDB stop (sid)adm \$> hdbnsutil -sr_register -name=siteB -remotehost=hdbhost / --remoteInstance=00 -replicationMode=async -operationMode=log replay (sid)adm \$> HDB start</pre> <p>또는 등록하기 위해 -online 옵션을 선택해도 됩니다. 이 경우 SAP HANA DB를 중지하고 시작할 필요가 없습니다.</p>	<p>SAP Basis 관리</p>

작업	설명	필요한 기술
동기화를 검증합니다.	<p>(비동기식 복제이므로) 소스 SAP HANA DB에서 모든 로그가 대상 시스템에 적용되었는지 확인합니다.</p> <p>복제를 확인하려면 소스에서 다음 명령을 실행합니다.</p> <pre data-bbox="597 569 1027 768">(sid)adm \$> cdp (sidadm \$> python systemReplicationS tatus.py</pre>	SAP Basis 관리
소스 SAP 애플리케이션과 SAP HANA DB를 종료합니다.	마이그레이션 전환 중에 소스 시스템(SAP 애플리케이션 및 SAP HANA 데이터베이스)을 종료합니다.	SAP Basis 관리
타겟에서 테이크오버를 수행하세요.	AWS의 타겟에서 테이크오버를 수행하려면 <code>hdbnsutil -sr_takeover</code> 명령을 실행하세요.	SAP Basis 관리
대상 SAP HANA DB에서 복제를 해제합니다.	<p>복제 메타데이터를 지우려면 <code>hdbnsutil -sr_disable</code> 명령을 실행하여 대상 시스템에서 복제를 중지하세요.</p> <p>참고: 이는 SAP 노트 2693441에 따른 것입니다. 오류로 인해 SAP HANA 시스템의 이름을 바꾸지 못했습니다.</p>	SAP Basis 관리
대상 SAP HANA DB를 백업합니다.	인계가 성공하면 전체 SAP HANA DB 백업을 수행하는 것이 좋습니다.	SAP Basis 관리

대상 시스템의 원래 호스트 이름으로 되돌립니다.

작업	설명	필요한 기술
<p>대상 SAP HANA DB 호스트 이름을 원래대로 되돌립니다.</p>	<ol style="list-style-type: none"> 대상 SAP HANA DB 호스트 이름을 원래 가상 호스트 이름으로 되돌리려면 레지던트 hdblcm을 사용하세요. <div style="border: 1px solid #ccc; border-radius: 10px; padding: 10px; margin: 10px 0;"> <pre>root \$> cd /hana/shared/<SID>/hdblcm root \$> ./hdblcm</pre> </div> 9 rename_system Rename the SAP HANA Database System 옵션을 선택합니다. 새 이름 (hdbhost)을 입력합니다. <p>필요에 따라 다른 옵션을 검증할 수 있습니다. 그러나 호스트 이름 변경을 SID 변경과 혼동하지 않도록 주의하세요(SAP 참고 2598814 - hdblcm: SID 이름 바꾸기 실패).</p>	<p>SAP Basis 관리</p>
<p>hdbuserstore를 조정합니다.</p>	<p>소스 hdbuserstore 세부 정보를 가리키는 schema/user 세부 정보를 조정하세요. 자세한 단계는 SAP 설명서를 참조하세요.</p> <p>이 단계를 확인하려면 R3trans -d 명령을 실행합니다. 결과는 SAP HANA 데이터</p>	<p>SAP Basis 관리</p>

작업	설명	필요한 기술
	베이스에 성공적으로 연결되었음을 반영해야 합니다.	
클라이언트 연결을 시작합니다.	대상 환경에서 SAP 애플리케이션 서버 및 기타 클라이언트 연결을 시작합니다.	SAP Basis 관리

관련 리소스

SAP 참조

SAP 설명서 참조는 SAP에서 자주 업데이트합니다. 최신 정보를 확인하려면 SAP Note 2407186 - SAP HANA 고가용성을 위한 사용 방법 가이드 및 백서를 참조하세요.

추가 SAP 참고 사항

- 2550327 - SAP HANA 시스템의 이름을 바꾸는 방법
- 1999880 - 자주 묻는 질문: SAP HANA 시스템 복제
- 2078425 - SAP HANA 플랫폼 수명 주기 관리 도구 hdblcm에 대한 문제 해결 노트
- 2592227 - HANA 시스템의 FQDN 접미사 변경
- 2048681 - SSH 또는 루트 보안 인증 없이 다중 호스트 시스템에서 SAP HANA 플랫폼 수명 주기 관리 작업 수행

SAP 설명서

- [시스템 복제 네트워크 연결](#)
- [시스템 복제를 위한 호스트 이름 확인](#)

AWS 참조

- [다른 플랫폼에서 AWS로 SAP HANA 마이그레이션](#)

추가 정보

호스트 이름 변경 활동의 일환으로 hdb1cm에서 수행한 변경 사항은 다음 세부 로그에 통합되어 있습니다.

분산된 가용성 그룹을 사용하여 SQL Server를 AWS로 마이그레이션

작성자: Praveen Marthala(AWS)

소스: SQL 서버 온프레미스	대상: EC2 기반 SQL 서버	R 타입: 리호스팅
환경: PoC 또는 파일럿	기술: 데이터베이스, 마이그레이션	워크로드: Microsoft
AWS 서비스: Amazon EC2		

요약

Microsoft SQL Server 올웨이즈 온 가용성 그룹은 SQL Server를 위한고가용성(HA) 및 재해 복구(DR) 솔루션을 제공합니다. 가용성 그룹은 읽기/쓰기 트래픽을 허용하는 기본 복제본과 읽기 트래픽을 허용하는 최대 8개의 보조 복제본으로 구성됩니다. 가용성 그룹은 두 개 이상의 노드가 있는 Windows Server 장애 조치 클러스터(WAFC)에 구성됩니다.

Microsoft SQL Server 올웨이즈 온 분산 가용성 그룹은 두 개의 독립적인 WAFC 간에 두 개의 개별 가용성 그룹을 구성하는 솔루션을 제공합니다. 분산 가용성 그룹에 속하는 가용성 그룹은 같은 데이터 센터에 있지 않아도 됩니다. 가용성 그룹 하나는 온프레미스에 있을 수 있고, 다른 가용성 그룹은 다른 도메인의 Amazon Elastic Compute Cloud(Amazon EC2) 인스턴스에 있는 Amazon Web Services(AWS) 클라우드에 있을 수 있습니다.

이 패턴은 분산 가용성 그룹을 사용하여 Amazon EC2에 가용성 그룹을 설정한 상태에서 기존 가용성 그룹에 속하는 온프레미스 SQL Server 데이터베이스를 SQL Server로 마이그레이션하는 단계를 설명합니다. 이 패턴을 따르면 전환 중에 가동 중지 시간을 최소화하면서 데이터베이스를 AWS 클라우드로 마이그레이션할 수 있습니다. 데이터베이스는 전환 후 즉시 AWS에서고가용성을 제공합니다. 또한 이 패턴을 사용하여 SQL 서버의 동일한 버전을 유지하면서 기본 운영 체제를 온프레미스에서 AWS로 변경할 수 있습니다.

사전 조건 및 제한 사항

사전 조건

- 활성 상태의 AWS 계정

- AWS Direct Connect 또는 AWS Site-to-Site VPN
- 온프레미스와 AWS의 두 노드에 설치된 동일한 버전의 SQL 서버

제품 버전

- SQL Server 버전 2016 이상
- SQL Server Enterprise Edition

아키텍처

소스 기술 스택

- 온프레미스 상시 가동 가용성 그룹을 지원하는 Microsoft SQL Server 데이터베이스

대상 기술 스택

- AWS 클라우드의 Amazon EC2에 올웨이즈 온 가용성 그룹이 있는 Microsoft SQL Server 데이터베이스

마이그레이션 아키텍처

용어

- WSFC 1 - WSFC 온프레미스
- WSFC 2 - AWS 클라우드의 WSFC
- AG 1 - WSFC 1에 속하는 첫 번째 가용성 그룹
- AG 2 - WSFC 2에 속하는 두 번째 가용성 그룹
- SQL Server 기본 복제본 - 모든 쓰기의 글로벌 기본 복제본으로 간주되는 AG 1의 노드
- SQL Server 전달자 - SQL Server 기본 복제본에서 비동기적으로 데이터를 수신하는 AG 2의 노드
- SQL Server 보조 복제본 - 기본 복제본 또는 전달자로부터 데이터를 동기적으로 수신하는 AG 1 또는 AG 2의 노드

도구

- [AWS Direct Connect](#)는 표준 이더넷 광섬유 케이블을 통해 내부 네트워크를 AWS Direct Connect에 연결합니다. 이 연결을 통해 네트워크 경로의 인터넷 서비스 공급자를 우회하여 퍼블릭 AWS 서비스에 직접 가상 인터페이스를 생성할 수 있습니다.
- [Amazon EC2](#) – Amazon Elastic Compute Cloud(Amazon EC2)는 AWS 클라우드에서 확장 가능한 컴퓨팅 용량을 제공합니다. Amazon EC2를 사용하여 필요에 따라 많거나 적은 수의 가상 서버를 시작하고 스케일 아웃 또는 스케일 인할 수 있습니다.
- [AWS 사이트 간 VPN — AWS 사이트 간 VPN](#)은 가상 사설망 (VPN) 생성을 지원합니다. site-to-site AWS에서 시작하는 인스턴스와 자체 원격 네트워크 간에 트래픽을 전달하도록 VPN을 구성할 수 있습니다.
- [마이크로소프트 SQL 서버 관리 스튜디오](#) - 마이크로소프트 SQL 서버 관리 스튜디오(SSMS)는 SQL 서버 인프라를 관리하기 위한 통합 환경입니다. SQL Server와 상호 작용하는 다양한 스크립트 편집기와 함께 사용자 인터페이스와 도구 그룹을 제공합니다.

에픽

AWS에 두 번째 가용성 그룹 설정

작업	설명	필요한 기술
AWS에서 WSFC를 생성합니다.	HA용 노드 2개가 있는 Amazon EC2 인스턴스에서 WSFC 2를 생성합니다. 이 장애 조치 클러스터를 사용하여 AWS에 두 번째 가용성 그룹 (AG 2)을 생성합니다.	시스템 관리자, 관리자 SysOps
WSFC 2에 두 번째 가용성 그룹을 생성합니다.	SSMS를 사용하여 WSFC 2의 두 노드에 AG 2를 생성합니다. WSFC 2의 첫 번째 노드가 전달자 역할을 합니다. WSFC 2의 두 번째 노드는 AG 2의 보조 복제본 역할을 합니다.	DBA, 개발자

작업	설명	필요한 기술
	<p>이 단계에서는 AG 2에서 사용할 수 있는 데이터베이스가 없습니다. 분산 가용성 그룹을 설정하기 위한 출발점입니다.</p>	
<p>AG 2에서 복구 옵션 없이 데이터베이스를 생성합니다.</p>	<p>온프레미스 가용성 그룹(AG 1)에 데이터베이스를 백업합니다.</p> <p>복구 옵션 없이 데이터베이스를 전달자와 AG 2의 보조 복제본 모두에 복원합니다. 데이터베이스를 복원하는 동안 데이터베이스 데이터 파일과 로그 파일을 위한 충분한 디스크 공간이 있는 위치를 지정하세요.</p> <p>이 단계에서는 데이터베이스가 복원 상태에 있습니다. AG 2 또는 분산 가용성 그룹에 속하지 않으며 동기화되지도 않습니다.</p>	<p>DBA, 개발자</p>

분산 가용성 그룹 구성

작업	설명	필요한 기술
<p>AG 1에 분산 가용성 그룹을 생성합니다.</p>	<p>AG 1에 분산 가용성 그룹을 생성하려면 DISTRIBUTED 옵션이 있는 CREATE AVAILABILITY GROUP 을 사용하세요.</p> <p>1. AG 1 및 AG 2의 LISTENER_URL 엔드포인트 주소를 사용합니다.</p>	<p>DBA, 개발자</p>

작업	설명	필요한 기술
	<ol style="list-style-type: none"> 2. AVAILABILITY-MODE 의 경우 네트워크 지연을 방지하기 위해 ASYNCHRONOUS_COMMIT 을 사용하세요. 이는 데이터베이스 성능에 영향을 주지 않습니다. 3. FAILOVER_MODE 에는 MANUAL을 사용합니다. 분산 가용성 그룹에서 작동하는 유일한 가용성 모드입니다. 4. AG 2에서 데이터베이스를 수동으로 복원하고 대규모 데이터베이스를 더 잘 제어하려면 SEEDING_MODE 에 대한 MANUAL을 사용하세요. 	

작업	설명	필요한 기술
<p>AG 2에 분산 가용성 그룹을 생성합니다.</p>	<p>AG 2에 분산 가용성 그룹을 생성하려면 DISTRIBUTED 옵션이 있는 ALTER AVAILABILITY GROUP 을 사용하세요.</p> <ol style="list-style-type: none"> AG 1 및 AG 2의 LISTENER_URL 엔드포인트 주소를 사용합니다. AVAILABILITY-MODE 의 경우 네트워크 지연을 방지하기 위해 ASYNCHRONOUS_COMMIT 을 사용하세요. 이는 데이터베이스 성능에 영향을 주지 않습니다. FAILOVER_MODE 에는 MANUAL을 사용합니다. 분산 가용성 그룹에서 작동하는 유일한 가용성 모드입니다. AG 2에서 데이터베이스를 수동으로 복원하고 대규모 데이터베이스를 더 잘 제어하려면 SEEDING_MODE 에 대한 MANUAL을 사용하세요. <p>분산 가용성 그룹은 AG 1과 AG 2 사이에 생성됩니다.</p> <p>AG 2의 데이터베이스는 아직 AG 1에서 AG 2로의 데이터 흐름에 참여하도록 구성되지 않았습니다.</p>	<p>DBA, 개발자</p>

작업	설명	필요한 기술
<p>AG 2의 전달자 및 보조 복제본에 데이터베이스를 추가합니다.</p>	<p>AG 2의 ALTER DATABASE 전달자와 보조 복제본 모두에서 SET HADR AVAILABILITY GROUP 옵션을 사용하여 데이터베이스를 분산 가용성 그룹에 추가합니다.</p> <p>그러면 AG 1과 AG 2의 데이터베이스 간에 비동기 데이터 흐름이 시작됩니다.</p> <p>글로벌 기본 복제본은 쓰기 작업을 수행하고 AG 1의 보조 복제본에 동기적으로 데이터를 전송하고 AG 2의 전달자에 비동기적으로 데이터를 전송합니다. AG 2의 전달자는 AG 2의 보조 복제본에 동기적으로 데이터를 전송합니다.</p>	<p>DBA, 개발자</p>

AG 1과 AG 2 간의 비동기 데이터 흐름을 모니터링합니다.

작업	설명	필요한 기술
<p>DMV 및 SQL 서버 로그를 사용합니다.</p>	<p>동적 관리 보기(DMV)와 SQL Server 로그를 사용하여 두 가용성 그룹 간의 데이터 흐름 상태를 모니터링합니다.</p> <p>모니터링 대상 DMV에는 sys.dm_hadr_availability_replica_states 및 sys.dm_hadr_automatic_seeding 이 포함됩니다.</p>	<p>DBA, 개발자</p>

작업	설명	필요한 기술
	전달자 동기화 상태를 확인하려면 전달자의 SQL Server 로그에서 동기화 상태를 모니터링하세요.	

최종 마이그레이션을 위한 전환 활동 수행

작업	설명	필요한 기술
기본 복제본에 대한 모든 트래픽을 중지합니다.	AG 1의 기본 복제본으로 들어오는 트래픽을 중지하여 데이터베이스에서 쓰기 작업이 발생하지 않고 데이터베이스를 마이그레이션할 준비가 되도록 하세요.	앱 소유자, 개발자
AG 1의 분산 가용성 그룹의 가용성 모드를 변경합니다.	기본 복제본에서 분산 가용성 그룹의 가용성 모드를 동기모드로 설정합니다. 가용성 모드를 동기모드로 변경하면 데이터가 AG 1의 기본 복제본에서 AG 2의 전달자로 동기적으로 전송됩니다.	DBA, 개발자
두 가용성 그룹의 LSN을 확인하세요.	AG 1과 AG 2 모두에서 마지막 로그 시퀀스 번호(LSN)를 확인합니다. AG 1의 기본 복제본에서는 쓰기가 수행되지 않으므로 데이터가 동기화되고 두 가용성 그룹의 마지막 LSN이 일치해야 합니다.	DBA, 개발자
AG 1을 보조 역할로 업데이트합니다.	AG 1을 보조 역할로 업데이트하면 AG 1은 기본 복제 역할을	DBA, 개발자

작업	설명	필요한 기술
	읽고 쓰기를 허용하지 않으며 두 가용성 그룹 간의 데이터 흐름이 중지됩니다.	

두 번째 가용성 그룹으로 페일오버

작업	설명	필요한 기술
AG 2로 수동 페일오버합니다.	<p>AG 2의 전달자에서 데이터 손실을 허용하도록 분산 가용성 그룹을 변경하세요. AG 1과 AG 2의 마지막 LSN이 일치하는지 이미 확인하고 확인했으므로 데이터 손실은 걱정할 필요가 없습니다.</p> <p>AG 2에서 전달자의 데이터 손실을 허용하면 AG 1과 AG 2의 역할이 변경됩니다.</p> <ul style="list-style-type: none"> AG 2는 기본 복제본과 보조 복제본이 있는 가용성 그룹이 됩니다. AG 1은 전달자 및 보조 복제본이 있는 가용성 그룹이 됩니다. 	DBA, 개발자
AG 2의 분산 가용성 그룹의 가용성 모드를 변경합니다.	<p>AG 2의 기본 복제본에서 가용성 모드를 비동기로 변경합니다.</p> <p>이렇게 하면 데이터 이동이 AG 2에서 AG 1로, 즉 동기식에서 비동기식으로 변경됩니다. 이 단계는 AG 2와 AG 1 사이의 네</p>	DBA, 개발자

작업	설명	필요한 기술
	트위크 지연 시간(있는 경우)을 방지하기 위해 필요하며 데이터베이스 성능에는 영향을 미치지 않습니다.	
새 기본 복제본으로 트래픽 전송을 시작합니다.	<p>AG 2의 리스너 URL 엔드포인트를 사용하여 데이터베이스로 트래픽을 전송하도록 연결 문자열을 업데이트합니다.</p> <p>이제 AG 2는 쓰기를 수락하고 AG 1의 전달자에게 데이터를 전송하며, AG 2의 자체 보조 복제본에도 데이터를 전송합니다. 데이터는 AG 2에서 AG 1로 비동기적으로 이동합니다.</p>	앱 소유자, 개발자

전환 이후 활동 수행

작업	설명	필요한 기술
분산 가용성 그룹을 AG 2에 삭제합니다.	<p>계획된 시간에 마이그레이션을 모니터링합니다. 그런 다음 AG 2에 분산 가용성 그룹을 삭제하여 AG 2와 AG 1 사이의 분산 가용성 그룹 설정을 제거합니다. 이렇게 하면 분산 가용성 그룹 구성이 제거되고 AG 2에서 AG 1로 이동하는 데이터 흐름이 중지됩니다.</p> <p>현재 AG 2는 쓰기 작업을 수행하는 기본 복제본과 동일한 가용성 그룹 내에 보조 복제본이</p>	DBA, 개발자

작업	설명	필요한 기술
	있어 AWS에서 가용성이 높습니다.	
온프레미스 서버를 서비스 해제합니다.	AG 1에 속하는 WSFC 1의 온프레미스 서버를 서비스 해제합니다.	시스템 관리자, SysOps 관리자

관련 리소스

- [분산 가용성 그룹](#)
- [SQL 문서: 분산 가용성 그룹](#)
- [SQL Docs: Always On 가용성 그룹: 고가용성 및 재해 복구 솔루션](#)

AWS DMS를 사용하여 오라클 8i 또는 9i에서 오라클용 아마존 RDS로 마이그레이션할 수 있습니다. SharePlex

작성자: Ramu Jagini(AWS)

환경: PoC 또는 파일럿	소스: 데이터베이스: 관계형	대상: Amazon RDS
R 유형: 리플랫폼	워크로드: 오픈 소스, Oracle	기술: 데이터베이스, 클라우드 네이티브, 마이그레이션
AWS 서비스: AWS DMS, Amazon RDS		

요약

이 패턴은 온프레미스 Oracle 8i 또는 9i 데이터베이스를 Oracle 데이터베이스용 Amazon Relational Database Service(RDS)로 마이그레이션하는 방법을 설명합니다. 이 패턴을 사용하면 동기식 복제를 위한 Quest를 사용하여 다운타임을 줄이고 마이그레이션을 완료할 수 있습니다. SharePlex

AWS Database Migration Service(AWS DMS)는 Oracle 8i 또는 9i를 소스 환경으로 지원하지 않으므로 마이그레이션에 중간 Oracle 데이터베이스 인스턴스를 사용해야 합니다. [SharePlex 7.6.3](#)을 사용하여 이전 Oracle 데이터베이스 버전에서 이후 Oracle 데이터베이스 버전으로 복제할 수 있습니다. 중간 Oracle 데이터베이스 인스턴스는 SharePlex 7.6.3의 대상으로 호환되며 AWS DMS 또는 최신 릴리스의 소스로 지원됩니다. SharePlex 이 지원을 통해 Amazon RDS for Oracle 대상 환경으로 데이터를 추후에 복제할 수 있습니다.

사용 중단된 여러 데이터 유형 및 기능이 Oracle 8i 또는 9i에서 최신 버전의 Oracle 데이터베이스로의 마이그레이션에 영향을 미칠 수 있다는 점을 고려하세요. 이 패턴은 이러한 영향을 완화하기 위해 Oracle 11.2.0.4를 중간 데이터베이스 버전으로 사용하여 Amazon RDS for Oracle 대상 환경으로 마이그레이션하기 전에 스키마 코드를 최적화하도록 지원합니다.

사전 조건 및 제한 사항

사전 조건

- 활성 상태의 AWS 계정
- 온프레미스 환경의 소스 Oracle 8i 또는 9i 데이터베이스

- Amazon Elastic Compute Cloud(Amazon EC2)에서 스테이지하기 위한 [Oracle Database 12c 릴리스 2\(12CR2\)](#)
- 퀘스트 SharePlex 7.6.3 (상용 등급)

제한 사항

- [RDS for Oracle 제한 사항](#)

제품 버전

- 소스 데이터베이스: Oracle 8i 또는 9i
- 스테이징 데이터베이스용 Oracle 12CR2(Amazon RDS for Oracle 버전과 일치해야 함)
- 대상 데이터베이스의 경우 Oracle 12CR2 이상(Amazon RDS for Oracle)

아키텍처

소스 기술 스택

- Oracle 8i 또는 9i 데이터베이스
- SharePlex

대상 기술 스택

- Amazon RDS for Oracle

마이그레이션 아키텍처

다음 다이어그램은 온프레미스 환경에서 AWS 클라우드의 Amazon RDS for Oracle DB 인스턴스로 Oracle 8i 또는 9i 데이터베이스를 마이그레이션하는 방법을 보여줍니다.

이 다이어그램은 다음 워크플로를 보여줍니다.

1. 아카이브 로그 모드, 강제 로깅 및 추가 로깅으로 Oracle 원본 데이터베이스를 활성화합니다.
2. [복구 관리자 \(RMAN\) point-in-time 복구 및 FLASHBACK_SCN을 사용하여 오라클 소스 데이터베이스에서 Oracle 스테이징 데이터베이스를 복원합니다.](#)

3. (RMAN에서 사용) 를 사용하여 Oracle 소스 데이터베이스에서 리두 로그를 SharePlex 읽도록 구성합니다. FLASHBACK_SCN
4. SharePlex 복제를 시작하여 Oracle 소스 데이터베이스의 데이터를 Oracle 스테이징 데이터베이스와 동기화합니다.
5. FLASHBACK_SCN과 함께 EXPDP 및 IMPDP를 사용하여 Amazon RDS for Oracle 대상 데이터베이스를 복원합니다.
6. (EXPDP에서 사용한) FLASHBACK_SCN을 사용하여 AWS DMS와 해당 소스 작업을 Oracle 스테이징 데이터베이스로 구성하고, Amazon RDS for Oracle을 대상 데이터베이스로 구성합니다.
7. AWS DMS 작업을 시작하여 Oracle 스테이징 데이터베이스의 데이터를 Oracle 대상 데이터베이스로 동기화합니다.

도구

- [Amazon Relational Database Service\(RDS\)](#)는 AWS 클라우드에서 관계형 데이터베이스(DB)를 설정, 운영 및 조정하는 데 도움이 됩니다.
- [AWS Database Migration Service\(AWS DMS\)](#)를 사용하면 데이터 스토어를 AWS 클라우드로 마이그레이션하거나 클라우드와 온프레미스 설정 조합 간에 데이터 스토어를 마이그레이션할 수 있습니다.
- [SharePlexQuest](#)는 다운타임을 최소화하고 데이터 손실 없이 데이터를 이동할 수 있는 Oracle 간 데이터 복제 도구입니다.
- [복구 관리자\(RMAN\)](#)는 데이터베이스에서 백업 및 복구 작업을 수행하는 Oracle Database 클라이언트입니다. 데이터베이스 파일의 백업, 복원 및 복구를 대폭 단순화합니다.
- [데이터 펌프 내보내기](#)를 이용하면 데이터와 메타데이터를 덤프 파일 모음이라는 운영 체제 파일 모음에 업로드할 수 있습니다. 덤프 파일 세트는 [데이터 펌프 가져오기](#) 유틸리티 또는 [DBMS_DATAPUMP](#) 패키지를 통해서만 가져올 수 있습니다.

에픽

Amazon SharePlex EC2에 Oracle 스테이징 데이터베이스를 설정하고 설치합니다.

작업	설명	필요한 기술
EC2 인스턴스를 생성합니다.	1. EC2 인스턴스를 생성합니다.	Oracle 관리

작업	설명	필요한 기술
	<p>2. Oracle 12CR2를 EC2 인스턴스에 설치하여 Oracle 스테이징 데이터베이스로 사용하세요.</p>	
<p>스테이징 데이터베이스를 준비합니다.</p>	<p>Oracle 8i 또는 9i 데이터베이스 소스 환경에서 RMAN 백업을 가져와 Oracle 12CR2 업그레이드로 복원할 수 있도록 Oracle 스테이징 데이터베이스를 준비합니다.</p> <p>자세한 내용은 Oracle 설명서의 Oracle 9i 복구 관리자 사용 설명서 및 데이터베이스 백업 및 복구 사용 설명서를 참조하세요.</p>	<p>Oracle 관리</p>
<p>구성하십시오. SharePlex</p>	<p>SharePlex 원본을 온프레미스 Oracle 8i 또는 9i 데이터베이스로 구성하고, 대상을 Amazon EC2에서 호스팅되는 Oracle 12CR2 스테이징 데이터베이스로 구성합니다.</p>	<p>SharePlex, 오라클 관리</p>

Amazon RDS for Oracle을 대상 환경으로 설정합니다.

작업	설명	필요한 기술
<p>Oracle DB 인스턴스를 생성합니다.</p>	<p>Amazon RDS for Oracle 데이터베이스를 생성한 다음 Oracle 12CR2를 데이터베이스에 연결합니다.</p> <p>자세한 내용은 Amazon RDS 설명서의 Oracle DB 인스턴스</p>	<p>DBA</p>

작업	설명	필요한 기술
	생성 및 Oracle DB 인스턴스의 데이터베이스에 연결을 참조하세요.	
스태이징 데이터베이스에서 Amazon RDS for Oracle을 복원합니다.	<ol style="list-style-type: none"> FLASHBACK_SCN 을 사용하여 Oracle 스테이징 데이터베이스 서버에서 EXPDP 백업을 생성합니다. 스태이징 데이터베이스에서 Amazon RDS for Oracle을 복원합니다. <p>자세한 내용은 Oracle 설명서의 54 DBMS_DATAPUMP를 참조하세요.</p>	DBA

AWS DMS 설정

작업	설명	필요한 기술
데이터베이스의 엔드포인트를 생성합니다.	<p>Oracle 스테이징 데이터베이스의 소스 엔드포인트와 Amazon RDS for Oracle 데이터베이스의 대상 엔드포인트를 생성합니다.</p> <p>자세한 내용은 AWS 지식 센터에서 AWS DMS를 사용하여 소스 또는 대상 엔드포인트를 생성하려면 어떻게 해야 하나요?를 참조하세요.</p>	DBA
복제 인스턴스를 생성합니다.	AWS DMS를 사용하여 Oracle 스테이징 데이터베이스의 복제 인스턴스를 Amazon RDS for	DBA

작업	설명	필요한 기술
	<p>Oracle 데이터베이스로 시작합니다.</p> <p>자세한 내용은 AWS 지식 센터에서 AWS DMS 복제 인스턴스를 생성하려면 어떻게 해야 하나요?를 참조하세요.</p>	
복제 태스크를 생성하고 시작합니다.	<p>(EXPDP를 통해 전체 로드가 이미 발생했으므로) EXPDP에서 FLASHBACK_SCN 을 사용하여 변경 데이터 캡처 (CDC) 를 위한 AWS DMS 복제 태스크를 생성합니다.</p> <p>태스크 생성에 대한 자세한 내용은 AWS DMS 설명서를 참조하세요.</p>	DBA

Amazon RDS for Oracle로 전환

작업	설명	필요한 기술
애플리케이션 워크로드를 중지하세요.	계획된 전환 기간 동안 애플리케이션 서버와 해당 애플리케이션을 중지하세요.	앱 개발자, DBA
온프레미스 Oracle 스테이징 데이터베이스와 EC2 인스턴스의 동기화를 확인합니다.	온프레미스 소스 데이터베이스에서 몇 번의 로그 전환을 수행하여 SharePlex 복제 인스턴스에서 Amazon EC2의 Oracle 스테이징 데이터베이스로 복제 작업에 대한 모든 메시지가 게시되었는지 확인합니다.	DBA

작업	설명	필요한 기술
	자세한 내용은 Oracle 설명서의 6.4.2 로그 파일 전환 을 참조하세요.	
Oracle 스테이징 데이터베이스와 Amazon RDS for Oracle 데이터베이스의 동기화를 확인합니다.	모든 AWS DMS 태스크에 지연 및 오류가 없는지 확인한 다음 태스크의 검증 상태를 확인합니다.	DBA
SharePlex 및 Amazon RDS의 복제를 중지하십시오.	AWS DMS SharePlex 복제와 AWS DMS 복제 모두에서 오류가 표시되지 않는 경우 두 복제를 모두 중지하십시오.	DBA
Amazon RDS에 애플리케이션을 다시 매핑합니다.	Amazon RDS for Oracle 엔드포인트 세부 정보를 애플리케이션 서버 및 해당 애플리케이션과 공유한 다음 애플리케이션을 시작하여 비즈니스 운영을 재개합니다.	앱 개발자, DBA

AWS 대상 환경 테스트

작업	설명	필요한 기술
AWS에서 Oracle 스테이징 데이터베이스 환경을 테스트하세요.	<ol style="list-style-type: none"> SharePlex 복제를 테스트하여 Oracle 스테이징 데이터베이스에 동기화 간격이나 복제 오류가 없는지 확인합니다. 온프레미스 환경에 정의된 벤치마크를 통해 애플리케이션이 예상대로 작동하는지 확인합니다. 	SharePlex, 오라클 관리

작업	설명	필요한 기술
아마존 RDS 환경을 테스트합니다.	<ol style="list-style-type: none"> 복제 후 Amazon RDS로 전파된 모든 데이터에 오류가 없는지 확인하세요. 다른 애플리케이션이 Amazon RDS DB 인스턴스를 가리키도록 한 다음에 성능 테스트를 수행하여 예상 동작을 확인합니다. <p>자세한 내용은 Amazon RDS 설명서의 Amazon RDS for Oracle에서 참조하세요.</p>	Oracle 관리

관련 리소스

- [확신을 갖고 마이그레이션하세요.](#)
- [Amazon EC2](#)
- [Amazon RDS for Oracle](#)
- [AWS Database Migration Service](#)
- [AWS DMS 마이그레이션 디버깅: 문제가 발생했을 때 대처 방법\(1부\)](#)
- [AWS DMS 마이그레이션 디버깅: 문제가 발생했을 때 대처 방법\(2부\)](#)
- [AWS DMS 마이그레이션 디버깅: 문제가 발생했을 때 대처 방법 \(3부\)](#)
- [SharePlex 데이터베이스 복제용](#)
- [SharePlex: 모든 환경을 위한 데이터베이스 복제](#)

암호화를 사용하지 않는 인스턴스가 있는지 Amazon Aurora를 모니터링

작성자: Mansi Suratwala(AWS)

환경: 프로덕션

기술: 보안, 자격 증명, 규정 준수, 스토리지 및 백업, 데이터베이스

워크로드: 오픈 소스, 기타 모든 워크로드

AWS 서비스: 아마존 SNS, 아마존 오로라, AWS, CloudTrail
| 아마존 CloudWatch, AWS Lambda

요약

이 패턴은 암호화를 켜지 않고 Amazon Aurora 인스턴스가 생성될 때 자동 알림을 설정하도록 배포할 수 있는 Amazon Web Services (AWS) CloudFormation 템플릿을 제공합니다.

Aurora는 MySQL 및 PostgreSQL과 호환되는 완전 관리형 관계형 데이터베이스 엔진입니다. 일부 워크로드의 경우 Aurora은 기존 애플리케이션을 거의 변경하지 않고도 MySQL의 처리량을 최대 5배, PostgreSQL의 처리량을 최대 3배 제공할 수 있습니다.

CloudFormation 템플릿은 Amazon CloudWatch 이벤트 이벤트와 AWS Lambda 함수를 생성합니다. 이 이벤트는 AWS를 CloudTrail 사용하여 Aurora 인스턴스 생성 또는 기존 인스턴스의 특정 시점 복원을 모니터링합니다. Cloudwatch Events 이벤트는 암호화의 활성화 여부를 확인하는 Lambda 함수를 시작합니다. 암호화가 켜져 있지 않은 경우, Lambda 함수는 Amazon Simple Notification Service(SNS) 알림을 전송하여 위반 사실을 알립니다.

사전 조건 및 제한 사항

필수 조건

- 활성 상태의 AWS 계정.

제한 사항

- 이 서비스 제어는 Amazon Aurora 인스턴스에서만 작동합니다. 다른 Amazon Relational Database Service(RDS) 인스턴스는 지원하지 않습니다.
- CloudFormation 템플릿은 RestoreDBClusterToPointInTime용으로만 **CreateDBInstance** 배포해야 합니다.

제품 버전

- Amazon Aurora에서 지원되는 PostgreSQL 버전
- Amazon Aurora에서 지원되는 MySQL 버전

아키텍처

대상 기술 스택

- Amazon Aurora
- AWS CloudTrail
- 아마존 CloudWatch
- AWS Lambda
- Amazon Simple Storage Service (S3)
- Amazon SNS

대상 아키텍처

자동화 및 규모 조정

여러 지역 및 계정에 대해 CloudFormation 템플릿을 여러 번 사용할 수 있습니다. 각 리전 또는 계정에 한 번만 실행해야 합니다.

도구

도구

- [Amazon Aurora](#) – Amazon Aurora는 MySQL 및 PostgreSQL과 호환되는 완전 관리형 관계형 데이터베이스 엔진입니다.

- [AWS CloudTrail](#) — AWS는 AWS 계정의 거버넌스, 규정 준수, 운영 및 위험 감사를 관리할 수 있도록 CloudTrail 지원합니다. 사용자, 역할 또는 AWS 서비스가 수행한 작업은 에서 이벤트로 기록됩니다 CloudTrail.
- [Amazon CloudWatch Events](#) — Amazon CloudWatch Events는 AWS 리소스의 변경 사항을 설명하는 시스템 이벤트 near-real-time 스트림을 제공합니다.
- [AWS Lambda](#) – AWS Lambda는 서버를 프로비저닝하거나 관리하지 않고도 코드를 실행하도록 지원하는 컴퓨팅 서비스입니다. Lambda는 필요 시에만 코드를 실행하며, 일일 몇 개의 요청에서 초당 수천 개의 요청까지 자동으로 규모를 조정합니다.
- [Amazon S3](#) – Amazon Simple Storage Service(S3)는 웹 사이트, 모바일 애플리케이션, 백업 및 데이터 레이크를 포함하여 다양한 스토리지 솔루션에 사용할 수 있는 확장성이 뛰어난 객체 스토리지 서비스입니다.
- [Amazon SNS](#)-Amazon Simple Notification Service(SNS)은 Lambda, HTTP, 이메일, 모바일 푸시 알림 및 모바일 문자 메시지(SMS)를 사용하여 메시지를 전송하는 관리형 서비스입니다.

코드

프로젝트의 .zip 파일은 첨부 파일로 제공됩니다.

에픽

Lambda 스크립트용 S3 버킷을 생성

작업	설명	필요한 기술
S3 버킷을 정의합니다.	Amazon S3 콘솔을 열고 S3 버킷을 선택하거나 생성합니다. 이 S3 버킷은 Lambda 코드 .zip 파일을 호스팅합니다. S3 버킷은 Aurora와 같은 리전에 있어야 합니다. S3 버킷 이름에는 선행 슬래시를 포함할 수 없습니다.	클라우드 아키텍트

Lambda 코드를 S3 버킷에 업로드

작업	설명	필요한 기술
Lambda 코드를 업로드합니다.	첨부 파일 섹션에 제공된 Lambda 코드 .zip 파일을 사용 자가 정의한 S3 버킷에 업로드 합니다.	클라우드 아키텍트

CloudFormation 템플릿을 배포하십시오.

작업	설명	필요한 기술
CloudFormation 템플릿을 배포 합니다.	CloudFormation 콘솔에서 이 패턴의 첨부 파일로 제공 된 RDS_Aurora_Encryption_At_Rest.yml CloudFormation 템플릿을 배포 합니다. 다음 에픽에서 템플릿 파라미터에 대한 값을 입력합니다.	클라우드 아키텍트

CloudFormation 템플릿의 파라미터를 완성하세요.

작업	설명	필요한 기술
S3 버킷 이름을 제공합니다.	첫 번째 에픽에서 생성하거나 선택한 S3 버킷의 이름을 입력 합니다.	클라우드 아키텍트
S3 키를 입력합니다.	S3 버킷의 Lambda 코드 .zip 파일 위치를 선행 슬래시 없이 입력합니다(예: <directory>/<file-name>.zip).	클라우드 아키텍트

작업	설명	필요한 기술
이메일 주소를 입력합니다.	Amazon SNS 알림을 수신할 활성 이메일 주소를 입력합니다.	클라우드 아키텍트
로깅 수준을 정의합니다.	Lambda 함수의 로깅 수준 및 빈도를 정의합니다. Info는 애플리케이션 진행 상황에 대한 자세한 정보 메시지를 지정합니다. Error는 애플리케이션을 계속 실행할 수 있게 해주는 오류 이벤트를 지정합니다. Warning은 잠재적으로 유해한 상황을 지정합니다.	클라우드 아키텍트

구독 확인

작업	설명	필요한 기술
구독을 확인합니다.	템플릿이 성공적으로 배포되면 입력한 이메일 주소로 구독 이메일 메시지가 전송됩니다. 알림을 받기 시작하려면 이 이메일 구독을 확인해야 합니다.	클라우드 아키텍트

관련 리소스

- [S3 버킷 생성](#)
- [S3 버킷에 파일 업로드](#)
- [Amazon Aurora DB 클러스터 생성](#)
- [AWS를 사용하여 AWS API 호출에서 트리거되는 CloudWatch 이벤트 규칙 생성 CloudTrail](#)

첨부

이 문서와 관련된 추가 콘텐츠에 액세스하려면 [attachment.zip](#) 파일의 압축을 풉니다.

Amazon을 사용하여 오라클 GoldenGate 로그를 모니터링하십시오. CloudWatch

작성자: Chithra Krishnamurthy(AWS)

환경: 프로덕션

기술: 데이터베이스

워크로드: Oracle

AWS 서비스: 아마존
CloudWatch, 아마존 SNS

요약

GoldenGate 오라클은 오라클 데이터베이스의 경우 아마존 관계형 데이터베이스 서비스 (Amazon RDS) 간 또는 아마존 Elastic Compute Cloud (Amazon EC2) 에 호스팅되는 오라클 데이터베이스 간에 실시간 복제를 제공합니다. 단방향 복제와 양방향 복제를 모두 지원합니다.

복제에 사용하는 GoldenGate 경우, 모니터링을 통해 GoldenGate 프로세스가 가동 및 실행 중인지 확인하고 소스 데이터베이스와 대상 데이터베이스가 동기화되어 있는지 확인하는 것이 중요합니다.

이 패턴은 GoldenGate 오류 로그에 대한 Amazon CloudWatch 모니터링을 구현하는 단계와 복제를 신속하게 재개하기 위한 적절한 조치를 취할 수 ABEND 있도록 특정 이벤트 (예: STOP 또는) 에 대한 알림을 보내도록 경보를 설정하는 방법을 설명합니다.

사전 조건 및 제한 사항

사전 조건

- GoldenGate EC2 인스턴스에 설치 및 구성되어 있으므로 해당 EC2 인스턴스에 대한 CloudWatch 모니터링을 설정할 수 있습니다. 양방향 복제를 위해 AWS 지역 GoldenGate 전체를 모니터링하려면 GoldenGate 프로세스가 실행되는 각 EC2 인스턴스에 CloudWatch 에이전트를 설치해야 합니다.

제한 사항

- 이 패턴은 를 사용하여 GoldenGate 프로세스를 모니터링하는 방법을 설명합니다. CloudWatch CloudWatch 복제 중 복제 지연 또는 데이터 동기화 문제는 모니터링하지 않습니다. [설명서에 설명된 대로 별도의 SQL 쿼리를 실행하여 복제 지연 또는 데이터 관련 오류를 모니터링해야 합니다. GoldenGate](#)

제품 버전

- 이 문서는 Linux x86-64에서 Oracle용 Oracle GoldenGate 19.1.0.0.4를 구현한 내용을 기반으로 합니다. 그러나 이 솔루션은 의 모든 주요 버전에 적용할 수 있습니다. GoldenGate

아키텍처

대상 기술 스택

- GoldenGate EC2 인스턴스에 설치된 Oracle용 바이너리
- 아마존 CloudWatch
- Amazon Simple Notification Service(SNS)

대상 아키텍처

도구

서비스

- [CloudWatchAmazon](#)은 이 패턴에서 GoldenGate 오류 로그를 모니터링하는 데 사용되는 모니터링 서비스입니다.
- [Amazon SNS](#)는 이메일 알림을 보내는 데 이 패턴으로 사용되는 메시지 알림 서비스입니다.

기타 도구

- [GoldenGate Oracle](#)은 [Amazon RDS for Oracle](#) 데이터베이스 또는 Amazon EC2에서 호스팅되는 오라클 데이터베이스에 사용할 수 있는 데이터 복제 도구입니다.

높은 수준의 구현 단계

1. CloudWatch 에이전트의 AWS ID 및 액세스 관리 (IAM) 역할을 생성합니다.
2. GoldenGate 오류 로그가 생성되는 EC2 인스턴스에 IAM 역할을 연결합니다.
3. EC2 CloudWatch 인스턴스에 에이전트를 설치합니다.
4. CloudWatch 에이전트 구성 파일 구성: `awsccli.conf` 및 `awslogs.conf`.
5. CloudWatch 에이전트를 시작합니다.

6. 로그 그룹에서 지표 필터를 생성합니다.
7. Amazon SNS를 설정합니다.
8. 지표 필터에 대한 경보를 생성합니다. Amazon SNS는 해당 필터가 이벤트를 포착하면 이메일 경보를 보냅니다.

자세한 지침은 다음 섹션을 참조하세요.

에픽

단계 1. 에이전트의 CloudWatch IAM 역할을 생성합니다.

작업	설명	필요한 기술
IAM 역할을 생성합니다.	<p>AWS 리소스에 액세스하려면 권한이 필요하므로 각 서버에서 CloudWatch 에이전트를 실행하는 데 필요한 권한을 포함하도록 IAM 역할을 생성합니다.</p> <p>IAM 역할을 생성하려면:</p> <ol style="list-style-type: none"> 1. AWS Management Console에 로그인하여 https://console.aws.amazon.com/iam/에서 IAM 콘솔을 엽니다. 2. 탐색 창에서 역할을 선택한 후 역할 생성을 선택합니다. 3. 신뢰할 수 있는 엔터티 유형에서 서비스를 선택합니다. 4. 사용 사례에 대해 EC2를 선택하고 다음을 선택합니다. 5. 정책 목록에서 CloudWatchAgentServerPolicy 옆의 확 	AWS 일반

작업	설명	필요한 기술
	<p>인란을 선택합니다. 필요한 경우 검색 상자를 사용하여 정책을 찾습니다.</p> <p>6. 다음을 선택합니다.</p> <p>7. 역할 이름에 새 역할의 이름(예: goldengate-cw-monitoring-role 또는 자신이 선호하는 다른 이름)을 입력합니다.</p> <p>8. (선택 사항)역할 설명에 설명을 입력합니다.</p> <p>9. 정책 이름 아래에 CloudWatchAgentServerPolicy 표시되는지 확인합니다.</p> <p>10(선택 사항)이 역할에 대한 액세스를 구성, 추적 또는 제어할 태그 키-값 페어를 하나 이상 추가한 후 역할 생성을 선택합니다.</p>	

단계 2. IAM 역할을 GoldenGate EC2 인스턴스에 연결합니다.

작업	설명	필요한 기술
GoldenGate 오류 로그가 생성되는 EC2 인스턴스에 IAM 역할을 연결합니다.	<p>에서 생성된 GoldenGate 오류 로그는 CloudWatch 채워져 모니터링되어야 하므로 1단계에서 생성한 IAM 역할을 실행 중인 EC2 인스턴스에 연결해야 합니다. GoldenGate</p> <p>IAM 역할을 인스턴스에 연결하려면:</p>	AWS 일반

작업	설명	필요한 기술
	<ol style="list-style-type: none"> 1. https://console.aws.amazon.com/ec2/에서 Amazon EC2 콘솔을 엽니다. 2. 탐색 창에서 [Instances]를 선택한 다음 실행 중인 GoldenGate 인스턴스를 찾습니다. 3. 인스턴스를 선택하고 작업, 보안(, IAM 역할 수정을 선택합니다. 4. 인스턴스에 연결할 IAM 역할을 선택한 후 저장을 선택합니다. 	

3~5단계. Goldengate CloudWatch EC2 인스턴스에 에이전트를 설치하고 구성합니다.

작업	설명	필요한 기술
EC2 인스턴스에 CloudWatch 에이전트를 설치합니다. GoldenGate	에이전트를 설치하려면 다음 명령을 실행합니다. <pre>sudo yum install -y awslogs</pre>	AWS 일반
에이전트 구성 파일을 편집합니다.	<ol style="list-style-type: none"> 1. 다음 명령을 실행합니다. <pre>sudo su -</pre> 2. 필요에 따라 이 파일을 편집하여 AWS 리전을 업데이트하세요. <pre>cat /etc/awslogs/conf [plugins] cwlogs = cwlogs</pre> 	AWS 일반

작업	설명	필요한 기술
	<div data-bbox="630 205 1027 306" style="border: 1px solid #ccc; border-radius: 10px; padding: 5px; margin-bottom: 10px;"> <pre>[default] region = us-east-1</pre> </div> <p data-bbox="589 321 1027 785">3. /etc/awslogs/awslogs.conf 파일을 편집하여 파일 이름, 로그 그룹 이름 및 날짜/시간 형식을 업데이트하세요. 의 날짜 형식과 일치하도록 날짜/시간을 지정해야 합니다ggerror.log . 그렇지 않으면 로그 스트림이 유입되지 않습니다. CloudWatch 예:</p> <div data-bbox="630 825 1027 1104" style="border: 1px solid #ccc; border-radius: 10px; padding: 5px;"> <pre>datetime_format = %Y-%m-%dT%H:%M:%S%z file = /u03/oracle/oragg/ggserr.log log_group_name = goldengate_monitor</pre> </div>	
<p data-bbox="115 1140 451 1224">에이전트를 시작합니다. CloudWatch</p>	<p data-bbox="589 1140 992 1224">에이전트를 시작하려면 다음 명령을 실행합니다.</p> <div data-bbox="589 1262 1027 1381" style="border: 1px solid #ccc; border-radius: 10px; padding: 5px; margin-bottom: 10px;"> <pre>\$ sudo service awslogsd start</pre> </div> <p data-bbox="589 1417 1027 1644">에이전트를 시작한 후 CloudWatch 콘솔에서 로그 그룹을 볼 수 있습니다. 로그 스트림에 파일 내용이 포함되어 있습니다.</p>	<p data-bbox="1065 1140 1214 1176">AWS 일반</p>

6단계. 로그 그룹에서 지표 필터를 생성

작업	설명	필요한 기술
<p>ABEND 및 STOPPED 키워드에 대한 지표 필터를 생성하세요.</p>	<p>로그 그룹에 대한 지표 필터를 생성하면 필터가 오류 로그를 식별할 때마다 경보가 시작되고 Amazon SNS 구성을 기반으로 이메일 알림이 전송됩니다.</p> <p>지표 필터를 생성하려면:</p> <ol style="list-style-type: none"> 1. https://console.aws.amazon.com/cloudwatch/ 에서 CloudWatch 콘솔을 엽니다. 2. 로그 그룹의 이름을 선택합니다. 3. 작업을 선택한 후 지표 필터 생성을 선택합니다. 4. 필터 패턴의 경우 ABEND와 같은 패턴을 지정하세요. 5. 다음을 선택하고 지표 필터의 이름을 입력합니다. 6. 메트릭 세부 정보에서 메트릭 네임스페이스에 메트릭이 게시될 CloudWatch 네임스페이스의 이름을 입력합니다. 네임스페이스가 아직 없는 경우 새로 생성이 선택되는지 확인하세요. 7. 지표 필터가 필터에 포함된 키워드의 발생을 계산하는 경우 지표 값에 1을 입력합니다. 	<p>CloudWatch</p>

작업	설명	필요한 기술
	<p>8. 단위를 없음으로 설정합니다.</p> <p>9. 지표 필터 생성을 선택합니다. 생성한 지표 필터를 탐색창에서 찾을 수 있습니다.</p> <p>10 STOPPED 패턴에 대한 또 다른 지표 필터를 생성합니다. 한 로그 그룹 내에서 여러 지표 필터를 만들고 개별적으로 경보를 설정할 수 있습니다.</p>	

7단계. Amazon SNS 설정

작업	설명	필요한 기술
SNS 주제를 생성합니다.	<p>이 단계에서는 지표 필터에 대한 경보를 생성하도록 Amazon SNS를 구성합니다.</p> <p>SNS 주제를 생성하려면:</p> <ol style="list-style-type: none"> https://console.aws.amazon.com/sns/home에서 Amazon SNS 콘솔에 로그인합니다. 주제 생성에서 goldengate-alert 로 주제의 이름을 입력한 다음, 다음 단계를 선택합니다. 유형에서 표준을 선택합니다. 양식의 끝으로 스크롤하고 주제 생성을 선택합니다. 콘 	Amazon SNS

작업	설명	필요한 기술
	<p>솔에서 새 주제의 세부 정보 페이지가 열립니다.</p>	
<p>구독을 생성합니다.</p>	<p>주제 구독을 생성하려면:</p> <ol style="list-style-type: none"> 1. 왼쪽의 탐색 창에서 구독을 선택합니다. 2. 구독 페이지에서 구독 생성을 선택합니다. 3. 구독 생성 페이지에서 주제 ARN 필드를 선택하여 에서 주제 목록을 확인합니다. 4. 이전 단계에서 생성한 주제를 선택합니다. 5. 프로토콜에서 이메일을 선택합니다. 6. 엔드포인트에 알림을 받는 데 사용할 수 있는 이메일 주소를 입력합니다. 7. 구독 생성을 선택합니다. 콘솔에서 새 구독의 세부 정보 페이지를 엽니다. 8. 이메일 받은 편지함을 확인하고 AWS 알림의 이메일에서 구독 확인을 선택합니다. <p>Amazon SNS가 웹 브라우저를 열고 구독 ID와 함께 구독 확인을 표시합니다.</p>	<p>Amazon SNS</p>

8단계. 경보를 생성하여 지표 필터에 대한 알림을 전송

작업	설명	필요한 기술
<p>SNS 주제에 대한 경보를 생성하세요.</p>	<p>로그 그룹 지표 필터를 기반으로 경고 생성하려면:</p> <ol style="list-style-type: none"> 1. https://console.aws.amazon.com/cloudwatch/ 에서 콘솔을 엽니다. CloudWatch 2. 왼쪽 탐색 창에서 로그를 선택한 다음, 로그 그룹을 선택합니다. 3. 지표 필터가 포함된 로그 그룹을 선택합니다. 4. 지표 필터를 선택합니다. 5. 지표 필터 탭에서 경보의 기반으로 사용할 지표 필터의 확인란을 선택합니다. 6. 경고 생성을 선택하세요. 7. 조건의 경우 각 섹션에서 다음을 지정합니다. <ul style="list-style-type: none"> • 임계값 유형에서 정적을 선택합니다. • <metric_name>이 ...과 같을 때마다의 경우 더 큼을 선택합니다. • ...보다에는 0을 지정하세요. 8. 다음을 선택합니다. 9. 알림에서: <ul style="list-style-type: none"> • 경고 상태 트리거에서 경보를 선택합니다. 	<p>CloudWatch</p>

작업	설명	필요한 기술
	<ul style="list-style-type: none"> • 다음 주소로 알림 전송에서 기존 SNS 주제를 선택합니다. • 이메일 상자에서 이전 단계에서 생성한 Amazon SNS 주제를 선택합니다. <p>10.다음을 선택합니다.</p> <p>11.이름 및 설명에서, 경보에 대한 이름과 설명을 입력하세요.</p> <p>참고: 설명에는 알림 이메일에 설명이 포함되도록 인스턴스 이름을 지정할 수 있습니다.</p> <p>12.미리 보기 및 생성에서 구성이 올바른지 확인한 다음 경보 생성을 선택합니다.</p> <p>이 단계를 수행한 후에는 모니터링 중인 GoldenGate 오류 로그 파일 (ggserr.log)에서 이러한 패턴이 탐지될 때마다 이메일 알림을 받게 됩니다.</p>	

문제 해결

문제	Solution
<p>GoldenGate 오류 로그의 로그 스트림은 유입되지 않습니다 CloudWatch.</p>	<p>/etc/awslogs/awslogs.conf 파일을 확인하여 파일 이름, 로그 그룹 이름 및 날짜/시간 형식을 확인하세요. ggserror.log 의 날짜 형식과 일치하도록 날짜/시간을 지정해야 합니다.</p>

문제	Solution
	그렇지 않으면 로그 스트림이 유입되지 않습니다 CloudWatch.

관련 리소스

- [아마존 CloudWatch 설명서](#)
- [CloudWatch 에이전트를 통한 지표 및 로그 수집](#)
- [Amazon SNS 설명서](#)

Amazon RDS for Oracle에서 Oracle Database Enterprise Edition을 Standard Edition 2로 리플랫폼

작성자: Lanre showunmi 및 Tarun Chawla

환경: 프로덕션	소스: 온프레미스	대상: Amazon RDS
R 유형: 리플랫폼	워크로드: Oracle	기술: 데이터베이스
AWS 서비스: Amazon RDS		

요약

Oracle Database Enterprise Edition(EE)은 많은 기업에서 애플리케이션을 실행하는 데 널리 사용되고 있습니다. 그러나 애플리케이션에서 Oracle Database EE 기능을 거의 또는 전혀 사용하지 않는 경우도 있기 때문에 막대한 라이선스 비용이 발생한다는 근거가 부족합니다. Amazon RDS로 마이그레이션할 때 이러한 데이터베이스를 Oracle Database Standard Edition 2(SE2)로 다운그레이드하면 비용을 절감할 수 있습니다.

이 패턴은 온프레미스에서 [Amazon RDS for Oracle](#)로 마이그레이션할 때 Oracle Database EE에서 Oracle Database SE2로 다운그레이드하는 방법을 설명합니다. 이 패턴에 제시된 절차는 EE Oracle 데이터베이스가 이미 Amazon RDS 또는 [Amazon Elastic Compute Cloud](#)(Amazon EC2) 인스턴스에서 실행 중인 경우에도 적용됩니다.

자세한 내용은 [Oracle 데이터베이스를 Standard Edition 2로 다운그레이드하는 것을 평가](#)하는 방법에 대한 AWS Prescriptive Guidance 가이드를 참조하십시오.

사전 조건 및 제한 사항

사전 조건

- 활성 상태의 AWS 계정.
- Oracle Database Enterprise Edition
- Oracle 데이터베이스에서 SQL 명령에 연결되거나 실행하기 위한 [Oracle SQL Developer](#) 또는 SQL*Plus과 같은 클라이언트 도구

- 평가를 수행하기 위한 데이터베이스 사용자의 예는 다음 중 하나입니다.
 - [Schema Conversion Tool\(SCT\)](#) 평가를 실행하기 위한 충분한 [권한](#)을 가진 사용자
 - Oracle 데이터베이스 디렉터리 테이블에서 SQL 쿼리를 실행하기 위한 충분한 권한을 가진 사용자
- 데이터베이스 마이그레이션을 수행하기 위한 사용자의 예는 다음 중 하나입니다.
 - [AWS Database Migration Service\(DMS\)](#)를 실행하기 위한 충분한 [권한](#)을 가진 사용자
 - [Oracle Data Pump 내보내기 및 가져오기](#)를 수행하기 위한 충분한 [권한](#)을 가진 사용자
 - [Oracle을 실행할 수 있는 충분한 권한](#)을 가진 사용자 GoldenGate

제한 사항

- Amazon RDS for Oracle은 최대 데이터베이스 크기를 가집니다. 자세한 내용은 [Amazon RDS DB 인스턴스 스토리지](#)를 참조하세요.

제품 버전

이 문서에 설명된 일반 로직은 9i 이상의 Oracle 버전에 적용됩니다. Oracle 데이터베이스용 자체 관리형 데이터베이스 및 Amazon RDS for Oracle 의 지원되는 버전은 [DMS 설명서](#)를 참조하십시오.

SCT가 지원되지 않는 경우 기능 사용을 식별하려면 소스 데이터베이스에서 SQL 쿼리를 실행합니다. DMS 및 Oracle Data Pump가 지원되지 않는 이전 버전의 Oracle에서 마이그레이션하려면 [Oracle 내보내기 및 가져오기 유틸리티](#)를 사용합니다.

지원되는 버전과 에디션의 현재 목록은 설명서의 [Amazon RDS에서의 Oracle](#)을 참조하세요. 요금과 지원되는 인스턴스 클래스에 대한 자세한 내용은 [Amazon RDS for Oracle 요금](#)을 참조하세요.

아키텍처

소스 기술 스택

- 온프레미스 또는 Amazon EC2에서 실행되는 Oracle Database Enterprise Edition

네이티브 Oracle 도구를 사용한 대상 기술 스택

- Oracle Database SE2를 실행하는 Amazon RDS for Oracle

1. Oracle Data Pump를 사용하여 데이터를 내보냅니다.
2. 데이터베이스 링크를 통해 Amazon RDS에 덤프 파일을 복사합니다.
3. Oracle Data Pump를 사용하여 Amazon RDS로 덤프 파일을 가져옵니다.

AWS DMS를 사용한 대상 기술 스택

- Oracle Database SE2를 실행하는 Amazon RDS for Oracle
- DMS

1. FLASHBACK_SCN를 가진 Oracle Data Pump를 사용하여 데이터를 내보냅니다.
2. 데이터베이스 링크를 통해 Amazon RDS에 덤프 파일을 복사합니다.
3. Oracle Data Pump를 사용하여 Amazon RDS로 덤프 파일을 가져옵니다.
4. DMS [변경 데이터 캡처\(CDC\)](#)를 사용합니다.

도구

서비스

- [AWS Database Migration Service\(DMS\)](#)를 사용하면 데이터 스토어를 클라우드로 마이그레이션하거나 클라우드와 온프레미스 설정 간에 데이터 스토어를 마이그레이션할 수 있습니다.
- [Amazon Relational Database Service\(RDS\)](#)는 AWS 클라우드에서 관계형 데이터베이스를 설정, 운영 및 규모를 조정하는 데 도움이 됩니다. 이 패턴은 Amazon RDS for Oracle을 사용합니다.
- [SCT](#)는 소스 Oracle 데이터베이스의 데이터베이스 스키마를 Amazon RDS for Oracle과 호환되는 형식으로 평가, 변환 및 복사하기 위한 프로젝트 기반 사용자 인터페이스를 제공합니다. SCT를 사용하면 사용자의 라이선스 유형을 Oracle의 Enterprise 에디션에서 Standard 에디션으로 변경함으로써 얻을 수 있는 잠재적인 비용 절감을 분석할 수 있습니다. SCT 보고서의 라이선스 평가 및 클라우드 지원 섹션에서는 사용 중인 Oracle 기능에 대한 자세한 정보를 제공하므로 Amazon RDS for Oracle로 마이그레이션하는 동안 정보에 입각한 결정을 내릴 수 있습니다.

기타 도구

- Native Oracle 가져오기 및 내보내기 유틸리티는 Oracle 데이터를 Oracle 데이터베이스의 내부 및 외부로 이동할 수 있습니다. Oracle은 [Original Export and Import](#)(이전 릴리스의 경우)와 [Oracle Data](#)

[Pump Export and Import](#)(Oracle Database 10g 릴리스 1 이상에서 사용 가능)라는 두 가지 유형의 데이터베이스 가져오기 및 내보내기 유틸리티를 제공합니다.

- [GoldenGate오라클은](#) 실시간 복제 기능을 제공하므로 초기 로드 후 대상 데이터베이스를 동기화할 수 있습니다. 이 옵션은 가동 중 애플리케이션 가동 중지 시간을 줄이는 데 도움이 될 수 있습니다.

에픽

마이그레이션 전 평가하기

작업	설명	필요한 기술
애플리케이션의 데이터베이스 요구 사항을 검증합니다.	애플리케이션이 Oracle Database SE2에서 실행되도록 인증되었는지 확인합니다. 소프트웨어 공급업체, 개발자 또는 설명서를 확인하세요.	앱 개발자, DBA, 앱 소유자
데이터베이스에서 직접 EE 기능의 사용을 조사합니다.	EE 기능 사용을 결정하려면 다음 중 하나를 수행합니다. <ul style="list-style-type: none"> • Oracle EE 데이터베이스에 관한 SCT 평가 보고서를 생성합니다. 이 보고서는 라이선스 유형을 변경하려는 경우 현재 EE 데이터베이스에서 어떤 기능을 제거해야 하는지 알려줍니다. • Oracle Support 계정이 있는 경우 지원 문서 1317265.1안의 <code>options_packs_usage_statistics.sql</code> 스크립트를 구해서 실행하여 오라클 데이터베이스에서 사용 중인 옵션 및 기능에 대한 보고서를 생성합니다. 	앱 소유자, DBA, 앱 개발자

작업	설명	필요한 기술
	<ul style="list-style-type: none"> • DBA_FEATURE_USAGE_STATISTICS를 쿼리하여 사용 중인 모든 기능의 세부 정보를 표시합니다. 	
<p>운영 활동을 위한 EE 기능의 사용을 식별합니다.</p>	<p>데이터베이스 또는 애플리케이션 관리자는 운영 활동을 위해 EE 전용 기능을 사용하는 경우가 있습니다. 일반적인 예로는 온라인 유지 관리 활동(인덱스 재빌드, 테이블 이동) 및 배치 작업에 의한 병렬 처리 사용이 있습니다.</p> <p>가능한 경우 작업을 수정하여 이러한 종속성을 완화할 수 있습니다. 이러한 기능의 사용을 식별하고 비용과 이점을 비교하여 결정을 내립니다.</p> <p>Oracle Database EE와 SE2 기능 비교 표를 가이드로서 사용하여 Oracle Database SE2에서 사용할 수 있는 기능을 식별합니다.</p>	<p>앱 개발자, DBA, 앱 소유자</p>

작업	설명	필요한 기술
<p>EE Oracle 데이터베이스의 워크로드 패턴을 검토합니다.</p>	<p>Oracle Database SE2는 언제든지 최대 16개의 CPU 스레드까지 사용을 자동으로 제한합니다.</p> <p>Oracle EE 데이터베이스에 Oracle Diagnostic Pack을 사용할 수 있는 라이선스가 부여된 경우, Automatic Workload Repository(AWR) 도구 또는 DBA_HIST_* 보기를 사용하여 데이터베이스 워크로드 패턴을 분석해서 SE2로 다운그레이드할 때 최대 제한인 16개의 CPU 스레드가 서비스 수준에 악영향을 미치는지 여부를 판단합니다.</p> <p>평가가 하루, 월말 또는 연말 처리와 같이 활동이 가장 많은 기간을 포함하는지 확인합니다.</p>	<p>앱 소유자, DBA, 앱 개발자</p>

대상 인프라 준비

작업	설명	필요한 기술
<p>네트워킹 인프라를 배포하고 구성합니다.</p>	<p>Virtual Private Cloud(VPC) 및 서브넷, 보안 그룹 및 네트워크 액세스 제어 목록을 생성합니다.</p>	<p>AWS 관리자, 클라우드 아키텍트, 네트워크 관리자, DevOps 엔지니어</p>
<p>Amazon RDS for Oracle SE2 데이터베이스를 프로비저닝합니다.</p>	<p>대상 Amazon RDS for Oracle SE2 데이터베이스를 프로비저닝하여 애플리케이션의 성능, 가용성 및 보안 요구 사항을 충족</p>	<p>클라우드 관리자, 클라우드 아키텍트, DBA, DevOps 엔지니어, AWS 관리자</p>

작업	설명	필요한 기술
	<p>족합니다. 프로덕션 워크로드에는 Multi-AZ를 권장합니다. 하지만 마이그레이션 성능을 개선하기 위해 Multi-AZ 활성화를 데이터 마이그레이션 이후까지 연기할 수 있습니다.</p>	
<p>Amazon RDS 환경을 사용자 지정합니다.</p>	<p>사용자 지정 파라미터 및 옵션을 구성하고 추가 모니터링을 활성화합니다. 더 자세한 내용은, Amazon RDS for Oracle로 마이그레이션하기 위한 모범 사례를 참조하십시오.</p>	<p>AWS 관리자, AWS 시스템 관리자, Cloud 관리자, DBA, Cloud 아키텍트</p>

마이그레이션 예행 연습 및 애플리케이션 테스트 수행

작업	설명	필요한 기술
<p>데이터를 마이그레이션합니다 (예행 연습).</p>	<p>특정 환경에 가장 적합한 접근 방식을 사용하여 소스 Oracle EE 데이터베이스에서 Amazon RDS for Oracle SE2 데이터베이스 인스턴스로 데이터를 마이그레이션합니다. 크기, 복잡성, 이용 가능한 가동 중지 기간 등의 요인을 기반으로 마이그레이션 전략을 선택합니다. 다음 중 하나 또는 조합을 사용합니다.</p> <ul style="list-style-type: none"> • 오라클 데이터 펌프 (권장), 오라클 임포트-익스포트 유틸리티, 오라클과 같은 네이티브 오라클 도구 GoldenGate 	<p>DBA</p>

작업	설명	필요한 기술
	<ul style="list-style-type: none"> CDC를 통한 연속 복제와 함께 전체 로드를 사용하는 DMS. 	
대상 데이터베이스의 개수를 검사합니다.	데이터베이스 스토리지 및 코드 객체의 마이그레이션 후 검증을 수행합니다. 마이그레이션 로그를 검토하고 식별된 문제를 모두 수정합니다. 자세한 내용은 Oracle 데이터베이스를 클라우드로 마이그레이션하기 가이드 를 참조하십시오.	DBA
애플리케이션을 테스트합니다.	<p>애플리케이션 및 데이터베이스 관리자는 적합한 경우 기능, 성능, 운영 테스트를 수행해야 합니다. 더 자세한 내용은, Amazon RDS for Oracle로 마이그레이션하기 위한 모범 사례를 참조하십시오.</p> <p>마지막으로, 이해관계자로부터 테스트 결과에 대한 승인을 받습니다.</p>	앱 개발자, 앱 소유자, DBA, 마이그레이션 엔지니어, 마이그레이션 책임자

전환

작업	설명	필요한 기술
Oracle Database EE에서 데이터를 새로 고칩니다.	애플리케이션 가용성 요구 사항에 따라 데이터 새로 고침 방식을 선택합니다. 자세한 내용은 백서 Strategies for Migrating Oracle Database에	앱 소유자, 전환 리드, DBA, 마이그레이션 엔지니어, 마이그레이션 책임자

작업	설명	필요한 기술
	<p>서 AWS로의 마이그레이션을 위한 전략을 참조하십시오.</p> <p>예를 들어, 지속적인 복제와 함께 Oracle 또는 GoldenGate AWS DMS와 같은 도구를 사용하면 다운타임을 거의 제로에 가깝게 달성할 수 있습니다. 가동 중지 기간이 허용하는 경우 Oracle Data Pump 또는 Original Export-Import 유틸리티와 같은 오프라인 방법을 사용하여 최종 데이터 전환을 수행할 수 있습니다.</p>	
<p>애플리케이션이 대상 데이터베이스 인스턴스를 가리키도록 합니다.</p>	<p>Amazon RDS for Oracle SE2 데이터베이스를 가리키도록 애플리케이션 및 기타 클라이언트의 연결 파라미터를 업데이트합니다.</p>	<p>앱 개발자, 앱 소유자, 마이그레이션 엔지니어, 마이그레이션 책임자, 전환 리드</p>
<p>마이그레이션 후 작업을 수행합니다.</p>	<p>Multi-AZ 활성화, 데이터 검증 및 기타 검사와 같은 데이터 마이그레이션 사후 작업을 수행합니다.</p>	<p>DBA, 마이그레이션 엔지니어</p>
<p>전환 후 모니터링을 수행합니다.</p>	<p>Amazon CloudWatch 및 Amazon RDS Performance Insights와 같은 도구를 사용하여 Oracle SE2용 Amazon RDS 데이터베이스를 모니터링할 수 있습니다.</p>	<p>앱 소유자, 앱 소유자, AWS 관리자, DBA, 앱 개발자</p>

관련 리소스

AWS 권장 가이드

- [Oracle 데이터베이스를 AWS 클라우드로 마이그레이션하기\(가이드\)](#)
- [Evaluate downgrading Oracle 데이터베이스에서 AWS 상에서 표준 에디션 2로의 다운로드 평가\(가이드\)](#)
- [Oracle Data Pump를 사용한 온프레미스 Oracle 데이터베이스에서 Amazon RDS for Oracle로의 마이그레이션\(패턴\)](#)
- [Oracle Data Pump를 사용한 온프레미스 Oracle 데이터베이스에서 Amazon RDS for Oracle로의 마이그레이션\(패턴\)](#)

블로그 게시물

- [AWS DMS를 사용하여 가동 중지 시간이 거의 없이 Oracle 데이터베이스를 마이그레이션하기](#)
- [Amazon RDS for Oracle을 사용하여 Oracle SE에서의 성능 관리 분석](#)
- [Amazon RDS for Oracle을 가지고 Oracle SE에서의 SQL 계획 관리](#)
- [Oracle Standard Edition에서 테이블 파티셔닝 구현: 파트 1](#)

Precisely Connect를 사용하여 메인프레임 데이터베이스를 AWS에 복제하기

작성자: Lucio Pereira, Balaji Mohan, Sayantan Giri

환경: 프로덕션	소스: 온프레미스 메인프레임	대상: AWS 데이터베이스
R 유형: 리아키텍트	워크로드: 기타 모든 워크로드	기술: 데이터베이스, 클라우드 네이티브, 메인프레임, 현대화
AWS 서비스: 아마존 다이내모 DB, 아마존 키스페이스, 아마존 MSK, 아마존 RDS, 아마존 ElastiCache		

요약

이 패턴은 Precisely Connect를 사용하여 거의 실시간으로 데이터를 메인프레임 데이터베이스에서 Amazon 데이터 스토어에 복제하는 절차를 개략적으로 설명합니다. 이 패턴은 Amazon Managed Streaming for Apache Kafka(Amazon MSK)와 함께 이벤트 기반 아키텍처와 클라우드의 사용자 지정 데이터베이스 커넥터를 구현하여 확장성, 복원력, 성능을 개선합니다.

Precisely Connect는 레거시 메인프레임 시스템에서 데이터를 캡처하여 클라우드 환경에 통합하는 복제 도구입니다. 지연 시간이 짧고 처리량이 많은 이기종 데이터 파이프라인을 갖춘 거의 실시간 메시지 흐름을 사용하여 변경 데이터 캡처(CDC)를 통해 데이터를 메인프레임에서 AWS에 복제합니다.

또한 이 패턴에는 다중 리전 데이터 복제 및 장애 조치 라우팅을 사용하는 복원력 있는 데이터 파이프라인을 위한 재해 복구 전략도 포함됩니다.

사전 조건 및 제한 사항

사전 조건

- AWS 클라우드에 복제하려는 기존 메인프레임 데이터베이스(예: IBM DB2, IBM 정보 관리 시스템 (IMS) 또는 가상 스토리지 액세스 방법(VSAM))
- 활성화된 [계정](#)

- 기업 환경에서 AWS로 연결되는 [Direct Connect](#) 또는 [Virtual Private Network \(VPN\)](#)
- 기존 플랫폼에서 연결할 수 있는 서브넷이 있는 [가상 프라이빗 클라우드](#)

아키텍처

소스 기술 스택

다음 데이터베이스 중 하나 이상을 포함하는 메인프레임 환경:

- IBM IMS 데이터베이스
- IBM DB2 데이터베이스
- VSAM 파일

대상 기술 스택

- Amazon MSK
- Amazon Elastic Kubernetes Service(Amazon EKS) 및 Amazon EKS Anywhere
- Docker
- 다음과 같은 AWS 관계형 또는 NoSQL 데이터베이스:
 - Amazon DynamoDB
 - Oracle용 Amazon Relational Database Service(RDS), Amazon RDS for PostgreSQL 또는 Amazon Aurora
 - 아마존 포 ElastiCache 레디스용
 - Amazon Keyspaces(Apache Cassandra용)

대상 아키텍처

메인프레임 데이터를 AWS 데이터베이스에 복제

다음 다이어그램은 DynamoDB, Amazon RDS, Amazon 또는 Amazon Keyspace와 같은 AWS 데이터베이스로 메인프레임 데이터를 복제하는 것을 보여줍니다. ElastiCache 복제는 온프레미스 메인프레임 환경에서 Precisely Capture 및 게시자를 사용하고, 온프레미스 분산 환경에서 Amazon EKS Anywhere의 Precisely Dispatcher를 사용하며, AWS 클라우드에서 Precisely Apply Engine 및 데이터베이스 커넥터를 사용하여 거의 실시간으로 발생합니다.

이 다이어그램은 다음 워크플로를 보여줍니다.

1. Precisely Capture는 CDC 로그에서 메인프레임 데이터를 가져와 내부 임시 스토리지에 그 데이터를 보관합니다.
2. Precisely Publisher는 내부 데이터 스토리지의 변경 사항을 수신 대기하고 TCP/IP 연결을 통해 Precisely Dispatcher로 CDC 레코드를 전송합니다.
3. Precisely Dispatcher는 게시자로부터 CDC 레코드를 수신하여 Amazon MSK로 전송합니다. Dispatcher는 사용자 구성 및 여러 작업자 작업을 기반으로 Kafka 키를 생성하여 데이터를 병렬로 푸시합니다. 레코드가 Amazon MSK에 저장되면 Dispatcher는 게시자에게 확인 메시지를 회송합니다.
4. Amazon MSK는 클라우드 환경에서 CDC 레코드를 보관합니다. 주제의 파티션 크기는 처리량에 대한 트랜잭션 처리 시스템(TPS) 요구 사항에 따라 달라집니다. Kafka 키는 추가 변환 및 트랜잭션 순서 지정을 위해 필수입니다.
5. Precisely Apply Engine은 Amazon MSK로부터 CDC 레코드를 수신하고 대상 데이터베이스 요구 사항에 따라 데이터를 변환합니다(예: 필터링 또는 매핑을 활용하여 변환합니다). Precise SQD 스크립트에 사용자 지정 로직을 추가할 수 있습니다. (SQD는 Precisely의 독점 언어입니다.) Precisely Apply Engine은 각 CDC 레코드를 Apache Avro 또는 JSON 형식으로 변환하고 요구 사항에 따라 다양한 주제에 배포합니다.
6. 대상 Kafka 주제는 대상 데이터베이스를 기반으로 여러 주제에 CDC 레코드를 보관하고 Kafka는 정의된 Kafka키를 기반으로 트랜잭션 순서 지정을 용이하게 합니다. 파티션 키는 해당 파티션에 맞게 정렬되어 순차적 프로세스를 지원합니다.
7. 데이터베이스 커넥터(사용자 지정 Java 애플리케이션)는 Amazon MSK로부터 CDC 레코드를 수신하여 대상 데이터베이스에 저장합니다.
8. 요구 사항에 따라 대상 데이터베이스를 선택할 수 있습니다. 이 패턴은 NoSQL 및 관계형 데이터베이스를 모두 지원합니다.

재해 복구

비즈니스 연속성은 조직 성공을 위한 핵심입니다. AWS 클라우드는 고가용성(HA) 및 재해 복구(DR) 기능을 제공하고 조직의 장애 조치 및 폴백 계획을 지원합니다. 이 패턴은 액티브/패시브 DR 전략을 따르며 RTO 및 RPO 요구 사항을 충족하는 DR 전략을 구현하기 위한 높은 수준의 지침을 제공합니다.

다음 다이어그램은 DR 워크플로를 보여 줍니다.

이 다이어그램은 다음을 보여 줍니다.

1. AWS 리전 1에서 장애가 발생할 경우 반자동 장애 조치가 필요합니다. 리전 1에서 장애가 발생하는 경우 시스템은 Precisely Dispatcher를 리전 2에 연결하기 위해 라우팅 변경을 시작해야 합니다.
2. Amazon MSK는 리전 간 미러링을 통해 데이터를 복제합니다. 따라서 장애 조치 중에는 리전 2의 Amazon MSK 클러스터를 주요 리더로 승격해야 합니다.
3. Precisely Apply Engine과 데이터베이스 커넥터는 어느 리전에서나 작동할 수 있는 상태 비저장 애플리케이션입니다.
4. 데이터베이스 동기화는 대상 데이터베이스에 따라 달라집니다. 예를 들어, DynamoDB는 글로벌 테이블을 사용할 수 있고 글로벌 데이터스토어를 사용할 수 있습니다. ElastiCache

데이터베이스 커넥터를 통한 짧은 지연 시간 및 높은 처리량의 프로세싱

데이터베이스 커넥터는 이 패턴의 중요한 구성 요소입니다. 커넥터는 리스너 기반 접근 방식을 따라 Amazon MSK에서 데이터를 수집하고 미션 크리티컬 애플리케이션(계층 0 및 1)을 위한 높은 처리량과 짧은 지연 시간의 프로세싱을 통해 데이터베이스로 트랜잭션을 전송합니다. 다음 다이어그램에서 이 프로세스를 보여 줍니다.

이 패턴은 멀티스레드 처리 엔진을 통해 단일 스레드를 사용하는 사용자 지정 애플리케이션 개발을 지원합니다.

1. 커넥터 기본 스레드는 Amazon MSK의 CDC 레코드를 사용하고 스레드 풀로 전송하여 처리합니다.
2. 스레드 풀의 스레드는 CDC 레코드를 처리하여 대상 데이터베이스로 보냅니다.
3. 모든 스레드가 사용 중이면 스레드 대기열에 의해 CDC 레코드가 보류 상태로 유지됩니다.
4. 기본 스레드는 스레드 대기열에서 모든 레코드가 지워질 때까지 기다렸다가 오프셋을 Amazon MSK에 커밋합니다.
5. 하위 스레드는 실패를 처리합니다. 처리 중에 실패가 발생하면 실패한 메시지가 DLQ(Dead Letter Queue) 주제로 전송됩니다.
6. 하위 스레드는 데이터베이스의 중복이나 업데이트를 방지하기 위해 메인프레임 타임스탬프를 기반으로 조건부 업데이트 (DynamoDB 설명서의 [조건식](#) 참조) 를 시작합니다. out-of-order

멀티스레딩 기능을 갖춘 Kafka 소비자 애플리케이션을 구현하는 방법에 대한 내용은 Confluent 웹 사이트의 [Apache Kafka Consumer를 사용한 멀티스레드 메시지 사용](#) 블로그 게시물을 참조하십시오.

도구

서비스

- [Amazon Managed Streaming for Apache Kafka\(Amazon MSK\)](#)는 Apache Kafka를 사용하여 스트리밍 데이터를 처리하는 애플리케이션의 구축 및 실행에 도움이 되는 완전 관리형 서비스입니다.
- [Amazon Elastic Kubernetes Service\(Amazon EKS\)](#)는 자체 Kubernetes 컨트롤 플레인이나 노드를 설치하거나 유지 관리할 필요 없이 AWS에서 Kubernetes를 실행할 수 있도록 도와줍니다.
- [Amazon EKS Anywhere](#)를 사용하면 자체 데이터 센터에서 실행되는 Kubernetes 클러스터를 배포, 사용, 관리할 수 있습니다.
- [Amazon DynamoDB](#)는 빠르고 예측 가능하고 확장 가능한 성능을 제공하는 완전 관리형 NoSQL 데이터베이스 서비스입니다.
- [Amazon Relational Database Service\(Amazon RDS\)](#)는 AWS 클라우드에서 관계형 데이터베이스를 설정, 운영 및 조정하는 데 도움이 됩니다.
- [ElastiCacheAmazon](#)은 AWS 클라우드에서 분산된 인메모리 캐시 환경을 설정, 관리 및 확장할 수 있도록 지원합니다.
- [Amazon Keyspaces\(Apache Cassandra용\)](#)는 AWS 클라우드에서 Cassandra 워크로드를 마이그레이션, 실행, 확장할 수 있도록 지원하는 관리형 데이터베이스 서비스입니다.

기타 도구

- [Precisely Connect](#)는 VSAM 데이터 세트 또는 IBM 메인프레임 데이터베이스와 같은 레거시 메인프레임 시스템의 데이터를 차세대 클라우드 및 데이터 플랫폼에 통합합니다.

모범 사례

- 최적의 성능과 비용 균형을 맞출 수 있는 Kafka 파티션과 멀티스레드 커넥터의 최적 조합을 찾습니다. Precisely Capture 및 Dispatcher 인스턴스를 여러 개 사용하면 MIPS(초당 백만 개의 명령) 사용량이 높아져 비용이 증가할 수 있습니다.
- 데이터베이스 커넥터에 데이터 조작 및 변환 로직을 추가하지 마십시오. 이를 위해 마이크로초 단위로 처리 시간을 제공하는 Precisely Apply Engine을 사용합니다.
- 데이터베이스 커넥터에서 데이터베이스에 대한 요청 또는 상태 확인 호출(하트비트)을 정기적으로 생성하여 연결을 자주 워밍업하고 지연 시간을 줄입니다.
- 스레드 풀 검증 로직을 구현하여 스레드 큐에서 보류 중인 작업을 이해하고, 다음 Kafka 폴링 전에 모든 스레드가 완료될 때까지 기다립니다. 이렇게 하면 노드, 컨테이너 또는 프로세스가 충돌하는 경우 데이터 손실을 방지할 수 있습니다.
- 상태 엔드포인트를 통해 지연 시간 지표를 노출하여 대시보드 및 추적 메커니즘을 통해 관찰성 기능을 개선합니다.

에픽

소스 환경(온프레미스) 준비

작업	설명	필요한 기술
메인프레임 프로세스(배치 또는 온라인 유틸리티)를 설정하여 메인프레임 데이터베이스에서 CDC 프로세스를 시작합니다.	<ol style="list-style-type: none"> 메인프레임 환경을 식별합니다. CDC 프로세스에 포함될 메인프레임 데이터베이스를 식별합니다. 메인프레임 환경에서, CDC 도구를 실행하여 소스 데이터베이스의 변경 사항을 캡처하는 프로세스를 개발합니다. 지침은 메인프레임 설명서를 참조하십시오. 구성을 포함한 CDC 프로세스를 문서화합니다. 테스트 및 프로덕션 환경 모두에 프로세스를 배포합니다. 	메인프레임 엔지니어
메인프레임 데이터베이스 로그 스트림을 활성화합니다.	<ol style="list-style-type: none"> 메인프레임 환경에서 로그 스트림을 구성하여 CDC 로그를 캡처합니다. 지침은 메인프레임 설명서를 참조하십시오. 로그 스트림을 테스트하여 필요한 데이터가 캡처되는지 확인합니다. 테스트 및 프로덕션 환경에 로그 스트림을 배포합니다. 	메인프레임 DB 전문가
캡처 구성 요소를 사용하여 CDC 레코드를 캡처합니다.	<ol style="list-style-type: none"> 메인프레임 환경에 Precisely Capture 구성 요소 	메인프레임 엔지니어, Precisely Connect SME

작업	설명	필요한 기술
	<p>를 설치하고 구성합니다. 자세한 지침은 Precisely 설명서를 참조하십시오.</p> <ol style="list-style-type: none"> 구성을 테스트하여 Capture 구성 요소가 제대로 작동하는지 확인합니다. 캡처 구성 요소를 통해 캡처된 CDC 레코드를 복제하도록 복제 프로세스를 설정합니다. 각 소스 데이터베이스의 캡처 구성을 문서화합니다. 캡처 구성 요소가 시간이 지남에 따라 로그를 제대로 수집하는지 확인할 수 있도록 모니터링 시스템을 개발합니다. 테스트 및 프로덕션 환경에 설치 및 구성을 배포합니다. 	

작업	설명	필요한 기술
<p>캡처 구성 요소를 수신하도록 게시자 구성 요소를 구성합니다.</p>	<ol style="list-style-type: none"> 1. 메인프레임 환경에 Precisely Publisher 구성 요소를 설치하고 구성합니다. 자세한 지침은 Precisely 설명서를 참조하십시오. 2. 구성을 테스트하여 Publisher 구성 요소가 제대로 작동하는지 확인합니다. 3. 게시자의 Precisely Dispatcher 구성 요소에 CDC 레코드를 게시하도록 복제 프로세스를 설정합니다. 4. 게시자 구성을 문서화합니다. 5. 게시자 구성 요소가 시간이 지남에 따라 제대로 작동하는지 확인할 수 있도록 모니터링 시스템을 개발합니다. 6. 테스트 및 프로덕션 환경에 설치 및 구성을 배포합니다. 	<p>메인프레임 엔지니어, Precisely Connect SME</p>

작업	설명	필요한 기술
<p>온프레미스 분산 환경에서 Amazon EKS Anywhere를 프로비저닝합니다.</p>	<ol style="list-style-type: none"> 1. 온프레미스 인프라에 Amazon EKS Anywhere를 설치하고 제대로 구성되었는지 확인합니다. 자세한 지침은 Amazon EKS Anywhere 설명서를 참조하십시오. 2. 방화벽을 포함하여 Kubernetes 클러스터를 위한 안전한 네트워크 환경을 설정합니다. 3. Amazon EKS Anywhere 클러스터에 대한 샘플 애플리케이션 배포를 구현하고 테스트합니다. 4. 클러스터의 자동 규모 조정 기능을 구현합니다. 5. 백업 및 재해 복구 절차를 개발하고 구현합니다. 	<p>DevOps 엔지니어</p>

작업	설명	필요한 기술
분산 환경에서 Dispatcher 구성 요소를 배포 및 구성하여 AWS 클라우드에 주제를 게시합니다.	<ol style="list-style-type: none"> 1. Precisely Dispatcher 구성 요소를 구성하고 컨테이너화합니다. 자세한 지침은 Precisely 설명서를 참조하십시오. 2. Dispatcher Docker 이미지를 온프레미스 Amazon EKS Anywhere 환경에 배포합니다. 3. AWS 클라우드와 Dispatcher 간에 보안 연결을 설정합니다. 4. Dispatcher 구성 요소가 시간이 지남에 따라 제대로 작동하는지 확인할 수 있도록 모니터링 시스템을 개발합니다. 5. 테스트 및 프로덕션 환경에 설치 및 구성을 배포합니다. 	DevOps 엔지니어, 프리시커넥트 중소기업

대상 환경(AWS) 준비

작업	설명	필요한 기술
지정된 AWS 리전에 Amazon EKS 클러스터를 프로비저닝합니다.	<ol style="list-style-type: none"> 1. AWS 계정에 로그인하고, Amazon EKS 클러스터를 생성 및 관리하는 데 필요한 권한이 설정되어 있는지 확인할 수 있도록 구성합니다. 2. 선택된 리전에 Virtual Private Cloud(VPC) 및 서브넷을 생성합니다. 자세한 지 	DevOps 엔지니어, 네트워크 관리자

작업	설명	필요한 기술
	<p>침은 Amazon EKS 설명서를 참조하십시오.</p> <p>3. Amazon EKS 클러스터와 VPC의 다른 리소스 간 통신을 허용하는 데 필요한 네트워크 보안 그룹을 생성하고 구성합니다. 자세한 내용은 Amazon EKS 설명서를 참조하십시오.</p> <p>4. Amazon EKS 클러스터를 생성하고 올바른 노드 그룹 크기 및 인스턴스 유형으로 구성합니다.</p> <p>5. 샘플 애플리케이션을 배포하여 Amazon EKS 클러스터를 검증합니다.</p>	
<p>MSK 클러스터를 프로비저닝하고 해당하는 Kafka 주제를 구성합니다.</p>	<p>1. MSK 클러스터를 생성 및 관리하는 데 필요한 권한이 설정되어 있는지 확인할 수 있도록 AWS 계정을 구성합니다.</p> <p>2. MSK 클러스터와 VPC의 다른 리소스 간 통신을 허용하는 데 필요한 네트워크 보안 그룹을 생성하고 구성합니다. 자세한 내용은 Amazon VPC 설명서를 참조하세요.</p> <p>3. MSK 클러스터를 생성하고 애플리케이션에서 사용할 Kafka 주제를 포함하도록 구성합니다. 자세한 내용은 Amazon MSK 설명서를 참조하세요.</p>	<p>DevOps 엔지니어, 네트워크 관리자</p>

작업	설명	필요한 기술
<p>복제된 Kafka 주제를 수신하도록 Apply Engine 구성 요소를 구성합니다.</p>	<ol style="list-style-type: none"> 1. Precisely Apply Engine 구성 요소를 구성하고 컨테이너 화합니다. 2. Apply Engine Docker 이미지를 AWS 계정의 Amazon EKS 클러스터에 배포합니다. 3. MSK 주제를 수신하도록 Apply Engine을 설정합니다. 4. Apply Engine에서 필터링과 변환을 처리할 SQD 스크립트를 개발하고 구성합니다. 자세한 내용은 Precisely 설명서를 참조하세요. 5. 테스트 및 프로덕션 환경에 Apply Engine을 배포합니다. 	<p>Precisely Connect SME</p>

작업	설명	필요한 기술
<p>AWS 클라우드에 DB 인스턴스를 프로비저닝합니다.</p>	<ol style="list-style-type: none"> 1. DB 클러스터와 테이블을 생성 및 관리하는 데 필요한 권한이 설정되어 있는지 확인할 수 있도록 AWS 계정을 구성합니다. 지침은 사용하는 AWS 데이터베이스 서비스에 대한 AWS 설명서를 참조하십시오. (링크는 리소스 섹션을 참조하십시오.) 2. 선택된 리전에 VPC 및 서브넷을 생성합니다. 3. DB 인스턴스와 VPC의 다른 리소스 간 통신을 허용하는 데 필요한 네트워크 보안 그룹을 생성하고 구성합니다. 4. 데이터베이스를 생성해서 애플리케이션에서 사용할 테이블을 포함하도록 구성합니다. 5. 데이터베이스 스키마를 설계하고 검증합니다. 	<p>데이터 엔지니어, DevOps 엔지니어</p>
<p>Apply Engine에서 게시한 주제를 수신하도록 데이터베이스 커넥터를 구성하고 배포합니다.</p>	<ol style="list-style-type: none"> 1. Kafka 주제를 이전 단계에서 생성한 AWS 데이터베이스와 연결하도록 데이터베이스 커넥터를 설계합니다. 2. 대상 데이터베이스를 기반으로 커넥터를 개발합니다. 3. Apply Engine에서 게시한 Kafka 주제를 수신하도록 커넥터를 구성합니다. 4. 커넥터를 Amazon EKS 클러스터에 배포합니다. 	<p>앱 개발자, 클라우드 아키텍트, 데이터 엔지니어</p>

비즈니스 연속성 및 재해 복구 설정

작업	설명	필요한 기술
비즈니스 애플리케이션의 재해 복구 목표를 정의합니다.	<ol style="list-style-type: none"> 비즈니스 요구 사항 및 영향 분석을 기반으로 CDC 파이프라인의 RPO 및 RTO 목표를 정의합니다. 모든 이해관계자가 재해 복구 계획을 숙지할 수 있도록 커뮤니케이션 및 알림 절차를 정의합니다. 재해 복구 계획을 시행하는데 필요한 예산과 리소스를 결정합니다. RPO 및 RTO 목표를 포함한 재해 복구 목표를 문서화합니다. 	클라우드 아키텍트, 데이터 엔지니어, 앱 소유자
정의된 RTO/RPO를 기반으로 재해 복구 전략을 설계합니다.	<ol style="list-style-type: none"> 중요도 및 복구 요구 사항을 기반으로 CDC 파이프라인에 가장 적합한 재해 복구 전략을 결정합니다. 재해 복구 아키텍처 및 토폴로지를 정의합니다. CDC 파이프라인이 백업 리전으로 빠르고 원활하게 전환될 수 있도록 CDC 파이프라인의 장애 조치 및 페일백 절차를 정의합니다. 재해 복구 전략과 절차를 문서화하고 모든 이해관계자가 설계를 명확하게 이해할 수 있도록 합니다. 	클라우드 아키텍트, 데이터 엔지니어

작업	설명	필요한 기술
<p>재해 복구 클러스터 및 구성을 프로비저닝합니다.</p>	<ol style="list-style-type: none"> 1. 재해 복구를 위해 보조 AWS 리전을 프로비저닝합니다. 2. 보조 AWS 리전에서 기본 AWS 리전과 동일한 환경을 생성합니다. 3. 기본 지역과 보조 지역 MirrorMaker 간에 Apache Kafka를 구성합니다. 자세한 내용은 Amazon MSK 설명서를 참조하십시오. 4. 보조 리전에 스탠바이 애플리케이션을 구성합니다. 5. 기본 리전과 보조 리전 간 데이터베이스 복제를 구성합니다. 	<p>DevOps 엔지니어, 네트워크 관리자, 클라우드 아키텍트</p>
<p>재해 복구를 위한 CDC 파이프라인을 테스트합니다.</p>	<ol style="list-style-type: none"> 1. 테스트 시나리오와 달성해야 할 RTO를 포함하여 CDC 파이프라인의 재해 복구 테스트 범위와 목표를 정의합니다. 2. 재해 복구 테스트를 수행하기 위한 테스트 환경과 인프라를 식별합니다. 3. 장애 시나리오를 시뮬레이션하기 위한 테스트 데이터 세트와 스크립트를 준비합니다. 4. 데이터 무결성과 일관성을 검증하여 데이터 손실이 없는지 확인합니다. 	<p>앱 소유자, 데이터 엔지니어, 클라우드 아키텍트</p>

관련 리소스

AWS 리소스

- [Amazon DynamoDB](#)
- [Amazon DynamoDB를 사용한 조건식](#)
- [Amazon EKS](#)
- [Amazon EKS Anywhere](#)
- [아마존 ElasticCache](#)
- [Amazon Keyspaces](#)
- [Amazon MSK](#)
- [Amazon RDS 및 Amazon Aurora](#)
- [Amazon VPC](#)

Precisely Connect 리소스

- [Precisely Connect 개요](#)
- [Precisely Connect를 사용한 데이터 캡처 변경](#)

Confluent 리소스

- [Apache Kafka Consumer를 사용한 멀티스레드 메시지 사용](#)

Lambda와 Secrets Manager를 사용하여 Amazon RDS for PostgreSQL 및 Aurora PostgreSQL 작업 예약하기

작성자: Yaser Raja (AWS)

환경: PoC 또는 파일럿	소스: 데이터베이스: 관계형	대상: PostgreSQL on AWS
R 유형: 해당 없음	워크로드: 오픈 소스	기술: 데이터베이스
AWS 서비스: AWS Lambda; Amazon RDS; AWS Secrets Manager; Amazon Aurora		

요약

Amazon Elastic Compute Cloud(Amazon EC2) 인스턴스에 호스팅되는 데이터베이스 및 온프레미스 데이터베이스의 경우 데이터베이스 관리자가 cron 유틸리티를 사용하여 작업 예약을 하는 경우가 많습니다.

예를 들어, 데이터 추출 작업이나 데이터 제거 작업은 cron을 사용하여 쉽게 예약할 수 있습니다. 이러한 작업의 경우 데이터베이스 보안 인증 정보는 일반적으로 하드 코딩되거나 속성 파일에 저장됩니다. 하지만 Amazon Relational Database Service(Amazon RDS) 또는 Amazon Aurora PostgreSQL-Compatible Edition으로 마이그레이션하면 호스트 인스턴스에 로그인하여 cron 작업을 예약할 수 없게 됩니다.

마이그레이션 후 AWS Lambda 및 AWS Secrets Manager를 사용하여 Amazon RDS for PostgreSQL 및 Aurora PostgreSQL-Compatible 데이터베이스의 작업을 예약합니다.

사전 조건 및 제한 사항

사전 조건

- 활성 상태의 AWS 계정.
- Amazon RDS for PostgreSQL 또는 Aurora PostgreSQL-Compatible 데이터베이스

제한 사항

- 작업은 Lambda 함수 제한 시간인 15분 이내에 완료되어야 합니다. 기본 제한은 [AWS 설명서](#)를 참조하세요.
- 작업 코드는 [Lambda에서 지원하는 언어](#)로 작성해야 합니다.

아키텍처

소스 기술 스택

이 스택의 특징은 Bash, Python, Java와 같은 언어로 작성된 작업입니다. 데이터베이스 보안 인증 정보는 속성 파일에 저장되며 작업은 Linux cron을 사용하여 예약됩니다.

대상 기술 스택

이 스택에는 Secrets Manager에 저장된 보안 인증 정보를 사용하여 데이터베이스에 연결하고 활동을 수행하는 Lambda 함수가 있습니다. Lambda 함수는 Amazon Events를 사용하여 예약된 간격에 따라 시작됩니다. CloudWatch

대상 아키텍처

도구

- [AWS Lambda](#)는 서버를 프로비저닝하거나 관리하지 않고도 코드를 실행할 수 있게 해주는 컴퓨팅 서비스입니다. AWS Lambda는 필요 시에만 코드를 실행하며, 일일 몇 개의 요청에서 초당 수천 개의 요청까지 자동으로 규모를 조정합니다. 사용한 컴퓨팅 시간에 대해서만 요금을 지불하면 되고 코드가 실행되지 않을 때는 요금이 부과되지 않습니다. AWS Lambda에서는 사실상 모든 유형의 애플리케이션이나 백엔드 서비스에 대한 코드를 별도의 관리 없이 실행할 수 있습니다. Lambda는 고가용성 컴퓨팅 인프라에서 코드를 실행하고 서버와 운영 체제 유지 관리, 용량 프로비저닝 및 자동 조정, 코드 및 보안 패치 배포, 코드 모니터링 및 로깅 등 모든 컴퓨팅 리소스 관리를 수행합니다. [AWS Lambda가 지원하는 언어](#) 중 하나로 코드를 공급하기만 하면 됩니다.
- [Amazon CloudWatch Events](#)는 AWS 리소스의 변경 사항을 설명하는 시스템 이벤트의 스트림을 거의 실시간으로 제공합니다. 빠르게 설정할 수 있는 간단한 규칙을 사용하여 이벤트를 매칭하고 하나 이상의 대상 함수 또는 스트림으로 라우팅할 수 있습니다. CloudWatch 이벤트는 운영상의 변화가 발생하는 즉시 이를 인지합니다. 또한 환경에 응답하기 위한 메시지를 전송하고 함수를 활성화하고 변경을 수행하고 상태 정보를 기록하는 등 이러한 운영 변경 사항에 응답하고 필요에 따라 교정 조치를 취합니다. 또한 CloudWatch 이벤트를 사용하여 cron 또는 rate 표현식을 사용하여 특정 시간에 자동으로 시작되는 자동화된 작업을 예약할 수 있습니다.

- [AWS Secrets Manager](#)는 애플리케이션, 서비스, IT 리소스에 액세스하는 데 필요한 보안 암호를 지키도록 도와줍니다. 수명 주기 동안 데이터베이스 자격 증명, API 키 및 기타 보안 암호를 손쉽게 교체, 관리 및 검색할 수 있습니다. 사용자와 애플리케이션은 Secrets Manager API를 호출하여 보안 암호를 검색하면 되므로 중요한 정보를 일반 텍스트로 하드 코딩할 필요가 없습니다. Secrets Manager는 Amazon RDS, Amazon Redshift 및 Amazon DocumentDB에 대한 통합 기능이 내장된 보안 로테이션을 제공합니다. 이 서비스는 API 키 및 OAuth 토큰을 비롯한 다른 유형의 보안 암호로 확장할 수 있습니다. Secrets Manager를 사용하면 세분화된 권한을 사용하여 보안 암호에 대한 액세스를 제어하고 AWS 클라우드, 타사 서비스 및 온프레미스의 리소스에 대해 중앙에서 시크릿 로테이션을 감사할 수 있습니다.

에픽

Secrets Manager에 데이터베이스 보안 인증 정보 저장

작업	설명	필요한 기술
Lambda 함수에 대해 데이터베이스 사용자를 생성합니다.	애플리케이션의 각 부분에 별도의 데이터베이스 사용자를 사용하는 것이 좋습니다. cron 작업에 별도의 데이터베이스 사용자가 이미 있는 경우 해당 사용자를 사용하십시오. 그렇지 않은 경우에는 새 데이터베이스 사용자를 생성하십시오. 자세한 내용은 PostgreSQL 사용자 및 역할 관리 (AWS 블로그 게시물)를 참고하십시오.	DBA
DB 보안 인증 정보를 Secrets Manager에 저장합니다.	데이터베이스 보안 암호 생성 지침 (Secrets Manager 설명서)을 따릅니다.	DBA, DevOps

Lambda 함수의 코드 작성

작업	설명	필요한 기술
AWS Lambda에서 지원하는 프로그래밍 언어를 선택합니다.	지원 언어의 목록은 Lambda 런타임 (Lambda 설명서)을 참고하십시오.	개발자
Secrets Manager에서 데이터베이스 보안 인증 정보를 가져오는 로직을 작성합니다.	샘플 코드는 AWS Secrets Manager를 사용하여 Lambda 함수에 데이터베이스 보안 인증 정보를 안전하게 제공하는 방법 (AWS 블로그 게시물)을 참고하십시오.	개발자
예약된 데이터베이스 활동을 수행하는 로직을 작성합니다.	온프레미스에서 사용 중인 예약 작업의 기존 코드를 AWS Lambda 함수로 마이그레이션합니다. 자세한 내용은 Python을 사용하여 Lambda 함수 빌드 (Lambda 설명서) 를 참조하세요.	개발자

코드를 배포하고 Lambda 함수를 생성합니다.

작업	설명	필요한 기술
Lambda 함수 배포 패키지를 생성합니다.	이 패키지에는 코드와 해당 종속 항목을 포함되어 있습니다. 자세한 내용은 배포 패키지 (Lambda 설명서)를 참고하십시오.	개발자
Lambda 함수를 생성합니다.	AWS Lambda 콘솔에서 함수 생성을 선택하고, 함수 이름을 입력하고, 런타임 환경을 선택	DevOps

작업	설명	필요한 기술
	한 다음, 함수 생성을 선택합니다.	
배포 패키지를 업로드합니다.	생성한 Lambda 함수를 선택하여 해당 구성을 엽니다. 코드 섹션에서 직접 코드를 작성하거나 배포 패키지를 업로드할 수 있습니다. 패키지를 업로드하려면 함수 코드 섹션으로 이동합니다. 그리고 코드 입력 유형을 선택하여 .zip 파일을 업로드한 다음 패키지를 선택합니다.	DevOps
요구 사항에 따라 Lambda 함수를 구성합니다.	예를 들어 제한 시간 파라미터를 Lambda 함수가 걸릴 것으로 예상되는 지속 시간으로 설정할 수 있습니다. 자세한 내용은 함수 옵션 구성 (Lambda 설명서)을 참조하세요.	DevOps
Lambda 함수 역할에 대해 Secrets Manager에 액세스할 수 있는 권한을 설정합니다.	지침은 AWS Lambda 함수에서 시크릿 사용 (Secrets Manager 설명서)을 참고하십시오.	DevOps
Lambda 함수 테스트	함수를 수동으로 시작하여 예상대로 작동하는지 확인하십시오.	DevOps

이벤트를 사용하여 Lambda 함수를 스케줄링합니다. CloudWatch

작업	설명	필요한 기술
예약된 일정에 따라 Lambda 함수를 실행하도록 규칙을 생성합니다.	이벤트를 사용하여 Lambda 함수를 스케줄링합니다. CloudWatch 지침은 이벤트를 CloudWatch 사용한 Lambda 함수 스케줄 지정 CloudWatch (이벤트 자습서) 을 참조하십시오.	DevOps

관련 리소스

- [AWS Secrets Manager](#)
- [Lambda 시작하기](#)
- [CloudWatch 이벤트를 트리거하는 이벤트 규칙 생성](#)
- [AWS Lambda 제한](#)
- [서버리스 애플리케이션에서 AWS 데이터베이스 쿼리](#)(블로그 게시물)

신뢰할 수 있는 컨텍스트를 사용하여 AWS에서 Db2 페더레이션 데이터베이스의 사용자 액세스 보호 및 간소화

작성자: Sai Parthasaradhi(AWS)

환경: PoC 또는 파일럿

기술: 데이터베이스, 보안, 자격 증명, 규정 준수

워크로드: IBM

AWS 서비스: Amazon EC2

요약

많은 기업이 기존 메인프레임 워크로드를 Amazon Web Services(AWS)로 마이그레이션하고 있습니다. 이 마이그레이션에는 Amazon Elastic Compute Cloud(Amazon EC2)에서 IBM Db2 for z/OS 데이터베이스를 Db2 for Linux, Unix and Windows(LUW)로 전환하는 작업이 포함됩니다. 온프레미스에서 AWS로 단계별 마이그레이션하는 동안 사용자는 모든 애플리케이션과 데이터베이스가 Db2 LUW로 완전히 마이그레이션될 때까지 Amazon EC2의 IBM Db2 z/OS 및 Db2 LUW에 있는 데이터에 액세스해야 할 수 있습니다. 이러한 원격 데이터 액세스 시나리오에서는 플랫폼마다 다른 인증 메커니즘을 사용하기 때문에 사용자 인증이 어려울 수도 있습니다.

이 패턴은 Db2 for z/OS를 원격 데이터베이스로 사용하여 Db2 for LUW에서 페더레이션 서버를 설정하는 방법을 다룹니다. 이 패턴은 원격 데이터베이스에서 재인증하지 않고도 신뢰할 수 있는 컨텍스트를 사용하여 사용자 ID를 Db2 LUW에서 Db2 z/OS로 전파합니다. 신뢰할 수 있는 컨텍스트에 대한 자세한 내용은 [추가 정보](#) 섹션을 참조하세요.

사전 조건 및 제한 사항

사전 조건

- 활성 상태의 AWS 계정.
- Amazon EC2 인스턴스에서 실행되는 Db2 인스턴스
- 온프레미스에서 실행되는 원격 Db2 for z/OS 데이터베이스
- [AWS Site-to-Site VPN](#) 또는 [AWS Direct Connect](#)를 통해 AWS에 연결된 온프레미스 네트워크

아키텍처

대상 아키텍처

도구

서비스

- [Amazon Elastic Compute Cloud\(Amazon EC2\)](#)는 AWS 클라우드에서 확장 가능한 컴퓨팅 용량을 제공합니다. 필요한 만큼 가상 서버를 시작하고 빠르게 스케일 업하거나 스케일 다운할 수 있습니다.
- [AWS Site-to-Site VPN](#)을 사용하면 AWS에서 시작하는 인스턴스와 자체 원격 네트워크 간에 트래픽을 전달할 수 있습니다.

기타 서비스

- [db2cli](#)는 Db2 대화형 명령줄 인터페이스(CLI) 명령입니다.

에픽

AWS에서 실행되는 Db2 LUW 데이터베이스에서 페더레이션 활성화

작업	설명	필요한 기술
DB2 LUW DB에서 페더레이션을 활성화합니다.	DB2 LUW에서 페더레이션을 활성화하려면 다음 명령을 실행합니다. <pre>update dbm cfg using federated YES</pre>	DBA
데이터베이스를 다시 시작합니다.	데이터베이스를 다시 시작하려면 다음 명령을 실행합니다. <pre>db2stop force; db2start;</pre>	DBA

원격 데이터베이스 카탈로그화

작업	설명	필요한 기술
원격 Db2 z/OS 하위 시스템을 카탈로그화합니다.	<p>AWS에서 실행되는 Db2 LUW의 원격 Db2 z/OS 데이터베이스를 카탈로그화하려면 다음 예제 명령을 사용합니다.</p> <pre>catalog TCPIP NODE tcpname REMOTE mainframehost SERVER mainframeport</pre>	DBA
원격 데이터베이스를 카탈로그화합니다.	<p>원격 데이터베이스를 카탈로그화하려면 다음 예제 명령을 사용합니다.</p> <pre>catalog db dbnam1 as ndbnam1 at node tcpname</pre>	DBA

원격 서버 정의 생성

작업	설명	필요한 기술
원격 Db2 z/OS 데이터베이스의 사용자 보안 인증 정보를 수집합니다.	<p>다음 단계를 진행하기 전에 다음 정보를 수집합니다.</p> <ul style="list-style-type: none"> Db2 z/OS 하위 시스템 이름-이전 단계에서 카탈로그화된 LUW의 Db2 z/OS 이름(예: ndbnam1) Db2 z/OS 버전-Db2 z/OS 하위 시스템 버전(예: 12) Db2 z/OS 사용자 ID-서버 정의만 생성하는 데 필요한 	DBA

작업	설명	필요한 기술
	<p>BIND 권한을 가진 사용자 (예: dbuser1)</p> <ul style="list-style-type: none"> • Db2 z/OS 암호-dbuser1의 암호(예: dbpasswd) • Db2 z/OS 프록시 사용자-신뢰할 수 있는 연결을 설정하는데 사용할 프록시 사용자 ID(예: zproxy) • Db2 z/OS 프록시 암호-zproxy 사용자의 암호(예: zproxy) 	
DRDA 래퍼를 생성합니다.	<p>DRDA 래퍼를 생성하려면 다음 명령을 실행합니다.</p> <pre>CREATE WRAPPER DRDA;</pre>	DBA

작업	설명	필요한 기술
서버 정의를 생성합니다.	<p>서버 정의를 생성하려면 다음 예제 명령을 실행합니다.</p> <pre>CREATE SERVER ndbserver TYPE DB2/ZOS VERSION 12 WRAPPER DRDA AUTHORIZATION "dbuser1" PASSWORD "dbpasswd" " OPTIONS (DBNAME 'ndbnam1 ',FED_PROXY _USER 'ZPROXY');</pre> <p>이 정의에서 FED_PROXY_USER 는 Db2 z/OS 데이터베이스에 대한 신뢰할 수 있는 연결을 설정하는 데 사용할 프록시 사용자를 지정합니다. 인증 사용자 ID와 암호는 Db2 LUW 데이터베이스에서 원격 서버 객체를 생성하는 데에만 필요합니다. 나중에 런타임 중에는 사용되지 않습니다.</p>	DBA

사용자 매핑 생성

작업	설명	필요한 기술
프록시 사용자를 위한 사용자 매핑을 생성합니다.	<p>프록시 사용자를 위한 사용자 매핑을 생성하려면 다음 명령을 실행합니다.</p> <pre>CREATE USER MAPPING FOR ZPROXY SERVER ndbserver OPTIONS (REMOTE_AUTHID</pre>	DBA

작업	설명	필요한 기술
	<pre>'ZPROXY', REMOTE_PA SSWORD 'zproxy');</pre>	
<p>Db2 LUW에서 각 사용자에게 대한 사용자 매핑을 생성합니다.</p>	<p>프록시 사용자를 통해 원격 데이터에 액세스해야 하는 AWS의 Db2 LUW 데이터베이스에 있는 모든 사용자에게 대한 사용자 매핑을 생성합니다. 사용자 매핑을 생성하려면 다음 명령을 실행합니다.</p> <pre>CREATE USER MAPPING FOR PERSON1 SERVER ndbserver OPTIONS (REMOTE_AUTHID 'USERZID', USE_TRUST ED_CONTEXT 'Y');</pre> <p>이 문은 Db2 LUW의 사용자 (PERSON1)가 원격 Db2 z/OS 데이터베이스(USE_TRUSTED_CONTEXT 'Y')에 신뢰할 수 있는 연결을 설정할 수 있음을 지정합니다. 프록시 사용자를 통해 연결이 설정되면 사용자는 Db2 z/OS 사용자 ID(REMOTE_AUTHID 'USERZID')를 사용하여 데이터에 액세스할 수 있습니다.</p>	<p>DBA</p>

신뢰할 수 있는 컨텍스트 객체 생성

작업	설명	필요한 기술
<p>신뢰할 수 있는 컨텍스트 객체를 생성합니다.</p>	<p>원격 Db2 z/OS 데이터베이스에서 신뢰할 수 있는 컨텍스트 객체를 생성하려면 다음 예제 명령을 사용합니다.</p> <pre data-bbox="594 548 1027 1104"> CREATE TRUSTED CONTEXT CTX_LUW_ZOS BASED UPON CONNECTION USING SYSTEM AUTHID ZPROXY ATTRIBUTES (ADDRESS '10.10.10.10') NO DEFAULT ROLE ENABLE WITH USE FOR PUBLIC WITHOUT AUTHENTICATION; </pre> <p>이 정의에서 CTX_LUW_ZOS 는 신뢰할 수 있는 컨텍스트 객체의 임의 이름입니다. 객체에는 프록시 사용자 ID와 신뢰할 수 있는 연결이 시작되어야 하는 서버의 IP 주소가 포함됩니다. 이 예에서는 서버가 AWS의 Db2 LUW 데이터베이스입니다. IP 주소 대신 도메인 이름을 사용할 수 있습니다. WITH USE FOR PUBLIC WITHOUT AUTHENTICATION 절은 모든 사용자 ID에 대해 신뢰할 수 있는 연결에서 사용자 ID를 전환</p>	<p>DBA</p>

작업	설명	필요한 기술
	할 수 있음을 나타냅니다. 암호는 제공할 필요가 없습니다.	

관련 리소스

- [IBM Resource Access Control Facility\(RACF\)](#)
- [IBM Db2 LUW 페더레이션](#)
- [신뢰할 수 있는 컨텍스트](#)

추가 정보

Db2 신뢰할 수 있는 컨텍스트

신뢰할 수 있는 컨텍스트는 페더레이션된 서버와 원격 데이터베이스 서버 간의 신뢰 관계를 정의하는 Db2 데이터베이스 객체입니다. 신뢰할 수 있는 관계를 정의하기 위해 신뢰할 수 있는 컨텍스트는 신뢰 속성을 지정합니다. 신뢰 속성에는 다음 세 가지 유형이 있습니다.

- 초기 데이터베이스 연결 요청을 수행하는 시스템 인증 ID
- 연결이 이루어진 IP 주소 또는 도메인 이름
- 데이터베이스 서버와 데이터베이스 클라이언트 간의 데이터 통신을 위한 암호화 설정

연결 요청의 모든 속성이 서버에 정의된 신뢰할 수 있는 컨텍스트 객체에 지정된 속성과 일치할 때 신뢰할 수 있는 연결이 설정됩니다. 신뢰할 수 있는 연결에는 암시적 연결과 명시적 연결의 두 가지 유형이 있습니다. 암시적으로 신뢰할 수 있는 연결이 설정되면 사용자는 해당 신뢰할 수 있는 연결 정의의 범위 밖에서 사용할 수 없는 역할을 상속받습니다. 명시적으로 신뢰할 수 있는 연결이 설정되면 인증을 사용하거나 사용하지 않고 사용자를 동일한 물리적 연결로 전환할 수 있습니다. 또한 신뢰할 수 있는 연결 내에서만 사용할 수 있는 권한을 지정하는 역할을 Db2 사용자에게 부여할 수 있습니다. 이 패턴은 명시적으로 신뢰할 수 있는 연결을 사용합니다.

이 패턴의 신뢰할 수 있는 컨텍스트

패턴이 완료되면 Db2 LUW의 PERSON1은 페더레이션된 신뢰할 수 있는 컨텍스트를 사용하여 Db2 z/OS의 원격 데이터에 액세스합니다. PERSON1의 연결이 신뢰할 수 있는 컨텍스트 정의에 지정된 IP 주소 또는 도메인 이름에서 시작된 경우 연결은 프록시 사용자를 통해 설정됩니다. 연결이 설정되면

PERSON1의 해당 Db2 z/OS 사용자 ID는 재인증 없이 전환되며 사용자는 해당 사용자에 대해 설정된 Db2 권한에 따라 데이터 또는 객체에 액세스할 수 있습니다.

페더레이션된 신뢰할 수 있는 컨텍스트의 이점

- 이 접근 방식은 모든 사용자에게 필요한 모든 권한의 상위 집합이 필요한 공통 사용자 ID나 애플리케이션 ID를 사용하지 않으므로 최소 권한 원칙을 유지합니다.
- 페더레이션된 데이터베이스와 원격 데이터베이스 모두에서 트랜잭션을 수행하는 사용자의 실제 ID는 항상 알려져 있으며 감사할 수 있습니다.
- 페더레이션된 서버를 재인증할 필요 없이 사용자 간에 물리적 연결이 재사용되므로 성능이 향상됩니다.

온프레미스 SMTP 서버 및 Database Mail을 사용하여 Amazon RDS for SQL Server 데이터베이스 인스턴스에 대한 알림 전송하기

작성자: Nishad Mankar (AWS)

환경: PoC 또는 파일럿

기술: 데이터베이스, 관리 및 거버넌스

워크로드: Microsoft

AWS 서비스: Amazon RDS

요약

[Database Mail](#)(Microsoft 설명서)은 SMTP(단순 메일 전송 프로토콜) 서버를 사용하여 Microsoft SQL Server 데이터베이스에서 알림 또는 경고와 같은 이메일 메시지를 보냅니다. Microsoft SQL Server용 Amazon Relational Database Service(Amazon RDS) 설명서에는 Amazon Simple Email Service(Amazon SES)를 Database Mail용 SMTP 서버로 사용하는 방법에 대한 지침이 나와 있습니다. 자세한 내용은 [Amazon RDS for SQL Server에 Database Mail 사용](#) 섹션을 참조하세요. 다른 구성으로, 이 패턴은 온프레미스 SMTP 서버를 메일 서버로 사용하여 Amazon RDS for SQL Server 데이터베이스(DB) 인스턴스에서 이메일을 보내도록 Database Mail을 구성하는 방법을 설명합니다.

사전 조건 및 제한 사항

사전 조건

- 활성 상태의 AWS 계정.
- 표준 또는 엔터프라이즈 버전의 SQL Server를 실행하는 Amazon RDS DB 인스턴스
- 온프레미스 SMTP 서버의 IP 주소 또는 호스트 이름
- SMTP 서버의 IP 주소에서 Amazon RDS for SQL Server DB 인스턴스로의 연결을 허용하는 인바운드 [보안 그룹 규칙](#)
- 온프레미스 네트워크와 Amazon RDS DB 인스턴스가 포함된 가상 사설 클라우드(VPC) 간의 연결 (예: [AWS Direct Connect](#) 연결)

제한 사항

- Express 버전의 SQL Server는 지원되지 않습니다.

- 제한 사항에 대한 자세한 내용은 Amazon RDS 설명서의 [Amazon RDS for SQL Server에 Database Mail 사용 제한](#) 섹션을 참고하십시오.

제품 버전

- [RDS에서 지원되는 SQL Server 버전](#)의 스탠다드 및 엔터프라이즈 에디션

아키텍처

대상 기술 스택

- Amazon RDS for SQL Server 데이터베이스 인스턴스
- Amazon Route 53 전달 규칙
- 데이터베이스 메일
- 온프레미스 SMTP 서버
- Microsoft SQL Server Management Studio(SSMS)

대상 아키텍처

다음 이미지는 이 패턴의 대상 아키텍처를 보여줍니다. 데이터베이스 인스턴스와 관련된 알림 또는 경고를 시작하는 이벤트 또는 작업이 발생하면 Amazon RDS for SQL Server는 Database Mail을 사용하여 이메일 알림을 보냅니다. Database Mail은 온프레미스 SMTP 서버를 사용하여 이메일을 전송합니다.

도구

서비스

- [Microsoft SQL Server용 Amazon Relational Database Service\(Amazon RDS\)](#)를 사용하여 AWS 클라우드에서 SQL Server 관계형 데이터베이스를 설정, 운영 및 확장할 수 있습니다.
- [Amazon Route 53](#)은 가용성과 확장성이 뛰어난 DNS 웹 서비스입니다.

기타 도구

- [Database Mail](#)은 SQL Server 데이터베이스 엔진에서 사용자에게 알림 및 경고와 같은 이메일 메시지를 보내는 도구입니다.
- [Microsoft SQL Server Management Studio\(SSMS\)](#)는 SQL 서버 구성 요소에 대한 액세스, 구성 및 관리를 포함하여 SQL Server를 관리하기 위한 도구입니다. 이 패턴에서는 SSMS를 사용하여 SQL 명령을 실행하는 방법으로 Amazon RDS for SQL Server DB 인스턴스에서 Database Mail을 설정합니다.

에픽

온프레미스 SMTP 서버와의 네트워크 연결 활성화

작업	설명	필요한 기술
RDS DB 인스턴스에서 다중 AZ를 제거합니다.	다중 영역 RDS DB 인스턴스를 사용하는 경우 다중 AZ 인스턴스를 단일 AZ 인스턴스로 변환합니다. Database Mail 구성을 마치면 DB 인스턴스가 다시 다중 AZ 배포로 변환됩니다. 그러면 Database Mail 구성이 기본 노드와 보조 노드에서 모두 작동합니다. 지침은 Microsoft SQL Server DB 인스턴스에서 다중 AZ 제거하기 를 참조하세요.	DBA
온프레미스 SMTP 서버의 Amazon RDS 엔드포인트 또는 IP 주소에 대한 허용 목록을 생성합니다.	SMTP 서버는 AWS 네트워크 외부에 있습니다. 온프레미스 SMTP 서버에서 서버가 Amazon RDS에서 호스팅되는 Amazon RDS 인스턴스 또는 Amazon Elastic Compute Cloud(Amazon EC2) 인스턴스의 아웃바운드 엔드포인트 또는 IP 주소와 통신할 수 있도록 허용하는 허용 목록을 생성합니다. 이 절차는 조직마다 다름	DBA

작업	설명	필요한 기술
	<p>니다. DB 인스턴스 엔드포인트에 대한 자세한 내용은 DB 인스턴스 엔드포인트 및 포트 번호 찾기를 참고하십시오.</p>	
<p>포트 25 제한을 제거합니다.</p>	<p>기본적으로 AWS는 EC2 인스턴스에서 포트 25를 제한합니다. 포트 25 제한을 제거하려면 다음 작업을 수행합니다.</p> <ol style="list-style-type: none"> 1. AWS 계정으로 로그인한 다음 이메일 전송 제한 제거 요청 양식을 엽니다. 2. AWS Support에서 요청에 대한 업데이트 내용을 알려 줄 수 있도록 이메일 주소를 입력하십시오. 3. 사용 사례 설명 필드에 필수 정보를 입력합니다. 4. 제출을 선택합니다. <p>참고:</p> <ul style="list-style-type: none"> • 둘 이상의 AWS 리전에 인스턴스가 있는 경우 각 리전에 대해 별도의 요청을 제출합니다. • 요청이 처리되기까지 최대 48시간이 걸릴 수 있습니다. 	<p>일반 AWS</p>

작업	설명	필요한 기술
Route 53 규칙을 추가하여 SMTP 서버에 대한 DNS 쿼리를 해결합니다.	Route 53을 사용하여 AWS 리소스와 온프레미스 SMTP 서버 간의 DNS 쿼리를 해결할 수 있습니다. DNS 쿼리를 SMTP 서버 도메인으로 전달하는 규칙(예: example.com)을 생성해야 합니다. 지침은 Route 53 설명서의 전달 규칙 생성 을 참고하십시오.	네트워크 관리자

Amazon RDS for SQL Server DB 인스턴스에서 Database Mail 설정하기

작업	설명	필요한 기술
Database Mail을 활성화합니다.	Database Mail용 매개변수 그룹을 만들고 database mail xps 매개변수를 1로 설정한 다음 Database Mail 매개변수 그룹을 대상 RDS DB 인스턴스와 연결합니다. 지침은 Amazon RDS 설명서의 Database Mail 활성화 를 참고하십시오. 이 지침의 Database Mail 구성 섹션으로 진행하지 마십시오. 온프레미스 SMTP 서버의 구성은 Amazon SES와 다릅니다.	DBA
DB 인스턴스에 연결합니다.	배스천 호스트에서 Microsoft SQL Server Management Studio(SSMS)를 사용하여 Amazon RDS for SQL Server 데이터베이스 인스턴스에 연결합니다. 지침은 Microsoft SQL Server 데이터베이스 엔진을	DBA

작업	설명	필요한 기술
	<p>실행하는 DB 인스턴스에 연결하기를 참조하세요. 오류가 발생하는 경우 관련 리소스 섹션의 연결 문제 해결 참조를 참고하십시오.</p>	
<p>프로필을 생성합니다.</p>	<p>SSMS에 다음 SQL 문을 입력하여 Database Mail 프로필을 생성합니다. 다음 값을 교체합니다.</p> <ul style="list-style-type: none"> • <code>profile_name</code>에는 새 프로필의 이름을 입력합니다. • <code>description</code>에는 새 프로파일에 대한 간략한 설명을 입력합니다. <p>저장된 이 프로시저와 해당 인수에 대한 자세한 내용은 Microsoft 설명서의 sysmail_add_profile_sp를 참조하십시오.</p> <pre>EXECUTE msdb.dbo. sysmail_add_profile_sp @profile_name = 'SQL Alerts profile', @description = 'Profile used for sending outgoing notifications using OM SMTP Server.';</pre>	<p>DBA</p>

작업	설명	필요한 기술
<p>프로필에 보안 주체를 추가합니다.</p>	<p>다음과 같은 SQL 문을 입력하여 Database Mail 프로필에 공용 또는 개인 보안 주체를 추가합니다. 보안 주체는 SQL Server 리소스를 요청할 수 있는 엔터티입니다. 다음 값을 교체합니다.</p> <ul style="list-style-type: none"> • <code>profile_name</code> 에는 앞에서 생성한 프로필의 이름을 입력합니다. • <code>principal_name</code> 에는 데이터베이스 사용자 또는 역할의 이름을 입력합니다. 이 값은 SQL Server 인증 사용자, Windows 인증 사용자 또는 Windows 인증 그룹에 매핑되어야 합니다. <p>이 저장 프로시저와 해당 인수에 대한 자세한 내용은 Microsoft 설명서의 sysmail_add_principalprofile_sp를 참조하세요.</p> <pre>EXECUTE msdb.dbo. sysmail_add_principalprofile_sp @profile_name = 'SQL Alerts profile', @principal_name = 'public', @is_default = 1 ;</pre>	<p>DBA</p>

작업	설명	필요한 기술
계정을 생성합니다.	<p>다음 SQL 문을 입력하여 Database Mail 계정을 생성합니다. 다음 값을 교체합니다.</p> <ul style="list-style-type: none"> • <code>account_name</code> 에는 새 계정의 이름을 입력합니다. • <code>description</code> 에는 새 계정에 대한 간략한 설명을 입력합니다. • <code>email_address</code> 에는 Database Mail 메시지를 보낼 이메일 주소를 입력합니다. • <code>display_address</code> 에는 이 계정의 발신 메시지에 사용할 표시 이름(예: SQL Server Automated Notification)을 입력합니다. <code>email_address</code> 에 입력한 값을 사용할 수도 있습니다. • <code>mailserver_name</code> 에는 SMTP 메일 서버의 이름 또는 IP 주소를 입력합니다. • <code>port</code>에는 25의 값은 그대로 두십시오. • <code>enable_ssl</code> 에는 Database Mail이 SSL을 사용하여 통신을 암호화하지 않도록 하려면 1에 값을 그대로 두거나 0을 입력하십시오. • <code>username</code>에는 SMTP 메일 서버에 로그인하기 위한 사 	DBA

작업	설명	필요한 기술
	<p>용자 이름을 입력합니다. 서버에 인증이 필요하지 않은 경우 NULL을 입력합니다.</p> <ul style="list-style-type: none"> password에는 SMTP 메일 서버에 로그인하기 위한 암호를 입력합니다. 서버에 인증이 필요하지 않은 경우 NULL을 입력합니다. <p>이 저장 프로시저와 해당 인수에 대한 자세한 내용은 Microsoft 설명서의 sysmail_add_account_sp를 참조하세요.</p> <pre>EXECUTE msdb.dbo. sysmail_add_account_sp @account_name = 'SQL Alerts account', @description = 'Database Mail account for sending outgoing notifications.', @email_address = 'xyz@example.com', @display_name = 'xyz@example.com', @mailserver_name = 'test_smtp.example .com', @port = 25, @enable_ssl = 1, @username = 'SMTP-use rname', @password = 'SMTP-pas sword';</pre>	

작업	설명	필요한 기술
<p>프로필에 계정을 추가합니다.</p>	<p>다음과 같은 SQL 문을 입력하여 Database Mail 계정을 Database Mail 프로필에 추가합니다. 다음 값을 교체합니다.</p> <ul style="list-style-type: none"> • profile_name 에는 앞에서 생성한 프로필의 이름을 입력합니다. • account_name 에는 앞서 생성한 계정의 이름을 입력합니다. <p>이 저장 프로시저와 해당 인수에 대한 자세한 내용은 Microsoft 설명서의 sysmail_add_profileaccount_sp를 참조하세요.</p> <pre>EXECUTE msdb.dbo. sysmail_add_profileaccount_sp @profile_name = 'SQL Alerts profile', @account_name = 'SQL Alerts account', @sequence_number = 1;</pre>	<p>DBA</p>
<p>(선택 사항) RDS DB 인스턴스에 다중 AZ를 추가합니다.</p>	<p>Database Mirroring(DBM) 또는 Always On 가용 그룹(AG)을 이용한 다중 AZ를 추가하려면 Microsoft SQL Server DB 인스턴스에 다중 AZ 추가에 있는 지침을 참고하십시오.</p>	<p>DBA</p>

관련 리소스

- [Amazon RDS for SQL Server에서 Database Mail 사용하기](#) (Amazon RDS 설명서)
- [파일 첨부 작업](#) (Amazon RDS 설명서)
- [SQL Server DB 인스턴스에 대한 연결 문제 해결](#)(Amazon RDS 설명서)
- [Amazon RDS DB 인스턴스에 연결할 수 없음](#) (Amazon RDS 설명서)

AWS 기반 IBM Db2에서 SAP를 위한 재해 복구 설정

제작자: Ambarish Satarkar(AWS) 및 Debasis Sahoo(AWS)

환경: 프로덕션

기술: 데이터베이스, 운영

워크로드: SAP

AWS 서비스: Amazon EC2, AWS Elastic Disaster Recovery

요약

이 패턴은 데이터베이스 플랫폼으로 IBM Db2를 사용하여 Amazon Web Services(AWS) 클라우드에서 실행되는 SAP 워크로드용 재해 복구(DR) 시스템을 설정하는 단계를 설명합니다. 목표는 정전 발생 시 비즈니스 연속성을 제공하는 저비용 솔루션을 제공하는 것입니다.

이 패턴은 [파일럿 라이트 접근 방식](#)을 사용합니다. AWS에서 파일럿 라이트 DR을 구현하여 다운타임을 줄이고 비즈니스 연속성을 유지할 수 있습니다. 파일럿 라이트 접근 방식은 프로덕션 환경과 동기화된 SAP 시스템 및 스탠바이 Db2 데이터베이스를 포함하여 AWS에서 최소 DR 환경을 설정하는 데 중점을 둡니다.

이 솔루션은 확장이 가능합니다. 필요에 따라 전체 규모 재해 복구 환경으로 확장할 수 있습니다.

사전 조건 및 제한 사항

사전 조건

- Amazon Elastic Compute Cloud(Amazon EC2) 인스턴스에서 실행되는 SAP 인스턴스
- IBM Db2 데이터베이스
- SAP 제품 가용성 매트릭스(PAM)에서 지원하는 운영 체제
- 운영 및 스탠바이 데이터베이스 호스트의 서로 다른 물리적 데이터베이스 호스트 이름
- [교차 리전 복제\(CRR\)](#)를 지원하는 각 AWS 리전의 Amazon Simple Storage Service(S3) 버킷

제품 버전

- IBM Db2 데이터베이스 버전 11.5.7 이상

아키텍처

대상 기술 스택

- Amazon EC2
- Amazon Simple Storage Service(S3)
- Amazon Virtual Private Cloud(VPC 피어링)
- Amazon Route 53
- IBM Db2 고가용성 재해 복구(HADR)

대상 아키텍처

이 아키텍처는 Db2를 데이터베이스 플랫폼으로 사용하여 SAP 워크로드용 DR 솔루션을 구현합니다. 프로덕션 데이터베이스는 AWS 리전 1에 배포되고 스탠바이 데이터베이스는 두 번째 리전에 배포됩니다. 스탠바이 데이터베이스를 DR 시스템이라고 합니다. Db2 데이터베이스는 다중 스탠바이 데이터베이스(최대 3개)를 지원합니다. Db2 HADR을 사용하여 DR 데이터베이스를 설정하고 프로덕션 및 스탠바이 데이터베이스 간의 로그 전달을 자동화합니다.

리전 1을 사용할 수 없게 될 정도의 재해가 발생하는 경우 DR 리전의 스탠바이 데이터베이스가 프로덕션 데이터베이스 역할을 대신합니다. Recovery Time Objective(RTO) 요구 사항을 충족하기 위해 SAP 애플리케이션 서버를 미리 구축하거나 [AWS Elastic Disaster Recovery](#) 또는 Amazon Machine Image(AMI)를 사용하여 구축할 수 있습니다. 이 패턴은 AMI를 사용합니다.

Db2 HADR은 프로덕션-스탠바이 설정을 구현합니다. 이 경우 프로덕션이 기본 서버 역할을 하고 모든 사용자가 이 서버에 연결됩니다. 모든 트랜잭션은 TCP/IP를 사용하여 스탠바이 서버로 전송되는 로그 파일에 기록됩니다. 스탠바이 서버는 전송된 로그 레코드를 롤포워드하여 로컬 데이터베이스를 업데이트하므로 프로덕션 서버와 동기화된 상태를 유지하는 데 도움이 됩니다.

VPC 피어링은 프로덕션 지역과 DR 지역의 인스턴스가 서로 통신할 수 있도록 하는 데 사용됩니다. Amazon Route 53은 최종 사용자를 인터넷 애플리케이션으로 라우팅합니다.

1. 리전 1에 애플리케이션 서버의 [AMI를 생성](#)하고 리전 2에 [AMI를 복사](#)합니다. 재해 발생 시 AMI를 사용하여 리전 2에서 서버를 시작합니다.

2. 프로덕션 데이터베이스(리전 1)와 스탠바이 데이터베이스(리전 2) 간에 Db2 HADR 복제를 설정합니다.
3. 재해 발생 시 프로덕션 인스턴스와 일치하도록 EC2 인스턴스 유형을 변경합니다.
4. 리전 1에서는 LOGARCHMETH1이 db2remote: S3 path로 설정됩니다.
5. 리전 2에서는 LOGARCHMETH1이 db2remote: S3 path로 설정됩니다.
6. 교차 리전 복제는 S3 버킷 간에 수행됩니다.

도구

서비스

- [Amazon Elastic Compute Cloud\(Amazon EC2\)](#)는 AWS 클라우드에서 확장 가능한 컴퓨팅 용량을 제공합니다. 필요한 만큼 가상 서버를 시작하고 빠르게 스케일 업하거나 스케일 다운할 수 있습니다.
- [Amazon Route 53](#)는 가용성과 확장성이 뛰어난 DNS 웹 서비스입니다.
- [Amazon Simple Storage Service\(S3\)](#)는 원하는 양의 데이터를 저장, 보호 및 검색하는 데 도움이 되는 클라우드 기반 객체 스토리지 서비스입니다.
- [Amazon Virtual Private Cloud\(VPC\)](#)를 이용하면 사용자가 정의한 가상 네트워크로 AWS 리소스를 시작할 수 있습니다. 이 가상 네트워크는 사용자의 자체 데이터 센터에서 운영하는 기존 네트워크와 유사하며 AWS의 확장 가능한 인프라를 사용한다는 이점이 있습니다. 이 패턴은 [VPC 피어링](#)을 사용합니다.

모범 사례

- 네트워크는 HADR 복제 모드를 결정하는 데 중요한 역할을 합니다. AWS 리전 전반의 DR의 경우 Db2 HADR ASYNC 또는 SUPERASYNC 모드를 사용하는 것이 좋습니다.
- Db2 HADR의 복제 모드에 대한 자세한 내용은 [IBM 설명서](#)를 참조하십시오.
- AWS Management Console 또는 AWS Command Line Interface(AWS CLI)를 사용하여 기존 SAP 시스템의 [새로운 AMI를 생성](#)할 수 있습니다. 그런 다음 AMI를 사용하여 기존 SAP 시스템을 복구하거나 클론을 생성할 수 있습니다.
- [AWS Systems Manager Automation](#)은 EC2 인스턴스 및 기타 AWS 리소스의 일반적인 유지 관리 및 배포 작업에 도움을 줄 수 있습니다.
- AWS는 AWS의 인프라 및 애플리케이션을 모니터링하고 관리할 수 있는 여러 기본 서비스를 제공합니다. Amazon CloudWatch 및 AWS와 같은 서비스를 사용하여 각각 기본 인프라 및 API 작업을

CloudTrail 모니터링할 수 있습니다. 자세한 내용은 [SAP on AWS-Pacemaker가 포함된 IBM Db2 HADR](#)을 참조하십시오.

에픽

환경 준비

작업	설명	필요한 기술
시스템과 로그를 확인합니다.	<ol style="list-style-type: none"> 1. Db2 시스템의 프로덕션 SAP가 설정되었는지 확인합니다. 2. 로그 백업이 켜져 있고 S3 버킷에 로그를 저장하도록 구성되어 있는지 확인합니다. 이는 Db2 파라미터 LOGARCHMETH1 으로 확인할 수 있습니다. 3. 추가 애플리케이션 서버의 AMI를 생성합니다. 	AWS 관리자, SAP Basis 관리자

서버 및 복제 설정

작업	설명	필요한 기술
SAP 및 데이터베이스 서버를 생성합니다.	<ol style="list-style-type: none"> 1. DR 지역용 인프라를 배포하려면 AWS CloudFormation 스크립트를 사용하거나 프로덕션 인스턴스의 AMI를 사용하십시오. 파일럿 라이트 접근 방식의 일환으로 프로덕션 인스턴스와 동일한 계열의 더 작은 EC2 인스턴스를 사용할 수 있습니다. 예를 들어 프로덕션 인스턴스 	SAP Basis 관리자

작업	설명	필요한 기술
	<p>유형이 r6i.12xlarge 인 경우 해당 r6i.xlarge 인스턴스 유형을 DR 빌드에 사용할 수 있습니다. 하지만 프로덕션 데이터베이스 백업을 복원하려면 DR 인스턴스에 동일한 스토리지 용량을 할당해야 합니다.</p> <ol style="list-style-type: none"> 2. /sapmnt/<SID>/ 에 대한 Amazon Elastic File System(Amazon EFS) 마운트 포인트를 생성하고 기본 시스템에서 복제되도록 설정되어 있는지 확인합니다. 3. 프로덕션 시스템에서 전체 데이터베이스 백업(온라인 또는 오프라인)을 가져옵니다. 이 백업을 사용하여 DR 데이터베이스를 구축합니다. 4. DR 시스템에서 SAP 소프트웨어 프로비저닝 관리자 (SWPM) 시스템 복사 방법과 함께 HA/DR 용도로 백업/복원을 포함한 시스템 복사 방법 사용하여 DR SAP 시스템을 구축합니다. 5. SWPM의 요청에 따라 프로덕션에서 가져온 백업으로 DR의 데이터베이스를 복원합니다. DR 데이터베이스는 롤포워드 보류 상태가 됩니다. 	

작업	설명	필요한 기술
	<p>롤포워드 보류 상태는 전체 백업이 복원된 후에 기본적으로 설정됩니다. 롤포워드 보류 상태는 데이터베이스를 복원하는 중이며 일부 변경 사항을 적용해야 할 수 있음을 나타냅니다. 자세한 내용은 IBM 설명서를 참조하십시오.</p>	

작업	설명	필요한 기술
구성을 확인합니다.	<p>1. HADR에 대한 로그 아카이빙을 설정하려면 프로덕션 데이터베이스와 DR 데이터베이스 모두 모든 로그 아카이브 위치에서 로그를 자동으로 검색할 수 있어야 합니다. DR 데이터베이스의 LOGARCHMETH1 파라미터가 프로덕션 데이터베이스의 파라미터와 동일한 위치로 설정되어 있는지 확인합니다. 리전 제한으로 인해 동일한 위치에 액세스할 수 없는 경우 DR 시스템이 기본 시스템에서 자동으로 로그를 가져올 수 있는지 확인합니다.</p> <p>2. 데이터베이스 복제 활성화를 위해 TCP/IP 포트를 활성화하려면 다음 두 항목을 추가하여 프로덕션 및 DR 호스트에서 /etc/services 를 수정하십시오. 코드에서 <SID>는 Db2 데이터베이스의 시스템 ID(SID)를 나타냅니다(예: PR1).</p> <pre data-bbox="630 1503 1029 1780"> <SID>_HADR_1 55001/tcp # DB2 HADR Port1 <SID>_HADR_2 55002/tcp # DB2 HADR Port2 </pre>	AWS 관리자, SAP Basis 관리자

작업	설명	필요한 기술
	<p>두 포트 모두 기본 데이터베이스와 스탠바이 데이터베이스 간의 인바운드 및 아웃바운드 트래픽을 허용하는지 확인합니다.</p> <p>3. 프로덕션 및 DR 호스트에서 <code>/etc/hosts</code> 를 점검하여 프로덕션 호스트와 스탠바이 호스트 모두의 호스트 이름이 올바른 IP 주소를 가리키는지 확인합니다.</p>	

작업	설명	필요한 기술
<p>프로덕션 DB에서 DR DB로의 복제를 설정합니다(ASYNC 모드 사용).</p>	<p>1. 프로덕션 데이터베이스에서 다음 명령을 실행하여 파라미터를 업데이트합니다.</p> <pre data-bbox="634 394 1029 1665"> db2 UPDATE DB CFG FOR <SID> USING HADR_LOCAL_HOST HOST1 db2 UPDATE DB CFG FOR <SID> USING HADR_LOCAL_SVC <SID>_HADR_1 db2 UPDATE DB CFG FOR <SID> USING HADR_REMOTE_HOST HOST2 db2 UPDATE DB CFG FOR <SID> USING HADR_REMOTE_SVC <SID>_HADR_2 db2 UPDATE DB CFG FOR <SID> USING HADR_REMOTE_INST db2<sid> db2 UPDATE DB CFG FOR <SID> USING HADR_TIMEOUT 120 db2 UPDATE DB CFG FOR <SID> USING HADR_SYNC_MODE ASYNC db2 UPDATE DB CFG FOR <SID> USING HADR_SPOOL_LIMIT 1000 db2 UPDATE DB CFG FOR <SID> USING HADR_PEER_WINDOW 240 db2 UPDATE DB CFG FOR <SID> USING indexrec RESTART logindexb uild ON </pre> <p>2. DR 데이터베이스에서 다음 명령을 실행하여 파라미터를 업데이트합니다.</p>	<p>SAP Basis 관리자</p>

작업	설명	필요한 기술
	<pre> db2 UPDATE DB CFG FOR <SID> USING HADR_LOCA L_HOST HOST2 db2 UPDATE DB CFG FOR <SID> USING HADR_LOCA L_SVC <SID>_HADR_2 db2 UPDATE DB CFG FOR <SID> USING HADR_REMO TE_HOST HOST1 db2 UPDATE DB CFG FOR <SID> USING HADR_REMO TE_SVC <SID>_HADR_1 db2 UPDATE DB CFG FOR <SID> USING HADR_REMO TE_INST db2<sid> db2 UPDATE DB CFG FOR <SID> USING HADR_TIME OUT 120 db2 UPDATE DB CFG FOR <SID> USING HADR_SYNC MODE ASYNC db2 UPDATE DB CFG FOR <SID> USING HADR_SPOO L_LIMIT 1000 db2 UPDATE DB CFG FOR <SID> USING HADR_PEER _WINDOW 240 db2 UPDATE DB CFG FOR <SID> USING indexrec RESTART logindexb uild ON </pre> <p>이러한 파라미터는 두 데이터베이스 모두에 HADR 관련 정보를 제공하는 데 필요합니다. Db2 데이터베이스에서 HADR은 이전에 설정된 각 파라미터의 값을 기반으로 활성화됩니다. 이러한</p>	

작업	설명	필요한 기술
	<p>파라미터에 대한 자세한 내용은 IBM 설명서를 참조하십시오.</p> <p>3. 다음 명령을 사용하여 새로 생성된 스탠바이 데이터베이스에서 먼저 HADR을 시작합니다.</p> <pre data-bbox="630 579 1029 779">db2 deactivate db <SID> db2 start hadr on db <SID> as standby</pre> <p>4. 다음 명령을 사용하여 프로덕션 데이터베이스에서 HADR을 시작합니다.</p> <pre data-bbox="630 961 1029 1161">db2 deactivate db <SID> db2 start hadr on db <SID> as primary</pre> <p>5. 프로덕션 및 스탠바이 Db2 데이터베이스가 동기화되어 있고 로그 전달이 진행 중인 지 확인합니다.</p> <p>HADR 복제 상태를 모니터링하려면 다음 db2pd 명령을 사용합니다.</p> <pre data-bbox="630 1566 1029 1646">db2pd -d <SID> -hadr</pre> <p>HADR 모니터링에 대한 자세한 내용은 IBM 설명서를 참조하십시오.</p>	

DR 장애 조치 작업 테스트

작업	설명	필요한 기술
DR 테스트를 위한 프로덕션 비즈니스 다운타임을 계획합니다.	DR 장애 조치 시나리오를 테스트하려면 프로덕션 환경에서 필요한 비즈니스 다운타임을 계획해야 합니다.	SAP Basis 관리자
테스트 사용자를 생성합니다.	DR 호스트에서 검증할 수 있는 테스트 사용자(또는 모든 테스트 변경 사항)를 생성하여 DR 장애 조치 후 로그 복제를 확인합니다.	SAP Basis 관리자
콘솔에서 프로덕션 EC2 인스턴스를 중지합니다.	이 단계에서는 재해 시나리오를 모방하기 위해 비정상 종료가 시작됩니다.	AWS 시스템 관리자
요구 사항에 맞게 DR EC2 인스턴스를 스케일 업합니다.	<p>EC2 콘솔에서 DR 리전의 인스턴스 유형을 변경합니다.</p> <ol style="list-style-type: none"> 인스턴스 중지: 인스턴스가 실행 중인 경우 인스턴스 유형을 변경하기 전에 인스턴스를 중지해야 합니다. EC2 콘솔에서 인스턴스를 선택하고 중지를 선택합니다. 인스턴스 유형 수정: EC2 콘솔에서 인스턴스를 선택하고 작업, 인스턴스 설정, 인스턴스 유형 변경을 선택합니다. 기본 인스턴스와 일치하는 인스턴스 유형을 선택하고 적용을 선택합니다. 인스턴스 시작: 인스턴스 유형 변경이 완료되면 인스턴 	SAP Basis 관리자

작업	설명	필요한 기술
	<p>스를 선택하고 시작을 선택하여 EC2 콘솔에서 인스턴스를 시작합니다.</p> <p>4. Db2 데이터베이스를 삭제하려면 다음 명령을 사용합니다.</p> <pre data-bbox="634 531 1027 688">db2start db2 start HADR on db <SID> as standby</pre>	

작업	설명	필요한 기술
인계를 시작합니다.	<p>DR 시스템(host2)에서 인계 프로세스를 시작하고 DR 데이터베이스를 기본 데이터베이스로 불러옵니다.</p> <pre data-bbox="594 443 1029 562">db2 takeover hadr on database <SID> by force</pre> <p>선택적으로 다음 파라미터를 설정하여 인스턴스 유형에 따라 데이터베이스 메모리 할당을 자동으로 조정할 수 있습니다. Db2 데이터베이스에 할당할 메모리 전용 부분에 따라 INSTANCE_MEMORY 값을 결정할 수 있습니다.</p> <pre data-bbox="594 1010 1029 1482">db2 update db cfg for <SID> using INSTANCE_ MEMORY <FIXED VALUE> IMMEDIATE; db2 get db cfg for <SID> grep -i DATABASE_ MEMORY AUTOMATIC IMMEDIATE; db2 update db cfg for <SID> using self_tuni ng_mem ON IMMEDIATE;</pre> <p>다음 명령을 사용하여 문제를 확인합니다.</p> <pre data-bbox="594 1644 1029 1732">db2 get db cfg for <SID> grep -i MEMORY</pre>	SAP Basis 관리자

작업	설명	필요한 기술
	<pre>db2 get db cfg for <SID> grep -i self_tuning_mem</pre>	
DR 리전에서 SAP용 애플리케이션 서버를 시작합니다.	프로덕션 시스템에서 만든 AMI를 사용하여 DR 리전에서 새로운 추가 애플리케이션 서버를 시작합니다.	SAP Basis 관리자
SAP 애플리케이션을 시작하기 전에 검증을 수행합니다.	<ol style="list-style-type: none"> 1. /etc/hosts 및 /etc/fstab 항목을 검증합니다. 2. DR 시스템에 /sapmnt/<SID>/ 를 마운트합니다. 3. DR 파일 시스템 /sapmnt/<SID>/ 가 프로덕션 /sapmnt/<SID>/ 와 동기화되었는지 확인합니다. 4. <sid>adm 사용자로 로그인하여 R3trans -d를 실행하고 trans.log 파일의 출력을 확인합니다. trans.log 파일은 R3trans -d 명령을 실행한 위치와 동일한 위치에 생성됩니다. 	AWS 관리자, SAP Basis 관리자

작업	설명	필요한 기술
DR 시스템에서 SAP 애플리케이션을 시작합니다.	<p><sid>adm 사용자를 사용하여 DR 시스템에서 SAP 애플리케이션을 시작합니다. XX가 SAP ABAP SAP Central Services(ASCS) 서버의 인스턴스 번호를 나타내는 다음의 코드를 사용하십시오. YY는 SAP 애플리케이션 서버의 인스턴스 번호를 나타냅니다.</p> <pre>sapcontrol -nr XX - function StartService <SID> sapcontrol -nr XX - function StartSystem sapcontrol -nr YY - function StartService <SID> sapcontrol -nr YY - function StartSystem</pre>	SAP Basis 관리자
SAP 검증을 수행합니다.	이는 증거를 제공하거나 DR 리전으로의 데이터 복제 성공 여부를 확인하기 위한 DR 테스트로 수행됩니다.	테스트 엔지니어

DR 파일백 작업 수행

작업	설명	필요한 기술
운영 SAP 및 데이터베이스 서버를 시작합니다.	콘솔에서 프로덕션 시스템의 SAP와 데이터베이스를 호스팅하는 EC2 인스턴스를 시작합니다.	SAP Basis 관리자

작업	설명	필요한 기술
<p>프로덕션 데이터베이스를 시작하고 HADR을 설정합니다.</p>	<p>프로덕션 시스템(host1)에 로그인하고 다음 명령을 사용하여 DB가 복구 모드에 있는지 확인합니다.</p> <pre>db2start db2 start HADR on db P3V as standby db2 connect to <SID></pre> <p>HADR 상태가 connected 인지 확인합니다. 복제 상태는 peer여야 합니다.</p> <pre>db2pd -d <SID> -hadr</pre> <p>데이터베이스가 일관되지 않고 connected 및 peer 상태가 아닌 경우 (host1의) 데이터베이스를 (host2 DR 리전의) 현재 활성 데이터베이스와 동기화하려면 백업 및 복원이 필요할 수 있습니다. 이 경우 host2 DR 리전의 데이터베이스에서 host1 프로덕션 리전의 데이터베이스로 DB 백업을 복원합니다.</p>	<p>SAP Basis 관리자</p>

작업	설명	필요한 기술
<p>데이터베이스를 프로덕션 리전으로 페일백합니다.</p>	<p>일반적인 business-as-usual 시나리오에서 이 단계는 예정된 다운타임 내에 수행됩니다. DR 시스템에서 실행 중인 애플리케이션이 중지되고 데이터베이스가 프로덕션 리전(리전 1)으로 페일백되어 프로덕션 리전에서 작업이 재개됩니다.</p> <ol style="list-style-type: none"> 1. DR 리전의 SAP 애플리케이션 서버에 로그인하고 SAP 애플리케이션을 중지합니다. 2. DR 시스템에서 /sapmnt/<SID> 를 마운트 해제하여 변경 내용이 프로덕션 시스템의 /sapmnt/<SID> 에 역복제되도록 합니다. 3. 프로덕션 리전의 데이터베이스 서버(host1)에 로그인하여 인계를 수행합니다. <div style="border: 1px solid #ccc; border-radius: 10px; padding: 10px; margin: 10px 0;"> <pre>db2 takeover hadr on database <SID></pre> </div> <ol style="list-style-type: none"> 4. HADR 상태 확인: HADR_ROLE 은 host1의 PRIMARY 및 host2의 StandBy여야 합니다. <div style="border: 1px solid #ccc; border-radius: 10px; padding: 10px; margin: 10px 0;"> <pre>db2pd -d <SID> -hadr</pre> </div>	<p>SAP Basis 관리자</p>

작업	설명	필요한 기술
SAP 애플리케이션을 시작하기 전에 검증을 수행합니다.	<ol style="list-style-type: none"> 1. /etc/hosts 및 /etc/fstab 항목을 검증합니다. 2. 프로덕션 시스템에 /sapmnt/<SID>/ 를 마운트합니다. 3. DR 시스템 /sapmnt/<SID>/ 와 동기화되어 있는지 확인합니다. 4. <sid>adm 사용자에게 로그인하여 R3trans -d를 실행하고 trans.log 파일의 출력을 확인합니다. trans.log 파일은 R3trans -d 명령을 실행한 위치와 동일한 위치에 생성됩니다. 	AWS 관리자, SAP Basis 관리자

작업	설명	필요한 기술
<p>SAP 애플리케이션을 시작합니다.</p>	<p>1. <sid>adm 사용자를 사용하여 프로덕션 시스템에서 SAP 애플리케이션을 시작합니다. 다음 코드를 사용하십시오. 여기서 XX는 SAP ASCS 서버의 인스턴스 번호를 나타내며 YY는 SAP 애플리케이션 서버의 인스턴스 번호를 나타냅니다.</p> <pre style="border: 1px solid #ccc; border-radius: 10px; padding: 10px; margin: 10px 0;"> sapconrol -nr XX - function StartService <SID> sapconrol -nr XX - function StartSystem sapconrol -nr YY - function StartService <SID> sapconrol -nr YY - function StartSystem </pre> <p>2. 애플리케이션 서버를 사용할 수 있는지 확인하려면 SAP에 로그인하고 SICK 및 SM51 트랜잭션을 사용하여 검사를 수행하십시오.</p>	<p>SAP Basis 관리자</p>

문제 해결

문제	Solution
<p>HADR 관련 문제를 해결하기 위한 주요 로그 파일 및 명령</p>	<ul style="list-style-type: none"> • db2 get db cfg grep -i hadr • db2pd -d sid -hadr

문제	Solution
	<ul style="list-style-type: none"> • Db2diag.log (이 파일은 일반적으로 db2dump 디렉터리 내에 있으며 db2dump 경로는 파라미터 DIAGPATH로 정의됩니다.)
Db2 UDB의 HADR 문제 해결을 위한 SAP 노트	SAP Note 1154013-DB6: HADR 환경에서의 DB 문제를 참조하십시오. (이 노트에 액세스하려면 SAP 포털 보안 인증 정보가 필요합니다.)

관련 리소스

- [AWS의 Db2 데이터베이스에 대한 재해 복구 접근 방식](#)(블로그 게시물)
- [SAP on AWS-Pacemaker를 포함한 IBM Db2 HADR](#)
- [DB2 데이터베이스 간 HADR 복제를 설정하는 단계별 절차](#)
- [Db2 HADR Wiki](#)

추가 정보

이 패턴을 사용하여 Db2 데이터베이스에서 실행되는 SAP 시스템에 대한 재해 복구 시스템을 설정할 수 있습니다. 재해 상황에서 비즈니스는 정의된 Recovery Time Objective(RTO) 및 Recovery Point Objective(RPO) 요구 사항 내에서 업무를 계속할 수 있어야 합니다.

- RTO는 서비스 중단과 서비스 복원 사이의 허용 가능한 최대 지연 시간입니다. 이는 서비스를 이용할 수 없을 때 허용 가능한 기간으로 간주되는 기간을 결정합니다.
- RPO는 마지막 데이터 복구 시점 이후 허용되는 최대 시간입니다. 이에 따라 마지막 복구 시점과 서비스 중단 시점 사이에 허용 가능한 데이터 손실이 결정됩니다.

HADR과 관련된 자주 묻는 질문은 [SAP note #1612105-DB6: Db2 고가용성 재해 복구\(HADR\)에 대한 자주 묻는 질문](#)을 참조하십시오. (이 노트에 액세스하려면 SAP 포털 보안 인증 정보가 필요합니다.)

활성 대기 데이터베이스를 사용하여 Amazon RDS Custom에서 Oracle E-Business Suite를 위한 HA/DR 아키텍처를 설정합니다.

작성자: Simon Cunningham(AWS) 및 Nitin Saxena

환경: 프로덕션

기술: 데이터베이스, 인프라

워크로드: Oracle

AWS 서비스: Amazon RDS

요약

이 패턴은 다른 Amazon Web Services(AWS) 가용 영역에 Amazon RDS Custom 읽기 전용 복제본 데이터베이스를 설정하고 이를 활성 대기 데이터베이스로 변환하여 Amazon Relational Database Service(RDS) 사용자 정의 고가용성(HA) 및 재해 복구(DR)를 기반으로 Oracle E-Business 솔루션을 설계하는 방법을 설명합니다. Amazon RDS Custom 읽기 전용 복제본의 생성은 AWS Management Console을 통해 완전히 자동화됩니다.

이 패턴에서는 HA/DR 아키텍처의 일부일 수도 있는 추가 애플리케이션 티어와 공유 파일 시스템을 추가하는 단계를 설명하지 않습니다. 이러한 항목에 대한 자세한 내용은 Oracle 지원 노트 1375769.1, 1375670.1 및 1383621.1(섹션 5, 고급 클로닝 옵션)을 참조하세요. (액세스하려면 [Oracle Support](#) 계정이 필요합니다.)

Amazon Web Services(AWS)에서 E-Business Suite 시스템을 단일 계층, 단일 AZ 아키텍처로 마이그레이션하려면 [Oracle E-Business Suite를 Amazon RDS Custom으로 마이그레이션](#)하는 패턴을 참조하세요.

Oracle E-Business Suite는 재무, 인사, 공급망, 제조 등 전사적 프로세스를 자동화하기 위한 전사적 자원 계획(ERP) 솔루션입니다. 클라이언트, 애플리케이션, 데이터베이스의 3계층 아키텍처를 갖추고 있습니다. 이전에는 자체 관리형 [Amazon Elastic Compute Cloud\(Amazon EC2\) 인스턴스](#)에서 E-Business Suite 데이터베이스를 실행해야 했지만 이제는 [Amazon RDS Custom](#)의 이점을 활용할 수 있습니다.

사전 조건 및 제한 사항

사전 조건

- Amazon RDS Custom에 설치된 기존 E-Business Suite, [Oracle E-Business Suite를 Amazon RDS Custom으로 마이그레이션하는 패턴](#) 참조
- 읽기 전용 복제본을 읽기 전용으로 변경하고 이를 사용하여 보고 기능을 예비 복제본으로 오프로드하려는 경우, [Oracle Active Data Guard 데이터베이스 라이선스](#)(Oracle Technology 상용 가격 목록 참조)

제한 사항

- [Amazon RDS Custom의 Oracle 데이터베이스](#)에 대한 제한 사항 및 지원되지 않는 구성
- [Amazon RDS Custom for Oracle 읽기 전용 복제본](#)에 대한 제한 사항

제품 버전

Amazon RDS Custom에서 지원하는 Oracle Database 버전 및 인스턴스 클래스에 대한 자세한 내용은 [Oracle용 Amazon RDS Custom의 요구 사항 및 제한 사항](#)을 참조하십시오.

아키텍처

다음 다이어그램은 액티브/패시브 설정에 여러 가용 영역 및 애플리케이션 티어를 포함하는 AWS 기반 E-Business Suite의 대표적인 아키텍처를 보여줍니다. 데이터베이스는 Amazon RDS Custom DB 인스턴스와 Amazon RDS Custom 읽기 전용 복제본을 사용합니다. 읽기 전용 복제본은 Active Data Guard를 사용하여 다른 가용 영역으로 복제합니다. 또한 읽기 전용 복제본을 사용하여 기본 데이터베이스의 읽기 트래픽을 오프로드하고 보고 목적으로 사용할 수 있습니다.

자세한 내용은 Amazon RDS 문서의 [Amazon RDS Custom for Oracle의 읽기 적용 복제본 작업](#)을 참조하세요.

Amazon RDS Custom 읽기 전용 복제본은 마운트된 상태로 기본적으로 생성됩니다. 하지만 일부 읽기 전용 워크로드를 대기 데이터베이스로 오프로드하여 기본 데이터베이스의 부하를 줄이려면 [에픽](#) 섹션의 단계에 따라 마운트된 복제본의 모드를 읽기 전용으로 수동으로 변경할 수 있습니다. 일반적인 사용 사례는 대기 데이터베이스에서 보고서를 실행하는 것입니다. 읽기 전용으로 변경하려면 활성 대기 데이터베이스 라이선스가 필요합니다.

AWS에서 읽기 전용 복제본을 생성하면 시스템은 기본적으로 Oracle Data Guard 브로커를 사용합니다. 이 구성은 다음과 같이 최대 성능 모드에서 자동으로 생성되고 설정됩니다.

```
DGMGRL> show configuration
```

```
Configuration - rds_dg
Protection Mode: MaxPerformance
Members:
vis_a - Primary database
vis_b - Physical standby database
Fast-Start Failover: DISABLED
Configuration Status:
SUCCESS (status updated 58 seconds ago)
```

도구

서비스

- [Amazon RDS Custom for Oracle](#)은 기본 운영 체제 및 데이터베이스 환경에 액세스해야 하는 레거시, 커스텀 및 패키지 애플리케이션을 위한 관리형 데이터베이스 서비스입니다. 이는 데이터베이스 관리 작업과 운영을 자동화하는 동시에 데이터베이스 관리자로서 데이터베이스 환경과 운영 체제에 액세스하고 사용자 정의할 수 있도록 해줍니다.

기타 도구

- Oracle Data Guard는 Oracle 대기 데이터베이스를 생성하고 관리하는 데 도움이 되는 도구입니다. 이 패턴은 Oracle Data Guard를 사용하여 Amazon RDS Custom에 활성 대기 데이터베이스를 설정합니다.

에픽

읽기 전용 복제본 생성

작업	설명	필요한 기술
Amazon RDS Custom DB 인스턴스의 읽기 전용 복제본을 생성합니다.	읽기 전용 복제본을 생성하려면 Amazon RDS 문서 의 지침에 따라 생성한 Amazon RDS Custom DB 인스턴스(사전 요구 사항 섹션 참조)를 원본 데이터베이스로 사용하세요. 기본적으로 Amazon RDS Custom 읽기 전용 복제본은 물	DBA

작업	설명	필요한 기술
	<p>리적 예비 복제본으로 생성되며 마운트된 상태입니다. 이는 Oracle Active Data Guard 라이선스 규정 준수를 보장하기 위한 것입니다. 읽기 전용 복제본을 읽기 전용 모드로 전환하려면 다음 단계를 따르세요.</p>	

읽기 전용 복제본을 읽기 전용 액티브 스탠바이로 변경

작업	설명	필요한 기술
<p>Amazon RDS Custom 읽기 전용 복제본에 연결합니다.</p>	<p>다음 명령을 사용하여 물리적 대기 데이터베이스를 활성 대기 데이터베이스로 변환합니다.</p> <p>중요: 이 명령에는 Oracle 액티브 스탠바이 라이선스가 필요합니다. 라이선스를 받으려면 Oracle 담당자에게 문의하세요.</p> <pre data-bbox="594 1312 1029 1881"> \$ sudo su - rdsdb -bash-4.2\$ sql SQL> select process,status,sequence# from v \$managed_standby; PROCESS STATUS SEQUENCE# ----- ARCH CLOSING 3956 ARCH CONNECTED 0 </pre>	<p>DBA</p>

작업	설명	필요한 기술
	<pre> ARCH CLOSING 3955 ARCH CLOSING 3957 RFS IDLE 0 RFS IDLE 3958 MRP0 APPLYING_LOG 3958 SQL> select name, database_role, open_mode from v \$database; NAME DATABASE_ ROLE OPEN_MODE ----- ----- ----- VIS PHYSICAL STANDBY MOUNTED SQL> alter database recover managed standby database cancel; Database altered. Open the standby database SQL> alter database open; Database altered. SQL> select name, database_role, open_mode from v \$database; NAME DATABASE_ ROLE OPEN_MODE ----- ----- ----- </pre>	

작업	설명	필요한 기술
	<pre> VIS PHYSICAL STANDBY READ ONLY </pre>	
<p>실시간 로그 적용으로 미디어 복구를 시작합니다.</p>	<p>실시간 로그 적용 기능을 활성화하려면 다음 명령을 사용합니다. 이는 대기 데이터베이스 (읽기 전용 복제본)를 활성 대기 데이터베이스로 변환하고 유효성을 검사하므로 읽기 전용 쿼리를 연결하고 실행할 수 있습니다.</p> <pre> SQL> alter database recover managed standby database using current logfile disconnect from session; Database altered </pre>	<p>DBA</p>
<p>데이터베이스 상태를 확인하세요.</p>	<p>데이터베이스 상태를 확인하려면 다음 명령을 사용하세요.</p> <pre> SQL> select name, database_role, open_mode from v \$database; NAME DATABASE_ROLE OPEN_MODE ----- ----- ----- VIS PHYSICAL STANDBY READ ONLY WITH APPLY </pre>	<p>DBA</p>

작업	설명	필요한 기술
재실행 적용 모드를 확인하세요.	<p>재실행두 적용 모드를 점검하세요.</p> <pre> SQL> select process,s tatus,sequence# from v \$managed_standby; PROCESS STATUS SEQUENCE# ----- ARCH CLOSING 3956 ARCH CONNECTED 0 ARCH CLOSING 3955 ARCH CLOSING 3957 RFS IDLE 0 RFS IDLE 3958 MRP0 APPLYING_LOG 3958 SQL> select open_mode from v\$database; OPEN_MODE ----- READ ONLY WITH APPLY </pre>	DBA

관련 리소스

- [Oracle E-Business Suite를 Amazon RDS Custom으로 마이그레이션\(AWS 권장 가이드\)](#)
- [Amazon RDS Custom 작업\(Amazon RDS 설명서\)](#)
- [Amazon RDS Custom for Oracle의 읽기 전용 복제본으로 작업\(Amazon RDS 문서\)](#)
- [Amazon RDS Custom for Oracle - 데이터베이스 환경의 새로운 제어 기능\(AWS 뉴스 블로그\)](#)

- [AWS 기반 Oracle E-Business Suite 마이그레이션](#)(AWS 백서)
- [AWS 기반 Oracle E-Business Suite 아키텍처](#)(AWS 백서)

GTID를 사용하여 Amazon RDS for MySQL와 Amazon EC2의 MySQL 간에 데이터 복제를 설정합니다.

작성자: Rajesh Madiwale (AWS)

환경: PoC 또는 파일럿

기술: 데이터베이스

워크로드: 오픈 소스

요약

이 패턴은 MySQL 고유 글로벌 트랜잭션 식별자(GTID) 복제를 사용하여 MySQL DB 인스턴스용 Amazon 관계형 데이터베이스 서비스(RDS)와 Amazon Elastic Compute Cloud(Amazon EC2) 인스턴스의 MySQL 데이터베이스 간에 Amazon Web Services(AWS) 클라우드의 데이터 복제를 설정하는 방법을 설명합니다.

GTID를 사용하면 트랜잭션이 오리진 서버에서 커밋되고 복제본으로 적용될 때 트랜잭션이 식별 및 추적됩니다. 장애 조치 중에 새 복제본을 시작할 때는 로그 파일을 참조하지 않아도 됩니다.

사전 조건 및 제한 사항

사전 조건

- 활성 상태의 AWS 계정.
- Amazon Linux 인스턴스가 배포되었습니다.

제한 사항

- 이 설정을 사용하려면 내부 팀이 읽기 전용 쿼리를 실행해야 합니다.
- 소스 및 타겟 MySQL 버전이 같아야 합니다.
- 복제는 동일한 AWS 리전 및 Virtual Private Cloud(VPC)에 설정됩니다.

제품 버전

- Amazon RDS 버전 5.7.23 및 [GTID](#)를 지원하는 버전은 무엇입니까?

아키텍처

소스 기술 스택

- Amazon RDS for MySQL

대상 기술 스택

- Amazon EC2의 MySQL

대상 아키텍처

도구

서비스

- [Amazon Elastic Compute Cloud\(Amazon EC2\)](#)는 AWS 클라우드에서 확장 가능한 컴퓨팅 용량을 제공합니다. 필요한 만큼 가상 서버를 시작하고 빠르게 스케일 업하거나 스케일 다운할 수 있습니다.
- [Amazon Relational Database Service\(RDS\) for MySQL](#)는 AWS 클라우드에서 관계형 데이터베이스를 설정, 운영 및 조정하는 데 도움이 됩니다.

기타 서비스

- [전역 트랜잭션 ID\(GTIDs\)](#) are unique identifiers generated for committed MySQL transactions.
- [mysqldump](#)는 소스 데이터베이스 객체 정의 및 테이블 데이터를 재생하기 위해 실행할 수 있는 SQL 문을 생성하여 논리적 백업을 수행하는 클라이언트 유틸리티입니다.
- [mysql](#)은 MySQL을 위한 커맨드 라인 클라이언트입니다.

에픽

Amazon RDS for MySQL DB 인스턴스 생성 및 준비

작업	설명	필요한 기술
RDS for MySQL 인스턴스를 생성합니다.	RDS for MySQL 인스턴스를 생성하려면 다음 작업에서 다루는 파라미터 값을 사용하여 Amazon RDS 설명서 의 단계를 따르세요.	DBA, 엔지니어 DevOps
DB 파라미터 그룹에서 GTID 관련 설정을 활성화합니다.	Amazon RDS for MySQL DB 파라미터 그룹에서 다음 파라미터를 활성화합니다. enforce_gtid_consistency 을(를) on(으)로 설정하고, gtid-mode 을(를) on(으)로 설정합니다.	DBA
Amazon RDS for MySQL 인스턴스를 재부팅합니다.	파라미터 변경 사항을 적용하려면 재부팅해야 합니다.	DBA
사용자를 생성하고 사용자에게 복제 권한을 부여합니다.	다음 명령을 사용하여 MySQL 을 설치합니다. <pre>CREATE USER 'repl'@'%' IDENTIFIED BY 'xxxx'; GRANT REPLICATI ON slave ON *.* TO 'repl'@'%' ; FLUSH PRIVILEGES;</pre>	DBA

Amazon EC2 인스턴스에 MySQL을 설치하고 준비합니다.

작업	설명	필요한 기술
<p>Amazon Linux에 MySQL을 설치합니다.</p>	<p>다음 명령을 사용하여 MySQL을 설치합니다.</p> <pre> sudo yum update sudo wget https://dev.mysql.com/get/mysql57-community-release-el7-11.noarch.rpm sudo yum localinstall mysql57-community-release-el7-11.noarch.rpm sudo yum install mysql-community-server sudo systemctl start mysqld </pre>	DBA
<p>EC2 인스턴스에서 MySQL에 로그인하고 데이터베이스를 생성합니다.</p>	<p>데이터베이스 이름은 Amazon RDS for MySQL의 데이터베이스 이름과 동일해야 합니다. 다음 예에서 데이터베이스 이름은 replication 입니다.</p> <pre> create database replication; </pre>	DBA
<p>MySQL 구성 파일을 편집하고 데이터베이스를 다시 시작합니다.</p>	<p>다음 파라미터를 추가하여 /etc/에 있는 my.conf 파일을 편집합니다.</p> <pre> server-id=3 gtid_mode=ON enforce_gtid_consistency=0N </pre>	DBA

작업	설명	필요한 기술
	<pre>replicate-ignore-db =mysql binlog-format=ROW log_bin=mysql-bin</pre> <p>그런 다음 mysqld 서비스를 다시 시작합니다.</p> <pre>systemctl mysqld restart</pre>	

복제 설정

작업	설명	필요한 기술
Amazon RDS for MySQL 데이터베이스에서 데이터 덤프를 내보냅니다.	<p>Amazon RDS for MySQL에서 덤프를 내보내려면 다음 명령을 사용하세요.</p> <pre>mysqldump --single-transaction -h mydb.xxxxxxx.amaz naws.com -uadmin -p -- databases replication > replication-db.sql</pre>	DBA
Amazon EC2의 MySQL 데이터베이스에 있는 .sql 덤프 파일을 복원합니다.	<p>Amazon EC2에 있는 MySQL 데이터베이스로 덤프를 가져오려면 다음 명령을 사용합니다.</p> <pre>mysql -D replication -uroot -p < replicati on-db.sql</pre>	DBA
Amazon EC2에서 MySQL 데이터베이스를 복제본으로 구성합니다.	<p>복제를 시작하고 복제 상태를 확인하려면 Amazon EC2의 MySQL 데이터베이스에 로그</p>	DBA

작업	설명	필요한 기술
	<p>인하고 다음 명령을 사용합니다.</p> <pre>CHANGE MASTER TO MASTER_HOST="mydb. xxxxxxxx.amazonaws. com", MASTER_US ER="rep1", MASTER_PA SSWORD="rep123", MASTER_PORT=3306, MASTER_AUTO_POSITION = 1; START SLAVE; SHOW SLAVE STATUS\G</pre>	

관련 리소스

- [Linux 인스턴스용 Amazon EC2 사용 설명서](#)
- [MySQL Yum 리포지토리를 사용하여 리눅스에 MySQL 설치](#)
- [글로벌 트랜잭션 식별자를 사용한 복제](#)
- [Amazon RDS for MySQL GTID 기반 복제 사용](#)

Amazon RDS 기반 오라클 PeopleSoft 애플리케이션의 역할 전환 (오라클용 사용자 지정)

작성자: sampath kathirvel(AWS)

환경: 프로덕션

기술: 데이터베이스, 인프라

워크로드: Oracle

AWS 서비스: Amazon RDS

요약

Amazon Web Services (AWS) 에서 [오라클 PeopleSoft](#) 엔터프라이즈 리소스 계획 (ERP) 솔루션을 실행하려면 기본 운영 체제 (OS) 및 데이터베이스 환경에 액세스해야 하는 레거시, 사용자 지정 및 패키지 애플리케이션을 지원하는 [Amazon RDS \(Amazon RDS\)](#) 또는 Amazon RDS Custom을 사용할 수 있습니다. 마이그레이션을 계획할 때 고려해야 할 주요 요소는 AWS 권장 가이드의 [Oracle 데이터베이스 마이그레이션 전략](#)을 참조하세요.

이 패턴은 Amazon RDS Custom에서 읽기 전용 복제본 데이터베이스가 있는 기본 데이터베이스로 실행되는 PeopleSoft 애플리케이션 데이터베이스에 대해 Oracle Data Guard 전환 또는 역할 전환을 수행하는 단계를 중점적으로 다룹니다. 패턴에는 [패스트 스타트 장애 조치\(FSFO\)](#)를 구성하는 단계가 포함됩니다. 이 프로세스 중에도 Oracle Data Guard 구성의 데이터베이스는 새로운 역할로 계속 작동합니다. Oracle Data Guard 전환의 일반적인 사용 사례로는 재해 복구(DR) 훈련, 데이터베이스의 예정된 유지 관리 작업, 그리고 [Standby-First Patch Apply](#) 롤링 패치가 있습니다. 자세한 내용은 블로그 게시물, [Amazon RDS Custom에서 데이터베이스 패치 작업 중단 시간 단축](#)을 참조하세요.

사전 조건 및 제한 사항

사전 조건

- [읽기 전용 복제본 패턴을 사용하여 Amazon RDS 사용자 PeopleSoft 지정에서 Oracle에 HA 추가](#)를 완료했습니다.

제한 사항

- [Oracle용 RDS Custom의 제한 사항 및 지원되지 않는 구성](#)
- [Amazon RDS Custom for Oracle 읽기 전용 복제본](#)에 대한 제한 사항

제품 버전

- Amazon RDS Custom에서 지원하는 Oracle Database 버전에 대해서는 [Oracle용 RDS Custom](#)을 참조하세요.
- Amazon RDS Custom에서 지원하는 Oracle Database 인스턴스 클래스에 대해서는 [Oracle용 RDS Custom에 대한 DB 인스턴스 클래스 지원](#)을 참조하세요.

아키텍처

기술 스택

- Amazon RDS Custom for Oracle

대상 아키텍처

다음 다이어그램에서는 Amazon RDS Custom DB 인스턴스와 Amazon RDS 사용자 지정 읽기 전용 복제본을 보여줍니다. Oracle Data Guard는 DR에 대한 장애 조치 중에 역할 전환을 제공합니다.

PeopleSoft AWS에서 Oracle을 사용하는 대표적인 아키텍처는 [AWS에서 고가용성 PeopleSoft 아키텍처 설정](#)을 참조하십시오.

도구

서비스

- [Oracle용 Amazon RDS Custom](#)은 기본 운영 체제 및 데이터베이스 환경에 액세스해야 하는 레거시, 커스텀 및 패키지 애플리케이션을 위한 관리형 데이터베이스 서비스입니다.
- [AWS Secrets Manager](#)를 사용하면 암호를 포함하여 코드에 하드코딩된 보안 인증을 Secrets Manager에 대한 API 호출로 대체하여 프로그래밍 방식으로 암호를 검색할 수 있습니다. 이 패턴에서는 Secrets Manager에서 암호 이름 do-not-delete-rds-custom-+<<RDS Resource ID>> +-dg를 사용하여 RDS_DATAGUARD에 대한 데이터베이스 사용자 암호를 검색합니다.

기타 서비스

- [Oracle Data Guard](#)를 사용하면 대기 데이터베이스를 생성, 유지, 관리 및 모니터링할 수 있습니다. 이 패턴은 역할 전환([Oracle Data Guard 전환](#))을 위해 Oracle Data Guard 최대 성능을 사용합니다.

모범 사례

프로덕션 배포의 경우 기본 및 읽기 전용 복제본 노드와 분리된 세 번째 가용 영역에서 오퍼버 인스턴스를 시작하는 것이 좋습니다.

에픽

역할 전환 시작

작업	설명	필요한 기술
<p>기본 및 복제본 모두에 대한 데이터베이스 자동화를 일시 중지합니다.</p>	<p>RDS Custom 자동화 프레임워크가 역할 전환 프로세스를 방해하지는 않지만 Oracle Data Guard 전환 중에 자동화를 일시 중지하는 것이 좋습니다.</p> <p>RDS Custom 데이터베이스 자동화를 일시 중지했다가 다시 시작하려면 RDS Custom 자동화 일시 중지 및 재개의 지침을 준수합니다.</p>	<p>클라우드 관리자, DBA</p>
<p>Oracle Data Guard 상태를 확인합니다.</p>	<p>Oracle Data Guard 상태를 확인하려면 기본 데이터베이스에 로그인합니다. 이 패턴에는 멀티테넌트 컨테이너 데이터베이스 (CDB) 또는 비CDB 인스턴스를 사용하기 위한 코드가 포함됩니다.</p> <p>비 CDB</p> <pre data-bbox="597 1650 1026 1862">-bash-4.2\$ dgmgrl RDS_DATAGUARD@RDS_ CUSTOM_ORCL_A DGMGRL for Linux: Release 19.0.0.0.0 -</pre>	<p>DBA</p>

작업	설명	필요한 기술
	<pre> Production on Mon Nov 28 20:55:50 2022 Version 19.10.0.0.0 Copyright (c) 1982, 2019, Oracle and/or its affiliates. All rights reserved. Welcome to DGMGRL, type "help" for informati on. Password: Connected to "ORCL_A" Connected as SYSDBG. DGMGRL> show configura tion Configuration - rds_dg Protection Mode: MaxAvailability Members: orcl_a - Primary database orcl_d - Physical standby database Fast-Start Failover: Disabled Configuration Status: SUCCESS (status updated 59 seconds ago) DGMGRL> CDB CDB-bash-4.2\$ dgmgrl C##RDS_DATAGUARD@R DS_CUSTOM_RDSCDB_A DGMGRL for Linux: Release 19.0.0.0.0 - Production on Wed Jan 18 06:13:07 2023 Version 19.16.0.0.0 </pre>	

작업	설명	필요한 기술
	<pre> Copyright (c) 1982, 2019, Oracle and/or its affiliates. All rights reserved. Welcome to DGMGRL, type "help" for informati on. Password: Connected to "RDSCDB_A " Connected as SYSDBA. DGMGRL> show configura tion Configuration - rds_dg Protection Mode: MaxAvailability Members: rdsbdb_a - Primary database rdsbdb_b - Physical standby database Fast-Start Failover: Disabled Configuration Status: SUCCESS (status updated 52 seconds ago) DGMGRL> </pre>	

작업	설명	필요한 기술
인스턴스 역할을 확인합니다.	<p>AWS Management Console에 열고 Amazon RDS 콘솔로 이동합니다. 데이터베이스의 복제 섹션의 연결 및 보안 탭에서 기본 및 복제본의 인스턴스 역할을 확인합니다.</p> <p>기본 역할은 Oracle Data Guard 기본 데이터베이스와 일치해야 하고, 복제본 역할은 Oracle Data Guard 물리적 대기 데이터베이스와 일치해야 합니다.</p>	클라우드 관리자, DBA

작업	설명	필요한 기술
전환을 실시합니다.	<p>전환을 실시하려면 프라이머리 노드에서 DGMGRL로 연결합니다.</p> <p>비 CDB</p> <pre>DGMGRL> switchover to orcl_d; Performing switchover NOW, please wait... Operation requires a connection to database "orcl_d" Connecting ... Connected to "ORCL_D" Connected as SYSDG. New primary database "orcl_d" is opening... Operation requires start up of instance "ORCL" on database "orcl_a" Starting instance "ORCL"... Connected to an idle instance. ORACLE instance started. Connected to "ORCL_A" Database mounted. Database opened. Connected to "ORCL_A" Switchover succeeded, new primary is "orcl_d" DGMGRL></pre> <p>CDB</p> <pre>DGMGRL> switchover to rdscdb_b</pre>	DBA

작업	설명	필요한 기술
	<pre> Performing switchover NOW, please wait... New primary database "rdscdb_b" is opening... Operation requires start up of instance "RDSCDB" on database "rdscdb_a" Starting instance "RDSCDB"... Connected to an idle instance. ORACLE instance started. Connected to "RDSCDB_A " Database mounted. Database opened. Connected to "RDSCDB_A " Switchover succeeded , new primary is "rdscdb_b" </pre>	

작업	설명	필요한 기술
Oracle Data Guard 연결을 확인합니다.	<p>전환 후 프라이머리 노드에서 DGMGRL로의 Oracle Data Guard 연결을 확인합니다.</p> <p>비 CDB</p> <pre> DGMGRL> show configuration; Configuration - rds_dg Protection Mode: MaxAvailability Members: orcl_d - Primary database orcl_a - Physical standby database Fast-Start Failover: Disabled Configuration Status: SUCCESS (status updated 60 seconds ago) DGMGRL> DGMGRL> show configuration lag; Configuration - rds_dg Protection Mode: MaxAvailability Members: orcl_d - Primary database orcl_a - Physical standby database Transport Lag: 0 seconds (computed 0 seconds ago) Apply Lag: 0 seconds (computed 0 seconds ago) Fast-Start Failover: Disabled </pre>	DBA

작업	설명	필요한 기술
	<pre> Configuration Status: SUCCESS (status updated 44 seconds ago) DGMGRL> CDB DGMGRL> show configura tion DGMGRL> show configura tion Configuration - rds_dg Protection Mode: MaxAvailability Members: rdsbdb_b - Primary database rdsbdb_a - Physical standby database Fast-Start Failover: Disabled Configuration Status: SUCCESS (status updated 52 seconds ago) DGMGRL> DGMGRL> show configura tion lag Configuration - rds_dg Protection Mode: MaxAvailability Members: rdsbdb_b - Primary database rdsbdb_a - Physical standby database Transport Lag: 0 seconds (computed 0 seconds ago) </pre>	

작업	설명	필요한 기술
	<pre> Apply Lag: 0 seconds (computed 0 seconds ago) Fast-Start Failover: Disabled Configuration Status: SUCCESS (status updated 53 seconds ago) DGMGRL> </pre>	
<p>Amazon RDS 콘솔에서 인스턴스 역할을 확인합니다.</p>	<p>역할 전환을 한 후 Amazon RDS 콘솔에 데이터베이스 아래에 연결 및 보안 탭에 있는 복제본 섹션 아래에 새로운 역할이 표시됩니다. 복제 상태가 비어 있다가 복제 중으로 업데이트되는 데 몇 분 정도 걸릴 수 있습니다.</p>	<p>DBA</p>

FSFO 구성

작업	설명	필요한 기술
<p>전환을 재설정합니다.</p>	<p>전환을 프라이머리 노드로 다시 설정합니다.</p>	<p>DBA</p>
<p>옵저버를 설치하고 시작합니다.</p>	<p>옵저버 프로세스는 일반적으로 기본 및 대기 데이터베이스와는 다른 시스템에서 실행되고 있는 DGMGRL 클라이언트 구성 요소입니다. 옵저버용 ORACLE_HOME 설치에 Oracle Client Administrator 설치일 수도 있거나, Oracle Database Enterprise Edition</p>	<p>DBA</p>

작업	설명	필요한 기술
	<p>또는 Personal Edition을 설치할 수도 있습니다. 데이터베이스 릴리스의 옵저버 설치에 대한 자세한 내용은 옵저버 설치 및 시작을 참조하세요. 옵저버 프로세스의 고가용성을 구성하려면 다음을 수행할 수 있습니다.</p> <ul style="list-style-type: none"> • 옵저버를 실행 중인 EC2 인스턴스에 대해 EC2 인스턴스 자동 복구를 활성화합니다. OS 스타트업의 일부로 옵저버 스타트업 프로세스를 자동화해야 합니다. • EC2 인스턴스에 옵저버를 배포하고 Amazon EC2 Auto Scaling 그룹을 사이즈 일(1)로 구성합니다. EC2 인스턴스에 장애가 발생하는 경우 자동 규모 조정 그룹이 자동으로 다른 EC2 인스턴스를 가동합니다. <p>Oracle 12c Release 2 이상의 경우 최대 3명의 옵저버를 배포할 수 있습니다. 한 명의 옵저버가 기본 옵저버이고 나머지는 백업 옵저버입니다. 기본 옵저버가 실패하면 백업 옵저버 중 한 명이 기본 역할을 맡습니다.</p>	

작업	설명	필요한 기술
<p>옵저버 호스트에서 DGMGRL에 연결합니다.</p>	<p>옵저버 호스트는 기본 및 대기 데이터베이스 연결을 위한 <code>tnsnames.ora</code> 항목으로 구성됩니다. 데이터 손실이 FastStartFailoverLagLimit 구성 (초 단위) 내에 있는 한 최대 성능 보호 모드로 FSFO를 활성화할 수 있지만, 데이터 손실이 전혀 없도록 하려면 최대 가용성 보호 모드를 사용해야 합니다 (RPO=0).</p> <p>비 CDB</p> <pre>DGMGRL> show configuration; Configuration - rds_dg Protection Mode: MaxAvailability Members: orcl_a - Primary database orcl_d - Physical standby database Fast-Start Failover: Disabled Configuration Status: SUCCESS (status updated 58 seconds ago) DGMGRL> show configuration lag Configuration - rds_dg Protection Mode: MaxAvailability Members: orcl_a - Primary database orcl_d - Physical standby database</pre>	<p>DBA</p>

작업	설명	필요한 기술
	<pre> Transport Lag: 0 seconds (computed 1 second ago) Apply Lag: 0 seconds (computed 1 second ago) Fast-Start Failover: Disabled Configuration Status: SUCCESS (status updated 5 seconds ago) DGMGRL> CDB -bash-4.2\$ dgmgrl C##RDS_DATAGUARD@R DS_CUSTOM_RDSCDB_A DGMGRL for Linux: Release 19.0.0.0.0 - Production on Wed Jan 18 06:55:09 2023 Version 19.16.0.0.0 Copyright (c) 1982, 2019, Oracle and/or its affiliates. All rights reserved. Welcome to DGMGRL, type "help" for informati on. Password: Connected to "RDSCDB_A " Connected as SYSDBG. DGMGRL> show configura tion Configuration - rds_dg Protection Mode: MaxAvailability Members: rdscdb_a - Primary database </pre>	

작업	설명	필요한 기술
	<pre>rdscdb_b - Physical standby database Fast-Start Failover: Disabled Configuration Status: SUCCESS (status updated 18 seconds ago) DGMGRL></pre>	

작업	설명	필요한 기술
<p>대기 데이터베이스를 장애 조치 대상으로 수정합니다.</p>	<p>프라이머리 노드 또는 옵저버 노드에서 하나의 대기 데이터베이스로 접속합니다. (구성에 여러 대기 데이터베이스가 있을 수 있지만 지금은 하나만 연결하면 됩니다.)</p> <p>비 CDB</p> <pre>DGMGRL> edit database orcl_a set property FastStartFailoverT arget='orcl_d'; Property "faststar tfailovertarget" updated DGMGRL> edit database orcl_d set property FastStartFailoverT arget='orcl_a'; Property "faststar tfailovertarget" updated DGMGRL> show database orcl_a FastStart FailoverTarget; FastStartFailoverTar get = 'orcl_d' DGMGRL> show database orcl_d FastStart FailoverTarget; FastStartFailoverTar get = 'orcl_a' DGMGRL></pre> <p>CDB</p> <pre>DGMGRL> edit database orcl_a set property</pre>	<p>DBA</p>

작업	설명	필요한 기술
	<pre> FastStartFailoverT arget='rdscdb_b'; Object "orcl_a" was not found DGMGRL> edit database rdscdb_a set property FastStartFailoverT arget='rdscdb_b'; Property "faststar tfailovertarget" updated DGMGRL> edit database rdscdb_b set property FastStartFailoverT arget='rdscdb_a'; Property "faststar tfailovertarget" updated DGMGRL> show database rdscdb_a FastStart FailoverTarget; FastStartFailoverT arget = 'rdscdb_b' DGMGRL> show database rdscdb_b FastStart FailoverTarget; FastStartFailoverT arget = 'rdscdb_a' DGMGRL> </pre>	

작업	설명	필요한 기술
<p>FastStartFailoverThreshold DGMGRL에 연결하도록 구성하십시오.</p>	<p>Oracle 19c의 기본값은 30초이고 최소값은 6초입니다. 값이 낮으면 장애 조치 중에 Recovery Time Objective (RTO)가 단축될 가능성이 있습니다. 값이 높을수록 기본 데이터베이스에서 불필요한 장애 조치 일시적 오류가 발생할 가능성을 줄이는 데 도움이 됩니다.</p> <p>Oracle용 RDS Custom 자동화 프레임워크는 데이터베이스 상태를 모니터링하고 몇 초마다 수정 작업을 수행합니다. 따라서 10초보다 큰 값으로 설정하는 FastStartFailoverThreshold 것이 좋습니다. 다음 예제에서는 임계값을 35초로 구성합니다.</p> <p>비·CBD 또는 CDB</p> <pre>DGMGRL> edit configuration set property FastStartFailoverThreshold=35; Property "faststartfailoverthreshold" updated DGMGRL> show configuration FastStart FailoverThreshold; FastStartFailoverThreshold = '35' DGMGRL></pre>	<p>DBA</p>

작업	설명	필요한 기술
<p>기본 또는 옵저버 노드에서 DGMGRL에 연결하여 FSFO를 활성화합니다.</p>	<p>데이터베이스에 플래시백 데이터베이스가 활성화되지 않은 경우 경고 메시지 ORA-16827가 나타납니다. 선택적 플래시백 데이터베이스를 사용하면 FastStartFailoverAutoReinst ate 구성 속성이 TRUE (기본값)으로 설정된 경우 장애가 발생한 기본 데이터베이스를 장애 조치 이전의 특정 시점으로 자동으로 복원할 수 있습니다.</p> <p>비 CDB</p> <pre>DGMGRL> enable fast_start failover; Warning: ORA-16827: Flashback Database is disabled Enabled in Zero Data Loss Mode. DGMGRL> DGMGRL> show configuration Configuration - rds_dg Protection Mode: MaxAvailability Members: orcl_a - Primary database Warning: ORA-16819: fast-start failover observer not started orcl_d - (*) Physical standby database Warning: ORA-16819: fast-start failover observer not started</pre>	<p>DBA</p>

작업	설명	필요한 기술
	<pre> Fast-Start Failover: Enabled in Zero Data Loss Mode Configuration Status: WARNING (status updated 29 seconds ago) DGMGRL> CDB DGMGRL> enable fast_star t failover; Warning: ORA-16827: Flashback Database is disabled Enabled in Zero Data Loss Mode. DGMGRL> show configura tion; Configuration - rds_dg Protection Mode: MaxAvailability Members: rdscdb_a - Primary database Warning: ORA-16819 : fast-start failover observer not started rdscdb_b - (*) Physical standby database Fast-Start Failover: Enabled in Zero Data Loss Mode Configuration Status: WARNING (status updated 11 seconds ago) DGMGRL> </pre>	

작업	설명	필요한 기술
<p>FSFO 모니터링을 위한 옵저버를 시작하고 상태를 확인합니다.</p>	<p>FSFO를 활성화하기 전 또는 활성화한 후에 옵저버를 시작할 수 있습니다. FSFO가 이미 활성화된 경우 옵저버는 즉시 기본 및 대상 대기 데이터베이스에 대한 상태 및 연결 모니터링을 시작합니다. FSFO가 활성화되지 않은 경우 옵저버는 FSFO가 활성화될 때까지 모니터링을 시작하지 않습니다.</p> <p>옵저버를 시작하면 이전 <code>show configuration</code> 명령에서 알 수 있듯이 기본 DB 구성이 오류 메시지 없이 표시됩니다.</p> <p>비 CDB</p> <pre>DGMGRL> start observer; [W000 2022-12-0 1T06:16:51.271+00:00] FSFO target standby is orcl_d Observer 'ip-10-0- 1-89' started [W000 2022-12-0 1T06:16:51.352+00:00] Observer trace level is set to USER DGMGRL> show configura tion Configuration - rds_dg Protection Mode: MaxAvailability Members: orcl_a - Primary database</pre>	<p>DBA</p>

작업	설명	필요한 기술
	<pre> orcl_d - (*) Physical standby database Fast-Start Failover: Enabled in Zero Data Loss Mode Configuration Status: SUCCESS (status updated 56 seconds ago) DGMGRL> DGMGRL> show observer Configuration - rds_dg Primary: orcl_a Active Target: orcl_d Observer "ip-10-0- 1-89" - Master Host Name: ip-10-0-1 -89 Last Ping to Primary: 1 second ago Last Ping to Target: 1 second ago DGMGRL> CDB DGMGRL> start observer; Succeeded in opening the observer file "/home/oracle/fsfo _ip-10-0-1-56.dat". [W000 2023-01-1 8T07:31:32.589+00:00] FSFO target standby is rdscdb_b Observer 'ip-10-0- 1-56' started The observer log file is '/home/oracle/obse rver_ip-10-0-1-56. log'. </pre>	

작업	설명	필요한 기술
	<pre> DGMGRL> show configura tion Configuration - rds_dg Protection Mode: MaxAvailability Members: rdscdb_a - Primary database rdscdb_b - (*) Physical standby database Fast-Start Failover: Enabled in Zero Data Loss Mode Configuration Status: SUCCESS (status updated 12 seconds ago) DGMGRL> DGMGRL> show observer; Configuration - rds_dg Primary: rdscdb_a Active Target: rdscdb_b Observer "ip-10-0- 1-56" - Master Host Name: ip-10-0-1-56 Last Ping to Primary: 1 second ago Last Ping to Target: 2 seconds ago DGMGRL> </pre>	

작업	설명	필요한 기술
장애 조치 확인합니다.	<p>이 시나리오에서는 기본 EC2 인스턴스를 수동으로 중지하여 장애 조치 테스트를 수행할 수 있습니다. EC2 인스턴스를 중지하기 전에 <code>tail</code> 명령을 사용하여 구성을 기반으로 오퍼저버 로그 파일을 모니터링합니다. DGMGRL을 사용하여 사용자 RDS_DATAGUARD 로 대기 데이터베이스 <code>orc1_d</code>에 로그인하고 Oracle Data Guard 상태를 확인합니다. <code>orc1_d</code>은 새로운 기본 데이터베이스라고 표시되어야 합니다.</p> <p>참고: 이 장애 조치 테스트 시나리오에서는 <code>orc1_d</code>가 비-CDB 데이터베이스입니다.</p> <p>장애 조치 전에는 플래시백 데이터베이스가 <code>orc1_a</code>에서 활성화되었습니다. 이전의 기본 데이터베이스가 온라인 상태로 돌아와 MOUNT 상태로 시작되면 오퍼저버는 해당 데이터베이스를 새로운 대기 데이터베이스로 복원합니다. 복원된 데이터베이스는 새로운 기본 데이터베이스의 FSFO 대상 역할을 합니다. 오퍼저버 로그에서 세부 정보를 확인할 수 있습니다.</p> <pre data-bbox="597 1703 1024 1831">DGMGRL> show configuration Configuration - rds_dg</pre>	DBA

작업	설명	필요한 기술
	<pre> Protection Mode: MaxAvailability Members: orcl_d - Primary database Warning: ORA-16824 : multiple warnings, including fast-start failover-related warnings, detected for the database orcl_a - (*) Physical standby database (disabled) ORA-16661: the standby database needs to be reinstated Fast-Start Failover: Enabled in Zero Data Loss Mode Configuration Status: WARNING (status updated 25 seconds ago) DGMGRL> </pre> <p>다음은 observer.log 에서 의 출력 예입니다.</p> <pre> \$ tail -f /tmp/observer.log Unable to connect to database using rds_custom_orcl_a [W000 2023-01-1 8T07:50:32.589+00:00] Primary database cannot be reached. [W000 2023-01-1 8T07:50:32.589+00:00] Fast-Start Failover threshold has expired. </pre>	

작업	설명	필요한 기술
	<pre> [W000 2023-01-1 8T07:50:32.590+00:00] Try to connect to the standby. [W000 2023-01-1 8T07:50:32.590+00: 00] Making a last connection attempt to primary database before proceeding with Fast- Start Failover. [W000 2023-01-1 8T07:50:32.591+00:00] Check if the standby is ready for failover. [S002 2023-01-1 8T07:50:32.591+00:00] Fast-Start Failover started... 2023-01-18T07:50 :32.591+00:00 Initiating Fast-Star t Failover to databse "orcl_d"... [S002 2023-01-1 8T07:50:32.592+00:00] Initiating Fast-start Failover. Performing failover NOW, please wait... Failover succeeded, new primary is "orcl_d" 2023-01-18T07:55:3 2.101+00:00 [S002 2023-01-1 8T07:55:32.591+00:00] Fast-Start Failover finished... [W000 2023-01-1 8T07:55:32.591+00:00] Failover succeeded. Restart pinging. </pre>	

작업	설명	필요한 기술
	<pre> [W000 2023-01-1 8T07:55:32.603+00:00] Primary database has changed to orcl_d. [W000 2023-01-1 8T07:55:33.618+00:00] Try to connect to the primary. [W000 2023-01-1 8T07:55:33.622+00: 00] Try to connect to the primary rds_custo m_orcl_d. [W000 2023-01-1 8T07:55:33.634+00: 00] The standby orcl_a needs to be reinstated [W000 2023-01-1 8T07:55:33.654+00:00] Try to connect to the new standby orcl_a. [W000 2023-01-1 8T07:55:33.654+00: 00] Connection to the primary restored! [W000 2023-01-1 8T07:55:35.654+00: 00] Disconnecting from database rds_custo m_orcl_d. [W000 2023-01-1 8T07:55:57.701+00:00] Try to connect to the new standby orcl_a. ORA-12170: TNS:Connect timeout occurred </pre>	

Oracle Peoplesoft 애플리케이션과 데이터베이스 간의 연결 구성

작업	설명	필요한 기술
<p>기본 데이터베이스에서 서비스를 생성하고 시작합니다.</p>	<p>구성에 기본 및 대기 데이터베이스 엔드포인트가 모두 포함된 TNS 항목을 사용하면 역할 전환 중에 애플리케이션 구성이 변경되지 않도록 할 수 있습니다. 읽기/쓰기 및 읽기 전용 워크로드를 모두 지원하는 두 개의 역할 기반 데이터베이스 서비스를 정의할 수 있습니다. 다음 예에서는 기본 데이터베이스에서 orcl_rw는 기본 데이터베이스에 활성화된 읽기/쓰기 서비스입니다. orcl_ro는 읽기 전용 서비스이며 읽기 전용 모드로 열린 대기 데이터베이스에서 활성화됩니다.</p> <pre data-bbox="597 1167 1026 1850"> SQL> select name,open _mode from v\$database; NAME OPEN_MODE ----- ORCL READ WRITE SQL> exec dbms_service.create_service ('orcl_rw','orcl_rw'); PL/SQL procedure successfully completed . SQL> exec dbms_service.create_service ('orcl_ro','orcl_ro');</pre>	DBA

작업	설명	필요한 기술
	<pre> PL/SQL procedure successfully completed . SQL> exec dbms_service.start_service('orcl_rw'); PL/SQL procedure successfully completed . SQL> </pre>	
<p>대기 데이터베이스에서 서비스를 시작합니다.</p>	<p>읽기 전용 대기 데이터베이스에서 서비스를 시작하려면 다음 코드를 사용합니다.</p> <pre> SQL> select name,open _mode from v\$database; NAME OPEN_MODE ----- ORCL READ ONLY WITH APPLY SQL> exec dbms_service.start_service('orcl_ro'); PL/SQL procedure successfully completed . SQL> </pre>	DBA

작업	설명	필요한 기술
<p>기본 DB 재시작 시 서비스 시작을 자동화합니다.</p>	<p>서비스를 다시 시작할 때 기본 데이터베이스에서 서비스를 자동으로 시작하려면 다음 코드를 사용합니다.</p> <pre data-bbox="597 443 1029 1633"> SQL> CREATE OR REPLACE TRIGGER TrgDgServices after startup on database DECLARE db_role VARCHAR(30); db_open_mode VARCHAR(30); BEGIN SELECT DATABASE_ROLE, OPEN_MODE INTO db_role, db_open_mode FROM V \$DATABASE; IF db_role = 'PRIMARY' THEN DBMS_SERV_2 ICE.START _SERVICE('orcl_rw'); END IF; IF db_role = 'PHYSICAL STANDBY' AND db_open_m ode LIKE 'READ ONLY%' THEN DBMS_SERVICE.START_SER VICE('orcl_ro'); END IF; END; / Trigger created. SQL> </pre>	<p>DBA</p>

작업	설명	필요한 기술
<p>읽기/쓰기 데이터베이스와 읽기 전용 데이터베이스 간의 연결을 구성합니다.</p>	<p>다음 애플리케이션 구성 예제를 읽기/쓰기 및 읽기 전용 연결에 사용할 수 있습니다.</p> <pre data-bbox="597 394 1027 1877"> ORCL_RW = (DESCRIPTION = (CONNECT_TIMEOUT= 120)(RETRY_COUNT=2 0)(RETRY_DELAY=3)(TRANSPORT_CONNECT_ TIMEOUT=3) (ADDRESS_LIST = (ADDRESS = (PROTOCOL = TCP)(HOST=devpsftd b.*****.us-west-2 .rds.amazonaws.com) (PORT=1521)) (ADDRESS = (PROTOCOL = TCP)(HOST=psftread .*****.us-west-2. rds.amazonaws.com) (PORT=1521))) (CONNECT_DATA=(SERVIC E_NAME = orcl_rw))) ORCL_RO = (DESCRIPTION = (CONNECT_TIMEOUT= 120)(RETRY_COUNT=2 0)(RETRY_DELAY=3)(TRANSPORT_CONNECT_ TIMEOUT=3) (ADDRESS_LIST = (ADDRESS = (PROTOCOL = TCP)(HOST=devpsftd b.*****.us-west-2 .rds.amazonaws.com) (PORT=1521)) (ADDRESS = (PROTOCOL = TCP)(HOST=psftread </pre>	<p>DBA</p>

작업	설명	필요한 기술
	<pre> *****.us-west-2. ids.amazonaws.com) (PORT=1521))) (CONNECT_DATA=(SERVIC E_NAME = orcl_ro))) </pre>	

관련 리소스

- [Oracle용 Amazon RDS Custom에서 Data Guard를 사용하여 고가용성 활성화](#)(AWS 기술 가이드)
- [Amazon RDS를 오라클 PeopleSoft 데이터베이스로 구성](#) (AWS 백서)
- [Oracle Data Guard Broker 가이드](#)(Oracle 참조 문서)
- [Oracle Data Guard Concepts 및 Administration](#)(Oracle 참조 문서)
- [Oracle Data Guard Specific FAN 및 FCF Configuration Requirements](#)(Oracle 참조 문서)

워크로드별 데이터베이스 마이그레이션 패턴

주제

- [IBM](#)
- [Microsoft](#)
- [해당 사항 없음](#)
- [오픈 소스](#)
- [Oracle](#)
- [SAP](#)

IBM

- [AWS DMS를 사용하여 Db2 데이터베이스를 Amazon EC2에서 Aurora MySQL과 호환되는 Aurora로 마이그레이션](#)
- [로그 전송을 사용하여 운영 중단 시간을 줄임으로써 Db2 for LUW를 Amazon EC2로 마이그레이션합니다.](#)
- [고가용성 재해 복구 기능을 갖춘 Db2 for LUW를 Amazon EC2로 마이그레이션하세요.](#)
- [AWS DMS 및 AWS SCT를 사용하여 Amazon EC2의 IBM Db2에서 PostgreSQL과 호환되는 Aurora PostgreSQL로 마이그레이션하십시오.](#)
- [IBM WebSphere 애플리케이션 서버에서 Amazon EC2의 아파치 톰캣으로 마이그레이션](#)
- [신뢰할 수 있는 컨텍스트를 사용하여 AWS에서 Db2 페더레이션 데이터베이스의 사용자 액세스 보호 및 간소화](#)

Microsoft

- [Microsoft 워크로드의 검색 및 AWS로의 마이그레이션을 가속화합니다.](#)
- [연결된 서버를 사용하여 Amazon EC2의 Microsoft SQL Server에서 온프레미스 Microsoft SQL Server 테이블에 액세스합니다.](#)
- [SQL Server 데이터베이스를 AWS의 MongoDB Atlas로 마이그레이션하기 위한 쿼리 성능 평가](#)
- [Microsoft SQL Server에서 Amazon Aurora PostgreSQL-Compatible Edition으로 데이터베이스 마이그레이션을 지원하도록 Python 및 Perl 애플리케이션 변경](#)
- [AWS 기반 SQL Server의 Always On 가용성 그룹에서 읽기 전용 라우팅 구성](#)
- [마이크로소프트 엑셀과 Python을 사용하여 AWS DMS 작업을 위한 AWS CloudFormation 템플릿을 생성합니다.](#)
- [AWS DMS를 사용하여 Microsoft SQL Server 데이터베이스를 Amazon S3로 내보내기](#)
- [AWS DMS를 사용하여 Amazon RDS for SQL Server 테이블을 S3 버킷으로 내보내기](#)
- [EC2 Windows 인스턴스를 수집하여 AWS Managed Services 계정으로 마이그레이션합니다.](#)
- [메시지 대기열을 Microsoft Azure 서비스 버스에서 Amazon SQS로 마이그레이션](#)
- [AWS DMS를 사용하여 Microsoft SQL 서버 데이터베이스를 Amazon EC2에서 Amazon DocumentDB로 마이그레이션](#)
- [AWS DMS와 AWS SCT를 사용하여 Microsoft SQL Server 데이터베이스를 Aurora MySQL로 마이그레이션](#)
- [Microsoft Azure 앱 서비스의 .NET 애플리케이션을 AWS Elastic Beanstalk로 마이그레이션합니다.](#)
- [온프레미스 Microsoft SQL Server 데이터베이스를 Amazon EC2로 마이그레이션](#)
- [온프레미스 Microsoft SQL Server 데이터베이스를 Amazon RDS for SQL Server로 마이그레이션](#)
- [연결된 서버를 사용하여 온프레미스 Microsoft SQL Server 데이터베이스를 Amazon RDS for SQL Server로 마이그레이션](#)
- [기본 백업 및 복원 수단을 사용하여 온프레미스 Microsoft SQL Server 데이터베이스를 Amazon RDS for SQL Server로 마이그레이션](#)
- [AWS DMS를 사용하여 온프레미스 Microsoft SQL Server 데이터베이스를 Amazon Redshift로 마이그레이션](#)
- [SCT 데이터 추출 에이전트를 사용하여 온프레미스 Microsoft SQL Server 데이터베이스를 Amazon Redshift로 마이그레이션](#)
- [Linux가 실행되는 Amazon EC2의 Microsoft SQL Server로 온프레미스 Microsoft SQL Server 데이터베이스의 마이그레이션](#)

- [Rclone를 사용하여 Microsoft Azure Blob에서 Amazon S3로 데이터 마이그레이션하기](#)
- [분산된 가용성 그룹을 사용하여 SQL Server를 AWS로 마이그레이션](#)
- [ACM을 사용하여 Windows SSL 인증서를 Application Load Balancer로 마이그레이션](#)
- [온프레미스 SMTP 서버 및 Database Mail을 사용하여 Amazon RDS for SQL Server 데이터베이스 인스턴스에 대한 알림 전송하기](#)
- [Amazon FSx를 사용하여 SQL Server Always On FCI용 다중 AZ 인프라 설정](#)

해당 사항 없음

- [AWS로 리호스팅 마이그레이션하는 동안 방화벽 요청에 대한 승인 프로세스를 생성](#)
- [기존 Amazon RDS for PostgreSQL DB 인스턴스 암호화하기](#)
- [Amazon DynamoDB 테이블의 스토리지 비용 추정](#)
- [AWS DMS와 Amazon Aurora를 사용하여 지역 간 재해 복구 구현](#)

오픈 소스

- [pgAdmin에서 SSH 터널을 사용하여 연결](#)
- [Aurora PostgreSQL 호환에서 애플리케이션 사용자 및 역할을 생성](#)
- [Amazon RDS에서 PostgreSQL DB 인스턴스에 대한 암호화된 연결 활성화하기](#)
- [온프레미스 MariaDB 데이터베이스를 기본 도구를 사용하여 Amazon RDS for MariaDB로 마이그레이션](#)
- [온프레미스 MySQL 데이터베이스를 Amazon EC2로 마이그레이션](#)
- [온프레미스 MySQL 데이터베이스를 Amazon RDS for MySQL로 마이그레이션](#)
- [온프레미스 MySQL 데이터베이스를 Aurora MySQL로 마이그레이션](#)
- [온프레미스 PostgreSQL 데이터베이스를 Aurora PostgreSQL로 마이그레이션하기](#)
- [Auto Scaling을 사용하여 IBM WebSphere 애플리케이션 서버에서 Amazon EC2의 아파치 톰캣으로 마이그레이션하세요](#)
- [AWS DMS를 사용하여 오라클 8i 또는 9i에서 오라클용 아마존 RDS로 마이그레이션할 수 있습니다. SharePlex](#)
- [오라클에서 AWS Elastic GlassFish Beanstalk로 마이그레이션](#)
- [pglogical을 사용하여 Amazon EC2의 PostgreSQL에서 Amazon RDS for PostgreSQL로 마이그레이션합니다.](#)
- [AWS App2Container를 사용하여 온프레미스 Java 애플리케이션을 AWS로 마이그레이션](#)
- [페르코나, Amazon EFS 및 Amazon S3를 사용하여 온프레미스 MySQL 데이터베이스를 Aurora MySQL로 XtraBackup 마이그레이션합니다.](#)
- [Oracle 외부 테이블을 Amazon Aurora PostgreSQL 호환으로 마이그레이션](#)
- [100개 이상의 인수가 있는 Oracle 함수 및 프로시저를 PostgreSQL로 마이그레이션](#)
- [Redis 워크로드를 AWS의 Redis Enterprise Cloud로 마이그레이션](#)
- [암호화를 사용하지 않는 인스턴스가 있는지 Amazon Aurora를 모니터링](#)
- [RHEL 소스 서버를 재부팅한 후 SELinux를 비활성화하지 않고 Replication Agent를 자동으로 다시 시작](#)
- [Lambda와 Secrets Manager를 사용하여 Amazon RDS for PostgreSQL 및 Aurora PostgreSQL 작업 예약하기](#)
- [GTID를 사용하여 Amazon RDS for MySQL와 Amazon EC2의 MySQL 간에 데이터 복제를 설정합니다.](#)
- [pg_transport를 사용하여 두 Amazon RDS DB 인스턴스 간에 PostgreSQL 데이터베이스 전송](#)

Oracle

- [읽기 전용 복제본을 사용하여 Amazon RDS 커스텀 기반 PeopleSoft 오라클에 HA를 추가합니다.](#)
- [Oracle Database와 Aurora PostgreSQL-Compatible 간의 링크 구성](#)
- [JSON Oracle 쿼리를 PostgreSQL 데이터베이스 SQL로 변환](#)
- [Oracle의 VARCHAR2\(1\) 데이터 유형을 Amazon Aurora PostgreSQL의 부울 데이터 유형으로 변환](#)
- [PostgreSQL-compatible Aurora 글로벌 데이터베이스를 사용하여 Oracle DR 에뮬레이션하기](#)
- [Aurora PostgreSQL의 사용자 지정 엔드포인트를 사용하여 Oracle RAC 워크로드 에뮬레이션하기](#)
- [AWR 보고서를 사용하여 오라클 데이터베이스의 Amazon RDS 엔진 크기를 추정합니다.](#)
- [Aurora PostgreSQL의 동적 SQL 명령문에서 익명 블록 처리](#)
- [Aurora PostgreSQL-Compatible에서 오버로드된 Oracle 함수 처리](#)
- [Oracle SQL Developer 및 AWS SCT를 사용하여 Amazon RDS for Oracle에서 Amazon RDS for PostgreSQL로 점진적으로 마이그레이션](#)
- [Aurora PostgreSQL-Compatible에서 파일 인코딩을 사용하여 BLOB 파일을 TEXT에 로드](#)
- [Amazon RDS for Oracle DB 인스턴스를 AMS를 사용하는 다른 계정으로 마이그레이션](#)
- [AWS DMS를 사용하여 SSL 모드에서 Amazon RDS for Oracle를 Amazon RDS for PostgreSQL로 마이그레이션](#)
- [AWS CLI와 AWS를 사용하여 AWS SCT 및 AWS DMS를 사용하여 오라클용 Amazon RDS를 PostgreSQL용 Amazon RDS로 마이그레이션합니다. CloudFormation](#)
- [지속적인 복제를 위해 AWS DMS를 사용하여 Amazon RDS for Oracle 데이터베이스를 다른 AWS 계정 및 AWS 리전으로 마이그레이션](#)
- [Amazon RDS for Oracle DB 인스턴스를 다른 VPC로 마이그레이션](#)
- [Oracle Data Pump를 사용하여 온프레미스 Oracle 데이터베이스를 Amazon EC2 로 마이그레이션](#)
- [Logstash를 사용하여 온프레미스 Oracle 데이터베이스를 아마존 OpenSearch 서비스로 마이그레이션합니다.](#)
- [DMS 및 SCT를 사용하여 온프레미스 Oracle 데이터베이스를 Amazon RDS for MySQL로 마이그레이션](#)
- [온프레미스 Oracle 데이터베이스를 Amazon RDS for Oracle로 마이그레이션](#)
- [데이터베이스 링크를 통한 직접 Oracle 데이터 펌프 가져오기를 사용하여 온프레미스 Oracle 데이터베이스를 Amazon RDS for Oracle로 마이그레이션](#)
- [Oracle Data Pump를 사용하여 온프레미스 Oracle 데이터베이스를 Amazon RDS for Oracle로 마이그레이션](#)

- [Oracle bystander 및 AWS DMS를 사용하여 온프레미스 Oracle 데이터베이스를 Amazon RDS for PostgreSQL로 마이그레이션](#)
- [온프레미스 Oracle 데이터베이스를 Amazon EC2의 Oracle로 마이그레이션](#)
- [AWS DMS 및 AWS SCT를 사용하여 Amazon EC2에서 Amazon RDS for MariaDB로 Oracle 데이터베이스 마이그레이션](#)
- [AWS DMS를 사용하여 Amazon EC2에서 Amazon RDS for Oracle로 Oracle 데이터베이스 마이그레이션](#)
- [AWS DMS를 사용하여 Amazon DynamoDB로 Oracle 데이터베이스 마이그레이션](#)
- [Oracle 플랫폼 파일 어댑터를 사용하여 GoldenGate Oracle 데이터베이스를 Amazon RDS for Oracle로 마이그레이션합니다.](#)
- [AWS DMS 및 AWS SCT를 사용하여 Amazon Redshift로 Oracle 데이터베이스 마이그레이션](#)
- [AWS DMS 및 AWS SCT를 사용하여 Aurora PostgreSQL로 Oracle 데이터베이스를 마이그레이션하기](#)
- [오라클 데이터 펌프와 AWS DMS를 사용하여 오라클 JD 에드워즈 EnterpriseOne 데이터베이스를 AWS로 마이그레이션하십시오.](#)
- [AWS DMS를 사용하여 Oracle 파티션형 테이블을 PostgreSQL로 마이그레이션하기](#)
- [AWS DMS를 사용하여 오라클 PeopleSoft 데이터베이스를 AWS로 마이그레이션합니다.](#)
- [Aurora PostgreSQL로 온프레미스 Oracle 데이터베이스의 데이터를 마이그레이션하기](#)
- [Amazon RDS for Oracle에서 Amazon RDS for MySQL로 마이그레이션](#)
- [구체화된 뷰와 AWS DMS를 사용하여 Oracle 8i 또는 9i에서 Amazon RDS for PostgreSQL로 마이그레이션](#)
- [AWS DMS를 사용하여 Oracle 8i 또는 9i에서 PostgreSQL용 아마존 RDS로 마이그레이션하십시오. SharePlex](#)
- [오라클을 사용하여 Oracle 데이터베이스에서 PostgreSQL용 Amazon RDS로 마이그레이션하십시오. GoldenGate](#)
- [AWS DMS 및 AWS SCT를 사용하여 Amazon EC2의 오라클에서 Amazon RDS for MySQL로 마이그레이션하십시오.](#)
- [AWS DMS를 사용하여 Oracle에서 Amazon DocumentDB로 마이그레이션](#)
- [오라클에서 Amazon WebLogic ECS의 아파치 톰캣 \(TomEE\) 으로 마이그레이션](#)
- [함수 기반 인덱스를 Oracle에서 PostgreSQL로 마이그레이션](#)
- [레거시 애플리케이션을 Oracle Pro*C에서 ECPG로 마이그레이션](#)
- [AWS에서 PostgreSQL의 개별 행으로 Oracle CLOB 값을 마이그레이션](#)

- [Oracle Database 오류 코드를 Amazon Aurora PostgreSQL Compatible 데이터베이스로 마이그레이션](#)
- [Oracle E-Business Suite를 Amazon RDS Custom으로 마이그레이션](#)
- [확장 기능을 사용하여 Oracle 네이티브 함수를 PostgreSQL로 마이그레이션](#)
- [Oracle OUT 바인드 변수를 PostgreSQL 데이터베이스로 마이그레이션](#)
- [PeopleSoft 오라클을 Amazon RDS 커스텀으로 마이그레이션](#)
- [Oracle ROWID 기능을 AWS 기반 PostgreSQL로 마이그레이션](#)
- [Oracle SERIALLY_REUSABLE 프라그마 패키지를 PostgreSQL로 마이그레이션](#)
- [가상으로 생성된 열을 오라클에서 PostgreSQL로 마이그레이션](#)
- [Amazon을 사용하여 오라클 GoldenGate 로그를 모니터링하십시오. CloudWatch](#)
- [Amazon RDS for Oracle에서 Oracle Database Enterprise Edition을 Standard Edition 2로 리플랫폼](#)
- [활성 대기 데이터베이스를 사용하여 Amazon RDS Custom에서 Oracle E-Business Suite를 위한 HA/DR 아키텍처를 설정합니다.](#)
- [Aurora PostgreSQL 호환에서 Oracle UTL_FILE 기능 설정](#)
- [Amazon RDS 기반 오라클 PeopleSoft 애플리케이션의 역할 전환 \(오라클용 사용자 지정\)](#)
- [오라클에서 Amazon Aurora PostgreSQL로 마이그레이션한 다음 데이터베이스 객체를 검증합니다.](#)

SAP

- [Systems Manager를 사용하여 SAP HANA 데이터베이스를 자동으로 백업하고 EventBridge](#)
- [온프레미스 SAP ASE 데이터베이스를 Amazon EC2로 마이그레이션](#)
- [AWS DMS를 사용하여 SAP ASE에서 Amazon RDS for SQL Server로 마이그레이션](#)
- [AWS SCT 및 AWS DMS를 사용하여 SAP ASE에 있는 Amazon EC2를 Amazon Aurora PostgreSQL-Compatible로 마이그레이션하기](#)
- [동일한 호스트 이름을 가진 SAP HSR을 사용하여 SAP HANA를 AWS로 마이그레이션](#)
- [Application Migration Service를 사용하여 동종 SAP 마이그레이션 전환 시간 단축](#)
- [AWS 기반 IBM Db2에서 SAP를 위한 재해 복구 설정](#)

패턴 더 보기

- [Athena를 사용한 Amazon DynamoDB 테이블 액세스, 쿼리 및 조인](#)
- [Athena의 ML 예측을 위한 Amazon DynamoDB의 데이터 집계](#)
- [AMS 계정의 S3 버킷에 대한 EC2 인스턴스 쓰기 액세스 허용](#)
- [Amazon Athena와 Amazon을 사용하여 중첩된 JSON 데이터를 분석하고 시각화합니다. QuickSight](#)
- [AWS Directory Service를 사용하여 Amazon EC2에서 Microsoft SQL Server 인증하기](#)
- [AWS Batch를 사용하여 Amazon RDS for PostgreSQL DB 인스턴스 백업 자동화](#)
- [DynamoDB TTL을 사용하여 Amazon S3에 항목 자동으로 보관](#)
- [Python 애플리케이션을 사용하여 Amazon DynamoDB용 PynamoDB 모델 및 CRUD 함수를 자동으로 생성합니다.](#)
- [암호화되지 않은 Amazon RDS DB 인스턴스 및 클러스터를 자동으로 수정하기](#)
- [AWS Systems Manager Maintenance Windows를 사용하여 Amazon RDS DB 인스턴스를 자동 중지 및 시작](#)
- [DevOps 프랙티스와 AWS Cloud9를 사용하여 마이크로서비스로 느슨하게 연결된 아키텍처를 구축 하십시오.](#)
- [Microsoft SQL Server에서 Amazon Aurora PostgreSQL-Compatible Edition으로 데이터베이스 마이그레이션을 지원하도록 Python 및 Perl 애플리케이션 변경](#)
- [Amazon DynamoDB에 대한 크로스 계정 액세스 구성](#)
- [Oracle Database와 Aurora PostgreSQL-Compatible 간의 링크 구성](#)
- [Python을 사용하여 AWS에서 EBCDIC 데이터를 ASCII로 변환 및 압축 해제](#)
- [Teradata NORMALIZE 임시 기능을 Amazon Redshift SQL로 변환](#)
- [Teradata RESET WHEN 기능을 Amazon Redshift SQL로 변환](#)
- [Oracle의 VARCHAR2\(1\) 데이터 유형을 Amazon Aurora PostgreSQL의 부울 데이터 유형으로 변환](#)
- [Aurora PostgreSQL 호환에서 애플리케이션 사용자 및 역할을 생성](#)
- [마이크로소프트 엑셀과 Python을 사용하여 AWS DMS 작업을 위한 AWS CloudFormation 템플릿을 생성합니다.](#)
- [Kinesis Data Streams를 사용하고 AWS CDK를 사용하는 Amazon Data Firehose를 사용하여 Amazon S3에 DynamoDB 레코드를 제공합니다.](#)
- [프라이빗 고정 IP를 사용하여 Amazon EC2에 Cassandra 클러스터를 배포하여 리밸런싱 방지](#)
- [RAG 및 프롬프트를 사용하여 고급 제너레이티브 AI 채팅 기반 어시스턴트 개발 ReAct](#)
- [PostgreSQL-compatible Aurora 글로벌 데이터베이스를 사용하여 Oracle DR 에뮬레이션하기](#)

- [Amazon RDS for SQL Server에서 투명한 데이터 암호화 활성화하기](#)
- [AWS DMS를 사용하여 Microsoft SQL Server 데이터베이스를 Amazon S3로 내보내기](#)
- [Oracle SQL Developer 및 AWS SCT를 사용하여 Amazon RDS for Oracle에서 Amazon RDS for PostgreSQL로 점진적으로 마이그레이션](#)
- [Aurora PostgreSQL-Compatible에서 파일 인코딩을 사용하여 BLOB 파일을 TEXT에 로드](#)
- [AWS Secrets Manager를 사용한 보안 인증 정보 관리](#)
- [AWS DMS를 사용하여 Db2 데이터베이스를 Amazon EC2에서 Aurora MySQL과 호환되는 Aurora로 마이그레이션](#)
- [AWS DMS를 사용하여 Microsoft SQL 서버 데이터베이스를 Amazon EC2에서 Amazon DocumentDB로 마이그레이션](#)
- [AWS DMS와 AWS SCT를 사용하여 Microsoft SQL Server 데이터베이스를 Aurora MySQL로 마이그레이션](#)
- [자체 호스팅된 MongoDB 환경을 AWS 클라우드의 MongoDB Atlas로 마이그레이션](#)
- [AWS SCT 데이터 추출 에이전트를 사용하여 Teradata 데이터베이스를 Amazon Redshift로 마이그레이션](#)
- [AWS DMS를 사용하여 SSL 모드에서 Amazon RDS for Oracle를 Amazon RDS for PostgreSQL로 마이그레이션](#)
- [AWS CLI와 AWS를 사용하여 AWS SCT 및 AWS DMS를 사용하여 오라클용 Amazon RDS를 PostgreSQL용 Amazon RDS로 마이그레이션합니다. CloudFormation](#)
- [Amazon RDS DB 인스턴스를 다른 VPC 또는 계정으로 마이그레이션](#)
- [지속적인 복제를 위해 AWS DMS를 사용하여 Amazon RDS for Oracle 데이터베이스를 다른 AWS 계정 및 AWS 리전으로 마이그레이션](#)
- [Amazon RDS for Oracle DB 인스턴스를 다른 VPC로 마이그레이션](#)
- [Amazon Redshift 클러스터를 중국의 AWS 리전으로 마이그레이션](#)
- [온프레미스 MariaDB 데이터베이스를 기본 도구를 사용하여 Amazon RDS for MariaDB로 마이그레이션](#)
- [온프레미스 Microsoft SQL Server 데이터베이스를 Amazon EC2로 마이그레이션](#)
- [온프레미스 Microsoft SQL Server 데이터베이스를 Amazon RDS for SQL Server로 마이그레이션](#)
- [연결된 서버를 사용하여 온프레미스 Microsoft SQL Server 데이터베이스를 Amazon RDS for SQL Server로 마이그레이션](#)
- [기본 백업 및 복원 수단을 사용하여 온프레미스 Microsoft SQL Server 데이터베이스를 Amazon RDS for SQL Server로 마이그레이션](#)

- [AWS DMS를 사용하여 온프레미스 Microsoft SQL Server 데이터베이스를 Amazon Redshift로 마이그레이션](#)
- [SCT 데이터 추출 에이전트를 사용하여 온프레미스 Microsoft SQL Server 데이터베이스를 Amazon Redshift로 마이그레이션](#)
- [Linux가 실행되는 Amazon EC2의 Microsoft SQL Server로 온프레미스 Microsoft SQL Server 데이터베이스의 마이그레이션](#)
- [온프레미스 MySQL 데이터베이스를 Amazon EC2로 마이그레이션](#)
- [온프레미스 MySQL 데이터베이스를 Amazon RDS for MySQL로 마이그레이션](#)
- [온프레미스 MySQL 데이터베이스를 Aurora MySQL로 마이그레이션](#)
- [Oracle Data Pump를 사용하여 온프레미스 Oracle 데이터베이스를 Amazon EC2 로 마이그레이션](#)
- [Logstash를 사용하여 온프레미스 Oracle 데이터베이스를 아마존 OpenSearch 서비스로 마이그레이션합니다.](#)
- [DMS 및 SCT를 사용하여 온프레미스 Oracle 데이터베이스를 Amazon RDS for MySQL로 마이그레이션](#)
- [온프레미스 Oracle 데이터베이스를 Amazon RDS for Oracle로 마이그레이션](#)
- [데이터베이스 링크를 통한 직접 Oracle 데이터 펌프 가져오기를 사용하여 온프레미스 Oracle 데이터베이스를 Amazon RDS for Oracle로 마이그레이션](#)
- [Oracle Data Pump를 사용하여 온프레미스 Oracle 데이터베이스를 Amazon RDS for Oracle로 마이그레이션](#)
- [Oracle bystander 및 AWS DMS를 사용하여 온프레미스 Oracle 데이터베이스를 Amazon RDS for PostgreSQL로 마이그레이션](#)
- [온프레미스 Oracle 데이터베이스를 Amazon EC2의 Oracle로 마이그레이션](#)
- [온프레미스 PostgreSQL 데이터베이스를 Aurora PostgreSQL로 마이그레이션하기](#)
- [온프레미스 SAP ASE 데이터베이스를 Amazon EC2로 마이그레이션](#)
- [온프레미스 ThoughtSpot 팔콘 데이터베이스를 Amazon Redshift로 마이그레이션하기](#)
- [AWS SCT 데이터 추출 에이전트를 사용하여 온프레미스 Vertica 데이터베이스를 Amazon Redshift로 마이그레이션하기](#)
- [AWS DMS 및 AWS SCT를 사용하여 Amazon EC2에서 Amazon RDS for MariaDB로 Oracle 데이터베이스 마이그레이션](#)
- [AWS DMS를 사용하여 Amazon EC2에서 Amazon RDS for Oracle로 Oracle 데이터베이스 마이그레이션](#)
- [AWS DMS를 사용하여 Amazon DynamoDB로 Oracle 데이터베이스 마이그레이션](#)

- [Oracle 플랫폼 파일 어댑터를 사용하여 GoldenGate Oracle 데이터베이스를 Amazon RDS for Oracle로 마이그레이션합니다.](#)
- [AWS DMS 및 AWS SCT를 사용하여 Amazon Redshift로 Oracle 데이터베이스 마이그레이션](#)
- [AWS DMS 및 AWS SCT를 사용하여 Aurora PostgreSQL로 Oracle 데이터베이스를 마이그레이션하기](#)
- [오라클 데이터 펌프와 AWS DMS를 사용하여 오라클 JD 에드워즈 EnterpriseOne 데이터베이스를 AWS로 마이그레이션하십시오.](#)
- [AWS DMS를 사용하여 Oracle 파티션형 테이블을 PostgreSQL로 마이그레이션하기](#)
- [AWS DMS를 사용하여 오라클 PeopleSoft 데이터베이스를 AWS로 마이그레이션합니다.](#)
- [Aurora PostgreSQL로 온프레미스 Oracle 데이터베이스의 데이터를 마이그레이션하기](#)
- [Starburst를 사용하여 데이터를 AWS 클라우드로 마이그레이션하십시오.](#)
- [로그 전송을 사용하여 운영 중단 시간을 줄임으로써 Db2 for LUW를 Amazon EC2로 마이그레이션합니다.](#)
- [고가용성 재해 복구 기능을 갖춘 Db2 for LUW를 Amazon EC2로 마이그레이션하세요.](#)
- [Amazon RDS for Oracle에서 Amazon RDS for MySQL로 마이그레이션](#)
- [카우치베이스 서버에서 AWS의 카우치베이스 카펠라로 마이그레이션](#)
- [AWS DMS 및 AWS SCT를 사용하여 Amazon EC2의 IBM Db2에서 PostgreSQL과 호환되는 Aurora PostgreSQL로 마이그레이션하십시오.](#)
- [구체화된 뷰와 AWS DMS를 사용하여 Oracle 8i 또는 9i에서 Amazon RDS for PostgreSQL로 마이그레이션](#)
- [AWS DMS를 사용하여 Oracle 8i 또는 9i에서 PostgreSQL용 아마존 RDS로 마이그레이션하십시오. SharePlex](#)
- [오라클을 사용하여 Oracle 데이터베이스에서 PostgreSQL용 Amazon RDS로 마이그레이션하십시오. GoldenGate](#)
- [AWS DMS 및 AWS SCT를 사용하여 Amazon EC2의 오라클에서 Amazon RDS for MySQL로 마이그레이션하십시오.](#)
- [AWS DMS를 사용하여 Oracle에서 Amazon DocumentDB로 마이그레이션](#)
- [pglogical을 사용하여 Amazon EC2의 PostgreSQL에서 Amazon RDS for PostgreSQL로 마이그레이션합니다.](#)
- [AWS DMS를 사용하여 SAP ASE에서 Amazon RDS for SQL Server로 마이그레이션](#)
- [합수 기반 인덱스를 Oracle에서 PostgreSQL로 마이그레이션](#)
- [레거시 애플리케이션을 Oracle Pro*C에서 ECPG로 마이그레이션](#)

- [온프레미스 Cloudera 워크로드를 AWS의 Cloudera 데이터 플랫폼으로 마이그레이션](#)
- [페르코나, Amazon EFS 및 Amazon S3를 사용하여 온프레미스 MySQL 데이터베이스를 Aurora MySQL로 XtraBackup 마이그레이션합니다.](#)
- [온프레미스 서버에서 Oracle Business Intelligence 12c를 AWS 클라우드로 마이그레이션.](#)
- [AWS에서 PostgreSQL의 개별 행으로 Oracle CLOB 값을 마이그레이션](#)
- [Oracle Database 오류 코드를 Amazon Aurora PostgreSQL Compatible 데이터베이스로 마이그레이션](#)
- [Oracle E-Business Suite를 Amazon RDS Custom으로 마이그레이션](#)
- [Oracle 외부 테이블을 Amazon Aurora PostgreSQL 호환으로 마이그레이션](#)
- [확장 기능을 사용하여 Oracle 네이티브 함수를 PostgreSQL로 마이그레이션](#)
- [PeopleSoft 오라클을 Amazon RDS 커스텀으로 마이그레이션](#)
- [Oracle ROWID 기능을 AWS 기반 PostgreSQL로 마이그레이션](#)
- [Oracle SERIALLY_REUSABLE 프로그래밍 패키지를 PostgreSQL로 마이그레이션](#)
- [Redis 워크로드를 AWS의 Redis Enterprise Cloud로 마이그레이션](#)
- [AWS SCT 및 AWS DMS를 사용하여 SAP ASE에 있는 Amazon EC2를 Amazon Aurora PostgreSQL-Compatible로 마이그레이션하기](#)
- [가상으로 생성된 열을 오라클에서 PostgreSQL로 마이그레이션](#)
- [Amazon ElastiCache 클러스터의 미사용 암호화 모니터링](#)
- [보안 그룹의 ElastiCache 클러스터 모니터링](#)
- [Application Migration Service를 사용하여 동종 SAP 마이그레이션 전환 시간 단축](#)
- [컨테이너를 다시 시작하지 않고 데이터베이스 보안 인증 교체](#)
- [AWS Fargate를 사용하여 메시지 기반 워크로드를 대규모로 실행](#)
- [AWS에서 고가용성 PeopleSoft 아키텍처 설정](#)
- [Aurora PostgreSQL 호환에서 Oracle UTL_FILE 기능 설정](#)
- [대규모 Db2 z/OS 데이터를 CSV 파일로 Amazon S3에 전송](#)
- [pg_transport를 사용하여 두 Amazon RDS DB 인스턴스 간에 PostgreSQL 데이터베이스 전송](#)
- [온프레미스 데이터베이스의 재해 CloudEndure 복구에 사용](#)
- [오라클에서 Amazon Aurora PostgreSQL로 마이그레이션한 다음 데이터베이스 객체를 검증합니다.](#)
- [새 Amazon Redshift 클러스터가 VPC에서 시작되는지 확인](#)

DevOps

주제

- [AWS 리소스 평가 자동화](#)
- [오픈소스 도구를 사용하여 SAP 시스템을 자동으로 설치](#)
- [AWS CDK를 사용하여 AWS Service Catalog 포트폴리오 및 제품 배포 자동화](#)
- [및 이벤트를 사용하여 CodeBuild Amazon CodeCommit S3로 이벤트 기반 백업을 자동화합니다. CloudWatch](#)
- [AWS CodePipeline 및 AWS를 사용하여 스택 세트 배포를 자동화합니다. CodeBuild](#)
- [Cloud Custodian 및 AWS CDK를 사용하여 Systems Manager용 AWS 관리형 정책을 EC2 인스턴스 프로파일에 자동으로 연결](#)
- [AWS CDK를 사용하여 마이크로서비스용 CI/CD 파이프라인 및 Amazon ECS 클러스터 자동으로 구축](#)
- [DevOps 프랙티스와 AWS Cloud9를 사용하여 마이크로서비스로 느슨하게 연결된 아키텍처를 구축 하십시오.](#)
- [GitHub 액션과 테라폼을 사용하여 Docker 이미지를 빌드하고 Amazon ECR로 푸시합니다.](#)
- [AWS CodeCommit, AWS 및 AWS Device Farm을 사용하여 iOS 앱을 구축하고 테스트할 수 있습니다. CodePipeline](#)
- [cdk-nag 규칙 팩을 사용하여 AWS CDK 애플리케이션 또는 CloudFormation 템플릿에서 모범 사례를 확인하십시오.](#)
- [Amazon DynamoDB에 대한 크로스 계정 액세스 구성](#)
- [Amazon EKS에서 실행되는 애플리케이션에 대한 상호 TLS 인증을 구성합니다.](#)
- [Firelens 로그 라우터를 사용하여 Amazon ECS용 사용자 지정 로그 구문 분석기를 생성](#)
- [HashiCorp 팩터를 사용하여 CodePipeline 파이프라인 및 AMI 생성](#)
- [파이프라인을 생성하고 다음을 사용하여 온프레미스 EC2 인스턴스에 아티팩트 업데이트를 배포합니다. CodePipeline](#)
- [Java 및 Python 프로젝트를 위한 동적 CI 파이프라인을 자동으로 생성](#)
- [테라폼을 사용하여 CloudWatch Synthetics 카나리아를 배포하세요](#)
- [Amazon ECS에 Java 마이크로서비스를 위한 CI/CD 파이프라인 배포](#)
- [CodeCommit AWS와 AWS를 사용하여 여러 AWS CodePipeline 계정에 CI/CD 파이프라인을 배포 할 수 있습니다.](#)

- [AWS Network Firewall과 AWS Transit Gateway를 사용하여 방화벽 배포](#)
- [AWS CodePipeline CI/CD 파이프라인을 사용하여 AWS Glue 작업을 배포하십시오.](#)
- [EC2 인스턴스 프로파일을 사용하여 AWS Cloud9에서 Amazon EKS 클러스터의 배포](#)
- [AWS CodePipeline, AWS 및 AWS를 사용하여 여러 AWS 지역에 코드를 배포합니다. CodeCommit CodeBuild](#)
- [AWS Organizations의 조직 전체에서 AWS Backup 보고서를 CSV 파일로 내보내기](#)
- [Amazon EC2 인스턴스 목록의 태그를 CSV 파일로 내보내기](#)
- [트로포스피어를 사용하여 AWS Config 관리형 규칙이 포함된 AWS CloudFormation 템플릿을 생성합니다.](#)
- [SageMaker 노트북 인스턴스에 다른 AWS 계정의 CodeCommit 리포지토리에 대한 임시 액세스 권한 부여](#)
- [다중 DevOps 계정 GitHub 환경을 위한 Flow 분기 전략 구현](#)
- [다중 계정 환경을 위한 Gitflow 브랜칭 전략 구현 DevOps](#)
- [다중 DevOps 계정 환경을 위한 트렁크 브랜칭 전략 구현](#)
- [변경 사항을 자동으로 감지하고 단일 저장소에 대한 다양한 CodePipeline 파이프라인을 시작합니다. CodeCommit](#)
- [AWS를 사용하여 비트버킷 리포지토리를 AWS Amplify와 통합 CloudFormation](#)
- [Step Functions와 Lambda 프록시 함수를 사용하여 여러 AWS 계정에서 CodeBuild 프로젝트를 시작합니다.](#)
- [AWS 코드 서비스 및 AWS KMS 다중 리전 키를 사용하여 여러 계정 및 리전에 대한 마이크로서비스의 블루/그린 배포를 관리](#)
- [AWS 및 AWS Config를 사용하여 Amazon ECR 리포지토리에서 와일드카드 권한을 모니터링합니다. CloudFormation](#)
- [AWS CodeCommit 이벤트에서 사용자 지정 작업 수행](#)
- [Amazon CloudWatch 메트릭을 CSV 파일에 게시](#)
- [pytest 프레임워크를 사용하여 AWS Glue에서 Python ETL 작업에 대한 유닛 테스트 실행](#)
- [Amazon S3에서 Helm v3 차트 리포지토리 설정](#)
- [AWS 및 CodePipeline AWS CDK를 사용하여 CI/CD 파이프라인을 설정합니다.](#)
- [인증서 관리자 및 Let's Encrypt를 사용하여 Amazon EKS의 애플리케이션에 대한 end-to-end 암호화를 설정합니다.](#)
- [Flux를 사용하여 Amazon EKS 멀티테넌트 애플리케이션 배포를 간소화합니다.](#)
- [사용자 지정 리소스를 사용하여 여러 이메일 엔드포인트에서 SNS 주제를 구독하십시오.](#)

- [인프라 코드의 테스트 기반 개발을 위해 Serverspec을 사용합니다](#)
- [AWS의 타사 Git 소스 리포지토리 사용 CodePipeline](#)
- [AWS를 사용하여 테라폼 구성을 검증하기 위한 CI/CD 파이프라인을 생성합니다. CodePipeline](#)
- [패턴 더 보기](#)

AWS 리소스 평가 자동화

작성자: Naveen Suthar(AWS), Arun Bagal(AWS), Manish Garg(AWS), Sandeep Gawande(AWS)

코드 리포지토리: infrastructure-assessment-iac-automation	환경: PoC 또는 파일럿	기술: DevOps; 인프라; 관리 및 거버넌스, 운영, 서버리스
AWS 서비스: 아마존 아테나, AWS, AWS Lambda CloudTrail, 아마존 S3, 아마존 QuickSight		

요약

이 패턴은 [AWS Cloud Development Kit\(AWS CDK\)](#)를 사용하여 리소스 평가 기능을 설정하는 자동화된 접근 방식을 설명합니다. 운영팀은 이 패턴을 사용하여 자동화된 방식으로 리소스 감사 세부 정보를 수집하고 단일 대시보드에서 AWS 계정에 배포된 모든 리소스의 세부 정보를 볼 수 있습니다. 이는 다음과 같은 경우에 유용합니다.

- 코드형 인프라 (IaC) 도구를 식별하고 [HashiCorp Terraform](#), AWS CloudFormation, AWS CDK 및 AWS [명령줄](#) 인터페이스 (AWS CLI) 와 같은 다양한 IaC 솔루션에서 생성된 리소스를 격리합니다.
- 리소스 감사 정보 가져오기

또한 이 솔루션은 경영진이 단일 대시보드에서 AWS 계정의 리소스 및 활동에 대한 통찰력을 얻는 데도 도움이 됩니다.

참고: QuickSight [Amazon](#)은 유료 서비스입니다. 이를 실행하여 데이터를 분석하고 대시보드를 생성하기 전에 [Amazon QuickSight 요금](#)을 검토하십시오.

사전 조건 및 제한 사항

사전 조건

- 활성 상태의 AWS 계정.

- AWS Identity and Access Management(IAM) 역할과 리소스 프로비저닝에 대한 액세스 권한
- [아마존 심플 스토리지 서비스 \(Amazon S3\) 와 아마존 아테나에 액세스할 수 있도록 생성된 아마존 QuickSight 계정](#)
- AWS CDK 버전 2.55.1 이상 설치
- [Python](#) 버전 3.9 이상 설치

제한 사항

- 이 솔루션은 단일 AWS 계정에 배포됩니다.
- CloudTrail AWS가 이미 설정되어 있고 S3 버킷에 데이터를 저장하지 않는 한 솔루션은 배포 전에 발생한 이벤트를 추적하지 않습니다.

제품 버전

- AWS CDK 버전 2.55.1 이상
- Python 버전 3.9 이상

아키텍처

대상 기술 스택

- Amazon Athena
- AWS CloudTrail
- Glue
- Lambda
- 아마존 QuickSight
- Amazon S3

대상 아키텍처

AWS CDK 코드는 AWS 계정에서 리소스 평가 기능을 설정하는 데 필요한 모든 리소스를 배포합니다. 다음 다이어그램은 AWS Glue, Amazon Athena 및 로 CloudTrail 로그를 보내는 프로세스를 보여줍니다. QuickSight

1. CloudTrail 저장을 위해 S3 버킷으로 로그를 전송합니다.
2. 이벤트 알림은 로그를 처리하고 필터링된 데이터를 생성하는 Lambda 함수를 간접 호출합니다.
3. 필터링된 데이터는 다른 S3 버킷에 저장됩니다.
4. S3 버킷의 필터링된 데이터에 AWS Glue 크롤러가 설정되어 AWS Glue 데이터 카탈로그 테이블에 스키마를 생성합니다.
5. Amazon Athena는 필터링된 데이터를 쿼리할 준비가 되었습니다.
6. 는 시각화를 QuickSight 위해 쿼리된 데이터에 액세스합니다.

자동화 및 규모 조정

- AWS Organizations에 조직 전반의 추적이 있는 경우 이 솔루션을 하나의 AWS 계정에서 여러 CloudTrail AWS 계정으로 확장할 수 있습니다. 조직 CloudTrail 수준에서 배포하면 이 솔루션을 사용하여 필요한 모든 리소스에 대한 리소스 감사 세부 정보를 가져올 수도 있습니다.
- 이 패턴은 AWS 서버리스 리소스를 사용하여 솔루션을 배포합니다.

도구

서비스

- [Amazon Athena](#)는 표준 SQL을 사용하여 Amazon S3에 있는 데이터를 직접 분석할 수 있는 대화형 쿼리 서비스입니다.
- [AWS Cloud Development Kit\(AWS CDK\)](#)는 AWS 클라우드 인프라를 코드로 정의하고 프로비저닝하는 데 도움이 되는 소프트웨어 개발 프레임워크입니다.
- [AWS는 AWS CloudFormation](#) 리소스를 설정하고, 빠르고 일관되게 프로비저닝하고, AWS 계정 및 AWS 지역의 수명 주기 전반에 걸쳐 리소스를 관리할 수 있도록 지원합니다.
- [AWS는 AWS](#) 계정의 거버넌스, 규정 준수 및 운영 위험을 감사할 수 있도록 CloudTrail 지원합니다.
- [AWS Glue](#)는 완전 관리형 추출, 전환, 적재(ETL) 서비스입니다. 이를 통해 데이터 스토어와 데이터 스트림 간에 데이터를 안정적으로 분류, 정리, 보강하고 이동할 수 있습니다. 이 패턴은 AWS Glue 크롤러와 AWS Glue 데이터 카탈로그 테이블을 사용합니다.
- [AWS Lambda](#)는 서버를 프로비저닝하거나 관리할 필요 없이 코드를 실행할 수 있도록 도와주는 컴퓨팅 서비스입니다. 필요할 때만 코드를 실행하며 자동으로 확장이 가능하므로 사용한 컴퓨팅 시간 만큼만 비용을 지불합니다.
- [QuickSightAmazon](#)은 단일 대시보드에서 데이터를 시각화, 분석 및 보고하는 데 도움이 되는 클라우드 규모의 비즈니스 인텔리전스 (BI) 서비스입니다.

- [Amazon Simple Storage Service\(S3\)](#)는 원하는 양의 데이터를 저장, 보호 및 검색하는 데 도움이 되는 클라우드 기반 객체 스토리지 서비스입니다.

코드 리포지토리

이 패턴의 코드는 리포지토리에서 사용할 수 있습니다. GitHub [infrastructure-assessment-iac-automation](#)

코드 리포지토리에는 다음 파일과 폴더가 포함되어 있습니다.

- lib 폴더 - AWS CDK는 AWS 리소스를 생성하는 데 사용되는 Python 파일을 구성합니다.
- src/lambda_code - Lambda 함수에서 실행되는 Python 코드
- requirements.txt - 설치해야 하는 모든 Python 종속성 목록
- cdk.json - 리소스를 스핀업하는 데 필요한 값을 제공하는 입력 파일

모범 사례

Lambda 함수에 대한 모니터링 및 알림을 설정합니다. 자세한 내용은 [Lambda 함수 모니터링 및 문제 해결](#)을 참조하십시오. Lambda 함수를 사용할 때의 일반적인 모범 사례는 [AWS 설명서](#)를 참조하십시오.

에픽

환경을 설정합니다.

작업	설명	필요한 기술
로컬 머신에서 저장소를 복제합니다.	리포지토리를 복제하려면 <code>git clone https://github.com/aws-samples/infrastructure-assessment-iac-automation.git</code> 명령을 실행합니다.	AWS DevOps, DevOps 엔지니어
Python 가상 환경을 설정하고 필요한 종속성을 설치합니다.	다음 명령을 실행하여 Python 가상 환경을 설정합니다.	AWS DevOps, DevOps 엔지니어

작업	설명	필요한 기술
	<pre>cd infrastructure-assessment-iac-automation python3 -m venv .venv source .venv/bin/activate</pre> <p>필요한 종속성을 설정하려면 <code>pip install -r requirements.txt</code> 명령을 실행합니다.</p>	
AWS CDK 환경을 설정하고 AWS CDK 코드를 합성합니다.	<ol style="list-style-type: none"> 1. AWS 계정에서 AWS CDK 환경을 설정하려면 <code>cdk bootstrap aws://ACCOUNT-NUMBER/REGION</code> 명령을 실행합니다. 2. 코드를 AWS CloudFormation stack 구성으로 변환하려면 명령을 실행합니다 <code>cdk synth</code>. 	AWS DevOps, DevOps 엔지니어

로컬 머신에서 AWS 보안 인증 설정

작업	설명	필요한 기술
스택이 배포될 계정 및 리전의 내보내기 변수	<p>환경 변수를 사용하여 AWS CDK용 AWS 보안 인증을 제공하려면 다음 명령을 실행합니다.</p> <pre>export CDK_DEFAULT_ACCOUNT=<12 Digit AWS Account Number></pre>	AWS DevOps, DevOps 엔지니어

작업	설명	필요한 기술
	<pre>export CDK_DEFAU LT_REGION=<region></pre>	
<p>AWS CLI 프로필을 설치합니다.</p>	<p>계정에 대한 AWS CLI 프로필을 설정하려면 AWS 설명서의 지침을 따르십시오.</p>	<p>AWS DevOps, DevOps 엔지니어</p>

리소스 평가 도구 구성 및 배포

작업	설명	필요한 기술
<p>계정에 리소스를 배포하십시오.</p>	<p>AWS CDK를 사용하여 AWS 계정에 리소스를 배포하려면 다음을 수행하십시오.</p> <ol style="list-style-type: none"> 복제된 리포지토리의 루트에 있는 <code>cdk.json</code> 파일에서 다음 파라미터에 대한 입력을 제공합니다. <ul style="list-style-type: none"> <code>s3_context</code> <code>ct_context</code> <code>kms_context</code> <code>lambda_context</code> <code>glue_context</code> <code>qs_context</code> <p>이러한 값은 리소스 구성 및 명명법을 정의합니다. 기본 값이 설정되며 필요한 경우 변경할 수 있습니다.</p> <p>참고: S3 버킷이 이미 존재한다는 오류가 발생하지 않도록 하려면 <code>ct</code> 및 <code>output</code></p>	<p>AWS DevOps</p>

작업	설명	필요한 기술
	<p>섹션의 <code>s3_context</code> 에 고유한 이름을 입력해야 합니다.</p> <p>2. 리소스를 배포하려면 <code>cdk deploy</code> 명령을 실행합니다.</p> <p>이 <code>cdk deploy</code> 명령은 이벤트를 기록하고 입력 S3 버킷에 로그 파일을 저장하는 CloudTrail 리소스를 생성합니다. Lambda 함수에서 트레일의 로그 파일을 처리합니다. 필터링된 결과는 출력 S3 버킷에 저장되며 Amazon Athena와 Amazon에서 바로 사용할 수 있습니다. QuickSight</p>	

작업	설명	필요한 기술
<p>AWS Glue 크롤러를 실행하며 데이터 카탈로그 테이블을 생성합니다.</p>	<p>AWS Glue 크롤러는 데이터 스키마를 동적으로 유지하는 데 사용됩니다. 이 솔루션은 AWS Glue 크롤러 스케줄러에서 정의한 대로 정기적으로 크롤러를 실행하여 AWS Glue 데이터 카탈로그 테이블에 파티션을 생성하고 업데이트합니다. 출력 S3 버킷에서 데이터를 사용할 수 있게 되면 다음 단계에 따라 AWS Glue 크롤러를 실행하고 테스트용 데이터 카탈로그 테이블 스키마를 생성합니다.</p> <ol style="list-style-type: none"> 1. AWS Management Console에 로그인하고 에서 AWS Glue 콘솔을 엽니다. 2. 탐색 창의 데이터 카탈로그에서 크롤러를 선택합니다. 3. <code>iac-tool-qa-resource-iac-json-crawler</code> 크롤러를 선택합니다. 4. 크롤러를 실행합니다. 5. 크롤러가 성공적으로 실행되면 AWS Glue 데이터 카탈로그 테이블을 생성합니다. QuickSight AWS는 테이블을 사용하여 데이터를 시각화합니다. <p>참고: AWS CDK 코드는 AWS Glue 크롤러가 특정 시간에 실행</p>	<p>AWS DevOps, DevOps 엔지니어</p>

작업	설명	필요한 기술
	행되도록 구성하지만 온디맨드로 실행할 수도 있습니다.	
QuickSight 구조물을 배포하십시오.	<ol style="list-style-type: none"> 구문을 배포하려면 QuickSight #QuickSight setup - ends 인터-인에서 resource_iac_tool_stack.py 코드의 주석 처리를 제거해야 #QuickSight setup - start 합니다. 주석 처리를 제거한 후 cdk deploy 명령어를 실행하여 QuickSight DataSet 계정에서 and를 생성합니다 QuickSight DataSource . QuickSight 	AWS DevOps, DevOps 엔지니어

작업	설명	필요한 기술
QuickSight 대시보드 생성.	<p>예제 QuickSight 대시보드와 분석을 만들려면 다음과 같이 하십시오.</p> <ol style="list-style-type: none"> 1. QuickSight 콘솔로 이동하여 리소스가 배포되는 AWS 지역을 선택합니다. 2. 탐색 창에서 Datasets를 선택하고 Amazon QuickSight 데이터세트에 이름이 지정된 데이터세트가 ct-operations-iac-ds 생성되었는지 확인합니다. <p>데이터세트가 보이지 않는 경우 구문을 재배포하세요.</p> <p>QuickSight</p> <ol style="list-style-type: none"> 3. ct-operations-iac-ds 데이터 세트를 선택하고 분석에 사용을 선택합니다. 4. 기본 시트를 선택합니다. 5. 왼쪽의 필드 목록에서 각 열을 선택합니다. 6. 필요한 열을 선택한 후 적절한 시각적 유형을 선택하여 데이터를 확인합니다. <p>자세한 내용은 Amazon에서 분석 시작 QuickSight 및 Amazon의 시각적 유형을 참조하십시오 QuickSight.</p>	AWS DevOps, DevOps 엔지니어

솔루션의 모든 AWS 리소스를 정리하십시오.

작업	설명	필요한 기술
AWS 리소스를 제거합니다.	<ol style="list-style-type: none"> 솔루션에서 배포한 AWS 리소스를 제거하려면 <code>cdk destroy</code> 명령을 실행합니다. 두 S3 버킷에서 모든 객체를 삭제한 다음 버킷을 제거합니다. <p>자세한 내용은 버킷 삭제를 참조하십시오.</p>	AWS DevOps, DevOps 엔지니어

AWS 리소스 평가 도구 자동화에 추가 기능 설정

작업	설명	필요한 기술
수동으로 생성한 리소스를 모니터링하고 정리합니다.	<p>(선택 사항) 조직에 IaC 도구를 사용하여 리소스를 생성해야 하는 규정 준수 요구 사항이 있는 경우, AWS 리소스 평가 도구 자동화를 사용하여 수동으로 프로비저닝된 리소스를 가져옴으로써 규정 준수를 달성할 수 있습니다. 도구를 사용하여 리소스를 IaC 도구로 가져오거나 다시 생성할 수도 있습니다. 수동으로 프로비저닝된 리소스를 모니터링하려면 다음과 같은 높은 수준의 작업을 수행합니다.</p> <ol style="list-style-type: none"> AWS 리소스 평가 도구 자동화를 배포하십시오. 	AWS DevOps, DevOps 엔지니어

작업	설명	필요한 기술
	<p>2. Lambda 함수를 설정하여 매일 Athena 테이블을 쿼리하고, 수동으로 프로비저닝된 리소스에 대한 관련 데이터를 찾고, 이를 쉼표로 구분된 값(CSV) 파일로 내보냅니다.</p> <p>3. Lambda 함수가 실행되면 필요한 데이터가 포함된 알림을 각 이해 관계자에게 보낼 수 있습니다.</p> <p>4. 보존 기간을 연장하려면.csv 파일을 S3 버킷에 저장할 수 있습니다.</p> <p>5. .csv 파일의 정보를 기반으로 수동으로 생성한 리소스를 삭제하거나 기존 IaC 솔루션으로 가져옵니다.</p>	

문제 해결

문제	Solution
AWS CDK가 오류를 반환합니다.	AWS CDK 문제에 대한 도움이 필요하다면 일반적인 AWS CDK 문제 해결 을 참조하십시오.

관련 리소스

- [Python을 사용하여 Lambda 함수 빌드](#)
- [AWS CDK로 시작하기](#)
- [Python에서 AWS CDK로 작업하기](#)
- [CloudTrail 로그 트레일 생성](#)

- [아마존에서 시작하세요 QuickSight](#)

추가 정보

여러 계정

여러 계정에 대해 AWS CLI 보안 인증을 설정하려면 AWS 프로필을 사용하십시오. 자세한 내용은 [AWS CLI 설정](#)의 다중 프로필 구성 섹션을 참조하십시오.

AWS CDK 명령

AWS CDK로 작업할 때는 다음과 같은 유용한 명령을 유념하십시오.

- 앱의 모든 스택 나열하기

```
cdk ls
```

- 합성된 AWS 템플릿을 내보냅니다. CloudFormation

```
cdk synth
```

- 스택을 기본 AWS 계정 및 리전에 배포하기

```
cdk deploy
```

- 배포된 스택을 현재 상태와 비교하기

```
cdk diff
```

- AWS CDK 설명서 열기

```
cdk docs
```

오픈소스 도구를 사용하여 SAP 시스템을 자동으로 설치

작성자: Guilherme Sesterheim(AWS)

코드 리포지토리: 메인 리포지토리	환경: 프로덕션	기술: DevOps
워크로드: SAP	AWS 서비스: Amazon EC2, Amazon S3	

요약

이 패턴은 오픈소스 도구를 사용하여 다음 리소스를 생성함으로써 SAP 시스템 설치를 자동화하는 방법을 보여줍니다.

- SAP S/4HANA 1909 데이터베이스
- SAP ABAP Central Services(ASCS) 인스턴스
- SAP Primary Application Server(PAS) 인스턴스

HashiCorp Terraform은 SAP 시스템의 인프라를 생성하고 Ansible은 운영 체제 (OS) 를 구성하고 SAP 애플리케이션을 설치합니다. Jenkins는 설치를 실행합니다.

이 설정은 SAP 시스템 설치를 반복 가능한 프로세스로 바꾸어 배포 효율성과 품질을 높이는 데 도움이 될 수 있습니다.

참고: 이 패턴에 제공된 예제 코드는고가용성(HA) 시스템과 비 HA 시스템 모두에서 작동합니다.

사전 조건 및 제한 사항

사전 조건

- 활성 상태의 AWS 계정.
- 모든 SAP 미디어 파일이 포함된 Amazon Simple Storage Service(S3) 버킷
- [액세스 키와 비밀 키](#)가 있는 Identity and Access Management(IAM) 보안 주체는 다음과 같은 권한을 가집니다.

- 읽기 전용 권한: Amazon Route 53, Key Management Service(KMS)
- 읽기 및 쓰기 권한: 아마존 S3, 아마존 엘라스틱 컴퓨트 클라우드 (아마존 EC2), 아마존 엘라스틱 파일 시스템 (아마존 EFS), IAM, 아마존, 아마존 DynamoDB CloudWatch
- Route 53 [프라이빗 호스팅 영역](#)
- Amazon Marketplace에서 [HA 및 업데이트 서비스 8.2](#) Amazon Machine Image(AMI)가 있는 SAP용 Red Hat Enterprise Linux에 대한 구독
- [KMS 고객 관리형 키](#)
- [Secure Shell\(SSH\) 키 페어](#)
- Jenkins를 설치하는 곳의 호스트 이름을 통해 포트 22의 SSH 연결을 허용하는 [Amazon EC2 보안 그룹](#)(호스트 이름은 localhost일 가능성이 높음)
- [Vagrant](#) (설치 및 구성 기준) HashiCorp
- [VirtualBox](#)오라클별 설치 및 구성
- Git, Terraform, Ansible, Jenkins에 대한 익숙함

제한 사항

- SAP S/4HANA 1909가 이 특정 시나리오에 대한 완전한 테스트를 거쳤습니다. 다른 버전의 SAP HANA를 사용하는 경우 이 패턴의 예제 Ansible 코드를 수정해야 합니다.
- 이 패턴의 예제 절차는 Mac OS 및 Linux 운영 체제에서 작동합니다. 일부 명령은 Unix 기반 터미널에서만 실행할 수 있습니다. 하지만 각기 다른 명령과 Windows OS를 사용하면 비슷한 결과를 얻을 수 있습니다.

제품 버전

- SAP S/4HANA 1909
- Red Hat Enterprise Linux (RHEL) 8.2 이상 버전

아키텍처

다음 다이어그램은 오픈소스 도구를 사용하여 계정에서 SAP 시스템 설치를 자동화하는 예제 워크플로를 보여 줍니다.

이 다이어그램은 다음 워크플로를 보여줍니다.

1. Jenkins는 Terraform 및 Ansible 코드를 실행하여 SAP 시스템 설치 실행을 오케스트레이션합니다.
2. Terraform 코드는 SAP 시스템의 인프라를 빌드합니다.
3. Ansible 코드는 OS를 구성하고 SAP 애플리케이션을 설치합니다.
4. 정의된 사전 조건을 모두 포함하는 SAP S/4HANA 1909 데이터베이스, ASCS 인스턴스 및 PAS 인스턴스가 Amazon EC2 인스턴스에 설치됩니다.

참고: 이 패턴의 예제 설정은 Terraform 상태 파일을 저장하기 위해 계정에 Amazon S3 버킷을 자동으로 생성합니다.

기술 스택

- Terraform
- Ansible
- Jenkins
- SAP S/4HANA 1909 데이터베이스
- SAP ASCS 인스턴스
- SAP PAS 인스턴스
- Amazon EC2

도구

서비스

- [Amazon Elastic Compute Cloud\(Amazon EC2\)](#)는 클라우드에서 확장 가능한 컴퓨팅 용량을 제공합니다. 필요한 만큼 많은 가상 서버를 시작하고 빠르게 규모를 확장하거나 축소할 수 있습니다.
- [Identity and Access Management\(IAM\)](#)를 사용하여 리소스에 대한 액세스를 안전하게 제어할 수 있습니다.
- [Key Management Service\(AWS KMS\)](#)를 사용하면 암호화 키를 생성하고 제어하여 데이터를 보호할 수 있습니다.
- [Amazon Virtual Private Cloud\(VPC\)](#)를 이용하면 사용자가 정의한 가상 네트워크로 AWS 리소스를 시작할 수 있습니다. 이 가상 네트워크는 사용자의 자체 데이터 센터에서 운영하는 기존 네트워크와 유사하며 AWS의 확장 가능한 인프라를 사용한다는 이점이 있습니다.

기타 도구

- [HashiCorp Terraform](#)은 코드를 사용하여 클라우드 인프라 및 리소스를 프로비저닝하고 관리하는 데 도움이 되는 명령줄 인터페이스 애플리케이션입니다.
- [Ansible](#)은 애플리케이션, 구성 및 IT 인프라를 자동화하는 데 도움이 되는 코드형 오픈소스 구성 (CaC) 도구입니다.
- [Jenkins](#)는 개발자가 그의 소프트웨어를 빌드, 테스트 및 배포할 수 있는 오픈 소스 자동화 서버입니다.

코드

[이 패턴의 코드는 -jenkins-ansible 저장소에서 사용할 수 있습니다. GitHub aws-install-sap-with](#)

에픽

필수 구성 요소 구성

작업	설명	필요한 기술
SAP 미디어 파일을 Amazon S3 버킷에 추가합니다.	모든 SAP 미디어 파일이 포함된 Amazon S3 버킷을 생성 합니다. 중요: Launch Wizard 설명서 에 있는 S/4HANA에 대한 Launch Wizard의 폴더 계층 구조를 따라야 합니다.	클라우드 관리자
설치. VirtualBox	VirtualBox 오라클에서 설치 및 구성합니다.	DevOps 엔지니어
Vagrant를 설치합니다.	를 사용하여 Vagrant 를 HashiCorp 설치 및 구성합니다.	DevOps 엔지니어
계정을 구성합니다.	1. 액세스 키와 비밀 키 를 가진 IAM 보안 주체를 가지고 있고 다음 권한이 있는지 확인합니다.	일반 AWS

작업	설명	필요한 기술
	<ul style="list-style-type: none"> • 읽기 전용 권한: Amazon Route 53, Key Management Service(KMS) • 읽기 및 쓰기 권한: 아 마존 S3, 아마존 엘라스 틱 컴퓨터 클라우드 (아 마존 EC2), 아마존 엘 라스틱 파일 시스템 (아 마존 EFS), IAM, 아 존, 아마존 DynamoDB CloudWatch <ol style="list-style-type: none"> 2. 나중에 참조할 수 있도록 IAM 보안 주체의 액세스 키와 비밀 키를 저장합니다. 3. Route 53 프라이빗 호스팅 영역이 아직 없는 경우 이를 생성합니다. 나중에 참조할 수 있도록 영역 이름(예: sapteam.net)을 저장합니다. 4. Amazon Marketplace에서 HA 및 업데이트 서비스 8.2 Amazon Machine Image(AMI)가 있는 SAP용 Red Hat Enterprise Linux를 구독합니다. 나중에 참조할 수 있도록 AMI ID(예: ami-0000000)를 저장합니다. 5. KMS 고객 관리형 키를 생성합니다. 나중에 참조할 수 있도록 KMS 키의 Amazon 리소스 이름(ARN)을 저장합니다. 	

작업	설명	필요한 기술
	<p>참고: 다음은 KMS 고객 관리형 키 ARN의 예시입니다(arn:aws:kms:us-east-1:123412341234:key/uuid).</p> <p>6. SSH 키 페어를 생성합니다. 나중에 참조할 수 있도록 키 페어의 이름과 .pem 파일을 저장합니다.</p> <p>7. Jenkins를 설치하는 호스트 이름을 통해 포트 22의 SSH 연결을 허용하는 Amazon EC2 보안 그룹을 생성합니다. 나중에 참조할 수 있도록 보안 그룹 ID를 저장합니다.</p> <p>참고: 호스트 이름은 대부분 localhost입니다.</p>	

SAP 설치의 빌드 및 실행

작업	설명	필요한 기술
에서 코드 리포지토리를 복제하십시오. GitHub	aws-install-sap-with-jenkins-ansible 리포지토리를 복제하십시오. GitHub	DevOps 엔지니어
Jenkins 서비스를 시작합니다.	<p>Linux 터미널을 엽니다. 그런 다음 복제된 코드 리포지토리 폴더가 들어 있는 로컬 폴더로 이동하여 다음 명령을 실행합니다.</p> <pre>sudo vagrant up</pre>	DevOps 엔지니어

작업	설명	필요한 기술
	참고: Jenkins를 시작하는 데 약 20분이 소요됩니다. 이 명령은 성공 시 서비스가 설정되었으며 실행 중이라는 메시지를 반환합니다.	
웹 브라우저에서 Jenkins를 열고 로그인합니다.	<ol style="list-style-type: none"> 1. 웹 브라우저에 <code>http://localhost:5555</code>를 입력합니다. Jenkins가 열립니다. 2. 사용자 이름으로 <code>admin</code>을 사용하고 암호로 <code>my_secret_pass_from_vault</code>를 사용하여 Jenkins에 로그인합니다. 	DevOps 엔지니어

작업	설명	필요한 기술
SAP 시스템 설치 파라미터를 구성합니다.	<ol style="list-style-type: none"> Jenkins에서 Jenkins 관리를 선택합니다. 그런 다음 보안 인증 정보 관리를 선택합니다. 구성할 수 있는 보안 인증 정보 변수 목록이 나타납니다. 다음 자격 증명 변수를 모두 구성합니다. <ul style="list-style-type: none"> AWS_ACCOUNT_CREDENTIALS의 경우 IAM 보안 주체의 액세스 키 ID와 비밀 액세스 키 ID를 입력합니다. AMI_ID의 경우 HA 및 업데이트 서비스 8.2 AMI의 AMI ID로 SAP용 Red Hat Enterprise Linux를 입력합니다. KMS_KEY_ARN의 경우 KMS 고객 관리형 키의 ARN을 입력합니다. SSH_KEYPAIR_NAME의 경우 .pem 파일 유형을 입력하지 않고 SSH 키 페어의 이름을 입력합니다. SSH_KEYPAIR_FILE의 경우 키 페어의 .pem 파일의 전체 이름을 입력합니다(예: mykeypair.pem). 키 페어의 .pem 파일도 Jenkins에 업로드해야 합니다. S3_ROOT_FOLDER_INSTALL_FILES의 경우, 	AWS 시스템 관리자, DevOps 엔지니어

작업	설명	필요한 기술
	<p>Amazon S3 버킷의 이름 및 해당하는 경우 SAP 미디어 파일이 포함된 폴더 (예: s3:///S4H1909) 의 이름을 입력합니다. my-media-bucket</p> <ul style="list-style-type: none"> PRIVATE_DNS_ZONE_NAME의 경우 Route 53 프라이빗 호스팅 영역의 이름을 입력합니다(예: myprivatecompanyurl.net). VPC_ID의 경우 SAP 리소스를 생성하고 있는 Amazon VPC의 VPC ID를 입력합니다(예: vpc-12345). SUBNET_IDS의 경우 테스트 환경에서 작업하고 있다면(미래 HA 기능을 위해) 2개의 퍼블릭 서브넷 ID를 입력합니다. 프로덕션 환경에서 작업하는 경우 Bastion Host와 함께 두 개의 프라이빗 서브넷을 사용하는 것이 가장 좋습니다. SECURITY_GROUP_ID의 경우 Jenkins를 설치한 호스트 이름을 통해 포트 22의 SSH 연결을 허용하는 Amazon EC2 보안 그룹의 ID를 입력합니다. <p>참고: 사용 사례를 기준으로 필요에 따라 기타 비필수 파라미터를 구성할 수 있습니다. 예를 들어 SAP 시스템에 관하여 인</p>	

작업	설명	필요한 기술
<p>SAP 시스템 설치를 실행합니다.</p>	<p>스턴스의 SAP 시스템 ID(SID), 기본 암호, 이름, 태그를 변경할 수 있습니다. 모든 필수 변수는 이름 앞에 (필수)라고 표시됩니다.</p> <ol style="list-style-type: none"> 1. Jenkins에서 Jenkins Home을 선택합니다. 그런 다음 SAP Hana+ASCS +PAS 3 Instances를 선택합니다. 2. 스펀업 및 설치를 선택합니다. 그런 다음, Main을 선택합니다. 3. 지금 빌드를 선택합니다. <p>파이프라인 절차에 대한 정보는 AWS 블로그에서 오픈소스 도구를 활용한 SAP 설치 자동화의 파이프라인 절차에 대한 이해 섹션을 참조하십시오.</p> <p>참고: 오류가 발생하는 경우 나타나는 빨간색 오류 상자 위로 커서를 이동하고 로그를 선택합니다. 오류가 발생한 파이프라인 단계의 로그가 나타납니다. 대부분의 오류는 잘못된 파라미터 설정으로 인해 발생합니다.</p>	<p>DevOps 엔지니어, AWS 시스템 관리자</p>

관련 리소스

- [DevOps SAP의 경우 — SAP 설치: 2개월에서 2시간](#) (DevOps 엔터프라이즈 서밋 비디오 라이브러리)

AWS CDK를 사용하여 AWS Service Catalog 포트폴리오 및 제품 배포 자동화

제작자: 샌딕 가완데 (AWS), 라즈네시 야기 (AWS), 비오마 사흐데바 (AWS)

코드 리포지토리: aws-cdk-servicelog-automation	환경: PoC 또는 파일럿	기술: DevOps; 인프라, 관리 및 거버넌스
워크로드: 오픈 소스	AWS 서비스: AWS Service Catalog, AWS CDK	

요약

AWS Service Catalog를 사용하면 조직의 AWS 환경에서 사용하도록 승인된 IT 서비스 또는 제품 카탈로그를 중앙에서 관리할 수 있습니다. 제품 컬렉션을 포트폴리오라고 하며 포트폴리오에는 구성 정보도 포함됩니다. AWS Service Catalog를 사용하면 조직의 각 사용자 유형에 대해 사용자 지정 포트폴리오를 생성한 다음 해당 포트폴리오에 대한 액세스 권한을 부여할 수 있습니다. 그러면 해당 사용자는 포트폴리오 내에서 필요한 모든 제품을 신속하게 배포할 수 있습니다.

다중 리전 및 다중 계정 아키텍처와 같은 복잡한 네트워킹 인프라를 사용하는 경우 단일의 중앙 계정으로 Service Catalog 포트폴리오를 만들고 관리하는 것이 좋습니다. 이 패턴은 AWS Cloud Development Kit(AWS CDK)를 사용하여 중앙 계정에서 Service Catalog 포트폴리오를 자동으로 생성하고, 최종 사용자에게 포트폴리오에 대한 액세스 권한을 부여한 다음, 선택적으로 하나 이상의 대상 AWS 계정에서 제품을 프로비저닝하는 방법을 설명합니다. 이 ready-to-use 솔루션은 소스 계정에 Service Catalog 포트폴리오를 생성합니다. 또한 선택적으로 AWS CloudFormation Stacks를 사용하여 대상 계정의 제품을 프로비저닝하고 제품에 TagOptions 맞게 구성하는 데 도움이 됩니다.

- AWS CloudFormation StackSets — 여러 AWS 지역 및 계정에서 Service Catalog 제품을 시작하는 StackSets 데 사용할 수 있습니다. 이 솔루션에서는 이 솔루션을 배포할 때 제품을 자동으로 프로비저닝할 수 있는 옵션이 있습니다. 자세한 내용은 [AWS 사용 CloudFormation StackSets](#) (Service Catalog 설명서) 및 [StackSets 개념](#) (CloudFormation 설명서) 을 참조하십시오.
- TagOption 라이브러리 — 라이브러리를 사용하여 TagOption 프로비저닝된 제품의 태그를 관리할 수 있습니다. TagOptionA는 AWS Service Catalog에서 관리되는 키-값 쌍입니다. AWS 태그는 아니지만, 를 기반으로 AWS 태그를 생성하기 위한 템플릿 역할을 TagOption 합니다. 자세한 내용은 [TagOption 라이브러리](#) (Service Catalog 설명서) 를 참조하십시오.

사전 조건 및 제한 사항

사전 조건

- Service Catalog 포트폴리오 관리를 위한 소스 계정으로 사용할 활성 AWS 계정.
- 이 솔루션을 사용하여 하나 이상의 대상 계정에 제품을 프로비저닝하는 경우 대상 계정이 이미 존재하고 활성화되어 있어야 합니다.
- AWS 서비스 카탈로그, AWS 및 AWS IAM에 액세스할 수 있는 AWS ID 및 액세스 관리 (IAM) 권한 CloudFormation

제품 버전

- AWS CDK 버전 2.27.0

아키텍처

대상 기술 스택

- 중앙 집중식 AWS 계정의 Service Catalog 포트폴리오
- 대상 계정에 배포된 Service Catalog 제품

대상 아키텍처

1. 포트폴리오(또는 소스) 계정에서 config.json 파일을 사용 사례에 맞는 AWS 계정, AWS 리전, IAM 역할, 포트폴리오 및 제품 정보로 업데이트합니다.
2. AWS CDK 애플리케이션을 배포합니다.
3. AWS CDK 애플리케이션은 배포 IAM 역할을 맡아 config.json 파일에 정의된 Service Catalog 포트폴리오 및 제품을 생성합니다.

대상 계정에 제품을 StackSets 배포하도록 구성한 경우 프로세스가 계속됩니다. 제품을 StackSets 프로비저닝하도록 구성하지 않은 경우 프로세스가 완료됩니다.

4. AWS CDK 애플리케이션은 StackSet 관리자 역할을 맡아 config.json 파일에 정의한 AWS CloudFormation 스택 세트를 배포합니다.
5. 대상 계정에서 실행 역할을 StackSets 맡고 제품을 프로비저닝합니다. StackSet

도구

서비스

- [AWS Cloud Development Kit\(AWS CDK\)](#)는 AWS 클라우드 인프라를 코드로 정의하고 프로비저닝하는 데 도움이 되는 소프트웨어 개발 프레임워크입니다.
- [AWS CDK 툴킷](#)은 AWS CDK 앱과 상호 작용하는 데 도움이 되는 명령줄 클라우드 개발 키트입니다.
- [AWS는 AWS CloudFormation](#) 리소스를 설정하고, 빠르고 일관되게 프로비저닝하고, AWS 계정 및 지역 전반의 수명 주기 전반에 걸쳐 리소스를 관리할 수 있도록 지원합니다.
- [AWS Identity and Access Management\(IAM\)](#)은 누구에게 인증 및 사용 권한이 있는지 제어하여 AWS 리소스에 대한 액세스를 안전하게 관리할 수 있도록 도와줍니다.
- [AWS Service Catalog](#)를 사용하면 AWS에 승인된 IT 서비스의 카탈로그를 중앙에서 관리할 수 있습니다. 최종 사용자는 조직에서 규정한 제약에 따라, 필요에 따라 승인된 IT 서비스만 신속하게 배포할 수 있습니다.

코드 리포지토리

이 패턴의 코드는 [aws-cdk-servicecatalog-automation](#) 리포지토리에서 GitHub 사용할 수 있습니다. 코드 리포지토리에는 다음 파일과 폴더가 포함되어 있습니다.

- `cdk-sevicecatalog-app`— 이 폴더에는 이 솔루션을 위한 AWS CDK 애플리케이션이 들어 있습니다.
- `config` - 이 폴더에는 `config.json` 파일과 Service Catalog 포트폴리오의 제품을 배포하기 위한 CloudFormation 템플릿이 들어 있습니다.
- `config/config.json` - 이 파일에는 모든 구성 정보가 들어 있습니다. 이 파일을 업데이트하여 사용 사례에 맞게 이 솔루션을 사용자 지정합니다.
- `구성/템플릿` - 이 폴더에는 서비스 센터 제품용 템플릿이 들어 있습니다. CloudFormation
- `setup.sh` - 이 스크립트는 솔루션을 배포합니다.
- `uninstall.sh` - 이 스크립트는 이 솔루션을 배포할 때 생성된 스택과 모든 AWS 리소스를 삭제합니다.

샘플 코드를 사용하려면 [에픽](#) 섹션의 지침을 따르십시오.

모범 사례

- 이 솔루션을 배포하는 데 사용되는 IAM 역할은 [최소 권한 원칙](#)(IAM 설명서)을 준수해야 합니다.
- [AWS CDK를 사용한 클라우드 애플리케이션 개발 모범 사례](#)(AWS 블로그 게시물)를 준수하십시오.

- [AWS CloudFormation 모범 사례](#) (CloudFormation 설명서) 를 준수하십시오.

에픽

환경을 설정합니다.

작업	설명	필요한 기술
AWS CDK 툴킷을 설치합니다.	<p>AWS CDK 툴킷이 설치되어 있는지 확인하십시오. 다음 명령을 입력하여 설치 여부를 확인하고 버전을 확인합니다.</p> <pre>cdk --version</pre> <p>AWS CDK가 설치되지 않은 경우에는 다음 명령을 실행하여 설치합니다.</p> <pre>npm install -g aws-cdk@2.27.0</pre> <p>AWS CDK 툴킷 버전이 2.27.0 이전인 경우 다음 명령을 입력하여 버전 2.27.0으로 업데이트하십시오.</p> <pre>npm install -g aws-cdk@2.27.0 --force</pre>	AWS DevOps, DevOps 엔지니어
리포지토리를 복제합니다.	<p>다음 명령을 입력하십시오. 추가 정보 섹션의 리포지토리 복제에서 리포지토리의 URL이 포함된 전체 명령을 복사할 수 있습니다. 이렇게 하면 aws-cdk-servicecatalog-auto</p>	AWS DevOps, DevOps 엔지니어

작업	설명	필요한 기술
	<p>automation 리포지토리가 복제됩니다. GitHub</p> <pre>git clone <repository-URL>.git</pre> <p>그러면 대상 디렉터리에 <code>cd aws-cdk-servicecat alog-automation</code> 폴더가 생성됩니다. 다음 명령을 입력하여 이 폴더로 이동합니다.</p> <pre>cd aws-cdk-servicecat alog-automation</pre>	
<p>AWS 보안 인증을 설정합니다.</p>	<p>다음 명령을 입력합니다. 이들은 스택을 배포하는 AWS 계정 및 리전을 정의하는 다음 변수를 내보냅니다.</p> <pre>export CDK_DEFAULT_ACCOUNT=<12-digit AWS account number></pre> <pre>export CDK_DEFAULT_REGION=<AWS Region></pre> <p>AWS CDK용 AWS 보안 인증은 환경 변수를 통해 제공됩니다.</p>	<p>AWS DevOps, DevOps 엔지니어</p>

작업	설명	필요한 기술
최종 사용자 IAM 역할에 대해 권한을 구성하십시오.	IAM 역할을 사용하여 포트폴리오와 포트폴리오에 포함된 제품에 대한 액세스 권한을 부여하려는 경우 역할에 <code>servicecatalog.amazonaws.com</code> 서비스 보안 주체가 수입할 수 있는 권한이 있어야 합니다. 이러한 권한을 부여하는 방법에 대한 지침은 Service Catalog를 통한 신뢰할 수 있는 액세스 활성화 (AWS Organizations 설명서)를 참조하십시오.	AWS DevOps, DevOps 엔지니어

작업	설명	필요한 기술
<p>에 StackSets 필요한 IAM 역할을 구성합니다.</p>	<p>를 사용하여 StackSets 대상 계정의 제품을 자동으로 프로비저닝하려면 스택 세트를 관리하고 실행하는 IAM 역할을 구성해야 합니다.</p> <ol style="list-style-type: none"> 1. 소스 계정에서 <code>AWSCloudFormationStackSetAdministrationRole</code> 이 이미 존재하는지 확인하십시오. 대상 계정에서 <code>AWSCloudFormationStackSetExecutionRole</code> 이 이미 존재하는지 확인하십시오. 이러한 역할이 이미 존재하는 경우 다음에 픽으로 건너뛸 수 있습니다. 2. 서비스 관리형 권한 부여(IAM 설명서)의 지침에 따라 포트폴리오 계정에서 스택 세트 관리 역할을 생성하고 각 대상 계정에서 실행 역할을 생성하십시오. 	<p>AWS DevOps, DevOps 엔지니어</p>

솔루션 사용자 지정 및 배포

작업	설명	필요한 기술
<p>CloudFormation 템플릿 생성.</p>	<p>포트폴리오에 포함하려는 모든 제품의 CloudFormation 템플릿을 <code>config/templates</code> 폴더에서 생성합니다. 자세한 내용은 AWS CloudFormation 템</p>	<p>앱 개발자, AWS DevOps, DevOps 엔지니어</p>

작업	설명	필요한 기술
	플릿 사용 (CloudFormation 설치 명서) 을 참조하십시오.	

작업	설명	필요한 기술
구성 파일을 사용자 지정합니다.	<p>config 폴더에서 config.json 파일을 열고 사용 사례에 적합한 파라미터를 정의합니다. 달리 명시되지 않는 한 다음 파라미터가 요구됩니다.</p> <ul style="list-style-type: none"> • <code>portfolios</code> 섹션에서 다음 파라미터를 정의하여 하나 이상의 Service Catalog 포트폴리오를 생성합니다. <ul style="list-style-type: none"> • <code>portfolioName</code> - 포트폴리오의 이름. • <code>providerName</code> - 포트폴리오를 관리하는 사람, 팀 또는 조직의 이름. • <code>description</code> - 포트폴리오에 대한 간단한 설명. • <code>roles</code> - (선택 사항) 이 포트폴리오에 액세스할 수 있어야 하는 모든 IAM 역할의 이름. 이 역할을 가진 사용자는 이 포트폴리오의 제품에 액세스할 수 있습니다. • <code>users</code> - (선택 사항) 이 포트폴리오와 이 포트폴리오의 제품에 액세스할 수 있어야 하는 모든 IAM 사용자의 이름. • <code>groups</code> - (선택 사항) 이 포트폴리오와 이 포트폴리오의 제품에 액세스할 수 	앱 개발자, DevOps 엔지니어, AWS DevOps

작업	설명	필요한 기술
	<p>있어야 하는 모든 IAM 사용자 그룹의 이름.</p> <p>경고: IAM 사용자는 장기 자격 증명을 보유하므로 보안 위험이 발생할 수 있습니다. 이 위험을 줄이려면 이러한 사용자에게 작업을 수행하는데 필요한 권한만 제공하고 더 이상 필요하지 않을 경우 이러한 사용자를 제거하는 것이 좋습니다.</p> <p>중요: roles, users 및 groups는 모두 선택적 파라미터이지만 이러한 파라미터 중 하나를 정의하지 않으면 아무도 Service Catalog 콘솔에서 포트폴리오 제품을 볼 수 없습니다. 다음 파라미터 중 하나 이상을 정의합니다. 자세한 내용은 Service Catalog 최종 사용자에게 권한 부여(Service Catalog 설명서)를 참조하십시오.</p> <ul style="list-style-type: none"> (선택 사항) tagOption 섹션에서 제품에 TagOptions 대해 다음을 정의합니다. <ul style="list-style-type: none"> key— TagOption 키 이름 value— 에 대해 허용된 문자열 값 TagOption <p>자세한 내용은 TagOption 라이브러리 (Service Catalog 설명서) 를 참조하십시오.</p>	

작업	설명	필요한 기술
	<ul style="list-style-type: none"> • <code>products</code> 섹션에서 제품에 대해 다음 파라미터를 정의합니다. • <code>portfolioName</code> - 제품을 추가하려는 포트폴리오의 이름. 포트폴리오는 하나만 지정할 수 있습니다. • <code>productName</code> - 제품의 이름. • <code>owner</code> - 제품의 소유자. • <code>productVersionName</code> - 문자열 값의 제품 버전 이름(예: v1). • <code>templatePath</code> — 제품 CloudFormation 템플릿의 파일 경로입니다. • <code>deployWithStackSets</code> — (선택 사항) 포트폴리오에서 제품을 자동으로 프로비전하는 StackSets 데 사용할 하나 이상의 계정 및 지역을 지정합니다. 이 배포 옵션을 사용하는 경우 이 섹션의 다음 파라미터가 모두 필요합니다. <ul style="list-style-type: none"> • <code>accounts</code> - 대상 계정. • <code>regions</code> - 대상 리전. • <code>stackSetAdministrationRoleName</code> — 구성을 관리하는 데 사용되는 IAM 역할의 이름. StackSets 이 값은 변경 	

작업	설명	필요한 기술
	<p>하지 마십시오. 이 역할에는 정확한 이름이 있어야 합니다.</p> <ul style="list-style-type: none"> stackSetExecutionRoleName - 스택 인스턴스를 배포하는 대상 계정의 IAM 역할 이름. 이 값은 변경하지 마십시오. 이 역할에는 정확한 이름이 있어야 합니다. <p>완성된 구성 파일의 예는 추가 정보 섹션의 샘플 구성 파일을 참조하십시오.</p>	
<p>솔루션을 배포합니다.</p>	<p>다음 명령을 입력하십시오. 그러면 AWS CDK 앱이 배포되고 config.json 파일에 지정된 대로 Service Catalog 포트폴리오 및 제품이 프로비저닝됩니다.</p> <pre>sh +x setup.sh</pre>	<p>앱 개발자, DevOps 엔지니어, AWS DevOps</p>

작업	설명	필요한 기술
<p>배포를 확인합니다.</p>	<p>다음을 수행하여 배포가 성공했는지 확인하십시오.</p> <ol style="list-style-type: none"> 1. 구성 파일에 정의한 하나 이상의 포트폴리오에 액세스할 수 있는 보안 인증으로 AWS Management Console에 로그인합니다. 2. Service Quotas 콘솔(https://console.aws.amazon.com/servicecatalog/)을 엽니다. 3. 탐색 창의 프로비저닝에서 제품을 선택합니다. 포트폴리오에 지정한 제품 목록이 표시되는지 확인합니다. 4. 제품 시작 지침(Service Catalog 설명서)에 따라 사용할 가능한 제품 중 하나를 실행하십시오. 사용할 가능한 제품 버전 및 태그가 구성 파일에 제공한 값과 일치하는지 확인하십시오. 5. 를 사용하여 StackSets 하나 이상의 대상 계정에 제품을 자동으로 프로비저닝하도록 선택한 경우 다음을 수행하십시오. <ol style="list-style-type: none"> a. 대상 계정 중 하나에서 프로비저닝된 제품을 볼 수 있는 권한을 부여하는 보안 인증으로 로그인하십시오. b. Service Catalog 콘솔의 탐색 창에서 프로비저 	<p>일반 AWS</p>

작업	설명	필요한 기술
<p>(선택 사항) 포트폴리오와 제품을 업데이트하십시오.</p>	<p>아래 프로비저닝된 제품을 선택합니다.</p> <p>c. 목록에 예상 제품이 나타나는지 확인합니다.</p> <p>이 솔루션을 사용하여 포트폴리오 또는 제품을 업데이트하거나 새 제품을 제공하려는 경우.</p> <ol style="list-style-type: none"> 1. config.json 파일에서 필요에 따라 변경합니다. 2. config/template 폴더에 필요에 따라 CloudFormation 템플릿을 추가하거나 수정하십시오. 3. 솔루션을 재배포합니다. <p>예를 들면 포트폴리오를 추가하거나 리소스를 더 프로비저닝할 수 있습니다. AWS CDK 앱은 변경 사항만 구현합니다. 이전에 배포한 포트폴리오 또는 제품에 변경 사항이 없는 경우 재배포는 영향을 받지 않습니다.</p>	<p>앱 개발자, DevOps 엔지니어, 일반 AWS</p>

솔루션 정리

작업	설명	필요한 기술
<p>(선택 사항) 이 솔루션에서 배포한 AWS 리소스를 제거합니다.</p>	<p>프로비저닝된 제품을 삭제하려면 프로비저닝된 제품 삭제</p>	<p>AWS DevOps, DevOps 엔지니어, 앱 개발자</p>

작업	설명	필요한 기술
	<p>제(Service Catalog 설명서)의 지침을 따르십시오.</p> <p>이 솔루션으로 생성된 리소스를 모두 삭제하려면 다음 명령을 입력하십시오.</p> <pre>sh uninstall.sh</pre>	

관련 리소스

- [AWS Service Catalog 구성 라이브러리](#)(AWS API 참조)
- [StackSets 개념](#) (CloudFormation 설명서)
- [AWS Service Catalog](#)(AWS 마케팅)
- [AWS CDK와 함께 Service Catalog 사용](#)(AWS 워크숍)

추가 정보

추가 정보

리포지토리 복제

다음 명령을 입력하여 리포지토리를 GitHub 복제합니다.

```
git clone https://github.com/aws-samples/aws-cdk-servicecatalog-automation.git
```

샘플 구성 파일

다음은 예제 값이 포함된 샘플 config.json 파일입니다.

```
{
  "portfolios": [
    {
      "displayName": "EC2 Product Portfolio",
      "providerName": "User1",
      "description": "Test1",

```

```
    "roles": [
      "<Names of IAM roles that can access the products>"
    ],
    "users": [
      "<Names of IAM users who can access the products>"
    ],
    "groups": [
      "<Names of IAM user groups that can access the products>"
    ]
  },
  {
    "displayName": "Autoscaling Product Portfolio",
    "providerName": "User2",
    "description": "Test2",
    "roles": [
      "<Name of IAM role>"
    ]
  }
],
"tagOption": [
  {
    "key": "Group",
    "value": [
      "finance",
      "engineering",
      "marketing",
      "research"
    ]
  },
  {
    "key": "CostCenter",
    "value": [
      "01",
      "02",
      "03",
      "04"
    ]
  },
  {
    "key": "Environment",
    "value": [
      "dev",
      "prod",
      "stage"
    ]
  }
]
```

```
    ]
  }
],
"products": [
  {
    "portfolioName": "EC2 Product Profile",
    "productName": "Ec2",
    "owner": "owner1",
    "productVersionName": "v1",
    "templatePath": "../../config/templates/template1.json"
  },
  {
    "portfolioName": "Autoscaling Product Profile",
    "productName": "autoscaling",
    "owner": "owner1",
    "productVersionName": "v1",
    "templatePath": "../../config/templates/template2.json",
    "deployWithStackSets": {
      "accounts": [
        "012345678901",
      ],
      "regions": [
        "us-west-2"
      ],
      "stackSetAdministrationRoleName":
"AWSCloudFormationStackSetAdministrationRole",
      "stackSetExecutionRoleName": "AWSCloudFormationStackSetExecutionRole"
    }
  }
]
}
```

및 이벤트를 사용하여 CodeBuild Amazon CodeCommit S3로 이벤트 기반 백업을 자동화합니다. CloudWatch

작성자: Kirankumar Chandrashekar(AWS)

환경: 프로덕션

기술: DevOps; 스토리지 및 백업

워크로드: 기타 모든 워크로드

AWS 서비스: 아마존 S3, 아마존 CloudWatch, AWS CodeBuild, AWS CodeCommit

요약

Amazon Web Services (AWS) 클라우드에서는 AWS를 사용하여 안전한 Git 기반 CodeCommit 리포지토리를 호스팅할 수 있습니다. CodeCommit 완전 관리형 소스 제어 서비스입니다. 그러나 CodeCommit 저장소가 실수로 삭제되면 해당 내용도 삭제되며 [복원할 수 없습니다](#).

이 패턴은 CodeCommit 리포지토리를 변경한 후 Amazon Simple Storage Service (Amazon S3) 버킷에 리포지토리를 자동으로 백업하는 방법을 설명합니다. CodeCommit 리포지토리가 나중에 삭제되는 경우 이 백업 전략은 point-in-time 복구 옵션을 제공합니다.

사전 조건 및 제한 사항

사전 조건

- 활성 상태의 AWS 계정.
- 요구 사항에 따라 사용자 액세스가 구성된 기존 CodeCommit 리포지토리 자세한 내용은 CodeCommit 설명서의 [AWS CodeCommit 설정](#)을 참조하십시오.
- CodeCommit 백업 업로드를 위한 S3 버킷.

제한 사항

- 이 패턴은 모든 CodeCommit 리포지토리를 자동으로 백업합니다. 개별 CodeCommit 리포지토리를 백업하려면 Amazon CloudWatch Events 규칙을 수정해야 합니다.

아키텍처

이 다이어그램은 이 패턴의 워크플로우를 보여줍니다.

이 워크플로우는 다음 단계로 구성됩니다.

1. 코드가 리포지토리로 CodeCommit 푸시됩니다.
2. CodeCommit 리포지토리는 리포지토리 변경 (예: `git push` 명령) 을 CloudWatch Events에 알립니다.
3. CloudWatch 이벤트는 AWS를 CodeBuild 호출하여 CodeCommit 리포지토리 정보를 보냅니다.
4. CodeBuild 전체 CodeCommit 리포지토리를 복제하고.zip 파일로 패키징합니다.
5. CodeBuild .zip 파일을 S3 버킷에 업로드합니다.

기술 스택

- CloudWatch 이벤트
- CodeBuild
- CodeCommit
- Amazon S3

도구

- [Amazon CloudWatch CloudWatch Events](#) — Events는 AWS 리소스의 변경 사항을 설명하는 시스템 이벤트의 스트림을 거의 실시간으로 제공합니다.
- [AWS CodeBuild](#) — 소스 코드를 컴파일하고, 테스트를 실행하고, 배포 준비가 완료된 소프트웨어 패키지를 생성하는 완전관리형 지속적 통합 CodeBuild 서비스입니다.
- [AWS CodeCommit](#) — 안전한 Git 기반 리포지토리를 호스팅하는 완전 관리형 소스 제어 CodeCommit 서비스입니다.
- [AWS Identity and Access Management\(IAM\)](#) – IAM은 AWS 리소스에 대한 사용자의 액세스를 안전하게 제어할 수 있게 지원하는 웹 서비스입니다.
- [Amazon S3](#) – Amazon Simple Storage Service(S3)는 인터넷에 대한 스토리지입니다.

에픽

프로젝트 생성 CodeBuild

작업	설명	필요한 기술
CodeBuild 서비스 역할을 생성하세요.	AWS Management Console에 로그인하고 IAM 콘솔을 엽니다. 역할을 선택하고 역할 생성을 선택합니다. CodeCommit 리포지토리를 CodeBuild 복제하고, S3 버킷에 파일을 업로드하고, Amazon에 로그를 전송할 서비스 역할을 생성합니다 CloudWatch. 자세한 내용은 CodeBuild 설명서의 CodeBuild 서비스 역할 생성을 참조하십시오 .	클라우드 관리자
CodeBuild 프로젝트 만들기.	CodeBuild 콘솔에서 CodeBuild 프로젝트 생성을 선택합니다. 추가 정보 섹션의 buildspec.yml 템플릿을 사용하여 CodeBuild 프로젝트를 생성합니다. 이 스토리에 대한 도움이 필요하면 CodeBuild 설명서의 빌드 프로젝트 만들기 를 참조하십시오.	클라우드 관리자

CloudWatch 이벤트 규칙 생성 및 구성

작업	설명	필요한 기술
CloudWatch 이벤트에 대한 IAM 역할을 생성합니다.	IAM 콘솔에서 [Roles]를 선택하고 이벤트에 대한 IAM 역할을 생성합니다. CloudWatch에 대한 자세한 내용은 IAM	클라우드 관리자

작업	설명	필요한 기술
	<p>설명서의 CloudWatch 이벤트 IAM 역할을 참조하십시오.</p> <p>중요: 이벤트의 IAM 역할에 <code>codebuild:StartBuild</code> 권한을 추가해야 합니다.</p> <p>CloudWatch</p>	

작업	설명	필요한 기술
CloudWatch 이벤트 규칙 생성.	<ol style="list-style-type: none"> 1. CloudWatch 콘솔에서 이벤트를 선택한 다음 규칙을 선택합니다. 규칙 생성을 선택하고 추가 정보 섹션의 CloudWatch 이벤트 규칙을 사용합니다. 그러면 CodeCommit 리포지토리의 이벤트 변경 (예: git push 또는 git commit 명령) 을 수신하는 규칙이 생성됩니다. 자세한 내용은 AWS CodePipeline 설명서의 CodeCommit 소스에 대한 CloudWatch 이벤트 규칙 생성을 참조하십시오. 2. 대상을 선택하고 주제를 선택한 다음 입력 구성을 선택합니다. 입력 변환기를 선택하고 추가 정보 섹션의 입력 경로와 입력 템플릿을 사용합니다. 이렇게 하면 CodeCommit 리포지토리 세부 정보가 파싱되어 CodeBuild 프로젝트에 환경 변수로 전송됩니다. 자세한 내용은 CloudWatch 설명서의 입력 변환기 튜토리얼을 참조하십시오. 3. 세부 정보 구성을 선택하고 규칙에 대한 이름과 설명을 입력합니다. 규칙 생성(Create rule)을 선택합니다. 	클라우드 관리자

작업	설명	필요한 기술
	중요: 이 CloudWatch 이벤트 규칙은 모든 CodeCommit 리포지토리의 변경 사항을 설명합니다. 개별 CodeCommit 리포지토리를 백업하거나 다른 리포지토리 백업에 별도의 S3 버킷을 사용하려면 CloudWatch 이벤트 규칙을 수정해야 합니다.	

관련 리소스

프로젝트 생성 CodeBuild

- [CodeBuild 서비스 역할 생성](#)
- [CodeBuild 프로젝트 생성](#)
- [Git 클라이언트 명령을 위한 필수 권한](#)

CloudWatch 이벤트 규칙 생성 및 구성

- [CodeCommit 소스에 대한 CloudWatch 이벤트 규칙 생성](#)
- [입력 변환기를 사용하여 이벤트 대상에 전달되는 내용을 사용자 지정](#)
- [CloudWatch 이벤트에서 시작되는 이벤트 규칙 만들기](#)
- [CloudWatch 이벤트 IAM 역할 생성](#)

추가 정보

CodeBuild buildspec.yml 템플릿

```
version: 0.2
phases:
  install:
    commands:
      - pip install git-remote-codecommit
```

```

build:
  commands:
    - env
    - git clone -b $REFERENCE_NAME codecommit::$REPO_REGION://$REPOSITORY_NAME
    - dt=$(date '+%d-%m-%Y-%H:%M:%S');
    - echo "$dt"
    - zip -yr $dt-$REPOSITORY_NAME-backup.zip ./
    - aws s3 cp $dt-$REPOSITORY_NAME-backup.zip s3:// #substitute a valid S3 Bucket
      Name here

```

CloudWatch 이벤트 규칙

```

{
  "source": [
    "aws.codecommit"
  ],
  "detail-type": [
    "CodeCommit Repository State Change"
  ],
  "detail": {
    "event": [
      "referenceCreated",
      "referenceUpdated"
    ]
  }
}

```

CloudWatch 이벤트 규칙 타겟용 샘플 입력 변환기

입력 경로:

```

{"referenceType":"$.detail.referenceType","region":"$.region","repositoryName":"$.detail.repositoryName"}

```

입력 템플릿(값을 적절히 입력하세요):

```

{
  "environmentVariablesOverride": [
    {
      "name": "REFERENCE_NAME",
      "value": ""
    },
    {

```

```
        "name": "REFERENCE_TYPE",
        "value": ""
    },
    {
        "name": "REPOSITORY_NAME",
        "value": ""
    },
    {
        "name": "REPO_REGION",
        "value": ""
    },
    {
        "name": "ACCOUNT_ID",
        "value": ""
    }
]
}
```

AWS CodePipeline 및 AWS를 사용하여 스택 세트 배포를 자동화합니다. CodeBuild

작성자: Thiyagarajan Mani, Mihir Borkar, Raghu Gowda

코드 리포지토리: [automated-code-pipeline-stackset -배포](#)

환경: 프로덕션

기술: DevOps; 소프트웨어 개발 및 테스트

AWS 서비스: AWS, AWS CodeBuild, AWS CodeCommit, AWS CodePipeline, AWS Organizations, AWS CloudFormation

요약

지속적 통합 및 지속적 전달(CI/CD) 프로세스에서는 기존의 모든 AWS 계정 및 Organizations의 조직에 추가한 새 계정에 애플리케이션을 자동으로 배포하고자 할 수 있습니다. 이 요구 사항에 맞게 CI/CD 솔루션을 설계할 때 AWS의 [위임 스택 세트 관리자](#) CloudFormation 기능은 관리 계정에 대한 액세스를 제한하여 보안 계층을 활성화하므로 유용합니다. 하지만 CodePipeline AWS는 서비스 관리형 권한 모델을 사용하여 애플리케이션을 여러 계정 및 지역에 배포합니다. AWS는 위임 스택 세트 관리자 기능을 CodePipeline 지원하지 않으므로 스택 세트와 함께 배포하려면 AWS Organizations 관리 계정을 사용해야 합니다.

이 패턴은 이러한 제한을 해결할 수 있는 방법을 설명합니다. 이 패턴은 CodeBuild AWS와 사용자 지정 스크립트를 사용하여 AWS에서의 스택 세트 배포를 자동화합니다 CodePipeline. 다음과 같은 애플리케이션 배포 활동을 자동화합니다.

- 애플리케이션을 기존 조직 단위(OU)에 스택 세트로 배포하기
- 애플리케이션 배포를 추가 OU 및 리전으로 확장하기
- 모든 또는 특정 OU나 리전에서 배포된 애플리케이션을 제거하기

사전 조건 및 제한 사항

사전 조건

이 패턴의 절차를 따르기 전에:

- Organizations 관리 계정에서 조직을 생성합니다. 지침은 [AWS Organizations 설명서](#)를 참조하십시오.
- AWS Organizations 간에 신뢰할 수 있는 액세스를 지원하고 서비스 관리 권한을 사용할 CloudFormation 수 있습니다. 지침은 설명서에서 [AWS Organizations를 통한 신뢰할 수 있는 액세스 활성화](#)를 CloudFormation 참조하십시오.

제한 사항

이 패턴과 함께 제공되는 코드에는 다음과 같은 제한이 있습니다.

- 애플리케이션에는 단일 CloudFormation 템플릿만 배포할 수 있습니다. 다중 템플릿 배포는 현재 지원되지 않습니다.
- 현재 구현을 사용자 지정하려면 DevOps 전문 지식이 필요합니다.
- 이 패턴은 AWS 키 관리 시스템(AWS KMS) 키를 사용하지 않습니다. 하지만 이 패턴에 포함된 CloudFormation 템플릿을 재구성하여 이 기능을 활성화할 수 있습니다.

아키텍처

이 CI/CD 배포 파이프라인 아키텍처는 다음을 처리합니다.

- 애플리케이션 배포를 위한 스택 세트 관리자인 전용 CI/CD 계정에 스택 세트 배포 책임을 위임하여 관리 계정에 대한 직접 액세스를 제한합니다.
- 서비스 관리형 권한 모델을 사용하여, 새 계정을 생성하고 OU에 매핑할 때마다 애플리케이션을 자동으로 배포합니다.
- 환경 수준에서 모든 계정 전반적으로 애플리케이션 버전 일관성을 보장합니다.
- 리포지토리 및 파이프라인 수준에서 여러 승인 단계를 사용하여 배포된 애플리케이션에 보안 및 거버넌스의 추가된 계층을 제공합니다.
- 에서 CodeBuild 사용자 지정 배포 스크립트를 사용하여 스택 세트 및 스택 인스턴스를 자동으로 배포하거나 CodePipeline 제거함으로써 기존의 한계를 극복합니다. 사용자 지정 스크립트로 구현된 API 직접 호출의 플로 제어 및 계층 구조에 대한 그림은 [추가 정보](#) 섹션을 참조하십시오.
- 개발, 테스트, 프로덕션 환경을 위한 개별 스택 세트를 생성합니다. 또한 모든 단계에서 다중 OU와 리전을 결합하는 스택 세트를 생성할 수 있습니다. 예를 들어, 개발 배포 단계 내에서 샌드박스 및 개발 OU를 결합할 수 있습니다.

- 계정 또는 OU 목록의 하위 집합에 애플리케이션을 배포 또는 제외할 수 있도록 지원합니다.

자동화 및 규모 조정

이 패턴과 함께 제공된 코드를 사용하여 애플리케이션을 위한 AWS CodeCommit 리포지토리와 코드 파이프라인을 생성할 수 있습니다. 그런 다음 이를 OU 수준에서 여러 계정에 스택 세트로서 배포할 수 있습니다. 또한 이 코드는 승인자에게 알릴 Amazon Simple Notification Service(SNS) 주제, 필수 Identity and Access Management(IAM) 역할, 그리고 관리 계정에서 적용할 서비스 제어 정책(SCP)와 같은 구성요소를 자동화합니다.

도구

서비스

- [AWS는 AWS CloudFormation](#) 리소스를 설정하고, 빠르고 일관되게 프로비저닝하고, AWS 계정 및 지역 전반의 수명 주기 전반에 걸쳐 리소스를 관리할 수 있도록 지원합니다.
- [CodeBuildAWS](#)는 소스 코드를 컴파일하고, 단위 테스트를 실행하고, 배포할 준비가 된 아티팩트를 생성하는 데 도움이 되는 완전관리형 빌드 서비스입니다.
- [CodeCommitAWS](#)는 자체 소스 제어 시스템을 관리할 필요 없이 Git 리포지토리를 비공개로 저장하고 관리할 수 있는 버전 제어 서비스입니다.
- [AWS](#)는 아마존 엘라스틱 CodeDeploy 컴퓨터 클라우드 (Amazon EC2) 또는 온프레미스 인스턴스, AWS Lambda 함수 또는 아마존 Elastic Container Service (아마존 ECS) 서비스로의 배포를 자동화합니다.
- [AWS](#)는 소프트웨어 릴리스의 여러 단계를 신속하게 모델링 및 구성하고 소프트웨어 변경 사항을 지속적으로 릴리스하는 CodePipeline 데 필요한 단계를 자동화하도록 지원합니다.
- [Organizations](#)는 여러 계정을 사용자가 생성하고 중앙에서 관리하는 단일 조직으로 통합할 수 있는 계정 관리 서비스입니다.
- [Amazon Simple Notification Service\(Amazon SNS\)](#)를 사용하면 웹 서버 및 이메일 주소를 포함하여 게시자와 클라이언트 간의 메시지 교환을 조정하고 관리할 수 있습니다.

코드 리포지토리

이 패턴의 코드는 GitHub [automated-code-pipeline-stackset-deploy](#) 리포지토리에서 사용할 수 있습니다. 폴더 구조 및 기타 세부 정보는 리포지토리에 대한 [readme 파일](#)을 참조하십시오.

모범 사례

이 패턴은 OU 수준에서 애플리케이션을 배포하는 동안 관리 계정에 대한 직접 액세스를 감독합니다. 파이프라인 및 리포지토리 프로세스에 다중 승인 단계를 추가하면 이 접근 방식을 사용하여 배포하는 애플리케이션과 구성 요소에 대한 추가적인 보안 및 거버넌스를 제공하는데 도움이 됩니다.

에픽

AWS Organizations에서 계정 구성

작업	설명	필요한 기술
관리 계정에서 모든 기능을 활성화합니다.	Organizations 설명서 의 지침에 따라 조직의 관리 계정에 있는 모든 기능을 활성화합니다.	AWS 관리자, 플랫폼 관리자
CI/CD 계정을 생성합니다.	조직의 AWS Organizations에서 전용 CI/CD 계정을 생성하고, 계정에 대한 액세스를 소유하고 제어할 팀을 지정합니다.	AWS 관리자
위임된 관리자를 추가합니다.	관리 계정에서, 이전 단계에서 생성한 CI/CD 계정을 위임된 스택 세트 관리자로 등록합니다. 지침은 AWS CloudFormation 설명서를 참조하십시오 .	AWS 관리자, 플랫폼 관리자

애플리케이션 리포지토리 및 CI/CD 파이프라인 생성

작업	설명	필요한 기술
코드 리포지토리를 복제합니다.	<ol style="list-style-type: none"> 이 패턴과 함께 제공된 코드 리포지토리를 다음과 같이 컴퓨터에 복제합니다. <pre>git clone https://github.com/aws-samples/automated-code</pre>	AWS DevOps

작업	설명	필요한 기술
	<p data-bbox="630 205 1029 306"><code>-pipeline-stackset-deployment.git</code></p> <p data-bbox="591 319 1029 453">2. readme 파일을 검토하여 디렉터리 구조 및 기타 세부 사항을 이해합니다.</p>	

작업	설명	필요한 기술
SNS 주제를 생성합니다.	<p>GitHub 리포지토리에 제공된 <code>sns-template.yaml</code> 템플릿을 사용하여 SNS 주제를 생성하고 승인 요청에 대한 구독을 구성할 수 있습니다.</p> <ol style="list-style-type: none"> 1. 콘솔에서 CI/CD 계정에 로그인합니다. 2. https://console.aws.amazon.com/cloudformation 에서 CloudFormation 콘솔을 엽니다. 3. 새 리소스를 가진 새 스택을 생성합니다(표준 옵션). 4. 템플릿 지정에서 템플릿 파일 업로드, 파일 선택을 선택한 다음 복제된 GitHub 리포지토리의 <code>templates</code> 폴더에서 <code>sns-template.yaml</code> 파일을 선택합니다. 다음을 선택합니다. 5. 의미있는 애플리케이션 스택 이름을 입력합니다. 6. 리소스의 접두사를 지정합니다. 7. 다음, 다음, 그리고 제출을 선택합니다. 8. 스택이 성공적으로 생성되면 출력 탭을 선택하고 필요 요청, 테스트 환경, 프로덕션 환경에 대한 SNS 주제의 Amazon 리소스 이름(ARN)을 기록합니다. 이 정보는 후 	AWS DevOps

작업	설명	필요한 기술
	속 단계에서 사용할 것입니다.	

작업	설명	필요한 기술
<p>CI/CD 구성 요소에 대한 IAM 역할을 생성합니다.</p>	<p>GitHub 리포지토리에 제공된 <code>cicd-role-template.yaml</code> 템플릿을 사용하여 CI/CD 구성 요소에 필요한 IAM 역할 및 정책을 생성할 수 있습니다.</p> <ol style="list-style-type: none"> 1. 콘솔에서 CI/CD 계정에 로그인합니다. 2. https://console.aws.amazon.com/cloudformation 에서 CloudFormation 콘솔을 엽니다. 3. 새 리소스를 가진 새 스택을 생성합니다(표준 옵션). 4. 템플릿 지정에서 템플릿 파일 업로드, 파일 선택을 선택한 다음 복제된 GitHub 리포지토리의 <code>templates</code> 폴더에서 <code>cicd-role-template.yaml</code> 파일을 선택합니다. 다음을 선택합니다. 5. 의미있는 애플리케이션 스택 이름을 입력합니다. 6. 다음 파라미터에 대해 값을 입력합니다. <ul style="list-style-type: none"> • 권한 경계 정책에 대한 ARN 이 ARN은 IAM 콘솔의 권한 경계 정책의 정책 세부 정보 섹션에서 얻을 수 있습니다. 	<p>AWS DevOps</p>

작업	설명	필요한 기술
	<ul style="list-style-type: none"> • 이전에 적어둔 SNS 제작 승인 주제에 대한 ARN입니다. • 이전에 적어둔 SNS 제작 승인 주제에 대한 ARN입니다. • 템플릿으로 생성된 리소스의 접두사 <p>7. 다음, 다음, 그리고 제출을 선택합니다.</p> <p>8. 스택이 성공적으로 생성되면 출력 탭을 선택하고, 생성된 IAM 역할의 ARN을 기록해 둡니다. 이 정보는 후속 단계에서 사용할 것입니다.</p>	

작업	설명	필요한 기술
<p>애플리케이션을 위한 CodeCommit 리포지토리와 코드 파이프라인을 생성하십시오.</p>	<p>리포지토리에서 제공되는 <code>cicd-pipeline-template.yaml</code> 템플릿을 사용하여 애플리케이션을 위한 GitHub CodeCommit 리포지토리와 코드 파이프라인을 만들 수 있습니다.</p> <ol style="list-style-type: none"> 1. 콘솔에서 CI/CD 계정에 로그인합니다. 2. https://console.aws.amazon.com/cloudformation 에서 CloudFormation 콘솔을 엽니다. 3. 새 리소스를 가진 새 스택을 생성합니다(표준 옵션). 4. 템플릿 지정에서 템플릿 파일 업로드, 파일 선택을 선택한 다음 복제된 GitHub 리포지토리의 <code>templates</code> 폴더에서 <code>cicd-pipeline-template.yaml</code> 파일을 선택합니다. 다음을 선택합니다. 5. 의미있는 애플리케이션 스택 이름을 입력합니다. 6. 다음 파라미터에 대해 값을 입력합니다. <ul style="list-style-type: none"> • <code>AppRepositoryName</code>— 응용 프로그램용으로 만들 CodeCommit 리포지토리의 이름. 	<p>AWS DevOps</p>

작업	설명	필요한 기술
	<ul style="list-style-type: none"> • <code>AppRepositoryDescription</code>— 응용 프로그램용으로 생성될 <code>CodeCommit</code> 저장소에 대한 간략한 설명입니다. • <code>ApplicationName</code>— 애플리케이션 이름. 이 문자열은 <code>CodeCommit</code> 리포지토리 이름 및 CI/CD 파이프라인의 접두사로 사용됩니다. • <code>CloudWatchEventRoleARN</code> — 이전 작업의 <code>CloudWatch</code> 이벤트 역할 ARN입니다. • <code>CodeBuildProjectRoleARN</code> — 이전 작업의 <code>CodeBuild</code> 프로젝트 역할 ARN입니다. • <code>CodePipelineRoleARN</code> — 이전 작업의 <code>CodePipeline</code> 역할 ARN입니다. • <code>DeploymentConfigBucket</code>— 배포 구성 파일 및 스크립트 <code>.zip</code> 파일이 저장되는 Amazon Simple S3 (Amazon S3) 버킷 이름입니다. • <code>DeploymentConfigKey</code>— 경로와 <code>.zip</code> 파일 이름 (아마존 S3 키). 	

작업	설명	필요한 기술
	<ul style="list-style-type: none"> • PRApprovalSNSARN – 풀 요청 알림을 위한 SNS 주제의 ARN입니다. • ProdApprovalSNSARN – 프로덕션 승인을 위한 SNS 주제에 대한 ARN입니다. • TESTApprovalSNSARN – 테스트 승인을 위한 SNS 주제의 ARN입니다. • TemplateBucket— CI/CD 파이프라인 생성 템플릿이 저장될 CI/CD 계정의 S3 버킷 이름. <p>7. 다음, 다음, 그리고 제출을 선택합니다.</p> <p>8. 스택이 성공적으로 완료되면 지정된 이름과 기본 디렉터리 구조, 배포 구성 파일, 스크립트 및 CodeCommit 리포지토리의 코드 파이프라인이 있는 리포지토리가 생성됩니다.</p>	

스택 세트 배포

작업	설명	필요한 기술
애플리케이션 리포지토리를 복제합니다.	이전에 사용한 CI/CD 파이프라인 템플릿은 샘플 애플리케이션 리포지토리와 코드 파이프라인을 생성합니다. 리포지토리를 복제하고 확인하려면:	앱 개발자, 데이터 엔지니어

작업	설명	필요한 기술
	<ol style="list-style-type: none"> 1. CI/CD 계정에 로그인합니다. 2. 이전 에픽에서 생성한 애플리케이션 리포지토리와 CI/CD 파이프라인을 찾습니다. 3. 리포지토리의 URL을 복사하고, <code>git clone</code> 명령어를 사용하여 로컬 시스템에서 리포지토리를 복제합니다. 4. 디렉터리 구조 및 파일이 다음과 일치하는지 확인합니다. <div data-bbox="630 835 1029 1474" style="border: 1px solid #ccc; border-radius: 10px; padding: 10px; margin: 10px 0;"> <pre> root - deploy_configs - deployment_config.json - parameters - template-parameter-dev.json - template-parameter-test.json - template-parameter-prod.json - templates - template.yml - buildspec.yml </pre> </div> <p>여기서 <code>deploy_configs</code> 폴더에는 배포 구성 파일이 들어 있고, <code>templates</code> 및 <code>parameters</code> 폴더에는 자체 CloudFormation 템플릿 및 파라미터 파일로 대체할 기본 파일이 들어 있습니다.</p>	

작업	설명	필요한 기술
	<p>중요: 폴더 구조를 사용자 지정하지 마십시오.</p> <p>5. 기능 브랜치를 생성합니다.</p>	

작업	설명	필요한 기술
<p>애플리케이션 아티팩트를 추가합니다.</p>	<p>CloudFormation 템플릿을 사용하여 응용 프로그램 리포지토리를 업데이트하십시오.</p> <p>참고: 이 솔루션은 단일 CloudFormation 템플릿의 배포만 지원합니다.</p> <ol style="list-style-type: none"> 1. 애플리케이션 코드 변경 내용을 배포하기 위한 CloudFormation 템플릿을 빌드하고 이름을 지정하십시오 <application-name>.yaml . 2. 애플리케이션 리포지토리의 templates 폴더에 있는 template.yml 파일을 1 단계에서 만든 CloudFormation 템플릿으로 바꿉니다. 3. 각 환경(개발, 테스트, 프로덕션)에 대한 파라미터 파일을 준비합니다. 4. <cloudformation-template-name>-parameter-<environment-name>.json 형식을 사용하여 파라미터 파일의 이름을 지정합니다. 5. parameters 폴더의 기본 파라미터 파일을 4단계의 파일로 교체합니다. 	<p>앱 개발자, 데이터 엔지니어</p>

작업	설명	필요한 기술
<p>배포 구성 파일을 업데이트합니다.</p>	<p>deployment_config.json 파일을 다음과 같이 업데이트 합니다.</p> <ol style="list-style-type: none"> 1. 애플리케이션 리포지토리에서 deploy_configs 폴더로 이동합니다. 2. 다음과 같이 deployment_config.json 파일을 엽니다. <pre data-bbox="630 722 1029 1810"> { "deployment_action": "<deploy/delete>", "stack_set_name": "<stack set name>", "stack_set_description": "<stack set description>", "deployment_targets": { "dev": { "org_units": ["list of OUs"], "regions": ["list of regions"], "filter_accounts": ["list of accounts"], } } } </pre>	<p>앱 개발자, 데이터 엔지니어</p>

작업	설명	필요한 기술
	<pre> "filter_type": "<DIFFERENCE/INTER SECTION/UNION>" }, "test": { "org_units": ["list of OUs"], "regions": ["list of regions"], "filter_accounts": ["list of accounts"], "filter_type": "<DIFFERENCE/INTER SECTION/UNION>" }, "prod": { "org_units": ["list of OUs"], "regions": ["list of regions"], </pre>	

작업	설명	필요한 기술
	<pre> "filter_accounts": ["list of accounts"], "filter_type": "<DIFFERENCE/INTERSECTION/UNION>" } }, "cft_capabilities": ["CAPABILITY_IAM", "CAPABILITY_NAMED_IAM"], "auto_deployment": "<True/False>", "retain_stacks_on_account_removal": "<True/False>", "region_deployment_concurrency": "<SEQUENTIAL/PARALLEL>" } </pre> <p>3. 배포 작업, 스택 세트 이름, 스택 세트 설명, 배포 대상의 값을 업데이트합니다.</p> <p>예를 들어 deployment_action (을)를 delete(으)로 설정하여 전체 스택 세트 및 관련 스택 인스턴스를 삭제합니다. deploy(을)를 사용하여 새</p>	

작업	설명	필요한 기술
	<p>스택 세트를 생성하거나, 기존 스택 세트를 업데이트하거나, 추가 OU 또는 리전에 대한 스택 인스턴스를 추가 또는 제거합니다. 자세한 예제는 추가 정보를 참조하세요.</p> <p>이 패턴은 배포 구성 파일에 제공하는 스택 세트 이름에 환경 이름을 추가하여 각 환경에 대한 개별 스택 세트를 생성합니다.</p>	

작업	설명	필요한 기술
<p>변경 사항을 커밋하고 스택 세트를 배포합니다.</p>	<p>애플리케이션 템플릿에서 지정된 변경 사항을 커밋하고, 다음과 같은 방식으로 스택 세트를 여러 환경 단계에 병합 및 배포합니다.</p> <ol style="list-style-type: none"> 1. 모든 파일을 저장하고 로컬 애플리케이션 리포지토리의 기능 브랜치에 변경 사항을 커밋합니다. 2. 기능 브랜치를 원격 리포지토리로 푸시합니다. 3. 풀 요청을 생성하여 변경 사항을 메인 브랜치에 병합합니다. <p>풀 요청이 승인되고 변경 사항이 메인 브랜치에 병합되면 CI/CD 파이프라인이 시작됩니다.</p> <ol style="list-style-type: none"> 4. 개발 배포 단계가 성공적으로 완료되면 CloudFormation 콘솔 StackSets, 서비스 관리 탭을 확인하십시오. <p>dev 접미사가 붙은 새 스택 세트를 볼 수 있습니다.</p> <ol style="list-style-type: none"> 5. 개발 배포 단계의 CodeBuild 로그에서 문제가 있는지 확인하세요. 6. 승인자에게 해당 단계의 배포를 승인하도록 요청하고 5 단계와 6 단계를 반복하여 스택 세트를 테스트 및 프로덕 	<p>앱 개발자, 데이터 엔지니어</p>

작업	설명	필요한 기술
	선 환경에 배포합니다. 테스트 및 프로덕션 환경의 스택 세트에는 test 및 prod 접미사가 있습니다.	

문제 해결

문제	Solution
다음과 같은 예외의 경우에 배포가 실패합니다. 템플릿 파라미터 파일의 이름을 <application name>-parameter-<envn>.json으로 변경합니다. 기본 이름은 허용되지 않습니다.	CloudFormation 템플릿 매개 변수 파일은 지정된 이름 지정 규칙을 따라야 합니다. 파라미터 파일 이름을 업데이트하고 다시 시도합니다.
다음과 같은 예외의 경우에 배포가 실패합니다. 템플릿 이름을 .yaml로 변경하십시오. 기본 CloudFormation 템플릿 .yaml 또는 template.yaml은 허용되지 않습니다. <application name>	템플릿 이름은 지정된 이름 지정 규칙을 따라야 합니다. CloudFormation 파일 이름을 업데이트하고 다시 시도합니다.
다음과 같은 예외의 경우에 배포가 실패합니다. {environment name} 환경에 적합한 CloudFormation 템플릿과 해당 매개 변수 파일을 찾을 수 없습니다.	CloudFormation 템플릿의 파일 이름 지정 규칙과 지정된 환경에 대한 매개 변수 파일을 확인하십시오.
다음과 같은 예외의 경우에 배포가 실패합니다. 배포 구성 파일에 잘못된 배포 작업이 제공되었습니다. 유효한 옵션은 '배포' 및 '삭제'입니다.	배포 구성 파일에서 deployment_action 파라미터에 대한 잘못된 값이 지정되었습니다. 파라미터에는 2개의 유효한 값(deploy 및 delete)이 있습니다. deploy(을)를 사용하여 스택 세트 및 관련 스택 인스턴스를 생성하고 업데이트합니다. 전체 스택 세트 및 관련 스택 인스턴스를 제거하려는 경우에만 delete(을)를 사용합니다.

관련 리소스

- GitHub [automated-code-pipeline-stackset-배포 리포지토리](#)
- [조직 내 모든 기능의 활성화](#) (Organizations 설명서)
- [위임 관리자 등록](#) (AWS CloudFormation 설명서)
- [서비스 관리형 스택 세트의 계정 수준 목표](#) (AWS CloudFormation 설명서)

추가 정보

순서도

다음 순서도는 스택 세트 배포를 자동화하기 위해 사용자 지정 스크립트로 구현되는 API 호출의 흐름 제어 및 계층 구조를 보여줍니다.

샘플 배포 구성 파일

새 스택 세트 생성

다음 배포 구성 파일은 sample-stack-set 리전에서 us-east-1 호출되는 새 스택 세트를 3개의 OU에 생성합니다.

```
{
  "deployment_action": "deploy",
  "stack_set_name": "sample-stack-set",
  "stack_set_description": "this is a sample stack set",
  "deployment_targets": {
    "dev": {
      "org_units": ["dev-org-unit-1"],
      "regions": ["us-east-1"],
      "filter_accounts": [],
      "filter_type": ""
    },
    "test": {
      "org_units": ["test-org-unit-1"],
      "regions": ["us-east-1"],
      "filter_accounts": [],
      "filter_type": ""
    },
    "prod": {
```

```

        "org_units": ["prod-org-unit-1"],
        "regions": ["us-east-1"],
        "filter_accounts": [],
        "filter_type": ""
    }
},
"cft_capabilities": ["CAPABILITY_IAM", "CAPABILITY_NAMED_IAM"],
"auto_deployment": "True",
"retain_stacks_on_account_removal": "True",
"region_deployment_concurrency": "PARALLEL"
}

```

기존 스택 세트를 다른 OU에 배포

이전 예제에 표시된 구성을 배포하고 개발 환경에서 dev-org-unit-2 호출되는 추가 OU에 스택 세트를 배포하려는 경우 배포 구성 파일은 다음과 같을 수 있습니다.

```

{
  "deployment_action": "deploy",
  "stack_set_name": "sample-stack-set",
  "stack_set_description": "this is a sample stack set",
  "deployment_targets": {
    "dev": {
      "org_units": ["dev-org-unit-1", "dev-org-
unit-2"],
      "regions": ["us-east-1"],
      "filter_accounts": [],
      "filter_type": ""
    },
    "test": {
      "org_units": ["test-org-unit-1"],
      "regions": ["us-east-1"],
      "filter_accounts": [],
      "filter_type": ""
    },
    "prod": {
      "org_units": ["prod-org-unit-1"],
      "regions": ["us-east-1"],
      "filter_accounts": [],
      "filter_type": ""
    }
  },
  "cft_capabilities": ["CAPABILITY_IAM", "CAPABILITY_NAMED_IAM"],
}

```

```

    "auto_deployment": "True",
    "retain_stacks_on_account_removal": "True",
    "region_deployment_concurrency": "PARALLEL"
  }

```

기존 스택 세트를 다른 리전에 배포

이전 예제에 표시된 구성을 배포하고 두 개의 OU(dev-org-unit-1 및 dev-org-unit-2)에 대한 개발 환경에서 호출되는 추가 리전(us-east-2)에 스택 세트를 배포하려는 경우 배포 구성 파일은 다음처럼 보일 수 있습니다.

참고: CloudFormation 템플릿의 리소스는 유효하고 지역별로 달라야 합니다.

```

{
  "deployment_action": "deploy",
  "stack_set_name": "sample-stack-set",
  "stack_set_description": "this is a sample stack set",
  "deployment_targets": {
    "dev": {
      "org_units": ["dev-org-unit-1", "dev-org-unit-2"],
      "regions": ["us-east-1", "us-east-2"],
      "filter_accounts": [],
      "filter_type": ""
    },
    "test": {
      "org_units": ["test-org-unit-1"],
      "regions": ["us-east-1"],
      "filter_accounts": [],
      "filter_type": ""
    },
    "prod": {
      "org_units": ["prod-org-unit-1"],
      "regions": ["us-east-1"],
      "filter_accounts": [],
      "filter_type": ""
    }
  },
  "cft_capabilities": ["CAPABILITY_IAM", "CAPABILITY_NAMED_IAM"],
  "auto_deployment": "True",
  "retain_stacks_on_account_removal": "True",
  "region_deployment_concurrency": "PARALLEL"
}

```

OU 또는 리전에서 스택 인스턴스 제거

이전 예제에 표시된 배포 구성이 배포되었다고 가정해 보겠습니다. 다음 구성 파일은 dev-org-unit-2OU의 두 리전에서 스택 인스턴스를 제거합니다.

```
{
  "deployment_action": "deploy",
  "stack_set_name": "sample-stack-set",
  "stack_set_description": "this is a sample stack set",
  "deployment_targets": {
    "dev": {
      "org_units": ["dev-org-unit-1"],
      "regions": ["us-east-1", "us-east-2"],
      "filter_accounts": [],
      "filter_type": ""
    },
    "test": {
      "org_units": ["test-org-unit-1"],
      "regions": ["us-east-1"],
      "filter_accounts": [],
      "filter_type": ""
    },
    "prod": {
      "org_units": ["prod-org-unit-1"],
      "regions": ["us-east-1"],
      "filter_accounts": [],
      "filter_type": ""
    }
  },
  "cft_capabilities": ["CAPABILITY_IAM", "CAPABILITY_NAMED_IAM"],
  "auto_deployment": "True",
  "retain_stacks_on_account_removal": "True",
  "region_deployment_concurrency": "PARALLEL"
}
```

다음 구성 파일은 개발 환경의 두 OU에 대해 us-east-1 리전에서 스택 인스턴스를 제거합니다.

```
{
  "deployment_action": "deploy",
  "stack_set_name": "sample-stack-set",
  "stack_set_description": "this is a sample stack set",
  "deployment_targets": {
```

```

        "dev": {
            "org_units": ["dev-org-unit-1", "dev-org-
unit-2"],
            "regions": ["us-east-2"],
            "filter_accounts": [],
            "filter_type": ""
        },
        "test": {
            "org_units": ["test-org-unit-1"],
            "regions": ["us-east-1"],
            "filter_accounts": [],
            "filter_type": ""
        },
        "prod": {
            "org_units": ["prod-org-unit-1"],
            "regions": ["us-east-1"],
            "filter_accounts": [],
            "filter_type": ""
        }
    },
    "cft_capabilities": ["CAPABILITY_IAM", "CAPABILITY_NAMED_IAM"],
    "auto_deployment": "True",
    "retain_stacks_on_account_removal": "True",
    "region_deployment_concurrency": "PARALLEL"
}

```

전체 스택 세트 삭제

다음 배포 구성 파일은 전체 스택 세트 및 모든 관련 스택 인스턴스를 삭제합니다.

```

{
    "deployment_action": "delete",
    "stack_set_name": "sample-stack-set",
    "stack_set_description": "this is a sample stack set",
    "deployment_targets": {
        "dev": {
            "org_units": ["dev-org-unit-1", "dev-org-
unit-2"],
            "regions": ["us-east-2"],
            "filter_accounts": [],
            "filter_type": ""
        },
        "test": {
            "org_units": ["test-org-unit-1"],

```

```

        "regions": ["us-east-1"],
        "filter_accounts": [],
        "filter_type": ""
      },
      "prod": {
        "org_units": ["prod-org-unit-1"],
        "regions": ["us-east-1"],
        "filter_accounts": [],
        "filter_type": ""
      }
    },
    "cft_capabilities": ["CAPABILITY_IAM", "CAPABILITY_NAMED_IAM"],
    "auto_deployment": "True",
    "retain_stacks_on_account_removal": "True",
    "region_deployment_concurrency": "PARALLEL"
  }
}

```

배포에서 계정 제외

다음 배포 구성 파일은 dev-org-unit-1OU의 일부인 111122223333계정을 배포에서 제외합니다.

```

{
  "deployment_action": "deploy",
  "stack_set_name": "sample-stack-set",
  "stack_set_description": "this is a sample stack set",
  "deployment_targets": {
    "dev": {
      "org_units": ["dev-org-unit-1"],
      "regions": ["us-east-1"],
      "filter_accounts": ["111122223333"],
      "filter_type": "DIFFERENCE"
    },
    "test": {
      "org_units": ["test-org-unit-1"],
      "regions": ["us-east-1"],
      "filter_accounts": [],
      "filter_type": ""
    },
    "prod": {
      "org_units": ["prod-org-unit-1"],
      "regions": ["us-east-1"],
      "filter_accounts": [],
      "filter_type": ""
    }
  }
}

```

```

    },
    "cft_capabilities": ["CAPABILITY_IAM", "CAPABILITY_NAMED_IAM"],
    "auto_deployment": "True",
    "retain_stacks_on_account_removal": "True",
    "region_deployment_concurrency": "PARALLEL"
}

```

OU의 계정 하위 집합에 애플리케이션 배포

다음 배포 구성 파일은 dev-org-unit-1OU의 3개 계정(111122223333, 444455556666, 777788889999)에만 애플리케이션을 배포합니다.

```

{
  "deployment_action": "deploy",
  "stack_set_name": "sample-stack-set",
  "stack_set_description": "this is a sample stack set",
  "deployment_targets": {
    "dev": {
      "org_units": ["dev-org-unit-1"],
      "regions": ["us-east-1"],
      "filter_accounts": ["111122223333",
"444455556666", "777788889999"],
      "filter_type": "INTERSECTION"
    },
    "test": {
      "org_units": ["test-org-unit-1"],
      "regions": ["us-east-1"],
      "filter_accounts": [],
      "filter_type": ""
    },
    "prod": {
      "org_units": ["prod-org-unit-1"],
      "regions": ["us-east-1"],
      "filter_accounts": [],
      "filter_type": ""
    }
  },
  "cft_capabilities": ["CAPABILITY_IAM", "CAPABILITY_NAMED_IAM"],
  "auto_deployment": "True",
  "retain_stacks_on_account_removal": "True",
  "region_deployment_concurrency": "PARALLEL"
}

```

Cloud Custodian 및 AWS CDK를 사용하여 Systems Manager용 AWS 관리형 정책을 EC2 인스턴스 프로파일에 자동으로 연결

작성자: Ali Asfour(AWS) 및 Aaron Lennon(AWS)

환경: PoC 또는 파일럿

기술: DevOps; 소프트웨어 개발 및 테스트, 관리 및 거버넌스, 보안, ID, 규정 준수, 인프라

워크로드: 오픈 소스

AWS 서비스: 아마존 SNS, 아마존 SQS, AWS, AWS, CodeBuild AWS Systems Manager CodePipeline, AWS CodeCommit

요약

Amazon Elastic Compute Cloud(Amazon EC2) 인스턴스를 AWS Systems Manager와 통합하여 운영 작업을 자동화하고 가시성과 제어를 강화할 수 있습니다. Systems Manager와 통합하려면 EC2 인스턴스에 [AWS Systems Manager Agent\(SSM Agent\)](#)가 설치되어 있고 AmazonSSMManagedInstanceCore AWS Identity and Access Management(IAM) 정책이 인스턴스 프로파일에 연결되어 있어야 합니다.

하지만 모든 EC2 인스턴스 프로파일에 AmazonSSMManagedInstanceCore 정책이 연결되도록 하려는 경우, 인스턴스 프로파일이 없는 새 EC2 인스턴스나 인스턴스 프로파일은 있지만 AmazonSSMManagedInstanceCore 정책이 없는 EC2 인스턴스를 업데이트하는 데 어려움을 겪을 수 있습니다. 또한 여러 Amazon Web Services(AWS) 계정과 AWS 리전에 이 정책을 추가하는 것도 어려울 수 있습니다.

이 패턴은 AWS 계정에 세 가지 [클라우드 관리](#) 정책을 배포하여 이러한 문제를 해결하는 데 도움이 됩니다.

- 첫 번째 Cloud Custodian 정책은 인스턴스 프로파일은 있지만 AmazonSSMManagedInstanceCore 정책이 없는 기존 EC2 인스턴스를 확인합니다. 그러면 AmazonSSMManagedInstanceCore 정책이 첨부됩니다.

- 두 번째 Cloud Custodian 정책은 인스턴스 프로파일이 없는 기존 EC2 인스턴스를 확인하고 AmazonSSMManagedInstanceCore 정책이 연결된 기본 인스턴스 프로파일을 추가합니다.
- 세 번째 클라우드 관리 정책은 계정에 [AWS Lambda 함수](#)를 생성하여 EC2 인스턴스 및 인스턴스 프로파일 생성을 모니터링합니다. 이렇게 하면 EC2 인스턴스가 생성될 때 AmazonSSMManagedInstanceCore 정책이 자동으로 연결됩니다.

이 패턴은 [AWS DevOps](#) 도구를 사용하여 별도의 컴퓨팅 환경을 프로비저닝하지 않고도 다중 계정 환경에 Cloud Custodian 정책을 대규모로 지속적으로 배포합니다.

사전 조건 및 제한 사항

사전 조건

- 두 개 이상의 활성 AWS 계정. 한 계정은 보안 계정이고 다른 계정은 멤버 계정입니다.
- 보안 계정에서 AWS 리소스를 프로비저닝할 수 있는 권한. 이 패턴은 [관리자 권한](#)을 사용하지만 조직의 요구 사항 및 정책에 따라 권한을 부여해야 합니다.
- 보안 계정에서 구성원 계정으로 IAM 역할을 수임하고 필요한 IAM 역할을 생성하는 기능. 자세한 내용은 IAM 설명서의 [IAM 역할을 사용한 AWS 계정 간 액세스 권한 위임](#)을 참조하십시오.
- AWS Command Line Interface(AWS CLI), 설치 및 구성됨. 테스트 목적으로 `aws configure` 명령을 사용하거나 환경 변수를 설정하여 AWS CLI를 구성할 수 있습니다. 중요: 프로덕션 환경에서는 권장되지 않으므로 이 계정에는 최소 권한 액세스 권한만 부여하는 것이 좋습니다. 자세한 내용은 IAM 설명서의 [최소 권한 부여](#)를 참조하십시오.
- `devops-cdk-cloudcustodian.zip` 파일(첨부)이 로컬 컴퓨터에 다운로드됩니다.
- Python에 대해 숙지.
- 필수 도구(Node.js, AWS Cloud Development Kit(AWS CDK) 및 Git)가 설치 및 구성됩니다. `devops-cdk-cloudcustodian.zip` 파일의 `install-prerequisites.sh` 파일을 사용하여 이러한 도구를 설치할 수 있습니다. 이 파일을 루트 권한으로 실행해야 합니다.

제한 사항

- 프로덕션 환경에서도 이 패턴을 사용할 수 있지만 모든 IAM 역할 및 정책이 조직의 요구 사항 및 정책을 충족하는지 확인하십시오.

패키지 버전

- Cloud Custodian 버전 0.9 이상

- TypeScript 버전 3.9.7 이상
- Node.js 버전 14.15.4 이상
- npm 버전 7.6.1 이상
- AWS CDK 버전 1.96.0 이상

아키텍처

이 다이어그램은 다음 워크플로를 보여줍니다.

1. 클라우드 관리자 정책은 보안 계정의 AWS CodeCommit 리포지토리로 푸시됩니다. Amazon CloudWatch Events 규칙은 AWS CodePipeline 파이프라인을 자동으로 시작합니다.
2. 파이프라인은 AWS에서 처리하는 지속적 통합 CodeCommit 및 지속적 전달 (CI/CD) 파이프라인의 지속적 통합 부분에서 가장 최신 코드를 가져와서 전송합니다. CodeBuild
3. CodeBuild Cloud Custodian 정책에 대한 정책 구문 검증은 비롯한 전체 DevSecOps 작업을 수행하고 이러한 정책을 `--dryrun` 모드에서 실행하여 식별된 리소스를 확인합니다.
4. 오류가 없는 경우 다음 작업에서는 관리자에게 변경 사항을 검토하고 구성원 계정에 배포를 승인하도록 알립니다.

기술 스택

- AWS CDK
- CodeBuild
- CodeCommit
- CodePipeline
- IAM
- Cloud Custodian

자동화 및 규모 조정

AWS CDK 파이프라인 모듈은 AWS 스택으로 AWS 리소스를 배포하는 것 외에도 소스 코드의 구축 및 테스트를 CodePipeline 오케스트레이션하는 데 사용하는 CI/CD 파이프라인을 제공합니다. CodeBuild CloudFormation 이 패턴은 조직의 모든 멤버 계정 및 리전에 사용할 수 있습니다. Roles creation 스택을 확장하여 멤버 계정에 다른 IAM 역할을 배포할 수도 있습니다.

도구

- [AWS Cloud Development Kit \(AWS CDK\)](#) 는 코드로 클라우드 인프라를 정의하고 AWS를 통해 프로비저닝하기 위한 소프트웨어 개발 프레임워크입니다. CloudFormation
- [AWS Command Line Interface\(AWS CLI\)](#)는 명령줄 셸에서 명령을 사용하여 AWS 서비스와 상호 작용할 수 있는 오픈 소스 도구입니다.
- [CodeBuildAWS](#)는 클라우드의 완전 관리형 빌드 서비스입니다.
- [CodeCommitAWS](#)는 자산을 비공개로 저장하고 관리하는 데 사용할 수 있는 버전 제어 서비스입니다.
- [CodePipelineAWS](#)는 소프트웨어를 출시하는 데 필요한 단계를 모델링, 시각화 및 자동화하는 데 사용할 수 있는 지속적 전송 서비스입니다.
- [AWS Identity and Access Management](#)는 AWS 리소스에 대한 사용자의 액세스를 안전하게 제어할 수 있게 지원하는 웹 서비스입니다.
- [Cloud Custodian](#)은 많은 조직이 퍼블릭 클라우드 계정을 관리하는 데 사용하는 도구와 스크립트를 하나의 오픈소스 도구로 통합합니다.
- [Node.js](#) 는 구글 크롬의 V8 JavaScript 엔진을 기반으로 구축된 JavaScript 런타임입니다.

code

이 패턴에 사용되는 모듈, 계정 함수, 파일, 배포 명령의 자세한 목록은 devops-cdk-cloudcustodian.zip 파일(첨부)의 README 파일을 참조하십시오.

에픽

AWS CDK로 파이프라인 설정

작업	설명	필요한 기술
CodeCommit 리포지토리를 설정합니다.	<ol style="list-style-type: none"> 1. 로컬 컴퓨터의 작업 디렉터리에 devops-cdk-cloudcustodian.zip 파일(첨부)을 압축 해제합니다. 2. 보안 계정의 AWS Management 콘솔에 로그인 	개발자

작업	설명	필요한 기술
	<p>인하고 CodeCommit 콘솔을 연 다음 새 devops-cdk-cloudcustodian 리포지토리를 생성합니다.</p> <p>3. 프로젝트 디렉터리로 변경하고 CodeCommit 리포지토리를 오리진으로 설정하고 변경 내용을 커밋한 다음 다음 명령을 실행하여 오리진 브랜치에 푸시합니다.</p> <ul style="list-style-type: none"> • <code>cd devops-cdk-cloudcustodian</code> • <code>git init --initial-branch=main</code> • <code>git add . git commit -m 'initial commit'</code> • <code>git remote add origin https://git-codecommit.us-east-1.amazonaws.com/v1/devops-cdk-cloudcustodian</code> • <code>git push origin main</code> <p>이에 대한 자세한 내용은 AWS CodeCommit 설명서의 CodeCommit 리포지토리 생성을 참조하십시오.</p>	

작업	설명	필요한 기술
필수 도구를 설치합니다.	<p>이 <code>install-prerequisites.sh</code> 파일을 사용하여 Amazon Linux에 필요한 모든 도구를 설치할 수 있습니다. AWS CLI는 사전 설치되어 제공되므로 여기에 포함되지 않습니다.</p> <p>이에 대한 자세한 내용은 AWS CDK 설명서의 AWS CDK 시작의 사전 조건 섹션을 참조하십시오.</p>	개발자

작업	설명	필요한 기술
필수 AWS CDK 패키지를 설치합니다.	<ol style="list-style-type: none"> 1. AWS CLI에서 다음 \$ python3 -m venv .env 명령을 실행하여 가상 환경을 설정합니다. 2. AWS CLI에서 다음 \$ source .env/bin/ activate 명령을 실행하여 가상 환경을 활성화합니다. 3. 가상 환경을 활성화한 후 다음 \$ pip install -r requirements.txt 명령을 실행하여 필요한 종속성을 설치합니다. 4. 종속 항목(예를 들어, 다른 AWS CDK 라이브러리)을 추가하려면 requirements.txt 파일에 종속성을 추가한 후 다음 pip install -r requirements.txt 명령을 실행합니다. <p>다음 패키지는 AWS CDK에 필요하며 requirements.txt 파일에 포함되어 있습니다.</p> <ul style="list-style-type: none"> • aws-cdk.aws-cloudwatch • aws-cdk.aws-codebuild • aws-cdk.aws-codecommit 	개발자

작업	설명	필요한 기술
	<ul style="list-style-type: none"> • <code>aws-cdk.aws-codedeploy</code> • <code>aws-cdk.aws-codepipeline</code> • <code>aws-cdk.aws-codepipeline-actions</code> • <code>aws-cdk.aws-events</code> • <code>aws-cdk.aws-eventstargets</code> • <code>aws-cdk.aws-iam</code> • <code>aws-cdk.aws-logs</code> • <code>aws-cdk.aws-s3</code> • <code>aws-cdk.aws-sns</code> • <code>aws-cdk.aws-sns-subscriptions</code> • <code>aws-cdk.aws-sqs</code> • <code>aws-cdk.core</code> 	

환경 구성

작업	설명	필요한 기술
필수 변수를 업데이트하십시오.	<p>CodeCommit 리포지토리의 루트 폴더에서 <code>vars.py</code> 파일을 열고 다음 변수를 업데이트하십시오.</p> <ul style="list-style-type: none"> • 파이프라인을 배포하려는 AWS 리전으로 <code>var_deploy_region = 'us-east-1'</code> 을 업데이트합니다. 	개발자

작업	설명	필요한 기술
	<ul style="list-style-type: none"> • CodeCommit 리포지토리 var_codecommit_rep o_name = "cdk-clou dcustodian" 이름으로 업데이트하십시오. • CodeCommit 브랜치 var_codecommit_bra nch_name = "main" 이 름으로 업데이트하세요. • 변경을 승인하는 관리자의 이메일 주소로 var_admin Email=notifyadmin@ email.com' 을 업데이트 합니다. • 변경 시 Cloud Custodian 알림을 보내는 데 사용 되는 Slack 웹후크로 var_slackWebHookUrl = https://hooks.slac k.com/services/T00 000000/B00000000/X XXXXXXXXXXXXXXXXXXXX XXXXX 를 업데이트합니다. • 조직 ID로 var_orgId = 'o-yyyyyyyyyyy' 를 업데 이트합니다. • 파이프라인이 배포된 계정의 AWS 계정 ID로 security_ account = '12345678 9011' 을 업데이트합니다. • AWS CDK 스택을 부트스 트랩하려는 멤버 계정으로 member_accounts = 	

작업	설명	필요한 기술
	<p>['111111111111', '111111111112', '111111111113'] 을 업데이트 하고 필요한 IAM 역할을 배포하십시오.</p> <ul style="list-style-type: none"> • 파이프라인이 AWS CDK를 멤버 계정에 자동으로 부트스트랩하도록 하려면 <code>cdk_bootstrap_member_accounts = True</code>를 <code>True</code>로 설정하십시오. <code>True</code>로 설정하면 보안 계정에서 위임할 수 있는 멤버 계정의 기존 IAM 역할 이름도 필요합니다. 또한 이 IAM 역할에는 AWS CDK를 부트스트랩하는 데 필요한 권한이 있어야 합니다. • 보안 계정에서 위임할 수 있는 멤버 계정의 기존 IAM 역할로 <code>cdk_bootstrap_role = 'AWSControlTowerExecution'</code> 을 업데이트합니다. 또한 이 역할에는 AWS CDK를 부트스트랩하는 권한이 있어야 합니다. 참고: 이는 <code>cdk_bootstrap_member_accounts</code> 가 <code>True</code>로 설정된 경우에만 적용됩니다. 	

작업	설명	필요한 기술
account.yml 파일을 멤버 계정 정보로 업데이트하십시오.	<p>여러 계정에 대해 c7n-org Cloud Custodian 도구를 실행하려면 accounts.yml 구성 파일을 리포지토리의 루트에 배치해야 합니다. 다음은 AWS 용 Cloud Custodian 구성 파일 예제입니다.</p> <pre>accounts: - account_id: '123123123123' name: account-1 regions: - us-east-1 - us-west-2 role: arn:aws:iam::123123123123:role/CloudCustodian vars: charge_code: xyz tags: - type:prod - division:some division - partition:us - scope:pci</pre>	개발자

AWS 계정 부트스트랩

작업	설명	필요한 기술
보안 계정을 부트스트랩합니다.	다음 명령을 실행하여 cloudcustodian_stack 애플리케이션을 통해 deploy_account 를 부트스트랩합니다.	개발자

작업	설명	필요한 기술
	<pre>cdk bootstrap -a 'python3 cloudcustodian/cl oudcustodian_stack.py</pre>	
<p>옵션 1 - 멤버 계정을 자동으로 부트스트랩합니다.</p>	<p>vars.py파일에서 cdk_bootstrap_member_accounts 변수를 True로 설정하면 member_accounts 변수에 지정된 계정이 파이프라인에 의해 자동으로 부트스트랩됩니다.</p> <p>필요한 경우 보안 계정에서 위임할 수 있고 AWS CDK를 부트스트랩하는 데 필요한 권한이 있는 IAM 역할로 *cdk_bootstrap_role* 을 업데이트할 수 있습니다.</p> <p>member_accounts 변수에 추가된 새 계정은 파이프라인에 의해 자동으로 부트스트랩되므로 필요한 역할을 배포할 수 있습니다.</p>	개발자

작업	설명	필요한 기술
<p>옵션 2 - 멤버 계정을 수동으로 부트스트랩합니다.</p>	<p>이 방법을 사용하는 것은 권장되지 않지만 다음 명령을 실행하여 <code>cdk_bootstrap_member_accounts</code> 값을 <code>False</code>로 설정하고 이 단계를 수동으로 수행할 수 있습니다.</p> <pre data-bbox="592 583 1027 1738"> \$ cdk bootstrap -a 'python3 cloudcustodian/member_accounts_roles_stack.py' \ --trust {security_account_id} \ --context assume-role-credentials:writeIamRoleName={role_name} \ --context assume-role-credentials:readIamRoleName={role_name} \ --mode=ForWriting \ --context bootstrap=true \ --cloudformation-execution-policies arn:aws:iam::aws:policy/AdministratorAccess </pre> <p>중요: 보안 계정에서 위임할 수 있고 AWS CDK를 부트</p>	<p>개발자</p>

작업	설명	필요한 기술
	<p>스트랩하는 데 필요한 권한을 가진 IAM 역할의 이름으로 {security_account_id} 및 {role_name} 값을 업데이트해야 합니다.</p> <p>다른 접근 방식을 사용하여 회원 계정을 부트스트랩할 수도 있습니다 (예: AWS CloudFormation 사용). 이에 대한 자세한 내용은 AWS CDK 설명서의 부트스트래핑을 참조하십시오.</p>	

AWS CDK 스택 배포

작업	설명	필요한 기술
<p>멤버 계정에서 IAM 역할을 생성합니다.</p>	<p>다음 명령을 실행하여 member_account_roles_stack 스택을 배포하고 멤버 계정에 IAM 역할을 생성합니다.</p> <pre>cdk deploy --all -a 'python3 cloudcustodian/member_account_roles_stack.py' --require-approval never</pre>	<p>개발자</p>
<p>Cloud Custodian 파이프라인 스택을 배포하십시오.</p>	<p>다음 명령을 실행하여 보안 계정에 배포되는 Cloud Custodian cloudcustodian_stack.py 파이프라인을 생성합니다.</p>	<p>개발자</p>

작업	설명	필요한 기술
	<pre>cdk deploy -a 'python3 cloudcustodian/clo udcustodian_stack.py'</pre>	

관련 리소스

- [AWS SDK 시작하기](#)

첨부

이 문서와 관련된 추가 콘텐츠에 액세스하려면 [attachment.zip](#) 파일의 압축을 풉니다.

AWS CDK를 사용하여 마이크로서비스용 CI/CD 파이프라인 및 Amazon ECS 클러스터 자동으로 구축

작성자: Varsha Raju(AWS)

환경: PoC 또는 파일럿

기술: DevOps; 컨테이너 및 마이크로서비스; 현대화; 인프라

AWS 서비스: AWS CodeBuild, AWS CodeCommit, AWS CodePipeline, 아마존 ECS, AWS CDK

요약

이 패턴은 Amazon Elastic Container Service(Amazon ECS)에서 마이크로서비스를 구축하고 배포하기 위한 지속적 통합 및 지속적 전송(CI/CD) 파이프라인과 기본 인프라를 자동으로 생성하는 방법을 설명합니다. proof-of-concept CI/CD 파이프라인을 설정하여 조직에 CI/CD, 마이크로서비스 등의 이점을 보여주려는 경우 이 접근 방식을 사용할 수 있습니다. DevOps 또한 이 접근 방식을 사용하여 초기 CI/CD 파이프라인을 만든 다음 조직의 요구 사항에 따라 사용자 지정하거나 변경할 수 있습니다.

이 패턴의 접근 방식은 각각 Virtual Private Cloud(VPC)와 두 개의 가용 영역에서 실행되도록 구성된 Amazon ECS 클러스터를 포함하는 프로덕션 환경과 비프로덕션 환경을 생성합니다. 이러한 환경은 모든 마이크로서비스에서 공유되며 그런 다음 각 마이크로서비스에 대한 CI/CD 파이프라인을 생성합니다. 이러한 CI/CD 파이프라인은 CodeCommit AWS의 소스 리포지토리에서 변경 내용을 가져와 변경 사항을 자동으로 구축한 다음 프로덕션 및 비프로덕션 환경에 배포합니다. 파이프라인이 모든 단계를 성공적으로 완료하면 URL을 사용하여 프로덕션 및 비프로덕션 환경에서 마이크로서비스에 액세스할 수 있습니다.

사전 조건 및 제한 사항

사전 조건

- 활성 Amazon Web Services(AWS) 계정.
- starter-code.zip 파일을 포함하는 기존 Amazon Simple Storage Service(S3) 버킷(첨부).
- 계정에 설치 및 구성된 AWS Cloud Development Kit(AWS CDK) 이에 대한 자세한 내용은 AWS CDK 설명서의 [AWS CDK 시작](#) 섹션을 참조하십시오.
- Python 3 및 pip, 설치 및 구성됨. 자세한 내용은 [Python 설명서](#)를 참조하십시오.

- AWS CDK, AWS CodeBuild, CodePipeline AWS, CodeCommit 아마존 엘라스틱 컨테이너 레지스트리 (Amazon ECR), 아마존 ECS 및 AWS Fargate에 익숙해야 합니다.
- Docker에 대한 숙지.
- CI/CD에 대한 이해 및 DevOps

제한 사항

- 일반 AWS 계정 한도가 적용됩니다. 자세한 내용은 AWS General Reference 설명서의 [AWS service quotas](#)을 참조하십시오.

제품 버전

- 이 코드는 Node.js 버전 16.13.0 및 AWS CDK 버전 1.132.0을 사용하여 테스트되었습니다.

아키텍처

이 다이어그램은 다음 워크플로를 보여줍니다.

1. 애플리케이션 개발자가 리포지토리에 코드를 커밋합니다. CodeCommit
2. 파이프라인이 시작됩니다.
3. CodeBuild Docker 이미지를 빌드하여 Amazon ECR 리포지토리로 푸시합니다.
4. CodePipeline 비프로덕션 Amazon ECS 클러스터의 기존 Fargate 서비스에 새 이미지를 배포합니다.
5. Amazon ECS는 Amazon ECR 리포지토리의 이미지를 비프로덕션 Fargate 서비스로 가져옵니다.
6. 테스트는 비프로덕션 URL을 사용하여 수행됩니다.
7. 릴리스 관리자가 프로덕션 배포를 승인합니다.
8. CodePipeline 새 이미지를 프로덕션 Amazon ECS 클러스터의 기존 Fargate 서비스에 배포합니다.
9. Amazon ECS는 Amazon ECR 리포지토리의 이미지를 프로덕션 Fargate 서비스로 가져옵니다.
10. 프로덕션 사용자는 프로덕션 URL을 사용하여 기능에 액세스합니다.

기술 스택

- AWS CDK

- CodeBuild
- CodeCommit
- CodePipeline
- Amazon ECR
- Amazon ECS
- Amazon VPC

자동화 및 규모 조정

이 패턴의 접근 방식을 사용하여 공유 AWS CloudFormation 스택에 배포된 마이크로서비스를 위한 파이프라인을 생성할 수 있습니다. 자동화를 통해 각 VPC에 Amazon ECS 클러스터를 한 개 이상 생성할 수 있으며 공유 Amazon ECS 클러스터에 배포된 마이크로서비스를 위한 파이프라인을 생성할 수도 있습니다. 하지만 이를 위해서는 새 리소스 정보를 파이프라인 스택에 대한 입력으로 제공해야 합니다.

도구

- [AWS CDK](#) — AWS Cloud Development Kit (AWS CDK) 는 코드로 클라우드 인프라를 정의하고 AWS를 통해 프로비저닝하기 위한 소프트웨어 개발 프레임워크입니다. CloudFormation
- [AWS CodeBuild](#) — CodeBuild AWS는 클라우드의 완전 관리형 빌드 서비스입니다. CodeBuild 소스 코드를 컴파일하고, 단위 테스트를 실행하고, 배포할 준비가 된 아티팩트를 생성합니다.
- [AWS CodeCommit](#) — CodeCommit AWS는 AWS 클라우드에서 Git 리포지토리를 비공개로 저장하고 관리할 수 있는 버전 제어 서비스입니다. CodeCommit 자체 소스 제어 시스템을 관리하거나 인프라 확장에 대해 걱정할 필요가 없습니다.
- [AWS CodePipeline](#) — CodePipeline AWS는 소프트웨어 릴리스에 필요한 단계를 모델링, 시각화 및 자동화하는 데 사용할 수 있는 지속적 전송 서비스입니다. 소프트웨어 출시 프로세스의 여러 단계를 신속하게 모델링하고 구성할 수 있습니다. CodePipeline 소프트웨어 변경 사항을 지속적으로 릴리스하는 데 필요한 단계를 자동화합니다.
- [Amazon ECS](#) - Amazon Elastic Container Service(Amazon ECS)는 클러스터에서 컨테이너를 실행, 중지 및 관리하기 위해 사용하는 컨테이너 관리 서비스로서 확장성과 속도가 뛰어납니다. AWS Fargate에서 관리하는 서버리스 인프라에서 작업 및 서비스를 실행할 수 있습니다. 또는 인프라에 대한 더 세부적인 제어를 위해, 관리하는 Amazon Elastic Compute Cloud(Amazon EC2) 인스턴스의 클러스터에서 작업과 서비스를 실행할 수 있습니다.
- [Docker](#) - Docker를 사용하면 개발자가 모든 애플리케이션을 가볍고 휴대가 간편하며 자급자족할 수 있는 컨테이너로 포장, 배송 및 실행할 수 있습니다.

코드

이 패턴의 코드는 `cicdstarter.zip` 및 `starter-code.zip` 파일(첨부)에서 확인할 수 있습니다.

에픽

환경을 설정합니다.

작업	설명	필요한 기술
AWS CDK의 작업 디렉터리를 설정합니다.	<ol style="list-style-type: none"> 1. 로컬 머신에서 <code>cicdproject</code> 로 이름을 지정한 디렉터리를 생성합니다. 2. <code>cicdstarter.zip</code> 파일(첨부)을 <code>cicdproject</code> 디렉터리에 다운로드 하고 압축을 풉니다. 그러면 <code>cicdstarter</code> 라는 이름의 폴더가 만들어집니다. 3. <code>cd <user-home>/cicdproject/cicdstarter</code> 명령을 실행합니다. 4. <code>python3 -m venv .venv</code> 명령을 실행하여 Python 가상 환경을 설정합니다. 5. <code>source ./venv/bin/activate</code> 명령을 실행합니다. 6. <code>aws configure</code> 명령을 실행하거나 다음 환경 변수를 사용하여 AWS 환경을 구성합니다. <ul style="list-style-type: none"> • <code>AWS_ACCESS_KEY_ID</code> 	AWS DevOps, 클라우드 인프라

작업	설명	필요한 기술
	<ul style="list-style-type: none"> • AWS_SECRET_ACCESS_KEY • AWS_DEFAULT_REGION 	

공유 인프라 생성

작업	설명	필요한 기술
공유 인프라를 만드십시오.	<ol style="list-style-type: none"> 1. 작업 디렉터리에서 <code>cd cicdvpcecs</code> 명령을 실행합니다. 2. <code>pip3 install -r requirements.txt</code> 명령을 실행하여 필요한 모든 Python 종속성을 설치합니다. 3. <code>cdk bootstrap command</code>를 실행하여 AWS CDK를 위한 AWS 환경을 설정합니다. 4. <code>cdk synth --context aws_account=<aws_account_ID> --context aws_region=<aws-region></code> 명령을 실행합니다. 5. <code>cdk deploy --context aws_account=<aws_account_ID> --context aws_region=<aws-region></code> 명령을 실행합니다. 	AWS DevOps, 클라우드 인프라

작업	설명	필요한 기술
	<p>6. AWS CloudFormation 스택은 다음과 같은 인프라를 생성합니다.</p> <ul style="list-style-type: none"> • <code>cicd-vpc-ecs/cicd-vpc-nonprod</code> 로 이름이 지정된 비프로덕션 VPC • <code>cicd-vpc-ecs/cicd-vpc-prod</code> 로 이름이 지정된 프로덕션 VPC • <code>cicd-ecs-nonprod</code> 로 이름이 지정된 비프로덕션 Amazon ECS 클러스터 • <code>cicd-ecs-prod</code> 로 이름이 지정된 프로덕션 Amazon ECS 클러스터 	
<p>AWS CloudFormation 스택을 모니터링합니다.</p>	<ol style="list-style-type: none"> 1. AWS 관리 콘솔에 로그인하고 AWS CloudFormation 콘솔을 연 다음 목록에서 <code>cicd-vpc-ecs</code> 스택을 선택합니다. 2. 스택 세부 정보 창에서 이벤트 탭을 선택하고 스택 생성 진행 상황을 모니터링합니다. 	<p>AWS DevOps, 클라우드 인프라</p>

작업	설명	필요한 기술
<p>AWS CloudFormation 스택을 테스트합니다.</p>	<ol style="list-style-type: none"> 1. <code>cicd-vpc-ecs</code> AWS CloudFormation 스택이 생성된 후 <code>cicd-vpc-ecs/cicd-vpc-nonprod</code> 및 <code>cicd-vpc-ecs/cicd-vpc-prod</code> VPC가 생성되었는지 확인합니다. 2. <code>cicd-ecs-nonprod</code> 및 <code>cicd-ecs-prod</code> Amazon ECS 클러스터가 생성되었는지 확인하십시오. <p>중요: 두 VPC의 ID와 두 VPC 모두의 기본 보안 그룹에 대한 보안 그룹 ID를 모두 기록해야 합니다.</p>	<p>AWS DevOps, 클라우드 인프라</p>

마이크로서비스를 위한 CI/CD 파이프라인 생성

작업	설명	필요한 기술
<p>마이크로서비스를 위한 인프라를 만드십시오.</p>	<ol style="list-style-type: none"> 1. 마이크로서비스의 이름을 지정합니다. 예를 들어, 이 패턴은 마이크로서비스 이름으로 <code>myservice1</code> 을 사용합니다. 2. 작업 디렉터리에서 <code>cd <working-directory>/cdkpipeline</code> 명령을 실행합니다. 3. <code>pip3 install -r requirements.txt</code> 명령을 실행합니다. 	<p>AWS DevOps, 클라우드 인프라</p>

작업	설명	필요한 기술
	<p>4. 이 패턴의 추가 정보 섹션에서 사용할 수 있는 전체 <code>cdk synth</code> 명령을 실행합니다.</p> <p>5. 이 패턴의 추가 정보 섹션에서 사용할 수 있는 전체 <code>cdk deploy</code> 명령을 실행합니다.</p> <p>참고: 디렉터리의 <code>cdk.json</code> 파일을 사용하여 두 명령 모두에 값을 제공할 수도 있습니다.</p>	
<p>AWS CloudFormation 스택을 모니터링합니다.</p>	<p>AWS CloudFormation 콘솔을 열고 <code>myservice1-cicd-stack</code> 스택의 진행 상황을 모니터링합니다. 결국 상태가 <code>CREATE_COMPLETE</code> 로 변경됩니다.</p>	<p>AWS DevOps, 클라우드 인프라</p>

작업	설명	필요한 기술
<p>AWS CloudFormation 스택을 테스트합니다.</p>	<ol style="list-style-type: none"> 1. AWS CodeCommit 콘솔에서 이름이 지정된 리포지토리가 myservice1 존재하고 시작 코드를 포함하고 있는지 확인합니다. 2. AWS CodeBuild 콘솔에서 이름이 지정된 빌드 프로젝트가 myservice1 존재하는지 확인합니다. 3. Amazon ECR 콘솔에서, myservice1 로 이름이 지정된 Amazon ECR 리포지토리가 존재하는지 확인합니다. 4. Amazon ECS 콘솔에서, myservice1 로 이름이 지정된 Fargate 서비스가 비프로덕션 및 프로덕션 Amazon ECS 클러스터 모두에 존재하는지 확인합니다. 5. Amazon Elastic Compute Cloud(Amazon EC2) 콘솔에서 비프로덕션 및 프로덕션 Application Load Balancer가 생성되었는지 확인합니다. ALB의 DNS 이름을 기록하십시오. 6. AWS CodePipeline 콘솔에서 이름이 지정된 파이프라인이 myservice1 존재하는지 확인합니다. Source, Build, Deploy-NonProd 및 Deploy-Pr 	

작업	설명	필요한 기술
	<p>od 스테이지가 있어야 합니다. 파이프라인에도 in progress 상태가 있어야 합니다.</p> <p>7. 모든 단계가 완료될 때까지 파이프라인을 모니터링하십시오.</p> <p>8. 프로덕션을 위해 수동으로 승인하십시오.</p> <p>9. 브라우저 창에 ALB의 DNS 이름을 입력합니다.</p> <p>10. 애플리케이션은 비프로덕션 및 프로덕션 URL에 Hello World를 표시해야 합니다.</p>	
파이프라인을 사용합니다.	<ol style="list-style-type: none"> 이전에 만든 CodeCommit 리포지토리를 열고 index.js 파일을 엽니다. Hello World를 Hello CI/CD로 바꿉니다. 변경 내용을 저장한 후 기본 브랜치에 커밋합니다. 파이프라인이 시작되고, Build, Deploy-NonProd 및 Deploy-Prod 스테이지를 거쳐 변경이 진행되는지 확인하십시오. 수동으로 프로덕션을 승인하십시오. 이제 프로덕션 URL과 비프로덕션 URL이 모두 Hello CI/CD를 표시해야 합니다. 	AWS DevOps, 클라우드 인프라

작업	설명	필요한 기술
각 마이크로서비스에 대해 이 에픽을 반복하십시오.	이 에픽의 작업을 반복하여 각 마이크로서비스에 대한 CI/CD 파이프라인을 생성하십시오.	AWS DevOps, 클라우드 인프라

관련 리소스

- [AWS CDK를 통해 Python 사용](#)
- [AWS CDK Python 참조](#)
- [AWS CDK를 사용하여 AWS Fargate 서비스 생성](#)

추가 정보

cdk synth 명령

```
cdk synth --context aws_account=<aws_account_number> --context
aws_region=<aws_region> --context vpc_nonprod_id=<id_of_non_production
VPC> --context vpc_prod_id=<id_of_production_VPC> --context
ecssg_nonprod_id=< default_security_group_id_of_non-production_VPC>
--context ecssg_prod_id=<default_security_group_id_of_production_VPC>
--context code_commit_s3_bucket_for_code=<S3 bucket name> --context
code_commit_s3_object_key_for_code=<Object_key_of_starter_code> --context
microservice_name=<name_of_microservice>
```

cdk deploy 명령

```
cdk deploy --context aws_account=<aws_account_number> --context
aws_region=<aws_region> --context vpc_nonprod_id=<id_of_non_production_VPC>
--context vpc_prod_id=<id_of_production_VPC> --context ecssg_nonprod_id=<
default_security_group_id_of_non-production_VPC> --context
ecssg_prod_id=<default_security_group_id_of_production_VPC> --
context code_commit_s3_bucket_for_code=<S3 bucket name> --context
code_commit_s3_object_key_for_code=<Object_key_of_starter_code> --context
microservice_name=<name_of_microservice>
```

첨부

이 문서와 관련된 추가 콘텐츠에 액세스하려면 [attachment.zip](#) 파일의 압축을 풉니다.

DevOps 프랙티스와 AWS Cloud9를 사용하여 마이크로서비스로 느슨하게 연결된 아키텍처를 구축하십시오.

작성자: Alexandre Nardi (AWS)

환경: PoC 또는 파일럿

기술: DevOps; 서버리스; 웹 및 모바일 앱; 데이터베이스

AWS 서비스: AWS Cloud9, AWS, CloudFormation AWS, 아마존 DynamoDB, CodePipeline AWS CodeCommit

요약

이 패턴은 Amazon Web Services (AWS) 에서 DevOps 사례를 테스트하기 시작하는 개발자와 개발 책임자를 위해 서버리스 아키텍처에서 일반적인 웹 애플리케이션을 개발하는 방법을 보여줍니다. 책을 검색하고 구매할 수 있는 스토어프론트 및 백엔드를 생성하는 샘플 애플리케이션을 구축하고 독립적으로 개발할 수 있는 마이크로서비스를 제공합니다. 이 패턴은 AWS Cloud9를 개발 환경으로, Amazon DynamoDB 데이터베이스를 데이터 스토어로 사용하고, 지속적 통합 및 지속적 배포 (CI/CD) 기능을 위한 AWS CodePipeline 및 AWS와 같은 CodeBuild AWS 서비스를 사용합니다.

이 패턴은 다음과 같은 개발 활동을 안내합니다.

- 표준 AWS Cloud9 개발 환경 생성
- AWS CloudFormation 템플릿을 사용하여 도서용 웹 애플리케이션 및 마이크로서비스 생성
- AWS Cloud9를 사용하여 프론트엔드를 수정하고, 변경 사항을 커밋하고, 변경 사항을 테스트
- 마이크로서비스에 대한 CI/CD 파이프라인 생성 및 테스트
- 유닛 테스트 자동화

이 패턴의 코드는 [AWS DevOps 엔드-투-엔드 워크숍](#) 리포지토리의 에서 GitHub 제공됩니다.

사전 조건 및 제한 사항

사전 조건

- 활성 상태의 AWS 계정.
- [AWS DevOps 엔드-투-엔드 워크숍의](#) 파일이 사용자 컴퓨터에 다운로드됨

중요: AWS 계정에서 이 데모 애플리케이션을 구축하면 AWS 리소스가 생성되고 소비됩니다. 애플리케이션을 생성하고 실행하는 데 사용된 AWS 서비스 및 리소스의 비용은 사용자가 부담합니다. 작업을 마친 후에는 계속 요금이 부과되지 않도록 모든 리소스를 삭제해야 합니다. 지침은 에픽 섹션을 참조하세요.

제한 사항

이 안내는 데모 및 개발 목적으로만 제공됩니다. 프로덕션 환경에서 사용하려면 AWS ID 및 액세스 관리(IAM) 설명서의 [보안 모범 사례](#)를 참고하고 IAM 역할, Amazon DynamoDB 및 기타 사용되는 서비스를 필요에 따라 변경하십시오. 이 웹 애플리케이션은 [AWS Bookstore 데모 앱](#)에서 파생되었습니다. 추가 고려 사항은 README 파일의 [알려진 제한 사항](#) 섹션을 참고하십시오.

아키텍처

서점 애플리케이션의 아키텍처는 [AWS Bookstore 데모 앱용 README](#) 파일의 [아키텍처](#) 섹션에 설명되어 있습니다.

배포 관점에서 Bookstore 데모 앱은 단일 CloudFormation 템플릿을 사용하여 모든 서비스와 객체를 하나의 스택에 배포합니다. 이 패턴은 특정 개발자 또는 팀이 특정 제품(도서)에서 작업하고 애플리케이션의 나머지 부분과 독립적으로 업데이트할 수 있는 방법을 보여주기 위해 몇 가지 변경 사항을 적용했습니다. 이러한 이유로 이 패턴의 코드는 Books 마이크로서비스의 AWS Lambda 함수 및 관련 객체를 Books 스택을 생성하는 두 번째 템플릿으로 CloudFormation 분리합니다. 따라서 CI/CD 관행을 사용하여 마이크로서비스가 업데이트되는 것을 확인할 수 있습니다. 다음 다이어그램에서 점선 테두리는 Books 마이크로서비스를 나타냅니다.

도구

도구

- 테스트를 위한 Jest 프레임워크 JavaScript
- Python 3.9

코드

이 패턴의 소스 코드와 템플릿은 [AWS GitHub DevOps End-to-End Workshop](#) 리포지토리에서 제공됩니다. 에픽 섹션의 단계를 따르기 전에 리포지토리의 모든 파일을 컴퓨터로 다운로드하십시오.

참고: 에픽 섹션에서는 프로세스에 대한 일반적인 정보를 제공하기 위해 이 안내를 위한 고급 단계를 제공합니다. 각 단계를 완료하려면 AWS DevOps 엔드-투-엔드 워크숍 리포지토리의 [README 파일에](#) [서](#) 자세한 지침을 참조하십시오.

[AWS DevOps 엔드투엔드 워크숍 리포지토리는 AWS 서점 데모 앱 리포지토리를 확장하고 수정된 버전의 AWS Cloud9 \[부트스트래핑 코드를 사용하여 AWS Cloud9 IDE를 생성합니다.\]\(#\)](#)

모범 사례

서점 애플리케이션을 사용하는 방법은 간단합니다. 권장되는 몇 가지 모범 사례는 다음과 같습니다.

- 애플리케이션을 설치할 때 원하는 프로젝트 이름을 사용하거나 편의를 위해 기본 이름 (demobookstore)을 사용할 수 있습니다.
- 애플리케이션을 가동하고 실행한 후 하루 더 테스트를 계속하려면 Amazon Neptune 데이터베이스를 종료하는 것이 좋습니다. 데이터베이스 인스턴스로 인해 추가 요금이 발생할 수 있기 때문입니다. 하지만 7일 후에는 데이터베이스가 자동으로 시작된다는 점에 유의하십시오.
- 코드 세부 정보는 [AWS Bookstore 데모 앱 리포지토리 설명서](#)를 참고하십시오. 각 마이크로서비스와 테이블에 대해 설명합니다.
- 추가 모범 사례는 시간이 있다면 몇 가지 과제를 참조하십시오. AWS DevOps 엔드-투-엔드 워크숍 리포지토리의 [README 파일](#) 섹션. 이 정보를 검토하여 보안을 위한 추가 기능을 자세히 살펴보고 디커플링 서비스를 연습해 보는 것이 좋습니다.

에픽

소스 코드 다운로드

작업	설명	필요한 기술
에서 소스 코드를 다운로드하십시오. GitHub	이 패턴의 소스 코드와 템플릿은 AWS DevOps End-to-End Workshop 리포지토리 의 에서 GitHub 제공됩니다. 에픽 섹션의 다음 단계를 따르기 전에 리포지토리의 모든 파일을 컴퓨터로 다운로드하세요. 참고: 에픽 섹션에서는 프로세스에 대한 일반적인 정보를 제	앱 개발자

작업	설명	필요한 기술
	<p>공하기 위해 이 안내를 위한 고급 단계를 제공합니다. 각 단계를 완료하려면 AWS DevOps 엔드-투-엔드 워크숍 리포지토리의 README 파일에서 자세한 지침을 참조하십시오.</p> <p>AWS DevOps 엔드투엔드 워크숍 리포지토리는 AWS 서점 데모 앱 리포지토리를 확장하고 수정된 버전의 AWS Cloud9 부트스트래핑 코드를 사용하여 AWS Cloud9 IDE를 생성합니다.</p>	

Bookstore 웹 애플리케이션과 Books 마이크로서비스를 구축합니다.

작업	설명	필요한 기술
Bookstore 앱을 위한 프론트엔드 및 Lambda 함수를 생성합니다.	<ol style="list-style-type: none"> CloudFormation 콘솔에 로그인하고 DemoBookstoreMainTemplate.yaml 템플릿을 배포하여 DemoBookStoreStack 스택을 생성합니다. 그러면 Books 마이크로서비스 외부에 있는 프론트엔드 및 Lambda 함수가 생성됩니다. 스택의 출력 탭에서 WebApplication레이블 아래에 있는 웹 사이트 URL을 기록해 둡니다. 	개발자
Books 마이크로서비스를 생성합니다.	CloudFormation 콘솔에서 DemoBookstoreBooks	개발자

작업	설명	필요한 기술
	ServiceTemplate.yml 템플릿을 배포하여 DemoBooksServiceStack 스택을 생성합니다.	
애플리케이션을 테스트합니다.	DemoBookStoreStack 스택의 웹사이트 URL을 사용하여 Bookstore 애플리케이션에 액세스할 수 있습니다.	개발자

Cloud9 환경을 사용하여 애플리케이션을 유지 관리합니다.

작업	설명	필요한 기술
AWS Cloud9 IDE를 생성합니다.	CloudFormation 콘솔에서 C9EnvironmentTemplate.yml 템플릿을 배포하여 AWS Cloud9 환경을 생성합니다.	개발자, 개발자 책임자
리포지토리를 생성합니다 CodeCommit .	<ol style="list-style-type: none"> AWS CodeCommit 콘솔에 로그인하여 프론트 엔드 애플리케이션용 코드가 들어 있는 demobookstore-WebAssets 리포지토리가 있는지 확인합니다. demobookstore-BooksService 라는 Books 마이크로서비스를 위한 리포지토리를 생성하십시오. git clone 명령을 사용하여 AWS Cloud9(demobookstore-WebAssets 및 	개발자

작업	설명	필요한 기술
	demobookstore-BooksService)에 있는 두 리포지토리를 복제합니다.	
프론트엔드에서 코드를 변경하고 파이프라인을 확인합니다.	<ol style="list-style-type: none"> 1. AWS Cloud9를 사용하여 웹페이지에서 일부 코드를 변경합니다. 그러면 demobookstore-WebAssets 리포지토리가 업데이트됩니다. 2. AWS CodePipeline 콘솔에서 데모북스토어-자산-파이프라인이 실행 중인지 확인합니다. 3. 브라우저에서 새로 고쳐 웹 애플리케이션을 테스트하십시오(Firefox에서 Ctrl+F5). 	개발자

Books 마이크로서비스를 위한 CI/CD 파이프라인을 구현합니다.

작업	설명	필요한 기술
빌드 및 서비스 업데이트를 위한 YAML 파일을 추가합니다.	<ol style="list-style-type: none"> 1. AWS Cloud9에서 buildspec.yml 및 DemoBookstoreBooksServiceUpdateTemplate.yml 파일을 업로드합니다. <ul style="list-style-type: none"> • buildspec.yml 에는 구축 지침이 있으며 자동화된 테스트를 위한 테스트 지침도 포함되어 있습니다. 지금 설명하고 나중에 사용할 예정입니다. 	개발자

작업	설명	필요한 기술
	<ul style="list-style-type: none"> DemoBookstoreBooks ServiceUp dateTemplate.yml 은 파이프라인의 배포 단계에서 사용할 수 있는 DemoBookstoreBooks ServiceTemplate.yml 의 업데이트된 버전입니다. <p>2. 파일을 커밋하고 푸시합니다.</p>	
<p>빌드 파이프라인용 S3 버킷을 생성합니다.</p>	<p>S3 버킷을 만들려면 Amazon S3 설명서에 있는 지침을 따르세요.</p> <ul style="list-style-type: none"> 버킷 이름은 전역적으로 고유해야 합니다(예: demobookstore-books-service-pipeline-bucket-<YYYYMMDDHHMM>). 모든 공개 액세스 차단 확인란의 선택을 취소하고 승인합니다... 확인란을 선택합니다. 	<p>개발자</p>
<p>IAM을 사용하여 배포용 역할을 생성합니다. CloudFormation</p>	<p>demobookstore-CloudFormation-role 역할을 생성하여 AdministratorAccess 정책을 연결합니다. 다음 에픽에서는 이 역할을 최소 권한으로 재구성할 수 있습니다.</p>	<p>개발자</p>

작업	설명	필요한 기술
새 파이프라인을 생성하여 Books 마이크로서비스 구축 및 배포를 자동화합니다.	README 파일에 설명된 대로 커밋, 빌드, 배포 단계를 포함하는 BooksService파이프라인 (예: demobookstore- -Pipeline) 을 생성합니다.	개발자
AWS Cloud9에서 마이크로서비스를 테스트합니다.	ListBooks함수를 변경하고 파이프라인이 제대로 작동하는지 확인하세요.	개발자
ListBooks Lambda 함수의 단위 테스트를 자동화합니다.	AWS Cloud9 IDE에서 빌드가 유닛 테스트를 실행하고 테스트 결과를 확인할 수 있도록 합니다. 지침은 README 파일을 참조하세요.	개발자

(선택 사항) 추가 기능 구현

작업	설명	필요한 기술
솔루션을 안전하게 만듭니다.	최소 권한을 갖도록 demobookstore-CloudFormation-role 을 구성하고 다른 사용자 역할도 확인합니다.	개발자
템플릿의 종속성을 제거합니다. CloudFormation	DemoBookstoreMainTemplate.yml 템플릿과 DemoBookstoreBooksServiceTemplate.yml 1 템플릿 간에 정보를 교환하는 방법은 출력과 가져오기를 기반으로 합니다. 이 두 템플릿 간에 값을 전달하면 종속성이 추가됩니다. 종속성을 없애	개발자

작업	설명	필요한 기술
	려면 AWS Systems Manager Parameter Store 를 사용해 보십시오.	
Cart 마이크로서비스를 생성합니다.	예를 들어, Books 마이크로서비스를 사용하여 DemoBookstoreMainTemplate.yaml 템플릿에서 쇼핑 카트 관련 기능을 제거하고 Cart 마이크로서비스를 생성합니다.	개발자

정리

작업	설명	필요한 기술
S3 버킷을 삭제합니다.	<p>Amazon S3 콘솔에서 샘플 웹 애플리케이션과 연결된 다음 버킷을 삭제합니다.</p> <ul style="list-style-type: none"> • AWS Bookstore 데모 애플리케이션으로 두 개의 버킷이 생성되었습니다. 버킷 이름은 프론트엔드를 생성할 CloudFormation 때 AWS에 제공한 스택 이름으로 시작됩니다 (예:). DemoBookStoreStack • <YYYYMMDDHHMM>빌드 파이프라인용 버킷 1개 (예: demobookstore-books-service-pipeline-bucket -). 	개발자
스택을 삭제합니다.	CloudFormation 콘솔 에서 샘플 웹 애플리케이션과 연결된 스택을 삭제합니다.	개발자

작업	설명	필요한 기술
	<ul style="list-style-type: none"> • DemoBooksServiceStack • DemoBookStoreStack <p>제거하는 데 90분 이상 걸릴 수 있습니다. 제거에 실패하면 다시 삭제하고 알림에 따라 수동 리소스(예: VPC 또는 네트워크 인터페이스)도 삭제합니다.</p>	
IAM 역할을 삭제합니다.	<p>IAM 콘솔에서 다음 역할을 삭제합니다.</p> <ul style="list-style-type: none"> • demobookstore-Cloudformation-role • demobookstore-BooksService-BuildProject-service-role <p>step-by-step 지침은 IAM 설명서를 참조하십시오.</p>	개발자

관련 리소스

- [AWS Bookstore 데모 앱](#)
- [AWS Cloud9 부트스트래핑 예제](#)
- [AWS CloudFormation 콘솔에서 스택 생성](#) (AWS CloudFormation 설명서)
- [버킷 생성](#)(Amazon S3 설명서)

추가 정보

자세한 step-by-step 지침은 [AWS DevOps 엔드-투-엔드 워크숍 리포지토리의 README 파일을](#) 참조하십시오. GitHub

2023년 5월 업데이트 정보: 이 패턴은 최신 버전의 Node 및 Python을 사용하도록 업데이트되었습니다. 소스 코드의 많은 패키지를 업데이트했으며 Glyphicon은 더 이상 무료가 아니기 때문에 삭제했습니다. 또한, 이제 두 리포지토리가 독립적으로 발전할 수 있도록 [AWS Bookstore 데모 앱](#) 리포지토리의 모든 종속성을 제거했습니다.

GitHub 액션과 테라폼을 사용하여 Docker 이미지를 빌드하고 Amazon ECR로 푸시합니다.

작성자: 루치카 모디 (AWS)

코드 리포지토리: docker-ecr-actions-workflow	환경: 프로덕션	기술: DevOps; 컨테이너 및 마이크로서비스; 인프라
워크로드: 기타 모든 워크로드	AWS 서비스: 아마존 ECR	

요약

이 패턴은 재사용 가능한 GitHub 워크플로를 생성하여 Dockerfile을 구축하고 결과 이미지를 Amazon Elastic Container Registry (Amazon ECR) 로 푸시하는 방법을 설명합니다. 이 패턴은 Terraform과 Actions를 사용하여 Dockerfile의 빌드 프로세스를 자동화합니다. GitHub 이를 통해 인적 오류 가능성이 최소화되고 배포 시간이 크게 단축됩니다.

GitHub 리포지토리의 기본 브랜치에 대한 GitHub 푸시 작업을 수행하면 리소스 배포가 시작됩니다. 워크플로는 GitHub 조직과 리포지토리 이름의 조합을 기반으로 고유한 Amazon ECR 리포지토리를 생성합니다. 그런 다음 Dockerfile 이미지를 Amazon ECR 리포지토리로 푸시합니다.

사전 조건 및 제한 사항

사전 조건

- 활성 상태의 AWS 계정.
- 활성 계정. GitHub
- [GitHub 리포지토리](#).
- Terraform 버전 1 이상이 [설치 및](#) 구성되었습니다.
- 테라폼 백엔드용 아마존 심플 스토리지 서비스 (Amazon S3) 버킷입니다.
- 테라폼 [상태 잠금 및 일관성을 위한 Amazon](#) DynamoDB 테이블입니다. 테이블에는 유형이 다음과 같은 파티션 키가 있어야 합니다. LockID String 이를 구성하지 않으면 상태 잠금이 비활성화됩니다.
- Terraform용 Amazon S3 백엔드를 설정할 권한이 있는 AWS ID 및 액세스 관리 (IAM) 역할입니다. [구성 지침은 Terraform 설명서를 참조하십시오.](#)

제한 사항

이 재사용 가능한 코드는 Actions에서만 테스트되었습니다. GitHub

아키텍처

대상 기술 스택

- 아마존 ECR 리포지토리
- GitHub 액션
- Terraform

대상 아키텍처

다이어그램은 다음을 보여 줍니다.

1. 사용자가 Dockerfile 및 Terraform 템플릿을 리포지토리에 추가합니다. GitHub
2. 이러한 추가 기능을 사용하면 액션 워크플로가 시작됩니다. GitHub
3. 워크플로는 Amazon ECR 리포지토리가 존재하는지 여부를 확인합니다. 그렇지 않은 경우 GitHub 조직 및 리포지토리 이름을 기반으로 리포지토리를 생성합니다.
4. 워크플로는 Dockerfile을 빌드하고 이미지를 Amazon ECR 리포지토리로 푸시합니다.

도구

Amazon 서비스

- [Amazon Elastic 컨테이너 레지스트리 \(Amazon ECR\)](#) 는 안전하고 확장 가능하며 신뢰할 수 있는 관리형 컨테이너 레지스트리 서비스입니다.

기타 도구

- [GitHub Actions](#) 는 GitHub 플랫폼에 통합되어 GitHub 리포지토리 내에서 워크플로를 생성, 공유 및 실행하는 데 도움이 됩니다. GitHub Actions를 사용하여 코드 작성, 테스트, 배포와 같은 작업을 자동화할 수 있습니다.
- [Terraform](#) 은 클라우드 및 온프레미스 인프라를 만들고 관리하는 데 도움이 HashiCorp 되는 코드형 오픈 소스 인프라 (IaC) 도구입니다.

코드 리포지토리

이 패턴의 코드는 GitHub [Docker ECR](#) 작업 워크플로 리포지토리에서 사용할 수 있습니다.

- GitHub 액션을 생성하면 Docker 워크플로 파일이 이 리포지토리의 `/.github/workflows/` 폴더에 저장됩니다. 이 솔루션의 워크플로는 [workflow.yaml](#) 파일에 있습니다.
- 이 `e2e-test` 폴더는 참조 및 테스트를 위한 샘플 Dockerfile을 제공합니다.

모범 사례

- [Dockerfile 작성에 대한 모범 사례는 Docker 설명서를 참조하십시오.](#)
- [Amazon ECR용 VPC 엔드포인트](#)를 사용하십시오. VPC 엔드포인트는 프라이빗 IP 주소를 통해 Amazon ECR API에 비공개로 액세스할 수 있게 해주는 기술인 PrivateLink AWS를 기반으로 합니다. Fargate 시작 유형을 사용하는 Amazon ECS 작업의 경우, VPC 엔드포인트를 사용하면 작업에 퍼블릭 IP 주소를 할당하지 않고도 Amazon ECR에서 프라이빗 이미지를 가져올 수 있습니다.

에픽

OIDC 공급자 및 리포지토리를 설정합니다. GitHub

작업	설명	필요한 기술
OpenID Connect를 구성합니다.	OIDC (OpenID Connect) 공급자를 만드세요. 이 작업에 사용되는 IAM 역할에 대한 신뢰 정책의 공급자를 사용하게 됩니다. 자세한 지침은 설명서의 Amazon Web Services에서 OpenID Connect 구성을 참조 하십시오. GitHub	AWS 관리자, AWS DevOps, 일반 AWS
GitHub 리포지토리를 복제합니다.	GitHub Docker ECR 작업 워크플로 리포지토리를 로컬 폴더에 복제합니다. <pre>\$git clone https://github.com/aws-samp</pre>	DevOps 엔지니어

작업	설명	필요한 기술
	<pre>les/docker-ecr-actions-workflow</pre>	

GitHub 재사용 가능한 워크플로를 사용자 지정하고 Docker 이미지를 배포하세요.

작업	설명	필요한 기술
<p>Docker 워크플로를 시작하는 이벤트를 사용자 지정합니다.</p>	<p>이 솔루션의 워크플로는 workflow.yaml에 있습니다. 이 스크립트는 현재 이벤트 수신 시 리소스를 배포하도록 구성되어 있습니다. workflow_dispatch 이벤트를 다른 상위 워크플로로 변경하고 워크플로를 workflow_call 호출하여 이 구성을 사용자 지정할 수 있습니다.</p>	<p>DevOps 엔지니어</p>
<p>워크플로를 사용자 지정하세요.</p>	<p>workflow.yaml 파일은 재사용이 가능한 동적 워크플로를 생성하도록 구성되어 있습니다. GitHub 이 파일을 편집하여 기본 구성을 사용자 지정하거나 workflow_dispatch 이벤트를 사용하여 수동으로 배포를 시작하는 경우 GitHub Actions 콘솔의 입력 값을 전달할 수 있습니다.</p> <ul style="list-style-type: none"> 올바른 AWS 계정 ID와 대상 지역을 지정해야 합니다. Amazon ECR 수명 주기 정책 (샘플 정책 참조) 을 생성하고 그에 따라 기본 경로 	<p>DevOps 엔지니어</p>

작업	설명	필요한 기술
	<p>(<code>e2e-test/policy.json</code>) 를 업데이트하십시오.</p> <ul style="list-style-type: none"> 워크플로 파일에는 입력으로 두 개의 IAM 역할이 필요합니다. Terraform용 Amazon S3 백엔드를 설정할 권한이 있는 IAM 역할 (사전 요구 사항 섹션 참조). 에서 기본 역할 이름을 업데이트할 수 있습니다. <code>workload-assumable-role.yaml</code> 파일을 적절히 입력하세요. 액세스 GitHub 권한이 있는 IAM 역할. 또한 이 역할은 Amazon ECR 정책에서 Amazon ECR 작업을 제한하는 데 사용됩니다. 자세한 내용은 data.tf 파일을 참조하십시오. 	
<p>테라폼 템플릿을 배포하세요.</p>	<p>워크플로는 구성된 이벤트를 기반으로 Amazon ECR 리포지토리를 생성하는 Terraform 템플릿을 자동으로 배포합니다 GitHub . 이러한 템플릿은 Github 리포지토리의 루트에서 .tf 파일로 제공됩니다.</p>	<p>AWS DevOps, DevOps 엔지니어</p>

문제 해결

문제	Solution
<p>Amazon S3 및 DynamoDB를 Terraform 원격 백엔드로 구성할 때 발생하는 문제 또는 오류가 발생합니다.</p>	<p>Terraform 설명서의 지침에 따라 원격 백엔드 구성을 위한 Amazon S3 및 DynamoDB 리소스에 필요한 권한을 설정합니다.</p>
<p>이벤트와 함께 워크플로를 실행하거나 시작할 수 없습니다. workflow_dispatch</p>	<p>workflow_dispatch 이벤트에서 배포하도록 구성된 워크플로는 워크플로가 기본 브랜치에도 구성된 경우에만 작동합니다.</p>

관련 리소스

- [워크플로 재사용](#) (GitHub 문서)
- [워크플로우 트리거 \(문서\)](#) GitHub

AWS CodeCommit, AWS 및 AWS Device Farm을 사용하여 iOS 앱을 구축하고 테스트할 수 있습니다. CodePipeline

작성자: Abdullahi Olaoye(AWS)

R 유형: 해당 없음	출처: 온프레미스 프로세스 DevOps	타겟: AWS 기반 iOS 앱 개발을 위한 CI/CD 파이프라인
작성자: AWS	환경: PoC 또는 파일럿	기술: 웹 및 모바일 앱, DevOps

AWS 서비스: AWS, AWS CodeCommit CodePipeline, AWS Device Farm

요약

이 패턴은 AWS를 사용하여 AWS의 실제 디바이스에서 iOS 애플리케이션을 구축하고 CodePipeline 테스트하는 지속적 통합 및 지속적 전송 (CI/CD) 파이프라인을 생성하는 단계를 설명합니다. 이 패턴은 AWS를 CodeCommit 사용하여 애플리케이션 코드를 저장하고, Jenkins 오픈 소스 도구를 사용하여 iOS 애플리케이션을 구축하며, AWS Device Farm을 사용하여 실제 디바이스에서 빌드된 애플리케이션을 테스트합니다. 이 세 단계는 AWS를 사용하여 하나의 파이프라인에서 함께 오케스트레이션됩니다. CodePipeline

이 패턴은 AWS 블로그의 [AWS DevOps 및 모바일 서비스를 사용하여 iOS 및 iPadOS 앱을 구축하고 테스트하는](#) 게시물을 기반으로 합니다. DevOps 자세한 지침은 블로그 게시물을 참조하세요.

사전 조건 및 제한 사항

사전 조건

- 활성 상태의 AWS 계정.
- Apple 개발자 계정
- 빌드 서버(macOS)
- [Xcode](#) 버전 11.3(빌드 서버에 설치 및 설정)
- 워크스테이션에 [설치](#) 및 [구성된](#) AWS Command Line Interface(AWS CLI)
- [Git](#)에 대한 기본 지식

제한 사항

- 애플리케이션 빌드 서버는 macOS를 실행해야 합니다.
- 빌드 서버에 퍼블릭 IP 주소가 있어야 원격으로 연결하여 빌드를 시작할 CodePipeline 수 있습니다.

아키텍처

소스 기술 스택

- 물리적 기기에서 시뮬레이터 또는 수동 테스트를 사용하는 온프레미스 iOS 애플리케이션 빌드 프로세스

대상 기술 스택

- 애플리케이션 소스 코드를 저장하는 AWS CodeCommit 리포지토리
- Xcode를 사용하여 애플리케이션을 빌드하기 위한 Jenkins 서버
- 실제 디바이스에서 애플리케이션을 테스트하기 위한 AWS Device Farm 디바이스 풀

대상 아키텍처

사용자가 소스 리포지토리에 변경 사항을 커밋하면 파이프라인 (AWS CodePipeline) 이 소스 리포지토리에서 코드를 가져와 Jenkins 빌드를 시작하고 애플리케이션 코드를 Jenkins에 전달합니다. 빌드 후 파이프라인은 빌드 아티팩트를 검색하고 AWS Device Farm 작업을 시작하여 디바이스 풀에서 애플리케이션을 테스트합니다.

도구

- [CodePipelineAWS](#)는 빠르고 안정적인 애플리케이션 및 인프라 업데이트를 위해 릴리스 파이프라인을 자동화하는 데 도움이 되는 완전관리형 지속적 전송 서비스입니다. CodePipeline 정의한 릴리스 모델을 기반으로 코드가 변경될 때마다 릴리스 프로세스의 빌드, 테스트 및 배포 단계를 자동화합니다.
- [CodeCommitAWS](#)는 안전한 Git 기반 리포지토리를 호스팅하는 완전 관리형 소스 제어 서비스입니다. 이를 통해 팀은 안전하고 확장성이 뛰어난 에코시스템에서 코드 작업을 손쉽게 수행할 수 있습니다. CodeCommit 자체 소스 제어 시스템을 운영하거나 인프라 확장에 대해 걱정할 필요가 없습니다.

- [AWS Device Farm](#)은 테스트 인프라를 프로비저닝하고 관리할 필요 없이 광범위한 데스크톱 브라우저와 실제 모바일 디바이스에서 테스트하여 웹 및 모바일 앱의 품질을 개선할 수 있는 애플리케이션 테스트 서비스입니다.
- [Jenkins](#)는 개발자가 그의 소프트웨어를 빌드, 테스트 및 배포할 수 있는 오픈 소스 자동화 서버입니다.

에픽

빌드 환경 설정

작업	설명	필요한 기술
macOS를 실행하는 빌드 서버에 Jenkins를 설치합니다.	Jenkins가 애플리케이션을 빌드하는 데 사용되므로 먼저 빌드 서버에 Jenkins 명령을 사용해야 합니다. 이 작업과 후속 작업에 대한 자세한 지침은 이 패턴 끝에 있는 관련 리소스 섹션에서 AWS, 모바일 서비스 및 기타 리소스를 사용하여 iOS DevOps 및 iPadOS 앱을 구축하고 테스트하는 AWS 블로그 게시물을 참조하십시오.	DevOps
Jenkins를 설정합니다.	화면에 표시되는 지시 사항에 따라 Jenkins를 구성합니다.	DevOps
젠킨스용 AWS CodePipeline 플러그인을 설치합니다.	Jenkins가 AWS CodePipeline 서비스와 상호 작용하려면 Jenkins 서버에 이 플러그인을 설치해야 합니다.	DevOps
Jenkins 프리스타일 프로젝트를 생성하세요.	Jenkins에서 프리스타일 프로젝트를 만드세요. 트리거 및 기타 빌드 구성 옵션을 지정하도록 프로젝트를 구성합니다.	DevOps

AWS Device Farm을 구성합니다.

작업	설명	필요한 기술
Device Farm 프로젝트를 생성합니다.	AWS Device Farm 콘솔을 엽니다. 테스트용 프로젝트와 디바이스 풀을 생성합니다. 지침은 블로그 게시물을 참조하세요.	개발자

소스 리포지토리 구성

작업	설명	필요한 기술
리포지토리를 생성하십시오. CodeCommit	소스 코드를 저장할 리포지토리를 만드세요.	DevOps
리포지토리에 애플리케이션 코드를 커밋합니다.	생성한 CodeCommit 리포지토리에 연결합니다. 로컬 시스템에서 리포지토리로 코드를 푸시합니다.	DevOps

파이프라인 구성

작업	설명	필요한 기술
AWS에서 파이프라인을 생성합니다 CodePipeline.	AWS CodePipeline 콘솔을 열고 파이프라인을 생성합니다. 파이프라인은 CI/CD 프로세스의 모든 단계를 조정합니다. 지침은 AWS 및 모바일 서비스를 사용하여 iOS 및 iPadOS 앱을 구축하고 테스트하는 AWS DevOps 블로그 게시물을 참조하십시오.	DevOps

작업	설명	필요한 기술
파이프라인에 테스트 단계를 추가합니다.	테스트 단계를 추가하고 이를 AWS Device Farm과 통합하려면 파이프라인을 편집하십시오.	DevOps
파이프라인을 시작합니다.	파이프라인 및 CI/CD 프로세스를 시작하려면 [Release change] 를 선택합니다.	DevOps

애플리케이션 테스트 결과 보기

작업	설명	필요한 기술
테스트 결과를 확인합니다.	AWS Device Farm 콘솔에서 생성한 프로젝트를 선택하고 테스트 결과를 검토합니다. 콘솔에는 각 테스트의 세부 정보가 표시됩니다.	개발자

관련 리소스

이 패턴에 대한 [tep-by-step 지침](#)

- [AWS 및 모바일 서비스를 사용하여 iOS 및 iPadOS 앱을 구축하고 테스트하기](#) (AWS DevOps 블로그 게시물) DevOps

AWS Device Farm 구성

- [AWS Device Farm 콘솔](#)

소스 리포지토리 구성

- [AWS CodeCommit 리포지토리 생성](#)
- [AWS CodeCommit 리포지토리에 연결](#)

파이프라인 구성

- [AWS CodePipeline 콘솔](#)

추가 리소스

- [AWS CodePipeline 설명서](#)
- [AWS CodeCommit 설명서](#)
- [AWS Device Farm 설명서](#)
- [Jenkins 설명서](#)
- [macOS에 Jenkins 설치](#)
- [젠킨스용 AWS CodePipeline 플러그인](#)
- [Xcode 설치](#)
- [AWS CLI 설치 및 구성](#)
- [Git 설명서](#)

cdk-nag 규칙 팩을 사용하여 AWS CDK 애플리케이션 또는 CloudFormation 템플릿에서 모범 사례를 확인하십시오.

작성자: Arun Donti

환경: 프로덕션

기술: DevOps; 보안, ID, 규정 준수

워크로드: 오픈 소스

AWS 서비스: AWS CDK

요약

이 패턴은 규칙 팩 조합하여 [cdk-nag](#) 유틸리티를 사용하여 [AWS Cloud Development Kit\(AWS CDK\)](#) 애플리케이션에서 모범 사례를 확인할 방법을 설명합니다. cdk-nag는 [cfn_nag](#)에서 영감을 받은 오픈 소스 프로젝트입니다. [AWS CDK Aspect](#)를 사용하여 AWS Solutions Library, 미국 건강 보험 양도 및 책임에 관한 법(HIPAA)과 국립 표준 기술 연구소(NIST) 800-53과 같은 평가 팩의 규칙을 구현합니다. 이러한 팩의 규칙을 사용하여 AWS CDK 애플리케이션에서 모범 사례를 확인하고, 모범 사례를 기반으로 코드를 탐지 및 문제를 해결하고, 평가에 사용하지 않으려는 규칙을 제한할 수 있습니다.

[또한 cloudformation-include 모듈을 사용하여 cdk-nag를 사용하여 AWS CloudFormation 템플릿을 확인할 수 있습니다.](#)

사용 가능한 모든 팩에 대한 자세한 내용은 [cdk-nag](#) 리포지토리의 [규칙](#) 섹션을 참조하세요. 평가 팩은 다음과 같은 경우에 사용할 수 있습니다.

- [AWS Solutions Library](#)
- [HIPAA 보안](#)
- [NIST 800-53 버전 4](#)
- [NIST 800-53 버전 5](#)
- [지불 카드 산업 데이터 보안 표준\(PCI DSS\) 3.2.1](#)

사전 조건 및 제한 사항

사전 조건

- [AWS CDK](#)를 사용하는 애플리케이션

도구

- [AWS CDK](#) — 클라우드 개발 키트 (AWS CDK) 는 코드로 클라우드 인프라를 정의하고 AWS를 통해 프로비저닝하기 위한 소프트웨어 개발 프레임워크입니다. CloudFormation
- [AWS CloudFormation](#) — AWS는 AWS CloudFormation 리소스를 모델링 및 설정하고, 빠르고 일관되게 프로비저닝하고, 수명 주기 전반에 걸쳐 관리할 수 있도록 지원합니다. 템플릿을 사용하여 리소스와 종속성을 설명하고 리소스를 개별적으로 관리하는 대신 스택으로 함께 시작하고 구성할 수 있습니다. 여러 AWS 계정 및 AWS 리전에서 스택을 관리하고 프로비저닝할 수 있습니다.

에픽

cdk-nag를 AWS CDK 애플리케이션과 통합

작업	설명	필요한 기술
cdk-nag에 대해 알아봅니다.	cdk-nag GitHub 리포지토리로 이동하여 설명서를 읽어보십시오.	앱 개발자
AWS CDK 애플리케이션에 cdk-nag 패키지를 설치합니다.	AWS CDK 애플리케이션에서 cdk-nag를 사용하려면 먼저 CDK-nag를 설치해야 합니다. cdk-nag는 PyPi, npm, 아파치 메이븐에서 다운로드할 수 있습니다. NuGet 사용 가능한 버전 및 다운로드 위치에 대한 최신 정보는 리포지토리의 Readme 파일 을 참조하세요.	앱 개발자
NagPacks원하는 것을 선택하세요.	cdk-nag에는 이름이 다른 규칙 팩이 있습니다. NagPacks 각 NagPack 항목에는 특정 표준을 준수하는 규칙이 포함되어 있습니다. 예를 들어, AWS NagPack 솔루션에는 일반적인 모범 사례가 포함되어 있으며	앱 개발자

작업	설명	필요한 기술
	<p>NIST 800-53 rev 5는 규정 준수에 도움이 NagPack 될 수 있습니다. 애플리케이션에 여러 NagPacks 개를 적용할 수 있으며 필요에 따라 팩을 추가 및 제거할 수 있습니다. 사용 가능한 팩 목록은 GitHub 저장소의 Readme 파일을 참조하십시오. 각 팩의 개별 규칙에 대한 자세한 내용은 GitHub 리포지토리의 규칙 섹션을 참조하십시오.</p>	

작업	설명	필요한 기술
<p>cdk-nag를 AWS CDK 애플리케이션에 통합합니다.</p>	<p>cdk-nag를 애플리케이션 전체 수준에서 애플리케이션에 통합하거나 애플리케이션의 개별 단계 또는 스택에 통합할 수 있습니다. 예를 들어, 애플리케이션 전체 수준에서 AWS 솔루션과 HIPAA 보안을 NagPacks AWS CDK v2 TypeScript 애플리케이션에 통합하려면 다음 코드를 사용할 수 있습니다.</p> <pre data-bbox="597 730 1026 1717"> import { App, Aspects } from 'aws-cdk-lib'; import { CdkTestStack } from '../lib/cdk-test-stack'; import { AwsSolutionsChecks, HIPAASecurityChecks } from 'cdk-nag'; const app = new App(); new CdkTestStack(app, 'CdkNagDemo'); // Simple rule informational messages Aspects.of(app).add(new AwsSolutionsChecks()); // Additional explanations on the purpose of triggered rules Aspects.of(app).add(new HIPAASecurityChecks({ verbose: true })); </pre>	<p>앱 개발자</p>

관련 리소스

- [cdk-nag 코드 리포지토리](#)
- [Construct Hub의 cdk-nag](#)

Amazon DynamoDB에 대한 크로스 계정 액세스 구성

작성자: Shashi Dalmia(AWS) 및 Jay Enjamoori(AWS)

환경: 프로덕션

기술: DevOps; 데이터베이스;
보안, ID, 규정 준수

AWS 서비스: Amazon
DynamoDB, AWS Identity and
Access Management, AWS
Lambda

요약

이 패턴은 Amazon DynamoDB에 대한 크로스 계정 액세스 구성 단계를 설명합니다. Amazon Web Services(AWS) 서비스는 데이터베이스에 적절한 AWS ID 및 액세스 관리(IAM) 권한이 설정된 경우 동일한 AWS 계정에 있는 DynamoDB 테이블에 액세스할 수 있습니다. 하지만 다른 AWS 계정에서 액세스하려면 IAM 권한을 설정하고 두 계정 간에 신뢰 관계를 설정해야 합니다.

이 패턴은 한 계정에서 다른 계정의 DynamoDB 테이블을 읽고 쓰도록 AWS Lambda 함수를 구성하는 방법을 보여주는 단계 및 샘플 코드를 제공합니다.

사전 조건 및 제한 사항

- 활성 상태의 AWS 계정 두 개. 이 패턴은 이러한 계정을 계정 A와 계정 B라고 합니다.
- DynamoDB 데이터베이스를 생성하기 위해 계정 A에 액세스하도록 AWS Command Line Interface(AWS CLI)가 [설치](#) 및 [구성](#)되었습니다. 이 패턴의 다른 단계는 IAM, DynamoDB 및 Lambda 콘솔 사용에 대한 지침을 제공합니다. AWS CLI를 대신 사용할 계획이라면 두 계정 모두에 액세스할 수 있도록 구성하십시오.

아키텍처

다음 다이어그램에서 AWS Lambda, Amazon EC2 및 DynamoDB는 모두 동일한 계정에 있습니다. 이 시나리오에서는 Lambda 함수와 Amazon Elastic Compute Cloud(Amazon EC2) 인스턴스에서 DynamoDB에 액세스할 수 있습니다.

다른 AWS 계정의 리소스가 DynamoDB에 액세스하려는 경우 크로스 계정 액세스 및 신뢰 관계를 설정해야 합니다. 예를 들어, 다음 다이어그램에서 계정 A의 DynamoDB와 계정 B의 Lambda 함수 간에 액

세스를 활성화하려면 [에픽](#) 섹션에 설명된 대로 계정 간에 신뢰 관계를 생성하고 Lambda 서비스와 사용자에게 적절한 액세스 권한을 부여해야 합니다.

도구

서비스

- [Amazon DynamoDB](#)는 완전관리형 NoSQL 데이터베이스 서비스로서 원활한 확장성과 함께 빠르고 예측 가능한 성능을 제공합니다.
- [AWS Lambda](#)는 서버를 프로비저닝하거나 관리하지 않고도 코드를 실행할 수 있게 해주는 컴퓨팅 서비스입니다. Lambda는 필요 시에만 코드를 실행하며, 일일 몇 개의 요청에서 초당 수천 개의 요청까지 자동으로 규모를 조정합니다. 사용한 컴퓨팅 시간만큼만 비용을 지불하고, 코드가 실행되지 않을 때는 요금이 부과되지 않습니다.
- [AWS Identity and Access Management\(IAM\)](#)를 사용하면 사용자에 대해 인증 및 권한 부여를 제어함으로써 AWS 리소스에 대한 액세스를 안전하게 관리할 수 있습니다.

코드

이 패턴에는 계정 A의 DynamoDB 테이블에서 쓰고 읽도록 계정 B의 Lambda 함수를 구성하는 방법을 설명하는 [추가 정보](#) 섹션의 샘플 코드가 포함되어 있습니다. 코드는 설명 및 테스트 목적으로만 제공됩니다. 프로덕션 환경에서 이 패턴을 구현하는 경우 코드를 참조로 사용하고 사용자 환경에 맞게 사용자 정의하십시오.

이 패턴은 Lambda 및 DynamoDB를 사용한 크로스 계정 액세스를 보여줍니다. 다른 AWS 서비스에도 동일한 단계를 사용할 수 있지만 두 계정 모두에 적절한 권한을 부여하고 구성해야 합니다. 예를 들어 계정 A의 Amazon Relational Database Service(RDS) 데이터베이스에 대한 액세스 권한을 부여하려면 해당 데이터베이스의 역할을 생성하고 신뢰 관계에 바인딩하십시오. 계정 B에서 AWS Lambda 대신 Amazon EC2를 사용하려면 해당 IAM 정책 및 역할을 생성한 다음 이를 EC2 인스턴스에 연결합니다.

에픽

계정 A에서 DynamoDB 테이블 생성

작업	설명	필요한 기술
계정 A에서 DynamoDB 테이블을 생성합니다.	계정 A에 대해 AWS CLI를 구성한 후 다음 AWS CLI 명령을	AWS DevOps

작업	설명	필요한 기술
	<p>사용하여 DynamoDB 테이블을 생성합니다.</p> <pre>aws dynamodb create-table \ --table-name Table- Account-A \ --attribute-definitions \ Attribute Name=category,AttributeType=S \ Attribute Name=item,AttributeType=S \ --key-schema \ Attribute Name=category,KeyType=HASH \ Attribute Name=item,KeyType=RANGE \ --provisioned-throughput \ ReadCapacityUnits=5,WriteCapacityUnits=5</pre> <p>테이블 생성에 대한 자세한 내용은 DynamoDB 설명서를 참조하십시오.</p>	

계정 A에서 역할 생성

작업	설명	필요한 기술
계정 A에서 역할을 생성합니다.	계정 B는 이 역할을 사용하여 계정 A에 액세스할 수 있는 권	AWS DevOps

작업	설명	필요한 기술
	<p>한을 연습합니다. 역할을 생성하려면:</p> <ol style="list-style-type: none"> 1. <a href="https://<account-ID-for-Account-A>.signin.aws.amazon.com/console">https://<account-ID-for-Account-A>.signin.aws.amazon.com/console 에서 계정 A에 로그인합니다. 2. https://console.aws.amazon.com/iam/에서 IAM 콘솔을 엽니다. 3. 콘솔의 탐색 창에서 역할을 선택한 후 역할 생성을 선택합니다. 4. 신뢰할 수 있는 엔티티에서 AWS 계정을 선택하고, AWS 계정 섹션에서 다른 AWS 계정을 선택합니다. 5. 계정 ID에 계정 B의 ID를 입력합니다. 6. 다음: 권한을 선택합니다. 7. 필터 정책란에 DynamoDB를 입력합니다. 8. DynamoDB 정책 목록에서 DB를 선택합니다. AmazonDynamo FullAccess <p>참고: 이 정책은 DynamoDB에서의 모든 작업을 허용합니다. 보안 모범 사례로서 항상 필요한 권한만 부여해야 합니다. 대신 선택할 수 있는 다른 정책 목록은 IAM 설명</p>	

작업	설명	필요한 기술
	<p>서의 정책 예제를 참조하십시오.</p> <p>9. 다음: 이름, 검토 및 생성을 선택합니다.</p> <p>10. 역할 이름에 역할의 고유한 이름 (예: DynamoDB FullAccess - -For-Account-B) 을 입력하고 역할 설명 (선택 사항) 을 추가합니다.</p> <p>11. 모든 섹션을 검토하고 (선택적으로) 태그를 키-값 페어로 연결하여 메타데이터를 역할에 추가합니다.</p> <p>12. Create role(역할 생성)을 선택합니다.</p> <p>역할 생성에 대한 자세한 내용은 IAM 설명서를 참조하십시오.</p>	
<p>계정 A에서 역할에 대한 ARN을 기록합니다.</p>	<ol style="list-style-type: none"> 1. IAM 콘솔의 탐색 창에서 역할을 선택합니다. 2. 검색 상자에 DynamoDB FullAccess - -For-Account-B (또는 이전 스토리에서 만든 역할 이름) 를 입력하고 역할을 선택합니다. 3. 역할에 대한 요약 페이지에서 Amazon 리소스 이름 (ARN)을 복사합니다. 계정 B에서 Lambda 코드를 설정할 때 ARN을 사용하게 됩니다. 	<p>AWS DevOps</p>

계정 B에서 계정 A에 대한 액세스 구성

작업	설명	필요한 기술
계정 A에 액세스하기 위한 정책을 생성합니다.	<ol style="list-style-type: none"> 1. <code>https://<account-ID-for-Account-B>.signin.aws.amazon.com/console</code> 에서 계정 B에 로그인합니다. 2. https://console.aws.amazon.com/iam/에서 IAM 콘솔을 엽니다. 3. 콘솔의 탐색 창에서 정책을 선택한 후 정책 생성을 선택합니다. 4. JSON 탭을 선택합니다. 5. 다음 JSON 문서를 입력하거나 붙여넣습니다. <pre data-bbox="630 1062 1029 1818"> { "Version": "2012-10-17", "Statement": [{ "Effect": "Allow", "Action": "sts:AssumeRole", "Resource ": "arn:aws: iam::<Account-A-ID >:role/DynamoDB-Fu llAccess-For-Accou nt-B" }] } </pre> 	AWS DevOps

작업	설명	필요한 기술
	<p>여기서 Resource 속성에는 이전 스토리에서 계정 A에서 생성한 역할의 ARN이 포함됩니다.</p> <ol style="list-style-type: none"> 6. 다음: 태그를 선택합니다. 7. (선택 사항) 태그를 키 값 페어로 연결하여 메타데이터를 정책에 추가합니다. 8. 다음: 검토를 선택합니다. 9. 정책 이름에 정책의 고유한 이름 (예: DynamoDB FullAccess - -Policy-in-Account-A) 을 입력하고 선택적 정책 설명을 추가합니다. 10. 정책 생성을 선택합니다. <p>정책 생성에 대한 자세한 내용은 IAM 설명서를 참조하십시오.</p>	

작업	설명	필요한 기술
<p>이 정책에 따라 역할을 생성합니다.</p>	<p>이 역할은 계정 B의 Lambda 함수가 계정 A의 DynamoDB 테이블에서 읽고 쓰는 데 사용됩니다.</p> <ol style="list-style-type: none"> 계정 B에서 IAM 콘솔의 탐색 창에서 역할을 선택하고 역할 생성을 선택합니다. 신뢰할 수 있는 엔터티 유형 선택에서 AWS 서비스를 선택합니다. 사용 사례에서 Lambda를 선택합니다. 다음: 권한을 선택합니다. 필터 정책란에 DynamoDB를 입력합니다. DynamoDB 정책 목록에서 이전 스토리에서 생성한 DynamoDB FullAccess - -계정 내 정책-A를 선택합니다. 다음: 이름, 검토 및 생성을 선택합니다. 역할 이름에는 역할의 고유한 이름 (예: DynamoDB FullAccess - -in-Account-A)을 입력하고 역할 설명 (선택 사항)을 추가합니다. 모든 섹션을 검토하고 (선택적으로) 태그를 키-값 페어로 연결하여 메타데이터를 역할에 추가합니다. Create role(역할 생성)을 선택합니다. 	<p>AWS DevOps</p>

작업	설명	필요한 기술
	<p>이제 다음 에픽의 Lambda 함수에 이 역할을 연결할 수 있습니다.</p> <p>역할 생성에 대한 자세한 내용은 IAM 설명서를 참조하십시오.</p>	

계정 B에서 Lambda 함수 생성

작업	설명	필요한 기술
DynamoDB에 데이터를 쓰기 위해 Lambda 함수를 생성합니다.	<ol style="list-style-type: none"> <a href="https://<account-ID-for-Account-B>.signin.aws.amazon.com/console">https://<account-ID-for-Account-B>.signin.aws.amazon.com/console 에서 계정 B에 로그인합니다. https://console.aws.amazon.com/lambda/에서 Lambda 콘솔을 엽니다. 콘솔의 탐색 창에서 함수를 선택한 후 함수 생성을 선택합니다. 이름에 <code>lambda_write_function</code>을 입력합니다. 런타임에서 Python 3.8 이상을 선택합니다. 권한에서 기본 실행 역할 변경, 기존 역할 사용을 선택합니다. 기존 역할의 경우 FullAccessDynamoDB - -인-어카운트-A를 선택합니다. 	AWS DevOps

작업	설명	필요한 기술
	<p>8. 함수 생성을 선택합니다.</p> <p>9. 코드 탭에서 추가 정보 섹션에 제공된 Lambda 쓰기 함수 샘플 코드를 이 패턴으로 붙여넣습니다. RoleArn 필드에 올바른 역할 ARN(계정 A에서 역할 생성 에픽에서) 을 제공하고 계정 A에서 DynamoDB 테이블이 생성되는 위치 (계정 A에서 DynamoDB 테이블 생성 에픽에서)로 region_name 을 변경해야 합니다. 이렇게 하지 않으면 ResourceNotFoundException 오류가 발생합니다.</p> <p>10. 코드를 배포하려면 배포를 선택합니다.</p> <p>11. 함수를 실행하려면 테스트를 선택합니다. 그러면 테스트 이벤트를 구성하라는 메시지가 표시됩니다. 원하는 이름 (예:) 으로 새 이벤트를 생성하고 구성을 저장합니다. MyTestEventForWrite</p> <p>12. 함수를 다시 실행하려면 테스트를 선택합니다. 그러면 입력한 이벤트 이름으로 코드가 실행됩니다.</p> <p>13. 함수의 출력을 확인합니다. 추가 정보의 Lambda 쓰기 함수 섹션에 표시된 출력과 비슷해야 합니다.</p>	

작업	설명	필요한 기술
	<p>이 출력은 함수가 계정 A의 DynamoDB 테이블에 액세스하여 데이터를 쓸 수 있었음을 나타냅니다.</p> <p>Lambda 함수 생성에 대한 자세한 내용은 Lambda 설명서를 참조하십시오.</p>	

작업	설명	필요한 기술
<p>DynamoDB에서 데이터를 읽기 위해 Lambda 함수를 생성합니다.</p>	<ol style="list-style-type: none"> 1. Lambda 콘솔의 탐색 창에서 함수를 선택한 후 함수 생성을 선택합니다. 2. 이름에 <code>lambda_read_function</code>을 입력합니다. 3. 런타임에서 Python 3.8 이상을 선택합니다. 4. 권한에서 기본 실행 역할 변경, 기존 역할 사용을 선택합니다. 5. 기존 역할의 경우 FullAccessDynamoDB --인-어카운트-A를 선택합니다. 6. 함수 생성을 선택합니다. 7. 코드 탭에서 추가 정보 섹션에 제공된 Lambda 읽기 함수 샘플 코드를 이 패턴으로 붙여넣습니다. RoleArn 필드에 올바른 역할 ARN(계정 A에서 역할 생성 에픽에서)을 제공하고 계정 A에서 DynamoDB 테이블이 생성되는 위치(계정 A에서 DynamoDB 테이블 생성 에픽에서)로 <code>region_name</code> 을 변경해야 합니다. 이렇게 하지 않으면 <code>ResourceNotFoundException</code> 오류가 발생합니다. 8. 코드를 배포하려면 배포를 선택합니다. 	<p>AWS DevOps</p>

작업	설명	필요한 기술
	<p>9. 함수를 실행하려면 테스트를 선택합니다. 그러면 테스트 이벤트를 구성하라는 메시지가 표시됩니다. 원하는 이름 (예:) 으로 새 이벤트를 생성하고 구성을 저장합니다. MyTestEventForRead</p> <p>10. 함수를 다시 실행하려면 테스트를 선택합니다. 그러면 입력한 이벤트 이름으로 코드가 실행됩니다.</p> <p>11. 함수의 출력을 확인합니다. 추가 정보의 Lambda 읽기 함수 섹션에 표시된 출력과 비슷해야 합니다. 이 출력은 함수가 계정 A의 DynamoDB 테이블에 액세스하여 테이블에 추가한 데이터를 읽을 수 있었음을 나타냅니다.</p> <p>Lambda 함수 생성에 대한 자세한 내용은 Lambda 설명서를 참조하십시오.</p>	

리소스 정리

작업	설명	필요한 기술
생성한 리소스를 삭제합니다.	테스트 또는 개념 증명(PoC) 환경에서 이 패턴을 실행하는 경우, 생성한 리소스를 삭제하여 비용이 발생하지 않도록 하십시오.	AWS DevOps

작업	설명	필요한 기술
	<ol style="list-style-type: none"> 계정 B에서 DynamoDB에 연결하기 위해 생성한 Lambda 함수 2개와 기타 리소스를 삭제합니다. 계정 A에서 생성한 DynamoDB 테이블을 삭제합니다. IAM 정책은 비용이 들지 않으므로 그대로 유지할 수 있습니다. 그러나 보안을 위해 이 패턴에 대해 생성한 다음 역할 및 정책을 삭제하는 것이 좋습니다. <ul style="list-style-type: none"> 계정 A: DymamoDB-Full-Access-for-Account-A 역할 계정 B: DynamoDB-FullAccess -계정 내 역할 계정 B: DynamoDB- -계정 내 FullAccess 정책-A 정책 	

관련 리소스

- [AWS CLI 시작하기](#)(AWS CLI 설명서)
- [AWS CLI 구성](#)(AWS CLI 설명서)
- [DynamoDB 시작하기](#)(DynamoDB 설명서)
- [Lambda 시작하기](#)(AWS Lambda 설명서)
- [역할을 생성하여 IAM 사용자에게 권한 위임](#)(IAM 설명서)
- [IAM 정책 생성](#)(IAM 설명서)
- [크로스 계정 정책 평가 로직](#)(IAM 설명서)
- [IAM JSON 정책 요소 참조](#)(IAM 설명서)

추가 정보

이 섹션의 코드는 설명 및 테스트 목적으로만 제공됩니다. 프로덕션 환경에서 이 패턴을 구현하는 경우 코드를 참조로 사용하고 사용자 환경에 맞게 사용자 정의하십시오.

Lambda 쓰기 함수

샘플 코드

```
import boto3
from datetime import datetime

sts_client = boto3.client('sts')
sts_session = sts_client.assume_role(RoleArn='arn:aws:iam::<Account-A ID>:role/
DynamoDB-FullAccess-For-Account-B', RoleSessionName='test-dynamodb-session')

KEY_ID = sts_session['Credentials']['AccessKeyId']
ACCESS_KEY = sts_session['Credentials']['SecretAccessKey']
TOKEN = sts_session['Credentials']['SessionToken']

dynamodb_client = boto3.client('dynamodb',
                                region_name='<DynamoDB-table-region-in-account-A>',
                                aws_access_key_id=KEY_ID,
                                aws_secret_access_key=ACCESS_KEY,
                                aws_session_token=TOKEN)

def lambda_handler(event, context):
    now = datetime.now()
    date_time = now.strftime("%m/%d/%Y, %H:%M:%S")
    data = dynamodb_client.put_item(TableName='Table-Account-A', Item={"category":
{"S": "Fruit"}, "item": {"S": "Apple"}, "time": {"S": date_time}})
    return data
```

샘플 출력

Lambda 읽기 함수

샘플 코드

```
import boto3
from datetime import datetime

sts_client = boto3.client('sts')
sts_session = sts_client.assume_role(RoleArn='arn:aws:iam::<Account-A ID>:role/
DynamoDB-FullAccess-For-Account-B', RoleSessionName='test-dynamodb-session')

KEY_ID = sts_session['Credentials']['AccessKeyId']
ACCESS_KEY = sts_session['Credentials']['SecretAccessKey']
TOKEN = sts_session['Credentials']['SessionToken']

dynamodb_client = boto3.client('dynamodb',
                                region_name='<DynamoDB-table-region-in-account-A>',
                                aws_access_key_id=KEY_ID,
                                aws_secret_access_key=ACCESS_KEY,
                                aws_session_token=TOKEN)

def lambda_handler(event, context):
    response = dynamodb_client.get_item(TableName='Table-Account-A', Key={'category':
{'S':'Fruit'}, 'item':{'S':'Apple'}})
    return response
```

샘플 출력

Amazon EKS에서 실행되는 애플리케이션에 대한 상호 TLS 인증을 구성합니다.

작성자: Mahendra Siddappa(AWS)

환경: PoC 또는 파일럿

기술: DevOps; 보안, ID, 규정 준수

AWS 서비스: Amazon EKS, Amazon Route 53

요약

인증서 기반 상호 전송 계층 보안(TLS)은 서버와 클라이언트 간에 양방향 피어 인증을 제공하는 선택적 TLS 구성 요소입니다. 상호 TLS를 사용하는 경우 클라이언트는 세션 협상 프로세스 중에 X.509 인증서를 제공해야 합니다. 서버는 이 인증서를 사용하여 클라이언트를 식별하고 인증합니다.

상호 TLS는 사물 인터넷 (IoT) 애플리케이션의 일반적인 요구 사항이며 [오픈 बैंकिंग과](#) 같은 business-to-business 애플리케이션 또는 표준에 사용할 수 있습니다.

이 패턴은 NGINX 인그레스 컨트롤러를 사용하여 Amazon Elastic Kubernetes Service(Amazon EKS) 클러스터에서 실행되는 애플리케이션의 상호 TLS를 구성하는 방법을 설명합니다. 인그레스 리소스에 주석을 달아 NGINX 인그레스 컨트롤러에 내장된 상호 TLS 기능을 활성화할 수 있습니다. NGINX 컨트롤러의 상호 TLS 주석에 대한 자세한 내용은 Kubernetes 설명서의 [클라이언트 인증서 인증](#)을 참조하세요.

중요: 이 패턴은 자체 서명된 인증서를 사용합니다. 이 패턴은 테스트 클러스터에만 사용하고 프로덕션 환경에서는 사용하지 않는 것이 좋습니다. 프로덕션 환경에서 이 패턴을 사용하려는 경우, [AWS 프라이빗 인증 기관\(AWS 프라이빗 CA\)](#) 또는 기존 공개 키 인프라(PKI) 표준을 사용하여 프라이빗 인증서를 발급할 수 있습니다.

사전 조건 및 제한 사항

사전 조건

- 활성 Amazon Web Services(AWS) 계정
- 기존 Amazon EKS 클러스터.
- macOS, Linux 또는 Windows에 설치 및 구성된 AWS Command Line Interface(AWS CLI) 버전 1.7 이상.

- Amazon EKS 클러스터에 액세스하도록 설치 및 구성된 kubectl 명령줄 유틸리티. 이에 대한 자세한 내용은 Amazon EKS 설명서의 [kubectl 설치](#)를 참조하세요.
- 애플리케이션을 테스트하는 데 사용할 기존 도메인 이름 시스템(DNS) 이름.

제한 사항

- 이 패턴은 자체 서명된 인증서를 사용합니다. 이 패턴은 테스트 클러스터에만 사용하고 프로덕션 환경에서는 사용하지 않는 것이 좋습니다.

아키텍처

기술 스택

- Amazon EKS
- Amazon Route 53
- Kubectl

도구

- [Amazon Elastic Kubernetes Service\(Amazon EKS\)](#)는 자체 Kubernetes 컨트롤 플레인 또는 노드를 설치하거나 유지 관리할 필요 없이 AWS에서 Kubernetes를 실행하는 데 도움이 됩니다.
- [Amazon Route 53](#)은 가용성과 확장성이 뛰어난 DNS 웹 서비스입니다.
- [Kubectl](#)은 Amazon EKS 클러스터와 상호 작용하는 데 사용하는 명령줄 유틸리티입니다.

에픽

자체 서명된 인증서 생성

작업	설명	필요한 기술
CA 키 및 인증서를 생성합니다.	다음 명령을 실행하여 인증 기관(CA) 키와 인증서를 생성합니다.	DevOps 엔지니어

작업	설명	필요한 기술
	<pre>openssl req -x509 - sha256 -newkey rsa:4096 -keyout ca.key -out ca.crt -days 356 -nodes -subj '/CN=Test Cert Authority'</pre>	
<p>서버 키와 인증서를 생성하고 CA 인증서로 서명합니다.</p>	<p>서버 키와 인증서를 생성하고 다음 명령을 실행하여 CA 인증서로 서명합니다.</p> <pre>openssl req -new - newkey rsa:4096 - keyout server.key - out server.csr -nodes -subj '/CN= <your_dom ain_name> ' && openssl x509 -req -sha256 -days 365 -in server.csr - CA ca.crt -CAkey ca.key -set_serial 01 -out server.crt</pre> <p>중요: 반드시 기존 도메인 이름으로 <your_domain_name> 을 변경합니다.</p>	DevOps 엔지니어

작업	설명	필요한 기술
클라이언트 키와 인증서를 생성하고 CA 인증서로 서명합니다.	<p>다음 명령을 실행하여 클라이언트 키와 인증서를 생성하고 CA 인증서로 서명합니다.</p> <pre>openssl req -new -newkey rsa:4096 -keyout client.key -out client.csr -nodes -subj '/CN=Test' && openssl x509 -req -sha256 -days 365 -in client.csr -CA ca.crt -CAkey ca.key -set_serial 02 -out client.crt</pre>	DevOps 엔지니어

NGINX 인그레스 컨트롤러 배포

작업	설명	필요한 기술
Amazon EKS 클러스터에 NGINX 인그레스 컨트롤러를 배포합니다.	<p>다음 명령으로 NGINX 인그레스 컨트롤러를 배포합니다.</p> <pre>kubectl apply -f https://raw.githubusercontent.com/kubernetes/ingress-nginx/controller-v1.7.0/deploy/static/provider/aws/deploy.yaml</pre>	DevOps 엔지니어
NGINX 인그레스 컨트롤러 서비스가 실행 중인지 확인합니다.	<p>다음 명령을 사용하여 NGINX 인그레스 컨트롤러 서비스가 실행 중인지 확인합니다.</p>	DevOps 엔지니어

작업	설명	필요한 기술
	<pre>kubectl get svc -n ingress-nginx</pre> <p>중요: 서비스 주소 필드에 Network Load Balancer의 도메인 이름이 포함되어 있는지 확인합니다.</p>	

Amazon EKS 클러스터에 네임스페이스를 생성하여 상호 TLS를 테스트합니다.

작업	설명	필요한 기술
Amazon EKS 클러스터에서 네임스페이스를 생성합니다.	<p>다음 명령을 실행하여 Amazon EKS 클러스터에서 mtls로 호출되는 네임스페이스를 생성합니다.</p> <pre>kubectl create ns mtls</pre> <p>그러면 샘플 애플리케이션이 배포되어 상호 TLS를 테스트할 수 있습니다.</p>	DevOps 엔지니어

샘플 애플리케이션의 배포 및 서비스 생성

작업	설명	필요한 기술
mtls 네임스페이스에서 Kubernetes 배포 및 서비스를 생성합니다.	<p>mtls.yaml 이라는 이름의 파일을 만듭니다. 다음 코드를 파일에 붙여 넣습니다.</p> <pre>kind: Deployment apiVersion: apps/v1 metadata:</pre>	DevOps 엔지니어

작업	설명	필요한 기술
	<pre> name: mtls-app labels: app: mtls spec: replicas: 1 selector: matchLabels: app: mtls template: metadata: labels: app: mtls spec: containers: - name: mtls-app image: hashicorp /http-echo args: - "-text=mTLS is working" --- kind: Service apiVersion: v1 metadata: name: mtls-service spec: selector: app: mtls ports: - port: 5678 # Default port for image </pre> <p>다음 명령을 실행하여 mTLS 네임스페이스에서 Kubernetes 배포 및 서비스를 생성합니다.</p>	

작업	설명	필요한 기술
	<pre>kubectl create -f mtls.yaml -n mtl</pre>	
Kubernetes 배포가 생성되었는지 확인합니다.	<p>다음 명령을 실행하여 배포가 생성되었고 사용 가능한 상태인 포드가 하나 있는지 확인합니다.</p> <pre>kubectl get deploy -n mtl</pre>	DevOps 엔지니어
Kubernetes 서비스가 생성되었는지 확인합니다.	<p>다음 명령을 실행하여 Kubernetes 서비스가 생성되었는지 확인합니다.</p> <pre>kubectl get service -n mtl</pre>	DevOps 엔지니어

mtls 네임스페이스에 보안 암호 생성

작업	설명	필요한 기술
인그레스 리소스의 보안 암호를 생성합니다.	<p>다음 명령어를 실행하여 이전에 생성한 인증서를 사용하여 NGINX 인그레스 컨트롤러용 보안 암호를 생성합니다.</p> <pre>kubectl create secret generic mtl-certs --from-file=tl.cr t=server.crt --from- file=tl.key=server. key --from-file=ca.cr t=ca.crt -n mtl</pre>	DevOps 엔지니어

작업	설명	필요한 기술
	보안 암호에는 서버를 식별하는 클라이언트용 서버 인증서와 클라이언트 인증서를 확인하는 서버용 CA 인증서가 있습니다.	

mtls 네임스페이스에 인그레스 리소스를 생성합니다.

작업	설명	필요한 기술
mtls 네임스페이스에 인그레스 리소스를 생성합니다.	<p>ingress.yaml 이라는 이름의 파일을 만듭니다. 다음 코드를 파일에 붙여넣습니다(기존 도메인 이름을 <your_domain_name> 으로 대체).</p> <pre> apiVersion: networking.k8s.io/v1 kind: Ingress metadata: annotations: nginx.ingress.kubernetes.io/auth-tls-verify-client: "on" nginx.ingress.kubernetes.io/auth-tls-secret: mtls/mtls-certs name: mtl-ingress spec: ingressClassName: nginx rules: - host: ".*<your_domain_name>" http: paths: - path: / pathType: Prefix </pre>	DevOps 엔지니어

작업	설명	필요한 기술
	<pre> backend: service: name: mtls- service port: number: 6678 tls: - hosts: - ".*.<your_ domain_name>" secretName: mtl- certs </pre> <p>다음 명령을 실행하여 mtl- 네임스페이스에 인그레스 리소스를 생성합니다.</p> <pre> kubect- create -f ingress.yaml -n mtl- </pre> <p>즉, NGINX 인그레스 컨트롤러는 트래픽을 샘플 애플리케이션으로 라우팅할 수 있습니다.</p>	
<p>인그레스 리소스가 생성되었는지 확인합니다.</p>	<p>다음 명령을 실행하여 인그레스 리소스가 생성되었는지 확인합니다.</p> <pre> kubect- get ing -n mtl- </pre> <p>중요: 인그레스 리소스의 주소에 NGINX 인그레스 컨트롤러용으로 생성된 로드 밸런서가 표시되는지 확인합니다.</p>	<p>DevOps 엔지니어</p>

호스트 이름이 로드 밸런서를 가리키도록 DNS를 구성합니다.

작업	설명	필요한 기술
NGINX 인그레스 컨트롤러의 로드 밸런서를 가리키는 CNAME 레코드를 생성합니다.	<p>AWS Management Console에 로그인하고 Amazon Route 53 콘솔을 열고 <code>mtls.<your_domain_name></code> 을 NGINX 인그레스 컨트롤러의 로드 밸런서로 가리키는 정식 이름(CNAME) 레코드를 생성합니다.</p> <p>자세한 내용은 Route 53 설명서의 Route 53 콘솔을 사용하여 레코드 생성을 참조하세요.</p>	DevOps 엔지니어

애플리케이션 테스트

작업	설명	필요한 기술
인증서 없이 상호 TLS 설정을 테스트할 수 있습니다.	<p>다음 명령을 실행합니다.</p> <pre>curl -k https://mtls.<your_domain_name></pre> <p>"400 No required SSL certificate was sent"라는 오류 응답을 받아야 합니다.</p>	DevOps 엔지니어
인증서를 사용하여 상호 TLS 설정을 테스트합니다.	<p>다음 명령을 실행합니다.</p> <pre>curl -k https://mtls.<your_domain_name> --cert client.crt --key client.key</pre>	DevOps 엔지니어

작업	설명	필요한 기술
	“mTLS is working”이라는 응답을 받아야 합니다.	

관련 리소스

- [Amazon Route 53 콘솔을 사용하여 레코드 생성](#)
- [Amazon EKS에서 NGINX 인그레스 컨트롤러와 함께 Network Load Balancer 사용](#)
- [클라이언트 인증서 인증](#)

Firelens 로그 라우터를 사용하여 Amazon ECS용 사용자 지정 로그 구문 분석기를 생성

작성자: Varun Sharma(AWS)

환경: 프로덕션

기술: DevOps; 컨테이너 및 마이크로서비스

워크로드: 기타 모든 워크로드

AWS 서비스: Amazon ECS

요약

Firelens는 Amazon Elastic Container Service(Amazon ECS)와 AWS Fargate를 위한 로그 라우터입니다. Firelens를 사용하여 Amazon ECS에서 Amazon CloudWatch 및 기타 대상 (예: [스플링크](#) 또는 [Sumo Logic](#)) 으로 컨테이너 로그를 라우팅할 수 있습니다. Firelens는 [Fluentd](#) 또는 [Fluent Bit](#)를 로깅 에이전트로 사용하여 작동합니다. 즉, [Amazon ECS 작업 정의 파라미터](#)를 사용하여 로그를 라우팅할 수 있습니다.

소스 수준에서 로그를 구문 분석하기로 선택하면 로깅 데이터를 분석하고 쿼리를 수행하여 운영 문제에 더 효율적이면서 효과적으로 대응할 수 있습니다. 애플리케이션마다 로깅 패턴이 다르기 때문에 로그를 구조화하고 최종 대상에서 더 쉽게 검색할 수 있는 사용자 지정 구문 분석기를 사용해야 합니다.

이 패턴은 사용자 지정 파서가 있는 Firelens 로그 라우터를 사용하여 Amazon CloudWatch ECS에서 실행되는 샘플 Spring Boot 애플리케이션에서 로그를 푸시합니다. 그런 다음 Amazon CloudWatch Logs Insights를 사용하여 사용자 지정 파서에서 생성한 사용자 지정 필드를 기반으로 로그를 필터링할 수 있습니다.

사전 조건 및 제한 사항

사전 조건

- 활성 Amazon Web Services(AWS) 계정.
- 로컬 시스템에 설치 및 구성된 Command Line Interface(CLI).
- 로컬 시스템에 설치 및 구성된 Docker.
- Amazon Elastic Container Registry(Amazon ECR)의 기존 Spring Boot 기반 컨테이너식 애플리케이션.

아키텍처

기술 스택

- CloudWatch
- Amazon ECR
- Amazon ECS
- Fargate
- Docker
- Fluent Bit

도구

- [Amazon ECR](#) – Amazon Elastic Container Registry(Amazon ECR)는 안전하고 확장 가능하고 신뢰할 수 있는 AWS 관리형 컨테이너 이미지 레지스트리 서비스입니다.
- [Amazon ECS](#) – Amazon Elastic Container Service(Amazon ECS)는 클러스터에서 컨테이너를 손쉽게 실행, 중지 및 관리할 수 있게 하는 컨테이너 관리 서비스로서 확장성과 속도가 뛰어납니다.
- [Identity and Access Management\(IAM\)](#) – IAM은 AWS 리소스에 대한 사용자의 액세스를 안전하게 제어할 수 있게 지원하는 웹 서비스입니다.
- [CLI](#) – Command Line Interface(CLI)는 명령줄 쉘에서 명령을 사용하여 AWS 서비스와 상호 작용할 수 있는 오픈 소스 도구입니다.
- [Docker](#) — Docker는 애플리케이션 개발, 배송, 실행을 위한 개방형 플랫폼입니다.

코드

이 패턴에는 다음 파일이 연결됩니다.

- `customFluentBit.zip` — 사용자 지정 구문 분석 및 구성을 추가하기 위한 파일이 들어 있습니다.
- `firelens_policy.json` — IAM 정책을 생성하기 위한 정책 문서가 들어 있습니다.
- `Task.json` — Amazon ECS의 샘플 작업 정의가 들어 있습니다.

에픽

사용자 지정 Fluent Bit 이미지 생성

작업	설명	필요한 기술
Amazon ECR 리포지토리를 생성합니다.	<p>AWS Management Console에 로그인한 다음, Amazon ECR 콘솔을 열고 <code>fluentbit_custom</code> 라고 불리는 리포지토리를 생성합니다.</p> <p>이에 대한 자세한 내용은 Amazon ECR 설명서의 리포지토리 생성을 참조하십시오.</p>	시스템 관리자, 개발자
customFluentBit.zip 패키지의 압축을 풉니다.	<ol style="list-style-type: none"> 로컬 머신에 첨부된 <code>customFluentBit.zip</code> 패키지를 다운로드합니다. 다음 명령을 실행하여 <code>customFluentBit</code> 디렉터리로 압축을 풉니다 (<code>unzip -d customFluentBit.zip</code>). 디렉터리에는 사용자 지정 구문 분석 및 구성을 추가하는데 필요한 다음 파일이 들어 있습니다. <ul style="list-style-type: none"> <code>parsers/springboot_parser.conf</code> — 구문 분석기 지시문이 들어 있고 사용자 지정 구문 분석기의 정규식(regex) 패턴을 정의합니다. 특정 구 	

작업	설명	필요한 기술
	<p>문 분석기에 대한 regex 패턴을 추가할 수 있습니다.</p> <ul style="list-style-type: none"> • <code>conf/pars_e_springboot.conf</code> — 필터 및 서비스 지시문이 들어 있습니다. • <code>Dockerfile</code> 	
사용자 지정 도커 이미지를 생성합니다.	<ol style="list-style-type: none"> 1. 디렉터리를 <code>customFluentBit</code> 로 변경합니다. 2. Amazon ECR 콘솔을 열고 <code>fluentbit_custom</code> 리포지토리를 선택한 다음 푸시 명령 보기를 선택합니다. 3. 프로젝트를 업로드합니다. 4. 업로드가 완료된 후 빌드의 URL을 복사합니다. Amazon ECS에서 컨테이너를 생성할 때 이 URL이 필요합니다. <p>자세한 내용은 Docker 설명서에서 Docker 이미지 푸시하기를 참조하십시오.</p>	시스템 관리자, 개발자

Amazon ECS 클러스터를 설정

작업	설명	필요한 기술
Amazon ECS 클러스터를 생성합니다.	Amazon ECS 설명서에서 클러스터 생성 의 네트워킹 전용 템플릿 섹션에 나와 있는 지침을	시스템 관리자, 개발자

작업	설명	필요한 기술
	<p>따라 Amazon ECS 클러스터를 생성합니다.</p> <p>참고: Amazon ECS 클러스터용 새 Virtual Private Cloud(VPC)를 생성하기 위해 VPC 생성을 선택해야 합니다.</p>	

Amazon ECS 작업 설정

작업	설명	필요한 기술
Amazon ECS 태스크 실행 IAM 역할을 설정합니다.	<p>AmazonECSTaskExecutionRolePolicy 관리형 정책을 사용하여 Amazon ECS 태스크 실행 IAM 역할을 생성합니다. 이에 대한 자세한 내용은 Amazon ECS 사용 설명서의 Amazon ECS 태스크 실행 IAM 역할을 참조하십시오.</p> <p>참고: IAM 역할의 Amazon 리소스 이름(ARN)을 기록해 두어야 합니다.</p>	시스템 관리자, 개발자
IAM 정책을 Amazon ECS 작업 실행 IAM 역할에 연결합니다.	<ol style="list-style-type: none"> firelens_policy.json (첨부됨) 정책 문서를 사용하여 IAM 정책을 생성합니다. 자세한 내용은 IAM 설명서의 JSON 탭에서 정책 만들기를 참조하세요. 이 정책을 이전에 생성한 Amazon ECS 작업 실행 IAM 역할에 연결합니다. 이에 대한 자세한 내용은 IAM 	시스템 관리자, 개발자

작업	설명	필요한 기술
	설명서에서 IAM 정책(CLI) 추가 를 참조하십시오.	

작업	설명	필요한 기술
<p>Amazon ECS 태스크 정의를 설정합니다.</p>	<ol style="list-style-type: none"> 1. Task.json 샘플 작업 정의(첨부됨)에서 다음 섹션을 업데이트합니다. <ul style="list-style-type: none"> • 작업 실행 IAM 역할의 ARN으로 executionRoleArn 및 taskRoleArn (을)를 업데이트합니다. • 이전에 생성한 사용자 지정 Fluent Bit 도커 이미지로 container Definitions 의 이미지를 업데이트합니다. • 애플리케이션 이미지의 이름으로 container Definitions 의 이미지를 업데이트합니다. 2. Amazon ECS 콘솔을 열어 작업 정의를 선택하고 새 작업 정의 생성을 선택한 다음 기능 선택 페이지에서 Fargate를 선택합니다. 3. Json을 통한 구성을 선택하고 텍스트 영역에 업데이트된 Task.json 파일을 붙여넣은 다음 저장을 선택합니다. 4. 태스크 정의를 생성합니다. <p>이에 대한 자세한 내용은 Amazon ECS 설명서의 <u>작업 정의 생성</u>을 참조하십시오.</p>	<p>시스템 관리자, 개발자</p>

Amazon ECS 태스크 실행

작업	설명	필요한 기술
Amazon ECS 태스크를 실행합니다.	<p>Amazon ECS 콘솔에서 클러스터를 선택하고 이전에 생성한 클러스터를 선택한 다음 독립 실행형 작업을 실행합니다.</p> <p>이에 대한 자세한 내용은 Amazon ECS 설명서에서 독립 실행형 작업 실행을 참조하십시오.</p>	시스템 관리자, 개발자

로그 확인 CloudWatch

작업	설명	필요한 기술
로그를 확인합니다.	<ol style="list-style-type: none"> CloudWatch 콘솔을 열고 로그 그룹을 선택한 다음 선택합니다/aws/ecs/containerinsights/{cluster_ARN}/firehose/application 로그, 특히 사용자 지정 구문 분석기가 추가한 사용자 지정 필드를 확인합니다. 사용자 지정 필드를 기반으로 로그를 필터링하는 CloudWatch 데 사용합니다. 	시스템 관리자, 개발자

관련 리소스

- [Amazon ECS용 도커 기본 사항](#)

- [Fargate의 Amazon ECS](#)
- [기본 서비스 파라미터 구성](#)

첨부

이 문서와 관련된 추가 콘텐츠에 액세스하려면 [attachment.zip](#) 파일의 압축을 풉니다.

HashiCorp 패커를 사용하여 CodePipeline 파이프라인 및 AMI 생성

작성자: Akash Kumar(AWS)

환경: PoC 또는 파일럿	출처: DevOps	대상: Amazon Machine Images(AMI)
R 유형: 리호스팅	워크로드: 기타 모든 워크로드	기술: DevOps; 현대화, 웹 및 모바일 앱

요약

이 패턴은 AWS를 사용하여 Amazon Web Services (AWS) 클라우드에서 파이프라인을 생성하고 HashiCorp Packer를 사용하여 Amazon 머신 이미지 (AMI) 를 생성하는 코드 CodePipeline 샘플과 단계를 제공합니다. 이 패턴은 Git 기반 버전 제어 시스템으로 코드의 빌드와 테스트를 자동화하는 [지속적 통합](#) 방식을 기반으로 합니다. 이 패턴에서는 AWS를 사용하여 코드 리포지토리를 생성하고 CodeCommit 복제합니다. 그런 다음 AWS를 사용하여 프로젝트를 생성하고 소스 코드를 구성합니다 CodeBuild. 마지막으로 리포지토리에 커밋되는 AMI를 생성합니다.

사전 조건 및 제한 사항

사전 조건

- 활성 상태의 AWS 계정
- Amazon Elastic Compute Cloud(Amazon EC2) 인스턴스를 시작하기 위한 Amazon Linux AMI
- [HashiCorp 패커](#) 0.12.3 이상
- 아마존 CloudWatch 이벤트 (선택 사항)
- 아마존 CloudWatch 로그 (선택 사항)

아키텍처

다음 다이어그램은 이 패턴의 아키텍처를 사용하여 AMI 생성을 자동화하는 애플리케이션 코드의 예제를 보여줍니다.

이 다이어그램은 다음 워크플로를 보여줍니다.

1. 개발자는 비공개 CodeCommit Git 리포지토리에 코드 변경 사항을 커밋합니다. 그런 다음 CodeBuild 를 CodePipeline 사용하여 빌드를 시작하고 Amazon Simple Storage Service (Amazon S3) 버킷에 배포할 준비가 된 새 [아티팩트](#)를 추가합니다.
2. CodeBuild Packer를 사용하여 JSON 템플릿을 기반으로 AMI를 번들하고 패키징합니다. 활성화된 경우 소스 코드가 변경될 때 CloudWatch 이벤트가 자동으로 파이프라인을 시작할 수 있습니다.

기술 스택

- CodeBuild
- CodeCommit
- CodePipeline
- CloudWatch 이벤트 (선택 사항)

도구

- [AWS CodeBuild](#) — CodeBuild AWS는 클라우드의 완전 관리형 빌드 서비스입니다. CodeBuild 소스 코드를 컴파일하고, 단위 테스트를 실행하고, 배포할 준비가 된 아티팩트를 생성합니다.
- [AWS CodeCommit](#) — CodeCommit AWS는 AWS 클라우드에서 Git 리포지토리를 비공개로 저장하고 관리할 수 있는 버전 제어 서비스입니다. CodeCommit 자체 소스 제어 시스템을 관리하거나 인프라 확장에 대해 걱정할 필요가 없습니다.
- [AWS CodePipeline](#) — CodePipeline AWS는 소프트웨어 릴리스에 필요한 단계를 모델링, 시각화 및 자동화하는 데 사용할 수 있는 지속적 전송 서비스입니다.
- [HashiCorp HashiCorp Packer](#) — Packer는 단일 소스 구성에서 동일한 머신 이미지를 자동으로 생성할 수 있는 오픈 소스 도구입니다. Packer는 가볍고, 모든 주요 운영 체제에서 실행되며, 여러 플랫폼에 대한 머신 이미지를 병렬로 생성합니다.

코드

이 패턴에는 다음과 같은 첨부 파일이 포함됩니다.

- `buildspec.yml` — 이 파일은 배포용 CodeBuild 아티팩트를 빌드하고 생성하는 데 사용됩니다.
- `amazon-linux_packer-template.json` — 이 파일은 패커를 사용하여 Amazon Linux AMI를 생성합니다.

에픽

코드 리포지토리 설정

작업	설명	필요한 기술
리포지토리를 생성합니다.	CodeCommit 리포지토리를 생성합니다.	AWS 시스템 관리자
리포지토리를 복제합니다.	리포지토리를 복제하여 CodeCommit 리포지토리에 연결합니다.	앱 개발자
소스 코드를 원격 리포지토리로 푸시합니다.	<ol style="list-style-type: none"> 커밋을 생성하여 buildspec.yml 및 amazon-linux_package-template.json 파일을 로컬 리포지토리에 추가합니다. 로컬 리포지토리에서 원격 CodeCommit 리포지토리로 커밋을 푸시합니다. 	앱 개발자

애플리케이션을 위한 CodeBuild 프로젝트를 생성하세요.

작업	설명	필요한 기술
빌드 프로젝트를 생성합니다.	<ol style="list-style-type: none"> AWS Management 콘솔에 로그인하고 AWS CodeBuild 콘솔을 연 다음 빌드 프로젝트 생성을 선택합니다. 프로젝트 이름에 사용자의 프로젝트 이름을 입력합니다. 소스 공급자로 AWS를 선택합니다 CodeCommit. 	앱 개발자, AWS 시스템 관리자

작업	설명	필요한 기술
	<ol style="list-style-type: none"> 4. 리포지토리의 경우 코드 파이프라인을 구축하려는 리포지토리를 선택합니다. 5. 환경 이미지에, 관리형 이미지 또는r 사용자 지정 이미지를 선택합니다. 6. [Operating system]에서 [Ubuntu]를 선택합니다. 7. RunTime[s] 에서 [스탠다드]를 선택합니다. 8. 이미지에서 aws/codebuild/standard:4.0을 선택합니다. 9. Image version(이미지 버전)에서 Always use the latest image for this runtime version(이 실행 시간 버전에 항상 최신 이미지 사용)을 선택합니다. 10.환경에서 Linux를 선택합니다. 11.권한 확인란을 선택합니다. 12.서비스 역할에서, 새 서비스 역할 또는 기존 서비스 역할을 선택합니다. 13.Build 사양에서 buildspec 파일 사용 또는r 빌드 명령 삽입을 선택합니다. 14(선택 사항) 아티팩트 섹션에서 유형의 경우 아티팩트 없음을 선택합니다. 15(권장) 빌드 출력 로그를 Logs에 업로드하려면 	

작업	설명	필요한 기술
	<p>CloudWatch CloudWatch logs를 선택합니다.</p> <p>16.(선택 사항) 빌드 출력 로그를 Amazon S3에 업로드하려면 S3 로그 확인란을 선택합니다.</p> <p>17.빌드 프로젝트 생성을 선택합니다.</p>	

파이프라인 설정

작업	설명	필요한 기술
파이프라인 이름	<ol style="list-style-type: none"> 1. AWS Management 콘솔에 로그인하고 AWS CodePipeline 콘솔을 연 다음 파이프라인 생성을 선택합니다. 2. 파이프라인 이름에 파이프라인의 이름을 입력합니다. 3. 서비스 역할에서, 새 서비스 역할 또는 기존 서비스 역할을 선택합니다. 4. 역할 이름에 역할의 이름을 입력합니다. 5. Amazon S3에서 버킷을 생성하고 해당 버킷에 아티팩트를 저장하도록 하려면 고급 설정 섹션에서 아티팩트 스토어에 대하여 기본 위치를 선택합니다. 기존 S3 버킷을 사용하려면 사용자 지정 위치를 선택합니다. 다음을 선택합니다. 	앱 개발자, AWS 시스템 관리자

작업	설명	필요한 기술
	<p>6. 소스 공급자로 AWS를 선택합니다 CodeCommit.</p> <p>7. 리포지토리 이름의 경우 이전에 복제한 리포지토리를 선택합니다. 브랜치 이름의 경우 소스 코드 브랜치를 선택합니다.</p> <p>8. 변경 감지 옵션의 경우 Amazon CloudWatch Events (권장) 를 선택하여 파이프라인을 시작하거나 CodePipelineAWS를 선택하여 변경 사항을 정기적으로 확인합니다. 다음을 선택합니다.</p> <p>9. 빌드 공급자의 경우 AWS를 선택합니다 CodeBuild.</p> <p>10. 프로젝트 이름의 경우 애플리케이션 에픽용 CodeBuild 프로젝트 생성에서 생성한 빌드 프로젝트를 선택합니다.</p> <p>11. 빌드 옵션을 선택한 후 다음을 선택합니다.</p> <p>12. 배포 단계 건너뛰기를 선택합니다.</p> <p>13. 파이프라인 생성을 선택합니다.</p>	

관련 리소스

- [AWS의 리포지토리 사용 CodeCommit](#)

- [빌드 프로젝트 작업](#)
- [에서 파이프라인으로 작업하기 CodePipeline](#)

첨부

이 문서와 관련된 추가 콘텐츠에 액세스하려면 [attachment.zip](#) 파일의 압축을 풉니다.

파이프라인을 생성하고 다음을 사용하여 온프레미스 EC2 인스턴스에 아티팩트 업데이트를 배포합니다. CodePipeline

작성자: Akash Kumar(AWS)

환경: PoC 또는 파일럿	출처: DevOps	대상: Amazon EC2/온프레미스
R 유형: 리호스팅	기술: DevOps; 현대화, 웹 및 모바일 앱	AWS 서비스: AWS CodeBuild, AWS CodeCommit, AWS CodeDeploy, AWS CodePipeline

요약

이 패턴은 Amazon Web Services (AWS) 클라우드에서 파이프라인을 생성하고 업데이트된 [아티팩트](#)를 AWS의 온프레미스 Amazon Elastic Compute Cloud (Amazon EC2) 인스턴스에 배포하기 위한 코드 샘플과 단계를 제공합니다. CodePipeline 패턴은 [지속적 통합](#) 방식을 기반으로 합니다. 이 방식은 Git 기반 버전 제어 시스템을 사용하여 코드의 빌드 및 테스트를 자동화합니다. 이 패턴에서는 AWS를 사용하여 코드 리포지토리를 생성하고 CodeCommit 복제합니다. 그런 다음 AWS를 사용하여 프로젝트를 생성하고 소스 코드를 구성합니다 CodeBuild. 마지막으로, AWS를 사용하여 애플리케이션을 생성하고 온프레미스 EC2 인스턴스의 대상 환경을 구성합니다. CodeDeploy

사전 조건 및 제한 사항

사전 조건

- 활성 상태의 AWS 계정
- 배포 중에 EC2 인스턴스를 식별하기 위한 [사용자 정의 태그](#)
- [CodeDeploy 에이전트](#), EC2 인스턴스에 설치
- EC2 인스턴스에 설치된 필수 런타임 소프트웨어
- Java 개발 키트용 [Amazon Corretto 8](#)
- 설치되는 [Apache Tomcat](#) 웹 서버
- 아마존 CloudWatch 이벤트 (선택 사항)
- 웹 서버에 로그인하기 위한 키 페어(선택 사항)

- 웹 애플리케이션을 위한 Apache Maven 애플리케이션 프로젝트

아키텍처

다음 다이어그램은 이 패턴의 아키텍처를 사용하여 온프레미스 EC2 인스턴스에 배포된 Java 웹 애플리케이션의 예제를 보여줍니다.

이 다이어그램은 다음 워크플로를 보여줍니다.

1. 개발자는 비공개 CodeCommit Git 리포지토리에 코드 변경 사항을 커밋합니다.
2. CodePipeline CodeBuild 빌드를 시작하고 Amazon Simple Storage Service (Amazon S3) 버킷에 배포할 준비가 된 새 아티팩트를 추가하는 데 사용합니다.
3. CodePipeline CodeDeploy 에이전트를 사용하여 배포 아티팩트 변경에 필요한 모든 종속성을 사전 설치합니다.
4. CodePipeline CodeDeploy 에이전트를 사용하여 S3 버킷의 아티팩트를 대상 EC2 인스턴스로 배포합니다. 활성화된 경우 소스 코드가 변경될 때 CloudWatch 이벤트가 자동으로 파이프라인을 시작할 수 있습니다.

기술 스택

- CodeBuild
- CodeCommit
- CodeDeploy
- CodePipeline
- CloudWatch 이벤트 (선택 사항)

도구

- [CodeBuildAWS](#)는 소스 코드를 컴파일하고, 단위 테스트를 실행하고, 배포할 준비가 된 아티팩트를 생성하는 데 도움이 되는 완전관리형 빌드 서비스입니다. CodeBuild 소스 코드를 컴파일하고, 단위 테스트를 실행하고, 배포할 준비가 된 아티팩트를 생성합니다.
- [CodeCommitAWS](#)는 자체 소스 제어 시스템을 관리할 필요 없이 Git 리포지토리를 비공개로 저장하고 관리할 수 있는 버전 제어 서비스입니다.

- [AWS](#)는 아마존 엘라스틱 CodeDeploy 컴퓨트 클라우드 (Amazon EC2) 또는 온프레미스 인스턴스, AWS Lambda 함수 또는 아마존 Elastic Container Service (아마존 ECS) 서비스로의 배포를 자동화합니다.
- [AWS](#)는 소프트웨어 릴리스의 여러 단계를 신속하게 모델링 및 구성하고 소프트웨어 변경 사항을 지속적으로 릴리스하는 CodePipeline 데 필요한 단계를 자동화하도록 지원합니다.

코드

이 패턴에는 다음과 같은 첨부 파일이 포함됩니다.

- `buildspec.yml`— 이 파일은 배포용 아티팩트를 빌드하고 생성하는 데 CodeBuild 필요한 작업을 지정합니다.
- `appspec.yml`— 이 파일은 애플리케이션을 생성하고 온프레미스 EC2 인스턴스의 대상 환경을 구성하는 데 CodeDeploy 필요한 작업을 지정합니다.
- `install_dependencies.sh` — 이 파일은 Apache Tomcat 웹 서버를 위한 종속성을 설치합니다.
- `start_server.sh` — 이 파일은 Apache Tomcat 웹 서버를 시작합니다.
- `stop_server.sh` — 이 파일은 Apache Tomcat 웹 서버를 정지합니다.

에픽

코드 리포지토리 설정

작업	설명	필요한 기술
리포지토리를 생성합니다.	리포지토리 생성. CodeCommit	AWS 시스템 관리자
리포지토리를 복제합니다.	리포지토리를 복제하여 CodeCommit 리포지토리에 연결합니다.	앱 개발자
소스 코드를 원격 리포지토리로 푸시합니다.	1. 커밋을 생성하여 buildspec.yml 및 appspec.yml 파일을 로컬 리포지토리에 추가합니다.	앱 개발자

작업	설명	필요한 기술
	2. 로컬 리포지토리에서 원격 CodeCommit 리포지토리로 커밋을 푸시합니다.	

애플리케이션을 위한 CodeBuild 프로젝트를 생성하세요.

작업	설명	필요한 기술
빌드 프로젝트를 생성합니다.	<ol style="list-style-type: none"> 1. AWS Management 콘솔에 로그인하고 AWS CodeBuild 콘솔을 연 다음 빌드 프로젝트 생성을 선택합니다. 2. 프로젝트 이름에 사용자의 프로젝트 이름을 입력합니다. 3. 소스 공급자로 AWS를 선택합니다 CodeCommit. 4. 리포지토리의 경우 코드 파이프라인을 구축하려는 리포지토리를 선택합니다. 5. 환경 이미지에서, 관리형 이미지 또는 사용자 지정 이미지를 선택합니다. 6. 운영 체제에서 Amazon Linux 2를 선택합니다. 7. RunTime[s] 에서 [스탠다드]를 선택합니다. 8. 이미지의 경우 aws/codebuild/amazonlinux2-aarch64-standard:2.0을 선택합니다. 9. Image version(이미지 버전)에서 Always use the latest image for this runtime 	AWS 관리자, 앱 개발자

작업	설명	필요한 기술
	<p>version(이 실행 시간 버전에 항상 최신 이미지 사용)을 선택합니다.</p> <p>10서비스 역할에서, 새 서비스 역할 또는 기존 서비스 역할을 선택합니다.</p> <p>11Build 사양에서 buildspec 파일 사용 또는 빌드 명령 삽입을 선택합니다.</p> <p>12(선택 사항) 아티팩트를 추가하여 아티팩트 구성을 선택합니다.</p> <p>13(선택 사항) 빌드 출력 로그를 CloudWatch Amazon에 업로드하려면 CloudWatch logs를 선택합니다.</p> <p>14.빌드 프로젝트 생성을 선택합니다.</p>	

온프레미스 EC2 인스턴스를 위한 아티팩트 배포를 구성합니다.

작업	설명	필요한 기술
애플리케이션을 생성합니다.	<ol style="list-style-type: none"> AWS Management 콘솔에 로그인하고 AWS 콘솔을 연 다음 [애플리케이션 생성] 을 선택합니다. CodeDeploy 애플리케이션 이름에 애플리케이션의 이름을 입력합니다. 컴퓨팅 플랫폼에서 EC2/온프레미스를 선택합니다. 	시스템 관리자, 앱 개발자

작업	설명	필요한 기술
	<ol style="list-style-type: none"> 4. 애플리케이션 생성을 선택한 다음 배포 그룹 생성을 선택합니다. 5. 배포 그룹 이름의 경우 이름을 입력합니다. 6. 에 대한 서비스 역할을 생성합니다 CodeDeploy. 참고: 서비스 역할에는 대상 환경에 대한 CodeDeploy 액세스 권한을 부여할 권한이 있어야 합니다. 7. 서비스 역할에서 6단계에서 생성한 서비스 역할을 선택합니다. 8. 배포 유형의 경우 비즈니스 요구 사항에 따라 인 플레이스 또는 블루/그린을 선택합니다. 9. 환경 구성의 경우 비즈니스 요구 사항에 적합한 옵션을 선택합니다. 10(선택 사항) Amazon EC2 콘솔에서 로드 밸런서의 대상 그룹을 별도로 생성한 다음, CodeDeploy AWS 콘솔의 배포 그룹 생성 페이지로 돌아가서 로드 밸런서와 대상 그룹을 선택합니다. 11[Create deployment group]을 선택합니다. 	

파이프라인 설정

작업	설명	필요한 기술
파이프라인을 생성합니다.	<ol style="list-style-type: none"> 1. AWS Management 콘솔에 로그인하고 AWS CodePipeline 콘솔을 연 다음 파이프라인 생성을 선택합니다. 2. 파이프라인 이름에 파이프라인의 이름을 입력합니다. 3. 서비스 역할에서, 새 서비스 역할 또는 기존 서비스 역할을 선택합니다. 4. 역할 이름에 역할의 이름을 입력합니다. 5. Amazon S3에서 버킷을 생성하고 해당 버킷에 아티팩트를 저장하도록 하려면 고급 설정 섹션에서 아티팩트 스토어에 대하여 기본 위치를 선택합니다. 기존 S3 버킷을 사용하려면 사용자 지정 위치를 선택합니다. 다음을 선택합니다. 6. 소스 공급자로 AWS를 선택합니다 CodeCommit. 7. 리포지토리 이름의 경우 이전에 복제한 리포지토리를 선택합니다. 브랜치 이름의 경우 소스 코드 브랜치를 선택합니다. 8. 변경 감지 옵션의 경우 Amazon CloudWatch Events (권장) 또는 AWS를 	시스템 관리자, 앱 개발자

작업	설명	필요한 기술
	<p>선택합니다 CodePipeline. 다음을 선택합니다.</p> <p>9. 빌드 공급자의 경우 AWS를 선택합니다 CodeBuild.</p> <p>10. 프로젝트 이름의 경우 이 패턴의 애플리케이션용 CodeBuild 프로젝트 생성 섹 션에서 생성한 빌드 프로젝 트를 선택합니다.</p> <p>11. 빌드 옵션을 선택한 후 다 음을 선택합니다.</p> <p>12. 배포 공급자의 경우 AWS를 선택합니다 CodeDeploy.</p> <p>13. 애플리케이션 이름과 배포 그룹을 선택한 후 다음을 선 택합니다.</p> <p>14. [파이프라인 생성]을 선택합 니다.</p>	

관련 리소스

- [AWS의 리포지토리 사용 CodeCommit](#)
- [빌드 프로젝트 작업](#)
- [애플리케이션으로 작업하기 CodeDeploy](#)
- [에서 파이프라인으로 작업하기 CodePipeline](#)

첨부

이 문서와 관련된 추가 콘텐츠에 액세스하려면 [attachment.zip](#) 파일의 압축을 풉니다.

Java 및 Python 프로젝트를 위한 동적 CI 파이프라인을 자동으로 생성

작성자: Aromal Raj Jayarajan(AWS), Amarnath Reddy(AWS), MAHESH RAGHUNANDANAN(AWS), Vijesh Vijayakumaran Nair(AWS)

코드 리포지토리: automated-ci-pipeline-creation	환경: PoC 또는 파일럿	기술: DevOps; 인프라, 서버리스, 클라우드 네이티브
워크로드: 기타 모든 워크로드	AWS 서비스: AWS, AWS CodeBuild, AWS Lambda CodePipeline, AWS Step Functions, AWS CodeCommit	

요약

이 패턴은 AWS 개발자 도구를 사용하여 Java 및 Python 프로젝트를 위한 동적 지속적 통합(CI) 파이프라인을 자동으로 생성하는 법을 보여줍니다.

기술 스택이 다양해지고 개발 활동이 증가하면 조직 전체에서 일관된 CI 파이프라인을 만들고 유지 관리하기가 어려워질 수 있습니다. AWS Step Functions에서 프로세스를 자동화하면 CI 파이프라인의 사용 및 접근 방식이 일관되도록 할 수 있습니다.

동적 CI 파이프라인 생성을 자동화하기 위해 이 패턴은 다음과 같은 변수 입력을 사용합니다.

- 프로그래밍 언어(Java 또는 Python만 해당)
- 파이프라인 이름
- 필수 파이프라인 단계

참고: Step Functions는 여러 AWS 서비스를 사용하여 파이프라인 생성을 오케스트레이션합니다. 이 솔루션에서 사용되는 AWS 서비스에 대한 자세한 내용은 이 패턴의 도구 섹션을 참조하십시오.

사전 조건 및 제한 사항

사전 조건

- 활성 상태의 AWS 계정
- 이 솔루션이 배포되는 동일한 AWS 리전에 있는 Amazon S3 버킷
- 이 솔루션에 필요한 리소스를 생성하는 데 필요한 AWS CloudFormation 권한을 보유한 AWS ID 및 액세스 관리 (IAM) 보안 [주체](#)

제한 사항

- 이 패턴은 Java 및 Python 프로젝트만 지원합니다.
- 이 패턴에 프로비저닝된 IAM 역할은 최소 권한 원칙을 따릅니다. IAM 역할의 권한은 CI 파이프라인이 생성해야 하는 특정 리소스를 기반으로 업데이트되어야 합니다.

아키텍처

대상 기술 스택

- AWS CloudFormation
- AWS CodeBuild
- AWS CodeCommit
- AWS CodePipeline
- IAM
- Amazon Simple Storage Service(S3)
- AWS Systems Manager
- AWS Step Functions
- AWS Lambda
- Amazon DynamoDB

대상 아키텍처

다음 다이어그램은 AWS 개발자 도구를 사용하여 Java 및 Python 프로젝트용 동적 CI 파이프라인을 자동으로 생성하는 예제 워크플로를 보여줍니다.

이 다이어그램은 다음 워크플로를 보여줍니다.

1. AWS 사용자는 CI 파이프라인 생성을 위한 입력 파라미터를 JSON 형식으로 제공합니다. 이 입력은 AWS 개발자 도구를 사용하여 CI 파이프라인을 생성하는 Step Functions 워크플로(상태 머신)를 시작합니다.
2. Lambda 함수는 Amazon S3 버킷에 저장된 input-reference라는 이름의 폴더를 읽은 다음 buildspec.yml 파일을 생성합니다. 이렇게 생성된 파일은 CI 파이프라인 단계를 정의하며 파라미터 참조를 저장하는 동일한 Amazon S3 버킷에 다시 저장됩니다.
3. Step Functions는 CI 파이프라인 생성 워크플로의 종속성에 변경 사항이 있는지 확인하고 필요에 따라 종속성 스택을 업데이트합니다.
4. Step Functions는 CodeCommit 리포지토리, CodeBuild 프로젝트, 파이프라인을 비롯한 CI CodePipeline 파이프라인 리소스를 CloudFormation 스택에 생성합니다.
5. 스택은 선택한 기술 CloudFormation 스택 (Java 또는 Python) 의 샘플 소스 코드와 buildspec.yml 파일을 저장소에 복사합니다. CodeCommit
6. CI 파이프라인 런타임 세부 정보는 DynamoDB 테이블에 저장됩니다.

자동화 및 규모 조정

- 이 패턴은 단일 개발 환경에서만 사용할 수 있습니다. 여러 개발 환경에서 사용하려면 구성을 변경해야 합니다.
- 두 개 이상의 CloudFormation 스택에 대한 지원을 추가하려면 추가 템플릿을 생성할 수 있습니다. CloudFormation 자세한 내용은 CloudFormation 설명서의 [AWS CloudFormation 시작하기](#)를 참조하십시오.

도구

도구

- [AWS Step Functions](#)은 Lambda 함수와 기타 AWS 서비스를 결합할 수 있는 서버리스 오케스트레이션 서비스로, 비즈니스 크리티컬 애플리케이션을 구축합니다.
- [AWS Lambda](#)는 서버를 프로비저닝하거나 관리할 필요 없이 코드를 실행할 수 있도록 도와주는 컴퓨팅 서비스입니다. 필요할 때만 코드를 실행하며 자동으로 확장이 가능하므로 사용한 컴퓨팅 시간 만큼만 비용을 지불합니다.
- [CodeBuildAWS](#)는 소스 코드를 컴파일하고, 단위 테스트를 실행하고, 배포할 준비가 된 아티팩트를 생성하는 데 도움이 되는 완전관리형 빌드 서비스입니다.
- [CodeCommitAWS](#)는 자체 소스 제어 시스템을 관리할 필요 없이 Git 리포지토리를 비공개로 저장하고 관리할 수 있는 버전 제어 서비스입니다.

- [AWS](#)는 소프트웨어 릴리스의 여러 단계를 신속하게 모델링 및 구성하고 소프트웨어 변경 사항을 지속적으로 릴리스하는 CodePipeline 데 필요한 단계를 자동화하도록 지원합니다.
- [AWS Identity and Access Management\(IAM\)](#)는 누구에게 인증 및 사용 권한이 있는지 제어하여 AWS 리소스에 대한 액세스를 안전하게 관리할 수 있도록 도와줍니다.
- [AWS Key Management Service\(AWS KMS\)](#)를 사용하면 암호화 키를 생성하고 제어하여 데이터를 보호할 수 있습니다.
- [Amazon Simple Storage Service\(S3\)](#)는 원하는 양의 데이터를 저장, 보호 및 검색하는 데 도움이 되는 클라우드 기반 객체 스토리지 서비스입니다.
- [AWS는 AWS CloudFormation](#) 리소스를 설정하고, 빠르고 일관되게 프로비저닝하고, AWS 계정 및 지역 전반의 수명 주기 전반에 걸쳐 리소스를 관리할 수 있도록 지원합니다.
- [Amazon DynamoDB](#)는 빠르고 예측 가능하고 확장 가능한 성능을 제공하는 완전 관리형 NoSQL 데이터베이스 서비스입니다.
- [AWS Systems Manager Parameter Store](#)는 구성 데이터 관리 및 암호 관리를 위한 안전한 계층적 스토리지를 제공합니다.

코드

이 패턴의 코드는 GitHub [automated-ci-pipeline-creation](#) 리포지토리에서 사용할 수 있습니다. 리포지토리에는 이 패턴에 설명된 대상 아키텍처를 만드는 데 필요한 CloudFormation 템플릿이 들어 있습니다.

모범 사례

- 토큰이나 암호와 같은 자격 증명 (암호) 을 CloudFormation 템플릿 또는 Step Functions 작업 구성에 직접 입력하지 마십시오. 그러면 DynamoDB 로그에 정보가 표시됩니다. 대신 AWS Secrets Manager를 사용하여 보안 정보를 설정하고 저장하십시오. 그런 다음 필요에 따라 CloudFormation 템플릿 및 Step Functions 작업 구성 내에서 Secrets Manager에 저장된 암호를 참조합니다. 자세한 내용은 AWS Secrets Manager 사용 설명서의 [AWS Secrets Manager란 무엇입니까?](#)를 참조하십시오.
- Amazon S3에 저장된 CodePipeline 아티팩트에 대해 서버 측 암호화를 구성합니다. 자세한 내용은 설명서의 [Amazon S3에 저장된 아티팩트에 대한 서버 측 암호화 구성](#)을 참조하십시오.
CodePipeline CodePipeline
- IAM 역할을 구성할 때 최소 권한을 적용합니다. 자세한 내용은 IAM 설명서의 [최소 권한 적용](#)을 참조하십시오.

- Amazon S3 버킷에 공개적으로 액세스할 수 없어야 합니다. 자세한 내용은 Amazon S3 설명서의 [S3 버킷에 대한 퍼블릭 액세스 차단 설정 구성](#)을 참조하십시오.
- Amazon S3 버킷에 대한 버전 관리를 활성화해야 합니다. 자세한 내용은 Amazon S3 버킷 설명서의 [S3 버킷에서 버전 관리 사용](#)을 참조하십시오.
- IAM 정책을 구성할 때는 IAM Access Analyzer를 사용하십시오. 이 도구는 안전하고 기능적인 IAM 정책을 작성하는 데 도움이 되는 실행 가능한 권장 사항을 제공합니다. 자세한 내용은 IAM 설명서의 [AWS Identity 및 Access Management Access Analyzer 사용](#)을 참조하십시오.
- 가능하면 IAM 정책을 구성할 때 특정 액세스 조건을 정의하십시오.
- 모니터링 및 감사 CloudWatch 목적으로 Amazon 로깅을 활성화합니다. 자세한 내용은 [Amazon CloudWatch Logs란 무엇입니까?](#)를 참조하십시오. CloudWatch 설명서에서

에픽

필수 구성 요소 구성

작업	설명	필요한 기술
Amazon S3 버킷을 생성합니다.	<p>Amazon S3 버킷을 생성 (또는 기존 버킷 사용) 하여 솔루션에 필요한 CloudFormation 템플릿, 소스 코드 및 입력 파일을 저장합니다.</p> <p>자세한 내용은 Amazon S3 설명서의 1단계: 첫 S3 버킷 생성을 참조하십시오.</p> <p>참고: Amazon S3 버킷은 솔루션을 배포하는 것과 동일한 AWS 리전에 있어야 합니다.</p>	AWS DevOps
GitHub 리포지토리를 복제합니다.	터미널 창에서 다음 명령을 실행하여 GitHub automated-ci-pipeline-creation 리포지토리를 복제합니다.	AWS DevOps

작업	설명	필요한 기술
<p>복제된 GitHub 리포지토리의 솔루션 템플릿 폴더를 Amazon S3 버킷으로 업로드합니다.</p>	<pre data-bbox="594 226 1024 407">git clone https://github.com/aws-samples/automated-ci-pipeline-creation.git</pre> <p data-bbox="594 443 1003 573">자세한 내용은 GitHub 설명서의 리포지토리 복제를 참조하십시오.</p> <p data-bbox="594 842 1003 972">자세한 내용은 Amazon S3 설명서의 객체 업로드를 참조하십시오.</p> <p data-bbox="594 1018 1003 1241">참고: 솔루션-템플릿 폴더의 콘텐츠만 업로드해야 합니다. Amazon S3 버킷의 루트 수준에서만 파일을 업로드할 수 있습니다.</p>	<p>AWS DevOps</p>

솔루션 배포

작업	설명	필요한 기술
<p>복제된 CloudFormation 리포지토리의 template.yml 파일을 사용하여 솔루션을 배포할 스택을 생성합니다. GitHub</p>	<ol data-bbox="594 1539 1024 1871" style="list-style-type: none"> 1. AWS 관리 콘솔에 로그인한 다음 AWS CloudFormation 콘솔을 엽니다. 2. 스택 생성을 선택합니다. 드롭다운 목록이 나타납니다. 3. 드롭다운 목록에서 새 리소스 포함(표준)을 선택합니다. 	<p>AWS 관리자, AWS DevOps</p>

작업	설명	필요한 기술
	<p>스택 생성 페이지가 열립니다.</p> <ol style="list-style-type: none"> 4. 템플릿 지정 섹션에서 템플릿 파일 업로드를 선택합니다. 5. 파일 선택을 선택합니다. 그런 다음 복제된 GitHub 리포지토리의 루트 폴더로 이동하여 <code>template.yml</code> 파일을 선택합니다. 그런 다음 열기를 선택합니다. 6. 다음을 선택합니다. 스택 세부 정보 지정 페이지가 열립니다. 7. 파라미터 섹션에서 다음 파라미터를 지정합니다. <ul style="list-style-type: none"> • S3의 경우 <code>TemplateBucketName</code>, 이전에 생성한 Amazon S3 버킷의 이름을 입력합니다. 이 버킷에는 이 솔루션의 소스 코드와 참조가 들어 있습니다. 버킷 이름 파라미터가 소문자인지 확인합니다. • DynamoDBTable의 경우 스택이 생성하는 DynamoDB 테이블의 이름을 입력합니다. CloudFormation • 의 <code>StateMachineName</code> 경우 CloudFormation 스택이 생성하는 Step 	

작업	설명	필요한 기술
	<p>Functions 상태 머신의 이름을 입력합니다.</p> <p>8. 다음을 선택합니다. 스택 옵션 구성 페이지가 열립니다.</p> <p>9. 스택 옵션 구성 페이지에서 다음을 선택합니다. 어떤 기본값도 변경하지 마십시오. 검토 페이지가 열립니다.</p> <p>10. 스택 생성 설정을 검토합니다. 그런 다음 스택 생성을 선택하여 스택을 시작합니다.</p> <p>참고: 스택이 생성되는 동안 스택 페이지에 CREATE_IN_PROGRESS 상태로 나열됩니다. 이 패턴의 나머지 단계를 완료하기 전에 스택 상태가 CREATE_COMPLETE로 변경될 때까지 기다려야 합니다.</p>	

설정 테스트

작업	설명	필요한 기술
<p>생성한 Step Function을 실행합니다.</p>	<ol style="list-style-type: none"> 1. AWS Management Console에 로그인하고 Step Functions 콘솔을 엽니다. 2. 생성한 Step Function을 엽니다. 3. 실행 시작을 선택합니다. 그런 다음 워크플로의 입력 값 	<p>AWS 관리자, AWS DevOps</p>

작업	설명	필요한 기술
	<p>을 JSON 형식으로 입력합니다(다음 예제 입력 참조).</p> <p>4. 실행 시작을 선택합니다.</p> <p>JSON 형식</p> <pre data-bbox="597 499 1026 1692"> { "details": { "tech_stack": "Name of the Tech Stack (python/java)", "project_name": "Name of the Project that you want to create with", "pre_build": "Choose the step if it required in the buildspec.yml file i.e., yes/no", "build": "Choose the step if it required in the buildspec.yml file i.e., yes/no", "post_build": "Choose the step if it required in the buildspec.yml file i.e., yes/no", "reports": "Choose the step if it required in the buildspec.yml file i.e., yes/no", } } </pre> <p>Java JSON 입력 예제</p> <pre data-bbox="597 1801 1026 1856"> { </pre>	

작업	설명	필요한 기술
	<pre data-bbox="609 210 1015 703"> "details": { "tech_stack": "java", "project_name": "pipeline-java-pjt", "pre_build": "yes", "build": "yes", "post_build": "yes", "reports": "yes" } } </pre> <p data-bbox="592 735 933 777">Python JSON 입력 예제</p> <pre data-bbox="609 808 1015 1407"> { "details": { "tech_stack": "python", "project_name": "pipeline-python-pjt", "pre_build": "yes", "build": "yes", "post_build": "yes", "reports": "yes" } } </pre>	

작업	설명	필요한 기술
<p>CI 파이프라인의 CodeCommit 리포지토리가 생성되었는지 확인합니다.</p>	<ol style="list-style-type: none"> 1. AWS 관리 콘솔에 로그인한 다음 CodeCommit 콘솔을 엽니다. 2. 리포지토리 페이지에서 생성한 리포지토리 이름이 CodeCommit 리포지토리 목록에 나타나는지 확인합니다. 리포지토리 이름은 다음과 같이 추가됩니다. -Repo pipeline-java-pjt 3. CodeCommit 리포지토리를 열고 buildspec.yml 파일과 함께 샘플 소스 코드가 메인 브랜치에 푸시되는지 확인합니다. 	<p>AWS DevOps</p>

작업	설명	필요한 기술
CodeBuild 프로젝트 리소스를 확인하십시오.	<ol style="list-style-type: none"> 1. AWS 관리 콘솔에 로그인한 다음 CodeBuild 콘솔을 엽니다. 2. 빌드 프로젝트 페이지에서 생성한 CodeBuild 프로젝트 이름이 프로젝트 목록에 나타나는지 확인합니다. 프로젝트 이름은 다음과 pipeline-java-pjt 같이 추가됩니다. -Build 3. 프로젝트 이름을 선택하여 CodeBuild 프로젝트를 엽니다. 그런 다음 다음 구성을 검토하고 검증하십시오. <ul style="list-style-type: none"> • 프로젝트 구성 • 소스 • 환경 • BuildSpec • 배치 구성 • 아티팩트 	AWS DevOps

작업	설명	필요한 기술
CodePipeline 스테이지 검증.	<ol style="list-style-type: none"> 1. AWS 관리 콘솔에 로그인한 다음 CodePipeline 콘솔을 엽니다. 2. 파이프라인 페이지에서, 생성한 파이프라인 이름이 파이프라인 목록에 나타나는지 확인합니다. 파이프라인 이름은 다음과 pipeline-java-pjt 같이 추가됩니다. - Pipeline 3. 파이프라인 이름을 선택하여 파이프라인을 엽니다. 그런 다음 커밋 및 배포를 포함하여 파이프라인의 각 단계를 검토하고 검증하십시오. 	AWS DevOps
CI 파이프라인이 성공적으로 실행되었는지 확인합니다.	<ol style="list-style-type: none"> 1. CodePipeline 콘솔의 Pipelines 페이지에서 파이프라인 이름을 선택하여 파이프라인 상태를 확인합니다. 2. 파이프라인의 각 단계가 성공 상태인지 확인하십시오. 	AWS DevOps

리소스 정리

작업	설명	필요한 기술
에서 리소스 스택을 삭제합니다 CloudFormation.	<p>에서 CI 파이프라인의 리소스 스택을 삭제합니다 CloudFormation.</p> <p>자세한 내용은 CloudFormation 설명서의 AWS</p>	AWS DevOps

작업	설명	필요한 기술
	<p>CloudFormation 콘솔에서 스택 삭제를 참조하십시오.</p> <p>참고: <project_name>-stack이라는 이름의 스택을 삭제해야 합니다.</p>	
<p>Amazon S3에서 CI 파이프라인의 종속성을 삭제하고 CloudFormation</p>	<ol style="list-style-type: none"> 이름이 지정된 Amazon S3 버킷을 비웁니다 DeploymentArtifactBucket. 자세한 내용은 Amazon S3 설명서의 버킷 비우기를 참조하십시오. 에서 CI 파이프라인의 종속성 스택을 삭제합니다. CloudFormation 자세한 내용은 CloudFormation 설명서의 AWS CloudFormation 콘솔에서 스택 삭제를 참조하십시오. <p>참고: 이름이 지정된 스택을 삭제해야 pipeline-creation-dependencies-stack합니다.</p>	<p>AWS DevOps</p>
<p>Amazon S3 템플릿 버킷을 삭제하십시오.</p>	<p>이 솔루션의 템플릿을 저장하는 해당 패턴의 사전 조건 구성 섹션에서 생성한 Amazon S3 버킷을 삭제합니다.</p> <p>자세한 내용은 Amazon S3 설명서의 버킷 삭제를 참조하십시오.</p>	<p>AWS DevOps</p>

관련 리소스

- [Lambda를 사용하는 Step Functions 상태 머신 생성](#) (AWS Step Functions 설명서)
- [AWS Step Functions WorkFlow 스튜디오](#) (AWS Step Functions 설명서)
- [DevOps 및 AWS](#)
- [AWS는 어떻게 CloudFormation 작동합니까?](#) (AWS CloudFormation 설명서)
- AWS, AWS [CodeDeploy](#), AWS 및 [CodeCommit CodeBuild AWS가 포함된 완전한 CI/CD](#) ([CodePipelineAWS](#) 블로그 게시물)
- [IAM 및 AWS STS 할당량, 이름 요구 사항 및 문자 제한](#) (IAM 설명서)

테라폼을 사용하여 CloudWatch Synthetics 카나리아를 배포하세요

작성자: Dhruvajyoti Mukherjee(AWS) 및 Jean-Francois Landreau(AWS)

코드 리포지토리: Terraform으로 [CloudWatch Synthetics 카나리아 배포하기](#)

환경: 프로덕션

기술: DevOps; 비즈니스 생산성; 소프트웨어 개발 및 테스트; 인프라; 웹 및 모바일 앱

AWS 서비스: 아마존 CloudWatch, 아마존 S3, 아마존 SNS, 아마존 VPC, AWS ID 및 액세스 관리

요약

고객 관점에서 시스템 상태를 검증하고 고객이 연결할 수 있는지 확인하는 것이 중요합니다. 고객이 엔드포인트에 계속 직접 호출하지 않을 때는 더 어려워집니다. [Amazon CloudWatch Synthetics](#)는 퍼블릭 엔드포인트와 프라이빗 엔드포인트를 모두 테스트할 수 있는 카나리아 생성을 지원합니다. Canary를 사용하면 사용 중이 아닌 경우에도 시스템 상태를 알 수 있습니다. 이 canary는 Node.js Puppeteer 스크립트 또는 Python Selenium 스크립트입니다.

이 패턴은 HashiCorp Terraform을 사용하여 프라이빗 엔드포인트를 테스트하는 카나리아를 배포하는 방법을 설명합니다. 여기에는 URL이 200-OK를 반환되는지 여부를 테스트하는 Puppeteer 스크립트가 포함되어 있습니다. 그런 다음 Terraform 스크립트를 프라이빗 엔드포인트를 배포하는 스크립트와 통합할 수 있습니다. 또한 퍼블릭 엔드포인트를 모니터링하도록 솔루션을 수정할 수 있습니다.

사전 조건 및 제한 사항

사전 조건

- Virtual Private Cloud(VPC)와 프라이빗 서브넷이 있는 활성 Amazon Web Services(AWS) 계정
- 프라이빗 서브넷에서 연결할 수 있는 엔드포인트의 URL
- 배포 환경에 Terraform 설치

제한 사항

현재 솔루션은 다음 CloudWatch Synthetics 런타임 버전에서 작동합니다.

- syn-nodejs-puppeteer-3.4
- syn-nodejs-puppeteer-3.5
- syn-nodejs-puppeteer-3.6
- syn-nodejs-puppeteer-3.7

새 런타임 버전이 출시되면 현재 솔루션을 업데이트해야 할 수 있습니다. 또한 보안 업데이트에 뒤처지지 않도록 솔루션을 수정해야 합니다.

제품 버전

- Terraform 1.3.0

아키텍처

아마존 CloudWatch 신세틱스는 람다와 아마존 CloudWatch 심플 스토리지 서비스 (Amazon S3) 를 기반으로 합니다. CloudWatch Amazon은 카나리아를 생성하는 마법사와 카나리아 실행 상태를 표시하는 대시보드를 제공합니다. Lambda 함수는 스크립트를 실행합니다. Amazon S3는 canary 실행의 로그와 스크린샷을 저장합니다.

이 패턴은 대상 서브넷에 배포된 Amazon Elastic Compute Cloud(Amazon EC2) 인스턴스를 통해 프라이빗 엔드포인트를 시뮬레이션합니다. Lambda 함수를 사용하려면 프라이빗 엔드포인트가 배포된 VPC에 탄력적 네트워크 인터페이스가 필요합니다.

이 다이어그램은 다음을 보여 줍니다.

1. Synthetics canary는 canary Lambda 함수를 시작합니다.
2. Canary Lambda 함수는 탄력적 네트워크 인터페이스에 연결됩니다.
3. Canary Lambda 함수는 엔드포인트의 상태를 모니터링합니다.
4. Synthetics 카나리아는 실행 데이터를 S3 버킷 및 지표로 푸시합니다. CloudWatch
5. 지표에 따라 CloudWatch 경보가 시작됩니다.
6. CloudWatch 경보는 아마존 심플 알림 서비스 (Amazon SNS) 주제를 시작합니다.

도구

서비스

- [Amazon](#)은 AWS 리소스의 지표와 AWS에서 실행하는 애플리케이션을 실시간으로 모니터링할 수 있도록 CloudWatch 도와줍니다.
- [AWS Lambda](#)는 서버를 프로비저닝하거나 관리할 필요 없이 코드를 실행하는 데 도움이 되는 컴퓨팅 서비스입니다. 필요할 때만 코드를 실행하며 자동으로 확장이 가능하므로 사용한 컴퓨팅 시간만큼만 비용을 지불합니다.
- [Amazon Simple Notification Service\(Amazon SNS\)](#)를 사용하면 웹 서버 및 이메일 주소를 포함하여 게시자와 클라이언트 간의 메시지 교환을 조정하고 관리할 수 있습니다.
- [Amazon Simple Storage Service\(S3\)](#)는 원하는 양의 데이터를 저장, 보호 및 검색하는 데 도움이 되는 클라우드 기반 객체 스토리지 서비스입니다.
- [Amazon Virtual Private Cloud\(VPC\)](#)를 이용하면 사용자가 정의한 가상 네트워크로 AWS 리소스를 시작할 수 있습니다. 이 가상 네트워크는 사용자의 자체 데이터 센터에서 운영하는 기존 네트워크와 유사하며 AWS의 확장 가능한 인프라를 사용한다는 이점이 있습니다. 이 패턴은 VPC 엔드포인트와 탄력적 네트워크 인터페이스를 사용합니다.

기타 서비스

- [HashiCorp Terraform](#)은 코드를 사용하여 클라우드 인프라 및 리소스를 프로비저닝하고 관리할 수 있도록 지원하는 오픈 소스 코드형 인프라 (IaC) 도구입니다. 이 패턴은 Terraform을 사용하여 인프라를 배포합니다.
- [Puppeteer](#)는 Node.js 라이브러리입니다. CloudWatch Synthetics 런타임은 Puppeteer 프레임워크를 사용합니다.

코드

[솔루션은 클라우드 리포지토리에서 사용할 수 있습니다. \[GitHub watch-synthetics-canary-terraform\]\(#\) 자세한 내용은 추가 정보 섹션을 참조하세요.](#)

에픽

프라이빗 URL 모니터링 솔루션 구현

작업	설명	필요한 기술
프라이빗 URL 모니터링을 위한 요구 사항을 수집하십시오.	전체 URL 정의(도메인, 파라미터, 헤더)를 수집하십시오. Amazon S3 및 Amazon과 비	클라우드 아키텍트, 네트워크 관리자

작업	설명	필요한 기술
	<p>공개로 통신하려면 VPC 엔드포인트를 사용하십시오.</p> <p>CloudWatch 엔드포인트에서 VPC와 서브넷에 어떻게 액세스할 수 있는지 확인하십시오.</p> <p>Canary 실행 빈도를 생각해 보십시오.</p>	
<p>기존 솔루션을 수정하여 프라이빗 URL을 모니터링하십시오.</p>	<p>terraform.tfvars 파일을 수정합니다.</p> <ul style="list-style-type: none"> • name - canary 이름. • runtime_version - canary의 런타임 버전. -3.7을 사용하는 것이 좋습니다. syn-nodejs-puppeteer • take_screenshot - 스크린샷을 찍어야 하는지 여부. • api_hostname - 모니터링되는 엔드포인트의 호스트 이름. • api_path - 모니터링되는 엔드포인트의 경로. • vpc_id - Canary Lambda 함수에서 사용하는 VPC ID. • subnet_ids - Canary Lambda 함수에서 사용하는 서브넷 ID. • frequency - Canary의 실행 빈도(분). • alert_sns_topic — CloudWatch 알람 알림이 전송되는 SNS 주제. 	<p>클라우드 아키텍트</p>

작업	설명	필요한 기술
솔루션 배포 및 운영	<p>솔루션을 배포하려면 다음을 수행합니다.</p> <ol style="list-style-type: none"> 개발 환경의 <code>cloudwatch-synthetics-canary-terraform</code> 디렉터리에서 Terraform을 초기화합니다. <pre>terraform init</pre> 변경 사항을 계획하고 검토하십시오. <pre>terraform plan</pre> 솔루션을 배포합니다. <pre>terraform apply</pre> 	클라우드 아키텍트, DevOps 엔지니어

문제 해결

문제	Solution
프로비저닝된 리소스 삭제가 중단됩니다.	Canary Lambda 함수, 해당하는 탄력적 네트워크 인터페이스 및 보안 그룹을 순서대로 수동으로 삭제합니다.

관련 리소스

- [Synthetics 모니터링 사용](#)
- [Amazon CloudWatch Synthetics를 사용하여 API Gateway 엔드포인트를 모니터링하세요 \(블로그 게시물\)](#)

추가 정보

리포지토리 아티팩트

리포지토리 아티팩트의 구조는 다음과 같습니다.

```
.  
### README.md  
### main.tf  
### modules  
#   ### canary  
#   ### canary-infra  
### terraform.tfvars  
### tf.plan  
### variable.tf
```

main.tf 파일은 코어 모듈을 포함하며 두 개의 하위 모듈을 배포합니다.

- canary-infra는 canary에 필요한 인프라를 배포합니다.
- canary는 canary를 배포합니다.

솔루션의 입력 파라미터는 terraform.tfvars 파일에 있습니다. 다음 코드 예제를 사용하여 canary 하나를 생성할 수 있습니다.

```
module "canary" {  
    source = "./modules/canary"  
    name   = var.name  
    runtime_version = var.runtime_version  
    take_screenshot = var.take_screenshot  
    api_hostname = var.api_hostname  
    api_path = var.api_path  
    reports-bucket = module.canary_infra.reports-bucket  
    role = module.canary_infra.role  
    security_group_id = module.canary_infra.security_group_id  
    subnet_ids = var.subnet_ids  
    frequency = var.frequency  
    alert_sns_topic = var.alert_sns_topic  
}
```

해당 var 파일은 다음과 같습니다.

```
name      = "my-canary"
runtime_version = "syn-nodejs-puppeteer-3.7"
take_screenshot = false
api_hostname = "mydomain.internal"
api_path = "/path?param=value"
vpc_id = "vpc_id"
subnet_ids = ["subnet_id1"]
frequency = 5
alert_sns_topic = "arn:aws:sns:eu-central-1:111111111111:yyyyy"
```

솔루션 정리하기

개발 환경에서 테스트하는 경우 솔루션을 정리하여 비용이 발생하지 않도록 할 수 있습니다.

1. AWS Management Console에서 Amazon S3 콘솔로 이동합니다. 솔루션에서 생성한 Amazon S3 버킷을 비웁니다. 필요한 경우 데이터를 백업해야 합니다.
2. 개발 환경의 `cloudwatch-synthetics-canary-terraform` 디렉터리에서 `destroy` 명령을 실행합니다.

```
terraform destroy
```

Amazon ECS에 Java 마이크로서비스를 위한 CI/CD 파이프라인 배포

작성자: 비제이 톰슨(AWS)과 산카르 산구보틀라(AWS)

환경: PoC 또는 파일럿

기술: DevOps; 컨테이너 및 마이크로서비스

AWS 서비스: AWS CodeBuild, 아마존 EC2 컨테이너 레지스트리, 아마존 ECS, AWS Fargate, AWS CodePipeline

요약

이 패턴은 AWS를 사용하여 기존 Amazon Elastic Container Service (Amazon ECS) 클러스터에 Java 마이크로서비스를 위한 지속적 통합 및 지속적 전달 (CI/CD) 파이프라인을 배포하는 단계를 안내합니다. CodeBuild 개발자가 변경 사항을 적용하면 CI/CD 파이프라인이 시작되고 빌드 프로세스가 시작됩니다. CodeBuild 빌드가 완료되면 아티팩트가 Amazon Elastic Container Registry(Amazon ECR)로 푸시되고 Amazon ECR의 최신 빌드가 픽업되어 Amazon ECS 서비스로 푸시됩니다.

사전 조건 및 제한 사항

사전 조건

- Amazon ECS에서 실행되는 기존 Java 마이크로서비스 애플리케이션
- AWS CodeBuild 및 AWS에 대한 지식 CodePipeline

아키텍처

소스 기술 스택

- Amazon ECS에서 실행되는 Java 마이크로서비스
- Amazon ECR의 코드 리포지토리
- AWS Fargate

소스 아키텍처

대상 기술 스택

- Amazon ECR
- Amazon ECS
- AWS Fargate
- AWS CodePipeline
- AWS CodeBuild

대상 아키텍처

자동화 및 규모 조정

CodeBuild `buildspec.yml` 파일:

```
version: 0.2

phases:
  pre_build:
    commands:
      - echo Logging in to Amazon ECR...
      - aws --version
      - $(aws ecr get-login --region $AWS_DEFAULT_REGION --no-include-email)
      - REPOSITORY_URI=$AWS_ACCOUNT_ID.dkr.ecr.$AWS_DEFAULT_REGION.amazonaws.com/
        $IMAGE_REPO
      - COMMIT_HASH=$(echo $CODEBUILD_RESOLVED_SOURCE_VERSION | cut -c 1-7)
      - IMAGE_TAG=build-$(echo $CODEBUILD_BUILD_ID | awk -F":" '{print $2}')
```

build:

```
  commands:
    - echo Build started on `date`
    - echo building the Jar file
    - mvn clean install
    - echo Building the Docker image...
    - docker build -t $REPOSITORY_URI:$BUILD_TAG .
    - docker tag $REPOSITORY_URI:$BUILD_TAG $REPOSITORY_URI:$IMAGE_TAG
```

post_build:

```
  commands:
    - echo Build completed on `date`
```

```

- echo Pushing the Docker images...
- docker push $REPOSITORY_URI:$BUILD_TAG
- docker push $REPOSITORY_URI:$IMAGE_TAG
- echo Writing image definitions file...
- printf '[{"name": "%s", "imageUri": "%s"}]' $DOCKER_CONTAINER_NAME
$REPOSITORY_URI:$IMAGE_TAG > imagedefinitions.json
- cat imagedefinitions.json
artifacts:
  files:
    - imagedefinitions.json
    - target/DockerDemo.jar

```

도구

서비스

- [CodeBuild](#)은 소스 코드를 컴파일하고, 단위 테스트를 실행하고, 배포할 준비가 된 아티팩트를 생성하는 데 도움이 되는 완전관리형 빌드 서비스입니다. AWS는 지속적으로 CodeBuild 규모를 조정하고 여러 빌드를 동시에 처리하므로 빌드가 대기열에 남지 않습니다.
- [AWS](#)는 소프트웨어 릴리스의 여러 단계를 신속하게 모델링 및 구성하고 소프트웨어 변경 사항을 지속적으로 릴리스하는 CodePipeline 데 필요한 단계를 자동화하도록 지원합니다. AWS를 CodePipeline 와 같은 GitHub 타사 서비스와 통합하거나 AWS CodeCommit 또는 Amazon ECR과 같은 AWS 서비스를 사용할 수 있습니다.
- [Amazon Elastic Container Registry\(Amazon ECR\)](#)는 개발자가 Docker 컨테이너 이미지를 간편하게 저장, 관리 및 배포할 수 있게 해주는 완전 관리형 레지스트리입니다. Amazon ECR은 Amazon ECS 와 통합되어 워크플로를 단순화합니다. development-to-production Amazon ECR은 가용성과 확장성이 뛰어난 아키텍처에서 이미지를 호스팅하므로 애플리케이션을 위한 컨테이너를 안정적으로 배포할 수 있습니다. AWS Identity and Access Management(IAM)와 통합하면 각 리포지토리를 리소스 수준에서 제어할 수 있습니다.
- [Amazon Elastic Container Service\(Amazon ECS\)](#)는 Docker 컨테이너를 지원하고 AWS에서 컨테이너화된 애플리케이션을 쉽게 실행하고 확장할 수 있도록 도와주는 확장성이 뛰어난 고성능 컨테이너 오케스트레이션 서비스입니다. Amazon ECS를 사용하면 자체 컨테이너 오케스트레이션 소프트웨어를 설치 및 운영하거나, 가상 시스템 클러스터를 관리 및 확장하거나, 가상 시스템에서 컨테이너를 예약할 필요가 없습니다.
- [AWS Fargate](#)는 서버나 클러스터를 관리할 필요 없이 컨테이너를 실행할 수 있는 Amazon ECS용 컴퓨팅 엔진입니다. AWS Fargate를 사용하면 더 이상 컨테이너를 실행하기 위해 가상 머신의 클러스터를 프로비저닝, 구성 또는 확장할 필요가 없습니다. 따라서 서버 유형을 선택하거나, 클러스터를 조정할 시점을 결정하거나, 클러스터 패킹을 최적화할 필요가 없습니다.

기타 도구

- [Docker](#)는 컨테이너라는 패키지로 애플리케이션을 구축, 테스트 및 제공할 수 있는 플랫폼입니다.
- [Git](#)은 소프트웨어 개발 중에 소스 코드의 변경 사항을 추적하기 위한 분산 버전 제어 시스템입니다. 프로그래머 간의 작업 조정을 위해 설계되었지만 모든 파일 세트의 변경 사항을 추적하는 데 사용할 수 있습니다. 그 목표에는 속도, 데이터 무결성, 분산된 비선형 워크플로우 지원이 포함됩니다. Git의 CodeCommit 대안으로 AWS를 사용할 수도 있습니다.

에픽

AWS에서 빌드 프로젝트 설정 CodeBuild

작업	설명	필요한 기술
CodeBuild 빌드 프로젝트 만들기.	AWS CodeBuild 콘솔에서 빌드 프로젝트를 생성하고 이름을 지정합니다.	앱 개발자, AWS 시스템 관리자
소스를 선택합니다.	이 패턴은 코드 저장소로 Git을 사용하므로 사용 가능한 옵션 GitHub 목록에서 선택하십시오. 공용 리포지토리 또는 GitHub 계정에서 선택하세요.	앱 개발자, AWS 시스템 관리자
리포지토리를 선택합니다.	코드를 빌드할 리포지토리를 선택합니다.	앱 개발자, AWS 시스템 관리자
환경을 선택합니다.	관리형 이미지 목록에서 선택하거나 Docker를 사용하여 사용자 지정 이미지를 선택할 수 있습니다. 이 패턴은 다음과 같은 관리형 이미지를 사용합니다. <ul style="list-style-type: none"> • Amazon Linux 2 • 런타임: 표준 • 이미지 버전 1.0 	앱 개발자, AWS 시스템 관리자

작업	설명	필요한 기술
서비스 역할을 선택합니다.	서비스 역할을 생성하거나 기존 역할 목록에서 선택할 수 있습니다.	앱 개발자, AWS 시스템 관리자
환경 변수를 추가합니다.	<p>추가 구성 섹션에서 다음 환경 변수를 구성합니다.</p> <ul style="list-style-type: none"> • 기본 AWS 리전의 <code>AWS_DEFAULT_REGION</code> • 사용자 계정 번호의 <code>AWS_ACCOUNT_ID</code> • 아마존 ECR 프라이빗 리포지토리의 <code>IMAGE_REPO</code> • 빌드 버전의 <code>BUILD_TAG</code>(최신 빌드는 이 변수의 값임) • 해당 작업의 컨테이너 이름의 <code>DOCKER_CONTAINER_NAME</code> <p>이러한 변수는 <code>buildspec.yml</code> 파일의 자리 표시자이며 해당 값으로 대체됩니다.</p>	앱 개발자, AWS 시스템 관리자
buildspec 파일을 생성합니다.	pom.xml와(과) 동일한 위치에 <code>buildspec.yml</code> 파일을 생성하고 이 패턴에서 제공되는 구성을 추가하거나 온라인 buildspec 편집기를 사용하여 구성을 추가할 수 있습니다. 제공된 단계에 따라 적절한 값으로 환경 변수를 구성합니다.	앱 개발자, AWS 시스템 관리자

작업	설명	필요한 기술
아티팩트를 위한 프로젝트를 구성합니다.	(선택 사항) 필요한 경우 아티팩트에 대한 빌드 프로젝트를 구성합니다.	앱 개발자, AWS 시스템 관리자
Amazon CloudWatch 로그를 구성합니다.	(선택 사항) 필요한 경우 빌드 프로젝트에 대해 Amazon CloudWatch Logs를 구성합니다. 이 단계는 선택 사항이며, 권장 사항은 아닙니다.	앱 개발자, AWS 시스템 관리자
Amazon S3 로그를 구성합니다.	(선택 사항) 로그를 저장하고자 하는 경우 빌드 프로젝트에 대한 Amazon Simple Storage Service(Amazon S3) 로그를 구성합니다.	앱 개발자, AWS 시스템 관리자

AWS에서 파이프라인을 구성합니다. CodePipeline

작업	설명	필요한 기술
파이프라인 생성.	AWS CodePipeline 콘솔에서 파이프라인을 생성하고 이름을 지정합니다. 파이프라인 생성에 대한 자세한 내용은 AWS CodePipeline 설명서를 참조하십시오.	앱 개발자, AWS 시스템 관리자
서비스 역할을 선택합니다.	서비스 역할을 생성하거나 기존 서비스 역할 목록에서 선택합니다. 서비스 역할을 생성하는 경우 역할 이름을 제공하고 역할 생성 옵션을 선택하십시오. CodePipeline	앱 개발자, AWS 시스템 관리자

작업	설명	필요한 기술
아티팩트 스토어를 선택합니다.	고급 설정에서 Amazon S3가 버킷을 생성하고 그 안에 아티팩트를 저장하도록 하려면 아티팩트 스토어의 기본 위치를 사용해야 합니다. 또는 사용자 지정 위치를 선택하고 기존 버킷을 지정합니다. 암호화 키를 사용하여 아티팩트 암호화를 선택할 수도 있습니다.	앱 개발자, AWS 시스템 관리자
소스 공급자를 지정합니다.	소스 제공자의 경우 GitHub (버전 2) 를 선택합니다.	앱 개발자, AWS 시스템 관리자
리포지토리와 코드의 브랜치를 선택합니다.	로그인하지 않은 경우 연결할 GitHub 연결 세부 정보를 제공한 다음 리포지토리 이름과 분기 이름을 선택합니다.	앱 개발자, AWS 시스템 관리자
탐지 옵션을 변경합니다.	소스 코드 변경 시 파이프라인 시작을 선택하고 다음 페이지로 이동합니다.	앱 개발자, AWS 시스템 관리자
빌드 공급자를 선택합니다.	빌드 공급자의 경우 AWS를 CodeBuild 선택한 다음 빌드 프로젝트에 대한 AWS 지역 및 프로젝트 이름 세부 정보를 제공합니다. 빌드 유형에서 단일 빌드를 선택합니다.	앱 개발자, AWS 시스템 관리자

작업	설명	필요한 기술
배포 공급자를 선택합니다.	배포 공급자의 경우, Amazon ECS를 선택합니다. 필요한 경우 클러스터 이름, 서비스 이름, 이미지 정의 파일(있는 경우) 및 배포 제한 시간 값을 선택합니다. 파이프라인 생성을 선택합니다.	앱 개발자, AWS 시스템 관리자

관련 리소스

- [AWS ECS 설명서](#)
- [AWS ECR 설명서](#)
- [AWS CodeBuild 설명서](#)
- [AWS CodeCommit 설명서](#)
- [AWS CodePipeline 설명서](#)
- [Amazon ECR을 소스로 사용하여 컨테이너 이미지를 위한 지속적 전송 파이프라인 구축 \(블로그 게시물\)](#)

CodeCommit AWS와 AWS를 사용하여 여러 AWS CodePipeline 계정에 CI/CD 파이프라인을 배포할 수 있습니다.

작성자: Kirankumar Chandrashekar(AWS)

환경: PoC 또는 파일럿	기술: DevOps	워크로드: 기타 모든 워크로드
AWS 서비스: AWS CodeCommit, AWS CodePipeline		

요약

이 패턴은 애플리케이션 코드 워크로드에 대한 CI/CD (지속적 통합 및 지속적 전송) 파이프라인을 별도의 Amazon Web Services (AWS) 계정 DevOps, 개발자, 스테이징 및 프로덕션 워크플로에 배포하는 방법을 보여줍니다.

[여러 AWS 계정 전략](#)을 사용하여 높은 수준의 [리소스 또는 보안 격리](#)를 제공하고, [비용을 최적화](#)하며, 프로덕션 워크플로를 분리할 수 있습니다.

애플리케이션 코드는 이러한 개별 AWS 계정에서 동일하게 유지되며 계정에서 호스팅하는 중앙 AWS CodeCommit 리포지토리에서 유지 관리됩니다. DevOps 개발자, 스테이징 및 프로덕션 계정의 이 리포지토리에는 별도의 Git 브랜치가 있습니다 CodeCommit .

예를 들어 중앙 CodeCommit 리포지토리의 개발자 Git 브랜치에 코드가 커밋되면 계정의 EventBridge Amazon이 개발자 DevOps 계정에 리포지토리 변경 사항을 알립니다 EventBridge . 개발자 계정에서는 CodePipeline AWS와 [소스 단계가](#) InProgress 상태로 전환됩니다. 소스 단계는 중앙 CodeCommit 리포지토리의 개발자 Git 브랜치에서 구성되며 CodePipeline 계정의 [서비스 역할을 맡습니다](#). DevOps

개발자 브랜치에 있는 CodeCommit 리포지토리의 콘텐츠는 Amazon Simple Storage Service (Amazon S3) 버킷의 아티팩트 스토어에 업로드되고 AWS KMS (키 관리 서비스) 키로 암호화됩니다. [소스 단계의 상태가 in으로 Succeeded 변경되면 코드는 파이프라인 실행의 다음 단계로 전환됩니다](#). [CodePipeline](#)

사전 조건 및 제한 사항

사전 조건

- 각 필수 환경 (DevOps, 개발자, 스테이징, 프로덕션) 에 대한 기존 AWS 계정 이러한 계정은 [AWS Organizations](#)에서 호스팅할 수 있습니다.
- AWS Command Line Interface(AWS CLI), [설치](#) 및 [구성됨](#)

아키텍처

기술 스택

- AWS CodeBuild
- AWS CodeCommit
- AWS CodePipeline
- 아마존 EventBridge
- AWS Identity and Access Management(IAM)
- AWS KMS
- AWS Organizations
- Amazon S3

도구

- [AWS CodeBuild](#) — 소스 코드를 컴파일하고, 테스트를 실행하고, 배포 준비가 완료된 소프트웨어 패키지를 생성하는 완전관리형 지속적 통합 CodeBuild 서비스입니다.
- [AWS CodeCommit](#) — [안전한 Git](#) 기반 리포지토리를 호스팅하는 완전 관리형 소스 제어 CodeCommit 서비스입니다.
- [AWS CodePipeline](#) — 빠르고 안정적인 애플리케이션 및 인프라 업데이트를 위해 릴리스 파이프라인을 자동화하는 데 도움이 되는 완전관리형 지속적 전송 CodePipeline 서비스입니다.
- [Amazon EventBridge](#) — EventBridge 애플리케이션을 다양한 소스의 데이터와 연결하는 서버리스 이벤트 버스 서비스입니다.
- [AWS Identity and Access Management\(IAM\)](#) – IAM을 통해 AWS 서비스와 리소스에 대한 액세스를 안전하게 관리할 수 있습니다.

- [AWS KMS](#) – AWS Key Management Service(AWS KMS)를 사용하면 다양한 AWS 서비스와 애플리케이션에서 암호화 키를 생성 및 관리하고 암호화 키 사용을 제어할 수 있습니다.
- [Amazon S3](#) – Amazon Simple Storage Service(S3)는 인터넷 스토리지 서비스입니다.

에픽

DevOps AWS 계정에서 리소스 생성

작업	설명	필요한 기술
CodeCommit 리포지토리 생성.	DevOps 계정의 AWS Management 콘솔에 로그인하고 CodeCommit 콘솔을 엽니다. 리포지토리를 생성하고 개발자, 스테이징 및 프로덕션 AWS 계정에 필요한 모든 Git 브랜치를 설정합니다. 이 스토리와 다른 스토리에 대한 도움이 필요하면 “관련 리소스” 섹션을 참조하십시오.	DevOps 엔지니어
CodeCommit 리포지토리에 대한 액세스 자격 증명을 생성합니다.	IAM 콘솔에서 애플리케이션 개발자가 CodeCommit 리포지토리에서 애플리케이션의 코드베이스를 푸시하고 가져올 수 있도록 액세스 자격 증명을 생성합니다.	DevOps 엔지니어
CodePipeline 서비스 역할을 위한 IAM 역할을 생성합니다.	IAM 콘솔에서 모든 CodePipeline 서비스 역할이 중앙 리포지토리에 액세스하는 데 사용할 수 있는 IAM 역할을 생성합니다. CodeCommit	클라우드 관리자
다른 AWS 계정에 대한 EventBridge 규칙을 설정합니다.	Amazon EventBridge 콘솔에서 개별 개발자, 스테이징 및 프로덕션 AWS 계정의 관련	클라우드 관리자

작업	설명	필요한 기술
	CodeCommit 리포지토리 변경 사항에 EventBridge 대한 알림을 보내는 규칙을 설정합니다.	
AWS KMS 키를 생성합니다.	AWS KMS 콘솔에서 개별 개발자, 스테이징 및 프로덕션 AWS CodePipeline 계정에서 아티팩트를 암호화하고 해독할 수 있는 KMS 키를 생성합니다.	클라우드 관리자

다른 AWS 계정에서 리소스 생성

작업	설명	필요한 기술
DevOps AWS 계정에서 이벤트를 EventBridge 수신하도록 설정합니다.	개별 AWS 계정(개발자, 스테이징 또는 프로덕션) 중 하나에 대해 AWS Management Console에 로그인합니다. Amazon EventBridge 콘솔에서 DevOps 계정으로부터 CodeCommit 리포지토리 변경 이벤트를 EventBridge 수신하도록 설정합니다.	클라우드 관리자
S3 버킷을 생성합니다.	Amazon S3 콘솔에서 S3 버킷을 생성하여 CodePipeline 아티팩트를 저장합니다.	클라우드 관리자
CodePipeline 스테이지에 필요한 모든 AWS 리소스를 생성합니다.	CodePipeline 단계에 필요한 다른 모든 AWS 리소스를 생성합니다. 이러한 리소스는 CI/CD 파이프라인에서 각 AWS 계정의 역할에 따라 달라집니다.	클라우드 관리자

작업	설명	필요한 기술
IAM 역할을 생성합니다.	IAM 콘솔에서 CodePipeline 서비스 역할에 대한 IAM 역할을 생성합니다. 이 서비스 역할은 DevOps 계정에서 IAM 역할을 맡아 리포지토리에 액세스할 수 있어야 합니다. CodeCommit	클라우드 관리자
에서 CodePipeline 파이프라인을 생성하십시오.	CodePipeline 콘솔에서 파이프라인을 생성합니다. 그런 다음 개별 Git 브랜치의 DevOps 계정에 있는 CodeCommit 리포지토리를 가리키는 소스 단계를 생성합니다.	클라우드 관리자
모든 AWS 계정에 대해 이 단계를 반복합니다.	CI/CD 전략의 일부로 필요한 모든 AWS 계정에 대해 이 단계를 반복합니다.	클라우드 관리자

관련 리소스

DevOps AWS 계정에서 리소스 생성

- [CodeCommit 리포지토리 생성](#)
- [CodeCommit 리포지토리 설정](#)
- [CodeCommit 리포지토리에 브랜치를 만들고 공유하세요.](#)
- [CodeCommit 리포지토리에 대한 액세스 자격 증명 생성](#)
- [CodePipeline 서비스 역할을 위한 IAM 역할 생성](#)
- [에서 규칙을 설정합니다. EventBridge](#)
- [AWS KMS 키 생성](#)
- [에 대한 계정 정책 및 역할을 설정합니다. CodePipeline](#)

다른 AWS 계정에서 리소스 생성

- [DevOps AWS 계정에서 이벤트를 EventBridge 수신하려면 켜십시오.](#)
- [CodePipeline 아티팩트를 위한 S3 버킷 생성](#)
- [CodePipeline 스테이지에 필요한 기타 모든 AWS 리소스 생성](#)
- [CodePipeline 서비스 역할을 위한 IAM 역할 생성](#)
- [에서 파이프라인을 생성하십시오. CodePipeline](#)
- [다른 AWS 계정의 리소스를 CodePipeline 사용하는 파이프라인을 생성합니다.](#)

기타 리소스

- [모범 사례 AWS 환경 구축](#)
- [에 대한 인증 및 액세스 제어 CodeCommit](#)

AWS Network Firewall과 AWS Transit Gateway를 사용하여 방화벽 배포

작성자: 슈리칸트 파틸(AWS)

코드 리포지토리: [aws-network-firewall-deployment - with-transit-gateway](#)

환경: PoC 또는 파일럿

기술: DevOps; 네트워킹; 보안, ID, 규정 준수

AWS 서비스: AWS 네트워크 방화벽, AWS Transit Gateway, 아마존 VPC, 아마존 CloudWatch

요약

이 패턴은 AWS Network Firewall 및 AWS Transit Gateway를 사용하여 방화벽을 배포하는 방법을 보여줍니다. Network Firewall 리소스는 AWS CloudFormation 템플릿을 사용하여 배포됩니다. Network Firewall은 네트워크 트래픽에 따라 자동으로 확장되며 수십만 개의 연결을 지원할 수 있으므로 자체 네트워크 보안 인프라를 구축하고 유지 관리에 대해 걱정할 필요가 없습니다. Transit Gateway는 가상 사설 클라우드(VPC)와 온프레미스 네트워크를 상호 연결하는 데 사용할 수 있는 네트워크 전송 허브입니다.

이 패턴에서는 네트워크 아키텍처에 검사 VPC를 포함하는 방법도 학습합니다. 마지막으로, 이 패턴은 CloudWatch Amazon을 사용하여 방화벽에 실시간 활동 모니터링을 제공하는 방법을 설명합니다.

팁: Network Firewall 서브넷을 사용하여 다른 AWS 서비스를 배포하지 않는 것이 좋습니다. 이는 Network Firewall이 방화벽의 서브넷 내 소스 또는 목적지에서 오는 트래픽을 검사할 수 없기 때문입니다.

사전 조건 및 제한 사항

사전 조건

- 활성 상태의 AWS 계정
- AWS Identity and Access Management(IAM) 역할 및 정책 권한

- CloudFormation 템플릿 권한

제한 사항

도메인 필터링에 문제가 있을 수 있으며 다른 종류의 구성이 필요할 수 있습니다. 자세한 내용은 Network Firewall 설명서의 [AWS Network Firewall의 상태 저장 도메인 목록 규칙 그룹](#)을 참조하세요.

아키텍처

기술 스택

- 아마존 CloudWatch 로그
- Amazon VPC
- AWS Network Firewall
- AWS Transit Gateway

대상 아키텍처

다음 다이어그램은 Network Firewall 및 Transit Gateway를 사용하여 트래픽을 검사하는 방법을 보여줍니다.

아키텍처에는 다음 구성 요소가 포함되어 있습니다.

- 애플리케이션은 두 개의 스포크 VPC에서 호스팅됩니다. VPC는 Network Firewall에 의해 모니터링됩니다.
- 송신 VPC는 인터넷 게이트웨이에 직접 액세스할 수 있지만 Network Firewall의 보호를 받지 못합니다.
- 검사 VPC는 Network Firewall이 배포되는 곳입니다.

자동화 및 규모 조정

[인프라를 코드로](#) 사용하여 이 패턴을 생성할 수 있습니다. [CloudFormation](#)

도구

서비스

- [Amazon CloudWatch Logs](#)를 사용하면 모든 시스템, 애플리케이션 및 AWS 서비스의 로그를 중앙 집중화하여 로그를 모니터링하고 안전하게 보관할 수 있습니다.
- [Amazon Virtual Private Cloud\(VPC\)](#)를 이용하면 사용자가 정의한 가상 네트워크로 AWS 리소스를 시작할 수 있습니다. 이 가상 네트워크는 사용자의 자체 데이터 센터에서 운영하는 기존 네트워크와 유사하며 AWS의 확장 가능한 인프라를 사용한다는 이점이 있습니다.
- [AWS Network Firewall](#)은 AWS Cloud를 위한 상태 저장형, 관리형 네트워크 방화벽이자, 침입 탐지 및 방지 서비스입니다.
- [AWS Transit Gateway](#)는 VPC와 온프레미스 네트워크를 연결하는 중앙 허브입니다.

코드

이 패턴의 코드는 [Transit Gateway 리포지토리를 사용한 GitHub AWS Network Firewall 배포에서](#) 사용할 수 있습니다. 이 리포지토리의 CloudFormation 템플릿을 사용하여 Network Firewall을 사용하는 단일 검사 VPC를 배포할 수 있습니다.

에픽

스포크 VPC 및 검사 VPC 생성

작업	설명	필요한 기술
CloudFormation 템플릿을 준비하고 배포하세요.	<ol style="list-style-type: none"> 1. GitHub 리포지토리에서 cloudformation/aws_nw_fw.yml 템플릿을 다운로드합니다. 2. 템플릿을 원하는 값으로 업데이트합니다. 3. 템플릿을 배포합니다. 	AWS DevOps

전송 게이트웨이 및 경로 생성

작업	설명	필요한 기술
전송 게이트웨이를 생성합니다.	<ol style="list-style-type: none"> 1. AWS Management Console에 로그인한 후 Amazon VPC 콘솔을 엽니다. 	AWS DevOps

작업	설명	필요한 기술
	<ol style="list-style-type: none"> 2. 탐색 창에서 Transit Gateway를 선택합니다. 3. 전송 게이트웨이 생성을 선택합니다. 4. 이름 태그에 전송 게이트웨이의 이름을 입력합니다. 5. 설명에 전송 게이트웨이에 대한 설명을 입력합니다. 6. Amazon 측 Autonomous System Number(ASN)에는 기본 ASN 값을 그대로 사용합니다. 7. DNS 지원 옵션을 선택합니다. 8. VPN ECMP 지원 옵션을 선택합니다. 9. 기본 라우팅 테이블 연결 옵션을 선택합니다. 이 옵션은 Transit Gateway Attachment를 전송 게이트웨이의 기본 경로 테이블과 자동으로 연결합니다. 10. 기본 라우팅 테이블 전파 옵션을 선택합니다. 이 옵션은 전송 게이트웨이의 기본 라우팅 테이블에 Transit Gateway Attachment를 자동으로 전파합니다. 11. 전송 게이트웨이 생성을 선택합니다. 	

작업	설명	필요한 기술
<p>전송 게이트웨이 연결을 생성합니다.</p>	<p>다음에 대한 전송 게이트웨이 연결 생성:</p> <ul style="list-style-type: none"> • 검사 VPC 및 Transit Gateway 서브넷의 Inspection 연결 • 스포크 VPCA 및 프라이빗 서브넷의 SpokeVPCA 연결 • 스포크 VPCB 및 프라이빗 서브넷의 SpokeVPCB 연결 • 송신 VPC 및 프라이빗 서브넷의 EgressVPC 연결 	<p>AWS DevOps</p>

작업	설명	필요한 기술
<p>전송 게이트웨이 라우팅 테이블을 생성합니다.</p>	<ol style="list-style-type: none"> 1. 스포크 VPC에 대한 전송 게이트웨이 라우팅 테이블을 생성합니다. 이 라우팅 테이블은 검사 VPC를 제외한 모든 VPC에 연결되어야 합니다. 2. 방화벽에 대한 전송 게이트웨이 라우팅 테이블을 생성합니다. 이 라우팅 테이블은 검사 VPC에만 연결되어야 합니다. 3. 방화벽에 대한 라우팅 테이블에 경로를 추가합니다. <ul style="list-style-type: none"> • 0.0.0/0의 경우 EgressVPC 연결을 사용합니다. • SpokeVPCA CIDR 블록의 경우 SpokeVPC1 연결을 사용합니다. • SpokeVPCB CIDR 블록의 경우 SpokeVPC2 연결을 사용합니다. 4. 스포크 VPC의 전송 게이트웨이 라우팅 테이블에 경로를 추가합니다. 0.0.0/0의 경우 Inspection VPC 연결을 사용합니다. 	<p>AWS DevOps</p>

방화벽 및 경로 생성

작업	설명	필요한 기술
검사 VPC에서 방화벽을 생성합니다.	<ol style="list-style-type: none"> 1. AWS Management Console에 로그인한 후 Amazon VPC 콘솔을 엽니다. 2. 탐색 창의 Network Firewall에서 방화벽을 선택합니다. 3. 방화벽 생성을 선택합니다. 4. 이름에 이 방화벽을 식별하는 데 사용할 이름을 입력합니다. 방화벽을 생성한 후에는 해당 이름을 변경할 수 없습니다. 5. VPC의 경우 검사 VPC를 선택합니다. 6. 가용 영역 및 서브넷의 경우 식별한 영역 및 방화벽 서브넷을 선택합니다. 7. 연결된 방화벽 정책 섹션에서 기존 방화벽 연결 정책을 선택한 다음 이전에 만든 방화벽 정책을 선택합니다. 8. 방화벽 생성을 선택합니다. 	AWS DevOps
방화벽 정책을 생성합니다.	<ol style="list-style-type: none"> 1. AWS Management Console에 로그인한 후 Amazon VPC 콘솔을 엽니다. 2. 탐색 창의 Network Firewall에서 방화벽 정책을 선택합니다. 	AWS DevOps

작업	설명	필요한 기술
	<ol style="list-style-type: none"> 3. 방화벽 정책 설명 페이지에서 방화벽 정책 생성을 선택합니다. 4. 이름의 경우 방화벽 정책에 사용할 이름을 입력합니다. 나중에 이 패턴에서 정책을 방화벽에 연결할 때 이 이름을 사용하여 정책을 식별합니다. 트래픽 정책을 생성한 후에는 해당 이름을 변경할 수 없습니다. 5. 다음을 선택합니다. 6. 규칙 그룹 추가 페이지의 상태 비저장 규칙 그룹 섹션에서 상태 비저장 규칙 그룹 추가를 선택합니다. 7. 기존 규칙 그룹에서 추가 대화 상자에서 이전에 생성한 비저장 규칙 그룹의 확인란을 선택합니다. 규칙 그룹 추가를 선택합니다. 참고: 페이지 하단의 방화벽 정책 용량 카운터는 방화벽 정책에 허용된 최대 용량 옆에 이 규칙 그룹을 추가하여 소비한 용량을 보여줍니다. 8. 상태 비저장 기본 작업을 상태 저장 규칙으로 전달로 설정합니다. 9. 상태 저장 규칙 그룹 섹션에서 상태 저장 규칙 그룹 추가를 선택한 다음 이전에 생성한 상태 저장 규칙 그룹의 	

작업	설명	필요한 기술
	<p>확인란을 선택합니다. 규칙 그룹 추가를 선택합니다.</p> <p>10.다음을 선택하여 나머지 설치 마법사를 단계별로 진행한 다음 방화벽 정책 생성을 선택합니다.</p>	

작업	설명	필요한 기술
<p>VPC 라우팅 테이블을 업데이트합니다.</p>	<p>Inspection VPC 라우팅 테이블</p> <ol style="list-style-type: none"> 1. ANF 서브넷 라우팅 테이블 (Inspection-ANFRT) 에서 Transit Gateway ID에 0.0.0/0을(를) 추가합니다. 2. Transit Gateway 서브넷 라우팅 테이블 (Inspection-TGWRT) 에서 EgressVPC에 0.0.0/0을(를) 추가합니다. <p>SpokeVPCA 라우팅 테이블</p> <p>프라이빗 라우팅 테이블에 서 Transit Gateway ID에 0.0.0.0/0 을(를) 추가합니다.</p> <p>Spoke VPCB 라우팅 테이블</p> <p>프라이빗 라우팅 테이블에 서 Transit Gateway ID에 0.0.0.0/0 을(를) 추가합니다.</p> <p>Egress VPC 라우팅 테이블</p> <p>송신 퍼블릭 라우팅 테이블 에서 Transit Gateway ID에 SpokeVPCA 및 Spoke VPCB CIDR 블록을 추가합니다. 프라이빗 서브넷에 대해 동일한 단계를 반복합니다.</p>	<p>AWS DevOps</p>

실시간 네트워크 검사를 CloudWatch 수행하도록 설정

작업	설명	필요한 기술
방화벽의 로깅 구성을 업데이트합니다.	<ol style="list-style-type: none"> 1. AWS Management Console에 로그인한 후 Amazon VPC 콘솔을 엽니다. 2. 탐색 창의 Network Firewall에서 방화벽을 선택합니다. 3. 방화벽 페이지에서 편집하려는 방화벽 이름을 선택합니다. 4. 방화벽 세부 정보 탭을 선택합니다. 로깅 섹션에서 편집을 선택합니다. 5. 필요에 따라 로그 유형 선택을 조정합니다. 알림 및 흐름 로그에 대한 로깅을 구성할 수 있습니다. <ul style="list-style-type: none"> • 알림 - 작업이 알림 또는 드롭으로 설정된 모든 상태 저장 규칙과 일치하는 트래픽에 대한 로그를 전송합니다. 상태 저장 규칙 및 규칙 그룹에 대한 자세한 내용은 AWS Network Firewall의 규칙 그룹을 참조하세요. • 흐름 - 상태 저장 엔진이 상태 저장 규칙 엔진에 전달하는 모든 네트워크 트래픽에 대한 로그를 전송합니다. 	AWS DevOps

작업	설명	필요한 기술
	<p>6. 선택한 각 로그 유형에 대해 대상 유형을 선택한 다음 로깅 대상에 대한 정보를 제공합니다. 자세한 내용은 Network Firewall 설명서의 AWS Network Firewall 로깅 대상을 참조하세요.</p> <p>7. 저장을 선택합니다.</p>	

설정 확인

작업	설명	필요한 기술
<p>EC2 인스턴스를 시작하여 설정을 테스트합니다.</p>	<p>스포크 VPC에서 2개의 Amazon Elastic Compute Cloud(EC2) 인스턴스를 시작합니다. 하나는 Jumpbox용 이고 다른 하나는 테스트 연결 용입니다.</p>	<p>AWS DevOps</p>
<p>지표를 확인합니다.</p>	<p>지표는 먼저 서비스 네임스페이스별로 그룹화된 다음, 각 네임스페이스 내에서 다양한 차원 조합별로 그룹화됩니다. Network Firewall의 CloudWatch 네임스페이스는 <code>AWS/NetworkFirewall</code></p> <ol style="list-style-type: none"> 1. AWS Management Console에 로그인하고 CloudWatch 콘솔을 엽니다. 2. 탐색 창에서 [지표]를 선택합니다. 	<p>AWS DevOps</p>

작업	설명	필요한 기술
	3. 모든 지표 탭에서 지역을 선택한 다음 NetworkFi rewallAWS/를 선택합니다.	

관련 리소스

- [인터넷 게이트웨이가 있는 간단한 단일 영역 아키텍처](#)
- [인터넷 게이트웨이가 있는 다중 영역 아키텍처](#)
- [인터넷 게이트웨이와 NAT 게이트웨이가 있는 아키텍처](#)

AWS CodePipeline CI/CD 파이프라인을 사용하여 AWS Glue 작업을 배포하십시오.

작성자: Bruno Klein(AWS) 및 Luis Henrique Massao Yamada(AWS)

환경: 프로덕션	기술: DevOps; 빅 데이터	AWS 서비스: AWS Glue, AWS, AWS CodeCommit CodePipeline, AWS Lambda
----------	-------------------	---

요약

이 패턴은 Amazon Web Services (AWS) CodeCommit 와 AWS를 AWS CodePipeline Glue와 통합하고 개발자가 변경 내용을 원격 AWS 리포지토리로 푸시하는 즉시 AWS Lambda를 사용하여 작업을 시작하는 방법을 보여줍니다. CodeCommit

개발자가 ETL (추출, 변환, 로드) 리포지토리에 변경 사항을 제출하고 변경 내용을 CodeCommit AWS 에 푸시하면 새 파이프라인이 호출됩니다. 파이프라인은 이러한 변경과 함께 Glue 작업을 시작하는 Lambda 함수를 시작합니다. Glue 작업은 ETL 작업을 수행합니다.

이 솔루션은 기업, 개발자, 데이터 엔지니어가 변경 내용을 커밋하고 대상 리포지토리에 푸시하는 즉시 작업을 시작하려는 경우에 유용합니다. 이를 통해 자동화 및 재현성 수준을 높여 작업 시작 및 수명 주기 동안 오류를 방지할 수 있습니다.

사전 조건 및 제한 사항

사전 조건

- 활성 상태의 AWS 계정.
- 로컬 시스템에 설치되는 [Git](#)
- 로컬 시스템에 설치되는 [Amazon Cloud Development Kit\(Amazon CDK\)](#)
- 로컬 머신에 설치되는 [Python](#)
- 첨부 파일 섹션의 코드

제한 사항

- 파이프라인은 Glue 작업이 성공적으로 시작되는 즉시 완료됩니다. 작업이 마무리될 때까지 기다리지 않습니다.
- 첨부 파일에 나와 있는 코드는 데모용으로만 사용됩니다.

아키텍처

대상 기술 스택

- Glue
- Lambda
- AWS CodePipeline
- AWS CodeCommit

대상 아키텍처

프로세스는 다음 단계로 구성됩니다.

1. 개발자 또는 데이터 엔지니어가 ETL 코드를 수정하고 변경 사항을 커밋하고 AWS에 푸시합니다. CodeCommit
2. 푸시는 파이프라인을 시작합니다.
3. 파이프라인은 리포지토리의 `codecommit:GetFile(을)`를 호출하고 파일을 Amazon Simple Storage Service(S3)에 업로드하는 Lambda 함수를 시작합니다.
4. Lambda 함수는 ETL 코드를 사용하여 새로운 Glue 작업을 시작합니다.
5. Lambda 함수는 파이프라인을 완료합니다.

자동화 및 규모 조정

샘플 첨부 파일은 AWS Glue를 CodePipeline AWS와 통합하는 방법을 보여줍니다. 용도에 맞게 사용자 지정하거나 확장할 수 있는 기본 예제를 제공합니다. 자세한 내용은 에픽 섹션을 참조하세요.

도구

- [AWS CodePipeline](#) — CodePipeline AWS는 빠르고 안정적인 애플리케이션 및 인프라 업데이트를 위해 릴리스 파이프라인을 자동화하는 데 도움이 되는 완전관리형 [지속적 전송](#) 서비스입니다.

- [AWS CodeCommit](#) — CodeCommit AWS는 안전한 Git 기반 리포지토리를 호스팅하는 완전 관리형 [소스 제어](#) 서비스입니다.
- [Lambda](#) – Lambda는 서버를 프로비저닝하거나 관리하지 않고도 코드를 실행할 수 있게 해주는 서버리스 컴퓨팅 서비스입니다.
- [Glue](#) – Glue는 분석, 기계 학습, 애플리케이션 개발을 위해 데이터를 쉽게 검색, 준비, 결합할 수 있게 해주는 서버리스 데이터 통합 서비스입니다.
- [Git client](#) — Git은 GUI 도구를 제공하거나, 명령줄 또는 데스크톱 도구를 사용하여 필요한 아티팩트를 체크아웃할 수 있습니다. GitHub
- [CDK](#) - CDK는 오픈 소스 소프트웨어 개발 프레임워크로, 익숙한 프로그래밍 언어를 사용하여 클라우드 애플리케이션 리소스를 정의하도록 지원합니다.

에픽

샘플 코드의 배포

작업	설명	필요한 기술
CLI를 구성합니다.	현재의 계정을 대상으로 지정하고 인증하도록 Command Line Interface(CLI)를 구성합니다. 자세한 지침은 CLI 설명서 를 참조하십시오.	개발자, 엔지니어 DevOps
샘플 프로젝트 파일을 추출합니다.	첨부 파일에서 파일을 추출하여, 샘플 프로젝트 파일이 포함된 폴더를 생성합니다.	개발자, DevOps 엔지니어
샘플 코드를 배포합니다.	파일을 추출한 후 추출 위치에서 다음 명령을 실행하여 기본 예제를 생성합니다.	개발자, DevOps 엔지니어

```

cdk bootstrap
cdk deploy
git init
git remote add origin
<code-commit-repository-url>
    
```

작업	설명	필요한 기술
	<pre>git stage . git commit -m "adds sample code" git push --set-upstream origin main</pre> <p>마지막 명령 후 파이프라인의 상태와 Glue 작업을 모니터링 할 수 있습니다.</p>	
코드를 사용자 지정합니다.	비즈니스 요구 사항에 따라 etl.py 파일의 코드를 사용자 지정합니다. ETL 코드를 수정하거나, 파이프라인 단계를 수정하거나, 솔루션을 확장할 수 있습니다.	데이터 엔지니어

관련 리소스

- [SDK 시작하기](#)
- [AWS Glue에 작업 추가](#)
- [소스 액션 통합 CodePipeline](#)
- [다음 파이프라인에서 AWS Lambda 함수를 호출합니다. CodePipeline](#)
- [Glue 프로그래밍](#)
- [AWS CodeCommit GetFile API](#)

첨부

이 문서와 관련된 추가 콘텐츠에 액세스하려면 [attachment.zip](#) 파일의 압축을 풉니다.

EC2 인스턴스 프로파일을 사용하여 AWS Cloud9에서 Amazon EKS 클러스터의 배포

작성자: Sagar Panigrahi

환경: 프로덕션

기술: DevOps; 컨테이너 및 마이크로서비스

워크로드: 기타 모든 워크로드

AWS 서비스: 아마존 EKS, AWS Cloud9, AWS 자격 증명 및 액세스 관리, AWS CloudFormation

요약

이 패턴은 AWS Cloud9와 AWS를 사용하여 Amazon Web Services (CloudFormation AWS) 계정의 사용자가 프로그래밍 방식으로 액세스할 수 없도록 운영할 수 있는 Amazon Elastic Kubernetes Service (Amazon EKS) 클러스터를 만드는 방법을 설명합니다.

Cloud9은 브라우저를 사용하여 코드를 작성, 실행 및 디버깅하는 데 사용하는 클라우드 기반 통합 개발 환경(IDE)입니다. AWS Cloud9는 아마존 Elastic Compute Cloud (Amazon EC2) 인스턴스 프로파일과 AWS 템플릿을 사용하여 Amazon EKS 클러스터를 프로비저닝하는 제어 센터로 사용됩니다. CloudFormation

Identity and Access Management(IAM) 사용자를 생성하지 않고 대신 IAM 역할을 사용하려는 경우 이 패턴을 사용할 수 있습니다. 역할 기반 액세스 제어(RBAC)는 개별 사용자의 역할을 기준으로 리소스에 대한 액세스를 규제합니다. 이 패턴은 Amazon EKS 클러스터 내에서 RBAC를 업데이트하여 특정 IAM 역할에 대한 액세스를 허용하는 방법을 보여줍니다.

또한 패턴 설정은 DevOps 팀이 AWS Cloud9 기능을 사용하여 Amazon EKS 인프라 생성을 위한 코드형 인프라 (IaC) 리소스를 유지 관리하고 개발하는 데도 도움이 됩니다.

사전 조건 및 제한 사항

사전 조건

- 활성 상태의 AWS 계정.

- 계정에 대한 IAM 역할 및 정책을 생성할 수 있는 권한 사용자의 IAM 역할에는 AWSCloud9Administrator 정책이 포함되어야 합니다. 또한 AWSServiceRoleForAmazonEKS 및 eksNodeRoles 역할은 Amazon EKS 클러스터를 생성하는 데 필요하므로 생성해야 합니다.
- Kubernetes 개념에 대한 지식.

제한 사항

- 이 패턴은 기본 Amazon EKS 클러스터를 생성하는 방법을 설명합니다. 프로덕션 클러스터의 경우 AWS CloudFormation 템플릿을 업데이트해야 합니다.
- 이 패턴은 추가 Kubernetes 구성 요소를 배포하지 않습니다(예: [Fluentd](#), [인그레스 컨트롤러](#) 또는 [스토리지 컨트롤러](#)).

아키텍처

기술 스택

- Cloud9
- AWS CloudFormation
- Amazon EKS
- IAM

자동화 및 규모 조정

이 패턴을 확장하고 지속적 통합 및 지속적 배포(CI/CD) 파이프라인에 통합하여 Amazon EKS의 완전한 프로비저닝을 자동화할 수 있습니다.

도구

- [AWS CloudFormation](#) — AWS를 CloudFormation 사용하면 AWS 리소스를 모델링하고 설정하여 리소스를 관리하는 시간을 줄이고 애플리케이션에 더 많은 시간을 집중할 수 있습니다.
- [AWS Cloud9](#) – AWS Cloud9은 여러 프로그래밍 언어와 런타임 디버거 및 기본 제공 터미널을 지원하는 풍부한 코드 편집 경험을 제공합니다.

- [CLI](#) – Command Line Interface(CLI))는 명령줄 쉘에서 명령을 사용하여 서비스와 상호 작용할 수 있는 오픈 소스 도구입니다.
- [Kubectl](#) — kubect1(은)는 Amazon EKS 클러스터와 상호 작용하는 데 사용할 수 있는 명령줄 유틸리티입니다.

에픽

EC2 인스턴스 프로파일을 위한 IAM 역할 생성

작업	설명	필요한 기술
IAM 정책을 생성합니다.	<p>AWS Management Console에 로그인하고 IAM 콘솔을 열어 정책을 선택한 다음 정책 생성을 선택합니다. JSON 탭을 선택하고 policy-role-eks-instance - profile-for-cloud9.json 파일 (첨부)의 내용을 붙여넣습니다.</p> <p>정책 검증 동안 생성된 모든 보안 경고, 오류 또는 일반 경고를 해결한 다음 정책 검토(Review policy)를 선택합니다. 정책의 이름을 입력합니다. 정책 이름으로 eks-instance-profile-for-cloud9 을(를) 사용하는 것이 좋습니다.</p> <p>정책 요약을 검토하여 정책이 부여한 권한을 확인합니다. 그런 다음 정책 생성을 선택합니다.</p>	클라우드 관리자
정책을 사용하여 IAM 역할을 생성합니다.	IAM 콘솔의 탐색 창에서 역할(Roles)을 선택하고 역할 생성(Create role)을 선택합니다. 목	클라우드 관리자

작업	설명	필요한 기술
	<p>목록에서 서비스와 EC2를 차례대로 선택합니다.</p> <p>다음: 권한을 선택하고, 사용자가 이전에 생성한 IAM 정책을 검색합니다. 요구 사항에 맞는 적절한 태그를 선택합니다.</p> <p>검토 섹션에서 역할의 이름을 입력합니다. 역할 이름에는 role-eks-instance-profile-for-cloud9 을(를) 사용하는 것이 좋습니다. 그런 다음 역할 생성을 선택합니다.</p>	

Amazon EKS RBAC에 대한 IAM 정책 및 역할을 생성합니다.

작업	설명	필요한 기술
IAM 정책을 생성합니다.	<p>IAM 콘솔에서 정책을 선택하고 정책 생성을 선택합니다. JSON 탭을 policy-for-eks-rbac 선택하고.json 파일 (첨부) 의 내용을 붙여넣습니다.</p> <p>정책 검증 동안 생성된 모든 보안 경고, 오류 또는 일반 경고를 해결한 다음 정책 검토(Review policy)를 선택합니다. 정책의 이름을 입력합니다. 정책 이름으로 policy-for-eks-rbac 을(를) 사용하는 것이 좋습니다. 정책 요약을 검토하여 정책이 부여한 권한을 확인합니다.</p>	클라우드 관리자

작업	설명	필요한 기술
	그런 다음 정책 생성을 선택합니다.	
정책을 사용하여 IAM 역할을 생성합니다.	<p>IAM 콘솔의 탐색 창에서 역할 (Roles)을 선택하고 역할 생성 (Create role)을 선택합니다. 목록에서 서비스와 EC2를 차례대로 선택합니다. 다음: 권한을 선택하고, 사용자가 이전에 생성한 IAM 정책을 검색합니다. 요구 사항에 맞는 적절한 태그를 선택합니다.</p> <p>검토 섹션에서 역할의 이름을 입력합니다. 역할 이름에는 <code>role-eks-admin-for-rbac</code>(를) 사용하는 것이 좋습니다. 그런 다음 역할 생성을 선택합니다.</p>	클라우드 관리자

AWS Cloud9 환경 생성

작업	설명	필요한 기술
AWS Cloud9 환경을 생성합니다.	AWS Cloud9 콘솔을 열고 환경 생성을 선택합니다. 환경 이름 페이지의 [이름]에 환경의 이름을 입력합니다. 환경 이름에 대해 <code>eks-management-env</code> (을)를 사용하는 것이 좋습니다. 요구 사항에 따라 나머지 설정을 구성한 다음, 다음 단계를 선택합니다.	클라우드 관리자

작업	설명	필요한 기술
	<p>Review(검토) 페이지에서 Create environment(환경 생성)를 선택합니다. AWS Cloud9이 환경을 생성하는 동안 기다립니다. 몇 분 정도 걸릴 수 있습니다.</p> <p>사용 가능한 구성 옵션에 대한 자세한 내용은 AWS Cloud9 설명서에서 EC2 환경 생성하기를 참조하십시오.</p>	
<p>AWS Cloud9의 임시 IAM 자격 증명을 제거합니다.</p>	<p>AWS Cloud9 환경을 프로비저닝한 후 기어 모양 아이콘에서 설정을 선택합니다. 기본 설정에서 설정을 선택한 다음 자격 증명을 선택합니다.</p> <p>관리형 임시 자격 증명을 비활성화하고 탭을 닫습니다.</p>	<p>클라우드 관리자</p>

작업	설명	필요한 기술
EC2 인스턴스 프로파일을 기본 EC2 인스턴스에 연결합니다.	<p>Amazon EC2 콘솔을 열고, AWS Cloud9의 환경에 맞는 EC2 인스턴스를 선택합니다. 권장하는 이름을 사용한 경우 EC2 인스턴스가 <code>aws-cloud9-eks-management-environment</code> 호출됩니다.</p> <p>EC2 인스턴스를 선택하고 작업을 선택한 다음 인스턴스 설정을 선택합니다. IAM 역할 연결/교체를 선택합니다. <code>role-eks-instance-profile-for-cloud9</code> 또는 이전에 생성한 IAM 역할의 이름을 검색한 다음 적용을 선택합니다.</p>	클라우드 관리자

Amazon EKS 클러스터 생성

작업	설명	필요한 기술
Amazon EKS 클러스터를 생성합니다.	<p>AWS용 <code>eks-cfn.yaml</code> (첨부) 템플릿을 다운로드하여 엽니다. CloudFormation 요구 사항에 따라 템플릿을 편집합니다.</p> <p>AWS Cloud9 환경을 열고 새 파일을 선택합니다. 이전에 생성한 AWS CloudFormation 템플릿을 필드에 붙여넣습니다. 템플릿 이름에는 <code>eks-cfn.yaml</code> 을 사용하는 것이 좋습니다.</p>	클라우드 관리자

작업	설명	필요한 기술
	<p>AWS Cloud9 터미널에서, 다음 명령을 실행하여 Amazon EKS 클러스터를 생성합니다.</p> <pre>aws cloudformation create-stack -- stack-name eks-clust er --template-body file://eks-cfn.yam l --region <your_AWS _Region></pre> <p>AWS CloudFormation 호출이 성공하면 출력에서 AWS CloudFormation 스택의 Amazon 리소스 이름 (ARN) 을 받게 됩니다. 스택 생성에는 10~20분이 소요될 수 있습니다.</p>	
<p>Amazon EKS 클러스터의 상태를 확인합니다.</p>	<p>AWS CloudFormation 콘솔에서 Stacks 페이지를 연 다음 스택 이름을 선택합니다.</p> <p>스택 상태 코드에 CREATE_COMPLETE (이)가 나타날 때 스택이 생성됩니다. 자세한 내용은 AWS CloudFormation 설명서의 AWS CloudFormation 스택 데이터 및 리소스 보기를 참조 하십시오.</p>	<p>클라우드 관리자</p>

Amazon EKS 클러스터의 Kubernetes 리소스에 액세스합니다.

작업	설명	필요한 기술
AWS Cloud9 환경에 kubectl을 설치합니다.	Amazon EKS 설명서의 kubectl 설치 에 나와 있는 지침에 따라 AWS Cloud9 환경에서 kubectl(을)를 설치합니다.	클라우드 관리자
AWS Cloud9에서 새로운 Amazon EKS 구성을 업데이트합니다.	<p>AWS Cloud9 터미널에서 다음 명령을 실행하여 Amazon EKS 클러스터에서 AWS Cloud9 환경으로 kubeconfig (을)를 업데이트합니다.</p> <pre>aws eks update-kubeconfig --name EKS-DEV2 --region <your_AWS_Region></pre> <p>중요: EKS-DEV2 클러스터를 생성하는 데 사용한 AWS CloudFormation 템플릿의 Amazon EKS 클러스터 이름입니다.</p> <p>kubectl get all -A 명령을 실행하여 모든 Kubernetes 리소스를 확인합니다.</p>	클라우드 관리자
관리자 IAM 역할을 Kubernetes RBAC에 추가합니다.	<p>AWS Cloud9 터미널에서 다음 명령을 실행하여, 편집 모드에서 Amazon EKS의 RBAC 구성 맵을 엽니다.</p> <pre>kubectl edit cm/aws-auth -n kube-system</pre>	클라우드 관리자

작업	설명	필요한 기술
	<p>mapRoles 섹션에서 다음 줄을 추가합니다.</p> <pre data-bbox="594 331 1027 611">- groups: - system:masters rolearn: <ARN_of_IAM_role_from_second_epic> username: eksadmin</pre> <p>YAML 형식 파일을 린트하여 구문 오류를 방지합니다. vi 명령을 사용하여 파일을 저장한 다음 파일을 종료합니다.</p> <p>참고: 이 섹션을 추가하면, <ARN_of_IAM_role_from_second_epic> (이)가 Amazon EKS 클러스터에 대한 완전한 관리자 액세스 권한을 받게 됨을 Kubernetes RBAC에 알리게 됩니다. 즉, 식별된 IAM 역할은 Kubernetes 클러스터에 대한 관리 작업을 수행할 수 있습니다. AWS는 Amazon EKS 클러스터가 프로비저닝되는 동안 mapRoles에 기존 섹션을 추가합니다.</p>	

관련 리소스

참조

- [모듈식 및 확장 가능한 Amazon EKS 아키텍처\(Quick Start\)](#)
- [Amazon EKS 클러스터에 대한 사용자 또는 IAM 역할 관리](#)

- [새 Amazon EKS 컨트롤 플레인을 생성하기 위한 AWS CloudFormation 템플릿](#)

첨부

이 문서와 관련된 추가 콘텐츠에 액세스하려면 [attachment.zip](#) 파일의 압축을 풉니다.

AWS CodePipeline, AWS 및 AWS를 사용하여 여러 AWS 지역에 코드를 배포합니다. CodeCommit CodeBuild

작성자: Rama Anand Krishna Varanasi(AWS)

작성자: AWS

환경: PoC 또는 파일럿

기술: 관리 및 거버넌스, DevOps

AWS 서비스: AWS CodeCommit, AWS CodePipeline, AWS CodeBuild

요약

이 패턴은 AWS를 사용하여 여러 Amazon Web Services (AWS) 지역에 인프라 또는 아키텍처를 구축하는 방법을 보여줍니다. CloudFormation 배포 속도를 높이기 위해 여러 AWS 리전에 걸친 지속적 통합(CI) / 지속적 배포(CD)가 포함됩니다. 이 패턴의 단계는 예를 들어 세 개의 AWS 지역에 배포할 AWS CodePipeline 작업을 생성하기 위한 테스트를 거쳤습니다. 사용 사례를 바탕으로 리전의 수를 변경할 수 있습니다.

사전 조건 및 제한 사항

사전 조건

- 활성 상태의 AWS 계정
- 테스트, 번들링, 아티팩트 패키징, 여러 AWS 지역에 병렬로 CodeBuild 배포하는 등의 CI 작업을 수행하기 위한 적절한 정책이 CloudFormation 적용된 AWS CodeBuild 및 AWS용 AWS Identity 및 Access Management (IAM) 역할 2개 참고: 에서 생성한 정책을 CodePipeline 검토하여 AWS가 CI CodeBuild 및 CD 단계에서 적절한 권한을 CloudFormation 가지고 있는지 확인하십시오.
 - FullAccessAmazonS3 및 정책 관련 CodeBuild 역할. CloudWatchFullAccess 이러한 정책을 통해 Amazon을 CodeCommit 통해 AWS의 이벤트를 CloudWatch 시청하고 Amazon Simple Storage Service (Amazon S3) 를 아티팩트 스토어로 사용할 수 있습니다. CodeBuild
 - 최종 빌드 단계에서 AWS에 AWS CloudFormation Lambda 함수를 생성 또는 업데이트하고, CloudWatch Amazon 로그를 푸시하거나 감시하고, 변경 세트를 생성 및 업데이트할 수 있는 권한을 부여하는 다음 정책이 적용된 AWS CloudFormation 역할

- AWSLambdaFullAccess
- AWSCodeDeployFullAccess
- CloudWatchFullAccess
- AWSCloudFormationFullAccess
- AWSCodePipelineFullAccess

아키텍처

이 패턴의 다중 리전 아키텍처 및 워크플로는 다음 단계로 구성됩니다.

1. 리포지토리로 코드를 전송합니다. CodeCommit
2. 코드 업데이트 또는 커밋을 받으면 이벤트를 CodeCommit 호출하고, CloudWatch 이벤트가 다시 CodePipeline 작업을 시작합니다.
3. CodePipeline 에서 처리하는 CI를 참여시킵니다. CodeBuild 다음과 같은 작업이 수행됩니다.
 - AWS CloudFormation 템플릿 테스트 (선택 사항)
 - 배포에 포함된 각 지역의 AWS CloudFormation 템플릿 패키징 예를 들어, 이 패턴은 세 개의 AWS 지역에 병렬로 배포되므로 AWS CloudFormation 템플릿을 지정된 각 지역에 하나씩 총 3개의 S3 버킷으로 CodeBuild 패키징합니다. 에서 S3 버킷은 아티팩트 CodeBuild 리포지토리로만 사용됩니다.
4. CodeBuild 아티팩트를 다음 배포 단계의 입력으로 패키징합니다. 이 단계는 세 개의 AWS 리전에서 병렬로 실행됩니다. 지역 수를 다르게 CodePipeline 지정하면 해당 지역에 배포됩니다.

도구

도구

- [AWS CodePipeline](#) — 소프트웨어 변경 사항을 지속적으로 릴리스하는 데 필요한 단계를 모델링, 시각화 및 자동화하는 데 사용할 수 있는 지속적 전송 CodePipeline 서비스입니다.
- [AWS CodeBuild](#) — 소스 코드를 컴파일하고, 단위 테스트를 실행하고, 배포 준비가 완료된 아티팩트를 생성하는 완전관리형 빌드 CodeBuild 서비스입니다.
- [AWS CodeCommit](#) — CodeCommit Amazon Web Services에서 호스팅하는 버전 제어 서비스로, 클라우드의 자산 (예: 소스 코드 및 바이너리 파일) 을 비공개로 저장하고 관리하는 데 사용할 수 있습니다.

- [AWS CloudFormation](#) — CloudFormation AWS는 Amazon Web Services 리소스를 모델링하고 설정하여 리소스를 관리하는 시간을 줄이고 AWS에서 실행되는 애플리케이션에 더 많은 시간을 집중할 수 있도록 지원하는 서비스입니다.
- [AWS Identity and Access Management](#) - AWS Identity and Access Management(IAM)는 AWS 리소스에 대한 사용자의 액세스를 안전하게 제어하는 데 도움이 되는 웹 서비스입니다.
- [Amazon S3](#) – Amazon Simple Storage Service(S3)는 인터넷에 대한 스토리지입니다. 이 서비스는 개발자가 더 쉽게 웹 규모 컴퓨팅 작업을 수행할 수 있도록 설계되었습니다.

코드

다음 샘플 코드는 BuildSpec.yaml 파일용 샘플 코드입니다(빌드 단계).

```

---
artifacts:
  discard-paths: true
  files:
  - packaged-first-region.yaml
  - packaged-second-region.yaml
  - packaged-third-region.yaml
  phases:
  build:
  commands:
  - echo "*****BUILD PHASE - CF PACKAGING*****"
  - "aws cloudformation package --template-file sam-template.yaml --s3-bucket
    $S3_FIRST_REGION --output-template-file packaged-first-region.yaml --region
    $FIRST_REGION"
  - "aws cloudformation package --template-file sam-template.yaml --s3-bucket
    $S3_SECOND_REGION --output-template-file packaged-second-region.yaml --region
    $SECOND_REGION"
  - "aws cloudformation package --template-file sam-template-anand.yaml --s3-bucket
    $S3_THIRD_REGION --output-template-file packaged-third-region.yaml --region
    $THIRD_REGION"
  install:
  commands:
  - echo "*****BUILD PHASE - PYTHON SETUP*****"
  runtime-versions:
  python: 3.8
  post_build:
  commands:
  - echo "*****BUILD PHASE - PACKAGING COMPLETION*****"
  pre_build:

```

```

commands:
- echo "*****BUILD PHASE - DEPENDENCY SETUP*****"
- "npm install --silent --no-progress"
- echo "*****BUILD PHASE - DEPENDENCY SETUP DONE*****"
version: 0.2
    
```

에픽

코드와 CodeCommit 리포지토리를 준비하십시오.

작업	설명	필요한 기술
배포할 기본 AWS 리전을 선택합니다.	AWS 계정에 로그인하고 배포할 기본 리전을 선택합니다. CodeCommit 리포지토리는 기본 지역에 위치합니다.	DevOps
CodeCommit 리포지토리를 생성합니다.	CodeCommit 리포지토리를 만들고 필요한 코드를 해당 리포지토리에 푸시합니다. 코드에는 일반적으로 AWS CloudFormation 또는 AWS SAM 템플릿, Lambda 코드 (있는 경우), AWS에 CodeBuild buildspec.yaml 입력되는 파일이 포함됩니다. CodePipeline	DevOps
코드를 리포지토리로 푸시합니다. CodeCommit	첨부 섹션에서 이 예제의 코드를 다운로드한 다음 필요한 코드를 여기에 푸시합니다. 일반적으로 코드에는 AWS CloudFormation 또는 AWS SAM 템플릿, Lambda 코드 및 파일을 파이프라인에 CodeBuild buildspec.yaml 대한 입력으로 포함할 수 있습니다.	DevOps

소스 단계: 파이프라인 생성

작업	설명	필요한 기술
작업을 생성합니다. CodePipeline	CodePipeline 콘솔에서 파이프라인 생성을 선택합니다.	DevOps
작업을 생성합니다. CodePipeline	작업을 생성합니다. CodePipeline 콘솔에서 파이프라인 생성을 선택합니다.	DevOps
CodePipeline 작업 이름을 지정하고 서비스 역할 설정을 선택합니다.	작업 이름을 입력하고 기본 서비스 역할 설정을 그대로 유지하여 필요한 정책이 첨부된 역할을 CodePipeline 생성합니다.	DevOps
아티팩트 스토어의 위치를 지정합니다.	고급 설정에서 기본 옵션을 유지하여 코드 아티팩트 스토리지에 사용할 S3 버킷을 CodePipeline 생성합니다. 기존 S3 버킷을 대신 사용하는 경우 버킷은 첫 번째 에픽에서 지정한 기본 리전에 있어야 합니다.	DevOps
암호화 키를 지정합니다.	기본 옵션인 기본 AWS 관리형 키를 사용하거나, 자체 AWS Key Management Service(AWS KMS) 고객 관리형 키를 사용하도록 선택합니다.	DevOps
소스 공급자를 지정합니다.	소스 공급자에서 AWS를 선택합니다 CodeCommit.	DevOps
리포지토리를 지정합니다.	첫 번째 에픽에서 만든 CodeCommit 리포지토리를 선택합니다. 브랜치에 코드를 삽입했다면 브랜치를 선택하십시오.	DevOps

작업	설명	필요한 기술
코드 변경을 감지하는 방법을 지정하십시오.	CodePipeline 작업을 시작하기 위한 변경 트리거로 기본값인 Amazon CloudWatch CodeCommit Events를 유지하십시오.	DevOps

빌드 단계: 파이프라인 구성

작업	설명	필요한 기술
빌드 제공자를 지정하십시오.	빌드 공급자의 경우 AWS를 선택합니다 CodeBuild.	DevOps
AWS 리전을 지정합니다.	첫 번째 에픽에서 지정한 기본 리전을 선택합니다.	DevOps

빌드 단계: 프로젝트 생성 및 구성

작업	설명	필요한 기술
프로젝트 생성	프로젝트 생성을 선택한 다음, 프로젝트의 이름을 입력합니다.	DevOps
환경 이미지를 지정하십시오.	이 패턴 데모에서는 기본 CodeBuild 관리형 이미지를 사용하십시오. 사용자 지정 도커 이미지가 있는 경우 이를 사용할 수도 있습니다.	DevOps
운영 체제를 지정하십시오.	Amazon Linux 2 또는 Ubuntu를 선택합니다.	DevOps
서비스 역할을 지정합니다.	CodePipeline 작업 생성을 CodeBuild 시작하기 전에 만든	DevOps

작업	설명	필요한 기술
	역할을 선택합니다. (사전 조건 섹션 참조)	
추가 옵션을 설정하십시오.	제한 시간 및 대기 중인 제한시간의 경우 기본값을 유지하십시오. 인증서의 경우 사용하려는 사용자 지정 인증서가 없는 한 기본 설정을 유지하십시오.	DevOps
환경 변수를 생성합니다.	배포하려는 각 AWS 리전에 대해 S3 버킷 이름과 리전 이름 (예를 들어, 미국 동부-1)을 제공하여 환경 변수를 생성합니다.	DevOps
buildspec.yml이 아닌 경우 buildspec 파일 이름을 입력하십시오.	파일 이름이 기본값인 경우 이 필드를 비워 두세요, buildspec.yaml . buildspec 파일의 이름을 변경한 경우 여기에 이름을 입력하십시오. CodeCommit 저장소에 있는 파일 이름과 일치하는지 확인하십시오.	DevOps
로깅을 지정합니다.	Amazon CloudWatch Events의 로그를 보려면 기본 설정을 유지하십시오. 또는 특정 그룹 또는 로거 이름을 정의할 수 있습니다.	DevOps

배포 단계를 건너뛰십시오.

작업	설명	필요한 기술
배포 단계를 건너뛰고 파이프라인 생성을 완료하십시오.	파이프라인을 설정하면 배포 단계에서 한 단계만 생성할 CodePipeline 수 있습니다. 여러 AWS 리전에 배포하려면 이 단계를 건너뛰십시오. 파이프라인이 생성된 후 여러 배포 단계를 추가할 수 있습니다.	DevOps

배포 단계: 첫 번째 리전에 배포할 파이프라인을 구성합니다.

작업	설명	필요한 기술
배포 단계에 스테이지를 추가합니다.	파이프라인을 편집하고 배포 단계에서 스테이지 추가를 선택합니다. 이 첫 번째 스테이지는 기본 리전을 위한 것입니다.	DevOps
스테이지의 액션 이름을 입력합니다.	첫 번째 (기본) 스테이지와 리전을 반영하는 고유한 이름을 입력합니다. 예를 들어, primary_<region>_deploy를 입력합니다.	DevOps
작업 제공자를 지정하십시오.	액션 프로바이더에서는 AWS를 선택합니다 CloudFormation.	DevOps
첫 번째 스테이지를 위한 리전을 구성합니다.	첫 번째 (기본) 지역, CodePipeline 및 CodeBuild가 설정된 동일한 지역을 선택합니다. 스택을 배포할 기본 리전입니다.	DevOps

작업	설명	필요한 기술
입력 아티팩트를 지정합니다.	선택하세요 BuildArtifact. 빌드 단계의 출력입니다.	DevOps
취할 작업을 지정합니다.	작업 모드에서 스택 생성 또는 업데이트를 선택합니다.	DevOps
CloudFormation 스택 이름을 입력합니다.		DevOps
첫 번째 리전의 템플릿을 지정합니다.	첫 번째 (기본) 지역의 S3 버킷에서 CodeBuild 패키징하여 덤포한 지역별 패키지 이름을 선택합니다.	DevOps
기능을 지정합니다.	스택 템플릿에 IAM 리소스가 포함되어 있거나 매크로가 포함된 템플릿에서 직접 스택을 생성하는 경우 기능이 필요합니다. 이 패턴에는 CAPABILITY_IAM, CAPABILITY_NAMED_IAM, CAPABILITY_AUTO_EXPAND를 사용합니다.	DevOps

배포 단계: 두 번째 리전에 배포할 파이프라인을 구성합니다.

작업	설명	필요한 기술
배포 단계에 두 번째 스테이지를 추가합니다.	두 번째 리전에 스테이지를 추가하려면 파이프라인을 편집하고 배포 단계에서 스테이지 추가를 선택합니다. 중요: 두 번째 리전을 만드는 과정은 다음 값을 제외하고 첫 번째 리전을 만드는 프로세스와 동일합니다.	DevOps

작업	설명	필요한 기술
두 번째 스테이지의 액션 이름을 입력합니다.	두 번째 스테이지와 두 번째 리전을 반영하는 고유한 이름을 입력합니다.	DevOps
두 번째 스테이지를 위한 리전을 구성합니다.	스택을 배포하려는 두 번째 리전을 선택합니다.	DevOps
두 번째 리전의 템플릿을 지정합니다.	두 번째 지역의 경우 패키징하여 S3 버킷에 덤프한 지역별 CodeBuild 패키지 이름을 선택합니다.	DevOps

배포 단계: 세 번째 리전에 배포할 파이프라인을 구성합니다.

작업	설명	필요한 기술
배포 단계에 세 번째 스테이지를 추가합니다.	세 번째 리전에 스테이지를 추가하려면 파이프라인을 편집하고 배포 단계에서 스테이지 추가를 선택합니다. 중요: 세 번째 리전을 만드는 과정은 다음 값을 제외하고 이전 두 개 리전을 만드는 프로세스와 동일합니다.	DevOps
세 번째 스테이지의 액션 이름을 입력합니다.	세 번째 스테이지와 세 번째 리전을 반영하는 고유한 이름을 입력합니다.	DevOps
세 번째 스테이지를 위한 리전을 구성합니다.	스택을 배포하려는 세 번째 리전을 선택합니다.	DevOps
세 번째 리전의 템플릿을 지정합니다.	패키징하여 세 번째 지역의 S3 버킷에 덤프한 지역별	DevOps

작업	설명	필요한 기술
	CodeBuild 패키지 이름을 선택합니다.	

배포 정리

작업	설명	필요한 기술
AWS 리소스를 삭제합니다.	배포를 정리하려면 각 지역의 CloudFormation 스택을 삭제하십시오. 그런 다음 기본 지역에서 CodeCommit CodeBuild,, CodePipeline 리소스를 삭제합니다.	DevOps

관련 리소스

- [AWS란 CodePipeline 무엇입니까?](#)
- [AWS Serverless Application Model](#)
- [AWS CloudFormation](#)
- [AWS용 AWS CloudFormation 아키텍처 구조 참조 CodePipeline](#)

첨부

이 문서와 관련된 추가 콘텐츠에 액세스하려면 [attachment.zip](#) 파일의 압축을 풉니다.

AWS Organizations의 조직 전체에서 AWS Backup 보고서를 CSV 파일로 내보내기

작성자: Aromal Raj Jayarajan(AWS)와 Purushotham G K(AWS)

코드 리포지토리: aws-backup-report-generator	환경: PoC 또는 파일럿	기술: DevOps; 인프라
워크로드: 기타 모든 워크로드	AWS 서비스: AWS 백업, AWS 자격 증명 및 액세스 관리, AWS Lambda, 아마존 S3, 아마존 EventBridge	

요약

이 패턴은 AWS Organizations의 조직 전체에서 AWS Backup 작업 보고서를 CSV 파일로 내보내는 방법을 보여줍니다. 이 솔루션은 AWS Lambda와 EventBridge Amazon을 사용하여 AWS Backup 작업 보고서를 상태에 따라 분류하므로 상태 기반 자동화를 구성할 때 도움이 될 수 있습니다.

AWS Backup은 조직이 AWS 서비스 전체, 클라우드 및 온프레미스에서 데이터 보호를 중앙에서 관리하고 자동화하는 데 도움을 줍니다. 하지만 AWS Organizations 내에 구성된 AWS Backup 작업의 경우 통합 보고는 각 조직의 관리 계정의 AWS Management Console에서만 사용할 수 있습니다. 이 보고를 관리 계정 외부로 가져오면 감사에 필요한 노력을 줄이고 자동화, 알림 및 경보의 범위를 늘릴 수 있습니다.

사전 조건 및 제한 사항

사전 조건

- 활성 상태의 AWS 계정
- 하나 이상의 관리 계정과 멤버 계정을 포함하는 AWS Organizations에서 활성 [조직](#)
- AWS Organizations의 조직 수준에서 구성된 AWS 백업 (자세한 내용은 [AWS 블로그의 AWS Backup을 사용하여 AWS 서비스 전반에 걸쳐 대규모 중앙 집중식 백업 자동화](#)를 참조)
- [Git](#), 로컬 머신에 설치 및 구성됨

제한 사항

이 패턴으로 제공된 솔루션은 AWS Backup 작업에만 구성된 AWS 리소스를 식별합니다. 보고서에서는 AWS Backup을 통해 백업하도록 구성되지 않은 AWS 리소스를 식별할 수 없습니다.

아키텍처

대상 기술 스택

- AWS Backup
- AWS CloudFormation
- 아마존 EventBridge
- AWS Lambda
- AWS Security Token Service (AWS STS)
- Amazon Simple Storage Service(S3)
- AWS Identity and Access Management(IAM)

대상 아키텍처

다음 다이어그램은 AWS Organizations의 조직 전체에서 AWS Backup 작업 보고서를 CSV 파일로 내보내는 예제 워크플로우를 보여줍니다.

이 다이어그램은 다음 워크플로를 보여줍니다.

1. 예약된 EventBridge 이벤트 규칙은 멤버 (보고) AWS 계정에서 Lambda 함수를 호출합니다.
2. 그런 다음 Lambda 함수는 AWS STS를 사용하여 관리 계정에 연결하는 데 필요한 권한이 있는 IAM 역할을 수임합니다.
3. 그런 다음 Lambda 함수는 다음 작업을 수행합니다.

- AWS 백업 서비스에 통합된 AWS 백업 작업 보고서 요청
- AWS Backup 작업 상태를 기반으로 결과를 분류
- 응답을 CSV 파일로 변환
- 생성 날짜를 기준으로 레이블이 지정된 폴더 내의 보고 계정의 Amazon S3 버킷에 결과를 업로드

도구

도구

- [AWS Backup](#)은 AWS 서비스 전체, 클라우드 및 온프레미스에서 데이터 보호를 중앙 집중화하고 자동화하는 데 도움을 주는 완전관리형 서비스입니다.
- [AWS는 AWS CloudFormation](#) 리소스를 설정하고, 빠르고 일관되게 프로비저닝하고, AWS 계정 및 지역 전반의 수명 주기 전반에 걸쳐 리소스를 관리할 수 있도록 지원합니다.
- [EventBridgeAmazon](#)은 다양한 소스의 실시간 데이터와 애플리케이션을 연결하는 데 도움이 되는 서버리스 이벤트 버스 서비스입니다. AWS Lambda 함수, API 대상을 사용하는 HTTP 간접 호출 엔드포인트 또는 다른 AWS 계정의 이벤트 버스를 예로 들 수 있습니다.
- [AWS Identity and Access Management\(IAM\)](#)를 사용하면 사용자에 대해 인증 및 권한 부여를 제어함으로써 AWS 리소스에 대한 액세스를 안전하게 관리할 수 있습니다.
- [AWS Lambda](#)는 서버를 프로비저닝하거나 관리할 필요 없이 코드를 실행하는 데 도움이 되는 컴퓨팅 서비스입니다. 필요할 때만 코드를 실행하며 자동으로 확장이 가능하므로 사용한 컴퓨팅 시간만큼만 비용을 지불합니다.
- [Amazon Simple Storage Service\(Amazon S3\)](#)는 원하는 양의 데이터를 저장, 보호 및 검색하는 데 도움이 되는 클라우드 기반 객체 스토리지 서비스입니다.

코드

이 패턴의 코드는 GitHub [aws-backup-report-generator](#) 리포지토리에서 사용할 수 있습니다.

모범 사례

- [Amazon S3의 보안 모범 사례](#)(Amazon S3 사용 설명서)
- [AWS Lambda 함수를 사용한 모범 사례](#)(AWS Lambda 개발자 가이드)
- [관리 계정 모범 사례](#)(AWS Organizations 사용 설명서)

에픽

솔루션 구성 요소 배포

작업	설명	필요한 기술
<p>GitHub 리포지토리를 복제합니다.</p>	<p>터미널 창에서 다음 명령을 실행하여 GitHub aws-backup-report-generator 리포지토리를 복제합니다.</p> <pre data-bbox="594 646 1029 848">git clone https://github.com/aws-samples/aws-backup-report-generator.git</pre> <p>자세한 내용은 GitHub 문서에서 리포지토리 복제를 참조하십시오.</p>	<p>AWS DevOps, DevOps 엔지니어</p>
<p>멤버(보고) AWS 계정에 솔루션 구성 요소를 배포합니다.</p>	<ol style="list-style-type: none"> 1. 회원 (신고) 계정에서 AWS Management 콘솔에 로그인한 다음 CloudFormation 콘솔을 엽니다. 2. 스택 생성을 선택한 다음 새 리소스 사용(표준)을 선택합니다. 3. 스택 생성 페이지, 템플릿 지정 섹션에서 템플릿 파일 업로드를 선택합니다. 4. 파일 선택을 선택합니다. 그런 다음 로컬 워크스테이션에 있는 복제된 GitHub 리포지토리의 루트 폴더로 이동하여 <code>template-reporting.yaml</code>을 선택합니다. 	<p>DevOps 엔지니어, AWS DevOps</p>

작업	설명	필요한 기술
	5. 열기를 선택한 후 다음을 선택합니다. 6. 스택 세부 정보 지정 페이지에서 스택 이름에 스택 이름을 입력합니다. CloudFormation 7. ManagementAccountID에는 AWS Organizations에 있는 조직의 관리 계정에 대한 AWS 계정 ID를 입력합니다. 8. 다음을 선택합니다. 9. 스택 옵션 구성 페이지에서 다음을 선택합니다. 10. 검토 페이지에서 확인란을 선택하여 구성을 검토했음을 확인합니다. 11. 스택 생성을 선택합니다. 멤버(보고) 계정에 솔루션 구성 요소를 배포하면 스택에는 CREATE_COMPLETE 상태가 표시됩니다.	

솔루션 테스트

작업	설명	필요한 기술
테스트 전에 EventBridge 규칙이 실행되는지 확인하십시오.	최소 24시간 이상 기다리거나 CloudFormation 템플릿의 template-reporting.yml 파일에서 신고 빈도를 늘려 EventBridge 규칙이 실행되는지 확인하십시오.	AWS DevOps, DevOps 엔지니어

작업	설명	필요한 기술
	<p>보고 빈도를 높이려면</p> <ol style="list-style-type: none"> 1. 복제된 리포포지토리에서 <code>template-reporting.yml</code> 파일을 엽니다. 2. 논리적 ID가 "인 이벤트 규칙에서" '을 찾으십시오. <code>LambdaScheduleScheduleExpression</code> 3. 유효한 cron 표현식이 포함 되도록 <code>ScheduleExpression</code> " 키를 편집하십시오. 예를 들어, 다음 cron 표현식 "<code>cron (* /5 * * *</code>"은(는) 5분마다 실행되도록 이벤트 규칙을 예약합니다. 	

작업	설명	필요한 기술
생성된 보고서에 대한 Amazon S3 버킷을 확인합니다.	<ol style="list-style-type: none"> 1. 회원 (신고) 계정에서 AWS Management 콘솔에 로그인한 다음 CloudFormation 콘솔을 엽니다. 2. 스택 창에서 생성한 스택의 이름을 선택합니다. 그런 다음 리소스 탭을 선택합니다. 3. 리소스 창의 논리 ID 열에서 BackupReportS3Bucket을 찾으십시오. 그런 다음 해당 논리적 ID 옆에 있는 물리적 ID 열에서 링크를 선택하여 새 탭에서 연결된 Amazon S3 버킷을 엽니다. 4. <Mon><yyyy>버킷에 다음 형식으로 생성된 보고서가 포함되어 있는지 확인하십시오. BackupReports///BackupReport- - .csv <yyyy><mm><dd><BACKUP JOB STATUS><dd> 	AWS DevOps, DevOps 엔지니어

리소스 정리

작업	설명	필요한 기술
멤버(보고) 계정에서 솔루션 구성 요소를 삭제합니다.	<ol style="list-style-type: none"> 1. 멤버(보고) 계정에서 솔루션의 Amazon S3 버킷을 엽니다. 지침은 이 패턴의 솔루션 테스트 섹션에서 생성된 보고서에 대한 S3 버킷 확인 스토리의 2-4단계를 참조하세요. 	AWS DevOps, DevOps 엔지니어

작업	설명	필요한 기술
	<ol style="list-style-type: none"> 2. 버킷의 콘텐츠를 삭제하고 버킷을 비웁니다. 지침은 Amazon S3 사용 설명서의 버킷 비우기를 참조하세요. 3. 회원 (신고) 계정에서 AWS Management 콘솔에 로그인한 다음 CloudFormation 콘솔을 엽니다. 4. 스택 창에서 생성한 스택 이름 옆의 확인란을 선택합니다. 그런 다음 삭제를 선택합니다. 	
<p>관리 계정에서 솔루션 구성 요소를 삭제합니다.</p>	<ol style="list-style-type: none"> 1. 관리 계정에서 AWS Management 콘솔에 로그인한 다음 CloudFormation 콘솔을 엽니다. 2. 스택 창에서 생성한 스택 이름 옆의 확인란을 선택합니다. 그런 다음 삭제를 선택합니다. 	<p>AWS DevOps, DevOps 엔지니어</p>

관련 리소스

- [자습서: 예정된 이벤트와 함께 AWS Lambda 사용](#) (AWS Lambda 설명서)
- [AWS Lambda 함수를 실행하기 위한 예약 이벤트 생성](#) (설명서용 AWS SDK) JavaScript
- [IAM 자습서: IAM 역할을 사용하여 AWS 계정 전체에 대한 액세스 위임](#) (IAM 설명서)
- [AWS Organizations 용어 및 개념](#) (AWS Organizations 설명서)
- [AWS 백업 콘솔을 사용하여 보고서 계획 생성](#) (AWS 백업 설명서)
- [감사 보고서 생성](#) (AWS 백업 설명서)
- [온디맨드 보고서 생성](#) (AWS 백업 설명서)
- [AWS Backup이란 무엇입니까?](#) (AWS 백업 설명서)

- AWS Backup을 [사용하여 AWS 서비스 전반에서 대규모 중앙 집중식 백업을 자동화하십시오](#) (AWS 블로그 게시물)

Amazon EC2 인스턴스 목록의 태그를 CSV 파일로 내보내기

작성자: Sida Ju(AWS) 및 Pac Joonhyun(AWS)

코드 리포지토리: [EC2 태그 검색 및 내보내기](#)

환경: 프로덕션

기술: DevOps

AWS 서비스: Amazon EC2

요약

이 패턴은 프로그래밍 방식으로 Amazon Elastic Compute Cloud(Amazon EC2) 인스턴스 목록의 태그를 CSV 파일로 내보내는 방법을 보여줍니다.

제공된 예제 Python 스크립트를 사용하면 Amazon EC2 인스턴스를 검토하고 특정 태그별로 분류하는데 걸리는 시간을 줄일 수 있습니다. 예를 들어, 이 스크립트를 사용하여 보안팀이 소프트웨어 업데이트를 위해 신고한 인스턴스 목록을 빠르게 식별하고 분류할 수 있습니다.

사전 조건 및 제한 사항

사전 조건

- Python 3 설치 및 구성
- AWS Command Line Interface(AWS CLI) 설치 및 구성

제한 사항

이 패턴에 제공된 예제 Python 스크립트는 다음 속성만을 기반으로 Amazon EC2 인스턴스를 검색할 수 있습니다.

- 인스턴스 ID
- 프라이빗 IPv4 주소
- 퍼블릭 IPv4 주소

도구

- [Python](#)은 범용 컴퓨터 프로그래밍 언어입니다.
- [virtualenv](#)를 사용하면 격리된 Python 환경을 생성할 수 있습니다.
- [AWS Command Line Interface\(AWS CLI\)](#)는 명령줄 셸에서 명령을 사용하여 AWS 서비스와 상호 작용할 수 있는 오픈 소스 도구입니다.

코드 리포지토리

이 패턴에 대한 예제 Python 스크립트는 GitHub [search-ec2](#) - 저장소에서 사용할 수 있습니다.
instances-export-tags

에픽

사전 조건 설치 및 구성

작업	설명	필요한 기술
리포지토리를 복제합니다. GitHub	<p>참고: AWS CLI 명령을 실행할 때 오류가 발생하는 경우 최신 AWS CLI 버전을 사용하고 있는지 확인하십시오.</p> <p>터미널 창에서 다음 Git 명령을 실행하여 GitHub search-ec2-instances-export-tags 리포지토리를 복제합니다.</p> <pre>git clone https://github.com/aws-samples/search-ec2-instances-export-tags.git</pre>	DevOps 엔지니어
가상 환경을 설치하고 활성화합니다.	1. 다음 명령을 실행하여 가상 환경을 설치합니다.	DevOps 엔지니어

작업	설명	필요한 기술
	<pre>python3 -m pip install virtualenv</pre> <p>2. 다음 명령을 실행하여 새 가상 환경을 생성합니다.</p> <pre>python3 -m venv env</pre> <p>3. AWS CLI에서 다음 명령을 실행하여 새 가상 환경을 활성화합니다.</p> <pre>source env/bin/activate</pre> <p>자세한 내용은 가상 환경 사용 설명서를 참조하십시오.</p>	
종속 항목 설치	<p>1. 터미널에서 다음 명령을 실행하여 코드 디렉터리를 엽니다.</p> <pre>cd search-ec2-instances-export-tags</pre> <p>2. 다음 pip 명령을 실행하여 requirements.txt 파일을 설치합니다.</p> <pre>pip3 install -r requirements.txt</pre>	DevOps 엔지니어

작업	설명	필요한 기술
AWS 명명된 프로파일을 구성합니다.	<p>아직 구성하지 않았다면 스크립트 실행에 필요한 자격 증명에 포함된 AWS 명명된 프로파일을 구성합니다. 명명된 프로파일을 생성하려면 aws configure 명령을 실행합니다.</p> <p>자세한 내용은 AWS CLI 설명서에서 네임드 프로파일 사용을 참조하세요.</p>	DevOps 엔지니어

Python 스크립트 구성 및 실행

작업	설명	필요한 기술
입력 파일을 생성합니다.	<p>스크립트에서 태그를 검색하고 내보낼 Amazon EC2 인스턴스 목록이 포함된 입력 파일을 생성합니다. 인스턴스 ID, 프라이빗 IPv4 주소 또는 퍼블릭 IPv4 주소를 나열할 수 있습니다.</p> <p>중요: 각 Amazon EC2 인스턴스가 입력 파일의 한 줄에 나열되어 있는지 확인하십시오.</p> <p>입력 파일 예제</p> <pre> 1 i-0547c351bdfe85b9 f 2 54.157.194.156 3 172.31.85.33 4 54.165.198.144 5 i-0b6223b5914111a4 b 6 172.31.85.44 </pre>	DevOps 엔지니어

작업	설명	필요한 기술
	<pre>7 54.165.198.145 8 172.31.80.219 9 172.31.94.199</pre>	
<p>Python 스크립트를 실행합니다.</p>	<p>터미널에서 다음 명령을 실행하여 스크립트를 실행합니다.</p> <pre>python search_in stances.py -i INPUTFILE -o OUTPUTFIL E -r REGION [-p PROFILE]</pre> <p>참고: INPUTFILE 을 입력 파일의 이름으로 바꿉니다. OUTPUTFILE 을 CSV 출력 파일에 부여할 이름으로 바꿉니다. REGION을 Amazon EC2 리소스가 있는 AWS 리전으로 바꿉니다. AWS 명명된 프로파일을 사용하는 경우 PROFILE을 사용 중인 명명된 프로파일로 바꾸십시오.</p> <p>지원되는 파라미터 및 설명 목록을 가져오려면 다음 명령을 실행합니다.</p> <pre>python search_in stances.py -h</pre> <p>자세한 내용과 출력 파일 예제를 보려면 GitHub search-ec2 - README.md 저장소의 파일을 참조하십시오. instances-export-tags</p>	<p>DevOps 엔지니어</p>

관련 리소스

- [AWS CLI 구성](#) (AWS CLI 사용 설명서)

트로포스피어를 사용하여 AWS Config 관리형 규칙이 포함된 AWS CloudFormation 템플릿을 생성합니다.

작성자: Lucas Nation(AWS) 및 Freddie Wilson(AWS)

환경: 프로덕션

기술: DevOps; 관리 및 거버넌스; 보안, ID, 규정 준수

워크로드: Microsoft, 오픈 소스

AWS 서비스: AWS Config, AWS CloudFormation

요약

많은 조직이 [AWS Config 관리형](#) 규칙을 사용하여 일반적인 모범 사례에 대한 Amazon Web Services(AWS) 리소스의 규정 준수를 평가합니다. 하지만 이러한 규칙은 유지 관리하는 데 시간이 많이 걸릴 수 있으며 이 패턴을 사용하면 Python 라이브러리인 [Troposphere](#)를 활용하여 AWS Config 관리형 규칙을 생성하고 관리할 수 있습니다.

이 패턴은 Python 스크립트를 사용하여 AWS 관리형 규칙이 포함된 Microsoft Excel 스프레드시트를 AWS 템플릿으로 변환함으로써 AWS Config 관리형 규칙을 관리하는 데 도움이 됩니다. CloudFormation Troposphere는 코드형 인프라(IAC) 역할을 하므로 JSON 또는 YAML 형식의 파일을 사용하는 대신 관리형 규칙을 사용하여 Excel 스프레드시트를 업데이트할 수 있습니다. 그런 다음 템플릿을 사용하여 AWS 계정에서 관리형 규칙을 생성하고 업데이트하는 AWS CloudFormation 스택을 시작합니다.

AWS CloudFormation 템플릿은 Excel 스프레드시트를 사용하여 각 AWS Config 관리 규칙을 정의하므로 AWS Management Console에서 개별 규칙을 수동으로 생성하지 않아도 됩니다. 스크립트는 각 관리형 규칙의 파라미터를 빈 딕셔너리로 기본 설정하고 범위의 기본값은 다음과 같습니다. `ComplianceResourceTypes.THE_RULE_IDENTIFIER.template file` 규칙 식별자에 대한 자세한 내용은 [AWS Config 설명서에서 AWS CloudFormation 템플릿을 사용하여 AWS Config 관리형 규칙 생성](#)을 참조하십시오.

사전 조건 및 제한 사항

사전 조건

- 활성 상태의 AWS 계정.

- AWS CloudFormation 템플릿을 사용하여 AWS Config 관리형 규칙을 생성하는 데 익숙해야 합니다. 이에 대한 자세한 내용은 [AWS Config 설명서에서 AWS CloudFormation 템플릿을 사용하여 AWS Config 관리형 규칙 생성을](#) 참조하십시오.
- 설치 및 구성된 Python 3. 이에 대한 자세한 내용은 [Python 설명서](#)를 참조하세요.
- AWS Cloud9와 같은 기존의 통합 개발 환경(IDE). 이에 대한 자세한 내용은 AWS Cloud9 설명서의 [AWS Cloud9란 무엇인가요?](#)를 참조하세요
- 첨부된 샘플 excel_config_rules.xlsx Excel 스프레드시트의 열에서 조직 단위(OU)를 식별합니다.

에픽

AWS Config 관리형 규칙을 사용자 지정하고 구성합니다.

작업	설명	필요한 기술
샘플 Excel 스프레드시트를 업데이트합니다.	<p>첨부된 샘플 excel_config_rules.xlsx Excel 스프레드시트와 레이블을 다운로드하여 사용하려는 AWS Config 관리형 규칙으로 Implemented 로 레이블을 지정합니다.</p> <p>로 표시된 규칙은 AWS CloudFormation 템플릿에 Implemented 추가됩니다.</p>	개발자
(선택 사항) AWS Config 규칙 파라미터를 사용하여 config_rules_params.json 파일을 업데이트합니다.	일부 AWS Config 관리형 규칙에는 파라미터가 필요하며 --param-file 옵션을 사용하여 Python 스크립트에 JSON 파일로 전달해야 합니다. 예를 들어, access-keys-rotated 관리형 규칙은 다음 maxAccessKeyAge 파라미터를 사용합니다.	개발자

작업	설명	필요한 기술
	<pre data-bbox="594 212 1024 646"> { "access-keys-rotated": { "InputParameters": { "maxAccessKeyAge": 90 } } } </pre> <p data-bbox="594 680 1013 957">이 샘플 파라미터에서는 <code>maxAccessKeyAge</code> 가 90일로 설정되어 있습니다. 스크립트는 파라미터 파일을 읽고 찾은 <code>InputParameters</code> 를 추가합니다.</p>	

작업	설명	필요한 기술
<p>(선택 사항) AWS Config를 사용하여 config_rules_parameters.json 파일을 업데이트하십시오. ComplianceResourceTypes</p>	<p>기본적으로 Python 스크립트는 AWS 정의 템플릿에서 ComplianceResourceTypes 를 검색합니다. 특정 AWS Config 관리형 규칙의 범위를 재정의하려면 --param-file 옵션을 사용하여 Python 스크립트에 JSON 파일로 전달해야 합니다.</p> <p>예를 들어, 다음 샘플 코드는 ec2-volume-inuse-check 를 위한 ComplianceResourceTypes 가 ["AWS::EC2::Volume"] 목록에 어떻게 설정되는지 보여줍니다.</p> <pre data-bbox="594 1050 1029 1608"> { "ec2-volume-inuse-check": { "Scope": { "ComplianceResourceTypes": ["AWS::EC2::Volume"] } } } </pre>	<p>개발자</p>

Python 스크립트 실행

작업	설명	필요한 기술
requirements.txt 파일에서 pip 패키지를 설치합니다.	<p>첨부된 requirements.txt 파일을 다운로드하고 IDE에서 다음 명령을 실행하여 Python 패키지를 설치합니다.</p> <pre>pip3 install -r requirements.txt</pre>	개발자
Python 스크립트를 실행합니다.	<ol style="list-style-type: none"> 1. 로컬 시스템에 첨부된 aws_config_rules.py 파일을 다운로드합니다. 2. - python3 aws_config_rules.py --ou <OU_NAME> 명령을 실행합니다. 참고: --ou는 Excel 스프레드시트에서 선택할 OU 열을 정의합니다. <p>다음과 같은 선택적 파라미터도 추가할 수 있습니다.</p> <ul style="list-style-type: none"> • --config-rule-option -Excel 스프레드시트에서 선택할 규칙을 정의합니다. 기본값은 Implemented 파라미터입니다. • --excel-file -Excel 스프레드시트의 경로입니다. 기본값은 aws_config_rules.xlsx 입니다. • --param-file -파라미터 JSON 파일의 경로입니다. 기본값은 config_ru 	개발자

작업	설명	필요한 기술
	<p>les_params.json 입니다.</p> <ul style="list-style-type: none"> --max-execution-frequency -AWS Config 관리형 규칙을 평가하는 빈도를 정의합니다. 선택 항목은 One_Hour, Three_Hours, Six_Hours, Twelve_Hours, 또는 TwentyFour_Hours 입니다. 기본값은 TwentyFour_Hours 입니다. 	

AWS Config 관리형 규칙을 배포합니다.

작업	설명	필요한 기술
AWS CloudFormation 스택을 시작합니다.	<ol style="list-style-type: none"> AWS 관리 콘솔에 로그인하고 AWS 콘솔을 연 다음 [스택 생성] 을 선택합니다. CloudFormation 템플릿 지정 페이지에서 템플릿 파일 업로드를 선택한 다음 AWS CloudFormation 템플릿을 업로드합니다. 스택 이름을 지정하고 다음을 선택합니다. 태그를 지정하고 다음을 선택합니다. 스택 생성을 선택합니다. 	개발자

첨부

이 문서와 관련된 추가 콘텐츠에 액세스하려면 [attachment.zip](#) 파일의 압축을 풉니다.

SageMaker 노트북 인스턴스에 다른 AWS 계정의 CodeCommit 리포지토리에 대한 임시 액세스 권한 부여

작성자: Helge Aufderheide(AWS)

환경: 프로덕션

기술: DevOps; 분석; 기계 학습
및 AI; 관리 및 거버넌스

AWS 서비스: AWS
CodeCommit, AWS 자격 증명 및 액세스 관리, 아마존 SageMaker

요약

이 패턴은 Amazon SageMaker 노트북 인스턴스와 사용자에게 다른 AWS 계정에 있는 AWS CodeCommit 리포지토리에 대한 임시 액세스 권한을 부여하는 방법을 보여줍니다. 또한 이 패턴은 각 엔티티가 각 리포지토리에서 수행할 수 있는 특정 작업에 대해 세분화된 권한을 부여하는 방법을 보여줍니다.

조직은 종종 개발 환경을 호스팅하는 계정과 다른 AWS 계정에 CodeCommit 리포지토리를 저장합니다. 이 다중 계정 설정은 리포지토리에 대한 액세스를 제어하고 실수로 삭제될 위험을 줄이는 데 도움이 됩니다. 이러한 교차 계정 권한을 부여하기 위해 AWS Identity and Access Management(IAM) 역할을 사용하는 것이 좋습니다. 그러면 각 AWS 계정의 사전 정의된 IAM ID가 일시적으로 역할을 맡아 계정 전체에 통제된 신뢰 체인을 만들 수 있습니다.

참고: 유사한 절차를 적용하여 다른 IAM ID에 리포지토리에 대한 교차 계정 액세스 권한을 부여할 수 있습니다. CodeCommit 자세한 내용은 AWS 사용 CodeCommit 설명서의 [역할을 사용하여 AWS CodeCommit 리포지토리에 대한 교차 계정 액세스 구성](#)을 참조하십시오.

사전 조건 및 제한 사항

사전 조건

- CodeCommit 리포지토리가 있는 활성 AWS 계정 (계정 A)
- SageMaker 노트북 인스턴스가 있는 두 번째 활성 AWS 계정 (계정 B)
- 계정 A에서 IAM 역할을 생성하고 수정할 수 있는 충분한 권한을 가진 AWS 사용자
- 계정 B에서 IAM 역할을 생성하고 수정할 수 있는 충분한 권한을 가진 두 번째 AWS 사용자

아키텍처

다음 다이어그램은 SageMaker 노트북 인스턴스와 하나의 AWS 계정의 사용자에게 리포지토리에 대한 교차 계정 액세스 권한을 부여하는 예제 워크플로를 보여줍니다. CodeCommit

이 다이어그램은 다음 워크플로를 보여줍니다.

1. 계정 B의 AWS 사용자 역할 및 SageMaker 노트북 인스턴스 역할은 [이름이 지정된 프로필을](#) 가정합니다.
2. 지정된 프로필의 권한 정책은 계정 A의 CodeCommit 액세스 역할을 지정하며, 이 액세스 역할은 해당 프로필이 위임합니다.
3. 계정 A의 CodeCommit 액세스 역할 신뢰 정책에 따라 계정 B의 명명된 프로필이 CodeCommit 액세스 역할을 맡을 수 있습니다.
4. 계정 A의 CodeCommit 리포지토리 IAM 권한 정책은 CodeCommit 액세스 역할이 CodeCommit 리포지토리에 액세스하는 것을 허용합니다.

기술 스택

- CodeCommit
- Git
- IAM
- pip
- SageMaker

도구

- [CodeCommitAWS](#)는 자체 소스 제어 시스템을 관리할 필요 없이 Git 리포지토리를 비공개로 저장하고 관리할 수 있는 버전 제어 서비스입니다.
- [AWS Identity and Access Management\(IAM\)](#)은 사용자에게 대한 인증 및 권한 부여를 제어함으로써 AWS 리소스에 대한 액세스를 안전하게 제어할 수 있습니다.
- [Git](#)은 소프트웨어 개발 중에 소스 코드의 변경 사항을 추적하기 위한 분산 버전 제어 시스템입니다.
- [git-remote-codecommit](#) CodeCommit Git을 확장하여 리포지토리에서 코드를 푸시하고 가져오는 데 도움이 되는 유틸리티입니다.

- [pip](#)는 Python용 패키지 인스톨러입니다. pip를 사용하여 Python 패키지 색인 및 기타 색인에서 패키지를 설치할 수 있습니다.

모범 사례

IAM 정책을 사용하여 권한을 설정하는 경우 작업을 수행하는 데 필요한 권한만 부여했는지 확인합니다. 자세한 내용은 IAM 설명서의 [최소 권한 적용](#)을 참조하십시오.

이 패턴을 구현할 때는 다음을 수행해야 합니다.

- IAM 원칙에 각 리포지토리 내에서 필요한 특정 작업을 수행하는 데 필요한 권한만 있는지 확인합니다. 예를 들어 승인된 IAM 원칙이 특정 리포지토리 브랜치에 변경 내용을 푸시하고 병합하도록 허용하되 보호된 브랜치에 대한 병합만 요청할 수 있도록 허용하는 것이 좋습니다.
- IAM 원칙에 각 프로젝트의 역할 및 책임에 따라 서로 다른 IAM 역할이 할당되었는지 확인합니다. 예를 들어, 개발자는 릴리스 관리자 또는 AWS 관리자와는 다른 액세스 권한을 가집니다.

에픽

IAM 역할 구성

작업	설명	필요한 기술
CodeCommit 액세스 역할 및 권한 정책을 구성합니다.	<p>참고: 이 에픽에 설명된 수동 설정 프로세스를 자동화하려면 AWS CloudFormation 템플릿을 사용할 수 있습니다.</p> <p>CodeCommit 리포지토리가 포함된 계정 (계정 A) 에서 다음을 수행하십시오.</p> <ol style="list-style-type: none"> 1. 계정 B의 SageMaker 노트북 인스턴스 역할이 위임할 수 있는 IAM 역할을 생성합니다. 2. 리포지토리에 대한 액세스 권한을 부여하는 IAM 정 	일반 AWS, AWS DevOps

작업	설명	필요한 기술
	<p>책을 생성하고 정책을 역할에 연결합니다. 테스트 목적으로만 AWSCodeCommitPowerUserAWS 관리형 정책을 선택하십시오. 이 정책은 리소스 삭제 CodeCommit 권한을 제외한 모든 권한을 부여합니다.</p> <p>3. 계정 B가 신뢰할 수 있는 엔티티로 나열되도록 역할의 신뢰 정책을 수정합니다.</p> <p>중요: 이 설정을 프로덕션 환경으로 이전하기 전에 최소 권한을 적용하는 자체 IAM 정책을 작성하는 것이 좋습니다. 자세한 내용은 이 패턴의 추가 정보 섹션을 참조하세요.</p>	

작업	설명	필요한 기술
<p>계정 B의 SageMaker 노트북 인스턴스 역할에 계정 A에서 CodeCommit 액세스 역할을 맡을 수 있는 권한을 부여합니다.</p>	<p>SageMaker 노트북 인스턴스의 IAM 역할 (계정 B) 이 포함된 계정에서 다음을 수행하십시오.</p> <ol style="list-style-type: none"> 1. IAM 역할 또는 사용자가 계정 A의 CodeCommit 액세스 역할을 수임하도록 허용하는 IAM 정책을 생성합니다. <p>IAM 역할 또는 사용자가 크로스 계정 역할을 맡도록 허용하는 IAM 권한 정책의 예</p> <pre data-bbox="630 842 1029 1514"> { "Version": "2012-10-17", "Statement": [{ "Sid": "VisualEditor0", "Effect": "Allow", "Action": "sts:AssumeRole", "Resource": "arn:aws:iam:::accountA_ID:role/accountArole_ID" }] } </pre> <ol style="list-style-type: none"> 2. 정책을 계정 B에 있는 SageMaker 노트북 인스턴스의 역할에 연결합니다. 3. 계정 B의 SageMaker 노트북 인스턴스 역할이 계정 A의 CodeCommit 액세스 역할을 맡도록 하십시오. 	<p>일반 AWS, AWS DevOps</p>

작업	설명	필요한 기술
	<p>참고: 리포지토리의 Amazon 리소스 이름 (ARN) 을 보려면 AWS CodeCommit 사용 설명서의 CodeCommit 리포지토리 세부 정보 보기를 참조하십시오.</p>	

계정 B에서 SageMaker 노트북 인스턴스 설정

작업	설명	필요한 기술
<p>AWS SageMaker 노트북 인스턴스에서 계정 A에서 역할을 맡을 사용자 프로필을 설정합니다.</p>	<p>중요: 최신 버전의 AWS Command Line Interface(AWS CL) 가 설치되어 있어야 합니다.</p> <p>SageMaker 노트북 인스턴스가 포함된 계정 (계정 B) 에서 다음을 수행하십시오.</p> <ol style="list-style-type: none"> 1. AWS Management Console 에 로그인하고 SageMaker 콘솔을 엽니다. 2. SageMaker 노트북 인스턴스에 액세스합니다. Jupyter 인터페이스가 열립니다. 3. 새로 만들기를 선택한 다음 터미널을 선택합니다. Jupyter 환경에 새 터미널 창이 열립니다. 4. SageMaker 노트북 인스턴스의 ~/.aws/config 파일로 이동합니다. 그런 다음 아래의 명령문을 입력하여 파일 	<p>일반 AWS, AWS DevOps</p>

작업	설명	필요한 기술
	<p>에 사용자 프로필을 추가합니다.</p> <pre> ----- .aws/config- ----- [profile remoterep ouser] role_arn = arn:aws:i am::<ID of Account A>:role/<rolename> role_session_name = remoteaccesssession region = eu-west-1 credential_source = Ec2InstanceMetadata ----- ----- </pre>	
<p>git-remote-codecommit 유틸리티 설치.</p>	<p>AWS 사용 CodeCommit 설명서의 2단계: git-remote-codecommit 설치의 지침을 따르십시오.</p>	<p>데이터 사이언티스트</p>

리포지토리에 액세스

작업	설명	필요한 기술
<p>Git 명령 또는 을 사용하여 CodeCommit 저장소에 액세스합니다. SageMaker</p>	<p>Git 사용하기</p> <p>계정 B에서 SageMaker 노트북 인스턴스의 역할을 맡은 IAM 보안 주체는 이제 Git 명령을 실행하여 계정 A의 CodeCommit 저장소에 액세스할 수 있습니다. 예를 들어,, 사용자는, 등의 명령을 실행할 수 있습니다.</p>	<p>Git, bash 콘솔</p>

작업	설명	필요한 기술
	<p>git clone git pull git push</p> <p>지침은 AWS 사용 CodeCommit 설명서의 AWS CodeCommit 리포지토리에 연결을 참조하십시오.</p> <p>Git을 CodeCommit 에서 사용하는 방법에 대한 자세한 내용은 AWS 사용 CodeCommit 설명서의 AWS 시작하기를 참조하십시오. CodeCommit</p> <p>사용하려면 SageMaker</p> <p>SageMaker 콘솔에서 Git을 사용하려면 Git이 리포지토리에 서 자격 증명을 검색하도록 허용해야 합니다. CodeCommit 지침은 설명서에서 다른 AWS 계정의 CodeCommit 리포지토리를 노트북 인스턴스와 연결을 SageMaker 참조하십시오.</p>	

관련 리소스

- [역할을 사용하여 AWS CodeCommit 리포지토리에 대한 교차 계정 액세스 구성 \(AWS CodeCommit 설명서\)](#)
- [IAM 자습서: IAM 역할을 사용하여 AWS 계정 전체에 대한 액세스 위임 \(IAM 설명서\)](#)

추가 정보

특정 작업에 대한 권한 제한 CodeCommit

IAM 보안 주체가 CodeCommit 저장소에서 수행할 수 있는 작업을 제한하려면 CodeCommit 액세스 정책에서 허용되는 작업을 수정하십시오.

CodeCommit API 작업에 대한 자세한 내용은 AWS 사용 CodeCommit 설명서의 [CodeCommit 권한 참조](#)를 참조하십시오.

참고: 사용 사례에 맞게 [AWSCodeCommitPowerUser](#) AWS 관리형 정책을 편집할 수도 있습니다.

특정 리포지토리에 CodeCommit 대한 권한 제한

특정 사용자만 둘 이상의 코드 리포지토리에 액세스할 수 있는 멀티테넌트 환경을 만들려면 다음을 수행합니다.

1. 계정 A에서 여러 CodeCommit 액세스 역할을 만든 다음, 계정 B의 특정 사용자가 역할을 수임할 수 있도록 각 액세스 역할의 신뢰 정책을 구성합니다.
2. 각 CodeCommit 액세스 역할의 정책에 “리소스” 조건을 추가하여 각 역할이 맡을 수 있는 코드 저장소를 제한하세요.

IAM 보안 주체의 특정 리포지토리 액세스를 제한하는 “리소스” 조건의 예 CodeCommit

```
"Resource" : [ <REPOSITORY_ARN>, <REPOSITORY_ARN> ]
```

참고: 동일한 AWS 계정에서 여러 코드 리포지토리를 식별하고 구별하기 위해 리포지토리 이름에 다른 접두사를 할당할 수 있습니다. 예를 들어 myproject-subproject1-repo1 및 myproject-subproject2-repo1과 같이 다양한 개발자 그룹에 맞는 접두사를 사용하여 코드 리포지토리의 이름을 지정할 수 있습니다. 그런 다음 개발자 그룹에 할당된 접두사를 기반으로 각 개발자 그룹에 대한 IAM 역할을 생성할 수 있습니다. 예를 들어 myproject-subproject1-repoaccess라는 역할을 생성하고 myproject-subproject1 접두사가 포함된 모든 코드 리포지토리에 대한 액세스 권한을 부여할 수 있습니다.

특정 접두사가 포함된 코드 리포지토리 ARN을 참조하는 “리소스” 조건의 예

```
"Resource" : arn:aws:codecommit:<region>:<account-id>:myproject-subproject1-*
```

다중 DevOps 계정 GitHub 환경을 위한 Flow 분기 전략 구현

마이크 스티븐스 (AWS) 와 아빌라쉬 비노드 (AWS) 가 제작했습니다.

[코드 리포지토리: -멀티어카운트-데브옵스 git-branching-strategies-for](#)

환경: 프로덕션

기술: DevOps; 소프트웨어 개발 및 테스트; 다중 계정 전략

AWS 서비스: AWS CodeArtifact, AWS CodeBuild, AWS CodeCommit, AWS CodeDeploy, AWS CodePipeline

요약

소스 코드 리포지토리를 관리할 때는 다양한 브랜칭 전략이 개발 팀이 사용하는 소프트웨어 개발 및 릴리스 프로세스에 영향을 미칩니다. 일반적인 분기 전략의 예로는 트렁크, GitHub 플로우, Gitflow 등이 있습니다. 이러한 전략은 서로 다른 브랜치를 사용하며 각 환경에서 수행되는 활동도 다릅니다. DevOps 프로세스를 구현하는 조직은 이러한 분기 전략 간의 차이점을 이해하는 데 도움이 되는 시각적 가이드를 유용하게 활용할 수 있습니다. 조직에서 이 비주얼을 사용하면 개발팀이 업무를 조율하고 조직 표준을 준수하는 데 도움이 됩니다. 이 패턴은 이러한 시각적 정보를 제공하며 조직에서 GitHub Flow 분기 전략을 구현하는 프로세스를 설명합니다.

이 패턴은 여러 조직이 있는 조직을 위한 DevOps 분기 전략 선택 및 구현에 대한 설명서 시리즈의 일부입니다. AWS 계정 이 시리즈는 처음부터 올바른 전략과 모범 사례를 적용하여 클라우드에서의 경험을 간소화하는 데 도움이 되도록 설계되었습니다. GitHub Flow는 조직에서 사용할 수 있는 분기 전략 중 하나일 뿐입니다. 이 설명서 시리즈에서는 [Trunk](#) 및 [Gitflow](#) 분기 모델도 다룹니다. 아직 검토하지 않았다면 이 패턴의 지침을 구현하기 전에 [다중 계정 DevOps 환경을 위한 Git 브랜칭 전략 선택](#)을 검토하는 것이 좋습니다. 실사를 통해 조직에 적합한 브랜칭 전략을 선택하시기 바랍니다.

이 가이드는 조직이 GitHub Flow 전략을 구현하는 방법을 보여주는 다이어그램을 제공합니다. [AWS DevOps Well-Architected](#) 지침을 검토하여 모범 사례를 검토하는 것이 좋습니다. 이 패턴에는 프로세스의 각 단계에 대한 권장 작업, 단계 및 제한 사항이 포함됩니다. DevOps

사전 조건 및 제한 사항

사전 조건

- [Git이 설치되었습니다.](#) 소스 코드 리포지토리 도구로 사용됩니다.
- [Draw.io가 설치되었습니다.](#) 이 응용 프로그램은 다이어그램을 보고 편집하는 데 사용됩니다.

아키텍처

대상 아키텍처

다음 다이어그램은 [퍼넷 스퀘어](#) (Wikipedia) 처럼 사용할 수 있습니다. 세로 축의 분기를 가로 축의 AWS 환경과 정렬하여 각 시나리오에서 수행할 작업을 결정합니다. 숫자는 워크플로우에서의 작업 순서를 나타냅니다. 이 예제에서는 feature 브랜치부터 프로덕션 환경으로의 배포까지 안내합니다.

GitHub Flow 접근 방식의 AWS 계정, 환경 및 분기에 대한 자세한 내용은 다중 계정 환경을 [위한 Git 분기 전략 선택](#)을 참조하십시오. DevOps

자동화 및 규모 조정

지속적 통합 및 지속적 전달 (CI/CD) 은 소프트웨어 릴리스 라이프사이클을 자동화하는 프로세스입니다. 새 코드를 처음 커밋한 후 프로덕션에 이르기까지 기존에 필요했던 수동 프로세스의 대부분 또는 전체를 자동화합니다. CI/CD 파이프라인은 샌드박스, 개발, 테스트, 스테이징 및 프로덕션 환경을 포함합니다. 각 환경에서 CI/CD 파이프라인은 코드를 배포하거나 테스트하는 데 필요한 모든 인프라를 제공합니다. 개발팀은 CI/CD를 사용하여 코드를 변경한 다음 자동으로 테스트 및 배포할 수 있습니다. 또한 CI/CD 파이프라인은 기능 수용 및 배포에 대한 일관성, 표준, 모범 사례, 최소 허용 수준을 적용함으로써 개발 팀을 위한 거버넌스와 보호 수단을 제공합니다. 자세한 내용은 [지속적 통합 및 지속적 전달 실천](#)을 참조하십시오. AWS

AWS CI/CD 파이프라인을 구축하는 데 도움이 되도록 설계된 개발자 서비스 제품군을 제공합니다. 예를 들어 [AWS CodePipeline](#)는 빠르고 안정적인 애플리케이션 및 인프라 업데이트를 위해 릴리스 파이프라인을 자동화하는 데 도움이 되는 완전관리형 지속적 전달 서비스입니다. [AWS CodeCommit](#) 확장 가능한 Git 리포지토리를 안전하게 호스팅하고, 소스 코드를 [AWS CodeBuild](#) 컴파일하고, 테스트를 실행하고, 소프트웨어 패키지를 생성하도록 설계되었습니다. ready-to-deploy [자세한 내용은 개발자 도구 on](#)을 참조하십시오. AWS

도구

AWS 서비스 및 도구

AWS 이 패턴을 구현하는 데 사용할 수 있는 개발자 서비스 제품군을 제공합니다.

- [AWS CodeArtifact](#) 애플리케이션 개발을 위한 소프트웨어 패키지를 저장하고 공유하는 데 도움이 되는 확장성이 뛰어난 관리형 아티팩트 리포지토리 서비스입니다.
- [AWS CodeBuild](#) 소스 코드를 컴파일하고, 단위 테스트를 실행하고, 배포할 준비가 된 아티팩트를 생성하는 데 도움이 되는 완전 관리형 빌드 서비스입니다.
- [AWS CodeCommit](#) 자체 소스 제어 시스템을 관리할 필요 없이 Git 리포지토리를 비공개로 저장하고 관리할 수 있는 버전 제어 서비스입니다.
- [AWS CodeDeploy](#) 아마존 Elastic Compute Cloud (Amazon EC2) 또는 온프레미스 인스턴스 AWS Lambda, 함수 또는 아마존 Elastic Container Service (Amazon ECS) 서비스로의 배포를 자동화합니다.
- [AWS CodePipeline](#) 소프트웨어 릴리스의 여러 단계를 신속하게 모델링 및 구성하고 소프트웨어 변경 사항을 지속적으로 릴리스하는 데 필요한 단계를 자동화할 수 있도록 지원합니다.

기타 도구

- [Draw.io Desktop](#)은 플로우차트와 다이어그램을 만드는 데 사용되는 애플리케이션입니다. 코드 리포지토리에는 Draw.io용.drawio 형식의 템플릿이 들어 있습니다.
- [Figma](#)는 협업을 위해 설계된 온라인 디자인 도구입니다. 코드 리포지토리에는 Figma용.fig 형식의 템플릿이 포함되어 있습니다.

코드 리포지토리

이 패턴의 다이어그램에 대한 이 소스 파일은 Flow용 GitHub [Git 브랜칭 전략](#) 리포지토리에서 사용할 수 있습니다. GitHub 여기에는 PNG, draw.io 및 Figma 형식의 파일이 포함됩니다. 조직의 프로세스를 지원하도록 이러한 다이어그램을 수정할 수 있습니다.

모범 사례

[AWS DevOps Well-Architected 지침 및 다중 계정 환경을 위한 Git 브랜칭 전략 선택의](#) 모범 사례 및 권장 사항을 따르세요. DevOps 이를 통해 GitHub Flow 기반 개발을 효과적으로 구현하고, 협업을 촉진하고, 코드 품질을 개선하고, 개발 프로세스를 간소화할 수 있습니다.

에픽

Flow 워크플로 검토 GitHub

작업	설명	필요한 기술
표준 GitHub Flow 프로세스를 검토하세요.	<ol style="list-style-type: none"> 1. 샌드박스 환경에서 개발자는 브랜치에서 feature 브랜치를 만들고 이름 지정 패턴을 사용합니다. <code>main feature/<ticket>_<initials>_<short description></code> 2. 개발자는 feature 브랜치에 하나 이상의 커밋을 추가합니다. 각 커밋은 개별 변경 또는 개선을 나타냅니다. 3. 개발자는 MR (병합 요청) 을 열어 변경 내용을 브랜치에 병합합니다. main 그러면 검토 프로세스가 시작됩니다. 4. 검토 과정에서 개발자는 코드 변경 사항에 대해 논의하고 피드백을 제공합니다. 목표는 변경 내용의 품질이 우수하고 프로젝트 표준을 충족하는지 확인하는 것입니다. 5. 개발자가 병합 요청을 생성하면 자동화된 빌드 프로세스가 시작되어 feature 브랜치의 변경 내용을 개발 환경에 배포합니다. 6. 자동화된 테스트는 병합 요청에 캡슐화된 변경 사항의 무결성과 품질을 확인합니다. 	DevOps 엔지니어

작업	설명	필요한 기술
	<p>다. 병합 요청을 완료하려면 성공적인 빌드, 성공적인 배포, 성공적인 테스트가 필요합니다.</p> <p>7. 검토 프로세스가 완료되면 변경 내용이 main 브랜치에 병합됩니다.</p> <p>8. 승인자는 테스트 환경에 릴리스 아티팩트를 배포하는 것을 수동으로 승인합니다.</p> <p>9. 승인자는 스테이징 환경에 릴리스 아티팩트를 배포하는 것을 수동으로 승인합니다.</p> <p>10. 승인자는 프로덕션 환경에 릴리스 아티팩트를 배포하는 것을 수동으로 승인합니다.</p>	

작업	설명	필요한 기술
버그 수정 GitHub 흐름 프로세스를 검토하세요.	<ol style="list-style-type: none"> 1. 개발자는 브랜치에서 bugfix 브랜치를 만들고 이를 지정 패턴을 사용합니다. main bugfix/<ticket number>_<developer initials>_<descriptor> 2. 개발자는 문제를 수정하고 수정 사항을 커밋한 다음 브랜치를 빌드합니다. bugfix 3. 개발자가 병합 요청을 열어 bugfix 브랜치를 브랜치에 병합합니다main. 그러면 검토 프로세스가 시작됩니다. 4. 검토 과정에서 개발자는 코드 변경 사항에 대해 논의하고 피드백을 제공합니다. 5. 검토가 완료되고 승인되면 개발자는 bugfix 브랜치에 대한 병합 요청을 완료합니다. main 6. 승인자는 상위 환경에 릴리스 아티팩트를 배포하는 것을 수동으로 승인합니다. 	DevOps 엔지니어

작업	설명	필요한 기술
<p>핫픽스 GitHub 플로우 프로세스를 검토하십시오.</p>	<p>GitHub Flow는 코드 변경이 빈번하고 안정적으로 상위 환경에 배포되는 지속적 전달이 가능하도록 설계되었습니다. 핵심은 모든 feature 브랜치를 언제든지 배포할 수 있다는 것입니다.</p> <p>Hotfix 브랜치와 비슷한 브랜치는 다른 feature bugfix 브랜치와 동일한 프로세스를 따를 수 있습니다. 하지만 긴급성을 고려하면 일반적으로 핫픽스의 우선 순위가 더 높습니다. 팀의 정책과 상황의 즉각성에 따라 프로세스의 특정 단계가 빨라질 수 있습니다. 예를 들어 핫픽스에 대한 코드 검토는 신속하게 처리될 수 있습니다. 따라서 핫픽스 프로세스는 기능 또는 버그 수정 프로세스와 유사하지만 핫픽스와 관련된 긴급성으로 인해 절차 준수의 수정이 필요할 수 있습니다. 핫픽스가 효율적이고 안전하게 처리되도록 하려면 핫픽스 관리에 대한 지침을 수립하는 것이 중요합니다.</p>	<p>DevOps 엔지니어</p>

문제 해결

문제	Solution
브랜치 충돌	GitHub Flow 모델에서 발생할 수 있는 일반적인 문제는 프로덕션 환경에서 핫픽스를 생성해야 하지만 동일한 리소스가 수정되는 feature 경우 bugfix, 또는 hotfix 브랜치에서 해당 변경이 발생해야 하는 경우입니다. main에 병합할 때 심각한 충돌을 피하려면 하위 브랜치에서 하위 브랜치로 변경 내용을 자주 병합하는 것이 main 좋습니다.
팀 성숙도	GitHub Flow는 진정한 지속적 통합 및 지속적 전달 (CI/CD) 을 수용하여 상위 환경으로의 일일 배포를 장려합니다. 기능을 구축하고 이를 위한 자동화 테스트를 만들려면 팀이 엔지니어링 성숙도를 갖추어야 합니다. 팀은 변경 사항을 승인하기 전에 철저한 병합 요청 검토를 수행해야 합니다. 이를 통해 개발 프로세스의 품질, 책임 및 효율성을 증진하는 견고한 엔지니어링 문화가 조성됩니다.

관련 리소스

이 가이드에는 Git에 대한 교육이 포함되어 있지 않지만, 이 교육이 필요한 경우 인터넷에서 많은 고품질 리소스를 이용할 수 있습니다. [Git 설명서](#) 사이트에서 시작하는 것이 좋습니다.

다음 리소스는 에서의 GitHub Flow 브랜칭 여정에 도움이 될 수 있습니다. AWS 클라우드

AWS DevOps 지침

- [AWS DevOps 지침](#)
- [AWS 배포 파이프라인 참조 아키텍처](#)
- [무엇입니까 DevOps?](#)
- [DevOps 리소스](#)

GitHub 흐름 지침

- [GitHub 플로우 퀵스타트 튜토리얼](#) () GitHub
- [왜 GitHub 플로우인가?](#)

기타 리소스

- [트웰-팩터 앱 방법론](#) (12factor.net)

다중 계정 환경을 위한 Gitflow 브랜칭 전략 구현 DevOps

제작: 마이크 스티븐스 (AWS), 스티븐 (DiCato AWS), 팀 원더젬 (AWS), 아빌라쉬 비노드 (AWS)

코드 [git-branching-strategies-for 리포지토리](#): -멀티어카운트-데브옵스

환경: 프로덕션

기술: DevOps; 소프트웨어 개발 및 테스트; 다중 계정 전략

AWS 서비스: AWS CodeArtifact, AWS CodeBuild, AWS CodeCommit, AWS CodeDeploy, AWS CodePipeline

요약

소스 코드 리포지토리를 관리할 때는 다양한 브랜칭 전략이 개발 팀이 사용하는 소프트웨어 개발 및 릴리스 프로세스에 영향을 미칩니다. 일반적인 분기 전략의 예로는 트렁크, Gitflow, Flow 등이 있습니다. GitHub 이러한 전략은 서로 다른 브랜치를 사용하며 각 환경에서 수행되는 활동도 다릅니다. DevOps 프로세스를 구현하는 조직은 이러한 분기 전략 간의 차이점을 이해하는 데 도움이 되는 시각적 가이드를 유용하게 활용할 수 있습니다. 조직에서 이 비주얼을 사용하면 개발팀이 업무를 조율하고 조직 표준을 준수하는 데 도움이 됩니다. 이 패턴은 이러한 시각적 효과를 제공하며 조직에서 Gitflow 브랜칭 전략을 구현하는 프로세스를 설명합니다.

이 패턴은 여러 조직이 있는 조직을 위한 DevOps 분기 전략을 선택하고 구현하는 방법에 대한 설명서 시리즈의 일부입니다. AWS 계정 이 시리즈는 처음부터 올바른 전략과 모범 사례를 적용하여 클라우드에서의 경험을 간소화하는 데 도움이 되도록 설계되었습니다. Gitflow는 조직에서 사용할 수 있는 분기 전략 중 하나일 뿐입니다. 이 설명서 시리즈에서는 [트렁크](#) 및 [GitHub 플로우](#) 브랜칭 모델도 다릅니다. 아직 검토하지 않았다면 이 패턴의 지침을 구현하기 전에 [다중 계정 DevOps 환경을 위한 Git 브랜칭 전략 선택](#)을 검토하는 것이 좋습니다. 실사를 통해 조직에 적합한 브랜칭 전략을 선택하시기 바랍니다.

이 가이드는 조직이 Gitflow 전략을 구현하는 방법을 보여주는 다이어그램을 제공합니다. [AWS DevOps Well-Architected](#) 지침을 검토하여 모범 사례를 검토하는 것이 좋습니다. 이 패턴에는 프로세스의 각 단계에 대한 권장 작업, 단계 및 제한 사항이 포함됩니다. DevOps

사전 조건 및 제한 사항

사전 조건

- [Git이 설치되었습니다.](#) 소스 코드 리포지토리 도구로 사용됩니다.
- [Draw.io가 설치되었습니다.](#) 이 응용 프로그램은 다이어그램을 보고 편집하는 데 사용됩니다.
- [\(선택 사항\) Gitflow 플러그인, 설치됨.](#)

아키텍처

대상 아키텍처

다음 다이어그램은 [퍼넷 스퀘어](#) (위키백과) 처럼 사용할 수 있습니다. 세로 축의 분기를 가로 축의 AWS 환경과 정렬하여 각 시나리오에서 수행할 작업을 결정합니다. 숫자는 워크플로우에서의 작업 순서를 나타냅니다. 이 예제에서는 기능 브랜치부터 프로덕션 환경에서의 배포까지 안내합니다.

Gitflow 접근 방식의 환경 AWS 계정, 브랜치에 대한 자세한 내용은 다중 계정 환경을 위한 [Git 브랜칭 전략 선택](#)을 참조하십시오. DevOps

자동화 및 규모 조정

지속적 통합 및 지속적 전달 (CI/CD) 은 소프트웨어 릴리스 라이프사이클을 자동화하는 프로세스입니다. 새 코드를 처음 커밋한 후 프로덕션에 이르기까지 기존에 필요했던 수동 프로세스의 대부분 또는 전체를 자동화합니다. CI/CD 파이프라인은 샌드박스, 개발, 테스트, 스테이징 및 프로덕션 환경을 포함합니다. 각 환경에서 CI/CD 파이프라인은 코드를 배포하거나 테스트하는 데 필요한 모든 인프라를 제공합니다. 개발팀은 CI/CD를 사용하여 코드를 변경한 다음 자동으로 테스트 및 배포할 수 있습니다. 또한 CI/CD 파이프라인은 기능 수용 및 배포에 대한 일관성, 표준, 모범 사례, 최소 허용 수준을 적용함으로써 개발 팀을 위한 거버넌스와 보호 수단을 제공합니다. 자세한 내용은 [지속적 통합 및 지속적 전달 실천](#)을 참조하십시오. AWS

AWS CI/CD 파이프라인을 구축하는 데 도움이 되도록 설계된 개발자 서비스 제품군을 제공합니다. 예를 들어 [AWS CodePipeline](#)는 빠르고 안정적인 애플리케이션 및 인프라 업데이트를 위해 릴리스 파이프라인을 자동화하는 데 도움이 되는 완전관리형 지속적 전달 서비스입니다. [AWS CodeCommit](#) 확장 가능한 Git 리포지토리를 안전하게 호스팅하고, 소스 코드를 [AWS CodeBuild](#) 컴파일하고, 테스트를 실행하고, 소프트웨어 패키지를 생성하도록 설계되었습니다. ready-to-deploy [자세한 내용은 개발자 도구 on](#)을 참조하십시오. AWS

도구

AWS 서비스 및 도구

AWS 이 패턴을 구현하는 데 사용할 수 있는 개발자 서비스 제품군을 제공합니다.

- [AWS CodeArtifact](#) 애플리케이션 개발을 위한 소프트웨어 패키지를 저장하고 공유하는 데 도움이 되는 확장성이 뛰어난 관리형 아티팩트 리포지토리 서비스입니다.
- [AWS CodeBuild](#) 소스 코드를 컴파일하고, 단위 테스트를 실행하고, 배포할 준비가 된 아티팩트를 생성하는 데 도움이 되는 완전 관리형 빌드 서비스입니다.
- [AWS CodeCommit](#) 자체 소스 제어 시스템을 관리할 필요 없이 Git 리포지토리를 비공개로 저장하고 관리할 수 있는 버전 제어 서비스입니다.
- [AWS CodeDeploy](#) 아마존 Elastic Compute Cloud (Amazon EC2) 또는 온프레미스 인스턴스 AWS Lambda, 함수 또는 아마존 Elastic Container Service (Amazon ECS) 서비스로의 배포를 자동화합니다.
- [AWS CodePipeline](#) 소프트웨어 릴리스의 여러 단계를 신속하게 모델링 및 구성하고 소프트웨어 변경 사항을 지속적으로 릴리스하는 데 필요한 단계를 자동화할 수 있도록 지원합니다.

기타 도구

- [Draw.io Desktop](#)은 플로우차트와 다이어그램을 만드는 응용 프로그램입니다. 코드 리포지토리에는 Draw.io용.drawio 형식의 템플릿이 들어 있습니다.
- [Figma](#)는 협업을 위해 설계된 온라인 디자인 도구입니다. 코드 리포지토리에는 Figma용.fig 형식의 템플릿이 포함되어 있습니다.
- (선택 사항) [Gitflow 플러그인은](#) Gitflow 분기 모델을 위한 고급 리포지토리 작업을 제공하는 Git 확장 컬렉션입니다.

코드 리포지토리

이 패턴의 다이어그램에 대한 이 소스 파일은 리포지토리의 GitHub [Git Branching Strategy](#)에서 사용할 수 있습니다. GitFlow 여기에는 PNG, draw.io 및 Figma 형식의 파일이 포함됩니다. 조직의 프로세스를 지원하도록 이러한 다이어그램을 수정할 수 있습니다.

모범 사례

[AWS DevOps Well-Architected 지침 및 다중 계정 환경을 위한 Git 브랜칭 전략 선택의](#) 모범 사례 및 권장 사항을 따르세요. DevOps 이를 통해 GitFlow 기반 개발을 효과적으로 구현하고, 협업을 촉진하고, 코드 품질을 개선하고, 개발 프로세스를 간소화할 수 있습니다.

에픽

Gitflow 워크플로 검토

작업	설명	필요한 기술
표준 Gitflow 프로세스를 검토 하세요.	<ol style="list-style-type: none"> 샌드박스 환경에서 개발자는 브랜치에서 feature 브랜치를 만들고 이름 develop 지정 패턴을 사용합니다. feature/<ticket>_<initials>_<short description> 개발자는 티켓을 완료하기 위해 코드를 개발하고 샌드박스 환경에 코드를 반복적으로 배포합니다. 참고: 개발자는 선택적으로 sandbox 브랜치를 만들어 샌드박스 환경에서 자동화된 빌드 또는 배포 파이프라인을 실행할 수 있습니다. 개발자는 스쿼시 병합을 사용하여 feature 브랜치에서 develop 브랜치로 병합 요청을 생성합니다. 지속적 통합 및 지속적 전달 (CI/CD) 파이프라인은 develop 브랜치를 자동으 	DevOps 엔지니어

작업	설명	필요한 기술
	<p>로 빌드하고 개발 환경에 배포합니다.</p> <p>5. (선택 사항) 개발자는 릴리스 활동을 계속하기 전에 추가 feature 브랜치를 개발 브랜치에 통합합니다.</p> <p>6. 브랜치에서 기능을 릴리스할 준비가 되면 개발자는 <code>develop release/v<number></code> 브랜치에서 이름을 딴 <code>release</code> 브랜치를 만듭니다. <code>develop</code></p> <p>7. 개발자는 릴리스 브랜치를 빌드하여 다른 환경에서 재사용할 수 있도록 아티팩트를 게시합니다.</p> <p>8. 승인자는 테스트 환경에 릴리스 아티팩트를 배포하는 것을 수동으로 승인합니다.</p> <p>9. 승인자는 스테이징 환경에 릴리스 아티팩트를 배포하는 것을 수동으로 승인합니다.</p> <p>10. 승인자는 프로덕션 환경에 릴리스 아티팩트를 배포하는 것을 수동으로 승인합니다.</p> <p>11. 개발자는 <code>release</code> 브랜치를 브랜치에 병합합니다. <code>main</code> 개발자가 자동화된 스크립트를 사용하여 빠른 병합을 수행하는 것이 이상적</p>	

작업	설명	필요한 기술
	<p>입니다. 스쿼시 병합은 사용하지 마세요.</p> <p>12개발자가 release 브랜치를 브랜치에 병합합니다. develop 개발자가 자동화된 스크립트를 사용하여 빠른 병합을 수행하는 것이 이상적입니다. 스쿼시 병합은 사용하지 마세요.</p>	

작업	설명	필요한 기술
<p>핫픽스 Gitflow 프로세스를 검토하세요.</p>	<ol style="list-style-type: none"> 1. 개발자는 브랜치에서 hotfix 브랜치를 만들고 이름 main 지정 패턴을 사용합니다. hotfix/<ticket>_<initials>_<short description> 2. 개발자는 브랜치에서 release main 브랜치를 만들고 이름을 지정합니다. release/v<number> 3. 개발자는 문제를 수정하고 수정 사항을 커밋한 다음 hotfix 브랜치를 빌드합니다. 4. 개발자는 스쿼시 병합을 사용하여 hotfix 브랜치에서 release/v<number> 브랜치로 병합 요청을 생성합니다. 5. 개발자는 release 브랜치를 빌드하여 다른 환경에서 재사용할 수 있도록 아티팩트를 게시합니다. 6. 승인자는 테스트 환경에 릴리스 아티팩트를 배포하는 것을 수동으로 승인합니다. 7. 승인자는 스테이징 환경에 릴리스 아티팩트를 배포하는 것을 수동으로 승인합니다. 8. 승인자는 프로덕션 환경에 릴리스 아티팩트를 배포하 	<p>DevOps 엔지니어</p>

작업	설명	필요한 기술
	<p>는 것을 수동으로 승인합니다.</p> <p>9. 개발자는 release 브랜치를 브랜치에 병합합니다. main 개발자가 자동화된 스크립트를 사용하여 빠른 병합을 수행하는 것이 이상적입니다. 스쿼시 병합은 사용하지 마세요.</p> <p>10. 개발자가 release 브랜치를 브랜치에 병합합니다. develop 개발자가 자동화된 스크립트를 사용하여 빠른 병합을 수행하는 것이 이상적입니다. 스쿼시 병합은 사용하지 마세요.</p> <p>11. 충돌이 감지되면 개발자는 알림을 받고 병합 요청을 통해 충돌을 해결합니다.</p>	

작업	설명	필요한 기술
버그 수정 Gitflow 프로세스를 검토하세요.	<ol style="list-style-type: none"> 1. 개발자는 현재 브랜치에서 bugfix 브랜치를 만들고 이름 지정 <code>release/v<number></code> 패턴을 사용합니다. <code>bugfix/<ticket number>_<developer initials>_<descriptor></code> 2. 개발자는 문제를 수정하고 수정 사항을 커밋한 다음 브랜치를 빌드합니다. <code>bugfix</code> 3. 개발자는 스쿼시 병합을 사용하여 <code>bugfix</code> 브랜치에서 <code>release/v<number></code> 브랜치로 병합 요청을 생성합니다. 4. 개발자는 <code>release</code> 브랜치를 빌드하여 다른 환경에서 재사용할 수 있도록 아티팩트를 게시합니다. 5. 승인자는 테스트 환경에 릴리스 아티팩트를 배포하는 것을 수동으로 승인합니다. 6. 승인자는 Stage 환경에 릴리스 아티팩트를 배포하는 것을 수동으로 승인합니다. 7. 승인자는 프로덕션 환경에 릴리스 아티팩트를 배포하는 것을 수동으로 승인합니다. 8. 개발자는 <code>release</code> 브랜치를 브랜치에 병합합니다. <code>main</code> 개발자가 자동화된 스 	DevOps 엔지니어

작업	설명	필요한 기술
	<p>크립트를 사용하여 빠른 병합을 수행하는 것이 이상적입니다. 스쿼시 병합은 사용하지 마세요.</p> <p>9. 개발자가 release 브랜치를 브랜치에 병합합니다. develop 개발자가 자동화된 스크립트를 사용하여 빠른 병합을 수행하는 것이 이상적입니다. 스쿼시 병합은 사용하지 마세요.</p> <p>10. 충돌이 감지되면 개발자는 알림을 받고 병합 요청을 통해 충돌을 해결합니다.</p>	

문제 해결

문제	Solution
브랜치 충돌	<p>Gitflow 모델에서 발생할 수 있는 일반적인 문제는 프로덕션 환경에서 핫픽스를 생성해야 하지만 다른 브랜치가 동일한 리소스를 수정하는 하위 환경에서 해당 변경이 발생해야 하는 경우입니다. 한 번에 하나의 릴리스 브랜치만 활성화하는 것이 좋습니다. 한 번에 둘 이상의 활성 상태가 있는 경우 환경의 변경 사항이 충돌하여 브랜치를 프로덕션으로 전환하지 못할 수 있습니다.</p>
병합	<p>릴리스를 기본 브랜치에 다시 병합하고 가능한 빨리 개발하여 작업을 기본 브랜치로 다시 통합해야 합니다.</p>
스쿼시 머징	<p>브랜치에서 feature 브랜치로 병합할 때만 스쿼시 머지를 사용하세요. develop 상위 브랜치</p>

문제	Solution
	에서 스쿼시 병합을 사용하면 변경 내용을 하위 브랜치에 다시 병합할 때 어려움이 발생합니다.

관련 리소스

이 가이드에는 Git에 대한 교육이 포함되어 있지 않지만, 이 교육이 필요한 경우 인터넷에서 많은 고품질 리소스를 이용할 수 있습니다. [Git 설명서](#) 사이트에서 시작하는 것이 좋습니다.

다음 리소스는 Gitflow 브랜칭 여정에 도움이 될 수 있습니다. AWS 클라우드

AWS DevOps 지침

- [AWS DevOps 지침](#)
- [AWS 배포 파이프라인 참조 아키텍처](#)
- [무엇입니까 DevOps?](#)
- [DevOps 리소스](#)

깃플로우 지침

- [오리지널 Gitflow 블로그 \(빈센트 드리센 블로그 게시물\)](#)
- [깃플로우 워크플로 \(아틀라시안\)](#)
- [Gitflow on GitHub: GitHub 기반 리포지토리에서 Git Flow 워크플로를 사용하는 방법 \(비디오\) YouTube](#)
- [Git 플로우 초기화 예제 \(비디오\) YouTube](#)
- [Gitflow 릴리즈 브랜치를 처음부터 끝까지 \(비디오\) YouTube](#)

기타 리소스

[트웰팩터 앱 방법론 \(12factor.net\)](#)

다중 DevOps 계정 환경을 위한 트렁크 브랜칭 전략 구현

작성자: 마이크 스티븐스 (AWS) 와 레이잔 윌슨 (AWS)

코드 리포지토리: [-멀티어카운트-데브옵스 git-branching-strategies-for](#)

환경: 프로덕션

기술: DevOps; 소프트웨어 개발 및 테스트; 다중 계정 전략

AWS 서비스: AWS CodeArtifact, AWS CodeBuild, AWS CodeCommit, AWS CodeDeploy, AWS CodePipeline

요약

소스 코드 리포지토리를 관리할 때는 다양한 브랜칭 전략이 개발 팀이 사용하는 소프트웨어 개발 및 릴리스 프로세스에 영향을 미칩니다. 일반적인 분기 전략의 예로는 트렁크, GitHub 플로우, Gitflow 등이 있습니다. 이러한 전략은 서로 다른 브랜치를 사용하며 각 환경에서 수행되는 활동도 다릅니다. DevOps 프로세스를 구현하는 조직은 이러한 분기 전략 간의 차이점을 이해하는 데 도움이 되는 시각적 가이드를 유용하게 활용할 수 있습니다. 조직에서 이 비주얼을 사용하면 개발팀이 업무를 조율하고 조직 표준을 준수하는 데 도움이 됩니다. 이 패턴은 이러한 시각적 정보를 제공하며 조직에서 트렁크 브랜칭 전략을 구현하는 프로세스를 설명합니다.

이 패턴은 여러 조직이 있는 조직을 위한 DevOps 분기 전략을 선택하고 구현하는 방법에 대한 설명서 시리즈의 일부입니다. AWS 계정 이 시리즈는 처음부터 올바른 전략과 모범 사례를 적용하여 클라우드에서의 경험을 간소화하는 데 도움이 되도록 설계되었습니다. 트렁크는 조직에서 사용할 수 있는 분기 전략 중 하나일 뿐입니다. 이 설명서 시리즈에서는 [GitHub Flow 및 Gitflow 분기 모델도](#) 다릅니다. 아직 검토하지 않았다면 이 패턴의 지침을 구현하기 전에 [다중 계정 DevOps 환경을 위한 Git 브랜칭 전략 선택](#)을 검토하는 것이 좋습니다. 실사를 통해 조직에 적합한 브랜칭 전략을 선택하시기 바랍니다.

이 가이드는 조직이 트렁크 전략을 구현하는 방법을 보여주는 다이어그램을 제공합니다. 공식 [AWS DevOps Well-Architected](#) 지침을 검토하여 모범 사례를 검토하는 것이 좋습니다. 이 패턴에는 프로세스의 각 단계에 대한 권장 작업, 단계 및 제한 사항이 포함됩니다. DevOps

사전 조건 및 제한 사항

사전 조건

- [Git이 설치되었습니다.](#) 이것은 소스 코드 리포지토리 도구로 사용됩니다.
- [Draw.io가 설치되었습니다.](#) 이 응용 프로그램은 다이어그램을 보고 편집하는 데 사용됩니다.

아키텍처

대상 아키텍처

다음 다이어그램은 [퍼넷 스퀘어](#) (Wikipedia) 처럼 사용할 수 있습니다. 세로 축의 분기를 가로 축의 AWS 환경과 정렬하여 각 시나리오에서 수행할 작업을 결정합니다. 숫자는 워크플로우에서의 작업 순서를 나타냅니다. 이 예제에서는 feature 브랜치부터 프로덕션 환경으로의 배포까지 안내합니다.

트렁크 접근 방식의 AWS 계정, 환경 및 브랜치에 대한 자세한 내용은 다중 계정 환경을 [위한 Git 브랜칭 전략 선택](#)을 참조하십시오. DevOps

자동화 및 규모 조정

지속적 통합 및 지속적 전달 (CI/CD) 은 소프트웨어 릴리스 라이프사이클을 자동화하는 프로세스입니다. 새 코드를 처음 커밋한 후 프로덕션에 이르기까지 기존에 필요했던 수동 프로세스의 대부분 또는 전체를 자동화합니다. CI/CD 파이프라인은 샌드박스, 개발, 테스트, 스테이징 및 프로덕션 환경을 포함합니다. 각 환경에서 CI/CD 파이프라인은 코드를 배포하거나 테스트하는 데 필요한 모든 인프라를 제공합니다. 개발팀은 CI/CD를 사용하여 코드를 변경한 다음 자동으로 테스트 및 배포할 수 있습니다. 또한 CI/CD 파이프라인은 기능 수용 및 배포에 대한 일관성, 표준, 모범 사례, 최소 허용 수준을 적용함으로써 개발 팀을 위한 거버넌스와 보호 수단을 제공합니다. 자세한 내용은 [지속적 통합 및 지속적 전달 실천](#)을 참조하십시오. AWS

AWS CI/CD 파이프라인을 구축하는 데 도움이 되도록 설계된 개발자 서비스 제품군을 제공합니다. 예를 들어 [AWS CodePipeline](#)는 빠르고 안정적인 애플리케이션 및 인프라 업데이트를 위해 릴리스 파이프라인을 자동화하는 데 도움이 되는 완전관리형 지속적 전달 서비스입니다. [AWS CodeCommit](#) 확장 가능한 Git 리포지토리를 안전하게 호스팅하고, 소스 코드를 [AWS CodeBuild](#) 컴파일하고, 테스트를 실행하고, 소프트웨어 패키지를 생성하도록 설계되었습니다. ready-to-deploy [자세한 내용은 개발자 도구 on](#)을 참조하십시오. AWS

도구

AWS 서비스 및 도구

AWS 이 패턴을 구현하는 데 사용할 수 있는 개발자 서비스 제품군을 제공합니다.

- [AWS CodeArtifact](#) 애플리케이션 개발을 위한 소프트웨어 패키지를 저장하고 공유하는 데 도움이 되는 확장성이 뛰어난 관리형 아티팩트 리포지토리 서비스입니다.
- [AWS CodeBuild](#) 소스 코드를 컴파일하고, 단위 테스트를 실행하고, 배포할 준비가 된 아티팩트를 생성하는 데 도움이 되는 완전 관리형 빌드 서비스입니다.
- [AWS CodeCommit](#) 자체 소스 제어 시스템을 관리할 필요 없이 Git 리포지토리를 비공개로 저장하고 관리할 수 있는 버전 제어 서비스입니다.
- [AWS CodeDeploy](#) 아마존 Elastic Compute Cloud (Amazon EC2) 또는 온프레미스 인스턴스 AWS Lambda, 함수 또는 아마존 Elastic Container Service (Amazon ECS) 서비스로의 배포를 자동화합니다.
- [AWS CodePipeline](#) 소프트웨어 릴리스의 여러 단계를 신속하게 모델링 및 구성하고 소프트웨어 변경 사항을 지속적으로 릴리스하는 데 필요한 단계를 자동화할 수 있도록 지원합니다.

기타 도구

- [Draw.io Desktop](#) — 플로우차트와 다이어그램을 만드는 데 사용되는 애플리케이션입니다.
- [Figma](#)는 협업을 위해 설계된 온라인 디자인 도구입니다. 코드 리포지토리에는 Figma용.fig 형식의 템플릿이 포함되어 있습니다.

코드 리포지토리

이 패턴의 다이어그램에 대한 이 소스 파일은 Trunk용 GitHub [Git 브랜칭 전략](#) 리포지토리에서 사용할 수 있습니다. 여기에는 PNG, draw.io 및 Figma 형식의 파일이 포함됩니다. 조직의 프로세스를 지원하도록 이러한 다이어그램을 수정할 수 있습니다.

모범 사례

[AWS DevOps Well-Architected 지침 및 다중 계정 환경을 위한 Git 브랜칭 전략 선택의](#) 모범 사례 및 권장 사항을 따르세요. DevOps 이를 통해 트렁크 기반 개발을 효과적으로 구현하고, 협업을 촉진하고, 코드 품질을 개선하고, 개발 프로세스를 간소화할 수 있습니다.

에픽

트렁크 워크플로 검토

작업	설명	필요한 기술
표준 트렁크 프로세스를 검토하십시오.	<ol style="list-style-type: none"> 1. 샌드박스 환경에서 개발자는 브랜치에서 feature 브랜치를 만들고 이름 지정 패턴을 사용합니다. <code>main feature/<ticket>_<initials>_<short description></code> 2. 개발자는 티켓을 완료하기 위해 코드를 개발하고 샌드박스 환경에 코드를 반복적으로 배포합니다. 참고: 개발자는 선택적으로 sandbox 브랜치를 만들어 샌드박스 환경에서 자동화된 빌드 또는 배포 파이프라인을 실행할 수 있습니다. 3. 개발자는 스쿼시 병합을 사용하여 feature 브랜치에서 main 브랜치로 병합 요청을 생성합니다. 4. 지속적 통합 및 지속적 전달 (CI/CD) 파이프라인은 main 브랜치의 아티팩트를 자동으로 빌드하고 개발 환경에 게시합니다. 5. 승인자는 릴리스 아티팩트를 개발 환경에 배포하는 것을 수동으로 승인합니다. 	DevOps 엔지니어

작업	설명	필요한 기술
	<p>6. 승인자는 테스트 환경에 릴리스 아티팩트를 배포하는 것을 수동으로 승인합니다.</p> <p>7. 승인자는 스테이징 환경에 릴리스 아티팩트를 배포하는 것을 수동으로 승인합니다.</p> <p>8. 승인자는 프로덕션 환경에 릴리스 아티팩트를 배포하는 것을 수동으로 승인합니다.</p>	

문제 해결

문제	Solution
브랜치 충돌	<p>트렁크 모델에서 발생할 수 있는 일반적인 문제는 프로덕션 환경에서 핫픽스를 생성해야 하지만 동일한 리소스가 수정되는 feature 브랜치에서 해당 변경이 발생해야 하는 경우입니다. 병합 시 심각한 충돌을 피하려면 하위 브랜치의 변경 내용을 자주 병합하는 것이 main 좋습니다.</p> <p>main</p>

관련 리소스

이 가이드에는 Git에 대한 교육이 포함되어 있지 않지만, 이 교육이 필요한 경우 인터넷에서 많은 고품질 리소스를 이용할 수 있습니다. [Git 설명서](#) 사이트에서 시작하는 것이 좋습니다.

다음 리소스는 Trunk 브랜칭 여정에 도움이 될 수 있습니다. AWS 클라우드

AWS DevOps 지침

- [AWS DevOps 지침](#)

- [AWS 배포 파이프라인 참조 아키텍처](#)
- [무엇입니까 DevOps?](#)
- [DevOps 리소스](#)

트렁크 가이드

- [트렁크 기반 개발](#)

기타 리소스

- [트웰-팩터 앱 방법론 \(12factor.net\)](#)

변경 사항을 자동으로 감지하고 단일 저장소에 대한 다양한 CodePipeline 파이프라인을 시작합니다. CodeCommit

제작: 헬튼 리베이로 (AWS), 페트루스 바탈라 (AWS), 리카르도 모라이스 (AWS)

코드 리포지토리: [AWS CodeCommit 모노레포](#) 멀티파 이프라인 트리거

환경: PoC 또는 파일럿

기술:; 인프라; 서버리스 DevOps

AWS 서비스: AWS, AWS CodeCommit CodePipeline, AWS Lambda

요약

이 패턴을 사용하면 모노레포 기반 애플리케이션의 소스 코드 변경을 자동으로 감지한 다음 각 마이크로서비스에 대한 지속적 통합 AWS CodeCommit 및 지속적 전송 (CI/CD) 자동화를 실행하는 파이프라인을 시작할 수 있습니다. AWS CodePipeline 이 접근 방식은 모노리포지토리 기반 애플리케이션의 각 마이크로서비스가 전용 CI/CD 파이프라인을 가질 수 있다는 것을 의미하며, 이를 통해 가시성을 높이고, 코드를 더 쉽게 공유하고, 협업, 표준화 및 검색 가능성을 개선할 수 있습니다.

이 패턴에 설명된 솔루션은 모노레포 내 마이크로서비스 간의 종속성 분석을 수행하지 않습니다. 소스 코드의 변경 사항만 감지하고 일치하는 CI/CD 파이프라인을 시작합니다.

패턴은 통합 개발 환경 (IDE) AWS Cloud9 으로 사용되며, 두 AWS CloudFormation 스택, 즉 및 AWS Cloud Development Kit (AWS CDK) 를 사용하여 인프라를 정의합니다. MonoRepoStack PipelinesStack MonoRepoStack스택은 모노레포를 생성하고 CI/CD AWS CodeCommit 파이프라인을 시작하는 AWS Lambda 함수를 생성합니다. PipelinesStack 스택은 파이프라인 인프라를 정의합니다.

중요: 이 패턴의 워크플로는 개념 증명 (PoC) 입니다. 테스트 환경에서만 사용하는 것이 좋습니다. 프로덕션 환경에서 이 패턴의 접근 방식을 사용하려면 [AWS Identity and Access Management \(IAM\) 설명서의 IAM의 보안 모범 사례](#)를 참조하고 IAM 역할을 필요에 따라 변경하십시오. AWS 서비스

사전 조건 및 제한 사항

사전 조건

- 활성 계정. AWS
- AWS Command Line Interface (AWS CLI), 설치 및 구성 자세한 내용은 AWS CLI 설명서의 [설치, 업데이트 및 제거를 AWS CLI](#) 참조하십시오.
- Python 3 및 pip 로컬 머신에 설치. 자세한 내용은 [Python 설명서](#)를 참조하세요.
- AWS CDK, 설치 및 구성. 자세한 내용은 AWS CDK 설명서의 ["사용 시작하기 AWS CDK"](#)를 참조하십시오.
- 설치 및 구성된 AWS Cloud9 IDE 자세한 내용은 AWS Cloud9 AWS Cloud9 설명서의 [설정을](#) 참조하십시오.
- GitHub [AWS CodeCommit monorepo 다중 파이프라인은 로컬 시스템에 복제된 리포지토리를 트리거합니다.](#)
- 빌드하고 배포하는 데 사용할 애플리케이션 코드가 들어 있는 기존 디렉터리입니다. CodePipeline
- 의 DevOps 모범 사례에 익숙하고 경험이 있어야 합니다. AWS 클라우드친속도를 높이려면 [DevOps 프랙티스를 사용하여 마이크로서비스와 느슨하게 결합된 아키텍처 구축하기 패턴 및 규범적 지침 웹 사이트](#)를 사용할 수 AWS Cloud9있습니다. DevOps AWS

아키텍처

다음 다이어그램은 를 사용하여 두 개의 스택으로 구성된 인프라를 정의하는 AWS CDK 방법을 보여줍니다. AWS CloudFormation MonoRepoStack PipelinesStack

이 다이어그램은 다음 워크플로를 보여줍니다.

1. 부트스트랩 프로세스는 AWS CDK 를 사용하여 AWS CloudFormation MonoRepoStack 스택과 을 생성합니다. PipelinesStack
2. MonoRepoStack스택은 애플리케이션과 각 커밋 후에 시작되는 monorepo-event-handler Lambda 함수를 위한 CodeCommit 리포지토리를 생성합니다.
3. PipelinesStack스택은 Lambda 함수에 CodePipeline 의해 시작되는 파이프라인을 생성합니다. 각 마이크로서비스에는 정의된 인프라 파이프라인이 있어야 합니다.
4. 의 파이프라인은 Lambda 함수에 의해 시작되고 의 소스 코드를 기반으로 격리된 CI/CD 단계를 시작합니다. microservice-n CodeCommit
5. 의 파이프라인은 Lambda 함수에 의해 시작되고 의 소스 코드를 기반으로 격리된 CI/CD 단계를 시작합니다. microservice-1 CodeCommit

다음 다이어그램은 AWS CloudFormation MonoRepoStack 스택과 계정의 배포를 보여줍니다.
PipelinesStack

1. 사용자가 애플리케이션의 마이크로서비스 중 하나에서 코드를 변경합니다.
2. 사용자가 로컬 리포지토리에서 리포지토리로 변경 내용을 푸시합니다. CodeCommit
3. 푸시 활동은 리포지토리로 보내는 모든 푸시를 수신하는 Lambda 함수를 시작합니다. CodeCommit
4. Lambda 함수는 파라미터 스토어에서 파라미터를 읽고, AWS Systems Manager의 기능으로, 가장 최근의 커밋 ID를 검색합니다. 파라미터의 이름 지정 형식은 다음과 같습니다. / MonoRepoTrigger/{repository}/{branch_name}/LastCommit 파라미터를 찾을 수 없는 경우 Lambda 함수는 리포지토리에서 마지막 커밋 ID를 읽고 반환된 값을 Parameter Store에 저장합니다. CodeCommit
5. 커밋 ID와 변경된 파일을 식별한 후 Lambda 함수는 각 마이크로서비스 디렉터리의 파이프라인을 식별하고 필요한 파이프라인을 시작합니다. CodePipeline

도구

- [AWS Cloud Development Kit \(AWS CDK\)](#) 코드로 클라우드 인프라를 정의하고 이를 통해 프로비저닝하기 위한 소프트웨어 개발 프레임워크입니다. AWS CloudFormation
- [Python](#)은 빠르게 작업하고 시스템을 더 효과적으로 통합할 수 있게 해주는 프로그래밍 언어입니다.

code

이 패턴의 소스 코드와 템플릿은 GitHub [AWS CodeCommit monorepo 다중 파이프라인](#) 트리거 리포지토리에서 사용할 수 있습니다.

모범 사례

- 이 샘플 아키텍처에는 배포된 인프라에 대한 모니터링 솔루션이 포함되어 있지 않습니다. 이 솔루션을 프로덕션 환경에 배포하려면 모니터링을 활성화하는 것이 좋습니다. 자세한 내용은 AWS Serverless Application Model (AWS SAM) 설명서의 [CloudWatch Application Insights를 사용하여 서버리스 애플리케이션 모니터링](#)을 참조하십시오.
- 이 패턴에서 제공하는 샘플 코드를 편집할 때는 설명서에 있는 [클라우드 인프라 개발 및 배포에 대한 모범 사례](#)를 따르세요. AWS CDK

- 마이크로서비스 파이프라인을 정의할 때는 설명서의 [보안 모범 사례를](#) 검토하세요. AWS CodePipeline
- [cdk-nag](#) 유틸리티를 사용하여 AWS CDK 코드에서 모범 사례를 확인할 수도 있습니다. 이 도구는 팩 별로 그룹화된 일련의 규칙을 사용하여 코드를 평가합니다. 사용 가능한 팩은 다음과 같습니다.
 - [AWS 솔루션 라이브러리](#)
 - [건강 보험 양도 및 책임에 관한 법률 \(HIPAA\) 보안](#)
 - [국립 표준 기술 연구소 \(NIST\) 800-53 rev 4](#)
 - [NIST 800-53 버전 5](#)
 - [지불 카드 산업 데이터 보안 표준\(PCI DSS\) 3.2.1](#)

에픽

환경 설정

작업	설명	필요한 기술
가상 Python 환경을 생성합니다.	AWS Cloud9 IDE에서 다음 명령을 실행하여 가상 Python 환경을 만들고 필요한 종속성을 설치합니다. make install	개발자
AWS 계정 와 를 AWS 리전 위한 부트스트랩을 실행하십시오. AWS CDK	다음 명령을 실행하여 필수 AWS 계정 및 지역을 부트스트랩합니다. make bootstrap account-id=<your-AWS-account-ID> region=<required-region>	개발자

마이크로서비스를 위한 새 파이프라인 추가

작업	설명	필요한 기술
애플리케이션 디렉터리에 샘플 코드를 추가합니다.	샘플 애플리케이션 코드가 들어 있는 디렉터리를 복제된 GitHub AWS CodeCommit 모노레포 멀티파이프라인 트리거 monorepo-sample 리포지토리의 디렉터리에 추가합니다.	개발자
monorepo-main.json 파일을 편집합니다.	애플리케이션 코드의 디렉터리 이름과 파이프라인 이름을 복제된 리포지토리의 monorepo-main.json 파일에 추가합니다.	개발자
파이프라인을 생성하십시오.	리포지토리의 Pipelines 디렉터리에 애플리케이션의 class 파이프라인을 추가합니다. 디렉터리에는 두 개의 샘플 파일 pipeline_hotsite.py 및 이 포함되어 pipeline_demo.py 있습니다. 각 파일에는 소싱, 빌드, 배포의 세 단계가 있습니다. 애플리케이션 요구 사항에 따라 파일 중 하나를 복사하고 변경할 수 있습니다.	개발자
monorepo_config.py 파일을 편집합니다.	service_map 에서 애플리케이션의 디렉터리 이름과 파이프라인용으로 만든 클래스를 추가합니다.	개발자

작업	설명	필요한 기술
	<p>예를 들어, 다음 코드는 MySamplePipeline 클래스로 이름이 pipeline_mysample.py 로 지정된 파일을 사용하는 Pipelines 디렉터리의 파이프라인 정의를 보여줍니다.</p> <pre data-bbox="594 569 1027 1686"> ... # Pipeline definition imports from pipelines .pipeline_demo import DemoPipeline from pipelines.pipeline _hotsite import HotsitePipeline from pipelines .pipeline_mysample import MySampleP ipeline ### Add your pipeline configuration here service_map: Dict[str, ServicePipeline] = { # folder-name -> pipeline-class 'demo': DemoPipel ine(), 'hotsite': HotsitePipeline(), 'mysample': MySamplePipeline() } </pre>	

MonoRepoStack 스택 배포

작업	설명	필요한 기술
<p>AWS CloudFormation 스택을 배포하세요.</p>	<p>make deploy-core 명령을 실행하여 복제된 리포지토리의 루트 디렉터리에 기본 파라미터 값이 있는 AWS CloudFormation MonoRepoStack 스택을 배포합니다.</p> <p>make deploy-core monorepo-name=<repo_name> 명령을 실행하여 리포지토리 이름을 변경할 수 있습니다.</p> <p>참고: make deploy monorepo-name=<repo_name> 명령을 사용하여 두 파이프라인을 동시에 배포할 수 있습니다.</p>	개발자
<p>CodeCommit 리포지토리를 검증합니다.</p>	<p>aws codecommit get-repository --repository-name <repo_name> 명령을 실행하여 리소스가 생성되었는지 확인합니다.</p> <p>중요: AWS CloudFormation 스택은 모노레포가 CodeCommit 저장되는 저장소를 생성하므로 수정 내용을 푸시하기 시작한 경우에는 cdk destroy MonoRepoStack 명령을 실행하지 마세요.</p>	개발자

작업	설명	필요한 기술
AWS CloudFormation 스택 결과를 검증하세요.	<p>다음 명령을 실행하여 AWS CloudFormation MonoRepoStack 스택이 올바르게 생성되고 구성되었는지 확인합니다.</p> <pre>aws cloudformation list-stacks -- stack-status-filter CREATE_COMPLETE -- query 'StackSummaries[? StackName == 'MonoRepo Stack']'</pre>	개발자

PipelinesStack 스택을 배포합니다.

작업	설명	필요한 기술
AWS CloudFormation 스택을 배포하세요.	<p>AWS CloudFormation PipelinesStack 스택을 배포한 후에는 MonoRepoStack 스택을 배포해야 합니다. 새 마이크로서비스가 모노리포지토리의 코드 베이스에 추가되면 스택의 크기가 증가하고 새 마이크로서비스가 온보딩되면 재배포됩니다.</p> <pre>make deploy-pi pelines</pre> <p>명령을 실행하여 PipelinesStack 스택을 배포합니다.</p> <p>참고: <code>make deploy monorepo-name=<rep</code></p>	개발자

작업	설명	필요한 기술
	<p>o_name> 명령을 실행하여 두 파이프라인을 동시에 배포할 수도 있습니다.</p> <p>다음 샘플 출력은 구현 종료 시 PipelinesStacks 배포가 마이크로서비스의 URL을 인쇄하는 방법을 보여줍니다.</p> <pre>Outputs: PipelinesStack.dem ourl = .cloudfront.net PipelinesStack.hotsi teurl = .cloudfro nt.net</pre>	
<p>AWS CloudFormation 스택 결과를 확인합니다.</p>	<p>다음 명령을 실행하여 AWS CloudFormation PipelinesStacks 스택이 올바르게 생성되고 구성되었는지 확인합니다.</p> <pre>aws cloudformation list-stacks --stack-s tatus-filter CREATE_CO MPLETE UPDATE_COMPLETE --query 'StackSum maries[?StackName == 'PipelinesStack']'</pre>	<p>개발자</p>

리소스 정리

작업	설명	필요한 기술
<p>AWS CloudFormation 스택을 삭제합니다.</p>	<p>make destroy 명령을 실행합니다.</p>	<p>개발자</p>

작업	설명	필요한 기술
파이프라인의 S3 버킷을 삭제 하십시오.	<ol style="list-style-type: none"> 1. AWS Management Console 로그인하고 Amazon 심플 스토리지 서비스 (Amazon S3) 콘솔을 엽니다. 2. 파이프라인과 연결된 S3 버킷을 삭제하고 다음 이름을 사용합니다. pipelines-stack-codepipeline* 	개발자

문제 해결

문제	Solution
문제가 발생했습니다 AWS CDK .	AWS CDK 설명서의 일반적인 AWS CDK 문제 해결 을 참조하십시오.
마이크로서비스 코드를 푸시했지만 마이크로서비스 파이프라인이 실행되지 않았습니다.	<p>설정 유효성 검사</p> <p>브랜치 구성 확인:</p> <ul style="list-style-type: none"> • 코드를 올바른 브랜치로 푸시하고 있는지 확인하세요. 이 파이프라인은 main 브랜치가 변경될 때만 실행되도록 구성되어 있습니다. 다른 브랜치로 푸시해도 특별히 구성되지 않는 한 파이프라인이 시작되지 않습니다. • 코드를 푸시한 후 커밋이 보이는지 확인하여 푸시가 성공했는지, 로컬 환경과 리포지토리 간의 연결이 손상되지 않았는지 확인하세요. AWS CodeCommit 코드를 푸시하는 데 문제가 있는 경우 자격 증명을 새로 고치십시오. <p>구성 파일 확인:</p>

문제	Solution
	<ul style="list-style-type: none"> 의 <code>service_map</code> 변수가 마이크로서비스의 현재 디렉터리 구조를 <code>monorepo_config.py</code> 정확하게 반영하는지 확인하십시오. 이 변수는 코드 푸시를 해당 파이프라인에 매핑하는 데 중요한 역할을 합니다. 마이크로서비스를 위한 새 매핑을 포함하도록 <code>monorepo-main.json</code> 업데이트되었는지 확인하세요. 이 파일은 파이프라인이 마이크로서비스의 변경 사항을 인식하고 올바르게 처리하는 데 필수적입니다. <p>콘솔에서의 문제 해결</p> <p>AWS CodePipeline 점검:</p> <ul style="list-style-type: none"> 에서 AWS Management Console 파이프라인이 호스팅되는 AWS 리전 위치에 있는지 확인하세요. CodePipeline 콘솔을 열고 마이크로서비스에 해당하는 파이프라인이 시작되었는지 확인하세요. <p>오류 분석: 파이프라인이 시작되었지만 실패한 경우 에서 제공한 오류 메시지나 로그를 CodePipeline 검토하여 무엇이 잘못되었는지 파악하세요.</p> <p>AWS Lambda 문제 해결:</p> <ul style="list-style-type: none"> AWS Lambda 콘솔에서 <code>monorepo-event-handler</code> Lambda 함수를 엽니다. 코드 푸시에 대한 응답으로 함수가 시작되었는지 확인하십시오. <p>로그 분석: Lambda 함수의 로그에서 문제가 있는지 검사하십시오. 로그는 함수가 실행되었을 때 어떤 일이 발생했는지에 대한 자세한</p>

문제	Solution
	정보를 제공하고 함수가 예상대로 이벤트를 처리했는지 여부를 식별하는 데 도움이 될 수 있습니다.

문제	Solution
<p>모든 마이크로서비스를 재배포해야 합니다.</p>	<p>모든 마이크로서비스를 강제로 재배포하는 방법에는 두 가지가 있습니다. 요구 사항에 맞는 옵션을 선택하세요.</p> <p>접근법 1: 파라미터 스토어에서 파라미터 삭제</p> <p>이 방법에는 배포에 사용된 마지막 커밋 ID를 추적하는 Systems Manager Parameter Store에서 특정 매개 변수를 삭제하는 작업이 포함됩니다. 이 파라미터를 제거하면 시스템은 새로운 상태로 인식하기 때문에 다음 트리거 시 모든 마이크로서비스를 재배포해야 합니다.</p> <p>단계:</p> <ol style="list-style-type: none"> 1. 모노레포의 커밋 ID 또는 관련 배포 마커가 들어 있는 특정 파라미터 스토어 항목을 찾으세요. 파라미터 이름은 다음 형식을 따릅니다. <code>"/MonoRepoTrigger/{repository}/{branch_name}/LastCommit"</code> 2. 매개 변수 값이 중요하거나 재설정하기 전에 배포 상태를 기록해 두려면 매개 변수 값을 백업해 두는 것이 좋습니다. 3. AWS Management Console AWS CLI, 또는 SDK를 사용하여 식별된 파라미터를 삭제하십시오. 이 작업을 수행하면 배포 마커가 재설정됩니다. 4. 삭제 후 다음 번에 리포지토리로 푸시하면 시스템이 모든 마이크로서비스를 배포해야 합니다. 배포 시 고려할 최신 커밋을 찾기 때문입니다. <p>장점:</p>

문제	Solution
	<ul style="list-style-type: none"> • 최소한의 단계로 간단하고 빠르게 구현할 수 있습니다. • 배포를 시작하기 위해 코드를 임의로 변경할 필요가 없습니다. <p>단점:</p> <ul style="list-style-type: none"> • 배포 프로세스에 대한 세밀한 제어가 쉽지 않습니다. • 파라미터 저장소를 다른 중요 구성을 관리하는 데 사용하는 경우 잠재적으로 위험할 수 있습니다. <p>접근법 2: 각 monorepo 하위 폴더에 커밋을 푸시합니다.</p> <p>이 방법은 약간만 변경한 다음 이를 모노레포 내의 각 마이크로서비스 하위 폴더에 푸시하여 개별 파이프라인을 시작하는 것입니다.</p> <p>단계:</p> <ol style="list-style-type: none"> 1. 모노레포 내에서 재배포가 필요한 모든 마이크로서비스를 나열하세요. 2. 각 마이크로서비스에 대해 해당 하위 폴더를 영향을 주지 않고 최소한으로 변경하십시오. README파일 업데이트, 구성 파일에 설명 추가 또는 서비스 기능에 영향을 주지 않는 변경 등이 이에 해당할 수 있습니다. 3. 명확한 메시지 (예: “마이크로서비스 재배포 시작”)와 함께 이러한 변경 사항을 적용하고 리포지토리로 푸시하십시오. 배포를 시작하는 브랜치에 변경 내용을 푸시해야 합니다.

문제	Solution
	<p>4. 각 마이크로서비스의 파이프라인을 모니터링 하여 성공적으로 시작되고 완료되었는지 확인하세요.</p> <p>장점:</p> <ul style="list-style-type: none"> • 재배포할 마이크로서비스를 세밀하게 제어할 수 있습니다. • 다른 용도로 사용될 수 있는 구성 파라미터를 삭제할 필요가 없으므로 더 안전합니다. <p>단점:</p> <ul style="list-style-type: none"> • 특히 마이크로서비스가 많은 경우 시간이 더 많이 걸립니다. • 커밋 기록을 복잡하게 만들 수 있는 불필요한 코드 변경이 필요합니다.

관련 리소스

- [CDK Pipeline을 사용한 지속적 통합 및 전달 \(CI/CD\)](#) (설명서)AWS CDK
- [aws-cdk/파이프라인 모듈](#) (API 레퍼런스)AWS CDK

AWS를 사용하여 비트버킷 리포지토리를 AWS Amplify와 통합 CloudFormation

작성자: Alwin Abraham(AWS)

환경: 프로덕션

기술: DevOps

AWS 서비스: AWS Amplify,
AWS CloudFormation

요약

AWS Amplify를 사용하면 일반적으로 필요한 인프라를 설정하지 않고도 정적 웹 사이트를 빠르게 배포하고 테스트할 수 있습니다. 조직에서 기존 애플리케이션 코드를 마이그레이션하든 새 애플리케이션을 구축하든 관계없이 소스 제어에 Bitbucket을 사용하려는 경우 이 패턴의 접근 방식을 배포할 수 있습니다. CloudFormation AWS를 사용하여 Amplify를 자동으로 설정하면 사용 중인 구성에 대한 가시성을 제공할 수 있습니다.

이 패턴은 AWS를 사용하여 Bitbucket 리포지토리를 AWS CloudFormation Amplify와 통합하여 프론트 엔드 지속적 통합 및 지속적 배포 (CI/CD) 파이프라인 및 배포 환경을 만드는 방법을 설명합니다. 이 패턴의 접근 방식은 반복 가능한 배포를 위한 Amplify 프론트 엔드 파이프라인을 구축할 수 있음을 의미합니다.

사전 조건 및 제한 사항

사전 조건

- 활성 Amazon Web Services(AWS) 계정
- 관리자 액세스 권한이 있는 활성 Bitbucket 계정
- [cURL](#) 또는 [Postman](#) 애플리케이션을 사용하는 터미널에 액세스
- Amplify에 대한 지식
- AWS에 대한 지식 CloudFormation
- YAML 형식의 파일에 대한 지식

아키텍처

기술 스택

- Amplify
- AWS CloudFormation
- Bitbucket

도구

- [AWS Amplify](#) - Amplify는 개발자가 클라우드 기반 모바일 및 웹 앱을 개발하고 배포할 수 있도록 지원합니다.
- [AWS CloudFormation](#) — CloudFormation AWS는 AWS 리소스를 모델링하고 설정하여 리소스를 관리하는 시간을 줄이고 AWS에서 실행되는 애플리케이션에 더 많은 시간을 집중할 수 있도록 도와주는 서비스입니다.
- [Bitbucket](#) - Bitbucket은 전문가 팀을 위해 설계된 Git 리포지토리 관리 솔루션입니다. Git 리포지토리를 관리하고, 소스 코드에 대해 협업하고, 개발 흐름을 안내할 수 있는 중앙 장소를 제공합니다.

code

bitbucket-amplify.yml파일 (첨부) 에는 이 패턴에 대한 AWS CloudFormation 템플릿이 들어 있습니다.

에픽

Bitbucket 리포지토리 구성

작업	설명	필요한 기술
(선택 사항) Bitbucket 리포지토리를 생성합니다.	<ol style="list-style-type: none"> 1. Bitbucket 계정에 로그인하고 새 리포지토리를 생성합니다. 이에 대한 자세한 내용은 Bitbucket 설명서의 Git 리포지토리 생성을 참조하세요. 2. 작업 영역 이름을 기록합니다. 	DevOps 엔지니어

작업	설명	필요한 기술
	참고: 기존 Bitbucket 리포지토리를 사용할 수도 있습니다.	
워크스페이스 설정을 엽니다.	<ol style="list-style-type: none"> 1. 작업 영역을 열고 리포지토리 탭을 선택합니다. 2. Amplify와 통합할 리포지토리를 선택합니다. 3. 리포지토리 이름 위에 있는 작업 영역의 이름을 선택합니다. 4. 사이드바에서 설정을 선택합니다. 	DevOps 엔지니어

작업	설명	필요한 기술
OAuth 소비를 생성합니다.	<ol style="list-style-type: none"> 1. 앱 및 기능 섹션에서 OAuth 소비자를 선택한 다음 소비자 추가를 선택합니다. 2. 소비자의 이름을 입력합니다(예: Amplify Integration). 3. 콜백 URL을 입력합니다. 이 필드는 필수 입력이지만 통합을 완료하는 데 사용되지 않으므로 값은 <code>http://localhost:3000</code> 이 될 수 있습니다 4. 개인 소비자입니다 확인란에 체크 표시하세요. 5. 다음 권한을 선택합니다. <ul style="list-style-type: none"> • 프로젝트 – Read • 리포지토리 – Admin • 풀 요청 – Read • 웹훅 - Read 및 Write 6. 다른 필드를 모두 기본값으로 그대로 두고 제출을 선택합니다. 7. 생성된 키와 암호를 기록합니다 	DevOps 엔지니어

작업	설명	필요한 기술
OAuth 액세스 토큰을 확보합니다.	<p>1. 터미널 창을 열고 다음 명령을 실행합니다.</p> <pre>curl -X POST -u "KEY:SECRET" https://bitbucket.org/site/oauth2/access_token -d grant_type=client_credentials</pre> <p>중요: KEY 및 SECRET을 이전에 기록한 키와 비밀번호로 바꿉니다.</p> <p>2. 따옴표를 사용하지 않고 액세스 토큰을 기록합니다. 토큰은 제한된 시간에만 유효하며 기본 시간은 2시간입니다. 이 기간 내에 AWS CloudFormation 템플릿을 실행해야 합니다.</p>	DevOps 엔지니어

AWS CloudFormation 스택 생성 및 배포

작업	설명	필요한 기술
AWS CloudFormation 템플릿을 다운로드합니다.	<pre>bitbucket-amplify.yml</pre> <p>AWS CloudFormation 템플릿 다운로드 (첨부) 이 템플릿은 Amplify 프로젝트 및 브랜치 외에도 Amplify에서 CI/CD 파이프라인을 생성합니다.</p>	

작업	설명	필요한 기술
<p>AWS CloudFormation 스택을 생성하고 배포합니다.</p>	<ol style="list-style-type: none"> 1. 배포하려는 AWS 리전의 AWS 관리 콘솔에 로그인하고 AWS CloudFormation 콘솔을 엽니다. 2. 스택 생성(새 리소스 포함)을 선택한 다음 템플릿 파일 업로드를 선택합니다. 3. bitbucket-amplify.yml 파일을 업로드합니다. 4. 다음을 선택하고 스택 이름을 입력한 후 다음 파라미터를 입력합니다. <ul style="list-style-type: none"> • 액세스 토큰: 이전에 생성한 OAuth 액세스 토큰을 붙여넣습니다. • 리포지토리 URL: Bitbucket 프로젝트 리포지토리의 URL을 추가합니다. 일반적으로 URL은 <code>https://bitbucket.org/<WORKSPACE_NAME>/<REPO_NAME></code> 형식에 있습니다. • 브랜치 이름: Bitbucket 리포지토리의 브랜치 이름과 일치해야 합니다. 이 브랜치는 AWS CloudFormation 스택을 실행할 때 존재할 필요는 없지만 환경에 코드를 배포하는 데에는 필요합니다. 	<p>DevOps 엔지니어</p>

작업	설명	필요한 기술
	<ul style="list-style-type: none"> 프로젝트 이름: Amplify 프로젝트와 연결할 이름입니다. <p>5. 다음을 선택한 다음 스택 생성을 선택합니다.</p>	

CI/CD 파이프라인 테스트

작업	설명	필요한 기술
리포지토리의 브랜치에 코드를 배포합니다.	<ol style="list-style-type: none"> <code>git clone https://bitbucket.org/<WORKSPACE_NAME>/<REPO_NAME></code> 명령을 실행하여 Bitbucket 리포지토리를 복제합니다. AWS CloudFormation 스크립트를 실행할 때 사용된 브랜치 이름을 확인하십시오. 새 브랜치를 생성하고 확인하려면 <code>git checkout -b <BRANCH_NAME></code> 명령을 실행하세요. 기존 브랜치를 확인하려면 <code>git checkout <BRANCH_NAME></code> 명령을 실행하세요. 코드를 브랜치에 커밋하고 <code>git commit</code> 및 <code>git push</code> 명령을 실행하여 원격 브랜치로 푸시합니다. 	앱 개발자

작업	설명	필요한 기술
	<p>4. 그런 다음 Amplify는 애플리케이션을 빌드하고 배포합니다.</p> <p>자세한 내용은 Bitbucket 설명서의 기본 Git 명령을 참조하세요.</p>	

관련 리소스

[인증 방법](#)(Atlassian 설명서)

첨부

이 문서와 관련된 추가 콘텐츠에 액세스하려면 [attachment.zip](#) 파일의 압축을 풉니다.

Step Functions와 Lambda 프록시 함수를 사용하여 여러 AWS 계정에서 CodeBuild 프로젝트를 시작합니다.

작성자: 리처드 밀너-와츠 (AWS) 와 아미트 안잘레카르 (AWS)

코드 [리포지토리 CodeBuild :
크로스 어카운트 프록시](#)

환경: 프로덕션

기술: DevOps; 관리 및 거버넌스; 운영; 서버리스

AWS 서비스: AWS CodeBuild, AWS Lambda, AWS Step Functions, AWS X-Ray, AWS CloudFormation

요약

이 패턴은 AWS Step Functions와 AWS Lambda 프록시 함수를 사용하여 CodeBuild 여러 AWS 계정에서 AWS 프로젝트를 비동기적으로 시작하는 방법을 보여줍니다. 패턴의 샘플 Step Functions 상태 머신을 사용하여 CodeBuild 프로젝트의 성공 여부를 테스트할 수 있습니다.

CodeBuild 완전 관리형 런타임 환경에서 AWS 명령줄 인터페이스 (AWS CLI) 를 사용하여 운영 작업을 시작할 수 있도록 도와줍니다. 환경 변수를 재정의하여 런타임에 CodeBuild 프로젝트 동작을 변경할 수 있습니다. 또한 워크플로를 관리하는 CodeBuild 데 사용할 수 있습니다. 자세한 내용은 AWS 워크숍 웹 사이트의 [서비스 카탈로그 도구](#) 및 AWS 데이터베이스 블로그의 [AWS EventBridge 및 Amazon을 CodeBuild 사용하는 PostgreSQL용 Amazon RDS의 작업 일정 잡기를 참조하십시오.](#)

사전 조건 및 제한 사항

사전 조건

- 활성 AWS 계정 2개: Step Functions를 사용하여 Lambda 프록시 함수를 호출하기 위한 소스 계정과 원격 샘플 프로젝트를 빌드하기 위한 대상 계정 CodeBuild

제한 사항

- 이 패턴은 계정 간에 [아티팩트](#)를 복사하는 데 사용할 수 없습니다.

아키텍처

다음 다이어그램은 이 패턴이 구축하는 아키텍처를 보여줍니다.

이 다이어그램은 다음 워크플로를 보여줍니다.

1. Step Functions 상태 머신은 제공된 입력 맵을 파싱하고 정의한 각 계정, 지역 및 프로젝트에 대해 Lambda 프록시 함수 (codebuild-proxy-lambda) 를 호출합니다.
2. Lambda 프록시 함수는 AWS 보안 토큰 서비스 (AWS STS) 를 사용하여 대상 계정의 IAM 정책 codebuild-proxy-role () 과 연결된 IAM 프록시 역할 codebuild-proxy-policy () 을 말합니다.
3. Lambda 함수는 위임된 역할을 사용하여 프로젝트를 시작하고 작업 CodeBuild ID를 반환합니다. CodeBuild Step Functions 상태 머신은 성공 또는 실패 상태를 수신할 때까지 CodeBuild 작업을 반복하고 폴링합니다.

상태 머신 로직은 다음 이미지에 나와 있습니다.

기술 스택

- AWS CloudFormation
- CodeBuild
- IAM
- Lambda
- Step Functions
- X-Ray

도구

- [AWS는 AWS CloudFormation](#) 리소스를 설정하고, 빠르고 일관되게 프로비저닝하고, AWS 계정 및 지역 전반의 수명 주기 전반에 걸쳐 리소스를 관리할 수 있도록 지원합니다.
- [AWS CloudFormation Designer](#)는 템플릿을 보고 편집하는 데 도움이 되는 통합 JSON 및 YAML CloudFormation 편집기를 제공합니다.

- [CodeBuildAWS](#)는 소스 코드를 컴파일하고, 단위 테스트를 실행하고, 배포할 준비가 된 아티팩트를 생성하는 데 도움이 되는 완전관리형 빌드 서비스입니다.
- [AWS Identity and Access Management\(IAM\)](#)를 사용하면 사용자에 대해 인증 및 권한 부여를 제어함으로써 AWS 리소스에 대한 액세스를 안전하게 관리할 수 있습니다.
- [AWS Lambda](#)는 서버를 프로비저닝하거나 관리할 필요 없이 코드를 실행하는 데 도움이 되는 컴퓨팅 서비스입니다. 필요할 때만 코드를 실행하며 자동으로 확장이 가능하므로 사용한 컴퓨팅 시간만큼만 비용을 지불합니다.
- [AWS Step Functions](#)는 AWS Lambda 함수와 기타 AWS 서비스를 결합할 수 있는 서버리스 오케스트레이션 서비스로, 비즈니스 크리티컬 애플리케이션을 구축합니다.
- [AWS X-Ray](#)는 애플리케이션이 처리하는 요청에 대한 데이터를 수집하는 웹 서비스이며 해당 데이터를 보고, 필터링하고, 통찰을 얻어 문제와 최적화 기회를 식별할 수 있는 도구를 제공합니다.

코드

이 패턴의 샘플 코드는 GitHub [Cross Account CodeBuild Proxy](#) 리포지토리에서 사용할 수 있습니다. 이 패턴은 AWS Lambda Powertools for Python 라이브러리를 사용하여 로깅 및 추적 기능을 제공합니다. 이 라이브러리 및 해당 유틸리티에 대한 자세한 내용은 [Powertools for AWS Lambda \(Python\)](#)를 참조하세요.

모범 사례

1. Step Function 상태 시스템에서 대기 시간 값을 조정하여 작업 상태에 대한 폴링 요청을 최소화하십시오. CodeBuild 프로젝트의 예상 실행 시간을 사용하세요.
2. Step Functions에서 맵의 MaxConcurrency 속성을 조정하여 병렬로 실행할 수 있는 CodeBuild 프로젝트 수를 제어합니다.
3. 필요한 경우 샘플 코드를 검토하여 프로덕션 준비 상태를 확인하세요. 솔루션에서 어떤 데이터를 기록할 수 있는지, 그리고 기본 Amazon CloudWatch 암호화가 충분한지 생각해 보십시오.

에픽

소스 계정에서 Lambda 프록시 함수 및 관련 IAM 역할을 생성합니다.

작업	설명	필요한 기술
<p>AWS 계정 ID를 기록합니다.</p>	<p>여러 계정에 대한 액세스를 설정하려면 AWS 계정 ID가 필요합니다.</p> <p>소스 및 대상 계정의 AWS 계정 ID를 기록해 둡니다. 자세한 내용은 IAM 설명서의 AWS 계정 ID 찾기를 참조하세요.</p>	<p>AWS DevOps</p>
<p>AWS CloudFormation 템플릿을 다운로드하십시오.</p>	<ol style="list-style-type: none"> 1. GitHub 리포지토리에서 이 패턴에 대한 sample_target_codebuild_template.yaml AWS CloudFormation 템플릿을 다운로드하십시오. 2. GitHub 리포지토리에서 이 패턴에 대한 codebuild_lambda_proxy_template.yaml AWS CloudFormation 템플릿을 다운로드하십시오. <p>참고: AWS CloudFormation 템플릿에서 <SourceAccountId> 는 원본 계정의 AWS 계정 ID이고, 대상 계정의 AWS 계정 <TargetAccountId> ID입니다.</p>	<p>AWS DevOps</p>
<p>AWS CloudFormation 스택을 생성하고 배포합니다.</p>	<ol style="list-style-type: none"> 1. 소스 계정의 AWS Management 콘솔에 로그인 	<p>AWS DevOps</p>

작업	설명	필요한 기술
	<p>하고 AWS CloudFormation 콘솔을 연 다음 [Stacks] 를 선택합니다.</p> <ol style="list-style-type: none"> 2. 스택 생성을 선택한 다음 새 리소스 사용(표준)을 선택합니다. 3. 템플릿 소스로 템플릿 파일 업로드를 선택합니다. 4. 템플릿 파일 업로드에서 파일을 선택한 다음 다운로드한 codebuild _lambda_proxy_template.yaml 파일을 선택합니다. 다음을 선택합니다. 5. 이름에 스택의 이름을 입력합니다(예: codebuild-lambda-proxy). 6. crossAccountTarget RoleArn 파라미터를 <TargetAccountId> 로 바꿉니다(예: <arn:aws:iam::123456789012:role/proxy-lambda-codebuild-role>). 참고: targetCodeBuildProject 파라미터의 기본값은 업데이트하지 않아도 됩니다. 7. 다음을 선택하고 기본 스택 생성 옵션을 적용한 다음 다음을 선택합니다. 8. AWS가 사용자 지정 이름으로 IAM 리소스를 생성할 	

작업	설명	필요한 기술
	<p>CloudFormation 수 있음을 인정함 확인란을 선택한 다음 Create stack (스택 생성)을 선택합니다.</p> <p>참고: 대상 계정에서 리소스를 생성하기 전에 프록시 Lambda 함수에 대한 AWS CloudFormation 스택을 생성해야 합니다. 대상 계정에서 신뢰 정책을 생성하면 IAM 역할이 역할 이름에서 내부 식별자로 변환됩니다. 따라서 IAM 역할이 이미 존재해야 합니다.</p>	
<p>프록시 함수 및 상태 머신의 생성을 확인합니다.</p>	<ol style="list-style-type: none"> 1. AWS CloudFormation 스택이 CREATE_COMPLETE 상태에 도달할 때까지 기다리십시오. 1분 내에 완료됩니다. 2. AWS Lambda 콘솔을 열고 함수를 선택한 후 lambda-proxy-ProxyLambda-<GUID> 함수를 찾습니다. 3. AWS Step Functions 콘솔을 열고 상태 머신을 선택한 다음 sample-crossaccount-codebuild-state-machine 상태 머신을 찾습니다. 	<p>AWS DevOps</p>

대상 계정에서 IAM 역할을 생성하고 샘플 CodeBuild 프로젝트를 시작합니다.

작업	설명	필요한 기술
AWS CloudFormation 스택을 생성하고 배포합니다.	<ol style="list-style-type: none"> 대상 계정의 AWS Management 콘솔에 로그인하고 AWS CloudFormation 콘솔을 연 다음 [Stacks] 를 선택합니다. 스택 생성을 선택한 다음 새 리소스 사용(표준)을 선택합니다. 템플릿 소스로 템플릿 파일 업로드를 선택합니다. 템플릿 파일 업로드에서 파일 선택을 선택한 다음 <code>sample_target_codebuild_template.yaml</code> 파일을 선택합니다. 다음을 선택합니다. 스택 이름에 스택의 이름을 입력합니다(예: <code>sample-codebuild-stack</code>). <code>crossAccountSourceRoleArn</code> 파라미터를 <code><SourceAccountId></code> 로 바꿉니다(예: <code><arn:aws:iam::123456789012:role/codebuild-proxy-lambda-role></code>). 다음을 선택하고 기본 스택 생성 옵션을 적용한 후 다음을 선택합니다. AWS가 사용자 지정 이름으로 IAM 리소스를 생성할 	AWS DevOps

작업	설명	필요한 기술
	CloudFormation 수 있음을 인정함 확인란을 선택한 다음 Create stack (스택 생성)을 선택합니다.	
샘플 CodeBuild 프로젝트 생성을 확인합니다.	<ol style="list-style-type: none"> 1. AWS CloudFormation 스택이 CREATE_COMPLETE 상태에 도달할 때까지 기다리십시오. 1분 내에 완료됩니다. 2. AWS CodeBuild 콘솔을 열고 sample-codebuild-project 프로젝트를 찾습니다. 	AWS DevOps

교차 계정 Lambda 프록시 함수 테스트

작업	설명	필요한 기술
상태 머신을 실행합니다.	<ol style="list-style-type: none"> 1. 소스 계정의 AWS Management Console에 로그인하고, AWS Step Functions 콘솔을 열고, 상태 머신을 선택합니다. 2. sample-crossaccount-codebuild-state-machine 상태 머신을 선택한 다음 실행 시작을 선택합니다. 3. 입력 편집기에서 다음 JSON을 입력하고 CodeBuild 프로젝트가 포함된 계정의 AWS 계정 <TargetAc 	AWS DevOps

작업	설명	필요한 기술
	<p>countID> ID로 대체합니다.</p> <pre data-bbox="630 327 1029 1205"> { "crossAccountTargetRoleArns": [{ "arn": "arn:aws:iam::<TargetAccountID>:role/proxy-lambda-codebuild-role", "region": "eu-west-1", "codeBuildProject": "sample-codebuild-project", "SampleValue1": "Value1", "SampleValue2": "Value2" }] } </pre> <p>참고: 키값 쌍은 소스 계정의 함수에서 대상 계정의 CodeBuild 프로젝트로 환경 변수로 전달됩니다.</p> <ol style="list-style-type: none"> 4. 실행 시작을 선택합니다. 5. 상태 시스템 페이지의 세부 정보 탭에서 실행 상태가 성공으로 설정되어 있는지 확인합니다. 이렇게 하면 상태 머신이 실행 중임을 확인할 수 있습니다. 참고: 상태 머신이 성공 상태에 도달하는 	

작업	설명	필요한 기술
	<p>데 약 30초가 걸릴 수 있습니다.</p> <p>6. 상태 머신에서 단계의 출력 및 입력을 보려면 실행 이벤트 기록 섹션에서 해당 단계를 확장합니다. 예를 들어, Lambda CodeBuild - 프록시 — 시작 단계를 확장하십시오. 출력에는 재정의된 환경 변수, 원본 페이로드 및 작업 ID에 대한 세부 정보가 포함됩니다. CodeBuild</p>	
환경 변수를 확인합니다.	<ol style="list-style-type: none"> 1. 대상 계정의 AWS Management Console에 로그인합니다. 2. AWS CodeBuild 콘솔을 열고 Build를 확장한 다음 Build projects를 선택합니다. 3. sample-codebuild-project 프로젝트를 선택한 다음 세부 정보 보기를 선택합니다. 4. 빌드 기록 탭에서 프로젝트의 최신 빌드를 선택한 다음 로그 보기를 선택합니다. 5. 로그 출력에서 STDOUT에 인쇄된 환경 변수가 Step Functions 샘플 상태 머신의 환경 변수와 일치하는지 확인합니다. 	AWS DevOps

문제 해결

문제	Solution
Step Functions 실행이 예상보다 오래 걸립니다.	Step Function 상태 머신에서 맵의 <code>MaxConcurrency</code> 프로퍼티를 조정하여 병렬로 실행할 수 있는 CodeBuild 프로젝트 수를 제어합니다.
CodeBuild 작업 실행이 예상보다 오래 걸립니다.	<ol style="list-style-type: none"> 1. Step Functions 상태 시스템에서 대기 시간 값을 조정하여 작업 상태에 대한 폴링 요청을 최소화합니다. CodeBuild 프로젝트의 예상 실행 시간을 사용하세요. 2. 사용할 도구가 CodeBuild 적절한지 생각해 보세요. 예를 들어 CodeBuild 작업을 초기화하는 데 필요한 시간은 AWS Lambda보다 훨씬 더 길 수 있습니다. 높은 처리량과 빠른 완료 시간이 필요한 경우 비즈니스 로직을 AWS Lambda로 마이그레이션하고 팬아웃 아키텍처를 사용하는 것을 고려해 보십시오.

AWS 코드 서비스 및 AWS KMS 다중 리전 키를 사용하여 여러 계정 및 리전에 대한 마이크로서비스의 블루/그린 배포를 관리

작성자: Balaji Vedagiri(AWS), Ashish Kumar(AWS), Faisal Shahdad(AWS), Anand Krishna Varanasi(AWS), Vanitha Dontireddy(AWS), Vivek Thangamuthu(AWS)

코드 리포지토리: [ecs-blue-green-global -- deployment-with-multiregion-cmk](#) 코드 파이프라인

환경: PoC 또는 파일럿

기술: DevOps; 컨테이너 및 마이크로서비스

AWS 서비스: AWS CloudFormation, AWS CodeBuild, AWS CodeDeploy, AWS CodePipeline, 아마존 ECS

요약

이 패턴은 블루/그린 배포 전략에 따라 중앙 AWS 계정에서 여러 워크로드 계정 및 리전으로 글로벌 마이크로서비스 애플리케이션을 배포하는 방법을 설명합니다. 이 패턴은 다음을 지원합니다.

- 소프트웨어는 중앙 계정에서 개발되는 반면, 워크로드와 애플리케이션은 여러 계정과 AWS 리전에 분산되어 있습니다.
- 단일 AWS Key Management System(AWS KMS) 다중 리전 키는 재해 복구를 위한 암호화 및 복호화에 사용됩니다.
- KMS 키는 리전별로 다르며 파이프라인 아티팩트를 위해 세 개의 다른 리전에서 유지 관리하거나 생성해야 합니다. KMS 다중 리전 키는 여러 리전에서 동일한 키 ID를 유지하는 데 도움이 됩니다.
- Git 워크플로 브랜칭 모델은 두 개의 브랜치(개발 및 메인)로 구현되며 풀 리퀘스트(PR)를 사용하여 코드를 병합합니다. 이 스택에서 배포되는 AWS Lambda 함수는 개발 브랜치에서 메인 브랜치로 PR을 생성합니다. 메인 브랜치에 대한 PR 병합은 지속적 통합 및 지속적 전달(CI/CD) 흐름을 오케스트레이션하고 계정 전체에 스택을 배포하는 AWS CodePipeline 파이프라인을 시작합니다.

이 패턴은 AWS CloudFormation 스택을 통해 코드 (IaC) 로 설정된 샘플 인프라를 제공하여 이러한 사용 사례를 보여줍니다. 마이크로서비스의 블루/그린 배포는 AWS를 사용하여 구현됩니다. CodeDeploy

사전 조건 및 제한 사항

사전 조건

- 활성 AWS 계정 4개:
 - 코드 파이프라인을 관리하고 AWS CodeCommit 리포지토리를 유지하기 위한 도구 계정입니다.
 - 마이크로서비스 워크로드를 배포하기 위한 세 가지 워크로드(테스트) 계정.
- 이 패턴은 다음 리전을 사용합니다. 다른 지역을 사용하려면 AWS CodeDeploy 및 AWS KMS 다중 지역 스택을 적절하게 수정해야 합니다.
 - 도구 (AWS CodeCommit) 계정: ap-south-1
 - 워크로드(테스트) 계정 1: ap-south-1
 - 워크로드(테스트) 계정 2: eu-central-1
 - 워크로드(테스트) 계정 3: us-east-1
- 각 워크로드 계정의 배포 리전을 위한 Amazon Simple Storage Service(S3) 버킷 3개. (나중에 이 패턴에서 S3BUCKETNAMETESTACCOUNT1, S3BUCKETNAMETESTACCOUNT2 , S3BUCKETNAMETESTACCOUNT3 으로 불립니다.)

예를 들어 다음과 같이 고유한 버킷 이름을 사용하여 특정 계정 및 리전에 이러한 버킷을 생성할 수 있습니다(xxxx를 임의의 숫자로 대체).

```
##In Test Account 1
aws s3 mb s3://ecs-codepipeline-xxxx-ap-south-1 --region ap-south-1
##In Test Account 2
aws s3 mb s3://ecs-codepipeline-xxxx-eu-central-1 --region eu-central-1
##In Test Account 3
aws s3 mb s3://ecs-codepipeline-xxxx-us-east-1 --region us-east-1

#Example
##In Test Account 1
aws s3 mb s3://ecs-codepipeline-18903-ap-south-1 --region ap-south-1
##In Test Account 2
aws s3 mb s3://ecs-codepipeline-18903-eu-central-1 --region eu-central-1
##In Test Account 3
aws s3 mb s3://ecs-codepipeline-18903-us-east-1 --region us-east-1
```

제한 사항

패턴은 AWS CodeBuild 및 기타 구성 파일을 사용하여 샘플 마이크로서비스를 배포합니다. 여러 워크로드 유형(예: 서버리스)이 있는 경우 모든 관련 구성을 업데이트해야 합니다.

아키텍처

대상 기술 스택

- AWS CloudFormation
- AWS CodeCommit
- AWS CodeBuild
- AWS CodeDeploy
- AWS CodePipeline

대상 아키텍처

자동화 및 규모 조정

설정은 AWS CloudFormation 스택 템플릿 (IaC) 을 사용하여 자동화됩니다. 여러 환경 및 계정에 맞게 쉽게 확장할 수 있습니다.

도구

서비스

- [AWS는 AWS CloudFormation](#) 리소스를 설정하고, 빠르고 일관되게 프로비저닝하고, 전체 AWS 계정 및 지역의 수명 주기 전반에 걸쳐 리소스를 관리할 수 있도록 지원합니다.
- [CodeBuildAWS는](#) 소스 코드를 컴파일하고, 단위 테스트를 실행하고, 배포할 준비가 된 아티팩트를 생성하는 데 도움이 되는 완전관리형 빌드 서비스입니다.
- [CodeCommitAWS는](#) 자체 소스 제어 시스템을 관리할 필요 없이 Git 리포지토리를 비공개로 저장하고 관리할 수 있는 버전 제어 서비스입니다.
- [AWS는](#) 아마존 엘라스틱 CodeDeploy 컴퓨터 클라우드 (Amazon EC2) 또는 온프레미스 인스턴스, AWS Lambda 함수 또는 아마존 Elastic Container Service (아마존 ECS) 서비스로의 배포를 자동화합니다.

- [AWS](#)는 소프트웨어 릴리스의 여러 단계를 신속하게 모델링 및 구성하고 소프트웨어 변경 사항을 지속적으로 릴리스하는 CodePipeline 데 필요한 단계를 자동화하도록 지원합니다.
- [Amazon Elastic Container Registry\(Amazon ECR\)](#)는 안전하고 확장 가능하고 신뢰할 수 있는 관리형 컨테이너 이미지 레지스트리 서비스입니다.
- [Amazon Elastic Container Service\(Amazon ECS\)](#)는 클러스터에서 컨테이너를 실행, 중지 및 관리하는 데 도움이 되는 빠르고 확장 가능한 컨테이너 관리 서비스입니다.
- [AWS Key Management Service\(AWS KMS\)](#)를 사용하면 암호화 키를 생성하고 제어하여 데이터를 보호할 수 있습니다.
- [Amazon Simple Storage Service\(S3\)](#)는 원하는 양의 데이터를 저장, 보호 및 검색하는 데 도움이 되는 클라우드 기반 객체 스토리지 서비스입니다.

추가 도구

- [Git](#)은 AWS 리포지토리와 함께 작동하는 오픈 소스 분산 버전 제어 시스템입니다. CodeCommit
- [Docker](#)는 운영 체제 수준의 가상화를 사용하여 컨테이너에 소프트웨어를 제공하는 서비스형 플랫폼(PaaS) 제품 세트입니다. 이 패턴은 Docker를 사용하여 로컬에서 컨테이너 이미지를 구축하고 테스트합니다.
- [cfn-lint](#) 및 [cfn-nag](#)는 스택에 오류 및 보안 문제가 있는지 검토하는 데 도움이 되는 오픈 소스 도구입니다. CloudFormation

코드 리포지토리

[이 패턴의 코드는 여러 지역 및 계정 저장소의 글로벌 블루/그린 배포에서 사용할 수 있습니다. GitHub](#)

에픽

환경 변수 설정

작업	설명	필요한 기술
스택 배포를 위한 환경 변수를 내보냅니다. CloudFormation	이 패턴에서 나중에 CloudFormation 스택에 입력으로 사용할 환경 변수를 정의합니다.	AWS DevOps

작업	설명	필요한 기술
	<p>1. 사전 요구 사항 섹션의 앞부분에 설명된 대로 세 개의 계정 및 리전에서 생성한 버킷 이름을 업데이트합니다.</p> <pre data-bbox="630 424 1029 823">export S3BUCKETN AMETESTACCOUNT1=<S 3BUCKETACCOUNT1> export S3BUCKETN AMETESTACCOUNT2=<S 3BUCKETACCOUNT2> export S3BUCKETN AMETESTACCOUNT3=<S 3BUCKETACCOUNT3></pre> <p>2. 버킷 이름은 전 세계적으로 고유해야 하므로 임의의 문자열을 정의하여 아티팩트 버킷을 생성하세요.</p> <pre data-bbox="630 1054 1029 1255">export BUCKETSTA RTNAME=ecs-codepip eline-artifacts-19 992</pre> <p>3. 계정 ID와 리전 정의 및 내보내기:</p> <pre data-bbox="630 1390 1029 1839">export TOOLSACCO UNT=<TOOLSACCOUNT> export CODECOMMI TACCOUNT=<CODECOMM ITACCOUNT> export CODECOMMI TREGION=ap-south-1 export CODECOMMI TREPONAME=Poc export TESTACCOU NT1=<TESTACCOUNT1></pre>	

작업	설명	필요한 기술
	<pre>export TESTACCOUNT2=<TESTACCOUNT2> export TESTACCOUNT3=<TESTACCOUNT3> export TESTACCOUNT1REGION=ap-south-1 export TESTACCOUNT2REGION=eu-central-1 export TESTACCOUNT3REGION=us-east-1 export TOOLSACCOUNTREGION=ap-south-1 export ECRREPOSITORYNAME=web</pre>	

인프라용 CloudFormation 스택을 패키징하고 배포합니다.

작업	설명	필요한 기술
리포지토리를 복제합니다.	<p>샘플 리포지토리를 작업 위치의 새 리포지토리로 복제하세요.</p> <pre>##In work location git clone https://github.com/aws-samples/ecs-blue-green-global-deployment-with-multiregion-cmk-codepipeline.git</pre>	AWS DevOps
CloudFormation 리소스를 패키징하세요.	이 단계에서는 CloudFormation 템플릿이 참조하는 로컬 아티팩트를 패키징하여 Amazon Virtual Private Cloud (Amazon	AWS DevOps

작업	설명	필요한 기술
	<p>VPC) 및 Application Load Balancer와 같은 서비스에 필요한 인프라 리소스를 생성합니다.</p> <p>템플릿은 코드 리포지토리의 Infra 폴더에서 사용할 수 있습니다.</p> <pre data-bbox="609 604 1029 1276">##In TestAccount1## aws cloudformation package \ --template-file mainInfraStack.yaml \ --s3-bucket \$S3BUCKETNAMETESTA CCOUNT1 \ --s3-prefix infraStack \ --region \$TESTACCO UNT1REGION \ --output-template- file infrastructure_ \${TESTACCOUNT1}.templ ate</pre> <pre data-bbox="609 1310 1029 1877">##In TestAccount2## aws cloudformation package \ --template-file mainInfraStack.yaml \ --s3-bucket \$S3BUCKETNAMETESTA CCOUNT2 \ --s3-prefix infraStack \ --region \$TESTACCO UNT2REGION \ --output-template- file infrastructure_</pre>	

작업	설명	필요한 기술
	<pre> \${TESTACCOUNT2}.template </pre> <pre> ##In TestAccount3## aws cloudformation package \ --template-file mainInfraStack.yaml \ --s3-bucket \$S3BUCKETNAMETESTA CCOUNT3 \ --s3-prefix infraStack \ --region \$TESTACCO UNT3REGION \ --output-template- file infrastructure_ \${TESTACCOUNT3}.template </pre>	

작업	설명	필요한 기술
패키지 템플릿을 검증하세요.	<p>패키지 템플릿 검증:</p> <pre>aws cloudformation validate-template \ --template-body file://infrastruct ure_\${TESTACCOUNT1 }.template aws cloudformation validate-template \ --template-body file://infrastruct ure_\${TESTACCOUNT2 }.template aws cloudformation validate-template \ --template-body file://infrastruct ure_\${TESTACCOUNT3 }.template</pre>	AWS DevOps

작업	설명	필요한 기술
패키지 파일을 워크로드 계정에 배포합니다.	<ol style="list-style-type: none"> 1. 설정에 따라 <code>nfraParameters.json</code> 스크립트의 자리 표시자 값과 계정 이름을 업데이트합니다. 2. 패키지 템플릿을 세 개의 워크로드 계정에 배포하세요. <pre data-bbox="634 548 1029 1873"> ##In TestAccount1## aws cloudformation deploy \ --template-file infrastructure_\${T ESTACCOUNT1}.templ ate \ --stack-name mainInfrastack \ --parameter- overrides file://in fraParameters.json \ --region \$TESTACCO UNT1REGION \ --capabilities CAPABILITY_IAM CAPABILITY_NAMED_I AM ##In TestAccount2## aws cloudformation deploy \ --template-file infrastructure_\${T ESTACCOUNT2}.templ ate \ --stack-name mainInfrastack \ --parameter- overrides file://in fraParameters.json \ --region \$TESTACCO UNT2REGION \ </pre>	AWS DevOps

작업	설명	필요한 기술
	<pre> --capabilities CAPABILITY_IAM CAPABILITY_NAMED_I AM ##In TestAccount3## aws cloudformation deploy \ --template-file infrastructure_\${T ESTACCOUNT3}.templ ate \ --stack-name mainInfrastack \ --parameter- overrides file://in fraParameters.json \ --region \$TESTACCO UNT3REGION \ --capabilities CAPABILITY_IAM CAPABILITY_NAMED_I AM </pre>	

샘플 이미지를 푸시하고 Amazon ECS를 확장

작업	설명	필요한 기술
<p>Amazon ECR 리포지토리에 샘플 이미지를 푸시합니다.</p>	<p>샘플(NGINX) 이미지를 web이라는 Amazon Elastic Container Registry(Amazon ECR) 리포지토리에 (파라미터에 설정된 대로)푸시합니다. 필요에 따라 이미지를 사용자 지정할 수 있습니다.</p> <p>Amazon ECR에 이미지를 푸시하기 위해 로그인하고 보안 인</p>	<p>AWS DevOps</p>

작업	설명	필요한 기술
	<p>중 정보를 설정하려면 Amazon ECR 설명서의 지침을 따르세요.</p> <p>명령은 다음과 같습니다.</p> <pre>docker pull nginx docker images docker tag <imageid> aws_account_id.dkr .ecr.region.amazon aws.com/<web>:latest docker push <aws_account_id>.dkr.ecr.<region>.amazonaws.com/ <web>:tag</pre>	
<p>Amazon ECS를 확장하고 액세스를 확인합니다.</p>	<ol style="list-style-type: none"> 1. Amazon ECS를 확장하여 두 개의 복제본을 생성합니다. <pre>aws ecs update-service --cluster QA-Cluster --service Poc-Service --desired-count 2</pre> <p>여기서 Poc-Service 는 사용자의 샘플 애플리케이션을 의미합니다.</p> 2. 브라우저의 정규화된 도메인 이름(FQDN) 또는 DNS를 사용하거나 curl 명령을 사용하여 Application Load Balancer에서 서비스에 액세스할 수 있는지 확인합니다. 	<p>AWS DevOps</p>

코드 서비스 및 리소스 설정

작업	설명	필요한 기술
<p>도구 계정에 CodeCommit 리포지토리를 생성합니다.</p>	<p>CodeCommit 리포지토리 code 폴더에 있는 codecommit.yaml 템플릿을 사용하여 도구 계정에 GitHub 리포지토리를 생성합니다. 코드를 개발하려는 리전 하나에만 이 리포지토리를 생성해야 합니다.</p> <pre data-bbox="597 688 1026 1247"> aws cloudformation deploy --stack-name codecommitrepoStack --parameter-overrides CodeCommitReponame= \$CODECOMMITREPONAME \ ToolsAccount=\$TO OLSACCOUNT --templat e-file codecommit.yaml --region \$TOOLSACC OUNTREGION \ --capabilities CAPABILITY_NAMED_IAM </pre>	<p>AWS DevOps</p>
<p>에서 생성한 CodePipeline 아티팩트를 관리하기 위한 S3 버킷을 생성합니다.</p>	<p>리포지토리 code 폴더에 있는 pre-reqs-bucket.yaml 템플릿을 사용하여 CodePipeline 생성된 아티팩트를 관리하기 위한 S3 버킷을 생성합니다. GitHub 스택은 세 개의 워크로드(테스트) 및 도구 계정과 리전 모두에 배포되어야 합니다.</p> <pre data-bbox="597 1696 1026 1864"> aws cloudformation deploy --stack-name pre-reqs-artifacts -bucket --parameter- </pre>	<p>AWS DevOps</p>

작업	설명	필요한 기술
	<pre> overrides BucketStartName=\$BUCKETSTARTNAME \ TestAccount1=\$TESTACCOUNT1 TestAccount2=\$TESTACCOUNT2 \ TestAccount3=\$TESTACCOUNT3 CodeCommitAccount=\$CODECOMMITACCOUNT ToolsAccount=\$TOOLSACCOUNT \ --template-file pre-reqs_bucket.yaml --region \$TESTACCOUNT1REGION --capabilities CAPABILITY_NAMED_IAM aws cloudformation deploy --stack-name pre-reqs-artifacts -bucket --parameter- overrides BucketStartName=\$BUCKETSTARTNAME \ TestAccount1=\$TESTACCOUNT1 TestAccount2=\$TESTACCOUNT2 \ TestAccount3=\$TESTACCOUNT3 CodeCommitAccount=\$CODECOMMITACCOUNT ToolsAccount=\$TOOLSACCOUNT \ --template-file pre-reqs_bucket.yaml --region \$TESTACCOUNT2REGION --capabilities CAPABILITY_NAMED_IAM aws cloudformation deploy --stack-name </pre>	

작업	설명	필요한 기술
	<pre> pre-reqs-artifacts -bucket --parameter- overrides BucketSta rtName=\$BUCKETSTAR TNAME \ TestAccount1=\$TE STACCOUNT1 TestAccou nt2=\$TESTACCOUNT2 \ TestAccount3=\$TE STACCOUNT3 CodeComm itAccount=\$CODECOMM ITACCOUNT ToolsAcco unt=\$TOOLSACCOUNT \ --template-file pre- reqs_bucket.yaml --region \$TESTACCO UNT3REGION --capabil ities CAPABILIT Y_NAMED_IAM aws cloudformation deploy --stack-name pre-reqs-artifacts -bucket --parameter- overrides BucketSta rtName=\$BUCKETSTAR TNAME \ TestAccount1=\$TE STACCOUNT1 TestAccou nt2=\$TESTACCOUNT2 \ TestAccount3=\$TE STACCOUNT3 CodeComm itAccount=\$CODECOMM ITACCOUNT ToolsAcco unt=\$TOOLSACCOUNT \ --template-file pre- reqs_bucket.yaml --region \$TOOLSACC OUNTREGION --capabil ities CAPABILIT Y_NAMED_IAM </pre>	

작업	설명	필요한 기술
멀티 리전 KMS 키를 설정합니다.	<p>1. 사용할 기본 키와 복제 키로 다중 지역 KMS 키를 생성합니다. CodePipeline 이 예시에서는 ToolsAccount1region - ap-south-1 이 기본 리전이 될 것입니다.</p> <pre data-bbox="630 583 1029 1339">aws cloudformation deploy --stack-name ecs-codepipeline-p re-reqs-KMS \ --template-file pre- reqs_KMS.yaml -- parameter-overrides \ TestAccount1=\$TE STACCOUNT1 TestAccou nt2=\$TESTACCOUNT2 \ TestAccount3=\$TE STACCOUNT3 CodeComm itAccount=\$CODECOMM ITACCOUNT ToolsAcco unt=\$TOOLSACCOUNT --region \$TOOLSACC OUNTREGION</pre> <p>2. 프로젝트에 전달할 CMKARN 변수를 설정합니다. CodeBuild 값은 ecs-codepipeline-pre-reqs -KMS 템플릿 스택의 출력에서 사용할 수 있습니다 (키 ID는 모든 지역에서 동일하고 로 시작됨). mrk- 또는 도구 계정에서 CMKARN 값을 가져올 수 있습니다. 모든 계정 세션에서 내보내기:</p>	AWS DevOps

작업	설명	필요한 기술
	<pre> export CMKARN1=arn:aws:kms:ap-south-1:<TOOLSACCOUNTID>:key/mrk-xxx export CMKARN2=arn:aws:kms:eu-central-1:<TOOLSACCOUNTID>:key/mrk-xxx export CMKARN3=arn:aws:kms:us-east-1:<TOOLSACCOUNTID>:key/mrk-xxx export CMARNTOOLS=arn:aws:kms:ap-south-1:<TOOLSACCOUNTID>:key/mrk-xxx </pre>	

작업	설명	필요한 기술
<p>도구 계정에서 CodeBuild 프로젝트를 설정합니다.</p>	<ol style="list-style-type: none"> 1. GitHub 리포지토리 code 폴더의 codebuild_IAM.yaml 템플릿을 사용하여 도구 계정의 단일 리전에서 AWS용 AWS CodeBuild ID 및 액세스 관리 (IAM) 를 설정합니다. <pre data-bbox="634 590 1029 1062">#In ToolsAccount aws cloudformation deploy --stack-name ecs-codebuild-iam \ --template-file codebuild_IAM.yaml --region \$TOOLSACCOUNTREGION \ --capabilities CAPABILITY_NAMED_IAM</pre> <ol style="list-style-type: none"> 2. codebuild.yaml 템플릿을 사용하여 빌드 프로젝트를 CodeBuild 설정하십시오. 이 템플릿을 다음과 같이 세 리전 모두에 배포하세요. <pre data-bbox="634 1346 1029 1831">aws cloudformation deploy --stack-name ecscodebuildstack -- parameter-overrides ToolsAccount=\$TOOLSACCOUNT \ CodeCommitRepoName= \$CODECOMMITREPONAME ECRRepositoryName= \$ECRREPOSITORYNAME APPACCOUNTID=\$TESTACCOUNT1 \</pre>	<p>AWS DevOps</p>

작업	설명	필요한 기술
	<pre> TestAccount3=\$TE STACCOUNT3 CodeCommi tRegion=\$CODECOMMI TREGION CMKARN=\$C MKARN1 \ --template-file codebuild.yaml --region \$TESTACCO UNT1REGION --capabil ities CAPABILIT Y_NAMED_IAM aws cloudformation deploy --stack-name ecscodebuildstack -- parameter-overrides ToolsAccount=\$TOOL SACCOUNT \ CodeCommitRepoName= \$CODECOMMITREPONAME ECRRepositoryName= \$ECRREPOSITORYNAME APPACCOUNTID=\$TEST ACCOUNT2 \ TestAccount3=\$TE STACCOUNT3 CodeCommi tRegion=\$CODECOMMI TREGION CMKARN=\$C MKARN2 \ --template-file codebuild.yaml --region \$TESTACCO UNT2REGION --capabil ities CAPABILIT Y_NAMED_IAM aws cloudformation deploy --stack-name ecscodebuildstack -- parameter-overrides ToolsAccount=\$TOOL SACCOUNT \ </pre>	

작업	설명	필요한 기술
	<pre>CodeCommitRepoName= \$CODECOMMITREPONAME ECRRepositoryName= \$ECRREPOSITORYNAME APPACCOUNTID=\$TEST ACCOUNT3 \ CodeCommitRegion= \$CODECOMMITREGION CMKARN=\$CMKARN3 \ --template-file codebuild.yaml --region \$TESTACCO UNT3REGION --capabil ities CAPABILIT Y_NAMED_IAM</pre>	

작업	설명	필요한 기술
<p>워크로드 CodeDeploy 계정에 설정.</p>	<p>GitHub 리포지토리 code 폴더에 있는 <code>codedeploy.yaml</code> 템플릿을 사용하여 세 개의 워크로드 계정 모두에 설정할 CodeDeploy 수 있습니다. <code>mainInfraStack</code> 의 출력에는 Amazon ECS 클러스터의 Amazon 리소스 이름(ARN)과 Application Load Balancer 리스너가 포함됩니다.</p> <p>참고: 인프라 스택의 값은 이미 보내냈으므로 CodeDeploy 스택 템플릿에서 가져옵니다.</p> <pre data-bbox="592 903 1031 1831"> ###WorkloadAccount1## aws cloudformation deploy --stack-name ecscodedeploystack \ --parameter-overrides ToolsAccount=\$TOOL SACCOUNT mainInfra stackname=mainInfr astack \ --template-file codedeploy.yaml --region \$TESTACCO UNT1REGION --capabil ities CAPABILIT Y_NAMED_IAM ###WorkloadAccount2## aws cloudformation deploy --stack-name ecscodedeploystack \ --parameter-overrides ToolsAccount=\$TOOL SACCOUNT mainInfra </pre>	<p>AWS DevOps</p>

작업	설명	필요한 기술
	<pre> stackname=mainInfr astack \ --template-file codedeploy.yaml --region \$TESTACCO UNT2REGION --capabil ities CAPABILIT Y_NAMED_IAM ##WorkloadAccount3## aws cloudformation deploy --stack-name ecscodedeploystack \ --parameter-overrides ToolsAccount=\$TOOL SACCOUNT mainInfra stackname=mainInfr astack \ --template-file codedeploy.yaml --region \$TESTACCO UNT3REGION --capabil ities CAPABILIT Y_NAMED_IAM </pre>	

도구 CodePipeline 계정에서 설정

작업	설명	필요한 기술
<p>도구 계정에 코드 파이프라인을 생성하세요.</p>	<p>도구 계정에서 다음 명령어를 실행합니다.</p> <pre> aws cloudformation deploy --stack-name ecscodepipelinestack --parameter-overrides \ TestAccount1=\$TE STACCOUNT1 TestAccou </pre>	<p>AWS DevOps</p>

작업	설명	필요한 기술
	<pre> nt1Region=\$TESTACC OUNT1REGION \ TestAccount2=\$TE STACCOUNT2 TestAccou nt2Region=\$TESTACC OUNT2REGION \ TestAccount3=\$TE STACCOUNT3 TestAccou nt3Region=\$TESTACC OUNT3REGION \ CMKARNTools=\$CMK TROOLSARN CMKARN1= \$CMKARN1 CMKARN2=\$ CMKARN2 CMKARN3=\$ CMKARN3 \ CodeCommitRepoName= \$CODECOMMITREPONAME BucketStartName=\$B UCKETSTARTNAME \ --template-file codepipeline.yaml -- capabilities CAPABILIT Y_NAMED_IAM </pre>	

작업	설명	필요한 기술
<p>AWS KMS 키 정책 CodePipeline 및 S3 버킷 정책에 대한 액세스 및 CodeBuild 역할을 제공합니다.</p>	<p>1. AWS KMS 키 정책에 대한 액세스 권한 CodePipeline 및 CodeBuild 역할 제공:</p> <pre data-bbox="634 401 1029 1230">aws cloudformation deploy --stack-name ecs-codepipeline-p re-reqs-KMS \ --template-file pre- reqs_KMS.yaml -- parameter-overrides \ CodeBuildCondi tion=true TestAccou nt1=\$TESTACCOUNT1 TestAccount2=\$TEST ACCOUNT2 \ TestAccount3=\$TE STACCOUNT3 CodeComm itAccount=\$CODECOMM ITACCOUNT ToolsAcco unt=\$TOOLSACCOUNT --region \$TOOLSACC OUNTREGION</pre> <p>2. 다음과 같은 CodeDeploy 역할에 대한 액세스를 허용하도록 S3 버킷 정책을 CodePipeline 업데이트하십시오.</p> <pre data-bbox="634 1507 1029 1877">aws cloudformation deploy --stack-name pre-reqs-artifacts -bucket --parameter- overrides BucketSta rtName=\$BUCKETSTAR TNAME \ PutS3BucketPolic y=true TestAccou</pre>	<p>AWS DevOps</p>

작업	설명	필요한 기술
	<pre> nt1=\$TESTACCOUNT1 TestAccount2=\$TEST ACCOUNT2 \ TestAccount3=\$TE STACCOUNT3 CodeCommi tAccount=\$CODECOMM ITACCOUNT ToolsAcco unt=\$TOOLSACCOUNT \ --template-file pre- reqs_bucket.yaml --region \$TESTACCO UNT1REGION --capabil ities CAPABILIT Y_NAMED_IAM aws cloudformation deploy --stack-name pre-reqs-artifacts -bucket --parameter- overrides BucketSta rtName=\$BUCKETSTAR TNAME \ PutS3BucketPolic y=true TestAccou nt1=\$TESTACCOUNT1 TestAccount2=\$TEST ACCOUNT2 \ TestAccount3=\$TE STACCOUNT3 CodeCommi tAccount=\$CODECOMM ITACCOUNT ToolsAcco unt=\$TOOLSACCOUNT \ --template-file pre- reqs_bucket.yaml --region \$TESTACCO UNT2REGION --capabil ities CAPABILIT Y_NAMED_IAM aws cloudformation deploy --stack-name pre-reqs-artifacts </pre>	

작업	설명	필요한 기술
	<pre> -bucket --parameter- overrides BucketSta rtName=\$BUCKETSTAR TNAME \ PutS3BucketPolic y=true TestAccou nt1=\$TESTACCOUNT1 TestAccount2=\$TEST ACCOUNT2 \ TestAccount3=\$TE STACCOUNT3 CodeComm tAccount=\$CODECOMM ITACCOUNT ToolsAcco unt=\$TOOLSACCOUNT \ --template-file pre- reqs_bucket.yaml --region \$TESTACCO UNT3REGION --capabil ities CAPABILIT Y_NAMED_IAM aws cloudformation deploy --stack-name pre-reqs-artifacts -bucket --parameter- overrides BucketSta rtName=\$BUCKETSTAR TNAME \ PutS3BucketPolic y=true TestAccou nt1=\$TESTACCOUNT1 TestAccount2=\$TEST ACCOUNT2 \ TestAccount3=\$TE STACCOUNT3 CodeComm tAccount=\$CODECOMM ITACCOUNT ToolsAcco unt=\$TOOLSACCOUNT \ --template-file pre- reqs_bucket.yaml --region \$TOOLSACC OUNTREGION --capabil </pre>	

작업	설명	필요한 기술
	ities CAPABILIT Y_NAMED_IAM	

파이프라인 직접 호출 및 테스트

작업	설명	필요한 기술
CodeCommit 리포지토리에 변경 사항을 푸시합니다.	<ol style="list-style-type: none"> 1. AWS CodeCommit 설명서에 설명된 대로 <code>git clone</code> 명령을 사용하여 에서 만든 CodeCommit 리포지토리를 복제합니다. <code>codecommitrepoStack</code> 2. 입력 아티팩트에 필수 세부 정보를 업데이트하세요. <ul style="list-style-type: none"> • JSON 파일: 이 파일의 세 위치에서 파일의 AccountID 를 업데이트합니다. 계정 ID를 포함하도록 세 파일의 이름을 변경합니다. • YAML 파일: 작업 정의 ARN 및 버전을 업데이트합니다. 계정 ID를 포함하도록 세 파일의 이름을 변경합니다. 3. 홈 페이지를 약간 변경하도록 <code>index.html</code> 파일을 수정하세요. 4. 다음 파일을 리포지토리에 복사하고 커밋합니다. <div style="border: 1px solid #ccc; border-radius: 10px; padding: 5px; margin-top: 10px; width: fit-content;">index.html</div>	

작업	설명	필요한 기술
	<pre>Dockerfile buildspec.yaml appspec_<accountid>.yaml (3 files - one per account) taskdef<accountid>.json (3 files - one per account)</pre> <p>5. 파이프라인을 시작하거나 다시 시작하고 결과를 확인합니다.</p> <p>6. FQDN 또는 DNS를 사용하여 Application Load Balancer에서 서비스에 액세스하고 업데이트가 배포되었는지 확인합니다.</p>	

정리

작업	설명	필요한 기술
<p>배포된 모든 리소스를 정리합니다.</p>	<p>1. Amazon ECS를 0개의 인스턴스로 스케일 다운하세요.</p> <pre>aws ecs update-service --cluster QA-Cluster --service Poc-Service --desired-count 0</pre> <p>2. 각 계정 및 지역의 CloudFormation 스택을 삭제합니다.</p> <pre>##In Tools Account##</pre>	

작업	설명	필요한 기술
	<pre> aws cloudformation delete-stack -- stack-name ecscodepi pelinestack --region \$TOOLSACCOUNTREGION aws cloudformation delete-stack -- stack-name ecscodebu ildstack --region \$TESTACCOUNT1REGION aws cloudformation delete-stack -- stack-name ecscodebu ildstack --region \$TESTACCOUNT2REGION aws cloudformation delete-stack -- stack-name ecscodebu ildstack --region \$TESTACCOUNT3REGION aws cloudformation delete-stack -- stack-name ecs-codep ipeline-pre-reqs-K MS --region \$TOOLSACC OUNTREGION aws cloudformation delete-stack -- stack-name codecommi trepoStack --region \$TOOLSACCOUNTREGION aws cloudformation delete-stack -- stack-name pre-reqs- artifacts-bucket --region \$TESTACCO UNT1REGION aws cloudformation delete-stack -- stack-name pre-reqs- artifacts-bucket </pre>	

작업	설명	필요한 기술
	<pre> --region \$TESTACCO UNT2REGION aws cloudformation delete-stack -- stack-name pre-reqs- artifacts-bucket --region \$TESTACCO UNT3REGION aws cloudformation delete-stack -- stack-name pre-reqs- artifacts-bucket --region \$TOOLSACC OUNTREGION aws cloudformation delete-stack -- stack-name ecs-codeb uild-iam --region \$TOOLSACCOUNTREGION ##NOTE: Artifact buckets will not get deleted if there are artifacts so it has to be emptied manually before deleting.## ##In Workload / Test Accounts## ##Account:1## aws cloudformation delete-stack -- stack-name ecscodede ploystack --region \$TESTACCOUNT1REGION aws cloudformation delete-stack -- stack-name mainInfra stack --region \$TESTACCOUNT1REGION </pre>	

작업	설명	필요한 기술
	<pre>##Account:2## aws cloudformation delete-stack -- stack-name ecscodede ploystack --region \$TESTACCOUNT2REGION aws cloudformation delete-stack -- stack-name mainInfra stack --region \$TESTACCOUNT2REGION ##Account:3## aws cloudformation delete-stack -- stack-name ecscodede ploystack --region \$TESTACCOUNT3REGION aws cloudformation delete-stack -- stack-name mainInfra stack --region \$TESTACCOUNT3REGION ##NOTE: Amazon ECR (web) will not get deleted if the registry still includes images. It can be manually cleaned up if not required.</pre>	

문제 해결

문제	Solution
리포지토리에 커밋한 변경 사항은 배포되지 않습니다.	<ul style="list-style-type: none"> Docker 빌드 작업의 CodeBuild 로그에서 오류를 확인하세요. 자세한 내용은 CodeBuild 설 명서를 참조하십시오.

문제	Solution
	<ul style="list-style-type: none">• CodeDeploy 배포에서 Amazon ECS 배포 문제가 있는지 확인하십시오.

관련 리소스

- [도커 이미지 푸시](#)(Amazon ECR 설명서)
- [AWS CodeCommit 리포지토리에 연결](#) (AWS CodeCommit 설명서)
- [AWS 문제 해결 CodeBuild](#) (AWS CodeBuild 설명서)

AWS 및 AWS Config를 사용하여 Amazon ECR 리포지토리에서 와일드카드 권한을 모니터링합니다. CloudFormation

작성자: Vikrant Telkar(AWS), Sajid Momin(AWS), 및 Wassim Benhallam(AWS)

환경: 프로덕션	기술: DevOps; 컨테이너 및 마이크로서비스	AWS 서비스: AWS CloudFormation, AWS Config, 아마존 ECR, 아마존 SNS, AWS Lambda
----------	----------------------------	---

요약

Amazon Web Services(AWS) Cloud에서 Amazon Elastic Container Registry(Amazon ECR)는 AWS Identity 및 Access Management(IAM)를 사용하여 리소스 기반 권한이 있는 프라이빗 리포지토리를 지원하는 관리형 컨테이너 이미지 레지스트리 서비스입니다.

IAM은 리소스 및 작업 속성 모두에서 "*" 와일드카드를 지원하므로 이로 인해 일치하는 여러 항목을 자동으로 더 쉽게 선택할 수 있습니다. 테스트 환경에서는 인증된 모든 AWS 사용자가 `ecr:*` [와일드카드 권한](#)을 사용하여 [리포지토리 정책 설명](#)의 주요 요소에서 Amazon ECR 리포지토리에 액세스하도록 허용할 수 있습니다. `ecr:*` 와일드카드 권한은 프로덕션 데이터에 액세스할 수 없는 개발 계정에서 개발하고 테스트할 때 유용할 수 있습니다.

하지만 `ecr:*` 와일드카드 권한은 심각한 보안 취약성을 유발할 수 있으므로 프로덕션 환경에서 사용하지 않도록 해야 합니다. 이 패턴의 접근 방식을 사용하면 리포지토리 정책 설명에 `ecr:*` 와일드카드 권한이 포함된 Amazon ECR 리포지토리를 식별하는 데 도움을 줍니다. 패턴은 AWS Config에서 사용자 지정 규칙을 생성하기 위한 단계와 AWS CloudFormation 템플릿을 제공합니다. 그러면 AWS Lambda 함수가 `ecr:*` 와일드카드 권한에 대한 Amazon ECR 리포지토리 정책 설명을 모니터링합니다. 규정을 준수하지 않는 리포지토리 정책 설명을 발견하면 Lambda는 Amazon에 이벤트를 보내도록 AWS Config에 EventBridge 알리고 Amazon Simple Notification Service (EventBridge Amazon SNS) 주제를 시작합니다. SNS 주제는 비준수 리포지토리 정책 설명에 대해 이메일로 알려줍니다.

사전 조건 및 제한 사항

사전 조건

- 활성 상태의 AWS 계정.

- AWS Command Line Interface(AWS CLI), 설치 및 구성됨. 이에 대한 자세한 내용은 AWS CLI 설명서의 [AWS CLI 설치, 업데이트 및 제거](#)를 참조하세요.
- 정책 설명이 첨부되어 있는 기존 Amazon ECR 리포지토리가 테스트 환경에 설치 및 구성됩니다. 이에 대한 자세한 내용은 Amazon ECR 설명서의 [프라이빗 리포지토리 생성 및 리포지토리 정책 설명 설정](#)을 참조하세요.
- AWS Config, 원하는 AWS 리전에 구성됩니다. 이에 대한 자세한 내용은 AWS Config 설명서의 [AWS Config 시작하기](#)를 참조하세요.
- `aws-config-cloudformation.template` 파일(첨부), 로컬 시스템에 다운로드됩니다.

제한 사항

- 이 패턴의 솔루션은 리전의 솔루션이며 리소스는 동일한 리전에 생성되어야 합니다.

아키텍처

다음 다이어그램은 AWS Config가 Amazon ECR 리포지토리 정책 설명을 평가하는 방법을 보여줍니다.

이 다이어그램은 다음 워크플로우를 보여줍니다.

1. AWS Config는 사용자 지정 규칙을 시작합니다.
2. 사용자 지정 규칙은 Lambda 함수를 호출하여 Amazon ECR 리포지토리 정책 설명의 규정 준수를 평가합니다. 그런 다음 Lambda 함수는 규정을 준수하지 않는 리포지토리 정책 설명을 식별합니다.
3. Lambda 함수는 규정 미준수 상태를 AWS Config에 전송합니다.
4. AWS Config는 이 이벤트를 전송합니다. EventBridge
5. EventBridge 비준수 알림을 SNS 주제에 게시합니다.
6. Amazon SNS에서는 귀하 또는 승인된 사용자에게 이메일 알림을 전송합니다.

자동화 및 규모 조정

이 패턴의 솔루션은 Amazon ECR 리포지토리 정책 설명을 얼마든지 모니터링할 수 있지만 평가하려는 모든 리소스는 동일한 리전에서 생성되어야 합니다.

도구

- [AWS CloudFormation](#) — AWS는 AWS CloudFormation 리소스를 모델링 및 설정하고, 빠르고 일관되게 프로비저닝하고, 수명 주기 전반에 걸쳐 관리할 수 있도록 지원합니다. 템플릿을 사용하여 리소스와 해당 종속성을 설명하고, 리소스를 개별적으로 관리하는 대신 스택으로 함께 시작하고 구성할 수 있습니다. 여러 AWS 계정 및 AWS 리전에서 스택을 관리하고 프로비저닝할 수 있습니다.
- [AWS Config](#) - AWS Config는 AWS 계정의 AWS 리소스 구성에 대한 세부 정보 보기를 제공합니다. 이러한 보기에는 리소스 간에 어떤 관계가 있는지와 리소스가 과거에 어떻게 구성되었는지도 포함되므로, 시간이 지나면서 구성과 관계가 어떻게 변하는지 확인할 수 있습니다.
- [Amazon ECR](#) – Amazon Elastic Container Registry(Amazon ECR)는 안전하고 확장 가능하고 신뢰할 수 있는 관리형 컨테이너 이미지 레지스트리 서비스입니다. Amazon ECR은 IAM을 사용하여 리소스 기반 권한을 가진 프라이빗 리포지토리를 지원합니다.
- [Amazon EventBridge](#) — EventBridge Amazon은 애플리케이션을 다양한 소스의 데이터와 연결하는데 사용할 수 있는 서버리스 이벤트 버스 서비스입니다. EventBridge 애플리케이션, 서비스형 소프트웨어 (SaaS) 애플리케이션 및 AWS 서비스의 실시간 데이터 스트림을 AWS Lambda 함수, API 대상을 사용하는 HTTP 호출 엔드포인트 또는 다른 계정의 이벤트 버스와 같은 대상으로 전달합니다.
- [AWS Lambda](#) – AWS Lambda는 서버를 프로비저닝하거나 관리하지 않고도 코드를 실행하도록 지원하는 컴퓨팅 서비스입니다. Lambda는 필요 시에만 코드를 실행하며, 일일 몇 개의 요청에서 초당 수천 개의 요청까지 자동으로 규모를 조정합니다. 사용한 컴퓨팅 시간만큼만 비용을 지불하고, 코드가 실행되지 않을 때는 요금이 부과되지 않습니다.
- [Amazon SNS](#) – Amazon Simple Notification Service(SNS)는 웹 서버와 이메일 주소를 포함하여 게시자와 클라이언트 간에 메시지를 전달 또는 전송하는 것을 조정하고 관리합니다. 구독자는 구독하는 주제에 게시된 모든 메시지를 수신하며 주제에 대한 모든 구독자는 동일한 메시지를 수신합니다.

코드

이 패턴의 코드는 `aws-config-cloudformation.template` 파일(첨부)에서 확인할 수 있습니다.

에픽

AWS CloudFormation 스택 생성

작업	설명	필요한 기술
AWS CloudFormation 스택을 생성합니다.	<p>AWS CLI에서 다음 명령을 실행하여 AWS CloudFormation 스택을 생성합니다.</p> <pre> \$ aws cloudformation create-stack --stack-name=AWSConfigECR \ --template-body file://aws-config-cloudformation.template \ --parameters ParameterKey=<email>,ParameterValue= <myemail@example.com> \ --capabilities CAPABILITY_NAMED_IAM </pre>	AWS DevOps

AWS Config 사용자 지정 규칙 테스트

작업	설명	필요한 기술
AWS Config 사용자 지정 규칙을 테스트합니다.	<ol style="list-style-type: none"> 1. AWS Management Console에 로그인하여 AWS Config 콘솔을 연 다음 리소스를 선택합니다. 2. 리소스 인벤토리 페이지에서 리소스 카테고리, 리소스 유형, 규정 준수 상태별로 필터링할 수 있습니다. 	AWS DevOps

작업	설명	필요한 기술
	<p>3. <code>ecr:*</code>를 포함하는 Amazon ECR 리포지토리는 NON-COMPLIANT? 이고 <code>ecr:*</code>을(를) 포함하지 않는 Amazon ECR 리포지토리는 COMPLIANT 입니다.</p> <p>4. Amazon ECR 리포지토리에 비준수 정책 설명이 포함된 경우 SNS 주제를 구독한 이메일 주소로 알림이 전송됩니다.</p>	

첨부

이 문서와 관련된 추가 콘텐츠에 액세스하려면 [attachment.zip](#) 파일의 압축을 풉니다.

AWS CodeCommit 이벤트에서 사용자 지정 작업 수행

작성자: Abdullahi Olaoye(AWS)

환경: PoC 또는 파일럿

기술: DevOps; 관리 및 거버넌스

AWS 서비스: AWS CodeCommit, 아마존 SNS

요약

AWS CodeCommit 리포지토리를 사용하여 코드를 저장하는 경우 리포지토리를 모니터링하고 특정 이벤트가 발생할 때 작업 워크플로를 시작하고 싶을 수 있습니다. 예를 들어, 사용자가 커밋의 코드 줄에 댓글을 남기면 이메일 알림을 보내거나 커밋 후 리포지토리 콘텐츠에 대한 보안 스캔을 수행하도록 AWS Lambda 함수를 시작하길 원할 수도 있습니다. 이 패턴은 사용자 지정 작업을 위한 CodeCommit 리포지토리를 구성하는 단계를 설명합니다. 패턴은 AWS CodeCommit 알림 규칙을 사용하여 관심 있는 이벤트를 캡처한 다음 구성된 대상으로 해당 이벤트를 전송합니다.

사전 조건 및 제한 사항

사전 조건

- 활성 상태의 AWS 계정.
- Git 명령에 익숙해야 합니다.
- AWS CodeCommit, 설정하세요. 지침은 [AWS 설정을 참조하십시오 CodeCommit](#).
- (권장) AWS Command Line Interface(AWS CLI), 설치 및 구성합니다. 지침은 [AWS CLI 시작하기](#)를 참조하세요.

아키텍처

도구

서비스

- [CodeCommitAWS](#)는 안전한 Git 기반 리포지토리를 호스팅하는 완전 관리형 소스 제어 서비스입니다. 이를 통해 팀은 안전하고 확장성이 뛰어난 에코시스템에서 코드 작업을 손쉽게 수행할 수 있습니다. CodeCommit 자체 소스 제어 시스템을 운영하거나 인프라 확장에 대해 걱정할 필요가 없습니다.
- [Amazon Simple Notification Service\(SNS\)](#)는 애플리케이션, 최종 사용자 및 디바이스가 클라우드에서 알림을 전송하고 수신할 수 있게 해 주는 웹 서비스입니다. Amazon SNS는 처리량이 높은 푸시 기반 메시징을 위한 주제 (통신 채널) 를 제공합니다. many-to-many 게시자는 Amazon SNS 주제를 사용하여 Amazon Simple Queue Service (Amazon SQS) 대기열, AWS Lambda 함수, HTTP/S 웹후크 등 병렬 처리를 위해 많은 구독자에게 메시지를 배포할 수 있습니다. Amazon SNS를 사용하여 모바일 푸시, SMS 및 이메일을 사용하여 최종 사용자에게 알림을 전송할 수도 있습니다.

에픽

리포지토리 설정 CodeCommit

작업	설명	필요한 기술
CodeCommit 리포지토리 만들기.	CodeCommit 콘솔 또는 AWS CLI를 사용하여 리포지토리를 생성합니다. CodeCommit 지침은 CodeCommit 리포지토리 생성 을 참조하십시오.	DevOps 엔지니어
콘텐츠를 CodeCommit 리포지토리로 푸시합니다.	리포지토리를 생성한 후 Git 명령을 사용하여 리포지토리에 콘텐츠를 추가합니다. 기존 Git 리포지토리의 콘텐츠 또는 버전이 지정되지 않은 로컬 콘텐츠를 컴퓨터에서 마이그레이션할 수 있습니다. 지침은 리포지토리에 파일 추가 또는 AWS로 마이그레이션 을 참조하십시오 CodeCommit.	DevOps 엔지니어

Amazon SNS 설정

작업	설명	필요한 기술
SNS 주제를 생성합니다.	이 SNS 주제는 에서 이벤트를 CodeCommit 수신합니다. 지침은 Amazon SNS 주제 생성 을 참조하세요.	클라우드 아키텍트, DevOps 엔지니어
사용자 지정 작업을 위한 리소스를 생성합니다.	사용자 지정 작업을 수행하려면 해당 리소스를 만들어야 합니다. 예를 들어, 사용자 지정 작업이 Lambda 코드를 실행하고 메시지를 SQS 대기열로 보내는 것이라면 Lambda 함수와 SQS 대기열을 생성해야 합니다. 이메일 및 SMS 알림과 같은 작업에는 리소스가 필요하지 않습니다. 자세한 정보는 생성 중인 리소스의 유형에 대한 AWS 문서 를 참조하세요.	클라우드 아키텍트, DevOps 엔지니어
SNS 주제에 대한 사용자 지정 작업 리소스를 구독합니다.	사용자 지정 작업에 따라 적절한 프로토콜에 대한 구독을 생성합니다. 예를 들어, 이메일 알림을 받기 위해 이메일 주소를 구독하거나, 사용자 지정 코드를 실행하는 Lambda 함수를 구독하거나, Amazon SQS로 이벤트를 전송하기 위한 SQS 대기열을 구독할 수 있습니다. 이메일 및 SMS와 같은 구독 프로토콜의 경우, 각각 이메일 또는 전화 번호로 전송되는 링크를 통해 구독을 확인해야 합니다. 자세한 내용은 Amazon SNS 주제 구독 을 참조하세요.	클라우드 아키텍트, DevOps 엔지니어

알림 규칙 구성

작업	설명	필요한 기술
CodeCommit 리포지토리에 대한 알림 규칙을 생성합니다.	알림 규칙을 생성할 때 알림을 시작해야 하는 Git 이벤트를 선택하고 대상 유형으로 SNS 주제를 선택한 다음 이전에 생성한 SNS 주제를 선택합니다. 리포지토리에 여러 대상을 구성할 수도 있습니다. 지침은 알림 규칙 생성 을 참조하세요.	DevOps 엔지니어
사용자 지정 작업을 테스트합니다.	알림을 시작하도록 구성된 이벤트 중 하나를 수행합니다. 예를 들어, 해당 이벤트를 트리거로 선택한 경우 폴 요청을 생성합니다. 수행 중인 사용자 지정 작업이 나타납니다. 예를 들어, SNS 주제에 대해 이메일 주소를 구독한 경우 이메일 알림을 받습니다.	DevOps 엔지니어

관련 리소스

- [AWS CodeCommit 설명서](#)
- [Amazon SNS 설명서](#)
- [Git 설명서](#)

Amazon CloudWatch 메트릭을 CSV 파일에 게시

작성자: Abdullahi Olaoye(AWS)

환경: PoC 또는 파일럿

기술: DevOps

AWS 서비스: 아마존
CloudWatch

요약

이 패턴은 Python 스크립트를 사용하여 Amazon CloudWatch 지표를 검색하고 지표 정보를 쉼표로 구분된 값 (CSV) 파일로 변환하여 가독성을 높입니다. 이 스크립트는 지표를 검색해야 하는 AWS 서비스를 필수 인수로 사용합니다. AWS 리전 및 AWS 보안 인증 프로필을 선택적 인수로 지정할 수 있습니다. 이러한 인수를 지정하지 않는 경우 스크립트는 스크립트가 실행되는 워크스테이션에 구성된 기본 리전 및 프로필을 사용합니다. 스크립트가 실행되면 동일한 디렉터리에 CSV 파일이 생성되어 저장됩니다.

이 패턴으로 제공된 스크립트 및 관련 파일은 첨부 섹션을 참조하세요.

사전 조건 및 제한 사항

사전 조건

- Python 3.x
- AWS Command Line Interface(AWS CLI)

제한 사항

이 스크립트는 다음 AWS 서비스를 지원합니다.

- AWS Lambda
- Amazon Elastic Compute Cloud(Amazon EC2)
 - 기본적으로 스크립트는 Amazon Elastic Block Store(Amazon EBS) 볼륨 지표를 수집하지 않습니다. Amazon EBS 지표를 수집하려면 첨부된 `metrics.yaml` 파일을 수정해야 합니다.
- Amazon Relational Database Service(Amazon RDS)
 - 하지만 스크립트는 Amazon Aurora를 지원하지 않습니다.
- Application Load Balancer

- Network Load Balancer
- Amazon API Gateway

도구

- [CloudWatchAmazon](#)은 DevOps 엔지니어, 개발자, 사이트 신뢰성 엔지니어 (SRE) 및 IT 관리자를 위해 구축된 모니터링 서비스입니다. CloudWatch 애플리케이션을 모니터링하고, 시스템 전반의 성능 변화에 대응하고, 리소스 사용률을 최적화하고, 운영 상태를 통합적으로 파악하는 데 도움이 되는 데이터와 실행 가능한 통찰력을 제공합니다. CloudWatch 로그, 지표 및 이벤트의 형태로 모니터링 및 운영 데이터를 수집하고, AWS 및 온프레미스 서버에서 실행되는 AWS 리소스, 애플리케이션 및 서비스에 대한 통합 보기를 제공합니다.

에픽

사전 조건 설치 및 구성

작업	설명	필요한 기술
사전 조건을 설치합니다.	다음 명령을 실행합니다: <pre>\$ pip3 install -r requirements.txt</pre>	개발자
CLI를 구성합니다.	다음 명령을 실행합니다: <pre>\$ aws configure</pre>	개발자

Python 스크립트 구성

작업	설명	필요한 기술
스크립트를 엽니다.	스크립트의 기본 구성을 변경하려면 <code>metrics.yaml</code> 을 (를) 엽니다.	개발자

작업	설명	필요한 기술
스크립트 기간을 설정합니다.	<p>이는 가져오는 기간입니다. 기본 기간은 5분(300초)입니다. 기간을 변경할 수 있지만 다음 제한 사항에 유의하세요.</p> <ul style="list-style-type: none"> 지정한 시간 값이 3시간에서 15일 전 사이인 경우 해당 기간에 60초(1분)의 배수를 사용합니다. 지정한 시간 값이 15시간에서 63일 전 사이인 경우 해당 기간에 300초(5분)의 배수를 사용합니다. 지정한 시간 값이 63일 전보다 큰 경우 해당 기간에 3,600초(1시간)의 배수를 사용합니다. <p>그렇지 않으면 API 작업에서 데이터 포인트를 반환하지 않습니다.</p>	개발자
스크립트 시간을 설정합니다.	<p>이 값은 가져오고자 하는 지표 시간을 지정합니다. 기본값은 1시간입니다. 며칠 간의 지표를 검색하려면 값을 시간 단위로 제공합니다. 예를 들어, 2일의 경우 48을 지정합니다.</p>	개발자

작업	설명	필요한 기술
스크립트의 통계 값을 변경합니다.	(선택 사항) 글로벌 통계 값은 Average이며, 이는 특정 통계 값이 할당되지 않은 지표를 가져올 때 사용됩니다. 스크립트는 통계 값 Maximum, SampleCount 및 Sum을(를) 지원합니다.	개발자

Python 스크립트 실행

작업	설명	필요한 기술
스크립트를 실행합니다.	<p>다음 명령을 사용합니다.</p> <pre>\$ python3 cwreport.py <service></pre> <p>서비스 값 목록과 선택적 region 및 profile 파라미터를 보려면 다음 명령을 실행합니다.</p> <pre>\$ python3 cwreport.py -h</pre> <p>선택적 파라미터에 대한 자세한 내용은 추가 정보 섹션을 참조하세요.</p>	개발자

관련 리소스

- [AWS CLI 구성](#)
- [아마존 CloudWatch 지표 사용](#)

- [아마존 CloudWatch 설명서](#)
- [EC2 매트릭스 CloudWatch](#)
- [AWS Lambda 지표](#)
- [Amazon RDS 지표](#)
- [Application Load Balancer 지표](#)
- [Network Load Balancer 지표](#)
- [Amazon API Gateway 지표](#)

추가 정보

스크립트 사용

```
$ python3 cwreport.py -h
```

예제 구문

```
python3 cwreport.py <service> <--region=Optional Region> <--profile=Optional credential profile>
```

Parameters

- 서비스(필수) - 스크립트를 실행하려는 서비스입니다. 스크립트는 현재 AWS Lambda, Amazon EC2, Amazon RDS, Application Load Balancer, Network Load Balancer 및 API Gateway 서비스를 지원합니다.
- 리전(선택 사항) - 지표를 가져올 AWS 리전입니다. 기본 리전은 ap-southeast-1입니다.
- 프로필(선택 사항) - 사용할 AWS CLI의 명명된 프로필입니다. 이 파라미터를 지정하지 않으면 구성된 기본 보안 인증 프로필이 사용됩니다.

예제

- 기본 리전 ap-southeast-1 및 기본 구성 보안 인증 정보를 사용하여 Amazon EC2 지표를 가져오려면: `$ python3 cwreport.py ec2`
- 리전을 지정하고 API Gateway 지표를 가져오려면: `$ python3 cwreport.py apigateway --region us-east-1`

- AWS 프로필을 지정하고 Amazon EC2 지표를 가져오려면: `$ python3 cwreport.py ec2 --profile testprofile`
- Amazon EC2 지표를 가져올 리전과 프로필을 모두 지정하려면: `$ python3 cwreport.py ec2 --region us-east-1 --profile testprofile`

첨부

이 문서와 관련된 추가 콘텐츠에 액세스하려면 [attachment.zip](#) 파일의 압축을 풉니다.

pytest 프레임워크를 사용하여 AWS Glue에서 Python ETL 작업에 대한 유닛 테스트 실행

작성자: Praveen Kumar Jeyarajan(AWS) 및 Vaidy Sankaran(AWS)

코드 리포지토리: [aws-glue-jobs-unit-테스트](#)

환경: 프로덕션

기술: DevOps; 빅 데이터; 소프트웨어 개발 및 테스트

AWS 서비스: AWS CloudFormation, AWS, AWS CodeBuild, AWS CodeCommit CodePipeline, AWS Glue

요약

[로컬 개발 환경에서](#) AWS Glue의 Python 추출, 변환 및 로드 (ETL) 작업에 대한 단위 테스트를 실행할 수 있지만 DevOps 파이프라인에서 이러한 테스트를 복제하는 것은 어렵고 시간이 많이 걸릴 수 있습니다. 유닛 테스트는 AWS 기술 스택에서 메인프레임 ETL 프로세스를 현대화할 때 특히 어려울 수 있습니다. 이 패턴은 기존 기능을 그대로 유지하고, 새 기능을 출시할 때 주요 애플리케이션 기능이 중단되지 않도록 하고, 고품질 소프트웨어를 유지하면서 유닛 테스트를 간소화하는 방법을 보여줍니다. 이 패턴의 단계와 코드 샘플을 사용하여 AWS의 pytest 프레임워크를 사용하여 AWS Glue에서 Python ETL 작업에 대한 단위 테스트를 실행할 수 있습니다. CodePipeline 또한 이 패턴을 사용하여 여러 AWS Glue 작업을 테스트하고 배포할 수 있습니다.

사전 조건 및 제한 사항

사전 조건

- 활성 상태의 AWS 계정
- [Amazon ECR 퍼블릭 갤러리](#)에서 다운로드한 AWS Glue 라이브러리의 Amazon Elastic Container Registry(Amazon ECR) 이미지 URI
- 대상 AWS 계정 및 AWS 리전에 대한 프로필이 있는 Bash 터미널(모든 운영 체제)
- [파이썬 3.10](#) 이상
- [Pytest](#)

- AWS 서비스 테스트를 위한 [Moto](#) Python 라이브러리

아키텍처

기술 스택

- Amazon Elastic Container Registry (Amazon ECR)
- AWS CodeBuild
- AWS CodeCommit
- AWS CodePipeline
- Glue
- Pytest
- Python
- AWS Glue용 Python ETL 라이브러리

대상 아키텍처

다음 다이어그램은 Python을 기반으로 하는 AWS Glue ETL 프로세스의 단위 테스트를 일반적인 엔터프라이즈 규모의 AWS DevOps 파이프라인에 통합하는 방법을 설명합니다.

이 다이어그램은 다음 워크플로를 보여줍니다.

1. 소스 단계에서 샘플 Python ETL job (`sample.py`), 단위 테스트 파일 (`test_sample.py`), AWS CloudFormation 템플릿을 비롯한 소스 코드용 CodeCommit 리포지토리를 CodePipeline 사용합니다. 그런 다음 추가 처리를 위해 메인 브랜치에서 CodeBuild 프로젝트로 최신 코드를 CodePipeline 전송합니다.
2. 빌드 및 게시 스테이지에서는 AWS Glue 퍼블릭 Amazon ECR 이미지를 사용하여 이전 소스 단계의 최신 코드를 유닛 테스트합니다. 그런 다음 테스트 보고서가 CodeBuild 보고서 그룹에 게시됩니다. AWS Glue 라이브러리를 퍼블릭 Amazon ECR 리포지토리의 컨테이너 이미지에 로컬에서 AWS Glue에서 실행하는 데 필요한 모든 바이너리와 단위 테스트 [PySpark기반](#) ETL 작업이 포함되어 있습니다. 퍼블릭 컨테이너 리포지토리에는 AWS Glue에서 지원하는 버전마다 하나씩, 총 3개의 이미지 태그가 있습니다. 데모를 위해 이 패턴은 `glue_libs_4.0.0_image_01` 이미지 태그를 사용합니다. 에서 CodeBuild 이 컨테이너 이미지를 런타임 이미지로 사용하려면 사용하려는 이미지 태그

- 에 해당하는 이미지 URI를 복사한 다음 리소스의 GitHub 리포지토리에서 `pipeline.yml` 파일을 업데이트하십시오. `TestBuild`
3. 배포 단계에서 CodeBuild 프로젝트가 시작되고 모든 테스트를 통과하면 Amazon Simple Storage Service (Amazon S3) 버킷에 코드를 게시합니다.
 4. 사용자는 `deploy` 폴더의 CloudFormation 템플릿을 사용하여 AWS Glue 작업을 배포합니다.

도구

AWS 도구

- [Amazon Elastic Container Registry\(Amazon ECR\)](#)는 안전하고 확장 가능하고 신뢰할 수 있는 관리형 컨테이너 이미지 레지스트리 서비스입니다.
- [CodeBuildAWS](#)는 소스 코드를 컴파일하고, 단위 테스트를 실행하고, 배포할 준비가 된 아티팩트를 생성하는 데 도움이 되는 완전관리형 빌드 서비스입니다.
- [CodeCommitAWS](#)는 자체 소스 제어 시스템을 관리할 필요 없이 Git 리포지토리를 비공개로 저장하고 관리할 수 있는 버전 제어 서비스입니다.
- [AWS](#)는 소프트웨어 릴리스의 여러 단계를 신속하게 모델링 및 구성하고 소프트웨어 변경 사항을 지속적으로 릴리스하는 CodePipeline 데 필요한 단계를 자동화하도록 지원합니다.
- [AWS Glue](#)는 완전 관리형 ETL 서비스입니다. 이를 통해 데이터 스토어와 데이터 스트림 간에 데이터를 안정적으로 분류, 정리, 보강하고 이동할 수 있습니다.

기타 도구

- [Python](#)은 높은 수준의, 해석된 범용 프로그래밍 언어입니다.
- [Moto](#)는 AWS 서비스를 테스트하기 위한 Python 라이브러리입니다.
- [Pytest](#)는 애플리케이션 및 라이브러리의 복잡한 기능 테스트를 지원하도록 규모가 조정되는 소규모 유닛 테스트를 작성하기 위한 프레임워크입니다.
- AWS Glue용 [Python ETL 라이브러리는](#) AWS Glue용 PySpark 배치 작업의 로컬 개발에 사용되는 Python 라이브러리를 리포지토리입니다.

코드

이 패턴의 코드는 GitHub [aws-glue-jobs-unit-testing](#) 리포지토리에서 사용할 수 있습니다. 리포지토리에는 다음 리소스가 포함됩니다.

- src 폴더에 있는 샘플 Python 기반 AWS Glue 작업
- tests 폴더의 관련 유닛 테스트 사례(pytest 프레임워크를 사용하여 구축됨)
- 폴더에 있는 CloudFormation 템플릿 (YAML로 작성) deploy

모범 사례

리소스 보안 CodePipeline

의 파이프라인에 연결되는 소스 리포지토리에 암호화 및 인증을 사용하는 것이 가장 좋습니다. CodePipeline 자세한 내용은 설명서의 [보안 모범 사례](#)를 참조하십시오. CodePipeline

CodePipeline 리소스 모니터링 및 로깅

AWS 로깅 기능을 사용하여 사용자가 계정에서 어떤 작업을 수행하고 어떤 리소스를 사용하는지 확인하는 것이 가장 좋습니다. 로그 파일은 다음을 보여 줍니다.

- 작업의 시간과 날짜
- 작업의 소스 IP 주소
- 부족한 권한으로 인해 실패한 작업

로깅 기능은 AWS CloudTrail 및 Amazon CloudWatch 이벤트에서 사용할 수 있습니다. 를 사용하여 CloudTrail AWS 계정에서 또는 AWS 계정을 대신하여 이루어진 AWS API 호출 및 관련 이벤트를 기록할 수 있습니다. 자세한 내용은 CloudTrail CodePipeline 설명서에서 [AWS를 사용한 CodePipeline API 호출 로깅](#)을 참조하십시오.

CloudWatch 이벤트를 사용하여 AWS에서 실행되는 AWS 클라우드 리소스 및 애플리케이션을 모니터링할 수 있습니다. CloudWatch Events에서 알림을 생성할 수도 있습니다. 자세한 내용은 CodePipeline 설명서의 [CodePipeline 이벤트 모니터링](#)을 참조하십시오.

에픽

소스 코드 배포

작업	설명	필요한 기술
배포할 코드 아카이브를 준비하십시오.	1. GitHubaws-glue-jobs-unit-testing code.zip 리포지토리에서 다운로드하거나 명	DevOps 엔지니어

작업	설명	필요한 기술
	<p>명줄 도구를 사용하여.zip 파일을 직접 만드십시오. 예를 들어 터미널에서 다음 명령을 실행하여 Linux 또는 Mac에서.zip 파일을 만들 수 있습니다.</p> <pre data-bbox="630 520 1029 919">git clone https://github.com/aws-samples/aws-glue-jobs-unit-testing.git cd aws-glue-jobs-unit-testing git checkout master zip -r code.zip src/ tests/ deploy/</pre> <ol style="list-style-type: none"> 2. AWS Management Console에 로그인하고 원하는 AWS 리전을 선택합니다. 3. S3 버킷을 생성한 다음 생성한 S3 버킷에.zip 패키지 및 code.zip 파일(이전에 다운로드함)을 업로드합니다. 	

작업	설명	필요한 기술
CloudFormation 스택 생성.	<ol style="list-style-type: none"> 1. AWS 관리 콘솔에 로그인한 다음 CloudFormation 콘솔 을 엽니다. 2. 스택 페이지에서 스택 생성을 선택한 다음 기존 리소스 사용(리소스 가져오기)를 선택합니다. 3. 스택 생성 페이지의 템플릿 지정 섹션에서 템플릿 파일 업로드를 선택한 다음, Pipeline.yml 템플릿 (리포지토리에서 다운로드) 을 선택합니다. GitHub 그리고 다음을 선택합니다. 4. 스택 이름에 원하는 스택 이름을 glue-unit-testing-pipeline 입력하거나 선택합니다. 5. 예는 ApplicationStackName 미리 glue-codepipeline-app 채워진 이름을 사용하십시오. 파이프라인에서 생성된 CloudFormation 스택의 이름입니다. 6. 예는 BranchName 미리 채워진 마스터 이름을 사용하십시오. S3 버킷의.zip 파일에서 코드를 체크인하기 위해 CodeCommit 리포지토리에 생성된 브랜치의 이름입니다. 7. 의 BucketName 경우 미리 채워진 aws-glue-artifacts- 	AWS DevOps, DevOps 엔지니어

작업	설명	필요한 기술
	<p>us-east-1 버킷 이름을 사용하십시오. 이것은 .zip 파일이 들어 있는 S3 버킷의 이름이며, 파이프라인에서 코드 아티팩트를 저장하는 데 사용됩니다.</p> <p>8. 예는 미리 CodeZipFile 채워진 code.zip 값을 사용하십시오. 이것은 샘플 코드 S3 객체의 키 이름입니다. 객체는 .zip 파일이어야 합니다.</p> <p>9. 경우에는 미리 RepositoryName aws-glue-unit-testing 채워진 이름을 사용하십시오. 스택에서 생성된 CodeCommit 리포지토리의 이름입니다.</p> <p>10. TestReportGroupName 경우에는 미리 glue-unit-test-report 채워진 이름을 사용하십시오. 유닛 CodeBuild 테스트 보고서를 저장하기 위해 만든 테스트 보고서 그룹의 이름입니다.</p> <p>11. 다음을 선택한 후 스택 옵션 구성 페이지에서 다시 다음을 선택합니다.</p> <p>12. 검토 페이지의 기능에서 사용자 지정 이름으로 IAM 리소스를 생성할 CloudFormation 수 있음을 인정합니다. 옵션을 선택합니다.</p> <p>13. 제출을 선택합니다. 스택 생성이 완료되면 리소스 탭에</p>	

작업	설명	필요한 기술
	<p>서 생성된 리소스를 볼 수 있습니다. 스택 생성에는 약 5~7분 가량 걸립니다.</p> <p>스택은 .zip 파일에서 체크인되어 S3 버킷에 업로드된 초기 코드로 CodeCommit 리포지토리를 자동으로 생성합니다. 또한 스택은 CodeCommit 리포지토리를 소스로 사용하여 CodePipeline 뷰를 생성합니다. 위 단계에서 CodeCommit 리포지토리는 aws-glue-unit-test이고 파이프라인은 aws-glue-unit-test-pipeline 입니다.</p>	
<p>환경에서 리소스를 정리합니다.</p>	<p>추가 인프라 비용을 피하려면 이 패턴에 제공된 예제를 시험해 본 후 스택을 삭제해야 합니다.</p> <ol style="list-style-type: none"> 1. CloudFormation 콘솔을 열고 생성한 스택을 선택합니다. 2. 삭제를 선택합니다. 그러면 CodeCommit 리포지토리, AWS Identity and Access Management (IAM) 역할 또는 정책, 프로젝트를 포함하여 스택에서 생성한 모든 리소스가 삭제됩니다. CodeBuild 	<p>AWS DevOps, DevOps 엔지니어</p>

유닛 테스트 실행

작업	설명	필요한 기술
<p>파이프라인에서 유닛 테스트를 실행하십시오.</p>	<ol style="list-style-type: none"> 1. 배포된 파이프라인을 테스트하려면 AWS Management 콘솔에 로그인한 다음 CodePipeline 콘솔을 엽니다. 2. CloudFormation 스택에서 생성한 파이프라인을 선택한 다음 Release change 를 선택합니다. 파이프라인이 실행되기 시작합니다 (CodeCommit 리포지토리의 최신 코드 사용). 3. Test_and_Build 단계가 끝나면 세부 정보 탭을 선택한 다음 로그를 검사합니다. 4. 보고서 탭을 선택한 다음, 보고서 기록에서 테스트 보고서를 선택하여 유닛 테스트 결과를 확인합니다. 5. 배포 단계가 완료되면 AWS Glue 콘솔에서 배포된 AWS Glue 작업을 실행하고 모니터링합니다. 자세한 내용은 AWS Glue 설명서의 AWS Glue 모니터링을 참조하십시오. 	<p>AWS DevOps, DevOps 엔지니어</p>

문제 해결

문제	Solution
<p>Amazon S3, Amazon ECR 또는 CodeCommit 소스를 사용하는 파이프라인은 더 이상 자동으로 시작되지 않습니다.</p>	<p>변경 감지를 위해 Amazon EventBridge 또는 CloudWatch Events의 이벤트 규칙을 사용하는 작업의 구성 설정을 변경하는 경우, AWS Management Console은 소스 식별자가 유사하고 첫 문자가 동일한 경우 변경을 감지하지 못할 수 있습니다. 콘솔에서 새 이벤트 규칙을 생성하지 않으므로 파이프라인이 더 이상 자동으로 시작되지 않습니다.</p> <p>예를 들어 CodeCommit 브랜치 이름을 에서 MyTestBranch-1 로 변경하는 MyTestBranch-2 것은 사소한 변경입니다. 브랜치 이름 끝에 변경 사항이 적용되므로 소스 작업의 이벤트 규칙이 업데이트되지 않거나 새 소스 설정에 대한 규칙이 생성되지 않을 수 있습니다.</p> <p>이는 변경 감지를 위해 이벤트의 CloudWatch 이벤트를 사용하는 다음 소스 작업에 적용됩니다.</p> <ul style="list-style-type: none"> • 소스 작업이 Amazon S3에 있을 때의 S3 버킷 이름 및 S3 객체 키 파라미터 또는 콘솔 식별자 • 소스 작업이 Amazon ECR에 있을 때의 리포지토리 이름 및 이미지 태그 키 파라미터 또는 콘솔 식별자 • 소스 작업이 실행될 때의 리포지토리 이름 및 분기 이름 매개 변수 또는 콘솔 식별자 CodeCommit <p>이 문제를 해결하려면 다음 중 한 가지를 사용하십시오.</p>

문제	Solution
	<ul style="list-style-type: none"> • Amazon S3, Amazon ECR 또는 CodeCommit 에서 구성 설정을 변경하여 파라미터 값의 시작 부분이 변경되도록 하십시오. 예를 들어, 브랜치 이름을 release-branch 에서 2nd-release-branch 로 변경합니다. 이름 끝부분을 release-branch-2 처럼 변경하지 마십시오. • Amazon S3, Amazon ECR 또는 CodeCommit 각 파이프라인의 구성 설정을 변경합니다. 예를 들어, 브랜치 이름을 myRepo/myBranch 에서 myDeployRepo/myDeployBranch 로 변경합니다. 이름 끝부분을 myRepo/myBranch2 처럼 변경하지 마십시오. • AWS 관리 콘솔을 사용하는 대신, AWS 명령줄 인터페이스 (AWS CLI) 또는 CloudFormation AWS를 사용하여 변경 감지 이벤트 규칙을 생성하고 업데이트하십시오. Amazon S3 소스 작업에 대한 이벤트 규칙을 생성하는 방법에 대한 지침은 Amazon S3 소스 작업 및 CloudWatch 이벤트를 참조하십시오. Amazon ECR 작업에 대한 이벤트 규칙을 생성하는 방법에 대한 지침은 Amazon ECR 소스 작업 및 CloudWatch 이벤트를 참조하십시오. CodeCommit 작업에 대한 이벤트 규칙을 생성하는 방법에 대한 지침은 CodeCommit 소스 작업 및 CloudWatch 이벤트를 참조하십시오. 콘솔에서 작업 구성을 편집한 후에는 콘솔에서 생성한 업데이트된 변경 감지 리소스를 수락하십시오.

관련 리소스

- [AWS Glue](#)

- [로컬에서 AWS Glue 작업 개발 및 테스트](#)
- [AWS CloudFormation for AWS Glue](#)

추가 정보

또한, AWS CLI를 사용하여 AWS CloudFormation 템플릿을 배포할 수 있습니다. 자세한 내용은 [설명서의 변환이 포함된 빠른 템플릿 배포](#)를 참조하십시오. CloudFormation

Amazon S3에서 Helm v3 차트 리포지토리 설정

작성자: Abhishek Sharma(AWS)

환경: PoC 또는 파일럿

기술: DevOps; 컨테이너 및 마이크로서비스; 현대화

워크로드: 기타 모든 워크로드

AWS 서비스: Amazon S3

요약

이 패턴은 Amazon Web Services(AWS) 클라우드 기반 Amazon Simple Storage Service(S3)로 통합하여 Helm v3 리포지토리를 효과적으로 관리하도록 도와줍니다. 이 패턴을 사용하려면 Kubernetes와 Kubernetes 패키지 관리자인 Helm에 대해 잘 알고 있어야 합니다. Helm 리포지토리를 사용하여 차트를 저장하고 차트 버전을 관리하면 정전 중 평균 복원 시간(MTTR)을 개선할 수 있습니다.

이 패턴은 Helm 리포지토리 CodeCommit 생성에 AWS를 사용하며, S3 버킷을 Helm 차트 리포지토리로 사용하므로 조직 전체의 개발자가 차트를 중앙에서 관리하고 액세스할 수 있습니다.

사전 조건 및 제한 사항

사전 조건

- 활성 상태의 AWS 계정.
- Python 버전 2.7.12 이상
- pip
- 서브넷과 Amazon Elastic Compute Cloud(Amazon EC2) 인스턴스를 갖춘 Virtual Private Cloud(VPC)
- EC2 인스턴스에 Git를 설치합니다.
- S3 버킷을 생성하기 위한 AWS Identity and Access Management(IAM) 액세스
- 클라이언트 머신에서 Amazon S3에 대한 IAM(프로그래밍 방식 또는 역할) 액세스
- AWS CodeCommit 리포지토리
- AWS Command Line Interface(AWS CLI)

제품 버전

- Helm v3
- Python 버전 2.7.12 이상

아키텍처

대상 기술 스택

- Amazon S3
- AWS CodeCommit
- Helm
- Kubectl
- Python 및 pip
- Git
- helm-s3 플러그인

대상 아키텍처

자동화 및 규모 조정

- Helm을 기존의 지속적 통합/지속적 전달(CI/CD) 자동화 도구(이 패턴의 적용 범위 제외)에 통합하여 차트 Helm의 패키징 및 버전 제어를 자동화할 수 있습니다.
- GitVersion 또는 Jenkins 빌드 번호를 사용하여 차트의 버전 관리를 자동화할 수 있습니다.

도구

- [Helm](#)-Helm은 Kubernetes용 패키지 관리자, Kubernetes 클러스터에서 애플리케이션을 설치하고 관리할 때 유용합니다.
- [Amazon S3](#) – Amazon Simple Storage Service(S3)는 인터넷에 대한 스토리지입니다. Amazon S3를 사용하면 인터넷을 통해 언제 어디서든 원하는 양의 데이터를 저장하고 검색할 수 있습니다.
- [helm-s3 플러그인](#)-helm-s3 플러그인은 Amazon S3와의 상호 작용을 지원합니다. Helm v2 또는 Helm v3와 함께 사용할 수 있습니다.

에픽

Helm v3 설치 및 검증

작업	설명	필요한 기술
Helm v3 클라이언트를 설치합니다.	Helm 클라이언트를 로컬 시스템에 다운로드하고 설치하려면 <code>sudo curl https://raw.githubusercontent.com/helm/helm/main/scripts/get-helm-3 bash</code> 명령을 실행합니다.	클라우드 관리자, 엔지니어 DevOps
Helm 설치를 검사합니다.	Helm 클라이언트를 검증하려면 <code>helm version --short</code> 명령을 실행합니다.	클라우드 관리자, DevOps 엔지니어

S3 버킷을 Helm 리포지토리로 초기화

작업	설명	필요한 기술
차트 Helm을 위한 S3 버킷을 생성합니다.	고유한 S3 버킷을 생성합니다. 버킷에서 이름이 <code>stable/myapp</code> 인 폴더를 생성합니다. 이 패턴의 예시는 <code>s3://my-helm-charts/stable/myapp</code> 를 대상 차트 리포지토리로 사용합니다.	클라우드 관리자, DevOps 엔지니어
Amazon S3용 helm-s3 플러그인을 설치합니다.	helm-s3 플러그인을 클라이언트 컴퓨터에 설치하려면 <code>helm plugin install https://github.com</code>	클라우드 관리자, DevOps 엔지니어

작업	설명	필요한 기술
	<code>/hypnoglows/helm-s3</code> <code>.git</code> 명령을 실행합니다.	
Amazon S3 Helm 리포지토리를 초기화합니다.	Helm 리포지토리로 대상 폴더를 초기화하려면 <code>helm s3 init s3://my-helm-charts/stable/myapp</code> 명령을 사용합니다. 이 명령은 대상에 <code>index.yaml</code> 파일을 생성하여 해당 위치에 저장된 모든 차트 정보를 추적합니다.	클라우드 관리자, DevOps 엔지니어
새로 생성된 Helm 리포지토리를 확인합니다.	<code>index.yaml</code> 파일이 생성되었는지 확인하려면 <code>aws s3 ls s3://my-helm-charts/stable/myapp/</code> 명령을 실행합니다.	클라우드 관리자, DevOps 엔지니어
Amazon S3 리포지토리를 클라이언트 시스템의 Helm에 추가합니다.	대상 리포지토리 별칭을 Helm 클라이언트 시스템에 추가하려면 <code>helm repo add stable-myapp s3://my-helm-charts/stable/myapp/</code> 명령을 사용하세요.	클라우드 관리자, DevOps 엔지니어

Amazon S3 헬름 리포지토리에 차트를 패키징하고 게시합니다.

작업	설명	필요한 기술
차트 Helm을 복제하세요.	리포지토리에 로컬 헬름 차트가 없는 경우, 다음 명령어를 실행하여 CodeCommit GitHub 리포지토리에서 로컬 헬름 차	클라우드 관리자, 엔지니어 DevOps

작업	설명	필요한 기술
	<p>트를 복제하세요. <code>git clone <url_of_your_helm_source_code>.git</code></p>	
<p>로컬 차트 Helm을 패키징합니다.</p>	<p>생성 또는 복제한 차트를 패키징하려면 <code>helm package ./my-app</code> 명령을 사용합니다.</p> <p>예를 들어 이 패턴은 <code>my-app</code> 차트를 사용합니다. 이 명령은 <code>my-app</code> 차트 폴더의 모든 내용을 아카이브 파일로 패키징하며, 아카이브 파일은 <code>Chart.yaml</code> 파일에 언급된 버전 번호를 사용하여 이름이 지정됩니다.</p>	<p>클라우드 관리자, DevOps 엔지니어</p>
<p>Amazon S3 Helm 리포지토리에 로컬 패키지를 저장합니다.</p>	<p>Amazon S3에서 Helm 리포지토리에 로컬 패키지를 업로드하려면 <code>helm s3 push ./my-app-0.1.0.tgz stable-myapp</code> 명령을 실행합니다.</p> <p>명령에서 <code>my-app</code>는 차트 폴더 이름이며, <code>0.1.0</code>은 <code>Chart.yaml</code> 에서 언급한 차트 버전이고, <code>stable-myapp</code> 은 대상 리포지토리 별칭입니다.</p>	<p>클라우드 관리자, DevOps 엔지니어</p>
<p>차트 Helm을 검색하십시오.</p>	<p>차트가 로컬과 Amazon S3 Helm 리포지토리에 모두 표시되는지 확인하려면 <code>helm search repo stable-myapp</code> 명령을 실행합니다.</p>	<p>클라우드 관리자, DevOps 엔지니어</p>

Helm 리포지토리 업그레이드

작업	설명	필요한 기술
<p>차트를 수정하고 패키징합니다.</p>	<p>values.yaml 에서 replicaCount 값을 1로 설정한 다음 차트를 패키징합니다. 이번에는 버전을 Chart.yaml 에서 0.1.1로 변경합니다. 버전 제어는 CI/CD 파이프라인에서 Jenkins 빌드 번호와 같은 GitVersion 도구를 사용하여 자동화하는 것이 이상적입니다. 버전 번호 자동화는 이 패턴의 범위를 벗어납니다. 이 패키지를 설치하려면 <code>helm package ./my-app/</code> 명령을 실행합니다.</p>	<p>클라우드 관리자, 엔지니어 DevOps</p>
<p>Amazon S3의 Helm 리포지토리에 새 버전을 푸시합니다.</p>	<p>새 패키지 버전 0.1.1을 Amazon S3의 my-helm-chartsHelm 리포지토리로 푸시하려면 다음 명령을 실행합니다. <code>helm s3 push ./my-app-0.1.1.tgz stable-myapp</code></p>	<p>클라우드 관리자, 엔지니어 DevOps</p>
<p>차트 Helm을 확인합니다.</p>	<p>업데이트된 차트가 로컬과 Amazon S3 Helm 리포지토리에 모두 나타나는지 확인하려면 다음 명령을 실행합니다.</p> <pre>helm repo update</pre> <pre>helm search repo stable-myapp</pre>	<p>클라우드 관리자, DevOps 엔지니어</p>

Amazon S3 Helm 리포지토리에서 차트 검색 및 설치

작업	설명	필요한 기술
my-app 차트의 모든 버전을 검색합니다.	<p>사용 가능한 차트 버전을 모두 보려면 <code>--versions</code> 플래그를 사용하여 <code>helm search repo my-app --versions</code> 명령을 실행하세요.</p> <p>플래그가 없는 경우 Helm은 기본적으로 가장 최근에 업로드된 차트 버전을 표시합니다.</p>	DevOps 엔지니어
Amazon S3 Helm 리포지토리에서 차트를 설치합니다.	<p>자동 설치의 이 패턴의 범위를 벗어나지만 수동으로 설치할 수 있습니다. 이전 작업의 검색 결과에는 my-app 차트의 여러 버전이 표시됩니다. Amazon S3 Helm 리포지토리에서 새 버전(0.1.1)을 설치하려면 <code>helm upgrade --install my-app-release stable-my-app/my-app --version 0.1.1 --namespace dev</code> 명령을 사용하세요.</p>	DevOps 엔지니어

Helm을 사용하여 이전 버전으로 롤백

작업	설명	필요한 기술
특정 개정에 대한 세부 정보를 검토합니다.	<p>자동 롤백은 이 패턴의 범위를 벗어나지만 수동으로 이전 버전으로 롤백할 수 있습니다. 작동하는 버전으로 전환하거나 롤백하기 전에 그리고 수정 버전을 설치하기 전에 추가 검</p>	DevOps 엔지니어

작업	설명	필요한 기술
	<p>중 레이어를 추가하려면 helm get values --revision=2 my-app-release 명령을 사용하여 각 수정 버전에 전달된 값을 확인하세요.</p>	
<p>이전 버전으로 롤백합니다.</p>	<p>자동 롤백은 이 패턴의 범위를 벗어납니다. 이전 수정 버전으로 수동 롤백하려면 helm rollback my-app-release 1 명령을 사용하세요.</p> <p>이 예제는 개정 번호 1로 롤백하는 것입니다.</p>	<p>DevOps 엔지니어</p>

관련 리소스

- [Helm 설명서](#)
- [helm-s3 플러그인\(MIT 라이선스\)](#)
- [Amazon S3](#)

AWS 및 CodePipeline AWS CDK를 사용하여 CI/CD 파이프라인을 설정합니다.

작성자: Konstantin Zarudaev (AWS), Cizer Pereira (AWS), Lars Kinder (AWS), Yasha Dabas (AWS)

코드 리포지토리: CodePipeline
 in [CI/CD가 포함된 AWS](#)

환경: PoC 또는 파일럿

기술: DevOps

워크로드: 오픈 소스

AWS 서비스: AWS CodePipeline

요약

지속적 통합 및 지속적 전달(CI/CD)을 통해 소프트웨어 구축 및 릴리스 프로세스를 자동화하면 반복 가능한 구축을 지원하고 사용자에게 새로운 기능을 신속하게 제공할 수 있습니다. 각 코드 변경을 쉽고 빠르게 테스트할 수 있으며 소프트웨어를 릴리스하기 전에 버그를 발견하고 수정할 수 있습니다. 스테이징 및 릴리스 프로세스를 통해 각 변경 사항을 실행하여 애플리케이션 또는 인프라 코드의 품질을 확인할 수 있습니다. CI/CD는 애플리케이션 개발 팀이 코드 변경을 더 자주, 안정적으로 제공할 수 있도록 지원하는 문화, 운영 원칙, [관행](#)을 구현합니다. 이 구현을 CI/CD 파이프라인이라고도 합니다.

이 패턴은 Amazon Web Service(AWS)에서 재사용 가능한 지속적 통합 및 지속적 전달(CI/CD) 파이프라인을 정의합니다. AWS CodePipeline 파이프라인은 AWS [클라우드 개발 키트 \(AWS CDK\) v2](#)를 사용하여 작성되었습니다.

를 사용하면 AWS 관리 콘솔 인터페이스 CodePipeline, AWS 명령줄 인터페이스 (AWS CLI), AWS 또는 AWS SDK를 통해 소프트웨어 출시 프로세스의 여러 단계를 모델링할 수 있습니다.

CloudFormation 이 패턴은 AWS CDK를 사용한 구현 CodePipeline 및 구성 요소를 보여줍니다. 라이브러리 구성 외에도 AWS CDK에는 AWS CDK 앱과 상호 작용하는 기본 도구인 툴킷(CLI 명령 cdk)이 포함되어 있습니다. 다른 기능 중에서도 툴킷은 하나 이상의 스택을 CloudFormation 템플릿으로 변환하여 AWS 계정에 배포하는 기능을 제공합니다.

파이프라인에는 타사 라이브러리의 보안을 검증하기 위한 테스트가 포함되어 있으며, 지정된 환경에서 신속하고 자동화된 릴리스를 보장하는 데 도움이 됩니다. 검증 프로세스를 통해 애플리케이션의 전반적인 보안을 강화할 수 있습니다.

이 패턴의 목적은 배포하는 리소스가 모범 사례를 준수하도록 보장하는 동시에 CI/CD 파이프라인 사용을 가속화하여 코드를 배포하는 것입니다. DevOps [예제 코드를](#) 구현하면 린팅, 테스트, 보안 검사,

배포 및 배포 후 프로세스를 CodePipeline 갖춘 [AWS](#)를 갖게 됩니다. 이 패턴에는 Makefile 단계도 포함됩니다. 개발자는 Makefile을 사용하여 CI/CD 단계를 로컬에서 재현하고 개발 프로세스의 속도를 높일 수 있습니다.

사전 조건 및 제한 사항

사전 조건

- 활성 상태의 AWS 계정.
- 기본적인 이해는 다음과 같습니다.
 - AWS CDK
 - AWS CloudFormation
 - AWS CodePipeline
 - TypeScript

제한 사항

이 패턴은 [AWS CDK](#)를 TypeScript 용도로만 사용합니다. AWS CDK에서 지원하는 다른 언어는 다루지 않습니다.

제품 버전

다음 도구의 최신 버전을 사용하십시오.

- AWS Command Line Interface(AWS CLI)
- cfn_nag
- git-remote-codecommit
- Node.js

아키텍처

대상 기술 스택

- AWS CDK
- AWS CloudFormation
- AWS CodeCommit
- AWS CodePipeline

대상 아키텍처

파이프라인은 AWS CodeCommit 리포지토리 (SampleRepository) 의 변경에 의해 트리거됩니다. 처음에는 아티팩트를 CodePipeline 빌드하고 자체 업데이트한 다음 배포 프로세스를 시작합니다. 그 결과로 생성되는 파이프라인은 솔루션을 세 개의 독립적인 환경에 배포합니다.

- 개발 — 활성 개발 환경에서의 3단계 코드 검사
- 테스트 — 통합 및 회귀 테스트 환경
- Prod - 프로덕션 환경

개발 단계에 포함되는 세 단계는 린팅, 보안 및 유닛 테스트입니다. 이러한 단계는 병렬로 실행되어 프로세스 속도를 높입니다. 파이프라인이 작동하는 아티팩트만 제공하도록 하기 위해 프로세스의 한 단계가 실패할 때마다 파이프라인 실행이 중지됩니다. 개발 단계 배포 후 파이프라인은 검증 테스트를 실행하여 결과를 확인합니다. 성공하면 파이프라인은 배포 후 검증이 포함된 테스트 환경에 아티팩트를 배포합니다. 마지막 단계는 아티팩트를 프로덕션 환경에 배포하는 것입니다.

다음 다이어그램은 CodeCommit 리포지토리에서 수행된 빌드 및 업데이트 프로세스까지의 워크플로 CodePipeline, 세 가지 개발 환경 단계, 세 환경 각각에서의 후속 배포 및 검증을 보여줍니다.

도구

서비스

- [AWS Cloud Development Kit\(AWS CDK\)](#)는 AWS 클라우드 인프라를 코드로 정의하고 프로비저닝하는 데 도움이 되는 소프트웨어 개발 프레임워크입니다.
- [AWS는 AWS CloudFormation](#) 리소스를 설정하고, 빠르고 일관되게 프로비저닝하고, 전체 AWS 계정 및 지역의 수명 주기 전반에 걸쳐 리소스를 관리할 수 있도록 지원합니다. 이 패턴에서는 CloudFormation 템플릿을 사용하여 CodeCommit 리포지토리와 CodePipeline CI/CD 파이프라인을 생성할 수 있습니다.
- [CodeCommitAWS](#)는 자체 소스 제어 시스템을 관리할 필요 없이 Git 리포지토리를 비공개로 저장하고 관리할 수 있는 버전 제어 서비스입니다.
- [CodePipelineAWS](#)는 소프트웨어 릴리스의 여러 단계를 신속하게 모델링 및 구성하고 소프트웨어 변경 사항을 지속적으로 릴리스하는 데 필요한 단계를 자동화하는 데 도움이 되는 CI/CD 서비스입니다.
- [AWS Command Line Interface\(AWS CLI\)](#)는 명령줄 쉘에서 명령을 사용하여 AWS 서비스와 상호 작용할 수 있는 오픈 소스 도구입니다.

기타 도구

- [cfn_nag](#)는 [템플릿의](#) 패턴을 찾아 잠재적 보안 문제를 식별하는 오픈 소스 도구입니다.
CloudFormation
- [git-remote-codecommit](#) CodeCommit Git을 확장하여 리포지토리에서 코드를 푸시하고 가져오는 유틸리티입니다.
- [Node.js](#) 는 [확장](#) 가능한 네트워크 애플리케이션을 구축하기 위해 설계된 이벤트 기반 JavaScript 런타임 환경입니다.

코드

이 패턴의 코드는 GitHub [AWS CodePipeline with CI/CD 프랙티스](#) 리포지토리에서 제공됩니다.

모범 사례

AWS Identity 및 Access Management(IAM) 정책과 같은 리소스를 검토하여 해당 정책이 조직의 모범 사례에 부합하는지 확인합니다.

에픽

도구 설치

작업	설명	필요한 기술
macOS 또는 Linux에 도구를 설치합니다.	<p>MacOS 또는 Linux를 사용하는 경우, 선호하는 터미널에서 다음 명령을 실행하거나 Linux용 Homebrew를 사용하여 도구를 설치할 수 있습니다.</p> <pre>brew install brew install git-remot e-codecommit brew install ruby brew- gem brew-gem install cfn- nag</pre>	DevOps 엔지니어

작업	설명	필요한 기술
<p>AWS Cloud9를 사용하여 도구를 설치합니다.</p>	<p>AWS Cloud9를 사용하는 경우 다음 명령을 실행하여 도구를 설치합니다.</p> <pre>gem install cfn-nag</pre> <p>참고: AWS Cloud9에는 Node.js 및 npm이 설치되어 있어야 합니다. 설치 또는 버전을 확인하려면 다음 명령을 실행합니다.</p> <pre>node -v npm -v</pre>	<p>DevOps 엔지니어</p>
<p>AWS CLI를 설정합니다.</p>	<p>AWS CLI를 설정하려면, 운영 체제에 대한 지침을 사용합니다.</p> <ul style="list-style-type: none"> Windows: AWS CLI 자격 증명 도우미를 사용하여 Windows의 AWS CodeCommit 리포지토리에 대한 HTTPS 연결 설정 단계 리눅스, macOS, 유닉스: AWS CLI 자격 증명 도우미를 사용하여 리눅스, macOS 또는 유닉스의 AWS CodeCommit 리포지토리에 HTTPS 연결을 설정하기 위한 설정 단계 	<p>DevOps 엔지니어</p>

초기 배포 설정

작업	설명	필요한 기술
<p>코드를 다운로드하거나 복제합니다.</p>	<p>이 패턴에서 사용되는 코드를 가져오려면 다음 중 하나를 수행합니다.</p> <ul style="list-style-type: none"> • GitHub 리포지토리의 릴리스에서 최신 소스 코드를 다운로드하고 다운로드한 파일을 폴더에 압축 해제합니다. • 다음 명령을 실행하여 프로젝트를 복제합니다. <pre data-bbox="594 863 1029 1062">git clone --depth 1 https://github.com /aws-samples/aws-codepipeline-cicd.git</pre> <p>복제된 리포지토리에서 .git 디렉터리를 제거합니다.</p> <pre data-bbox="594 1220 1029 1377">cd ./aws-codepipeline-cicd rm -rf ./git</pre> <p>나중에 새로 만든 AWS CodeCommit 리포지토리를 원격 오리지인으로 사용하게 됩니다.</p>	<p>DevOps 엔지니어</p>
<p>AWS 계정에 연결합니다.</p>	<p>임시 보안 토큰 또는 랜딩 존 인증을 사용하여 연결할 수 있습니다. 올바른 계정과 AWS 리전을 사용하고 있는지 확인하려면 다음 명령을 실행하십시오.</p>	<p>DevOps 엔지니어</p>

작업	설명	필요한 기술
	<pre>AWS_REGION="eu-west-1" ACCOUNT_NUMBER=\$(aws sts get-caller-identit y --query Account -- output text) echo "\${ACCOUNT T_NUMBER}"</pre>	
<p>환경을 부트스트랩합니다.</p>	<p>AWS CDK 환경을 부트스트랩하려면 다음 명령을 실행합니다.</p> <pre>npm install npm run cdk bootstrap "aws://\${ACCOUNT_N UMBER}/\${AWS_REGION}"</pre> <p>환경을 성공적으로 부트스트랩한 후에는 다음 출력이 표시되어야 합니다.</p> <pre># Bootstrapping environment aws://{ac count}/{region}... # Environment aws:// {account}/{region} bootstrapped</pre> <p>AWS CDK 부트스트래핑에 대한 자세한 내용은 AWS CDK 설명서를 참조하세요.</p>	<p>DevOps 엔지니어</p>

작업	설명	필요한 기술
템플릿을 합성합니다.	<p>AWS CDK 앱을 합성하려면 <code>cdk synth</code> 명령을 사용합니다.</p> <pre data-bbox="597 394 1026 474">npm run cdk synth</pre> <p>다음과 같이 출력되어야 합니다.</p> <pre data-bbox="597 634 1026 1024">Successfully synthesized to <path-to-directory>/aws-codepipeline-cicd/cdk.out Supply a stack id (CodePipeline, DevMainStack) to display its template.</pre>	DevOps 엔지니어

작업	설명	필요한 기술
CodePipeline 스택 배포.	<p>이제 CloudFormation 템플릿을 부트스트랩하고 합성했으니 배포할 수 있습니다. 배포를 통해 CodePipeline 파이프라인과 리포지토리가 생성되며, 이 CodeCommit 리포지토리가 파이프라인의 소스이자 트리거가 됩니다.</p> <pre data-bbox="594 632 1029 793">npm run cdk -- deploy CodePipeline --require -aproval never</pre> <p>명령을 실행하면 CodePipeline 스택과 출력 정보가 성공적으로 배포된 것을 확인할 수 있습니다. 는 AWS 계정의 CodeCommit 리포지토리 이름을 CodePipeline.RepositoryName 제공합니다.</p> <pre data-bbox="594 1188 1029 1822">CodePipeline: deploying ... CodePipeline: creating CloudFormation changeset... # CodePipeline Outputs: CodePipeline.R epositoryName = SampleRepository Stack ARN: arn:aws:cloudformation :REGION:ACCOUNT-ID :stack/CodePipeline/ STACK-ID</pre>	DevOps 엔지니어

작업	설명	필요한 기술
<p>원격 CodeCommit 리포지토리 및 브랜치를 설정합니다.</p>	<p>배포에 CodePipeline 성공하면 AWS CodePipeline 콘솔에서 찾을 수 있는 파이프라인의 첫 번째 실행이 시작됩니다. AWS CDK는 기본 브랜치를 시작하지 CodeCommit 않기 때문에 이 초기 파이프라인 실행은 실패하고 다음 오류 메시지를 반환합니다.</p> <div data-bbox="597 682 1026 1075" style="border: 1px solid #ccc; border-radius: 10px; padding: 10px; background-color: #f9f9f9;"> <pre>The action failed because no branch named main was found in the selected AWS CodeComm it repository SampleRep ository. Make sure you are using the correct branch name, and then try again. Error: null</pre> </div> <p>이 오류를 수정하려면 원격 오리지널을 SampleRepository 로 설정하고 필요한 main 브랜치를 생성하십시오.</p> <div data-bbox="597 1333 1026 1822" style="border: 1px solid #ccc; border-radius: 10px; padding: 10px; background-color: #f9f9f9;"> <pre>RepoName=\$(aws cloudformation describe-stacks -- stack-name CodePipel ine --query "Stacks[0].Outputs[?OutputK ey=='RepositoryNam e'].OutputValue" -- output text) echo "\${RepoName}" # git init</pre> </div>	<p>DevOps 엔지니어</p>

작업	설명	필요한 기술
	<pre>git branch -m master main git remote add origin codecommit://\${RepoName} git add . git commit -m "Initial commit" git push -u origin main</pre>	

배포된 CodePipeline 파이프라인 테스트

작업	설명	필요한 기술
<p>파이프라인을 활성화하려면 변경 사항을 커밋하십시오.</p>	<p>초기 배포에 성공하면 SampleRepository 의 main 브랜치를 소스 브랜치로 사용하는 완전한 CI/CD 파이프라인을 갖추어야 합니다. main 브랜치에 변경 사항을 커밋하는 즉시 파이프라인은 다음과 같은 일련의 작업을 시작하고 실행합니다.</p> <ol style="list-style-type: none"> 1. CodeCommit 리포지토리어서 코드를 가져오세요. 2. 코드를 구축합니다. 3. 파이프라인 자체를 업데이트합니다(UpdatePipeline). 4. 린팅, 보안 및 유닛 테스트 검사를 위해 세 개의 병렬 작업을 실행합니다. 5. 성공하면 파이프라인은 Main 스택을 ./lib/mai 	<p>DevOps 엔지니어</p>

작업	설명	필요한 기술
	<p>n-stack.ts 에서 개발 환경으로 배포합니다.</p> <p>6. 배포된 리소스에 대해 배포 후 검사를 실행합니다. CodePipeline 콘솔에서 모든 CodePipeline 단계와 결과를 따를 수 있습니다.</p> <p>7. 성공하면 파이프라인은 테스트 및 프로덕션 환경에 대한 배포와 검증을 반복합니다.</p>	

Makefile을 사용하여 로컬에서 테스트합니다.

작업	설명	필요한 기술
Makefile을 사용하여 개발 프로세스를 실행합니다.	<p>make 명령을 사용하여 전체 파이프라인을 로컬에서 실행하거나 개별 단계를 실행할 수 있습니다(예: make linting).</p> <p>make를 사용하여 테스트하려면 다음 작업을 수행합니다.</p> <ul style="list-style-type: none"> 로컬 파이프라인 구현: make 유닛 테스트만 실행: make unittest 현재 계정에 배포: make deploy 환경 정리: make clean 	앱 개발자, DevOps 엔지니어

리소스 정리

작업	설명	필요한 기술
AWS CDK 앱 리소스를 삭제합니다.	<p>AWS CDK 앱을 정리하려면 다음 명령을 실행합니다.</p> <pre>cdk destroy --all</pre> <p>부트스트래핑 중에 생성된 Amazon Simple Storage Service(Amazon S3) 버킷은 자동으로 삭제되지 않는다는 점을 유의해 주십시오. 삭제를 허용하는 보존 정책이 필요하거나 AWS 계정에서 수동으로 삭제해야 합니다.</p>	DevOps 엔지니어

문제 해결

문제	Solution
템플릿이 예상대로 작동하지 않습니다.	<p>문제가 발생하여 템플릿이 작동하지 않으면 다음 사항을 확인해야 합니다.</p> <ul style="list-style-type: none"> 적절한 버전의 도구. 대상 AWS 계정에 대한 액세스(네트워크 연결) 대상 AWS 계정에 대한 충분한 권한

관련 리소스

- [IAM ID 센터에서 일반 작업을 시작해 보세요.](#)
- [AWS CodePipeline 설명서](#)
- [AWS CDK](#)

인증서 관리자 및 Let's Encrypt를 사용하여 Amazon EKS의 애플리케이션에 대한 end-to-end 암호화를 설정합니다.

작성자: Mahendra Siddappa(AWS) 및 Vasanth Jeyaraj(AWS)

코드 리포지토리: Amazon EKS의 End-to-end 암호화	환경: PoC 또는 파일럿	기술: DevOps; 컨테이너 및 마이크로서비스; 보안, ID, 규정 준수
워크로드: 기타 모든 워크로드	AWS 서비스: Amazon EKS, Amazon Route 53	

요약

end-to-end 암호화를 구현하는 것은 복잡할 수 있으며 마이크로서비스 아키텍처의 각 자산에 대한 인증서를 관리해야 합니다. 네트워크 로드 밸런서 또는 Amazon API Gateway를 사용하여 Amazon Web Services (AWS) 네트워크 엣지에서 전송 계층 보안 (TLS) 연결을 종료할 수 있지만 일부 end-to-end 조직에서는 암호화가 필요합니다.

이 패턴은 인그레스에 NGINX Ingress Controller를 사용합니다. Kubernetes 인그레스를 생성할 때 인그레스 리소스가 Network Load Balancer를 사용하기 때문입니다. Network Load Balancer는 클라이언트 인증서 업로드를 허용하지 않습니다. 따라서 Kubernetes 인그레스로는 상호 TLS를 달성할 수 없습니다.

이 패턴은 애플리케이션의 모든 마이크로서비스 간에 상호 인증이 필요한 조직을 위한 것입니다. 상호 TLS는 사용자 이름이나 암호를 유지 관리하는 부담을 줄이고 터키 보안 프레임워크를 사용할 수도 있습니다. 이 패턴의 접근 방식은 조직에 연결된 디바이스 수가 많거나 엄격한 보안 지침을 준수해야 하는 경우에 적합합니다.

이 패턴은 Amazon Elastic Kubernetes Service (Amazon EKS) 에서 실행되는 애플리케이션에 대한 end-to-end 암호화를 구현하여 조직의 보안 태세를 강화하는 데 도움이 됩니다. 이 패턴은 [Amazon EKS 리포지토리의 GitHub End-to-end 암호화에 있는 샘플 애플리케이션과 코드를 제공하여 Amazon EKS에서](#) end-to-end 암호화를 사용하여 마이크로서비스를 실행하는 방법을 보여줍니다. 이 패턴의 접근 방식은 [Let's Encrypt](#)를 인증 기관(CA)으로 사용하는 Kubernetes의 추가 기능인 [cert-manager](#)를 사용합니다. Let's Encrypt는 인증서 관리를 위한 비용 효율적인 솔루션으로 90일 동안 유효한 무료 인증

서를 제공합니다. Cert-manager는 Amazon EKS에 새 마이크로서비스를 배포할 때 인증서의 온디맨드 프로비저닝 및 교체를 자동화합니다.

대상

이 패턴은 Kubernetes, TLS, Amazon Route 53 및 도메인 이름 시스템(DNS) 사용 경험이 있는 사용자에게 권장됩니다.

사전 조건 및 제한 사항

사전 조건

- 활성 상태의 AWS 계정.
- 기존 Amazon EKS 클러스터.
- macOS, Linux 또는 Windows에 설치 및 구성된 AWS Command Line Interface(AWS CLI) 버전 1.7 이상.
- Amazon EKS 클러스터에 액세스하도록 설치 및 구성된 kubectl 명령줄 유틸리티. 이에 대한 자세한 내용은 Amazon EKS 설명서의 [kubectl 설치](#)를 참조하세요.
- 애플리케이션을 테스트하기 위한 기존 DNS 이름. 이에 대한 자세한 내용은 Amazon Route 53 설명서의 [Amazon Route 53을 사용하여 도메인 이름 등록](#)을 참조하세요.
- 로컬 컴퓨터에 설치된 Helm 최신 버전. [이에 대한 자세한 내용은 Amazon EKS 설명서 및 Helm 리포지토리의 Amazon EKS와 함께 헬름 사용을 참조하십시오. GitHub](#)
- [Amazon EKS 리포지토리의 GitHub End-to-end 암호화는](#) 로컬 시스템에 복제됩니다.
- [Amazon EKS 리포지토리의 복제된 GitHub End-to-end 암호화에](#) 있는 policy.json 및 trustpolicy.json 파일의 다음 값을 바꾸십시오.
 - <account number>-솔루션을 배포하려는 계정의 AWS 계정 ID로 바꿉니다.
 - <zone id>-도메인 이름의 Route 53 영역 ID로 바꿉니다.
 - <node_group_role>-Amazon EKS 노드와 연결된 AWS Identity and Access Management(IAM) 역할의 이름으로 바꿉니다.
 - <namespace>-NGINX Ingress Controller와 샘플 애플리케이션을 배포하는 Kubernetes 네임스페이스로 바꿉니다.
 - <application-domain-name>-Route 53의 DNS 도메인 이름으로 바꿉니다.

제한 사항

- 이 패턴은 인증서를 교체하는 방법을 설명하지 않으며 Amazon EKS의 마이크로서비스에서 인증서를 사용하는 방법만 설명합니다.

아키텍처

다음 다이어그램은 이 패턴의 워크플로 및 구성 요소를 보여 줍니다.

이 다이어그램은 다음 워크플로를 보여줍니다.

1. 클라이언트는 DNS 이름으로 애플리케이션 액세스 요청을 보냅니다.
2. Route 53 레코드는 Network Load Balancer에 대한 CNAME입니다.
3. Network Load Balancer는 TLS 리스너로 구성된 NGINX Ingress Controller로 요청을 전달합니다. NGINX Ingress Controller와 Network Load Balancer 간의 통신은 HTTPS 프로토콜을 따릅니다.
4. NGINX Ingress Controller는 애플리케이션 서비스에 대한 클라이언트의 요청에 따라 경로 기반 라우팅을 수행합니다.
5. 애플리케이션 서비스는 애플리케이션 포드로 요청을 전달합니다. 애플리케이션은 보안 암호를 호출하여 동일한 인증서를 사용하도록 설계되었습니다.
6. 포드는 cert-manager 인증서를 사용하여 샘플 애플리케이션을 실행합니다. NGINX Ingress Controller와 포드 간의 통신은 HTTPS를 사용합니다.

참고: Cert-manager는 자체 네임스페이스에서 실행됩니다. Kubernetes 클러스터 역할을 사용하여 인증서를 특정 네임스페이스에 보안 암호로 프로비저닝합니다. 해당 네임스페이스를 애플리케이션 포드 및 NGINX Ingress Controller에 연결할 수 있습니다.

도구

서비스

- [Amazon Elastic Kubernetes Service\(Amazon EKS\)](#)는 자체 Kubernetes 컨트롤 플레인이나 노드를 설치, 운영 및 유지 관리할 필요 없이 AWS에서 Kubernetes를 실행하는 데 사용할 수 있는 관리형 서비스입니다.
- [Elastic Load Balancing](#)은 여러 대상, 컨테이너 및 IP 주소에 걸쳐 수신되는 트래픽을 자동으로 분산합니다.

- [AWS Identity and Access Management\(IAM\)](#)를 이용하면 사용자에 대해 인증 및 권한 부여를 제어함으로써 AWS 리소스에 대한 액세스를 안전하게 관리할 수 있습니다.
- [Amazon Route 53](#)는 가용성과 확장성이 뛰어난 DNS 웹 서비스입니다.

기타 도구

- [cert-manager](#)는 인증서를 요청하고 Kubernetes 컨테이너에 배포하며 인증서 갱신을 자동화하는 Kubernetes의 추가 기능입니다.
- [NGINX Ingress Controller](#)는 Kubernetes 및 컨테이너화된 환경의 클라우드 네이티브 앱을 위한 트래픽 관리 솔루션입니다.

에픽

Route 53을 사용하여 퍼블릭 호스팅 영역 생성 및 구성

작업	설명	필요한 기술
Route 53에서 퍼블릭 호스팅 영역을 생성합니다.	<p>AWS Management Console에 로그인하여 Amazon Route 53 콘솔을 열고 호스팅 영역을 선택한 다음 호스팅 영역 생성을 선택합니다. 퍼블릭 호스팅 영역을 생성하고 영역 ID를 기록합니다. 이에 대한 자세한 내용은 Amazon Route 53 설명서의 퍼블릭 호스팅 영역 생성을 참조하세요.</p> <p>참고: ACME DNS01에서는 DNS 공급자를 사용하여 cert-manager가 인증서를 발급하도록 챌린지를 게시합니다. 이 챌린지는 해당 도메인 이름의 TXT 레코드에 특정 값을 입력하여 도메인 이름의 DNS 제어 권한이 사용자에게</p>	AWS DevOps

작업	설명	필요한 기술
	<p>있음을 증명하도록 요청합니다. Let's Encrypt가 ACME 클라이언트에게 토큰을 제공한 후 클라이언트는 해당 토큰과 계정 키에서 파생된 TXT 레코드를 생성하여 해당 레코드를 <code>_acme-challenge.<YOURDOMAIN></code> 에 저장합니다. 그런 다음 Let's Encrypt는 해당 레코드에 대해 DNS를 쿼리합니다. 일치하는 항목을 찾으면 인증서 발급을 진행할 수 있습니다.</p>	

cert-manager가 퍼블릭 호스팅 영역에 액세스할 수 있도록 IAM 역할 구성

작업	설명	필요한 기술
<p>cert-manager를 위한 IAM 정책을 생성합니다.</p>	<p>cert-manager에게 사용자가 Route 53 도메인을 소유하고 있는지 확인할 수 있는 권한을 제공하려면 IAM 정책이 필요합니다. <code>policy.json</code> 샘플 IAM 정책은 Amazon EKS 리포지토리의 복제된 GitHub End-to-end 암호화 1-IAMRole 디렉터리에 제공됩니다.</p> <p>AWS CLI에 다음 명령을 입력하여 IAM 정책을 생성합니다.</p> <pre>aws iam create-policy \ --policy-name PolicyForCertManager \</pre>	<p>AWS DevOps</p>

작업	설명	필요한 기술
	<pre>--policy-document file://policy.json</pre>	
<p>cert-manager를 위한 IAM 역할을 생성합니다.</p>	<p>IAM 정책을 생성한 후에는 IAM 역할을 생성해야 합니다. trustpolicy.json 샘플 IAM 역할은 1-IAMRole 디렉터리에 제공됩니다.</p> <p>AWS CLI에 다음 명령을 입력하여 IAM 역할을 생성합니다.</p> <pre>aws iam create-role \ --role-name RoleForCertManager \ --assume-role-policy-document file://trustpolicy.json</pre>	AWS DevOps
<p>정책을 역할에 연결합니다.</p>	<p>AWS CLI에 다음 명령을 입력하여 IAM 정책을 IAM 역할에 연결합니다. AWS_ACCOUNT_ID 를 AWS 계정의 ID로 바꿉니다.</p> <pre>aws iam attach-role-policy \ --policy-arn arn:aws:iam::AWS_ACCOUNT_ID:policy/PolicyForCertManager \ --role-name RoleForCertManager</pre>	AWS DevOps

Amazon EKS에서 NGINX Ingress Controller 설정

작업	설명	필요한 기술
<p>NGINX Ingress Controller를 배포합니다.</p>	<p>Helm을 사용하여 <code>nginx-ingress</code> 최신 버전을 설치합니다. 배포하기 전에 요구 사항에 따라 <code>nginx-ingress</code> 구성을 수정할 수 있습니다. 이 패턴은 주석이 달린 내부 Network Load Balancer를 사용하며 이를 <code>5-Nginx-Ingress-Controller</code> 디렉터리에서 사용할 수 있습니다.</p> <p><code>5-Nginx-Ingress-Controller</code> 디렉터리에서 다음 Helm 명령을 실행하여 NGINX Ingress Controller를 설치합니다.</p> <pre>helm install test-nginx nginx-stable/nginx-ingress -f 5-Nginx-Ingress-Controller/values_internal_nlb.yaml</pre>	AWS DevOps
<p>NGINX Ingress Controller가 설치되어 있는지 확인합니다.</p>	<p><code>helm list</code> 명령을 입력합니다. 출력에 NGINX Ingress Controller가 설치되어 있음이 표시되어야 합니다.</p>	AWS DevOps
<p>Route 53 A 레코드를 생성합니다.</p>	<p>A 레코드는 NGINX Ingress Controller에서 생성한 Network Load Balancer를 가리킵니다.</p>	AWS DevOps

작업	설명	필요한 기술
	<ol style="list-style-type: none"> 1. Network Load Balancer의 DNS 이름을 가져옵니다. 지침은 ELB 로드 밸런서의 DNS 이름 가져오기를 참조하세요. 2. Amazon Route 53 콘솔에서 호스팅 영역을 선택합니다. 3. 레코드를 생성하려는 퍼블릭 호스팅 영역을 선택한 다음 레코드 생성을 선택합니다. 4. 레코드의 이름을 입력합니다. 5. 레코드 유형에서 A-IPv4 및 일부 AWS 리소스로 트래픽 라우팅을 선택합니다. 6. 별칭을 활성화합니다. 7. 트래픽 라우팅 대상에서 다음을 수행합니다. <ol style="list-style-type: none"> a. Network Load Balancer에 대한 별칭을 선택합니다. b. Network Load Balancer가 배포된 AWS 리전을 선택합니다. c. Network Load Balancer의 DNS 이름을 입력합니다. 8. 레코드 생성을 선택합니다. 	

VirtualServer 아마존 EKS에서 NGINX를 설정하세요

작업	설명	필요한 기술
NGINX를 배포하세요. VirtualServer	<p>NGINX VirtualServer 리소스는 인그레스 리소스 대신 사용할 수 있는 로드 밸런싱 구성입니다. NGINX VirtualServer 리소스를 생성하기 위한 구성은 디렉터리의 <code>nginx_virtualserver.yaml</code> 파일에서 사용할 수 있습니다. 6-Nginx-Virtual-Server 다음 명령을 <code>kubectl</code> 입력하여 VirtualServer NGINX 리소스를 생성합니다.</p> <pre>kubectl apply -f nginx_virtualserver.yaml</pre> <p>중요: <code>nginx_virtualserver.yaml</code> 파일에서 애플리케이션 도메인 이름, 인증서 보안 암호 및 애플리케이션 서비스 이름을 업데이트해야 합니다.</p>	AWS DevOps
NGINX가 생성되었는지 확인하십시오 VirtualServer .	<p>다음 명령을 <code>kubectl</code> 입력하여 NGINX VirtualServer 리소스가 성공적으로 생성되었는지 확인합니다.</p> <pre>kubectl get virtualserver</pre>	AWS DevOps

작업	설명	필요한 기술
<p>TLS가 활성화된 상태로 NGINX 웹 서버를 배포합니다.</p>	<p>참고: Host 열이 애플리케이션의 도메인 이름과 일치하는지 확인하세요.</p> <p>이 패턴은 TLS가 활성화된 NGINX 웹 서버를 암호화 테스트용 애플리케이션으로 사용합니다. end-to-end 테스트 애플리케이션을 배포하는 데 필요한 구성 파일은 <code>demo-webserver</code> 디렉터리에서 사용할 수 있습니다.</p> <p>kubect1에 다음 명령을 입력하여 테스트 애플리케이션을 배포합니다.</p> <pre>kubect1 apply -f nginx-tls-ap.yaml</pre>	<p>AWS DevOps</p>

작업	설명	필요한 기술
<p>테스트 애플리케이션 리소스가 생성되었는지 확인합니다.</p>	<p>kubectl에 다음 명령을 입력하여 테스트 애플리케이션에 필요한 리소스가 생성되었는지 확인합니다.</p> <ul style="list-style-type: none"> • <code>kubectl get deployments</code> <p>참고: Ready 열과 Available 열을 검증하세요.</p> <ul style="list-style-type: none"> • <code>kubectl get pods grep -i example-deploy</code> <p>참고: 포드는 running 상태여야 합니다.</p> <ul style="list-style-type: none"> • <code>kubectl get configmap</code> • <code>kubectl get svc</code> 	<p>AWS DevOps</p>
<p>애플리케이션을 검증합니다.</p>	<ol style="list-style-type: none"> 1. <application-domain-name> 을 이전에 생성한 Route53 DNS 이름으로 바꾸어 다음 명령을 입력합니다. <pre>curl --verbose https://<application-domain-name></pre> <ol style="list-style-type: none"> 2. 애플리케이션에 액세스할 수 있는지 확인합니다. 	<p>AWS DevOps</p>

관련 리소스

AWS 리소스

- [Amazon Route 53 콘솔을 사용하여 레코드 생성](#)(Amazon Route 53 설명서)
- [Using a Network Load Balancer with the NGINX ingress controller on Amazon EKS](#)(AWS 블로그 게시물)

기타 리소스

- [Route 53](#)(cert-manager 설명서)
- [Configuring DNS01 Challenge Provider](#)(cert-manager 설명서)
- [Let's encrypt DNS 챌린지](#)(Let's Encrypt 설명서)

Flux를 사용하여 Amazon EKS 멀티테넌트 애플리케이션 배포를 간소화합니다.

작성자: 나딘 라하만 (AWS), 아디티야 암바티 (AWS), 아니켓 데카테 (AWS), 슈리칸트 파틸 (AWS)

코드 리포지토리: [aws-eks-multitenancy-deployment](#)

환경: PoC 또는 파일럿

기술: DevOps; 컨테이너 및 마이크로서비스

AWS 서비스: AWS, AWS CodeBuild, AWS CodeCommit CodePipeline, 아마존 EKS, 아마존 VPC

요약

제품 및 서비스를 제공하는 많은 회사는 내부 비즈니스 기능 간에 데이터 장벽을 유지해야 하는 데이터 규제 대상 산업입니다. 이 패턴은 Amazon Elastic Kubernetes Service (Amazon EKS) 의 멀티 테넌시 기능을 사용하여 단일 Amazon EKS 클러스터를 공유하는 테넌트 또는 사용자 간에 논리적이고 물리적으로 격리되는 데이터 플랫폼을 구축하는 방법을 설명합니다. 이 패턴은 다음과 같은 접근 방식을 통해 격리를 제공합니다.

- 쿠버네티스 네임스페이스 격리
- 역할 기반 액세스 제어(RBAC)
- 네트워크 정책
- 리소스 할당량
- AWS Identity and Access Management 서비스 계정의 (IAM) 역할 (IRSA)

또한 이 솔루션은 Flux를 사용하여 애플리케이션을 배포할 때 테넌트 구성을 변경할 수 없도록 유지합니다. 구성에 Flux kustomization.yaml 파일이 포함된 테넌트 리포지토리를 지정하여 테넌트 애플리케이션을 배포할 수 있습니다.

이 패턴은 다음을 구현합니다.

- Terraform 스크립트를 수동으로 배포하여 생성되는 AWS CodeCommit 리포지토리, AWS CodeBuild 프로젝트 및 AWS CodePipeline 파이프라인입니다.

- 테넌트 호스팅에 필요한 네트워크 및 컴퓨팅 구성 요소 이들은 Terraform을 CodeBuild 통해 CodePipeline 그리고 Terraform을 사용하여 생성됩니다.
- 테넌트 네임스페이스, 네트워크 정책 및 리소스 할당량은 Helm 차트를 통해 구성됩니다.
- Flux를 사용하여 배포한 다양한 테넌트에 속하는 애플리케이션.

고유한 요구 사항 및 보안 고려 사항에 따라 멀티테넌시를 위한 자체 아키텍처를 신중하게 계획하고 구축하는 것이 좋습니다. 이 패턴은 구현을 위한 출발점을 제공합니다.

사전 조건 및 제한 사항

사전 조건

- 활성 상태의 AWS 계정
- AWS Command Line Interface ([AWS CLI](#)) [버전 2.11.4 이상](#), 설치 및 구성
- [Terraform](#) 버전 0.12 이상이 로컬 컴퓨터에 설치됨
- [테라폼 AWS 공급자](#) 버전 3.0.0 이상
- [쿠버네티스 프로바이더](#) 버전 2.10 이상
- [헬름 프로바이더](#) 버전 2.8.0 이상
- [쿠백틀 프로바이더](#) 버전 1.14 이상

제한 사항

- Terraform 수동 배포에 대한 종속성: CodeCommit 리포지토리, CodeBuild 프로젝트 및 파이프라인 생성을 포함한 워크플로의 초기 설정은 수동 Terraform 배포에 의존합니다. CodePipeline 이는 인프라 변경 시 수동 개입이 필요하기 때문에 자동화 및 확장성 측면에서 잠재적인 제한을 초래합니다.
- CodeCommit 리포지토리 종속성: 워크플로는 CodeCommit 리포지토리를 소스 코드 관리 솔루션으로 사용하며 서비스와 긴밀하게 연계됩니다. AWS

아키텍처

대상 아키텍처

이 패턴은 다음 다이어그램에 나와 있는 것처럼 세 개의 모듈을 배포하여 데이터 플랫폼을 위한 파이프라인, 네트워크, 컴퓨팅 인프라를 구축합니다.

파이프라인 아키텍처:

네트워크 아키텍처:

컴퓨팅 아키텍처:

도구

서비스

- [AWS CodeBuild](#) 소스 코드를 컴파일하고, 단위 테스트를 실행하고, 배포할 준비가 된 아티팩트를 생성하는 데 도움이 되는 완전관리형 빌드 서비스입니다.
- [AWS CodeCommit](#) 자체 소스 제어 시스템을 관리할 필요 없이 Git 리포지토리를 비공개로 저장하고 관리할 수 있는 버전 제어 서비스입니다.
- [AWS CodePipeline](#) 소프트웨어 릴리스의 여러 단계를 신속하게 모델링 및 구성하고 소프트웨어 변경 사항을 지속적으로 릴리스하는 데 필요한 단계를 자동화할 수 있도록 도와줍니다.
- [Amazon Elastic Kubernetes Service \(Amazon EKS\)](#) 를 사용하면 자체 AWS 쿠버네티스 컨트롤 플레인 또는 노드를 설치하거나 유지 관리할 필요 없이 쿠버네티스를 실행할 수 있습니다.
- [AWS Transit Gateway](#) 는 Virtual Private Cloud(VPC)와 온프레미스 네트워크를 연결하는 중앙 허브입니다.
- [Amazon VPC \(Virtual Private Cloud\)](#) 를 사용하면 사용자가 정의한 가상 네트워크로 AWS 리소스를 시작할 수 있습니다. 이 가상 네트워크는 AWS의 확장 가능한 인프라를 사용한다는 이점과 함께 고객의 자체 데이터 센터에서 운영하는 기존 네트워크와 유사합니다.

기타 도구

- [실리움 네트워크 정책은 쿠버네티스 L3 및 L4 네트워킹 정책을](#) 지원합니다. L7 정책으로 확장하여 HTTP, Kafka, gRPC 및 기타 유사한 프로토콜에 대한 API 수준 보안을 제공할 수 있습니다.
- [Flux](#) 는 쿠버네티스에서 애플리케이션 배포를 자동화하는 Git 기반 지속적 전달 (CD) 도구입니다.
- [헬름](#) 은 쿠버네티스 클러스터에 애플리케이션을 설치하고 관리하는 데 도움이 되는 쿠버네티스용 오픈 소스 패키지 관리자입니다.
- [Terraform](#) 은 클라우드 및 온프레미스 리소스를 생성하고 관리하는 데 도움이 HashiCorp 되는 코드 형 인프라 (IaC) 도구입니다.

코드 리포지토리

이 패턴의 코드는 GitHub [EKS](#) 멀티테넌시 Terraform 솔루션 리포지토리에서 사용할 수 있습니다.

모범 사례

이 구현을 사용하기 위한 지침 및 모범 사례는 다음을 참조하십시오.

- [Amazon EKS 멀티테넌시 모범 사례](#)
- [플럭스 설명서](#)

에픽

Terraform 빌드, 테스트, 배포 단계를 위한 파이프라인 생성

작업	설명	필요한 기술
프로젝트 리포지토리를 복제하세요.	<p>터미널 창에서 다음 명령을 실행하여 GitHub EKS 멀티테넌시 Terraform 솔루션 리포지토리를 복제합니다.</p> <pre>git clone https://github.com/aws-samples/aws-eks-multitenancy-deployment.git</pre>	AWS DevOps
테라폼 S3 버킷과 아마존 DynamoDB를 부트스트랩합니다.	<p>1. bootstrap 폴더에서 bootstrap.sh 파일을 열고 S3 버킷 이름, DynamoDB 테이블 이름 및 다음에 대한 변수 값을 업데이트합니다. AWS 리전</p> <pre>S3_BUCKET_NAME=" S3_BUCKET_NAME>" DYNAMODB_TABLE_NAME=" DYNAMODB_NAME >"</pre>	AWS DevOps

작업	설명	필요한 기술
	<pre>REGION="<AWS_REGION>"</pre> <p>2. bootstrap.sh 스크립트 실행. 스크립트에는 필수 구성 AWS CLI 요소의 일부로 설치한 가 필요합니다.</p> <pre>cd bootstrap ./bootstrap.sh</pre>	

작업	설명	필요한 기술
<p>run.sh 및 locals.tf 파일을 업데이트하십시오.</p>	<ol style="list-style-type: none"> 부트스트랩 프로세스가 성공적으로 완료되면 스크립트 섹션에서 S3 버킷과 DynamoDB 테이블 이름을 복사합니다. variables bootstrap.sh <pre data-bbox="630 537 1029 779"># Variables S3_BUCKET_NAME=" S3_BUCKET_NAME>" DYNAMODB_TABLE_NAME =" <DYNAMODB_NAME"</pre> 해당 값을 프로젝트의 루트 디렉터리에 있는 run.sh 스크립트에 붙여넣습니다. <pre data-bbox="630 961 1029 1241">BACKEND_BUCKET_ID= "<SAME_NAME_AS_S3_ BUCKET_NAME>" DYNAMODB_ID=" <SAME_NAME_AS_DYNA MODB_NAME>"</pre> 프로젝트 코드를 CodeCommit 리포지토리에 업로드합니다. 파일에 다음 변수를 로 설정하여 Terraform을 통해 이 리포지토리를 자동으로 만들 수 true 있습니다. demo/pipeline/locals.tf <pre data-bbox="630 1661 1029 1780">create_new_repo = true</pre> 요구 사항에 따라 locals.tf 파일을 업데이트 	<p>AWS DevOps</p>

작업	설명	필요한 기술
	이트하여 파이프라인 리소스를 생성하세요.	
파이프라인 모듈 배포.	<p>파이프라인 리소스를 생성하려면 다음 Terraform 명령을 수동으로 실행하세요. 이러한 명령을 자동으로 실행하기 위한 오케스트레이션은 없습니다.</p> <pre>./run.sh -m pipeline -e demo -r <AWS_REGION> -t init ./run.sh -m pipeline -e demo -r <AWS_REGION> -t plan ./run.sh -m pipeline -e demo -r <AWS_REGION> -t apply</pre>	AWS DevOps

네트워크 인프라 생성

작업	설명	필요한 기술
파이프라인을 시작합니다.	<ol style="list-style-type: none"> 1. <code>templates</code> 폴더에서 <code>buildspec</code> 파일에 다음 변수가 로 설정되어 있는지 확인합니다 <code>network</code>. <pre>TF_MODULE_TO_BUILD: "network"</pre> 2. CodePipeline 콘솔의 파이프라인 세부정보 페이지에서 <code>Release change</code>를 선택하여 파이프라인을 시작합니다. 	AWS DevOps

작업	설명	필요한 기술
	<p>처음 실행한 후에는 CodeCommit 리포지토리 기본 브랜치에 변경 사항을 커밋할 때마다 파이프라인이 자동으로 시작됩니다.</p> <p>파이프라인에는 다음과 같은 단계가 포함됩니다.</p> <ul style="list-style-type: none"> • validateTerraform을 초기화하고, checkkov 및 tfsec 도구를 사용하여 Terraform 보안 스캔을 실행하고, 스캔 보고서를 S3 버킷에 업로드합니다. • plan Terraform 계획을 보여주고 계획을 S3 버킷에 업로드합니다. • applyS3 버킷의 Terraform 계획 출력을 적용하고 리소스를 생성합니다. AWS • destroy스테이지에서 생성된 AWS 리소스를 제거합니다. apply 이 선택적 단계를 활성화하려면 true demo/pipeline/locals.tf 파일에 다음 변수를 로 설정합니다. <pre style="border: 1px solid #ccc; border-radius: 10px; padding: 10px; margin-top: 10px;">enable_destroy_stage = true</pre>	

작업	설명	필요한 기술
네트워크 모듈을 통해 생성된 리소스를 검증합니다.	<p>파이프라인이 성공적으로 배포된 후 다음 AWS 리소스가 생성되었는지 확인하세요.</p> <ul style="list-style-type: none"> 퍼블릭 서브넷 3개와 프라이빗 서브넷 3개, 인터넷 게이트웨이, NAT 게이트웨이가 있는 이그레스 VPC입니다. 세 개의 프라이빗 서브넷이 있는 Amazon EKS VPC 각각 3개의 프라이빗 서브넷이 있는 테넌트 1과 테넌트 2 VPC 모든 VPC 연결 및 각 프라이빗 서브넷에 대한 경로를 포함하는 전송 게이트웨이. 대상 CIDR 블록이 인 Amazon EKS 송신 VPC의 정적 전송 게이트웨이 경로입니다. 0.0.0.0/0 이는 모든 VPC가 Amazon EKS 송신 VPC를 통해 아웃바운드 인터넷에 액세스할 수 있도록 하는 데 필요합니다. 	AWS DevOps

컴퓨팅 인프라 생성

작업	설명	필요한 기술
VPC에 대한 CodeBuild 프로젝트 액세스를 locals.tf 활성화하도록 업데이트하십시오.	Amazon EKS 사설 클러스터용 애드온을 배포하려면 CodeBuild 프로젝트를	AWS DevOps

작업	설명	필요한 기술
	<p>Amazon EKS VPC에 연결해야 합니다.</p> <ol style="list-style-type: none"> demo/pipeline 폴더에서 locals.tf 파일을 열고 변수를 로 설정합니다. vpc_enabled true run.sh 스크립트를 실행하여 파이프라인 모듈에 변경 내용을 적용합니다. <pre>demo/pipeline/locals.tf ./run.sh -m pipeline -env demo -region <AWS_REGION> -tfcmd init ./run.sh -m pipeline -env demo -region <AWS_REGION> -tfcmd plan ./run.sh -m pipeline -env demo -region <AWS_REGION> -tfcmd apply</pre>	
<p>buildspec 파일을 업데이트하여 컴퓨팅 모듈을 빌드합니다.</p>	<p>templates 폴더의 모든 buildspec YAML 파일에서 TF_MODULE_TO_BUILD 변수 값을 에서 network 로 compute 설정합니다.</p> <pre>TF_MODULE_TO_BUILD: "compute"</pre>	<p>AWS DevOps</p>

작업	설명	필요한 기술
<p>테넌트 관리 Helm 차트의 values 파일을 업데이트하십시오.</p>	<p>1. 다음 위치에서 values.yaml 파일을 엽니다.</p> <pre data-bbox="634 348 1029 506">cd cfg-terraform/demo /compute/cfg-tenant-mgmt</pre> <p>파일 모양은 다음과 같습니다.</p> <pre data-bbox="634 663 1029 1824">--- global: clusterRoles: operator: platform-tenant flux: flux-tenant-applier flux: tenantClusterBaseUrl: \${TEANT_BASE_URL} repoSecret: \${TENANT_REPO_SECRET} tenants: tenant-1: quotas: limits: cpu: 1 memory: 1Gi flux: path: overlays/tenant-1 tenant-2: quotas: limits: cpu: 1 memory: 2Gi flux:</pre>	<p>AWS DevOps</p>

작업	설명	필요한 기술
	<pre>path: overlays/tenant-2</pre> <p>2. global 및 tenants 섹션에서 요구 사항에 따라 구성을 업데이트하십시오.</p> <ul style="list-style-type: none"> • <code>tenantCloneBaseUrl</code> — 모든 테넌트의 코드를 호스팅하는 리포지토리의 경로 (모든 테넌트에 동일한 Git 리포지토리를 사용) • <code>repoSecret</code> — 글로벌 테넌트 Git 리포지토리에 인증하기 위해 SSH 키와 알려진 호스트를 보관하는 Kubernetes 시크릿 • <code>quotas</code> — 각 테넌트에 적용하려는 쿠버네티스 리소스 할당량 • <code>flux path</code> — 글로벌 테넌트 리포지토리의 테넌트 애플리케이션 YAML 파일 경로 	

작업	설명	필요한 기술
컴퓨팅 리소스 검증.	<p>이전 단계에서 파일을 업데이트하면 자동으로 CodePipeline 시작됩니다. 컴퓨팅 인프라를 위한 다음 AWS 리소스가 생성되었는지 확인하십시오.</p> <ul style="list-style-type: none"> • 프라이빗 엔드포인트가 있는 Amazon EKS 클러스터 • Amazon EKS 워커 노드 • Amazon EKS 애드온: 외부 비밀, 및 <code>aws-loadbalancer-controller metrics-server</code> • GitOps 모듈, 플렉스 헬름 차트, 실리움 헬름 차트, 테넌트 관리 헬름 차트 	AWS DevOps

테넌트 관리 및 기타 리소스 확인

작업	설명	필요한 기술
Kubernetes의 테넌트 관리 리소스를 검증하십시오.	<p>다음 명령을 실행하여 헬름의 도움을 받아 테넌트 관리 리소스가 성공적으로 생성되었는지 확인합니다.</p> <ol style="list-style-type: none"> 1. 에 지정된 대로 테넌트 네임스페이스가 생성되었습니다. <code>values.yaml</code> <pre>kubectl get ns -A</pre> <ol style="list-style-type: none"> 2. 할당량은 다음에 지정된 대로 각 테넌트 네임스페이스 	AWS DevOps

작업	설명	필요한 기술
	<p>에 할당됩니다. <code>values.yaml</code></p> <pre>kubectl get quota --namespace=<tenant_namespace></pre> <p>3. 할당량 세부 정보는 각 테넌트 네임스페이스에 대해 정확합니다.</p> <pre>kubectl describe quota cpu-memory-resource-quota-limit -n <tenant_namespace></pre> <p>4. 실리움 네트워크 정책이 각 테넌트 네임스페이스에 적용되었습니다.</p> <pre>kubectl get CiliumNetworkPolicy -A</pre>	

작업	설명	필요한 기술
<p>테넌트 애플리케이션 배포를 확인합니다.</p>	<p>다음 명령을 실행하여 테넌트 애플리케이션이 배포되었는지 확인합니다.</p> <ol style="list-style-type: none"> 1. Flux는 GitOps 모듈에 지정된 CodeCommit 리포지토리에 연결할 수 있습니다. <pre>kubectl get gitrepositories -A</pre> 2. Flux 커스터마이징 컨트롤러가 YAML 파일을 리포지토리에 배포했습니다. CodeCommit <pre>kubectl get kustomizations -A</pre> 3. 모든 애플리케이션 리소스는 테넌트 네임스페이스에 배포됩니다. <pre>kubectl get all -n <tenant_namespace></pre> 4. 각 테넌트에 대해 인그레스가 생성되었습니다. <pre>kubectl get ingress -n <tenant_namespace></pre> 	

문제 해결

문제	Solution
<p>다음과 유사한 오류 메시지가 나타납니다.</p> <pre>Failed to checkout and determine revision: unable to clone unknown error: You have successfully authenticated over SSH. You can use Git to interact with AWS CodeCommit.</pre>	<p>다음 단계에 따라 문제를 해결하십시오.</p> <ol style="list-style-type: none"> 1. 테넌트 애플리케이션 리포지토리 확인: 비어 있거나 잘못 구성된 리포지토리로 인해 오류가 발생할 수 있습니다. 테넌트 애플리케이션 리포지토리에 필수 코드가 포함되어 있는지 확인하십시오. 2. tenant_mgmt 모듈 재배포: tenant_mgmt 모듈 구성 파일에서 app 블록을 찾은 다음 deploy 매개변수를 다음과 같이 설정합니다. 0 <div data-bbox="867 890 1507 972" style="border: 1px solid #ccc; border-radius: 10px; padding: 5px; margin: 10px 0;"> <pre>deploy = 0</pre> </div> <p>Terraform apply 명령을 실행한 후 deploy 매개변수 값을 다시 다음과 같이 변경합니다. 1</p> <div data-bbox="867 1178 1507 1257" style="border: 1px solid #ccc; border-radius: 10px; padding: 5px; margin: 10px 0;"> <pre>deploy = 1</pre> </div> 3. 상태 재확인: 이전 단계를 실행한 후 다음 명령을 사용하여 문제가 지속되는지 확인하십시오. <div data-bbox="867 1440 1507 1520" style="border: 1px solid #ccc; border-radius: 10px; padding: 5px; margin: 10px 0;"> <pre>kubectl get gitrepositories -A</pre> </div> <p>문제가 지속되면 Flux 로그를 자세히 살펴보거나 Flux 일반 문제 해결 안내서를 참조하십시오.</p>

관련 리소스

- [테라폼용 Amazon EKS 블루프린트](#)
- [Amazon EKS 모범 사례 가이드, 멀티 테넌시 섹션](#)
- [플렉스 웹 사이트](#)
- [헬름 웹사이트](#)

추가 정보

다음은 테넌트 애플리케이션을 배포하기 위한 예제 리포지토리 구조입니다.

```
applications
sample_tenant_app
### README.md
### base
#   ### configmap.yaml
#   ### deployment.yaml
#   ### ingress.yaml
#   ### kustomization.yaml
#   ### service.yaml
### overlays
### tenant-1
#   ### configmap.yaml
#   ### deployment.yaml
#   ### kustomization.yaml
### tenant-2
###   configmap.yaml
###   kustomization.yaml
```

사용자 지정 리소스를 사용하여 여러 이메일 엔드포인트에서 SNS 주제를 구독하십시오.

작성자: Ricardo Morais(AWS)

환경: 프로덕션	기술: DevOps	AWS 서비스: 아마존 SNS, AWS CloudFormation, AWS Lambda
----------	------------	--

요약

참고, 2022년 8월: AWS는 CloudFormation 이제 `AWS::SNS::Topic` 객체 및 해당 구독 속성을 통해 여러 리소스의 구독을 지원합니다.

이 패턴은 Amazon Simple Notification Service(SNS) 주제에 대해 알림을 받을 이메일 주소를 구독 설정하는 방법을 설명합니다. AWS Lambda 함수를 AWS 템플릿의 사용자 지정 리소스로 사용합니다. CloudFormation Lambda 함수는 SNS 주제에 대한 이메일 엔드포인트를 지정하는 입력 파라미터와 연결됩니다.

현재는 AWS CloudFormation 템플릿 객체를 [AWS::SNS::Topic](#) 사용하여 단일 엔드포인트에서 SNS 주제를 [AWS::SNS::Subscription](#) 구독할 수 있습니다. 여러 엔드포인트를 구독하려면 객체를 여러 번 간접적으로 호출해야 합니다. Lambda 함수를 사용자 지정 리소스로 사용하면 입력 파라미터를 통해 여러 엔드포인트를 구독할 수 있습니다. 모든 AWS 템플릿에서 이 Lambda 함수를 사용자 지정 리소스로 사용할 수 있습니다. CloudFormation

사전 조건 및 제한 사항

사전 조건

- 활성 상태의 AWS 계정
- 액세스 키와 보안 키로 로컬 환경에 구성된 AWS 프로파일입니다. [AWS Cloud9](#)에서 이 코드를 실행할 수도 있습니다.
- 다음 항목에 대한 권한:
 - AWS Identity and Access Management(IAM) 역할 및 정책
 - AWS Lambda 함수
 - Lambda 함수 업로드를 위한 Amazon Simple Storage Service(S3)

- Amazon SNS 주제 및 정책
- AWS CloudFormation 스택

제한 사항

- 이 코드는 Linux 및 macOS 워크스테이션을 지원합니다.

제품 버전

- AWS Command Line Interface(AWS CLI) 버전 2 이상

아키텍처

대상 기술 스택

- AWS CloudFormation
- Amazon SNS
- AWS Lambda

도구

도구

- [AWS CLI 버전 2](#)

코드

첨부 파일에는 다음 파일이 포함됩니다.

- Lambda 함수: `lambda_function.py`
- AWS CloudFormation 템플릿: `template.yaml`
- 다중 또는 단일 이메일 엔드포인트 구독을 처리하기 위한 두 개의 파라미터 파일: `parameters-multiple-values.json` (기본값으로 사용됨) 및 `parameters-one-value.json`

스택을 배포하려면 두 파라미터 파일 중 하나를 사용할 수 있습니다. 여러 이메일 엔드포인트 지정:

```
./deploy.sh -p <YOUR_AWS_PROFILE_NAME> -r <YOUR_AWS_PROFILE_REGION>
```

단일 이메일 엔드포인트 지정:

```
./deploy.sh -p <YOUR_AWS_PROFILE_NAME> -r <YOUR_AWS_PROFILE_REGION> -f parameters-one-value.json
```

에픽

옵션 1 - 이메일 구독 한 번으로 SNS 주제 배포

작업	설명	필요한 기술
SNS 주제 구독을 위한 이메일 엔드포인트를 구성합니다.	파일 parameters-one-value.json (첨부)을 편집하고 사용할 이메일 주소(예: someone@example.com)를 반영하도록 pSNSNotificationsEmail 파라미터 값을 변경합니다.	
리소스 및 CloudFormation 구독을 생성하는 AWS 스택을 배포하십시오.	AWS 프로파일 이름, AWS 리전, parameters-one-value.json 파일을 사용하여 deploy.sh 명령을 실행합니다. <pre>./deploy.sh -p <YOUR_AWS_PROFILE_NAME> -r <YOUR_AWS_PROFILE_REGION> -f parameters-one-value.json</pre>	적절한 권한이 있는 IAM 역할입니다.

옵션 2 - 2개 이상의 이메일 구독이 포함된 SNS 주제 배포

작업	설명	필요한 기술
<p>SNS 주제 구독을 위한 이메일 엔드포인트를 구성합니다.</p>	<p>파일 <code>parameters-multiple-values.json</code> (첨부)을 편집하고 사용할 이메일 주소를 반영하도록 다음의 <code>someone1@example.com</code>, <code>someone2@example.com</code> 처럼 <code>pSNSNotificationsEmail</code> 파라미터 값을 쉼표로 구분하여 변경합니다.</p>	
<p>리소스 및 CloudFormation 구독을 생성하는 AWS 스택을 배포하십시오.</p>	<p>AWS 프로파일 이름 및 AWS 리전을 사용하여 <code>deploy.sh</code> 명령을 실행합니다. <code>parameters-multiple-values.json</code> 파일은 기본적으로 사용되므로 지정하지 않아도 됩니다.</p> <pre data-bbox="597 1188 1024 1381"> ./deploy.sh -p <YOUR_AWS_PROFILE_NAME> -r <YOUR_AWS_PROFILE_REGION> </pre>	<p>적절한 권한이 있는 IAM 역할입니다.</p>

옵션 3 - AWS CloudFormation 템플릿을 통해 SNS 주제 배포

작업	설명	필요한 기술
<p>SNS 주제를 생성합니다.</p>	<p>템플릿 객체에 구독 엔드포인트를 지정하지 않고 AWS CloudFormation 템플릿을 통해 SNS 주제를 생성합니다. <code>AWS::SNS::Topic</code></p>	<p>적절한 권한이 있는 IAM 역할입니다.</p>

작업	설명	필요한 기술
	첨부 파일에서 시작점으로 <code>template.yaml</code> 을 사용할 수 있습니다.	
SNS 주제 정책을 생성합니다.	AWS CloudFormation 템플릿에서 SNS 주제 정책을 생성합니다.	적절한 권한이 있는 IAM 역할입니다.
이메일 엔드포인트 목록에서 SNS 주제를 구독하십시오.	이메일 엔드포인트 목록(하나 이상)에 따라 생성한 SNS 주제에 엔드포인트를 구독하십시오.	적절한 권한이 있는 IAM 역할입니다.

관련 리소스

참조

- [AWS CloudFormation 사용자 지정 리소스 \(AWS 설명서\)](#)
- [Python, AWS Lambda 및 크레이퍼를 사용한 AWS CloudFormation 사용자 지정 리소스 생성 \(블로그 게시물\)](#)

필수 도구

- [AWS CLI 버전 2](#)

첨부

이 문서와 관련된 추가 콘텐츠에 액세스하려면 [attachment.zip](#) 파일의 압축을 풉니다.

인프라 코드의 테스트 기반 개발을 위해 Serverspec을 사용합니다

작성자: Sushant Jagdale(AWS)

환경: PoC 또는 파일럿	기술: DevOps; 인프라; 하이브리드 클라우드	AWS 서비스: 아마존 EC2, AWS, AWS CodeBuild, CodeDeploy
----------------	-----------------------------	--

요약

이 패턴은 Amazon Web Services(AWS) 클라우드에서 인프라 코드를 작성할 때 [Serverspec](#)을 사용하여 테스트 기반 개발(TDD)을 사용하는 방법을 보여줍니다. 이 패턴에는 AWS를 사용한 자동화도 포함됩니다. CodePipeline. TDD는 인프라 코드가 수행해야 하는 작업에 초점을 맞추고 수행에 대한 명확한 정의를 제시합니다. Serverspec을 사용하여 AWS CloudFormation, Terraform by HashiCorp 및 Ansible과 같은 도구로 만든 인프라를 테스트할 수 있습니다.

Serverspec은 인프라 코드를 리팩터링하는 데 도움이 됩니다. Serverspec을 사용하면 RSpec 테스트를 작성하여 다양한 패키지 및 소프트웨어의 설치를 확인하고, 명령을 실행하고, 실행 중인 프로세스와 포트를 확인하고, 파일 권한 설정을 확인하는 등의 작업을 수행할 수 있습니다. Serverspec은 서버가 올바르게 구성되어 있는지 확인합니다. 서버에는 Ruby만 설치합니다. 에이전트 소프트웨어를 설치할 필요가 없습니다.

테스트 기반 인프라는 다음과 같은 이점이 있습니다.

- 교차 플랫폼 테스트
- 기대치 검증
- 자동화에 대한 자신감
- 인프라 일관성 및 안정성
- 조기 실패

이 패턴을 사용하여 Apache 소프트웨어에 대한 Serverspec 유닛 테스트를 실행하고 Amazon Machine Image(AMI) 생성 중에 파일 권한 설정을 확인할 수 있습니다. AMI는 모든 테스트 사례를 통과한 경우에만 생성됩니다. Serverspec은 다음과 같은 테스트를 수행합니다.

- Apache 프로세스가 실행 중입니다.

- Apache 포트가 실행 중입니다.
- Apache 구성 파일 및 디렉토리는 특정 위치에 있는 식입니다.
- 파일 권한이 올바르게 구성되었습니다.

사전 조건 및 제한 사항

사전 조건

- 활성 상태의 AWS 계정
- AWS CodeBuild
- AWS CodeCommit
- AWS CodePipeline
- 퍼블릭 서브넷이 있는 Virtual Private Cloud(VPC)
- AWS Command Line Interface(AWS CLI) 및 Git 설치

제품 버전

- HashiCorp 팩커 버전: 1.6.6
- Ruby 버전 2.5.1 이상
- AWS CLI 버전 1.18.185

아키텍처

대상 아키텍처

1. CodeCommit 리포지토리로 코드를 푸시하면 Amazon CloudWatch Events 이벤트가 실행됩니다. CodePipeline 파이프라인의 첫 번째 단계에서는 에서 코드를 가져옵니다. CodeCommit
2. Packer 템플릿을 검증하고 빌드하는 두 번째 파이프라인 단계가 실행됩니다 CodeBuild.
3. Packer는 Packer 빌드 프로비저너의 일부로 Apache 및 Ruby 소프트웨어를 설치합니다. 그런 다음 제공자는 Serverspec을 사용하는 셸 스크립트를 호출하여 Apache 프로세스, 포트, 파일 및 디렉터리를 유닛 테스트합니다. Packer 포스트 프로세서는 실행 중에 Packer가 생성한 모든 아티팩트 목록이 포함된 JavaScript 객체 표기법 (JSON) 파일을 작성합니다.

4. 마지막으로 Packer에서 만든 AMI ID를 사용하여 Amazon Elastic Compute Cloud(Amazon EC2) 인스턴스가 생성됩니다.

도구

- [AWS CLI](#) – Amazon Command Line Interface(AWS CLI)는 명령줄 셸의 명령을 사용하여 AWS 서비스와 상호 작용하기 위한 오픈 소스 도구입니다.
- [Amazon CloudWatch Events](#) — Amazon CloudWatch Events는 Amazon Web Services (AWS) 리소스의 변경 사항을 설명하는 시스템 이벤트 near-real-time 스트림을 제공합니다.
- [AWS CodeBuild](#) — CodeBuild AWS는 클라우드의 완전 관리형 빌드 서비스입니다. CodeBuild 소스 코드를 컴파일하고, 단위 테스트를 실행하고, 배포할 준비가 된 아티팩트를 생성합니다.
- [AWS CodeCommit](#) — CodeCommit AWS는 아마존 웹 서비스에서 호스팅하는 버전 제어 서비스입니다. 를 CodeCommit 사용하여 클라우드의 자산 (예: 문서, 소스 코드, 바이너리 파일) 을 비공개로 저장하고 관리할 수 있습니다.
- [AWS CodePipeline](#) — CodePipeline AWS는 소프트웨어 릴리스에 필요한 단계를 모델링, 시각화 및 자동화하는 데 사용할 수 있는 지속적 전송 서비스입니다. 소프트웨어 릴리스 프로세스를 구성하는 여러 단계를 신속하게 모델링하고 구성할 수 있습니다.
- [HashiCorp HashiCorp Packer](#) — Packer는 단일 소스 구성에서 동일한 머신 이미지를 자동으로 생성하는 도구입니다.
- [Serverspec](#) - Serverspec은 RSpec 테스트를 실행하여 서버 구성을 확인합니다. Serverspec은 Ruby를 사용하므로 에이전트 소프트웨어를 설치할 필요가 없습니다.

code

코드가 첨부되었습니다. 코드는 디렉터리 3개와 파일 8개로 구성된 다음과 같은 구조를 사용합니다.

```
### amazon-linux_packer-template.json (Packer template)
### buildspec.yaml (CodeBuild .yaml file)
### pipeline.yaml (AWS CloudFormation template to automate CodePipeline)
### rspec_tests (RSpec required files and spec)
#   ### Gem-file
#   ### Rakefile
#   ### spec
#       ### apache_spec.rb
#       ### spec_helper.rb
### scripts
    ### rspec.sh (Installation of Ruby and initiation of RSpec)
```

에픽

AWS 보안 인증 정보 구성

작업	설명	필요한 기술
IAM 사용자를 생성합니다.	프로그래밍 및 콘솔 액세스를 제공하는 AWS ID 및 액세스 관리(IAM) 사용자를 생성합니다. 자세한 내용은 AWS 설명서 를 참조하십시오.	개발자, 시스템 관리자, 엔지니어 DevOps
AWS 보안 인증 정보 구성	로컬 컴퓨터 또는 사용자 환경에서 IAM 사용자에게 대한 AWS 보안 인증 정보를 구성합니다. 지침은 AWS 설명서 를 참조하십시오.	개발자, 시스템 관리자, DevOps 엔지니어
보안 인증 정보 테스트	구성된 보안 인증 정보를 검증하려면 다음 명령을 실행합니다. <pre>aws sts get-caller-identity --profile <profile></pre>	개발자, 시스템 관리자, DevOps 엔지니어

AWS CodePipeline

작업	설명	필요한 기술
CodeCommit 리포지토리 생성.	CodeCommit 리포지토리를 만들려면 다음 명령을 실행합니다.	개발자, 시스템 관리자, DevOps 엔지니어

작업	설명	필요한 기술
	<pre>aws codecommit create-repository --repository-name "<provide repository-name>" --repository-description "repository to unit test the infrastructure code"</pre>	
RSpec 테스트를 작성합니다.	인프라를 위한 RSpec 테스트 케이스를 생성합니다. 자세한 내용은 추가 정보 섹션을 참조하세요.	개발자, DevOps 엔지니어
CodeCommit 리포지토리로 코드를 푸시합니다.	<p>첨부된 코드를 CodeCommit 리포지토리로 푸시하려면 다음 명령을 실행합니다.</p> <pre>git clone <repository url> cp -R /tmp/<code folder>/ <repository_folder>/ git add . git commit -m"initial commit" git push</pre>	개발자, 시스템 관리자, DevOps 엔지니어
파이프라인을 생성합니다.	파이프라인을 생성하려면 추가 정보 섹션에 있는 AWS CLI 명령을 실행하세요.	개발자, 시스템 관리자, DevOps 엔지니어
파이프라인을 시작합니다.	CodeCommit 리포지토리에 코드를 커밋합니다. 리포지토리를 커밋하면 파이프라인이 시작됩니다.	개발자, 시스템 관리자, DevOps 엔지니어

작업	설명	필요한 기술
Apache URL을 테스트합니다.	<p>AMI 설치를 테스트하려면 다음 URL을 사용하세요.</p> <pre>http://<your instance public ip>/hello.html</pre> <p>페이지에 “Hello from Apache” 메시지가 표시됩니다.</p>	개발자, 시스템 관리자, DevOps 엔지니어

관련 리소스

- [HashiCorp](#)
- [HashiCorp 팩커](#)
- [Serverspec](#)
- [소개 ServerSpec: Serverspec이란 무엇이며 Stelligent에서는 이를 어떻게 사용하나요? \(외부 블로그 게시물\)](#)
- [테스트 기반 인프라 코드 개발\(외부 블로그 게시물\)](#)
- [HashiCorp Packer를 사용한 이미지 생성 및 테스트 ServerSpec \(외부 기사\)](#)

추가 정보

RSpec 테스트 작성

이 패턴에 대한 RSpec 테스트는 <repository folder>/rspec_tests/spec/apache_spec.rb에 있습니다.

```
require 'spec_helper'

describe service('httpd') do
  it { should be_enabled }
  it { should be_running }
end
```

```
describe port(80) do
  it { should be_listening }
end

describe file('/etc/httpd/conf/httpd.conf') do
  it { should exist }
  it { should be_owned_by 'root' }
  it { should contain 'ServerName www.example.com' }
end

describe file('/etc/httpd/conf/httpd.conf') do
  its(:content) { should match /ServerName www.example.com/ }
end

describe file('/var/www/html/hello.html') do
  it { should exist }
  it { should be_owned_by 'ec2-user' }
end

describe file('/var/log/httpd') do
  it { should be_directory }
end

describe file('/etc/sudoers') do
  it { should be_mode 440 }
end

describe group('root') do
  it { should have_gid 0 }
end
```

/spec 디렉터리에 자체 테스트를 추가할 수 있습니다.

파이프라인 생성

```
aws cloudformation create-stack --stack-name myteststack --template-body file://
pipeline.yaml --parameters ParameterKey=RepositoryName,ParameterValue=<provide
repository-name> ParameterKey=ApplicationName,ParameterValue=<provide
```

```
application-name> ParameterKey=SecurityGroupId,ParameterValue=<provide
SecurityGroupId> ParameterKey=VpcId,ParameterValue=<provide VpcId>
ParameterKey=SubnetId,ParameterValue=<provide SubnetId> ParameterKey=Region,ParameterValue=<pr
AccountId> --capabilities CAPABILITY_NAMED_IAM
```

파라미터 세부 정보

`repository-name`— AWS CodeCommit 리포지토리 이름

`application-name` - Amazon 리소스 이름(ARN)은 `ApplicationName`에 연결되어 있습니다. 이름을 원하는 대로 입력하세요

`SecurityGroupId` - 포트 80이 열려 있는 AWS 계정의 모든 보안 그룹 ID

`VpcId` - VPC의 ID

`SubnetId` - VPC의 퍼블릭 서브넷 ID

`Region` - 이 패턴을 실행하는 AWS 리전입니다

`Keypair` - EC2 인스턴스에 로그인하기 위한 Secure Shell(SSH) 키 이름

`AccountId` - AWS 계정 ID.

AWS Management Console을 사용하고 이전 명령줄에 있는 것과 동일한 파라미터를 전달하여 CodePipeline 파이프라인을 생성할 수도 있습니다.

첨부

이 문서와 관련된 추가 콘텐츠에 액세스하려면 [attachment.zip](#) 파일의 압축을 풉니다.

AWS의 타사 Git 소스 리포지토리 사용 CodePipeline

작성자: Kirankumar Chandrashekar(AWS)

환경: PoC 또는 파일럿	기술: DevOps	워크로드: 오픈 소스
AWS 서비스: AWS, AWS CodeBuild CodePipeline, AWS Lambda		

요약

이 패턴은 타사 Git 소스 CodePipeline 리포지토리와 함께 AWS를 사용하는 방법을 설명합니다.

[CodePipelineAWS](#)는 소프트웨어 구축, 테스트 및 배포 작업을 자동화하는 지속적 전송 서비스입니다. 이 서비스는 현재 GitHub [CodeCommitAWS](#)에서 관리하는 Git 리포지토리와 Atlassian Bitbucket을 지원합니다. 그러나 일부 기업은 AWS Single Sign-On(SSO) 서비스 및 Microsoft Active Directory와 통합된 타사 Git 리포지토리를 인증에 사용합니다. 사용자 지정 작업 및 웹후크를 CodePipeline 생성하여 이러한 타사 Git 리포지토리를 소스로 사용할 수 있습니다.

웹후크는 GitHub 리포지토리와 같은 다른 도구에서 이벤트를 탐지하고 이러한 외부 이벤트를 파이프라인에 연결하는 HTTP 알림입니다. 에서 CodePipeline 웹후크를 만들면 서비스가 Git 리포지토리 웹후크에서 사용할 수 있는 URL을 반환합니다. Git 리포지토리의 특정 브랜치에 코드를 푸시하면 Git 웹후크가 이 URL을 통해 웹후크를 시작하고 파이프라인의 소스 단계를 In Progress로 설정합니다. CodePipeline 파이프라인이 이 상태이면 작업 작업자가 사용자 지정 작업을 CodePipeline 폴링하고 작업을 실행한 다음 성공 또는 실패 상태를 에 보냅니다. CodePipeline 이 경우 파이프라인이 소스 단계에 있으므로 작업 워커는 Git 리포지토리의 콘텐츠를 가져와 콘텐츠를 압축한 다음 폴링된 작업에서 제공한 객체 키를 사용하여 파이프라인의 아티팩트가 저장되는 Amazon Simple Storage Service(S3) 버킷에 업로드합니다. 또한 사용자 지정 작업에 대한 전환을 CloudWatch Amazon의 이벤트와 연결하고 이벤트를 기반으로 작업 작업자를 시작할 수 있습니다. 이 설정을 사용하면 서비스가 기본적으로 소스로 지원하지 않는 타사 Git 리포지토리를 사용할 수 있습니다. CodePipeline

사전 조건 및 제한 사항

사전 조건

- 활성 상태의 AWS 계정.

- 웹후크를 지원하고 인터넷을 통해 웹후크 URL에 연결할 수 있는 Git 리포지토리 CodePipeline
- AWS Command Line Interface (AWS CLI)가 AWS 계정과 함께 작동하도록 [설치](#) 및 [구성](#)

아키텍처

패턴에는 다음 단계가 포함됩니다.

1. 사용자가 Git 리포지토리에 코드를 커밋합니다.
 2. Git 웹후크를 호출합니다.
 3. CodePipeline 웹후크가 호출됩니다.
 4. 파이프라인은 진행 중으로 설정되고 소스 단계는 진행 중 상태로 설정됩니다.
 5. 소스 스테이지 액션은 CloudWatch 이벤트 규칙을 시작하여 해당 규칙이 시작되었음을 나타냅니다.
 6. CloudWatch 이벤트는 Lambda 함수를 시작합니다.
 7. Lambda 함수는 사용자 지정 실행 작업의 세부 정보를 가져옵니다.
 8. Lambda 함수는 CodeBuild AWS를 시작하고 모든 작업 관련 정보를 전달합니다.
 9. CodeBuild Secrets Manager로부터 HTTPS Git 액세스를 위한 공개 SSH 키 또는 사용자 자격 증명을 가져옵니다.
 10. CodeBuild 특정 브랜치의 Git 리포지토리를 복제합니다.
 11. CodeBuild 아카이브를 압축하여 아티팩트 저장소 역할을 하는 S3 버킷에 업로드합니다.
- CodePipeline

도구

- [AWS CodePipeline](#) — CodePipeline AWS는 빠르고 안정적인 애플리케이션 및 인프라 업데이트를 위해 릴리스 파이프라인을 자동화하는 데 도움이 되는 완전관리형 [지속적 전송](#) 서비스입니다. CodePipeline 정의한 릴리스 모델을 기반으로 각 코드 변경에 대한 릴리스 프로세스의 빌드, 테스트 및 배포 단계를 자동화합니다. 이를 통해 기능과 업데이트를 신속하고 안정적으로 제공할 수 있습니다. CodePipeline AWS를 타사 서비스 (예: 사용자 지정 플러그인) 와 통합할 수 있습니다. GitHub
- [AWS Lambda](#) – AWS Lambda는 서버를 프로비저닝하거나 관리할 필요 없이 코드를 실행할 수 있습니다. Lambda를 사용하면 관리할 필요 없이 거의 모든 유형의 애플리케이션 또는 백엔드 서비스에 대한 코드를 실행할 수 있습니다. 코드를 업로드하면 고가용성을 유지한 채로 코드를 실행하고 확장

하는 데 필요한 모든 것을 Lambda가 알아서 처리해 줍니다. 코드가 다른 AWS 서비스에서 자동으로 시작되도록 설정하거나 어떤 웹 또는 모바일 앱에서도 코드를 직접 호출할 수 있습니다.

- [AWS CodeBuild](#) — CodeBuild AWS는 소스 코드를 컴파일하고, 테스트를 실행하고, 배포 준비가 완료된 소프트웨어 패키지를 생성하는 완전관리형 [지속적 통합](#) 서비스입니다. 사용하면 자체 빌드 서버를 프로비저닝, 관리 및 확장할 필요가 없습니다. CodeBuild CodeBuild 지속적으로 확장하고 여러 빌드를 동시에 처리하므로 빌드가 대기열에 남아 있지 않아도 됩니다. 사전 패키징된 빌드 환경을 사용하면 신속하게 시작할 수 있으며 혹은 자체 빌드 도구를 사용하는 사용자 지정 빌드 환경을 만들 수 있습니다.
- [AWS Secrets Manager](#) – Secrets Manager는 애플리케이션, 서비스, IT 리소스에 액세스하는 데 필요한 보안 암호를 지키도록 도와줍니다. 서비스를 사용하면 수명 주기 동안 데이터베이스 보안 인증 정보, API 키 및 기타 보안 암호를 교체, 관리 및 검색할 수 있습니다. 사용자와 애플리케이션은 민감한 정보를 일반 텍스트로 하드코딩할 필요 없이 Secrets Manager API를 호출하여 보안 정보를 검색합니다. Secrets Manager는 Amazon Relational Database Service(RDS), Amazon Redshift 및 Amazon DocumentDB에 대한 통합 기능이 내장되어 있는 암호 로테이션을 제공합니다. API 키 및 OAuth 토큰을 비롯한 다른 유형의 암호를 지원하도록 서비스를 확장할 수 있습니다. 또한 Secrets Manager를 사용하면 세분화된 권한을 사용하여 비밀에 대한 액세스를 제어하고, AWS 클라우드, 타사 서비스 및 온프레미스 환경의 리소스에 대한 비밀 로테이션을 중앙에서 감사할 수 있습니다.
- [Amazon CloudWatch](#) — CloudWatch Amazon은 DevOps 엔지니어, 개발자, 사이트 신뢰성 엔지니어 (SRE), IT 관리자를 위해 구축된 모니터링 및 관찰 서비스입니다. CloudWatch 애플리케이션을 모니터링하고, 시스템 전반의 성능 변화에 대응하고, 리소스 사용률을 최적화하고, 운영 상태를 통합적으로 파악할 수 있는 데이터와 실행 가능한 통찰력을 제공합니다. CloudWatch 로그, 지표 및 이벤트의 형태로 모니터링 및 운영 데이터를 수집하여 AWS 및 온프레미스 서버에서 실행되는 AWS 리소스, 애플리케이션 및 서비스에 대한 통합 보기를 제공합니다. 를 사용하여 환경의 비정상적인 동작을 감지하고, 경보를 설정하고, 로그와 지표를 나란히 시각화하고, 자동화된 조치를 취하고, 문제를 해결하고, 애플리케이션을 원활하게 CloudWatch 실행하기 위한 통찰력을 발견할 수 있습니다.
- [Amazon S3](#) - Amazon Simple Storage Service(S3)는 웹사이트, 모바일 애플리케이션, 백업 및 복원, 아카이브, 엔터프라이즈 애플리케이션, IoT 디바이스, 빅 데이터 분석 등 다양한 사용 사례에서 원하는 양의 데이터를 저장하고 보호할 수 있는 객체 스토리지 서비스입니다. Amazon S3는 데이터를 구성하고 특정 비즈니스, 조직 및 규정 준수 요구 사항에 맞게 세밀하게 조정된 액세스 제어를 구성하는 데 도움이 되는 easy-to-use 관리 기능을 제공합니다.

에픽

에서 사용자 지정 작업을 생성하십시오. CodePipeline

작업	설명	필요한 기술
<p>AWS CLI 또는 AWS를 사용하여 사용자 지정 작업을 생성합니다. CloudFormation</p>	<p>이 단계에는 특정 지역의 AWS 계정에 있는 파이프라인의 소스 단계에서 사용할 수 있는 사용자 지정 소스 작업을 생성하는 작업이 포함됩니다. 사용자 지정 소스 작업을 생성하려면 AWS CLI 또는 AWS CloudFormation (콘솔 아님)를 사용해야 합니다. 이 에픽과 다른 에픽에 설명된 명령 및 단계에 대한 자세한 내용은 이 패턴 끝에 있는 '관련 리소스' 섹션을 참조하세요. AWS CLI에서는 명령을 사용합니다. <code>create-custom-action-type --configuration-properties</code>를 사용하여 작업 작업자가 작업을 풀링할 때 처리하는 데 필요한 모든 파라미터를 제공합니다. CodePipeline 이 사용자 지정 소스 스테이지로 파이프라인을 생성할 때 동일한 값을 사용할 수 있도록 <code>--provider</code> 및 <code>--action-version</code> 옵션에 제공된 값을 기록합니다. 리소스 <code>AWS::CodePipeline::CustomAction</code> 유형 Type을 사용하여 CloudFormation AWS에서 사용자 지정 소스 작업을 생성할 수도 있습니다.</p>	<p>일반 AWS</p>

인증 설정

작업	설명	필요한 기술
SSH 키 페어를 생성합니다.	Secure Shell(SSH) 키 페어를 생성합니다. 지침은 GitHub 설명서를 참조하십시오.	시스템/엔지니어 DevOps
AWS Secrets Manager에서 보안 암호를 생성합니다.	SSH 키 페어에서 프라이빗 키의 콘텐츠를 복사하고 AWS Secrets Manager에서 암호를 생성합니다. 이 비밀번호는 Git 리포지토리에 액세스할 때 인증하는 데 사용됩니다.	일반 AWS
Git 리포지토리에 퍼블릭 키를 추가합니다.	개인 키에 대한 인증을 위해 SSH 키 쌍의 공개 키를 Git 저장소 계정 설정에 추가합니다.	시스템/엔지니어 DevOps

파이프라인 및 웹후크 생성

작업	설명	필요한 기술
사용자 지정 소스 작업이 포함된 파이프라인을 생성합니다.	에서 파이프라인을 CodePipeline 생성하세요. 소스 단계를 구성할 때 이전에 만든 사용자 지정 소스 작업을 선택합니다. AWS CodePipeline 콘솔 또는 AWS CLI에서 이 작업을 수행할 수 있습니다. CodePipeline 사용자 지정 작업에 설정한 구성 속성을 입력하라는 메시지가 표시됩니다. 이 정보는 작업 워커가 사용자 지정 작업에 대한 작업을 처리하는 데 필요합니다. 마법사의 지시에 따라 파	일반 AWS

작업	설명	필요한 기술
	이프라인의 다음 단계를 생성하세요.	
CodePipeline 웹훅 만들기.	<p>사용자 지정 소스 작업으로 생성한 파이프라인에 대한 웹훅을 생성합니다. 웹훅을 생성하려면 AWS CLI 또는 AWS CloudFormation (콘솔 아님)를 사용해야 합니다. AWS CLI에서 <code>put-webhook</code> 명령을 실행하고 웹훅 옵션에 적절한 값을 제공합니다. 명령이 반환하는 웹훅 URL을 기록합니다. CloudFormation AWS를 사용하여 웹훅을 생성하는 경우 리소스 유형을 <code>AWS::CodePipeline::Webhook</code> 사용하십시오. 생성된 리소스에서 웹훅 URL을 출력하고 기록합니다.</p>	일반 AWS

작업	설명	필요한 기술
<p>Lambda 함수 및 프로젝트를 생성합니다. CodeBuild</p>	<p>이 단계에서는 Lambda를 사용하여 사용자 지정 작업에 대한 작업 요청을 CodeBuild CodePipeline 폴링하고, 작업을 실행하고, 상태 결과를 반환하는 작업 작업자를 생성합니다. CodePipeline 파이프라인의 사용자 지정 소스 작업 단계가 “진행 중”으로 전환될 때 CloudWatch Amazon Events 규칙에 의해 시작되는 Lambda 함수를 생성합니다. Lambda 함수가 시작되면 작업 폴링을 통해 사용자 지정 작업 세부 정보를 가져와야 합니다. PollForJobs API를 사용하여 이 정보를 반환할 수 있습니다. 폴링된 작업 정보를 획득한 후 Lambda 함수는 승인을 반환한 다음 사용자 지정 작업에 대한 구성 속성에서 획득한 데이터로 정보를 처리해야 합니다. 작업자가 Git 리포지토리와 통신할 준비가 되면 CodeBuild 프로젝트를 시작할 수 있습니다. SSH 클라이언트를 사용하여 Git 작업을 처리하는 것이 편리하기 때문입니다.</p>	<p>일반 AWS, 코드 개발자</p>

에서 이벤트를 생성하세요. CloudWatch

작업	설명	필요한 기술
CloudWatch 이벤트 규칙을 생성합니다.	파이프라인의 사용자 지정 작업 단계가 “In Progress”로 전환될 때마다 Lambda 함수를 대상으로 시작하는 CloudWatch Events 규칙을 생성합니다.	일반 AWS

관련 리소스

에서 사용자 지정 작업 생성 CodePipeline

- [에서 사용자 지정 액션 만들기 및 추가 CodePipeline](#)
- [AWS::CodePipeline::CustomAction 리소스 입력](#)

인증 설정

- [AWS Secrets Manager를 사용한 암호 생성 및 관리](#)

파이프라인 및 웹훅 생성

- [에서 파이프라인 생성 CodePipeline](#)
- [put-webhook 명령 레퍼런스](#)
- [AWS::CodePipeline::Webhook 리소스](#)
- [PollForJobs API 참조](#)
- [에서 사용자 지정 작업 생성 및 추가 CodePipeline](#)
- [AWS에서 빌드 프로젝트 생성 CodeBuild](#)

이벤트 생성

- [Amazon CloudWatch Events를 사용하여 파이프라인 상태의 변화를 감지하고 이에 대응합니다.](#)

추가 참조

- [에서 파이프라인으로 작업하기 CodePipeline](#)
- [AWS Lambda 개발자 안내서](#)

AWS를 사용하여 테라폼 구성을 검증하기 위한 CI/CD 파이프라인을 생성합니다. CodePipeline

작성자: Aromal Raj Jayarajan(AWS), Vijesh Vijayakumaran Nair(AWS)

코드 리포지토리: aws-codepipeline-terraform-cicd -샘플	환경: PoC 또는 파일럿	기술: DevOps
워크로드: 기타 모든 워크로드	AWS 서비스: AWS CodeBuild, AWS CodeCommit, AWS CodePipeline, 아마존 S3, AWS ID 및 액세스 관리	

요약

이 패턴은 AWS에서 배포한 지속적 통합 및 지속적 전달 (CI/CD) 파이프라인을 사용하여 HashiCorp Terraform 구성을 테스트하는 방법을 보여줍니다. CodePipeline

Terraform은 코드를 사용하여 클라우드 인프라 및 리소스를 프로비저닝하고 관리하는 데 도움이 되는 명령줄 인터페이스 애플리케이션입니다. [이 패턴에서 제공하는 솔루션은 다음과 같은 5단계를 실행하여 Terraform 구성의 무결성을 검증하는 데 도움이 되는 CI/CD 파이프라인을 생성합니다.](#)

[CodePipeline](#)

1. “checkout” 테스트 중인 Terraform 구성을 AWS 리포지토리에서 가져옵니다. CodeCommit
2. “validate” [tfsec](#), [TFlint](#) 및 [checkov](#)를 포함한 [infrastructure-as-cod \(IaC\) 검증 도구를 실행합니다.](#) 또한 스테이지는 terraform validate 및 terraform fmt와 같은 Terraform IAc 유효성 검사 명령어를 실행합니다.
3. “plan”은 Terraform 구성이 적용된 경우 인프라에 어떤 변경 사항이 적용되는지를 보여 줍니다.
4. “apply”는 생성된 계획을 사용하여 테스트 환경에 필요한 인프라를 프로비저닝합니다.
5. “destroy”는 “apply” 단계에서 생성된 테스트 인프라를 제거합니다.

사전 조건 및 제한 사항

사전 조건

- 활성 상태의 AWS 계정
- AWS Command Line Interface(AWS CLI), [설치](#) 및 [구성됨](#)
- 로컬 머신에 설치 및 구성된 [Git](#)
- 로컬 머신에 설치 및 구성된 [Terraform](#)

제한 사항

- 이 패턴의 접근 방식은 AWS를 CodePipeline 하나의 AWS 계정과 AWS 지역에만 배포합니다. 다중 계정 및 다중 리전 배포에는 구성을 변경해야 합니다.
- 이 패턴이 제공하는 AWS Identity 및 Access Management(IAM) 역할(codePipeline_iam_role)은 최소 권한 원칙을 따릅니다. 파이프라인이 생성해야 하는 특정 리소스를 기반으로 이 IAM 역할의 권한을 업데이트해야 합니다.

제품 버전

- AWS CLI버전 2.9.15 이상
- Terraform 버전 1.3.7 이상

아키텍처

대상 기술 스택

- AWS CodePipeline
- AWS CodeBuild
- AWS CodeCommit
- AWS IAM
- Amazon Simple Storage Service(S3)
- AWS Key Management Service (AWS KMS)
- Terraform

대상 아키텍처

다음 다이어그램은 Terraform 구성을 테스트하기 위한 예제 CI/CD 파이프라인 워크플로를 보여줍니다. CodePipeline

이 다이어그램은 다음 워크플로를 보여줍니다.

1. CodePipeline에서는 AWS 사용자가 AWS CLI에서 `terraform apply` 명령을 실행하여 Terraform 계획에 제안된 작업을 시작합니다.
2. AWS는 CodePipeline 액세스에 필요한 정책 CodeCommit CodeBuild, AWS KMS 및 Amazon S3를 포함하는 IAM 서비스 역할을 맡습니다.
3. CodePipeline “checkout”파이프라인 단계를 실행하여 테스트를 위해 AWS CodeCommit 리포지토리에서 Terraform 구성을 가져옵니다.
4. CodePipeline IaC 검증 도구를 실행하고 프로젝트에서 Terraform IaC 검증 명령을 실행하여 Terraform 구성을 테스트하는 “validate” 단계를 실행합니다. CodeBuild
5. CodePipeline “plan”스테이지를 실행하여 Terraform 구성을 기반으로 CodeBuild 프로젝트에서 계획을 생성합니다. AWS 사용자는 변경 사항을 테스트 환경에 적용하기 전에 이 계획을 검토할 수 있습니다.
6. Code Pipeline은 CodeBuild 프로젝트를 사용하여 테스트 환경에 필요한 인프라를 프로비저닝하여 계획을 구현하는 “apply” 단계를 실행합니다.
7. CodePipeline “destroy”스테이지를 실행하여 “apply” 스테이지 중에 생성된 테스트 인프라를 제거하는 데 사용합니다 CodeBuild .
8. Amazon S3 버킷은 파이프라인 아티팩트를 저장하며, 이 아티팩트는 AWS KMS [고객 관리형 키](#)를 사용하여 암호화되고 복호화됩니다.

도구

도구

서비스

- [AWS](#)는 소프트웨어 릴리스의 여러 단계를 신속하게 모델링 및 구성하고 소프트웨어 변경 사항을 지속적으로 릴리스하는 CodePipeline 데 필요한 단계를 자동화하도록 지원합니다.
- [CodeBuildAWS](#)는 소스 코드를 컴파일하고, 단위 테스트를 실행하고, 배포할 준비가 된 아티팩트를 생성하는 데 도움이 되는 완전관리형 빌드 서비스입니다.
- [CodeCommitAWS](#)는 자체 소스 제어 시스템을 관리할 필요 없이 Git 리포지토리를 비공개로 저장하고 관리할 수 있는 버전 제어 서비스입니다.
- [AWS Identity and Access Management\(IAM\)](#)는 누구에게 인증 및 사용 권한이 있는지 제어하여 AWS 리소스에 대한 액세스를 안전하게 관리할 수 있도록 도와줍니다.

- [AWS Key Management Service\(AWS KMS\)](#)를 사용하면 암호화 키를 생성하고 제어하여 데이터를 보호할 수 있습니다.
- [Amazon Simple Storage Service\(S3\)](#)는 원하는 양의 데이터를 저장, 보호 및 검색하는 데 도움이 되는 클라우드 기반 객체 스토리지 서비스입니다.

기타 서비스

- [HashiCorp Terraform](#)은 코드를 사용하여 클라우드 인프라 및 리소스를 프로비저닝하고 관리하는 데 도움이 되는 명령줄 인터페이스 애플리케이션입니다.

code

이 패턴의 코드는 리포지토리에서 사용할 수 있습니다. GitHub [aws-codepipeline-terraform-cicdsamples](#) 리포지토리에는 이 패턴에 설명된 대상 아키텍처를 생성하는 데 필요한 Terraform 구성이 포함되어 있습니다.

에픽

솔루션 구성 요소 제공

작업	설명	필요한 기술
GitHub 리포지토리를 복제합니다.	<p>터미널 창에서 다음 명령을 실행하여 GitHub aws-codepipeline-terraform-cicdsamples 리포지토리를 복제합니다.</p> <pre>git clone https://github.com/aws-samples/aws-codepipeline-terraform-cicdsamples.git</pre> <p>자세한 내용은 GitHub 설명서의 리포지토리 복제를 참조하십시오.</p>	DevOps 엔지니어

작업	설명	필요한 기술
<p>Terraform 변수 정의 파일을 생성합니다.</p>	<p>사용 사례 요구 사항에 따라 terraform.tfvars 파일을 생성합니다. 복제된 리포지토리에 있는 examples/terraform.tfvars 파일의 변수를 업데이트할 수 있습니다.</p> <p>자세한 내용은 Terraform 설명서의 루트 모듈 변수에 값 할당을 참조하세요.</p> <p>참고: 리포지토리의 Readme.md 파일에는 필수 변수에 대한 자세한 정보가 포함되어 있습니다.</p>	<p>DevOps 엔지니어</p>
<p>AWS를 Terraform 공급자로 구성합니다.</p>	<ol style="list-style-type: none"> 1. 코드 편집기에서 복제된 리포지토리의 main.tf 파일을 엽니다. 2. 대상 AWS 계정에 연결을 설정하는 데 필요한 구성을 추가합니다. <p>자세한 내용은 Terraform 설명서의 AWS 공급자를 참조하세요.</p>	<p>DevOps 엔지니어</p>

작업	설명	필요한 기술
<p>Amazon S3 복제 버킷을 생성하기 위한 Terraform 공급자 구성을 업데이트합니다.</p>	<ol style="list-style-type: none"> 다음 명령을 실행하여 리포지토리의 S3 디렉터리를 엽니다. <div data-bbox="630 394 1029 474" style="border: 1px solid #ccc; border-radius: 10px; padding: 5px; margin: 10px 0;"> <pre>cd ./modules/s3</pre> </div> tf 파일의 region 값을 업데이트하여 Amazon S3 복제 버킷을 생성하기 위한 Terraform 공급자 구성을 업데이트합니다. Amazon S3가 객체를 복제할 리전을 입력해야 합니다. (선택 사항) 기본적으로 Terraform은 로컬 상태 파일을 상태 관리에 사용합니다. Amazon S3를 원격 백엔드로 추가하려면 Terraform 구성을 업데이트해야 합니다. 자세한 내용은 Terraform 설명서의 백엔드 구성을 참조하세요. <p>참고: S3 복제를 활성화하면 Amazon S3 버킷 전체에 걸쳐 객체를 비동기식으로 자동 복사할 수 있습니다.</p>	<p>DevOps 엔지니어</p>

작업	설명	필요한 기술
Terraform 구성을 초기화합니다.	<p>Terraform 구성 파일이 포함된 작업 디렉토리를 초기화하려면 복제된 리포지토리의 루트 폴더에서 다음 명령을 실행하세요.</p> <pre data-bbox="594 489 1027 569">terraform init</pre>	DevOps 엔지니어
Terraform 플랜을 생성합니다.	<p>Terraform 계획을 생성하려면 복제된 리포지토리의 루트 폴더에서 다음 명령을 실행하세요.</p> <pre data-bbox="594 825 1027 982">terraform plan --var-file=terraform.tfvars -out=tfplan</pre> <p>참고: Terraform은 구성 파일을 평가하여 선언된 리소스의 대상 상태를 결정합니다. 그런 다음 대상 상태를 현재 상태와 비교하고 계획을 생성합니다.</p>	DevOps 엔지니어
Terraform 계획을 검증합니다.	Terraform 계획을 검토하고 대상 AWS 계정에 필요한 아키텍처가 구성되어 있는지 확인합니다.	DevOps 엔지니어

작업	설명	필요한 기술
솔루션을 배포합니다.	<ol style="list-style-type: none"> Terraform 플랜을 적용하려면 복제된 리포지토리의 루트 폴더에서 다음 명령을 실행하세요. <div style="border: 1px solid #ccc; border-radius: 10px; padding: 10px; margin: 10px 0;"> <pre>terraform apply "tfplan"</pre> </div> <ol style="list-style-type: none"> 예를 입력하여 리소스를 배포하고자 함을 확인합니다. <p>참고: Terraform은 구성 파일에 명시된 목표 상태를 달성하기 위해 인프라를 생성, 업데이트 또는 파괴합니다.</p>	DevOps 엔지니어

파이프라인을 실행하여 Terraform 구성을 검증합니다.

작업	설명	필요한 기술
소스 코드 리포지토리를 설정합니다.	<ol style="list-style-type: none"> Terraform 출력에서 검증하려는 Terraform 구성이 포함된 리포지토리의 소스 리포지토리 세부 정보를 가져옵니다. AWS Management Console에 로그인합니다. 그런 다음 CodeCommit 콘솔을 엽니다. main이라는 이름의 소스 리포지토리에 새 브랜치를 생성합니다. 지침은 설명서의 CodeCommitAWS에서 브랜 	DevOps 엔지니어

작업	설명	필요한 기술
	<p>치 생성을 CodeCommit 참조하십시오.</p> <p>4. 소스 리포지토리의 main 브랜치를 로컬 워크스테이션에 복제합니다. 지침은 설명서에서 AWS CLI 자격 증명 도우미를 사용하여 Windows의 AWS CodeCommit 리포지토리에 HTTPS 연결을 위한 설정 단계를 참조하십시오.</p> <p>CodeCommit</p> <p>5. 다음 명령을 실행하여 GitHub aws-codepipeline-terraform-cicdsamples 리포지토리에서 templates 폴더를 복사합니다.</p> <pre data-bbox="630 1066 1029 1226">cp -r templates \$YOUR_CODECOMMIT_REPO_ROOT</pre> <p>참고: templates 폴더에는 소스 리포지토리의 루트 디렉터리에 대한 빌드 사양 파일과 유효성 검사 스크립트가 들어 있습니다.</p> <p>6. 필요한 Terraform IaC 구성을 소스 리포지토리의 루트 폴더에 추가합니다.</p> <p>7. 프로젝트의 Terraform 구성에 원격 백엔드에 대한 세부 정보를 추가합니다. 자세한</p>	

작업	설명	필요한 기술
	<p>내용은 Terraform 설명서의 패턴 S3를 참조하세요</p> <p>8. (선택 사항) templates 폴더의 변수를 업데이트하여 사전 구성된 스캔, 도구 변경 버전을 활성화 또는 비활성화하고 사용자 지정 스크립트 파일에 디렉토리를 지정합니다. 자세한 내용은 이 패턴의 추가 정보 섹션을 참조하세요.</p> <p>9. 변경 내용을 소스 리포지토리의 main 브랜치에 푸시합니다.</p>	

작업	설명	필요한 기술
<p>파이프라인 단계를 검증합니다.</p>	<ol style="list-style-type: none"> 1. AWS Management Console에 로그인하고 CodePipeline 콘솔을 엽니다. 2. 이전 에픽 섹션의 <code>terraform apply "tfplan"</code> 커맨드에서 생성된 출력에서 생성된 CodePipeline에픽의 이름을 찾으세요. 3. CodePipeline 콘솔에서 파이프라인을 열고 Release change를 선택합니다. 4. 각 파이프라인 단계를 검토하고 예상대로 작동하는지 확인합니다. <p>자세한 내용은 AWS CodePipeline 사용 설명서의 파이프라인 세부 정보 및 기록 보기 (콘솔) 를 참조하십시오.</p> <p>중요: 소스 리포지토리의 메인 브랜치에 변경 사항이 커밋되면 테스트 파이프라인이 자동으로 활성화됩니다.</p>	<p>DevOps 엔지니어</p>

작업	설명	필요한 기술
보고서 출력을 검증합니다.	<ol style="list-style-type: none"> CodePipeline 콘솔의 왼쪽 탐색 창에서 Build를 선택합니다. 그런 다음 보고서 기록을 선택합니다. 파이프라인이 생성하는 tfsec 및 checkov 스캔 보고서를 검토하세요. 이러한 보고서는 시각화와 그래픽 표현을 통해 문제를 식별하는데 도움이 될 수 있습니다. <p>참고: <project_name>-validate CodeBuild 프로젝트는 “validate” 단계 중에 코드에 대한 취약성 보고서를 생성합니다.</p>	DevOps 엔지니어

리소스 정리

작업	설명	필요한 기술
파이프라인 및 관련 리소스를 정리합니다.	<p>AWS 계정에서 테스트 리소스를 삭제하려면 복제된 리포지토리의 루트 폴더에서 다음 명령을 실행하세요.</p> <pre>terraform destroy --var-file=terraform.tfvars</pre>	DevOps 엔지니어

문제 해결

문제	Solution
“apply”스태이지 중에 AccessDenied 오류가 발생합니다.	<ol style="list-style-type: none"> “apply”스태이지와 관련된 CodeBuild 프로젝트의 실행 로그를 검토하여 누락된 IAM 권한이 있는지 확인하십시오. 자세한 내용은 AWS CodeBuild 사용 설명서의 CodeBuild AWS에서의 빌드 세부 정보 보기를 참조하십시오. 코드 편집기에서 복제된 리포지토리의 modules 폴더를 엽니다. 그런 다음 해당 iam-role 폴더로 이동하여 해당 폴더에 있는 main.tf 파일을 엽니다. codepipeline_policy 설명서에 AWS 계정의 리소스를 프로비저닝하는 데 필요한 IAM 정책을 추가합니다.

관련 리소스

- [모듈 블록](#)(Terraform 설명서)
- [CI/CD를 사용하여 Terraform으로 AWS 보안 서비스를 배포하고 구성하는 방법](#) (AWS 블로그 게시물)
- [서비스 연결 역할 사용](#) (IAM 설명서)
- [파이프라인 생성](#) (AWS CLI 설명서)
- [Amazon S3에 저장된 아티팩트에 대한 서버 측 암호화 구성](#) (CodePipelineAWS 설명서)
CodePipeline
- AWS용 [할당량](#) (CodeBuildAWS CodeBuild 설명서)
- [AWS에서의 데이터 보호 CodePipeline](#) (AWS CodePipeline 설명서)

추가 정보

Terraform 모듈 사용자 지정

다음 사항은 이 패턴에 사용되는 사용자 지정 Terraform 모듈 목록입니다.

- `codebuild_terraform`파이프라인의 각 단계를 구성하는 CodeBuild 프로젝트를 생성합니다.
- `codecommit_infrastructure_source_repo`소스 CodeCommit 리포지토리를 캡처하고 생성합니다.
- `codepipeline_iam_role`은 파이프라인에 필요한 IAM 역할을 생성합니다.
- `codepipeline_kms`는 Amazon S3 객체 암호화 및 복호화에 필요한 AWS KMS 키를 생성합니다.
- `codepipeline_terraform`소스 CodeCommit 리포지토리의 테스트 파이프라인을 생성합니다.
- `s3_artifacts_bucket`은 Amazon S3 버킷을 생성하여 파이프라인 아티팩트를 관리합니다.

빌드 사양 파일

다음 사항은 이 패턴이 각 파이프라인 단계를 실행하는 데 사용하는 빌드 사양(buildspec) 파일 목록입니다.

- `buildspec_validate.yml`은 "validate" 단계를 실행합니다.
- `buildspec_plan.yml`은 "plan" 단계를 실행합니다.
- `buildspec_apply.yml`은 "apply" 단계를 실행합니다.
- `buildspec_destroy.yml`은 "destroy" 단계를 실행합니다.

빌드 사양 파일 변수

각 buildspec 파일은 다음 변수를 사용하여 다양한 빌드별 설정을 활성화합니다.

변수	기본값	설명
<code>CODE_SRC_DIR</code>	<code>"."</code>	소스 CodeCommit 디렉토리를 정의합니다.
<code>TF_VERSION</code>	<code>"1.3.7"</code>	빌드 환경을 위한 Terraform 버전 정의

`buildspec_validate.yml` 파일은 또한 다음과 같은 변수를 지원하여 다양한 빌드별 설정을 활성화합니다.

변수	기본값	설명
SCRIPT_DIR	"/templates/scripts"	스크립트 디렉터리 정의
ENVIRONMENT	"dev"	환경 이름 정의
SKIPVALIDATIONFAILURE	"Y"	실패 시 검증을 생략합니다.
ENABLE_TFVALIDATE	"Y"	Terraform 검증 활성화
ENABLE_TFFORMAT	"Y"	Terraform 형식 활성화
ENABLE_TFCHECKOV	"Y"	checkov 스캔 활성화
ENABLE_TFSEC	"Y"	tfsec 스캔 활성화
TFSEC_VERSION	"v1.28.1"	tfsec 버전 정의

패턴 더 보기

- [PrivateLink AWS와 Network Load Balancer를 사용하여 Amazon EKS에서 비공개로 컨테이너 애플리케이션에 액세스합니다.](#)
- [한 AWS 계정의 AWS CodeCommit 리포지토리를 다른 계정의 SageMaker Studio와 연결](#)
- [AWS Systems Manager를 사용하여 Windows 레지스트리 항목의 추가 또는 업데이트 자동화](#)
- [이상 탐지를 위한 Amazon Lookout for Vision 교육 및 배포를 자동화합니다.](#)
- [AWS Batch를 사용하여 Amazon RDS for PostgreSQL DB 인스턴스 백업 자동화](#)
- [aws-nuke를 사용하여 AWS 리소스 자동 삭제](#)
- [AWS SAM을 사용하여 중첩된 애플리케이션 자동 배포](#)
- [CI/CD 파이프라인을 사용하여 Amazon EKS에서 노드 종료 핸들러 배포 자동화](#)
- [Amazon MQ에서 RabbitMQ 구성의 자동화](#)
- [AWS를 사용하여 AppStream 2.0 리소스 생성을 자동화합니다. CloudFormation](#)
- [AWS 계정 전반에 걸쳐 Amazon RDS 인스턴스의 복제를 자동화](#)
- [CI/CD 파이프라인을 사용하여 Amazon EKS에 Java 애플리케이션 자동 구축 및 배포](#)
- [Python 애플리케이션을 사용하여 Amazon DynamoDB용 PynamoDB 모델 및 CRUD 함수를 자동으로 생성합니다.](#)
- [IAM 액세스 분석기 및 AWS 매크로를 사용하여 CodePipeline AWS 계정의 IAM 정책 및 역할을 자동으로 검증하고 배포합니다. CloudFormation](#)
- [AWS Cloud의 Stomasys Charon-SSP 에뮬레이터에서 Sun SPARC 서버 백업하기](#)
- [AWS DataOps 개발 키트를 사용하여 Google 애널리틱스 데이터를 수집, 변환 및 분석할 수 있는 데이터 파이프라인을 구축하십시오.](#)
- [Amazon EC2 Auto Scaling 및 Systems Manager를 사용하여 Micro Focus Enterprise Server PAC 구축하기](#)
- [EC2 Image Builder와 Terraform을 사용하여 강화된 컨테이너 이미지를 위한 파이프라인을 구축하십시오.](#)
- [SageMaker Amazon과 Azure를 사용하여 MLOps 워크플로를 구축하세요. DevOps](#)
- [서버리스 접근 방식을 사용하여 AWS 서비스를 함께 연결](#)
- [CloudWatch NLog를 사용하여 Amazon Logs에서.NET 애플리케이션에 대한 로깅을 구성합니다.](#)
- [AWS 리포지토리에서 최신 AWS Amplify 웹 애플리케이션을 지속적으로 배포하십시오. CodeCommit](#)

- [AWS Step Functions에서 사용자 지정 Docker 컨테이너 이미지를 SageMaker 생성하여 모델 교육에 사용하십시오.](#)
- [AWS를 지원하지 않는 AWS 지역에 파이프라인을 생성하십시오. CodePipeline](#)
- [Amazon CloudWatch 이상 탐지를 사용하여 사용자 지정 지표에 대한 경고 생성](#)
- [여러 코드 결과물에서 보안 문제를 동시에 탐지하는 파이프라인을 배포하십시오.](#)
- [AWS 클라우드에서 인프라를 코드로 사용하여 서버리스 데이터 레이크 배포와 관리](#)
- [Amazon EKS와 Amazon S3의 차트 Helm 리포지토리를 사용하여 Kubernetes 리소스 및 패키지 배포](#)
- [AWS CDK를 사용하여 다중 스택 애플리케이션을 배포할 수 있습니다. TypeScript](#)
- [Terraform을 사용하여 AWS WAF의 보안 자동화 솔루션 배포](#)
- [RAG 및 프롬프트를 사용하여 고급 제너레이티브 AI 채팅 기반 어시스턴트 개발 ReAct](#)
- [AWS CloudFormation 템플릿을 사용하여 Amazon을 GuardDuty 조건부로 활성화합니다.](#)
- [Amazon Personalize를 사용하여 개인화되고 순위가 다시 매겨진 추천 생성](#)
- [AWS KMS 키의 키 상태가 변경될 때 Amazon SNS 알림 받기](#)
- [AWS CDK를 사용하여 여러 AWS 지역, 계정 및 DevOps OU에서 Amazon Guru를 활성화하여 운영 성능을 개선하십시오.](#)
- [쿠버네티스를 사용하여 Amazon EKS 작업자 노드에 SSM 에이전트를 설치합니다. DaemonSet](#)
- [Stonebranch 유니버설 컨트롤러를 AWS Mainframe Modernization과 통합](#)
- [메인프레임 현대화: 마이크로 포커스를 DevOps 활용한 AWS 기반](#)
- [AWS를 사용하여 AWS IAM ID 센터 권한 세트를 코드로 관리합니다. CodePipeline](#)
- [AWS CDK로 Amazon ECS Anywhere를 설정하여 온프레미스 컨테이너 애플리케이션을 관리](#)
- [DNS 레코드를 Amazon Route 53 프라이빗 호스팅 영역으로 대량 마이그레이션합니다.](#)
- [AWS 개발자 도구를 SageMaker 사용하여 ML, 빌드, 교육 및 워크로드를 Amazon으로 마이그레이션, 배포](#)
- [여러 AWS 계정에 공유된 Amazon Machine Image의 사용을 모니터링](#)
- [AWS App2Container 생성 도커 이미지 최적화](#)
- [AWS Step Functions를 사용하여 검증, 변환 및 파티셔닝을 통해 ETL 파이프라인 오케스트레이션](#)
- [비 워크로드 서브넷을 위한 다중 계정 VPC 설계에서 라우팅 가능한 IP 공간 보존](#)
- [코드 리포지토리를 사용하여 AWS Service Catalog에서 Terraform 제품을 프로비저닝합니다.](#)
- [필터링된 Amazon ECR 컨테이너 이미지를 계정 또는 리전 전반적으로 복제](#)
- [컨테이너를 다시 시작하지 않고 데이터베이스 보안 인증 교체](#)

- [AWS Step Functions에서 AWS Systems Manager Automation 작업을 동기적으로 실행](#)
- [AWS CDK를 사용하여 Amazon ECS Anywhere에서 하이브리드 워크로드를 위한 CI/CD 파이프라인을 설정하고 GitLab](#)
- [Amazon FSx를 사용하여 SQL Server Always On FCI용 다중 AZ 인프라 설정](#)
- [AWS를 사용하여 Amazon EC2에 UiPath RPA 봇을 자동으로 설정합니다. CloudFormation](#)
- [C# 및 AWS CDK를 사용한 사일로 모델을 위한 SaaS 아키텍처의 테넌트 온보딩](#)
- [Terraform을 사용하여 조직에서 GuardDuty Amazon을 자동으로 활성화하십시오.](#)
- [Account Factory에 대한 테라폼\(AFT\) 코드를 로컬에서 검증합니다.](#)
- [Flask와 Elastic Beanstalk를 사용하여 AI/ML 모델 결과를 시각화합니다.](#)

최종 사용자 컴퓨팅

주제

- [AWS를 사용하여 AppStream 2.0 리소스 생성을 자동화합니다. CloudFormation](#)
- [더 많은 패턴](#)

AWS를 사용하여 AppStream 2.0 리소스 생성을 자동화합니다.

CloudFormation

작성자: Ram Kandaswamy(AWS), Raheem Jiwani(AWS)

환경: 프로덕션

기술: 최종 사용자 컴퓨팅, 클라우드는 네이티브, 비용 관리, SaaS DevOps

워크로드: Microsoft

AWS 서비스: 아마존 AppStream 2.0, AWS CloudFormation

요약

이 패턴은 AWS CloudFormation 템플릿을 사용하여 Amazon Web Services (AWS) 클라우드에서 Amazon AppStream 2.0 리소스 생성을 자동화하는 코드 샘플 및 단계를 제공합니다. 이 패턴은 AWS CloudFormation 스택을 사용하여 이미지 빌더, 이미지, 플릿 인스턴스 및 스택을 비롯한 AppStream 2.0 애플리케이션 리소스의 생성을 자동화하는 방법을 보여줍니다. 데스크톱 또는 애플리케이션 AppStream 전송 모드를 사용하여 HTML5 호환 브라우저에서 최종 사용자에게 2.0 애플리케이션을 스트리밍할 수 있습니다.

사전 조건 및 제한 사항

사전 조건

- 활성 상태의 AWS 계정
- 2.0 이용 약관 AppStream 동의
- [스택, 플릿, 이미지 빌더와 같은 AppStream 리소스에 대한 기본 지식](#)

제한 사항

- 2.0 인스턴스가 생성된 후에는 AppStream 2.0 인스턴스에 연결된 AWS ID 및 액세스 관리 (IAM) 역할을 수정할 수 없습니다.

- 이미지 빌더가 생성된 후에는 AppStream 2.0 이미지 빌더 인스턴스의 속성 (예: 서브넷 또는 보안 그룹) 을 수정할 수 없습니다.

아키텍처

다음 다이어그램은 AWS CloudFormation 템플릿을 사용하여 AppStream 2.0 리소스 생성을 자동화하는 방법을 보여줍니다.

이 다이어그램은 다음 워크플로를 보여줍니다.

1. 이 패턴의 추가 정보 섹션에 있는 YAML 코드를 기반으로 AWS CloudFormation 템플릿을 생성합니다.
2. AWS CloudFormation 템플릿은 AWS CloudFormation 테스트 스택을 생성합니다.
 - a. (선택 사항) AppStream 2.0을 사용하여 이미지 빌더 인스턴스를 생성합니다.
 - b. (선택 사항) 사용자 지정 소프트웨어를 사용하여 Windows 이미지를 생성합니다.
3. AWS CloudFormation 스택은 AppStream 2.0 플릿 인스턴스와 스택을 생성합니다.
4. HTML5 호환 브라우저에서 최종 사용자에게 AppStream 2.0 리소스를 배포합니다.

기술 스택

- 아마존 AppStream 2.0
- AWS CloudFormation

도구

- [Amazon AppStream 2.0](#) — Amazon AppStream 2.0은 어디서나 데스크톱 애플리케이션에 즉시 액세스할 수 있는 완전 관리형 애플리케이션 스트리밍 서비스입니다. AppStream 2.0은 애플리케이션을 호스팅하고 실행하는 데 필요한 AWS 리소스를 관리하고, 자동으로 확장되며, 필요에 따라 사용자에게 액세스를 제공합니다.
- [AWS CloudFormation](#) — AWS는 AWS CloudFormation 리소스를 모델링 및 설정하고, 빠르고 일관되게 프로비저닝하고, 수명 주기 전반에 걸쳐 관리할 수 있도록 지원합니다. 템플릿을 사용하여 리소스와 해당 종속성을 설명하고, 리소스를 개별적으로 관리하는 대신 스택으로 함께 시작하고 구성할 수 있습니다. 여러 AWS 계정 및 AWS 리전에서 스택을 관리하고 프로비저닝할 수 있습니다.

에픽

(선택 사항) AppStream 2.0 이미지 생성

작업	설명	필요한 기술
<p>맞춤형 소프트웨어를 설치하고 이미지를 생성하세요.</p>	<ol style="list-style-type: none"> 1. 사용자에게 배포하려는 AppStream 2.0 애플리케이션을 설치합니다. 2. Photon 이미지 생성 에이전트 또는 PowerShell 스크립트를 사용하여 사용자 지정 소프트웨어를 위한 새 Windows 이미지를 생성하십시오. <p>참고: 이미지를 추가로 잠그려면 Windows AppLocker 기능을 사용하는 것을 고려해 보십시오.</p>	<p>AWS DevOps, 클라우드 아키텍트</p>

AWS CloudFormation 템플릿 배포

작업	설명	필요한 기술
<p>AWS CloudFormation 템플릿을 업데이트합니다.</p>	<ol style="list-style-type: none"> 1. 이 패턴의 추가 정보 섹션에 있는 코드를 YAML 파일로 저장합니다. 2. 사용자 환경의 파라미터에 필요한 값으로 YAML 파일을 업데이트합니다. 	<p>AWS 시스템 관리자, 클라우드 관리자, 클라우드 아키텍트, 일반 AWS, AWS 관리자</p>
<p>템플릿을 사용하여 AWS CloudFormation 스택을 생성합니다.</p>	<ol style="list-style-type: none"> 1. AWS 관리 콘솔에 로그인하고 AWS CloudFormation 콘솔을 엽니다. 	<p>앱 소유자, AWS 시스템 관리자, Windows 엔지니어</p>

작업	설명	필요한 기술
	<ol style="list-style-type: none"> 2. 탐색 창에서 스택을 선택합니다. 3. 스택 생성을 선택한 다음 새 리소스 사용(표준)을 선택합니다. 4. 사전 조건 - 템플릿 준비 섹션에서 템플릿 준비 완료를 선택합니다. 5. 템플릿 지정 섹션에서 템플릿 파일 업로드를 선택합니다. 6. 파일 선택을 선택한 다음 업데이트된 AWS CloudFormation 템플릿을 선택합니다. 7. 마법사의 나머지 단계를 완료하여 스택을 생성합니다. 	

관련 리소스

참조

- [Amazon AppStream 2.0 시작하기: 샘플 애플리케이션을 사용하여 설정](#)
- [AppStream 2.0 플릿 및 스택 생성](#)

튜토리얼 및 동영상

- [아마존 AppStream 2.0 사용자 워크플로](#)
- [레거시 Windows Forms 앱을 Amazon AppStream 2.0으로 마이그레이션하는 방법](#)
- [AWS re:Invent 2018: Amazon AppStream 2.0 \(BAP201\) 을 사용하여 데스크톱 애플리케이션을 안전하게 제공](#)

추가 정보

다음 코드는 AppStream 2.0 리소스를 자동으로 생성할 수 있는 AWS CloudFormation 템플릿의 예입니다.

```
AWSTemplateFormatVersion: 2010-09-09
Parameters:
  SubnetIds:
    Type: 'List<AWS::EC2::Subnet::Id>'
  testSecurityGroup:
    Type: 'AWS::EC2::SecurityGroup::Id'
  ImageName:
    Type: String
Resources:

  AppStreamFleet:
    Type: 'AWS::AppStream::Fleet'
    Properties:
      ComputeCapacity:
        DesiredInstances: 5
      InstanceType: stream.standard.medium
      Name: appstream-test-fleet
      DisconnectTimeoutInSeconds: 1200
      FleetType: ON_DEMAND
      IdleDisconnectTimeoutInSeconds: 1200
      ImageName: !Ref ImageName
      MaxUserDurationInSeconds: 345600
      VpcConfig:
        SecurityGroupIds:
          - !Ref testSecurityGroup
        SubnetIds: !Ref SubnetIds

  AppStreamStack:
    Type: 'AWS::AppStream::Stack'
    Properties:
      Description: AppStream stack for test
      DisplayName: AppStream test Stack
      Name: appstream-test-stack
      StorageConnectors:
        - ConnectorType: HOMEFOLDERS
      UserSettings:
        - Action: CLIPBOARD_COPY_FROM_LOCAL_DEVICE
          Permission: ENABLED
        - Action: CLIPBOARD_COPY_TO_LOCAL_DEVICE
```

```
Permission: ENABLED
- Action: FILE_DOWNLOAD
Permission: ENABLED
- Action: PRINTING_TO_LOCAL_DEVICE
Permission: ENABLED
```

AppStreamFleetAssociation:

```
Type: 'AWS::AppStream::StackFleetAssociation'
```

Properties:

```
FleetName: appstream-test-fleet
StackName: appstream-test-stack
```

DependsOn:

```
- AppStreamFleet
- AppStreamStack
```

더 많은 패턴

- [Session Manager를 사용하여 Amazon EC2 인스턴스에 연결](#)
- [Amazon Connect 콜센터의 에이전트 워크스테이션의 통화 품질을 개선합니다.](#)
- [AWS Step Functions에서 AWS Systems Manager Automation 작업을 동기적으로 실행](#)

고성능 컴퓨팅

주제

- [AWS용 Grafana 모니터링 대시보드 설정 ParallelCluster](#)
- [NICE EnginFrame 및 NICE DCV 세션 관리자를 사용하여 자동 스케일링 가상 데스크톱 인프라 \(VDI\) 를 설정합니다.](#)

AWS용 Grafana 모니터링 대시보드 설정 ParallelCluster

작성자: Dario La Porta (AWS) 및 William Lu (AWS)

코드 리포지토리: parallelcuster-monitoring-dashboard	환경: PoC 또는 파일럿	기술: 고성능 컴퓨팅, 분석, 관리 및 거버넌스
워크로드: 오픈 소스	AWS 서비스: AWS ParallelCluster	

요약

AWS는 고성능 컴퓨팅 (HPC) 클러스터를 배포하고 관리할 수 있도록 ParallelCluster 지원합니다. AWS Batch 및 Slurm 오픈 소스 작업 스케줄러를 지원합니다. ParallelCluster AWS는 로깅 및 CloudWatch 메트릭을 위해 Amazon과 통합되어 있지만 워크로드에 대한 모니터링 대시보드를 제공하지는 않습니다.

AWS용 [Grafana 대시보드 GitHub \(\)](#) 는 [ParallelCluster AWS용](#) 모니터링 대시보드입니다.

ParallelCluster 운영 체제(OS) 수준에서 작업 스케줄러 인사이트와 상세한 모니터링 지표를 제공합니다. 이 솔루션에 포함된 대시보드에 대한 자세한 내용은 리포지토리의 [예제 대시보드를](#) 참조하십시오. GitHub 이러한 메트릭은 HPC 워크로드와 성능을 더 잘 이해하는 데 도움이 됩니다. 하지만 대시보드 코드는 최신 버전의 AWS ParallelCluster 또는 솔루션에서 사용되는 오픈 소스 패키지에 대해서는 업데이트되지 않습니다. 이 패턴은 다음과 같은 장점을 제공하도록 솔루션을 향상시킵니다.

- AWS ParallelCluster v3를 지원합니다.
- Prometheus, Grafana, Prometheus Slurm Exporter, NVIDIA DCGM-Exporter를 포함한 최신 버전의 오픈 소스 패키지를 사용합니다.
- Slurm 작업이 사용하는 CPU 코어 및 GPU 수를 증가시킵니다.
- 작업 모니터링 대시보드 추가
- 그래픽 처리 장치(GPU)가 4개 또는 8개인 노드의 GPU 노드 모니터링 대시보드를 개선합니다.

이 버전의 향상된 솔루션은 AWS 고객의 HPC 프로덕션 환경에서 구현 및 검증되었습니다.

사전 조건 및 제한 사항

사전 조건

- [AWS ParallelCluster CLI](#), 설치 및 구성.
- AWS에 지원되는 [네트워크 구성 ParallelCluster](#). 이 패턴은 퍼블릭 서브넷, 프라이빗 서브넷, 인터넷 게이트웨이 및 NAT 게이트웨이가 필요한 두 개의 서브넷 구성을 ParallelCluster 사용하는 AWS를 사용합니다.
- 모든 AWS ParallelCluster 클러스터 노드는 인터넷에 액세스할 수 있어야 합니다. 이는 설치 스크립트가 오픈 소스 소프트웨어 및 Docker 이미지를 다운로드할 수 있도록 하기 위해 필요합니다.
- Amazon Elastic Compute Cloud(Amazon EC2)의 [키 쌍](#) 이 키 쌍이 있는 리소스는 헤드 노드에 대한 SSH(Secure Shell) 액세스 권한을 가집니다.

제한 사항

- 이 패턴은 Ubuntu 20.04 LTS를 지원하도록 설계되었습니다. 다른 버전의 Ubuntu를 사용하거나 Amazon Linux 또는 CentOS를 사용하는 경우 이 솔루션과 함께 제공되는 스크립트를 수정해야 합니다. 이러한 수정은 이 패턴에 포함되지 않습니다.

제품 버전

- Ubuntu 20.04 LTS
- ParallelCluster 3.X

청구 및 비용 고려 사항

- 이 패턴으로 배포된 솔루션에는 프리 티어가 적용되지 않습니다. Amazon EC2, Amazon FSx for Lustre, Amazon VPC의 NAT 게이트웨이, Amazon Route 53에는 요금이 부과됩니다.

아키텍처

대상 아키텍처

다음 다이어그램은 사용자가 헤드 노드에서 ParallelCluster AWS의 모니터링 대시보드에 액세스하는 방법을 보여줍니다. 헤드 노드는 NICE DCV, Prometheus, Grafana, Prometheus Slurm Exporter, Prometheus Node Exporter, NGINX 오픈 소스를 운영합니다. 컴퓨팅 노드는 Prometheus Node Exporter를 실행하며, 노드에 GPU가 포함된 경우 NVIDIA DCGM-Exporter도 실행합니다. 헤드 노드는 컴퓨팅 노드에서 정보를 검색하고 Grafana 대시보드에 해당 데이터를 표시합니다.

대부분의 경우 작업 스케줄러에 상당한 양의 CPU나 메모리가 필요하지 않기 때문에 헤드 노드의 부하가 많지 않습니다. 사용자는 포트 443에서 SSL을 사용하여 헤드 노드의 대시보드에 액세스합니다.

승인된 모든 뷰어는 모니터링 대시보드를 익명으로 볼 수 있습니다. Grafana 관리자만 대시보드를 수정할 수 있습니다. `aws-parallelcluster-monitoring/docker-compose/docker-compose.head.yml` 파일에서 Grafana 관리자의 비밀번호를 구성합니다.

도구

서비스

- [NICE DCV](#)는 다양한 네트워크 조건에서 모든 클라우드 또는 데이터 센터에서 모든 디바이스로 원격 데스크톱 및 애플리케이션 스트리밍을 제공할 수 있는 고성능 원격 디스플레이 프로토콜입니다.
- [AWS](#)는 고성능 컴퓨팅 (HPC) 클러스터를 배포하고 관리할 수 있도록 ParallelCluster 지원합니다. AWS Batch 및 Slurm 오픈 소스 작업 스케줄러를 지원합니다.
- [Amazon Simple Storage Service\(S3\)](#)는 원하는 양의 데이터를 저장, 보호 및 검색하는 데 도움이 되는 클라우드 기반 객체 스토리지 서비스입니다.
- [Amazon Virtual Private Cloud\(VPC\)](#)를 이용하면 사용자가 정의한 가상 네트워크로 AWS 리소스를 시작할 수 있습니다.

기타 도구

- [Docker](#)는 운영 체제 수준의 가상화를 사용하여 컨테이너에 소프트웨어를 제공하는 서비스형 플랫폼 (PaaS) 제품 세트입니다.
- [Grafana](#)는 지표, 로그 및 추적을 쿼리, 시각화, 경고 및 탐색하는 데 도움이 되는 오픈 소스 소프트웨어입니다.
- [NGINX 오픈 소스](#)는 오픈 소스 웹 서버이자 리버스 프록시입니다.
- [NVIDIA 데이터 센터 GPU 관리자\(DCGM\)](#)는 클러스터 환경에서 NVIDIA 데이터 센터 그래픽 처리 장치(GPU)를 관리하고 모니터링하기 위한 도구 모음입니다. 이 패턴에서는 Prometheus에서 GPU 지표를 내보내는 데 도움이 되는 [DCGM-Exporter](#)를 사용합니다.
- [Prometheus](#)는 지표를 관련 키값 쌍(레이블이라고 함)이 있는 시계열 데이터로 수집하고 저장하는 오픈 소스 시스템 모니터링 툴킷입니다. 이 패턴에서는 [Prometheus Slurm Exporter](#)를 사용하여 지표를 수집 및 내보내고 [Prometheus Node Exporter](#)를 사용하여 컴퓨팅 노드에서 지표를 내보냅니다.
- [Ubuntu](#)는 엔터프라이즈 서버, 데스크톱, 클라우드 환경 및 IoT를 위해 설계된 오픈 소스 Linux 기반 운영 체제입니다.

코드 리포지토리

이 패턴의 코드는 GitHub [pcluster-monitoring-dashboard](https://github.com/aws-samples/parallelcluster-monitoring-dashboard) 리포지토리에서 사용할 수 있습니다.

에픽

필수 리소스 만들기

작업	설명	필요한 기술
S3 버킷을 생성합니다.	Amazon S3 버킷을 생성합니다. 이 버킷을 사용하여 구성 스크립트를 저장합니다. 지침은 Amazon S3 설명서의 버킷 생성 을 참조하세요.	일반 AWS
리포지토리를 복제합니다.	다음 명령을 실행하여 GitHub pcluster-monitoring-dashboa rd 리포지토리를 복제합니다. <pre>git clone https://github.com/aws-samples/parallelcluster-monitoring-dashboard.git</pre>	DevOps 엔지니어
관리 비밀번호를 생성합니다.	1. aws-parallelcluster-monitoring 폴더를 선택하고 docker-compose 폴더를 선택한 다음, docker-compose.head.yml 파일을 엽니다. 2. GF_SECURITY_ADMIN_PASSWORD 변수에서 Grafana4PC! 를 원하는 암호로 바꿉니다. 이것은 Grafana 계정을 관리하는 데	Linux Shell 스크립팅

작업	설명	필요한 기술
	<p>사용하는 관리자 비밀번호입니다.</p> <p>3. docker-compose.head.yml 파일을 저장하고 닫습니다.</p>	
<p>필요한 파일을 S3 버킷에 복사합니다.</p>	<p>post_install.sh 스크립트와 aws-parallelcluster-monitoring 폴더를 생성한 S3 버킷에 복사합니다. 이에 관한 지침은 Amazon S3 설명서의 객체 업로드를 참조하세요.</p>	<p>일반 AWS</p>
<p>헤드 노드용 추가 보안 그룹을 구성합니다.</p>	<ol style="list-style-type: none"> 1. 헤드 노드용 보안 그룹을 생성합니다. 이 보안 그룹은 헤드 노드의 모니터링 대시보드의 인바운드 트래픽을 허용합니다. 지침은 Amazon VPC 설명서의 NAT 게이트웨이 생성을 참조하세요. 2. 인바운드 규칙을 보안 그룹에 추가합니다. 지침은 Amazon VPC 설명서의 NAT 게이트웨이 생성을 참조하세요. 규칙에 다음 파라미터를 사용합니다. <ul style="list-style-type: none"> • 유형 - HTTPS • 프로토콜 - TCP • 포트 범위 - 443 • 소스 - IP 주소를 입력합니다. • 설명 - 사용자가 모니터링 대시보드에 액세스할 수 있도록 허용 	<p>AWS 관리자</p>

작업	설명	필요한 기술
<p>헤드 노드의 IAM 정책을 구성합니다.</p>	<p>헤드 노드에 대한 자격 증명 기반 정책을 생성합니다. 이 정책을 통해 노드는 Amazon에서 지표 데이터를 검색할 수 CloudWatch 있습니다. GitHub 리포지토리에는 예제 정책 이 포함되어 있습니다. 지침은 AWS Identity 및 Access Management(IAM) 설명서의 IAM 정책 생성을 참고하십시오.</p>	<p>AWS 관리자</p>
<p>컴퓨팅 노드의 IAM 정책을 구성합니다.</p>	<p>컴퓨팅 노드에 대한 자격 증명 기반 정책을 생성합니다. 이 정책을 통해 노드는 작업 ID 및 작업 소유자가 포함된 태그를 생성할 수 있습니다. GitHub 리포지토리에는 예제 정책이 포함되어 있습니다. 지침은 AWS Identity 및 Access Management(IAM) 설명서의 IAM 정책 생성을 참조하세요.</p> <p>제공된 예제 파일을 사용할 경우 아래의 값을 대체합니다.</p> <ul style="list-style-type: none"> • <REGION> – 클러스터가 호스팅되는 AWS 리전 • <ACCOUNT_ID> – AWS 계정 ID 	<p>AWS 관리자</p>

클러스터를 생성합니다.

작업	설명	필요한 기술
<p>제공된 클러스터 템플릿 파일을 수정합니다.</p>	<p>AWS ParallelCluster 클러스터를 생성합니다. 제공된 cluster.yaml CloudFormation AWS 템플릿 파일을 출발점으로 사용하여 클러스터를 생성합니다. 제공된 템플릿에서 다음 값을 바꾸십시오.</p> <ul style="list-style-type: none"> • <REGION> – 클러스터가 호스팅되는 AWS 리전입니다. • <HEADNODE_SUBNET> – VPC의 공개 서브넷 • <ADDITIONAL_HEAD_NODE_SG> – 헤드 노드용으로 만든 보안 그룹 이름 • <KEY_NAME> – 기존의 Amazon EC2 키 쌍의 이름을 입력합니다. 이 키 쌍이 있는 리소스는 헤드 노드에 대한 SSH(Secure Shell) 액세스 권한을 가집니다. • <ALLOWED_IPS> – 헤드 노드에 SSH 연결을 허용하는 CIDR 형식의 IP 주소 범위를 입력합니다. • <ADDITIONAL_HEAD_NODE_POLICY> – 헤드 노드에 대해 생성한 IAM 정책의 이름을 입력합니다. • <BUCKET_NAME> – 생성한 S3 버킷의 이름을 입력합니다. 	<p>AWS 관리자</p>

작업	설명	필요한 기술
	<ul style="list-style-type: none"> • <COMPUTE_SUBNET> – VPC에 있는 개인 서브넷의 이름을 입력합니다. • <ADDITIONAL_COMPUTE_NODE_POLICY> – 컴퓨팅 노드에 대해 생성한 IAM 정책의 이름을 입력합니다. 	
클러스터를 생성합니다.	<p>AWS ParallelCluster CLI에서 다음 명령을 입력합니다. 그러면 CloudFormation 템플릿이 배포되고 클러스터가 생성됩니다. 이 명령에 대한 자세한 내용은 AWS 설명서의 pcluster 생성-클러스터를 참조하십시오.</p> <p>ParallelCluster</p> <pre>pcluster create-cluster -n <cluster_name> -c cluster.yaml</pre>	AWS 관리자
클러스터 생성을 모니터링하십시오.	<p>다음 명령을 입력하여 클러스터 생성을 모니터링합니다. 이 명령에 대한 자세한 내용은 AWS 설명서의 pcluster 설명-클러스터를 참조하십시오.</p> <p>ParallelCluster</p> <pre>pcluster describe-cluster -n <cluster_name></pre>	AWS 관리자

Grafana 대시보드 사용

작업	설명	필요한 기술
Grafana 포털에 접속합니다.	<p>1. 다음 명령을 입력하여 헤드 노드의 공개 IP 주소를 검색합니다.</p> <pre data-bbox="630 499 1029 737">pcluster describe-cluster -n <cluster_name> --query headNode.publicIpAddress</pre> <p>2. 웹 브라우저에서 Grafana 대시보드에 액세스하려면 다음 URL로 이동합니다.</p> <p><code>https://<head_node_public_ip_address></code></p> <p>3. Grafana 프론트 페이지의 왼쪽 메뉴에서 4사각형 대시보드 아이콘을 선택한 다음 일반을 선택합니다. 그러면 구성된 대시보드 목록이 표시됩니다. Grafana에서 다음 대시보드를 사용할 수 있습니다.</p> <ul data-bbox="630 1423 1029 1854" style="list-style-type: none"> • 클러스터 비용 - 클러스터 비용에 대한 정보가 있습니다. • 클러스터 비용 - 클러스터 비용에 대한 정보가 있습니다. • 컴퓨팅 노드 세부 정보 - 컴퓨팅 노드의 사용 통계에 대한 정보가 있습니다. 	AWS 관리자

작업	설명	필요한 기술
	<ul style="list-style-type: none"> • 컴퓨팅 노드 목록 - 클러스터의 컴퓨팅 노드 목록이 있습니다. • GPU 노드 - GPU 노드의 사용 통계에 대한 정보가 있습니다. • 작업 세부 정보 - 작업 리소스 사용률에 대한 정보가 있습니다. • 헤드 노드 세부 정보 - 헤드 노드의 사용 통계에 대한 정보 포함 • ParallelCluster 요약 — 클러스터 사용에 대한 정보가 들어 있습니다. 	

관련 비용이 발생하지 않도록 솔루션을 정리합니다.

작업	설명	필요한 기술
클러스터를 삭제합니다.	<p>클러스터를 삭제하려면 다음 명령을 입력합니다. 이 명령에 대한 자세한 내용은 AWS 설명서의 pcluster 삭제 클러스터를 참조하십시오. ParallelCluster</p> <pre data-bbox="592 1522 1031 1638">pcluster delete-cluster -n <cluster_name></pre>	AWS 관리자
IAM 정책을 삭제합니다.	헤드 노드와 컴퓨팅 노드에 대해 생성한 정책을 삭제합니다. 정책 삭제에 대한 자세한 내용	AWS 관리자

작업	설명	필요한 기술
	은 IAM 설명서의 IAM 정책 삭제 를 참조하세요.	
보안 그룹 및 규칙을 삭제합니다.	헤드 노드용으로 생성한 보안 그룹을 삭제합니다. 자세한 정보는 Amazon VPC 설명서의 보안 그룹 규칙 삭제 및 보안 그룹 삭제 를 참조하세요.	AWS 관리자
S3 버킷을 삭제합니다.	구성 스크립트를 저장하기 위해 생성한 S3 버킷을 삭제합니다. 자세한 내용은 Amazon S3 설명서의 버킷 삭제 를 참조하세요.	일반 AWS

문제 해결

문제	Solution
브라우저에서 헤드 노드에 액세스할 수 없습니다.	보안 그룹을 확인하고 인바운드 포트 443이 열려 있는지 확인합니다.
Grafana가 열리지 않습니다.	헤드 노드에서 <code>docker logs Grafana</code> 의 컨테이너 로그를 확인하십시오.
일부 지표에 데이터가 없습니다.	헤드 노드에서 모든 컨테이너의 컨테이너 로그를 확인하십시오.

관련 리소스

AWS 설명서

- [Amazon EC2에 대한 IAM 정책](#)

기타 AWS 리소스

- [AWS ParallelCluster](#)
- [AWS용 모니터링 대시보드 ParallelCluster](#) (AWS 블로그 게시물)

기타 리소스

- [Prometheus 모니터링 시스템](#)
- [Grafana](#)

NICE EnginFrame 및 NICE DCV 세션 관리자를 사용하여 자동 스케일링 가상 데스크톱 인프라 (VDI) 를 설정합니다.

작성자: Dario La Porta 및 Salvatore Maccarone(AWS)

코드 리포지토리: [elastic-vdi-infrastructure](#)

환경: PoC 또는 파일럿

기술: 고성능 컴퓨팅, 인프라

AWS 서비스: AWS CDK, AWS, 아마존 EC2 Auto CloudFormation Scaling, Elastic Load Balancing (ELB)

요약

NICE DCV는 다양한 네트워크 조건에서 모든 클라우드 또는 데이터 센터에서 모든 디바이스로 원격 데스크톱 및 애플리케이션을 스트리밍할 수 있는 고성능 원격 디스플레이 프로토콜입니다. NICE DCV와 Amazon Elastic Compute Cloud(Amazon EC2)를 사용하면 그래픽 집약적인 애플리케이션을 EC2 인스턴스에서 원격으로 실행하고 해당 사용자 인터페이스를 더 간단한 원격 클라이언트 시스템으로 스트리밍할 수 있습니다. 따라서 값비싼 전용 워크스테이션이 필요하지 않으며 클라우드와 클라이언트 시스템 간에 대량의 데이터를 전송할 필요가 없습니다.

이 패턴은 웹 기반 사용자 인터페이스를 통해 액세스할 수 있는 모든 기능을 갖춘 Auto Scaling Linux 및 Windows 가상 데스크톱 인프라(VDI)를 설정합니다. VDI 솔루션은 연구 및 개발(R&D) 사용자에게 그래픽 집약적인 분석 요청을 제출하고 결과를 원격으로 검토할 수 있는 접근성과 성능이 뛰어난 사용자 인터페이스를 제공합니다.

사전 조건 및 제한 사항

사전 조건

- 활성 상태의 AWS 계정
- 관리자 권한 및 액세스 키 세트.
- AWS Cloud Development Kit(AWS CDK) 툴킷 설치 및 구성. 자세한 내용은 [AWS CDK 설치](#)를 참조하세요.

- AWS 계정에 대한 AWS Command Line Interface(AWS CLI) 설치 및 구성. 자세한 내용은 [최신 버전의 AWS CLI 설치 또는 업데이트](#)를 참조하세요.
- Python 설치 및 구성. 자세한 내용은 [소스 릴리스](#)(Python 웹사이트)를 참조하세요.
- 하나 이상의 Virtual Private Cloud(VPC) 사용 가능.
- 두 개 이상의 탄력적 IP 주소 사용 가능. 기본 한도에 대한 자세한 내용은 [탄력적 IP 주소 한도](#)를 참조하세요.
- Linux EC2 인스턴스의 경우 Secure Shell(SSH) 키 페어를 설정합니다. 자세한 내용은 [키 페어 및 Linux 인스턴스](#)를 참조하세요.

제품 버전

- AWS CDK 버전 2.26.0 이상
- Python 버전 3.8 이상

아키텍처

대상 아키텍처

다음 그림은 VDI 솔루션의 다양한 구성 요소를 보여 줍니다. 사용자는 윈도우 및 리눅스용 Amazon EC2 Auto Scaling 그룹의 EnginFrame NICE DCV 인스턴스에 따라 NICE와 상호 작용하여 Amazon EC2 인스턴스를 시작합니다.

자동화 및 규모 조정

이 패턴에 포함된 코드는 사용자 지정 VPC, 퍼블릭 및 프라이빗 서브넷, 인터넷 게이트웨이, NAT 게이트웨이, Application Load Balancer, 보안 그룹, IAM 정책을 생성합니다. CloudFormation AWS는 리눅스 및 윈도우 NICE DCV 서버를 만드는 데에도 사용됩니다.

도구

서비스

- [AWS Cloud Development Kit\(AWS CDK\)](#)는 AWS 클라우드 인프라를 코드로 정의하고 프로비저닝하는 데 도움이 되는 소프트웨어 개발 프레임워크입니다.
- [AWS는 AWS CloudFormation](#) 리소스를 설정하고, 빠르고 일관되게 프로비저닝하고, AWS 계정 및 지역 전반의 수명 주기 전반에 걸쳐 리소스를 관리할 수 있도록 지원합니다.

- [NICE DCV](#)는 다양한 네트워크 조건에서 모든 클라우드 또는 데이터 센터에서 모든 디바이스로 원격 데스크톱 및 애플리케이션 스트리밍을 제공할 수 있는 고성능 원격 디스플레이 프로토콜입니다. 이 패턴에서는 고성능 컴퓨팅(HPC) 3D 그래픽을 원격으로 스트리밍하는 대역폭 효율적인 환경을 제공합니다.
- [NICE DCV Session Manager](#)를 사용하면 NICE DCV 서버의 플릿에서 NICE DCV 세션의 수명 주기를 생성하고 관리할 수 있습니다.
- [EnginFrameNICE](#)는 클라우드의 기술 및 과학 애플리케이션에 액세스할 수 있는 고급 프론트엔드 웹 인터페이스입니다.

코드 리포지토리

이 패턴의 코드는 NICE [EnginFrame](#) 및 NICE DCV 세션 관리자 저장소가 포함된 [Auto Scaling VDI 솔루션에서](#) 사용할 수 있습니다.

에픽

가상 데스크톱 인프라 배포

작업	설명	필요한 기술
리포지토리를 복제합니다.	코드가 들어 있는 리포지토리를 복제합니다. <pre>git clone https://github.com/aws-samples/elastic-vdi-infrastructure.git</pre>	클라우드 아키텍트
필수 AWS CDK 라이브러리를 설치합니다.	AWS CDK 라이브러리를 설치합니다. <pre>cd elastic-vdi-infrastructure python3 -m venv .venv source .venv/bin/activate pip3 install -r requirements.txt</pre>	클라우드 아키텍트

작업	설명	필요한 기술
<p>파라미터를 업데이트합니다.</p>	<ol style="list-style-type: none"> 원하는 텍스트 편집기에서 <code>app.py</code> 파일을 엽니다. 다음 필수 파라미터의 <code>CHANGE_ME</code> 값을 바꿉니다. <ul style="list-style-type: none"> <code>region</code> - 대상 AWS 리전입니다. 전체 목록은 AWS 리전을 참조하세요. <code>account</code> - 대상 AWS 계정의 ID입니다. 자세한 내용은 AWS 계정 ID 찾기를 참조하세요. <code>key_name</code> - Linux EC2 인스턴스에 액세스하는데 사용되는 키 페어입니다. (선택 사항) 다음 파라미터의 값을 수정하여 환경에 맞게 솔루션을 사용자 지정합니다. <ul style="list-style-type: none"> <code>ec2_type_enginframe</code> — 인스턴스 유형 EnginFrame <code>ec2_type_broker</code> - Session Manager Broker 인스턴스 유형 <code>ebs_enginframe_size</code> — 인스턴스의 아마존 엘라스틱 블록 스토어 (Amazon EBS) 볼륨 크기 EnginFrame <code>ebs_broker_size</code> - Session Manager Broker 	<p>클라우드 아키텍트</p>

작업	설명	필요한 기술
	<p>인스턴스의 EBS 볼륨 크기</p> <ul style="list-style-type: none"> • TagName and TagValue - 리소스의 청구 태그 • eadmin_uid — EnginFrame 관리자 (eadmin) 사용자의 고유 식별자 • linux_shared_storage_size - 기비바이트 (GiB) 단위의 OpenZFS 크기 • Shared_Storage_Linux - 공유 스토리지의 탑재 지점 • Enginframe_installer — 다운로드 링크 EnginFrame • Session_Manager_Broker_Installer - Session Manager Broker 다운로드 링크 <p>4. app.py 파일을 저장하고 닫습니다.</p>	

작업	설명	필요한 기술
<p>솔루션을 배포합니다.</p>	<p>다음 명령을 순서대로 실행합니다.</p> <pre data-bbox="597 348 1027 583">cdk bootstrap cdk deploy Assets-Stack Parameters-Stack cdk deploy Elastic-Vdi-Infrastructure</pre> <p>배포가 완료되면 다음 두 가지 출력이 반환됩니다.</p> <ul data-bbox="597 751 1027 1228" style="list-style-type: none"> • Elastic-Vdi-Infrastructure.EngineFrameURL — 포털의 EngineFrame HTTPS 주소 • Elastic-Vdi-Infrastructure.SecretEFadminPassword - eadmin 사용자의 암호가 포함된 보안 암호의 Amazon 리소스 이름 (ARN) <p>위의 값을 기록해 둡니다. 나중에 이 패턴에서 해당 값을 사용합니다.</p>	<p>클라우드 아키텍트</p>

작업	설명	필요한 기술
Linux 서버 플릿을 배포합니다.	<ol style="list-style-type: none"> 1. AWS Management Console에 로그인하고 CloudFormation 콘솔을 엽니다. 2. 스택 생성을 선택한 다음 새 리소스 사용(표준)을 선택합니다. 3. 클라우드포메이션_files 폴더에서.yaml 파일을 선택합니다. dcv-linux-fleet 4. 스택 세부 정보 지정 페이지에서 다음 파라미터를 정의합니다. <ul style="list-style-type: none"> • Stack name - 스택의 이름입니다. • DcvFleet— 닌스 DCV 플릿의 이름 이 값을 비워 두거나 공백을 사용하지 마세요. • InstanceType— 플릿의 인스턴스 유형. • RootVolumeSize— Linux EC2 인스턴스의 루트 볼륨 크기. • MinSize— DCV 세션을 실행하지 않고 사용할 수 있어야 하는 최소 노드 수입니다. 예를 들어 2을(를) 입력하면 솔루션이 2개의 노드로 시작됩니다. 사용자가 세션을 생성하면 사용 가능한 노드 수가 1로 줄어들고 솔루션은 최소 노드 수를 유지하기 위해 	클라우드 아키텍트

작업	설명	필요한 기술
	<p>또 다른 노드를 생성합니다.</p> <ul style="list-style-type: none"> • MaxSize— 플릿의 최대 노드 수 최댓값에 도달하면 사용자는 새 세션을 시작할 수 없습니다. • BillingTagName— 대금 청구에 사용되는 태그 이름. 이 태그 이름은 Windows 스택에 사용된 이름과 달라야 합니다. • BillingTagValue— 대금 청구에 사용되는 태그 값입니다. <p>5. 스택 생성 마법사를 완료한 다음 전송을 선택하여 스택 생성을 시작합니다.</p>	

작업	설명	필요한 기술
<p>Windows 서버 플릿을 배포합니다.</p>	<ol style="list-style-type: none"> 1. AWS Management Console에 로그인하고 CloudFormation 콘솔을 엽니다. 2. 스택 생성을 선택한 다음 새 리소스 사용(표준)을 선택합니다. 3. 클라우드포메이션_files 폴더에서.yaml 파일을 선택합니다. dcv-windows-fleet 4. 스택 세부 정보 지정 페이지에서 다음 파라미터를 정의합니다. <ul style="list-style-type: none"> • Stack name - 스택의 이름입니다. • DcvFleet— 닐스 DCV 플릿의 이름 이 값을 비워 두거나 공백을 사용하지 마세요. • InstanceType— 플릿의 인스턴스 유형. • RootVolumeSize— Windows EC2 인스턴스의 루트 볼륨 크기. • MinSize— DCV 세션을 실행하지 않고 사용할 수 있어야 하는 최소 노드 수입니다. • MaxSize— 플릿의 최대 노드 수 • BillingTagName— 대금 청구에 사용되는 태그 이름. 이 태그 이름은 Linux 	<p>클라우드 아키텍트</p>

작업	설명	필요한 기술
	<p>스택에 사용된 이름과 달라야 합니다.</p> <ul style="list-style-type: none"> • BillingTagValue— 대금 청구에 사용되는 태그 값입니다. <p>5. 스택 생성 마법사를 완료한 다음 전송을 선택하여 스택 생성을 시작합니다.</p>	

배포한 환경에 액세스

작업	설명	필요한 기술
EnginFrame 관리자 비밀번호를 검색하세요.	<p>EnginFrame 관리 계정의 이름은 eadmin이고 암호는 AWS Secrets Manager에 비밀번호로 저장됩니다. 보안 암호의 ARN은 동적으로 생성되며 AWS CDK 배포의 출력에 표시됩니다.</p> <ol style="list-style-type: none"> 1. 이전 에픽의 솔루션 배포 스토리의 Elastic-Verdi-Infrastructure.SecretEFAdminPassword 출력 아래에서 생성된 보안 암호의 ARN을 찾습니다. 2. 보안 암호를 검색하려면 다음 중 하나를 수행합니다. <ul style="list-style-type: none"> • Secrets Manager 콘솔을 사용합니다. 자세한 내용은 보안 암호 검색을 참조하세요. 	클라우드 아키텍트

작업	설명	필요한 기술
	<ul style="list-style-type: none"> • get-secret-value 명령을 입력합니다. <pre data-bbox="662 331 1029 646">aws secretsmanager get-secret-value \ --secret-id <secret_arn> \ --query SecretString \ --output text</pre>	
EnginFrame 포털에 액세스하십시오.	<ol style="list-style-type: none"> 1. 이전 에픽의 솔루션 배포 스토리의 Elastic-Verdi-Infrastructure. EnginFrameURL 결과 아래에서 포털의 EnginFrame HTTPS 주소를 찾으세요. 2. 웹 브라우저에서 포털의 HTTPS 주소를 입력합니다. 3. eadmin 사용자의 보안 인증 정보를 입력합니다. 	클라우드 아키텍트
Windows 세션을 시작합니다.	<ol style="list-style-type: none"> 1. EnginFrame 포털의 메뉴에서 Windows 데스크톱을 선택합니다. 2. Windows 관리자로 로그인 하라는 메시지가 표시되면 eadmin 사용자에게 사용한 것과 동일한 암호를 입력합니다. 3. Windows 세션이 성공적으로 시작되었는지 확인합니다. 	클라우드 아키텍트

작업	설명	필요한 기술
Linux 세션을 시작합니다.	<ol style="list-style-type: none"> 1. EnginFrame 포털의 메뉴에서 Linux 데스크톱을 선택합니다. 2. 로그인하라는 메시지가 표시되면 eadmin 사용자의 보안 인증 정보를 입력합니다. 3. Linux 세션이 성공적으로 시작되었는지 확인합니다. 	클라우드 아키텍트

정리

작업	설명	필요한 기술
스택을 삭제합니다.	AWS CloudFormation 콘솔에서 Windows 및 Linux 서버 플릿의 스택을 삭제합니다. 자세한 내용은 스택 삭제 를 참조하세요.	클라우드 아키텍트
인프라를 삭제합니다.	다음 AWS CDK 명령을 사용하여 배포한 인프라를 삭제합니다. <pre>cdk destroy --all</pre>	클라우드 아키텍트

문제 해결

문제	Solution
배포가 중단되어 완료되지 않았습니다.	정리 에픽의 지침을 따른 다음 이 패턴을 반복하여 환경을 다시 배포합니다.

관련 리소스

- [NICE DCV](#)
- [멋지네요 EnginFrame](#)

하이브리드 클라우드

주제

- [하이브리드 연결 모드를 사용하여 AWS의 VMware Cloud로의 데이터 센터 확장 구성](#)
- [AWS의 VMware Cloud에서 VM을 프로비저닝하도록 VMware vRealize Automation을 구성합니다.](#)
- [AWS의 VMware Cloud를 사용하여 AWS에 VMware SDDC 배포](#)
- [VMware vRealize 네트워크 인사이트를 VMware Cloud on AWS와 통합](#)
- [HCX OS 지원 마이그레이션을 사용하여 VM을 AWS의 VMware Cloud로 마이그레이션](#)
- [VMware Aria Operations for Logs를 사용하여 VMware Cloud on AWS에서 스포링크로 로그를 전송합니다.](#)
- [AWS CDK를 사용하여 Amazon ECS Anywhere에서 하이브리드 워크로드를 위한 CI/CD 파이프라인을 설정하고 GitLab](#)
- [더 많은 패턴](#)

하이브리드 연결 모드를 사용하여 AWS의 VMware Cloud로의 데이터 센터 확장 구성

작성자: Deepak Kumar(AWS)

환경: 프로덕션

기술: 하이브리드 클라우드, 인프라, 마이그레이션

워크로드: 기타 모든 워크로드

AWS 서비스: AWS Direct Connect

요약

이 패턴은 [하이브리드 연결 모드](#)를 사용하여 단일 VMware vSphere Client 인터페이스를 사용하여 온프레미스 데이터 센터 및 AWS의 VMware Cloud 소프트웨어 정의 데이터 센터(SDDC)의 인벤토리를 보고 관리하는 방법을 설명합니다.

하이브리드 연결 모드를 구성하면 온프레미스 가상 머신과 애플리케이션을 클라우드 SDDC로 마이그레이션할 수 있습니다. 그러면 IT 팀은 새로운 도구 없이도 친숙한 VMware 도구를 사용하여 클라우드 기반 리소스를 관리할 수 있습니다. 또한 [VMware Cloud Gateway Appliance](#)를 사용하여 일관된 운영과 간소화된 관리를 보장할 수 있습니다.

이 패턴은 하이브리드 연결 모드를 구성하는 두 가지 옵션을 제공하지만 한 번에 한 가지 옵션만 사용할 수 있습니다. 첫 번째 옵션은 클라우드 게이트웨이 어플라이언스를 설치하고 이를 사용하여 온프레미스 vCenter Server에서 클라우드 SDDC로 연결하는 것입니다. 두 번째 옵션은 클라우드 SDDC에서 하이브리드 연결 모드를 구성하는 것입니다.

사전 조건 및 제한 사항

사전 요건(두 옵션 모두)

- 기존 온프레미스 데이터 센터 및 클라우드 SDDC.
- AWS Direct Connect나 VPN 또는 둘 다를 사용하는 온프레미스 데이터 센터와 클라우드 SDDC 간의 기존 연결입니다.
- 온프레미스 데이터 센터와 클라우드 SDDC는 네트워크 시간 프로토콜(NTP) 또는 다른 신뢰할 수 있는 시간 소스와 동기화됩니다.

- 온프레미스 데이터 센터와 클라우드 SDDC 간 왕복 시간의 최대 지연 시간은 100ms를 초과하지 않습니다.
- 온프레미스 환경에 액세스할 수 있는 클라우드 관리자.
- vCenter Server의 정규화된 도메인 이름(FQDN)은 프라이빗 IP 주소로 확인되어야 합니다.

옵션 1의 사전 요건

- 온프레미스 환경은 vSphere 6.5.0d 이상에서 실행되어야 합니다.
- 클라우드 게이트웨이 어플라이언스와 vCenter 서버는 AWS Direct Connect, VPN 또는 둘 다를 통해 통신할 수 있습니다.
- 클라우드 게이트웨이 어플라이언스는 하드웨어 요구 사항을 충족합니다.
- 방화벽 포트가 열려 있습니다.

옵션 2의 사전 조건

- 온프레미스 vCenter 서버는 vSphere 6.0 업데이트 3 이상 또는 vSphere 6.5.0d 이상에서 실행됩니다.
- 온프레미스 vSphere AWS Single Sign-On(SSO) 도메인에 대한 로그인 보안 인증을 사용할 수 있습니다.
- 온프레미스 환경의 사용자는 기본 고유 이름 (Base DN) 에 대한 읽기 전용 액세스 권한을 가집니다.
- 온프레미스 도메인 이름 시스템(DNS) 서버는 VMware Management Gateway용으로 구성되어 있습니다.
- VMware Connectivity Validator를 사용하여 네트워크 연결 테스트를 구현하십시오.
- 방화벽 포트가 열려 있습니다.

제한 사항

- 하이브리드 연결 모드는 하나의 온-프레미스 [vCenter Sever Enhanced Linked Mode](#) 도메인만 연결할 수 있습니다.
- 하이브리드 연결 모드는 버전 6.7 이상을 실행하는 온프레미스 vCenter Server만 지원합니다.

아키텍처

다음 다이어그램은 하이브리드 연결 모드를 구성하기 위한 두 가지 옵션을 보여줍니다.

하이브리드 연결 모드를 사용하여 다양한 워크로드 유형을 마이그레이션하기

하이브리드 연결 모드는 [VMware vSphere vMotion](#)을 사용한 [콜드 마이그레이션](#) 또는 라이브 마이그레이션을 사용하여 온프레미스 데이터 센터와 클라우드 SDDC 간에 워크로드를 마이그레이션할 수 있도록 지원합니다. 마이그레이션 방법을 선택할 때 고려해야 하는 요소에는 가상 스위치 유형 및 버전, 클라우드 SDDC에 대한 연결 유형, 가상 하드웨어 버전 등이 있습니다.

콜드 마이그레이션은 다운타임이 발생하는 VM에 적합합니다. VM을 종료하고 마이그레이션한 다음 다시 켤 수 있습니다. 활성 메모리를 복사할 필요가 없으므로 마이그레이션 시간이 더 빠릅니다. 다운타임을 허용하는 애플리케이션(예: Tier 3 애플리케이션 또는 개발 및 테스트 워크로드)에는 콜드 마이그레이션을 사용하는 것이 좋습니다. VM에서 다운타임이 발생하지 않는 경우 vMotion을 사용하여 업무상 중요한 애플리케이션에 라이브 마이그레이션을 수행하는 것을 고려해야 합니다.

다음 다이어그램은 하이브리드 연결 모드를 사용하는 다양한 워크로드 마이그레이션 유형에 대한 개요를 제공합니다.

도구

- [AWS의 VMware Cloud](#)는 AWS와 VMware가 공동으로 개발한 통합 클라우드 서비스입니다.
- [VMware 클라우드 게이트웨이 어플라이언스](#)는 온프레미스 리소스가 클라우드 리소스에 연결된 다양한 하이브리드 클라우드 사용 사례를 지원합니다.
- [VMware vSphere](#)는 데이터 센터를 CPU, 스토리지 및 네트워킹 리소스를 포함하는 통합 컴퓨팅 인프라로 전환하는 VMware의 가상화 플랫폼입니다.

에픽

옵션 1 - 클라우드 게이트웨이 어플라이언스와 함께 하이브리드 연결 모드 사용

작업	설명	필요한 기술
클라우드 게이트웨이 어플라이언스를 구성합니다.	1. AWS의 VMware Cloud 콘솔에 로그인하고 클라우드 게이트웨이 어플라이언스를 다운로드합니다.	클라우드 관리자

작업	설명	필요한 기술
	<p>2. 다음 두 단계에 따라 온프레미스 환경에 클라우드 게이트웨이 어플라이언스를 설치합니다.</p> <ul style="list-style-type: none"> 시작을 선택하여 클라우드 게이트웨이 어플라이언스를 구성한 다음 배포합니다. 하이브리드 연결 모드를 구성합니다. <p>자세한 내용 및 세부 단계는 VMware 설명서에서 vCenter Cloud Gateway 어플라이언스를 사용한 하이브리드 연결 모드 구성을 참조하십시오.</p>	

옵션 2 - 클라우드 SDDC에서 하이브리드 연결 모드 사용

작업	설명	필요한 기술
<p>클라우드 SDDC에서 하이브리드 연결 모드를 구성합니다.</p>	<ol style="list-style-type: none"> AWS의 VMware Cloud 콘솔에 로그인하고 연결 검사기를 사용하여 필요한 모든 네트워크 연결을 확인합니다. 이에 대한 자세한 내용은 VMware 설명서의 하이브리드 연결 모드에 대한 네트워크 연결 검증을 참조하십시오. 클라우드 SDDC의 vSphere Client에 로그인하고 메뉴를 	<p>클라우드 관리자</p>

작업	설명	필요한 기술
	<p>선택하고 관리를 선택한 다음 도메인을 선택합니다.</p> <p>3. 하이브리드 클라우드 섹션에서 연결된 도메인을 선택한 다음 온프레미스 vCenter Server에 연결합니다.</p> <p>4. 클라우드 SDDC SDDC Lightweight Directory Access Protocol(LDAP) 도메인에 자격 증명 소스를 추가합니다. 이에 대한 자세한 내용은 VMware 설명서의 SDDC LDAP 도메인에 ID 소스 추가를 참조하십시오.</p>	

관련 리소스

- [하이브리드 연결 모드 구성](#)
- [AWS의 VMware Cloud를 위한 하이브리드 연결 모드 구성](#)

AWS의 VMware Cloud에서 VM을 프로비저닝하도록 VMware vRealize Automation을 구성합니다.

작성자: Deepak Kumar (AWS)

환경: 프로덕션

기술: 하이브리드 클라우드, 인 프라

워크로드: 기타 모든 워크로드

AWS 서비스: AWS Direct Connect, AWS Site-to-Site VPN

요약

[VMware vRealize Automation](#)은 IT 리소스를 요청하고 관리하는 데 사용할 수 있는 자동화 소프트웨어입니다. AWS의 VMware Cloud를 사용하여 vRealize Automation을 구성하면 여러 데이터 센터 및 클라우드 환경에서 가상 머신(VM), 애플리케이션 및 IT 서비스의 전송을 자동화할 수 있습니다.

그런 다음 IT 팀이 카탈로그 항목을 생성하여 사용자가 기존 vRealize Automation 도구로 요청하고 사용할 수 있는 서비스 프로비저닝 및 운영 기능을 구성할 수 있습니다. 또한 AWS의 VMware Cloud를 [vRealize Automation Cloud Assembly](#)와 통합하여 IT 민첩성과 효율성을 개선할 수 있습니다.

이 패턴은 AWS의 VMware Cloud에서 VM 또는 애플리케이션 기능을 자동으로 구축하도록 VMware vRealize Automation을 구성하는 방법을 설명합니다.

사전 조건 및 제한 사항

사전 조건

- 기존 온프레미스 데이터 센터 및 AWS의 VMware Cloud 소프트웨어로 정의된 데이터 센터(SDDC). 클라우드 SDDC에 대한 자세한 내용은 VMware 설명서의 [소프트웨어 정의 데이터 센터 정보](#)를 참조하세요.
- AWS Direct Connect나 VPN(경로 또는 정책 기반) 또는 둘 다를 사용한 온프레미스 데이터 센터와 클라우드 SDDC 간의 기존 연결.
- 온프레미스 데이터 센터와 클라우드 SDDC는 네트워크 시간 프로토콜(NTP) 또는 다른 신뢰할 수 있는 시간 소스로 동기화됩니다.

- 온프레미스 데이터 센터와 클라우드 SDDC 간 왕복 시간의 최대 지연 시간은 100ms를 초과하지 않습니다.
- vCenter Server의 정규화된 도메인 이름(FQDN)은 사실 IP 주소로 확인되어야 합니다.
- 온프레미스 환경에 액세스할 수 있는 클라우드 SDDC 사용자.
- vRealize Automation Cloud Assembly 서비스 역할의 조직 소유자 액세스.
- 서비스를 이용하려는 vRealize Automation Service Broker의 이용 권한이 있는 최종 사용자.
- AWS의 VMware Cloud 콘솔에서 API 토큰을 생성하려면 온프레미스 데이터 센터의 Classless Inter-Domain Routing (CIDR) 범위가 개방되어야 합니다. 다음 목록은 API 토큰 생성에 필요한 최소 역할을 제공합니다.
 - 조직 멤버
 - 조직 소유자
 - 서비스 역할 - AWS의 VMware Cloud
 - 관리자
 - NSX Cloud Administrator
 - NSX Cloud Auditor

이에 대한 자세한 내용은 AWS 파트너 네트워크 블로그의 [AWS의 VMware Cloud SDDC용 연결 옵션](#)을 참조하세요.

제한 사항

- 하나의 vRealize Automation에서 퍼블릭 엔드포인트가 있는 VMware Cloud 계정을 20개만 구성할 수 있습니다. 이에 대한 자세한 내용은 VMware 설명서의 [확장성 및 동시성 최대값](#)을 참조하세요.

제품 버전

- vRealize Automation 버전 8.x 이상
- VMware vRealize Identity Manager 버전 3.x 이상
- VMware vRealize Suite Lifecycle Manager 버전 8.x 이상

아키텍처

다음 다이어그램은 온프레미스와 AWS의 VMware Cloud 환경 모두에서 인프라를 사용할 수 있는 vRealize Automation 서비스를 보여줍니다.

VMware Cloud Assembly 구성 요소

VMware Cloud Assembly는 vRealize Automation의 핵심 구성 요소이며 이를 사용하여 VM 및 컴퓨팅 리소스를 배포하고 프로비저닝할 수 있습니다. 다음 표는 AWS의 VMware Cloud에서 VM을 프로비저닝하기 위해 구성해야 하는 VMware Cloud Assembly 구성 요소에 대해 설명합니다.

구성 요소	정의
클라우드 계정	클라우드 계정은 연결 세부 정보(예: 서버 이름, 사용자 이름 및 암호, 액세스 키, API 토큰)를 제공합니다. VMware Cloud Assembly는 클라우드 계정을 사용하여 리소스 인벤토리를 수집합니다.
클라우드 영역	클라우드 영역은 클라우드 계정의 리소스 경계를 식별합니다(예: AWS 리전 및 클라우드 SDDC). 클라우드 영역은 컴퓨팅 리소스를 Cloud Assembly 프로젝트와 연결합니다.
프로젝트	프로젝트는 사용자와 리소스(예: 클라우드 영역)로 구성된 논리적 객체입니다. 또한 VM을 구축할 때 사용되는 리소스 할당량 및 VM 이름 지정 정책으로 구성됩니다.
플레이버 매핑	플레이버 매핑은 클라우드 템플릿에서 사용되는 VM의 용량(예: CPU 수 및 메모리 양)에 대한 정보를 제공합니다.
이미지 매핑	이미지 매핑은 클라우드 템플릿에서 사용되는 VMware vSphere VM 템플릿과 Amazon Web Services (AWS) 이미지를 매핑합니다. 이에 대한 자세한 내용은 VMware 설명서의 vRealize Automation의 이미지 매핑에 대한 자세한 내용을 참조하세요 .
네트워크 프로필	네트워크 프로필은 VM 프로비저닝 중에 네트워크를 선택하기 위한 배치 결정을 제어합니다.

스토리지 프로파일

스토리지 프로파일은 VM 프로비저닝 중에 스토리지를 선택하기 위한 배치 결정을 제어합니다.

클라우드 템플릿

VMware 클라우드 템플릿은 클라우드 인프라 프로비저닝과 오케스트레이션을 정의하므로 vRealize Automation의 중요한 구성 요소입니다. 클라우드 템플릿은 리소스의 사양이며 리소스 유형, 리소스 속성, 사용자로부터 수집할 입력을 포함합니다.

도구

- [VMware vRealize Automation](#)—vRealize Automation은 이벤트 기반 상태 관리 및 규정 준수를 지원하는 인프라 자동화 플랫폼입니다. 조직이 셀프 서비스 클라우드, 거버넌스를 통한 멀티클라우드 자동화, DevOps 기반 인프라 제공을 제어하고 보호할 수 있도록 설계되었습니다.
- [AWS의 VMware Cloud](#)—AWS의 VMware Cloud는 AWS와 VMware가 공동으로 개발한 통합 클라우드 서비스입니다.

에픽

API 토큰 생성

작업	설명	필요한 기술
AWS의 VMware Cloud 계정에서 API 토큰을 생성합니다.	<ol style="list-style-type: none"> 1. VMware Cloud 콘솔에 로그인합니다. 2. VMware Cloud 서비스 도구 모음에서 내 계정을 선택한 다음 API 토큰을 선택합니다. 3. API 토큰의 이름을 입력하고, 필요한 수명을 제공하고, 토큰의 범위를 정의합니다. 4. Open ID 확인란을 선택한 다음 생성을 선택합니다. 	클라우드 관리자

작업	설명	필요한 기술
	<p>5. API 토큰의 보안 인증 정보를 기록합니다.</p> <p>이에 대한 자세한 내용은 VMware 설명서의 API 토큰을 생성하는 방법을 참조하세요.</p>	

온프레미스 데이터 센터에 vRealize Automation을 설치합니다.

작업	설명	필요한 기술
<p>필수 소프트웨어를 다운로드합니다.</p>	<p>My VMware 포털에서 VMware vRealize Suite ISO 파일을 다운로드합니다. 이 패키지에는 vRealize Suite Lifecycle Manager, VMware Identity Manager 및 vRealize Automation이 포함되어 있습니다.</p>	<p>클라우드 관리자</p>
<p>소프트웨어를 설치합니다.</p>	<p>VMware 설명서에 있는 vRealize Automated VMware Identity Manager용 간편 설치 프로그램을 사용한 vRealize Suite Lifecycle Manager 설치의 지침에 따라 소프트웨어를 설치하고 클라우드 SDCC에 연결합니다.</p> <p>중요: 설치 시 다음 항목을 사용할 수 있는지 확인합니다.</p> <ul style="list-style-type: none"> 온프레미스 VMware vCenter 서버 설치 및 로그인 보안 인증 정보 	<p>클라우드 관리자, 클라우드 아키텍트</p>

작업	설명	필요한 기술
	<ul style="list-style-type: none"> vRealize Automation IP 및 서브넷에 대한 네트워크 세부 정보 vRealize Automation 라이선스 키 	

AWS의 VMware Cloud를 VMware Cloud Assembly와 연결

작업	설명	필요한 기술
Cloud 계정 구성	<ol style="list-style-type: none"> VMware Cloud 콘솔에서 인 프라 탭을 열고 관리-클라우드 계정을 선택한 다음 클라우드 계정 추가를 선택합니다. 유형으로 AWS의 VMware Cloud를 선택합니다. 이전에 기록한 API 토큰 정보를 붙여 넣습니다. 이렇게 하면 AWS의 VMware Cloud 조직에서 사용 가능한 모든 클라우드 SDDC가 채워집니다. 필요한 클라우드 SDCC를 선택한 다음 SDDC에 대한 vCenter 사용자 이름과 암호를 제공합니다. 인증에 성공하면 통합된 AWS의 VMware Cloud 계정을 OK 상태로 볼 수 있습니다. 	클라우드 아키텍트, 클라우드 관리자

작업	설명	필요한 기술
	<p>이에 대한 자세한 내용은 VMware 설명서의 vRealize Automation에서 AWS의 VMware Cloud 계정 생성을 참조하세요.</p>	
<p>프로젝트를 구성합니다.</p>	<ol style="list-style-type: none"> 1. VMware Cloud 콘솔에서 프로젝트 탭을 연 다음 새 프로젝트를 선택합니다. 2. 프로젝트 이름을 입력합니다. 3. 클라우드 영역 탭을 열고 AWS의 VMware Cloud 기본 클라우드 계정을 선택합니다. 	<p>클라우드 관리자</p>
<p>클라우드 영역을 구성합니다.</p>	<ol style="list-style-type: none"> 1. VMware Cloud 콘솔에서 클라우드 영역을 열고 SDDC 데이터 센터의 클라우드 영역을 선택합니다. 2. 기본적으로 <code>cloudadmin@vmc.local</code> (클라우드 SDDC의 vCenter에 대한 기본 로컬 사용자 ID)으로만 Compute-ResourcePool에서 프로비저닝할 수 있습니다. 3. 클라우드 영역 아래의 컴퓨팅 탭을 연 다음 Compute-를 선택합니다. ResourcePool 	<p>클라우드 관리자</p>

작업	설명	필요한 기술
플레이버 매핑을 구성합니다.	<ol style="list-style-type: none"> 1. 플레이버 매핑 탭을 열고 새 플레이버 매핑을 생성합니다. 2. 플레이버 이름을 입력하고 AWS의 VMware Cloud 계정을 선택한 다음 vCPU 수와 메모리 양을 입력합니다. 	클라우드 관리자
이미지 매핑을 구성합니다.	<ol style="list-style-type: none"> 1. 이미지 매핑을 열고 새 이미지 매핑을 생성합니다. 2. 이미지 이름을 입력합니다. 3. AWS의 VMware Cloud 계정을 선택하고 필요한 클라우드 계정 템플릿을 제공합니다. 	클라우드 관리자
네트워크 프로필을 구성합니다.	<ol style="list-style-type: none"> 1. 네트워크 프로필을 열고 새 네트워크 프로필을 생성합니다. 2. 네트워크 프로필 이름을 입력합니다. 3. 네트워크 탭을 열고 프로비저닝에 사용할 기존 네트워크를 선택합니다. 	클라우드 관리자

작업	설명	필요한 기술
스토리지 프로필을 구성합니다.	<ol style="list-style-type: none">1. 스토리지 프로필을 열고 새 스토리지 프로필을 선택합니다.2. 스토리지 프로필 이름을 입력합니다.3. 정책 페이지에서 새 정책 생성을 선택합니다.4. 워크로드 데이터 스토어를 선택합니다. 기본적으로 워크로드의 데이터 스토어에서 <code>cloudadmin@vmc.local</code> 만 프로비저닝할 수 있습니다.	클라우드 관리자

작업	설명	필요한 기술
클라우드 템플릿을 생성합니다.	<ol style="list-style-type: none"> 1. 디자인 탭을 열고 클라우드 템플릿을 선택한 다음 새로 만들기 및 빈 캔버스를 선택합니다. 2. 클라우드 템플릿의 이름과 설명을 입력합니다. 3. 앞서 생성한 프로젝트를 선택합니다. 4. 클라우드 템플릿 리소스 디자인 페이지에서 요구 사항에 따라 구성 요소를 빈 캔버스에 드래그합니다. 5. 테스트를 선택하여 템플릿을 테스트하고 문제를 해결합니다. 6. 배포를 선택하고 VM을 배포할 배포 이름을 입력합니다. <p>이에 대한 자세한 내용은 VMware 설명서의 기본 클라우드 템플릿 생성을 참조하세요.</p>	클라우드 관리자

관련 리소스

- [vRealize Automation 버전 8.x를 SDDC에 연결합니다.](#)
- [AWS의 VMware Cloud 콘솔에서 SDDC를 배포](#)
- [AWS Direct Connect와 AWS의 VMware Cloud 통합](#)

AWS의 VMware Cloud를 사용하여 AWS에 VMware SDDC 배포

작성자: Deepak Kumar (AWS)

환경: 프로덕션

기술: 하이브리드 클라우드, 인
프라

워크로드: 기타 모든 워크로드

서비스: Amazon VPC

요약

이 패턴은 Amazon Web Services(AWS) 클라우드에서 호스팅되는 VMware 기반 소프트웨어 정의 데이터 센터(SDDC)를 생성하는 방법을 설명합니다. SDDC를 배포하여 VMware vSphere 기반 워크로드를 클라우드로 마이그레이션하고, 기존 VMware 도구 및 기술을 사용하면서 서비스를 활용할 수 있습니다. 이 SDDC를 사용하면 서비스에 대한 액세스를 최적화하면서 VMware vSphere 기반 프라이빗, 퍼블릭, 하이브리드 클라우드 환경 전반적으로 프로덕션 애플리케이션을 실행할 수 있습니다. 예를 들어 SDDC를 재해 복구용 보조 사이트로 사용하거나 데이터 센터를 다른 지리적 위치로 확장할 수 있습니다.

VMware Cloud on AWS는 pay-as-you-go 모든 규모의 기업이 다양한 AWS 서비스를 사용하여 VMware vSphere 기반 클라우드 환경에서 워크로드를 실행할 수 있도록 지원하는 (온디맨드) 서비스입니다. SDDC 클러스터당 최소 2개의 호스트로 시작하여 프로덕션 환경에서 클러스터당 최대 16개의 호스트로 확장할 수 있습니다. 자세한 내용은 [AWS의 VMware Cloud](#) 웹사이트를 참조하세요. SDDC에 대한 자세한 내용은 VMware 설명서의 [소프트웨어 정의 데이터 센터](#)를 참조하세요.

사전 조건 및 제한 사항

사전 조건

- [MyVMware 계정](#)에 가입하고 모든 필드를 작성합니다.
- [계정](#)에 가입합니다. 지침은 [지식 센터](#)를 참조하십시오.
- MyVMware 클라우드 계정에 가입합니다. 가입 시 지정한 이메일 주소로 활성화 링크가 전송됩니다.

제한 사항

- VMware 웹 사이트에서 [AWS의 VMware Cloud 구성 제한](#) 페이지를 참조하십시오.

제품 버전

- VMware 설명서에서 [VMware Cloud 릴리스 정보](#)를 참조하십시오.

아키텍처

대상 기술 스택

다음 다이어그램은 베어 메탈 전용 인프라에서 실행되는 vSphere, vCenter, vSAN, NSX-T를 포함한 VMware 소프트웨어 스택을 보여줍니다. Amazon Elastic Compute Cloud(Amazon EC2), Amazon Simple Storage Service(S3), Amazon Redshift, Direct Connect, Amazon Relational Database Service(Amazon RDS), Amazon DynamoDB와 같은 다른 서비스와의 지속적인 통합을 활용하여 VMware 기반 리소스 및 도구를 관리할 수 있습니다.

VMware Cloud의 기본 개체는 SDDC이며, 여기에는 다음과 같은 구성 요소가 포함됩니다.

- 컴퓨팅: 컴퓨팅 구성 요소는 VMware Cloud SDDC의 최하위 계층입니다. VMware Cloud는 Amazon EC2 베어 메탈 인스턴스 유형에서 실행됩니다. 여기에서는 `i3.metal`, `i3en.metal`, `i4i.metal`(이)가 포함되며 프로세서 및 메모리와 같은 물리적 리소스에 직접 액세스할 수 있습니다.

중요: 1년 및 3년 약정의 온디맨드 및 구독 옵션을 포함한 VMware Cloud에 대한 `i3.metal` 인스턴스 유형은 2026년 12월 31일에 수명이 종료되고 지원이 종료될 예정입니다. 또한 신규 고객은 현재 `i3.metal` 인스턴스를 요청할 수 없습니다. 자세한 내용은 [VMware Cloud 블로그의 공지](#)를 참조하십시오.

- 스토리지: SDDC 클러스터는 고속 고성능 스토리지를 제공하는 NVMe (비휘발성 메모리 익스프레스) 플래시 스토리지를 사용하는 스토리지를 위한 올플래시 구성으로 VMware vSAN을 지원합니다. SDDC 버전 1.20부터 VMware Cloud on AWS는 ONTAP용 Amazon NetApp FSx와 VMware 클라우드 플렉스 스토리지라는 두 가지 유형의 외부 스토리지를 지원합니다.
- 네트워킹: 네트워킹 기능 및 정책은 SDDC 클러스터에서 VMware NSX-T를 사용하여 관리합니다. 네트워크 리소스를 물리적 장비와 분리하기 위해 SDDC 클러스터에 다중 계층 가상 네트워크가 생성됩니다. 이를 통해 VMware Cloud 사용자는 논리적인 소프트웨어 정의 네트워크를 생성할 수 있습니다.

도구

- [VMware Cloud](#)는 AWS와 VMware가 공동으로 개발한 통합 클라우드 서비스입니다.

에픽

계정에서 VPC와 서브넷 생성

작업	설명	필요한 기술
계정에 로그인.	관리자 권한이 있는 자격 증명으로 계정 에 로그인합니다.	클라우드 관리자
새 VPC를 생성합니다.	<p>이 단계에서는 SDDC에 연결되는 Virtual Private Cloud(VPC)를 정의합니다. SDDC에 사용하려는 VPC가 이미 있는 경우 이 단계를 건너뛴니다.</p> <ol style="list-style-type: none"> 1. 리전을 선택하여 VMware Cloud SDDC를 배포합니다. 2. https://console.aws.amazon.com/vpc/에서 Amazon VPC 콘솔을 엽니다. 3. 탐색 창에서 사용자 VPC(Your VPCs)를 선택합니다. 4. VPC 생성을 선택합니다. 5. VPC 이름 태그, IPv4 CIDR 블록, 테넌시(기본 값으로 유지)와 같은 VPC 설정을 지정한 다음 VPC 생성을 선택합니다. 6. VPC 생성을 마치면 닫기를 선택합니다. 	클라우드 관리자

작업	설명	필요한 기술
	<p>자세한 내용은 설명서에서 VPC 생성 및 구성을 참조하십시오.</p>	
<p>프라이빗 서브넷을 생성합니다.</p>	<p>이제 각 가용 영역에 대해 탄력적 네트워크 인터페이스(ENI)에 대한 프라이빗 서브넷을 생성합니다. 인터넷 게이트웨이 가 연결되지 않은 서브넷을 사용하는 것이 좋습니다.</p> <ol style="list-style-type: none"> 1. https://console.aws.amazon.com/vpc/에서 Amazon VPC 콘솔을 엽니다. 2. 탐색 창에서 서브넷(Subnets)을 선택합니다. 3. 서브넷 생성을 선택합니다. 4. 서브넷 생성 페이지에서, 앞서 생성한 VPC를 선택합니다. 5. 서브넷 이름, 가용 영역, IPv4 CIDR 블록을 포함하여 서브넷에 대한 설정을 완료합니다. 6. 서브넷 생성을 선택합니다. <p>리전의 나머지 가용 영역마다 이 단계들을 반복하여 서브넷을 생성합니다.</p>	<p>클라우드 관리자</p>

VMware Cloud 활성화

작업	설명	필요한 기술
서비스를 활성화합니다.	<p>MyVMware 계정에 가입하면 VMware에서 지정 이메일 주소로 환영 이메일과 활성화 링크를 발송합니다.</p> <ol style="list-style-type: none"> 1. 브라우저의 환영 이메일에서 서비스 활성화 링크를 엽니다. 2. MyVMware 자격 증명을 사용하여 로그인합니다. 3. 서비스 사용 약관을 검토하고 동의합니다. 4. 계정 활성화 프로세스를 완료합니다. AWS의 VMware Cloud 콘솔로 리디렉션됩니다. (참고: VMware Cloud 계정은 계정에 가입한 그룹 또는 사업부를 대표하는 조직을 기반으로 합니다. 이 조직은 AWS Organizations와 아무런 관련이 없습니다.) 5. 조직 선택 또는 생성 페이지에서, MyVMware 계정에 연결된 조직을 생성합니다. 6. 논리적으로 구분할 수 있도록 조직 이름과 주소를 입력합니다. 7. 조직 생성(Create Organization)을 선택하여 프로세스를 마칩니다. 	클라우드 관리자

작업	설명	필요한 기술
	<p>이 프로세스에 대한 자세한 내용은 설명서에서 SDDC 배포 및 모범 사례 가이드를 참조하십시오.</p>	
<p>IAM 역할을 할당합니다.</p>	<p>조직이 생성되면 특정 사용자에게 클라우드 서비스 및 SDDC 콘솔, SDDC, NSX 구성 요소에 액세스할 수 있는 권한을 할당합니다. 지침은 VMware 설명서에서 조직 구성원에게 VMC 서비스 역할 할당을 참조하십시오.</p> <p>조직 역할에는 두 가지 유형이 있습니다.</p> <ul style="list-style-type: none"> 조직 소유자는 사용자를 추가, 제거, 수정하고 모든 클라우드 리소스에 액세스할 수 있습니다. 조직 구성원은 클라우드 리소스에만 액세스할 수 있습니다. 	<p>클라우드 관리자</p>

SDDC 배포

작업	설명	필요한 기술
<p>VMware Cloud에서 SDDC를 배포합니다.</p>	<p>중요: AWS 계정이 등록 판매자로 VMware 조직에 연결된 후에는 AWS 계정 번호를 업데이트할 수 없습니다. VMware 조직당 AWS 등록 판매자는 한 명만 있을 수 있습니다.</p>	<p>클라우드 관리자, 클라우드 아키텍트</p>

작업	설명	필요한 기술
	<p>SDDC를 배포하려면:</p> <ol style="list-style-type: none"> 1. https://vmc.vmware.com에서 VMC 콘솔에 로그인합니다. 2. 사용 가능한 서비스 중에서 VMware Cloud 서비스를 선택합니다. 3. SDDC 생성을 선택합니다. 4. 리전, 배포(단일 호스트, 다중 호스트 또는 확장 클러스터), 호스트 유형, SDDC 이름, 호스트 수, 호스트 용량, 총 용량과 같은 SDDC 속성을 입력한 후 다음을 선택합니다. 5. 계정에 연결한 후 다음을 선택합니다. 6. 이전에 구성한 VPC와 서브넷을 선택한 후 다음을 선택합니다. 7. SDDC의 관리 서브넷 CIDR 블록을 입력한 후 다음을 선택합니다. 자세한 내용은 VMware 클라우드 블로그에서 SDDC의 IP 서브넷 및 연결 선택을 참조하십시오. 8. 두 개의 확인란을 선택하여 SDDC 배포 비용은 본인이 부담한다는 점을 확인한 다음 SDDC 배포를 선택합니다. 	

작업	설명	필요한 기술
	<p>SDDC 배포를 선택하면 요금이 부과됩니다. 배포 프로세스를 일시 중지하거나 취소할 수 없으며, 완료하는 데 시간이 좀 걸립니다.</p> <p>SDDC 생성에 대한 자세한 내용은 VMware 설명서에서 VMC 콘솔에서 SDDC 배포를 참조하십시오.</p>	

관련 리소스

- [소프트웨어 정의 데이터 센터 배포 및 관리](#)(VMware 설명서)
- [VMware Cloud 기능](#)(웹 사이트)
- [VMware Cloud를 사용하여 클라우드 마이그레이션 및 현대화를 가속화하십시오](#)(동영상)

VMware vRealize 네트워크 인사이트를 VMware Cloud on AWS와 통합

제작: 디팍 쿠마르 (AWS), 피터 피테라 (AWS), 사친 트리베디 (AWS)

환경: PoC 또는 파일럿	출처: VMware vRealize 네트워크 인사이트	대상: AWS의 VMware Cloud
R 유형: 재배치	워크로드: 기타 모든 워크로드	기술: 하이브리드 클라우드, 인프라, 마이그레이션
AWS 서비스: AWS의 VMware Cloud		

요약

이 패턴은 VMware vRealize Network Insight를 VMware Cloud와 AWS 통합하고 가상 시스템의 트래픽 흐름을 검사하는 방법을 설명합니다. 또한 이러한 통합을 통해 VMware Cloud on으로의 애플리케이션 마이그레이션을 계획할 수 있습니다. AWS

vRealize Network Insight는 네트워크 인프라에 대한 가시성을 제공합니다. 보안을 개선하고 마이그레이션 위험을 완화하며 성능을 최적화하는 네트워크 모니터링 및 분석 기능을 제공합니다. 이 도구를 사용하여 가상 시스템의 트래픽 흐름을 모니터링하고 관찰된 트래픽을 기반으로 권장되는 보안 규칙을 볼 수 있습니다. vRealize Network Insight에 대한 자세한 내용은 [VMware](#) 설명서를 참조하십시오.

VMware Cloud AWS on은 모든 규모의 기업이 다양한 솔루션을 사용하여 VMware vSphere 기반 클라우드 환경에서 워크로드를 실행할 수 있도록 하는 pay-as-you-go (온디맨드) 서비스입니다. AWS 서비스 SDDC 클러스터당 최소 2개의 호스트로 시작하여 프로덕션 환경에서 클러스터당 최대 16개의 호스트로 확장할 수 있습니다. [자세한 내용은 VMware Cloud on 웹 사이트를 참조하십시오.](#) AWS SDCC에 대한 자세한 내용은 VMware 설명서의 [소프트웨어 정의 데이터 센터](#)를 참조하세요.

사전 조건 및 제한 사항

사전 조건

- VMware 클라우드 온 AWS SDDC, 배포

제한 사항

- 알려진 제한 사항은 [VMware](#) 설명서를 참조하십시오.

제품 버전

- vRealize 네트워크 인사이트 버전 5.0.0
- VMware 클라우드 온 AWS SDDC 버전 1.24

아키텍처

소스 기술 스택

- vRealize 네트워크 인사이트

대상 기술 스택

- VMware 클라우드 온 AWS

대상 아키텍처

다음 다이어그램은 VMware 클라우드 AWS 온과 vRealize Network Insight 온프레미스 간의 연결을 보여줍니다.

도구

- [VMware Cloud on AWS](#)는 VMware가 공동으로 AWS 개발한 통합 클라우드 서비스입니다.
- [VMware vRealize Network Insight](#)는 보안 계획 및 문제 해결을 위해 네트워크 인프라에 대한 가시성을 제공하는 모니터링 및 분석 도구입니다.

에픽

vRealize 네트워크 인사이트를 위한 환경을 설정하십시오.

작업	설명	필요한 기술
VMware 사용자 계정을 생성하십시오.	<p>VMware 사용자 계정을 만들거나 기존 VMware 계정에 로그인합니다.</p> <p>새 계정을 개설하려면:</p> <ol style="list-style-type: none"> 등록 양식을 작성하여 VMware Customer Connect 계정을 등록하십시오. 신규 사용자는 계정을 활성화하라는 이메일을 받게 됩니다. 이메일에 있는 인증 코드를 입력합니다. 고객 연결에 로그인합니다. 	클라우드 관리자
vRealize 네트워크 인사이트용 OVA 파일을 다운로드하십시오.	<p>vRealize 네트워크 인사이트용 OVA 파일 다운로드:</p> <ol style="list-style-type: none"> https://my.vmware.com/group/vmware/home 에서 VMware 제품 다운로드 페이지로 이동하십시오. vRealize 네트워크 인사이트를 검색하십시오. 최신 vRealize 네트워크 인사이트 버전 5.0.0 플랫폼을 다운로드하고 OVA 파일을 수집하십시오. 	클라우드 관리자

작업	설명	필요한 기술
vRealize 네트워크 인사이트를 배포하십시오.	배포 지침은 VMware 설명서를 참조하십시오.	클라우드 관리자

데이터 소스 및 컬렉터 추가

작업	설명	필요한 기술
데이터 소스 추가.	<ol style="list-style-type: none"> vRealize 네트워크 인사이트에 로그인합니다. 설정, 계정 및 데이터 소스, 소스 추가를 선택합니다. 유형에서 온프레미스 vCenter 서버를 선택합니다. <p>자세한 내용은 VMware 설명서를 참조하십시오.</p>	클라우드 관리자
데이터 소스에 대한 컬렉터를 설정합니다.	지침은 VMware 설명서 를 참조하십시오.	클라우드 관리자

애플리케이션 종속성 분석

작업	설명	필요한 기술
애플리케이션을 생성합니다.	vRealize Network Insight에 기존 애플리케이션이 없는 경우 VMware 설명서의 단계에 따라 새로 만드십시오.	클라우드 관리자
애플리케이션을 검색하고 분석하십시오.	1. vRealize 네트워크 인사이트를 사용하여 애플리케이션을 검색하십시오. 자세한 지	클라우드 관리자

작업	설명	필요한 기술
	<p>침은 VMware 설명서를 참조하십시오.</p> <p>2. 애플리케이션을 분석하십시오. 지침은 VMware 설명서를 참조하십시오.</p>	

관련 리소스

- [VMware 클라우드 온을 사용하여 AWS에 VMware SDDC를 배포하십시오 AWS](#) (AWS 규범적 지침).
- [하이브리드 연결 모드를 AWS 사용하여 VMware Cloud로의 데이터 센터 확장 구성](#) (규범적 지침)AWS
- [VMware HCX를 AWS 사용하여 VMware SDDC를 VMware 클라우드로 마이그레이션하기](#) (규범적 지침)AWS
- VMware vRealize 네트워크 인사이트 설명서 ([VMware 웹 사이트](#))

HCX OS 지원 마이그레이션을 사용하여 VM을 AWS의 VMware Cloud로 마이그레이션

작성자: Deepak Kumar(AWS)

환경: PoC 또는 파일럿	소스: 비 vSphere 환경	대상: VMware 클라우드 온 AWS SDDC
R 유형: 재배치	워크로드: 기타 모든 워크로드	기술: 하이브리드 클라우드, 마이그레이션

요약

이 패턴은 OS 지원 마이그레이션(OSAM)을 사용하여 가상 머신(VM)을 vSphere 이외의 환경에서 Amazon Web Services(AWS)의 VMware 클라우드로 마이그레이션하는 방법을 설명합니다.

OSAM은 VMware 하이브리드 클라우드 확장(HCX)의 일부로, AWS의 VMware Cloud에 포함되어 있습니다. OSAM을 사용하여 VMware KVM 또는 Hyper-V와 같은 비 vSphere 환경을 VMware Cloud on AWS로 마이그레이션할 수 있습니다. OSAM은 Windows 또는 Linux 게스트 VM에 설치하는 Sentinel 소프트웨어를 사용하여 온프레미스 환경에서 AWS의 VMware Cloud 소프트웨어 정의 데이터 센터(SDDC)로 VM을 복제할 수 있도록 지원합니다.

이 패턴은 OSAM을 활성화하고, Windows VM에 Sentinel 소프트웨어를 설치하고, 소스 사이트에서 HCX Sentinel Gateway(SGW) 어플라이언스에 연결 및 등록하고, 대상 사이트에서 HCX Sentinel Data Receiver(SDR) 어플라이언스와의 포워딩 연결을 설정하여 마이그레이션을 시작하는 방법을 설명합니다.

OSAM에 대한 자세한 내용은 [VMware 설명서](#)를 참조하세요.

사전 조건 및 제한 사항

사전 조건

- 소스 및 대상 환경에 HCX를 설치합니다. HCX 사전 조건은 AWS 권장 가이드에서 [VMware HCX를 사용하여 VMware SDDC를 AWS의 VMware Cloud로 마이그레이션](#)을 참조하세요.
- OSAM 사전 조건은 VMware 설명서의 [설치 체크리스트](#)를 참조하세요.

- OSAM 포트 정보는 VMware 포트 및 프로토콜 웹사이트의 [VMware HCX 포트 요구 사항](#)을 참조하세요.

제한 사항

- [VMware HCX 4.2.0 구성 제한](#)
- [OSAM 배포에 대한 고려 사항](#)
- [지원되는 게스트 운영 체제](#)
- [게스트 운영 체제 고려 사항](#)

제품 버전

- VMware HCX 4.2.0
- VMware SDDC 1.12

아키텍처

다음 다이어그램은 HCX OSAM이 Sentinel 소프트웨어와 연동하여 온프레미스 환경에서 AWS의 VMware Cloud로 vSphere 이외의 VM을 복제하는 방법을 보여줍니다.

OSAM은 세 가지 구성 요소로 구성되어 있습니다.

- Sentinel Gateway(SGW) 어플라이언스는 소스 VMware 기반 환경에서 워크로드와 애플리케이션을 연결하고 전달하는 데 사용됩니다.
- 대상 AWS의 VMware Cloud 환경에서 원본으로부터 마이그레이션된 워크로드를 수신하는 데 사용되는 센티넬 데이터 수신기(SDR)
- 마이그레이션하려는 각 게스트 VM에 설치해야 하는 Sentinel 소프트웨어

OSAM은 Windows 또는 Linux 게스트 VM에 설치된 Sentinel 소프트웨어를 사용하여 온프레미스에서 VMware SDDC로 VM을 복제하는 작업을 지원합니다. 게스트 VM에 설치하는 Sentinel 소프트웨어는 게스트 VM에서 시스템 구성을 수집하여 데이터 복제를 지원합니다. 또한 이 정보는 마이그레이션할 게스트 VM의 인벤토리를 생성하는 데 사용되며 복제 및 마이그레이션을 위해 복제 VM의 디스크를 준비하는 데도 사용됩니다.

도구

- VMware HCX 4.2.0
- AWS의 VMware Cloud SDDC

에픽

HCX 구성

작업	설명	필요한 기술
HCX 클라우드 및 HCX 커넥터 배포	VMware 설명서의 HCX 커넥터 및 HCX 클라우드 설치 의 지침을 따르세요.	클라우드 관리자, 시스템 관리자

OSAM 구성 및 VM 마이그레이션

작업	설명	필요한 기술
HCX Sentinel을 설치합니다.	<p>Linux에 Sentinel을 설치하는 방법:</p> <ol style="list-style-type: none"> 1. HCX 커넥터용 vCenter Server에서 상호 연결, 다중 사이트 서비스 메시, Sentinel 관리를 선택합니다. 2. Linux 번들 다운로드를 선택합니다. 3. Linux 시스템에 Sentinel 에이전트를 설치합니다. <p>자세한 내용은 VMware 설명서의 HCX Sentinel Agent 소프트웨어 다운로드 및 설치를 참조하세요.</p>	클라우드 관리자

작업	설명	필요한 기술
VM을 마이그레이션합니다.	<p>그룹(모빌리티 그룹이라고 함)으로 VM을 마이그레이션하려면 다음 단계를 따르세요.</p> <ol style="list-style-type: none"> 1. vSphere Client의 HCX 플러그인에서 서비스, 마이그레이션을 선택합니다. 2. [Migrate]를 선택합니다. 3. 비 vSphere 인벤토리, 원격 연결을 선택합니다. 그러면 HCX Sentinel을 설치한 VM 목록이 표시됩니다. 4. 그룹 이름에는 VM에 대해 생성하려는 모빌리티 그룹의 이름을 입력합니다. 5. 마이그레이션할 VM을 선택한 다음 추가를 선택하여 모빌리티 그룹에 추가합니다. 6. 각 VM의 경우: <ol style="list-style-type: none"> a. 대상 컴퓨팅 컨테이너를 선택합니다. b. 대상 스토리지를 선택합니다. c. 마이그레이션 프로파일을 선택합니다. d. 다음과 같이 대상 폴더를 선택합니다. 7. 마이그레이션 프로세스를 시작하려면 이동을 선택하세요. 	클라우드 관리자

작업	설명	필요한 기술
	<p>HCX는 마이그레이션이 시작 되기 전에 VM 선택을 검증합니다.</p> <p>자세한 내용은 VMware 설명서의 모빌리티 그룹을 사용한 가상 시스템 마이그레이션 및 모빌리티 그룹을 사용한 마이그레이션 모니터링 및 추정을 참조를 참조하십시오.</p>	

관련 리소스

VMware 설명서:

- [VMware HCX 사용 설명서](#)
- [설치 체크리스트 B - VMC SDDC 대상 환경을 사용한 HCX](#)
- [AWS의 VMware Cloud에서의 VMware HCX](#)
- [AWS의 VMware Cloud용 HCX OS Assisted Migration](#)
- [VMware HCX 4.2.1 릴리스 노트](#)

VMware Aria Operations for Logs를 사용하여 VMware Cloud on AWS에서 스플링크로 로그를 전송합니다.

제작: 디팍 쿠마르 (AWS) 와 피터 피테라 (AWS)

환경: 프로덕션	출처: VMware 클라우드 온 AWS 로그 및 이벤트	대상: 스플링크 온프레미스 엔드포인트
R 유형: 재배치	워크로드: 기타 모든 워크로드	기술: 하이브리드 클라우드, 인프라, 마이그레이션
AWS 서비스: AWS의 VMware Cloud		

요약

이 패턴은 VMware Aria Operations for Logs를 사용하여 VMware Cloud on AWS 이벤트 또는 로그를 syslog 또는 Splunk와 같은 HTTP 엔드포인트로 전달하는 방법을 설명합니다.

로그용 VMware Aria Operations for Logs는 VMware Cloud on 환경에서 가시성을 높이고 문제 해결을 가속화하는 로그 분석 도구입니다. AWS VMware Cloud의 로그 또는 이벤트 전체 또는 일부를 syslog 또는 HTTP 엔드포인트로 AWS 전송하도록 이 도구를 구성할 수 있습니다. 엔드포인트는 SaaS (서비스형 소프트웨어) 엔드포인트 또는 Splunk와 같은 온프레미스 엔드포인트일 수 있습니다. (이 패턴은 Splunk에 대한 지침을 제공합니다.) 로그를 위한 VMware Aria 오퍼레이션에 대한 자세한 내용은 [VMware](#) 설명서를 참조하십시오.

VMware Cloud AWS on은 모든 규모의 기업이 다양한 솔루션을 사용하여 VMware vSphere 기반 클라우드 환경에서 워크로드를 실행할 수 있도록 하는 pay-as-you-go (온디맨드) 서비스입니다. AWS 서비스 소프트웨어 정의 데이터 센터 (SDDC) 클러스터당 최소 2개의 호스트로 시작하여 프로덕션 환경에서 클러스터당 최대 16개의 호스트로 확장할 수 있습니다. 자세한 내용은 [VMware](#) Cloud on 웹 사이트를 참조하십시오. AWS SDCC에 대한 자세한 내용은 VMware 설명서의 [소프트웨어 정의 데이터 센터](#)를 참조하세요.

사전 조건 및 제한 사항

사전 조건

- Splunk, 온프레미스로 구성

제한 사항

로그용 VMware Aria Operations for Logs의 무료 평가판 구독에 가입할 수 있습니다. 이 구독은 30일 동안 유효하며 다음과 같은 제한이 있습니다.

- 전달할 수 있는 최대 로그 크기: 하루 50GB 로그
- 만들 수 있는 최대 로그 전달 구성 수: 10
- 활성화할 수 있는 최대 로그 전달 구성 수: 5

모든 서비스 기능을 이용하려면 프리미엄 구독으로 업그레이드해야 합니다.

평가판 및 프리미엄 구독에 대한 자세한 내용은 VMware 설명서에서 [VMware Aria Operations for Logs \(SaaS\) 구독 및 청구를](#) 참조하십시오. 사용 제한에 대한 자세한 내용은 VMware 설명서의 [기능 사용 제한을](#) 참조하십시오.

제품 버전

- AWS SDDC 기반 VMware 클라우드 버전 1.24
- VMware Aria 오퍼레이션즈 포 로그 버전 8.10
- 온프레미스 스플링크 버전 9.x

아키텍처

소스 기술 스택

- VM웨어 클라우드 온 AWS
- VMware Aria Operations for Logs

대상 기술 스택

- 온프레미스 스플링크

대상 아키텍처

다음 다이어그램은 기업 데이터 센터와 VMware Cloud on의 VMware Aria Operations for Logs 간의 연결을 보여줍니다. AWS

도구

- [VMware Cloud AWS on](#)은 VMware가 공동으로 개발한 AWS 통합 클라우드 제품입니다.
- [VMware Aria Operations for Logs](#)는 VMware Cloud on을 위한 로그 분석 및 문제 해결 도구입니다. AWS

에픽

SDDC를 배포하고 로그를 위한 VMware Aria Operation을 활성화하십시오.

작업	설명	필요한 기술
SDDC에 VMware 클라우드를 배포하십시오. AWS	규범적 지침의 VMware 클라우드 온을 사용하여 VMware SDDC AWS 배포에 나와 있는 지침을 따르십시오. AWS AWS	클라우드 아키텍트, 클라우드 관리자
로그를 위한 VMware Aria 오퍼레이션에 가입하십시오.	자세한 지침은 VMware 설명서를 참조하십시오.	클라우드 아키텍트

클라우드 프록시 배포

작업	설명	필요한 기술
클라우드 프록시 배포.	Splunk의 온프레미스 인스턴스로 로그를 전달하려면 로그용 VMware Aria Operations for Logs용 클라우드 프록시를 추가해야 합니다. 이 프록시는 온프레미스 데이터 센터로부터 정보를 수신하여 분석을 위해	클라우드 관리자, 클라우드 아키텍트

작업	설명	필요한 기술
	<p>VMware Aria Operations for Logs로 전송합니다.</p> <p>클라우드 프록시를 다운로드하고 설치하려면:</p> <ol style="list-style-type: none"> 1. 온프레미스 환경과 VMware Cloud ON 사이에 포트 443, 22, 514가 열려 있는지 확인하십시오. AWS포트를 추가하려면 1514/TCP 또는 6514/TCP를 사용할 수 있습니다. 포트에 대한 자세한 내용은 VMware 설명서에서 VMware Aria 운영 로그 방화벽 권장 사항을 참조하십시오. 2. 로그용 VMware Aria 오퍼레이션에 로그인하십시오. 3. 홈 페이지의 위젯에서 컬렉터 추가를 선택합니다. 4. 클라우드 프록시 가상 어플라이언스 화면에서 토큰 키를 복사합니다. 다음 단계를 완료하려면 24시간 이내에 이 키를 사용해야 합니다. 5. OVA 파일의 다운로드 링크를 선택합니다. 6. VMware vSphere 웹 클라이언트로 이동하여 클러스터를 선택한 다음 OVF 템플릿 배포를 선택합니다. 	

작업	설명	필요한 기술
	<p>7. 키를 입력하라는 메시지가 표시되면 4단계에서 복사한 토큰 키를 붙여넣습니다.</p> <p>8. Finish를 선택하여 클라우드 프록시를 설치합니다.</p>	

온프레미스 Splunk 엔드포인트로 로그 전달

작업	설명	필요한 기술
로그 전달을 구성합니다.	<p>로그를 Splunk 엔드포인트로 전달하려면:</p> <ol style="list-style-type: none"> 로그용 VMware Aria 오퍼레이션에 로그인하십시오. 로그 관리로 이동합니다. 로그 전달을 선택합니다. 새 구성을 선택하고 다음 설정을 완료합니다. <ul style="list-style-type: none"> 로그 전달 구성의 이름을 입력합니다. 목적지로 온-프레미스를 선택합니다. 클라우드 프록시의 경우 이전에 설치한 클라우드 프록시를 선택합니다. 엔드포인트 유형에서 TCP를 선택합니다. 엔드포인트 URL의 경우 온프레미스 Splunk URL을 다음 형식으로 제공하십시오. 	

작업	설명	필요한 기술
	<div style="border: 1px solid #ccc; border-radius: 10px; padding: 10px; margin-bottom: 10px;"> <pre>tcp://x.x.x.x (your Splunk IP address): 514</pre> </div> <ul style="list-style-type: none"> • (선택 사항) 태그의 경우 쿼리를 용이하게 하기 위해 태그 이름과 값을 지정할 수 있습니다. • 모든 로그에 적용 또는 특정 로그에 적용을 선택합니다. 모든 VMware Cloud on AWS 로그를 Splunk로 보내려면 [모든 로그에 적용] 을 선택합니다. <p>5. 확인을 선택합니다.</p> <p>6. 저장을 선택합니다.</p> <p>자세한 내용은 VMware 설명서의 로그용 VMware Aria 오퍼레이션의 로그 전달을 참조하십시오.</p>	

관련 리소스

- [웹 사이트의 VMware 클라우드 AWS](#)
- [소프트웨어 정의 데이터 센터 정보 \(VMware 설명서\)](#)
- VMware 클라우드 [온을 사용하여 VMware SDDC를 배포하십시오 AWS](#) (지침 지침). AWSAWS
- VMware HCX ([규범 지침](#)) 를 사용하여 VMware 클라우드로 워크로드를 마이그레이션하십시오. AWS AWS
- 하이브리드 연결 [모드를 사용하여 VMware Cloud로의 데이터 센터 확장 구성 \(규범 지침\) AWSAWS](#)

AWS CDK를 사용하여 Amazon ECS Anywhere에서 하이브리드 워크로드를 위한 CI/CD 파이프라인을 설정하고 GitLab

작성자: Dr. Rahul Sharad Gaikwad(AWS)

코드 amazon-ecs-anywhere-cicd리포지토리 : - pipeline-cdk-sample	환경: PoC 또는 파일럿	기술: 하이브리드 클라우드, 컨테이너 및 마이크로서비스, 인프라, DevOps
워크로드: 오픈 소스	AWS 서비스: AWS CDK, AWS, 아마존 ECS CodePipeline, AWS Systems Manager, AWS CodeCommit	

요약

Amazon ECS Anywhere는 Amazon Elastic Container Service(Amazon ECS)의 확장입니다. 온프레미스 서버 또는 가상 머신(VM)과 같은 외부 인스턴스를 Amazon ECS 클러스터에 등록하도록 지원합니다. 이 기능은 비용을 절감하고 복잡한 로컬 컨테이너 오케스트레이션 및 운영을 완화하는 데 도움이 됩니다. ECS Anywhere를 사용하여 온프레미스와 클라우드 환경 모두에서 컨테이너 애플리케이션을 배포하고 실행할 수 있습니다. 따라서 팀이 여러 도메인과 기술을 배우거나 복잡한 소프트웨어를 자체적으로 관리할 필요가 없습니다.

이 패턴은 Amazon Web Services (AWS) 클라우드 개발 키트 (AWS CDK) 스택을 사용하여 Amazon ECS Anywhere 인스턴스로 Amazon ECS 클러스터를 프로비저닝하는 step-by-step 접근 방식을 설명합니다. 그런 다음 AWS를 CodePipeline 사용하여 지속적 통합 및 지속적 배포 (CI/CD) 파이프라인을 설정합니다. 그런 다음 GitLab 코드 리포지토리를 AWS에 CodeCommit 복제하고 컨테이너식 애플리케이션을 Amazon ECS 클러스터에 배포합니다.

이 패턴은 온프레미스 인프라를 사용하여 컨테이너 애플리케이션을 실행하고 애플리케이션 코드 베이스를 관리하는 데 사용하는 사용자를 돕기 위해 설계되었습니다. 기존의 온프레미스 인프라를 방해하지 않고 AWS Cloud 서비스를 사용하여 이러한 워크로드를 관리할 수 있습니다.

사전 조건 및 제한 사항

사전 조건

- 활성 상태의 AWS 계정
- 온프레미스 인프라에서 실행되는 컨테이너 애플리케이션입니다.
- 애플리케이션 코드베이스를 관리하는 GitLab 리포지토리입니다. 자세한 내용은 [리포지토리 \(GitLab\)](#) 를 참조하십시오.
- AWS Command Line Interface(AWS CLI), 설치 및 구성됨 자세한 내용은 [최신 버전의 AWS CLI 설치 또는 업데이트](#)(AWS CLI 설명서)를 참조하세요.
- AWS CDK Toolkit, 전역적으로 설치 및 구성됨. 자세한 내용은 [AWS CDK 설치](#)(AWS CDK 설명서)를 참조하세요.
- npm, AWS CDK용으로 설치 및 구성되었습니다. TypeScript 자세한 내용은 [Node.js 및 npm 다운로드 및 설치](#)(npm 설명서)를 참조하세요.

제한 사항

- 제한 사항 및 고려 사항은 Amazon ECS 설명서의 [외부 인스턴스\(Amazon ECS Anywhere\)](#)를 참조하세요.

제품 버전

- AWS CDK Toolkit 버전 2.27.0 이상
- npm 버전 7.20.3 이상
- Node.js 버전 16.6.1 이상

아키텍처

대상 기술 스택

- AWS CDK
- AWS CloudFormation
- AWS CodeBuild
- AWS CodeCommit
- AWS CodePipeline
- Amazon ECS Anywhere
- Amazon Elastic Container Registry (Amazon ECR)
- AWS Identity and Access Management(IAM)

- AWS Systems Manager
- GitLab 리포지토리

대상 아키텍처

이 다이어그램은 다음과 같이 이 패턴에 설명된 두 가지 기본 워크플로우를 나타냅니다. 하나는 Amazon ECS 클러스터를 프로비저닝하는 것이고 다른 하나는 CI/CD 파이프라인을 설정하고 배포하는 CI/CD 파이프라인을 설정하는 것입니다.

1. Amazon ECS 클러스터 프로비저닝

- 첫 번째 AWS CDK 스택을 배포하면 AWS에 CloudFormation 스택이 생성됩니다.
- 이 CloudFormation 스택은 Amazon ECS 클러스터 및 관련 AWS 리소스를 프로비저닝합니다.
- Amazon ECS 클러스터에 외부 인스턴스를 등록하려면 VM에 AWS Systems Manager Agent(SSM Agent)를 설치하고 해당 VM을 AWS Systems Manager 관리형 인스턴스로 등록해야 합니다.
- 또한 VM을 Amazon ECS 클러스터에 외부 인스턴스로 등록하려면 VM에 Amazon ECS 컨테이너 에이전트와 Docker를 설치해야 합니다.
- Amazon ECS 클러스터에 외부 인스턴스를 등록하고 구성하면, 외부 인스턴스로 등록된 VM에서 여러 컨테이너를 실행할 수 있습니다.
- Amazon ECS 클러스터는 활성 상태이며 컨테이너를 통해 애플리케이션 워크로드를 실행할 수 있습니다. Amazon ECS Anywhere 컨테이너 인스턴스는 온프레미스 환경에서 실행되지만 클라우드의 Amazon ECS 클러스터와 연결되어 있습니다.

2. CI/CD 파이프라인 설정 및 배포

- 두 번째 AWS CDK 스택을 배포하면 AWS에 또 다른 CloudFormation 스택이 생성됩니다.
- 이 CloudFormation 스택은 파이프라인을 CodePipeline 비롯한 관련 AWS 리소스를 제공합니다.
- 애플리케이션 코드 변경 내용을 온프레미스 GitLab 리포지토리로 푸시하고 병합합니다.
- GitLab 리포지토리는 리포지토리에 자동으로 복제됩니다. CodeCommit
- CodeCommit 리포지토리 업데이트가 자동으로 시작됩니다. CodePipeline
- CodePipeline 배포 가능한 애플리케이션 빌드에서 코드를 CodeCommit 복사하고 빌드인을 생성합니다. CodeBuild
- CodePipeline CodeBuild 빌드 환경의 Docker 이미지를 생성하여 Amazon ECR 리포지토리에 푸시

- h. CodePipeline Amazon ECR 리포지토리에서 컨테이너 이미지를 가져오는 CodeDeploy 작업을 시작합니다.
- i. CodePipeline Amazon ECS 클러스터에 컨테이너 이미지를 배포합니다.

자동화 및 규모 조정

이 패턴은 AWS CDK를 코드형 인프라(IaC) 도구로 사용하여 이 아키텍처를 구성하고 배포합니다. AWS CDK를 사용하면 AWS 리소스를 오케스트레이션하고 Amazon ECS Anywhere 및 CI/CD 파이프라인을 설정할 수 있습니다.

도구

서비스

- [AWS Cloud Development Kit\(AWS CDK\)](#)는 AWS 클라우드 인프라를 코드로 정의하고 프로비저닝하는 데 도움이 되는 소프트웨어 개발 프레임워크입니다.
- [CodeCommitAWS](#)는 자체 소스 제어 시스템을 관리할 필요 없이 Git 리포지토리를 비공개로 저장하고 관리할 수 있는 버전 제어 서비스입니다.
- [AWS](#)는 소프트웨어 릴리스의 여러 단계를 신속하게 모델링 및 구성하고 소프트웨어 변경 사항을 지속적으로 릴리스하는 CodePipeline 데 필요한 단계를 자동화하도록 지원합니다.
- [AWS Command Line Interface\(AWS CLI\)](#)는 명령줄 셸에서 명령을 사용하여 AWS 서비스와 상호 작용할 수 있는 오픈 소스 도구입니다.
- [Amazon Elastic Container Registry\(Amazon ECR\)](#)는 안전하고 확장 가능하고 신뢰할 수 있는 관리형 컨테이너 이미지 레지스트리 서비스입니다.
- [Amazon Elastic Container Service\(Amazon ECS\)](#)는 클러스터에서 컨테이너를 실행, 중지 및 관리하는 데 도움이 되는 빠르고 확장 가능한 컨테이너 관리 서비스입니다. 또한 이 패턴은 온프레미스 서버 또는 VM을 Amazon ECS 클러스터에 등록하도록 지원하는 [Amazon ECS Anywhere](#)를 사용합니다.

기타 도구

- [Node.js](#) 는 확장 가능한 네트워크 애플리케이션을 구축하도록 설계된 이벤트 기반 JavaScript 런타임 환경입니다.
- [npm](#)은 Node.js 환경에서 실행되는 소프트웨어 레지스트리로, 패키지를 공유 또는 대여하고 개인 패키지의 배포를 관리하는 데 사용됩니다.

- [Vagrant](#)는 휴대용 가상 소프트웨어 개발 환경을 구축하고 유지 관리하기 위한 오픈 소스 유틸리티입니다. 이 패턴은 데모용으로 Vagrant를 사용하여 온프레미스 VM을 만듭니다.

코드 리포지토리

이 패턴의 코드는 [AWS CDK 리포지토리를 사용하는 Amazon ECS Anywhere의 GitHub CI/CD 파이프라인에서](#) 사용할 수 있습니다.

모범 사례

이 패턴을 배포할 때는 다음과 같은 모범 사례를 고려해 보십시오.

- [AWS CDK를 사용한 클라우드 인프라 개발 및 배포 모범 사례](#)
- [AWS CDK를 사용한 클라우드 애플리케이션 개발 모범 사례](#)(AWS 블로그 게시물)

에픽

AWS CDK 구성 확인

작업	설명	필요한 기술
AWS CDK 버전 확인	<p>다음 명령을 입력하여 AWS CDK Toolkit의 버전을 확인합니다.</p> <pre>cdk --version</pre> <p>이 패턴을 사용하려면 버전 2.27.0 이상이 필요합니다. 이전 버전이 있을 경우 AWS CDK 설명서의 지침에 따라 업데이트하세요.</p>	DevOps 엔지니어
npm 버전을 확인합니다.	<p>다음 명령을 입력하여 npm 버전을 확인합니다.</p> <pre>npm --version</pre>	DevOps 엔지니어

작업	설명	필요한 기술
	<p>이 패턴을 사용하려면 버전 7.20.3 이상이 필요합니다. 이전 버전이 있을 경우 npm 설명서의 지침에 따라 업데이트하세요.</p>	
<p>AWS 보안 인증 설정</p>	<p>aws configure 명령을 입력하고 프롬프트를 따라 AWS 보안 인증을 설정합니다.</p> <pre data-bbox="594 646 1027 1167"> \$aws configure AWS Access Key ID [None]: <your-access-key-ID> AWS Secret Access Key [None]: <your-secret-access-key> Default region name [None]: <your-Region-name> Default output format [None]: </pre>	<p>DevOps 엔지니어</p>

AWS CDK 환경 부트스트랩

작업	설명	필요한 기술
<p>AWS CDK 코드 리포지토리를 복제합니다.</p>	<p>1. 다음 명령을 입력하여 이 패턴의 AWS CDK를 사용하여 Amazon ECS Anywhere용 CI/CD 파이프라인 리포지토리를 복제합니다.</p> <pre data-bbox="630 1717 1027 1852"> git clone https://github.com/aws-samples/amazon-ecs-any </pre>	<p>DevOps 엔지니어</p>

작업	설명	필요한 기술
	<pre>where-cicd-pipelin e-cdk-sample.git</pre> <p>2. 다음 명령을 입력하여 복제된 디렉터리로 이동합니다.</p> <pre>cd amazon-ecs-anywher e-cicd-pipeline-cd k-sample</pre>	
<p>환경을 부트스트랩합니다.</p>	<p>다음 명령을 입력하여 사용하려는 계정과 AWS 지역에 CloudFormation 템플릿을 배포합니다.</p> <pre>cdk bootstrap <account- number>/<Region></pre> <p>자세한 내용은 AWS CDK 설명서의 부트스트래핑을 참조하세요.</p>	<p>DevOps 엔지니어</p>

Amazon ECS Anywhere에 대한 인프라 구축 및 배포

작업	설명	필요한 기술
<p>패키지 종속성을 설치하고 파일을 컴파일합니다. TypeScript</p>	<p>다음 명령을 입력하여 패키지 종속성을 설치하고 TypeScript 파일을 컴파일합니다.</p> <pre>\$cd EcsAnywhereCdk \$npm install \$npm fund</pre> <p>이 명령은 샘플 저장소의 모든 패키지를 설치합니다. 자세</p>	<p>DevOps 엔지니어</p>

작업	설명	필요한 기술
	<p>한 내용은 npm 설명서에 있는 npm ci와 npm install을 참조하세요. 이러한 명령을 입력할 때 누락된 패키지와 관련된 오류가 발생하는 경우 이 패턴의 문제 해결 섹션을 참고하십시오.</p>	
<p>프로젝트를 빌드합니다.</p>	<p>프로젝트 코드를 빌드하려면 다음 명령을 입력하세요.</p> <pre data-bbox="594 646 1027 730">npm run build</pre> <p>프로젝트 구축 및 배포에 대한 자세한 내용은 AWS CDK 설명서의 첫 AWS CDK 앱을 참조하세요.</p>	<p>DevOps 엔지니어</p>
<p>Amazon ECS Anywhere 인프라 스택을 배포합니다.</p>	<ol style="list-style-type: none"> 다음 명령을 입력하여 스택을 나열합니다. <pre data-bbox="630 1108 1027 1192">\$cdk list</pre> 출력이 EcsAnywhereInfraStack 및 ECSAnywherePipelineStack 스택을 반환하는지 확인합니다. 다음 명령을 입력하여 EcsAnywhereInfraStack 스택을 배포합니다. <pre data-bbox="630 1623 1027 1745">\$cdk deploy EcsAnywhereInfraStack</pre> 	<p>DevOps 엔지니어</p>

작업	설명	필요한 기술
스택 생성 및 출력을 확인합니다.	<ol style="list-style-type: none"> 1. AWS 관리 콘솔에 로그인하고 https://console.aws.amazon.com/cloudformation/ 에서 CloudFormation 콘솔을 엽니다. 2. 스택 페이지에서 EcsAnywhereInfrastructure 스택을 선택합니다. 3. 스택 상태가 CREATE_IN_PROGRESS 또는 CREATE_COMPLETE 인지 확인합니다. <p>Amazon ECS 클러스터를 설정하는 데 시간이 걸릴 수 있습니다. 스택 생성이 완료 될 때까지는 진행하지 마십시오.</p>	DevOps 엔지니어

온프레미스 VM 설정

작업	설명	필요한 기술
VM을 설정합니다.	Vagrantfile이 있는 루트 디렉터리에서 <code>vagrant up</code> 명령을 입력하여 Vagrant VM을 생성합니다. 자세한 내용은 Vagrant 설명서 를 참조하세요.	DevOps 엔지니어
VM을 외부 인스턴스로 등록합니다.	1. <code>vagrant ssh</code> 명령을 사용하여 Vagrant VM에 로그인합니다. 자세한 내용은	DevOps 엔지니어

작업	설명	필요한 기술
	<p>Vagrant 설명서를 참조하세요.</p> <p>2. AWS CLI 설치 지침에 따라 다음 명령을 입력하여 VM에 AWS CLI를 설치합니다.</p> <pre data-bbox="634 485 1029 1356"> \$ curl "https:// awscli.amazonaws.c om/awscli-exe-linu x-x86_64.zip" \ > -o "awscliv2.zip" \$sudo apt install unzip \$unzip awscliv2.zip \$sudo ./aws/install \$aws configure AWS Access Key ID [None]: <your-acc ess-key-ID> AWS Secret Access Key [None]: <your-sec ret-access-key> Default region name [None]: <your-Reg ion-name> Default output format [None]: </pre> <p>1. AWS Systems Manager에 VM을 등록하고 외부 인스턴스를 활성화하는 데 사용할 수 있는 활성화 코드와 ID를 생성합니다. 이 명령의 출력에는 활성화 ID 및 활성화 코드 값이 포함됩니다.</p> <pre data-bbox="634 1787 1029 1877"> aws ssm create-ac tivation \ </pre>	

작업	설명	필요한 기술
	<pre data-bbox="634 212 997 426">> --iam-role EcsAnywhereInstanc eRole \ > tee ssm-activ ation.json</pre> <p data-bbox="630 464 1019 594">이 명령을 실행할 때 오류가 발생하는 경우 문제 해결 섹션을 참고하십시오.</p> <p data-bbox="589 615 1027 699">2. 활성화 ID 및 코드 값 내보내기:</p> <pre data-bbox="634 737 997 1014">export ACTIVATIO N_ID=<activation-I D> export ACTIVATIO N_CODE=<activation- code></pre> <p data-bbox="589 1031 1027 1115">3. VM에 설치 스크립트를 다운로드합니다.</p> <pre data-bbox="634 1157 997 1507">curl --proto "https" - o "ecs-anywhere-inst all.sh" \ > "https://amazon-ec s-agent.s3.amazona ws.com/ecs-anywher e-install-latest.s h"</pre> <p data-bbox="589 1524 1027 1608">4. VM에서 설치 스크립트를 실행합니다.</p> <pre data-bbox="634 1650 997 1814">sudo bash ecs-anywh ere-install.sh \ --cluster EcsAnywhe reCluster \</pre>	

작업	설명	필요한 기술
	<pre data-bbox="634 212 1027 468">--activation-id \$ACTIVATION_ID \ --activation-code \$ACTIVATION_CODE \ --region <region-name></pre> <p data-bbox="591 533 1024 995">이렇게 하면 VM이 Amazon ECS Anywhere 외부 인스턴스로 설정되고 해당 인스턴스가 Amazon ECS 클러스터에 등록됩니다. 자세한 내용은 Amazon ECS 설명서의 클러스터에 외부 인스턴스 등록을 참조하세요. 문제가 발생하는 경우 문제 해결 섹션을 참고하십시오.</p>	
<p>Amazon ECS Anywhere와 외부 VM의 상태를 확인합니다.</p>	<p>VM이 Amazon ECS 컨트롤 플레인에 연결되어 실행 중인지 확인하려면 다음 명령을 사용하세요.</p> <pre data-bbox="591 1262 1027 1499">\$aws ssm describe-instance-information \$aws ecs list-container-instances --cluster \$CLUSTER_NAME</pre>	<p>DevOps 엔지니어</p>

CI/CD 파이프라인 배포

작업	설명	필요한 기술
<p>CodeCommit 리포지토리에 브랜치를 생성하세요.</p>	<p>main 리포지토리에 대한 첫 번째 커밋을 생성하여</p>	<p>DevOps 엔지니어</p>

작업	설명	필요한 기술
	<p>CodeCommit 리포지토리에 이름이 지정된 브랜치를 생성합니다. AWS 설명서에 따라 커밋을 생성할 수 CodeCommit 있습니다. 다음 명령은 예제입니다.</p> <pre data-bbox="592 520 1031 1117">aws codecommit put-file \ --repository-name EcsAnywhereRepo \ --branch-name main \ --file-path README.md \ --file-content "Test" \ --name "Dev Ops" \ --email "devops@example.com" \ --commit-message "Adding README."</pre>	

작업	설명	필요한 기술
<p>리포지토리 미러링을 설정합니다.</p>	<p>외부 소스와 GitLab 리포지토리를 미러링할 수 있습니다. 소스로 사용할 리포지토리를 선택할 수 있습니다. 브랜치, 태그, 커밋은 자동으로 동기화됩니다. 애플리케이션을 호스팅하는 GitLab 리포지토리와 리포지토리 간에 푸시 미러를 설정하십시오. CodeCommit 자세한 지침은 to에서 푸시 미러 설정 CodeCommit (GitLab 설명서) GitLab 을 참조하십시오.</p> <p>참고: 기본적으로 미러링은 리포지토리를 자동으로 동기화합니다. 리포지토리를 수동으로 업데이트하려면 미러 업데이트 (GitLab 설명서) 를 참조하십시오.</p>	<p>DevOps 엔지니어</p>
<p>CI/CD 파이프라인 스택을 배포합니다.</p>	<p>다음 명령을 입력하여 EcsAnywherePipelineStack 스택을 배포합니다.</p> <pre data-bbox="597 1318 1026 1436">\$cdk deploy EcsAnywherePipelineStack</pre>	<p>DevOps 엔지니어</p>

작업	설명	필요한 기술
<p>CI/CD 파이프라인을 테스트합니다.</p>	<ol style="list-style-type: none"> 1. 애플리케이션 코드를 변경하여 소스 온-프레미스 GitLab 리포지토리에 푸시합니다. 자세한 내용은 푸시 옵션 (GitLab 설명서) 을 참조하십시오. 예를 들어 <code>../application/index.html</code> 파일을 편집하여 애플리케이션 버전 값을 업데이트합니다. 2. 코드가 CodeCommit 리포지토리에 복제되면 CI/CD 파이프라인이 시작됩니다. 다음 중 하나를 수행합니다. <ul style="list-style-type: none"> • 자동 미러링을 사용하여 GitLab 리포지토리를 리포지토리와 동기화하는 경우 다음 단계를 계속 진행하세요. CodeCommit • 수동 미러링을 사용하는 경우 미러 업데이트 (설명서) 의 지침에 따라 애플리케이션 코드 변경 내용을 CodeCommit 리포지토리로 푸시하십시오. GitLab 3. 로컬 컴퓨터의 웹 브라우저에서 http://localhost:80을 입력합니다. 이렇게 하면 포트 80이 Vagrantfile의 로컬 호스트로 전달되기 때문에 NGINX 웹 페이지가 열립니다. 업데이트된 애플리케이션 버전 값을 볼 수 있는지 확인하십시오. 이렇게 하면 	<p>DevOps 엔지니어</p>

작업	설명	필요한 기술
	<p>파이프라인 및 이미지 배포가 검증됩니다.</p> <p>4. (선택 사항) AWS Management Console에서 배포를 확인하려면 다음을 수행합니다.</p> <ol style="list-style-type: none"> https://console.aws.amazon.com/ecs/에서 Amazon ECS 콘솔을 엽니다. 탐색 모음에서 사용할 리전을 선택합니다. 탐색 창에서 클러스터 (Clusters)를 선택합니다. 클러스터 페이지에서 클러스터를 선택합니다. EcsAnywhereCluster 작업 정의를 선택합니다. 컨테이너가 실행 중인지 확인합니다. 	

정리

작업	설명	필요한 기술
리소스를 정리하고 삭제합니다.	<p>이 패턴을 살펴본 후에는 생성한 proof-of-concept 리소스를 제거해야 합니다. 정리하려면 다음 명령을 입력합니다.</p> <pre>\$cdk destroy EcsAnywherePipelineStack</pre>	DevOps 엔지니어

작업	설명	필요한 기술
	<pre>\$cdk destroy EcsAnywhereInfraStack</pre>	

문제 해결

문제	Solution
<p>패키지 종속 항목을 설치할 때 누락된 패키지와 관련된 오류가 발생합니다.</p>	<p>다음 명령 중 하나를 입력하여 누락된 패키지를 해결하십시오.</p> <pre>\$npm ci</pre> <p>또는</p> <pre>\$npm install -g @aws-cdk/<package_name></pre>
<p>VM에서 <code>aws ssm create-activation</code> 명령을 실행하면 다음 오류가 수신됩니다.</p> <pre>An error occurred (ValidationException) when calling the CreateActivation operation: Nonexistent role or missing ssm service principal in trust policy: arn:aws:iam::000000000000:role/EcsAnywhereInstanceRole</pre>	<p>EcsAnywhereInfraStack 스택이 완전히 배포되지 않았고 이 명령을 실행하는 데 필요한 IAM 역할이 아직 생성되지 않았습니다. CloudFormation 콘솔에서 스택 상태를 확인합니다. 상태가 CREATE_COMPLETE 로 변경된 후 명령을 재시도합니다.</p>
<p>Amazon ECS 상태 확인이 UNHEALTHY 를 반환되고 Amazon ECS 콘솔에 있는 클러스터의 서비스 섹션에 다음 오류가 표시됩니다.</p> <pre>service EcsAnywhereService was unable to place a task because no container instance met all</pre>	<p>다음 명령을 입력하여 Vagrant VM에서 Amazon ECS 에이전트를 다시 시작합니다.</p> <pre>\$vagrant ssh \$sudo systemctl restart ecs \$sudo systemctl status ecs</pre>

문제	Solution
of its requirements. Reason: No Container Instances were found in your cluster.	

관련 리소스

- [Amazon ECS Anywhere 마케팅 페이지](#)
- [Amazon ECS Anywhere 설명서](#)
- [Amazon ECS Anywhere 데모\(동영상\)](#)
- [Amazon ECS Anywhere 워크숍 샘플 GitHub \(\)](#)
- [리포지토리 미러링 \(설명서\) GitLab](#)

더 많은 패턴

- [AWS Transit Gateway를 사용하여 리전 간 피어링 설정을 자동화하십시오.](#)
- [AWS CDK로 Amazon ECS Anywhere를 설정하여 온프레미스 컨테이너 애플리케이션을 관리](#)
- [WanDisco 마이그레이션기를 사용하여 하둡 데이터를 Amazon S3로 마이그레이션합니다. LiveData](#)
- [PowerCLI를 사용하여 HCX 자동화를 통해 VMware VM 마이그레이션](#)
- [VMware HCX를 사용하여 워크로드를 AWS의 VMware Cloud로 마이그레이션합니다.](#)
- [F5에서 AWS의 Application Load Balancer로 마이그레이션할 때 HTTP 헤더를 수정](#)
- [BMC Discovery 쿼리를 사용하여 마이그레이션 계획을 위한 마이그레이션 데이터 추출](#)
- [인프라 코드의 테스트 기반 개발을 위해 Serverspec을 사용합니다](#)

인프라

주제

- [Session Manager 및 Amazon EC2 인스턴스 연결을 사용한 Bastion Host 액세스](#)
- [AWS Managed Microsoft AD와 온프레미스 Microsoft Active Directory를 사용하여 DNS 확인 중앙 집중화](#)
- [Amazon CloudWatch 옵저버빌리티 액세스 관리자를 사용하여 모니터링을 중앙 집중화합니다.](#)
- [시작 시 EC2 인스턴스에 필수 태그가 있는지 확인](#)
- [Session Manager를 사용하여 Amazon EC2 인스턴스에 연결](#)
- [AWS를 지원하지 않는 AWS 지역에 파이프라인을 생성하십시오. CodePipeline](#)
- [프라이빗 고정 IP를 사용하여 Amazon EC2에 Cassandra 클러스터를 배포하여 리밸런싱 방지](#)
- [AWS Transit Gateway Connect를 사용하여 VRF를 AWS로 확장](#)
- [AWS KMS 키의 키 상태가 변경될 때 Amazon SNS 알림 받기](#)
- [메인프레임 현대화: 마이크로 포커스를 DevOps 활용한 AWS 기반](#)
- [비 워크로드 서브넷을 위한 다중 계정 VPC 설계에서 라우팅 가능한 IP 공간 보존](#)
- [코드 리포지토리를 사용하여 AWS Service Catalog에서 Terraform 제품을 프로비저닝합니다.](#)
- [Amazon SES를 사용하여 단일 이메일 주소로 여러 AWS 계정 등록](#)
- [다중 계정 AWS 환경에서 하이브리드 네트워크를 위한 DNS 확인 설정](#)
- [단일 계정 AWS 환경에서 하이브리드 네트워크를 위한 DNS 확인 설정](#)
- [AWS를 사용하여 Amazon EC2에 UiPath RPA 봇을 자동으로 설정합니다. CloudFormation](#)
- [EnterpriseOne AWS Elastic 재해 복구를 사용하여 Oracle JD Edwards의 재해 복구 설정](#)
- [SAP Pacemaker 클러스터를 ENSA1에서 ENSA2 클러스터로 업그레이드](#)
- [여러 AWS 계정의 VPC에서 일관된 가용 영역 사용](#)
- [Account Factory에 대한 테라폼\(AFT\) 코드를 로컬에서 검증합니다.](#)
- [패턴 더 보기](#)

Session Manager 및 Amazon EC2 인스턴스 연결을 사용한 Bastion Host 액세스

작성자: Piotr Chotkowski(AWS)와 Witold Kowalik(AWS)

코드 리포지토리: [세션 관리자 및 Amazon EC2 Instance Connect를 사용하여 배스천 호스트에 액세스합니다.](#)

환경: PoC 또는 파일럿

기술: 인프라, 클라우드 네이티브, 보안, 자격 증명, 규정 준수, 네트워킹

AWS 서비스: Amazon EC2, AWS Systems Manager, Amazon VPC

요약

점프 박스라고도 하는 Bastion Host는 외부 네트워크에서 프라이빗 네트워크에 있는 리소스에 대한 단일 액세스 지점을 제공하는 서버입니다. 인터넷과 같은 외부 공용 네트워크에 노출된 서버는 무단 액세스로 인한 잠재적 보안 위험을 초래할 수 있습니다. 이러한 서버에 대한 액세스를 보호하고 제어하는 것이 중요합니다.

이 패턴은 [Session Manager](#)와 [Amazon EC2 인스턴스 연결](#)을 사용하여 AWS 계정에 배포된 Amazon Elastic Compute Cloud(Amazon EC2) Bastion Host에 안전하게 연결하는 방법을 설명합니다. Session Manager는 AWS Systems Manager의 기능입니다. 이 패턴의 이점은 다음과 같습니다.

- 배포된 Bastion Host에는 공용 인터넷에 노출되는 개방형 인바운드 포트가 없습니다. 이렇게 하면 잠재적 공격 표면이 줄어듭니다.
- AWS 계정에 장기 Secure Shell(SSH) 키를 저장하고 유지할 필요가 없습니다. 대신 사용자는 Bastion Host에 연결할 때마다 새 SSH 키 페어를 생성합니다. 사용자의 AWS 보안 인증 정보에 연결된 AWS Identity and Access Management(IAM) 정책은 Bastion Host에 대한 액세스를 제어합니다.

수강 대상

이 패턴은 Amazon EC2, Amazon Virtual Private Cloud(VPC) 및 Hashicorp Terraform에 대한 기본적인 이해가 있는 독자를 대상으로 합니다.

사전 조건 및 제한 사항

사전 조건

- 활성 상태의 AWS 계정
- AWS Command Line Interface(AWS CLI) 버전 2, [설치](#) 및 [구성됨](#)
- AWS CLI용 Session Manager 플러그인, [설치됨](#)
- Terraform CLI, [설치됨](#)
- Terraform 상태를 저장하기 위한 원격 백엔드 역할을 하는 Amazon Simple Storage Service(S3) 버킷 및 Amazon DynamoDB 테이블과 같은 Terraform [상태용](#) 스토리지입니다. Terraform 상태에 원격 백엔드를 사용하는 방법에 대한 자세한 내용은 [S3 백엔드](#)(Terraform 설명서)를 참조하세요. S3 백엔드로 원격 상태 관리를 설정하는 코드 샘플은 [remote-state-s3-백엔드](#) (Terraform 레지스트리) 를 참조하십시오. 다음과 같은 요구 사항을 확인합니다.
 - S3 버킷과 DynamoDB 테이블은 같은 AWS 리전에 있어야 합니다.
 - DynamoDB 테이블을 생성할 때 파티션 키는 LockID이어야 하고(대소문자 구분) 파티션 키 유형은 String이어야 합니다. 기타 모든 설정은 기본값을 유지합니다. 자세한 내용은 DynamoDB 설명서의 [프라이머리 키 정보](#) 및 [테이블 생성](#)을 참조하세요.
- SSH 클라이언트, 설치됨

제한 사항

- 이 패턴은 개념 증명(PoC) 또는 추가 개발을 위한 기반으로 사용됩니다. 프로덕션 환경에서는 현재 형식으로 사용하면 안 됩니다. 배포하기 전에 리포지토리의 샘플 코드를 필요 및 사용 사례에 맞게 조정하세요.
- 이 패턴은 대상 Bastion Host가 Amazon Linux 2를 운영 체제로 사용한다고 가정합니다. 다른 Amazon Machine Image(AMI)도 사용할 수 있지만, 다른 운영 체제는 이 패턴의 적용 범위를 벗어납니다.
- 이 패턴에서 Bastion Host는 NAT 게이트웨이 및 인터넷 게이트웨이가 없는 프라이빗 서브넷에 위치합니다. 이 설계는 EC2 인스턴스를 공용 인터넷에서 격리합니다. 인터넷과 통신할 수 있는 특정 네트워크 구성을 추가할 수 있습니다. 자세한 내용은 Amazon VPC 설명서의 [Virtual Private Cloud\(VPC\)를 다른 네트워크에 연결](#)을 참조하세요. 마찬가지로 [최소 권한 원칙](#)에 따라 Bastion Host는 명시적으로 권한을 부여하지 않는 한 AWS 계정의 다른 리소스에 액세스할 수 없습니다. 자세한 내용은 IAM 설명서의 [IAM 정책 생성](#)을 참조하세요.

제품 버전

- CLI version 2
- Terraform 버전 1.3.9

아키텍처

대상 기술 스택

- 단일 퍼블릭 서브넷이 있는 VPC
- 다음 [인터페이스 VPC](#) 엔드포인트:
 - `amazonaws.<region>.ssm` – Systems Manager 서비스의 엔드포인트입니다.
 - `amazonaws.<region>.ec2messages` – Systems Manager는 이 엔드포인트를 사용하여 SSM 에이전트에서 Systems Manager 서비스를 직접 호출합니다.
 - `amazonaws.<region>.ssmmessages` – Session Manager는 이 엔드포인트를 사용하여 보안 데이터 채널을 통해 EC2 인스턴스에 연결합니다.
- Amazon Linux 2를 실행하는 `t3.nano` EC2 인스턴스
- IAM 역할 및 인스턴스 프로파일
- 엔드포인트 및 EC2 인스턴스에 대한 Amazon VPC 보안 그룹 및 보안 그룹 규칙

대상 아키텍처

이 다이어그램은 다음을 보여 줍니다.

1. 사용자는 다음 작업을 수행할 권한이 있는 IAM 역할을 맡습니다.
 - EC2 인스턴스에 인증, 권한 부여 및 연결
 - Session Manager를 사용하여 세션 시작
2. 사용자는 Session Manager를 통해 SSH 세션을 시작합니다.
3. Session Manager는 사용자를 인증하고, 관련 IAM 정책에서 권한을 확인하고, 구성 설정을 확인하고, SSM 에이전트에 메시지를 보내 양방향 연결을 엽니다.
4. 사용자는 Amazon EC2 메타데이터를 통해 SSH 퍼블릭 키를 Bastion Host로 푸시합니다. 이 작업은 각 연결 전에 수행해야 합니다. SSH 퍼블릭 키는 60초 동안 사용할 수 있습니다.
5. Bastion Host는 Systems Manager 및 Amazon EC2의 인터페이스 VPC 엔드포인트와 통신합니다.
6. 사용자는 TLS 1.2로 암호화된 양방향 통신 채널을 사용하여 Session Manager를 통해 Bastion Host에 액세스합니다.

자동화 및 규모 조정

배포를 자동화하거나 이 아키텍처를 확장하는 데 사용할 수 있는 옵션은 다음과 같습니다.

- 지속적인 통합 및 지속적 전달(CI/CD) 파이프라인을 통해 아키텍처를 배포할 수 있습니다.
- 코드를 수정하여 Bastion Host의 인스턴스 유형을 변경할 수 있습니다.
- 코드를 수정하여 여러 Bastion Host를 배포할 수 있습니다. bastion-host/main.tf 파일의 aws_instance 리소스 블록에 count 메타 인수를 추가합니다. 자세한 내용은 [Terraform 설명서](#)를 참조하세요.

도구

서비스

- [AWS Command Line Interface\(AWS CLI\)](#)는 명령줄 셸에서 명령을 사용하여 AWS 서비스와 상호 작용할 수 있는 오픈 소스 도구입니다.
- [Amazon Elastic Compute Cloud\(Amazon EC2\)](#)는 AWS 클라우드에서 확장 가능한 컴퓨팅 용량을 제공합니다. 필요한 만큼 가상 서버를 시작하고 빠르게 스케일 업하거나 스케일 다운할 수 있습니다.
- [AWS Identity and Access Management\(IAM\)](#)를 사용하면 사용자에 대해 인증 및 권한 부여를 제어함으로써 AWS 리소스에 대한 액세스를 안전하게 관리할 수 있습니다.
- [AWS Systems Manager](#)는 AWS 클라우드에서 실행되는 애플리케이션과 인프라를 관리하는 데 도움이 됩니다. 애플리케이션 및 리소스 관리를 간소화하고, 운영 문제의 감지 및 해결 시간을 단축하며, AWS 리소스를 규모에 따라 안전하게 관리하는 데 도움이 됩니다. 이 패턴은 Systems Manager의 기능인 [Session Manager](#)를 사용합니다.
- [Amazon Virtual Private Cloud\(VPC\)](#)를 이용하면 사용자가 정의한 가상 네트워크로 AWS 리소스를 시작할 수 있습니다. 이 가상 네트워크는 사용자의 자체 데이터 센터에서 운영하는 기존 네트워크와 유사하며 AWS의 확장 가능한 인프라를 사용한다는 이점이 있습니다.

기타 도구

- [HashiCorp Terraform](#)은 코드를 사용하여 클라우드 인프라 및 리소스를 프로비저닝하고 관리하는 데 도움이 되는 오픈 소스 IaC (코드형 인프라) 도구입니다. 이 패턴은 [Terraform CLI](#)를 사용합니다.

코드 리포지토리

이 패턴의 코드는 [세션 관리자 및 Amazon EC2 Instance Connect 리포지토리를 사용한 배스천 호스트 GitHub 액세스에서](#) 사용할 수 있습니다.

모범 사례

- 코드의 보안 및 품질을 개선하려면 자동화된 코드 스캔 도구를 사용하는 것이 좋습니다. 이 패턴은 IaC용 정적 코드 분석 도구인 [Checkov](#)를 사용하여 스캔했습니다. 최소한 terraform validate 및 terraform fmt -check -recursive Terraform 명령을 사용하여 기본 유효성 검사 및 형식 검사를 수행하는 것이 좋습니다.
- IaC에 대한 자동 테스트를 추가하는 것이 좋습니다. Terraform 코드를 테스트하기 위한 다양한 접근 방식에 대한 자세한 내용은 Terraform 테스트 ([HashiCorp Terraform 블로그 게시물](#)) 를 참조하십시오.
- 배포 중에 Terraform은 [Amazon Linux 2 AMI](#)의 새 버전이 감지될 때마다 대체 EC2 인스턴스를 사용합니다. 이렇게 하면 패치와 업그레이드를 포함한 새 버전의 운영 체제가 배포됩니다. 배포 일정이 잦지 않은 경우 인스턴스에 최신 패치가 없기 때문에 보안 위험이 발생할 수 있습니다. 자주 업데이트하고 보안 패치를 배포된 EC2 인스턴스에 적용하는 것이 중요합니다. 자세한 내용은 [Amazon EC2 업데이트 관리](#)를 참조하세요.
- 이 패턴은 개념 증명이므로 AmazonSSMManagedInstanceCore와 같은 AWS 관리형 정책을 사용합니다. AWS 관리형 정책은 일반적인 사용 사례를 포함하지만 최소 권한 권한은 부여하지 않습니다. 사용 사례에 따라 이 아키텍처에 배포된 리소스에 대해 최소 권한 권한을 부여하는 사용자 지정 정책을 생성하는 것이 좋습니다. 자세한 내용은 [AWS 관리형 정책으로 시작하고 최소 권한을 향해 진행](#)을 참조하세요.
- 암호를 사용하여 SSH 키에 대한 액세스를 보호하고 키를 안전한 위치에 저장합니다.
- Bastion Host에 대한 로깅 및 모니터링을 설정합니다. 로깅 및 모니터링은 운영 및 보안 관점에서 시스템 유지 관리의 중요한 부분입니다. Bastion Host의 연결 및 활동을 모니터링하는 방법은 여러 가지가 있습니다. 자세한 내용은 설명서에서 다음 주제를 참조하세요.
 - [AWS Systems Manager 모니터링](#)
 - [AWS Systems Manager에서 로깅 및 모니터링](#)
 - [세션 활동 감사](#)
 - [세션 활동 로그](#)

에픽

리소스 배포

작업	설명	필요한 기술
<p>코드 리포지토리를 복제합니다.</p>	<ol style="list-style-type: none"> 명령줄 인터페이스에서 작업 디렉토리를 샘플 파일을 저장하고자 하는 위치로 변경합니다. 다음 명령을 입력합니다. <pre>git clone https://github.com/aws-samples/secured-bastion-host-terraform.git</pre>	<p>DevOps 엔지니어, 개발자</p>
<p>Terraform 작업 디렉토리를 초기화합니다.</p>	<p>이 단계는 첫 배포에만 필요합니다. 이 패턴을 다시 배포하는 경우, 다음 단계로 건너뛰세요.</p> <p>복제된 리포지토리의 루트 디렉터리에 다음 명령을 입력합니다.</p> <ul style="list-style-type: none"> \$S3_STATE_BUCKET 은 Terraform 상태가 포함된 S3 버킷의 이름입니다. \$PATH_TO_STATE_FILE 은 infra/bastion-host/tetfstate 와 같은 Terraform 상태 파일의 키입니다. \$AWS_REGION 은 배포된 S3 버킷이 위치한 리전입니다. 	<p>DevOps 엔지니어, 개발자, 테라폼</p>

작업	설명	필요한 기술
	<pre>terraform init \ -backend-config="bucket=\$S3_STATE_BUCKET" \ -backend-config="key=\$PATH_TO_STATE_FILE" \ -backend-config="region=\$AWS_REGION"</pre> <p>참고: config.tf 파일을 열고 terraform 섹션에서 이러한 값을 수동으로 제공할 수도 있습니다.</p>	
<p>리소스를 배포합니다.</p>	<ol style="list-style-type: none"> 복제된 리포지토리의 루트 디렉터리에 다음 명령을 입력합니다. <pre>terraform apply -var-file="dev.tfvars"</pre> <ol style="list-style-type: none"> AWS 계정에 적용되는 모든 변경 사항 목록을 검토한 다음 배포를 확정합니다. 모든 리소스가 배포될 때까지 기다리세요. 	<p>DevOps 엔지니어, 개발자, 테라폼</p>

로컬 환경 설정

작업	설명	필요한 기술
<p>SSH 연결을 구성합니다.</p>	<p>Session Manager를 통한 SSH 연결을 허용하도록 SSH 구성 파일을 업데이트합니다. 자세한 지침은 Session Manager</p>	<p>DevOps 엔지니어</p>

작업	설명	필요한 기술
	<p>의 SSH 연결 허용을 참조하세요. 이렇게 하면 인증된 사용자가 프록시 명령을 입력하여 Session Manager 세션을 시작하고 양방향 연결을 통해 모든 데이터를 전송할 수 있습니다.</p>	
SSH 키를 생성합니다.	<p>다음 명령을 입력하여 로컬 프라이빗 및 퍼블릭 SSH 키 페어를 생성합니다. 이 키 페어를 사용하여 Bastion Host에 연결합니다.</p> <pre>ssh-keygen -t rsa -f my_key</pre>	DevOps 엔지니어, 개발자

Session Manager를 사용하여 Bastion Host에 연결

작업	설명	필요한 기술
인스턴스 ID를 가져옵니다.	<ol style="list-style-type: none"> 배포된 Bastion Host에 연결하려면 EC2 인스턴스의 ID가 필요합니다. ID를 찾으려면 다음 작업 중 하나를 수행합니다. <ul style="list-style-type: none"> https://console.aws.amazon.com/ec2/에서 Amazon EC2 콘솔을 엽니다. 탐색 창에서 인스턴스를 선택합니다. Bastion Host 인스턴스를 찾습니다. AWS CLI에서 다음 명령을 입력합니다. 	일반 AWS

작업	설명	필요한 기술
	<pre data-bbox="662 210 1029 327">aws ec2 describe- instances</pre> <p data-bbox="662 365 1008 735">결과를 필터링하려면 다음 명령을 입력합니다. 여기서 \$BASTION_HOST_TAG 는 Bastion Host에 할당한 태그입니다. 이 옵션의 기본값은 sandbox-dev-bastion-host 입니다.</p> <pre data-bbox="662 772 1029 1289">aws ec2 describe- instances \ --filters "Name=tag:Name,Values=\$BASTION_HOST_ TAG" \ --output text \ --query 'Reservations[*].Instances[*].InstanceId' \ --output text</pre> <p data-bbox="591 1306 1026 1436">2. EC2 인스턴스의 ID를 복사합니다. 이 ID는 나중에 사용됩니다.</p>	

작업	설명	필요한 기술
SSH 퍼블릭 키를 전송합니다.	<p>참고: 이 섹션에서는 Bastion Host의 인스턴스 메타데이터에 퍼블릭 키를 업로드합니다. 키를 업로드한 후 60초 이내에 Bastion Host와 연결을 시작할 수 있습니다. 60초 후에는 퍼블릭 키가 제거됩니다. 자세한 내용은 이 안내서의 문제 해결 부분을 참조하세요. Bastion Host에 연결하기 전에 키가 제거되지 않도록 다음 단계를 빠르게 완료하세요.</p> <ol style="list-style-type: none"> 1. EC2 인스턴스 Connect를 사용하여 SSH 키를 Bastion Host에 전송합니다. 다음 명령을 입력합니다. <ul style="list-style-type: none"> • \$INSTANCE_ID 는 EC2 인스턴스의 ID입니다. • \$PUBLIC_KEY_FILE 은 my_key.pub 과 같은 퍼블릭 키 파일의 경로입니다 <p>중요: 프라이빗 키가 아닌 퍼블릭 키를 사용해야 합니다.</p> <pre>aws ec2-instance-connect send-ssh-public-key \ --instance-id \$INSTANCE_ID \ --instance-os-user ec2-user \</pre> 	일반 AWS

작업	설명	필요한 기술
	<pre data-bbox="630 205 1027 346">--ssh-public-key file://\$PUBLIC_KEY _FILE</pre> <p data-bbox="589 359 1016 537">2. 키가 성공적으로 업로드되었다는 메시지가 나타날 때까지 기다리세요. 바로 다음 단계를 진행합니다.</p>	
<p data-bbox="115 583 505 617">Bastion Host에 연결합니다.</p>	<p data-bbox="589 583 1016 711">1. 다음 명령을 입력하여 Session Manager를 통해 Bastion Host에 연결합니다.</p> <ul data-bbox="630 737 1016 1020" style="list-style-type: none"> • \$PRIVATE_KEY_FILE 은 my_key와 같은 프라이빗 키의 경로입니다 • \$INSTANCE_ID 는 EC2 인스턴스의 ID입니다. <pre data-bbox="630 1058 1027 1213">ssh -i \$PRIVATE_KEY_FILE ec2-user@ \$INSTANCE_ID</pre> <p data-bbox="589 1234 1016 1413">2. yes를 입력하여 연결을 확인합니다. 그러면 Session Manager를 사용하여 SSH 연결이 열립니다.</p> <p data-bbox="589 1486 1016 1808">참고: Bastion Host와의 SSH 연결을 여는 다른 옵션도 있습니다. 자세한 내용은 이 패턴의 추가 정보 섹션에서 Bastion Host와 SSH 연결을 설정하는 대안적인 접근법을 참조하세요.</p>	<p data-bbox="1068 583 1214 617">일반 AWS</p>

(선택 사항)정리

작업	설명	필요한 기술
배포된 리소스를 제거합니다.	<ol style="list-style-type: none"> 1. 배포된 모든 리소스를 제거하려면 복제된 리포지토리의 루트 디렉터리에서 다음 명령을 실행합니다. <pre>terraform destroy - var-file="dev.tfvars"</pre> <ol style="list-style-type: none"> 2. 리소스 제거를 확인합니다. 	DevOps 엔지니어, 개발자, 테라폼

문제 해결

문제	Solution
Bastion Host에 연결하려고 할 때 TargetNot Connected 오류가 발생	<ol style="list-style-type: none"> 1. Amazon EC2 설명서의 인스턴스 재부팅에 나와 있는 지침에 따라 Bastion Host를 재부팅합니다. 2. 인스턴스가 성공적으로 재부팅되면 Bastion Host에 퍼블릭 키를 재전송하고 연결을 다시 시도합니다.
Bastion Host에 연결하려고 할 때 Permission denied 오류가 발생	퍼블릭 키가 Bastion Host에 업로드된 후 60초 이내에 연결을 시작해야 합니다. 60초 후에는 키가 자동으로 제거되며, 이 키를 사용하여 인스턴스에 연결할 수 없습니다. 이 경우 단계를 반복하여 키를 인스턴스에 재전송할 수 있습니다.

관련 리소스

AWS 설명서

- [AWS Systems Manager Session Manager](#)(Systems Manager 설명서)
- [AWS CLI에 Session Manager 플러그인 설치](#)(Systems Manager 설명서)
- [Session Manager에 SSH 연결 허용](#)(Systems Manager 설명서)
- [EC2 인스턴스 Connect 사용 정보](#)(Amazon EC2 설명서)
- [EC2 인스턴스 Connect를 사용한 연결](#)(Amazon EC2 설명서)
- [Amazon EC2 자격 증명 및 액세스 관리](#)(Amazon EC2 설명서)
- [IAM 역할을 사용하여 Amazon EC2 인스턴스에서 실행되는 애플리케이션에 권한 부여](#)(IAM 설명서)
- [IAM의 보안 모범 사례](#)(IAM 설명서)
- [자세한 내용은 보안 그룹을 사용하여 리소스에 대한 트래픽 제어](#)(Amazon VPC 설명서)

기타 리소스

- [Terraform 개발자 웹페이지](#)
- [명령: 유효성](#)(Terraform 설명서)
- [명령: fmt](#)(Terraform 설명서)
- [HashiCorp 테라폼 테스트 \(블로그 게시물\)](#) HashiCorp
- [Checkov 웹페이지](#)

추가 정보

Bastion Host와 SSH 연결을 설정하는 대체 접근 방식

포트 전달

-D 8888 옵션을 사용하여 동적 포트 포워딩으로 SSH 연결을 열 수 있습니다. 자세한 내용은 [explainshell.com](#)에서 [이 지침](#)을 참조하세요. 다음은 포트 포워딩을 사용하여 SSH 연결을 여는 명령의 예제입니다.

```
ssh -i $PRIVATE_KEY_FILE -D 8888 ec2-user@$INSTANCE_ID
```

이런 종류의 연결은 Bastion Host를 통해 로컬 브라우저의 트래픽을 전달할 수 있는 SOCKS 프록시를 여는 방식입니다. Linux 또는 macOS를 사용하는 경우 모든 옵션을 보려면 `man ssh`를 입력하세요. 그러면 SSH 참조 설명서가 표시됩니다.

제공된 스크립트 사용

[예픽](#) 섹션의 Session Manager를 사용하여 Bastion Host에 연결에 설명된 단계를 수동으로 실행하는 대신 코드 리포지토리에 포함된 connect.sh 스크립트를 사용할 수 있습니다. 이 스크립트는 SSH 키 페어를 생성하고, 퍼블릭 키를 EC2 인스턴스로 푸시하며 Bastion Host와의 연결을 시작합니다. 스크립트를 실행할 때 태그와 키 이름을 인수로 전달합니다. 다음은 스크립트를 실행하는 명령의 예입니다.

```
./connect.sh sandbox-dev-bastion-host my_key
```

AWS Managed Microsoft AD와 온프레미스 Microsoft Active Directory를 사용하여 DNS 확인 중앙 집중화

작성자: Brian Westmoreland(AWS)

환경: 프로덕션

기술: 인프라, 네트워킹

워크로드: Microsoft

AWS 서비스: AWS Managed Microsoft AD, Amazon Route 53, AWS RAM

요약

이 패턴은 AWS Directory Service for Microsoft Active Directory(AWS Managed Microsoft AD)를 사용하여 AWS 다중 계정 환경 내에서 도메인 이름 시스템(DNS) 확인을 중앙 집중화하기 위한 지침을 제공합니다. 이 패턴에서 AWS DNS 네임스페이스는 온프레미스 DNS 네임스페이스의 하위 도메인입니다. 또한 이 패턴은 온프레미스 DNS 솔루션에서 Microsoft Active Directory를 사용할 때 AWS로 쿼리를 전달하도록 온프레미스 DNS 서버를 구성하는 방법에 대한 지침을 제공합니다.

사전 조건 및 제한 사항

사전 조건

- AWS 다중 계정 환경이 AWS Organizations를 사용하여 설정되어 있습니다.
- AWS 계정 간에 네트워크 연결이 설정되어 있습니다.
- AWS와 온프레미스 환경 간에 네트워크 연결이 설정되어 있습니다(AWS Direct Connect 또는 모든 유형의 VPN 연결 사용).
- 로컬 워크스테이션에 AWS Command Line Interface (AWS CLI)가 구성되어 있습니다.
- 계정 간에 Amazon Route 53 규칙을 공유하는 데 AWS Resource Access Manager(AWS RAM)를 사용합니다. 따라서 에픽 섹션에 설명된 대로 AWS Organizations 환경 내에서 공유를 활성화해야 합니다.

제한 사항

- AWS Managed Microsoft AD Standard Edition의 공유 한도는 5입니다.

- AWS Managed Microsoft AD Enterprise Edition의 공유 한도는 125입니다.
- 이 패턴의 솔루션은 AWS RAM을 통한 공유를 지원하는 AWS 리전으로 제한됩니다.

제품 버전

- Windows Server 2008, 2012, 2012 R2 또는 2016에서 실행되는 Microsoft Active Directory

아키텍처

대상 아키텍처

이 설계에서는 AWS Managed Microsoft AD를 공유 서비스 AWS 계정에 설치합니다. 필수 사항은 아니지만 이 패턴은 이 구성을 가정합니다. 다른 AWS 계정에서 AWS Managed Microsoft AD를 구성하는 경우 그에 따라 에픽 섹션의 단계를 수정해야 할 수 있습니다.

이 설계에서는 Route 53 Resolver를 사용하여 Route 53 규칙 사용을 통한 이름 확인을 지원합니다. 온프레미스 DNS 솔루션에서 Microsoft DNS를 사용하는 경우 회사 DNS 네임스페이스(company.com)의 하위 도메인인 AWS 네임스페이스(aws.company.com)에 대한 조건부 전달 규칙을 생성하는 것은 간단하지 않습니다. 기존 조건부 전달자를 생성하려고 하면 오류가 발생합니다. Microsoft Active Directory가 이미 company.com의 모든 하위 도메인에 대해 신뢰할 수 있는 것으로 간주되기 때문입니다. 이 오류를 해결하려면 먼저 aws.company.com에 대한 위임을 생성하여 해당 네임스페이스의 권한을 위임해야 합니다. 그런 다음 조건부 전달자를 생성할 수 있습니다.

각 스포크 계정의 Virtual Private Cloud(VPC)에는 루트 AWS 네임스페이스를 기반으로 하는 고유한 DNS 네임스페이스가 있을 수 있습니다. 이 설계에서는 각 스포크 계정이 계정 이름의 약어를 기본 AWS 네임스페이스에 추가합니다. 스포크 계정의 프라이빗 호스팅 영역이 생성되면 영역은 스포크 계정의 VPC뿐만 아니라 중앙 AWS 네트워크 계정의 VPC와도 연결됩니다. 이렇게 하면 중앙 AWS 네트워크 계정이 스포크 계정과 관련된 DNS 쿼리에 응답할 수 있습니다.

자동화 및 규모 조정

이 설계는 Route 53 Resolver 엔드포인트를 사용하여 AWS와 온프레미스 환경 간에 DNS 쿼리를 확장합니다. 각 Route 53 Resolver 엔드포인트는 여러 개의 탄력적 네트워크 인터페이스(여러 가용 영역에 분산되어 있음)로 구성되며, 각 네트워크 인터페이스는 초당 최대 10,000개의 쿼리를 처리할 수 있습니다. Route 53 Resolver는 엔드포인트당 6개까지 IP 주소를 지원하므로 이 설계에서는고가용성을 위해 DNS 쿼리를 초당 60,000개까지 여러 가용 영역에서 분산 지원합니다.

또한 이 패턴은 AWS 내 향후 성장을 자동으로 고려합니다. AWS에 추가된 새 VPC 및 관련 프라이빗 호스팅 영역을 지원하도록 온프레미스에 구성된 DNS 전달 규칙을 수정할 필요는 없습니다.

도구

서비스

- [AWS Directory Service for Microsoft Active Directory](#)를 이용하면 디렉터리 인식 워크로드와 AWS 리소스가 AWS 클라우드에서 Microsoft Active Directory를 사용할 수 있습니다.
- [AWS Organizations](#)는 사용자가 생성하고 중앙에서 관리하는 조직으로 여러 AWS 계정을 통합할 수 있는 계정 관리 서비스입니다.
- [AWS Resource Access Manager\(AWS RAM\)](#)를 이용하면 AWS 계정 간에 리소스를 안전하게 공유하여 운영 오버헤드를 줄이고 가시성과 감사 가능성을 제공할 수 있습니다.
- [Amazon Route 53](#)는 가용성과 확장성이 뛰어난 DNS 웹 서비스입니다.

도구

- [AWS Command Line Interface\(AWS CLI\)](#)는 명령줄 쉘에서 명령을 사용하여 AWS 서비스와 상호 작용할 수 있는 오픈 소스 도구입니다. 이 패턴에서는 AWS CLI를 사용하여 Route 53 권한 부여를 구성합니다.

에픽

AWS Managed Microsoft AD 디렉터리 생성 및 공유

작업	설명	필요한 기술
AWS Managed Microsoft AD 를 배포합니다.	<ol style="list-style-type: none"> 1. 새 디렉터리를 생성하고 구성합니다. 자세한 단계는 AWS Directory Service 관리 가이드의 AWS Managed Microsoft AD 디렉터리 생성을 참조하세요. 2. AWS Managed Microsoft AD 도메인 컨트롤러의 IP 주 	AWS 관리자

작업	설명	필요한 기술
	소를 기록합니다. 이후 단계에서 이 주소를 참조합니다.	
디렉터리를 공유합니다.	<p>디렉터리를 구축한 후에는 AWS 조직의 다른 AWS 계정과 디렉터리를 공유합니다. 지침은 AWS Directory Service 관리 가이드의 내 디렉터리 공유를 참조하세요.</p> <p>참고: AWS Managed Microsoft AD Standard Edition의 공유 한도는 5개입니다. Enterprise Edition의 공유 한도는 125개입니다.</p>	AWS 관리자

Route 53 구성

작업	설명	필요한 기술
Route 53 Resolver를 생성합니다.	<p>Route 53 Resolver는 AWS와 온프레미스 데이터 센터 간의 DNS 쿼리 확인을 용이하게 합니다.</p> <ol style="list-style-type: none"> Route 53 개발자 안내서의 지침에 따라 Route 53 Resolver를 설치합니다. 고가용성을 위해 중앙 AWS 네트워크 계정 VPC 내 최소 두 가용 영역에 있는 프라이빗 서브넷에 Route 53 Resolver를 구성합니다. 	AWS 관리자

작업	설명	필요한 기술
	참고: 중앙 AWS 네트워크 계정 VPC를 반드시 사용해야 하는 것은 아니지만 나머지 단계에서는 이 구성을 가정합니다.	

작업	설명	필요한 기술
Route 53 규칙을 생성합니다.	<p>특정 사용 사례에는 많은 수의 Route 53 규칙이 필요할 수 있지만 다음 규칙을 기준으로 구성해야 합니다.</p> <ul style="list-style-type: none"> • 아웃바운드 Route 53 Resolver를 사용하여 온 프레미스 네임스페이스 (company.com)에 대한 발신 규칙. <ul style="list-style-type: none"> • 이 규칙을 스포크 AWS 계정과 공유합니다. • 이 규칙을 스포크 계정 VPC와 연결합니다. • 중앙 네트워크 계정 Route 53 인바운드 Resolver를 가리키는 AWS 네임스페이스 (aws.company.com)에 대한 수신 규칙. <ul style="list-style-type: none"> • 이 규칙을 스포크 AWS 계정과 공유합니다. • 규칙을 스포크 계정 VPC와 연결합니다. • 이 규칙을 중앙 AWS 네트워크 계정 VPC(Route 53 Resolver가 있음)와 연결하지 마세요. • AWS Managed Microsoft AD 도메인 컨트롤러(이전 에픽의 IP 사용)를 가리키는 AWS 네임스페이스 (aws.company.com)에 대한 두 번째 수신 규칙. 	AWS 관리자

작업	설명	필요한 기술
	<ul style="list-style-type: none"> 이 규칙을 중앙 AWS 네트워크 계정 VPC(Route 53 Resolver가 있음)와 연결합니다. 이 규칙을 다른 AWS 계정과 공유하거나 연결하지 마세요. <p>자세한 내용은 Route 53 개발자 안내서의 전달 규칙 관리를 참조하세요.</p>	

온프레미스 Active Directory DNS 구성

작업	설명	필요한 기술
<p>위임을 생성합니다.</p>	<p>Microsoft DNS 스냅인 (dnsmgmt.msc)을 사용하여 Active Directory 내에서 company.com 네임스페이스에 대한 새 위임을 생성합니다. 위임된 도메인의 이름은 aws여야 합니다. 이렇게 하면 위임의 정규화된 도메인 이름(FQDN)은 aws.company.com 이 됩니다. 이름 서버의 경우 중앙 DNS AWS 계정에 있는 AWS 인바운드 Route 53 Resolver의 IP 주소를 IP 값으로 사용하고 server.aws.company.com 을 이름으로 사용합니다.</p>	<p>Active Directory</p>

작업	설명	필요한 기술
조건부 전달자를 생성합니다.	Microsoft DNS 스냅인 (dnsmgmt.msc)을 사용하여 aws.company.com 에 대한 새 조건부 전달자를 생성합니다. 조건부 전달자의 대상으로 AWS Managed Microsoft AD 도메인 컨트롤러의 IP 주소를 사용합니다.	Active Directory

스포크 AWS 계정을 위한 Route 53 프라이빗 호스팅 영역 생성

작업	설명	필요한 기술
Route 53 프라이빗 호스팅 영역을 생성합니다.	각 스포크 계정에 Route 53 프라이빗 호스팅 영역을 생성합니다. 이 프라이빗 호스팅 영역을 스포크 계정 VPC와 연결합니다. 자세한 단계는 Route 53 개발자 안내서의 프라이빗 호스팅 영역 생성 을 참조하세요.	AWS 관리자
권한 부여를 생성합니다.	AWS CLI를 사용하여 중앙 AWS 네트워크 계정 VPC에 대한 권한 부여를 생성합니다. 각 스포크 AWS 계정의 컨텍스트에서 다음 명령을 실행합니다. <pre>aws route53 create-vc c-association-auth orization --hosted- zone-id <hosted-zone- id> \</pre>	AWS 관리자

작업	설명	필요한 기술
	<pre data-bbox="597 205 1024 346">--vpc VPCRegion =<region>,VPCId=<vpc- id></pre> <p data-bbox="597 380 1011 464">여기서 각 항목은 다음과 같습니다.</p> <ul data-bbox="597 512 1019 842" style="list-style-type: none"> • <hosted-zone-id> 는 스포크 계정의 Route 53 프라이빗 호스팅 영역입니다. • <region> 및 <vpc-id>는 중앙 AWS 네트워크 계정 VPC의 AWS 리전 및 VPC ID입니다. 	

작업	설명	필요한 기술
연결을 생성합니다.	<p>AWS CLI를 사용하여 중앙 AWS 네트워크 계정 VPC에 대한 Route 53 프라이빗 호스팅 영역 연결을 생성합니다. 중앙 AWS 네트워크 계정의 컨텍스트에서 다음 명령을 실행합니다.</p> <pre data-bbox="592 583 1031 903">aws route53 associate -vpc-with-hosted-zone one --hosted-zone-id <hosted-zone-id> \ --vpc VPCRegion =<region>,VPCId=<vpc-id></pre> <p>여기서 각 항목은 다음과 같습니다.</p> <ul data-bbox="592 1066 1031 1396" style="list-style-type: none"> • <hosted-zone-id> 는 스포크 계정의 Route 53 프라이빗 호스팅 영역입니다. • <region> 및 <vpc-id>는 중앙 AWS 네트워크 계정의 AWS 리전 및 VPC ID입니다. 	AWS 관리자

관련 리소스

- [Simplify DNS management in a multi-account environment with Route 53 Resolver](#)(Mahmoud Matouk의 AWS 블로그 게시물)
- [AWS Managed Microsoft AD로 디렉터리 생성](#)(AWS Directory Service 설명서)
- [AWS Managed Microsoft AD 디렉터리 공유](#)(AWS Directory Service 설명서)
- [Route 53 Resolver 설치](#)(Amazon Route 53 설명서)

- [Route 53 프라이빗 호스팅 영역 생성](#)(Amazon Route 53 설명서)

Amazon CloudWatch 옵저버빌리티 액세스 관리자를 사용하여 모니터링을 중앙 집중화합니다.

작성자: 아난드 크리슈나 바라나시 (AWS), 지미 모건 (AWS), 아시시 쿠마르 (AWS), 발라지 베다기리 (AWS), 재그디시 코마콜라 (AWS), 사라트 찬드라 포톨라 (AWS), 비벡 탄가무투 (AWS)

[코드 리포지토리: -terraform cloudwatch-obervability-access-manager](#)

환경: 프로덕션

기술: 인프라, 다중 계정 전략, 운영

AWS 서비스: 아마존 CloudWatch, 아마존 CloudWatch 로그

요약

관찰성은 애플리케이션을 모니터링하고, 이해하고, 문제를 해결하는 데 매우 중요합니다. AWS Control Tower 또는 랜딩 존 구현과 같이 여러 계정을 아우르는 애플리케이션은 대량의 로그와 추적 데이터를 생성합니다. 문제를 빠르게 해결하거나 사용자 분석 또는 비즈니스 분석을 이해하려면 모든 계정에서 공통의 관찰성 플랫폼이 필요합니다. Amazon CloudWatch Observability Access Manager를 사용하면 중앙 위치에서 여러 계정 로그에 액세스하고 이를 제어할 수 있습니다.

Observability Access Manager를 사용하여 소스 계정에서 생성된 관찰성 데이터 로그를 보고 관리할 수 있습니다. 소스 계정은 리소스에 대한 관찰성 데이터를 생성하는 개별 AWS 계정입니다. 관찰성 데이터는 소스 계정과 모니터링 계정 간에 공유됩니다. 공유 옵저버빌리티 데이터에는 Amazon의 지표 CloudWatch, Amazon Logs의 로그, AWS X-Ray의 추적이 포함될 수 있습니다. CloudWatch 자세한 내용은 [Observability Access Manager 설명서](#)를 참조하세요.

이 패턴은 여러 AWS 계정에서 실행되는 애플리케이션 또는 인프라를 보유하고 로그를 볼 수 있는 공통 장소가 필요한 사용자를 위한 것입니다. Terraform을 사용하고 Observability Access Manager를 설정하여 이러한 애플리케이션 또는 인프라의 상태를 모니터링하는 방법을 설명합니다. 이 솔루션은 여러 가지 방법으로 설치할 수 있습니다.

- 수동으로 설정하는 독립형 Terraform 모듈로
- 지속적 통합 및 지속적 전달 (CI/CD) 파이프라인 사용
- [AWS Control Tower Account Factory for Terraform](#) (AFT) 과 같은 다른 솔루션과 통합하여

[에픽](#) 섹션의 지침은 수동 구현을 다룹니다. AFT 설치 단계는 GitHub [옵저버빌리티](#) 액세스 관리자 리포지토리의 readme 파일을 참조하십시오.

사전 조건 및 제한 사항

사전 조건

- [Terraform](#)이 시스템 또는 자동화된 파이프라인에 설치되거나 참조됩니다. ([최신 버전](#)을 사용하는 것이 좋습니다.)
- 중앙 모니터링 계정으로 사용할 수 있는 계정. 다른 계정은 로그를 보기 위해 중앙 모니터링 계정에 대한 링크를 생성합니다.
- (선택 사항) AWS GitHub CodeCommit, Atlassian Bitbucket 또는 유사한 시스템과 같은 소스 코드 리포지토리. 자동화된 CI/CD 파이프라인을 사용하는 경우에는 소스 코드 리포지토리가 필요하지 않습니다.
- (선택 사항) 에서 코드 검토 및 코드 협업을 위한 풀 리퀘스트 (PR) 를 생성할 수 있는 권한 GitHub

제한 사항

Observability Access Manager에는 다음과 같은 서비스 할당량이 있으며 이는 변경할 수 없습니다. 이 기능을 배포하기 전에 이러한 할당량을 고려하세요. 자세한 내용은 설명서의 [CloudWatch 서비스 할당량](#)을 참조하십시오. CloudWatch

- 소스 계정 링크: 각 소스 계정을 최대 다섯 모니터링 계정에 연결할 수 있습니다.
- 싱크: 계정당 싱크를 하나만 사용할 수 있습니다.

또한 다음과 같습니다.

- 싱크와 링크는 동일한 AWS 리전에서 생성해야 하며, 리전 간으로는 생성할 수 없습니다.
- 지역 간, 교차 계정 모니터링의 경우 로그와 추적을 제외한 경보 및 지표에 대한 [교차 계정 및 교차 지역 CloudWatch 대시보드를](#) 만들 수 있습니다. 또 다른 옵션은 [Amazon OpenSearch Service를 사용하여 중앙 집중식 로깅을 생성하는 것입니다.](#)

아키텍처

구성 요소

Amazon CloudWatch Observability Access Manager는 계정 간 옵저버빌리티를 가능하게 하는 두 가지 주요 구성 요소로 구성되어 있습니다.

- 싱크는 소스 계정이 중앙 모니터링 계정으로 관찰성 데이터를 전송할 수 있는 기능을 제공합니다. 싱크는 기본적으로 소스 계정을 연결할 수 있는 게이트웨이 정션을 제공합니다. 싱크 게이트웨이 또는 연결은 하나만 있을 수 있으며 여러 계정을 연결할 수 있습니다.
- 각 소스 계정에는 싱크 게이트웨이 정션으로 연결되는 링크가 있으며 이 링크를 통해 관찰성 데이터가 전송됩니다. 각 소스 계정에서 링크를 만들기 전에 싱크를 만들어야 합니다.

아키텍처

다음 다이어그램은 Observability Access Manager와 그 구성 요소를 보여줍니다.

도구

서비스

- [Amazon](#)은 AWS 리소스의 지표와 AWS에서 실행하는 애플리케이션을 실시간으로 모니터링할 수 있도록 CloudWatch 도와줍니다.
- [AWS Organizations](#)는 사용자가 생성하고 중앙에서 관리하는 조직으로 여러 AWS 계정을 통합할 수 있는 계정 관리 서비스입니다.
- [AWS Identity and Access Management\(IAM\)](#)를 이용하면 사용자에 대해 인증 및 권한 부여를 제어함으로써 AWS 리소스에 대한 액세스를 안전하게 관리할 수 있습니다.

도구

- [Terraform](#)은 클라우드 및 온 프레미스 리소스를 생성하고 관리하는 데 도움이 HashiCorp 되는 코드 형 인프라 (IaC) 도구입니다.
- [AWS Control Tower Account Factory for Terraform\(AFT\)](#)은 AWS Control Tower에서 계정을 프로비저닝하고 사용자 지정하는 데 도움이 되는 Terraform 파이프라인을 설정합니다. 선택적으로 AFT를 사용하여 여러 계정에 걸쳐 대규모로 Observability Access Manager를 설정할 수 있습니다.

코드 리포지토리

이 패턴의 코드는 GitHub [Observability](#) Access Manager 리포지토리에서 사용할 수 있습니다.

모범 사례

- AWS Control Tower 환경에서는 로깅 계정을 중앙 모니터링 계정(싱크)으로 표시합니다.
- AWS Organizations에 여러 계정을 보유한 조직이 여러 개 있는 경우 구성 정책에 개별 계정 대신 조직을 포함하는 것이 좋습니다. 계정 수가 적거나 계정이 싱크 구성 정책에 조직의 일부가 아닌 경우 개별 계정을 대신 포함할 수 있습니다.

에픽

싱크 모듈 설정

작업	설명	필요한 기술
리포지토리를 복제합니다.	GitHub 오픈서버빌리티 액세스 관리자 리포지토리를 복제하십시오. <pre>git clone https://github.com/aws-samples/cloudwatch-observability-access-manager-terraform</pre>	AWS DevOps, 클라우드 관리자, AWS 관리자
싱크 모듈의 속성 값을 지정합니다.	main.tf 파일(리포지토리의 deployments/aft-account-customizations/LOGGING/terraform/ 폴더)에서 다음 속성의 값을 지정합니다. <ul style="list-style-type: none"> • sink_name : 아마존 CloudWatch 싱크대의 이름. • allowed_oam_resource_types : 오픈서버빌리티 액세스 관리자는 현재 CloudWatch 지표, 로그 그룹 	AWS DevOps, 클라우드 관리자, AWS 관리자

작업	설명	필요한 기술
	<p>및 AWS X-Ray 추적을 지원 합니다.</p> <ul style="list-style-type: none"> allowed_source_accounts : 중앙 CloudWatch 싱크 계정으로 로그를 전송할 수 있는 소스 계정. allowed_source_organizations : 중앙 CloudWatch 싱크 계정으로 로그를 전송할 수 있는 소스 Control Tower 조직 <p>자세한 내용은 AWS::Oam::Sink AWS CloudFormation 설명서를 참조하십시오.</p>	
<p>싱크 모듈을 설치합니다.</p>	<p>모니터링 계정으로 선택한 AWS 계정의 보안 인증 정보를 내보내고 Observability Access Manager 싱크 모듈을 설치합니다.</p> <div style="border: 1px solid #ccc; border-radius: 10px; padding: 10px; margin-top: 10px;"> <pre>Terraform Init Terraform Plan Terraform Apply</pre> </div>	<p>AWS DevOps, 클라우드 관리자, AWS 관리자</p>

링크 모듈 설정

작업	설명	필요한 기술
<p>링크 모듈의 속성 값을 지정합니다.</p>	<p>main.tf 파일(리포지토리의 deployments/aft-account-customizations/LOGGING/terrafo</p>	<p>AWS DevOps, 클라우드 관리자, 클라우드 아키텍트</p>

작업	설명	필요한 기술
	<p>rm/ 폴더)에서 다음 속성의 값을 지정합니다.</p> <ul style="list-style-type: none"> • <code>account_label</code> : 다음 값 중 하나를 사용합니다. <ul style="list-style-type: none"> • <code>\$AccountName</code> : 계정의 이름입니다. • <code>\$AccountEmail</code> : 이 메일 도메인을 포함하는 전역적으로 고유한 이메일 주소(예: <code>hello@example.com</code>)입니다. • <code>\$AccountEmailNoDomain</code> : 도메인 이름이 없는 이메일 주소입니다. • <code>allowed_oam_resource_types</code> : 옵저버빌리티 액세스 관리자는 현재 CloudWatch 지표, 로그 그룹 및 AWS X-Ray 추적을 지원합니다. <p>자세한 내용은 AWS::Oam:Link AWS CloudFormation 설명서를 참조하십시오.</p>	

작업	설명	필요한 기술
<p>개별 계정용 링크 모듈을 설치합니다.</p>	<p>개별 계정의 보안 인증 정보를 내보내고 Observability Access Manager 링크 모듈을 설치합니다.</p> <div style="border: 1px solid #ccc; border-radius: 10px; padding: 5px; margin: 10px 0;"> <p>Terraform Plan Terraform Apply</p> </div> <p>각 계정에 대해 개별적으로 링크 모듈을 설정하거나 AFT를 사용하여 많은 계정에서 이 모듈을 자동으로 설치할 수 있습니다.</p>	<p>AWS DevOps, 클라우드 관리자, 클라우드 아키텍트</p>

연결 승인 sink-to-link

작업	설명	필요한 기술
<p>상태 메시지를 확인합니다.</p>	<ol style="list-style-type: none"> 모니터링 계정에 로그인합니다. https://console.aws.amazon.com/cloudwatch/에서 CloudWatch 콘솔을 엽니다. 왼쪽 탐색 창에서 설정을 선택합니다. <p>오른쪽에 녹색 체크 표시가 있는 모니터링 계정 활성화 상태 메시지가 보일 것입니다. 즉, 모니터링 계정에는 다른 계정의 링크가 연결되는 Observability Access Manager 싱크가 있습니다.</p>	

작업	설명	필요한 기술
<p>link-to-sink 연결을 승인합니다.</p>	<ol style="list-style-type: none"> 상태 메시지 아래에서 계정을 연결할 리소스 옵션을 선택합니다. 이 정보는 이 계정이 모니터링 계정임을 확인하고, 테넌트 소스 계정에서 공유된 데이터 (로그, 메트릭, 트레이스) 를 나열하고, 계정 레이블을 \$로 표시합니다. AccountName <p>이 화면에서는 테넌트 계정을 모니터링 계정에 연결하는 두 가지 옵션인 조직 수준 승인 또는 계정 수준 승인을 제공합니다. 각 옵션에 대해 승인을 위해 AWS CloudFormation 템플릿을 다운로드하거나 각 계정을 개별적으로 승인할 수 있습니다.</p> <ol style="list-style-type: none"> 간소화를 위해 모든 계정을 선택하여 각 계정 수준에서 승인합니다. 이 옵션은 계정에 대한 승인 링크를 제공합니다. URL 복사를 선택하여 링크를 복사합니다. 각 소스 계정에 로그인합니다. 브라우저 창에서 링크를 붙여넣고 싱크에 링크 연결 승인을 선택합니다. 추가 소스 계정에 대해 반복합니다. 	<p>AWS DevOps, 클라우드 관리자, 클라우드 아키텍트</p>

작업	설명	필요한 기술
	자세한 내용은 Amazon CloudWatch 설명서의 모니터링 계정과 소스 계정 연결을 참조 하십시오.	

교차 계정 관찰성 데이터 확인

작업	설명	필요한 기술
교차 계정 데이터를 확인합니다.	<ol style="list-style-type: none"> 중앙의 모니터링 계정에 로그인합니다. https://console.aws.amazon.com/cloudwatch/에서 CloudWatch 콘솔을 엽니다. 왼쪽 탐색 창에서 교차 계정 로그, 지표 및 추적을 볼 수 있는 옵션을 선택합니다. 	AWS DevOps, 클라우드 관리자, 클라우드 아키텍트

(선택 사항) 소스 계정이 모니터링 계정을 신뢰할 수 있도록 설정

작업	설명	필요한 기술
다른 계정의 지표, 대시보드, 로그, 위젯, 경보를 확인합니다.	<p>추가 기능으로 CloudWatch 지표, 대시보드, 로그, 위젯 및 경보를 다른 계정과 공유할 수 있습니다. 각 계정은 CloudWatch-라는 IAM 역할을 사용하여 이 데이터에 대한 액세스 CrossAccountSharingRole 권한을 얻습니다.</p> <p>중앙 모니터링 계정과 신뢰 관계가 있는 소스 계정은 이 역할</p>	AWS DevOps, 클라우드 관리자, 클라우드 아키텍트

작업	설명	필요한 기술
	<p>을 맡아 모니터링 계정의 데이터를 볼 수 있습니다.</p> <p>CloudWatch 역할을 생성하기 위한 샘플 CloudFormation 스크립트를 제공합니다. IAM에서 역할 관리를 선택하고 데이터를 확인할 계정에서 이 스크립트를 실행합니다.</p> <pre data-bbox="592 646 1031 1837"> { "Version": "2012-10-17", "Statement": [{ "Effect": "Allow", "Principal": { "AWS": ["arn:aws:iam::XXXX XXXX:root", "arn:aws:iam::XXXX XXXX:root", "arn:aws:iam::XXXX XXXX:root", "arn:aws:iam::XXXX XXXX:root"] }, "Action": "sts:AssumeRole" }] } </pre>	

작업	설명	필요한 기술
	<p>자세한 내용은 설명서의 계정 간 기능 활성화를 참조하십시오. CloudWatch CloudWatch</p>	

(선택 사항) 모니터링 계정에서 교차 계정 교차 리전 보기

작업	설명	필요한 기술
<p>교차 계정 교차 리전 액세스를 설정합니다.</p>	<p>중앙 모니터링 계정에서 필요에 따라 계정 선택기를 추가하여 인증 없이 계정 간에 쉽게 전환하고 데이터를 볼 수 있습니다.</p> <ol style="list-style-type: none"> 1. 중앙의 모니터링 계정에 로그인합니다. 2. https://console.aws.amazon.com/cloudwatch/에서 CloudWatch 콘솔을 엽니다. 3. 왼쪽 탐색 창에서 설정을 선택합니다. 4. 교차 계정 교차 리전 보기 섹션에서 구성을 선택합니다. 5. 활성화를 선택한 다음 콘솔에서 선택기 보기 확인란을 선택합니다. 6. 다음 옵션 중 하나를 선택합니다. <ul style="list-style-type: none"> • 계정 ID 입력: 이 옵션을 선택하면 교차 계정 데이터를 보기 위해 계정을 변경할 때마다 계정 ID를 수 	<p>AWS DevOps, 클라우드 관리자, 클라우드 아키텍트</p>

작업	설명	필요한 기술
	<p>동으로 입력하라는 메시지가 표시됩니다.</p> <ul style="list-style-type: none"> • AWS 조직 계정 선택기: AWS CloudWatch Organizations와 통합한 경우 이 옵션은 조직의 전체 계정 목록이 포함된 드롭다운 선택기를 제공합니다. • 사용자 지정 계정 선택기: 이 옵션을 이용하면 계정 ID 목록을 수동으로 입력하여 선택기를 채울 수 있습니다. <p>7. 변경 사항 저장을 선택합니다.</p> <p>자세한 내용은 설명서의 교차 CloudWatch 계정 리전 콘솔을 참조하십시오. CloudWatch</p>	

관련 리소스

- [CloudWatch 계정 간 옵저버빌리티](#) (Amazon CloudWatch 설명서)
- [아마존 CloudWatch 옵저버빌리티 액세스 관리자 API 레퍼런스](#) (아마존 CloudWatch 설명서)
- [리소스: aws_oam_sink](#)(Terraform 설명서)
- [데이터 소스: aws_oam_link](#)(Terraform 설명서)
- [CloudWatchObservabilityAccessManager](#)(AWS Boto3 설명서)

시작 시 EC2 인스턴스에 필수 태그가 있는지 확인

작성자: Susanne Kangnoh(AWS)

환경: 프로덕션

기술: 인프라, 관리 및 거버넌스, 보안, 자격 증명, 규정 준수, 클라우드 네이티브

AWS 서비스: 아마존 EC2, AWS, CloudTrail 아마존 CloudWatch, 아마존 SNS

요약

Amazon Elastic Compute Cloud(Amazon EC2)는 Amazon Web Services(AWS) 클라우드에서 확장 가능 컴퓨팅 용량을 제공합니다. Amazon EC2를 사용하면 하드웨어에 사전 투자할 필요가 없어 더 빠르게 애플리케이션을 개발하고 배포할 수 있습니다.

태그를 사용하면 다양한 방식으로 AWS 리소스를 분류할 수 있습니다. EC2 인스턴스 태깅은 계정에 리소스가 많을 때 태그에 따라 특정 리소스를 빠르게 식별하려는 경우에 유용합니다. 태그를 사용하여 EC2 인스턴스에 사용자 지정 메타데이터를 할당할 수 있습니다. 각 태그는 사용자 정의 키와 값으로 구성됩니다. 조직의 요구 사항에 맞는 일관된 태그 집합을 생성하는 것이 좋습니다.

이 패턴은 EC2 인스턴스에서 특정 태그를 모니터링하는 데 도움이 되는 AWS CloudFormation 템플릿을 제공합니다. 템플릿은 새로운 EC2 인스턴스 태그 지정 또는 태그 제거를 탐지하기 위해 AWS CloudTrail TagResource 또는 UntagResource 이벤트를 감시하는 Amazon CloudWatch Events 이벤트를 생성합니다. 사전 정의된 태그가 누락된 경우, AWS Lambda 함수를 호출하고, Amazon Simple Notification Service(SNS)를 사용하여 사용자가 제공한 이메일 주소로 위반 메시지를 보냅니다.

사전 조건 및 제한 사항

사전 조건

- 활성 상태의 AWS 계정.
- 제공된 Lambda 코드를 업로드하기 위한 Amazon Simple Storage Service(S3).
- 위반 알림을 받으려는 이메일 주소입니다.

제한 사항

- 이 솔루션은 CloudTrail TagResource OR UntagResource 이벤트를 지원합니다. 다른 이벤트에 대한 알림은 생성되지 않습니다.

- 이 솔루션은 태그 키만 확인합니다. 키 값은 모니터링하지 않습니다.

아키텍처

워크플로 아키텍처

자동화 및 규모 조정

- 여러 AWS 지역 및 계정에 대해 AWS CloudFormation 템플릿을 여러 번 사용할 수 있습니다. 템플릿은 각 리전 또는 계정에서 한 번만 실행하면 됩니다.

도구

서비스

- [Amazon EC2](#)—Amazon Elastic Compute Cloud(Amazon EC2)는 클라우드에서 안전하고 확장 가능한 컴퓨팅 용량을 제공하는 웹 서비스입니다. 개발자가 보다 쉽게 웹 규모의 클라우드 컴퓨팅 작업을 할 수 있도록 설계되었습니다.
- [AWS CloudTrail](#) — CloudTrail AWS 계정의 거버넌스, 규정 준수, 운영 및 위험 감사를 지원하는 AWS 서비스입니다. 사용자, 역할 또는 AWS 서비스가 수행한 작업은 에서 이벤트로 기록됩니다 CloudTrail.
- [Amazon CloudWatch Events](#) — Amazon CloudWatch Events는 AWS 리소스의 변경 사항을 설명하는 시스템 이벤트의 스트림을 거의 실시간으로 제공합니다. CloudWatch 이벤트는 운영 변경이 발생하는 즉시 이를 인지하고 필요에 따라 수정 조치를 취하여 환경에 대응하고, 기능을 활성화하고, 변경하고, 상태 정보를 캡처합니다.
- [AWS Lambda](#) - Lambda는 서버를 프로비저닝하거나 관리할 필요 없이 코드를 실행할 수 있게 해주는 컴퓨팅 서비스입니다. Lambda는 필요 시에만 코드를 실행하며, 일일 몇 개의 요청에서 초당 수천 개의 요청까지 자동으로 규모를 조정합니다.
- [Amazon S3](#)—Amazon Simple Storage Service(S3)는 웹 사이트, 모바일 애플리케이션, 백업, 데이터 레이크 등 다양한 스토리지 솔루션에 사용할 수 있는 확장성이 뛰어난 객체 스토리지 서비스입니다.
- [Amazon SNS](#)—Amazon Simple Notification Service(SNS)는 애플리케이션, 최종 사용자 및 디바이스가 클라우드에서 즉시 알림을 전송하고 수신할 수 있게 해 주는 웹 서비스입니다.

코드

이 패턴에는 두 개의 파일이 포함된 첨부 파일이 포함됩니다.

- `index.zip`은(는) 이 패턴의 Lambda 코드가 포함된 압축 파일입니다.
- `ec2-require-tags.yaml` Lambda 코드를 배포하는 CloudFormation 템플릿입니다.

이러한 파일을 사용하는 방법에 대한 자세한 내용은 에픽 섹션을 참조하세요.

에픽

Lambda 코드 배포

작업	설명	필요한 기술
코드를 S3 버킷에 업로드합니다.	새 S3 버킷을 생성하거나 기존 S3 버킷을 사용하여 첨부한 <code>index.zip</code> 파일(Lambda 코드)을 업로드합니다. 이 버킷은 모니터링하려는 리소스(EC2 인스턴스)와 동일한 AWS 리전에 있어야 합니다.	클라우드 아키텍트
템플릿을 배포합니다. CloudFormation	S3 버킷과 동일한 AWS 리전에서 CloudFormation 콘솔을 열고 첨부 파일에 제공된 <code>ec2-require-tags.yaml</code> 파일을 배포합니다. 다음 에픽에서 템플릿 파라미터에 대한 값을 입력합니다.	클라우드 아키텍트

CloudFormation 템플릿의 파라미터를 완성하세요.

작업	설명	필요한 기술
S3 버킷 이름을 제공합니다.	첫 번째 에픽에서 생성하거나 선택한 S3 버킷의 이름을 입력합니다. 이 S3 버킷에는 Lambda 코드용 <code>zip</code> 파일이 들	클라우드 아키텍트

작업	설명	필요한 기술
	어 있으며 모니터링하려는 템플릿 및 EC2 인스턴스와 동일한 CloudFormation AWS 리전에 있어야 합니다.	
S3 키를 입력합니다.	S3 버킷에 있는 Lambda 코드 .zip 파일의 위치를 앞에 슬래시 없이 입력합니다(예: index.zip 또는 controls/index.zip).	클라우드 아키텍트
이메일 주소를 입력합니다.	위반 알림을 받을 사용 중인 이메일 주소를 입력합니다.	클라우드 아키텍트
로깅 수준을 정의합니다.	로깅 수준 및 상세 정보를 지정합니다. Info는 애플리케이션 진행 상황에 대한 자세한 정보 메시지를 지정하며 디버깅용으로만 사용해야 합니다. Error는 애플리케이션을 계속 실행할 수 있는 오류 이벤트를 지정합니다. Warning은 잠재적으로 유해한 상황을 지정합니다.	클라우드 아키텍트
필수 태그 키를 입력합니다.	확인할 태그 키를 입력합니다. 키를 여러 개 지정하려면 스페이스 없이 쉼표로 구분합니다. (예를 들어, ApplicationId, CreatedBy, Environment, Organization 은 네 개의 키를 검색합니다.) CloudWatch Events 이벤트는 이러한 태그 키를 검색하고 찾을 수 없는 경우 알림을 보냅니다.	클라우드 아키텍트

구독 확인

작업	설명	필요한 기술
이메일 구독을 확인합니다.	CloudFormation 템플릿이 성공적으로 배포되면 입력한 이메일 주소로 구독 이메일 메시지가 전송됩니다. 알림을 받기 시작하려면 이 이메일 구독을 확인해야 합니다.	클라우드 아키텍트

관련 리소스

- [버킷 생성](#)(Amazon S3 설명서)
- [객체 업로드](#)(Amazon S3 설명서)
- [Amazon EC2 리소스 태깅](#)(Amazon EC2 설명서)
- [AWS를 사용하여 AWS API 호출에서 트리거되는 CloudWatch 이벤트 규칙 생성 CloudTrail](#) (아마존 CloudWatch 설명서)

첨부

이 문서와 관련된 추가 콘텐츠에 액세스하려면 [attachment.zip](#) 파일의 압축을 풉니다.

Session Manager를 사용하여 Amazon EC2 인스턴스에 연결

작성자: Jason Cornick(AWS), Abhishek Bastikoppa(AWS) 및 Yaniv Ron(AWS)

환경: 프로덕션

기술: 인프라, 클라우드 네이티브, 최종 사용자 컴퓨팅, 운영

AWS 서비스: 아마존 CloudWatch 로그, AWS Systems Manager, 아마존 EC2

요약

이 패턴은 AWS Systems Manager의 기능인 Session Manager를 사용하여 Amazon Elastic Compute Cloud(Amazon EC2) 인스턴스에 연결하는 방법을 설명합니다. 이 패턴을 사용하면 웹 브라우저를 통해 EC2 인스턴스에서 bash 명령을 실행할 수 있습니다. Session Manager는 인바운드 포트를 열 필요가 없으며 EC2 인스턴스의 퍼블릭 IP 주소도 필요하지 않습니다. 또한 Bastion Host를 다른 Secure Shell(SSH) 키로 관리할 필요가 없습니다. AWS Identity and Access Management(IAM) 정책을 사용하여 Session Manager에 대한 액세스를 관리하고 인스턴스 액세스 및 작업과 같은 중요한 정보를 기록하는 로깅을 구성할 수 있습니다.

이 패턴에서는 IAM 역할을 구성하고 Amazon Machine Image(AMI)를 사용하여 프로비저닝한 Linux EC2 인스턴스에 연결합니다. 그런 다음 Amazon CloudWatch Logs에서 로깅을 구성하고 세션 관리자를 사용하여 인스턴스로 세션을 시작합니다.

이 패턴은 Amazon Web Services(AWS) 클라우드의 Linux EC2 인스턴스에 연결되지만 이 접근 방식을 사용하여 온 프레미스 서버 또는 다른 가상 머신과 같은 다른 서버와의 연결에 Session Manager를 사용할 수 있습니다.

사전 조건 및 제한 사항

사전 조건

- 활성 상태의 AWS 계정
- 관리형 노드에 액세스할 수 있는 권한. 지침은 [관리형 노드에 대한 사용자 세션 액세스 제어](#)를 참조하십시오.
- ssm, ec2, ec2messages, ssmessages 및 s3의 VPC 엔드포인트. 지침은 Systems Manager 설명서의 [VPC 엔드포인트 생성](#)을 참조하십시오.

아키텍처

대상 기술 스택

- Session Manager
- Amazon EC2
- CloudWatch 로그

대상 아키텍처

1. 사용자는 IAM을 통해 자신의 자격 증명과 보안 인증 정보를 인증합니다.
2. 사용자는 Session Manager를 통해 SSH 세션을 시작하고 EC2 인스턴스에 API 직접 호출을 보냅니다.
3. EC2 인스턴스에 설치된 AWS Systems Manager SSM 에이전트는 Session Manager에 연결하여 명령을 실행합니다.
4. 감사 및 모니터링 목적으로 세션 관리자는 로깅 데이터를 Logs로 보냅니다. CloudWatch 또는 로그 데이터를 Amazon Simple Storage Service(S3) 버킷으로 보낼 수 있습니다. 자세한 내용은 [Amazon S3를 사용하여 세션 데이터 로깅](#) (Systems Manager 설명서)을 참조하십시오.

도구

서비스

- [Amazon CloudWatch Logs](#)를 사용하면 모든 시스템, 애플리케이션 및 AWS 서비스의 로그를 중앙 집중화하여 로그를 모니터링하고 안전하게 보관할 수 있습니다.
- [Amazon Elastic Compute Cloud\(Amazon EC2\)](#)는 AWS 클라우드에서 확장 가능한 컴퓨팅 용량을 제공합니다. 필요한 만큼 가상 서버를 시작하고 빠르게 스케일 업하거나 스케일 다운할 수 있습니다. 이 패턴은 Amazon Machine Image(AMI)를 사용하여 Linux EC2 인스턴스를 프로비저닝합니다.
- [AWS Identity and Access Management\(IAM\)](#)를 사용하면 사용자에 대해 인증 및 권한 부여를 제어함으로써 AWS 리소스에 대한 액세스를 안전하게 관리할 수 있습니다.
- [AWS Systems Manager](#)는 AWS 클라우드에서 실행되는 애플리케이션과 인프라를 관리하는 데 도움이 됩니다. 애플리케이션 및 리소스 관리를 간소화하고, 운영 문제의 감지 및 해결 시간을 단축하며, AWS 리소스를 규모에 따라 안전하게 관리하는 데 도움이 됩니다. 이 패턴은 Systems Manager의 기능인 [Session Manager](#)를 사용합니다.

모범 사례

AWS Well-Architected Framework의 [보안 원칙](#)에 대해 자세히 읽고, 암호화 옵션을 살펴보며, [Session Manager 설정](#) (Systems Manager 설명서) 의 보안 권장 사항을 적용하는 것이 좋습니다.

에픽

인프라 설정

작업	설명	필요한 기술
IAM 역할을 생성합니다.	<p>SSM 에이전트의 IAM 역할을 생성합니다. AWS 서비스 역할 생성(IAM 설명서)의 지침을 따르고 다음 내용을 참고하십시오.</p> <ol style="list-style-type: none"> 1. AWS 서비스에서 EC2를 선택합니다. 2. 권한 정책에서 AmazonSSManagedInstanceCore 를 선택합니다. 3. 역할 이름에 EC2_SSM_Role 을 입력합니다. 	AWS 시스템 관리자
EC2 인스턴스를 생성합니다.	<ol style="list-style-type: none"> 1. EC2 인스턴스를 생성합니다. 인스턴스 시작(Amazon EC2 설명서)의 지침을 따르고 다음 내용을 참조하십시오. <ol style="list-style-type: none"> a. 이름 및 태그 섹션에서 추가 태그 추가를 선택합니다. [Key]에 Name을 입력하고 [Value]에 Production_Server_One 을 입력합니다. 	AWS 시스템 관리자

작업	설명	필요한 기술
	<p>b. SSM 에이전트가 사전 설치된 Amazon Linux AMI를 선택하십시오. 전체 목록은 SSM 에이전트가 사전 설치된 AMI(Systems Manager 설명서)를 참조하십시오.</p> <p>c. 고급 세부 정보 섹션의 IAM 인스턴스 프로파일에서 EC2_SSM_role을 선택합니다.</p> <p>2. https://console.aws.amazon.com/systems-manager/에서 Systems Manager 콘솔을 엽니다.</p> <p>3. 탐색 창에서 Fleet Manager를 선택합니다.</p> <p>4. 인스턴스가 관리형 노드 목록에 나타나는지 확인합니다.</p>	

작업	설명	필요한 기술
로깅을 설정합니다.	<ol style="list-style-type: none"> 로그에서 CloudWatch 로그 그룹을 생성합니다. 로그 그룹 만들기 (CloudWatch 로그 문서)의 지침을 따르세요. 새로운 로그 그룹 SessionManager 를 선택합니다. Session Manager에 대한 로깅을 구성합니다. Amazon CloudWatch Logs (Systems Manager 설명서) 를 사용하여 세션 데이터 로깅의 지침을 따르고 다음 내용을 참고하십시오. <ol style="list-style-type: none"> 암호화된 CloudWatch 로그 그룹만 허용을 선택하지 마십시오. 목록에서 로그 그룹 선택에서 을 선택합니다 SessionManager. 	AWS 시스템 관리자

인스턴스에 연결

작업	설명	필요한 기술
EC2 인스턴스에 연결합니다	<ol style="list-style-type: none"> Systems Manager 콘솔에서 세션을 시작합니다. 지침은 세션 시작(Systems Manager 설명서)을 참조하십시오. 대상 인스턴스의 경우 Production_Server_One 인스턴스 왼쪽에 있는 옵션 버튼을 선택합니다. 	AWS 시스템 관리자

작업	설명	필요한 기술
	<ol style="list-style-type: none"> 연결이 완료되면 여러 bash 명령을 실행합니다. Systems Manager 콘솔에서 세션을 종료합니다. 지침은 세션 종료(Systems Manager 설명서)를 참조하십시오. 	
로깅을 검증합니다.	<ol style="list-style-type: none"> CloudWatch 로그에서 로그 그룹의 로그 스트림을 엽니다. 지침은 로그 데이터 보기 (CloudWatch 로그 문서)를 참조하십시오. 로그 데이터에 이전 스토리에서 실행한 명령이 나열되어 있는지 확인하십시오. 	AWS 시스템 관리자

문제 해결

문제	Solution
IAM 문제	지원이 필요하다면 문제 해결 (IAM 설명서)을 참조하십시오.

관련 리소스

- [Complete Session Manager 사전 조건](#)(Systems Manager 설명서)
- [Amazon을 통한 로깅 및 모니터링 설계 및 구현 CloudWatch](#) (AWS Prictive Guidance)

AWS를 지원하지 않는 AWS 지역에 파이프라인을 생성하십시오. CodePipeline

작성자: Anand Krishna Varanasi

코드 리포지토리: [invisible-codepipeline-unsupported-regions](#)

환경: PoC 또는 파일럿

기술: 인프라, DevOps

AWS 서비스: AWS CodeBuild, AWS CodeCommit, AWS CodeDeploy, AWS CodePipeline

요약

CodePipeline AWS는 Amazon Web Services (AWS) 의 DevOps 도구 세트에 속하는 지속적 전송 (CD) 오케스트레이션 서비스입니다. 다양한 소스 (예: 버전 제어 시스템 및 스토리지 솔루션), AWS 및 AWS 파트너의 지속적 통합 (CI) 제품 및 서비스, 오픈 소스 제품과 통합되어 빠른 애플리케이션 및 인프라 배포를 위한 end-to-end 워크플로 서비스를 제공합니다.

하지만 모든 AWS 지역에서 지원되는 CodePipeline 것은 아니며, AWS CI/CD 서비스를 연결하는 보이지 않는 오케스트레이터가 있으면 유용합니다. 이 패턴은 AWS, AWS CodeBuild, AWS와 같은 AWS CI/CD 서비스를 사용하여 아직 지원되지 않는 AWS 지역에서 end-to-end 워크플로 파이프라인을 구현하는 방법을 설명합니다. CodePipeline CodeCommit CodeDeploy

사전 조건 및 제한 사항

사전 조건

- 활성 상태의 AWS 계정
- Cloud Development Kit(CDK) CLI 버전 2.28 이상

아키텍처

대상 기술 스택

다음 다이어그램은 아프리카 (케이프타운) 지역과 같이 지원하지 CodePipeline 애플리케이션 지역에서 생성된 파이프라인을 보여줍니다. 개발자가 CodeDeploy 구성 파일 (배포 라이프사이클 후크 스크립트라고도 함) 을 에서 호스팅하는 Git 리포지토리로 푸시합니다. CodeCommit (이 패턴과 함께 제공된 [GitHub 리포지토리를](#) 참조하십시오.) Amazon EventBridge 규칙이 자동으로 시작됩니다. CodeBuild

CodeDeploy 구성 파일은 파이프라인 소스 단계의 CodeCommit 일부로 가져와 파이프라인으로 전송됩니다. CodeBuild

다음 단계에서는 다음 CodeBuild 작업을 수행합니다.

1. 애플리케이션 소스 코드 TAR 파일을 다운로드합니다. AWS Systems Manager의 기능인 파라미터 스토어를 사용하여 이 파일의 이름을 구성할 수 있습니다.
2. CodeDeploy 구성 파일을 다운로드합니다.
3. 애플리케이션 유형별 애플리케이션 소스 코드와 CodeDeploy 구성 파일의 통합 아카이브를 생성합니다.
4. 결합된 아카이브를 사용하여 Amazon Elastic Compute Cloud (Amazon EC2) 인스턴스에 CodeDeploy 배포를 시작합니다.

도구

서비스

- [CodeBuildAWS](#)는 소스 코드를 컴파일하고, 단위 테스트를 실행하고, 배포할 준비가 된 아티팩트를 생성하는 데 도움이 되는 완전관리형 빌드 서비스입니다.
- [CodeCommitAWS](#)는 자체 소스 제어 시스템을 관리할 필요 없이 Git 리포지토리를 비공개로 저장하고 관리할 수 있는 버전 제어 서비스입니다.
- [AWS](#)는 Amazon EC2 또는 온프레미스 인스턴스, AWS Lambda 함수 또는 Amazon Elastic Container Service (Amazon ECS) 서비스로의 배포를 CodeDeploy 자동화합니다.
- [AWS](#)는 소프트웨어 릴리스의 여러 단계를 신속하게 모델링 및 구성하고 소프트웨어 변경 사항을 지속적으로 릴리스하는 CodePipeline 데 필요한 단계를 자동화하도록 지원합니다.
- [AWS Cloud Development Kit\(AWS CDK\)](#)는 AWS 클라우드 인프라를 코드로 정의하고 프로비저닝하는 데 도움이 되는 소프트웨어 개발 프레임워크입니다.

코드

이 패턴의 코드는 GitHub [CodePipeline 미지원 지역](#) 리포지토리에서 사용할 수 있습니다.

에픽

개발자 워크스테이션 설정

작업	설명	필요한 기술
CDK CLI를 설치합니다.	지침은 CDK 설명서 를 참조하세요.	AWS DevOps
Git 클라이언트를 설치합니다.	커밋을 만들려면 로컬 컴퓨터에 설치된 Git 클라이언트를 사용하여 커밋을 저장소로 푸시하면 됩니다. CodeCommit Git 클라이언트를 CodeCommit 사용하여 설정하려면 설명서를 참조하십시오. CodeCommit	AWS DevOps
npm을 설치합니다.	npm 패키지 매니저를 설치합니다. 자세한 내용은 npm 설명서 를 참조하세요.	AWS DevOps

파이프라인 설정

작업	설명	필요한 기술
코드 리포지토리를 복제합니다.	다음 명령을 실행하여 GitHub CodePipeline 지원되지 않는 지역 리포지토리를 로컬 시스템에 복제합니다. <pre>git clone https://github.com/aws-samples/invisible-code-pipeline-unsupported-regions</pre>	DevOps 엔지니어

작업	설명	필요한 기술
<p>cdk.json에서 파라미터를 설정합니다.</p>	<p>cdk.json 파일을 열고 다음 파라미터 값을 입력합니다.</p> <pre data-bbox="597 348 1027 1062"> "pipeline_account" :"XXXXXXXXXXXX", "pipeline_region":" us-west-2", "repo_name": "app-dev- repo", "ec2_tag_key": "test- vm", "configName" : "cbdeployconfig", "deploymentGroupNa me": "cbdeploygroup", "applicationName" : "cbdeployapplicati on", "projectName" : "CodeBuildProject" </pre> <p>여기서 각 항목은 다음과 같습니다.</p> <ul data-bbox="597 1230 1027 1877" style="list-style-type: none"> • pipeline_account (은)는 파이프라인이 빌드될 AWS 계정입니다. • pipeline_region (은)는 파이프라인이 빌드될 AWS 리전입니다. • repo_name CodeCommit 저장소의 이름입니다. • ec2_tag_key (은)는 코드를 배포하려는 EC2 인스턴스에 연결된 태그입니다. • configName CodeDeploy 구성 파일의 이름입니다. 	<p>AWS DevOps</p>

작업	설명	필요한 기술
<p>CDK 구성 라이브러리를 설정합니다.</p>	<ul style="list-style-type: none"> • deploymentGroupName CodeDeploy 배포 그룹의 이름입니다. • applicationName CodeDeploy 애플리케이션 이름입니다. • projectName CodeBuild 프로젝트 이름입니다. <p>복제된 GitHub 리포지토리에 서 다음 명령을 사용하여 AWS CDK 구성 라이브러리를 설치하고, 애플리케이션을 구축하고, 합성하여 애플리케이션용 AWS CloudFormation 템플릿을 생성합니다.</p> <pre>npm i aws-cdk-lib npm run build cdk synth</pre>	<p>AWS DevOps</p>
<p>샘플 CDK 애플리케이션을 배포합니다.</p>	<p>지원되지 않는 리전(예: af-south-1)에서 다음 명령을 실행하여 코드를 배포합니다.</p> <pre>cdk deploy</pre>	<p>AWS DevOps</p>

CodeCommit 리포지토리 설정: CodeDeploy

작업	설명	필요한 기술
<p>애플리케이션용 CI/CD를 설정합니다.</p>	<p>cdk.json파일에 지정한 CodeCommit 리포지토리 (app-dev-repo 기본적으로</p>	<p>AWS DevOps</p>

작업	설명	필요한 기술
	<p>호출됨) 를 복제하여 애플리케이션의 CI/CD 파이프라인을 설정합니다.</p> <pre data-bbox="594 380 1024 575">git clone https://github.com:aws-ia/cdk-bootstrap.git cd bootstrap git checkout v1.0.0 cd est-2 cd amazonaws.com cd v1/repos/app-dev-repo</pre> <p>여기서 리포지토리 이름과 리전은 cdk.json 파일에 입력한 값에 따라 달라집니다.</p>	

파이프라인 테스트

작업	설명	필요한 기술
<p>배포 지침으로 파이프라인을 테스트합니다.</p>	<p>GitHub CodePipeline 미 지원 지역 리포지토리의 CodeDeploy_Files 폴더에는 애플리케이션 배포를 CodeDeploy 지시하는 샘플 파일이 포함되어 있습니다. 이 appspec.yml 파일은 응용 프로그램 배포 흐름을 제어하는 후크가 포함된 CodeDeploy 구성 파일입니다. 샘플 파일인 index.html , start_server.sh , stop_server.sh , install_dependencies.sh (을)를 사용하여 Apache에서 호스팅되는 웹 사이트를 업데이트할 수 있습니다. 예를 들어 GitHub 리포지토리의 코드를 사용하</p>	<p>AWS DevOps</p>

작업	설명	필요한 기술
	여 모든 유형의 애플리케이션을 배포할 수 있습니다. 파일이 CodeCommit 리포지토리로 푸시되면 보이지 않는 파이프라인이 자동으로 시작됩니다. 배포 결과를 보려면 CodeBuild 및 CodeDeploy 콘솔에서 개별 단계의 결과를 확인하세요.	

관련 리소스

- [시작하기](#)(CDK 설명서)
- [Cloud Development Kit\(CDK\) 소개](#)(AWS 워크숍 스튜디오)
- [CDK .NET 워크숍](#)

프라이빗 고정 IP를 사용하여 Amazon EC2에 Cassandra 클러스터를 배포하여 리밸런싱 방지

작성자: 디핀 자인(AWS)

환경: PoC 또는 파일럿	소스: 온프레미스 VM	대상: Amazon EC2
R 유형: 리호스팅	워크로드: 오픈 소스	기술: 인프라, 데이터베이스, 마이그레이션
AWS 서비스: Amazon EC2		

요약

Amazon Elastic Compute Cloud(Amazon EC2) 인스턴스의 프라이빗 IP는 수명 주기 내내 유지됩니다. 하지만 Amazon Machine Image(AMI) 업그레이드와 같이 계획된 또는 예상치 못한 시스템 충돌 중에 프라이빗 IP가 변경될 수 있습니다. 일부 시나리오에서는 프라이빗 고정 IP를 유지하면 워크로드 성능 및 복구 시간을 개선할 수 있습니다. 예를 들어, Apache Cassandra 시드 노드에 고정 IP를 사용하면 클러스터에 리밸런싱 오버헤드가 발생하는 것을 방지할 수 있습니다.

이 패턴은 보조 엘라스틱 네트워크 인터페이스를 EC2 인스턴스에 연결하여 리호스팅 중에 IP를 고정 상태로 유지하는 방법을 설명합니다. 이 패턴은 또한 Cassandra 클러스터에 초점을 맞추지만, 이 구현 방법을 프라이빗 고정 IP의 이점을 활용하는 모든 아키텍처에 사용할 수 있습니다.

사전 조건 및 제한 사항

사전 조건

- 활성 Amazon Web Services(AWS) 계정

제품 버전

- DataStax 버전 5.11.1
- 운영 체제: Ubuntu 16.04.6 LTS

아키텍처

소스 아키텍처

소스는 온프레미스 가상 머신(VM)의 Cassandra 클러스터 또는 AWS 클라우드의 EC2 인스턴스일 수 있습니다. 다음 다이어그램은 이 두 번째 시나리오를 설명합니다. 이 예제에는 4개의 클러스터 노드(시드 노드 3개와 관리 노드 1개)가 포함됩니다. 소스 아키텍처에는 각 노드에 단일 네트워크 인터페이스가 연결되어 있습니다.

대상 아키텍처

대상 클러스터는 다음 다이어그램에 설명된 대로 각 노드에 보조 엘라스틱 네트워크 인터페이스가 연결된 EC2 인스턴스에서 호스팅됩니다.

자동화 및 규모 조정

또한 [AWS 지식 센터](#) 동영상에 설명된 대로 두 번째 엘라스틱 네트워크 인터페이스를 EC2 Auto Scaling 그룹에 자동으로 연결할 수 있습니다.

에픽

Amazon EC2에서 Cassandra 클러스터 구성

작업	설명	필요한 기술
EC2 노드를 실행하여 Cassandra 클러스터를 호스팅합니다.	Amazon EC2 콘솔 에서 AWS 계정의 Ubuntu 노드에 대해 EC2 인스턴스 4개를 시작합니다. 세 개의 (시드) 노드는 Cassandra 클러스터에 사용되고 네 번째 노드는 엔터프라이즈 (DSE) 를 설치할 DataStax 클러스터 관리 노드 역할을 합니다. OpsCenter 자세한 지침은 Amazon EC2 설명서 를 참조하세요.	클라우드 엔지니어

작업	설명	필요한 기술
노드 통신을 확인합니다.	네 개의 노드가 데이터베이스 및 클러스터 관리 포트를 통해 서로 통신할 수 있는지 확인하세요.	네트워크 엔지니어
관리 노드에 DSE를 OpsCenter 설치합니다.	데비안 패키지에서 DSE OpsCenter 6.1을 관리 노드에 설치합니다. 지침은 설명서를 참조하십시오. DataStax	DBA

작업	설명	필요한 기술
<p>보조 네트워크 인터페이스를 생성합니다.</p>	<p>Cassandra는 해당 노드에 대한 EC2 인스턴스의 IP 주소를 기반으로 각 노드에 대한 범용 고유 식별자(UUID)를 생성합니다. 이 UUID는 링에 가상 노드(vnode)를 배포하는 데 사용됩니다. Cassandra를 EC2 인스턴스에 배포하면 인스턴스가 생성될 때 IP 주소가 인스턴스에 자동으로 할당됩니다. 계획된 또는 예상치 못한 중단이 발생하는 경우, 새 EC2 인스턴스의 IP 주소가 변경되고 데이터 배포가 변경되며 링 전체를 재조정해야 합니다. 이는 바람직하지 않습니다. 할당된 IP 주소를 보존하려면 고정 IP 주소가 있는 보조 엘라스틱 네트워크 인터페이스를 사용합니다.</p> <ol style="list-style-type: none"> 1. Amazon EC2 콘솔에서 네트워크 인터페이스, 네트워크 인터페이스 생성을 선택합니다. 2. 서브넷의 경우, EC2 인스턴스를 생성한 서브넷을 선택합니다. 3. 프라이빗 IPv4 주소의 경우 자동 할당을 선택합니다. 4. 보안 그룹의 경우 보안 그룹을 선택한 다음 네트워크 인터페이스 생성을 선택합니다. 	<p>클라우드 엔지니어</p>

작업	설명	필요한 기술
	<p>네트워크 인터페이스를 생성하는 방법에 대한 자세한 내용은 Amazon EC2 설명서를 참조하세요.</p>	
<p>보조 네트워크 인터페이스를 클러스터 노드에 연결합니다.</p>	<ol style="list-style-type: none"> 1. Amazon EC2 콘솔에서 인스턴스를 선택합니다. 2. 이전에 생성한 EC2 인스턴스의 확인란을 선택합니다. 3. 작업, 네트워킹, 네트워크 인터페이스 연결을 차례로 선택합니다. 4. 이전 단계에서 생성한 네트워크 인터페이스를 선택한 다음 연결을 선택합니다. <p>네트워크 인터페이스 연결에 대한 자세한 내용은 Amazon EC2 설명서를 참조하세요.</p>	<p>클라우드 엔지니어</p>

작업	설명	필요한 기술
Amazon EC2에 경로를 추가하여 비대칭 라우팅을 처리합니다.	두 번째 네트워크 인터페이스를 연결하면 네트워크가 비대칭 라우팅을 수행할 가능성이 큼니다. 이를 방지하기 위해 새 네트워크 인터페이스에 경로를 추가할 수 있습니다. 비대칭 라우팅에 대한 자세한 설명과 해결 방법은 AWS Knowledge Center 등 영상 또는 멀티홉 서버에서의 비대칭 라우팅 극복하기 (McManusPatrick의 Linux Journal 기사, 2004년 4월 5일) 를 참조하십시오.	네트워크 엔지니어
보조 네트워크 인터페이스 IP를 가리키도록 DNS 항목을 업데이트합니다.	노드의 정규화된 도메인 이름(FQDN)이 보조 네트워크 인터페이스의 IP를 가리키도록 합니다.	네트워크 엔지니어
DSE를 사용하여 카산드라 클러스터를 설치하고 구성하십시오. OpsCenter	클러스터 노드에 보조 네트워크 인터페이스가 준비되면 Cassandra 클러스터를 설치하고 구성할 수 있습니다.	DBA

노드 장애로부터 클러스터 복구

작업	설명	필요한 기술
클러스터 시드 노드에 대한 AMI를 생성합니다.	노드 장애 발생 시 데이터베이스 바이너리로 복원할 수 있도록 노드를 백업합니다. 자세한 지침은 Amazon EC2 설명서의 AMI 생성 을 참조하세요.	백업 관리자

작업	설명	필요한 기술
노드 장애로부터 복구합니다.	장애가 발생한 노드를 AMI에서 시작된 새 EC2 인스턴스로 교체하고 장애가 발생한 노드의 보조 네트워크 인터페이스를 연결합니다.	백업 관리자
Cassandra 클러스터가 정상인지 확인합니다.	교체 노드가 가동되면 DSE에서 클러스터 상태를 확인합니다. OpsCenter	DBA

관련 리소스

- [데비안 패키지에서 DSE OpsCenter 6.1 설치 \(설명서\)](#) DataStax
- [Ubuntu EC2 인스턴스에서 보조 네트워크 인터페이스를 작동시키는 방법\(AWS 지식 센터 동영상\)](#)
- [Amazon EC2에서 Apache Cassandra 실행에 대한 모범 사례](#) (AWS 블로그 게시물)

AWS Transit Gateway Connect를 사용하여 VRF를 AWS로 확장

작성자: 애덤 킬 (AWS), 야샤르 아라기 (AWS), 비카스 드왕건 (AWS), 모히딘 (AWS) HajaMohideen

환경: PoC 또는 파일럿

기술: 인프라, 네트워킹

AWS 서비스: AWS Direct Connect, AWS Transit Gateway

요약

가상 라우팅 및 포워딩(VRF)은 기존 네트워크의 기능입니다. 라우팅 테이블 형태의 격리된 논리적 라우팅 도메인을 사용하여 동일한 물리적 인프라 내에서 네트워크 트래픽을 분리합니다. 온프레미스 네트워크를 AWS에 연결할 때 VRF 격리를 지원하도록 AWS Transit Gateway를 구성할 수 있습니다. 이 패턴은 샘플 아키텍처를 사용하여 온프레미스 VRF를 다양한 트랜짓 게이트웨이 라우팅 테이블에 연결합니다.

이 패턴은 AWS Direct Connect의 전송 가상 인터페이스(VIF)와 트랜짓 게이트웨이 Connect 첨부 파일을 사용하여 VRF를 확장합니다. [트랜짓 VIF](#)는 Direct Connect 게이트웨이와 연결된 하나 이상의 Amazon VPC 트랜짓 게이트웨이에 액세스하는 데 사용됩니다. [트랜짓 게이트웨이 Connect 첨부 파일](#)은 트랜짓 게이트웨이를 VPC에서 실행되는 타사 가상 어플라이언스와 연결합니다. 트랜짓 게이트웨이 Connect 첨부 파일은 고성능을 위한 Generic Routing Encapsulation(GRE) 터널 프로토콜과 동적 경로를 위한 Border Gateway Protocol(BGP)을 지원합니다.

이 패턴에 설명된 접근 방식은 다음과 같은 장점이 있습니다.

- Transit Gateway Connect를 사용하면 최대 1,000개의 경로를 Transit Gateway Connect 피어에 알리고 해당 피어로부터 최대 5,000개의 경로를 수신할 수 있습니다. Transit Gateway Connect를 사용하지 않고 Direct Connect 트랜짓 VIF 기능을 사용하는 것은 트랜짓 게이트웨이당 접두사 20개로 제한됩니다.
- 고객이 사용하는 IP 주소 스키마와 상관없이 트래픽 격리를 유지하고 Transit Gateway Connect를 사용하여 AWS에서 호스팅 서비스를 제공할 수 있습니다.
- VRF 트래픽은 퍼블릭 가상 인터페이스를 통과할 필요가 없습니다. 따라서 많은 조직의 규정 준수 및 보안 요구 사항을 더 쉽게 준수할 수 있습니다.
- 각 GRE 터널은 최대 5Gbps를 지원하며 트랜짓 게이트웨이 Connect 첨부 파일당 최대 4개의 GRE 터널을 사용할 수 있습니다. 이는 최대 1.25Gbps를 지원하는 AWS Site-to-Site VPN 연결과 같은 다른 많은 연결 유형보다 빠릅니다.

사전 조건 및 제한 사항

사전 조건

- 필수 AWS 계정이 생성되었습니다(자세한 내용은 아키텍처 참조).
- 각 계정에서 AWS Identity and Management(IAM) 역할을 맡을 수 있는 권한입니다.
- 각 계정의 IAM 역할에는 AWS Transit Gateway 및 AWS Direct Connect 리소스를 프로비저닝할 수 있는 권한이 있어야 합니다. 자세한 내용은 [트랜짓 게이트웨이의 인증 및 액세스 제어](#) 및 [Direct Connect의 자격 증명 및 액세스 관리](#)를 참조하십시오.
- Direct Connect 연결이 성공적으로 생성되었습니다. 자세한 내용은 [연결 마법사를 사용하여 연결 생성](#)을 참조하십시오.

제한 사항

- 프로덕션, QA 및 개발 계정의 VPC에 대한 Transit Gateway Attachment에는 제한이 있습니다. 자세한 내용은 [VPC에 대한 Transit Gateway Attachment](#)를 참조하십시오.
- 생성하고 사용할 수 있는 Direct Connect 게이트웨이 수에 제한이 있습니다. 자세한 내용은 [AWS Direct Connect 할당량](#)을 참조하십시오.

아키텍처

대상 아키텍처

다음 샘플 아키텍처는 트랜짓 게이트웨이 Connect 첨부파일이 있는 트랜짓 VIF를 배포하기 위한 재사용 가능한 솔루션을 제공합니다. 이 아키텍처는 여러 Direct Connect 위치를 사용하여 복원력을 제공합니다. 자세한 내용은 Direct Connect 문서의 [최대 복원력](#)을 참조하십시오. 온프레미스 네트워크에는 AWS로 확장되고 전용 라우팅 테이블을 사용하여 격리되는 프로덕션, QA 및 개발 VRF가 있습니다.

AWS 환경에서는 두 개의 계정, 즉 Direct Connect 계정과 네트워크 허브 계정이 VRF 확장 전용으로 사용됩니다. Direct Connect 계정에는 각 라우터의 연결 및 트랜짓 VIF가 포함됩니다. Direct Connect 계정에서 트랜짓 VIF를 생성하지만 네트워크 허브 계정에 배포하여 네트워크 허브 계정의 Direct Connect 게이트웨이와 연결할 수 있습니다. 네트워크 허브 계정에는 Direct Connect 게이트웨이와 트랜짓 게이트웨이가 포함되어 있습니다. AWS 리소스는 다음과 같이 연결됩니다.

1. 트랜짓 VIF는 Direct Connect 위치의 라우터를 Direct Connect 계정의 AWS Direct Connect와 연결합니다.

2. 트랜짓 VIF는 Direct Connect를 네트워크 허브 계정의 Direct Connect 게이트웨이와 연결합니다.
3. [트랜짓 게이트웨이 연결](#)은 Direct Connect 게이트웨이를 네트워크 허브 계정의 트랜짓 게이트웨이와 연결합니다.
4. [트랜짓 게이트웨이 Connect 첨부 파일](#)은 트랜짓 게이트웨이를 프로덕션, QA 및 개발 계정의 VPC와 연결합니다.

트랜짓 VIF 아키텍처

다음 다이어그램은 트랜짓 VIF에 대한 구성 세부 정보를 보여줍니다. 이 샘플 아키텍처는 터널 소스에 VLAN을 사용하지만 루프백을 사용할 수도 있습니다.

다음은 트랜짓 VIF의 구성 세부 정보(예: Autonomous System Number(ASN))입니다.

Resource	Item	세부 정보
라우터-01	ASN	65534
라우터-02	ASN	65534
라우터-03	ASN	65534
라우터-04	ASN	65534
Direct Connect 게이트웨이	ASN	64601
Transit Gateway	ASN	64600
	CIDR 블록	10.100.254.0/24

트랜짓 게이트웨이 Connect 아키텍처

다음 다이어그램과 표에서는 트랜짓 게이트웨이 Connect 첨부 파일을 통해 단일 VRF를 구성하는 방법을 설명합니다. 추가 VRF의 경우 CIDR 블록 내에 고유한 터널 ID, 트랜짓 게이트웨이 GRE IP 주소 및 BGP를 할당하십시오. 피어 GRE IP 주소는 트랜짓 VIF의 라우터 피어 IP 주소와 일치합니다.

다음 표는 라우터 구성 세부 정보를 포함합니다.

라우터	터널	IP 주소	소스(Source)	대상
라우터-01	터널 1	169.254.101.17	VLAN 60 169.254.100.1	10.100.254.1
라우터-02	터널 11	169.254.101.81	VLAN 61 169.254.100.5	10.100.254.11
라우터-03	터널 21	169.254.101.145	VLAN 62 169.254.100.9	10.100.254.21
라우터-04	터널 31	169.254.101.209	VLAN 63 169.254.100.13	10.100.254.31

다음 표는 라우터 구성 세부 정보를 포함합니다.

터널	Transit 게이트웨이 GRE IP 주소	피어 GRE IP 주소	CIDR 블록 내부의 BGP
터널 1	10.100.254.1	VLAN 60 169.254.100.1	169.254.101.16/29
터널 11	10.100.254.11	VLAN 61 169.254.100.5	169.254.101.80/29
터널 21	10.100.254.21	VLAN 62 169.254.100.9	169.254.101.144/29
터널 31	10.100.254.31	VLAN 63 169.254.100.13	169.254.101.208/29

배포

[예픽](#) 섹션에서는 여러 고객 라우터에 단일 VRF용 샘플 구성을 배포하는 방법을 설명합니다. 1~5단계를 완료한 후 AWS로 확장하는 모든 새 VRF에 대해 6~7단계를 사용하여 새 트랜짓 게이트웨이 Connect 첨부 파일을 생성할 수 있습니다.

1. 트랜짓 게이트웨이를 생성합니다.
2. 각 VRF의 Transit Gateway 라우팅 테이블을 생성합니다.
3. 트랜짓 가상 인터페이스를 생성합니다.
4. Direct Connect 게이트웨이를 생성합니다.
5. Direct Connect 게이트웨이 가상 인터페이스와 허용된 접두사를 사용하여 게이트웨이 연결을 생성합니다.
6. 트랜짓 게이트웨이 Connect 첨부 파일을 생성합니다.
7. Transit Gateway Connect 피어를 생성합니다.
8. 트랜짓 게이트웨이 Connect 첨부 파일을 라우팅 테이블에 연결합니다.
9. 라우터에 경로를 알립니다.

도구

서비스

- [Direct Connect](#)를 사용하면 표준 Ethernet 광섬유 케이블을 통해 내부 네트워크를 Direct Connect 위치에 연결할 수 있습니다. 이 연결을 구성하면 네트워크 경로에서 인터넷 서비스 제공업체를 우회하여 퍼블릭 AWS 서비스에 직접 가상 인터페이스를 생성할 수 있습니다.
- [AWS Transit Gateway](#)는 Virtual Private Cloud(VPC)와 온프레미스 네트워크를 연결하는 중앙 허브입니다.
- [Amazon Virtual Private Cloud\(VPC\)](#)를 이용하면 사용자가 정의한 가상 네트워크로 AWS 리소스를 시작할 수 있습니다. 이 가상 네트워크는 사용자의 자체 데이터 센터에서 운영하는 기존 네트워크와 유사하며 AWS의 확장 가능한 인프라를 사용한다는 이점이 있습니다.

에픽

아키텍처 계획

작업	설명	필요한 기술
<p>사용자 지정 아키텍처 다이어그램을 생성합니다.</p>	<ol style="list-style-type: none"> 1. 연결 섹션에서 다이어그램 템플릿을 다운로드합니다. 2. 첨부된 다이어그램을 Microsoft Office에서 엽니다 PowerPoint. 3. 아키텍처 개요 슬라이드에서 환경에 맞게 아키텍처 다이어그램을 사용자 지정합니다. AWS 환경으로 확장해야 하는 온프레미스 VRF를 식별하십시오. 4. Transit VIF 슬라이드에서 아키텍처 다이어그램을 사용자 정의합니다. 라우터의 AS 번호, Direct Connect 게이트웨이, 트랜짓 게이트웨이를 확인합니다. 트랜짓 VIF의 양쪽 끝에 있는 IP 주소를 식별합니다. 5. Transit Gateway Connect 슬라이드에서 각 VRF의 아키텍처 다이어그램을 사용자 지정합니다. 라우터 및 Transit Gateway Connect 피어를 구성하는 데 필요한 모든 필수 IP 주소를 식별합니다. 	<p>클라우드 아키텍트, 네트워크 관리자</p>

Transit Gateway 리소스 생성

작업	설명	필요한 기술
<p>트랜짓 게이트웨이를 생성합니다.</p>	<ol style="list-style-type: none"> 1. 네트워크 허브 계정에 로그인합니다. 2. 트랜짓 게이트웨이 생성 지침을 따르십시오. 이 패턴에 대해서는 다음을 참조하십시오. <ul style="list-style-type: none"> • Amazon 측 Autonomous System Number(ASN)에 고유한 ASN을 입력합니다. 이 예제에서 ASN은 64600입니다. • DNS 지원을 선택합니다. • 이 샘플 아키텍처에서는 VPN ECMP 지원, 기본 라우팅 테이블 연결, 기본 라우팅 테이블 비례 할당 및 멀티캐스트 지원이 필요하지 않습니다. • 트랜짓 게이트웨이 CIDR 블록에서 트랜짓 게이트웨이의 IPv4 블록을 입력합니다. 이 예제에서 CIDR 블록은 10.100.254.0/24 입니다. 	<p>네트워크 관리자, 클라우드 아키텍트</p>
<p>트랜짓 게이트웨이 라우팅 테이블을 생성합니다.</p>	<p>트랜짓 게이트웨이 라우팅 테이블 생성 지침을 따르십시오. 이 패턴에 대해서는 다음을 참조하십시오.</p> <ul style="list-style-type: none"> • (선택 사항) 이름 태그에 트랜짓 게이트웨이 라우팅 테 	<p>클라우드 아키텍트, 네트워크 관리자</p>

작업	설명	필요한 기술
	<p>이블의 이름을 입력합니다. VRF에 해당하는 이름(예: routetable-dev-vrf)을 사용하는 것이 좋습니다.</p> <ul style="list-style-type: none"> Transit Gateway ID에서 이전에 생성한 트랜짓 게이트웨이를 선택합니다. 	

트랜짓 가상 인터페이스 생성

작업	설명	필요한 기술
트랜짓 가상 인터페이스를 생성합니다.	<ol style="list-style-type: none"> Direct Connect 계정에 로그인합니다. Direct Connect 게이트웨이에 대한 전송 가상 인터페이스 생성 지침을 따르십시오. 이 패턴에 대해서는 다음을 참조하십시오. <ul style="list-style-type: none"> 가상 인터페이스 이름에 트랜짓 VIF의 이름을 입력합니다. 라우터에 해당하는 이름(예: transit-vif-router01)을 사용하는 것이 좋습니다. 연결에서 라우터를 선택합니다(예: router-01). 가상 인터페이스 소유자에 네트워크 허브 계정의 계정 ID를 입력합니다. 지침은 AWS 계정 ID 보기를 참조하십시오. 	클라우드 아키텍트, 네트워크 관리자

작업	설명	필요한 기술
	<ul style="list-style-type: none"> • Direct Connect 게이트웨이의 경우 아무 것도 선택하지 마십시오. 다음 단계에서 Direct Connect 게이트웨이를 연결합니다. • VLAN에 라우터의 VLAN을 입력합니다(예: 60). • BGP ASN에 라우터의 ASN(예: 65534)을 입력합니다. • 추가 설정에서 다음을 수행합니다: <ul style="list-style-type: none"> • IPv4를 선택합니다. • 라우터 피어 IP에 라우터 피어 IP 주소(예: 169.254.100.1)를 입력합니다. • Amazon 라우터 피어 IP에 Amazon 라우터 피어 IP(예: 169.254.100.2)를 입력합니다. • BGP 인증 키의 경우 암호가 필요합니다. 이 필드를 비워 두면 AWS는 이 계정에서만 액세스할 수 있는 키를 생성합니다. <p>3. 이 지침을 반복하여 VRF에 대한 모든 트랜짓 VIF를 생성하십시오.</p>	

Direct Connect 리소스 생성

작업	설명	필요한 기술
<p>Direct Connect 게이트웨이를 생성합니다.</p>	<ol style="list-style-type: none"> 1. 네트워크 허브 계정에 로그인합니다. 2. Direct Connect 게이트웨이 생성 지침을 따르십시오. 이 패턴에 대해서는 다음을 참조하십시오. <ul style="list-style-type: none"> • Amazon 측 ASN에 Direct Connect 게이트웨이의 ASN을 입력합니다 (예: 64601). • 가상 프라이빗 게이트웨이를 선택하지 마십시오. 	<p>클라우드 아키텍트, 네트워크 관리자</p>
<p>Direct Connect 게이트웨이를 트랜짓 VIF에 연결합니다.</p>	<ol style="list-style-type: none"> 1. 네트워크 허브 계정의 https://console.aws.amazon.com/directconnect/v2/에서 AWS Direct Connect 콘솔을 엽니다. 2. 탐색 창에서 가상 인터페이스를 선택합니다. 3. 새 트랜짓 VIF를 선택한 다음 수락을 선택합니다. 4. 생성한 Direct Connect 게이트웨이를 선택합니다. 5. 각 트랜짓 VIF에 대해 이 지침을 반복합니다. 	<p>클라우드 아키텍트, 네트워크 관리자</p>
<p>허용된 접두사를 사용하여 Direct Connect 게이트웨이 연결을 생성합니다.</p>	<p>네트워크 허브 계정에서 트랜짓 게이트웨이 연결하기의 지침을 따르십시오. 이 패턴에 대해서는 다음을 참조하십시오.</p>	<p>클라우드 아키텍트, 네트워크 관리자</p>

작업	설명	필요한 기술
	<ul style="list-style-type: none"> 게이트웨이에서 이전에 생성한 트랜짓 게이트웨이를 선택합니다. Allowed 접두사에 트랜짓 게이트웨이에 할당된 CIDR 블록(예: 10.100.254.0/24)을 입력합니다. <p>이 연결을 만들면 Direct Connect Gateway 리소스 유형을 가진 Transit Gateway attachment가 자동으로 생성됩니다. 이 연결을 트랜짓 게이트웨이 라우팅 테이블과 연결할 필요는 없습니다.</p>	

작업	설명	필요한 기술
<p>트랜짓 게이트웨이 Connect 첨부 파일을 생성합니다.</p>	<ol style="list-style-type: none"> 1. 네트워크 허브 계정의 https://console.aws.amazon.com/vpc/에서 Amazon VPC 콘솔을 엽니다. 2. 탐색 창에서 Transit Gateway Attachment를 선택합니다. 3. Transit Gateway Attachment 생성을 선택합니다. 4. 이름 태그에서 연결의 이름을 입력합니다. VRF에 해당하는 이름(예: PROD-VRF)을 사용하는 것이 좋습니다. 5. 트랜짓 게이트웨이 ID에서 이전에 생성한 트랜짓 게이트웨이를 선택합니다. 6. 연결 유형에서 Connect를 선택합니다. 7. 트랜짓 연결 ID에서 이전에 생성한 Direct Connect 게이트웨이를 선택합니다. 8. Transit Gateway Attachment 생성을 선택합니다. 9. 확장할 각 VRF에 대해 이 단계를 반복합니다. 	<p>클라우드 아키텍트, 네트워크 관리자</p>

작업	설명	필요한 기술
<p>Transit Gateway Connect 피어를 생성합니다.</p>	<p>1. 네트워크 허브 계정에서 Transit Gateway Connect 피어(GRE 터널) 생성 지침을 따르십시오. 이 패턴에 대해서는 다음을 참조하십시오.</p> <ul style="list-style-type: none"> • (선택 사항) 이름 태그에 Transit Gateway Connect 피어의 이름을 입력합니다. 라우터에 해당하는 이름(예connectpeer-router01 :)을 사용하는 것이 좋습니다. • 트랜짓 게이트웨이 GRE 주소에 트랜짓 게이트웨이 CIDR 블록에서 할당된 IP 주소(예: 10.100.254.1)를 입력합니다. • 피어 GRE 주소에 트랜짓 VIF를 위해 라우터에 생성된 VLAN에 할당된 IP 주소(예: 169.254.100.1)를 입력합니다. AWS가 IP 주소에 연결할 수 있다면 피어 GRE 주소로 VLAN 또는 Loopback 과 같은 인터페이스를 사용할 수 있습니다. • BGP 내부 CIDR 블록 (IPv4)에 BGP 내부 CIDR 블록 IP 주소(예: 169.254.101.16/29)를 입력합니다. 	

작업	설명	필요한 기술
	<ul style="list-style-type: none"> • 피어 ASN에 라우터의 ASN(예: 65534)을 입력합니다. <p>2. 이 지침을 반복하여 각 라우터에 대해 GRE 터널을 생성합니다.</p>	

라우터에 라우트 광고

작업	설명	필요한 기술
라우트를 광고합니다.	<p>새 트랜짓 게이트웨이 Connect 첨부 파일을 이 VRF에 대해 이전에 생성한 라우팅 테이블과 연결합니다. 예를 들어 프로덕션 트랜짓 게이트웨이 Connect 첨부 파일을 Production-VRF 라우팅 테이블에 연결합니다.</p> <p>라우터에 광고되는 접두사에 대한 정적 경로를 생성합니다.</p> <ol style="list-style-type: none"> 1. 네트워크 허브 계정에 로그인합니다. 2. https://console.aws.amazon.com/vpc/에서 Amazon VPC 콘솔을 여세요. 3. 탐색 창에서 Transit Gateways, 트랜짓 게이트웨이 라우팅 테이블을 선택합니다. 4. Production-VRF 라우팅 테이블을 선택합니다. 	네트워크 관리자, 클라우드 아키텍트

작업	설명	필요한 기술
	5. 작업 메뉴에서 정적 라우트 생성을 선택합니다. 6. CIDR에 대상 VPC의 Transit Gateway Attachment에 대한 광고 라우트의 CIDR 블록을 입력합니다(예: 10.100.1.0/24). 7. 연결 선택에서 관련 트랜짓 게이트웨이 Connect 첨부 파일을 선택합니다. 8. 정적 경로 생성을 선택합니다.	

관련 리소스

AWS 설명서

- Direct Connect 설명서
 - [Direct Connect 게이트웨이 사용](#)
 - [트랜짓 게이트웨이 연결](#)
 - [AWS Direct Connect 가상 인터페이스](#)
- Transit Gateway 설명서
 - [트랜짓 게이트웨이 사용](#)
 - [Direct Connect 게이트웨이에 대한 Transit Gateway Attachment](#)
 - [트랜짓 게이트웨이 Connect 첨부 파일 및 Transit Gateway Connect 피어](#)
 - [트랜짓 게이트웨이 Connect 첨부 파일 생성](#)

AWS 블로그 게시물

- [AWS Transit Gateway 연결을 통한 하이브리드 네트워크 세분화](#)
- [AWS Transit Gateway 연결을 사용하여 VRF를 확장하고 IP 접두사 광고를 늘리십시오](#)

첨부

이 문서와 관련된 추가 콘텐츠에 액세스하려면 [attachment.zip](#) 파일의 압축을 풉니다.

AWS KMS 키의 키 상태가 변경될 때 Amazon SNS 알림 받기

작성자: Shubham Harsora(AWS), Aromal Raj Jayarajan(AWS), Navdeep Pareek(AWS)

코드 리포지토리: aws-kms-deletion-notification	환경: PoC 또는 파일럿	기술: 인프라, 클라우드 네이티브, 보안 DevOps, ID, 규정 준수
워크로드: 기타 모든 워크로드	AWS 서비스: 아마존 EventBridge, AWS KMS, 아마존 SNS	

요약

AWS Key Management Service(AWS KMS) 키와 연결된 데이터 및 메타데이터는 해당 키를 삭제한 후에 손실됩니다. 삭제는 되돌릴 수 없으며 손실된 데이터(암호화된 데이터 포함)는 복구할 수 없습니다. AWS KMS 키의 [키 상태](#)에 대한 상태 변경을 알려주는 알림 시스템을 설정하여 데이터 손실을 방지할 수 있습니다.

이 패턴은 Amazon 및 Amazon Simple Notification Service (EventBridge Amazon SNS) 를 사용하여 AWS KMS 키의 키 상태가 또는 로 변경될 때마다 자동 알림을 발행하여 AWS KMS 키의 상태 변경을 모니터링하는 방법을 보여줍니다. Disabled PendingDeletion 예를 들어, 사용자가 AWS KMS 키를 비활성화하거나 삭제하려고 하면 시도된 상태 변경에 대한 세부 정보가 포함된 이메일 알림을 받게 됩니다. 또한 이 패턴을 사용하여 AWS KMS 키 삭제를 예약할 수 있습니다.

사전 조건 및 제한 사항

사전 조건

- AWS Identity and Access Management (IAM) 사용자가 있는 활성 AWS 계정
- [AWS KMS 키](#)

아키텍처

기술 스택

- 아마존 EventBridge

- AWS Key Management Service (AWS KMS)
- Amazon Simple Notification Service(SNS)

대상 아키텍처

다음 다이어그램은 AWS KMS 키의 상태 변경을 감지하기 위한 자동 모니터링 및 알림 프로세스를 구축하는 아키텍처를 보여줍니다.

이 다이어그램은 다음 워크플로우를 보여줍니다.

1. 사용자가 AWS KMS 키를 비활성화하거나 삭제를 예약합니다.
2. EventBridge 규칙은 예정된 PendingDeletion 이벤트 Disabled 또는 이벤트를 평가합니다.
3. 이 EventBridge 규칙은 Amazon SNS 주제를 호출합니다.
4. Amazon SNS가 사용자에게 이메일 알림 메시지를 보냅니다.

참고: 조직의 필요에 맞게 이메일 메시지를 사용자 지정할 수 있습니다. AWS KMS 키가 사용되는 엔티티에 대한 정보를 포함하는 것이 좋습니다. 이를 통해 사용자는 AWS KMS 키 삭제가 미치는 영향을 이해할 수 있습니다. 또한 AWS KMS 키가 삭제되기 하루나 이틀 전에 이메일 알림을 보내도록 예약할 수 있습니다.

자동화 및 규모 조정

AWS CloudFormation 스택은 이 패턴이 작동하는 데 필요한 모든 리소스와 서비스를 배포합니다. 단일 계정에서 독립적으로 패턴을 구현하거나 AWS Organizations의 여러 독립 계정 또는 [조직 단위에 CloudFormation StackSets](#) [AWS](#)를 사용하여 패턴을 구현할 수 있습니다.

도구

- [AWS는 AWS CloudFormation](#) 리소스를 설정하고, 빠르고 일관되게 프로비저닝하고, AWS 계정 및 AWS 지역의 수명 주기 전반에 걸쳐 리소스를 관리할 수 있도록 지원합니다. 이 패턴의 CloudFormation 템플릿은 원하는 모든 AWS 리소스를 설명하고 해당 리소스를 CloudFormation 프로비저닝 및 구성합니다.
- [EventBridgeAmazon](#)은 다양한 소스의 실시간 데이터와 애플리케이션을 연결하는 데 도움이 되는 서버리스 이벤트 버스 서비스입니다. EventBridge 자체 애플리케이션 및 AWS 서비스의 실시간 데이터 스트림을 제공하고 해당 데이터를 AWS Lambda와 같은 대상으로 라우팅합니다. EventBridge 이벤트 기반 아키텍처 구축 프로세스를 간소화합니다.

- [AWS Key Management Service\(AWS KMS\)](#)를 사용하면 암호화 키를 생성하고 제어하여 데이터를 보호할 수 있습니다.
- [Amazon Simple Notification Service\(Amazon SNS\)](#)를 사용하면 웹 서버 및 이메일 주소를 포함하여 게시자와 클라이언트 간의 메시지 교환을 조정하고 관리할 수 있습니다.

코드

이 패턴의 코드는 [AWS KMS 키 비활성화 및 예약 삭제 리포지토리 GitHub 모니터링에서](#) 사용할 수 있습니다.

에픽

템플릿을 배포하십시오. CloudFormation

작업	설명	필요한 기술
리포지토리를 복제합니다.	<p>다음 명령을 실행하여 GitHub Monitor AWS KMS 키 비활성화 및 예약 삭제 리포지토리를 로컬 시스템에 복제합니다.</p> <pre>git clone https://github.com/aws-samples/aws-kms-deletion-notification</pre>	AWS 관리자, 클라우드 아키텍트
템플릿의 파라미터를 업데이트합니다.	<p>코드 편집기에서 리포지토리에서 복제한 <code>Alerting-KMS-Events.yaml</code> CloudFormation 템플릿을 열고 다음 파라미터를 업데이트하십시오.</p> <ul style="list-style-type: none"> • <code>DestinationEmailAddress</code> 의 경우, SNS 알림을 수신하는 데 사용할 활성 이메일 주소를 입력합니다. 	AWS 관리자, 클라우드 아키텍트

작업	설명	필요한 기술
CloudFormation 템플릿을 배포합니다.	<ul style="list-style-type: none"> • SNSTopicName 의 경우 이름에 SNS 주제의 이름을 입력합니다. <ol style="list-style-type: none"> 1. AWS Management Console에 로그인하고 CloudFormation 콘솔을 엽니다. 2. 탐색 창에서 스택 생성을 선택한 다음 새 리소스 사용(표준)을 선택합니다. 3. 리소스 식별 페이지에서 다음을 선택합니다. 4. 템플릿 지정 페이지의 템플릿 소스에서 템플릿 파일 업로드를 선택합니다. 5. 파일 선택을 선택하고 복제된 GitHub 저장소에서 Alerting-KMS-Events.yaml 파일을 선택한 후 다음을 선택합니다. 6. 스택 이름에 스택 이름을 입력합니다. 7. 제출을 선택합니다. 	AWS 관리자, 클라우드 아키텍트

구독 확인

작업	설명	필요한 기술
구독 이메일을 확인합니다.	CloudFormation 템플릿이 성공적으로 배포되면 Amazon SNS는 CloudFormation 템플릿에 제공한 이메일 주소로 구독 확인 메시지를 보냅니다.	AWS 관리자, 클라우드 아키텍트

작업	설명	필요한 기술
	알림을 받기 시작하려면 이 이메일 구독을 확인해야 합니다. 자세한 내용은 Amazon SNS 개발자 가이드의 구독 확인 을 참조하세요.	

구독 알림 테스트

작업	설명	필요한 기술
AWS KMS 키를 비활성화합니다.	<ol style="list-style-type: none"> 1. AWS Management Console에 로그인하고 AWS KMS 콘솔을 엽니다. 2. 리전을 변경하려면 현재 표시된 리전의 이름을 선택한 다음 전환하려는 리전을 선택합니다. 3. 탐색 창에서 고객 관리형 키를 선택합니다. 4. 활성화 또는 비활성화하려는 AWS KMS 키의 확인란을 선택합니다. 5. AWS KMS 키를 비활성화하려면 키 작업을 선택한 후에 비활성화를 선택합니다. 	AWS 관리자
구독을 확인합니다.	Amazon SNS 알림 이메일 수신을 확인합니다.	AWS 관리자

리소스 정리

작업	설명	필요한 기술
CloudFormation 스택을 삭제합니다.	<ol style="list-style-type: none"> 1. AWS Management Console에 로그인하고 CloudFormation 콘솔을 엽니다. 2. 탐색 창에서 스택을 선택합니다. 3. 이전에 생성한 스택을 선택한 다음 삭제를 선택합니다. 	AWS 관리자

관련 리소스

- [AWS CloudFormation](#) (AWS 설명서)
- [AWS CloudFormation 콘솔에서 스택 생성](#) (AWS CloudFormation 설명서)
- [AWS에서 이벤트 기반 아키텍처 구축](#)(AWS 워크숍 스튜디오 설명서)
- [AWS Key Management Service 모범 사례](#)(AWS 백서)
- [AWS Key Management Service의 보안 모범 사례](#)(AWS KMS 개발자 가이드)

추가 정보

Amazon SNS는 기본적으로 전송 중 암호화를 제공합니다. 보안 모범 사례에 따라 AWS KMS 고객 관리 키를 사용하여 Amazon SNS의 서버 측 암호화를 활성화할 수도 있습니다.

메인프레임 현대화: 마이크로 포커스를 DevOps 활용한 AWS 기반

작성자: Kevin Yung(AWS)

소스: IBM z/OS Mainframe	대상: AWS	R 유형: 해당 없음
환경: PoC 또는 파일럿	기술: DevOps; 인프라	AWS 서비스: 아마존 EC2, AWS, AWS, CloudFormation AWS, CodeBuild AWS, CodeCommit AWS Systems Manager CodeDeploy, AWS CodePipeline

요약

고객의 과제

메인프레임 하드웨어에서 핵심 애플리케이션을 실행하는 조직은 일반적으로 디지털 혁신의 요구를 충족하기 위해 하드웨어를 확장해야 할 때 몇 가지 문제에 직면합니다. 이러한 문제에는 다음과 같은 제약이 포함됩니다.

- 메인프레임 하드웨어 구성 요소의 유연성이 낮고 변경 비용이 높아 메인프레임 개발 및 테스트 환경을 확장할 수 없습니다.
- 신규 개발자들은 기존 메인프레임 개발 도구에 익숙하지 않고 관심도 없기 때문에 메인프레임 개발은 기술 부족에 직면하고 있습니다. 메인프레임 개발에서는 컨테이너, 지속적 통합/지속적 전달(CI/CD) 파이프라인, 최신 테스트 프레임워크와 같은 최신 기술을 사용할 수 없습니다.

패턴 결과

이러한 문제를 해결하기 위해 Amazon Web Services(AWS)와 파트너 네트워크(APN) 파트너인 Micro Focus는 협력하여 이 패턴을 만들었습니다. 솔루션은 다음과 같은 결과를 달성하는 데 도움이 되도록 설계되었습니다.

- 개발자 생산성이 향상되었습니다. 몇 분 안에 개발자에게 새 메인프레임 개발 인스턴스를 제공할 수 있습니다.
- 클라우드를 사용하여 용량이 거의 무제한인 새로운 메인프레임 테스트 환경을 구축합니다.

- 새 메인프레임 CI/CD 인프라의 신속한 프로비저닝. AWS CloudFormation 및 AWS Systems Manager를 사용하면 한 시간 내에 AWS에서 프로비저닝을 완료할 수 있습니다.
- AWS, AWS, CodeBuild AWS, 아마존 Elastic Container Registry (Amazon ECR) CodeDeploy 등 메인프레임 개발을 위한 CodeCommit CodePipeline AWS DevOps 도구의 기본 사용
- 기존 워터폴 개발을 메인프레임 프로젝트의 애자일 개발로 전환합니다.

기술 요약

이 패턴에서 대상 스택에는 다음 구성 요소가 들어 있습니다.

논리적 구성 요소	구현 솔루션	설명
소스 코드 리포지토리	마이크로 포커스 AccuRev 서버 CodeCommit, 아마존 ECR	<p>소스 코드 관리 — 솔루션은 두 가지 유형의 소스 코드를 사용합니다.</p> <ul style="list-style-type: none"> • 메인프레임 소스 코드(예: COBOL, JCL 등) • AWS 인프라 템플릿 및 자동화 스크립트
Enterprise Developer 인스턴스	Amazon Elastic Compute Cloud(Amazon EC2), Micro	<p>두 가지 유형의 소스 코드는 모두 버전 제어가 필요하지만 서로 다른 SCM에서 관리됩니다. 메인프레임 또는 마이크로 포커스 엔터프라이즈 서버에 배포된 소스 코드는 마이크로 포커스 서버에서 관리됩니다. AccuRev AWS 템플릿과 자동화 스크립트는 에서 관리됩니다 CodeCommit. Amazon ECR 은 도커 이미지 리포지토리에 사용됩니다.</p> <p>메인프레임 개발자는 Micro Focus Enterprise Developer for Eclipse를 사용하여</p>

	Focus Enterprise Developer for Eclipse	Amazon EC2에서 코드를 개발할 수 있습니다. 따라서 메인프레임 하드웨어를 사용하여 코드를 작성하고 테스트할 필요가 없습니다.
Micro Focus 라이선스 관리	Micro Focus 라이선스 관리자	중앙 집중식 Micro Focus 라이선스 관리 및 거버넌스를 위해 솔루션은 Micro Focus License Manager를 사용하여 필요한 라이선스를 호스팅합니다.
CI/CD 파이프라인	CodePipeline, CodeBuild, CodeDeploy, 컨테이너의 마이크로 포커스 엔터프라이즈 개발자, 컨테이너의 마이크로 포커스 엔터프라이즈 테스트 서버, 마이크로 포커스 엔터프라이즈 서버	메인프레임 개발 팀은 코드 컴파일, 통합 테스트, 회귀 테스트를 수행하기 위해 CI/CD 파이프라인이 필요합니다. AWS에서는 CodePipeline 기본적으로 컨테이너에서 Micro Focus 엔터프라이즈 개발자 및 엔터프라이즈 테스트 서버와 함께 CodeBuild 작동할 수 있습니다.

사전 조건 및 제한 사항

사전 조건

명칭	설명
py3270	py3270은 IBM 3270 터미널 에뮬레이터인 x3270에 대한 Python 인터페이스입니다. x3270 또는 s3270 하위 프로세스에 API를 제공합니다.
x3270	x3270은 X Window System 및 Windows용 IBM 3270 터미널 에뮬레이터입니다. 개발자가 로컬에서 유닛 테스트를 수행하는 데 사용할 수 있습니다.

Robot-Framework-Mainframe-3270-Library	Mainframe3270은 py3270 프로젝트를 기반으로 하는 로봇 프레임워크용 라이브러리입니다.
Micro Focus Verastream	Micro Focus Verastream은 모바일 앱, 웹 애플리케이션, SOA 웹 서비스를 테스트하는 것과 같은 방식으로 메인프레임 자산을 테스트할 수 있는 통합 플랫폼입니다.
Micro Focus Unified Functional Testing(UFT) 설치 프로그램 및 라이선스	Micro Focus Unified Functional Testing는 소프트웨어 애플리케이션 및 환경을 위한 기능 및 회귀 테스트 자동화를 제공하는 소프트웨어입니다.
Micro Focus Enterprise Server 소프트웨어 및 라이선스.	Enterprise Server는 메인프레임 애플리케이션을 위한 런타임 환경을 제공합니다.
Micro Focus Enterprise Server 설치자와 라이선스.	Micro Focus Enterprise Test Server는 IBM 메인프레임 애플리케이션 테스트 환경입니다.
서버용 마이크로 포커스 AccuRev 설치 프로그램 및 라이선스, 윈도우 및 리눅스 운영 체제용 마이크로 포커스 AccuRev 설치 및 라이선스	AccuRev 소스 코드 관리 (SCM) 를 제공합니다. 이 AccuRev 시스템은 파일 세트를 개발하는 사람들로 구성된 팀이 사용할 수 있도록 설계되었습니다.
Micro Focus Enterprise Developer for Eclipse 설치 프로그램, 패치, 라이선스	Enterprise Developer는 메인프레임 개발자에게 코어 메인프레임 온라인 및 배치 애플리케이션을 개발하고 유지 관리할 수 있는 플랫폼을 제공합니다.

제한 사항

- 예서는 Windows Docker 이미지 빌드가 지원되지 않습니다. CodeBuild [보고된 이 문제](#)는 Windows Kernel/HCS 및 Docker 팀의 지원이 필요합니다. 해결 방법은 Systems Manager를 사용하여 도커 이미지 빌드 런북을 생성하는 것입니다. 이 패턴은 해결 방법을 사용하여 Micro Focus Enterprise Developer for Eclipse 및 Micro Focus Enterprise Test Server Container 이미지를 빌드합니다.
- Windows에서는 가상 사설 클라우드 (VPC) 연결이 아직 지원되지 않으므로 패턴에서는 Micro Focus License Manager를 사용하여 Micro Focus Enterprise Developer 및 Micro Focus 엔터프라이즈 테스트 서버 컨테이너의 라이선스를 관리하지 않습니다. CodeBuild

제품 버전

- Micro Focus Enterprise Developer 5.5 이상
- Micro Focus Enterprise Test Server 5.5 이상
- Micro Focus Enterprise Server 5.5 이상
- 마이크로 포커스 AccuRev 7.x 이상
- Micro Focus Enterprise Developer 및 Enterprise Test Server용 Windows Docker 베이스 이미지: microsoft/dotnet-framework-4.7.2-runtime
- AccuRev 클라이언트용 리눅스 도커 기본 이미지: 아마존 리눅스:2

아키텍처

메인프레임 환경

기존 메인프레임 개발에서는 개발자가 메인프레임 하드웨어를 사용하여 프로그램을 개발하고 테스트해야 합니다. 예를 들어 개발/테스트 환경에서는 초당 백만 개의 명령(MIPS)으로 제한되는 등 용량 제한에 직면하며 메인프레임 컴퓨터에서 사용할 수 있는 도구에 의존해야 합니다.

많은 조직에서 메인프레임 개발은 워터폴 개발 방법론을 따르며, 팀은 긴 주기에 의존하여 변경 사항을 배포합니다. 이러한 릴리스 주기는 일반적으로 디지털 제품 개발보다 더 깁니다.

다음 다이어그램은 개발을 위해 메인프레임 하드웨어를 공유하는 여러 메인프레임 프로젝트를 보여줍니다. 메인프레임 하드웨어에서는 더 많은 프로젝트를 위해 개발 및 테스트 환경을 확장하는 데 많은 비용이 듭니다.

AWS 아키텍처

이 패턴은 메인프레임 개발을 AWS 클라우드로 확장합니다. 먼저 Micro Focus AccuRev SCM을 사용하여 AWS에서 메인프레임 소스 코드를 호스팅합니다. 그러면 메인프레임 코드를 빌드하고 테스트하는 데 Micro Focus Enterprise Developer 및 Micro Focus Enterprise Test Server를 사용할 수 있습니다.

다음 섹션에서는 패턴의 세 가지 주요 구성 요소에 대해 설명합니다.

1: SCM

AWS에서 패턴은 Micro AccuRev Focus를 사용하여 메인프레임 소스 코드에 대한 SCM 작업 영역 및 버전 제어 세트를 생성합니다. 스트림 기반 아키텍처를 통해 여러 팀을 위한 병렬 메인프레임 개발이 가능합니다. 변경 내용을 병합하려면 프로모션 개념을 AccuRev 사용합니다. 해당 변경 내용을 다른 작업 영역에 추가하려면 업데이트 개념을 AccuRev 사용합니다.

프로젝트 수준에서 각 팀은 하나 이상의 스트림을 만들어 프로젝트 수준 변경 사항을 AccuRev 추적할 수 있습니다. 이를 프로젝트 스트림이라고 합니다. 이러한 프로젝트 스트림은 동일한 상위 스트림에서 상속됩니다. 상위 스트림은 여러 프로젝트 스트림의 변경 사항을 병합하는 데 사용됩니다.

각 프로젝트 스트림은 코드를 승격할 수 있으며 AccuRev, 프로모션 사후 트리거는 AWS CI/CD 파이프라인을 시작하도록 설정됩니다. 프로젝트 스트림 변경을 위한 성공적인 빌드는 추가 회귀 테스트를 위해 상위 스트림으로 승격될 수 있습니다.

일반적으로 상위 스트림을 시스템 통합 스트림이라고 합니다. 프로젝트 스트림에서 시스템 통합 스트림으로 승격이 있을 경우 사후 승격 트리거는 다른 CI/CD 파이프라인을 시작하여 회귀 테스트를 실행합니다.

이 패턴에는 메인프레임 코드 외에도 AWS CloudFormation 템플릿, Systems Manager 자동화 문서 및 스크립트가 포함됩니다. infrastructure-as-code 모범 사례에 따라 AWS에서는 버전이 제어됩니다. CodeCommit

배포를 위해 메인프레임 코드를 메인프레임 환경으로 다시 동기화해야 하는 경우 Micro Focus는 SCM의 코드를 다시 메인프레임 SCM으로 동기화하는 엔터프라이즈 동기화 솔루션을 제공합니다.

AccuRev

2. 개발자 및 테스트 환경

대규모 조직에서 100명 이상 또는 1,000명 이상의 메인프레임 개발자를 확대하는 것은 어려운 일입니다. 이러한 제약을 해결하기 위해 패턴은 Amazon EC2 Windows 인스턴스를 개발에 사용합니다. 인스턴스에는 Micro Focus Enterprise Developer for Eclipse 도구가 설치되어 있습니다. 개발자는 인스턴스에서 로컬로 모든 메인프레임 코드 테스트 및 디버깅을 수행할 수 있습니다.

AWS Systems Manager State Manager 및 Automation 문서는 개발자 인스턴스 프로비저닝을 자동화하는 데 사용됩니다. 개발자 인스턴스를 생성하는 데 걸리는 평균 시간은 15분 이내입니다. 다음과 같은 소프트웨어 및 구성이 준비되어 있습니다.

- AccuRev 소스 코드를 체크아웃하고 커밋하기 위한 Windows 클라이언트 AccuRev
- 로컬에서 메인프레임 코드를 작성, 테스트, 디버깅하기 위한 Micro Focus Enterprise Developers for Eclipse 도구

- 애플리케이션을 테스트하는 스크립트를 생성하기 위한 오픈 소스 테스트 프레임워크 Python 동작 기반 개발(BDD) 테스트 프레임워크 Behave, py3270, x3270 에뮬레이터
- Enterprise Test Server Docker 컨테이너에서 Enterprise Test Server 도커 이미지를 빌드하고 애플리케이션을 테스트하기 위한 도커 개발자 도구

개발 주기에서 개발자는 EC2 인스턴스를 사용하여 로컬에서 메인프레임 코드를 개발하고 테스트합니다. 로컬 변경 사항을 성공적으로 테스트하면 개발자가 변경 내용을 AccuRev 서버로 프로모션합니다.

3. CI/CD 파이프라인

패턴에서 CI/CD 파이프라인은 프로덕션 환경에 배포하기 전에 통합 테스트 및 회귀 테스트에 사용됩니다.

SCM 섹션에 설명된 대로 에서는 프로젝트 스트림과 통합 스트림이라는 두 가지 유형의 스트림을 AccuRev 사용합니다. 각 스트림은 CI/CD 파이프라인과 연결됩니다. 이 패턴은 AccuRev 서버와 AWS CodePipeline 간의 통합을 수행하기 위해 AccuRev 사후 프로모션 스크립트를 사용하여 이벤트를 생성하여 CI/CD를 시작합니다.

예를 들어, 개발자가 에서 프로젝트 스트림의 변경 사항을 프로모션하면 AccuRev 서버에서 실행할 사후 프로모션 스크립트를 시작합니다. AccuRev 그런 다음 스크립트는 변경 사항의 메타데이터를 Amazon Simple Storage Service(S3) 버킷에 업로드하여 Amazon S3 이벤트를 생성합니다. 이 이벤트는 CodePipeline 구성된 파이프라인 실행을 시작합니다.

통합 스트림 및 관련 파이프라인에도 동일한 이벤트 시작 메커니즘이 사용됩니다.

CI/CD 파이프라인에서 Micro Focus AccuRev Linux 클라이언트 CodeBuild 컨테이너와 함께 CodePipeline 사용하여 스트림의 최신 코드를 체크아웃합니다. AccuRev 그런 다음 파이프라인은 Micro Focus Enterprise Developer Windows 컨테이너를 사용하여 소스 코드를 컴파일하고 Micro Focus 엔터프라이즈 테스트 서버 Windows 컨테이너를 사용하여 메인프레임 애플리케이션을 CodeBuild 테스트하기 시작합니다 CodeBuild .

CI/CD 파이프라인은 AWS CloudFormation 템플릿을 사용하여 구축되며 청사진은 새 프로젝트에 사용됩니다. 템플릿을 사용하면 프로젝트가 새 CI/CD 파이프라인을 생성하는 데 1시간도 걸리지 않습니다.

AWS에서 메인프레임 테스트 기능을 확장하기 위해 패턴은 마이크로 포커스 DevOps 테스트 스위트, 마이크로 포커스 베라스트림 및 마이크로 포커스 UFT 서버를 구축합니다. 최신 DevOps 도구를 사용하면 AWS에서 필요한 만큼 테스트를 실행할 수 있습니다.

Micro Focus on AWS를 사용하는 메인프레임 개발 환경의 예제는 다음 다이어그램에 나와 있습니다.

대상 기술 스택

이 섹션에서는 패턴에 있는 각 구성 요소의 아키텍처를 자세히 살펴봅니다.

1: 소스 코드 리포지토리 — AccuRev SCM

Micro Focus AccuRev SCM은 메인프레임 소스 코드 버전을 관리하도록 설정됩니다. 고가용성을 위해 기본 및 AccuRev 복제 모드를 지원합니다. 운영자는 프라이머리 노드에서 유지 관리를 수행할 때 복제본으로 페일오버할 수 있습니다.

CI/CD 파이프라인의 응답 속도를 높이기 위해 패턴은 Amazon CloudWatch Events를 사용하여 소스 코드 변경을 감지하고 파이프라인 시작을 시작합니다.

1. CodePipeline 은 Amazon S3 소스를 사용하도록 설정되어 있습니다.
2. CloudWatch 이벤트 규칙은 소스 S3 버킷에서 S3 이벤트를 캡처하도록 설정됩니다.
3. CloudWatch 이벤트 규칙은 대상을 파이프라인에 설정합니다.
4. AccuRev SCM은 프로모션이 완료된 후 사후 프로모션 스크립트를 로컬에서 실행하도록 구성되어 있습니다.
5. AccuRev SCM은 프로모션의 메타데이터가 포함된 XML 파일을 생성하고 스크립트는 XML 파일을 원본 S3 버킷에 업로드합니다.
6. 업로드 후 원본 S3 버킷은 이벤트 규칙과 일치하는 CloudWatch 이벤트를 전송하고 CloudWatch 이벤트 규칙이 실행을 시작합니다. CodePipeline

파이프라인이 실행되면 AccuRev Linux 클라이언트 컨테이너를 사용하여 관련 스트림에서 최신 메인프레임 코드를 체크아웃하는 CodeBuild 프로젝트를 시작합니다. AccuRev

다음 다이어그램은 서버 설정을 보여줍니다. AccuRev

2. Enterprise Developer 템플릿

이 패턴은 Amazon EC2 템플릿을 사용하여 개발자 인스턴스 생성을 간소화합니다. State Manager를 사용하면 소프트웨어 및 라이선스 설정을 EC2 인스턴스에 일관되게 적용할 수 있습니다.

Amazon EC2 템플릿은 VPC 컨텍스트 설정 및 기본 인스턴스 설정을 기반으로 빌드되며 엔터프라이즈 태깅 요구 사항을 따릅니다. 템플릿을 사용하면 팀에서 자체 신개발 인스턴스를 생성할 수 있습니다.

개발자 인스턴스가 시작되면 Systems Manager는 태그와 연결하여 State Manager를 사용해 자동화를 적용합니다. 자동화에는 다음과 같은 일반 절차가 포함됩니다.

1. Micro Focus Enterprise Developer 소프트웨어를 설치하고 패치를 설치합니다.
2. 윈도우용 마이크로 포커스 AccuRev 클라이언트를 설치합니다.
3. 개발자가 AccuRev 스트림에 참여할 수 있도록 사전 구성된 스크립트를 설치합니다. Eclipse 워크스페이스를 초기화합니다.
4. x3270, py3270, 도커를 포함한 개발 도구를 설치합니다.
5. Micro Focus License Manager 로드 밸런서를 가리키도록 라이선스 설정을 구성합니다.

다음 다이어그램은 State Manager가 인스턴스에 소프트웨어 및 구성을 적용하면서 Amazon EC2 템플릿으로 생성한 엔터프라이즈 개발자 인스턴스를 보여줍니다. 엔터프라이즈 개발자 인스턴스는 Micro Focus License Manager에 연결하여 라이선스를 활성화합니다.

3. CI/CD 파이프라인

AWS 아키텍처 섹션에서 설명했듯이 패턴에는 프로젝트 수준의 CI/CD 파이프라인과 시스템 통합 파이프라인이 있습니다. 각 메인프레임 프로젝트 팀은 프로젝트에서 개발 중인 프로그램을 빌드하기 위한 파이프라인 또는 여러 CI/CD 파이프라인을 생성합니다. 이 프로젝트 CI/CD 파이프라인은 관련 스트림에서 소스 코드를 체크아웃합니다. AccuRev

프로젝트 팀에서 개발자는 관련 스트림에서 코드를 홍보합니다. AccuRev 그런 다음 승격을 통해 코드를 빌드하고 테스트를 실행 및 통합하기 위한 프로젝트 파이프라인이 시작됩니다.

각 프로젝트 CI/CD 파이프라인은 마이크로 포커스 엔터프라이즈 개발자 도구 Amazon ECR 이미지 및 마이크로 포커스 엔터프라이즈 테스트 서버 도구 Amazon ECR 이미지가 있는 CodeBuild 프로젝트를 사용합니다.

CodePipeline CI/CDS 파이프라인을 생성하는 데 CodeBuild 사용됩니다. 선결제 금액이나 약정이 CodePipeline 없으므로 사용한 만큼만 비용을 지불하면 됩니다. CodeBuild 메인프레임 하드웨어에 비해 AWS 솔루션은 하드웨어 프로비저닝 리드 타임을 크게 줄이고 테스트 환경 비용을 낮춥니다.

현대의 개발에서는 여러 테스트 방법론이 사용됩니다. 테스트 기반 개발(TDD), BDD, 로봇 프레임워크 등을 예로 들 수 있습니다. 이 패턴을 통해 개발자는 이러한 최신 도구를 메인프레임 테스트에 사용할 수 있습니다. 예를 들어 x3270, py3270, Behave python 테스트 도구를 사용하여 온라인 애플리케이션의 동작을 정의할 수 있습니다. 또한 이러한 CI/CD 파이프라인에서 메인프레임 3270 로봇 프레임워크를 빌드하는 데에도 사용할 수 있습니다.

다음 다이어그램은 팀 스트림 CI/CD 파이프라인을 보여줍니다.

다음 다이어그램은 메인프레임3270 로봇 CodePipeline 프레임워크에서 생성한 프로젝트 CI/CD 테스트 보고서를 보여줍니다.

다음 다이어그램은 Py3270 및 Behave BDD에서 생성한 프로젝트 CI/CD 테스트 보고서를 보여줍니다. CodePipeline

프로젝트 수준 테스트가 성공적으로 통과되면 테스트된 코드가 SCM의 통합 스트림으로 수동으로 승격됩니다. AccuRev 팀이 프로젝트 파이프라인의 테스트 적용 범위에 대해 신뢰를 가진 후에 이 단계를 자동화할 수 있습니다.

코드가 승격되면 시스템 통합 CI/CD 파이프라인이 병합된 코드를 체크아웃하고 회귀 테스트를 수행합니다. 병합된 코드는 모든 병행 프로젝트 스트림에서 승격됩니다.

필요한 테스트 환경의 세분화 정도에 따라 고객은 UAT, Pre-Production과 같은 다양한 환경에서 더 많은 시스템 통합 CI/CD 파이프라인을 보유할 수 있습니다.

패턴에서는 시스템 통합 파이프라인에 사용되는 도구가 Micro Focus Enterprise Test Server, Micro Focus UFT Server, Micro Focus Verastream입니다. 이러한 모든 도구를 Docker 컨테이너에 배포하여 함께 사용할 수 있습니다. CodeBuild

메인프레임 프로그램을 성공적으로 테스트한 후 아티팩트는 버전 제어와 함께 S3 버킷에 저장됩니다.

다음 다이어그램은 시스템 통합 CI/CD 파이프라인을 보여줍니다.

아티팩트가 시스템 통합 CI/CD 파이프라인에서 성공적으로 테스트되면 프로덕션 배포를 위해 승격될 수 있습니다.

소스 코드를 메인프레임에 다시 배포해야 하는 경우 Micro Focus는 소스 코드를 AccuRev 다시 메인프레임 엔데버로 동기화하는 Enterprise Sync 솔루션을 제공합니다.

다음 다이어그램은 아티팩트를 Micro Focus Enterprise Server에 배포하는 프로덕션 CI/CD 파이프라인을 보여줍니다. 이 예제에서는 테스트된 메인프레임 아티팩트를 Micro CodeDeploy Focus Enterprise Server에 배포하는 과정을 오케스트레이션합니다.

CI/CD 파이프라인의 아키텍처 설명 외에도 [Micro Focus Enterprise Suite를 사용하여 AWS에서 수천 개의 메인프레임 테스트를 자동화하는 AWS DevOps 블로그 게시물을 읽고 및 에서 메인프레임 애플리케이션을 테스트하는 방법에 대한 자세한 내용을 확인할 수 있습니다.](#) CodeBuild CodePipeline 메인프레임 테스트 수행에 대한 모범 사례 및 세부 정보는 블로그 게시물을 참조하십시오.

도구

도구

자동화 도구

- [AWS CloudFormation](#)
- [아마존 CloudWatch 이벤트](#)
- [AWS CodeBuild](#)
- [AWS CodeDeploy](#)
- [AWS CodePipeline](#)
- [Amazon ECR](#)
- [Amazon S3](#)
- [AWS Secrets Manager](#)
- [AWS Systems Manager](#)

Micro Focus 도구

- [Micro Focus Enterprise Developer for Eclipse](#)
- [Micro Focus Enterprise Test Server](#)

- [Micro Focus Enterprise Server](#)(프로덕션 배포)
- [Micro Focus AccuRev](#)
- [Micro Focus License Manager](#)
- [Micro Focus Verastream Host Integrator](#)
- [Micro Focus UFT One](#)

기타 도구

- x3270
- [py3270](#)
- [Robot-Framework-Mainframe-3270-Library](#)

에픽

AccuRev SCM 인프라를 생성하십시오.

작업	설명	필요한 기술
CloudFormationAWS를 사용하여 기본 AccuRev SCM 서버를 배포하십시오.		AWS CloudFormation
AccuRev 관리자 사용자 생성.	AccuRev SCM Server에 로그인하고 CLI 명령을 실행하여 관리자 사용자를 생성합니다.	AccuRev SCM 서버 관리자
AccuRev 스트림 생성.	프로덕션, 시스템 통합, 팀 스트림 등 상위 스트림에서 순서대로 상속되는 스트림을 AccuRev 생성하세요.	AccuRev SCM 관리자
개발자 AccuRev 로그인 계정을 생성합니다.	AccuRev SCM CLI 명령을 사용하여 메인프레임 개발자를 위한 사용자 로그인 계정을 AccuRev 생성합니다.	AccuRev SCM 관리자

Enterprise Developer Amazon EC2 시작 템플릿 생성

작업	설명	필요한 기술
AWS를 사용하여 Amazon EC2 시작 템플릿을 배포하십시오. CloudFormation	AWS를 CloudFormation 사용하여 마이크로 포커스 엔터프라이즈 개발자 인스턴스용 Amazon EC2 시작 템플릿을 배포하십시오. 템플릿에는 Micro Focus Enterprise Developer 인스턴스에 대한 Systems Manager Automation 문서가 포함됩니다.	AWS CloudFormation
Amazon EC2 시작 템플릿에서 Enterprise Developer 인스턴스를 생성합니다.		콘솔 로그인 및 메인프레임 개발자 기술

Micro Focus Enterprise Developer 도구 도커 이미지 생성

작업	설명	필요한 기술
Micro Focus Enterprise Developer 도구 Docker 이미지를 생성합니다.	Docker 명령과 Micro Focus Enterprise Developer 도구 Dockerfile을 사용하여 도커 이미지를 생성합니다.	Docker
Amazon ECR에서 Docker 리포지토리를 생성합니다.	Amazon ECR 콘솔에서 Micro Focus Enterprise Developer 도구 이미지를 위한 리포지토리를 생성합니다.	Amazon ECR
Micro Focus Enterprise Developer 도구 Docker 이미지를 생성합니다.	Docker 푸시 명령을 실행하여 Enterprise Developer 도구 도커 이미지를 푸시해서 Amazon ECR의 Docker 리포지토리에 저장합니다.	Docker

Micro Focus Enterprise Test Server Docker 이미지 생성

작업	설명	필요한 기술
Micro Focus Enterprise Test Server 도커 이미지를 생성합니다.	Docker 명령과 Micro Focus Enterprise Test Server Dockerfile을 사용하여 도커 이미지를 생성합니다.	Docker
Amazon ECR에서 Docker 리포지토리를 생성합니다.	Amazon ECR 콘솔에서 Micro Focus Enterprise Test Server Docker image를 위한 Amazon ECR리포지토리를 생성합니다.	Amazon ECR
Micro Focus Enterprise Test Server Docker 이미지를 Amazon ECR로 푸시합니다.	Docker 푸시 명령을 실행하여 Enterprise Test Server 도커 이미지를 푸시해서 Amazon ECR에 저장합니다.	Docker

팀 스트림 CI/CD 파이프라인을 생성합니다.

작업	설명	필요한 기술
AWS CodeCommit 리포지토리를 생성합니다.	CodeCommit 콘솔에서 인프라 및 AWS CloudFormation 코드를 위한 Git 기반 리포지토리를 생성합니다.	AWS CodeCommit
AWS CloudFormation 템플릿과 자동화 코드를 CodeCommit 리포지토리에 업로드합니다.	Git push 명령을 실행하여 AWS CloudFormation 템플릿과 자동화 코드를 리포지토리에 업로드합니다.	Git
를 통해 팀 스트림 CI/CD 파이프라인을 배포하십시오. CloudFormation	준비된 AWS CloudFormation 템플릿을 사용하여 팀 스트림 CI/CD 파이프라인을 배포하십시오.	AWS CloudFormation

시스템 통합 CI/CD 파이프라인을 생성합니다.

작업	설명	필요한 기술
Micro Focus UFT 도커 이미지를 생성합니다.	Docker 명령과 Micro Focus UFT Dockerfile을 사용하여 Micro Focus 도커 이미지를 생성합니다.	Docker
Amazon ECR에서 Micro Focus UFT 이미지를 위한 Docker 리포지토리를 생성합니다.	Amazon ECR 콘솔에서 Micro Focus UFT 이미지를 위한 Docker 리포지토리를 생성합니다.	Amazon ECR
Micro Focus UFT 도커 이미지를 Amazon ECR로 푸시합니다.	Docker 푸시 명령을 실행하여 Enterprise Test Server 도커 이미지를 푸시해서 Amazon ECR에 저장합니다.	Docker
Micro Focus Verastream Docker 이미지를 생성합니다.	Docker 명령과 Micro Focus Verastream Dockerfile을 사용하여 도커 이미지를 생성합니다.	Docker
Micro Focus Verastream 이미지를 위한 Amazon ECR에서 Docker 리포지토리를 생성합니다.	Amazon ECR 콘솔에서 Micro Focus Verastream 이미지를 위한 Docker 리포지토리를 생성합니다.	Amazon ECR
를 통해 시스템 통합 CI/CD 파이프라인을 배포하십시오. CloudFormation	준비된 AWS CloudFormation 템플릿을 사용하여 시스템 통합 CI/CD 파이프라인을 배포하십시오.	AWS CloudFormation

프로덕션 배포 CI/CD 파이프라인을 생성합니다.

작업	설명	필요한 기술
Quick Start를 사용하여 Micro Focus Enterprise Server를 배포합니다.	AWS를 사용하여 마이크로 포커스 엔터프라이즈 서버를 배포하려면 AWS CloudFormation Quick Start에서 마이크로 포커스 엔터프라이즈 서버를 시작하십시오.	AWS CloudFormation
프로덕션 배포 CI/CD 파이프라인을 배포합니다.	AWS CloudFormation 콘솔에서 AWS CloudFormation 템플릿을 사용하여 프로덕션 배포 CI/CD 파이프라인을 배포합니다.	AWS CloudFormation

관련 리소스

참조

- [AWS DevOps 블로그 — Micro Focus 엔터프라이즈 스위트를 사용하여 AWS에서 수천 개의 메인프레임 테스트를 자동화하십시오.](#)
- [py3270/py3270 리포지토리 GitHub](#)
- [Altran-PT-GDC/로봇 프레임워크-메인프레임-3270-라이브러리 GitHub 리포지토리](#)
- [Welcome to behave!](#)
- [APN 파트너 블로그 - 태그: Micro Focus](#)
- [시작 템플릿에서 인스턴스 시작](#)

AWS Marketplace

- [Micro Focus UFT One](#)

Quick Starts

- [Micro Focus Enterprise Server on AWS](#)

비 워크로드 서브넷을 위한 다중 계정 VPC 설계에서 라우팅 가능한 IP 공간 보존

작성자: Adam Spicer(AWS)

코드 리포지토리: 라우팅이 [불가능한](#) 보조 CIDR 패턴

환경: 프로덕션

기술: 인프라; 관리 및 거버넌스 DevOps; 네트워킹

AWS 서비스: AWS Transit Gateway, Amazon VPC, Elastic Load Balancing(ELB)

요약

Amazon Web Services(AWS)는 [Transit Gateway Attachment](#)와 [Gateway Load Balancer 엔드포인트\(AWS Network Firewall](#) 또는 타사 어플라이언스 지원) 모두에 대해 Virtual Private Cloud(VPC)에서 전용 서브넷 사용을 권장하는 모범 사례를 발표했습니다. 이러한 서브넷은 이러한 서비스를 위한 엘라스틱 네트워크 인터페이스를 포함하는 데 사용됩니다. AWS Transit Gateway와 Gateway Load Balancer를 모두 사용하는 경우 VPC의 각 가용 영역에 서브넷 2개가 생성됩니다. VPC의 설계 방식 때문에 이러한 추가 서브넷은 [/28 마스크보다 작을 수 없으며](#) 라우팅 가능한 워크로드에 사용할 수 있는 귀중한 라우팅 가능한 IP 공간을 차지할 수 있습니다. 이 패턴은 이러한 전용 서브넷에 대해 라우팅이 불가능한 보조 Classless Inter-Domain Routing(CIDR) 범위를 사용하여 라우팅 가능한 IP 공간을 보존하는 방법을 보여줍니다.

사전 조건 및 제한 사항

사전 조건

- 라우팅 가능한 IP 공간을 위한 [다중 VPC 전략](#)
- 사용 중인 서비스에 대한 라우팅이 불가능한 CIDR 범위([Transit Gateway Attachment](#) 및 [Gateway Load Balancer](#) 또는 [Network Firewall 엔드포인트](#))

아키텍처

대상 아키텍처.

이 패턴에는 두 개의 참조 아키텍처가 포함됩니다. 한 아키텍처에는 Transit Gateway(TGW) 연결 및 Gateway Load Balancer 엔드포인트(GWLBe)를 위한 서브넷이 있고, 다른 아키텍처에는 TGW 연결 전용 서브넷이 있습니다.

아키텍처 1 – 어플라이언스에 대한 수신 라우팅을 지원하는 TGW 연결 VPC

다음 다이어그램은 두 개의 가용 영역에 걸친 VPC의 참조 아키텍처를 나타냅니다. [인그레스 시 VPC는 인그레스 라우팅 패턴을 사용하여 퍼블릭 서브넷으로 향하는 트래픽을 방화벽 검사를 위해 어플라이언스로 전달합니다. bump-in-the-wire](#) TGW 연결은 프라이빗 서브넷에서 별도의 VPC로의 송신을 지원합니다.

이 패턴은 TGW 연결 서브넷과 GWLBe 서브넷에 대해 라우팅할 수 없는 CIDR 범위를 사용합니다. TGW 라우팅 테이블에서 이 라우팅 불가능한 CIDR은 보다 구체적인 일련의 경로를 사용하여 블랙홀(정적) 경로로 구성됩니다. 경로가 TGW 라우팅 테이블로 전파되는 경우 이러한 보다 구체적인 블랙홀 경로가 적용됩니다.

이 예시에서는 /23 라우팅 가능한 CIDR이 분할되어 라우팅 가능한 서브넷에 완전히 할당됩니다.

아키텍처 2 - TGW 연결 VPC

다음 다이어그램은 두 개의 가용 영역에 걸친 VPC의 또 다른 참조 아키텍처를 나타냅니다. TGW 연결은 프라이빗 서브넷에서 별도의 VPC로의 아웃바운드 트래픽(송신)을 지원합니다. TGW 연결 서브넷에만 라우팅할 수 없는 CIDR 범위를 사용합니다. TGW 라우팅 테이블에서 이 라우팅 불가능한 CIDR은 보다 구체적인 일련의 경로를 사용하여 블랙홀 경로로 구성됩니다. 경로가 TGW 라우팅 테이블로 전파되는 경우 이러한 보다 구체적인 블랙홀 경로가 적용됩니다.

이 예시에서는 /23 라우팅 가능한 CIDR이 분할되어 라우팅 가능한 서브넷에 완전히 할당됩니다.

도구

AWS 서비스 및 리소스

- [Amazon Virtual Private Cloud\(VPC\)](#)를 이용하면 사용자가 정의한 가상 네트워크로 AWS 리소스를 시작할 수 있습니다. 이 가상 네트워크는 사용자의 자체 데이터 센터에서 운영하는 기존 네트워크와 유사하며 AWS의 확장 가능한 인프라를 사용한다는 이점이 있습니다. 이 패턴에서 VPC 보조 CIDR은 워크로드 CIDR에서 라우팅 가능한 IP 공간을 보존하는 데 사용됩니다.

- [인터넷 게이트웨이 수신 라우팅](#)(엣지 연결)을 라우팅이 불가능한 전용 서브넷의 Gateway Load Balancer 엔드포인트와 함께 사용할 수 있습니다.
- [AWS Transit Gateway](#)는 VPC와 온프레미스 네트워크를 연결하는 중앙 허브입니다. 이 패턴에서 VPC는 전송 게이트웨이에 중앙에서 연결되고 Transit Gateway Attachment는 라우팅이 불가능한 전용 서브넷에 있습니다.
- [Gateway Load Balancer](#)를 사용하면 방화벽, 침입 탐지 및 방지 시스템, 심층 패킷 검사 시스템과 같은 가상 어플라이언스를 배포, 조정 및 관리할 수 있습니다. 게이트웨이는 모든 트래픽의 단일 출입 지점 역할을 합니다. 이 패턴에서 Gateway Load Balancer의 엔드포인트는 라우팅이 불가능한 전용 서브넷에서 사용할 수 있습니다.
- [AWS Network Firewall](#)은 AWS Cloud를 위한 상태 저장형, 관리형 네트워크 방화벽이자, 침입 탐지 및 방지 서비스입니다. 이 패턴에서 방화벽의 엔드포인트는 라우팅이 불가능한 전용 서브넷에서 사용할 수 있습니다.

코드 리포지토리

이 패턴에 대한 런북과 AWS CloudFormation 템플릿은 GitHub [라우팅이 불가능한 보조 CIDR](#) 패턴 리포지토리에서 제공됩니다. 샘플 파일을 사용하여 사용자 환경에 작업실을 설정할 수 있습니다.

모범 사례

AWS Transit Gateway

- 각 Transit Gateway VPC 첨부 파일에 대해 별도의 서브넷을 사용합니다.
- 라우팅이 불가능한 보조 CIDR 범위의 /28 서브넷을 Transit Gateway Attachment 서브넷에 할당합니다.
- 각 전송 게이트웨이 라우팅 테이블에 라우팅이 불가능한 CIDR 범위에 대한 보다 구체적인 정적 경로를 블랙홀로 추가합니다.

Gateway Load Balancer 및 수신 라우팅

- 수신 라우팅을 사용하여 인터넷에서 Gateway Load Balancer 엔드포인트로 트래픽을 전달합니다.
- 각 Gateway Load Balancer 엔드포인트에 대해 별도의 서브넷을 사용합니다.
- Gateway Load Balancer 엔드포인트 서브넷에 라우팅할 수 없는 보조 CIDR 범위에서 /28 서브넷을 할당합니다.

에픽

VPC 생성

작업	설명	필요한 기술
라우팅할 수 없는 CIDR 범위를 결정합니다.	Transit Gateway Attachment 서브넷 및 (선택 사항) 모든 Gateway Load Balancer 또는 Network Firewall 엔드포인트 서브넷에 사용할 라우팅 불가능한 CIDR 범위를 결정합니다. 이 CIDR 범위는 VPC의 보조 CIDR로 사용됩니다. VPC의 기본 CIDR 범위 또는 대규모 네트워크에서 라우팅할 수 없어야 합니다.	클라우드 아키텍트
VPC의 라우팅 가능한 CIDR 범위를 결정합니다.	VPC에 사용할 라우팅 가능한 CIDR 범위 세트를 결정합니다. 이 CIDR 범위는 VPC의 기본 CIDR로 사용됩니다.	클라우드 아키텍트
VPC를 생성합니다.	VPC를 생성하고 전송 게이트웨이에 연결합니다. 각 VPC에는 이전 두 단계에서 결정한 범위를 바탕으로 라우팅할 수 있는 기본 CIDR 범위와 라우팅할 수 없는 보조 CIDR 범위가 있어야 합니다.	클라우드 아키텍트

전송 게이트웨이 블랙홀 경로 구성

작업	설명	필요한 기술
라우팅이 불가능한 보다 구체적인 CIDR을 블랙홀로 생성합니다.	각 전송 게이트웨이 라우팅 테이블에는 라우팅할 수 없는 CIDR에 대해 생성된 일련의 블랙홀 경로가 있어야 합니다. 이는 보조 VPC CIDR의 모든 트래픽이 라우팅되지 않고 대규모 네트워크로 누출되지 않도록 구성되어 있습니다. 이러한 경로는 VPC에서 보조 CIDR로 설정된 라우팅 불가능한 CIDR보다 더 구체적이어야 합니다. 예를 들어, 라우팅이 불가능한 보조 CIDR이 100.64.0.0/26인 경우 전송 게이트웨이 라우팅 테이블의 블랙홀 경로는 100.64.0.0/27 및 100.64.0.32/27이어야 합니다.	클라우드 아키텍트

관련 리소스

- [Gateway Load Balancer 배포 모범 사례](#)
- [Gateway Load Balancer를 사용한 분산 검사 아키텍처](#)
- [Networking Immersion Day – 인터넷-VPC 방화벽 연구실](#)
- [전송 게이트웨이 설계 모범 사례](#)

추가 정보

라우팅이 불가능한 보조 CIDR 범위는 대규모 IP 주소 집합이 필요한 대규모 컨테이너 배포를 작업할 때에도 유용할 수 있습니다. 프라이빗 NAT 게이트웨이와 함께 이 패턴을 사용하여 라우팅 불가능한 서브넷을 사용하여 컨테이너 배포를 호스팅할 수 있습니다. 자세한 내용은 블로그 게시물 [How to solve Private IP exhaustion with Private NAT Solution](#)을 참조하세요.

코드 리포지토리를 사용하여 AWS Service Catalog에서 Terraform 제품을 프로비저닝합니다.

작성자: Dr. Rahul Sharad Gaikwad(AWS) 및 Tamilselvan P(AWS)

환경: PoC 또는 파일럿

기술: 인프라, DevOps

워크로드: 기타 모든 워크로드

AWS 서비스: AWS 서비스 카탈로그, 아마존 EC2

요약

AWS Service Catalog는 [HashiCorp Terraform](#) 구성에 대한 거버넌스를 통해 셀프 서비스 프로비저닝을 지원합니다. Terraform을 사용하는 경우 Service Catalog를 단일 도구로 사용하여 AWS 내에서 Terraform 구성을 대규모로 구성, 관리 및 배포할 수 있습니다. 표준화되고 사전 승인된 IaC (Infrastructure as Code) 템플릿의 카탈로그 작성, 액세스 제어, 최소 권한 액세스로 클라우드 리소스 프로비저닝, 버전 관리, 수천 개의 AWS 계정과의 공유, 태깅 등 Service Catalog의 주요 기능에 액세스할 수 있습니다. 엔지니어, 데이터베이스 관리자, 데이터 과학자와 같은 최종 사용자는 액세스 권한이 있는 제품 및 버전 목록을 확인하고 한 번의 작업으로 이를 배포할 수 있습니다.

이 패턴은 Terraform 코드를 사용하여 AWS 리소스를 배포하는 데 도움이 됩니다. GitHub 리포지토리의 Terraform 코드는 Service Catalog를 통해 액세스할 수 있습니다. 이 접근 방식을 사용하면 제품을 기존 Terraform 워크플로와 통합할 수 있습니다. 관리자는 Terraform을 사용하여 Service Catalog 포트폴리오를 생성하고 AWS Launch Wizard 제품을 포트폴리오에 추가할 수 있습니다.

이 솔루션의 이점은 다음과 같습니다.

- Service Catalog의 롤백 기능으로 인해 배포 중에 문제가 발생하는 경우 제품을 이전 버전으로 되돌릴 수 있습니다.
- 제품 버전 간의 차이점을 쉽게 식별할 수 있습니다. 이렇게 하면 배포 중에 문제를 해결하는 데 도움이 됩니다.
- 서비스 카탈로그에서 GitHub GitLab, 또는 CodeCommit AWS와 같은 리포지토리 연결을 구성할 수 있습니다. 리포지토리를 통해 직접 제품을 변경할 수 있습니다.

AWS Service Catalog의 전반적인 이점에 대한 자세한 내용은 서비스 [카탈로그란 무엇입니까?](#) 를 참조하십시오.

사전 조건 및 제한 사항

사전 조건

- 활성 상태의 AWS 계정
- A GitHub BitBucket, 또는 ZIP 형식의 Terraform 구성 파일이 들어 있는 기타 리포지토리.
- [AWS 서버리스 애플리케이션 모델 명령줄 인터페이스 \(AWS SAM CLI\)](#), [설치](#)
- AWS Command Line Interface(AWS CLI), [설치](#) 및 [구성됨](#)
- [Go](#), [설치됨](#).
- Python 버전 3.9가 [설치되었습니다](#). AWS SAM CLI에는 이 버전의 Python이 필요합니다.
- AWS Lambda 함수를 작성하고 실행할 수 있는 권한과 Service Catalog 제품 및 포트폴리오를 액세스하고 관리할 수 있는 권한.

아키텍처

대상 기술 스택

- AWS Service Catalog
- AWS Lambda

대상 아키텍처

이 다이어그램은 다음 워크플로를 보여줍니다.

1. Terraform 구성이 준비되면 개발자는 모든 테라폼 코드가 포함된.zip 파일을 만듭니다. 개발자는.zip 파일을 Service Catalog에 연결된 코드 저장소에 업로드합니다.
2. 관리자가 Terraform 제품을 Service Catalog의 포트폴리오에 연결합니다. 또한 관리자는 최종 사용자가 제품을 프로비저닝할 수 있도록 하는 시작 제약 조건을 생성합니다.
3. Service Catalog에서 최종 사용자는 Terraform 구성을 사용하여 AWS 리소스를 시작합니다. 배포할 제품 버전을 선택할 수 있습니다.

도구

AWS 서비스 및 도구

- [AWS Lambda](#)는 서버를 프로비저닝하거나 관리할 필요 없이 코드를 실행하는 데 도움이 되는 컴퓨팅 서비스입니다. 필요할 때만 코드를 실행하며 자동으로 확장이 가능하므로 사용한 컴퓨팅 시간만큼만 비용을 지불합니다.
- [AWS Service Catalog](#)를 사용하면 AWS에 승인된 IT 서비스의 카탈로그를 중앙에서 관리할 수 있습니다. 최종 사용자는 조직에서 규정한 제약에 따라, 필요에 따라 승인된 IT 서비스만 신속하게 배포할 수 있습니다.

기타 서비스

- [Go](#)는 Google에서 지원하는 오픈소스 프로그래밍 언어입니다.
- [Python](#)은 범용 컴퓨터 프로그래밍 언어입니다.

코드 리포지토리

Service Catalog를 통해 배포할 수 있는 샘플 Terraform 구성이 필요한 경우 Terraform을 사용한 [Amazon GitHub Macie 조직 설정](#) 리포지토리의 구성을 사용할 수 있습니다. 이 리포지토리의 코드 샘플을 사용할 필요는 없습니다.

모범 사례

- Terraform 구성 파일 (terraform.tfvars) 에서 변수 값을 제공하는 대신 Service Catalog를 통해 제품을 시작할 때 변수 값을 구성하십시오.
- 특정 사용자 또는 관리자에게만 포트폴리오에 대한 액세스 권한을 부여합니다.
- 최소 권한 원칙에 따라 작업 수행에 필요한 최소 권한을 부여하십시오. 자세한 내용은 IAM 설명서의 [최소 권한 부여 및 보안 모범 사례](#)를 참조하세요.

에픽

로컬 워크스테이션 설정

작업	설명	필요한 기술
(선택 사항) Docker를 설치합니다.	개발 환경에서 AWS Lambda 함수를 실행하려면 Docker를 설치하십시오. 자세한 내용은	DevOps 엔지니어

작업	설명	필요한 기술
	<p>도커 설명서의 도커 엔진 설치 단원을 참조하세요.</p>	
<p>테라폼용 AWS Service Catalog 엔진을 설치합니다.</p>	<ol style="list-style-type: none"> 다음 명령을 입력하여 Terraform용 AWS Service Catalog 엔진 리포지토리를 복제합니다. <pre data-bbox="630 554 1029 793">git clone https://github.com/aws-samples/service-catalog-engine-for-terraform-os.git</pre> 복제된 리포지토리의 루트 디렉터리로 이동합니다. 다음 명령을 입력합니다. 그러면 엔진이 설치됩니다. <pre data-bbox="630 1031 1029 1150">run ./bin/bash/deploy-tre.sh -r</pre> <p>기본 프로필에 설정된 AWS 지역은 자동 설치 중에 사용되지 않습니다. 대신 이 명령을 실행할 때 지역을 제공합니다.</p>	<p>DevOps 엔지니어, AWS 관리자</p>

GitHub 리포지토리 연결

작업	설명	필요한 기술
<p>GitHub 리포지토리에 대한 연결을 생성합니다.</p>	<ol style="list-style-type: none"> AWS 관리 콘솔에 로그인한 다음 개발자 도구 콘솔을 엽니다. AWS CodePipeline, AWS 또는 AWS와 같 	<p>AWS 관리자</p>

작업	설명	필요한 기술
	<p>은 서비스를 선택하여 개발자 도구 콘솔에 액세스할 수 CodeDeploy 있습니다. CodeCommit</p> <ol style="list-style-type: none"> 2. 왼쪽 탐색 창에서 [설정] 을 선택한 다음 [연결] 을 선택합니다. 3. 연결 생성을 선택합니다. 4. Terraform 소스 코드를 유지 관리하는 리포지토리를 선택합니다. 예를 들어, Bitbucket 또는 GitHub 엔터프라이즈 서버를 선택할 수 있습니다. GitHub 5. 연결 이름을 입력한 다음 Connect를 선택합니다. 6. 메시지가 표시되면 리포지토리를 인증합니다. <p>인증이 완료되면 연결이 생성되고 상태가 활성으로 변경됩니다.</p>	

Service Catalog에서 테라폼 제품 생성

작업	설명	필요한 기술
<p>Service Catalog 제품을 생성합니다.</p>	<ol style="list-style-type: none"> 1. AWS Service Catalog 콘솔 을 엽니다. 2. 관리 섹션으로 이동한 다음 제품 목록을 선택합니다. 3. 제품 생성을 선택합니다. 	<p>AWS 관리자</p>

작업	설명	필요한 기술
	<ol style="list-style-type: none"> 4. 제품 세부 정보 섹션의 제품 생성 페이지에서 외부 제품 유형을 선택합니다. Service Catalog는 이 제품 유형을 사용하여 Terraform 커뮤니티 에디션 제품을 지원합니다. 5. Service Catalog 제품의 이름과 소유자를 입력합니다. 6. CodeStar 공급자를 사용하여 코드 리포지토리 지정을 선택합니다. 7. 리포지토리에 대해 다음 정보를 입력합니다. <ul style="list-style-type: none"> • 다음을 사용하여 공급자에 연결 AWS CodeConnections — 이전에 생성한 연결을 선택합니다. • 리포지토리 - 리포지토리를 선택합니다. • 브랜치 — 브랜치를 선택합니다. • 템플릿 파일 경로 - 코드 템플릿 파일이 저장된 경로를 선택합니다. 파일 이름은 로 끝나야 tar.gz 합니다. 8. 버전 이름 및 설명에 제품 버전에 대한 정보를 입력합니다. 9. 제품 생성을 선택합니다. 	

작업	설명	필요한 기술
포트폴리오를 생성합니다.	<ol style="list-style-type: none"> 1. AWS Service Catalog 콘솔을 엽니다. 2. 관리 섹션으로 이동한 다음 포트폴리오를 선택하고 선택합니다. 3. 포트폴리오 생성을 선택합니다. 4. 다음 값을 입력합니다. <ul style="list-style-type: none"> • 포트폴리오 이름 - Sample terraform • 포트폴리오 설명 - Sample portfolio for Terraform configurations • 소유자 — 연락처 정보 (예: 이메일) 5. 생성을 선택하세요. 	AWS 관리자
Terraform 제품을 포트폴리오에 추가합니다.	<ol style="list-style-type: none"> 1. AWS Service Catalog 콘솔을 엽니다. 2. 관리 섹션으로 이동한 다음 제품 목록을 선택합니다. 3. 이전에 만든 Terraform 제품을 선택합니다. 4. 작업을 선택한 다음 포트폴리오에 제품 추가를 선택합니다. 5. Sample terraform 포트폴리오를 선택합니다. 6. 포트폴리오에 제품 추가를 선택합니다. 	AWS 관리자

작업	설명	필요한 기술
액세스 정책을 생성합니다.	<ol style="list-style-type: none"> 1. AWS 자격 증명 및 액세스 관리 (IAM) 콘솔을 엽니다. 2. 탐색 창에서 정책을 선택합니다. 3. 콘텐츠 창에서 정책 생성을 선택합니다. 4. JSON 옵션을 선택합니다. 5. 이 패턴의 추가 정보 섹션에 있는 액세스 정책에 샘플 JSON 정책을 입력합니다. 6. 다음을 선택합니다. 7. 검토 및 생성 페이지의 정책 이름 상자에 TerraformResourceCreationAndArtifactAccessPolicy 8. 정책 생성(Create policy)을 선택합니다. 	AWS 관리자

작업	설명	필요한 기술
<p>사용자 지정 신뢰 정책을 생성합니다.</p>	<ol style="list-style-type: none"> 1. AWS 자격 증명 및 액세스 관리 (IAM) 콘솔을 엽니다. 2. 탐색 창에서 역할을 선택합니다. 3. 역할 생성을 선택합니다. 4. 신뢰할 수 있는 개체 유형에서 사용자 지정 신뢰 정책을 선택합니다. 5. JSON 정책 편집기에서 이 패턴의 추가 정보 섹션에 있는 신뢰 정책에 샘플 JSON 정책을 입력합니다. 6. 다음을 선택합니다. 7. 권한 정책에서 Terraform ResourceCreationAndArtifactAccessPolicy 이전에 만든 정책을 선택합니다. 8. 다음을 선택합니다. 9. 역할 세부 정보의 역할 이름 상자에 <code>SLaunch-product</code>를 입력합니다. <p>중요: 역할 이름은 로 시작해야 합니다SLaunch.</p> <ol style="list-style-type: none"> 10. 역할 생성을 선택합니다. 	<p>AWS 관리자</p>

작업	설명	필요한 기술
<p>Service Catalog 제품에 시작 제약 조건을 추가합니다.</p>	<ol style="list-style-type: none"> 1. 관리 권한이 있는 사용자로 AWS Management 콘솔에 로그인합니다. 2. AWS Service Catalog 콘솔을 엽니다. 3. 탐색 창에서 포트폴리오를 선택합니다. 4. 이전에 만든 포트폴리오를 선택합니다. 5. 포트폴리오 세부 정보 페이지에서 제약 조건 탭을 선택한 다음 제약 조건 생성을 선택합니다. 6. 제품의 경우 이전에 만든 Terraform 제품을 선택합니다. 7. 시작 제약에서 방법에 대해 역할 이름 입력을 선택합니다. 8. 역할 이름 상자에 를 입력합니다. SCLaunch-product . 9. 생성을 선택하세요. 	<p>AWS 관리자</p>

작업	설명	필요한 기술
<p>제품에 대한 액세스 권한을 부여합니다.</p>	<ol style="list-style-type: none"> 1. AWS Service Catalog 콘솔 을 엽니다. 2. 탐색 창에서 포트폴리오를 선택합니다. 3. 이전에 만든 포트폴리오를 선택합니다. 4. [액세스] 탭을 선택한 다음 [액세스 권한 부여] 를 선택합니다. 5. 역할 탭을 선택한 다음 이 제품을 배포하는 데 필요한 액세스 권한을 가져야 하는 역할을 선택합니다. 6. 액세스 권한 부여를 선택합니다. 	<p>AWS 관리자</p>
<p>제품을 실행합니다.</p>	<ol style="list-style-type: none"> 1. Service Catalog 제품을 배포할 권한이 있는 사용자로 AWS Management 콘솔에 로그인합니다. 2. AWS Service Catalog 콘솔 을 엽니다. 3. 탐색 창에서 제품을 선택합니다. 4. 이전에 만든 제품을 선택한 다음 제품 출시를 선택합니다. 5. 제품 이름을 입력하고 필수 매개변수를 정의합니다. 6. 제품 시작을 선택합니다. 	<p>DevOps 엔지니어</p>

배포 확인

작업	설명	필요한 기술
<p>배포를 검증합니다.</p>	<p>Service Catalog 프로비저닝 워크플로를 위한 AWS Step Functions 상태 머신은 두 개 있습니다.</p> <ul style="list-style-type: none"> • <code>ManageProvisionedProductStateMachine</code> —서비스 카탈로그는 새 Terraform 제품을 프로비저닝하고 기존 Terraform 프로비저닝 제품을 업데이트할 때 이 상태 시스템을 호출합니다. • <code>TerminateProvisionedProductStateMachine</code> —서비스 카탈로그는 기존 Terraform 프로비저닝 제품을 종료할 때 이 상태 시스템을 호출합니다. <p><code>ManageProvisionedProductStateMachine</code> 상태 머신의 로그를 확인하여 제품이 프로비저닝되었는지 확인합니다.</p> <ol style="list-style-type: none"> 1. AWS 관리 콘솔에 로그인한 다음 AWS Step Functions 콘솔을 엽니다. 2. 왼쪽 탐색 창에서 스테이트 머신을 선택합니다. 	<p>DevOps 엔지니어</p>

작업	설명	필요한 기술
	<ol style="list-style-type: none"> 3. 선택하세요 <code>ManagedProvisionedProductStateMachine</code>. 4. 실행 목록에서 프로비저닝된 제품 ID를 입력하여 실행을 찾습니다. <p>참고: 상태 파일 백엔드 버킷 이름은 로 시작합니다. <code>sc-terraform-engine-state-</code></p> <ol style="list-style-type: none"> 5. 계정에 필요한 리소스가 모두 생성되었는지 확인하세요. 	

인프라 정리

작업	설명	필요한 기술
프로비저닝된 제품을 삭제합니다.	<ol style="list-style-type: none"> 1. Service Catalog 제품을 배포할 권한이 있는 사용자로 AWS Management 콘솔에 로그인합니다. 2. AWS Service Catalog 콘솔을 엽니다. 3. 왼쪽 탐색에서 프로비저닝된 제품을 선택합니다. 4. 생성한 제품을 선택합니다. 5. 작업 목록에서 종료를 선택합니다. 6. 확인 텍스트 상자에 를 입력한 <code>terminate</code> 다음 프로 	DevOps 엔지니어

작업	설명	필요한 기술
	<p>비저닝된 제품 종료를 선택합니다.</p> <p>7. 이 단계를 반복하여 프로비전된 제품을 모두 종료합니다.</p>	

작업	설명	필요한 기술
테라폼용 AWS Service Catalog 엔진을 제거합니다.	<ol style="list-style-type: none"> 1. 관리 권한이 있는 사용자로 AWS Management 콘솔에 로그인합니다. 2. Amazon S3 콘솔을 엽니다. 3. 탐색 창에서 버킷을 선택합니다. 4. sc-terraform-engine-logging-XXXX 버킷을 선택합니다. 5. 비우기를 선택합니다. 6. 다음 버킷에 대해 4~5단계를 반복합니다. <ul style="list-style-type: none"> • sc-terraform-engine-state-XXXX • terraform-engine-bootstrap-XXXX 7. AWS CloudFormation 콘솔을 열고 올바른 AWS 지역에 있는지 확인합니다. 8. 왼쪽 탐색 창에서 [Stacks]를 선택합니다. 9. 선택한 SAM-TRE 다음 삭제를 선택합니다. 스택이 삭제될 때까지 기다리십시오. 10. 선택한 Bootstrap-TRE 다음 삭제를 선택합니다. 스택이 삭제될 때까지 기다리십시오. 	AWS 관리자

관련 리소스

AWS 설명서

- [Terraform 제품 시작하기](#)

Terraform 설명서

- [Terraform 설치](#)
- [테라폼 백엔드 구성](#)
- [테라폼 AWS 공급자 설명서](#)

추가 정보

액세스 정책

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "VisualEditor0",
      "Effect": "Allow",
      "Action": "s3:GetObject",
      "Resource": "*",
      "Condition": {
        "StringEquals": {
          "s3:ExistingObjectTag/servicecatalog:provisioning": "true"
        }
      }
    },
    {
      "Action": [
        "s3:CreateBucket*",
        "s3>DeleteBucket*",
        "s3:Get*",
        "s3:List*",
        "s3:PutBucketTagging"
      ],
      "Resource": "arn:aws:s3:::*",
      "Effect": "Allow"
    },
    {
      "Action": [
        "resource-groups:CreateGroup",
        "resource-groups:ListGroupResources",

```

```

        "resource-groups:DeleteGroup",
        "resource-groups:Tag"
    ],
    "Resource": "*",
    "Effect": "Allow"
},
{
    "Action": [
        "tag:GetResources",
        "tag:GetTagKeys",
        "tag:GetTagValues",
        "tag:TagResources",
        "tag:UntagResources"
    ],
    "Resource": "*",
    "Effect": "Allow"
}
]
}

```

신뢰 정책

```

{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "GivePermissionsToServiceCatalog",
            "Effect": "Allow",
            "Principal": {
                "Service": "servicecatalog.amazonaws.com"
            },
            "Action": "sts:AssumeRole"
        },
        {
            "Effect": "Allow",
            "Principal": {
                "AWS": "arn:aws:iam::account_id:root"
            },
            "Action": "sts:AssumeRole",
            "Condition": {
                "StringLike": {
                    "aws:PrincipalArn": [

```

```
        "arn:aws:iam::accounti_id:role/TerraformEngine/  
TerraformExecutionRole*",  
        "arn:aws:iam::accounti_id:role/TerraformEngine/  
ServiceCatalogExternalParameterParserRole*",  
        "arn:aws:iam::accounti_id:role/TerraformEngine/  
ServiceCatalogTerraformOSParameterParserRole*" ]  
    }  
}  
]  
}
```

Amazon SES를 사용하여 단일 이메일 주소로 여러 AWS 계정 등록

작성자: 조 워즈니악(AWS) 및 슈방기 비슈와카르마(AWS)

코드 리포지토리: [GitHub aws-account-factory-email](#)

환경: PoC 또는 파일럿

기술: 인프라, 관리 및 거버넌스, 메시징 및 통신

AWS 서비스: AWS Lambda, Amazon SES, Amazon DynamoDB

요약

이 패턴은 실제 이메일 주소를 AWS 계정과 연결된 이메일 주소에서 분리하는 방법을 설명합니다. AWS 계정은 계정 생성 시 고유한 이메일 주소를 제공해야 합니다. 일부 조직에서는 AWS 계정을 관리하는 팀이 메시징 팀과 함께 많은 고유한 이메일 주소를 관리해야 합니다. 이는 여러 AWS 계정을 관리하는 대규모 조직에서는 어려울 수 있습니다.

이 패턴은 AWS 계정 소유자가 하나의 이메일 주소를 여러 AWS 계정과 연결할 수 있는 고유한 이메일 주소 벤딩 솔루션을 제공합니다. 그러면 AWS 계정 소유자의 실제 이메일 주소가 테이블의 생성된 이메일 주소와 연결됩니다. 이 솔루션은 고유한 이메일 계정에 대해 수신되는 모든 이메일을 처리하고, 각 계정의 소유자를 조회한 다음, 수신된 모든 메시지를 소유자에게 전달합니다.

사전 조건 및 제한 사항

사전 조건

- AWS 계정에 대한 관리 액세스가 필요합니다.
- 개발 환경에 대한 액세스가 필요합니다. 필요한 도구와 액세스 키를 직접 설정하지 않아도 되도록 AWS Cloud9를 사용하는 것이 좋습니다.
- (선택 사항) AWS Cloud Development Kit(AWS CDK) 워크플로우와 Python 프로그래밍 언어에 익숙하면 문제를 해결하거나 수정하는 데 도움이 됩니다.

제한 사항

- 벤딩되는 전체 이메일 주소 길이는 64자입니다. 자세한 내용은 AWS [CreateAccountOrganizations](#) API 레퍼런스를 참조하십시오.

제품 버전

- Node.js 버전 12.7.0 이상
- Python 3.9 이상
- Python 패키지 pip 및 virtualenv
- AWS CDK 버전 2.23.0 이상
- Docker 20.10.x 이상

아키텍처

대상 기술 스택

- AWS CloudFormation 스택
- AWS Lambda 함수
- Amazon Simple Email Service(Amazon SES) 규칙 및 규칙 세트
- AWS Identity and Access Management(IAM) 역할 및 정책
- Amazon Simple Storage Service(S3) 버킷 및 버킷 정책
- AWS Key Management Service(AWS KMS) 키 및 키 정책
- Amazon Simple Notification Service(SNS) 주제 및 주제 정책
- Amazon DynamoDB 테이블

대상 아키텍처

이 다이어그램은 두 가지 흐름을 보여줍니다.

- 이메일 주소 벤딩 흐름: 다이어그램에서 이메일 주소 벤딩 흐름(하단 섹션)은 일반적으로 계정 벤딩 솔루션 또는 외부 자동화로 시작되거나 수동으로 호출됩니다. 요청 시 필요한 메타데이터가 포함된 페이로드와 함께 Lambda 함수가 호출됩니다. 함수는 이 정보를 사용하여 고유한 계정 이름과 이메일 주소를 생성하고 DynamoDB 데이터베이스에 저장한 다음 호출자에게 값을 반환합니다. 그런 다음 이 값을 사용하여 새 AWS 계정을 만들 수 있습니다(일반적으로 AWS Organizations 사용).
- 이메일 전달 흐름: 이 흐름은 이전 다이어그램의 위쪽 섹션에 설명되어 있습니다. 이메일 주소 벤딩 흐름에서 생성된 계정 이메일을 사용하여 AWS 계정을 생성하면 AWS는 계정 등록 확인, 정기 알림 등 다양한 이메일을 해당 이메일 주소로 보냅니다. 이 패턴의 단계에 따라 전체 도메인에 대한 이메일

일을 수신하도록 Amazon SES로 AWS 계정을 구성합니다. 이 솔루션은 Lambda가 모든 수신 이메일을 처리하고, TO 주소가 DynamoDB 테이블에 있는지 확인한 다음, 대신 메시지를 계정 소유자의 이메일 주소로 전달하도록 하는 전달 규칙을 구성합니다. 이 프로세스를 사용하면 계정 소유자가 여러 계정을 하나의 이메일 주소와 연결할 수 있습니다.

자동화 및 규모 조정

이 패턴은 AWS CDK를 사용하여 배포를 완전히 자동화합니다. 이 솔루션은 사용자 요구를 충족하기 위해 자동으로 확장하거나 그러도록 구성될 수 있는 AWS 관리형 서비스를 사용합니다. Lambda 함수는 확장 요구 사항을 충족하기 위해 추가 구성이 필요할 수 있습니다. 자세한 내용은 Lambda 설명서의 [Lambda 함수 크기 조정](#)을 참조하세요.

도구

서비스

- [AWS Cloud9](#)는 소프트웨어를 코딩, 구축, 실행, 테스트 및 디버깅하는 데 도움이 되는 통합 개발 환경(IDE)입니다. 또한, 소프트웨어를 AWS 클라우드로 릴리스하는 데도 도움이 됩니다.
- [AWS는 AWS CloudFormation](#) 리소스를 설정하고, 빠르고 일관되게 프로비저닝하고, AWS 계정 및 지역 전반의 수명 주기 전반에 걸쳐 리소스를 관리할 수 있도록 지원합니다.
- [AWS Command Line Interface\(AWS CLI\)](#)는 명령줄 셸에서 명령을 사용하여 AWS 서비스와 상호 작용할 수 있는 오픈 소스 도구입니다.
- [Amazon DynamoDB](#)는 빠르고 예측 가능하고 확장 가능한 성능을 제공하는 완전 관리형 NoSQL 데이터베이스 서비스입니다.
- [AWS Identity and Access Management\(IAM\)](#)는 사용자에게 대한 인증 및 권한 부여를 제어함으로써 AWS 리소스에 대한 액세스를 안전하게 제어할 수 있습니다.
- [AWS Key Management Service\(AWS KMS\)](#)를 사용하면 암호화 키를 생성하고 제어하여 데이터를 보호할 수 있습니다.
- [Lambda](#)는 서버를 프로비저닝하거나 관리할 필요 없이 코드를 실행할 수 있도록 도와주는 컴퓨팅 서비스입니다. 필요할 때만 코드를 실행하며 자동으로 확장이 가능하므로 사용한 컴퓨팅 시간만큼만 비용을 지불합니다.
- [Amazon Simple Email Service\(Amazon SES\)](#)를 사용하면 자신의 이메일 주소와 도메인을 사용하여 이메일을 보내고 받을 수 있습니다.
- [Amazon Simple Notification Service\(Amazon SNS\)](#)를 사용하면 웹 서버 및 이메일 주소를 포함하여 게시자와 클라이언트 간의 메시지 교환을 조정하고 관리할 수 있습니다.

- [Amazon Simple Storage Service\(S3\)](#)는 원하는 양의 데이터를 저장, 보호 및 검색하는 데 도움이 되는 클라우드 기반 객체 스토리지 서비스입니다.

배포에 필요한 도구

- AWS 계정에 대한 AWS CLI 및 IAM 액세스 권한이 있는 개발 환경입니다. 자세한 내용은 [관련 리소스](#) 섹션의 링크를 참조하세요. 설정 프로세스를 간소화하려면 AWS Cloud9를 사용하는 것이 좋습니다.
- AWS Cloud9를 사용하는 경우 다음이 자동으로 구성됩니다. AWS Cloud9를 사용하지 않기로 선택한 경우 다음을 설치해야 합니다.
 - AWS CDK에 대한 액세스 보안 인증을 구성하기 위한 AWS CLI입니다. 자세한 내용은 [AWS CLI 설명서](#)를 참조하세요.
 - Python 버전 3.9 이상
 - Python 패키지 pip 및 virtualenv
 - Node.js 버전 12.7.0 이상
 - AWS CDK 버전 2.23.0 이상
 - Docker 버전 20.10.x 이상.

code

이 패턴의 코드는 GitHub [AWS 어카운트 팩토리 이메일](#) 리포지토리에서 제공됩니다.

에픽

대상 배포 환경 할당

작업	설명	필요한 기술
AWS 계정을 확인 또는 생성합니다.	이메일 솔루션을 배포하려면 전체 관리 액세스 권한이 있는 기존 또는 새 AWS 계정을 확인합니다.	AWS 관리자, 클라우드 관리자
배포 환경을 설정합니다.	다음 단계에 따라 사용하기 쉬운 배포 환경을 구성하고 종속성을 설정합니다.	AWS DevOps, 앱 개발자

작업	설명	필요한 기술
	<p>1. AWS Cloud9 인스턴스를 전용 배포 환경으로 배포합니다. 지침은 AWS Cloud9 시작하기를 참조하세요.</p> <p>2. 다음 명령을 GitHub 사용하여 AWS 계정 팩토리 이메일 리포지토리 코드 베이스를 AWS Cloud9 인스턴스에 복제합니다.</p> <div style="border: 1px solid #ccc; border-radius: 10px; padding: 10px; margin: 10px 0;"> <pre>git clone https://github.com/aws-samples/aws-account-factory-email</pre> </div> <p>3. requirements.txt 파일 (리포지토리의 루트 내)에서 aws-cdk-lib== (으)로 시작하는 줄을 사용자 환경에서 실행 중인 AWS CDK 버전과 일치하도록 업데이트합니다. 버전을 확인하려면 <code>cdk --version</code> 명령을 사용합니다.</p>	

확인된 도메인 설정

작업	설명	필요한 기술
<p>도메인을 식별 및 할당합니다.</p>	<p>이메일 전달 기능을 사용하려면 전용 도메인이 필요합니다. Amazon SES로 확인할 수 있는 도메인 또는 하위 도메인을 식별하고 할당합니다. 이 도메인은 이메일 전달 솔루션이 배</p>	<p>클라우드 관리자, 네트워크 관리자, DNS 관리자</p>

작업	설명	필요한 기술
	<p>포된 AWS 계정 내에서 수신 이메일을 수신할 수 있어야 합니다.</p> <p>도메인 요구 사항:</p> <ul style="list-style-type: none"> • 도메인은 표준 도메인 또는 하위 도메인이어야 합니다. • 도메인은 조직 외부에서 이메일을 수신하는 데 사용되므로 외부에서 DNS로 확인할 수 있어야 합니다. 	
도메인을 확인합니다.	<p>식별된 도메인을 사용하여 수신 이메일을 수락할 수 있는지 확인합니다.</p> <p>Amazon SES 설명서의 Amazon SES 이메일 수신을 위한 도메인 확인 지침을 완료합니다. 이를 위해서는 도메인의 DNS 레코드를 담당하는 사람 또는 팀과의 협력이 필요합니다.</p>	앱 개발자, AWS DevOps
MX 레코드를 설정합니다.	<p>AWS 계정 및 리전의 Amazon SES 엔드포인트를 가리키는 MX 레코드로 도메인을 설정합니다. 자세한 내용은 Amazon SES 설명서에서 Amazon SES 이메일 수신을 위한 MX 레코드 게시하기를 참조하세요.</p>	클라우드 관리자, 네트워크 관리자, DNS 관리자

이메일 벤딩 및 전달 솔루션 배포

작업	설명	필요한 기술
<p>cdk.json의 기본값을 수정합니다.</p>	<p>솔루션이 배포된 후 제대로 작동하도록 cdk.json 파일(리포지토리의 루트 내)에서 일부 기본값을 편집합니다.</p> <ol style="list-style-type: none"> 이전에 확인한 도메인 이름과 일치하도록 SES_DOMAIN_NAME 값을 수정합니다. SES_DOMAIN_NAME 에 있는 도메인과 동일한 도메인을 포함하도록 ADDRESS_FROM 값을 수정합니다. 주소의 로컬 부분은 클라우드 팀에서 결정해야 합니다. 이 주소는 솔루션을 통해 전달되는 모든 이메일에 대한 FROM 주소가 됩니다. 일치하지 않는 모든 수신 메시지가 전달되는 이메일 주소와 일치하도록 ADDRESS_ADMIN 값을 수정합니다. 이 값은 유효한 운영 중인 이메일 주소여야 합니다. 	<p>앱 개발자, AWS DevOps</p>
<p>이메일 벤딩 및 전달 솔루션을 배포합니다.</p>	<ol style="list-style-type: none"> Python 가상 환경 생성: <pre>python -m venv .venv</pre> Python 가상 환경 활성화: <pre>source .venv/bin/activate</pre> 	<p>앱 개발자, AWS DevOps</p>

작업	설명	필요한 기술
	<p>또는 Windows 플랫폼에서는 다음을 사용:</p> <pre data-bbox="630 331 1027 447">% .venv\Scripts\activate.bat</pre> <p>3. 모든 Python 요구 사항을 오류 없이 설치:</p> <pre data-bbox="630 583 1027 699">pip install -r requirements.txt</pre> <p>4. CloudFormation 템플릿 합성:</p> <pre data-bbox="630 835 1027 919">cdk synth</pre> <p>오류가 없고 전체 CloudFormation 템플릿에 예상 출력이 포함되어 있는지 확인합니다.</p> <p>5. (선택 사항) AWS CDK 코드를 현재 AWS 계정 또는 리전에 처음으로 배포하는 경우 환경을 부트스트랩합니다. 자세한 내용은 AWS CDK 설명서의 부트스트래핑을 참조하세요.</p> <pre data-bbox="630 1518 1027 1675">cdk bootstrap aws://AWS-ACCOUNT-NUMBER/REGION</pre> <p>AWS-ACCOUNT-NUMBER 및 REGION을(를) 실제 값으로 바꿉니다.</p>	

작업	설명	필요한 기술
	<p>6. 솔루션 배포:</p> <div style="border: 1px solid #ccc; border-radius: 10px; padding: 10px; margin: 10px 0;"> <pre>cdk bootstrap cdk deploy</pre> </div> <p>명령은 오류 없이 완료되어야 합니다.</p>	
<p>솔루션이 배포되었는지 확인합니다.</p>	<p>테스트를 시작하기 전에 솔루션이 성공적으로 배포되었는지 확인:</p> <ol style="list-style-type: none"> 1. AWS CloudFormation 콘솔을 열고 이름이 들어 있는 CloudFormation 스택을 찾습니다 <code>AwsMailFwdStack</code> . 2. 이 <code>AwsMailFwdStack</code> 스택에 다음과 같은 리소스가 있는지 확인합니다. <ul style="list-style-type: none"> • Lambda 함수 • Amazon SES 규칙 및 규칙 세트 • IAM 역할 및 정책 • Amazon S3 버킷 및 버킷 정책 • AWS KMS 키 및 키 정책 • Amazon SNS 주제 및 주제 정책 • DynamoDB 테이블 	<p>앱 개발자, AWS DevOps</p>

이메일 벤딩 및 전달이 예상대로 작동하는지 확인

작업	설명	필요한 기술
<p>API가 작동 중인지 확인합니다.</p>	<p>이 단계에서는 솔루션의 API에 테스트 데이터를 제출하고 솔루션이 예상 출력을 생성하는지, 백엔드 작업이 예상대로 수행되었는지 확인합니다.</p> <p>테스트 입력을 사용하여 Vend Email Lambda 함수를 수동으로 실행합니다. (예를 보려면 sample_vend_request.json 파일을 참조하세요.) <code>OwnerAddress</code> 의 경우 유효한 이메일 주소를 사용합니다. API는 예상대로 값이 포함된 계정 이름과 계정 이메일을 반환해야 합니다.</p>	<p>앱 개발자, AWS DevOps</p>
<p>이메일이 전달되고 있는지 확인합니다.</p>	<p>이 단계에서는 시스템을 통해 테스트 이메일을 보내고 이메일이 예상 수신자에게 전달되는지 확인합니다.</p> <ol style="list-style-type: none"> 마지막 단계에서 계정 이메일을 가져옵니다. 테스트 제목과 본문 텍스트를 포함하여 이 주소로 이메일을 보냅니다. 계정 소유자의 이메일 주소로 이메일을 수신했는지 확인합니다. 수신한 이메일에 <code>cdk.json</code>의 <code>ADDRESS_FROM</code> 설정과 일치하는 	<p>앱 개발자, AWS DevOps</p>

작업	설명	필요한 기술
	<p>FROM 주소가 있는지 확인합니다.</p> <p>5. 수신한 이메일의 제목과 본문이 원래 보낸 메시지와 동일한지 확인합니다.</p>	

문제 해결

문제	Solution
<p>시스템이 예상대로 이메일을 전달하지 않습니다.</p>	<p>설정이 올바른지 확인합니다.</p> <ol style="list-style-type: none"> 도메인에 대한 Amazon SES 확인 프로세스를 완료했어야 합니다. MX 레코드가 AWS 계정 및 리전의 Amazon SES 엔드포인트를 가리키도록 도메인을 올바르게 설정해야 합니다. 자세한 내용은 Amazon SES 설명서에서 Amazon SES 이메일 수신을 위한 MX 레코드 게시하기를 참조하세요. <p>도메인 설정을 확인한 후 다음 단계를 따르세요.</p> <ol style="list-style-type: none"> 솔루션을 배포한 계정 및 지역의 AWS CloudWatch 콘솔을 열고 탐색 창에서 CloudWatch 로그 그룹으로 이동합니다. 로그 그룹 목록에서 SesMailForwardLogGroup 을(를) 검색합니다. 이 그룹의 로그를 조사하여 이메일 벤딩 및 전달 프로세스 중에 오류가 발생하는지 확인합니다.
<p>AWS CDK 스택을 배포하려고 하면 다음 같은 오류가 표시됩니다.</p>	<p>대부분의 경우 이 오류 메시지는 대상 리전에 사용 가능한 AWS 서비스가 모두 제공되지 않음을</p>

문제	Solution
<p>“템플릿 형식 오류: 인식할 수 없는 리소스 유형”</p>	<p>의미합니다. AWS Cloud9를 사용하여 솔루션을 배포하는 경우, AWS Cloud9 인스턴스가 실행되는 리전과 다른 리전을 대상으로 할 수 있습니다.</p> <p>참고: 기본적으로 AWS CDK는 AWS CLI에서 구성된 리전 및 계정에 배포됩니다.</p> <p>가능한 해결책:</p> <ol style="list-style-type: none"> 1. 리전별 AWS 서비스를 검토하여 이 솔루션에 필요한 모든 서비스(이 패턴 앞부분의 대상 기술 스택 섹션 참조)가 대상 AWS 리전에 있는지 조사합니다. 2. AWS Cloud9를 사용하고 AWS Cloud9 인스턴스가 실행되는 리전과 다른 리전을 대상으로 하는 경우 솔루션을 배포하기 전에 <code>AWS_DEFAULT_REGION</code> 환경 변수를 설정하거나 AWS CLI로 리전을 설정해야 합니다. 자세한 내용은 AWS CLI 설명서의 AWS CLI를 구성하기 위한 환경 변수를 참조하세요. 또는 환경에 대한 AWS CDK 설명서의 지침에 따라 하드 코딩된 계정 ID 및 리전을 포함하도록 리포지토리 루트의 <code>app.py</code> 파일을 수정할 수 있습니다.

문제	Solution
<p>솔루션을 배포할 때 다음과 같은 오류 메시지가 나타납니다.</p> <p>“배포 실패: 오류: AwsMailFwdStack: SSM 파라미터 /cdk-bootstrap/hnb659fds/버전을 찾을 수 없습니다. 환경이 부트스트랩되었습니까? 'cdk bootstrap'을 실행하세요.”</p>	<p>대상 AWS 계정 및 리전에 AWS CDK 리소스를 배포한 적이 없는 경우 먼저 오류가 나타내는 대로 <code>cdk bootstrap</code> 명령을 실행해야 합니다. 부트스트래핑 명령을 실행한 후에도 이 오류가 계속 표시된다면, AWS Cloud9 인스턴스가 실행되는 리전과 다른 리전에 솔루션을 배포하려고 하는 것일 수 있습니다.</p> <p>이 문제를 해결하려면 솔루션을 배포하기 전에 AWS CLI로 <code>AWS_DEFAULT_REGION</code> 환경 변수를 설정하거나 리전을 설정합니다. 또는 환경에 대한 AWS CDK 설명서의 지침에 따라 하드 코딩된 계정 ID 및 리전을 포함하도록 리포지토리 루트의 <code>app.py</code> 파일을 수정할 수 있습니다.</p>

관련 리소스

- AWS CLI 설치에 대한 도움이 필요하다면 [AWS CLI의 최신 버전 설치 또는 업데이트](#)를 참조하세요.
- IAM 액세스 보인 인증으로 AWS CLI를 설정하는 데 도움이 필요하다면 [AWS CLI 구성](#)을 참조하세요.
- AWS CDK에 대한 도움이 필요하다면 [AWS CDK 시작하기](#)를 참조하세요.

추가 정보

비용

이 솔루션을 배포할 때 AWS 계정 소유자에게 다음 서비스 사용과 관련된 비용이 발생할 수 있습니다. 잠재적 비용을 인지하려면 이러한 서비스의 청구 방식을 이해하는 것이 중요합니다. 가격 정보는 다음 페이지를 참조하세요.

- [Amazon SES 요금](#)
- [Amazon S3 요금](#)
- [AWS Cloud9 요금](#)
- [AWS KMS 요금](#)

- [AWS Lambda 요금](#)
- [Amazon DynamoDB 요금](#)

다중 계정 AWS 환경에서 하이브리드 네트워크를 위한 DNS 확인 설정

작성자: Amir Durrani

환경: 프로덕션

기술: 인프라, 네트워킹

AWS 서비스: AWS RAM,
Amazon Route 53; AWS
Control Tower

요약

이 패턴은 Amazon Route 53 Resolver 규칙 및 아웃바운드 확인자 엔드포인트와 함께 온프레미스 도메인 이름 시스템(DNS) 서비스를 사용하여 이름 확인을 수행하는 방법을 설명합니다.

DNS는 네트워크 환경 전반에서 통신을 설정하고 유지 관리하는 데 필수적입니다. 하이브리드 네트워크 연결 환경을 사용하는 경우 여러 계정과 Virtual Private Cloud(VPC)에 분산된 환경을 관리하는 운영상의 부담 없이 DNS 및 Active Directory와 같은 중요한 네트워크 서비스를 공유할 수 있습니다. 이 접근 방식은 많은 계정을 포괄하는 애플리케이션을 구축하고 지원하는 데 도움이 됩니다. 예를 들어 하이브리드 연결 요구 사항이 있는 멀티 리전 계정이 수백 또는 수천 개 있는 경우 AWS 조직 내의 모든 연결된 환경 전반에서 DNS 서비스를 안전하고 효율적으로 공유할 수 있습니다.

DNS는 애플리케이션의 모든 계층(웹, 애플리케이션 및 데이터베이스) 간의 IP 네트워킹에 매우 중요합니다. DNS 전문가로 구성된 팀에게만 이 리소스를 구성, 운영 및 지원할 수 있는 전체 액세스 권한을 부여하는 것이 좋습니다. 하이브리드 연결 환경에서는 조건부 전달을 사용하여 다른 계정에 있는 리소스에서 발생하는 이름 확인 요청에 온프레미스 DNS를 계속 사용할 수 있습니다.

이 패턴은 AWS 다중 계정 환경의 하이브리드 DNS 확인을 다룹니다. 단일 계정의 경우 [단일 계정 AWS 환경에서 하이브리드 네트워크를 위한 DNS 확인 설정](#) 패턴을 참조하세요.

사전 조건 및 제한 사항

사전 조건

- 모범 사례를 기반으로 하고 [AWS Control Tower](#)를 사용하여 구축된 AWS 다중 계정 환경. 다음 섹션의 다이어그램은 이러한 환경의 일반적인 아키텍처를 보여줍니다.
- [AWS Transit Gateway](#)를 사용한 계정과 VPC 간에 확장 가능한 라우팅 인프라.

- [Amazon Route 53](#)를 사용한 아웃바운드 확인자 엔드포인트 및 확인자 규칙.
- [AWS Resource Access Manager\(AWS RAM\)](#)를 사용한 아웃바운드 확인자 규칙의 리소스 공유.

아키텍처

AWS 다중 계정 아키텍처

대상 기술 스택

- 다수의 AWS 보안 주체 전반에서 아웃바운드 이름 확인을 위한 기존 온프레미스 DNS 인프라
- Route 53 Resolver 규칙 및 아웃바운드 확인자 엔드포인트
- Route 53 Resolver 규칙을 AWS 조직 내부 및 외부의 다른 AWS 보안 주체와 공유하기 위한 AWS RAM

대상 아키텍처

다음 다이어그램은 end-to-end 하이브리드 DNS 해상도를 구성하는 단계를 보여줍니다. AWS RAM은 중앙 Shared Services 계정에서 구성 및 관리되는 Route 53 Resolver 규칙 및 확인자 엔드포인트를 공유하는 데 사용됩니다. Route 53 Resolver 엔드포인트는 각 가용 영역에 대해 온프레미스 데이터 센터에 있는 리소스에 대한 아웃바운드 이름 확인 요청을 수신한 다음에 이러한 요청을 온프레미스 DNS 확인자로 전달하도록 구성됩니다. 온프레미스 DNS 확인자는 이름 확인 응답을 아웃바운드 엔드포인트로 보내고, 아웃바운드 엔드포인트는 응답을 VPC 확인자에 전달합니다. 이 단계는 IP 주소 대신 호스트 이름을 사용하여 end-to-end 통신을 설정합니다.

다음의 다이어그램은 아키텍처를 더 자세히 보여줍니다.

자동화 및 규모 조정

AWS 템플릿을 사용하여 AWS RAM을 통해 Route 53 리졸버 규칙을 구성하고 공유할 수 있습니다.
CloudFormation

도구

서비스

- [AWS Control Tower](#)는 권장 모범 사례에 따라 AWS 다중 계정 환경을 설정하고 관리할 수 있도록 지원합니다.
- [AWS Resource Access Manager\(AWS RAM\)](#)를 이용하면 AWS 계정 간에 리소스를 안전하게 공유하여 운영 오버헤드를 줄이고 가시성과 감사 가능성을 제공할 수 있습니다.
- [Amazon Route 53](#)는 가용성과 확장성이 뛰어난 DNS 웹 서비스입니다.

추가 도구

- nslookup 및 dig는 DNS 레코드를 쿼리하기 위한 유틸리티입니다.

에픽

확인자 엔드포인트 및 규칙 구성

작업	설명	필요한 기술
Route 53 아웃바운드 확인자 엔드포인트 및 규칙을 구성합니다.	<ol style="list-style-type: none"> 1. Route 53 아웃바운드 확인자 규칙을 구성하고 공유할 AWS 계정의 AWS Management Console에 로그인합니다. 2. https://console.aws.amazon.com/route53/에서 Route 53 콘솔을 엽니다. 3. 탐색 모음에서 확인자 엔드포인트를 구성하고 싶은 리전을 선택합니다. 4. 탐색 창에서 아웃바운드 엔드포인트를 선택하고 엔드포인트 구성을 선택합니다. 5. 일반 설정, IP 주소, 선택적 태그 정보를 제공하고 다음을 선택합니다. 6. 규칙을 한 개 이상 생성하여 네트워크에 전달할 DNS 쿼 	일반 AWS

작업	설명	필요한 기술
	<p>리의 도메인 이름을 지정한 다음에 저장을 선택합니다.</p> <p>자세한 내용은 Route 53 설명서의 네트워크로 아웃바운드 DNS 쿼리 전달을 참조하세요.</p>	

작업	설명	필요한 기술
<p>Route 53 아웃바운드 확인자 규칙을 생성하여 AWS 보안 주체와 공유합니다.</p>	<ol style="list-style-type: none"> 1. https://console.aws.amazon.com/ram/ 에서 AWS RAM 콘솔을 엽니다. 2. 탐색 창에서 리소스 공유를 선택한 다음 리소스 공유 생성을 선택합니다. 3. 공유 이름을 입력합니다. 4. 리소스 유형으로는 확인자 규칙을 선택합니다. 5. 공유하려는 확인자 규칙을 선택하고 선택적 태그 키 및 값 정보를 제공한 후 다음을 선택합니다. 6. 확인자 규칙 리소스를 공유할 보안 주체를 선택합니다. 보안 주체는 AWS 조직 내부 또는 외부에 있을 수 있습니다. 예를 들어 AWS 조직, 조직 내 특정 조직 단위(OU) 또는 특정 계정을 선택할 수 있습니다. 7. 리소스 공유 검토 및 생성합니다. <p>리소스가 생성되고 공유되면 공유되는 보안 주체에 대한 탐색 창의 나와 공유됨 섹션에 해당 리소스가 나타납니다.</p> <ol style="list-style-type: none"> 8. (보안 주체)계정의 VPC를 공유된 서비스 또는 네트워킹 계정에서 공유한 확인자 규칙에 연결합니다. 	<p>일반 AWS</p>

작업	설명	필요한 기술
	자세한 내용은 AWS RAM 설명서의 AWS 리소스 공유 를 참조하세요.	
아웃바운드 DNS 이름 확인을 테스트합니다.	<p>확인자 규칙을 공유한 계정의 VPC 인스턴스에서 nslookup 또는 dig 유틸리티를 사용하여 이름 확인을 테스트합니다.</p> <p>쿼리는 온프레미스 데이터 센터 내에 있는 리소스의 IP 주소로 확인되어야 합니다.</p>	일반 AWS

관련 리소스

- [하이브리드 환경에서 온프레미스 DNS 해결](#)(비디오)
- [네트워크로 아웃바운드 DNS 쿼리 전달](#)(Route 53 설명서)
- [AWS 리소스 공유](#)(AWS RAM 설명서)

단일 계정 AWS 환경에서 하이브리드 네트워크를 위한 DNS 확인 설정

작성자: Abdullahi Olaoye(AWS)

환경: 프로덕션

기술: 인프라

AWS 서비스: Amazon Route 53, Amazon VPC

요약

이 패턴은 관리 오버헤드 없이 온프레미스 리소스, AWS 리소스 및 인터넷 DNS 쿼리의 end-to-end DNS 확인을 지원하는 완전한 하이브리드 도메인 이름 시스템 (DNS) 아키텍처를 설정하는 방법을 설명합니다. 이 패턴은 도메인 이름을 기반으로 AWS에서 시작된 DNS 쿼리를 어디로 보내야 하는지를 결정하는 Amazon Route 53 Resolver 전달 규칙을 설정하는 방법을 설명합니다. 온프레미스 리소스에 대한 DNS 쿼리는 온프레미스 DNS 확인자로 전달됩니다. AWS 리소스에 대한 DNS 쿼리와 인터넷 DNS 쿼리는 Route 53 Resolver에 의해 해결됩니다.

이 패턴은 AWS 단일 계정 환경의 하이브리드 DNS 확인을 다룹니다. AWS 다중 계정 환경에서 아웃바운드 DNS 쿼리를 설정하는 방법에 대한 자세한 내용은 [다중 계정 AWS 환경에서 하이브리드 네트워크를 위한 DNS 해상도 설정](#) 패턴을 참조하세요.

사전 조건 및 제한 사항

사전 조건

- AWS 계정
- AWS 계정의 Virtual Private Cloud(VPC)
- AWS Virtual Private Network(AWS VPN) 또는 AWS Direct Connect를 통해 온프레미스 환경과 VPC 간의 네트워크 연결
- 온프레미스 DNS 확인자의 IP 주소(VPC에서 연결 가능)
- 온프레미스 해석기에 전달할 도메인/하위 도메인 이름(예: onprem.mydc.com)
- AWS 프라이빗 호스팅 영역의 도메인/하위 도메인 이름(예: myvpc.cloud.com)

아키텍처

대상 기술 스택

- Amazon Route 53 프라이빗 호스팅 영역
- Amazon Route 53 Resolver
- Amazon VPC
- AWS VPN 또는 Direct Connect

대상 아키텍처

도구

- [Amazon Route 53 Resolver](#)는 전체 하이브리드 클라우드에서 원활한 DNS 쿼리 해결을 지원하여 기업 고객이 하이브리드 클라우드를 더 쉽게 사용할 수 있도록 합니다. DNS 엔드포인트와 조건부 전달 규칙을 생성하여 온프레미스 데이터 센터와 VPC 간의 DNS 네임스페이스를 해결할 수 있습니다.
- [Amazon Route 53 프라이빗 호스팅 영역](#)은 Amazon VPC 서비스로 생성한 하나 이상의 VPC 내에 있는 도메인과 그 하위 도메인에 대하여 Route 53의 DNS 쿼리 응답 정보가 담긴 컨테이너입니다.

에픽

프라이빗 호스팅 영역 구성

작업	설명	필요한 기술
myvpc.cloud.com과 같은 AWS 예약 도메인 이름을 위한 Route 53 프라이빗 호스팅 영역을 생성합니다.	이 영역에는 온프레미스 환경에서 해결해야 하는 AWS 리소스에 대한 DNS 레코드가 보관됩니다. 지침은 Route 53 설명서의 프라이빗 호스팅 영역 생성 을 참조하세요.	네트워크 관리자, 시스템 관리자
프라이빗 호스팅 영역을 VPC와 연결합니다.	VPC의 리소스가 이 프라이빗 호스팅 영역의 DNS 레코드를	네트워크 관리자, 시스템 관리자

작업	설명	필요한 기술
	해결할 수 있도록 지원하려면 VPC를 호스팅 영역과 연결해야 합니다. 지침은 Route 53 설명서의 프라이빗 호스팅 영역 생성 을 참조하세요.	

Route 53 Resolver 엔드포인트 설정

작업	설명	필요한 기술
인바운드 엔드포인트를 생성합니다.	Route 53 Resolver는 인바운드 엔드포인트를 사용하여 온프레미스 DNS 확인자에서 DNS 쿼리를 받습니다. 자세한 지침은 Route 53 설명서의 인바운드 DNS 쿼리를 VPC에 전달 을 참조하세요. 인바운드 엔드포인트 IP 주소를 기록해 둡니다.	네트워크 관리자, 시스템 관리자
아웃바운드 엔드포인트를 생성합니다.	Route 53 Resolver는 아웃바운드 엔드포인트를 사용하여 온프레미스 DNS 확인자에 DNS 쿼리를 보냅니다. 지침은 Route 53 설명서의 네트워크로 아웃바운드 DNS 쿼리 전달 을 참조하세요. 출력 엔드포인트 ID를 기록해 둡니다.	네트워크 관리자, 시스템 관리자

전달 규칙 설정 및 VPC와 연결

작업	설명	필요한 기술
온프레미스 도메인에 대한 전달 규칙을 생성합니다.	이 규칙은 Route 53 Resolver가 온프레미스 도메인(예:	네트워크 관리자, 시스템 관리자

작업	설명	필요한 기술
	<p>onprem.mydc.com)에 대한 DNS 쿼리를 온프레미스 DNS 확인자로 전달하도록 지시합니다. 이 규칙을 생성하려면 온프레미스 DNS 확인자의 IP 주소와 Route 53 Resolver의 아웃바운드 엔드포인트 ID가 필요합니다. 지침은 Route 53 설명서의 전달 규칙 관리를 참조하세요.</p>	
<p>전달 규칙을 VPC와 연결합니다.</p>	<p>전달 규칙을 적용하려면 규칙을 VPC와 연결해야 합니다. 그러면 Route 53 Resolver는 도메인을 확인할 때 이 규칙을 고려합니다. 지침은 Route 53 설명서의 전달 규칙 관리를 참조하세요.</p>	<p>네트워크 관리자, 시스템 관리자</p>

온프레미스 DNS 확인자 구성

작업	설명	필요한 기술
<p>온프레미스 DNS 확인자에서 조건부 전달을 구성합니다.</p>	<p>온프레미스 환경에서 Route 53 프라이빗 호스팅 영역으로 DNS 쿼리를 보내려면 온프레미스 DNS 확인자에서 조건부 전달을 구성해야 합니다. 이렇게 하면 DNS 확인자가 AWS 도메인(예: myvpc.cloud.com)에 대한 모든 DNS 쿼리를 Route 53 Resolver의 인바운드 엔드포인트 IP 주소로 전달하도록 지시합니다.</p>	<p>네트워크 관리자, 시스템 관리자</p>

DNS 해상도 테스트 end-to-end

작업	설명	필요한 기술
AWS에서 온프레미스 환경으로 DNS 확인을 테스트합니다.	VPC의 서버에서 온프레미스 도메인(예: server1.onprem.mydc.com)에 대한 DNS 쿼리를 수행합니다.	네트워크 관리자, 시스템 관리자
온프레미스 환경에서 AWS로 DNS 확인을 테스트합니다.	온프레미스 서버에서 AWS 도메인(예: server1.myvpc.cloud.com)에 대한 DNS 확인을 수행합니다.	네트워크 관리자, 시스템 관리자

관련 리소스

- [Centralized DNS management of hybrid cloud with Amazon Route 53 and AWS Transit Gateway](#)(AWS 네트워킹 및 콘텐츠 전송 블로그)
- [Simplify DNS management in a multi-account environment with Route 53 Resolver](#)(AWS 보안 블로그)
- [프라이빗 호스팅 영역 작업](#)(Route 53 설명서)
- [Route 53 Resolver 시작하기](#)(Route 53 설명서)

AWS를 사용하여 Amazon EC2에 UiPath RPA 봇을 자동으로 설정합니다. CloudFormation

작성자: Dr. Rahul Sharad Gaikwad(AWS) 및 Tamilselvan P(AWS)

환경: PoC 또는 파일럿

기술: 인프라, DevOps

워크로드: 기타 모든 워크로드

AWS 서비스: 아마존
CloudWatch, 아마존 EC2
Image Builder, AWS Systems
Manager, AWS CloudForm
ation

요약

이 패턴은 Amazon Elastic Compute Cloud(Amazon EC2) 인스턴스에서 로봇 프로세스 자동화(RPA) 봇을 배포하는 방법을 설명합니다. [EC2 Image Builder](#) 파이프라인을 사용하여 사용자 지정 Amazon Machine Image(AMI)를 만듭니다. AMI는 EC2 인스턴스를 배포하기 위해 운영 체제(OS)와 사전 설치된 소프트웨어를 포함하는 사전 구성된 가상 머신(VM) 이미지입니다. 이 패턴은 AWS CloudFormation 템플릿을 사용하여 사용자 지정 AMI에 [UiPath Studio 커뮤니티 에디션](#)을 설치합니다. UiPath 작업을 자동화하도록 로봇을 설정하는 데 도움이 되는 RPA 도구입니다.

이 솔루션의 일부로 기본 AMI를 사용하여 EC2 Windows 인스턴스를 시작하고 인스턴스에 UiPath Studio 애플리케이션을 설치합니다. 이 패턴은 Microsoft System Preparation(Sysprep) 도구를 사용해서 사용자 지정 Windows 설치를 복제합니다. 그런 다음 호스트 정보를 제거하고 인스턴스에서 최종 AMI를 생성합니다. 이후에 자체 명명 규칙 및 모니터링 설정이 적용된 최종 AMI를 사용하여 필요에 따라 인스턴스를 시작할 수 있습니다.

참고: 이 패턴은 RPA 봇 사용에 대한 어떠한 정보도 제공하지 않습니다. 자세한 내용은 [UiPath 설명서](#)를 참조하십시오. 요구 사항에 따라 설치 단계를 사용자 지정하여 이 패턴으로 다른 RPA 봇 애플리케이션을 설정할 수도 있습니다.

이 패턴은 다음과 같은 자동화 및 이점을 제공합니다.

- 애플리케이션 배포 및 공유: CloudFormation AWS 템플릿을 코드형 인프라 (IaC) 스크립트로 사용하는 EC2 Image Builder 파이프라인을 통해 애플리케이션 배포용 Amazon EC2 AMI를 구축하고 여러 계정 간에 공유할 수 있습니다.
- Amazon EC2 프로비저닝 및 확장: CloudFormation IaC 템플릿은 사용자 지정 컴퓨터 이름 시퀀스와 Active Directory 가입 자동화를 제공합니다.
- 관찰 가능성 및 모니터링: 이 패턴은 Amazon EC2 지표 (예: CPU 및 디스크 사용량) 를 모니터링하는 데 도움이 되는 Amazon CloudWatch 대시보드를 설정합니다.
- 비즈니스에 미치는 RPA의 이점: 로봇이 할당된 작업을 자동으로 일관되게 수행할 수 있으므로 RPA를 통해 정확성이 향상됩니다. 또한 RPA는 가치를 창출하지 않는 작업을 없애고 반복적인 활동을 처리하므로 속도와 생산성을 높여줍니다.

사전 조건 및 제한 사항

사전 조건

- 활성 상태의 [AWS 계정](#)
- 템플릿 CloudFormation 배포를 위한 [AWS ID 및 액세스 관리 \(IAM\) 권한](#)
- EC2 Image Builder를 사용하여 교차 계정 AMI 배포를 설정하는 [IAM 정책](#)

아키텍처

1. 관리자는 `ec2-image-builder.yaml` 파일에 기본 Windows AMI를 제공하고 CloudFormation 콘솔에 스택을 배포합니다.
2. CloudFormation 스택은 다음 리소스를 포함하는 EC2 Image Builder 파이프라인을 배포합니다.
 - Ec2ImageInfraConfiguration
 - Ec2ImageComponent
 - Ec2ImageRecipe
 - Ec2AMI
3. EC2 Image Builder 파이프라인은 기본 AMI를 사용하여 임시 Windows EC2 인스턴스를 시작하고 필요한 구성 요소 (이 경우 UiPath Studio) 를 설치합니다.
4. EC2 Image Builder는 모든 호스트 정보를 제거하고 Windows 서버에서 AMI를 생성합니다.

5. 사용자 지정 AMI로 ec2-provisioning yaml 파일을 업데이트하고 요구 사항에 따라 여러 EC2 인스턴스를 시작합니다.
6. 템플릿을 사용하여 Count 매크로를 배포합니다. CloudFormation 이 매크로는 CloudFormation 리소스의 Count 속성을 제공하므로 동일한 유형의 여러 리소스를 쉽게 지정할 수 있습니다.
7. CloudFormation ec2-provisioning.yaml 파일에서 매크로 이름을 업데이트하고 스택을 배포합니다.
8. 관리자는 요구 사항에 따라 ec2-provisioning.yaml 파일을 업데이트하고 스택을 시작합니다.
9. 템플릿은 UiPath Studio 애플리케이션과 함께 EC2 인스턴스를 배포합니다.

도구

서비스

- [AWS](#)는 자동화되고 안전한 방식으로 인프라 리소스를 모델링하고 관리할 수 있도록 CloudFormation 지원합니다.
- [CloudWatchAmazon](#)은 AWS, 온프레미스 및 기타 클라우드의 리소스와 애플리케이션을 관찰하고 모니터링할 수 있도록 지원합니다.
- [Amazon Elastic Compute Cloud\(Amazon EC2\)](#)는 AWS 클라우드에서 안전하고 확장 가능한 컴퓨팅 용량을 제공합니다. 필요한 만큼 가상 서버를 시작하고 빠르게 스케일 업하거나 스케일 다운할 수 있습니다.
- [EC2 Image Builder](#)는 AWS 또는 온프레미스에서 사용할 가상 머신 및 컨테이너 이미지의 구축, 테스트 및 배포를 간소화합니다.
- [EventBridgeAmazon](#)은 AWS, 기존 시스템 또는 서비스형 소프트웨어 (SaaS) 애플리케이션 전반에서 이벤트 기반 애플리케이션을 대규모로 구축할 수 있도록 지원합니다.
- [AWS Identity and Access Management \(IAM\)](#)를 사용하여 AWS 리소스에 대한 액세스를 안전하게 제어할 수 있습니다. IAM을 사용하면 사용자가 액세스할 수 있는 AWS 리소스 제어 권한을 중앙에서 관리할 수 있습니다. IAM을 사용하여 리소스를 사용하도록 인증(로그인) 및 권한 부여(권한 있음)된 대상을 제어합니다.
- [AWS Lambda](#)는 서버를 프로비저닝하거나 관리하지 않고도 사실상 모든 유형의 애플리케이션이나 백엔드 서비스에 대한 코드를 실행할 수 있는 서버리스 이벤트 기반 컴퓨팅 서비스입니다. 200개 이상의 AWS 서비스와 SaaS 애플리케이션에서 Lambda 함수를 호출할 수 있으며, 사용한 내역에 대해서만 요금을 지불합니다.
- [Amazon Simple Storage Service\(S3\)](#)는 원하는 양의 데이터를 저장, 보호 및 검색하는 데 도움이 되는 클라우드 기반 객체 스토리지 서비스입니다.

- [AWS Systems Manager Agent\(SSM Agent\)](#)는 Systems Manager가 EC2 인스턴스, 엣지 디바이스, 온프레미스 서버 및 가상 머신(VM)을 업데이트, 관리 및 구성할 수 있도록 합니다.

코드 리포지토리

이 패턴의 코드는 리포지토리를 사용하는 GitHub [UiPath RPA 봇](#) 설정에서 사용할 수 있습니다.

CloudFormation 또한 패턴은 [AWS CloudFormation Macros 리포지토리에서 사용할 수 있는 매크로를](#) 사용합니다.

모범 사례

- AWS는 매달 새로운 [Windows AMI](#)를 출시합니다. 여기에는 최신 OS 패치, 드라이버 및 시작 에이전트가 포함됩니다. 새 인스턴스를 시작하거나 자체 사용자 지정 이미지를 만들 때는 최신 AMI를 사용해야 합니다.
- 이미지 빌드 중에 사용 가능한 모든 Windows 또는 Linux 보안 패치를 적용합니다.

에픽

기본 이미지에 대한 이미지 파이프라인을 배포합니다.

작업	설명	필요한 기술
EC2 Image Builder 파이프라인을 설정합니다.	<ol style="list-style-type: none"> 1. CloudFormation 리포지토리를 사용하여 UiPath RPA 봇 설정을 복제하거나 리포지토리에서 <code>ec2-image-builder.yaml</code> 템플릿을 다운로드하십시오. 2. AWS 관리 콘솔에 로그인하고 AWS CloudFormation 콘솔을 엽니다. 3. 스택 생성을 선택합니다. 4. 템플릿 지정 섹션에서 템플릿 파일 업로드를 선택합니다. 	AWS DevOps

작업	설명	필요한 기술
	<p>5. 컴퓨터에서 <code>ec2-image-builder.yaml</code> 템플릿을 찾아 업로드한 후 다음을 선택합니다.</p> <p>6. 스택에 대한 입력 파라미터를 제공하거나 기본값을 수락합니다. 다음을 선택합니다.</p> <p>참고: 파라미터의 수와 값은 입력 값에 따라 달라질 수 있습니다.</p> <p>7. 선택적으로 스택 옵션을 구성한 후 다음을 선택합니다.</p> <p>8. 스택 세부 정보를 검토합니다.</p> <p>9. 화면 끝에서 확인란을 선택하여 기능을 확인한 다음 제출을 선택합니다.</p> <p>10. 스택의 진행 상황을 모니터링합니다. 상태가 <code>CREATE_COMPLETE</code> 이면 배포가 준비된 것입니다.</p>	

작업	설명	필요한 기술
<p>EC2 Image Builder 설정을 확인합니다.</p>	<p>EC2 Image Builder 설정에는 인프라 구성, 배포 설정 및 보안 스캔 설정이 포함됩니다. 다음과 같이 설정을 확인합니다.</p> <ol style="list-style-type: none"> 1. EC2 Image Builder 콘솔을 엽니다. 2. 탐색 창에서 다양한 Image Builder 설정으로 이동합니다. <p>참고: 가장 좋은 방법은 템플릿을 통해서만 EC2 Image Builder를 업데이트하는 CloudFormation 것입니다.</p>	<p>AWS DevOps</p>
<p>이미지 파이프라인을 확인합니다.</p>	<p>다음과 같이 배포된 이미지 파이프라인을 확인합니다.</p> <ol style="list-style-type: none"> 1. EC2 Image Builder 콘솔의 탐색 창에서 이미지 파이프라인을 선택합니다. 2. 생성한 이미지 파이프라인을 선택합니다. 3. 출력 이미지, 이미지 레시피, 인프라 구성, 배포 설정, Amazon EventBridge 규칙 및 태그의 구성 세부 정보를 확인합니다. 	<p>AWS DevOps</p>

작업	설명	필요한 기술
<p>Image Builder 로그를 확인합니다.</p>	<p>EC2 Image Builder 로그는 로그 그룹으로 집계됩니다 CloudWatch . 로그인 로그를 보려면: CloudWatch</p> <ol style="list-style-type: none"> 1. CloudWatch 콘솔을 엽니다. 2. 왼쪽 탐색 창에서 로그, 로그 그룹을 선택합니다. 3. 로그 그룹 이름을 선택합니다. EC2 Image Builder 로그는 /aws/imagebuilder/XXX 로그 그룹에 집계됩니다. 4. 이미지 파이프라인을 실행할 때 발생한 오류가 있는지 각 로그 스트림의 최신 로그를 확인합니다. <p>EC2 Image Builder 로그는 S3 버킷에도 저장됩니다. 다음과 같이 버킷에서 로그를 확인합니다.</p> <ol style="list-style-type: none"> 1. Amazon S3 콘솔을 엽니다. 2. 버킷 목록에서 버킷 이름을 선택합니다. 로그는 S3 버킷 <stack-name>-XXXXXX X 에 집계됩니다. 	<p>AWS DevOps</p>

작업	설명	필요한 기술
UiPath 파일을 S3 버킷에 업로드합니다.	<ol style="list-style-type: none"> https://download.uipath.com/UiPathStudioCommunity.msi 위치에서 UiPath 스튜디오용 .msi 파일을 다운로드합니다. 파일을 S3 버킷에 업로드합니다. ec2-image-builder.yaml 템플릿의 사용자 데이터 섹션 라인 번호 310에서 버킷 이름과 파일 키를 업데이트합니다. 	AWS DevOps

Count 매크로 배포 및 테스트

작업	설명	필요한 기술
Count 매크로를 배포합니다.	<ol style="list-style-type: none"> 카운트 CloudFormation 매크로를 복제하거나 다운로드하십시오. Count 폴더로 이동합니다. CloudFormation 아티팩트를 저장하려면 S3 버킷이 필요합니다. 아직 S3 버킷이 없는 경우, <code>aws s3 mb s3://<bucket name></code> 이름으로 한 개를 생성합니다. Count 매크로 템플릿을 패키징합니다. 템플릿은 AWS Serverless Application Model(SAM)을 사용하므로 	DevOps 엔지니어

작업	설명	필요한 기술
	<p>배포하려면 먼저 변환해야 합니다.</p> <pre>aws cloudformation package \ --template-file template.yaml \ --s3-bucket <your bucket name here> \ --output- template-file packaged.yaml</pre> <p>예:</p> <pre>aws cloudformation package \ --template-file template.yaml \ --s3-bucket count-macro-ec2 \ --output- template-file packaged.yaml</pre> <p>5. 패키징된 템플릿을 배포하여 CloudFormation 스택을 생성합니다.</p> <pre>aws cloudformation deploy \ --stack-name Count-macro \ --template-file packaged.yaml \ --capabilities CAPABILITY_IAM</pre>	

작업	설명	필요한 기술
	<p>콘솔을 사용하려면 이전 에픽이나 CloudFormation 설명서의 지침을 따르세요.</p>	
<p>Count 매크로를 테스트합니다.</p>	<p>매크로의 기능을 테스트하려면 매크로와 함께 제공된 예제 템플릿을 실행합니다.</p> <pre data-bbox="597 556 1024 913"> aws cloudformation deploy \ --stack-name Count- test \ --template-file test.yaml \ --capabilities CAPABILITY_IAM </pre>	<p>DevOps 엔지니어</p>

CloudFormation 스택을 배포하여 사용자 지정 이미지로 인스턴스를 프로비저닝합니다.

작업	설명	필요한 기술
<p>Amazon EC2 프로비저닝 템플릿을 배포합니다.</p>	<p>다음을 사용하여 CloudFormation EC2 이미지 파이프라인을 배포하려면</p> <ol style="list-style-type: none"> 리포지토리에서 ec2-provisioning.yaml 템플릿을 다운로드하거나, GitHub 리포지토리를 복제한 경우 컴퓨터에서 템플릿을 찾을 수 있습니다. CloudFormation 콘솔을 엽니다. 첫 번째 에픽의 단계를 반복하거나 CloudFormation 	<p>AWS DevOps</p>

작업	설명	필요한 기술
	<p>설명서의 지침을 따라 배포하세요. ec2-provisioning.yaml</p>	
<p>Amazon EC2 설정을 확인합니다.</p>	<p>Amazon EC2 설정에는 보안, 네트워킹, 스토리지, 상태 확인, 모니터링 및 태그 구성이 포함됩니다. 다음과 같이 이러한 구성을 확인합니다.</p> <ol style="list-style-type: none"> 1. Amazon EC2 콘솔을 엽니다. 2. 탐색 창에서 인스턴스를 선택한 다음 Amazon EC2 프로비저닝 템플릿에서 만든 EC2 인스턴스를 선택합니다. 3. 인스턴스 요약에서 탭을 선택하여 해당하는 Amazon EC2 설정을 확인합니다. 	<p>AWS DevOps</p>

작업	설명	필요한 기술
CloudWatch 대시보드 보기.	<ol style="list-style-type: none"> 1. CloudWatch 콘솔을 엽니다. 2. 탐색 창에서 대시보드를 선택합니다. 3. 해당하는 스택 이름이 있는 대시보드를 선택합니다. <p>참고: 스택을 프로비저닝한 후 대시보드를 지표로 채우는 데 시간이 걸립니다.</p> <p>대시보드는 CPUUtilization , DiskUtilization , MemoryUtilization , NetworkIn , NetworkOut , StatusCheckFailed 지표를 제공합니다.</p>	AWS DevOps
메모리 및 디스크 사용량에 대한 사용자 지정 지표를 확인합니다.	<ol style="list-style-type: none"> 1. CloudWatch 콘솔에서 대시보드를 선택합니다. 2. 탐색 창에서 지표, 모든 지표를 선택합니다. 3. 사용자 지정 네임스페이스 및 CWAgent를 선택합니다. 	AWS DevOps
메모리 및 디스크 사용량에 대한 경보를 확인합니다.	<ol style="list-style-type: none"> 1. CloudWatch 콘솔의 탐색 창에서 대시보드를 선택합니다. 2. 모든 경보를 선택합니다. 	AWS DevOps

작업	설명	필요한 기술
스냅샷 수명 주기 규칙을 확인합니다.	<ol style="list-style-type: none"> 1. Amazon EC2 콘솔을 엽니다. 2. 탐색 창에서 수명 주기 관리자를 선택합니다. 3. AMI 수명 주기에 대한 설정을 확인합니다. 	AWS DevOps

환경 삭제(선택 사항)

작업	설명	필요한 기술
스택을 삭제합니다.	<p>PoC 또는 파일럿 프로젝트가 완료되면 이러한 리소스에 대한 요금이 부과되지 않도록 생성한 스택을 삭제하는 것이 좋습니다.</p> <ol style="list-style-type: none"> 1. AWS CloudFormation 콘솔을 엽니다. 2. 탐색 창에서 스택을 선택한 다음 이전에 만든 스택 중 삭제하려는 하나 또는 두 개의 스택을 선택합니다. 스택이 현재 실행 중이어야 합니다. 3. 스택 세부 정보 창에서 삭제를 선택합니다. 4. 메시지가 나타나면 스택 삭제를 선택합니다. <p>중요: 스택 삭제 작업은 시작된 후에는 중지할 수 없습니다. 스택이 DELETE_IN_PROGRESS 상태로 바뀝니다.</p>	AWS DevOps

작업	설명	필요한 기술
	<p>삭제에 실패하면 스택이 DELETE_FAILED 상태가 됩니다. 해결 방법은 AWS CloudFormation 문제 해결 설명서에서 스택 삭제 실패를 참조하십시오.</p> <p>스택이 실수로 삭제되지 않도록 보호하는 방법에 대한 자세한 내용은 AWS CloudFormation 설명서의 스택 삭제 보호를 참조하십시오.</p>	

문제 해결

문제	Solution
<p>Amazon EC2 프로비저닝 템플릿을 배포할 때 다음 오류가 발생할 수 있습니다. Received malformed response from transform 123xxxx:: Count.</p>	<p>이는 알려진 문제입니다. (AWS CloudFormation 매크로 리포지토리의 사용자 지정 솔루션 및 PR을 참조하십시오.)</p> <p>이 문제를 해결하려면 AWS Lambda 콘솔을 열고 리포지토리의 콘텐츠로 index.py 업데이트 하십시오. GitHub</p>

관련 리소스

GitHub 리포지토리

- [UiPath RPA 봇 설정 \(사용\) CloudFormation](#)
- [카운트 매크로 CloudFormation](#)

AWS 참조

- [AWS CloudFormation 콘솔에서 스택 생성](#) (CloudFormation 설명서)
- [문제 해결 CloudFormation](#) (CloudFormation 설명서)
- [Amazon EC2 인스턴스의 메모리 및 디스크 지표 모니터링](#)(Amazon EC2 설명서)
- [CloudWatch 에이전트를 사용하여 Windows 서버의 성능 모니터에 대한 메트릭을 보려면 어떻게 해야 합니까?](#) (AWS re:Post article)

추가 참조

- [UiPath 설명서](#)
- [SysPreped AMI에서 호스트 이름 설정](#) (브라이언 비치의 블로그 게시물)
- [파라미터가 변경될 때 Cloudformation이 매크로를 사용하여 템플릿을 재처리하도록 하려면 어떻게 해야 하나요?](#) (스택 오버플로)

EnterpriseOne AWS Elastic 재해 복구를 사용하여 Oracle JD Edwards의 재해 복구 설정

작성자: Thanigaivel Thirumalai(AWS)

환경: 프로덕션

기술: 인프라, 마이그레이션, 네트워킹

워크로드: Oracle

AWS 서비스: AWS Elastic Disaster Recovery, Amazon EC2

요약

자연 재해, 애플리케이션 장애 또는 서비스 중단으로 인한 재해는 수익에 손해를 끼치고 기업 애플리케이션의 가동 중지를 초래합니다. 이러한 이벤트의 영향을 줄이려면 JD Edwards ERP (전사적 자원 관리) 시스템 및 기타 업무상 중요하고 비즈니스에 중요한 소프트웨어를 채택하는 EnterpriseOne 기업의 경우 재해 복구 (DR) 계획이 매우 중요합니다.

이 패턴은 기업이 AWS Elastic 재해 복구를 JD Edwards EnterpriseOne 애플리케이션의 DR 옵션으로 사용하는 방법을 설명합니다. 또한 탄력적 재해 복구 장애 조치 및 장애 복구를 사용하여 AWS 클라우드의 Amazon Elastic Compute Cloud (Amazon EC2) 인스턴스에 호스팅되는 데이터베이스에 대한 지역 간 DR 전략을 수립하는 단계를 설명합니다.

참고: 이 패턴을 사용하려면 리전 간 DR 구현을 위한 기본 및 보조 리전을 AWS에서 호스팅해야 합니다.

[Oracle JD EnterpriseOne Edwards](#)는 다양한 산업의 중견기업에서 대기업까지 사용할 수 있는 통합 ERP 소프트웨어 솔루션입니다.

AWS Elastic Disaster Recovery는 저렴한 스토리지, 최소한의 컴퓨팅 및 복구를 사용하여 온프레미스 및 클라우드 기반 애플리케이션을 빠르고 안정적으로 복구함으로써 다운타임과 데이터 손실을 최소화합니다. point-in-time

AWS는 [네 가지 핵심 DR 아키텍처 패턴](#)을 제공합니다. 이 문서는 [파일럿 라이트 전략](#)을 사용한 설정, 구성 및 최적화에 중점을 둡니다. 이 전략을 사용하면 소스 데이터베이스의 데이터를 복제하기 위한 복

제 서버를 처음에 프로비저닝하고 DR 드릴 및 복구를 시작할 때만 실제 데이터베이스 서버를 프로비저닝하는 저렴한 DR 환경을 생성할 수 있습니다. 이 전략은 DR 리전에서 데이터베이스 서버를 유지 관리하는 데 드는 비용을 제거합니다. 대신 복제 서버 역할을 하는 더 작은 EC2 인스턴스에 대한 비용을 지불합니다.

사전 조건 및 제한 사항

사전 조건

- 활성 상태의 AWS 계정
- 오라클 데이터베이스 또는 Microsoft SQL Server에서 실행되는 JD Edwards EnterpriseOne 애플리케이션으로, 지원되는 데이터베이스가 관리형 EC2 인스턴스에서 실행 중인 상태입니다. 이 애플리케이션에는 하나의 AWS 지역에 설치된 모든 JD Edwards EnterpriseOne 기본 구성 요소 (엔터프라이즈 서버, HTML 서버, 데이터베이스 서버)가 포함되어야 합니다.
- Elastic Disaster Recovery 서비스를 설정하기 위한 AWS Identity 및 Access Management(IAM) 역할.
- 필수 [연결 설정](#)에 따라 구성된 Elastic Disaster Recovery를 실행하기 위한 네트워크.

제한 사항

- 이 패턴을 사용하여 모든 티어를 복제할 수 있습니다. 단, 데이터베이스를 Amazon Relational Database Service(Amazon RDS)에서 호스팅하는 경우 Amazon RDS의 [리전 간 복사 기능](#)을 사용하는 것이 좋습니다.
- Elastic 재해 복구는 CloudEndure 재해 복구와 호환되지 않지만 재해 복구에서 CloudEndure 업그레이드할 수 있습니다. 자세한 내용은 Elastic Disaster Recovery 설명서의 [FAQ](#)를 참조하세요.
- Amazon Elastic Block Store(Amazon EBS)는 스냅샷을 생성할 수 있는 속도를 제한합니다. Elastic Disaster Recovery를 사용하면 하나의 AWS 계정에서 최대 300개의 서버를 복제할 수 있습니다. 더 많은 서버를 복제하려면 여러 AWS 계정이나 여러 대상 AWS 리전을 사용할 수 있습니다. (각 계정 및 리전에 대해 Elastic Disaster Recovery를 별도로 설정해야 합니다.) 자세한 내용은 Elastic Disaster Recovery 설명서의 [모범 사례](#)를 참조하세요.
- 원본 워크로드 (JD Edwards EnterpriseOne 애플리케이션 및 데이터베이스)는 EC2 인스턴스에 호스팅되어야 합니다. 이 패턴은 온프레미스 또는 다른 클라우드 환경에 있는 워크로드를 지원하지 않습니다.
- 이 패턴은 JD Edwards 구성 요소에 중점을 둡니다. EnterpriseOne 전체 DR 및 비즈니스 연속성 계획(BCP)에는 다음과 같은 기타 핵심 서비스가 포함되어야 합니다.
 - 네트워킹(Virtual Private Cloud, 서브넷, 보안 그룹)

- Active Directory
- 아마존 WorkSpaces
- Elastic Load Balancing
- Amazon Relational Database Service (Amazon RDS)와 같은 관리형 데이터베이스 서비스

사전 조건, 구성 및 제한 사항에 대한 추가 정보는 [Elastic Disaster Recovery 설명서](#)를 참조하세요.

제품 버전

- 오라클 JD 에드워즈 EnterpriseOne (오라클 최소 기술 요구 사항에 기반한 오라클 및 SQL Server 지원 버전)

아키텍처

대상 기술 스택

- 프로덕션 및 비프로덕션용 단일 리전 및 단일 Virtual Private Cloud(VPC), DR용 두 번째 리전
- 서버 간 지연 시간을 줄이기 위한 단일 가용 영역
- 네트워크 트래픽을 분산하여 여러 가용 영역에서 애플리케이션의 확장성과 가용성을 개선하기 위해 네트워크 트래픽을 분산하는 Application Load Balancer
- 도메인 이름 시스템(DNS) 구성을 제공하는 Amazon Route 53
- Amazon WorkSpaces , 사용자에게 클라우드에서의 데스크톱 경험 제공
- 백업, 파일 및 객체를 저장하기 위한 Amazon Simple Storage Service(S3)
- 애플리케이션 로깅, 모니터링 및 경보를 CloudWatch 위한 Amazon
- 재해 복구를 위한 Amazon Elastic Disaster Recovery

대상 아키텍처

다음 다이어그램은 Elastic 재해 복구를 EnterpriseOne 사용하는 JD Edwards의 지역 간 재해 복구 아키텍처를 보여줍니다.

절차

다음은 프로세스에 대한 높은 수준의 검토입니다. 자세한 내용은 에픽 섹션을 참조하세요.

- Elastic Disaster Recovery 복제는 초기 동기화로 시작됩니다. 초기 동기화 중 AWS Replication Agent는 소스 디스크의 모든 데이터를 스테이징 영역 서브넷의 적절한 리소스로 복제합니다.
- 연속 복제는 초기 동기화가 완료된 후에도 무기한 계속됩니다.
- 에이전트가 설치되고 복제가 시작된 후 서비스별 구성 및 Amazon EC2 시작 템플릿이 포함된 시작 파라미터를 검토합니다. 소스 서버가 복구 준비 완료로 표시되면 인스턴스를 시작할 수 있습니다.
- Elastic Disaster Recovery가 일련의 API 호출을 실행하여 시작 작업을 시작하면 시작 설정에 따라 복구 인스턴스가 즉시 AWS에서 시작됩니다. 서비스는 시작 중에 변환 서버를 자동으로 가동합니다.
- 새 인스턴스는 변환이 완료되고 사용할 준비가 완료되면 AWS에서 실행됩니다. 시작 시점의 소스 서버 상태는 시작된 인스턴스와 관련된 볼륨으로 표시됩니다. 전환 프로세스에는 인스턴스가 AWS에서 기본적으로 부팅되도록 드라이버, 네트워크 및 운영 체제 라이선스를 변경하는 작업이 포함됩니다.
- 실행 후에 새로 생성된 볼륨은 더 이상 소스 서버와 동기화되지 않습니다. AWS Replication Agent는 소스 서버의 변경 사항을 스테이징 영역 볼륨에 계속 정기적으로 복제하지만 시작된 인스턴스는 해당 변경 사항을 반영하지 않습니다.
- 새 드릴 또는 복구 인스턴스를 시작하면 데이터는 항상 소스 서버에서 스테이징 영역 서브넷으로 복제된 가장 최근 상태에 반영됩니다.
- 소스 서버가 복구 준비 중으로 표시되면 인스턴스를 시작할 수 있습니다.

참고: 이 프로세스는 기본 AWS 리전에서 DR 리전으로 장애 조치하는 경우와 복구되면 기본 사이트로 페일백하는 두 가지 방식으로 작동합니다. 완전히 오케스트레이션된 방식으로 대상 시스템에서 소스 시스템으로 데이터 복제 방향을 반대로 바꾸어 페일백에 대비할 수 있습니다.

이 패턴에 설명된 이 프로세스의 이점은 다음과 같습니다.

- 유연성: 복제 서버는 데이터 세트 및 복제 시간에 따라 스케일 아웃 및 스케일 인하므로 소스 워크로드나 복제를 중단하지 않고도 DR 테스트를 수행할 수 있습니다.
- 신뢰성: 복제는 강력하고 중단이 없으며 지속적입니다.
- 자동화: 이 솔루션은 테스트, 복구 및 페일백을 위한 통합되고 자동화된 프로세스를 제공합니다.
- 비용 최적화: 필요한 볼륨만 복제하고 비용을 지불하며 해당 리소스가 활성화된 경우에만 DR 사이트의 컴퓨팅 리소스 비용을 지불할 수 있습니다. 여러 소스나 대규모 EBS 볼륨이 큰 단일 소스에 비용 최적화된 복제 인스턴스(컴퓨팅 최적화 인스턴스 유형 사용 권장)를 사용할 수 있습니다.

자동화 및 규모 조정

대규모 재해 복구를 수행할 경우 JD Edwards EnterpriseOne 서버는 해당 환경의 다른 서버에 종속됩니다. 예:

- 부팅 시 JD Edwards EnterpriseOne 지원 데이터베이스에 연결하는 JD Edwards EnterpriseOne 애플리케이션 서버는 해당 데이터베이스에 종속됩니다.
- 인증이 필요하고 서비스를 시작하기 위해 부팅 시 도메인 컨트롤러에 연결해야 하는 JD Edwards EnterpriseOne 서버는 도메인 컨트롤러에 종속되어 있습니다.

따라서 장애 조치 작업을 자동화하는 것이 좋습니다. 예를 들어, AWS Lambda 또는 AWS Step Functions를 사용하여 JD EnterpriseOne Edwards 시작 스크립트와 로드 밸런서 변경을 자동화하여 장애 조치 프로세스를 자동화할 수 있습니다. end-to-end 자세한 내용은 [Creating a scalable disaster recovery plan with AWS Elastic Disaster Recovery](#) 블로그 게시물을 참조하세요.

도구

서비스

- [Amazon Elastic Block Store\(Amazon EBS\)](#)는 EC2 인스턴스에 사용할 수 있는 블록 수준 스토리지 볼륨을 제공합니다.
- [Amazon Elastic Compute Cloud\(Amazon EC2\)](#)는 AWS 클라우드에서 확장 가능한 컴퓨팅 용량을 제공합니다. 필요한 만큼 가상 서버를 시작하고 빠르게 스케일 업하거나 스케일 다운할 수 있습니다.
- [AWS Elastic Disaster Recovery](#)는 저렴한 스토리지, 최소한의 컴퓨팅 및 복구를 사용하여 온프레미스 및 클라우드 기반 애플리케이션을 빠르고 안정적으로 복구함으로써 다운타임과 데이터 손실을 최소화합니다. point-in-time
- [Amazon Virtual Private Cloud\(Amazon VPC\)](#)를 사용하면 리소스 배치, 연결 및 보안을 포함하여 가상 네트워킹 환경을 완전히 제어할 수 있습니다.

모범 사례

일반 모범 사례

- 실제 복구 이벤트 발생 시 어떻게 해야 할지에 대한 계획을 서면으로 작성해 둡니다.
- Elastic Disaster Recovery를 올바르게 설정한 후 필요에 따라 온디맨드 구성을 생성할 수 있는 AWS CloudFormation 템플릿을 생성하십시오. 서버와 애플리케이션을 시작해야 하는 순서를 결정하고 이를 복구 계획에 기록합니다.
- 정기적인 드릴을 수행합니다(표준 Amazon EC2 요금 적용).

- Elastic Disaster Recovery 콘솔을 사용하거나 프로그래밍 방식으로 진행 중인 복제 상태를 모니터링합니다.
- point-in-time 스냅샷을 보호하고 인스턴스를 종료하기 전에 확인하십시오.
- AWS Replication Agent 설치를 위한 IAM 역할을 생성합니다.
- 실제 DR 시나리오에서 복구 인스턴스에 대한 종료 보호를 활성화합니다.
- 실제 복구 이벤트가 발생한 경우에도 복구 인스턴스를 시작한 서버에 대해 Elastic Disaster Recovery 콘솔의 AWS에서 연결 해제 작업을 사용하지 마세요. 연결을 끊으면 point-in-time (PIT) 복구 지점을 포함하여 이러한 원본 서버와 관련된 모든 복제 리소스가 종료됩니다.
- PIT 정책을 변경하여 스냅샷 보존 일수를 변경합니다.
- Elastic Disaster Recovery 시작 설정에서 시작 템플릿을 편집하여 대상 서버의 서브넷, 보안 그룹 및 인스턴스 유형을 올바르게 설정합니다.
- Lambda 또는 Step Functions를 사용하여 EnterpriseOne JD Edwards 시작 스크립트와 로드 밸런서 변경을 자동화하여 end-to-end 페일오버 프로세스를 자동화합니다.

JD Edwards 최적화 및 고려 사항 EnterpriseOne

- PrintQueue데이터베이스로 이동.
- MediaObjects데이터베이스로 이동.
- 배치 및 로직 서버에서 로그 및 임시 폴더를 제외합니다.
- WebLogicOracle에서 임시 폴더를 제외하십시오.
- 장애 조치 후 시작을 위한 스크립트를 생성합니다.
- SQL Server의 tempdb를 제외합니다.
- Oracle의 임시 파일을 제외합니다.

에픽

초기 작업 및 구성 수행

작업	설명	필요한 기술
복제 네트워크를 설정합니다.	기본 AWS 리전에 JD Edwards EnterpriseOne 시스템을 구현하고 DR을 위한 AWS 지역을 식별하십시오. Elastic Disaster	AWS 관리자

작업	설명	필요한 기술
	Recovery 설명서의 복제 네트워크 요구 사항 섹션에 있는 단계에 따라 복제 및 DR 네트워크를 계획하고 설정하십시오.	
RPO 및 RTO를 결정합니다.	애플리케이션 서버 및 데이터베이스의 Recovery Time Objective (RTO) 및 Recovery Point Objective (RPO)를 식별합니다.	클라우드 아키텍트, DR 아키텍트
Amazon EFS에 대한 복제를 활성화합니다.	해당하는 경우, AWS DataSync, rsync 또는 다른 적절한 도구를 사용하여 Amazon Elastic File System (Amazon EFS) 과 같은 공유 파일 시스템에 대해 AWS 프라이머리에서 DR 리전으로 복제할 수 있도록 하십시오.	클라우드 관리자
DR의 경우 DNS를 관리합니다.	DR 드릴 또는 실제 DR 중에도 메인 이름 시스템(DNS)을 업데이트하는 프로세스를 식별합니다.	클라우드 관리자
설정용 IAM 역할을 생성합니다.	Elastic Disaster Recovery 설명서의 Elastic Disaster Recovery 초기화 및 권한 섹션에 있는 지침에 따라 AWS 서비스를 초기화하고 관리하는 IAM 역할을 생성합니다.	클라우드 관리자

작업	설명	필요한 기술
VPC 피어링을 설정합니다.	소스 및 대상 VPC가 피어링되고 서로 액세스할 수 있는지 확인합니다. 구성 지침은 Amazon VPC 설명서 를 참조하세요.	AWS 관리자

Elastic Disaster Recovery 복제 설정 구성

작업	설명	필요한 기술
Elastic Disaster Recovery를 초기화합니다.	Elastic Disaster Recovery 콘솔 을 열고 대상 AWS 리전(데이터를 복제하고 복구 인스턴스를 시작할 리전)을 선택한 다음 기본 복제 설정 설정을 선택합니다.	AWS 관리자
복제 서버를 설정합니다.	<ol style="list-style-type: none"> 복제 서버 설정 창에서 스테이징 영역 서브넷 및 복제 서버 인스턴스 유형을 입력합니다. t3.small 인스턴스 유형이 기본적으로 선택됩니다. 요구 사항에 따라 이 설정을 구성하고 인스턴스 요금을 고려해야 합니다. 자세한 설명은 Amazon EC2 요금을 참조하세요. 서비스 액세스 섹션에서 세부 정보 보기를 선택하여 서비스 초기화 중에 생성된 서비스 연결 역할과 추가 정책을 검토합니다. 다음을 선택합니다. 	AWS 관리자

작업	설명	필요한 기술
<p>볼륨 및 보안 그룹을 구성합니다.</p>	<ol style="list-style-type: none"> 1. 볼륨 및 보안 그룹 창에서 복제 서버의 EBS 볼륨 유형을 선택하고 Amazon EBS 암호화를 기본값으로 설정합니다. 2. Elastic Disaster Recovery가 기본 보안 그룹을 자동으로 연결하고 모니터링하도록 하려면 항상 AWS Elastic Disaster Recovery 보안 그룹 사용을 선택합니다. 3. 다음을 선택합니다. 	<p>AWS 관리자</p>

작업	설명	필요한 기술
추가 설정을 구성합니다.	<ol style="list-style-type: none"> 추가 설정 창에서 데이터 라우팅 및 제한, PIT 정책, 태그를 구성합니다. <ul style="list-style-type: none"> 데이터 라우팅 및 제한은 외부 서버에서 복제 서버로 데이터가 전달되는 방식을 제어합니다. 데이터 복제에 프라이빗 IP 사용을 선택합니다. 그렇지 않으면 복제 서버에 자동으로 퍼블릭 IP가 할당되며 데이터는 공용 인터넷을 통해 전달됩니다. 특정 시점(PIT) 정책 섹션에서 스냅샷이 필요하지 않은 기간을 결정하는 보존 정책을 구성합니다. 기본 보존 기간은 7일입니다. 태그 섹션에서 AWS 계정의 Elastic Disaster Recovery에서 생성한 리소스에 사용자 지정 태그를 추가합니다. 다음을 선택하고 다음 창의 설정을 검토한 다음 기본값 생성을 선택하여 기본 템플릿을 생성합니다. 	AWS 관리자

AWS Replication Agent 설치

작업	설명	필요한 기술
IAM 역할을 생성합니다.	AWSElasticDisasterRecoveryAgentInstallationPolicy 정책을 포함하는 IAM 역할을 생성합니다. AWS 액세스 유형 선택 섹션에서 프로그래밍 방식 액세스를 활성화합니다. 액세스 키 ID 및 비밀 액세스 키를 기록합니다. AWS Replication Agent를 설치하는 동안 해당 정보가 필요합니다.	AWS 관리자
요구 사항을 확인합니다.	Elastic Disaster Recovery 설명서에서 AWS Replication Agent 설치를 위한 사전 조건 을 확인하고 완료합니다.	AWS 관리자
AWS Replication Agent를 설치합니다.	<p>운영 체제의 설치 지침을 따라 AWS Replication Agent를 설치합니다.</p> <ul style="list-style-type: none"> • Microsoft Windows의 경우: 설치 파일을 다운로드하고 관리자 권한으로 .exe 파일을 실행합니다. 프롬프트에 응답하여 설치를 완료합니다. • Linux의 경우: 다음 명령(표시된 순서대로)을 복사하여 Secure Shell(SSh) 세션에 붙여넣습니다. 첫 번째 명령은 설치 프로그램을 다운로드 	AWS 관리자

작업	설명	필요한 기술
	<p>드하고 두 번째 명령은 설치 프로그램을 실행합니다.</p> <pre data-bbox="625 331 1029 730">wget -O ./aws-replication-installer-init.py https://aws-elastic-disaster-recovery-us-west-2.s3.amazonaws.com/latest/linux/aws-replication-installer-init.py</pre> <p>참고: 리전을 반영하도록 URL을 변경하세요.</p> <pre data-bbox="625 884 1029 1045">sudo python3 aws-replication-installer-init.py</pre> <p>프롬프트에 응답하여 설치를 완료합니다.</p> <p>나머지 서버에 대해 이 단계를 반복합니다.</p>	

작업	설명	필요한 기술
복제를 모니터링합니다.	<p>Elastic Disaster Recovery 소스 서버 창으로 돌아가 복제 상태를 모니터링합니다. 데이터 전송 크기에 따라 초기 동기화에 다소 시간이 걸립니다.</p> <p>소스 서버가 완전히 동기화되면 서버 상태가 준비 완료로 업데이트됩니다. 즉, 스테이징 영역에 복제 서버가 생성되었으며 EBS 볼륨이 소스 서버에서 스테이징 영역으로 복제되었음을 의미합니다.</p>	AWS 관리자

시작 설정 구성

작업	설명	필요한 기술
시작 설정을 편집합니다.	<p>드릴 및 복구 인스턴스의 시작 설정을 업데이트하려면 Elastic Disaster Recovery 콘솔에서 소스 서버를 선택한 다음 작업, 시작 설정 편집을 선택합니다. 또는 소스 서버 페이지에서 복제하는 소스 시스템을 선택한 다음 시작 설정 탭을 선택합니다. 이 탭에는 일반 시작 설정과 EC2 시작 템플릿이라는 두 개의 섹션이 있습니다.</p>	AWS 관리자
일반 시작 설정을 구성합니다.	<p>요구 사항에 따라 일반 시작 설정을 수정합니다.</p> <ul style="list-style-type: none"> 올바른 인스턴스 유형 크기 조정: 기본을 선택하면 	AWS 관리자

작업	설명	필요한 기술
	<p>Elastic Disaster Recovery는 Amazon EC2 시작 템플릿에서 선택한 인스턴스 유형을 건너뛰고 소스 서버의 운영 체제, CPU 및 RAM을 기반으로 인스턴스 유형을 자동으로 선택합니다.</p> <ul style="list-style-type: none"> • 프라이빗 IP 복사: 드릴 또는 복구 인스턴스에서 사용하는 프라이빗 IP가 소스 서버에서 사용되는 프라이빗 IP와 일치하는지 확인하기 위해 Elastic Disaster Recovery를 사용할지 여부를 선택합니다. 예를 선택한 경우 Amazon EC2 시작 템플릿에서 설정한 서브넷의 IP 범위에 프라이빗 IP 주소가 포함되는지 확인합니다. <p>자세한 내용은 Elastic Disaster Recovery 설명서의 일반 시작 설정을 참조하세요.</p>	

작업	설명	필요한 기술
Amazon EC2 시작 템플릿을 구성합니다.	<p>Elastic Disaster Recovery는 Amazon EC2 시작 템플릿을 사용하여 각 소스 서버에 대한 드릴 및 복구 인스턴스를 시작합니다. 시작 템플릿은 AWS Replication Agent를 설치한 후 Elastic Disaster Recovery에 추가하는 각 소스 서버에 대해 자동으로 생성됩니다.</p> <p>Amazon EC2 시작 템플릿을 Elastic Disaster Recovery와 함께 사용하려는 경우 기본 시작 템플릿으로 설정해야 합니다.</p> <p>자세한 내용은 Elastic Disaster Recovery 설명서의 EC2 시작 템플릿을 참조하세요.</p>	AWS 관리자

DR 드릴 및 장애 조치 시작

작업	설명	필요한 기술
드릴 시작	<ol style="list-style-type: none"> Elastic Disaster Recovery 콘솔에서 소스 서버 페이지를 열고 소스 서버의 상태가 준비 완료인지 확인합니다. DR 드릴을 수행할 소스 서버를 모두 선택합니다. 복구 작업 시작 메뉴에서 드릴 시작을 선택하고 적절한 스냅샷을 선택합니다. point-in-time 선택한 소스 서버에 대해 복구 작업이 시작됩니 	AWS 관리자

작업	설명	필요한 기술
	<p>다. 복구 작업 기록 탭에서 작업 상태를 모니터링할 수 있습니다.</p> <p>참고: 소스 서버의 추가 변경 사항은 드릴 인스턴스가 아닌 복제 서버에 동기화됩니다.</p> <p>시작된 드릴 인스턴스는 복구 인스턴스 페이지에도 표시됩니다.</p> <ol style="list-style-type: none"> 4. DR 드릴 인스턴스를 테스트하고 확인합니다. 5. 복구 인스턴스 페이지에서 드릴 인스턴스를 선택한 다음 작업, AWS에서 연결 해제를 선택합니다. 이렇게 하면 복구 인스턴스에서 AWS Replication Agent가 삭제되고 Elastic Disaster Recovery에서 복구 인스턴스와 관련된 모든 리소스가 제거됩니다. 6. 복구 인스턴스 삭제를 선택합니다. 이렇게 하면 Elastic Disaster Recovery 콘솔에서 해당 인스턴스의 표현이 삭제되고 Elastic Disaster Recovery 서비스와 인스턴스의 연결이 완전히 해제됩니다. 기본 EC2 인스턴스는 삭제되지 않습니다. 	

작업	설명	필요한 기술
	<p>7. Amazon EC2 콘솔에서 DR 드릴 인스턴스를 종료합니다.</p> <p>자세한 내용은 Elastic Disaster Recovery 설명서의 장애 조치 준비를 참조하세요.</p>	
<p>드릴을 검증합니다.</p>	<p>이전 단계에서는 DR 리전에서 새 대상 인스턴스를 시작했습니다. 대상 인스턴스는 시작 시 생성된 스냅샷을 기반으로 하는 소스 서버의 복제본입니다.</p> <p>이 절차에서는 Amazon EC2 대상 시스템에 연결하여 예상대로 실행되고 있는지 확인합니다.</p> <ol style="list-style-type: none"> 1. Amazon EC2 콘솔을 엽니다. 2. 인스턴스(실행 중)를 선택합니다. 3. 대상 인스턴스를 선택하고 프라이빗 IPv4 주소를 기록합니다. 4. EC2 인스턴스에 연결할 수 있고 JD Edwards EnterpriseOne 및 관련 구성 요소가 예상대로 복제되었는지 확인하십시오. 	

작업	설명	필요한 기술
장애 조치를 시작합니다.	<p>장애 조치는 기본 시스템에서 보조 시스템으로 트래픽을 리디렉션하는 것입니다. Elastic Disaster Recovery를 사용하면 AWS에서 복구 인스턴스를 시작하여 장애 조치를 수행할 수 있습니다. 복구 인스턴스가 시작되면 기본 시스템의 트래픽을 해당 인스턴스로 리디렉션합니다.</p> <ol style="list-style-type: none"> 1. Elastic Disaster Recovery 콘솔에서 소스 서버 페이지를 열고 소스 서버의 복구 준비 완료 옆에 준비 완료 표시되고 데이터 복제 상태 옆에 정상이 표시되는지 확인합니다. 2. 소스 서버를 선택합니다. 복구 작업 시작 메뉴에서 복구 시작을 선택합니다. 3. 복구 인스턴스를 시작할 point-in-time 스냅샷을 선택한 다음 복구 시작을 선택합니다. <p>복구 작업이 시작됩니다. 복구 인스턴스 페이지에서 작업 상태를 모니터링할 수 있습니다.</p> <ol style="list-style-type: none"> 4. 복구 인스턴스를 테스트하고 확인합니다. 필요한 경우 DNS 구성을 조정하고 JD Edwards EnterpriseOne 애 	AWS 관리자

작업	설명	필요한 기술
	<p>플리케이션을 데이터베이스에 연결합니다.</p> <p>5. 모든 변경 사항이 새 복구 인스턴스에 기록되었으므로 이제 원본 JD Edwards EnterpriseOne 서버의 연결을 끊고 서비스 해제할 수 있습니다.</p> <p>6. AWS Replication Agent 설치 에픽에 설명된 절차에 따라 복구 인스턴스를 DR 리전의 소스 서버로 등록합니다.</p> <p>자세한 내용은 Elastic Disaster Recovery 설명서의 장애 조치 수행을 참조하세요.</p>	

작업	설명	필요한 기술
<p>페일백을 시작합니다.</p>	<p>페일백 시작 프로세스는 장애 조치 시작 프로세스와 비슷합니다.</p> <ol style="list-style-type: none"> 1. 기본 리전에서 Elastic Disaster Recovery 콘솔을 엽니다. 복구 인스턴스 페이지로 이동하여 드릴 인스턴스를 선택한 다음 작업, AWS에서 연결 해제, 복구 인스턴스 삭제를 선택합니다. 2. DR 리전에서 Elastic Disaster Recovery 콘솔을 엽니다. AWS 복제 에이전트를 설치하여 새 JD Edwards EnterpriseOne 서버를 DR 지역의 소스 서버로 등록하십시오. 데이터는 새 스테이징 서브넷에 프로비저닝된 새 복제 서버와 동기화됩니다. <p>참고: 새 JD Edwards EnterpriseOne 서버를 원본 서버로 등록하면 Elastic Disaster Recovery 콘솔에 두 개의 원본 서버가 표시될 수 있습니다. 하나는 기본 EC2 인스턴스에서 생성된 서버이고 다른 하나는 복구 인스턴스에서 생성된 새 서버입니다. 서버에 올바르게 태그를 지정하여 혼동을 방지하고 가급적이면 시작 템</p>	<p>AWS 관리자</p>

작업	설명	필요한 기술
	<p>플릿에 새 서버를 추가하는 것이 좋습니다.</p> <p>3. 기본 리전에서 DR 복제를 다시 시작하려면 DR 리전의 Elastic Disaster Recovery 콘솔에서 시작된 복구 인스턴스의 연결을 해제하고 호스트를 기본 리전의 소스 서버로 등록합니다.</p> <p>자세한 내용은 Elastic Disaster Recovery 설명서의 페일백 수행을 참조하세요.</p>	

작업	설명	필요한 기술
<p>JD Edwards 구성 요소를 시작합니다. EnterpriseOne</p>	<ol style="list-style-type: none"> 1. 데이터베이스 서버에 로그인하여 JD Edwards EnterpriseOne 데이터베이스를 시작합니다. 2. 데이터베이스가 실행되면 JD Edwards EnterpriseOne 로직 및 배치 서버를 시작합니다. 3. 웹 WebLogic 서버에서 시작하고 JAS 서버에서 JAS 인스턴스를 시작합니다. 4. 프로비전 서버와 SM 콘솔용 WebLogic 서버에서 시작합니다. 5. 서버에서 SM Agent를 시작합니다. 6. JD Edwards 로그인 EnterpriseOne 제대로 작동하는지 확인하십시오. <p>JD Edwards EnterpriseOne 링크가 작동하려면 Route 53 및 Application Load Balancer의 변경 사항을 통합해야 합니다.</p> <p>Lambda, Step Functions 및 Systems Manager(Run Command)를 사용하여 이 단계를 자동화할 수 있습니다.</p> <p>참고: Elastic Disaster Recovery는 운영 체제 및 파일 시스템을 호스팅하는 소스 EC2 인스턴스 EBS 볼륨의 블</p>	<p>JD 에드워즈 CNC EnterpriseOne</p>

작업	설명	필요한 기술
	<p>록 수준 복제를 수행합니다. Amazon EFS를 사용하여 생성된 공유 파일 시스템은 이 복제에 포함되지 않습니다. 첫 번째 에픽에서 언급한 것처럼 DataSync AWS를 사용하여 공유 파일 시스템을 DR 지역에 복제한 다음 이러한 복제된 파일 시스템을 DR 시스템에 마운트할 수 있습니다.</p>	

문제 해결

문제	Solution
<p>소스 서버 데이터 복제 상태가 멈춤이며 복제가 지연됩니다. 세부 정보를 확인하면 데이터 복제 상태에 에이전트가 표시되지 않음이 표시됩니다.</p>	<p>멈춘 소스 서버가 실행 중인지 확인합니다.</p> <p>참고: 소스 서버가 다운되면 복제 서버가 자동으로 종료됩니다.</p> <p>지연 문제에 대한 자세한 내용은 Elastic Disaster Recovery 설명서의 복제 지연 문제를 참조하세요.</p>
<p>디스크를 스캔한 후 RHEL 8.2에서 소스 EC2 인스턴스에 AWS Replication Agent를 설치할 수 없습니다. <code>aws_replication_agent_installer.log</code> 가 커널 헤더가 누락되었음을 나타냅니다.</p>	<p>RHEL 8, CentOS 8 또는 Oracle Linux 8에 AWS Replication Agent를 설치하기 전에 다음을 실행합니다.</p> <pre data-bbox="836 1539 1507 1654">sudo yum install elfutils-libelf-devel</pre> <p>자세한 내용은 Elastic Disaster Recovery 설명서의 Linux 설치 요구 사항을 참조하세요.</p>

문제	Solution
<p>Elastic Disaster Recovery 콘솔에서 소스 서버가 지연이 있는 준비 완료로 표시되고 데이터 복제 상태가 멈춤으로 표시됩니다.</p> <p>AWS Replication Agent를 사용할 수 없는 기간에 따라 상태가 지연이 심하다고 표시될 수 있지만 문제는 동일합니다.</p>	<p>운영 체제 명령을 사용하여 AWS Replication Agent가 소스 EC2 인스턴스에서 실행되고 있는지 확인하거나 인스턴스가 실행 중인지 확인합니다.</p> <p>문제를 수정하면 Elastic Disaster Recovery가 스캔을 다시 시작합니다. DR 드릴을 시작하기 전에 모든 데이터가 동기화되고 복제 상태가 정상일 때까지 기다리세요.</p>
<p>초기 복제 지연이 심합니다. Elastic Disaster Recovery 콘솔에서 소스 서버의 초기 동기화 상태가 매우 느린 것을 확인할 수 있습니다.</p>	<p>Elastic Disaster Recovery 설명서의 복제 지연 문제 섹션에 나와 있는 복제 지연 문제를 확인합니다.</p> <p>내장 컴퓨팅 작업으로 인해 복제 서버가 부하를 처리하지 못할 수 있습니다. 이 경우 AWS 기술 지원팀에 문의한 후 인스턴스 유형을 업그레이드해보세요.</p>

관련 리소스

- [AWS Elastic Disaster Recovery 사용 설명서](#)
- [Creating a scalable disaster recovery plan with AWS Elastic Disaster Recovery](#)(AWS 블로그 게시물)
- [AWS Elastic Disaster Recovery-A Technical Introduction](#)(AWS Skill Builder 과정, 로그인 필요)
- [AWS Elastic Disaster Recovery 빠른 시작 설명서](#)

SAP Pacemaker 클러스터를 ENSA1에서 ENSA2 클러스터로 업그레이드

작성자: Gergely Cserdi(AWS) 및 Balazs Sandor Skublics(AWS)

환경: 프로덕션	소스: ENSA1 기반 Pacemaker 클러스터	대상: ENSA2 기반 Pacemaker 클러스터
R 유형: 리아키텍트	워크로드: SAP	기술: 인프라, 현대화
AWS 서비스: Amazon EC2		

요약

이 패턴은 Standalone Enqueue Server(ENSA1)를 기반으로 하는 SAP Pacemaker 클러스터를 ENSA2 버전으로 업그레이드하기 위한 단계 및 고려 사항을 설명합니다. 이 패턴의 정보는 SUSE Linux Enterprise Server (SLES) 및 Red Hat Enterprise Linux(RHEL) 운영 체제에 모두 적용됩니다.

SAP NetWeaver 7.52 또는 S/4HANA 1709 및 이전 버전의 페이스메이커 클러스터는 ENSA1 아키텍처에서 실행되며 ENSA1 전용으로 구성되어 있습니다. Amazon Web Services(AWS)에서 SAP 워크로드를 실행하고 있는데 ENSA2로의 전환에 관심이 있다면 SAP, SUSE 및 RHEL 설명서가 포괄적인 정보를 제공하지 않을 수 있습니다. 이 패턴은 SAP 파라미터와 Pacemaker 클러스터를 재구성하여 ENSA1 버전에서 ENSA2 버전으로 업그레이드하는 데 필요한 기술 단계를 설명합니다. SUSE 시스템의 예를 제공하지만 RHEL 클러스터의 개념은 동일합니다.

참고: ENSA1 및 ENSA2 는 SAP 애플리케이션에만 적용되는 개념이므로 이 패턴의 정보는 SAP HANA 또는 다른 유형의 클러스터에는 적용되지 않습니다.

엄밀히 따지자면 ENSA2 는 Enager Replicator 2와 함께 또는 사용하지 않고 사용할 수 있습니다. 그러나 고가용성(HA) 및 (클러스터 솔루션을 통한) 장애 조치 자동화를 위해서는 Enqueue Replicator 2가 필요합니다. 이 패턴은 ENSA2 클러스터라는 용어를 사용하여 Standalone Enqueue Server 2와 Enqueue Replicator 2가 있는 클러스터를 나타냅니다..

사전 조건 및 제한 사항

사전 조건

- SLES 또는 RHEL에서 Pacemaker와 Corosync를 사용하는 작동 중인 EnSA1 기반 클러스터입니다.
- (ABAP) SAP Central Services(ASCS/SCS) 및 Enqueue Replication Server(ERS) 인스턴스가 실행되는 Amazon Elastic Compute Cloud(Amazon EC2) 인스턴스가 두 개 이상 있어야 합니다.
- SAP 애플리케이션 및 클러스터 관리에 대한 지식.
- 루트 사용자로 Linux 환경에 액세스할 수 있습니다.

제한 사항

- EnSA1 기반 클러스터는 2노드 아키텍처만 지원합니다.
- 7.52 이전의 SAP 버전에는 EnSA2 기반 클러스터를 배포할 수 없습니다. NetWeaver
- 클러스터의 EC2 인스턴스는 서로 다른 AWS 가용 영역에 있어야 합니다.

제품 버전

- SAP 버전 7.52 이상 NetWeaver
- S/4HANA 2020부터 ENSA2 클러스터만 지원
- Kernel 7.53 이상, ENSA2 및 Enqueue Replicator 2 지원
- SAP 애플리케이션용 SLES 버전 12 이상
- HA(고가용성) 버전 7.9 이상을 지원하는 SAP용 RHEL

아키텍처

소스 기술 스택

- SAP NetWeaver 7.52 (SAP 커널 7.53 이상 포함)
- SLES 또는 RHEL 운영 체제

대상 기술 스택

- SAP NetWeaver 7.52 (SAP 커널 7.53 이상) (ABAP 플랫폼을 사용하는 S/4HANA 2020 포함)
- SLES 또는 RHEL 운영 체제

대상 아키텍처

다음 다이어그램은 ENSA2 클러스터를 기반으로 하는 ASCS/SCS 및 ERS 인스턴스의 HA 구성을 보여줍니다.

ENSA1 및 ENSA2 클러스터의 비교

SAP는 ENSA1 후속 모델로 ENSA2 제품을 소개했습니다. ENSA1 기반 클러스터는 오류 발생 시 ASCS/SCS 인스턴스가 ERS로 장애 조치되는 2노드 아키텍처를 지원합니다. 이 제한은 ASCS/SCS 인스턴스가 장애 조치 후 ERS 노드의 공유 메모리에서 잠금 테이블 정보를 다시 얻는 방식에서 비롯됩니다. Enqueue Replicator 2를 사용하는 EnSA2 기반 클러스터는 ASCS/SCS 인스턴스가 네트워크를 통해 ERS 인스턴스로부터 잠금 정보를 수집할 수 있기 때문에 이러한 제한을 없애줍니다. ASCS/SCS 인스턴스가 더 이상 ERS 노드로 장애 조치를 할 필요가 없으므로 ENSA2 기반 클러스터에는 두 개 이상의 노드가 있을 수 있습니다. (하지만 2노드 ENSA2 클러스터 환경에서는 클러스터에 장애 조치할 다른 노드가 없기 때문에 ASCS/SCS 인스턴스는 여전히 ERS 노드로 장애 조치됩니다.) ENSA2 버전은 SAP 커널 7.50부터 지원되지만 몇 가지 제한이 있습니다. [대기열에 넣기 리플리케이터 2를 지원하는 HA 설정의 경우 최소 요구 사항은 NetWeaver 7.52입니다 \(SAP OSS Note 2630416 참조\)](#). S/4HANA 1809에는 기본적으로 ENSA2 아키텍처가 권장되는 반면, S/4HANA는 2020 버전부터 ENSA2 아키텍처만 지원합니다.

자동화 및 규모 조정

대상 아키텍처의 HA 클러스터는 ASCS가 다른 노드로 자동으로 장애 조치가 되도록 합니다.

ENSA2 기반 클러스터로 전환하기 위한 시나리오

EnSA2 기반 클러스터로 업그레이드하기 위한 두 가지 주요 시나리오는 다음과 같습니다.

- 시나리오 1: SAP 릴리스와 커널 버전이 ENSA2 지원을 한다고 가정하면 함께 제공되는 SAP 업그레이드 또는 S/4HANA 변환 없이 ENSA2 버전으로 업그레이드하도록 선택합니다.
- 시나리오 2: 업그레이드 또는 변환(예: S/4HANA 1809 이상)의 일환으로 SUM을 사용하여 ENSA2 버전으로 이동합니다.

예픽 섹션에서는 이 두 시나리오의 단계를 다룹니다. 첫 번째 시나리오에서는 ENSA2 클러스터 구성을 변경하기 전에 SAP 관련 파라미터를 수동으로 설정해야 합니다. 두 번째 시나리오에서는 바이너리와 SAP 관련 파라미터가 SUM을 통해 배포되며 남은 작업은 HA에 대한 클러스터 구성을 업데이트하는 것뿐입니다. SUM을 사용한 후에도 SAP 파라미터의 유효성을 검사하는 것이 좋습니다. 대부분의 경우 S/4HANA 변환이 클러스터 업그레이드의 주요 원인입니다.

도구

- OS 패키지 관리자의 경우 Zypper(SLES용) 또는 YUM(RHEL용) 도구를 사용하는 것이 좋습니다.
- 클러스터 관리의 경우 crm(SLES용) 또는 pcs(RHEL용) 셸을 사용하는 것이 좋습니다.
- SAPControl과 같은 SAP 인스턴스 관리 도구.
- (선택 사항) S/4HANA 변환 업그레이드를 위한 SUM 도구.

모범 사례

- AWS에서 SAP 워크로드를 사용하는 모범 사례는 AWS Well-Architected Framework를 위한 [SAP Lens](#)를 참조하십시오.
- ENSA2 다중 노드 아키텍처의 클러스터 노드 수(홀수 또는 짝수)를 고려해 보십시오.
- SAP S/4-HA-CLU 1.0 인증 표준에 따라 SLES 15용 ENSA2 클러스터를 설정하십시오.
- ENSA2 버전으로 업그레이드하기 전에 항상 기존 클러스터 및 애플리케이션 상태를 저장하거나 백업하십시오.

에픽

ENSA2 관련 SAP 파라미터를 수동으로 구성(시나리오 1만 해당)

작업	설명	필요한 기술
기본 프로파일에서 파라미터를 구성합니다.	<p>동일한 SAP 릴리스를 유지하면서 ENSA2 버전으로 업그레이드하거나 대상 릴리스의 기본값이 ENSA1인 경우 기본 프로파일(DEFAULT.PFL 파일)의 파라미터를 다음 값으로 설정하십시오.</p> <pre> enq/enable=TRUE enq/serverhost=sapas csvirt enq/serverinst=10 (instance number of ASCS/SCS instance) </pre>	SAP

작업	설명	필요한 기술
	<pre data-bbox="609 210 1015 541">enqueue/process_location=REMOTESA enq/replicatorhost=sapersvirt enq/replicatorinst=11 (instance number of ERS instance)</pre> <p data-bbox="592 577 1031 856">여기서 <code>sapascsvirt</code> 는 ASCS 인스턴스의 가상 호스트 이름, <code>sapersvirt</code> 는 ERS 인스턴스의 가상 호스트 이름입니다. 이를 대상 환경에 맞게 변경할 수 있습니다.</p> <p data-bbox="592 892 1031 1123">참고: 이 업그레이드 옵션을 사용하려면 SAP 릴리스 및 커널 버전에서 ENSA2 및 Enqueue Replicator 2를 지원해야 합니다.</p>	

작업	설명	필요한 기술
<p>ASCS/SCS 인스턴스 프로파일을 구성하십시오.</p>	<p>동일한 SAP 릴리스를 유지하면서 ENSA2 버전으로 업그레이드하거나 대상 릴리스의 기본값이 ENSA1인 경우 ASCS/SCS 인스턴스 프로파일에서 다음 파라미터를 설정하십시오.</p> <p>프로파일에 ENSA1이 정의된 섹션은 다음과 같습니다.</p> <pre data-bbox="594 709 1027 1585"> #----- ----- ----- ----- Start SAP enqueue server #----- ----- ----- _EN = en.sap\$(S APSYSTEMNAME)\$(INST ANCE_NAME) Execute_04 = local rm - f \$_EN Execute_05 = local ln - s -f \$(DIR_EXECUTABLE)/ enserver\$(FT_EXE) \$_EN Start_Program_01 = local \$_EN pf=\$_PF </pre> <p>이 섹션을 ENSA2용으로 재구성하려면:</p> <ol style="list-style-type: none"> 1. <code>_EN</code> 프로그램 접두사를 SAP의 최신 정보를 기반으 	<p>SAP</p>

작업	설명	필요한 기술
	<p>로 <code>_ENQ</code>로 변경합니다(OSS Note 2501860, SAP ONE Support Launchpad 사용자 계정 필요).</p> <ol style="list-style-type: none"> 인큐 서버의 바이너리를 <code>enserver</code>에서 <code>enq_server</code> 로 변경합니다. 새로운 <code>enq/server/replication/enable</code> 파라미터를 <code>TRUE</code>로 설정합니다. <code>Autostart = 0</code>이 확실한지 확인합니다. <p>변경 후 이 프로파일 섹션은 다음과 같아야 합니다.</p> <pre> #----- ----- ----- ----- Start SAP enqueue server #----- ----- ----- ----- _ENQ = enq.sap\$(SAPSYSTEMNAME)\$(INST STANCE_NAME) Execute_04 = local rm - f \$_ENQ) Execute_05 = local ln - s -f \$(DIR_EXECUTABLE)/ enq_server\$(FT_EXE) \$_ENQ) </pre>	

작업	설명	필요한 기술
	<pre data-bbox="597 205 1024 506">Start_Program_01 = local \$_ENQ pf= \$_PF ... enq/server/replic ation/enable = TRUE Autostart = 0</pre> <p data-bbox="597 541 1024 1010">중요: _ENQ의 경우 재시작 옵션이 활성화되어 있지 않아야 합니다. RestartProgram_01 로 설정된 _ENQ의 경우 StartProgram_01 로 변경하십시오. 이렇게 하면 SAP가 서비스를 다시 시작하거나 클러스터 관리형 리소스를 방해하는 것을 방지할 수 있습니다.</p>	

작업	설명	필요한 기술
ERS 프로파일을 구성합니다.	<p>동일한 SAP 릴리스를 유지하면서 ENSA2 버전으로 업그레이드하거나 대상 릴리스의 기본값이 ENSA1인 경우 ERS 인스턴스 프로파일에서 다음 파라미터를 설정하십시오.</p> <p>인큐 복제기가 정의된 섹션을 찾으십시오. 출력은 다음과 비슷합니다.</p> <pre data-bbox="592 709 1027 1585"> #----- ----- ----- Start enqueue replicati on server #----- ----- ----- _ER = er.sap\$(S APSYSTEMNAME)\$(INS TANCE_NAME) Execute_03 = local rm - f \$_ER Execute_04 = local ln - s -f \$(DIR_EXECUTABLE)/ enrepserver\$(FT_EXE) \$_ER Start_Program_00 = local \$_ER pf=\$_PF) NR=\$(SCSID) </pre> <p>Enqueue Replicator 2에 맞게 이 섹션 재구성:</p> <ol style="list-style-type: none"> 1. <code>_ER</code> 프로그램 접두사를 SAP의 최신 메모를 기반 	SAP

작업	설명	필요한 기술
	<p>으로 _ENQR로 변경합니다 (OSS Note 2501860, SAP ONE Support Launchpad 사용자 계정 필요).</p> <ol style="list-style-type: none"> 인큐 복제기의 바이너리를 enq_replicator 대신 enrepserver 로 변경하십시오. Autostart = 0이 확실한지 확인합니다. <p>변경 후 이 프로파일 섹션은 다음과 같아야 합니다.</p> <pre data-bbox="597 892 1026 1793"> #----- ----- ----- Start enqueue replicati on server #----- ----- ----- _ENQR = enqr.sap\$ (SAPSYSTEMNAME)\$(I NSTANCE_NAME) Execute_01 = local rm - f \$_ENQR Execute_02 = local ln - s -f \$(DIR_EXECUTABLE)/ enq_replicator\$(FT _EXE) \$_ENQR Start_Program_00 = local \$_ENQR pf= \$_PF) NR=\$(SCSID) ... Autostart = 0 </pre>	

작업	설명	필요한 기술
	<p>중요: _ENQR의 경우 재시작 옵션이 활성화되어 있지 않아야 합니다. RestartProgram_01 로 설정된 _ENQR의 경우 StartProgram_01 로 변경하십시오. 이렇게 하면 SAP가 서비스를 다시 시작하거나 클러스터 관리형 서비스를 방해하는 것을 방지할 수 있습니다.</p>	
<p>SAP Start 서비스를 다시 시작하십시오.</p>	<p>이 에픽에서 앞서 설명한 프로파일을 변경한 후에는 ASCS/SCS 및 ERS 모두에 대해 SAP Start 서비스를 다시 시작하십시오.</p> <pre> sapcontrol -nr 10 - function RestartService SCT sapcontrol -nr 11 - function RestartService SCT </pre> <p>여기서 SCT는 SAP 시스템 ID를 참조하며, 각각 10과 11이 ASCS/SCS 및 ERS 인스턴스의 인스턴스 번호라고 가정합니다.</p>	<p>SAP</p>

클러스터를 ENSA2 재구성(두 시나리오 모두에 필요)

작업	설명	필요한 기술
<p>SAP 리소스 에이전트에서 버전 번호를 확인합니다.</p>	<p>SUM을 사용하여 SAP를 S/4HANA 1809 이상으로 업그레이드하는 경우 SUM은 SAP 프로파일의 파라미터 변경을 처리합니다. 클러스터만 수동 조정이 필요합니다. 하지만 클러스터를 변경하기 전에 파라미터 설정을 확인하는 것이 좋습니다.</p> <p>참고: 이 예픽의 예는 SUSE 운영 체제를 사용하고 있다고 가정합니다. RHEL을 사용하는 경우 Zypper 및 crm 대신 YUM 및 pcs 셸과 같은 도구를 사용해야 합니다.</p> <p>아키텍처의 두 노드를 모두 검사하여 resource-agents 패키지가 SAP에서 권장하는 최소 버전과 일치하는지 확인하십시오. SLES에 대해서는 SAP OSS 노트 2641019를 참조하십시오. RHEL에 대해서는 SAP OSS 노트 2641322를 참조하십시오. (SAP 노트는 SAP ONE Support Launchpad 사용자 계정이 필요합니다.)</p> <pre data-bbox="592 1663 1031 1873">sapers:sctadm 23> zypper search -s -i resource-agents Loading repository data...</pre>	<p>AWS 시스템 관리자</p>

작업	설명	필요한 기술
	<pre> Reading installed packages... S Name Type Version Arch Repository --+-+----- ----+-----+--- ----- -----+--- -----+----- ----- i resource-agents package 4.8.0+git 30.d0077df0-150300 .8.28.1 x86_64 SLE-Product-HA15-SP3- Updates </pre> <p>필요한 경우 resource-agents 버전을 업데이트하십시오.</p>	
<p>클러스터 구성을 백업합니다.</p>	<p>CRM 클러스터 구성을 다음과 같이 백업합니다.</p> <pre> crm configure show > / tmp/cluster_config_backup.txt </pre>	<p>AWS 시스템 관리자</p>
<p>유지 관리 모드를 설정합니다.</p>	<p>클러스터를 유지 관리 모드로 설정합니다.</p> <pre> crm configure property maintenance-mode=" true" </pre>	<p>AWS 시스템 관리자</p>

작업	설명	필요한 기술
클러스터 구성을 확인합니다.	<p>현재 클러스터 구성을 확인합니다.</p> <pre>crm configure show</pre> <p>다음은 전체 출력에서 발췌한 내용입니다.</p> <pre>node 1: sapascs node 2: sapers ... primitive rsc_sap_S CT_ASCS10 SAPInstance \ operations \$id=rsc_s ap_SCT_ASCS10-oper ations \ op monitor interval=120 timeout=60 on-fail=r estart \ params InstanceN ame=SCT_ASCS10_sap ascsvirt START_PRO FILE="/sapmnt/SCT/ profile/SCT_ASCS10 _sapascsvirt" \ AUTOMATIC_RECOVER= false \ meta resource-stickines s=5000 failure-t imeout=60 migration- threshold=1 priority= 10 primitive rsc_sap_S CT_ERS11 SAPInstance \ operations \$id=rsc_s ap_SCT_ERS11-opera tions \ op monitor interval=120 timeout=60 on-fail=r estart \</pre>	AWS 시스템 관리자

작업	설명	필요한 기술
	<pre> params InstanceName=SCT_ERS11_sapersvirt START_PROFILE="/sapmnt/SCT/profile/SCT_ERS11_sapersvirt" \ AUTOMATIC_RECOVER=false IS_ERS=true \ meta priority=1000 ... colocation col_sap_S CT_no_both -5000: grp_SCT_ERS11 grp_SCT_ASCS10 location loc_sap_S CT_failover_to_ers rsc_sap_SCT_ASCS10 \ rule 2000: runs_ers_SCT eq 1 order ord_sap_S CT_first_start_asc Optional: rsc_sap_S CT_ASCS10:start rsc_sap_SCT_ERS11: stop symmetrical=false ... </pre> <p>여기서 <code>sapascsvirt</code> 는 ASCS 인스턴스의 가상 호스트 이름, <code>sapersvirt</code> 는 ERS 인스턴스의 가상 호스트 이름, <code>SCT</code>는 SAP 시스템 ID를 나타냅니다.</p>	

작업	설명	필요한 기술
<p>장애 조치 콜로케이션 제약을 제거합니다.</p>	<p>이전 예제에서 위치 제약 조건 <code>loc_sap_SCT_failover_to_ers</code> 는 ASCS의 ENSA1 기능이 장애 조치 시 항상 ERS 인스턴스를 따르도록 지정했습니다. ENSA2 사용 시 ASCS는 모든 참여 노드로 자유롭게 장애 조치할 수 있어야 하므로 이 제약을 없앨 수 있습니다.</p> <pre>crm configure delete loc_sap_SCT_failover_to_ers</pre>	<p>AWS 시스템 관리자</p>

작업	설명	필요한 기술
<p>프리미티브를 조정합니다.</p>	<p>또한 ASCS 및 ERS SAP 인스턴스 프리미티브를 약간 변경해야 합니다.</p> <p>다음은 ENSA1 용으로 구성된 ASCS SAP 인스턴스 프리미티브의 예입니다.</p> <pre data-bbox="597 569 1027 1486"> primitive rsc_sap_SCT_ASCS10 SAPInstance \ operations \$id=rsc_sap_SCT_ASCS10-operations \ op monitor interval=120 timeout=60 on-fail=r estart \ params InstanceName=SCT_ASCS10_sapascsvirt START_PROFILE="/sapmnt/SCT/profile/SCT_ASCS10_sapascsvirt" \ AUTOMATIC_RECOVER=false \ meta resource-stickiness=5000 failure-timeout=60 migration-threshold=1 priority=10 </pre> <p>ENSA2 버전으로 업그레이드하려면 이 구성을 다음과 같이 변경하십시오.</p> <pre data-bbox="597 1696 1027 1822"> primitive rsc_sap_SCT_ASCS10 SAPInstance \ </pre>	<p>AWS 시스템 관리자</p>

작업	설명	필요한 기술
	<pre>operations \$id=rsc_sap_SCT_ASCS10-operations \ op monitor interval=120 timeout=60 on-fail=r estart \ params InstanceName=SCT_ASCS10_sapascsvirt START_PRO FILE="/sapmnt/SCT/profile/SCT_ASCS10_sapascsvirt" \ AUTOMATIC_RECOVER=false \ meta resource-stickiness=3000</pre> <p>이것은 ENSA1 용으로 구성된 ERS SAP 인스턴스 프리미티브의 예입니다.</p> <pre>primitive rsc_sap_SCT_ERS11 SAPInstance \ operations \$id=rsc_sap_SCT_ERS11-operations \ op monitor interval=120 timeout=60 on-fail=r estart \ params InstanceName=SCT_ERS11_sapersvirt START_PRO FILE="/sapmnt/SCT/profile/SCT_ERS11_sapersvirt" \ AUTOMATIC_RECOVER=false IS_ERS=true \ meta priority=1000</pre>	

작업	설명	필요한 기술
	<p>ENSA2 버전으로 업그레이드하려면 이 구성을 다음과 같이 변경하십시오.</p> <pre data-bbox="597 380 1024 1052">primitive rsc_sap_SCT_ERS11 SAPInstance \ operations \$id=rsc_sap_SCT_ERS11-operations \ op monitor interval=120 timeout=60 on-fail=r estart \ params InstanceName=SCT_ERS11_sapersvirt START_PROFILE="/sapmnt/SCT/profile/SCT_ERS11_sapersvirt" \ AUTOMATIC_RECOVER=false IS_ERS=true</pre> <p>다양한 방법으로 프리미티브를 변경할 수 있습니다. 예를 들어 다음 예제와 같이 vi와 같은 편집기에서 수정할 수 있습니다.</p> <pre data-bbox="597 1318 938 1398">crm configure edit rsc_sap_SCT_ERS11</pre>	

작업	설명	필요한 기술
<p>유지 관리 모드를 비활성화합니다.</p>	<p>클러스터에서 유지 관리 모드를 비활성화합니다.</p> <pre>crm configure property maintenance-mode="false"</pre> <p>클러스터가 유지 관리 모드를 벗어나면 새 ENSA2 설정을 사용하여 ASCS 및 ERS 인스턴스를 온라인 상태로 전환하려고 시도합니다.</p>	<p>AWS 시스템 관리자</p>

(선택 사항) 클러스터 노드 추가

작업	설명	필요한 기술
<p>모범 사례를 검토합니다.</p>	<p>노드를 더 추가하기 전에 출수 또는 짝수 노드 사용 여부와 같은 모범 사례를 이해해야 합니다.</p>	<p>AWS 시스템 관리자</p>
<p>노드를 추가합니다.</p>	<p>노드를 더 추가하려면 운영 체제를 업데이트하고, 기존 노드에 맞는 소프트웨어 패키지를 설치하며, 마운트를 사용할 수 있게 하는 등의 일련의 작업이 필요합니다. SAP Software Provisioning Manager(SWPM)의 추가 호스트 준비 옵션을 사용하여 호스트의 SAP별 기준을 생성할 수 있습니다. 자세한 정보는 다음 섹션의 SAP 가이드를 참조하십시오.</p>	<p>AWS 시스템 관리자</p>

관련 리소스

SAP 및 SUSE 참조

SAP 노트에 액세스하려면 SAP ONE Support Launchpad 사용자 계정이 있어야 합니다. 자세한 내용은 [SAP 지원 웹사이트](#)를 참조하십시오.

- [SAP 노트 2501860 – ABAP 7.52용 SAP 애플리케이션 서버 설명서 NetWeaver](#)
- [SAP 노트 2641019 – SUSE HA 환경에 ENSA2 설치 및 ENSA1 버전에서 ENSA2 버전으로 업데이트](#)
- [SAP 노트 2641322 – SAP용 Red HA 솔루션 사용 시 ENSA2 설치 및 ENSA1 에서 ENSA2로 업데이트](#)
- [SAP 노트 2711036 – HA 환경에서의 독립형 Enqueue Server 2 사용](#)
- [독립형 Enqueue Server 2\(SAP 설명서\)](#)
- [SAP S/4 HANA – Enqueue Replication 2 고가용성 클러스터 - 설정 가이드\(SUSE 설명서\)](#)

AWS 참조

- [AWS의 SAP HANA: SLES 및 RHEL을 위한 고가용성 구성 가이드](#)
- [SAP Lens - AWS Well-Architected Framework](#)

여러 AWS 계정의 VPC에서 일관된 가용 영역 사용

작성자: Adam Spicer(AWS)

코드 리포지토리: 다중 계정 가용 영역 매핑	환경: 프로덕션	기술: 인프라
AWS 서비스: AWS CloudFormation, 아마존 VPC, AWS Lambda		

요약

Amazon Web Services(AWS) 클라우드에서 가용 영역은 AWS 계정마다 다를 수 있는 이름과 위치를 식별하는 [가용 영역 ID\(AZ ID\)](#)를 가질 수 있습니다. CloudFormation AWS를 사용하여 가상 사설 클라우드 (VPC) 를 생성하는 경우 서브넷을 생성할 때 가용 영역의 이름 또는 ID를 지정해야 합니다. 여러 계정에서 VPC를 생성하는 경우 가용 영역 이름이 무작위로 지정되므로 서브넷은 각 계정에서 서로 다른 가용 영역을 사용합니다.

계정 전체에서 동일한 가용 영역을 사용하려면 각 계정의 가용 영역 이름을 동일한 AZ ID에 매핑해야 합니다. 예를 들어, 다음 다이어그램은 use1-az6 AZ ID가 AWS 계정 A에서 us-east-1a, AWS 계정 Z에서 us-east-1c로 이름이 지정되었음을 보여줍니다.

이 패턴은 서브넷에서 동일한 가용 영역을 사용할 수 있는 확장 가능한 계정 간 솔루션을 제공하여 영역 일관성을 보장하는 데 도움이 됩니다. 영역 일관성을 보장하면 계정 간 네트워크 트래픽이 가용 영역 간 네트워크 경로를 피하므로 데이터 전송 비용을 줄이고 워크로드 간 네트워크 지연 시간을 줄일 수 있습니다.

이 패턴은 AWS CloudFormation [AvailabilityZoneId 속성에](#) 대한 대안적인 접근 방식입니다.

사전 조건 및 제한 사항

사전 조건

- 동일한 AWS 리전에 2개 이상의 활성 AWS 계정이 있어야 합니다.
- 해당 리전의 VPC 요구 사항을 지원하는 데 필요한 가용 영역 수를 평가하십시오.

- 지원해야 하는 각 가용 영역의 AZ ID를 식별하고 기록하십시오. 이에 대한 자세한 내용은 AWS Resource Access Manager 설명서에서 [AWS 리소스의 가용 영역 ID](#)를 참조하십시오.
- 쉼표로 구분하여 정렬된 AZ ID 목록입니다. 예를 들어 목록의 첫 번째 가용 영역은 az1로 매핑되고 두 번째 가용 영역은 az2로 매핑되며 이 매핑 구조는 쉼표로 구분된 목록이 완전히 매핑될 때까지 계속됩니다. 매핑할 수 있는 최대 AZ ID 수는 없습니다.
- GitHub [다중 계정 가용 영역 매핑](#) 리포지토리의 az-mapping.yaml 파일이 로컬 시스템에 복사되었습니다.

아키텍처

다음 다이어그램은 계정에 배포되고 AWS Systems Manager Parameter Store 값을 생성하는 아키텍처를 보여줍니다. 이 파라미터 스토어 값은 계정에서 VPC를 생성할 때 사용됩니다.

이 다이어그램은 다음 워크플로를 보여줍니다.

1. 이 패턴의 솔루션은 VPC의 영역 일관성이 필요한 모든 계정에 배포됩니다.
2. 솔루션은 각 AZ ID에 대한 파라미터 스토어 값을 생성하고 새 가용 영역 이름을 저장합니다.
3. AWS CloudFormation 템플릿은 각 Parameter Store 값에 저장된 가용 영역 이름을 사용하므로 영역 일관성이 보장됩니다.

다음 다이어그램은 이 패턴의 솔루션으로 VPC를 생성하는 워크플로를 보여줍니다.

이 다이어그램은 다음 워크플로를 보여줍니다.

1. VPC를 생성하기 위한 템플릿을 AWS에 제출하십시오. CloudFormation
2. AWS는 CloudFormation 각 가용 영역의 파라미터 스토어 값을 확인하고 각 AZ ID의 가용 영역 이름을 반환합니다.
3. VPC는 영역 일관성에 필요한 올바른 AZ ID로 생성됩니다.

이 패턴의 솔루션을 배포한 후에는 파라미터 스토어 값을 참조하는 서브넷을 생성할 수 있습니다. CloudFormationAWS를 사용하는 경우 다음 YAML 형식의 샘플 코드에서 가용 영역 매핑 파라미터 값을 참조할 수 있습니다.

```
Resources:
  PrivateSubnet1AZ1:
    Type: AWS::EC2::Subnet
    Properties:
      VpcId: !Ref VPC
      CidrBlock: !Ref PrivateSubnetAZ1CIDR
      AvailabilityZone:
        !Join
          - ''
          - - '{{resolve:ssm:/az-mapping/az1:1}}'
```

이 샘플 코드는 GitHub [다중 계정](#) 가용 영역 매핑 리포지토리의 `vpc-example.yaml` 파일에 포함되어 있습니다. 영역 일관성을 위해 파라미터 스토어 값에 맞춰 VPC와 서브넷을 생성하는 방법을 보여줍니다.

기술 스택

- AWS CloudFormation
- AWS Lambda
- AWS Systems Manager Parameter Store

자동화 및 규모 조정

AWS CloudFormation StackSets 또는 AWS Control Tower용 사용자 지정 솔루션을 사용하여 이 패턴을 모든 AWS 계정에 배포할 수 있습니다. 자세한 내용은 [AWS 클라우드포메이션 설명서의 CloudFormation StackSets와 함께 작업하기 및 AWS 솔루션 라이브러리의 AWS Control Tower에 대한 사용자 지정](#)을 참조하십시오.

AWS CloudFormation 템플릿을 배포한 후 Parameter Store 값을 사용하도록 업데이트하고 파이프라인 또는 요구 사항에 따라 VPC를 배포할 수 있습니다.

도구

서비스

- [AWS는 AWS CloudFormation](#) 리소스를 모델링 및 설정하고, 빠르고 일관되게 프로비저닝하고, 수명 주기 전반에 걸쳐 관리할 수 있도록 지원합니다. 템플릿을 사용하여 리소스와 해당 종속성을 설명하고, 리소스를 개별적으로 관리하는 대신 스택으로 함께 시작하고 구성할 수 있습니다. 여러 AWS 계정 및 AWS 리전에서 스택을 관리하고 프로비저닝할 수 있습니다.

- [AWS Lambda](#)는 서버를 프로비저닝하거나 관리하지 않고도 코드를 실행할 수 있게 해주는 컴퓨팅 서비스입니다. Lambda는 필요 시에만 코드를 실행하며, 일일 몇 개의 요청에서 초당 수천 개의 요청까지 자동으로 규모를 조정합니다. 사용한 컴퓨팅 시간만큼만 비용을 지불하고, 코드가 실행되지 않을 때는 요금이 부과되지 않습니다.
- [AWS Systems Manager Parameter Store](#)는 AWS Systems Manager의 기능입니다. 구성 데이터 관리 및 암호 관리를 위한 안전한 계층적 스토리지를 제공합니다.

코드

이 패턴의 코드는 GitHub [다중 계정 가용 영역 매핑](#) 리포지토리에서 제공됩니다.

에픽

az-mapping.yaml 파일 배포

작업	설명	필요한 기술
리전에 필요한 가용 영역을 결정합니다.	<ol style="list-style-type: none"> 1. 해당 리전에서 일관되게 사용해야 하는 AZ ID를 결정합니다. 2. 쉼표로 구분된 목록에 AZ ID를 적용하려는 순서대로 AZ ID를 기록합니다. 예를 들어 목록의 첫 번째 가용 영역은 az1로 매핑되고 두 번째 가용 영역은 az2로 매핑됩니다. 매핑할 수 있는 최대 AZ ID 수는 없습니다. 	클라우드 아키텍트
az-mapping.yaml 파일을 배포합니다.	<p>az-mapping.yaml 파일을 사용하여 필요한 모든 AWS 계정에 AWS CloudFormation 스택을 생성합니다. AZIDs 파라미터에는 앞서 생성한 쉼표로 구분된 목록을 사용하십시오.</p> <p>AWS CloudFormation StackSets 또는 AWS Control</p>	클라우드 아키텍트

작업	설명	필요한 기술
	Tower 솔루션을 위한 사용자 지정 솔루션을 사용하는 것이 좋습니다.	

계정에 VPC 배포

작업	설명	필요한 기술
AWS CloudFormation 템플릿을 사용자 지정합니다.	CloudFormationAWS를 사용하여 서브넷을 생성할 때는 이전에 생성한 Parameter Store 값을 사용하도록 템플릿을 사용자 지정하십시오. 샘플 템플릿은 GitHub 다중 계정 가용 영역 매핑 리포지토리 의 vpc-example.yaml 파일을 참조하십시오.	클라우드 아키텍트
VPC를 배포합니다.	사용자 지정된 AWS CloudFormation 템플릿을 계정에 배포하십시오. 그러면 해당 리전의 각 VPC는 서브넷에 사용되는 가용 영역에서 영역 일관성을 유지합니다.	클라우드 아키텍트

관련 리소스

- [AWS 리소스의 가용 영역 ID](#)(AWS Resource Access Manager 설명서)
- [AWS::EC2::Subnet](#)(AWS CloudFormation 설명서)

Account Factory에 대한 테라폼(AFT) 코드를 로컬에서 검증합니다.

작성자: Alexandru Pop(AWS), Michal Gorniak(AWS)

환경: 프로덕션

기술: 인프라, 현대화 DevOps,
소프트웨어 개발 및 테스트

워크로드: 오픈 소스

AWS 서비스: AWS Control
Tower

요약

이 패턴은 AWS Control Tower Account Factory for HashiCorp Terraform (AFT) 에서 관리하는 Terraform 코드를 로컬에서 테스트하는 방법을 보여줍니다. Terraform은 코드를 사용하여 클라우드 인프라 및 리소스를 프로비저닝하고 관리하는 데 도움이 되는 오픈 소스 코드형 인프라(IaC) 도구입니다. AFT는 AWS Control Tower에서 여러 AWS 계정을 프로비저닝하고 사용자 지정하는 데 도움이 되는 Terraform 파이프라인을 설정합니다.

코드 개발 중에는 AFT 파이프라인 외부에서 로컬에서 코드형 인프라(IaC)를 테스트하는 것이 유용할 수 있습니다. 이 패턴은 다음 작업을 수행하는 방법을 보여줍니다.

- AFT 관리 계정의 AWS CodeCommit 리포지토리에 저장된 Terraform 코드의 로컬 사본을 검색하십시오.
- 검색된 코드를 사용하여 AFT 파이프라인을 로컬에서 시뮬레이션합니다.

이 프로시저를 사용하여 일반 AFT 파이프라인에 속하지 않는 Terraform 명령을 실행할 수도 있습니다. 예를 들어 이 메서드를 사용하여, `terraform validate`, `terraform plan terraform destroy`, `terraform import` 같은 명령을 실행할 수 있습니다.

사전 조건 및 제한 사항

사전 조건

- [AWS Control Tower](#)를 사용하는 활성 AWS 다중 계정 환경
- 완전히 배포된 [AFT 환경](#)
- AWS Command Line Interface(AWS CLI), [설치](#) 및 [구성됨](#)

- 설치 및 구성된 [Code Commit용 AWS CLI 보안 인증 도우미](#)
- Python 3.x
- 로컬 머신에 설치 및 구성된 [Git](#)
- git-remote-commit [유틸리티, 설치 및 구성](#)
- 설치 및 구성된 [Terraform](#)(로컬 Terraform 패키지 버전은 AFT 배포에 사용되는 버전과 일치해야 함)

제한 사항

- 이 패턴은 AWS Control Tower, AFT 또는 특정 Terraform 모듈에 필요한 배포 단계를 다루지 않습니다.
- 이 절차 중에 로컬에서 생성된 출력은 AFT 파이프라인 런타임 로그에 저장되지 않습니다.

아키텍처

대상 기술 스택

- AWS Control Tower 배포 내에 배포된 AFT 인프라
- Terraform
- Git
- CLI version 2

자동화 및 규모 조정

이 패턴은 단일 AFT 관리 AWS 계정에서 AFT 글로벌 계정 사용자 지정을 위한 Terraform 코드를 로컬로 호출하는 방법을 보여줍니다. Terraform 코드의 유효성이 검사되면 다중 계정 환경의 나머지 계정에 적용할 수 있습니다. 자세한 내용은 AWS Control Tower 설명서의 [사용자 지정 재호출](#)을 참조하세요.

유사한 프로세스를 사용하여 로컬 터미널에서 AFT 계정 사용자 지정을 실행할 수도 있습니다.

AFT 계정 사용자 지정에서 Terraform 코드를 로컬로 호출하려면 AFT 관리 계정에서 aft-account-customizations CodeCommit 리포지토리 대신 aft-global-account-customizations 리포지토리를 복제하십시오.

도구

서비스

- [AWS Control Tower](#)는 권장 모범 사례에 따라 AWS 다중 계정 환경을 설정하고 관리할 수 있도록 지원합니다.
- [AWS Command Line Interface\(AWS CLI\)](#)는 명령줄 셸에서 명령을 사용하여 AWS 서비스와 상호 작용할 수 있는 오픈 소스 도구입니다.

기타 서비스

- [HashiCorp Terraform](#)은 코드를 사용하여 클라우드 인프라 및 리소스를 프로비저닝하고 관리하는 데 도움이 되는 오픈 소스 IaC (코드형 인프라) 도구입니다.
- [Git](#)은 오픈 소스 분산 버전 제어 시스템입니다.

code

다음 사항은 AFT에서 관리하는 Terraform 코드를 로컬에서 실행하는 데 사용할 수 있는 예제 bash 스크립트입니다. 스크립트를 사용하려면 이 패턴의 에픽 섹션에 있는 지침을 따르세요.

```
#!/bin/bash
# Version: 1.1 2022-06-24 Unsetting AWS_PROFILE since, when set, it interferes with
script operation
#           1.0 2022-02-02 Initial Version
#
# Purpose: For use with AFT: This script runs the local copy of TF code as if it were
running within AFT pipeline.
#           * Facilitates testing of what the AFT pipeline will do
#           * Provides the ability to run terraform with custom arguments (like 'plan'
or 'move') which are currently not supported within the pipeline.
#
# © 2021 Amazon Web Services, Inc. or its affiliates. All Rights Reserved.
# This AWS Content is provided subject to the terms of the AWS Customer Agreement
# available at http://aws.amazon.com/agreement or other written agreement between
# Customer and either Amazon Web Services, Inc. or Amazon Web Services EMEA SARL or
both.
#
# Note: Arguments to this script are passed directly to 'terraform' without parsing nor
validation by this script.
#
# Prerequisites:
#   1. local copy of ct GIT repositories
#   2. local backend.tf and aft-providers.tf filled with data for the target account
on which terraform is to be run
```

```
# Hint: The contents of above files can be obtain from the logs of a previous
# execution of the AFT pipeline for the target account.
# 3. 'terraform' binary is available in local PATH
# 4. Recommended: .gitignore file containing 'backend.tf', 'aft_providers.tf' so the
# local copy of these files are not pushed back to git

readonly credentials=$(aws sts assume-role \
  --role-arn arn:aws:iam::$(aws sts get-caller-identity --query "Account" --output
  text ):role/AWSAFTAdmin \
  --role-session-name AWSAFT-Session \
  --query Credentials )

unset AWS_PROFILE
export AWS_ACCESS_KEY_ID=$(echo $credentials | jq -r '.AccessKeyId')
export AWS_SECRET_ACCESS_KEY=$(echo $credentials | jq -r '.SecretAccessKey')
export AWS_SESSION_TOKEN=$(echo $credentials | jq -r '.SessionToken')
terraform "$@"
```

에픽

예제 코드를 로컬 파일로 저장합니다.

작업	설명	필요한 기술
예제 코드를 로컬 파일로 저장합니다.	<ol style="list-style-type: none"> 이 패턴의 코드 섹션에 있는 예제 bash 스크립트를 복사하여 코드 편집기에 붙여넣습니다. 파일 이름을 <code>ct_terraform.sh</code> 로 지정합니다. 그런 다음 파일을 전용 폴더 (예: <code>~/scripts</code> 또는 <code>~/bin</code>)에 로컬로 저장합니다. 	AWS 관리자
예제 코드를 실행 가능하게 만드세요.	<p>터미널 창을 열고 다음 중 하나를 수행하여 AWS AFT 관리 계정을 인증합니다.</p> <ul style="list-style-type: none"> AFT 관리 계정에 액세스하는데 필요한 권한으로 구 	AWS 관리자

작업	설명	필요한 기술
	<p>성된 기존 AWS CLI 프로파일을 사용합니다. 프로파일을 사용하려면 다음 명령을 실행하시면 됩니다.</p> <pre>export AWS_PROFILE=<aft account profile name></pre> <ul style="list-style-type: none"> 조직에서 SSO를 사용하여 AWS에 액세스하려는 경우 조직의 SSO 페이지에서 AFT 관리 계정의 보안 인증 정보를 입력하세요. <p>참고: 조직에 AWS 환경에 인증 보안 인증 정보를 제공하는 사용자 지정 도구가 있을 수도 있습니다.</p>	

작업	설명	필요한 기술
<p>올바른 AWS 리전의 AFT 관리 계정에 대한 액세스 권한을 검증합니다.</p>	<p>중요: AFT 관리 계정으로 인증한 것과 동일한 터미널 세션을 사용해야 합니다.</p> <ol style="list-style-type: none"> 다음 명령을 실행하여 AFT 배포의 AWS 리전으로 이동합니다. <pre data-bbox="630 569 1029 688">export AWS_REGION N=<aft_region></pre> 다음과 같은 작업을 수행하여 계정을 올바르게 사용하고 있는지 확인합니다. <ul style="list-style-type: none"> 다음 명령을 실행합니다: <pre data-bbox="630 930 1029 1050">aws code-commit list-repositories</pre> 그런 다음 출력에 나열된 리포지토리가 AFT 관리 계정에 있는 리포지토리의 이름과 일치하는지 확인합니다. 	AWS 관리자
<p>AFT 리포지토리 코드를 저장할 새 로컬 디렉터리를 생성합니다.</p>	<p>동일한 터미널 세션에서 다음 명령을 실행합니다.</p> <pre data-bbox="597 1476 1029 1591">mkdir my_aft cd my_aft</pre>	AWS 관리자

작업	설명	필요한 기술
원격 AFT 리포지토리 코드를 복제합니다.	<p>1. 로컬 터미널에서 다음 명령을 실행합니다.</p> <pre>git clone codecommit:::\$AWS_REGION://aft-global-customizations</pre> <p>참고: 단순화를 위해 이 프로시저와 AFT는 기본 코드 브랜치만 사용합니다. 코드 브랜치를 사용하려면 여기에 코드 브랜칭 명령을 입력할 수 있습니다. 하지만 기본 브랜치가 아닌 브랜치에서 적용된 모든 변경 사항은 AFT 자동화가 기본 브랜치의 코드를 적용할 때 롤백됩니다.</p> <p>2. 그런 다음에는 다음과 같은 명령을 실행하여 복제된 디렉터리로 이동합니다.</p> <pre>cd aft-global-customizations/terraform</pre>	AWS 관리자

AFT 파이프라인이 로컬에서 실행되는 데 필요한 Terraform 구성 파일을 생성합니다.

작업	설명	필요한 기술
이전에 실행한 AFT 파이프라인을 열고 Terraform 구성 파일을 로컬 폴더에 복사합니다.	<p>참고: AFT 파이프라인이 로컬에서 실행되려면 이 에픽에서 생성된 backend.tf 및 aft-providers.tf 구성 파일이 필요합니</p>	AWS 관리자

작업	설명	필요한 기술
	<p>다. 이러한 파일은 클라우드 기반 AFT 파이프라인 내에서 자동으로 생성되지만, 파이프라인을 로컬에서 실행하려면 수동으로 생성해야 합니다. AFT 파이프라인을 로컬에서 실행하려면 단일 AWS 계정 내에서 파이프라인 실행을 나타내는 파일 세트 하나가 필요합니다.</p> <ol style="list-style-type: none"> 1. AWS Control Tower 관리 계정 보안 인증 정보를 사용하여 AWS Management Console에 로그인합니다. 그런 다음 AWS CodePipeline 콘솔을 엽니다. AFT를 배포한 동일한 AWS 리전에 있는지 확인합니다. 2. 왼쪽 탐색 창에서 파이프라인을 클릭합니다. 3. #####-customizations-pipeline을 선택합니다. (#####은 Terraform 코드를 로컬에서 실행하는 데 사용하는 AWS 계정 ID입니다). 4. 가장 최근 실행 표시에 성공 값이 표시되는지 확인합니다. 값이 다른 경우 AFT 파이프라인에서 사용자 지정을 다시 호출해야 합니다. 자세한 내용은 AWS Control Tower 설명서의 사용자 지정 재호출을 참조하세요. 	

작업	설명	필요한 기술
	<p>5. 세부 정보를 불러오려면 최신 런타임을 선택하세요.</p> <p>6. Apply-AFT-Global-Customizations 섹션에서 Apply-Terraform 단계를 찾습니다.</p> <p>7. Terraform 적용 단계의 세부 정보 섹션을 선택합니다.</p> <p>8. Terraform 적용 스테이지의 런타임 로그를 찾을 수 있습니다.</p> <p>9. 런타임 로그에서 “\n\naft-providers.tf ... \n\nbackend.tf” 줄로 시작하고 끝나는 섹션을 찾아보세요.</p> <p>10. 이 두 레이블 사이의 출력을 복사하고 로컬 Terraform 폴더(터미널 세션의 현재 작업 디렉터리) 내에 aft-providers.tf 라는 이름이 지정된 로컬 파일로 저장합니다.</p> <p>자동 생성 공급자.tf 문 예시</p> <pre data-bbox="630 1367 1029 1814"> ## Autogenerated providers.tf ## ## Updated on: 2022-05-31 16:27:45 ## provider "aws" { region = "us-east-2" assume_role { role_arn = "arn:aws:iam:##### </pre>	

작업	설명	필요한 기술
	<pre data-bbox="646 212 977 625">#####:role/AWSA FTExecution" } default_tags { tags = { managed_by = "AFT" } } }</pre> <p data-bbox="592 640 1015 819">11.런타임 로그에서 “\n\n tf ... “\n \n backup.tf” 줄로 시작 하고 끝나는 섹션을 찾아보 세요.</p> <p data-bbox="592 840 1015 1113">12.이 두 레이블 사이의 출력을 복사하고 로컬 Terraform 폴 더(터미널 세션의 현재 작업 디렉터리) 내에 tf라는 이름 이 지정된 로컬 파일로 저장 합니다.</p> <p data-bbox="592 1186 1015 1228">자동 생성된 백엔드.tf 문 예제</p> <pre data-bbox="609 1281 998 1827">## Autogenerated backend.tf ## ## Updated on: 2022-05-3 1 16:27:45 ## terraform { required_version = ">= 0.15.0" backend "s3" { region = "us-east-2" bucket = "aft-backend-##### #####-primary-re gion"</pre>	

작업	설명	필요한 기술
	<pre> key = "#####-aft- global-customizati ons/terraform.tfst ate" dynamodb_table = "aft-backend-#### #####" encrypt = "true" kms_key_id = "cbdc21d6-e04d-4c3 7-854f-51e199cfcb7c" kms_key_id = "#####-####-####- ####-#####" role_arn = "arn:aws:iam::#### #####:role/AWS AFTExecution" } } </pre> <p>참고: backend.tf 및 aft-providers.tf 파일은 특정 AWS 계정, AFT 배포 및 폴더에 연결되어 있습니다. 이러한 파일은 동일한 AFT 배포 내의 aft-global-customizations 리포지토리와 aft-account-customizations 리포지토리에 있는지에 따라 달라집니다. 동일한 런타임 목록에서 두 파일을 모두 생성해야 합니다.</p>	

예제 bash 스크립트를 사용하여 AFT 파이프라인을 로컬에서 실행합니다.

작업	설명	필요한 기술
<p>검증하려는 Terraform 구성 변경 사항을 구현합니다.</p>	<ol style="list-style-type: none"> 다음 명령을 실행하여 복제된 <code>aft-global-customizations</code> 리포지토리로 이동합니다. <div data-bbox="630 541 1029 667" style="border: 1px solid #ccc; border-radius: 10px; padding: 10px; margin: 10px 0;"> <pre>cd aft-global-customizations/terraform</pre> </div> <p>참고: 파일은 <code>backend.tf</code> 및 디렉터리의 <code>aft-providers.tf</code> 에 있습니다. 디렉터리에는 리포지토리의 Terraform 파일도 포함되어 있습니다. <code>aft-global-customizations</code></p> 로컬에서 테스트하려는 Terraform 코드 변경 사항을 구성 파일에 통합합니다. 	AWS 관리자
<p><code>ct_terraform.sh</code> 스크립트를 실행하고 출력을 검토합니다.</p>	<ol style="list-style-type: none"> <code>sh</code> 스크립트가 들어 있는 로컬 폴더로 이동합니다. 수정된 Terraform 코드의 유효성을 검사하려면 다음 명령을 실행하여 <code>ct_terraform.sh</code> 스크립트를 실행합니다. <div data-bbox="630 1591 1029 1717" style="border: 1px solid #ccc; border-radius: 10px; padding: 10px; margin: 10px 0;"> <pre>~/scripts/ct_terraform.sh apply</pre> </div> <p>참고 : 이 단계에서 모든 Terraform 명령을 실행할 수 있습니다. Terraform 명령 전</p> 	AWS 관리자

작업	설명	필요한 기술
	<p>체 목록을 보려면 다음 명령을 실행하세요.</p> <pre>terraform --help</pre> <p>3. 명령 출력을 확인합니다. 그런 다음 코드 변경 사항을 로컬에서 디버깅한 다음 변경 사항을 커밋하여 AFT 리포지토리로 다시 푸시합니다.</p> <p>중요:</p> <ul style="list-style-type: none"> 로컬에서 변경한 후 원격 리포지토리로 푸시하지 않은 모든 변경 사항은 일시적이며 AFT 파이프라인 자동화를 실행하여 언제든지 취소할 수 있습니다. AFT 자동화는 다른 사용자와 AFT 자동화 트리거가 호출할 수 있으므로 언제든지 실행할 수 있습니다. AFT는 항상 리포지토리의 기본 브랜치에서 코드를 적용하여 커밋되지 않은 변경 사항을 취소합니다. 	

로컬 코드 변경 사항을 커밋하고 AFT 리포지토리로 다시 푸시합니다.

작업	설명	필요한 기술
<p>backend.tf 및 aft-providers.tf 파일에 대한 참조를 .gitignore 파일에 추가합니다.</p>	<p>다음 명령을 실행하여 만든 backend.tf 및 aft-providers.tf 파일을 .gitignore 파일에 추가합니다.</p> <pre data-bbox="594 594 1027 793"> echo backend.tf >> .gitignore echo aft-providers.tf >>.gitignore </pre> <p>참고: 파일을 파일로 이동하면 .gitignore 파일이 커밋되어 원격 AFT 리포지토리로 푸시되지 않습니다.</p>	<p>AWS 관리자</p>
<p>코드 변경 사항을 커밋하고 원격 AFT 리포지토리에 푸시합니다.</p>	<ol style="list-style-type: none"> 1. 새 Terraform 구성 파일을 리포지토리에 추가하려면 다음 명령을 실행하세요. <pre data-bbox="630 1224 1027 1304"> git add <filename> </pre> 2. 변경 사항을 적용하고 CodeCommit AWS의 원격 AFT 리포지토리로 푸시하려면 다음 명령을 실행합니다. <pre data-bbox="630 1535 1027 1654"> git commit -a git push </pre> <p>중요: 이 시점까지 이 절차에 따라 도입한 코드 변경 사항은 하</p>	<p>AWS 관리자</p>

작업	설명	필요한 기술
	나의 AWS 계정에만 적용됩니다.	

AFT에서 관리하는 여러 계정에 변경 사항을 적용하세요.

작업	설명	필요한 기술
AFT에서 관리하는 모든 계정에 변경 사항을 적용하세요.	AFT에서 관리하는 여러 AWS 계정에 변경 사항을 적용하려면 AWS Control Tower 설명서의 사용자 지정 재호출 에 있는 지침을 따르세요.	AWS 관리자

패턴 더 보기

- [읽기 전용 복제본을 사용하여 Amazon RDS 커스텀 기반 PeopleSoft 오라클에 HA를 추가합니다.](#)
- [AWS Systems Manager를 사용하여 Windows 레지스트리 항목의 추가 또는 업데이트 자동화](#)
- [AWS 리소스 평가 자동화](#)
- [AWS CDK를 사용하여 AWS Service Catalog 포트폴리오 및 제품 배포 자동화](#)
- [Amazon MQ에서 RabbitMQ 구성의 자동화](#)
- [AWS 계정 전반에 걸쳐 Amazon RDS 인스턴스의 복제를 자동화](#)
- [Cloud Custodian 및 AWS CDK를 사용하여 Systems Manager용 AWS 관리형 정책을 EC2 인스턴스 프로파일에 자동으로 연결](#)
- [AWS CDK를 사용하여 마이크로서비스용 CI/CD 파이프라인 및 Amazon ECS 클러스터 자동으로 구축](#)
- [변경 사항을 자동으로 감지하고 단일 저장소에 대한 다양한 CodePipeline 파이프라인을 시작합니다. CodeCommit](#)
- [AWS Config에서 사용자 지정 수정 규칙을 CloudTrail 사용하여 AWS를 자동으로 다시 활성화합니다.](#)
- [AWS DataOps 개발 키트를 사용하여 Google 애널리틱스 데이터를 수집, 변환 및 분석할 수 있는 데이터 파이프라인을 구축하십시오.](#)
- [Amazon EC2 Auto Scaling 및 Systems Manager를 사용하여 Micro Focus Enterprise Server PAC 구축하기](#)
- [GitHub 액션과 테라폼을 사용하여 Docker 이미지를 빌드하고 Amazon ECR로 푸시합니다.](#)
- [Terraform을 사용하여 AWS Organizations의 IAM 액세스 키 관리를 중앙 집중화하십시오.](#)
- [Terraform을 사용하여 AWS Organizations에서 소프트웨어 패키지 배포를 중앙 집중화합니다.](#)
- [서버리스 접근 방식을 사용하여 AWS 서비스를 함께 연결](#)
- [하이브리드 연결 모드를 사용하여 AWS의 VMware Cloud로의 데이터 센터 확장 구성](#)
- [AWS 기반 SQL Server의 Always On 가용성 그룹에서 읽기 전용 라우팅 구성](#)
- [AWS의 VMware Cloud에서 VM을 프로비저닝하도록 VMware vRealize Automation을 구성합니다.](#)
- [Java 및 Python 프로젝트를 위한 동적 CI 파이프라인을 자동으로 생성](#)
- [AWS의 VMware Cloud를 사용하여 AWS에 VMware SDDC 배포](#)
- [프라이빗 엔드포인트와 Application Load Balancer 사용하여 내부 웹 사이트에 Amazon API Gateway API 배포](#)
- [Amazon EKS 클러스터의 배포 및 디버깅](#)

- [AWS CDK 및 AWS를 사용하여 AWS 컨트롤 타워 컨트롤을 배포하고 관리합니다. CloudFormation](#)
- [Terraform을 사용하여 AWS Control Tower 컨트롤을 배포하고 관리합니다.](#)
- [테라폼을 사용하여 CloudWatch Synthetics 카나리아를 배포하세요](#)
- [Terraform을 사용하여 AWS WAF의 보안 자동화 솔루션 배포](#)
- [AWS 랜딩 존 설계를 문서화하십시오](#)
- [IAM 프로파일이 EC2 인스턴스와 연결되었는지 확인](#)
- [AWS Organizations의 조직 전체에서 AWS Backup 보고서를 CSV 파일로 내보내기](#)
- [Amazon Personalize를 사용하여 개인화되고 순위가 다시 매겨진 추천 생성](#)
- [Amazon Data Firehose 리소스가 AWS KMS 키로 암호화되지 않은 경우 식별 및 알림](#)
- [부트스트랩 파이프라인을 사용하여 테라폼용 Account Factory \(AFT\) 구현](#)
- [쿠버네티스를 사용하여 Amazon EKS 작업자 노드에 SSM 에이전트를 설치합니다. DaemonSet](#)
- [다음을 사용하여 Amazon EKS 작업자 노드에 SSM CloudWatch 에이전트 및 에이전트를 설치합니다. preBootstrapCommands](#)
- [VMware vRealize 네트워크 인사이트를 VMware Cloud on AWS와 통합](#)
- [여러 AWS 계정 및 AWS 리전의 AWS Service Catalog 제품을 관리](#)
- [AWS CDK로 Amazon ECS Anywhere를 설정하여 온프레미스 컨테이너 애플리케이션을 관리](#)
- [DNS 레코드를 Amazon Route 53 프라이빗 호스팅 영역으로 대량 마이그레이션합니다.](#)
- [Oracle E-Business Suite를 Amazon RDS Custom으로 마이그레이션](#)
- [PeopleSoft 오라클을 Amazon RDS 커스텀으로 마이그레이션](#)
- [AWS MGN을 사용하여 RHEL BYOL 시스템을 AWS 라이선스가 포함된 인스턴스로 마이그레이션하기](#)
- [VMware HCX를 사용하여 VMware SDDC를 AWS의 VMware Cloud로 마이그레이션](#)
- [Amazon ElastiCache 클러스터의 미사용 암호화 모니터링](#)
- [보안 그룹의 ElastiCache 클러스터 모니터링](#)
- [AWS 서비스를 사용하여 SAP RHEL 페이스메이커 클러스터를 모니터링합니다.](#)
- [다중 VPC에서 중앙 AWS 서비스 엔드포인트에 비공개로 액세스](#)
- [컨테이너를 다시 시작하지 않고 데이터베이스 보안 인증 교체](#)
- [IAM 사용자 생성 시 알림 전송](#)
- [VMware Aria Operations for Logs를 사용하여 VMware Cloud on AWS에서 스포링크로 로그를 전송합니다.](#)

- [AWS CDK를 사용하여 Amazon ECS Anywhere에서 하이브리드 워크로드를 위한 CI/CD 파이프라인을 설정하고 GitLab](#)
- [AWS에서 고가용성 PeopleSoft 아키텍처 설정](#)
- [NICE EnginFrame 및 NICE DCV 세션 관리자를 사용하여 자동 스케일링 가상 데스크톱 인프라 \(VDI\) 를 설정합니다.](#)
- [활성 대기 데이터베이스를 사용하여 Amazon RDS Custom에서 Oracle E-Business Suite를 위한 HA/DR 아키텍처를 설정합니다.](#)
- [다중 지역, 다중 계정 조직에서 AWS CloudFormation 드리프트 감지 설정](#)
- [Amazon FSx를 사용하여 SQL Server Always On FCI용 다중 AZ 인프라 설정](#)
- [Aurora PostgreSQL 호환에서 Oracle UTL_FILE 기능 설정](#)
- [AWS 프라이빗 CA와 AWS RAM을 사용하여 프라이빗 인증서 관리를 간소화합니다.](#)
- [AWS Organizations를 사용하여 Transit Gateway Attachment에 자동으로 태그 지정](#)
- [Amazon RDS 기반 오라클 PeopleSoft 애플리케이션의 역할 전환 \(오라클용 사용자 지정\)](#)
- [인프라 코드의 테스트 기반 개발을 위해 Serverspec을 사용합니다](#)

IoT

주제

- [AWS IoT 환경에서 보안 이벤트에 대한 로깅 및 모니터링을 구성합니다.](#)
- [데이터 레이크의 AWS IoT SiteWise 메타데이터 속성 추출 및 쿼리](#)
- [클라이언트 디바이스로 AWS IoT Greengrass를 설정하고 문제를 해결합니다.](#)
- [더 많은 패턴](#)

AWS IoT 환경에서 보안 이벤트에 대한 로깅 및 모니터링을 구성합니다.

작성자: Prateek Prakash(AWS)

환경: 프로덕션	기술: IoT, 보안, 자격 증명, 규정 준수, 운영	워크로드: 기타 모든 워크로드
AWS 서비스: 아마존, 아마존 OpenSearch 서비스, CloudWatch, 아마존, AWS IoT Core, AWS IoT Device Defender, AWS IoT Device Management, 아마존 로그 GuardDuty CloudWatch		

요약

특히 조직에서 IT 환경에 수십억 개의 장치를 연결하고 있기 때문에 사물 인터넷(IoT) 환경의 보안을 보장하는 것이 중요한 우선 순위입니다. 이 패턴은 Amazon Web Services(AWS) 클라우드의 IoT 환경 전반에서 보안 이벤트에 대한 로깅 및 모니터링을 구현하는 데 사용할 수 있는 참조 아키텍처를 제공합니다. 일반적으로 AWS 클라우드의 IoT 환경은 다음과 같은 세 가지 계층으로 구성됩니다.

- 관련 텔레메트리 데이터를 생성하는 IoT 디바이스.
- IoT 디바이스를 다른 디바이스 및 AWS 서비스에 연결하는 AWS IoT 서비스(예: [AWS IoT Core](#), [AWS IoT Device Management](#) 또는 [AWS IoT Device Defender](#)).
- 원격 측정 데이터를 처리하고 다양한 비즈니스 사용 사례에 대한 유용한 통찰력을 제공하는 백엔드 AWS 서비스.

[AWS IoT Lens - AWS Well-Architected Framework](#) 백서에서 제공하는 모범 사례는 클라우드 기반 아키텍처를 검토 및 개선하고 설계 결정이 비즈니스에 미치는 영향을 더 잘 이해하는 데 도움이 될 수 있습니다. 중요한 권장 사항은 디바이스와 AWS 클라우드에서 애플리케이션 로그와 지표를 분석하는 것입니다. 다양한 접근 방식과 기법(예: [위협 모델링](#))을 활용하여 잠재적 보안 문제를 탐지하기 위해 모니터링해야 하는 지표와 이벤트를 식별함으로써 이를 달성할 수 있습니다.

이 패턴은 AWS IoT 및 보안 서비스를 사용하여 AWS 클라우드의 IoT 환경을 위한 보안 로깅 및 모니터링 참조 아키텍처를 설계하고 구현하는 방법을 설명합니다. 이 아키텍처는 기존 AWS 보안 모범 사례를 기반으로 구축되며 이를 IoT 환경에 적용합니다.

사전 조건 및 제한 사항

사전 조건

- 기존 랜딩 존 환경. 이에 대한 자세한 내용은 AWS 권장 가이드 웹사이트에서 [안전하고 확장 가능한 다중 계정 AWS 환경 설정](#) 가이드를 참조하세요.
- 랜딩 존에서 다음 계정을 사용할 수 있어야 합니다.
 - 로그 아카이브 계정-이 계정은 랜딩 영역의 조직 단위(OU) 계정에 대한 로깅 정보에 액세스해야 하는 사용자를 위한 것입니다. 이에 대한 자세한 내용은 AWS 권장 가이드 웹사이트의 [AWS 보안 참조 아키텍처](#) 가이드의 [보안 OU - 로그 아카이브 계정](#) 섹션을 참조하세요.
 - 보안 계정-보안 및 규정 준수 팀은 이 계정을 감사하거나 긴급 보안 작업을 수행하는 데 사용합니다. 이 계정은 Amazon의 관리자 계정으로도 GuardDuty 지정됩니다. 관리자 계정의 사용자는 자신의 계정과 모든 구성원 계정에 대한 GuardDuty 결과를 보고 관리할 수 있을 뿐만 아니라 구성할 GuardDuty 수 있습니다. 이에 대한 자세한 내용은 Amazon GuardDuty 설명서의 [여러 계정 관리를](#) 참조하십시오. GuardDuty
 - IoT 계정-이 계정은 IoT 환경을 위한 것입니다.

아키텍처

이 패턴은 AWS Solutions Library의 [중앙 집중식 로깅 솔루션](#)을 확장하여 보안 관련 IoT 이벤트를 수집하고 처리합니다. 중앙 집중식 로깅 솔루션은 보안 계정에 배포되며 단일 대시보드에서 Amazon CloudWatch 로그를 수집, 분석 및 표시하는 데 도움이 됩니다. 이 솔루션은 여러 소스의 로그 파일을 통합, 관리 및 분석합니다. 마지막으로, 중앙 집중식 로깅 솔루션은 Amazon OpenSearch Service 및 OpenSearch Dashboards를 사용하여 모든 로그 이벤트를 통합적으로 보여줍니다.

다음 아키텍처 다이어그램은 AWS 클라우드의 IoT 보안 로깅 및 참조 아키텍처의 주요 구성 요소를 보여줍니다.

이 다이어그램은 다음 워크플로우를 보여줍니다.

1. IoT는 비정상적인 보안 이벤트가 있는지 모니터링해야 하는 장치입니다. 이러한 디바이스는 에이전트를 실행하여 보안 이벤트 또는 지표를 AWS IoT Core 및 AWS IoT Device Defender에 게시합니다.
2. AWS IoT 로깅이 활성화되면 AWS IoT는 디바이스에서 메시지 브로커 및 규칙 엔진을 통해 Amazon CloudWatch Logs로 전달되는 각 메시지에 대한 진행 이벤트를 전송합니다. CloudWatch 로그 구독을 사용하여 이벤트를 [중앙 집중식 로깅 솔루션으로](#) 푸시할 수 있습니다. 이에 대한 자세한 내용은 AWS IoT Core 설명서의 [AWS IoT 지표 및 차원](#)을 참조하세요.
3. AWS IoT Device Defender는 IoT 디바이스의 안전하지 않은 구성 및 보안 지표를 모니터링하는 데 도움이 됩니다. 이상 징후가 감지되면 AWS Lambda 함수를 구독자로 사용하는 Amazon Simple Notification Service(SNS)에 경보를 전송합니다. Lambda 함수는 Logs에 경보를 메시지로 보냅니다. CloudWatch CloudWatch 로그 구독을 사용하여 중앙 집중식 로깅 솔루션으로 이벤트를 푸시할 수 있습니다. 이에 대한 자세한 내용은 AWS IoT Core 설명서의 [감사 검사, 디바이스 측 지표 및 클라우드 측 지표](#)를 참조하세요.
4. AWS는 변경(예: API 생성, 업데이트 또는 연결)을 수행하는 AWS IoT Core 컨트롤 플레인 작업을 CloudTrail 기록합니다. CloudTrail Landing Zone 구현의 일부로 설정된 경우 CloudWatch Logs에 이벤트를 전송하고 구독을 사용하여 중앙 집중식 로깅 솔루션으로 이벤트를 푸시할 수 있습니다.
5. AWS Config 관리형 규칙 또는 사용자 지정 규칙은 IoT 환경의 일부인 리소스를 평가합니다. CloudWatch 로그를 포함한 CloudWatch 이벤트를 대상으로 사용하여 [규정 준수 변경 알림](#)을 모니터링하세요. 규정 준수 변경 알림이 CloudWatch Logs로 전송된 후 구독을 사용하여 중앙 집중식 로깅 솔루션으로 이벤트를 푸시할 수 있습니다.
6. Amazon은 GuardDuty 지속적으로 CloudTrail 관리 이벤트를 분석하고 알려진 악성 IP 주소, 특이한 지리적 위치 또는 익명화 프록시에서 AWS IoT Core 엔드포인트로 이루어진 API 호출을 식별하는 데 도움을 줍니다. CloudWatch Logs 내의 로그 그룹을 대상으로 하는 Amazon CloudWatch Events를 사용하여 GuardDuty 알림을 모니터링합니다. GuardDuty 알림이 CloudWatch Logs로 전송되면 구독을 사용하여 중앙 모니터링 솔루션으로 이벤트를 푸시하거나 보안 계정의 GuardDuty 콘솔을 사용하여 알림을 볼 수 있습니다.
7. AWS Security Hub는 보안 모범 사례를 사용하여 IoT 계정을 모니터링합니다. CloudWatch 로그의 로그 그룹을 대상으로 하는 CloudWatch 이벤트를 사용하여 Security Hub 알림을 모니터링합니다. Security Hub 알림이 CloudWatch 로그로 전송되면 구독을 사용하여 중앙 모니터링 솔루션으로 이벤트를 푸시하거나 보안 계정의 Security Hub 콘솔을 사용하여 알림을 확인하십시오.
8. Amazon Detective는 정보를 평가 및 분석하여 근본 원인을 찾아내고, AWS IoT 엔드포인트 또는 IoT 아키텍처의 다른 서비스에 대한 비정상적인 호출에 대한 보안 탐지 결과에 따라 조치를 취합니다.
9. Amazon Athena는 Log Archive 계정에 저장된 로그를 쿼리하여 보안 탐지 결과에 대한 이해를 높이고 동향과 악의적인 활동을 식별합니다.

도구

- [Amazon Athena](#)는 표준 SQL을 사용하여 Amazon Simple Storage Service(S3)에 있는 데이터를 직접 간편하게 분석할 수 있는 대화형 쿼리 서비스입니다.
- [CloudTrail](#)은 [AWS](#) 계정의 거버넌스, 규정 준수, 운영 및 위험 감사를 가능하게 합니다.
- [Amazon](#)은 AWS 리소스와 AWS에서 실행하는 애플리케이션을 실시간으로 CloudWatch 모니터링합니다. 리소스와 애플리케이션에 대해 측정할 수 있는 변수인 지표를 수집하고 추적하는 데 사용할 CloudWatch 수 있습니다.
- [Amazon CloudWatch Logs](#)는 사용하는 모든 시스템, 애플리케이션 및 AWS 서비스의 로그를 중앙 집중화합니다. 로그를 보고 모니터링하고, 특정 오류 코드 또는 패턴이 있는지 검색하고, 특정 필드를 기반으로 필터링하거나, 향후 분석을 위해 안전하게 보관할 수 있습니다.
- [AWS Config](#)는 사용자의 AWS 계정에서 AWS 리소스의 구성을 상세하게 볼 수 있도록 합니다.
- [Amazon Detective](#)는 사용자가 보안 조사 결과 또는 의심스러운 활동의 근본 원인을 쉽게 분석 및 조사하고 신속하게 식별할 수 있게 합니다.
- [AWS Glue](#)는 완전 관리형 추출, 변환 및 로드(ETL) 서비스로서 간단하고 경제적으로 데이터를 분류, 정리, 보강하고, 다양한 데이터 스토어와 데이터 스트림 간에 안정적으로 이동할 수 있습니다.
- [GuardDuty](#)은 [Amazon](#)은 지속적인 보안 모니터링 서비스입니다.
- [AWS IoT Core](#)는 인터넷에 연결된 디바이스 (예: 센서, 액추에이터, 임베디드 디바이스, 무선 디바이스, 스마트 어플라이언스) 를 MQTT, HTTPS 및 WAN을 통해 AWS 클라우드에 연결할 수 있는 안전한 양방향 통신을 제공합니다. LoRa
- [AWS IoT Device Defender](#)는 디바이스의 구성을 감사하고, 연결된 디바이스를 모니터링하여 비정상적인 동작을 감지하고, 보안 위협을 완화할 수 있는 보안 서비스입니다.
- [Amazon OpenSearch Service](#)는 AWS 클라우드에서 OpenSearch 클러스터를 쉽게 배포, 운영 및 확장할 수 있는 관리형 서비스입니다.
- [AWS Organizations](#)는 사용자가 생성해 중앙에서 관리하는 단일 조직으로 여러 AWS 계정을 통합할 수 있는 계정 관리 서비스입니다.
- [AWS Security Hub](#)에서는 AWS에서 보안 상태를 포괄적으로 파악할 수 있으며 보안 업계 표준 및 모범 사례와 비교하여 환경을 확인할 수 있습니다.
- [Amazon Virtual Private Cloud\(VPC\)](#)를 사용하면 정의한 가상 네트워크에서 AWS 리소스를 시작할 수 있도록 AWS 클라우드의 논리적으로 격리된 섹션을 프로비저닝할 수 있습니다. 이 가상 네트워크는 AWS의 확장 가능한 인프라를 사용한다는 이점과 함께 고객의 자체 데이터 센터에서 운영하는 기존 네트워크와 매우 유사합니다.

에픽

랜딩 존 환경에서 IoT 계정을 설정하세요

작업	설명	필요한 기술
IoT 계정의 보안 가드레일을 검증합니다.	IoT 계정에서 CloudTrail, AWS Config GuardDuty 및 Security Hub의 가드레일이 활성화되어 있는지 확인하십시오.	AWS 관리자
IoT 계정이 보안 계정의 멤버 계정으로 구성되어 있는지 확인합니다.	IoT 계정이 보안 계정의 Security Hub와 Security Hub의 멤버 계정으로 구성되고 연결되어 있는지 확인하십시오. GuardDuty 이에 대한 자세한 내용은 Amazon GuardDuty 설명서의 AWS Organizations를 통한 GuardDuty 계정 관리 및 AWS Security Hub 설명서의 관리자 및 회원 계정 관리 를 참조하십시오.	AWS 관리자
로그 아카이빙을 검증합니다.	로그 아카이브 CloudTrail 계정에 AWS Config 및 VPC 흐름 로그가 저장되어 있는지 확인합니다.	AWS 관리자

중앙 집중식 로깅 솔루션 설정

작업	설명	필요한 기술
보안 계정에서 중앙 집중식 로깅 솔루션을 설정합니다.	보안 계정으로 AWS Management Console에 로그인하고 AWS 솔루션	AWS 관리자

작업	설명	필요한 기술
	<p>라이브러리에서 중앙 집중식 로깅 솔루션을 설정하여 Amazon OpenSearch 서비스 및 OpenSearch 대시보드에서 CloudWatch 로그를 수집, 분석 및 표시할 수 있습니다.</p> <p>이에 대한 자세한 내용은 AWS 솔루션 라이브러리의 중앙 집중식 로깅 구현 안내서의 중앙 집중식 로깅 솔루션을 사용하여 단일 대시보드에서 Amazon CloudWatch Logs를 수집, 분석 및 표시하는 섹션을 참조하십시오.</p>	

IoT 계정에서 AWS 리소스를 설정하고 구성합니다.

작업	설명	필요한 기술
<p>AWS IoT 로깅을 설정합니다.</p>	<p>IoT 계정을 사용하여 AWS Management Console에 로그인합니다. 로그를 로그로 전송하도록 AWS IoT Core를 CloudWatch 설정하고 구성합니다.</p> <p>이에 대한 자세한 내용은 AWS IoT Core 설명서에서 AWS IoT 로깅 구성 및 CloudWatch 로그를 사용하여 AWS IoT 모니터링을 참조하십시오.</p>	<p>AWS 관리자</p>

작업	설명	필요한 기술
AWS IoT Device Defender를 설정합니다.	<p>AWS IoT Device Defender를 설정하여 IoT 리소스를 감사하고 이상 징후를 탐지합니다.</p> <p>이에 대한 자세한 내용은 AWS IoT Core 설명서의 AWS IoT Device Defender 시작하기를 참조하세요.</p>	AWS 관리자
설정 CloudTrail.	<p>CloudWatch Logs에 이벤트를 CloudTrail 전송하도록 설정합니다.</p> <p>이에 대한 자세한 내용은 AWS CloudTrail 설명서의 CloudWatch 로그로 이벤트 전송을 참조하십시오.</p>	AWS 관리자
AWS Config 및 AWS Config 규칙을 설정합니다.	<p>AWS Config 및 필요한 AWS Config 규칙을 설정합니다. 이에 대한 자세한 내용은 AWS Config 설명서의 콘솔을 이용한 AWS Config 설정 및 콘솔을 이용한 AWS Config 규칙 설정을 참조하세요.</p>	AWS 관리자

작업	설명	필요한 기술
<p>설정 GuardDuty.</p>	<p>CloudWatch Logs 내 로그 그룹을 대상으로 하여 Amazon CloudWatch Events에 결과를 GuardDuty 전송하도록 설정 및 구성합니다.</p> <p>이에 대한 자세한 내용은 Amazon GuardDuty 설명서의 Amazon CloudWatch Events를 사용하여 GuardDuty 결과에 대한 사용자 지정 응답 생성을 참조하십시오.</p>	<p>AWS 관리자</p>
<p>Security Hub 설정.</p>	<p>Security Hub를 설정하고 CIS AWS 파운데이션 벤치마크 및 AWS 기본 보안 모범 사례 표준을 활성화합니다.</p> <p>이에 대한 자세한 내용은 AWS Security Hub 설명서의 자동 응답 및 문제 해결을 참조하세요.</p>	<p>AWS 관리자</p>
<p>Amazon Detective 설정</p>	<p>보안 탐지 결과를 쉽게 분석할 수 있도록 Detective를 설정합니다.</p> <p>자세한 내용은 Amazon Detective 설명서의 Amazon Detective 설정을 참조하세요.</p>	<p>AWS 관리자</p>

작업	설명	필요한 기술
Amazon Athena 및 AWS Glue 설정.	Athena와 AWS Glue를 설정하여 보안 사고 조사를 수행하는 AWS 서비스 로그를 쿼리합니다. 이에 대한 자세한 내용은 Amazon Athena 설명서의 AWS 서비스 로그 쿼리 를 참조하세요.	AWS 관리자

관련 리소스

- [랜딩 존이란 무엇인가요?](#)

데이터 레이크의 AWS IoT SiteWise 메타데이터 속성 추출 및 쿼리

작성자: Ambarish Dongaonkar(AWS)

환경: 프로덕션

기술: IoT, 분석, 빅 데이터

AWS 서비스: AWS IoT SiteWise, AWS Lambda, AWS Glue

요약

AWS IoT는 자산 모델 및 계층 구조를 SiteWise 사용하여 산업 장비, 프로세스 및 시설을 나타냅니다. 각 모델 또는 자산은 환경에 특정한 여러 속성을 가질 수 있습니다. 메타데이터 속성의 예로는 자산의 부지 또는 물리적 위치, 공장 세부 정보, 장비 식별자 등이 있습니다. 이러한 속성 값은 자산 측정 데이터를 보완하여 비즈니스 가치를 극대화합니다. 기계 학습이 메타데이터에 대한 추가 통찰력을 제공하고 엔지니어링 작업을 간소화할 수 있습니다.

하지만 메타데이터 속성은 AWS IoT SiteWise 서비스에서 직접 쿼리할 수 없습니다. 속성을 쿼리할 수 있게 하려면 속성을 추출하여 데이터 레이크에 수집해야 합니다. 이 패턴은 Python 스크립트를 사용하여 모든 AWS IoT SiteWise 자산의 속성을 추출하고 이를 Amazon Simple Storage Service (Amazon S3) 버킷의 데이터 레이크에 수집합니다. 이 프로세스를 완료하면 Amazon Athena의 SQL 쿼리를 사용하여 AWS IoT SiteWise 메타데이터 속성 및 기타 데이터 세트 (예: 측정 데이터 세트)에 액세스할 수 있습니다. 메타데이터 속성 정보는 AWS IoT SiteWise 모니터 또는 대시보드로 작업할 때도 유용합니다. S3 버킷에서 추출된 속성을 사용하여 AWS QuickSight 대시보드를 구축할 수도 있습니다.

패턴에는 참조 코드가 있으며, AWS Lambda 또는 AWS Glue와 같은 사용 사례에 가장 적합한 컴퓨팅 서비스를 사용하여 코드를 구현할 수 있습니다.

사전 조건 및 제한 사항

사전 조건

- 활성 상태의 AWS 계정.
- AWS Lambda 함수 또는 AWS Glue 작업을 설정할 수 있는 권한
- Amazon S3 버킷.
- 자산 모델 및 계층 구조는 AWS IoT에서 설정됩니다. SiteWise 자세한 내용은 [자산 모델 생성 \(AWS IoT SiteWise 설명서\)](#) 을 참조하십시오.

아키텍처

Lambda 함수 또는 AWS Glue 작업을 사용하여 이 프로세스를 완료할 수 있습니다. 모델이 100개 미만 이고 각 모델의 속성이 평균 15개 이하인 경우 Lambda를 사용하는 것이 좋습니다. 다른 모든 사용 사례에서는 AWS Glue를 사용하는 것이 좋습니다.

솔루션 아키텍처 및 워크플로는 다음 다이어그램에 나와 있습니다.

1. 예약된 AWS Glue 작업 또는 Lambda 함수가 실행됩니다. AWS SiteWise IoT에서 자산 메타데이터 속성을 추출하여 S3 버킷으로 수집합니다.
2. AWS Glue 크롤러는 S3 버킷에서 추출된 데이터를 크롤링하여 AWS Glue 데이터 카탈로그에 테이블을 생성합니다.
3. Amazon Athena는 표준 SQL을 사용하여 AWS Glue 데이터 카탈로그의 테이블을 쿼리합니다.

자동화 및 규모 조정

AWS IoT 자산 구성의 업데이트 빈도에 따라 Lambda 함수 또는 AWS Glue 작업이 매일 또는 매주 실행되도록 예약할 수 있습니다. SiteWise

샘플 코드가 처리할 수 있는 AWS IoT SiteWise 자산 수에는 제한이 없지만, 자산 수가 많으면 프로세스를 완료하는 데 필요한 시간이 늘어날 수 있습니다.

도구

- [Amazon Athena](#)는 표준 SQL을 사용하여 Amazon Simple Storage Service(S3)에 있는 데이터를 직접 간편하게 분석할 수 있는 대화형 쿼리 서비스입니다.
- [AWS Glue](#)는 완전관리형 추출, 변환, 적재(ETL) 서비스입니다. 이를 통해 데이터 스토어와 데이터 스트림 간에 데이터를 안정적으로 분류, 정리, 보강하고 이동할 수 있습니다.
- [AWS Identity and Access Management\(IAM\)](#)를 사용하면 사용자에 대해 인증 및 권한 부여를 제어함으로써 AWS 리소스에 대한 액세스를 안전하게 관리할 수 있습니다.
- [AWS IoT를 SiteWise](#) 사용하면 산업용 장비로부터 대규모로 데이터를 수집, 모델링, 분석 및 시각화할 수 있습니다.
- [AWS Lambda](#)는 서버를 프로비저닝하거나 관리할 필요 없이 코드를 실행하는 데 도움이 되는 컴퓨팅 서비스입니다. 필요할 때만 코드를 실행하며 자동으로 확장이 가능하므로 사용한 컴퓨팅 시간만큼만 비용을 지불합니다.

- [Amazon Simple Storage Service\(Amazon S3\)](#)는 원하는 양의 데이터를 저장, 보호 및 검색하는 데 도움이 되는 클라우드 기반 객체 스토리지 서비스입니다.
- [AWS SDK for Python\(Boto3\)](#)는 Python 애플리케이션, 라이브러리 또는 스크립트를 AWS 서비스와 통합하는 데 도움이 되는 소프트웨어 개발 키트입니다.

에픽

작업 또는 기능 설정

작업	설명	필요한 기술
IAM에서 권한을 구성합니다.	<p>IAM 콘솔에서 Lambda 함수 또는 AWS Glue 작업이 위임한 IAM 역할에 권한을 부여하여 다음 작업을 수행할 수 있습니다.</p> <ul style="list-style-type: none"> • AWS IoT SiteWise 서비스에서 읽어보기 • S3 버킷에 쓰기 <p>자세한 내용은 AWS 서비스의 역할 생성(IAM 설명서)을 참조하십시오.</p>	일반 AWS
Lambda 함수 또는 AWS Glue 작업을 생성합니다.	<p>Lambda를 사용하는 경우 새 Lambda 함수를 생성하십시오. 런타임에서 Python을 선택합니다. 자세한 내용은 Python을 사용하여 Lambda 함수 빌드(Lambda 설명서)를 참조하십시오.</p> <p>AWS Glue를 사용하는 경우, AWS Glue 콘솔에서 새 Python 셸 작업을 생성하십시오. 자세한 내용은 Python 셸 작업 추</p>	일반 AWS

작업	설명	필요한 기술
	가 (AWS Glue 설명서)를 참조하십시오.	
Lambda 함수 또는 AWS Glue 작업을 업데이트합니다.	새 Lambda 함수 또는 AWS Glue 작업을 수정하고 추가 정보 섹션에 코드 샘플을 입력합니다. 사용 사례에 맞게 코드를 수정합니다. 자세한 내용은 콘솔 편집기를 사용한 코드 편집 (Lambda 설명서) 및 스크립트 작업 (AWS Glue 설명서)을 참조하십시오.	일반 AWS

작업 또는 함수 실행

작업	설명	필요한 기술
Lambda 함수 또는 AWS Glue 작업을 실행합니다.	Lambda 함수 또는 AWS Glue 작업을 실행합니다. 자세한 내용은 Lambda 함수 간접 호출 (Lambda 설명서) 또는 트리거를 사용하여 작업 시작하기 (AWS Glue 설명서)를 참조하십시오. 이는 AWS IoT SiteWise 계층 구조에서 자산 및 모델의 메타데이터 속성을 추출하여 지정된 S3 버킷에 저장합니다.	일반 AWS
AWS Glue 크롤러를 설정합니다.	CSV 형식 파일에 필요한 형식 분류기를 사용하여 AWS Glue 크롤러를 설정합니다. Lambda 함수 또는 AWS Glue 작업에 사용된 S3 버킷 및 접두사 세부 정보를 사용하십시오. 자세한	일반 AWS

작업	설명	필요한 기술
	내용은 크롤러 정의 (AWS Glue 설명서)를 참조하십시오.	
AWS Glue 크롤러를 실행합니다.	크롤러를 실행하여 Lambda 함수 또는 AWS Glue 작업에서 생성된 데이터 파일을 처리합니다. 크롤러는 지정된 AWS Glue 데이터 카탈로그에서 테이블을 생성합니다. 자세한 내용은 트리거를 사용하여 크롤러 시작하기 (AWS Glue 설명서)를 참조하십시오.	일반 AWS
메타데이터 속성을 쿼리합니다.	Amazon Athena를 사용하면 사용 사례에 따라 표준 SQL을 사용하여 AWS Glue 데이터 카탈로그를 쿼리할 수 있습니다. 메타데이터 속성 테이블을 다른 데이터베이스 및 테이블과 조인할 수 있습니다. 자세한 내용은 시작하기 (Amazon Athena 설명서)를 참조하십시오.	일반 AWS

관련 리소스

- [Amazon Athena 설명서](#)
- [AWS Glue 설명서](#)
- [AWS IoT SiteWise API 레퍼런스](#)
- [AWS IoT SiteWise 사용 설명서](#)
 - [시작하기](#)
 - [산업 자산 모델링](#)
 - [자산 모델 간의 관계 정의\(계층 구조\)](#)
 - [자산 연결 및 연결 해제](#)

- [AWS IoT SiteWise 데모 만들기](#)
- [IOT SiteWise \(Python용 SDK 설명서\)](#)
- [Lambda 설명서](#)

추가 정보

코드

제공된 샘플 코드는 참조용이며 사용 사례에 따라 이 코드를 사용자 지정할 수 있습니다.

```
# Following code can be used in an AWS Lambda function or in an AWS Glue Python shell
job.
# IAM roles used for this job need read access to the AWS IoT SiteWise service and
write access to the S3 bucket.
sw_client = boto3.client('iotsitewise')
s3_client = boto3.client('s3')
output = io.StringIO()

attribute_list=[]
bucket = '{s3_bucket name}'
prefix = '{s3_bucket prefix}'
output.write("model_id,model_name,asset_id,asset_name,attribute_id,attribute_name,attribute_val
\n")

m_resp = sw_client.list_asset_models()
for m_rec in m_resp['assetModelSummaries']:
    model_id = m_rec['id']
    model_name = m_rec['name']

    attribute_list.clear()
    dam_response = sw_client.describe_asset_model(assetModelId=model_id)
    for rec in dam_response['assetModelProperties']:
        if 'attribute' in rec['type']:
            attribute_list.append(rec['name'])

    response = sw_client.list_assets(assetModelId=model_id, filter='ALL')
    for asset in response['assetSummaries']:
        asset_id = asset['id']
        asset_name = asset['name']
        resp = sw_client.describe_asset(assetId=asset_id)
        for rec in resp['assetProperties']:
```

```
    if rec['name'] in attribute_list:
        p_resp = sw_client.get_asset_property_value(assetId=asset_id,
propertyId=rec['id'])
        if 'propertyValue' in p_resp:
            if p_resp['propertyValue']['value']:
                if 'stringValue' in p_resp['propertyValue']['value']:
                    output.write(model_id + "," + model_name + ","
+ asset_id + "," + asset_name + "," + rec['id'] + "," + rec['name'] + "," +
str(p_resp['propertyValue']['value']['stringValue']) + "\n")

                if 'doubleValue' in p_resp['propertyValue']['value']:
                    output.write(model_id + "," + model_name + ","
+ asset_id + "," + asset_name + "," + rec['id'] + "," + rec['name'] + "," +
str(p_resp['propertyValue']['value']['doubleValue']) + "\n")
                if 'integerValue' in p_resp['propertyValue']['value']:
                    output.write(model_id + "," + model_name + ","
+ asset_id + "," + asset_name + "," + rec['id'] + "," + rec['name'] + "," +
str(p_resp['propertyValue']['value']['integerValue']) + "\n")
                if 'booleanValue' in p_resp['propertyValue']['value']:
                    output.write(model_id + "," + model_name + ","
+ asset_id + "," + asset_name + "," + rec['id'] + "," + rec['name'] + "," +
str(p_resp['propertyValue']['value']['booleanValue']) + "\n")

output.seek(0)
s3_client.put_object(Bucket=bucket, Key= prefix + '/data.csv', Body=output.getvalue())
output.close()
```

클라이언트 디바이스로 AWS IoT Greengrass를 설정하고 문제를 해결합니다.

작성자: Marouane Sefiani와 Akalanka De Silva(AWS)

환경: PoC 또는 파일럿	기술: IoT	AWS 서비스: AWS IoT Greengrass, AWS IoT Core
----------------	---------	---

요약

AWS IoT Greengrass는 엣지 디바이스에서 사물 인터넷(IoT) 소프트웨어를 구축, 배포 및 관리하기 위한 오픈 소스 엣지 런타임 및 클라우드 서비스입니다. AWS IoT Greengrass 사용 사례는 다음과 같습니다.

- AWS IoT Greengrass 게이트웨이를 홈 오토메이션의 허브로 사용하는 스마트 홈
- AWS IoT Greengrass가 작업 현장의 데이터 수집 및 로컬 처리를 촉진할 수 있는 스마트 팩토리

AWS IoT Greengrass는 일반적으로 AWS IoT Core에 직접 연결되는 다른 엣지 디바이스(클라이언트 디바이스라고도 함)에 대한 안전하고 인증된 MQTT 연결 엔드포인트 역할을 할 수 있습니다. 이 기능은 클라이언트 디바이스가 AWS IoT Core 엔드포인트에 네트워크로 직접 액세스할 수 없는 경우에 유용합니다.

다음 사용 사례에서 클라이언트 디바이스와 함께 사용하도록 AWS IoT Greengrass를 설정할 수 있습니다.

- 클라이언트 디바이스가 AWS IoT Greengrass로 데이터를 전송하는 경우
- AWS IoT Greengrass가 데이터를 AWS IoT Core로 전달하는 경우
- 고급 AWS IoT Core 규칙 엔진 기능을 활용하는 경우

이러한 기능을 사용하려면 AWS IoT Greengrass 디바이스에 다음 구성 요소를 설치하고 구성해야 합니다.

- MQTT 브로커
- MQTT 브리지

- 클라이언트 디바이스 인증
- IP 감지기

또한 클라이언트 디바이스에서 게시된 메시지는 JSON 형식 또는 [프로토콜 버퍼\(protobuf\)](#) 형식이어야 합니다.

이 패턴은 이러한 필수 구성 요소를 설치 및 구성하는 방법을 설명하고 문제 해결 팁과 모범 사례를 제공합니다.

사전 조건 및 제한 사항

사전 조건

- 활성 상태의 AWS 계정.
- [AWS Command Line Interface\(AWS CLI\) 버전 2](#)
- Python 3.7 이상을 실행하는 2개의 클라이언트 디바이스
- [Java 런타임 환경\(JRE\) 버전 8 이상과 Amazon Corretto 11 또는 OpenJDK 11](#)을 실행하는 코어 디바이스 1개

제한 사항

- AWS IoT Core를 사용할 수 있는 AWS 리전을 선택해야 합니다. AWS IoT Core의 현재 지역 목록은 [리전별 AWS 서비스](#)를 참조하세요.
- 코어 디바이스에는 최소 172MB RAM과 512MB의 디스크 공간이 있어야 합니다.

아키텍처

다음 다이어그램은 이 패턴의 솔루션 아키텍처를 보여 줍니다.

아키텍처에는 다음이 포함됩니다.

- 클라이언트 디바이스 2개 각 디바이스에는 프라이빗 키, 디바이스 인증서 및 루트 인증 기관(CA) 인증서가 포함되어 있습니다. MQTT 클라이언트가 포함된 AWS IoT 디바이스 SDK도 각 클라이언트 디바이스에 설치됩니다.
- 다음 구성 요소와 함께 AWS IoT Greengrass가 배포된 코어 디바이스

- MQTT 브로커
- MQTT 브리지
- 클라이언트 디바이스 인증
- IP 감지기

이 아키텍처는 다음 시나리오를 지원합니다.

- 클라이언트 디바이스는 MQTT 클라이언트를 사용하여 코어 디바이스의 MQTT 브로커를 통해 서로 통신할 수 있습니다.
- 또한 클라이언트 디바이스는 코어 디바이스의 MQTT 브로커 및 MQTT 브리지를 통해 클라우드의 AWS IoT Core와 통신할 수 있습니다.
- 클라우드의 AWS IoT Core는 MQTT 테스트 클라이언트와 코어 디바이스의 MQTT 브리지 및 MQTT 브로커를 통해 클라이언트 디바이스에 메시지를 보낼 수 있습니다.

클라이언트 디바이스와 코어 디바이스 간의 통신에 대한 자세한 내용은 [추가 정보](#) 섹션을 참조하세요.

도구

서비스

- [AWS IoT Greengrass](#)는 디바이스에서 IoT 애플리케이션을 구축, 배포 및 관리하는 데 도움이 되는 오픈 소스 사물 인터넷(IoT) 에지 런타임 및 클라우드 서비스입니다.
- [AWS IoT Core](#)는 인터넷에 연결된 디바이스를 AWS 클라우드에 연결할 수 있도록 안전한 양방향 통신을 제공합니다.
- [AWS IoT 디바이스 SDK](#)는 오픈 소스 라이브러리, 샘플 포함 개발자 설명서, 포팅 안내서를 포함하고 있어 사용자는 선택한 하드웨어 플랫폼에 따라 혁신적인 IoT 제품 또는 솔루션을 구축할 수 있습니다.
- [AWS Identity and Access Management\(IAM\)](#)를 사용하면 사용자에 대해 인증 및 권한 부여를 제어함으로써 AWS 리소스에 대한 액세스를 안전하게 관리할 수 있습니다.

모범 사례

- 변환 및 조건부 작업과 같은 AWS IoT Core 규칙 엔진의 고급 기능을 활용하려면 클라이언트 디바이스의 메시지 페이로드가 JSON 또는 Protobuf 형식이어야 합니다.
- 양방향 통신을 허용하도록 MQTT 브리지를 구성합니다.

- 코어 디바이스의 IP 주소가 MQTT 브로커 인증서의 주체 대체 이름(SAN) 필드에 포함되도록 AWS IoT Greengrass에서 IP 탐지기 구성 요소를 구성하고 배포합니다.

에픽

코어 디바이스 설정

작업	설명	필요한 기술
코어 디바이스에 AWS IoT Greengrass를 설정합니다.	개발자 안내서 의 지침에 따라 AWS IoT Greengrass Core 소프트웨어를 설치합니다.	AWS IoT Greengrass
설치 상태를 확인합니다.	다음 명령을 사용하여 코어 디바이스에서 AWS IoT Greengrass 서비스의 상태를 확인합니다. <pre>sudo systemctl status greengrass.service</pre> 해당 명령 예상 출력은 다음과 같습니다. <pre>Launched Nucleus successfully</pre>	일반 AWS
IAM 정책을 설정하고 Greengrass 서비스 역할에 연결합니다.	1. MQTT 브리지와의 통신을 허용하는 IAM 정책을 생성합니다. 다음은 정책의 예입니다. <pre>{ "Version": "2012-10-17", "Statement": [{ </pre>	일반 AWS

작업	설명	필요한 기술
	<pre data-bbox="646 210 993 1255"> "Effect": "Allow", "Action": ["iot:*"], "Resource ": "*" }, { "Sid": "GreengrassActions ", "Effect": "Allow", "Action": ["greengrass:*"], "Resource ": "*" }] } </pre> <p data-bbox="591 1276 1019 1453">2. 이 정책을 Greengrass 서비스 역할에 연결합니다. 서비스 역할을 가져오려면 다음 명령 사용합니다</p> <pre data-bbox="646 1495 993 1684"> aws greengrassv2 get-service-role-f or-account --region <region> </pre> <p data-bbox="630 1726 987 1810">여기서 <region>은(는) AWS 리전을 나타냅니다.</p>	

작업	설명	필요한 기술
<p>AWS IoT Greengrass 코어 디바이스에서 필수 구성 요소를 구성하고 배포합니다.</p>	<p>다음 구성 요소를 구성하고 배포합니다.</p> <ul style="list-style-type: none"> • greengrass.clientdevices.mqtt.Moquette (구성 세부 정보 참조) • greengrass.clientdevices.mqtt.Bridge (구성 세부 정보 및 다음 작업 참조) • greengrass.clientdevices.Auth (구성 세부 정보 및 다음 작업 이후 작업 참조) • aws.greengrass.clientdevices.IPDetector (구성 세부 정보 참조) 	<p>AWS IoT Greengrass</p>

작업	설명	필요한 기술
<p>MQTT 브리지가 양방향 통신을 허용하는지 확인합니다.</p>	<p>클라이언트 디바이스와 AWS IoT Core 간에 MQTT 메시지를 릴레이하려면 MQTT 브리지 구성 요소를 구성 및 배포하고 릴레이할 주제를 지정합니다. 다음은 그 예입니다:</p> <pre data-bbox="597 537 1029 1415"> { "mqttTopicMapping": { "ClientDevicesToCloud": { "topic": "dt/#", "source": "LocalMqtt", "target": "IotCore" }, "CloudToClientDevices": { "topic": "cmd/#", "source": "IotCore", "target": "LocalMqtt" } } } </pre>	<p>AWS IoT Greengrass</p>

작업	설명	필요한 기술
<p>인증 구성 요소가 클라이언트 디바이스에 연결하여 주제를 게시하거나 구독할 수 있도록 허용하는지 확인합니다.</p>	<p>다음 <code>aws.greengrass.clientdevices.Auth</code> 구성을 사용하면 모든 클라이언트 장치가 연결되고 메시지를 게시하며 모든 주제를 구독할 수 있습니다.</p> <pre data-bbox="597 541 1024 1862"> { "deviceGroups": { "formatVersion": "2021-03-05", "definitions": { "MyPermissiveDeviceGroup": { "selectionRule": "thingName: *", "policyName": "MyPermissivePolicy" } }, "policies": { "MyPermissivePolicy": { "AllowAll": { "statementDescription": "Allow client devices to perform all actions.", "operations": ["*"], "resources": ["*"] } } } } } </pre>	<p>AWS IoT Greengrass</p>

작업	설명	필요한 기술
	<pre> } } </pre>	

클라이언트 디바이스 설정

작업	설명	필요한 기술
AWS IoT 디바이스 SDK를 설치합니다.	<p>클라이언트 디바이스에 AWS IoT 디바이스 SDK를 설치합니다. 지원되는 언어 및 관련 SDK의 전체 목록은 AWS IoT Core 설명서를 참조하세요.</p> <p>예를 들어 Python SDK용 AWS IoT 디바이스 SDK는 여기에 있습니다. GitHub 이 SDK를 설치하려면</p> <ol style="list-style-type: none"> 리포지토리의 사전 요구 사항 페이지의 지침에 따라 Python 3.7 이상이 설치되어 있는지 확인합니다. GitHub pip 명령을 사용하여 SDK를 설치합니다. <p>MacOS와 Linux의 경우</p> <pre>python3 -m pip install awsiotsdk</pre> <p>Windows의 경우</p> <pre>python -m pip install awsiotsdk</pre>	일반 AWS IoT

작업	설명	필요한 기술
	<p>또는 소스 리포지토리에서 SDK를 설치할 수 있습니다.</p> <pre data-bbox="597 331 1026 1003"># Create a workspace directory to hold all the SDK files mkdir sdk-workspace cd sdk-workspace # Clone the repository git clone https://github.com/aws/aws-iot-device-sdk-python-v2.git # Install using Pip (use 'python' instead of 'python3' on Windows) python3 -m pip install ./aws-iot-device-sdk-python-v2</pre>	

작업	설명	필요한 기술
사물 생성.	<ol style="list-style-type: none"> 1. AWS IoT 콘솔에서, 시작하기 버튼이 표시되면 선택합니다. 또는 탐색 창에서 Security(보안)를 확장한 후 Policies(정책)를 선택합니다. 2. 아직 정책이 없습니다 대화 상자가 나타나면 정책 생성을 선택합니다. 그렇지 않은 경우, Create(생성)를 선택합니다. 3. AWS IoT 정책의 이름을 입력합니다(예: ClientDevicePolicy). 4. 설명문 추가 부분에서 기존 정책을 다음 JSON 코드로 바꿉니다. <region> 및 <account> 을(를) AWS 리전 및 AWS 계정 번호로 대체합니다. <pre data-bbox="630 1234 1029 1845"> { "Version": "2012-10-17", "Statement": [{ "Effect": "Allow", "Action": "iot:Connect", "Resource": "arn:aws:iot:region:account:client/*" }], { "Effect": "Allow", </pre>	AWS IoT Core

작업	설명	필요한 기술
	<pre> "Action": "iot:Publish", "Resource": "*" }, { "Effect": "Allow", "Action": "iot:Receive", "Resource": "*" }, { "Effect": "Allow", "Action": "iot:Subscribe", "Resource": "*" }, { "Effect": "Allow", "Action": ["iot:GetThingShadow", "iot:UpdateThingShadow", "iot:DeleteThingShadow"], "Resource": "arn:aws:iot:region:account:thing/*" }] } </pre> <p>5. Create(생성)을 선택합니다.</p>	

작업	설명	필요한 기술
	<p>6. AWS IoT 콘솔의 탐색 창에서 관리, 사물을 선택합니다.</p> <p>7. 아직 사물이 없습니다 대화 상자가 표시되면 사물 등록을 선택합니다. 그렇지 않은 경우, Create(생성)를 선택합니다.</p> <p>8. Creating AWS IoT things(AWS IoT 사물 생성) 페이지에서 Create a single thing(1개의 사물 생성)을 선택합니다.</p> <p>9. Add your device to the device registry(디바이스를 디바이스 레지스트리에 추가) 페이지에서 IoT 사물 이름(예: ClientDevice1)을 입력하고 다음을 선택합니다.</p> <p>주의: 사물을 생성한 후에는 사물 이름을 변경할 수 없습니다. 사물 이름을 변경하려면 새 사물을 생성하고 새 이름을 지정한 다음 이전 사물을 삭제해야 합니다.</p> <p>10 Add a certificate for your thing(사물에 인증서 추가) 페이지에서 Create certificate(인증서 생성)를 선택하세요.</p> <p>11 다운로드 링크를 선택하여 인증서, 프라이빗 키 및 루트</p>	

작업	설명	필요한 기술
	<p>CA 인증서를 다운로드합니다.</p> <p>중요: 이 방법으로만 인증서와 프라이빗 키를 다운로드할 수 있습니다.</p> <p>12.인증서를 활성화하려면 활성화를 선택합니다. 디바이스를 AWS IoT에 연결하려면 인증서가 활성 상태여야 합니다.</p> <p>13.Attach a policy(정책 연결)를 선택합니다.</p> <p>14.사물에 대한 정책 추가에서 사물 등록을 선택합니다 ClientDevicePolicy.</p>	

작업	설명	필요한 기술
<p>Greengrass 코어 디바이스에서 CA 인증서를 다운로드합니다.</p>	<p>Greengrass 코어 디바이스가 오프라인 환경에서 작동할 것으로 예상하는 경우, Greengrass 코어 CA에서 발급한 MQTT 브로커의 인증서를 확인할 수 있도록 Greengrass 코어 CA 인증서를 클라이언트 디바이스에서 사용할 수 있도록 해야 합니다. 따라서 이 인증서의 사본을 확보하는 것이 중요합니다. 다음 방법 중 하나를 사용하여 CA 인증서를 다운로드합니다.</p> <ul style="list-style-type: none"> • PC에서 AWS IoT Greengrass 디바이스에 네트워크로 액세스할 수 있는 경우, 웹 브라우저에 들어가서 <code>https://<device IP>:8883</code> 을 (를) 입력하고 MQTT 브로커 인증서 및 CA 인증서를 확인하세요. CA 인증서를 클라이언트 디바이스에 저장할 수도 있습니다. • 또는 OpenSSL 명령 라인을 사용할 수도 있습니다. <div style="border: 1px solid #ccc; border-radius: 10px; padding: 10px; margin-top: 10px;"> <pre>openssl s_client - showcerts -connect <device IP>:8883</pre> </div>	<p>일반 AWS</p>
<p>클라이언트 디바이스에서 보안 인증 정보를 복사합니다.</p>	<p>Greengrass 코어 CA 인증서, 디바이스 인증서 및 클라이언트 디바이스의 프라이빗 키를 복사합니다.</p>	<p>일반 AWS</p>

작업	설명	필요한 기술
<p>클라이언트 디바이스를 코어 디바이스와 연결합니다.</p>	<p>클라이언트 디바이스를 코어 디바이스와 연결하여 코어 디바이스를 검색할 수 있도록 합니다. 그러면 클라이언트 디바이스는 Greengrass 검색 API를 사용하여 관련 코어 디바이스의 연결 정보 및 인증서를 검색할 수 있습니다. 자세한 내용은 AWS IoT Greengrass 설명서에서 클라이언트 디바이스 연결을 참조하세요.</p> <ol style="list-style-type: none"> 1. AWS IoT Greengrass 콘솔에서, 코어 디바이스를 선택합니다. 2. 관리할 코어 디바이스를 선택합니다. 3. 코어 디바이스의 세부 정보 페이지에서 클라이언트 디바이스 탭을 선택합니다. 4. 연결된 클라이언트 디바이스 부분에서, 클라이언트 디바이스 연결을 선택합니다. 5. 클라이언트 디바이스를 코어 디바이스와 연결 모달에서, 연결할 각 클라이언트 디바이스에 대해 다음을 수행하세요. <ol style="list-style-type: none"> a. 클라이언트 디바이스로 연결할 AWS IoT 사물의 이름을 입력합니다. b. Add(추가)를 선택합니다. 	<p>AWS IoT Greengrass</p>

작업	설명	필요한 기술
	<p>6. Associate(연결)를 선택합니다.</p> <p>연결한 클라이언트 디바이스는 이제 Greengrass 검색 API를 사용하여 이 코어 디바이스를 검색할 수 있습니다.</p>	

데이터 전송 및 수신

작업	설명	필요한 기술
한 클라이언트 디바이스에서 다른 클라이언트 디바이스로 데이터를 전송합니다.	디바이스의 MQTT 클라이언트를 사용하여 <code>dt/client1/sensor</code> 주제에 대한 메시지를 게시합니다.	일반 AWS
클라이언트 디바이스에서 AWS IoT Core로 데이터를 전송합니다.	<p>디바이스의 MQTT 클라이언트를 사용하여 <code>dt/client1/sensor</code> 주제에 대한 메시지를 게시합니다.</p> <p>MQTT 테스트 클라이언트에서 디바이스가 메시지를 보내는 주제를 구독하거나 #를 구독하여 모든 주제를 구독합니다(세부 정보 참조).</p>	일반 AWS
클라이언트 디바이스에서 AWS IoT Core로 데이터를 전송합니다.	MQTT 클라이언트 페이지에서 주제 게시탭의 주제 이름 필드에 메시지의 주제 이름을 입력합니다. 이 예시에서는 주제에 <code>cmd/client1</code> 을 사용합니다.	일반 AWS

문제 해결

문제	Solution
<p>서버 인증서 오류를 확인할 수 없습니다.</p>	<p>이 오류는 MQTT 클라이언트가 TLS 핸드셰이크 중에 MQTT 브로커가 제공한 인증서를 확인할 수 없을 때 발생합니다. 가장 일반적인 이유는 MQTT 클라이언트에 CA 인증서가 없기 때문입니다. 다음 단계에 따라 CA 인증서가 MQTT 클라이언트에 제공되는지 확인합니다.</p> <ol style="list-style-type: none"> 1. PC에서 AWS IoT Greengrass 디바이스에 네트워크로 액세스할 수 있는 경우, 웹 브라우저에 들어가서 <code>https://<device IP>:8883</code>을(를) 입력하고 MQTT 브로커 인증서 및 CA 인증서를 확인하세요. CA 인증서를 클라이언트 디바이스에 저장할 수도 있습니다. <ul style="list-style-type: none"> 또는 OpenSSL 명령 라인을 사용할 수도 있습니다. <pre>openssl s_client -showcerts -connect <device IP>:8883</pre> 2. Moquette CA 및 Greengrass Core CA 인증서의 내용을 파일에 저장한 후, 다음 명령을 사용하여 디코딩된 내용을 확인합니다. <pre>openssl x509 -in <Name of CA>.pem -text</pre> <p>Moquette CA 인증서에는 다음 예와 같이 SAN 필드가 표시되어야 합니다.</p>

문제	Solution
	<pre>X509v3 Subject Alternative Name: IP Address:XXX.XXX.XXX.XXX, IP Address:127.0.0.1, DNS:localhost</pre>
<p>서버 이름 오류를 확인할 수 없습니다.</p>	<p>이 오류는 MQTT 클라이언트가 올바른 서버에 연결되어 있는지 확인할 수 없을 때 발생합니다. 가장 일반적인 이유는 Greengrass 디바이스의 IP 주소가 인증서의 SAN 필드에 나열되지 않기 때문입니다.</p> <p>이전 솔루션의 지침에 따라 MQTT 브로커 인증서를 얻고 추가 정보 섹션에 설명된 대로 SAN 필드에 AWS IoT Greengrass 디바이스의 IP 주소가 포함되어 있는지 확인합니다. 그렇지 않은 경우, IP 탐지기 구성 요소가 제대로 설치되었는지 확인하고 코어 디바이스를 다시 시작하세요.</p>
<p>내장된 클라이언트 디바이스에서 연결할 때만 서버 이름을 확인할 수 없습니다.</p>	<p>임베디드 디바이스에서 널리 사용되는 TLS 라이브러리인 Mbed TLS는 현재, Mbed TLS 라이브러리 코드에서 볼 수 있듯이, 인증서의 SAN 필드에서만 DNS 이름 확인을 지원합니다. 코어 디바이스는 자체 도메인 이름이 없고 IP 주소에 종속되므로 Mbed TLS를 사용하는 TLS 클라이언트는 TLS 핸드셰이크 중에 서버 이름 확인에 실패하여 연결 실패가 발생합니다. x509_cert_check_san 함수에서 Mbed TLS 라이브러리에 SAN IP 주소 확인 기능을 추가하는 것이 좋습니다.</p>

관련 리소스

- [AWS IoT Greengrass 설명서](#)
- [AWS IoT Core 설명서](#)
- [MQTT 브로커 구성 요소](#)

- [MQTT 브리지 구성 요소](#)
- [클라이언트 디바이스 인증 구성 요소](#)
- [IP 탐지기 구성 요소](#)
- [AWS IoT 디바이스 SDK들](#)
- [AWS IoT Greengrass를 사용한 로컬 클라이언트 디바이스 구현](#)(AWS 블로그 게시물)
- [RFC 5280 - 인터넷 X.509 공개 키 인프라 인증서 및 인증서 취소 목록\(CRL\) 프로파일](#)

추가 정보

이 섹션은 클라이언트 디바이스와 코어 디바이스 간의 통신에 대한 추가 정보를 제공합니다.

MQTT 브로커는 코어 디바이스의 포트 8883에서 TLS 클라이언트 연결 시도를 수신 대기합니다. 다음 그림은 MQTT 브로커의 서버 인증서의 예시를 나타냅니다.

예시 인증서에는 다음 세부 정보가 표시됩니다.

- 이 인증서는 코어 디바이스에 국한되고 특정되는 AWS IoT Greengrass Core CA에서 발급합니다. 즉, 로컬 CA 역할을 합니다.
- 이 인증서는 다음 그림과 같이 클라이언트 인증 구성 요소에 의해 매주 자동으로 교체됩니다. 클라이언트 인증 구성 요소 구성에서 이 간격을 설정할 수 있습니다.
- 주체 대체 이름(SAN)은 TLS 클라이언트 측의 서버 이름 확인에서 중요한 역할을 합니다. 이를 통해 TLS 클라이언트가 올바른 서버에 연결되도록 하고 TLS 세션 설정 중에 man-in-the-middle 공격을 피할 수 있습니다. 예시 인증서에서, SAN 필드는 이 서버가 localhost(로컬 Unix 도메인 소켓)에서 수신 대기 중이고 네트워크 인터페이스의 IP 주소가 192.168.1.12임을 나타냅니다.

TLS 클라이언트는 서버 확인 중에 인증서의 SAN 필드를 사용하여 합법적인 서버에 연결되어 있는지 확인합니다. 반대로, HTTP 서버와 브라우저 간의 일반적인 TLS 핸드셰이크에서는 CN(Common Name) 필드 또는 SAN 필드의 도메인 이름을 사용하여 서버 확인 프로세스 중에 브라우저가 실제로 연결하는 도메인을 교차 확인합니다. 코어 디바이스에 도메인 이름이 없는 경우, SAN 필드에 포함된 IP 주소도 같은 용도로 사용됩니다. 자세한 내용은 RFC 5280 — Internet X.509 공개 키 인프라 인증서 및 인증서 취소 목록(CRL) 프로파일의 [주체 대체 이름 섹션](#)을 참조하세요.

AWS IoT Greengrass의 IP 탐지기 구성 요소는 인증서의 SAN 필드에 올바른 IP 주소가 포함되도록 합니다.

예시의 인증서는 로컬 CA 역할을 하는 AWS IoT Greengrass 디바이스에서 서명합니다. TLS 클라이언트(MQTT 클라이언트)는 이 CA를 인식하지 못하므로 다음과 같은 CA 인증서를 제공해야 합니다.

더 많은 패턴

- [AWS IoT Greengrass를 사용하여 IoT 데이터를 Amazon S3에 직접 비용 효율적으로 수집할 수 있습니다](#)

기계 학습 및 AI

주제

- [Athena의 ML 예측을 위한 Amazon DynamoDB의 데이터 집계](#)
- [한 AWS 계정의 AWS CodeCommit 리포지토리를 다른 계정의 SageMaker Studio와 연결](#)
- [이상 탐지를 위한 Amazon Lookout for Vision 교육 및 배포를 자동화합니다.](#)
- [Amazon Textract를 사용하여 PDF 파일에서 콘텐츠 자동 추출하기](#)
- [SageMaker Amazon과 Azure를 사용하여 MLOps 워크플로를 구축하세요. DevOps](#)
- [AWS Step Functions에서 사용자 지정 Docker 컨테이너 이미지를 SageMaker 생성하여 모델 교육에 사용하십시오.](#)
- [Amazon의 추론 파이프라인을 사용하여 단일 엔드포인트의 ML 모델에 전처리 로직을 배포하십시오. SageMaker](#)
- [RAG 및 프롬프트를 사용하여 고급 제너레이티브 AI 채팅 기반 어시스턴트 개발 ReAct](#)
- [Amazon Bedrock 에이전트 및 지식 베이스를 사용하여 완전 자동화된 채팅 기반 도우미를 개발하십시오.](#)
- [Amazon Personalize를 사용하여 개인화되고 순위가 다시 매겨진 추천 생성](#)
- [Amazon에서 사용자 지정 GPU 지원 ML 모델을 트레이닝하고 배포하기 SageMaker](#)
- [테라바이트 SageMaker 규모의 ML 데이터세트의 분산 기능 엔지니어링을 위한 프로세싱 사용](#)
- [Flask와 Elastic Beanstalk를 사용하여 AI/ML 모델 결과를 시각화합니다.](#)
- [더 많은 패턴](#)

Athena의 ML 예측을 위한 Amazon DynamoDB의 데이터 집계

작성자: Sachin Doshi(AWS)와 Peter Molnar(AWS)

코드 리포지토리: Amazon Athena ML을 통해 Amazon DynamoDB 데이터에 대한 ML 예측 사용	환경: 프로덕션	기술: 기계 학습 및 AI, 데이터 베이스, 서버리스
워크로드: 오픈 소스	AWS 서비스: 아마존 아테나, 아마존 DynamoDB, AWS Lambda, 아마존, 아마존 SageMaker QuickSight	

요약

이 패턴은 Amazon Athena를 사용하여 Amazon DynamoDB 테이블에서 복잡한 사물 인터넷(IoT) 데이터 집계를 생성하는 방법을 보여줍니다. 또한 Amazon을 사용하여 기계 학습 (ML) 추론으로 데이터를 보강하는 SageMaker 방법과 Athena를 사용하여 지리공간 데이터를 쿼리하는 방법도 알아봅니다. 이 패턴을 기반으로 조직의 요구 사항을 충족하는 ML 예측 솔루션을 만들 수 있습니다.

데모를 위해 이 패턴은 공유 스쿠터 서비스를 운영하는 기업의 시나리오 예제를 사용하며 여러 도시 지역에 고객을 위해 배치해야 하는 최적의 스쿠터 수를 추론하고자 합니다. 이 회사에서는 지난 4시간을 기반으로 하여 앞으로 한 시간 동안의 고객 수요를 예측하는 사전 학습된 ML 모델을 사용합니다. 이 시나리오는 루이빌 메트로 정부를 위한 [시민 혁신 및 기술 사무소](#)의 공개 데이터 세트를 사용합니다. 이 시나리오의 리소스는 리포지토리에서 사용할 수 있습니다. [GitHub](#)

사전 조건 및 제한 사항

- 활성 상태의 AWS 계정.
- 다음 항목에 대해 AWS ID 및 액세스 관리 (IAM) 역할을 사용하여 AWS CloudFormation 스택을 생성할 수 있는 권한:
 - Amazon Simple Storage Service(S3) 버킷
 - Athena
 - DynamoDB

- SageMaker
- AWS Lambda

아키텍처

기술 스택

- 아마존 QuickSight
- Amazon S3
- Athena
- DynamoDB
- Lambda
- SageMaker

대상 아키텍처

다음 다이어그램은 Athena의 쿼리 기능, Lambda 함수, Amazon S3 스토리지, 엔드포인트 및 대시보드를 사용하여 DynamoDB에서 복잡한 데이터 집계를 구축하는 아키텍처를 보여줍니다. SageMaker QuickSight

이 다이어그램은 다음 워크플로를 보여줍니다.

1. DynamoDB 테이블은 스쿠터 플릿에서 전송되는 IoT 데이터를 수집합니다.
2. Lambda 함수는 수집된 데이터와 함께 DynamoDB 테이블을 로드합니다.
3. Athena 쿼리는 도시 지역을 나타내는 지리공간 데이터에 대한 새 DynamoDB 테이블을 생성합니다.
4. 쿼리 위치는 S3 버킷에 저장됩니다.
5. Athena 함수는 사전 학습된 ML 모델을 호스팅하는 SageMaker 엔드포인트에서 ML 추론을 쿼리합니다.
6. Athena는 DynamoDB 테이블에서 직접 데이터를 쿼리하고 분석을 위해 데이터를 집계합니다.
7. 사용자는 대시보드에서 분석된 데이터의 출력을 확인합니다. QuickSight

도구

AWS 도구

- [Amazon Athena](#)는 표준 SQL을 사용하여 Amazon S3에 있는 데이터를 직접 분석할 수 있는 대화형 쿼리 서비스입니다.
- [Amazon DynamoDB](#)는 빠르고 예측 가능하고 확장 가능한 성능을 제공하는 완전 관리형 NoSQL 데이터베이스 서비스입니다.
- [SageMakerAmazon](#)은 ML 모델을 구축 및 교육한 다음 프로덕션 준비가 완료된 호스팅 환경에 배포할 수 있도록 지원하는 관리형 ML 서비스입니다.
- [Amazon Simple Storage Service\(S3\)](#)는 원하는 양의 데이터를 저장, 보호 및 검색하는 데 도움이 되는 클라우드 기반 객체 스토리지 서비스입니다.
- [QuickSightAmazon](#)은 단일 대시보드에서 데이터를 시각화, 분석 및 보고하는 데 도움이 되는 클라우드 규모의 비즈니스 인텔리전스 (BI) 서비스입니다.
- [AWS Lambda](#)는 서버를 프로비저닝하거나 관리할 필요 없이 코드를 실행하는 데 도움이 되는 컴퓨팅 서비스입니다. 필요할 때만 코드를 실행하며 자동으로 확장이 가능하므로 사용한 컴퓨팅 시간만큼만 비용을 지불합니다.

코드

이 패턴의 코드는 Amazon [Athena ML 리포지토리의 Amazon DynamoDB 데이터에 대한 ML GitHub 사용 예측에서](#) 사용할 수 있습니다. 리포지토리의 CloudFormation 템플릿을 사용하여 예제 시나리오에 사용되는 다음 리소스를 생성할 수 있습니다.

- DynamoDB 테이블
- 테이블에 관련 데이터를 로드하는 Lambda 함수
- Amazon S3에 저장된 사전 학습된 XGBoost 모델을 사용한 추론 요청용 SageMaker 엔드포인트
- V2EngineWorkGroup(으)로 명명된 Athena 작업 그룹
- Athena라는 쿼리를 통해 지리공간 셰이프파일을 검색하고 스쿼터 수요를 예측
- 사전 구축된 [Amazon Athena DynamoDB 커넥터](#)로, 이를 통해 Athena는 DynamoDB와 통신하고 [AWS Serverless Application Model\(AWS SAM\)](#)을 사용하여 DynamoDB 커넥터를 참조하여 애플리케이션을 구축

에픽

예제 데이터 세트 받기

작업	설명	필요한 기술
<p>데이터 세트와 리소스를 다운로드하세요.</p>	<ol style="list-style-type: none"> 1. 개인형 이동장치 대여에 대한 공개 데이터 세트를 다운로드하세요. 이 데모 데이터는 사용 사례의 일부로 DynamoDB에 미리 생성되지만 프로덕션 환경에서는 IoT 디바이스 또는 Amazon Kinesis 소비자과 같은 다양한 메커니즘을 통해 이 데이터를 DynamoDB로 전송합니다. 이러한 메커니즘은 Lambda를 사용하여 DynamoDB에 데이터를 삽입합니다. 2. 켄터키주 루이빌 시의 역사적, 문화적 지역의 경계를 나타내는 GIS 셰이프파일을 다운로드하세요. 이 공개 데이터 세트는 켄터키주 루이빌 및 제퍼슨 카운티 정보 컨소시엄에서 제공됩니다. 원본 셰이프파일은 이미 Athena로 쿼리할 수 있는 텍스트 파일로 변환되어 있지만 Jupyter 노트북에서 Amazon Athena를 사용한 GIS 셰이프파일의 지리 공간 처리에서 셰이프파일을 변환하는 Python 코드를 찾을 수 있습니다. GitHub 	<p>앱 개발자, 데이터 사이언티스트</p>

작업	설명	필요한 기술
	<ol style="list-style-type: none"> 3. 및 SageMaker Athena를 사용하여 시간별 예측에 대해 ML 모델을 학습시키는 사전 학습된 Python 코드를 다운로드하십시오. 4. Athena의 SQL 쿼리를 사용하면 DynamoDB에 저장된 데이터에서 실시간 예측을 위해 모든 것을 통합할 수 있습니다. 5. (선택 사항) 켄터키주 루이빌 지도에서 지리공간 데이터를 시각화하는 QuickSight 데 사용합니다. 	

템플릿을 사용하여 필요한 리소스를 배포하세요. CloudFormation

작업	설명	필요한 기술
CloudFormation 스택을 생성하세요.	<ol style="list-style-type: none"> 1. GitHub 리포지토리에서 CloudFormation 템플릿을 다운로드합니다. 2. AWS Management Console에 로그인하고 us-east-1을 선택합니다. 참고: ML 모델은 us-east-1 AWS 리전의 Amazon Elastic Container Registry(Amazon ECR)에 저장되지만 패턴은 리전에 구매받지 않습니다. 이 패턴에 사용되는 AWS 서비스가 지원되는 모든 리전에서 패턴을 복제할 수 있습니다. 	AWS DevOps

작업	설명	필요한 기술
	<ol style="list-style-type: none"> 3. CloudFormation 콘솔을 열고 탐색 창에서 [Stacks] 를 선택합니다. 4. 스택 생성을 선택한 다음 기존 리소스 사용(리소스 가져오기))를 선택합니다. 5. 리소스 식별 페이지에서 다음을 선택합니다. 6. 템플릿 지정 섹션의 템플릿 소스에서 템플릿 파일 업로드를 선택합니다. 7. 파일을 선택한 다음 이전에 다운로드한 CloudFormation 템플릿을 선택합니다. 8. 다음을 선택하고 기본 파라미터 값을 적용한 다음, 다음을 선택하여 나머지 설치 마법사를 단계별로 진행합니다. 9. AWS가 사용자 지정 이름으로 IAM 리소스를 생성할 CloudFormation 수 있음을 인정함 확인란을 선택합니다. 10. 스택 생성을 선택합니다. <p>참고: CloudFormation 스택에서 이러한 리소스를 생성하는데 15~20분이 소요될 수 있습니다.</p>	

작업	설명	필요한 기술
<p>CloudFormation 배포를 확인합니다.</p>	<p>CloudFormation 템플릿의 샘플 데이터가 DynamoDB에 로드되었는지 확인하려면 다음을 수행하십시오.</p> <ol style="list-style-type: none"> 1. DynamoDB 콘솔을 열고 탐색 창에서 테이블을 선택합니다. 2. 테이블 섹션에서 DynamoDBTableDocklessVehicles 테이블을 확인합니다. 3. 리소스 생성이 완료되면 Athena 콘솔을 열고 탐색 창에서 워크그룹을 선택합니다. 4. V2EngineWorkGroup 워크그룹을 선택한 다음 워크그룹 전환을 선택합니다. 5. 쿼리 결과 위치를 저장하라는 메시지가 표시되면 쓰기 권한이 있는 Amazon S3 위치를 선택합니다. 6. 저장을 선택합니다. 7. 탐색 창에서 athena-m1-db-<code><your-AWS-account-number></code> 데이터베이스를 선택한 다음 쿼리 에디터를 선택합니다. 	<p>앱 개발자</p>

Athena에 지리 위치 파일 불러오기

작업	설명	필요한 기술
<p>지리 위치 데이터가 담긴 Athena 테이블을 만듭니다.</p>	<p>Athena에 지리 위치 파일을 로드하려면 다음을 수행합니다.</p> <ol style="list-style-type: none"> 1. Athena 콘솔을 열고 탐색 창에서 쿼리 편집기를 선택합니다. 2. 샘플 쿼리 탭을 선택합니다. 3. Q1: 지역을 검색하고 선택합니다. 4. 쿼리 편집기로 돌아가려면 편집기 탭을 선택합니다. 5. Run(실행)을 선택합니다. 그러면 데이터베이스에 이름이 <code>louisville_ky_neighborhoods</code>인 테이블이 만들어집니다. <code>athena-m1-db-<your-AWS-account-number></code> 데이터베이스에서 테이블이 생성됩니다. <p>쿼리는 도시 지역을 나타내는 지리 공간 데이터에 대한 새 테이블을 생성합니다. 데이터 테이블은 GIS shapefile에서 생성됩니다. CREATE EXTERNAL TABLE 명령문은 테이블의 스키마와 기본 데이터 파일의 위치 및 형식을 정의합니다.</p> <p>쉐이프파일을 처리하고 이 테이블을 생성하는 Python 코</p>	<p>데이터 엔지니어</p>

작업	설명	필요한 기술
	<p>드는 AWS 샘플에서 Amazon Athena를 사용한 GIS 웨이프 파일의 지리 공간 처리를 참조하세요. 자세한 SQL 코드는 create_neighborhood_table.sql을 참조하십시오. GitHub</p>	

집계된 DynamoDB 데이터를 바탕으로 지역별 스쿠터 수요를 예측

작업	설명	필요한 기술
<p>Athena에서 쿼리할 함수를 선언합니다. SageMaker</p>	<ol style="list-style-type: none"> 1. Athena 콘솔을 열고 탐색 창에서 쿼리 편집기를 선택한 다음 편집기 탭을 선택합니다. 2. 다음 DDL 문을 복사하여 콘솔 쿼리 편집기에 붙여 넣습니다. <div style="border: 1px solid #ccc; border-radius: 10px; padding: 10px; margin-top: 10px;"> <pre> USING EXTERNAL FUNCTION predict_demand (location_id BIGINT, hr BIGINT , dow BIGINT, n_pickup_1 BIGINT, n_pickup_2 BIGINT, n_pickup_3 BIGINT, n_pickup_4 BIGINT, n_dropoff_1 BIGINT, n_dropoff_2 BIGINT, n_dropoff_3 BIGINT, n_dropoff_4 BIGINT) RETURNS DOUBLE SAGEMAKER '<Your SageMaker endpoint>' </pre> </div>	<p>데이터 사이언티스트, 데이터 엔지니어</p>

작업	설명	필요한 기술
	<p>SQL 문의 첫 번째 부분에서는 사전 학습된 모델을 호스팅하는 SageMaker 엔드포인트에서 ML 추론을 쿼리하는 외부 함수를 선언합니다.</p> <p>뒤이어 다음과 같이 하세요.</p> <ol style="list-style-type: none"> 1. 입력 파라미터의 순서와 유형, 그리고 반환값의 유형을 정의합니다. 2. Run(실행)을 선택합니다. 	

작업	설명	필요한 기술
<p>집계된 DynamoDB 데이터를 바탕으로 지역별 스쿠터 수요를 예측합니다.</p>	<p>이제 Athena를 사용하여 DynamoDB에서 직접 트랜잭션 데이터를 쿼리한 다음 분석 및 예측을 위한 데이터를 집계할 수 있습니다. DynamoDB NoSQL 데이터베이스를 직접 쿼리하는 방식으로는 이 작업을 쉽게 수행할 수 없습니다.</p> <ol style="list-style-type: none"> 1. Athena 콘솔을 열고 탐색 창에서 쿼리 편집기를 선택합니다. 2. Sample queries(샘플 쿼리) 탭을 선택합니다. 3. Q2: DynamoDathenAML을 검색하여 선택합니다. ScooterPredict 4. 쿼리 편집기로 돌아가려면 편집기 탭을 선택합니다. 5. Run(실행)을 선택합니다. <p>SQL 문은 다음과 같은 작업을 수행합니다.</p> <ul style="list-style-type: none"> • Athena 페더레이션 쿼리를 사용하여 원시 트립 데이터로 DynamoDB 테이블을 쿼리 • Athena의 지리공간 함수를 사용하여 지역에 지리적 좌표를 배치 • 다음을 사용하여 ML 추론으로 데이터를 보강합니다. SageMaker 	<p>앱 개발자, 데이터 사이언티스트</p>

작업	설명	필요한 기술
	Athena에서 SQL을 사용하여 DynamoDB 데이터 및 SageMaker 추론 데이터를 집계하는 방법에 대한 자세한 내용은 의 athena_long.sql 를 참조하십시오. GitHub	
출력을 확인합니다.	<p>결과 테이블에는 지역 중심의 이웃, 경도, 위도가 포함됩니다. 여기에는 다음 한 시간 동안의 차량 수 예측도 포함됩니다.</p> <p>쿼리는 선택한 시점에 대한 예측을 생성합니다. 명령문의 모든 위치에서 표현식 <code>TIMESTAMP '2019-09-07 15:00'</code> 을 변경하여 원하는 시간에 대한 예측을 할 수 있습니다.</p> <p>DynamoDB 테이블에 실시간 데이터 피드가 있는 경우 타임스탬프를 <code>NOW()</code> 로 변경하세요.</p>	앱 개발자, 데이터 사이언티스트

환경 정리

작업	설명	필요한 기술
리소스를 삭제합니다.	<ol style="list-style-type: none"> Athena 콘솔을 열고 <u>스택의 CloudFormation 일부로 만든 버킷을 비웁니다.</u> CloudFormation 콘솔을 열고 이름이 지정된 <u>bdb-1462-athena-dy</u> 	앱 개발자, AWS DevOps

작업	설명	필요한 기술
	<p>namodb-ml-stack 스택을 삭제합니다.</p> <p>3. Amazon CloudWatch 콘솔을 열고 이름이 지정된 로그 그룹을 삭제합니다/ aws/sagemaker/Endpoints/Sg-athena-ml-dynamodb-model-endpoint .</p>	

관련 리소스

- [아마존 아테나 쿼리 페더레이션 SDK](#) () GitHub
- [지리공간 데이터 쿼리](#)(Amazon Athena 사용 설명서)
- [Amazon Athena ML을 통해 Amazon DynamoDB 데이터에 대한 ML 예측 사용](#)(AWS 빅 데이터 블로그)
- [ElastiCache Redis용 Amazon](#) (AWS 설명서)
- [Amazon Neptune](#)(AWS 설명서)

한 AWS 계정의 AWS CodeCommit 리포지토리를 다른 계정의 SageMaker Studio와 연결

작성자: Laurens van der Maas(AWS) 및 Aubrey Oosthuizen(AWS)

환경: 프로덕션

기술: 기계 학습 및 AI, 보안, ID, 규정 DevOps 준수, 클라우드 네이티브

AWS 서비스: AWS CodeCommit, 아마존 SageMaker, AWS ID 및 액세스 관리

요약

이 패턴은 한 AWS 계정 (계정 A) 의 AWS CodeCommit 리포지토리를 다른 AWS 계정 (계정 B) 의 Amazon SageMaker Studio와 연결하는 방법에 대한 지침과 코드를 제공합니다. 연결을 설정하려면 계정 A에서 AWS Identity and Access Management (IAM) 정책과 역할을 생성하고 계정 B에서 IAM 인라인 정책을 생성한 다음 셸 스크립트를 사용하여 계정 A의 CodeCommit 리포지토리를 계정 B의 SageMaker Studio로 복제해야 합니다.

사전 조건 및 제한 사항

사전 조건

- 두 개의 [AWS 계정](#) (하나는 CodeCommit 리포지토리를 포함하고 다른 하나는 사용자가 있는 SageMaker 도메인을 포함하는 계정)
- 가상 사설망 (VPC) 엔드포인트를 통해 인터넷 액세스 또는 AWS Security Token Service (AWS STS) 에 액세스할 수 있는 프로비저닝된 [SageMaker 도메인 CodeCommit 및 사용자](#)
- [IAM](#) 대한 기본적인 이해
- [Studio에 대한 기본 이해 SageMaker](#)
- [Git](#)에 대한 기본적인 이해 및 [CodeCommit](#)

제한 사항

이 패턴은 SageMaker 스튜디오에만 적용되며 SageMaker 아마존의 RStudio에는 적용되지 않습니다.

아키텍처

기술 스택

- 아마존 SageMaker
- 아마존 SageMaker 스튜디오
- AWS CodeCommit
- AWS Identity and Access Management(IAM)
- Git

대상 아키텍처

다음 다이어그램은 계정 A의 CodeCommit 리포지토리를 계정 B의 SageMaker Studio에 연결하는 아키텍처를 보여줍니다.

이 다이어그램은 다음 워크플로를 보여줍니다.

1. 사용자는 역할을 통해 계정 A의 MyCrossAccountRepositoryContributorRole 역할을 수임하고, 계정 B의 SageMaker Studio에서는 SageMaker 실행 역할을 사용하며, 수임된 역할에는 지정된 저장소를 복제하고 상호 작용할 수 있는 CodeCommit 권한이 포함됩니다. sts:AssumeRole
2. 사용자는 Studio의 시스템 터미널에서 SageMaker Git 명령을 수행합니다.

자동화 및 규모 조정

이 패턴은 [AWS 클라우드 개발 키트 \(AWS CDK\) CloudFormation, AWS](#) 또는 [Terraform](#)을 사용하여 자동화할 수 있는 수동 단계로 구성되어 있습니다.

도구

도구

- [SageMakerAmazon](#)은 ML 모델을 구축 및 교육한 다음 프로덕션에 바로 사용할 수 있는 호스팅 환경에 배포하는 데 도움이 되는 관리형 기계 학습 (ML) 서비스입니다.
- [Amazon SageMaker Studio](#)는 기계 학습 모델을 구축, 교육, 디버그, 배포 및 모니터링할 수 있는 기계 학습을 위한 웹 기반 통합 개발 환경 (IDE)입니다.

- [CodeCommitAWS](#)는 자체 소스 제어 시스템을 관리할 필요 없이 Git 리포지토리를 비공개로 저장하고 관리할 수 있는 버전 제어 서비스입니다.
- [AWS Identity and Access Management\(IAM\)](#)는 누구에게 인증 및 사용 권한이 있는지 제어하여 AWS 리소스에 대한 액세스를 안전하게 관리할 수 있도록 도와줍니다.

기타 도구

- [Git](#)은 소프트웨어 개발 중에 소스 코드의 변경 사항을 추적하기 위한 분산 버전 제어 시스템입니다.

에픽

계정 A에 IAM 정책 및 IAM 역할 생성

작업	설명	필요한 기술
계정 A에서 리포지토리 액세스에 대한 IAM 정책을 만듭니다.	<ol style="list-style-type: none"> 1. AWS Management Console에 로그인하고 IAM 콘솔을 엽니다. 2. 탐색 창에서 정책을 선택한 후 정책 생성을 선택합니다. 3. JSON 탭을 선택합니다. 4. 이 패턴의 추가 정보 섹션에 있는 예제 IAM 정책의 정책 명령문을 복사한 다음 해당 명령문을 JSON 편집기에 붙여 넣습니다. 정책의 모든 플레이스홀더 값을 바꿔야 합니다. 5. 다음: 태그를 선택한 후 다음: 검토를 선택합니다. 6. 이름에 정책 이름을 입력합니다. 참고: 이 패턴에서는 IAM 정책을 CrossAccountAccessForMySharedDemoRepo 로 가리키지 	AWS DevOps

작업	설명	필요한 기술
	<p>만 원하는 정책 이름을 선택할 수 있습니다.</p> <p>7. 정책 생성을 선택합니다.</p> <p>팁: IAM 정책의 범위를 사용 사례에 필요한 최소 권한으로 제한하는 것이 가장 좋습니다.</p>	

작업	설명	필요한 기술
계정 A에서 리포지토리 액세스를 위한 IAM 역할을 생성합니다.	<ol style="list-style-type: none"> 1. IAM 콘솔의 탐색 창에서 역할을 선택하고 역할 생성을 선택합니다. 2. 신뢰할 수 있는 엔터티 유형에서 AWS 계정을 선택합니다. 3. AWS 계정 섹션에서 다른 AWS 계정을 선택합니다. 4. 계정 ID에 계정 B의 계정 ID를 입력합니다. 5. 권한 추가 페이지에서, 이전에 생성한 CrossAccountAccessForMySharedDemoRepo 정책을 검색하여 선택합니다. 6. 다음을 선택합니다. 7. 역할 이름에 이름을 입력합니다. 참고: 이 패턴에서는 IAM 역할을 MyCrossAccountRepositoryContributorRole 로 가리키지만 원하는 역할 이름을 선택할 수 있습니다. 8. 역할 생성을 선택한 다음 새 역할의 Amazon 리소스 이름 (ARN)을 복사합니다. 	AWS DevOps

계정 B에서 IAM 인라인 정책 생성

작업	설명	필요한 기술
<p>계정 B의 SageMaker 도메인 사용자에게 연결된 실행 역할에 인라인 정책을 연결합니다.</p>	<ol style="list-style-type: none"> 1. IAM 콘솔의 탐색 창에서 역할을 선택합니다. 2. 계정 B에서 SageMaker 도메인 사용자에게 할당된 실행 역할을 검색하고 선택합니다. 3. 권한 추가를 선택하고 인라인 정책 생성을 선택합니다. 4. JSON 탭을 선택합니다. 5. 다음 정책 명령문을 복사한 다음 JSON 편집기에 붙여 넣습니다. <pre data-bbox="630 968 1029 1759"> { "Version": "2012-10-17", "Statement": [{ "Sid": "VisualEditor0", "Effect": "Allow", "Action": "sts:AssumeRole", "Resource": "arn:aws:iam::<Account_A_ID>:role/<Account_A_Role_Name>" }] } </pre> 6. <Account_A_ID> 를 계정 A의 계정 ID를 바꾸십시오. 	<p>AWS DevOps</p>

작업	설명	필요한 기술
	<p><Account_A_Role_Name> 을 이전에 생성한 IAM 역할의 이름으로 바꾸십시오.</p> <p>7. 정책 검토를 선택합니다.</p> <p>8. 이름에는 인라인 정책의 이름을 입력합니다.</p> <p>9. 정책 생성을 선택합니다.</p>	

SageMaker 스튜디오에서 계정 B용 리포지토리를 복제하십시오.

작업	설명	필요한 기술
SageMaker Studio에서 계정 B의 셸 스크립트를 생성합니다.	<ol style="list-style-type: none"> 1. SageMaker 콘솔의 탐색 창에서 Studio를 선택합니다. 2. 사용자 프로필을 선택한 다음 Studio 열기를 선택합니다. 3. 홈 섹션에서 런처 열기를 선택합니다. 4. 유틸리티 및 파일 섹션에서 텍스트 파일을 선택합니다. 5. 이 패턴의 추가 정보 섹션에 있는 예제 SageMaker 셸 스크립트에서 스크립트를 복사한 다음 명령문을 새 파일에 붙여넣습니다. 스크립트의 모든 플레이스홀더 값을 바꿔야 합니다. 6. 새 파일의 untitled.txt 탭을 마우스 오른쪽 버튼으로 클릭한 다음 텍스트 이름 바꾸기를 선택합니다. 새 이 	AWS DevOps

작업	설명	필요한 기술
	<p>름에 cross_account_git_clone.sh를 입력한 다음 이름 바꾸기를 선택합니다.</p>	
<p>시스템 터미널에서 셸 스크립트를 간접 호출합니다.</p>	<ol style="list-style-type: none"> 1. SageMaker 콘솔의 홈 섹션에서 [런처 열기] 를 선택합니다. 2. 유틸리티 및 파일 섹션에서 시스템 터미널을 선택합니다. 3. 터미널에서 다음 명령을 실행합니다. <pre data-bbox="634 814 1029 1010"> chmod u+x ./cross_account_git_clone.sh && ./cross_account_git_clone.sh </pre> <p>SageMaker Studio 크로스 계정에서 CodeCommit 리포지토리를 복제했습니다. 이제 시스템 터미널에서 모든 Git 명령을 수행할 수 있습니다.</p>	<p>AWS DevOps</p>

추가 정보

IAM 정책 예제

이 예제 정책을 사용하려면 다음을 수행하십시오.

- 리포지토리의 경우 AWS 리전으로 <CodeCommit_Repository_Region>을 대체하십시오.
- 계정 A의 경우 해당 계정 ID로 <Account_A_ID>를 바꿉니다.
- 계정 A의 CodeCommit 리포지토리 <CodeCommit_Repository_Name> 이름으로 바꾸십시오.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "codecommit:BatchGet*",
        "codecommit:Create*",
        "codecommit>DeleteBranch",
        "codecommit:Get*",
        "codecommit:List*",
        "codecommit:Describe*",
        "codecommit:Put*",
        "codecommit:Post*",
        "codecommit:Merge*",
        "codecommit:Test*",
        "codecommit:Update*",
        "codecommit:GitPull",
        "codecommit:GitPush"
      ],
      "Resource": [
        "arn:aws:codecommit:<CodeCommit_Repository_Region>:<Account_A_ID>:<CodeCommit_Repository_Name>"
      ]
    }
  ]
}
```

SageMaker 셸 스크립트 예시

이 예제 스크립트를 사용하려면 다음을 수행하십시오.

- 계정 A의 경우 해당 계정 ID로 <Account_A_ID>를 바꿉니다.
- 이전에 생성한 IAM 역할의 이름을 <Account_A_Role_Name>으로 바꾸십시오.
- 리포지토리의 경우 AWS 리전으로 <CodeCommit_Repository_Region>을 대체하십시오.
- 계정 A의 CodeCommit 리포지토리 <CodeCommit_Repository_Name> 이름으로 바꾸십시오.

```
#!/usr/bin/env bash
#Launch from system terminal
pip install --quiet git-remote-codecommit
```

```
mkdir -p ~/.aws
touch ~/.aws/config

echo "[profile CrossAccountAccessProfile]
region = <CodeCommit_Repository_Region>
credential_source=EcsContainer
role_arn = arn:aws:iam::<Account_A_ID>:role/<Account_A_Role_Name>
output = json" > ~/.aws/config

echo '[credential "https://git-
codecommit.<CodeCommit_Repository_Region>.amazonaws.com"]
  helper = !aws codecommit credential-helper $@ --profile
  CrossAccountAccessProfile
  UseHttpPath = true' > ~/.gitconfig

git clone codecommit::<CodeCommit_Repository_Region>://
CrossAccountAccessProfile@<CodeCommit_Repository_Name>
```

이상 탐지를 위한 Amazon Lookout for Vision 교육 및 배포를 자동화합니다.

작성자: 마이클 월너 (AWS), 가브리엘 로드리게스 가르시아 (AWS), 캉캉 왕 (AWS), 슈크라트 코드자 예프 (AWS), 산제이 아쇼크 (AWS), 야신 자푸리 (AWS), 가브리엘 질카 (AWS)

코드 리포지토리 [detection-using-amazon-lookout](#) - 비전을 위한 [automated-silicon-wafer-anomaly](#)

환경: 프로덕션

기술: 기계 학습 및 AI, 클라우드 네이티브, DevOps

AWS 서비스: AWS, AWS CloudFormation, AWS, AWS CodeBuild, AWS CodeCommit, AWS CodePipeline Lambda, Amazon Lookout for Vision

요약

이 패턴은 시각적 검사를 위한 [Amazon Lookout for Vision](#) 기계 학습 모델의 교육 및 배포를 자동화하는 데 도움이 됩니다. 이 패턴은 실리콘 웨이퍼의 이상 감지에 중점을 두고 있지만 다양한 제품 및 산업에서 사용할 수 있도록 솔루션을 조정할 수 있습니다.

2020년, 세계 최대 반도체 제조업체 중 한 곳의 연간 생산 능력은 12인치 상당 웨이퍼 1,200만 장을 넘어섰습니다. 이러한 웨이퍼의 품질과 신뢰성을 보장하기 위한 육안 검사는 생산 공정의 필수 단계입니다. 수동 샘플링이나 통계적 측정에 의존하는 구식 레거시 도구를 사용하는 것과 같은 기존의 육안 검사 방법은 시간이 많이 걸리고 비효율적일 수 있습니다. 이 프로세스의 규모와 광범위한 반도체 산업에 미치는 중요성을 고려할 때 고급 인공 지능 (AI) 기술을 사용하여 육안 검사를 최적화하고 자동화할 수 있는 상당한 기회가 있습니다.

Lookout for Vision을 사용하면 이미지 및 개체 검사 프로세스를 간소화하여 비용이 많이 들고 일관되지 않은 수동 검사의 필요성을 줄일 수 있습니다. 이 솔루션은 품질 관리를 개선하고 정확한 결함 및 손상 평가를 촉진하며 산업 표준 준수를 보장합니다. 또한 기계 학습에 대한 전문 지식이 없어도 Lookout for Vision 검사 프로세스를 자동화할 수 있습니다.

이 솔루션을 사용하면 컴퓨터 비전 모델을 모든 시스템에 통합할 수 있습니다. 예를 들어, 사용자가 이미지를 업로드하고 이미지를 분석하여 결함이 있는지 분석할 수 있는 웹 사이트에 모델을 통합할 수 있습니다. 다음 이미지는 화학적 기계적 연마 (CMP) 공정에서 스크래치 결함이 있는 실리콘 웨이퍼의 예를 보여줍니다. Lookout for Vision을 사용하여 이러한 이상 현상을 탐지할 수 있습니다. 예를 들어, Lookout for Vision은 이 이미지에서 99.04% 신뢰도로 이상 징후를 감지했습니다.

이 솔루션은 [Amazon Lookout for Vision을 사용하여 이벤트 기반 추적 솔루션 구축하기 블로그 게시물에 설명된 코드와 사용 사례를 기반으로](#) 합니다. 이 솔루션은 CI/CD 파이프라인 자동화를 활성화하고 오픈 소스 [Amazon Lookout for Vision Python SDK](#) () 를 통합하도록 원본 코드를 수정합니다. GitHub Python SDK에 대한 자세한 내용은 Python SDK 블로그 게시물을 [사용하여 Amazon Lookout for Vision 모델의 구축, 교육 및 배포를](#) 참조하십시오.

사전 조건 및 제한 사항

사전 조건

- 활성 상태의 AWS 계정
- AWS 계정의 관리 권한
- AWS Command Line Interface(AWS CLI), [설치](#) 및 [구성됨](#)
- AWS CDK, [설치 및 구성](#)
- [Python 버전 3.10](#), [설치됨](#)

아키텍처

대상 아키텍처

이 아키텍처는 CI/CD 파이프라인을 통한 Amazon Lookout for Vision 모델의 빌드, 교육 및 배포 자동화를 보여줍니다. 이 다이어그램은 다음 워크플로를 보여줍니다.

1. 코드는 Amazon CodeCommit 리포지토리에 저장됩니다. 개발자는 코드를 수정하거나 입력 이미지를 변경하거나 자동화 파이프라인에 다른 단계를 추가할 수 있습니다.
2. 솔루션을 배포하거나 CodeCommit 리포지토리의 기본 브랜치를 업데이트한 후 Amazon은 CodePipeline 자동으로 코드를 Amazon에 푸시합니다. CodeBuild

3. CodeBuild Lookout for Vision Python SDK를 사용하여 이미지 분류 모델을 훈련하고 배포합니다. 교육에 사용되는 이미지는 Amazon Simple Storage 서비스 (Amazon S3) 버킷에 저장됩니다. CodeBuild 이러한 이미지를 자동으로 다운로드하여 저장합니다. 솔루션을 필요에 맞게 사용자 정의하기 위해 자체 이미지를 가져올 수 있습니다.
4. Lookout for Vision 모델은 AWS Lambda를 통해 최종 사용자에게 공개됩니다. 하지만 이 접근 방식에만 국한되는 것은 아닙니다. IoT 디바이스의 엣지에 Lookout for Vision을 배포하거나 일정에 따라 배치 프로세스로 실행하여 예측을 생성할 수도 있습니다.

도구

서비스

- [CodeBuildAWS](#)는 소스 코드를 컴파일하고, 단위 테스트를 실행하고, 배포할 준비가 된 아티팩트를 생성하는 데 도움이 되는 완전관리형 빌드 서비스입니다.
- [CodeCommitAWS](#)는 자체 소스 제어 시스템을 관리할 필요 없이 Git 리포지토리를 비공개로 저장하고 관리할 수 있는 버전 제어 서비스입니다.
- [AWS](#)는 소프트웨어 릴리스의 여러 단계를 신속하게 모델링 및 구성하고 소프트웨어 변경 사항을 지속적으로 릴리스하는 CodePipeline 데 필요한 단계를 자동화하도록 지원합니다.
- [AWS Key Management Service\(AWS KMS\)](#)를 사용하면 암호화 키를 생성하고 제어하여 데이터를 보호할 수 있습니다.
- [Lambda](#)는 서버를 프로비저닝하거나 관리할 필요 없이 코드를 실행할 수 있도록 도와주는 컴퓨팅 서비스입니다. 필요할 때만 코드를 실행하며 자동으로 확장이 가능하므로 사용한 컴퓨팅 시간만큼만 비용을 지불합니다.
- [Amazon Lookout for Vision](#)은 컴퓨터 비전을 사용하여 산업용 제품의 시각적 감지 결과를 대규모로 정확하고 정확하게 찾습니다.
- [Amazon Simple Storage Service\(S3\)](#)는 원하는 양의 데이터를 저장, 보호 및 검색하는 데 도움이 되는 클라우드 기반 객체 스토리지 서비스입니다.

코드 리포지토리

이 패턴의 코드는 [실리콘 웨이퍼 이상 탐지 리포지토리에 대한 Amazon Lookout for Vision 교육 및 배포 GitHub 자동화에서](#) 사용할 수 있습니다.

모범 사례

코드를 실험용으로 실행할 때는 [Amazon Lookout for Vision 엔드포인트를 중지해야](#) 합니다.

에픽

솔루션 배포

작업	설명	필요한 기술
GitHub 리포지토리를 복제하십시오.	<p>실리콘 웨이퍼 이상 탐지 리포지토리를 위한 GitHub 자동 Amazon Lookout for Vision 교육을 복제하여 로컬 워크스테이션에 배포하십시오.</p> <pre>git clone https://github.com/aws-samples/automated-silicon-wafer-anomaly-detection-using-amazon-lookout-for-vision.git</pre>	Bash
가상 환경을 생성합니다.	<p>다음 명령을 입력하여 로컬 워크스테이션에 가상 환경을 생성합니다.</p> <pre>python3 -m venv .venv</pre>	Python
종속 항목 설치	<p>가상 환경을 만든 후 다음 명령을 입력하여 필요한 종속성을 설치합니다.</p> <pre>pip install -r requirements.txt</pre>	Python
(Linux 사용자만 해당) 가상 환경을 활성화합니다.	<p>초기화가 완료되고 가상 환경이 생성되면 다음 명령을 사용하여 가상 환경을 활성화합니다.</p>	Bash

작업	설명	필요한 기술
(Windows 사용자만 해당) 가상 환경을 활성화합니다.	<pre>source .venv/bin/activate</pre> <p>초기화가 완료되고 가상 환경이 생성되면 다음 명령을 사용하여 가상 환경을 활성화합니다.</p> <pre>.venv\Scripts\activate.bat</pre>	PowerShell
스택을 배포합니다.	<ol style="list-style-type: none"> 1. AWS CDK CLI에서 다음 명령을 입력하여 AWS 템플릿을 합성합니다. CloudFormation <pre>cdk synth</pre> 2. 다음 명령을 입력하여 스택을 배포합니다. CloudFormation <pre>cdk deploy --all --require-approval never</pre> <p>모든 구성 요소가 한 번에 <code>--all</code> flag 설치되도록 합니다. <code>--require-approval</code> 따라서 각 구성 요소 배포를 승인할 필요가 전혀 없습니다.</p>	AWS 관리자

솔루션 테스트

작업	설명	필요한 기술
예제 테스트 이벤트를 입력합니다.	<ol style="list-style-type: none"> Lambda 콘솔의 함수 페이지를 엽니다. amazon-lookout-for-vision-project-lambda 함수를 선택합니다. 테스트 탭을 선택합니다. 테스트 이벤트에서 새 이벤트 만들기를 선택합니다. 다음을 입력합니다. 테스트(Test)를 선택합니다. <pre>{ "tbd": "tbd" }</pre> <ol style="list-style-type: none"> 테스트 결과를 확인하려면 실행 결과(Execution result)에서 세부 정보(Details)를 확장합니다. 	일반 AWS

관련 리소스

AWS 설명서

- [아마존 룩아웃 포 비전 시작하기](#)
- [AWS CDK 시작하기](#)

AWS 블로그 게시물

- [Python SDK를 사용하여 Amazon Lookout for Vision 모델을 구축, 교육 및 배포하십시오.](#)
- [Amazon Lookout for Vision을 사용하여 이벤트 기반 추적 솔루션을 구축하십시오.](#)
- [Amazon Lookout for Vision Python SDK: 교차 검증 및 다른 AWS 서비스와의 통합](#)

Amazon Textract를 사용하여 PDF 파일에서 콘텐츠 자동 추출하기

작성자: Tianxia Jia (AWS)

환경: 프로덕션

기술: 기계 학습 및 AI, 분석,
빅 데이터

AWS 서비스: 아마존 S3,
아마존 텍스트트랙, 아마존
SageMaker

요약

많은 조직에서는 비즈니스 애플리케이션에 업로드되는 PDF 파일에서 정보를 추출해야 합니다. 예를 들어, 조직은 세금 분석이나 의료 청구 처리를 위해 세금 또는 의료 PDF 파일에서 정보를 정확하게 추출해야 할 수 있습니다.

Amazon Web Services(AWS) 클라우드에서 Amazon Textract는 PDF 파일에서 정보(예: 인쇄된 텍스트, 양식 및 표)를 자동으로 추출하고 소스 PDF 파일의 정보가 포함된 JSON 형식의 파일을 생성합니다. AWS 관리 콘솔에서 또는 API 호출을 구현하여 Amazon Textract를 사용할 수 있습니다. [프로그래밍 방식의 API 호출](#)을 사용하여 많은 수의 PDF 파일을 확장하고 자동으로 처리하는 것을 권장합니다.

Amazon Textract는 파일을 처리할 때 페이지, 텍스트 줄 및 단어, 양식(키-값 쌍), 테이블 및 셀, 선택 요소 등의 BLock 객체 목록을 생성합니다. [경계 상자](#), 신뢰 구간, ID 및 관계와 같은 다른 객체 정보도 포함됩니다. Amazon Textract는 콘텐츠 정보를 문자열로 추출합니다. 다운스트림 애플리케이션에서 더 쉽게 사용할 수 있으므로 데이터 값을 정확하게 식별하고 변환해야 합니다.

이 패턴은 Amazon Textract를 사용하여 PDF 파일에서 콘텐츠를 자동으로 추출하고 깨끗한 출력으로 처리하는 step-by-step 워크플로를 설명합니다. 이 패턴은 템플릿 매칭 기법을 사용하여 필수 필드, 키 이름 및 테이블을 정확하게 식별한 후 각 데이터 유형에 사후 처리 수정 사항을 적용합니다. 이 패턴을 사용하여 다양한 유형의 PDF 파일을 처리한 다음 이 워크플로우를 확장 및 자동화하여 형식이 동일한 PDF 파일을 처리할 수 있습니다.

사전 조건 및 제한 사항

사전 조건

- 활성 상태의 AWS 계정.

- Amazon Textract에서 처리할 수 있도록 PDF 파일을 JPEG 형식으로 변환한 후 해당 파일을 저장할 기존 Amazon Simple S3(S3) 버킷입니다. S3 버킷에 대한 자세한 내용은 Amazon S3 설명서의 [버킷 개요](#)를 참조하세요.
- Textract_PostProcessing.ipynb Jupyter Notebook (첨부됨), 설치 및 구성됨. Jupyter 노트북에 대한 자세한 내용은 Amazon 설명서의 [Jupyter 노트북 만들기를](#) 참조하십시오. SageMaker
- 형식이 동일한 기존 PDF 파일.
- Python에 대한 이해.

제한 사항

- PDF 파일은 품질이 좋고 읽기 쉬워야 합니다. 기본 PDF 파일을 사용하는 것이 좋지만, 개별 단어가 모두 명확하다면 PDF 형식으로 변환된 스캔 문서를 사용할 수 있습니다. 이에 대한 자세한 내용은 AWS Machine Learning 블로그에서 [Amazon Textract를 사용한 PDF 문서 사전 처리: 시각적 탐지 및 제거](#)를 참고하십시오.
- 여러 페이지 파일의 경우 비동기 작업을 사용하거나 PDF 파일을 단일 페이지로 분할하고 동기 작업을 사용할 수 있습니다. 이 두 옵션에 대한 자세한 내용은 Amazon Textract 설명서의 [여러 페이지 문서에서 텍스트 감지 및 분석 및 단일 페이지 문서의 텍스트 감지 및 분석](#)을 참고하십시오.

아키텍처

이 패턴의 워크플로우는 먼저 샘플 PDF 파일에서 Amazon Textract를 실행한(처음 실행) 후에 첫 번째 PDF와 형식이 동일한 PDF 파일에서 실행(반복 실행) 합니다. 다음 다이어그램은 형식이 동일한 PDF 파일에서 콘텐츠를 자동으로 반복적으로 추출하는 최초 실행 및 반복 실행 워크플로우를 함께 보여줍니다.

이 다이어그램은 이 패턴에 대해 다음 워크플로를 보여 줍니다.

1. PDF 파일을 JPEG 형식으로 변환하고 S3 버킷에 저장합니다.
2. Amazon Textract API를 호출하여 Amazon Textract 응답 JSON 파일을 파싱합니다.
3. 각 필수 필드에 올바른 KeyName:DataType 쌍을 추가하여 JSON 파일을 편집하십시오. 반복 실행 단계를 위한 TemplateJSON 파일을 생성합니다.
4. 각 데이터 유형(예: 부동 소수점, 정수, 날짜)에 대한 사후 처리 수정 함수를 정의합니다.
5. 첫 번째 PDF 파일과 형식이 동일한 PDF 파일을 준비합니다.
6. Amazon Textract API를 호출하고 Amazon Textract 응답 JSON을 파싱합니다.

7. 파싱된 JSON 파일을 TemplateJSON 파일과 일치시킵니다.
8. 사후 처리 수정을 구현합니다.

최종 JSON 출력 파일에는 각 필수 필드에 대한 올바른 KeyName 및 Value이 있습니다.

대상 기술 스택

- 아마존 SageMaker
- Amazon S3
- Amazon Textract

자동화 및 규모 조정

Amazon S3에 새 PDF 파일이 추가될 때 Amazon Textract를 시작하는 AWS Lambda 함수를 사용하여 반복 실행 워크플로우를 자동화할 수 있습니다. 그러면 Amazon Textract가 처리 스크립트를 실행하고 최종 출력을 스토리지 위치에 저장할 수 있습니다. 이에 대한 자세한 내용은 Lambda 설명서에서 [Amazon S3 트리거를 사용하여 Lambda 함수 호출](#)을 참고하십시오.

도구

- [SageMakerAmazon](#)은 ML 모델을 쉽고 빠르게 구축 및 교육한 다음 프로덕션 준비가 완료된 호스팅 환경에 직접 배포할 수 있도록 도와주는 완전 관리형 ML 서비스입니다.
- [Amazon Simple Storage Service\(S3\)](#)는 원하는 양의 데이터를 저장, 보호 및 검색하는 데 도움이 되는 클라우드 기반 객체 스토리지 서비스입니다.
- [Amazon Textract](#)를 사용하면 애플리케이션에 문서 텍스트 감지 및 분석을 쉽게 추가할 수 있습니다.

에픽

최초 실행

작업	설명	필요한 기술
PDF 파일을 변환합니다.	PDF 파일을 단일 페이지로 분할하고 Amazon Textract 동기 작업 (Syn API)을 위한 JPEG 형식으로 변환하여 처음 실행	데이터 사이언티스트, 개발자

작업	설명	필요한 기술
	<p>할 때 사용할 수 있도록 준비하십시오.</p> <p>참고: 여러 페이지로 구성된 PDF 파일에도 Amazon Textract의 비동기 작업(Asyn API)을 사용할 수 있습니다.</p>	
<p>Amazon Textract의 응답 JSON을 파싱합니다.</p>	<p>Textract_PostProcessing.ipynb Jupyter Notebook(첨부됨)을 열고 다음 코드를 사용하여 Amazon Textract API를 호출합니다.</p> <pre data-bbox="597 827 1026 1381"> response = textract. analyze_document(Document={ 'S3Object': { 'Bucket': BUCKET, 'Name': '{}'.format(filename) } }, FeatureTypes= ["TABLES", "FORMS"]) </pre> <p>다음 코드를 사용하여 응답 JSON을 양식 및 테이블로 구분 분석합니다.</p> <pre data-bbox="597 1591 1026 1822"> parseformKV=form_kv_from_JSON(response) parseformTables=get_tables_fromJSON(response) </pre>	<p>데이터 사이언티스트, 개발자</p>

작업	설명	필요한 기술
<p>TemplateJSON 파일을 편집합니다.</p>	<p>각 KeyName 및 해당 DataType(예: 문자열, 부동 소수점, 정수 또는 날짜) 및 테이블 헤더(예: ColumnNames 및 RowNames)에 대해 구문 분석된 JSON을 편집합니다.</p> <p>이 템플릿은 각 개별 PDF 파일 유형에 사용되므로 형식이 동일한 PDF 파일에 템플릿을 재사용할 수 있습니다.</p>	<p>데이터 사이언티스트, 개발자</p>
<p>사후 처리 수정 함수를 정의합니다.</p>	<p>TemplateJSON 파일에 대한 Amazon Textract의 응답 값은 문자열입니다. 날짜, 부동 소수점, 정수 또는 통화에는 차이가 없습니다. 이러한 값은 다운스트림 사용 사례에 맞는 올바른 데이터 유형으로 변환되어야 합니다.</p> <p>다음 코드를 사용하여 TemplateJSON 파일에 따라 각 데이터 유형을 수정하십시오.</p> <pre data-bbox="597 1402 1026 1596">finalJSON=postprocessingCorrection(parsedJSON,templateJSON)</pre>	<p>데이터 사이언티스트, 개발자</p>

반복 실행

작업	설명	필요한 기술
PDF 파일을 준비합니다.	<p>PDF 파일을 단일 페이지로 분할하고 Amazon Textract 동기 작업(Syn API)을 위한 JPEG 형식으로 변환하여 준비하세요.</p> <p>참고: 여러 페이지로 구성된 PDF 파일에도 Amazon Textract의 비동기 작업(Asyn API)을 사용할 수 있습니다.</p>	데이터 사이언티스트, 개발자
Amazon Textract API를 호출합니다.	<p>다음 코드를 사용하여 Amazon Textract API를 호출합니다.</p> <pre data-bbox="597 947 1027 1503"> response = textract. analyze_document(Document={ 'S3Object': { 'Bucket': BUCKET, 'Name': '{}'.format(filename) } }, FeatureTy pes=["TABLES", "FORMS"]) </pre>	데이터 사이언티스트, 개발자
Amazon Textract의 응답 JSON을 파싱합니다.	<p>다음 코드를 사용하여 응답 JSON을 양식 및 테이블로 구분 분석합니다.</p> <pre data-bbox="597 1709 1027 1803"> parseformKV=form_k v_from_JSON(response) </pre>	데이터 사이언티스트, 개발자

작업	설명	필요한 기술
	<pre>parseformTables=get_tables_fromJSON(response)</pre>	
<p>TemplateJSON 파일을 로드하고 파싱된 JSON과 일치시킵니다.</p>	<p>다음 명령을 사용하여 TemplateJSON 파일을 사용하여 올바른 키-값 쌍과 테이블을 추출합니다.</p> <pre>form_kv_corrected=form_kv_correction(parseformKV,templateJSON) form_table_corrected=form_Table_correction(parseformTables,templateJSON) form_kv_table_corrected_final={**form_kv_corrected, **form_table_corrected}</pre>	<p>데이터 사이언티스트, 개발자</p>
<p>사후 처리 수정.</p>	<p>TemplateJSON 파일의 DataType 및 사후 처리 함수에서 다음 코드를 사용하여 데이터를 수정할 수 있습니다.</p> <pre>finalJSON=postprocessingCorrection(form_kv_table_corrected_final,templateJSON)</pre>	<p>데이터 사이언티스트, 개발자</p>

관련 리소스

- [Amazon Textract를 사용하여 문서에서 텍스트 및 정형 데이터를 자동으로 추출하기](#)

- [Amazon Textract를 사용하여 텍스트 및 정형 데이터 추출하기](#)
- [Amazon Textract 리소스](#)

첨부

이 문서와 관련된 추가 콘텐츠에 액세스하려면 [attachment.zip](#) 파일의 압축을 풉니다.

SageMaker Amazon과 Azure를 사용하여 MLOps 워크플로를 구축하세요. DevOps

디피카 쿠마르 (AWS) 와 사라 반 드 무스디크 (AWS) 가 제작했습니다.

환경: 프로덕션	기술: 기계 학습 및 AI, 운영 DevOps	워크로드: Microsoft
AWS 서비스: 아마존 API 게이트웨이, 아마존 ECR, 아마존, AWS Lambda EventBridge, 아마존 SageMaker		

요약

기계 학습 운영 (MLOps) 은 기계 학습 (ML) 워크플로 및 배포를 자동화하고 간소화하는 일련의 관행입니다. MLOps는 ML 라이프사이클 자동화에 중점을 둡니다. 이를 통해 모델을 개발하는 데 그치지 않고 체계적이고 반복적으로 배포, 모니터링 및 재학습할 수 있습니다. 이는 ML에 DevOps 원칙을 적용합니다. MLOps를 사용하면 ML 모델을 더 빠르게 배포하고 시간이 지남에 따라 정확도가 향상되며 실제 비즈니스 가치를 제공한다는 확신을 강화할 수 있습니다.

조직은 MLOps 여정을 시작하기 전에 기존 DevOps 도구와 데이터 스토리지 솔루션을 보유하고 있는 경우가 많습니다. 이 패턴은 Microsoft Azure와 AWS의 장점을 모두 활용하는 방법을 보여줍니다. Azure를 DevOps Amazon과 통합하여 MLOps SageMaker 워크플로를 만드는 데 도움이 됩니다.

이 솔루션은 Azure와 AWS 간의 작업을 간소화합니다. 개발에는 Azure를 사용하고 기계 학습에는 AWS를 사용할 수 있습니다. AWS에서의 데이터 처리, 교육 및 배포를 포함하여 처음부터 끝까지 기계 학습 모델을 만드는 효과적인 프로세스를 촉진합니다. 효율성을 위해 Azure DevOps 파이프라인을 통해 이러한 프로세스를 관리합니다.

사전 조건 및 제한 사항

사전 조건

- Azure 구독 — 지속적 통합 및 지속적 배포 (CI/CD) 파이프라인을 설정하기 위한 Azure와 같은 Azure DevOps 서비스에 액세스할 수 있습니다.
- 활성 AWS 계정 — 이 패턴에 사용된 AWS 서비스를 사용할 수 있는 권한.

- 데이터 — 기계 학습 모델 교육을 위한 과거 데이터에 액세스합니다.
- ML 개념에 대한 이해 — Python, Jupyter 노트북 및 기계 학습 모델 개발에 대한 이해
- 보안 구성 — Azure와 AWS 모두에서 역할, 정책 및 권한을 적절히 구성하여 안전한 데이터 전송 및 액세스를 보장합니다.

제한 사항

- 이 지침은 안전한 클라우드 간 데이터 전송에 대한 지침을 제공하지 않습니다. 클라우드 간 데이터 전송에 대한 자세한 내용은 [하이브리드 및 멀티클라우드용 AWS 솔루션](#)을 참조하십시오.
- 멀티클라우드 솔루션은 실시간 데이터 처리 및 모델 추론의 지연 시간을 증가시킬 수 있습니다.
- 이 지침은 다중 계정 MLOps 아키텍처의 한 가지 예를 제공합니다. 기계 학습과 AWS 전략에 따라 조정이 필요합니다.

아키텍처

대상 아키텍처

대상 아키텍처는 Azure와 DevOps SageMaker Amazon을 통합하여 클라우드 간 ML 워크플로를 생성합니다. CI/CD 프로세스와 ML 모델 교육 및 SageMaker 배포에 Azure를 사용합니다. 모델 구축 및 배포를 통해 Amazon S3, Snowflake 및 Azure Data Lake와 같은 소스에서 데이터를 얻는 프로세스를 설명합니다. 주요 구성 요소에는 모델 구축 및 배포, 데이터 준비, 인프라 관리를 위한 CI/CD 파이프라인, ML 모델의 교육, 평가 및 배포를 SageMaker 위한 Amazon이 포함됩니다. 이 아키텍처는 클라우드 플랫폼 전반에서 효율적이고 자동화되고 확장 가능한 ML 워크플로를 제공하도록 설계되었습니다.

아키텍처는 다음과 같은 구성 요소로 구성되어 있습니다.

1. 데이터 과학자는 개발 계정에서 ML 실험을 수행하여 다양한 데이터 소스를 사용하여 ML 사용 사례에 대한 다양한 접근 방식을 탐색합니다. 데이터 사이언티스트는 단위 테스트 및 시험을 수행합니다. 모델 평가 후 데이터 과학자는 코드를 DevOps Azure에서 호스팅되는 Model Build 리포지토리로 푸시하고 병합합니다. 이 리포지토리에는 다단계 모델 구축 파이프라인을 위한 코드가 포함되어 있습니다.
2. DevOpsAzure에서는 코드를 기본 브랜치에 병합할 때 지속적 통합 (CI) 을 제공하는 모델 빌드 파이프라인을 자동 또는 수동으로 활성화할 수 있습니다. 자동화 계정에서 이렇게 하면 데이터 전처리, 모델 학습 및 평가, 정확도에 기반한 조건부 모델 등록을 위한 SageMaker 파이프라인이 활성화됩니다.

3. 자동화 계정은 ML 환경 (Amazon ECR), 모델 (Amazon S3), 모델 메타데이터 (SageMaker 모델 레지스트리), 기능 (SageMaker 기능 스토어), 자동화된 파이프라인 (파이프라인), ML 로그 인사이트 (SageMaker CloudWatch 및 OpenSearch 서비스) 를 호스팅하는 ML 플랫폼 전반의 중앙 계정입니다. 이 계정을 사용하면 ML 자산을 재사용할 수 있으며 모범 사례를 적용하여 ML 사용 사례 제공을 가속화할 수 있습니다.
4. 검토를 위해 최신 모델 버전이 SageMaker 모델 레지스트리에 추가됩니다. 모델 버전과 각 아티팩트 (계보 및 메타데이터) 를 추적합니다. 또한 모델 상태 (승인, 거부 또는 보류 중) 를 관리하고 다운스트림 배포를 위한 버전을 관리합니다.
5. Model Registry의 학습된 모델이 스튜디오 인터페이스 또는 API 호출을 통해 승인되면 EventBridge Amazon에 이벤트를 발송할 수 있습니다. EventBridge DevOpsAzure에서 모델 배포 파이프라인을 시작합니다.
6. 지속적 배포 (CD) 를 제공하는 모델 배포 파이프라인은 모델 배포 리포지토리에서 소스를 체크아웃합니다. 소스에는 코드, 모델 배포를 위한 구성, 품질 벤치마크를 위한 테스트 스크립트가 포함되어 있습니다. 모델 배포 파이프라인은 추론 유형에 맞게 조정할 수 있습니다.
7. 품질 관리 검사 후 모델 배포 파이프라인은 스테이징 계정에 모델을 배포합니다. 스테이징 계정은 프로덕션 계정의 사본이며 통합 테스트 및 평가에 사용됩니다. 일괄 변환의 경우 Model Deploy 파이프라인은 승인된 최신 모델 버전을 사용하도록 배치 추론 프로세스를 자동으로 업데이트할 수 있습니다. 실시간, 서버리스 또는 비동기 추론의 경우 해당 모델 엔드포인트를 설정하거나 업데이트합니다.
8. 스테이징 계정에서 성공적으로 테스트한 후에는 모델 배포 파이프라인을 통한 수동 승인을 통해 프로덕션 계정에 모델을 배포할 수 있습니다. 이 파이프라인은 배포 단계에서 모델 모니터링 및 데이터 피드백 메커니즘을 포함한 프로덕션 엔드포인트를 제공합니다.
9. 모델이 프로덕션 단계에 들어간 후에는 Model Monitor 및 SageMaker Clarify와 같은 SageMaker 도구를 사용하여 편향을 식별하고, 드리프트를 감지하고, 모델 성능을 지속적으로 모니터링하세요.

자동화 및 규모 조정

IaC (코드형 인프라) 를 사용하여 여러 계정 및 환경에 자동으로 배포할 수 있습니다. MLOps 워크플로 설정 프로세스를 자동화하면 여러 프로젝트를 진행하는 ML 팀이 사용하는 환경을 분리할 수 있습니다. [AWS](#)는 인프라를 코드로 취급하여 AWS 리소스를 모델링, 프로비저닝 및 관리할 수 있도록 CloudFormation 지원합니다.

도구

서비스

- [SageMakerAmazon](#)은 ML 모델을 구축 및 교육한 다음 프로덕션 준비가 완료된 호스팅 환경에 배포할 수 있도록 지원하는 관리형 ML 서비스입니다.
- [AWS Glue](#)는 완전 관리형 추출, 전환, 적재(ETL) 서비스입니다. 이를 통해 데이터 스토어와 데이터 스트림 간에 데이터를 안정적으로 분류, 정리, 보강하고 이동할 수 있습니다.
- [Amazon Simple Storage Service\(S3\)](#)는 원하는 양의 데이터를 저장, 보호 및 검색하는 데 도움이 되는 클라우드 기반 객체 스토리지 서비스입니다. 이 패턴에서 Amazon S3는 데이터 스토리지에 사용되고 모델 교육 및 모델 SageMaker 객체에 통합됩니다.
- [AWS Lambda](#)는 서버를 프로비저닝하거나 관리할 필요 없이 코드를 실행하는 데 도움이 되는 컴퓨팅 서비스입니다. 필요할 때만 코드를 실행하며 자동으로 확장이 가능하므로 사용한 컴퓨팅 시간만큼만 비용을 지불합니다. 이 패턴에서 Lambda는 데이터 사전 처리 및 사후 처리 작업에 사용됩니다.
- [Amazon Elastic Container Registry\(Amazon ECR\)](#)는 안전하고 확장성이 있고 신뢰할 수 있는 관리형 컨테이너 이미지 레지스트리 서비스입니다. 이 패턴에서는 교육 및 배포 환경으로 SageMaker 사용하는 Docker 컨테이너를 저장합니다.
- [EventBridgeAmazon](#)은 다양한 소스의 실시간 데이터와 애플리케이션을 연결하는 데 도움이 되는 서버리스 이벤트 버스 서비스입니다. 이 패턴에서는 자동 모델 재교육 또는 배포를 시작하는 이벤트 기반 또는 시간 기반 워크플로를 EventBridge 오케스트레이션합니다.
- [Amazon API Gateway](#)를 사용하면 어떤 규모에서든 REST, HTTP 및 WebSocket API를 생성, 게시, 유지 관리, 모니터링 및 보호할 수 있습니다. 이 패턴에서는 Amazon 엔드포인트에 대한 외부 대상 단일 진입 지점을 생성하는 데 사용됩니다. SageMaker

기타 도구

- [DevOpsAzure](#)는 CI/CD 파이프라인을 관리하고 코드 빌드, 테스트 및 배포를 용이하게 하는 데 도움이 됩니다.
- [Azure 데이터 레이크 스토리지 또는 Snowflake](#)는 ML 모델을 위한 교육 데이터의 타사 소스일 수 있습니다.

모범 사례

이 멀티클라우드 MLOps 워크플로의 구성 요소를 구현하기 전에 다음 활동을 완료하세요.

- 기계 학습 워크플로와 이를 지원하는 데 필요한 도구를 정의하고 이해하십시오. 사용 사례에 따라 필요한 워크플로와 구성 요소도 다릅니다. 예를 들어 개인화 사용 사례에서 기능 재사용과 지연 시간이 짧은 추론에는 기능 저장소가 필요할 수 있지만 다른 사용 사례에는 필요하지 않을 수 있습니다. 아

키텍처를 성공적으로 사용자 지정하려면 대상 워크플로, 사용 사례 요구 사항, 데이터 과학 팀이 선호하는 협업 방법을 이해해야 합니다.

- 아키텍처의 각 구성 요소에 대한 책임을 명확히 구분하세요. Azure 데이터 레이크 스토리지, Snowflake 및 Amazon S3에 데이터 스토리지를 분산하면 복잡성과 비용이 증가할 수 있습니다. 가능하다면 일관된 스토리지 메커니즘을 선택하세요. 마찬가지로 Azure와 AWS DevOps 서비스를 함께 사용하거나 Azure와 AWS ML 서비스를 함께 사용하지 마십시오.
- MLOps 워크플로를 end-to-end 테스트할 기존 모델 및 데이터세트를 하나 이상 선택합니다. 테스트 아티팩트는 플랫폼이 프로덕션에 들어갈 때 데이터 과학 팀이 개발하는 실제 사용 사례를 반영해야 합니다.

에픽

MLOps 아키텍처를 설계하세요.

작업	설명	필요한 기술
데이터 소스 식별.	현재 및 미래의 사용 사례, 사용 가능한 데이터 소스, 데이터 유형 (예: 기밀 데이터) 을 기반으로 MLOps 플랫폼과 통합해야 하는 데이터 소스를 문서화합니다. 데이터는 Amazon S3, Azure 데이터 레이크 스토리지, 스노우플레이크 또는 기타 소스에 저장할 수 있습니다. 이러한 소스를 플랫폼과 통합하고 올바른 리소스에 대한 액세스를 보호하기 위한 계획을 세우십시오.	데이터 엔지니어, 데이터 사이언티스트, 클라우드 아키텍트
해당 서비스를 선택하세요.	데이터 사이언스 팀의 원하는 워크플로, 적용 가능한 데이터 소스 및 기존 클라우드 아키텍처를 기반으로 서비스를 추가하거나 제거하여 아키텍처를 사용자 지정합니다. 예를	AWS 관리자, 데이터 엔지니어, 데이터 사이언티스트, ML 엔지니어

작업	설명	필요한 기술
	<p>들어, 데이터 엔지니어와 데이터 과학자는 AWS Glue 또는 Amazon EMR에서 SageMaker 데이터 전처리 및 기능 엔지니어링을 수행할 수 있습니다. 세 가지 서비스가 모두 필요할 것 같지는 않습니다.</p>	
<p>보안 요구 사항 분석.</p>	<p>보안 요구 사항을 수집하고 문서화하세요. 여기에는 다음 사항을 결정하는 것이 포함됩니다.</p> <ul style="list-style-type: none"> • 특정 데이터 소스에 액세스할 수 있는 팀 또는 엔지니어 • 팀이 다른 팀의 코드와 모델에 액세스할 수 있는지 여부 • 팀원이 비개발 계정에 대해 가져야 하는 권한 (있는 경우) • 클라우드 간 데이터 전송을 위해 구현해야 하는 보안 조치는 무엇입니까? 	<p>AWS 관리자, 클라우드 아키텍트</p>

AWS 조직 설정

작업	설명	필요한 기술
<p>AWS 조직을 설정합니다.</p>	<p>루트 AWS 계정에 AWS Organizations를 설정합니다. 이를 통해 다중 계정 MLOps 전략의 일환으로 생성한 후속 계정을 관리할 수 있습니다. 자세</p>	<p>AWS 관리자</p>

작업	설명	필요한 기술
	한 내용은 AWS Organizations 설명서를 참조하십시오.	

개발 환경 및 버전 관리를 설정합니다.

작업	설명	필요한 기술
AWS 개발 계정을 생성합니다.	데이터 엔지니어와 데이터 과학자가 ML 모델을 실험하고 생성할 권한이 있는 AWS 계정을 생성하십시오. 지침은 AWS Organizations 설명서에서 조직 내 회원 계정 생성을 참조하십시오.	AWS 관리자
Model Build 리포지토리 생성	데이터 과학자가 실험 단계가 완료된 후 모델 빌드 및 배포 코드를 푸시할 수 있는 Azure에서 Git 리포지토리를 만드세요. 지침은 Azure DevOps 설명서의 Git 리포지토리 설정을 참조하십시오.	DevOps 엔지니어, ML 엔지니어
Model Deploy 리포지토리 생성	Azure에서 표준 배포 코드 및 템플릿을 저장하는 Git 리포지토리를 만드세요. 여기에는 설계 단계에서 식별한 대로 조직에서 사용하는 모든 배포 옵션에 대한 코드가 포함되어야 합니다. 예를 들어 여기에는 실시간 엔드포인트, 비동기 엔드포인트, 서버리스 추론 또는 일괄 변환이 포함되어야 합니다. 지침은 Azure DevOps 설명서의	DevOps 엔지니어, ML 엔지니어

작업	설명	필요한 기술
	Git 리포지토리 설정 을 참조하십시오.	
Amazon ECR 리포지토리를 생성합니다.	승인된 ML 환경을 Docker 이미지로 저장하는 Amazon ECR 리포지토리를 설정합니다. 데이터 과학자와 ML 엔지니어가 새로운 환경을 정의할 수 있도록 허용하십시오. 자세한 내용은 Amazon ECR 설명서의 프라이빗 리포지토리 생성 을 참조하십시오.	ML 엔지니어
SageMaker Studio를 설정하십시오.	이전에 정의된 보안 요구 사항 및 선호하는 데이터 과학 도구 (예: 선택한 통합 개발 환경 (IDE) 에 따라 개발 계정으로 SageMaker Studio를 설정합니다. 라이프사이클 구성을 사용하여 주요 기능의 설치를 자동화하고 데이터 과학자를 위한 통일된 개발 환경을 조성하십시오. 자세한 내용은 SageMaker 설명서의 Amazon SageMaker Studio 를 참조하십시오.	ML 엔지니어, 데이터 사이언티스트

CI/CD 파이프라인 통합

작업	설명	필요한 기술
자동화 계정을 만드세요.	자동화된 파이프라인과 작업이 실행되는 AWS 계정을 생성합니다. 데이터 사이언스 팀에 이 계정에 대한 읽기 액세스 권한	AWS 관리자

작업	설명	필요한 기술
	<p>을 부여할 수 있습니다. 지침은 AWS Organizations 설명서에서 조직 내 회원 계정 생성을 참조하십시오.</p>	
<p>모델 레지스트리를 설정하십시오.</p>	<p>자동화 계정에서 SageMaker 모델 레지스트리를 설정합니다. 이 레지스트리는 ML 모델의 메타데이터를 저장하고 특정 데이터 과학자 또는 팀 리더가 모델을 승인하거나 거부하는 데 도움이 됩니다. 자세한 내용은 설명서의 모델 레지스트리를 사용하여 모델 등록 및 배포를 SageMaker 참조하십시오.</p>	<p>ML 엔지니어</p>
<p>Model Build파이프라인을 생성하세요.</p>	<p>코드를 리포지토리로 푸시할 때 수동 또는 자동으로 시작되는 CI/CD 파이프라인을 Azure에서 만드세요. Model Build 파이프라인은 소스 코드를 확인하고 자동화 계정에서 SageMaker 파이프라인을 만들거나 업데이트해야 합니다. 파이프라인은 모델 레지스트리에 새 모델을 추가해야 합니다. 파이프라인 만들기에 대한 자세한 내용은 Azure Pipelines 설명서를 참조하세요.</p>	<p>DevOps 엔지니어, ML 엔지니어</p>

배포 스택 구축

작업	설명	필요한 기술
AWS 스테이징 및 배포 계정을 생성합니다.	ML 모델의 스테이징 및 배포를 위한 AWS 계정을 생성합니다. 프로덕션으로 이동하기 전에 스테이징에서 모델을 정확하게 테스트하려면 이러한 계정이 동일해야 합니다. 데이터 사이언스 팀에 스테이징 계정에 대한 읽기 권한을 부여할 수 있습니다. 지침은 AWS Organizations 설명서에서 조직 내 멤버 계정 생성 을 참조하십시오.	AWS 관리자
모델 모니터링을 위한 S3 버킷을 설정합니다.	Model Deploy파이프라인에서 생성한 배포된 모델에 대해 모델 모니터링을 활성화하려면 이 단계를 완료하세요. 입력 및 출력 데이터를 저장하기 위한 Amazon S3 버킷을 생성합니다. S3 버킷 생성에 대한 자세한 내용은 Amazon S3 설명서의 버킷 생성 을 참조하십시오. 자동 모델 모니터링 작업이 Automation 계정에서 실행되도록 계정 간 권한을 설정하십시오. 자세한 내용은 SageMaker 설명서의 데이터 및 모델 품질 모니터링 을 참조하십시오.	ML 엔지니어
Model Deploy파이프라인을 생성하세요.	모델 레지스트리에서 모델이 승인될 때 시작되는 CI/CD 파이프라인을 Azure에서 만드세요. 파이프라인은 소스 코드 및 모델 아티팩트를 확인하고, 스	DevOps 엔지니어, ML 엔지니어

작업	설명	필요한 기술
	<p>테이징 및 프로덕션 계정에 모델을 배포하기 위한 인프라 템플릿을 빌드하고, 스테이징 계정에 모델을 배포하고, 자동 테스트를 실행하고, 수동 승인을 기다리고, 승인된 모델을 프로덕션 계정에 배포해야 합니다. 파이프라인 생성에 대한 자세한 내용은 Azure Pipelines 설명서를 참조하세요.</p>	

(선택 사항) ML 환경 인프라 자동화

작업	설명	필요한 기술
<p>AWS CDK 또는 CloudFormation 템플릿을 구축하십시오.</p>	<p>자동으로 배포해야 하는 모든 환경을 위한 AWS 클라우드 개발 키트 (AWS CDK) 또는 AWS CloudFormation 템플릿을 정의하십시오. 여기에는 개발 환경, 자동화 환경, 스테이징 및 배포 환경이 포함될 수 있습니다. 자세한 내용은 AWS CDK 및 CloudFormation 설명서를 참조하십시오.</p>	<p>AWS DevOps</p>
<p>Infrastructure 파이프라인 생성.</p>	<p>인프라 배포를 위해 Azure에서 CI/CD 파이프라인을 만드세요. 관리자는 이 파이프라인을 시작하여 새 AWS 계정을 만들고 ML 팀에 필요한 환경을 설정할 수 있습니다.</p>	<p>DevOps 엔지니어</p>

문제 해결

문제	Solution
<p>불충분한 모니터링 및 드리프트 감지 — 모니터링이 부적절하면 모델 성능 문제 또는 데이터 드리프트를 감지하지 못할 수 있습니다.</p>	<p>Amazon CloudWatch, SageMaker Model Monitor, Clatify와 같은 도구를 사용하여 모니터링 프레임워크를 SageMaker 강화하십시오. 식별된 문제에 대해 즉각적인 조치를 취할 수 있도록 알림을 구성하십시오.</p>
<p>CI 파이프라인 트리거 오류 - 잘못된 구성으로 인해 코드 병합 시 Azure의 CI 파이프라인이 트리거되지 DevOps 애플리케이션을 사용할 수 있습니다.</p>	<p>Azure DevOps 프로젝트 설정을 확인하여 웹훅이 제대로 설정되고 올바른 엔드포인트를 가리키는지 확인하세요. SageMaker</p>
<p>거버넌스 — 중앙 자동화 계정이 ML 플랫폼 전반에 모범 사례를 적용하지 않아 워크플로가 일관되지 않을 수 있습니다.</p>	<p>자동화 계정 설정을 감사하여 모든 ML 환경 및 모델이 사전 정의된 모범 사례 및 정책을 준수하는지 확인하세요.</p>
<p>모델 레지스트리 승인 지연 — 이는 사람들이 검토하는 데 시간이 걸리거나 기술적인 문제로 인해 모델 확인 및 승인이 지연될 때 발생합니다.</p>	<p>승인 대기 중인 모델에 대해 이해 관계자에게 알리고 검토 프로세스를 간소화하는 알림 시스템을 구현하십시오.</p>
<p>모델 배포 이벤트 실패 — 모델 배포 파이프라인을 시작하기 위해 전달된 이벤트가 실패하여 배포가 지연될 수 있습니다.</p>	<p>Amazon에 Azure DevOps 파이프라인을 EventBridge 성공적으로 호출할 수 있는 올바른 권한 및 이벤트 패턴이 있는지 확인하십시오.</p>
<p>프로덕션 배포 병목 현상 — 수동 승인 프로세스로 인해 병목 현상이 발생하여 모델의 프로덕션 배포가 지연될 수 있습니다.</p>	<p>모델 배포 파이프라인 내에서 승인 워크플로를 최적화하여 시기적절한 검토와 명확한 커뮤니케이션 채널을 촉진하세요.</p>

관련 리소스

AWS 설명서

- [아마존 SageMaker 설명서](#)
- [Machine Learning 렌즈](#) (AWS 웹 아키텍처 프레임워크)
- [성공적인 MLOP를 위한 계획](#) (AWS Crepiptive Guidance)

기타 AWS 리소스

- [Amazon을 사용하는 기업을 위한 MLOps 재단 로드맵 \(SageMakerAWS 블로그 게시물\)](#)
- [AWS 서밋 ANZ 2022 - 아키텍트를 위한 End-to-end MLOps \(동영상\) YouTube](#)

Azure 설명서

- [애저 설명서 DevOps](#)
- [Azure 파이프라인 설명서](#)

AWS Step Functions에서 사용자 지정 Docker 컨테이너 이미지를 SageMaker 생성하여 모델 교육에 사용하십시오.

작성자: Julia Bluszcz(AWS), Aubrey Oosthuizen(AWS), Mohan Gowda Purushothama(AWS), Mateusz Zaremba(AWS)

환경: 프로덕션

기술: 기계 학습 및 AI, DevOps

AWS 서비스: 아마존 ECR, 아마존 SageMaker, AWS Step Functions

요약

이 패턴은 SageMaker Amazon용 Docker 컨테이너 이미지를 생성하고 이를 AWS Step Functions의 교육 모델로 사용하는 방법을 보여줍니다. 사용자 지정 알고리즘을 컨테이너에 패키징하면 프로그래밍 언어, 프레임워크 또는 종속성에 관계없이 SageMaker 환경에서 거의 모든 코드를 실행할 수 있습니다.

제공된 예제 SageMaker 노트북에서는 사용자 지정 Docker 컨테이너 이미지가 Amazon Elastic Container Registry (Amazon ECR)에 저장됩니다. 그러면 Step Functions는 Amazon ECR에 저장된 컨테이너를 사용하여 Python SageMaker 처리 스크립트를 실행합니다. 그리고 나서, 컨테이너는 Amazon Simple Storage Service(S3)로 모델을 내보냅니다.

사전 조건 및 제한 사항

사전 조건

- 활성 상태의 AWS 계정
- [Amazon S3 권한이 있는 SageMaker AWS 자격 증명 및 액세스 관리 \(IAM\) 역할](#)
- [Step Functions용 IAM 실행 역할](#)
- Python에 대한 지식
- Amazon SageMaker Python SDK에 대한 지식
- AWS Command Line Interface(AWS CLI)에 대한 지식
- AWS SDK for Python(Boto3)에 대한 지식
- Amazon ECR에 대한 지식
- Docker에 대한 지식

제품 버전

- AWS Step Functions Data Science SDK v2.3.0
- 아마존 SageMaker 파이썬 SDK v2.78.0

아키텍처

다음 다이어그램은 Step Functions의 Docker 컨테이너 이미지를 만든 다음 이를 교육 모델에 사용하는 예제 워크플로를 보여줍니다. SageMaker

이 다이어그램은 다음 워크플로를 보여줍니다.

1. 데이터 과학자나 DevOps 엔지니어는 SageMaker 노트북을 사용하여 사용자 지정 Docker 컨테이너 이미지를 생성합니다.
2. 데이터 과학자 또는 DevOps 엔지니어는 Docker 컨테이너 이미지를 사설 레지스트리에 있는 Amazon ECR 사설 리포지토리에 저장합니다.
3. 데이터 과학자 또는 DevOps 엔지니어는 Docker 컨테이너를 사용하여 Step Functions 워크플로우에서 Python SageMaker 처리 작업을 실행합니다.

자동화 및 규모 조정

이 패턴의 예제 SageMaker 노트북은 m1.m5.xlarge 노트북 인스턴스 유형을 사용합니다. 사용 사례에 맞게 인스턴스 유형을 변경할 수 있습니다. SageMaker 노트북 인스턴스 유형에 대한 자세한 내용은 [Amazon SageMaker 요금을](#) 참조하십시오.

기술 스택

- SageMaker
- Amazon ECR
- Step Functions

도구

- [Amazon Elastic Container Registry\(Amazon ECR\)](#)는 안전하고 확장 가능하고 신뢰할 수 있는 관리형 컨테이너 이미지 레지스트리 서비스입니다.

- [SageMakerAmazon](#)은 ML 모델을 구축 및 교육한 다음 프로덕션에 바로 사용할 수 있는 호스팅 환경에 배포하는 데 도움이 되는 관리형 기계 학습 (ML) 서비스입니다.
- [Amazon SageMaker Python SDK](#)는 기계 학습 모델을 교육하고 배포하기 위한 오픈 소스 라이브러리입니다. SageMaker
- [AWS Step Functions](#)는 AWS Lambda 함수와 기타 AWS 서비스를 결합할 수 있는 서버리스 오케스트레이션 서비스로, 비즈니스 크리티컬 애플리케이션을 구축합니다.
- [AWS Step Functions Data Science Python SDK](#)는 기계 학습 모델을 처리하고 게시하는 Step Functions 워크플로우를 쉽게 생성할 수 있는 오픈 소스 라이브러리입니다.

에픽

사용자 지정 Docker 컨테이너 이미지를 생성하여 Amazon ECR에 저장합니다.

작업	설명	필요한 기술
Amazon ECR을 설정하고 새 프라이빗 레지스트리를 생성합니다.	아직 설정하지 않았다면 Amazon ECR 사용 설명서의 Amazon ECR로 설정 에 나와 있는 지침에 따라 Amazon ECR을 설정합니다. 각 AWS 계정은 기본 프라이빗 Amazon ECR 레지스트리와 함께 제공됩니다.	DevOps 엔지니어
Amazon ECR 프라이빗 리포지토리를 생성합니다.	Amazon ECR 사용 설명서의 프라이빗 리포지토리 생성 의 지침을 따릅니다. 참고: 생성한 리포지토리는 사용자 지정 Docker 컨테이너 이미지를 저장하는 곳입니다.	DevOps 엔지니어
처리 작업을 실행하는 데 필요한 사양이 포함된 Dockerfile을 만드세요. SageMaker	Dockerfile을 구성하여 SageMaker 처리 작업을 실행하는 데 필요한 사양이 포함된 Dockerfile을 생성하십시오. 지침은 Amazon SageMaker 개발	DevOps 엔지니어

작업	설명	필요한 기술
	<p>자 안내서의 자체 교육 컨테이너 조정을 참조하십시오.</p> <p>Dockerfile에 대한 자세한 내용은 Docker 설명서의 Dockerfile 참조를 참조하세요.</p> <p>Dockerfile을 생성하기 위한 Jupyter Notebook 코드 셀의 예</p> <p>셀 1</p> <pre data-bbox="597 709 1026 831"># Make docker folder !mkdir -p docker</pre> <p>셀 2</p> <pre data-bbox="597 940 1026 1495">%writefile docker/Dockerfile FROM python:3.7-slim-buster RUN pip3 install pandas==0.25.3 scikit-learn==0.21.3 ENV PYTHONUNBUFFERED=TRUE ENTRYPOINT ["python3"]</pre>	

작업	설명	필요한 기술
<p>Docker 컨테이너 이미지를 구축하고 Amazon ECR에 푸시합니다.</p>	<ol style="list-style-type: none"> 1. AWS CLI에서 <code>docker build</code> 명령을 실행하여 생성한 Dockerfile을 사용하여 컨테이너 이미지를 빌드합니다. 2. <code>docker push</code> 명령을 실행하여 컨테이너 이미지를 Amazon ECR로 푸시합니다. <p>자세한 내용은 자체 알고리즘 컨테이너 빌드에서 컨테이너 빌드 및 등록을 참조하십시오 GitHub.</p> <p>Docker 이미지를 빌드하고 등록하기 위한 Jupyter Notebook 코드 셀의 예</p> <p>중요: 다음 셀을 실행하기 전에 Dockerfile을 생성하여 <code>docker</code>라는 디렉터리에 저장했는지 확인합니다. 또한 Amazon ECR 리포지토리를 생성했는지 확인하고 첫 번째 셀의 <code>ecr_repository</code> 값을 리포지토리 이름으로 바꿔야 합니다.</p> <p>셀 1</p> <pre>import boto3 tag = ':latest' account_id = boto3.client('sts').get_cal</pre>	<p>DevOps 엔지니어</p>

작업	설명	필요한 기술
	<pre> ler_identity().get ('Account') region = boto3.Ses sion().region_name ecr_repository = 'byoc' image_uri = '{}.dkr.e cr.{}.amazonaws.com/ {}'.format(account_id, region, ecr_repository + tag) </pre> <p>셀 2</p> <pre> # Build docker image !docker build -t \$image_uri docker </pre> <p>셀 3</p> <pre> # Authenticate to ECR !aws ecr get-login -password --region {region} docker login --username AWS --password-stdin {account_id}.dkr.ecr. {region}.amazonaws.com </pre> <p>셀 4</p> <pre> # Push docker image !docker push \$image_ur i </pre> <p>참고: docker push 및 docker pull 명령을 사용할 수 있도록 Docker 클라이언트</p>	

작업	설명	필요한 기술
	<p>를 프라이빗 레지스트리에 인 증해야 합니다. 이러한 명령은 레지스트리의 리포지토리로 이미지를 푸시하거나 가져옵니다.</p>	

사용자 지정 Docker 컨테이너 이미지를 사용하는 Step Functions 워크플로우를 생성합니다.

작업	설명	필요한 기술
<p>사용자 지정 처리 및 모델 학습 로직이 포함된 Python 스크립트를 생성합니다.</p>	<p>데이터 처리 스크립트에서 실행할 사용자 지정 처리 로직을 작성합니다. 그런 다음 이라는 이름이 training.py 인 Python 스크립트로 저장합니다.</p> <p>자세한 내용은 SageMaker 스크립트 모드가 켜진 상태에서 자체 모델 가져오기를 참조하십시오 GitHub.</p> <p>사용자 지정 처리 및 모델 학습 로직이 포함된 예제 Python 스크립트</p> <pre data-bbox="594 1430 1029 1881">%%writefile training.py from numpy import empty import pandas as pd import os from sklearn import datasets, svm from joblib import dump, load</pre>	<p>데이터 사이언티스트</p>

작업	설명	필요한 기술
	<pre>if __name__ == '__main__': digits = datasets. load_digits() #create classifier object clf = svm.SVC(g amma=0.001, C=100.) #fit the model clf.fit(digits.dat a[: -1], digits.ta rget[: -1]) #model output in binary format output_path = os.path.join('/opt/ ml/processing/model', "model.joblib") dump(clf, output_pa th)</pre>	

작업	설명	필요한 기술
<p>SageMaker 처리 작업을 단계 중 하나로 포함하는 Step Functions 워크플로를 생성합니다.</p>	<p>AWS Step Functions 데이터 과학 SDK를 설치 및 가져오고 training.py 파일을 Amazon S3에 업로드합니다. 그런 다음 Amazon SageMaker Python SDK를 사용하여 Step Functions에서 처리 단계를 정의합니다.</p> <p>중요: AWS 계정에서 Step Functions에 대한 IAM 실행 역할을 생성했는지 확인합니다.</p> <p>Amazon S3에 업로드하기 위한 예제 환경 설정 및 사용자 지정 교육 스크립트</p> <pre data-bbox="597 982 1026 1871">!pip install stepfunctions import boto3 import stepfunctions import sagemaker import datetime from stepfunctions import steps from stepfunctions.inputs import ExecutionInput from stepfunctions.steps import (Chain) from stepfunctions.workflow import Workflow from sagemaker .processing import</pre>	<p>데이터 사이언티스트</p>

작업	설명	필요한 기술
	<pre> ScriptProcessor, ProcessingInput, ProcessingOutput sagemaker_session = sagemaker.Session() bucket = sagemaker _session.default_b ucket() role = sagemaker .get_execution_role() prefix = 'byoc-tra ining-model' # See prerequisites section to create this role workflow_execution_rol e = f"arn:aws:iam:: {account_id}:role/Ama zonSageMaker-StepF unctionsWorkflowEx ecutionRole" execution_input = ExecutionInput(schema={ "Preproce ssingJobName": str}) input_code = sagemaker _session.upload_data("training.py", bucket=bucket, key_prefix="prepro cessing.py",) </pre> <p>사용자 지정 Amazon ECR 이 미지 및 Python 스크립트를 사</p>	

작업	설명	필요한 기술
	<p>용하는 SageMaker 처리 단계 정의 예시</p> <p>참고: execution_input 파라미터를 사용하여 작업 이름을 지정해야 합니다. 파라미터 값은 작업이 실행될 때마다 고유해야 합니다. 또한 training.py 파일의 코드는 input 파라미터로 ProcessingStep 에 전달되므로 컨테이너 내에 복사됩니다. ProcessingInput 코드의 대상은 container_entrypoint 의 두 번째 인수와 동일합니다.</p> <pre> script_processor = ScriptProcessor(command=['python3'], image_uri=image_uri, role=role, instance_count=1, instance_type='ml.m5.xlarge') processing_step = steps.ProcessingStep("training-step", processor=script_processor, job_name=execution_input["PreprocessingJobName"], </pre>	

작업	설명	필요한 기술
	<pre data-bbox="592 205 1031 1375"> inputs=[Processin gInput(source=in put_code, destinati on="/opt/ml/proces sing/input/code", input_nam e="code",),], outputs=[Processin gOutput(source='/ opt/ml/processing/ model', destinati on="s3://{}/{}".fo rmat(bucket, prefix), output_na me='byoc-example')], container_entrypoi nt=["python3", "/opt/ ml/processing/input/c ode/training.py"],) </pre> <p data-bbox="592 1417 1031 1543">SageMaker 처리 작업을 실행하는 Step Functions 워크플로 예제</p> <p data-bbox="592 1585 1031 1858">참고: 이 예제 워크플로에는 SageMaker 처리 작업 단계만 포함되며 전체 Step Functions 워크플로우는 포함되지 않습니다. 전체 예제 워크플로는 AWS Step Functions 데이터 과학</p>	

작업	설명	필요한 기술
	<p>SDK 설명서의 예제 노트북을 참조하십시오. SageMaker</p> <pre> workflow_graph = Chain([processing_ step]) workflow = Workflow(name="ProcessingWo rkflow", definition=workflo w_graph, role=workflow_exec ution_role) workflow.create() # Execute workflow execution = workflow. execute(inputs={ "Preproce ssingJobName": str(datetime.datet ime.now().strftime ("%Y%m%d%H%M-%SS")), # Each pre processin g job (SageMaker processing job) requires a unique name, }) execution_output = execution.get_outp ut(wait=True) </pre>	

관련 리소스

- [프로세스 데이터](#) (Amazon SageMaker 개발자 가이드)

- [자체 교육 컨테이너 조정](#) (Amazon SageMaker 개발자 가이드)

Amazon의 추론 파이프라인을 사용하여 단일 엔드포인트의 ML 모델에 전처리 로직을 배포하십시오. SageMaker

작성자: Mohan Gowda Purushothama(AWS), Gabriel Rodriguez Garcia(AWS), Mateusz Zaremba(AWS)

환경: 프로덕션

기술: 기계 학습 및 AI, 컨테이너 및 마이크로서비스

AWS 서비스: 아마존 SageMaker, 아마존 ECR

요약

이 패턴은 SageMaker Amazon의 [추론 파이프라인을 사용하여 단일 엔드포인트에 여러 파이프라인 모델 객체를 배포하는 방법을 설명합니다.](#) 파이프라인 모델 객체는 사전 처리, 모델 추론, 사후 처리와 같은 다양한 기계 학습(ML) 워크플로 단계를 나타냅니다. [직렬로 연결된 파이프라인 모델 객체의 배포를 설명하기 위해 이 패턴은 사전 처리 SciKit-learn 컨테이너와 내장된 선형 학습기 알고리즘을 기반으로 하는 회귀 모델을 배포하는 방법을 보여줍니다.](#) SageMaker 배포는 단일 엔드포인트를 기반으로 호스팅됩니다. SageMaker

참고: 이 패턴의 배포는 ml.m4.2xlarge 인스턴스 유형을 사용합니다. 데이터 크기 요구 사항 및 워크플로의 복잡성에 맞는 인스턴스 유형을 사용하는 것이 좋습니다. 자세한 내용은 [Amazon SageMaker 요금을 참조하십시오.](#) 이 패턴은 [Scikit-learn용으로 사전 구축된 Docker 이미지를](#) 사용하지만 자체 Docker 컨테이너를 사용하여 워크플로에 통합할 수 있습니다.

사전 조건 및 제한 사항

사전 조건

- 활성 상태의 AWS 계정.
- [Python 3.9](#)
- [아마존 SageMaker 파이썬 SDK](#) 및 [보토3 라이브러리](#)
- [기본 SageMaker 권한 및 아마존 심플 스토리지 서비스 \(Amazon S3\) 권한을 가진 AWS ID 및 액세스 관리 \(AWS IAM\) 역할](#)

제품 버전

- [아마존 SageMaker 파이썬 SDK 2.49.2](#)

아키텍처

대상 기술 스택

- Amazon Elastic Container Registry (Amazon ECR)
- 아마존 SageMaker
- 아마존 SageMaker 스튜디오
- Amazon Simple Storage Service(S3)
- Amazon용 [실시간 추론](#) 엔드포인트 SageMaker

대상 아키텍처

다음 다이어그램은 Amazon SageMaker 파이프라인 모델 객체 배포를 위한 아키텍처를 보여줍니다.

이 다이어그램은 다음 워크플로를 보여줍니다.

1. SageMaker 노트북은 파이프라인 모델을 배포합니다.
2. S3 버킷은 모델 아티팩트를 저장합니다.
3. Amazon ECR은 S3 버킷에서 소스 컨테이너 이미지를 가져옵니다.

도구

AWS 도구

- [Amazon Elastic Container Registry\(Amazon ECR\)](#)는 안전하고 확장 가능하고 신뢰할 수 있는 관리형 컨테이너 이미지 레지스트리 서비스입니다.
- [SageMakerAmazon](#)은 ML 모델을 구축 및 교육한 다음 프로덕션 준비가 완료된 호스팅 환경에 배포하는 데 도움이 되는 관리형 ML 서비스입니다.
- [Amazon SageMaker Studio](#)는 ML 모델을 구축, 교육, 디버그, 배포 및 모니터링할 수 있는 ML용 웹 기반 통합 개발 환경 (IDE) 입니다.
- [Amazon Simple Storage Service\(S3\)](#)는 원하는 양의 데이터를 저장, 보호 및 검색하는 데 도움이 되는 클라우드 기반 객체 스토리지 서비스입니다.

코드

이 패턴의 코드는 [Scikit-Learn 및 Linear Learner GitHub 리포지토리가 포함된 추론 파이프라인에서](#) 사용할 수 있습니다.

에픽

데이터 세트 준비

작업	설명	필요한 기술
회귀 작업을 위한 데이터 세트를 준비합니다.	<p>Amazon SageMaker 스튜디오에서 노트북을 엽니다.</p> <p>필요한 모든 라이브러리를 가져오고 작업 환경을 초기화하려면 노트북에서 다음 예제 코드를 사용합니다.</p> <pre>import sagemaker from sagemaker import get_execution_role sagemaker_session = sagemaker.Session() # Get a SageMaker- compatible role used by this Notebook Instance. role = get_execu tion_role() # S3 prefix bucket = sagemaker _session.default_b ucket() prefix = "Scikit-L inearLearner-pipel ine-abalone-example"</pre>	데이터 사이언티스트

작업	설명	필요한 기술
	<p>샘플 데이터 세트를 다운로드하려면 노트북에 다음 코드를 추가합니다.</p> <pre data-bbox="597 380 1029 659">! mkdir abalone_data ! aws s3 cp s3://sagemaker-sample-files/datasets/tabular/uci_abalone/abalone.csv ./abalone_data</pre> <p>참고: 이 패턴의 예제에서는 UCI Machine Learning Repository의 Abalone 데이터 세트를 사용합니다.</p>	
<p>데이터 세트를 S3 버킷에 업로드합니다.</p>	<p>이전에 데이터 세트를 준비한 노트북에 다음 코드를 추가하여 샘플 데이터를 S3 버킷에 업로드합니다.</p> <pre data-bbox="597 1136 1029 1688">WORK_DIRECTORY = "abalone_data" train_input = sagemaker_session.upload_data(path="{}/{}".format(WORK_DIRECTORY, "abalone.csv"), bucket=bucket, key_prefix="{}/{}" .format(prefix, "train"),)</pre>	<p>데이터 사이언티스트</p>

SKLearn을 사용하여 데이터 프리프로세서 생성

작업	설명	필요한 기술
<p>preprocessor.py 스크립트를 준비합니다.</p>	<ol style="list-style-type: none"> 1. GitHub sklearn_abalone_feature_extractor.py 리포지토리의 Python 파일에서 전처리 로직을 복사한 다음 이라는 sklearn_abalone_feature_extractor.py 별도의 Python 파일에 코드를 붙여 넣습니다. 사용자 지정 데이터 세트와 사용자 지정 워크플로에 맞게 코드를 수정할 수 있습니다. 2. 프로젝트의 루트 디렉터리, 즉 SageMaker 노트북을 실행하는 동일한 위치에 sklearn_abalone_feature_extractor.py 파일을 저장합니다. 	<p>데이터 사이언티스트</p>
<p>SKLearn 프리프로세서 객체를 생성합니다.</p>	<p>최종 추론 파이프라인에 통합할 수 있는 SKLearn 프리프로세서 객체 (SKLearn Estimator라고 함) 를 만들려면 노트북에서 다음 코드를 실행하십시오.</p> <p>SageMaker</p> <pre> from sagemaker.sklearn.estimator import SKLearn FRAMEWORK_VERSION = "0.23-1" script_path = "sklearn_abalone_feature_extractor.py" </pre>	<p>데이터 사이언티스트</p>

작업	설명	필요한 기술
	<pre>sklearn_preprocessor = SKLearn(entry_point=script _path, role=role, framework_version= FRAMEWORK_VERSION, instance_type="ml. c4.xlarge", sagemaker_session= sagemaker_session,) sklearn_preproc essor.fit({"train": train_input})</pre>	

작업	설명	필요한 기술
<p>프리프로세서의 추론을 테스트합니다.</p>	<p>프리프로세서가 올바르게 정의되었는지 확인하려면 노트북에 다음 코드를 입력하여 일괄 변환 작업을 시작하십시오.</p> <p>SageMaker</p> <pre data-bbox="594 489 1029 1719"> # Define a SKLearn Transformer from the trained SKLearn Estimator transformer = sklearn_p reprocessor.transf ormer(instance_count=1, instance_type="ml. m5.xlarge", assemble_ with="Line", accept="t ext/csv") # Preprocess training input transformer.transform (train_input, content_type="text/ csv") print("Waiting for transform job: " + transformer.latest _transform_job.job _name) transformer.wait() preprocessed_train = transformer.output _path </pre>	

기계 학습 모델 생성

작업	설명	필요한 기술
<p>모델 객체를 생성합니다.</p>	<p>선형 학습기 알고리즘을 기반으로 모델 객체를 생성하려면 노트북에 다음 코드를 입력하십시오. SageMaker</p> <pre data-bbox="591 548 1024 1873"> import boto3 from sagemaker .image_uris import retrieve ll_image = retrieve("linear-learner", boto3.Session().re gion_name) s3_ll_output_key _prefix = "ll_train ing_output" s3_ll_output_location = "s3://{}/{}/{}/{" .format(bucket, prefix, s3_ll_output_key_p refix, "ll_model") ll_estimator = sagemaker.estimato r.Estimator(ll_image, role, instance_count=1, instance_type="ml. m4.2xlarge", volume_size=20, max_run=3600, input_mode="File", output_path=s3_ll_ output_location, </pre>	<p>데이터 사이언티스트</p>

작업	설명	필요한 기술
	<pre> sagemaker_session= sagemaker_session,) ll_estimator.s et_hyperparameters (feature_dim=10, predictor_type="re gressor", mini_batch_size=32) ll_train_data = sagemaker.inputs.TrainingInput(preprocessed_train , distribution="FullyReplicated", content_type="text/csv", s3_data_type="S3Prefix",) data_channels = {"train": ll_train_data} ll_estimator.fit(inputs=data_channels, logs=True) </pre> <p>위의 코드는 모델의 퍼블릭 Amazon ECR 레지스트리에서 관련 Amazon ECR 도커 이미지를 검색하고 예측기 객체를 생성한 다음 해당 객체를 사용하여 회귀 모델을 학습시킵니다.</p>	

최종 파이프라인 배포

작업	설명	필요한 기술
파이프라인 모델을 배포합니다.	<p>파이프라인 모델 객체 (즉, 전처리기 객체) 를 생성하고 객체를 배포하려면 노트북에 다음 코드를 입력합니다.</p> <p>SageMaker</p> <pre data-bbox="594 596 1029 1879"> from sagemaker.model import Model from sagemaker .pipeline import PipelineModel import boto3 from time import gmtime, strftime timestamp_prefix = strftime("%Y-%m-%d- %H-%M-%S", gmtime()) scikit_learn_inf erencee_model = sklearn_preprocess or.create_model() linear_learner_model = ll_estimator.creat e_model() model_name = "inferenc e-pipeline-" + timestamp_prefix endpoint_name = "inference-pipeline- ep-" + timestamp_prefix sm_model = PipelineM odel(name=model_name, role=role, models= [scikit_learn_infe </pre>	데이터 사이언티스트

작업	설명	필요한 기술
	<pre> rencee_model, linear_learner_model]) sm_model.deploy(initial_instance_count =1, instance_type="ml. c4.xlarge", endpoint_ name=endpoint_name) </pre> <p>참고: 모델 객체에 사용되는 인스턴스 유형을 필요에 맞게 조정할 수 있습니다.</p>	
<p>추론을 테스트합니다.</p>	<p>엔드포인트가 제대로 작동하는지 확인하려면 노트북에서 다음 샘플 추론 코드를 실행하세요. SageMaker</p> <pre> from sagemaker.predictor import Predictor from sagemaker.serializers import CSVSerializer payload = "M, 0.44, 0.365, 0.125, 0.516, 0.2155, 0.114, 0.155" actual_rings = 10 predictor = Predictor(endpoint_name=endp oint_name, sagemaker _session=sagemaker _session, serialize r=CSVSerializer()) print(predictor .predict(payload)) </pre>	<p>데이터 사이언티스트</p>

관련 리소스

- [Amazon SageMaker 추론 파이프라인과 Scikit-Learn \(AWS Machine Learning 블로그\) 을 사용하여 예측하기 전에 입력 데이터를 사전 처리합니다.](#)
- [Amazon을 통한 엔드투엔드 기계 학습 SageMaker \(GitHub\)](#)

RAG 및 프롬프트를 사용하여 고급 제너레이티브 AI 채팅 기반 어시스턴트 개발 ReAct

작성자: 프라빈 쿠마르 제야라잔 (AWS), 준동 차오 (AWS), 카라 양 (AWS), 키오와 잭슨 (AWS), 노아 해밀턴 (AWS), 슈아이 카오 (AWS)

코드 리포지토리: [genai-bed-rock-chatbot](#)

환경: PoC 또는 파일럿

기술: 기계 학습 및 AI, 데이터베이스 DevOps, 서버리스

AWS 서비스: 아마존 베드락, 아마존 ECS, 아마존 켄드라, AWS Lambda

요약

일반적인 기업은 데이터의 70% 가 사일로 시스템에 갇혀 있습니다. 생성형 AI 기반 채팅 기반 어시스턴트를 사용하면 자연어 상호 작용을 통해 이러한 데이터 사일로 간의 인사이트와 관계를 파악할 수 있습니다. 제너레이티브 AI를 최대한 활용하려면 출력이 신뢰할 수 있고 정확하며 사용 가능한 기업 데이터를 포함해야 합니다. 성공적인 채팅 기반 어시스턴트는 다음 사항에 따라 달라집니다.

- 제너레이티브 AI 모델 (예: 엔트로픽 클로드 2)
- 데이터 소스 벡터화
- 모델 실행을 위한 고급 추론 기법 (예: [ReAct 프레임워크](#))

이 패턴은 Amazon Simple Storage Service (Amazon S3) 버킷, AWS Glue 및 아마존 관계형 데이터베이스 서비스 (Amazon RDS) 와 같은 데이터 소스에서 데이터를 검색하는 접근 방식을 제공합니다. RAG ([검색 증강 세대](#)) 를 메서드와 함께 사용하면 해당 데이터에서 가치를 얻을 수 있습니다. chain-of-thought 결과는 회사에 저장된 데이터 전체를 활용하는 복잡한 채팅 기반 비서 대화를 지원합니다.

이 패턴은 Amazon SageMaker 설명서와 요금 데이터 표를 예로 들어 생성적 AI 채팅 기반 어시스턴트의 기능을 살펴봅니다. 요금 및 SageMaker 서비스 기능에 대한 질문에 답하여 고객이 서비스를 평가하는 데 도움이 되는 채팅 기반 어시스턴트를 구축해 보겠습니다. 이 솔루션은 Streamlit 라이브러리를 사용하여 프론트엔드 애플리케이션을 구축하고 대규모 언어 모델 (LLM) 을 기반으로 하는 애플리케이션 백엔드를 개발하기 위한 LangChain 프레임워크를 사용합니다.

채팅 기반 어시스턴트에 대한 문의는 가능한 세 가지 워크플로 중 하나로 라우팅하기 위한 초기 의도 분류를 통해 충족됩니다. 가장 정교한 워크플로는 일반적인 자문 지침과 복잡한 가격 분석을 결합합니다. 패턴을 기업, 기업 및 산업 사용 사례에 맞게 조정할 수 있습니다.

사전 조건 및 제한 사항

사전 조건

- [AWS 명령줄 인터페이스 \(AWS CLI\) 설치](#) 및 구성
- [AWS 클라우드 개발 키트 \(AWS CDK\) 툴킷 2.114.1 이상 설치](#) 및 구성
- Python과 AWS CDK에 대한 기본 지식
- [Git](#) 설치
- [도커 설치](#)
- [Python 3.11 이상](#) 설치 및 구성 (자세한 내용은 [도구](#) 섹션 참조)
- [AWS CDK를 사용하여 부트스트랩한 활성 AWS 계정](#)
- Amazon Bedrock 서비스에서 Amazon Titan 및 Anthropic Claude [모델 액세스가 가능해졌습니다](#).
- [AWS 보안 인증 정보](#)(터미널 환경에 올바르게 구성된 AWS_ACCESS_KEY_ID 포함)

제한 사항

- LangChain 스트리밍을 위한 모든 LLM을 지원하지는 않습니다. 엔트로픽 클로드 모델은 지원되지만 AI21 Labs의 모델은 지원되지 않습니다.
- 이 솔루션은 단일 AWS 계정에 배포됩니다.
- 이 솔루션은 Amazon Bedrock과 Amazon Kendra를 사용할 수 있는 AWS 지역에만 배포할 수 있습니다. 가용성에 대한 자세한 내용은 [Amazon Bedrock 및 Amazon Kendra](#) 설명서를 참조하십시오.

제품 버전

- 파이썬 버전 3.11 이상
- 스트림릿 버전 1.30.0 이상
- 스트림릿 채팅 버전 0.1.1 이상
- LangChain 버전 0.1.12 이상
- AWS CDK 버전 2.132.1 이상

아키텍처

대상 기술 스택

- Amazon Athena
- Amazon Bedrock
- Amazon Elastic Container Service(Amazon ECS)
- Glue
- Lambda
- Amazon S3
- Amazon Kendra
- Elastic Load Balancing

대상 아키텍처

AWS CDK 코드는 AWS 계정에서 채팅 기반 어시스턴트 애플리케이션을 설정하는 데 필요한 모든 리소스를 배포합니다. 다음 다이어그램에 표시된 채팅 기반 어시스턴트 애플리케이션은 사용자의 SageMaker 관련 쿼리에 응답하도록 설계되었습니다. 사용자는 Application Load Balancer를 통해 Streamlit 애플리케이션을 호스팅하는 Amazon ECS 클러스터가 포함된 VPC에 연결합니다. 오케스트레이션 Lambda 함수는 애플리케이션에 연결됩니다. S3 버킷 데이터 소스는 Amazon Kendra 및 AWS Glue를 통해 Lambda 함수에 데이터를 제공합니다. Lambda 함수는 Amazon Bedrock에 연결하여 채팅 기반 어시스턴트 사용자의 쿼리 (질문) 에 응답합니다.

1. 오케스트레이션 Lambda 함수는 아마존 베드락 모델 (클로드 2) 에 LLM 프롬프트 요청을 보냅니다.
2. Amazon Bedrock은 LLM 응답을 오케스트레이션 Lambda 함수로 다시 보냅니다.

오케스트레이션 Lambda 함수 내의 로직 흐름

사용자가 Streamlit 애플리케이션을 통해 질문을 하면 오케스트레이션 Lambda 함수가 직접 호출됩니다. 다음 다이어그램은 Lambda 함수가 호출될 때의 로직 흐름을 보여줍니다.

- 1단계 — 입력 query (질문) 은 다음 세 가지 인텐트 중 하나로 분류됩니다.
 - 일반 SageMaker 지침 질문

- 일반 SageMaker 가격 책정 (교육/추론) 질문
- 가격 책정과 관련된 복잡한 질문 SageMaker
- 2단계 — 입력으로 다음 세 가지 서비스 중 하나가 query 시작됩니다.
 - RAG Retrieval service는 [Amazon Kendra](#) 벡터 데이터베이스에서 관련 컨텍스트를 검색하고 [Amazon Bedrock](#)을 통해 LLM을 호출하여 검색된 컨텍스트를 응답으로 요약합니다.
 - Database Query service는 LLM, 데이터베이스 메타데이터 및 관련 테이블의 샘플 행을 사용하여 입력을 SQL 쿼리로 변환합니다. query 데이터베이스 쿼리 서비스는 [Amazon Athena](#)를 통해 SageMaker 요금 데이터베이스에 대해 SQL 쿼리를 실행하고 쿼리 결과를 응답으로 요약합니다.
 - In-context ReACT Agent service는 입력을 여러 단계로 query 분류한 다음 응답을 제공합니다. 에이전트는 추론 과정에서 관련 정보를 검색하기 위한 Database Query service 도구로 RAG Retrieval service 및 를 사용합니다. 추론 및 조치 프로세스가 완료되면 에이전트는 응답으로 최종 답변을 생성합니다.
- 3단계 — 오케스트레이션 Lambda 함수의 응답이 Streamlit 애플리케이션에 출력으로 전송됩니다.

도구

서비스

- [Amazon Athena](#)는 표준 SQL을 사용하여 Amazon Simple Storage Service(S3)에 있는 데이터를 직접 간편하게 분석할 수 있는 대화형 쿼리 서비스입니다.
- [Amazon Bedrock](#)은 선도적인 AI 스타트업과 Amazon의 고성능 기초 모델 (FM) 을 통합 API를 통해 사용할 수 있게 해주는 완전관리형 서비스입니다.
- [AWS Cloud Development Kit\(AWS CDK\)](#)는 AWS 클라우드 인프라를 코드로 정의하고 프로비저닝하는 데 도움이 되는 소프트웨어 개발 프레임워크입니다.
- [AWS Command Line Interface \(AWS CLI\)](#)는 명령줄 셸에서 명령을 사용하여 AWS 서비스와 상호 작용할 수 있는 오픈 소스 도구입니다.
- [Amazon Elastic Container Service\(Amazon ECS\)](#)는 클러스터에서 컨테이너를 실행, 중지 및 관리하는 데 도움이 되는 빠르고 확장 가능한 컨테이너 관리 서비스입니다.
- [AWS Glue](#)는 완전 관리형 추출, 전환, 적재(ETL) 서비스입니다. 이를 통해 데이터 스토어와 데이터 스트림 간에 데이터를 안정적으로 분류, 정리, 보강하고 이동할 수 있습니다. 이 패턴은 AWS Glue 크롤러와 AWS Glue 데이터 카탈로그 테이블을 사용합니다.
- [Amazon Kendra](#)는 자연어 처리 및 고급 기계 학습 알고리즘을 사용하여 데이터에서 검색 질문에 대한 구체적인 답변을 반환하는 지능형 검색 서비스입니다.

- [AWS Lambda](#)는 서버를 프로비저닝하거나 관리할 필요 없이 코드를 실행하는 데 도움이 되는 컴퓨팅 서비스입니다. 필요할 때만 코드를 실행하며 자동으로 확장이 가능하므로 사용한 컴퓨팅 시간만큼만 비용을 지불합니다.
- [Amazon Simple Storage Service\(Amazon S3\)](#)는 원하는 양의 데이터를 저장, 보호 및 검색하는 데 도움이 되는 클라우드 기반 객체 스토리지 서비스입니다.
- [Elastic Load Balancing\(ELB\)](#)은 들어오는 애플리케이션 또는 네트워크 트래픽을 여러 대상에 분산합니다. 예를 들어 하나 이상의 가용 영역에 있는 Amazon Elastic Compute Cloud(Amazon EC2) 인스턴스, 컨테이너, IP 주소 전반에 걸쳐 트래픽을 분산할 수 있습니다.

코드 리포지토리

이 패턴의 코드는 리포지토리에서 사용할 수 있습니다. GitHub [genai-bedrock-chatbot](#)

코드 리포지토리에는 다음 파일과 폴더가 포함되어 있습니다.

- assets폴더 - 정적 자산, 아키텍처 다이어그램, 공개 데이터세트
- code/lambda-container폴더 — Lambda 함수에서 실행되는 Python 코드
- code/streamlit-app폴더 — Amazon ECS에서 컨테이너 이미지로 실행되는 Python 코드
- tests폴더 — AWS CDK 구문을 단위 테스트하기 위해 실행되는 Python 파일
- code/code_stack.py— AWS CDK는 AWS 리소스를 생성하는 데 사용되는 Python 파일을 구성합니다.
- app.py— 대상 AWS 계정에 AWS 리소스를 배포하는 데 사용되는 AWS CDK 스택 Python 파일
- requirements.txt— AWS CDK에 설치해야 하는 모든 Python 종속성 목록
- requirements-dev.txt— 유닛 테스트 스위트를 실행하기 위해 AWS CDK에 설치해야 하는 모든 Python 종속성 목록
- cdk.json - 리소스를 스핀업하는 데 필요한 값을 제공하는 입력 파일

참고: AWS CDK 코드는 솔루션 배포를 위해 [L3 \(계층 3\) 구조와 AWS에서 관리하는 AWS ID 및 액세스 관리 \(IAM\) 정책을](#) 사용합니다.

모범 사례

- 여기에 제공된 코드 예제는 proof-of-concept (PoC) 또는 파일럿 데모용으로만 제공됩니다. 코드를 프로덕션으로 가져오려면 다음 모범 사례를 사용하십시오.

- [Amazon S3 액세스 로깅이 활성화되었습니다.](#)
- [VPC 흐름 로그가 활성화되었습니다.](#)
- [Amazon Kendra 엔터프라이즈 에디션 인덱스가 활성화되었습니다.](#)
- Lambda 함수에 대한 모니터링 및 알림을 설정합니다. 자세한 내용은 [Lambda 함수 모니터링 및 문제 해결](#)을 참조하십시오. Lambda 함수를 사용할 때의 일반적인 모범 사례는 [AWS 설명서](#)를 참조하십시오.

에픽

로컬 머신에서 AWS 보안 인증 설정

작업	설명	필요한 기술
스택이 배포될 계정 및 AWS 지역의 변수를 내보냅니다.	환경 변수를 사용하여 AWS CDK용 AWS 보안 인증을 제공하려면 다음 명령을 실행합니다. <pre>export CDK_DEFAULT_ACCOUNT=<12 Digit AWS Account Number> export CDK_DEFAULT_REGION=<region></pre>	DevOps 엔지니어, AWS DevOps
AWS CLI 프로필을 설치합니다.	계정에 대한 AWS CLI 프로필을 설정하려면 AWS 설명서 의 지침을 따르십시오.	DevOps 엔지니어, AWS DevOps

환경을 설정합니다.

작업	설명	필요한 기술
로컬 머신에서 저장소를 복제합니다.	리포지토리를 복제하려면 터미널에서 다음 명령을 실행합니다.	DevOps 엔지니어, AWS DevOps

작업	설명	필요한 기술
	<pre>git clone https://github.com/aws-labs/genai-bedrock-chat-bot.git</pre>	
<p>Python 가상 환경을 설정하고 필요한 종속성을 설치합니다.</p>	<p>다음 명령을 실행하여 Python 가상 환경을 설정합니다.</p> <pre>cd genai-bedrock-chat-bot python3 -m venv .venv source .venv/bin/activate</pre> <p>필요한 종속성을 설정하려면 다음 명령을 실행합니다.</p> <pre>pip3 install -r requirements.txt</pre>	<p>DevOps 엔지니어, AWS DevOps</p>
<p>AWS CDK 환경을 설정하고 AWS CDK 코드를 합성합니다.</p>	<ol style="list-style-type: none"> 1. AWS 계정에서 AWS CDK 환경을 설정하려면 다음 명령을 실행합니다. <pre>cdk bootstrap aws://ACCOUNT-NUMBER/REGION</pre> 2. 코드를 AWS CloudFormation stack 구성으로 변환하려면 명령을 실행합니다 <code>cdk synth</code>. 	<p>DevOps 엔지니어, AWS DevOps</p>

채팅 기반 어시스턴트 애플리케이션 구성 및 배포

작업	설명	필요한 기술
Claude 모델 액세스를 제공합니다.	AWS 계정에 대한 Anthropic Claude 모델 액세스를 활성화하려면 Amazon Bedrock 설명서의 지침을 따르십시오.	AWS DevOps
계정에 리소스를 배포하십시오.	<p>AWS CDK를 사용하여 AWS 계정에 리소스를 배포하려면 다음과 같이 하십시오.</p> <ol style="list-style-type: none"> 복제된 리포지토리의 루트에 있는 <code>cdk.json</code> 파일에 파라미터를 입력합니다. logging 예제 값은 INFO, DEBUGWARN, 및 ERROR입니다. 이러한 값은 Lambda 함수 및 Streamlit 애플리케이션에 대한 로그 수준 메시지를 정의합니다. 복제된 리포지토리의 루트에 있는 <code>app.py</code> 파일에는 배포에 사용된 AWS CloudFormation 스택 이름이 들어 있습니다. 기본 스택 이름은 <code>chatbot-stack</code>입니다. 리소스를 배포하려면 <code>cdk deploy</code> 명령을 실행합니다. <p>이 <code>cdk deploy</code> 명령은 L3 구조를 사용하여 문서 및 CSV 데이터세트 파일을 S3</p>	AWS DevOps, DevOps 엔지니어

작업	설명	필요한 기술
	<p>버킷으로 복사하기 위한 여러 Lambda 함수를 생성합니다.</p> <p>4. 명령이 완료되면 AWS Management 콘솔에 로그인하고 콘솔을 열고 스택이 CloudFormation 성공적으로 배포되었는지 검토하십시오.</p> <p>배포에 성공하면 CloudFormation 출력 섹션에 제공된 URL을 사용하여 채팅 기반 어시스턴트 애플리케이션에 액세스할 수 있습니다.</p>	

작업	설명	필요한 기술
<p>AWS Glue 크롤러를 실행하며 데이터 카탈로그 테이블을 생성합니다.</p>	<p>AWS Glue 크롤러는 데이터 스키마를 동적으로 유지하는 데 사용됩니다. 솔루션은 필요에 따라 크롤러를 실행하여 AWS Glue Data Catalog 테이블에 파티션을 생성하고 업데이트합니다. CSV 데이터세트 파일을 S3 버킷으로 복사한 후 AWS Glue 크롤러를 실행하고 테스트용 데이터 카탈로그 테이블 스키마를 생성합니다.</p> <ol style="list-style-type: none"> 1. AWS Glue 콘솔로 이동합니다. 2. 탐색 창의 데이터 카탈로그에서 크롤러를 선택합니다. 3. 접미사가 있는 크롤러를 선택합니다. sagemaker-pricing-crawler 4. 크롤러를 실행합니다. 5. 크롤러가 성공적으로 실행되면 AWS Glue 데이터 카탈로그 테이블을 생성합니다. <p>참고: AWS CDK 코드는 AWS Glue 크롤러가 온디맨드로 실행되도록 구성하지만 주기적으로 실행되도록 예약할 수도 있습니다.</p>	<p>DevOps 엔지니어, AWS DevOps</p>

작업	설명	필요한 기술
문서 인덱싱을 시작합니다.	<p>파일을 S3 버킷으로 복사한 후 Amazon Kendra를 사용하여 파일을 크롤링하고 인덱싱합니다.</p> <ol style="list-style-type: none"> 1. Amazon Kendra 콘솔로 이동합니다. 2. 접미사가 있는 색인을 선택합니다. chatbot-index 3. 탐색 창에서 데이터 소스를 선택하고 접미사가 있는 데이터 소스 커넥터를 선택합니다. chatbot-index 4. [Sync Now] 를 선택하여 색인 프로세스를 시작합니다. <p>참고: AWS CDK 코드는 Amazon Kendra 인덱스 동기화가 온디맨드 실행되도록 구성하지만, 스케줄 파라미터를 사용하여 주기적으로 실행할 수도 있습니다.</p>	AWS DevOps, DevOps 엔지니어

솔루션의 모든 AWS 리소스를 정리하십시오.

작업	설명	필요한 기술
AWS 리소스를 제거합니다.	<p>솔루션을 테스트한 후 리소스를 정리하십시오.</p> <ol style="list-style-type: none"> 1. 솔루션에서 배포한 AWS 리소스를 제거하려면 cdk destroy 명령을 실행합니다. 	DevOps 엔지니어, AWS DevOps

작업	설명	필요한 기술
	<p>2. 두 S3 버킷에서 모든 객체를 삭제한 다음 버킷을 제거합니다.</p> <p>자세한 내용은 버킷 삭제를 참조하십시오.</p>	

문제 해결

문제	Solution
AWS CDK가 오류를 반환합니다.	AWS CDK 문제에 대한 도움이 필요하다면 일반적인 AWS CDK 문제 해결 을 참조하십시오.

관련 리소스

- [아마존 베드락](#):
 - [모델 액세스](#)
 - [기초 모델의 추론 파라미터](#)
- [Python을 사용하여 Lambda 함수 빌드](#)
- [AWS CDK로 시작하세요](#)
- [Python에서 AWS CDK로 작업하기](#)
- [AWS 기반 제너레이티브 AI 애플리케이션 빌더](#)
- [LangChain 설명서](#)
- [간소화된 설명서](#)

추가 정보

AWS CDK 명령

AWS CDK로 작업할 때는 다음과 같은 유용한 명령을 유념하십시오.

- [앱의 모든 스택 나열하기](#)

```
cdk ls
```

- 합성된 AWS 템플릿을 내보냅니다. CloudFormation

```
cdk synth
```

- 스택을 기본 AWS 계정 및 리전에 배포하기

```
cdk deploy
```

- 배포된 스택을 현재 상태와 비교하기

```
cdk diff
```

- AWS CDK 설명서 열기

```
cdk docs
```

- CloudFormation 스택을 삭제하고 AWS에 배포된 리소스를 제거합니다.

```
cdk destroy
```

Amazon Bedrock 에이전트 및 지식 베이스를 사용하여 완전 자동화된 채팅 기반 도우미를 개발하십시오.

작성: 준둥 차오 (AWS), 카라 양 (AWS), 키오와 잭슨 (AWS), 노아 해밀턴 (AWS), 프라빈 쿠마르 제아라잔 (AWS), 슈아이 카오 (AWS)

코드 리포지토리: [genai-bedrock-agent-chatbot](#)

환경: PoC 또는 파일럿

기술: 기계 학습 및 AI, 서버리스

AWS 서비스: 아마존 베드락, AWS CDK, AWS Lambda

요약

많은 조직이 다양한 데이터 소스를 조정하여 포괄적인 답변을 제공할 수 있는 채팅 기반 어시스턴트를 만드는 데 어려움을 겪고 있습니다. 이 패턴은 간단한 배포를 통해 문서와 데이터베이스의 쿼리에 응답할 수 있는 채팅 기반 도우미를 개발하는 솔루션을 제공합니다.

[Amazon Bedrock](#)을 시작으로 이 완전 관리형 제너레이티브 인공 지능 (AI) 서비스는 다양한 고급 기반 모델 (FM) 을 제공합니다. 이를 통해 개인 정보 보호 및 보안에 중점을 둔 제너레이티브 AI 애플리케이션을 효율적으로 만들 수 있습니다. 문서 검색의 관점에서 볼 때 [검색 증강 세대 \(RAG\)](#) 는 매우 중요한 기능입니다. [지식 베이스를 사용하여 외부](#) 소스의 상황에 맞는 관련 정보로 FM 프롬프트를 보강합니다. [Amazon OpenSearch 서버리스](#) 인덱스는 Amazon Bedrock의 지식 베이스를 뒷받침하는 벡터 데이터베이스 역할을 합니다. 이러한 통합은 부정확성을 최소화하고 응답이 사실에 입각한 문서에 반영되도록 세심하고 신속한 엔지니어링을 통해 향상되었습니다. 데이터베이스 쿼리의 경우 Amazon Bedrock의 FM은 텍스트 쿼리를 특정 파라미터를 통합하여 구조화된 SQL 쿼리로 변환합니다. 이를 통해 [AWS Glue](#) 데이터베이스에서 관리되는 데이터베이스에서 데이터를 정확하게 검색할 수 있습니다. [Amazon Athena](#)는 이러한 쿼리에 사용됩니다.

더 복잡한 쿼리를 처리하려면 문서와 데이터베이스에서 모두 가져온 정보가 있어야 종합적인 답변을 얻을 수 있습니다. [Amazon Bedrock용 에이전트는](#) 복잡한 작업을 이해하고 오케스트레이션을 위해 이를 간단한 작업으로 분류할 수 있는 자율 에이전트를 구축하는 데 도움이 되는 생성형 AI 기능입니다. Amazon Bedrock 자율 에이전트가 지원하는 간소화된 작업에서 얻은 통찰력을 조합하면 정보 합성이 향상되어 더 철저하고 철저한 해답을 얻을 수 있습니다. 이 패턴은 Amazon Bedrock과 자동화된 솔루션 내에서 관련 제너레이티브 AI 서비스 및 기능을 사용하여 채팅 기반 어시스턴트를 구축하는 방법을 보여줍니다.

사전 조건 및 제한 사항

사전 조건

- 활성 상태의 AWS 계정
- [Docker, 설치](#)
- 또는 us-east-1 AWS 지역에 [설치](#) 및 [부트스트랩되는](#) AWS 클라우드 개발 키트 (AWS CDK) us-west-2
- [AWS CDK 툴킷 버전 2.114.1 이상, 설치됨](#)
- AWS Command Line Interface(AWS CLI), [설치](#) 및 [구성됨](#)
- [Python 버전 3.11 이상, 설치됨](#)
- Amazon Bedrock에서 클로드 2, 클로드 2.1, 클로드 인스턴트, 타이탄 임베딩 G1 — 텍스트에 [액세스할](#) 수 있습니다.

제한 사항

- 이 솔루션은 단일 AWS 계정에 배포됩니다.
- 이 솔루션은 Amazon Bedrock과 Amazon OpenSearch 서버리스가 지원되는 AWS 지역에만 배포할 수 있습니다. [자세한 내용은 Amazon Bedrock 및 Amazon 서버리스 설명서를 참조하십시오. OpenSearch](#)

제품 버전

- 라마-인덱스 버전 0.10.6 이상
- 알케미 버전 2.0.23 이상
- 오픈서치-PY 버전 2.4.2 이상
- 요청_AWS4Auth 버전 1.2.3 이상
- 파이썬용 AWS SDK (Boto3) 버전 1.34.57 이상

아키텍처

대상 기술 스택

[AWS Cloud Development Kit \(AWS CDK\)](#) 는 코드로 클라우드 인프라를 정의하고 AWS를 통해 프로비저닝하기 위한 오픈 소스 소프트웨어 개발 프레임워크입니다. CloudFormation 이 패턴에 사용된 AWS CDK 스택은 다음 AWS 리소스를 배포합니다.

- AWS Key Management Service (AWS KMS)
- Amazon Simple Storage Service(S3)
- AWS Glue 데이터베이스 구성 요소용 AWS Glue 데이터 카탈로그
- AWS Lambda
- AWS Identity and Access Management(IAM)
- 아마존 OpenSearch 서버리스
- Amazon Elastic Container Registry(Amazon ECR)
- Amazon Elastic Container Service(Amazon ECS)
- AWS Fargate
- Amazon Virtual Private Cloud(VPC)
- [Application Load Balancer](#)

대상 아키텍처

다이어그램은 여러 AWS 서비스를 사용하는 단일 AWS 지역 내에서 포괄적인 AWS 클라우드 네이티브 설정을 보여줍니다. 채팅 기반 어시스턴트의 기본 인터페이스는 Amazon ECS 클러스터에서 호스팅되는 [Streamlit](#) 애플리케이션입니다. [Application Load Balancer](#)는 접근성을 관리합니다. 이 인터페이스를 통해 이루어진 쿼리는 Invocation Lambda 함수를 활성화하고, 이 함수는 Amazon Bedrock용 에이전트와 인터페이스합니다. 이 에이전트는 Amazon Bedrock의 지식 베이스를 참조하거나 Lambda 함수를 호출하여 사용자 문의에 응답합니다. Agent executor 이 함수는 사전 정의된 API 스키마에 따라 에이전트와 관련된 일련의 작업을 트리거합니다. Amazon Bedrock의 지식 기반은 OpenSearch 서버리스 인덱스를 벡터 데이터베이스 기반으로 사용합니다. 또한 이 Agent executor 함수는 Amazon Athena를 통해 AWS Glue 데이터베이스에 대해 실행되는 SQL 쿼리를 생성합니다.

도구

서비스

- [Amazon Athena](#)는 표준 SQL을 사용하여 Amazon Simple Storage Service(S3)에 있는 데이터를 직접 간편하게 분석할 수 있는 대화형 쿼리 서비스입니다.

- [Amazon Bedrock](#)은 선도적인 AI 스타트업과 Amazon의 고성능 기초 모델 (FM) 을 통합 API를 통해 사용할 수 있게 해주는 완전관리형 서비스입니다.
- [AWS Cloud Development Kit\(AWS CDK\)](#)는 AWS 클라우드 인프라를 코드로 정의하고 프로비저닝하는 데 도움이 되는 소프트웨어 개발 프레임워크입니다.
- [AWS 명령줄 인터페이스 \(AWS CLI\)](#) 는 명령줄 셸의 명령을 통해 AWS 서비스와 상호 작용하는 데 도움이 되는 오픈 소스 도구입니다.
- [Amazon Elastic Container Service\(Amazon ECS\)](#)는 클러스터에서 컨테이너를 실행, 중지 및 관리하는 데 도움이 되는 빠르고 확장 가능한 컨테이너 관리 서비스입니다.
- [Elastic Load Balancing\(ELB\)](#)은 들어오는 애플리케이션 또는 네트워크 트래픽을 여러 대상에 분산합니다. 예를 들어 하나 이상의 가용 영역에 있는 Amazon Elastic Compute Cloud(Amazon EC2) 인스턴스, 컨테이너, IP 주소 전반에 걸쳐 트래픽을 분산할 수 있습니다.
- [AWS Glue](#)는 완전 관리형 추출, 전환, 적재(ETL) 서비스입니다. 이를 통해 데이터 스토어와 데이터 스트림 간에 데이터를 안정적으로 분류, 정리, 보강하고 이동할 수 있습니다. 이 패턴은 AWS Glue 크롤러와 AWS Glue 데이터 카탈로그 테이블을 사용합니다.
- [AWS Lambda](#)는 서버를 프로비저닝하거나 관리할 필요 없이 코드를 실행할 수 있도록 도와주는 컴퓨팅 서비스입니다. 필요할 때만 코드를 실행하며 자동으로 확장이 가능하므로 사용한 컴퓨팅 시간만큼만 비용을 지불합니다.
- [아마존 OpenSearch 서버리스](#)는 아마존 서비스를 위한 온디맨드 서버리스 구성입니다. OpenSearch 이 패턴에서 OpenSearch 서버리스 인덱스는 Amazon Bedrock의 지식 베이스를 위한 벡터 데이터베이스 역할을 합니다.
- [Amazon Simple Storage Service\(S3\)](#)는 원하는 양의 데이터를 저장, 보호 및 검색하는 데 도움이 되는 클라우드 기반 객체 스토리지 서비스입니다.

기타 도구

- [Streamlit](#)은 데이터 애플리케이션을 만들기 위한 오픈 소스 Python 프레임워크입니다.

코드 리포지토리

이 패턴의 코드는 GitHub [genai-bedrock-agent-chatbot](#) 리포지토리에서 사용할 수 있습니다. 코드 리포지토리에는 다음 파일과 폴더가 포함되어 있습니다.

- `assets` 폴더 - 정적 자산 (예: 아키텍처 다이어그램, 공개 데이터셋).
- `code/lambda/action-lambda` 폴더 — Amazon Bedrock 에이전트의 작업 역할을 하는 Lambda 함수의 Python 코드입니다.

- `code/lambda/create-index-lambda` 폴더 — 서버리스 인덱스를 생성하는 OpenSearch Lambda 함수의 Python 코드입니다.
- `code/lambda/invoke-lambda` 폴더 — Amazon Bedrock 에이전트를 호출하는 Lambda 함수의 Python 코드로, Streamlit 애플리케이션에서 직접 호출됩니다.
- `code/lambda/update-lambda` 폴더 — AWS CDK를 통해 AWS 리소스를 배포한 후 리소스를 업데이트하거나 삭제하는 Lambda 함수의 Python 코드입니다.
- `code/layer/boto3_layer` 폴더 — 모든 Lambda 함수에서 공유되는 Boto3 계층을 생성하는 AWS CDK 스택입니다.
- `code/layer/opensearch_layer` 폴더 — 모든 종속성을 설치하여 OpenSearch 인덱스를 생성하는 서버리스 계층을 생성하는 AWS CDK 스택입니다.
- `code/streamlit-app` 폴더 — Amazon ECS에서 컨테이너 이미지로 실행되는 Python 코드
- `code/code_stack.py` — AWS CDK는 AWS 리소스를 생성하는 Python 파일을 구성합니다.
- `app.py` — 대상 AWS 계정에 AWS 리소스를 배포하는 AWS CDK 스택 Python 파일.
- `requirements.txt` — AWS CDK에 설치해야 하는 모든 Python 종속성 목록입니다.
- `cdk.json` — 리소스를 생성하는 데 필요한 값을 제공하는 입력 파일입니다. 또한 컨텍스트/구성 필드에서 솔루션을 적절하게 사용자 지정할 수 있습니다. [사용자 지정에 대한 자세한 내용은 추가 정보 섹션을 참조하십시오.](#)

모범 사례

- 여기에 제공된 코드 예제는 proof-of-concept (PoC) 또는 파일럿 용도로만 사용됩니다. 코드를 프로덕션에 적용하려면 다음 모범 사례를 사용하십시오.
 - [Amazon S3 액세스 로깅](#) 활성화
 - [VPC 흐름 로그](#) 활성화
- Lambda 함수에 대한 모니터링 및 알림을 설정합니다. 자세한 내용은 [Lambda 함수 모니터링 및 문제 해결](#)을 참조하십시오. 모범 사례는 [AWS Lambda 함수 사용에 대한 모범 사례](#)를 참조하십시오.

에픽

로컬 워크스테이션에 AWS 자격 증명 설정

작업	설명	필요한 기술
계정 및 지역의 변수를 내보냅니다.	<p>환경 변수를 사용하여 AWS CDK에 대한 AWS 자격 증명을 제공하려면 다음 명령을 실행합니다.</p> <pre>export CDK_DEFAULT_ACCOUNT=<12-digit AWS account number> export CDK_DEFAULT_REGION=<Region></pre>	AWS DevOps, DevOps 엔지니어
AWS CLI 명명된 프로파일을 설정합니다.	계정에 대해 AWS CLI 명명된 프로필을 설정하려면 구성 및 자격 증명 파일 설정의 지침을 따르십시오.	AWS DevOps, DevOps 엔지니어

환경을 설정합니다.

작업	설명	필요한 기술
리포지토리를 로컬 워크스테이션에 복제합니다.	<p>리포지토리를 복제하려면 터미널에서 다음 명령을 실행합니다.</p> <pre>git clone https://github.com/aws-labs/genai-bedrock-agent-chatbot.git</pre>	DevOps 엔지니어, AWS DevOps
Python 가상 환경을 설정합니다.	다음 명령을 실행하여 Python 가상 환경을 설정합니다.	DevOps 엔지니어, AWS DevOps

작업	설명	필요한 기술
	<pre>cd genai-bedrock-agent-chatbot python3 -m venv .venv source .venv/bin/activate</pre> <p>필요한 종속성을 설정하려면 다음 명령을 실행합니다.</p> <pre>pip3 install -r requirements.txt</pre>	
<p>AWS CDK 환경을 설정합니다.</p>	<p>코드를 AWS CloudFormation 템플릿으로 변환하려면 명령을 실행합니다 <code>cdk synth</code>.</p>	<p>AWS DevOps, DevOps 엔지니어</p>

애플리케이션 구성 및 배포

작업	설명	필요한 기술
<p>계정에 리소스를 배포하십시오.</p>	<p>AWS CDK를 사용하여 AWS 계정에 리소스를 배포하려면 다음을 수행하십시오.</p> <ol style="list-style-type: none"> 복제된 리포지토리의 루트에 있는 <code>cdk.json</code> 파일에서 로깅 파라미터에 대한 입력을 제공합니다. 예제 값은 <code>INFO</code>, <code>DEBUGWARN</code>, 및 <code>ERROR</code>입니다. <p>이러한 값은 Lambda 함수 및 Streamlit 애플리케이션에 대한 로그 수준 메시지를 정의합니다.</p>	<p>DevOps 엔지니어, AWS DevOps</p>

작업	설명	필요한 기술
	<p>2. 복제된 리포지토리의 루트에 있는 <code>cdk.json</code> 파일에는 배포에 사용된 AWS CloudFormation 스택 이름이 들어 있습니다. 기본 스택 이름은 <code>입니다.chatbot-stack</code> 기본 Amazon Bedrock 에이전트 이름은 <code>이고 기본 Amazon Bedrock 에이전트 별칭은 입니다ChatbotBedrockAgent . Chatbot_Agent</code></p> <p>3. 리소스를 배포하려면 <code>cdk deploy</code> 명령을 실행합니다.</p> <p>이 <code>cdk deploy</code> 명령은 레이어 3 구조를 사용하여 문서 및 CSV 데이터세트 파일을 S3 버킷으로 복사하기 위한 여러 Lambda 함수를 생성합니다. 또한 Amazon 베드락 에이전트, 지식 베이스 및 Amazon 베드락 에이전트를 위한 Action group Lambda 함수를 배포합니다.</p> <p>4. AWS 관리 콘솔에 로그인한 다음 https://console.aws.amazon.com/cloudformation/ 에서 CloudFormation 콘솔을 엽니다.</p> <p>5. 스택이 성공적으로 배포되었는지 확인합니다. 지침은 AWS CloudFormation 콘솔</p>	

작업	설명	필요한 기술
	<p>에서의 스택 검토를 참조하십시오.</p> <p>배포에 성공하면 콘솔의 출력 탭에 제공된 URL을 사용하여 채팅 기반 어시스턴트 애플리케이션에 액세스할 수 있습니다. CloudFormation</p>	

솔루션의 모든 AWS 리소스를 정리하십시오.

작업	설명	필요한 기술
AWS 리소스를 제거합니다.	솔루션을 테스트한 후, 리소스를 정리하려면 명령을 실행합니다 <code>cdk destroy</code> .	AWS DevOps, DevOps 엔지니어

관련 리소스

AWS 설명서

- [아마존 베드락 리소스](#):
 - [모델 액세스](#)
 - [기초 모델의 추론 파라미터](#)
 - [아마존 베드락 에이전트](#)
 - [Amazon Bedrock용 지식 베이스](#)
- [Python을 사용하여 Lambda 함수 빌드](#)
- AWS CDK 리소스:
 - [AWS CDK로 시작하기](#)
 - [일반적인 AWS CDK 문제 해결](#)
 - [Python에서 AWS CDK로 작업하기](#)
- [AWS 기반 제너레이티브 AI 애플리케이션 빌더](#)

기타 AWS 리소스

- [Amazon OpenSearch 서버리스용 벡터 엔진](#)

기타 리소스

- [LlamaIndex 문서:](#)
- [간소화된 설명서](#)

추가 정보

자체 데이터로 채팅 기반 어시스턴트를 사용자 지정하세요.

솔루션 배포를 위해 사용자 지정 데이터를 통합하려면 다음과 같은 구조화된 가이드라인을 따르세요. 이러한 단계는 원활하고 효율적인 통합 프로세스를 보장하도록 설계되었으며, 이를 통해 맞춤형 데이터를 사용하여 솔루션을 효과적으로 배포할 수 있습니다.

지식참고 데이터 통합용

데이터 준비

1. `assets/knowledgebase_data_source/` 디렉토리를 찾으세요.
2. 데이터셋을 이 폴더에 넣으세요.

구성 조정

1. `cdk.json` 파일을 엽니다.
2. `context/configure/paths/knowledgebase_file_name` 필드로 이동한 다음 그에 따라 업데이트하십시오.
3. `bedrock_instructions/knowledgebase_instruction` 필드로 이동한 다음 새 데이터셋의 뉘앙스와 컨텍스트를 정확하게 반영하도록 업데이트하세요.

구조적 데이터 통합용

데이터 구성

1. 디렉터리 내에 하위 `assets/data_query_data_source/` 디렉터리 (예 `tabular_data:`) 를 생성합니다.

2. 구조화된 데이터셋 (CSV, JSON, ORC, Parquet 등 허용되는 형식) 를 새로 만든 이 하위 폴더에 넣으세요.
3. 기존 데이터베이스에 연결하는 경우 함수를 업데이트하여 데이터베이스에 연결하세요
`create_sql_engine(). code/lambda/action-lambda/build_query_engine.py`

구성 및 코드 업데이트

1. `cdk.json` 파일에서 `context/configure/paths/athena_table_data_prefix` 필드를 업데이트하여 새 데이터 경로에 맞춥니다.
2. 데이터셋에 맞는 새로운 Text to SQL 예제를 `code/lambda/action-lambda/dynamic_examples.csv` 통합하여 수정하세요.
3. 구조화된 데이터셋의 속성을 `code/lambda/action-lambda/prompt_templates.py` 반영하도록 수정하세요.
4. `cdk.json` 파일에서 `context/configure/bedrock_instructions/action_group_description` 필드를 업데이트하여 Action group Lambda 함수의 목적과 기능을 설명하십시오.
5. `assets/agent_api_schema/artifacts_schema.json` 파일에서 Action group Lambda 함수의 새로운 기능을 설명하십시오.

일반 업데이트

`cdk.json` 파일의 `context/configure/bedrock_instructions/agent_instruction` 섹션에는 새로 통합된 데이터를 고려하여 Amazon Bedrock 에이전트의 의도된 기능 및 설계 목적에 대한 포괄적인 설명을 제공합니다.

Amazon Personalize를 사용하여 개인화되고 순위가 다시 매겨진 추천 생성

작성자: Mason Cahill(AWS), Matthew Chasse(AWS), Tayo Olajide(AWS)

코드 리포지토리: personalize-pet-recommendations	환경: PoC 또는 파일럿	기술: 머신 러닝 및 AI, 클라우드 네이티브, 인프라 DevOps, 서버리스
워크로드: 오픈 소스	AWS 서비스: AWS CloudFormation, 아마존 키네시스 데이터 파이어호스, AWS Lambda, 아마존 Personalize, AWS Step Functions	

요약

이 패턴은 Amazon Personalize를 사용하여 해당 사용자의 실시간 사용자 상호 작용 데이터 수집을 기반으로 재순위 추천을 비롯한 개인화된 권장 사항을 생성하는 방법을 보여줍니다. 이 패턴에 사용된 예제 시나리오는 사용자의 상호 작용(예: 사용자가 방문하는 애완동물)을 기반으로 사용자를 위한 추천을 생성하는 애완동물 입양 웹사이트를 기반으로 합니다. 예제 시나리오를 따르면 Amazon Kinesis Data Streams를 사용하여 상호 작용 데이터를 수집하고, AWS Lambda를 사용하여 권장 사항을 생성하고 권장 사항의 순위를 조정하며, Amazon Data Firehose를 사용하여 Amazon Simple Storage Service (Amazon S3) 버킷에 데이터를 저장하는 방법을 배웁니다. 또한 AWS Step Functions를 사용하여 권장 사항을 생성하는 솔루션 버전(즉, 학습된 모델)을 관리하는 상태 머신을 구축하는 방법도 배웁니다.

사전 조건 및 제한 사항

사전 조건

- [부트스트랩된](#) AWS Cloud Development Kit(AWS CDK)가 있는 활성 [AWS 계정](#)
- 구성된 보안 인증 정보가 포함된 [AWS Command Line Interface\(AWS CLI\)](#)
- [Python 3.9](#)

제품 버전

- Python 3.9
- AWS CDK 2.23.0 이상
- AWS CLI 2.7.27 이상

아키텍처

기술 스택

- 아마존 데이터 파이어호스
- Amazon Kinesis Data Streams
- Amazon Personalize
- Amazon Simple Storage Service(S3)
- AWS Cloud Development Kit(AWS CDK)
- AWS Command Line Interface(AWS CLI)
- AWS Lambda
- AWS Step Functions

대상 아키텍처

다음 다이어그램은 실시간 데이터를 Amazon Personalize로 수집하기 위한 파이프라인을 보여줍니다. 그런 다음 파이프라인은 해당 데이터를 사용하여 사용자를 위한 개인화되고 순위가 조정된 추천을 생성합니다.

이 다이어그램은 다음 워크플로를 보여줍니다.

1. Kinesis Data Streams는 Lambda 및 Firehose에서 처리하기 위해 실시간 사용자 데이터 (예: 방문한 애완동물 같은 이벤트) 를 수집합니다.
2. Lambda 함수는 Kinesis Data Streams의 레코드를 처리하고 API 호출을 통해 레코드의 사용자 상호 작용을 Amazon Personalize의 이벤트 트래커에 추가합니다.
3. 시간 기반 규칙은 Step Functions 상태 머신을 호출하고 Amazon Personalize의 이벤트 추적기의 이벤트를 사용하여 추천 및 순위 재지정 모델에 대한 새 솔루션 버전을 생성합니다.
4. Amazon Personalize [캠페인](#)은 새 [솔루션 버전](#)을 사용하도록 상태 머신에 의해 업데이트됩니다.

5. Lambda는 Amazon Personalize 순위 조정 캠페인을 호출하여 권장 품목 목록의 순위를 다시 매깁니다.
6. Lambda는 Amazon Personalize 추천 캠페인을 호출하여 권장 품목 목록을 검색합니다.
7. Firehose는 이벤트를 과거 데이터로 액세스할 수 있는 S3 버킷에 저장합니다.

도구

AWS 도구

- [AWS Cloud Development Kit\(AWS CDK\)](#)는 AWS 클라우드 인프라를 코드로 정의하고 프로비저닝하는 데 도움이 되는 소프트웨어 개발 프레임워크입니다.
- [AWS Command Line Interface \(AWS CLI\)](#)는 명령줄 셸에서 명령을 사용하여 AWS 서비스와 상호 작용할 수 있는 오픈 소스 도구입니다.
- [Amazon Data Firehose](#)를 사용하면 지원되는 타사 서비스 공급자가 소유한 다른 AWS 서비스, 사용자 지정 HTTP 엔드포인트 및 HTTP 엔드포인트에 실시간 [스트리밍 데이터](#)를 전송할 수 있습니다.
- [Amazon Kinesis Data Streams](#)를 사용하여 대규모 데이터 레코드 스트림을 실시간으로 수집하고 처리할 수 있습니다.
- [AWS Lambda](#)는 서버를 프로비저닝하거나 관리할 필요 없이 코드를 실행하는 데 도움이 되는 컴퓨팅 서비스입니다. 필요할 때만 코드를 실행하며 자동으로 확장이 가능하므로 사용한 컴퓨팅 시간만큼만 비용을 지불합니다.
- [Amazon Personalize](#)는 데이터를 기반으로 사용자를 위한 품목 추천을 생성하는 데 도움이 되는 완전 관리형 기계 학습(ML) 서비스입니다.
- [AWS Step Functions](#)는 Lambda 함수와 기타 AWS 서비스를 결합하여 비즈니스 크리티컬 애플리케이션을 구축할 수 있게 지원하는 서버리스 오케스트레이션 서비스입니다.

기타 도구

- [pytest](#)는 작고 읽기 쉬운 테스트를 작성하기 위한 Python 프레임워크입니다.
- [Python](#)은 범용 컴퓨터 프로그래밍 언어입니다.

코드

이 패턴의 코드는 GitHub [Animal Recommender](#) 리포지토리에서 사용할 수 있습니다. 이 리포지토리의 AWS CloudFormation 템플릿을 사용하여 예제 솔루션의 리소스를 배포할 수 있습니다.

참고: Amazon Personalize 솔루션 버전, 이벤트 추적기 및 캠페인은 기본 [CloudFormation 리소스를 확장하는 사용자 지정 리소스](#) (인프라 내) 로 뒷받침됩니다.

에픽

인프라 생성

작업	설명	필요한 기술
격리된 Python 환경을 생성합니다.	<p>Mac 및 Linux 설치</p> <ol style="list-style-type: none"> 가상 환경을 수동으로 만들려면 터미널에서 <code>\$ python3 -m venv .venv</code> 명령을 실행합니다. init 프로세스가 완료되면 <code>\$ source .venv/bin/activate</code> 명령을 실행하여 가상 환경을 활성화합니다. <p>Windows 설정</p> <p>가상 환경을 수동으로 만들려면 터미널에서 <code>% .venv\Scripts\activate.bat</code> 명령을 실행합니다.</p>	DevOps 엔지니어
CloudFormation 템플릿을 합성하세요.	<ol style="list-style-type: none"> 필요한 종속성을 설치하려면 터미널에서 <code>\$ pip install -r requirements.txt</code> 명령을 실행합니다. AWS CLI에서 다음과 같은 환경 변수를 설정합니다 	DevOps 엔지니어

작업	설명	필요한 기술
	<ul style="list-style-type: none"> • <code>export ACCOUNT_ID=123456789</code> • <code>export CDK_DEPLOY_REGION=us-east-1</code> • <code>export CDK_ENVIRONMENT=dev</code> <p>3. <code>config/{env}.yaml</code> 파일에서 Virtual Private Cloud(VPC) ID와 일치하도록 <code>vpcId</code>를 업데이트합니다.</p> <p>4. 이 코드의 CloudFormation 템플릿을 합성하려면 명령을 실행합니다. <code>\$ cdk synth</code></p> <p>참고: 2단계에서는 <code>CDK_ENVIRONMENT</code> 가 <code>config/{env}.yaml</code> 파일을 참조합니다.</p>	

작업	설명	필요한 기술
리소스를 배포하고 인프라를 생성합니다.	<p>솔루션 리소스를 배포하려면 <code>./deploy.sh</code> 터미널에서 명령을 실행합니다.</p> <p>이 명령은 필수 Python 종속성을 설치합니다. Python 스크립트는 S3 버킷과 AWS Key Management Service(AWS KMS) 키를 생성한 다음, 초기 모델 생성을 위한 시드 데이터를 추가합니다. 마지막으로 스크립트가 <code>cdk deploy</code>를 실행하여 나머지 인프라를 생성합니다.</p> <p>참고: 초기 모델 학습은 스택 생성 중에 이루어집니다. 스택 생성 완료까지 최대 2시간이 걸릴 수 있습니다.</p>	DevOps 엔지니어

관련 리소스

- [동물 추천자 \(\)](#) GitHub
- [AWS CDK 참조 문서](#)
- [Boto3 설명서](#)
- [Amazon Personalize를 사용하여 선택한 비즈니스 지표에 대한 맞춤형 추천 최적화\(AWS 기계계 학습 블로그\)](#)

추가 정보

페이로드 및 응답 예시

Lambda 함수 권장 사항

권장 사항을 검색하려면 다음 형식의 페이로드와 함께 Lambda 함수 권장 사항 에 요청을 제출합니다.

```
{
  "userId": "3578196281679609099",
  "limit": 6
}
```

다음 예제 응답은 동물 그룹 목록을 포함합니다.

```
[{"id": "1-domestic short hair-1-1"},
{"id": "1-domestic short hair-3-3"},
{"id": "1-domestic short hair-3-2"},
{"id": "1-domestic short hair-1-2"},
{"id": "1-domestic short hair-3-1"},
{"id": "2-beagle-3-3"},
```

userId 필드를 생략하면 함수는 일반적인 권장 사항을 반환합니다.

Lambda 함수 순위 재지정

순위 재지정을 사용하려면 Lambda 함수 순위 재지정에 요청을 제출합니다. 페이로드에는 userId의 순위를 다시 매길 모든 항목 ID와 해당 메타데이터가 포함되어 있습니다. 다음 예제 데이터는 animal_species_id(1=cat, 2=dog)에는 Oxford Pets 클래스를 사용하고 animal_age_id 및 animal_size_id에는 정수 1~5를 사용합니다.

```
{
  "userId": "12345",
  "itemMetadataList": [
    {
      "itemId": "1",
      "animalMetadata": {
        "animal_species_id": "2",
        "animal_primary_breed_id": "Saint_Bernard",
        "animal_size_id": "3",
        "animal_age_id": "2"
      }
    },
    {
      "itemId": "2",
      "animalMetadata": {
        "animal_species_id": "1",
        "animal_primary_breed_id": "Egyptian_Mau",
        "animal_size_id": "1",
        "animal_age_id": "1"
      }
    }
  ]
}
```

```

    }
  },
  {
    "itemId":"3",
    "animalMetadata":{
      "animal_species_id":"2",
      "animal_primary_breed_id":"Saint_Bernard",
      "animal_size_id":"3",
      "animal_age_id":"2"
    }
  }
]
}

```

Lambda 함수는 이러한 항목의 순위를 다시 지정한 다음 Amazon Personalize의 직접 응답과 항목 ID가 포함된 정렬된 목록을 반환합니다. 이 목록은 항목이 속한 동물 그룹과 점수의 순위 목록입니다. Amazon Personalize는 [사용자 개인화](#) 및 [개인화 순위](#) 레시피를 사용하여 권장 사항의 각 항목에 대한 점수를 포함합니다. 이러한 점수는 사용자가 다음에 선택할 Amazon Personalize의 상대적 확실성을 나타냅니다. 점수가 높을수록 확실성이 높아집니다.

```

{
  "ranking":[
    "1",
    "3",
    "2"
  ],
  "personalizeResponse":{
    "ResponseMetadata":{
      "RequestId":"a2ec0417-9dcd-4986-8341-a3b3d26cd694",
      "HTTPStatusCode":200,
      "HTTPHeaders":{
        "date":"Thu, 16 Jun 2022 22:23:33 GMT",
        "content-type":"application/json",
        "content-length":"243",
        "connection":"keep-alive",
        "x-amzn-requestid":"a2ec0417-9dcd-4986-8341-a3b3d26cd694"
      },
      "RetryAttempts":0
    },
    "personalizedRanking":[
      {
        "itemId":"2-Saint_Bernard-3-2",
        "score":0.8947961
      }
    ]
  }
}

```

```
    },
    {
      "itemId": "1-Siamese-1-1",
      "score": 0.105204
    }
  ],
  "recommendationId": "RID-d97c7a87-bd4e-47b5-a89b-ac1d19386aec"
}
}
```

Amazon Kinesis 페이로드

Amazon Kinesis로 전송할 페이로드의 형식은 다음과 같습니다.

```
{
  "Partitionkey": "randomstring",
  "Data": {
    "userId": "12345",
    "sessionId": "sessionId4545454",
    "eventType": "DetailView",
    "animalMetadata": {
      "animal_species_id": "1",
      "animal_primary_breed_id": "Russian_Blue",
      "animal_size_id": "1",
      "animal_age_id": "2"
    },
    "animal_id": "98765"
  },
}
}
```

참고: 인증되지 않은 사용자의 경우 `userId` 필드가 제거됩니다.

Amazon에서 사용자 지정 GPU 지원 ML 모델을 트레이닝하고 배포하기 SageMaker

작성자: Ankur Shukla(AWS)

환경: PoC 또는 파일럿

기술: 기계 학습 및 AI, 컨테이너 및 마이크로서비스

AWS 서비스: 아마존 ECS, 아마존 SageMaker

요약

그래픽 처리 장치(GPU) 지원 기계 학습(ML) 모델을 교육 및 배포하려면 NVIDIA GPU의 이점을 완전히 활용하기 위해 특정 환경 변수를 초기 설정하고 초기화해야 합니다. 하지만 환경을 설정하고 Amazon Web Services (AWS) 클라우드의 Amazon SageMaker 아키텍처와 호환되도록 하려면 시간이 많이 걸릴 수 있습니다.

이 패턴은 Amazon을 사용하여 사용자 지정 GPU 지원 ML 모델을 학습하고 구축하는 데 도움이 됩니다. SageMaker 오픈 소스 Amazon 리뷰 데이터 세트를 기반으로 구축된 사용자 지정 CatBoost 모델을 교육하고 배포하는 단계를 제공합니다. 그런 다음 p3.16xlarge Amazon Elastic Compute Cloud(Amazon EC2) 인스턴스에서 성능을 벤치마킹할 수 있습니다.

이 패턴은 조직에서 기존 GPU 지원 ML 모델을 배포하려는 경우에 유용합니다. SageMaker 데이터 사이언티스트는 이 패턴의 단계에 따라 NVIDIA GPU 지원 컨테이너를 만들고 해당 컨테이너에 ML 모델을 배포할 수 있습니다.

사전 조건 및 제한 사항

사전 조건

- 활성 상태의 AWS 계정.
- 모델 아티팩트 및 예측을 저장하기 위한 Amazon Simple Storage Service(S3) 소스 버킷.
- SageMaker 노트북 인스턴스와 Jupyter 노트북에 대한 이해
- 기본 SageMaker 역할 권한, S3 버킷 액세스 및 업데이트 권한, Amazon Elastic Container Registry (Amazon ECR) 에 대한 추가 권한을 포함하는 AWS ID 및 액세스 관리 (IAM) 역할을 생성하는 방법을 이해해야 합니다.

제한 사항

- 이 패턴은 Python으로 작성된 학습 및 배포 코드를 사용하는 감독형 ML 워크로드를 위한 것입니다.

아키텍처

기술 스택

- SageMaker
- Amazon ECR

도구

도구

- [Amazon ECR](#) – Amazon Elastic Container Registry(Amazon ECR)는 안전하고 확장 가능하고 신뢰할 수 있는 AWS 관리형 컨테이너 이미지 레지스트리 서비스입니다.
- [Amazon SageMaker](#) — SageMaker 완전 관리형 ML 서비스입니다.
- [Docker](#)-Docker는 애플리케이션을 신속하게 구축, 테스트 및 배포하기 위한 소프트웨어 플랫폼입니다.
- [Python](#)-파이썬은 프로그래밍 언어입니다.

code

이 패턴의 코드는 [Catboost 및 SageMaker 리포지토리를 사용한 리뷰 분류 모델 GitHub 구현에서](#) 확인할 수 있습니다.

에픽

데이터 준비

작업	설명	필요한 기술
IAM 역할을 생성하고 필요한 정책을 연결합니다.	AWS Management Console에 로그인하여 IAM 콘솔을 열고 새로운 IAM 역할을 생성합니다.	데이터 사이언티스트

작업	설명	필요한 기술
	<p>다. 다음 정책을 IAM 역할에 연결합니다.</p> <ul style="list-style-type: none"> • AmazonEC2ContainerRegistryFullAccess • AmazonS3FullAccess • AmazonSageMakerFullAccess <p>이에 대한 자세한 내용은 Amazon SageMaker 설명서의 노트북 인스턴스 생성을 참조 하십시오.</p>	
<p>SageMaker 노트북 인스턴스를 생성하십시오.</p>	<p>SageMaker 콘솔을 열고 Notebook 인스턴스를 선택한 다음 노트북 인스턴스 만들기를 선택합니다. IAM 역할에는 앞에서 생성한 IAM 역할을 선택합니다. 요구 사항에 따라 노트북 인스턴스를 구성한 다음 노트북 인스턴스 생성을 선택합니다.</p> <p>자세한 단계 및 지침은 Amazon SageMaker 설명서의 노트북 인스턴스 생성을 참조 하십시오.</p>	<p>데이터 사이언티스트</p>

작업	설명	필요한 기술
리포지토리를 복제합니다.	<p>SageMaker 노트북 인스턴스에서 터미널을 열고 다음 명령을 실행하여 Catboost 및 SageMaker 리포지토리로 리뷰 분류 모델 GitHub 구현을 복제합니다.</p> <pre>git clone https://github.com/aws-samples/review-classification-using-catboost-sagemaker.git</pre>	
Jupyter Notebook 시작	<p>사전 정의된 단계가 포함된 Review classification model with Catboost and SageMaker.ipynb Jupyter Notebook을 시작합니다.</p>	데이터 사이언티스트

기능 엔지니어링

작업	설명	필요한 기술
Jupyter Notebook에서 명령을 실행합니다.	<p>Jupyter Notebook을 열고 다음 스토리에서 명령을 실행하여 ML 모델 학습에 필요한 데이터를 준비합니다.</p>	데이터 사이언티스트
S3 버킷에서 데이터를 읽습니다.	<pre>import pandas as pd import csv fname = 's3://amazon-reviews-pds/tsv/amazon_reviews_us_Digital_Video_Download_v1_00.tsv.gz'</pre>	데이터 사이언티스트

작업	설명	필요한 기술
<p>데이터를 전처리합니다.</p>	<pre data-bbox="609 210 1031 388">df = pd.read_csv(fname, sep='\t', delimiter ='\t', error_bad_li nes=False)</pre> <pre data-bbox="609 430 1031 1291">import numpy as np def pre_process(df): df.fillna(value={' review_body': '', 'review_headline': ''}, inplace=True) df.fillna(value={'v erified_purchase': 'Unk'}, inplace=True) df.fillna(0, inplace=True) return df df = pre_process(df) df.review_date = pd.to_datetime(df. review_date) df['target'] = np.where(df['star_ rating']>=4,1,0)</pre> <p data-bbox="592 1333 1031 1606">참고: 이 코드는 'review_b ody' 의 null 값을 빈 문자 열로 바꾸고 'verified _purchase' 열을 “알 수 없 음”을 의미하는 'Unk' 문자열 로 바꿉니다.</p>	<p>데이터 사이언티스트</p>

작업	설명	필요한 기술
<p>데이터를 훈련, 검증, 테스트 데이터 세트로 분할합니다.</p>	<p>대상 레이블의 분포를 분할된 집합에서 동일하게 유지하려면 scikit-learn 라이브러리를 사용하여 샘플링을 계층화해야 합니다.</p> <pre data-bbox="597 489 1027 1843"> from sklearn.model_selection import StratifiedShuffleSplit sss = StratifiedShuffleSplit(n_splits=2, test_size=0.10, random_state=0) sss.get_n_splits(df, df['target']) for train_index, test_index in sss.split(df, df['target']): X_train_val, X_test = df.iloc[train_index], df.iloc[test_index] sss.get_n_splits(X_train_val, X_train_val['target']) for train_index, test_index in sss.split(X_train_val, X_train_val['target']): X_train, X_val = X_train_val.iloc[train_index], X_train_val.iloc[test_index] </pre>	<p>데이터 사이언티스트</p>

Docker 이미지를 빌드하고 실행한 다음 Amazon ECR로 푸시합니다.

작업	설명	필요한 기술
<p>Docker 이미지를 준비하고 푸시합니다.</p>	<p>Jupyter Notebook에서 다음 스토리의 명령을 실행하여 Docker 이미지를 준비하고 Amazon ECR로 푸시합니다.</p>	<p>ML 엔지니어</p>
<p>Amazon ECR 리포지토리를 생성합니다.</p>	<pre> %%sh algorithm_name=sagemaker-catboost-github-gpu-img chmod +x code/train chmod +x code/serve account=\$(aws sts get-caller-identity --query Account --output text) # Get the region defined in the current configuration (default to us-west-2 if none defined) region=\$(aws configure get region) region=\${region:-us-east-1} fullname="\${account}.dkr.ecr.\${region}.amazonaws.com/\${algorithm_name}:latest" aws ecr create-repository --repository- </pre>	<p>ML 엔지니어</p>

작업	설명	필요한 기술
	<pre>name "\${algorithm_name}" > /dev/null</pre>	
Docker 이미지를 로컬로 구축합니다.	<pre>docker build -t "\${algorithm_name}" . docker tag \${algorithm_name} \${fullname}</pre>	ML 엔지니어
Docker 이미지를 실행하고 Amazon ECR에 푸시합니다.	<pre>docker push \${fullname}</pre>	ML 엔지니어

학습

작업	설명	필요한 기술
SageMaker 하이퍼파라미터 조정 작업을 생성합니다.	Jupyter 노트북에서 다음 스토리의 명령을 실행하여 Docker 이미지를 사용하여 SageMaker 하이퍼파라미터 조정 작업을 생성합니다.	데이터 사이언티스트
추정기를 만드세요. SageMaker	Docker 이미지 이름을 사용하여 SageMaker 추정기를 생성합니다 . <pre>import sagemaker as sage from time import gmtime, strftime sess = sage.Session() from sagemaker.tuner import IntegerParameter, CategoricalParameter, ContinuousParameter, HyperparameterTuner</pre>	데이터 사이언티스트

작업	설명	필요한 기술
	<pre> account = sess.boto _session.client('s ts').get_caller_id entity()['Account'] region = sess.boto _session.region_name image = '{}.dkr.e cr.{}.amazonaws.co m/sagemaker-catboo st-github-gpu-img: latest'.format(acc ount, region) tree_hpo = sage.esti mator.Estimator(im age, role, 1, 'ml.p3.16xlarge', train_volume_size = 100, output_path="s3:// {}/sagemaker/DEMO- GPU-Catboost/outpu t".format(bucket), sagemaker_session= sess) </pre>	

작업	설명	필요한 기술
HPO 작업을 생성합니다.	<p>파라미터 범위를 사용하여 하이퍼파라미터 최적화(HPO) 미세 조정 작업을 생성하고 훈련 세트와 검증 세트를 파라미터로 함수에 전달합니다.</p> <pre data-bbox="592 489 1027 1774"> hyperparameter_ranges = {'iterations': IntegerParameter(80000, 130000), 'max_depth': IntegerParameter(6, 10), 'max_ctr_complexity': IntegerParameter(4, 10), 'learning_rate': ContinuousParameter(0.01, 0.5)} objective_metric_name = 'auc' metric_definitions = [{'Name': 'auc', 'Regex': 'auc: ([0-9\\.]+)'}] tuner = HyperparameterTuner(tree_hpo, objective_metric_name, hyperparameter_ranges,</pre>	데이터 사이언티스트

작업	설명	필요한 기술
	<pre>metric_definitions , objective_type='Maximize', max_jobs=50, max_parallel_jobs=2)</pre>	
HPO 작업을 실행합니다.	<pre>train_location = 's3://' + bucket + '/sagemaker/DEMO-GPU-Catboost/data/train/' valid_location = 's3://' + bucket + '/sagemaker/DEMO-GPU-Catboost/data/valid/' tuner.fit({'train': train_location, 'validation': valid_location })</pre>	데이터 사이언티스트
가장 성과가 좋은 훈련 작업을 받습니다.	<pre>import sagemaker as sage from time import gmtime, strftime sess = sage.Session() best_job = tuner.best_training_job()</pre>	데이터 사이언티스트

배치 변환

작업	설명	필요한 기술
<p>모델 예측을 위해 테스트 데이터에 SageMaker 일괄 변환 작업을 생성합니다.</p>	<p>Jupyter 노트북에서 다음 스토리의 명령을 실행하여 SageMaker 하이퍼파라미터 조정 작업에서 모델을 만들고 모델 예측을 위해 테스트 데이터에 대한 SageMaker 일괄 변환 작업을 제출하십시오.</p>	<p>데이터 사이언티스트</p>
<p>모델을 생성하세요. SageMaker</p>	<p>최적의 교육 작업을 사용하여 SageMaker 모델 내에서 모델을 생성하십시오.</p> <pre data-bbox="597 867 1027 1808"> attached_estimator = sage.estimator.Estimator.attach(best_job) output_path = 's3://' + bucket + '/sagemaker/ DEMO-GPU-Catboost/ data/test-predictions/' input_path = 's3://' + bucket + '/sagemaker/ DEMO-GPU-Catboost/ data/test/' transformer = attached_estimator.transformer(instance_count=1, instance_type='ml.p3.16xlarge', </pre>	<p>데이터 사이언티스트</p>

작업	설명	필요한 기술
	<pre> assemble_with='Line', accept='text/csv', max_payload=1, output_path=output_path, env = {'SAGEMAKER_MODEL_SERVER_TIMEOUT' : '3600' }) </pre>	
<p>배치 변환 작업을 생성합니다.</p>	<p>테스트 데이터 세트에 배치 변환 작업을 생성합니다.</p> <pre> transformer.transform(input_path, content_type='text/csv', split_type='Line') </pre>	<p>데이터 사이언티스트</p>

결과 분석

작업	설명	필요한 기술
<p>결과를 읽고 모델의 성능을 평가합니다.</p>	<p>Jupyter Notebook에서 다음 스토리의 명령을 실행하여 결과를 읽고 ROC 곡선 아래 영역 (ROC-AUC) 및 정밀도 재현율 곡선 아래 영역(PR-AUC) 모델</p>	<p>데이터 사이언티스트</p>

작업	설명	필요한 기술
	<p>지표에서 모델의 성능을 평가합니다.</p> <p>이에 대한 자세한 내용은 Amazon Machine Learning(Amazon ML) 설명서의 Amazon Machine Learning 주요 개념을 참조하세요.</p>	
<p>배치 변환 작업 결과를 읽습니다.</p>	<p>배치 변환 작업 결과를 데이터 프레임으로 읽습니다.</p> <pre data-bbox="597 730 1026 1486"> file_name = 's3://' + bucket + '/sagemaker/ DEMO-GPU-Catboost/ data/test-predictions/ file_1.out' results = pd.read_csv(file_name, names=['review_id', 'target', 'score'], sep='\t', escapechar = '\\', quoting=csv.QUOTE_NONE, lineterminator='\n', quotechar='').dropna() </pre>	<p>데이터 사이언티스트</p>

작업	설명	필요한 기술
성능 지표를 평가합니다.	<p>ROC-AUC 및 PR-AUC에서 모델의 성능을 평가합니다.</p> <pre data-bbox="592 346 1031 1831"> from sklearn import metrics import matplotlib import pandas as pd matplotlib.use('agg', warn=False, force=True) from matplotlib import pyplot as plt %matplotlib inline def analyze_results(labels, predictions): precision, recall, thresholds = metrics.p recision_recall_cu rve(labels, predictio ns) auc = metrics.a uc(recall, precision) fpr, tpr, _ = metrics.roc_curve(labels, predictions) roc_auc_score = metrics.roc_auc_sc ore(labels, predictio ns) print('Neural- Nets: ROC auc=%.3f' % (roc_auc_score)) plt.plot(fpr, tpr, label="data 1, auc=" + str(roc_auc_score)) </pre>	데이터 사이언티스트

작업	설명	필요한 기술
	<pre> plt.xlabel('1-Specificity') plt.ylabel('Sensitivity') plt.legend(loc=4) plt.show() lr_precision, lr_recall, _ = metrics.precision_ recall_curve(labels, predictions) lr_auc = metrics.a uc(lr_recall, lr_precision) # summarize scores print('Neural- Nets: PR auc=%.3f' % (lr_auc)) # plot the precision -recall curves no_skill = len(label s[labels==1.0]) / len(labels) plt.plot([0, 1], [no_skill, no_skill] , linestyle='--', label='No Skill') plt.plot(lr_recall , lr_precision, marker='.', label='Ne ural-Nets') # axis labels plt.xlabel('Recall ') plt.ylabel('Precis ion') # show the legend plt.legend() # show the plot </pre>	

작업	설명	필요한 기술
	<pre>plt.show() return auc analyze_results(results['target'].values, results['score'].values)</pre>	

관련 리소스

- [Scikit Docker 컨테이너를 SageMaker 구축하여 아마존에서 Scikit-Learn 모델을 교육하고 호스팅하세요.](#)

추가 정보

다음 목록은 Docker 이미지를 Amazon ECR로 빌드, 실행 및 푸시 에픽에서 실행되는 Dockerfile의 다양한 요소를 보여줍니다.

aws-cli를 사용하여 Python을 설치합니다.

```
FROM amazonlinux:1

RUN yum update -y && yum install -y python36 python36-devel python36-libs python36-tools python36-pip && \
  yum install gcc tar make wget util-linux kmod man sudo git -y && \
  yum install wget -y && \
  yum install aws-cli -y && \
  yum install nginx -y && \
  yum install gcc-c++.noarch -y && yum clean all
```

Python 패키지를 설치합니다.

```
RUN pip-3.6 install --no-cache-dir --upgrade pip && \pip3 install --no-cache-dir --upgrade setuptools && \
```

```
pip3 install Cython && \
pip3 install --no-cache-dir numpy==1.16.0 scipy==1.4.1 scikit-learn==0.20.3
pandas==0.24.2 \
flask gevent gunicorn boto3 s3fs matplotlib joblib catboost==0.20.2
```

CUDA 및 CuDNN을 설치합니다.

```
RUN wget https://developer.nvidia.com/compute/cuda/9.0/Prod/local_installers/
cuda_9.0.176_384.81_linux-run \
&& chmod u+x cuda_9.0.176_384.81_linux-run \
&& ./cuda_9.0.176_384.81_linux-run --tmpdir=/data --silent --toolkit --override \
&& wget https://custom-gpu-sagemaker-image.s3.amazonaws.com/installation/cudnn-9.0-
linux-x64-v7.tgz \
&& tar -xvzf cudnn-9.0-linux-x64-v7.tgz \
&& cp /data/cuda/include/cudnn.h /usr/local/cuda/include \
&& cp /data/cuda/lib64/libcudnn* /usr/local/cuda/lib64 \

&& chmod a+r /usr/local/cuda/include/cudnn.h /usr/local/cuda/lib64/libcudnn* \
&& rm -rf /data/*
```

에 필요한 디렉터리 구조를 생성하십시오. SageMaker

```
RUN mkdir /opt/ml /opt/ml/input /opt/ml/input/config /opt/ml/input/data /opt/ml/input/
data/training /opt/ml/model /opt/ml/output /opt/ml/program
```

NVIDIA 환경 변수를 설정합니다.

```
ENV PYTHONPATH=/opt/program
ENV PYTHONUNBUFFERED=TRUE
ENV PYTHONDONTWRITEBYTECODE=TRUE
ENV PATH="/opt/program:${PATH}"

# Set NVIDIA mount environments
ENV LD_LIBRARY_PATH=/usr/local/nvidia/lib:/usr/local/nvidia/lib64:$LD_LIBRARY_PATH
ENV NVIDIA_VISIBLE_DEVICES="all"
ENV NVIDIA_DRIVER_CAPABILITIES="compute,utility"
ENV NVIDIA_REQUIRE_CUDA "cuda>=9.0"
```

훈련 및 추론 파일을 Docker 이미지에 복사합니다.

```
COPY code/* /opt/program/
```

```
WORKDIR /opt/program
```

테라바이트 SageMaker 규모의 ML 데이터세트의 분산 기능 엔지니어링을 위한 프로세싱 사용

작성자: Chris Boomhower (AWS)

환경: 프로덕션

기술: 기계 학습 및 AI, 빅 데이터

AWS 서비스: 아마존 SageMaker

요약

테라바이트 규모 이상의 많은 데이터 세트는 계층적 폴더 구조로 구성된 경우가 많으며, 데이터 세트에 있는 파일은 때때로 상호 종속성을 공유합니다. 이러한 이유로 기계 학습(ML) 엔지니어와 데이터 과학자는 모델 훈련 및 추론에 사용할 데이터를 준비하기 위해 신중한 설계 결정을 내려야 합니다. 이 패턴은 수동 매크로샤딩 및 마이크로샤딩 기법을 Amazon SageMaker Processing 및 vCPU (가상 CPU) 병렬화와 함께 사용하여 복잡한 빅데이터 ML 데이터세트에 대한 기능 엔지니어링 프로세스를 효율적으로 확장하는 방법을 보여줍니다.

이 패턴은 프로세싱을 위해 매크로샤딩을 여러 시스템에 걸쳐 데이터 디렉토리를 분할하는 것으로 정의하고, 마이크로샤딩은 여러 프로세싱 스레드에 걸쳐 각 시스템의 데이터를 분할하는 것으로 정의합니다. [이 패턴은 Amazon을 MIMIC-III 데이터 세트의 샘플 시계열 파형 레코드와 SageMaker 함께 사용하여 이러한 기법을 보여줍니다. PhysioNet](#) 이 패턴으로 기술을 구현하면 특성 추출의 처리 시간과 비용을 최소화하는 동시에 리소스 사용률과 처리 효율성을 극대화할 수 있습니다. 이러한 최적화는 데이터 유형에 관계없이 Amazon Elastic Compute Cloud (Amazon EC2) 인스턴스 및 vCPU의 분산 SageMaker 프로세싱을 기반으로 유사하고 큰 데이터 세트를 처리합니다.

사전 조건 및 제한 사항

사전 조건

- 자체 데이터 세트에 이 패턴을 구현하려는 경우 SageMaker 노트북 인스턴스 또는 SageMaker 스튜디오에 액세스하세요. SageMaker Amazon을 처음 사용하는 경우 AWS 설명서의 [Amazon SageMaker 시작하기](#)를 참조하십시오.
- SageMaker [PhysioNet MIMIC-III](#) 샘플 데이터로 이 패턴을 구현하려는 경우 스튜디오를 이용하십시오.
- 패턴은 SageMaker 프로세싱을 사용하지만 프로세싱 작업을 SageMaker 실행한 경험이 없어도 됩니다.

제한 사항

- 이 패턴은 상호 종속적인 파일이 포함된 ML 데이터셋에 매우 적합합니다. 이러한 상호 종속성은 수동 매크로샤딩과 여러 개의 단일 인스턴스 처리 SageMaker 작업을 병렬로 실행할 때 가장 큰 이점을 얻습니다. 이러한 상호 종속성이 존재하지 않는 데이터셋의 경우 SageMaker Processing ShardedByS3Key 기능이 매크로샤딩의 더 나은 대안일 수 있습니다. 동일한 프로세싱 작업으로 관리되는 여러 인스턴스로 샤딩된 데이터를 전송하기 때문입니다. 하지만 두 시나리오 모두에서 이 패턴의 마이크로샤딩 전략을 구현하여 인스턴스 vCPU를 가장 잘 활용할 수 있습니다.

제품 버전

- 아마존 SageMaker 파이썬 SDK 버전 2

아키텍처

대상 기술 스택

- Amazon Simple Storage Service(S3)
- 아마존 SageMaker

대상 아키텍처

매크로샤딩 및 분산형 EC2 인스턴스

이 아키텍처에 표시된 10개의 병렬 프로세스는 MIMIC-III 데이터 세트의 구조를 반영합니다. (다이어그램을 단순화하기 위해 프로세스가 타원으로 표시됩니다.) 수동 매크로샤딩을 사용하는 경우 모든 데이터 세트에 유사한 아키텍처가 적용됩니다. MIMIC-III의 경우, 최소한의 노력으로 각 환자 그룹 폴더를 개별적으로 처리하여 데이터 세트의 원시 구조를 유리하게 사용할 수 있습니다. 다음 다이어그램에서 레코드 그룹 블록은 왼쪽에 표시됩니다 (1). 데이터가 분산되어 있다는 점을 고려하면 환자 그룹별로 샤딩하는 것이 합리적입니다.

그러나 다이어그램 (2)의 중간 섹션에서 볼 수 있듯이 환자 그룹별로 수동으로 샤딩한다는 것은 여러 EC2 인스턴스가 포함된 단일 처리 작업 대신 각 환자 그룹 폴더마다 별도의 처리 작업이 필요하다는 것을 의미합니다. MIMIC-III의 데이터에는 이진 파형 파일 및 일치하는 텍스트 기반 헤더 파일이 모두 포함되며, 이진 데이터 추출을 위해 [wfdb 라이브러리](#)에 대한 필수 종속성이 있으므로 특정 환자에 대한 모든 레코드를 동일한 인스턴스에서 사용할 수 있어야 합니다. 각 이진 파형 파일의 관련 헤더 파일도

존재하는지 확인하는 유일한 방법은 수동 샤딩을 구현하여 자체 처리 작업 내에서 각 샤드를 실행하고 처리 작업 입력을 정의할 `s3_data_distribution_type='FullyReplicated'`를 지정하는 것입니다. 또는, 단일 디렉터리에서 모든 데이터를 사용할 수 있고 파일 간에 종속성이 없는 경우에는 여러 EC2 인스턴스 및 `s3_data_distribution_type='ShardedByS3Key'`를 지정하여 단일 처리 작업을 시작하는 것이 더 적합할 수 있습니다. Amazon S3 데이터 배포 유형으로 `ShardedByS3Key` 지정하면 인스턴스 간 데이터 샤딩을 자동으로 SageMaker 관리하도록 지시합니다.

여러 인스턴스를 동시에 실행하면 시간이 절약되므로 각 폴더의 처리 작업을 시작하면 데이터를 전처리하는 비용 효율적인 방법이 됩니다. 추가 비용 및 시간을 절약하기 위해 각 처리 작업 내에서 마이크로샤딩을 사용할 수 있습니다.

마이크로샤딩 및 병렬 vCPU

각 처리 작업 내에서 그룹화된 데이터는 완전 관리형 EC2 인스턴스에서 사용 가능한 모든 vCPU의 사용을 극대화하기 위해 추가로 SageMaker 분할됩니다. 다이어그램 (2)의 중간 섹션에 있는 블록은 각 기본 처리 작업에서 발생하는 상황을 나타냅니다. 환자 기록 폴더의 콘텐츠는 인스턴스에서 사용 가능한 vCPU 수에 따라 평탄화되고 균등하게 분할됩니다. 폴더 콘텐츠가 분할되면 동일한 크기의 파일 집합이 처리를 위해 모든 vCPU에 배포됩니다. 처리가 완료되면 각 vCPU의 결과가 각 처리 작업에 대한 단일 데이터 파일로 결합됩니다.

첨부된 코드에서 이러한 개념은 `src/feature-engineering-pass1/preprocessing.py` 파일의 다음 섹션에 나와 있습니다.

```
def chunks(lst, n):
    """
    Yield successive n-sized chunks from lst.

    :param lst: list of elements to be divided
    :param n: number of elements per chunk
    :type lst: list
    :type n: int
    :return: generator comprising evenly sized chunks
    :rtype: class 'generator'
    """
    for i in range(0, len(lst), n):
        yield lst[i:i + n]

# Generate list of data files on machine
data_dir = input_dir
d_subs = next(os.walk(os.path.join(data_dir, '.')))[1]
```

```

file_list = []
for ds in d_subs:
    file_list.extend(os.listdir(os.path.join(data_dir, ds, '.')))
dat_list = [os.path.join(re.split('_', f)[0].replace('\n', ''), f[:-4]) for f in
    file_list if f[-4:] == '.dat']

# Split list of files into sub-lists
cpu_count = multiprocessing.cpu_count()
splits = int(len(dat_list) / cpu_count)
if splits == 0: splits = 1
dat_chunks = list(chunks(dat_list, splits))

# Parallelize processing of sub-lists across CPUs
ws_df_list = Parallel(n_jobs=-1, verbose=0)(delayed(run_process)(dc) for dc in
    dat_chunks)

# Compile and pickle patient group dataframe
ws_df_group = pd.concat(ws_df_list)
ws_df_group = ws_df_group.reset_index().rename(columns={'index': 'signal'})
ws_df_group.to_json(os.path.join(output_dir, group_data_out))

```

`chunks` 함수는 주어진 목록을 먼저 길이가 `n` 인 일정한 크기의 덩어리로 나누고 이 결과를 생성기로 반환하여 해당 목록을 소비하도록 정의됩니다. 그런 다음에는 존재하는 모든 이진 파형 파일의 목록을 컴파일하여 환자 폴더 전체에 걸쳐 데이터를 평탄화합니다. 이렇게 하면 EC2 인스턴스에서 사용할 수 있는 vCPU 수가 구해집니다. `chunks`를 호출하여 이진 파형 파일 목록을 이러한 vCPU에 균등하게 분할한 다음, [joblib의 Parallel 클래스](#)를 사용하여 각 파형 하위 목록을 자체 vCPU에서 처리합니다. 결과는 처리 작업에 의해 단일 데이터 프레임 목록으로 자동 결합되며 SageMaker , 작업 완료 시 Amazon S3에 기록하기 전에 추가 처리가 이루어집니다. 이 예제에서는 프로세싱 작업에 의해 Amazon S3에 10개의 파일이 기록되어 있습니다(각 작업당 하나).

초기 프로세싱 작업이 모두 완료되면 다이어그램 (3)의 오른쪽 블록에 표시된 보조 프로세싱 작업이 각 기본 프로세싱 작업에서 생성된 출력 파일을 결합하고 결합된 출력을 Amazon S3 (4)에 씁니다.

도구

도구

- [Python](#) - 이 패턴에 사용되는 샘플 코드는 Python(버전 3)입니다.
- [SageMaker SageMaker Studio](#) — Amazon Studio는 기계 학습 모델을 구축, 교육, 디버그, 배포 및 모니터링할 수 있는 기계 학습을 위한 웹 기반 통합 개발 환경 (IDE) 입니다. Studio 내에서 Jupyter 노트북을 사용하여 SageMaker 처리 작업을 실행합니다. SageMaker

- [SageMaker 처리](#) — Amazon SageMaker Processing은 데이터 처리 워크로드를 실행하는 간소화된 방법을 제공합니다. 이 패턴에서는 SageMaker 처리 작업을 사용하여 기능 엔지니어링 코드를 대규모로 구현합니다.

코드

첨부된 .zip 파일은 이 패턴의 전체 코드를 제공합니다. 다음 섹션에서는 이 패턴에 대한 아키텍처를 구축하는 단계를 설명합니다. 각 단계는 첨부 파일의 샘플 코드로 설명됩니다.

에픽

SageMaker Studio 환경을 설정하세요.

작업	설명	필요한 기술
Amazon SageMaker 스튜디오에 액세스하세요.	Amazon SageMaker 설명서 에 제공된 지침에 따라 AWS 계정으로 SageMaker Studio에 온보딩하십시오.	데이터 사이언티스트, ML 엔지니어
wget 유틸리티를 설치합니다.	새 SageMaker Studio 구성으로 온보딩했거나 이전에 Studio에서 이러한 유틸리티를 사용한 적이 없는 경우 wget을 설치하십시오. SageMaker 설치하려면 SageMaker Studio 콘솔에서 터미널 창을 열고 다음 명령을 실행합니다. <pre>sudo yum install wget</pre>	데이터 사이언티스트, ML 엔지니어
샘플 코드를 다운로드하고 압축을 풉니다.	첨부 파일 섹션에서 attachments.zip 파일을 다운로드합니다. 터미널 창에서 파일을 다운로드한 폴더로 이동하여 다음과 같이 내용을 추출합니다.	데이터 사이언티스트, ML 엔지니어

작업	설명	필요한 기술
	<pre>unzip attachment.zip</pre> <p>.zip 파일을 추출한 위치로 이동하여 Scaled-Processing.zip 파일의 압축을 풉니다.</p> <pre>unzip Scaled-Processing.zip</pre>	
physionet.org에서 샘플 데이터 세트를 다운로드하여 Amazon S3에 업로드합니다.	Scaled-Processing 파일이 들어 있는 폴더 내에서 get_data.ipynb Jupyter Notebook을 실행합니다. 이 노트북은 physionet.org 에서 샘플 MIMIC-III 데이터세트를 다운로드하여 Amazon S3의 스튜디오 세션 버킷에 업로드합니다. SageMaker	데이터 사이언티스트, ML 엔지니어

첫 번째 전처리 스크립트 구성

작업	설명	필요한 기술
모든 하위 디렉토리의 파일 계층 구조를 평탄화합니다.	MIMIC-III와 같은 대규모 데이터 세트에서는 논리적 상위 그룹 내에서도 파일이 여러 하위 디렉터리에 분산되는 경우가 많습니다. 다음 코드에서 볼 수 있듯이 모든 하위 디렉터리의 모든 그룹 파일을 평탄화하도록 스크립트를 구성해야 합니다.	데이터 사이언티스트, ML 엔지니어

작업	설명	필요한 기술
	<pre data-bbox="609 226 1015 961"> # Generate list of .dat files on machine data_dir = input_dir d_subdirs = next(os.walk(os.path.join(data_dir, '.'))) file_list = [] for ds in d_subdirs: file_list.extend(os.listdir(os.path.join(data_dir, ds, '.'))) dat_list = [os.path.join(re.split('_', f)[0].replace('\n', ''), f[:-4]) for f in file_list if f[-4:] == '.dat'] </pre> <p data-bbox="592 997 998 1270">참고 이 에픽의 예제 코드 스니펫은 첨부 파일에 제공된 src/feature-engineering-pass1/preprocessing.py 파일에서 가져온 것입니다.</p>	

작업	설명	필요한 기술
<p>vCPU 수에 따라 파일을 하위 그룹으로 나눕니다.</p>	<p>스크립트를 실행하는 인스턴스에 있는 vCPU 수에 따라 파일을 균일한 크기의 하위 그룹 또는 청크로 나누어야 합니다. 이 단계에서는 다음과 비슷한 코드를 구현할 수 있습니다.</p> <pre data-bbox="597 537 1027 976"> # Split list of files into sub-lists cpu_count = multiprocessing.cpu_count() splits = int(len(dat_list) / cpu_count) if splits == 0: splits = 1 dat_chunks = list(chunks(dat_list, splits)) </pre>	<p>데이터 사이언티스트, ML 엔지니어</p>
<p>vCPU 전반의 하위 그룹 처리를 병렬화합니다.</p>	<p>모든 서브그룹을 병렬로 처리하도록 스크립트 로직을 구성해야 합니다. 이렇게 하려면 다음과 같이 Joblib 라이브러리의 Parallel 클래스와 delayed 메서드를 사용하십시오.</p> <pre data-bbox="597 1373 1027 1732"> # Parallelize processing of sub-lists across CPUs ws_df_list = Parallel(n_jobs=-1, verbose=0)(delayed(run_process)(dc) for dc in dat_chunks) </pre>	<p>데이터 사이언티스트, ML 엔지니어</p>

작업	설명	필요한 기술
Amazon S3에 단일 파일 그룹 출력을 저장합니다.	<p>병렬 vCPU 처리가 완료되면 각 vCPU의 결과를 결합하여 파일 그룹의 S3 버킷 경로에 업로드해야 합니다. 이 단계에서는 다음과 비슷한 코드를 사용할 수 있습니다.</p> <pre># Compile and pickle patient group dataframe ws_df_group = pd.concat (ws_df_list) ws_df_group = ws_df_group. reset_index().re name(columns={'index': 'signal'}) ws_df_group.to_j son(os.path.join(o utput_dir, group_dat a_out))</pre>	데이터 사이언티스트, ML 엔지니어

두 번째 전처리 스크립트 구성

작업	설명	필요한 기술
첫 번째 스크립트를 실행한 모든 프로세싱 작업에서 생성된 데이터 파일을 결합합니다.	<p>이전 스크립트는 데이터셋의 파일 그룹을 SageMaker 처리하는 각 처리 작업에 대해 단일 파일을 출력합니다. 다음으로, 이러한 출력 파일을 단일 객체로 결합하고 Amazon S3에 단일 출력 데이터 세트를 작성해야 합니다. 이는 첨부 파일에 제공된 <code>src/feature-engineering-pass1p5/preprocessing.py</code> 파일</p>	데이터 사이언티스트, ML 엔지니어

작업	설명	필요한 기술
	<p>에 다음과 같이 설명되어 있습니다.</p> <pre> def write_parquet(wavs _df, path): """ Write waveform summary dataframe to S3 in parquet format. :param wavs_df: waveform summary dataframe :param path: S3 directory prefix :type wavs_df: pandas dataframe :type path: str :return: None """ extra_args = {"ServerSideEncryp tion": "aws:kms"} wr.s3.to_parquet(df=wavs_df, path=path, compressi on='snappy', s3_additi onal_kwargs=extra_ args) def combine_data(): """ Get combined data and write to parquet. :return: waveform summary dataframe </pre>	

작업	설명	필요한 기술
	<pre> :rtype: pandas dataframe """ wavs_df = get_data() wavs_df = normalize _signal_names(wavs _df) write_parquet(wavs _df, "s3://{}/{}/" {}.format(buck et_xform, dataset_p refix, pass1p5ou t_data)) return wavs_df wavs_df = combine_d ata() </pre>	

프로세싱 작업 실행

작업	설명	필요한 기술
<p>첫 번째 프로세싱 작업을 실행합니다.</p>	<p>매크로샤딩을 수행하려면 각 파일 그룹에 대해 별도의 프로세싱 작업을 실행합니다. 각 작업에서 첫 번째 스크립트가 실행되므로 각 프로세싱 작업 내에서 마이크로샤딩이 수행됩니다. 다음 코드는 다음 스프레드(notebooks/FeatExtract_Pass1.ipynb 에 포함됨)의 각 파일 그룹 디렉터리에 대해 프로세싱 작업을 시작하는 방법을 보여줍니다.</p>	<p>데이터 사이언티스트, ML 엔지니어</p>

작업	설명	필요한 기술
	<pre> pat_groups = list(range(30,40)) ts = str(int(time.time())) for group in pat_groups: sklearn_processor = SKLearnProcessor(framework_version=' 0.20.0', role=role, instance_ type='ml.m5.4xlarge', instance_ count=1, volume_si ze_in_gb=5) sklearn_processor. run(code='../src/ feature-engineering- pass1/preprocessing.p y', job_name= '-'.join(['scaled- processing-p1', str(group), ts]), arguments=["input_pa th", "/opt/ml/ processing/input", "output_p ath", "/opt/ml/ processing/output", </pre>	

작업	설명	필요한 기술
	<pre> "group_data_out", "ws_df_group.json"], inputs= [ProcessingInput(source=f's3://{sess.default_bucket()}/data_inputs/{group}', destination='/opt/ml/processing/input', s3_data_distribution_type='FullyReplicated')], outputs= [ProcessingOutput(source='/opt/ml/processing/output', destination=f's3://{sess.default_bucket()}/data_outputs/{group}')], wait=False) </pre>	

작업	설명	필요한 기술
<p>두 번째 처리 작업을 실행합니다.</p>	<p>첫 번째 처리 작업 세트에서 생성된 출력을 결합하고 사전 처리를 위한 추가 계산을 수행하려면 단일 SageMaker 처리 작업을 사용하여 두 번째 스크립트를 실행합니다. 다음 코드는 이를 보여줍니다(notebooks/FeatExtract_Pass1p5.ipynb 에 포함됨).</p> <pre data-bbox="597 682 1027 1808"> ts = str(int(time.time())) bucket = sess.default_bucket() sklearn_processor = SKLearnProcessor(framework_version=' 0.20.0', role=role, instance_ type='ml.t3.2xlarge', instance_ count=1, volume_si ze_in_gb=5) sklearn_processor.run(code='../src/feature-engineering-pass1p5/preprocessing.py', job_name='-'.join(['scaled-processing', 'p1p5', ts]), </pre>	<p>데이터 사이언티스트, ML 엔지니어</p>

작업	설명	필요한 기술
	<pre> arguments=['bucket ', bucket, 'passlout _prefix', 'data_out puts', 'passlout _data', 'ws_df_gr oup.json', 'pass1p50 ut_data', 'waveform _summary.parquet', 'statsdat a_name', 'signal_s tats.csv'], wait=True) </pre>	

관련 리소스

- [퀵 스타트를 사용하여 Amazon SageMaker 스튜디오에 온보딩](#) (SageMaker 설명서)
- [프로세스 데이터](#) (SageMaker 문서)
- [scikit-learn을 사용한 데이터 처리](#) (문서) SageMaker
- [joblib.Parallel 설명서](#)
- Moody, B., Moody, G., Villarroel, M., Clifford, G. D., Silva, I. (2020). [MIMIC-III 파형 데이터베이스](#)(버전 1.0). PhysioNet.
- Johnson, A. E. W., Pollard, T. J., Shen, L., Lehman, L. H., Feng, M., Ghassemi, M., Moody, B., Szolovits, P., Celi, L. A., Mark, R. G. (2016). [MIMIC-III\(무료로 이용할 수 있는 중환자 치료 데이터베이스\)](#) Scientific Data, 3, 160035.
- [MIMIC-III 파형 데이터베이스 라이선스](#)

첨부

이 문서와 관련된 추가 콘텐츠에 액세스하려면 [attachment.zip](#) 파일의 압축을 풉니다.

Flask와 Elastic Beanstalk를 사용하여 AI/ML 모델 결과를 시각화합니다.

작성자: Chris Caudill(AWS) 및 Durga Sury(AWS)

환경: PoC 또는 파일럿

기술: 기계 학습 및 AI, 분석
DevOps, 웹 및 모바일 앱

워크로드: 오픈 소스

AWS 서비스: Amazon
Comprehend, AWS Elastic
Beanstalk

요약

인공 지능 및 기계 학습(AI/ML) 서비스의 결과를 시각화하려면 개발자와 엔지니어가 사용자 지정해야 하는 복잡한 API 직접 호출이 필요한 경우가 많습니다. 분석가가 새 데이터 세트를 빠르게 탐색하려는 경우 이는 단점이 될 수 있습니다.

사용자가 자신의 데이터를 업로드하고 대시보드에서 모델 결과를 시각화할 수 있는 웹 기반 사용자 인터페이스(UI)를 사용함으로써 서비스의 접근성을 높이고 더 인터랙티브한 형태의 데이터 분석을 제공할 수 있습니다.

이 패턴은 [Flask](#) 및 [Plotly](#)를 사용하여 Amazon Comprehend를 사용자 지정 웹 애플리케이션과 통합하고 사용자가 제공한 데이터에서 감정과 개체를 시각화합니다. 또한 이 패턴은 Elastic Beanstalk를 사용하여 애플리케이션을 배포하는 절차를 제공하기도 합니다. [Amazon Web Services \(AWS\) AI](#) 서비스를 사용하거나 엔드포인트에서 호스팅되는 사용자 지정 학습 모델 (예: [Amazon SageMaker 엔드포인트](#))을 사용하여 애플리케이션을 조정할 수 있습니다.

사전 조건 및 제한 사항

사전 조건

- 활성 상태의 AWS 계정.
- 로컬 시스템에 설치 및 구성된 Command Line Interface(CLI). 이에 대한 자세한 내용은 CLI 설명서에서 [기본 사항 구성](#)을 참조하십시오. AWS Cloud9 통합 개발 환경(IDE)을 사용할 수도 있습니다. 이

에 대한 자세한 내용은 Cloud9 설명서에서 [Cloud9에 관한 Python 자습서](#) 및 [Cloud9 IDE의 실행 중인 애플리케이션 미리보기](#)를 참조하십시오.

- Flask의 웹 애플리케이션 프레임워크에 대한 이해. Flask에 대한 자세한 내용은 Flask 설명서에서 [퀵 스타트](#)를 참조하십시오.
- Python 버전 3.6 이상이 설치 및 구성되었습니다. Elastic Beanstalk 설명서에서 [Python 개발 환경 설정](#)의 지침을 따라 Python을 설치할 수 있습니다.
- Elastic Beanstalk Command Line Interface(EB CLI)가 설치 및 구성되었습니다. 이에 대한 자세한 내용은 Elastic Beanstalk 설명서의 [EB CLI 설치](#) 및 [EB CLI 구성](#)을 참조하십시오.

제한 사항

- 이 패턴의 Flask 애플리케이션은 단일 텍스트 열을 사용하고 200행으로 제한되는 .csv 파일과 함께 작동하도록 설계되었습니다. 애플리케이션 코드는 다른 파일 유형 및 데이터 볼륨을 처리하도록 조정할 수 있습니다.
- 애플리케이션은 데이터 보존을 고려하지 않으며, 업로드된 사용자 파일을 수동으로 삭제할 때까지 계속 집계합니다. 영구 객체 스토리지를 위해 애플리케이션을 Amazon Simple Storage Service(Amazon S3)와 통합하거나, 서버리스 키 값 스토리지를 위해 Amazon DynamoDB와 같은 데이터베이스를 사용할 수 있습니다.
- 애플리케이션은 영어로 된 문서만 고려합니다. 하지만 Amazon Comprehend를 사용하여 문서의 기본 언어를 감지할 수 있습니다. 각 작업에 지원되는 언어에 대한 자세한 내용은 Amazon Comprehend 설명서에서 [API 참조](#)를 참조하십시오.
- 일반적인 오류와 해결 방법이 포함된 문제 해결 목록은 추가 정보 섹션에서 확인할 수 있습니다.

아키텍처

Flask 애플리케이션 아키텍처

Flask는 Python에서 웹 애플리케이션을 개발하기 위한 경량 프레임워크입니다. Python의 강력한 데이터 처리 기능과 풍부한 웹 UI를 결합하도록 설계되었습니다. 패턴의 Flask 애플리케이션은 사용자가 데이터를 업로드하고 추론을 위해 Amazon Comprehend로 데이터를 전송한 다음 결과를 시각화하는 웹 애플리케이션을 빌드하는 방법을 보여줍니다. 애플리케이션의 구조는 다음과 같습니다.

- `static`— 웹 UI를 지원하는 모든 정적 파일 (예: CSS JavaScript, 이미지) 을 포함합니다.
- `templates` — 애플리케이션의 모든 HTML 페이지를 포함합니다.
- `userData` — 업로드된 사용자 데이터를 저장합니다.

- `application.py` — Flask 애플리케이션 파일
- `comprehend_helper.py` — Amazon Comprehend에 대한 API 직접 호출 함수
- `config.py` — 애플리케이션 구성 파일
- `requirements.txt` — 애플리케이션에 필요한 Python 종속성

`application.py` 스크립트에는 4개의 Flask 경로로 구성된 웹 애플리케이션의 핵심 기능이 포함되어 있습니다. 다음 다이어그램은 이러한 Flask 경로를 보여줍니다.

- `/`(은)는 애플리케이션의 루트이며 사용자를 `upload.html` 페이지(`templates` 디렉토리에 저장되어 있음)로 디렉션합니다.
- `/saveFile`(은)는 사용자가 파일을 업로드한 후 호출되는 경로입니다. 이 경로는 사용자가 업로드한 파일이 포함된 HTML 양식을 통해 POST 요청을 받습니다. 파일은 `userData` 디렉토리에 저장되며 경로는 사용자를 `/dashboard` 경로로 리디렉션합니다.
- `/dashboard`(은)는 사용자를 `dashboard.html` 페이지로 보냅니다. 이 페이지의 HTML 내에서 `/data` 경로에서 데이터를 읽는 JavaScript 코드를 실행한 다음 페이지에 대한 시각화를 작성합니다. `static/js/core.js`
- `/data`(은)는 대시보드에서 시각화할 데이터를 제시하는 JSON API입니다. 이 경로는 사용자가 제공한 데이터를 읽고 `comprehend_helper.py` 내 함수를 사용하여 사용자 데이터를 Amazon Comprehend로 전송해서 감정 분석 및 명명된 개체 인식(NER)을 수행합니다. Amazon Comprehend의 응답은 형식이 지정되고 JSON 객체로 반환됩니다.

배포 아키텍처

클라우드의 Elastic Beanstalk를 사용하여 배포되는 애플리케이션의 설계 고려사항에 대한 자세한 내용은 Elastic Beanstalk 설명서에서 [설계 고려사항](#)을 참조하십시오.

기술 스택

- Amazon Comprehend
- Elastic Beanstalk
- Flask

자동화 및 툴링

Elastic Beanstalk 배포는 로드 밸런서 및 오토 스케일링 그룹을 사용하여 자동으로 설정됩니다. 추가 구성 옵션에 대해서는 Elastic Beanstalk 설명서에서 [Elastic Beanstalk 환경 구성](#)을 참조하십시오.

도구

- [Command Line Interface\(CLI\)](#) – CLI는 AWS의 모든 부분과 상호작용할 수 있도록 일관성 있는 인터페이스를 제공하는 통합 도구입니다.
- [Amazon Comprehend](#) -Amazon Comprehend는 특별한 전처리를 요구하지 않고 자연어 처리(NLP)를 사용하여 문서 내용에 대한 인사이트를 추출합니다.
- [Elastic Beanstalk](#) –Elastic Beanstalk를 사용하면 애플리케이션을 실행하는 인프라에 대해 자세히 알지 못해도 클라우드에서 애플리케이션을 신속하게 배포하고 관리할 수 있습니다.
- [Elastic Beanstalk CLI \(EB CLI\)](#) – EB CLI는 로컬 리포지토리에서 환경 생성, 업데이트 및 모니터링을 단순화하는 대화형 명령을 제공하는 Elastic Beanstalk용 명령줄 인터페이스입니다.
- [Flask](#) — Flask 프레임워크는 Python을 사용하여 데이터 처리 및 API 호출을 수행하고 Plotly를 사용하여 대화형 웹 시각화를 제공합니다.

code

이 패턴의 코드는 플라스크와 AWS [Elastic Beanstalk 리포지토리를 사용한 AI/ML 모델 결과 GitHub 시각화에서](#) 사용할 수 있습니다.

에픽

Flask 애플리케이션 설정

작업	설명	필요한 기술
리포지토리를 복제합니다. GitHub	다음 명령을 실행하여 Flask 및 AWS Elastic Beanstalk 리포지토리를 사용하여 GitHub 시각화 AI/ML 모델 결과에서 애플리케이션 코드를 가져옵니다. git clone git@github.com:aws-samples/	개발자

작업	설명	필요한 기술
	<p>aws-comprehend-elasticbeanstalk-for-flask.git</p> <p>참고: SSH 키는 를 사용하여 구성해야 합니다. GitHub</p>	
<p>Python 모듈을 설치합니다.</p>	<p>리포지토리를 복제하고 나면 새 로컬 aws-comprehend-elasticbeanstalk-for-flask 디렉터리가 생성됩니다. 이 디렉터리의 requirements.txt 파일에는 애플리케이션을 실행하는 Python 모듈과 버전이 들어 있습니다. 다음 명령을 사용하여 모듈을 설치합니다.</p> <pre>cd aws-comprehend-elasticbeanstalk-for-flask</pre> <pre>pip install -r requirements.txt</pre>	<p>Python 개발자</p>

작업	설명	필요한 기술
로컬로 애플리케이션 테스트합니다.	<p>다음 명령을 실행하여 Flask 서버를 시작합니다.</p> <pre>python application.py</pre> <p>이를 통해 실행 중인 서버에 대한 정보를 반환합니다. 브라우저를 열고 <code>http://localhost:5000</code> 을 방문하여 애플리케이션에 액세스할 수 있어야 합니다.</p> <p>참고: AWS Cloud9 IDE에서 애플리케이션을 실행하는 경우 <code>application.py</code> 파일의 <code>application.run()</code> 명령을 다음과 같은 줄로 바꿔야 합니다.</p> <pre>application.run(host=os.getenv('IP', '0.0.0.0'), port=int(os.getenv('PORT', 8080)))</pre> <p>배포하기 전에 이 변경 내용을 되돌려야 합니다.</p>	Python 개발자

Elastic Beanstalk에 애플리케이션 배포

작업	설명	필요한 기술
Elastic Beanstalk 애플리케이션을 시작합니다.	<p>프로젝트를 Elastic Beanstalk 애플리케이션으로 시작하려면 애플리케이션의 루트 디렉터리에서 다음의 명령을 실행합니다.</p>	아키텍트, 개발자

작업	설명	필요한 기술
	<pre>eb init -p python-3.6 comprehend_flask -- region us-east-1</pre> <p>중요:</p> <ul style="list-style-type: none"> • comprehend_flask (은) 는 Elastic Beanstalk 애플리케이션의 이름이며, 요구 사항에 따라 변경할 수 있습니다. • 리전을 원하는 리전으로 바꿀 수 있습니다. 리전을 지정하지 않으면 CLI의 기본 리전이 사용됩니다. • 이 애플리케이션은 Python 버전 3.6으로 제작되었습니다. 다른 Python 버전을 사용하는 경우 오류가 발생할 수 있습니다. <p>더 많은 배포 구성 옵션을 보려면 eb init -i 명령을 실행합니다.</p>	

작업	설명	필요한 기술
Elastic Beanstalk 환경을 배포합니다.	<p>애플리케이션의 루트 디렉터리에서 다음의 명령을 실행합니다.</p> <pre>eb create comprehend-flask-env</pre> <p>참고: <code>comprehend-flask-env</code> (은)는 Elastic Beanstalk 환경의 이름이며, 요구 사항에 따라 변경할 수 있습니다. 이름은 문자, 숫자 및 대시만 포함할 수 있습니다.</p>	아키텍트, 개발자

작업	설명	필요한 기술
<p>Amazon Comprehend를 사용할 수 있도록 배포를 승인합니다.</p>	<p>애플리케이션을 성공적으로 배포할 수 있더라도, Amazon Comprehend에 대한 액세스 권한과 함께 배포를 제공해야 합니다. <code>ComprehendFullAccess</code> (은)는 Amazon Comprehend에 API를 호출할 수 있는 권한을 배포된 애플리케이션에 제공하는 관리형 정책입니다.</p> <p>다음 명령을 실행하여 <code>ComprehendFullAccess</code> 정책을 <code>aws-elasticbeanstalk-ec2-role</code> (이 역할은 배포의 Amazon Elastic Compute Cloud(Amazon EC2) 인스턴스에 대하여 자동으로 생성됨)에 연결합니다.</p> <pre>aws iam attach-role-policy --policy-arn arn:aws:iam::aws:policy/ComprehendFullAccess --role-name aws-elasticbeanstalk-ec2-role</pre> <p>중요: <code>aws-elasticbeanstalk-ec2-role</code> (은)는 애플리케이션 배포 시 생성됩니다. Identity and Access Management(IAM) 정</p>	<p>개발자, 보안 아키텍트</p>

작업	설명	필요한 기술
	책을 연결하려면 배포 프로세스를 완료해야 합니다.	
배포된 애플리케이션을 방문합니다.	<p>애플리케이션이 성공적으로 배포되면 <code>eb open</code> 명령을 실행하여 해당 애플리케이션을 방문할 수 있습니다.</p> <p><code>eb status</code> 명령을 실행하여 배포에 대한 세부 정보를 받을 수도 있습니다. 배포 URL은 CNAME 아래에 나열되어 있습니다.</p>	아키텍트, 개발자

(선택 사항) 애플리케이션을 ML 모델에 사용자 지정합니다.

작업	설명	필요한 기술
새 모델에 액세스할 수 있도록 Elastic Beanstalk를 승인합니다.	<p>Elastic Beanstalk에 새 모델 엔드포인트에 필요한 액세스 권한이 있는지 확인합니다. 예를 들어 Amazon SageMaker 엔드포인트를 사용하는 경우 배포에 엔드포인트를 호출할 권한이 있어야 합니다.</p> <p>이에 대한 자세한 내용은 Amazon SageMaker 설명서를 참조하십시오. InvokeEndpoint</p>	개발자, 보안 아키텍트
사용자 데이터를 새 모델로 보냅니다.	<p>이 애플리케이션에서 기본 ML 모델을 변경하려면 다음 파일을 변경해야 합니다.</p> <ul style="list-style-type: none"> <code>comprehend_helper.py</code> — 이것은 Amazon 	데이터 사이언티스트

작업	설명	필요한 기술
	<p>Comprehend에 연결하여 응답을 처리하고 최종 결과를 애플리케이션에 반환하는 Python 스크립트입니다. 이 스크립트에서는 데이터를 클라우드의 다른 AI 서비스로 라우팅하거나 사용자 지정 모델 엔드포인트로 보낼 수 있습니다. 또한 이 패턴의 논리적 분리와 재사용성을 위해 이 스크립트에서 결과의 형식을 지정하는 것이 좋습니다.</p> <ul style="list-style-type: none"> • <code>application.py</code> — <code>comprehend_helper.py</code> 스크립트나 함수의 이름을 변경하는 경우 해당 변경 내용을 반영하도록 애플리케이션 <code>application.py</code> 스크립트를 업데이트해야 합니다. 	

작업	설명	필요한 기술
대시보드 시각화를 업데이트합니다.	<p>일반적으로 새 ML 모델을 통합하면 새 결과를 반영하도록 시각화를 업데이트해야 합니다. 이 변경은 다음과 같이 이루어졌습니다.</p> <ul style="list-style-type: none"> • <code>templates/dashboard.html</code> — 사전 빌드된 애플리케이션은 두 가지 기본 시각화만 담당합니다. 이 파일에서 페이지의 전체 레이아웃을 조정할 수 있습니다. • <code>static/js/core.js</code> — 이 스크립트는 Flask 서버의 <code>/data</code> 경로의 형식화된 출력을 캡처하고, Plotly를 사용하여 시각화를 만듭니다. 페이지의 차트를 추가하거나 업데이트할 수 있습니다. 	웹 개발자

(선택 사항)업데이트된 애플리케이션을 배포

작업	설명	필요한 기술
애플리케이션의 요구 사항 파일을 업데이트합니다.	<p>변경 내용을 Elastic Beanstalk에 보내기 전에 애플리케이션의 루트 디렉터리에서 다음 명령을 실행하여 새 Python 모듈을 반영하도록 <code>requirements.txt</code> 파일을 업데이트합니다.</p> <pre>pip freeze > requirements.txt</pre>	Python 개발자

작업	설명	필요한 기술
Elastic Beanstalk 환경을 재배포합니다.	<p>애플리케이션 변경 사항이 Elastic Beanstalk 배포에 반영되도록 하려면 애플리케이션의 루트 디렉터리로 이동하여 다음 명령을 실행합니다.</p> <pre>eb deploy</pre> <p>그러면 최신 버전의 애플리케이션 코드가 기존 Elastic Beanstalk 배포로 전송됩니다.</p>	시스템 관리자, 아키텍트

관련 리소스

- [Amazon API Gateway와 AWS Lambda를 사용하여 아마존 SageMaker 모델 엔드포인트를 호출합니다.](#)
- [Elastic Beanstalk에 Flask 애플리케이션 배포](#)
- [EB CLI 명령 참조](#)
- [Python 개발 환경 설정](#)

추가 정보

문제 해결 목록

다음은 6 가지 일반적인 오류와 해결 방법입니다.

오류 1

```
Unable to assume role "arn:aws:iam::xxxxxxxxxx:role/aws-elasticbeanstalk-ec2-role".
Verify that the role exists and is configured correctly.
```

해결 방법: `eb create` 실행 시 이 오류가 발생하면 Elastic Beanstalk 콘솔에서 샘플 애플리케이션을 생성하여 기본 인스턴스 프로파일을 생성합니다. 이에 대한 자세한 내용은 Elastic Beanstalk 설명서에서 [Elastic Beanstalk 환경 생성](#)을 참조하십시오.

오류 2

```
Your WSGIPath refers to a file that does not exist.
```

해결 방법: Elastic Beanstalk는 Flask 코드의 이름이 지정될 것으로 예상하기 때문에 배포 로그에서 이 오류가 발생합니다. application.py 다른 이름을 선택한 경우 다음 코드 샘플에 표시된 대로 eb config(을)를 실행하고 WSGIPath를 편집합니다.

```
aws:elasticbeanstalk:container:python:
  NumProcesses: '1'
  NumThreads: '15'
  StaticFiles: /static/=static/
  WSGIPath: application.py
```

application.py(을)를 파일 이름으로 바꿔야 합니다.

Gunicorn 및 Procfile을 적극 활용할 수도 있습니다. 이 접근 방식에 대한 자세한 내용은 Elastic Beanstalk 설명서에서 [Procfile을 사용하여 WSGI 서버 구성](#)을 참조하십시오.

오류 3

```
Target WSGI script '/opt/python/current/app/application.py' does not contain WSGI
application 'application'.
```

해결 방법: Elastic Beanstalk는 Flask 애플리케이션을 대표하는 변수의 이름이 지정될 것으로 예상합니다. application application.py 파일은 application(을)를 변수 이름으로 사용해야 합니다.

```
application = Flask(__name__)
```

오류 4

```
The EB CLI cannot find your SSH key file for keyname
```

해결 방법: EB CLI를 사용하여 어떤 키 페어를 사용할지 지정하거나 배포의 EC2 인스턴스를 위한 키 페어를 생성합니다. 오류를 해결하려면 eb init -i(을)를 실행합니다. 그러면 옵션 중 하나에 다음 메시지가 표시됩니다.

```
Do you want to set up SSH for your instances?
```

Y(으)로 응답하여 키 페어를 생성하거나 기존 키 페어를 지정합니다.

오류 5

코드를 업데이트하고 재배포했지만 배포에 변경 사항이 반영되지 않습니다.

해결 방법: 배포와 함께 Git 리포지토리를 사용하는 경우 재배포 전에 변경 내용을 추가하고 커밋해야 합니다.

오류 6

AWS Cloud9 IDE에서 Flask 애플리케이션을 미리보기 실행 중에 오류가 발생했습니다.

해결 방법: 이에 대한 자세한 내용은 AWS Cloud9 설명서에서 [AWS Cloud9 IDE의 실행 중인 애플리케이션 미리보기](#)를 참조하십시오.

Amazon Comprehend를 사용하는 자연어 처리

Amazon Comprehend를 사용을 선택하여 실시간 분석 또는 비동기 배치 작업을 실행함으로써 개별 텍스트 문서에서 사용자 지정 개체를 탐지할 수 있습니다. 또한 Amazon Comprehend를 사용하면 엔드 포인트를 생성하여 실시간으로 사용할 수 있는 사용자 지정 개체 인식 및 텍스트 분류 모델을 훈련할 수 있습니다.

이 패턴은 비동기 배치 작업을 사용하여 여러 문서가 포함된 입력 파일에서 감성과 개체를 탐지합니다. 이 패턴에서 제공하는 샘플 애플리케이션은 사용자가 행당 하나의 텍스트 문서와 함께 하나의 열이 들어 있는 .csv 파일을 업로드하도록 설계되어 있습니다. 플라스크와 comprehend_helper.py [AWS Elastic Beanstalk 리포지토리를 사용한 AI/ML 모델 결과 GitHub 시각화에](#) 있는 파일은 입력 파일을 읽고 처리를 위해 Amazon Comprehend로 입력을 전송합니다.

BatchDetectEntities

Amazon Comprehend는 이름이 지정된 개체가 있는지 문서 배치의 텍스트를 검사하여 탐지된 엔터티, 위치, [엔터티 유형](#), 그리고 Amazon Comprehend의 신뢰 수준을 표시하는 점수를 반환합니다. 한 번의 API 직접 호출로 최대 25개의 문서를 전송할 수 있으며, 각 문서의 크기는 5,000바이트 미만입니다. 사용 사례에 따라 특정 엔터티만 표시하도록 결과를 필터링할 수 있습니다. 예를 들어 'quantity' 개체 유형을 건너뛰고 탐지된 개체에 대한 임계값 점수(예: 0.75)를 설정할 수 있습니다. 임계값을 선택하기 전에 특정 사용 사례에 대한 결과를 살펴보는 것이 좋습니다. 이에 대한 자세한 내용은 Amazon [BatchDetectEntities](#) Comprehend 설명서를 참조하십시오.

BatchDetectSentiment

Amazon Comprehend는 수신 문서 배치를 검사하여 각 문서에 대한 일반적인 감성을 반환합니다 (POSITIVE, NEUTRAL, MIXED 또는 NEGATIVE). 한 번의 API 직접 호출로 최대 25개의 문서를 전송할 수 있으며, 각 문서의 크기는 5,000바이트 미만입니다. 감성에 대한 분석은 간단하며, 사용자가 가장 높은 점수의 감성을 선택하여 최종 결과에 표시합니다. 이에 대한 자세한 내용은 [Amazon BatchDetectSentimentComprehend](#) 설명서를 참조하십시오.

Flask 구성 처리

Flask 서버는 일련의 [구성 변수](#)를 사용하여 서버 실행 방법을 제어합니다. 이러한 변수에는 디버그 출력, 세션 토큰 또는 기타 애플리케이션 설정이 포함될 수 있습니다. 애플리케이션이 실행 중인 상태에서 액세스할 수 있는 사용자 지정 변수를 정의할 수도 있습니다. 구성 변수를 설정하는 방법은 여러 가지가 있습니다.

이 패턴에서는 구성이 `config.py`에 정의되고 `application.py` 내에서 상속됩니다.

- `config.py`에는 애플리케이션의 스타트업 시 설정되는 구성 변수가 포함됩니다. 이 애플리케이션에서는 서버를 [디버그 모드](#)에서 실행하도록 애플리케이션에 지시하는 `DEBUG` 변수가 정의됩니다. 참고: 프로덕션 환경에서 애플리케이션을 실행할 때는 디버그 모드를 사용하면 안 됩니다. `UPLOAD_FOLDER`(은)는 나중에 애플리케이션에서 참조하여 업로드된 사용자 데이터를 저장해야 하는 위치를 알려주도록 정의된 사용자 지정 변수입니다.
- `application.py`(은)는 Flask 애플리케이션을 시작하고 `config.py`에서 정의된 구성 설정을 상속합니다. 이는 다음 코드에 의해 수행됩니다.

```
application = Flask(__name__)
application.config.from_pyfile('config.py')
```

더 많은 패턴

- [SageMaker 노트북 인스턴스에 다른 AWS 계정의 CodeCommit 리포지토리에 대한 임시 액세스 권한 부여](#)
- [AWS 개발자 도구를 SageMaker 사용하여 ML, 빌드, 교육 및 워크로드를 Amazon으로 마이그레이션, 배포](#)
- [Amazon Redshift 기계 학습을 이용하여 고급 분석 수행](#)

메인프레임

주제

- [BMC AMI 클라우드 데이터를 사용하여 메인프레임 데이터를 Amazon S3에 백업하고 보관합니다.](#)
- [AWS 클라우드에서 고급 메인프레임 파일 뷰어 구축](#)
- [Blu Age로 현대화된 메인프레임 워크로드 컨테이너화](#)
- [Python을 사용하여 AWS에서 EBCDIC 데이터를 ASCII로 변환 및 압축 해제](#)
- [AWS Lambda를 사용하여 Amazon S3에서 메인프레임 파일을 EBCDIC 형식에서 문자로 구분된 ASCII 형식으로 변환합니다.](#)
- [Micro Focus를 사용하여 복잡한 레코드 레이아웃이 있는 메인프레임 데이터 파일 변환](#)
- [Terraform을 사용하여 컨테이너화된 Blu Age 애플리케이션을 위한 환경 배포](#)
- [Stonebranch 유니버설 컨트롤러를 AWS Mainframe Modernization과 통합](#)
- [Connect from Precisely을 사용하여 VSAM 파일을 Amazon RDS 또는 Amazon MSK로 마이그레이션하고 복제하기](#)
- [OpenText 마이크로 포커스 엔터프라이즈 서버 및 LRS X를 사용하여 AWS의 메인프레임 출력 관리를 현대화합니다. PageCenter](#)
- [Micro Focus Enterprise Server 및 LRS VPSX/MFI를 사용하여 AWS에서 메인프레임 배치 인쇄 워크로드를 현대화](#)
- [Micro Focus Enterprise Server 및 LRS VPSX/MFI를 사용하여 AWS에서 메인프레임 온라인 인쇄 워크로드를 현대화](#)
- [Transfer Family를 사용하여 메인프레임 파일을 Amazon S3로 직접 이동](#)
- [대규모 Db2 z/OS 데이터를 CSV 파일로 Amazon S3에 전송](#)
- [더 많은 패턴](#)

BMC AMI 클라우드 데이터를 사용하여 메인프레임 데이터를 Amazon S3에 백업하고 보관합니다.

작성자: Santosh Kumar Singh (AWS), Mikhael Liberman (Model9 Mainframe Software), Gilberto Biondo (AWS), Maggie Li (AWS)

환경: PoC 또는 파일럿	원본: 메인프레임	대상: Amazon S3
R 유형: 해당 없음	기술: 메인프레임, 스토리지 및 백업, 현대화	AWS 서비스: Amazon EC2, Amazon EFS, Amazon S3, AWS Direct Connect

요약

이 패턴은 메인프레임 데이터를 Amazon Simple Storage Service (Amazon S3) Simple Storage Service (Amazon S3) 에 직접 백업 및 보관한 다음 BMC AMI 클라우드 데이터 (이전에는 Model9 Manager라고 함) 를 사용하여 해당 데이터를 메인프레임으로 리콜하고 복원하는 방법을 보여줍니다. 메인프레임 현대화 프로젝트의 일환으로 백업 및 아카이브 솔루션을 현대화하거나 규정 준수 요구 사항을 충족하는 방법을 찾고 있다면 이 패턴이 이러한 목표를 달성하는 데 도움이 될 수 있습니다.

일반적으로 메인프레임에서 핵심 비즈니스 애플리케이션을 실행하는 조직은 VTL (Virtual Tape Library) 을 사용하여 파일 및 로그와 같은 데이터 저장소를 백업합니다. 이 방법은 요금이 청구되는 MIPS를 소비하고 메인프레임 외부의 테이프에 저장된 데이터에 액세스할 수 없기 때문에 비용이 많이 들 수 있습니다. 이러한 문제를 방지하기 위해 BMC AMI Cloud Data를 사용하여 운영 및 기간별 메인프레임 데이터를 Amazon S3로 빠르고 비용 효율적으로 직접 전송할 수 있습니다. BMC AMI Cloud Data를 사용하여 TCP/IP를 통해 데이터를 백업하고 보관하는 AWS 동시에 IBM z 통합 정보 프로세서 (zIIP) 엔진을 활용하여 비용, 병렬성 및 전송 시간을 줄일 수 있습니다.

사전 조건 및 제한 사항

사전 조건

- 활성 상태의 AWS 계정
- 유효한 라이선스 키가 있는 BMC AMI 클라우드 데이터
- 메인프레임과 AWS 간의 TCP/IP 연결
- S3 버킷에 대한 읽기/쓰기 액세스를 위한 AWS Identity and Access Management (IAM) 역할

- BMC AMI 클라우드 프로세스를 실행하기 위한 메인프레임 보안 제품 (RACF) 액세스 제공
- 사용 가능한 네트워크 포트, S3 버킷에 대한 액세스를 허용하는 방화벽 규칙 및 전용 z/fs 파일 시스템을 갖춘 BMC AMI Cloud z/OS 에이전트 (Java 버전 8 64비트 SR5 FP16 이상)
- BMC AMI 클라우드 관리 서버의 [요구 사항](#) 충족

제한 사항

- BMC AMI 클라우드 데이터는 관리 서버와 동일한 Amazon Elastic Compute Cloud (Amazon EC2) 인스턴스에서 도커 컨테이너로 실행되는 PostgreSQL 데이터베이스에 운영 데이터를 저장합니다. Amazon 관계형 데이터베이스 서비스 (Amazon RDS) 는 현재 BMC AMI 클라우드 데이터의 백업으로 지원되지 않습니다. [최신 제품 업데이트에 대한 자세한 내용은 새로운 소식을 참조하십시오.](#) BMC 설명서에서
- 이 패턴은 z/OS 메인프레임 데이터만 백업하고 보관합니다. BMC AMI Cloud Data는 메인프레임 파일만 백업하고 보관합니다.
- 이 패턴은 데이터를 JSON 또는 CSV와 같은 표준 공개 형식으로 변환하지 않습니다. [BMC AMI 클라우드 분석](#) (이전에는 Model9 Gravity로 알려짐) 과 같은 추가 변환 서비스를 사용하여 데이터를 표준 공개 형식으로 변환할 수 있습니다. 클라우드 네이티브 애플리케이션 및 데이터 분석 도구는 데이터가 클라우드에 기록된 후 데이터에 액세스할 수 있습니다.

제품 버전

- BMC AMI 클라우드 데이터 버전 2.x

아키텍처

소스 기술 스택

- z/OS를 실행하는 메인프레임
- 데이터세트 및 z/OS UNIX 시스템 서비스 (USS) 파일과 같은 메인프레임 파일
- 메인프레임 디스크(예: 직접 액세스 스토리지 디바이스) (DASD)
- 메인프레임 테이프(가상 또는 물리적 테이프 라이브러리)

대상 기술 스택

- Amazon S3

- Virtual Private Cloud(VPC)의 Amazon EC2 인스턴스
- AWS Direct Connect
- Amazon Elastic File System(Amazon EFS)

대상 아키텍처

다음 다이어그램은 메인프레임의 BMC AMI Cloud Data 소프트웨어 에이전트가 Amazon S3에 데이터를 저장하는 레거시 데이터 백업 및 아카이브 프로세스를 구동하는 참조 아키텍처를 보여줍니다.

이 다이어그램은 다음 워크플로를 보여줍니다.

1. BMC AMI Cloud Data 소프트웨어 에이전트는 메인프레임 논리적 파티션 (LPAR) 에서 실행됩니다. 소프트웨어 에이전트는 DASD 또는 테이프에서 TCP/IP를 통해 Amazon S3에 직접 메인프레임 데이터를 읽고 씁니다.
2. AWS Direct Connect 온프레미스 네트워크와 사이에 물리적이고 격리된 물리적 연결을 설정합니다. AWS 보안을 강화하려면 위에 site-to-site VPN을 AWS Direct Connect 실행하여 전송 데이터를 암호화하세요.
3. S3 버킷은 메인프레임 파일을 오브젝트 스토리지 데이터로 저장하고, BMC AMI Cloud Data 에이전트는 S3 버킷과 직접 통신합니다. 인증서는 에이전트와 Amazon S3 간의 모든 통신에 대한 HTTPS 암호화에 사용됩니다. Amazon S3 데이터 암호화는 저장 데이터를 암호화하고 보호하는 데 사용됩니다.
4. BMC AMI 클라우드 데이터 관리 서버는 EC2 인스턴스에서 Docker 컨테이너로 실행됩니다. 인스턴스는 메인프레임 LPAR 및 S3 버킷에서 실행되는 에이전트와 통신합니다.
5. Amazon EFS는 액티브 및 패시브 EC2 인스턴스 모두에 탑재되어 NFS(네트워크 파일 시스템) 스토리지를 공유합니다. 이는 장애 조치 시 관리 서버에 생성된 정책과 관련된 메타데이터가 손실되지 않도록 하기 위한 것입니다. 액티브 서버에서 페일오버를 수행하는 경우 데이터 손실 없이 패시브 서버에 액세스할 수 있습니다. 패시브 서버에 장애가 발생하는 경우 데이터 손실 없이 액티브 서버에 액세스할 수 있습니다.

도구

서비스

- [Amazon Elastic Compute Cloud \(Amazon EC2\)](#) 는 확장 가능한 컴퓨팅 용량을 제공합니다. AWS 클라우드 필요한 만큼 가상 서버를 시작하고 빠르게 스케일 업하거나 스케일 다운할 수 있습니다.

- [Amazon Elastic File System \(Amazon EFS\)](#) 을 사용하면 에서 공유 파일 시스템을 생성하고 구성할 수 AWS 클라우드 있습니다.
- [Amazon Simple Storage Service \(Amazon S3\)](#) 는 거의 모든 양의 데이터를 저장, 보호 및 검색하는데 도움이 되는 클라우드 기반 객체 스토리지 서비스입니다.
- [Amazon VPC \(Virtual Private Cloud\)](#) 를 사용하면 사용자가 정의한 가상 네트워크로 AWS 리소스를 시작할 수 있습니다. 이 가상 네트워크는 AWS의 확장 가능한 인프라를 사용한다는 이점과 함께 고객의 자체 데이터 센터에서 운영하는 기존 네트워크와 유사합니다.
- [AWS Direct Connect](#) 표준 이더넷 광섬유 케이블을 통해 내부 네트워크를 특정 AWS Direct Connect 위치에 연결합니다. 이 연결을 사용하면 네트워크 경로에서 인터넷 서비스 공급자를 우회하면서 공용 AWS 서비스에 직접 연결되는 가상 인터페이스를 만들 수 있습니다.
- [AWS Identity and Access Management \(IAM\)](#) 을 사용하면 리소스 인증 및 사용 권한을 부여받은 사용자를 제어하여 AWS 리소스에 대한 액세스를 안전하게 관리할 수 있습니다.

BMC 도구

- [BMC AMI 클라우드 관리 서버](#)는 아마존 EC2용 아마존 리눅스 아마존 머신 이미지 (AMI) 에서 도커 컨테이너로 실행되는 GUI 애플리케이션입니다. 관리 서버는 보고, 정책 생성 및 관리, 아카이브 실행, 백업, 리콜 및 복원 수행과 같은 BMC AMI Cloud 활동을 관리하는 기능을 제공합니다.
- [BMC AMI Cloud 에이전트](#)는 TCP/IP를 사용하여 오브젝트 스토리지에 직접 파일을 읽고 쓰는 온프레미스 메인프레임 LPAR에서 실행됩니다. 시작된 작업은 메인프레임 LPAR에서 실행되며 Amazon S3에서 백업 및 아카이브 데이터를 읽고 쓰는 역할을 담당합니다.
- [BMC AMI 클라우드 메인프레임 명령줄 인터페이스 \(M9CLI\)](#) 는 관리 서버에 종속되지 않고 TSO/E에서 직접 또는 일괄 작업으로 BMC AMI 클라우드 작업을 수행할 수 있는 명령 세트를 제공합니다.

에픽

S3 버킷 및 IAM 정책 생성

작업	설명	필요한 기술
S3 버킷을 생성합니다.	S3 버킷을 생성하여 백업하고 메인프레임 환경에서 아카이브하려는 파일과 볼륨을 저장합니다.	일반 AWS

작업	설명	필요한 기술
IAM 정책을 생성합니다.	<p>모든 BMC AMI Cloud 관리 서버 및 에이전트에는 이전 단계에서 생성한 S3 버킷에 대한 액세스 권한이 필요합니다.</p> <p>필요한 액세스 권한을 부여하려면 다음과 같은 IAM 정책을 생성하십시오.</p> <pre data-bbox="597 617 1027 1862"> { "Version": "2012-10-17", "Statement": [{ "Sid": "Listfolder", "Action": ["s3:ListBucket", "s3:GetBucketLocation", "s3:ListBucketVersions"], "Effect": "Allow", "Resource": ["arn:aws:s3:::<Bucket Name>"] }, { "Sid": "Objectaccess", "Effect": "Allow", </pre>	일반 AWS

작업	설명	필요한 기술
	<pre> "Action": ["s3:PutObject", "s3:GetObjectAcl", "s3:GetObject", "s3:DeleteObjectVersion", "s3:DeleteObject", "s3:PutObjectAcl", "s3:GetObjectVersion"], "Resource": ["arn:aws:s3:::<Bucket Name>/*"]] } } </pre>	

BMC AMI Cloud 소프트웨어 라이선스 받기 및 소프트웨어 다운로드

작업	설명	필요한 기술
<p>BMC AMI 클라우드 소프트웨어 라이선스를 받으세요.</p>	<p>소프트웨어 라이선스 키를 받으려면 BMC AMI Cloud 팀에 문의하십시오. 라이선스를 생성하려면 z/OS D M=CPU 명령의 출력이 필요합니다.</p>	<p>리드 구축</p>

작업	설명	필요한 기술
BMC AMI 클라우드 소프트웨어 및 라이선스 키를 다운로드합니다.	BMC 설명서의 지침에 따라 설치 파일 및 라이선스 키를 구하십시오.	메인프레임 인프라 관리자

메인프레임에 BMC AMI Cloud 소프트웨어 에이전트를 설치합니다.

작업	설명	필요한 기술
BMC AMI 클라우드 소프트웨어 에이전트를 설치합니다.	<ol style="list-style-type: none"> 설치 프로세스를 시작하기 전에 에이전트의 최소 소프트웨어 및 하드웨어 요구 사항이 충족되었는지 확인하십시오. 에이전트를 설치하려면 BMC 설명서의 지침을 따르십시오. 메인프레임 LPAR에서 에이전트가 실행되기 시작한 후 스킵에서 ZM91000I MODEL9 BACKUP AGENT INITIALIZED 메시지를 확인합니다. 에이전트의 STDOUT에서 Object store connectivity has been established successfully 메시지를 찾아 에이전트와 S3 버킷 간에 연결이 성공적으로 설정되었는지 확인합니다. 	메인프레임 인프라 관리자

EC2 인스턴스에 BMC AMI 클라우드 관리 서버를 설정합니다.

작업	설명	필요한 기술
<p>Amazon EC2 리눅스 2 인스턴스를 생성합니다.</p>	<p>Amazon EC2 설명서의 1단계: 인스턴스 시작의 지침에 따라 서로 다른 가용 영역에서 두 개의 Amazon EC2 Linux 2 인스턴스를 시작합니다.</p> <p>인스턴스는 다음과 같은 권장 하드웨어 및 소프트웨어 요구 사항을 충족해야 합니다.</p> <ul style="list-style-type: none"> • CPU - 최소 4코어 • RAM - 최소 8GB • 드라이브 - 40GB • 권장 EC2 인스턴스 — C5.xlarge • OS - Linux • 소프트웨어 — Docker, unzip, vi/VIM • 네트워크 대역폭 — 최소 1GB <p>자세한 내용은 BMC 설명서를 참조하십시오.</p>	<p>클라우드 아키텍트, 클라우드 관리자</p>
<p>Amazon EFS 파일 시스템을 생성합니다.</p>	<p>Amazon EFS 설명서의 1단계: Amazon EFS 파일 시스템 생성의 지침에 따라 Amazon EFS 파일 시스템을 생성합니다.</p> <p>파일 시스템을 생성할 때는 다음과 같이 하십시오.</p>	<p>클라우드 관리자, 클라우드 아키텍트</p>

작업	설명	필요한 기술
	<ul style="list-style-type: none">표준 스토리지 클래스를 선택합니다.EC2 인스턴스를 시작하는데 사용한 것과 동일한 VPC를 선택합니다.	

작업	설명	필요한 기술
<p>Docker를 설치하고 관리 서버를 구성합니다.</p>	<p>EC2 인스턴스에 연결:</p> <p>Amazon EC2 설명서의 Linux 인스턴스에 연결의 지침에 따라 EC2 인스턴스에 연결합니다.</p> <p>EC2 인스턴스 구성:</p> <p>각 EC2 인스턴스에 대해 다음을 수행합니다.</p> <ol style="list-style-type: none"> 1. Docker를 설치하려면 다음 명령을 실행합니다. <pre>sudo yum install docker</pre> 2. Docker를 시작하려면 다음 명령을 실행합니다. <pre>sudo service docker start</pre> 3. Docker의 상태를 확인하려면 다음 명령을 실행합니다. <pre>sudo service docker status</pre> 4. /etc/selinux 폴더에서 config 파일을 SELINUX=permissive로 변경합니다. 5. model9-v2.x.y_build_build-id-server.zip 및 Verificat 	<p>클라우드 아키텍트, 클라우드 관리자</p>

작업	설명	필요한 기술
	<p>ionScripts.zip 파일 (이전에 다운로드한 파일)을 EC2 인스턴스 중 하나의 임시 폴더 (예: 인스턴스의 /var/tmp 폴더)에 업로드합니다.</p> <p>6. tmp폴더로 이동하려면 다음 명령을 실행합니다.</p> <pre>cd/var/tmp</pre> <p>7. 확인 스크립트의 압축을 풀려면 다음 명령을 실행합니다.</p> <pre>unzip VerificationScripts.zip</pre> <p>8. 디렉토리를 변경하려면 다음 명령을 실행합니다.</p> <pre>cd /var/tmp/sysutils/PrereqsScripts</pre> <p>9. 확인 스크립트를 실행하려면 다음 명령을 실행합니다.</p> <pre>./M9VerifyPrereqs.sh</pre> <p>10. 확인 스크립트에서 입력을 요구하는 메시지가 표시되면 Amazon S3 URL 및 포트 번호를 입력합니다. 그런 다음 z/OS IP/DNS 및 포트 번호를 입력합니다.</p>	

작업	설명	필요한 기술
	<p>참고: 스크립트는 EC2 인스턴스가 메인프레임에서 실행되는 S3 버킷 및 에이전트에 연결할 수 있는지 확인하는 검사를 실행합니다. 연결이 설정되면 성공 메시지가 표시됩니다.</p>	

작업	설명	필요한 기술
<p>관리 서버 소프트웨어를 설치합니다.</p>	<ol style="list-style-type: none"> 1. 액티브 서버로 만들려는 EC2 인스턴스의 루트 디렉터리(예: /data/model9)에 폴더와 하위 폴더를 생성합니다. 2. amazon-efs-utils 패키지를 설치하고 이전에 생성한 Amazon EFS 파일 시스템을 마운트하려면 다음 명령을 실행합니다. <div data-bbox="634 743 1029 974" style="border: 1px solid #ccc; border-radius: 10px; padding: 10px; margin: 10px 0;"> <pre>sudo yum install -y amazon-efs-utils sudo mount -t efs -o tls <File System ID>:/ /data/model9</pre> </div> 3. Amazon EFS /etc/fstab 파일 시스템에 대한 항목으로 EC2 인스턴스의 파일을 업데이트하려면 (Amazon EC2 재부팅 시 Amazon EFS가 자동으로 다시 마운트되도록) 다음 명령을 실행합니다. <div data-bbox="634 1402 1029 1598" style="border: 1px solid #ccc; border-radius: 10px; padding: 10px; margin: 10px 0;"> <pre><Amazon-EFS-file-system-id>:/ /data/ model9 efs defaults, _netdev 0 0</pre> </div> 4. BMC AMI Cloud 설치 파일의 경로와 대상 설치 위치를 정의하려면 다음 명령을 실행하여 변수를 내보냅니다. 	<p>클라우드 아키텍트, 클라우드 관리자</p>

작업	설명	필요한 기술
	<pre data-bbox="634 212 1029 407">export MODEL9_HOME=/ data/model9 export M9INSTALL=/ var/tmp</pre> <p data-bbox="634 443 1016 573">참고: .bashrc 스크립트에 이러한 EXPORT 명령을 추가하는 것을 권장합니다.</p> <p data-bbox="591 600 1024 821">5. 디렉토리를 변경하려면 cd \$MODEL9_HOME 명령을 실행한 다음 mkdir diag 명령을 실행하여 다른 하위 디렉토리를 생성합니다.</p> <p data-bbox="591 848 1008 926">6. 설치 파일의 압축을 풀려면 다음 명령을 실행합니다.</p> <pre data-bbox="634 968 1029 1163">unzip \$M9INSTALL/ model9-<v2.x.y>_ build_<build-id>-s erver.zip</pre> <p data-bbox="634 1205 1016 1283">참고: x.y(버전) 및 build-id와 값을 바꾸십시오.</p> <p data-bbox="591 1310 1024 1388">7. 애플리케이션을 배포하려면 다음 명령을 실행하세요.</p> <pre data-bbox="634 1430 1029 1787">docker load -i \$MODEL9_HOME/model 9-<v2.x.y>_build_< build-id>.docker docker load -i \$MODEL9_HOME/postg res-12.10-x86.dock er.gz</pre>	

작업	설명	필요한 기술
	<p>참고: v2.x.y(버전) 및 build-id와 값을 바꾸십시오.</p> <p>8. \$MODEL9_HOME/conf 폴더에서 model9-local.yml 파일을 업데이트합니다.</p> <p>참고: 일부 매개변수에는 기본값이 있고 다른 매개변수는 필요에 따라 업데이트할 수 있습니다. 자세한 내용은 model9-local.yml 파일의 지침을 참조하세요.</p> <p>9. 라는 \$MODEL9_HOME/conf 파일을 만든 다음 파일에 다음 매개 변수를 추가합니다.</p> <pre data-bbox="630 1100 1027 1257">TZ=America/New_York EXTRA_JVM_ARGS=-Xmx2048m</pre> <p>10 Docker 네트워크 브리지를 만들려면 다음 명령을 실행합니다.</p> <pre data-bbox="630 1444 1027 1602">docker network create -d bridge model9network</pre> <p>11 BMC AMI 클라우드용 PostgreSQL 데이터베이스 컨테이너를 시작하려면 다음 명령을 실행합니다.</p>	

작업	설명	필요한 기술
	<pre data-bbox="634 212 1029 842">docker run -p 127.0.0.1:5432:5432 \ -v \$MODEL9_HOME/db/data:/var/lib/postgresql/data:z \ --name model9db -- restart unless-stopped \ --network model9network \ -e POSTGRES_PASSWORD=model9 -e POSTGRES_DB=model9 -d postgres:12.10</pre> <p data-bbox="594 856 1029 1037">12 PostgreSQL 컨테이너가 실행되기 시작한 후, 다음 명령을 실행하여 애플리케이션 서버를 시작합니다.</p> <pre data-bbox="634 1073 1029 1850">docker run -d -p 0.0.0.0:443:443 -p 0.0.0.0:80:80 \ --sysctl net.ipv4.tcp_keepalive_time=600 \ --sysctl net.ipv4.tcp_keepalive_intvl=30 \ --sysctl net.ipv4.tcp_keepalive_probes=10 \ -v \$MODEL9_HOME:/model9:z -h \$(hostname) --restart unless-stopped \ --env-file \$MODEL9_HOME/conf/model9.env \</pre>	

작업	설명	필요한 기술
	<pre data-bbox="634 212 980 415">--network model9network \ --name model9-v2.x.y model9:<v2.x.y>.<build-id></pre> <p data-bbox="630 464 1024 590">참고: v2.x.y(버전) 및 build-id와 값을 바꾸십시오.</p> <p data-bbox="594 617 1008 695">13.두 컨테이너의 상태를 확인하려면 명령을 실행합니다.</p> <pre data-bbox="634 737 846 800">docker ps -a</pre> <p data-bbox="594 831 1024 957">14.패시브 EC2 인스턴스에 관리 서버를 설치하려면 1~4, 7, 10~13단계를 반복합니다.</p> <p data-bbox="594 1037 1024 1262">참고: 문제를 해결하려면 / data/model9/logs/ 폴더에 저장된 로그로 이동하십시오. 자세한 내용은 BMC 설명서를 참조하십시오.</p>	

에이전트를 추가하고 BMC AMI Cloud 관리 서버에 백업 또는 아카이브 정책을 정의합니다.

작업	설명	필요한 기술
<p>새로운 에이전트를 추가합니다.</p>	<p>새 에이전트를 추가하기 전에 다음을 확인하십시오.</p> <ul style="list-style-type: none"> BMC AMI Cloud 에이전트는 메인프레임 LPAR에서 실행 중이며 완전히 초기화되었습니다. 스펴에서 ZM91000I 	<p>메인프레임 스토리지 관리자 또는 개발자</p>

작업	설명	필요한 기술
	<p>MODEL9 BACKUP AGENT INITIALIZED 초기화 메시지를 찾아 에이전트를 식별하십시오.</p> <ul style="list-style-type: none"> 관리 서버의 Docker 컨테이너가 완전히 초기화되어 실행 중입니다. <p>백업 및 아카이브 정책을 정의하기 전에 관리 서버에 에이전트를 생성해야 합니다. 에이전트를 생성하려면 다음을 수행합니다.</p> <ol style="list-style-type: none"> 웹 브라우저를 사용하여 Amazon EC2 시스템에 배포된 관리 서버에 액세스한 다음 메인프레임 자격 증명으로 로그인합니다. 에이전트 탭을 선택한 후 에이전트 추가를 선택합니다. 이름에 에이전트 이름을 입력합니다. 호스트 이름/IP 주소에는 메인프레임의 호스트 이름 또는 IP 주소를 입력합니다. 포트에 포트 번호를 입력합니다. 연결 테스트를 선택합니다. 연결이 성공적으로 설정되면 성공 메시지를 볼 수 있습니다. 	

작업	설명	필요한 기술
	<p>7. 생성(CREATE)을 선택합니다.</p> <p>에이전트가 생성되면 표에 나타나는 새 창에서 오브젝트 스토리지 및 메인프레임 에이전트에 대한 연결 상태를 확인할 수 있습니다.</p>	
<p>백업 또는 아카이브 정책을 생성합니다.</p>	<ol style="list-style-type: none"> 1. 정책을 선택합니다. 2. 정책 생성을 선택합니다. 3. 새 정책 생성 페이지에서 정책 사양을 입력합니다. <p>참고: 사용 가능한 사양에 대한 자세한 내용은 BMC 설명서의 새 정책 생성을 참조하십시오.</p> <ol style="list-style-type: none"> 4. 마침을 클릭합니다. 5. 이제 새 정책이 표로 나열됩니다. 이 표를 보려면 정책 탭을 선택합니다. 	<p>메인프레임 스토리지 관리자 또는 개발자</p>

관리 서버에서 백업 또는 아카이브 정책을 실행합니다.

작업	설명	필요한 기술
<p>백업 또는 아카이브 정책을 실행합니다.</p>	<p>이전에 관리 서버에서 생성한 데이터 백업 또는 아카이브 정책을 수동 또는 자동으로 (일정에 따라) 실행합니다. 정책을 수동으로 실행하려면:</p>	<p>메인프레임 스토리지 관리자 또는 개발자</p>

작업	설명	필요한 기술
	<ol style="list-style-type: none"> 1. 탐색 메뉴에서 정책 탭을 선택합니다. 2. 실행하려는 정책의 표 오른쪽에서 점 3개로 된 메뉴를 선택합니다. 3. 지금 실행을 선택합니다. 4. 팝업 확인 창에서 예, 지금 정책 실행을 선택합니다. 5. 정책이 실행된 후 정책 활동 섹션에서 실행 상태를 확인합니다. 6. 실행한 정책에 대해 점 3개 메뉴를 선택한 다음 [실행 로그 보기] 를 선택하여 로그를 확인합니다. 7. 백업이 생성되었는지 확인하려면 S3 버킷을 확인합니다. 	

작업	설명	필요한 기술
백업 또는 아카이브 정책을 복원합니다.	<ol style="list-style-type: none"> 1. 탐색 메뉴에서 정책 탭을 선택합니다. 2. 복원 프로세스를 실행할 정책을 선택합니다. 여기에는 특정 정책에 대해 과거에 실행된 모든 백업 또는 아카이브 활동이 나열됩니다. 3. 복원할 백업을 선택하려면 날짜-시간 열을 선택합니다. file/Volume/Storage 그룹 이름에는 정책의 실행 세부 정보가 표시됩니다. 4. 표 오른쪽에서 점 3개로 된 메뉴를 선택한 다음 RESTORE를 선택합니다. 5. 팝업 창에서 대상 이름, 볼륨, 스토리지 그룹을 입력한 다음 복원을 선택합니다. 6. 메인프레임 보안 인증 정보를 입력한 다음, 복원을 다시 선택합니다. 7. 복원이 성공했는지 확인하려면 로그 또는 메인프레임을 확인합니다. 	메인프레임 스토리지 관리자 또는 개발자

메인프레임에서 백업 또는 아카이브 정책을 실행합니다.

작업	설명	필요한 기술
M9CLI 를 사용하여 백업 또는 아카이브 정책을 실행합니다.	BMC AMI 클라우드 관리 서버에서 규칙을 설정하지 않고도 M9CLI 를 사용하여 TSO/E, REXX 또는 JCL을 통해 백업	메인프레임 스토리지 관리자 또는 개발자

작업	설명	필요한 기술
	<p>및 복원 프로세스를 수행할 수 있습니다.</p> <p>TSO/E 사용:</p> <p>TSO/E를 사용하는 경우 TSO/E가 연결되었는지 확인하십시오. M9CLI REXX TSO TSO/E를 통해 데이터세트를 백업하려면 명령을 사용합니다. TSO M9CLI BACKDSN <DSNAME></p> <p>참고: M9CLI 명령에 대한 자세한 내용은 BMC 설명서의 CLI 참조를 참조하십시오.</p> <p>JCL 사용:</p> <p>JCL을 사용하여 백업 및 아카이브 정책을 실행하려면 M9CLI 명령을 실행합니다.</p> <p>배치 작업 사용:</p> <p>다음 예제는 M9CLI 명령어를 일괄적으로 실행하여 데이터세트를 보관하는 방법을 보여줍니다.</p> <pre data-bbox="592 1459 1031 1837"> //JOBNAME JOB ... //M9CLI EXEC PGM=IKJEF T01 //STEPLIB DD DISP=SHR, DSN=<MODEL9 LOADLIB> //SYSEXEC DD DISP=SHR, DSN=<MODEL9 EXEC LIB> //SYSTSPRT DD SYSOUT=* //SYSPRINT DD SYSOUT=* </pre>	

작업	설명	필요한 기술
	<pre>//SYSTSIN DD TSO M9CLI ARCHIVE M9CLI ARCHIVE <DSNNAME OR DSN PATTERN> /</pre>	
<p>JCL 배치에서 백업 또는 아카이브 정책을 실행합니다.</p>	<p>BMC AMI 클라우드는 M9SAPIJ 이라는 샘플 JCL 루틴을 제공합니다. JCL을 사용하여 관리 서버에 생성된 특정 정책을 실행하도록 M9SAPIJ 사용자 지정할 수 있습니다. 이 작업은 백업 및 복원 프로세스를 자동으로 실행하기 위한 일괄 작업 스케줄러의 일부일 수도 있습니다.</p> <p>작업에는 다음과 같은 필수 값이 필요합니다.</p> <ul style="list-style-type: none"> • 관리 서버 IP 주소/호스트 이름 • 포트 번호 • 정책 ID 또는 정책 이름(관리 서버에 생성됨) <p>참고: 샘플 작업의 지침에 따라 다른 값을 변경할 수도 있습니다.</p>	<p>메인프레임 스토리지 관리자 또는 개발자</p>

관련 리소스

- [AWS를 사용한 메인프레임 현대화](#) (AWS 설명서)

- [Model9와 AWS를 사용하여 메인프레임용 클라우드 백업으로 비용을 절감하는 방법](#) (AWS 파트너 네트워크 블로그)
- [Model9를 사용하여 AWS에서 메인프레임 데이터 분석을 활성화하는 방법](#) (AWS 파트너 네트워크 블로그)
- [AWS Direct Connect Resiliency 권장 사항](#) (AWS 설명서)
- [BMC AMI 클라우드 설명서](#) (BMC 웹 사이트)

AWS 클라우드에서 고급 메인프레임 파일 뷰어 구축

제작자: Boopathy GOPALSAMY(AWS)와 Jeremiah O'Connor(AWS)

환경: PoC 또는 파일럿

기술: 메인프레임, 마이그레이션, 서버리스

워크로드: IBM

AWS 서비스: 아마존 아테나, AWS Lambda, 아마존 서비스, AWS OpenSearch Step Functions

요약

AWS 서버리스 서비스를 사용함으로써 메인프레임 파일을 검색하고 검토하기 위한 고급 도구를 구축하기 위해 이 패턴의 단계와 코드 샘플을 이용하세요. 패턴은 탐색 및 검색을 위해 메인프레임 입력 파일을 Amazon OpenSearch Service 문서로 변환하는 방법의 예를 제공합니다. 파일 뷰어 도구를 사용하면 다음과 같은 결과를 얻을 수 있습니다.

- AWS 대상 마이그레이션 환경의 일관성을 위해 동일한 메인프레임 파일 구조 및 레이아웃 유지(예: 파일을 외부로 전송하는 배치 애플리케이션에서 파일에 대해 동일한 레이아웃을 유지할 수 있음)
- 메인프레임 마이그레이션 중에 개발 및 테스트 속도 향상
- 마이그레이션 후 유지 관리 활동 지원

사전 조건 및 제한 사항

사전 조건

- 활성 상태의 AWS 계정
- 기존 플랫폼에서 연결할 수 있는 서브넷이 있는 Virtual Private Cloud (VPC)
- 입력 파일 및 해당 COBOL 공용 비즈니스 지향 언어 (COBOL) 카피북 (참고: 입력 파일 및 COBOL 카피북 예제는 리포지토리를 참조하십시오. [gfs-mainframe-solutions](#) GitHub COBOL 카피북에 대한 자세한 내용은 IBM 웹 사이트의 z/OS용 [엔터프라이즈 COBOL](#) 6.3 프로그래밍 가이드를 참조하십시오.)

제한 사항

- 카피북 구문 분석은 최대 두 개의 중첩 수준으로 제한(OCCURS)

아키텍처

소스 기술 스택

- [FB\(고정 차단\)](#) 형식의 입력 파일
- COBOL 카피북 레이아웃

대상 기술 스택

- Amazon Athena
- 아마존 OpenSearch 서비스
- Amazon Simple Storage Service(S3)
- AWS Lambda
- AWS Step Functions

대상 아키텍처

다음 다이어그램은 메인프레임 입력 파일을 구문 분석하여 탐색 및 검색을 위한 OpenSearch 서비스 문서로 변환하는 프로세스를 보여줍니다.

이 다이어그램은 다음 워크플로를 보여줍니다.

1. 관리자 또는 애플리케이션이 입력 파일을 S3 버킷 하나에 푸시하고 COBOL 카피북을 다른 S3 버킷으로 푸시합니다.
2. 입력 파일이 있는 S3 버킷은 서버리스 Step Functions 워크플로를 시작하는 Lambda 함수를 간접적으로 호출합니다. 참고: 이 패턴에서 S3 이벤트 트리거와 Lambda 함수를 사용하여 Step Functions 워크플로를 구동하는 것은 선택 사항입니다. 이 패턴의 GitHub 코드 샘플에는 이러한 서비스의 사용이 포함되지 않지만 요구 사항에 따라 이러한 서비스를 사용할 수 있습니다.
3. Step Functions 워크플로는 다음 Lambda 함수의 모든 배치 프로세스를 조정합니다.
 - `s3copybookparser.py` 함수는 카피북 레이아웃을 구문 분석하고 필드 속성, 데이터 유형 및 오피셋(입력 데이터 처리에 필요)을 추출합니다.

- 이 `s3toathena.py` 함수는 Athena 테이블 레이아웃을 생성합니다. Athena는 `s3toathena.py` 함수로 처리된 입력 데이터를 구문 분석하고 데이터를 CSV 파일로 변환합니다.
 - `s3toelasticsearch.py` 함수는 S3 버킷에서 결과 파일을 수집하여 서비스로 푸시합니다. OpenSearch
4. 사용자는 OpenSearch Service와 함께 OpenSearch 대시보드에 액세스하여 다양한 테이블 및 열 형식의 데이터를 검색한 다음 인덱싱된 데이터에 대해 쿼리를 실행합니다.

도구

서비스

- [Amazon Athena](#)는 표준 SQL을 사용하여 Amazon Simple Storage Service(Amazon S3)에 있는 데이터를 직접 간편하게 분석할 수 있는 대화형 쿼리 서비스입니다.
- [AWS Lambda](#)는 서버를 프로비저닝하거나 관리할 필요 없이 코드를 실행하는 데 도움이 되는 컴퓨팅 서비스입니다. 필요할 때만 코드를 실행하며 자동으로 확장이 가능하므로 사용한 컴퓨팅 시간만큼만 비용을 지불합니다. 이 패턴에서는 Lambda를 사용하여 파일 파싱, 데이터 변환, 대화형 파일 액세스를 위한 Service로 데이터 OpenSearch 로드와 같은 핵심 로직을 구현합니다.
- [Amazon OpenSearch Service](#)는 AWS 클라우드에서 서비스 클러스터를 배포, 운영 및 확장할 수 있도록 지원하는 관리형 OpenSearch 서비스입니다. 이 패턴에서는 OpenSearch 서비스를 사용하여 변환된 파일을 인덱싱하고 사용자에게 대화형 검색 기능을 제공합니다.
- [Amazon Simple Storage Service\(S3\)](#)는 원하는 양의 데이터를 저장, 보호 및 검색하는 데 도움이 되는 클라우드 기반 객체 스토리지 서비스입니다.
- [AWS Command Line Interface\(AWS CLI\)](#)는 명령줄 쉘에서 명령을 사용하여 AWS 서비스와 상호 작용할 수 있는 오픈 소스 도구입니다.
- [AWS Identity and Access Management\(IAM\)](#)를 사용하면 사용자에 대해 인증 및 권한 부여를 제어함으로써 AWS 리소스에 대한 액세스를 안전하게 관리할 수 있습니다.
- [AWS Step Functions](#)는 Lambda 함수와 기타 서비스를 결합할 수 있는 서버리스 오케스트레이션 서비스로, 비즈니스 크리티컬 애플리케이션을 구축합니다. 이 패턴에서는 Step Functions를 사용하여 Lambda 함수를 오케스트레이션합니다.

기타 도구

- [GitHub](#) 공동 작업 도구 및 버전 제어를 제공하는 코드 호스팅 서비스입니다.
- [Python](#)은 고급 프로그래밍 언어입니다.

코드

이 패턴의 코드는 저장소에서 사용할 수 있습니다. GitHub [gfs-mainframe-patterns](#)

에픽

대상 환경 준비

작업	설명	필요한 기술
S3 버킷을 생성합니다.	<p>카피북, 입력 파일 및 출력 파일을 저장하기 위한 S3 버킷을 생성합니다. S3 버킷에는 다음과 같은 폴더 구조를 사용하는 것이 좋습니다.</p> <ul style="list-style-type: none"> • copybook/ • input/ • output/ • query/ • results/ 	일반 AWS
s3copybookparser 함수를 생성합니다.	<ol style="list-style-type: none"> 1. s3copybookparser 라는 Lambda 함수를 생성하고 리포지토리에서 소스 코드 s3copybookparser.py (copybook.py 및) 를 업로드합니다. GitHub 2. S3ReadOnly Lambda 함수에 IAM 정책을 연결합니다. 	일반 AWS
s3toathena 함수를 생성합니다.	<ol style="list-style-type: none"> 1. s3toathena 라는 Lambda 함수를 생성하고 리포지토리에서 소스 코드 s3toathena.py () 를 업로드합니다. GitHub Lambda 타임아웃을 60초 이상으로 구성합니다. 	일반 AWS

작업	설명	필요한 기술
	<p>2. 필요한 리소스에 대한 액세스를 제공하려면 IAM 정책 AmazonAthenaFullAccess 과 S3FullAccess 를 Lambda 함수에 연결합니다.</p>	
<p>s3toelasticsearch 함수를 생성합니다.</p>	<ol style="list-style-type: none"> 1. Lambda 환경에 Python 종속성을 추가합니다. 중요: Lambda 함수는 Python Elasticsearch 클라이언트 종속성(Elasticsearch==7.9.0 및 requests_aws4auth) 을 사용하기 때문에 s3toelasticsearch 함수를 사용하려면 Python 종속성을 추가해야 합니다. 2. s3toelasticsearch 라는 Lambda 함수를 생성하고 리포지토리에서 소스 코드 s3toelasticsearch.py () 를 업로드합니다. GitHub 3. Python 종속성을 Lambda 계층으로서 가져옵니다. 4. Lambda 함수에 IAM 정책S3ReadOnly 과 AmazonOpenSearchServiceReadOnlyAccess 를 연결합니다. 	<p>일반 AWS</p>

작업	설명	필요한 기술
<p>OpenSearch 서비스 클러스터를 생성합니다.</p>	<p>클러스터를 생성합니다</p> <ol style="list-style-type: none"> 1. OpenSearch 서비스 클러스터를 생성합니다. 클러스터를 생성할 때 다음 작업을 수행합니다. <ul style="list-style-type: none"> • OpenSearch 대시보드에 로그인하는 데 사용할 수 있는 클러스터의 마스터 사용자 및 암호를 생성합니다. 참고: Amazon Cognito를 통한 인증을 사용하는 경우에는 이 단계가 필요하지 않습니다. • 세분화된 액세스 제어를 선택합니다. 이렇게 하면 OpenSearch Service에서 데이터에 대한 액세스를 제어할 수 있는 추가 방법이 제공됩니다. 2. 도메인 URL을 복사하여 Lambda 함수 <code>s3toelasticsearch</code> 에 환경 변수 '호스트'로서 전달합니다. <p>IAM 역할에 대한 액세스 권한 부여</p> <p>Lambda 함수의 IAM 역할 (<code>arn:aws:iam::*:role/service-role/s3toelasticsearch-role-*</code>)에 대한 세분화된 액</p>	<p>일반 AWS</p>

작업	설명	필요한 기술
	<p>세스를 제공하려면 다음을 수행하세요.</p> <ol style="list-style-type: none"> 1. OpenSearch 대시보드에 마스터 사용자로 로그인합니다. 2. 보안 탭을 선택한 다음 역할, all_access, 맵 사용자, 백엔드 역할을 선택합니다. 3. Lambda 함수의 IAM 역할의 Amazon 리소스 이름(ARN)을 추가한 다음 저장을 선택합니다. 자세한 내용은 OpenSearch 서비스 설명서의 사용자에게 역할 매핑을 참조하십시오. 	
오케스트레이션을 위한 Step Functions를 생성합니다.	<ol style="list-style-type: none"> 1. 표준 플로우를 사용하여 Step Functions 상태 머신을 생성합니다. 정의는 GitHub 리포지토리에 포함되어 있습니다. 2. JSON 스크립트에서 Lambda 함수의 ARN을 사용자 환경에 있는 Lambda 함수의 ARN으로 교체합니다. 	일반 AWS

배포 및 실행

작업	설명	필요한 기술
입력 파일 및 카피북을 S3 버킷에 업로드합니다.	GitHub 리포지토리 샘플 폴더에서 샘플 파일을 다운로드하	일반 AWS

작업	설명	필요한 기술
	<p>고 이전에 생성한 S3 버킷에 파일을 업로드합니다.</p> <ol style="list-style-type: none"><li data-bbox="592 338 1015 520">1. Mockedcopy.cpy 및 acctix.cpy 를 <S3_Bucket>/copybook 폴더에 업로드합니다.<li data-bbox="592 541 1015 766">2. Modedupdate.txt 및 acctindex.cpy 샘플 입력 파일을 <S3_Bucket>/input 폴더에 업로드합니다.	

작업	설명	필요한 기술
Step 함수를 호출합니다.	<ol style="list-style-type: none"> 1. AWS Management Console에 로그인하고 Step Functions 콘솔을 엽니다. 2. 탐색 창에서 상태 머신을 선택합니다. 3. 상태 머신을 선택한 다음 실행 시작을 선택합니다. 4. 입력 상자에 다음 카피북/파일 경로를 S3 버킷의 JSON 변수로 입력한 다음 실행 시작을 선택합니다. <pre data-bbox="597 842 1024 1360"> { "s3_copybook_bucket_name": "<BUCKET NAME>", "s3_copybook_bucket_key": "<COPYBOOK PATH>", "s3_source_bucket_name": "<BUCKET NAME", "s3_source_bucket_key": "INPUT FILE PATH" } </pre> <p>예:</p> <pre data-bbox="597 1472 1024 1839"> { "s3_copybook_bucket_name": "fileaidtest", "s3_copybook_bucket_key": "copybook/acctix.cpy", "s3_source_bucket_name": "fileaidtest", </pre>	일반 AWS

작업	설명	필요한 기술
<p>Step Functions에서 워크플로 실행을 검증합니다.</p>	<pre data-bbox="597 203 1023 388">"s3_source_bucket_key": "input/accountindex" }</pre> <p>Step Functions 콘솔에서 그래프 인스펙터의 워크플로 실행을 검토합니다. 실행 상태는 색으로 구분되어 현재 실행 상태를 나타냅니다. 예를 들어, 파란색은 진행 중, 녹색은 성공, 빨간색은 실패를 나타냅니다. 실행 이벤트에 대한 자세한 내용은 실행 이벤트 기록 섹션의 표를 검토할 수도 있습니다.</p> <p>그래픽 워크플로우 실행의 예는 이 패턴의 추가 정보 섹션에 있는 Step Functions 그래프를 참조하세요.</p>	<p>일반 AWS</p>
<p>Amazon에서 배송 로그를 확인합니다 CloudWatch.</p>	<ol data-bbox="597 1136 1023 1596" style="list-style-type: none"> 1. AWS Management Console에 로그인하고 CloudWatch 콘솔을 엽니다. 2. 왼쪽 탐색 창에서 로그를 선택한 다음, 로그 그룹을 선택합니다. 3. 검색 상자에서 s3toelasticsearch 함수의 로그 그룹을 검색합니다. <p>성공적인 배송 로그의 예는 이 패턴의 추가 정보 섹션에 있는 CloudWatch 배송 로그를 참조하십시오.</p>	<p>일반 AWS</p>

작업	설명	필요한 기술
<p>OpenSearch 대시보드에서 형식이 지정된 파일의 유효성을 검사하고 파일 작업을 수행합니다.</p>	<ol style="list-style-type: none"> 1. AWS Management Console에 로그인합니다. 애널리틱스에서 Amazon OpenSearch 서비스를 선택합니다. 2. 탐색 창에서 도메인을 선택합니다. 3. 검색 상자에 OpenSearch 대시보드의 도메인 URL을 입력합니다. 4. 대시보드를 선택한 다음, 마스터 사용자로 로그인합니다. 5. 색인된 데이터를 표 형식으로 찾아보십시오. 6. 입력 파일을 대시보드의 형식이 지정된 출력 파일 (색인된 문서) 과 비교합니다. OpenSearch 대시보드 뷰에는 서식이 지정된 파일에 추가된 열 헤더가 표시됩니다. 형식이 지정되지 않은 입력 파일의 소스 데이터가 대시보드 보기의 대상 데이터와 일치하는지 확인합니다. 7. 인덱싱된 파일에 대해 검색(예: 필드 이름, 값 또는 표현식 사용), 필터 및 DQL(Dashboard Query Language) 작업과 같은 작업을 수행합니다. 	<p>일반 AWS</p>

관련 리소스

참조

- [예시 COBOL 카피북 \(IBM 설명서\)](#)
- [BMC 컴퓨웨어 파일 지원 \(BMC 설명서\)](#)

자습서

- [Tutorial: Amazon S3 트리거를 사용하여 Lambda 함수 호출 \(AWS Lambda 설명서\)](#)
- [AWS Step Functions와 AWS Lambda를 사용하여 서버리스 워크플로를 생성하려면 어떻게 해야 하나요? \(AWS 설명서\)](#)
- [Amazon OpenSearch 서비스에서 OpenSearch 대시보드 사용 \(AWS 설명서\)](#)

추가 정보

Step Functions 그래프

다음 예제에서는 Step Functions 그래프를 보여줍니다. 이 그래프는 이 패턴에 사용된 Lambda 함수의 실행 상태를 보여줍니다.

CloudWatch 전송 로그

다음 예제는 s3toelasticsearch 실행의 실행에 대한 성공적인 전송 로그를 보여줍니다.

```
2022-08-10T15:53:33.033-05:   처리 문서 수: 100개
00
```

```
2022-08-10T15:53:33.171-05: [정보] 2022-08-10T20:53:3
3.171Z a1b2c3d4-5678-90ab
-cdef-EXAMPLE11111
POST https://search-ess
earch-3h4uqclifeqaj2vg4mphe
7ffle.us-east-2.es.amazonaw
s.com:443/_bulk [상태: 200 요
청: 0.100초]
```

2022-08-10T15:53:33.172-05: 대량 쓰기 성공: 100개의 문서
00

Blu Age로 현대화된 메인프레임 워크로드 컨테이너화

작성자: Richard Milner-Watts(AWS)

코드 리포지토리: 블루에이지 애플리케이션 컨테이너 예제	환경: 프로덕션	소스: 메인프레임 워크로드
대상: 컨테이너	R 유형: 리아키텍트	워크로드: IBM, 기타 모든 워크로드
기술: 메인프레임, 컨테이너 및 마이크로서비스, 마이그레이션, 현대화	AWS 서비스: Amazon ECS, Amazon ECR	

요약

이 패턴은 [Blu Age](#) 도구를 사용하여 현대화된 메인프레임 워크로드를 실행하기 위한 샘플 컨테이너 환경을 제공합니다. Blu Age는 레거시 메인프레임 워크로드를 최신 Java 코드로 변환합니다. 이 패턴은 Java 애플리케이션에 대한 래퍼를 제공하므로 [Amazon Elastic Container Service\(Amazon ECS\)](#) 또는 [Amazon Elastic Kubernetes Service\(Amazon EKS\)](#)와 같은 컨테이너 오케스트레이션 서비스를 사용하여 Java 애플리케이션을 실행할 수 있습니다.

Blu Age 및 AWS 서비스를 사용하여 워크로드를 현대화하는 방법에 대한 자세한 내용은 다음 AWS 권장 가이드 간행물을 참조하십시오.

- [서버리스 AWS 인프라에서 현대화된 Blu Age 메인프레임 워크로드 실행](#)
- [Terraform을 사용하여 컨테이너화된 Blu Age 애플리케이션을 위한 환경 배포](#)

Blu Age를 사용하여 메인프레임 워크로드를 현대화하는 데 도움이 필요하면 [Blu Age 웹사이트](#)에서 전문가에게 문의하기를 선택하여 Blu Age 팀에 문의하십시오. 현대화된 워크로드를 AWS로 마이그레이션하고, 이를 AWS 서비스와 통합하고, 프로덕션 환경으로 이전하는 데 도움이 필요하면 AWS 계정 관리자에게 문의하거나 [AWS Professional Services 양식](#)을 작성하십시오.

사전 조건 및 제한 사항

사전 조건

- Blu Age에서 만든 현대화된 Java 애플리케이션입니다. 테스트 목적으로 이 패턴은 개념 증명으로 사용할 수 있는 샘플 Java 애플리케이션을 제공합니다.
- 컨테이너를 빌드하는 데 사용할 수 있는 [Docker](#) 환경입니다.

제한 사항

사용하는 컨테이너 오케스트레이션 플랫폼에 따라 컨테이너에서 사용할 수 있는 리소스(예: CPU, RAM, 스토리지)가 제한될 수 있습니다. 예를 들어, AWS Fargate와 함께 Amazon ECS를 사용하는 경우, [Amazon ECS 설명서](#)에서 제한 및 고려 사항을 참조하십시오.

아키텍처

소스 기술 스택

- Blu Age
- Java

대상 기술 스택

- Docker

대상 아키텍처

다음 다이어그램에서는 Docker 컨테이너 내 Blu Age 애플리케이션의 아키텍처를 보여줍니다.

1. 컨테이너의 진입점은 래퍼 스크립트입니다. 이 bash 스크립트는 Blu Age 애플리케이션의 런타임 환경을 준비하고 출력을 처리하는 역할을 합니다.
2. 컨테이너 내의 환경 변수는 Amazon Simple Storage Service(S3) 버킷 이름 및 데이터베이스 보안 인증과 같은 래퍼 스크립트의 변수를 구성하는 데 사용됩니다. 환경 변수는 AWS Secrets Manager 또는 AWS Systems Manager의 기능인 파라미터 스토어에서 제공합니다. Amazon ECS를 컨테이너 오케스트레이션 서비스로 사용하는 경우 Amazon ECS 작업 정의에서 환경 변수를 하드코딩할 수도 있습니다.
3. 래퍼 스크립트는 Blu Age 애플리케이션을 실행하기 전에 S3 버킷의 모든 입력 파일을 컨테이너로 가져오는 역할을 합니다. AWS Command Line Interface(AWS CLI)는 컨테이너 내에 설치됩니다. 이 메커니즘은 Virtual Private Cloud(VPC) 엔드포인트를 통한 Amazon S3 액세스용 메커니즘을 제공합니다.

4. Blu Age 애플리케이션용 Java Archive(JAR) 파일은 Amazon Aurora와 같은 다른 데이터 소스와 통신해야 할 수 있습니다.
5. 완료 후 래퍼 스크립트는 추가 처리 (예: Amazon CloudWatch 로깅 서비스) 를 위해 결과 출력 파일을 S3 버킷으로 전달합니다. 표준 CloudWatch 로깅 대신 사용하는 경우 이 패턴은 압축된 로그 파일을 Amazon S3로 전송하는 것도 지원합니다.

도구

서비스

- [Amazon Elastic Container Registry\(Amazon ECR\)](#)는 안전하고 확장 가능하고 신뢰할 수 있는 관리형 컨테이너 이미지 레지스트리 서비스입니다.
- [Amazon Elastic Container Service\(Amazon ECS\)](#)는 클러스터에서 컨테이너를 실행, 중지 및 관리하는 데 도움이 되는 빠르고 확장 가능한 컨테이너 관리 서비스입니다.

도구

- [Docker](#)는 애플리케이션을 구축, 테스트 및 배포하기 위한 소프트웨어 플랫폼입니다. Docker는 소프트웨어를 컨테이너라는 표준화된 단위로 패키징합니다. [컨테이너](#)에는 라이브러리, 시스템 도구, 코드, 런타임을 포함하여 소프트웨어 실행에 필요한 모든 것이 들어 있습니다. Docker를 사용하면 모든 환경에 애플리케이션을 배포하고 규모를 조정할 수 있습니다.
- [Bash](#)는 GNU 운영 체제의 명령 언어 인터페이스(셸)입니다.
- [Java](#)는 이 패턴에 사용되는 프로그래밍 언어 및 개발 환경입니다.
- [Blu Age](#)는 애플리케이션 코드, 종속성, 인프라를 비롯한 기존 메인프레임 워크로드를 클라우드용 최신 워크로드로 변환하는 AWS Mainframe Modernization 도구입니다.

코드 리포지토리

이 패턴의 코드는 GitHub [Blu Age 샘플 컨테이너](#) 리포지토리에서 사용할 수 있습니다.

모범 사례

- 환경 변수를 사용하여 애플리케이션 동작을 변경하는 변수를 외부화하십시오. 이러한 변수를 사용하면 컨테이너 오케스트레이션 솔루션이 컨테이너를 다시 빌드하지 않고도 런타임 환경을 변경할 수 있습니다. 이 패턴에는 Blu Age 애플리케이션에 유용할 수 있는 환경 변수의 예제가 포함되어 있습니다.

- Blu Age 애플리케이션을 실행하기 전에 애플리케이션 종속성을 확인하십시오. 예를 들어 데이터베이스를 사용할 수 있고 보안 인증이 유효한지 확인합니다. 래퍼 스크립트에 테스트를 작성하여 종속성을 확인하고, 종속성이 충족되지 않으면 조기에 실패합니다.
- 래퍼 스크립트 내에서 자세한 로깅을 사용하십시오. 오케스트레이션 플랫폼과 작업에 걸리는 시간에 따라 실행 중인 컨테이너와 직접 상호 작용하는 것은 어려울 수 있습니다. 문제 진단에 도움이 되도록 유용한 출력이 STDOUT에 작성되었는지 확인하십시오. 예를 들어, 출력에는 애플리케이션 실행 전과 실행 후의 애플리케이션 작업 디렉토리 내용이 포함될 수 있습니다.

에픽

Blu Age 애플리케이션 JAR 파일 입수하기

작업	설명	필요한 기술
<p>옵션 1 - Blu Age를 사용하여 애플리케이션의 JAR 파일을 구합니다.</p>	<p>이 패턴의 컨테이너에는 Blu Age 애플리케이션이 필요합니다. 또는 이 패턴과 함께 제공된 샘플 Java 애플리케이션을 프로토타입에 사용할 수 있습니다.</p> <p>Blu Age 팀과 협력하여 컨테이너에 적용할 수 있는 애플리케이션용 JAR 파일을 구하십시오. JAR 파일을 사용할 수 없는 경우 다음 작업을 참조하여 샘플 애플리케이션을 대신 사용하십시오.</p>	클라우드 아키텍트
<p>옵션 2 - 제공된 샘플 애플리케이션 JAR 파일을 빌드하거나 사용합니다.</p>	<p>이 패턴은 미리 빌드된 샘플 JAR 파일을 제공합니다. 이 파일은 30초 동안 휴면 상태로 있다가 종료되기 전에 애플리케이션의 환경 변수를 STDOUT에 출력합니다.</p>	앱 개발자

작업	설명	필요한 기술
	<p>이 파일은 이름이 bluAgeSample.jar 지정되고 저장소의 docker 폴더에 있습니다.</p> <p>GitHub</p> <p>코드를 변경하여 JAR 파일의 자체 버전을 빌드하려면 에 있는 소스 코드를 사용하십시오. GitHub 리포지토리의 / java_sample/src/sample_java_app.java. Java 소스를 컴파일하고 새 JAR 파일을 빌드하려면 ./java_sample/build.sh에서 빌드 스크립트를 사용할 수 있습니다.</p>	

Blu Age 컨테이너 구축

작업	설명	필요한 기술
GitHub 리포지토리를 복제합니다.	<p>다음 명령을 사용하여 샘플 코드 리포지토리를 복제합니다.</p> <pre>git clone https://github.com/aws-samples/aws-blu-age-sample-container</pre>	AWS DevOps
Docker를 사용하여 컨테이너를 구축하십시오.	컨테이너를 Amazon ECR과 같은 Docker 레지스트리에 푸시하기 전에 Docker를 사용하여 컨테이너를 빌드하십시오.	AWS DevOps

작업	설명	필요한 기술
	<p>1. 선택한 터미널에서 로컬 GitHub 리포지토리 내의 docker 폴더로 이동합니다.</p> <p>2. 다음 명령어를 사용하여 컨테이너를 빌드합니다.</p> <pre data-bbox="630 485 1027 604">docker build -t <tag> .</pre> <p>사용하려는 컨테이너 이름은 <tag> 어디에 있습니까?</p>	
<p>Blu Age 컨테이너를 테스트합니다.</p>	<p>(선택 사항) 필요한 경우 다음 명령을 사용하여 컨테이너를 로컬에서 테스트합니다.</p> <pre data-bbox="594 982 1027 1102">docker run -it <tag> /bin/bash</pre>	<p>AWS DevOps</p>

작업	설명	필요한 기술
<p>Docker 리포지토리에 인증합니다.</p>	<p>Amazon ECR을 사용할 계획이 있는 경우 Amazon ECR 설명서의 지침에 따라 AWS CLI를 설치 및 구성하고 Docker CLI를 기본 레지스트리에 인증하십시오.</p> <p>인증에는 get-login-password 명령을 사용하는 것이 좋습니다.</p> <p>참고: 푸시 명령 보기 버튼을 사용하는 경우 Amazon ECR 콘솔은 이 명령의 미리 채워진 버전을 제공합니다. 자세한 내용은 Amazon ECR 설명서를 참조하십시오.</p> <pre data-bbox="597 1031 1026 1388">aws ecr get-login -password --region <region> docker login --username AWS --password-stdin <account>.dkr.ecr. <region>.amazonaws .com</pre> <p>Amazon ECR을 사용할 계획이 없는 경우 컨테이너 레지스트리 시스템에 제공된 지침을 따르십시오.</p>	<p>AWS DevOps</p>

작업	설명	필요한 기술
<p>컨테이너 리포지토리를 생성합니다.</p>	<p>Amazon ECR 리포지토리를 생성합니다. 지침은 Terraform을 사용하여 컨테이너화된 Blu Age 애플리케이션을 위한 환경 배포 패턴을 참조하십시오.</p> <p>다른 컨테이너 레지스트리 시스템을 사용하는 경우 해당 시스템에 제공된 지침을 따르십시오.</p>	<p>AWS DevOps</p>

작업	설명	필요한 기술
<p>컨테이너에 태그를 지정하고 대상 리포지토리로 푸시합니다.</p>	<p>Amazon ECR을 사용하는 경우:</p> <ol style="list-style-type: none"> 로컬 도커 이미지에 Amazon ECR 레지스트리 및 리포지토리를 태그하여 원격 리포지토리로 푸시할 수 있습니다. <pre data-bbox="630 617 1029 894">docker tag <tag>:latest <account>.dkr.ecr.<region>.amazonaws.com/<repository>:<versionNumber></pre> <ol style="list-style-type: none"> 원격 리포지토리에 이미지 푸시합니다. <pre data-bbox="630 1031 1029 1268">docker push <account>.dkr.ecr.<region>.amazonaws.com/<repository>:<versionNumber></pre> <p>자세한 내용은 Amazon ECR 사용 설명서의 도커 이미지 푸시하기를 참조하십시오.</p>	<p>AWS DevOps</p>

관련 리소스

AWS 리소스

- [AWS Blu Age 샘플 컨테이너 리포지토리](#)
- [서버리스 AWS 인프라에서 현대화된 Blu Age 메인프레임 워크로드 실행](#)
- [Terraform을 사용하여 컨테이너화된 Blu Age 애플리케이션을 위한 환경 배포](#)

- [AWS CLI와 함께 Amazon ECR 사용](#)(Amazon ECR 사용 설명서)
- [프라이빗 레지스트리 인증](#)(Amazon ECR 사용 설명서)
- [Amazon ECS 설명서](#)
- [Amazon EKS 설명서](#)

추가 리소스

- [Blu Age 웹사이트](#)
- [도커 웹사이트](#)

Python을 사용하여 AWS에서 EBCDIC 데이터를 ASCII로 변환 및 압축 해제

작성자: Luis Gustavo Dantas(AWS)

코드 리포지토리: 메인프레임 데이터 유틸리티	환경: PoC 또는 파일럿	소스: Mainframe EBCDIC 데이터
대상: 분산 또는 클라우드로 현대화된 ASCII 데이터	R 유형: 리플랫폼	워크로드: IBM
기술: 메인프레임, 데이터베이스, 스토리지 및 백업, 현대화	AWS 서비스: Amazon EBS, Amazon EC2	

요약

메인프레임은 일반적으로 중요한 비즈니스 데이터를 호스팅하므로 데이터를 Amazon Web Services(AWS) 클라우드 또는 기타 미국 정보 교환 표준 코드(ASCII) 환경으로 마이그레이션할 때 데이터를 현대화하는 것이 가장 중요한 작업 중 하나입니다. 메인프레임에서 데이터는 일반적으로 확장 이진 코드 십진 교환 코드(EBCDIC) 형식으로 인코딩됩니다. 데이터베이스, 가상 스토리지 액세스 방법(VSAM) 또는 플랫폼 파일을 익스포트하면 일반적으로 압축된 바이너리 EBCDIC 파일이 생성되므로 마이그레이션하기가 더 복잡합니다. 가장 일반적으로 사용되는 데이터베이스 마이그레이션 솔루션은 대부분의 경우 데이터 인코딩을 자동으로 변환하는 변경 데이터 캡처(CDC)입니다. 그러나 이러한 데이터베이스, VSAM 또는 플랫폼 파일에는 CDC 메커니즘을 사용하지 못할 수 있습니다. 이러한 파일의 경우 데이터를 현대화하기 위한 대체 접근 방식이 필요합니다.

이 패턴은 EBCDIC 데이터를 ASCII 형식으로 변환하여 현대화하는 방법을 설명합니다. 변환 후에는 데이터를 분산 데이터베이스에 로드하거나 클라우드의 애플리케이션이 데이터를 직접 처리하도록 할 수 있습니다. 패턴은 저장소의 변환 스크립트와 샘플 파일을 사용합니다. [mainframe-data-utilities](#) GitHub

사전 조건 및 제한 사항

사전 조건

- 활성 상태의 AWS 계정
- EBCDIC 입력 파일 및 해당 COBOL(공용 비즈니스 지향 언어) 카피북. 샘플 EBCDIC 파일과 COBOL 카피북이 리포지토리에 포함되어 있습니다. [mainframe-data-utilities](#) GitHub COBOL 카피북

에 대한 자세한 내용은 IBM 웹사이트의 [zEnterprise COBOL for z/OS 6.4 프로그래밍 가이드](#)를 참조하세요.

제한 사항

- COBOL 프로그램 내에 정의된 파일 레이아웃은 지원되지 않습니다. 이는 별도로 제공되어야 합니다.

제품 버전

- Python 버전 3.8 이상

아키텍처

소스 기술 스택

- 메인프레임의 EBCDIC 데이터
- COBOL 카피북

대상 기술 스택

- Virtual Private Cloud(VPC)의 Amazon Elastic Compute Cloud(Amazon EC2) 인스턴스
- Amazon Elastic Block Store(Amazon EBS)
- Python과 그 필수 패키지, JavaScript 객체 표기법 (JSON), sys, 그리고 날짜/시간
- 최신 애플리케이션에서 읽을 수 있거나 관계형 데이터베이스 테이블에 로드할 수 있는 ASCII 플랫폼 파일

대상 아키텍처

아키텍처 다이어그램은 EC2 인스턴스에서 EBCDIC 파일을 ASCII 파일로 변환하는 프로세스를 보여줍니다.

1. parse_copybook_to_json.py 스크립트를 사용하여 COBOL 카피북을 JSON 파일로 변환합니다.
2. JSON 파일과 extract_ebcdic_to_ascii.py 스크립트를 사용하여 EBCDIC 데이터를 ASCII 파일로 변환합니다.

자동화 및 규모 조정

첫 번째 수동 파일 변환에 필요한 리소스를 확보한 후 파일 변환을 자동화할 수 있습니다. 이 패턴에는 자동화 지침이 포함되어 있지 않습니다. 여러 가지 방법으로 변환을 자동화할 수 있습니다. 다음 사항은 한 가지 가능한 접근 방식에 대한 개요입니다.

1. AWS Command Line Interface(AWS CLI) 및 Python 스크립트 명령을 셸 스크립트로 캡슐화합니다.
2. 셸 스크립트 작업을 EC2 인스턴스에 비동기적으로 제출하는 AWS Lambda 함수를 생성합니다. 자세한 내용은 [AWS Lambda를 사용한 SSH 작업 예약](#)을 참조하세요.
3. 레거시 파일이 업로드될 때마다 Lambda 함수를 간접 호출하는 Amazon Simple Storage Service(S3) 트리거를 생성합니다. 자세한 내용은 [Amazon S3 트리거를 사용하여 Lambda 함수 간접 호출](#)을 참조하세요.

도구

서비스

- [Amazon Elastic Compute Cloud\(Amazon EC2\)](#)는 클라우드에서 확장 가능한 컴퓨팅 용량을 제공합니다. 필요한 만큼 가상 서버를 시작하고 빠르게 스케일 업하거나 스케일 다운할 수 있습니다.
- [Amazon Elastic Block Store\(Amazon EBS\)](#)는 Amazon Elastic Compute Cloud(Amazon EC2) 인스턴스에 사용할 수 있는 블록 스토리지 볼륨을 제공합니다.
- [AWS Command Line Interface\(AWS CLI\)](#)는 명령줄 셸에서 명령을 사용하여 AWS 서비스와 상호 작용할 수 있는 오픈 소스 도구입니다.
- [AWS Identity and Access Management\(IAM\)](#)는 사용자에 대한 인증 및 권한 부여를 제어함으로써 AWS 리소스에 대한 액세스를 안전하게 관리할 수 있습니다.

기타 도구

- [GitHub](#) 협업 도구와 버전 제어를 제공하는 코드 호스팅 서비스입니다.
- [Python](#)은 고급 프로그래밍 언어입니다.

코드 리포지토리

이 패턴의 코드는 저장소에서 사용할 수 있습니다. [mainframe-data-utilities](#) GitHub

에픽

EC2 인스턴스 준비

작업	설명	필요한 기술
EC2 인스턴스를 시작합니다.	<p>EC2 인스턴스에는 아웃바운드 인터넷 액세스가 있어야 합니다. 이렇게 하면 인스턴스가 에서 제공되는 Python 소스 코드에 액세스할 수 GitHub 있습니다. 인스턴스를 생성하는 방법:</p> <ol style="list-style-type: none"> 1. https://console.aws.amazon.com/ec2 에서 Amazon EC2 콘솔을 엽니다. 2. EC2 Linux 인스턴스를 실행합니다. 퍼블릭 IP 주소를 사용하고 포트 22를 통한 인바운드 액세스를 허용합니다. 인스턴스의 스토리지 크기가 EBCDIC 데이터 파일 크기의 두 배 이상인지 확인합니다. 지침은 Amazon EC2 설명서를 참조하세요. 	일반 AWS
Git을 설치합니다.	<ol style="list-style-type: none"> 1. Secure Shell(SSH) 클라이언트를 사용하여 방금 시작한 EC2 인스턴스에 연결합니다. 자세한 내용은 Linux 인스턴스에 연결을 참조하세요. 2. Amazon EC2 콘솔에서 다음 명령을 실행합니다. EC2 인스턴스에 Git가 설치됩니다. 	일반 AWS, Linux

작업	설명	필요한 기술
	<pre>sudo yum install git</pre> <p>3. 다음 명령을 실행하고 Git이 성공적으로 설치되었는지 확인합니다.</p> <pre>git --version</pre>	

작업	설명	필요한 기술
Python을 설치합니다.	<ol style="list-style-type: none"> 1. Amazon EC2 콘솔에서 다음 명령을 실행합니다. EC2 인스턴스에 Python이 설치됩니다. <pre data-bbox="630 443 1027 562">sudo yum install python3</pre> 2. Amazon EC2 콘솔에서 다음 명령을 실행합니다. EC2 인스턴스에 Pip3가 설치됩니다. <pre data-bbox="630 793 1027 913">sudo yum install python3-pip</pre> 3. Amazon EC2 콘솔에서 다음 명령을 실행합니다. EC2 인스턴스에 AWS SDK for Python(Boto3)가 설치됩니다. <pre data-bbox="630 1192 1027 1312">sudo pip3 install boto3</pre> 4. Amazon EC2 콘솔에서 다음 명령을 실행합니다. 여기에서 <us-east-1> 는 AWS 리전의 코드입니다. 리전 코드의 전체 목록은 Amazon EC2 설명서의 사용 가능한 리전을 참조하세요. <pre data-bbox="630 1682 1027 1843">export AWS_DEFAULT_REGION=<us-east-1></pre> 	일반 AWS, Linux

작업	설명	필요한 기술
GitHub 리포지토리를 복제합니다.	<p>1. Amazon EC2 콘솔에서 다음 명령을 실행합니다. 그러면 기본 복사 위치인 home 폴더에서 mainframe-data-utilities 리포지토리가 GitHub 복제되고 열립니다.</p> <pre>git clone https://github.com/aws-samples/mainframe-data-utilities.git</pre> <p>2. home 폴더에 mainframe-data-utilities 폴더가 있는지 확인합니다.</p>	일반 AWS, GitHub

EBCDIC 데이터에서 ASCII 파일 생성

작업	설명	필요한 기술
COBOL 카피북을 JSON 레이아웃 파일로 파싱합니다.	<p>mainframe-data-utilities 폴더 내에서 parse_copybook_to_json.py 스크립트를 실행합니다. 이 자동화 모듈은 COBOL 카피북에서 파일 레이아웃을 읽고 JSON 파일을 생성합니다. JSON 파일에는 소스 파일에서 데이터를 해석하고 추출하는 데 필요한 정보가 들어 있습니다. 그러면 COBOL 카피북에서 JSON 메타데이터가 생성됩니다.</p> <p>다음 명령은 COBOL 카피북을 JSON 파일로 변환합니다.</p>	일반 AWS, Linux

작업	설명	필요한 기술
	<pre>python3 parse_copybook_to_json.py \ -copybook LegacyReference/COBPack2.cpy \ -output sample-data/cobpack2-list.json \ -dict sample-data/cobpack2-dict.json \ -ebcdic sample-data/COBPack.outfile.txt \ -ascii sample-data/COBPack.ascii.txt \ -print 10000</pre> <p>스크립트는 수신된 인수를 인쇄합니다.</p> <pre>----- ----- ----- ----- Copybook file..... LegacyReference/COBPack2.cpy Parsed copybook (JSON List). sample-data/cobpack2-list.json JSON Dict (documentation)... sample-data/cobpack2-dict.json ASCII file..... sample-data/COBPack.ascii.txt EBCDIC file..... sample-data/COBPack.outfile.txt Print each..... 10000</pre>	

작업	설명	필요한 기술
	<div style="border: 1px solid #ccc; border-radius: 10px; padding: 10px; margin-bottom: 10px;"> <p>-----</p> <p>-----</p> <p>-----</p> <p>-----</p> </div> <p>인수에 대한 자세한 내용은 GitHub 리포지토리의 README 파일을 참조하십시오.</p>	

작업	설명	필요한 기술
<p>JSON 레이아웃 파일을 검사합니다.</p>	<ol style="list-style-type: none"> 1. parse_copybook_to_json.py 스크립트에 정의된 출력 경로로 이동합니다. 2. sample-data/cobpack2-list.json 파일의 생성 시간을 확인하여 적절한 JSON 레이아웃 파일을 선택했는지 확인합니다. 3. JSON 파일을 검사하여 내용이 다음과 비슷한지 확인합니다. <pre data-bbox="597 835 1026 1625"> "input": "extract-ebcdic-to-ascii/COBPACK.OUTFILE.txt", "output": "extract-ebcdic-to-ascii/COBPACK.ASCII.txt", "max": 0, "skip": 0, "print": 10000, "lrecl": 150, "rem-low-values": true, "separator": " ", "transf": [{ "type": "ch", "bytes": 19, "name": "OUTFILE-TEXT" } </pre> <p>JSON 레이아웃 파일의 가장 중요한 속성은 다음과 같습니다.</p>	<p>일반 AWS, JSON</p>

작업	설명	필요한 기술
	<ul style="list-style-type: none"> • <code>input</code> - 변환할 EBCDIC 파일의 경로를 포함합니다 • <code>output</code> - ASCII 파일이 생성될 경로를 정의합니다 • <code>lrecl</code> - 로직 레코드 길이의 크기를 바이트 단위로 지정합니다 • <code>transf</code> - 모든 필드와 해당 크기를 바이트 단위로 나열합니다. <p>JSON 레이아웃 파일에 대한 자세한 내용은 리포지토리의 README 파일을 참조하십시오. GitHub</p>	

작업	설명	필요한 기술
ASCII 파일을 생성합니다.	<p>복제된 GitHub 리포지토리에 포함된 <code>extract_ebcdic_to_ascii.py</code> 스크립트를 실행합니다. 이 스크립트는 EBCDIC 파일을 읽고 변환하여 읽을 수 있는 ASCII 파일을 작성합니다.</p> <pre data-bbox="597 537 1026 735">python3 extract_ebcdic_to_ascii.py -local-json sample-data/cobpack2-list.json</pre> <p>스크립트는 EBCDIC 데이터를 처리할 때 10,000개 레코드의 모든 배치에 대해 메시지를 인쇄합니다. 다음 예를 참조하세요.</p> <pre data-bbox="597 1037 1026 1759">----- ----- ----- ----- 2023-05-15 21:21:46. 322253 Local Json file -local-json sample-data/cobpack2- list.json 2023-05-15 21:21:47. 034556 Records processed 10000 2023-05-15 21:21:47. 736434 Records processed 20000 2023-05-15 21:21:48. 441696 Records processed 30000</pre>	일반 AWS

작업	설명	필요한 기술
	<pre> 2023-05-15 21:21:49. 173781 Records processed 40000 2023-05-15 21:21:49. 874779 Records processed 50000 2023-05-15 21:21:50. 705873 Records processed 60000 2023-05-15 21:21:51. 609335 Records processed 70000 2023-05-15 21:21:52. 292989 Records processed 80000 2023-05-15 21:21:52. 938366 Records processed 89280 2023-05-15 21:21:52. 938448 Seconds 6.616232 </pre> <p>인쇄 빈도를 변경하는 방법에 대한 자세한 내용은 저장소의 README 파일을 참조하십시오. GitHub</p>	

작업	설명	필요한 기술
ASCII 파일을 검사합니다.	<ol style="list-style-type: none"> extract-ebcdic-to-ascii/COBPACK.ASCII.txt 파일의 생성 시간을 확인하여 파일이 최근에 생성되었는지 확인하십시오. Amazon EC2 콘솔에서 다음 명령을 입력합니다. 그러면 ASCII 파일의 첫 번째 레코드가 열립니다. <div data-bbox="630 688 1027 848" style="border: 1px solid #ccc; border-radius: 10px; padding: 10px; margin: 10px 0;"> <pre>head sample-data/COBPACK.ASCII.txt -n 1 xxd</pre> </div> 첫 번째 레코드의 내용을 살펴보세요. EBCDIC 파일은 일반적으로 바이너리이므로 캐리지 리턴 및 라인 피드(CRLF) 특수 문자가 없습니다. extract_ebcdic_to_ascii.py 스크립트는 스크립트 파라미터에 정의된 열 구분 기호로 파이프 문자를 추가합니다. <p>제공된 샘플 EBCDIC 파일을 사용한 경우 ASCII 파일의 첫 번째 레코드는 다음과 같습니다.</p> <div data-bbox="597 1619 1027 1831" style="border: 1px solid #ccc; border-radius: 10px; padding: 10px; margin: 10px 0;"> <pre>00000000: 2d30 3030 3030 3030 3030 3130 3030 3030 -0000000000100000 00000010: 3030 307c 3030 3030 3030 3030</pre> </div>	

작업	설명	필요한 기술
	<pre> 3031 3030 000 00000 0000100 00000020: 3030 3030 3030 7c2d 3030 3030 3030 3030 000000 -0 00000000 00000030: 3031 3030 3030 3030 3030 7c30 7c30 7c31 0100000000 0 0 1 00000040: 3030 3030 3030 3030 7c2d 3130 3030 3030 00000000 -100000 00000050: 3030 307c 3130 3030 3030 3030 307c 2d31 000 10000 0000 -1 00000060: 3030 3030 3030 3030 7c30 3030 3030 7c30 00000000 00000 0 00000070: 3030 3030 7c31 3030 3030 3030 3030 7c2d 0000 1000 00000 - 00000080: 3130 3030 3030 3030 307c 3030 3030 3030 100000000 000000 00000090: 3030 3030 3130 3030 3030 3030 307c 2d30 000010000 0000 -0 000000a0: 3030 3030 3030 3030 3031 3030 3030 3030 000000000 1000000 000000b0: 3030 7c41 7c41 7c0a 00 A A . </pre>	

작업	설명	필요한 기술
<p>EBCDIC 파일을 평가하세요.</p>	<p>Amazon EC2 콘솔에서 다음 명령을 입력합니다. 그러면 EBCDIC 파일의 첫 번째 레코드가 열립니다.</p> <pre data-bbox="597 443 1026 600">head sample-data/COBPAC K.OUTFILE.txt -c 150 xxd</pre> <p>샘플 EBCDIC 파일을 사용한 경우 결과는 다음과 같습니다.</p> <pre data-bbox="597 758 1026 1799">00000000: 60f0 f0f0 f0f0 f0f0 f0f0 f1f0 f0f0 f0f0 `..... 00000010: f0f0 f0f0 f0f0 f0f0 f0f0 f0f0 f1f0 f0f0 00000020: f0f0 f0f0 f0f0 f0f0 f0f0 f0f0 f0f0 f1f0 00000030: f0f0 f0f0 f0f0 d000 0000 0005 f5e1 00fa 00000040: 0a1f 0000 0000 0005 f5e1 00ff ffff fffa 00000050: 0a1f 0000 000f 0000 0c10 0000 000f 1000 00000060: 0000 0d00 0000 0000 1000 0000</pre>	<p>일반 AWS, Linux, EBCDIC</p>

작업	설명	필요한 기술
	<pre> 0f00 0000 00000070: 0000 1000 0000 0dc1 c100 0000 0000 0000 00000080: 0000 0000 0000 0000 0000 0000 0000 0000 00000090: 0000 0000 0000 </pre> <p>소스 파일과 대상 파일 간의 동등성을 평가하려면 EBCDIC에 대한 포괄적인 지식이 필요합니다. 예를 들어 샘플 EBCDIC 파일의 첫 문자는 하이픈(-)입니다. EBCDIC 파일의 16진수 표기법에서는 이 문자가 60으로 표시되고 ASCII 파일의 16진수 표기법에서는 이 문자가 2D로 표시됩니다. EBCDIC에서 ASCII로의 변환 표는 IBM 웹사이트의 EBCDIC에서 ASCII로 변환하는 표를 참조하세요.</p>	

관련 리소스

참조

- [EBCDIC 문자 세트](#) (IBM 설명서)
- [EBCDIC에서 ASCII로](#)(IBM 설명서)
- [COBOL](#)(IBM 설명서)
- [기본 JCL 개념](#)(IBM 설명서)

- [Linux 인스턴스에 연결](#)(Amazon EC2 설명서)

자습서

- [AWS Lambda를 사용하여 SSH 작업 예약](#)(AWS 블로그 게시물)
- [Amazon S3 트리거를 사용하여 Lambda 함수 간접 호출](#)(AWS Lambda 설명서)

AWS Lambda를 사용하여 Amazon S3에서 메인프레임 파일을 EBCDIC 형식에서 문자로 구분된 ASCII 형식으로 변환합니다.

작성자: Luis Gustavo Dantas(AWS)

코드 리포지토리: 메인프레임 데이터 유틸리티	환경: PoC 또는 파일럿	소스: IBM EBCDIC 파일
대상: 구분된 ASCII 파일	R 유형: 리플랫폼	워크로드: IBM
기술: 메인프레임	AWS 서비스: AWS CloudShell, AWS Lambda, 아마존 S3, 아마존 CloudWatch	

요약

이 패턴은 메인프레임 EBCDIC (확장 이진 코딩 십진 교환 코드) 파일을 문자로 구분된 ASCII (정보 교환을 위한 미국 표준 코드) 파일로 자동 변환하는 AWS Lambda 함수를 시작하는 방법을 보여줍니다. Lambda 함수는 ASCII 파일이 Amazon Simple Storage Service(S3) 버킷에 업로드된 후에 실행됩니다. 파일 변환 후 x86 기반 워크로드에서 ASCII 파일을 읽거나 파일을 최신 데이터베이스로 로드할 수 있습니다.

이 패턴에서 설명하는 파일 변환 접근 방식은 최신 환경에서 EBCDIC 파일을 사용할 때 발생하는 문제를 해결하는 데 도움이 될 수 있습니다. EBCDIC로 인코딩된 파일에는 2진수 또는 압축 10진수 형식으로 표현된 데이터가 포함되는 경우가 많으며 필드는 고정 길이입니다. 최신 x86 기반 워크로드 또는 분산 환경은 일반적으로 ASCII로 인코딩된 데이터를 사용하며 EBCDIC 파일을 처리할 수 없기 때문에 이러한 특성으로 인해 장애가 발생합니다.

사전 조건 및 제한 사항

사전 조건

- 활성 상태의 AWS 계정.
- S3 버킷
- 관리자 권한이 있는 AWS Identity and Access Management (IAM) 사용자
- AWS CloudShell

- [Python 3.8.0](#) 이상
- EBCDIC 및 해당 데이터 구조로 공통 비즈니스 지향 언어 (COBOL) 카피북으로 인코딩된 플랫폼 파일

참고: 이 패턴은 샘플 EBCDIC 파일([CLIENT.EBCDIC.txt](#))과 해당 COBOL 카피북([COBKS05.cpy](#))을 사용합니다. 두 파일 모두 GitHub [mainframe-data-utilities](#) 리포지토리에서 사용할 수 있습니다.

제한 사항

- COBOL 카피북에는 일반적으로 여러 레이아웃 정의가 들어 있습니다. [mainframe-data-utilities](#) 프로젝트는 이런 종류의 카피북을 파싱할 수는 있지만 데이터 변환 시 어떤 레이아웃을 고려할지 유추할 수는 없습니다. 카피북에는 이 로직이 들어 있지 않기 때문입니다 (대신 COBOL 프로그램에 남아 있음). 따라서 카피북을 파싱한 후에는 레이아웃 선택 규칙을 수동으로 구성해야 합니다.
- 이 패턴에는 [Lambda 할당량](#)이 적용됩니다.

아키텍처

소스 기술 스택

- IBM z/OS, IBM i 및 기타 EBCDIC 시스템
- EBCDIC로 인코딩된 데이터가 포함된 순차 파일 (예: IBM Db2 언로드)
- COBOL 카피북 레이아웃

대상 기술 스택

- Amazon S3
- Amazon S3 이벤트 알림
- IAM
- Lambda 함수
- Python 3.8 이상
- 메인프레임 데이터 유틸리티
- JSON 메타데이터
- 문자로 구분된 ASCII 파일

대상 아키텍처

다음 다이어그램은 메인프레임 EBCDIC 파일을 ASCII 파일로 변환하는 아키텍처를 보여줍니다.

이 다이어그램은 다음 워크플로를 보여줍니다.

1. 사용자는 카피북 파서 스크립트를 실행하여 COBOL 카피북을 JSON 파일로 변환합니다.
2. 사용자가 JSON 메타데이터를 S3 버킷에 업로드합니다. 이렇게 하면 데이터 변환 Lambda 함수에서 메타데이터를 읽을 수 있습니다.
3. 사용자 또는 자동화된 프로세스가 EBCDIC 파일을 S3 버킷에 업로드합니다.
4. S3 알림 이벤트는 데이터 변환 Lambda 함수를 트리거합니다.
5. AWS는 Lambda 함수에 대한 S3 버킷 읽기/쓰기 권한을 확인합니다.
6. Lambda는 S3 버킷에서 파일을 읽고 EBCDIC에서 ASCII로 파일을 로컬로 변환합니다.
7. Lambda는 Amazon에 프로세스 상태를 기록합니다. CloudWatch
8. Lambda는 ASCII 파일을 Amazon S3에 다시 기록합니다.

참고: 카피북 파서 스크립트는 메타데이터를 JSON으로 변환한 다음 해당 데이터를 S3 버킷에 업로드한 후 한 번만 실행됩니다. 초기 변환 후에는 S3 버킷에 업로드된 것과 동일한 JSON 파일을 사용하는 EBCDIC 파일이 동일한 메타데이터를 사용합니다.

도구

도구

- [Amazon](#)은 AWS 리소스의 지표와 AWS에서 실행하는 애플리케이션을 실시간으로 모니터링할 수 있도록 CloudWatch 도와줍니다.
- [Amazon Simple Storage Service\(S3\)](#)는 원하는 양의 데이터를 저장, 보호 및 검색하는 데 도움이 되는 클라우드 기반 객체 스토리지 서비스입니다.
- [CloudShell](#)은 [AWS](#) 명령줄 인터페이스 (AWS CLI) 및 사전 설치된 다양한 개발 도구를 사용하여 AWS 서비스를 관리하는 데 사용할 수 있는 브라우저 기반 셸입니다.
- [AWS Identity and Access Management\(IAM\)](#)를 사용하면 사용자에 대해 인증 및 권한 부여를 제어함으로써 AWS 리소스에 대한 액세스를 안전하게 관리할 수 있습니다.
- [AWS Lambda](#)는 서버를 프로비저닝하거나 관리할 필요 없이 코드를 실행하는 데 도움이 되는 컴퓨팅 서비스입니다. Lambda는 필요 시에만 코드를 실행하며 자동으로 확장이 가능하므로 사용한 컴퓨팅 시간만큼만 비용을 지불하면 됩니다.

기타 도구

- [GitHub](#) 협업 도구와 버전 제어를 제공하는 코드 호스팅 서비스입니다.
- [Python](#)은 고급 프로그래밍 언어입니다.

코드

이 패턴의 코드는 저장소에서 사용할 수 있습니다. GitHub [mainframe-data-utilities](#)

모범 사례

다음 모범 사례를 고려하세요.

- Amazon 리소스 이름 (ARN) 수준에서 필요한 권한을 설정합니다.
- 항상 IAM 정책에 최소 권한 권한을 부여하십시오. 자세한 내용은 [IAM 설명서의 IAM의 보안 모범 사례](#)를 참조하십시오.

에픽

환경 변수 및 작업 폴더 생성

작업	설명	필요한 기술
환경 변수를 생성합니다.	<p>다음 환경 변수를 텍스트 편집기에 복사한 다음 <placeholder> 다음 예제의 값을 리소스 값으로 바꾸십시오.</p> <pre>bucket=<your_bucket_name> account=<your_account_number> region=<your_region_code></pre> <p>참고: 나중에 S3 버킷, AWS 계정 및 AWS 리전에 대한 참조를 생성할 것입니다.</p>	일반 AWS

작업	설명	필요한 기술
	<p>환경 변수를 정의하려면 CloudShell 콘솔을 열고 업데이트된 환경 변수를 복사하여 명령줄에 붙여넣습니다.</p> <p>참고: CloudShell 세션이 다시 시작될 때마다 이 단계를 반복해야 합니다.</p>	
작업 폴더 생성	<p>나중에 리소스 정리 프로세스를 단순화하려면 다음 명령을 CloudShell 실행하여 에서 작업 폴더를 만드십시오.</p> <pre>mkdir workdir; cd workdir</pre> <p>참고: CloudShell 세션과의 연결이 끊길 때마다 디렉토리를 작업 디렉토리 (workdir) 로 변경해야 합니다.</p>	일반 AWS

IAM 역할 및 정책 정의

작업	설명	필요한 기술
Lambda 함수에 대해 신뢰 정책을 생성합니다.	<p>EBCDIC 변환기는 람다 함수에서 실행됩니다. 이 함수에는 IAM 역할이 있어야 합니다. IAM 역할을 생성하기 전에 리소스가 해당 정책을 수임할 수 있도록 하는 신뢰 정책 문서를 정의해야 합니다.</p>	일반 AWS

작업	설명	필요한 기술
	<p>CloudShell 작업 폴더에서 다음 명령을 실행하여 정책 문서를 생성합니다.</p> <pre data-bbox="597 380 1024 1331">E2ATrustPol=\$(cat <<EOF { "Version": "2012-10-17", "Statement": [{ "Effect": "Allow", "Principa 1": { "Service": "lambda.a mazonaws.com" }, "Action": "sts:AssumeRole" }] } EOF) printf "\$E2ATrustPol" > E2ATrustPol.json</pre>	
<p>Lambda 변환을 위한 IAM 역할을 생성합니다.</p>	<p>IAM 역할을 생성하려면 작업 폴더에서 다음 AWS CLI 명령을 실행합니다 CloudShell .</p> <pre data-bbox="597 1541 1024 1814">aws iam create-role --role-name E2AConvLa mbdaRole --assume- role-policy-docume nt file://E2ATrustPol .json</pre>	<p>일반 AWS</p>

작업	설명	필요한 기술
<p>Lambda 함수에 대한 IAM 정책 문서를 생성합니다.</p>	<p>Lambda 함수는 S3 버킷에 대한 읽기/쓰기 액세스 권한과 Amazon Logs에 대한 쓰기 권한을 가져야 합니다.</p> <p>CloudWatch</p> <p>IAM 정책을 생성하려면 작업 폴더에서 다음 명령을 실행하십시오. CloudShell</p> <pre data-bbox="592 661 1031 1871"> E2APolicy=\$(cat <<EOF { "Version": "2012-10-17", "Statement": [{ "Sid": "Logs", "Effect": "Allow", "Action": ["logs:PutLogEvents", "logs:CreateLogStream", "logs:CreateLogGroup"], "Resource": ["arn:aws:logs:*:*:log-group:*", "arn:aws:logs:*:*:log-group:*:log-stream:*"] }], } </pre>	<p>일반 AWS</p>

작업	설명	필요한 기술
	<pre> { "Sid": "S3", "Effect": "Allow", "Action": ["s3:GetObject", "s3:PutObject", "s3:GetObjectVersion"], "Resource": ["arn:aws:s3:::%s/*", "arn:aws:s3:::%s"] }] } EOF) printf "\$E2APolicy" "\$bucket" "\$bucket" > E2AConvLambdaPolici y.json </pre>	
IAM 정책 문서를 IAM 역할에 첨부합니다.	<p>IAM 정책을 IAM 역할에 연결하려면 작업 폴더에서 다음 명령을 실행합니다. CloudShell</p> <pre> aws iam put-role-policy --role-name E2AConvLa mbdaRole --policy-name E2AConvLambdaPolici y --policy-document file://E2AConvLamb daPolicy.json </pre>	일반 AWS

EBCDIC 변환을 위한 Lambda 함수를 생성합니다.

작업	설명	필요한 기술
EBCDIC 변환 소스 코드를 다운로드하십시오.	CloudShell 작업 폴더에서 다음 명령을 실행하여 mainframe-data-utilities 소스 코드를 다운로드합니다. GitHub <pre data-bbox="594 548 1027 747">git clone https://github.com/aws-samples/mainframe-data-utilities.git mdu</pre>	일반 AWS
ZIP 패키지를 생성하십시오.	CloudShell 작업 폴더에서 다음 명령을 실행하여 EBCDIC 변환을 위한 Lambda 함수를 생성하는 ZIP 패키지를 생성합니다. <pre data-bbox="594 1003 1027 1121">cd mdu; zip ../mdu.zip *.py; cd ..</pre>	일반 AWS
Lambda 함수를 생성합니다.	CloudShell 작업 폴더에서 다음 명령을 실행하여 EBCDIC 변환을 위한 Lambda 함수를 생성합니다. <pre data-bbox="594 1377 1027 1862">aws lambda create-function \ --function-name E2A \ --runtime python3.9 \ --zip-file fileb://mdu.zip \ --handler extract_ebcdic_to_ascii.lambda_handler \ --role arn:aws:iam::<account>:role/E2AConvLambdaRole \</pre>	일반 AWS

작업	설명	필요한 기술
	<pre data-bbox="597 212 1024 386">--timeout 10 \ --environment "Variables={layout=\$bucket/layout/}"</pre> <p data-bbox="597 422 1024 596">참고: 환경 변수 레이아웃은 Lambda 함수에 JSON 메타데이터가 있는 위치를 알려줍니다.</p>	
<p data-bbox="110 646 513 726">Lambda 함수에 대한 리소스 기반 정책을 생성합니다.</p>	<p data-bbox="597 646 1024 869">CloudShell 작업 폴더에서 다음 명령을 실행하여 Amazon S3 이벤트 알림이 EBCDIC 변환을 위한 Lambda 함수를 트리거하도록 허용하십시오.</p> <pre data-bbox="597 911 1024 1423">aws lambda add-permission \ --function-name E2A \ --action lambda:InvokeFunction \ --principal s3.amazonaws.com \ --source-arn arn:aws:s3:::\$bucket \ --source-account \$account \ --statement-id 1</pre>	<p data-bbox="1068 646 1214 676">일반 AWS</p>

Amazon S3 이벤트 알림 만들기

작업	설명	필요한 기술
<p data-bbox="110 1701 529 1780">Amazon S3 이벤트 알림용 구성 문서를 생성합니다.</p>	<p data-bbox="597 1701 1024 1873">Amazon S3 이벤트 알림은 파일이 입력 폴더에 배치될 때 EBCDIC 변환 Lambda 함수를 시작합니다.</p>	<p data-bbox="1068 1701 1214 1730">일반 AWS</p>

작업	설명	필요한 기술
	<p>CloudShell 작업 폴더에서 다음 명령을 실행하여 Amazon S3 이벤트 알림을 위한 JSON 문서를 생성합니다.</p> <pre data-bbox="592 426 1029 1778"> { "LambdaFunctionConfigurations": [{ "Id": "E2A", "LambdaFunctionArn": "arn:aws:lambda:%s:%s:function:E2A", "Events": ["s3:ObjectCreated:Put"], "Filter": { "Key": { "FilterRules": [{ "Name": "prefix", "Value": "input/" }] }] }] } EOF) printf "\$S3E2AEvent" "\$region" "\$account" > S3E2AEvent.json </pre>	

작업	설명	필요한 기술
Amazon S3 이벤트 알림을 만듭니다.	CloudShell 작업 폴더에서 다음 명령을 실행하여 Amazon S3 이벤트 알림을 생성합니다. <pre>aws s3api put-bucket-notification-configuration --bucket \$bucket --notification-configuration file://S3E2AEvent.json</pre>	일반 AWS

JSON 메타데이터 생성 및 업로드

작업	설명	필요한 기술
COBOL 카피북을 분석하십시오.	CloudShell 작업 폴더에서 다음 명령을 실행하여 샘플 COBOL 카피북을 JSON 파일로 파싱합니다 (데이터 파일을 올바르게 읽고 분할하는 방법을 정의). <pre>python3 mdu/parse_copybook_to_json.py \ -copybook mdu/LegacyReference/COBK05.cpy \ -output CLIENT.json \ -output-s3key CLIENT.ASCII.txt \ -output-s3bkt \$bucket \ -output-type s3 \ -print 25</pre>	일반 AWS

작업	설명	필요한 기술
<p>변환 규칙을 추가합니다.</p>	<p>샘플 데이터 파일과 해당 COBOL 카피북은 다중 레이아웃 파일입니다. 즉, 변환 시 특정 규칙에 따라 데이터를 분할해야 합니다. 이 경우 각 행의 위치 3과 4에 있는 바이트가 레이아웃을 정의합니다.</p> <p>CloudShell 작업 폴더에서 CLIENT.json 파일을 편집하고 내용을 다음에서 다음으로 변경합니다. "transf-rule": [],</p> <pre data-bbox="597 856 1026 1453">"transf-rule": [{ "offset": 4, "size": 2, "hex": "0002", "transf": "transf1" }, { "offset": 4, "size": 2, "hex": "0000", "transf": "transf2" }],</pre>	<p>일반 AWS, IBM 메인프레임, 코볼</p>
<p>JSON 메타데이터를 S3 버킷에 업로드합니다.</p>	<p>CloudShell 작업 폴더에서 다음 AWS CLI 명령을 실행하여 JSON 메타데이터를 S3 버킷에 업로드합니다.</p> <pre data-bbox="597 1705 1026 1864">aws s3 cp CLIENT.json s3://\$bucket/layout/ CLIENT.json</pre>	<p>일반 AWS</p>

EBCDIC 파일을 변환하십시오.

작업	설명	필요한 기술
EBCDIC 파일을 S3 버킷으로 전송합니다.	<p>CloudShell 작업 폴더에서 다음 명령을 실행하여 EBCDIC 파일을 S3 버킷으로 전송합니다.</p> <pre data-bbox="597 499 1027 659">aws s3 cp mdu/sample-data/CLIENT.EBCDIC.txt s3://\$bucket/input/</pre> <p>참고: ASCII 파일이 S3 버킷에 업로드될 때 Lambda 변환 함수를 다시 직접 호출하지 않도록 입력 (EBCDIC) 및 출력 (ASCII) 파일에 대해 서로 다른 폴더를 설정하는 것이 좋습니다.</p>	일반 AWS
출력 결과를 확인합니다.	<p>CloudShell 작업 폴더에서 다음 명령을 실행하여 S3 버킷에 ASCII 파일이 생성되었는지 확인합니다.</p> <pre data-bbox="597 1276 1027 1360">awss3 ls s3://\$bucket/</pre> <p>참고: 데이터 변환은 몇 초 정도 걸릴 수 있습니다. ASCII 파일을 몇 번 확인하는 것이 좋습니다.</p> <p>ASCII 파일을 사용할 수 있게 되면 다음 명령을 실행하여 S3 버킷에서 현재 폴더로 파일을 다운로드합니다.</p>	일반 AWS

작업	설명	필요한 기술
	<pre>aws s3 cp s3://\$bucket/CLIENT.ASCII.txt .</pre> <p>ASCII 파일 내용을 확인하십시오.</p> <pre>head CLIENT.ASCII.txt</pre>	

환경을 청소합니다

작업	설명	필요한 기술
(선택 사항) 변수와 폴더를 준비합니다.	<p>연결이 끊긴 CloudShell 경우 다시 연결한 후 다음 명령을 실행하여 디렉터리를 작업 폴더로 변경합니다.</p> <pre>cd workdir</pre> <p>환경 변수가 정의되어 있는지 확인하십시오.</p> <pre>bucket=<your_bucket_name> account=<your_account_number> region=<your_region_code></pre>	일반 AWS
버킷의 알림 구성을 제거합니다.	CloudShell 작업 폴더에서 다음 명령을 실행하여 Amazon S3 이벤트 알림 구성을 제거합니다.	일반 AWS

작업	설명	필요한 기술
	<pre>aws s3api put-bucket-notification-configuration \ --bucket=\$bucket \ --notification-configuration="{}</pre>	
<p>Lambda 함수를 삭제합니다.</p>	<p>CloudShell 작업 폴더에서 다음 명령을 실행하여 EBCDIC 변환기의 Lambda 함수를 삭제합니다.</p> <pre>aws lambda delete-function --function-name E2A</pre>	<p>일반 AWS</p>
<p>IAM 역할 및 정책을 삭제합니다.</p>	<p>CloudShell 작업 폴더에서 다음 명령을 실행하여 EBCDIC 변환기 역할 및 정책을 제거합니다.</p> <pre>aws iam delete-role-policy --role-name E2AConvLambdaRole --policy-name E2AConvLambdaPolicy aws iam delete-role --role-name E2AConvLambdaRole</pre>	<p>일반 AWS</p>

작업	설명	필요한 기술
S3 버킷에서 생성된 파일을 삭제합니다.	CloudShell 작업 폴더에서 다음 명령을 실행하여 S3 버킷에서 생성된 파일을 삭제합니다. <pre>aws s3 rm s3://\$bucket/ layout --recursive aws s3 rm s3://\$bucket/ input --recursive aws s3 rm s3://\$bucket/ CLIENT.ASCII.txt</pre>	일반 AWS
작업 폴더를 삭제합니다.	CloudShell 작업 폴더에서 다음 명령을 <code>workdir</code> 실행하여 내용을 제거합니다. <pre>cd ..; rm -Rf workdir</pre>	일반 AWS

관련 리소스

- [메인프레임 데이터 유틸리티 README](#) () GitHub
- [EBCDIC 문자 세트](#) (IBM 설명서)
- [EBCDIC에서 ASCII로](#) (IBM 설명서)
- [COBOL](#) (IBM 설명서)
- [Amazon S3 트리거를 사용하여 Lambda 함수 간접 호출](#) (AWS Lambda 설명서)

Micro Focus를 사용하여 복잡한 레코드 레이아웃이 있는 메인프레임 데이터 파일 변환

작성자: Peter West

환경: 프로덕션	소스: Mainframe EBCDIC 데이터 파일	대상: Micro Focus ASCII 데이터 파일
R 유형: 리호스팅	워크로드: 기타 모든 워크로드	기술: 메인프레임, 현대화
AWS 서비스: AWS Mainframe Modernization		

요약

이 패턴은 Micro Focus 구조 파일을 사용하여 텍스트가 아닌 데이터와 복잡한 레코드 레이아웃을 포함하는 메인프레임 데이터 파일을 EBCDIC(확장 이진 코드 십진 교환 코드) 문자 인코딩에서 ASCII(정보 교환을 위한 미국 표준 코드) 문자 인코딩으로 변환하는 방법을 보여줍니다. 파일 변환을 완료하려면 다음을 수행해야 합니다.

1. 메인프레임 환경의 모든 데이터 항목과 레코드 레이아웃을 설명하는 단일 소스 파일을 준비합니다.
2. Micro Focus 클래식 데이터 파일 도구 또는 데이터 파일 도구의 일부로 Micro Focus 데이터 파일 편집기를 사용하여 데이터의 레코드 레이아웃이 포함된 구조 파일을 생성합니다. 구조 파일은 텍스트가 아닌 데이터를 식별하므로 메인프레임 파일을 EBCDIC에서 ASCII로 올바르게 변환할 수 있습니다.
3. 클래식 데이터 파일 도구 또는 데이터 파일 도구를 사용하여 구조 파일을 테스트합니다.

사전 조건 및 제한 사항

사전 조건

- 활성 상태의 AWS 계정
- Windows용 Micro Focus 엔터프라이즈 개발자, [AWS Mainframe Modernization](#)를 통해 사용 가능

제품 버전

- Micro Focus 엔터프라이즈 서버 7.0 이상

도구

- [Micro Focus 엔터프라이즈 개발자](#)는 엔터프라이즈 개발자의 모든 통합 개발 환경(IDE) 변형으로 만든 애플리케이션을 위한 실행 환경을 제공합니다.
- Micro Focus [클래식 데이터 파일 도구](#)를 사용하면 데이터 파일을 변환, 탐색, 편집 및 생성할 수 있습니다. 클래식 데이터 파일 도구에는 [데이터 파일 변환기](#), [레코드 레이아웃 편집기](#) 및 [데이터 파일 편집기](#)가 포함됩니다.
- Micro Focus [데이터 파일 도구](#)를 사용하면 데이터 파일을 만들고, 편집하고, 이동할 수 있습니다. 데이터 파일 도구에는 [데이터 파일 편집기](#), [파일 변환 유틸리티](#) 및 [데이터 파일 구조 명령줄 유틸리티](#)가 포함됩니다.

에픽

원본 파일 준비

작업	설명	필요한 기술
소스 구성 요소를 식별합니다.	<p>텍스트가 아닌 데이터를 포함하는 재정의의 포함하여 파일에 사용할 수 있는 모든 레코드 레이아웃을 식별합니다.</p> <p>재정의의 포함하는 레이아웃이 있는 경우 데이터 구조의 가능한 각 순열을 설명하는 고유한 레이아웃으로 이러한 레이아웃을 축소해야 합니다. 일반적으로 데이터 파일의 레코드 레이아웃은 다음 아키타입으로 설명할 수 있습니다.</p> <ul style="list-style-type: none"> • 텍스트 데이터만 포함된 레코드 레이아웃 	앱 개발자

작업	설명	필요한 기술
	<ul style="list-style-type: none"> • 텍스트가 아닌 데이터가 포함된 레코드 레이아웃 • REDEFINES 절에 종속된 텍스트가 아닌 데이터가 포함된 레코드 레이아웃 <p>복잡한 레코드 레이아웃이 포함된 파일의 평면화된 레코드 레이아웃을 생성하는 방법에 대한 자세한 내용은 메인프레임 마이그레이션을 위한 ASCII 환경의 EBCDIC 애플리케이션 리호스팅을 참조하세요.</p>	

작업	설명	필요한 기술
<p>레코드 레이아웃 조건을 식별합니다.</p>	<p>레코드 레이아웃이 여러 개 있는 파일 또는 REDEFINES 절이 있는 복잡한 레이아웃을 포함하는 파일의 경우 변환 중에 사용할 레이아웃을 정의하는데 사용할 수 있는 레코드 내 데이터 및 조건을 식별하세요. 이러한 파일을 처리하는 프로그램을 잘 아는 주제 전문가(SME)와 이 작업에 대해 논의하는 것이 좋습니다.</p> <p>예를 들어 파일에는 텍스트가 아닌 데이터를 포함하는 두 가지 레코드 유형이 포함될 수 있습니다. 소스를 검사하여 다음과 비슷한 코드를 찾을 수 있습니다.</p> <div data-bbox="594 1094 1027 1371" style="border: 1px solid #ccc; border-radius: 10px; padding: 10px; margin: 10px 0;"> <pre>MOVE "M" TO PART-TYPE MOVE "MAIN ASSEMBLY" TO PART-NAME MOVE "S" TO PART-TYPE MOVE "SUB ASSEMBLY 1" TO PART-NAME</pre> </div> <p>이 코드는 다음 사항을 식별하는데 도움이 됩니다.</p> <ul style="list-style-type: none"> • “PART-TYPE” 필드는 레코드 유형을 결정하는데 사용됩니다. • “M” 값은 “M-PART-RECORD”에 사용됩니다. • “S” 값은 “S-PART-RECORD”에 사용됩니다. 	<p>앱 개발자</p>

작업	설명	필요한 기술
	이 필드에서 사용하는 값을 문서화하여 레코드 레이아웃을 파일의 올바른 데이터 레코드와 연결할 수 있습니다.	
소스 파일을 빌드합니다.	<p>파일이 여러 소스 파일에 걸쳐 설명되거나 레코드 레이아웃에 REDEFINES 절에 종속된 텍스트가 아닌 데이터가 포함되어 있는 경우 레코드 레이아웃이 포함된 새 소스 파일을 생성하세요. 새 프로그램에서는 SELECT 및 FD 문을 사용하여 파일을 설명할 필요가 없습니다. 프로그램은 단순히 작업 스토리지 내에 레코드 설명을 01 레벨로 포함할 수 있습니다.</p> <p>참고: 각 데이터 파일의 소스 파일을 생성하거나 모든 데이터 파일을 설명하는 마스터 원본 파일을 생성할 수 있습니다.</p>	앱 개발자

작업	설명	필요한 기술
소스 파일을 컴파일합니다.	<p>소스 파일을 컴파일하여 데이터 사전을 빌드합니다. EBCDIC 문자 세트를 사용하여 소스 파일을 컴파일하는 것이 좋습니다. IBMCOMP 디렉티브 또는 ODOSLIDE 디렉티브를 사용하는 경우 소스 파일에서도 이러한 디렉티브를 사용해야 합니다.</p> <p>참고: IBMCOMP는 COMP 필드의 바이트 스토리지에 영향을 미치고 ODOSLIDE는 패딩에 영향을 미치며 구조는 다양합니다. 이러한 지시문을 잘못 설정하면 변환 도구가 데이터 레코드를 제대로 읽지 못합니다. 이로 인해 변환된 파일에 잘못된 데이터가 생성됩니다.</p>	앱 개발자

(옵션 A) 클래식 데이터 파일 도구를 사용하여 구조 파일 생성

작업	설명	필요한 기술
도구를 시작하고 사전을 로드합니다.	<ol style="list-style-type: none"> 1. Windows 시작 메뉴 아이콘을 선택하고 Micro Focus 엔터프라이즈 개발자를 검색하여 선택한 다음 클래식 데이터 파일 도구를 선택합니다. 2. 파일을 선택한 다음 레코드 레이아웃을 선택합니다. 3. 레이아웃을 구성할 파일 선택 대화 상자에서 파일 이 	앱 개발자

작업	설명	필요한 기술
	<p>름에 대해 이전에 소스 파일을 컴파일할 때 생성한 IDY(.idy) 파일을 선택합니다. 그런 다음 열기를 선택합니다.</p> <p>4. 클래식 데이터 파일 도구가 EBCDIC를 사용하고 있는지 확인하려면 데이터 파일 도구 대화 상자에서 IDY 파일이 EBCDIC로 설정되고 데이터 도구가 ANSI로 설정된 경우 예를 선택하세요.</p>	
<p>기본 레코드 레이아웃을 생성합니다.</p>	<p>조건부 레이아웃과 일치하지 않는 모든 레코드에 대해 기본 레코드 레이아웃을 사용합니다.</p> <ol style="list-style-type: none"> 레이아웃 창에서 데이터 구조를 확장한 다음 기본 레이아웃에 사용되는 01 레벨을 찾습니다. 01 항목을 마우스 오른쪽 버튼으로 클릭하고 새 레이아웃을 선택합니다. 새 레코드 레이아웃 마법사 대화 상자에서 기본 레이아웃을 선택한 후 다음을 선택합니다. 마침을 클릭합니다. <p>기본 레이아웃은 레이아웃 패널에 나타나며 빨간색 폴더 아이콘으로 식별할 수 있습니다.</p>	<p>앱 개발자</p>

작업	설명	필요한 기술
조건부 레코드 레이아웃을 생성합니다.	<p>파일에 한 개를 초과하는 레코드 레이아웃이 있는 경우 조건부 레코드 레이아웃을 사용합니다.</p> <ol style="list-style-type: none"> 레이아웃 창에서 데이터 구조를 확장한 다음 조건부 레이아웃에 사용되는 01 수준을 찾습니다. 01 항목을 마우스 오른쪽 버튼으로 클릭하고 새 레이아웃을 선택합니다. 새 레코드 레이아웃 마법사 대화 상자에서 조건부 레이아웃을 선택한 후 다음을 선택합니다. 마침을 클릭합니다. 조건부 레이아웃은 레이아웃 패널에 나타나며 노란색 폴더 아이콘으로 식별할 수 있습니다. 조건부 레이아웃을 확장하고 조건을 입력해야 하는 필드를 마우스 오른쪽 단추로 클릭한 다음 속성을 선택합니다. 필드 속성 대화 상자에 조건을 입력합니다. 문자 집합이 EBCDIC로 설정되었는지 확인한 다음 확인을 선택합니다. 조건이 설정된 필드 옆에 체크마크가 나타납니다. 	앱 개발자

작업	설명	필요한 기술
	<ol style="list-style-type: none"> 7. 이 레이아웃에 조건이 필요한 다른 필드에 대해 5~6단계를 반복합니다. 8. 추가해야 하는 다른 모든 조건부 레이아웃에 대해 1~6 단계를 반복합니다. 9. 파일을 선택하고 다른 이름으로 저장을 선택한 다음 구조 파일을 디스크에 저장합니다. 	

(옵션 B) 데이터 파일 도구를 사용하여 구조 파일 생성

작업	설명	필요한 기술
도구를 시작하고 사전을 로드합니다.	<ol style="list-style-type: none"> 1. Windows 시작 메뉴 아이콘을 선택하고 Micro Focus 엔터프라이즈 개발자를 검색하여 선택한 다음 데이터 파일 도구를 선택합니다. 2. 파일, 새로 만들기, 구조 파일을 선택합니다. 3. 열기 대화 상자의 파일 이름에서 이전에 소스 파일을 컴파일할 때 생성한 IDY(.idy) 파일을 선택합니다. 그런 다음 열기를 선택합니다. 4. 데이터 파일 도구가 EBCDIC를 사용하고 있는지 확인하려면 디버그 파일 섹션의 드롭다운 메뉴가 	앱 개발자

작업	설명	필요한 기술
	EBCDIC로 설정되어 있는지 확인하세요.	
기본 레코드 레이아웃을 생성합니다.	<p>조건부 레이아웃과 일치하지 않는 모든 레코드에 대해 기본 레코드 레이아웃을 사용합니다.</p> <ol style="list-style-type: none"> 1. 왼쪽 창의 사용 가능한 레이아웃 섹션에서 데이터 구조를 확장한 다음 기본 레이아웃에 사용되는 01 수준을 찾습니다. 2. 01 항목을 마우스 오른쪽 버튼으로 클릭하고 기본 레이아웃 생성을 선택합니다. <p>기본 레이아웃은 레이아웃 패널에 나타나며 파란색 “D” 아이콘으로 식별할 수 있습니다.</p>	앱 개발자

작업	설명	필요한 기술
조건부 레코드 레이아웃을 생성합니다.	<p>파일에 한 개를 초과하는 레코드 레이아웃이 있는 경우 조건부 레코드 레이아웃을 사용합니다.</p> <ol style="list-style-type: none"> 오른쪽 창의 선택된 레이아웃 섹션에서 데이터 구조를 확장한 다음 조건부 레이아웃에 사용되는 01 수준을 찾습니다. 01 항목을 마우스 오른쪽 버튼으로 클릭하고 조건부 레이아웃 생성을 선택합니다. 조건부 레이아웃은 오른쪽의 레이아웃 패널에 나타나며 녹색 "C" 아이콘으로 식별할 수 있습니다. 조건부 레이아웃을 확장하고 조건을 입력해야 하는 필드를 마우스 오른쪽 단추로 클릭한 다음 속성을 선택합니다. 필드 속성 대화 상자에 조건을 입력합니다. 문자 집합이 EBCDIC로 설정되었는지 확인한 다음 확인을 선택합니다. 조건이 설정된 필드 옆에 빨간색 "IF" 아이콘이 나타납니다. 이 레이아웃에 조건이 필요한 다른 필드에 대해 3~4단계를 반복합니다. 	앱 개발자

작업	설명	필요한 기술
	<ol style="list-style-type: none"> 추가해야 하는 다른 모든 조 건부 레이아웃에 대해 1~4 단계를 반복합니다. 파일을 선택하고 다른 이름 으로 저장을 선택한 다음 구조 파일을 디스크에 저장합니다. 	

(옵션 A) 클래식 데이터 파일 도구를 사용하여 구조 파일 테스트

작업	설명	필요한 기술
EBCDIC 데이터 파일을 테스트 합니다.	<p>구조 파일을 사용하여 EBCDIC 테스트 데이터 파일을 제대로 볼 수 있는지 확인합니다.</p> <ol style="list-style-type: none"> Windows 시작 메뉴 아이콘 을 선택하고 Micro Focus 엔 터프라이즈 개발자를 찾아 선택한 다음 클래식 데이터 도구를 선택합니다. 파일을 선택한 다음 열기를 선택합니다. 열기 대화 상자의 파일 이름에서 EBCDIC 데이터 세 트를 선택한 다음 열기를 선택합니다. 파일, 데이터 파일 편집기, 레코드 레이아웃 로드를 선택합니다. 열기 대화 상자의 파일 이름에서 구조 파일을 선택한 다음 열기를 선택합니다. 	앱 개발자

작업	설명	필요한 기술
	<p>6. 문자 세트 모드가 EBCDIC 로 설정되었는지 확인하려면 드롭다운 메뉴가 EBCDIC로 설정되어 있는지 확인하세요. 왼쪽 창에서 원시 레코드 데이터를, 오른쪽 창에서 형식이 지정된 데이터를 볼 수 있습니다.</p> <p>7. 다양한 레코드를 선택하여 모든 형식이 올바른 레이아웃으로 렌더링되도록 합니다.</p>	

(옵션 B) 데이터 파일 도구를 사용하여 구조 파일 테스트

작업	설명	필요한 기술
<p>EBCDIC 데이터 파일을 테스트합니다.</p>	<p>구조 파일을 사용하여 EBCDIC 테스트 데이터 파일을 제대로 볼 수 있는지 확인합니다.</p> <ol style="list-style-type: none"> 1. Windows 시작 메뉴 아이콘을 선택하고 Micro Focus 엔터프라이즈 개발자를 찾아 선택한 다음 데이터 파일 도구를 선택합니다. 2. 파일, 열기, 데이터 파일을 선택합니다. 3. 데이터 파일 열기 대화 상자의 로컬 탭에서 파일 이름으로 찾아보기를 선택하여 EBCDIC 테스트 파일의 위치를 찾습니다. 	<p>앱 개발자</p>

작업	설명	필요한 기술
	<ol style="list-style-type: none"> 4. 구조 파일(선택 사항)의 경우 찾아보기를 선택하여 구조 파일의 위치를 찾습니다. 5. 파일 세부 정보 섹션에서 파일의 세부 정보를 입력하고 인코딩이 EBCDIC로 설정되어 있는지 확인합니다. 6. 요구 사항에 따라 공유 열기 또는 독점 열기 모드를 선택합니다. 7. 도구 모음의 모양 섹션에 있는 드롭다운 메뉴가 EBCDIC로 설정되어 있는지 확인합니다. 왼쪽 창에 원시 레코드 데이터가 표시되고 오른쪽 창에 형식이 지정된 데이터가 표시됩니다. 8. 다양한 레코드를 선택하여 모든 형식이 올바른 레이아웃으로 렌더링되도록 합니다. 	

테스트 데이터 파일 변환

작업	설명	필요한 기술
<p>EBCDIC 파일의 변환을 테스트합니다..</p>	<ol style="list-style-type: none"> 1. Windows 시작 메뉴 아이콘을 선택하고 Micro Focus 엔터프라이즈 개발자를 찾아 선택한 다음 클래식 데이터 도구를 선택합니다. 2. 도구를 선택한 다음 전환을 선택합니다. 	<p>앱 개발자</p>

작업	설명	필요한 기술
	<ol style="list-style-type: none"> <li data-bbox="591 212 1027 533">3. 데이터 파일 변환 대화 상자의 입력 파일 섹션에서 파일 이름에 대해 찾아보기를 선택하여 EBCDIC 입력 파일을 찾아 선택합니다. 문자 세트가 EBCDIC으로 설정되어 있는지 확인합니다. <li data-bbox="591 554 1027 919">4. 문자 세트 변환 섹션에서 문자 세트 변환 및 레코드에 텍스트가 아닌 데이터 항목이 포함됨 확인란을 선택합니다. 변환할 레이아웃 선택을 선택한 다음 찾아보기를 선택하여 구조 파일을 찾아 선택합니다. <li data-bbox="591 940 1027 1306">5. 새 파일 섹션의 파일 이름에 생성하려는 ASCII 출력 파일의 경로와 파일 이름을 입력합니다. 기본적으로 전환 도구는 입력 파일과 동일한 형식을 사용합니다. 테스트하려면 옵션을 기본값으로 둡니다. <li data-bbox="591 1327 902 1367">6. 전환을 선택합니다. <li data-bbox="591 1388 1027 1753">7. (옵션 A)클래식 데이터 파일 도구를 사용하여 구조 파일 테스트 또는 (옵션 B)데이터 파일 도구를 사용하여 구조 파일 테스트 섹션의 단계를 따르되 EBCDIC 파일 대신 ASCII 출력 파일을 로드하세요. <li data-bbox="591 1774 1027 1864">8. EBCDIC 및 ASCII 파일을 모두 데이터 파일 편집기에 로 	

작업	설명	필요한 기술
	드한 다음 파일을 나란히 비교하여 변환의 정확성을 확인합니다.	

관련 리소스

- [Micro Focus](#)(Micro Focus 설명서)
- [메인프레임 및 레거시 코드](#)(ASW 블로그 게시물)
- [AWS 권장 가이드](#)(AWS 설명서)
- [AWS 설명서](#)(AWS 설명서)
- [AWS 일반 참조](#)(AWS 설명서)
- [AWS 용어집](#)(AWS 설명서)

Terraform을 사용하여 컨테이너화된 Blu Age 애플리케이션을 위한 환경 배포

작성자: Richard Milner-Watts(AWS)

코드 리포지토리: 블루 에이지 샘플 ECS 인프라 (테라폼)	환경: 프로덕션	소스: 메인프레임
대상: 컨테이너	R 유형: 리플랫폼	워크로드: IBM, 모든 기타 워크로드
기술: 메인프레임, 컨테이너 및 마이크로서비스	AWS 서비스: Amazon ECS, Step Functions, Amazon VPC, Amazon Aurora	

요약

레거시 메인프레임 워크로드를 최신 클라우드 아키텍처로 마이그레이션하면 메인프레임 유지 관리 비용을 없앨 수 있습니다. 이 비용은 환경이 노후화됨에 따라 증가합니다. 하지만 메인프레임에서 작업을 마이그레이션하는 데에는 고유한 문제가 발생할 수 있습니다. 내부 리소스는 작업 로직에 익숙하지 않을 수 있으며, 이러한 특수 작업을 수행하는 메인프레임의 고성능은 보편적인 상용 CPU와 비교할 때 복제하기 어려울 수 있습니다. 이러한 작업을 다시 작성하는 것은 큰 작업이 될 수 있으며 상당한 노력이 필요할 수 있습니다.

Blu Age는 레거시 메인프레임 워크로드를 컨테이너로 실행할 수 있는 최신 Java 코드로 변환합니다.

이 패턴은 Blu Age 도구로 현대화된 컨테이너식 애플리케이션을 실행하기 위한 샘플 서버리스 아키텍처를 제공합니다. 포함된 HashiCorp Terraform 파일은 Blu Age 컨테이너의 오케스트레이션을 위한 안전한 아키텍처를 구축하여 배치 작업과 실시간 서비스를 모두 지원합니다.

Blu Age 및 AWS 서비스를 사용하여 워크로드를 현대화하는 방법에 대한 자세한 내용은 다음 AWS 권장 가이드 간행물을 참조하십시오.

- [서버리스 인프라에서 Blu Age로 현대화된 메인프레임 워크로드 실행](#)
- [Blu Age로 현대화된 메인프레임 워크로드 컨테이너화](#)

Blu Age를 사용하여 메인프레임 워크로드를 현대화하는 데 도움이 필요하다면 [Blu Age 웹 사이트](#)에서 전문가에게 문의하기를 선택하여 Blu Age 팀에 문의하십시오. 현대화된 워크로드를 AWS로 마이그레이션하고, 이를 AWS 서비스와 통합하고, 프로덕션 환경으로 이전하는 데 도움이 필요하다면 AWS 계정 관리자에게 문의하거나 [AWS Professional Services 양식](#)을 작성하십시오.

사전 조건 및 제한 사항

사전 조건

- [Blu Age 패턴으로 현대화된 Containerize 메인프레임 워크로드](#)에서 제공하는 컨테이너화된 Blu Age 애플리케이션 샘플. 샘플 애플리케이션은 현대화된 애플리케이션의 입력 및 출력 처리를 다루기 위한 로직을 제공하며 이 아키텍처와 통합될 수 있습니다.
- 이러한 리소스를 배포하려면 Terraform이 필요합니다.

제한 사항

- Amazon Elastic Container Service(Amazon ECS)는 컨테이너에 제공할 수 있는 작업 리소스에 제한을 둡니다. 이러한 리소스에는 CPU, RAM, 스토리지가 포함됩니다. 예를 들어 AWS Fargate와 함께 Amazon ECS를 사용하는 경우 [작업 리소스 제한이 적용됩니다](#).

제품 버전

이 솔루션은 다음 버전으로 테스트되었습니다.

- Terraform 1.3.6
- Terraform AWS Provider 4.46.0

아키텍처

소스 기술 스택

- Blu Age
- Terraform

대상 기술 스택

- Amazon Aurora PostgreSQL 호환 에디션

- AWS Backup
- Amazon Elastic Container Registry (Amazon ECR)
- Amazon ECS
- Identity and Access Management Service(IAM)
- Key Management Service(KMS)
- AWS Secrets Manager
- Amazon Simple Notification Service(SNS)
- Amazon Simple Storage Service(S3)
- AWS Step Functions
- AWS Systems Manager

대상 아키텍처

다음 다이어그램은 솔루션 아키텍처를 보여 줍니다.

1. 솔루션은 다음의 IAM 역할을 배포합니다.

- Batch 태스크 역할
- 배치 태스크 실행 역할
- 서비스 태스크 역할
- 서비스 태스크 실행 역할
- Step Function 역할
- AWS Backup 역할
- RDS Enhanced Monitoring 역할

역할은 최소 권한 액세스 원칙을 준수합니다.

2. Amazon ECR은 이 패턴으로 오케스트레이션된 컨테이너 이미지를 저장하는 데 사용됩니다.
3. AWS Systems Manager Parameter Store는 런타임 시 각 환경에 대한 구성 데이터를 Amazon ECS 탬플릿, 커널, 정의에 제공합니다.
4. AWS Systems Manager Parameter Store는 런타임 시 각 환경에 대한 민감한 구성 데이터를 Amazon ECS 태스크 정의에 제공합니다. 데이터는 KMS에 의해 암호화되었습니다.
5. Terraform 모듈은 모든 실시간 및 배치 작업에 대한 Amazon ECS 작업 정의를 생성합니다.

6. Amazon ECS는 Fargate를 컴퓨팅 엔진으로 사용하여 배치 작업을 실행합니다. 이 작업은 Step Functions에서 필요에 따라 시작하는 단기 작업입니다.
7. Amazon Aurora PostgreSQL-Compatible은 현대화된 애플리케이션을 지원하기 위한 데이터베이스를 제공합니다. 이는 IBM Db2 또는 IBM IMS DB와 같은 메인프레임 데이터베이스를 대체합니다.
8. Amazon ECS는 수명이 긴 서비스를 실행하여 현대화된 실시간 워크로드를 제공합니다. 이러한 상태 비저장 애플리케이션은 가용 영역 전반적으로 분산된 컨테이너와 함께 영구적으로 실행됩니다.
9. Network Load Balancer는 실시간 워크로드에 대한 액세스 권한을 부여하는 데 사용됩니다. Network Load Balancer는 IBM CICS와 같은 이전 프로토콜을 지원합니다. 또는 Application Load Balancer를 HTTP 기반 워크로드와 함께 사용할 수 있습니다.
- 10 Amazon S3는 작업 입력 및 출력을 위한 객체 스토리지를 제공합니다. 컨테이너는 Amazon S3로의 풀 및 푸시 작업을 처리하여 Blu Age 애플리케이션을 위한 작동 가능한 디렉터리를 준비해야 합니다.
- 11 Step Functions 서비스는 배치 워크로드를 처리하기 위한 Amazon ECS 작업 실행을 오케스트레이션하는 데 사용됩니다.
- 12 각 배치 워크로드의 SNS 주제는 현대화된 애플리케이션을 이메일과 같은 다른 시스템과 통합하거나, Amazon S3에서 FTP로 출력 객체를 전송하는 등의 추가 작업을 시작하는 데 사용됩니다.

참고: 기본적으로 솔루션은 인터넷에 액세스할 수 없습니다. 이 패턴은 Virtual Private Cloud(VPC)가 [Transit Gateway](#)와 같은 서비스를 사용하여 다른 네트워크에 연결된다고 가정합니다. 따라서 솔루션에서 사용하는 AWS 서비스에 대한 액세스 권한을 부여하기 위해 여러 인터페이스 VPC 엔드포인트가 배포됩니다. 직접 인터넷 액세스를 활성화하기 위해 Terraform 모듈의 토글을 사용하여 VPC 엔드포인트를 인터넷 게이트웨이 및 관련 리소스로 바꿀 수 있습니다.

자동화 및 규모 조정

이 패턴 전반적으로 서버리스 리소스를 사용하면 스케일 아웃을 통해 이러한 설계의 규모에 제한이 거의 없도록 할 수 있습니다. 따라서 기존 메인프레임에서 발생할 수 있는 컴퓨팅 리소스에 대한 경쟁 등 잡음이 많은 이웃 문제를 줄일 수 있습니다. 필요에 따라 배치 작업이 동시에 실행되도록 예약할 수 있습니다.

개별 컨테이너는 Fargate에서 지원하는 최대 크기로 제한됩니다. 자세한 내용은 Amazon ECS 설명서에서 [작업 CPU 및 메모리](#) 섹션을 참조하십시오.

컨테이너를 추가하면 [실시간 워크로드를 수평적으로 확장할 수 있습니다](#).

도구

서비스

- [Amazon Aurora PostgreSQL 호환 에디션](#)은 PostgreSQL 배포를 설정, 운영 및 확장할 수 있는 완전 관리형 ACID 준수의 관계형 데이터베이스 엔진입니다.
- [AWS Backup](#)은 서비스, 클라우드 및 온프레미스에서 데이터 보호를 중앙 집중화하고 자동화하는 데 도움이 되는 완전관리형 서비스입니다.
- [Amazon Elastic Container Registry\(Amazon ECR\)](#)는 안전하고 확장 가능하고 신뢰할 수 있는 관리형 컨테이너 이미지 레지스트리 서비스입니다.
- [Amazon Elastic Container Service\(Amazon ECS\)](#)는 클러스터에서 컨테이너를 실행, 중지 및 관리하는 데 도움이 되는 빠르고 확장 가능한 컨테이너 관리 서비스입니다.
- [Identity and Access Management\(IAM\)](#)는 사용자에게 대한 인증 및 권한 부여를 제어함으로써 AWS 리소스에 대한 액세스를 안전하게 제어할 수 있습니다.
- [AWS Key Management Service\(AWS KMS\)](#)를 사용하면 암호화 키를 생성하고 제어하여 데이터를 보호할 수 있습니다.
- [AWS Secrets Manager](#)를 사용하면 암호를 포함하여 코드에 하드코딩된 보안 인증을 Secrets Manager에 대한 API 호출로 대체하여 프로그래밍 방식으로 암호를 검색할 수 있습니다.
- [Amazon Simple Notification Service\(Amazon SNS\)](#)를 사용하면 웹 서버 및 이메일 주소를 포함하여 게시자와 클라이언트 간의 메시지 교환을 조정하고 관리할 수 있습니다.
- [Amazon Simple Storage Service\(S3\)](#)는 원하는 양의 데이터를 저장, 보호 및 검색하는 데 도움이 되는 클라우드 기반 객체 스토리지 서비스입니다.
- [AWS Step Functions](#)는 Lambda 함수와 기타 AWS 서비스를 결합할 수 있는 서버리스 오케스트레이션 서비스로서 비즈니스 크리티컬 애플리케이션을 구축합니다.
- [AWS Systems Manager Parameter Store](#)는 구성 데이터 관리 및 암호 관리를 위한 안전한 계층적 스토리지를 제공합니다.

기타 서비스

- [HashiCorp Terraform](#)은 코드를 사용하여 클라우드 인프라 및 리소스를 프로비저닝하고 관리하는 데 도움이 되는 오픈 소스 IaC (코드형 인프라) 도구입니다. 이 패턴은 Terraform을 사용하여 샘플 아키텍처를 생성합니다.

코드 리포지토리

이 패턴의 소스 코드는 GitHub [블루에이지 샘플 ECS](#) 인프라 (Terraform) 리포지토리에서 확인할 수 있습니다.

모범 사례

- 테스트 환경의 경우 forceDate 옵션과 같은 기능을 사용하여, 알려진 기간 동안 항상 실행함으로써 일관된 테스트 결과를 생성하도록 현대식 애플리케이션을 구성합니다.
- 최적의 리소스 양을 사용하도록 각 작업을 개별적으로 조정합니다. [Amazon CloudWatch Container Insights](#)를 사용하여 잠재적 병목 현상에 대한 지침을 얻을 수 있습니다.

에픽

배포를 위한 환경 준비

작업	설명	필요한 기술
솔루션 소스 코드를 복제합니다.	프로젝트에서 솔루션 코드를 복제하십시오. GitHub	DevOps 엔지니어
Terraform 상태를 저장할 리소스를 배포하여 환경을 부트스트랩합니다.	<ol style="list-style-type: none"> 1. 터미널 창을 열어 Terraform 이 설치되어 있고 AWS 자격 증명을 사용할 수 있는지 확인합니다. 2. bootstrap-terraform 폴더로 이동합니다. 3. main.tf 파일을 편집하면, 원할 경우 S3 버킷 (<accountId>-terraform-backend) 및 Amazon DynamoDB 테이블 (terraform-lock)의 이름을 변경할 수 있습니다. 4. terraform apply 명령을 실행하여 리소스를 배포합니다. S3 버킷과 DynamoDB 	DevOps 엔지니어

작업	설명	필요한 기술
	테이블 이름을 기록해 둡니다.	

솔루션 인프라 배포

작업	설명	필요한 기술
Terraform 구성을 검토하고 업데이트합니다.	<p>루트 디렉터리에서 <code>main.tf</code>, 파일을 열고 내용을 검토한 후 다음과 같은 업데이트 실시를 고려합니다.</p> <ol style="list-style-type: none"> 1. <code>eu-west-1</code> 문자열을 검색하여 사용하고자 하는 리전으로 교체함으로써 AWS 리전을 업데이트합니다. 2. 이전 에픽에서 기본값이 변경된 경우 Terraform Backend 블록에서 버킷 이름을 업데이트합니다. 3. 이전 에픽에서 기본값이 변경된 경우 <code>dynamodb_table</code> 값을 업데이트합니다. 4. <code>stack_prefix</code> 변수 값을 원하는 문자열로 업데이트합니다. 이 문자열은 이 패턴으로 생성한 모든 리소스의 이름 앞에 붙습니다. 5. <code>vpc_cidr</code> 값을 업데이트합니다. 이는 최소한 <code>/24</code> 주소 범위여야 합니다. 	DevOps 엔지니어

작업	설명	필요한 기술
	<p>6. <code>Locals</code> 섹션을 검토합니다. 이는 배포될 <code>Blu Age</code> 작업을 정의하는 데 사용됩니다. 솔루션은 <code>bluage_batch_modules</code> 목록 객체를 반복하여 목록의 각 요소에 대한 관련 리소스(Step Functions 상태 시스템, 작업 정의, SNS 주제)를 생성합니다. 경우에 따라 다양한 환경에 맞게 변수를 조정하고자 할 수 있습니다. 예를 들어 테스트 환경에서 런타임을 강제 실행하기 위해 <code>force_execution_time</code> 변수 값을 변경할 수 있습니다.</p> <p>7. 인터넷 액세스를 활성화하려면 <code>direct_internet_access_required</code>에 대한 값을 <code>false</code>에서 <code>true(으)</code>로 변경합니다. 그러면 인프라의 공용 인터넷 액세스를 활성화하는 NAT 게이트웨이 및 라우팅 테이블과 함께 인터넷 게이트웨이가 배포됩니다. 기본적으로 솔루션은 인터넷에 직접 액세스하지 않고도 인터페이스 VPC 엔드포인트를 VPC에 배포합니다.</p> <p>8. Elastic Load Balancing을 통해 제공되는 모든 클라이</p>	

작업	설명	필요한 기술
	엔트-서버 워크로드에 대한 액세스 권한을 부여하려면 허용되어야 하는 CIDR 네트워크로 <code>additional_nlb_igress_cidrs</code> 의 값을 업데이트합니다.	
Terraform 파일을 배포합니다.	터미널에서, <code>terraform apply</code> 명령을 실행하여 모든 리소스를 배포합니다. Terraform에서 생성된 변경 사항을 검토하고 예를 입력하여 빌드를 시작합니다. 이 인프라를 배포하는 데 15분 이상 걸릴 수 있다는 점에 유의하십시오.	DevOps 엔지니어

(선택 사항) 유효한 Blu Age 컨테이너화된 애플리케이션을 배포

작업	설명	필요한 기술
Blu Age 컨테이너 이미지를 Amazon ECR로 푸시합니다.	이전 에픽에서 생성한 Amazon ECR 리포지토리에 컨테이너를 푸시합니다. 자세한 지침은 Amazon ECR 설명서 를 참조하십시오. 컨테이너 이미지 URI를 기록해 둡니다.	DevOps 엔지니어
Blu Age 컨테이너 이미지를 참조하도록 Terraform을 업데이트합니다.	업로드한 컨테이너 이미지를 참조하도록 <code>main.tf</code> 파일을 업데이트합니다.	DevOps 엔지니어

작업	설명	필요한 기술
Terraform 파일을 다시 배포합니다.	터미널에서 terraform apply(을)를 실행하여 모든 리소스를 배포합니다. Terraform에서 제안된 업데이트를 검토한 다음 예를 입력하여 배포를 진행합니다.	DevOps 엔지니어

관련 리소스

- [Blu Age](#)
- [서버리스 인프라에서 Blu Age로 현대화된 메인프레임 워크로드 실행](#)
- [Blu Age로 현대화된 메인프레임 워크로드 컨테이너화](#)

Stonebranch 유니버설 컨트롤러를 AWS Mainframe Modernization 과 통합

작성자: Vaidy Sankaran(AWS), Robert Lemieux(Stonebranch), Huseyin Gomleksizoglu(Stonebranch), Pablo Alonso Prieto(AWS)

코드 리포지토리: aws-mainframe-modernization-stonebranch - 통합	환경: PoC 또는 파일럿	기술: 메인프레임, 현대화, 운영 DevOps, SaaS
워크로드: 오픈 소스, Microsoft	AWS 서비스: AWS Mainframe Modernization, Amazon RDS, Amazon S3	

요약

이 패턴은 [Stonebranch 유니버설 오토메이션 센터\(UAC\) 워크로드 오케스트레이션](#)을 [Amazon Web Services\(AWS\) Mainframe Modernization 서비스](#)와 통합하는 방법을 설명합니다. AWS Mainframe Modernization 서비스는 메인프레임 애플리케이션을 AWS 클라우드로 마이그레이션하고 현대화합니다. Micro Focus 엔터프라이즈 기술을 사용한 [AWS Mainframe Modernization 리플랫폼](#)과 AWS Blu Age를 사용한 [AWS Mainframe Modernization 자동 리팩터링](#)이라는 두 가지 패턴을 제공합니다.

Stonebranch UAC는 실시간 IT 자동화 및 오케스트레이션 플랫폼입니다. UAC는 온프레미스에서 AWS에 이르는 하이브리드 IT 시스템 전반에서 작업, 활동 및 워크플로를 자동화하고 오케스트레이션 하도록 설계되었습니다. 메인프레임 시스템을 사용하는 엔터프라이즈 고객은 클라우드 중심의 현대화된 인프라 및 애플리케이션으로 전환하고 있습니다. Stonebranch의 도구 및 전문 서비스는 기존 스케줄러 및 자동화 기능을 AWS 클라우드로 쉽게 마이그레이션할 수 있습니다.

AWS Mainframe Modernization 서비스를 사용하여 메인프레임 프로그램을 AWS 클라우드로 마이그레이션하거나 현대화할 때 이 통합을 사용하여 배치 일정을 자동화하고, 민첩성을 높이고, 유지 관리를 개선하고, 비용을 절감할 수 있습니다.

이 패턴은 [Stonebranch 스케줄러](#)를 [AWS Mainframe Modernization 서비스인 Micro Focus Enterprise 런타임](#)으로 마이그레이션된 메인프레임 애플리케이션과 통합하기 위한 지침을 제공합니다. 이 패턴은 솔루션스 아키텍트, 개발자, 컨설턴트, 마이그레이션 전문가 및 마이그레이션, 현대화, 운영 등에 종사하는 사람들을 위한 것입니다. DevOps

목표 결과

이 패턴은 다음과 같은 목표 결과를 제공하는 데 중점을 둡니다.

- [Stonebranch 유니버설 컨트롤러](#)의 [AWS Mainframe Modernization 서비스\(Microfocus 런타임\)](#)에서 실행되는 메인프레임 배치 작업을 예약, 자동화 및 실행할 수 있습니다.
- Stonebranch 유니버설 컨트롤러에서 애플리케이션의 배치 프로세스를 모니터링합니다.
- Stonebranch 유니버설 컨트롤러에서 배치 프로세스를 자동 또는 수동으로 시작/재시작/재실행/중지할 수 있습니다.
- AWS Mainframe Modernization 배치 프로세스의 결과를 검색합니다.
- Stonebranch 유니버설 컨트롤러에서 배치 작업의 [AWS CloudWatch](#) 로그를 캡처합니다.

사전 조건 및 제한 사항

사전 조건

- 활성 상태의 AWS 계정
- 작업 제어 언어(JCL) 파일이 포함된 Micro Focus [Bankdemo](#) 애플리케이션과 [AWS Mainframe Modernization 서비스\(Micro Focus 런타임\)](#) 환경에 배포된 배치 프로세스
- Micro Focus [Enterprise Server](#)에서 실행되는 메인프레임 애플리케이션을 구축하고 배포하는 방법에 대한 기본 지식
- [Stonebranch 유니버설 컨트롤러](#)에 대한 기본 지식
- Stonebranch 체험판 라이선스([Stonebranch](#)에 문의)
- 최소 코어 4개, 메모리 8GB, 디스크 공간 2GB를 갖춘 Windows 또는 Linux Amazon Elastic Compute Cloud(Amazon EC2) 인스턴스(예: xlarge)
- Apache Tomcat 버전 8.5.x 또는 9.0.x
- Oracle Java 런타임 환경(JRE) 또는 OpenJDK 버전 8 또는 11
- [Amazon Aurora MySQL-Compatible Edition](#)
- 리포지토리를 내보내기 위한 [Amazon Simple Storage Service\(S3\)](#) 버킷
- 고가용성(HA)에 대한 에이전트 Stonebranch 유니버설 메시지 서비스(OMS) 연결을 위한 [Amazon Elastic File System\(Amazon EFS\)](#)
- Stonebranch 유니버설 컨트롤러 7.2 유니버설 에이전트 7.2 설치 파일
- AWS Mainframe Modernization [작업 일정 예약 템플릿](#)(.zip 파일의 최신 릴리스 버전)

제한 사항

- 제품 및 솔루션은 OpenJDK 8 및 11에서만 테스트되고 호환성이 검증되었습니다.
- [aws-mainframe-modernization-stonebranch-integration](#) 작업 일정 템플릿은 AWS 메인프레임 현대화 서비스에서만 작동합니다.
- 이 작업 일정 템플릿은 Stonebranch 에이전트의 Unix, Linux 또는 Windows 버전에서만 작동합니다.

아키텍처

대상 상태 아키텍처

다음 다이어그램은 이 파일럿에 필요한 AWS 환경의 예를 보여줍니다.

1. Stonebranch 유니버설 오토메이션 센터(UAC)에는 유니버설 컨트롤러와 유니버설 에이전트라는 두 가지 주요 구성 요소가 있습니다. Stonebranch OMS는 컨트롤러와 개별 에이전트 간의 메시지 버스로 사용됩니다.
2. 유니버설 컨트롤러는 Stonebranch UAC 데이터베이스를 사용합니다. MySQL, Microsoft SQL, Oracle 또는 Aurora MySQL과 호환되는 데이터베이스일 수 있습니다.
3. [AWS 메인프레임 현대화 서비스 — 애플리케이션이 배포된 Micro Focus 런타임 환경. BankDemo](#) BankDemo 애플리케이션 파일은 S3 버킷에 저장됩니다. 이 버킷에는 메인프레임 JCL 파일도 포함되어 있습니다.
4. Stonebranch UAC는 배치 실행을 위해 다음과 같은 함수를 실행할 수 있습니다.
 - a. AWS Mainframe Modernization 서비스에 연결된 S3 버킷에 있는 JCL 파일 이름을 사용하여 배치 작업을 시작합니다.
 - b. 배치 작업 실행 상태를 확인할 수 있습니다.
 - c. 배치 작업 실행이 완료될 때까지 기다리세요.
 - d. 배치 작업 실행 로그를 가져옵니다.
 - e. 실패한 일괄 작업을 다시 실행합니다.
 - f. 작업이 실행되는 동안 배치 작업을 취소합니다.
5. Stonebranch UAC는 애플리케이션에 대해 다음 기능을 실행할 수 있습니다.
 - a. 애플리케이션 시작
 - b. 애플리케이션 상태 가져오기
 - c. 애플리케이션이 시작되거나 중지될 때까지 기다리세요.

- d. 애플리케이션 중지
- e. 애플리케이션 작업 로그 가져오기

Stonebranch 작업 전환

다음 다이어그램은 현대화 여정 중 Stonebranch의 직무 전환 프로세스를 나타냅니다. 작업 일정과 작업 정의를 AWS Mainframe Modernization 배치 작업을 실행할 수 있는 호환 가능한 형식으로 변환하는 방법을 설명합니다.

1. 변환 프로세스의 경우 기존 메인프레임 시스템에서 작업 정의를 내보냅니다.
2. JCL 파일은 메인프레임 현대화 애플리케이션용 S3 버킷에 업로드하여 AWS Mainframe Modernization 서비스에서 이러한 JCL 파일을 배포할 수 있습니다.
3. 변환 도구는 내보낸 작업 정의를 UAC 작업으로 변환합니다.
4. 모든 작업 정의와 작업 일정이 생성되면 이러한 개체를 유니버설 컨트롤러로 가져옵니다. 그러면 변환된 작업은 메인프레임에서 실행하는 대신 AWS Mainframe Modernization 서비스에서 프로세스를 실행합니다.

Stonebranch UAC 아키텍처

다음 아키텍처 다이어그램은 고가용성 (HA) 범용 컨트롤러 active-active-passive 모델을 나타냅니다. Stonebranch UAC는 여러 가용 영역에 배포되어 고가용성을 제공하고 재해 복구(DR)를 지원합니다.

유니버설 컨트롤러

두 개의 Linux 서버가 유니버설 컨트롤러로 프로비저닝됩니다. 둘 다 동일한 데이터베이스 엔드포인트에 연결됩니다. 각 서버에는 유니버설 컨트롤러 애플리케이션과 OMS가 있습니다. 프로비저닝될 때 가장 최신 버전의 유니버설 컨트롤러가 사용됩니다.

유니버설 컨트롤러는 Tomcat 웹앱에서 ROOT 문서로 배포되며 포트 80에서 제공됩니다. 이 배포를 통해 프론트엔드 로드 밸런서를 쉽게 구성할 수 있습니다.

Stonebranch 와일드카드 인증서(예: <https://customer.stonebranch.cloud>)를 사용하여 TLS를 통한 HTTP 또는 HTTPS를 사용할 수 있습니다. 이렇게 하면 브라우저와 애플리케이션 간의 통신이 보호됩니다.

OMS

유니버설 에이전트와 OMS(Opwise 메시지 서비스)는 각 유니버설 컨트롤러 서버에 있습니다. 고객 측에서 배포한 모든 유니버설 에이전트는 두 OMS 서비스에 모두 연결되도록 설정됩니다. OMS는 유니버설 에이전트와 유니버설 컨트롤러 간의 공통 메시징 서비스 역할을 합니다.

Amazon EFS는 각 서버에 스푼 디렉터리를 마운트합니다. OMS는 이 공유 스푼 디렉터리를 사용하여 컨트롤러와 에이전트로부터 연결 및 작업 정보를 보관합니다. OMS는고가용성 모드에서 작동합니다. 액티브 OMS가 다운되면 패시브 OMS는 모든 데이터에 액세스할 수 있으며 액티브 작업을 자동으로 재개합니다. 유니버설 에이전트는 이러한 변경 사항을 감지하고 새 활성 OMS에 자동으로 연결합니다.

데이터베이스

Amazon Relational Database Service(RDS)는 UAC 데이터베이스를 포함하며, 엔진으로 Amazon Aurora MySQL과 호환됩니다. Amazon RDS는 정기 백업을 관리하고 정기적으로 제공하는 데 도움이 됩니다. 두 유니버설 컨트롤러 인스턴스는 동일한 데이터베이스 엔드포인트에 연결됩니다.

로드 밸런서

Application Load Balancer는 각 인스턴스에 대해 설정됩니다. 로드 밸런서는 언제든지 트래픽을 액티브 컨트롤러로 전달합니다. 인스턴스 도메인 이름은 해당 로드 밸런서 엔드포인트를 가리킵니다.

URL

다음 예에 표시된 대로 각 인스턴스에는 URL이 있습니다.

환경	인스턴스
프로덕션	customer.stonebranch.cloud
개발(비프로덕션)	Customerdev.stonebranch.cloud
테스트(비프로덕션)	customertest.stonebranch.cloud

참고: 비프로덕션 인스턴스 이름은 필요에 따라 설정할 수 있습니다.

높은 가용성

고가용성(HA)은 시스템이 지정된 기간에 장애 없이 지속적으로 작동할 수 있는 능력입니다. 이러한 장애에는 스토리지, CPU 또는 메모리 문제로 인한 서버 통신 응답 지연, 네트워킹 연결 등이 포함되며 이에 국한되지는 않습니다.

HA 요구 사항을 충족하는 방법:

- 모든 EC2 인스턴스, 데이터베이스 및 기타 구성은 동일한 AWS 리전 내 두 개의 개별 가용 영역에 미러링됩니다.
- 컨트롤러는 두 가용 영역에 있는 두 Linux 서버의 Amazon Machine Image(AMI)를 통해 프로비저닝됩니다. 예를 들어 유럽 eu-west-1 리전에서 프로비저닝하는 경우 가용 영역 eu-west-1a 및 가용 영역 eu-west-1c에 유니버설 컨트롤러가 있습니다.
- 애플리케이션 서버에서 직접 작업을 실행할 수 없으며 이러한 서버에는 데이터를 저장할 수 없습니다.
- Application Load Balancer는 각 유니버설 컨트롤러에서 상태 확인을 실행하여 활성 컨트롤러를 식별하고 트래픽을 해당 컨트롤러로 전달합니다. 한 서버에서 문제가 발생하는 경우 로드 밸런서는 자동으로 패시브 유니버설 컨트롤러를 활성 상태로 승격시킵니다. 그러면 로드 밸런서가 상태 확인에서 새 활성 유니버설 컨트롤러 인스턴스를 식별하고 트래픽을 전달하기 시작합니다. 장애 조치는 작업 손실 없이 4분 이내에 발생하며 프론트엔드 URL은 동일하게 유지됩니다.
- Aurora MySQL과 호환되는 데이터베이스 서비스는 유니버설 컨트롤러 데이터를 저장합니다. 프로덕션 환경의 경우 데이터베이스 클러스터는 단일 AWS 지역 내의 서로 다른 두 가용 영역에 있는 두 개의 데이터베이스 인스턴스로 구축됩니다. 두 유니버설 컨트롤러 모두 단일 데이터베이스 클러스터 엔드포인트를 가리키는 Java 데이터베이스 연결성(JDBC) 인터페이스를 사용합니다. 한 데이터베이스 인스턴스에 문제가 발생하는 경우 데이터베이스 클러스터 엔드포인트는 동적으로 정상 인스턴스를 가리킵니다. 따라서 수동 개입은 필요 없습니다.

백업 및 제거

Stonebranch 유니버설 컨트롤러는 표에 표시된 일정에 따라 오래된 데이터를 백업 및 제거하도록 설정되어 있습니다.

Type	Schedule
활동	7일
감사	90일
기록	60일

표시된 날짜보다 오래된 백업 데이터는.xml 형식으로 익스포트되어 파일 시스템에 저장됩니다. 백업 프로세스가 완료되면 이전 데이터는 데이터베이스에서 삭제되고 프로덕션 인스턴스용으로 최대 1년 동안 S3 버킷에 보관됩니다.

유니버설 컨트롤러 인터페이스에서 이 일정을 조정할 수 있습니다. 그러나 이러한 기간을 늘리면 유지 관리 중 가동 중지 시간이 길어질 수 있습니다.

도구

서비스

- [AWS Mainframe Modernization](#)는 메인프레임 애플리케이션을 AWS 관리형 런타임 환경으로 현대화하는 데 도움이 되는 AWS 클라우드 네이티브 플랫폼입니다. 마이그레이션과 현대화를 계획하고 구현하는 데 도움이 되는 도구와 리소스를 제공합니다.
- [Amazon Elastic Block Store\(Amazon EBS\)](#)는 Amazon EC2 인스턴스에 사용할 수 있는 블록 스토리지 볼륨을 제공합니다.
- [Amazon Elastic File System\(Amazon EFS\)](#)은 AWS 클라우드에서 공유 파일 시스템을 생성하고 구성하는 데 도움이 됩니다.
- [Amazon Relational Database Service\(RDS\)](#)는 AWS 클라우드에서 관계형 데이터베이스를 설정, 운영 및 조정하는 데 도움이 됩니다. 이 패턴은 Amazon Aurora MySQL Compatible Edition을 사용합니다.
- [Amazon Simple Storage Service\(S3\)](#)는 원하는 양의 데이터를 저장, 보호 및 검색하는 데 도움이 되는 클라우드 기반 객체 스토리지 서비스입니다.
- [Elastic Load Balancing\(ELB\)](#)은 들어오는 애플리케이션 또는 네트워크 트래픽을 여러 대상에 분산합니다. 예를 들어 하나 이상의 가용 영역에 있는 Amazon EC2 인스턴스, 컨테이너, IP 주소 전반적으로 트래픽을 분산할 수 있습니다. 이 패턴은 Application Load Balancer를 사용합니다.

Stonebranch

- [유니버설 오토메이션 센터\(UAC\)](#)는 엔터프라이즈 워크로드 자동화 제품 시스템입니다. 이 패턴은 다음과 같은 UAC 구성 요소를 사용합니다.
 - Tomcat 웹 컨테이너에서 실행되는 Java 웹 애플리케이션인 [유니버설 컨트롤러](#)는 [유니버설 오토메이션 센터](#)의 엔터프라이즈 작업 스케줄러 및 워크로드 자동화 브로커 솔루션입니다. 컨트롤러는 컨트롤러 정보를 생성, 모니터링 및 구성하기 위한 사용자 인터페이스를 제공하고, 스케줄링 로직을 처리하고, [유니버설 에이전트](#)와 주고받는 모든 메시지를 처리하고, Universal Automation Center의 [고가용성](#) 작업 대부분을 동기화합니다.

- [유니버설 에이전트](#)는 모든 주요 컴퓨팅 플랫폼(레거시 및 분산)에서 기존 작업 스케줄러와 협업하는 벤더 독립적인 스케줄링 에이전트입니다. Z/시리즈, I/시리즈, Unix, Linux 또는 Windows에서 실행되는 모든 스케줄러가 지원됩니다.
- [유니버설 에이전트](#)는 모든 주요 컴퓨팅 플랫폼(레거시 및 분산)에서 기존 작업 스케줄러와 협업하는 벤더 독립적인 스케줄링 에이전트입니다. Z/시리즈, I/시리즈, Unix, Linux 또는 Windows에서 실행되는 모든 스케줄러가 지원됩니다.
- [Stonebranch — aws-mainframe-modernization-stonebranch](#) 통합 [AWS 메인프레임 현대화 유니버설 익스텐션](#)은 [AWS 메인프레임 현대화 플랫폼](#)에서 배치 작업을 실행, 모니터링 및 재실행하기 위한 통합 템플릿입니다.

코드

[이 패턴의 코드는 -integration 리포지토리에서 사용할 수 있습니다.](#) [aws-mainframe-modernization-stonebranch](#) GitHub

에픽

Amazon EC2에 유니버설 컨트롤러 및 유니버설 에이전트 설치

작업	설명	필요한 기술
설치 파일을 다운로드합니다.	Stonebranch 서버에서 설치물을 다운로드합니다. 설치 파일을 받으려면 Stonebranch에 문의하세요.	클라우드 아키텍트
EC2 인스턴스를 실행합니다.	유니버설 컨트롤러와 유니버설 에이전트를 설치하려면 약 3GB의 추가 공간이 필요합니다. 따라서 인스턴스에 최소 30GB의 디스크 공간을 제공하세요. 액세스할 수 있도록 포트 8080을 보안 그룹에 추가합니다.	클라우드 아키텍트
전제 조건을 확인합니다.	설치하기 전에 다음 작업을 수행하세요.	클라우드 관리자, Linux 관리자

작업	설명	필요한 기술
	<p>1. Java 런타임 환경 다운로드에 설명된 대로 Java를 설치합니다.</p> <pre data-bbox="630 380 1029 575">\$ sudo yum -y update \$ sudo yum install java-11-amazon-corretto</pre> <p>지원되는 JAVA 버전 중 하나를 사용해야 합니다. 이전 명령은 java-11을 설치해야 합니다. 계속하기 전에 Java 버전을 확인하고 버전 11을 사용하고 있는지 확인하세요.</p> <p>2. Apache Tomcat 설치 문서에 설명된 대로 다음 명령을 실행합니다.</p> <pre data-bbox="630 1125 1029 1444">\$ sudo yum install tomcat tomcat-admin-webapps \$ sudo systemctl enable tomcat \$ sudo systemctl start tomcat</pre> <p>3. Aurora MySQL 클러스터 생성 및 연결에 설명된 대로 Amazon Aurora 데이터베이스를 생성합니다. Amazon Aurora MySQL 호환 버전을 사용합니다.</p> <p>마스터 사용자 이름과 마스터 암호를 선택합니다. 나머</p>	

작업	설명	필요한 기술
	지 설정은 기본값을 유지합니다.	

작업	설명	필요한 기술
<p>유니버설 컨트롤러를 설치합니다.</p>	<ol style="list-style-type: none"> 1. universal-controller-7.2.0.0.tar 설치 파일을 EC2 인스턴스에 업로드합니다. 2. 설치 파일을 temp 폴더에 보관 취소합니다. <div data-bbox="630 548 1029 701" style="border: 1px solid #ccc; border-radius: 10px; padding: 10px; margin: 10px 0;"> <pre>\$ tar -xvf universal-controller-7.2.0.0.tar</pre> </div> 3. 설치 스크립트에 실행 권한을 부여합니다. <div data-bbox="630 842 1029 957" style="border: 1px solid #ccc; border-radius: 10px; padding: 10px; margin: 10px 0;"> <pre>\$ chmod a+x install-controller.sh</pre> </div> 4. 컨트롤러를 설치합니다. 이 예제에서는 다음 명령을 사용하여 /usr/share/tomcat 아래에 유니버설 컨트롤러를 설치합니다. 이전 단계에서 생성한 Amazon Aurora 데이터베이스를 사용합니다. <div data-bbox="630 1381 1029 1875" style="border: 1px solid #ccc; border-radius: 10px; padding: 10px; margin: 10px 0;"> <pre>\$ sudo ./install-controller.sh --tomcat-dir /usr/share/tomcat/ --controller-file universal-controller-7.2.0.0-build.145.war --dbuser admin --dbpass "*****" --dbname uc --rdbms mysql --dburl jdbc:mysql://database-2-instance-1.c</pre> </div> 	<p>클라우드 아키텍트, Linux 관리자</p>

작업	설명	필요한 기술
	<pre data-bbox="634 205 1027 346">ih63miincgy.us-east-1.rds.amazonaws.com:3306/</pre> <p data-bbox="630 380 1008 464">스크립트 출력의 마지막 줄은 “설치 완료”여야 합니다.</p> <p data-bbox="591 485 966 569">5. EC2 인스턴스에서 다음 URL로 이동합니다.</p> <pre data-bbox="634 611 1027 730">http://<public_ip>:8080/uc</pre> <p data-bbox="591 741 1027 919">6. 로그인 화면에서 사용자 이름 섹션에 ops.admin을 입력하고 암호 필드는 비워 둡니다.</p> <p data-bbox="591 940 1000 1024">7. ops.admin 사용자용 새 암호를 설정합니다.</p>	

작업	설명	필요한 기술
<p>유니버설 에이전트를 설치합니다.</p>	<ol style="list-style-type: none"> 1. sb-7.2.0.1-linux-3.10-x86_64.tar.Z 설치 파일을 EC2 인스턴스에 업로드합니다. 2. EC2 인스턴스에 로그인합니다. 3. 유니버설 에이전트 설치 패키지의 보관을 취소합니다. <pre data-bbox="630 653 1029 814">\$ zcat sb-7.2.0.1-linux-3.10-x86_64.tar.Z tar xvf -</pre> 4. 다음 명령을 실행합니다. <pre data-bbox="630 898 1029 1136">\$ sudo ./unvinst --oms_servers 7878@localhost --omstart yes --python yes</pre> 5. PAM 파일을 생성합니다. <pre data-bbox="630 1220 1029 1339">\$ cp /etc/pam.d/login /etc/pam.d/ucmd</pre> 6. 유니버설 에이전트의 자동 시작을 활성화합니다. <pre data-bbox="630 1478 1029 1640">\$ /sbin/restorecon -v /etc/rc.d/init.d/ubrokerd</pre> 	<p>클라우드 관리자, Linux 관리자</p>

작업	설명	필요한 기술
유니버설 컨트롤러에 OMS를 추가합니다.	<ol style="list-style-type: none"> ops.admin 사용자로 유니버설 컨트롤러에 로그인합니다. 화면 왼쪽 상단의 서비스 메뉴를 선택한 다음 시스템에서 OMS 서버 메뉴를 선택합니다. OMS 서버 주소 필드에 localhost를 입력한 다음 저장합니다. OMS 서버의 상태는 연결됨으로, 세션 상태는 작동중으로 표시됩니다. 	유니버설 컨트롤러 관리자

AWS Mainframe Modernization 유니버설 확장 가져오기 및 작업 생성

작업	설명	필요한 기술
통합 템플릿을 가져옵니다.	<p>이 단계를 수행하려면 AWS Mainframe Modernization 유니버설 확장이 필요합니다. .zip 파일의 최신 버전이 다운로드되었는지 확인합니다.</p> <ol style="list-style-type: none"> ops.admin 사용자와 함께 유니버설 컨트롤러에 로그인합니다. 서비스, 통합 템플릿 가져오기로 이동합니다. 통합 템플릿 .zip 파일 (aws_mainframe_modernization_stonebranch_extension.zip) 	유니버설 컨트롤러 관리자

작업	설명	필요한 기술
	<p>을 선택하고 가져오기를 선택합니다.</p> <p>통합 템플릿을 가져오면 사용 가능한 서비스에 AWS Mainframe Modernization 작업이 표시됩니다.</p>	

작업	설명	필요한 기술
<p>확인 가능한 보안 인증 정보를 활성화합니다.</p>	<ol style="list-style-type: none"> 서비스, AWS Mainframe Modernization 태스크 로 이동합니다. 오른쪽 패널에서 필수 필드를 입력합니다. <ul style="list-style-type: none"> 이름: 새 메인프레임 현대화 작업 에이전트: 유일한 에이전트(AGNT0001)를 선택합니다. <p>AWS Mainframe Modernization 세부 정보의 경우:</p> <ul style="list-style-type: none"> 조치: 환경 목록 작성 AWS 보안 인증: EC2 인스턴스에 AWS Identity 및 Access Management (IAM) 역할을 추가한 경우 이 필드를 비워둘 수 있습니다. AWSAccessKeyID 및 AWSSecretKey 를 사용할 경우 필드 옆에 있는 아이콘()을 선택하세요. <p>열리는 자격 증명 세부 정보 창에서 다음 정보를 입력한 후 저장합니다.</p> <ul style="list-style-type: none"> 이름: AWS Mainframe Modernization 보안 인증 정보 런타임 사용자: 이 필드에 AWS 액세스 키 ID를 입력합니다. 	<p>유니버설 컨트롤러 관리자</p>

작업	설명	필요한 기술
	<ul style="list-style-type: none"> • 런타임 암호: 이 필드에 AWS 암호 키를 입력합니다. • 엔드포인트: 엔드포인트에 올바른 AWS 리전이 있는지 확인합니다. 기본값은 https://m2.us-east-1.amazonaws.com입니다. • 리전: AWS Mainframe Modernization 서비스의 리전을 입력합니다. 기본값은 us-east-1 입니다. <p>3. 나머지 필드의 기본값은 유지하고 작업은 저장합니다.</p>	

작업	설명	필요한 기술
<p>태스크를 실행합니다.</p>	<ol style="list-style-type: none"> 오른쪽 패널 상단에서 작업 시작을 선택합니다. 확인 창에서 실행을 선택합니다. 그 후 유니버설 컨트롤러 콘솔에 다음 메시지와 유사한 메시지가 표시됩니다. <p>2022-08-24 오전 10:11:49</p> <p>태스크 인스턴스 sys_id 166129149363414631 3NC8E38DB8OZJY를 사용하여 유니버설 태스크 “새 메인프레임 현대화 태스크”를 성공적으로 시작했습니다.</p> 인스턴스로 이동합니다. 인스턴스 탭이 보이지 않으면 오른쪽 화살표를 선택하여 오른쪽으로 스크롤하세요. 목록에서 작업 인스턴스의 컨텍스트 메뉴 (마우스 오른쪽 단추 클릭)를 열고 출력 검색을 선택한 다음 출력 검색에서 제출을 선택합니다 출력 검색 창에서 STDOUT의 환경 목록을 볼 수 있습니다. 	<p>유니버설 컨트롤러 관리자</p>

배치 작업 시작 테스트

작업	설명	필요한 기술
<p>배치 작업을 위한 태스크를 생성합니다.</p>	<ol style="list-style-type: none"> 서비스, AWS Mainframe Modernization 태스크 로 이동합니다. 오른쪽 패널에서 필수 필드를 입력합니다. <ul style="list-style-type: none"> 이름: 새 메인프레임 현대화 작업 에이전트: 유일한 에이전트(AGNT0001)를 선택합니다. <p>AWS Mainframe Modernization 세부 정보의 경우:</p> <ul style="list-style-type: none"> 조치: 배치 시작(또는 배치를 시작하고 배치 작업을 실행하기 위해 기다렸다가 AWS에서 작업이 완료될 때까지 기다림) AWS 보안 인증: EC2 인스턴스에 IAM 역할을 추가한 경우 이 필드를 비워둘 수 있습니다. AWSAccessKeyId 및 AWSSecretKey 를 사용할 경우 필드 옆에 있는 아이콘()을 선택하세요. 엔드포인트: 엔드포인트에 올바른 AWS 리전이 있는지 확인합니다. 기본값은 https://m2.us-east-1.amazonaws.com입니다. 	<p>유니버설 컨트롤러 관리자</p>

작업	설명	필요한 기술
	<ul style="list-style-type: none"> • 리전: AWS Mainframe Modernization 서비스의 리전을 입력합니다. 기본 값은 us-east-1 입니다. • 애플리케이션: 필드() 옆의 아이콘을 선택하고 애플리케이션 선택 새 로 고침에서 제출을 선택합니다. 그러면 AWS Mainframe Modernization 서비스에 연결되고 애플리케이션 목록이 반환됩니다. 이제 드롭다운 목록에서 애플리케이션을 선택할 수 있습니다. 배치 작업을 실행할 애플리케이션을 선택합니다. • JCL 파일 이름: RUNHELLO.jcl • 성공 또는 실패 대기: 이 옵션을 선택하면 배치 작업의 상태가 성공 또는 실패가 될 때까지 작업이 대기합니다. • 폴링 간격: 각 폴링 사이의 시간입니다. • 실행 로그 가져오기: 이 옵션을 선택하면 일괄 작업이 완료될 때 로그를 자동으로 가져옵니다. • 로그 형식: 출력되는 로그의 형식입니다. 텍스트 또 	

작업	설명	필요한 기술
	<p>는 JSON 형식일 수 있습니다.</p> <p>3. 나머지 필드의 기본값은 유지하고 작업은 저장합니다.</p>	
<p>태스크를 실행합니다.</p>	<ol style="list-style-type: none"> 오른쪽 패널 상단에서 작업 시작을 선택합니다. 확인 창에서 실행을 선택합니다. 그 후 유니버설 컨트롤러 콘솔에 다음 메시지와 유사한 메시지가 표시됩니다. <p>2022-08-24 오전 11:11:59</p> <p>작업 인스턴스 sys_id <sys id>를 사용하여 유니버설 태스크 “메인프레임 현대화 시작 배치”를 성공적으로 시작했습니다.</p> 인스턴스로 이동합니다. 인스턴스 탭이 보이지 않으면 오른쪽 화살표를 선택하여 오른쪽으로 스크롤하세요. 목록에서 작업 인스턴스의 컨텍스트 메뉴 (마우스 오른쪽 단추 클릭)를 열고 출력 검색을 선택한 다음 출력 검색에서 제출을 선택합니다 출력 검색 창에서 STDOUT의 환경 목록을 볼 수 있습니다. 	<p>유니버설 컨트롤러 관리자</p>

여러 작업을 위한 워크플로 생성

작업	설명	필요한 기술
작업을 복사합니다.	<ol style="list-style-type: none"> 1. 사본을 생성할 작업의 컨텍스트 마우스 오른쪽 버튼 클릭) 메뉴를 열고 복사를 선택합니다. 2. AWS Mainframe Modernization 작업 복사 창에서 새 작업의 새 이름을 다음과 같이 입력합니다. 메인프레임 현대화 시작 배치- RUNAWS2. 3. 메인프레임 현대화 시작 배치 - RUNAWS3라는 이름을 사용하여 작업과 함께 다시 복사합니다. 4. 메인프레임 현대화 시작 배치 - RUNAWS4라는 이름을 사용하여 작업과 함께 다시 복사합니다. 5. 메인프레임 현대화 시작 배치 - FOOBAR라는 이름을 사용하여 작업을 마지막으로 복사합니다. 	유니버설 컨트롤러 관리자
작업을 업데이트합니다.	<ol style="list-style-type: none"> 1. (두 번 클릭하여) 메인프레임 현대화 시작 배치 열기 - RUNAWS2 작업에서 JCL 파일 이름 필드를 RUNAWS2.jc1 로 변경하고 저장합니다. 2. (두 번 클릭하여) 메인프레임 현대화 시작 배치 열기 - RUNAWS3 작업에서 JCL 파일 이름 필드를 	유니버설 컨트롤러 관리자

작업	설명	필요한 기술
	<p>RUNAWS3.jc1 로 변경하고 저장합니다.</p> <p>3. (두 번 클릭하여) 메인프레임 현대화 시작 배치 열기 - RUNAWS4 작업에서 JCL 파일 이름 필드를 RUNAWS4.jc1 로 변경하고 저장합니다.</p> <p>4. (두 번 클릭하여) 메인프레임 현대화 시작 배치 열기 - FOOBAR 작업에서 JCL 파일 이름 필드를 MISSING.jc1 로 변경하고 저장합니다. JCL 파일 이름 값이 올바르지 않기 때문에 이 작업은 실패합니다.</p>	

작업	설명	필요한 기술
워크플로를 생성합니다.	<ol style="list-style-type: none"> 1. 서비스, 워크플로로 이동합니다. 2. 오른쪽 패널의 이름 필드에 메인프레임 현대화 워크플로를 입력하고 저장합니다. 3. 오른쪽 패널에서 워크플로 편집을 선택합니다. 4. 워크플로 편집기 탭의 작업 추가 버튼 (+). 5. 작업 찾기 창에서 검색을 선택하여 유니버설 컨트롤러의 모든 작업을 확인합니다. 6. 메인프레임 현대화 Start Batch Task 옆에 있는 아이콘을 클릭하고 아이콘을 워크플로 편집기의 빈 곳으로 드래그합니다. 7. 다른 메인프레임 현대화 작업에 대해서도 동일한 작업을 반복하여 추가 정보 섹션에 표시된 대로 배치합니다. 8. 연결 버튼()을 선택하고 작업을 서로 연결합니다. 작업을 다른 작업에 연결하려면 작업 중간을 클릭하여 대상 작업으로 드래그하세요. 9. 추가 정보 섹션에 표시된 대로 작업을 연결하고 워크플로를 저장합니다. 10. 워크플로 편집기의 빈 공간을 마우스 오른쪽 버튼으로 클릭하고 워크플로 시작을 	유니버설 컨트롤러 관리자

작업	설명	필요한 기술
	<p>선택한 다음 확인을 선택합니다.</p>	
워크플로의 상태를 확인합니다.	<ol style="list-style-type: none"> 왼쪽 메뉴에서 활동을 선택합니다. 창 가운데에서 시작을 선택합니다. <p>목록에서 작업 인스턴스 목록을 볼 수 있습니다.</p> <ol style="list-style-type: none"> (두 번 클릭하여) 목록에서 메인프레임 현대화 워크플로를 열거나 (마우스 오른쪽 버튼을 클릭하여) 컨텍스트 메뉴를 열고 워크플로 작업 명령, 워크플로 보기를 선택합니다. <p>추가 정보 섹션에 표시된 대로 작업이 표시됩니다. 누락된 JCL 파일을 사용했기 때문에 두 번째 작업은 실패할 것으로 예상되었습니다.</p>	유니버설 컨트롤러 관리자

실패한 배치 작업 문제 해결 및 재실행

작업	설명	필요한 기술
실패한 작업을 수정하고 다시 실행하세요.	<ol style="list-style-type: none"> (두 번 클릭하여) 실패한 작업을 열고 작업의 오류를 확인합니다. 실패한 작업을 수정하는 데는 두 가지 옵션이 있습니다. 	유니버설 컨트롤러 관리자

작업	설명	필요한 기술
	<ul style="list-style-type: none"> • JCL 파일 이름을 수정하고 FOOBAR.jcl 로 설정합니다. • JCL 파일 이름(임시)에 올바른 JCL 파일 이름을 추가합니다. 이 필드는 JCL 파일 이름 필드를 덮어씁니다. <p>이 파일럿의 경우 두 번째 옵션을 선택하고 작업 인스턴스를 저장합니다.</p> <p>3. 워크플로 모니터에서 실패한 작업의 컨텍스트(마우스 오른쪽 단추 클릭) 메뉴를 열고 명령, 재실행을 선택합니다.</p> <p>4. 그러면 모든 작업이 성공적으로 완료됩니다.</p>	

애플리케이션 생성, 시작, 애플리케이션 중지 태스크

작업	설명	필요한 기술
<p>애플리케이션 시작 작업을 생성합니다.</p>	<ol style="list-style-type: none"> 1. 서비스, AWS Mainframe Modernization 태스크 로 이동합니다. 2. 오른쪽 패널에서 필수 필드를 입력합니다. <ul style="list-style-type: none"> • 이름: 메인프레임 현대화 시작 애플리케이션 • 에이전트: 유일한 에이전트(AGNT0001) 선택 	<p>유니버설 컨트롤러 관리자</p>

작업	설명	필요한 기술
	<p>AWS Mainframe Modernization 세부 정보의 경우:</p> <ul style="list-style-type: none"> • 조치: 애플리케이션 시작 • AWS 보안 인증: EC2 인스턴스에 IAM 역할을 추가한 경우 이 필드를 비워 둘 수 있습니다. AWSAccessKeyID 및 AWSSecretKey 를 사용할 경우 이전에 생성한 보안 인증 정보를 선택하세요. • 엔드포인트: 엔드포인트에 올바른 리전이 있는지 확인합니다. 기본 값은 https://m2.us-east-1.amazonaws.com입니다. • 리전: AWS Mainframe Modernization 서비스의 리전을 입력합니다. 기본 값은 us-east-1 입니다. • 애플리케이션: 필드() 옆의 아이콘을 선택하고 애플리케이션 선택 새 로 고침에서 제출을 선택합니다. 그러면 AWS Mainframe Modernization 서비스에 연결되고 애플리케이션 목록이 반환됩니다. 이제 드롭다운 목록에서 애플리케이션을 선택 	

작업	설명	필요한 기술
	<p>택할 수 있습니다. 배치 작업을 실행할 애플리케이션을 선택합니다.</p> <ul style="list-style-type: none"> • 성공 또는 실패 대기: 이 옵션을 선택하면 배치 작업의 상태가 성공 또는 실패가 될 때까지 작업이 대기합니다. • 폴링 간격: 각 폴링 사이의 시간입니다. • 실행 로그 가져오기: 이 옵션을 선택하면 일괄 작업이 완료될 때 로그를 자동으로 가져옵니다. • 로그 형식: 출력되는 로그의 형식입니다. 텍스트 또는 JSON 형식일 수 있습니다. <p>3. 나머지 필드의 기본값은 유지하고 작업은 저장합니다.</p> <p>4. 이제 이 작업을 복사하고 애플리케이션 중지 작업을 생성하세요. 이름을 메인프레임 현대화 중지 애플리케이션으로 변경하고 작업을 애플리케이션 중지로 변경합니다.</p>	

Batch Execution 취소 작업 생성

작업	설명	필요한 기술
<p>배치 취소 작업을 생성합니다.</p>	<ol style="list-style-type: none"> 1. 서비스, AWS Mainframe Modernization 태스크 로 이동합니다. 2. 오른쪽 패널에서 필수 필드를 입력합니다. <ul style="list-style-type: none"> • 이름: 메인프레임 현대화 취소 배치 실행 • 에이전트: 유일한 에이전트(AGNT0001) 선택 <p>AWS Mainframe Modernization 세부 정보의 경우:</p> <ul style="list-style-type: none"> • 조치: 배치 실행 취소 • AWS 보안 인증: EC2 인스턴스에 IAM 역할을 추가한 경우 이 필드를 비워둘 수 있습니다. AWSAccessKeyId 및 AWSSecretKey 를 사용할 경우 이전에 생성한 보안 인증 정보를 선택하세요. • 엔드포인트: 엔드포인트에 올바른 리전이 있는지 확인합니다. 기본값은 https://m2.us-east-1.amazonaws.com입니다. • 리전: AWS Mainframe Modernization 서비스의 리전을 입력합니다. 기본 	

작업	설명	필요한 기술
	<p>값은 us-east-1 입니다.</p> <ul style="list-style-type: none"> • 애플리케이션: 필드() 옆의 아이콘을 선택하고 애플리케이션 선택 새 로 고침에서 제출을 선택합니다. 그러면 AWS Mainframe Modernization 서비스에 연결되고 애플리케이션 목록이 반환됩니다. 이제 드롭다운 목록에서 애플리케이션을 선택할 수 있습니다. 배치 작업을 실행할 애플리케이션을 선택합니다. • 성공 또는 실패 대기: 이 옵션을 선택하면 배치 작업의 상태가 성공 또는 실패가 될 때까지 작업이 대기합니다. • 폴링 간격: 각 폴링 사이의 시간입니다. • 실행 로그 가져오기: 이 옵션을 선택하면 일괄 작업이 완료될 때 로그를 자동으로 가져옵니다. • 로그 형식: 출력되는 로그의 형식입니다. 텍스트 또는 JSON 형식일 수 있습니다. <p>3. 나머지 필드의 기본값은 유지하고 작업은 저장합니다.</p>	

관련 리소스

- [유니버설 컨트롤러](#)
- [유니버설 에이전트](#)
- [LDAP 설정](#)
- [Single Sign-On 설정](#)
- [고가용성](#)
- [익스프레스 컨버전 도구](#)

추가 정보

워크플로 편집기의 아이콘

모든 작업이 연결되었습니다

워크플로 상태

Connect from Precisely를 사용하여 VSAM 파일을 Amazon RDS 또는 Amazon MSK로 마이그레이션하고 복제하기

작성자: Prachi Khanna (AWS) 및 Boopathy GOPALSAMY (AWS)

환경: PoC 또는 파일럿	소스: VSAM	대상: 데이터베이스
R 유형: 리아키텍트	워크로드: IBM	기술: 메인프레임, 현대화
AWS 서비스: Amazon MSK, Amazon RDS, AWS Mainframe Modernization		

요약

이 패턴은 [Connect](#) from Precisely를 사용하여 메인프레임에서 AWS Cloud의 대상 환경으로 가상 스토리지 액세스 방법(VSAM) 파일을 마이그레이션하고 복제하는 방법을 보여줍니다. 이 패턴에서 다루는 대상 환경에는 Amazon Relational Database Service(Amazon RDS) 및 Amazon Managed Streaming for Apache Kafka(Amazon MSK)가 포함됩니다. Connect는 [변경 데이터 캡처\(CDC\)](#)를 사용하여 소스 VSAM 파일의 업데이트를 지속적으로 모니터링한 다음 이러한 업데이트를 하나 이상의 AWS 대상 환경으로 전송합니다. 이 패턴을 사용하여 애플리케이션 현대화 또는 데이터 분석 목표를 달성할 수 있습니다. 예를 들어, Connect를 사용하여 짧은 지연 시간으로 VSAM 애플리케이션 파일을 AWS Cloud로 마이그레이션하거나, 애플리케이션 현대화에 필요한 것보다 긴 동기화 지연 시간을 견딜 수 있는 분석을 위해 VSAM 데이터를 AWS 데이터 웨어하우스 또는 데이터 레이크로 마이그레이션할 수 있습니다.

사전 조건 및 제한 사항

사전 조건

- [IBM z/OS V2R1](#) 이상
- [z/OS용 CICS 트랜잭션 서버 \(CICS TS\) V5.1](#) 이상 (CICS/VSAM 데이터 캡처)
- [IBM MQ 8.0](#) 이상
- [z/OS 보안 요구 사항](#) 준수(예: SQData 로드 라이브러리에 대한 APF 인증)
- VSAM 복구 로그 켜짐

- (선택 사항) [CDC 로그를 자동으로 캡처하기 위한 CICS VSAM 복구 버전 \(CICS VR\)](#)
- 활성 상태의 AWS 계정.
- 기존 플랫폼에서 연결할 수 있는 서브넷이 있는 [Amazon Virtual Private Cloud\(VPC\)](#)
- Precisely의 VSAM Connect 라이선스

제한 사항

- Connect는 소스 VSAM 스키마 또는 카피북을 기반으로 하는 자동 대상 테이블 생성을 지원하지 않습니다. 대상 테이블 구조를 처음으로 정의해야 합니다.
- Amazon RDS와 같은 비스트리밍 대상의 경우 Apply Engine 구성 스크립트에서 전환 소스와 대상 간 매핑을 지정해야 합니다.
- 로깅, 모니터링 및 경고 기능은 API를 통해 구현되며 외부 구성 요소 (예: Amazon CloudWatch) 가 제대로 작동하려면 필요합니다.

제품 버전

- z/OS용 SQData 40134
- Amazon Elastic Compute Cloud(Amazon EC2) 기반 Amazon Linux Amazon Machine Image(AMI)용 SQData 4.0.43

아키텍처

소스 기술 스택

- 작업 제어 언어(JCL)
- z/OS Unix shell 및 Interactive System Productivity Facility(ISPF)
- VSAM 유틸리티 (IDCAMS)

대상 기술 스택

- Amazon EC2
- Amazon MSK
- Amazon RDS
- Amazon VPC

대상 아키텍처

VSAM 파일을 Amazon RDS로 마이그레이션

다음 다이어그램은 소스 환경(온프레미스 메인프레임)의 CDC 에이전트/게시자와 대상 환경(AWS Cloud)의 [Apply Engine](#)을 사용하여 실시간 또는 거의 실시간으로 Amazon RDS와 같은 관계형 데이터베이스로 VSAM 파일을 마이그레이션하는 방법을 보여줍니다.

이 다이어그램은 다음 배치 워크플로를 보여 줍니다.

1. Connect는 변경 내용을 식별한 다음 변경 내용을 로그스트림으로 전송하기 위해 백업 파일의 VSAM 파일을 비교하여 파일의 변경 내용을 캡처합니다.
2. 게시자는 시스템 로그스트림의 데이터를 사용합니다.
3. 게시자는 캡처된 데이터 변경 사항을 TCP/IP를 통해 대상 엔진에 전달합니다. Controller Daemon은 소스 환경과 대상 환경 간의 통신을 인증합니다.
4. 대상 환경의 Apply Engine은 Publisher 에이전트로부터 변경 내용을 받아 관계형 또는 비관계형 데이터베이스에 적용합니다.

이 다이어그램은 다음 온라인 워크플로를 보여 줍니다.

1. Connect는 로그 복제를 사용하여 온라인 파일의 변경 사항을 캡처한 다음 캡처된 변경 사항을 로그 스트림으로 스트리밍합니다.
2. 게시자는 시스템 로그스트림의 데이터를 사용합니다.
3. 게시자는 캡처된 데이터 변경 사항을 TCP/IP를 통해 대상 엔진에 전달합니다. Controller Daemon은 소스 환경과 대상 환경 간의 통신을 인증합니다.
4. 대상 환경의 Apply Engine은 Publisher 에이전트로부터 변경 내용을 받아 관계형 또는 비관계형 데이터베이스에 적용합니다.

VSAM 파일을 Amazon MSK로 마이그레이션

다음 다이어그램은 고성능 모드에서 메인프레임에서 Amazon MSK로 VSAM 데이터 구조를 스트리밍하고 Amazon MSK와 통합되는 JSON 또는 AVRO 스키마 변환을 자동으로 생성하는 방법을 보여줍니다.

이 다이어그램은 다음 배치 워크플로를 보여 줍니다.

1. Connect는 변경 내용을 파악하기 위해 CICS VR을 사용하거나 백업 파일의 VSAM 파일을 비교하여 파일에 대한 변경 내용을 캡처합니다. 캡처된 변경 사항은 로그스트림으로 전송됩니다.
2. 게시자는 시스템 로그스트림의 데이터를 사용합니다.
3. 게시자는 캡처된 데이터 변경 사항을 TCP/IP를 통해 대상 엔진에 전달합니다. Controller Daemon은 소스 환경과 대상 환경 간의 통신을 인증합니다.
4. 병렬 처리 모드에서 작동하는 Replicator Engine은 데이터를 작업 캐시 단위로 분할합니다.
5. 작업자 스레드는 캐시에서 데이터를 캡처합니다.
6. 데이터는 작업자 스레드에서 Amazon MSK 주제에 게시됩니다.
7. [사용자는 커넥터를 사용하여 Amazon MSK의 변경 내용을 Amazon DynamoDB, Amazon S3 \(Amazon Simple Storage Service\) OpenSearch 또는 Amazon 서비스와 같은 대상에 적용합니다.](#)

이 다이어그램은 다음 온라인 워크플로를 보여 줍니다.

1. 온라인 파일의 변경 사항은 로그 복제를 사용하여 캡처됩니다. 캡처된 변경 사항은 로그스트림으로 스트리밍됩니다.
2. 게시자는 시스템 로그스트림의 데이터를 사용합니다.
3. 게시자는 캡처된 데이터 변경 사항을 TCP/IP를 통해 대상 엔진에 전달합니다. Controller Daemon은 소스 환경과 대상 환경 간의 통신을 인증합니다.
4. 병렬 처리 모드에서 작동하는 Replicator Engine은 데이터를 작업 캐시 단위로 분할합니다.
5. 작업자 스레드는 캐시에서 데이터를 캡처합니다.
6. 데이터는 작업자 스레드에서 Amazon MSK 주제에 게시됩니다.
7. [사용자는 커넥터를 사용하여 Amazon MSK의 변경 내용을 DynamoDB OpenSearch , Amazon S3 또는 서비스와 같은 대상에 적용합니다.](#)

도구

- [Amazon Managed Streaming for Apache Kafka\(Amazon MSK\)](#)는 Apache Kafka를 사용하여 스트리밍 데이터를 처리하는 애플리케이션의 구축 및 실행에 도움이 되는 완전 관리형 서비스입니다.
- [Amazon Relational Database Service\(RDS\)](#)는 AWS 클라우드에서 관계형 데이터베이스를 설정, 운영 및 조정하는 데 도움이 됩니다.

에픽

원본 환경을 준비하세요.

작업	설명	필요한 기술
<p>Connect CDC 4.1을 설치합니다.</p>	<ol style="list-style-type: none"> 1. 라이선스와 설치 패키지를 받으려면 Prececipty 지원 팀에 문의하십시오. 2. 예제 JCL을 사용하여 Connect CDC 4.1을 설치합니다. 자세한 지침은 Precisely 설명서에서 JCL을 사용하여 Connect CDC(SQData) 설치하기를 참고하십시오. 3. SETPROG APF 명령을 실행하여 Connect 로드 라이브러리 SQDATA.V4 nnn.LOADLIB를 승인합니다. 	<p>IBM 메인프레임 개발자/관리자</p>
<p>zFS 디렉토리를 설정합니다.</p>	<p>zFS 디렉토리를 설정하려면 Precisely 설명서에 있는 zFS 변수 디렉토리의 지침을 따르십시오.</p> <p>참고: Controller Daemon 및 Capture/Publisher 에이전트 구성은 z/OS UNIX 시스템 서비스 파일 시스템(zFS라고 함)에 저장됩니다. Controller Daemon, Capture, Storage, Publisher 에이전트에는 적은 수의 파일을 저장하기 위해 사전 정의된 zFS 디렉터리 구조가 필요합니다.</p>	<p>IBM 메인프레임 개발자/관리자</p>

작업	설명	필요한 기술
<p>TCP/IP 포트를 구성합니다.</p>	<p>TCP/IP 포트를 구성하려면 Precisely 설명서에 있는 TCP/IP 포트의 지침을 따르십시오.</p> <p>참고: Controller Daemon을 사용하려면 소스 시스템에 TCP/IP 포트가 필요합니다. 포트는 (캡처된 변경 데이터가 처리되는) 대상 시스템의 엔진에서 참조합니다.</p>	<p>IBM 메인프레임 개발자/관리자</p>
<p>z/OS 로그스트림을 생성합니다.</p>	<p>z/OS 로그스트림을 생성하려면 Preciply 설명서의 z/OS 시스템 로그스트림 생성 지침을 따르십시오.</p> <p>참고: Connect는 마이그레이션 중에 로그스트림을 사용하여 소스 환경과 대상 환경 간에 데이터를 캡처하고 스트리밍합니다.</p> <p>z/OS를 생성하는 JCL의 예는 Preciply 설명서의 z/OS LogStream 시스템 로그스트림 생성을 참조하십시오.</p>	<p>IBM 메인프레임 개발자</p>
<p>zFS 사용자 및 시작 작업의 ID를 식별하고 승인합니다.</p>	<p>RACF를 사용하여 OMVS zFS 파일 시스템에 대한 액세스 권한을 부여합니다. JCL의 예는 Precisely 설명서의 zFS 사용자 및 시작 작업 ID 식별 및 권한 부여를 참고하십시오.</p>	<p>IBM 메인프레임 개발자/관리자</p>

작업	설명	필요한 기술
<p>z/OS 공개/개인 키와 인증된 키 파일을 생성합니다.</p>	<p>JCL을 실행하여 키 쌍을 생성합니다. 예제는 이 패턴의 추가 정보 섹션에 있는 키 쌍 예제를 참고하십시오.</p> <p>자세한 지침은 Precisely 설명서의 z/OS 공개 및 개인 키와 인증된 키 파일 생성을 참고하십시오.</p>	<p>IBM 메인프레임 개발자/관리자</p>
<p>CICS VSAM 로그 복제를 활성화하고 이를 로그스트림에 연결합니다.</p>	<p>다음 SQL 스크립트를 실행합니다.</p> <pre data-bbox="594 793 1026 1188"> //STEP1 EXEC PGM=IDCAM S //SYSPRINT DD SYSOUT=* //SYSIN DD * ALTER SQDATA.CI CS.FILEA - LOGSTREAMID(SQDATA .VSAMCDC.LOG1) - LOGREPLICATE </pre>	<p>IBM 메인프레임 개발자/관리자</p>

작업	설명	필요한 기술
<p>FCT를 통해 VSAM 파일 복구 로그를 활성화합니다.</p>	<p>다음 매개변수 변경 사항을 반영하도록 FCT(파일 제어 테이블)를 수정하십시오.</p> <pre> Configure FCT Params CEDA ALT FILE(name) GROUP(groupname) DSNAME(data set name) RECOVERY(NONE BACK OUTONLY ALL) FWDRECOVLOG(NO 1-9 9) BACKUPTYPE(STATIC DYNAMIC) RECOVERY PARAMETERS RECOVery : None Backoutonly All Fwdrecovlog : No 1-99 BAckuptype : Static Dynamic </pre>	<p>IBM 메인프레임 개발자/관리자</p>
<p>퍼블리셔 에이전트용 CD를 설정합니다. CzLog</p>	<ol style="list-style-type: none"> 1. CD CzLog 게시자 CAB 파일을 생성합니다. 2. 게시된 데이터를 암호화합니다. 3. CD CzLog 퍼블리셔 런타임 JCL을 준비하십시오. 	<p>IBM 메인프레임 개발자/관리자</p>

작업	설명	필요한 기술
Controller Daemon을 활성화합니다.	<ol style="list-style-type: none"> 1. ISPF 패널을 열고 다음 명령을 실행하여 Precisely 메뉴(EXEC 'SQDATA.V4nnnnn.ISPFLIB(SQDC\$STA)' 'SQDATA.V4nnnnn')를 엽니다. 2. Controller Daemon을 설정하려면 메뉴에서 옵션 2를 선택합니다. 	IBM 메인프레임 개발자/관리자
게시자를 활성화합니다.	<ol style="list-style-type: none"> 1. ISPF 패널을 열고 다음 명령을 실행하여 Precisely 메뉴(EXEC 'SQDATA.V4nnnnn.ISPFLIB(SQDC\$STA)' 'SQDATA.V4nnnnn')를 엽니다. 2. 게시자를 설정하려면 메뉴에서 옵션 3을 선택하고 삽입은 I를 선택합니다. 	IBM 메인프레임 개발자/관리자
로그스트림을 활성화합니다.	<ol style="list-style-type: none"> 1. ISPF 패널을 열고 다음 명령을 실행하여 Precisely 메뉴(EXEC 'SQDATA.V4nnnnn.ISPFLIB(SQDC\$STA)' 'SQDATA.V4nnnnn')를 엽니다. 2. 로그스트림을 설정하려면 메뉴에서 옵션 4를 선택하고 삽입은 I를 선택합니다. 그런 다음 이전 단계에서 생성한 로그스트림의 이름을 입력합니다. 	IBM 메인프레임 개발자/관리자

대상 환경(AWS) 준비

작업	설명	필요한 기술
EC2 인스턴스에 Precisely를 설치합니다.	Amazon EC2용 Amazon Linux AMI에 Connect from Precisely를 설치하려면 Precisely 설명서에 있는 UNIX에 Connect CDC(SQData) 설치 의 지침을 따르십시오.	일반 AWS
TCP/IP 포트를 엽니다.	인바운드 및 아웃바운드 액세스를 위한 Controller Daemon 포트를 포함하도록 보안 그룹을 수정하려면 Precisely 설명서에 있는 TCP/IP 의 지침을 따르십시오.	일반 AWS
파일 디렉토리를 생성합니다.	파일 디렉토리를 생성하려면 Precisely 설명서의 대상 적용 환경 준비 지침을 따르십시오.	일반 AWS
Apply Engine 구성 파일을 생성합니다.	Apply Engine의 작업 디렉토리에 Apply Engine 구성 파일을 생성합니다. 다음 예제 구성 파일은 Apache Kafka를 대상으로 보여줍니다. <pre>builtin.features=S ASL_SCRAM security.protocol= SASL_SSL sasl.mechanism=SCR AM-SHA-512 sasl.username= sasl.password= metadata.broker.li st=</pre>	일반 AWS

작업	설명	필요한 기술
	참고: 자세한 내용은 Apache Kafka 설명서의 보안 을 참조하세요.	
Apply Engine 처리를 위한 스크립트를 생성합니다.	Apply Engine용 스크립트를 생성하여 소스 데이터를 처리하고 소스 데이터를 대상에 복제합니다. 자세한 내용은 Precisely 설명서의 Apply Engine 스크립트 생성 을 참고하십시오.	일반 AWS
스크립트를 실행합니다.	스크립트를 시작하려면 SQDPARSE 및 SQDENG 명령을 실행합니다. 자세한 내용은 Precisely 설명서의 zOS용 스크립트 구문 분석 을 참고하십시오.	일반 AWS

환경 검증

작업	설명	필요한 기술
CDC 처리를 위한 VSAM 파일 및 대상 테이블 목록을 검증합니다.	<ol style="list-style-type: none"> 복제 로그, 복구 로그, FCT 매개변수, 로그스트림을 포함한 VSAM 파일을 검증합니다. 테이블이 필수 스키마 정의, 테이블 액세스, 기타 기준에 따라 생성되었는지 여부를 포함하여 대상 데이터베이스 테이블을 검증합니다. 	일반 AWS, 메인프레임

작업	설명	필요한 기술
Connect CDC SQData 제품이 연결되어 있는지 확인하십시오.	<p>테스트 작업을 실행하고 이 작업의 반환 코드가 0(성공)인지 확인합니다.</p> <p>참고: Connect CDC SQData Apply Engine 상태 메시지에는 활성 연결 메시지가 표시되어야 합니다.</p>	일반 AWS, 메인프레임

테스트 케이스 실행 및 검증 (일괄)

작업	설명	필요한 기술
메인프레임에서 일괄 작업을 실행합니다.	<p>수정된 JCL을 사용하여 일괄 애플리케이션 작업을 실행합니다. 수정된 JCL에 다음을 수행하는 단계를 포함시키십시오.</p> <ol style="list-style-type: none"> 1. 데이터 파일을 백업합니다. 2. 백업 파일을 수정된 데이터 파일과 비교하고 델타 파일을 생성한 다음 메시지의 델타 레코드 수를 기록합니다. 3. 델타 파일을 z/OS 로그스트림으로 푸시합니다. 4. JCL을 실행합니다. 예제 JCL은 Precisely 설명서의 파일 비교 캡처 JCL 준비를 참고하십시오. 	일반 AWS, 메인프레임
로그스트림을 확인합니다.	로그스트림을 확인하여 완료된 메인프레임 일괄 작업에 대한 변경 데이터를 볼 수 있는지 확인합니다.	일반 AWS, 메인프레임

작업	설명	필요한 기술
소스 델타 변경 및 대상 테이블의 개수를 확인합니다.	<p>레코드가 집계되었는지 확인하려면 다음을 수행합니다.</p> <ol style="list-style-type: none"> 1. 일괄 JCL 메시지에서 소스 델타 수를 수집합니다. 2. Apply Engine을 모니터링하여 VSAM 파일에 삽입, 업데이트 또는 삭제된 레코드 수의 레코드 수준 수를 모니터링합니다. 3. 대상 테이블에서 레코드 수를 쿼리합니다. 4. 모든 다른 레코드 수를 비교하고 집계합니다. 	일반 AWS, 메인프레임

테스트 케이스 실행 및 검증(온라인)

작업	설명	필요한 기술
CICS 지역에서 온라인 거래를 실행합니다.	<ol style="list-style-type: none"> 1. 온라인 트랜잭션을 실행하여 테스트 케이스의 유효성을 검사합니다. 2. 트랜잭션 실행 코드의 유효성을 검사합니다(RC=0 — 성공). 	IBM 메인프레임 개발자
로그스트림을 확인합니다.	로그스트림이 특정 레코드 수준 변경으로 채워졌는지 확인합니다.	AWS 메인프레임 개발자
대상 데이터베이스의 개수를 확인합니다.	Apply Engine에서 레코드 수준 수를 모니터링합니다.	Precisely, Linux

작업	설명	필요한 기술
대상 데이터베이스의 레코드 수와 데이터 레코드를 확인합니다.	대상 데이터베이스를 쿼리하여 레코드 수와 데이터 레코드를 확인합니다.	일반 AWS

관련 리소스

- [VSAM z/OS](#) (Precisely 설명서)
- [Apply engine](#) (Precisely 설명서)
- [Replicator engine](#) (Precisely 설명서)
- [로그 스트림](#) (IBM 설명서)

추가 정보

구성 파일 예

다음은 소스 환경이 메인프레임이고 대상 환경이 Amazon MSK인 로그스트림의 구성 파일 예시입니다.

```
-- JOBNAME -- PASS THE SUBSCRIBER NAME
-- REPORT progress report will be produced after "n" (number) of Source records
processed.

JOBNAME VSMTOKFK;
--REPORT EVERY 100;
-- Change Op has been 'I' for insert, 'D' for delete , and 'R' for Replace. For RDS
it is 'U' for update
-- Character Encoding on z/OS is Code Page 1047, on Linux and UNIX it is Code Page
819 and on Windows, Code Page 1252
OPTIONS
CDCOP('I', 'U', 'D'),
PSEUDO NULL = NO,
USE AVRO COMPATIBLE NAMES,
APPLICATION ENCODING SCHEME = 1208;

-- SOURCE DESCRIPTIONS
```

```

BEGIN GROUP VSAM_SRC;
DESCRIPTION COBOL ../copybk/ACCOUNT AS account_file;
END GROUP;

--          TARGET DESCRIPTIONS

BEGIN GROUP VSAM_TGT;
DESCRIPTION COBOL ../copybk/ACCOUNT AS account_file;
END GROUP;

--          SOURCE DATASTORE (IP & Publisher name)

DATASTORE cdc://10.81.148.4:2626/vsmcdct/VSMTOKFK
OF VSAMCDC
AS CDCIN
DESCRIBED BY GROUP VSAM_SRC ACCEPT ALL;

--          TARGET DATASTORE(s) - Kafka and topic name

DATASTORE 'kafka:///MSKTutorialTopic/key'
OF JSON
AS CDCOUT
DESCRIBED BY GROUP VSAM_TGT FOR INSERT;

--          MAIN SECTION

PROCESS INTO
CDCOUT
SELECT
{
SETURL(CDCOUT, 'kafka:///MSKTutorialTopic/key')
REMAP(CDCIN, account_file, GET_RAW_RECORD(CDCIN, AFTER), GET_RAW_RECORD(CDCIN,
BEFORE))
REPLICATE(CDCOUT, account_file)
}
FROM CDCIN;

```

키 쌍 예제

다음은 JCL을 실행하여 키 쌍을 생성하는 방법의 예입니다.

```

//SQDUTIL EXEC PGM=SQDUTIL //SQDPUBL DD DSN=&USER..NACL.PUBLIC, //
DCB=(RECFM=FB,LRECL=80,BLKSIZE=21200), // DISP=(,CATLG,DELETE),UNIT=SYSDA, //
SPACE=(TRK,(1,1)) //SQDPKEY DD DSN=&USER..NACL.PRIVATE, //

```

```
DCB=(RECFM=FB,LRECL=80,BLKSIZE=21200), // DISP=(,CATLG,DELETE),UNIT=SYSDA, //  
SPACE=(TRK,(1,1)) //SQDPARMS DD keygen //SYSPPRINT DD SYSOUT= //SYSOUT DD SYSOUT=* //  
SQDLOG DD SYSOUT=* //*SQDLOG8 DD DUMMY
```

OpenText 마이크로 포커스 엔터프라이즈 서버 및 LRS X를 사용하여 AWS의 메인프레임 출력 관리를 현대화합니다. PageCenter

작성자: 슈밤 로이(AWS), 에이브러햄 론돈(마이크로 포커스), 가이 터커(리바이, 레이 앤 쇼프 주식회사)

환경: PoC 또는 파일럿	소스: IBM 메인프레임	대상: AWS
R 유형: 리플랫폼	워크로드: IBM	기술: 메인프레임, 마이그레이션, 현대화
AWS 서비스: AWS Managed Microsoft AD, Amazon EC2, Amazon FSx for Windows File Server, Amazon RDS, AWS Mainframe Modernization		

요약

메인프레임 출력 관리를 현대화하면 DevOps Amazon Web Services (AWS) 클라우드 네이티브 기술을 통해 비용을 절감하고, 레거시 시스템 유지 관리에 따른 기술적 부채를 줄이고, 복원력과 민첩성을 개선할 수 있습니다. 이 패턴은 AWS 클라우드에서 비즈니스에 중요한 메인프레임 출력 관리 워크로드를 현대화하는 방법을 보여줍니다. 이 패턴은 [OpenText 마이크로 포커스 엔터프라이즈 서버를](#) 현대화된 메인프레임 애플리케이션의 런타임으로 사용하고, Levi, Ray & Shoup, Inc. (LRS) VPSX/MFI (마이크로 포커스 인터페이스) 를 인쇄 서버로, LRS PageCenter X를 아카이브 서버로 사용합니다. LRS PageCenter X는 비즈니스 결과물을 보고, 인덱싱하고, 검색하고, 보관하고, 액세스 보안을 유지하기 위한 출력 관리 솔루션을 제공합니다.

이 패턴은 [리플랫폼](#) 메인프레임 현대화 접근 방식을 기반으로 합니다. Amazon Elastic Compute Cloud(Amazon EC2)의 [AWS Mainframe Modernization](#)를 통해 메인프레임 애플리케이션이 마이그레이션됩니다. 메인프레임 출력 관리 워크로드는 Amazon EC2로 마이그레이션되고, IBM Db2 for z/OS와 같은 메인프레임 데이터베이스는 Amazon Relational Database Service(RDS)로 마이그레이션됩니다. LRS Directory Integration Server(LRS/DIS)는 출력 관리 워크플로 인증 및 권한 부여를 위해 AWS Directory Service for Microsoft Active Directory와 함께 작동합니다.

사전 조건 및 제한 사항

사전 조건

- 활성 상태의 AWS 계정
- 메인프레임 출력 관리 워크로드.
- Micro Focus Enterprise Server에서 실행되는 메인프레임 애플리케이션을 재구축하고 제공하는 방법에 대한 기본 지식 OpenText 자세한 내용은 OpenText Micro Focus [설명서의 엔터프라이즈 서버 데이터시트를 참조하십시오](#).
- LRS 클라우드 프린팅 솔루션 및 개념에 대한 기본 지식. 자세한 내용은 LRS 설명서의 출력 현대화를 참조하세요.
- Micro Focus Enterprise Server 소프트웨어 및 라이선스. 자세한 내용은 OpenText [Micro Focus 영업 팀에 문의하십시오](#).
- LRS VPSX/MFI, LRS PageCenter X, LRS/큐, LRS/DIS 소프트웨어 및 라이선스 자세한 내용은 [Elastic에 문의](#)하세요. LRS 제품이 설치될 EC2 인스턴스의 호스트 이름을 제공해야 합니다.

참고: 메인프레임 출력 관리 워크로드의 구성 고려 사항에 대한 자세한 내용은 이 패턴의 [추가 정보](#) 섹션에 있는 고려 사항을 참조하세요.

제품 버전

- [OpenText 마이크로 포커스 엔터프라이즈 서버 8.0 이상](#)
- [LRS VPSX/MFI](#)
- [LRS PageCenter X V1R3 이상](#)

아키텍처

소스 기술 스택

- 운영 체제 – IBM z/OS
- 프로그래밍 언어 – 공통 비즈니스 지향 언어(COBOL), 직무 제어 언어(JCL) 및 고객 정보 제어 시스템(CICS)
- 데이터베이스 – IBM Db2 for z/OS, IBM Information Management System(IMS) 데이터베이스, Virtual Storage Access Method(VSAM)

- 보안 – Resource Access Control Facility(RACF), zCA Top Secret for z/OS, Access Control Facility 2(ACF2)
- 인쇄 및 아카이브 솔루션 – IBM 메인프레임 z/OS 출력 및 인쇄 제품(z/OS, LRS 및 CA Deliver용 IBM Infoprint Server) 및 아카이빙 솔루션(CA Deliver, ASG Mobius, 또는 CA Bundle)

소스 아키텍처

다음 다이어그램은 메인프레임 출력 관리 워크로드의 일반적인 현재 상태 아키텍처를 보여줍니다.

이 다이어그램은 다음 워크플로를 보여줍니다.

1. 사용자는 COBOL로 작성된 IBM CICS 애플리케이션을 기반으로 구축된 참여 시스템(SoE)에서 비즈니스 트랜잭션을 수행합니다.
2. SoE는 IBM Db2 for z/OS와 같은 system-of-records (SoR) 데이터베이스에 비즈니스 트랜잭션 데이터를 기록하는 메인프레임 서비스를 호출합니다.
3. SoR은 SoE의 비즈니스 데이터를 유지합니다.
4. 배치 작업 스케줄러는 인쇄 출력을 생성하는 일괄 작업을 시작합니다.
5. 배치 작업은 데이터베이스에서 데이터를 추출합니다. 비즈니스 요구 사항에 따라 데이터 형식을 지정한 다음 청구서, ID 카드 또는 대출 명세서와 같은 비즈니스 결과를 생성합니다. 마지막으로 배치 작업은 출력을 출력 관리로 라우팅하여 비즈니스 요구 사항에 따라 출력의 형식 지정, 게시 및 저장을 수행합니다.
6. 출력 관리는 배치 작업의 출력을 받습니다. 출력 관리는 출력을 인덱싱하고 정렬하여 PageCenter LRS X 솔루션 (이 패턴에서 설명함) 또는 CA View와 같은 출력 관리 시스템의 지정된 대상에 게시합니다.
7. 사용자는 출력을 확인 및 검색할 수 있습니다.

대상 기술 스택

- 운영 체제 – Amazon EC2에서 실행되는 Windows Server
- 컴퓨팅 – Amazon EC2
- 스토리지 – Amazon Elastic Block Store(Amazon EBS) 및 Amazon FSx for Windows File Server
- 프로그래밍 언어 – COBOL, JCL, CICS
- 데이터베이스 – Amazon RDS

- 보안 – AWS Managed Microsoft AD
- 인쇄 및 보관 — AWS 기반 LRS 인쇄 (VPSX) 및 보관 (PageCenterX) 솔루션
- 메인프레임 런타임 환경 — 마이크로 포커스 엔터프라이즈 서버 OpenText

대상 아키텍처

다음 다이어그램은 AWS 클라우드에 배포된 메인프레임 출력 관리 워크로드의 아키텍처를 보여줍니다.

이 다이어그램은 다음 워크플로를 보여줍니다.

1. 배치 작업 스케줄러는 일괄 작업을 시작하여 청구서, ID 카드 또는 대출 명세서와 같은 출력을 생성합니다.
2. 메인프레임 배치 작업 ([Amazon EC2로 재구성](#))은 [OpenText Micro Focus Enterprise Server](#) 런타임을 사용하여 애플리케이션 데이터베이스에서 데이터를 추출하고, 데이터에 비즈니스 로직을 적용하고, 데이터 형식을 지정합니다. 그런 다음 [Micro Focus 프린터 종료 모듈을 사용하여 출력 대상으로 데이터를 전송합니다](#) ([OpenText OpenText Micro Focus 설명서](#)).
3. 애플리케이션 데이터베이스(Amazon RDS에서 실행되는 SoR)는 인쇄 출력용 데이터를 보관합니다.
4. LRS VPSX/MFI 인쇄 솔루션은 Amazon EC2에 배포되며, 운영 데이터는 Amazon EBS에 저장됩니다. LRS VPSX/MFI는 TCP/IP 기반 LRS/큐 전송 에이전트를 사용하여 마이크로 포커스 JES 프린트 액시트 API를 통해 출력 데이터를 수집합니다. OpenText

LRS VPSX/MFI는 EBCDIC에서 ASCII로의 변환과 같은 데이터 사전 처리를 수행합니다. 또한 IBM Advanced Function Presentation(AFP) 및 Xerox Line Conditioned Data Stream(LCDS)과 같은 메인프레임 전용 데이터 스트림을 Printer Command Language(PCL) 및 PDF와 같은 보다 일반적인 보기 및 인쇄 데이터 스트림으로 변환하는 등 보다 복잡한 작업을 수행합니다.

LRS X의 유지 관리 기간 동안 LRS PageCenter VPSX/MFI는 출력 대기열을 유지하고 출력 대기열의 백업 역할을 합니다. LRS VPSX/MFI는 LRS/큐 프로토콜을 사용하여 LRS X에 연결하고 출력을 전송합니다. PageCenter LRS/Queue는 작업 준비 상태와 완료 상태를 교환하여 데이터가 전송되도록 합니다.

참고:

OpenText [Micro Focus Print Exit](#)에서 LRS/Queue로 전달되는 인쇄 데이터 및 LRS VPSX/MFI 지원 메인프레임 배치 메커니즘에 대한 자세한 내용은 추가 정보 섹션의 인쇄 데이터 캡처를 참조하십시오.

LRS VPSX/MFI는 프린터 플릿 수준에서 상태 확인을 수행할 수 있습니다. 자세한 내용은 이 패턴의 [추가 정보](#) 섹션에 있는 프린터 플릿 상태 확인을 참조하세요.

5. LRS PageCenter X 출력 관리 솔루션은 아마존 EC2에 배포되며, LRS X 출력 관리 솔루션은 Windows File Server용 Amazon FSx에 저장됩니다. LRS PageCenter X는 모든 사용자가 파일에 액세스할 수 있는 동시에 LRS PageCenter X로 가져온 모든 파일에 대한 중앙 보고서 관리 시스템을 제공합니다. 사용자는 특정 파일 내용을 보거나 여러 파일을 검색하여 일치하는 조건을 찾을 수 있습니다.

LRS/NetX 구성 요소는 LRS X 응용 프로그램 및 기타 LRS 응용 프로그램에 대한 공통 런타임 환경을 제공하는 다중 스레드 웹 응용 프로그램 서버입니다. PageCenter LRS/Web Connect 구성 요소는 사용자의 웹 서버에 설치되며, 웹 서버에서 LRS/NetX 웹 애플리케이션 서버로 연결되는 커넥터를 제공합니다.

6. LRS X는 파일 시스템 객체를 위한 스토리지를 제공합니다. PageCenter LRS PageCenter X의 운영 데이터는 Windows File Server용 Amazon FSx에 저장됩니다.
7. 출력 관리 인증 및 권한 부여는 LRS/DIS를 사용하는 AWS Managed Microsoft AD에서 수행합니다.

참고: 대상 솔루션은 일반적으로 IBM AFP 또는 Xerox LCDS와 같은 메인프레임 형식 언어를 수용하기 위해 애플리케이션을 변경할 필요가 없습니다.

AWS 인프라 아키텍처

다음 다이어그램은 메인프레임 출력 관리 워크로드를 위한 가용성이 높고 안전한 AWS 인프라 아키텍처를 보여줍니다.

이 다이어그램은 다음 워크플로를 보여줍니다.

1. 배치 스케줄러는 배치 프로세스를 시작하고 고가용성(HA)을 위해 여러 [가용 영역](#)에 걸쳐 Amazon EC2에 배포됩니다.

참고: 이 패턴은 배치 스케줄러 구현을 다루지 않습니다. 구현에 대한 자세한 내용은 스케줄러의 소프트웨어 공급업체 설명서를 참조하세요.

2. JCL 또는 COBOL과 같은 프로그래밍 언어로 작성된 메인프레임 배치 작업은 핵심 비즈니스 로직을 사용하여 청구서, ID 카드, 대출 명세서와 같은 인쇄 결과를 처리하고 생성합니다. 배치 작업은 HA를 위해 두 가용 영역에 걸쳐 Amazon EC2에 배포됩니다. OpenText 마이크로 포커스 프린트 엑시트 API를 사용하여 데이터 사전 처리를 위해 인쇄 출력을 LRS VPSX/MFI로 라우팅합니다.
3. LRS VPSX/MFI 프린트 서버는 Amazon EC2에서 HA를 위해 두 가용 영역(액티브-스탠바이 중복 쌍)에 배포됩니다. [Amazon EBS](#)를 운영 데이터 스토어로 사용합니다. Network Load Balancer는 LRS VPSX/MFI EC2 인스턴스에서 상태 확인을 수행합니다. 활성 인스턴스가 비정상 상태인 경우 로드 밸런서는 트래픽을 다른 가용 영역의 상시 대기 방식 인스턴스로 라우팅합니다. 인쇄 요청은 각 EC2 인스턴스의 LRS Job Queue에 로컬로 유지됩니다. 오류가 발생한 경우 실패한 인스턴스를 다시 시작해야 LRS 서비스가 인쇄 요청 처리를 재개할 수 있습니다.

참고: LRS VPSX/MFI는 프린터 플릿 수준에서도 상태 확인을 수행할 수 있습니다. 자세한 내용은 이 패턴의 [추가 정보](#) 섹션에 있는 프린터 플릿 상태 확인을 참조하세요.

4. LRS PageCenter X 출력 관리는 HA (액티브-스탠바이 중복 쌍)를 위한 두 개의 가용 영역에 걸쳐 Amazon EC2에 배포됩니다. 운영 데이터 스토어로 [Amazon FSx for Windows File Server](#)를 사용합니다. 활성 인스턴스가 비정상 상태인 경우 로드 밸런서는 LRS PageCenter X EC2 인스턴스의 상태 확인을 수행하고 트래픽을 다른 가용 영역의 대기 인스턴스로 라우팅합니다.
5. [네트워크 로드 밸런서는](#) LRS VPSX/MFI 서버를 LRS X와 통합하기 위한 DNS 이름을 제공합니다. PageCenter

참고: LRS X는 레이어 4 로드 밸런서를 지원합니다. PageCenter

6. LRS PageCenter X는 Windows File Server용 Amazon FSx를 HA용 두 가용 영역에 배포된 운영 데이터 스토어로 사용합니다. LRS PageCenter X는 외부 데이터베이스가 아닌 파일 공유에 있는 파일만 인식합니다.
7. [AWS Managed Microsoft AD](#)는 LRS/DIS와 함께 사용하여 출력 관리 워크플로 인증 및 권한 부여를 수행합니다. 자세한 내용은 [추가 정보](#) 섹션의 인쇄 출력 인증 및 권한 부여를 참조하세요.

도구

서비스

- [AWS Directory Service for Microsoft Active Directory](#)을 사용하면 디렉터리 인식 워크로드와 AWS 리소스가 AWS 클라우드에서 관리형 Active Directory를 사용할 수 있습니다.
- [Amazon Elastic Block Store\(Amazon EBS\)](#)는 Amazon Elastic Compute Cloud(Amazon EC2) 인스턴스에 사용할 수 있는 블록 스토리지 볼륨을 제공합니다.
- [Amazon Elastic Compute Cloud\(Amazon EC2\)](#)는 AWS 클라우드에서 확장 가능한 컴퓨팅 용량을 제공합니다. 필요한 만큼 가상 서버를 시작하고 빠르게 스케일 업하거나 스케일 다운할 수 있습니다.
- [Elastic Load Balancing\(ELB\)](#)은 들어오는 애플리케이션 또는 네트워크 트래픽을 여러 대상에 분산합니다. 예를 들어 하나 이상의 가용 영역에 있는 Amazon EC2 인스턴스, 컨테이너, IP 주소 전반적으로 트래픽을 분산할 수 있습니다. 이 패턴은 Network Load Balancer를 사용합니다.
- [Amazon FSx](#)는 업계 표준 연결 프로토콜을 지원하고 AWS 리전 전반에 걸쳐고가용성 및 복제를 제공하는 파일 시스템을 제공합니다. 이 패턴은 Amazon FSx for Windows File Server를 사용합니다.
- [Amazon Relational Database Service\(RDS\)](#)는 AWS 클라우드에서 관계형 데이터베이스를 설정, 운영 및 조정하는 데 도움이 됩니다.

기타 도구

- [LRS PageCenter X](#) 소프트웨어는 자동 인덱싱, 암호화 및 고급 검색 기능을 통해 사용자가 정보로부터 최대 가치를 얻을 수 있도록 지원하는 확장 가능한 문서 및 보고서 콘텐츠 관리 솔루션을 제공합니다.
- [LRS와 마이크로 포커스가 공동 개발한 LRS VPSX/MFI \(마이크로 포커스 인터페이스\)](#)는 OpenText 마이크로 포커스 엔터프라이즈 서버 JES 스톱에서 출력을 캡처하여 지정된 인쇄 대상으로 안정적으로 전달합니다. OpenText
- LRS/Queue는 TCP/IP 기반의 전송 에이전트입니다. LRS VPSX/MFI는 LRS/Queue를 사용하여 마이크로 포커스 JES 프린트 액시트 프로그래밍 인터페이스를 통해 인쇄 데이터를 수집하거나 캡처합니다. OpenText
- LRS Directory Integration Server(LRS/DIS)는 인쇄 워크플로 중 인증 및 권한 부여에 사용됩니다.
- [OpenText Micro Focus Enterprise Server](#)는 메인프레임 애플리케이션을 위한 애플리케이션 배포 환경입니다. 모든 버전의 Micro Focus Enterprise Developer를 사용하여 마이그레이션하거나 생성한 메인프레임 애플리케이션을 위한 런타임 환경을 제공합니다. OpenText

에픽

OpenText Micro Focus 런타임을 설정하고 메인프레임 배치 애플리케이션을 배포하십시오.

작업	설명	필요한 기술
런타임을 설정하고 데모 애플리케이션을 배포하세요.	<p>Amazon EC2에 OpenText Micro Focus 엔터프라이즈 서버를 설치하고 OpenText Micro Focus BankDemo 데모 애플리케이션을 배포하려면 AWS 메인프레임 현대화 사용 설명서의 지침을 따르십시오.</p> <p>BankDemo 애플리케이션은 인쇄 출력을 생성하고 시작하는 메인프레임 배치 애플리케이션입니다.</p>	클라우드 아키텍트

Amazon EC2에 LRS 인쇄 서버 설치

작업	설명	필요한 기술
Amazon EC2 인스턴스를 생성합니다.	<p>Amazon EC2 Windows 인스턴스를 시작하려면 Amazon EC2 설명서의 1단계: 인스턴스 시작의 지침을 따르세요. LRS 제품 라이선스에 사용한 것과 동일한 호스트 이름을 사용하세요.</p> <p>인스턴스는 LRS VPSX/MFI에 대한 다음 하드웨어 및 소프트웨어 요구 사항을 충족해야 합니다.</p> <ul style="list-style-type: none"> • CPU - 듀얼 코어 	클라우드 아키텍트

작업	설명	필요한 기술
	<ul style="list-style-type: none"> • 램 – 16GB • 드라이브 – 500GB • 최소 EC2 인스턴스 – m5.xlarge • OS – Windows • 소프트웨어 – Internet Information Services(IIS) 또는 Apache <p>참고: 위의 하드웨어 및 소프트웨어 요구 사항은 소형 프린터 제품군(약 500~1000대)을 대상으로 합니다. 전체 요구 사항을 알아보려면 LRS 및 AWS 담당자에게 문의하세요.</p> <ol style="list-style-type: none"> 1. Windows 인스턴스를 생성할 때 EC2 호스트 이름이 LRS 제품 라이선스에 사용된 호스트 이름과 동일한지 확인하세요. 2. Amazon EC2 설명서의 2단계: 인스턴스에 연결 지침에 따라 EC2 인스턴스에 연결합니다. 3. Windows 시작 메뉴에서 서버 관리자를 찾아 엽니다. 4. Server Manager에서 대시보드, 빠른 시작, 역할 및 기능 추가를 선택한 다음 서버 역할을 선택합니다. 	

작업	설명	필요한 기술
	<ol style="list-style-type: none"> 5. 서버 역할에서 WebServer (IIS) 를 선택한 다음 응용 프로그램 개발을 선택합니다. 6. 애플리케이션 개발에서 CGI 확인란에 체크합니다. 7. CGI를 설치하려면 Windows Server Manger 역할 및 기능 추가 마법사의 지침을 따르세요. 8. LRS/Queue 통신을 위해 EC2 인스턴스의 Windows 방화벽에 포트 5500을 엽니다. 	

작업	설명	필요한 기술
<p>EC2 인스턴스에 LRS VPSX/MFI를 설치합니다.</p>	<ol style="list-style-type: none"> 1. EC2 인스턴스에 연결합니다. 2. 수신한 LRS 이메일 메시지에서 제품 다운로드 페이지 링크를 여세요. <p>참고: LRS 제품은 전자 파일 전송(EFT)을 통해 배포됩니다.</p> <ol style="list-style-type: none"> 3. LRS VPSX/MFI를 다운로드하고 파일의 압축을 풉니다 (기본 폴더:c:\LRS). 4. LRS VPSX/MFI를 설치하려면 압축을 푼 폴더에서 LRS 제품 설치 프로그램을 실행합니다. 5. 기능 선택 메뉴에서 VPSX® Server를 선택하고 다음을 선택하여 설치 프로세스를 시작합니다. 설치가 완료되면 성공 메시지가 나타납니다. 	<p>클라우드 아키텍트</p>

작업	설명	필요한 기술
LRS/Queue를 설치합니다.	<ol style="list-style-type: none"> 1. OpenText Micro Focus 엔터프라이즈 서버 EC2 인스턴스에 연결합니다. 2. 수신한 LRS 이메일 메시지에서 LRS 제품 다운로드 페이지 링크를 열고 LRS/Queue를 다운로드한 다음 파일의 압축을 풉니다. 3. 파일을 다운로드한 위치로 이동한 다음 LRS 제품 설치 프로그램을 실행하여 LRS/Queue를 설치합니다. 4. LRS 제품 설치 프로그램의 지침에 따라 설치 프로세스를 완료하세요. 	클라우드 아키텍트

작업	설명	필요한 기술
LRS/DIS를 설치합니다.	<p>LRS/DIS 제품은 LRS VPSX 설치에 포함되는 경우가 많습니다. 그러나 LRS/DIS가 LRS VPSX와 함께 설치되지 않은 경우 다음 단계를 통해 설치하세요.</p> <ol style="list-style-type: none"> 1. LRS VPSX/MFI EC2 인스턴스에 연결합니다. 2. 수신한 LRS 이메일 메시지에서 LRS 제품 다운로드 페이지 링크를 열고 LRS/DIS를 다운로드한 다음 파일의 압축을 풉니다. 3. 파일을 다운로드한 위치로 이동한 다음 LRS 제품 설치 프로그램을 시작합니다. 4. LRS 제품 설치 프로그램에서 LRS 기타 도구를 확장하고 LRS DIS를 선택한 후 다음을 선택합니다. 5. LRS 제품 설치 프로그램의 나머지 지침을 따라 설치 프로세스를 완료하세요. 	클라우드 아키텍트

작업	설명	필요한 기술
대상 그룹을 생성합니다.	<p>Network Load Balancer에 대한 대상 그룹 생성의 지침에 따라 대상 그룹을 생성합니다. 대상 그룹을 생성할 때 LRS VPSX/MFI EC2 인스턴스를 대상으로 등록하세요.</p> <ol style="list-style-type: none"> 1. 그룹 세부 정보 지정 페이지의 대상 유형 선택에서 인스턴스를 선택합니다. 2. 프로토콜에서 TCP를 선택합니다. 3. 포트는 5500을 선택합니다. 4. 대상 등록 페이지의 사용 가능한 인스턴스 섹션에서 LRS VPSX/MFI EC2 인스턴스를 선택합니다. 	클라우드 아키텍트

작업	설명	필요한 기술
Network Load Balancer를 생성합니다.	<p>Network Load Balancer를 만들려면 Elastic Load Balancing 설명서의 지침을 따르세요.</p> <p>Network Load Balancer는 OpenText 마이크로 포커스 엔터프라이즈 서버의 트래픽을 LRS VPSX/MFI EC2 인스턴스로 라우팅합니다.</p> <p>Network Load Balancer를 생성할 때 리스너 및 라우팅 페이지에서 다음 값을 선택하세요.</p> <ol style="list-style-type: none"> 1. 프로토콜에서 TCP를 선택합니다. 2. 포트는 5500을 선택합니다. 3. 기본 작업에서 앞서 생성한 대상 그룹에 대해 전달을 선택합니다. 	클라우드 아키텍트

OpenText 마이크로 포커스 엔터프라이즈 서버를 LRS/큐 및 LRS VPSX/MFI와 통합합니다.

작업	설명	필요한 기술
LRS/Queue 통합을 위해 Micro Focus Enterprise Server를 구성하세요.	<ol style="list-style-type: none"> 1. Amazon EC2 설명서의 지침에 따라 OpenText Micro Focus 엔터프라이즈 서버 EC2 인스턴스에 연결합니다. 2. Windows 시작 메뉴에서 OpenText 마이크로 포커스 엔터프라이즈 서버 관리 UI를 엽니다. 	클라우드 아키텍트

작업	설명	필요한 기술
	<p>3. 메뉴 모음에서 네이티브를 선택합니다.</p> <p>4. 탐색 창에서 디렉토리 서버를 선택한 다음 해당 엔터프라이즈 서버 리전의 BANKDEMO를 선택합니다.</p> <p>5. 왼쪽 탐색 창의 일반에서 추가 섹션으로 스크롤하여 LRSQ를 가리키도록 환경 변수(LRSQ_ADDRESS , LRSQ_PORT , LRSQ_COMMAND)를 구성합니다.</p> <ul style="list-style-type: none"> • LRSQ_ADDRESS의 경우 이전에 생성한 Network Load Balancer의 IP 주소 또는 DNS 이름을 입력합니다. • LRSQ_PORT의 경우 VPSX LRSQ 리스너 포트 (5500)를 입력합니다. • LRSQ_COMMAND의 경우 LRSQ 실행 파일의 경로 위치를 입력합니다. <p>참고: LRS는 현재 DNS 이름에 대해 최대 50자를 지원합니다. DNS 이름이 50자를 초과하는 경우 Network Load Balancer의 IP 주소를 대안으로 사용할 수 있습니다.</p>	

작업	설명	필요한 기술
<p>LRS VPSX/MFI 통합을 위해 OpenText 마이크로 포커스 엔터프라이즈 서버를 구성합니다.</p>	<ol style="list-style-type: none"> 1. LRS VPSX/MFI 설치 프로그램의 VPSX_MFI_R2 폴더를 C\BANKDEMO\print 의 Micro Focus Enterprise Server 위치로 복사하세요. 2. Amazon EC2 설명서의 지침에 따라 Micro Focus Enterprise Server EC2 인스턴스에 연결합니다. 3. Windows 시작 메뉴에서 Micro Focus Enterprise Server 관리 UI를 엽니다. 4. 메뉴 모음에서 네이티브를 선택합니다. 5. 탐색 창에서 디렉토리 서버를 선택한 다음 BANKDEMO를 선택합니다. 6. BANKDEMO에서 JES를 선택합니다. 7. JES 프로그램 경로에 C\BANKDEMO\print 의 DLL(VPSX_MFI_R2) 경로를 추가합니다. 	<p>클라우드 아키텍트</p>

인쇄 대기열 및 인쇄 사용자 설정

작업	설명	필요한 기술
<p>OpenText 마이크로 포커스 프린트 액시트 모듈을 마이크로 포커스 엔터프라이즈 서버 배치 프린터 서버 실행 프로세스와 연결합니다.</p>	<ol style="list-style-type: none"> 1. Amazon EC2 설명서의 지침에 따라 OpenText Micro Focus 엔터프라이즈 서버 EC2 인스턴스에 연결합니다. 	<p>클라우드 아키텍트</p>

작업	설명	필요한 기술
	<ol style="list-style-type: none"> 2. Windows 시작 메뉴에서 OpenText 마이크로 포커스 엔터프라이즈 서버 관리 UI 를 엽니다. 3. 메뉴 모음에서 네이티브을 선택합니다. 4. 탐색 창에서 디렉토리 서버를 선택한 다음 BANKDEMO를 선택합니다. 5. BANKDEMO에서 JES를 선택하고 아래로 스크롤하여 프린터를 선택합니다. 6. 프린터에서 OpenText 마이크로 포커스 프린트 종료 모듈 (LRSPRTE6 for Batch) 을 OpenText 마이크로 포커스 엔터프라이즈 서버 배치 프린터 서버 실행 프로세스 (SEP) 에 연결합니다. 이를 통해 인쇄 출력을 LRS VPSX/MFI로 라우팅할 수 있습니다. <p>구성에 대한 자세한 내용은 OpenText Micro Focus 설명서의 Exit 사용을 참조하십시오.</p>	

작업	설명	필요한 기술
LRS VPSX/MFI에서 인쇄 출력 대기열을 생성하고 이를 LRS X와 통합하십시오. PageCenter	<ol style="list-style-type: none"> 1. LRS VPSX/MFI EC2 인스턴스에 연결합니다. 2. Windows 시작 메뉴에서 VPSX 웹 인터페이스를 엽니다. 3. 탐색 창에서 프린터를 선택합니다. 4. 추가를 선택한 다음 프린터 추가를 선택합니다. 5. 프린터 구성 페이지에서 프린터 이름에 Local을 입력합니다. 6. VPSX ID의 경우 VPS1을 입력합니다. 7. 그러려면 TCP/PIP/LRSQ를 선택하십시오. CommType 8. 호스트/IP 주소에는 LRS X EC2 인스턴스 앞에 있는 Network Load Balancer의 IP 주소를 입력합니다. PageCenter 9. 원격 포트의 경우 5800을 입력합니다. 10. 원격 대기열의 경우 출력을 저장할 LRS PageCenter X 문서 폴더의 이름을 입력합니다. 11. 추가를 선택합니다. 	클라우드 아키텍트

작업	설명	필요한 기술
LRS VPSX/MFI에서 프린트 사용자를 생성하세요.	<ol style="list-style-type: none"> 1. LRS VPSX/MFI EC2 인스턴스에 연결합니다. 2. Windows 시작 메뉴에서 VPSX 웹 인터페이스를 엽니다. 3. 탐색 창에서 보안을 선택한 다음 사용자를 선택합니다. 4. 사용자 이름 열에서 관리자를 선택한 다음 복사를 선택합니다. 5. 사용자 프로필 관리 창에서 사용자 이름에 사용자 이름 (예: PrintUser) 을 입력합니다. 6. 설명에 간단한 설명(예: 테스트 인쇄용 사용자)을 입력합니다. 7. 업데이트를 선택합니다. 그러면 인쇄 사용자 (예: PrintUser) 가 생성됩니다. 8. 탐색 창의 사용자에서 생성한 새 사용자를 선택합니다. 9. 명령 메뉴에서 보안을 선택합니다. 10. 보안 규칙 페이지에서 해당하는 프린터 보안 및 작업 보안 옵션을 모두 선택한 다음 저장을 선택합니다. 11. 새 인쇄 사용자를 관리자 그룹에 추가하려면 탐색 창의 보안을 선택한 다음 구성을 선택합니다. 	클라우드 아키텍트

작업	설명	필요한 기술
	12보안 구성 창에서 관리자 열에 새 인쇄 사용자를 추가합니다.	

아마존 EC2에 LRS PageCenter X 서버 설정

작업	설명	필요한 기술
Amazon EC2 인스턴스를 생성합니다.	<p>Amazon EC2 설명서의 1단계: 인스턴스 시작의 지침에 따라 Amazon EC2 Windows 인스턴스를 시작합니다. LRS 제품 라이선스에 사용한 것과 동일한 호스트 이름을 사용하세요.</p> <p>인스턴스는 PageCenter LRS X에 대한 다음 하드웨어 및 소프트웨어 요구 사항을 충족해야 합니다.</p> <ul style="list-style-type: none"> • CPU – 듀얼 코어 • 램 – 16GB • 드라이브 – 500GB • 최소 EC2 인스턴스 – m5.xlarge • OS - Windows • 소프트웨어 – IIS 또는 Apache <p>참고: 위의 하드웨어 및 소프트웨어 요구 사항은 소형 프린터 플릿(약 500~1000대)을 대상으로 합니다. 전체 요구 사항을 알</p>	클라우드 아키텍트

작업	설명	필요한 기술
	<p>아보려면 LRS 및 AWS 담당자에게 문의하세요.</p> <ol style="list-style-type: none"> 1. Windows 인스턴스를 생성할 때 EC2 호스트 이름이 LRS 제품 라이선스에 사용된 호스트 이름과 동일한지 확인하세요. 2. Amazon EC2 설명서의 지침에 따라 EC2 인스턴스에 연결합니다. 3. Windows 시작 메뉴에서 서버 관리자를 찾아 엽니다. 4. Server Manager에서 대시보드, 빠른 시작, 역할 및 기능 추가를 선택한 다음 서버 역할을 선택합니다. 5. 서버 역할에서 WebServer (IIS) 를 선택한 다음 애플리케이션 개발을 선택합니다. 6. 애플리케이션 개발에서 CGI 확인란에 체크합니다. 7. CGI를 설치하려면 Windows Server Manger 역할 및 기능 추가 마법사의 지침을 따르세요. 8. EC2 인스턴스의 Windows 방화벽에서 인바운드 TCP/IP 트래픽을 위한 포트 5800을 엽니다. LRS VPSX는 5800 포트의 TCP/LRSQ 프로토콜을 사용하여 LRS X와 통신합니다. PageCenter 	

작업	설명	필요한 기술
<p>EC2 인스턴스에 LRS PageCenter X를 설치합니다.</p>	<ol style="list-style-type: none"> 1. EC2 인스턴스에 연결합니다. 2. 수신한 LRS 이메일 메시지에서 제품 다운로드 페이지 링크를 여세요. <p>참고: LRS 제품은 전자 파일 전송(EFT)을 통해 배포됩니다.</p> <ol style="list-style-type: none"> 3. LRS PageCenter X를 다운로드하고 파일의 압축을 풉니다 (기본 폴더: c:\LRS) 4. LRS PageCenter X를 설치하려면 압축을 푼 폴더에서 LRS 제품 설치 프로그램을 실행합니다. 5. 기능 선택 메뉴에서 PageCenterX를 선택하고 다음을 선택하여 설치 프로세스를 시작합니다. 설치가 완료되면 성공 메시지가 나타납니다. 	<p>클라우드 아키텍트</p>

작업	설명	필요한 기술
LRS/DIS를 설치합니다.	<p>LRS/DIS 제품은 LRS VPSX 설치에 포함되는 경우가 많습니다. 그러나 LRS/DIS가 LRS VPSX와 함께 설치되지 않은 경우 다음 단계를 통해 설치하세요.</p> <ol style="list-style-type: none"> 1. LRS PageCenter X EC2 인스턴스에 연결합니다. 2. 수신한 LRS 이메일에서 LRS 제품 다운로드 페이지 링크를 열고 LRS/DIS를 다운로드한 다음 파일의 압축을 풉니다. 3. 파일을 다운로드한 위치로 이동한 다음 LRS 제품 설치 프로그램을 시작합니다. 4. LRS 제품 설치 프로그램에서 LRS 기타 도구를 확장하고 LRS DIS를 선택한 후 다음을 선택합니다. 5. LRS 제품 설치 프로그램의 나머지 지침을 따라 설치 프로세스를 완료하세요. 	클라우드 아키텍트

작업	설명	필요한 기술
대상 그룹을 생성합니다.	<p>Network Load Balancer에 대한 대상 그룹 생성의 지침에 따라 대상 그룹을 생성합니다. 대상 그룹을 생성할 때 LRS PageCenter X EC2 인스턴스를 대상으로 등록하십시오.</p> <ol style="list-style-type: none"> 1. 그룹 세부 정보 지정 페이지의 대상 유형 선택에서 인스턴스를 선택합니다. 2. 프로토콜에서 TCP를 선택합니다. 3. 포트는 5800을 선택합니다. 4. 대상 등록 페이지의 사용 가능한 인스턴스 섹션에서 LRS PageCenter X EC2 인스턴스를 선택합니다. 	클라우드 아키텍트

작업	설명	필요한 기술
Network Load Balancer를 생성합니다.	<p>Network Load Balancer를 만들려면 Elastic Load Balancing 설명서의 지침을 따르세요.</p> <p>네트워크 로드 밸런서는 LRS VPSX/MFI의 트래픽을 LRS X EC2 인스턴스로 라우팅합니다. PageCenter</p> <p>Network Load Balancer를 생성할 때 리스너 및 라우팅 페이지에서 다음 값을 선택하세요.</p> <ol style="list-style-type: none"> 1. 프로토콜에서 TCP를 선택합니다. 2. 포트는 5800을 선택합니다. 3. 기본 작업에서 앞서 생성한 대상 그룹에 대해 전달을 선택합니다. 	클라우드 아키텍트

LRS PageCenter X에서 출력 관리 기능을 설정합니다.

작업	설명	필요한 기술
LRS X에서 가져오기 기능을 활성화합니다. PageCenter	<p>LRS PageCenter X 가져오기 기능을 사용하면 작업 이름 또는 양식 ID와 같은 기준에 따라 LRS PageCenter X에 도착하는 출력을 인식할 수 있습니다. 그런 다음 출력을 LRS X의 특정 폴더로 라우팅할 수 있습니다. PageCenter</p> <p>가져오기 함수를 활성화하려면 다음을 수행합니다.</p>	클라우드 아키텍트

작업	설명	필요한 기술
	<ol style="list-style-type: none"> 1. Amazon EC2 설명서의 지침에 따라 LRS PageCenter X EC2 인스턴스에 연결합니다. 2. Windows 시작 메뉴에서 PCX 웹 인터페이스를 엽니다. 3. 폴더 탐색기에서 관리자를 선택합니다. 4. 구성 페이지에서 고급, 파라미터 가져오기를 선택합니다. 5. 파라미터 가져오기 섹션에서 고급 가져오기 확인란에 체크합니다. 6. 변경 내용을 적용하려면 업데이트를 선택합니다. 	

작업	설명	필요한 기술
문서 보존 정책을 구성합니다.	<p>LRS PageCenter X는 문서 보존 정책을 사용하여 LRS X에 문서를 보관할 기간을 결정합니다. PageCenter</p> <p>문서 보존 정책을 구성하려면 다음을 수행합니다.</p> <ol style="list-style-type: none"> 1. LRS PageCenter X EC2 인스턴스에 연결합니다. 2. Windows 시작 메뉴에서 PCX 웹 인터페이스를 엽니다. 3. 폴더 탐색기에서 관리자를 선택합니다. 4. 관리 페이지에서 아카이브 그룹 목록/일반 관리자를 선택한 다음 보존 정책을 선택합니다. 5. 보존 정책 섹션에서 추가를 선택하여 보존 정책을 생성합니다. 6. 보존 정책 정보 페이지에서 보존 정책 이름, 설명, 문서 보존 기간을 입력합니다. 7. 확인을 선택하여 변경 내용을 저장하고 정책을 생성합니다. 	클라우드 아키텍트

작업	설명	필요한 기술
<p>출력 문서를 LRS X의 특정 폴더로 라우팅하는 규칙을 생성합니다. PageCenter</p>	<p>LRS PageCenter X에서 대상은 보고서 정의에 의해 이 대상이 호출될 때 출력이 전송될 폴더 경로를 결정합니다. 이 예제에서는 보고서 정의의 양식 ID 폴더를 기반으로 폴더를 만들고 그 결과를 해당 폴더에 저장합니다.</p> <ol style="list-style-type: none"> 1. LRS PageCenter X EC2 인스턴스에 연결합니다. 2. Windows 시작 메뉴에서 PCX 웹 인터페이스를 엽니다. 3. 폴더 탐색기에서 관리자, 고급 가져오기, 대상을 선택합니다. 4. 대상 섹션에서 추가를 선택하여 대상 유지 관리 양식을 엽니다. 5. 대상 유지 관리 양식에 다음 값을 입력합니다. <ul style="list-style-type: none"> • 대상 이름 - 양식 • 설명 - 대상 설명(예: 양식 기반 폴더 구조) • 대상 유형 - 폴더 • 폴더 매개 변수 — 가져오기 폴더 경로 (문서가 도착할 때 PageCenter X에 생성되는 폴더 경로. 예를 들어, 경로를 /Test/&FORM/&IMPORTDATE/&IMPORTTIME 따라 기 	<p>클라우드 아키텍트</p>

작업	설명	필요한 기술
	<p>본 Test 폴더, Form-Id 이름을 기반으로 한 하위 폴더, 가져오기 날짜를 기준으로 하위 폴더 STD, 가져오기 시간을 기준으로 하위 폴더가 생성됨)</p> <ul style="list-style-type: none"> • 문서 이름 - 문서가 폴더에 저장될 때 문서에 할당되는 동적 이름입니다. <p>6. 드롭다운 목록에서 보존 정책을 선택합니다. 예를 들어 문서를 1년 동안 보존하려면 Year1을 선택합니다.</p> <p>7. 확인을 선택하여 변경 사항을 저장합니다.</p>	

작업	설명	필요한 기술
작업 정의를 생성합니다.	<ol style="list-style-type: none"> 1. LRS PageCenter X EC2 인스턴스에 연결합니다. 2. Windows 시작 메뉴에서 PCX 웹 인터페이스를 엽니다. 3. 폴더 탐색기에서 관리, 고급 가져오기, 보고서 정의를 선택한 다음 추가를 선택합니다. 4. 보고서 정의 유지 관리 페이지의 일반 탭에 보고서 정의 이름을 입력합니다. 5. 일반 탭의 필드에서 작업 이름, 양식, 클래스, 작성자 등의 선택 기준을 지정할 수 있습니다. 예를 들어, MFIDEMO라는 작업 이름을 입력할 수 있습니다. 작업 이름 값은 인쇄 출력을 생성할 배치 작업의 이름입니다. 6. 대상 탭의 사용 가능한 대상에서 이전에 만든 대상(양식)을 선택합니다. 7. 추가를 선택하여 양식 대상을 지정된 대상으로 추가합니다. <p>참고: 이 예에는 MFIDEMO에서 생성되어 LRS PageCenter X로 라우팅되는 출력이 대상 정의에 정의된 폴더 구조에 저장되는 보</p>	클라우드 아키텍트

작업	설명	필요한 기술
	고서 정의가 포함되어 있습니다.	

출력 관리를 위한 인증 및 권한 설정

작업	설명	필요한 기술
사용자 및 그룹을 사용하여 AWS Managed Microsoft AD 도메인을 생성합니다.	<ol style="list-style-type: none"> 1. AWS Managed Microsoft AD에 디렉터리를 생성하려면 AWS Managed Microsoft AD 디렉터리 생성의 지침을 따르세요. 2. EC2 인스턴스(Active Directory 관리자)를 배포하고 Active Directory 도구를 설치하여 AWS Managed Microsoft AD를 관리하려면 3단계: AWS Managed Microsoft AD를 관리하기 위한 EC2 인스턴스 배포의 지침을 따르세요. 3. EC2 인스턴스에 연결하려면 Amazon EC2 설명서의 지침을 따릅니다. 참고: EC2 인스턴스에 연결할 때 Windows 보안 창에 1 단계에서 생성한 디렉터리의 관리자 보안 인증 정보를 입력합니다. 4. 로그인 후 시작 메뉴의 Windows 관리 도구 아래에서 Active Directory 사용자 및 컴퓨터를 선택합니다. 	클라우드 아키텍트

작업	설명	필요한 기술
	5. Active Directory 도메인에서 인쇄 사용자를 생성하려면 사용자 생성 의 지침을 따르세요.	
EC2 인스턴스는 AWS Managed Microsoft AD 도메인에 조인됩니다.	LRS VPSX/MFI 및 LRS X PageCenter EC2 인스턴스를 AWS 관리형 Microsoft AD 도메인에 자동으로 (AWS 지식 센터 설명서) 또는 수동으로 (AWS 디렉토리 서비스 설명서) 가입하십시오.	클라우드 아키텍트
LRS X EC2 인스턴스용 LRS/DIS를 AWS 관리형 마이크로소프트 AD와 구성하고 통합합니다 PageCenter.	<ol style="list-style-type: none"> 1. LRS PageCenter X EC2 인스턴스에 연결합니다. 2. Windows 시작 메뉴에서 PCX 웹 인터페이스를 엽니다. 3. 폴더 탐색기에서 관리자를 선택합니다. 4. 구성 페이지의 보안 파라미터 섹션에서 보안 유형에 LRS/DIS를 선택합니다. 5. 보안 파라미터 섹션의 나머지 옵션에 원하는 설정을 입력합니다. 6. Windows 시작 메뉴에서 PageCenterX 폴더를 열고 [서버 시작] 을 선택한 다음 [서버 중지] 를 선택합니다. 7. Active Directory 사용자 이름과 암호를 사용하여 LRS PageCenter X에 로그인합니다. 	클라우드 아키텍트

작업	설명	필요한 기술
<p>LRS VPSX에서 LRS X로 출력을 가져오도록 가져오기 그룹을 구성하십시오. PageCenter</p>	<ol style="list-style-type: none"> 1. LRS PageCenter X EC2 인스턴스에 연결합니다. 2. Windows 시작 메뉴에서 PCX 웹 인터페이스를 엽니다. 3. 폴더 탐색기에서 관리자, 보안 관리자, 그룹을 선택합니다. 4. 그룹 섹션에서 추가를 선택하여 그룹 기본 설정 양식을 엽니다. 5. 그룹 기본 설정 양식에서 그룹 이름 및 설명에 값을 입력합니다. 6. 일반 옵션을 확장한 다음 가져오기 확인란에 체크합니다. 7. 확인을 선택하여 변경 사항을 저장합니다. 	<p>클라우드 아키텍트</p>
<p>가져오기 그룹에 보안 규칙을 추가합니다.</p>	<ol style="list-style-type: none"> 1. 가져오기 그룹에 대한 컨텍스트 메뉴를 엽니다(마우스 오른쪽 버튼 클릭). 2. 고급을 선택한 다음 보안을 선택합니다. 3. 보안 섹션에서 가져오기를 선택하고 하위 폴더 확인란에 체크합니다. 4. 변경 사항을 저장하려면 적용을 선택합니다. 	<p>클라우드 아키텍트</p>

작업	설명	필요한 기술
<p>LRS X에서 사용자를 생성하여 LRS PageCenter VPSX/MFI에서 출력 가져오기를 수행할 수 있습니다.</p>	<p>LRS PageCenter X에서 출력 가져오기를 수행할 사용자를 생성할 때 사용자 이름은 LRS VPSX/MFI에 있는 인쇄 출력 대기열의 VPSX ID와 동일해야 합니다. 이 예시에서의 VPSX ID는 VPS1입니다.</p> <ol style="list-style-type: none"> 1. LRS PageCenter X EC2 인스턴스에 연결합니다. 2. Windows 시작 메뉴에서 PCX 웹 인터페이스를 엽니다. 3. 폴더 탐색기에서 관리자, 보안 관리자, 사용자를 선택합니다. 4. 추가를 선택하여 사용자 프로필 유지 관리 양식을 엽니다. 5. 사용자 프로필 유지 관리에서 사용자 이름에 VPS1을 입력합니다. 	<p>클라우드 아키텍트</p>

작업	설명	필요한 기술
<p>LRS PageCenter X 가져오기 사용자를 가져오기 전용 그룹에 추가합니다.</p>	<p>LRS VPSX에서 LRS X로 문서를 가져오는 데 필요한 권한을 제공하려면 다음과 같이 PageCenter 하십시오.</p> <ol style="list-style-type: none"> 1. LRS PageCenter X EC2 인스턴스에 연결합니다. 2. Windows 시작 메뉴에서 PCX 웹 인터페이스를 엽니다. 3. 폴더 탐색기에서 관리자, 보안 관리자, 그룹을 선택합니다. 4. 그룹 섹션에서 마우스 오른쪽 버튼 클릭으로 가져오기 전용 그룹의 컨텍스트 메뉴를 연 다음 고급, 보안을 선택합니다. 5. 폴더 보안 레코드 (Import Only) 페이지에서 사용자 탭을 선택합니다. 6. 사용자 탭 이름의 드롭다운 목록에서 사용자 VPS1을 선택하고 적용을 선택합니다. 	<p>클라우드 아키텍트</p>

작업	설명	필요한 기술
LRS/DIS와 LRS VPSX/MFI EC2 인스턴스용 AWS Managed Microsoft AD를 구성합니다.	<ol style="list-style-type: none"> 1. LRS VPSX/MFI EC2 인스턴스에 연결합니다. 2. Windows 시작 메뉴에서 VPSX 웹 인터페이스를 엽니다. 3. 탐색 창에서 보안을 선택한 다음 구성을 선택합니다. 4. 보안 구성 페이지의 보안 파라미터 섹션에서 보안 유형에 대해 LRS/DIS(외부)를 선택합니다. 5. 보안 파라미터 섹션의 나머지 옵션에 원하는 설정을 입력합니다. 6. Windows 시작 메뉴에서 LRS 출력 관리 폴더를 열고 서버 시작을 선택한 다음 서버 중지를 선택합니다. 7. Active Directory 사용자 이름 및 암호를 사용하여 LRS VPSX/MFI에 로그인합니다. 	클라우드 아키텍트

Windows File Server용 Amazon FSx를 LRS X용 운영 데이터 스토어로 구성합니다. PageCenter

작업	설명	필요한 기술
LRS X용 파일 시스템을 생성하십시오. PageCenter	다중 AZ 환경에서 Windows File Server용 Amazon FSx를 LRS PageCenter X용 운영 데이터 스토어로 사용하려면 1단계: 파일 시스템 생성의 지침을 따르십시오.	클라우드 아키텍트

작업	설명	필요한 기술
<p>파일 공유를 PageCenter LRS X EC2 인스턴스에 매핑합니다.</p>	<p>이전 단계에서 생성한 파일 공유를 LRS PageCenter X EC2 인스턴스에 매핑하려면 2단계: Windows Server를 실행하는 EC2 인스턴스에 파일 공유 매핑의 지침을 따르십시오.</p>	<p>클라우드 아키텍트</p>
<p>LRS PageCenter X 제어 디렉터리 및 마스터 폴더 디렉터리를 Amazon FSx 네트워크 공유 드라이브에 매핑합니다.</p>	<ol style="list-style-type: none"> 1. Amazon EC2 설명서의 지침에 따라 LRS PageCenter X EC2 인스턴스에 연결합니다. 2. Windows 시작 메뉴에서 PCX 웹 인터페이스를 엽니다. 3. 폴더 탐색기에서 관리, 구성을 선택합니다. 4. 구성 페이지에서 디렉터리를 선택한 다음 디렉터리 제어를 선택합니다. 5. 디렉터리 제어에 \\FSx file share DNS name \share\cnt1 을 입력합니다. 6. 마스터 폴더 디렉터리에 \FSx file share DNS name\share\mstr 를 입력합니다. 	<p>클라우드 아키텍트</p>

출력 관리 워크플로 테스트

작업	설명	필요한 기술
<p>Micro Focus 앱에서 일괄 인쇄 요청을 시작하십시오. OpenText BankDemo</p>	<ol style="list-style-type: none"> 1. OpenText 마이크로 포커스 엔터프라이즈 서버 EC2 인스턴스에서 3270 터미널 에뮬레이터를 엽니다. 2. 명령을 실행하여 BankDemo 앱에 연결합니다connect 127.0.0.1:9278 . 3. BankDemo 명령줄 인터페이스의 사용자 ID에 B0001 를 입력합니다. 비밀번호에는 비어 있지 않은 키를 입력합니다. 4. 인쇄된 명세서 요청 옵션의 경우 빈 줄에 X를 입력합니다. 5. 명세서 전송 섹션의 메일에 Y를 입력한 다음 F10을 누릅니다. 	<p>테스트 엔지니어</p>
<p>LRS PageCenter X에서 인쇄 출력을 확인합니다.</p>	<ol style="list-style-type: none"> 1. Amazon EC2 설명서의 지침에 따라 LRS PageCenter X EC2 인스턴스에 연결합니다. 2. Windows 시작 메뉴에서 PCX 웹 인터페이스를 엽니다. 3. 탐색 창에서 테스트 폴더를 열고 STD 폴더를 연 다음 08-03-2023 (MM-DD-YY YY)와 같은 작업 실행 날짜가 있는 폴더를 엽니다. 	<p>테스트 엔지니어</p>

작업	설명	필요한 기술
	<p>참고: 이는 스토리에 정의된 것과 동일한 폴더 구조입니다. 출력 문서를 LRS X의 특정 폴더로 라우팅하는 규칙을 생성하십시오.</p> <p>PageCenter</p> <p>4. formtest-STD.txt 파일을 엽니다.</p> <p>이제 계좌 번호, 설명, 날짜, 금액 및 잔액 열이 포함된 계좌 명세서의 인쇄 출력을 볼 수 있습니다. 예제를 확인하려면 이 패턴에 대한 batch_print_output 첨부 파일을 참조하세요.</p>	

관련 리소스

- [LRS](#)
- [Advanced Function Presentation 데이터 스트림](#)(IBM 설명서)
- [Line Conditioned Data Stream\(LCDS\)](#)(설명서 첨부)
- [AWS 기반 Micro Focus Enterprise Server](#)(AWS Quick Starts)
- [Micro Focus를 통한 AWS 기반 엔터프라이즈 메인프레임 워크로드 강화](#)(블로그 게시물)
- [AWS에서 메인프레임 온라인 인쇄 워크로드 현대화](#)(AWS 권장 가이드)
- [AWS에서 메인프레임 배치 인쇄 워크로드 현대화](#)(AWS 권장 가이드)

추가 정보

고려 사항

현대화 과정에서 메인프레임 배치 및 온라인 프로세스에 대한 다양한 구성과 이러한 구성에서 생성되는 출력을 고려할 수 있습니다. 메인프레임 플랫폼은 인쇄에 직접적인 영향을 미치는 특정 요구 사항

에 따라 이를 사용하는 모든 고객 및 공급업체에 의해 맞춤화되었습니다. 예를 들어, 사용자의 현재 플랫폼은 IBM AFP 데이터 스트림 또는 Xerox LCDS를 현재 워크플로에 통합할 수 있습니다. 또한 [메인 프레임 캐리지 제어 문자](#)와 [채널 명령어](#)는 인쇄된 페이지의 모양에 영향을 줄 수 있으며 특별한 처리가 필요할 수 있습니다. 현대화 계획 프로세스의 일환으로 사용자의 인쇄 환경의 구성을 평가하고 이해하는 것이 좋습니다.

인쇄 데이터 캡처

OpenText 마이크로 포커스 프린트 엑시트는 LRS VPSX/MFI가 스폴 파일을 효과적으로 처리하는 데 필요한 정보를 전달합니다. 이 정보는 다음과 같이 관련 제어 블록에 전달된 필드로 구성됩니다.

- JOBNAME
- OWNER (USERID)
- DESTINATION
- FORM
- FILENAME
- WRITER

LRS VPSX/MFI는 마이크로 포커스 엔터프라이즈 서버에서 데이터를 캡처하기 위한 다음과 같은 메인 프레임 배치 메커니즘을 지원합니다. OpenText

- 표준 z/OS JCL SYSOUT DD/OUTPUT 문을 사용한 BATCH COBOL 프린트/스폴 프로세싱.
- 표준 z/OS JCL CA-SPOOL SUBSYS DD 문을 사용한 BATCH COBOL 인쇄/스폴 프로세싱.
- CBLTDLI 인터페이스를 사용한 IMS/COBOL 프린트/스폴 프로세싱. 지원되는 방법 및 프로그래밍 예제의 전체 목록은 제품 라이선스에 포함된 LRS 설명서를 참조하세요.

프린터 플릿 상태 확인

LRS VPSX/MFI(LRS LoadX)는 장치 관리 및 운영 최적화를 포함한 심층 상태 확인을 수행할 수 있습니다. 장치 관리는 프린터 장치의 오류를 감지하고 인쇄 요청을 정상 프린터로 라우팅할 수 있습니다. 프린터 플릿의 심층 상태 확인에 대한 자세한 내용은 제품 라이선스에 포함된 LRS 설명서를 참조하세요.

인증 및 권한 부여 인쇄

LRS/DIS를 사용하면 LRS 애플리케이션이 Microsoft Active Directory 또는 Lightweight Directory Access Protocol(LDAP) 서버를 사용하여 사용자 ID와 암호를 검증할 수 있습니다. 기본 인쇄 승인 외에도 LRS/DIS는 다음과 같은 사용 사례에서 세분화된 수준의 인쇄 보안 제어를 적용할 수 있습니다.

- 프린터 작업을 탐색하는지 관리할 수 있습니다.
- 다른 사용자 작업의 탐색 수준을 관리할 수 있습니다.
- 운영 작업 관리(예: 보류 또는 해제, 삭제, 수정, 복사, 재라우팅과 같은 명령 수준 보안). 보안은 Active Directory 보안 그룹 또는 LDAP 그룹과 마찬가지로 사용자 ID 또는 그룹으로 설정할 수 있습니다.

첨부

이 문서와 관련된 추가 콘텐츠에 액세스하려면 [attachment.zip](#) 파일의 압축을 풉니다.

Micro Focus Enterprise Server 및 LRS VPSX/MFI를 사용하여 AWS에서 메인프레임 배치 인쇄 워크로드를 현대화

작성자: Shubham Roy(AWS), Abraham Rondon(Micro Focus), Guy Tucker(Levi, Ray and Shoup Inc), Kevin Yung(AWS)

환경: PoC 또는 파일럿	소스: IBM 메인프레임	대상: AWS
R 유형: 리플랫폼	워크로드: IBM	기술: 메인프레임, 현대화

AWS 서비스: AWS Managed Microsoft AD, Amazon EC2, Amazon S3, Amazon EBS

요약

이 패턴은 Micro Focus Enterprise Server를 현대화된 메인프레임 애플리케이션의 런타임으로 사용하고 LRS VPSX/MFI(Micro Focus Interface)를 인쇄 서버로 사용하여 Amazon Web Services(AWS) 클라우드의 비즈니스에 중요한 메인프레임 배치 인쇄 워크로드를 현대화하는 방법을 보여줍니다. 이 패턴은 [리플랫폼](#) 메인프레임 현대화 접근 방식을 기반으로 합니다. 이 접근 방식에서는 메인프레임 배치 작업을 Amazon Elastic Compute Cloud(Amazon EC2)로 마이그레이션하고 IBM DB2 for z/OS와 같은 메인프레임 데이터베이스를 Amazon Relational Database Service(RDS)로 마이그레이션합니다. 현대화된 인쇄 워크플로에 대한 인증 및 권한 부여는 AWS Managed Microsoft AD라고도 알려진 Microsoft Active Directory용 AWS Directory Service에서 수행됩니다. LRS Directory Information Server(LRS/DIS)는 AWS Managed Microsoft AD와 통합됩니다. 배치 인쇄 워크로드를 현대화하면 IT 인프라 비용을 줄이고, 레거시 시스템 유지 관리에 따르는 기술적 부채를 완화하고, 데이터 사일로를 제거하고, DevOps 모델을 통해 민첩성과 효율성을 높이고, AWS 클라우드의 온디맨드 리소스 및 자동화를 활용할 수 있습니다.

사전 조건 및 제한 사항

사전 조건

- 활성 상태의 AWS 계정.
- 메인프레임 인쇄 또는 출력 관리 워크로드

- Micro Focus Enterprise Server에서 실행되는 메인프레임 애플리케이션을 재구축하고 전달하는 방법에 대한 기본 지식(자세한 내용은 Micro Focus 설명서의 [Enterprise Server](#) 데이터시트를 참조하세요.)
- LRS 클라우드 프린팅 솔루션 및 개념에 대한 기본 지식(자세한 내용은 LRS 설명서의 [출력 현대화](#) 참조)
- Micro Focus Enterprise Server 소프트웨어 및 라이선스(자세한 내용은 [Micro Focus 영업팀](#)에 문의하세요.)
- LRS VPSX/MFI, LRS/Queue, LRS/DIS 소프트웨어 및 라이선스(자세한 내용은 [LRS 영업팀](#)에 문의하세요.)

참고: 메인프레임 배치 인쇄 워크로드의 구성 고려 사항에 대한 자세한 내용은 이 패턴의 추가 정보 섹션에 있는 고려 사항을 참조하세요.

제품 버전

- [Micro Focus Enterprise Server](#) 6.0(제품 업데이트 7)
- [LRS VPSX/MFI](#) V1R3 이상

아키텍처

소스 기술 스택

- 운영 체제 – IBM z/OS
- 프로그래밍 언어 – 공통 비즈니스 지향 언어(COBOL), 직무 제어 언어(JCL) 및 고객 정보 제어 시스템(CICS)
- 데이터베이스 – z/OS용 IBM DB2 및 가상 스토리지 액세스 방법(VSAM)
- 보안 – Resource Access Control Facility (RACF), CA Top Secret for z/OS, and Access Control Facility 2 (ACF2)
- 인쇄 및 출력 관리 – IBM 메인프레임 z/OS 인쇄 제품(z/OS, LRS, CA View용 IBM Tivoli Output Manager)

대상 기술 스택

- 운영 체제 – Amazon EC2에서 실행되는 Microsoft Windows 서버
- 컴퓨팅 – Amazon EC2

- 프로그래밍 언어 – COBOL, JCL, CICS
- 데이터베이스 – Amazon RDS
- 보안 – AWS Managed Microsoft AD
- 인쇄 및 출력 관리 – AWS 기반 LRS 인쇄 솔루션
- 메인프레임 런타임 환경 – Micro Focus Enterprise Server

소스 아키텍처

다음 다이어그램은 메인프레임 일괄 인쇄 워크로드의 일반적인 현재 상태 아키텍처를 보여줍니다.

이 다이어그램은 다음 워크플로를 보여줍니다.

1. 사용자는 COBOL로 작성된 IBM CICS 애플리케이션을 기반으로 구축된 참여 시스템(SoE)에서 비즈니스 트랜잭션을 수행합니다.
2. SoE는 IBM DB2 for z/OS와 같은 system-of-records (oR) 데이터베이스에 비즈니스 트랜잭션 데이터를 기록하는 메인프레임 서비스를 호출합니다.
3. SoR은 SoE의 비즈니스 데이터를 유지합니다.
4. 배치 작업 스케줄러는 인쇄 출력을 생성하는 일괄 작업을 시작합니다.
5. 배치 작업은 데이터베이스에서 데이터를 추출하고 비즈니스 요구 사항에 따라 데이터 형식을 지정 한 다음 청구서, ID 카드 또는 대출 명세서와 같은 비즈니스 결과를 생성합니다. 마지막으로, 배치 작업은 비즈니스 요구 사항에 따라 출력을 인쇄 출력 관리로 라우팅하여 처리 및 출력 전달을 수행합니다.
6. 인쇄 출력 관리는 배치 작업에서 인쇄 출력을 받은 다음 해당 출력을 이메일, 보안 FTP를 사용하는 파일 공유, LRS 인쇄 솔루션을 사용하는 물리적 프린터(이 패턴에서 설명함) 또는 IBM Tivoli와 같은 지정된 대상으로 전달합니다.

대상 아키텍처

다음 다이어그램은 AWS 클라우드에 배포된 메인프레임 일괄 인쇄 워크로드의 아키텍처를 보여줍니다.

이 다이어그램은 다음 워크플로를 보여줍니다.

1. 배치 작업 스케줄러는 일괄 작업을 시작하여 청구서, ID 카드 또는 대출 명세서와 같은 인쇄 출력을 생성합니다.
2. 메인프레임 배치 작업([Amazon EC2로 리플랫폼](#))은 Micro Focus Enterprise Server 런타임을 사용하여 애플리케이션 데이터베이스에서 데이터를 추출하고, 데이터에 비즈니스 로직을 적용하고, 데이터 형식을 지정한 다음, [Micro Focus Print Exit](#)(Micro Focus 설명서)를 사용하여 인쇄 대상으로 데이터를 전송합니다.
3. 애플리케이션 데이터베이스(Amazon RDS에서 실행되는 SoR)는 인쇄 출력용 데이터를 보관합니다.
4. LRS VPSX/MFI 인쇄 솔루션은 Amazon EC2에 배포되며 운영 데이터는 Amazon Elastic Block Store(Amazon EBS)에 저장됩니다. LRS VPSX/MFI는 TCP/IP 기반 LRS/Queue 전송 에이전트를 사용하여 Micro Focus JES Print Exit API의 인쇄 데이터를 수집하고, 이 데이터를 지정된 프린터 목적지로 전송합니다.

참고: 대상 솔루션은 일반적으로 IBM Advanced Function Presentation(AFP) 또는 Xerox Line Condition Data Stream(LCDS)과 같은 메인프레임 형식 지정 언어를 수용하기 위해 애플리케이션을 변경할 필요가 없습니다. Micro Focus를 사용하여 AWS에서 메인프레임 애플리케이션을 마이그레이션 및 현대화하는 방법에 대한 자세한 내용은 AWS 설명서의 [Micro Focus를 통한 AWS 기반 엔터프라이즈 메인프레임 워크로드 강화](#)를 참조하세요.

AWS 인프라 아키텍처

다음 다이어그램은 메인프레임 일괄 인쇄 워크로드를 위한 가용성이 높고 안전한 AWS 인프라 아키텍처를 보여줍니다.

이 다이어그램은 다음 워크플로를 보여줍니다.

1. 배치 스케줄러는 배치 프로세스를 시작하고 고가용성(HA)을 위해 여러 [가용 영역](#)에 걸쳐 Amazon EC2에 배포됩니다. 참고: 이 패턴은 배치 스케줄러 구현을 다루지 않습니다. 구현에 대한 자세한 내용은 스케줄러의 소프트웨어 공급업체 설명서를 참조하세요.
2. JCL 또는 COBOL과 같은 프로그래밍 언어로 작성된 메인프레임 배치 작업은 핵심 비즈니스 로직을 사용하여 청구서, ID 카드, 대출 명세서와 같은 인쇄 결과를 처리하고 생성합니다. 작업은 HA를 위해 두 가용 영역에 걸쳐 Amazon EC2에 배포되며, 최종 사용자가 인쇄할 수 있도록 Micro Focus Print Exit을 사용하여 인쇄 출력을 LRS VPSX/MFI로 라우팅합니다.
3. LRS VPSX/MFI는 TCP/IP 기반 LRS/Queue 전송 에이전트를 사용하여 Micro Focus JES Print Exit 프로그래밍 인터페이스에서 인쇄 데이터를 수집하거나 캡처합니다. Print Exit는 LRS VPSX/MFI가

스플 파일을 효과적으로 처리하고 LRS/Queue 명령을 동적으로 구축할 수 있도록 필요한 정보를 전달합니다. 그런 다음 Micro Focus의 표준 내장 함수를 사용하여 명령을 실행합니다. 참고: Micro Focus Print Exit에서 LRS/Queue로 전달되는 인쇄 데이터 및 LRS VPSX/MFI 지원 메인프레임 배치 메커니즘에 대한 자세한 내용은 이 패턴의 추가 정보 섹션에서 인쇄 데이터 캡처를 참조하세요.

4. [Network Load Balancer](#)는 Micro Focus Enterprise Server를 LRS VPSX/MFI와 통합하기 위한 DNS 이름을 제공합니다. 참고: LRS VPSX/MFI는 Layer 4 로드 밸런서를 지원합니다. 또한 Network Load Balancer는 LRS VPSX/MFI에 대한 기본 상태 확인을 수행하고 정상으로 판명된 등록된 대상으로 트래픽을 라우팅합니다.
5. LRS VPSX/MFI 프린트 서버는 HA를 위해 두 가용 영역에 걸쳐 Amazon EC2에 배포되며 [Amazon EBS](#)를 운영 데이터 스토어로 사용합니다. LRS VPSX/MFI는 액티브-액티브 및 액티브-패시브 서비스 모드를 모두 지원합니다. 이 아키텍처는 액티브-패시브 페어의 여러 AZ를 액티브 및 상시 대기 방식으로 사용합니다. Network Load Balancer는 LRS VPSX/MFI EC2 인스턴스에서 상태 확인을 수행하고 활성 인스턴스가 비정상 상태인 경우 다른 AZ의 상시 대기 방식 인스턴스로 트래픽을 라우팅합니다. 인쇄 요청은 각 EC2 인스턴스의 LRS Job Queue에 로컬로 유지됩니다. 복구가 발생한 경우 LRS 서비스가 인쇄 요청 처리를 재개하려면 실패한 인스턴스를 다시 시작해야 합니다. 참고: LRS VPSX/MFI는 프린터 플릿 수준에서 상태 확인을 수행할 수도 있습니다. 자세한 내용은 이 패턴의 추가 정보 섹션에 있는 프린터 플릿 상태 확인을 참조하세요.
6. [AWS Managed Microsoft AD](#)는 LRS/DIS와 통합되어 인쇄 워크플로 인증 및 권한 부여를 수행합니다. 자세한 내용은 이 패턴의 추가 정보 섹션에 있는 인쇄 검증 및 권한 부여를 참조하세요.
7. LRS VPSX/MFI는 블록 스토리지에 Amazon EBS를 사용합니다. 활성 EC2 인스턴스의 Amazon EBS 데이터를 Amazon S3에 point-in-time 스냅샷으로 백업하고 핫 스탠바이 EBS 볼륨에 복원할 수 있습니다. Amazon EBS 볼륨 스냅샷의 생성, 보존 및 삭제를 자동화하려면 [Amazon Data Lifecycle Manager](#)를 사용하여 자동 스냅샷의 빈도를 설정하고 [RTO/RPO 요구 사항](#)에 따라 복원할 수 있습니다.

도구

서비스

- [Amazon EBS](#) – Amazon Elastic Block Store(Amazon EBS)는 Amazon EC2 인스턴스에 사용할 수 있는 블록 수준 스토리지 볼륨을 제공합니다. EBS 볼륨은 형식이 지정되지 않은 원시 블록 디바이스처럼 동작합니다. 이러한 볼륨을 인스턴스에 디바이스로 마운트할 수 있습니다.
- [Amazon EC2](#) – Amazon Elastic Compute Cloud(Amazon EC2)는 AWS 클라우드에서 확장 가능한 컴퓨팅 용량을 제공합니다. Amazon EC2를 사용하여 필요에 따라 많거나 적은 수의 가상 서버를 시작하고 스케일 아웃 또는 스케일 인할 수 있습니다.

- [Amazon RDS](#) - Amazon Relational Database Service(RDS)는 AWS 클라우드의 관계형 데이터베이스를 더 쉽게 설치, 운영 및 확장할 수 있게 하는 웹 서비스입니다. 이 서비스는 관계형 데이터베이스를 위한 경제적이고 크기 조절이 가능한 용량을 제공하고 공통 데이터베이스 관리 작업을 관리합니다.
- [AWS Managed Microsoft AD](#) – AWS Managed Microsoft Active Directory라고도 하는 Microsoft Active Directory용 AWS Directory Service를 사용하면 디렉터리 인식 워크로드와 AWS 리소스가 AWS의 관리형 Active Directory를 사용할 수 있습니다.

기타 도구

- [LRS VPSX/MFI\(Micro Focus Interface\)](#) – LRS와 Micro Focus가 공동 개발한 VPSX/MFI는 Micro Focus Enterprise Server JES 스펙에서 출력을 캡처하여 지정된 인쇄 대상으로 안정적으로 전달합니다.
- LRS Directory Information Server(LRS/DIS) – LRS/DIS는 인쇄 워크플로 중 검증 및 권한 부여에 사용됩니다.
- LRS/Queue – LRS VPSX/MFI는 TCP/IP 기반 LRS/Queue 전송 에이전트를 사용하여 Micro Focus JES Print Exit 프로그래밍 인터페이스에서 인쇄 데이터를 수집하거나 캡처합니다.
- [Micro Focus Enterprise Server](#) – Micro Focus Enterprise Server는 메인프레임 애플리케이션을 위한 애플리케이션 배포 환경입니다. Micro Focus Enterprise Developer의 모든 버전을 사용하여 마이그레이션하거나 생성한 메인프레임 애플리케이션의 실행 환경을 제공합니다.

에픽

Amazon EC2에 Micro Focus Enterprise Server를 설치하고 메인프레임 배치 애플리케이션을 배포

작업	설명	필요한 기술
Micro Focus Enterprise Server를 설치하고 데모 애플리케이션을 배포하세요.	Amazon EC2에 마이크로 포커스 엔터프라이즈 서버를 설치한 다음 AWS Quick Start의 마이크로 포커스 엔터프라이즈 서버 배포 가이드의 지침에 따라 Amazon EC2에 마이크로 포커스 BankDemo 데모 애플리케이션을 배포하십시오.	클라우드 아키텍트

작업	설명	필요한 기술
	<p>BankDemo 애플리케이션은 인쇄 출력을 생성하고 시작하는 메인프레임 배치 애플리케이션입니다.</p>	

Amazon EC2에 LRS 인쇄 서버 설치

작업	설명	필요한 기술
<p>인쇄용 LRS 제품 라이선스를 받습니다.</p>	<p>LRS VPSX/MFI, LRS/Queue 및 LRS/DIS에 대한 LRS 제품 라이선스를 받으려면 LRS 출력 관리 팀에 문의하세요. LRS 제품을 설치할 EC2 인스턴스의 호스트 이름을 제공해야 합니다.</p>	<p>리드 구축</p>
<p>LRS VPSX/MFI를 설치할 Amazon EC2 Windows 인스턴스를 생성합니다.</p>	<p>Amazon EC2 설명서의 1단계: 인스턴스 시작의 지침에 따라 Amazon EC2 Windows 인스턴스를 시작합니다. 인스턴스는 LRS VPSX/MFI에 대한 다음 하드웨어 및 소프트웨어 요구 사항을 충족해야 합니다.</p> <ul style="list-style-type: none"> • CPU – 듀얼 코어 • 램 – 16GB • 드라이브 – 500GB • 최소 EC2 인스턴스 – m5.xlarge • OS – Windows/Linux • 소프트웨어 – 인터넷 정보 서비스(IIS) 또는 Apache 	<p>클라우드 아키텍트</p>

작업	설명	필요한 기술
	<p>참고: 위의 하드웨어 및 소프트웨어 요구 사항은 소형 프린터 플릿(약 500~1000대)을 대상으로 합니다. 전체 요구 사항을 알아보려면 LRS 및 AWS 담당자에게 문의하세요.</p> <p>Windows 인스턴스를 생성할 때 다음을 수행하세요.</p> <ol style="list-style-type: none"> 1. EC2 호스트 이름이 LRS 제품 라이선스에 사용된 호스트 이름과 동일한지 확인하세요. 2. 다음을 완료하여 Amazon EC2에서 CGI를 활성화하세요. <ol style="list-style-type: none"> a. Amazon EC2 설명서의 2 단계: 인스턴스에 연결 지침에 따라 EC2 인스턴스에 연결합니다. b. Windows 시작 메뉴에서 서버 관리자를 찾아 엽니다. c. 서버 관리자에서 대시보드, 빠른 시작, 역할 및 기능 추가를 선택합니다. 그런 다음 서버 역할을 선택합니다. d. 서버 역할에서 WebServer (IIS) 를 선택한 다음 응용 프로그램 개발을 선택합니다. 	

작업	설명	필요한 기술
	<ul style="list-style-type: none">e. 애플리케이션 개발에서 CGI 확인란에 체크합니다.f. Windows 서버 관리자 역할 및 기능 추가 마법사의 지침에 따라 CGI를 설치합니다.g. LRS/Queue 통신을 위해 EC2 인스턴스의 Windows 방화벽에 포트 5500을 엽니다.	

작업	설명	필요한 기술
<p>EC2 인스턴스에 LRS VPSX/MFI를 설치합니다.</p>	<ol style="list-style-type: none"> 1. Amazon EC2 설명서의 2단계: 인스턴스에 연결 지침에 따라 EC2 인스턴스에 연결합니다. 2. 수신한 LRS 이메일에서 제품 다운로드 페이지 링크를 여세요. 참고: LRS 제품은 전자 파일 전송(EFT)을 통해 배포됩니다. 3. LRS VPSX/MFI를 다운로드 하고 파일의 압축을 풉니다 (기본 폴더: c:\LRS). 4. 압축을 푼 폴더에서 LRS 제품 설치 프로그램을 실행하여 LRS VPSX/MFI를 설치합니다. 5. 기능 선택 메뉴에서 VPSX® Server(V1R3.022)를 선택하고 다음을 선택하여 설치 프로세스를 시작합니다. 설치가 완료되면 성공 메시지가 나타납니다. 	<p>클라우드 아키텍트</p>

작업	설명	필요한 기술
LRS/Queue를 설치합니다.	<ol style="list-style-type: none"> 1. Amazon EC2 설명서의 2단계: 인스턴스에 연결 지침에 따라 Micro Focus Enterprise Server EC2 인스턴스에 연결합니다. 2. 수신한 LRS 이메일에서 LRS 제품 다운로드 페이지 링크를 열고 LRS/Queue를 다운로드한 다음 파일의 압축을 풉니다. 3. 파일을 다운로드한 위치로 이동한 다음 LRS 제품 설치 프로그램을 실행하여 LRS/Queue를 설치합니다. 	클라우드 아키텍트

작업	설명	필요한 기술
LRS/DIS를 설치합니다.	<ol style="list-style-type: none"> 1. Amazon EC2 설명서의 2단계: 인스턴스에 연결 지침에 따라 LRS VPSX/MFI EC2 인스턴스에 연결합니다. 2. 수신한 LRS 이메일에서 LRS 제품 다운로드 페이지 링크를 열고 LRS/DIS를 다운로드한 다음 파일의 압축을 풉니다. 3. 파일을 다운로드한 위치로 이동한 다음 LRS 제품 설치 프로그램을 시작합니다. 4. LRS 제품 설치 프로그램에서 LRS 기타 도구를 확장하고 LRS DIS를 선택한 후 다음을 선택합니다. 5. LRS 제품 설치 프로그램의 나머지 지침을 따라 설치 프로세스를 완료하세요. 	클라우드 아키텍트

작업	설명	필요한 기술
<p>대상 그룹을 생성하고 LRS VPSX/MFI EC2를 대상으로 등록합니다.</p>	<p>Elastic Load Balancing 설명서의 Network Load Balancer에 대한 대상 그룹 생성 지침에 따라 대상 그룹을 생성합니다.</p> <p>대상 그룹을 생성할 때 다음 작업을 수행합니다.</p> <ol style="list-style-type: none"> 1. 그룹 세부 정보 지정 페이지의 대상 유형 선택에서 인스턴스를 선택합니다. 2. 프로토콜에서 TCP를 선택합니다. 3. 포트는 5500을 선택합니다. 4. 대상 등록 페이지의 사용 가능한 인스턴스 섹션에서 LRS VPSX/MFI EC2 인스턴스를 선택합니다. 	<p>클라우드 아키텍트</p>

작업	설명	필요한 기술
<p>Network Load Balancer를 생성합니다.</p>	<p>Elastic Load Balancing 설명서의 Network Load Balancer 생성 지침을 따르세요. Network Load Balancer는 Micro Focus Enterprise Server에서 LRS VPSX/MFI EC2로 트래픽을 라우팅합니다.</p> <p>Network Load Balancer를 생성할 때 리스너 및 라우팅 페이지에서 다음 작업을 수행하세요.</p> <ol style="list-style-type: none"> 1. 프로토콜에서 TCP를 선택합니다. 2. 포트는 5500을 선택합니다. 3. 기본 작업에서 앞서 생성한 대상 그룹에 대해 전달을 선택합니다. 	<p>클라우드 아키텍트</p>

Micro Focus Enterprise Server를 LRS VPSX/MFI 및 LRS/Queue와 통합

작업	설명	필요한 기술
<p>LRS/Queue 통합을 위해 Micro Focus Enterprise Server를 구성하세요.</p>	<ol style="list-style-type: none"> 1. Amazon EC2 설명서의 2단계: 인스턴스에 연결 지침에 따라 Micro Focus Enterprise Server EC2 인스턴스에 연결합니다. 2. Windows 시작 메뉴에서 Micro Focus Enterprise Server 관리 UI를 엽니다. 3. 메뉴 모음에서 네이티브를 선택합니다. 	<p>클라우드 아키텍트</p>

작업	설명	필요한 기술
	<p>4. 탐색 창에서 디렉토리 서버를 선택한 다음 BANKDEMO를 선택합니다.</p> <p>5. 왼쪽 탐색 창의 일반에서 추가 섹션으로 스크롤하여 LRSQ를 가리키도록 환경 변수(LRSQ_ADDRESS, LRSQ_PORT, LRSQ_COMMAND)를 구성합니다.</p> <p>6. LRSQ_ADDRESS의 경우 이전에 생성한 Network Load Balancer의 IP 주소 또는 DNS 이름을 입력합니다.</p> <p>7. LRSQ_PORT의 경우 VPSX LRSQ 리스너 포트(5500)를 입력합니다.</p> <p>8. LRSQ_COMMAND의 경우 LRSQ 실행 파일의 경로 위치를 입력합니다.</p> <p>참고: LRS는 현재 DNS 이름에 최대 50자 제한을 지원하지만 향후 변경될 수 있습니다. DNS 이름이 50자보다 길면 Network Load Balancer의 IP 주소를 대안으로 사용할 수 있습니다.</p>	

작업	설명	필요한 기술
<p>LRS VPSX/MFI 통합을 위해 Micro Focus Enterprise Server 를 구성하세요.</p>	<ol style="list-style-type: none"> 1. LRS VPSX/MFI 설치 프로그램의 VPSX_MFI_R2 폴더를 C\BANKDEMO\print 의 Micro Focus Enterprise Server 위치로 복사하세요. 2. Amazon EC2 설명서의 2단계: 인스턴스에 연결 지침에 따라 Micro Focus Enterprise Server EC2 인스턴스에 연결합니다. 3. Windows 시작 메뉴에서 Micro Focus Enterprise Server 관리 UI를 엽니다. 4. 메뉴 모음에서 네이티브를 선택합니다. 5. 탐색 창에서 디렉토리 서버를 선택한 다음 BANKDEMO를 선택합니다. 6. BANKDEMO에서 JES를 선택합니다. 7. JES 프로그램 경로에 C\BANKDEMO\print 위치의 DLL(VPSX_MFI_R2) 경로를 추가합니다. 	<p>클라우드 아키텍트</p>

Micro Focus Enterprise Server 및 LRS VPSX/MFI에서 프린터와 프린트 사용자를 설정

작업	설명	필요한 기술
<p>Micro Focus Print Exit 모듈을 Micro Focus Enterprise Server 배치 프린터 서버 실행 프로세스에 연결합니다.</p>	<ol style="list-style-type: none"> 1. Amazon EC2 설명서의 2단계: 인스턴스에 연결 지침에 따라 Micro Focus Enterprise Server EC2 인스턴스에 연결합니다. 	<p>클라우드 아키텍트</p>

작업	설명	필요한 기술
	<p>e Server EC2 인스턴스에 연결합니다.</p> <ol style="list-style-type: none"> 2. Windows 시작 메뉴에서 Micro Focus Enterprise Server 관리 UI를 엽니다. 3. 메뉴 모음에서 네이티브를 선택합니다. 4. 탐색 창에서 디렉토리 서버를 선택한 다음 BANKDEMO를 선택합니다. 5. BANKDEMO에서 JES를 선택하고 아래로 스크롤하여 프린터를 찾습니다. 6. 프린터에서 Micro Focus Print Exit 모듈(Batch용 LRSPRTE6)을 Micro Focus Enterprise Server 배치 프린터 Server Execution Process(SEP)에 연결합니다. 이렇게 하면 인쇄 출력을 LRS VPSX/MFI로 라우팅할 수 있습니다. 7. Enterprise Server Administration UI에 로그인합니다. <p>구성에 대한 자세한 내용은 Micro Focus 설명서의 Exit 사용을 참조하세요.</p>	

작업	설명	필요한 기술
<p>LRS VPSX/MFI에 프린터를 추가하세요.</p>	<ol style="list-style-type: none"> 1. Amazon EC2 설명서의 2단계: 인스턴스에 연결 지침에 따라 LRS VPSX/MFI EC2 인스턴스에 연결합니다. 2. Windows 시작 메뉴에서 VPSX 웹 인터페이스를 엽니다. 3. 탐색 창에서 프린터를 선택합니다. 4. 추가를 선택한 다음 프린터 추가를 선택합니다. 5. 프린터 구성 페이지에서 프린터 이름에 Local을 입력합니다. 6. VPSX ID의 경우 VPS1을 입력합니다. 7. 의 CommType 경우 TCP/PIP/LRSQ를 선택합니다. 8. 호스트/IP 주소에는 추가하려는 물리적 프린터의 IP 주소를 입력합니다. 9. 장치에 장치 이름을 입력합니다. 10.Windows 드라이버 또는 Linux/Mac 드라이버를 선택합니다. 11. 추가를 선택합니다. 	<p>클라우드 아키텍트</p>

작업	설명	필요한 기술
LRS VPSX/MFI에서 프린트 사용자를 생성하세요.	<ol style="list-style-type: none"> 1. Amazon EC2 설명서의 2단계: 인스턴스에 연결 지침에 따라 LRS VPSX/MFI EC2 인스턴스에 연결합니다. 2. Windows 시작 메뉴에서 VPSX 웹 인터페이스를 엽니다. 3. 탐색 창에서 보안을 선택한 다음 사용자를 선택합니다. 4. 사용자 이름 열에서 관리자를 선택한 다음 복사를 선택합니다. 5. 사용자 프로필 유지 관리 창에서 사용자 이름에 사용자 이름 (예:) 을 입력합니다. PrintUser 6. 설명에 간단한 설명(예: 테스트 인쇄용 사용자)을 입력합니다. 7. 업데이트를 선택합니다. 그러면 인쇄 사용자 (예: PrintUser) 가 생성됩니다. 8. 탐색 창의 사용자에서 생성한 새 사용자를 선택합니다. 9. 명령 메뉴에서 보안을 선택합니다. 10. 보안 규칙 페이지에서 해당하는 프린터 보안 및 작업 보안 옵션을 모두 선택한 다음 저장을 선택합니다. 11. 새 인쇄 사용자를 관리자 그룹에 추가하려면 탐색 창으 	클라우드 아키텍트

작업	설명	필요한 기술
	<p>로 이동하여 보안을 선택한 다음 구성을 선택합니다.</p> <p>12보안 구성 창에서 관리자 열에 새 인쇄 사용자를 추가합니다.</p>	

인쇄 인증 및 권한 부여 설정

작업	설명	필요한 기술
<p>사용자 및 그룹을 사용하여 AWS Managed Microsoft AD 도메인을 생성합니다.</p>	<ol style="list-style-type: none"> 1. AWS Directory Service 설명서의 AWS Managed Microsoft AD 디렉터리 생성 지침에 따라 AWS Managed Microsoft AD에 Active Directory를 생성합니다. 2. AWS Directory Service 설명서의 3단계: AWS Managed Microsoft AD를 관리하기 위한 EC2 인스턴스 배포 지침에 따라 EC2 인스턴스(Active Directory 관리자)를 배포하고 Active Directory 도구를 설치하여 AWS Managed Microsoft AD를 관리합니다. 3. Amazon EC2 설명서의 2단계: 인스턴스에 연결 지침에 따라 EC2 인스턴스에 연결합니다. 참고: EC2 인스턴스에 연결할 때 Windows 보안 창에 관리자 보안 인증 정보(1단계에서 생성한 디렉터리의)를 입력합니다. 	클라우드 아키텍트

작업	설명	필요한 기술
	<p>4. Windows 시작 메뉴에서 Windows 관리 도구 아래에서 Active Directory 사용자 및 컴퓨터를 선택합니다.</p> <p>5. AWS Directory Service 설명서의 사용자 생성 단계에 따라 Active Directory 도메인에서 인쇄 사용자를 생성합니다.</p>	
<p>AWS Managed Microsoft AD 도메인에 LRS VPSX/MFI EC2를 조인합니다.</p>	<p>LRS VPSX/MFI EC2를 AWS Managed Microsoft AD 도메인에 자동으로(AWS 지식 센터 설명서) 또는 수동으로(AWS Directory Service 설명서) 조인하세요.</p>	<p>클라우드 아키텍트</p>

작업	설명	필요한 기술
<p>AWS Managed Microsoft AD 와 LRS/DIS를 구성 및 통합합니다.</p>	<ol style="list-style-type: none"> 1. Amazon EC2 설명서의 2단계: 인스턴스에 연결 지침에 따라 LRS VPSX/MFI EC2 인스턴스에 연결합니다. 2. Windows 시작 메뉴에서 VPSX Web Interface를 엽니다. 3. 탐색 창에서 보안을 선택한 다음 구성을 선택합니다. 4. 보안 구성 페이지의 보안 파라미터 섹션에서 보안 유형에 대해 내부를 선택합니다. 5. 보안 파라미터 섹션의 나머지 옵션에 원하는 설정을 입력합니다. 6. Microsoft Windows 시작 메뉴에서 LRS 출력 관리 폴더를 열고 서버 시작을 선택한 다음 서버 중지를 선택합니다. 7. Active Directory 사용자 이름 및 암호를 사용하여 LRS VPSX/MFI에 로그인합니다. 	<p>클라우드 아키텍트</p>

인쇄 워크플로 테스트

작업	설명	필요한 기술
<p>Micro Focus BankDemo 앱에서 일괄 인쇄 요청을 시작하십시오.</p>	<ol style="list-style-type: none"> 1. Micro Focus Enterprise Server EC2 인스턴스에서 3270 터미널 에뮬레이터를 엽니다. 	<p>테스트 엔지니어</p>

작업	설명	필요한 기술
	<ol style="list-style-type: none">2. 다음 명령어를 실행하여 BankDemo 앱에 연결합니다. <code>connect 127.0.0.1:9278</code>3. BankDemo 명령줄 인터페이스의 사용자 ID에 B0001 를 입력합니다. 비밀번호에는 비어 있지 않은 키를 입력합니다.4. 인쇄된 명세서 요청 옵션의 경우 빈 줄에 X를 입력합니다.5. 명세서 전송 섹션의 메일에 Y를 입력한 다음 F10을 누릅니다.	

작업	설명	필요한 기술
<p>LRS VPSX/MFI에서 인쇄 출력을 확인합니다.</p>	<ol style="list-style-type: none"> 1. Amazon EC2 설명서의 2단계: 인스턴스에 연결 지침에 따라 LRS VPSX/MFI EC2 인스턴스에 연결합니다. 2. Windows 시작 메뉴에서 VPSX Web Interface를 엽니다. 3. 탐색 창에서 프린터를 선택한 다음 출력 대기열을 선택합니다. 4. 스펴 ID 열에서 프린터 대기열에 있는 요청의 스펴 ID를 선택합니다. 5. 작업 탭의 명령 열에서 찾아보기를 선택합니다. <p>이제 계좌 번호, 설명, 날짜, 금액 및 잔액 열이 포함된 계좌 명세서의 인쇄 출력을 볼 수 있습니다. 예제를 확인하려면 이 패턴에 대한 <code>batch_print_output</code> 첨부 파일을 참조하세요.</p>	<p>테스트 엔지니어</p>

관련 리소스

- [LRS 출력 현대화\(LRS 문서\)](#)
- [ANSI 및 머신 캐리지 제어\(IBM 설명서\)](#)
- [채널 명령어\(IBM 설명서\)](#)
- [Micro Focus를 통한 AWS 기반 엔터프라이즈 메인프레임 워크로드 강화\(AWS 파트너 네트워크 블로그\)](#)
- [Amazon EC2 Auto Scaling 및 Systems Manager를 사용하여 Micro Focus Enterprise Server PAC 구축\(AWS 권장 가이드 설명서\)](#)

- [Advanced Function Presentation\(AFP\) 데이터 스트림](#)(IBM 설명서)
- [Line Conditioned Data Stream\(LCDS\)](#)(설명서 첨부)
- [AWS 기반 Micro Focus Enterprise Server](#)(AWS Quick Starts)

추가 정보

고려 사항

현대화 과정에서 메인프레임 배치 프로세스와 해당 프로세스에서 생성되는 출력 모두에 대한 다양한 구성을 고려할 수 있습니다. 메인프레임 플랫폼은 인쇄에 직접적인 영향을 미치는 특정 요구 사항에 따라 이를 사용하는 모든 고객 및 공급업체에 의해 맞춤화되었습니다. 예를 들어, 현재 플랫폼은 IBM Advanced Function Presentation(AFP) 또는 Xerox Line Condition Data Stream(LCDS)을 현재 워크플로에 통합할 수 있습니다. 또한 [메인프레임 캐리지 제어 문자](#)와 [채널 명령어](#)는 인쇄된 페이지의 모양에 영향을 줄 수 있으며 특별한 처리가 필요할 수 있습니다. 현대화 계획 프로세스의 일환으로 사용자의 인쇄 환경의 구성을 평가하고 이해하는 것이 좋습니다.

인쇄 데이터 캡처

Micro Focus Print Exit는 LRS VPSX/MFI가 스폴 파일을 효과적으로 처리할 수 있도록 필요한 정보를 전달합니다. 이 정보는 다음과 같이 관련 제어 블록에 전달된 필드로 구성됩니다.

- JOBNAME
- OWNER (USERID)
- DESTINATION
- FORM
- FILENAME
- WRITER

LRS VPSX/MFI는 Micro Focus Enterprise Server에서 데이터를 캡처하기 위한 다음과 같은 메인프레임 배치 메커니즘을 지원합니다.

- 표준 z/OS JCL SYSOUT DD/OUTPUT 문을 사용한 BATCH COBOL 프린트/스폴 프로세싱
- 표준 z/OS JCL CA-SPOOL SUBSYS DD 문을 사용한 BATCH COBOL 인쇄/스폴 프로세싱
- CBLTDLI 인터페이스를 사용한 IMS/COBOL 인쇄/스폴 처리(지원되는 방법 및 프로그래밍 예제의 전체 목록은 제품 라이선스에 포함된 LRS 설명서를 참조하세요.)

프린터 플릿 상태 확인

LRS VPSX/MFI(LRS LoadX)는 장치 관리 및 운영 최적화를 포함한 심층 상태 확인을 수행할 수 있습니다. 장치 관리는 프린터 장치의 오류를 감지하고 인쇄 요청을 정상 프린터로 라우팅할 수 있습니다. 프린터 플릿의 심층 상태 확인에 대한 자세한 내용은 제품 라이선스에 포함된 LRS 설명서를 참조하세요.

인쇄 인증 및 권한 부여

LRS/DIS를 사용하면 LRS 애플리케이션이 Microsoft Active Directory 또는 LDAP 서버를 사용하여 사용자 ID와 암호를 검증할 수 있습니다. 기본 인쇄 승인 외에도 LRS/DIS는 다음과 같은 사용 사례에서 세분화된 수준의 인쇄 보안 제어를 적용할 수 있습니다.

- 프린터 작업을 탐색하는지 관리할 수 있습니다.
- 다른 사용자 작업의 탐색 수준을 관리할 수 있습니다.
- 운영 작업을 관리할 수 있습니다. 보류/해제, 삭제, 수정, 복사, 재라우팅과 같은 명령 수준 보안을 예로 들 수 있습니다. 보안은 User-ID 또는 그룹(AD 그룹 또는 LDAP 그룹과 유사)으로 설정할 수 있습니다.

첨부

이 문서와 관련된 추가 콘텐츠에 액세스하려면 [attachment.zip](#) 파일의 압축을 풉니다.

Micro Focus Enterprise Server 및 LRS VPSX/MFI를 사용하여 AWS에서 메인프레임 온라인 인쇄 워크로드를 현대화

작성자: Shubham Roy(AWS), Abraham Rondon(Micro Focus), Guy Tucker(Levi, Ray and Shoup Inc), Kevin Yung(AWS)

환경: PoC 또는 파일럿	소스: 메인프레임	대상: AWS
R 유형: 리플랫폼	워크로드: IBM	기술: 메인프레임, 마이그레이션, 현대화

AWS 서비스: AWS Managed Microsoft AD, Amazon EC2, Amazon RDS, Amazon EBS

요약

이 패턴은 Micro Focus Enterprise Server를 현대화된 메인프레임 애플리케이션의 런타임으로 사용하고 LRS VPSX/MFI(Micro Focus Interface)를 인쇄 서버로 사용하여 Amazon Web Services(AWS) 클라우드의 비즈니스에 중요한 메인프레임 온라인 인쇄 워크로드를 현대화하는 방법을 보여줍니다. 이 패턴은 [리플랫폼](#) 메인프레임 현대화 접근 방식을 기반으로 합니다. 이 접근 방식에서는 메인프레임 온라인 애플리케이션을 Amazon Elastic Compute Cloud(Amazon EC2)로 마이그레이션하고 IBM DB2 for z/OS와 같은 메인프레임 데이터베이스를 Amazon Relational Database Service(RDS)로 마이그레이션합니다. 현대화된 인쇄 워크플로에 대한 인증 및 권한 부여는 AWS Managed Microsoft AD라고도 알려진 Microsoft Active Directory용 AWS Directory Service에서 수행됩니다. LRS Directory Information Server(LRS/DIS)는 인쇄 워크플로 인증 및 권한 부여를 위해 AWS Managed Microsoft AD와 통합됩니다. 온라인 인쇄 워크로드를 현대화하면 IT 인프라 비용을 줄이고, 레거시 시스템 유지 관리에 따르는 기술적 부채를 줄이고, 데이터 사일로 제거하고, DevOps 모델을 통해 민첩성과 효율성을 높이고, AWS 클라우드의 온디맨드 리소스 및 자동화를 활용할 수 있습니다.

사전 조건 및 제한 사항

사전 조건

- 활성 상태의 AWS 계정.
- 메인프레임 온라인 인쇄 또는 출력 관리 워크로드

- Micro Focus Enterprise Server에서 실행되는 메인프레임 애플리케이션을 재구축하고 전달하는 방법에 대한 기본 지식(자세한 내용은 Micro Focus 설명서의 [Enterprise Server](#) 데이터시트를 참조하세요.)
- LRS 클라우드 프린팅 솔루션 및 개념에 대한 기본 지식(자세한 내용은 LRS 설명서의 [출력 현대화](#) 참조)
- Micro Focus Enterprise Server 소프트웨어 및 라이선스(자세한 내용은 [Micro Focus 영업팀](#)에 문의하세요.)
- LRS VPSX/MFI, LRS/Queue, LRS/DIS 소프트웨어 및 라이선스(자세한 내용은 [LRS 영업팀](#)에 문의하세요.)

참고: 메인프레임 온라인 인쇄 워크로드의 구성 고려 사항에 대한 자세한 내용은 이 패턴의 추가 정보 섹션에 있는 고려 사항을 참조하세요.

제품 버전

- [마이크로 포커스 엔터프라이즈 서버 8.0 이상](#)
- [LRS VPSX/MFI V1R3 이상](#)

아키텍처

소스 기술 스택

- 운영 체제 – IBM z/OS
- 프로그래밍 언어 – 공통 비즈니스 지향 언어(COBOL), 및 고객 정보 제어 시스템(CICS)
- 데이터베이스 – IBM DB2 for z/OS IBM Information Management System(IMS) 및 Virtual Storage Access Method(VSAM)
- 보안 – Resource Access Control Facility(RACF), zCA Top Secret for z/OS, Access Control Facility 2(ACF2)
- 인쇄 및 출력 관리 – IBM 메인프레임 z/OS 인쇄 제품(z/OS, LRS, CA View용 IBM Infoprint Server)

대상 기술 스택

- 운영 체제 – Amazon EC2에서 실행되는 Microsoft Windows 서버
- 컴퓨팅 – Amazon EC2
- 프로그래밍 언어 – COBOL 및 CICS

- 데이터베이스 – Amazon RDS
- 보안 – AWS Managed Microsoft AD
- 인쇄 및 출력 관리 – AWS 기반 LRS 인쇄 솔루션
- 메인프레임 런타임 환경 – Micro Focus Enterprise Server

소스 아키텍처

다음 다이어그램은 메인프레임 온라인 인쇄 워크로드의 일반적인 현재 상태 아키텍처를 보여줍니다.

이 다이어그램은 다음 워크플로를 보여줍니다.

1. 사용자는 COBOL로 작성된 IBM CICS 애플리케이션을 기반으로 구축된 참여 시스템(SoE)에서 비즈니스 트랜잭션을 수행합니다.
2. SoE는 IBM DB2 for z/OS와 같은 system-of-records (SoR) 데이터베이스에 비즈니스 트랜잭션 데이터를 기록하는 메인프레임 서비스를 호출합니다.
3. SoR은 SoE의 비즈니스 데이터를 유지합니다.
4. 사용자가 CICS SoE에서 인쇄 출력을 생성하라는 요청을 시작하면 인쇄 요청을 처리하기 위한 인쇄 트랜잭션 애플리케이션이 시작됩니다.
5. 인쇄 트랜잭션 애플리케이션(예: CICS 및 COBOL 프로그램)은 데이터베이스에서 데이터를 추출하고 비즈니스 요구 사항에 따라 데이터 형식을 지정한 다음 청구서, ID 카드 또는 대출 명세서와 같은 비즈니스 출력(인쇄 데이터)을 생성합니다. 그런 다음 애플리케이션은 Virtual Telecommunications Access Method(VTAM)를 사용하여 인쇄 요청을 보냅니다. z/OS 인쇄 서버 (예: IBM Inforprint 서버)는 NetSpool 또는 유사한 VTAM 구성 요소를 사용하여 인쇄 요청을 가로채고 JES 출력 매개변수를 사용하여 JES 스펴에 인쇄 출력 데이터 세트를 생성합니다. JES 출력 파라미터는 인쇄 서버가 출력을 특정 네트워크 프린터로 전송하는 데 사용하는 라우팅 정보를 지정합니다. VTAM이라는 용어는 z/OS Communications Server 및 z/OS의 System Network Architecture(SNA) 서비스 요소를 나타냅니다.
6. 인쇄 출력 전송 컴포넌트는 JES 스펴의 출력 인쇄 데이터 세트를 LRS(이 패턴에서 설명함), IBM InfoPrint Server 또는 이메일 목적지와 같은 원격 프린터 또는 인쇄 서버로 전송합니다.

대상 아키텍처

다음 다이어그램은 AWS 클라우드에 배포된 메인프레임 온라인 인쇄 워크로드의 아키텍처를 보여줍니다.

이 다이어그램은 다음 워크플로를 보여줍니다.

1. 사용자가 온라인(CICS) 사용자 인터페이스에서 인쇄 요청을 시작하여 청구서, ID 카드 또는 대출 명세서와 같은 인쇄 출력을 생성합니다.
2. 메인프레임 온라인 애플리케이션([Amazon EC2로 리플랫폼됨](#))은 Micro Focus Enterprise Server 런타임을 사용하여 애플리케이션 데이터베이스에서 데이터를 추출하고, 데이터에 비즈니스 로직을 적용하고, 데이터 형식을 지정한 다음, [Micro Focus CICS Print Exit](#)(DFHUPRNT)를 사용하여 인쇄 대상으로 데이터를 전송합니다.
3. 애플리케이션 데이터베이스(Amazon RDS에서 실행되는 SoR)는 인쇄 출력용 데이터를 보관합니다.
4. LRS VPSX/MFI 인쇄 솔루션은 Amazon EC2에 배포되며 운영 데이터는 Amazon Elastic Block Store(Amazon EBS)에 저장됩니다. LRS VPSX/MFI는 TCP/IP 기반 LRS/Queue 전송 에이전트를 사용하여 Micro Focus CICS Print Exit API(DFHUPRNT)의 인쇄 데이터를 수집하고, 이 데이터를 지정된 프린터 목적지로 전송합니다. 현대화된 CICS 애플리케이션에서 사용되는 기존 TERMIID(TERM)는 VPSX/MFI 대기열 이름으로 사용됩니다.

참고: 대상 솔루션은 일반적으로 IBM Advanced Function Presentation(AFP) 또는 Xerox Line Condition Data Stream(LCDS)과 같은 메인프레임 형식 지정 언어를 수용하기 위해 애플리케이션을 변경할 필요가 없습니다. Micro Focus를 사용하여 AWS에서 메인프레임 애플리케이션을 마이그레이션 및 현대화하는 방법에 대한 자세한 내용은 AWS 설명서의 [Micro Focus를 통한 AWS 기반 엔터프라이즈 메인프레임 워크로드 강화](#)를 참조하세요.

AWS 인프라 아키텍처

다음 다이어그램은 메인프레임 온라인 인쇄 워크로드를 위한 가용성이 높고 안전한 AWS 인프라 아키텍처를 보여줍니다.

이 다이어그램은 다음 워크플로를 보여줍니다.

1. CICS 또는 COBOL과 같은 프로그래밍 언어로 작성된 메인프레임 온라인 애플리케이션은 핵심 비즈니스 로직을 사용하여 청구서, ID 카드, 대출 명세서와 같은 인쇄 결과를 처리하고 생성합니다. 온라인 애플리케이션은 고가용성(HA)을 위해 두 개의 [가용 영역](#)(AZ)에 걸쳐 Amazon EC2에 배포되며, 최종 사용자가 인쇄할 수 있도록 Micro Focus CICS Print Exit를 사용하여 인쇄 출력을 LRS VPSX/MFI로 라우팅합니다.

2. LRS VPSX/MFI는 TCP/IP 기반 LRS/Queue 전송 에이전트를 사용하여 Micro Focus Online Print Exit 프로그래밍 인터페이스에서 인쇄 데이터를 수집하거나 캡처합니다. Online Print Exit는 LRS VPSX/MFI가 프린트 파일을 효과적으로 처리하고 LRS/Queue 명령을 동적으로 구축할 수 있도록 필요한 정보를 전달합니다. 참고: 인쇄를 위한 다양한 CICS 애플리케이션 프로그래밍 방법 및 Micro Focus Enterprise 서버 및 LRS VPSX/MFI가 이를 지원하는 방법에 대한 자세한 내용은 이 패턴의 추가 정보 섹션에 있는 인쇄 데이터 캡처를 참조하세요.
3. [Network Load Balancer](#)는 Micro Focus Enterprise Server를 LRS VPSX/MFI와 통합하기 위한 DNS 이름을 제공합니다. 참고: LRS VPSX/MFI는 Layer 4 로드 밸런서를 지원합니다. 또한 Network Load Balancer는 LRS VPSX/MFI에 대한 기본 상태 확인을 수행하고 정상으로 판명된 등록된 대상으로 트래픽을 라우팅합니다.
4. LRS VPSX/MFI 프린트 서버는 HA를 위해 두 가용 영역에 걸쳐 Amazon EC2에 배포되며 [Amazon EBS](#)를 운영 데이터 스토어로 사용합니다. LRS VPSX/MFI는 액티브-액티브 및 액티브-패시브 서비스 모드를 모두 지원합니다. 이 아키텍처는 액티브-패시브 페어의 여러 가용 영역을 액티브 및 상시 대기 방식으로 사용합니다. Network Load Balancer는 LRS VPSX/MFI EC2 인스턴스에서 상태 확인을 수행하고 활성 인스턴스가 비정상 상태인 경우 다른 가용 영역의 상시 대기 방식 인스턴스로 트래픽을 라우팅합니다. 인쇄 요청은 각 EC2 인스턴스의 LRS Job Queue에 로컬로 유지됩니다. 복구가 발생한 경우 LRS 서비스가 인쇄 요청 처리를 재개하려면 실패한 인스턴스를 다시 시작해야 합니다. 참고: LRS VPSX/MFI는 프린터 플릿 수준에서 상태 확인을 수행할 수도 있습니다. 자세한 내용은 이 패턴의 추가 정보 섹션에 있는 프린터 플릿 상태 확인을 참조하세요.
5. [AWS Managed Microsoft AD](#)는 LRS/DIS와 통합되어 인쇄 워크플로 인증 및 권한 부여를 수행합니다. 자세한 내용은 이 패턴의 추가 정보 섹션에 있는 인쇄 검증 및 권한 부여를 참조하세요.
6. LRS VPSX/MFI는 블록 스토리지에 Amazon EBS를 사용합니다. 활성 EC2 인스턴스의 Amazon EBS 데이터를 Amazon S3에 point-in-time 스냅샷으로 백업하고 핫 스탠바이 EBS 볼륨에 복원할 수 있습니다. Amazon EBS 볼륨 스냅샷의 생성, 보존 및 삭제를 자동화하려면 [Amazon Data Lifecycle Manager](#)를 사용하여 자동 스냅샷의 빈도를 설정하고 [RTO/RPO 요구 사항](#)에 따라 복원할 수 있습니다.

도구

서비스

- [Amazon Elastic Block Store\(Amazon EBS\)](#)는 Amazon EC2 인스턴스에 사용할 수 있는 블록 스토리지 볼륨을 제공합니다. EBS 볼륨은 형식이 지정되지 않은 원시 블록 디바이스처럼 동작합니다. 이러한 볼륨을 인스턴스에 디바이스로 마운트할 수 있습니다.
- [Amazon Elastic Compute Cloud\(Amazon EC2\)](#)는 AWS 클라우드에서 확장 가능한 컴퓨팅 용량을 제공합니다. 필요한 만큼 가상 서버를 시작하고 빠르게 스케일 업하거나 스케일 다운할 수 있습니다.

- [Amazon Relational Database Service\(RDS\)](#)는 AWS 클라우드에서 관계형 데이터베이스를 설정, 운영 및 조정하는 데 도움이 됩니다.
- AWS Managed Microsoft Active Directory라고도 하는 [AWS Directory Service for Microsoft Active Directory\(AD\)](#)를 사용하면 디렉터리 인식 워크로드와 AWS 리소스가 AWS의 관리형 Active Directory를 사용할 수 있습니다.

기타 도구

- [LRS VPSX/MFI\(Micro Focus Interface\)](#) – LRS와 Micro Focus가 공동 개발했으며, Micro Focus Enterprise Server JES 스펙에서 출력을 캡처하여 지정된 인쇄 대상으로 안정적으로 전달합니다.
- LRS Directory Information Server(LRS/DIS) – 인쇄 워크플로 중 검증 및 권한 부여에 사용됩니다.
- LRS/Queue는 LRS VPSX/MFI에서 Micro Focus 온라인 Print Exit 프로그래밍 인터페이스를 통해 인쇄 데이터를 수집하거나 캡처하는 데 사용되는 TCP/IP 기반 LRS/Queue 전송 에이전트입니다.
- [Micro Focus Enterprise Server](#)는 메인프레임 애플리케이션을 위한 애플리케이션 배포 환경입니다. Micro Focus Enterprise Developer의 모든 버전을 사용하여 마이그레이션하거나 생성한 메인프레임 애플리케이션의 실행 환경을 제공합니다.

에픽

Amazon EC2에 Micro Focus Enterprise Server를 설치하고 메인프레임 온라인 애플리케이션을 배포

작업	설명	필요한 기술
Micro Focus Enterprise Server를 설치하고 데모 온라인 애플리케이션을 배포하세요.	Amazon EC2에 Micro Focus Enterprise Server를 설치한 다음 Micro Focus 설명서의 자습서: CICS Support 지침에 따라 Amazon EC2에 Micro Focus Account Demo 애플리케이션 (ACCT Demo)을 배포하세요. ACCT Demo 애플리케이션은 인쇄 출력을 생성하고 시작하는 메인프레임 온라인(CICS) 애플리케이션입니다.	클라우드 아키텍트

Amazon EC2에 LRS 인쇄 서버 설치

작업	설명	필요한 기술
<p>인쇄용 LRS 제품 라이선스를 받습니다.</p>	<p>LRS VPSX/MFI, LRS/Queue 및 LRS/DIS에 대한 LRS 제품 라이선스를 받으려면 LRS 출력 관리 팀에 문의하세요. LRS 제품을 설치할 EC2 인스턴스의 호스트 이름을 제공해야 합니다.</p>	<p>리드 구축</p>
<p>LRS VPSX/MFI를 설치할 Amazon EC2 Windows 인스턴스를 생성합니다.</p>	<p>Amazon EC2 설명서의 1단계: 인스턴스 시작의 지침에 따라 Amazon EC2 Windows 인스턴스를 시작합니다. 인스턴스는 LRS VPSX/MFI에 대한 다음 하드웨어 및 소프트웨어 요구 사항을 충족해야 합니다.</p> <ul style="list-style-type: none"> • CPU – 듀얼 코어 • 램 – 16GB • 드라이브 – 500GB • 최소 EC2 인스턴스 – m5.xlarge • OS – Windows/Linux • 소프트웨어 – 인터넷 정보 서비스(IIS) 또는 Apache <p>참고: 위의 하드웨어 및 소프트웨어 요구 사항은 소형 프린터 플릿(약 500~1000대)을 대상으로 합니다. 전체 요구 사항을 알아보려면 LRS 및 AWS 담당자에게 문의하세요.</p>	<p>클라우드 아키텍트</p>

작업	설명	필요한 기술
	<p>Windows 인스턴스를 생성할 때 다음을 수행하세요.</p> <ol style="list-style-type: none"> 1. EC2 호스트 이름이 LRS 제품 라이선스에 사용된 호스트 이름과 동일한지 확인하세요. 2. 다음을 완료하여 Amazon EC2에서 CGI를 활성화하세요. <ol style="list-style-type: none"> a. Amazon EC2 설명서의 2 단계: 인스턴스에 연결 지침에 따라 EC2 인스턴스에 연결합니다. b. Windows 시작 메뉴에서 서버 관리자를 찾아 엽니다. c. 서버 관리자에서 대시보드, 빠른 시작, 역할 및 기능 추가를 선택합니다. 그런 다음 서버 역할을 선택합니다. d. 서버 역할에서 WebServer (IIS) 를 선택한 다음 애플리케이션 개발을 선택합니다. e. 애플리케이션 개발에서 CGI 확인란에 체크합니다. f. CGI를 설치하려면 Windows Server Manager 역할 및 기능 추가 마법사의 지침을 따르세요. 	

작업	설명	필요한 기술
	<p>g. LRS/Queue 통신을 위해 EC2 인스턴스의 Windows 방화벽에 포트 5500을 엽니다.</p>	
<p>EC2 인스턴스에 LRS VPSX/MFI를 설치합니다.</p>	<ol style="list-style-type: none"> 1. Amazon EC2 설명서의 2단계: 인스턴스에 연결 지침에 따라 EC2 인스턴스에 연결합니다. 2. 수신한 LRS 이메일에서 제품 다운로드 페이지 링크를 여세요. 참고: LRS 제품은 전자 파일 전송(EFT)을 통해 배포됩니다. 3. LRS VPSX/MFI를 다운로드하고 파일의 압축을 풉니다 (기본 폴더: c:\LRS). 4. 압축을 푼 폴더에서 LRS 제품 설치 프로그램을 실행하여 LRS VPSX/MFI를 설치합니다. 5. 기능 선택 메뉴에서 VPSX® Server(V1R3.022)를 선택하고 다음을 선택하여 설치 프로세스를 시작합니다. 설치가 완료되면 성공 메시지가 나타납니다. 	<p>클라우드 아키텍트</p>

작업	설명	필요한 기술
LRS/Queue를 설치합니다.	<ol style="list-style-type: none">1. Amazon EC2 설명서의 2단계: 인스턴스에 연결 지침에 따라 Micro Focus Enterprise Server EC2 인스턴스에 연결합니다.2. 수신한 LRS 이메일에서 LRS 제품 다운로드 페이지 링크를 열고 LRS/Queue를 다운로드한 다음 파일의 압축을 풉니다.3. 파일을 다운로드한 위치로 이동한 다음 LRS 제품 설치 프로그램을 실행하여 LRS/Queue를 설치합니다.	클라우드 아키텍트

작업	설명	필요한 기술
LRS/DIS를 설치합니다.	<ol style="list-style-type: none"> 1. Amazon EC2 설명서의 2단계: 인스턴스에 연결 지침에 따라 LRS VPSX/MFI EC2 인스턴스에 연결합니다. 2. 수신한 LRS 이메일에서 LRS 제품 다운로드 페이지 링크를 열고 LRS/DIS를 다운로드한 다음 파일의 압축을 풉니다. 3. 파일을 다운로드한 위치로 이동한 다음 LRS 제품 설치 프로그램을 시작합니다. 4. LRS 제품 설치 프로그램에서 LRS 기타 도구를 확장하고 LRS DIS를 선택한 후 다음을 선택합니다. 5. LRS 제품 설치 프로그램의 나머지 지침을 따라 설치 프로세스를 완료하세요. 	클라우드 아키텍트

작업	설명	필요한 기술
<p>대상 그룹을 생성하고 LRS VPSX/MFI EC2를 대상으로 등록합니다.</p>	<p>Elastic Load Balancing 설명서의 Network Load Balancer에 대한 대상 그룹 생성 지침에 따라 대상 그룹을 생성합니다.</p> <p>대상 그룹을 생성할 때 다음 작업을 수행합니다.</p> <ol style="list-style-type: none"> 1. 그룹 세부 정보 지정 페이지의 대상 유형 선택에서 인스턴스를 선택합니다. 2. 프로토콜에서 TCP를 선택합니다. 3. 포트는 5500을 선택합니다. 4. 대상 등록 페이지의 사용 가능한 인스턴스 섹션에서 LRS VPSX/MFI EC2 인스턴스를 선택합니다. 	<p>클라우드 아키텍트</p>

작업	설명	필요한 기술
Network Load Balancer를 생성합니다.	<p>Elastic Load Balancing 설명서의 Network Load Balancer 생성 지침을 따르세요. Network Load Balancer는 Micro Focus Enterprise Server에서 LRS VPSX/MFI EC2로 트래픽을 라우팅합니다.</p> <p>Network Load Balancer를 생성할 때 리스너 및 라우팅 페이지에서 다음 작업을 수행하세요.</p> <ol style="list-style-type: none"> 1. 프로토콜에서 TCP를 선택합니다. 2. 포트는 5500을 선택합니다. 3. 기본 작업에서 앞서 생성한 대상 그룹에 대해 전달을 선택합니다. 	클라우드 아키텍트

Micro Focus Enterprise Server를 LRS VPSX/MFI 및 LRS/Queue와 통합

작업	설명	필요한 기술
LRS/Queue 통합을 위해 Micro Focus Enterprise Server를 구성하세요.	<ol style="list-style-type: none"> 1. Amazon EC2 설명서의 2단계: 인스턴스에 연결 지침에 따라 Micro Focus Enterprise Server EC2 인스턴스에 연결합니다. 2. Windows 시작 메뉴에서 Micro Focus Enterprise Server 관리 UI를 엽니다. 3. 메뉴 모음에서 네이티브를 선택합니다. 	클라우드 아키텍트

작업	설명	필요한 기술
	<ol style="list-style-type: none"> 4. 탐색 창에서 디렉토리 서버를 선택한 다음 BANKDEMO 또는 사용자의 Enterprise 서버 리전을 선택합니다. 5. 왼쪽 탐색 창의 일반에 추가 섹션으로 스크롤하여 LRSQ를 가리키도록 환경 변수(LRSQ_ADDRESS, LRSQ_PORT, LRSQ_COMMAND)를 구성합니다. 6. LRSQ_ADDRESS의 경우 이전에 생성한 Network Load Balancer의 IP 주소 또는 DNS 이름을 입력합니다. 7. LRSQ_PORT의 경우 VPSX LRSQ 리스너 포트(5500)를 입력합니다. 8. LRSQ_COMMAND의 경우 LRSQ 실행 파일의 경로 위치를 입력합니다. 9. 참고: LRS는 현재 DNS 이름에 최대 50자 제한을 지원하지만 향후 변경될 수 있습니다. DNS 이름이 50자보다 길면 Network Load Balancer의 IP 주소를 대안으로 사용할 수 있습니다. 	

작업	설명	필요한 기술
<p>Micro Focus Enterprise Server 초기화에 CICS Print Exit(DFHUPRNT)를 사용할 수 있도록 하세요.</p>	<ol style="list-style-type: none"> 1. Amazon EC2 설명서의 2단계: 인스턴스에 연결 지침에 따라 Micro Focus Enterprise Server EC2 인스턴스에 연결합니다. 2. 이름이 VPSX_MFI_R2 로 지정된 LRS VPSX/MFI 실행 파일 폴더에서 CICS Print Exit(DFHUPRNT)를 Micro Focus Enterprise Server EC2 인스턴스 위치로 복사합니다. 32비트 시스템의 경우 위치는 C:\Program Files (x86) \Micro Focus\Enterprise Server\bin 입니다. 64 비트 시스템의 경우 위치는 C:\Program Files (x86) \Micro Focus \Enterprise Server \bin64 입니다. 참고: 복사할 때 DFHUPRNT_64.dll 파일 이름을 DFHUPRNT.dll 로 바꿔야 합니다. <p>Micro Focus Enterprise Server 가 CICS Print Exit(DFHUPRNT)를 감지했는지 확인</p> <ol style="list-style-type: none"> 1. Micro Focus Enterprise Server를 중지하고 시작합니다. 2. Micro Focus Enterprise Server의 관리 패널에서 모 	<p>클라우드 아키텍트</p>

작업	설명	필요한 기술
	<p>니터, 로그, 콘솔 로그를 엽니다.</p> <p>3. 콘솔 로그에서 '3270 프린터 사용자 종료 DFHUPRNT가 성공적으로 설치되었습니다.'라는 메시지를 확인하세요.</p>	

작업	설명	필요한 기술
<p>CICS 프린터의 터미널 ID(TERMIDs)를 Micro Focus Enterprise Server로 정의합니다.</p>	<p>Micro Focus Enterprise Server에서 3270 인쇄를 활성화</p> <ol style="list-style-type: none"> 1. Micro Focus Enterprise Server의 관리 패널에서 CICS, 리소스, 그룹별을 엽니다. 2. 왼쪽 탐색 패널에서 SIT(시스템 초기화 테이블)를 선택한 다음 BNKCICV를 선택합니다. 3. 일반 섹션에서 3270까지 아래로 스크롤한 다음 3270 인쇄 확인란에 체크합니다. <p>Micro Focus Enterprise Server에서 CICS 프린터의 터미널을 정의</p> <ol style="list-style-type: none"> 1. Micro Focus Enterprise Server의 관리 패널에서 CICS, 리소스, 유형별을 엽니다. 2. 왼쪽의 탐색 창에서를 기간을 선택한 다음, 신규를 선택합니다. 터미널 리소스 생성 양식이 열립니다. 3. 이름에는 LRS Print Queue의 이름을 입력합니다. (참고: 이 패턴은 CICS 프린터의 터미널 ID로 'P275'와 LRS VPSX Print Queue를 사용합니다.) 	<p>클라우드 아키텍트</p>

작업	설명	필요한 기술
	<ol style="list-style-type: none"> 4. 그룹에는 BANKTERM을 입력합니다. 5. 자동 설치 - 모델의 경우 NO를 입력합니다. 6. 터미널 식별자 - 터미널 유형에는 DFHPRT32를 입력합니다. 7. 넷 이름에는 VTAMP275를 입력합니다. 8. 터미널 사용의 경우 서비스 중 확인란에 체크합니다. 9. 페이지 상단으로 스크롤한 다음 저장을 선택합니다. 10. 설치를 선택합니다. 설치 성공 메시지가 팝업 메시지로 표시됩니다. 	

Micro Focus Enterprise Server 및 LRS VPSX/MFI에서 프린터와 프린트 사용자를 설정

작업	설명	필요한 기술
<p>LRS VPSX에서 인쇄 대기열을 생성합니다.</p>	<ol style="list-style-type: none"> 1. Amazon EC2 설명서의 2단계: 인스턴스에 연결 지침에 따라 LRS VPSX/MFI EC2 인스턴스에 연결합니다. 2. Windows 시작 메뉴에서 VPSX 웹 인터페이스를 엽니다. 3. 탐색 창에서 프린터를 선택합니다. 4. 추가를 선택한 다음, 프린터 추가를 선택합니다. 	<p>클라우드 아키텍트</p>

작업	설명	필요한 기술
	<p>5. 프린터 구성페이지에서 프린터 이름에 P275를 입력합니다.</p> <p>6. VPSX ID의 경우 VPS1을 입력합니다.</p> <p>7. 의 CommType 경우 TCP/PIP/LRSQ를 선택합니다.</p> <p>8. 호스트/IP 주소에는 추가하려는 물리적 프린터의 IP 주소를 입력합니다.</p> <p>9. 장치에 장치 이름을 입력합니다.</p> <p>10.Windows 드라이버 또는 Linux/Mac 드라이버를 선택합니다.</p> <p>11.추가를 선택합니다.</p> <p>참고: 인쇄 대기열은 Micro Focus Enterprise Server에서 만든 Print TERMIDs와 동일해야 합니다.</p>	

작업	설명	필요한 기술
LRS VPSX/MFI에서 프린트 사용자를 생성하세요.	<ol style="list-style-type: none"> 1. Amazon EC2 설명서의 2단계: 인스턴스에 연결 지침에 따라 LRS VPSX/MFI EC2 인스턴스에 연결합니다. 2. Windows 시작 메뉴에서 VPSX 웹 인터페이스를 엽니다. 3. 탐색 창에서 보안을 선택한 다음 사용자를 선택합니다. 4. 사용자 이름 열에서 관리자를 선택한 다음 복사를 선택합니다. 5. 사용자 프로필 유지 관리 창에서 사용자 이름에 사용자 이름 (예:) 을 입력합니다. PrintUser 6. 설명에 간단한 설명(예: 테스트 인쇄용 사용자)을 입력합니다. 7. 업데이트를 선택합니다. 그러면 인쇄 사용자 (예: PrintUser) 가 생성됩니다. 8. 탐색 창의 사용자에서 생성한 새 사용자를 선택합니다. 9. 명령 메뉴에서 보안을 선택합니다. 10. 보안 규칙 페이지에서 해당하는 프린터 보안 및 작업 보안 옵션을 모두 선택한 다음 저장을 선택합니다. 11. 새 인쇄 사용자를 관리자 그룹에 추가하려면 탐색 창으 	클라우드 아키텍트

작업	설명	필요한 기술
	<p>로 이동하여 보안을 선택한 다음 구성을 선택합니다.</p> <p>12보안 구성 창에서 관리자 열에 새 인쇄 사용자를 추가합니다.</p>	

인쇄 인증 및 권한 부여 설정

작업	설명	필요한 기술
<p>사용자 및 그룹을 사용하여 AWS Managed Microsoft AD 도메인을 생성합니다.</p>	<ol style="list-style-type: none"> 1. AWS Directory Service 설명서의 AWS Managed Microsoft AD 디렉터리 생성 지침에 따라 AWS Managed Microsoft AD에 Active Directory를 생성합니다. 2. AWS Directory Service 설명서의 3단계: AWS Managed Microsoft AD를 관리하기 위한 EC2 인스턴스 배포 지침에 따라 EC2 인스턴스(Active Directory 관리자)를 배포하고 Active Directory 도구를 설치하여 AWS Managed Microsoft AD를 관리합니다. 3. Amazon EC2 설명서의 2단계: 인스턴스에 연결 지침에 따라 EC2 인스턴스에 연결합니다. 참고: EC2 인스턴스에 연결할 때 Windows 보안 창에 관리자 보안 인증 정보(1단계에서 생성한 디렉터리의)를 입력합니다. 	<p>클라우드 아키텍트</p>

작업	설명	필요한 기술
	<p>4. Windows 시작 메뉴에서 Windows 관리 도구 아래에서 Active Directory 사용자 및 컴퓨터를 선택합니다.</p> <p>5. AWS Directory Service 설명서의 사용자 생성 단계에 따라 Active Directory 도메인에서 인쇄 사용자를 생성합니다.</p>	
<p>AWS Managed Microsoft AD 도메인에 LRS VPSX/MFI EC2를 조인합니다.</p>	<p>LRS VPSX/MFI EC2를 AWS Managed Microsoft AD 도메인에 자동으로(AWS 지식 센터 설명서) 또는 수동으로(AWS Directory Service 설명서) 조인하세요.</p>	<p>클라우드 아키텍트</p>

작업	설명	필요한 기술
AWS Managed Microsoft AD 와 LRS/DIS를 구성 및 통합합니다.	<ol style="list-style-type: none"> 1. Amazon EC2 설명서의 2단계: 인스턴스에 연결 지침에 따라 LRS VPSX/MFI EC2 인스턴스에 연결합니다. 2. Windows 시작 메뉴에서 VPSX 웹 인터페이스를 엽니다. 3. 탐색 창에서 보안을 선택한 다음 구성을 선택합니다. 4. 보안 구성 페이지의 보안 파라미터 섹션에서 보안 유형에 대해 내부를 선택합니다. 5. 보안 파라미터 섹션의 나머지 옵션에 원하는 설정을 입력합니다. 6. Microsoft Windows 시작 메뉴에서 LRS 출력 관리 폴더를 열고 서버 시작을 선택한 다음 서버 중지를 선택합니다. 7. Active Directory 사용자 이름 및 암호를 사용하여 LRS VPSX/MFI에 로그인합니다. 	클라우드 아키텍트

온라인 인쇄 워크플로 테스트

작업	설명	필요한 기술
Micro Focus ACCT Demo 앱에서 온라인 인쇄 요청을 시작하세요.	1. Micro Focus Enterprise Server EC2 인스턴스에서 TN3270 터미널 에뮬레이터를 엽니다. (참고: 이 패턴은	클라우드 아키텍트

작업	설명	필요한 기술
	<p>3270 터미널 에뮬레이터를 사용합니다.)</p> <ol style="list-style-type: none"> 2. TN3270 터미널 에뮬레이터(Rumba)에 연결합니다. 호스트 이름 주소인 경우 127.0.0.1을 사용합니다. Telnet 포트의 경우 9270을 사용합니다. 3. 3270 화면에 연결한 후 CTL+SHIFT+Z를 눌러 화면을 지웁니다. 4. ACCT Demo 애플리케이션을 시작하려면 투명 화면에서 ACCT를 입력합니다. 그러면 ACCT Demo 온라인(CICS) 애플리케이션 메인 화면이 열립니다. 참고: 기본 화면에는 계정 파일, 이름으로 검색, 입력, 요청 유형, 계정 및 프린터와 같은 메뉴 옵션이 있습니다. 5. ACCT Demo 온라인(CICS) 애플리케이션에서 인쇄 요청을 제출하려면 요청 유형 필드에 P, 계정 필드에 11111, 프린터 필드에 P275를 입력합니다. 프린터 필드의 값을 CICS 프린터의 터미널 ID 값으로 설정해야 합니다. 6. 입력 키를 누릅니다. 	

작업	설명	필요한 기술
	<p>화면 아래쪽에 '인쇄 요청 예약됨' 메시지가 나타납니다. 이는 ACCT Demo 애플리케이션에서 온라인 인쇄 요청이 생성되어 인쇄 처리를 위해 LRS VPS/MFI로 전송되었음을 확인합니다.</p>	

작업	설명	필요한 기술
<p>LRS VPSX/MFI에서 인쇄 출력을 확인합니다.</p>	<ol style="list-style-type: none"> 1. Amazon EC2 설명서의 2단계: 인스턴스에 연결 지침에 따라 LRS VPSX/MFI EC2 인스턴스에 연결합니다. 2. Windows 시작 메뉴에서 VPSX 웹 인터페이스를 엽니다. 3. 탐색 창에서 프린터를 선택한 다음 출력 대기열을 선택합니다. 이전에 온라인 인쇄용으로 만든 P275 인쇄 대기열을 찾으세요. 4. 인쇄 대기열(P275)의 경우 스포 ID 열에서 프린터 대기열에 있는 요청의 스포 ID를 선택합니다. 5. 작업 탭의 명령 열에서 찾아보기를 선택합니다. <p>이제 계정 번호, 성, 이름, 주소, 전화 번호, 카드 발급 번호, 발급일, 금액, 잔액 등의 열이 있는 계정 명세서의 인쇄 결과를 볼 수 있습니다.</p> <p>예제를 확인하려면 이 패턴에 대한 <code>online_print_output</code> 첨부 파일을 참조하세요.</p>	<p>테스트 엔지니어</p>

관련 리소스

- [LRS 출력 현대화](#)(LRS 문서)
- [VTAM 네트워킹 개념](#)(IBM 설명서)

- [논리 유닛\(LU\) 유형 요약](#)(IBM 설명서)
- [ANSI 및 머신 캐리지 제어](#)(IBM 설명서)
- [Micro Focus를 통한 AWS 기반 엔터프라이즈 메인프레임 워크로드 강화](#)(AWS 파트너 네트워크 블로그)
- [Amazon EC2 Auto Scaling 및 Systems Manager를 사용하여 Micro Focus Enterprise Server PAC 구축](#)(AWS 권장 가이드 설명서)
- [Advanced Function Presentation\(AFP\) 데이터 스트림](#)(IBM 설명서)
- [Line Conditioned Data Stream\(LCDS\)](#)(설명서 첨부)

추가 정보

고려 사항

현대화 과정에서 메인프레임 온라인 프로세스와 해당 프로세스에서 생성되는 출력에 대한 다양한 구성을 고려할 수 있습니다. 메인프레임 플랫폼은 인쇄에 직접적인 영향을 미치는 특정 요구 사항에 따라 이를 사용하는 모든 고객 및 공급업체에 의해 맞춤화되었습니다. 예를 들어, 현재 플랫폼은 IBM Advanced Function Presentation(AFP) 또는 Xerox Line Condition Data Stream(LCDS)을 현재 워크플로에 통합할 수 있습니다. 또한 [메인프레임 캐리지 제어 문자](#)와 [채널 명령어](#)는 인쇄된 페이지의 모양에 영향을 줄 수 있으며 특별한 처리가 필요할 수 있습니다. 현대화 계획 프로세스의 일환으로 사용자의 인쇄 환경의 구성을 평가하고 이해하는 것이 좋습니다.

인쇄 데이터 캡처

이 섹션에서는 IBM 메인프레임 환경에서 인쇄에 사용할 수 있는 CICS 애플리케이션 프로그래밍 방법을 요약합니다. LRS VPSX/MFI 구성 요소는 동일한 애플리케이션 프로그램이 동일한 방식으로 데이터를 생성할 수 있도록 하는 기술을 제공합니다. 다음 테이블에는 LRS VPSX/MFI 인쇄 서버가 있는 AWS 및 Micro Focus Enterprise Server에서 실행되는 현대화된 CICS 애플리케이션에서 각 애플리케이션 프로그래밍 방법이 어떻게 지원되는지 설명되어 있습니다.

메서드	설명	현대화된 환경에서의 방법 지원
EXEC CICS SEND TEXT.. 또는 EXEC CICS SEND MAP..	이러한 CICS 및 VTAM 방법은 3270/SCS 인쇄 데이터 스트림을 생성하여 LUTYPE0, LUTYPE1 및 LUTYPE3 인쇄	이러한 방법 중 하나를 사용하여 3270/SCS 인쇄 데이터 스트림을 생성할 때 Micro Focus 온라인 Print Exit(DFHUPRNT) 애

장치에 전달하는 역할을 합니다.

폴리케이션 프로그래밍 인터페이스(API)를 사용하면 VPSX/MFI에서 인쇄 데이터를 처리할 수 있습니다.

EXEC CICS SEND TEXT..
또는 EXEC CICS SEND MAP.. (타사 IBM 메인프레임 소프트웨어 사용)

이 CICS 및 VTAM 방법은 3270/SCS 인쇄 데이터 스트림을 생성하여 LUTYPE0, LUTYPE1 및 LUTYPE3 인쇄 장치에 전달하는 역할을 합니다. 타사 소프트웨어 제품은 인쇄 데이터를 가로채서 데이터를 ASA/MCH 제어 문자가 있는 표준 인쇄 형식 데이터로 변환한 다음 JES를 사용하는 메인프레임 기반 인쇄 시스템에서 처리할 수 있도록 JES 스푼에 데이터를 저장합니다.

이러한 방법 중 하나를 사용하여 3270/SCS 인쇄 데이터 스트림을 생성할 때 Micro Focus 온라인 Print Exit(DFHUPRNT) API를 사용하면 VPSX/MFI에서 인쇄 데이터를 처리할 수 있습니다.

EXEC CICS SPOOLOPEN

이 방법은 CICS 애플리케이션에서 JES 스푼에 데이터를 직접 쓰는 데 사용됩니다. 그러면 JES를 사용하는 메인프레임 기반 인쇄 시스템에서 데이터를 처리할 수 있게 됩니다.

Micro Focus Enterprise Server는 데이터를 Enterprise Server 스푼로 스푼링하며, 여기서 VPSX로 데이터를 스푼하는 VPSX/MFI Batch Print Exit(LRSPRTE6)로 처리할 수 있습니다.

DRS/API

LRS에서 제공하는 프로그래밍 인터페이스는 JES에 인쇄 데이터를 쓰는 데 사용됩니다.

VPSX/MFI는 인쇄 데이터를 VPSX로 직접 스푼하는 대체 인터페이스를 제공합니다.

프린터 플릿 상태 확인

LRS VPSX/MFI(LRS LoadX)는 장치 관리 및 운영 최적화를 포함한 심층 상태 확인을 수행할 수 있습니다. 장치 관리는 프린터 장치의 오류를 감지하고 인쇄 요청을 정상 프린터로 라우팅할 수 있습니다. 프린터 플릿의 심층 상태 확인에 대한 자세한 내용은 제품 라이선스에 포함된 LRS 설명서를 참조하세요.

인쇄 인증 및 권한 부여

LRS/DIS를 사용하면 LRS 애플리케이션이 Microsoft Active Directory 또는 LDAP 서버를 사용하여 사용자 ID와 암호를 검증할 수 있습니다. 기본 인쇄 승인 외에도 LRS/DIS는 다음과 같은 사용 사례에서 세분화된 수준의 인쇄 보안 제어를 적용할 수 있습니다.

- 프린터 작업을 탐색하는지 관리할 수 있습니다.
- 다른 사용자 작업의 탐색 수준을 관리할 수 있습니다.
- 운영 작업을 관리할 수 있습니다. 보류/해제, 삭제, 수정, 복사, 재라우팅과 같은 명령 수준 보안을 예로 들 수 있습니다. 보안은 User-ID 또는 그룹(AD 그룹 또는 LDAP 그룹과 유사)으로 설정할 수 있습니다.

첨부

이 문서와 관련된 추가 콘텐츠에 액세스하려면 [attachment.zip](#) 파일의 압축을 풉니다.

Transfer Family를 사용하여 메인프레임 파일을 Amazon S3로 직접 이동

작성자: Luis Gustavo Dantas(AWS)

환경: 프로덕션	소스: 메인프레임	대상: Amazon S3
R 유형: 해당 없음	워크로드: IBM	기술: 메인프레임, 스토리지 및 백업, 현대화
AWS 서비스: AWS Transfer Family, Amazon S3		

요약

현대화 여정의 일환으로 온프레미스 서버와 Amazon Web Services(AWS) 클라우드 간에 파일을 전송해야 하는 어려움을 겪을 수 있습니다. 일반적으로 메인프레임은 Amazon Simple Storage Service(S3), Amazon Elastic Block Store(Amazon EBS) 또는 Amazon Elastic File System(Amazon EFS)과 같은 최신 데이터 스토어에 액세스할 수 없기 때문에 메인프레임에서 데이터를 전송하는 것은 매우 어려운 일입니다.

많은 고객이 온프레미스 Linux, Unix 또는 Windows 서버와 같은 중간 스테이징 리소스를 사용하여 파일을 AWS 클라우드로 전송합니다. AWS Transfer Family를 Secure Shell(SSH) File Transfer Protocol(SFTP)과 함께 사용하여 메인프레임 파일을 Amazon S3에 직접 업로드하면 이러한 간접적인 방법을 피할 수 있습니다.

사전 조건 및 제한 사항

사전 조건

- 활성 상태의 AWS 계정
- 기존 플랫폼에서 연결할 수 있는 서브넷이 있는 Virtual Private Cloud (VPC)
- VPC용 Transfer Family 엔드포인트
- 순차적인 [고정 길이의 파일](#)로 변환된 메인프레임 가상 스토리지 액세스 방법(VSAM) 파일(IBM 설명서)

제한 사항

- SFTP는 기본적으로 바이너리 모드로 파일을 전송합니다. 즉, EBCDIC 인코딩이 유지된 상태로 파일이 Amazon S3에 업로드됩니다. 파일에 바이너리 또는 압축 데이터가 포함되어 있지 않은 경우 [sftp ascii 하위 명령](#)(IBM 설명서)을 사용하여 전송 중에 파일을 텍스트로 변환할 수 있습니다.
- 대상 환경에서 이러한 파일을 사용하려면 압축 및 바이너리 콘텐츠가 포함된 [메인프레임 파일\(AWS 권장 가이드\)의 압축을 풀어야](#) 합니다.
- Amazon S3 객체의 크기는 최소 0바이트에서 최대 5TB까지 다양합니다. Amazon S3 기능에 대한 자세한 내용은 [Amazon S3 FAQ](#)를 참조하세요.

아키텍처

소스 기술 스택

- 작업 제어 언어(JCL)
- z/OS Unix 셸 및 ISPF
- SFTP
- VSAM 및 플랫 파일

대상 기술 스택

- Transfer Family
- Amazon S3
- Amazon Virtual Private Cloud(VPC)

대상 아키텍처

다음 다이어그램은 SFTP와 함께 Transfer Family를 사용하여 메인프레임 파일을 S3 버킷에 직접 업로드하기 위한 참조 아키텍처를 보여줍니다.

이 다이어그램은 다음 워크플로를 보여줍니다.

1. JCL 작업을 사용하면 Direct Connect를 통해 메인프레임 파일을 기존 메인프레임에서 AWS 클라우드로 전송할 수 있습니다.

2. Direct Connect를 사용하면 네트워크 트래픽이 AWS 글로벌 네트워크에 남아 있고 퍼블릭 인터넷을 우회할 수 있습니다. 또한 Direct Connect는 네트워크 속도를 50Mbps부터 최대 100Gbps까지 향상 시킵니다.
3. VPC 엔드포인트를 사용하면 공용 인터넷을 사용하지 않고도 VPC 리소스와 지원되는 서비스를 연결할 수 있습니다. Transfer Family와 Amazon S3에 대한 액세스는 두 개의 프라이빗 서브넷과 가용 영역에 위치한 탄력적 네트워크 인터페이스를 통해 이루어져고가용성을 달성합니다.
4. Transfer Family는 사용자를 인증하고, SFTP를 사용하여 기존 환경에서 파일을 수신하고, 이를 S3 버킷으로 옮깁니다.

자동화 및 규모 조정

Transfer Family 서비스가 도입되면 JCL 작업을 SFTP 클라이언트로 사용하여 메인프레임에서 Amazon S3로 파일을 무제한으로 전송할 수 있습니다. 메인프레임 파일을 전송할 준비가 되면 SFTP 작업을 실행할 메인프레임 배치 작업 스케줄러를 사용하여 파일 전송을 자동화할 수도 있습니다.

도구

- [Amazon Simple Storage Service\(S3\)](#)는 원하는 양의 데이터를 저장, 보호 및 검색하는 데 도움이 되는 클라우드 기반 객체 스토리지 서비스입니다.
- [Amazon Virtual Private Cloud\(VPC\)](#)를 이용하면 사용자가 정의한 가상 네트워크로 AWS 리소스를 시작할 수 있습니다. 이 가상 네트워크는 사용자의 자체 데이터 센터에서 운영하는 기존 네트워크와 유사하며 AWS의 확장 가능한 인프라를 사용한다는 이점이 있습니다.
- [AWS Transfer Family](#)를 사용하면 SFTP, FTPS 및 FTP 프로토콜을 사용하여 Amazon S3 및 Amazon EFS로 반복적으로 전송되는 business-to-business 파일을 안전하게 확장할 수 있습니다.

모범 사례

< 작성자: 참고 사항 삭제: 사용자가 이 패턴을 보다 효과적으로 구현하는 데 도움이 될 수 있는 지침 및 권장 사항 목록을 제공하십시오. >

에픽

S3 버킷 및 액세스 정책 생성

작업	설명	필요한 기술
S3 버킷을 생성합니다.	레거시 환경에서 전송하는 파일을 호스팅할 S3 버킷을 생성 합니다.	일반 AWS
IAM 역할 및 정책을 생성합니다.	<p>Transfer Family는 사용자의 AWS Identity and Access Management(IAM) 역할을 사용해 이전에 생성한 S3 버킷에 대한 액세스 권한을 부여합니다.</p> <p>다음 IAM 정책을 포함하는 IAM 정책을 생성합니다.</p> <pre data-bbox="594 1035 1029 1885"> { "Version": "2012-10-17", "Statement": [{ "Sid": "UserFolderListing", "Action": ["s3:ListBucket", "s3:GetBucketLocation"], "Effect": "Allow", "Resource": ["arn:aws:s3:::<your-bucket-name>"] }] } </pre>	일반 AWS

작업	설명	필요한 기술
	<pre>] }, { "Sid": "HomeDirObjectAcce ss", "Effect": "Allow", "Action": ["s3:PutObject", "s3:GetObjectAcl", "s3:GetObject", "s3:DeleteObjectVe rsion", "s3:DeleteObject", "s3:PutObjectAcl", "s3:GetObjectVersion"], "Resource": "arn:aws:s3:::<your- bucket-name>/*" }] } </pre> <p>참고: IAM 역할을 생성할 때 Transfer 사용 사례를 선택해야 합니다.</p>	

전송 서비스 정의

작업	설명	필요한 기술
SFTP 서버를 생성합니다.	<ol style="list-style-type: none"> 1. AWS Management 콘솔에 로그인하고 Transfer Family 콘솔을 연 다음 서버 생성을 선택합니다. 2. SFTP(SSH 파일 전송 프로토콜) Secure Shell 프로토콜을 통한 파일 전송만 선택한 후 다음을 선택합니다. 3. ID 공급자의 경우 서비스 관리자를 선택하고 다음을 선택합니다. 4. 엔드포인트 유형 편집에서 VPC 호스팅을 선택합니다. 5. 액세스에서 내부를 선택합니다. 6. VPC에서 VPC를 선택합니다. 7. 가용 영역 섹션에서 가용 영역 및 관련 서브넷을 선택합니다. 8. 보안 그룹 섹션에서 보안 그룹을 선택하고 나서 다음을 선택합니다. 9. 도메인의 경우 Amazon S3를 선택한 후 다음을 선택합니다. 10. 추가 세부 정보 구성 페이지의 기본 옵션을 그대로 두고 다음을 선택합니다. 11. 서버 생성을 선택합니다. 	일반 AWS

작업	설명	필요한 기술
	<p>참고: SFTP 서버를 설정하는 방법에 대한 자세한 내용은 SFTP 지원 서버 생성(AWS Transfer Family 사용 설명서)을 참조하세요.</p>	
<p>서버 주소를 가져옵니다.</p>	<ol style="list-style-type: none"> 1. Transfer Family 콘솔을 열고 서버 ID 열에서 서버 ID를 선택합니다. 2. 엔드포인트 세부 정보 섹션의 엔드포인트 유형에서 엔드포인트 ID를 선택합니다. 이로 인해 Amazon VPC 콘솔로 돌아갑니다. 3. Amazon VPC 콘솔의 세부 정보 탭에서 DNS 이름 옆에 있는 DNS 이름을 찾으세요. 	<p>일반 AWS</p>
<p>SFTP 클라이언트 키 페어를 생성합니다.</p>	<p>Microsoft Windows 또는 macOS/Linux/UNIX용 SSH 키 페어를 생성합니다.</p>	<p>일반 AWS, SSH</p>

작업	설명	필요한 기술
SFTP 사용자를 생성합니다.	<ol style="list-style-type: none"> 1. Transfer Family 콘솔을 열고 탐색 창에서 서버를 선택한 다음 서버를 선택합니다. 2. 서버 ID 옆에서 서버의 서버 ID를 선택한 다음 사용자 추가를 선택합니다. 3. 사용자 이름에 SSH 키 페어 사용자 이름과 일치하는 사용자 이름을 입력합니다. 4. 역할에는 앞에서 생성한 IAM 역할을 선택합니다. 5. 홈 디렉터리의 경우 이전에 생성한 S3 버킷을 선택합니다. 6. SSH 퍼블릭 키의 경우, 이전에 생성한 키 페어를 입력합니다. 7. 추가를 선택합니다. 	일반 AWS

메인프레임 파일 전송

작업	설명	필요한 기술
SSH 개인 키를 메인프레임으로 전송하세요.	<p>SFTP 또는 SCP를 사용하여 SSH 개인 키를 기존 환경에 전송하세요.</p> <p>SFTP 예제:</p> <pre>sftp [USERNAME@mainframeIP] [password] cd [/u/USERNAME]</pre>	메인프레임, z/OS Unix 셸, FTP, SCP

작업	설명	필요한 기술
	<pre data-bbox="597 205 1026 306">put [your-key-pair-file]</pre> <p data-bbox="597 340 740 378">SCP 예제:</p> <pre data-bbox="597 415 1026 575">scp [your-key-pair-file] [USERNAME@MainframeIP]:/[u/USERNAME]</pre> <p data-bbox="597 609 1026 835">그런 다음 나중에 파일 전송 배치 작업(예: /u/CONTROLM)을 실행할 사용자 이름으로 z/OS Unix 파일 시스템에 SSH 키를 저장합니다.</p> <p data-bbox="597 877 1026 1012">참고: z/OS Unix 셸에 대한 자세한 내용은 z/OS 셸 소개(IBM 설명서)를 참조하세요.</p>	

작업	설명	필요한 기술
<p>JCL SFTP 클라이언트를 생성합니다.</p>	<p>메인프레임에는 네이티브 SFTP 클라이언트가 없으므로 BPXBATCH 유틸리티를 사용하여 z/OS Unix 셸에서 SFTP 클라이언트를 실행해야 합니다.</p> <p>ISPF 편집기에서 JCL SFTP 클라이언트를 생성합니다. 예:</p> <pre data-bbox="597 663 1027 1619"> //JOBNAM JOB ... //***** ***** ***** ***** **** //SFTP EXEC PGM=BPXB TCH,REGION=0M //STDPARM DD * SH cp "//'MAINF RAME.FILE.NAME'" filename.txt; echo 'put filename.txt' > uplcmd; sftp -b uplcmd -i ssh_private_key_fi le ssh_username@<tran sfer service ip or DNS>; //SYSPRINT DD SYSOUT=* //STDOUT DD SYSOUT=* //STDENV DD * //STDERR DD SYSOUT=* </pre> <p>참고: z/OS Unix 셸에서 명령을 실행하는 방법에 대한 자세한 내용은 BPXBATCH 유틸리티(IBM 설명서)를 참조하세요.</p>	<p>JCL, 메인프레임, z/OS Unix 셸</p>

작업	설명	필요한 기술
	<p>z/OS에서 JCL 작업을 만들거나 편집하는 방법에 대한 자세한 내용은 ISPF란?과 ISPF 편집기(IBM 설명서)를 참조하세요.</p>	
<p>JCL SFTP 클라이언트를 실행합니다.</p>	<ol style="list-style-type: none"> 1. ISPF 편집기에서 SUB를 입력한 다음 JCL 작업이 생성되면 ENTER 키를 누릅니다. 2. SDSF에서 메인프레임의 파일 전송 배치 작업 활동을 모니터링합니다. <p>참고: 배치 작업의 활동을 확인하는 방법에 대한 자세한 내용은 z/OS SDSF 사용 설명서(IBM 설명서)를 참조하세요.</p>	<p>메인프레임, JCL, ISPF</p>
<p>파일 전송을 확인합니다.</p>	<ol style="list-style-type: none"> 1. AWS Management 콘솔에 로그인하고 Amazon S3 콘솔을 연 다음, 탐색 창에서 버킷을 선택합니다. 2. Transfer Family와 연결된 버킷을 선택하세요. 3. 객체 탭의 객체 섹션에서 메인프레임에서 전송한 파일을 찾습니다. 	<p>일반 AWS</p>

작업	설명	필요한 기술
<p>JCL SFTP 클라이언트를 자동화합니다.</p>	<p>작업 스케줄러를 사용하여 JCL SFTP 클라이언트를 자동으로 트리거합니다.</p> <p>참고: BMC Control-M 또는 CA Workload Automation과 같은 메인프레임 작업 스케줄러를 사용하여 시간 및 기타 배치 작업 종속성을 기반으로 파일 전송을 위한 배치 작업을 자동화할 수 있습니다.</p>	<p>작업 스케줄러</p>

관련 리소스

- [AWS Transfer Family의 작동 방식](#)
- [AWS를 통한 메인프레임 현대화](#)

대규모 Db2 z/OS 데이터를 CSV 파일로 Amazon S3에 전송

작성자: Bruno Sahinoglu(AWS), Ivan Schuster(AWS), Abhijit Kshirsagar(AWS)

코드 리포지토리: DB2 z/OS를 S3로 언로드	환경: 프로덕션	소서: Db2
대상: Amazon S3	R 유형: 리플랫폼	워크로드: IBM
기술: 메인프레임, 데이터 레이크, 데이터베이스, 소프트웨어 개발 및 테스트, 마이그레이션	AWS 서비스: Amazon Aurora, AWS Glue, Amazon S3, AWS Transfer Family, Amazon Athena	

요약

메인프레임은 여전히 많은 기업에서 기록 시스템으로 사용되고 있으며, 여기에는 현재 및 과거 비즈니스 거래 기록이 있는 마스터 데이터 엔티티를 비롯한 방대한 양의 데이터가 포함되어 있습니다. 하지만 정보가 사일로화되어 같은 기업 내의 분산 시스템에서 쉽게 액세스할 수 없는 경우가 많습니다. 클라우드 기술이 등장하고 빅 데이터가 대중화되면서 기업들은 메인프레임 데이터에 숨겨진 통찰력을 활용하여 새로운 비즈니스 역량을 개발하는 데 관심을 갖고 있습니다.

이러한 목표를 달성하기 위해 기업은 메인프레임 Db2 데이터를 Amazon Web Services(AWS) 클라우드 환경에 공개하려고 합니다. 비즈니스상의 이유는 여러 가지이며 전송 방법은 사례마다 다릅니다. 애플리케이션을 메인프레임에 직접 연결하는 것을 선호할 수도 있고 데이터를 거의 실시간으로 복제하는 것을 선호할 수도 있습니다. 사용 사례가 데이터 웨어하우스나 데이터 레이크에 데이터를 공급하는 것이라면 up-to-date 복사본을 보유하는 것은 더 이상 문제가 되지 않으며, 특히 타사 제품 라이선스 비용을 피하려는 경우에는 이 패턴에 설명된 절차만으로도 충분할 수 있습니다. 또 다른 사용 사례는 마이그레이션 프로젝트를 위한 메인프레임 데이터 전송을 들 수 있습니다. 마이그레이션 시나리오에서는 기능적 동등성 테스트를 수행하려면 데이터가 필요합니다. 이 기사물에 설명된 접근 방식은 Db2 데이터를 AWS 클라우드 환경으로 전송하는 비용 효율적인 방법입니다.

Amazon Simple Storage Service (Amazon S3) 는 가장 통합된 AWS 서비스 중 하나이므로 Amazon Athena, AWS Lambda 함수 또는 Amazon과 같은 다른 AWS 서비스를 사용하여 거기에서 데이터에 액세스하고 직접 통찰력을 수집할 수 있습니다. QuickSight AWS Glue 또는 AWS Database Migration Service(AWS DMS)를 사용하여 Amazon Aurora 또는 Amazon DynamoDB로 데이터를 로드할 수 있습니다.

니다. 이를 옆두에 두고 메인프레임에서 ASCII 형식의 CSV 파일에 있는 Db2 데이터를 언로드하고 이 파일을 Amazon S3로 전송하는 방법을 설명합니다.

이를 위해 필요한 만큼 Db2 테이블을 언로드하고 전송하는 작업 제어 언어(JCL)를 생성하는 데 도움이 되는 [메인프레임 스크립트](#)가 개발되었습니다.

사전 조건 및 제한 사항

사전 조건

- 재구성된 확장 실행기(REXX) 및 JCL 스크립트를 실행할 권한이 있는 IBM z/OS 운영 체제 사용자.
- SSH(보안 셸) 프라이빗 및 퍼블릭 키를 생성하기 위한. z/OS 유닉스 시스템 서비스 (USS) 액세스.
- 쓰기 가능한 S3 버킷. 자세한 내용은 Amazon S3 설명서의 [첫 번째 S3 버킷 생성](#)을 참조하세요.
- 서비스 관리형을 ID 공급자로 사용하고 Amazon S3를 AWS 스토리지 서비스로 사용하는 AWS Transfer Family SSH File Transfer Protocol(SFTP) 지원 서버입니다. 자세한 내용은 AWS Transfer Family 설명서의 [SFTP 지원 서버 생성](#)을 참조하세요.

제한 사항

- 이 접근 방식은 거의 실시간 또는 실시간 데이터 동기화에는 적합하지 않습니다.
- 데이터는 Db2 z/OS에서 Amazon S3로만 이동할 수 있으며 그 반대로는 이동할 수 없습니다.

아키텍처

소스 기술 스택

- z/OS에서 Db2를 실행하는 메인프레임

대상 기술 스택

- AWS Transfer Family
- Amazon S3
- Amazon Athena
- 아마존 QuickSight
- Glue
- Amazon Relational Database Service(Amazon RDS)

- Amazon Aurora
- Amazon Redshift

소스 및 대상 아키텍처

다음 다이어그램은 ASCII CSV 형식의 Db2 z/OS 데이터를 생성, 추출 및 S3 버킷으로 전송하는 프로세스를 보여줍니다.

1. Db2 카탈로그에서 데이터 마이그레이션을 위해 테이블 목록이 선택됩니다.
2. 이 목록은 외부 형식의 숫자 및 데이터 열을 사용하여 언로드 작업을 생성하는 데 사용됩니다.
3. 그런 다음 AWS Transfer Family를 사용하여 Amazon S3로 데이터를 전송합니다.
4. AWS Glue 추출, 변환, 적재(ETL) 작업이 데이터를 변환하여 지정된 형식으로 처리된 버킷에 로드하거나, AWS Glue가 데이터를 데이터베이스에 직접 공급할 수 있습니다.
5. Amazon Athena와 Amazon을 사용하여 데이터를 쿼리하고 렌더링하여 분석을 추진할 QuickSight 수 있습니다.

다음 다이어그램에는 전체 프로세스의 논리적 흐름이 나와 있습니다.

1. TABNAME이라는 첫 번째 JCL은 Db2 유틸리티 DSNTIAUL을 사용하여 Db2에서 언로드하려는 테이블 목록을 추출하고 생성합니다. 테이블을 선택하려면 SQL 입력을 수동으로 조정하여 하나 이상의 Db2 스키마를 포함하도록 필터 기준을 선택하고 추가해야 합니다.
2. REXXEXEC라고 하는 두 번째 JCL은 제공된 JCL 스켈레톤과 REXX 프로그램을 사용하여 JCL TABNAME으로 생성된 테이블 목록을 처리하고 테이블 이름당 하나의 JCL을 생성합니다. 각 JCL에는 테이블을 언로드하는 한 단계와 SFTP 프로토콜을 사용하여 파일을 S3 버킷으로 보내는 다른 단계가 포함됩니다.
3. 마지막 단계는 JCL을 실행하여 테이블을 언로드하고 파일을 AWS로 전송하는 것입니다. 온프레미스 또는 AWS의 스케줄러를 사용하여 전체 프로세스를 자동화할 수 있습니다.

도구

서비스

- [Amazon Athena](#)는 표준 SQL을 사용하여 Amazon Simple Storage Service(S3)에 있는 데이터를 직접 간편하게 분석할 수 있는 대화형 쿼리 서비스입니다.
- [Amazon Aurora](#)는 MySQL 및 PostgreSQL과 호환되는 완전 관리형 관계형 데이터베이스 엔진입니다.
- [AWS Glue](#)는 완전 관리형 추출, 전환, 적재(ETL) 서비스입니다. 이를 통해 데이터 스토어와 데이터 스트림 간에 데이터를 안정적으로 분류, 정리, 보강하고 이동할 수 있습니다.
- [QuickSightAmazon](#)은 단일 대시보드에서 데이터를 시각화, 분석 및 보고하는 데 도움이 되는 클라우드 규모의 비즈니스 인텔리전스 (BI) 서비스입니다.
- [Amazon Redshift](#)는 클라우드에서 완벽하게 관리되는 페타바이트급 데이터 웨어하우스 서비스입니다.
- [Amazon Relational Database Service\(Amazon RDS\)](#)는 AWS 클라우드에서 관계형 데이터베이스를 설정, 운영 및 조정하는 데 도움이 됩니다.
- [Amazon Simple Storage Service\(S3\)](#)는 원하는 양의 데이터를 저장, 보호 및 검색하는 데 도움이 되는 클라우드 기반 객체 스토리지 서비스입니다.
- [AWS Transfer Family](#)는 AWS 스토리지 서비스에 들어오고 나가도록 파일을 전송할 수 있는 보안 전송 서비스입니다.

메인프레임 툴

- [SSH File Transfer Protocol\(SFTP\)](#)은 서버에 원격으로 로그인하고 서버 간에 파일을 전송할 수 있는 안전한 파일 전송 프로토콜입니다. SSH는 모든 트래픽을 암호화하여 보안을 제공합니다.
- [DSNTIAUL](#)은 데이터 언로드를 위해 IBM에서 제공하는 샘플 프로그램입니다.
- [DSNUTILB](#)는 DSNTIAUL과 다른 옵션을 사용하여 데이터를 언로드하기 위해 IBM에서 제공하는 유틸리티 배치 프로그램입니다.
- [z/OS OpenSSH](#)는 IBM 운영 체제 z/OS의 Unix System Service에서 실행되는 오픈 소스 소프트웨어 (SSH)의 포트입니다. SSH는 TCP/IP 네트워크에서 실행되는 두 컴퓨터 간의 안전하고 암호화된 연결 프로그램입니다. ssh-keygen을 비롯한 여러 유틸리티를 제공합니다.
- [REXX\(Restructured Extended Executor\)](#) 스크립트는 Db2 언로드 및 SFTP 단계를 통해 JCL 생성을 자동화하는 데 사용됩니다.

코드

이 패턴의 코드는 GitHub [unloaddb2](#) 리포지토리에서 사용할 수 있습니다.

모범 사례

처음 언로드할 때는 생성된 JCL이 전체 테이블 데이터를 언로드해야 합니다.

첫 번째 전체 언로드 후에는 증분 언로드를 수행하여 성능을 향상시키고 비용을 절감합니다. 언로드 프로세스에 대한 변경 사항을 수용하도록 템플릿 JCL 데크의 SQL 쿼리를 업데이트합니다.

스키마를 수동으로 변환하거나 Lambda에서 Db2 SYSPUNCH를 입력으로 사용하는 스크립트를 사용하여 스키마를 변환할 수 있습니다. 산업 프로세스의 경우 [AWS Schema Conversion Tool\(SCT\)](#)이 선호되는 옵션입니다.

마지막으로, 메인프레임 기반 스케줄러 또는 AWS의 스케줄러를 메인프레임의 에이전트와 함께 사용하면 전체 프로세스를 관리하고 자동화할 수 있습니다.

에픽

S3 버킷 선택

작업	설명	필요한 기술
S3 버킷을 생성합니다.	지침은 첫 번째 S3 버킷 생성 을 참조하세요.	일반 AWS

Transfer Family 서버 설정

작업	설명	필요한 기술
SFTP 지원 서버를 생성합니다.	<p>AWS Transfer Family 콘솔에서 SFTP 서버를 열고 생성하려면 다음과 같이 합니다.</p> <ol style="list-style-type: none"> 1. 프로토콜 선택 페이지에서 SFTP(SSH File Transfer Protocol)-보안 셸을 통한 파일 전송 확인란을 선택합니다. 2. ID 공급자의 경우 서비스 관리자를 선택합니다. 	일반 AWS

작업	설명	필요한 기술
	3. 엔드포인트의 경우 공개 액세스를 선택합니다. 4. 도메인의 경우 Amazon S3를 선택합니다. 5. 추가 세부내용 구성 페이지에서 기본값 설정을 유지합니다. 6. 서버 생성.	
Transfer Family에 대한 IAM 역할을 생성합니다.	Transfer Family가 Amazon S3에 액세스하도록 허용하는 AWS Identity and Access Management(IAM) 역할을 생성하려면 IAM 역할 및 정책 생성 의 지침을 따릅니다.	AWS 관리자
Amazon S3 서비스 관리 사용자를 추가합니다.	Amazon S3 서비스 관리 사용자를 추가하려면 AWS 설명서 의 지침을 따르고 메인프레임 사용자 ID를 사용합니다.	일반 AWS

통신 프로토콜 보안

작업	설명	필요한 기술
SSH 키를 생성합니다.	메인프레임 USS 환경에서 다음 명령을 실행합니다. <div style="border: 1px solid #ccc; border-radius: 10px; padding: 10px; width: fit-content; margin: 10px auto;"> <pre>ssh-keygen -t rsa</pre> </div> 참고: 패스프레이즈를 입력하라는 메시지가 표시되면 비워둡니다.	메인프레임 개발자

작업	설명	필요한 기술
<p>SSH 폴더와 키 파일에 적절한 권한 수준을 부여합니다.</p>	<p>기본적으로 퍼블릭 키와 프라이빗 키는 사용자 디렉터리 <code>/u/home/username/.ssh</code> 에 저장됩니다.</p> <p>키 파일에는 권한 644를, 폴더에는 700을 부여해야 합니다.</p> <pre>chmod 644 .ssh/id_rsa chmod 700 .ssh</pre>	<p>메인프레임 개발자</p>
<p>퍼블릭 키 콘텐츠를 Amazon S3 서비스 관리 사용자에게 복사합니다.</p>	<p>미국에서 생성한 퍼블릭 키 콘텐츠를 복사하려면 AWS Transfer Family 콘솔을 엽니다.</p> <ol style="list-style-type: none"> 1. 탐색 창에서 서버를 선택합니다. 2. 서버 ID 옆에서 식별자를 선택하면 서버 세부 정보를 볼 수 있습니다. 3. 사용자에서 사용자 이름을 선택하면 사용자 세부 정보가 표시됩니다. 4. SSH 퍼블릭 키에서 SSH 퍼블릭 키 추가를 선택해 새 SSH 퍼블릭 키를 사용자에게 추가합니다. SSH 퍼블릭 키의 경우 퍼블릭 키를 입력합니다. 새 사용자를 추가하기 전에 서비스에서 키의 유효성을 검사합니다. 5. 키 추가를 선택합니다. 	<p>메인프레임 개발자</p>

JCL을 생성합니다.

작업	설명	필요한 기술
범위 내 Db2 테이블 목록을 생성합니다.	<p>입력 SQL을 제공하여 데이터 마이그레이션 범위가 지정된 테이블 목록을 생성합니다. 이 단계에서는 SQL where 절을 사용하여 Db2 카탈로그 테이블 SYSIBM.SYSTABLES를 쿼리하는 선택 기준을 지정해야 합니다. 특정 접두사로 시작하거나 증분 언로드의 타임스탬프를 기반으로 하는 특정 스키마 또는 테이블 이름을 포함하도록 필터를 사용자 정의할 수 있습니다. 출력은 메인프레임의 물리적 순차(PS) 데이터 세트에서 캡처됩니다. 이 데이터 세트는 JCL 생성의 다음 단계를 위한 입력 역할을 합니다.</p> <p>JCL TABNAME(필요한 경우 이름을 바꿀 수 있음)을 사용하기 전에 다음을 변경합니다.</p> <ol style="list-style-type: none"> 1. <Jobcard>를 Db2 유틸리티를 실행할 권한이 있는 작업 클래스와 사용자로 대체합니다. 2. <HLQ1>을 사이트 표준에 맞게 출력 데이터 세트 이름으로 대체하거나 사용자 정의합니다. 3. 사이트 표준에 따라 PDSE(분할된 데이터 세트 확장)의 STEPLIB 스택을 업 	메인프레임 개발자

작업	설명	필요한 기술
	<p>데이트합니다. 이 패턴의 예제에서는 IBM 기본값을 사용합니다.</p> <ol style="list-style-type: none"> 4. PLAN 이름과 LIB를 설치별 값으로 대체합니다. 5. <Schema>와 <Prefix>를 Db2 카탈로그에 대한 선택 기준에 따라 대체합니다. 6. 그에 따른 JCL을 PDS(분할된 데이터 세트) 라이브러리에 저장합니다. 7. JCL을 제출합니다. <p>Db2 테이블 목록 추출 작업</p> <pre data-bbox="592 955 1031 1837"> <Jobcard> /* /* UNLOAD ALL THE TABLE NAMES FOR A PARTICULAR SCHEMA /* //STEP01 EXEC PGM=IEFBR 14 /* //DD1 DD DISP=(MOD ,DELETE,DELETE), // UNIT=SYSDA, // SPACE=(1000, (1,1)), // DSN=<HLQ1 >.DSN81210.TABLIST /* //DD2 DD DISP=(MOD ,DELETE,DELETE), // UNIT=SYSDA, // SPACE=(1000, (1,1)), </pre>	

작업	설명	필요한 기술
	<pre>// DSN=<HLQ1 >.DSN81210.SYSPUNCH //* //UNLOAD EXEC PGM=IKJEF T01,DYNAMNBR=20 //SYSTSPRT DD SYSOUT=* //STEPLIB DD DISP=SHR,DSN=DSNC1 0.DBCG.SDSNEXIT // DD DISP=SHR, DSN=DSNC10.SDSNLOAD // DD DISP=SHR, DSN=CEE.SCEERUN // DD DISP=SHR, DSN=DSNC10.DBCG.RU NLIB.LOAD //SYSTSIN DD * DSN SYSTEM(DBCG) RUN PROGRAM(D SNTIAUL) PLAN(DSNT IB12) PARS('SQL') - LIB('DSNC 10.DBCG.RUNLIB.LOAD') END //SYSPRINT DD SYSOUT=* //* //SYSUDUMP DD SYSOUT=* //* //SYSRECO0 DD DISP=(NEW ,CATLG,DELETE), // UNIT=SYSD A,SPACE=(32760,(10 00,500)), // DSN=<HLQ1 >.DSN81210.TABLST //* //SYSPUNCH DD DISP=(NEW ,CATLG,DELETE), // UNIT=SYSD A,SPACE=(32760,(10 00,500)),</pre>	

작업	설명	필요한 기술
	<pre>// VOL=SER=S CR03,RECFM=FB,LREC L=120,BLKSIZE=12 // DSN=<HLQ1 >.DSN81210.SYSPUNCH //* //SYSIN DD * SELECT CHAR(CREA TOR), CHAR(NAME) FROM SYSIBM.SY STABLES WHERE OWNER = '<Schema>' AND NAME LIKE '<Prefix>%' AND TYPE = 'T'; /*</pre>	

작업	설명	필요한 기술
<p>JCL 템플릿을 수정합니다.</p>	<p>이 패턴과 함께 제공되는 JCL 템플릿에는 일반 작업 카드와 라이브러리 이름이 포함되어 있습니다. 하지만 대부분의 메인프레임 사이트에는 데이터 세트 이름, 라이브러리 이름, 작업 카드에 대한 자체 명명 표준이 있습니다. 예를 들어 Db2 작업을 실행하려면 특정 작업 클래스가 필요할 수 있습니다. Job Entry Subsystem 구현 JES2 및 JES3는 추가 변경 사항이 필요할 수 있습니다. 표준 로드 라이브러리에는 IBM의 기본값인 SYS1이 아닌 다른 첫 번째 한정자가 있을 수 있습니다. 따라서 템플릿을 실행하기 전에 사이트별 표준에 맞게 템플릿을 사용자 정의합니다.</p> <p>스켈레톤 JCL UNLDSKEL을 다음과 같이 변경합니다.</p> <ol style="list-style-type: none"> 1. Db2 유틸리티를 실행할 권한이 있는 작업 클래스 및 사용자로 작업 카드를 수정합니다. 2. 사이트 표준에 맞게 출력 데이터 세트 이름을 사용자 지정합니다. 3. 사이트 표준에 따라 PDSE의 STEPLIB 스택을 업데이트합니다. 이 패턴의 예제에서는 IBM 기본값을 사용합니다. 	<p>메인프레임 개발자</p>

작업	설명	필요한 기술
	<p>4. Db2 서브시스템 이름 및 상호연관 ID로 <DSN>을 대체합니다.</p> <p>5. 그에 따른 JCL을 ISPF의 표준 스켈레톤 템플릿 라이브러리인 ISPSLIB 스택의 일부인 PDS 라이브러리에 저장합니다.</p> <p>언로드 및 SFTP JCL 스켈레톤</p> <pre data-bbox="592 741 1029 1864"> //&USRPFX.U JOB (DB2UNLOAD), 'JOB', CLASS=A,MSGCLASS=A, // TIME=1440 ,NOTIFY=&USRPFX //* DELETE DATASETS //STEP01 EXEC PGM=IEFBR14 //DD01 DD DISP=(MOD ,DELETE,DELETE), // UNIT=SYSD A, // SPACE=(TR K,(1,1)), // DSN=&USRPFX..DB2.P UNCH.&JOBNAME //DD02 DD DISP=(MOD ,DELETE,DELETE), // UNIT=SYSD A, // SPACE=(TR K,(1,1)), // DSN=&USRPFX..DB2.U NLOAD.&JOBNAME //* //* RUNNING DB2 EXTRACTION BATCH JOB FOR AWS DEMO </pre>	

작업	설명	필요한 기술
	<pre> /** //UNLD01 EXEC PGM=DSNUTILB,REGIO N=0M, // PARM= '<DSN>,UNLOAD' //STEPLIB DD DISP=SHR,DSN=DSNC1 0.DBCG.SDSNEXIT // DD DISP=SHR, DSN=DSNC10.SDSNLOAD //SYSPRINT DD SYSOUT=* //UTPRINT DD SYSOUT=* //SYSOUT DD SYSOUT=* //SYSPUN01 DD DISP=(NEW,CATLG,DE LETE), // SPACE=(CY L,(1,1),RLSE), // DSN=&USRPF..DB2.P UNCH.&JOBNAME //SYSREC01 DD DISP=(NEW,CATLG,DE LETE), // SPACE=(CY L,(10,50),RLSE), // DSN=&USRPF..DB2.U NLOAD.&JOBNAME //SYSPRINT DD SYSOUT=* //SYSIN DD * UNLOAD DELIMITED COLDEL ',' FROM TABLE &TABNAME UNLDDN SYSREC01 PUNCHDDN SYSPUN01 SHRLEVEL CHANGE ISOLATION UR; /** /** /** FTP TO AMAZON S3 BACKED FTP SERVER IF UNLOAD WAS SUCCESSFUL /** </pre>	

작업	설명	필요한 기술
	<pre> //SFTP EXEC PGM=BPXBA TCH,COND=(4,LE),RE GION=0M //STDPARM DD * SH cp "'/'&USRP FX..DB2.UNLOAD.&JO BNAME '" &TABNAME..csv; echo "ascii " >> uplcmd; echo "PUT &TABNAME. .csv " >>>> uplcmd; sftp -b uplcmd -i .ssh/ id_rsa &FTPUSER. @&FTPSITE; im &TABNAME..csv; //SYSPRINT DD SYSOUT=* //STDOUT DD SYSOUT=* //STDENV DD * //STDERR DD SYSOUT=* </pre>	

작업	설명	필요한 기술
<p>매스 언로드 JCL을 생성합니다.</p>	<p>이 단계에는 JCL을 사용하여 ISPF 환경에서 REXX 스크립트를 실행하는 작업이 포함됩니다. 첫 번째 단계에서 만든 범위 내 테이블 목록을 TABLIST DD 이름에 대한 대량 JCL 생성을 위한 입력으로 제공합니다. JCL은 ISPF DD 이름에 대해 지정된 사용자 지정 파티션을 나눈 데이터 세트에서 테이블 이름당 하나의 새 JCL을 생성합니다. 이 라이브러리를 미리 할당합니다. 각각의 새 JCL에는 두 단계가 있습니다. 하나는 Db2 테이블을 파일로 언로드하는 것이고 다른 하나는 파일을 S3 버킷으로 보내는 단계입니다.</p> <p>JCL REXXEXEC에서 다음과 같이 변경합니다(이름을 변경할 수 있음).</p> <ol style="list-style-type: none"> 테이블에 대한 언로드 권한이 있는 메인프레임 사용자 ID로 Job card user ID를 대체합니다. SYSPROC, ISPPLIB, ISPSLIB, ISPM LIB 및 ISPTLIB와 <HLQ1> 값을 대체하거나 사이트 표준에 맞게 DSN을 사용자 지정합니다. 설치별 값을 찾으려면 TSO ISRDDN 명령을 사용합니다. 	<p>메인프레임 개발자</p>

작업	설명	필요한 기술
	<p>2. 설치 시 작업 실행 권한이 있는 사용자 ID로 <MFUSER>를 대체합니다.</p> <p>3. 설치 시 USS 및 FTP 권한이 있는 사용자 ID로 <FTPUSER> 를 대체합니다. 이 사용자 ID와 해당 SSH 보안 키는 메인프레임의 해당 Unix Systems Services 디렉터리에 있는 것으로 가정합니다.</p> <p>4. AWS Transfer Family IP 주소 또는 도메인 이름으로 <AWS TransferFamily IP>를 대체합니다. 이 주소는 SFTP 단계에 사용됩니다.</p> <p>5. 아래 설명과 같이 사이트 표준 편의를 적용하고 REXX 프로그램을 업데이트한 후 JCL을 제출합니다.</p> <p>대량 JCL 생성 작업</p> <pre data-bbox="592 1375 1031 1869"> //RUNREXX JOB (CREATEJCL), 'RUNS ISPF TABLIST', CLASS=A,MSGCLASS=A, // TIME=1440 ,NOTIFY=&SYSUID /* Most of the values required can be updated to your site specific /* values using the command 'TSO ISRDDN' in your ISPF session. </pre>	

작업	설명	필요한 기술
	<pre> /** Update all the lines tagged with //update marker to desired /** site specific values. //ISPF EXEC PGM=IKJEF T01,REGION=2048K,D YNAMNBR=25 //SYSPROC DD DISP=SHR,DSN=USER. Z23D.CLIST //SYSEXEC DD DISP=SHR,DSN=<HLQ1 >.TEST.REXXLIB //ISPPLIB DD DISP=SHR,DSN=ISP.S ISPPENU //ISPSLIB DD DISP=SHR,DSN=ISP.S ISPSENU // DD DISP=SHR,DSN=<HLQ1 >.TEST.ISPSLIB //ISPMLIB DD DSN=ISP.SISPMENU,D ISP=SHR //ISPTLIB DD DDNAME=ISPTABL // DD DSN=ISP.S ISPTENU,DISP=SHR //ISPTABL DD LIKE=ISP.SISPTENU, UNIT=VIO //ISPPROF DD LIKE=ISP.SISPTENU, UNIT=VIO //ISPLLOG DD SYSOUT=*,RECFM=VA, LRECL=125 //SYSPRINT DD SYSOUT=* //SYSTSPRT DD SYSOUT=* </pre>	

작업	설명	필요한 기술
	<pre data-bbox="609 210 1015 1333"> //SYSUDUMP DD SYSOUT=* //SYSDBOUT DD SYSOUT=* //SYSTSPRT DD SYSOUT=* //SYSUDUMP DD SYSOUT=* //SYSDBOUT DD SYSOUT=* //SYSHELP DD DSN=SYS1.HELP,DISP =SHR //SYSOUT DD SYSOUT=* /* Input list of tablenames //TABLIST DD DISP=SHR,DSN=<HLQ1 >.DSN81210.TABLIST /* Output pds //ISPFIL DD DISP=SHR,DSN=<HLQ1 >.TEST.JOBGEN //SYSTSIN DD * ISPSTART CMD(ZSTEPS <MFUSER> <FTPUSER> <AWS TransferFamily IP>) /* </pre> <p data-bbox="592 1375 1031 1459">REXX 스크립트를 사용하기 전에 다음과 같이 변경합니다.</p> <ol data-bbox="592 1501 1031 1816" style="list-style-type: none"> 1. ZSTEPS를 멤버 이름으로 사용하여 이전 단계에서 편집한 JCL REXXEXEC의 SYSEXEC 스택 아래에 정의된 PDS 라이브러리에 REXX 스크립트를 저장합니다. 이름을 바꾸려면 필요에 	

작업	설명	필요한 기술
	<p>맞게 JCL을 업데이트해야 합니다.</p> <ol style="list-style-type: none"> 이 스크립트는 오류가 있는 경우에 대비하여 추적 옵션을 사용하여 추가 정보를 인쇄합니다. 대신 EXECIO, ISPEXEC, TSO 문 뒤에 오류 처리 코드를 추가하고 추적 행을 제거할 수 있습니다. 이 스크립트는 최대 100,000 명의 멤버를 지원할 수 있는 LODnnnnn 명명 규칙을 사용하여 멤버 이름을 생성합니다. 테이블이 100,000개 이상인 경우 더 짧은 접두사를 사용하고 tempjob 명령문의 숫자를 조정합니다. <p>ZSTEPS REXX 스크립트</p> <pre data-bbox="592 1176 1031 1791"> /*REXX - - - - - - - - - - - - - - - */ /* 10/27/2021 - added new parms to accommoda te ftp */ Trace "o" parse arg usrpfx ftpuser ftpsite Say "Start" Say "Ftpuser: " ftpuser "Ftpsite:" ftpsite Say "Reading table name list" </pre>	

작업	설명	필요한 기술
	<pre> "EXECIO * DISKR TABLIST (STEM LINE. FINIS" DO I = 1 TO LINE.0 Say I suffix = I Say LINE.i Parse var LINE.i schema table rest tabname = schema !! "." !! table Say tabname tempjob= "LOD" !! RIGHT("0000" !! i, 5) jobname=tempjob Say tempjob ADDRESS ISPEXEC "FTOPEN " ADDRESS ISPEXEC "FTINCL UNLDSKEL" /* member will be saved in ISPDSN library allocated in JCL */ ADDRESS ISPEXEC "FTCLOSE NAME("tem pjob")" END ADDRESS TSO "FREE F(TABLIST) " ADDRESS TSO "FREE F(ISPFILE) " exit 0 </pre>	

JCL을 실행합니다.

작업	설명	필요한 기술
<p>Db2 언로드 단계를 수행합니다.</p>	<p>JCL을 생성한 후에는 언로드해야 하는 테이블 수만큼의 JCL을 갖게 됩니다.</p> <p>이 스토리에서는 JCL에서 생성한 예제를 사용하여 구조와 가장 중요한 단계를 설명합니다.</p> <p>사용자는 아무 작업도 수행할 필요가 없습니다. 다음 정보는 참조용입니다. 이전 단계에서 생성한 JCL을 제출하려면 Lodnnnnn JCLS 제출 작업으로 건너뛸니다.</p> <p>IBM에서 제공한 DSNUTILB Db2 유틸리티와 함께 JCL을 사용하여 Db2 데이터를 언로드할 때는 언로드된 데이터에 압축된 수치 데이터가 포함되어 있지 않은지 확인해야 합니다. 이 작업을 수행하려면 DSNUTILB DELIMITED 파라미터를 사용합니다.</p> <p>이 DELIMITED 파라미터는 텍스트 필드에 문자를 구분자와 큰따옴표로 추가하고, VARCHAR 열의 패딩을 제거하고, DATE 필드를 포함한 모든 숫자 필드를 EXTERNAL FORMAT으로 변환하여 CSV 형식으로 데이터를 언로드할 수 있도록 지원합니다.</p>	<p>메인프레임 개발자, 시스템 엔지니어</p>

작업	설명	필요한 기술
	<p>다음 예제는 쉼표를 구분 기호로 사용하여 생성된 JCL의 언로드 단계를 보여줍니다.</p> <pre data-bbox="592 378 1031 814">UNLOAD DELIMITED COLDEL ',' FROM TABLE SCHEMA_NAME.TBNAME UNLDDN SYSREC01 PUNCHDDN SYSPUN01 SHRLEVEL CHANGE ISOLATION UR;</pre>	

작업	설명	필요한 기술
SFTP 단계를 수행합니다.	<p>JCL에서 SFTP 프로토콜을 사용하려면 BPXBATCH 유틸리티를 사용합니다.</p> <p>SFTP 유틸리티는 MVS 데이터 세트에 직접 액세스할 수 없습니다. 복사 명령(cp)을 사용하여 순차 파일을 &USRPFX..DB2.UNLOAD.&JOBNAME 을 USS 디렉터리에 복사할 수 있으며, 이 디렉터리에서 &TABNAME..csv 이 됩니다.</p> <p>프라이빗 키(id_rsa)를 사용하고 RACF 사용자 ID를 사용자 이름으로 사용해서 sftp 명령을 실행하여 AWS Transfer Family IP 주소에 연결합니다.</p> <pre data-bbox="597 1081 1026 1591"> SH cp "'/'&USRP FX..DB2.UNLOAD.&JO BNAME'" &TABNAME..csv; echo "ascii " >> uplcmd; echo "PUT &TABNAME. .csv " >>>> uplcmd; sftp -b uplcmd -i .ssh/ id_rsa &FTPUSER. @&FTP_TF_SITE; rm &TABNAME..csv; </pre>	메인프레임 개발자, 시스템 엔지니어

작업	설명	필요한 기술
LODnnnnn JCL을 제출합니다.	이전 JCL에서는 언로드하여 CSV로 변환하고 S3 버킷으로 전송해야 하는 모든 LODnnnnn JCL 테이블을 생성했습니다. 생성된 모든 JCL에서 submit 명령을 실행합니다.	메인프레임 개발자, 시스템 엔지니어

관련 리소스

이 문서에 사용된 다양한 도구 및 솔루션에 대한 자세한 내용은 다음을 참조하세요.

- [z/OS OpenSSH 사용 설명서](#)
- [Db2 z/OS-샘플 언로드 제어 명령문](#)
- [Db2 z/OS-구분된 파일 언로드](#)
- [Transfer Family-SFTP 지원 서버 생성](#)
- [Transfer Family-서비스 관리 사용자와 협력하기](#)

추가 정보

Amazon S3에 Db2 데이터를 저장한 후에는 다양한 방법으로 새로운 통찰력을 개발할 수 있습니다. Amazon S3는 AWS 데이터 분석 서비스와 통합되므로 분산 측에서 이 데이터를 자유롭게 사용하거나 노출할 수 있습니다. 예를 들어 다음을 수행할 수 있습니다.

- [Amazon S3에 데이터 레이크를](#) 구축하고, 데이터를 이동하지 않고도 분석 및 기계 학습 도구를 사용하여 query-in-place 귀중한 통찰력을 추출하십시오.
- AWS Transfer Family와 통합된 업로드 후 처리 워크플로우를 설정하여 [Lambda 함수](#)를 시작할 수 있습니다.
- 분석, 기계 학습, 애플리케이션 개발을 위해 데이터를 쉽게 검색, 준비, 결합할 수 있게 해주는 서버리스 데이터 통합 서비스인 [AWS Glue](#)를 사용하여 Amazon S3 또는 [완전관리형 데이터베이스](#)의 데이터에 액세스하기 위한 새로운 마이크로서비스를 개발합니다.

마이그레이션 사용 사례에서는 메인프레임에서 S3로 모든 데이터를 전송할 수 있으므로 다음을 수행할 수 있습니다.

- Amazon S3 Glacier 및 S3 Glacier Deep Archive를 사용하여 물리적 인프라를 폐기하고 비용 효율적인 데이터 보관 전략을 수립합니다.
- Amazon S3와 S3 Glacier 및 Amazon Elastic File System(Amazon EFS)과 같은 다른 AWS 서비스를 사용하여 확장 가능하고 안정적이며 안전한 백업 및 복원 솔루션을 구축하여 기존 온프레미스 기능을 보강하거나 대체합니다.

더 많은 패턴

- [Precisely Connect를 사용하여 메인프레임 데이터베이스를 AWS에 복제하기](#)

관리 및 거버넌스

주제

- [Amazon Data Firehose 리소스가 AWS KMS 키로 암호화되지 않은 경우 식별 및 알림](#)
- [AWS Systems Manager를 사용하여 Windows 레지스트리 항목의 추가 또는 업데이트 자동화](#)
- [aws-nuke를 사용하여 AWS 리소스 자동 삭제](#)
- [AWS Systems Manager Maintenance Windows를 사용하여 Amazon RDS DB 인스턴스를 자동 중지 및 시작](#)
- [Terraform을 사용하여 AWS Organizations에서 소프트웨어 패키지 배포를 중앙 집중화합니다.](#)
- [AWS 계정 간 중앙 집중화를 위한 VPC 흐름 로그 구성](#)
- [CloudWatch NLog를 사용하여 Amazon Logs에서.NET 애플리케이션에 대한 로깅을 구성합니다.](#)
- [여러 AWS 계정 및 AWS 리전에 걸쳐 AWS Service Catalog 제품 복사](#)
- [Amazon CloudWatch 이상 탐지를 사용하여 사용자 지정 지표에 대한 경보 생성](#)
- [AWS 랜딩 존 설계를 문서화하십시오](#)
- [다중 지역, 다중 계정 조직에서 AWS CloudFormation 드리프트 감지 설정](#)
- [AWS CDK를 사용하여 여러 AWS 지역, 계정 및 DevOps OU에서 Amazon Guru를 활성화하여 운영 성능을 개선하십시오.](#)
- [부트스트랩 파이프라인을 사용하여 테라폼용 Account Factory \(AFT\) 구현](#)
- [여러 AWS 계정 및 AWS 리전의 AWS Service Catalog 제품을 관리](#)
- [AWS Organizations의 AWS 멤버 계정을 AWS Control Tower로 마이그레이션](#)
- [여러 AWS 계정에 공유된 Amazon Machine Image의 사용을 모니터링](#)
- [AWS Organizations의 프로그래밍 방식 계정 폐쇄에 대한 알림 설정](#)
- [패턴 더 보기](#)

Amazon Data Firehose 리소스가 AWS KMS 키로 암호화되지 않은 경우 식별 및 알림

작성자: Ram Kandaswamy(AWS)

환경: 프로덕션

기술: 관리 및 거버넌스, 분석, 빅 데이터, 클라우드 네이티브, 인프라, 보안, 자격 증명, 규정 준수

AWS 서비스: AWS, 아마존 CloudTrail CloudWatch, AWS ID 및 액세스 관리, 아마존 키 네시스, AWS Lambda, 아마존 SNS

요약

규정 준수를 위해 일부 조직에서는 Amazon Data Firehose와 같은 데이터 전송 리소스에 암호화를 활성화해야 합니다. 이 패턴은 리소스가 규정을 준수하지 않는 경우 모니터링, 탐지 및 통지하는 방법을 보여줍니다.

암호화 요구 사항을 유지하기 위해 Amazon Web Services (AWS) 에서 이 패턴을 사용하여 AWS Key Management Service (AWS KMS) 키로 암호화되지 않은 Firehose 전송 리소스를 자동으로 모니터링 하고 탐지할 수 있습니다. 솔루션은 알림 알림을 전송하며, 자동 문제 해결을 수행하도록 확장할 수 있습니다. 이 솔루션은 개별 계정 또는 다중 계정 환경(예: AWS 란딩 존 또는 AWS Control Tower)을 사용하는 환경에 적용할 수 있습니다.

사전 조건 및 제한 사항

사전 조건

- Firehose 전송 스트림
- 이 인프라 자동화에 사용되는 CloudFormation AWS에 대한 충분한 권한 및 지식

제한 사항

이 솔루션은 탐지에 AWS CloudTrail 이벤트를 사용하기 때문에 실시간이 아니며 암호화되지 않은 리소스가 생성되는 시간과 알림이 전송되는 시간 사이에 지연이 있습니다.

아키텍처

대상 기술 스택

이 솔루션은 서버리스 기술과 다음 서비스를 사용합니다.

- AWS CloudTrail
- 아마존 CloudWatch
- AWS Command Line Interface(AWS CLI)
- AWS Identity and Access Management(IAM)
- 아마존 데이터 파이어호스
- AWS Lambda
- Amazon Simple Notification Service(Amazon SNS)

대상 아키텍처

1. 사용자가 Firehose를 만들거나 수정합니다.
2. CloudTrail 이벤트가 감지되고 매칭됩니다.
3. Lambda가 간접적으로 호출됩니다.
4. 규정을 준수하지 않는 리소스가 식별됩니다.
5. 이메일 알림을 전송됩니다.

자동화 및 규모 조정

CloudFormation StackSetsAWS를 사용하면 단일 명령으로 여러 AWS 지역 또는 계정에 이 솔루션을 적용할 수 있습니다.

도구

- [AWS CloudTrail](#) — CloudTrail AWS는 AWS 계정의 거버넌스, 규정 준수, 운영 및 위험 감사를 지원하는 AWS 서비스입니다. 사용자, 역할 또는 AWS 서비스가 수행한 작업은 에서 이벤트로 기록됩니다 CloudTrail. 이벤트에는 AWS Management Console, AWS Command Line Interface, AWS SDK 및 API 작업에서 수행된 작업이 포함됩니다.

- [Amazon CloudWatch Events](#) — Amazon CloudWatch Events는 AWS 리소스의 변경 사항을 설명하는 시스템 이벤트 near-real-time 스트림을 제공합니다.
- [AWS CLI](#) – AWS Command Line Interface(AWS CLI)는 명령줄 셸에서 명령을 사용하여 AWS 서비스와 상호 작용할 수 있는 오픈 소스 도구입니다.
- [IAM](#) – AWS Identity and Access Management(IAM)는 AWS 리소스에 대한 사용자의 액세스를 안전하게 제어할 수 있게 지원하는 웹 서비스입니다. IAM을 사용하여 리소스를 사용하도록 인증(로그인) 및 권한 부여(권한 있음)된 대상을 제어합니다.
- [Amazon Data Firehose](#) — Amazon Data Firehose는 실시간 스트리밍 데이터를 전송하는 완전 관리형 서비스입니다. Firehose를 사용하면 애플리케이션을 작성하거나 리소스를 관리할 필요가 없습니다. Firehose에 데이터를 보내도록 데이터 생산자를 구성하면 Firehose가 지정한 대상으로 데이터를 자동으로 전송합니다.
- [AWS Lambda](#) – AWS Lambda는 서버를 프로비저닝하거나 관리하지 않고도 코드를 실행하도록 지원하는 컴퓨팅 서비스입니다. Lambda는 필요 시에만 코드를 실행하며, 일일 몇 개의 요청에서 초당 수천 개의 요청까지 자동으로 규모를 조정합니다. 사용한 컴퓨팅 시간만큼만 비용을 지불하고, 코드가 실행되지 않을 때는 요금이 부과되지 않습니다.
- [Amazon SNS](#) – Amazon Simple Notification Service(Amazon SNS)는 게시자에서 구독자(생산자 및 소비자라고도 함)로 메시지를 전송하는 관리형 서비스입니다.

에픽

규정 준수를 위한 암호화 적용

작업	설명	필요한 기술
AWS를 CloudFormation StackSets 배포하십시오.	AWS CLI에서 <code>firehose-encryption-checker</code> .yaml 템플릿 (첨부됨) 을 사용하여 다음 명령을 실행하여 스택 세트를 생성합니다. 파라미터에 대한 유효한 Amazon SNS 주제 Amazon 리소스 이름(ARN) 을 제공합니다. 배포 시 템플릿에 설명된 대로 필요한 권한이 있는 CloudWatch Events 규칙, Lambda 함수 및	클라우드 아키텍트, 시스템 관리자

작업	설명	필요한 기술
	<p>IAM 역할을 성공적으로 생성해야 합니다.</p> <pre>aws cloudformation create-stack-set --stack-set-name my-stack-set -- template-body file:// firehose-encryption- checker.yaml</pre>	
<p>스택 인스턴스를 생성합니다.</p>	<p>스택은 선택한 AWS 리전과 하나 이상의 계정에서 생성해야 합니다. 스택 인스턴스 생성을 위해 다음 명령을 실행하여 스택 이름, 계정 번호, 자신의 리전 이름을 바꿉니다.</p> <pre>aws cloudformation create-stack-insta nces --stack-s et-name my-stack- set --account s 123456789012 223456789012 -- regions us-east-1 us- east-2 us-west-1 us- west-2 --operati on-preferences FailureToleranceCo unt=1</pre>	<p>클라우드 아키텍트, 시스템 관리자</p>

관련 리소스

- [AWS와 함께 일하기 CloudFormation StackSets](#)

- [아마존 CloudWatch 이벤트란 무엇입니까?](#)

추가 정보

AWS Config는 Firehose 전송 스트림 리소스 유형을 지원하지 않으므로 솔루션에서 AWS Config 규칙을 사용할 수 없습니다.

첨부

이 문서와 관련된 추가 콘텐츠에 액세스하려면 [attachment.zip](#) 파일의 압축을 풉니다.

AWS Systems Manager를 사용하여 Windows 레지스트리 항목의 추가 또는 업데이트 자동화

작성자: Appasaheb Bagali(AWS)

작성자: AWS	환경: PoC 또는 파일럿	기술: 클라우드 네이티브, 인프라, 현대화, 보안, ID, 규정 준수, 관리 및 거버넌스 DevOps
워크로드: Microsoft	AWS 서비스: AWS Systems Manager	

요약

AWS Systems Manager는 Amazon Elastic Compute Cloud(Amazon EC2) 인스턴스에 대한 원격 관리 도구입니다. Systems Manager는 Amazon Web Services의 인프라에 대한 가시성과 제어 기능을 제공합니다. 이 다용도 도구를 사용하여 보안 취약성 스캔 보고서에서 취약성으로 식별된 Windows 레지스트리 변경 사항을 수정할 수 있습니다.

이 패턴은 환경 안전을 위해 권장되는 레지스트리 변경을 자동화하여 Windows 운영 체제를 실행하는 EC2 인스턴스의 보안을 유지하는 단계를 다룹니다. 이 패턴은 Run Command를 사용하여 명령 문서를 실행합니다. 코드가 연락되어 있으며 코드 일부가 코드 섹션에 포함되어 있습니다.

사전 조건 및 제한 사항

- 활성 상태의 AWS 계정
- EC2 인스턴스 및 Systems Manager에 액세스할 수 있는 권한

아키텍처

대상 기술 스택

- 두 개의 서브넷과 Network Address Translation(NAT) 게이트웨이가 있는 Virtual Private Cloud(VPC)
- 레지스트리 이름 및 값을 추가하거나 업데이트하기 위한 Systems Manager 명령 문서
- 정된 EC2 인스턴스에서 명령 문서를 실행하기 위한 Systems Manager Run Command

대상 아키텍처

도구

도구

- [IAM 정책 및 역할](#) - AWS Identity and Access Management(IAM)는 AWS 리소스에 대한 사용자의 액세스를 안전하게 제어할 수 있게 지원하는 웹 서비스입니다. IAM을 사용하여 리소스를 사용하도록 인증(로그인) 및 권한 부여(권한 있음)된 대상을 제어합니다.
- [Amazon Simple Storage Service](#) - Amazon Simple Storage Service(S3)는 인터넷 스토리지 서비스입니다. 이 서비스는 개발자가 더 쉽게 웹 규모 컴퓨팅 작업을 수행할 수 있도록 설계되었습니다. 이 패턴에서 S3 버킷은 Systems Manager 로그를 저장하는 데 사용됩니다.
- [AWS Systems Manager](#) - AWS Systems Manager는 AWS에서 인프라를 보고 제어하기 위해 사용할 수 있는 AWS 서비스입니다. Systems Manager는 관리형 인스턴스를 검사하고 탐지된 정책 위반을 보고(또는 시정 조치)함으로써 보안 및 규정 준수를 유지하는 데 도움이 됩니다.
- [AWS Systems Manager 명령 문서](#) — AWS Systems Manager 명령 문서는 Run Command에 사용됩니다. 대부분의 명령 문서는 Systems Manager에서 지원하는 모든 Linux 및 Windows 서버 운영 체제에서 지원됩니다.
- [AWS Systems Manager Run Command](#) — AWS Systems Manager Run Command를 사용하면 관리형 인스턴스의 구성을 원격으로 안전하게 관리할 수 있습니다. Run Command를 사용하면 일반적인 관리 작업을 자동화하고 일회성 구성 변경을 대규모로 수행할 수 있습니다.

코드

다음 예제 코드를 사용하여 Microsoft Windows 레지스트리 이름을 Version에, 레지스트리 경로를 HKCU:\Software\ScriptingGuys\Scripts에, 그리고 값을 2에 추가하거나 업데이트할 수 있습니다.

```
#Windows registry path which needs to add/update
$registryPath = 'HKCU:\\Software\\ScriptingGuys\\Scripts'
#Windows registry Name which needs to add/update
$Name = 'Version'
#Windows registry value which needs to add/update
$value = 2
# Test-Path cmdlet to see if the registry key exists.
IF(!(Test-Path $registryPath))
```

```

    {
        New-Item -Path $registryPath -Force | Out-Null
        New-ItemProperty -Path $registryPath -Name $name -Value $value ` -
PropertyType DWORD -
        Force | Out-
        Null
    } ELSE {
        New-ItemProperty -Path $registryPath -Name $name -Value $value `
-PropertyType
        DWORD
        -Force | Out-Null
    }
}
echo 'Registry Path:$registryPath
echo 'Registry Name:$registryPath
echo 'Registry Value:(Get-ItemProperty -Path $registryPath -Name $Name).version

```

전체 Systems Manager 명령 문서 JavaScript 객체 표기법 (JSON) 코드 예제가 첨부되어 있습니다.

에픽

VPC를 설정

작업	설명	필요한 기술
VPC를 생성합니다.	AWS Management Console에서 퍼블릭 및 프라이빗 서브넷과 NAT 게이트웨이가 있는 VPC를 생성합니다. 자세한 내용은 AWS 설명서 를 참조하십시오.	클라우드 관리자
보안 그룹을 생성합니다.	각 보안 그룹이 소스 IP 주소에서 원격 데스크톱 프로토콜 (RDP)에 액세스할 수 있도록 허용하는지 확인하십시오.	클라우드 관리자

IAM 정책 및 IAM 역할 생성

작업	설명	필요한 기술
IAM 정책을 생성합니다.	Amazon S3, Amazon EC2 및 Systems Manager에 대한 액세스	클라우드 관리자

작업	설명	필요한 기술
	스 권한을 부여하는 IAM 정책을 생성합니다.	
IAM 역할을 생성합니다.	IAM 역할을 생성하고 Amazon S3, Amazon EC2 및 Systems Manager에 대한 액세스 권한을 부여하는 IAM 정책을 연결하십시오.	클라우드 관리자

자동화 실행

작업	설명	필요한 기술
Systems Manager 명령 문서를 생성하십시오.	추가하거나 업데이트할 Microsoft Windows 레지스트리 변경 내용을 배포할 Systems Manager 명령 문서를 만드십시오.	클라우드 관리자
Systems Manager Run Command를 실행합니다.	명령 문서와 Systems Manager 대상 인스턴스를 선택하여 Systems Manager Run Command를 실행합니다. 이렇게 하면 선택한 명령 문서의 Microsoft Windows 레지스트리 변경 내용이 대상 인스턴스로 푸시됩니다.	클라우드 관리자

관련 리소스

- [AWS Systems Manager](#)
- [AWS Systems Manager 문서](#)
- [AWS Systems Manager Run Command](#)

첨부

이 문서와 관련된 추가 콘텐츠에 액세스하려면 [attachment.zip](#) 파일의 압축을 풉니다.

aws-nuke를 사용하여 AWS 리소스 자동 삭제

작성자: 스리니바스 가네산(AWS)

코드 aws-nuke-account-c
leanser 리포지토리: [-예제](#)

환경: PoC 또는 파일럿

기술: 관리 및 거버넌스, 클라우드 네이티브, 비용 관리, 서버리스 DevOps, 소프트웨어 개발 및 테스트

AWS 서비스: AWS, AWS CloudFormation CodeBuild, 아마존 SNS, AWS Step Functions, 아마존 EventBridge

요약

경고: aws-nuke는 대상 Amazon Web Services(AWS) 계정과 AWS 리전의 거의 모든 리소스를 삭제하는 오픈 소스 도구입니다. 도구를 사용하여 리소스를 삭제하기 전에 대상 환경에 미칠 영향을 완전히 이해해야 합니다. 이 솔루션은 프로덕션 환경에 사용하기 위한 것이 아닙니다. 이 솔루션은 샌드박스 또는 개발 환경에서만 구현하는 것이 좋습니다. 모의 실습을 수행하여 솔루션이 여전히 필요한 리소스를 삭제하지 않는지 확인합니다. 자세한 내용은 AWS-Nuke README () 의 [주의](#) 섹션을 참조하십시오. GitHub

샌드박스 또는 개발 AWS 계정에 미사용 리소스를 누적하는 것은 매우 흔한 일입니다. 개발자는 일반적인 개발 주기의 일환으로 다양한 서비스와 리소스를 생성하여 실험한 다음 더 이상 필요하지 않아도 해당 리소스를 삭제하지 않습니다. 사용하지 않는 리소스는 조직에 불필요하며 때로는 높은 비용을 초래할 수 있습니다. 이러한 리소스를 삭제하면 이러한 환경을 운영하는 데 드는 비용을 절감할 수 있습니다.

이 패턴은 aws-Nuke, AWS Step Functions, Amazon 및 AWS를 사용하여 개발 또는 샌드박스 계정에서 오래된 리소스를 주기적으로 삭제하는 자동화된 솔루션을 제공합니다. EventBridge CodeBuild 대상 리전에서는 계정을 기본적으로 “1일차” 상태로 복원합니다. 이 상태에는 기본 리소스 및 AWS에서 관리하는 리소스만 포함됩니다. 먼저 이 솔루션을 모의 실습(기본) 모드에서 실행하고 식별된 모든 리소스를 삭제할지 확인합니다. 그런 다음 모의 실습 모드를 끄고 이 솔루션을 실행하여 해당 리소스를 삭제합니다.

EventBridge 규칙을 사용하여 이 자동화된 솔루션이 일정에 따라 실행되도록 구성할 수 있습니다. 이 EventBridge 규칙은 Step Functions 워크플로를 시작합니다. 지역 간 확장성을 위해 워크플로는 각 지역에서 별도의 CodeBuild 프로젝트를 호출합니다. CodeBuild 프로젝트는 병렬로 실행되며 AWS-nuke를 사용하여 해당 지역의 리소스를 삭제합니다. 이 솔루션은 비용을 절감하고, 확장성을 제공하고, 리소스 관리에 필요한 시간을 줄이고, 모니터링 효율성을 개선하도록 설계되었습니다. 이 솔루션을 배포하는 데 도움이 되도록 이 패턴의 코드 리포지토리에 포함된 AWS CloudFormation 템플릿을 사용하여 필요한 모든 리소스를 스택으로 생성하고 구성합니다.

이 솔루션은 다음 기능을 제공합니다.

- EventBridge에서는 이 자동화된 솔루션을 실행하기 위한 일정을 사용자 지정할 수 있습니다. 일반적으로 이 솔루션은 하루 중 대부분의 활동이 완료되는 사용량이 적은 시간에 실행하는 것이 가장 좋습니다.
- Step Functions 워크플로우를 통한 오케스트레이션은 계정 내 모든 리전에 걸쳐 확장성을 제공하고 리소스를 삭제하는 전체 시간을 줄여줍니다.
- Step Functions 워크플로우는 각 리전에서 성공 완료를 기다립니다. CodeBuild 프로젝트에 오류가 있거나 구성된 기간 내에 완료되지 않으면 워크플로우는 해당 프로젝트를 재시도합니다. 이렇게 하면 수동 개입 없이 일정에 따라 리소스를 삭제할 수 있습니다.
- 각 지역에 별도의 CodeBuild 프로젝트를 구성하면 각 지역에서 aws-nuke를 병렬로 또는 동기적으로 실행할 수 있습니다.
- aws-nuke 구성 파일의 속성 (예: regions 속성)은 프로젝트 내의 사용자 지정 Python 필터링 클래스를 사용하여 동적으로 업데이트됩니다. CodeBuild 이를 통해 사용자가 제공하는 오버라이드 파라미터를 기반으로 리소스 필터 및 리전 제약을 유연하게 처리할 수 있습니다.
- 이 패턴의 액세스 및 권한 부여 접근 방식은 자동으로 새로 고쳐지며 AWS-nuke 바이너리가 프로젝트 내에서 역할을 맡아 실행을 완료하는 데 최대 8시간이 걸립니다. CodeBuild 8시간 후에 세션 시간이 초과됩니다. 이는 AWS Identity and Access Management(IAM) 역할 체인의 표준 세션 제한인 1시간보다 깁니다. 리전에서 삭제해야 할 리소스가 많은 경우 이 추가 시간을 통해 프로세스가 완료되기 전에 시간 초과가 발생하는 것을 방지할 수 있습니다.
- 워크플로우가 완료되면 구독한 활성 이메일 주소를 사용하여 Amazon Simple Notification Service(Amazon SNS) 주제에 자세한 보고서를 전송합니다. AWS 리전마다 별도의 보고서를 받습니다. 보고서에는 삭제된 리소스 목록과 프로젝트 완료 상태가 포함됩니다. CodeBuild 이 보고서를 사용하면 aws-nuke에서 생성된 복잡한 로그를 탐색하고 파싱할 필요가 없습니다. 또한 Step Functions 상태 시스템 워크플로우가 끝나면 각 리전의 완료 상태가 포함된 요약된 이메일 보고서를 받게 됩니다.

사전 조건 및 제한 사항

사전 조건

- 모든 리소스를 삭제하려는 활성 샌드박스 또는 개발 AWS 계정입니다.

중요: 프로덕션 계정에는 이 솔루션을 배포하지 마십시오. 이 솔루션을 사용하여 리소스를 삭제하기 전에 모의 실행 옵션을 활성화하여 결과를 확인하는 것이 좋습니다.

- AWS 계정에서 다음 작업을 수행할 수 있는 권한:
 - CloudFormation 템플릿에 정의된 CloudFormation 스택과 리소스를 생성합니다.
 - IAM 역할을 생성하고 업데이트합니다.
- AWS Command Line Interface(AWS CLI), 설치 및 구성됨. 지침은 AWS CLI 설명서의 [AWS CLI의 최신 버전 설치](#)를 참조하세요.
- Python 사용 경험이 있습니다.
- 대상 계정의 경우 AWS 계정 별칭이 IAM 콘솔을 설정합니다. 자세한 내용은 GitHub aws-nuke 리포지토리의 [주의](#) 사항을 참조하십시오. 지침은 IAM 설명서의 [계정 별칭 생성](#)을 참조하세요.
- 솔루션이 실행될 때 보고서를 받으려는 활성 이메일 주소입니다. 이 이메일 주소를 구독하면 이 패턴과 함께 제공된 CloudFormation 템플릿을 통해 배포하는 Amazon SNS 주제를 구독할 수 있습니다.

제한 사항

- 이 패턴은 종속성 위반 오류가 있는 시나리오에는 적용되지 않습니다. aws-nuke는 리소스가 삭제되거나 오류가 있는 리소스만 남을 때까지 모든 리소스를 다시 삭제하려고 다시 시도합니다. 자세한 내용은 GitHub aws-nuke 리포지토리의 [사용을](#) 참조하십시오.
- 이 솔루션은 샌드박스 및 개발 환경을 위해 설계되었습니다. 프로덕션 환경에서는 이 솔루션을 사용하지 마십시오.
- 이 솔루션은 단일 계정에 배포되며 해당 계정의 리소스만 삭제합니다. 이 솔루션을 확장하여 여러 계정의 리소스를 삭제하는 방법에 대한 자세한 내용은 이 패턴의 [아키텍처](#) 섹션에서 자동화 및 규모 조정을 참조하세요.
- 이 솔루션은 코드 리포지토리에 연결된 자동화된 배포 파이프라인을 제공하지 않습니다. 이 솔루션을 사용자 지정하여 AWS에서 소스 코드를 CodeCommit 호스팅하고 AWS에서 배포 파이프라인을 생성할 수 CodePipeline 있습니다.

제품 버전

- aws-`nuke` 버전 2.21.2 이상입니다. aws-`nuke` 버전을 업데이트할 때는 릴리스 노트를 검토하여 새 버전의 aws-`nuke`가 계정에서 제거하고 싶지 않은 새로운 유형의 리소스를 삭제하지 않는지 확인합니다.

아키텍처

대상 기술 스택

- EventBridge 규칙
- Step Functions 워크플로우
- Amazon Simple Notification Service(Amazon SNS) 주제
- CodeBuild 프로젝트
- Amazon Simple Storage Service(S3) 버킷
- 대상 계정의 IAM 역할

대상 아키텍처

이 다이어그램은 다음 프로세스를 보여줍니다.

1. EventBridge 규칙은 구성된 일정에 따라 Step Functions 워크플로를 호출합니다.
2. Step Functions 상태 머신은 파라미터를 수집하고, 맵 상태에서 Step Functions가 프로젝트를 호출합니다 CodeBuild .
3. CodeBuild 프로젝트는 전달된 파라미터를 사용하여 S3 버킷에서 `nuke_generic_config.yaml` 파일을 가져옵니다. CodeBuild 그런 다음 `nuke_config_update.py` 스크립트를 사용하여 구성 파일의 자리 표시자 속성을 대상 지역의 값으로 바꿉니다.
4. CodeBuild 프로젝트는 `nuke-auto-account-cleanser` IAM 역할을 맡고 각 대상 지역에서 aws-`nuke`를 시작합니다.
5. aws-`nuke`가 모의 실습 모드(기본값)인 경우 대상 리전에서 삭제할 리소스를 스캔하고 확인합니다.
aws-`nuke`가 모의 실습 모드가 아닌 경우 대상 리전의 리소스를 스캔하고 삭제합니다.
6. Step Functions 상태 머신은 성공 또는 실패 상태를 받을 때까지 CodeBuild 작업을 반복하고 폴링합니다. 작업이 실패할 경우 Step Functions는 구성된 횟수만큼 재시도합니다.
7. CodeBuild 프로젝트가 모든 지역에서 완료되면 Step Functions 워크플로는 Amazon SNS를 사용하여 각 지역의 빌드 상태에 대한 정보가 포함된 상세한 요약 보고서를 이메일로 보냅니다. 또한 각 리

전에 대해 별도의 이메일을 받게 되며, 여기에는 해당 리전에서 확인되거나 삭제된 리소스가 나열됩니다.

자동화 및 규모 조정

현재 이 패턴은 단일 계정의 여러 AWS 리전에서 자동화되고 확장 가능한 방식으로 aws-nuke 바이너리를 실행합니다. Step Functions의 맵 상태를 사용하면 aws-nuke가 각 리전에서 병렬로 실행됩니다. 이 동시 솔루션은 잠재적으로 많은 수의 리소스를 처리하고 장애 및 재시도 워크플로우를 독립적으로 처리하기에 충분한 시간을 제공합니다.

여러 계정의 리소스를 삭제하도록 이 솔루션을 수정하려면 hub-and-spoke 토폴로지를 사용하면 됩니다. CloudFormation 템플릿을 정의하고 사용하여 대상 스포크 계정에서 계정 간 IAM 역할을 구성할 수 있습니다. nuke-cfn-stack또한.yaml CloudFormation 템플릿을 수정하여 Step Functions 정의를 업데이트하여 맵 상태에서 반복할 계정 목록을 수락할 수 있습니다. Step Functions 워크플로를 중앙 허브 계정에 배포합니다. aws-nuke는 허브 계정의 CodeBuild 프로젝트에서 실행되며 리소스를 삭제하기 위해 대상 스포크 계정에서 계정 간 IAM 역할을 맡습니다.

도구

서비스

- [AWS는 AWS CloudFormation](#) 리소스를 설정하고, 빠르고 일관되게 프로비저닝하고, 전체 AWS 계정 및 지역의 수명 주기 전반에 걸쳐 리소스를 관리할 수 있도록 지원합니다.
- [CodeBuildAWS](#)는 소스 코드를 컴파일하고, 단위 테스트를 실행하고, 배포할 준비가 된 아티팩트를 생성하는 데 도움이 되는 완전관리형 빌드 서비스입니다.
- [EventBridgeAmazon](#)은 다양한 소스의 실시간 데이터와 애플리케이션을 연결하는 데 도움이 되는 서버리스 이벤트 버스 서비스입니다.
- [Amazon Simple Notification Service\(Amazon SNS\)](#)를 사용하면 웹 서버 및 이메일 주소를 포함하여 게시자와 클라이언트 간의 메시지 교환을 조정하고 관리할 수 있습니다.
- [Amazon Simple Storage Service\(S3\)](#)는 원하는 양의 데이터를 저장, 보호 및 검색하는 데 도움이 되는 클라우드 기반 객체 스토리지 서비스입니다.
- [AWS Step Functions](#)는 AWS Lambda 함수와 기타 AWS 서비스를 결합할 수 있는 서버리스 오케스트레이션 서비스로서 비즈니스 크리티컬 애플리케이션을 구축합니다.

기타 도구

- [aws-nuke](#)는 대상 AWS 계정 및 리전의 리소스를 삭제하는 데 도움이 되는 오픈 소스 도구입니다. 기본 리소스가 아니거나 AWS에서 관리하는 모든 리소스를 삭제합니다.
- [Python](#)은 범용 컴퓨터 프로그래밍 언어입니다.

코드 리포지토리

이 패턴의 코드는 [AWS-Nuke 리포지토리를 사용하는 GitHub AWS 계정 정리 도구 프레임워크에서](#) 사용할 수 있습니다. 여기에는 다음이 포함됩니다.

- `nuke_generic_config.yaml` — 이 CloudFormation 템플릿은 대상 지역의 리소스를 스캔하고 삭제하기 위해 AWS-nuke 바이너리에 필요한 구성 파일입니다. 이 파일에는 프로젝트 내에서 사용자 지정 Python 필터링 클래스를 사용하여 런타임에 동적으로 업데이트되는 일부 자리 표시자가 포함되어 있습니다. CodeBuild
- `nuke-cfn-stack.yaml` — 이 CloudFormation 템플릿은 이 솔루션을 운영하는 데 필요한 모든 샘플 리소스를 정의합니다. 이를 CloudFormation 스택으로 배포하면 대상 계정에 다음과 같은 리소스가 생성됩니다.
 - EventBridge 규칙
 - Step Functions 상태 시스템
 - 대상 지역의 샘플 CodeBuild 프로젝트
 - 대상 리전에서 무작위로 생성된 이름 및 버킷 정책을 사용하는 S3 버킷
 - 이메일 알림 수신을 위해 구독한 활성 이메일 주소가 있는 Amazon SNS 주제
 - 솔루션을 지원하는 IAM 역할 및 정책
- `nuke_config_update.py` - Python Config Parser라고도 하는 이 Python 스크립트는 Step Functions 워크플로우에서 정의한 입력 파라미터를 기반으로 런타임에 각 리전의 `nuke_generic_config.yaml` 파일을 구문 분석하고 동적으로 업데이트합니다. 이 스크립트에는 범용 태그를 기반으로 하는 사용자 지정 필터링 로직이 포함되어 있어 모든 글로벌 및 IAM 제외 목록을 필터링하고 처리하기 위한 추가 보호 계층이 추가됩니다. 또한 이 스크립트는 중요한 태그와 기타 메타데이터를 기반으로 스택 이름을 확인하여 해당 리소스가 삭제되지 않도록 합니다. 각 리전의 요구 사항에 따라 이 파일을 사용자 지정할 수 있습니다.

에픽

솔루션 설정

작업	설명	필요한 기술
<p>리포지토리를 복제합니다.</p>	<p>다음 명령을 실행하여 aws- nuke repo를 사용하여 GitHub AWS 계정 클렌저 프레임워크 를 복제합니다.</p> <pre data-bbox="594 646 1029 848">git clone https://github.com/aws-samples/aws- nuke-account-cleanser-example.git</pre>	<p>DevOps 엔지니어</p>
<p>대상 계정에서 스택을 생성합니다.</p>	<ol style="list-style-type: none"> 이 솔루션을 배포할 대상 계정과 리전을 식별합니다. 텍스트 nuke-cfn-stack 편집기에서 .yaml CloudFormation 템플릿을 엽니다. EventBridgeNukeSchedule 섹션에서 AWS Step Functions 워크플로를 실행하기 위한 일정을 사용자 지정합니다. 자세한 내용은 EventBridge 설명서의 Cron 표현식을 참조하십시오. 템플릿을 저장하고 닫습니다. CloudFormation nuke-cfn-stack CloudFormation .yaml 템플릿을 사용하여 스택을 생성합니다. 다음 명령을 입력합니다. <pre data-bbox="631 1791 1029 1885">aws cloudformation create-stack \</pre>	<p>AWS DevOps, 클라우드 관리자, AWS 관리자</p>

작업	설명	필요한 기술
<p>구성 파일을 수정합니다.</p>	<pre data-bbox="630 205 1024 583"> --stack-name NukeCleanser \ --template-body file://nuke-cfn-st ack.yaml \ --region <Region> \ --capabilities CAPABILITY_NAMED_I AM \ </pre> <p>복제된 리포지토리의 <code>aws-nuke-account-cleanser-example</code> 폴더에서 <code>nuke_generic_config.yaml</code> 파일을 편집하여 사용 사례에 맞게 사용자 지정합니다. aws-nuke 구성 파일을 사용자 지정하는 방법에 대한 자세한 내용은 aws-nuke 저장소에서의 사용을 참조하십시오. GitHub</p> <p>중요: <code>TARGET_REGION</code> 및 <code>ACCOUNT</code> 자리 표시자 값은 변경하지 마십시오. 이들은 런타임 시 동적으로 업데이트됩니다.</p>	<p>DevOps 엔지니어, 클라우드 관리자</p>

작업	설명	필요한 기술
S3 버킷을 준비합니다.	<p>1. 수정된 <code>nuke_generic_config.yaml</code> 및 <code>nuke_config_update.py</code> 파일을 스택을 배포할 때 생성된 S3 버킷에 업로드합니다. CloudFormation 다음 명령을 입력하고 <code><Region></code> 자리 표시자를 대상 리전으로 변경합니다.</p> <pre data-bbox="634 636 1029 1507">aws s3 cp \ config/nuke_generic_config.yaml --region <Region> \ s3://nuke-account-cleanser-config-{AWS::AccountId}-{AWS::Region}-{random-id-generated} aws s3 cp \ config/nuke_config_update.py --region <Region> \ s3://nuke-account-cleanser-config-{AWS::AccountId}-{AWS::Region}-{random-id-generated}</pre> <p>2. 에서 GitHub AWS-Nuke 바이너리를 CodeBuild 다운로드하므로 이 솔루션을 실행 중인 VPC (가상 사설 클라우드) 에서 네트워크 연결이 충분한지 확인하십시오. 제한된 환경에서 실행하거나 대</p>	클라우드 관리자, 엔지니어 DevOps

작업	설명	필요한 기술
	역폭이 충분하지 않은 경우 aws-nuke 바이너리를 S3 버킷에 업로드하거나 내부 리포지토리에서 소싱합니다.	

솔루션 테스트

작업	설명	필요한 기술
Step Functions 워크플로우를 수동으로 시작합니다.	<p>이 nuke-cfn-stack 솔루션은 .yaml 파일의 EventBridge 규칙에서 구성한 일정에 따라 자동으로 실행되도록 구성되어 있습니다. 솔루션을 수동으로 실행하려면 다음 명령을 입력하고 <Region> 자료 표시자를 솔루션을 실행하려는 대상 리전으로 변경합니다.</p> <pre> { "InputPayload": { "nuke_dry_run": "true", "nuke_version": "2.21.2", "region_list": ["<Region A>", "<Region B>", "global"] } } </pre> <p>이 명령을 입력하면 Step Functions 워크플로가 각 지역</p>	AWS DevOps, DevOps 엔지니어

작업	설명	필요한 기술
	에서 별도의 CodeBuild 프로젝트를 시작합니다.	
(선택 사항) 진행 상황을 모니터링합니다.	Amazon CloudWatch Logs에서 로그 이벤트를 쿼리하여 진행 상황을 모니터링할 수 있습니다. 샘플 쿼리는 이 패턴의 추가 정보 섹션을 참조하세요.	AWS DevOps, AWS 시스템 관리자, DevOps 엔지니어
결과를 확인합니다.	<ol style="list-style-type: none"> 1. aws-nuke가 대상 지역 스캔을 완료할 수 있도록 허용합니다. CodeBuild 프로젝트가 한 지역에서 완료되면 워크플로는 구성된 SNS 주제를 통해 해당 지역의 자세한 결과 보고서를 이메일로 보냅니다. 리전마다 별도의 보고서를 받습니다. 모든 지역이 완료되고 Step Functions 워크플로가 성공하면 워크플로우는 각 CodeBuild 작업의 완료 상태를 요약하는 또 다른 이메일을 보냅니다. 이 보고서의 샘플은 추가 정보 섹션을 참조하세요. 2. 보고서의 결과를 검토합니다. 	AWS DevOps, DevOps 엔지니어

리소스 삭제

작업	설명	필요한 기술
삭제하지 않으려는 리소스는 제외합니다.	1. 모의 실행 모드에서 솔루션을 실행한 테스트의 출력 결과를 검토합니다.	AWS 시스템 관리자, DevOps 엔지니어, AWS 관리자

작업	설명	필요한 기술
	<p>2. 유지하려는 리소스가 확인되면 <code>nuke_generic_config.yaml</code> 파일을 수정하여 해당 리소스를 제외합니다.</p> <p>참고: 이 파일은 이 솔루션에서 배포한 리소스를 제외하도록 이미 구성되어 있습니다.</p> <p>3. 다음 명령을 입력하여 수정된 구성 파일을 S3 버킷에 업로드합니다.</p> <pre data-bbox="630 800 1029 1276">aws s3 cp \ config/nuke_generic_config.yaml --region <Region> \ s3://nuke-account-cleanser-config-{AWS::AccountId}-{AWS::Region}-{random-id-generated}</pre>	

작업	설명	필요한 기술
<p>실행 모드를 변경합니다.</p>	<p>식별된 리소스를 삭제할 준비가 되었는지 확인하고 보존할 리소스를 모두 제외했으면 프로덕션 모드에서 솔루션을 실행할 수 있습니다. 프로덕션 모드에서는 기본값이 아니거나, AWS에서 aws-nuke-config 관리하거나.yaml 파일에 제외된 모든 리소스를 삭제합니다. 프로덕션 모드로 변경하거나 모의 실행을 비활성화하려면 AWSNukeDryRunFlag 파라미터를 false(으)로 변경해야 합니다. 설명서의 스택 템플릿 수정에 나와 있는 지침에 따라 스택을 수정하십시오. CloudFormation 이렇게 하면 EventBridge 규칙에서 Step Functions 상태 머신 타겟으로 전달되는 입력 페이로드가 변경됩니다.</p>	<p>AWS 관리자, AWS 시스템 관리자, DevOps 엔지니어</p>

작업	설명	필요한 기술
<p>Step Functions 워크플로우를 수동으로 시작합니다.</p>	<p>다음 명령을 입력하여 솔루션을 수동으로 실행하고 <Region> 자리 표시자를 솔루션을 실행하려는 대상 리전으로 변경합니다.</p> <pre data-bbox="592 489 1027 1005"> { "InputPayload": { "nuke_dry_run": "false", "nuke_version": "2.21.2", "region_list": ["<Region A>", "<Region B>"] } } </pre>	<p>AWS DevOps, DevOps 엔지니어</p>
<p>(선택 사항) 진행 상황을 모니터링합니다.</p>	<p>Amazon CloudWatch Logs에서 로그 이벤트를 쿼리하여 진행 상황을 모니터링할 수 있습니다. 샘플 쿼리는 이 패턴의 추가 정보 섹션을 참조하세요.</p>	<p>AWS 관리자, AWS 시스템 관리자, DevOps 엔지니어</p>
<p>결과를 확인합니다.</p>	<p>워크플로우가 완료될 때까지 기다립니다. 보고서를 받으면 결과를 확인하고 리소스가 성공적으로 삭제되었는지 확인합니다. 이제 EventBridge 규칙에서 구성한 일정에 따라 솔루션이 자동으로 실행됩니다.</p>	<p>DevOps 엔지니어, AWS DevOps</p>

관련 리소스

AWS 설명서

- [CloudFormation 템플릿 사용](#)
- [CloudFormation 스택 다루기](#)
- [일정에 따라 실행되는 Amazon EventBridge 규칙 생성](#)
- [Step Functions의 작동 방식](#)
- [Step Functions의 맵 상태 처리 모드](#)

GitHub 리포지토리

- [aws-nuke](#)
- [aws-nuke-account-cleanser](#)

추가 정보

쿼리 모니터링

Amazon Logs에서 로그 이벤트를 쿼리하여 aws-nuke의 진행 상황을 모니터링할 수 있습니다.

CloudWatch

다음은 AWS CLI의 샘플 쿼리입니다. 자세한 내용은 AWS CLI 명령 참조를 참조하십시오 [filter-log-events](#).

```
aws logs filter-log-events \  
  --log-group-name AccountNuker-nuke-auto-account-cleanser \  
  --start-time <value> \  
  --end-time <value> \  
  --log-stream-names <value> \  
  --filter-pattern removed \  
  --no-interleaved \  
  --output text \  
  --limit 5
```

다음은 CloudWatch Logs Insights의 샘플 쿼리 명령입니다. 자세한 내용은 CloudWatch 설명서에서 [CloudWatch Logs Insights를 사용한 로그 데이터 분석](#)을 참조하십시오.

```
fields @timestamp, @message  
| filter userIdentity.sessionContext.sessionIssuer.userName = "nuke-auto-account-cleanser" and ispresent(errorCode)  
| sort @timestamp desc
```

```

| limit 500

fields @timestamp, @message
| filter ispresent(errorCode) and userIdentity.sessionContext.sessionIssuer.userName =
  "nuke-auto-account-cleanser"
and errorCode != "AccessDenied" and eventName like "Delete"
| sort @timestamp desc
| limit 500

fields @timestamp, @message
| filter ispresent(errorCode) and userIdentity.sessionContext.sessionIssuer.userName =
  "nuke-auto-account-cleanser"
and errorCode == "AccessDenied" and eventName like "Delete"
| sort @timestamp desc
| limit 500

```

이메일 보고서

Step Functions 상태 맵은 각 지역에 대해 CodeBuild 작업을 한 번 재시도합니다. 오류나 재시도가 발생하는 경우 각 작업에 대해 별도의 이메일을 받게 됩니다. 이메일 콘텐츠 및 출력은 CodeBuild 프로젝트의 BuildSpec 섹션 내에서 구성됩니다. AWS CLI 명령과 기본 Linux 스크립팅을 사용하여 aws-nuke 바이너리가 생성하는 로그 파일에서 관련 정보를 추출합니다. 필요에 따라 이메일 보고 템플릿을 사용자 지정할 수 있습니다.

샘플 출력

다음은 Step Functions 워크플로우가 프로덕션 모드에서 성공적으로 완료될 때 전송되는 알림 및 보고서의 샘플입니다.

```

Account Cleansing Process Completed;

-----
Summary of the process:
-----

DryRunMode           : false
Account ID           : 000000000000
Target Region        : us-west-1
Build State           : JOB SUCCEEDED
Build ID              : AccountNuker-NukeCleanser:a0761233-578e-4f23-8a2d-
c123215a1bef
CodeBuild Project Name : AccountNuker-NukeCleanser

```

```

Process Start Time      : Tue Mar 28 18:20:13 UTC 2023
Process End Time       : Tue Mar 28 18:20:48 UTC 2023
Log Stream Path        : AccountNuker-NukeCleanser/a0761233-578e-4f23-8a2d-
c123215a1bef

```

```

-----
##### Nuke Cleanser Logs #####

```

```

Number of Resources that is filtered by config:
2

```

```

-----
FAILED RESOURCES

```

```

-----
SUCCESSFULLY NUKED RESOURCES

```

```

us-west-1 - S3Bucket - s3://samples3bucket-nuke - [CreationDate: "2023-03-27 21:24:59
+0000 UTC", Name: "samples3bucket-nuke"] - removed

```

```

us-west-1 - S3Bucket - s3://test-nuke-s3-bucket - [CreationDate: "2023-03-28 14:27:06
+0000 UTC", Name: "test-nuke-s3-bucket"] - removed

```

```

us-west-1 - SQSQueue - https://sqs.us-west-1.amazonaws.com/000000000000/sample-test-
nuke-queue - removed

```

```

us-west-1 - S3Bucket - s3://samples3bucket-nuke - [CreationDate: "2023-03-27 21:24:59
+0000 UTC", Name: "samples3bucket-nuke"] - removed

```

```

us-west-1 - S3Bucket - s3://test-nuke-s3-bucket - [CreationDate: "2023-03-28 14:27:06
+0000 UTC", Name: "test-nuke-s3-bucket"] - removed

```

```

us-west-1 - SQSQueue - https://sqs.us-west-1.amazonaws.com/000000000000/sample-test-
nuke-queue - removed

```

다음은 Step Functions 워크플로우가 모의 실행 모드에서 성공적으로 완료될 때 전송되는 알림 및 보고서의 샘플입니다.

```

Account Cleansing Process Completed;

```

```

-----
Summary of the process:

```

```

-----
DryRunMode           : true
Account ID           : 000000000000
Target Region        : us-west-1

```

```

Build State           : JOB SUCCEEDED
Build ID              : AccountNuker-
NukeCleanser:69e0d2de-5f48-46cf-98f3-2df22d11991e
CodeBuild Project Name : AccountNuker-NukeCleanser
Process Start Time    : Mon Mar 27 19:42:49 UTC 2023
Process End Time      : Mon Mar 27 19:43:30 UTC 2023
Log Stream Path       : AccountNuker-
NukeCleanser/69e0d2de-5f48-46cf-98f3-2df22d11991e
-----
##### Nuke Cleanser Logs #####

Number of Resources that is filtered by config:
1
-----
RESOURCES THAT WOULD BE REMOVED:
-----
3
us-west-1 - SQSQueue - https://sqs.us-east-1.amazonaws.com/000000000000/test-nuke-queue
- would remove
us-west-1 - SNSTopic - TopicARN: arn:aws:sns:us-east-1: 000000000000:test-nuke-topic -
[TopicARN: "arn:aws:sns:us-east-1: 000000000000:test-topic"] - would remove
us-west-1 - S3Bucket - s3://test-nuke-bucket-us-west-1 - [CreationDate: "2023-01-25
11:13:14 +0000 UTC", Name: "test-nuke-bucket-us-west-1"] - would remove

```

AWS Systems Manager Maintenance Windows를 사용하여 Amazon RDS DB 인스턴스를 자동 중지 및 시작

작성자: Ashita Dsilva(AWS)

환경: 프로덕션

기술: 관리 및 거버넌스, 비용 관리, 데이터베이스, 클라우드 네이티브

AWS 서비스: AWS Systems Manager, Amazon RDS

요약

이 패턴은 AWS Systems Manager Maintenance Windows를 사용하여 특정 일정에 따라 Amazon Relational Database Service(RDS) DB 인스턴스를 자동으로 중지하고 시작하는 방법(예: 비용 절감을 위해 업무 시간 외에 DB 인스턴스를 종료)을 보여줍니다.

AWS Systems Manager Automation은 Amazon RDS DB 인스턴스를 중지하고 시작하는 데 필요한 AWS-StopRdsInstance 및 AWS-StartRdsInstance 런북을 제공합니다. 즉, AWS Lambda 함수를 사용하여 사용자 지정 로직을 작성하거나 Amazon CloudWatch Events 규칙을 생성할 필요가 없습니다.

AWS Systems Manager는 작업 예약을 위한 두 가지 기능, 즉 [State Manager](#) 및 [Maintenance Windows](#)를 제공합니다. State Manager는 Amazon Web Services(AWS) 계정의 리소스에 필요한 상태 구성을 한 번 또는 특정 일정에 따라 설정하고 유지 관리합니다. Maintenance Windows는 특정 기간 동안 계정의 리소스에서 작업을 실행합니다. 이 패턴의 접근 방식을 State Manager 또는 Maintenance Windows와 함께 사용할 수 있지만, Maintenance Windows는 할당된 우선 순위에 따라 하나 이상의 작업을 실행할 수 있고 AWS Lambda 함수 및 AWS Step Functions 작업도 실행할 수 있으므로 Maintenance Windows를 사용하는 것이 좋습니다. State Manager 및 Maintenance Windows에 대한 자세한 내용은 AWS Systems Manager 설명서에서 [State Manager와 Maintenance Windows 중 선택](#)을 참조하세요.

이 패턴은 cron 표현식을 사용하여 Amazon RDS DB 인스턴스를 중지한 후 시작하는 두 개별 유지 관리 기간을 구성하는 세부 단계를 제공합니다.

사전 조건 및 제한 사항

사전 조건

- 활성 상태의 AWS 계정.
- 중지하고 특정 일정에 따라 시작하려는 기존 Amazon RDS DB 인스턴스.
- 필요한 일정에 맞는 Cron 표현식. 예를 들어 (0 9 * * 1-5) cron 표현식은 월요일부터 금요일까지 오전 9시에 실행됩니다.
- Systems Manager 속지.

제한 사항

- Amazon RDS DB 인스턴스는 한 번에 최대 7일 동안 중지할 수 있습니다. 7일이 지나면 DB 인스턴스가 자동으로 다시 시작되어 필요한 유지 관리 업데이트를 받을 수 있습니다.
- 읽기 전용 복제본인 또는 읽기 전용 복제본을 포함한 DB 인스턴스는 중지할 수 없습니다.
- 다중 AZ 구성에서는 Amazon RDS for SQL Server DB 인스턴스를 중지할 수 없습니다.
- Service Quotas는 Maintenance Windows 및 Systems Manager Automation에 적용됩니다. Service Quotas에 대한 자세한 내용은 AWS General Reference 설명서의 [AWS Systems Manager 엔드포인트 및 할당량](#)을 참조하세요.

아키텍처

다음 다이어그램은 Amazon RDS DB 인스턴스를 자동으로 중지하고 시작하는 워크플로를 보여줍니다.

워크플로는 다음 단계로 구성됩니다.

1. 유지 관리 기간을 만들고 cron 표현식을 사용하여 Amazon RDS DB 인스턴스의 중지 및 시작 일정을 정의합니다.
2. AWS-StopRdsInstance 또는 AWS-StartRdsInstance 런북을 사용하여 유지 관리 기간에 Systems Manager Automation 작업을 등록합니다.
3. Amazon RDS DB 인스턴스의 태그 기반 리소스 그룹을 사용하여 유지 관리 기간에 대상을 등록합니다.

기술 스택

- AWS CloudFormation
- AWS Identity and Access Management(IAM)

- Amazon RDS
- Systems Manager

자동화 및 규모 조정

필요한 Amazon RDS DB 인스턴스에 태그를 지정하고, 태그가 지정된 모든 DB 인스턴스를 포함하는 리소스 그룹을 만들고, 이 리소스 그룹을 유지 관리 기간의 대상으로 등록하여 여러 Amazon RDS DB 인스턴스를 동시에 중지하고 시작할 수 있습니다.

도구

- [CloudFormation](#)은 AWS 리소스를 모델링하고 설정하는 데 도움이 되는 서비스입니다.
- [AWS ID 및 액세스 관리 \(IAM\)](#) 는 AWS 리소스에 대한 액세스를 안전하게 제어하는 데 도움이 되는 웹 서비스입니다.
- [Amazon 관계형 데이터베이스 서비스 \(Amazon RDS\)](#) 는 AWS 클라우드에서 관계형 데이터베이스를 더 쉽게 설치, 운영 및 확장할 수 있게 해주는 웹 서비스입니다.
- [AWS Resource Groups](#)를 사용하면 AWS 리소스를 그룹으로 구성하고, 리소스에 태그를 지정하고, 그룹화된 리소스의 작업을 관리, 모니터링 및 자동화할 수 있습니다.
- [AWS Systems Manager](#)는 AWS 기반 인프라를 보고 제어하는 데 사용할 수 있는 AWS 서비스입니다.
- [AWS Systems Manager Automation](#)은 Amazon Elastic Compute Cloud(Amazon EC2) 인스턴스 및 기타 AWS 리소스의 일반적인 유지 관리 및 배포 작업을 간소화합니다.
- [AWS Systems Manager 유지 관리 윈도우](#)를 사용하면 인스턴스에 지장을 일으킬 수 있는 작업을 언제 수행할지 일정을 정의할 수 있습니다.

에픽

IAM을 생성하고 Systems Manager Automation에 대한 IAM 서비스 역할을 구성합니다.

작업	설명	필요한 기술
Systems Manager Automation에 대한 서비스 역할을 구성합니다.	AWS Management Console에 로그인하고 Systems Manager Automation에 대한 서비스 역할을 생성합니다. 다음 두 방법	AWS 관리자

작업	설명	필요한 기술
	<p>중 하나를 사용하여 이 서비스 역할을 생성할 수 있습니다.</p> <ul style="list-style-type: none"> • AWS를 CloudFormation 사용하여 Systems Manager 자동화에 대한 서비스 역할을 구성하십시오. • IAM을 사용하여 Systems Manager Automation에 대한 역할 구성 <p>Systems Manager Automation 워크플로는 서비스 역할을 사용하여 Amazon RDS DB 인스턴스에서 시작 및 중지 작업을 수행하여 Amazon RDS를 간접적으로 호출합니다.</p> <p>Amazon RDS DB 인스턴스를 시작하고 중지할 권한이 있는 다음과 같은 인라인 정책으로 서비스 역할을 구성해야 합니다.</p> <pre data-bbox="592 1325 1029 1816"> { "Version": "2012-10-17", "Statement": [{ "Sid": "RdsStartStop", "Effect": "Allow", "Action": ["rds:StopDBInstance", </pre>	

작업	설명	필요한 기술
	<pre data-bbox="592 241 1031 1018"> "rds:StartDBInstance"], "Resource": "<RDS_Instance_ARN>" }, { "Sid": "RdsDescribe", "Effect": "Allow", "Action": "rds:DescribeDBInstances", "Resource": "*" }] } </pre> <p data-bbox="592 1060 1031 1239"><RDS_Instance_ARN> 을 Amazon RDS DB 인스턴스의 Amazon 리소스 이름(ARN)으로 바꿔야 합니다.</p> <p data-bbox="592 1281 1031 1354">중요: 서비스 역할의 ARN을 기록해야 합니다.</p>	

리소스 그룹 생성

작업	설명	필요한 기술
Amazon RDS DB 인스턴스에 태그를 지정합니다.	Amazon RDS 콘솔 을 열고 리소스 그룹에 추가하려는 Amazon RDS DB 인스턴스에 태그를 지정합니다. 태그는 AWS 리소스에 할당된 메	AWS 관리자

작업	설명	필요한 기술
	<p>타데이터이며 키-값 페어로 구성됩니다. Action을 태그 키와 StartStop값으로 사용하는 것이 좋습니다.</p> <p>이에 대한 자세한 내용은 Amazon RDS 설명서의 태그 추가, 나열 및 제거를 참조하세요.</p>	
<p>태그가 지정된 Amazon RDS DB 인스턴스용 리소스 그룹을 생성합니다.</p>	<p>AWS Resource Groups 콘솔을 열고 Amazon RDS DB 인스턴스용으로 생성한 태그를 기반으로 리소스 그룹을 생성합니다.</p> <p>그룹화 기준에서 리소스 유형으로 AWS: :RDS: :DbInstance를 선택한 다음 태그의 키-값 쌍 (예: "Action-") 을 제공해야 합니다. StartStop 이렇게 하면 서비스가 Amazon RDS DB 인스턴스만 확인하고 이 태그가 있는 다른 리소스는 확인하지 않습니다. 리소스 그룹 이름을 기록해 두어야 합니다.</p> <p>자세한 내용 및 세부 단계는 AWS Resource Groups 설명서의 태그 기반 쿼리 빌드 및 그룹 생성을 참조하세요.</p>	<p>AWS 관리자</p>

Amazon RDS DB 인스턴스를 중지하도록 유지 관리 기간을 구성하세요.

작업	설명	필요한 기술
유지 관리 기간을 생성합니다.	<ol style="list-style-type: none"> <li data-bbox="591 327 1024 743">1. AWS Systems Manager 콘솔을 열고 유지 관리 기간을 선택한 다음에 유지 관리 기간 생성을 선택합니다. 유지 관리 기간의 이름 (예: "StopRdsInstance") 을 입력하고 설명을 입력한 다음 등록되지 않은 대상 허용을 선택 취소하십시오. <li data-bbox="591 764 1024 1325">2. CRON/rate 표현식을 선택하고 Amazon RDS DB 인스턴스를 중지해야 하는 시기를 정의하는 일정 표현식을 제공합니다. 기간에는 1을 입력하고 작업 시작 중지에는 0을 입력합니다. 기본적으로 시간대는 UTC로 표시됩니다. cron 표현식에 정의된 타임스탬프를 기반으로 시간대를 변경하여 유지 관리 기간을 시작할 수 있습니다. <li data-bbox="591 1346 1024 1577">3. 유지 관리 기간 생성을 선택합니다. 시스템에서 유지 관리 기간 페이지로 돌아가고 유지 관리 기간의 상태가 활성화됩니다. <p data-bbox="591 1650 1024 1879">중요: DB 인스턴스를 중지하는 작업은 시작과 동시에 거의 즉시 실행되며 유지 관리 기간의 전체 기간을 포함하지는 않습니다. 이 패턴은 유지 관리 기간</p>	AWS 관리자

작업	설명	필요한 기술
	<p>의 필수 매개 변수이므로 기간 및 작업 시작 중지 에 대한 최소 값을 제공합니다.</p> <p>자세한 내용 및 세부 단계는 AWS Systems Manager 설명서의 유지 관리 기간 생성(콘솔)을 참조하세요.</p>	
<p>유지 관리 기간에 대상을 할당합니다.</p>	<ol style="list-style-type: none"> 1. AWS Systems Manager 콘솔에서 유지 관리 기간을 선택하고 작업을 선택한 다음에 대상 등록을 선택합니다. 2. 대상 영역에서 리소스 그룹 선택을 지정한 다음에 계정에 있는 기존 리소스 그룹의 이름을 선택합니다. 3. 리소스 유형으로는 <code>AWS::RDS::DBInstance</code>를 선택한 다음에 대상 등록을 선택합니다. <p>자세한 내용 및 세부 단계는 AWS Systems Manager 설명서에서 유지 관리 기간(콘솔)에 대상 할당을 참조하세요.</p>	<p>AWS 관리자</p>

작업	설명	필요한 기술
<p>유지 관리 기간에 태스크를 할당합니다.</p>	<ol style="list-style-type: none"> 1. AWS Systems Manager 콘솔에서 유지 관리 기간을 선택한 후 해당 유지 관리 기간을 선택합니다. 작업, 자동화 작업 등록을 차례로 선택합니다. 2. 문서에서는 AWS-를 선택합니다 StopRdsInstance. 3. 대상 섹션에서 등록된 대상 그룹 선택을 선택한 다음에 현재 유지 관리 기간에 등록된 유지 관리 기간 대상을 선택합니다. 4. 속도 제어의 경우 동시성 및 오류 임계값을 100% 로 지정합니다. 작업 동시성 및 오류 임계값에 대한 요구 사항에 따라 속도 제어 값을 변경할 수 있습니다. 이에 대한 자세한 내용은 AWS Systems Manager 설명서의 동시성 및 오류 임계값에 대한 정보를 참조하세요. 5. IAM 서비스 역할 섹션의 서비스 역할에서 이 상자를 비워 두거나 사용자 지정 역할을 생성하십시오. 이 상자를 비워 두면 Systems Manager가 자동으로 서비스 연결 역할을 AWSServiceRoleForAmazonSSM 만든 다음 역할을 작업에 연결합니다. 사용자 지정 역할을 직접 만들려면 유지 관리 기간 	<p>AWS 관리자</p>

작업	설명	필요한 기술
	<p>(콘솔) 용 사용자 지정 서비스 역할 만들기를 참조한 다음 해당 사용자 지정 역할을 작업에 연결합니다.</p> <p>6. 입력 파라미터 섹션에서 런북에 대한 다음의 파라미터를 지정합니다.</p> <ul style="list-style-type: none"> • InstanceId: {{RESOURCE_ID}} • AutomationAssumeRole: Systems Manager 자동화를 위해 생성한 서비스 역할의 ARN을 제공합니다. • 참고: 의 경우 InstanceId, ARN에서 Amazon RDS DB 리소스 ID를 추출하는데 유사 파라미터가 사용됩니다. 유사 파라미터에 대해 자세히 알아보려면 AWS Systems Manager 설명서에서 유사 파라미터에 대한 정보를 참조하세요. <p>7. 자동화 작업 등록을 선택합니다.</p> <p>중요: 서비스 역할 옵션은 유지 관리 기간 동안 작업을 실행하는 데 필요한 서비스 역할을 정의합니다. 하지만 이 역할은 이전에 Systems Manager Automation을 위해 만든 서비스 역할과 동일하지 않습니다.</p>	

작업	설명	필요한 기술
	자세한 내용 및 세부 단계는 AWS Systems Manager 설명서의 유지 관리 기간(콘솔)에 작업 할당 을 참조하세요.	

Amazon RDS DB 인스턴스를 시작하도록 유지 관리 기간을 구성하세요.

작업	설명	필요한 기술
유지 관리 기간을 구성하여 Amazon RDS DB 인스턴스를 시작합니다.	Amazon RDS DB 인스턴스를 중지하도록 유지 관리 기간 구성 에픽의 단계를 반복하여 예약된 시간에 Amazon RDS DB 인스턴스를 시작하도록 다른 유지 관리 기간을 구성합니다. 중요: DB 인스턴스를 시작하도록 유지 관리 기간을 구성할 때 다음과 같이 변경해야 합니다. <ul style="list-style-type: none"> • 유지 관리 기간에 새 이름을 사용합니다 (예: ""). StartRdsInstance • cron 표현식을 DB 인스턴스를 시작하는 데 사용할 cron 표현식으로 바꾸세요. • 작업의 AWS-StopRdsInstance 런북을 AWS-StartRdsInstance 로 바꿉니다. 	AWS 관리자

관련 리소스

- [Systems Manager Automation](#) 문서를 사용하여 인스턴스를 관리하고 근무 시간 외 비용 절감(AWS 블로그 게시물)

Terraform을 사용하여 AWS Organizations에서 소프트웨어 패키지 배포를 중앙 집중화합니다.

제작: 프라딧 쿠마르 판디 (AWS), 아르티 라즈푸트 (AWS), 친타마니 아파레 (AWS), T.V.R.L. 파니 쿠마르 다디 (AWS), 마유리 신데 (AWS), 프라탑 쿠마르 난다 (AWS)

환경: 프로덕션

기술: 관리 및 거버넌스, 인프라

AWS 서비스: AWS 조직, AWS Systems Manager

요약

기업은 워크로드 간에 강력한 격리 장벽을 만들기 위해 여러 AWS 리전 곳에 분산된 여러 개를 유지 관리하는 경우가 많습니다. AWS 계정 [보안 및 규정 준수를 유지하기 위해 관리 팀은, 또는 보안 스캔을 위한 도구와 Amazon CloudWatch 에이전트 CrowdStrikeSentinelOne, Datadog Agent 또는 모니터링용 에이전트와 같은 에이전트 기반 TrendMicro도구를 설치합니다. AppDynamics](#) 이러한 팀은 이러한 대규모 환경에서 소프트웨어 패키지 관리 및 배포를 중앙에서 자동화하려는 경우 종종 어려움을 겪습니다.

[배포자 \(Distributor\)](#) 는 단순화된 단일 인터페이스를 통해 클라우드 및 온프레미스 서버 전반의 관리형 Microsoft Windows 및 Linux 인스턴스에 소프트웨어를 패키징하고 게시하는 프로세스를 자동화합니다. [AWS Systems Manager](#) 이 패턴은 Terraform을 사용하여 소프트웨어 설치 관리 프로세스를 더욱 단순화하고 최소한의 노력으로 다수의 인스턴스와 구성원 계정에서 스크립트를 실행하는 방법을 보여줍니다. AWS Organizations

이 솔루션은 Systems Manager에서 관리하는 Amazon, Linux 및 Windows 인스턴스에서 작동합니다.

사전 조건 및 제한 사항

- 설치할 소프트웨어가 들어 있는 [디스트리뷰터 패키지](#)
- [테라폼](#) 버전 0.15.0 이상
- [Systems Manager에서](#) 관리하고 대상 계정의 아마존 [심플 스토리지 서비스 \(Amazon S3\) 에 액세스할 수 있는 기본 권한을 가진 Amazon Elastic Compute Cloud \(Amazon EC2\) 인스턴스](#)
- 다음을 사용하여 설정된 조직의 랜딩 존 [AWS Control Tower](#)
- (선택 사항) [테라폼용 어카운트 팩토리 \(AFT\)](#)

아키텍처

리소스 세부 정보

이 패턴은 [Account Factory for Terraform \(AFT\)](#) 을 사용하여 필요한 모든 AWS 리소스를 생성하고 코드 파이프라인을 사용하여 배포 계정에 리소스를 배포합니다. 코드 파이프라인은 두 개의 리포지토리에서 실행됩니다.

- 글로벌 사용자 지정에는 AFT에 등록된 모든 계정에서 실행되는 Terraform 코드가 포함되어 있습니다.
- 계정 사용자 지정에는 배포 계정에서 실행할 Terraform 코드가 포함됩니다.

계정 사용자 지정 폴더에서 [Terraform](#) 명령을 실행하여 AFT를 사용하지 않고 이 솔루션을 배포할 수도 있습니다.

Terraform 코드는 다음 리소스를 배포합니다.

- AWS Identity and Access Management(IAM) 역할 및 정책
 - [SystemsManager](#)- 대상 계정에서 자동화를 실행할 권한을 사용자에게 AutomationExecutionRole 부여합니다.
 - [SystemsManager](#)- 사용자에게 여러 계정 및 OU (조직 단위) 에서 자동화를 실행할 수 있는 권한을 AutomationAdministrationRole 부여합니다.
- 패키지의 압축 파일 및 manifest.json
 - Systems Manager의 [패키지에는](#) 소프트웨어 또는 설치 가능한 에셋의.zip 파일이 하나 이상 포함되어 있습니다.
 - JSON 매니페스트에는 패키지 코드 파일에 대한 포인터가 포함되어 있습니다.
- S3 버킷
 - 조직 전체에서 공유되는 분산 패키지는 Amazon S3 버킷에 안전하게 저장됩니다.
- AWS Systems Manager문서 (SSM 문서)
 - DistributeSoftwarePackage멤버 계정의 모든 대상 인스턴스에 소프트웨어 패키지를 배포하는 로직이 포함되어 있습니다.
 - AddSoftwarePackageToDistributor설치 가능한 소프트웨어 자산을 패키징하여 Automation에 추가하는 로직 (기능) 이 포함되어 있습니다. AWS Systems Manager
- Systems Manager 연결
 - Systems Manager 연결을 사용하여 솔루션을 배포합니다.

아키텍처 및 워크플로

다이어그램은 다음 단계들을 보여줍니다.

1. 중앙 집중식 계정에서 솔루션을 실행하려면 배포 단계와 함께 패키지 또는 소프트웨어를 S3 버킷에 업로드합니다.
2. 사용자 정의된 패키지는 Systems Manager 콘솔 [문서](#) 섹션의 Owned by me 탭에서 사용할 수 있습니다.
3. Systems Manager의 기능인 State Manager는 조직 전체에서 패키지에 대한 연결을 만들고, 일정을 잡고, 실행합니다. 연결은 소프트웨어 패키지를 대상 노드에 설치하려면 먼저 관리 노드에 소프트웨어 패키지를 설치하고 실행해야 한다고 지정합니다.
4. 연결은 Systems Manager에 대상 노드에 패키지를 설치하도록 지시합니다.
5. 후속 설치 또는 변경의 경우 사용자는 단일 위치에서 정기적으로 또는 수동으로 동일한 연결을 실행하여 계정 전체에 배포를 수행할 수 있습니다.
6. 회원 계정에서 자동화는 배포자에게 배포 명령을 보냅니다.
7. 배포자는 소프트웨어 패키지를 인스턴스 전체에 배포합니다.

이 솔루션은 내의 관리 계정을 사용하지만 AWS Organizations 조직을 대신하여 관리 계정을 관리할 계정 (위임된 관리자) 을 지정할 수도 있습니다.

도구

AWS 서비스

- [Amazon Simple Storage Service\(S3\)](#)는 원하는 양의 데이터를 저장, 보호 및 검색하는 데 도움이 되는 클라우드 기반 객체 스토리지 서비스입니다. 이 패턴은 Amazon S3를 사용하여 분산 패키지를 중앙 집중화하고 안전하게 저장합니다.
- [AWS Systems Manager](#)은 AWS 클라우드에서 실행되는 애플리케이션 및 인프라를 관리하는 데 도움을 줍니다. Systems Manager는 애플리케이션 및 리소스 관리를 간소화하고, 운영 문제를 감지하고 해결하는 시간을 단축하며, AWS 리소스를 대규모로 안전하게 관리하는 데 도움이 됩니다. 이 패턴은 다음과 같은 Systems Manager 기능을 사용합니다.
 - [배포자](#)를 통해 소프트웨어를 패키징하고 Systems Manager 관리형 인스턴스에 게시할 수 있습니다.
 - [자동화](#)는 많은 서비스에 대한 일반적인 유지 관리, 배포 및 수정 작업을 간소화합니다. AWS

- [문서](#)는 조직 및 계정 전체의 Systems Manager 관리형 인스턴스에서 작업을 수행합니다.
- [AWS Organizations](#) 여러 계정을 하나의 조직으로 통합하여 사용자가 만들고 중앙에서 관리하는 데 도움이 되는 AWS 계정 관리 서비스입니다.

기타 도구

- [Terraform](#)은 클라우드 및 온프레미스 리소스를 생성하고 관리하는 데 도움이 HashiCorp 되는 코드 형 인프라 (IaC) 도구입니다.

코드 리포지토리

이 패턴에 대한 지침과 코드는 GitHub [중앙 집중식](#) 패키지 배포 리포지토리에서 확인할 수 있습니다.

모범 사례

- 연결에 태그를 할당하려면 [AWS Command Line Interface\(AWS CLI\)](#) 또는 를 사용하십시오 [AWS Tools for PowerShell](#). Systems Manager 콘솔을 사용하여 연결에 태그를 추가하는 것은 지원되지 않습니다. 자세한 내용은 Systems Manager 설명서에서 [Systems Manager 리소스 태깅](#)을 참조하십시오.
- 다른 계정에서 공유한 새 버전의 문서를 사용하여 연결을 실행하려면 문서 버전을 로 default 설정 하십시오.
- 대상 노드에만 태그를 지정하려면 태그 키 하나를 사용하십시오. 여러 태그 키를 사용하여 노드를 타겟팅하려면 리소스 그룹 옵션을 사용하십시오.

에픽

소스 파일 및 계정을 구성합니다.

작업	설명	필요한 기술
리포지토리를 복제합니다.	1. GitHub 중앙 집중식 패키지 배포 리포지토리를 복제하십시오. <pre>git clone https://github.com/aws-samp</pre>	DevOps 엔지니어

작업	설명	필요한 기술
	<pre>les/aws-organization-centralised-package-distribution</pre> <p>2. Terraform 코드 리포지토리에는 AFT에서 관리하는 두 개의 사용자 지정 폴더가 필요합니다. 리포지토리의 로컬 사본에 다음 폴더가 포함되어 있는지 확인하십시오.</p> <pre>\$ cd centralised-package-distribution \$ ls global-customization account-customization</pre>	
<p>글로벌 변수를 업데이트합니다.</p>	<p>global-customization/variables.tf 파일에서 다음 입력 매개변수를 업데이트합니다. 이러한 변수는 AFT에서 생성하고 관리하는 모든 계정에 적용됩니다.</p> <ul style="list-style-type: none"> • <code>account_id</code> : 디스트리뷰터 솔루션이 배포될 계정의 ID. • <code>aws_region</code> : 연결이 배포될 AWS 리전 위치입니다. 	<p>DevOps 엔지니어</p>

작업	설명	필요한 기술
계정 변수 업데이트	<p>account-customization/variables.tf 파일에서 다음 입력 매개변수를 업데이트하십시오. 이러한 변수는 AFT에서 생성하고 관리하는 특정 계정에만 적용됩니다.</p> <ul style="list-style-type: none"> • package_bucket_name : 패키지 배포 파일이 포함된 S3 버킷의 이름. • package_name : 패키지 배포 파일의 이름. • package_version : 설치 프로그램의 패키지 버전. 	DevOps 엔지니어

파라미터 및 배포 파일 사용자 지정

작업	설명	필요한 기술
State Manager 연결을 위한 입력 매개변수를 업데이트하십시오.	<p>account-customization/association.tf 파일에서 다음 입력 파라미터를 업데이트하여 인스턴스에 유지할 상태를 정의하십시오. 사용 사례를 지원하는 경우 기본 파라미터 값을 사용할 수 있습니다.</p> <ul style="list-style-type: none"> • targetAccounts : 배포할 대상 인스턴스의 계정을 나타내는 AWS Organizations 내의 조직 단위 (OU) ID입니다. OU ID는 "ou"로 시작합니다. 	DevOps 엔지니어

작업	설명	필요한 기술
	<ul style="list-style-type: none"> • <code>targetRegions</code> : 대상 인스턴스가 실행 중인 위치 AWS 리전 (예: “us-east-1” 또는 “ap-southeast-2”). • <code>action</code>: 패키지 설치 또는 제거 여부를 지정합니다. • <code>installationType</code> : 다음 설치 유형 중 하나: <ul style="list-style-type: none"> • <code>uninstall</code> : 패키지가 제거되었습니다. • <code>reinstall</code> : 재설치 프로세스가 완료될 때까지 애플리케이션이 오프라인 상태로 전환됩니다. • <code>In-place update</code>: 새 파일이나 업데이트된 파일이 설치에 추가되는 동안 응용 프로그램을 사용할 수 있습니다. • <code>name</code>: 설치 또는 제거할 패키지 이름. • <code>version</code>: 설치 또는 제거할 패키지 버전. 패키지 버전이 설치되지 않은 경우 시스템에서 오류를 반환합니다. • <code>bucketName</code> : 패키지가 배포된 S3 버킷 이름. 이 버킷은 패키지와 매니페스트 파일로만 구성되어야 합니다. • <code>bucketPrefix</code> : 패키지 자산이 저장되는 S3 접두사. • <code>AutomationAssumeRole</code> : 의 아마존 리소스 이 	

작업	설명	필요한 기술
	<p>름 (ARN). SystemsManager-AutomationAdministrationRole</p>	
<p>압축 파일과 패키지용 manifest.json 파일을 준비합니다.</p>	<p>이 패턴은 폴더에 PowerShell 설치 및 제거 스크립트가 포함된 설치 가능한 샘플 파일 (Windows의 경우 .msi, Linux의 경우 .rpm) 을 제공합니다. account-customization/package</p> <ol style="list-style-type: none"> 1. PowerShell 설치 가능한 파일을 자체 파일로 바꾸거나 설치 파일, 설치 및 제거 스크립트, 매니페스트 파일을 제공하여 계정의 폴더에 패키지를 생성하십시오. account-customization 2. 요구 사항에 따라 Terraform 이 폴더에 생성하는 기본 manifest.json 파일을 사용자 정의하십시오. account-customization . 	<p>DevOps 엔지니어</p>

Terraform 명령을 실행하여 리소스를 프로비저닝합니다.

작업	설명	필요한 기술
<p>Terraform 구성을 초기화합니다.</p>	<p>AFT를 사용하여 솔루션을 자동으로 배포하려면 코드를 다음 위치로 푸시하십시오. AWS CodeCommit</p>	<p>DevOps 엔지니어</p>

작업	설명	필요한 기술
	<pre>\$ git add *</pre> <pre>\$ git commit -m</pre> <pre> "message"</pre> <pre>\$ git push</pre> <p>폴더에서 Terraform 명령을 실행하여 AFT를 사용하지 않고 이 솔루션을 배포할 수도 있습니다. <code>account-customization</code> Terraform 파일이 포함된 작업 디렉토리를 초기화하려면 다음을 실행하십시오.</p> <pre>\$ terraform init</pre>	
변경 사항 미리 보기	<p>Terraform이 인프라에 적용할 변경 사항을 미리 보려면 다음 명령을 실행하십시오.</p> <pre>\$ terraform plan</pre> <p>이 명령은 Terraform 구성을 평가하여 선언된 리소스의 원하는 상태를 확인합니다. 또한 원하는 상태를 작업 공간 내에서 프로비저닝할 실제 인프라와 비교합니다.</p>	DevOps 엔지니어
변경 사항 적용.	<p>다음 명령을 실행하여 <code>variables.tf</code> 파일에 적용한 변경 사항을 구현합니다.</p> <pre>\$ terraform apply</pre>	DevOps 엔지니어

리소스 검증

작업	설명	필요한 기술
SSM 문서 생성을 검증합니다.	<ol style="list-style-type: none"> Systems Manager 콘솔의 왼쪽 탐색 창에서 [문서] 를 선택합니다. 내 소유(Owned by me) 탭을 선택합니다. <p>DistributeSoftware Package 및 AddSoftwarePackageToDistributor 패키지가 표시되어야 합니다.</p>	DevOps 엔지니어
자동화가 성공적으로 배포되었는지 검증하십시오.	<ol style="list-style-type: none"> Systems Manager 콘솔의 왼쪽 탐색 창에서 자동화를 선택합니다. 자동화 실행 목록에서 가장 최근 DistributeSoftwarePackage 및 AddSoftwarePackageToDistributor 배포를 확인할 수 있습니다. 실행 ID를 선택하여 성공적으로 완료되었는지 확인합니다. 	DevOps 엔지니어
패키지가 대상 멤버 계정 인스턴스에 배포되었는지 확인합니다.	<ol style="list-style-type: none"> Systems Manager 콘솔의 탐색 창에서 명령 실행을 선택합니다. 명령 기록에서 각 호출 및 상태를 확인할 수 있습니다. 	DevOps 엔지니어

작업	설명	필요한 기술
	<p>3. 명령 ID를 선택하면 각 대상 인스턴스의 배포 기록을 볼 수 있습니다.</p> <p>4. 인스턴스 ID를 선택하고 Output 섹션에서 배포를 확인합니다.</p>	

문제 해결

문제	Solution
State Manager 연결이 실패했거나 보류 중 상태로 멈췄습니다.	AWS 지식 센터의 문제 해결 정보 를 참조하십시오.
예약된 연결을 실행하지 못했습니다.	스케줄 지정이 유효하지 않을 수 있습니다. State Manager는 현재 연관성을 위해 크론 표현식으로 월을 지정하는 것을 지원하지 않습니다. cron 또는 rate 표현식 을 사용하여 일정을 확인하세요.

관련 리소스

- [중앙 집중식 패키지 배포](#) (GitHub 리포지토리)
- [테라폼용 어카운트 팩토리 \(AFT\)](#)
- [사용 사례 및 모범 사례 \(설명서\)](#) AWS Systems Manager

AWS 계정 간 중앙 집중화를 위한 VPC 흐름 로그 구성

작성자: Benjamin Morris(AWS) 및 Aman Kaur Gandhi(AWS)

환경: 프로덕션

기술: 관리 및 거버넌스

AWS 서비스: Amazon VPC,
Amazon S3

요약

Amazon Web Services(AWS) Virtual Private Cloud(VPC)에서 VPC 흐름 로그 기능은 운영 및 보안 문제 해결에 유용한 데이터를 제공할 수 있습니다. 하지만 다중 계정 환경에서 VPC 흐름 로그를 사용하는 데에는 제한이 있습니다. 특히 Amazon Logs의 계정 간 흐름 CloudWatch 로그는 지원되지 않습니다. 대신 적절한 버킷 정책으로 Amazon Simple Storage Service(S3) 버킷을 구성하여 로그를 중앙 집중화할 수 있습니다.

참고: 이 패턴은 흐름 로그를 중앙 위치로 전송하기 위한 요구 사항을 설명합니다. 하지만 멤버 계정에서도 로그를 로컬로 사용할 수 있도록 하려면 각 VPC에 대해 여러 흐름 로그를 만들 수 있습니다. Log Archive 계정에 액세스할 수 없는 사용자는 문제 해결을 위해 트래픽 로그를 볼 수 있습니다. 또는 로그를 Logs로 보내는 각 VPC에 대해 단일 흐름 로그를 구성할 수 있습니다. CloudWatch 그런 다음 Amazon Data Firehose 구독 필터를 사용하여 로그를 S3 버킷으로 전달할 수 있습니다. 자세한 내용은 [관련 리소스](#) 섹션을 참조하세요.

사전 조건 및 제한 사항

사전 조건

- 활성 상태의 AWS 계정.
- 로그를 중앙 집중화하는 데 사용되는 계정을 보유한 AWS Organizations 조직(예: 로그 아카이브)

제한 사항

AWS Key Management Service(AWS KMS) 관리형 키 `aws/s3`를 사용하여 중앙 버킷을 암호화하는 경우 다른 계정으로부터 로그를 수신하지 않습니다. 대신 다음과 같은 오류가 표시됩니다.

```
"Unsuccessful": [
  {
    "Error": {
```

```

        "Code": "400",
        "Message": "LogDestination: <bucketName> is undeliverable"
    },
    "ResourceId": "vpc-1234567890123456"
}
]

```

계정의 AWS 관리형 키를 계정 간에 공유할 수 없기 때문입니다.

해결 방법은 Amazon S3 관리형 암호화(SSE-S3) 또는 멤버 계정과 공유할 수 있는 AWS KMS 고객 관리형 키를 사용하는 것입니다.

아키텍처

대상 기술 스택

다음 다이어그램에서는 각 VPC에 대해 두 개의 흐름 로그가 배포되어 있습니다. 하나는 로컬 로그 그룹에 CloudWatch 로그를 전송합니다. 다른 하나는 중앙 로깅 계정의 S3 버킷으로 로그를 전송합니다. 버킷 정책은 로그 전송 서비스가 버킷에 로그를 기록할 수 있도록 허용합니다.

중요: 이 솔루션에 필요한 버킷 정책과 관련된 위험을 이해해야 합니다. 이 버킷에 쓰는 주체는 AWS Identity and Access Management(IAM) 주체가 아니라 서비스 주체이므로 `aws:PrincipalOrgID` 조건은 유효한 조건이 아닙니다. 즉, 현재로서는 계정의 상위 조직에 따라 쓰기를 제한할 방법이 없습니다.

버킷을 보호하려면 hard-to-guess 버킷 이름을 사용하고, 버킷 이름은 조직 외부에 노출되어서는 안 되는 민감한 값으로 취급해야 합니다. 버킷 정책에서 최소 권한 권한을 사용하고, `s3:putObject` 및 `s3:GetBucketAcl` 이상의 권한은 부여하지 않습니다. 정적 계정 집합이 있는 환경에서 작업하는 경우 거부 효과를 사용하여 특정 계정을 제외한 액세스를 차단할 수 있지만 대부분의 조직에서는 운영상 불가능합니다.

대상 아키텍처

자동화 및 규모 조정

각 VPC는 중앙 로깅 계정의 S3 버킷으로 로그를 전송하도록 구성됩니다. 다음 자동화 솔루션 중 하나를 사용하면 흐름 로그가 적절하게 구성되도록 할 수 있습니다.

- [AWS CloudFormation StackSets](#)
- [AWS Control Tower Account Factory for Terraform\(AFT\)](#)

- [수정 사항이 포함된 AWS Config 규칙](#)

도구

도구

- [Amazon CloudWatch Logs](#)를 사용하면 모든 시스템, 애플리케이션 및 AWS 서비스의 로그를 중앙 집중화하여 로그를 모니터링하고 안전하게 보관할 수 있습니다.
- [Amazon Simple Storage Service\(S3\)](#)는 원하는 양의 데이터를 저장, 보호 및 검색하는 데 도움이 되는 클라우드 기반 객체 스토리지 서비스입니다.
- [Amazon Virtual Private Cloud\(VPC\)](#)를 이용하면 사용자가 정의한 가상 네트워크로 AWS 리소스를 시작할 수 있습니다. 이 가상 네트워크는 사용자의 자체 데이터 센터에서 운영하는 기존 네트워크와 유사하며 AWS의 확장 가능한 인프라를 사용한다는 이점이 있습니다. 이 패턴은 [VPC 플로우 로그](#) 기능으로 VPC의 네트워크 인터페이스에서 송수신되는 IP 트래픽에 대한 정보를 수집합니다.

모범 사례

코드형 인프라(IaC)를 사용하면 VPC 흐름 로그 배포 프로세스를 크게 단순화할 수 있습니다. 흐름 로그 리소스 구성을 포함하도록 VPC 배포 정의를 추상화하면 흐름 로그와 함께 VPC가 자동으로 배포됩니다. 이 내용은 다음 섹션에서 설명합니다.

중앙 집중식 흐름 로그

Terraform의 VPC 모듈에 중앙 집중식 흐름 로그를 추가하기 위한 예제 구문 HashiCorp

이 코드는 VPC에서 중앙 집중식 S3 버킷으로 로그를 보내는 흐름 로그를 생성합니다. 참고로 이 패턴에는 S3 버킷 생성이 포함되지 않습니다.

권장 버킷 정책 설명은 [추가 정보](#) 섹션을 참조하세요.

```
variable "vpc_id" {
  type          = string
  description = "ID of the VPC for which you want to create a Flow Log"
}

locals {
  # For more details: https://docs.aws.amazon.com/vpc/latest/userguide/flow-logs.html#flow-logs-custom
  custom_log_format_v5 = "${version} ${account-id} ${interface-id} ${srcaddr} ${dstaddr} ${srcport} ${dstport} ${protocol} ${packets} ${bytes} ${start} ${end}"
}
```

```

    ${action} ${log-status} ${vpc-id} ${subnet-id} ${instance-id} ${tcp-flags} $
    ${type} ${pkt-srcaddr} ${pkt-dstaddr} ${region} ${az-id} ${sublocation-type} $
    ${sublocation-id} ${pkt-src-aws-service} ${pkt-dst-aws-service} ${flow-direction} $
    ${traffic-path}"
  }

resource "aws_flow_log" "centralized" {
  log_destination      = "arn:aws:s3:::centralized-vpc-flow-logs-
<log_archive_account_id>" # Optionally, a prefix can be added after the ARN.
  log_destination_type = "s3"
  traffic_type         = "ALL"
  vpc_id               = var.vpc_id
  log_format           = local.custom_log_format_v5 # If you want fields from VPC Flow
  Logs v3+, you will need to create a custom log format.
  tags                 = {
    Name = "centralized_flow_log"
  }
}

```

로컬 플로우 로그

필요한 권한을 사용하여 Terraform의 VPC 모듈에 로컬 흐름 로그를 추가하는 예제 구문

이 코드는 VPC에서 로컬 Logs 그룹으로 로그를 보내는 흐름 CloudWatch 로그를 생성합니다.

```

data "aws_region" "current" {}

variable "vpc_id" {
  type          = string
  description = "ID of the VPC for which you want to create a Flow Log"
}

resource "aws_iam_role" "local_flow_log_role" {
  name = "flow-logs-policy-${var.vpc_id
}"

  assume_role_policy = <<EOF
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "",
      "Effect": "Allow",
      "Principal": {

```

```

        "Service": "vpc-flow-logs.amazonaws.com"
    },
    "Action": "sts:AssumeRole"
}
]
}
EOF
}

resource "aws_iam_role_policy" "logs_permissions" {
    name = "flow-logs-policy-${var.vpc_id}"
    role = aws_iam_role.local_flow_log_role.id

    policy = <<EOF
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Action": [
                "logs:CreateLogGroup",
                "logs:CreateLogStream",
                "logs:PutLogEvents",
                "logs:DescribeLogGroups",
                "logs:DescribeLogStreams",
                "logs:CreateLogDelivery",
                "logs>DeleteLogDelivery"
            ],
            "Effect": "Allow",
            "Resource": "arn:aws:logs:${data.aws_region.current.name}:*:log-group:vpc-flow-logs*"
        }
    ]
}
EOF
}

resource "aws_cloudwatch_log_group" "local_flow_logs" {
    # checkov:skip=CKV_AWS_338:local retention is set to 30, centralized S3 bucket can
    # retain for long-term
    name = "vpc-flow-logs/${var.vpc_id}"
    retention_in_days = 30
}

resource "aws_flow_log" "local" {

```

```

iam_role_arn    = aws_iam_role.local_flow_log_role.arn
log_destination = aws_cloudwatch_log_group.local_flow_logs.arn
traffic_type    = "ALL"
vpc_id         = var.vpc_id
tags           = {
  Name = "local_flow_log"
}
}

```

에픽

VPC 흐름 로그 인프라 배포

작업	설명	필요한 기술
암호화 전략을 결정하고 중앙 S3 버킷에 대한 정책을 생성합니다.	중앙 버킷은 aws/s3 AWS KMS 키를 지원하지 않으므로 SSE-S3 또는 AWS KMS 고객 관리형 키를 사용해야 합니다. AWS KMS 키를 사용하는 경우 멤버 계정에서 키를 사용할 수 있도록 키 정책에서 허용해야 합니다.	규정 준수
중앙 흐름 로그 버킷을 생성합니다.	흐름 로그를 전송할 중앙 버킷을 만들고 이전 단계에서 선택한 암호화 전략을 적용합니다. 이는 Log Archive 또는 이와 유사한 용도의 계정에 있어야 합니다. 추가 정보 섹션에서 버킷 정책을 가져와 환경별 값으로 플레이스홀더를 업데이트한 후 중앙 버킷에 적용합니다.	일반 AWS
로그를 중앙 흐름 로그 버킷으로 보내도록 VPC 흐름 로그를 구성합니다.	데이터를 수집하려는 각 VPC에 흐름 로그를 추가합니다. 이를 위한 가장 자 조절할 수 있는	네트워크 관리자

작업	설명	필요한 기술
	<p>방법은 AFT 또는 AWS Cloud Development Kit(AWS CDK)와 같은 IaC 도구를 사용하는 것입니다. 예를 들어 흐름 로그와 함께 VPC를 배포하는 Terraform 모듈을 만들 수 있습니다. 필요한 경우 흐름 로그를 수동으로 추가합니다.</p>	
<p>로컬 CloudWatch 로그로 전송하도록 VPC 흐름 로그를 구성합니다.</p>	<p>(선택 사항) 로그가 생성되는 계정에서 흐름 로그를 표시하려면 다른 흐름 로그를 만들어 로컬 계정의 CloudWatch 로그로 데이터를 보내십시오. 또는 로컬 계정의 계정별 S3 버킷으로 데이터를 보낼 수도 있습니다.</p>	<p>일반 AWS</p>

관련 리소스

- [중앙 집중식 흐름 로그 데이터를 사용하여 데이터 분석을 촉진하고 보안 요구 사항을 충족하는 방법\(블로그 게시물\)](#)
- [AWS Config 규칙을 사용하여 VPC 흐름 로그를 자동으로 활성화하는 방법\(블로그 게시물\)](#)

추가 정보

버킷 정책

이 버킷 정책 예제는 자리 표시자 이름에 값을 추가한 후 흐름 로그용 중앙 S3 버킷에 적용할 수 있습니다.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
```

```

    "Sid": "AWSLogDeliveryWrite",
    "Effect": "Allow",
    "Principal": {
      "Service": "delivery.logs.amazonaws.com"
    },
    "Action": "s3:PutObject",
    "Resource": "arn:aws:s3:::<BUCKET_NAME>/*",
    "Condition": {
      "StringEquals": {
        "s3:x-amz-acl": "bucket-owner-full-control"
      }
    }
  },
  {
    "Sid": "AWSLogDeliveryCheck",
    "Effect": "Allow",
    "Principal": {
      "Service": "delivery.logs.amazonaws.com"
    },
    "Action": "s3:GetBucketAcl",
    "Resource": "arn:aws:s3:::<BUCKET_NAME>"
  },
  {
    "Sid": "DenyUnencryptedTraffic",
    "Effect": "Deny",
    "Principal": {
      "AWS": "*"
    },
    "Action": "s3:*",
    "Resource": [
      "arn:aws:s3:::<BUCKET_NAME>/*",
      "arn:aws:s3:::<BUCKET_NAME>"
    ],
    "Condition": {
      "Bool": {
        "aws:SecureTransport": "false"
      }
    }
  }
]
}

```

정적 계정 목록이 있는 경우 다음 명령문을 추가하여 해당 목록에 없는 계정을 거부할 수 있습니다.

```
{
  "Sid": "AccountDenyList",
  "Effect": "Deny",
  "Principal": "*",
  "Action": "s3:PutObject",
  "NotResource": [
    "arn:aws:s3:::<BUCKET_NAME>/<OPTIONAL_PREFIX>/AWSLogs/<ACCOUNT_ID1>/*",
    "arn:aws:s3:::<BUCKET_NAME>/<OPTIONAL_PREFIX>/AWSLogs/<ACCOUNT_ID2>/*",
    "arn:aws:s3:::<BUCKET_NAME>/<OPTIONAL_PREFIX>/AWSLogs/<ACCOUNT_ID3>/*",
  ]
}
```

이전 NotResource Deny 패턴 대신 각 Allow 명세서에 조건을 추가하여 승인된 계정을 지정할 수 있습니다.

```
"Condition": {
  "StringEquals": {
    "aws:SourceAccount": [
      "111111111111",
      "222222222222"
    ]
  }
}
```

접두사 추가

또한 버킷 이름이 공개적으로 노출되는 시나리오에서 버킷에 대한 원치 않는 외부 쓰기가 우려되는 경우 버킷 내 알려진 접두사로 쓰기를 제한할 수 있습니다. 이를 구현하는 경우 버킷 Amazon 리소스 이름(ARN) 뒤에 접두사를 포함하도록 `aws_flow_log` 리소스의 `log_destination`를 업데이트합니다. 예를 들어 다음 명령문은 쓰기를 특정 접두사로 제한합니다.

```
{
  "Sid": "PrefixAllowList",
  "Effect": "Deny",
  "Principal": "*",
  "Action": "s3:PutObject",
  "NotResource": [
    "arn:aws:s3:::<BUCKET_NAME>/<PREFIX>/*"
  ]
}
```

CloudWatch NLog를 사용하여 Amazon Logs에서.NET 애플리케이션에 대한 로깅을 구성합니다.

작성자: Bibhuti Sahu(AWS) 및 Rob Hill(AWS)(AWS)

환경: 프로덕션

기술: 관리 및 거버넌스
DevOps, 웹 및 모바일 앱

워크로드: Microsoft

AWS 서비스: 아마존
CloudWatch 로그

요약

[이 패턴은 NLog 오픈 소스 로깅 프레임워크를 사용하여 Amazon Logs에.NET 애플리케이션 사용 및 이벤트를 기록하는 방법을 설명합니다.](#) [CloudWatch](#) CloudWatch 콘솔에서 애플리케이션의 로그 메시지를 거의 실시간으로 볼 수 있습니다. 또한 [지표](#)를 설정하고 지표 임계값이 초과될 경우 알리도록 [경보](#)를 구성할 수 있습니다. CloudWatch Application Insights를 사용하면 모니터링되는 애플리케이션의 잠재적 문제를 보여주는 자동화된 대시보드 또는 사용자 지정 대시보드를 볼 수 있습니다. CloudWatch Application Insights는 애플리케이션 및 인프라와 관련된 지속적인 문제를 신속하게 격리할 수 있도록 설계되었습니다.

CloudWatch Logs에 로그 메시지를 쓰려면 AWS.Logger.NLog NuGet 패키지를 .NET 프로젝트에 추가합니다. 그런 다음 CloudWatch 로그를 대상으로 사용하도록 NLog.config 파일을 업데이트합니다.

사전 조건 및 제한 사항

사전 조건

- 활성 상태의 AWS 계정.
- 다음을 수행하는.NET 웹 또는 콘솔 애플리케이션
 - 지원되는 .NET 프레임워크 또는 .NET Core 버전 사용. 자세한 내용은 제품 버전을 참조하세요.
 - NLogs로 로그 데이터를 Application Insights로 보내기.
- AWS 서비스에 대한 IAM 역할 생성 권한. 자세한 내용은 [서비스 역할 권한](#)을 참조하세요.

- AWS 서비스에 역할 전달 권한. 자세한 내용은 [사용자에게 AWS 서비스 역할을 전달할 수 있는 권한 부여](#)를 참조하세요.

제품 버전

- .NET Framework 버전 3.5 이상
- .NET Core 버전 1.0.1, 2.0.0 이상

아키텍처

대상 기술 스택

- NLog
- 아마존 CloudWatch 로그

대상 아키텍처

1. .NET 애플리케이션은 NLog 로깅 프레임워크에 로그 데이터를 작성합니다.
2. NLog는 로그 데이터를 로그에 기록합니다. CloudWatch
3. CloudWatch 경보 및 사용자 지정 대시보드를 사용하여 .NET 애플리케이션을 모니터링합니다.

도구

서비스

- [Amazon CloudWatch Application Insights](#)는 애플리케이션 및 기본 AWS 리소스의 상태를 관찰하는데 도움이 됩니다.
- [Amazon CloudWatch Logs](#)를 사용하면 모든 시스템, 애플리케이션 및 AWS 서비스의 로그를 중앙 집중화하여 로그를 모니터링하고 안전하게 보관할 수 있습니다.
- [AWS Identity and Access Management\(IAM\)](#)를 사용하면 사용자에 대해 인증 및 권한 부여를 제어함으로써 AWS 리소스에 대한 액세스를 안전하게 관리할 수 있습니다.
- [AWS Tools for PowerShell](#) 는 PowerShell 명령줄에서 AWS 리소스의 작업을 스크립팅하는 데 도움이 되는 PowerShell 모듈 세트입니다.

기타 도구

- [Logger.nLog](#)는 로그에 로그 데이터를 기록하는 NLog 대상입니다. CloudWatch
- [NLog](#)는 데이터베이스, 로그 파일 또는 콘솔과 같은 대상에 로그 데이터를 쓰는 데 도움이 되는 .NET 플랫폼용 오픈 소스 로깅 프레임워크입니다.
- [PowerShell](#) Windows, Linux 및 macOS에서 실행되는 Microsoft 자동화 및 구성 관리 프로그램입니다.
- [Visual Studio](#)는 컴파일러, 코드 완성 도구, 그래픽 디자이너 및 소프트웨어 개발을 지원하는 기타 기능을 포함하는 통합 개발 환경(IDE)입니다.

모범 사례

- 대상 로그 그룹에 대한 [보존 정책](#)을 설정합니다. 이 작업은 NLog 구성 외부에서 수행해야 합니다. 기본적으로 로그 데이터는 CloudWatch Logs에 무기한 저장됩니다.
- [AWS; 액세스 키 관리를 위한 모범 사례](#)를 준수합니다.

에픽

액세스 및 도구 설정

작업	설명	필요한 기술
IAM 정책을 생성합니다.	<p>IAM 설명서의 JSON 편집기를 사용하여 정책 생성 지침을 따릅니다. 로그가 CloudWatch 로그를 읽고 쓸 수 있도록 허용하는 데 필요한 최소 권한이 있는 다음 JSON 정책을 입력합니다.</p> <pre> { "Version": "2012-10-17", "Statement": [{ "Effect": "Allow", </pre>	AWS 관리자, AWS DevOps

작업	설명	필요한 기술
	<pre> "Action": ["logs:CreateLogGroup", "logs:CreateLogStream", "logs:GetLogEvents", "logs:PutLogEvents", "logs:DescribeLogGroups", "logs:DescribeLogStreams", "logs:PutRetentionPolicy"], "Resource": ["*"] } </pre>	
IAM 역할을 생성합니다.	IAM 설명서의 AWS 서비스에 권한을 위임할 역할 생성 지침을 따릅니다. 이전에 생성한 정책을 선택합니다. 이는 로깅 작업을 수행하는 CloudWatch Logs의 역할입니다.	AWS 관리자, AWS DevOps

작업	설명	필요한 기술
에 대한 AWS 도구를 설정합니다 PowerShell.	<ol style="list-style-type: none"> AWS 도구 설치의 운영 체제별 지침을 따르십시오 PowerShell. PowerShell cmdlet용 AWS Tools를 사용하여 액세스 키와 비밀 키를 프로필에 저장합니다. 지침은 AWS Tools에서의 프로필 관리 PowerShell 설명서를 참조하십시오. 	일반 AWS

NLog 구성

작업	설명	필요한 기술
NuGet 패키지를 설치합니다.	<ol style="list-style-type: none"> Visual Studio에서 파일을 선택한 다음 프로젝트 또는 솔루션 열기를 선택합니다. NLogs를 설치할 프로젝트를 선택합니다. Visual Studio에서 도구, NuGet 패키지 관리자, 패키지 관리자 콘솔을 선택합니다. 다음 명령을 입력하여 AWS.Logger.NLog NuGet 패키지를 설치합니다. <div style="border: 1px solid #ccc; border-radius: 10px; padding: 10px; margin-top: 10px;"> <pre>Install-Package AWS.Logger.NLog - Version 3.1.0</pre> </div>	앱 개발자

작업	설명	필요한 기술
로깅 대상을 구성합니다.	<ol style="list-style-type: none"> 1. NLog.config 파일을 엽니다. 2. 대상 type에 AWSTarget을 입력합니다. 3. 대상 logGroup에 사용할 로그 그룹 이름을 입력합니다. 로그 그룹이 아직 존재하지 않는 경우 제공된 이름을 가진 새 로그 그룹이 자동으로 생성됩니다. 4. 대상의 region 경우 CloudWatch 로그가 구성된 AWS 지역을 입력합니다. 5. 대상 profile에 액세스 키와 보안 암호 키를 저장하기 위해 이전에 생성한 프로필의 이름을 입력합니다. 6. NLog.config 파일을 저장하고 닫습니다. <p>샘플 구성 파일은 이 패턴의 추가 정보 섹션을 참조하세요. 애플리케이션을 실행하면 NLog가 로그 메시지를 작성하여 CloudWatch Logs로 전송합니다.</p>	앱 개발자

로그 검증 및 모니터링

작업	설명	필요한 기술
로깅 검증.	로그 설명서에서 로그로 CloudWatch 전송된 로그 데이	일반 AWS

작업	설명	필요한 기술
	<p>터 보기의 CloudWatch 지침을 따르세요. .NET 애플리케이션에 대한 로그 이벤트가 기록되고 있는지 확인합니다. 로그 이벤트가 기록되지 않는 경우 이 패턴의 문제 해결 섹션을 참조하세요.</p>	
<p>.NET 애플리케이션 스택을 모니터링합니다.</p>	<p>사용 사례에 따라 CloudWatch 필요에 따라 모니터링을 구성하십시오. CloudWatch 로그 인사이트, CloudWatch 메트릭 인사이트, CloudWatch 애플리케이션 인사이트를 사용하여 .NET 워크로드를 모니터링할 수 있습니다. 알림을 받을 수 있도록 경보를 구성하고 단일 보기에서 워크로드를 모니터링하기 위한 사용자 지정 대시보드를 만들 수도 있습니다.</p>	<p>일반 AWS</p>

문제 해결

문제	Solution
<p>로그 데이터는 CloudWatch 로그에 표시되지 않습니다.</p>	<p>CloudWatch Logs가 맡는 IAM 역할에 IAM 정책이 연결되어 있는지 확인하십시오. 지침은 에픽 섹션의 액세스 및 도구 설정 섹션을 참조하세요.</p>

관련 리소스

- [로그 그룹 및 로그 스트림 사용 \(로그 설명서\)](#) CloudWatch
- [Amazon CloudWatch 로그 및 .NET 로깅 프레임워크](#) (AWS 블로그 게시물)

추가 정보

다음은 샘플 NLog.config 파일입니다.

```
<?xml version="1.0" encoding="utf-8" ?>
<configuration>
  <configSections>
    <section name="nlog" type="NLog.Config.ConfigSectionHandler, NLog" />
  </configSections>
  <startup>
    <supportedRuntime version="v4.0" sku=".NETFramework,Version=v4.7.2" />
  </startup>
  <nlog>
    <extensions>
      <add assembly="NLog.AWS.Logger" />
    </extensions>
    <targets>
      <target name="aws" type="AWSTarget" logGroup="NLog.TestGroup" region="us-east-1"
profile="demo"/>
    </targets>
    <rules>
      <logger name="*" minlevel="Info" writeTo="aws" />
    </rules>
  </nlog>
</configuration>
```

여러 AWS 계정 및 AWS 리전에 걸쳐 AWS Service Catalog 제품 복사

작성자: Sachin Vighe(AWS), Santosh Kale(AWS)

환경: 프로덕션

기술: 관리 및 거버넌스, 서버리스

워크로드: 기타 모든 워크로드

AWS 서비스: AWS Service Catalog, AWS Lambda

요약

AWS Service Catalog는 지역 서비스이므로 AWS Service Catalog 포트폴리오 및 제품은 [포트폴리오와 제품](#)이 생성된 AWS 리전에서만 볼 수 있습니다. 새 지역에 [AWS Service Catalog 허브](#)를 설정하는 경우 기존 제품을 다시 생성해야 하며 이 프로세스에는 시간이 많이 걸릴 수 있습니다.

이 패턴의 접근 방식은 소스 AWS 계정 또는 리전의 AWS Service Catalog 허브에서 대상 계정 또는 리전의 새 허브로 제품을 복사하는 방법을 설명함으로써 이 프로세스를 간소화하는 데 도움이 됩니다. AWS Service Catalog 허브 및 스포크 모델에 대한 자세한 내용은 AWS 관리 및 거버넌스 블로그에서 [AWS Service Catalog 허브 및 스포크 모델: AWS Service Catalog를 여러 계정에 자동으로 배포 및 관리하는 방법](#)을 참조하세요.

또한 이 패턴은 AWS Service Catalog 제품을 계정 간 또는 다른 리전으로 복사하는 데 필요한 별도의 코드 패키지를 제공합니다. 조직은 이 패턴을 사용하여 시간을 절약하고, 기존 및 이전 제품 버전을 새 AWS Service Catalog 허브에서 사용할 수 있게 하고, 수동 오류의 위험을 최소화하고, 여러 계정 또는 리전으로 접근 방식을 확장할 수 있습니다.

참고: 이 패턴의 에픽 섹션은 제품 복사를 위한 두 가지 옵션을 제공합니다. 옵션 1을 사용하여 계정 간에 제품을 복사하거나 옵션 2를 선택하여 지역 간에 제품을 복사할 수 있습니다.

사전 조건 및 제한 사항

사전 조건

- 활성 상태의 AWS 계정.
- 소스 계정 또는 리전의 기존 AWS Service Catalog 제품

- 대상 계정 또는 리전의 기존 AWS Service Catalog 허브.
- 계정 간에 제품을 복사하려면 제품이 포함된 AWS Service Catalog 포트폴리오를 공유한 다음 대상 계정으로 가져와야 합니다. 이에 대한 자세한 내용은 AWS Service Catalog 설명서의 [포트폴리오 공유 및 가져오기](#)를 참조하세요.

제한 사항

- 여러 지역 또는 계정에 복사하려는 AWS Service Catalog 제품은 둘 이상의 포트폴리오에 속할 수 없습니다.

아키텍처

다음 다이어그램은 원본 계정에서 대상 계정으로 AWS Service Catalog 제품을 복사하는 것을 보여줍니다.

다음 다이어그램은 원본 지역에서 대상 리전으로 AWS Service Catalog 제품을 복사하는 것을 보여줍니다.

기술 스택

- 아마존 CloudWatch
- AWS Identity and Access Management(IAM)
- AWS Lambda
- AWS Service Catalog

자동화 및 규모 조정

수신된 요청 수 또는 복사해야 하는 AWS Service Catalog 제품 수에 따라 확장할 수 있는 Lambda 함수를 사용하여 이 패턴의 접근 방식을 확장할 수 있습니다. 이에 대한 자세한 내용은 AWS Lambda 설명서의 [Lambda 함수 규모 조정](#)을 참조하세요.

도구

- [AWS Command Line Interface\(AWS CLI\)](#)는 명령줄 쉘에서 명령을 사용하여 AWS 서비스와 상호 작용할 수 있는 오픈 소스 도구입니다.

- [AWS Identity and Access Management\(IAM\)](#)는 사용자에게 대한 인증 및 권한 부여를 제어함으로써 AWS 리소스에 대한 액세스를 안전하게 관리할 수 있습니다.
- [AWS Lambda](#)는 서버를 프로비저닝하거나 관리할 필요 없이 코드를 실행하는 데 도움이 되는 컴퓨팅 서비스입니다. 필요할 때만 코드를 실행하며 자동으로 확장이 가능하므로 사용한 컴퓨팅 시간만큼만 비용을 지불합니다.
- [AWS Service Catalog](#)를 사용하면 AWS에 승인된 IT 서비스의 카탈로그를 중앙에서 관리할 수 있습니다. 최종 사용자는 조직에서 규정한 제약에 따라, 필요에 따라 승인된 IT 서비스만 신속하게 배포할 수 있습니다.

코드

cross-account-copy 패키지(첨부)를 사용하여 계정 간에 AWS Service Catalog 제품을 복사하거나 cross-region-copy 패키지(첨부)를 사용하여 리전 간에 제품을 복사할 수 있습니다.

cross-account-copy 패키지에는 다음 파일이 포함되어 있습니다.

- copyconf.properties - 계정 간에 제품을 복사하기 위한 지역 및 AWS 계정 ID 파라미터가 포함된 구성 파일입니다.
- scProductCopyLambda.py - 계정 간에 제품을 복사하는 Python 함수입니다.
- createDestAccountRole.sh - 대상 계정에서 IAM 역할을 생성하기 위한 스크립트입니다.
- createSrcAccountRole.sh - 소스 계정에서 IAM 역할을 생성하기 위한 스크립트입니다.
- copyProduct.sh - Lambda 함수를 생성하고 호출하여 계정 간에 제품을 복사하는 스크립트입니다.

cross-region-copy 패키지에는 다음 파일이 포함되어 있습니다.

- copyconf.properties - 리전 간 제품 복사를 위한 리전 및 AWS 계정 ID 파라미터가 포함된 구성 파일입니다.
- scProductCopyLambda.py - 지역 간에 제품을 복사하기 위한 Python 함수입니다.
- copyProduct.sh - IAM 역할을 생성하고 Lambda 함수를 생성 및 호출하여 여러 리전에 제품을 복사하는 스크립트입니다.

에픽

옵션 1 - 계정 전체에 AWS Service Catalog 제품 복사

작업	설명	필요한 기술
성 파일을 업데이트합니다.	<ol style="list-style-type: none"> 1. 로컬 머신에 첨부된 <code>cross-account-copy</code> 패키지를 다운로드합니다. 2. <code>copyconf.properties</code> 구성 파일을 다음 값으로 업데이트합니다. <ul style="list-style-type: none"> • <code>srcRegion</code> - 제품이 포함된 소스 리전을 입력합니다. • <code>destRegion</code> - 제품의 대상 리전을 입력합니다. • <code>sourceAccountId</code> - 소스 계정의 AWS 계정 ID를 제공합니다. • <code>destAccountId</code> - 대상 계정의 AWS 계정 ID를 제공합니다. 	AWS 관리자, AWS 시스템 관리자, 클라우드 관리자
대상 계정에서 AWS CLI용 보안 인증을 구성합니다.	<p><code>aws configure</code> 명령을 실행하고 다음 값을 제공하여 대상 계정의 AWS CLI에 액세스하도록 보안 인증 정보를 구성합니다.</p> <pre>\$aws configure AWS Access Key ID [None]: <your_access_key_id> AWS Secret Access Key [None]: <your_secret_access_key></pre>	AWS 관리자, AWS 시스템 관리자, 클라우드 관리자

작업	설명	필요한 기술
<p>원본 계정에서 AWS CLI용 보안 인증 정보를 구성합니다.</p>	<pre data-bbox="597 205 1019 384">Default region name [None]: Region Default output format [None]:</pre> <p data-bbox="597 422 1019 554">이에 대한 자세한 내용은 AWS 명령줄 인터페이스 설명서의 구성 기본을 참조하세요.</p> <pre data-bbox="597 863 1019 1339">\$aws configure AWS Access Key ID [None]: <your_access_key_id> AWS Secret Access Key [None]: <your_secret_access_key> Default region name [None]: Region Default output format [None]:</pre> <p data-bbox="597 1377 1019 1509">이에 대한 자세한 내용은 AWS 명령줄 인터페이스 설명서의 구성 기본을 참조하세요.</p>	<p>AWS 관리자, AWS 시스템 관리자, 클라우드 관리자</p>

작업	설명	필요한 기술
<p>대상 계정에서 Lambda 실행 역할을 생성합니다.</p>	<p>대상 계정에서 createDestAccountRole.sh 스크립트를 실행합니다. 이 스크립트는 다음 작업을 구현합니다.</p> <ul style="list-style-type: none"> 대상 계정에 Lambda 실행 역할 생성 Lambda 실행 역할을 위한 IAM 정책 생성 및 연결 	<p>AWS 관리자, AWS 시스템 관리자, 클라우드 관리자</p>
<p>소스 계정에서 교차 계정 IAM 역할을 생성합니다.</p>	<p>소스 계정에서 createSrcAccountRole.sh 스크립트를 실행합니다. 이 스크립트는 다음 작업을 구현합니다.</p> <ul style="list-style-type: none"> 대상 계정에서 제품을 복사하기 위해 Lambda 실행 역할을 위임하는 교차 계정 IAM 역할을 소스 계정에 생성합니다 소스 계정의 교차 계정 역할을 위한 IAM 정책 생성 및 연결 	<p>AWS 관리자, AWS 시스템 관리자, 클라우드 관리자</p>
<p>대상 계정에서 copyProduct 스크립트를 실행합니다.</p>	<p>대상 계정에서 copyProduct.sh 스크립트를 실행합니다. 이 스크립트는 다음 작업을 구현합니다.</p> <ul style="list-style-type: none"> Lambda 함수를 생성하고 호출하여 소스 계정에서 대상 계정으로 제품 복사 	<p>AWS 관리자, AWS 시스템 관리자, 클라우드 관리자</p>

옵션 2 - 원본 지역의 AWS Service Catalog 제품을 대상 리전으로 복사

작업	설명	필요한 기술
<p>성 파일을 업데이트합니다.</p>	<ol style="list-style-type: none"> 1. 로컬 머신에 첨부된 cross-region-copy 패키지를 다운로드합니다. 2. copyconf.properties 구성 파일을 다음 값으로 업데이트합니다. <ul style="list-style-type: none"> • srcRegion - 제품이 포함된 소스 리전을 입력합니다. • destRegion - 제품의 대상 리전을 입력합니다. • accountId - AWS 계정 ID를 입력합니다. 	<p>AWS 시스템 관리자, 클라우드 관리자, AWS 관리자</p>
<p>AWS CLI에 대한 보안 인증 정보를 구성합니다.</p>	<p>aws configure 명령을 실행하고 다음 값을 제공하여 소스 계정의 AWS CLI에 액세스하도록 보안 인증 정보를 구성합니다.</p> <pre style="border: 1px solid #ccc; border-radius: 10px; padding: 10px; margin-top: 10px;">\$aws configure AWS Access Key ID [None]: <your_access_key_id> AWS Secret Access Key [None]: <your_secret_access_key> Default region name [None]: Region Default output format [None]:</pre>	<p>AWS 관리자, AWS 시스템 관리자, 클라우드 관리자</p>

작업	설명	필요한 기술
	이에 대한 자세한 내용은 AWS 명령줄 인터페이스 설명서의 구성 기본 을 참조하세요.	
copyProduct 스크립트를 실행합니다.	<p>대상 리전에서 copyProduct.sh 스크립트를 실행합니다. 이 스크립트는 다음 작업을 구현합니다.</p> <ul style="list-style-type: none"> • Lambda 실행 역할 생성 • Lambda 실행 역할을 위한 IAM 정책 생성 및 연결 • Lambda 함수를 생성하고 호출하여 소스 리전에서 대상 리전으로 제품 복사 	AWS 관리자, AWS 시스템 관리자, 클라우드 관리자

관련 리소스

- [Lambda 실행 역할 생성](#)(AWS Lambda 설명서)
- [Lambda 함수 생성](#)(AWS Lambda 설명서)
- [AWS Service Catalog API 참조](#)
- [AWS Service Catalog 설명서](#)

첨부

이 문서와 관련된 추가 콘텐츠에 액세스하려면 [attachment.zip](#) 파일의 압축을 풉니다.

Amazon CloudWatch 이상 탐지를 사용하여 사용자 지정 지표에 대한 경보 생성

작성자: Ram Kandaswamy(AWS), Raheem Jiwani(AWS)

환경: 프로덕션

기술: 관리 및 거버넌스, 운영,
클라우드 네이티브 DevOps

AWS 서비스: 아마존
CloudWatch

요약

Amazon Web Services (AWS) 클라우드에서는 CloudWatch Amazon을 사용하여 지표를 모니터링하고 알림을 보내는 경보를 생성하거나 임계값 위반 시 자동으로 변경할 수 있습니다.

[정적 임계값](#)의 제한을 피하려면 과거 패턴을 기반으로 특정 지표가 정상 작동 범위를 벗어나는 경우 알려주는 경보를 생성할 수 있습니다. 예를 들어 Amazon API Gateway에서 API의 응답 시간을 모니터링하고 서비스 수준에 관한 계약(SLA)을 충족하지 못하게 하는 이상 현상에 대한 알림을 받을 수 있습니다.

이 패턴은 사용자 지정 지표에 대해 CloudWatch 이상 탐지를 사용하는 방법을 설명합니다. 이 패턴은 Amazon CloudWatch Logs Insights에서 사용자 지정 지표를 생성하거나 AWS Lambda 함수를 사용하여 사용자 지정 지표를 게시한 다음 Amazon Simple Notification Service (Amazon SNS) 를 사용하여 예외 항목 탐지를 설정하고 알림을 생성하는 방법을 보여줍니다.

사전 조건 및 제한 사항

사전 조건

- 활성 상태의 AWS 계정.
- 이메일 알림을 보내도록 SNS 주제를 구성할 수 있습니다. 이에 대한 자세한 내용은 Amazon SNS 설명서의 [Amazon SNS 시작하기](#)를 참조하세요.
- [로그로 구성된 기존 애플리케이션. CloudWatch](#)

제한 사항

- CloudWatch 지표는 밀리초 시간 간격을 지원하지 않습니다. 일반 지표와 사용자 지정 지표의 세분성에 대한 자세한 내용은 [Amazon CloudWatch](#) FAQ를 참조하십시오.

아키텍처

이 다이어그램은 다음 워크플로를 보여줍니다.

1. Logs에서 생성하고 업데이트한 지표를 사용하는 CloudWatch 로그는 스트리밍됩니다. CloudWatch
2. 임계값에 따라 경보가 시작되고 SNS 주제에 알림을 보냅니다.
3. Amazon SNS에서 이메일로 이를 알립니다.

기술 스택

- Cloudwatch
- AWS Lambda
- Amazon SNS

도구

- [Amazon Cloudwatch](#) — 안정적이고 확장 가능하며 유연한 모니터링 솔루션을 CloudWatch 제공합니다.
- [AWS Lambda](#) - Lambda는 서버를 프로비저닝하거나 관리하지 않고도 코드를 실행할 수 있게 해주는 컴퓨팅 서비스입니다.
- [Amazon SNS](#) – Amazon Simple Notification Service(SNS)는 게시자에서 구독자(생산자 및 소비자라고도 함)로 메시지를 전송하는 관리형 서비스입니다.

에픽

사용자 지정 지표에 대한 이상 탐지를 설정

작업	설명	필요한 기술
<p>옵션 1 - Lambda 함수를 사용하여 사용자 지정 지표를 생성합니다.</p>	<p>lambda_function.py 파일을 (첨부) 을 다운로드한 다음 AWS 설명서의 aws-lambda-developer-guide 리포지토리에 있는 샘플 lambda_function.py 파일을 GitHub 교체하십시오. 사용자 지정 측정치를 Logs로 전송하는 샘플 Lambda 함수를 제공합니다. CloudWatch Lambda 함수는 Boto3 API를 사용하여 통합합니다. CloudWatch</p> <p>Lambda 함수를 실행한 후 AWS Management Console에 로그인하여 콘솔을 CloudWatch 열면 게시된 네임스페이스에서 게시된 지표를 사용할 수 있습니다.</p>	<p>DevOps 엔지니어, AWS DevOps</p>
<p>옵션 2 — CloudWatch 로그 그룹에서 사용자 지정 메트릭을 생성합니다.</p>	<p>AWS Management 콘솔에 로그인하고 CloudWatch 콘솔을 연 다음 로그 그룹을 선택합니다. 지표를 생성하려는 로그 그룹을 선택합니다.</p> <p>작업을 선택한 후 지표 필터 생성을 선택합니다. 필터 패턴에 사용할 필터 패턴을 입력합니다. 자세한 내용은 CloudWatch 설명서의 필터 및 패턴 구문을 참조하십시오.</p>	<p>DevOps 엔지니어, AWS DevOps</p>

작업	설명	필요한 기술
	<p>(선택 사항) 필터 패턴을 테스트하려면 테스트 패턴에 패턴을 테스트할 로그 이벤트를 하나 이상 입력합니다. 줄 바꿈은 로그 이벤트 메시지 상자에서 로그 이벤트를 구분할 때 사용하므로 각 로그 이벤트는 한 줄을 넘지 않아야 합니다. 패턴을 테스트한 후 지표 세부 정보에 지표의 이름과 값을 입력할 수 있습니다.</p> <p>사용자 지정 지표를 만드는 방법에 대한 자세한 내용 및 단계는 CloudWatch 설명서의 로그 그룹에 대한 지표 필터 만들기를 참조하십시오.</p>	

작업	설명	필요한 기술
<p>사용자 지정 지표에 대한 경보를 생성합니다.</p>	<p>CloudWatch 콘솔에서 [Alarms] 를 선택한 다음 [알람 생성] 을 선택합니다. 지표 선택을 선택하고 검색 상자에 이전에 생성한 지표의 이름을 입력합니다. 그래프로 표시된 지표 탭을 선택하고 필요에 따라 옵션을 구성합니다.</p> <p>조건에서 정적 임계값 대신 이상 현상 탐지를 선택합니다. 이것은 두 개의 표준 디폴트 편차를 기반으로 한 밴드를 보여줍니다. 임계값을 설정하고 필요에 따라 조정할 수 있습니다.</p> <p>다음을 선택합니다.</p> <p>참고: 밴드는 동적이며 데이터 포인트의 품질에 따라 달라집니다. 더 많은 데이터를 집계하기 시작하면 밴드와 임계값이 자동으로 업데이트됩니다.</p>	<p>DevOps 엔지니어, AWS DevOps</p>

작업	설명	필요한 기술
SNS 알림을 설정합니다.	<p>알림에서 경보가 ALARM 상태, OK 상태 또는 INSUFFICIENT_DATA 상태일 때 알릴 SNS 주제를 선택합니다.</p> <p>경보가 동일한 경보 상태 또는 다른 경보 상태에 대해 여러 개의 알림을 보내도록 설정하려면 알림 추가를 선택합니다. 다음을 선택합니다. 경보 이름 및 설명을 입력합니다. 이름은 ASCII 문자만 포함해야 합니다. 다음을 선택합니다.</p> <p>미리 보기 및 생성에서 정보 및 조건이 원하는 내용인지 확인한 다음 경보 생성을 선택합니다.</p>	DevOps 엔지니어, AWS DevOps

관련 리소스

- [에 사용자 지정 메트릭을 게시합니다. CloudWatch](#)
- [CloudWatch 이상 탐지 사용](#)
- [알람 이벤트 및 아마존 EventBridge](#)
- [사용자 지정 지표를 Cloud Watch로 푸시할 때 따라야 할 모범 사례는 무엇입니까?\(동영상\)](#)
- [CloudWatch 애플리케이션 인사이트 소개 \(동영상\)](#)
- [\(비디오\) 를 사용하여 이상 징후를 CloudWatch 탐지합니다.](#)

첨부

이 문서와 관련된 추가 콘텐츠에 액세스하려면 [attachment.zip](#) 파일의 압축을 풉니다.

AWS 랜딩 존 설계를 문서화하십시오

작성자: 마이클 다넷 (AWS), 플로리안 랭거 (AWS), 마이클 로드만 (AWS)

환경: 프로덕션

기술: 관리 및 거버넌스, 인프라, 보안, ID, 규정 준수

AWS 서비스: AWS Control Tower

요약

Landing Zone은 보안 및 규정 준수 모범 사례를 기반으로 잘 설계된 다중 계정 환경입니다. 조직 단위 (OU) AWS 계정, 사용자 및 기타 리소스를 모두 담는 전사적 컨테이너입니다. Landing Zone은 모든 규모의 기업 요구에 맞게 확장할 수 있습니다. AWS에는 랜딩 존을 생성하는 두 가지 옵션이 있습니다. 하나는 사용하는 [AWS Control Tower](#) 서비스 기반 랜딩 존이고 다른 하나는 사용자가 직접 구축하는 맞춤형 랜딩 존입니다. 각 옵션에는 서로 다른 수준의 AWS 지식이 필요합니다.

AWS landing Zone 설정을 자동화하여 시간을 절약할 수 있도록 AWS Control Tower 있도록 만들어졌습니다. AWS Control Tower 기본 환경을 만드는 데 도움이 되는 모범 사례와 지침에 따라 관리되며 이를 사용합니다. AWS AWS Control Tower [AWS Service Catalog](#) 및 와 같은 통합 서비스를 사용하여 [AWS Organizations](#) landing Zone에서 계정을 제공하고 해당 계정에 대한 액세스를 관리합니다.

AWS landing Zone 프로젝트는 요구 사항, 구현 세부 사항 및 운영 조치 항목이 다양합니다. 모든 landing Zone 구현에는 사용자 지정 측면을 처리해야 합니다. 여기에는 액세스 관리 처리 방법, 사용되는 기술 스택, 운영 우수성을 위한 모니터링 요구 사항 등이 포함되며 이에 국한되지 않습니다. 이 패턴은 landing zone 프로젝트를 문서화하는 데 도움이 되는 템플릿을 제공합니다. 템플릿을 사용하면 프로젝트를 더 빠르게 문서화하고 개발 및 운영 팀이 착륙 지대를 이해하는 데 도움이 됩니다.

사전 조건 및 제한 사항

제한 사항

이 패턴은 랜딩 존이 무엇인지 또는 랜딩 존을 구현하는 방법을 설명하지 않습니다. 이러한 주제에 대한 자세한 내용은 [관련 리소스](#) 섹션을 참조하십시오.

에픽

설계 문서 생성

작업	설명	필요한 기술
주요 이해관계자를 파악합니다.	귀하의 landing Zone과 연결된 주요 서비스 및 팀 관리자를 식별하십시오.	프로젝트 관리자
템플릿을 사용자 지정하세요.	<p>첨부 파일 섹션에서 템플릿을 다운로드한 다음 다음과 같이 템플릿을 업데이트합니다.</p> <ol style="list-style-type: none"> 조직의 landing Zone 또는 프로세스에 적용되지 않는 섹션은 모두 삭제하세요. 조직에 고유한 섹션을 모두 추가하세요. 	프로젝트 관리자
템플릿을 완성하세요.	<p>이해관계자와의 미팅에서 또는 write-and-review 프로세스를 사용하여 다음과 같이 템플릿을 완성하십시오.</p> <ol style="list-style-type: none"> 파란색 상자에 있는 지침과 정보를 사용하여 각 섹션을 작성하세요. 노란색 필드를 조직의 사용자 지정 값으로 바꾸거나 제거하세요. 이미지 필드를 사용자 지정 아키텍처 또는 흐름도로 바꾸거나 제거하세요. 템플릿의 수정 기록 및 기여자 섹션을 작성하세요. 	프로젝트 관리자

작업	설명	필요한 기술
디자인 문서를 공유하세요.	Landing Zone 설계 문서가 완성되면 모든 이해 관계자가 액세스할 수 있는 공유 저장소 또는 중앙 위치에 저장하십시오. 표준 문서 관리 프로세스를 사용하여 설계 문서의 개정을 기록하고 승인하는 것이 좋습니다.	프로젝트 관리자

관련 리소스

- [AWS Control Tower 문서화](#)
 - [AWS Control Tower 랜딩 존 계획하기](#)
 - [AWSAWS Control Tower 랜딩 존을 위한 다중 계정 전략](#)
 - [landing Zone 설정을 위한 관리 팁](#)
 - [Landing Zone 구성에 대한 기대치](#)
- [AWS Control Tower\(AWS 솔루션 라이브러리\) 에 대한 사용자 지정](#)
- [안전하고 확장 가능한 다중 계정 AWS 환경 설정 \(AWS 규범적 지침\)](#)

첨부

이 문서와 관련된 추가 콘텐츠에 액세스하려면 [attachment.zip](#) 파일의 압축을 풉니다.

다중 지역, 다중 계정 조직에서 AWS CloudFormation 드리프트 감지 설정

작성자: Ram Kandaswamy(AWS)

환경: 프로덕션

기술: 관리 및 거버넌스, 클라우드 네이티브, 인프라, 운영, 현대화

워크로드: 기타 모든 워크로드

AWS 서비스: 아마존 SNS, AWS Config, AWS Lambda, AWS CloudFormation

요약

Amazon Web Services (AWS) 를 사용하는 고객은 AWS CloudFormation 스택의 드리프트를 포함하여 리소스 구성 불일치를 탐지하고 가능한 한 빨리 수정할 수 있는 효율적인 방법을 찾는 경우가 많습니다. AWS Control Tower 또는 AWS Landing Zone 솔루션을 사용하는 경우 특히 그렇습니다.

이 패턴은 통합 리소스 구성 변경을 사용하고 해당 변경 사항에 따라 조치를 취하여 결과를 생성함으로써 문제를 효율적으로 해결하는 규범적 솔루션을 제공합니다. 이 솔루션은 하나 이상의 지역에서 여러 CloudFormation 스택이 생성되거나 둘 이상의 계정 또는 이들의 조합으로 생성되는 시나리오를 위해 설계되었습니다. 이 솔루션의 목표는 다음과 같습니다.

- 드리프트 감지 프로세스 간소화
- 알림 및 경고 설정
- 통합 보고 설정

사전 조건 및 제한 사항

사전 조건

- 모니터링해야 하는 모든 리전 및 계정에서 활성화된 AWS Config

제한 사항

- 생성된 보고서는 .csv 또는.json 출력 형식만 지원합니다.

아키텍처

대상 기술 스택

현재 지침은 조직이 다음과 같은 서비스의 조합을 사용하여 목표를 달성하는 데 도움이 됩니다.

- AWS Config 규칙
- 아마존 CloudWatch 룰
- AWS Identity and Access Management(IAM)
- AWS Lambda
- Amazon Simple Notification Service(Amazon SNS)

1. AWS Config 규칙은 드리프트를 감지합니다.
2. 다른 계정의 드리프트 감지 결과는 관리 계정으로 전송됩니다.
3. 이 CloudWatch 규칙은 Lambda를 호출합니다.
4. Lambda는 AWS Config 규칙을 쿼리하여 집계된 결과를 확인합니다.
5. Lambda는 Amazon SNS에 알리고 Amazon SNS는 드리프트에 대한 이메일 알림을 보냅니다.

자동화 및 규모 조정

여기에 제시된 솔루션은 추가 리전 및 계정 모두에 맞게 확장할 수 있습니다.

도구

[AWS Config](#) - AWS Config는 사용자의 AWS 계정에서 AWS 리소스의 구성을 상세하게 볼 수 있도록 합니다. 이러한 보기에는 리소스 간에 어떤 관계가 있는지와 리소스가 과거에 어떻게 구성되었는지도 포함되므로, 시간이 지나면서 구성과 관계가 어떻게 변하는지 확인할 수 있습니다. AWS Config를 통해 AWS 리소스 구성을 측정, 감사 및 평가할 수 있습니다.

[Amazon CloudWatch](#) — Amazon은 AWS 리소스와 AWS에서 실행하는 애플리케이션을 실시간으로 CloudWatch 모니터링합니다. 리소스와 애플리케이션에 대해 측정할 수 있는 변수인 지표를 수집하고 추적하는 데 사용할 CloudWatch 수 있습니다.

AWS Lambda – AWS Lambda는 서버를 프로비저닝하거나 관리하지 않고도 코드를 실행하도록 지원하는 컴퓨팅 서비스입니다. Lambda는 필요 시에만 코드를 실행하며, 일일 몇 개의 요청에서 초당 수천 개의 요청까지 자동으로 규모를 조정합니다. 사용한 컴퓨팅 시간만큼만 비용을 지불하고, 코드가 실행되지 않을 때는 요금이 부과되지 않습니다.

Amazon SNS – Amazon Simple Notification Service(Amazon SNS)는 게시자에서 구독자(생산자 및 소비자라고도 함)로 메시지를 전송하는 관리형 서비스입니다.

에픽

에 대한 드리프트 감지를 자동화하십시오. CloudFormation

작업	설명	필요한 기술
애그리게이터를 만듭니다.	AWS Config 콘솔에서 관리 계정에 애그리게이터를 생성합니다. AWS Config가 소스 계정에서 데이터를 가져올 수 있도록 데이터 복제가 켜져 있는지 확인하십시오. 또한, 해당하는 모든 리전 및 계정을 선택합니다. 조직을 기반으로 계정을 선택할 수 있습니다. 조직에 있는 새 계정은 자동으로 애그리게이터에 속하므로 이 방법을 사용하는 것을 권장합니다.	클라우드 아키텍트
AWS 관리형 규칙을 생성합니다.	cloudformation-stack-drift-detection-checkAWS 관리형 규칙을 추가합니다. 이 규칙에는 하나의 파라미터 값(cloudformationArn)이 필요합니다. 스택 드리프트 감지 권한이 있는 IAM 역할 Amazon 리소스 이름(ARN)을 입력합니다. 또한, 역할에는 AWS Config가 역할을	클라우드 아키텍트

작업	설명	필요한 기술
	<p>맡는 것을 가능하게 하는 신뢰 정책이 있어야 합니다.</p>	
<p>애그리게이터의 고급 쿼리 섹션을 생성합니다.</p>	<p>여러 소스에서 드리프트된 스택을 가져오려면 다음과 같은 쿼리를 생성합니다.</p> <pre>SELECT resourceId, configuration.driftInformation.stackDriftStatus WHERE resourceType = 'AWS::CloudFormation::Stack' AND configuration.driftInformation.stackDriftStatus IN ('DRIFTED')</pre>	<p>클라우드 아키텍트, 개발자</p>
<p>쿼리 실행을 자동화하고 게시합니다.</p>	<p>첨부된 코드를 사용하여 Lambda 함수를 생성합니다. Lambda는 Lambda 함수에서 환경 변수로 제공되는 Amazon SNS 주제에 결과를 게시합니다. 또한, 알림을 받으려면 기존 Amazon SNS 주제에 대해 이메일 구독을 만듭니다.</p>	<p>클라우드 아키텍트, 개발자</p>
<p>CloudWatch 규칙 만들기.</p>	<p>알림을 담당하는 Lambda 함수를 호출하는 스케줄 기반 CloudWatch 규칙을 생성합니다.</p>	<p>클라우드 아키텍트</p>

관련 리소스

리소스

- [AWS Config란 무엇입니까?](#)
- [개념: 다중 계정 다중 리전 데이터 집계](#)
- [다중 계정 다중 리전 데이터 집계](#)
- [스택 및 리소스에 대한 비관리형 구성 변경 감지](#)
- [IAM: IAM 역할을 특정 AWS 서비스로 전달](#)
- [Amazon SNS란 무엇인가요?](#)

추가 정보

고려 사항

특정 간격으로 API 호출을 포함하는 사용자 지정 솔루션을 사용하여 각 CloudFormation 스택 또는 스택 세트에서 드리프트 감지를 시작하는 것은 최적이 아닙니다. 이렇게 하면 많은 수의 API 호출이 발생하므로 성능에 영향을 미칩니다. API 호출 수로 인해 스로틀링이 발생할 수 있습니다. 또 다른 잠재적 문제는 예약만을 기준으로 리소스 변경 사항을 파악하면 감지가 지연되는 것입니다.

FAQ

질문: AWS Landing Zone과 함께 애드온 기반 솔루션을 사용해야 합니까?

답변: 애그리게이터와 함께 AWS Config에서 고급 쿼리 기능을 사용할 수 있으므로 추가 기능 대신 AWS Config를 사용하는 것이 좋습니다.

Q: 이 솔루션은 어떻게 해결됩니까? CloudFormation StackSets

답변: 스택 세트는 스택으로 구성되어 있으므로 이 솔루션을 사용할 수 있습니다. 스택 인스턴스 세부 정보를 이 솔루션의 일부로 사용할 수도 있습니다.

첨부

이 문서와 관련된 추가 콘텐츠에 액세스하려면 [attachment.zip](#) 파일의 압축을 풉니다.

AWS CDK를 사용하여 여러 AWS 지역, 계정 및 DevOps OU에서 Amazon Guru를 활성화하여 운영 성능을 개선하십시오.

작성자: Dr. Rahul Gaikwad(AWS)

코드 리포지토리: [Amazon DevOps Guru 샘플 코드](#)

환경: PoC 또는 파일럿

기술: 관리 및 거버넌스, 클라우드 네이티브, DevOps 운영, 보안, ID, 규정 준수, 서버리스

AWS 서비스: 아마존 API 게이트웨이, AWS CDK, 아마존 DevOps 구루, 아마존 DynamoDB, AWS Organizations

요약

이 패턴은 여러 AWS 클라우드 개발 키트 (AWS CDK) 를 사용하여 여러 Amazon Web Services (AWS) 지역, 계정 및 조직 단위 (DevOps OU) 에서 Amazon Guru 서비스를 활성화하는 단계를 보여줍니다. TypeScript 각 계정에 로그인하여 계정별로 DevOps Guru를 활성화하는 대신, AWS CDK 스택을 사용하여 관리자 (기본) AWS CloudFormation StackSets 계정으로 AWS를 배포하여 여러 계정에서 Amazon DevOps Guru를 활성화할 수 있습니다.

Amazon DevOps Guru는 애플리케이션의 가용성을 개선하고 운영 문제를 더 빠르게 해결하는 데 도움이 되는 인공 지능 운영 (AIOps) 기능을 제공합니다. DevOps Guru는 ML 전문 지식 없이도 기계 학습 (ML) 기반 권장 사항을 적용하여 수동 작업을 줄여줍니다. DevOps Guru는 리소스와 운영 데이터를 분석합니다. 이상 징후가 감지되면 문제 해결에 도움이 되는 지표, 이벤트, 권장 사항을 제공합니다.

이 패턴은 Amazon DevOps Guru를 활성화하기 위한 세 가지 배포 옵션을 설명합니다.

- 여러 계정과 리전에 있는 모든 스택 리소스용
- OU에 있는 모든 스택 리소스용
- 여러 계정과 리전에 있는 특정 스택 리소스용

사전 조건 및 제한 사항

사전 조건

- 활성 상태의 AWS 계정
- AWS Command Line Interface(AWS CLI), 설치 및 구성됨. (AWS CLI 문서에서 [AWS CLI 설치, 업데이트, 제거](#) 참조)
- AWS CDK Toolkit, 설치 및 구성됨. (AWS CDK 설명서에서 [AWS CDK Toolkit](#) 참조)
- AWS CDK용으로 설치 및 구성된 노드 패키지 관리자 (npm) TypeScript (npm 문서에서 [Node.js 및 npm 다운로드 및 설치하기](#) 참조)
- 샘플 서버리스 애플리케이션에 트래픽을 유입하기 위한 Python 스크립트를 실행할 수 있도록 설치 및 구성된 Python3 ([Python 문서에서 Python 설정 및 사용법](#) 참조)
- Python 요청 라이브러리 설치를 위해 설치 및 구성된 Pip (웹 사이트의 [pip 설치 지침](#)을 참조하십시오.) PyPI

제품 버전

- AWS CDK Toolkit 버전 1.107.0 이상
- npm 버전 7.9.0 이상
- Node.js 버전 15.3.0 이상

아키텍처

기술

이 패턴의 아키텍처에는 다음과 같은 서비스가 포함됩니다.

- [아마존 DevOps 전문가](#)
- [AWS CloudFormation](#)
- [Amazon API Gateway](#)
- [AWS Lambda](#)
- [Amazon DynamoDB](#)
- [아마존 CloudWatch](#)
- [AWS CloudTrail](#)

AWS CDK 스택

패턴은 다음과 같은 AWS CDK 스택을 사용합니다.

- `CdkStackSetAdminRole` - 관리자와 대상 계정 간의 신뢰 관계를 설정하기 위해 AWS IAM(신원 및 액세스 관리) 관리자 역할을 생성
- `CdkStackSetExecRole` - 관리자 계정을 신뢰하는 IAM 역할을 생성
- `CdkDevopsGuruStackMultiAccReg`— DevOps Guru가 여러 AWS 지역 및 모든 스택에 대해 계정을 사용할 수 있도록 하고 Amazon Simple Notification Service (Amazon SNS) 알림을 설정할 수 있습니다.
- `CdkDevopsGuruStackMultiAccRegSpecStacks`— DevOps Guru가 여러 AWS 지역 및 계정에서 특정 스택을 사용하도록 지원하고 Amazon SNS 알림을 설정할 수 있습니다.
- `CdkDevopsGuruStackOrgUnit`— DevOps OU 전반에 걸쳐 Guru를 활성화하고 Amazon SNS 알림을 설정할 수 있습니다.
- `CdkInfrastructureStack`— 관리자 계정에 API 게이트웨이, Lambda, DynamoDB와 같은 샘플 서버리스 애플리케이션 구성 요소를 배포하여 결합 삽입 및 인사이트 생성

샘플 애플리케이션 아키텍처

다음 다이어그램은 여러 계정과 리전에 배포된 샘플 서버리스 애플리케이션 아키텍처를 보여줍니다. 이 패턴은 관리자 계정을 사용하여 모든 AWS CDK 스택을 배포합니다. 또한 관리자 계정을 DevOps Guru를 설정하는 대상 계정 중 하나로 사용합니다.

1. DevOps Guru를 활성화하면 먼저 각 리소스의 동작 기준을 정한 다음 제공된 지표에서 운영 데이터를 수집합니다. CloudWatch
2. 이상 징후가 감지되면 해당 데이터를 의 이벤트와 연관시키고 통찰력을 생성합니다. CloudTrail
3. 인사이트는 운영자가 원인 리소스를 식별할 수 있도록 규정된 권장 사항과 함께 상호 연관된 이벤트 시퀀스를 제공합니다.
4. Amazon SNS는 운영자에게 알림 메시지를 보냅니다.

자동화 및 규모 조정

이 패턴으로 제공된 [GitHub 리포지토리](#)는 AWS CDK를 코드형 인프라 (IaC) 도구로 사용하여 이 아키텍처에 대한 구성을 생성합니다. AWS CDK를 사용하면 여러 AWS 계정, 지역 및 DevOps OU에서 리소스를 오케스트레이션하고 Guru를 지원할 수 있습니다.

도구

서비스

- [AWS CDK](#) — AWS Cloud Development Kit (AWS CDK) 를 사용하면 지원되는 5가지 프로그래밍 언어 (TypeScript, JavaScript Python, Java, C #) 중 하나로 클라우드 인프라를 코드로 정의할 수 있습니다.
- [AWS CLI](#) - AWS Command Line Interface(AWS CLI)는 AWS 서비스, 리소스와 상호 작용할 수 있는 일관된 명령줄 인터페이스를 제공하는 통합 도구입니다.

코드

이 패턴의 소스 코드는 [Amazon DevOps Guru CDK 샘플](#) 리포지토리에서 확인할 수 있습니다. GitHub AWS CDK 코드가 작성되었습니다. TypeScript 리포지토리를 복제하여 사용하려면 다음 섹션의 지침을 따르세요.

중요: 이 패턴의 일부 사례에는 Unix, Linux, macOS용으로 포맷된 AWS CDK, AWS CLI 명령어 예제가 포함되어 있습니다. Windows에서는 각 줄 끝에 있는 백슬래시(\) 연속 문자를 캐럿(^)으로 바꿉니다.

에픽

배포에 필요한 AWS 리소스 준비

작업	설명	필요한 기술
AWS 네임드 프로필을 구성	<p>다중 계정 환경에서 스택을 배포하려면 다음과 같이 AWS 네임드 프로필을 설정하세요.</p> <p>관리자 계정의 경우</p> <pre>\$aws configure --profile administrator AWS Access Key ID [****]: <your-administrator-access-key-ID> AWS Secret Access Key [****]: <your-administrator-access-key-ID></pre>	DevOps 엔지니어

작업	설명	필요한 기술
	<pre> inistrator-secret- access-key> Default region name [None]: <your-adm inistrator-region> Default output format [None]: json </pre> <p>대상 계정의 경우</p> <pre> \$aws configure --profile target AWS Access Key ID [****]: <your-target- access-key-ID> AWS Secret Access Key [****]: <your-target- secret-access-key> Default region name [None]: <your-target- region> Default output format [None]: json </pre> <p>자세한 내용은 AWS CLI 설 명서에서 네임드 프로파일 사 용을 참조하세요.</p>	
AWS 프로파일 구성 확인	(선택 사항) AWS CLI 설명서의 설정 및 보기 구성 설정 에 있는 지침에 따라 credentials 및 config 파일에서 AWS 프로파일 구성을 확인할 수 있습니다.	DevOps 엔지니어

작업	설명	필요한 기술
AWS CDK 버전 확인	<p>다음 명령을 실행하여 AWS CDK Toolkit 버전을 확인하세요.</p> <pre>\$cdk --version</pre> <p>이 패턴을 사용하려면 버전 1.107.0 이상이 필요합니다. 이전 버전의 AWS CDK를 사용 중이면 AWS CDK 문서에 있는 지침에 따라 업데이트하세요.</p>	DevOps 엔지니어
프로젝트 코드 복제	<p>다음 명령을 사용하여 이 패턴의 GitHub 저장소를 복제합니다.</p> <pre>\$git clone https://github.com/aws-samples/amazon-devops-guru-cdk-samples.git</pre>	DevOps 엔지니어

작업	설명	필요한 기술
패키지 종속성을 설치하고 파일을 컴파일합니다. TypeScript	<p>다음 명령을 실행하여 패키지 종속성을 설치하고 TypeScript 파일을 컴파일합니다.</p> <pre data-bbox="594 394 1026 592">\$cd amazon-devopsguru-cdk-samples \$npm install \$npm fund</pre> <p>이 명령은 샘플 저장소의 모든 패키지를 설치합니다.</p> <p>중요: 누락된 패키지와 관련된 오류가 발생하면 다음 명령 중 하나를 사용하세요.</p> <pre data-bbox="594 926 1026 1003">\$npm ci</pre> <p>—또는—</p> <pre data-bbox="594 1115 1026 1234">\$npm install -g @aws-cdk/<package-name></pre> <p>패키지 이름과 버전 목록은 <code>/amazon-devopsguru-cdk-samples/package.json</code> 파일의 <code>Dependencies</code> 섹션에서 확인할 수 있습니다. 자세한 내용은 npm 설명서에 있는 npm ci와 npm install을 참조하세요.</p>	DevOps 엔지니어

AWS CDK 스택 작성(합성)

작업	설명	필요한 기술
<p>Amazon SNS 알림 이메일 주소를 구성합니다.</p>	<p>Amazon SNS 알림 이메일 주소를 제공하려면 다음 단계를 따르세요.</p> <ol style="list-style-type: none"> 1. 파일 <code>/amazon-devopsguru-cdk-samples/lib/cdk-devopsguru-multi-acc-reg-stack.ts</code> 및 <code>/amazon-devopsguru-cdk-samples/lib/cdk-devopsguru-org-uni-stack.ts</code> 편집 2. <code>DevOpsGuruTopic</code>, <code>Subscription</code> 섹션에서 <code>Endpoint</code> 매개변수를 이메일 주소로 업데이트합니다. 3. 파일을 저장하고 닫습니다. 	<p>DevOps 엔지니어</p>
<p>프로젝트 코드 작성</p>	<p>명령을 실행하여 프로젝트 코드를 작성하고 스택을 합성하세요.</p> <pre data-bbox="597 1394 1026 1514">npm run build && cdk synth</pre> <p>다음과 유사한 출력 화면이 표시되어야 합니다.</p> <pre data-bbox="597 1671 1026 1881">\$npm run build && cdk synth > cdk-devopsguru@0.1.0 build > tsc</pre>	<p>DevOps 엔지니어</p>

작업	설명	필요한 기술
	<p>Successfully synthesized to ~/amazon-devopsguru-cdk-samples/cdk.out</p> <p>Supply a stack id (CdkDevopsGuruStackMultiAccReg, CdkDevopsGuruStackMultiAccRegSpecStacks, CdkDevopsGuruStackOrgUnit, CdkInfrastructureStack, CdkStackSetAdminRole, CdkStackSetExecRole) to display its template.</p> <p>자세한 정보와 절차는 AWS CDK 문서에서 첫 번째 AWS CDK 앱을 참조하세요.</p>	

작업	설명	필요한 기술
AWS CDK 스택을 나열합니다.	<p>다음 명령을 실행하여 모든 AWS CDK 스택을 나열하세요.</p> <pre>\$cdk list</pre> <p>이 명령은 다음 목록을 표시합니다.</p> <pre>CdkDevopsGuruStack MultiAccReg CdkDevopsGuruStack ackMultiAccRegSpec Stacks CdkDevopsguruStackOr gUnit CdkInfrastructureStack CdkStackSetAdminRole CdkStackSetExecRole</pre>	DevOps 엔지니어

옵션 1 - 여러 계정의 모든 스택 리소스에 대해 DevOps Guru를 활성화합니다.

작업	설명	필요한 기술
IAM 역할 생성에 필요한 AWS CDK 스택 배포	<p>이 패턴은 CloudFormation StackSetsAWS를 사용하여 여러 계정에서 스택 작업을 수행합니다. 첫 번째 스택 집합을 생성하는 경우, AWS 계정에 필요한 권한을 설정하려면 다음 IAM 역할을 만들어야 합니다.</p> <ul style="list-style-type: none"> AWSCloudFormationStackSetAdministrationRole 	DevOps 엔지니어

작업	설명	필요한 기술
	<ul style="list-style-type: none"> • <code>AWSCloudFormationStackSetExecutionRole</code> <p>참고: 역할에는 다음과 같은 정확한 이름이 있어야 합니다.</p> <ol style="list-style-type: none"> 1. 다음 CLI 명령을 실행하여 관리자(기본) 계정에서 IAM <code>AWSCloudFormationStackSetAdministrationRole</code> 역할을 만듭니다. <pre>\$cdk deploy CdkStackSetAdminRole --profile administrator</pre> <ol style="list-style-type: none"> 2. 스택 인스턴스를 실행하려는 모든 대상 계정에서 IAM <code>AWSCloudFormationStackSetExecutionRole</code> 역할을 만듭니다. 이 역할을 만들려면 다음 CLI 명령을 실행하세요. <pre>\$cdk deploy CdkStackSetExecRole \ --parameters AdministratorAccountId=<administrator-account-ID> \ --profile administrator</pre> <pre>\$cdk deploy CdkStackSetExecRole \</pre>	

작업	설명	필요한 기술
	<pre data-bbox="630 205 1029 428">--parameters AdministratorAccou ntId=<administrato r-account-ID> \ --profile target</pre> <p data-bbox="591 491 1029 674">자세한 내용은 AWS CloudFormation 설명서의 자체 관리 권한 부여를 참조하십시오.</p>	

작업	설명	필요한 기술
<p>AWS CDK 스택을 배포하여 여러 계정에 DevOps Guru를 활성화하십시오.</p>	<p>AWS CDK CdkDevops GuruStackMultiAccReg 스택은 여러 계정과 리전에 걸쳐 스택 인스턴스를 배포하기 위한 스택 세트를 생성합니다. 스택을 배포하려면 지정된 매개변수와 함께 다음 CLI 명령을 실행합니다.</p> <pre data-bbox="597 636 1027 1270"> \$cdk deploy CdkDevops GuruStackMultiAccReg \ --profile administrator \ --parameters AdministratorAccountId=<administrator-account-ID> \ --parameters TargetAccountId=<target-account-ID> \ --parameters RegionIds="<region-1>,<region-2>" </pre> <p>현재 Amazon DevOps Guru는 Guru FAQ에 나열된 AWS 지역에서 사용할 수 있습니다. DevOps</p>	<p>DevOps 엔지니어</p>

옵션 2 - DevOps OU 전체의 모든 스택 리소스에 대해 Guru를 활성화합니다.

작업	설명	필요한 기술
OU ID 추출	AWS Organizations 콘솔에서 DevOps Guru를 활성화하려는 조직 단위의 ID를 식별합니다.	DevOps 엔지니어
OU에 대한 서비스 관리형 권한 활성화	계정 관리에 AWS Organizations를 사용하는 경우 서비스 관리 권한을 부여하여 Guru를 DevOps 활성화해야 합니다. IAM 역할을 수동으로 만드는 대신 조직 기반 신뢰할 수 있는 액세스 및 서비스 연결 역할 (SLR) 을 사용하세요.	DevOps 엔지니어
AWS CDK 스택을 배포하여 DevOps OU 전체에 Guru를 활성화하십시오.	AWS CDK CdkDevops guruStackOrgUnit 스택은 OU 전반에 걸쳐 DevOps Guru 서비스를 지원합니다. 스택을 배포하려면 지정된 매개 변수와 함께 다음 명령을 실행하세요.	DevOps 엔지니어

```

$cdk deploy CdkDevops
guruStackOrgUnit \
  --profile administrator \
  --parameters
  RegionIds="<region-1>,<region-2>" \
  --parameters
  OrganizationalUnit
  Ids="<OU-1>,<OU-2>"
    
```

옵션 3 - 여러 계정의 특정 스택 리소스에 대해 DevOps Guru를 활성화합니다.

작업	설명	필요한 기술
IAM 역할 생성에 필요한 AWS CDK 스택 배포	<p>첫 번째 옵션에 표시된 필수 IAM 역할을 아직 만들지 않았다면 먼저 만들어야 합니다.</p> <ol style="list-style-type: none"> 다음 CLI 명령을 실행하여 관리자(기본) 계정에서 IAM <code>AWSCloudFormationStackSetAdministrationRole</code> 역할을 만듭니다. <pre>\$cdk deploy CdkStackSetAdminRole --profile administrator</pre> <ol style="list-style-type: none"> 스택 인스턴스를 실행하려는 모든 대상 계정에서 IAM <code>AWSCloudFormationStackSetExecutionRole</code> 역할을 만듭니다. 이 역할을 만들려면 CLI 명령을 실행하세요. <pre>\$cdk deploy CdkStackSetExecRole \ --parameters AdministratorAccountId=<administrator-account-ID> \ --profile administrator</pre> <pre>\$cdk deploy CdkStackSetExecRole \ --parameters AdministratorAccou</pre>	DevOps 엔지니어

작업	설명	필요한 기술
	<pre>ntId=<administrator-account-ID> \ --profile target</pre> <p>자세한 내용은 AWS CloudFormation 설명서의 자체 관리 권한 부여를 참조하십시오.</p>	
<p>기존 스택을 삭제하세요.</p>	<p>모든 스택 리소스에 대해 DevOps Guru를 활성화하는 첫 번째 옵션을 이미 사용한 경우 다음 명령을 사용하여 이전 스택을 삭제할 수 있습니다.</p> <pre>\$cdk destroy CdkDevopsGuruStackMultiAccR eg --profile administrator</pre> <p>또는 스택을 다시 배포할 때 <code>RegionIds</code> 매개변수를 변경하여 스택이 이미 존재함 오류를 방지할 수 있습니다.</p>	<p>DevOps 엔지니어</p>

작업	설명	필요한 기술
<p>스택 목록으로 AWS CDK 스택을 업데이트하세요.</p>	<ol style="list-style-type: none"> 1. <code>/amazon-devopsguru-cdk-samples/lib/cdk-devopsguru-multi-acc-reg-spec-stack.ts</code> 파일을 편집합니다. 2. <code>Resources</code>, <code>CloudFormation StackNames</code>, 아래에 DevOps Guru를 활성화하려는 스택을 나열하십시오. 데모용으로 이 매개변수는 <code>CdkInfrastructureStack</code> 스택을 지정하지만 요구 사항에 따라 이 항목을 편집할 수 있습니다. 3. 파일을 저장하고 닫습니다. 4. 스택 템플릿을 합성하고 업데이트하려면 다음을 실행합니다. <div style="border: 1px solid #ccc; border-radius: 10px; padding: 5px; width: fit-content; margin: 10px auto;"> <code>\$cdk synth</code> </div>	<p>데이터 엔지니어</p>

작업	설명	필요한 기술
<p>DevOps Guru가 여러 계정의 특정 스택 리소스를 사용할 수 있도록 AWS CDK 스택을 배포하십시오.</p>	<p>AWS CDK CdkDevops GuruStackMultiAccR egSpecStacks 스택을 사용하면 DevOps Guru가 여러 계정의 특정 스택 리소스를 사용할 수 있습니다. 스택을 배포하려면 다음 명령을 실행하십시오.</p> <pre data-bbox="597 636 1024 1268"> \$cdk deploy CdkDevops GuruStackMultiAccR egSpecStacks \ --profile administrator \ --parameters AdministratorAccountID=<administrator-account-ID> \ --parameters TargetAccountId=<target-account-ID> \ --parameters RegionIds="<region-1>,<region-2>" </pre> <p>참고: 이전에 옵션 1을 위해 이 스택을 배포했다면 RegionIds 매개변수를 변경(사용 가능한 리전 중에서 선택)하여 스택이 이미 존재함 오류를 방지하십시오.</p>	<p>DevOps 엔지니어</p>

AWS CDK 인프라 스택 배포

작업	설명	필요한 기술
<p>샘플 서버리스 인프라 스택을 배포하세요.</p>	<p>AWS CDK CdkInfrastructureStack 스택은 API Gateway, Lambda 및 DynamoDB 테이블과 같은 서버리스 구성 요소를 배포하여 Guru 통찰력을 입증합니다. DevOps 스택을 배포하려면 다음 명령을 실행하세요.</p> <pre data-bbox="597 722 1026 884"> \$cdk deploy CdkInfrastructureStack --profile administrator </pre>	<p>DevOps 엔지니어</p>
<p>DynamoDB에 샘플 레코드를 삽입하세요.</p>	<p>다음 명령을 실행하여 샘플 레코드로 DynamoDB 테이블을 채웁니다. populate-shops-dynamodb-table.json 스크립트의 올바른 경로를 입력합니다.</p> <pre data-bbox="597 1234 1026 1591"> \$aws dynamodb batch-write-item \ --request-items file://scripts/populate-shops-dynamodb-table.json \ --profile administrator </pre> <p>이 명령은 다음 출력을 표시합니다.</p> <pre data-bbox="597 1751 1026 1881"> { "UnprocessedItems" : {} </pre>	<p>DevOps 엔지니어</p>

작업	설명	필요한 기술
	}	
<p>DynamoDB에서 삽입된 레코드를 확인하세요.</p>	<p>DynamoDB 테이블에 populate-shops-dynamodb-table.json 파일의 샘플 레코드가 포함되어 있는지 확인하려면 AWS CDK 스택의 출력으로 게시되는 ListRestApiEndpointMonitorOperator API의 URL에 액세스합니다. CdkInfrastructureStack 스택용 AWS CloudFormation 콘솔의 출력 탭에서도 이 URL을 찾을 수 있습니다. AWS CDK 출력은 다음과 유사하게 표시됩니다.</p> <pre data-bbox="594 1094 1029 1812"> CdkInfrastructureStack.CreateRestApiEndpointD1D00045 = https://oure17c5vob.execute-api.<your-region>.amazonaws.com/prod/ CdkInfrastructureStack.ListRestApiEndpointABDB8D8 = https://cdff8icfrn4.execute-api.<your-region>.amazonaws.com/prod/ </pre>	<p>DevOps 엔지니어</p>

작업	설명	필요한 기술
리소스가 기존선 설정을 완료할 때까지 기다립니다.	이 서버리스 스택에는 몇 가지 리소스가 있습니다. 다음 단계를 수행하기 전에 2시간 정도 기다리는 것이 좋습니다. 프로덕션 환경에 이 스택을 배포한 경우 Guru에서 DevOps 모니터링하도록 선택한 리소스 수에 따라 기존 설정을 완료하는 데 최대 24시간이 걸릴 수 있습니다.	DevOps 엔지니어

DevOps 전문가 인사이트 생성

작업	설명	필요한 기술
AWS CDK 인프라 스택을 업데이트하세요.	<p>DevOps Guru Insights를 사용해 보려면 구성을 일부 변경하여 일반적인 운영 문제를 재현할 수 있습니다.</p> <ol style="list-style-type: none"> 1. <code>/amazon-devopsguru-cdk-samples/lib/infrastructure-stack.ts</code> 파일을 편집합니다. 2. DDB Table 섹션에서 DynamoDB 테이블의 읽기 용량을 5에서 1로 변경합니다. 3. 파일을 저장하고 닫습니다. 4. 다음 명령을 실행하여 업데이트된 AWS CDK 인프라 스택을 합성하고 배포합니다. <pre>\$cdk synth</pre>	DevOps 엔지니어

작업	설명	필요한 기술
	<pre>\$cdk deploy CdkInfras tructureStack -- profile administrator</pre>	

작업	설명	필요한 기술
<p>API에 HTTP 요청을 삽입합니다.</p>	<p>ListRestApiMonitor OperatorEndpointxx xx API에 HTTP 요청 형태로 인그레스 트래픽을 삽입합니다.</p> <ol style="list-style-type: none"> 1. Python 스크립트 <code>/amazon-devopsguru-cdk-samples/scripts/sendAPIRequest.py</code> 편집 2. ListRestApiMonitor OperatorEndpointxx xx 에 대한 API 링크로 <code>url</code> 변수를 업데이트합니다. 이 URL은 AWS CDK 배포 명령의 출력 또는 AWS Cloudformation 콘솔의 스택에 대한 출력 탭에서 찾을 수 있습니다. 3. 파일을 저장하고 닫습니다. 4. 다음 명령을 사용하여 Python 스크립트를 실행합니다. <div data-bbox="630 1409 1029 1528" style="border: 1px solid #ccc; border-radius: 10px; padding: 10px; margin: 10px 0;"> <pre>\$python sendAPIRequest.py</pre> </div> <ol style="list-style-type: none"> 5. 200 상태 코드를 받았는지 확인하세요. 6. 빠른 속도로 트래픽을 유입하려면 여러 대(가급적 4대)의 터미널에서 스크립트를 실행해야 할 수 있습니다. 	<p>DevOps 엔지니어</p>

작업	설명	필요한 기술
	<p>7. 스크립트가 약 10분 동안 루프로 실행되면 DevOps Guru 콘솔에서 운영 통찰력을 확인할 수 있습니다.</p>	
<p>DevOps Guru 인사이트를 검토하세요.</p>	<p>표준 조건에서 DevOps Guru 대시보드는 진행 중인 인사이트 카운터에 0을 표시합니다. 이상 징후가 감지되면 인사이트 형태로 경보를 발령합니다. 탐색 창에서 인사이트를 선택하여 개요, 집계된 메트릭, 관련 이벤트, 권장 사항 등 이상 징후에 대한 세부 정보를 확인합니다. 인사이트 검토에 대한 자세한 내용은 Amazon DevOps Guru를 사용하여 AIOps로 운영 인사이트 확보하기 블로그 게시물을 참조하십시오.</p>	<p>DevOps 엔지니어</p>

정리

작업	설명	필요한 기술
<p>리소스를 정리하고 삭제합니다.</p>	<p>이 패턴을 수행한 다음에는 추가 요금이 발생하지 않도록 생성한 리소스를 제거해야 합니다. 명령을 실행합니다.</p> <pre data-bbox="597 1591 1024 1881"> \$cdk destroy CdkDevops GuruStackMultiAccR eg --profile administrator \$cdk destroy CdkDevops guruStackOrgUnit -- profile administrator </pre>	<p>DevOps 엔지니어</p>

작업	설명	필요한 기술
	<pre> \$cdk destroy CdkDevops GuruStackMultiAccR egSpecStacks --profile administrator \$cdk destroy CdkInfras tructureStack -- profile administrator \$cdk destroy CdkStackS etAdminRole --profile administrator \$cdk destroy CdkStackS etExecRole --profile administrator \$cdk destroy CdkStackS etExecRole --profile target </pre>	

관련 리소스

- [Amazon DevOps Guru를 사용하여 AIOps를 통한 운영 인사이트 확보](#)
- [AWS를 사용하여 여러 계정 및 지역에서 Amazon DevOps Guru를 쉽게 구성할 수 있습니다. CloudFormation StackSets](#)
- [DevOps 전문가 워크숍](#)

부트스트랩 파이프라인을 사용하여 테라폼용 Account Factory (AFT) 구현

작성자: 비니시우스 엘리아스 (AWS) 와 에드가 코스타 필류 (AWS)

코드 리포지토리: aft-boots trap-pipeline	환경: 프로덕션	기술: 관리 및 거버넌스, 인프라
워크로드: 오픈 소스	AWS 서비스: AWS, AWS CodeBuild, AWS CodeCommit, AWS 컨트롤 타워 CodePipeline, AWS Organizations	

요약

이 패턴은 의 관리 AWS Control Tower 계정에서 Account Factory for Terraform (AFT) 을 배포하는 간단하고 안전한 방법을 제공합니다. AWS Organizations 솔루션의 핵심은 초기 배포 또는 후속 업데이트에 쉽게 적용할 수 있도록 구성된 Terraform 파이프라인을 생성하여 AFT 구성을 자동화하는 AWS CloudFormation 템플릿입니다.

에서는 AWS 보안 및 데이터 무결성이 최우선 사항이므로 관리되는 인프라 및 구성의 상태를 추적하는 중요한 구성 요소인 Terraform 상태 파일은 Amazon Simple Storage Service (Amazon S3) 버킷에 안전하게 저장됩니다. 이 버킷은 서버 측 암호화 및 퍼블릭 액세스 차단 정책을 비롯한 여러 보안 조치로 구성되어 있어 Terraform 상태를 무단 액세스 및 데이터 침해로부터 보호하는 데 도움이 됩니다.

관리 계정은 전체 환경을 오케스트레이션하고 감독하므로 관리 계정은 중요한 리소스입니다. AWS Control Tower이 패턴은 AWS 모범 사례를 따르며 배포 프로세스가 효율적일 뿐만 아니라 보안 및 거버넌스 표준에 부합하도록 보장하여 환경에 AFT를 배포하는 포괄적이고 안전하며 효율적인 방법을 제공합니다. AWS

AFT에 대한 자세한 내용은 [AWS Control Tower 설명서](#)를 참조하십시오.

사전 조건 및 제한 사항

사전 조건

- 최소한 관리 AWS 계정, 로그 아카이브 계정, 감사 계정 및 AFT 관리용 추가 계정 1개가 있는 기본 다중 계정 환경
- 확립된 AWS Control Tower 환경. CloudFormation 템플릿이 관리 계정 내에 배포되므로 관리 계정을 적절하게 구성해야 합니다.
- AWS 관리 계정에 필요한 권한. S3 버킷, AWS Lambda 함수, AWS Identity and Access Management (IAM) 역할, 프로젝트 등의 리소스를 생성하고 관리하려면 충분한 권한이 필요합니다. AWS CodePipeline
- 테라폼에 익숙해야 합니다. 배포에는 Terraform 구성 생성 및 관리가 포함되므로 Terraform의 핵심 개념과 워크플로를 이해하는 것이 중요합니다.

제한 사항

- 계정의 [AWS 리소스 할당량을 확인하세요](#). 배포로 인해 여러 리소스가 생성될 수 있으며 서비스 할당량이 발생하면 배포 프로세스가 방해될 수 있습니다.
- 템플릿은 특정 버전의 Terraform 및 용으로 설계되었습니다. AWS 서비스버전을 업그레이드하거나 변경하려면 템플릿 수정이 필요할 수 있습니다.

제품 버전

- 테라폼 버전 1.5.7 이상
- AFT 버전 1.1.1 이상

아키텍처

대상 기술 스택

- AWS CloudFormation
- AWS CodeBuild
- AWS CodeCommit
- AWS CodePipeline
- 아마존 EventBridge
- IAM
- AWS Lambda
- Amazon S3

대상 아키텍처

다음 다이어그램은 이 패턴에서 설명하는 구현을 보여줍니다.

워크플로는 리소스 생성, 콘텐츠 생성, 파이프라인 실행이라는 세 가지 주요 작업으로 구성됩니다.

리소스 생성

[이 패턴과 함께 제공되는 CloudFormation 템플릿](#)은 템플릿을 배포할 때 선택한 매개 변수에 따라 필요한 모든 리소스를 생성하고 설정합니다. 템플릿은 최소한 다음과 같은 리소스를 생성합니다.

- AFT Terraform 부트스트랩 코드를 저장하는 CodeCommit 저장소입니다.
- AFT 구현과 관련된 테라폼 상태 파일을 저장하는 S3 버킷
- 파이프라인 CodePipeline
- Terraform 계획을 구현하고 파이프라인의 여러 단계에서 명령을 적용하는 두 개의 CodeBuild 프로젝트
- IAM 역할 및 서비스 CodeBuild CodePipeline
- 파이프라인 런타임 아티팩트를 저장하는 두 번째 S3 버킷
- 브랜치의 CodeCommit 리포지토리 변경 사항을 캡처하기 위한 EventBridge 규칙 main
- 규칙의 또 다른 IAM 역할 EventBridge

또한 템플릿의 `Generate AFT Files` 파라미터를 `true` 설정하면 CloudFormation 템플릿에서 다음과 같은 추가 리소스를 생성하여 콘텐츠를 생성합니다.

- 생성된 콘텐츠를 저장하고 CodeCommit 리포지토리의 소스로 사용하기 위한 S3 버킷
- 지정된 파라미터를 처리하고 적절한 콘텐츠를 생성하는 Lambda 함수
- Lambda 함수를 실행하기 위한 IAM 함수
- 템플릿이 배포될 때 Lambda 함수를 실행하는 CloudFormation 사용자 지정 리소스

콘텐츠 생성

AFT 부트스트랩 파일과 해당 콘텐츠를 생성하기 위해 솔루션은 Lambda 함수와 S3 버킷을 사용합니다. 이 함수는 버킷에 폴더를 생성한 다음 폴더 내에 두 개의 파일 (및) 을 생성합니다. `main.tf`

backend.tf 또한 함수는 제공된 CloudFormation 파라미터를 처리하고 이러한 파일을 사전 정의된 코드로 채워 각 파라미터 값을 대체합니다.

[파일을 생성하는 데 템플릿으로 사용되는 코드를 보려면 솔루션의 GitHub 리포지토리를 참조하십시오.](#) 기본적으로 파일은 다음과 같이 생성됩니다.

main.tf

```
module "aft" {
  source = "github.com/aws-ia/terraform-aws-control_tower_account_factory?
  ref=<aft_version>"

  # Required variables
  ct_management_account_id = "<ct_management_account_id>"
  log_archive_account_id   = "<log_archive_account_id>"
  audit_account_id         = "<audit_account_id>"
  aft_management_account_id = "<aft_management_account_id>"
  ct_home_region           = "<ct_home_region>"

  # Optional variables
  tf_backend_secondary_region = "<tf_backend_secondary_region>"
  aft_metrics_reporting       = "<false|true>"

  # AFT Feature flags
  aft_feature_cloudtrail_data_events      = "<false|true>"
  aft_feature_enterprise_support          = "<false|true>"
  aft_feature_delete_default_vpcs_enabled = "<false|true>"

  # Terraform variables
  terraform_version      = "<terraform_version>"
  terraform_distribution = "<terraform_distribution>"
}
```

백엔드.tf

```
terraform {
  backend "s3" {
    region = "<aft-main-region>"
    bucket = "<s3-bucket-name>"
    key    = "aft-setup.tfstate"
  }
}
```

```
}
```

CodeCommit 리포지토리 생성 중에 Generate AFT Files 파라미터를 로 true 설정하면 템플릿은 생성된 콘텐츠가 포함된 S3 버킷을 main 브랜치의 소스로 사용하여 리포지토리를 자동으로 채웁니다.

파이프라인 실행

리소스가 생성되고 부트스트랩 파일이 구성되면 파이프라인이 실행됩니다. 첫 번째 단계 (소스) 는 리포지토리의 메인 브랜치에서 소스 코드를 가져오고, 두 번째 단계 (Build) 는 Terraform plan 명령을 실행하고 검토할 결과를 생성합니다. 세 번째 단계 (승인) 에서 파이프라인은 마지막 단계 (배포) 를 승인하거나 거부하기 위한 수동 조치를 기다립니다. 마지막 단계에서 파이프라인은 이전 Terraform apply 명령의 결과를 입력으로 사용하여 Terraform 명령을 실행합니다. plan 마지막으로, 교차 계정 역할과 관리 계정의 권한을 사용하여 AFT 관리 계정에서 AFT 리소스를 생성합니다.

도구

서비스

- [AWS CloudFormation](#) AWS 리소스를 설정하고, 빠르고 일관되게 프로비저닝하고, 전체 AWS 계정 및 지역의 수명 주기 전반에 걸쳐 리소스를 관리할 수 있도록 도와줍니다.
- [AWS CodeBuild](#) 소스 코드를 컴파일하고, 단위 테스트를 실행하고, 배포할 준비가 된 아티팩트를 생성하는 데 도움이 되는 완전관리형 빌드 서비스입니다.
- [AWS CodeCommit](#) 자체 소스 제어 시스템을 관리할 필요 없이 Git 리포지토리를 비공개로 저장하고 관리할 수 있는 버전 제어 서비스입니다.
- [AWS CodePipeline](#) 소프트웨어 릴리스의 여러 단계를 신속하게 모델링 및 구성하고 소프트웨어 변경 사항을 지속적으로 릴리스하는 데 필요한 단계를 자동화할 수 있도록 도와줍니다.
- [AWS Lambda](#) 이벤트에 응답하여 코드를 실행하고 컴퓨팅 리소스를 자동으로 관리하는 컴퓨팅 서비스로, 프로덕션용 최신 서버리스 애플리케이션을 빠르게 만들 수 있는 방법을 제공합니다.
- [AWS SDK for Python \(Boto3\)](#) Python 애플리케이션, 라이브러리 또는 스크립트를 AWS 서비스와 통합하는 데 도움이 되는 소프트웨어 개발 키트입니다.

기타 도구

- [Terraform](#)은 인프라를 안전하고 효율적으로 구축, 변경 및 버전 관리할 수 있게 해주는 코드형 인프라 (IaC) 도구입니다. 여기에는 컴퓨팅 인스턴스, 스토리지, 네트워킹과 같은 하위 수준 구성 요소와 DNS 항목 및 SaaS 기능과 같은 상위 수준 구성 요소가 포함됩니다.

- **Python**은 배우기 쉽고 강력한 프로그래밍 언어입니다. 효율적인 고급 데이터 구조를 가지고 있으며 객체 지향 프로그래밍에 대한 간단하지만 효과적인 접근 방식을 제공합니다.

코드 리포지토리

이 패턴의 코드는 GitHub [AFT 부트스트랩](#) 파이프라인 리포지토리에서 사용할 수 있습니다.

공식 AFT 리포지토리에 대해서는 의 [테라폼용 AWS Control Tower Account Factory](#)를 참조하십시오.
GitHub

모범 사례

제공된 CloudFormation 템플릿을 사용하여 AFT를 배포할 때는 안전하고 효율적이며 성공적인 구현을 보장하는 데 도움이 되는 모범 사례를 따르는 것이 좋습니다. AFT를 구현하고 운영하기 위한 주요 지침 및 권장 사항은 다음과 같습니다.

- 매개변수 철저한 검토: CloudFormation 템플릿의 각 매개변수를 주의 깊게 검토하고 이해하십시오. AFT의 올바른 설정과 기능을 위해서는 정확한 파라미터 구성이 매우 중요합니다.
- 정기 템플릿 업데이트: 템플릿을 최신 AWS 기능 및 Terraform 버전으로 업데이트하십시오. 정기 업데이트는 새로운 기능을 활용하고 보안을 유지하는 데 도움이 됩니다.
- 버전 관리: AFT 모듈 버전을 고정하고 가능하면 별도의 AFT 배포를 사용하여 테스트하십시오.
- 범위: AFT는 인프라 가드레일과 사용자 지정을 배포할 때만 사용하십시오. 애플리케이션을 배포하는 데 사용하지 마십시오.
- 린팅 및 검증: AFT 파이프라인에는 린트 및 검증을 거친 Terraform 구성이 필요합니다. 컨피그레이션을 AFT 리포지토리로 푸시하기 전에 Lint를 실행하고, 검증하고, 테스트하세요.
- Terraform 모듈: 재사용 가능한 Terraform 코드를 모듈로 빌드하고 항상 조직의 요구 사항에 맞게 Terraform 및 공급자 버전을 지정하십시오. AWS

에픽

환경 설정 및 구성 AWS

작업	설명	필요한 기술
AWS Control Tower 환경을 준비하십시오.	AWS 환경에 설치 및 AWS Control Tower 구성하여 중앙	클라우드 관리자

작업	설명	필요한 기술
	집중식 관리 및 거버넌스를 보장하십시오 AWS 계정. 자세한 내용은 AWS Control Tower 설명서의 AWS Control Tower 시작하기 를 참조하십시오.	
AFT 관리 계정을 실행하세요.	AWS Control Tower Account Factory를 사용하여 AFT 관리 계정으로 사용할 새 AWS 계정 계정을 개설하십시오. 자세한 내용은 AWS Control Tower 설명서의 Account AWS Service Catalog Factory 를 통한 계정 프로비저닝 을 참조하십시오.	클라우드 관리자

관리 계정에 CloudFormation 템플릿을 배포하십시오.

작업	설명	필요한 기술
CloudFormation 템플릿을 실행합니다.	이 에픽에서는 이 솔루션과 함께 제공된 CloudFormation 템플릿을 배포하여 AWS 관리 계정에 AFT 부트스트랩 파이프라인을 설정합니다. 파이프라인은 이전 에픽에서 설정한 AFT 관리 계정에 AFT 솔루션을 배포합니다. 1단계: 콘솔 열기 AWS CloudFormation • 에 AWS Management Console 로그인하고 AWS CloudFormation 콘솔 을 엽니다. 올바른 AWS Control	클라우드 관리자

작업	설명	필요한 기술
	<p>Tower 주 지역 내에서 운영하고 있는지 확인하세요.</p> <p>2단계: 새 스택 생성</p> <ol style="list-style-type: none"> 1. 새 스택 생성을 선택합니다. 2. 템플릿 파일을 업로드하는 옵션을 선택하고 이 패턴과 함께 제공된 CloudFormation 템플릿을 업로드합니다. <p>3단계: 스택 파라미터 구성</p> <ul style="list-style-type: none"> • Repository Name : AFT 부트스트랩 모듈을 저장하기 위한 리포지토리 이름을 지정합니다. • Branch Name: 소스 리포지토리 브랜치를 지정합니다. • CodeBuild Docker Image: CodeBuild Docker 기본 이미지로 사용할 파일을 선택합니다. <p>4단계: 파일 생성 결정</p> <ul style="list-style-type: none"> • Generate AFT Files 매개변수는 기본 AFT 배포 파일을 생성할지 여부를 제어합니다. 이 파라미터를 다음과 같이 설정합니다. 	

작업	설명	필요한 기술
	<ul style="list-style-type: none"> • true지정된 리포지토리에 AFT 배포 파일을 자동으로 만들고 저장합니다. • false파일 생성을 수동으로 처리하거나 파일을 이미 제자리에 두고 싶은 경우 <p>선택한 경우 8단계로 이동하고 false, 그렇지 않으면 5~7단계를 먼저 수행하십시오.</p> <p>5단계: AFT 계정 세부 정보 입력 AWS Control Tower 및</p> <ul style="list-style-type: none"> • 입력 AWS Control Tower 및 AFT 계정별 정보: <ul style="list-style-type: none"> • Log Archive Account ID: 내 로그 아카이브 계정 ID의 ID. AWS Control Tower • Audit Account ID: 에 있는 감사 계정의 ID AWS Control Tower. • AFT Management Account ID: 첫 번째 에 픽에서 만든 AFT 관리 계정의 ID입니다. • AFT Main Region및AFT Secondary Region: AFT 배포를 AWS 리전 위한 기본 및 보조. 	

작업	설명	필요한 기술
	<p>6단계: AFT 옵션 구성</p> <ul style="list-style-type: none"> • 지표 보고 설정: <ul style="list-style-type: none"> • AFT Enable Metrics Reporting : AFT 지표 보고를 활성화 또는 비활성화합니다. 자세한 내용은 AWS Control Tower 설명서의 운영 지표를 참조 하십시오. • AFT 기능 옵션 설정: <ul style="list-style-type: none"> • Enable AFT CloudTrail Data Events: 모든 AFT 관리 계정에서 CloudTrail 데이터 이벤트를 활성화합니다. 자세한 내용은 AWS Control Tower 설명서의 AWS CloudTrail 데이터 이벤트를 참조 하십시오. • Enable AFT Enterprise Support : 모든 AFT 관리 계정에서 엔터프라이즈 지원을 활성화합니다. 자세한 내용은 AWS Control Tower 설명서의 AWS Enterprise Support 플랜을 참조 하십시오. • Enable AFT Delete Default VPC: AFT 관리 계정의 모든 VPC만 삭제하십시오. 자세한 내용 	

작업	설명	필요한 기술
	<p>은 설명서의 AWS 기본 VPC 삭제를 참조하십시오. AWS Control Tower</p> <p>7단계: 버전 지정</p> <ul style="list-style-type: none"> AFT Terraform Version: AFT 파이프라인에서 사용할 Terraform 버전을 선택합니다. AFT Version: 배포할 AFT 버전을 정의합니다. 최신 AFT 버전을 사용하려면 기본 설정 (latest) 을 유지하십시오. <p>8단계: 스택 검토 및 생성</p> <ul style="list-style-type: none"> 모든 파라미터와 설정을 검토하세요. 모든 것이 정상이면 스택 생성을 진행하세요. <p>9단계: 스택 생성 모니터링</p> <ul style="list-style-type: none"> AWS CloudFormation 정의한 리소스를 프로비저닝하고 구성합니다. 콘솔에서 스택 생성 프로세스를 모니터링합니다. CloudFormation 이 프로세스는 몇 분 정도 걸릴 수 있습니다. <p>10단계: 배포 확인</p>	

작업	설명	필요한 기술
	<ul style="list-style-type: none"> • 스택 상태가 CREATE_COMPLETE로 표시되면 모든 리소스가 올바르게 생성되었는지 확인하십시오. • 출력 섹션에서 값을 기록해 둡니다. Terraform BackendBucketName 	

AFT 부트스트랩 리포지토리와 파이프라인을 채우고 검증합니다.

작업	설명	필요한 기술
<p>AFT 부트스트랩 저장소를 채웁니다.</p>	<p>(선택 사항) CloudFormation 템플릿을 배포한 후 새로 만든 AFT 부트스트랩 리포지토리의 콘텐츠를 채우거나 검증하고 파이프라인이 성공적으로 실행되었는지 테스트할 수 있습니다.</p> <p>Generate AFT Files파라미터를 로 true 설정한 경우 다음 스토리 (파이프라인 검증)로 넘어가세요.</p> <p>1단계: 리포지토리 채우기</p> <ol style="list-style-type: none"> 1. AWS CodeCommit 콘솔을 열고 새로 만든 리포지토리를 선택합니다. 기본 이름을 유지한 경우 리포지토리가 호출됩니다aft-setup . 2. SSH, HTTPS 또는 HTTPS (GRC) 를 사용하여 로컬 시 	<p>클라우드 관리자</p>

작업	설명	필요한 기술
	<p>시스템에 리포지토리를 복제하고 편집기에서 엽니다.</p> <p>3. 라는 폴더를 만들고 그 안에 빈 파일 두 개를 생성합니다: terraform 및 backend.tf main.tf</p> <p>4. backend.tf 파일을 열고 다음 코드 스니펫을 추가합니다.</p> <pre data-bbox="630 682 1029 1121"> terraform { backend "s3" { region = "<aft-main-region>" bucket = "<s3-bucket-name>" key = "aft-setup" } } </pre> <p>파일에서:</p> <ul style="list-style-type: none"> • 기본 AFT 영역으로 <aft-main-region> 교체하십시오. AWS Control Tower 기본 지역과 일치해야 합니다. • Terraform 백엔드 버킷의 <s3-bucket-name> 이름으로 바꾸십시오. 이전에 배포한 CloudFormation 템플릿에서 생성된 TerraformBackendBucketName 출력에서 이를 찾을 수 있습니다. 	

작업	설명	필요한 기술
	<p>5. <code>main.tf</code> 파일을 열고 AFT 리포지토리에 있는 예제 중 하나를 사용하여 AFT를 배포하십시오. 예를 들어, 선호하는 버전 제어 시스템 (VCS) 공급자 (CodeCommit GitHub, 또는 Bitbucket) 와 협력하거나 AFT VPC를 사용자 지정할 수 있습니다. 추가 AFT 입력 옵션은 AFT 저장소의 README 파일을 참조하십시오.</p> <p>2단계: 변경 사항 커밋 및 푸시</p> <ul style="list-style-type: none"> 폴더와 파일을 만들고 채운 후 변경 사항을 확인하고 코드를 리포지토리에 업로드합니다. 파이프라인은 자동으로 시작되어 소스 및 빌드 단계를 거친 다음 배포 단계 전에 승인 작업이 완료될 때까지 기다립니다. 	

작업	설명	필요한 기술
<p>AFT 부트스트랩 파이프라인을 검증하세요.</p>	<p>1단계: 파이프라인 보기</p> <ul style="list-style-type: none"> • CodePipeline 콘솔을 열고 <code>aft-bootstrap-pipeline</code> 파이프라인이 성공적으로 시작되었는지 확인합니다. Terraform 플랜을 실행 중이거나 수동 승인 작업을 기다리는 중이어야 합니다. <p>2단계: 테라폼 계획 결과 승인</p> <ul style="list-style-type: none"> • 빌드 단계의 실행 로그를 보고 Terraform 계획의 결과를 검토한 다음 승인 단계에서 실행을 승인하거나 거부할 수 있습니다. 승인하면 파이프라인이 제공된 AFT 관리 계정에 AFT 리소스 배포를 시작합니다. <p>3단계: 배포가 완료될 때까지 기다립니다.</p> <ul style="list-style-type: none"> • 파이프라인이 성공적으로 실행될 때까지 기다리세요. 약 30분이 소요됩니다. 발생할 수 있는 모든 오류는 API 할당량 때문에 발생하는 경우가 많습니다. 이러한 경우 파이프라인을 다시 실행하여 배포를 계속할 수 있습니다. <p>4단계: 생성된 리소스 확인</p>	<p>클라우드 관리자</p>

작업	설명	필요한 기술
	<ul style="list-style-type: none"> AFT 관리 계정에 액세스하여 리소스가 생성되었는지 확인합니다. 	

문제 해결

문제	Solution
<p>템플릿에 포함된 사용자 지정 Lambda 함수는 배포 중에 CloudFormation 실패합니다.</p>	<p>Amazon CloudWatch 로그에서 Lambda 함수를 확인하여 오류를 식별하십시오. 로그는 자세한 정보를 제공하며 특정 문제를 정확히 찾아내는데 도움이 될 수 있습니다. Lambda 함수에 필요한 권한이 있고 환경 변수가 올바르게 설정되었는지 확인합니다.</p>
<p>부적절한 권한으로 인해 리소스 생성 또는 관리에 장애가 발생했습니다.</p>	<p>Lambda 함수와 CodeBuild 배포와 관련된 기타 서비스에 연결된 IAM 역할 및 정책을 검토하십시오. 이들에게 필요한 권한이 있는지 확인하십시오. 권한 문제가 있는 경우 IAM 정책을 조정하여 필요한 액세스 권한을 부여하십시오.</p>
<p>최신 버전 AWS 서비스 또는 Terraform 버전에서 오래된 버전의 CloudFormation 템플릿을 사용하고 있습니다.</p>	<p>최신 AWS 및 Terraform 릴리스와 호환되도록 CloudFormation 템플릿을 정기적으로 업데이트하세요. 버전별 변경 사항 또는 요구 사항은 릴리스 노트 또는 설명서를 확인하세요.</p>
<p>배포 중에 AWS 서비스 할당량에 도달할 수 있습니다.</p>	<p>파이프라인을 배포하기 전에 S3 버킷, IAM 역할, Lambda 함수와 같은 리소스의 AWS 서비스 할당량을 확인하십시오. 필요한 경우 요청이 증가합니다. 자세한 내용은 웹 사이트의 AWS 서비스 할당량을 참조하십시오. AWS</p>
<p>템플릿의 잘못된 입력 매개변수로 인해 오류가 발생했습니다. CloudFormation</p>	<p>모든 입력 매개 변수에 오타나 잘못된 값이 있는지 다시 확인하십시오. 계정 ID 및 지역 이름과 같은 리소스 식별자가 정확한지 확인하세요.</p>

관련 리소스

이 패턴을 성공적으로 구현하려면 다음 리소스를 검토하세요. 이러한 리소스는 를 사용하여 AWS CloudFormation AFT를 설정하고 관리하는 데 매우 유용할 수 있는 추가 정보와 지침을 제공합니다.

AWS문서:

- [AWS Control Tower 사용 설명서에는](#) 설정 및 관리에 대한 자세한 정보가 나와 AWS Control Tower 있습니다.
- [AWS CloudFormation 설명서는](#) CloudFormation 템플릿, 스택 및 리소스 관리에 대한 통찰력을 제공합니다.

IAM 정책 및 모범 사례:

- [IAM의 보안 모범 사례는 IAM](#) 역할 및 정책을 사용하여 AWS 리소스를 보호하는 방법을 설명합니다.

테라폼 기반: AWS

- [Terraform AWS 공급자 설명서는](#) Terraform과 함께 사용하는 방법에 대한 포괄적인 정보를 제공합니다. AWS

AWS 서비스 할당량:

- [AWS 서비스 할당량은 할당량을](#) 보는 방법과 증가를 요청하는 방법에 대한 정보를 제공합니다. AWS 서비스

여러 AWS 계정 및 AWS 리전의 AWS Service Catalog 제품을 관리

작성자: Ram Kandaswamy(AWS)

환경: 프로덕션

기술: 관리 및 거버넌스, 클라우드 네이티브, 인프라, 현대화

워크로드: 기타 모든 워크로드

AWS 서비스: AWS 서비스 카탈로그, AWS CloudFormation

요약

Amazon Web Services(AWS) Service Catalog는 엔터프라이즈용 코드형 인프라(IaC) 템플릿의 거버넌스 및 배포를 간소화하고 가속화합니다. AWS CloudFormation 템플릿을 사용하여 제품에 필요한 AWS 리소스 (스택) 컬렉션을 정의합니다. AWS는 한 번의 작업으로 여러 계정 및 AWS 리전에 걸쳐 스택을 생성, 업데이트 또는 삭제할 수 있도록 하여 이 기능을 CloudFormation StackSets 확장합니다.

AWS Service Catalog 관리자는 개발자가 작성한 CloudFormation 템플릿을 사용하여 제품을 만들고 게시합니다. 그런 다음 이러한 제품은 포트폴리오와 연결되며, 거버넌스를 위해 제약이 적용됩니다. 다른 AWS 계정이나 조직 단위(OU)의 사용자가 제품을 사용할 수 있도록 하려면 일반적으로 [포트폴리오를 공유](#)합니다. 이 패턴은 AWS를 기반으로 하는 AWS Service Catalog 제품 오퍼링을 관리하기 위한 대체 접근 방식을 설명합니다 CloudFormation StackSets. 포트폴리오를 공유하는 대신 스택 세트 제약 조건을 사용하여 제품을 배포하고 사용할 수 있는 AWS 리전 및 계정을 설정할 수 있습니다. 이 접근 방식을 사용하면 거버넌스 요구 사항을 충족하면서 여러 계정, OU 및 AWS 리전에서 AWS Service Catalog 제품을 프로비저닝하고 중앙 위치에서 관리할 수 있습니다.

이 접근 방식의 이점:

- 제품은 기본 계정에서 프로비저닝되고 관리되며 다른 계정과 공유되지 않습니다.
- 이 접근 방식을 사용하면 특정 제품을 기반으로 프로비저닝된 모든 제품(스택)을 통합적으로 볼 수 있습니다.
- AWS Service Management Connector를 사용한 구성은 하나의 계정만 대상으로 하므로 더 쉽습니다.
- AWS Service Catalog에서 제품을 쿼리하고 사용하는 것이 더 쉽습니다.

사전 조건 및 제한 사항

사전 조건

- IaC 및 버전 관리용 AWS CloudFormation 템플릿
- AWS 리소스 프로비저닝 및 관리를 위한 다중 계정 설정 및 AWS Service Catalog

제한 사항

- 이 접근 방식은 AWS를 CloudFormation StackSets 사용하며 StackSets 적용 제한은 다음과 같습니다.
 - StackSets 매크로를 통한 CloudFormation 템플릿 배포는 지원하지 않습니다. 매크로를 사용하여 템플릿을 사전 처리하는 경우 StackSets 기반 배포를 사용할 수 없습니다.
 - StackSets 스택 세트에서 스택을 분리할 수 있는 기능을 제공하므로 특정 스택을 대상으로 문제를 해결할 수 있습니다. 하지만 연결이 끊긴 스택은 스택 세트와 다시 연결할 수 없습니다.
- AWS Service Catalog는 StackSet 이름을 자동으로 생성합니다. 사용자 지정은 현재 지원되지 않습니다.

아키텍처

대상 아키텍처

1. 사용자는 AWS CloudFormation 템플릿을 생성하여 JSON 또는 YAML 형식으로 AWS 리소스를 프로비저닝합니다.
2. CloudFormation 템플릿은 AWS Service Catalog에 제품을 생성하여 포트폴리오에 추가합니다.
3. 사용자가 프로비저닝된 제품을 생성하여 대상 계정에 CloudFormation 스택을 생성합니다.
4. 각 스택은 템플릿에 지정된 리소스를 제공합니다. CloudFormation

도구

서비스

- [AWS는 AWS CloudFormation](#) 리소스를 설정하고, 빠르고 일관되게 프로비저닝하고, 전체 AWS 계정 및 지역의 수명 주기 전반에 걸쳐 리소스를 관리할 수 있도록 지원합니다.
- [AWS Command Line Interface\(AWS CLI\)](#)는 명령줄 셸에서 명령을 사용하여 AWS 서비스와 상호 작용할 수 있는 오픈 소스 도구입니다.
- [AWS Identity and Access Management\(IAM\)](#)를 사용하면 사용자에게 대해 인증 및 권한 부여를 제어함으로써 AWS 리소스에 대한 액세스를 안전하게 관리할 수 있습니다.
- [AWS Service Catalog](#)를 사용하면 AWS에 승인된 IT 서비스의 카탈로그를 중앙에서 관리할 수 있습니다. 최종 사용자는 조직에서 규정한 제약에 따라, 필요에 따라 승인된 IT 서비스만 신속하게 배포할 수 있습니다.

에픽

전 계정에 제품 프로비저닝

작업	설명	필요한 기술
<p>포트폴리오를 생성합니다.</p>	<p>포트폴리오는 특정 기준에 따라 그룹화된 하나 이상의 제품을 포함하는 컨테이너입니다. 제품에 포트폴리오를 사용하면 제품 세트 전체에 공통 제약 조건을 적용할 수 있습니다.</p> <p>키 페어를 생성하려면 AWS 설명서의 지침을 따르세요. 다음은 AWS CLI를 사용하는 경우의 명령 예제입니다.</p> <pre data-bbox="594 1503 1029 1745">aws servicecatalog create-portfolio -- provider-name my-provider --display-name my-portfolio</pre> <p>자세한 내용은 AWS CLI 설명서를 참조하세요.</p>	<p>AWS Service Catalog, IAM</p>

작업	설명	필요한 기술
<p>CloudFormation 템플릿 생성.</p>	<p>리소스를 설명하는 CloudFormation 템플릿을 만드세요. 해당하는 경우 리소스 속성 값을 파라미터화해야 합니다.</p>	<p>AWS CloudFormation, JSON/YAML</p>
<p>버전 정보를 사용하여 제품을 생성하세요.</p>	<p>CloudFormation 템플릿은 AWS Service Catalog에 게시하면 제품이 됩니다. 버전 제목 및 설명과 같은 선택적 버전 세부 파라미터의 값을 제공하십시오. 이렇게 하면 나중에 제품을 쿼리할 때 유용합니다.</p> <p>키 페어를 생성하려면 AWS Service Catalog 설명서의 지침을 따르세요. AWS CLI를 사용하는 경우 명령 예제는 다음과 같습니다.</p> <pre data-bbox="592 1079 1027 1318">aws servicecatalog create-product --cli- input-json file://cr eate-product-input .json</pre> <p>create-product-input.json 은 제품의 파라미터를 전달하는 파일입니다. 참고: 예제 파일은 이 패턴의 추가 정보 섹션을 참조하세요. 자세한 내용은 AWS CLI 설명서를 참조하세요.</p>	<p>AWS Service Catalog</p>

작업	설명	필요한 기술
<p>제약 조건을 적용합니다.</p>	<p>포트폴리오에 스택 세트 제약을 적용하여 여러 AWS 계정, 리전 및 권한과 같은 제품 배포 옵션을 구성할 수 있습니다. 지침은 AWS Service Catalog 설명서를 참조하세요.</p>	<p>AWS Service Catalog</p>
<p>권한을 추가합니다.</p>	<p>사용자에게 포트폴리오에 있는 제품을 실행할 수 있는 권한을 부여합니다. 콘솔 지침은 AWS Service Catalog 설명서를 참조하세요. 다음은 AWS CLI를 사용하는 경우의 명령 예제입니다.</p> <pre data-bbox="594 905 1029 1339">aws servicecatalog associate-principal- with-portfolio \ --portfolio-id port-2s6abcdefwdh4 \ --principal-arn arn:aws:iam::44445 5556666:role/Admin \ --principal-type IAM</pre> <p>자세한 내용은 AWS CLI 설명서를 참조하세요.</p>	<p>AWS Service Catalog, IAM</p>

작업	설명	필요한 기술
<p>제품을 프로비저닝합니다.</p>	<p>프로비저닝한 제품은 제품의 리소스가 제공된 인스턴스입니다. CloudFormation 템플릿을 기반으로 제품을 프로비저닝하면 CloudFormation 스택과 기본 리소스가 시작됩니다.</p> <p>스택 세트 제약 조건에 따라 해당 AWS 리전 및 계정을 대상으로 하여 제품을 프로비저닝합니다. 다음은 CLI 명령의 예입니다.</p> <pre data-bbox="597 810 1027 1245">aws servicecatalog provision-product \ --product-id prod- abcdfz3syn2rg \ --provisioning- artifact-id pa-abc347 pcscfm \ --provisioned-product-name "mytestpp name3"</pre> <p>자세한 내용은 AWS CLI 설명서를 참조하세요.</p>	<p>AWS Service Catalog</p>

관련 리소스

참조

- [AWS Service Catalog 개요](#)
- [AWS 사용 CloudFormation StackSets](#)

자습서 및 동영상

- [AWS re:Invent 2019: 모든 것을 자동화: 옵션 및 모범 사례\(동영상\)](#)

추가 정보

create-product 명령을 사용할 때 cli-input-json 파라미터는 제품 소유자, 지원 이메일, CloudFormation 템플릿 세부 정보와 같은 정보를 지정하는 파일을 가리킵니다. 다음은 이러한 파일의 예입니다.

```
{
  "Owner": "Test admin",
  "SupportDescription": "Testing",
  "Name": "SNS",
  "SupportEmail": "example@example.com",
  "ProductType": "CLOUD_FORMATION_TEMPLATE",
  "AcceptLanguage": "en",
  "ProvisioningArtifactParameters": {
    "Description": "SNS product",
    "DisableTemplateValidation": true,
    "Info": {
      "LoadTemplateFromURL": "<url>"
    }
  },
  "Name": "version 1"
}
```

AWS Organizations의 AWS 멤버 계정을 AWS Control Tower로 마이그레이션

작성자: Rodolfo Jr. Cerrada(AWS)

환경: 프로덕션

기술: 관리 및 거버넌스, 현대화

AWS 서비스: AWS Organizations; AWS Control Tower

요약

이 패턴은 관리 계정으로 관리되는 멤버 계정인 AWS Organizations에서 AWS Control Tower로 Amazon Web Services(AWS) 계정을 마이그레이션하는 방법을 설명합니다. 계정을 AWS Control Tower에 등록하면 계정 거버넌스를 간소화하는 예방 및 탐지 가드레일과 기능을 활용할 수 있습니다. 또한 AWS Organizations 관리 계정이 침해된 경우, 회원 계정을 AWS Control Tower가 관리하는 새 조직으로 마이그레이션하고 싶을 수도 있습니다.

AWS Control Tower는 AWS Organizations를 비롯한 여러 다른 AWS 서비스의 기능을 결합 및 통합하는 프레임워크를 제공하며, 다중 계정 환경 전반에 걸쳐 일관된 규정 준수 및 거버넌스를 보장합니다. AWS Control Tower를 사용하면 AWS Organizations의 기능을 확장하는 일련의 규정된 규칙 및 정의를 따를 수 있습니다. 예를 들어 가드레일을 사용하여 보안 로그와 필요한 크로스 계정 액세스 권한이 생성되고 변경되지 않도록 할 수 있습니다.

사전 조건 및 제한 사항

사전 조건

- 활성 상태의 AWS 계정
- AWS Organizations의 대상 조직에 AWS Control Tower 설치(지침은 [AWS Control Tower 설명서의 설정 참조](#))
- AWS Control Tower의 관리자 자격 증명 (AWSControlTowerAdmins그룹 구성원)
- 소스 AWS 계정의 관리자 보안 인증

제한 사항

- AWS Organizations의 소스 관리 계정은 AWS Control Tower의 대상 관리 계정과 달라야 합니다.

제품 버전

- AWS Control Tower 버전 2.3(2020년 2월) 이상([출시 정보](#) 참조)

아키텍처

다음 그림은 마이그레이션 프로세스와 참조 아키텍처를 보여 줍니다. 이 패턴은 소스 조직의 AWS 계정을 AWS Control Tower가 관리하는 대상 조직으로 마이그레이션합니다.

등록 프로세스는 세 단계로 구성됩니다.

1. 계정은 AWS Organizations의 소스 조직을 떠납니다.
2. 계정은 독립형 계정이 됩니다. 즉, 어떤 조직에도 속하지 않으므로 거버넌스와 청구는 계정 관리자가 독립적으로 관리합니다.
3. 대상 조직은 해당 계정이 조직에 가입하도록 초대장을 보냅니다.
4. 독립형 계정은 초대를 수락하고 대상 조직의 구성원이 됩니다.
5. 계정이 AWS Control Tower에 등록되고 등록된 조직 유닛(OU)으로 이동됩니다. (AWS Control Tower 대시보드를 확인하여 등록을 확인하는 것이 좋습니다.) 이때 등록된 OU에서 활성화된 모든 가드레일이 적용됩니다.

도구

서비스

- [AWS Organizations](#)는 여러 AWS 계정을 사용자가 생성하고 중앙에서 관리하는 단일 엔터티(조직)로 통합할 수 있는 계정 관리 서비스입니다.
- [AWS Control Tower](#)는 AWS Organizations, AWS IAM Identity Center(AWS Single Sign-On의 후속) 및 AWS Service Catalog를 비롯한 다른 서비스의 기능을 통합하여 AWS 클라우드의 모든 조직과 계정에 걸쳐 보안, 운영 및 규정 준수에 대한 거버넌스 규칙을 대규모로 시행하고 관리할 수 있도록 지원합니다.

에픽

소스 조직에서 멤버 계정 제거

작업	설명	필요한 기술
<p>멤버 계정을 독립형 계정으로 운영할 수 있는지 확인하십시오.</p>	<p>소스 조직을 탈퇴할 구성원 계정이 독립형 계정으로 작동하는 데 필요한 정보를 가지고 있는지 확인하십시오. 예를 들어, 회원 계정에 결제 정보가 없으면 독립형 계정으로 운영할 수 없습니다. 계정이 조직에 연결되어 있지 않을 때 발생하는 청구 가능한 AWS activity에 대해 AWS는 결제 정보를 사용하여 요금을 청구하기 때문입니다.</p> <p>일반적으로 AWS Organizations 콘솔, API 또는 AWS 명령줄 인터페이스(CLI) 명령을 사용하여 멤버 계정을 생성한 경우 독립형 계정에 필요한 정보는 자동으로 수집되지 않습니다. 이 정보를 추가하려면 계정에 로그인하고 지원 계획, 연락처 정보, 결제 방법을 지정하십시오.</p> <p>조직에서 계정을 제거하기 전에 알아야 할 사항에 대한 자세한 내용은 AWS Organizations 설명서의 조직에서 계정을 제거하기 전에를 참조하십시오.</p>	<p>계정 관리자</p>
<p>소스 조직에서 멤버 계정을 제거합니다.</p>	<p>AWS Organizations 설명서의 지침에 따라 조직에서 멤버 계정을 제거합니다. 조직의 관리</p>	<p>관리 계정 관리자 또는 계정 관리자</p>

작업	설명	필요한 기술
	<p>계정에 로그인하여 멤버 계정을 제거하거나, 멤버 계정에 로그인하여 조직을 탈퇴할 수 있습니다.</p> <p>계정을 제거하거나 탈퇴할 수 있는 관리자 수준의 보안 인증이 없는 경우 조직 관리자에게 도움을 요청하십시오.</p> <p>회원 계정에 지원 플랜, 연락처 정보 또는 결제 정보가 누락된 경우 해당 정보를 제공하고 확인하라는 메시지가 표시됩니다.</p> <p>조직을 탈퇴하는 경우 AWS Organizations 콘솔의 시작하기 페이지로 리디렉션되어 다른 조직에 대한 계정 가입 초대를 확인할 수 있습니다.</p> <p>중요: 현재 계정은 독립형 계정입니다. AWS 프리 티어가 적용되지 않는 워크로드를 실행하는 경우 계정에 제공한 결제 및 청구 정보에 따라 요금이 부과됩니다.</p>	
<p>멤버 계정이 더 이상 소스 조직에 속하지 않는지 확인합니다.</p>	<p>AWS Organizations 콘솔에 더 이상 조직 탈퇴 버튼이 표시되지 않을 것입니다. 대신 다른 조직에서 보류 중인 초대가 있는 경우 이를 확인할 수 있습니다.</p>	<p>계정 관리자</p>

작업	설명	필요한 기술
<p>계정에 대한 액세스 권한을 부여하는 IAM 역할을 탈퇴하는 조직에서 제거합니다.</p>	<p>소스 조직에서 계정을 제거해도 AWS Organizations나 관리자가 생성한 AWS Identity and Access Management(IAM) 역할은 자동으로 삭제되지 않습니다. 소스 조직의 관리 계정에서 이 액세스를 종료하려면 IAM 역할을 수동으로 삭제해야 합니다. 자세한 내용은 IAM 사용 설명서에서 역할 또는 인스턴스 프로필 삭제를 참조하십시오.</p> <p>멤버 계정이 조직을 나가면 계정에 연결된 모든 태그가 삭제됩니다. 독립형 계정은 태그를 지원하지 않습니다.</p>	<p>계정 관리자</p>

계정을 초대하여 AWS Control Tower와 함께 새 조직에 가입

작업	설명	필요한 기술
<p>AWS Control Tower에 로그인합니다.</p>	<p>AWS Control Tower 콘솔에 관리자로 로그인합니다.</p> <p>현재 AWS 계정을 소스 조직에서 AWS Control Tower가 관리하는 OU의 조직으로 직접 이동할 수 있는 방법은 없습니다. 하지만 이미 AWS Control Tower에서 관리하는 OU에 계정을 등록하면 기존 AWS 계정으로 AWS Control Tower 거버넌스를 확장할 수 있습니다. 따라서 이 단계를 수행하려면 AWS</p>	<p>AWS Control Tower 관리자</p>

작업	설명	필요한 기술
	Control Tower에 로그인해야 합니다.	
멤버 계정을 초대합니다.	<ol style="list-style-type: none"> 1. AWS Organizations 콘솔에 로그인하고 AWS 계정 페이지로 이동합니다. 2. AWS 계정 추가 페이지에서 기존 AWS 계정 초대를 선택합니다. 3. 12자리 계정 번호(대시 제외)와 선택 사항으로 설명 및 태그를 포함한 계정 정보를 작성한 다음 초대 보내기를 선택합니다. <p>중요: 계정 이전으로 인해 애플리케이션 또는 네트워크 연결이 영향을 받지 않는지 확인하십시오.</p> <p>이 작업을 수행하면 멤버 계정 링크가 포함된 초대 이메일이 전송됩니다. 계정 관리자가 링크를 따라 초대를 수락하면 멤버 계정이 AWS 계정 페이지에 나타납니다. 더 자세한 내용은 AWS Organizations 설명서의 <u>조직에 가입할 AWS 계정 초대</u>를 참조하십시오.</p>	AWS Control Tower 관리자

작업	설명	필요한 기술
<p>애플리케이션 및 연결성을 테스트합니다.</p>	<p>멤버 계정이 새 조직에 등록되면 루트 내 OU에 표시됩니다. 또한 아직 OU에 등록된 AWS Control Tower에 등록되지 않았으므로 계정에 등록되지 않은 것으로 플래그가 지정되어 AWS Control Tower 콘솔에도 나타납니다.</p> <p>다음을 확인합니다.</p> <ul style="list-style-type: none"> • AWS Control Tower 대시보드를 확인하여 가드레일 위반이 있는지 확인하십시오. • 네트워크 연결(VPN 또는 AWS Direct Connect)을 확인하여 전송의 영향을 받지 않았는지 확인하십시오. • (애플리케이션 소유자) 이 계정과 연결된 애플리케이션을 테스트하여 예상대로 실행되는지, 계정 이전으로 인해 종속성이 영향을 받지 않았는지 확인합니다. 	<p>AWS Control Tower 관리자, 멤버 계정 관리자, 애플리케이션 소유자</p>

등록을 위해 계정을 준비하십시오.

작업	설명	필요한 기술
<p>가드레일을 검토하고 위반 사항을 수정하십시오.</p>	<p>대상 OU에 정의된 가드레일, 특히 예방적 가드레일을 검토하고 위반 사항을 수정하십시오.</p>	<p>AWS Control Tower 관리자, 멤버 계정 관리자</p>

작업	설명	필요한 기술
	<p>AWS Control Tower 랜딩 존을 설정할 때 여러 필수 예방 가드레일이 기본적으로 활성화됩니다. 비활성화할 수 없습니다. 계정을 등록하기 전에 이러한 필수 가드레일을 검토하고 멤버 계정을 (수동으로 또는 스크립트를 사용하여) 수정해야 합니다.</p> <p>참고: 예방적 가드레일은 AWS Control Tower 등록 계정이 규정을 준수하도록 유지하고 정책 위반을 방지합니다. 예방 가드레일을 위반하면 등록에 영향을 미칠 수 있습니다. 등록이 성공적으로 완료된 후 탐지 가드레일 위반이 감지되면 AWS Control Tower 대시보드에 표시됩니다. 등록 프로세스에는 영향을 주지 않습니다. 자세한 내용은 AWS 설명서의 AWS Control Tower의 가드레일을 참조하십시오.</p>	
<p>가드레일 위반을 수정한 후 연결 문제를 확인하십시오.</p>	<p>가드레일 위반을 수정하기 위해 특정 포트를 닫거나 서비스를 비활성화해야 하는 경우도 있습니다. 계정을 등록하기 전에 해당 포트 및 서비스를 사용하는 애플리케이션을 수정해야 합니다.</p>	<p>애플리케이션 소유자</p>

계정을 AWS Control Tower에 등록

작업	설명	필요한 기술
AWS Control Tower 콘솔에 로그인합니다.	AWS Control Tower에 대한 관리 권한이 있는 로그인 보안 인증을 사용하십시오. 루트 사용자(관리 계정) 보안 인증을 사용하여 AWS Organizations 계정을 등록하지 마십시오. 그러면 오류 메시지가 표시됩니다.	AWS Control Tower 관리자
계정을 등록하십시오.	<ol style="list-style-type: none"> 1. AWS Control Tower의 계정 팩토리 페이지에서 계정 등록을 선택합니다. 2. 등록하려는 계정과 연결된 이메일 주소, AWS Control Tower에 표시되는 표시 이름, IAM Identity Center 이메일 주소, 계정 소유자의 이름과 성, 계정을 등록하려는 OU를 비롯한 세부 정보를 입력합니다. IAM Identity Center 이메일 주소는 선호하는 사용자 이메일 주소입니다. 계정 이메일과 동일한 이메일 주소를 사용할 수 있습니다. 3. 계정 등록을 선택합니다. <p>자세한 내용은 AWS Control Tower 설명서의 기존 계정 등록을 참조하십시오.</p>	AWS Control Tower 관리자

등록 후 계정 확인

작업	설명	필요한 기술
계정을 확인하십시오.	AWS Control Tower에서 계정을 선택합니다. 방금 등록한 계정의 초기 상태는 등록 중입니다. 등록이 완료되면 상태가 등록됨으로 변경됩니다.	AWS Control Tower 관리자, 멤버 계정 관리자
가드레일 위반 여부를 확인하십시오.	OU에 정의된 가드레일은 등록된 멤버 계정에 자동으로 적용됩니다. AWS Control Tower 대시보드에서 위반 사항을 모니터링하고 그에 따라 수정하십시오. 자세한 내용은 AWS 설명서의 AWS Control Tower의 가드레일 을 참조하십시오.	AWS Control Tower 관리자, 멤버 계정 관리자

문제 해결

문제	Solution
다음과 같은 오류 메시지가 나타납니다. 알 수 없는 오류가 발생했습니다. 나중에 다시 시도하거나 AWS Support에 문의하십시오.	이 오류는 AWS Control Tower의 루트 사용자 보안 인증(관리 계정)을 사용하여 새 계정을 등록할 때 발생합니다. AWS Service Catalog는 계정 팩토리 포트폴리오 또는 제품을 루트 사용자에게 매핑할 수 없으므로 오류 메시지가 표시됩니다. 이 오류를 해결하려면 루트가 아닌 전체 액세스 권한을 가진 사용자(관리자) 보안 인증을 사용하여 새 계정을 등록하십시오. 관리자 사용자에게 관리 액세스 권한을 할당하는 방법에 대한 자세한 내용은 AWS IAM Identity Center(AWS Single Sign-On의 후속) 설명서의 시작하기 를 참조하십시오.

문제	Solution
AWS Control Tower 활동 페이지에는 치명적인 드리프트 발생 작업이 표시됩니다.	이 조치는 서비스의 드리프트 점검을 반영하며, AWS Control Tower 설정에 문제가 있다는 것을 의미하지는 않습니다. 아무 조치도 필요하지 않습니다.

관련 리소스

설명서

- [AWS Organizations 용어 및 개념](#) (AWS Organizations 설명서)
- [AWS Control Tower로 무엇입니까?](#) (AWS Control Tower 설명서)
- [조직에서 멤버 계정 제거](#) (AWS Organizations 설명서)
- [AWS Control Tower에서 관리자 계정 생성](#) (AWS Control Tower 설명서)

자습서 및 비디오

- [AWS Control Tower 워크숍](#) (자습형 워크숍)
- [AWS Control Tower란 무엇입니까?](#) (동영상)
- [AWS Control Tower의 사용자 프로비저닝](#) (동영상)
- [기존 조직에 AWS Control Tower 활성화](#) (동영상)

여러 AWS 계정에 공유된 Amazon Machine Image의 사용을 모니터링

작성자: Naveen Suthar(AWS)와 Sandeep Gawande(AWS)

코드 리포지토리: [cross-account-ami-auditing](#) -[테라폼 샘플](#)

환경: PoC 또는 파일럿

기술: 관리 및 거버넌스, 서버리스, 운영 DevOps

AWS 서비스: 아마존 DynamoDB, AWS Lambda, 아마존 EventBridge

요약

[Amazon Machine Image\(AMI\)](#)는 Amazon Web Services(AWS) 환경에서 Amazon Elastic Compute Cloud(Amazon EC2) 인스턴스를 만드는 데 사용됩니다. 이 패턴에서는 생성자 계정이라고 하는 별도의 중앙 집중식 AWS 계정에서 AMI를 생성할 수 있습니다. 그런 다음 동일한 AWS 리전에 있는 여러 AWS 계정(이 패턴에서는 소비자 계정이라고 함)에서 AMI를 공유할 수 있습니다. 단일 계정에서 AMI를 관리하면 확장성이 제공되고 거버넌스가 간소화됩니다. 소비자 계정에서는 Amazon EC2 Auto Scaling [시작 템플릿](#)과 Amazon Elastic Kubernetes Service(Amazon EKS) [노드 그룹](#)의 공유 AMI를 참조할 수 있습니다.

공유 AMI가 더 이상 [사용되지 않거나](#), [등록 취소되거나](#), [공유되지 않는](#) 경우, 소비자 계정에서 AMI를 참조하는 AWS 서비스는 이 AMI를 사용하여 새 인스턴스를 시작할 수 없습니다. 동일한 인스턴스의 모든 Auto Scaling 이벤트 또는 재시작이 실패합니다. 이로 인해 프로덕션 환경에서 애플리케이션 가동 중지나 성능 저하와 같은 문제가 발생할 수 있습니다. 여러 AWS 계정에서 AMI 공유 및 사용 이벤트가 발생하면 이 활동을 모니터링하기 어려울 수 있습니다.

이 패턴을 사용하면 동일한 리전의 계정 간에 공유된 AMI 사용 및 상태를 모니터링할 수 있습니다. 아마존, 아마존 다이내모DB, AWS Lambda EventBridge, 아마존 심플 이메일 서비스 (Amazon SES)와 같은 서버리스 AWS 서비스를 사용합니다. Terraform을 사용하여 인프라를 코드 (IaC) 로 프로비저닝합니다. HashiCorp 이 솔루션은 소비자 계정의 서비스가 등록 취소되거나 공유되지 않은 AMI를 참조할 때 알림을 제공합니다.

사전 조건 및 제한 사항

사전 조건

- 2개 이상의 활성 AWS 계정: 생성자 계정 1개와 하나 이상의 소비자 계정
- 생성자 계정에서 소비자 계정으로 공유되는 하나 이상의 AMI
- Terraform CLI, [설치됨](#)(Terraform 설명서)
- Terraform AWS Provider, [구성됨](#)(Terraform 설명서)
- (선택 사항이지만 권장됨) Terraform 백엔드, [구성됨](#)(Terraform 설명서)
- Git, [설치됨](#)

제한 사항

- 이 패턴은 계정 ID를 사용하여 특정 계정에 공유된 AMI를 모니터링합니다. 이 패턴은 조직 ID를 사용하여 조직에 공유된 AMI를 모니터링하지 않습니다.
- AMI는 동일한 AWS 리전 내에 있는 계정에만 공유할 수 있습니다. 이 패턴은 단일 대상 리전 내의 AMI를 모니터링합니다. 여러 리전의 AMI 사용을 모니터링하려면 이 솔루션을 각 리전에 배포하세요.
- 이 패턴은 이 솔루션이 배포되기 전에 공유된 AMI를 모니터링하지 않습니다. 이전에 공유한 AMI를 모니터링하려면 AMI를 공유 해제한 다음 소비자 계정에 다시 공유하면 됩니다.

제품 버전

- Terraform 버전 1.2.0 이상
- Terraform AWS Provider 버전 4.20 이상

아키텍처

대상 기술 스택

Terraform을 통해 IaC로 프로비저닝되는 리소스는 다음과 같습니다.

- Amazon DynamoDB 테이블
- 아마존 EventBridge 규칙
- AWS Identity and Access Management(IAM) 역할
- AWS Lambda 함수
- Amazon SES

대상 아키텍처

이 다이어그램은 다음 워크플로를 보여줍니다.

1. 생성자 계정의 AMI는 동일한 AWS 리전의 소비자 계정과 공유됩니다.
2. AMI를 공유하면 생성자 계정의 Amazon EventBridge 규칙이 ModifyImageAttribute 이벤트를 캡처하고 생성자 계정에서 Lambda 함수를 시작합니다.
3. Lambda 함수는 AMI와 관련된 데이터를 생성자 계정의 DynamoDB 테이블에 저장합니다.
4. 소비자 계정의 AWS 서비스가 공유 AMI를 사용하여 Amazon EC2 인스턴스를 시작하거나 공유 AMI가 시작 템플릿과 연결된 경우, 소비자 계정의 EventBridge 규칙이 공유 AMI 사용을 캡처합니다.
5. 이 EventBridge 규칙은 소비자 계정에서 Lambda 함수를 시작합니다. Lambda 함수는 다음 작업을 수행합니다.
 - a. Lambda 함수는 소비자 계정의 DynamoDB 테이블에 있는 AMI 관련 데이터를 업데이트합니다.
 - b. Lambda 함수는 생성자 계정에서 IAM 역할을 맡고 생성자 계정의 DynamoDB 테이블을 업데이트합니다. Mapping 테이블에서 인스턴스 ID 또는 시작 템플릿 ID를 해당 AMI ID에 매핑하는 항목을 생성합니다.
6. 생성자 계정의 중앙에서 관리되는 AMI는 더 이상 사용되지 않거나, 등록 취소되거나, 공유되지 않습니다.
7. 생성자 계정의 EventBridge 규칙은 remove 작업과 함께 ModifyImageAttribute 또는 DeregisterImage 이벤트를 캡처하고 Lambda 함수를 시작합니다.
8. Lambda 함수는 DynamoDB 테이블을 확인하여 AMI가 소비자 계정에서 사용되는지 확인합니다. Mapping 테이블에 AMI와 연결된 인스턴스 ID 또는 시작 템플릿 ID가 없는 경우 프로세스가 완료됩니다.
9. 인스턴스 ID 또는 시작 템플릿 ID가 Mapping 테이블의 AMI와 연결되어 있는 경우 Lambda 함수는 Amazon SES를 사용하여 구성된 구독자에게 이메일 알림을 보냅니다.

도구

서비스

- [Amazon DynamoDB](#)는 빠르고 예측 가능하고 확장 가능한 성능을 제공하는 완전 관리형 NoSQL 데이터베이스 서비스입니다.

- [EventBridgeAmazon](#)은 다양한 소스의 실시간 데이터와 애플리케이션을 연결하는 데 도움이 되는 서버리스 이벤트 버스 서비스입니다. AWS Lambda 함수, API 대상을 사용하는 HTTP 간접 호출 엔드포인트 또는 다른 AWS 계정의 이벤트 버스를 예로 들 수 있습니다.
- [AWS Identity and Access Management\(IAM\)](#)를 사용하면 사용자에 대해 인증 및 권한 부여를 제어함으로써 AWS 리소스에 대한 액세스를 안전하게 관리할 수 있습니다.
- [AWS Lambda](#)는 서버를 프로비저닝하거나 관리할 필요 없이 코드를 실행하는 데 도움이 되는 컴퓨팅 서비스입니다. 필요할 때만 코드를 실행하며 자동으로 확장이 가능하므로 사용한 컴퓨팅 시간만큼만 비용을 지불합니다.
- [Amazon Simple Email Service\(Amazon SES\)](#)를 사용하면 자신의 이메일 주소와 도메인을 사용하여 이메일을 보내고 받을 수 있습니다.

기타 도구

- [HashiCorp Terraform](#)은 코드를 사용하여 클라우드 인프라 및 리소스를 프로비저닝하고 관리할 수 있도록 지원하는 오픈 소스 코드형 인프라 (IaC) 도구입니다.
- [Python](#)은 범용 컴퓨터 프로그래밍 언어입니다.

코드 리포지토리

[이 패턴의 코드는 -terraform-samples 저장소에서 사용할 수 있습니다. GitHub cross-account-ami-monitoring](#)

모범 사례

- [AWS Lambda 함수 사용에 대한 모범 사례](#)를 따르세요.
- [AMI 구축 모범 사례](#)를 따르세요.
- IAM 역할을 생성할 때는 최소 권한 원칙에 따라 작업을 수행하는 데 필요한 최소 권한을 부여하세요. 자세한 내용은 IAM 설명서의 [최소 권한 부여](#) 및 [보안 모범 사례](#)를 참조하세요.
- AWS Lambda 함수에 대한 모니터링 및 알림을 설정하세요. 자세한 내용은 [Lambda 함수 모니터링 및 문제 해결](#)을 참조하세요.

에픽

Terraform 구성 파일을 사용자 지정

작업	설명	필요한 기술
<p>AWS CLI 명령된 프로파일을 생성합니다.</p>	<p>생성자 계정과 각 소비자 계정에 대해 AWS Command Line Interface(AWS CLI)가 명령된 프로파일을 생성합니다. 지침은 AWS 시작하기 리소스 센터의 AWS CLI 설정을 참조하세요.</p>	<p>DevOps 엔지니어</p>
<p>리포지토리를 복제합니다.</p>	<p>다음 명령을 입력합니다. 이것은 SSH를 사용하여 cross-account-ami-monitoring-테라폼-샘플 저장소를 복제합니다. GitHub</p> <pre data-bbox="597 1066 1026 1297">git clone git@github.com:aws-samples/cross-account-ami-monitoring-terraform-samples.git</pre>	<p>DevOps 엔지니어</p>
<p>provider.tf 파일을 업데이트합니다.</p>	<ol style="list-style-type: none"> 다음 명령을 입력하여 복제된 리포지토리의 terraform 폴더로 이동합니다. <pre data-bbox="630 1556 1029 1717">cd cross-account-ami-monitoring/terraform</pre> <ol style="list-style-type: none"> provider.tf 파일을 업데이트합니다. 	<p>DevOps 엔지니어</p>

작업	설명	필요한 기술
	<p>3. 다음과 같이 생성자 계정 및 소비자 계정에 대한 Terraform AWS Provider 구성을 업데이트하세요.</p> <ul style="list-style-type: none"> • <code>alias</code>에 공급자 구성의 이름을 입력합니다. • <code>region</code>에 이 솔루션을 배포할 대상 AWS 리전을 입력합니다. • <code>profile</code>에 계정 액세스를 위한 AWS CLI 이름이 지정된 프로파일을 입력합니다. <p>4. 두 개 이상의 소비자 계정을 구성하는 경우 각 추가 소비자 계정에 대한 프로파일을 생성하세요.</p> <p>5. <code>provider.tf</code> 파일을 저장하고 닫습니다.</p> <p>공급자 구성에 대한 자세한 내용은 Terraform 설명서의 다중 공급자 구성을 참조하세요.</p>	

작업	설명	필요한 기술
<p>terraform.tfvars 파일을 업데이트합니다.</p>	<ol style="list-style-type: none"> 1. terraform.tfvars 파일을 엽니다. 2. account_email_mapping 파라미터에서 다음과 같이 생성자 계정 및 소비자 계정에 대한 알림을 구성합니다. <ul style="list-style-type: none"> • account에 계정 ID를 입력합니다. • email에 알림을 보낼 이메일 주소를 입력합니다. 계정 당 하나의 이메일 주소만 입력할 수 있습니다. 3. 두 개 이상의 소비자 계정을 구성하는 경우 각 추가 소비자 계정에 대한 계정 및 이메일 주소를 입력합니다. 4. terraform.tfvars 파일을 저장하고 닫습니다. 	<p>DevOps 엔지니어</p>

작업	설명	필요한 기술
main.tf 파일을 업데이트합니다.	<p>이 솔루션을 두 개 이상의 소비자 계정에 배포하는 경우에만 이 단계를 완료하세요. 이 솔루션을 한 소비자 계정에만 배포하는 경우 이 파일을 수정할 필요가 없습니다.</p> <ol style="list-style-type: none"> 1. main.tf 파일을 엽니다. 2. 각 추가 소비자 계정에 대해 템플릿의 consumer_account_A 모듈을 기반으로 새 모듈을 생성하세요. 각 소비자 계정 provider의 값은 provider.tf 파일에 입력한 별칭과 일치해야 합니다. 3. main.tf 파일을 저장하고 닫습니다. 	DevOps 엔지니어

Terraform을 사용하여 솔루션을 배포

작업	설명	필요한 기술
솔루션을 배포합니다.	<p>Terraform CLI에서 다음 명령을 입력하여 생성자 및 소비자 계정에 AWS 리소스를 배포하세요.</p> <ol style="list-style-type: none"> 1. 다음 명령을 입력하여 Terraform을 초기화합니다. <pre>terraform init</pre>	DevOps 엔지니어

작업	설명	필요한 기술
	<p>2. Terraform 구성을 검증하려면 다음 명령을 입력합니다.</p> <pre>terraform validate</pre> <p>3. Terraform 실행 계획을 생성하려면 다음 명령을 입력합니다.</p> <pre>terraform plan</pre> <p>4. Terraform 계획의 구성 변경 사항을 검토하고 이러한 변경 사항을 구현할지 확인합니다.</p> <p>5. 다음 명령을 입력하여 리소스를 배포합니다.</p> <pre>terraform apply</pre>	
이메일 주소 ID를 확인합니다.	Terraform 계획을 배포했을 때 Terraform은 Amazon SES의 각 소비자 계정에 대한 이메일 주소 ID를 생성했습니다. 해당 이메일 주소로 알림을 보내려면 먼저 이메일 주소를 확인해야 합니다. 지침은 Amazon SES 설명서의 이메일 주소 ID 확인 을 참조하세요.	일반 AWS

리소스 배포 검증

작업	설명	필요한 기술
<p>생성자 계정에서 배포를 검증하세요.</p>	<ol style="list-style-type: none"> 1. 작성자 계정에 로그인합니다. 2. 탐색 모음에서 대상 리전이 표시되는지 확인합니다. 다른 리전에 있는 경우 현재 표시된 리전의 이름을 선택한 다음 대상 리전을 선택합니다. 3. https://console.aws.amazon.com/dynamodb/에서 DynamoDB 콘솔을 엽니다. 4. 탐색 창에서 테이블을 선택합니다. 5. 테이블 목록에서 AmiShare 테이블이 있는지 확인합니다. 6. https://console.aws.amazon.com/lambda/에서 Lambda 콘솔을 엽니다. 7. 탐색 창에서 함수를 선택합니다. 8. 함수 목록에서 ami-share 함수가 존재하는지 확인합니다. 9. https://console.aws.amazon.com/iamv2/에서 IAM 콘솔을 엽니다. 10. 탐색 창에서 역할을 선택합니다. 	<p>DevOps 엔지니어</p>

작업	설명	필요한 기술
	<p>11.역할 목록에 external-ddb-role 역할이 있는지 확인합니다.</p> <p>12.https://console.aws.amazon.com/events/ 에서 EventBridge 콘솔을 엽니다.</p> <p>13.탐색 창에서 규칙을 선택합니다.</p> <p>14.규칙 목록에서 modify_image_attribute_event 규칙이 있는지 확인합니다.</p> <p>15.https://console.aws.amazon.com/ses/에서 Amazon SES 콘솔을 엽니다.</p> <p>16.왼쪽 탐색 창에서 확인된 ID를 선택합니다.</p> <p>17>ID 목록에서 각 소비자 계정에 대해 이메일 주소 자격 증명 등록 및 확인되었는지 확인합니다.</p>	

작업	설명	필요한 기술
<p>소비자 계정에서 배포를 검증하세요.</p>	<ol style="list-style-type: none"> 1. 소비자 계정에 로그인합니다. 2. 탐색 모음에서 대상 리전이 표시되는지 확인합니다. 다른 리전에 있는 경우 현재 표시된 리전의 이름을 선택한 다음 대상 리전을 선택합니다. 3. https://console.aws.amazon.com/dynamodb/에서 DynamoDB 콘솔을 엽니다. 4. 탐색 창에서 테이블을 선택합니다. 5. 테이블 목록에서 Mapping 테이블이 있는지 확인합니다. 6. https://console.aws.amazon.com/lambda/에서 Lambda 콘솔을 엽니다. 7. 탐색 창에서 함수를 선택합니다. 8. 함수 목록에 ami-usage-function 및 ami-deregister-function 함수가 있는지 확인합니다. 9. https://console.aws.amazon.com/events/에서 EventBridge 콘솔을 엽니다. 10. 탐색 창에서 규칙을 선택합니다. 11. 규칙 목록에 ami_usage_events 및 ami_dereg 	<p>DevOps 엔지니어</p>

작업	설명	필요한 기술
	ister_events 규칙이 있는지 확인합니다.	

모니터링 검증

작업	설명	필요한 기술
생성자 계정에 AMI를 생성합니다.	<ol style="list-style-type: none"> 1. 생성자 계정에 프라이빗 AMI를 생성합니다. 자세한 내용은 Amazon EC2 인스턴스에서 AMI 생성을 참조하세요. 2. 새 AMI를 소비자 계정 중 하나와 공유합니다. 지침은 특정 AWS 계정과 AMI 공유를 참조하세요. 	DevOps 엔지니어
소비자 계정에서 AMI 사용.	<p>소비자 계정에서 공유 AMI를 사용하여 EC2 인스턴스 또는 시작 템플릿을 생성합니다. 지침은 사용자 지정 AMI에서 EC2 인스턴스를 시작하는 방법(AWS re:post 지식 센터) 또는 시작 템플릿을 만드는 방법(Amazon EC2 Auto Scaling 설명서)을 참조하세요.</p>	DevOps 엔지니어
모니터링 및 경고를 검증합니다.	<ol style="list-style-type: none"> 1. 작성자 계정에 로그인합니다. 2. https://console.aws.amazon.com/ec2/에서 Amazon EC2 콘솔을 엽니다. 3. 탐색 창에서 AMI를 선택합니다. 	DevOps 엔지니어

작업	설명	필요한 기술
	<ol style="list-style-type: none"> 4. 목록에서 AMI를 선택한 후 작업, AMI 권한 수정을 선택합니다. 5. 공유 계정 섹션에서 소비자 계정을 선택한 다음 선택한 항목 제거를 선택합니다. 6. 변경 사항 저장을 선택합니다. 7. 소비자 계정에 정의한 대상 이메일 주소에 AMI에 대한 공유가 취소되었다는 알림이 수신되는지 확인합니다. 	

(선택 사항)공유 AMI 모니터링 중지

작업	설명	필요한 기술
리소스를 삭제합니다.	<ol style="list-style-type: none"> 1. 다음 명령을 입력하여 이 패턴으로 배포된 리소스를 제거하고 공유 AMI의 모니터링을 중단합니다. <div style="border: 1px solid #ccc; border-radius: 10px; padding: 5px; width: fit-content; margin: 10px auto;"> <pre>terraform destroy</pre> </div> <ol style="list-style-type: none"> 2. yes를 입력하여 destroy 명령을 확인합니다. 	DevOps 엔지니어

문제 해결

문제	Solution
이메일 알림을 받지 못했습니다.	<p>Amazon SES 이메일이 전송되지 않은 데에는 여러 가지 이유가 있을 수 있습니다. 다음을 확인하십시오.</p> <ol style="list-style-type: none"> 에픽 섹션에서 리소스 배포 검증 에픽을 사용하여 인프라가 모든 AWS 계정에서 제대로 프로비저닝되었는지 확인하세요. Amazon Logs에서 Lambda 함수 이벤트를 검증하십시오. CloudWatch 지침은 Lambda 설명서의 CloudWatch 콘솔 사용을 참조하십시오. ID 기반 또는 리소스 기반 정책에 포함된 명시적 거부와 같은 권한 문제가 없는지 확인하세요. 자세한 내용은 IAM 설명서의 정책 평가 로직을 참조하십시오. Amazon SES에서 이메일 주소 ID의 상태가 확인되었는지 확인합니다. 자세한 내용은 이메일 주소 ID 확인을 참조하십시오.

관련 리소스

AWS 설명서

- [Python을 사용한 Lambda 함수 구축](#)(Lambda 설명서)
- [AMI 생성](#)(Amazon EC2 설명서)
- [특정 AWS 계정과 AMI 공유](#)(Amazon EC2 설명서)
- [AMI 등록 취소](#)(Amazon EC2 설명서)

Terraform 설명서

- [Terraform 설치](#)
- [Terraform 백엔드 구성](#)

- [Terraform AWS Provider](#)
- [Terraform 바이너리 다운로드](#)

AWS Organizations의 프로그래밍 방식 계정 폐쇄에 대한 알림 설정

작성자: Richard Milner-Watts(AWS), Debojit Bhadra(AWS) 및 Manav Yadav(AWS)

코드 리포지토리: [AWS 계정 폐쇄 알림](#)

환경: 프로덕션

기술: 관리 및 거버넌스

AWS 서비스: AWS, 아마존 CloudTrail, AWS Lambda EventBridge, AWS Organizations, 아마존 SNS

요약

[AWS Organizations용 CloseAccount API](#)를 사용하면 루트 자격 증명으로 계정에 로그인할 필요 없이 프로그래밍 방식으로 조직 내 회원 계정을 폐쇄할 수 있습니다. [RemoveAccountFromOrganization API](#)는 AWS Organizations의 조직에서 계정을 가져와서 독립 실행형 계정이 됩니다.

이러한 API는 AWS 계정을 폐쇄하거나 제거할 수 있는 운영자의 수를 잠재적으로 늘릴 수 있습니다. AWS Organizations 관리 계정의 AWS Identity and Access Management(IAM)를 통해 조직에 액세스할 수 있는 모든 사용자는 이러한 API를 호출할 수 있으므로, 연결된 다중 인증(MFA) 디바이스가 있는 계정 루트 이메일 소유자만 액세스할 수 있습니다.

이 패턴은 CloseAccount 및 RemoveAccountFromOrganization API가 호출될 때 알림을 구현하므로 이러한 활동을 모니터링할 수 있습니다. 알림을 위해 [Amazon Simple Notification Service](#)(Amazon SNS) 항목을 설정하십시오. [웹훅](#)을 통해 Slack 알림을 설정할 수도 있습니다.

사전 조건 및 제한 사항

사전 조건

- 활성 상태의 AWS 계정
- AWS Organizations 내의 한 조직
- 조직의 루트 아래에 있는 조직 관리 계정에 액세스하여 필요한 리소스를 생성할 수 있음

제한 사항

- [AWS Organizations API 참조에](#) 설명된 바와 같이, CloseAccount API를 사용하면 연속 30일 기간 내에 활성 회원 계정의 10%만 폐쇄할 수 있습니다.
- AWS 계정이 폐쇄되면 상태가 SUSPENDED로 변경됩니다. 이 상태 전환 후 90일 동안, AWS Support는 계정을 재개할 수 있습니다. 계정은 90일 후에 영구적으로 삭제됩니다.
- AWS Organizations 관리 계정 및 API에 액세스할 수 있는 사용자는 이러한 알림을 비활성화할 권한도 가질 수 있습니다. 실수로 삭제한 것이 아닌 악의적인 행동이 주요 우려 사항인 경우, 이 패턴으로 생성된 리소스를 [IAM](#) 권한 경계로 보호하는 것을 고려해 보세요.
- API의 CloseAccount 및 RemoveAccountFromOrganization에 대한 직접 호출이 미국 동부(버지니아 북부) 지역(us-east-1)에서 처리됩니다. 따라서, 이 이벤트를 관찰하려면 us-east-1에서 이 솔루션을 배포해야 합니다.

아키텍처

대상 기술 스택

- AWS Organizations
- AWS CloudTrail
- 아마존 EventBridge
- AWS Lambda
- Amazon SNS

대상 아키텍처

다음 다이어그램은 이 패턴의 솔루션 아키텍처를 보여 줍니다.

1. AWS Organizations는 CloseAccount 또는 RemoveAccountFromOrganization 요청을 처리합니다.
2. EventBridge Amazon은 AWS와 통합되어 이러한 이벤트를 기본 이벤트 CloudTrail 버스로 전달합니다.
3. 사용자 지정 Amazon EventBridge 규칙은 AWS Organizations의 요청을 매칭하고 AWS Lambda 함수를 호출합니다.
4. Lambda 함수는 사용자가 이메일 알림 또는 추가 처리를 위해 구독할 수 있는 SNS 주제에 메시지를 전송합니다.

5. Slack 알림이 활성화된 경우, Lambda 함수는 Slack 웹후크에 메시지를 전송합니다.

도구

서비스

- [CloudFormationAWS](#)는 인프라를 코드로 취급하여 관련 AWS 및 타사 리소스 모음을 모델링하고, 빠르고 일관되게 프로비저닝하고, 수명 주기 전반에 걸쳐 관리할 수 있는 방법을 제공합니다.
- [EventBridgeAmazon](#)은 애플리케이션을 다양한 소스의 데이터와 연결하는 데 사용할 수 있는 서버리스 이벤트 버스 서비스입니다. EventBridge 환경 변화를 나타내는 이벤트를 수신하고 규칙을 적용하여 이벤트를 대상으로 라우팅합니다. 규칙은 이벤트 패턴이라고 하는 이벤트 구조 또는 일정에 따라 이벤트를 대상과 일치시킵니다.
- [AWS Lambda](#)는 서버를 프로비저닝하거나 관리하지 않고도 코드를 실행할 수 있게 해주는 컴퓨팅 서비스입니다. Lambda는 필요 시에만 코드를 실행하며, 일일 몇 개의 요청에서 초당 수천 개의 요청까지 자동으로 확장이 가능합니다. 사용한 컴퓨팅 시간에 대해서만 비용을 지불하면 됩니다. 코드가 실행되지 않을 때는 비용이 부과되지 않습니다.
- [AWS Organizations](#)가 AWS 리소스 조정 및 확장 시 중앙에서 환경을 관리하고 통제하는 데 도움이 됩니다. AWS Organizations를 사용하면 프로그래밍 방식으로 새 AWS 계정을 만들고 리소스를 할당하며, 계정을 그룹화하여 워크플로를 구성하고, 거버넌스를 위해 계정 또는 그룹에 정책을 적용하고, 모든 계정에 대해 단일 결제 방법을 사용하여 청구를 간소화할 수 있습니다.
- [CloudTrailAWS](#)는 AWS 인프라 전반의 계정 활동을 모니터링 및 기록하고, 스토리지, 분석 및 수정 조치를 제어할 수 있도록 합니다.
- [Amazon Simple Notification Service \(Amazon SNS\)](#) 는 (A2A) application-to-person 및 (A2P) 통신 application-to-application 모두를 위한 완전 관리형 메시징 서비스입니다.

기타 도구

- [Python 라이브러리용 AWS Lambda Powertools](#)는 Lambda 함수에 대한 추적, 로깅, 지표 및 이벤트 처리 기능을 제공하는 유틸리티 세트입니다.

code

이 패턴의 코드는 GitHub [AWS 어카운트 클로저 알리미 리포지토리에](#) 있습니다.

솔루션에는 이 패턴의 아키텍처를 배포하는 CloudFormation 템플릿이 포함되어 있습니다. 이것은 [Python용 라이브러리용 AWS Lambda Powertools](#)을 사용하여 로깅 및 추적 기능을 제공합니다.

에픽

아키텍처 배포

작업	설명	필요한 기술
<p>솔루션 스택용 CloudFormation 템플릿을 실행합니다.</p>	<p>이 패턴의 CloudFormation 템플릿은 GitHub 리포지토리의 기본 브랜치에 있습니다. IAM 역할, EventBridge 규칙, Lambda 함수 및 SNS 주제를 배포합니다.</p> <p>템플릿을 실행하려면</p> <ol style="list-style-type: none"> 1. GitHub 리포지토리를 복제하여 솔루션 코드의 사본을 확보하십시오. 2. AWS Organizations 관리 계정을 위한 AWS Management Console을 엽니다. 3. 미국 동부 (버지니아 북부) 지역 (us-east-1) 을 선택한 다음 CloudFormation 콘솔을 엽니다. 4. account-closure-notifier.yml 템플릿을 사용하고 다음 값을 지정하여 스택을 생성합니다. <ul style="list-style-type: none"> • 스택 이름: aws-account-closure-notifier-stack • ResourcePrefix 파라미터: aws-account-closure-notifier 	<p>AWS 관리자</p>

작업	설명	필요한 기술
	<ul style="list-style-type: none"> • SlackNotification 파라미터: Slack 알림이 필요한 경우, 이 설정을 true(으)로 변경합니다. • SlackWebhookEndpoint 파라미터: Slack 알림이 필요한 경우, 웹훅 URL을 지정합니다. <p>CloudFormation 스택 시작에 대한 자세한 내용은 AWS 설명서를 참조하십시오.</p>	
<p>솔루션이 성공적으로 시작되었는지 확인하십시오.</p>	<ol style="list-style-type: none"> 1. CloudFormation 스택이 CREATE_COMPLETE 상태에 도달할 때까지 기다리십시오. 2. 에서 콘솔을 엽니다. EventBridge us-east-1 3. 이름 aws-account-closure-notifier-event-rule 을(를) 사용하여 새 규칙이 생성되었는지 확인합니다. 	<p>AWS 관리자</p>

작업	설명	필요한 기술
SNS 주제를 구독합니다.	<p>(선택 사항) SNS 주제를 구독하려는 경우:</p> <ol style="list-style-type: none"> us-east-1 에서 Amazon SNS 콘솔을 열고 aws-account-closure-notifier-sns-topic (이)라는 이름이 지정된 주제를 찾습니다. 주제 이름을 선택한 다음, 구독 생성을 선택합니다. 프로토콜에서 이메일을 선택합니다. 엔드포인트에 알림을 받는 데 사용할 수 있는 이메일 주소를 입력한 다음 구독 생성을 선택합니다. 이메일 수신함에서 AWS 알림에서 보낸 메시지를 확인합니다. 이메일에 포함된 링크를 사용하여 구독을 확인합니다. <p>SNS 알림 설정에 대한 자세한 내용은 Amazon SNS 설명서를 참조하세요.</p>	AWS 관리자

솔루션 확인

작업	설명	필요한 기술
기본 이벤트 버스로 테스트 이벤트를 전송합니다.	GitHub 리포지토리는 테스트를 위해 EventBridge 기본 이	AWS 관리자

작업	설명	필요한 기술
	<p>벤트 버스로 전송할 수 있는 샘플 이벤트를 제공합니다. EventBridge 규칙은 사용자 지정 이벤트 소스를 <code>account.closure.notifier</code> 사용하는 이벤트에도 반응합니다.</p> <p>참고: 이벤트를 AWS 서비스로 전송할 수 없으므로 CloudTrail 이벤트 소스를 사용하여 이 이벤트를 전송할 수 없습니다.</p> <p>테스트 이벤트를 전송하려면</p> <ol style="list-style-type: none"> 1. 에서 EventBridge 콘솔 <code>us-east-1</code> 여십시오. 2. 탐색 창의 버스에서 이벤트 버스를 선택한 다음, 기본 이벤트 버스를 선택합니다. 3. 이벤트 보내기를 선택합니다. 4. 이벤트 소스에서 <code>account.closure.notifier</code> 을(를) 입력합니다. 5. 세부 정보 유형에서 AWS API Call via CloudTrail 을(를) 입력합니다. 6. 이벤트 세부 정보를 보려면 GitHub 저장소의 <code>tests/dummy-event.json</code> 내용을 복사하여 텍스트 상자에 붙여넣습니다. 	

작업	설명	필요한 기술
	7. 전송을 선택하여 알림 워크 플로를 시작합니다.	
이메일 알림이 수신되었는지 확인합니다.	SNS 주제를 구독한 사서함에 서 알림을 확인하세요. 폐쇄된 계정과 API 호출을 수행한 주요 주체의 세부 정보가 포함된 이메일을 받게 됩니다.	AWS 관리자
Slack 알림이 수신되었는지 확인합니다.	(선택 사항) CloudFormation 템플릿을 배포할 때 SlackWebhookEndpoint 매개 변수에 웹훅 URL을 지정한 경우 웹훅에 매핑된 Slack 채널을 확인하십시오. 폐쇄된 계정과 API 호출을 수행한 주요 주체의 세부 정보가 포함된 메시지를 표시하게 됩니다.	AWS 관리자

관련 리소스

- [CloseAccount 조치](#) (AWS Organizations API 참조)
- [RemoveAccountFromOrganization 조치](#) (AWS Organizations API 참조)
- [Python용 AWS Lambda Powertools](#)

패턴 더 보기

- [AWS 리소스 평가 자동화](#)
- [AWS CDK를 사용하여 AWS Service Catalog 포트폴리오 및 제품 배포 자동화](#)
- [Cloud Custodian 및 AWS CDK를 사용하여 Systems Manager용 AWS 관리형 정책을 EC2 인스턴스 프로파일에 자동으로 연결](#)
- [기존 및 새 Amazon EBS 볼륨을 자동으로 암호화하기](#)
- [중앙 집중식 로깅 및 다중 계정 보안 가드레일](#)
- [시작 시 EC2 인스턴스에 필수 태그가 있는지 확인](#)
- [클라우드 운영 모델을 위한 RACI 또는 RASCI 지표 생성](#)
- [Amazon ECS 작업 정의를 생성하고 Amazon EFS를 사용하여 EC2 인스턴스에 파일 시스템을 마운트](#)
- [AWS 가드 정책을 사용하여 AWS Config 사용자 지정 규칙을 생성합니다. CloudFormation](#)
- [AWS Config 및 AWS Systems Manager로 사용하지 않는 Amazon Elastic Block Store\(Amazon EBS\) 볼륨 삭제](#)
- [AWS CDK 및 AWS를 사용하여 AWS 컨트롤 타워 컨트롤을 배포하고 관리합니다. CloudFormation](#)
- [Terraform을 사용하여 AWS Control Tower 컨트롤을 배포하고 관리합니다.](#)
- [AWS CodePipeline, AWS 및 AWS를 사용하여 여러 AWS 지역에 코드를 배포합니다. CodeCommit CodeBuild](#)
- [를 사용하여 AWS IAM ID 센터 자격 증명 및 할당에 대한 보고서를 내보냅니다. PowerShell](#)
- [트로포스피어를 사용하여 AWS Config 관리형 규칙이 포함된 AWS CloudFormation 템플릿을 생성합니다.](#)
- [SageMaker 노트북 인스턴스에 다른 AWS 계정의 CodeCommit 리포지토리에 대한 임시 액세스 권한 부여](#)
- [Step Functions와 Lambda 프록시 함수를 사용하여 여러 AWS 계정에서 CodeBuild 프로젝트를 시작합니다.](#)
- [ACM을 사용하여 Windows SSL 인증서를 Application Load Balancer로 마이그레이션](#)
- [IAM 루트 사용자 활동 모니터링](#)
- [AWS CodeCommit 이벤트에서 사용자 지정 작업 수행](#)
- [비 워크로드 서브넷을 위한 다중 계정 VPC 설계에서 라우팅 가능한 IP 공간 보존](#)
- [Amazon SES를 사용하여 단일 이메일 주소로 여러 AWS 계정 등록](#)
- [컨테이너를 다시 시작하지 않고 데이터베이스 보안 인증 교체](#)

- [온프레미스 SMTP 서버 및 Database Mail을 사용하여 Amazon RDS for SQL Server 데이터베이스 인스턴스에 대한 알림 전송하기](#)
- [AWS용 Grafana 모니터링 대시보드 설정 ParallelCluster](#)
- [AWS Organizations를 사용하여 Transit Gateway Attachment에 자동으로 태그 지정](#)
- [BMC Discovery 쿼리를 사용하여 마이그레이션 계획을 위한 마이그레이션 데이터 추출](#)
- [Amazon을 사용하여 모든 AWS 계정에 대한 IAM 자격 증명 보고서를 시각화합니다. QuickSight](#)

메시징 및 커뮤니케이션

주제

- [Amazon MQ에서 RabbitMQ 구성의 자동화](#)
- [Amazon Connect 콜센터의 에이전트 워크스테이션의 통화 품질을 개선합니다.](#)
- [패턴 더 보기](#)

Amazon MQ에서 RabbitMQ 구성의 자동화

작성자: Yogesh Bhatia 및 Afroz Khan

환경: PoC 또는 파일럿

기술: 메시징 및 통신; 인프라
DevOps

AWS 서비스: 아마존 MQ,
AWS CloudFormation

요약

[Amazon MQ](#)는 널리 사용되는 다양한 메시지 브로커와 호환되는 관리형 메시지 브로커 서비스입니다. RabbitMQ와 함께 Amazon MQ를 사용하면 여러 브로커 및 구성 옵션을 갖춘 Amazon Web Services(AWS) 클라우드에서 관리되는 강력한 RabbitMQ 클러스터가 제공됩니다. Amazon MQ는 가용성이 높고 안전하며 확장 가능한 인프라를 제공하고, 초당 대량의 메시지를 쉽게 처리할 수 있습니다. 여러 애플리케이션이 서로 다른 가상 호스트, 대기열, 교환이 있는 인프라를 사용할 수 있습니다. 하지만 이러한 구성 옵션을 관리하거나 인프라를 수동으로 생성하려면 시간과 노력이 필요할 수 있습니다. 이 패턴은 단일 파일을 통해 RabbitMQ 구성을 한 단계에서 관리하는 방법을 설명합니다. 이 패턴과 함께 제공되는 코드를 Jenkins 또는 Bamboo와 같은 지속적 통합(CI) 도구 내에 포함할 수 있습니다.

이 패턴을 사용하여 모든 RabbitMQ 클러스터를 구성할 수 있습니다. 필요한 것은 클러스터에 대한 연결뿐입니다. RabbitMQ 구성을 관리하는 다른 방법이 많이 있지만, 이 솔루션은 전체 애플리케이션 구성을 한 단계에서 생성하므로 대기열 및 기타 세부 정보를 쉽게 관리할 수 있습니다.

사전 조건 및 제한 사항

사전 조건

- 계정을 포인팅하도록 설치되고 구성되는 Command Line Interface(CLI)(지침은 [CLI 설명서](#)를 참조)
- 플레이북을 실행하여 구성을 생성할 수 있도록 하기 위해 설치되는 Ansible
- rabbitmqadmin 설치됨(지침은 [RabbitMQ 설명서](#)를 참조)
- 양호한 아마존 지표로 생성된 아마존 MQ의 RabbitMQ 클러스터 CloudWatch

추가 요구 사항

- 가상 호스트와 사용자에 대한 구성은 JSON의 일부로서가 아니라 별도로 생성해야 합니다.
- 구성 JSON은 리포지토리의 일부이고 버전 제어식이어야 합니다.

- rabbitmqadmin CLI의 버전은 RabbitMQ 서버의 버전과 동일해야 하므로 RabbitMQ 콘솔에서 CLI를 다운로드하는 것이 가장 좋습니다.
- 파이프라인의 일환으로서, 매번 실행하기 전에 JSON 구문을 검증해야 합니다.

제품 버전

- CLI 버전 2.0
- Ansible 버전 2.9.13
- rabbitmqadmin 버전 3.9.13(RabbitMQ 서버 버전과 동일해야 함)

아키텍처

소스 기술 스택

- 기존 온프레미스 가상 머신(VM) 또는 Kubernetes 클러스터에서 실행되는 RabbitMQ 클러스터(온프레미스 또는 클라우드에서)

대상 기술 스택

- RabbitMQ용 Amazon MQ의 자동화된 RabbitMQ 구성

대상 아키텍처

RabbitMQ를 구성하는 방법에는 여러 가지가 있습니다. 이 패턴은 단일 JSON 파일에 모든 구성이 포함된 가져오기 구성 기능을 사용합니다. 이 파일은 모든 설정에 적용되며 Bitbucket 또는 Git와 같은 버전 제어 시스템으로 관리할 수 있습니다. 이 패턴은 Ansible을 사용하여 rabbitmqadmin CLI를 통해 구성을 구현합니다.

도구

도구

- [rabbitmqadmin](#)은 RabbitMQ HTTP 기반 API를 위한 명령줄 도구입니다. RabbitMQ 노드 및 클러스터를 관리하고 모니터링하는 데 사용됩니다.
- [Ansible](#)은 애플리케이션 및 IT 인프라를 자동화하기 위한 오픈소스 도구입니다.

- [CLI](#)는 명령줄 셸의 명령을 사용하여 서비스와 상호 작용할 수 있습니다.

서비스

- [Amazon MQ](#)는 관리형 메시지 브로커 서비스로서, 클라우드에서 메시지 브로커를 쉽게 설정하고 운영할 수 있게 해줍니다.
- [AWS](#)는 코드형 인프라를 사용하여 AWS 인프라를 설정하고 클라우드 프로비저닝 속도를 높일 수 있도록 CloudFormation 지원합니다.

코드

이 패턴에 사용된 JSON 구성 파일과 샘플 Ansible 플레이북은 첨부 파일에 나와 있습니다.

에픽

사용자의 인프라 생성

작업	설명	필요한 기술
AWS에서 RabbitMQ 클러스터를 생성합니다.	RabbitMQ 클러스터가 아직 없는 경우 AWS를 사용하여 AWS에 CloudFormation 스택을 생성할 수 있습니다. 또는 Ansible의 Cloudformation 모듈 을 사용하여 스택을 생성할 수 있습니다. 후자의 접근 방식을 사용하면 두 가지 작업, 즉 RabbitMQ 인프라 생성 및 구성 관리 작업에 Ansible을 사용할 수 있습니다.	AWS CloudFormation, 앤서블

RabbitMQ용 Amazon MQ 브로커 구성

작업	설명	필요한 기술
속성 파일을 생성합니다.	첨부 파일에 있는 JSON 구성 파일(rabbitmqc	JSON

작업	설명	필요한 기술
	<p>onfig.json)을 다운로드 하거나 RabbitMQ 콘솔에서 이를 내보냅니다. 대기열, 교환, 바인딩을 구성하도록 수정합니다. 이 구성 파일은 다음을 보여줍니다.</p> <ul style="list-style-type: none"> - 대기열 2개 생성: sample-queue1 및 sample-queue2 - 교환 2개 생성: sample-exchange1 및 sample-exchange2 - 대기열과 교환 간 바인딩 구현 <p>이러한 구성은 rabbitmqadmin에서 요구하는 대로 루트 (/) 가상 호스트에서 수행됩니다.</p>	

작업	설명	필요한 기술
<p>RabbitMQ 인프라에 대한 Amazon MQ의 세부 정보를 검색합니다.</p>	<p>AWS에서 RabbitMQ 인프라에 대한 다음 세부 정보를 검색합니다.</p> <ul style="list-style-type: none"> • 브로커 이름 • RabbitMQ 호스트 • RabbitMQ 사용자 이름(클러스터 생성 중에 생성된 관리자 사용자) • RabbitMQ 암호 <p>AWS Management Console 또는 CLI를 사용하여 이 정보를 검색할 수 있습니다. 이러한 세부 정보를 통해 Ansible 플레이북은 계정에 연결하고 RabbitMQ 클러스터를 사용하여 명령을 실행할 수 있습니다.</p> <p>중요: Ansible 플레이북을 실행하는 컴퓨터가 계정에 액세스할 수 있어야 하며, 사전 조건 섹션에 설명된 대로 CLI가 이미 구성되어 있어야 합니다.</p>	<p>CLI, Amazon MQ</p>

작업	설명	필요한 기술
<p>hosts_var 파일을 생성합니다.</p>	<p>Ansible용 hosts_var 파일을 생성하고, 파일에 모든 변수가 정의되어 있는지 확인합니다. Ansible 볼트를 사용하여 암호를 저장하는 것을 고려합니다. 다음과 같이 hosts_var 파일을 구성할 수 있습니다(별표를 사용자 정보로 대체).</p> <pre data-bbox="592 632 1027 993"> RABBITMQ_HOST: "*****.mq.us-east-2.amazonaws.com" RABBITMQ_VHOST: "/" RABBITMQ_USERNAME: "admin" RABBITMQ_PASSWORD: "*****" </pre>	<p>Ansible</p>
<p>Ansible 플레이북을 생성합니다.</p>	<p>샘플 플레이북은 첨부 파일에서 <code>ansible-rabbit-config.yaml</code> (을)를 참조하십시오. 이 파일을 다운로드하고 저장합니다. Ansible 플레이북은 애플리케이션에 필요한 모든 RabbitMQ 구성(예: 대기열, 교환, 바인딩)을 가져오고 관리합니다.</p> <p>암호 보안과 같은 Ansible 플레이북의 모범 사례를 따릅니다. Ansible 볼트를 사용하여 암호를 암호화하고, 암호화된 파일에서 RabbitMQ 암호를 검색합니다.</p>	<p>Ansible</p>

구성 배포

작업	설명	필요한 기술
플레이북을 실행합니다.	<p>이전 에픽에서 생성한 Ansible 플레이북을 실행합니다.</p> <pre>ansible-playbook ansible-rabbit-con fig.yaml</pre> <p>RabbitMQ 콘솔의 새 구성을 검증할 수 있습니다.</p>	RabbitMQ, Amazon MQ, Ansible

관련 리소스

- [RabbitMQ에서 Amazon MQ로 마이그레이션](#)(AWS 블로그 게시물)
- [관리 명령줄 도구](#)(RabbitMQ 설명서)
- [AWS CloudFormation 스택 생성 또는 삭제](#) (Ansible 설명서)
- [메시지 기반 애플리케이션을 RabbitMQ용 Amazon MQ로 마이그레이션](#)(AWS 블로그 게시물)

첨부

이 문서와 관련된 추가 콘텐츠에 액세스하려면 [attachment.zip](#) 파일의 압축을 풉니다.

Amazon Connect 콜센터의 에이전트 워크스테이션의 통화 품질을 개선합니다.

작성자: Ernest Ozdoba(AWS)

환경: 프로덕션

기술: 메시징 및 통신, 최종 사용자 컴퓨팅

AWS 서비스: Amazon Connect

요약

통화 품질 문제는 콜 센터에서 해결하기가 가장 어려운 문제 중 일부입니다. 음성 품질 문제와 복잡한 문제 해결 절차를 피하려면 에이전트의 작업 환경과 워크스테이션 설정을 최적화해야 합니다. 이 패턴은 Amazon Connect 콜센터의 상담원 워크스테이션에 대한 음성 품질 최적화 기술을 설명합니다. 다음 영역의 권장 사항을 제공합니다.

- 작업 환경 조정. 에이전트의 주변 환경은 네트워크를 통해 음성이 전송되는 방식에는 영향을 미치지 않지만 통화 품질에는 영향을 미칩니다.
- 에이전트 워크스테이션 설정. 연락 센터 워크스테이션의 하드웨어 및 네트워크 구성은 통화 품질에 상당한 영향을 미칩니다.
- 브라우저를 설정합니다. 에이전트는 웹 브라우저를 사용하여 Amazon Connect 연락 제어판(CCP) 웹사이트에 액세스하고 고객과 소통하므로 브라우저 설정이 통화 품질에 영향을 미칠 수 있습니다.

다음 구성 요소도 통화 품질에 영향을 줄 수 있지만 워크스테이션의 범위를 벗어나므로 이 패턴에 포함되지 않습니다.

- 트래픽은 AWS Direct Connect, 풀 터널 VPN 또는 스플릿 터널 VPN을 통해 Amazon Web Services(AWS) 클라우드로 이동합니다.
- 회사 사무실 내부 또는 외부에서 작업할 때의 네트워크 상태
- 공중 전화망(PSTN) 연결
- 고객의 장치 및 전화 통신 사업자
- 가상 데스크톱 인프라(VDI) 설정

이러한 영역과 관련된 자세한 내용은 Amazon Connect 설명서에서 [일반적인 연락 제어판\(CCP\) 문제 및 엔드포인트 테스트 유틸리티 사용](#)을 참조하세요.

사전 조건 및 제한 사항

사전 조건

- 헤드셋과 워크스테이션은 [Amazon Connect 관리자 안내서](#)에 명시된 요구 사항을 준수해야 합니다.

제한 사항

- 이 패턴의 최적화 기법은 스마트폰 음성 품질에 적용됩니다. 데스크폰 모드에서 Amazon Connect CCP를 구성할 때는 적용되지 않습니다. 하지만 스마트폰 설정이 통화에 적합한 음성 품질을 제공하지 않는 경우 데스크폰 모드를 사용할 수 있습니다.

제품 버전

- 지원되는 브라우저 및 버전은 [Amazon Connect 관리자 안내서](#)를 참조하세요.

아키텍처

이 패턴은 에이전트 워크스테이션 설정을 대상으로 하므로 아키텍처에 구애받지 않습니다. 다음 다이어그램에서 볼 수 있듯이 에이전트에서 고객으로의 음성 경로는 에이전트의 헤드셋, 브라우저, 운영 체제, 워크스테이션 하드웨어 및 네트워크의 영향을 받습니다.

Amazon Connect 콜센터에서는 사용자의 오디오 연결이 WebRTC를 통해 설정됩니다. 음성은 [Opus 대화형 오디오 코덱](#)으로 인코딩되고 전송 중에는 보안 실시간 전송 프로토콜(SRTP)로 암호화됩니다. VPN, 사설 WAN/LAN 및 ISP 네트워크를 비롯한 다른 네트워크 아키텍처도 가능합니다.

도구

- [Amazon Connect 엔드포인트 테스트 유틸리티](#) - 이 유틸리티는 네트워크 연결 및 브라우저 설정을 확인합니다.
- WebRTC 설정을 위한 브라우저 구성 편집기:
 - Firefox의 경우: `about:config`
 - Chrome의 경우: `chrome://flags`
- [CCP 로그 파서](#) - 이 도구를 사용하면 문제 해결을 위해 CCP 로그를 분석할 수 있습니다.

에픽

작업 환경을 조정합니다.

작업	설명	필요한 기술
배경 소음을 줄입니다.	<p>시끄러운 환경을 피하세요. 불가능한 경우 다음과 같은 방음 요령을 사용하여 환경을 최적화하세요.</p> <ul style="list-style-type: none"> 커튼, 카펫, 부드러운 가구와 같이 소음을 분산시키는 표면을 사용하여 소음을 흡수하세요. 책상 사이에 가림막을 설치하여 소음을 차단합니다. 집중력을 높이고 프라이버시를 보장하는 백색 소음 발생기와 같은 능동형 소음 제거(ANC) 솔루션이나 잡음 제거 헤드셋을 사용하는 것이 좋습니다. 호출 시 에코를 방지합니다. 크고 빈 공간은 반향 효과를 일으키거나 소음을 증폭시킬 수 있습니다. 소리가 튀는 표면을 가리면 반향을 줄이는데 도움이 됩니다. 	에이전트, 매니저

에이전트 워크스테이션 설정 최적화

작업	설명	필요한 기술
적합한 헤드셋을 선택합니다.	<ul style="list-style-type: none"> 환경이 시끄럽다면 스테레오 헤드셋을 선택하세요. 양 	에이전트, 매니저

작업	설명	필요한 기술
	<p>쪽 귀로 사운드를 전달하면 에이전트가 집중하고 고객의 목소리를 더 잘 들을 수 있으며 에이전트가 목소리를 높일 가능성이 줄어들어 전반적인 소음이 줄어듭니다.</p> <ul style="list-style-type: none"> • 시끄러운 스피커나 내장된 컴퓨터 오디오를 사용하지 마세요. 최상의 품질을 얻려면 연락 센터 전용 유선 헤드셋을 사용하세요. 무선 헤드셋은 편리하지만 무선 간섭과 트랜스코딩으로 인해 오디오 지연이 가중되고 오디오 품질이 저하될 수 있습니다. 	

작업	설명	필요한 기술
<p>헤드셋을 의도대로 사용하합니다.</p>	<ul style="list-style-type: none"> • 헤드셋의 액티브 노이즈 캔슬링 및 음성 향상 기능이 있는 경우 이를 활성화하세요. ANC 또는 ANR과 같은 설정을 찾습니다. 이러한 설정을 활성화하는 방법에 대한 지침은 헤드셋의 사용 설명서를 참조하세요. • 마이크에 직접 대고 말할 수 있도록 마이크를 조정합니다. 마이크를 놓기에 가장 적합한 위치는 턱 바로 아래입니다. 올바르게 배치하면 사운드 레벨이 10데시벨(dB) 차이가 날 수 있습니다. 대부분의 헤드셋은 마이크 암(붐)을 돌리거나 구부릴 수 있으므로 말할 때는 올바른 위치에 두는 것이 중요합니다. • 일부 헤드셋에는 여러 개의 마이크가 탑재되어 있으며 음성 빔포밍과 같은 고급 기능이 탑재되어 있어 큰 소리 없이 음성을 캡처할 수 있습니다. 제조업체가 의도한 대로 기본 마이크를 사용하고 있는지 확인하려면 장치의 사용 설명서를 참조하세요. 	<p>에이전트</p>

작업	설명	필요한 기술
<p>워크스테이션 리소스를 확인합니다.</p>	<p>에이전트의 컴퓨터가 제대로 작동하는지 확인합니다. 리소스를 소비하는 타사 애플리케이션을 사용하는 경우 컴퓨터가 CCP를 실행하는 데 필요한 최소 하드웨어 요구 사항을 충족하지 못할 수 있습니다. 상담원이 통화 품질 문제를 겪는 경우 CCP에 사용할 수 있는 처리 능력(CPU), 디스크 공간, 네트워크 대역폭 및 메모리가 충분한지 확인하세요. 에이전트는 불필요한 애플리케이션과 탭을 모두 닫아야 CCP 성능과 통화 품질을 개선할 수 있습니다.</p>	<p>관리자</p>

작업	설명	필요한 기술
<p>운영 체제의 사운드 설정을 구성합니다.</p>	<p>마이크 레벨 및 부스트의 기본 설정은 일반적으로 정상적으로 작동합니다. 아웃바운드 음성이 작거나 마이크가 너무 많이 들리는 경우 이러한 설정을 조정하는 것이 도움이 될 수 있습니다. 마이크 설정은 컴퓨터의 시스템 사운드 구성(사운드, MacOS의 입력, Windows의 마이크 속성)에서 찾을 수 있습니다. 시스템 도구 또는 타사 애플리케이션을 통해 음성 품질에 영향을 줄 수 있는 고급 설정에 액세스할 수 있습니다. 확인할 수 있는 몇 가지 설정은 다음과 같습니다.</p> <ul style="list-style-type: none"> • 샘플 비율 - 이 값은 초당 사운드가 프로빙되는 횟수를 결정합니다. 기본 설정은 일반적으로 44 또는 48킬로헤르츠(kHz)입니다. Amazon Connect의 최적 값은 48kHz입니다. 브라우저 설정을 사용하여 기본값을 재정의할 수 있습니다. 자세한 내용은 Amazon Connect 관리자 안내서의 문제 해결 섹션을 참조하세요. • 게인 - 이 값은 마이크가 사운드를 증폭하는 정도를 결정합니다. 게인을 높이면 마이크에 배경 잡음이 더 많이 들릴 수 있습니다. 	<p>에이전트, 관리자</p>

작업	설명	필요한 기술
	<ul style="list-style-type: none"> • 비트 심도 - 이 디지털 해상도 값은 인식되는 사운드 진폭 레벨을 나타냅니다. 비트 심도가 높을수록 음성이 더 부드럽게 들립니다. 그러나 기존의 많은 전화 네트워크에서는 8비트 해상도만 지원하는 펄스 코드 변조(PCM) 표준을 사용합니다. • 개방 임계값 - 마이크가 포착하는 최소 사운드 진폭입니다. <p>음성 품질 문제가 발생하는 경우 추가 조사를 시작하기 전에 이 값을 기본 설정으로 복원하세요.</p> <p>설정 및 기타 조정 가능한 설정에 대한 자세한 내용은 장치 설명서를 참조하세요.</p>	

작업	설명	필요한 기술
<p>유선 네트워크를 사용합니다.</p>	<p>일반적으로 유선 이더넷은 지연 시간이 짧기 때문에 음성 데이터 전송에 필요한 일관된 전송 품질을 제공하기가 더 쉽습니다. 통화당 최소 100KB의 대역폭을 사용하는 것이 좋습니다.</p> <ul style="list-style-type: none"> • 상담원이 재택근무를 하는 경우 무선 연결을 통한 유선 연결을 사용하는 것이 좋습니다. 고객의 의견을 듣는 데 150밀리초 이상 걸리지 않아야 합니다. Amazon Connect 엔드포인트 테스트 유틸리티에서 Amazon Connect의 지연 시간 테스트에 액세스할 수 있습니다. 그러나 이 유틸리티는 브라우저에서 Amazon Connect 지역까지의 지연을 측정하며 고객까지는 걸리는 시간을 측정하지 않습니다. 150밀리초의 단방향 지연 권장 사항으로 에이전트와 고객이 서로 의견을 나눌 수 없게 되었습니다. 값은 끝에서 끝까지 측정되며 Amazon Connect 리전과 고객 간의 통화 부분을 포함하여 각 요소에 지연이 추가됩니다. • 상담원이 사무실에서 근무하는 경우 파라미터가 권장 범위 내에 있고 실시간 전송 프로토콜(RTP) 트래픽의 우선 	<p>네트워크 관리자, 에이전트</p>

작업	설명	필요한 기술
	<p>순위가 지정되는 한 기업 Wi-Fi는 허용됩니다.</p>	
<p>하드웨어 드라이버를 업데이트합니다.</p>	<p>자체 펌웨어가 있는 USB 또는 기타 유형의 헤드셋을 사용하는 경우 최신 버전으로 업데이트하는 것이 좋습니다. 보조 포트를 사용하는 간단한 헤드셋은 컴퓨터에 내장된 오디오 장치를 사용하므로 운영 체제 하드웨어 드라이버가 최신 상태인지 확인하세요. 드문 경우이긴 하지만 오디오 드라이버 업데이트로 인해 오디오 문제가 발생하여 롤백해야 할 수도 있습니다. 펌웨어 및 드라이버 버전 변경에 대한 자세한 내용은 장치 설명서를 참조하세요.</p>	<p>관리자</p>
<p>USB 허브와 동글을 사용하지 마세요.</p>	<p>헤드셋을 연결할 때 동글, 포트형 컨버터, 허브, 연장 케이블 등의 추가 장치는 사용하지 마세요.</p> <p>이러한 장치는 통화 품질에 영향을 미칠 수 있습니다. 대신 장치를 컴퓨터의 포트에 직접 연결하세요.</p>	<p>에이전트</p>

작업	설명	필요한 기술
CCP 로그를 확인합니다.	<p>CCP 로그 파서는 애플리케이션 로그를 확인하는 쉬운 방법을 제공합니다.</p> <ol style="list-style-type: none"> 호출 후 CCP 로그를 다운로드합니다. CCP 로그 파서를 엽니다. 로그 파일을 끌어다 놓아 분석을 위해 로그를 업로드합니다. 로그가 분석되면 기본적으로 스냅샷 및 로그 탭이 선택됩니다. 옆에 있는 지표 탭을 선택하여 인사이트를 확인합니다. WebRTC 지표 - audio_input 섹션에서 다음 사항을 확인합니다. <ul style="list-style-type: none"> 오디오 레벨 그래프를 통해 수신된 오디오 레벨이 0보다 높은지 확인할 수 있습니다. 이는 발신자로부터 오디오를 수신했음을 나타냅니다. 손실된 패킷에 대한 패킷 그래프. 이 차트에서 크게 증가한 것으로 나타나면 IT 지원팀에 문의하세요. WebRTC 지표 - audio_output 섹션에서 다음 사항을 확인합니다. <ul style="list-style-type: none"> 오디오 레벨 그래프는 오디오가 장치에서 전송되 	상담원 (고급 기술)

작업	설명	필요한 기술
	<p>있는지 확인하기 위한 것입니다.</p> <ul style="list-style-type: none"> 패킷 그래프. 패킷 손실이 급증하는 경우 IT 지원팀에 신고하세요. Jitter Buffer 및 RTT 그래프. 왕복 시간(RTT) 값이 300을 초과하면 호출 경험에 영향을 미칩니다. 이를 IT 지원팀에 보고합니다. 	

브라우저 설정 최적화

작업	설명	필요한 기술
<p>기본 WebRTC 설정으로 복원합니다.</p>	<p>WebRTC가 CCP와 소프트 폰 통화를 할 수 있어야 합니다. WebRTC 관련 기능에 대한 기본 설정을 유지하는 것이 좋습니다.</p> <ul style="list-style-type: none"> Chrome에서는 URL(chrome://flags)로 이동하여 플래그를 설정할 수 있습니다. 검색 상자에 WebRTC를 입력하여 CCP를 방해할 수 있는 설정을 찾고 이를 기본값으로 설정합니다. Firefox의 경우 주소 표시줄에 about:config를 입력한 다음 구성 페이지의 검색 상자에 WebRTC를 입력합니다. 기본이 아닌 설정은 굵은 텍스트 	<p>관리자</p>

작업	설명	필요한 기술
	스트로 표시되며 기본값으로 변경할 수 있습니다.	
문제 해결 시 브라우저 확장 프로그램을 비활성화하세요.	일부 브라우저 확장 프로그램은 통화 품질에 영향을 미치거나 통화가 제대로 연결되지 않을 수도 있습니다. 브라우저의 시크릿 창 또는 비공개 모드를 사용하고 모든 확장 프로그램을 비활성화합니다. 그래도 문제가 해결되면 브라우저 확장 프로그램을 검토하여 의심스러운 추가 기능을 찾거나 개별적으로 사용 중지하세요.	에이전트, 관리자
브라우저 샘플링 속도를 확인합니다.	마이크 입력이 최적의 48kHz 샘플링 레이트로 설정되어 있는지 확인합니다. 지침은 Amazon Connect 관리자 안내서 를 참조하세요.	에이전트, 관리자

관련 리소스

이 패턴의 단계를 따랐는데도 여전히 통화 품질 문제가 발생하는 경우 다음 리소스에서 문제 해결 팁을 참조하세요.

- [일반적인 연락 제어판\(CCP\) 문제](#)를 검토합니다.
- [엔드포인트 테스트 유틸리티](#)와의 연결을 확인합니다.
- 다른 문제는 [문제 해결 가이드](#)를 따르세요.

문제 해결 및 조정으로도 통화 품질 문제가 해결되지 않는 경우 근본 원인은 워크스테이션 외부에 있을 수 있습니다. 추가 문제 해결은 IT 지원팀에 문의하세요.

패턴 더 보기

- [CQRS 및 이벤트 소싱을 사용하여 모놀리식 유형을 마이크로서비스로 분해하기](#)
- [Amazon API Gateway를 Amazon SQS와 통합하여 비동기식 REST API를 처리합니다.](#)
- [Amazon SES를 사용하여 단일 이메일 주소로 여러 AWS 계정 등록](#)
- [AWS Fargate를 사용하여 메시지 기반 워크로드를 대규모로 실행](#)

마이그레이션

주제

- [를 사용하여 마이그레이션 전략 식별 및 계획을 자동화합니다. AppScore](#)
- [마이크로소프트 엑셀과 Python을 사용하여 AWS DMS 작업을 위한 AWS CloudFormation 템플릿을 생성합니다.](#)
- [자동화된 포트폴리오 검색으로 시작하기](#)
- [온프레미스 Cloudera 워크로드를 AWS의 Cloudera 데이터 플랫폼으로 마이그레이션](#)
- [RHEL 소스 서버를 재부팅한 후 SELinux를 비활성화하지 않고 Replication Agent를 자동으로 다시 시작](#)
- [리아키텍트](#)
- [리호스팅](#)
- [재배치](#)
- [리플랫폼](#)
- [워크로드별 마이그레이션 패턴](#)
- [패턴 더 보기](#)

를 사용하여 마이그레이션 전략 식별 및 계획을 자동화합니다.

AppScore

제작: 레흐 미그달 (AWS), 제프 데이비스 (테크놀로지 리미티드) AppScore

환경: 프로덕션	소스: 모든 워크로드	대상: AWS 클라우드
R 유형: 해당 없음	워크로드: 기타 모든 워크로드	기술: 마이그레이션, 현대화, 웹 및 모바일 앱, SaaS

AWS 서비스: AWS Application Discovery Service, AWS Migration Hub

요약

온프레미스 애플리케이션에는 Amazon Web Services(AWS) 클라우드의 이점을 활용할 수 있는 혁신적인 접근 방식이 필요합니다. [7가지 일반적인 마이그레이션 전략\(7R\)](#)은 온프레미스 데이터베이스 서버의 기술 변경부터 클라우드 네이티브 마이크로서비스 아키텍처를 사용하여 애플리케이션을 재구축하는 것까지 다양한 변환 옵션을 제공합니다.

전체 7R 모델을 사용하기로 선택하면 마이그레이션을 위해 서버를 평가하고 준비하는 데 그치지 않고 애플리케이션 및 비즈니스 수준에서 운영할 수 있습니다. [AWS 마이그레이션 평가기](#)와 같은 도구를 사용하여 서버 데이터를 얻을 수 있지만 다른 애플리케이션 정보는 기록되지 않는 경우가 많습니다(예를 들어, 로드맵 상태, 필요한 Recovery Time Objective(RTO) 및 Recovery Point Objective(RPO), 또는 데이터 개인 정보 보호 요구 사항).

이 패턴은 애플리케이션 중심의 [AppScore](#) 포트폴리오 뷰를 사용하여 이러한 문제를 방지하는 방법을 설명합니다. 여기에는 전체 7R 모델을 대상으로 각 애플리케이션에 대한 AWS 클라우드로의 권장 변환 경로가 포함됩니다. AppScore 애플리케이션 정보를 캡처하고, 이상적인 변환 경로를 결정하고, 클라우드 채택의 위험, 복잡성 및 이점을 식별하고, 마이그레이션 범위, 이동 그룹 및 일정을 신속하게 정의하는 데 도움이 됩니다.

이 패턴은 AWS와 AWS 파트너인 [AppScore 테크놀로지 리미티드](#)에 의해 만들어졌습니다.

사전 조건 및 제한 사항

사전 조건

- AWS 클라우드로 마이그레이션하려는 기존 애플리케이션.
- [AWS 마이그레이션 평가기](#)와 같은 도구에서 제공하는 기존 서버 인벤토리 정보. 마이그레이션의 나중 단계에서 이 데이터를 가져올 수도 있습니다.
- 파워 유저 권한이 있는 기존 AppScore 계정. AppScore 사용자 계정에 대한 자세한 내용은 사용자에게 [RBAC \(역할 기반 액세스 제어\) 를 할당하려면 어떻게 해야 하나요?](#) 를 참조하십시오. AppScore 설명서에서
- 에서 RBAC 역할을 할당하는 방법에 대한 이해 AppScore AppScore 채점 단계에서 묻는 질문에 맞는 세 가지 주제 전문가 (SME) 역할을 제공합니다. 즉, SME는 자신의 전문 지식 및 역할과 관련된 질문에만 답변합니다. 이에 대한 자세한 내용은 [사용자에게 역할 기반 액세스 제어 \(RBAC\) 를 할당하려면 어떻게 해야 하나요?](#) 를 참조하십시오. 설명서에서. AppScore
- 다음 세 가지 범주의 애플리케이션 속성을 기반으로 하는 AppScore 의 권장 사항에 대한 이해
 - 위험 - 애플리케이션의 비즈니스 중요도, 즉 기밀 데이터, 데이터 주권 요구 사항, 애플리케이션 사용자 또는 인터페이스 수 포함 여부
 - 복잡성 - 애플리케이션의 개발 언어(예를 들어, COBOL은 .NET 또는 PHP보다 높은 점수를 가짐), 수명, UI 또는 인터페이스 수
 - 이점 - 배치 처리 수요, 애플리케이션 프로파일, 재해 복구 모델, 개발 및 테스트 환경 사용
- AppScore의 4단계 반복 데이터 캡처에 대한 이해:
 - Signposting - 질문을 서버 데이터와 결합하여 7R 평가를 산출합니다. 자세한 내용은 설명서에서 [애플리케이션에 사인을 등록하고 점수를 매기는 방법을](#) 참조하십시오. AppScore
 - 채점 - 위험, 이점, 복잡성에 대한 점수를 산출하는 질문입니다.
 - 현재 상태 평가 - 애플리케이션의 현재 상태 평가를 제공하는 질문입니다.
 - 변환 - 미래 상태 설계를 위한 애플리케이션을 종합적으로 평가하는 질문입니다.

중요: 신청 점수를 받고, 7R 평가를 받고, 그룹 계획을 수립하려면 Signposting 및 채점 단계만 거치면 됩니다. 신청서와 양식 범위를 그룹화한 후에는 현재 상태 평가 및 변환 단계를 완료하여 신청서에 대한 보다 상세한 개요를 작성할 수 있습니다.

아키텍처

다음 다이어그램은 애플리케이션 및 서버 데이터를 사용하여 마이그레이션 전략 및 변환 계획에 대한 권장 사항을 만드는 AppScore 워크플로를 보여줍니다.

도구

- [AppScore](#)— AppScore 전체 7R 모델에 대해 애플리케이션별로 클라우드로 전환하는 권장 경로를 애플리케이션 중심으로 파악하여 포트폴리오를 제공하여 검색과 마이그레이션 구현 간의 격차를 해소할 수 있도록 지원합니다.
- [AWS 마이그레이션 평가기](#) - AWS 마이그레이션 평가기는 계획 및 마이그레이션을 위한 방향성 있는 비즈니스 사례를 만들 수 있도록 지원하는 마이그레이션 평가 서비스입니다.

에픽

초기 애플리케이션 목록 생성 및 로드

작업	설명	필요한 기술
지원서 목록을 준비하십시오.	<p>사용자 자격 증명으로 AppScore 포털에 로그인합니다. 애플리케이션 페이지에서 Import Template을 다운로드한 다음 Import Template을 애플리케이션의 비기술적 속성(예: 데이터 분류 또는 사용자 정의할 수 있는 속성 목록)으로 업데이트하십시오.</p> <p>이에 대한 자세한 내용은 설명서에서 AppScore 애플리케이션 및 비즈니스 설문지를 어떻게 변경하나요? 를 참조하십시오. AppScore</p> <p>참고: 애플리케이션 페이지에서 새 애플리케이션을 선택하여 애플리케이션을 수동으로 추가할 수도 있습니다. 그런 다</p>	마이그레이션 엔지니어

작업	설명	필요한 기술
	음 애플리케이션의 비기술적 속성을 입력할 수 있습니다.	
애플리케이션 데이터를 가져옵니다.	애플리케이션 페이지에서 애플리케이션 가져오기를 선택하여 애플리케이션 데이터를 가져옵니다.	마이그레이션 엔지니어

애플리케이션 및 비즈니스 데이터 캡처

작업	설명	필요한 기술
사인포스팅 및 채점 질문을 검토하고 답변하십시오.	<p>서버 페이지를 열고 서버 가져오기를 선택합니다. 서버 데이터가 포함된.csv 파일을 선택합니다.</p> <p>파일에는 이름, 데이터 센터, 운영 체제, 가상 또는 물리적, 애플리케이션 이름, 역할, 데이터베이스 기술, 환경, CPU 코어 수 및 사용률, RAM 크기 및 사용률, 디스크 크기 및 사용률, 일치하는 머신 유형, 현재 및 예상 월별 비용 등의 속성이 포함될 수 있습니다.</p> <p>열 매핑을 확인하고 확인 및 가져오기를 선택합니다. 가져온 데이터에서 누락된 정보는 서버 페이지에 강조 표시됩니다. 이 페이지에서 또는 대량 편집 옵션을 사용하여 이러한 격차를 해결할 수 있습니다. 서버는 관련 애플리케이션과 연결됩니</p>	앱 소유자

작업	설명	필요한 기술
	<p>다. 그러나 애플리케이션이 존재하지 않는 경우 애플리케이션이 자동으로 생성되고 서버가 연결됩니다. AppScore</p> <p>API 연결을 사용하여 AWS Migration Hub에서 데이터를 검색할 수도 있습니다. 이에 대한 자세한 내용은 API를 통해 AWS Migration Hub에서 서버를 가져오려면 어떻게 해야 합니까?를 참조하십시오. AppScore 설명서에서.</p> <p>참고: 검색 도구 (예: AWS Migration Evaluator) 를 사용하여 시간 경과에 따른 성능을 캡처한 경우, 가능한 한 빨리 서버 데이터의 초기 추출을 로드하고 성능 지표가 완전히 캡처되면 데이터를 새로 고쳐야 합니다. AppScore 서버 이름, 운영 체제 및 데이터베이스 버전, 데이터 센터, 환경을 사용하여 점수 및 7Rs 권장 사항을 제공합니다.</p>	
<p>애플리케이션 점수를 확인하십시오.</p>	<p>지원 페이지를 열어 지원서에 대한 점수 및 7R 평가를 확인하십시오. 현재 실행 비용도 계산됩니다. 이러한 계산은 새 정보를 애플리케이션 또는 서버 페이지로 가져올 때 업데이트됩니다.</p>	<p>앱 소유자</p>

작업	설명	필요한 기술
개별 애플리케이션을 분석하십시오.	애플리케이션 페이지에서 애플리케이션을 선택하여 자세한 권장 사항을 검토하십시오. 애플리케이션 평가 보고서를 선택하여 특정 애플리케이션에 대한 자세한 평가 데이터가 포함된.pdf 또는.docx 파일을 생성할 수 있습니다.	앱 소유자

마이그레이션 일정 만들기

작업	설명	필요한 기술
이동 그룹에 사용할 애플리케이션을 선택합니다.	<p>계획 페이지를 열고 그룹 빌더를 선택한 다음 요구 사항에 따라 애플리케이션 이동 그룹을 생성합니다.</p> <p>열 섹션의 애플리케이션 목록에서 속성을 추가하거나 제거할 수 있습니다. 필터 섹션의 애플리케이션 속성을 사용하여 특정 애플리케이션을 선택할 수도 있습니다. 여기에는 이미 기존 이동 그룹에 속한 모든 애플리케이션을 필터링하는 작업이 포함됩니다.</p>	마이그레이션 엔지니어
이동 그룹을 생성합니다.	선택한 그룹을 선택하고 이동 그룹의 이름을 입력하고 이동 그룹에 포함할 애플리케이션을 선택한 다음 그룹에 추가를 선택합니다.	마이그레이션 엔지니어

작업	설명	필요한 기술
<p>마이그레이션을 예약하십시오.</p>	<p>변환 일정 페이지에서는 이동 그룹에 대한 예상 변환 기간, 작업량 및 비용을 AppScore 제공합니다. 이동 그룹은 전체 변환 일정에 자동으로 추가됩니다.</p> <p>참고: 계획 설정 페이지에서 작업량 추정을 뒷받침하는 가정을 사용자 지정할 수 있습니다. 이렇게 하면 조직의 요구 사항에 맞게 조정할 수 있습니다. 이에 대한 자세한 내용은 AppScore 설명서의 계획 설정을 구성하는 방법을 참조하십시오.</p>	<p>마이그레이션 엔지니어</p>
<p>전체 변환 보고서를 생성하십시오.</p>	<p>그룹 관리자 페이지를 열고 애플리케이션 변환 보고서 문서 만들기를 선택합니다. 이동 그룹을 선택한 다음 내보내기를 선택합니다. 그러면 각 이동 그룹의 세부 정보를 포함하여 변환을 요약하는 .docx 파일이 생성됩니다.</p> <p>샘플 애플리케이션 변환 보고서는 AppScore 웹 사이트의 샘플 애플리케이션 변환 보고서를 참조하십시오.</p>	<p>마이그레이션 엔지니어</p>

관련 리소스

- [애플리케이션 마이그레이션의 7R은 무엇인가요?](#)
- [자세히 살펴보기 AppScore](#)

- [AppScore AWS 마켓플레이스에서](#)

마이크로소프트 엑셀과 Python을 사용하여 AWS DMS 작업을 위한 AWS CloudFormation 템플릿을 생성합니다.

작성자: Venkata Naveen Koppula(AWS)

환경: PoC 또는 파일럿	소스: 자동화	대상: AWS 클라우드의 데이터베이스
R 유형: 해당 없음	워크로드: Microsoft	기술: 마이그레이션, 데이터베이스

요약

이 패턴은 Microsoft Excel과 Python을 사용하여 [AWS 데이터베이스 마이그레이션 서비스](#) (AWS DMS) 용 AWS CloudFormation 템플릿을 자동으로 생성하는 단계를 설명합니다.

AWS DMS를 사용하여 데이터베이스를 마이그레이션하려면 AWS DMS 작업을 프로비저닝하기 위한 AWS CloudFormation 템플릿을 만들어야 하는 경우가 많습니다. 이전에는 AWS CloudFormation 템플릿을 만들려면 JSON 또는 YAML 프로그래밍 언어에 대한 지식이 필요했습니다. 이 도구를 사용하면 Excel에 대한 기본 지식과 터미널 또는 명령 창을 사용하여 Python 스크립트를 실행하는 방법만 알면 됩니다.

이 도구는 마이그레이션할 테이블의 이름, AWS DMS 엔드포인트의 Amazon 리소스 이름(ARN), AWS DMS 복제 인스턴스가 포함된 Excel 워크북을 입력으로 가져옵니다. 그런 다음 도구는 필요한 AWS DMS 작업을 위한 AWS CloudFormation 템플릿을 생성합니다.

자세한 단계 및 배경 정보는 [AWS 데이터베이스 블로그의 Microsoft Excel을 사용하여 AWS DMS 작업을 위한 AWS CloudFormation 템플릿 만들기](#) 블로그 게시물을 참조하십시오.

사전 조건 및 제한 사항

사전 조건

- 활성 상태의 AWS 계정.
- Microsoft Excel 버전 2016 이상

- Python 버전 2.7 이상
- xldr Python 모듈(명령 프롬프트에 pip install xldr 명령을 사용하여 설치됨)
- AWS DMS 소스 및 대상 엔드포인트와 AWS DMS 복제 인스턴스

제한 사항

- 대상 엔드포인트에서 스키마, 테이블 및 관련 열의 이름이 소문자로 변환됩니다.
- 이 도구는 AWS DMS 엔드포인트 및 복제 인스턴스의 생성을 다루지 않습니다.
- 현재 이 도구는 각 AWS DMS 작업에 대해 하나의 스키마만 지원합니다.

아키텍처

소스 기술 스택

- 온프레미스 데이터베이스
- Microsoft Excel

대상 기술 스택

- AWS CloudFormation 템플릿
- AWS 클라우드의 데이터베이스

아키텍처

도구

- [Pycharm IDE](#) 또는 Python 버전 3.6을 지원하는 모든 통합 개발 환경(IDE)
- Microsoft Office 2016(Microsoft Excel용)

에픽

네트워크, AWS DMS 복제 인스턴스, 엔드포인트 구성

작업	설명	필요한 기술
필요한 경우 서비스 할당량 증가를 요청합니다.	필요한 경우 AWS DMS 작업에 대한 서비스 할당량 증가를 요청합니다.	일반 AWS
AWS 리전, Virtual Private Cloud(VPC), CIDR 범위, 가용 영역 및 서브넷을 구성합니다.		일반 AWS
AWS DMS 복제 인스턴스를 구성합니다.	AWS DMS 복제 인스턴스는 온프레미스와 AWS 데이터베이스 모두에 연결할 수 있습니다.	일반 AWS
AWS DMS 엔드포인트를 구성합니다.	소스와 대상 데이터베이스의 모두의 엔드포인트를 구성합니다.	일반 AWS

AWS DMS 작업 및 태그에 대한 워크시트를 준비합니다.

작업	설명	필요한 기술
테이블 목록을 구성합니다.	마이그레이션과 관련된 모든 테이블을 나열하세요.	데이터베이스
작업 워크시트를 준비합니다.	구성한 테이블 목록을 사용하여 Excel 워크시트를 준비합니다.	일반 AWS, Microsoft Excel
태그 워크시트를 준비합니다.	AWS DMS 작업에 연결할 AWS 리소스 태그를 자세히 설명합니다.	일반 AWS, Microsoft Excel

도구 다운로드 및 실행

작업	설명	필요한 기술
GitHub 리포지토리에서 템플릿 생성 도구를 다운로드하고 추출합니다.	GitHub 리포지토리: https://github.com/aws-samples/dms-cloudformation-templates-generator/	
도구를 실행합니다.	‘참조 및 도움말’에 나열된 블로그 게시물의 자세한 지침을 따르세요.	

관련 리소스

- [Microsoft Excel을 사용하여 AWS DMS 작업을 위한 AWS CloudFormation 템플릿 생성 \(블로그 게시물\)](#)
- [DMS CloudFormation 템플릿 생성기 \(GitHub 리포지토리\)](#)
- [Python 설명서](#)
- [xlrd 설명 및 다운로드](#)
- [AWS DMS 설명서](#)
- [AWS CloudFormation 설명서](#)

자동화된 포트폴리오 검색으로 시작하기

작성자: Pratik Chunawala(AWS) 및 Rodolfo Jr. Cerrada(AWS)

환경: 프로덕션	소스: 온프레미스	대상: 온프레미스
R 유형: 해당 없음	워크로드: 기타 모든 워크로드	기술: 마이그레이션

요약

포트폴리오를 평가하고 메타데이터를 수집하는 것은 애플리케이션과 서버를 Amazon Web Services(AWS) 클라우드로 마이그레이션할 때, 특히 서버가 300대 이상인 대규모 마이그레이션의 경우 매우 중요한 과제입니다. 자동화된 포트폴리오 검색 도구를 사용하면 사용자 수, 사용 빈도, 종속성, 애플리케이션 인프라 정보 등 애플리케이션에 대한 정보를 수집할 수 있습니다. 이 정보는 유사한 특성을 가진 애플리케이션의 우선 순위를 적절하게 지정하고 그룹화할 수 있도록 마이그레이션 변동을 계획할 때 필수적입니다. 검색 도구를 사용하면 포트폴리오 팀이 메타데이터를 수동으로 수집하는 대신 검색 도구의 결과를 검증할 수 있으므로 포트폴리오 팀과 애플리케이션 소유자 간의 커뮤니케이션이 간소화됩니다. 이 패턴은 자동 검색 도구를 선택할 때 고려해야 할 주요 고려 사항과 사용자 환경에서 자동 검색 도구를 배포하고 테스트하는 방법에 대한 정보를 설명합니다.

이 패턴에는 높은 수준의 활동에 대한 자체 체크리스트를 작성하기 위한 출발점이 되는 템플릿이 포함되어 있습니다. 체크리스트 옆에는 책임감 있고 설명적인 컨설팅된 정보(RACI) 지표를 위한 템플릿이 있습니다. 이 RACI 지표를 사용하여 체크리스트의 각 작업을 누가 담당하는지 결정할 수 있습니다.

에픽

검색 도구 선택

작업	설명	필요한 기술
검색 도구가 사용 사례에 적합한지 판단합니다.	검색 도구가 사용 사례에 가장 적합한 솔루션이 아닐 수도 있습니다. 검색 도구를 선택, 조달, 준비 및 배포하는 데 필요한 시간을 고려하세요. 사용자 환경에서 에이전트 없는 검색 도구로 검사 기기를 설정하거	마이그레이션 리드, 마이그레이션 엔지니어

작업	설명	필요한 기술
	<p>나 범위 내 모든 워크로드에 에이전트를 설치하는 데 4~8주가 소요될 수 있습니다. 일단 배포되면 검색 도구가 애플리케이션 워크로드를 스캔하고 애플리케이션 스택 분석을 수행하여 메타데이터를 수집하는 데 4~12주가 소요됩니다. 100대 미만의 서버를 마이그레이션하는 경우 자동 검색 도구를 사용하여 메타데이터를 배포하고 수집하는 데 필요한 시간보다 빠르게 메타데이터를 수동으로 수집하고 종속성을 분석할 수 있습니다.</p>	
<p>검색 도구를 선택합니다.</p>	<p>추가 정보 섹션에서 자동 검색 도구를 선택하기 위한 고려 사항을 검토합니다. 사용 사례에 맞는 검색 도구를 선택하기 위한 적절한 기준을 결정한 다음, 해당 기준에 따라 각 도구를 평가합니다. 자동 검색 도구의 전체 목록은 검색, 계획 및 권장 사항 마이그레이션 도구를 참조하세요.</p>	<p>마이그레이션 리드, 마이그레이션 엔지니어</p>

설치 준비하기

작업	설명	필요한 기술
<p>배포 전 체크리스트를 준비합니다.</p>	<p>도구를 배포하기 전에 완료해야 하는 작업의 체크리스트를 만듭니다. 예를 들어 Flexera</p>	<p>빌드 리드, 마이그레이션 엔지니어, 마이그레이션 리드, 네트워크 관리자</p>

작업	설명	필요한 기술
	설명서 웹사이트의 배포 전 체크리스트 를 참조하세요.	
네트워크 요구 사항을 준비합니다.	도구를 실행하고 대상 서버에 액세스하는 데 필요한 포트, 프로토콜, IP 주소 및 라우팅을 제공합니다. 자세한 내용은 해당하는 검색 도구의 설치 안내서를 참조하세요. 예를 들어, 보려면 Flexera 설명서 웹사이트의 배포 요구 사항 을 참조하세요.	마이그레이션 엔지니어, 네트워크 관리자, 클라우드 아키텍트
계정 및 보안 인증 요구 사항을 준비합니다.	대상 서버에 액세스하고 도구의 모든 구성 요소를 설치하는 데 필요한 보안 인증을 식별합니다.	클라우드 관리자, 일반 AWS, 마이그레이션 엔지니어, 마이그레이션 리드, 네트워크 관리자, AWS 관리자
도구를 설치할 기기를 준비합니다.	도구 구성 요소를 설치할 기기가 도구의 사양 및 플랫폼 요구 사항을 충족하는지 확인합니다.	마이그레이션 엔지니어, 마이그레이션 리드, 네트워크 관리자
변경 주문을 준비합니다.	조직의 변경 관리 프로세스에 따라 필요한 모든 변경 지시를 준비하고 이러한 변경 지시가 승인되었는지 확인합니다.	빌드 책임자, 마이그레이션 책임자
이해관계자에게 요구사항을 전달합니다.	배포 전 체크리스트와 네트워크 요구 사항을 이해 관계자에게 보냅니다. 이해 관계자는 배포를 진행하기 전에 필요한 요구 사항을 검토, 평가 및 준비해야 합니다.	빌드 책임자, 마이그레이션 책임자

도구 배포

작업	설명	필요한 기술
설치 관리자를 다운로드합니다.	설치 프로그램 또는 가상 머신 이미지를 다운로드합니다. 가상 머신 이미지는 일반적으로 개방형 가상화 형식(OVF)으로 제공됩니다.	빌드 책임자, 마이그레이션 책임자
파일의 압축을 풉니다.	설치 프로그램을 사용하는 경우 온프레미스 서버에서 설치 프로그램을 다운로드하여 실행해야 합니다.	빌드 책임자, 마이그레이션 책임자
서버에 도구를 배포합니다.	다음과 같이 대상 온프레미스 서버에 검색 도구를 배포합니다. <ul style="list-style-type: none"> 소스 파일이 가상 머신 이미지인 경우 이를 VMware와 같은 가상 머신 환경에 배포합니다. 소스 파일이 설치 프로그램인 경우 설치 프로그램을 실행하여 도구를 설치하고 설정합니다. 	빌드 책임자, 마이그레이션 책임자, 네트워크 관리자
검색 도구에 로그인합니다.	화면에 나타나는 메시지에 따라 로그인하고 도구를 시작합니다.	마이그레이션 리더, 빌드 리더
제품을 활성화합니다.	라이선스 키를 입력합니다.	빌드 책임자, 마이그레이션 책임자
도구를 구성합니다.	Windows, VMware, 단순 네트워크 관리 프로토콜(SNMP), 보안 셸 프로토콜(SSH) 또는 데	빌드 책임자, 마이그레이션 책임자

작업	설명	필요한 기술
	이터베이스의 보안 인증 정보와 같이 대상 서버에 액세스하는 데 필요한 모든 보안 인증 정보를 입력합니다.	

도구 테스트

작업	설명	필요한 기술
테스트 서버를 선택합니다.	검색 도구를 테스트하는 데 사용할 수 있는 소량의 비프로덕션 서브넷 또는 IP 주소를 식별합니다. 이를 통해 스캔을 신속하게 검증하고, 오류를 신속하게 식별하여 문제를 해결하고, 테스트를 프로덕션 환경과 분리할 수 있습니다.	빌드 책임자, 마이그레이션 책임자, 네트워크 관리자
선택한 테스트 서버 스캔을 시작합니다.	에이전트가 없는 검색 도구의 경우 검색 도구 콘솔에 선택한 테스트 서버의 서브넷 또는 IP 주소를 입력하고 스캔을 시작합니다. 에이전트 기반 검색 도구의 경우 선택한 테스트 서버에 에이전트를 설치합니다.	빌드 책임자, 마이그레이션 책임자, 네트워크 관리자
검색 결과를 검토합니다.	테스트 서버의 스캔 결과를 검토합니다. 오류가 발견되면 문제를 해결하고 오류를 수정합니다. 오류와 해결 방법을 문서화합니다. 나중에 이 정보를 참조할 수 있으며 이 정보를 포트	빌드 책임자, 마이그레이션 책임자, 네트워크 관리자

작업	설명	필요한 기술
	폴리오 런북에 추가할 수 있습니다.	
테스트 서버를 다시 스캔합니다.	재스캔이 완료되면 오류가 없을 때까지 스캔을 반복합니다.	빌드 책임자, 마이그레이션 책임자, 네트워크 관리자

관련 리소스

AWS 리소스

- [AWS 클라우드 마이그레이션을 위한 애플리케이션 포트폴리오 평가 가이드](#)
- [검색, 계획 및 추천 마이그레이션 도구](#)

일반적으로 선택된 검색 도구에 대한 배포 가이드

- [RN150 가상 기기 배포](#)(Flexera 설명서)
- [수집기 설치](#)(modelizeIT 설명서)
- [온프레미스 분석 서버 설치](#)(modelizeIT 설명서)

추가 정보

자동 검색 도구 선택 시 고려 사항

각 검색 도구에는 이점과 한계가 있습니다. 사용 사례에 적합한 도구를 선택할 때 다음 항목을 고려하세요.

- 포트폴리오 평가 목표를 달성하는 데 필요한 메타데이터의 전부는 아니더라도 대부분을 수집할 수 있는 검색 도구를 선택합니다.
- 도구가 지원하지 않아서 수동으로 수집해야 하는 메타데이터를 식별합니다.
- 이해 관계자에게 검색 도구 요구 사항을 제공하여 이해 관계자가 내부 보안 및 규정 준수 요구 사항(예: 서버, 네트워크 및 보안 인증 요구 사항)을 기반으로 도구를 검토하고 평가할 수 있도록 합니다.
 - 도구를 사용하려면 범위 내 워크로드에 에이전트를 설치해야 합니까?
 - 도구를 사용하려면 사용자 환경에 가상 기기를 설정해야 합니까?

- 데이터 레지던시 요구 사항을 결정합니다. 일부 조직에서는 데이터를 환경 외부에 저장하고 싶어하지 않습니다. 이 문제를 해결하려면 도구의 일부 구성 요소를 온프레미스 환경에 설치해야 합니다.
- 도구가 범위 내 워크로드의 운영 체제(OS) 및 OS 버전을 지원하는지 확인합니다.
- 포트폴리오에 메인프레임, 미드레인지, 레거시 서버가 포함되는지 확인합니다. 대부분의 검색 도구는 이러한 워크로드를 종속성으로 감지할 수 있지만 일부 도구는 사용자 및 서버 종속성과 같은 디바이스 세부 정보를 가져오지 못할 수 있습니다. Device42와 modernizeIT 검색 도구는 모두 메인프레임 및 미드레인지 서버를 지원합니다.

첨부

이 문서와 관련된 추가 콘텐츠에 액세스하려면 [attachment.zip](#) 파일의 압축을 풉니다.

온프레미스 Cloudera 워크로드를 AWS의 Cloudera 데이터 플랫폼으로 마이그레이션

작성자: Battulga Purevragchaa(AWS), Nijjwol Lamsal(Cloudera, Inc.) 및 Nidhi Gupta(AWS)

환경: PoC 또는 파일럿	소스: Cloudera 워크로드	대상: Cloudera Data Platform (CDP) Public Cloud
R 유형: 해당 없음	워크로드: 기타 모든 워크로드	기술: 마이그레이션, 빅 데이터, 데이터베이스, 분석
AWS 서비스: Amazon EC2, Amazon EKS, AWS Identity and Access Management, Amazon S3, Amazon RDS		

요약

이 패턴은 온프레미스 Cloudera Distributed Hadoop(CDH), Hortonworks Data Platform(HDP) 및 Cloudera Data Platform(CDP) 워크로드를 AWS의 CDP Public Cloud로 마이그레이션하기 위한 상위 단계를 설명합니다. Cloudera Professional Services 및 시스템 통합업체(SI)와 협력하여 이러한 단계를 구현하는 것이 좋습니다.

Cloudera 고객이 온프레미스 CDH, HDP 및 CDP 워크로드를 클라우드로 이전하려는 데에는 여러 가지 이유가 있습니다. 몇 가지 일반적인 이유는 다음과 같습니다.

- 데이터 레이크하우스 또는 데이터 메시와 같은 새로운 데이터 플랫폼 패러다임의 채택 간소화
- 비즈니스 민첩성 향상, 기존 데이터 자산에 대한 액세스 및 추론 민주화
- 총 소유 비용(TCO) 절감
- 워크로드 탄력성 강화
- 기존 온프레미스 설치 기반에 비해 확장성 향상, 데이터 서비스 프로비저닝 시간 대폭 단축
- 레거시 하드웨어 사용 중지, 하드웨어 교체 주기 대폭 단축
- 클라우데라 라이선스 모델 (CCU) 을 통해 AWS의 클라우데라 워크로드로 확대되는 pay-as-you-go 요금 혜택을 누리세요.

- 지속적 통합 및 지속적 전달(CI/CD) 플랫폼을 통한 더 빠른 배포와 개선된 통합을 활용하십시오.
- 여러 워크로드에 단일 통합 플랫폼(CDP) 사용

Cloudera는 기계 학습, 데이터 엔지니어링, 데이터 웨어하우스, 운영 데이터베이스, CSP(스트림 프로세싱), 데이터 보안 및 거버넌스를 포함한 모든 주요 워크로드를 지원합니다. Cloudera는 수년 동안 온프레미스에서 이러한 워크로드를 제공해 왔으며, 워크로드 관리자 및 Replication Manager와 함께 CDP Public Cloud를 사용하면 이러한 워크로드를 AWS 클라우드로 마이그레이션할 수 있습니다.

Cloudera Shared Data Experience(SDX)는 이러한 워크로드 전반에 걸쳐 공유 메타데이터 카탈로그를 제공하여 일관된 데이터 관리 및 운영을 지원합니다. 또한 SDX에는 위협으로부터 보호하기 위한 포괄적이고 세분화된 보안과 결제 카드 산업 데이터 보안 표준(PCI DSS) 및 GDPR과 같은 표준 준수를 위한 감사 및 검색 기능을 위한 통합 거버넌스가 포함되어 있습니다.

CDP 마이그레이션 개요

	소스 워크로드	CDH, HDP 및 CDP 프라이빗 클라우드
워크로드	소스 환경	<ul style="list-style-type: none"> • Windows, Linux • 온프레미스, 콜로케이션 또는 AWS가 아닌 모든 환경
	대상 워크로드	AWS 기반 CDP 퍼블릭 클라우드
	대상 환경	<ul style="list-style-type: none"> • 배포 모델: 고객 계정 • 운영 모델: 고객/Cloudera 컨트롤 플레인
마이그레이션	마이그레이션 전략(7Rs)	리호스팅, 리플랫폼 또는 리팩터링
	워크로드 버전의 업그레이드입니까?	예
	마이그레이션 기간	<ul style="list-style-type: none"> • 배포: 고객 계정, Virtual Private Cloud(VPC) 및 CDP Public Cloud 고객 관리형 환

경을 만드는 데 약 1주가 소요됩니다.

- 마이그레이션 기간: 워크로드의 복잡성과 규모에 따라 1~4개월.

비용

AWS에서 워크로드를 실행하는 데 드는 비용

- 높은 수준에서 SAS 워크로드를 AWS로 마이그레이션하는 비용에는 AWS에 새로운 환경을 구축하는 것을 전제로 합니다. 여기에는 직원의 시간과 노력을 고려하는 것은 물론 새로운 환경을 위한 컴퓨팅 리소스 프로비저닝 및 소프트웨어 라이선싱도 포함됩니다.
- Cloudera 클라우드 사용량 기반 요금 모델은 세분화 및 규모 자동 조정 기능을 활용할 수 있는 유연성을 제공합니다. 자세한 내용은 Cloudera 웹사이트의 [CDP Public Cloud 서비스 요금](#)을 참조하십시오.
- Cloudera Enterprise [Data Hub](#)는 Amazon Elastic Compute Cloud(Amazon EC2)를 기반으로 하며 기존 클러스터를 밀접하게 모델링합니다. 데이터 허브는 [사용자 지정](#)할 수 있지만 이는 비용에 영향을 미칩니다.
- [CDP Public Cloud 데이터 웨어하우스](#), [Cloudera 기계 학습](#) 및 [Cloudera Data Engineering\(CDE\)](#)은 컨테이너 기반이며 자동으로 규모를 조정하도록 구성할 수 있습니다.

	시스템 요구 사항	사전 조건 섹션을 참조하십시오.
인프라 계약 및 프레임워크	SLA	CDP 퍼블릭 클라우드에 대한 Cloudera 서비스 수준 계약 을 참조하십시오.
	DR	Cloudera 설명서의 재해 복구 를 참조하십시오.
규정 준수	라이선스 및 운영 모델(대상 AWS 계정용)	기존 보유 라이선스 사용 (BYOL) 모델
	보안 요구 사항	Cloudera 설명서에서 Cloudera 보안 개요 를 참조하십시오.
	기타 규정 준수 인증	Cloudera 웹사이트에서 일반 데이터 보호 규정(GDPR) 규정 준수 및 CDP Trust Center 에 대한 정보를 참조하십시오.

사전 조건 및 제한 사항

사전 조건

- 계정, 리소스, 서비스, 권한을 포함한 [AWS 계정 요구 사항](#)(예: AWS Identity and Access Management(IAM) 역할 및 정책 설정)
- Cloudera 웹사이트에서 [CDP를 배포하기 위한 사전 조건](#)

마이그레이션에는 다음과 같은 역할과 전문 지식이 필요합니다.

역할	기술 및 책임
마이그레이션 책임자	경영진 지원, 팀 협업, 계획, 구현 및 평가 보장
Cloudera SME	CDH, HDP, CDP 관리, 시스템 관리 및 아키텍처 분야의 전문 기술

AWS 아키텍트

AWS 서비스, 네트워킹, 보안 및 아키텍처 관련
기술

아키텍처

적절한 아키텍처를 구축하는 것은 마이그레이션과 성능이 기대에 부합하는지 확인하는 중요한 단계입니다. 이 플레이북의 가정을 충족하기 위한 마이그레이션 노력을 위해서는 Virtual Private Cloud(VPC) 호스팅 인스턴스든 CDP든 AWS 클라우드의 대상 데이터 환경이 운영 체제 및 소프트웨어 버전 및 주요 시스템 사양 측면에서 소스 환경과 동일해야 합니다.

다음 다이어그램([Cloudera Shared Data Experience 데이터시트](#)의 허가를 받아 재현)은 CDP 환경의 인프라 구성 요소와 계층 또는 인프라 구성 요소가 상호 작용하는 방식을 보여줍니다.

아키텍처에는 다음 CDP 구성 요소가 포함되어 있습니다.

- Data Hub는 Cloudera Runtime으로 구동되는 워크로드 클러스터를 시작하고 관리하기 위한 서비스입니다. Data Hub의 클러스터 정의를 사용하여 사용자 지정 사용 사례에 맞게 워크로드 클러스터를 프로비저닝 및 액세스하고 사용자 지정 클러스터 구성을 정의할 수 있습니다. 자세한 내용은 [Cloudera 웹사이트](#)를 참조하십시오.
- 데이터 흐름 및 스트리밍은 기업이 데이터를 이동할 때 직면하는 주요 문제를 해결합니다. 다음 작업을 관리합니다.
 - 대용량 및 대규모로 실시간 데이터 스트리밍을 처리합니다.
 - 데이터 출처 및 스트리밍 데이터의 계보 추적
 - 엣지 애플리케이션 및 스트리밍 소스 관리 및 모니터링

[자세한 내용은 클라우데라 웹 사이트의 클라우데라 및 CSP를 DataFlow 참조하십시오.](#)

- 데이터 엔지니어링에는 조직이 데이터 파이프라인과 워크플로를 구축하고 유지하는 데 도움이 되는 데이터 통합, 데이터 품질, 데이터 거버넌스가 포함됩니다. 자세한 내용은 [Cloudera 웹사이트](#)를 참조하십시오. Cloudera Data Engineering 워크로드에 대한 [AWS의 비용 절감을 촉진하기 위한 스팟 인스턴스 지원](#)에 대해 알아보십시오.
- 데이터 웨어하우스를 사용하면 워크로드 수요에 맞게 자동으로 확장되는 독립적인 데이터 웨어하우스와 데이터 마트를 만들 수 있습니다. 이 서비스는 각 데이터 웨어하우스 및 데이터 마트에 대해 격리된 컴퓨팅 인스턴스와 자동화된 최적화를 제공하며 SLA를 충족하는 동시에 비용을 절감할 수 있

도록 도와줍니다. 자세한 내용은 [Cloudera 웹사이트](#)를 참조하십시오. AWS의 Cloudera 데이터 웨어하우스에 대한 [비용 관리](#) 및 [자동 규모 조정](#)에 대해 알아보십시오.

- CDP의 운영 데이터베이스는 확장 가능한 고성능 애플리케이션을 위한 안정적이고 유연한 기반을 제공합니다. 통합 운영 및 웨어하우스 플랫폼 내에서 기존의 정형 데이터와 새로운 비정형 데이터를 함께 제공하는 상시 사용 가능하고 확장 가능한 실시간 데이터베이스를 제공합니다. 자세한 내용은 [Cloudera 웹사이트](#)를 참조하십시오.
- 기계 학습은 셀프 서비스 데이터 과학 및 데이터 엔지니어링 기능을 엔터프라이즈 데이터 클라우드 내에서 하나의 휴대용 서비스로 통합하는 클라우드 네이티브 기계 학습 플랫폼입니다. 이를 통해 어디서나 데이터에 기계 학습과 인공지능을 확장 가능하게 배포할 수 있습니다. 자세한 내용은 [Cloudera 웹사이트](#)를 참조하십시오.

AWS 기반 CDP

다음 다이어그램(Cloudera 웹사이트의 허가를 받아 수정)은 AWS 기반 CDP의 상위 수준 아키텍처를 보여줍니다. CDP는 [자체 보안 모델](#)을 구현하여 계정과 데이터 흐름을 모두 관리합니다. 이는 [크로스 계정 역할](#)을 사용하여 [IAM](#)과 통합됩니다.

CDP 컨트롤 플레인인 자체 VPC의 Cloudera 마스터 계정에 있습니다. 각 고객 계정에는 고유한 하위 계정과 고유한 VPC가 있습니다. 크로스 계정 IAM 역할 및 SSL 기술은 컨트롤 플레인에서 들어오고 나가는 관리 트래픽을 각 고객 VPC 내의 인터넷 라우팅 가능한 퍼블릭 서브넷에 있는 고객 서비스로 라우팅합니다. 고객의 VPC에서 Cloudera Shared Data Experience (SDX)는 통합 거버넌스 및 규정 준수를 통해 엔터프라이즈급 보안을 제공하므로 데이터에서 더 빠르게 통찰력을 얻을 수 있습니다. SDX는 모든 Cloudera 제품에 통합된 설계 철학입니다. [SDX](#) 및 [AWS용 CDP Public Cloud 네트워크 아키텍처](#)에 대한 자세한 내용은 Cloudera 설명서를 참조하십시오.

도구

서비스

- [Amazon Elastic Compute Cloud\(Amazon EC2\)](#)는 AWS 클라우드에서 확장 가능한 컴퓨팅 용량을 제공합니다. 필요한 만큼 가상 서버를 시작하고 빠르게 스케일 업하거나 스케일 다운할 수 있습니다.
- [Amazon Elastic Kubernetes Service\(Amazon EKS\)](#)는 자체 Kubernetes 컨트롤 플레인 또는 노드를 설치하거나 유지 관리할 필요 없이 AWS의 Kubernetes를 실행하는 데 도움이 됩니다.
- [AWS Identity and Access Management\(IAM\)](#)를 사용하면 사용자에 대해 인증 및 권한 부여를 제어함으로써 AWS 리소스에 대한 액세스를 안전하게 관리할 수 있습니다.

- [Amazon Relational Database Service\(Amazon RDS\)](#)는 AWS 클라우드에서 관계형 데이터베이스를 설정, 운영 및 조정하는 데 도움이 됩니다.
- [Amazon Simple Storage Service\(S3\)](#)는 원하는 양의 데이터를 저장, 보호 및 검색하는 데 도움이 되는 클라우드 기반 객체 스토리지 서비스입니다.

자동화 및 툴링

- 추가 툴링이 필요한 경우 [Cloudera Backup Data Recovery \(BDR\)](#), [AWS Snowball](#) 및 [AWS Snowmobile](#)을 사용하여 온프레미스 CDH, HDP 및 CDP에서 AWS 호스팅 CDP로 데이터를 마이그레이션하는 데 도움을 줄 수 있습니다.
- 새로 배포하는 경우 CDP용 [AWS 파트너 솔루션](#)을 사용하는 것이 좋습니다.

에픽

마이그레이션 준비

작업	설명	필요한 기술
Cloudera 팀과 소통하십시오.	<p>Cloudera는 고객과 함께 표준화된 참여 모델을 추구하며 시스템 통합업체(SI)와 협력하여 동일한 접근 방식을 장려할 수 있습니다. Cloudera 고객 팀에 문의하면 프로젝트를 시작하는 데 필요한 지침과 필요한 기술 리소스를 제공받을 수 있습니다. Cloudera 팀에 문의하면 필요한 모든 팀이 마이그레이션 날짜가 다가옴에 따라 마이그레이션을 준비할 수 있습니다.</p> <p>Cloudera Professional Services에 문의하여 Cloudera 배포를 파일럿 환경에서 프로덕션 환경으로 신속하게 이전하여 비용을 절감</p>	마이그레이션 책임자

작업	설명	필요한 기술
	<p>하고 성능을 극대화할 수 있습니다. 전체 서비스 목록은 Cloudera 웹사이트를 참조하십시오.</p>	
<p>AWS에서 VPC를 위한 CDP Public Cloud 환경을 생성합니다.</p>	<p>Cloudera Professional Services 또는 SI와 협력하여 CDP 퍼블릭 클라우드를 계획하고 AWS 기반 VPC에 배포합니다.</p>	<p>클라우드 아키텍트, Cloudera SME</p>

작업	설명	필요한 기술
<p>마이그레이션을 위한 워크로드의 우선순위를 정하고 평가합니다.</p>	<p>모든 온프레미스 워크로드를 평가하여 마이그레이션하기 가장 쉬운 워크로드를 결정합니다. 업무상 중요하지 않은 애플리케이션은 고객에게 미치는 영향이 최소화되므로 먼저 이동하는 것이 가장 좋습니다. 중요한 워크로드는 다른 워크로드를 성공적으로 마이그레이션한 후 마지막에 사용할 수 있도록 저장합니다.</p> <p>참고: 임시(CDP 데이터 엔지니어링) 워크로드는 영구(CDP 데이터 웨어하우스) 워크로드보다 마이그레이션하기가 더 쉽습니다. 마이그레이션할 때 데이터 양과 위치를 고려하는 것도 중요합니다. 온프레미스 환경에서 클라우드로 데이터를 지속적으로 복제하고 데이터를 클라우드로 직접 가져오도록 데이터 수집 파이프라인을 변경하는 것이 문제일 수 있습니다.</p>	<p>마이그레이션 책임자</p>

작업	설명	필요한 기술
<p>CDH, HDP, CDP 및 레거시 애플리케이션 마이그레이션 활동에 대해 논의합니다.</p>	<p>Cloudera 워크로드 매니저와 함께 다음 활동을 고려하고 계획을 시작합니다.</p> <ul style="list-style-type: none"> • AWS 환경에 복사할 데이터 및 워크로드 • 클라우드에서 바로 사용할 수 있는 데이터 • 리소스를 소모하고 다른 테넌트에게 문제를 야기하는 시끄러운 이웃 • 탄력적 워크로드 • 운영 오버헤드가 높은 소규모 클러스터 	<p>마이그레이션 책임자</p>

작업	설명	필요한 기술
<p>Cloudera Replication Manager 요구 사항 및 권장 사항을 완료합니다.</p>	<p>Cloudera Professional Services 및 SI와 협력하여 AWS의 CDP Public Cloud 환경으로 워크로드를 마이그레이션할 준비를 합니다. 다음 요구 사항 및 권장 사항을 이해하면 Replication Manager 서비스를 설치하는 동안과 설치 후에 흔히 발생하는 문제를 방지하는데 도움이 될 수 있습니다.</p> <ul style="list-style-type: none"> • Replication Manager 지원 문서를 검토하여 환경 및 시스템 요구 사항을 충족하는지 확인합니다. 자세한 내용은 Cloudera 웹사이트의 CDP Public Cloud Replication Manager 지원 매트릭스를 참조하십시오. • Replication Manager 앱 및 Data Lifecycle Manager (DLM) 엔진을 설치할 노드에는 루트 액세스 권한이 필요하지 않습니다. • 향후에 하이브 복제를 사용하지 않을 것이 확실하다면 Replication Manager를 처음 설치할 때 Apache Hive를 설치합니다. Replication Manager에서 HDFS 복제 정책을 생성한 후 Hive를 설치하려면 Hive를 추가한 후 모든 HDFS 복제 정책을 삭제 	<p>마이그레이션 책임자</p>

작업	설명	필요한 기술
	<p>한 다음 다시 생성해야 합니다.</p> <ul style="list-style-type: none"> Replication Manager에서 사용되는 클러스터는 대칭 구성을 가져야 합니다. 복제 관계의 각 클러스터는 보안 (Kerberos), 사용자 관리 (LDAP/AD) 및 Knox 프록시에 대해 정확히 동일하게 구성되어야 합니다. Hadoop 분산 파일 시스템(HDFS), Apache Hive, Apache Knox, Apache Range 및 Apache Atlas와 같은 클러스터 서비스는고가용성(HA)을 위해 다른 구성을 가질 수 있습니다. 예를 들어 소스 클러스터와 대상 클러스터에는 별도의 HA 구성과 비 HA 구성이 있을 수 있습니다. 	

CDP를 AWS로 마이그레이션

작업	설명	필요한 기술
<p>Cloudera Workload Manager를 사용하여 개발/테스트 환경을 위한 첫 번째 워크로드를 마이그레이션합니다.</p>	<p>SI는 첫 번째 워크로드를 AWS 클라우드로 마이그레이션하는데 도움을 줄 수 있습니다. 이는 고객을 대상으로 하거나 업무상 중요하지 않은 애플리케이션이어야 합니다. 개발 및 테스트 마이그레이션의 이상적인 대상은 CDP 데이터 엔지니어링 워크로드와 같이 클라우드</p>	<p>마이그레이션 책임자</p>

작업	설명	필요한 기술
	<p>에서 쉽게 수집할 수 있는 데이터가 있는 애플리케이션입니다. 이는 중단 없는 액세스가 필요한 사용자가 많은 CDP 데이터 웨어하우스 워크로드와 같은 영구 워크로드에 비해 일반적으로 액세스하는 사용자 수가 적은 일시적인 워크로드입니다. 데이터 엔지니어링 워크로드는 지속적이지 않으므로 문제가 발생할 경우 비즈니스에 미치는 영향을 최소화합니다. 하지만 이러한 작업은 프로덕션 보고에 매우 중요할 수 있으므로 영향이 적은 데이터 엔지니어링 워크로드에 우선 순위를 둡니다.</p>	

작업	설명	필요한 기술
<p>필요에 따라 마이그레이션 단계를 반복합니다.</p>	<p>Cloudera Workload Manager는 클라우드에 가장 적합한 워크로드를 식별하는 데 도움이 됩니다. 클라우드 성능 등급, 대상 환경의 규모/용량 계획, 복제 계획과 같은 지표를 제공합니다. 마이그레이션하기에 가장 적합한 대상은 계절별 워크로드, 임시 보고, 리소스를 많이 소비하지 않는 간헐적 작업입니다.</p> <p>Cloudera Replication Manager는 데이터를 온프레미스에서 클라우드로, 클라우드로서 온프레미스로 이동합니다.</p> <p>Workload Manager를 사용하여 데이터 웨어하우징, 데이터 엔지니어링 및 기계 학습을 위한 워크로드, 애플리케이션, 성능 및 인프라 용량을 사전에 최적화합니다. 데이터 웨어하우스를 현대화하는 방법에 대한 전체 가이드는 Cloudera 웹사이트를 참조하십시오.</p>	<p>Cloudera SME</p>

관련 리소스

Cloudera 설명서:

- [CDP, Cloudera 관리자, 복제 관리자에 클래식 클러스터 등록:](#)
 - [관리 콘솔](#)
 - [복제 관리자 하이브 복제](#)

- [센트리 복제](#)
- [센트리 권한](#)
- [Data Hub 클러스터 계획 체크리스트](#)
- [워크로드 관리자 아키텍처](#)
- [복제 관리자 요구 사항](#)
- [Cloudera 데이터 플랫폼 관찰성](#)
- [AWS 요구 사항](#)

AWS 설명서:

- [클라우드 데이터 마이그레이션](#)

RHEL 소스 서버를 재부팅한 후 SELinux를 비활성화하지 않고 Replication Agent를 자동으로 다시 시작

작성자: Anil Kunapareddy, Shanmugam Shanker, Venkatramana Chintha

환경: 프로덕션

기술: 마이그레이션, 운영 체제

워크로드: 오픈 소스

AWS 서비스: AWS 애플리케이션 마이그레이션 서비스

요약

AWS 애플리케이션 마이그레이션 서비스는 Red Hat Enterprise Linux(RHEL) 워크로드를 Amazon Web Services(AWS) 클라우드로 마이그레이션하는 작업을 단순화하고, 가속화하며, 자동화하는 데 도움이 됩니다. 소스 서버를 애플리케이션 마이그레이션 서비스에 추가하려면 AWS Replication Agent를 서버에 설치합니다.

애플리케이션 마이그레이션 서비스는 실시간, 비동기식, 블록 수준 복제를 제공합니다. 즉, 전체 복제 프로세스 동안 정상적인 IT 작업을 계속할 수 있습니다. 이러한 IT 작업을 수행하려면 마이그레이션 중에 RHEL 소스 서버를 재부팅하거나 다시 시작해야 할 수 있습니다. 이 경우 AWS Replication Agent가 자동으로 다시 시작되지 않고 데이터 복제가 중지됩니다. 일반적으로 Security-Enhanced Linux(SELinux)를 비활성화 또는 허용 모드로 설정하여 AWS Replication Agent를 자동으로 다시 시작할 수 있습니다. 하지만 조직의 보안 정책에 따라 SELinux를 비활성화하는 것이 금지될 수 있으며, [파일 레이블을 다시 지정해야 할 수도 있습니다](#).

이 패턴은 마이그레이션 중에 RHEL 소스 서버가 재부팅되거나 재시작될 때 SELinux를 끄지 않고 AWS Replication Agent를 자동으로 다시 시작하는 방법을 설명합니다.

사전 조건 및 제한 사항

사전 조건

- 활성 상태의 AWS 계정.
- 클라우드로 마이그레이션하려는 온프레미스 RHEL 워크로드.

- 애플리케이션 마이그레이션 서비스 콘솔에서 초기화되는 애플리케이션 마이그레이션 서비스. 초기화는 이 서비스를 처음 사용할 때만 필요합니다. 지침은 [Application Migration Service Catalog 설명서](#)를 참조하십시오.
- 애플리케이션 마이그레이션 서비스에 대한 기존 [Identity and Access Management\(IAM\) 정책](#). 자세한 내용은 [Application Migration Service 설명서](#)를 참조하세요.

버전

- RHEL 버전 7 이상

도구

서비스

- [AWS 애플리케이션 마이그레이션 서비스](#)는 AWS로 애플리케이션을 마이그레이션하는 작업을 간소화, 가속화하고 비용을 절감하는 고도로 자동화된 lift-and-shift (재호스팅) 솔루션입니다.

Linux 명령

다음 표에는 RHEL 소스 서버에서 실행할 Linux 명령 목록이 나와 있습니다. 이러한 내용은 이 패턴에 대한 에픽과 스토리에도 설명되어 있습니다.

명령	설명
<code>#systemctl -version</code>	시스템 버전을 식별합니다.
<code>#systemctl list-units --type=service</code>	RHEL 서버에서 사용할 수 있는 모든 활성 서비스를 나열합니다.
<code>#systemctl list-units --type=service grep running</code>	현재 RHEL 서버에서 실행 중인 모든 서비스를 나열합니다.
<code>#systemctl list-units --type=service grep failed</code>	RHEL 서버가 재부팅되거나 다시 시작된 후 로드되지 않은 모든 서비스를 나열합니다.
<code>restorecon -Rv /etc/rc.d/init.d/aws-replication-service</code>	컨텍스트를 <code>aws-replication-service</code> (으)로 변경합니다.

<code>yum install policycoreutils*</code>	SELinux 시스템 운영에 필요한 정책 코어 유틸리티를 설치합니다.
<code>ausearch -c "insmod" --raw audit2allow -M my-modprobe</code>	감사 로그를 검색하고 정책용 모듈을 생성합니다.
<code>semodule -i my-modprobe.pp</code>	정책을 활성화합니다.
<code>cat my-modprobe.te</code>	<code>my-modprobe.te</code> 파일의 내용을 표시합니다.
<code>semodule -l grep my-modprobe</code>	정책이 SELinux 모듈에 로드되었는지 확인합니다.

에픽

Replication Agent를 설치하고 RHEL 소스 서버를 재부팅합니다.

작업	설명	필요한 기술
액세스 키와 비밀 액세스 키를 사용하여 애플리케이션 마이그레이션 서비스 사용자를 생성합니다.	Replication Agent를 설치하려면 필수 AWS 자격 증명을 사용하여 애플리케이션 마이그레이션 서비스 사용자를 생성해야 합니다. 지침은 Application Migration Service Catalog 설명서 를 참조하십시오.	마이그레이션 엔지니어
AWS Replication Agent를 설치합니다.	<ol style="list-style-type: none"> 1. Management Console에 로그인하여 https://console.aws.amazon.com/mgn/home에서 마이그레이션 서비스 콘솔을 엽니다. 2. 애플리케이션 마이그레이션 서비스 설명서의 지침에 따라 복제 설정을 구성합니다. 3. 애플리케이션 마이그레이션 서비스 설명서의 지침에 따 	마이그레이션 엔지니어

작업	설명	필요한 기술
	<p>라 Replication Agent을 설치합니다.</p> <p>4. 소스 서버 페이지에서 RHEL 소스 서버를 선택한 다음 복제를 선택하여 초기 복제를 시작합니다. 자세한 내용은 Application Migration Service 설명서를 참조하세요.</p>	
<p>RHEL 소스 서버를 다시 시작하거나 재부팅합니다.</p>	<p>마이그레이션 대시보드에서 데이터 복제 상태가 정상으로 표시되면 RHEL 소스 서버를 다시 시작하거나 재부팅합니다.</p>	<p>마이그레이션 엔지니어</p>
<p>데이터 복제 상태를 확인합니다.</p>	<p>한 시간 정도 기다린 다음 마이그레이션 대시보드에서 데이터 복제 상태를 다시 확인합니다. 중단됨 상태여야 합니다.</p>	<p>마이그레이션 엔지니어</p>

RHEL 소스 서버에서 AWS Replication Agent 상태를 확인합니다.

작업	설명	필요한 기술
<p>시스템 버전을 식별합니다.</p>	<p>RHEL 소스 서버의 명령줄 인터페이스를 열고 다음 명령을 실행하여 시스템 버전을 확인합니다.</p> <pre>#systemctl -version</pre>	<p>마이그레이션 엔지니어</p>
<p>모든 활성 서비스를 나열합니다.</p>	<p>RHEL 서버에서 사용 가능한 모든 활성 서비스를 나열하려면 다음 명령을 실행합니다.</p>	<p>마이그레이션 엔지니어</p>

작업	설명	필요한 기술
실행 중인 모든 서비스를 나열합니다.	<pre>#systemctl list-units --type=service</pre> <p>현재 RHEL 서버에서 실행 중인 모든 서비스를 나열하려면 다음과 같은 명령을 사용합니다.</p> <pre>#systemctl list-units --type=service grep running</pre>	마이그레이션 엔지니어
로드에 실패한 모든 서비스를 나열합니다.	<p>RHEL 서버가 재부팅되거나 다시 시작된 후 로드되지 않은 모든 서비스를 나열하려면, 다음의 명령을 실행합니다.</p> <pre>#systemctl list-units --type=service grep failed</pre>	마이그레이션 엔지니어

SELinux 모듈 생성 및 실행

작업	설명	필요한 기술
보안 컨텍스트를 변경합니다.	<p>RHEL 소스 서버의 명령줄 인터페이스에서 다음 명령을 실행하여 보안 컨텍스트를 복제 서비스로 변경합니다.</p> <pre>restorecon -Rv /etc/rc.d/init.d/aws-replication-service</pre>	마이그레이션 엔지니어
코어 유틸리티를 설치합니다.	SELinux 시스템 및 정책 운영에 필요한 코어 유틸리티를 설	마이그레이션 엔지니어

작업	설명	필요한 기술
	<p>치하려면 다음 명령을 실행합니다.</p> <pre>yum install polycor eutils*</pre>	
<p>감사 로그를 검색하고 정책용 모듈을 생성합니다.</p>	<p>명령 실행:</p> <pre>ausearch -c "insmod" --raw audit2allow - M my-modprobe</pre>	<p>마이그레이션 엔지니어</p>
<p>파일 내용을 표시합니다. my-modprobe-te</p>	<p>my-modprobe.te 파일은 audit2allow 명령으로 생성됩니다. 여기에는 SELinux 도메인, 정책 소스 디렉터리, 하위 디렉터리가 포함되며 도메인과 관련된 액세스 벡터 규칙 및 전환이 지정됩니다. 파일의 내용을 표시하려면 다음 명령을 실행합니다.</p> <pre>cat my modprobe.te</pre>	<p>마이그레이션 엔지니어</p>
<p>정책을 활성화합니다.</p>	<p>모듈을 삽입하고 정책 패키지를 활성화하려면 다음 명령을 실행합니다.</p> <pre>semodule -i my-modpro be.pp</pre>	<p>마이그레이션 엔지니어</p>

작업	설명	필요한 기술
<p>모듈이 로드되었는지 확인합니다.</p>	<p>다음 명령을 실행합니다.</p> <pre>semodule -l grep my-modprobe</pre> <p>SELinux 모듈이 로드되면 마이그레이션 중에 더 이상 SELinux를 비활성화 또는 허용 모드로 설정할 필요가 없습니다.</p>	<p>마이그레이션 엔지니어</p>
<p>RHEL 소스 서버를 재부팅하거나 다시 시작하고 데이터 복제 상태를 확인합니다.</p>	<p>AWS Migration Service 콘솔을 열고 데이터 복제 진행 상황으로 이동한 다음 RHEL 소스 서버를 재부팅하거나 다시 시작합니다. 이제 RHEL 소스 서버가 재부팅되면 데이터 복제가 자동으로 재개되어야 합니다.</p>	<p>마이그레이션 엔지니어</p>

관련 리소스

- [Application Migration 서비스 설명서](#)
- [기술 교육 자료](#)
- [Replication Agent 문제 해결](#)
- [Application Migration Service 정책](#)

리아키텍트

주제

- [Oracle의 VARCHAR2\(1\) 데이터 유형을 Amazon Aurora PostgreSQL의 부울 데이터 유형으로 변환](#)
- [Aurora PostgreSQL 호환에서 애플리케이션 사용자 및 역할을 생성](#)
- [PostgreSQL-compatible Aurora 글로벌 데이터베이스를 사용하여 Oracle DR 에뮬레이션하기](#)
- [Oracle SQL Developer 및 AWS SCT를 사용하여 Amazon RDS for Oracle에서 Amazon RDS for PostgreSQL로 점진적으로 마이그레이션](#)
- [Aurora PostgreSQL-Compatible에서 파일 인코딩을 사용하여 BLOB 파일을 TEXT에 로드](#)
- [AWS DMS를 사용하여 SSL 모드에서 Amazon RDS for Oracle를 Amazon RDS for PostgreSQL로 마이그레이션](#)
- [AWS CLI와 AWS를 사용하여 AWS SCT 및 AWS DMS를 사용하여 오라클용 Amazon RDS를 PostgreSQL용 Amazon RDS로 마이그레이션합니다. CloudFormation](#)
- [Oracle SERIALLY_REUSABLE 프라그마 패키지를 PostgreSQL로 마이그레이션](#)
- [Oracle 외부 테이블을 Amazon Aurora PostgreSQL 호환으로 마이그레이션](#)
- [합수 기반 인덱스를 Oracle에서 PostgreSQL로 마이그레이션](#)
- [확장 기능을 사용하여 Oracle 네이티브 함수를 PostgreSQL로 마이그레이션](#)
- [AWS DMS를 사용하여 Db2 데이터베이스를 Amazon EC2에서 Aurora MySQL과 호환되는 Aurora 로 마이그레이션](#)
- [AWS DMS를 사용하여 Microsoft SQL 서버 데이터베이스를 Amazon EC2에서 Amazon DocumentDB로 마이그레이션](#)
- [온프레미스 ThoughtSpot 팔콘 데이터베이스를 Amazon Redshift로 마이그레이션하기](#)
- [AWS DMS를 사용하여 Amazon DynamoDB로 Oracle 데이터베이스 마이그레이션](#)
- [AWS DMS를 사용하여 Oracle 파티션형 테이블을 PostgreSQL로 마이그레이션하기](#)
- [Amazon RDS for Oracle에서 Amazon RDS for MySQL로 마이그레이션](#)
- [AWS DMS 및 AWS SCT를 사용하여 Amazon EC2의 IBM Db2에서 PostgreSQL과 호환되는 Aurora PostgreSQL로 마이그레이션하십시오.](#)
- [AWS DMS를 사용하여 Oracle 8i 또는 9i에서 PostgreSQL용 아마존 RDS로 마이그레이션하십시오. SharePlex](#)
- [구체화된 뷰와 AWS DMS를 사용하여 Oracle 8i 또는 9i에서 Amazon RDS for PostgreSQL로 마이그레이션](#)

- [AWS DMS 및 AWS SCT를 사용하여 Amazon EC2의 오라클에서 Amazon RDS for MySQL로 마이그레이션하십시오.](#)
- [AWS DMS를 사용하여 Oracle에서 Amazon DocumentDB로 마이그레이션](#)
- [AWS DMS 및 AWS SCT를 사용하여 Amazon EC2에서 Amazon RDS for MariaDB로 Oracle 데이터베이스 마이그레이션](#)
- [DMS 및 SCT를 사용하여 온프레미스 Oracle 데이터베이스를 Amazon RDS for MySQL로 마이그레이션](#)
- [Oracle bystander 및 AWS DMS를 사용하여 온프레미스 Oracle 데이터베이스를 Amazon RDS for PostgreSQL로 마이그레이션](#)
- [오라클을 사용하여 Oracle 데이터베이스에서 PostgreSQL용 Amazon RDS로 마이그레이션하십시오. GoldenGate](#)
- [AWS DMS 및 AWS SCT를 사용하여 Amazon Redshift로 Oracle 데이터베이스 마이그레이션](#)
- [AWS DMS 및 AWS SCT를 사용하여 Aurora PostgreSQL로 Oracle 데이터베이스를 마이그레이션하기](#)
- [Aurora PostgreSQL로 온프레미스 Oracle 데이터베이스의 데이터를 마이그레이션하기](#)
- [AWS DMS를 사용하여 SAP ASE에서 Amazon RDS for SQL Server로 마이그레이션](#)
- [AWS DMS를 사용하여 온프레미스 Microsoft SQL Server 데이터베이스를 Amazon Redshift로 마이그레이션](#)
- [SCT 데이터 추출 에이전트를 사용하여 온프레미스 Microsoft SQL Server 데이터베이스를 Amazon Redshift로 마이그레이션](#)
- [AWS SCT 데이터 추출 에이전트를 사용하여 Teradata 데이터베이스를 Amazon Redshift로 마이그레이션](#)
- [AWS SCT 데이터 추출 에이전트를 사용하여 온프레미스 Vertica 데이터베이스를 Amazon Redshift로 마이그레이션하기](#)
- [레거시 애플리케이션을 Oracle Pro*C에서 ECPG로 마이그레이션](#)
- [가상으로 생성된 열을 오라클에서 PostgreSQL로 마이그레이션](#)
- [Aurora PostgreSQL 호환에서 Oracle UTL_FILE 기능 설정](#)
- [오라클에서 Amazon Aurora PostgreSQL로 마이그레이션한 다음 데이터베이스 객체를 검증합니다.](#)

Oracle의 VARCHAR2(1) 데이터 유형을 Amazon Aurora PostgreSQL의 부울 데이터 유형으로 변환

작성자: Naresh Damera(AWS)

환경: PoC 또는 파일럿	소스: Oracle	대상: Amazon Aurora PostgreSQL
R 타입: 리아키텍트	워크로드: Oracle	기술: 마이그레이션, 소프트웨어 개발 및 테스트, 스토리지 및 백업, 데이터베이스
AWS 서비스: Amazon Aurora, AWS DMS, Amazon RDS, AWS SCT		

요약

Oracle용 Amazon Relational Database Service(Amazon RDS)에서 Amazon Aurora PostgreSQL-Compatible Edition으로 마이그레이션하는 동안 Amazon Web Services(AWS) Database Migration Service(AWS DMS)에서 마이그레이션을 검증할 때 데이터 불일치가 발생할 수 있습니다. 이러한 불일치를 방지하기 위해 VARCHAR2(1) 데이터 유형을 부울 데이터 유형으로 변환할 수 있습니다.

VARCHAR2 데이터 유형은 가변 길이 텍스트 문자열을 저장하며, VARCHAR2(1)는 문자열 길이가 1자 또는 1바이트임을 나타냅니다. VARCHAR2에 대한 자세한 내용은 [Oracle built-in data types](#)(Oracle 설명서)를 참조하세요.

이 패턴의 샘플 소스 데이터 테이블 열에서 VARCHAR2(1) 데이터는 Yes인 경우 Y이고, No인 경우 N입니다. 이 패턴에는 AWS DMS와 AWS Schema Conversion Tool(AWS SCT)을 사용하여 이 데이터 유형을 VARCHAR2(1)의 Y 및 N 값에서 부울의 참 또는 거짓 값으로 변환하기 위한 지침이 포함되어 있습니다.

대상

이 패턴은 AWS DMS를 사용하여 Oracle 데이터베이스를 Aurora PostgreSQL-Compatible로 마이그레이션한 경험이 있는 사용자에게 권장됩니다. 마이그레이션을 완료할 때 [Converting Oracle to Amazon RDS for PostgreSQL or Amazon Aurora PostgreSQL](#)(AWS SCT 설명서)의 권장 사항을 준수하세요.

사전 조건 및 제한 사항

사전 조건

- 활성 상태의 AWS 계정
- 보안 인증, 권한 및 보안 그룹 설정을 포함하여 환경이 Aurora를 사용할 준비가 되었는지 확인합니다. 자세한 내용은 [Amazon Aurora 환경 설정](#)(Aurora 설명서)을 참조하세요.
- VARCHAR2(1) 데이터가 있는 테이블 열을 포함하는 소스 Amazon RDS for Oracle 데이터베이스.
- 대상 Amazon Aurora PostgreSQL-Compatible 데이터베이스 인스턴스. 자세한 내용은 [데이터베이스 클러스터 생성 및 Aurora PostgreSQL 데이터베이스 클러스터의 데이터베이스에 연결](#)(Aurora 설명서)을 참조하세요.

제품 버전

- Amazon RDS for Oracle 버전 12.1.0.2 이상.
- AWS DMS 버전 3.1.4 이상. 자세한 내용은 [Oracle 데이터베이스를 AWS DMS의 소스로 사용 및 PostgreSQL 데이터베이스를 AWS DMS의 대상으로 사용](#)(AWS DMS 설명서)를 참조하세요. 가장 종합적인 버전 및 기능 지원을 위해 최신 버전의 AWS DMS를 사용하는 것을 권장합니다.
- AWS Schema Conversion Tool(AWS SCT) 버전 1.0.632 이상. 가장 종합적인 버전 및 기능 지원을 위해 최신 버전의 AWS DMS를 사용하는 것을 권장합니다.
- Aurora는 [PostgreSQL-Compatible 데이터베이스 엔진 버전](#)(Aurora 설명서)에 나와 있는 PostgreSQL 버전을 지원합니다.

아키텍처

소스 기술 스택

Amazon RDS for Oracle 데이터베이스 인스턴스

대상 기술 스택

Amazon Aurora PostgreSQL-Compatible 데이터베이스 인스턴스

소스 및 대상 아키텍처

도구

서비스

- [Amazon Aurora PostgreSQL-Compatible Edition](#)은 PostgreSQL 배포를 설정, 운영 및 확장할 수 있는 완전 관리형 ACID 준수 관계형 데이터베이스 엔진입니다.
- [AWS Database Migration Service\(AWS DMS\)](#)를 사용하면 데이터 스토어를 AWS 클라우드로 마이그레이션하거나 클라우드 조합과 온프레미스 설정 간에 마이그레이션할 수 있습니다.
- [Amazon Relational Database Service\(RDS\) for Oracle](#)을 사용하면 AWS 클라우드에서 관계형 데이터베이스를 설정, 운영 및 확장할 수 있습니다.
- [AWS Schema Conversion Tool\(AWS SCT\)](#)은 소스 데이터베이스 스키마와 대부분의 사용자 지정 코드를 대상 데이터베이스와 호환되는 형식으로 자동 변환하여 이기종 데이터베이스 마이그레이션을 지원합니다.

기타 서비스

- [Oracle SQL Developer](#)는 기존 배포와 클라우드 기반 배포 모두에서 Oracle 데이터베이스의 개발 및 관리를 간소화하는 통합 개발 환경입니다. 이 패턴에서는 이 도구를 사용하여 Amazon RDS for Oracle 데이터베이스 인스턴스에 연결하고 데이터를 쿼리합니다.
- [pgAdmin](#)은 PostgreSQL을 위한 오픈 소스 관리 도구입니다. 데이터베이스 객체를 생성, 유지 관리 및 사용하는 데 도움이 되는 그래픽 인터페이스를 제공합니다. 이 패턴에서는 이 도구를 사용하여 Aurora 데이터베이스 인스턴스에 연결하고 데이터를 쿼리합니다.

에픽

마이그레이션 준비

작업	설명	필요한 기술
데이터베이스 마이그레이션 보고서 생성합니다.	<ol style="list-style-type: none"> 1. AWS SCT에서 데이터베이스 마이그레이션 평가 보고서를 생성합니다. 자세한 내용은 마이그레이션 평가 보고서 생성을 참조하세요. 2. 마이그레이션 평가 보고서의 작업 항목을 검토하고 수 	DBA, 개발자

작업	설명	필요한 기술
	<p>행합니다. 자세한 내용은 평가 보고서 작업 항목을 참조하세요.</p>	
<p>대상 데이터베이스에서 외래 키 제약 조건을 비활성화합니다.</p>	<p>PostgreSQL에서는 트리거를 사용하여 외래 키를 구현합니다. 전체 로드 단계 동안 AWS DMS는 한 번에 하나씩 각 테이블을 로드합니다. 다음 방법 중 하나를 사용하여 전체 로드 동안 외래 키 제약 조건을 비활성화할 것을 강력히 권장합니다.</p> <ul style="list-style-type: none"> 인스턴스에서 모든 트리거를 임시로 비활성화한 후 전체 로드를 완료합니다. PostgreSQL에서 <code>session_replication_role</code> 파라미터를 사용합니다. <p>외래 키 제약 조건을 비활성화할 수 없는 경우 상위 테이블 및 하위 테이블과 관련된 기본 데이터에 대한 AWS DMS 마이그레이션 작업을 생성합니다.</p>	<p>DBA, 개발자</p>

작업	설명	필요한 기술
<p>대상 데이터베이스에서 프라이머리 키와 고유 키를 비활성화합니다.</p>	<p>다음 명령을 사용하여 대상 데이터베이스의 프라이머리 키와 제약 조건을 비활성화합니다. 이는 초기 로드 작업의 성능을 개선하는 데 도움이 됩니다.</p> <pre>ALTER TABLE <table> DISABLE PRIMARY KEY;</pre> <pre>ALTER TABLE <table> DISABLE CONSTRAINT <constraint_name>;</pre>	<p>DBA, 개발자</p>
<p>초기 로드 작업을 생성합니다.</p>	<p>AWS DMS에서 초기 로드를 위한 마이그레이션 작업을 생성합니다. 지침은 작업 생성을 참조하세요. 마이그레이션 방법으로 기존 데이터 마이그레이션을 선택합니다. 이 마이그레이션 방법은 API에서 Full Load라고 합니다. 아직 이 작업을 시작하지 마세요.</p>	<p>DBA, 개발자</p>

작업	설명	필요한 기술
<p>초기 로드 작업에 대한 작업 설정을 편집합니다.</p>	<p>작업 설정을 편집하여 데이터 검증을 추가합니다. 이러한 검증 설정은 JSON 파일로 생성해야 합니다. 지침 및 예제는 작업 설정 지정을 참조하세요. 다음 검증을 추가합니다.</p> <ul style="list-style-type: none"> 대상 데이터베이스에서 VARCHAR2(1) 데이터가 부울로 정확하게 변환되었는지 확인하려면 이 패턴의 추가 정보 섹션에 있는 데이터 검증 스크립트의 코드를 추가합니다. 검증 스크립트는 대상 테이블의 부울 값 1을 Y로, 0을 N으로 변환한 다음 대상 테이블의 값을 소스 테이블과 비교합니다. <p>나머지 데이터 마이그레이션을 검증하려면 작업에서 데이터 검증을 활성화합니다. 자세한 내용은 데이터 검증 작업 설정을 참조하세요.</p>	<p>AWS 관리자, DBA</p>
<p>지속적 복제 작업을 생성합니다.</p>	<p>AWS DMS에서 대상 데이터베이스를 소스 데이터베이스와 동기화 상태로 유지하는 마이그레이션 작업을 생성합니다. 지침은 작업 생성을 참조하세요. 마이그레이션 방법으로 데이터 변경 사항만 복제를 선택합니다. 아직 이 작업을 시작하지 마세요.</p>	<p>DBA</p>

마이그레이션 작업 테스트

작업	설명	필요한 기술
테스트용 샘플 데이터를 생성합니다.	소스 데이터베이스에서 테스트용으로 데이터가 포함된 샘플 테이블을 생성합니다.	개발자
충돌하는 활동이 없는지 확인합니다.	pg_stat_activity 을(를) 사용하여 마이그레이션에 영향을 미칠 수 있는 서버 활동을 확인합니다. 자세한 내용은 The Statistics Collector (PostgreSQL 설명서)를 참조하세요.	AWS 관리자
AWS DMS 마이그레이션 작업을 실행합니다.	AWS DMS 콘솔의 대시보드 페이지에서 이전 에픽에서 생성한 초기 로드 및 지속적 복제 작업을 시작합니다.	AWS 관리자
작업 및 테이블 로드 상태를 모니터링합니다.	마이그레이션하는 동안 작업 상태 와 테이블 상태 를 모니터링합니다. 초기 로드 작업이 완료되면 테이블 통계 탭에서: <ul style="list-style-type: none"> 로드 상태가 테이블 완료여야 합니다. 검증 상태가 검증됨이어야 합니다. 	AWS 관리자
마이그레이션 결과를 확인합니다.	pgAdmin을 사용하여 대상 데이터베이스의 테이블을 쿼리합니다. 쿼리에 성공하면 데이터가 성공적으로 마이그레이션된 것입니다.	개발자

작업	설명	필요한 기술
대상 데이터베이스에서 프라이머리 키와 외래 키를 추가합니다.	대상 데이터베이스에서 프라이머리 키와 외래 키를 생성합니다. 자세한 내용은 ALTER TABLE (PostgreSQL 웹 사이트)을 참조하세요.	DBA
테스트 데이터를 정리합니다.	소스 및 대상 데이터베이스에서 유닛 테스트용으로 생성한 데이터를 정리합니다.	개발자

전환

작업	설명	필요한 기술
마이그레이션을 완료합니다.	실제 소스 데이터를 사용하여 이전 에픽 마이그레이션 작업 테스트를 반복합니다. 이렇게 하면 데이터가 소스에서 대상 데이터베이스로 마이그레이션 됩니다.	개발자
소스 및 대상 데이터베이스가 동기화되어 있는지 검증합니다.	소스 및 대상 데이터베이스가 동기화되어 있는지 검증합니다. 자세한 내용 및 지침은 AWS DMS 데이터 검증 을 참조하세요.	개발자
소스 데이터베이스를 중지합니다.	Amazon RDS for Oracle 데이터베이스를 중지합니다. 자세한 내용은 Amazon RDS DB 인스턴스의 일시적 중지 를 참조하세요. 소스 데이터베이스를 중지하면 AWS DMS의 초기 로드 및 지속적 복제 작업이 자동으로 중지됩니다. 이러한 작업	개발자

작업	설명	필요한 기술
	을 중지하기 위한 추가 조치는 필요하지 않습니다.	

관련 리소스

AWS 참조

- [AWS DMS 및 AWS SCT를 사용하여 Aurora PostgreSQL로 Oracle 데이터베이스 마이그레이션\(AWS 권장 가이드\)](#)
- [Oracle을 Amazon RDS for PostgreSQL 또는 Amazon Aurora PostgreSQL로 변환\(AWS SCT 설명서\)](#)
- [AWS DMS 작동 방식\(AWS DMS 설명서\)](#)

기타 참조

- [Boolean data type](#)(PostgreSQL 설명서)
- [Oracle built-in data types](#)(Oracle 설명서)
- [pgAdmin](#)(pgAdmin 웹 사이트)
- [SQL Developer](#)(Oracle 웹 사이트)

자습서 및 동영상

- [DMS 시작하기](#)
- [Amazon RDS 시작하기](#)
- [AWS DMS 소개](#) (동영상)
- [Amazon RDS 이해](#)(동영상)

추가 정보

데이터 검증 스크립트

다음 데이터 검증 스크립트는 1을 Y로, 0을 N으로 변환합니다. 이를 통해 AWS DMS 작업이 테이블 검증을 성공적으로 완료하고 통과할 수 있습니다.

```
{
  "rule-type": "validation",
  "rule-id": "5",
  "rule-name": "5",
  "rule-target": "column",
  "object-locator": {
    "schema-name": "ADMIN",
    "table-name": "TEMP_CHRA_BOOL",
    "column-name": "GRADE"
  },
  "rule-action": "override-validation-function",
  "target-function": "case grade when '1' then 'Y' else 'N' end"
}
```

스크립트의 case 문이 검증을 수행합니다. 검증에 실패할 경우 AWS DMS는 대상 데이터베이스 인스턴스의 public.awsdms_validation_failures_v1 테이블에 레코드를 삽입합니다. 이 레코드에는 테이블 이름, 오류 시간, 소스 및 대상 테이블의 불일치 값에 대한 세부 정보가 포함됩니다.

이 데이터 검증 스크립트를 AWS DMS 작업에 추가하지 않고 데이터가 대상 테이블에 삽입되면 AWS DMS 작업은 검증 상태를 일치하지 않는 레코드로 표시합니다.

AWS SCT 변환 중에 AWS DMS 마이그레이션 작업은 VARCHAR2(1) 데이터 유형의 데이터 유형을 부울로 변경하고 "NO" 열에 프라이머리 키 제약 조건을 추가합니다.

Aurora PostgreSQL 호환에서 애플리케이션 사용자 및 역할을 생성

작성자: Abhishek Verma(AWS)

환경: PoC 또는 파일럿	소스: 모든 데이터베이스	대상: PostgreSQL 데이터베이스
R 유형: 리아키텍트	워크로드: 오픈 소스	기술: 마이그레이션, 데이터베이스
AWS 서비스: Amazon RDS, Amazon Aurora		

요약

Amazon Aurora PostgreSQL 호환 에디션으로 마이그레이션하는 경우, 소스 데이터베이스에 있는 데이터베이스 사용자 및 역할을 Aurora PostgreSQL 호환 데이터베이스에서 생성해야 합니다. 다음과 같이 두 가지 접근 방식을 사용하여 Aurora PostgreSQL 호환에서 사용자 및 역할을 생성할 수 있습니다.

- 대상에서도 소스 데이터베이스와 비슷한 사용자 및 역할을 사용하세요. 이 접근 방식에서는 소스 데이터베이스에서 사용자 및 역할에 대한 데이터 정의 언어(DDL)를 추출합니다. 그런 다음 변환하여 대상 Aurora PostgreSQL 호환 데이터베이스에 적용됩니다. 예를 들어, [SQL을 사용하여 Oracle에서 PostgreSQL로 사용자, 역할 및 권한 부여를 매핑](#) 블로그 게시물은 Oracle 소스 데이터베이스 엔진에서 추출하는 방법을 다룹니다.
- 개발, 관리 및 데이터베이스에서 기타 관련 작업을 수행할 때 일반적으로 사용되는 표준화된 사용자 및 역할을 사용하세요. 여기에는 각 사용자가 수행하는 읽기 전용, 읽기/쓰기, 개발, 관리 및 배포 작업이 포함됩니다.

이 패턴에는 표준화된 사용자 및 역할 접근 방식에 필요한 Aurora PostgreSQL 호환 사용자 및 역할 생성에 필요한 권한 부여가 포함되어 있습니다. 사용자 및 역할 생성 단계는 데이터베이스 사용자에게 최소 권한을 부여하는 보안 정책에 따라 조정됩니다. 다음 표에는 데이터베이스의 사용자, 해당 역할 및 세부 정보가 나열되어 있습니다.

사용자	역할	용도
-----	----	----

APP_read	APP_R0	스키마 APP에 대한 읽기 전용 액세스에 사용
APP_WRITE	APP_RW	스키마 APP의 쓰기 및 읽기 작업에 사용
APP_dev_user	APP_DEV	스키마 APP에 대한 읽기 전용 액세스 권한이 있으며, 스키마 APP_DEV의 개발 목적으로 사용
Admin_User	rds_superuser	데이터베이스에서 관리자 작업을 수행하는 데 사용
APP	APP_DEP	APP 스키마에서 개체를 만들고 APP 스키마에 개체를 배포하는 데 사용

사전 조건 및 제한 사항

사전 조건

- 활성 Amazon Web Services(AWS) 계정
- PostgreSQL 데이터베이스, Amazon Aurora PostgreSQL 호환 에디션 데이터베이스 또는 PostgreSQL 데이터베이스용 Amazon Relational Database Service(RDS)

제품 버전

- 모든 PostgreSQL 버전

아키텍처

소스 기술 스택

- 모든 데이터베이스

대상 기술 스택

• Amazon Aurora PostgreSQL 호환

대상 아키텍처

다음 다이어그램은 Aurora PostgreSQL 호환 데이터베이스의 사용자 역할과 스키마 아키텍처를 보여줍니다.

자동화 및 규모 조정

이 패턴에는 소스 또는 대상 데이터베이스의 기존 사용자에게 영향을 주지 않고 여러 번 실행할 수 있는 사용자, 역할 및 스키마 생성 스크립트가 포함됩니다.

도구

서비스

- [Amazon Aurora PostgreSQL 호환 버전](#)은 PostgreSQL 배포를 설정, 운영 및 확장할 수 있고 ACID를 준수하는 완전 관리형 관계형 데이터베이스 엔진입니다.

기타 서비스

- [psql](#)은 모든 PostgreSQL 데이터베이스 설치 시 함께 설치되는 터미널 기반 프론트 엔드 도구입니다. SQL, PL-PGSQL 및 운영 체제 명령을 실행하기 위한 명령줄 인터페이스를 갖추고 있습니다.
- [pgAdmin](#)은 PostgreSQL을 위한 오픈 소스 관리 도구입니다. 데이터베이스 객체를 생성, 유지 관리 및 사용하는 데 도움이 되는 그래픽 인터페이스를 제공합니다.

에픽

사용자 및 역할 생성

작업	설명	필요한 기술
배포 사용자를 생성합니다.	배포 사용자 APP은 배포 중에 데이터베이스 개체를 만들고 수정하는 데 사용됩니다. 스	DBA

작업	설명	필요한 기술
	<p>키마 APP에 배포 사용자 역할 APP_DEP을 생성하기 위해 다음 스크립트를 사용할 수 있습니다. 액세스 권한을 검증하여 이 사용자에게 필수 스키마 APP에서 객체를 생성할 수 있는 권한만 있는지 확인하세요.</p> <ol style="list-style-type: none"> 관리자 사용자에게 연결하고 스키마를 생성합니다. <pre data-bbox="634 695 1029 779">CREATE SCHEMA APP;</pre> 사용자를 생성합니다. <pre data-bbox="634 863 1029 1024">CREATE USER APP WITH PASSWORD <password > ;</pre> 역할을 생성합니다. <pre data-bbox="634 1108 1029 1507">CREATE ROLE APP_DEP ; GRANT all on schema APP to APP_DEP ; GRANT USAGE ON SCHEMA APP to APP_DEP ; GRANT connect on database <db_name> to APP_DEP ; GRANT APP_DEP to APP;</pre> 권한을 테스트하려면 APP에 연결하고 테이블을 생성합니다. <pre data-bbox="634 1696 1029 1824">set search_path to APP; SET</pre> 	

작업	설명	필요한 기술
	<pre>CREATE TABLE test(id integer); CREATE TABLE</pre> <p>5. 권한을 확인합니다.</p> <pre>select schemaname , tablename , tableowne r from pg_tables where tablename like 'test' ; schemaname tablename tableowner APP test APP</pre>	

작업	설명	필요한 기술
읽기 전용 사용자를 생성합니다.	<p>읽기 전용 사용자 APP_read는 스키마 APP에서 읽기 전용 작업을 수행하는 데 사용됩니다. 다음 스크립트를 사용하여 읽기 전용 사용자를 생성합니다. 이 사용자에게 스키마 APP의 개체를 읽을 수 있는 권한만 있는지 확인하고 스키마 APP에서 만든 새 개체에 대해 읽기 권한을 자동으로 부여할 수 있도록 액세스 권한을 검증하세요.</p> <ol style="list-style-type: none"> 1. 사용자 APP_read를 생성합니다. <pre data-bbox="630 905 1027 1104">create user APP_read ; alter user APP_read with password 'your_password' ;</pre> <ol style="list-style-type: none"> 2. 역할을 생성합니다. <pre data-bbox="630 1192 1027 1667">CREATE ROLE APP_ro ; GRANT SELECT ON ALL TABLES IN SCHEMA APP TO APP_RO ; GRANT USAGE ON SCHEMA APP TO APP_RO GRANT CONNECT ON DATABASE testdb TO APP_RO ; GRANT APP_RO TO APP_read;</pre> <ol style="list-style-type: none"> 3. 권한을 테스트하려면 APP_read 사용자를 사용하여 로그인하세요. 	DBA

작업	설명	필요한 기술
	<pre>set search_path to APP ; create table test1(id integer) ; ERROR: permission denied for schema APP LINE 1: create table test1(id integer) ; insert into test values (34) ; ERROR: permission denied for table test SQL state: 42501 select from test no rows selected</pre>	

작업	설명	필요한 기술
읽기/쓰기 사용자를 생성합니다.	<p>읽기/쓰기 사용자 APP_WRITE 는 스키마 APP에서 읽기 및 쓰기 작업을 수행하는 데 사용됩니다. 다음 스크립트를 사용하여 읽기/쓰기 사용자를 생성하고 APP_RW 역할을 부여합니다. 이 사용자에게 스키마 APP의 객체에 대한 읽기 및 쓰기 권한만 있는지 확인하고, 스키마 APP에서 만든 새 객체에 대해 읽기 및 쓰기 권한을 자동으로 부여할 수 있도록 액세스 권한을 검증하세요.</p> <ol style="list-style-type: none"> 1. 사용자를 생성합니다. <div data-bbox="630 951 1029 1192" style="border: 1px solid #ccc; border-radius: 10px; padding: 10px; margin: 10px 0;"> <pre>CREATE USER APP_WRITE ; alter user APP_WRITE with password 'your_password' ;</pre> </div> 2. 역할을 생성합니다. <div data-bbox="630 1278 1029 1808" style="border: 1px solid #ccc; border-radius: 10px; padding: 10px; margin: 10px 0;"> <pre>CREATE ROLE APP_RW; GRANT SELECT, INSERT, UPDATE, DELETE ON ALL TABLES IN SCHEMA APP TO APP_RW ; GRANT CONNECT ON DATABASE postgres to APP_RW ; GRANT USAGE ON SCHEMA APP to APP_RW ; ALTER DEFAULT PRIVILEGES IN SCHEMA APP</pre> </div> 	

작업	설명	필요한 기술
	<pre data-bbox="646 212 993 415">GRANT SELECT, INSERT, UPDATE, DELETE ON TABLES TO APP_RW ; GRANT APP_RW to APP_WRITE</pre> <p data-bbox="592 443 993 569">3. 권한을 테스트하려면 APP_WRITE 사용자를 사 용하여 로그인하세요.</p> <pre data-bbox="646 632 993 1220">SET SEARCH_PATH to APP; CREATE TABLE test1(id integer) ; ERROR: permission denied for schema APP LINE 1: create table test1(id integer) ; SELECT * FROM test ; id ---- 12 INSERT INTO test values (31) ; INSERT 0 1</pre>	

작업	설명	필요한 기술
관리자 사용자를 생성합니다.	<p>관리자 사용자 Admin_User 는 데이터베이스에서 관리자 작업을 수행하는 데 사용됩니다. 이러한 작업의 예로는 CREATE ROLE 및 CREATE DATABASE이 있습니다. Admin_User 는 내장된 역할 rds_superuser 를 사용하여 데이터베이스에서 관리자 작업을 수행합니다. 다음 스크립트를 사용하여 데이터베이스에서 관리자 사용자 Admin_User 의 권한을 생성하고 테스트할 수 있습니다.</p> <ol style="list-style-type: none"> 1. 사용자를 생성하고 역할을 부여합니다. <pre data-bbox="634 1052 1029 1367"> create user Admin_User WITH PASSWORD 'Your password' ALTER user Admin_user CREATEDB; ALTER user Admin_user CREATEROLE; </pre> <ol style="list-style-type: none"> 2. 권한을 테스트하려면 Admin_User 사용자로 로그인하세요. <pre data-bbox="634 1556 1029 1801"> SELECT * FROM APP.test ; id ---- 31 CREATE ROLE TEST ; </pre>	DBA

작업	설명	필요한 기술
	<pre>CREATE DATABASE test123 ;</pre>	

작업	설명	필요한 기술
개발 사용자를 생성합니다.	<p>개발 사용자 APP_dev_user 는 로컬 스키마 APP_DEV에 객체를 생성하고 스키마 APP에서 읽기 권한을 가질 수 있습니다. 다음 스크립트를 사용하여 데이터베이스에서 사용자 APP_dev_user 의 권한을 생성하고 테스트할 수 있습니다.</p> <ol style="list-style-type: none"> 1. 사용자를 생성합니다. <pre data-bbox="630 758 1029 919">CREATE USER APP1_dev_user with password 'your password';</pre> 2. App_dev_user 에 대한 APP_DEV 스키마를 생성합니다. <pre data-bbox="630 1104 1029 1224">CREATE SCHEMA APP1_DEV ;</pre> 3. APP_DEV 역할을 생성합니다. <pre data-bbox="630 1360 1029 1871">CREATE ROLE APP1_DEV ; GRANT APP1_R0 to APP1_DEV ; GRANT SELECT ON ALL TABLES IN SCHEMA APP1_DEV to APP1_dev_user GRANT USAGE, CREATE ON SCHEMA APP1_DEV to APP1_DEV_USER GRANT APP1_DEV to APP1_DEV_USER ;</pre> 	DBA

작업	설명	필요한 기술
	<p>4. 권한을 테스트하려면 APP_dev_user 으로 로그인하세요.</p> <pre>CREATE TABLE APP1_dev.test1(id integer); CREATE TABLE INSERT into APP1_dev.test1 (select * from APP1.test); INSERT 0 1 CREATE TABLE APP1.test4 (id int) ; ERROR: permission denied for schema APP1 LINE 1: create table APP1.test4 (id int) ;</pre>	

관련 리소스

PostgreSQL 설명서

- [역할 생성](#)
- [사용자 생성](#)
- [미리 정의된 역할](#)

추가 정보

PostgreSQL 14 개선 사항

PostgreSQL 14는 일반적으로 필요하고 권한이 있는 특정 기능 및 정보에 대한 액세스를 제공하는 사전 정의된 역할 세트를 제공합니다. 관리자(CREATE ROLE 권한이 있는 역할 포함)는 해당 환경에서 이

러한 역할 또는 기타 역할을 사용자에게 부여하여 지정된 기능과 정보에 대한 액세스 권한을 부여할 수 있습니다.

관리자는 GRANT 명령을 사용하여 사용자에게 이러한 역할에 대한 액세스 권한을 부여할 수 있습니다. 예를 들어 다음 명령을 실행하여 Admin_User에게 pg_signal_backend 역할을 부여할 수 있습니다.

```
GRANT pg_signal_backend TO Admin_User;
```

pg_signal_backend 역할은 관리자가 신뢰할 수 있는, 슈퍼유저가 아닌 역할을 활성화하여 다른 백엔드에 신호를 보낼 수 있도록 하기 위한 것입니다. 자세한 내용은 [PostgreSQL 14 개선 사항](#)을 참조하세요.

액세스 미세 조정

경우에 따라 사용자에게 더 세분화된 액세스를 제공해야 할 수 있습니다(예: 테이블 기반 액세스 또는 열 기반 액세스). 이러한 경우 추가 역할을 생성하여 사용자에게 해당 권한을 부여할 수 있습니다. 자세한 내용은 [PostgreSQL 권한 부여](#)를 참조하세요.

PostgreSQL-compatible Aurora 글로벌 데이터베이스를 사용하여 Oracle DR 에뮬레이션하기

작성자: HariKrishna 보가다 (AWS)

환경: PoC 또는 파일럿	소스: Oracle	대상: Aurora PostgreSQL
R 유형: 리아키텍트	워크로드: Oracle	기술: 마이그레이션, 현대화, 데이터베이스
AWS 서비스: Amazon Aurora		

요약

엔터프라이즈 재해 복구(DR)의 모범 사례는 기본적으로 최소한의 개입으로 그리고 이상적으로는 데이터 손실 없이 재해에서 살아남고(비즈니스 연속성), 정상 운영을 재개(비즈니스 재개) 할 수 있는 내결함성 하드웨어 및 소프트웨어 시스템을 설계하고 구현하는 것으로 구성됩니다. 엔터프라이즈 DR 목표를 충족하기 위해 내결함성 환경을 구축하는 것은 비용과 시간이 많이 소요될 수 있으며 기업의 강력한 노력이 필요합니다.

Oracle 데이터베이스는 다른 Oracle 데이터 보호 방식에 비해 최고 수준의 데이터 보호 및 가용성을 제공하는 DR 접근 방식 세 가지를 제공합니다.

- Oracle Zero Data Loss Recovery Appliance
- Oracle Active Data Guard
- 오라클 GoldenGate

이 패턴은 Amazon Aurora 글로벌 데이터베이스를 사용하여 Oracle GoldenGate DR을 에뮬레이션하는 방법을 제공합니다. 참조 아키텍처는 세 개의 AWS 지역에 걸쳐 재해 GoldenGate 복구에 Oracle을 사용합니다. 이 패턴은 Amazon Aurora PostgreSQL-Compatible 에디션으로 클라우드 네이티브 Aurora 글로벌 데이터베이스로 소스 아키텍처를 리플랫폼하는 과정을 안내합니다.

Aurora 전역 데이터베이스는 글로벌 설치 공간을 갖춘 애플리케이션 용으로 설계되었습니다. 단일 Aurora 데이터베이스는 최대 5개의 보조 리전과 함께 여러 AWS 리전에 걸쳐 있습니다. Aurora 글로벌 데이터베이스는 다음 기능을 제공합니다.

- 물리적 스토리지 수준 복제
- 짧은 대기 시간 글로벌 읽기
- 리전 전반의 정전 발생 시 신속한 재해 복구
- 빠른 리전 간 마이그레이션
- 리전 간 낮은 복제 지연
- 데이터베이스 little-to-no 성능이 미치는 영향

Aurora 글로벌 데이터베이스에 대한 자세한 내용은 [Amazon Aurora Global Database 사용](#)을 참조하세요. 계획되지 않은 장애 조치 및 관리형 장애 조치에 대한 자세한 내용은 [Amazon Aurora 글로벌 데이터베이스에서의 장애 조치 사용](#)을 참고하십시오.

사전 조건 및 제한 사항

사전 조건

- 활성 상태의 AWS 계정.
- 애플리케이션 연결을 위한 Java Database Connectivity(JDBC) PostgreSQL
- Amazon Aurora PostgreSQL-Compatible Edition 기반 Aurora 글로벌 데이터베이스
- Aurora PostgreSQL-Compatible 기반의 Aurora 글로벌 데이터베이스로 마이그레이션된 Oracle Real Application Clusters(RAC) 데이터베이스

Aurora 글로벌 데이터베이스에 적용되는 제한 사항

- Aurora 글로벌 데이터베이스를 모든 AWS 리전에서 사용할 수는 없습니다. 지원되는 리전의 목록을 보려면 [Aurora PostgreSQL이 포함된 Aurora 글로벌 데이터베이스](#)를 참고하십시오.
- 지원되지 않는 기능 및 Aurora 글로벌 데이터베이스의 기타 제한 사항에 대한 자세한 내용은 [Amazon Aurora 글로벌 데이터베이스의 제한 사항](#)을 참조하세요.

제품 버전

- Amazon Aurora PostgreSQL-Compatible Edition 버전 10.14 이상

아키텍처

소스 기술 스택

- Oracle RAC 4-노드 데이터베이스
- 오라클 GoldenGate

소스 아키텍처

다음 다이어그램은 Oracle을 사용하여 복제된 서로 다른 AWS 리전에 4노드 Oracle RAC가 있는 세 개의 클러스터를 보여줍니다. GoldenGate

대상 기술 스택

- Aurora PostgreSQL-Compatible을 기반으로 하는 3개 클러스터의 Amazon Aurora 글로벌 데이터베이스로 기본 리전에 클러스터 하나, 서로 다른 보조 리전에 클러스터 두 개가 있습니다.

대상 아키텍처

도구

서비스

- [Amazon Aurora PostgreSQL-Compatible Edition](#)은 PostgreSQL 배포를 설정, 운영 및 확장할 수 있는 완전 관리형 ACID 준수 관계형 데이터베이스 엔진입니다.
- [Amazon Aurora Global Database](#)는 여러 AWS 리전에 걸쳐 있으므로 지연 시간이 짧은 전역 읽기를 제공하며, 전체 AWS 리전에 영향을 미칠 수 있는 드물게 발생하는 중단을 신속하게 복구할 수 있습니다.

에픽

리더 DB 인스턴스와 함께 리전 추가

작업	설명	필요한 기술
하나 이상의 보조 Aurora 클러스터를 연결합니다.	AWS 관리 콘솔에서 Amazon Aurora를 선택합니다. 기본 클	DBA

작업	설명	필요한 기술
	러스터를 선택하고 작업을 선택한 다음, 드롭다운 목록에서 리전 추가를 선택합니다.	
인스턴스 클래스를 선택합니다.	보조 클러스터의 인스턴스 클래스를 변경할 수 있습니다. 하지만 기본 클러스터 인스턴스 클래스와 동일하게 유지하는 것을 권장합니다.	DBA
세 번째 리전을 추가합니다.	이 에픽의 단계를 반복하여 세 번째 리전에 클러스터를 추가합니다.	DBA

Aurora 글로벌 데이터베이스 장애 조치하기

작업	설명	필요한 기술
Aurora 글로벌 데이터베이스에서 기본 클러스터를 제거합니다.	<ol style="list-style-type: none"> 데이터베이스 페이지에서 기본 클러스터를 선택합니다. 보조 클러스터로 장애 조치를 시행하려면 글로벌에서 제거를 선택합니다. 	DBA
쓰기 트래픽이 새로 승격된 클러스터로 향하도록 애플리케이션을 다시 구성합니다.	애플리케이션의 엔드포인트를 새로 승격된 클러스터의 엔드포인트로 수정합니다.	DBA
사용할 수 없는 클러스터에 대한 쓰기 작업 실행을 중지합니다.	제거한 클러스터에 대해 애플리케이션 및 모든 데이터 조작 언어(DML) 활동을 중지합니다.	DBA
새 Aurora 글로벌 데이터베이스를 생성합니다.	새로 승격된 클러스터를 기본 클러스터로 사용하여 Aurora	DBA

작업	설명	필요한 기술
	글로벌 데이터베이스를 생성할 수 있습니다.	

프라이머리 클러스터를 시작합니다.

작업	설명	필요한 기술
글로벌 데이터베이스에서 시작할 기본 클러스터를 선택합니다.	Amazon Aurora 콘솔의 글로벌 데이터베이스 설정에서 기본 클러스터를 선택합니다.	DBA
클러스터를 시작합니다.	작업 드롭다운 목록에서 시작을 선택합니다. 이 프로세스는 다소 시간이 걸릴 수 있습니다. 화면을 새로 고쳐 상태를 보거나, 작업 완료 후 상태 열에서 클러스터의 현재 상태를 확인합니다.	DBA

리소스를 정리합니다.

작업	설명	필요한 기술
나머지 보조 클러스터를 삭제합니다.	장애 조치 파일럿을 완료한 후, 글로벌 데이터베이스에서 보조 클러스터를 제거합니다.	DBA
기본 클러스터를 삭제합니다.	클러스터를 제거합니다.	DBA

관련 리소스

- [Amazon Aurora Global Database 사용](#)
- [Amazon Aurora 글로벌 데이터베이스를 사용하는 Aurora PostgreSQL 재해 복구 솔루션 \(블로그 게시물\)](#)

Oracle SQL Developer 및 AWS SCT를 사용하여 Amazon RDS for Oracle에서 Amazon RDS for PostgreSQL로 점진적으로 마이그레이션

작성자: Pinesh Singal(AWS)

환경: PoC 또는 파일럿	소스: 데이터베이스: 관계형	대상: Amazon RDS PostgreSQL
R 유형: 리아키텍트	워크로드: Oracle, 오픈 소스	기술: 마이그레이션, 데이터베이스, 현대화
AWS 서비스: Amazon EC2, Amazon RDS		

요약

많은 마이그레이션 전략과 접근 방식은 몇 주에서 몇 달까지 지속될 수 있는 여러 단계로 진행됩니다. 이 기간에는 PostgreSQL DB 인스턴스로 마이그레이션하려는 소스 Oracle DB 인스턴스의 패치 또는 업그레이드로 인해 지연이 발생할 수 있습니다. 이러한 상황을 방지하려면 나머지 Oracle 데이터베이스 코드를 PostgreSQL 데이터베이스 코드로 점진적으로 마이그레이션하는 것이 좋습니다.

이 패턴은 초기 마이그레이션 후 수행된 트랜잭션 수가 많고 PostgreSQL 데이터베이스로 마이그레이션해야 하는 수 테라바이트의 Oracle DB 인스턴스에 대해 가동 중지 없이 증분 마이그레이션 전략을 제공합니다. 이 패턴의 step-by-step 접근 방식을 사용하면 Amazon Web Services (AWS) 관리 콘솔에 로그인하지 않고도 Oracle DB 인스턴스용 Amazon RDS (Amazon RDS) 를 PostgreSQL용 Amazon RDS로 점진적으로 마이그레이션할 수 있습니다.

이 패턴은 [Oracle SQL Developer](#)를 사용하여 원본 Oracle 데이터베이스에서 두 스키마 간의 차이점을 찾습니다. 그런 다음 AWS Schema Conversion Tool(AWS SCT)을 사용하여 Amazon RDS for Oracle 데이터베이스 스키마 객체를 PAmazon RDS for PostgreSQL 데이터베이스 스키마 객체로 변환합니다. 그런 다음 Windows 명령 프롬프트에서 Python 스크립트를 실행하여 소스 데이터베이스 객체의 증분 변경을 위한 AWS SCT 객체를 생성할 수 있습니다.

참고: 프로덕션 워크로드를 마이그레이션하기 전에 테스트 또는 비프로덕션 환경에서 이 패턴의 접근 방식에 대한 개념 증명(PoC)을 실행하는 것이 좋습니다.

사전 조건 및 제한 사항

사전 조건

- 활성 상태의 AWS 계정
- 기존 Amazon RDS for Oracle DB 인스턴스입니다.
- 기존 Amazon RDS for PostgreSQL DB 인스턴스입니다.
- AWS SCT는 Oracle 및 PostgreSQL 데이터베이스 엔진용 JDBC 드라이버와 함께 설치 및 구성되었습니다. 이에 대한 자세한 내용은 AWS SCT 설명서의 [AWS SCT 설치 및 필수 데이터베이스 드라이버 설치](#)를 참조하세요.
- 설치 및 구성된 Oracle SQL Developer. 이에 대한 자세한 내용은 [Oracle SQL 개발자](#) 설명서를 참조하세요.
- incremental-migration-sct-sql.zip 파일(첨부)이 로컬 컴퓨터에 다운로드됩니다.

제한 사항

- 소스 Amazon RDS for Oracle DB 인스턴스에 대한 최소 요구 사항은 다음과 같습니다.
 - Enterprise, Standard, Standard One 및 Standard Two 버전용 Oracle 버전 10.2 이상(버전 10.x의 경우), 11g(버전 11.2.0.3.v1 이상의 경우) 및 12.2 이하 및 18c.
- 대상 Amazon RDS for PostgreSQL DB 인스턴스의 최소 요구 사항은 다음과 같습니다.
 - PostgreSQL 버전 9.4 이상(버전 9.x의 경우), 10.x 및 11.x.
- 이 패턴은 Oracle SQL Developer를 사용합니다. 다른 도구를 사용하여 스키마 차이를 찾아 내보내는 경우 결과가 달라질 수 있습니다.
- Oracle SQL Developer에서 생성한 SQL [스크립트](#)는 변환 오류를 일으킬 수 있으며, 이 경우 수동 마이그레이션을 수행해야 합니다.
- AWS SCT 소스 및 대상 테스트 연결이 실패할 경우, 들어오는 트래픽을 수락하도록 Virtual Private Cloud(VPC) 보안 그룹의 JDBC 드라이버 버전과 인바운드 규칙을 구성해야 합니다.

제품 버전

- Amazon RDS for Oracle DB 인스턴스 버전 12.1.0.2(버전 10.2 이상)

- Amazon RDS for PostgreSQL DB 인스턴스 버전 11.5(버전 9.4 이상)
- Oracle SQL Developer의 버전 19.1 이상
- AWS SCT 버전 1.0.632 이상

아키텍처

소스 기술 스택

- Amazon RDS for Oracle DB 인스턴스

대상 기술 스택

- Amazon RDS for PostgreSQL DB 인스턴스

소스 및 대상 아키텍처

다음 다이어그램은 Amazon RDS for Oracle DB 인스턴스를 Amazon RDS for PostgreSQL DB 인스턴스로 마이그레이션하는 과정을 보여줍니다.

이 다이어그램은 다음 마이그레이션 워크플로를 보여 줍니다.

1. Oracle SQL 개발자를 열고 원본 및 대상 데이터베이스에 연결합니다.
2. [diff 보고서](#)를 생성한 다음 스키마 차이 객체에 대한 SQL 스크립트 파일을 생성합니다. 차이 보고서에 대한 자세한 내용은 Oracle 설명서의 [자세한 차이 보고서](#)를 참조하세요.
3. AWS SCT를 구성하고 Python 코드를 실행합니다.
4. SQL 스크립트 파일은 Oracle에서 PostgreSQL로 변환됩니다.
5. 대상 PostgreSQL DB 인스턴스에서 SQL 스크립트 파일을 실행합니다.

자동화 및 규모 조정

단일 프로그램의 여러 기능에 대한 추가 파라미터와 보안 관련 변경 사항을 Python 스크립트에 추가하여 이 마이그레이션을 자동화할 수 있습니다.

도구

- [AWS SCT](#) - AWS Schema Conversion Tool(AWS SCT)는 기존 데이터베이스 스키마를 한 데이터베이스 엔진에서 다른 데이터베이스 엔진으로 변환합니다.
- [Oracle SQL Developer](#) - Oracle SQL 개발자는 기존 배포와 클라우드 기반 배포 모두에서 Oracle 데이터베이스의 개발 및 관리를 간소화하는 통합 개발 환경(IDE)입니다.

code

incremental-migration-sct-sql.zip 파일(첨부됨)에는 이 패턴의 전체 소스 코드가 들어 있습니다.

에픽

소스 데이터베이스 스키마 차이에 대한 SQL 스크립트 파일 생성

작업	설명	필요한 기술
Oracle SQL Developer에서 데이터베이스 Diff를 실행합니다.	<ol style="list-style-type: none"> 1. 원본 Oracle DB 인스턴스에 로그인하고 도구를 선택한 다음 데이터 베이스 차이를 선택합니다. 2. 소스 연결에서 소스 데이터 베이스를 선택합니다. 3. 대상 연결에서 업데이트되거나 패치된 소스 데이터 베이스를 선택합니다. 4. 요구 사항에 따라 나머지 옵션을 구성하고 다음을 선택한 후 마침을 선택하여 diff 보고서를 생성합니다. 	DBA
SQL 스크립트 파일을 생성합니다.	SQL 파일에 차이를 생성하려면 스크립트 생성을 선택하세요.	DBA

작업	설명	필요한 기술
	그러면 AWS SCT가 데이터베이스를 Oracle에서 PostgreSQL로 변환하는 데 사용하는 SQL 스크립트 파일이 생성됩니다.	

Python 스크립트를 사용하여 AWS SCT에서 대상 DB 객체를 생성합니다.

작업	설명	필요한 기술
Windows 명령 프롬프트를 사용하여 AWS SCT를 구성합니다.	<ol style="list-style-type: none"> 1. 사전 설치된 AWS SCT 폴더에서 AWSSchemaConversionToolBatch.jar 파일을 복사하여 작업 디렉터리에 붙여넣습니다. 2. incremental-migration-sct-sql.zip 폴더(첨부)의 run_aws_sct_sql.py 파일에서 Python 코드를 배포합니다. 그러면 소스 및 대상 데이터베이스 환경 구성 세부 정보가 들어 있는.xml 파일 및.sct 파일이 projects 디렉터리에 생성됩니다. 또한 Oracle SQL Developer에서 생성한 SQL 스크립트 파일을 읽습니다. 마지막으로 output 디렉토리에.sql 파일 객체가 생성됩니다. 3. 다음 형식을 사용하여 database_migration.txt 파일에서 소스 및 대 	DBA

작업	설명	필요한 기술
	<p>상 환경 구성 세부 정보를 구성합니다.</p> <pre data-bbox="594 367 1029 1243"> #source_vendor,source_hostname,source_dbname,source_user,source_pwd,source_schema,source_port,source_sid,target_vendor,target_hostname,target_user,target_pwd,target_dbname,target_port ORACLE,myoracledb.cokmvis0v46q.us-east-1.rds.amazonaws.com,ORCL,orcl,orcl1234,orcl,1521,ORCL,POSTGRESQL,mypgdbinstance.cokmvis0v46q.us-east-1.rds.amazonaws.com,pguser,pgpassword,pgdb,5432 </pre> <p>4. 요구 사항에 따라 AWS SCT 구성 파라미터를 수정한 다음 SQL 스크립트 파일을 input 하위 디렉토리의 작업 디렉토리에 복사합니다.</p>	

작업	설명	필요한 기술
Python 스크립트를 실행합니다.	<ol style="list-style-type: none"> 1. <code>\$ python run_aws_sct_sql.py database_migration.txt</code> 명령을 사용하여 Python 스크립트를 실행합니다. 2. 그러면 DB 객체 SQL 파일이 생성됩니다. 변환 오류가 있는 변환되지 않은 코드는 수동으로 변환할 수 있습니다. 	DBA
Amazon RDS for PostgreSQL에서 객체 생성	Amazon RDS for PostgreSQL DB 인스턴스에서 SQL 파일을 실행하고 객체를 생성합니다.	DBA

관련 리소스

- [Amazon RDS의 Oracle](#)
- [Amazon RDS의 PostgreSQL](#)
- [AWS SCT 사용자 인터페이스 사용](#)
- [Oracle을 AWS SCT용 소스로 사용](#)

첨부

이 문서와 관련된 추가 콘텐츠에 액세스하려면 [attachment.zip](#) 파일의 압축을 풉니다.

Aurora PostgreSQL-Compatible에서 파일 인코딩을 사용하여 BLOB 파일을 TEXT에 로드

작성자: Bhanu Ganesh Gudivada와 Jeevan Shetty

환경: 프로덕션	소스: 온프레미스 Oracle 데이터베이스	대상: Aurora PostgreSQL-Compatible
R Type: 리아키텍트	워크로드: Oracle, 오픈 소스	기술: 마이그레이션, 데이터베이스

AWS 서비스: Amazon Aurora

요약

마이그레이션 중에 로컬 파일 시스템의 파일에서 로드되는 비정형 및 정형 데이터를 처리해야 하는 경우가 종종 있습니다. 데이터가 데이터베이스 문자 집합과 다른 문자 집합으로 되어 있을 수도 있습니다.

이러한 파일에는 다음과 같은 유형의 데이터가 들어 있습니다.

- 메타데이터 — 이 데이터는 파일 구조를 설명합니다.
- 반정형 데이터 — JSON 또는 XML과 같은 특정 형식의 텍스트 문자열입니다. 이러한 데이터에 대해 “항상 '<'로 시작” 또는 “줄 바꿈 문자를 포함하지 않음”과 같은 주장을 할 수 있습니다.
- 전체 텍스트 — 이 데이터에는 일반적으로 줄 바꿈 및 따옴표 문자를 비롯한 모든 유형의 문자가 포함됩니다. 또한 UTF-8 형식의 멀티바이트 문자로 구성될 수도 있습니다.
- 바이너리 데이터 — 이 데이터에는 null과 마커를 포함한 바이트 또는 바이트 조합이 포함될 수 있습니다. end-of-file

이러한 유형의 데이터를 혼합하여 로드하는 것은 어려울 수 있습니다.

이 패턴은 온프레미스 Oracle 데이터베이스, Amazon Web Services(AWS) Cloud에 있는 Amazon Elastic Compute Cloud(Amazon EC2) 인스턴스 상의 Oracle 데이터베이스 및 Amazon Relational Database Service (Amazon RDS) for Oracle 데이터베이스와 함께 사용될 수 있습니다. 예를 들어 이 패턴은 Amazon Aurora PostgreSQL-Compatible Edition을 사용합니다.

Oracle 데이터베이스에서는 BFILE (바이너리 파일) 포인터, DBMS_LOB 패키지, Oracle 시스템 함수를 사용하여 파일에서 로드하고 문자 인코딩을 사용해 CLOB로 변환할 수 있습니다. PostgreSQL은 Amazon Aurora PostgreSQL-Compatible Edition 데이터베이스로 마이그레이션할 때 BLOB 데이터 유형을 지원하지 않으므로 이러한 함수를 PostgreSQL 호환 스크립트로 변환해야 합니다.

이 패턴은 Amazon Aurora PostgreSQL-Compatible 데이터베이스의 단일 데이터베이스 열에 파일을 로드하는 두 가지 접근 방식을 제공합니다.

- 접근 방식 1 – 인코드 옵션과 함께 aws_s3 확장 프로그램의 table_import_from_s3 함수를 사용하여 Amazon Simple Storage Service(S3) 버킷에서 데이터를 가져옵니다.
- 접근 방식 2 – 데이터베이스를 벗어나 16진수로 인코딩한 다음 데이터베이스 내부에서 TEXT(을)를 볼 수 있도록 디코딩합니다.

Aurora PostgreSQL-Compatible은 aws_s3 확장 프로그램과 직접 통합되므로 접근 방식 1을 사용하는 것이 좋습니다.

이 패턴은 멀티바이트 문자와 고유한 형식을 가진 이메일 템플릿이 포함된 플랫폼 파일을 Amazon Aurora PostgreSQL-Compatible 데이터베이스로 로드하는 예제를 사용합니다.

사전 조건 및 제한 사항

사전 조건

- 활성 상태의 AWS 계정
- Amazon RDS 인스턴스 또는 Aurora PostgreSQL-Compatible 인스턴스
- SQL 및 관계형 데이터베이스 관리 시스템(RDBMS)에 대한 기본 이해
- Amazon Simple Storage Service(S3) 버킷
- Oracle 및 PostgreSQL의 시스템 함수에 대한 지식
- RPM 패키지 HexDump -XXD-0.1.1 (아마존 리눅스 2에 포함)

제한 사항

- TEXT 데이터 유형의 경우 저장할 수 있는 가장 긴 문자열은 약 1GB입니다.

제품 버전

- Aurora는 [Amazon Aurora PostgreSQL 업데이트](#)에 나열된 PostgreSQL 버전을 지원합니다.

아키텍처

대상 기술 스택

- Aurora PostgreSQL-Compatible

대상 아키텍처

접근 방식 1 –aws_s3.table_import_from_s3 사용

온프레미스 서버에서 멀티바이트 문자와 사용자 지정 형식이 있는 이메일 템플릿이 포함된 파일이 Amazon S3으로 전송됩니다. 이 패턴에서 제공하는 사용자 지정 데이터베이스 함수는 `file_encoding(와)`과 함께 `aws_s3.table_import_from_s3` 함수를 사용하여 파일을 데이터베이스에 로드하고 쿼리 결과를 TEXT 데이터 유형으로 반환합니다.

1. 파일이 스테이징 S3 버킷으로 전송됩니다.
2. 파일이 Amazon Aurora PostgreSQL-Compatible 데이터베이스에 업로드됩니다.
3. pgAdmin 클라이언트를 사용하여 사용자 지정 함수 `load_file_into_clob(을)`를 Aurora 데이터베이스에 배포합니다.
4. 사용자 지정 함수는 내부적으로 `file_encoding`과 함께 `table_import_from_s3(을)`를 사용합니다. 함수의 출력은 `array_to_string` 및 `array_agg(을)`를 TEXT 출력으로 사용하여 얻습니다.

접근 방식 2 – 데이터베이스를 벗어나 16진수로 인코딩한 다음 데이터베이스 내부에서 TEXT를 볼 수 있도록 디코딩합니다.

온프레미스 서버 또는 로컬 파일 시스템의 파일은 16진수 덤프로 변환됩니다. 그런 다음 파일을 PostgreSQL에 TEXT 필드로 가져옵니다.

1. `xxd -p` 옵션을 사용하여 명령줄에서 파일을 16진수 덤프로 변환합니다.
2. `\copy` 옵션을 사용하여 16진수 덤프 파일을 Aurora PostgreSQL-Compatible로 업로드한 다음 16진수 덤프 파일을 바이너리로 디코딩합니다.
3. 바이너리 데이터를 인코딩하여 TEXT(으)로 반환합니다.

도구

서비스

- [Amazon Aurora PostgreSQL 호환 에디션](#)은 PostgreSQL 배포를 설정, 운영 및 확장할 수 있는 완전 관리형 ACID 준수의 관계형 데이터베이스 엔진입니다.
- [Command Line Interface\(AWS CLI\)](#)는 명령줄 셸에서 명령을 사용하여 AWS 서비스와 상호 작용할 수 있는 오픈 소스 도구입니다.

기타 도구

- [pgAdmin4](#)는 PostgreSQL을 위한 오픈 소스 관리 및 개발 플랫폼입니다. pgAdmin4는 Linux, Unix, mac OS, Windows에서 PostgreSQL을 관리하는 데 사용할 수 있습니다.

에픽

접근 방식 1: Amazon S3에서 Aurora PostgreSQL-Compatible로 데이터 가져오기

작업	설명	필요한 기술
EC2 인스턴스를 시작합니다.	인스턴스 시작 지침은 인스턴스 시작 을 참조하십시오.	DBA
PostgreSQL 클라이언트 pGadmin 도구를 설치합니다.	pgAdmin 를 다운로드하고 설치합니다.	DBA
IAM 정책을 생성합니다.	파일이 저장되는 S3 버킷에 대한 액세스 권한을 부여하는 aurora-s3-access-policy (이)라 불리는 Identity and Access Management(IAM) 정책을 생성합니다. 다음 코드를 사용하여 S3 버킷의 이름으로 <bucket-name> (을)를 바꿉니다.	DBA

```
{
```

작업	설명	필요한 기술
	<pre> "Version": "2012-10-17", "Statement": [{ "Effect": "Allow", "Action": ["s3:GetObject", "s3:AbortMultipart Upload", "s3:DeleteObject", "s3:ListMultipartU ploadParts", "s3:PutObject", "s3:ListBucket"], "Resource": ["arn:aws:s3:::<buc ket-name>/*", "arn:aws:s3:::<buc ket-name>"] }] } </pre>	

작업	설명	필요한 기술
<p>Amazon S3에서 Aurora PostgreSQL-Compatible로 객체 가져오기를 위한 IAM 역할을 생성합니다.</p>	<p>다음 코드를 사용하여 신뢰 관계로 이름이 지정된 IAM 역할을 생성합니다. <code>aurora-s3-import-role</code> AssumeRole AssumeRole Aurora가 사용자를 대신하여 다른 AWS 서비스에 액세스할 수 있도록 허용합니다.</p> <pre data-bbox="597 632 1027 1268"> { "Version": "2012-10-17", "Statement": [{ "Effect": "Allow", "Principal": { "Service": "rds.amazonaws.com" }, "Action": "sts:AssumeRole" }] } </pre>	<p>DBA</p>

작업	설명	필요한 기술
IAM 역할을 클러스터와 연결합니다.	<p>IAM 역할을 Aurora PostgreSQL-Compatible 데이터베이스 클러스터와 연결하려면 다음 AWS CLI 명령을 실행합니다. Aurora PostgreSQL-Compatible 데이터베이스를 호스팅하는 AWS 계정의 ID로 <Account-ID> (을)를 변경합니다. 이렇게 하면 Aurora PostgreSQL-Compatible 데이터베이스에서 S3 버킷에 액세스할 수 있습니다.</p> <pre data-bbox="594 823 1026 1220">aws rds add-role-to-db-cluster --db-cluster-identifier aurora-postgres-cl --feature-name s3Import --role-arn arn:aws:iam::<account-id>:role/aurora-s3-import-role</account-id></pre>	DBA
예제를 Amazon S3으로 업로드합니다.	<ol style="list-style-type: none"> 이 패턴의 추가 정보 섹션에서 이메일 템플릿 코드를 salary.event.notification.email.vm (이라 불리는 파일에 복사합니다. 파일을 S3 버킷으로 업로드합니다. 	DBA, 앱 소유자

작업	설명	필요한 기술
사용자 지정 함수 배포.	<ol style="list-style-type: none">추가 정보 섹션에서 사용자 지정 함수 <code>load_file_into_clob</code> SQL 파일 콘텐츠를 임시 테이블에 복사합니다.Aurora PostgreSQL-Compatible 데이터베이스에 로그인하고, pgAdmin 클라이언트를 사용하여 데이터베이스 스키마에 배포합니다.	앱 소유자, DBA

작업	설명	필요한 기술
<p>데이터를 데이터베이스로 가져오기 위한 사용자 지정 함수를 실행합니다.</p>	<p>다음 SQL 명령을 실행하여 꺾쇠 괄호 안의 항목을 적절한 값으로 바꿉니다.</p> <pre data-bbox="597 394 1026 709">select load_file _into_clob('aws-s3 -import-test'::text, 'us-west-1'::text, 'employee.salary .event.notification.email.vm'::text);</pre> <p>명령을 실행하기 전에 다음 예제와 같이 꺾쇠 괄호 안의 항목을 적절한 값으로 바꿉니다.</p> <pre data-bbox="597 919 1026 1234">Select load_file _into_clob('aws-s3 -import-test'::text, 'us-west-1'::text, 'employee.salary .event.notification.email.vm'::text);</pre> <p>이 명령은 Amazon S3에서 파일을 로드하고 출력을 TEXT(을)로 반환합니다.</p>	<p>앱 소유자, DBA</p>

접근 방식 2: 로컬 Linux 시스템에서 템플릿 파일을 16진수 덤프로 변환

작업	설명	필요한 기술
<p>템플릿 파일을 16진수 덤프로 변환합니다.</p>	<p>Hexdump 유틸리티는 이진 파일의 내용을 16진수, 10진수, 8진수 또는 ASCII로 표시합니다. hexdump 명령은 util-</p>	<p>DBA</p>

작업	설명	필요한 기술
	<p>linux 패키지의 일부이며 Linux 배포판에 사전 설치되어 제공됩니다. Hexdump RPM 패키지도 Amazon Linux 2의 일부입니다.</p> <p>파일 내용을 16진 덤프로 변환하려면 다음 셸 명령을 실행합니다.</p> <pre data-bbox="597 646 1026 806">xxd -p </path/file.vm> tr -d '\n' > </path/file.hex></pre> <p>다음 예제와 같이 경로와 파일을 적절한 값으로 바꿉니다.</p> <pre data-bbox="597 961 1026 1243">xxd -p employee. salary.event.notification.email.vm tr -d '\n' > employee. salary.event.notification.email.vm.hex</pre>	

작업	설명	필요한 기술
<p>hexdump 파일을 데이터베이스 스키마에 로드합니다.</p>	<p>다음 명령을 사용하여 hexdump 파일을 Aurora PostgreSQL-Compatible 데이터베이스에 로드합니다.</p> <ol style="list-style-type: none"> Aurora PostgreSQL 데이터베이스에 로그인하고 <code>email_template_hex</code> (이)라 불리는 새 테이블을 생성합니다. <pre>CREATE TABLE email_template_hex(hex_data TEXT);</pre> 다음 명령을 사용하여 로컬 파일 시스템의 파일을 DB 스키마에 로드합니다. <pre>\copy email_template_hex FROM '/path/file.hex';</pre> <p>경로를 로컬 파일 시스템의 위치로 바꿉니다.</p> <pre>\copy email_template_hex FROM '/tmp/employee.salary.event.notification.email.vm.hex';</pre> <code>email_template_bytea</code> (이)라 불리는 테이블을 하나 더 생성합니다. 	<p>DBA</p>

작업	설명	필요한 기술
	<pre>CREATE TABLE email_template_bytea(hex_data bytea);</pre> <p>4. email_template_hex 의 데이터를 email_template_bytea 에 삽입합니다.</p> <pre>INSERT INTO email_template_bytea (hex_data) (SELECT decode(hex_data, 'hex') FROM email_template_hex limit 1);</pre> <p>5. 16진수 bytea 코드를 TEXT 데이터로 반환하려면 다음 명령을 실행합니다.</p> <pre>SELECT encode(hex_data::bytea, 'escape') FROM email_template_bytea;</pre>	

관련 리소스

참조

- [PostgreSQL 데이터베이스를 데이터베이스 마이그레이션 서비스용 대상으로 사용](#)
- [PostgreSQL Compatibility \(12.4\) Migration Playbook으로 Oracle Database 19c를 Amazon Aurora로](#)
- [IAM 정책 생성](#)
- [IAM 역할을 Amazon Aurora MySQL DB 클러스터와 연결](#)

- [pgAdmin](#)

자습서

- [Amazon RDS 시작](#)
- [Oracle에서 Amazon Aurora로 마이그레이션](#)

추가 정보

load_file_into_clob 사용자 지정 함수

```
CREATE OR REPLACE FUNCTION load_file_into_clob(
    s3_bucket_name text,
    s3_bucket_region text,
    file_name text,
    file_delimiter character DEFAULT '& '::bpchar,
    file_encoding text DEFAULT 'UTF8'::text)
    RETURNS text
    LANGUAGE 'plpgsql'
    COST 100
    VOLATILE PARALLEL UNSAFE
AS $BODY$
DECLARE
    blob_data BYTEA;
    clob_data TEXT;
    l_table_name CHARACTER VARYING(50) := 'file_upload_hex';
    l_column_name CHARACTER VARYING(50) := 'template';
    l_return_text TEXT;
    l_option_text CHARACTER VARYING(150);
    l_sql_stmt CHARACTER VARYING(500);

BEGIN

    EXECUTE format ('CREATE TEMPORARY TABLE %I (%I text, id_serial serial)',
l_table_name, l_column_name);

    l_sql_stmt := 'select ''(format text, delimiter '''' || file_delimiter || '''' ,
encoding '''' || file_encoding || '''' )'' ';

    EXECUTE FORMAT(l_sql_stmt)
    INTO l_option_text;
```

```

EXECUTE FORMAT('SELECT aws_s3.table_import_from_s3($1,$2,$6,
aws_commons.create_s3_uri($3,$4,$5))')
  INTO l_return_text
  USING l_table_name, l_column_name, s3_bucket_name,
file_name,s3_bucket_region,l_option_text;

EXECUTE format('select array_to_string(array_agg(%I order by id_serial),E''\n'')
from %I', l_column_name, l_table_name)
  INTO clob_data;

drop table file_upload_hex;

RETURN clob_data;
END;
$BODY$;

```

이메일 템플릿

```

#####
##
##
##   johndoe Template Type: email
##
##   File: johndoe.salary.event.notification.email.vm
##
##   Author: Aimée Étienne   Date 1/10/2021
##
## Purpose: Email template used by EmplmanagerEJB to inform a johndoe they   ##
##         have been given access to a salary event
##
##   Template Attributes:
##
##         invitedUser - PersonDetails object for the invited user
##
##         salaryEvent - OfferDetails object for the event the user was given access
##
##         buyercollege - CompDetails object for the college owning the salary event
##
##         salaryCoordinator - PersonDetails of the salary coordinator for the event
##

```

```

##      idp - Identity Provider of the email recipient
##
##      httpWebRoot - HTTP address of the server
##
##
#####

${invitedUser.firstname} ${invitedUser.lastname},

Ce courriel confirme que vous avez ete invite par ${salaryCoordinator.firstname} ${
salaryCoordinator.lastname} de $buyercollege.collegeName a participer a l'evenement
"$salaryEvent.offeringtitle" sur johndoeMaster Sourcing Intelligence.

Votre nom d'utilisateur est ${invitedUser.username}

Veuillez suivre le lien ci-dessous pour acceder a l'evenement.

${httpWebRoot}/myDashboard.do?idp=${idp}

Si vous avez oublie votre mot de passe, utilisez le lien "Mot de passe oublie" situe
sur l'ecran de connexion et entrez votre nom d'utilisateur ci-dessus.

Si vous avez des questions ou des preoccupations, nous vous invitons a
communiquer avec le coordonnateur de l'evenement ${salaryCoordinator.firstname} ${
salaryCoordinator.lastname} au ${salaryCoordinator.workphone}.

*****

johndoeMaster Sourcing Intelligence est une plateforme de soumission en ligne pour les
equipements, les materiaux et les services.

Si vous avez des difficultes ou des questions, envoyez un courriel a
support@johndoeMaster.com pour obtenir de l'aide.

```

AWS DMS를 사용하여 SSL 모드에서 Amazon RDS for Oracle를 Amazon RDS for PostgreSQL로 마이그레이션

작성자: Pinesh Singal(AWS)

환경: PoC 또는 파일럿	소스: Amazon RDS for Oracle	대상: Amazon RDS PostgreSQL
R 유형: 리아키텍트	워크로드: Oracle, 오픈 소스	기술: 마이그레이션, 보안, 자격 증명, 규정 준수, 데이터베이스
AWS 서비스: AWS DMS, Amazon RDS		

요약

이 패턴은 Amazon Web Services(AWS) 클라우드에서 Oracle 데이터베이스의 Amazon Relational Database Service(Amazon RDS) 인스턴스를 Amazon RDS for PostgreSQL 데이터베이스로 마이그레이션하는 방법을 설명합니다. 데이터베이스 간 연결을 암호화하기 위해 패턴은 Amazon RDS와 AWS Database Migration Service(AWS DMS)의 인증 기관(CA) 및 SSL 모드를 사용합니다.

이 패턴은 트랜잭션 수가 많은 멀티 테라바이트급 Oracle 소스 데이터베이스의 다운타임이 거의 또는 전혀 없는 온라인 마이그레이션 전략을 설명합니다. 데이터 보안에 대한 패턴은 데이터 전송 시에 SSL을 사용합니다.

이 패턴은 AWS Schema Conversion Tool(AWS SCT)을 사용하여 Amazon RDS for Oracle 데이터베이스 스키마를 Amazon RDS for PostgreSQL 스키마로 변환합니다. 그런 다음 패턴은 AWS DMS를 사용하여 Amazon RDS for Oracle 데이터베이스에서 Amazon RDS for PostgreSQL 데이터베이스로 데이터를 마이그레이션합니다.

사전 조건 및 제한 사항

사전 조건

- 활성 상태의 AWS 계정
- rds-ca-2019로만 구성된 Amazon RDS 데이터베이스 인증 기관(CA) (rds-ca-2015 인증서는 2020년 3월 5일에 만료됨)

- AWS SCT
- DMS
- pgAdmin
- SQL 도구(예: SQL Developer 또는 SQL*Plus)

제한 사항

- Amazon RDS for Oracle 데이터베이스 - Enterprise 및 Standard Two 에디션의 경우 요구되는 최소 Oracle 버전은 19c입니다.
- Amazon RDS for PostgreSQL 데이터베이스 — PostgreSQL의 경우 요구되는 최소 버전은 12 이상입니다(버전 9.x 이상).

제품 버전

- Amazon RDS for Oracle 데이터베이스 버전 12.1.0.2 인스턴스
- Amazon RDS for PostgreSQL 데이터베이스 버전 11.5 인스턴스

아키텍처

소스 기술 스택

- 버전 12.1.0.2.v18의 Amazon RDS for Oracle 데이터베이스 인스턴스입니다.

대상 기술 스택

- DMS
- 버전 11.5의 Amazon RDS for PostgreSQL 데이터베이스 인스턴스

대상 아키텍처

다음 다이어그램은 Oracle(소스) 데이터베이스와 PostgreSQL(대상) 데이터베이스 간의 데이터 마이그레이션 아키텍처를 보여줍니다. 아키텍처에는 다음이 포함되어 있습니다.

- Virtual Private Cloud(VPC)
- 가용 영역
- 프라이빗 서브넷

- Amazon RDS for Oracle 데이터베이스
- AWS DMS 복제 인스턴스
- RDS for PostgreSQL 데이터베이스

소스 및 대상 데이터베이스의 연결을 암호화하려면 Amazon RDS 및 AWS DMS에서 CA 및 SSL 모드를 활성화해야 합니다.

도구

서비스

- [AWS Database Migration Service\(AWS DMS\)](#)를 사용하면 데이터 스토어를 AWS 클라우드로 마이그레이션하거나 클라우드와 온프레미스 설정 조합 간에 마이그레이션할 수 있습니다.
- [Oracle용 Amazon Relational Database Service\(Amazon RDS\)](#)는 AWS 클라우드에서 Oracle 관계형 데이터베이스를 설정하고, 운영하고, 규모를 조정하도록 도와줍니다.
- [PostgreSQL용 Amazon Relational Database Service\(Amazon RDS\)](#)는 AWS 클라우드에서 PostgreSQL 관계형 데이터베이스를 설정, 운영 및 조정하는 데 도움이 됩니다.
- [AWS Schema Conversion Tool\(AWS SCT\)](#)은 소스 데이터베이스 스키마와 대부분의 사용자 지정 코드를 대상 데이터베이스와 호환되는 형식으로 자동 변환하여 이기종 데이터베이스 마이그레이션을 지원합니다.

기타 서비스

- [pgAdmin](#)은 PostgreSQL을 위한 오픈 소스 관리 도구입니다. 데이터베이스 객체를 생성, 유지 관리 및 사용하는 데 도움이 되는 그래픽 인터페이스를 제공합니다.

에픽

Amazon RDS for Oracle 인스턴스 구성

작업	설명	필요한 기술
Oracle 데이터베이스 인스턴스를 생성합니다.	AWS 계정에 로그인하고, AWS Management Console을 열고 Amazon RDS 콘솔로 이동합니	일반 AWS, DBA

작업	설명	필요한 기술
	다. 콘솔에서 데이터베이스 생성을 선택한 다음 Oracle을 선택합니다.	
보안 그룹을 구성합니다.	인바운드 및 아웃바운드 보안 그룹을 구성합니다.	일반 AWS
옵션 그룹을 생성합니다.	Amazon RDS for Oracle 데이터베이스와 동일한 VPC 및 보안 그룹에 옵션 그룹을 생성합니다. 옵션에서 SSL을 선택합니다. 포트의 경우 2484(SSL 연결용)를 선택합니다.	일반 AWS
옵션 설정을 구성합니다.	<p>다음 설정을 사용합니다.</p> <ul style="list-style-type: none"> • SQLNET.CIPHER_SUITE : SSL_RSA_WITH_AES_256_CBC_SHA • SQLNET.SSL_VERSION : 1.2 or 1.0 	일반 AWS
RDS for Oracle DB 인스턴스를 수정합니다.	CA 인증서를 rds-ca-2019로 설정합니다. 옵션 그룹에서 이전에 만든 옵션 그룹을 연결합니다.	DBA, 일반 AWS

작업	설명	필요한 기술
<p>RDS for Oracle DB 인스턴스를 사용할 수 있는지 확인합니다.</p>	<p>Amazon RDS for Oracle용 데이터베이스 인스턴스가 가동 및 실행 중이고 데이터베이스 스키마에 액세스할 수 있는지 확인하세요.</p> <p>RDS for Oracle DB에 연결하려면 명령줄에서 sqlplus 명령을 사용합니다.</p> <pre data-bbox="597 667 1026 1738"> \$ sqlplus orcl/**** @myoracledb.cokmvi s0v46q.us-east-1.r ds.amazonaws.com:1 521/ORCL SQL*Plus: Release 12.1.0.2.0 Production on Tue Oct 15 18:11:07 2019 Copyright (c) 1982, 2016, Oracle. All rights reserved. Last Successful login time: Mon Dec 16 2019 23:17:31 +05:30 Connected to: Oracle Database 12c Enterprise Edition Release 12.1.0.2.0 - 64bit Production With the Partition ing, OLAP, Advanced Analytics and Real Application Testing options SQL> </pre>	<p>DBA</p>

작업	설명	필요한 기술
RDS for Oracle 데이터베이스에서 객체와 데이터를 생성합니다.	객체를 생성하고 스키마에 데이터를 삽입합니다.	DBA

Amazon RDS for PostgreSQL 인스턴스 구성

작업	설명	필요한 기술
RDS for PostgreSQL 데이터베이스를 생성합니다.	Amazon RDS 콘솔 데이터베이스 생성 페이지에서 PostgreSQL을 선택하여 Amazon RDS for PostgreSQL 데이터베이스 인스턴스를 생성합니다.	DBA, 일반 AWS
보안 그룹을 구성합니다.	인바운드 및 아웃바운드 보안 그룹을 구성합니다.	일반 AWS
파라미터 그룹을 생성합니다.	PostgreSQL 버전 11.x를 사용하는 경우, 파라미터 그룹을 생성하여 SSL 파라미터를 설정합니다. PostgreSQL 버전 12에서는 SSL 파라미터 그룹이 기본적으로 활성화되어 있습니다.	일반 AWS
파라미터를 편집합니다.	<code>rds.force_ssl</code> 파라미터를 1 (on)로 변경합니다. 기본적으로 <code>ssl</code> 파라미터는 1 (on)입니다. <code>rds.force_ssl</code> 파라미터를 1(으)로 설정하면 모든 연결이 SSL 모드를 통해서만 연결되도록 강제할 수 있습니다.	일반 AWS

작업	설명	필요한 기술
RDS for PostgreSQL DB 인스턴스를 수정합니다.	CA 인증서를 rds-ca-2019로 설정합니다. PostgreSQL 버전에 따라 기본 파라미터 그룹 또는 이전에 생성한 파라미터 그룹을 연결합니다.	DBA, 일반 AWS

작업	설명	필요한 기술
<p>RDS for PostgreSQL DB 인스턴스를 사용할 수 있는지 확인합니다.</p>	<p>Amazon RDS for PostgreSQL 데이터베이스가 작동 및 실행 중인지 확인합니다.</p> <p>이 <code>psql</code> 명령은 명령줄에서 설정한 <code>sslmode</code>와(과) SSL 연결을 설정합니다.</p> <p>한 가지 옵션은 파라미터 그룹에서 <code>sslmode=1</code> 을(를) 설정하고 명령에 <code>sslmode</code> 파라미터를 포함하지 않고 <code>psql</code> 연결을 사용하는 것입니다.</p> <p>다음 출력은 SSL 연결이 설정되었음을 보여줍니다.</p> <pre data-bbox="597 968 1024 1724"> \$ psql -h mypgdbins tance.cokmvis0v46q .us-east-1.rds.ama zonaws.com -p 5432 "dbname=pgdb user=pguser" Password for user pguser: psql (11.3, server 11.5) SSL connection (protocol: TLSv1.2, cipher: ECDHE-RSA- AES256-GCM-SHA384, bits: 256, compressi on: off) Type "help" for help. pgdb=> </pre> <p>두 번째 옵션은 파라미터 그룹에서 <code>sslmode=1</code> 을(를) 설정</p>	<p>DBA</p>

작업	설명	필요한 기술
	<p>하고 <code>psql</code> 명령에 <code>sslmode</code> 파라미터를 포함하는 것입니다.</p> <p>다음 출력은 SSL 연결이 설정되었음을 보여줍니다.</p> <pre data-bbox="597 457 1024 1171"> \$ psql -h mypgdbins tance.cokmvis0v46q .us-east-1.rds.ama zonaws.com -p 5432 "dbname=pgdb user=pgus er sslmode=require" Password for user pguser: psql (11.3, server 11.5) SSL connection (protocol: TLSv1.2, cipher: ECDHE-RSA- AES256-GCM-SHA384, bits: 256, compressi on: off) Type "help" for help. pgdb=> </pre>	

AWS SCT 구성 및 실행

작업	설명	필요한 기술
AWS SCT를 설치합니다.	최신 버전의 AWS SCT 애플리케이션을 설치합니다.	일반 AWS
JDBC 드라이버를 사용하여 AWS SCT를 구성합니다.	Oracle(ojdbc8.jar) 및 PostgreSQL(postgresq-l-42.2.5.jar)의 Java Database Connectivity(JDBC) 드라이버를 다운로드합니다.	일반 AWS

작업	설명	필요한 기술
	AWS SCT에서 드라이버를 구성하려면 설정, 글로벌 설정, 드라이버를 선택합니다.	

작업	설명	필요한 기술
<p>AWS SCT 프로젝트를 생성합니다.</p>	<p>Oracle을 소스 DB 엔진으로 사용하고 Amazon RDS for PostgreSQL를 대상 DB 엔진으로 사용하여 AWS SCT 프로젝트 및 보고서를 생성합니다.</p> <p>1. 연결 세부 정보를 제공하여 소스 Oracle 데이터베이스 및 대상 Amazon RDS for PostgreSQL 데이터베이스에 대한 연결을 테스트합니다.</p> <p>소스 Oracle 데이터베이스의 경우 다음과 같은 승인 또는 권한이 필요합니다.</p> <ul style="list-style-type: none"> • CONNECT • SELECT_CATALOG_ROLE • SELECT ANY DICTIONARY • SELECT on SYS.USER\$ TO <sct_user> <p>자세한 내용은 Oracle 데이터베이스를 AWS SCT의 소스로 사용을 참조하세요.</p> <p>소스 및 대상 연결이 모두 성공해야 AWS SCT에서 마이그레이션 보고를 시작할 수 있습니다.</p>	<p>일반 AWS</p>

작업	설명	필요한 기술
	2. 보고서를 작성한 후 변환할 스키마를 입력하고 완료를 선택합니다.	
데이터베이스 객체를 검증합니다.	1. 스키마 로드를 선택합니다. AWS SCT는 오류가 있는 객체를 포함하여 소스 및 변환된 대상 객체를 표시합니다. 대상 데이터베이스에서 잘못된 객체를 업데이트합니다. 2. 오류를 검토하고 수동 개입을 통해 오류를 해결합니다. 3. 오류가 모두 지워지면 스키마 로드를 다시 선택합니다. 4. 데이터베이스에 적용을 선택합니다. 5. PgAdmin 또는 PostgreSQL DB 연결을 지원하는 다른 도구에 연결하고 스키마와 객체를 확인합니다.	DBA, 일반 AWS

AWS DMS 구성 및 실행

작업	설명	필요한 기술
복제 인스턴스를 생성합니다.	1. 계정에 로그인하고 AWS Management Console을 열고 AWS DMS 콘솔로 이동합니다. 2. VPC, 보안 그룹, 가용 영역 및 추가 연결 속성에 대한 유	일반 AWS

작업	설명	필요한 기술
	호한 설정을 사용하여 복제 인스턴스를 생성합니다.	
인증서를 가져옵니다.	<ol style="list-style-type: none">1. rds-ca-2019-root.pem 인증서를 다운로드합니다.2. 인증서 페이지에서 rds-ca-2019-root 으(로) 인증서를 가져옵니다.	일반 AWS

작업	설명	필요한 기술
<p>소스 엔드포인트를 생성합니다.</p>	<ol style="list-style-type: none"> 1. RDS DB 인스턴스 선택을 선택하여 Amazon RDS for Oracle의 소스 엔드포인트를 생성한 다음 생성한 RDS for Oracle DB 인스턴스를 선택합니다. 엔드포인트 구성 세부 정보가 자동으로 채워집니다. 2. 수동으로 액세스 정보 제공을 선택합니다. 포트에는 2484을 입력합니다. 3. Secure Socket Layer(SSL) 모드에서 verify-ca를, 선택한 다음 이전에 생성한 CA 인증서를 선택합니다. 4. 엔드포인트 설정에서 크기가 없는 NUMBER 데이터 유형을 지원하기 위해 추가 연결 속성 NumberDataScale=-2 을(를) 추가합니다. <p>자세한 내용은 Oracle 데이터베이스를 AWS Database Migration Service의 소스로 사용을 참조하세요.</p>	<p>일반 AWS</p>

작업	설명	필요한 기술
대상 엔드포인트를 생성합니다.	<ol style="list-style-type: none"> 1. RDS DB 인스턴스 선택을 선택하여 Amazon RDS for PostgreSQL의 대상 엔드포인트를 생성한 다음 RDS for PostgreSQL DB 인스턴스를 선택합니다. 엔드포인트 구성 세부 정보가 자동으로 채워집니다. 2. 수동으로 액세스 정보 제공을 선택합니다. 포트에는 2484을 입력합니다. <p>자세한 내용은 PostgreSQL 데이터베이스를 AWS Database Migration Service 대상으로 사용을 참조하세요.</p>	일반 AWS
엔드포인트를 테스트합니다.	<ol style="list-style-type: none"> 1. 소스 및 대상 엔드포인트를 테스트하여 둘 다 성공적이고 사용 가능한지 확인합니다. 2. 테스트가 실패할 경우 보안 그룹 인바운드 규칙이 유효한지 확인합니다. 	일반 AWS

작업	설명	필요한 기술
<p>마이그레이션 작업을 생성합니다.</p>	<p>전체 로드 및 변경 데이터 캡처 (CDC) 또는 데이터 검증을 위한 마이그레이션 작업을 생성하려면 다음을 수행합니다.</p> <ol style="list-style-type: none"> 1. 데이터베이스 마이그레이션 작업을 생성하려면 복제 인스턴스, 소스 데이터베이스 엔드포인트, 대상 데이터베이스 엔드포인트를 선택합니다. 마이그레이션 유형을 다음 중 하나로 지정합니다. <ul style="list-style-type: none"> • 기존 데이터 마이그레이션 (전체 로드) • 데이터 변경 내용 복제 전용(CDC) • 기존 데이터 마이그레이션 및 진행 중인 변경 사항 복제 (전체 로드 및 CDC) 2. 테이블 매핑에서 선택 규칙 및 변환 규칙을 GUI 또는 JSON 형식으로 구성할 수 있습니다. <ul style="list-style-type: none"> • 선택 규칙에서 스키마를 선택하고 테이블 이름을 입력한 다음 구성할 작업 (포함 또는 제외)을 선택합니다. 예를 들어, 스키마 ORCL, 테이블 이름 %, Action Include. • 변환 규칙에서 다음 작업 중 하나를 수행합니다. 	<p>일반 AWS</p>

작업	설명	필요한 기술
	<ul style="list-style-type: none"> 스키마를 선택하고 작업(대소문자, 접두사, 접미사)을 선택합니다. 예를 들어, 대상 스키마 ORCL, Action Make lowercase. 스키마를 선택하고 테이블 이름을 입력한 다음 작업(대소문자, 접두사, 접미사)을 선택합니다. 예를 들어, 대상 스키마 ORCL, 테이블 %, Action Make lowercase <p>3. Amazon CloudWatch 로그 모니터링을 활성화합니다.</p> <p>4. 매핑 규칙의 경우 다음 JSON 코드를 추가합니다.</p> <pre data-bbox="630 1121 1029 1860"> { "rules": [{ "rule-type": "transformation", "rule-id": "1", "rule-name": "1", "rule-target": "table", "object-locator": { "schema-name": "%", "table-name": "%" } }] } </pre>	

작업	설명	필요한 기술
	<pre> }, "rule- action": "convert- lowercase", "value": null, "old-valu e": null }, { "rule- type": "transfor mation", "rule-id" : "2", "rule-nam e": "2", "rule-tar get": "schema", "object-l ocator": { "schema-name": "ORCL", "table-name": "%" }, "rule- action": "convert- lowercase", "value": null, "old-valu e": null }, { "rule-typ e": "selection", "rule-id" : "3", "rule-nam e": "3", </pre>	

작업	설명	필요한 기술
<p>프로덕션 실행을 계획합니다.</p>	<pre> "object-1 ocator": { "schema-name": "ORCL", "table-name": "DEPT" }, "rule-act ion": "include", "filters" : [] }] } </pre>	<p>마이그레이션 책임자</p>

작업	설명	필요한 기술
<p>마이그레이션 작업을 실행합니다.</p>	<ol style="list-style-type: none"> Ready 상태인 AWS DMS 작업을 시작하고 CloudWatch Amazon의 마이그레이션 작업 로그에서 오류가 있는지 모니터링하십시오. <p>마이그레이션 유형으로 기존 데이터 마이그레이션 및 진행 중인 변경 사항 복제를 선택하고 상태가 완료 진행 중인 복제 로드인 경우 CDC 데이터 마이그레이션을 통한 전체 로드가 완료되고 검증이 계속 진행됩니다.</p> <ol style="list-style-type: none"> 마이그레이션을 시작한 후에는 에서 추가 SSL 연결 정보를 얻을 수 있습니다. CloudWatch Oracle의 경우 다음 연결 문자열을 CloudWatch 표시합니다. <pre>2019-12-17T09:15:11 [SOURCE_UNLOAD]I: Connecting to Oracle: Beginning session (oracle_endpoint_connection.c:834)</pre> <p>PostgreSQL 연결 문자열은 다음 예와 유사합니다.</p> <pre>2019-12-17T09:15:11 [TARGET_LOAD]I: Going to connect</pre>	<p>일반 AWS</p>

작업	설명	필요한 기술
	<pre>to ODBC connection string: PROTOCOL=7.4-0;DRIVER={PostgreSQL};SERVER=mypgsqlinstance.cokmvis0v46q.us-east-1.rds.amazonaws.com;DATABASE=pgdb;PORT=5432;sslmode=require;UID=pguser;(odbc_endpoint_imp.c:2218)</pre>	
데이터를 검증합니다.	<p>소스 Oracle 및 대상 PostgreSQL 데이터베이스에서 마이그레이션 작업 결과와 데이터를 검토합니다.</p> <ol style="list-style-type: none"> 1. PgAdmin에 연결하고 스키마 ORCL을(를) 사용하여 PostgreSQL 데이터베이스의 데이터를 확인합니다. 2. CDC의 경우 소스 Oracle 데이터베이스에 데이터를 삽입하거나 업데이트하여 진행 중인 변경 사항을 확인합니다. 	DBA
마이그레이션 작업을 중지합니다.	데이터 검증을 성공적으로 완료한 후 마이그레이션 작업을 중지합니다.	일반 AWS

리소스 정리

작업	설명	필요한 기술
AWS DMS 작업을 삭제합니다.	<ol style="list-style-type: none"> AWS DMS 콘솔에서 데이터베이스 마이그레이션 작업으로 이동하여 진행 중이거나 실행 중인 AWS DMS 작업을 중지합니다. 작업(task) 또는 작업들(tasks)을 선택한 후 작업을 선택하고 삭제를 선택합니다. 	일반 AWS
AWS DMS 엔드포인트를 삭제합니다.	생성한 소스 및 대상 엔드포인트를 선택한 후 작업을 선택한 다음 삭제를 선택합니다.	일반 AWS
AWS DMS 복제 인스턴스를 삭제합니다.	복제 인스턴스를 선택한 후 작업을 선택한 다음 삭제를 선택합니다.	일반 AWS
PostgreSQL 데이터베이스를 삭제합니다.	<ol style="list-style-type: none"> Amazon RDS 콘솔에서 데이터베이스를 선택합니다. 생성한 PostgreSQL 데이터베이스 인스턴스를 선택하고 작업을 선택한 다음 삭제를 선택합니다. 	일반 AWS
Oracle 데이터베이스를 삭제합니다.	Amazon RDS 콘솔에서 Oracle 데이터베이스 인스턴스를 선택한 후 작업을 선택한 다음 삭제를 선택합니다.	일반 AWS

문제 해결

문제	Solution
AWS SCT 소스 및 대상 테스트 연결이 실패합니다.	들어오는 트래픽을 수락하도록 JDBC 드라이버 버전과 VPC 보안 그룹 인바운드 규칙을 구성합니다.
Oracle 소스 엔드포인트 테스트 실행이 실패합니다.	엔드포인트 설정 및 복제 인스턴스가 사용 가능한지 확인합니다.
AWS DMS 작업 전체 로드 실행이 실패합니다.	소스 및 대상 데이터베이스의 데이터 유형 및 크기가 일치하는지 확인합니다.
AWS DMS 검증 마이그레이션 작업이 오류를 반환합니다.	<ol style="list-style-type: none"> 테이블에 프라이머리 키가 있는지 확인합니다. 프라이머리 키가 없는 테이블은 검증되지 않습니다. 테이블에 프라이머리 키가 있지만 오류가 반환되는 경우 소스 엔드포인트에서 추가 연결 속성을 확인합니다. 테이블에서 사용 가능한 데이터에 따라 동적으로 크기가 지정되지 않은 NUMBER 데이터 유형을 지원하려면, 추가 연결 속성에는 <code>numberDataTypeScale=-2</code> 이(가) 있어야 합니다.

관련 리소스

데이터베이스

- [Amazon RDS for Oracle](#)
- [Amazon RDS for PostgreSQL](#)

SSL DB 연결

- [SSL/TLS를 사용하여 DB 인스턴스 연결 암호화](#)
 - [RDS for Oracle DB 인스턴스에 SSL 사용](#)
 - [SSL/TLS를 사용한 RDS for PostgreSQL 연결 보안](#)

- [CA-2019 루트 인증서 다운로드](#)
- [옵션 그룹 작업](#)
 - [Oracle DB 인스턴스에 옵션 추가](#)
 - [Oracle Secure Sockets Layer](#)
- [파라미터 그룹 작업](#)
- [PostgreSQL sslmode 연결 파라미터](#)
- [JDBC의 SSL 사용](#)

AWS SCT

- [AWS Schema Conversion Tool](#)
- [AWS Schema Conversion Tool 사용 설명서](#)
- [AWS SCT 사용자 인터페이스 사용](#)
- [Oracle Database를 AWS SCT의 소스로 사용](#)

DMS

- [AWS Database Migration Service](#)
- [AWS Database Migration Service 사용자 가이드](#)
 - [Oracle 데이터베이스를 AWS DMS의 소스로 사용](#)
 - [PostgreSQL 데이터베이스를 AWS DMS 대상으로 사용](#)
- [AWS Database Migration Service과 함께 SSL 사용](#)
- [관계형 데이터베이스를 실행하는 애플리케이션을 AWS로 마이그레이션](#)

첨부

이 문서와 관련된 추가 콘텐츠에 액세스하려면 [attachment.zip](#) 파일의 압축을 풉니다.

AWS CLI와 AWS를 사용하여 AWS SCT 및 AWS DMS를 사용하여 오라클용 Amazon RDS를 PostgreSQL용 Amazon RDS로 마이그레이션합니다. CloudFormation

작성자: Pinesh Singal(AWS)

환경: PoC 또는 파일럿	소스: Amazon RDS for Oracle	대상: Amazon RDS for PostgreSQL
R 유형: 리아키텍트	워크로드: Oracle, 오픈 소스	기술: 마이그레이션, 데이터베이스

AWS 서비스: AWS DMS, Amazon RDS, AWS SCT

요약

이 패턴은 AWS Command Line Interface(AWS CLI)를 사용하여 멀티 테라바이트 [Oracle용 Amazon Relational Database Service\(Amazon RDS\)](#)를 [Amazon RDS for PostgreSQL](#) DB 인스턴스로 마이그레이션하는 방법을 보여줍니다. 이 접근 방식은 가동 중지 시간을 최소화하며 AWS Management Console에 로그인할 필요가 없습니다.

이 패턴을 사용하면 AWS Schema Conversion Tool(AWS SCT) 및 AWS Database Migration Service(AWS DMS) 콘솔을 사용하여 수동 구성 및 개별 마이그레이션을 피할 수 있습니다. 이 솔루션은 여러 데이터베이스에 대한 일회성 구성을 설정하고 AWS CLI에서 AWS SCT 및 AWS DMS를 사용하여 마이그레이션을 수행합니다.

이 패턴은 AWS SCT를 사용하여 데이터베이스 스키마 객체를 Amazon RDS for Oracle에서 Amazon RDS for PostgreSQL로 변환한 다음 AWS DMS를 사용하여 데이터를 마이그레이션합니다. AWS CLI에서 Python 스크립트를 사용하면 AWS 템플릿으로 AWS SCT 객체와 AWS DMS 작업을 생성할 수 있습니다. CloudFormation

사전 조건 및 제한 사항

사전 조건

- 활성 상태의 AWS 계정

- 기존 Amazon RDS for Oracle DB 인스턴스입니다.
- 기존 Amazon RDS for PostgreSQL DB 인스턴스입니다.
- 스크립트 실행을 위한 Windows 또는 Linux OS가 설치된 Amazon EC2 인스턴스 또는 로컬 시스템입니다.
- 다음 AWS DMS 마이그레이션 작업 유형 full-load, cdc, full-load-and-cdc에 관한 이해입니다. 자세한 내용은 AWS DMS 설명서의 [작업 생성](#)을 참조하세요.
- AWS SCT는 Oracle 및 PostgreSQL 데이터베이스 엔진용 Java Database Connectivity(JDBC) 드라이버와 함께 설치 및 구성되었습니다. 자세한 내용은 AWS SCT 설명서의 [AWS SCT 설치 및 필수 데이터베이스 드라이버 설치](#)를 참조하세요.
- 설치된 AWS SCT 폴더의 AWSSchemaConversionToolBatch.jar 파일이 작업 디렉토리로 복사되었습니다.
- 작업 디렉토리에서 cli-sct-dms-cft.zip 파일(첨부됨)이 다운로드 및 추출되었습니다.
- 최신 AWS DMS 복제 인스턴스 엔진 버전입니다. 자세한 내용은 AWS Support 설명서의 [AWS DMS 복제 인스턴스를 생성하는 방법](#) 및 AWS DMS 설명서의 [AWS DMS 3.4.4 릴리스 정보](#)를 참조하세요.
- AWS CLI 버전 2는 스크립트가 실행되는 Amazon Elastic Compute Cloud(Amazon EC2) 인스턴스 또는 운영 체제(OS)의 액세스 키 ID, 비밀 액세스 키 및 기본 AWS 리전 이름을 사용하여 설치 및 구성됩니다. 자세한 내용은 AWS CLI 설명서에서 [AWS CLI 버전 2 설치, 업데이트 및 제거](#) 및 [AWS CLI 구성](#)을 참조하세요.
- AWS CloudFormation 템플릿에 익숙해야 합니다. 자세한 내용은 [AWS CloudFormation 설명서의 AWS CloudFormation 개념](#)을 참조하십시오.
- Python 버전 3이 스크립트가 실행되는 Amazon EC2 인스턴스 또는 OS에 설치 및 구성되었습니다. 자세한 내용은 [Python 설명서](#)를 참조하세요.

제한 사항

- 소스 Amazon RDS for Oracle DB 인스턴스에 대한 최소 요구 사항은 다음과 같습니다.
 - Enterprise, Standard, Standard One 및 Standard Two 에디션용 Oracle 버전 12c(v12.1.0.2, v12.2.0.1), 18c (v18.0.0.0) 및 19c(v19.0.0.0)입니다.
 - Amazon RDS는 Oracle 18c (v18.0.0.0) 를 지원하지만, 오라클이 날짜 이후 18c에 대한 패치를 더 이상 제공하지 않기 때문에 이 버전은 지원 중단될 예정입니다. end-of-support 자세한 내용은 Amazon RDS 설명서에서 [Amazon RDS의 Oracle](#)을 참조하세요.

- Amazon RDS for Oracle 11g는 더 이상 지원되지 않습니다.
- 대상 Amazon RDS for PostgreSQL DB 인스턴스의 최소 요구 사항은 다음과 같습니다.
 - PostgreSQL 버전 9(버전 9.5 및 9.6) 10.x, 11.x, 12.x. 및 13.x

제품 버전

- Amazon RDS for Oracle DB 인스턴스 버전 12.1.0.2 이상
- Amazon RDS for PostgreSQL DB 인스턴스 버전 11.5 이상
- CLI version 2
- AWS SCT의 최신 버전
- Python 3의 최신 버전

아키텍처

소스 기술 스택

- Amazon RDS for Oracle

대상 기술 스택

- Amazon RDS for PostgreSQL

소스 및 대상 아키텍처

다음 다이어그램은 AWS DMS와 Python 스크립트를 사용하여 Amazon RDS for Oracle DB 인스턴스를 Amazon RDS for PostgreSQL DB 인스턴스로 마이그레이션하는 것을 보여줍니다.

이 다이어그램은 다음의 마이그레이션 워크플로우를 보여줍니다.

1. Python 스크립트는 AWS SCT를 사용하여 소스 및 대상 DB 인스턴스에 연결합니다.
2. 사용자는 Python 스크립트로 AWS SCT를 시작하고 Oracle 코드를 PostgreSQL 코드로 변환한 다음 대상 DB 인스턴스에서 실행합니다.
3. Python 스크립트는 소스 및 대상 DB 인스턴스에 대한 AWS DMS 복제 작업을 생성합니다.

4. 사용자는 Python 스크립트를 배포하여 AWS DMS 작업을 시작한 다음 데이터 마이그레이션이 완료된 후 작업을 중지합니다.

자동화 및 규모 조정

단일 프로그램의 여러 기능에 대한 추가 파라미터와 보안 관련 변경 사항을 Python 스크립트에 추가하여 이 마이그레이션을 자동화할 수 있습니다.

도구

- [AWS Command Line Interface\(AWS CLI\)](#)는 명령줄 쉘에서 명령을 사용하여 AWS 서비스와 상호 작용할 수 있는 오픈 소스 도구입니다.
- [AWS는 AWS CloudFormation](#) 리소스를 설정하고, 빠르고 일관되게 프로비저닝하고, 전체 AWS 계정 및 지역의 수명 주기 전반에 걸쳐 리소스를 관리할 수 있도록 지원합니다. 이 패턴은 Python 스크립트를 사용하여 .csv 입력 파일을 .json 입력 파일로 변환합니다. .json 파일은 AWS CLI 명령에서 Amazon 리소스 이름 (ARN), 마이그레이션 유형, 작업 설정 및 테이블 매핑으로 여러 AWS DMS 복제 작업을 생성하는 AWS CloudFormation 스택을 생성하는 데 사용됩니다.
- [AWS Database Migration Service\(DMS\)](#)를 사용하면 데이터 스토어를 클라우드로 마이그레이션하거나 클라우드와 온프레미스 설정 간에 데이터 스토어를 마이그레이션할 수 있습니다. 이 패턴은 AWS DMS를 사용하여 명령줄에서 실행되는 Python 스크립트로 작업을 생성, 시작 및 중지하고 AWS 템플릿을 생성합니다. CloudFormation
- [AWS Schema Conversion Tool\(AWS SCT\)](#)은 소스 데이터베이스 스키마와 대부분의 사용자 지정 코드를 대상 데이터베이스와 호환되는 형식으로 자동 변환하여 이기종 데이터베이스 마이그레이션을 지원합니다. 이 패턴에는 설치된 AWS SCT 디렉터리의 AWSSchemaConversionToolBatch.jar 파일이 필요합니다.

코드

cli-sct-dms-cft.zip 파일(첨부됨)에는 이 패턴의 전체 소스 코드가 들어 있습니다.

에픽

AWS CLI에서 AWS SCT 구성 및 데이터베이스 객체 생성

작업	설명	필요한 기술
AWS CLI에서 실행되도록 AWS SCT를 구성합니다.	1. 다음 형식을 사용하여 database_migration	DBA

작업	설명	필요한 기술
	<p>.txt 파일에서 소스 및 대상 환경 구성 세부 정보를 구성합니다.</p> <pre data-bbox="597 380 1026 1255">#source_vendor,source_hostname,source_dbname,source_user,source_pwd,source_schema,source_port,source_sid,target_vendor,target_hostname,target_user,target_pwd,target_dbname,target_port ORACLE,myoracledb.cokmvis@v46q.us-east-1.rds.amazonaws.com,ORCL,orcl,orcl1234,orcl,1521,ORCL,POSTGRESQL,mypgdbinstance.cokmvis@v46q.us-east-1.rds.amazonaws.com,pguser,pgpassword,pgdb,5432</pre> <p>2. project_settings.xml , Oracle_PG_Test_Batch.xml 및 ORACLE-orcl-to-POSTGRESQL.xml 파일의 요구 사항에 따라 AWS SCT 구성 파라미터를 수정합니다.</p>	

작업	설명	필요한 기술
run_aws_sct.py Python 스크립트를 실행합니다.	<p>다음 명령을 사용하여 run_aws_sct.py Python 스크립트를 실행합니다.</p> <pre>\$ python run_aws_sct.py database_migration.txt</pre> <p>Python 스크립트는 Oracle에서 PostgreSQL로 데이터베이스 객체를 변환하고 PostgreSQL 형식으로 SQL 파일을 생성합니다. 또한 스크립트는 데이터베이스 객체에 대한 자세한 권장 사항 및 변환 통계를 제공하는 Database migration assessment report .pdf 파일을 생성합니다.</p>	DBA
Amazon RDS for PostgreSQL에서 객체를 생성합니다.	<ol style="list-style-type: none"> 필요한 경우 AWS SCT에서 생성된 SQL 파일을 수동으로 수정합니다. Amazon RDS for PostgreSQL DB 인스턴스에서 SQL 파일을 실행하고 객체를 생성합니다. 	DBA

AWS CLI와 AWS를 사용하여 AWS DMS 작업을 구성하고 생성합니다. CloudFormation

작업	설명	필요한 기술
AWS DMS 복제 인스턴스를 생성합니다.	AWS Management Console에 로그인하고 AWS DMS 콘솔을 열고 요구 사항에 따라 구성된 복제 인스턴스를 생성합니다.	DBA

작업	설명	필요한 기술
	<p>자세한 내용은 AWS DMS 설명서의 복제 인스턴스 생성 및 AWS Support 설명서의 AWS DMS 복제 인스턴스를 생성하는 방법을 참조하세요.</p>	
<p>소스 엔드포인트를 생성합니다.</p>	<p>AWS DMS 콘솔에서 엔드포인트를 선택한 다음 요구 사항에 따라 Oracle 데이터베이스의 소스 엔드포인트를 생성합니다.</p> <p>참고: 추가 연결 속성은 -2 값이 포함된 numberDataScale 이어야 합니다.</p> <p>자세한 내용은 AWS DMS 설명서의 소스 및 대상 엔드포인트 생성을 참조하세요.</p>	DBA
<p>대상 엔드포인트를 생성합니다.</p>	<p>AWS DMS 콘솔에서 엔드포인트를 선택한 다음 요구 사항에 따라 PostgreSQL 데이터베이스의 대상 엔드포인트를 생성합니다.</p> <p>자세한 내용은 AWS DMS 설명서의 소스 및 대상 엔드포인트 생성을 참조하세요.</p>	DevOps 엔지니어

작업	설명	필요한 기술
<p>AWS CLI에서 실행되도록 AWS DMS 복제 세부 정보를 구성합니다.</p>	<p>다음 형식을 사용하여 소스 엔드포인트 ARN, 대상 엔드포인트 ARN, 복제 인스턴스 ARN으로 <code>dms-arn-list.txt</code> 파일의 AWS DMS 소스 및 대상 엔드포인트와 복제 세부 정보를 구성합니다.</p> <pre data-bbox="597 590 1027 1220"> #sourceARN,targetARN,repARN arn:aws:dms:us-east-1:123456789012:endpoint:EH7AINRUDZ5GOYIY6HVMXECMCQ arn:aws:dms:us-east-1:123456789012:endpoint:HHJVUV57N703CQF4PJZKGIOYY5 arn:aws:dms:us-east-1:123456789012:rep:LL57N77AQQAHHJF4PJFHNEZ5G </pre>	<p>DBA</p>

작업	설명	필요한 기술
<p>dms-create-task.py Python 스크립트를 실행하여 AWS DMS 작업을 생성합니다.</p>	<p>1. 다음 명령을 사용하여 dms-create-task.py Python 스크립트를 실행합니다.</p> <pre>\$ python dms-create-task.py database_migration.txt dms-arn-list.txt <cft-stack-name> <migration-type></pre> <ul style="list-style-type: none"> • database_migration.txt 은(는) 데이터베이스 마이그레이션 텍스트 파일임 • dms-arn-list.txt 은(는) AWS DMS의 ARN 목록임 • <cft-stack-name> 는 사용자 정의 AWS CloudFormation 스택 이름입니다. • <migration-type> 마이그레이션 유형 (전체 로드, cdc 또는) full-load-and-cdc <p>2. 마이그레이션 유형에 따라 다음 명령을 사용하여 세 가지 유형의 AWS DMS 작업을 생성할 수 있습니다.</p> <ul style="list-style-type: none"> • \$ python dms-create-task.py database_migration.txt dms-arn-list.txt dms- 	DBA

작업	설명	필요한 기술
	<pre>cli-cft-stack full-load</pre> <ul style="list-style-type: none"> \$ python dms-create-task.py database_migration.txt dms-arn-list.txt dms-cli-cft-stack cdc \$ python dms-create-task.py database_migration.txt dms-arn-list.txt dms-cli-cft-stack full-load-and-cdc <p>3. AWS CloudFormation 스택 및 AWS DMS 태스크가 생성됩니다.</p>	
AWS DMS 작업이 준비되었는지 확인합니다.	AWS Console에서 상태 섹션에서 AWS DMS 작업이 Ready 상태에 있는지 확인합니다.	DBA

AWS CLI를 사용하여 AWS DMS 작업 시작 및 중지

작업	설명	필요한 기술
AWS DMS 작업을 시작합니다.	다음 명령을 사용하여 dms-start-task.py Python 스크립트를 실행합니다.	DBA

작업	설명	필요한 기술
	<pre>\$ python dms-start-task.py start '<cdc-start-datetime>'</pre> <p>참고: 시작 날짜 및 시간은 'DD-MON-YYYY' 또는 'YYYY-MM-DDTHH:MI:SS' 타임스탬프 데이터 유형 형식이어야 합니다(예를 들어, '01-Dec-2019' 또는 '2018-03-08T12:12:12').</p> <p>AWS DMS 콘솔의 작업 페이지에 있는 마이그레이션 작업의 테이블 통계 탭에서 AWS DMS 작업 상태를 검토할 수 있습니다.</p>	

작업	설명	필요한 기술
데이터를 검증합니다.	<ol style="list-style-type: none"> 1. 전체 로드 마이그레이션이 완료된 후에도 CDC(지속적인 데이터 변경)를 위해 작업이 계속 실행됩니다. 2. CDC가 완료되거나 더 이상 변경 사항을 마이그레이션할 필요가 없으면 Oracle 및 PostgreSQL 데이터베이스에서 마이그레이션 작업 결과 및 데이터를 검토하고 검증합니다. 3. AWS DMS 콘솔의 작업 페이지에 있는 데이터베이스 마이그레이션 작업의 테이블 통계 탭에서 상태 및 개수 열(Validation state , Validation pending , Validation failed , Validation suspended 및 Validation details)을 확인하여 데이터를 검증할 수 있습니다. <p>자세한 내용은 AWS DMS 설명서의 AWS DMS 데이터 검증을 참조하세요.</p>	DBA

작업	설명	필요한 기술
AWS DMS 작업을 중지합니다.	<p>다음 명령을 사용하여 Python 스크립트를 실행합니다.</p> <pre>\$ python dms-start-task.py stop</pre> <p>참고: AWS DMS 작업은 검증 상태에 따라 failed 상태로 중지될 수 있습니다. 자세한 내용은 추가 정보 섹션의 문제 해결 표를 참조하세요.</p>	DBA

문제 해결

문제	Solution
AWS SCT 소스 및 대상 테스트 연결 실패	들어오는 트래픽을 수락하도록 JDBC 드라이버 버전과 VPC 보안 그룹 인바운드 규칙을 구성합니다.
소스 또는 대상 엔드포인트 테스트 실행 실패	<p>엔드포인트 설정 및 복제 인스턴스가 Available 상태에 있는지 확인합니다. 엔드포인트 연결 상태가 Successful 인지 확인합니다.</p> <p>자세한 내용은 AWS Support 설명서에서 AWS DMS 엔드포인트 연결 실패를 해결하는 방법을 참조하세요.</p>
전체 로드 실행 실패	<p>소스 및 대상 데이터베이스의 데이터 유형 및 크기가 일치하는지 확인합니다.</p> <p>자세한 내용은 AWS DMS 설명서의 AWS DMS의 마이그레이션 작업 문제 해결을 참조하세요.</p>

문제	Solution
실행 오류 검증	<p>비 프라이머리 키 테이블은 검증되지 않으므로 테이블에 프라이머리 키가 있는지 확인합니다.</p> <p>테이블에 프라이머리 키와 오류가 있는 경우 소스 엔드포인트의 추가 연결 속성에 <code>numberDataScale=-2</code> 이(가) 있는지 확인합니다.</p> <p>자세한 내용은 Oracle을 AWS DMS용 소스로 사용할 때의 추가 연결 속성 및 AWS DMS 설명서의 문제 해결을 참조하십시오. OracleSettings</p>

관련 리소스

- [AWS SCT 설치](#)
- [AWS DMS 소개 \(동영상\)](#)
- [AWS에서 AWS CLI 사용하기 CloudFormation](#)
- [AWS SCT 사용자 인터페이스 사용](#)
- [Oracle 데이터베이스를 AWS DMS의 소스로 사용](#)
- [Oracle을 AWS SCT의 소스로 사용](#)
- [PostgreSQL 데이터베이스를 AWS DMS 대상으로 사용](#)
- [AWS DMS에서 데이터 마이그레이션의 소스](#)
- [AWS DMS에서 데이터 마이그레이션의 대상](#)
- [cloudformation](#) (AWS CLI 설명서)
- [cloudformation 스택 생성](#) (AWS CLI 설명서)
- [dms](#) (AWS CLI 설명서)

첨부

이 문서와 관련된 추가 콘텐츠에 액세스하려면 [attachment.zip](#) 파일의 압축을 풉니다.

Oracle SERIALY_REUSEABLE 프라그마 패키지를 PostgreSQL로 마이그레이션

작성자: Vinay Paladi(AWS)

환경: PoC 또는 파일럿	소스: Oracle Database	대상: PostgreSQL
R 유형: 리아키텍트	워크로드: Oracle, 오픈 소스	기술: 마이그레이션, 데이터베이스
AWS 서비스: AWS SCT, Amazon Aurora		

요약

이 패턴은 SERIALY_REUSEABLE 프라그마로 정의된 Oracle 패키지를 Amazon Web Services (AWS)의 PostgreSQL로 마이그레이션하기 위한 step-by-step 접근 방식을 제공합니다. 이 접근 방식은 SERIALY_REUSEABLE 프라그마의 기능을 유지합니다.

PostgreSQL은 패키지 개념과 SERIALY_REUSEABLE 프라그마를 지원하지 않습니다. PostgreSQL에서 유사한 기능을 사용하려면 패키지용 스키마를 만들고 스키마 내에 모든 관련 객체(예: 함수, 프로시저, 형식)를 배포하면 됩니다. SERIALY_REUSEABLE 프라그마의 기능을 구현하기 위해 이 패턴에서 제공되는 예제 래퍼 함수 스크립트는 [AWS Schema Conversion Tool\(AWS SCT\) 확장 팩](#)을 사용합니다.

자세한 내용은 Oracle 설명서의 [SERIALY_REUSEABLE 프라그마](#)를 참조하십시오.

사전 조건 및 제한 사항

사전 조건

- 활성 상태의 AWS 계정.
- 최신 버전의 AWS SCT 및 필수 드라이버
- Amazon Aurora PostgreSQL-Compatible Edition 데이터베이스 또는 Amazon Relational Database Service(Amazon RDS) for PostgreSQL 데이터베이스

제품 버전

- Oracle Database 버전 10g 이상

아키텍처

소스 기술 스택

- Oracle Database 온프레미스

대상 기술 스택

- [Aurora PostgreSQL-Compatible](#) 또는 Amazon RDS for PostgreSQL
- AWS SCT

마이그레이션 아키텍처

도구

서비스

- [AWS Schema Conversion Tool\(AWS SCT\)](#)은 소스 데이터베이스 스키마와 대부분의 사용자 지정 코드를 대상 데이터베이스와 호환되는 형식으로 자동 변환하여 이기종 데이터베이스 마이그레이션을 지원합니다.
- [Amazon Aurora PostgreSQL-Compatible Edition](#)은 PostgreSQL 배포를 설정, 운영 및 확장할 수 있는 완전 관리형 ACID 준수 관계형 데이터베이스 엔진입니다.
- [Amazon Relational Database Service\(Amazon RDS\) for PostgreSQL](#)은 AWS 클라우드에서 PostgreSQL 관계형 데이터베이스를 설정, 운영 및 조정하는 데 도움이 됩니다.

기타 도구

- [pgAdmin](#)은 PostgreSQL을 위한 오픈 소스 관리 도구입니다. 데이터베이스 객체를 생성, 유지 관리 및 사용하는 데 도움이 되는 그래픽 인터페이스를 제공합니다.

에픽

AWS SCT를 사용하여 Oracle 패키지 마이그레이션

작업	설명	필요한 기술
AWS SCT를 설정합니다.	소스 데이터베이스에 대한 AWS SCT 연결을 구성합니다. 자세한 내용은 Oracle Database를 AWS SCT의 소스로 사용 을 참조하십시오.	DBA, 개발자
스크립트 변환.	AWS SCT를 사용하면 대상 데이터베이스를 Aurora PostgreSQL-Compatible로 선택하여 Oracle 패키지를 변환할 수 있습니다.	DBA, 개발자
.sql 파일을 저장합니다.	.sql 파일을 저장하기 전에 AWS SCT의 프로젝트 설정 옵션을 스테이지별 단일 파일로 수정합니다. AWS SCT는 객체 유형에 따라 .sql 파일을 여러 .sql 파일로 분리합니다.	DBA, 개발자
코드를 변경합니다.	AWS SCT에서 생성한 init 함수를 열고 추가 정보 섹션의 예제에 표시된 대로 변경합니다. pg_serialize = 0 함수를 구현하기 위한 변수가 추가됩니다.	DBA, 개발자
변환을 테스트합니다.	Aurora PostgreSQL-Compatible 데이터베이스에 init 함수를 배포하고 결과를 테스트합니다.	DBA, 개발자

관련 리소스

- [AWS Schema Conversion Tool](#)
- [Amazon RDS](#)
- [Amazon Aurora 기능](#)
- [SERIALLY_REUSABLE 프로그래밍](#)

추가 정보

Source Oracle Code:

```
CREATE OR REPLACE PACKAGE test_pkg_var
IS
PRAGMA SERIALLY_REUSABLE;
PROCEDURE function_1
(test_id number);
PROCEDURE function_2
(test_id number
);
END;

CREATE OR REPLACE PACKAGE BODY test_pkg_var
IS
PRAGMA SERIALLY_REUSABLE;
v_char VARCHAR2(20) := 'shared.airline';
v_num number := 123;

PROCEDURE function_1(test_id number)
IS
begin
dbms_output.put_line( 'v_char-'|| v_char);
dbms_output.put_line( 'v_num-'||v_num);
v_char:='test1';
function_2(0);
END;

PROCEDURE function_2(test_id number)
is
begin
dbms_output.put_line( 'v_char-'|| v_char);
dbms_output.put_line( 'v_num-'||v_num);
```

```
END;  
END test_pkg_var;
```

Calling the above functions

```
set serveroutput on
```

```
EXEC test_pkg_var.function_1(1);
```

```
EXEC test_pkg_var.function_2(1);
```

Target Postgresql Code:

```
CREATE SCHEMA test_pkg_var;
```

```
CREATE OR REPLACE FUNCTION test_pkg_var.init(pg_serialize IN INTEGER DEFAULT 0)
```

```
RETURNS void
```

```
AS
```

```
$BODY$
```

```
DECLARE
```

```
BEGIN
```

```
if aws_oracle_ext.is_package_initialized( 'test_pkg_var' ) AND pg_serialize = 0
```

```
then
```

```
return;
```

```
end if;
```

```
PERFORM aws_oracle_ext.set_package_initialized( 'test_pkg_var' );
```

```
PERFORM aws_oracle_ext.set_package_variable( 'test_pkg_var', 'v_char',  
'shared.airline.basecurrency'::CHARACTER
```

```
VARYING(100));
```

```
PERFORM aws_oracle_ext.set_package_variable('test_pkg_var', 'v_num', 123::integer);

END;

$BODY$

LANGUAGE plpgsql;

CREATE OR REPLACE FUNCTION test_pkg_var.function_1(pg_serialize int default 1)

RETURNS void
AS
$BODY$
DECLARE

BEGIN

PERFORM test_pkg_var.init(pg_serialize);

raise notice 'v_char%',aws_oracle_ext.get_package_variable( 'test_pkg_var', 'v_char');

raise notice 'v_num%',aws_oracle_ext.get_package_variable( 'test_pkg_var', 'v_num');

PERFORM aws_oracle_ext.set_package_variable( 'test_pkg_var', 'v_char',
' test1'::varchar);

PERFORM test_pkg_var.function_2(0);
END;

$BODY$
LANGUAGE plpgsql;

CREATE OR REPLACE FUNCTION test_pkg_var.function_2(IN pg_serialize integer default 1)

RETURNS void

AS

$BODY$

DECLARE
```

```
BEGIN

PERFORM test_pkg_var.init(pg_serialize);

raise notice 'v_char%',aws_oracle_ext.get_package_variable( 'test_pkg_var', 'v_char');

raise notice 'v_num%',aws_oracle_ext.get_package_variable( 'test_pkg_var', 'v_num');

END;
$BODY$
LANGUAGE plpgsql;
```

Calling the above functions

```
select test_pkg_var.function_1()

select test_pkg_var.function_2()
```

Oracle 외부 테이블을 Amazon Aurora PostgreSQL 호환으로 마이그레이션

작성자: Anuradha Chintha (AWS) 및 Rakesh Raghav (AWS)

환경: PoC 또는 파일럿	소스: Oracle	대상: Aurora PostgreSQL
R 유형: 리아키텍트	워크로드: 오픈 소스	기술: 마이그레이션, 데이터베이스, 현대화

AWS 서비스: AWS Identity and Access Management, AWS Lambda, Amazon S3, Amazon SNS, Amazon Aurora

요약

외부 테이블을 통해 Oracle은 데이터베이스 외부에 플랫폼 파일로 저장된 데이터를 쿼리할 수 있습니다. ORACLE_LOADER 드라이버를 사용하면 SQL*Loader 유틸리티로 로드할 수 있는 모든 형식으로 저장된 모든 데이터에 액세스할 수 있습니다. 외부 테이블에서는 데이터 조작 언어(DML)를 사용할 수 없지만 쿼리, 조인 및 정렬 작업에는 외부 테이블을 사용할 수 있습니다.

Amazon Aurora PostgreSQL 호환 버전은 Oracle의 외부 테이블과 유사한 기능을 제공하지 않습니다. 대신 현대화를 통해 기능 요구 사항을 충족하고 경제적인 확장 가능한 솔루션을 개발해야 합니다.

이 패턴은 aws_s3 확장 프로그램을 사용하여 다양한 유형의 Oracle 외부 테이블을 Amazon Web Services(AWS) 클라우드의 Aurora PostgreSQL 호환 버전으로 마이그레이션하는 단계를 제공합니다.

프로덕션 환경에 구현하기 전에 이 솔루션을 철저히 테스트하는 것이 좋습니다.

사전 조건 및 제한 사항

사전 조건

- 활성 상태의 AWS 계정.
- AWS Command Line Interface(AWS CLI)
- 사용 가능한 Aurora PostgreSQL 호환 데이터베이스 인스턴스입니다.
- 외부 테이블이 있는 온프레미스 Oracle 데이터베이스
- pg.Client API

• 데이터 파일

제한 사항

- 이 패턴은 Oracle 외부 테이블을 대체하는 기능을 제공하지 않습니다. 하지만 데이터베이스 현대화 목표를 달성하기 위해 단계와 샘플 코드를 더욱 개선할 수 있습니다.
- `aws_s3` 내보내기 및 가져오기 함수에 구분자로 전달되는 문자를 파일에 포함하지 않습니다.

제품 버전

- Amazon S3에서 PostgreSQL용 RDS로 가져오려면 데이터베이스에서 PostgreSQL 버전 10.7 이상을 실행 중이어야 합니다.

아키텍처

소스 기술 스택

- Oracle

소스 아키텍처

대상 기술 스택

- Amazon Aurora PostgreSQL 호환
- 아마존 CloudWatch
- AWS Lambda
- AWS Secrets Manager
- Amazon Simple Notification Service(SNS)
- Amazon Simple Storage Service(S3)

대상 아키텍처

다음은 솔루션의 고급 표현을 보이는 다이어그램입니다.

1. S3 버킷에 파일이 업로드됩니다.
2. Lambda 함수가 시작됩니다.
3. Lambda 함수는 DB 함수 직접 호출을 시작합니다.
4. Secrets Manager는 데이터베이스 액세스를 위한 보안 인증 정보를 제공합니다.
5. DB 함수에 따라 SNS 경보가 생성됩니다.

자동화 및 규모 조정

외부 테이블의 추가 또는 변경은 메타데이터 유지 관리를 통해 처리할 수 있습니다.

도구

- [Amazon Aurora PostgreSQL 호환](#)—Amazon Aurora PostgreSQL 호환 버전은 완전 관리형이며 PostgreSQL과 호환되고 ACID를 준수하는 관계형 데이터베이스 엔진으로 하이엔드 상용 데이터베이스의 속도와 신뢰성에 오픈 소스 데이터베이스의 비용 효율성을 결합합니다.
- [AWS CLI](#)—AWS Command Line Interface(AWS CLI)는 AWS 서비스를 관리하는 통합 도구입니다. 도구 하나만 다운로드하여 구성하면 여러 AWS 서비스를 명령줄에서 관리하고 스크립트를 통해 자동화할 수 있습니다.
- [아마존 CloudWatch](#) — 아마존은 아마존 S3 리소스 및 사용률을 CloudWatch 모니터링합니다.
- [AWS Lambda](#) – AWS Lambda는 서버를 프로비저닝하거나 관리하지 않고도 코드를 실행하고, 워크로드를 인식하는 클러스터 확장 로직을 생성하고, 이벤트 통합을 유지 관리하거나 런타임을 관리할 수 있도록 지원하는 서버리스 컴퓨팅 서비스입니다. 이 패턴에서 Lambda는 파일이 Amazon S3에 업로드될 때마다 데이터베이스 함수를 실행합니다.
- [AWS Secrets Manager](#)—AWS Secrets Manager는 보안 인증 정보 저장 및 검색을 위한 서비스입니다. Secrets Manager를 사용하면 보안 암호를 프로그래밍 방식으로 검색하도록 Secrets Manager에 API 직접 호출을 보내서 암호를 포함한 하드 코딩된 보안 인증 정보를 바꿀 수 있습니다.
- [Amazon S3](#)—Amazon Simple Storage Service(S3)는 Aurora PostgreSQL 호환 클러스터에서 사용하고 송수신할 파일을 수신하고 저장하는 스토리지 계층을 제공합니다.
- [aws_s3](#)—aws_s3 확장은 Amazon S3와 Aurora PostgreSQL 호환을 통합합니다.
- [Amazon SNS](#)—Amazon Simple Notification Service(SNS)는 게시자와 클라이언트에게 메시지를 전달 또는 전송하는 것을 조정하고 관리합니다. 이 패턴에서는 Amazon SNS를 사용하여 알림을 전송합니다.

code

파일이 S3 버킷에 배치될 때마다 처리 애플리케이션 또는 Lambda 함수에서 DB 함수를 생성하고 직접 호출해야 합니다. 자세한 내용은 코드(첨부)를 참조하세요.

에픽

외부 파일 생성

작업	설명	필요한 기술
외부 파일을 소스 데이터베이스에 추가합니다.	외부 파일을 생성하여 oracle 디렉터리로 이동합니다.	DBA

대성 구성 (Aurora PostgreSQL 호환)

작업	설명	필요한 기술
Aurora PostgreSQL 데이터베이스를 생성합니다.	Amazon Aurora PostgreSQL 호환 클러스터에서 DB 인스턴스를 생성합니다.	DBA
스키마, aws_s3 확장 프로그램, 테이블을 생성합니다.	추가 정보 섹션의 ext_tbl_scripts 에 있는 코드를 사용합니다. 테이블은 실제 테이블, 스테이징 테이블, 오류 및 로그 테이블, 메타테이블을 포함합니다.	DBA, 개발자
DB 함수를 생성합니다.	DB 함수를 생성하려면 추가 정보 섹션의 load_external_table_latest 함수 아래에 있는 코드를 사용합니다.	DBA, 개발자

Lambda 함수 생성 및 구성

작업	설명	필요한 기술
역할을 생성합니다.	Amazon S3 및 Amazon Relational Database Service(Amazon RDS)에 액세스할 수 있는 권한이 있는 역할을 생성합니다. 이 역할은 패턴 실행을 위해 Lambda에 할당됩니다.	DBA
Lambda 함수를 생성합니다.	<p>Amazon S3에서 파일 이름(예: <code>file_key = info.get('object', {}).get('key')</code>)을 읽고 파일 이름을 입력 파라미터로 사용하여 DB 함수(예: <code>cursor.callproc("load_external_tables", [file_key])</code>)를 직접 호출하는 Lambda 함수를 생성합니다.</p> <p>함수 호출 결과에 따라 SNS 알림이 시작됩니다 (예: <code>client.publish(TopicArn='arn:', Message='fileloadsuccess', Subject='fileloadsuccess')</code>).</p> <p>비즈니스 요구 사항에 따라 필요한 경우 추가 코드를 사용하여 Lambda 함수를 생성할 수 있습니다. 자세한 내용은 Lambda 설명서를 참조하세요.</p>	DBA

작업	설명	필요한 기술
S3 버킷 이벤트 트리거를 구성합니다.	S3 버킷의 모든 객체 생성 이벤트에 대해 Lambda 함수를 호출하는 메커니즘을 구성합니다.	DBA
보안 암호를 생성합니다.	Secrets Manager를 사용하여 데이터베이스 보안 인증 정보의 보안 암호 이름을 생성합니다. Lambda 함수로 시크릿을 전달합니다.	DBA
Lambda 지원 파일을 업로드합니다.	Lambda 지원 패키지와 Aurora PostgreSQL 호환 버전에 연결하기 위해 첨부된 Python 스크립트가 포함된.zip 파일을 업로드합니다. Python 코드는 데이터베이스에서 만든 함수를 호출합니다.	DBA
SNS 주제를 생성합니다.	SNS 주제를 만들어 데이터 로드의 성공 또는 실패에 대한 메일을 보냅니다.	DBA

Amazon S3와 통합 추가

작업	설명	필요한 기술
S3 버킷을 생성합니다.	Amazon S3 콘솔에서 선행 슬래시가 없는 고유한 이름의 S3 버킷을 생성합니다. S3 버킷 이름은 전역 수준에서 고유하며, 네임스페이스는 모든 AWS 계정이 공유합니다.	DBA

작업	설명	필요한 기술
IAM 정책을 생성합니다.	AWS Identity and Access Management(IAM) 정책을 생성하려면 추가 정보 섹션의 <code>s3bucketpolicy_for_import</code> 에 있는 코드를 사용합니다.	DBA
역할을 생성합니다.	Aurora PostgreSQL 호환 버전을 위한 두 개의 역할을 가져오기에 대한 역할 하나와 내보내기에 대한 역할 하나로 생성합니다. 역할에 해당하는 정책을 할당합니다.	DBA
Aurora PostgreSQL 호환 클러스터에 역할을 연결합니다.	역할 관리에서 Aurora PostgreSQL 클러스터에 역할 가져오기 및 내보내기를 연결합니다.	DBA
Aurora PostgreSQL 호환 버전에 대한 지원 객체를 생성합니다.	테이블 스크립트의 경우 추가 정보 섹션의 <code>ext_tbl_scripts</code> 에 있는 코드를 사용합니다. 사용자 지정 함수의 경우 추가 정보 섹션의 <code>load_external_Table_latest</code> 에 있는 코드를 사용합니다.	DBA

테스트 파일 처리

작업	설명	필요한 기술
S3 버킷에 파일을 업로드합니다.	<p>테스트 파일을 S3 버킷에 업로드하려면 AWS CLI에서 콘솔이나 다음 명령을 사용합니다.</p> <pre>aws s3 cp /Users/Desktop/ukpost/exttbl/"testing files"/aps s3://s3importtest/inputtext/aps</pre> <p>파일이 업로드되는 즉시 버킷 이벤트가 Lambda 함수를 시작합니다. 이 함수는 Aurora PostgreSQL 호환 함수를 실행합니다.</p>	DBA
데이터와 로그 및 오류 파일을 확인합니다.	Aurora PostgreSQL 호환 함수는 파일을 메인 테이블에 로드하고 S3 버킷에 .log 및 .bad 파일을 생성합니다.	DBA
솔루션을 모니터링합니다.	Amazon CloudWatch 콘솔에서 Lambda 함수를 모니터링합니다.	DBA

관련 리소스

- [Amazon S3 통합](#)
- [Amazon S3](#)
- [Amazon Aurora PostgreSQL 호환 버전 사용](#)
- [AWS Lambda](#)
- [아마존 CloudWatch](#)

- [AWS Secrets Manager](#)
- [SNS 알림 설정](#)

추가 정보

ext_table_scripts

```
CREATE EXTENSION aws_s3 CASCADE;
CREATE TABLE IF NOT EXISTS meta_EXTERNAL_TABLE
(
    table_name_stg character varying(100) ,
    table_name character varying(100) ,
    col_list character varying(1000) ,
    data_type character varying(100) ,
    col_order numeric,
    start_pos numeric,
    end_pos numeric,
    no_position character varying(100) ,
    date_mask character varying(100) ,
    delimiter character(1) ,
    directory character varying(100) ,
    file_name character varying(100) ,
    header_exist character varying(5)
);
CREATE TABLE IF NOT EXISTS ext_tbl_stg
(
    col1 text
);
CREATE TABLE IF NOT EXISTS error_table
(
    error_details text,
    file_name character varying(100),
    processed_time timestamp without time zone
);
CREATE TABLE IF NOT EXISTS log_table
(
    file_name character varying(50) COLLATE pg_catalog."default",
    processed_date timestamp without time zone,
    tot_rec_count numeric,
    proc_rec_count numeric,
    error_rec_count numeric
);
```

sample insert scripts of meta data:

```
INSERT INTO meta_EXTERNAL_TABLE (table_name_stg, table_name, col_list, data_type,
col_order, start_pos, end_pos, no_position, date_mask, delimiter, directory,
file_name, header_exist) VALUES ('F_EX_APS_TRANSACTIONS_STG', 'F_EX_APS_TRANSACTIONS',
'source_filename', 'character varying', 2, 8, 27, NULL, NULL, NULL, 'databasedev',
'externalinterface/loadaddr/APS', 'NO');
INSERT INTO meta_EXTERNAL_TABLE (table_name_stg, table_name, col_list, data_type,
col_order, start_pos, end_pos, no_position, date_mask, delimiter, directory,
file_name, header_exist) VALUES ('F_EX_APS_TRANSACTIONS_STG', 'F_EX_APS_TRANSACTIONS',
'record_type_identifer', 'character varying', 3, 28, 30, NULL, NULL, NULL,
'databasedev', 'externalinterface/loadaddr/APS', 'NO');
INSERT INTO meta_EXTERNAL_TABLE (table_name_stg, table_name, col_list, data_type,
col_order, start_pos, end_pos, no_position, date_mask, delimiter, directory,
file_name, header_exist) VALUES ('F_EX_APS_TRANSACTIONS_STG', 'F_EX_APS_TRANSACTIONS',
'fad_code', 'numeric', 4, 31, 36, NULL, NULL, NULL, 'databasedev', 'externalinterface/
loadaddr/APS', 'NO');
INSERT INTO meta_EXTERNAL_TABLE (table_name_stg, table_name, col_list, data_type,
col_order, start_pos, end_pos, no_position, date_mask, delimiter, directory,
file_name, header_exist) VALUES ('F_EX_APS_TRANSACTIONS_STG', 'F_EX_APS_TRANSACTIONS',
'session_sequence_number', 'numeric', 5, 37, 42, NULL, NULL, NULL, 'databasedev',
'externalinterface/loadaddr/APS', 'NO');
INSERT INTO meta_EXTERNAL_TABLE (table_name_stg, table_name, col_list, data_type,
col_order, start_pos, end_pos, no_position, date_mask, delimiter, directory,
file_name, header_exist) VALUES ('F_EX_APS_TRANSACTIONS_STG', 'F_EX_APS_TRANSACTIONS',
'transaction_sequence_number', 'numeric', 6, 43, 48, NULL, NULL, NULL, 'databasedev',
'externalinterface/loadaddr/APS', 'NO');
```

s3bucketpolicy_for import

```
---Import role policy
--Create an IAM policy to allow, Get, and list actions on S3 bucket
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "s3import",
      "Action": [
        "s3:GetObject",
        "s3:ListBucket"
      ],
      "Effect": "Allow",
      "Resource": [
        "arn:aws:s3:::s3importtest",
```

```

        "arn:aws:s3:::s3importtest/*"
    ]
}
}
--Export Role policy
--Create an IAM policy to allow, put, and list actions on S3 bucket
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "s3export",
      "Action": [
        "S3:PutObject",
        "s3:ListBucket"
      ],
      "Effect": "Allow",
      "Resource": [
        "arn:aws:s3:::s3importtest/*"
      ]
    }
  ]
}

```

샘플 DB 함수 load_external_tables_latest

```

CREATE OR REPLACE FUNCTION public.load_external_tables(pi_filename text)
  RETURNS character varying
  LANGUAGE plpgsql
AS $function$
/* Loading data from S3 bucket into a APG table */
DECLARE
  v_final_sql TEXT;
  pi_ext_table TEXT;
  r refCURSOR;
  v_sqlerrm text;
  v_chunk numeric;
  i integer;
  v_col_list TEXT;
  v_postion_list CHARACTER VARYING(1000);
  v_len integer;
  v_delim varchar;
  v_file_name CHARACTER VARYING(1000);

```

```
v_directory CHARACTER VARYING(1000);
v_table_name_stg CHARACTER VARYING(1000);
v_sql_col TEXT;
v_sql TEXT;
v_sql1 TEXT;
v_sql2 TEXT;
v_sql3 TEXT;
v_cnt integer;
v_sql_dynamic TEXT;
v_sql_ins TEXT;
proc_rec_COUNT integer;
error_rec_COUNT integer;
tot_rec_COUNT integer;
v_rec_val integer;
rec record;
v_col_cnt integer;
kv record;
v_val text;
v_header text;
j integer;
ERCODE VARCHAR(5);
v_region text;
cr CURSOR FOR
SELECT distinct DELIMITER,
       FILE_NAME,
       DIRECTORY
FROM meta_EXTERNAL_TABLE
WHERE table_name = pi_ext_table
       AND DELIMITER IS NOT NULL;

cr1 CURSOR FOR
SELECT col_list,
       data_type,
       start_pos,
       END_pos,
       concat_ws(' ',' ',TABLE_NAME_STG) as TABLE_NAME_STG,
       no_position,date_mask
FROM meta_EXTERNAL_TABLE
WHERE table_name = pi_ext_table
order by col_order asc;
cr2 cursor FOR
SELECT distinct table_name,table_name_stg
FROM meta_EXTERNAL_TABLE
```

```
WHERE upper(file_name) = upper(pi_filename);

BEGIN
-- PERFORM utl_file_utility.init();
v_region := 'us-east-1';
/* find tab details from file name */

--DELETE FROM ERROR_TABLE WHERE file_name= pi_filename;
-- DELETE FROM log_table WHERE file_name= pi_filename;

BEGIN

SELECT distinct table_name,table_name_stg INTO strict pi_ext_table,v_table_name_stg
FROM meta_EXTERNAL_TABLE
WHERE upper(file_name) = upper(pi_filename);
EXCEPTION
WHEN NO_DATA_FOUND THEN
raise notice 'error 1,%',sqlerrm;
pi_ext_table := null;
v_table_name_stg := null;
RAISE USING errcode = 'NTFIP' ;
when others then
raise notice 'error others,%',sqlerrm;
END;
j :=1 ;

for rec in cr2
LOOP

pi_ext_table := rec.table_name;
v_table_name_stg := rec.table_name_stg;
v_col_list := null;

IF pi_ext_table IS NOT NULL
THEN
--EXECUTE concat_ws('','truncate table ',pi_ext_table) ;
```

```

EXECUTE concat_ws('','truncate table ',v_table_name_stg) ;

SELECT distinct DELIMITER INTO STRICT v_delim
FROM meta_EXTERNAL_TABLE
WHERE table_name = pi_ext_table;

IF v_delim IS NOT NULL THEN
SELECT distinct DELIMITER,
FILE_NAME,
DIRECTORY ,
concat_ws('',' ',table_name_stg),
case header_exist when 'YES' then 'CSV HEADER' else 'CSV' end as header_exist
INTO STRICT v_delim,v_file_name,v_directory,v_table_name_stg,v_header
FROM meta_EXTERNAL_TABLE
WHERE table_name = pi_ext_table
AND DELIMITER IS NOT NULL;

IF upper(v_delim) = 'CSV'
THEN
v_sql := concat_ws('','SELECT aws_s3.table_import_FROM_s3 ( ','
v_table_name_stg,','','' ,
'DELIMITER ','',' CSV HEADER QUOTE ''''''''''', aws_commons.create_s3_uri
( ','
v_directory,','','v_file_name,',' ',v_region,')')');
ELSE
v_sql := concat_ws('','SELECT aws_s3.table_import_FROM_s3(','
v_table_name_stg, ','','' , 'DELIMITER AS ''''^''''','','' ,
aws_commons.create_s3_uri
( ','v_directory, ','','' ,
v_file_name, ',' ,
'''' ,v_region,')')
)');
raise notice 'v_sql , %',v_sql;
begin
EXECUTE v_sql;
EXCEPTION
WHEN OTHERS THEN
raise notice 'error 1';
RAISE USING errcode = 'S3IMP' ;

```

```

END;

select count(col_list) INTO v_col_cnt
from meta_EXTERNAL_TABLE where table_name = pi_ext_table;

-- raise notice 'v_sql 2, %',concat_ws('','update ',v_table_name_stg, ' set
col1 = col1||''',v_delim,''');

execute concat_ws('','update ',v_table_name_stg, ' set col1 =
col1||''',v_delim,''');

i :=1;
FOR rec in cr1
loop
v_sql1 := concat_ws('','v_sql1','split_part(col1,''',v_delim,','', i,')',' as
',rec.col_list,',');
v_sql2 := concat_ws('','v_sql2,rec.col_list,',');
-- v_sql3 := concat_ws('','v_sql3','rec.',rec.col_list,'::',rec.data_type,',');

case
WHEN upper(rec.data_type) = 'NUMERIC'
THEN v_sql3 := concat_ws('','v_sql3,' case WHEN
length(trim(split_part(col1,''',v_delim,','', i,))) =0
THEN null
ELSE
coalesce((trim(split_part(col1,''',v_delim,','', i,)))::NUMERIC,0)::',rec.data_type,' END as ',rec.col_list,',') ;
WHEN UPPER(rec.data_type) = 'TIMESTAMP WITHOUT TIME ZONE' AND rec.date_mask =
'YYYYMMDD'
THEN v_sql3 := concat_ws('','v_sql3,' case WHEN
length(trim(split_part(col1,''',v_delim,','', i,))) =0
THEN null
ELSE

```

```

        to_date(coalesce((trim(split_part(col1,'',v_delim,'',',',
i,'))),'99990101'),'YYYYMMDD'))::',rec.data_type,' END as ',rec.col_list,',');
        WHEN UPPER(rec.data_type) = 'TIMESTAMP WITHOUT TIME ZONE' AND rec.date_mask =
'MM/DD/YYYY hh24:mi:ss'
        THEN v_sql3 := concat_ws('',v_sql3,' case WHEN
length(trim(split_part(col1,'',v_delim,'',',', i,'))) =0
        THEN null
        ELSE
        to_date(coalesce((trim(split_part(col1,'',v_delim,'',',',
i,'))),'01/01/9999 0024:00:00'),'MM/DD/YYYY hh24:mi:ss'))::',rec.data_type,' END as
',rec.col_list,',');
        ELSE
        v_sql3 := concat_ws('',v_sql3,' case WHEN
length(trim(split_part(col1,'',v_delim,'',',', i,'))) =0
        THEN null
        ELSE
        coalesce((trim(split_part(col1,'',v_delim,'',',',
i,'))),''')::',rec.data_type,' END as ',rec.col_list,',') ;
        END case;

i :=i+1;
end loop;

-- raise notice 'v_sql 3, %',v_sql3;

SELECT trim(trailing ' ' FROM v_sql1) INTO v_sql1;
SELECT trim(trailing ',' FROM v_sql1) INTO v_sql1;

SELECT trim(trailing ' ' FROM v_sql2) INTO v_sql2;
SELECT trim(trailing ',' FROM v_sql2) INTO v_sql2;

SELECT trim(trailing ' ' FROM v_sql3) INTO v_sql3;
SELECT trim(trailing ',' FROM v_sql3) INTO v_sql3;

END IF;
raise notice 'v_delim , %',v_delim;

```

```

EXECUTE concat_ws('','SELECT COUNT(*) FROM ',v_table_name_stg) INTO v_cnt;

raise notice 'stg cnt , %',v_cnt;

/* if upper(v_delim) = 'CSV' then
   v_sql_ins := concat_ws('',' SELECT * from ' ,v_table_name_stg );
else
   -- v_sql_ins := concat_ws('',' SELECT ',v_sql1,' from (select col1 from
',v_table_name_stg , ')sub ');
   v_sql_ins := concat_ws('',' SELECT ',v_sql3,' from (select col1 from
',v_table_name_stg , ')sub ');
END IF;*/

v_chunk := v_cnt/100;

for i in 1..101
loop
  BEGIN
  -- raise notice 'v_sql , %',v_sql;
  -- raise notice 'Chunk number , %',i;
  v_sql_ins := concat_ws('',' SELECT ',v_sql3,' from (select col1 from
',v_table_name_stg , ' offset ',v_chunk*(i-1), ' limit ',v_chunk,') sub ');

  v_sql := concat_ws('','insert into ', pi_ext_table , ' ', v_sql_ins);
  -- raise notice 'select statement , %',v_sql_ins;
  -- v_sql := null;
  -- EXECUTE concat_ws('','insert into ', pi_ext_table , ' ', v_sql_ins, 'offset
',v_chunk*(i-1), ' limit ',v_chunk );
  --v_sql := concat_ws('','insert into ', pi_ext_table , ' ', v_sql_ins );

  -- raise notice 'insert statement , %',v_sql;

  raise NOTICE 'CHUNK START %',v_chunk*(i-1);
  raise NOTICE 'CHUNK END %',v_chunk;

```

```

EXECUTE v_sql;

EXCEPTION
  WHEN OTHERS THEN
    -- v_sql_ins := concat_ws('',' SELECT ',v_sql1, ' from (select col1 from
',v_table_name_stg , ' )sub ');
    -- raise notice 'Chunk number for cursor , %',i;

    raise NOTICE 'Cursor - CHUNK START %',v_chunk*(i-1);
    raise NOTICE 'Cursor -  CHUNK END %',v_chunk;
    v_sql_ins := concat_ws('',' SELECT ',v_sql3, ' from (select col1 from
',v_table_name_stg , ' )sub ');

    v_final_sql := REPLACE (v_sql_ins, '''::text, '''''::text);
    -- raise notice 'v_final_sql %',v_final_sql;
    v_sql :=concat_ws('','do $$ declare r refcursor;v_sql text; i
numeric;v_conname text; v_typ ',pi_ext_table,'[]; v_rec ', 'record',';
    begin

        open r for execute ''select col1 from ',v_table_name_stg ,' offset
',v_chunk*(i-1), ' limit ',v_chunk,''';
        loop
        begin
        fetch r into v_rec;
        EXIT WHEN NOT FOUND;

        v_sql := concat_ws('','','insert into ',pi_ext_table,' SELECT ',REPLACE
(v_sql3, '''::text, '''''::text) , ' from ( select ''''',v_rec.col1,''''' as
col1) v''');
        execute v_sql;

```

```

        exception
        when others then
            v_sql := 'INSERT INTO ERROR_TABLE VALUES (concat_ws(''''''''', ''''Error
Name: ''', $$''||SQLERRM||''$$, ''''Error State: ''', ''''''''||
SQLSTATE||''''''', ''''record : ''', $$''||v_rec.col1||''$$), ''''''||
pi_filename||''''',now())''';

            execute v_sql;
            continue;
        end ;
    end loop;
    close r;
    exception
    when others then
        raise;
    end ; $$');
-- raise notice ' inside excp v_sql %',v_sql;
    execute v_sql;
-- raise notice 'v_sql %',v_sql;
    END;
END LOOP;
ELSE

SELECT distinct DELIMITER,FILE_NAME,DIRECTORY ,concat_ws('',' ',table_name_stg),
    case header_exist when 'YES' then 'CSV HEADER' else 'CSV' end as header_exist
    INTO STRICT v_delim,v_file_name,v_directory,v_table_name_stg,v_header
FROM meta_EXTERNAL_TABLE
WHERE table_name = pi_ext_table ;
v_sql := concat_ws('','SELECT aws_s3.table_import_FROM_s3(''','
    v_table_name_stg, ''','''', ''DELIMITER AS ''''#'''' ',v_header,' ''',''
aws_commons.create_s3_uri
( ''',v_directory, ''','''',
    v_file_name, ''','',
    ''',v_region, ''')
)');
    EXECUTE v_sql;

FOR rec in cr1
LOOP

```

```

IF rec.start_pos IS NULL AND rec.END_pos IS NULL AND rec.no_position = 'recnum'
THEN
  v_rec_val := 1;
ELSE

  case
    WHEN upper(rec.data_type) = 'NUMERIC'
    THEN v_sql1 := concat_ws('',' case WHEN length(trim(substring(COL1,
',rec.start_pos ',' , rec.END_pos, '-',rec.start_pos ,'+1))) =0
    THEN null
    ELSE
      coalesce((trim(substring(COL1, ',rec.start_pos ',' ,
rec.END_pos, '-',rec.start_pos ,'+1)))::NUMERIC,0)::',rec.data_type,' END as
',rec.col_list,',') ;
    WHEN UPPER(rec.data_type) = 'TIMESTAMP WITHOUT TIME ZONE' AND rec.date_mask =
'YYYYMMDD'
    THEN v_sql1 := concat_ws('','case WHEN length(trim(substring(COL1,
',rec.start_pos ',' , rec.END_pos, '-',rec.start_pos ,'+1))) =0
    THEN null
    ELSE
      to_date(coalesce((trim(substring(COL1, ',rec.start_pos ',' ,
rec.END_pos, '-',rec.start_pos ,'+1))), '99990101'), 'YYYYMMDD')::',rec.data_type,'
END as ',rec.col_list,',');
    WHEN UPPER(rec.data_type) = 'TIMESTAMP WITHOUT TIME ZONE' AND rec.date_mask =
'YYYYMMDDHH24MISS'
    THEN v_sql1 := concat_ws('','case WHEN length(trim(substring(COL1,
',rec.start_pos ',' , rec.END_pos, '-',rec.start_pos ,'+1))) =0
    THEN null
    ELSE
      to_date(coalesce((trim(substring(COL1, ',rec.start_pos ',' ,
rec.END_pos, '-',rec.start_pos ,'+1))), '9999010100240000'), 'YYYYMMDDHH24MISS')::',rec.data_
END as ',rec.col_list,',');
    ELSE
      v_sql1 := concat_ws('',' case WHEN length(trim(substring(COL1,
',rec.start_pos ',' , rec.END_pos, '-',rec.start_pos ,'+1))) =0
    THEN null
    ELSE
      coalesce((trim(substring(COL1, ',rec.start_pos ',' ,
rec.END_pos, '-',rec.start_pos ,'+1))), '')::',rec.data_type,' END as
',rec.col_list,',') ;
  END case;

```

```

END IF;
v_col_list := concat_ws(',',v_col_list ,v_sql1);
END LOOP;

SELECT trim(trailing ' ' FROM v_col_list) INTO v_col_list;
SELECT trim(trailing ',' FROM v_col_list) INTO v_col_list;

v_sql_col := concat_ws(',',trim(trailing ',' FROM v_col_list) , ' FROM
',v_table_name_stg,' WHERE col1 IS NOT NULL AND length(col1)>0 ');

v_sql_dynamic := v_sql_col;

EXECUTE concat_ws('','SELECT COUNT(*) FROM ',v_table_name_stg) INTO v_cnt;

IF v_rec_val = 1 THEN
    v_sql_ins := concat_ws('',' select row_number() over(order by ctid) as
line_number ,' ,v_sql_dynamic) ;

ELSE
    v_sql_ins := concat_ws('',' SELECT' ,v_sql_dynamic) ;
END IF;

BEGIN
EXECUTE concat_ws('','insert into ', pi_ext_table ,' ', v_sql_ins);
EXCEPTION
    WHEN OTHERS THEN
        IF v_rec_val = 1 THEN
            v_final_sql := ' select row_number() over(order by ctid) as
line_number ,col1 from ';
        ELSE
            v_final_sql := ' SELECT col1 from';
        END IF;

```

```

        END IF;
        v_sql :=concat_ws('','do $$ declare  r refcursor;v_rec_val numeric :=
',coalesce(v_rec_val,0),';line_number numeric; col1 text; v_typ  ',pi_ext_table,'[];
v_rec  ',pi_ext_table,');
        begin
            open r for execute ''' ,v_final_sql, ' ',v_table_name_stg,' WHERE col1 IS
NOT NULL AND length(col1)>0 ' ' ;
            loop
                begin
                    if  v_rec_val = 1 then
                        fetch r into line_number,col1;
                    else
                        fetch r into col1;
                    end if;

                    EXIT WHEN NOT FOUND;
                    if v_rec_val = 1 then
                        select line_number,',trim(trailing ', ' FROM v_col_list) ,' into v_rec;
                    else
                        select ',trim(trailing ', ' FROM v_col_list) ,' into v_rec;
                    end if;

                    insert into  ',pi_ext_table,' select v_rec.*;
                    exception
                    when others then
                        INSERT INTO  ERROR_TABLE VALUES (concat_ws('','','Error Name:
'',SQLERRM,'Error State: ',SQLSTATE,'record : ',v_rec),'',pi_filename,'',now());
                        continue;
                    end ;
                    end loop;
                close r;
                exception
                when others then
                    raise;
                end ; $$');
            execute v_sql;

            END;

            END IF;

```

```
EXECUTE concat_ws('','SELECT COUNT(*) FROM ',pi_ext_table) INTO proc_rec_COUNT;

EXECUTE concat_ws('','SELECT COUNT(*) FROM error_table WHERE file_name
=''||pi_filename, '' and processed_time::date = clock_timestamp()::date') INTO
error_rec_COUNT;

EXECUTE concat_ws('','SELECT COUNT(*) FROM ',v_table_name_stg) INTO tot_rec_COUNT;

INSERT INTO log_table values(pi_filename,now(),tot_rec_COUNT,proc_rec_COUNT,
error_rec_COUNT);

raise notice 'v_directory, %',v_directory;

raise notice 'pi_filename, %',pi_filename;

raise notice 'v_region, %',v_region;

perform aws_s3.query_export_to_s3('SELECT
replace(trim(substring(error_details,position('(' in
error_details)+1),''),''),'','',';'),file_name,processed_time FROM error_table WHERE
file_name = ''||pi_filename||'',
aws_commons.create_s3_uri(v_directory, pi_filename||'.bad', v_region),
options :='FORmat csv, header, delimiter $$,$$'
);

raise notice 'v_directory, %',v_directory;

raise notice 'pi_filename, %',pi_filename;

raise notice 'v_region, %',v_region;
```

```

perform aws_s3.query_export_to_s3('SELECT * FROM log_table WHERE file_name = '''||
pi_filename||''',
  aws_commons.create_s3_uri(v_directory, pi_filename||'.log', v_region),
  options := 'FORmat csv, header, delimiter $$,$$'
);

END IF;
j := j+1;
END LOOP;

RETURN 'OK';
EXCEPTION
  WHEN OTHERS THEN
raise notice 'error %',sqlerrm;
  ERCODE=SQLSTATE;
  IF ERCODE = 'NTFIP' THEN
    v_sqlerrm := concat_ws(' ',sqlerrm,'No data for the filename');
  ELSIF ERCODE = 'S3IMP' THEN
    v_sqlerrm := concat_ws(' ',sqlerrm,'Error While exporting the file from S3');
  ELSE
    v_sqlerrm := sqlerrm;
  END IF;

select distinct directory into v_directory from meta_EXTERNAL_TABLE;

raise notice 'exc v_directory, %',v_directory;

raise notice 'exc pi_filename, %',pi_filename;

raise notice 'exc v_region, %',v_region;

perform aws_s3.query_export_to_s3('SELECT * FROM error_table WHERE file_name = '''||
pi_filename||''',

```

```
aws_commons.create_s3_uri(v_directory, pi_filename||'.bad', v_region),
options :='FORmat csv, header, delimiter $$,$$'
);
RETURN null;
END;
$function$
```

함수 기반 인덱스를 Oracle에서 PostgreSQL로 마이그레이션

작성자: Veeranjaneeyulu Grandhi(AWS) 및 Navakanth Talluri(AWS)

환경: 프로덕션	소스: Oracle	대상: PostgreSQL
R 유형: 리아키텍트	워크로드: Oracle	기술: 마이그레이션, 데이터베이스

요약

인덱스는 데이터베이스 성능을 향상시키는 일반적인 방법입니다. 인덱스를 사용하면 데이터베이스 서버가 인덱스가 없을 때보다 훨씬 빠르게 특정 열을 찾고 검색할 수 있습니다. 그러나 인덱스는 데이터베이스 시스템 전체에 오버헤드를 가중시키기도 하므로 현명하게 사용해야 합니다. 함수나 표현식을 기반으로 하는 함수 기반 인덱스에는 여러 열과 수학 표현식이 포함될 수 있습니다. 함수 기반 인덱스는 인덱스 표현식을 사용하는 쿼리의 성능을 개선합니다.

PostgreSQL은 기본적으로 변동성이 안정적이라고 정의된 함수를 사용하여 함수 기반 인덱스를 만드는 것을 지원하지 않습니다. 하지만 IMMUTABLE처럼 변동성이 있는 유사한 함수를 만들어 인덱스 생성에 사용할 수 있습니다.

IMMUTABLE 함수는 데이터베이스를 수정할 수 없으며 동일한 인수가 주어지면 영원히 동일한 결과를 반환할 수 있습니다. 이 범주를 사용하면 쿼리가 상수 인수를 사용하여 함수를 직접적으로 호출할 때 최적화 프로그램이 함수를 미리 평가할 수 있습니다.

이 패턴은 to_char, to_date 및 to_number와 같은 함수와 함께 사용할 경우 Oracle 함수 기반 인덱스를 해당하는 PostgreSQL로 마이그레이션하는 데 도움이 됩니다.

사전 조건 및 제한 사항

사전 조건

- 활성 Amazon Web Services(AWS) 계정
- 리스너 서비스가 설정되어 실행 중인 소스 Oracle 데이터베이스 인스턴스
- PostgreSQL 데이터베이스에 대한 지식

제한 사항

- 데이터베이스 크기 제한은 64TB입니다.
- 인덱스 생성에 사용되는 함수는 변경할 수 없어야 합니다.

제품 버전

- 버전 11g(버전 11.2.0.3.v1 이상) 및 12.2 이하, 18c의 모든 Oracle 데이터베이스 에디션이 포함됩니다.
- PostgreSQL 버전 9.6 이상

아키텍처

소스 기술 스택

- Oracle 데이터베이스 온프레미스 또는 Amazon Elastic Compute Cloud(Amazon EC2) 인스턴스 또는 Amazon RDS for Oracle DB 인스턴스

대상 기술 스택

- 모든 PostgreSQL 엔진

도구

- pgAdmin 4는 Postgres를 위한 오픈 소스 관리 도구입니다. pgAdmin 4 도구는 데이터베이스 객체를 생성, 유지 관리 및 사용하기 위한 그래픽 인터페이스를 제공합니다.
- Oracle SQL Developer는 기존 배포와 클라우드 배포 모두에서 Oracle Database를 개발하고 관리하기 위한 통합 개발 환경(IDE)입니다.

에픽

기본 함수를 사용하여 함수 기반 인덱스 생성

작업	설명	필요한 기술
to_char 함수를 사용하여 열에 함수 기반 인덱스를 생성합니다.	다음 코드를 사용하여 함수 기반 인덱스를 생성합니다.	DBA, 앱 개발자

작업	설명	필요한 기술
	<pre> postgres=# create table funcindex(col1 timestamp without time zone); CREATE TABLE postgres=# insert into funcindex values (now()); INSERT 0 1 postgres=# select * from funcindex; col1 ----- 2022-08-09 16:00:57. 77414 (1 rows) postgres=# create index funcindex_idx on funcindex(to_char(col1, 'DD-MM-YYYY HH24:MI:SS')); ERROR: functions in index expression must be marked IMMUTABLE </pre> <p>참고: PostgreSQL에서는 IMMUTABLE 절이 없는 함수 기반 인덱스를 생성할 수 없습니다.</p>	
<p>함수의 변동성을 확인합니다.</p>	<p>함수 변동성을 확인하려면 추가 정보 섹션의 코드를 사용합니다.</p>	<p>DBA</p>

래퍼 함수를 사용하여 함수 기반 인덱스 생성

작업	설명	필요한 기술
래퍼 함수를 생성합니다.	래퍼 함수를 생성하려면 추가 정보 섹션의 코드를 사용합니다.	PostgreSQL 개발자
래퍼 함수를 사용하여 인덱스를 생성합니다.	<p>추가 정보 섹션의 코드를 사용하여 애플리케이션과 동일한 스키마에서 키워드 IMMUTABLE 이 포함된 사용자 정의 함수를 만들고 인덱스 생성 스크립트에서 해당 함수를 참조할 수 있습니다.</p> <p>이전 예제의 공통 스키마에서 사용자 정의 함수를 만든 경우 search_path 를 그림과 같이 업데이트합니다.</p> <pre>ALTER ROLE <ROLENAME> set search_path=\$user, COMMON;</pre>	DBA, PostgreSQL 개발자

인덱스 생성 확인

작업	설명	필요한 기술
인덱스 생성 확인.	쿼리 액세스 패턴을 기반으로 인덱스 생성이 필요한지 확인합니다.	DBA
인덱스를 사용할 수 있는지 확인합니다.	PostgreSQL Optimizer에서 함수 기반 인덱스를 선택했는지 확인하려면 설명 또는 설명 분석을 사용하여 SQL 명령문을	DBA

작업	설명	필요한 기술
	<p>실행합니다. 추가 정보 섹션의 코드를 사용합니다. 가능하면 테이블 통계도 수집합니다.</p> <p>참고: 설명 계획을 보면 알 수 있듯이 PostgreSQL 옵티마이저가 조건자 조건 때문에 함수 기반 인덱스를 선택한 것입니다.</p>	

관련 리소스

- [함수 기반 인덱스](#)(오라클 설명서)
- [표현식에 대한 인덱스](#)(PostgreSQL 설명서)
- [PostgreSQL 변동성](#)(PostgreSQL 설명서)
- [PostgreSQL search_path](#)(PostgreSQL 설명서)
- [Oracle Database 19c - Amazon Aurora PostgreSQL Migration Playbook](#)

추가 정보

래퍼 함수 생성

```
CREATE OR REPLACE FUNCTION myschema.to_char(var1 timestamp without time zone, var2
varchar) RETURNS varchar AS $BODY$ select to_char(var1, 'YYYYMMDD'); $BODY$ LANGUAGE
sql IMMUTABLE;
```

래퍼 함수를 사용하여 인덱스 생성

```
postgres=# create function common.to_char(var1 timestamp without time zone, var2
varchar) RETURNS varchar AS $BODY$ select to_char(var1, 'YYYYMMDD'); $BODY$ LANGUAGE
sql IMMUTABLE;
CREATE FUNCTION
postgres=# create index funcindex_idx on funcindex(common.to_char(col1,'DD-MM-YYYY
HH24:MI:SS'));
CREATE INDEX
```

함수의 변동성 확인

```
SELECT DISTINCT p.proname as "Name",p.provolatile as "volatility" FROM
pg_catalog.pg_proc p
LEFT JOIN pg_catalog.pg_namespace n ON n.oid = p.pronamespace
LEFT JOIN pg_catalog.pg_language l ON l.oid = p.prolang
WHERE n.nspname OPERATOR(pg_catalog.~) '^(pg_catalog)$' COLLATE pg_catalog.default AND
p.proname='to_char'GROUP BY p.proname,p.provolatile
ORDER BY 1;
```

인덱스 사용 여부 확인

```
explain analyze <SQL>
```

```
postgres=# explain select col1 from funcindex where common.to_char(col1,'DD-MM-YYYY
HH24:MI:SS') = '09-08-2022 16:00:57';
```

QUERY PLAN

```
-----
Index Scan using funcindex_idx on funcindex (cost=0.42..8.44 rows=1 width=8)
  Index Cond: ((common.to_char(col1, 'DD-MM-YYYY HH24:MI:SS'::character
varying))::text = '09-08-2022 16:00:57'::text)
(2 rows)
```

확장 기능을 사용하여 Oracle 네이티브 함수를 PostgreSQL로 마이그레이션

작성자: Pinesh Singal(AWS)

환경: PoC 또는 파일럿	소스: 데이터베이스: 관계형	대상: Amazon RDS PostgreSQL
R 유형: 리아키텍트	워크로드: Oracle, 오픈 소스	기술: 마이그레이션, 데이터베이스
AWS 서비스: Amazon EC2, Amazon RDS		

요약

이 마이그레이션 패턴은 및 확장을 PostgreSQL () 네이티브 내장 코드로 수정하여 오라클용 Amazon RDS 데이터베이스 인스턴스를 PostgreSQL용 Amazon RDS 또는 Amazon Aurora PostgreSQL 호환 에디션 데이터베이스로 마이그레이션하기 위한 step-by-step 지침을 제공합니다. `aws_oracle_ext` `orafce` `psql` 이렇게 하면 처리 시간이 절감됩니다.

이 패턴은 트랜잭션 수가 많은 수 테라바이트의 Oracle 소스 데이터베이스에 대해 다운타임이 없는 오프라인 수동 마이그레이션 전략을 설명합니다.

마이그레이션 프로세스는 `aws_oracle_ext` 및 `orafce` 확장 기능과 함께 AWS Schema Conversion Tool(AWS SCT)을 사용하여 Amazon RDS for Oracle 데이터베이스 스키마를 Amazon RDS for PostgreSQL 또는 Aurora PostgreSQL-Compatible 데이터베이스 스키마로 변환합니다. 그런 다음 코드를 PostgreSQL이 지원하는 네이티브 `psql` 내장 코드로 수동으로 변경합니다. 이는 확장 호출이 PostgreSQL 데이터베이스 서버의 코드 처리에 영향을 미치며 모든 확장 코드가 PostgreSQL 코드와 완전히 호환되지 않거나 호환되지 않기 때문입니다.

이 패턴은 주로 AWS SCT와 확장 기능 `aws_oracle_ext` 및 `orafce`를 사용하여 SQL 코드를 수동으로 마이그레이션하는 데 중점을 둡니다. 이미 사용된 확장 기능을 네이티브 PostgreSQL(`psql`) 내장 기능으로 변환합니다. 그런 다음 확장 기능에 대한 모든 참조를 제거하고 그에 따라 코드를 변환합니다.

사전 조건 및 제한 사항

사전 조건

- 활성 상태의 AWS 계정.
- 운영 체제(Windows 또는 Mac) 또는 Amazon EC2 인스턴스(가동 및 실행 중)
- Orafce

제한 사항

aws_oracle_ext 또는 orafce 확장 기능을 사용하는 모든 Oracle 함수를 네이티브 PostgreSQL 함수로 변환할 수 있는 것은 아닙니다. PostgreSQL 라이브러리로 컴파일하려면 수동 재작업이 필요할 수 있습니다.

AWS SCT 확장 기능 사용의 한 가지 단점은 실행 및 결과 가져오기 성능이 느리다는 것입니다. 첨부 문서의 성능 비교 검사 섹션에 설명된 대로 세 코드(aws_oracle_ext, orafce 및 psql 기본값) 간에 Oracle SYSDATE 함수를 PostgreSQL NOW() 함수로 마이그레이션하는 간단한 [PostgreSQL EXPLAIN 계획](#)(명령문의 실행 계획)을 통해 비용을 이해할 수 있습니다.

제품 버전

- 소스: Amazon RDS for Oracle 데이터베이스 10.2 이상(10.x용), 11g(11.2.0.3.v1 이상) 및 Enterprise Edition, Standard Edition, Standard Edition 1 및 Standard Edition 2의 경우 최대 12.2, 18c, 19c(이상)
- 대상: Amazon RDS for PostgreSQL 또는 Aurora PostgreSQL-Compatible 데이터베이스 9.4 이상(9.x용), 10.x, 11.x, 12.x, 13.x 및 14.x(이상)
- AWS SCT: 최신 버전(이 패턴은 1.0.632에서 테스트됨)
- Oracle: 최신 버전(이 패턴은 3.9.0에서 테스트됨)

아키텍처

소스 기술 스택

- 버전 12.1.0.2.v18의 Amazon RDS for Oracle 데이터베이스 인스턴스

대상 기술 스택

- 버전 11.5의 Amazon RDS for PostgreSQL 또는 Aurora PostgreSQL-Compatible 데이터베이스 인스턴스

데이터베이스 마이그레이션 아키텍처

다음 다이어그램은 소스 Oracle과 대상 PostgreSQL 데이터베이스 간의 데이터베이스 마이그레이션 아키텍처를 나타냅니다. 아키텍처에는 AWS 클라우드, Virtual Private Cloud(VPC), 가용 영역, 프라이빗 서브넷, Amazon RDS for Oracle 데이터베이스, AWS SCT, Amazon RDS for PostgreSQL 또는 Aurora PostgreSQL-Compatible 데이터베이스, Oracle용 확장 기능(aws_oracle_ext 및 orafce), 구조화된 쿼리 언어(SQL) 파일이 포함됩니다.

1. Amazon RDS for Oracle DB 인스턴스(소스 DB)를 시작합니다.
2. aws_oracle_ext 및 orafce 확장 팩과 함께 AWS SCT를 사용하여 소스 코드를 Oracle에서 PostgreSQL로 변환할 수 있습니다.
3. 변환 과정에서 PostgreSQL이 지원하는 마이그레이션된 .sql 파일이 생성됩니다.
4. 변환되지 않은 Oracle 확장 코드를 PostgreSQL(psql) 코드로 수동으로 변환합니다.
5. 수동 변환은 PostgreSQL이 지원하는 변환된 .sql 파일을 생성합니다.
6. 이 .sql 파일을 Amazon RDS for PostgreSQL DB 인스턴스(대상 DB)에서 실행합니다.

도구

도구

서비스

- [AWS SCT](#) - AWS Schema Conversion Tool(AWS SCT)은 기존 데이터베이스 스키마를 한 데이터베이스 엔진에서 다른 데이터베이스 엔진으로 변환합니다. 관계형 OLTP(Online Transactional Processing) 스키마 또는 데이터 웨어하우스 스키마를 변환할 수 있습니다. 변환된 스키마는 Amazon RDS for MySQL DB 인스턴스, Amazon Aurora DB 클러스터, Amazon RDS for PostgreSQL DB 인스턴스 또는 Amazon Redshift 클러스터에 적합합니다. 변환된 스키마는 Amazon EC2 인스턴스에서 데이터베이스와 함께 사용하거나 Amazon S3 버킷에 데이터로 저장할 수 있습니다.

AWS SCT는 소스 데이터베이스의 데이터베이스 스키마를 대상 Amazon RDS 인스턴스와 호환되는 형식으로 자동 변환할 수 있는 프로젝트 기반 사용자 인터페이스를 제공합니다.

AWS SCT를 사용하여 Oracle 소스 데이터베이스에서 위에 나열된 대상 중 하나로 마이그레이션할 수 있습니다. AWS SCT를 사용하면 스키마, 뷰, 저장된 프로시저 및 함수 등의 소스 데이터베이스 객체 정의를 내보낼 수 있습니다.

이 외부 스키마를 사용하여 Amazon RDS for PostgreSQL 또는 Amazon Aurora PostgreSQL-Compatible Edition 데이터베이스에 연결할 수 있습니다.

이 패턴에서는 AWS SCT를 사용하여 확장 기능 `aws_oracle_ext` 및 `orafce`를 사용해 Oracle 코드를 PostgreSQL로 변환 및 마이그레이션하고 확장 코드를 `psql` 기본 또는 네이티브 내장 코드로 수동으로 마이그레이션합니다.

- [AWS SCT](#) 확장 팩은 객체를 대상 데이터베이스로 변환할 때 필요한 소스 데이터베이스의 함수를 에뮬레이션하는 애드온 모듈입니다. AWS SCT 확장 팩을 설치하려면 먼저 데이터베이스 스키마를 변환해야 합니다.

데이터베이스 또는 데이터 웨어하우스 스키마를 변환할 때 AWS SCT는 대상 데이터베이스에 추가 스키마를 추가합니다. 이 스키마는 변환된 스키마를 대상 데이터베이스에 쓸 때 필요한 소스 데이터베이스의 SQL 시스템 함수를 구현합니다. 이 추가 스키마를 확장 팩 스키마라고 합니다.

OLTP 데이터베이스의 확장 팩 스키마는 소스 데이터베이스에 따라 이름이 지정됩니다. Oracle 데이터베이스의 경우 확장 팩 스키마는 `AWS_ORACLE_EXT`입니다.

기타 도구

- [Orafce](#) – Orafce는 Oracle 호환 함수, 데이터 유형 및 패키지를 구현하는 모듈입니다. 누구나 사용할 수 있도록 Berkeley Source Distribution (BSD) 라이선스가 있는 오픈 소스 도구입니다. 이 orafce 모듈은 많은 Oracle 함수가 PostgreSQL에 구현되어 있기 때문에 Oracle에서 PostgreSQL로 마이그레이션하는 데 유용합니다.

코드

AWS SCT 확장 코드 사용을 방지하기 위해 일반적으로 사용되고 Oracle에서 PostgreSQL로 마이그레이션된 코드 목록은 첨부 문서를 참조하십시오.

에픽

Amazon RDS for Oracle 소스 데이터베이스 구성

작업	설명	필요한 기술
Oracle 데이터베이스 인스턴스를 생성합니다.	Amazon RDS 콘솔에서 Amazon RDS for Oracle 또는	일반 AWS, DBA

작업	설명	필요한 기술
	Aurora PostgreSQL-Compatible 데이터베이스 인스턴스를 생성합니다.	
보안 그룹 구성	인바운드 및 아웃바운드 보안 그룹 구성	일반 AWS
데이터베이스를 생성합니다.	필요한 사용자 및 스키마로 Oracle 데이터베이스를 생성합니다.	일반 AWS, DBA
객체를 생성합니다.	객체를 생성하고 스키마에 데이터를 삽입합니다.	DBA

Amazon RDS for PostgreSQL 대상 데이터베이스 구성

작업	설명	필요한 기술
PostgreSQL 데이터베이스 인스턴스를 생성합니다.	Amazon RDS 콘솔에서 Amazon RDS for PostgreSQL 또는 Amazon Aurora PostgreSQL 데이터베이스 인스턴스를 생성합니다.	일반 AWS, DBA
보안 그룹 구성	인바운드 및 아웃바운드 보안 그룹 구성	일반 AWS
데이터베이스를 생성합니다.	필요한 사용자 및 스키마를 사용하여 PostgreSQL 데이터베이스를 생성합니다.	일반 AWS, DBA
확장 기능을 확인합니다.	PostgreSQL 데이터베이스에 <code>aws_oracle_ext</code> 및 <code>orafce</code> 가 올바르게 설치 및 구성되어 있는지 확인합니다.	DBA

작업	설명	필요한 기술
PostgreSQL 데이터베이스를 사용할 수 있는지 확인합니다.	PostgreSQL 데이터베이스가 가동되고 실행 중인지 확인합니다.	DBA

AWS SCT 및 확장 기능을 사용하여 Oracle 스키마를 PostgreSQL로 마이그레이션

작업	설명	필요한 기술
AWS SCT를 설치합니다.	AWS SCT의 최신 버전을 설치합니다.	DBA
AWS SCT를 구성합니다.	Oracle(ojdbc8.jar) 및 PostgreSQL(postgresql-42.2.5.jar)용 자바 데이터베이스 연결 (JDBC) 드라이버를 사용하여 AWS SCT를 구성합니다.	DBA
AWS SCT 확장 팩 또는 템플릿을 활성화합니다.	AWS SCT 프로젝트 설정에서 Oracle 데이터베이스 스키마의 aws_oracle_ext 및 orafce 확장 기능을 사용하여 내장된 함수 구현을 활성화합니다.	DBA
스키마를 변환합니다.	AWS SCT에서 스키마 변환을 선택하여 스키마를 Oracle에서 PostgreSQL로 변환하고 .sql 파일을 생성합니다.	DBA

AWS SCT 확장 코드를 psql 코드로 변환

작업	설명	필요한 기술
코드를 수동으로 변환합니다.	첨부 문서에 설명된 대로 확장 지원 코드의 각 줄을 psql 기본 내장 코드로 수동으로 변환합니다. 예를 들면 AWS_ORACLE_EXT.SYSDATE() 또는 ORACLE.SYSDATE() 를 NOW()로 변경합니다.	DBA
코드 유효성 검사	(선택 사항) PostgreSQL 데이터베이스에서 임시로 실행하여 각 코드 줄을 검증합니다.	DBA
PostgreSQL 데이터베이스에서 객체를 생성합니다.	PostgreSQL 데이터베이스에서 객체를 생성하려면 AWS SCT에서 생성되고 이전 두 단계에서 수정된 .sql 파일을 실행합니다.	DBA

관련 리소스

- 데이터베이스
 - [Amazon RDS의 Oracle](#)
 - [Amazon RDS의 PostgreSQL](#)
 - [Amazon Aurora PostgreSQL 작업](#)
 - [PostgreSQL EXPLAIN 계획](#)
- AWS SCT
 - [AWS Schema Conversion Tool Overview](#)
 - [AWS SCT 사용 설명서](#)
 - [AWS SCT 사용자 인터페이스 사용](#)
 - [Oracle Database를 AWS SCT의 소스로 사용](#)
- AWS SCT용 확장 프로그램

- [AWS SCT 확장 팩 사용](#)
- [Oracle 기능\(영어\)](#)
- [PGN orafce](#)
- [GitHub 오라클은](#)

추가 정보

자세한 내용은 첨부 문서의 코드를 수동으로 변환하기 위한 세부 명령어(구문 및 예제 포함)를 따르십시오.

첨부

이 문서와 관련된 추가 콘텐츠에 액세스하려면 [attachment.zip](#) 파일의 압축을 풉니다.

AWS DMS를 사용하여 Db2 데이터베이스를 Amazon EC2에서 Aurora MySQL과 호환되는 Aurora로 마이그레이션

작성자: Pinesh Singal(AWS)

환경: PoC 또는 파일럿	소스: Amazon EC2 기반 IBM Db2	대상: Amazon Aurora MySQL-Compatible Edition
R 유형: 리아키텍트	워크로드: IBM	기술: 마이그레이션, 데이터베이스
AWS 서비스: AWS DMS, Amazon EC2, AWS SCT, Amazon Aurora		

요약

[IBM Db2 for LUW 데이터베이스](#)를 [Amazon Elastic Compute Cloud\(Amazon EC2\)](#)로 마이그레이션한 후에는 Amazon Web Services(AWS) 클라우드 네이티브 데이터베이스로 이동하여 데이터베이스를 재설계하는 것을 고려해 보세요. 이 패턴은 [Amazon EC2](#) 인스턴스에서 실행되는 LUW용 IBM [Db2](#) 데이터베이스를 AWS의 [Amazon Aurora MySQL-Compatible Edition](#) 데이터베이스로 마이그레이션하는 것을 다룹니다.

이 패턴은 트랜잭션 수가 많은 수 테라바이트급 Db2 소스 데이터베이스의 가동 중지 시간을 최소화하면서 온라인 마이그레이션 전략을 설명합니다.

이 패턴은 [AWS Schema Conversion Tool\(AWS SCT\)](#)을 사용하여 Db2 데이터베이스 스키마를 Aurora MySQL Compatible 스키마로 변환합니다. 그런 다음 패턴은 [AAWS Database Migration Service\(AWS DMS\)](#)를 사용하여 Db2 데이터베이스의 데이터를 Aurora MySQL Compatible 데이터베이스의 데이터로 마이그레이션합니다. AWS SCT에서 변환하지 않는 코드인 경우 수동 변환이 필요합니다.

사전 조건 및 제한 사항

사전 조건

- Virtual Private Cloud(VPC)를 사용하는 활성 AWS 계정
- AWS SCT

• DMS

제품 버전

- AWS SCT 최신 버전
- Linux용 Db2 버전 11.1.4.4 이상

아키텍처

소스 기술 스택

- EC2 인스턴스에 마운트된 DB2/Linux x86-64비트

대상 기술 스택

- An Amazon Aurora MySQL-Compatible Edition 데이터베이스 인스턴스

소스 및 대상 아키텍처

다음 다이어그램은 소스 Db2와 대상 Aurora MySQL 호환 데이터베이스 간의 데이터 마이그레이션 아키텍처를 보여줍니다. AWS 클라우드의 아키텍처에는 Virtual Private Cloud(VPC)(가상 사설 클라우드), 가용 영역, Db2 인스턴스 및 AWS DMS 복제 인스턴스를 위한 퍼블릭 서브넷, Aurora MySQL 호환 데이터베이스를 위한 프라이빗 서브넷이 포함됩니다.

도구

서비스

- [Amazon Aurora](#)는 클라우드용으로 구축되었으며 MySQL 및 PostgreSQL과 호환되는 완전 관리형 관계형 데이터베이스 엔진입니다.
- [AWS Database Migration Service\(AWS DMS\)](#)를 사용하면 데이터 스토어를 AWS 클라우드로 마이그레이션하거나 클라우드와 온프레미스 설정 조합 간에 마이그레이션할 수 있습니다.
- [Amazon Elastic Compute Cloud\(Amazon EC2\)](#)는 AWS 클라우드에서 규모를 조정할 수 있는 컴퓨팅 용량을 제공합니다. 필요한 만큼 가상 서버를 시작하고 빠르게 스케일 업하거나 스케일 다운할 수 있습니다.

- [AWS Schema Conversion Tool\(AWS SCT\)](#)은 소스 데이터베이스 스키마와 대부분의 사용자 지정 코드를 대상 데이터베이스와 호환되는 형식으로 자동 변환하여 이기종 데이터베이스 마이그레이션을 지원합니다. AWS SCT는 LUW 버전 9.1, 9.5, 9.7, 10.1, 10.5, 11.1 및 11.5용 소스 IBM Db2를 지원합니다.

모범 사례

모범 사례는 [AWS Database Migration Service의 모범 사례](#)를 참조하세요.

에픽

소스 IBM Db2 데이터베이스 구성

작업	설명	필요한 기술
Amazon EC2에 IBM Db2 데이터베이스를 생성합니다.	<p>AWS Marketplace의 Amazon Machine Image(AMI)를 사용하거나 EC2 인스턴스에 Db2 소프트웨어를 설치하여 EC2 인스턴스에 IBM Db2 데이터베이스를 생성할 수 있습니다.</p> <p>온프레미스 데이터베이스와 유사한 IBM Db2용 AMI(예: IBM Db2 v11.5.7 RHEL 7.9)를 선택하여 EC2 인스턴스를 실행합니다.</p>	DBA, 일반 AWS
보안 그룹을 구성합니다.	포트 22와 50000을 사용하여 SSH(보안 셸) 및 TCP에 대한 VPC 보안 그룹 인바운드 규칙을 각각 구성합니다.	일반 AWS
데이터베이스 인스턴스를 생성합니다.	새 인스턴스(사용자) 및 데이터베이스(스키마)를 생성하거나 기본 db2inst1 인스턴스 및 샘플 데이터베이스를 사용합니다.	DBA

작업	설명	필요한 기술
	<ol style="list-style-type: none"> 1. 터미널을 사용하여 Db2 데이터베이스에 연결하여 EC2 인스턴스에 연결합니다. 또는 Db2 데이터베이스에 연결할 모든 DB 클라이언트 소프트웨어를 설치할 수 있습니다. 2. db2inst1 사용자의 비밀번호를 설정하려면 <code>sudo passwd db2inst1</code> 명령을 실행하세요. 3. db2inst1 인스턴스에 연결하려면 <code>sudo su - db2inst1</code> 명령을 실행하세요. 4. Db2 데이터베이스에 연결하려면 <code>db2</code> 명령을 실행하세요. 5. 샘플 데이터베이스에 연결하려면 <code>connect to sample</code> 명령을 사용하세요. 또는 생성한 데이터베이스에 연결할 수도 있습니다. 6. 데이터베이스 인스턴스에 연결한 후 Db2 SQL 문을 사용하여 개체를 생성하고 이러한 개체에 데이터를 삽입합니다. 	
<p>Db2 DB 인스턴스를 사용할 수 있는지 확인합니다.</p>	<p>Db2 데이터베이스 인스턴스가 작동 및 실행 중인지 확인하려면 <code>Db2pd -</code> 명령을 사용하세요.</p>	<p>DBA</p>

대상 Aurora MySQL과 호환되는 데이터베이스 구성

작업	설명	필요한 기술
Aurora MySQL과 호환되는 데이터베이스를 생성합니다.	<p>AWS RDS 서비스에서 MySQL 호환 데이터베이스를 갖춘 Amazon Aurora 생성</p> <ul style="list-style-type: none"> MySQL 호환 버전 및 원하는 버전으로 Amazon Aurora에서 데이터베이스를 생성합니다(예: Aurora (MySQL)-5.6.10a) MySQL Workbench 애플리케이션 또는 MySQL 데이터베이스에 연결할 수 있는 선호하는 DB 클라이언트 소프트웨어를 설치합니다. 	DBA, 일반 AWS
보안 그룹을 구성합니다.	SSH 및 TCP 연결에 대한 VPC 보안 그룹 인바운드 규칙을 구성합니다.	일반 AWS
Aurora 데이터베이스를 사용할 수 있는지 확인합니다.	<p>Aurora MySQL과 호환되는 데이터베이스의 작동 및 실행 상태를 확인하려면 다음을 수행합니다.</p> <ol style="list-style-type: none"> SSH를 통해 EC2 인스턴스에 연결합니다. MySQL Workbench에서 Aurora MySQL 호환 인스턴스를 구성하고 해당 인스턴스에 연결합니다. 다음 예와 같이 엔드포인트를 호스트 이름으로 사용합니다. 	DBA

작업	설명	필요한 기술
	<pre>mysql-cluster-instance-1.cokmvis0v46q.us-east-1.rds.amazonaws.com</pre> <p>3. 새 스키마(예:mysql-sample-db2)를 생성하고 연결합니다.</p> <p>4. MySQL 문을 실행하여 데이터베이스의 스키마와 객체를 확인합니다.</p>	

AWS SCT 구성 및 실행

작업	설명	필요한 기술
AWS SCT를 설치합니다.	최신 버전의 AWS SCT (최신 버전 1.0.628)를 다운로드하고 설치합니다.	일반 AWS
AWS SCT를 구성합니다.	<ol style="list-style-type: none"> 1. IBM Db2(4.22.X 버전) 및 MySQL(8.x)용 Java 데이터베이스 연결(JDBC) 드라이버를 다운로드하세요. 2. AWS SCT에서 드라이버를 구성하려면 설정, 글로벌 설정, 드라이버를 선택합니다. 	일반 AWS
AWS SCT 프로젝트를 생성합니다.	LUW용 Db2를 소스 DB 엔진으로 사용하고 Aurora MySQL과 호환되는 대상 DB 엔진으로 사용하는 AWS SCT 프로젝트 및 보고서를 생성합니다.	일반 AWS

작업	설명	필요한 기술
	Db2 for LUW 데이터베이스에 연결하는 데 필요한 권한을 확인하려면 Db2 LUW를 AWS SCT의 소스로 사용 을 참조하세요.	

작업	설명	필요한 기술
<p>객체를 검증합니다.</p>	<p>로드 스키마를 선택하고 객체의 유효성을 검사합니다. 대상 데이터베이스에서 잘못된 객체를 업데이트합니다.</p> <ol style="list-style-type: none"> 1. 연결 세부 정보를 제공하여 Amazon Aurora MySQL 호환 서버에 연결하고 연결 테스트를 선택합니다. <p>소스 및 대상 연결이 모두 성공해야 AWS SCT에서 마이그레이션 보고를 시작할 수 있습니다.</p> <ol style="list-style-type: none"> 2. 보고서가 완성되면 변환할 스키마를 입력하고 마침을 선택합니다. <p>AWS SCT는 변환되고 오류가 있는 모든 소스 및 대상 객체를 나열합니다.</p> <ol style="list-style-type: none"> 3. 오류를 검토하고 수동으로 삭제합니다. 4. 모든 오류를 해결한 후 스키마의 컨텍스트 메뉴(마우스 오른쪽 버튼 클릭)를 열고 스키마 로드를 선택합니다. 5. 데이터베이스에 적용을 선택합니다. 6. MySQL Workbench에서 Aurora MySQL 호환 데이터베이스에 연결하고 스키마와 객체를 확인합니다. 	<p>DBA, 일반 AWS</p>

AWS DMS 구성 및 실행

작업	설명	필요한 기술
복제 인스턴스를 생성합니다.	AWS Management Console에 로그인하고, AWS DMS 서비스로 이동한 다음, 소스 및 타겟 데이터베이스에 대해 구성된 VPC 보안 그룹에 대한 유효한 설정을 사용하여 복제 인스턴스를 생성합니다.	일반 AWS
엔드포인트를 생성합니다.	<p>Db2 데이터베이스의 소스 엔드포인트를 생성하고 Aurora MySQL 호환 데이터베이스의 대상 엔드포인트를 생성합니다.</p> <ol style="list-style-type: none"> 1. RDS DB 인스턴스 선택을 선택한 다음 생성한 Db2 인스턴스를 선택하여 원본으로 사용할 IBM Db2용 엔드포인트를 생성합니다. 엔드포인트 구성 세부 정보가 자동으로 채워집니다. 2. 엔드포인트별 설정에서 다음과 같은 추가 연결 속성을 추가합니다. <pre data-bbox="630 1465 1029 1663">CurrentLSN=<scan>; MaxKBytesPerRead=64; SetDataCaptureChanges=true</pre> <p>이러한 속성을 언급하지 않으면 소스 엔드포인트 테스트 연결이 성공하지 못합니다. 자세한 내용은 IBM Db2</p>	일반 AWS

작업	설명	필요한 기술
	<p>LUW를 AWS DMS의 소스로 사용하기를 참조하세요.</p> <p>3. RDS DB 인스턴스 선택을 선택한 다음, 생성한 Aurora MySQL 호환 인스턴스를 선택하여 Aurora MySQL과 호환되는 엔드포인트를 대상으로 생성합니다. 엔드포인트 구성 세부 정보가 자동으로 채워집니다. 자세한 내용은 MySQL 호환 데이터베이스를 AWS Database Migration Service의 대상으로 사용을 참조하세요.</p> <p>4. 원본 및 대상 엔드포인트를 테스트합니다. 둘 다 성공적이고 사용 가능한지 확인합니다.</p> <p>5. 테스트가 실패할 경우 보안 그룹 인바운드 규칙이 유효한지 확인하세요.</p>	

작업	설명	필요한 기술
<p>마이그레이션 작업을 생성합니다.</p>	<p>전체 로드 및 CDC 또는 데이터 검증을 위해 단일 마이그레이션 작업 또는 여러 마이그레이션 작업을 생성합니다.</p> <ol style="list-style-type: none"> 1. 데이터베이스 마이그레이션 작업을 생성하려면 복제 인스턴스, 소스 데이터베이스 엔드포인트, 대상 데이터베이스 엔드포인트를 선택합니다. 마이그레이션 유형을 기존 데이터 마이그레이션(전체 로드), 데이터 변경 사항만 복제(CDC)하거나 기존 데이터를 마이그레이션하고 진행 중인 변경 사항 복제(전체 로드 및 CDC)하기로 지정합니다. 2. 테이블 매핑에서 선택 규칙 및 변환 규칙을 GUI 또는 JSON 형식으로 구성할 수 있습니다. 3. 선택 규칙에서 스키마를 선택하고 테이블 이름을 입력한 다음 구성할 작업(예: 스키마: 샘플, 테이블 이름: %, 작업: 포함)을 선택합니다. 4. 변환 규칙에서 대상(스키마, 테이블 또는 열)을 선택합니다. 스키마 이름을 선택하고 작업(대소문자, 접두사, 접미사)을 선택합니다(예: 대상: 스키마, mysql-sample-db, 작업: 소문자 만들기). 	<p>일반 AWS</p>

작업	설명	필요한 기술
	5. Amazon CloudWatch 로그 모니터링을 활성화합니다.	
프로덕션 실행을 계획합니다.	애플리케이션 소유자 등 이해 관계자와 함께 다운타임을 확인하여 프로덕션 시스템에서 AWS DMS를 실행합니다.	마이그레이션 책임자
마이그레이션 작업을 실행합니다.	<ol style="list-style-type: none"> 1. 준비 상태인 AWS DMS 작업을 시작합니다. 2. Amazon CloudWatch Logs의 마이그레이션 작업 로그를 모니터링하여 오류가 있는지 확인하십시오. 	일반 AWS
데이터를 검증합니다.	<p>원본 Db2 및 대상 MySQL 데이터베이스의 마이그레이션 작업 결과 및 데이터를 검토합니다.</p> <ol style="list-style-type: none"> 1. 상태가 전체 진행 중인 복제 로드인 경우 CDC 데이터 마이그레이션을 통한 전체 로드가 완료되고 검증이 진행 중입니다. 2. Aurora MySQL 호환 데이터베이스에 연결하고 데이터를 확인합니다. 3. Db2 데이터베이스에 데이터를 삽입하거나 업데이트하여 진행 중인 변경 사항을 확인합니다. 	DBA
마이그레이션 작업을 중지합니다.	데이터 검증이 성공적으로 완료되면 검증 마이그레이션 작업을 중지하세요.	일반 AWS

문제 해결

문제	Solution
AWS SCT 소스 및 대상 테스트 연결이 실패합니다.	들어오는 트래픽을 수락하도록 JDBC 드라이버 버전과 VPC 보안 그룹 인바운드 규칙을 구성합니다.
Db2 소스 엔드포인트 테스트 실행이 실패했습니다.	추가 연결 설정 <code>CurrentLSN=<scan>;</code> 을 구성합니다.
<p>AWSDMS 작업이 Db2 소스에 연결되지 않고 다음 오류가 반환됩니다.</p> <pre>database is recoverable if either or both of the database configura tion parameters LOGARCHMETH1 and LOGARCHMETH2 are set to ON</pre>	<p>오류를 방지하려면 다음 명령을 실행하세요.</p> <ol style="list-style-type: none"> 1. <code>\$ db2 update db cfg for sample using LOGARCHMETH1 DISK:/home/db2inst1/logs</code> 2. <code>\$ db2stop</code> 3. <code>\$ db2start</code> 4. <code>\$ db2 connect to sample</code> <div data-bbox="867 1066 1507 1264" style="border: 1px solid #ccc; border-radius: 10px; padding: 10px; margin: 10px 0;"> <p>SQL1116N A connection to or activation of database "SAMPLE" cannot be made because of BACKUP PENDING. SQLSTATE=57019</p> </div> <ol style="list-style-type: none"> 5. <code>\$ db2 backup database sample to ../logs</code> <div data-bbox="867 1402 1507 1520" style="border: 1px solid #ccc; border-radius: 10px; padding: 10px; margin: 10px 0;"> <p>SQL2036N The path for the file or device "../logs" is not valid</p> </div> <ol style="list-style-type: none"> 6. <code>\$ cd</code> 7. <code>\$ pwd</code> <div data-bbox="867 1667 1507 1747" style="border: 1px solid #ccc; border-radius: 10px; padding: 10px; margin: 10px 0;"> <p>/home/db2inst1</p> </div> <ol style="list-style-type: none"> 8. <code>\$ mkdir /tmp/backup</code>

문제	Solution
	<pre>9. \$ db2 backup database sample to / tmp/backup</pre> <div data-bbox="867 331 1507 491" style="border: 1px solid #ccc; border-radius: 10px; padding: 10px; margin: 10px 0;"> <pre>Backup successful. The timestamp for this backup image is : 201905300 84921</pre> </div> <pre>10\$ db2 connect to sample</pre> <div data-bbox="867 579 1507 814" style="border: 1px solid #ccc; border-radius: 10px; padding: 10px; margin: 10px 0;"> <pre>Database Connection Information Database server = DB2/LINUX 9.7.1 SQL authorization ID = DB2INST1 Local database alias = SAMPLE</pre> </div>

관련 리소스

Amazon EC2

- [Amazon EC2](#)
- [Amazon EC2 사용 설명서](#)

데이터베이스

- [IBM Datab2 데이터베이스](#)
- [Amazon Aurora](#)
- [Amazon Aurora MySQL을 사용한 작업](#)

AWS SCT

- [AWS DMS 스키마 전환](#)
- [AWS Schema Conversion Tool 사용 설명서](#)
- [AWS SCT 사용자 인터페이스 사용](#)
- [IBM Db2 LUW를 AWS SCT에 대한 소스로 사용](#)

DMS

- [AWS Database Migration Service](#)
- [AWS Database Migration Service 사용 설명서](#)
- [데이터 마이그레이션용 소스](#)
- [마이그레이션에 적합한 대상](#)
- [AWS Database Migration Service 및 AWS Schema Conversion Tool은 이제 IBM Db2 LUW를 소스로 지원합니다\(블로그 게시물\)](#)
- [관계형 데이터베이스를 실행하는 애플리케이션을 AWS로 마이그레이션](#)

AWS DMS를 사용하여 Microsoft SQL 서버 데이터베이스를 Amazon EC2에서 Amazon DocumentDB로 마이그레이션

작성자: Uma Maheswara Rao Nooka(AWS)

소스: Amazon EC2의 Microsoft SQL Server	대상: Amazon DocumentDB	R 유형: 리아키텍트
환경: PoC 또는 파일럿	기술: 클라우드 네이티브, 데이터베이스, 마이그레이션	워크로드: Microsoft
AWS 서비스: Amazon EC2, Amazon DocumentDB		

요약

이 패턴은 AWS Database Migration Service(AWS DMS)를 사용하여 Amazon Elastic Compute Cloud(Amazon EC2) 인스턴스에 호스팅된 Microsoft SQL Server 데이터베이스를 Amazon DocumentDB(MongoDB 호환) 데이터베이스로 마이그레이션하는 방법을 설명합니다.

AWS DMS 복제 작업은 SQL Server 데이터베이스의 테이블 구조를 읽고, Amazon DocumentDB에 해당 컬렉션을 생성하고, 전체 로드 마이그레이션을 수행합니다.

또한 이 패턴을 사용하여 온프레미스 SQL 서버 또는 SQL Server용 Amazon Relational Database Service(RDS) DB 인스턴스를 Amazon DocumentDB로 마이그레이션할 수 있습니다. 자세한 내용은 AWS 권장 가이드 웹 사이트에서 [Microsoft SQL Server 데이터베이스를 AWS 클라우드로 마이그레이션](#) 가이드를 참조하세요.

사전 조건 및 제한 사항

사전 조건

- 활성 상태의 AWS 계정.
- EC2 인스턴스의 기존 SQL 서버 데이터베이스.
- SQL Server 데이터베이스의 AWS DMS에 할당된 고정 데이터베이스(db_owner) 역할. 자세한 정보는 SQL Server 문서의 [데이터베이스 수준 역할](#)을 참조하세요.

- [Amazon DocumentDB 클러스터 내부 및 외부로 데이터를 이동](#)하려면 mongodump, mongorestore, mongoexport, mongoimport 유틸리티를 사용하는 방법을 잘 알고 있어야 합니다.
- [Microsoft SQL Server Management Studio](#), 설치 및 구성됨.

제한 사항

- Amazon DocumentDB의 클러스터 크기 제한은 64TB입니다. 자세한 내용은 Amazon DocumentDB 설명서의 [클러스터 제한](#)을 참조하세요.
- AWS DMS는 여러 소스 테이블을 단일 Amazon DocumentDB 모음에 병합하는 기능을 지원하지 않습니다.
- AWS DMS가 프라이머리 키 없이 원본 테이블의 변경 사항을 처리하는 경우 소스 테이블의 대형 객체(LOB) 열을 무시할 수 있습니다.

아키텍처

소스 기술 스택

- Amazon EC2

대상 아키텍처

대상 기술 스택

- Amazon DocumentDB

도구

- [AWS DMS](#) – AWS Database Migration Service(AWS DMS)를 사용하여 데이터베이스를 쉽고 안전하게 마이그레이션할 수 있습니다.
- [Amazon DocumentDB](#) – Amazon DocumentDB(MongoDB 호환)는 빠르고 안정적이며 완전하게 관리되는 데이터베이스 서비스입니다.
- [Amazon EC2](#) – Amazon Elastic Compute Cloud(Amazon EC2)는 AWS 클라우드에서 확장 가능한 컴퓨팅 용량을 제공합니다.

- [Microsoft SQL 서버](#) - SQL 서버는 관계형 데이터베이스 관리 시스템입니다.
- [SQL 서버 관리 스튜디오\(SSMS\)](#) - SSMS는 SQL 서버 구성 요소에 대한 액세스, 구성 및 관리를 포함하여 SQL Server를 관리하는 도구입니다.

에픽

VPC 생성 및 구성

작업	설명	필요한 기술
VPC를 생성합니다.	AWS Management Console에 로그인한 후 Amazon VPC 콘솔을 엽니다. IPv4 CIDR 블록 범위를 사용하여 Virtual Private Cloud(VPC)를 생성합니다.	시스템 관리자
보안 그룹 및 네트워크 ACL을 생성합니다.	Amazon VPC 콘솔에서 요구 사항에 따라 VPC의 보안 그룹 및 네트워크 액세스 제어 목록(네트워크 ACL)을 생성합니다. 이러한 구성의 기본 설정을 사용할 수도 있습니다. 이 스토리와 다른 스토리에 대한 자세한 내용은 '관련 리소스' 섹션을 참조하세요.	시스템 관리자

Amazon DocumentDB 클러스터 생성 및 구성

작업	설명	필요한 기술
Amazon DocumentDB 클러스터를 생성합니다.	Amazon MSK 콘솔을 열고 '클러스터'를 선택합니다. '생성'을 선택하고 인스턴스 하나가 포함된 Amazon DocumentDB 클러스터를 생성합니다. 중요:	시스템 관리자

작업	설명	필요한 기술
	VPC의 보안 그룹으로 이 클러스터를 구성해야 합니다.	
mongo 셸을 설치합니다.	mongo 셸은 Amazon DocumentDB 클러스터에 연결하고 쿼리하는 데 사용하는 명령줄 유틸리티입니다. 설치하려면 <code>/etc/yum.repos.d/mongodb-org-3.6.repo</code> 명령을 실행하여 리포지토리 파일을 생성하세요. <code>sudo yum install -y mongodb-org-shell</code> 명령을 실행하여 몽고 셸을 설치합니다. 전송 중 데이터를 암호화하려면 Amazon DocumentDB용 퍼블릭 키를 다운로드한 다음 Amazon DocumentDB 인스턴스에 연결합니다. 이 스토리와 다른 스토리에 대한 자세한 내용은 '관련 리소스' 섹션을 참조하세요.	시스템 관리자
Amazon DocumentDB 클러스터에 데이터베이스 생성합니다.	데이터베이스 이름과 함께 <code>'use'</code> 명령을 실행하여 Amazon DocumentDB 클러스터에 데이터베이스를 생성합니다.	시스템 관리자

AWS DMS 복제 인스턴스를 생성하고 구성

작업	설명	필요한 기술
AWS DMS 복제 인스턴스를 생성합니다.	AWS DMS 콘솔을 열고 '복제 인스턴스 생성'을 선택합니다. 복제 작업에 대한 이름과 설명을 입력합니다. 인스턴스 클래	시스템 관리자

작업	설명	필요한 기술
	스, 엔진 버전, 스토리지, VPC, 다중 AZ를 선택하고 공개적으로 액세스할 수 있도록 설정합니다. '고급' 탭을 선택하여 네트워크와 암호화 설정을 설정합니다. 유지 관리 설정을 지정한 다음 '복제 인스턴스 생성'을 선택합니다.	
SQL Server 데이터베이스를 구성합니다.	Microsoft SQL Server에 로그인하고 원본 엔드포인트와 AWS DMS 복제 인스턴스 간의 통신을 위한 인바운드 규칙을 추가합니다. 복제 인스턴스의 프라이빗 IP 주소를 소스로 사용합니다. 중요: 복제 인스턴스와 대상 엔드포인트는 동일한 VPC에 있어야 합니다. 소스 인스턴스와 복제 인스턴스의 VPC가 다른 경우 보안 그룹의 대체 소스를 사용하세요.	시스템 관리자

AWS DMS에서 원본 및 대상 엔드포인트를 생성하고 테스트

작업	설명	필요한 기술
소스와 대상 데이터베이스 엔드포인트를 생성합니다.	AWS DMS 콘솔을 열고 '소스 및 대상 데이터베이스 엔드포인트 연결'을 선택합니다. 원본 데이터베이스 및 대상 데이터베이스에 대한 연결 정보를 지정합니다. 필요한 경우 "고급" 탭을 선택하여 "추가 연결 속성"의 값을 설정합니다. 엔드포	시스템 관리자

작업	설명	필요한 기술
	인트 구성에서 인증서 번들을 다운로드하여 사용하세요.	
엔드포인트 연결을 테스트합니다.	연결을 테스트하려면 '테스트 실행'을 선택합니다. 보안 그룹 설정과 소스 및 대상 데이터베이스 인스턴스 모두에서 AWS DMS 복제 인스턴스로의 연결을 확인하여 오류 메시지 문제를 해결합니다.	시스템 관리자

데이터 마이그레이션

작업	설명	필요한 기술
AWS DMS 마이그레이션 작업을 생성합니다.	AWS DMS 콘솔에서 '작업', '작업 생성'을 선택합니다. 소스 및 대상 엔드포인트 이름, 복제 인스턴스 이름을 비롯한 작업 옵션을 지정합니다. '마이그레이션 유형'에서 '기존 데이터 마이그레이션'과 '데이터 변경 사항만 복제'를 선택합니다. '작업 시작'을 선택합니다.	시스템 관리자
AWS DMS 마이그레이션 작업을 실행합니다.	'작업 설정'에서 '아무것도 하지 않음', '대상에 테이블 삭제', '잘라내기', "복제에 LOB 열 포함"과 같은 테이블 준비 모드의 설정을 지정합니다. AWS DMS에서 허용할 최대 LOB 크기를 설정하고 '로깅 활성화'를 선택합니다. '고급 설정'을 기본값으로 두고 '작업 생성'을 선택합니다.	시스템 관리자

작업	설명	필요한 기술
마이그레이션을 모니터링합니다.	AWS DMS 콘솔에서 '작업'을 선택하고 마이그레이션 작업을 선택합니다. '작업 모니터링'을 선택하여 작업을 모니터링하세요. 전체 로드 마이그레이션이 완료되고 캐시된 변경 사항이 적용되면 작업이 중지됩니다.	시스템 관리자

마이그레이션 테스트 및 확인

작업	설명	필요한 기술
mongo 셸을 사용하여 Amazon DocumentDB 클러스터에 연결합니다.	Amazon DocumentDB 콘솔을 열고 '클러스터'에서 클러스터를 선택합니다. 연결 & 보안) 탭의 mongo 셸을 사용하여 이 클러스터에 연결에서 복사를 선택합니다.	시스템 관리자
마이그레이션 결과를 확인하세요.	데이터베이스 이름과 함께 'use' 명령을 실행한 다음 'show collections' 명령을 실행합니다. 데이터베이스 이름과 함께 'db. .count();' 명령을 실행합니다. 결과가 소스 데이터베이스와 일치하면 마이그레이션이 성공한 것입니다.	시스템 관리자

관련 리소스

VPC 생성 및 구성

- [VPC의 보안 그룹을 생성](#)
- [네트워크 ACL을 생성](#)

Amazon DocumentDB 클러스터 생성 및 구성

- [Amazon DocumentDB 클러스터 생성](#)
- [Amazon DocumentDB용 mongo 셸 설치](#)
- [Amazon DocumentDB 클러스터에 연결](#)

AWS DMS 복제 인스턴스를 생성하고 구성

- [퍼블릭 및 프라이빗 복제 인스턴스를 사용](#)

AWS DMS에서 소스 및 대상 엔드포인트 생성 및 테스트

- [Amazon DocumentDB를 AWS DMS의 대상으로 사용](#)
- [SQL Server 데이터베이스를 AWS DMS를 위한 소스로 사용](#)
- [AWS DMS 엔드포인트 사용](#)

데이터 마이그레이션

- [Amazon DocumentDB로 마이그레이션](#)

기타 리소스

- [AWS DMS용 소스로 SQL Server 사용 시 적용되는 제한 사항](#)
- [Amazon DocumentDB를 사용하여 대규모로 애플리케이션을 구축하고 관리하는 방법](#)

온프레미스 ThoughtSpot 팔콘 데이터베이스를 Amazon Redshift로 마이그레이션하기

바툴가 푸레브라차(AWS), 안토니 프라사드 테바라즈(AWS) 제작

환경: PoC 또는 파일럿	출처: 온프레미스 팔콘 데이터베이스 ThoughtSpot	대상: Amazon Redshift
R 유형: 리아키텍트	워크로드: 기타 모든 워크로드	기술: 마이그레이션, 데이터베이스
AWS 서비스: AWS DMS; Amazon Redshift		

요약

온프레미스 데이터 웨어하우스에는 특히 대규모 데이터 세트의 경우 상당한 관리 시간과 리소스가 필요합니다. 이러한 웨어하우스를 구축, 유지 및 확장하는 데 드는 재정적 비용도 매우 높습니다. 비용을 관리하고 ETL(추출, 변환, 로드) 복잡성을 낮게 유지하고, 데이터 증가에 따른 성능을 제공하려면 로드할 데이터와 보관할 데이터를 지속적으로 선택해야 합니다.

온프레미스 [ThoughtSpot Falcon 데이터베이스](#)를 Amazon Web Services (AWS) Cloud로 마이그레이션하면 전체 인프라 비용을 절감하는 것 외에도 비즈니스 민첩성, 보안 및 애플리케이션 안정성을 높이는 클라우드 기반 데이터 레이크 및 데이터 웨어하우스에 액세스할 수 있습니다. Amazon Redshift를 사용하면 데이터 웨어하우스의 비용과 운영 오버헤드를 크게 줄일 수 있습니다. 또한, Amazon Redshift Spectrum을 사용하여 데이터를 로드하지 않고도 네이티브 형식으로 대량의 데이터를 분석할 수 있습니다.

이 패턴은 ThoughtSpot Falcon 데이터베이스를 온프레미스 데이터 센터에서 AWS 클라우드의 Amazon Redshift 데이터베이스로 마이그레이션하는 단계와 프로세스를 설명합니다.

사전 조건 및 제한 사항

사전 조건

- 활성 상태의 AWS 계정
- 온프레미스 데이터 센터에서 호스팅되는 ThoughtSpot Falcon 데이터베이스입니다.

제품 버전

- ThoughtSpot 버전 7.0.1

아키텍처

이 다이어그램은 다음 워크플로를 보여줍니다.

1. 데이터는 온프레미스 관계형 데이터베이스에서 호스팅됩니다.
2. AWS Schema Conversion Tool(AWS SCT)는 Amazon Redshift와 호환되는 데이터 정의 언어(DDL)를 변환합니다.
3. 테이블이 생성된 후 AWS Database Migration Service(AWS DMS)를 사용하여 데이터를 마이그레이션할 수 있습니다.
4. 데이터는 Amazon Redshift에 로드됩니다.
5. Redshift Spectrum을 사용하거나 Amazon S3에서 데이터를 이미 호스팅하고 있는 경우 데이터는 Amazon Simple Storage Service(S3)에 저장됩니다.

도구

- [AWS DMS](#) - AWS Database Migration Service(AWS DMS)를 사용하면 데이터베이스를 빠르고 안전하게 마이그레이션할 수 있습니다.
- [Amazon Redshift](#) - Amazon Redshift는 속도가 빠른 페타바이트 규모의 완전 관리형 데이터 웨어하우스 서비스로, 간편하고 비용 효율적으로 모든 데이터를 기존 비즈니스 인텔리전스 도구를 사용하여 효율적으로 분석할 수 있게 해줍니다.
- [AWS SCT](#) - AWS Schema Conversion Tool(AWS SCT)는 기존 데이터베이스 스키마를 한 데이터베이스 엔진에서 다른 데이터베이스 엔진으로 변환합니다.

에픽

마이그레이션 준비

작업	설명	필요한 기술
적절한 Amazon Redshift 구성을 파악합니다.	<p>요구 사항 및 데이터 볼륨을 기반으로 적절한 Amazon Redshift 클러스터 구성을 파악합니다.</p> <p>자세한 내용은 Amazon Redshift 설명서의 Amazon Redshift 클러스터를 참조하세요.</p>	DBA
Amazon Redshift를 조사하여 요구 사항을 충족하는지 평가합니다.	Amazon Redshift FAQ 를 사용하여 Amazon Redshift가 요구 사항을 충족하는지 이해하고 평가합니다.	DBA

대상 Amazon Redshift 클러스터 준비

작업	설명	필요한 기술
Amazon Redshift 클러스터를 생성합니다.	<p>AWS Management Console에 로그인하고, Amazon Redshift 콘솔을 연 다음, Virtual Private Cloud(VPC)에서 Amazon Redshift 클러스터를 생성합니다.</p> <p>자세한 내용은 Amazon Redshift 설명서의 VPC에서 클러스터 생성을 참조하세요.</p>	DBA

작업	설명	필요한 기술
<p>Amazon Redshift 데이터베이스 설계를 위한 PoC를 수행합니다.</p>	<p>데이터베이스 설계를 위한 개념 증명(PoC)을 수행하여 Amazon Redshift 모범 사례를 따릅니다.</p> <p>자세한 내용은 Amazon Redshift 설명서의 Amazon Redshift에 대한 개념 증명 수행하기를 참고하십시오.</p>	DBA
<p>데이터베이스 사용자를 생성합니다.</p>	<p>Amazon Redshift 데이터베이스에서 사용자를 생성하고 스키마와 테이블에 액세스할 수 있는 적절한 역할을 부여합니다.</p> <p>자세한 내용은 Amazon Redshift 설명서의 사용자 또는 사용자 그룹에 액세스 권한 부여를 참고하십시오.</p>	DBA
<p>대상 데이터베이스에 구성 설정을 적용합니다.</p>	<p>요구 사항에 따라 Amazon Redshift 데이터베이스에 구성 설정을 적용합니다.</p> <p>데이터베이스, 세션 및 서버 수준 파라미터를 활성화하는 방법에 대한 자세한 내용은 Amazon Redshift 설명서의 구성 참조를 참고하십시오.</p>	DBA

Amazon Redshift 클러스터에서 객체를 생성합니다.

작업	설명	필요한 기술
Amazon Redshift에서 DL을 사용하여 수동으로 테이블을 생성합니다.	(선택 사항) AWS SCT를 사용하는 경우, 테이블이 자동으로 생성됩니다. 하지만 DDL을 복제할 때 오류가 발생하는 경우 수동으로 테이블을 생성해야 합니다.	DBA
Redshift Spectrum용 외부 테이블을 생성합니다.	Amazon Redshift Spectrum의 외부 스키마를 사용하여 외부 테이블을 생성합니다. 외부 테이블을 생성하려면 외부 스키마의 소유자이거나 데이터베이스 수퍼유저 여야 합니다. 자세한 내용은 Amazon Redshift 설명서의 Amazon Redshift Spectrum용 외부 테이블 만들기 를 참고하십시오.	DBA

AWS DMS를 사용하여 데이터를 마이그레이션합니다

작업	설명	필요한 기술
AWS DMS를 사용하여 데이터를 마이그레이션합니다.	Amazon Redshift 데이터베이스에서 테이블의 DDL을 생성한 후, AWS DMS를 사용하여 데이터를 Amazon Redshift로 마이그레이션합니다. 자세한 단계 및 지침은 AWS DMS 설명서의 Amazon Redshift 데이터베이스를 AWS	DBA

작업	설명	필요한 기술
	DMS의 대상으로 사용하기 섹션을 참고하십시오.	
COPY 명령을 사용하여 데이터를 로드합니다.	Amazon Redshift COPY 명령을 사용하여 Amazon S3에서 Amazon Redshift로 데이터를 로드합니다. 자세한 내용은 Amazon Redshift 설명서의 COPY 명령을 사용하여 Amazon S3에서 로드하기 섹션을 참고하십시오.	DBA

Amazon Redshift 클러스터 검증

작업	설명	필요한 기술
원본 및 대상 데이터를 검증합니다.	소스 시스템에서 로드된 소스 및 대상 레코드의 테이블 수를 검증합니다.	DBA
Amazon Redshift 성능 튜닝 모범 사례 구현	Amazon Redshift 테이블 및 데이터베이스 설계 모범 사례 구현 자세한 내용은 블로그 게시물 Amazon Redshift의 10가지 성능 튜닝 기법 을 참고하십시오.	DBA
쿼리 성능을 최적화합니다.	Amazon Redshift는 SQL 기반 쿼리를 사용하여 시스템의 데이터 및 객체와 상호 작용합니다. 데이터 조작 언어(DML)는 데이터를 보거나, 추가하거나, 변경하거나, 삭제하는 데 사용	DBA

작업	설명	필요한 기술
	<p>할 수 있는 SQL의 하위 집합입니다. DDL은 테이블 및 뷰 같은 데이터베이스 객체를 추가하거나, 변경하거나, 삭제하는 데 사용되는 SQL의 하위 집합입니다.</p> <p>자세한 내용은 Amazon Redshift 설명서의 쿼리 성능 조정을 참고하십시오.</p>	
<p>WLM을 구현합니다.</p>	<p>워크로드 관리(WLM)를 사용하여 다수의 쿼리 대기열을 정의한 후 실행 시간에 쿼리를 적합한 대기열로 라우팅할 수 있습니다.</p> <p>자세한 내용은 Amazon Redshift 설명서의 워크로드 관리 구현을 참고하십시오.</p>	<p>DBA</p>
<p>동시성 확장 작업</p>	<p>동시성 확장 기능을 사용하면 일관성 있게 빠른 쿼리 성능으로 동시 사용자 및 동시 쿼리를 사실상 무제한으로 지원할 수 있습니다.</p> <p>자세한 내용은 Amazon Redshift 설명서의 동시성 규모 조정 활용하기를 참고하십시오.</p>	<p>DBA</p>

작업	설명	필요한 기술
<p>Amazon Redshift 테이블 설계 모범 사례를 사용합니다.</p>	<p>데이터베이스를 계획할 때는 전반적인 쿼리 성능에 강력하게 영향을 미칠 수 있는 몇 가지 중요한 테이블 설계 조건이 있습니다.</p> <p>가장 적합한 테이블 설계 옵션을 선택하는 방법에 대한 자세한 내용은 Amazon Redshift 설명서의 Amazon Redshift 테이블 설계 모범 사례를 참고하십시오.</p>	DBA
<p>Amazon Redshift에서 구체화된 뷰를 생성합니다.</p>	<p>구체화된 보기에는 하나 이상의 기본 테이블에 대한 SQL 쿼리를 기반으로 사전 계산된 결과 집합이 포함됩니다. 데이터베이스의 다른 테이블이나 뷰를 쿼리할 때와 같은 방식으로 SELECT 문을 실행하여 구체화된 뷰를 쿼리할 수 있습니다.</p> <p>자세한 내용은 Amazon Redshift 설명서의 Amazon Redshift에서 구체화된 뷰 만들기를 참조하세요.</p>	DBA

작업	설명	필요한 기술
테이블 간 조인을 정의합니다.	<p>에서 ThoughtSpot 동시에 둘 이상의 테이블을 검색하려면 두 테이블에서 일치하는 데이터가 포함된 열을 지정하여 테이블 간 조인을 정의해야 합니다. 이러한 열은 조인의 primary key 및 foreign key를 나타냅니다.</p> <p>Amazon Redshift에서 ALTER TABLE 명령을 사용하거나 명령을 사용하여 정의할 수 있습니다. ThoughtSpot 자세한 내용은 Amazon Redshift 설명서에서 ALTER TABLE을 참조하세요.</p>	DBA

Amazon Redshift에 ThoughtSpot 연결 설정

작업	설명	필요한 기술
Amazon Redshift 연결 추가	<p>Amazon Redshift 연결을 온프레미스 ThoughtSpot 팔콘 데이터베이스에 추가합니다.</p> <p>자세한 내용은 설명서에서 Amazon Redshift 연결 추가를 참조하십시오. ThoughtSpot</p>	DBA
Amazon Redshift 연결을 편집합니다.	Amazon Redshift 연결을 편집하여 테이블과 열을 추가할 수 있습니다.	DBA

작업	설명	필요한 기술
	<p>자세한 내용은 설명서에서 Amazon Redshift 연결 편집을 참조하십시오. ThoughtSpot</p>	
<p>Amazon Redshift 연결을 다시 매핑합니다.</p>	<p>Amazon Redshift 연결을 추가할 때 생성된 소스 매핑 .yaml 파일을 편집하여 연결 매개변수를 수정합니다.</p> <p>예를 들어 기존 테이블 또는 열을 기존 데이터베이스 연결의 다른 테이블 또는 열에 다시 매핑할 수 있습니다. ThoughtSpot 연결에서 테이블 또는 열을 다시 매핑하기 전과 후에 종속성을 확인하여 필요에 따라 표시되는지 확인하는 것이 좋습니다.</p> <p>자세한 내용은 설명서에서 Amazon Redshift 연결 재매핑을 참조하십시오. ThoughtSpot</p>	<p>DBA</p>
<p>Amazon Redshift 연결에서 테이블을 삭제합니다.</p>	<p>(선택 사항) Amazon Redshift 연결에서 테이블을 제거하려고 하면 종속성을 ThoughtSpot 확인하고 종속 객체 목록을 표시합니다. 나열된 객체를 선택하여 삭제하거나 종속성을 제거할 수 있습니다. 그런 다음 테이블을 제거할 수 있습니다.</p> <p>자세한 내용은 설명서의 Amazon Redshift 연결에서 테이블 삭제를 참조하십시오. ThoughtSpot</p>	<p>DBA</p>

작업	설명	필요한 기술
Amazon Redshift 연결에서 종속 객체가 있는 테이블을 삭제합니다.	<p>(선택 사항) 종속 객체가 있는 테이블을 삭제하려고 하면 작업이 차단됩니다. 종속 객체에 대한 링크 목록이 있는 Cannot delete 창이 나타납니다. 모든 종속성이 제거되면 테이블을 삭제할 수 있습니다.</p> <p>자세한 내용은 설명서의 Amazon Redshift 연결에서 종속 객체가 있는 테이블 삭제를 참조하십시오. ThoughtSpot</p>	DBA
Amazon Redshift 연결을 삭제합니다.	<p>(선택 사항) 여러 데이터 소스 또는 시각화에서 연결을 사용할 수 있으므로 Amazon Redshift 연결을 삭제하려면 먼저 해당 연결을 사용하는 모든 소스와 작업을 삭제해야 합니다.</p> <p>자세한 내용은 설명서에서 Amazon Redshift 연결 삭제를 참조하십시오. ThoughtSpot</p>	DBA
Amazon Redshift의 연결 참조를 확인합니다.	<p>설명서의 연결 참조를 사용하여 Amazon Redshift 연결에 필요한 정보를 제공해야 합니다 ThoughtSpot .</p>	DBA

추가 정보

- [Amazon ThoughtSpot Redshift를 통한 모든 규모의 AI 기반 분석](#)
- [Amazon Redshift 요금](#)
- [AWS SCT 시작하기](#)

- [Amazon Redshift 시작하기](#)
- [데이터 추출 에이전트 사용](#)
- [Chick-fil-A는 AWS를 통해 인사이트 확보 속도를 향상시킵니다. ThoughtSpot](#)

AWS DMS를 사용하여 Amazon DynamoDB로 Oracle 데이터베이스 마이그레이션

작성자: Rambabu Karnena(AWS)

환경: PoC 또는 파일럿	소스: 데이터베이스: 관계형	대상: Amazon DynamoDB
R 유형: 리아키텍트	워크로드: Oracle	기술: 마이그레이션, 데이터베이스
AWS 서비스: Amazon DynamoDB		

요약

이 패턴은 AWS Database Migration Service([AWS DMS](#))를 사용하여 Oracle 데이터베이스를 [Amazon DynamoDB](#)로 마이그레이션하는 단계를 안내합니다. 여기에는 다음과 같은 세 가지 유형의 소스 데이터베이스를 포함합니다.

- 온프레미스 Oracle 데이터베이스
- Amazon Elastic Compute Cloud([Amazon EC2](#)) 기반 Oracle 데이터베이스
- Amazon Relational Database Service([RDS](#)) for Oracle DB 인스턴스

이 개념 증명에서 이 패턴은 Amazon RDS for Oracle DB 인스턴스로부터 마이그레이션하는 데 중점을 둡니다.

사전 조건 및 제한 사항

사전 조건

- 활성 상태의 AWS 계정
- Amazon RDS for Oracle 데이터베이스에 연결하는 애플리케이션
- 프라이머리 키 및 샘플 데이터를 사용하여 소스 Amazon RDS for Oracle 데이터베이스에 생성된 테이블

제한 사항

- Amazon DynamoDB는 이러한 데이터베이스 객체를 지원하지 않으므로 프로시저, 함수, 패키지, 트리거와 같은 Oracle 데이터베이스 객체는 마이그레이션 대상으로 고려하지 않습니다.

제품 버전

- 이 패턴은 AWS DMS에서 지원하는 Oracle 데이터베이스의 모든 에디션 및 버전에 적용됩니다. 자세한 내용은 [Oracle 데이터베이스를 AWS DMS용 소스로 사용](#) 및 [Amazon DynamoDB 데이터베이스를 AWS DMS의 대상으로](#) 사용을 참조하세요. 가장 포괄적인 버전 및 기능 지원을 위해 최신 버전의 AWS DMS를 사용하는 것이 좋습니다.

아키텍처

소스 기술 스택

- Amazon RDS for Oracle DB 인스턴스, Amazon EC2 기반 Oracle 또는 온프레미스 Oracle 데이터베이스

대상 기술 스택

- Amazon DynamoDB

AWS 데이터 마이그레이션 아키텍처

도구

- [AWS Database Migration Service\(AWS DMS\)](#)를 사용하면 데이터 스토어를 AWS 클라우드로 마이그레이션하거나 클라우드와 온프레미스 설정 조합 간에 데이터 스토어를 마이그레이션할 수 있습니다.
- [Amazon DynamoDB](#)는 빠르고 예측 가능하고 확장 가능한 성능을 제공하는 완전 관리형 NoSQL 데이터베이스 서비스입니다.
- [Amazon Relational Database Service\(RDS\)](#)는 AWS 클라우드에서 관계형 데이터베이스(DB)를 설정, 운영 및 조정하는 데 도움이 됩니다. 이 패턴은 Amazon RDS for Oracle을 사용합니다.

에픽

마이그레이션 계획

작업	설명	필요한 기술
VPC를 생성합니다.	AWS 계정에서 Virtual Private Cloud(VPC) 및 프라이빗 서브넷을 생성합니다.	시스템 관리자
보안 그룹 및 네트워크 액세스 제어 목록을 생성합니다.	자세한 내용은 AWS 설명서 를 참조하세요.	시스템 관리자
Amazon RDS for Oracle DB 인스턴스를 구성하고 시작합니다.	자세한 내용은 AWS 설명서 를 참조하세요.	DBA, 시스템 관리자

데이터 마이그레이션

작업	설명	필요한 기술
DynamoDB 액세스를 위한 IAM 역할을 생성합니다.	AWS Identity and Access Management(IAM) 콘솔에서 역할을 생성하고 AmazonDynamoDBFullAccess to it 정책을 연결한 다음 AWS DMS를 서비스로 선택합니다.	시스템 관리자
마이그레이션을 위한 AWS DMS 복제 인스턴스를 생성합니다.	복제 인스턴스는 소스 데이터베이스와 동일한 가용 영역 및 VPC에 있어야 합니다.	시스템 관리자
AWS DMS에 소스 및 대상 DB 엔드포인트를 생성합니다.	소스 데이터베이스 엔드포인트를 생성하는 데에는 다음의 두 가지 옵션이 있습니다. <ul style="list-style-type: none"> Amazon RDS 콘솔에서 데이터베이스, DB 식별자, 연결 	시스템 관리자

작업	설명	필요한 기술
	<p>및 보안을 선택하고 엔드포인트를 선택합니다.</p> <ul style="list-style-type: none"> AWS DMS 콘솔에서 RDS DB 인스턴스 선택을 선택합니다. <p>대상 데이터베이스 엔드포인트를 생성하려면 DynamoDB에 액세스하는 이전 작업에서 Amazon 리소스 이름(ARN) 역할을 선택합니다.</p>	
소스 Oracle 데이터베이스 테이블을 DynamoDB에 로드하는 AWS DMS 작업을 생성합니다.	이전 단계에서 소스 및 대상 엔드포인트 이름과 복제 인스턴스를 선택합니다. 유형은 전체 로드일 수 있습니다. Oracle 스키마를 선택하고 %를 지정하여 모든 테이블을 선택합니다.	시스템 관리자
DynamoDB의 테이블을 검증합니다.	마이그레이션 결과를 보려면 DynamoDB 콘솔의 왼쪽 탐색창에서 테이블을 선택합니다.	DBA

애플리케이션 마이그레이션

작업	설명	필요한 기술
애플리케이션 코드를 수정합니다.	DynamoDB에서 데이터에 연결하고 데이터를 검색하려면 애플리케이션 코드를 업데이트합니다.	앱 소유자, DBA, 시스템 관리자

전환

작업	설명	필요한 기술
DynamoDB를 사용하도록 애플리케이션 클라이언트를 전환합니다.		DBA, 앱 소유자, 시스템 관리자

프로젝트 닫기

작업	설명	필요한 기술
AWS 리소스를 종료합니다.	예를 들어 Amazon RDS for Oracle 인스턴스, DynamoDB 및 AWS DMS 복제 인스턴스를 종료합니다.	DBA, 시스템 관리자
지표를 수집합니다.	지표에는 마이그레이션 시간, 수동 작업 및 도구로 수행한 작업의 비율, 비용 절감 등이 포함됩니다.	DBA, 앱 소유자, 시스템 관리자

관련 리소스

- [AWS Database Migration Service and Amazon DynamoDB: What You Need to Know](#)(블로그 게시물)
- [Oracle 데이터베이스를 AWS DMS의 소스로 사용](#)
- [Amazon DynamoDB 데이터베이스를 AWS Database Migration Service의 대상으로 사용](#)
- [RDBMS에서 Amazon DynamoDB로 마이그레이션하기 위한 모범 사례](#)(백서)

AWS DMS를 사용하여 Oracle 파티션형 테이블을 PostgreSQL로 마이그레이션하기

작성: Saurav Mishra(AWS) 및 Eduardo Valentim(AWS)

환경: PoC 또는 파일럿	소스: Oracle 데이터베이스	대상: PostgreSQL 9.0
R 유형: 리아키텍트	워크로드: Oracle	기술: 마이그레이션, 데이터베이스, 스토리지 및 백업
AWS 서비스: AWS DMS		

요약

이 패턴은 네이티브 파티셔닝을 지원하지 않는 AWS Database Migration Service(AWS DMS)를 사용하여 파티션을 나눈 테이블을 Oracle에서 PostgreSQL로 빠르게 로드하는 방법을 설명합니다. 대상 PostgreSQL 데이터베이스는 Amazon Elastic Compute Cloud(Amazon EC2)에 설치하거나 PostgreSQL용 Amazon Relational Database Service(Amazon RDS) 또는 Amazon Aurora PostgreSQL-Compatible 에디션 DB 인스턴스가 될 수 있습니다.

파티션을 나눈 테이블 업로드에는 다음 단계가 포함됩니다.

1. Oracle 파티션 테이블과 비슷하지만 파티션을 포함하지 않는 상위 테이블을 생성합니다.
2. 1단계에서 만든 상위 테이블을 상속할 하위 테이블을 생성합니다.
3. 상위 테이블에서 삽입을 처리하는 프로시저 함수와 트리거를 생성합니다.

하지만, 트리거는 모든 삽입에 대해 실행되므로 AWS DMS를 사용한 초기 로드는 매우 느릴 수 있습니다.

Oracle에서 PostgreSQL 9.0으로의 초기 로드 속도를 높이기 위해 이 패턴은 각 파티션에 대해 별도의 AWS DMS 작업을 생성하고 해당 하위 테이블을 로드합니다. 그런 다음 컷오버 중에 트리거를 생성합니다.

PostgreSQL 버전 10은 기본 분할을 지원합니다. 하지만 경우에 따라 상속된 파티셔닝을 사용하기로 결정할 수도 있습니다. 자세한 내용은 [추가 정보](#) 섹션을 참조하세요.

사전 조건 및 제한 사항

사전 조건

- 활성 상태의 AWS 계정.
- 파티셔닝된 테이블이 있는 소스 Oracle 데이터베이스
- AWS 기반 PostgreSQL 데이터베이스

제품 버전

- PostgreSQL 9.0

아키텍처

소스 기술 스택

- Oracle에 있는 파티셔닝된 테이블

대상 기술 스택

- PostgreSQL에 있는 파티셔닝된 테이블(Amazon EC2, Amazon RDS for PostgreSQL, 또는 Aurora PostgreSQL 기반)

대상 아키텍처

도구

- [AWS Database Migration Service\(AWS DMS\)](#)를 사용하면 데이터 스토어를 AWS 클라우드로 마이그레이션하거나 클라우드와 온프레미스 설정 조합 간에 마이그레이션할 수 있습니다.

에픽

AWS DMS 설정

작업	설명	필요한 기술
PostgreSQL에서 테이블을 생성합니다.	파티션에 필요한 검사 조건을 사용하여 PostgreSQL에서 상위 및 해당 하위 테이블을 생성합니다.	DBA
각 파티션에 대해 AWS DMS 작업을 생성합니다.	AWS DMS 작업에 파티션의 필터 조건을 포함합니다. 파티션을 해당하는 PostgreSQL 하위 테이블에 매핑합니다.	DBA
CDC(전체 로드 및 변경 데이터 캡처)를 사용해 AWS DMS 작업을 실행합니다.	StopTaskCachedChangesApplied 파라미터가 true로 설정되어 있고 StopTaskCachedChangesNotApplied 파라미터가 false로 설정되어 있어야 합니다.	DBA

전환

작업	설명	필요한 기술
복제 작업을 중지합니다.	작업을 중지하기 전에 소스와 대상이 동기화되어 있는지 확인합니다.	DBA
상위 테이블에 트리거를 생성합니다.	부모 테이블은 모든 삽입 및 업데이트 명령을 수신하므로 파티셔닝 조건에 따라 이러한 명령을 각 하위 테이블로 라우팅하는 트리거를 만듭니다.	DBA

관련 리소스

- [DMS](#)
- [테이블 파티셔닝\(PostgreSQL 설명서\)](#)

추가 정보

PostgreSQL 버전 10은 네이티브 파티셔닝을 지원하지만 다음과 같은 사용 사례에서는 상속된 파티셔닝을 사용하기로 결정할 수 있습니다.

- 파티셔닝에서는 모든 파티션에 상위 파티션과 동일한 열 집합이 있어야 한다는 규칙이 적용되지만 테이블 상속은 하위 파티션에 추가 열을 가질 수 있도록 지원합니다.
- 테이블 상속은 다중 상속을 지원합니다.
- 선언적 파티셔닝은 목록 및 범위 파티셔닝만 지원합니다. 테이블 상속을 사용하면 데이터를 원하는 대로 나눌 수 있습니다. 그러나 제약 조건 제외가 파티션을 효과적으로 정리하지 못하면 쿼리 성능이 저하됩니다.
- 일부 작업에서는 선언적 파티셔닝을 사용할 때 테이블 상속을 사용할 때보다 더 강력한 잠금 장치가 필요합니다. 예를 들어, 파티셔닝 테이블에 파티션을 추가하거나 제거하려면 상위 테이블에 대해 ACCESS EXCLUSIVE 잠금 장치가 필요한 반면, 일반 상속에는 SHARE UPDATE EXCLUSIVE 잠금 장치만으로도 충분합니다.

별도의 작업 파티션을 사용하는 경우 AWS DMS 검증 문제가 있으면 파티션을 다시 로드할 수도 있습니다. 성능 및 복제 제어를 개선하려면 별도의 복제 인스턴스에서 작업을 실행하십시오.

Amazon RDS for Oracle에서 Amazon RDS for MySQL로 마이그레이션

지텐더 쿠마르 (AWS), 네하 샨마 (AWS), 스리니 라마스와미 (AWS) 가 제작했습니다.

환경: PoC 또는 파일럿	소스: Amazon RDS for Oracle	대상: Amazon RDS for MySQL
R 유형: 리아키텍트	워크로드: Oracle	기술: 마이그레이션, 데이터베이스

AWS 서비스: Amazon RDS

요약

이 패턴은 오라클 DB 인스턴스용 아마존 관계형 데이터베이스 서비스 (Amazon RDS) 를 아마존 웹 서비스 (AWS) 의 MySQL용 아마존 RDS DB 인스턴스로 마이그레이션하기 위한 지침을 제공합니다. 패턴은 AWS 데이터베이스 마이그레이션 서비스 (AWS DMS) 및 AWS 스키마 변환 도구 (AWS SCT) 를 사용합니다.

이 패턴은 저장 프로시저의 마이그레이션을 처리하는 모범 사례를 제공합니다. 또한 응용 프로그램 계층을 지원하기 위한 코드 변경 사항도 다룹니다.

사전 조건 및 제한 사항

사전 조건

- 활성 상태의 AWS 계정
- Amazon RDS for Oracle 소스 데이터베이스
- Amazon RDS for MySQL의 타겟 데이터베이스 소스 및 대상 데이터베이스는 동일한 가상 사설 클라우드 (VPC) 에 있어야 합니다. 여러 VPC를 사용하는 경우 또는 필요한 액세스 권한이 있어야 합니다.
- 소스 및 타겟 데이터베이스, AWS SCT, 애플리케이션 서버, AWS DMS 간의 연결을 허용하는 보안 그룹
- 원본 데이터베이스에서 AWS SCT를 실행하는 데 필요한 권한을 가진 사용자 계정
- 원본 데이터베이스에서 AWS DMS를 실행하기 위한 추가 로깅이 활성화되었습니다.

제한 사항

- 원본 및 타겟 Amazon RDS 데이터베이스 크기 제한은 64TB입니다. Amazon RDS 크기 정보는 [AWS 설명서를](#) 참조하십시오.
- Oracle은 데이터베이스 개체에 대해 대소문자를 구분하지 않지만 MySQL은 그렇지 않습니다. AWS SCT는 객체를 생성하는 동안 이 문제를 처리할 수 있습니다. 하지만 대소문자를 완전히 구분하지 않도록 지원하려면 몇 가지 수동 작업이 필요합니다.
- 이 마이그레이션에서는 MySQL 확장을 사용하여 Oracle 네이티브 함수를 활성화하지 않습니다. AWS SCT가 대부분의 변환을 처리하지만 코드를 수동으로 변경하려면 몇 가지 작업이 필요합니다.
- JDBC(Java Database Connectivity) 드라이버를 변경해야 합니다.

제품 버전

- 오라클 12.2.0.1 이상용 아마존 RDS 현재 지원되는 Oracle용 RDS 버전은 [AWS 설명서를](#) 참조하십시오.
- MySQL용 아마존 RDS 8.0.15 이상 현재 지원되는 MySQL용 RDS 버전은 [AWS 설명서를](#) 참조하십시오.
- AWS DMS 버전 3.3.0 이상 AWS DMS 지원 [소스 엔드포인트 및 대상 엔드포인트에](#) 대한 자세한 내용은 AWS 설명서를 참조하십시오.
- AWS SCT 버전 1.0.628 이상 AWS 설명서에서 [AWS SCT 소스 및 대상 엔드포인트 지원 매트릭스](#)를 참조하십시오.

아키텍처

소스 기술 스택

- Amazon RDS for Oracle 자세한 내용은 [Oracle 데이터베이스를 AWS DMS의 원본으로 사용을](#) 참조하십시오.

대상 기술 스택

- Amazon RDS for MySQL 자세한 내용은 [MySQL 호환 데이터베이스를 AWS DMS의 대상으로 사용을](#) 참조하십시오.

마이그레이션 아키텍처

다음 다이어그램에서 AWS SCT는 Oracle용 Amazon RDS의 원본 데이터베이스에서 스키마 객체를 복사 및 변환하고, 이 객체를 MySQL용 Amazon RDS 대상 데이터베이스로 보냅니다. AWS DMS는 원본 데이터베이스에서 데이터를 복제하여 MySQL용 Amazon RDS 인스턴스로 전송합니다.

도구

- [AWS Data Migration Service](#)를 사용하면 데이터 스토어를 AWS 클라우드로 마이그레이션하거나 클라우드와 온프레미스 설정 간에 데이터 스토어를 마이그레이션할 수 있습니다.
- [Amazon Relational Database Service\(RDS\)](#)는 AWS 클라우드에서 관계형 데이터베이스를 설정, 운영 및 규모를 조정하는 데 도움이 됩니다. 이 패턴은 오라클용 [아마존 RDS](#)와 MySQL용 [아마존 RDS](#)를 사용합니다.
- [AWS Schema Conversion Tool\(AWS SCT\)](#)은 소스 데이터베이스 스키마와 대부분의 사용자 지정 코드를 대상 데이터베이스와 호환되는 형식으로 자동 변환하여 이기종 데이터베이스 마이그레이션을 지원합니다.

에픽

마이그레이션 준비

작업	설명	필요한 기술
소스 및 대상 데이터베이스 버전과 엔진을 검증합니다.		DBA
대상 서버 인스턴스의 하드웨어 요구 사항을 식별합니다.		DBA, SysAdmin
스토리지 요구 사항(스토리지 유형 및 용량)을 식별합니다.		DBA, SysAdmin
적절한 인스턴스 유형(용량, 스토리지 특성, 네트워크 특성)을 선택합니다.		DBA, SysAdmin

작업	설명	필요한 기술
소스 및 대상 데이터베이스의 네트워크 액세스 보안 요구 사항을 식별합니다.		DBA, SysAdmin
애플리케이션 마이그레이션 전략을 선택합니다.	전환 활동을 위해 전체 다운타임을 원하는지 아니면 부분적 다운타임을 원하는지 고려하세요.	DBA SysAdmin, 앱 소유자

인프라 구성

작업	설명	필요한 기술
VPC 및 서브넷을 생성합니다.		SysAdmin
보안 그룹 및 네트워크 액세스 제어 목록(ACL)을 생성합니다.		SysAdmin
Amazon RDS for Oracle 인스턴스를 구성하고 시작합니다.		DBA, SysAdmin
Amazon RDS for MySQL 인스턴스를 구성하고 시작합니다.		DBA, SysAdmin
코드 변환 검증을 위한 테스트 케이스를 준비하세요.	이렇게 하면 변환된 코드의 유닛 테스트에 도움이 됩니다.	DBA, 개발자
AWS DMS 인스턴스를 구성합니다.		
AWS DMS에서 소스 및 타겟 엔드포인트를 구성합니다.		

데이터 마이그레이션

작업	설명	필요한 기술
AWS SCT를 사용하여 타겟 데이터베이스 스크립트를 생성합니다.	AWS SCT에서 변환한 코드의 정확성을 확인하세요. 일부 수동 작업이 필요할 수 있습니다.	DBA, 개발자
AWS SCT에서는 '대소문자 구분 안 함' 설정을 선택합니다.	AWS SCT에서 프로젝트 설정, 타겟 대소문자 구분, 대소문자 구분을 선택합니다.	DBA, 개발자
AWS SCT에서는 Oracle 네이티브 함수를 사용하지 않도록 선택하세요.	프로젝트 설정에서 TO_CHAR/TO_NUMBER/TO_DATE 함수를 확인하세요.	DBA, 개발자
'sql%not found' 코드를 변경합니다.	코드를 수동으로 변환해야 할 수도 있습니다.	
저장 프로시저의 테이블 및 개체에 대한 쿼리(소문자 쿼리 사용).		DBA, 개발자
모든 변경이 이루어진 후 기본 스크립트를 만든 다음 타겟 데이터베이스에 기본 스크립트를 배포합니다.		DBA, 개발자
샘플 데이터를 사용하여 저장된 프로시저와 애플리케이션 직접 호출을 유닛 테스트합니다.		
유닛 테스트 중에 생성된 데이터를 정리합니다.		DBA, 개발자
타겟 데이터베이스에 외래 키 제약 조건을 삭제하세요.	이 단계는 초기 데이터를 로드하는 데 필요합니다. 외래 키 제약 조건을 삭제하지 않으려면	DBA, 개발자

작업	설명	필요한 기술
	기본 및 보조 테이블 전용 데이터에 대한 마이그레이션 작업을 만들어야 합니다.	
타겟 데이터베이스에 프라이머리 키와 고유 키를 삭제합니다.	이 단계를 수행하면 초기 로드 성능이 향상됩니다.	DBA, 개발자
소스 데이터베이스에서 추가 로깅을 활성화합니다.		DBA
AWS DMS에서 초기 로드를 위한 마이그레이션 작업을 생성한 다음 실행합니다.	기존 데이터 마이그레이션 옵션을 선택합니다.	DBA
프라이머리 키와 외래 키를 타겟 데이터베이스에 추가합니다.	초기 로드 후 제약 조건을 추가해야 합니다.	DBA, 개발자
지속적인 복제를 위한 마이그레이션 작업을 생성합니다.	지속적인 복제를 통해 타겟 데이터베이스를 원본 데이터베이스와 동기화합니다.	DBA

애플리케이션 마이그레이션

작업	설명	필요한 기술
Oracle 네이티브 함수를 MySQL 네이티브 함수로 대체하세요.		앱 소유자
SQL 쿼리의 데이터베이스 개체에는 소문자만 사용해야 합니다.		DBA SysAdmin, 앱 소유자

대상 데이터베이스로 전환

작업	설명	필요한 기술
애플리케이션 서버를 종료합니다.		앱 소유자
소스 및 대상 데이터베이스가 동기화되어 있는지 검증합니다.		DBA, 앱 소유자
Amazon RDS for Oracle DB 인스턴스를 중지합니다.		DBA
마이그레이션 작업을 중지합니다.	이전 단계를 완료하면 자동으로 중지됩니다.	DBA
JDBC 연결을 Oracle에서 MySQL로 변경합니다.		앱 소유자, DBA
애플리케이션을 시작합니다.		DBA SysAdmin, 앱 소유자

프로젝트 닫기

작업	설명	필요한 기술
프로젝트 문서를 검토하고 검증하세요.		DBA, SysAdmin
마이그레이션 시간, 수동 작업과 도구 작업의 비율, 비용 절감 등에 대한 지표를 수집하세요.		DBA, SysAdmin
AWS DMS 인스턴스를 중지하고 삭제합니다.		DBA
원본 및 대상 엔드포인트를 제거합니다.		DBA

작업	설명	필요한 기술
마이그레이션 작업을 제거합니다.		DBA
Amazon RDS for Oracle DB 인스턴스의 스냅샷을 생성합니다.		DBA
Amazon RDS for Oracle DB 인스턴스를 삭제합니다.		DBA
사용한 다른 임시 AWS 리소스를 종료하고 삭제합니다.		DBA, SysAdmin
프로젝트를 종료하고 피드백을 제공하세요.		DBA

관련 리소스

- [DMS](#)
- [AWS SCT](#)
- [Amazon RDS 요금 책정](#)
- [AWS DMS 시작하기](#)
- [Amazon RDS 시작](#)

AWS DMS 및 AWS SCT를 사용하여 Amazon EC2의 IBM Db2에서 PostgreSQL과 호환되는 Aurora PostgreSQL로 마이그레이션하십시오.

시르센두 할더 (AWS) 와 사친 코트월(AWS)이 제작했습니다.

환경: PoC 또는 파일럿	소스: IBM Db2	대상: Aurora PostgreSQL-Compatible
R 타입: 리아키텍트	워크로드: IBM	기술: 마이그레이션, 데이터베이스

AWS 서비스: Amazon Aurora;
AWS DMS; AWS SCT

요약

이 패턴은 Amazon Elastic Compute Cloud(Amazon EC2) 인스턴스의 IBM Db2 데이터베이스를 Amazon Aurora PostgreSQL 호환 Edition DB 인스턴스로 마이그레이션하기 위한 지침을 제공합니다. 이 패턴은 데이터 마이그레이션 및 스키마 변환을 위해 AWS 데이터베이스 마이그레이션 서비스 (AWS DMS) 및 AWS Schema Conversion Tool (AWS SCT) 을 사용합니다.

이 패턴은 트랜잭션 수가 많은 테라바이트급 IBM Db2 데이터베이스의 다운타임이 거의 또는 전혀 없는 온라인 마이그레이션 전략을 대상으로 합니다. 성능 향상을 위해 데이터 유형이 NUMERIC인 프라이머리 키(PK)와 외래 키(FK)의 열을 PostgreSQL에서 INT 또는 BIGINT로 변환하는 것이 좋습니다.

사전 조건 및 제한 사항

사전 조건

- 활성 상태의 AWS 계정.
- EC2 인스턴스의 원본 IBM Db2 데이터베이스

제품 버전

- DB2/LINUX8664 버전 11.1.4.4 이상

아키텍처

소스 기술 스택

- EC2 인스턴스의 Db2 데이터베이스

대상 기술 스택

- Aurora PostgreSQL 호환 버전 10.18 이상의 DB 인스턴스

데이터베이스 마이그레이션 아키텍처

도구

- [AWS Database Migration Service\(AWS DMS\)](#)를 사용하면 데이터베이스를 AWS 클라우드로 마이그레이션하거나 클라우드와 온프레미스 설정 간에 데이터베이스를 마이그레이션할 수 있습니다. 소스 데이터베이스는 마이그레이션 중에도 완전히 작동하여 데이터베이스를 사용하는 애플리케이션의 가동 중지 시간을 최소화합니다. AWS DMS를 사용하여 가장 널리 사용되는 상용 및 오픈 소스 데이터베이스로 데이터를 마이그레이션할 수 있습니다. AWS DMS는 서로 다른 데이터베이스 플랫폼 간의 이기종 마이그레이션을 지원합니다. 예를 들어 IBM Db2에서 Aurora PostgreSQL 호환 버전 10.18 이상으로의 마이그레이션을 지원합니다. 자세한 내용은 AWS DMS 설명서의 [데이터 마이그레이션 소스 및 데이터 마이그레이션 대상](#)을 참조하십시오.
- [AWS Schema Conversion Tool\(AWS SCT\)](#)는 원본 데이터베이스 스키마와 대부분의 데이터베이스 코드 객체 (보기, 저장된 프로시저, 함수 등) 를 대상 데이터베이스와 호환되는 형식으로 자동 변환하여 이기종 데이터베이스 마이그레이션을 지원합니다. 자동으로 변환되지 않는 모든 객체는 명확하게 표시되므로 수동으로 변환하여 마이그레이션을 완료할 수 있습니다. 또한, AWS SCT는 내장된 SQL 문에 대한 애플리케이션 소스 코드를 스캔하고 이를 변환할 수 있습니다.

에픽

환경 설정

작업	설명	필요한 기술
Aurora PostgreSQL 호환 DB 인스턴스를 생성합니다.	DB 인스턴스를 생성하려면 AWS 설명서 의 지침을 따르세요	Amazon RDS

작업	설명	필요한 기술
	<p>요. 엔진 유형(Engine type)에서 Amazon Aurora를 선택합니다. 에디션의 경우 Amazon Aurora PostgreSQL 호환 에디션을 선택합니다.</p> <p>Aurora PostgreSQL 호환 버전 10.18 이상의 DB 인스턴스는 소스 IBM Db2 데이터베이스와 동일한 Virtual Private Cloud (Virtual Private Cloud) 에 있어야 합니다.</p>	

데이터베이스 스키마를 변환하세요.

작업	설명	필요한 기술
<p>AWS SCT를 설치하고 확인합니다.</p>	<ol style="list-style-type: none"> 1. AWS SCT 설명서의 단계에 따라 AWS SCT를 설치하십시오. 2. AWS SCT 설명서의 절차에 따라 설치를 확인하십시오. 	<p>AWS 관리자, DBA, 마이그레이션 엔지니어</p>
<p>AWS SCT를 시작하고 프로젝트를 생성합니다.</p>	<p>AWS SCT 도구를 시작하고 새 프로젝트를 생성하여 데이터베이스 마이그레이션 평가 보고서를 실행하려면 AWS SCT 설명서의 지침을 따르십시오.</p>	<p>마이그레이션 엔지니어</p>
<p>데이터베이스 서버를 추가하고 매핑 규칙을 생성합니다.</p>	<ol style="list-style-type: none"> 1. AWS SCT 설명서의 지침에 따라 원본 및 대상 데이터베이스 서버를 추가합니다. 2. 매핑 규칙을 생성하여 소스 데이터베이스의 대상 데이터베이스 플랫폼을 정의하 	<p>마이그레이션 엔지니어</p>

작업	설명	필요한 기술
	<p>십시오. 지침은 AWS SCT 설명서를 참조하세요.</p>	
<p>데이터베이스 마이그레이션 평가 보고서를 만드세요.</p>	<p>AWS SCT 설명서의 단계에 따라 데이터베이스 마이그레이션 평가 보고서를 생성합니다.</p>	<p>마이그레이션 엔지니어</p>
<p>평가 보고서</p>	<p>데이터베이스 마이그레이션 평가 보고서의 요약 탭을 사용하여 보고서를 보고 데이터를 분석할 수 있습니다. 이 분석을 통해 마이그레이션의 복잡성을 파악할 수 있습니다. 자세한 내용은 AWS SCT 설명서를 참조하십시오.</p>	<p>마이그레이션 엔지니어</p>
<p>스키마를 변환합니다.</p>	<p>소스 데이터베이스 스키마를 변환하려면:</p> <ol style="list-style-type: none"> 1. AWS SCT 콘솔에서 보기를 선택한 다음 기본 보기를 선택합니다. 2. 소스 스키마에서 객체 또는 상위 노드를 선택하고 컨텍스트 (마우스 오른쪽 버튼 클릭) 메뉴를 연 다음 스키마 변환을 선택합니다. <p>자세한 내용은 AWS SCT 설명서를 참조하십시오.</p>	<p>마이그레이션 엔지니어</p>

작업	설명	필요한 기술
<p>변환된 데이터베이스 스키마를 대상 DB 인스턴스에 적용합니다.</p>	<ol style="list-style-type: none"> 1. 대상 DB 인스턴스에 대해 계획된 스키마를 표시하는 프로젝트의 오른쪽 패널에서 스키마 요소를 선택합니다. 2. 스키마 요소의 컨텍스트 메뉴 (마우스 오른쪽 버튼 클릭) 를 연 다음 데이터베이스에 적용을 선택합니다. <p>자세한 내용은 AWS SCT 설명서를 참조하십시오.</p>	<p>마이그레이션 엔지니어</p>

데이터 마이그레이션

작업	설명	필요한 기술
<p>VPC 및 DB 파라미터 그룹을 설정합니다.</p>	<p>VPC 및 DB 파라미터 그룹을 설정하고 마이그레이션에 필요한 인바운드 규칙과 파라미터를 구성합니다. 지침은 AWS DMS 설명서를 참조하세요.</p> <p>VPC 보안 그룹의 경우 Db2 용 EC2 인스턴스와 Aurora PostgreSQL 호환 DB 인스턴스를 모두 선택합니다. 이 복제 인스턴스는 소스 및 대상 DB 인스턴스와 동일한 VPC에 있어야 합니다.</p>	<p>마이그레이션 엔지니어</p>
<p>소스 및 대상 DB 인스턴스를 준비합니다.</p>	<p>마이그레이션할 원본 및 대상 DB 인스턴스를 준비합니다. 프로덕션 환경에서는 원본 데이터베이스가 이미 존재합니다.</p>	<p>마이그레이션 엔지니어</p>

작업	설명	필요한 기술
	<p>원본 데이터베이스의 경우 서버 이름은 Db2가 실행 중인 EC2 인스턴스의 퍼블릭 도메인 이름 시스템 (DNS) 이어야 합니다. 사용자 이름의 경우 포트 db2inst1 뒤에 사용할 수 있습니다. IBM Db2의 경우 5000이 됩니다.</p>	
<p>Amazon EC2 클라이언트 및 엔드포인트를 생성하십시오.</p>	<ol style="list-style-type: none"> 1. Amazon EC2 클라이언트를 생성하십시오. 이 클라이언트를 사용하여 복제할 데이터로 원본 데이터베이스를 채웁니다. 또한 이 클라이언트를 사용하여 대상 데이터베이스에서 쿼리를 실행하여 복제를 확인할 수 있습니다. 2. 다음 단계에 사용할 원본 데이터베이스 및 대상 DB 인스턴스의 엔드포인트를 생성합니다. 지침은 AWS DMS 설명서를 참조하세요. 소스 데이터베이스와 대상 데이터베이스에 대해 별도의 엔드포인트를 만들어야 합니다. Aurora PostgreSQL 호환 버전 10.18 이상의 경우 포트는 5432가 되며 DB 인스턴스의 엔드포인트에서 서버 이름을 가져올 수 있습니다. 	<p>마이그레이션 엔지니어</p>

작업	설명	필요한 기술
복제 인스턴스를 만듭니다.	AWS DMS 콘솔을 사용하여 복제 인스턴스를 생성하고 소스 및 대상 엔드포인트를 지정합니다. 복제 인스턴스는 엔드포인트 간 데이터 마이그레이션을 수행합니다. 자세한 내용은 the AWS DMS 설명서 를 참조하십시오.	마이그레이션 엔지니어
데이터를 마이그레이션할 AWS DMS 작업을 만듭니다.	<p>AWS DMS 설명서의 단계에 따라 소스 IBM Db2 테이블을 대상 PostgreSQL DB 인스턴스로 로드하는 작업을 생성합니다.</p> <ul style="list-style-type: none"> 원본 및 대상에는 원본 및 대상 엔드포인트 이름을 사용하십시오. 마이그레이션 유형은 전체 로드일 수 있습니다. 스키마 규칙의 경우 Db2 데이터베이스의 inst1 스키마를 사용할 수 있습니다. 테이블 이름의 경우 모든 테이블을 % 마이그레이션하도록 지정합니다. 로드가 완료되면 Aurora PostgreSQL 호환 데이터베이스에 inst1 스키마의 Db2 테이블이 나타납니다. 	마이그레이션 엔지니어

관련 리소스

참조

- [Amazon Aurora 설명서](#)
- [PostgreSQL 외부 데이터 래퍼 \(FDW\) 설명서](#)
- [PostgreSQL 임포트 외부 스키마 문서](#)
- [AWS DMS 설명서](#)
- [AWS SCT 설명서](#)

자습서 및 동영상

- [AWS DMS 시작하기 \(둘러보기\)](#)
- [Amazon EC2 소개 - AWS 기반 탄력적 클라우드 서버 및 호스팅 \(동영상\)](#)

AWS DMS를 사용하여 Oracle 8i 또는 9i에서 PostgreSQL용 아마존 RDS로 마이그레이션하십시오. SharePlex

작성자: Kumar Babu P G(AWS)

환경: PoC 또는 파일럿	소스: 데이터베이스: 관계형	대상: Amazon RDS for PostgreSQL / Amazon Aurora PostgreSQL
R 유형: 리아키텍트	워크로드: Oracle	기술: 마이그레이션, 데이터베이스

AWS 서비스: Amazon RDS, Amazon Aurora

요약

이 패턴은 온프레미스 Oracle 8i 또는 9i 데이터베이스를 PostgreSQL용 Amazon Relational Database Service(Amazon RDS) 또는 Amazon Aurora PostgreSQL로 마이그레이션하는 방법을 설명합니다. AWS 데이터베이스 마이그레이션 서비스 (AWS DMS) 는 오라클 8i 또는 9i를 소스로 지원하지 않으므로 Quest는 온프레미스 8i 또는 9i 데이터베이스의 데이터를 AWS DMS와 호환되는 중간 오라클 데이터베이스 (오라클 10g 또는 11g) 로 SharePlex 복제합니다.

중간 Oracle 인스턴스에서 스키마와 데이터는 AWS Schema Conversion Tool(AWS SCT) 및 AWS DMS를 사용하여 AWS의 PostgreSQL 데이터베이스로 마이그레이션됩니다. 이 방법을 이용하면 복제 지연을 최소화하면서 소스 Oracle 데이터베이스에서 대상 PostgreSQL DB 인스턴스로 데이터를 지속적으로 스트리밍할 수 있습니다. 이 구현에서 가동 중지 시간은 대상 PostgreSQL 데이터베이스에서 모든 외부 키, 트리거 및 시퀀스를 생성하거나 검증하는 데 걸리는 시간으로 제한됩니다.

마이그레이션에서는 Oracle 10g 또는 11g가 설치된 Amazon Elastic Compute Cloud(Amazon EC2) 인스턴스를 사용하여 소스 Oracle 데이터베이스의 변경 사항을 호스팅합니다. AWS DMS는 이 중간 Oracle 인스턴스를 소스로 사용하여 Amazon RDS for PostgreSQL 또는 Aurora PostgreSQL로 데이터를 스트리밍합니다. 온프레미스 Oracle 데이터베이스에서 중간 Oracle 인스턴스로의 데이터 복제를 일시 중지했다가 다시 시작할 수 있습니다. 또한 중간 Oracle 인스턴스에서 대상 PostgreSQL 데이터베이스로 일시 중지했다가 다시 시작할 수 있으므로 AWS DMS 데이터 검증 또는 사용자 지정 데이터 검증 도구를 사용하여 데이터를 검증할 수 있습니다.

사전 조건 및 제한 사항

사전 조건

- 활성 상태의 AWS 계정
- 온프레미스 데이터 센터의 소스 Oracle 8i 또는 9i 데이터베이스
- 온프레미스 데이터 센터와 AWS 간에 구성된 AWS Direct Connect
- AWS SCT가 설치된 로컬 시스템 또는 EC2 인스턴스에 설치된 AWS SCT 커넥터용 Java Database Connectivity(JDBC) 드라이버
- [Oracle 데이터베이스를 AWS DMS 소스로 사용](#) 속지
- [PostgreSQL 데이터베이스를 AWS DMS 대상으로 사용](#) 속지
- Quest 데이터 복제에 익숙해야 함 SharePlex

제한 사항

- 데이터베이스 크기 제한: 64TB
- 온프레미스 Oracle 데이터베이스는 Enterprise Edition이어야 함

제품 버전

- 소스 데이터베이스: Oracle 8i 또는 9i
- 중간 데이터베이스: Oracle 10g 또는 11g
- PostgreSQL 9.6 이상

아키텍처

소스 기술 스택

- Oracle 8i 또는 9i 데이터베이스
- 퀘스트 SharePlex

대상 기술 스택

- Amazon RDS for PostgreSQL 또는 Aurora PostgreSQL

소스 및 대상 아키텍처

도구

- AWS DMS-[AWS Database Migration Service](#)(AWS DMS)를 이용하면 데이터베이스를 빠르고 안전하게 마이그레이션할 수 있습니다. 소스 데이터베이스는 마이그레이션 중에도 완전히 작동하여 데이터베이스를 사용하는 애플리케이션의 가동 중지 시간을 최소화합니다. AWS DMS는 광범위하게 사용되는 상용 및 오픈 소스 데이터베이스 간에 데이터를 마이그레이션할 수 있습니다.
- AWS SCT-[AWS Schema Conversion Tool](#)(AWS SCT)은 소스 데이터베이스 스키마와 대부분의 데이터베이스 코드 객체(뷰, 저장 프로시저, 함수 등)를 대상 데이터베이스와 호환되는 형식으로 자동 변환하여 이기종 데이터베이스 마이그레이션을 예측 가능하게 만듭니다. 자동으로 변환할 수 없는 객체는 명확하게 표시되므로 수동으로 변환하여 마이그레이션을 완료할 수 있습니다. AWS SCT는 애플리케이션 소스 코드에서 내장된 SQL 문을 스캔하고 이를 데이터베이스 스키마 변환 프로젝트의 일부로 변환할 수도 있습니다. 이 프로세스 중에 AWS SCT는 기존 Oracle 및 SQL Server 함수를 이와 동등한 AWS 함수로 변환하여 클라우드 네이티브 코드 최적화를 수행하므로 데이터베이스를 마이그레이션하는 동시에 애플리케이션을 현대화할 수 있습니다. 스키마 변환이 완료되면 AWS SCT는 내장된 데이터 마이그레이션 에이전트를 사용하여 다양한 데이터 웨어하우스에서 Amazon Redshift로 데이터를 마이그레이션하는 데 도움을 줄 수 있습니다.
- Quest SharePlex — [SharePlex Quest](#)는 다운타임을 최소화하고 데이터 손실 없이 데이터를 이동할 수 있는 Oracle-Oracle-Oracle의 데이터 복제 도구입니다.

에픽

EC2 인스턴스 생성 및 Oracle 설치

작업	설명	필요한 기술
Amazon EC2용 네트워크를 설정합니다.	Virtual Private Cloud(VPC), 서브넷, 인터넷 게이트웨이, 라우팅 테이블, 보안 그룹을 생성합니다.	AWS SysAdmin

작업	설명	필요한 기술
새 EC2 인스턴스를 생성합니다.	EC2 인스턴스의 Amazon Machine Image(AMI)를 선택합니다. 인스턴스 크기를 선택하고 인스턴스 세부 정보(인스턴스 수(1), 이전 단계의 VPC 및 서브넷, 퍼블릭 IP 자동 할당 및 기타 옵션)를 구성합니다. 스토리지를 추가하고, 보안 그룹을 구성하여 인스턴스를 시작합니다. 메시지가 표시되면 다음 단계를 위해 키 페어를 생성하고 저장합니다.	AWS SysAdmin
EC2 인스턴스에 Oracle을 설치합니다.	라이선스와 필요한 Oracle 바이너리를 가져오고 EC2 인스턴스에 Oracle 10g 또는 11g를 설치합니다.	DBA

EC2 인스턴스에 설정하고 SharePlex 데이터 복제를 구성합니다.

작업	설명	필요한 기술
설정. SharePlex	Amazon EC2 인스턴스를 생성하고 Oracle 8i 또는 9i와 호환되는 SharePlex 바이너리를 설치합니다.	AWS SysAdmin, DBA
데이터 복제를 구성합니다.	SharePlex 모범 사례에 따라 온프레미스 Oracle 8i/9i 데이터베이스에서 Oracle 10g/11g 인스턴스로 데이터 복제를 구성하십시오.	DBA

Oracle 데이터베이스 스키마를 PostgreSQL로 변환

작업	설명	필요한 기술
AWS SCT를 설정합니다.	새 보고서를 생성한 다음 Oracle에 소스로 연결하고 PostgreSQL에 대상으로 연결합니다. 프로젝트 설정에서 SQL 스크립팅 탭을 열고 대상 SQL 스크립트를 여러 파일로 변경합니다.	DBA
Oracle 데이터베이스 스키마를 변환합니다.	작업 탭에서 보고서 생성, 스키마 변환을 선택한 다음 SQL로 저장을 선택합니다.	DBA
AWS SCT에서 생성한 SQL 스크립트를 수정합니다.		DBA

Amazon RDS DB 인스턴스 생성 및 구성

작업	설명	필요한 기술
Amazon RDS DB 인스턴스를 생성합니다.	Amazon RDS 콘솔에서 새 PostgreSQL DB 인스턴스를 생성합니다.	AWS SysAdmin, DBA
DB 인스턴스를 구성합니다.	DB 엔진 버전, DB 인스턴스 클래스, 다중 AZ 배포, 스토리지 유형, 할당된 스토리지를 지정합니다. DB 인스턴스 식별자, 마스터 사용자 이름, 마스터 암호를 입력합니다.	AWS SysAdmin, DBA
네트워크 및 보안을 구성합니다.	VPC, 서브넷 그룹, 퍼블릭 액세스 가능성, 가용 영역 기본 설정, 보안 그룹을 지정합니다.	AWS SysAdmin, DBA

작업	설명	필요한 기술
데이터베이스 옵션을 구성합니다.	데이터베이스 이름, 포트, 파라미터 그룹, 암호화, 마스터 키를 지정합니다.	AWS SysAdmin, DBA
백업을 구성합니다.	백업 보존 기간, 백업 기간, 시작 시간, 기간 및 스냅샷에 태그를 복사할지 여부를 지정합니다.	AWS SysAdmin, DBA
모니터링 옵션을 구성합니다.	향상된 모니터링 및 성능 개선 도우미를 활성화하거나 비활성화합니다.	AWS SysAdmin, DBA
유지 관리 옵션을 구성합니다.	마이너 버전 자동 업그레이드, 유지 관리 기간, 시작 날짜, 시간 및 기간을 지정합니다.	AWS SysAdmin, DBA
AWS SCT에서 마이그레이션 전 스크립트를 실행합니다.	Amazon RDS 인스턴스에서 create_sequence.sql, create_table.sql, create_view.sql, create_function.sql, create_function.sql 스크립트를 실행합니다.	AWS SysAdmin, DBA

AWS DMS를 이용한 데이터 마이그레이션

작업	설명	필요한 기술
AWS DMS에 복제 인스턴스를 생성합니다.	이름, 인스턴스 클래스, VPC(EC2 인스턴스와 동일), 다중 AZ, 퍼블릭 액세스 가능성 필드를 입력합니다. 고급 구성 섹션에서 할당된 스토리지, 서브넷 그룹, 가용 영역, VPC 보안 그룹, AWS Key	AWS SysAdmin, DBA

작업	설명	필요한 기술
	<p>Management Service(AWS KMS) 루트 키를 지정합니다.</p>	
<p>소스 데이터베이스 엔드포인트를 생성합니다.</p>	<p>엔드포인트 이름, 유형, 소스 엔진(Oracle), 서버 이름(Amazon EC2 프라이빗 DNS 이름), 포트, SSL 모드, 사용자 이름, 암호, SID, VPC(복제 인스턴스가 있는 VPC 지정) 및 복제 인스턴스를 지정합니다. 연결을 테스트하려면 테스트 실행 선택한 다음 엔드포인트를 생성합니다. 다음과 같은 고급 설정 maxFileSize 및 numberDataTypes 스케일을 구성할 수도 있습니다.</p>	<p>AWS SysAdmin, DBA</p>
<p>AWS DMS 복제 작업을 생성합니다.</p>	<p>작업 이름, 복제 인스턴스, 소스 및 대상 엔드포인트, 복제 인스턴스를 지정합니다. 마이그레이션 유형에서 "기존 데이터 마이그레이션 및 진행 중인 변경 사항 복제"를 선택합니다. "생성 시 작업 시작" 확인란을 선택 취소합니다.</p>	<p>AWS SysAdmin, DBA</p>
<p>AWS DMS 복제 작업 설정을 구성합니다.</p>	<p>대상 테이블 준비 모드에서 "아무 작업 안 함"을 선택합니다. 전체 로드가 완료된 후 작업을 중지하여 프라이머리 키를 생성합니다. 제한 또는 전체 LOB 모드를 지정한 다음 제어 테이블을 활성화합니다. 선택적으로 CommitRate 고급 설정을 구성할 수 있습니다.</p>	<p>DBA</p>

작업	설명	필요한 기술
테이블 매핑을 구성합니다.	테이블 매핑 섹션에서 마이그레이션에 포함된 모든 스키마의 모든 테이블에 대한 Include 규칙을 생성한 다음 Exclude 규칙을 생성합니다. 스키마, 테이블 및 열 이름을 소문자로 변환하는 세 가지 변환 규칙을 추가하고, 이 특정 마이그레이션에 필요한 다른 규칙을 추가합니다.	DBA
작업을 시작합니다.	복제 작업을 시작합니다. 전체 로드가 실행 중인지 확인합니다. 기본 Oracle 데이터베이스에서 ALTER SYSTEM SWITCH LOGFILE을 실행하여 작업을 시작합니다.	DBA
AWS SCT에서 마이그레이션 중 스크립트를 실행합니다.	Amazon RDS for PostgreSQL에서 create_index.sql 및 create_constraint.sql 스크립트를 실행합니다.	DBA
변경 데이터 캡처(CDC)를 계속하려면 작업을 다시 시작합니다.	Amazon RDS for PostgreSQL DB 인스턴스에서 VACUUM을 실행하고 AWS DMS 작업을 다시 시작하여 캐시된 CDC 변경 사항을 적용합니다.	DBA

PostgreSQL 데이터베이스로 컷오버

작업	설명	필요한 기술
AWS DMS 로그 및 메타데이터 테이블을 확인합니다.	오류를 검증하고 필요한 경우 수정합니다.	DBA
모든 Oracle 종속성을 중지합니다.	Oracle 데이터베이스의 리스너를 종료하고 ALTER SYSTEM SWITCH LOGFILE을 실행합니다. 활동이 표시되지 않으면 AWS DMS 작업을 중지합니다.	DBA
AWS SCT에서 마이그레이션 후 스크립트를 실행합니다.	Amazon RDS for PostgreSQL에서는 create_foreign_key_constraint.sql 및 create_triggers.sql 스크립트를 실행합니다.	DBA
추가 Amazon RDS for PostgreSQL 단계를 완료합니다.	필요한 경우 Oracle과 일치하도록 시퀀스를 늘리고, VACUUM 및 ANALYZE를 실행하고, 규정 준수를 위한 스냅샷을 만듭니다.	DBA
Amazon RDS for PostgreSQL에 대한 연결을 엽니다.	Amazon RDS for PostgreSQL에서 AWS DMS 보안 그룹을 제거하고, 프로덕션 보안 그룹을 추가하여 애플리케이션이 새 데이터베이스를 가리키도록 합니다.	DBA
AWS DMS 리소스를 정리합니다.	엔드포인트, 복제 작업, 복제 인스턴스, EC2 인스턴스를 제거합니다.	SysAdmin, DBA

관련 리소스

- [AWS DMS 설명서](#)
- [AWS SCT 설명서](#)
- [Amazon RDS for PostgreSQL 요금](#)
- [Oracle 데이터베이스를 AWS DMS의 소스로 사용](#)
- [PostgreSQL 데이터베이스를 AWS DMS 대상으로 사용](#)
- [퀘스트 문서 SharePlex](#)

구체화된 뷰와 AWS DMS를 사용하여 Oracle 8i 또는 9i에서 Amazon RDS for PostgreSQL로 마이그레이션

작성자: Kumar Babu P G(AWS), Pragnesh Patel(AWS)

환경: PoC 또는 파일럿	소스: Oracle 8i 또는 9i	대상: Amazon RDS for PostgreSQL 또는 Aurora PostgreSQL 호환
R 유형: 리아키텍트	워크로드: Oracle	기술: 마이그레이션, 데이터베이스

AWS 서비스: Amazon RDS, Amazon Aurora

요약

이 패턴은 온프레미스 레거시 Oracle 8i 또는 9i 데이터베이스를 Amazon Relational Database Service(RDS) for PostgreSQL 또는 Amazon Aurora PostgreSQL 호환 버전으로 마이그레이션하는 방법을 설명합니다.

AWS Database Migration Service(AWS DMS)는 Oracle 8i 또는 9i를 소스로 지원하지 않으므로 이 패턴은 Oracle 10g 또는 11g와 같은 AWS DMS와 호환 가능한 중간 Oracle 데이터베이스 인스턴스를 사용합니다. 또한 구체화된 뷰 기능을 사용하여 소스 Oracle 8i/9i 인스턴스에서 중간 Oracle 10g/11g 인스턴스로 데이터를 마이그레이션합니다.

AWS Schema Conversion Tool(AWS SCT)은 데이터베이스 스키마를 변환하고, AWS DMS는 데이터를 대상 PostgreSQL 데이터베이스로 마이그레이션합니다.

이 패턴은 데이터베이스 다운타임을 최소화하면서 기존 Oracle 데이터베이스에서 마이그레이션하려는 사용자에게 도움이 됩니다. 이 구현에서 가동 중지 시간은 대상 데이터베이스에서 모든 외부 키, 트리거 및 시퀀스를 생성하거나 검증하는 데 걸리는 시간으로 제한됩니다.

이 패턴은 Amazon Elastic Compute Cloud(Amazon EC2) 인스턴스를 설치된 Oracle 10g/11g 데이터베이스와 함께 사용하여 AWS DMS를 통해 데이터를 스트림하도록 도와줍니다. 온프레미스 Oracle 데이터베이스에서 중간 Oracle 인스턴스로의 스트리밍 복제를 일시적으로 일시 중지하여 AWS DMS가 데이터 검증을 따라잡거나 다른 데이터 검증 도구를 사용하도록 할 수 있습니다. AWS DMS가 현재 변

경 사항의 마이그레이션을 완료하면 PostgreSQL DB 인스턴스와 중간 Oracle 데이터베이스가 동일한 데이터를 갖게 됩니다.

사전 조건 및 제한 사항

사전 조건

- 활성 상태의 AWS 계정
- 온프레미스 데이터 센터의 소스 Oracle 8i 또는 9i 데이터베이스
- 온프레미스 데이터 센터와 AWS 간에 구성된 AWS Direct Connect
- AWS SCT가 설치된 로컬 시스템 또는 EC2 인스턴스에 설치된 AWS SCT 커넥터용 Java Database Connectivity(JDBC) 드라이버
- [Oracle 데이터베이스를 AWS DMS 소스로 사용](#) 속지
- [PostgreSQL 데이터베이스를 AWS DMS 대상으로 사용](#) 속지

제한 사항

- 데이터베이스 크기 제한: 64TB

제품 버전

- 소스 데이터베이스: Oracle 8i 또는 9i
- 중간 데이터베이스: Oracle 10g 또는 11g
- PostgreSQL 10.17 이상

아키텍처

소스 기술 스택

- Oracle 8i 또는 9i 데이터베이스

대상 기술 스택

- Amazon RDS for PostgreSQL 또는 Aurora PostgreSQL 호환

대상 아키텍처

도구

- [AWS DMS](#)는 데이터베이스를 빠르고 안전하게 마이그레이션하는 데 도움이 됩니다. 소스 데이터베이스는 마이그레이션 중에도 완전히 작동하여 데이터베이스를 사용하는 애플리케이션의 가동 중지 시간을 최소화합니다. AWS DMS는 광범위하게 사용되는 상용 및 오픈 소스 데이터베이스 간에 데이터를 마이그레이션할 수 있습니다.
- [AWS SCT](#)는 소스 데이터베이스 스키마와 대부분의 데이터베이스 코드 객체(뷰, 저장 프로시저, 함수 등)를 대상 데이터베이스와 호환되는 형식으로 자동 변환합니다. 자동으로 변환할 수 없는 객체는 명확하게 표시되므로 수동으로 변환하여 마이그레이션을 완료할 수 있습니다. AWS SCT는 애플리케이션 소스 코드에서 내장된 SQL 문을 스캔하고 이를 데이터베이스 스키마 변환 프로젝트의 일부로 변환할 수도 있습니다. 이 프로세스 중에 AWS SCT는 기존 Oracle 및 SQL Server 함수를 이와 동등한 AWS 함수로 변환하여 클라우드 네이티브 코드 최적화를 수행하므로 데이터베이스를 마이그레이션하는 동시에 애플리케이션을 현대화할 수 있습니다. 스키마 변환이 완료되면 AWS SCT는 내장된 데이터 마이그레이션 에이전트를 사용하여 다양한 데이터 웨어하우스에서 Amazon Redshift로 데이터를 마이그레이션하는 데 도움을 줄 수 있습니다.

모범 사례

구체화된 뷰를 새로 고치는 모범 사례는 다음 Oracle 설명서를 참조하세요.

- [구체화된 뷰 새로 고침](#)
- [구체화된 뷰의 빠른 새로 고침](#)

에픽

EC2 인스턴스에 Oracle 설치 및 구체화된 뷰 생성

작업	설명	필요한 기술
EC2 인스턴스용 네트워크를 설정합니다.	Virtual Private Cloud(VPC), 서브넷, 인터넷 게이트웨이, 라우팅 테이블, 보안 그룹을 생성합니다.	AWS SysAdmin

작업	설명	필요한 기술
EC2 인스턴스를 생성하십시오.	EC2 인스턴스의 Amazon Machine Image(AMI)를 선택합니다. 인스턴스 크기를 선택하고 인스턴스 세부 정보(인스턴스 수(1), 이전 단계의 VPC 및 서브넷, 퍼블릭 IP 자동 할당 및 기타 옵션)를 구성합니다. 스토리지를 추가하고, 보안 그룹을 구성하여 인스턴스를 시작합니다. 메시지가 표시되면 다음 단계를 위해 키 페어를 생성하고 저장합니다.	AWS SysAdmin
EC2 인스턴스에 Oracle을 설치합니다.	라이선스와 필요한 Oracle 바이너리를 가져오고 EC2 인스턴스에 Oracle 10g 또는 11g를 설치합니다.	DBA
Oracle 네트워킹을 구성합니다.	listener.ora 의 항목을 수정하거나 추가하여 온프레미스 소스 Oracle 8i/9i 데이터베이스에 연결한 다음 데이터베이스 링크를 생성합니다.	DBA
구체화된 뷰를 생성합니다.	소스 Oracle 8i/9i 데이터베이스에서 복제할 데이터베이스 객체를 식별한 다음 데이터베이스 링크를 사용하여 모든 객체에 대한 구체화된 뷰를 생성합니다.	DBA

작업	설명	필요한 기술
스크립트를 배포하여 필요한 간격으로 구체화된 뷰를 새로 고칩니다.	Amazon EC2 Oracle 10g/11g 인스턴스에서 필요한 간격으로 구체화된 뷰를 새로 고치는 스크립트를 개발하고 배포합니다. 증분 새로 고침 옵션을 사용하여 구체화된 뷰를 새로 고칩니다.	DBA

Oracle 데이터베이스 스키마를 PostgreSQL로 변환

작업	설명	필요한 기술
AWS SCT를 설정합니다.	새 보고서를 생성한 다음 Oracle에 소스로 연결하고 PostgreSQL에 대상으로 연결합니다. 프로젝트 설정에서 SQL 스크립팅 탭을 엽니다. 대상 SQL 스크립트를 여러 파일로 변경합니다. (AWS SCT는 Oracle 8i/9i 데이터베이스를 지원하지 않으므로 중간 Oracle 10g/11g 인스턴스에서 스키마 전용 덤프를 복원하고 이를 AWS SCT의 소스로 사용해야 합니다.)	DBA
Oracle 데이터베이스 스키마를 변환합니다.	작업 탭에서 보고서 생성, 스키마 변환을 선택한 다음 SQL로 저장을 선택합니다.	DBA
SQL 스크립트를 수정합니다.	모범 사례에 따라 수정합니다. 예를 들어 적합한 데이터 유형으로 전환하고 Oracle별 함수	DBA, DevDBA

작업	설명	필요한 기술
	에 해당하는 PostgreSQL을 개발합니다.	

Amazon RDS DB 인스턴스를 생성 및 구성하여 변환된 데이터베이스 호스팅

작업	설명	필요한 기술
Amazon RDS DB 인스턴스를 생성합니다.	Amazon RDS 콘솔에서 새 PostgreSQL DB 인스턴스를 생성합니다.	AWS SysAdmin, DBA
DB 인스턴스를 구성합니다.	DB 엔진 버전, DB 인스턴스 클래스, 다중 AZ 배포, 스토리지 유형, 할당된 스토리지를 지정합니다. DB 인스턴스 식별자, 마스터 사용자 이름, 마스터 암호를 입력합니다.	AWS SysAdmin, DBA
네트워크 및 보안을 구성합니다.	VPC, 서브넷 그룹, 퍼블릭 액세스 가능성, 가용 영역 기본 설정, 보안 그룹을 지정합니다.	DBA, SysAdmin
데이터베이스 옵션을 구성합니다.	데이터베이스 이름, 포트, 파라미터 그룹, 암호화, 마스터 키를 지정합니다.	DBA, AWS SysAdmin
백업을 구성합니다.	백업 보존 기간, 백업 기간, 시작 시간, 기간 및 스냅샷에 태그를 복사할지 여부를 지정합니다.	AWS SysAdmin, DBA
모니터링 옵션을 구성합니다.	향상된 모니터링 및 성능 개선 도우미를 활성화하거나 비활성화합니다.	AWS SysAdmin, DBA

작업	설명	필요한 기술
유지 관리 옵션을 구성합니다.	마이너 버전 자동 업그레이드, 유지 관리 기간, 시작 날짜, 시간 및 기간을 지정합니다.	AWS SysAdmin, DBA
AWS SCT에서 마이그레이션 전 스크립트를 실행합니다.	대상 Amazon RDS for PostgreSQL 인스턴스에서 다른 수정 사항과 함께 AWS SCT의 SQL 스크립트를 사용하여 데이터베이스 스키마를 생성합니다. 여기에는 사용자 생성, 데이터베이스 생성, 스키마 생성, 테이블, 뷰, 함수 및 기타 코드 객체를 포함하여 여러 스크립트를 실행하는 것이 포함될 수 있습니다.	AWS SysAdmin, DBA

AWS DMS를 이용한 데이터 마이그레이션

작업	설명	필요한 기술
AWS DMS에 복제 인스턴스를 생성합니다.	이름, 인스턴스 클래스, VPC(EC2 인스턴스와 동일), 다중 AZ, 퍼블릭 액세스 가능성 필드를 입력합니다. 고급 구성 섹션에서 할당된 스토리지, 서브넷 그룹, 가용 영역, VPC 보안 그룹, AWS Key Management Service(AWS KMS) 키를 지정합니다.	AWS SysAdmin, DBA
소스 데이터베이스 엔드포인트를 생성합니다.	엔드포인트 이름, 유형, 소스 엔진(Oracle), 서버 이름(EC2 인스턴스의 프라이빗 DNS 이름), 포트, SSL 모드, 사용자 이름, 암호, SID, VPC(복제 인스턴	AWS SysAdmin, DBA

작업	설명	필요한 기술
	<p>스가 있는 VPC 지정) 및 복제 인스턴스를 지정합니다. 연결을 테스트하려면 테스트 실행 선택한 다음 엔드포인트를 생성합니다. 다음과 같은 고급 설정 maxFileSize 및 numberDataType 스케일을 구성할 수도 있습니다.</p>	
<p>Amazon RDS for PostgreSQL에 AWS DMS를 연결합니다.</p>	<p>PostgreSQL 데이터베이스가 다른 VPC에 있는 경우 VPC 간 연결을 위한 마이그레이션 보안 그룹을 생성합니다.</p>	<p>AWS SysAdmin, DBA</p>
<p>대상 데이터베이스 엔드포인트를 생성합니다.</p>	<p>엔드포인트 이름, 유형, 소스 엔진(PostgreSQL), 서버 이름(Amazon RDS 엔드포인트), 포트, SSL 모드, 사용자 이름, 암호, 데이터베이스 이름, VPC(복제 인스턴스가 있는 VPC 지정) 및 복제 인스턴스를 지정합니다. 연결을 테스트하려면 테스트 실행 선택한 다음 엔드포인트를 생성합니다. 다음과 같은 고급 설정 maxFileSize 및 numberDataType 스케일을 구성할 수도 있습니다.</p>	<p>AWS SysAdmin, DBA</p>

작업	설명	필요한 기술
AWS DMS 복제 작업을 생성합니다.	작업 이름, 복제 인스턴스, 소스 및 대상 엔드포인트, 복제 인스턴스를 지정합니다. 마이그레이션 유형에서 기존 데이터 마이그레이션 및 진행 중인 변경 사항 복제를 선택합니다. 생성 시 작업 시작 확인란을 선택 취소합니다.	AWS SysAdmin, DBA
AWS DMS 복제 작업 설정을 구성합니다.	대상 테이블 준비 모드에서 아무 작업 안 함을 선택합니다. 전체 로드 완료 후 작업을 중지(프라이머리 키를 생성)합니다. 제한 또는 전체 LOB 모드를 지정한 다음 제어 테이블을 활성화합니다. 선택적으로 CommitRate 고급 설정을 구성할 수 있습니다.	DBA
테이블 매핑을 구성합니다.	테이블 매핑 섹션에서 마이그레이션에 포함된 모든 스키마의 모든 테이블에 대한 포함 규칙을 생성한 다음 제외 규칙을 생성합니다. 스키마, 테이블 및 열 이름을 소문자로 변환하는 세 가지 변환 규칙을 추가하고, 이 특정 마이그레이션에 필요한 다른 규칙을 추가합니다.	DBA
작업을 시작합니다.	복제 작업을 시작합니다. 전체 로드가 실행 중인지 확인합니다. 기본 Oracle 데이터베이스에서 ALTER SYSTEM SWITCH LOGFILE을 실행하여 작업을 시작합니다.	DBA

작업	설명	필요한 기술
AWS SCT에서 마이그레이션 중 스크립트를 실행합니다.	Amazon RDS for PostgreSQL에서 <code>create_index.sql</code> 및 <code>create_constraint.sql</code> 스크립트를 실행합니다 (전체 스키마가 처음에 생성되지 않은 경우).	DBA
변경 데이터 캡처(CDC)를 계속하려면 작업을 재개합니다.	Amazon RDS for PostgreSQL DB 인스턴스에서 VACUUM을 실행하고 AWS DMS 작업을 다시 시작하여 캐시된 CDC 변경 사항을 적용합니다.	DBA

PostgreSQL 데이터베이스로 컷오버

작업	설명	필요한 기술
AWS DMS 로그 및 검증 테이블을 확인합니다.	복제 또는 검증 오류를 확인하고 수정합니다.	DBA
온프레미스 Oracle 데이터베이스와 해당 종속성 사용을 중단합니다.	모든 Oracle 종속성을 중지하고, Oracle 데이터베이스의 리스너를 종료한 다음, ALTER SYSTEM SWITCH LOGFILE을 실행합니다. 활동이 표시되지 않으면 AWS DMS 작업을 중지합니다.	DBA
AWS SCT에서 마이그레이션 후 스크립트를 실행합니다.	Amazon RDS for PostgreSQL에서 <code>create_foreign_key_constraint.sql</code> and <code>create_triggers.sql</code> 스크립트를 실행합니다. 시퀀	DBA

작업	설명	필요한 기술
	스가 최신 상태인지 확인합니다.	
추가 Amazon RDS for PostgreSQL 단계를 완료합니다.	필요한 경우 Oracle과 일치하도록 시퀀스를 늘리고, VACUUM 및 ANALYZE을 실행하고, 규정 준수를 위한 스냅샷을 만듭니다.	DBA
Amazon RDS for PostgreSQL에 대한 연결을 엽니다.	Amazon RDS for PostgreSQL에서 AWS DMS 보안 그룹을 제거하고, 프로덕션 보안 그룹을 추가하여 애플리케이션이 새 데이터베이스를 가리키도록 합니다.	DBA
AWS DMS 객체를 정리합니다.	엔드포인트, 복제 작업, 복제 인스턴스, EC2 인스턴스를 제거합니다.	SysAdmin, DBA

관련 리소스

- [AWS DMS 설명서](#)
- [AWS SCT 설명서](#)
- [Amazon RDS for PostgreSQL 요금](#)
- [Oracle 데이터베이스를 AWS DMS의 소스로 사용](#)
- [PostgreSQL 데이터베이스를 AWS DMS의 대상으로 사용](#)

AWS DMS 및 AWS SCT를 사용하여 Amazon EC2의 오라클에서 Amazon RDS for MySQL로 마이그레이션하십시오.

아닐 쿠나프레디 (AWS) 와 하샤드 고힐이 제작했습니다.

환경: PoC 또는 파일럿	소스: 데이터베이스: 관계형	대상: Amazon RDS for MySQL
R 유형: 리아키텍트	워크로드: Oracle	기술: 마이그레이션, 데이터베이스
AWS 서비스: Amazon RDS		

요약

Amazon Elastic Compute Cloud(Amazon EC2) 인스턴스에서 Oracle 데이터베이스를 관리하려면 리소스가 필요하며 비용이 많이 들 수 있습니다. 이러한 데이터베이스를 Amazon Relational Database Service(RDS) for MySQL DB 인스턴스로 이동하면 전체 IT 예산을 최적화하여 작업을 쉽게 수행할 수 있습니다. Amazon RDS for MySQL은 다중 AZ, 확장성 및 자동 백업과 같은 기능도 제공합니다.

이 패턴은 Amazon EC2의 원본 Oracle 데이터베이스를 대상 Amazon RDS for MySQL DB 인스턴스로 마이그레이션하는 과정을 안내합니다. AWS Database Migration Service(AWS DMS)를 사용하여 데이터를 마이그레이션하고, AWS Schema Conversion Tool(AWS SCT)를 사용하여 원본 데이터베이스 스키마와 객체를 Amazon RDS for MySQL와 호환되는 형식으로 변환합니다.

사전 조건 및 제한 사항

사전 조건

- 활성 상태의 AWS 계정.
- ARCHIVELOG 모드에서 인스턴스 및 리스너 서비스가 실행되는 소스 데이터베이스입니다.
- 데이터 마이그레이션을 위한 충분한 스토리지가 있는 대상 Amazon RDS for MySQL 데이터베이스

제한 사항

- AWS DMS는 대상 데이터베이스에 스키마를 생성하지 않으므로 사용자가 직접 스키마를 생성해야 합니다. 대상에 스키마 이름이 이미 있어야 합니다. 소스 스키마의 테이블을 사용자/스키마로 가져오

면 AWS DMS가 이를 사용하여 대상 인스턴스에 연결합니다. 여러 스키마를 마이그레이션해야 하는 경우, 여러 복제 작업을 만들어야 합니다.

제품 버전

- 버전 10.2 이상, 11g 이상, 12.2 이하, 18c의 모든 Oracle 데이터베이스 버전이 해당됩니다. 지원되는 버전의 최신 목록은 [AWS DMS용 원본으로 Oracle 데이터베이스 사용 및 AWS DMS용 대상으로 MySQL 호환 데이터베이스 사용을 참조하십시오](#). 가장 종합적인 버전 및 기능 지원을 위해 최신 버전의 AWS DMS를 사용하는 것을 권장합니다. AWS SCT에서 지원하는 Oracle 데이터베이스 버전에 대한 자세한 내용은 [AWS SCT 설명서](#)를 참고하십시오.
- AWS DMS는 MySQL 버전 5.5, 5.6, 5.7을 지원합니다.

아키텍처

소스 기술 스택

- EC2 인스턴스의 Oracle 데이터베이스

대상 기술 스택

- Amazon RDS for MySQL DB 인스턴스

데이터 마이그레이션 아키텍처

소스 및 대상 아키텍처

도구

- AWS DMS - [AWS Database Migration Service](#)(AWS DMS)는 온프레미스, Amazon RDS DB 인스턴스 또는 EC2 인스턴스의 데이터베이스에 있는 데이터베이스에서 Amazon RDS for MySQL 또는 EC2 인스턴스와 같은 AWS 서비스의 데이터베이스로 데이터를 마이그레이션하는 데 사용할 수 있는 웹 서비스입니다. 또한 데이터베이스를 AWS 서비스에서 온 프레미스 데이터베이스로 마이그레이션할 수 있습니다. 이기종 또는 동종 데이터베이스 엔진 간에 데이터를 마이그레이션할 수 있습니다.

- AWS SCT — [AWS Schema Conversion Tool](#)(AWS SCT)을 사용하면 원본 데이터베이스 스키마와 대부분의 데이터베이스 코드 객체(보기, 저장 프로시저, 함수 등)를 대상 데이터베이스와 호환되는 형식으로 자동 변환하여 예측 가능한 이기종 데이터베이스 마이그레이션을 수행할 수 있습니다. AWS SCT를 사용하여 데이터베이스 스키마와 코드 객체를 변환한 후, AWS DMS를 사용하여 소스 데이터베이스의 데이터를 대상 데이터베이스로 마이그레이션하여 마이그레이션 프로젝트를 완료할 수 있습니다.

모범 사례

< 작성자: 참고 사항 삭제: 사용자가 이 패턴을 보다 효과적으로 구현하는 데 도움이 될 수 있는 지침 및 권장 사항 목록을 제공하십시오. >

에픽

마이그레이션 계획

작업	설명	필요한 기술
소스 및 대상 데이터베이스 버전과 엔진을 식별합니다.		DBA/개발자
DMS 복제 인스턴스를 식별합니다.		DBA/개발자
스토리지 유형 및 용량과 같은 스토리지 요구 사항을 식별합니다.		DBA/개발자
지연 시간 및 대역폭과 같은 네트워크 요구 사항을 식별합니다.		DBA/개발자
Oracle 호환성 목록 및 용량 요구 사항을 기반으로 소스 및 대상 서버 인스턴스의 하드웨어 요구 사항을 식별합니다.		DBA/개발자

작업	설명	필요한 기술
원본 및 대상 데이터베이스의 네트워크 액세스 보안 요구 사항을 확인합니다.		DBA/개발자
AWS SCT 및 오라클 드라이버를 설치합니다.		DBA/개발자
백업 전략을 결정합니다.		DBA/개발자
가용성 요구 사항을 결정합니다.		DBA/개발자
애플리케이션 마이그레이션 및 전환 전략을 식별하십시오.		DBA/개발자
용량, 스토리지, 네트워크 기능에 따라 적절한 DB 인스턴스 유형을 선택합니다.		DBA/개발자

환경을 구성합니다

작업	설명	필요한 기술
Virtual Private Cloud(VPC)를 생성합니다. 원본, 대상 및 복제 인스턴스는 동일한 VPC에 있어야 합니다. 동일한 가용 영역에 들 수 있는 것도 좋습니다.		개발자
데이터베이스 액세스에 필요한 보안 그룹을 만듭니다.		개발자
키 페어를 생성하고 구성합니다.		개발자

작업	설명	필요한 기술
서브넷, 가용 영역, CIDR 블록을 구성합니다.		개발자

소스 구성: EC2 인스턴스의 Oracle 데이터베이스

작업	설명	필요한 기술
필수 사용자 및 역할을 사용하여 Amazon EC2에 Oracle 데이터베이스를 설치합니다.		DBA
다음 열의 세 단계를 수행하여 EC2 인스턴스 외부에서 Oracle에 액세스하십시오.	<ol style="list-style-type: none"> 1. tnsnames에 있는 로컬 호스트를 Amazon EC2 공개 DNS로 변경합니다. 2. listener에 있는 로컬 호스트를 Amazon EC2 공개 DNS로 변경합니다. 3. 리스너를 중지한 후 다시 시작합니다. 	DBA
Amazon EC2를 다시 시작하면 퍼블릭 DNS가 변경됩니다. 반드시 'tnsnames' 및 '리스너'에서 Amazon EC2 퍼블릭 DNS를 업데이트하거나 엘라스틱 IP 주소를 사용하십시오.		DBA/개발자
복제 인스턴스와 필요한 클라이언트가 원본 데이터베이스에 액세스할 수 있도록 EC2 인스턴스 보안 그룹을 구성합니다.		DBA/개발자

대상 구성: Amazon RDS for MySQL 구성

작업	설명	필요한 기술
Amazon RDS for MySQL DB 인스턴스를 구성하고 시작합니다.		개발자
Amazon RDS for MySQL DB 인스턴스에서 필요한 테이블스페이스를 생성합니다.		DBA
복제 인스턴스와 필요한 클라이언트가 대상 데이터베이스에 액세스할 수 있도록 보안 그룹을 구성합니다.		개발자

AWS SCT를 구성하고 대상 데이터베이스에 스키마를 생성합니다.

작업	설명	필요한 기술
AWS SCT 및 오라클 드라이버를 설치합니다.		개발자
적절한 파라미터를 입력하고 소스 및 대상에 연결합니다.		개발자
스키마 변환 보고서를 생성합니다.		개발자
필요에 따라 코드와 스키마, 특히 테이블스페이스와 다음표를 수정하고 대상 데이터베이스에서 실행합니다.		개발자
데이터를 마이그레이션하기 전에 소스 대 타겟의 스키마를 검증하십시오.		개발자

AWS DMS를 사용하여 데이터를 마이그레이션합니다

작업	설명	필요한 기술
전체 로드 및 변경 데이터 캡처 (CDC) 또는 CDC만 사용하려면 추가 연결 속성을 설정해야 합니다.		개발자
AWS DMS 소스 Oracle 데이터베이스 정의에 지정된 사용자에게 필요한 모든 권한을 부여해야 합니다. 전체 목록은 https://docs.aws.amazon.com/dms/latest/userguide/CHAP_Source.Oracle.html#CHAP_Source.Oracle.Self-Managed 단원을 참조하십시오.		DBA/개발자
소스 데이터베이스에서 추가 로깅을 활성화합니다.		DBA/개발자
전체 로드 및 변경 데이터 캡처(CDC) 또는 CDC만 사용하려면 원본 데이터베이스에서 ARCHIVELOG 모드를 활성화하십시오.		DBA
소스 및 대상 엔드포인트를 생성하고 연결을 테스트합니다.		개발자
엔드포인트가 성공적으로 연결되면 복제 작업을 생성합니다.		개발자
연속 복제에 대한 변경 사항만 캡처하려면 작업에서 CDC만 (또는) 전체 로드+CDC를 선택		개발자

작업	설명	필요한 기술
하여 각각 연속 복제 (또는) 전체 로드와 진행 중인 변경 사항을 캡처합니다.		
복제 작업을 실행하고 Amazon CloudWatch 로그를 모니터링합니다.		개발자
소스 및 대상 데이터베이스의 데이터를 검증합니다.		개발자

애플리케이션을 마이그레이션하고 잘라내세요.

작업	설명	필요한 기술
애플리케이션 마이그레이션 전략의 단계를 따르세요.		DBA, 개발자, 앱 소유자
애플리케이션 컷오버/전환 전략의 단계를 따르세요.		DBA, 개발자, 앱 소유자

프로젝트 닫기

작업	설명	필요한 기술
소스 데이터베이스와 대상 데이터베이스의 스키마와 데이터의 유효성을 검사합니다.		DBA/개발자
마이그레이션 시간, 수동 대비 도구 비율, 비용 절감 등에 대한 지표를 수집하십시오.		DBA/개발자/ AppOwner
프로젝트 문서 및 아티팩트를 검토하십시오.		DBA/개발자/ AppOwner

작업	설명	필요한 기술
임시 AWS 리소스를 종료합니다.		DBA/개발자
프로젝트를 마무리하고 피드백을 제공하세요.		DBA/개발자/ AppOwner

관련 리소스

- [AWS DMS 설명서](#)
- [AWS DMS 웹사이트](#)
- [AWS DMS 블로그 게시물](#)
- [Strategies for Migrating Oracle Database to AWS](#)
- [Amazon RDS for Oracle FAQ](#)
- [Oracle FAQ](#)
- [Amazon EC2](#)
- [Amazon EC2 FAQ](#)
- [클라우드 컴퓨팅 환경에서의 Oracle 소프트웨어 라이선스](#)

AWS DMS를 사용하여 Oracle에서 Amazon DocumentDB로 마이그레이션

작성자: Sashikanta Pattanayak(AWS)

R 유형: 리아키텍트	소스: 데이터베이스: 관계형	대상: Amazon DocumentDB
작성자: AWS	환경: PoC 또는 파일럿	기술: 데이터베이스; 마이그레이션
워크로드: Oracle	AWS 서비스: Amazon DocumentDB	

요약

이 패턴은 AWS Database Migration Service(AWS DMS)를 사용하여 Oracle 데이터베이스를 Amazon DocumentDB(MongoDB 호환) 데이터베이스로 마이그레이션하기 위한 지침을 제공합니다. 이 접근 방식은 Oracle DB 인스턴스용 Amazon Relational Database Service(RDS) 뿐만 아니라 온프레미스 Oracle 소스 데이터베이스에도 적용할 수 있습니다. 이 패턴은 Amazon RDS Oracle DB 소스 인스턴스를 예로 사용합니다.

Amazon DocumentDB(MongoDB 호환)는 MongoDB 호환 완전 관리형 도큐먼트 데이터베이스 서비스로, JSON 데이터를 쉽게 저장, 쿼리 및 인덱싱할 수 있습니다.

이 패턴의 사용 사례는 Oracle 데이터베이스 테이블을 Amazon DocumentDB 컬렉션에 one-to-one 복제하는 것입니다. 이 패턴은 AWS DMS 복제 작업을 사용하여 Oracle 데이터베이스의 테이블 구조를 읽고, Amazon DocumentDB에서 해당 모음을 생성하고, 전체 로드 마이그레이션을 수행합니다. MongoDB에서와 마찬가지로 Amazon DocumentDB에서 데이터를 보고 쿼리할 수 있습니다.

사전 조건 및 제한 사항

사전 조건

- 활성 상태의 AWS 계정
- Oracle 데이터베이스 사용 방법 숙지
- Amazon DocumentDB 사용 방법 숙지
- Oracle 사용자의 경우 모든 테이블 권한 선택

- Amazon DocumentDB를 사용하는 경우 데이터를 덤프하는 데 필요한 권한

제한 사항

Amazon DocumentDB를 AWS DMS의 대상으로 사용할 때는 다음의 제한 사항이 적용됩니다.

- Amazon DocumentDB에서 모음 이름에는 달러 기호(\$)가 포함될 수 없습니다. 또한 데이터베이스 이름에는 어떤 Unicode 문자도 포함될 수 없습니다.
- AWS DMS는 여러 원본 테이블을 단일 Amazon DocumentDB 모음에 병합하는 기능을 지원하지 않습니다.
- AWS DMS가 프라이머리 키가 없는 원본 데이터에서 변경 사항을 처리할 때 해당 테이블의 모든 바이너리 라지 오브젝트(LOB) 열이 무시됩니다.
- 변경 테이블 옵션이 활성화되고 AWS DMS에 "_id"라는 원본 열이 나타나면 해당 열은 변경 테이블에 "_id"(2개의 밑줄)로 표시됩니다.
- 소스 엔드포인트로 Oracle을 선택하면 Oracle 원본에서 전체 보충 로깅을 활성화해야 합니다. 그렇지 않으면 변경되지 않은 원본의 열이 있을 경우에 Amazon DocumentDB에 데이터가 null 값으로 로드됩니다.

제품 버전

- Amazon RDS for Oracle 버전 11.2.0.3 이상
- AWS DMS 버전 3.1.3 이상(최신 버전 정보는 AWS DMS 설명서의 [Amazon DocumentDB를 AWS DMS의 대상으로 사용](#) 참조)

아키텍처

소스 기술 스택

- Amazon RDS for Oracle DB 인스턴스

대상 기술 스택

- Amazon DocumentDB

소스 및 대상 아키텍처

도구

- AWS DMS-[AWS Database Migration Service](#)(AWS DMS)는 소스 데이터 스토어에서 대상 데이터 스토어로 마이그레이션하는 데 사용할 수 있는 웹 서비스입니다. [AWS DMS 사용 설명서](#)에는 AWS DMS와 함께 사용할 수 있는 Oracle 소스 데이터베이스 버전 및 에디션이 명시되어 있습니다. 이 패턴과 관련된 추가 정보는 [Amazon DocumentDB를 AWS DMS의 대상으로 사용](#)을 참조합니다.
- Amazon EC2-[Amazon Elastic Compute Cloud](#)(Amazon EC2)는 AWS 클라우드에서 확장 가능한 컴퓨팅 용량을 제공합니다. Amazon DocumentDB 클러스터는 기본 Virtual Private Cloud(VPC)에서 실행되어야 합니다. Amazon DocumentDB 클러스터와 상호 작용하려면 Amazon DocumentDB 클러스터를 생성한 것과 동일한 AWS 리전의 기본 VPC로 EC2 인스턴스를 시작해야 합니다. 자세한 내용은 Amazon DocumentDB 설명서의 [Amazon EC2 인스턴스 시작](#)을 참조합니다.

에픽

마이그레이션 계획

작업	설명	필요한 기술
소스 및 타겟 데이터베이스 버전과 엔진을 확인합니다.		AWS 관리자
적절한 인스턴스 유형(용량, 스토리지 특성, 네트워크 특성)을 선택합니다.		AWS 관리자
소스 및 대상 데이터베이스의 네트워크 또는 호스트 액세스 보안 요구 사항을 식별합니다.		AWS 관리자
소스 및 대상 데이터베이스에 대한 아웃바운드 보안 그룹을 생성합니다.		AWS 관리자
Amazon DocumentDB용 EC2 인스턴스를 생성하고 구성합니다.		AWS 관리자

인프라 구성

작업	설명	필요한 기술
VPC 및 서브넷을 생성합니다.		AWS 관리자
보안 그룹 및 네트워크 액세스 제어 목록(ACL)을 생성합니다.		AWS 관리자
소스 Amazon RDS for Oracle 인스턴스를 구성하고 시작합니다.		AWS 관리자
Amazon DocumentDB 인스턴스를 구성하고 시작합니다.		AWS 관리자

소스 데이터베이스 준비

작업	설명	필요한 기술
연결 세부 정보를 사용하여 Oracle 데이터베이스를 연결할 수 있는지 확인합니다.		AWS 관리자
Oracle 사용자에게 모든 테이블 선택 권한이 있는지 확인합니다.		AWS 관리자

대상 데이터베이스를 준비합니다.

작업	설명	필요한 기술
적절한 인스턴스 클래스와 인스턴스 수를 선택하여 Amazon DocumentDB 클러스터를 생성합니다.		AWS 관리자

Amazon EC2 구성

작업	설명	필요한 기술
EC2 인스턴스를 구성합니다.	Amazon DocumentDB 클러스터와 상호 작용하려면 Amazon DocumentDB 클러스터를 생성한 것과 동일한 AWS 리전의 기본 VPC로 EC2 인스턴스를 시작해야 합니다. EC2 인스턴스의 AWS 지역, VPC, 가용 영역 및 서브넷을 구성합니다.	AWS 관리자
키 페어를 구성합니다.	퍼블릭/프라이빗 키 페어는 실행 후 안전하게 EC2 인스턴스에 연결할 수 있도록 합니다.	AWS 관리자
Bastion Host CIDR 범위를 설정합니다(선택 사항).	외부 Secure Shell(SSH)에서 Bastion Host 인스턴스에 액세스할 수 있는 CIDR IP 범위를 설정합니다.	AWS 관리자

데이터 마이그레이션 — 전체 로드

작업	설명	필요한 기술
AWS DMS 복제 인스턴스를 생성합니다.		AWS 관리자
소스 및 대상 DB 엔드포인트를 생성합니다.		AWS 관리자
전체 로드를 위한 AWS DMS 복제 작업을 생성합니다.		AWS 관리자

마이그레이션 테스트

작업	설명	필요한 기술
EC2 인스턴스를 통해 Amazon DocumentDB 클러스터에 연결합니다.		AWS 관리자
mongo 셸을 사용하여 클러스터에 연결합니다.	지침은 참조 및 도움말 섹션의 Amazon DocumentDB 링크를 참조합니다.	AWS 관리자
마이그레이션 결과를 확인합니다.		AWS 관리자

관련 리소스

- [AWS DMS의 작동 방식](#)
- [Amazon DocumentDB로 마이그레이션](#)
- [Amazon DocumentDB를 AWS DMS의 대상으로 사용](#)
- [Amazon DocumentDB 개요](#)
- [mongo 셸을 사용하여 클러스터에 액세스 및 사용](#)
- [오프라인 방법을 사용하여 MongoDB에서 Amazon DocumentDB로 마이그레이션\(블로그 게시물\)](#)
- [Amazon DocumentDB\(MongoDB 호환\)를 사용하여 대규모로 애플리케이션을 구축하고 관리하는 방법\(블로그 게시물\)](#)

AWS DMS 및 AWS SCT를 사용하여 Amazon EC2에서 Amazon RDS for MariaDB로 Oracle 데이터베이스 마이그레이션

제작: 비란자네올루 그란디(AWS)와 비노드 쿠마르(AWS)

환경: PoC 또는 파일럿	소스: 데이터베이스: 관계형	대상: Amazon RDS for MariaDB
R 유형: 리아키텍트	워크로드: Oracle	기술: 마이그레이션, 데이터베이스
AWS 서비스: Amazon RDS		

요약

이 패턴은 Amazon Elastic Compute Cloud(Amazon EC2) 인스턴스의 Oracle 데이터베이스를 Amazon Relational Database Service(RDS) for MariaDB DB 인스턴스로 마이그레이션하는 단계를 안내합니다. 패턴은 데이터 마이그레이션에는 AWS Data Migration Service(AWS DMS)를 사용하고 스키마 변환에는 AWS Schema Conversion Tool(AWS SCT)을 사용합니다.

EC2 인스턴스에서 Oracle 데이터베이스를 관리하려면 Amazon RDS에서 데이터베이스를 사용하는 것보다 더 많은 리소스가 필요하고 비용도 많이 듭니다. Amazon RDS를 사용하면 클라우드에서 관계형 데이터베이스를 쉽게 설치, 운영 및 크기 조정할 수 있습니다. Amazon RDS는 하드웨어 프로비저닝, 데이터베이스 설정, 패치 및 백업과 같은 시간 소모적인 관리 작업을 자동화하는 동시에 비용 효율적이고 크기 조정 가능한 용량을 제공합니다.

사전 조건 및 제한 사항

사전 조건

- 활성 상태의 AWS 계정.
- 인스턴스 및 리스너 서비스가 가동되고 실행되는 소스 Oracle 데이터베이스 이 데이터베이스는 ARCHIVELOG 모드여야 합니다.
- [Oracle 데이터베이스를 AWS DMS 소스로 사용](#) 속지
- [Oracle을 AWS SCT 소스로 사용](#) 속지

제한 사항

- 데이터베이스 크기 제한: 64TB

제품 버전

- 버전 10.2 이상, 11g 이상, 12.2 이하, 18c의 모든 Oracle 데이터베이스 버전이 해당됩니다. 지원되는 버전의 최신 목록은 AWS 설명서의 [AWS DMS용 소스로 Oracle 데이터베이스 사용](#) 및 [AWS SCT 버전 표](#)를 참조하세요.
- Amazon RDS는 MariaDB Server Community Server 버전 10.3, 10.4, 10.5 및 10.6을 지원합니다. 지원되는 최신 버전 목록은 [Amazon RDS 설명서](#)를 참조하세요.

아키텍처

소스 기술 스택

- EC2 인스턴스의 Oracle 데이터베이스

대상 기술 스택

- Amazon RDS for MariaDB

데이터 마이그레이션 아키텍처

대상 아키텍처

도구

- [AWS Schema Conversion Tool](#)(AWS SCT)은 소스 데이터베이스 스키마와 대부분의 데이터베이스 코드 객체(보기, 저장된 절차, 기능 등)를 대상 데이터베이스와 호환되는 형식으로 자동 변환하여 이 기존 데이터베이스 마이그레이션을 지원합니다. AWS SCT를 사용하여 데이터베이스 스키마와 코드 객체를 변환한 후, AWS DMS를 사용하여 소스 데이터베이스의 데이터를 대상 데이터베이스로 마이그레이션하여 마이그레이션 프로젝트를 완료할 수 있습니다. 자세한 내용은 AWS SCT 설명서에서 [Oracle을 AWS SCT의 소스로 사용](#)을 참조하세요.

- [AWS Database Migration Service\(AWS DMS\)](#)를 사용하면 데이터베이스를 AWS로 빠르고 안전하게 마이그레이션할 수 있습니다. 소스 데이터베이스는 마이그레이션 중에도 완전히 작동하여 데이터베이스를 사용하는 애플리케이션의 가동 중지 시간을 최소화합니다. AWS DMS는 광범위하게 사용되는 상용 및 오픈 소스 데이터베이스 간에 데이터를 마이그레이션할 수 있습니다. AWS DMS는 Oracle에서 Oracle로의 동종 마이그레이션뿐만 아니라 Oracle 또는 Microsoft SQL Server에서 Amazon Aurora로 등 서로 다른 데이터베이스 플랫폼 간 이기종 마이그레이션도 지원합니다. Oracle 데이터베이스를 마이그레이션하는 방법에 대해 자세히 알아보려면 AWS DMS 설명서의 [AWS DMS 용 소스로 Oracle 데이터베이스 사용](#)을 참조하세요.

에픽

마이그레이션 계획

작업	설명	필요한 기술
버전 및 데이터베이스 엔진을 식별합니다.	소스 및 대상 데이터베이스 버전과 엔진을 식별합니다.	DBA, 개발자
복제 인스턴스를 식별합니다.	AWS DMS 복제 인스턴스를 식별합니다.	DBA, 개발자
스토리지 요구 사항을 식별합니다.	스토리지 유형 및 용량을 식별합니다.	DBA, 개발자
네트워크 요구 사항을 확인합니다.	네트워크 지연 시간 및 대역폭을 식별합니다.	DBA, 개발자
하드웨어 요구 사항을 식별합니다.	(Oracle 호환성 목록 및 용량 요구 사항을 기반으로) 소스 및 대상 서버 인스턴스의 하드웨어 요구 사항을 식별합니다.	DBA, 개발자
보안 요구 사항을 식별합니다.	소스 및 대상 데이터베이스의 네트워크 액세스 보안 요구 사항을 식별합니다.	DBA, 개발자
드라이버를 설치합니다.	최신 AWS SCT 및 Oracle 드라이버를 설치합니다.	DBA, 개발자

작업	설명	필요한 기술
백업 전략을 결정합니다.		DBA, 개발자
가용성 요구 사항을 결정합니다.		DBA, 개발자
애플리케이션 마이그레이션 전환 전략을 선택합니다.		DBA, 개발자
인스턴스 유형을 선택합니다.	용량, 스토리지 및 네트워크 기능에 따라 적절한 인스턴스 유형을 선택합니다.	DBA, 개발자

환경을 구성합니다.

작업	설명	필요한 기술
Virtual Private Cloud(VPC)를 생성합니다.	소스, 대상 및 복제 인스턴스는 동일한 VPC와 동일한 가용 영역에 있어야 합니다(권장).	개발자
보안 그룹을 생성합니다.	데이터베이스 액세스에 필요한 보안 그룹을 생성합니다.	개발자
키 페어를 생성합니다.	키 페어를 생성하고 구성합니다.	개발자
다른 리소스를 구성합니다.	서브넷, 가용 영역, CIDR 블록을 구성합니다.	개발자

소스를 구성합니다.

작업	설명	필요한 기술
EC2 인스턴스를 시작합니다.	자세한 지침은 Amazon EC2 설명서 를 참조하세요.	개발자

작업	설명	필요한 기술
Oracle 데이터베이스를 설치합니다.	필요한 사용자 및 역할과 함께 EC2 인스턴스에 Oracle 데이터베이스를 설치합니다.	DBA
작업 설명의 단계를 따라 EC2 인스턴스 외부에서 Oracle에 액세스합니다.	<ol style="list-style-type: none"> 1. tnsnames에 있는 로컬 호스트를 Amazon EC2 공개 DNS로 변경합니다. 2. listener에 있는 로컬 호스트를 Amazon EC2 공개 DNS로 변경합니다. 3. 리스너를 중지한 후 다시 시작합니다. 	DBA
Amazon EC2 퍼블릭 DNS를 업데이트합니다.	EC2 인스턴스가 재시작되면 퍼블릭 DNS가 변경됩니다. tnsnames 및 listener에서 Amazon EC2 퍼블릭 DNS를 업데이트하거나 탄력적 IP 주소를 사용해야 합니다.	DBA, 개발자
EC2 인스턴스 보안 그룹을 구성합니다.	복제 인스턴스와 필요한 클라이언트가 소스 데이터베이스에 액세스할 수 있도록 EC2 인스턴스 보안 그룹을 구성합니다.	DBA, 개발자

대상 Amazon RDS for MariaDB 환경을 구성합니다.

작업	설명	필요한 기술
RDS DB 인스턴스를 시작합니다.	Amazon RDS for MariaDB DB 인스턴스를 구성하고 시작합니다.	개발자

작업	설명	필요한 기술
테이블스페이스를 생성합니다.	Amazon RDS MariaDB 데이터베이스에 필요한 테이블스페이스를 생성합니다.	DBA
보안 그룹을 구성합니다.	복제 인스턴스와 필요한 클라이언트가 대상 데이터베이스에 액세스할 수 있도록 보안 그룹을 구성합니다.	개발자

AWS SCT를 구성합니다.

작업	설명	필요한 기술
드라이버를 설치합니다.	최신 AWS SCT 및 Oracle 드라이버를 설치합니다.	개발자
연결합니다.	적절한 파라미터를 입력한 다음 소스와 대상에 연결합니다.	개발자
스키마 변환 보고서를 생성합니다.	AWS SCT 스키마 전환 보고서를 생성합니다.	개발자
필요에 따라 코드와 스키마를 수정합니다.	코드와 스키마(특히 테이블스페이스 및 따옴표)를 필요에 따라 수정합니다.	DBA, 개발자
스키마를 검증합니다.	데이터를 로드하기 전에 대상의 스키마와 비교하여 소스 스키마를 검증합니다.	개발자

AWS DMS를 사용하여 데이터를 마이그레이션합니다

작업	설명	필요한 기술
연결 속성을 설정합니다.	전체 로드 및 변경 데이터 캡처 (CDC) 또는 CDC 전용에 대해 추가 연결 속성을 설정합니다. 자세한 내용은 Amazon RDS 설명서 를 참조하세요.	개발자
보충 로깅을 활성화합니다.	소스 데이터베이스에서 보충 로깅을 활성화합니다.	DBA, 개발자
아카이브 로그 모드를 활성화합니다.	전체 로드 및 CDC(또는 CDC 전용)의 경우 소스 데이터베이스에서 아카이브 로그 모드를 활성화합니다.	DBA
엔드포인트를 생성하고 테스트합니다.	소스 및 대상 엔드포인트를 생성하고 연결을 테스트합니다. 자세한 내용은 Amazon DMS 설명서 를 참조하세요.	개발자
복제 작업을 생성합니다.	엔드포인트가 성공적으로 연결되면 복제 작업을 생성합니다. 자세한 내용은 Amazon DMS 설명서 를 참조하세요.	개발자
복제 유형을 선택합니다.	태스크에서 CDC 전용 또는 전체 로드 및 CDC를 선택하여 지속 복제만을 위한 변경 사항 또는 전체 로드 및 진행 중인 변경에 대한 변경 사항을 각각 캡처합니다.	개발자
태스크를 시작하고 모니터링합니다.	복제 작업을 시작하고 Amazon CloudWatch 로그를 모니터링합니다. 자세한 내용은	개발자

작업	설명	필요한 기술
	Amazon DMS 설명서 를 참조하세요.	
데이터를 검증합니다.	소스 및 대상 데이터베이스의 데이터를 검증합니다.	개발자

애플리케이션을 마이그레이션하고 대상 데이터베이스로 전환합니다.

작업	설명	필요한 기술
선택한 애플리케이션 마이그레이션 전략을 따르세요.		DBA, 앱 소유자, 개발자
선택한 애플리케이션 컷오버, 전환 전략을 따릅니다.		DBA, 앱 소유자, 개발자

프로젝트 닫기

작업	설명	필요한 기술
스키마와 데이터를 검증합니다.	프로젝트가 종료되기 전에 스키마와 데이터가 대상과 비교하여 소스에서 성공적으로 검증되었는지 확인하세요.	DBA, 개발자
지표를 수집합니다.	마이그레이션 시간, 수동 작업과 도구 작업의 비율, 비용 절감 및 유사한 기준에 대한 지표를 수집합니다.	DBA, 앱 소유자, 개발자
설명서를 검토합니다.	프로젝트 문서 및 아티팩트를 검토하십시오.	DBA, 앱 소유자, 개발자
리소스를 종료합니다.	임시 AWS 리소스를 종료합니다.	DBA, 개발자

작업	설명	필요한 기술
프로젝트 닫기	마이그레이션 프로젝트를 종료하고 피드백을 제공합니다.	DBA, 앱 소유자, 개발자

관련 리소스

- [MariaDB Amazon RDS 개요](#)
- [Amazon RDS for MariaDB 제품 세부 정보](#)
- [Oracle 데이터베이스를 AWS DMS용 소스로 사용](#)
- [Oracle 데이터베이스를 AWS로 마이그레이션하는 전략](#)
- [클라우드 컴퓨팅 환경에서의 Oracle 소프트웨어 라이선스](#)
- [Amazon RDS for Oracle 자주 묻는 질문](#)
- [AWS DMS 개요](#)
- [AWS DMS 블로그 게시물](#)
- [Amazon EC2 개요](#)
- [Amazon EC2 FAQ](#)
- [AWS SCT 설명서](#)

DMS 및 SCT를 사용하여 온프레미스 Oracle 데이터베이스를 Amazon RDS for MySQL로 마이그레이션

작성자: Sergey Dmitriev(AWS)

R 유형: 리아키텍트	소스: 데이터베이스: 관계형	대상: Amazon RDS for MySQL
작성자: AWS	환경: PoC 또는 파일럿	기술: 데이터베이스; 마이그레이션
워크로드: Oracle	AWS 서비스: Amazon RDS	

요약

이 패턴은 온프레미스 Oracle 데이터베이스를 MySQL DB 인스턴스용 Amazon Relational Database Service(RDS)로 마이그레이션하는 절차를 안내합니다. AWS Database Migration Service(DMS)를 사용하여 데이터를 마이그레이션하고, AWS Schema Conversion Tool(SCT)를 사용하여 소스 데이터베이스 스키마와 객체를 Amazon RDS for MySQL과 호환되는 형식으로 변환합니다.

사전 조건 및 제한 사항

사전 조건

- 활성 상태의 AWS 계정
- 온프레미스 데이터 센터의 소스 Oracle 데이터베이스

제한 사항

- 데이터베이스 크기 제한: 64TB

제품 버전

- 버전 11g(버전 11.2.0.3.v1 이상) 및 12.2 까지, 18c에 대한 모든 Oracle 데이터베이스 에디션. 지원되는 버전의 최신 목록은 [Oracle 데이터베이스를 DMS용 소스로 사용하기](#) 섹션을 참조하십시오. 가장 종합적인 버전 및 기능 지원을 위해 최신 버전의 AWS DMS를 사용하는 것을 권장합니다. AWS SCT에서 지원하는 Oracle 데이터베이스 버전에 대한 자세한 내용은 [AWS SCT 설명서](#)를 참고하십시오.

- DMS는 현재 MySQL 버전 5.5, 5.6, 5.7을 지원합니다. 지원되는 버전의 최신 목록은 [MySQL 호환 데이터베이스를 DMS용 대상으로 사용](#)을 참조하십시오.

아키텍처

소스 기술 스택

- 온프레미스 Oracle 데이터베이스

대상 기술 스택

- Amazon RDS for MySQL DB 인스턴스

데이터 마이그레이션 아키텍처

도구

- DMS - [Database Migration Services](#)(DMS)를 사용하면 관계형 데이터베이스, 데이터 웨어하우스, NoSQL 데이터베이스 및 기타 유형의 데이터 스토어를 마이그레이션하는 데 도움이 됩니다. DMS를 사용하여 데이터를 온프레미스 인스턴스 간(AWS 클라우드 설정을 통해) 또는 클라우드와 온프레미스 데이터베이스 조합 간에 AWS Cloud로 마이그레이션할 수 있습니다.
- SCT - [Schema Conversion Tool](#)(SCT)는 기존 데이터베이스 스키마를 한 데이터베이스 엔진에서 다른 데이터베이스 엔진으로 변환하는 데 사용됩니다. 도구가 변환하는 사용자 지정 코드에는 보기, 저장된 절차 및 함수가 포함됩니다. 도구가 자동으로 변환할 수 없는 코드는 명확하게 표시되므로 직접 변환할 수 있습니다.

에픽

마이그레이션 계획

작업	설명	필요한 기술
소스 및 대상 데이터베이스 버전과 엔진의 유효성을 확인합니다.		DBA

작업	설명	필요한 기술
대상 서버 인스턴스의 하드웨어 요구 사항을 확인합니다.		DBA, SysAdmin
스토리지 요구 사항(스토리지 유형 및 용량)을 식별합니다.		DBA, SysAdmin
용량, 스토리지 기능, 네트워크 기능에 따라 적절한 인스턴스 유형을 선택합니다.		DBA, SysAdmin
소스 및 대상 데이터베이스의 네트워크 액세스 보안 요구 사항을 식별합니다.		DBA, SysAdmin
애플리케이션 마이그레이션 전략을 파악합니다.		DBA SysAdmin, 앱 소유자

인프라 구성

작업	설명	필요한 기술
Virtual Private Cloud(VPC) 및 서브넷을 생성합니다.		SysAdmin
보안 그룹 및 네트워크 액세스 제어 목록(ACL)을 생성합니다.		SysAdmin
Amazon RDS DB 인스턴스를 구성 및 시작합니다.		DBA, SysAdmin

데이터 마이그레이션

작업	설명	필요한 기술
SCT를 사용하여 데이터베이스 스키마를 마이그레이션합니다.		DBA
DMS를 사용하여 데이터 마이그레이션합니다.		DBA

애플리케이션 마이그레이션

작업	설명	필요한 기술
SCT를 사용하여 애플리케이션 코드 내부의 SQL을 분석하고 변환합니다.	자세한 내용은 https://docs.aws.amazon.com/ /Latest/ Userguide/CHAP_Converting.app.html 을 참조하십시오. SchemaConversionTool	앱 소유자
애플리케이션 마이그레이션 전략을 따릅니다.		DBA, 앱 소유자 SysAdmin

전환

작업	설명	필요한 기술
애플리케이션 클라이언트를 새 인프라로 전환합니다.		DBA SysAdmin, 앱 소유자

프로젝트 닫기

작업	설명	필요한 기술
임시 AWS 리소스를 종료합니다.		DBA, SysAdmin
프로젝트 문서를 검토하고 검증하세요.		DBA, SysAdmin
마이그레이션 시간, 수동 대비 도구 비율(%), 비용 절감 등에 대한 지표를 수집합니다.		DBA, SysAdmin
프로젝트를 마무리하고 피드백을 제공하세요.		

관련 리소스

참조

- [DMS 설명서](#)
- [SCT 설명서](#)
- [Amazon RDS 요금 책정](#)

자습서 및 동영상

- [DMS 시작하기](#)
- [Amazon RDS 시작](#)
- [DMS \(동영상\)](#)
- [Amazon RDS\(동영상\)](#)

Oracle bystander 및 AWS DMS를 사용하여 온프레미스 Oracle 데이터베이스를 Amazon RDS for PostgreSQL로 마이그레이션

작성자: Cady Motyka(AWS)

환경: PoC 또는 파일럿	소스: 데이터베이스: 관계형	대상: Amazon RDS for PostgreSQL / Amazon Aurora PostgreSQL
R 유형: 리아키텍트	워크로드: Oracle	기술: 마이그레이션, 데이터베이스
AWS 서비스: Amazon RDS		

요약

이 패턴은 가동 중지 시간을 최소화하면서 온프레미스 Oracle 데이터베이스를 다음 PostgreSQL 호환 AWS 데이터베이스 서비스 중 하나로 마이그레이션하는 방법을 설명합니다.

- Amazon Relational Database Service(RDS) for PostgreSQL
- Amazon Aurora PostgreSQL 호환 에디션

이 솔루션은 AWS Database Migration Service(AWS DMS)를 사용하여 데이터를 마이그레이션하고, AWS Schema Conversion Tool(AWS SCT)을 사용하여 데이터베이스 스키마를 변환하고, Oracle bystander 데이터베이스를 사용하여 마이그레이션을 관리합니다. 이 구현에서는 데이터베이스의 모든 외래 키를 생성하거나 검증하는 데 걸리는 시간으로 가동 중지 시간이 제한됩니다.

또한 이 솔루션은 Amazon Elastic Compute Cloud(Amazon EC2) 인스턴스를 Oracle bystander 데이터베이스와 함께 사용하여 AWS DMS를 통해 데이터 스트림을 제어합니다. 데이터 검증을 따라잡거나 다른 데이터 검증 도구를 사용하기 위해 AWS DMS를 활성화하기 위해 온프레미스 Oracle 데이터베이스에서 Oracle bystander로의 스트리밍 복제를 일시적으로 일시 중지할 수 있습니다. AWS DMS에서 현재 변경 사항 마이그레이션을 완료하면 Amazon RDS for PostgreSQL DB 인스턴스 또는 Aurora PostgreSQL 호환 DB 인스턴스와 bystander 데이터베이스는 동일한 데이터를 갖게 됩니다.

사전 조건 및 제한 사항

사전 조건

- 활성 상태의 AWS 계정
- 온프레미스 데이터 센터의 소스 Oracle 데이터베이스와 Active Data Guard 대기 데이터베이스가 구성된 경우
- 데이터베이스 보안 정보 저장을 위해 온프레미스 데이터 센터와 AWS Secrets Manager 간에 구성된 AWS Direct Connect
- AWS SCT가 설치된 로컬 머신 또는 EC2 인스턴스에 설치된 AWS SCT 커넥터용 Java Database Connectivity(JDBC) 드라이버
- [Oracle 데이터베이스를 AWS DMS 소스로 사용](#) 속지
- [PostgreSQL 데이터베이스를 AWS DMS의 대상으로 사용](#)하는 방법 속지

제한 사항

- 데이터베이스 크기 제한: 64TB

제품 버전

- AWS DMS는 버전 10.2 이상(버전 10.x의 경우), 11g 및 12.2 이하, 18c 및 19c의 모든 Oracle 데이터베이스 에디션을 지원합니다. 지원되는 버전의 최신 목록은 [Oracle 데이터베이스를 AWS DMS용 소스로 사용하기](#) 섹션을 참조하십시오. 가장 종합적인 버전 및 기능 지원을 위해 최신 버전의 AWS DMS를 사용하는 것을 권장합니다. AWS SCT에서 지원하는 Oracle 데이터베이스 버전에 대한 자세한 내용은 [AWS SCT 설명서](#)를 참고하십시오.
- AWS DMS는 9.4 이상(9.x 버전), 10.x, 11.x, 12.x 및 13.x 버전의 PostgreSQL을 지원합니다. 최신 정보는 AWS 설명서에서 [PostgreSQL 데이터베이스를 AWS DMS의 대상으로 사용하기 섹션](#)을 참조하십시오.

아키텍처

소스 기술 스택

- 온프레미스 Oracle 데이터베이스
- Oracle 데이터베이스의 bystander를 포함하는 EC2 인스턴스

대상 기술 스택

- Amazon RDS for PostgreSQL 또는 Aurora PostgreSQL 인스턴스, PostgreSQL 9.3 이상

대상 아키텍처

다음 다이어그램은 AWS DMS 및 Oracle bystander를 사용하여 Oracle 데이터베이스를 PostgreSQL 호환 AWS 데이터베이스로 마이그레이션하기 위한 예제 워크플로를 보여줍니다.

도구

- [AWS Database Migration Service\(AWS DMS\)](#)는 데이터 스토어를 AWS 클라우드로 마이그레이션하거나 클라우드와 온프레미스 설정의 조합 간에 마이그레이션하는 데 도움이 됩니다.
- [AWS Schema Conversion Tool\(AWS SCT\)](#)은 소스 데이터베이스 스키마와 대부분의 사용자 지정 코드를 대상 데이터베이스와 호환되는 형식으로 자동 변환하여 이기종 데이터베이스 마이그레이션을 지원합니다.
- [Amazon Relational Database Service\(RDS\)](#)는 AWS 클라우드에서 관계형 데이터베이스를 설정, 운영 및 규모를 조정하는 데 도움이 됩니다.

에픽

Oracle 데이터베이스 스키마를 PostgreSQL로 변환

작업	설명	필요한 기술
AWS SCT를 설정합니다.	<p>새 보고서를 생성한 다음 Oracle에 소스로 연결하고 PostgreSQL에 대상으로 연결합니다. 프로젝트 설정에서 SQL 스크립팅 탭으로 이동합니다. 대상 SQL 스크립트를 여러 파일로 변경합니다. 이러한 파일은 나중에 사용되며 이름은 다음과 같습니다.</p> <ul style="list-style-type: none"> • create_database.sql • create_sequence.sql • create_table.sql • create_view.sql 	DBA

작업	설명	필요한 기술
	<ul style="list-style-type: none"> create_function.sql 	
Oracle 데이터베이스 스키마를 변환합니다.	작업 탭에서 보고서 생성을 선택합니다. 그런 다음 스키마 변환을 선택하고 SQL로 저장을 선택합니다.	DBA
스크립트를 수정하십시오.	예를 들어 PostgreSQL에서 소스 스키마의 숫자를 숫자 형식으로 변환한 경우 스크립트를 수정하고 성능을 높이려면 BIGINT를 대신 사용해야 할 수 있습니다.	DBA

Amazon RDS DB 인스턴스 생성 및 구성

작업	설명	필요한 기술
Amazon RDS DB 인스턴스를 생성합니다.	올바른 AWS 리전에서 PostgreSQL DB 인스턴스를 새로 생성합니다. 이에 대한 자세한 내용은 Amazon RDS 설명서에서 PostgreSQL DB 인스턴스 생성 및 PostgreSQL DB 인스턴스의 데이터베이스에 연결 을 참고하십시오.	AWS SysAdmin, DBA
DB 인스턴스 사양을 구성합니다.	DB 엔진 버전, DB 인스턴스 클래스, 다중 AZ 배포, 스토리지 유형, 할당된 스토리지를 지정합니다. DB 인스턴스 식별자, 기본 사용자 이름, 기본 암호를 입력합니다.	AWS SysAdmin, DBA

작업	설명	필요한 기술
네트워크 및 보안을 구성합니다.	Virtual Private Cloud(VPC), 서브넷 그룹, 퍼블릭 액세스 가능성, 가용 영역 기본 설정, 보안 그룹을 지정합니다.	DBA, SysAdmin
데이터베이스 옵션을 구성합니다.	데이터베이스 이름, 포트, 파라미터 그룹, 암호화, KMS 키를 지정합니다.	AWS SysAdmin, DBA
백업을 구성합니다.	백업 보존 기간, 백업 기간, 시작 시간, 기간 및 스냅샷에 태그를 복사할지 여부를 지정합니다.	AWS SysAdmin, DBA
모니터링 옵션을 구성합니다.	향상된 모니터링 및 성능 인사이트를 활성화 또는 비활성화합니다.	AWS SysAdmin, DBA
유지 관리 옵션을 구성합니다.	자동 마이너 버전 업그레이드, 유지 관리 기간, 시작 요일, 시간 및 기간을 지정합니다.	AWS SysAdmin, DBA
AWS SCT에서 마이그레이션 전 스크립트를 실행합니다.	Amazon RDS 인스턴스에서, AWS SCT에서 생성한 다음 스크립트를 실행합니다. <ul style="list-style-type: none"> • create_database.sql • create_sequence.sql • create_table.sql • create_view.sql • create_function.sql 	AWS SysAdmin, DBA

Amazon EC2에서 Oracle Bystander 구성

작업	설명	필요한 기술
Amazon EC2용 네트워크를 설정합니다.	새 VPC, 서브넷, 인터넷 게이트웨이, 라우팅 테이블 및 보안 그룹을 생성합니다.	AWS SysAdmin
EC2 인스턴스를 생성하십시오.	적절한 AWS 리전에서 새 EC2 인스턴스를 생성합니다. Amazon Machine Image(AMI)를 선택하고, 인스턴스 크기를 선택한 다음, 인스턴스 세부 정보, 즉 인스턴스 수(1), 이전 작업에서 생성한 VPC 및 서브넷, 퍼블릭 IP 자동 할당 및 기타 옵션을 구성합니다. 스토리지를 추가하고, 보안 그룹을 구성하고, 시작합니다. 메시지가 표시되면 다음 단계를 위해 키 페어를 생성하고 저장합니다.	AWS SysAdmin
Oracle 소스 데이터베이스를 EC2 인스턴스에 연결합니다.	IPv4 퍼블릭 IP 주소와 DNS를 텍스트 파일에 복사하고 다음과 같이 SSH를 사용하여 연결합니다. <code>ssh -i "your_file.pem" EC2-사용자@<your-IP- -DNS>. address-or-public</code>	AWS SysAdmin
Amazon EC2에서 바이스탠더에 대한 초기 호스트를 설정합니다.	SSH 키, 배쉬 프로필, ORATAB 및 심볼 링크를 설정합니다. Oracle 디렉터리를 생성하십시오.	AWS SysAdmin, 리눅스 관리자
Amazon EC2에서 바이스탠더에 대한 데이터베이스 복사본 설정	RMAN을 사용하여 데이터베이스 사본을 생성하고, 추가 로깅을 활성화하고, 대기 제어 파일을 생성합니다. 복사가 완료되	AWS SysAdmin, DBA

작업	설명	필요한 기술
	면 데이터베이스를 복구 모드로 전환합니다.	
Oracle 데이터 가드 설정.	listener.ora 파일을 수정하고 리스너를 시작합니다. 새 아카이브 대상을 설정합니다. 방관자를 복구 모드로 설정하고, 향후 손상을 방지하기 위해 임시 파일을 교체하고, 필요한 경우 아카이브 디렉토리의 공간 부족을 방지하기 위해 crontab을 설치하고, 소스 및 대기 manage-trclog-files-oracle파일의.cfg 파일을 편집합니다.	AWS SysAdmin, DBA
배송을 동기화할 수 있도록 Oracle 데이터베이스를 준비하십시오.	대기 로그 파일을 추가하고 복구 모드를 변경합니다. 소스 프라이머리와 소스 대기 모두에서 로그 전달을 SYNC AFFIRM으로 변경합니다. 기본 로그로 전환하고 Amazon EC2 bystander 경고 로그를 통해 대기 로그 파일을 사용하고 있는지 확인하고 다시 실행 스트림이 SYNC로 흐르고 있는지 확인합니다.	AWS SysAdmin, DBA

AWS DMS를 사용하여 데이터 마이그레이션

작업	설명	필요한 기술
AWS DMS에 복제 인스턴스를 생성합니다.	이름, 인스턴스 클래스, VPC(Amazon EC2 인스턴스와 동일), 다중 AZ, 퍼블릭 액세스	AWS SysAdmin, DBA

작업	설명	필요한 기술
	<p>가능성 필드를 입력합니다. 고급에서 할당된 스토리지, 서브넷 그룹, 가용 영역, VPC 보안 그룹 및 AWS Key Management Service(AWS KMS) 키를 지정합니다.</p>	
<p>소스 데이터베이스 엔드포인트를 생성합니다.</p>	<p>엔드포인트 이름, 유형, 소스 엔진(Oracle), 서버 이름(Amazon EC2 프라이빗 DNS 이름), 포트, SSL 모드, 사용자 이름, 암호, SID, VPC(복제 인스턴스가 있는 VPC 지정) 및 복제 인스턴스를 지정합니다. 연결을 테스트하려면 테스트 실행 선택한 다음 엔드포인트를 생성합니다. 다음과 같은 고급 설정 maxFileSize 및 numberDataTypes 스케일을 구성할 수도 있습니다.</p>	<p>AWS SysAdmin, DBA</p>
<p>Amazon RDS for PostgreSQL에 AWS DMS를 연결합니다.</p>	<p>VPC 간 연결을 위한 마이그레이션 보안 그룹을 생성하십시오.</p>	<p>AWS SysAdmin, DBA</p>

작업	설명	필요한 기술
대상 데이터베이스 엔드포인트를 생성합니다.	엔드포인트 이름, 유형, 소스 엔진(PostgreSQL), 서버 이름(Amazon RDS 엔드포인트), 포트, SSL 모드, 사용자 이름, 암호, 데이터베이스 이름, VPC(복제 인스턴스가 있는 VPC 지정) 및 복제 인스턴스를 지정합니다. 연결을 테스트하려면 테스트 실행 선택한 다음 엔드포인트를 생성합니다. 다음과 같은 고급 설정 maxFileSize 및 numberDataType스케일을 구성할 수도 있습니다.	AWS SysAdmin, DBA
AWS DMS 복제 작업을 생성합니다.	작업 이름, 복제 인스턴스, 소스 및 대상 엔드포인트, 복제 인스턴스를 지정합니다. 마이그레이션 유형에서 기존 데이터 마이그레이션 및 진행 중인 변경 사항 복제를 선택합니다. 생성 시 작업 시작 확인란을 선택 취소합니다.	AWS SysAdmin, DBA
AWS DMS 복제 작업 설정을 구성합니다.	대상 테이블 준비 모드에서 아무 작업 안 함을 선택합니다. 전체 로드가 완료된 후 작업을 중지(프라이머리 키를 생성)합니다. 제한 또는 전체 LOB 모드를 지정하고 제어 테이블을 활성화합니다. 선택적으로 CommitRate고급 설정을 구성할 수 있습니다.	DBA

작업	설명	필요한 기술
테이블 매핑을 구성합니다.	테이블 매핑 섹션에서 마이그레이션에 포함된 모든 스키마의 모든 테이블에 대한 포함 규칙을 생성한 다음 제외 규칙을 생성합니다. 스키마, 테이블 및 열 이름을 소문자로 변환하는 세 가지 변환 규칙을 추가하고, 이 특정 마이그레이션에 필요한 다른 규칙을 추가합니다.	DBA
작업을 시작합니다.	복제 작업을 시작합니다. 전체 로드가 실행 중인지 확인합니다. 기본 Oracle 데이터베이스에서 ALTER SYSTEM SWITCH LOGFILE을 실행하여 작업을 시작합니다.	DBA
AWS SCT에서 마이그레이션 중 스크립트를 실행합니다.	Amazon RDS for PostgreSQL에서, AWS SCT에서 생성한 다음 스크립트를 실행합니다. <ul style="list-style-type: none"> • create_index.sql • create_constraint.sql 	DBA
변경 데이터 캡처(CDC)를 계속하려면 작업을 다시 시작합니다.	Amazon RDS for PostgreSQL DB 인스턴스에서 정리를 실행하고 AWS DMS 작업을 다시 시작하여 캐시된 CDC 변경 사항을 적용합니다.	DBA

PostgreSQL 데이터베이스로 컷오버

작업	설명	필요한 기술
AWS DMS 로그와 검증 테이블에 오류가 있는지 검토하십시오.	복제 또는 검증 오류를 확인하고 수정하십시오.	DBA
모든 Oracle 종속성을 중지합니다.	Oracle 종속성을 중지하고, Oracle 데이터베이스의 리스너를 종료한 다음, ALTER SYSTEM SWITCH LOGFILE을 실행합니다. 활동이 표시되지 않으면 AWS DMS 작업을 중지합니다.	DBA
AWS SCT에서 마이그레이션 후 스크립트를 실행합니다.	Amazon RDS for PostgreSQL에서, AWS SCT에서 생성한 다음 스크립트를 실행합니다. <ul style="list-style-type: none"> • create_foreign_key_constraint.sql • create_triggers.sql 	DBA
Amazon RDS for PostgreSQL의 추가 단계를 완료합니다.	필요한 경우 Oracle과 일치하도록 시퀀스를 늘리고, 정리 및 분석을 실행하고, 규정 준수를 위한 스냅샷을 만듭니다.	DBA
Amazon RDS for PostgreSQL에 대한 연결을 엽니다.	Amazon RDS for PostgreSQL에서 AWS DMS 보안 그룹을 제거하고, 프로덕션 보안 그룹을 추가하여 애플리케이션이 새 데이터베이스를 가리키도록 합니다.	DBA

작업	설명	필요한 기술
AWS DMS 객체를 정리합니다.	엔드포인트, 복제 작업, 복제 인스턴스, EC2 인스턴스를 제거합니다.	SysAdmin, DBA

관련 리소스

- [AWS DMS 설명서](#)
- [AWS SCT 설명서](#)
- [Amazon RDS for PostgreSQL 요금](#)

오라클을 사용하여 Oracle 데이터베이스에서 PostgreSQL용 Amazon RDS로 마이그레이션하십시오. GoldenGate

작성자: Dhairya Jindani(AWS), Rajeshkumar Sabankar(AWS), Sindhusa Paturu(AWS)

환경: PoC 또는 파일럿	소스: 데이터베이스: 관계형	대상: Amazon RDS for PostgreSQL
R 유형: 리아키텍트	워크로드: Oracle	기술: 마이그레이션, 데이터베이스
AWS 서비스: Amazon RDS		

요약

이 패턴은 Oracle 클라우드 인프라 (OCI) 를 사용하여 PostgreSQL용 아마존 관계형 데이터베이스 서비스 (Amazon RDS) 로 Oracle 데이터베이스를 마이그레이션하는 방법을 보여줍니다. GoldenGate

Oracle을 사용하면 GoldenGate 가동 중지 시간을 최소화하면서 원본 데이터베이스와 하나 이상의 대상 데이터베이스 간에 데이터를 복제할 수 있습니다.

참고: 소스 Oracle 데이터베이스는 온프레미스이거나 Amazon Elastic Compute Cloud(Amazon EC2) 인스턴스일 수 있습니다. 온프레미스 복제 도구를 사용할 때도 비슷한 절차를 사용할 수 있습니다.

사전 조건 및 제한 사항

사전 조건

- 활성 상태의 AWS 계정
- Oracle 라이선스 GoldenGate
- PostgreSQL 데이터베이스에 연결하기 위한 Java Database Connectivity(JDBC) 드라이버
- 대상 Amazon RDS for PostgreSQL 데이터베이스에서 [AWS Schema Conversion Tool\(AWS SCT\)](#)을 사용하여 생성한 스키마 및 테이블

제한 사항

- GoldenGate Oracle은 기존 테이블 데이터 (초기 로드) 및 진행 중인 변경 사항 (변경 데이터 캡처) 만 복제할 수 있습니다.

제품 버전

- Oracle Database Enterprise Edition 10g 또는 이후 버전
- 오라클 또는 최신 버전의 경우 오라클 GoldenGate 12.2.0.1.1
- PostgreSQL 또는 GoldenGate 최신 버전을 위한 오라클 12.2.0.1.1

아키텍처

다음 다이어그램은 Oracle을 사용하여 Oracle 데이터베이스를 PostgreSQL용 Amazon RDS로 마이그레이션하는 예제 워크플로를 보여줍니다. GoldenGate

이 다이어그램은 다음 워크플로를 보여줍니다.

1. Oracle GoldenGate [추출 프로세스](#)는 원본 데이터베이스를 대상으로 실행되어 데이터를 추출합니다.
2. Oracle GoldenGate [Replicat 프로세스](#)는 추출된 데이터를 PostgreSQL용 대상 Amazon RDS 데이터베이스에 전달합니다.

도구

- [GoldenGate오라클은 Oracle](#) Cloud Infrastructure에서 데이터 복제 및 스트리밍 데이터 처리 솔루션을 설계, 실행, 조정 및 모니터링할 수 있도록 지원합니다.
- [Amazon Relational Database Service\(RDS\) for PostgreSQL](#)는 AWS 클라우드에서 PostgreSQL 관계형 데이터베이스를 설정, 운영 및 조정하는 데 도움이 됩니다.

에픽

오라클 다운로드 및 설치 GoldenGate

작업	설명	필요한 기술
오라클을 GoldenGate 다운로드하세요.	<p>다음 버전의 Oracle을 다운로드하십시오 GoldenGate.</p> <ul style="list-style-type: none"> 오라클용 오라클 GoldenGate 12.2.0.1.1 또는 최신 버전 GoldenGatePostgreSQL용 오라클 12.2.0.1.1 또는 최신 버전 <p>소프트웨어를 다운로드하려면 오라클 웹 사이트의 Oracle 다운로드를 참조하십시오. GoldenGate</p>	DBA
원본 오라클 데이터베이스 서버에 GoldenGate 오라클용 오라클을 설치합니다.	<p>자세한 지침은 Oracle GoldenGate 설명서를 참조하십시오.</p>	DBA
Amazon EC2 인스턴스에 GoldenGate PostgreSQL용 오라클 데이터베이스를 설치합니다.	<p>지침은 Oracle 설명서를 참조하십시오. GoldenGate</p>	DBA

소스 및 대상 GoldenGate 데이터베이스에서 Oracle을 구성합니다.

작업	설명	필요한 기술
원본 데이터베이스에서 Oracle GoldenGate 데이터베이스용 Oracle을 설정합니다.	<p>지침은 Oracle GoldenGate 설명서를 참조하십시오.</p> <p>다음을 구성합니다.</p>	DBA

작업	설명	필요한 기술
	<ul style="list-style-type: none"> 보충 로깅 오라클 GoldenGate 사용자 필요한 모든 권한 부여 및 권한 파라미터 파일 관리자 프로세스 디렉터리 GLOBALS 파일 Oracle Wallet 	
대상 데이터베이스에서 GoldenGate PostgreSQL용 오라클을 설정합니다.	<p>자세한 지침은 오라클 웹 사이트의 6부 GoldenGate PostgreSQL용 오라클 사용을 참조하십시오.</p> <p>다음은 구성합니다.</p> <ul style="list-style-type: none"> 관리자 프로세스 GLOBALS 파일 Oracle Wallet 	DBA

데이터 캡처 구성

작업	설명	필요한 기술
소스 데이터베이스에서 Extract 프로세스를 설정합니다.	<p>소스 Oracle Database에서 데이터를 추출할 추출 파일을 생성합니다.</p> <p>지침은 Oracle 설명서의 ADD EXTRACT를 참조하세요.</p>	DBA

작업	설명	필요한 기술
	참고: 추출 파일에는 추출 파라미터 파일 및 트레일 파일 디렉터리 생성이 포함됩니다.	
소스에서 대상 데이터베이스로 트레일 파일을 전송하도록 데이터 펌프를 설정합니다.	<p>Oracle 웹 사이트 Database 유틸리티의 PARFILE에 있는 지침에 따라 EXTRACT 파라미터 파일 및 트레일 파일 디렉터리를 생성합니다.</p> <p>자세한 내용은 트레일이란 무엇입니까? 를 참조하십시오. 퓨전 미들웨어에서 오라클에 대한 이해 (오라클 GoldenGate 웹 사이트 참조).</p>	DBA

작업	설명	필요한 기술
<p>Amazon EC2 인스턴스에서 복제를 설정합니다.</p>	<p>복제 파라미터 파일 및 트레일 파일 디렉토리를 생성합니다.</p> <p>복제 파라미터 파일 생성에 대한 자세한 내용은 Oracle Database 설명서의 섹션 3.5 Validating a parameter file을 참조하세요.</p> <p>트레일 파일 디렉토리 생성에 대한 자세한 내용은 Oracle Cloud 설명서의 Creating a trail을 참조하세요.</p> <p>중요: 대상의 GLOBALS 파일에 체크포인트 테이블 항목을 추가해야 합니다.</p> <p>자세한 내용은 복제본이란 무엇입니까? 를 참조하십시오. 퓨전 미들웨어에서 오라클에 대한 이해 (오라클 웹 GoldenGate 사이트 참조).</p>	<p>DBA</p>

데이터 복제 구성

작업	설명	필요한 기술
<p>소스 데이터베이스에서 초기 로드를 위한 데이터를 추출할 파라미터 파일을 생성합니다.</p>	<p>Oracle Cloud 설명서의 Creating a parameter file in GGSCI에 있는 지침을 따릅니다.</p> <p>중요: 관리자가 대상에서 실행 중이어야 합니다.</p>	<p>DBA</p>

작업	설명	필요한 기술
대상 데이터베이스에서 초기 로드를 위한 데이터를 복제할 파라미터 파일을 생성합니다.	Oracle Cloud 설명서의 Creating a parameter file in GGSCI 에 있는 지침을 따릅니다. 중요: Replicat 프로세스를 추가하고 시작해야 합니다.	DBA

Amazon RDS for PostgreSQL 데이터베이스로 컷오버

작업	설명	필요한 기술
Replicat 프로세스를 중지하고 소스 데이터베이스와 대상 데이터베이스가 동기화되어 있는지 확인합니다.	소스 데이터베이스와 대상 데이터베이스 간의 행 수를 비교하여 데이터 복제가 성공했는지 확인합니다.	DBA
데이터 정의 언어(DDL) 지원을 구성합니다.	PostgreSQL에서 트리거, 시퀀스, 동의어 및 참조 키를 생성하기 위한 DDL 스크립트를 실행합니다. 참고: 표준 SQL 클라이언트 애플리케이션을 사용하여 DB 클러스터의 데이터베이스에 연결할 수 있습니다. 예를 들어 pgAdmin 을 사용하여 DB 인스턴스에 연결할 수 있습니다.	DBA

관련 리소스

- [Amazon RDS for PostgreSQL](#)(Amazon RDS 사용 설명서)
- [Amazon EC2 설명서](#)
- [오라클이 GoldenGate 지원하는 처리 방법 및 데이터베이스](#) (오라클 설명서)

AWS DMS 및 AWS SCT를 사용하여 Amazon Redshift로 Oracle 데이터베이스 마이그레이션

작성자: 피유시 고얏(AWS)

소스: Oracle	대상: Redshift	R 유형: 리아키텍트
환경: 프로덕션	기술: 마이그레이션, 분석, 데이터베이스	워크로드: Oracle
AWS 서비스: Amazon Redshift, AWS DMS		

요약

이 패턴은 AWS Database Migration Service(AWS DMS) 및 AWS Schema Conversion Tool(AWS SCT)을 사용하여 Oracle 데이터베이스를 Amazon Web Services(AWS) 클라우드의 Amazon Redshift 클라우드 데이터 웨어하우스로 마이그레이션하기 위한 지침을 제공합니다. 패턴은 온프레미스이거나 Amazon Elastic Compute Cloud(Amazon EC2) 인스턴스에 설치된 소스 Oracle 데이터베이스를 다룹니다. 또한 Oracle 데이터베이스용 Amazon Relational Database Service(RDS)에 대해서도 다룹니다.

사전 조건 및 제한 사항

사전 조건

- 온프레미스 데이터 센터 또는 AWS 클라우드에서 실행되는 Oracle 데이터베이스
- 활성 상태의 AWS 계정
- [Oracle 데이터베이스를 AWS DMS의 소스로 사용하는 방법](#) 속지
- [Amazon Redshift 데이터베이스를 AWS DMS의 대상으로 사용하는 방법](#) 속지
- Amazon RDS, Amazon Redshift, 적용 가능한 데이터베이스 기술 및 SQL에 대한 지식
- AWS SCT가 설치된 AWS SCT 커넥터용 Java 데이터베이스 연결(JDBC) 드라이버

제품 버전

- 자체 관리형 Oracle 데이터베이스의 경우, AWS DMS는 버전 10.2 이상(버전 10.x의 경우), 11g 및 최대 12.2, 18c 및 19c의 모든 Oracle 데이터베이스 버전을 지원합니다. AWS에서 관리하는 Amazon

RDS for Oracle 데이터베이스의 경우, AWS DMS는 버전 11g(버전 11.2.0.4 이상) 및 최대 12.2, 18c 및 19c의 모든 Oracle 데이터베이스 버전을 지원합니다. 가장 종합적인 버전 및 기능 지원을 위해 최신 버전의 AWS DMS를 사용하는 것을 권장합니다.

아키텍처

소스 기술 스택

다음 중 하나입니다.

- 온프레미스 Oracle 데이터베이스
- EC2 인스턴스의 Oracle 데이터베이스
- Amazon RDS for Oracle DB 인스턴스

대상 기술 스택

- Amazon Redshift

대상 아키텍처

AWS Cloud에서 실행 중인 Oracle 데이터베이스에서 Amazon Redshift로 전환

온프레미스 데이터 센터에서 실행 중인 Oracle 데이터베이스에서 Amazon Redshift로 전환

도구

- [AWS DMS](#)-AWS Data Migration Service(AWS DMS)를 사용하면 데이터베이스를 AWS로 빠르고 안전하게 마이그레이션할 수 있습니다. 소스 데이터베이스는 마이그레이션 중에도 완전히 작동하여 데이터베이스를 사용하는 애플리케이션의 가동 중지 시간을 최소화합니다. AWS DMS는 광범위하게 사용되는 상용 및 오픈 소스 데이터베이스 간에 데이터를 마이그레이션할 수 있습니다.
- [AWS SCT](#)-AWS Schema Conversion Tool(AWS SCT)을 사용하여 기존 데이터베이스 스키마를 한 데이터베이스 엔진에서 다른 데이터베이스 엔진으로 변환할 수 있습니다. Oracle, SQL Server 및 PostgreSQL을 비롯한 다양한 데이터베이스 엔진을 소스로 지원합니다.

에픽

마이그레이션 준비

작업	설명	필요한 기술
데이터베이스 버전을 검증합니다.	소스 및 대상 데이터베이스 버전을 검증하고 AWS DMS에서 지원하는지 검증합니다. 지원되는 Oracle 데이터베이스 버전에 대한 자세한 내용은 AWS DMS의 소스로서 Oracle 데이터베이스 사용 을 참조하세요. 대상으로서 Amazon Redshift 사용에 대한 자세한 내용은 AWS DMS 대상으로서 Amazon Redshift 데이터베이스 사용 을 참조하세요.	DBA
VPC 및 보안 그룹을 생성합니다.	Virtual Private Cloud(VPC)가 존재하지 않는 경우, AWS 계정에서 생성합니다. 소스 및 대상 데이터베이스로의 아웃바운드 트래픽을 위한 보안 그룹을 생성합니다. 자세한 내용은 Amazon Virtual Private Cloud(VPC) 설명서 를 참조하세요.	시스템 관리자
AWS SCT를 설치합니다.	최신 버전의 AWS SCT 및 해당 드라이버를 다운로드 및 설치합니다. 자세한 내용은 AWS SCT 설치, 확인 및 업데이트 를 참고하세요.	DBA
AWS DMS 작업을 위한 사용자를 생성합니다.	소스 데이터베이스에서 AWS DMS 사용자를 생성하고 읽기 권한을 부여합니다. 이 사용자	DBA

작업	설명	필요한 기술
	는 AWS SCT 및 AWS DMS에서 모두 사용됩니다.	
DB 연결을 테스트합니다.	Oracle DB 인스턴스와의 연결을 테스트합니다.	DBA
AWS SCT에서 새 프로젝트를 생성합니다.	AWS SCT 도구를 열고 새 프로젝트를 생성합니다.	DBA
마이그레이션할 Oracle 스키마를 분석합니다.	AWS SCT를 사용하여 마이그레이션할 스키마를 분석하고 데이터베이스 마이그레이션 평가 보고서를 생성합니다. 자세한 내용은 AWS SCT 설명서의 데이터베이스 마이그레이션 평가 보고서 생성 을 참고하세요.	DBA
평가 보고서를 검토합니다.	보고서의 마이그레이션 타당성을 검토하세요. 일부 DB 객체는 수동 변환이 필요할 수 있습니다. 보고서에 대한 자세한 내용은 AWS SCT 설명서의 평가 보고서 보기 를 참조하세요.	DBA

대상 데이터베이스를 준비합니다.

작업	설명	필요한 기술
Amazon Redshift 클러스터를 생성합니다.	이전에 생성한 VPC 내에 Amazon Redshift 클러스터를 생성합니다. 자세한 내용은 Amazon Redshift 설명서의 Amazon Redshift 클러스터 를 참조하세요.	DBA

작업	설명	필요한 기술
데이터베이스 사용자를 생성합니다.	Oracle 소스 데이터베이스에서 사용자, 역할 및 권한의 목록을 추출합니다. 대상 Amazon Redshift 데이터베이스에서 사용자를 생성하고 이전 단계의 역할을 적용합니다.	DBA
데이터베이스 파라미터를 평가합니다.	Oracle 소스 데이터베이스에서 데이터베이스 옵션, 파라미터, 네트워크 파일 및 데이터베이스 링크를 검토한 다음, 대상에 적용할 수 있는지 평가하세요.	DBA
대상에 관련 설정을 적용합니다.	이 단계에 대한 자세한 내용은 Amazon Redshift 설명서의 구성 참조 를 참고하세요.	DBA

대상 데이터베이스에 객체를 생성합니다.

작업	설명	필요한 기술
대상 데이터베이스에서 AWS DMS 사용자를 생성합니다.	대상 데이터베이스에서 AWS DMS 사용자를 생성하고 읽기와 쓰기 권한을 부여합니다. AWS SCT에서 연결을 검증합니다.	DBA
스키마를 변환하고, SQL 보고서 검토한 다음, 오류나 경고를 저장합니다.	자세한 내용은 AWS SCT 설명서의 AWS SCT를 사용한 데이터베이스 스키마 변환 을 참조하세요.	DBA

작업	설명	필요한 기술
스키마 변경 사항을 대상 데이터베이스에 적용하거나 .sql 파일로 저장합니다.	지침은 AWS SCT 설명서의 AWS SCT 설명서에서 변환된 스키마 저장 및 AWS SCT에 적용 을 참조하세요.	DBA
대상 데이터베이스의 객체를 검증합니다.	대상 데이터베이스의 이전 단계에서 만든 객체를 검증합니다. 성공적으로 변환되지 않은 모든 객체를 재작성하거나 재설계합니다.	DBA
외래 키 및 트리거를 비활성화합니다.	모든 외래 키 및 트리거를 비활성화합니다. 이로 인해 AWS DMS를 실행할 때 전체 로드 프로세스 중에 데이터 로드 문제가 발생할 수 있습니다.	DBA

AWS DMS를 사용하여 데이터를 마이그레이션합니다

작업	설명	필요한 기술
AWS DMS 복제 인스턴스를 생성합니다.	AWS Management Console에 로그인하고 AWS DMS 콘솔을 엽니다. 탐색 창에서 복제 인스턴스, 복제 인스턴스 생성을 선택합니다. 자세한 지침은 AWS DMS 설명서의 AWS DMS 시작에서 1단계 를 참조하세요.	DBA
소스 및 대상 DB 엔드포인트를 생성합니다.	소스 및 대상 엔드포인트를 생성하고, 복제 인스턴스에서 소스 및 대상 엔드포인트로의 연결을 모두 테스트합니다. 자세한 지침은 AWS DMS 설명서의	DBA

작업	설명	필요한 기술
	AWS DMS 시작에서 2단계 를 참조하세요.	
복제 작업을 생성합니다.	복제 작업을 생성하고 적절한 마이그레이션 방법을 선택합니다. 자세한 지침은 AWS DMS 설명서의 AWS DMS 시작에서 3단계 를 참조하세요.	DBA
데이터 복제를 시작합니다.	복제 작업을 시작하고 로그에서 오류가 있는지 모니터링합니다.	DBA

애플리케이션 마이그레이션

작업	설명	필요한 기술
애플리케이션 서버를 생성합니다.	AWS에 새 애플리케이션 서버를 생성합니다.	애플리케이션 소유자
애플리케이션 코드를 마이그레이션합니다.	애플리케이션 코드를 새 서버로 마이그레이션합니다.	애플리케이션 소유자
애플리케이션 서버를 구성합니다.	대상 데이터베이스 및 드라이버에 맞게 애플리케이션 서버를 구성합니다.	애플리케이션 소유자
애플리케이션 코드를 최적화합니다.	대상 엔진에 맞게 애플리케이션 코드를 최적화합니다.	애플리케이션 소유자

타겟 데이터베이스로 전환

작업	설명	필요한 기술
사용자를 검증합니다.	대상 Amazon Redshift 데이터베이스에서 사용자를 검증하고 역할 및 권한을 부여합니다.	DBA
애플리케이션이 잠겼는지 검증합니다.	추가 변경을 방지하려면 애플리케이션이 잠겨 있는지 확인하세요.	애플리케이션 소유자
데이터를 검증합니다.	대상 Amazon Redshift 데이터베이스에서 데이터를 검증합니다.	DBA
외래 키 및 트리거를 활성화합니다.	대상 Amazon Redshift 데이터베이스에서 외래 키와 트리거를 활성화합니다.	DBA
새로운 데이터베이스에 연결합니다.	애플리케이션이 새 Amazon Redshift 데이터베이스에 연결되도록 구성합니다.	애플리케이션 소유자
최종 점검을 수행합니다.	가동을 시작하기 전에 최종적이고 종합적인 시스템 점검을 수행합니다.	DBA, 애플리케이션 소유자
가동을 시작합니다.	대상 Amazon Redshift 데이터베이스를 사용하여 가동을 시작합니다.	DBA

마이그레이션 프로젝트 종료

작업	설명	필요한 기술
임시 AWS 리소스를 종료합니다.	AWS DMS 복제 인스턴스 및 AWS SCT에 사용되는 EC2 인	DBA, 시스템 관리자

작업	설명	필요한 기술
	스턴스와 같은 임시 AWS 리소스를 종료합니다.	
문서를 검토합니다.	마이그레이션 프로젝트 문서를 검토 및 검증합니다.	DBA, 시스템 관리자
지표를 수집합니다.	마이그레이션 시간, 수동 작업과 도구 작업의 비율, 총 비용 절감 등 마이그레이션 프로젝트에 대한 정보를 수집합니다.	DBA, 시스템 관리자
프로젝트를 마무리합니다.	프로젝트를 마무리하고 피드백을 제공하세요.	DBA, 시스템 관리자

관련 리소스

참조

- [AWS DMS 사용 설명서](#)
- [AWS SCT 사용 설명서](#)
- [Amazon Redshift 시작 안내서](#)

자습서 및 동영상

- [AWS SCT와 AWS DMS에 대해 자세히 알아보기](#)(AWS re:Invent 2019의 프레젠테이션)
- [AWS Database Migration Service 시작하기](#)

AWS DMS 및 AWS SCT를 사용하여 Aurora PostgreSQL로 Oracle 데이터베이스를 마이그레이션하기

작성자: Senthil Ramasamy (AWS)

환경: PoC 또는 파일럿	소스: Oracle 데이터베이스	대상: Amazon Aurora PostgreSQL-Compatible
R 유형: 리아키텍트	워크로드: Oracle	기술: 마이그레이션, 데이터베이스

AWS 서비스: Amazon Aurora

요약

이 패턴은 AWS Data Migration Service(AWS DMS) 및 AWS Schema Conversion Tool(AWS SCT)을 사용하여 Oracle 데이터베이스를 Amazon Aurora PostgreSQL-Compatible Edition으로 마이그레이션하는 방법을 설명합니다.

이 패턴에는 온프레미스에 있는 원본 Oracle 데이터베이스, Amazon Elastic Compute Cloud(Amazon EC2) 인스턴스에 설치된 Oracle 데이터베이스, Oracle 데이터베이스용 Amazon Relational Database Service(RDS)가 포함됩니다. 이 패턴은 이러한 데이터베이스를 Aurora PostgreSQL-Compatible로 변환합니다.

사전 조건 및 제한 사항

사전 조건

- 활성 상태의 AWS 계정
- 온프레미스 데이터 센터 또는 AWS 클라우드의 Oracle 데이터베이스.
- 로컬 시스템 또는 EC2 인스턴스에 설치된 SQL 클라이언트.
- AWS SCT가 설치된 로컬 시스템 또는 EC2 인스턴스에 설치되는 AWS SCT 커넥터용 자바 데이터베이스 연결(JDBC) 드라이버.

제한 사항

- 데이터베이스 크기 제한: 128TB
- 원본 데이터베이스가 상용 off-the-shelf (COTS) 응용 프로그램을 지원하거나 공급업체별 데이터베이스인 경우 다른 데이터베이스 엔진으로 변환하지 못할 수 있습니다. 이 패턴을 사용하기 전에 애플리케이션이 Aurora PostgreSQL-Compatible을 지원하는지 확인하십시오.

제품 버전

- 원본으로 자체 관리형 Oracle 데이터베이스의 경우, 는 버전 10.2 이상, 11g 및 12.2 이하, 18c 및 19c의 모든 Oracle 데이터베이스 버전을 지원합니다. 지원되는 Oracle 데이터베이스 버전(자체 관리형 버전과 Amazon RDS for Oracle)의 최신 목록은 [Oracle 데이터베이스를 AWS DMS의 소스로 사용](#) 및 [PostgreSQL 데이터베이스를 AWS DMS의 대상으로 사용](#)을 참고하십시오.
- 가장 종합적인 버전 및 기능 지원을 위해 최신 버전의 AWS DMS를 사용하는 것을 권장합니다. AWS SCT에서 지원하는 Oracle 데이터베이스 버전에 대한 자세한 내용은 [AWS SCT 설명서](#)를 참고하십시오.
- Aurora는 [Amazon Aurora PostgreSQL 릴리스 및 엔진 버전](#)에 나열된 PostgreSQL 버전을 지원합니다.

아키텍처

소스 기술 스택

다음 중 하나입니다.

- 온프레미스 Oracle 데이터베이스
- EC2 인스턴스의 Oracle 데이터베이스
- Amazon RDS for Oracle DB 인스턴스

대상 기술 스택

- Aurora PostgreSQL-Compatible

대상 아키텍처

데이터 마이그레이션 아키텍처

- AWS Cloud에서 실행 중인 Oracle 데이터베이스에서
- 온프레미스 데이터 센터에서 실행 중인 Oracle 데이터베이스에서

도구

- [AWS Database Migration Service\(AWS DMS\)](#)를 사용하면 데이터 스토어를 AWS 클라우드로 마이그레이션하거나 클라우드와 온프레미스 설정 조합 간에 마이그레이션할 수 있습니다.
- [AWS Schema Conversion Tool\(AWS SCT\)](#)은 소스 데이터베이스 스키마와 대부분의 사용자 지정 코드를 대상 데이터베이스와 호환되는 형식으로 자동 변환하여 이기종 데이터베이스 마이그레이션을 지원합니다.

에픽

마이그레이션 준비

작업	설명	필요한 기술
소스 데이터베이스를 준비합니다.	소스 데이터베이스를 준비하려면 AWS SCT 설명서의 AWS SCT용 소스로 Oracle 데이터베이스 사용하기 를 참고하십시오.	DBA
AWS SCT용 EC2 인스턴스를 생성합니다.	필요한 경우 AWS SCT용 EC2 인스턴스를 생성하고 구성합니다.	DBA
AWS SCT를 다운로드하십시오.	최신 버전의 AWS SCT 및 관련 드라이버를 다운로드합니다. 자세한 내용은 AWS SCT 설명서의 AWS SCT 설치, 확인 및 업데이트 를 참고하십시오.	DBA

작업	설명	필요한 기술
사용자 및 권한을 추가합니다.	소스 데이터베이스에 필수 사용자 및 권한을 추가하고 검증합니다.	DBA
AWS SCT 프로젝트를 생성합니다.	워크로드용 AWS SCT 프로젝트를 생성하고 소스 데이터베이스에 연결합니다. 지침은 AWS SCT 설명서의 AWS SCT 프로젝트 생성 및 데이터베이스 서버 추가 를 참고하십시오.	DBA
타당성을 평가합니다.	자동으로 변환할 수 없는 스키마의 작업 항목을 요약하고 수동 변환 작업에 대한 추정치를 제공하는 평가 보고서를 생성합니다. 자세한 내용은 AWS SCT 설명서의 데이터베이스 마이그레이션 평가 보고서 생성 및 검토 를 참고하십시오.	DBA

대상 데이터베이스를 준비합니다.

작업	설명	필요한 기술
대상 Amazon RDS DB 인스턴스를 생성합니다.	Amazon Aurora를 데이터베이스 엔진으로 사용하여 대상 Amazon RDS DB 인스턴스를 생성합니다. 지침은 Amazon RDS 설명서의 Amazon RDS DB 인스턴스 생성 을 참고하십시오.	DBA

작업	설명	필요한 기술
사용자, 역할, 권한을 추출합니다.	소스 데이터베이스에서 사용자, 역할 및 권한 목록을 추출합니다.	DBA
지도 사용자.	기존 데이터베이스 사용자를 새 데이터베이스 사용자에게 매핑합니다.	앱 소유자
사용자를 생성합니다.	대상 데이터베이스에서 사용자를 생성합니다.	DBA, 앱 소유자
역할을 적용합니다.	대상 데이터베이스에 이전 단계의 역할을 적용합니다.	DBA
옵션, 매개변수, 네트워크 파일, 데이터베이스 링크를 확인하십시오.	소스 데이터베이스에서 옵션, 매개변수, 네트워크 파일 및 데이터베이스 링크를 검토한 다음, 대상 데이터베이스에 적용할 수 있는지 평가하십시오.	DBA
설정을 적용합니다.	대상 데이터베이스에 관련 설정을 적용합니다.	DBA

객체 전송

작업	설명	필요한 기술
AWS SCT 연결을 구성합니다.	대상 데이터베이스에 대한 AWS SCT 연결을 구성합니다.	DBA
AWS SCT를 사용하여 스키마를 변환합니다.	AWS SCT는 소스 데이터베이스 스키마와 대부분의 사용자 지정 코드를 대상 데이터베이스와 호환되는 형식으로 자동 변환합니다. 도구가 자동으로 변환할 수 없는 코드는 명확하	DBA

작업	설명	필요한 기술
	게 표시되므로 수동으로 변환할 수 있습니다.	
보고서를 검토합니다.	생성된 SQL 보고서를 검토하고 모든 오류 및 경고를 저장합니다.	DBA
자동화된 스키마 변경 사항을 적용합니다.	자동화된 스키마 변경 사항을 대상 데이터베이스에 적용하거나 .sql 파일로 저장합니다.	DBA
객체를 검증합니다.	AWS SCT가 대상에 객체를 생성했는지 확인합니다.	DBA
변환되지 않은 항목을 처리합니다.	자동 변환에 실패한 모든 항목을 수동으로 재작성, 거부 또는 재설계하세요.	DBA, 앱 소유자
역할 및 사용자 권한을 적용합니다.	생성된 역할 및 사용자 권한을 적용하고 예외를 검토합니다.	DBA

데이터 마이그레이션

작업	설명	필요한 기술
방법을 결정합니다.	데이터 마이그레이션 방법을 결정합니다.	DBA
복제 인스턴스를 생성합니다.	AWS DMS 콘솔에서 복제 인스턴스를 생성합니다. 자세한 내용은 AWS DMS 설명서의 AWS DMS 복제 인스턴스 사용하기 를 참고하십시오.	DBA
원본 및 대상 엔드포인트를 생성합니다.	엔드포인트를 생성하려면 AWS DMS 설명서의 소스 및	DBA

작업	설명	필요한 기술
	대상 엔드포인트 생성 지침을 따르십시오.	
복제 작업을 생성합니다.	작업을 생성하려면 AWS DMS 설명서의 AWS DMS 작업 사용 을 참고하십시오.	DBA
복제 작업을 시작하고 로그를 모니터링합니다.	이 단계에 대한 자세한 내용은 AWS DMS 설명서의 AWS DMS 작업 모니터링 을 참고하십시오.	DBA

애플리케이션 마이그레이션

작업	설명	필요한 기술
애플리케이션 코드의 SQL 항목을 분석하고 변환합니다.	AWS SCT를 사용하여 애플리케이션 코드 내에서 SQL 항목을 분석하고 변환합니다. 한 엔진에서 다른 엔진으로 데이터베이스 스키마를 변환할 때는 이전 데이터베이스 엔진 대신 새 데이터베이스 엔진과 상호 작용하도록 애플리케이션의 SQL 코드도 업데이트해야 합니다. 변환된 SQL 코드를 보고, 분석하고, 편집하고, 저장할 수 있습니다.	앱 소유자
애플리케이션 서버를 생성합니다.	AWS에 새 애플리케이션 서버를 생성합니다.	앱 소유자
애플리케이션 코드를 마이그레이션합니다.	애플리케이션 코드를 새 서버로 마이그레이션합니다.	앱 소유자

작업	설명	필요한 기술
애플리케이션 서버를 구성합니다.	대상 데이터베이스 및 드라이버의 애플리케이션 서버를 구성합니다.	앱 소유자
코드를 수정합니다.	애플리케이션의 소스 데이터베이스 엔진과 관련된 모든 코드를 수정합니다.	앱 소유자
C 최적화	대상 데이터베이스 엔진에 맞게 애플리케이션 코드를 최적화합니다.	앱 소유자

전환

작업	설명	필요한 기술
타겟 데이터베이스로 전환	새 데이터베이스로 컷오버를 수행합니다.	DBA
애플리케이션을 잠급니다.	애플리케이션이 더 이상 변경되지 않도록 잠급니다.	앱 소유자
변경 사항을 검증합니다.	모든 변경 사항이 대상 데이터베이스에 전파되었는지 확인합니다.	DBA
대상 데이터베이스로 리디렉션합니다.	새 애플리케이션 서버를 대상 데이터베이스로 가리킵니다.	앱 소유자
모든 것을 확인합니다.	최종적이고 종합적인 시스템 점검을 수행합니다.	앱 소유자
가동을 시작합니다.	최종 컷오버 작업을 완료합니다.	앱 소유자

프로젝트 닫기

작업	설명	필요한 기술
임시 리소스를 종료합니다.	AWS DMS 복제 인스턴스 및 AWS SCT에 사용되는 EC2 인스턴스와 같은 임시 AWS 리소스를 종료합니다.	DBA, 앱 소유자
피드백을 업데이트합니다.	내부 팀을 위한 AWS DMS 프로세스에 대한 피드백을 업데이트합니다.	DBA, 앱 소유자
프로세스 및 템플릿을 수정합니다.	AWS DMS 프로세스를 수정하고 필요한 경우 템플릿을 개선합니다.	DBA, 앱 소유자
문서를 검증합니다.	프로젝트 문서를 검토하고 검증하세요.	DBA, 앱 소유자
지표를 수집합니다.	지표를 수집하여 마이그레이션 시간, 수동 대비 도구 비용 절감 비율 등을 평가합니다.	DBA, 앱 소유자
프로젝트 닫기	마이그레이션 프로젝트를 종료하고 이해관계자에게 피드백을 제공합니다.	DBA, 앱 소유자

관련 리소스

참조

- [Oracle 데이터베이스를 AWS DMS의 원본으로 사용](#)
- [PostgreSQL 데이터베이스를 AWS 데이터베이스 마이그레이션 서비스용 대상으로 사용](#)
- [PostgreSQL Compatibility \(9.6.x\) Migration Playbook으로 Oracle Database 11g/12c를 Amazon Aurora로](#)
- [PostgreSQL Compatibility\(12.4\) Migration Playbook으로 Oracle Database 19c를 Amazon Aurora로](#)

- [Amazon RDS for Oracle 데이터베이스를 Amazon Aurora PostgreSQL-Compatible Edition으로 마이그레이션](#)
- [AWS Data Migration Service](#)
- [AWS Schema Conversion Tool](#)
- [Oracle에서 Amazon Aurora로 마이그레이션](#)
- [Amazon RDS 요금](#)

자습서 및 동영상

- [데이터베이스 마이그레이션 단계별 안내](#)
- [AWS DMS 시작하기](#)
- [Amazon RDS 시작](#)
- [AWS Data Migration Service\(동영상\)](#)
- [Oracle 데이터베이스를 PostgreSQL로 마이그레이션하기 \(비디오\)](#)

추가 정보

Aurora PostgreSQL로 온프레미스 Oracle 데이터베이스의 데이터를 마이그레이션하기

작성자: Michelle Deng (AWS) 및 Shunan Xiang (AWS)

환경: PoC 또는 파일럿	소스: Oracle	대상: Aurora PostgreSQL-Compatible
R 유형: 리아키텍트	워크로드: Oracle	기술: 마이그레이션, 데이터베이스

AWS 서비스: Amazon Aurora;
AWS DMS; AWS SCT

요약

이 패턴은 온프레미스 Oracle 데이터베이스에서 Amazon Aurora PostgreSQL-Compatible Edition으로 데이터를 마이그레이션하기 위한 지침을 제공합니다. 이는 DML(데이터 조작 언어) 활동이 많은 대형 테이블을 포함하는 수 테라바이트 규모의 Oracle 데이터베이스를 대상으로 가동 중지 시간을 최소화하는 온라인 데이터 마이그레이션 전략을 목표로 합니다. Oracle Active Data Guard 스탠바이 데이터베이스는 기본 데이터베이스에서 데이터 마이그레이션을 오프로드하기 위한 소스로 사용됩니다. ORA-01555 오류를 방지하기 위해 전체 로드 중에 Oracle 기본 데이터베이스에서 대기 데이터베이스로의 복제를 일시 중단할 수 있습니다.

데이터 유형이 NUMBER인 기본 키(PK) 또는 외래 키(FK)의 테이블 열은 일반적으로 Oracle에 정수를 저장하는 데 사용됩니다. 더 나은 성능을 위해 PostgreSQL에서 이를 INT 또는 BIGINT로 변환하는 것을 권장합니다. AWS Schema Conversion Tool(AWS SCT)을 사용하여 PK 및 FK 열의 기본 데이터 유형 매핑을 변경할 수 있습니다. (자세한 내용은 AWS 블로그 게시물 [Oracle에서 PostgreSQL로 NUMBER 데이터 유형 변환](#)을 참고하십시오.) 이 패턴의 데이터 마이그레이션에서는 전체 로드와 변경 데이터 캡처(CDC) 모두에 AWS Database Migration Service(AWS DMS)를 사용합니다.

또한, 이 패턴을 사용하여 PostgreSQL용 Amazon Relational Database Service(Amazon RDS)로 온프레미스 Oracle 데이터베이스를 마이그레이션하거나, Amazon Elastic Compute Cloud(Amazon EC2)에서 호스팅되는 Oracle 데이터베이스를 Amazon RDS for PostgreSQL 또는 Aurora PostgreSQL-Compatible로 마이그레이션할 수 있습니다.

사전 조건 및 제한 사항

사전 조건

- 활성 상태의 AWS 계정.
- Active Data Guard 예비 데이터베이스가 구성된 온프레미스 데이터 센터의 Oracle 원본 데이터베이스
- 온프레미스 데이터 센터와 AWS 클라우드 간에 구성된 AWS Direct Connect
- [Oracle 데이터베이스를 AWS DMS의 원본으로 사용하는 방법](#) 속지
- [PostgreSQL 데이터베이스를 AWS DMS의 대상으로 사용하는 방법](#) 속지

제한 사항

- Amazon Aurora 데이터베이스 클러스터는 최대 128TiB의 스토리지로 생성할 수 있습니다. Amazon RDS for PostgreSQL 데이터베이스 인스턴스는 최대 64TiB의 스토리지로 생성할 수 있습니다. 최신 스토리지 정보는 AWS 설명서의 [Amazon Aurora 스토리지 및 신뢰성](#) 및 [Amazon RDS DB 인스턴스 스토리지](#)를 참고하십시오.

제품 버전

- AWS DMS는 버전 10.2 이상(버전 10.x의 경우), 11g 및 최대 12.2, 18c 및 19c까지의 모든 Oracle 데이터베이스 에디션을 지원합니다. 지원되는 버전의 최신 목록은 AWS 설명서에서 [Oracle 데이터베이스를 AWS DMS용 원본으로 사용하기](#) 섹션을 참조하세요.

아키텍처

소스 기술 스택

- Oracle Active Data Guard 예비 데이터베이스가 구성된 온프레미스 Oracle 데이터베이스

대상 기술 스택

- Aurora PostgreSQL-Compatible

데이터 마이그레이션 아키텍처

도구

- AWS DMS - [AWS Database Migration Service](#)(AWS DMS)는 여러 원본 및 대상 데이터베이스를 지원합니다. 지원되는 Oracle 원본 및 대상 데이터베이스 버전과 에디션의 목록은 AWS DMS 설명서에서 [Oracle 데이터베이스를 AWS DMS의 원본으로 사용하기](#) 섹션을 참조하세요. 원본 데이터베이스가 AWS DMS에서 지원되지 않는 경우, 6단계 (Epic 섹션) 에서 데이터를 마이그레이션할 다른 방법을 선택해야 합니다. 중요 참고: 이 마이그레이션은 이기종 마이그레이션이므로 먼저 데이터베이스가 상용 (COTS) 애플리케이션을 지원하는지 확인해야 합니다. off-the-shelf 애플리케이션이 COTS인 경우 진행하기 전에 공급업체에 문의하여 Aurora PostgreSQL-Compatible이 지원되는지 확인하십시오. 자세한 내용은 AWS 설명서의 [AWS DMS 단계별 마이그레이션 안내](#)를 참고하십시오.
- AWS SCT - [AWS Schema Conversion Tool](#)(AWS SCT)은 원본 데이터베이스 스키마와 대부분의 사용자 지정 코드를 대상 데이터베이스와 호환되는 형식으로 자동 변환하여 유형이 다른 데이터베이스 마이그레이션을 촉진합니다. 도구가 변환하는 사용자 지정 코드에는 보기, 저장된 절차 및 함수가 포함됩니다. 도구가 자동으로 변환할 수 없는 코드는 명확하게 표시되므로 직접 변환할 수 있습니다.

에픽

마이그레이션 계획

작업	설명	필요한 기술
소스 및 대상 데이터베이스 버전을 검증합니다.		DBA
AWS SCT 및 드라이버를 설치합니다.		DBA
원본 데이터베이스에 AWS SCT 필수 사용자 및 권한을 추가하고 검증합니다.		DBA
워크로드용 AWS SCT 프로젝트를 생성하고 소스 데이터베이스에 연결합니다.		DBA

작업	설명	필요한 기술
평가 보고서를 작성하고 타당성을 평가하세요.		DBA, 앱 소유자

대상 데이터베이스를 준비합니다.

작업	설명	필요한 기술
Aurora PostgreSQL-Compatible 대상 데이터베이스를 생성합니다.		DBA
원본 데이터베이스에서 사용자, 역할 및 권한 목록을 추출합니다.		DBA
기존 데이터베이스 사용자를 새 데이터베이스 사용자에게 매핑합니다.		앱 소유자
대상 데이터베이스에서 사용자를 생성합니다.		DBA
대상 Aurora PostgreSQL-Compatible 데이터베이스에 이전 단계의 역할을 적용합니다.		DBA
원본 데이터베이스의 데이터베이스 옵션, 파라미터, 네트워크 파일 및 데이터베이스 링크를 검토하고 대상 데이터베이스에 적용할 수 있는지 평가하세요.		DBA, 앱 소유자
대상 데이터베이스에 관련 설정을 적용합니다.		DBA

데이터베이스 객체 코드 변환 준비

작업	설명	필요한 기술
대상 데이터베이스에 대한 AWS SCT 연결을 구성합니다.		DBA
AWS SCT에서 스키마를 변환하고 변환된 코드를 .sql 파일로 저장합니다.		DBA, 앱 소유자
자동 변환에 실패한 모든 데이터베이스 객체를 수동으로 변환합니다.		DBA, 앱 소유자
데이터베이스 코드 변환을 최적화합니다.		DBA, 앱 소유자
객체 유형에 따라 .sql 파일을 여러 .sql 파일로 분리합니다.		DBA, 앱 소유자
대상 데이터베이스에서 SQL 스크립트를 확인합니다.		DBA, 앱 소유자

데이터 마이그레이션 준비

작업	설명	필요한 기술
AWS DMS 복제 인스턴스를 생성합니다.		DBA
원본 및 대상 DB 엔드포인트	PK 및 FK의 데이터 유형이 Oracle의 NUMBER에서 PostgreSQL의 BIGINT로 변환되는 경우 소스 엔드포인트를 생성할 때 연결 속성 <code>numberDataTypeScale</code>	DBA

작업	설명	필요한 기술
	e=-2 를 지정하는 것이 좋습니다.	

데이터 마이그레이션 — 전체 로드

작업	설명	필요한 기술
대상 데이터베이스에 스키마와 테이블을 생성합니다.		DBA
테이블을 그룹화하거나 테이블 크기에 따라 큰 테이블을 분할하여 AWS DMS 전체 로드 작업을 생성합니다.		DBA
소스 Oracle 데이터베이스에서 애플리케이션을 잠시 중지합니다.		앱 소유자
Oracle 스탠바이 데이터베이스가 기본 데이터베이스와 동기화되는지 확인하고 기본 데이터베이스에서 스탠바이 데이터베이스로의 복제를 중지합니다.		DBA, 앱 소유자
소스 Oracle 데이터베이스에서 애플리케이션을 시작합니다.		앱 소유자
Oracle 스탠바이 데이터베이스에서 Aurora PostgreSQL-Compatible 데이터베이스까지 AWS DMS 전체 로드 작업을 병렬로 시작합니다.		DBA

작업	설명	필요한 기술
전체 로드가 완료된 후 PK 및 보조 인덱스를 생성합니다.		DBA
데이터를 검증합니다.		DBA

데이터 마이그레이션 — CDC

작업	설명	필요한 기술
Oracle 스탠바이가 기본 데이터베이스와 동기화된 시점과 이전 작업에서 애플리케이션이 다시 시작되기 전에, 사용자 지정 CDC 시작 시간 또는 시스템 변경 번호(SCN)를 지정하여 AWS DMS 지속 복제 작업을 생성합니다.		DBA
AWS DMS 작업을 병렬로 시작하여 Oracle 예비 데이터베이스에서 Aurora PostgreSQL-Compatible 데이터베이스까지 지속적으로 변경되는 사항을 복제합니다.		DBA
Oracle 기본 데이터베이스에서 스탠바이 데이터베이스로 복제를 재설정합니다.		DBA
대상 Aurora PostgreSQL-Compatible 데이터베이스가 소스 Oracle 데이터베이스와 거의 동기화될 때 Oracle 데이터베이스에서 로그를 모니터링		DBA, 앱 소유자

작업	설명	필요한 기술
하고 애플리케이션을 중지합니다.		
대상이 소스 Oracle 데이터 베이스와 완전히 동기화되면 AWS DMS 작업을 중지합니다.		DBA
FK를 생성하고 원본 및 대상 데이터베이스의 데이터를 확인합니다.		DBA
대상 데이터베이스에 함수, 보기, 트리거, 시퀀스 및 기타 객체 유형을 생성합니다.		DBA
대상 데이터베이스에 역할 부여를 적용합니다.		DBA

애플리케이션 마이그레이션

작업	설명	필요한 기술
AWS SCT를 사용하여 애플리케이션 코드 내에서 SQL 문을 분석하고 변환합니다.		앱 소유자
AWS에 새 애플리케이션 서버를 생성합니다.		앱 소유자
애플리케이션 코드를 새 서버로 마이그레이션합니다.		앱 소유자
대상 데이터베이스 및 드라이버에 맞게 애플리케이션 서버를 구성합니다.		앱 소유자

작업	설명	필요한 기술
애플리케이션의 소스 데이터베이스 엔진 관련 코드를 모두 수정하세요.		앱 소유자
대상 데이터베이스에 맞게 애플리케이션 코드를 최적화합니다.		앱 소유자

전환

작업	설명	필요한 기술
새 애플리케이션 서버를 대상 데이터베이스로 가리킵니다.		DBA, 앱 소유자
온전성 검사를 수행합니다.		DBA, 앱 소유자
가동을 시작합니다.		DBA, 앱 소유자

프로젝트 닫기

작업	설명	필요한 기술
임시 AWS 리소스를 종료합니다.		DBA, 시스템 관리자
프로젝트 문서를 검토하고 검증하세요.		DBA, 앱 소유자
마이그레이션 시간, 수동 대비 도구 사용 비율, 비용 절감, 기타 유사한 데이터에 대한 지표를 수집합니다.		DBA, 앱 소유자

작업	설명	필요한 기술
프로젝트를 마무리하고 피드백을 제공하세요.		DBA, 앱 소유자

관련 리소스

참조

- [Oracle Database에서 Aurora PostgreSQL-Compatible로: 마이그레이션 플레이북](#)
- [Amazon RDS for Oracle Database를 Amazon Aurora MySQL로 마이그레이션](#)
- [AWS DMS 웹사이트](#)
- [AWS DMS 설명서](#)
- [AWS SCT 웹 사이트](#)
- [AWS SCT 설명서](#)
- [Oracle에서 Amazon Aurora로 마이그레이션](#)

자습서

- [AWS DMS 시작하기](#)
- [Amazon RDS 시작](#)
- [AWS 데이터베이스 마이그레이션 서비스 단계별 안내](#)

AWS DMS를 사용하여 SAP ASE에서 Amazon RDS for SQL Server로 마이그레이션

작성자: Amit Kumar(AWS)

환경: PoC 또는 파일럿	소스: SAP ASE	대상: Amazon RDS for SQL Server
R 유형: 리아키텍트	워크로드: SAP	기술: 마이그레이션, 데이터베이스, 현대화
AWS 서비스: Amazon RDS, AWS DMS		

요약

이 패턴은 SAP Adaptive Server Enterprise(ASE) 데이터베이스를 Microsoft SQL Server가 실행되는 Amazon Relational Database Service(Amazon RDS) DB 인스턴스로 마이그레이션하기 위한 지침을 제공합니다. 소스 데이터베이스는 온프레미스 데이터 센터 또는 Amazon Elastic Compute Cloud(Amazon EC2) 인스턴스에 있을 수 있습니다. 이 패턴은 AWS Database Migration Service(AWS DMS)를 사용하여 데이터를 마이그레이션하고, 선택 사항으로 컴퓨터 지원 소프트웨어 엔지니어링 (CASE) 도구를 사용하여 데이터베이스 스키마를 변환합니다.

사전 조건 및 제한 사항

사전 조건

- 활성 상태의 AWS 계정.
- 온프레미스 데이터 센터 또는 EC2 인스턴스의 SAP ASE 데이터베이스
- 실행 중인 대상 Amazon RDS for SQL Server 데이터베이스

제한 사항

- 데이터베이스 크기 제한: 64TB

제품 버전

- SAP ASE 버전 15.7 또는 16.x만 해당됩니다. 최신 정보는 [AWS DMS에서 SAP 데이터베이스를 소스로 사용](#)을 참조하세요.
- Amazon RDS 대상 데이터베이스의 경우, AWS DMS는 엔터프라이즈, 스탠다드, 웹 및 익스프레스 에디션용 [Amazon RDS의 Microsoft SQL Server 버전](#)을 지원합니다. 지원되는 버전에 대한 최신 정보는 [AWS DMS 설명서](#)를 참조하세요. 가장 종합적인 버전 및 기능 지원을 위해 최신 버전의 AWS DMS를 사용하는 것을 권장합니다.

아키텍처

소스 기술 스택

- 온프레미스 또는 Amazon EC2 인스턴스에 있는 SAP ASE 데이터베이스

대상 기술 스택

- Amazon RDS for SQL Server DB 인스턴스

소스 및 대상 아키텍처

Amazon EC2의 SAP ASE 데이터베이스에서 Amazon RDS for SQL Server DB 인스턴스로:

온프레미스 SAP ASE 데이터베이스에서 Amazon RDS for SQL Server DB 인스턴스로:

도구

- [AWS Database Migration Service](#)(AWS DMS)는 온프레미스, Amazon RDS DB 인스턴스 또는 EC2 인스턴스의 데이터베이스에서 Amazon RDS for SQL Server 또는 EC2 인스턴스와 같은 AWS 서비스의 데이터베이스로 데이터를 마이그레이션하는 데 사용할 수 있는 웹 서비스입니다. 또한 데이터베이스를 AWS 서비스에서 온 프레미스 데이터베이스로 마이그레이션할 수 있습니다. 이기종 또는 동종 데이터베이스 엔진 간에 데이터를 마이그레이션할 수 있습니다.
- [스키마 변환의 경우 선택적으로 erwin 데이터 모델러 또는 SAP를 사용할 수 있습니다. PowerDesigner](#)

에픽

마이그레이션 계획

작업	설명	필요한 기술
소스 및 대상 데이터베이스 버전을 검증합니다.		DBA
스토리지 요구 사항(스토리지 유형 및 용량)을 확인합니다.		DBA, SysAdmin
용량, 스토리지 기능, 네트워크 기능에 따라 적절한 인스턴스 유형을 선택합니다.		DBA, SysAdmin
소스 및 대상 데이터베이스의 네트워크 액세스 보안 요구 사항을 식별합니다.		DBA, SysAdmin
애플리케이션 마이그레이션 전략을 파악합니다.		DBA SysAdmin, 앱 소유자

인프라 구성

작업	설명	필요한 기술
Virtual Private Cloud(VPC) 및 서브넷을 생성합니다.		SysAdmin
보안 그룹 및 네트워크 액세스 제어 목록(ACL)을 생성합니다.		SysAdmin
Amazon RDS DB 인스턴스를 구성 및 시작합니다.		SysAdmin

데이터 마이그레이션 - 옵션 1

작업	설명	필요한 기술
데이터베이스 스키마를 수동으로 마이그레이션하거나 erwin 데이터 모델러 또는 SAP와 같은 CASE 도구를 사용하십시오. PowerDesigner		DBA

데이터 마이그레이션 - 옵션 2

작업	설명	필요한 기술
AWS DMS를 사용하여 데이터를 마이그레이션합니다.		DBA

애플리케이션 마이그레이션

작업	설명	필요한 기술
애플리케이션 마이그레이션 전략을 따릅니다.		DBA, 앱 소유자 SysAdmin

전환

작업	설명	필요한 기술
애플리케이션 클라이언트를 새 인프라로 전환합니다.		DBA SysAdmin, 앱 소유자

프로젝트 닫기

작업	설명	필요한 기술
임시 AWS 리소스를 종료합니다.		DBA, SysAdmin
프로젝트 문서를 검토하고 검증하세요.		DBA SysAdmin, 앱 소유자
마이그레이션 시간, 수동 작업과 자동 작업의 비율, 비용 절감과 같은 지표를 수집합니다.		DBA SysAdmin, 앱 소유자
프로젝트를 마무리하고 피드백을 제공하세요.		DBA SysAdmin, 앱 소유자

관련 리소스

참조

- [AWS DMS 웹사이트](#)
- [Amazon RDS 요금](#)
- [AWS DMS에서 SAP ASE 데이터베이스를 소스로 사용](#)
- [RDS Custom for SQL Server 제한 사항](#)

자습서 및 동영상

- [AWS DMS 시작하기](#)
- [Amazon RDS 시작](#)
- [DMS \(동영상\)](#)
- [Amazon RDS\(동영상\)](#)

AWS DMS를 사용하여 온프레미스 Microsoft SQL Server 데이터베이스를 Amazon Redshift로 마이그레이션

작성자: 마르셀로 페르난데스(AWS)

환경: PoC 또는 파일럿	소스: Microsoft SQL Server	대상: Amazon Redshift
R 유형: 리아키텍트	워크로드: Microsoft	기술: 마이그레이션, 데이터베이스
AWS 서비스: Amazon Redshift		

요약

이 패턴은 AWS Data Migration Service(AWS DMS)를 사용하여 온프레미스 Microsoft SQL Server 데이터베이스를 Amazon Redshift로 마이그레이션하기 위한 지침을 제공합니다.

사전 조건 및 제한 사항

사전 조건

- 활성 상태의 AWS 계정
- 온프레미스 데이터 센터의 Microsoft SQL Server 소스 데이터베이스
- [AWS DMS 설명서](#)에 설명된 대로 Amazon Redshift 데이터베이스를 AWS DMS의 대상으로 사용하기 위한 사전 조건 완료

제품 버전

- SQL Server 2005-2019, Enterprise, Standard, Workgroup, Developer 및 Web 에디션. 지원되는 버전의 최신 목록은 AWS 설명서에서 [Microsoft SQL Server 데이터베이스를 AWS DMS용 소스로 사용하기](#)를 참조하세요.

아키텍처

소스 기술 스택

- 온프레미스 Microsoft SQL Server 데이터베이스

대상 기술 스택

- Amazon Redshift

데이터 마이그레이션 아키텍처

도구

- [AWS DMS](#)는 여러 유형의 소스 및 대상 데이터베이스를 지원하는 데이터 마이그레이션 서비스입니다. AWS DMS와 함께 사용할 수 있는 Microsoft SQL Server 데이터베이스 버전 및 에디션에 대한 자세한 내용은 AWS DMS 설명서의 [Microsoft SQL Server 데이터베이스를 AWS DMS용 소스로 사용](#)을 참조하세요. AWS DMS가 소스 데이터베이스를 지원하지 않는 경우, 데이터 마이그레이션을 위한 대체 수단을 선택해야 합니다.

에픽

마이그레이션 계획

작업	설명	필요한 기술
소스 및 대상 데이터베이스 버전과 엔진의 유효성을 확인합니다.		DBA
대상 서버 인스턴스의 하드웨어 요구 사항을 확인합니다.		DBA, 시스템 관리자
스토리지 요구 사항(스토리지 유형 및 용량)을 확인합니다.		DBA, 시스템 관리자
용량, 스토리지 기능, 네트워크 기능에 따라 적절한 인스턴스 유형을 선택합니다.		DBA, 시스템 관리자

작업	설명	필요한 기술
소스 및 대상 데이터베이스의 네트워크 액세스 보안 요구 사항을 식별합니다.		DBA, 시스템 관리자
애플리케이션 마이그레이션 전략을 파악합니다.		DBA, 앱 소유자, 시스템 관리자

인프라 구성

작업	설명	필요한 기술
Virtual Private Cloud(VPC)를 생성합니다.	자세한 내용은 AWS 설명서의 VPC에서 DB 인스턴스로 작업 을 참조하세요.	시스템 관리자
보안 그룹을 생성합니다.		시스템 관리자
Amazon Redshift 클러스터를 구성하고 시작합니다.	자세한 내용은 Amazon Redshift 설명서의 Amazon Redshift 클러스터 샘플 생성 을 참조하세요.	DBA, 시스템 관리자

데이터 마이그레이션

작업	설명	필요한 기술
AWS DMS를 사용하여 Microsoft SQL Server 데이터베이스에서 데이터를 마이그레이션합니다.		DBA

애플리케이션 마이그레이션

작업	설명	필요한 기술
애플리케이션 마이그레이션 전략을 따릅니다.		DBA, 앱 소유자, 시스템 관리자

전환

작업	설명	필요한 기술
애플리케이션 클라이언트를 새 인프라로 전환합니다.		DBA, 앱 소유자, 시스템 관리자

프로젝트 닫기

작업	설명	필요한 기술
임시 리소스를 종료합니다.		DBA, 시스템 관리자
프로젝트 문서를 검토하고 검증하세요.		DBA, 앱 소유자, 시스템 관리자
마이그레이션 시간, 수동 작업과 자동 작업의 비율, 비용 절감과 같은 지표를 수집합니다.		DBA, 앱 소유자, 시스템 관리자
프로젝트를 마무리하고 피드백을 제공하세요.		DBA, 앱 소유자, 시스템 관리자

관련 리소스

참조

- [AWS DMS 설명서](#)
- [Amazon Redshift 설명서](#)

- [Amazon Redshift 요금](#)

자습서 및 동영상

- [AWS DMS 시작하기](#)
- [Amazon Redshift 시작하기](#)
- [Amazon Redshift 데이터베이스를 AWS Database Migration Service의 대상으로 사용](#)
- [AWS DMS\(동영상\)](#)

SCT 데이터 추출 에이전트를 사용하여 온프레미스 Microsoft SQL Server 데이터베이스를 Amazon Redshift로 마이그레이션

작성자: Neha Thakur

환경: PoC 또는 파일럿	소스: Microsoft SQL Server	대상: Amazon Redshift
R 유형: 리아키텍트	워크로드: Microsoft	기술: 마이그레이션, 데이터베이스
AWS 서비스: Amazon Redshift, SCT		

요약

이 패턴은 Schema Conversion Tool(SCT) 데이터 추출 에이전트를 사용하여 온프레미스 Microsoft SQL Server 소스 데이터베이스를 Amazon Redshift 대상 데이터베이스로 마이그레이션하는 절차를 설명합니다. 에이전트란 SCT와 통합되지만 다른 곳에서 데이터 변환을 수행하고 사용자를 대신하여 다른 서비스와 상호 작용하는 외부 프로그램입니다.

사전 조건 및 제한 사항

사전 조건

- 온프레미스 데이터 센터의 데이터 웨어하우스 워크로드에 사용되는 Microsoft SQL Server 소스 데이터베이스입니다.
- 활성 상태의 AWS 계정

제품 버전

- Microsoft SQL Server 버전 2008 이상. 지원되는 최신 버전 목록은 [SCT 설명서](#)를 참조하세요.

아키텍처

기술 스택소스

- 온프레미스 Microsoft SQL Server 데이터베이스

기술 스택대상

- Amazon Redshift

데이터 마이그레이션 아키텍처

도구

- [Schema Conversion Tool\(SCT\)](#)은 원본 데이터베이스 스키마와 대부분의 사용자 지정 코드를 대상 데이터베이스와 호환되는 형식으로 자동 변환하여 이기종 데이터베이스 마이그레이션을 처리합니다. 원본 및 대상 데이터베이스가 서로 매우 다른 경우, SCT 에이전트를 사용하여 추가 데이터 변환을 수행할 수 있습니다. 자세한 내용은 설명서의 [온프레미스 데이터 웨어하우스에서 Amazon Redshift로 데이터 마이그레이션](#)을 참조하십시오.

모범 사례

- [SCT의 모범 사례](#)
- [Amazon Redshift를 위한 모범 사례](#)

에픽

마이그레이션 준비

작업	설명	필요한 기술
소스 및 대상 데이터베이스 버전과 엔진을 검증합니다.		DBA
대상 서버 인스턴스의 하드웨어 요구 사항을 식별합니다.		DBA, SysAdmin
스토리지 요구 사항(스토리지 유형 및 용량)을 식별합니다.		DBA, SysAdmin

작업	설명	필요한 기술
적절한 인스턴스 유형(용량, 스토리지 특성, 네트워크 특성)을 선택합니다.		DBA, SysAdmin
소스 및 대상 데이터베이스의 네트워크 액세스 보안 요구 사항을 확인합니다.		DBA, SysAdmin
애플리케이션 마이그레이션 전략을 선택합니다.		DBA SysAdmin, 앱 소유자

인프라 구성

작업	설명	필요한 기술
Virtual Private Cloud(VPC) 및 서브넷을 생성합니다.		SysAdmin
보안 그룹을 생성합니다.		SysAdmin
Amazon Redshift 클러스터를 구성하고 시작합니다.		SysAdmin

데이터 마이그레이션

작업	설명	필요한 기술
SCT 데이터 추출 에이전트를 사용하여 데이터를 마이그레이션합니다.		DBA

애플리케이션 마이그레이션

작업	설명	필요한 기술
선택한 애플리케이션 마이그레이션 전략을 따르세요.		DBA SysAdmin, 앱 소유자

대상 데이터베이스로 전환

작업	설명	필요한 기술
애플리케이션 클라이언트를 새 인프라로 전환하십시오.		DBA SysAdmin, 앱 소유자

프로젝트 닫기

작업	설명	필요한 기술
임시 AWS 리소스를 종료합니다.		DBA, SysAdmin
프로젝트 문서를 검토하고 검증하세요.		DBA SysAdmin, 앱 소유자
마이그레이션 시간, 수동 작업과 자동 작업의 비율, 비용 절감과 같은 지표를 수집합니다.		DBA SysAdmin, 앱 소유자
프로젝트를 종료하고 피드백을 제공하세요.		DBA SysAdmin, 앱 소유자

관련 리소스

참조

- [AWS SCT 사용 설명서](#)

- [데이터 추출 에이전트 사용](#)
- [Amazon Redshift 요금](#)

자습서 및 동영상

- [AWS Schema Conversion Tool 시작하기](#)
- [Amazon Redshift 시작하기](#)

AWS SCT 데이터 추출 에이전트를 사용하여 Teradata 데이터베이스를 Amazon Redshift로 마이그레이션

작성자: Sergey Dmitriev(AWS)

R 유형: 리아키텍트	소스: 데이터베이스: 관계형	대상: Amazon Redshift
작성자: AWS	환경: PoC 또는 파일럿	기술: 데이터베이스, 마이그레이션
AWS 서비스: Amazon Redshift		

요약

이 패턴은 온프레미스 데이터 센터에서 데이터 웨어하우스로 사용되는 Teradata 데이터베이스를 Amazon Redshift 데이터베이스로 마이그레이션하는 단계를 안내합니다. 이 패턴은 AWS Schema Conversion Tool(AWS SCT) 데이터 추출 에이전트를 사용합니다. 에이전트란 AWS SCT와 통합되지만 다른 곳에서 데이터 변환을 수행하고 사용자를 대신하여 다른 AWS 서비스와 상호 작용하는 외부 프로그램입니다.

사전 조건 및 제한 사항

사전 조건

- 활성 상태의 AWS 계정.
- 온프레미스 데이터 센터의 Teradata 소스 데이터베이스

제품 버전

- Teradata 버전 13 이상. 지원되는 최신 버전 목록은 [AWS SCT 설명서](#)를 참조하세요.

아키텍처

소스 기술 스택

- 온프레미스 Teradata 데이터베이스

대상 기술 스택

- Amazon Redshift 클러스터

데이터 마이그레이션 아키텍처

도구

- AWS SCT – [AWS Schema Conversion Tool](#)(AWS SCT)은 원본 데이터베이스 스키마와 대부분의 사용자 지정 코드를 대상 데이터베이스와 호환되는 형식으로 자동 변환하여 이기종 데이터베이스 마이그레이션을 처리합니다. 원본 및 대상 데이터베이스가 서로 매우 다른 경우, AWS SCT 에이전트를 사용하여 추가 데이터 변환을 수행할 수 있습니다. 자세한 내용은 AWS 설명서의 [온프레미스 데이터 웨어하우스에서 Amazon Redshift로 데이터 마이그레이션](#)을 참조하세요.

에픽

마이그레이션 준비

작업	설명	필요한 기술
소스 및 대상 데이터베이스 버전과 엔진을 검증합니다.		DBA
대상 서버 인스턴스의 하드웨어 요구 사항을 식별합니다.		DBA, SysAdmin
스토리지 요구 사항(스토리지 유형 및 용량)을 식별합니다.		DBA, SysAdmin
적절한 인스턴스 유형(용량, 스토리지 특성, 네트워크 특성)을 선택합니다.		DBA, SysAdmin
소스 및 대상 데이터베이스의 네트워크 액세스 보안 요구 사항을 식별합니다.		DBA, SysAdmin

작업	설명	필요한 기술
애플리케이션 마이그레이션 전략을 선택합니다.		DBA SysAdmin, 앱 소유자

인프라 구성

작업	설명	필요한 기술
Virtual Private Cloud(VPC) 및 서브넷을 생성합니다.		SysAdmin
보안 그룹을 생성합니다.		SysAdmin
Amazon Redshift 클러스터를 구성하고 시작합니다.		SysAdmin

데이터 마이그레이션

작업	설명	필요한 기술
AWS SCT 데이터 추출 에이전트를 사용하여 데이터를 마이그레이션합니다.	AWS SCT 데이터 추출 에이전트 사용에 대한 자세한 내용은 참조 및 도움말 섹션의 링크를 참조하세요.	DBA

애플리케이션 마이그레이션

작업	설명	필요한 기술
선택한 애플리케이션 마이그레이션 전략을 따르세요.		DBA SysAdmin, 앱 소유자

대상 Amazon Redshift 데이터베이스로 전환

작업	설명	필요한 기술
애플리케이션 클라이언트를 새 인프라로 전환하세요.		DBA SysAdmin, 앱 소유자

프로젝트 닫기

작업	설명	필요한 기술
임시 AWS 리소스를 종료합니다.		DBA, SysAdmin
프로젝트 문서를 검토하고 검증하세요.		DBA SysAdmin, 앱 소유자
마이그레이션 시간, 수동 작업과 도구 작업의 비율, 비용 절감 등에 대한 지표를 수집하세요.		DBA SysAdmin, 앱 소유자
프로젝트를 종료하고 피드백을 제공하세요.		

관련 리소스

참조

- [AWS SCT 사용 설명서](#)
- [데이터 추출 에이전트 사용](#)
- [Amazon Redshift 요금](#)
- [Teradata RESET WHEN 기능을 Amazon Redshift SQL로 변환\(AWS 권장 가이드\)](#)
- [Teradata NORMALIZE 임시 기능을 Amazon Redshift SQL로 변환\(AWS 권장 가이드\)](#)

자습서

- [AWS Schema Conversion Tool 사용 시작하기](#)
- [Amazon Redshift 시작하기](#)

AWS SCT 데이터 추출 에이전트를 사용하여 온프레미스 Vertica 데이터베이스를 Amazon Redshift로 마이그레이션하기

작성자: Sergey Dmitriev(AWS)

R 유형: 리아키텍트	소스: 데이터베이스: 관계형	대상: Amazon Redshift
작성자: AWS	환경: PoC 또는 파일럿	기술: 데이터베이스, 마이그레이션
AWS 서비스: Amazon Redshift		

요약

이 패턴은 AWS Schema Conversion Tool(AWS SCT) 데이터 추출 에이전트를 사용하여 온프레미스 Vertica 데이터베이스를 Amazon Redshift 클러스터로 마이그레이션하기 위한 지침을 제공합니다. 에이전트란 AWS SCT와 통합되지만 다른 곳에서 데이터 변환을 수행하고 사용자를 대신하여 다른 AWS 서비스와 상호 작용하는 외부 프로그램입니다.

사전 조건 및 제한 사항

사전 조건

- 활성 상태의 AWS 계정.
- 온프레미스 데이터 센터의 데이터 웨어하우스 워크로드에 사용되는 Vertica 소스 데이터베이스입니다.
- Amazon Redshift 대상 클러스터

제품 버전

- Vertica(버전 7.2.2 이상). 지원되는 최신 버전 목록은 [AWS SCT 설명서](#)를 참조하세요.

아키텍처

소스 기술 스택

- 온프레미스 Vertica 데이터베이스

대상 기술 스택

- Amazon Redshift 클러스터

데이터 마이그레이션 아키텍처

도구

- AWS SCT – [AWS Schema Conversion Tool](#)(AWS SCT)은 원본 데이터베이스 스키마와 대부분의 사용자 지정 코드를 대상 데이터베이스와 호환되는 형식으로 자동 변환하여 이기종 데이터베이스 마이그레이션을 처리합니다. 원본 및 대상 데이터베이스가 서로 매우 다른 경우, AWS SCT 에이전트를 사용하여 추가 데이터 변환을 수행할 수 있습니다. 자세한 내용은 AWS 설명서의 [온프레미스 데이터 웨어하우스에서 Amazon Redshift로 데이터 마이그레이션](#)을 참조하세요.

에픽

마이그레이션 준비

작업	설명	필요한 기술
소스 및 대상 데이터베이스 버전을 검증합니다.		DBA
스토리지 요구 사항(스토리지 유형 및 용량)을 식별합니다.		DBA, SysAdmin
적절한 인스턴스 유형(용량, 스토리지 특성, 네트워크 특성)을 선택합니다.		DBA, SysAdmin
원본 및 대상 데이터베이스의 네트워크 액세스 보안 요구 사항을 확인합니다.		DBA, SysAdmin

작업	설명	필요한 기술
애플리케이션 마이그레이션 전략을 선택합니다.		DBA SysAdmin, 앱 소유자

인프라 구성

작업	설명	필요한 기술
Virtual Private Cloud(VPC) 및 서브넷을 생성합니다.		SysAdmin
보안 그룹을 생성합니다.		SysAdmin
Amazon Redshift 클러스터를 구성하고 시작합니다.		SysAdmin

데이터 마이그레이션

작업	설명	필요한 기술
AWS SCT 데이터 추출 에이전트를 사용하여 데이터를 마이그레이션합니다.	AWS SCT 데이터 추출 에이전트 사용에 대한 자세한 내용은 참조 및 도움말 섹션의 링크를 참조하세요.	DBA

애플리케이션 마이그레이션

작업	설명	필요한 기술
선택한 애플리케이션 마이그레이션 전략을 따르세요.		DBA SysAdmin, 앱 소유자

대상 데이터베이스로 전환

작업	설명	필요한 기술
애플리케이션 클라이언트를 새 인프라로 전환하세요.		DBA SysAdmin, 앱 소유자

프로젝트 닫기

작업	설명	필요한 기술
임시 AWS 리소스를 종료합니다.		DBA, SysAdmin
프로젝트 문서를 검토하고 검증하세요.		DBA SysAdmin, 앱 소유자
마이그레이션 시간, 수동 작업과 도구 작업의 비율, 비용 절감 등에 대한 지표를 수집하세요.		DBA SysAdmin, 앱 소유자
프로젝트를 종료하고 피드백을 제공하세요.		

관련 리소스

참조

- [AWS SCT 사용 설명서](#)
- [데이터 추출 에이전트 사용](#)
- [Amazon Redshift 요금](#)

자습서 및 동영상

- [AWS Schema Conversion Tool 시작하기](#)
- [Amazon Redshift 시작하기](#)

레거시 애플리케이션을 Oracle Pro*C에서 ECPG로 마이그레이션

작성자: Sai Parthasaradhi(AWS) 및 Mahesh Balumuri(AWS)

환경: PoC 또는 파일럿	소스: Oracle	대상: PostgreSQL
R 유형: 리아키텍트	워크로드: Oracle	기술: 마이그레이션, 데이터베이스

요약

SQL 코드가 내장된 대부분의 레거시 애플리케이션은 Oracle Pro*C 프리컴파일러를 사용하여 데이터베이스에 액세스합니다. 이러한 Oracle 데이터베이스를 Amazon Relational Database Service(RDS) for PostgreSQL 또는 Amazon Aurora PostgreSQL-Compatible Edition으로 마이그레이션할 때는 애플리케이션 코드를 PostgreSQL의 사전 컴파일러와 호환되는 형식, 즉 ECPG로 변환해야 합니다. 이 패턴은 PostgreSQL ECPG에서 Oracle Pro*C 코드를 해당 코드로 변환하는 방법을 설명합니다.

Pro*C에 대한 자세한 내용은 [Oracle 설명서](#)를 참조하십시오. ECPG에 대한 간략한 소개는 [추가 정보](#) 섹션을 참조하십시오.

사전 조건 및 제한 사항

사전 조건

- 활성 상태의 AWS 계정.
- Amazon RDS for PostgreSQL 또는 Aurora PostgreSQL-Compatible 데이터베이스
- 온프레미스에서 실행되는 오라클 데이터베이스

도구

- 다음 섹션에 나열된 PostgreSql 패키지입니다.
- [AWS CLI](#) – AWS Command Line Interface(AWS CLI)는 명령줄 셸에서 명령을 사용하여 AWS 서비스와 상호 작용하는 오픈 소스 도구입니다. 최소한의 구성으로 명령 프롬프트에서 브라우저 기반 AWS Management Console에서 제공되는 것과 동일한 기능을 구현하는 AWS CLI 명령을 실행할 수 있습니다.

에픽

CentOS 또는 RHEL에서 빌드 환경 설정

작업	설명	필요한 기술
<p>PostgreSQL 패키지를 설치합니다.</p>	<p>다음 명령을 사용하여 필요한 PostgreSQL 패키지를 설치합니다.</p> <pre data-bbox="594 592 1027 1068">yum update -y yum install -y yum- utils rpm -ivh https://d ownload.postgresql .org/pub/repos/yum /repopms/EL-8-x86 _64/pgdg-redhat-repo- latest.noarch.rpm dnf -qy module disable postgresql</pre>	<p>앱 개발자, DevOps 엔지니어</p>
<p>헤더 파일 및 라이브러리를 설치합니다.</p>	<p>다음 명령을 사용하여 헤더 파일 및 라이브러리가 포함된 postgresql12-devel 패키지를 설치합니다. 개발 환경과 런타임 환경 모두에 패키지를 설치하여 런타임 환경에서 오류가 발생하지 않도록 하십시오.</p> <pre data-bbox="594 1516 1027 1753">dnf -y install postgresq l12-devel yum install ncompress zip ghostscript jq unzip wget git -y</pre> <p>개발 환경에서만 다음 명령도 실행합니다.</p>	<p>앱 개발자, DevOps 엔지니어</p>

작업	설명	필요한 기술
	<pre> yum install zlib-devel make -y ln -s /usr/pgsql-12/ bin/ecpg /usr/bin/ </pre>	
<p>환경 경로 변수를 구성합니다.</p>	<p>PostgreSQL 클라이언트 라이브러리의 환경 경로를 설정합니다.</p> <pre> export PATH=\$PATH:/usr/ pgsql-12/bin </pre>	<p>앱 개발자, DevOps 엔지니어</p>

작업	설명	필요한 기술
<p>필요에 따라 추가 소프트웨어를 설치합니다.</p>	<p>필요한 경우 Oracle의 SQL*Loader 대신 pgLoader를 설치하십시오.</p> <pre data-bbox="597 394 1026 911"> wget -O /etc/yum.repos.d/pgloader-ccl.repo https://dl.packager.io/srv/opf/pgloader-ccl/master/installer/el/7.repo yum install pgloader-ccl -y ln -s /opt/pgloader-ccl/bin/pgloader /usr/bin/ </pre> <p>Pro*C 모듈에서 Java 애플리케이션을 호출하려면 Java를 설치하십시오.</p> <pre data-bbox="597 1117 1026 1192"> yum install java -y </pre> <p>ant를 설치하여 자바 코드를 컴파일하십시오.</p> <pre data-bbox="597 1352 1026 1428"> yum install ant -y </pre>	<p>앱 개발자, DevOps 엔지니어</p>

작업	설명	필요한 기술
<p>AWS CLI를 설치합니다.</p>	<p>AWS CLI를 설치하면 명령을 실행하여 애플리케이션에서 AWS Secrets Manager 및 Amazon Simple Storage Service(S3)와 같은 AWS 서비스와 상호 작용할 수 있습니다.</p> <pre data-bbox="594 537 1029 976"> cd /tmp/ curl "https://awscli.amazonaws.com/awscli-exe-linux-x86_64.zip" -o "awscliv2.zip" unzip awscliv2.zip ./aws/install -i /usr/local/aws-cli -b /usr/local/bin --update </pre>	<p>앱 개발자, DevOps 엔지니어</p>
<p>변할할 프로그램이 표시되어야 합니다.</p>	<p>Pro*C에서 ECPG로 변환하려는 애플리케이션을 식별하십시오.</p>	<p>앱 개발자, 앱 소유자</p>

Pro*C 코드를 ECPG로 변환

작업	설명	필요한 기술
<p>불필요한 헤더를 삭제합니다.</p>	<p>PostgreSQL에 필요하지 않은 include 헤더(예: oci.h, oratypes 및 sqllda)를 제거합니다.</p>	<p>앱 소유자, 앱 개발자</p>
<p>변수 선언을 업데이트합니다.</p>	<p>호스트 변수로 사용되는 모든 변수 선언에 EXEC SQL 명령문을 추가합니다.</p>	<p>앱 개발자, 앱 소유자</p>

작업	설명	필요한 기술
	<p>애플리케이션에서 다음과 같은 EXEC SQL VAR 선언을 제거합니다.</p> <pre>EXEC SQL VAR query IS STRING(2048);</pre>	

작업	설명	필요한 기술
<p>ROWNUM 기능을 업데이트합니다.</p>	<p>PostgreSQL에서는 이 ROWNUM 함수를 사용할 수 없습니다. 이 함수를 SQL 쿼리의 ROW_NUMBER 윈도우 함수로 바꾸십시오.</p> <p>Pro*C 코드:</p> <pre data-bbox="592 569 1029 1125"> SELECT SUBSTR(RTRIM(FILE_NAME, '.txt'),12) INTO :gcpclFileseq FROM (SELECT FILE_NAME FROM DEMO_FILES_TABLE WHERE FILE_NAME LIKE '%POC%' ORDER BY FILE_NAME DESC) FL2 WHERE ROWNUM <=1 ORDER BY ROWNUM; </pre> <p>ECPG 코드:</p> <pre data-bbox="592 1236 1029 1845"> SELECT SUBSTR(RTRIM(FILE_NAME, '.txt'),12) INTO :gcpclFileseq FROM (SELECT FILE_NAME , ROW_NUMBER() OVER (ORDER BY FILE_NAME DESC) AS ROWNUM FROM demo_schema.DEMO_FILES_TABLE WHERE FILE_NAME LIKE '%POC%' ORDER BY FILE_NAME DESC) FL2 </pre>	<p>앱 개발자, 앱 소유자</p>

작업	설명	필요한 기술
	<pre>WHERE ROWNUM <=1 ORDER BY ROWNUM;</pre>	
<p>별칭 변수를 사용하도록 함수 파라미터를 업데이트합니다.</p>	<p>PostgreSQL에서는 함수 파라미터를 호스트 변수로 사용할 수 없습니다. 별칭 변수를 사용하여 덮어씁니다.</p> <p>Pro*C 코드:</p> <pre>int processData(int referenceId){ EXEC SQL char col_val[100]; EXEC SQL select column_name INTO :col_val from table_name where col=:referenceId; }</pre> <p>ECPG 코드:</p> <pre>int processData(int referenceIdParam){ EXEC SQL int reference Id = referenceIdParam; EXEC SQL char col_val[100]; EXEC SQL select column_name INTO :col_val from table_name where col=:referenceId; }</pre>	<p>앱 개발자, 앱 소유자</p>

작업	설명	필요한 기술
구조 유형을 업데이트합니다.	<p>struct 유형 변수가 호스트 변수로 사용되는 경우 typedef를 사용하여 EXEC SQL BEGIN 및 END 블록에서 struct 유형을 정의합니다. 헤더(.h) 파일에 struct 유형이 정의된 경우 EXEC SQL 명령문을 사용하여 파일을 포함합니다.</p> <p>Pro*C 코드:</p> <p>헤더 파일(demo.h)</p> <pre data-bbox="594 842 1029 1677"> struct s_partiti on_ranges { char sc_table_ group[31]; char sc_table_ name[31]; char sc_range_ value[10]; }; struct s_partiti on_ranges_ind { short ss_table_ group; short ss_table_ name; short ss_range_ value; }; </pre> <p>ECPG 코드:</p> <p>헤더 파일(demo.h)</p>	앱 개발자, 앱 소유자

작업	설명	필요한 기술
	<pre data-bbox="609 220 1015 1165">EXEC SQL BEGIN DECLARE SECTION; typedef struct { char sc_table_ group[31]; char sc_table_ name[31]; char sc_range_ value[10]; } s_partition_ranges; typedef struct { short ss_table_ group; short ss_table_ name; short ss_range_ value; } s_partition_ranges _ind; EXEC SQL END DECLARE SECTION;</pre> <p data-bbox="592 1197 901 1239">Pro*C 파일(demo.pc)</p> <pre data-bbox="609 1270 1015 1669">#include "demo.h" struct s_partiti on_ranges gc_partit ion_data[MAX_PART_ TABLE] ; struct s_partiti on_ranges_ind gc_partition_data_ ind[MAX_PART_TABLE] ;</pre> <p data-bbox="592 1701 901 1743">ECPG 파일(demo.pc)</p> <pre data-bbox="609 1774 1015 1869">exec sql include "demo.h"</pre>	

작업	설명	필요한 기술
	<pre>EXEC SQL BEGIN DECLARE SECTION; s_partition_ranges gc_partition_data[MAX_PART_TABLE] ; s_partition_ranges_ind gc_partition_data_ ind[MAX_PART_TABLE] ; EXEC SQL END DECLARE SECTION;</pre>	
<p>커서에서 가져오도록 로직을 수정합니다.</p>	<p>배열 변수를 사용하여 커서에서 여러 열을 가져오려면 사용할 코드를 FETCH FORWARD로 변경하십시오.</p> <p>Pro*C 코드:</p> <pre>EXEC SQL char aPoeFiles [MAX_FILES][FILENA ME_LENGTH]; EXEC SQL FETCH filename_ cursor into :aPoeFile s;</pre> <p>ECPG 코드:</p> <pre>EXEC SQL char aPoeFiles [MAX_FILES][FILENA ME_LENGTH]; EXEC SQL int fetchSize = MAX_FILES; EXEC SQL FETCH FORWARD :fetchSiz e filename_cursor into :aPoeFiles;</pre>	<p>앱 개발자, 앱 소유자</p>

작업	설명	필요한 기술
<p>반환 값이 없는 패키지 호출을 수정합니다.</p>	<p>반환 값이 없는 Oracle 패키지 함수는 표시 변수를 사용하여 직접적으로 호출해야 합니다. 애플리케이션에 이름이 같은 함수가 여러 개 포함되어 있거나 알 수 없는 형식 함수로 인해 런타임 오류가 발생하는 경우 값을 데이터 유형으로 타입캐스트합니다.</p> <p>Pro*C 코드:</p> <pre data-bbox="592 758 1027 1356"> void ProcessData (char *data , int id) { EXEC SQL EXECUTE BEGIN pkg_demo. process_data (:data, :id); END; END-EXEC; } </pre> <p>ECPG 코드:</p> <pre data-bbox="592 1465 1027 1877"> void ProcessData (char *dataParam, int idParam) { EXEC SQL char *data = dataParam; EXEC SQL int id = idParam; EXEC SQL short rowInd; </pre>	<p>앱 개발자, 앱 소유자</p>

작업	설명	필요한 기술
	<pre>EXEC SQL short rowInd = 0; EXEC SQL SELECT pkg_demo.process_data (inp_data => :data::te xt, inp_id => :id) INTO :rowInd; }</pre>	

작업	설명	필요한 기술
<p>SQL_CURSOR 변수를 다시 작성하십시오.</p>	<p>SQL_CURSOR 변수와 해당 구현을 다시 작성하십시오.</p> <p>Pro*C 코드:</p> <pre data-bbox="597 428 1027 1020"> /* SQL Cursor */ SQL_CURSOR demo_cursor; EXEC SQL ALLOCATE :demo_cursor; EXEC SQL EXECUTE BEGIN pkg_demo. get_cursor(demo_cur= >:demo_cursor); END; END-EXEC; </pre> <p>ECPG 코드:</p> <pre data-bbox="597 1136 1027 1822"> EXEC SQL DECLARE demo_cursor CURSOR FOR SELECT * from pkg_demo.open_file name_rc(demo_cur= >refcursor); EXEC SQL char open_file name_rcInd[100]; # As the below function returns cursor_name as # return we need to use char[] type as indicator. </pre>	<p>앱 개발자, 앱 소유자</p>

작업	설명	필요한 기술
<p>일반적인 마이그레이션 패턴을 적용합니다.</p>	<pre data-bbox="602 212 1016 464">EXEC SQL SELECT pkg_demo.get_cursor (demo_cur= >'demo_cursor') INTO :open_fil ename_rcInd;</pre> <ul data-bbox="594 506 1024 1482" style="list-style-type: none"> • PostgreSQL과 호환되도록 SQL 쿼리를 변경합니다. • ECPG에서 지원되지 않는 익명 블록을 데이터베이스로 이동합니다. • PostgreSQL에서 지원하지 않는 dbms_application_info 로직을 제거합니다. • 커서가 닫힌 후 EXEC SQL COMMIT 명령문을 이동합니다. 루프 중에 쿼리를 커밋하여 커밋하여 커서에서 레코드를 가져오면 커서가 닫히고 커서가 존재하지 않음 오류가 표시됩니다. • ECPG의 예외 처리 및 오류 코드에 대한 자세한 내용은 PostgreSQL 설명서의 오류 처리를 참조하십시오. 	<p>앱 개발자, 앱 소유자</p>
<p>필요한 경우 디버깅을 활성화합니다.</p>	<p>ECPG 프로그램을 디버그 모드에서 실행하려면 기본 함수 블록 내에 다음 명령을 추가합니다.</p> <pre data-bbox="602 1745 1016 1818">ECPGdebug(1, stderr);</pre>	<p>앱 개발자, 앱 소유자</p>

ECPG 프로그램 컴파일

작업	설명	필요한 기술
<p>ECPG용 실행 파일을 생성합니다.</p>	<p>이름이 지정된 prog1.pgc 임베디드 SQL C 소스 파일이 있는 경우 다음 명령 시퀀스를 사용하여 실행 프로그램을 만들 수 있습니다.</p> <pre data-bbox="594 596 1029 873"> eapg prog1.pgc cc -I/usr/local/pgsql/ include -c prog1.c cc -o prog1 prog1.o -L/ usr/local/pgsql/lib - leapg </pre>	<p>앱 개발자, 앱 소유자</p>
<p>컴파일할 메이크 파일을 생성합니다.</p>	<p>다음 샘플 파일과 같이 make 파일이 생성되어 ECPG 프로그램이 표시되어야 합니다.</p> <pre data-bbox="594 1079 1029 1835"> CFLAGS ::= \$(CFLAGS) -I/ usr/pgsql-12/include - g -Wall LDFLAGS ::= \$(LDFLAGS) -L/usr/pgsql-12/li b -Wl,-rpath,/usr/pg sql-12/lib LDLIBS ::= \$(LDLIBS) - leapg PROGRAMS = test .PHONY: all clean %.c: %.pgc eapg \$< all: \$(PROGRAMS) clean: rm -f \$(PROGRAM S) \$(PROGRAMS:%=%.c) \$(PROGRAMS:%=%.o) </pre>	<p>앱 개발자, 앱 소유자</p>

애플리케이션 테스트

작업	설명	필요한 기술
코드를 테스트합니다.	변환된 애플리케이션 코드를 테스트하여 제대로 작동하는지 확인하십시오.	앱 개발자, 앱 소유자, 테스트 엔지니어

관련 리소스

- [ECPG - C의 임베디드 SQL](#)(PostgreSQL 설명서)
- [오류 처리](#) (PostgreSQL 설명서)
- [Oracle Pro*C/C++ 프리컴파일러를 사용하는 이유](#)(Oracle 설명서)

추가 정보

PostgreSQL에는 Oracle Pro*C 프리컴파일러와 동일한 임베디드 SQL 프리컴파일러인 ECPG가 있습니다. ECPG는 SQL 호출을 특수 함수 호출로 대체하여 임베디드 SQL 명령문이 있는 C 프로그램을 표준 C 코드로 변환합니다. 그러면 모든 C 컴파일러 툴 체인을 사용하여 출력 파일을 처리할 수 있습니다.

입력 및 출력 파일

ECPG는 명령줄에서 지정하는 각 입력 파일을 해당 C 출력 파일로 변환합니다. 입력 파일 이름에 파일 확장자가 없는 경우에는 .pgc로 간주됩니다. 파일 확장자는 .c로 대체되어 출력 파일 이름을 구성합니다. 그러나 -o 옵션을 사용하여 기본 출력 파일 이름을 재정의할 수 있습니다.

대시(-)를 입력 파일 이름으로 사용하는 경우 -o 옵션을 사용하여 재정의하지 않는 한 ECPG는 표준 입력에서 프로그램을 읽고 표준 출력에 씁니다.

헤더 파일

PostgreSQL 컴파일러는 사전 처리된 C 코드 파일을 컴파일할 때 PostgreSQL include 디렉터리에서 ECPG 헤더 파일을 찾습니다. 따라서 컴파일러가 올바른 디렉터리(예: -I/usr/local/pgsql/include)를 가리키도록 -I 옵션을 사용해야 할 수도 있습니다.

Libraries

임베디드 SQL을 포함한 C 코드를 사용하는 프로그램은 libecpg 라이브러리에 연결해야 합니다. 예를 들어 링커 옵션 `-L/usr/local/pgsql/lib -lecpg`을 사용할 수 있습니다.

변환된 ECPG 애플리케이션은 내장된 SQL libpq 라이브러리(ecpglib)를 통해 라이브러리의 함수를 직접적으로 호출하고 표준 프런트 엔드/백엔드 프로토콜을 사용하여 PostgreSQL 서버와 통신합니다.

가상으로 생성된 열을 오라클에서 PostgreSQL로 마이그레이션

제작: Veeranjanyulu Grandhi(AWS), Rajesh Madiwale(AWS), Ramesh Pathuri(AWS)

환경: 프로덕션	소스: Oracle 데이터베이스	대상: Amazon RDS for PostgreSQL 또는 Aurora PostgreSQL 호환
R 유형: 리아키텍트	워크로드: Oracle	기술: 마이그레이션; 데이터베이스
AWS 서비스: Amazon Aurora; Amazon RDS; AWS DMS		

요약

버전 11 이하에서는 PostgreSQL이 오라클 가상화 열과 직접적으로 동일한 특성을 제공하지 않습니다. 오라클 데이터베이스에서 PostgreSQL 버전 11 이하로 마이그레이션하는 동안 가상으로 생성된 열을 처리하는 것이 어려운 이유는 두 가지입니다.

- 마이그레이션 중에는 가상화 열이 보이지 않습니다.
- PostgreSQL은 12보다 낮은 버전에서는 generate 표현식을 지원하지 않습니다.

그러나 유사한 기능을 에뮬레이트하는 해결 방법이 있습니다. AWS Database Migration Service(AWS DMS)를 사용하여 오라클 데이터베이스에서 PostgreSQL 버전 11 이하로 데이터를 이전하면 트리거 함수를 사용하여 가상으로 생성된 열의 값을 채울 수 있습니다. 이 패턴은 이러한 목적으로 사용할 수 있는 오라클 데이터베이스 및 PostgreSQL 코드의 예를 제공합니다. AWS에서는 PostgreSQL용 Amazon Relational Database Service(RDS)를 사용하거나 PostgreSQL 데이터베이스에 Amazon Aurora PostgreSQL-Compatible Edition을 사용할 수 있습니다.

PostgreSQL 버전 12부터는 생성된 열이 지원됩니다. 생성된 열은 다른 열 값에서 바로 계산하거나 계산하여 저장할 수 있습니다. [PostgreSQL에서 생성된 열](#)은 오라클 가상화 열과 유사합니다.

사전 조건 및 제한 사항

사전 조건

- 활성 상태의 AWS 계정

- 소스 오라클 데이터베이스
- 대상 PostgreSQL 데이터베이스(Amazon RDS for PostgreSQL 또는 Aurora PostgreSQL 호환 가능)
- [PL/pgSQL](#) 코딩 전문성

제한 사항

- PostgreSQL 버전 12 이전에만 적용됩니다.
- 오라클 데이터베이스 버전 11g 이상에만 적용됩니다.
- 가상화 열은 데이터 마이그레이션 도구에서 지원되지 않습니다.
- 동일한 테이블에 정의된 열에만 적용됩니다.
- 가상으로 생성된 열이 결정론적 사용자 정의 함수를 참조하는 경우 파티셔닝 키 열로 사용할 수 없습니다.
- 표현식 출력은 스칼라 값이어야 합니다. 오라클에서 제공한 데이터 유형, 사용자 정의 유형, LOB, 또는 LONG RAW 반환은 안 됩니다.
- 가상화 열에 대해 정의된 인덱스는 PostgreSQL의 함수 기반 인덱스와 동일합니다.
- 테이블 통계를 수집해야 합니다.

도구

- [pgAdmin 4](#)는 PostgreSQL용 오픈 소스 관리 도구입니다. 이 도구는 데이터베이스 개체의 생성, 유지 관리 및 사용을 간소화하는 그래픽 인터페이스를 제공합니다.
- [Oracle SQL Developer](#)는 기존 배포 및 클라우드 배포 모두에서 오라클 데이터베이스의 SQL 작업을 위한 무료 통합 개발 환경입니다.

에픽

소스 및 타겟 데이터베이스 테이블 생성

작업	설명	필요한 기술
소스 오라클 데이터베이스 테이블을 만듭니다.	오라클 데이터베이스에서 다음 문을 사용하여 가상으로 생성된 열이 있는 테이블을 생성합니다.	DBA, 앱 개발자

작업	설명	필요한 기술
	<pre data-bbox="592 220 1031 724">CREATE TABLE test.generated_column (CODE NUMBER, STATUS VARCHAR2(12) DEFAULT 'PreOpen', FLAG CHAR(1) GENERATED ALWAYS AS (CASE UPPER(STATUS) WHEN 'OPEN' THEN 'N' ELSE 'Y' END) VIRTUAL VISIBLE);</pre> <p data-bbox="592 756 1031 1375">이 소스 테이블에서 STATUS 열에 있는 데이터는 AWS DMS를 통해 대상 데이터베이스로 마이그레이션됩니다. 그러나 이 FLAG 열은 generate by 기능을 사용하여 채워지므로 마이그레이션 중에는 이 열이 AWS DMS에 표시되지 않습니다. generated by의 기능을 구현하려면 다음 예픽과 같이 대상 데이터베이스에서 트리거와 함수를 사용하여 FLAG 열 값을 채워야 합니다.</p>	

작업	설명	필요한 기술
AWS에서 대상 PostgreSQL 테이블을 생성합니다.	<p>다음 문을 사용하여 AWS에서 PostgreSQL 테이블을 만듭니다.</p> <pre>CREATE TABLE test.generated_column (code integer not null, status character varying(12) not null , flag character(1));</pre> <p>이 표에서 status 열은 표준 열입니다. flag 열은 열에 있는 데이터를 기반으로 생성된 status 열이 됩니다.</p>	DBA, 앱 개발자

PostgreSQL에서 가상화 열을 처리하는 트리거 함수 만들기

작업	설명	필요한 기술
PostgreSQL 트리거를 만듭니다.	<p>PostgreSQL에서 트리거를 만듭니다.</p> <pre>CREATE TRIGGER tgr_generated_column AFTER INSERT OR UPDATE OF status ON test.generated_column FOR EACH ROW EXECUTE FUNCTION test.tgf_generated_column();</pre>	DBA, 앱 개발자

작업	설명	필요한 기술
<p>PostgreSQL 트리거 함수를 만듭니다.</p>	<p>PostgreSQL에서 트리거에 대한 함수를 만듭니다. 이 함수는 애플리케이션 또는 AWS DMS에 의해 삽입되거나 업데이트되는 가상화 열을 채우고 데이터 유효성 검사를 수행합니다.</p> <pre data-bbox="597 537 1029 1822"> CREATE OR REPLACE FUNCTION test.tgf_ gen_column() RETURNS trigger AS \$VIRTUAL_ COL\$ BEGIN IF (TG_OP = 'INSERT') THEN IF (NEW.flag IS NOT NULL) THEN RAISE EXCEPTION 'ERROR: cannot insert into column "flag" USING DETAIL = 'Column "flag" is a generated column.'; END IF; END IF; IF (TG_OP = 'UPDATE') THEN IF (NEW.flag::VARCHAR ! = OLD.flag::varchar) THEN RAISE EXCEPTION 'ERROR: cannot update column "flag" USING DETAIL = 'Column "flag" is a generated column.'; END IF; END IF; IF TG_OP IN ('INSERT' , 'UPDATE') THEN </pre>	<p>DBA, 앱 개발자</p>

작업	설명	필요한 기술
	<pre> IF (old.flag is NULL) OR (coalesce(old.stat us, '') != coalesce(new.status, '')) THEN UPDATE test.gene rated_column SET flag = (CASE UPPER(status) WHEN 'OPEN' THEN 'N' ELSE 'Y' END) WHERE code = new.code; END IF; END IF; RETURN NEW; END \$VIRTUAL_COL\$ LANGUAGE plpgsql; </pre>	

AWS DMS를 사용하여 데이터 마이그레이션 테스트

작업	설명	필요한 기술
복제 인스턴스를 만듭니다.	복제 인스턴스를 만들려면 AWS DMS 문서에 있는 지침 을 따르세요. 복제 인스턴스가 원본 및 대상 데이터베이스와 동일한 Virtual Private Cloud(VPC)에 있어야 합니다.	DBA, 앱 개발자
원본 및 대상 DB 엔드포인트를 만듭니다.	엔드포인트를 만들려면 AWS DMS 문서에 있는 지침 을 따르세요.	DBA, 앱 개발자
엔드포인트 연결을 테스트합니다.	VPC와 복제 인스턴스를 지정하고 테스트 실행을 선택하면 엔드포인트 연결을 테스트할 수 있습니다.	DBA, 앱 개발자

작업	설명	필요한 기술
전체 로드 작업을 생성하고 시작합니다.	자세한 내용은 AWS DMS 문서에서 작업 만들기 및 작업 폴로드 설정 을 참조하세요.	DBA, 앱 개발자
가상화 열의 데이터 유효성을 검사합니다.	원본 데이터베이스와 대상 데이터베이스의 가상화 열에 있는 데이터를 비교합니다. 이 단계에서 데이터 유효성 검사를 수동으로 수행하거나 스크립트를 작성할 수 있습니다.	DBA, 앱 개발자

관련 리소스

- [AWS Database Migration Service 시작하기](#)(AWS DMS 설명서)
- [오라클 데이터베이스를 AWS DMS 소스로 사용](#)(AWS DMS 문서)
- [AWS DMS의 대상으로 PostgreSQL 데이터베이스 사용](#) (AWS DMS 문서)
- [PostgreSQL에서 생성된 열](#)(PostgreSQL 문서)
- [트리거 함수](#) (PostgreSQL 문서)
- [가상화 열](#)(오라클 문서)

Aurora PostgreSQL 호환에서 Oracle UTL_FILE 기능 설정

작성자: 라케시 라가브(AWS)와 아누라다 친타(AWS)

환경: PoC 또는 파일럿	소스: Oracle	대상: Aurora PostgreSQL
R 유형: 리아키텍트	워크로드: Oracle	기술: 마이그레이션; 인프라; 데이터베이스
AWS 서비스: Amazon S3, Amazon Aurora		

요약

Oracle에서 Amazon Web Services(AWS) 클라우드의 Amazon Aurora PostgreSQL 호환 버전으로 마이그레이션하는 과정에서 여러 가지 문제가 발생할 수 있습니다. 예를 들어, Oracle UTL_FILE 유틸리티를 사용하는 코드를 마이그레이션하는 것은 항상 어려운 일입니다. Oracle PL/SQL에서 UTL_FILE 패키지는 기본 운영 체제와 함께 읽기 및 쓰기와 같은 파일 작업에 사용됩니다. 이 UTL_FILE 유틸리티는 서버 및 클라이언트 기기 시스템 모두에서 작동합니다.

Amazon Aurora PostgreSQL 호환은 관리형 데이터베이스 오퍼링입니다. 이 때문에 데이터베이스 서버의 파일에 액세스할 수 없습니다. 이 패턴은 Amazon Simple Storage Service(S3)와 Amazon Aurora PostgreSQL 호환의 통합 과정을 안내하여 UTL_FILE 기능의 일부를 구현합니다. 이 통합을 사용하면 타사 추출, 전환, 적재(ETL) 도구 또는 서비스를 사용하지 않고도 파일을 생성하고 사용할 수 있습니다.

선택적으로 Amazon CloudWatch 모니터링 및 Amazon SNS 알림을 설정할 수 있습니다.

프로덕션 환경에 구현하기 전에 이 솔루션을 철저히 테스트하는 것이 좋습니다.

사전 조건 및 제한 사항

사전 조건

- 활성 상태의 AWS 계정
- AWS Database Migration Service(AWS DMS) 전문 지식
- PL/PgSQL 코딩에 대한 전문 지식

- Amazon Aurora PostgreSQL 호환 클러스터
- S3 버킷

제한 사항

이 패턴은 Oracle UTL_FILE 유틸리티를 대체하는 기능을 제공하지 않습니다. 하지만 데이터베이스 현대화 목표를 달성하기 위해 단계와 샘플 코드를 더욱 개선할 수 있습니다.

제품 버전

- Amazon Aurora PostgreSQL 호환 버전 11.9

아키텍처

대상 기술 스택

- Amazon Aurora PostgreSQL 호환
- 아마존 CloudWatch
- Amazon Simple Notification Service(Amazon SNS)
- Amazon S3

대상 아키텍처

다음은 솔루션의 고급 표현을 보이는 다이어그램입니다.

1. 파일은 애플리케이션에서 S3 버킷으로 업로드됩니다.
2. aws_s3 확장은 PL/pgSQL을 사용하여 데이터에 액세스하고 데이터를 Aurora PostgreSQL 호환에 업로드합니다.

도구

- [Amazon Aurora PostgreSQL 호환 버전](#)-Amazon Aurora PostgreSQL 호환 버전은 완전 관리형으로, PostgreSQL 호환 및 ACID 준수 관계형 데이터베이스 엔진입니다. 이는 고급 상용 데이터베이스의 속도와 신뢰성을 오픈 소스 데이터베이스의 비용 효율성과 결합합니다.

- [AWS CLI](#)-AWS Command Line Interface(AWS CLI)는 AWS 서비스를 관리하는 통합 도구입니다. 도구 하나만 다운로드하여 구성하면 여러 AWS 서비스를 명령줄에서 관리하고 스크립트를 통해 자동화할 수 있습니다.
- [아마존 CloudWatch](#) — 아마존은 아마존 S3 리소스 및 사용을 CloudWatch 모니터링합니다.
- [Amazon S3](#) – Amazon Simple Storage Service(S3)는 인터넷에 대한 스토리지입니다. 이 패턴에서 Amazon S3는 Aurora PostgreSQL 호환 클러스터를 나가고 들어오는 소비 및 전송 파일을 수신하고 저장하는 스토리지 계층을 제공합니다.
- [aws_s3](#)-aws_s3 확장은 Amazon S3와 Aurora PostgreSQL 호환을 통합합니다.
- [Amazon SNS](#)-Amazon Simple Notification Service(SNS)는 게시자와 클라이언트에게 메시지를 전달 또는 전송하는 것을 조정하고 관리합니다. 이 패턴에서는 Amazon SNS를 사용하여 알림을 전송합니다.
- [pgAdmin](#)-pgAdmin은 Postgres를 위한 오픈 소스 관리 도구입니다. pgAdmin 4는 데이터베이스 개체를 생성, 유지 관리 및 사용하기 위한 그래픽 인터페이스를 제공합니다.

코드

필요한 기능을 구현하기 위해 패턴은 UTL_FILE과 비슷한 이름을 가진 여러 함수를 생성합니다. 추가 정보 섹션에는 이러한 함수의 코드 베이스가 포함되어 있습니다.

코드에서 testaurorabucket을 테스트 S3 버킷의 이름으로 바꿉니다. us-east-1을 테스트 S3 버킷이 위치한 AWS 리전으로 바꿉니다.

에픽

Amazon S3와 Aurora PostgreSQL 호환 통합

작업	설명	필요한 기술
IAM 정책을 설정합니다.	S3 버킷에 대한 액세스 권한을 부여하는 AWS Identity and Access Management(IAM) 정책 및 정책 내의 객체를 생성합니다. 코드에 대한 내용은 추가 정보 섹션을 참조하세요.	AWS 관리자, DBA
Aurora PostgreSQL에 Amazon S3 액세스 역할을 추가합니다.	두 개의 IAM 역할, 즉 하나는 Amazon S3에 대한 읽기 액세스	AWS 관리자, DBA

작업	설명	필요한 기술
	<p>스 역할이고 다른 하나는 쓰기 액세스 역할을 생성합니다. Aurora PostgreSQL 호환 클러스터에 다음의 두 가지 역할을 연결합니다.</p> <ul style="list-style-type: none"> • S3 내보내기 기능을 위한 역할 하나 • S3 불러오기 기능의 역할 하나 <p>자세한 내용은 Amazon S3로 데이터를 가져오고 내보내는 방법에 대한 Aurora PostgreSQL 호환 설명서를 참조하세요.</p>	

Aurora PostgreSQL 호환에서 확장 설정

작업	설명	필요한 기술
aws_commons 확장을 생성합니다.	aws_commons 확장은 aws_s3 확장의 종속성입니다.	DBA, 개발자
aws_s3 확장을 생성합니다.	aws_s3 확장은 Amazon S3와 상호 작용합니다.	DBA, 개발자

Amazon S3와 Aurora PostgreSQL 호환 통합 검증

작업	설명	필요한 기술
Amazon S3에서 Aurora PostgreSQL로 파일 가져오기를 테스트합니다.	파일을 Aurora PostgreSQL 호환으로 가져오는 것을 테스트하려면 샘플 CSV 파일을 생성	DBA, 개발자

작업	설명	필요한 기술
	하여 S3 버킷에 업로드하세요. CSV 파일을 기반으로 테이블 정의를 생성하고 <code>aws_s3.table_import_from_s3</code> 함수를 사용하여 파일을 테이블에 로드합니다.	
Aurora PostgreSQL에서 Amazon S3로 파일 내보내기를 테스트합니다.	파일을 Aurora PostgreSQL 호환으로 내보내는 것을 테스트하려면 테스트 테이블을 생성하고 데이터로 채운 후에 <code>aws_s3.query_export_to_s3</code> 함수를 이용하여 데이터를 내보냅니다.	DBA, 개발자

UTL_FILE 유틸리티를 모방하려면 래퍼 함수를 생성하세요.

작업	설명	필요한 기술
utl_file_utility 스키마를 생성합니다.	스키마는 래퍼 함수를 함께 유지합니다. 스키마를 생성하려면 다음의 명령을 실행하세요.	DBA, 개발자
	<pre>CREATE SCHEMA utl_file_utility;</pre>	
file_type 유형을 생성합니다.	file_type 유형을 생성하려면, 다음의 코드를 사용하세요.	DBA/개발자
	<pre>CREATE TYPE utl_file_utility.file_type AS (p_path character varying(30), p_file_name character varying);</pre>	

작업	설명	필요한 기술
init 함수를 생성합니다.	이 init 함수는 bucket 또는 region과 같은 공통 변수를 초기화합니다. 코드에 대한 내용은 추가 정보 섹션을 참조하세요.	DBA/개발자
래퍼 함수를 생성합니다.	래퍼 함수 fopen, put_line 및 fclose를 생성합니다. 코드에 대한 내용은 추가 정보 섹션을 참조하세요.	DBA, 개발자

래퍼 함수 테스트

작업	설명	필요한 기술
쓰기 모드에서 래퍼 함수를 테스트합니다.	쓰기 모드에서 래퍼 함수를 테스트하려면 추가 정보 섹션에 제공된 코드를 사용하세요.	DBA, 개발자
추가 모드에서 래퍼 함수를 테스트합니다.	추가 모드에서 래퍼 함수를 테스트하려면 추가 정보 섹션에 제공된 코드를 사용하세요.	DBA, 개발자

관련 리소스

- [Amazon S3 통합](#)
- [Amazon S3](#)
- [Aurora](#)
- [아마존 CloudWatch](#)
- [Amazon SNS](#)

추가 정보

IAM 정책 설정

다음의 정책을 생성합니다.

정책 이름

S3 IntRead

JSON

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "S3integrationtest",
      "Effect": "Allow",
      "Action": [
        "s3:GetObject",
        "s3:ListBucket"
      ],
      "Resource": [
        "arn:aws:s3:::testaurorabuc
ket/*",
        "arn:aws:s3:::testaurorabuc
ket"
      ]
    }
  ]
}
```

S3 IntWrite

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "S3integrationtest",
      "Effect": "Allow",
      "Action": [
        "s3:PutObject",
        "s3:ListBucket"
      ],
    }
  ]
}
```

```

        "Resource": [
            "arn:aws:s3:::testaurorabucket/",
            "arn:aws:s3:::testaurorabucket"
        ]
    }
}

```

init 함수 생성

bucket 또는 region과 같은 일반 변수를 초기화하려면 다음 코드를 사용하여 init 함수를 생성하세요.

```

CREATE OR REPLACE FUNCTION utl_file_utility.init(
)
RETURNS void
LANGUAGE 'plpgsql'

COST 100
VOLATILE
AS $BODY$
BEGIN
    perform set_config
    ( format( '%s.%s', 'UTL_FILE_UTILITY', 'region' )
    , 'us-east-1'::text
    , false );

    perform set_config
    ( format( '%s.%s', 'UTL_FILE_UTILITY', 's3bucket' )
    , 'testaurorabucket'::text
    , false );
END;
$BODY$;

```

래퍼 함수 생성

fopen, put_line 및 fclose 래퍼 함수를 생성합니다.

fopen

```

CREATE OR REPLACE FUNCTION utl_file_utility.fopen(

```

```

    p_file_name character varying,
    p_path character varying,
    p_mode character DEFAULT 'W'::bpchar,
    OUT p_file_type utl_file_utility.file_type)
    RETURNS utl_file_utility.file_type
    LANGUAGE 'plpgsql'

    COST 100
    VOLATILE
AS $BODY$
declare
    v_sql character varying;
    v_cnt_stat integer;
    v_cnt integer;
    v_tabname character varying;
    v_filewithpath character varying;
    v_region character varying;
    v_bucket character varying;

BEGIN
    /*initialize common variable */
    PERFORM utl_file_utility.init();
    v_region := current_setting( format( '%s.%s', 'UTL_FILE_UTILILITY', 'region' ) );
    v_bucket := current_setting( format( '%s.%s', 'UTL_FILE_UTILILITY', 's3bucket' ) );

    /* set tabname*/
    v_tabname := substring(p_file_name,1,case when strpos(p_file_name, '.') = 0 then
length(p_file_name) else strpos(p_file_name, '.') - 1 end );
    v_filewithpath := case when NULLif(p_path, '') is null then p_file_name else
concat_ws('/',p_path,p_file_name) end ;
    raise notice 'v_bucket %, v_filewithpath % , v_region %', v_bucket,v_filewithpath,
v_region;

    /* APPEND MODE HANDLING; RETURN EXISTING FILE DETAILS IF PRESENT ELSE CREATE AN
EMPTY FILE */
    IF p_mode = 'A' THEN
        v_sql := concat_ws('','create temp table if not exists ', v_tabname,' (col1
text)');
        execute v_sql;

        begin
            PERFORM aws_s3.table_import_from_s3
                ( v_tabname,
                  '',

```

```

        'DELIMITER AS ''#''',
        aws_commons.create_s3_uri
        (
            v_bucket,
            v_filewithpath ,
            v_region)
        );
    exception
        when others then
            raise notice 'File load issue ,%',sqlerrm;
            raise;
    end;
    execute concat_ws('','select count(*) from ',v_tabname) into v_cnt;

    IF v_cnt > 0
    then
        p_file_type.p_path := p_path;
        p_file_type.p_file_name := p_file_name;
    else
        PERFORM aws_s3.query_export_to_s3('select ''''',
            aws_commons.create_s3_uri(v_bucket, v_filewithpath,
v_region)
                );

        p_file_type.p_path := p_path;
        p_file_type.p_file_name := p_file_name;
    end if;
    v_sql := concat_ws('','drop table ', v_tabname);
    execute v_sql;
    ELSEIF p_mode = 'W' THEN
        PERFORM aws_s3.query_export_to_s3('select ''''',
            aws_commons.create_s3_uri(v_bucket, v_filewithpath,
v_region)
                );
        p_file_type.p_path := p_path;
        p_file_type.p_file_name := p_file_name;
    END IF;

EXCEPTION
    when others then
        p_file_type.p_path := p_path;
        p_file_type.p_file_name := p_file_name;
        raise notice 'fopenerror,%',sqlerrm;
        raise;

END;
```

```
$BODY$;
```

put_line

```
CREATE OR REPLACE FUNCTION utl_file_utility.put_line(
    p_file_name character varying,
    p_path character varying,
    p_line text,
    p_flag character DEFAULT 'W'::bpchar)
    RETURNS boolean
    LANGUAGE 'plpgsql'

    COST 100
    VOLATILE
AS $BODY$
/*****
 * Write line, p_line in windows format to file, p_fp - with carriage return
 * added before new line.
 *****/
declare
    v_sql varchar;
    v_ins_sql varchar;
    v_cnt INTEGER;
    v_filewithpath character varying;
    v_tabname character varying;
    v_bucket character varying;
    v_region character varying;

BEGIN
    PERFORM utl_file_utility.init();

    /* check if temp table already exist */

    v_tabname := substring(p_file_name,1,case when strpos(p_file_name, '.') = 0 then
length(p_file_name) else strpos(p_file_name, '.') - 1 end );

    v_sql := concat_ws('','select count(1) FROM pg_catalog.pg_class c LEFT JOIN
pg_catalog.pg_namespace n ON n.oid = c.relnamespace where n.nspname like 'pg_temp_
%'
                , ' AND pg_catalog.pg_table_is_visible(c.oid) AND
Upper(relname) = Upper(
                , v_tabname ,'' ) ');
```

```

execute v_sql into v_cnt;

IF v_cnt = 0 THEN
    v_sql := concat_ws('','create temp table ',v_tabname,' (col text)');
    execute v_sql;
    /* CHECK IF APPEND MODE */
    IF upper(p_flag) = 'A' THEN
        PERFORM utl_file_utility.init();
        v_region := current_setting( format( '%s.%s', 'UTL_FILE_UTILITY',
'region' ) );
        v_bucket := current_setting( format( '%s.%s', 'UTL_FILE_UTILITY',
's3bucket' ) );

        /* set tabname*/
        v_filewithpath := case when NULLif(p_path,'') is null then p_file_name else
concat_ws('/',p_path,p_file_name) end ;

        begin
            PERFORM aws_s3.table_import_from_s3
                ( v_tabname,
                  '',
                  'DELIMITER AS '#'',
                  aws_commons.create_s3_uri
                    ( v_bucket,
                      v_filewithpath,
                      v_region
                    )
                );
        exception
            when others then
                raise notice 'Error Message : %',sqlerrm;
                raise;
        end;
    END IF;
END IF;
/* INSERT INTO TEMP TABLE */
v_ins_sql := concat_ws('','insert into ',v_tabname,' values('',p_line,'')');
execute v_ins_sql;
RETURN TRUE;
exception
    when others then
        raise notice 'Error Message : %',sqlerrm;
        raise;
END;

```

```
$BODY$;
```

fclose

```
CREATE OR REPLACE FUNCTION utl_file_utility.fclose(
  p_file_name character varying,
  p_path character varying)
  RETURNS boolean
  LANGUAGE 'plpgsql'

  COST 100
  VOLATILE
AS $BODY$
DECLARE
  v_filewithpath character varying;
  v_bucket character varying;
  v_region character varying;
  v_tabname character varying;
  v_sql character varying;
BEGIN
  PERFORM utl_file_utility.init();

  v_region := current_setting( format( '%s.%s', 'UTL_FILE_UTILITY', 'region' ) );
  v_bucket := current_setting( format( '%s.%s', 'UTL_FILE_UTILITY', 's3bucket' ) );

  v_tabname := substring(p_file_name,1,case when strpos(p_file_name, '.') = 0 then
length(p_file_name) else strpos(p_file_name, '.') - 1 end );
  v_filewithpath := case when NULLIF(p_path, '') is null then p_file_name else
concat_ws('/',p_path,p_file_name) end ;

  raise notice 'v_bucket %, v_filewithpath % , v_region %', v_bucket,v_filewithpath,
v_region ;

  /* exporting to s3 */
  perform aws_s3.query_export_to_s3
    (concat_ws('','select * from ',v_tabname,' order by ctid asc'),
    aws_commons.create_s3_uri(v_bucket, v_filewithpath, v_region)
    );
  v_sql := concat_ws('','drop table ', v_tabname);
  execute v_sql;
  RETURN TRUE;
EXCEPTION
  when others then
```

```

        raise notice 'error fclose %',sqlerrm;
    RAISE;
END;
$BODY$;

```

래퍼 함수 설정 및 테스트

다음 익명 코드 블록을 사용하여 설정을 테스트합니다.

쓰기 모드 테스트

다음 코드는 S3 버킷에 s3inttest라는 파일을 작성합니다.

```

do $$
declare
l_file_name varchar := 's3inttest' ;
l_path varchar := 'integration_test' ;
l_mode char(1) := 'W';
l_fs utl_file_utility.file_type ;
l_status boolean;

begin
select * from
utl_file_utility.fopen( l_file_name, l_path , l_mode ) into l_fs ;
raise notice 'fopen : l_fs : %', l_fs;

select * from
utl_file_utility.put_line( l_file_name, l_path , 'this is test file:in s3bucket: for
test purpose', l_mode ) into l_status ;
raise notice 'put_line : l_status %', l_status;

select * from utl_file_utility.fclose( l_file_name , l_path ) into l_status ;
raise notice 'fclose : l_status %', l_status;

end;
$$

```

추가 모드 테스트

다음 코드는 이전 테스트에서 만든 s3inttest 파일에 줄을 추가합니다.

```

do $$
declare

```

```
l_file_name varchar := 's3intttest' ;
l_path varchar := 'integration_test' ;
l_mode char(1) := 'A';
l_fs utl_file_utility.file_type ;
l_status boolean;

begin
select * from
utl_file_utility.fopen( l_file_name, l_path , l_mode ) into l_fs ;
raise notice 'fopen : l_fs : %', l_fs;

select * from
utl_file_utility.put_line( l_file_name, l_path , 'this is test file:in s3bucket: for
test purpose : append 1', l_mode ) into l_status ;
raise notice 'put_line : l_status %', l_status;

select * from
utl_file_utility.put_line( l_file_name, l_path , 'this is test file:in s3bucket : for
test purpose : append 2', l_mode ) into l_status ;
raise notice 'put_line : l_status %', l_status;

select * from utl_file_utility.fclose( l_file_name , l_path ) into l_status ;
raise notice 'fclose : l_status %', l_status;

end;
$$
```

Amazon SNS 알림

선택적으로 S3 버킷에서 Amazon CloudWatch 모니터링 및 Amazon SNS 알림을 설정할 수 있습니다. 자세한 내용은 [Amazon S3 모니터링](#) 및 [Amazon SNS 알림 설정](#)을 참조하세요.

오라클에서 Amazon Aurora PostgreSQL로 마이그레이션한 다음 데이터베이스 객체를 검증합니다.

Venkatramana Chintha(AWS)와 Eduardo Valentim(AWS)이 작성

R 유형: 리아키텍트	소스: 관계형	대상: Amazon Aurora PostgreSQL, Amazon RDS for PostgreSQL
작성자: AWS	환경: PoC 또는 파일럿	기술: 데이터베이스; 마이그레이션
워크로드: Oracle	AWS 서비스: Amazon Aurora	

요약

이 패턴은 Oracle 데이터베이스를 Amazon Aurora PostgreSQL 호환 에디션으로 마이그레이션한 후 객체를 검증하는 step-by-step 방법을 설명합니다.

이 패턴은 데이터베이스 객체 유효성 검사에 대한 사용 시나리오와 단계를 간략하게 설명하며, 자세한 내용은 [AWS 데이터베이스 블로그](#)에서 [AWS SCT 및 AWS DMS를 사용하여 마이그레이션한 다음 데이터베이스 객체 유효성 검사하기](#)를 참조하세요.

사전 조건 및 제한 사항

사전 조건

- 활성 상태의 AWS 계정
- Aurora PostgreSQL 호환 데이터베이스로 마이그레이션된 온프레미스 오라클 데이터베이스
- Aurora PostgreSQL 호환 데이터베이스에 대해 [AmazonRDS DataFullAccess](#) 정책이 적용된 로그인 자격 증명입니다.
- 이 패턴은 Amazon Relational Database Service(Amazon RDS) 콘솔에서 사용할 수 있는 [Aurora 서버리스 DB 클러스터용 쿼리 에디터](#)를 사용합니다. 하지만 이 패턴은 다른 쿼리 편집기에서도 사용할 수 있습니다.

제한 사항

- Oracle SYNONYM 객체는 PostgreSQL에서 사용할 수 없지만 보기 또는 SET search_path 쿼리를 통해 부분적으로 유효성을 검사할 수 있습니다.
- Amazon RDS 쿼리 편집기는 [특정 AWS 리전과 특정 MySQL 및 PostgreSQL 버전](#)에서만 사용할 수 있습니다.

아키텍처

도구

도구

- [Amazon Aurora PostgreSQL 호환 버전](#) - Aurora PostgreSQL 호환 버전은 완전 관리형, PostgreSQL 호환, ACID 호환 관계형 데이터베이스 엔진으로, 고급 상용 데이터베이스의 속도와 신뢰성을 오픈 소스 데이터베이스의 단순성과 비용 효율성에 결합한 제품입니다.
- [Amazon RDS](#) - Amazon Relational Database Service(Amazon RDS)를 사용하면 AWS 클라우드에서 관계형 데이터베이스를 더 쉽게 설정, 운영, 확장할 수 있습니다. 업계 표준 관계형 데이터베이스를 위한 비용 효율적이고 크기 조정이 가능한 용량을 제공하며 일반적인 데이터베이스 관리 작업을 관리합니다.
- [Aurora Severless 용 쿼리 편집기](#) — 쿼리 편집기는 Amazon RDS 콘솔에서 SQL 쿼리를 실행하는 데 도움이 됩니다. 데이터 처리와 데이터 정의 문을 포함한 모든 유효한 SQL 문을 Aurora 서버리스 DB 클러스터에서 실행할 수 있습니다.

객체의 유효성을 검사하려면 '첨부 파일' 섹션의 '객체 유효성 검사 스크립트' 파일에 있는 전체 스크립트를 사용하세요. 다음 표를 참조하세요.

오라클 오브젝트	사용할 스크립트
패키지	쿼리 1
표	쿼리 3
보기	쿼리 5
시퀀스	쿼리 7

트리거	쿼리 9
프라이머리 키	쿼리 11
인덱스	쿼리 13
제약 조건 확인	쿼리 15
외래 키	쿼리 17

PostgreSQL 객체	사용할 스크립트
패키지	쿼리 2
표	쿼리 4
보기	쿼리 6
시퀀스	쿼리 8
트리거	쿼리 10
프라이머리 키	쿼리 12
인덱스	쿼리 14
제약 조건 확인	쿼리 16
외래 키	쿼리 18

에픽

소스 Oracle 데이터베이스의 개체 유효성 검사

작업	설명	필요한 기술
소스 오라클 데이터베이스에서 '패키지' 유효성 검사 쿼리를 실행합니다.	'첨부 파일' 섹션에서 '객체 유효성 검사 스크립트' 파일을 다운로드하여 엽니다. 클라이언	개발자, DBA

작업	설명	필요한 기술
	트 프로그램을 통해 소스 오라클 데이터베이스에 연결합니다. "'객체 유효성 검사 스크립트' 파일에서 '쿼리 1' 유효성 검사 스크립트를 실행합니다. 중요: 쿼리에 'your_schema' 대신 오라클 사용자 이름을 입력합니다. 쿼리 결과를 반드시 기록해 두세요.	
'테이블' 유효성 검사 쿼리를 실행합니다.	"'객체 유효성 검사 스크립트' 파일에서 '쿼리 3' 스크립트를 실행합니다. 쿼리 결과를 반드시 기록해 두세요.	개발자, DBA
'보기' 유효성 검사 쿼리를 실행합니다.	"'객체 유효성 검사 스크립트' 파일에서 '쿼리 5' 스크립트를 실행합니다. 쿼리 결과를 반드시 기록해 두세요.	개발자, DBA
'시퀀스' 개수 유효성 검사를 실행합니다.	"'객체 유효성 검사 스크립트' 파일에서 '쿼리 7' 스크립트를 실행합니다. 쿼리 결과를 반드시 기록해 두세요.	개발자, DBA
'트리거' 유효성 검사 쿼리를 실행합니다.	"'객체 유효성 검사 스크립트' 파일에서 '쿼리 9' 스크립트를 실행합니다. 쿼리 결과를 반드시 기록해 두세요.	개발자, DBA
'프라이머리 키' 유효성 검사 쿼리를 실행합니다.	"'객체 유효성 검사 스크립트' 파일에서 '쿼리 11' 스크립트를 실행합니다. 쿼리 결과를 반드시 기록해 두세요.	개발자, DBA

작업	설명	필요한 기술
'인덱스' 유효성 검사 쿼리를 실행합니다.	"'객체 유효성 검사 스크립트' 파일에서 '쿼리 13' 유효성 검사 스크립트를 실행합니다. 쿼리 결과를 반드시 기록해 두세요.	개발자, DBA
'제약 조건 확인' 유효성 검사 쿼리를 실행합니다.	"'객체 유효성 검사 스크립트' 파일에서 '쿼리 15' 스크립트를 실행합니다. 쿼리 결과를 반드시 기록해 두세요.	개발자, DBA
'외래 키' 유효성 검사 쿼리를 실행합니다.	"'객체 유효성 검사 스크립트' 파일에서 '쿼리 17' 유효성 검사 스크립트를 실행합니다. 쿼리 결과를 반드시 기록해 두세요.	개발자, DBA

대상 Aurora PostgreSQL 호환 데이터베이스의 객체 유효성 검사

작업	설명	필요한 기술
쿼리 편집기를 사용하여 대상 Aurora PostgreSQL 호환 데이터베이스에 연결합니다.	AWS Management Console에 로그인하고 Amazon RDS 콘솔을 엽니다. 오른쪽 위 모서리에서 Aurora PostgreSQL 호환 데이터베이스를 생성한 AWS 리전을 선택합니다. 탐색 창에서 '데이터베이스'를 선택하고 대상 Aurora PostgreSQL 호환 데이터베이스를 선택합니다. '작업'에서 '쿼리'를 선택합니다. 중요: 이전에 데이터베이스에 연결한 적이 없으면 '데이터베이스에 연결' 페이지가 열립니다. 그런 다음 사용자 이름과 비밀	개발자, DBA

작업	설명	필요한 기술
	번호 같은 데이터베이스 정보를 입력해야 합니다.	
'패키지' 유효성 검사 쿼리를 실행합니다.	'첨부 파일' 섹션의 '객체 유효성 검사 스크립트' 파일에서 '쿼리 2' 스크립트를 실행합니다. 쿼리 결과를 반드시 기록해 두세요.	개발자, DBA
'테이블' 유효성 검사 쿼리를 실행합니다.	Aurora PostgreSQL 호환 데이터베이스의 쿼리 편집기로 돌아가서 '객체 유효성 검사 스크립트' 파일에서 '쿼리 4' 스크립트를 실행합니다. 쿼리 결과를 반드시 기록해 두세요.	개발자, DBA
'보기' 유효성 검사 쿼리를 실행합니다.	Aurora PostgreSQL 호환 데이터베이스의 쿼리 편집기로 돌아가서 '객체 유효성 검사 스크립트' 파일에서 '쿼리 6' 스크립트를 실행합니다. 쿼리 결과를 반드시 기록해 두세요.	개발자, DBA
'시퀀스' 개수 유효성 검사를 실행합니다.	Aurora PostgreSQL 호환 데이터베이스의 쿼리 편집기로 돌아가서 '객체 유효성 검사 스크립트' 파일에서 '쿼리 8' 스크립트를 실행합니다. 쿼리 결과를 반드시 기록해 두세요.	개발자, DBA
'트리거' 유효성 검사 쿼리를 실행합니다.	Aurora PostgreSQL 호환 데이터베이스의 쿼리 편집기로 돌아가서 '객체 유효성 검사 스크립트' 파일에서 '쿼리 10' 스크립트를 실행합니다. 쿼리 결과를 반드시 기록해 두세요.	개발자, DBA

작업	설명	필요한 기술
'프라이머리 키' 유효성 검사 쿼리를 실행합니다.	Aurora PostgreSQL 호환 데이터베이스의 쿼리 편집기로 돌아가서 '객체 유효성 검사 스크립트' 파일에서 '쿼리 12' 스크립트를 실행합니다. 쿼리 결과를 반드시 기록해 두세요.	개발자, DBA
'인덱스' 유효성 검사 쿼리를 실행합니다.	Aurora PostgreSQL 호환 데이터베이스의 쿼리 편집기로 돌아가서 '객체 유효성 검사 스크립트' 파일에서 '쿼리 14' 스크립트를 실행합니다. 쿼리 결과를 반드시 기록해 두세요.	개발자, DBA
'계약 조건 확인' 유효성 검사 쿼리를 실행합니다.	"'객체 유효성 검사 스크립트' 파일에서 '쿼리 16' 스크립트를 실행합니다. 쿼리 결과를 반드시 기록해 두세요.	개발자, DBA
'외래 키' 유효성 검사 쿼리를 실행합니다.	"'객체 유효성 검사 스크립트' 파일에서 '쿼리 18' 유효성 검사 스크립트를 실행합니다. 쿼리 결과를 반드시 기록해 두세요.	개발자, DBA

소스 및 대상 데이터베이스 유효성 검사 레코드 비교

작업	설명	필요한 기술
두 쿼리 결과를 비교하고 검증하세요.	오라클과 Aurora PostgreSQL 호환 데이터베이스의 쿼리 결과를 비교하여 모든 객체의 유효성을 검사하세요. 모두 일치하면 모든 객체의 유효성 검사가 제대로 이루어진 것입니다.	개발자, DBA

관련 리소스

- [AWS SCT 및 AWS DMS를 사용하여 마이그레이션 후 데이터베이스 객체 유효성 검사](#)
- [Amazon Aurora 특성: PostgreSQL 호환 에디션](#)

첨부

이 문서와 관련된 추가 콘텐츠에 액세스하려면 [attachment.zip](#) 파일의 압축을 풉니다.

리호스팅

주제

- [Microsoft 워크로드의 검색 및 AWS로의 마이그레이션을 가속화합니다.](#)
- [Windows 기반 AWS Managed Services의 사전 워크로드 수집 활동 자동화](#)
- [AWS로 리호스팅 마이그레이션하는 동안 방화벽 요청에 대한 승인 프로세스를 생성](#)
- [EC2 Windows 인스턴스를 수집하여 AWS Managed Services 계정으로 마이그레이션합니다.](#)
- [로그 전송을 사용하여 운영 중단 시간을 줄임으로써 Db2 for LUW를 Amazon EC2로 마이그레이션합니다.](#)
- [고가용성 재해 복구 기능을 갖춘 Db2 for LUW를 Amazon EC2로 마이그레이션하세요.](#)
- [PowerCLI를 사용하여 HCX 자동화를 통해 VMware VM 마이그레이션](#)
- [F5 BIG-IP 워크로드를 AWS 클라우드의 F5 BIG-IP VE로 마이그레이션](#)
- [이진법을 사용하여 온프레미스 Go 웹 애플리케이션을 AWS Elastic Beanstalk로 마이그레이션](#)
- [SFTP용 AWS Transfer를 사용하여 온프레미스 SFTP 서버를 AWS로 마이그레이션](#)
- [AWS 애플리케이션 마이그레이션 서비스를 사용하여 온프레미스 VM을 Amazon EC2로 마이그레이션하기](#)
- [AWS SFTP를 사용하여 온프레미스에서 Amazon S3로 소규모 데이터 세트를 마이그레이션합니다.](#)
- [오라클에서 AWS Elastic GlassFish Beanstalk로 마이그레이션](#)
- [온프레미스 Oracle 데이터베이스를 Amazon EC2의 Oracle로 마이그레이션](#)
- [Oracle Data Pump를 사용하여 온프레미스 Oracle 데이터베이스를 Amazon EC2 로 마이그레이션](#)
- [온프레미스 SAP ASE 데이터베이스를 Amazon EC2로 마이그레이션](#)
- [온프레미스 Microsoft SQL Server 데이터베이스를 Amazon EC2로 마이그레이션](#)
- [온프레미스 MySQL 데이터베이스를 Amazon EC2로 마이그레이션](#)
- [Application Migration Service를 사용하여 동종 SAP 마이그레이션 전환 시간 단축](#)
- [AWS 클라우드의 온프레미스 워크로드 리호스팅: 마이그레이션 체크리스트](#)
- [Amazon FSx를 사용하여 SQL Server Always On FCI용 다중 AZ 인프라 설정](#)
- [BMC Discovery 쿼리를 사용하여 마이그레이션 계획을 위한 마이그레이션 데이터 추출](#)

Microsoft 워크로드의 검색 및 AWS로의 마이그레이션을 가속화합니다.

작성자: 알리 알랜드

환경: 프로덕션	출처: 온프레미스 또는 기타 클라우드 서비스 공급자를 실행하는 Microsoft 워크로드	대상: 아마존 EC2 윈도우
R 유형: 리호스팅	워크로드: Microsoft	기술: 마이그레이션
AWS 서비스: Amazon EC2		

요약

이 패턴은 [마이그레이션 검사기 툴킷 PowerShell 모듈](#)을 사용하여 Microsoft 워크로드를 검색하고 AWS로 마이그레이션하는 방법을 보여줍니다. 이 모듈은 Microsoft 워크로드와 관련된 일반적인 작업에 대해 여러 검사와 검증을 수행하여 작동합니다. 예를 들어 모듈은 여러 디스크가 연결되어 있을 수 있는 인스턴스 또는 많은 IP 주소를 사용하는 인스턴스를 확인합니다. 모듈이 수행할 수 있는 검사의 전체 목록은 모듈 GitHub 페이지의 [검사](#) 섹션을 참조하십시오.

Migration Validator Toolkit PowerShell 모듈을 사용하면 조직에서 Microsoft 워크로드에서 실행 중인 응용 프로그램 및 서비스를 검색하는 데 드는 시간과 노력을 줄일 수 있습니다. 또한 이 모듈을 사용하면 워크로드의 구성을 식별하여 해당 구성이 AWS에서 지원되는지 확인할 수 있습니다. 이 모듈에서는 마이그레이션 전, 도중, 후에 잘못된 구성을 방지할 수 있도록 다음 단계와 완화 조치에 대한 권장 사항도 제공합니다.

사전 조건 및 제한 사항

사전 조건

- 로컬 관리자 계정
- PowerShell 4.0

제한 사항

- 마이크로소프트 윈도우 서버 2012 R2 이상에서만 작동합니다.

도구

도구

- PowerShell 4.0

코드 리포지토리

[이 패턴에 대한 마이그레이션 검증기 툴킷 PowerShell 모듈은 -microsoft-workload 저장소에서 사용할 수 있습니다. GitHub migration-validator-toolkit-for](#)

에픽

마이그레이션 검사기 툴킷 모듈을 단일 대상에서 실행합니다. PowerShell

작업	설명	필요한 기술
모듈을 다운로드하고, 추출하고, 가져오고, 호출합니다.	<p>다음 방법 중 하나를 선택하여 모듈을 다운로드하고 배포하십시오.</p> <ul style="list-style-type: none"> • PowerShell 스크립트를 실행합니다. • .zip 파일을 다운로드하고 압축을 풉니다. • GitHub 리포지토리 복제 <p>PowerShell 스크립트 실행</p> <p>에서 PowerShell 다음 예제 코드를 실행합니다.</p> <pre>#MigrationValidatorToolkit \$uri = 'https://github.com/aws-samples/migration-validator-toolkit-for-microsoft-workloads/'</pre>	시스템 관리자

작업	설명	필요한 기술
	<pre> archive/refs/heads/ main.zip' \$destination = (Get- Location).Path if ((Test-Path -Path "\$destination\Migr ationValidatorTool kit.zip" -PathType Leaf) -or (Test-Path - Path "\$destination\Migr ationValidatorTool kit")) { write-host "File \$destination\Migra tionValidatorToolk it.zip or folder \$destination\Migra tionValidatorToolkit found, exiting" }else { Write-host "Enable TLS 1.2 for this PowerShell session only." [Net.ServicePointM anager]::SecurityP rotocol = [Net.Secu rityProtocolType]: :Tls12 \$webClient = New-Object System.Ne t.WebClient Write-host "Downloading Migration ValidatorToolkit.zip" \$webClient.Downloa dFile(\$uri, "\$destina tion\MigrationVali datorToolkit.zip") Write-host "MigrationValidato </pre>	

작업	설명	필요한 기술
	<pre> rToolkit.zip download successfully" Add-Type -Assembly "system.io.compres sion.filesystem" [System.IO.Compres sion.ZipFile]::Ext ractToDirectory("\$ destination\Migrat ionValidatorToolki t.zip", "\$destinati on\MigrationValida torToolkit") Write-host "Extracting Migration ValidatorToolkit.zip complete successfully" Import-Module "\$destination\Migr ationValidatorToolkit \migration-validator- toolkit-for-microsoft -workloads-main\Mi grationValidatorTo olkit.psm1"; Invoke- MigrationValidatorTo olkit } </pre> <p>코드는 .zip 파일에서 모듈을 다운로드합니다. 그러면 코드가 모듈을 추출하고, 가져오고, 호출합니다.</p> <p>.zip 파일을 다운로드하고 압축을 풉니다.</p> <ol style="list-style-type: none"> 1. .zip 파일 다운로드 (다운로드). 2. .zip 파일의 압축을 풉니다. 	

작업	설명	필요한 기술
	<p>3. 이 가이드의 모듈 수동 호출 스토리에 있는 단계를 따르세요.</p> <p>리포지토리를 GitHub 복제하세요.</p> <p>1. GitHubmigration-validator-toolkit-for-microsoft-workloads 리포지토리를 복제하려면 터미널 창에서 다음 Git 명령을 실행합니다.</p> <pre data-bbox="630 808 1029 1087">git clone https://github.com/aws-samples/migration-validator-toolkit-for-microsoft-workloads.git</pre> <p>2. 이 가이드의 모듈 수동 호출 스토리의 단계를 따르세요.</p>	

작업	설명	필요한 기술
<p>모듈을 수동으로 호출하십시오.</p>	<ol style="list-style-type: none"> 1. 다운로드한 모듈이 저장된 디렉토리로 이동합니다. 2. 원하는 출력을 생성하려면 다음 명령 중 하나를 관리자 로 실행하십시오 PowerShell. <p>형식-테이블 형식:</p> <pre>Import-Module .\MigrationValidatorToolkit.psm1;Invoke-MigrationValidatorToolkit</pre> <p>형식 목록 형식:</p> <pre>Import-Module .\MigrationValidatorToolkit.psm1;Invoke-MigrationValidatorToolkit -List</pre> <p>출력 포맷: GridView</p> <pre>Import-Module .\MigrationValidatorToolkit.psm1;Invoke-MigrationValidatorToolkit -GridView</pre> <p>ConvertTo-Csv 포맷:</p> <pre>Import-Module .\MigrationValidatorToolkit.psm1;Invoke-Migra</pre>	<p>시스템 관리자</p>

작업	설명	필요한 기술
	<pre>tionValidatorToolkit -csv</pre>	

여러 대상에서 마이그레이션 검사기 툴킷 PowerShell 모듈을 실행합니다.

작업	설명	필요한 기술
.zip 파일을 다운로드하거나 리포지토리를 복제합니다. GitHub	<p>다음 옵션 중 하나를 선택합니다:</p> <ul style="list-style-type: none"> zip 파일을 다운로드합니다. (다운로드). GitHub migration-validator-toolkit-for-microsoft-workloads 리포지토리를 복제하려면 터미널 창에서 다음 Git 명령을 실행합니다. <pre>git clone https://github.com/aws-samples/migration-validator-toolkit-for-microsoft-workloads.git</pre>	시스템 관리자
server.csv 목록을 업데이트하세요.	<p>.zip 파일을 다운로드한 경우 다음 단계를 따르십시오.</p> <ol style="list-style-type: none"> .zip 파일의 압축을 풉니다. MigrationValidatorToolkit\Inputs\ 디렉터리로 이동합니다. 	시스템 관리자

작업	설명	필요한 기술
<p>모듈을 호출합니다.</p>	<p>3. 대상 컴퓨터의 호스트 <code>serverlist.csv</code> 이름으로 업데이트하십시오.</p> <p>대상 컴퓨터에 대한 관리자 액세스 권한이 있는 도메인 사용자를 사용하는 도메인 내의 모든 컴퓨터를 사용할 수 있습니다.</p> <ol style="list-style-type: none"> 1. 소스 코드를 .zip 파일로 다운로드하고 파일을 추출합니다. 2. 의 PowerShell 관리자로 다음 명령을 실행합니다. <pre>Import-Module .\MigrationValidatorToolkit.psm1;Invoke-DomainComputers</pre> <p>출력.csv 파일은 접두사 이름과 MigrationValidatorToolkit\Outputs\folder 함께 저장됩니다. DomainComputers_MigrationAutomations_YYYY-MM-DDTHH-MM-SS</p>	<p>시스템 관리자</p>

문제 해결

문제	Solution
MigrationValidatorToolkit 실행, 명령 및 오류에 대한 정보를 실행 중인 호스트의 로그 파일에 기록합니다.	<p>다음 위치에서 수동으로 로그 파일을 볼 수 있습니다.</p> <ol style="list-style-type: none"> 1. MigrationValidatorToolkit\logs \ 디렉터리로 이동합니다. 2. 로그 파일을 찾으세요. 로그 파일 이름의 형식은 다음과 같습니다. ComputerName_MigrationValidatorToolkit_YYYY-MM-SSTHH-MM-SS.log

관련 리소스

- [Microsoft 워크로드를 AWS로 마이그레이션하기 위한 옵션, 도구 및 모범 사례 \(AWS 처방적 지침\)](#)
- [마이크로소프트 마이그레이션 패턴 \(AWS Creptive Guidance\)](#)
- AWS 기반 [무료 클라우드 마이그레이션 서비스 \(AWS 설명서\)](#)
- [사전 정의된 출시 후 조치 \(애플리케이션 마케팅 문서\)](#)

추가 정보

자주 묻는 질문

마이그레이션 검증기 툴킷 모듈을 어디에서 실행할 수 있습니까? PowerShell

이 모듈은 마이크로소프트 윈도우 서버 2012 R2 이상에서 실행할 수 있습니다.

이 모듈은 언제 실행하나요?

마이그레이션 과정의 [평가 단계에서](#) 모듈을 실행하는 것이 좋습니다.

모듈이 기존 서버를 수정하나요?

아니요. 이 모듈의 모든 작업은 읽기 전용입니다.

모듈을 실행하는 데 시간이 얼마나 걸립니까?

일반적으로 모듈을 실행하는 데 1~5분이 소요되지만 서버의 리소스 할당에 따라 다릅니다.

모듈을 실행하려면 어떤 권한이 필요한가요?

로컬 관리자 계정에서 모듈을 실행해야 합니다.

물리적 서버에서 모듈을 실행할 수 있나요?

예, 운영 체제가 마이크로소프트 윈도우 서버 2012 R2 이상이면 가능합니다.

여러 서버에서 모듈을 대규모로 실행하려면 어떻게 해야 합니까?

도메인에 가입된 여러 대의 컴퓨터에서 모듈을 대규모로 실행하려면 이 가이드의 다중 대상에서 마이그레이션 유효성 검사기 툴킷 PowerShell 모듈 실행 서사시의 단계를 따르세요. 도메인에 가입되지 않은 컴퓨터의 경우 이 가이드의 단일 대상에서 마이그레이션 유효성 검사기 툴킷 모듈 실행에 나와 있는 단계에 따라 원격 호출을 사용하거나 모듈을 로컬로 실행하십시오. PowerShell

Windows 기반 AWS Managed Services의 사전 워크로드 수집 활동 자동화

작성자: Jacob Zhang(AWS), Calvin Yeh(AWS), Dwayne Bordelon(AWS)

코드 리포지토리: GitHub	환경: 프로덕션	소스: Windows 서버
대상: AWS 관리형 서비스	R 유형: 리호스팅	기술: 마이그레이션
AWS 서비스: AWS CloudFormation, AWS Managed Services, AWS Systems Manager, 아마존 S3		

요약

Amazon Web Services(AWS) 클라우드에서 AWS Managed Services(AMS)는 AMS 워크로드 인제스트(WIGS)를 사용하여 기존 워크로드를 AMS 관리형 VPC로 이동합니다. 이 패턴은 .NET PowerShell 및 Windows를 업그레이드하고 AMS에서 유지 관리하는 Windows WIGS 사전 통합 검증을 실행하는 등 일반적인 사전 워크로드 수집 활동을 자동화하는 솔루션을 설명합니다. 또한 이 패턴은 실행 결과를 위한 통합 사용자 인터페이스를 제공합니다. 사전 통합 활동을 수행하는 AWS Systems Manager 명령 문서를 AWS 템플릿으로 패키징합니다. CloudFormation 템플릿은 Systems Manager 자체에 액세스하거나 AMS의 자동화와 충돌하지 않고도 반복적으로 배포할 수 있습니다.

비즈니스 배경

AMS로 마이그레이션하려면 AMS 구성 요소가 포함된 AMS에서 관리하는 Amazon Machine Image(AMI)를 사용하여 Amazon Elastic Compute Cloud(Amazon EC2) 인스턴스를 새로 프로비저닝해야 합니다. 기존 데이터 센터에서 실행되는 모든 워크로드 또는 애플리케이션을 이러한 AMS AMI에서 시작된 새 EC2 인스턴스로 재배포해야 합니다. 프로세스 중 잠재적으로 엄청난 양의 수동 작업을 피하기 위해 AMS 팀은 AMS 워크로드 인제스트(WIGS) 워크플로를 구축하여 사용자 지정 이미지를 AMS에 온보딩했습니다.

WIGS 프로세스를 실행하기 전에 Windows 인스턴스가 몇 가지 사전 조건을 충족해야 합니다. Windows PowerShell 스크립트는 일반적으로 필요한 준비 (WIGS 준비) 를 수행하고 인스턴스가 WIG를 사용할 준비가 되었는지 확인 (WIGS 사전 통합 검증) 하는 데 사용됩니다. 준비 및 검증 프로세스에서는 엔지니어가 각 서버에서 15~30분 동안 수동으로 로그인하여 스크립트를 하나씩 실행해야 합니다.

비즈니스 드라이버

일반적으로 Systems Manager를 사용하면 Windows PowerShell 스크립트 실행과 같은 운영 작업을 자동화할 수 있습니다. 그러나 AMS의 자동화와 사용자 자동화 간의 잦은 충돌 및 위험 증가로 인해 AMS는 일반적으로 사용자에게 Systems Manager에 대한 액세스 권한을 부여하지 않습니다.

AWS 애플리케이션 마이그레이션 서비스 (AWS MGN) 를 사용하는 대량 마이그레이션의 경우 테스트 또는 컷오버 인스턴스가 시작될 때 C:\Program Files (x86)\AWS Replication Agent\post_launch folder 일반적으로 Windows PowerShell 스크립트가 자동으로 실행됩니다. 그러나 이러한 스크립트는 인스턴스 시작 중에 즉시 실행되는 경우 AMS의 자동화와 충돌하는 경우가 많습니다. 따라서 장애 해결에 필요한 실행 결과를 제공하지 않으면 시작이 실패할 수 있습니다.

이 패턴은 이러한 문제를 해결하고 작동하는 자동화된 솔루션을 제공합니다.

사전 조건 및 제한 사항

사전 조건

- AMS 온보딩이 완료된 활성 AWS 계정.
- AWS 계정의 Amazon Simple Storage Service(S3) 버킷. 계정에 제어할 수 있는 S3 버킷이 없는 경우 변경 요청(RFC)을 사용하여 버킷을 생성하십시오.
- Prewigs_cfn.json 템플릿은 리포지토리에서 다운로드했습니다. [ams-auto-prewigs-windows](#)
- 이 패턴을 적용하는 서버는 다음 요구 사항을 충족해야 합니다.
 - Windows Server 2012 이상.
 - 샌드박스 VPC 마이그레이션 서브넷에서 시작하거나 시작할 준비가 되어 있어야 합니다.
 - AWS Systems Manager Agent(SSM Agent)가 설치되어 있어야 합니다.
 - AWS Identity and Access Management(IAM) 인스턴스 프로파일을 첨부하십시오. 인스턴스 프로파일에는 동일한 AWS 계정의 S3 버킷에서 파일을 다운로드할 수 있는 권한이 있어야 합니다. 위에서 언급한 요구 사항을 충족하는 인스턴스 프로파일은 일반적으로 마이그레이션의 초기 설정 중에 이미 설정됩니다.
 - AWS Systems Manager Fleet Manager에서 확인할 수 있습니다.

제한 사항

- WIGS 이전 활동은 환경 및 비즈니스 요구 사항에 따라 달라집니다. 필요에 따라 이 패턴을 약간 수정해야 할 수 있습니다.

제품 버전

- 이 패턴은 Windows 서버 2012, 2012 R2, 2016 및 2019에서 테스트되었습니다. 이론적으로 최신 Windows 버전에서 작동합니다. 이전 Windows 버전에서는 작동하지 않습니다.

아키텍처

다이어그램은 다음 아키텍처를 보여줍니다:

1. 준비되지 않은 서버가 포함된 마이그레이션 서브넷이 있는 샌드박스 VPC입니다.
2. 템플릿에서 사용하는 스크립트를 저장하는 S3 버킷입니다. CloudFormation
3. CloudFormation 템플릿은 Systems Manager 명령 문서를 배포합니다. 이 프로세스는 단계가 완료 될 때까지 반복됩니다.
4. 인스턴스가 준비되고 WIGS에 대한 RFC가 작성됩니다.
5. AMS 관리형 VPC에서 AMS 관리 서브넷에는 워크로드 통합 이후의 서버가 포함됩니다.

작동 방식

- 이 패턴은 코드형 인프라 (IaC) 를 반복 배포할 수 있는 AWS CloudFormation 템플릿에 패키징됩니다. 이 자동화가 필요한 각 AWS 계정에 대해 이 템플릿을 한 번만 배포하면 됩니다.
- 자동화는 이 패턴이 배포된 AWS 계정의 태그 키 AutoPreWIG가 있는 모든 EC2 인스턴스에 적용됩니다. 태그 키 AutoPreWIG가 있는 Amazon EC2 Windows 인스턴스를 처음 시작하면 자동화가 다음 작업을 수행합니다.
 1. 윈도우를 버전 5.1로, PowerShell .NET을 버전 4.5.2로 업그레이드합니다. 기존 Windows PowerShell 및 .NET 버전에 따라 인스턴스가 여러 번 재부팅될 수 있습니다. 재부팅할 때마다 업그레이드가 완료될 때까지 계속 진행됩니다. 이 단계에서는 [Windows PowerShell 스크립트에서](#) 수정된 CloudFormation 템플릿에 포함된 코드와 서버 재부팅에 대한 특정 Systems Manager 지침을 사용합니다.
 2. Amazon S3에서 다운로드하고 WIGS용 Amazon EC2 Windows 인스턴스를 준비하기 위해 사용자 지정한 Windows PowerShell 스크립트를 실행합니다. 자세한 정보는 에픽 섹션을 참조하십시오.
 3. AWS에서 Windows WIGS 사전 통합 검증 모듈을 설치합니다. PowerShell
 4. Windows WIGS 사전 통합 검증을 실행하고 Systems Manager 상태 관리자에서 결과를 볼 수 있도록 합니다.

도구

- [AWS CloudFormation](#) — CloudFormation AWS는 AWS 리소스를 모델링하고 설정하는 데 도움이 되는 서비스입니다. 원하는 모든 AWS 리소스와 해당 종속성을 설명하는 코드를 사용하여 해당 리소스를 스택으로 시작하고 구성할 수 있습니다., 이 패턴은 CloudFormation 템플릿을 사용하여 이 패턴의 리소스 배포를 자동화합니다.
- [AWS Managed Services](#) — AWS Managed Services(AMS)는 AWS 인프라를 지속적으로 관리하는 엔터프라이즈 서비스입니다. AMS 환경의 인프라를 변경하려면 RFC를 통해 변경해야 합니다.
- [AWS Systems Manager](#) - AWS Systems Manager(이전에 SSM으로 알려짐)는 AWS에서 인프라를 보고 제어하기 위해 사용할 수 있는 AWS 서비스입니다. Systems Manager 콘솔을 사용하여 여러 AWS 서비스의 운영 데이터를 보고 AWS 리소스에서 운영 작업을 자동화할 수 있습니다. 이 패턴은 Systems Manager를 사용하여 WIGS 이전 활동의 실행 결과를 실행하고 확인합니다.
- [Amazon S3](#) - Amazon Simple Storage Service(S3)는 업계 최고의 확장성, 데이터 가용성, 보안 및 성능을 제공하는 객체 스토리지 서비스입니다. 이 패턴은 Amazon S3를 사용하여 CloudFormation 템플릿과 다운로드된 Windows PowerShell 스크립트를 저장합니다.

에픽

사용자 지정 Windows PowerShell 스크립트를 생성하여 추가 작업을 자동화하십시오.

작업	설명	필요한 기술
비즈니스 요구 사항에 따라 서버를 필요에 따라 변경합니다.	수집 전에 변경 내용을 서버에 자동으로 적용하려면 이라는 Windows PowerShell 스크립트를 만드세요. ingestion-prep.ps1 중요: 스크립트에는 서버를 재부팅하기 위한 지침이 포함되어서는 안 되며 관리자 권한이 필요하지도 않아야 합니다.	PowerShell 스크립팅
AMS에서 지원하지 않는 소프트웨어를 제거합니다.	AMS를 사용하려면 WIGS를 실행하기 전에 바이러스 백신 애플리케이션 및 VMware Tools와 같은 특정 소프트웨어를 제	PowerShell 스크립팅

작업	설명	필요한 기술
	거해야 합니다. ingestion-prep.ps1 스크립트에 설치 제거를 포함시키십시오. 지원되지 않는 소프트웨어에 대한 자세한 내용은 AWS 설명서 를 참조하십시오.	

CloudFormation 템플릿과 선택적 Windows PowerShell 스크립트를 Amazon S3에 업로드합니다.

작업	설명	필요한 기술
S3에 폴더를 생성합니다.	이 패턴을 배포하는 동일한 AWS 계정의 S3 버킷에서 폴더를 생성합니다.	일반 AWS
스크립트를 업로드하십시오.	이전 에픽에서 만든 PreWIGs_CFN.json CloudFormation 템플릿과 ingestion-prep.ps1 Windows PowerShell 스크립트를 Amazon S3 폴더에 업로드합니다.	일반 AWS

스택을 배포하십시오. CloudFormation

작업	설명	필요한 기술
변경 유형을 선택합니다.	AMS 콘솔로 이동하여 RFC를 생성합니다. 스택 생성 출처 CloudFormation (CFN) 템플릿 변경 유형을 사용하십시오.	일반 AMS

작업	설명	필요한 기술
템플릿 경로에 대한 실행 매개변수를 설정합니다. CloudFormation	실행 구성 섹션에서 추가 구성을 확장합니다. CloudFormation 템플릿 S3 엔드포인트 상자에서 URL을 CloudFormation 템플릿에 붙여넣습니다.	일반 AMS
Amazon S3 폴더에 대한 경로를 지정합니다.	파라미터에서 ScriptSource이름으로 사용합니다. 값에 Windows PowerShell 스크립트가 들어 있는 S3 폴더의 경로를 입력합니다. https://xxx URL을 s3://xxx URI 대신 사용하고 끝에 /를 포함해야 합니다.	일반 AMS
스택을 배포합니다.	스택을 배포하려면 생성을 선택합니다.	일반 AMS
RFC를 AMS Ops로 에스컬레이션하십시오.	RFC는 Systems Manager를 사용하여 리소스를 배포하고 보안 검토를 필요로 하므로 AMS Ops 팀에서 수동으로 구현해야 합니다. RFC를 생성하자마자 시스템에서 자동으로 거부될 것입니다. RFC를 선택하고 RFC에 수동으로 실행하십시오라는 통신문을 추가합니다. RFC ID를 기록하고 서비스 요청과 함께 에스컬레이션하십시오.	일반 AMS

인스턴스에 자동화 적용

작업	설명	필요한 기술
인스턴스에 AutoPre WIGs 태그를 추가합니다.	<p>이 자동화를 적용하려는 모든 인스턴스의 ID를 기록하고 AMS에서 구현한 자동화를 인스턴스가 완료할 때까지 30분 이상 기다리십시오. 자동 RFC를 제출하여 AutoPreWIG를 키로 하고 모든 문자열 (예: 1) 을 값으로 사용하여 태그를 추가합니다.</p> <p>태그를 추가한 후 몇 분 후에 자동화가 적용됩니다.</p>	일반 AMS
자동화 결과를 확인하십시오.	Systems Manager 콘솔을 열고 State Manager를 선택합니다. AMS-PreWIG-Prep-and-Validation-Association라는 이름을 가진 연결 ID를 선택합니다. 실행 기록 탭에서 자동화 결과를 확인할 수 있습니다.	일반 AMS
오류를 모두 수정합니다.	자동화가 실패할 경우 실행 ID를 선택합니다. 각 EC2 인스턴스의 실행 결과를 볼 수 있습니다. 자동화의 각 단계에 대한 세부 정보를 보려면 출력을 선택합니다. 특정 단계가 실패할 경우 출력 및 오류 섹션의 정보를 사용하여 문제를 진단하십시오.	마이그레이션 엔지니어
AutoPreWiGs 태그를 제거합니다.	중요: 오류를 수정한 후 오류가 있는 경우 자동 RFC를 제출하여 AutoPreWIGs 태그를 제거	일반 AMS

작업	설명	필요한 기술
	하세요. 태그를 제거하지 않으면 WIGS는 실패합니다.	

준비된 인스턴스 수집

작업	설명	필요한 기술
WIGS에 대한 RFC를 제출하십시오.	이제 인스턴스를 워크로드 통합에 사용할 준비가 되었으므로 WIGS용 RFC를 제출하십시오.	일반 AMS

관련 리소스

- [AMS 워크로드 인제스트\(WIGS\)](#)
- [워크로드 마이그레이션: Windows 사전 통합 검증](#)
- [AWS Application Migration Service 빠른 시작 안내서](#)
- [AWS 시작하기 CloudFormation](#)
- [AWS Systems Manager 설정](#)

AWS로 리호스팅 마이그레이션하는 동안 방화벽 요청에 대한 승인 프로세스를 생성

작성자: Srikanth Rangavajhala(AWS)

R 유형: 리호스팅	환경: 프로덕션	기술: 마이그레이션
소스: 온프레미스	대상: AWS 클라우드	

요약

[AWS Application Migration Service](#) 또는 [AWS의 Cloud Migration Factory](#)를 사용하여 Amazon Web Services(AWS) 클라우드로 리호스팅 마이그레이션하려는 경우, 사전 요구 사항 중 하나는 TCP 포트 443과 1500을 열어 두어야 한다는 것입니다. 일반적으로 이러한 방화벽 포트를 열려면 정보 보안 (InfoSec) 팀의 승인이 필요합니다.

이 패턴은 AWS 클라우드로 재호스팅을 마이그레이션하는 동안 InfoSec 팀으로부터 방화벽 요청 승인을 받는 프로세스를 설명합니다. 이 프로세스를 사용하면 InfoSec 팀이 방화벽 요청을 거부하지 않도록 할 수 있으며, 이로 인해 비용과 시간이 많이 소요될 수 있습니다. 방화벽 요청 프로세스에는 고객 및 애플리케이션 팀과 협력하여 방화벽 포트를 여는 AWS 마이그레이션 컨설턴트와 리드 간에 두 가지 검토 InfoSec 및 승인 단계가 있습니다.

이 패턴은 조직의 AWS 컨설턴트 또는 마이그레이션 전문가와 함께 리호스팅 마이그레이션을 계획한다고 가정합니다. 조직에 방화벽 승인 프로세스 또는 방화벽 요청 포괄적 승인 양식이 없는 경우 이 패턴을 사용할 수 있습니다. 이에 대한 자세한 내용은 이 패턴의 제한 섹션을 참조하세요. Application Migration Service의 네트워크 요구 사항에 대한 자세한 내용은 Application Migration Service 설명서의 [네트워크 요구 사항](#)을 참조하세요.

사전 조건 및 제한 사항

사전 조건

- AWS 컨설턴트 또는 조직의 마이그레이션 전문가와 함께 계획된 리호스팅 마이그레이션
- 스택을 마이그레이션하는 데 필요한 포트 및 IP 정보
- 기존 및 미래 상태 아키텍처 다이어그램
- 온프레미스 및 대상 인프라, 포트, 트래픽 흐름에 대한 방화벽 정보 zone-to-zone

- 방화벽 요청 검토 체크리스트(첨부)
- 조직의 필요에 따라 구성된 방화벽 요청 문서
- 다음 역할을 포함한 방화벽 검토자 및 승인자를 위한 연락처 목록:
 - 방화벽 요청 제출자 – AWS 마이그레이션 전문가 또는 컨설턴트. 조직의 마이그레이션 전문가 또한 방화벽 요청을 제출할 수 있습니다.
 - 방화벽 요청 검토자 – 일반적으로 AWS의 단일 연락처(SPOC)입니다.
 - 방화벽 요청 승인자 - InfoSec 팀원.

제한 사항

- 이 패턴은 일반적인 방화벽 요청 승인 프로세스를 설명합니다. 요구 사항은 조직마다 다를 수 있습니다.
- 방화벽 요청 문서의 변경 내용을 추적해야 합니다.

다음 표에는 이 패턴의 사용 사례가 나와 있습니다.

조직에 기존 방화벽 승인 프로세스가 있나요?	조직에 기존 방화벽 요청 양식이 있나요?	권장 조치
예	예	AWS 컨설턴트 또는 마이그레이션 전문가와 협력하여 조직의 프로세스를 구현하세요.
아니요	예	이 패턴의 방화벽 승인 프로세스를 사용하세요. AWS 컨설턴트 또는 조직의 마이그레이션 전문가를 통해 방화벽 요청 포괄 승인 양식을 제출하세요.
아니요	아니요	이 패턴의 방화벽 승인 프로세스를 사용하세요. AWS 컨설턴트 또는 조직의 마이그레이션 전문가를 통해 방화벽 요청 포괄 승인 양식을 제출하세요.

아키텍처

다음 다이어그램은 방화벽 요청 승인 프로세스의 단계를 보여줍니다.

도구

[Palo Alto Networks](#)와 같은 스캐너 도구를 사용하거나 방화벽과 IP [SolarWinds](#) 주소를 분석하고 검증할 수 있습니다.

에픽

방화벽 요청 분석

작업	설명	필요한 기술
포트 및 IP 주소를 분석하세요.	방화벽 요청 제출자는 초기 분석을 완료하여 필요한 방화벽 포트 및 IP 주소를 파악합니다. 이 작업이 완료되면 InfoSec 팀에서 필요한 포트를 열고 IP 주소를 매핑하도록 요청합니다.	AWS 클라우드 엔지니어, 마이그레이션 전문가

방화벽 요청 검증

작업	설명	필요한 기술
방화벽 정보를 확인합니다.	AWS 클라우드 엔지니어가 InfoSec 팀과의 회의 일정을 잡습니다. 이 회의에서 엔지니어는 방화벽 요청 정보를 검토하고 검증합니다. 일반적으로 방화벽 요청 제출자는 방화벽 요청자와 동일한 사람입니다. 관찰되거나 권장되는 사항이 있는 경우 승인자	AWS 클라우드 엔지니어, 마이그레이션 전문가

작업	설명	필요한 기술
	가 제공한 피드백을 기반으로 이 검증 단계를 반복할 수 있습니다.	
방화벽 요청 문서 업데이트.	<p>InfoSec 팀에서 피드백을 공유한 후 방화벽 요청 문서를 편집, 저장 및 다시 업로드합니다. 이 문서는 반복할 때마다 업데이트됩니다.</p> <p>이 문서를 버전 관리 스토리지 폴더에 저장하는 것이 좋습니다. 따라서 모든 변경 사항이 추적되고 올바르게 적용되어야 합니다.</p>	AWS 클라우드 엔지니어, 마이그레이션 전문가

방화벽 요청 제출

작업	설명	필요한 기술
방화벽 요청을 제출하세요.	<p>방화벽 요청 승인자가 방화벽 포괄 승인 요청을 승인한 후, AWS 클라우드 엔지니어가 방화벽 요청을 제출합니다. 요청은 열려 있어야 하는 포트와 AWS 계정을 매핑하고 업데이트하는 데 필요한 IP 주소를 지정합니다.</p> <p>방화벽 요청이 제출된 후 제안을 하거나 피드백을 제공할 수 있습니다. 이 피드백 프로세스를 자동화하고 정의된 워크플로 메커니즘을 통해 편집 내용을 전송하는 것이 좋습니다.</p>	AWS 클라우드 엔지니어, 마이그레이션 전문가

첨부

이 문서와 관련된 추가 콘텐츠에 액세스하려면 [attachment.zip](#) 파일의 압축을 풉니다.

EC2 Windows 인스턴스를 수집하여 AWS Managed Services 계정으로 마이그레이션합니다.

작성자: Anil Kunapareddy(AWS), Venkatramana Chinthra(AWS)

환경: 프로덕션	소스: AWS 클라우드의 VPC	대상: AWS Managed Services에서 관리하는 VPC
R 유형: 리호스팅	워크로드: Microsoft	기술: 마이그레이션, 운영, 보안, 자격 증명, 규정 준수, 클라우드 네이티브
AWS 서비스: AWS 관리형 서비스		

요약

이 패턴은 아마존 Elastic Compute Cloud (Amazon EC2) Windows 인스턴스를 아마존 웹 서비스 (AWS) 관리 서비스 (AMS) 계정으로 마이그레이션하고 수집하는 step-by-step 프로세스를 설명합니다. AMS를 사용하면 인스턴스를 더 효율적이고 안전하게 관리할 수 있습니다. AMS는 운영 유연성을 제공하고 보안 및 규정 준수를 강화하며 용량을 최적화하고 비용을 절감하는 데 도움이 됩니다.

이 패턴은 AMS 계정의 스테이징 서브넷으로 마이그레이션한 EC2 Windows 인스턴스에서 시작됩니다. 이 작업을 수행하는 데 사용할 수 있는 다양한 마이그레이션 서비스 및 도구(예: AWS 애플리케이션 마이그레이션 서비스)가 있습니다.

AMS 관리형 환경을 변경하려면 특정 작업 또는 작업에 대한 변경 요청(RFC)을 생성하여 제출해야 합니다. AMS 워크로드 인제스트(WIGS) RFC를 사용하여 인스턴스를 AMS 계정으로 인제스트하고 사용자 지정 Amazon Machine Image(AMI)를 생성합니다. 그런 다음 다른 RFC를 제출하여 EC2 스택을 생성함으로써 AMS 관리형 EC2 인스턴스를 생성합니다. 자세한 내용은 AMS 설명서의 [AMS 워크로드 수집](#)을 참조하세요.

사전 조건 및 제한 사항

사전 조건

- AMS에서 관리하는 활성 AWS 계정
- 기존 랜딩 존

- AMS에서 관리하는 VPC에서 변경할 수 있는 권한
- AMS 계정의 스테이징 서브넷에 있는 Amazon EC2 Windows 인스턴스
- AMS WIGS를 사용하여 워크로드를 마이그레이션하기 위한 [일반 사전 조건](#) 완료
- AMS WIGS를 사용하여 워크로드를 마이그레이션하기 위한 [Windows 사전 조건](#) 완료

제한 사항

- 이 패턴은 Windows 서버를 운영하는 EC2 인스턴스를 위한 것입니다. 이 패턴은 Linux와 같은 다른 운영 체제를 실행하는 인스턴스에는 적용되지 않습니다.

아키텍처

소스 기술 스택

AMS 계정의 스테이징 서브넷에 있는 Amazon EC2 Windows 인스턴스

대상 기술 스택

AWS Managed Services(AMS)에서 관리하는 — Amazon EC2 Windows 인스턴스

대상 아키텍처

도구

서비스

- [Amazon Elastic Compute Cloud\(Amazon EC2\)](#)는 AWS 클라우드에서 규모를 조정할 수 있는 컴퓨팅 용량을 제공합니다. Amazon EC2를 사용하여 필요에 따라 많거나 적은 수의 가상 서버를 시작하고 스케일 아웃 또는 스케일 인할 수 있습니다.
- [AWS Identity and Access Management\(IAM\)](#)는 사용자에게 대한 인증 및 권한 부여를 제어함으로써 AWS 리소스에 대한 액세스를 안전하게 관리할 수 있습니다.
- [AWS Managed Services\(AMS\)](#)는 모니터링, 인시던트 관리, 보안 지침, 패치 지원, AWS 워크로드 백업 등 AWS 인프라에 대한 지속적인 관리를 제공하여 보다 효율적이고 안전하게 운영할 수 있도록 지원합니다.

기타 서비스

- [PowerShell](#) Windows, Linux 및 macOS에서 실행되는 Microsoft 자동화 및 구성 관리 프로그램입니다.

에픽

인스턴스에서 설정을 구성합니다.

작업	설명	필요한 기술
DNS 클라이언트 설정을 변경합니다.	<ol style="list-style-type: none"> 1. 소스 EC2 인스턴스에서 관리자 명령 프롬프트를 열고 <code>gpedit.msc</code> 를 입력한 다음 Enter를 누릅니다. 2. 로컬 그룹 정책 편집기에서 컴퓨터 구성, 관리 템플릿, 네트워크, DNS 클라이언트로 이동합니다. 3. 주 DNS 접미사의 경우 구성되지 않음을 선택합니다. 4. 기본 DNS 접미사 반환의 경우 구성되지 않음을 선택합니다. 	마이그레이션 엔지니어
Windows 업데이트 설정을 변경합니다.	<ol style="list-style-type: none"> 1. 로컬 그룹 정책 편집기에서 컴퓨터 구성, 관리 템플릿, Windows 구성 요소, Windows 업데이트로 이동합니다. 2. 인트라넷 Microsoft 업데이트 서비스 위치 지정에서 구성되지 않음을 선택합니다. 3. 자동 업데이트 구성에서 구성되지 않음을 선택합니다. 4. 자동 업데이트 감지 빈도의 경우 구성되지 않음을 선택합니다. 	마이그레이션 엔지니어

작업	설명	필요한 기술
	5. 로컬 그룹 정책 편집기를 닫습니다.	
방화벽을 사용합니다.	<ol style="list-style-type: none"> 1. 소스 EC2 인스턴스에서 관리자 명령 프롬프트를 열고 <code>services.msc</code> 를 입력한 다음 Enter를 누릅니다. 2. Windows 서비스에서 방화벽을 활성화합니다. 3. Windows 서비스를 닫습니다. 	마이그레이션 엔지니어

AMS WIGS용 인스턴스 준비

작업	설명	필요한 기술
인스턴스를 정리하고 준비합니다.	<ol style="list-style-type: none"> 1. Bastion Host와 로컬 보안 인증 정보를 사용하여 스테이징 서브넷에서 EC2 인스턴스의 (원격 데스크톱 프로토콜(RDP) 연결을 생성합니다. 2. AMS에 필요하지 않은 모든 레거시 소프트웨어, 바이러스 백신 소프트웨어 및 백업 솔루션을 제거합니다. 	마이그레이션 엔지니어
sppnp.dll 파일을 복구합니다.	<ol style="list-style-type: none"> 1. <code>C:\Windows\System32\sppnp.dll</code> 로 이동합니다. 2. <code>sppnp.dll</code> 을 <code>sppnp_old.dll</code> 로 바꿉니다. 	마이그레이션 엔지니어

작업	설명	필요한 기술
	<p>3. 관리자 자격 증명을 사용하여 PowerShell 다음 명령을 입력합니다.</p> <pre style="border: 1px solid #ccc; border-radius: 10px; padding: 10px; margin: 10px 0;">dism /online /cleanup-image /restorehealth sfc /scannow</pre> <p>4. EC2 Windows 인스턴스의 EC2 Windows 인스턴스의 재시작합니다.</p>	
<p>WIG 이전 검증 스크립트를 실행합니다.</p>	<ol style="list-style-type: none"> 1. AMS 설명서의 워크로드 마이그레이션: Windows 사전 통합 검증에서 Windows WIGS 사전 통합 검증 zip 파일(windows-prewings-validation.zip)을 다운로드합니다. 2. Windows 사전 WIG 검증 스크립트를 실행하고 결과를 확인합니다. 3. 유효성 검사에 실패할 경우 문제를 해결하고 유효성 검사가 성공할 때까지 유효성 검사 스크립트를 다시 실행하세요. 	<p>마이그레이션 엔지니어</p>

작업	설명	필요한 기술
페일세이프 AMI를 생성합니다.	<p>사전 WIG 검증이 통과한 후 다음과 같이 사전 통합 AMI를 생성합니다.</p> <ol style="list-style-type: none"> 1. 배포, [고급 스택 구성 요소, AMI, 생성을 선택합니다. 2. 생성 중에 Key=Name, Value=APPLICATION-ID_IngestReady 태그를 추가합니다. 3. AMI가 생성될 때까지 기다린 후 진행하세요. <p>자세한 내용은 AMS 설명서의 AMI 생성을 참조하세요.</p>	마이그레이션 엔지니어

인스턴스 수집 및 검증

작업	설명	필요한 기술
RFC를 제출하여 워크로드 인 제스트 스택을 생성합니다.	<p>변경 요청(RFC)을 제출하여 AMS WIGS를 시작하세요. 지침은 AMS 설명서의 워크로드 인 제스트 스택: 생성을 참조하세요. 그러면 워크로드 수집이 시작되고 백업 도구, Amazon EC2 관리 소프트웨어, 바이러스 백신 소프트웨어를 포함하여 AMS에 필요한 모든 소프트웨어가 설치됩니다.</p>	마이그레이션 엔지니어
성공적인 마이그레이션을 검증합니다.	<p>워크로드 통합이 완료되면 AMS에서 관리되는 인스턴스</p>	마이그레이션 엔지니어

작업	설명	필요한 기술
	<p>와 AMS에서 수집한 AMI를 볼 수 있습니다.</p> <ol style="list-style-type: none"> 1. 도메인 보안 인증 정보로 AMS 관리형 인스턴스에 로그인합니다. 2. 다음과 같이 도메인 가입을 검증합니다. <ol style="list-style-type: none"> a. Windows 탐색기에서 이 PC를 마우스 오른쪽 단추로 클릭하고 속성을 선택합니다. b. 장치 사양 섹션에서 도메인이 전체 장치 이름에 나타나는지 확인합니다. 3. 소스 및 대상 디스크 드라이브를 검증합니다. 	

대상 AMS 계정에서 인스턴스를 실행합니다.

작업	설명	필요한 기술
<p>RFC를 제출하여 EC2 스택을 생성합니다.</p>	<ol style="list-style-type: none"> 1. Windows 인스턴스의 AMS에서 수집한 AMI를 사용하여 AMS 설명서의 EC2 스택 인스턴스 생성에 나와 있는 지침에 따라 EC2 스택용 RFC를 준비합니다. EC2 스택 RFC에서 서버 이름, 태그, 대상 VPC, 대상 서브넷, 인스턴스 유형, 대상 보안 그룹, 통합 AMI, 역할 등 모든 파라미터를 제공합니다. 	<p>마이그레이션 엔지니어</p>

작업	설명	필요한 기술
	2. EC2 스택용 RFC를 제출한 다음 인스턴스가 성공적으로 생성될 때까지 기다립니다.	

관련 리소스

AWS 권장 가이드

- [Windows 기반 AWS Managed Services의 사전 워크로드 수집 활동 자동화](#)
- [Python을 사용하여 AMS에서 자동으로 RFC 생성](#)

AWS 설명서

- [AMS 워크로드 수집](#)
- [마이그레이션이 리소스를 변경하는 방법](#)
- [워크로드 마이그레이션: 표준 프로세스](#)

마케팅 리소스

- [AWS Managed Services](#)
- [AWS 관리형 서비스 FAQ](#)
- [AWS 관리형 서비스 리소스](#)
- [AWS 관리형 서비스 기능](#)

로그 전송을 사용하여 운영 중단 시간을 줄임으로써 Db2 for LUW를 Amazon EC2로 마이그레이션합니다.

펑카이 (AWS), 앰버리쉬 사타르카르 (AWS), 사우라브 샤프마 (AWS) 가 제작했습니다.

환경: 프로덕션	소스: 리눅스용 온프레미스 Db2	대상: Amazon EC2 기반 Db2
R 유형: 리호스팅	워크로드: IBM	기술: 마이그레이션, 데이터베이스

AWS 서비스: AWS Direct Connect, 아마존 EBS, 아마존 EC2, 아마존 S3, AWS 사이트 간 VPN

요약

고객이 LUW용 IBM Db2 (리눅스, UNIX, 윈도우) 워크로드를 Amazon Web Services (AWS) 로 마이그레이션할 때는 보유 라이선스 사용 (BYOL) 모델과 함께 Amazon Elastic Compute Cloud (Amazon EC2) 를 사용하는 것이 가장 빠른 방법입니다. 그러나 온프레미스 Db2에서 AWS로 대량의 데이터를 마이그레이션하는 것은 어려울 수 있으며, 특히 운영 중단 기간이 짧은 경우에는 더욱 그렇습니다. 많은 고객이 운영 중단 기간을 30분 미만으로 설정하려고 하는데, 이로 인해 데이터베이스 자체에 사용할 시간이 거의 없습니다.

이 패턴은 트랜잭션 로그 전달을 사용하여 짧은 중단 기간 내에 Db2 마이그레이션을 수행하는 방법을 다룹니다. 이 접근 방식은 리틀 엔디안 Linux 플랫폼의 Db2에 적용됩니다.

사전 조건 및 제한 사항

사전 조건

- 활성 상태의 AWS 계정.
- 온프레미스 파일 시스템 레이아웃과 일치하는 EC2 인스턴스에서 실행되는 Db2 인스턴스
- EC2 인스턴스에서 액세스 가능한 Amazon Simple Storage Service(S3) 버킷
- Amazon S3를 프로그래밍 방식으로 호출하기 위한 AWS 자격 증명 및 액세스 관리 (IAM) 정책 및 역할

- Amazon EC2와 온프레미스 서버의 동기화된 시간대 및 시스템 시계
- [AWS Site-to-Site VPN](#) 또는 [AWS Direct Connect](#)를 통해 AWS에 연결된 온프레미스 네트워크

제한 사항

- Db2 온프레미스 인스턴스와 Amazon EC2는 동일한 [플랫폼 패밀리](#)에 있어야 합니다.
- Db2 온프레미스 워크로드를 기록해야 합니다. 기록되지 않은 모든 트랜잭션을 차단하려면 데이터베이스 blocknonlogged=yes 구성에서 설정합니다.

제품 버전

- LUW 버전 11.5.9 이상용 Db2

아키텍처

소스 기술 스택

- Db2 on Linux x86_64

대상 기술 스택

- Amazon EBS
- Amazon EC2
- AWS Identity and Access Management(IAM)
- Amazon S3
- AWS 사이트 간 VPN 또는 다이렉트 커넥트

대상 아키텍처

다음 다이어그램은 Amazon EC2의 Db2에 대한 VPN (가상 사설망) 연결을 통해 온프레미스에서 실행되는 Db2 인스턴스 하나를 보여줍니다. 점선은 데이터 센터와 AWS 클라우드 사이의 VPN 터널을 나타냅니다.

도구

서비스

- [AWS Command Line Interface\(AWS CLI\)](#)는 명령줄 셸에서 명령을 사용하여 AWS 서비스와 상호 작용할 수 있는 오픈 소스 도구입니다.
- [Direct Connect](#)를 사용하면 표준 Ethernet 광섬유 케이블을 통해 내부 네트워크를 Direct Connect 위치에 연결할 수 있습니다. 이 연결을 구성하면, 네트워크 경로에서 인터넷 서비스 제공업체를 우회하여 퍼블릭 서비스에 직접 가상 인터페이스를 생성할 수 있습니다.
- [Amazon Elastic Block Store\(Amazon EBS\)](#)는 Amazon Elastic Compute Cloud(Amazon EC2) 인스턴스와 함께 사용할 수 있는 블록 스토리지 볼륨을 제공합니다.
- [Amazon Elastic Compute Cloud\(Amazon EC2\)](#)는 AWS 클라우드에서 확장 가능한 컴퓨팅 용량을 제공합니다. 필요한 만큼 가상 서버를 시작하고 빠르게 스케일 업하거나 스케일 다운할 수 있습니다.
- [AWS Identity and Access Management\(IAM\)](#)를 사용하면 사용자에 대해 인증 및 권한 부여를 제어함으로써 AWS 리소스에 대한 액세스를 안전하게 관리할 수 있습니다.
- [Amazon Simple Storage Service\(S3\)](#)는 원하는 양의 데이터를 저장, 보호 및 검색하는 데 도움이 되는 클라우드 기반 객체 스토리지 서비스입니다.
- [AWS Site-to-Site VPN](#)을 사용하면 AWS에서 시작하는 인스턴스와 자체 원격 네트워크 간에 트래픽을 전달할 수 있습니다.

기타 도구

- [db2cli](#)는 Db2 대화형 CLI 명령입니다.

모범 사례

- 타겟 데이터베이스에서 [Amazon S3의 게이트웨이 엔드포인트](#)를 사용하여 Amazon S3의 데이터베이스 백업 이미지 및 로그 파일에 액세스합니다.
- 원본 데이터베이스에서 [Amazon PrivateLink S3용 AWS](#)를 사용하여 데이터베이스 백업 이미지와 로그 파일을 Amazon S3로 전송합니다.

에픽

환경 변수 설정

작업	설명	필요한 기술
환경 변수를 설정합니다.	<p>이 패턴은 다음 이름을 사용합니다:</p> <ul style="list-style-type: none"> 인스턴스 이름: db2inst1 데이터베이스 이름: SAMPLE <p>환경에 맞게 변경할 수 있습니다.</p>	DBA

온프레미스 Db2 서버 구성

작업	설명	필요한 기술
AWS CLI를 설치합니다.	<p>최신 버전의 AWS CLI를 다운로드하고 설치하려면 다음 명령을 실행합니다.</p> <pre>\$ curl "https://awscli.amazonaws.com/awscli-exe-linux-x86_64.zip" -o "awscliv2.zip" unzip awscliv2.zip sudo ./aws/install</pre>	리눅스 관리자
Db2 아카이브 로그의 로컬 대상을 설정합니다.	<p>Amazon EC2의 대상 데이터베이스를 온프레미스 소스 데이터베이스와 동기화된 상태로 유지하려면 원본에서 최신 트랜잭션 로그를 검색해야 합니다.</p>	DBA

작업	설명	필요한 기술
	<p>이 설정에서 /db2logs은(는) 소스의 LOGARCHMETH2 에 의해 스테이징 영역으로 설정됩니다. 이 디렉터리에 보관된 로그는 Amazon S3에 동기화되고 Amazon EC2의 Db2에서 액세스할 수 있습니다. 이 패턴은 AWS CLI 명령이 액세스할 수 없는 타사 공급업체 도구를 사용하도록 LOGARCHMETH1 구성되었을 수 LOGARCHMETH2 있기 때문에 사용합니다. 로그를 검색하려면 다음 명령을 실행합니다.</p> <pre>db2 connect to sample db2 update db cfg for SAMPLE using LOGARCHMETH2 disk:/db2logs</pre>	
<p>온라인 데이터베이스 백업을 실행합니다.</p>	<p>온라인 데이터베이스 백업을 실행하고 로컬 백업 파일 시스템에 저장합니다.</p> <pre>db2 backup db sample online to /backup</pre>	<p>DBA</p>

S3 버킷 및 IAM 정책 설정

작업	설명	필요한 기술
<p>S3 버킷을 생성합니다.</p>	<p>온프레미스 서버가 백업 Db2 이미지 및 로그 파일을 AWS에 전송할 S3 버킷을 생성합니다.</p>	<p>AWS 시스템 관리자</p>

작업	설명	필요한 기술
	<p>Amazon EC2에서도 버킷에 액세스할 수 있습니다.</p> <pre data-bbox="597 331 1026 491">aws s3api create-bucket --bucket logshipmig- db2 --region us-east-1</pre>	

작업	설명	필요한 기술
IAM 정책을 생성합니다.	<p>이 db2bucket.json 파일에는 Amazon S3 버킷에 액세스하기 위한 IAM 정책이 포함되어 있습니다.</p> <pre data-bbox="597 443 1029 1770"> { "Version": "2012-10-17", "Statement": [{ "Effect": "Allow", "Action": ["kms:GenerateDataK ey", "kms:Decrypt", "s3:PutObject", "s3:GetObject", "s3:AbortMultipart Upload", "s3:ListBucket", "s3:DeleteObject", "s3:GetObjectVersi on", "s3:ListMultipartU ploadParts"], "Resource": [</pre>	AWS 관리자, AWS 시스템 관리자

작업	설명	필요한 기술
	<pre data-bbox="609 247 1015 619"> "arn:aws:s3:::logs hipmig-db2/*", "arn:aws:s3:::logs hipmig-db2"] }] } </pre> <p data-bbox="592 661 1015 745">정책을 생성하려면 다음 AWS CLI 명령을 사용하십시오.</p> <pre data-bbox="609 787 1015 1060"> aws iam create-policy \ --policy-name db2s3policy \ --policy-document file://db2bucket.j son </pre> <p data-bbox="592 1102 1015 1270">JSON 출력에는 정책의 Amazon 리소스 이름 (ARN) 이 표시되며, <code>aws_account_id</code> 여기서 계정 ID를 나타냅니다.</p> <pre data-bbox="609 1312 1015 1470"> "Arn": "arn:aws: iam::aws_account_i d:policy/db2s3policy" </pre>	

작업	설명	필요한 기술
<p>IAM 정책을 EC2 인스턴스에서 사용하는 IAM 역할에 연결합니다.</p>	<p>대부분의 AWS 환경에서 실행 중인 EC2 인스턴스에는 시스템 관리자가 설정한 IAM 역할이 있습니다. IAM 역할이 설정되지 않은 경우, 역할을 생성하고 EC2 콘솔에서 IAM 역할 수정을 선택하여 역할을 Db2 데이터베이스를 호스팅하는 EC2 인스턴스에 연결합니다. 정책 ARN을 사용하여 IAM 정책을 IAM 역할에 연결합니다.</p> <pre data-bbox="594 772 1029 1136">aws iam attach-role-policy \ --policy-arn \ "arn:aws:iam::aws_account_id:policy/db2s3policy" \ --role-name \ db2s3role</pre> <p>정책이 연결되면 IAM 역할에 연결된 모든 EC2 인스턴스가 S3 버킷에 액세스할 수 있습니다.</p>	<p>AWS 관리자, AWS 시스템 관리자</p>

원본 데이터베이스 백업 이미지 및 로그 파일을 Amazon S3로 전송

작업	설명	필요한 기술
<p>온프레미스 Db2 서버에서 AWS CLI를 구성합니다.</p>	<p>이전 단계에서 Secret Access Key 생성한 Access Key ID 및 를 사용하여 AWS CLI를 구성합니다.</p>	<p>AWS 관리자, AWS 시스템 관리자</p>

작업	설명	필요한 기술
	<pre>\$ aws configure AWS Access Key ID [None]: ***** AWS Secret Access Key [None]: ***** ***** Default region name [None]: us-east-1 Default output format [None]: json</pre>	
<p>Amazon S3에 백업 이미지를 보냅니다.</p>	<p>이전에는 온라인 데이터베이스 백업이 /backup 로컬 디렉터리에 저장되었습니다. 해당 백업 이미지를 S3 버킷으로 보내려면 다음 명령을 실행합니다.</p> <pre>aws s3 sync /backup s3://logshipmig-db2/ SAMPLE_backup</pre>	<p>AWS 관리자, 마이그레이션 엔지니어</p>

작업	설명	필요한 기술
<p>Db2 아카이브 로그를 Amazon S3에 보냅니다.</p>	<p>온프레미스 Db2 아카이브 로그를 Amazon EC2의 대상 Db2 인스턴스에서 액세스할 수 있는 S3 버킷과 동기화합니다.</p> <pre data-bbox="592 441 1027 604">aws s3 sync /db2logs s3://logshipmig-db2/ SAMPLE_LOG</pre> <p>cron 또는 기타 예약 도구를 사용하여 이 명령을 주기적으로 실행하세요. 빈도는 원본 데이터베이스가 트랜잭션 로그 파일을 보관하는 빈도에 따라 달라집니다.</p>	<p>AWS 관리자, 마이그레이션 엔지니어</p>

Amazon EC2의 Db2를 Amazon S3에 연결하고 데이터베이스 동기화를 시작합니다.

작업	설명	필요한 기술
<p>PKCS12 키스토어를 생성하세요.</p>	<p>Db2는 공개 키 암호화 표준 (PKCS) 암호화 키스토어를 사용하여 AWS 액세스 키를 안전하게 유지합니다. 키스토어를 생성하고 이를 사용하도록 원본 Db2 인스턴스를 구성합니다.</p> <pre data-bbox="592 1564 1027 1848">gsk8capicmd_64 -keydb -create -db "/home/db 2inst1/.keystore/d b2s3.p12" -pw "<password>" -type pkcs12 - stash</pre>	<p>DBA</p>

작업	설명	필요한 기술
	<pre>db2 "update dbm cfg using keystore_ location /home/db2 inst1/.keystore/db 2s3.p12 keystore_type pkcs12"</pre>	
<p>Db2 스토리지 액세스 에이리어스를 생성합니다.</p>	<p>스토리지 액세스 별칭을 만들려면 다음 스크립트 구문을 사용하십시오.</p> <pre>db2 "catalog storage access alias <alias_name> vendor S3 server <S3 endpoint> container '<bucket_name>'"</pre> <p>예를 들어 스크립트는 다음과 같을 수 있습니다.</p> <pre>db2 "catalog storage access alias DB2AWSS3 vendor S3 server s3.us-east-1.amazonaws.com container 'logshipmig-db2'"</pre>	DBA

작업	설명	필요한 기술
<p>스테이징 영역을 설정합니다.</p>	<p>기본적으로 Db2는 DB2_OBJECT_STORAGE_LOCAL_STAGING_PATH 를 준비 영역으로 사용하여 Amazon S3에 파일을 업로드하고 다운로드합니다. 기본 경로는 Db2 파티션 번호를 xxxx 참조하여 인스턴스 홈 디렉터리 sqllib/tmp/RemoteStorage.xxxx 아래에 있습니다. 참고로 스테이징 영역에는 백업 이미지와 로그 파일을 보관할 수 있는 충분한 용량이 있어야 합니다. 레지스트리를 사용하여 스테이징 영역이 다른 디렉터리를 가리키도록 할 수 있습니다.</p> <p>또한 데이터베이스 백업 및 복원을 위한 Amazon S3 스테이징 영역을 우회하려면 DB2_ENABLE_COS_SDK=ON DB2_OBJECT_STORAGE_SETTINGS=EnableStreamingRestore , 및 awssdk 라이브러리 링크를 사용하는 것이 좋습니다.</p> <div data-bbox="592 1575 1031 1864" style="border: 1px solid #ccc; border-radius: 10px; padding: 10px; margin-top: 10px;"> <pre>#By root: cp -rp /home/db2inst1/sqllib/lib64/awssdk/RHEL/7.6/* /home/db2inst1/sqllib/lib64/ #By db2 instance owner:</pre> </div>	<p>DBA</p>

작업	설명	필요한 기술
	<pre>db2set DB2_OBJEC T_STORAGE_LOCAL_ST AGING_PATH=/db2stage db2set DB2_ENABL E_COS_SDK=ON Db2set DB2_OBJEC T_STORAGE_SETTINGS =EnableStreamingRe store db2stop db2start</pre>	
<p>백업 이미지에서 데이터베이스를 복원합니다.</p>	<p>S3 버킷의 백업 이미지로부터 Amazon EC2의 대상 데이터베이스를 복원합니다.</p> <pre>db2 restore db sample from DB2REMOTE:// DB2AWSS3/logshipmig- db2/SAMPLE_backup replace existing</pre>	DBA

작업	설명	필요한 기술
<p>데이터베이스를 롤포워드하십시오.</p>	<p>복원이 완료되면 대상 데이터베이스는 롤포워드 보류 상태가 됩니다. Db2가 트랜잭션 로그 파일을 가져올 위치를 알 수 LOGARCHMETH2 있도록 LOGARCHMETH1 구성하십시오.</p> <pre data-bbox="594 583 1026 903"> db2 update db cfg for SAMPLE using LOGARCHME TH1 'DB2REMOTE://DB2AW SS3//SAMPLE_LOGS/' db2 update db cfg for SAMPLE using LOGARCHME TH2 OFF </pre> <p>데이터베이스 롤포워드 시작:</p> <pre data-bbox="594 1012 1026 1171"> db2 ROLLFORWARD DATABASE sample to END OF LOGS </pre> <p>이 명령은 S3 버킷으로 전송된 모든 로그 파일을 처리합니다. 온프레미스 Db2 서버의 s3 sync 명령 빈도에 따라 주기적으로 실행합니다. 예를 들어 매 시간마다 s3 sync 실행되고 모든 로그 파일을 동기화하는데 10분이 걸린다면 매시간 10분마다 실행되도록 명령을 설정하십시오.</p>	DBA

컷오버 기간 동안 Amazon EC2의 Db2를 온라인 상태로 전환하십시오.

작업	설명	필요한 기술
<p>대상 데이터베이스를 온라인으로 전환합니다.</p>	<p>컷오버 기간 중에 다음 중 하나를 수행합니다.</p> <ul style="list-style-type: none"> 온프레미스 데이터베이스를 넣고 ADMIN MODE s3 sync 명령을 실행하여 마지막 트랜잭션 로그를 강제로 보관합니다. 데이터베이스를 종료합니다. <p>마지막 트랜잭션 로그를 Amazon S3에 동기화한 후 ROLLFORWARD 명령을 마지막으로 실행합니다.</p> <pre data-bbox="592 1035 1027 1843"> db2 rollforward DB sample to END OF LOGS db2 rollforward DB sample complete Rollforward Status Rollforward status = not pending DB20000I The ROLLFORWARD command completed successfully. db2 activate db sample DB20000I The ACTIVATE DATABASE command </pre>	<p>DBA</p>

작업	설명	필요한 기술
	<p>completed successfully.</p> <p>대상 데이터베이스를 온라인으로 전환하고 Amazon EC2의 Db2를 가리키도록 애플리케이션 연결을 지정합니다.</p>	

문제 해결

문제	Solution
<p>여러 데이터베이스의 인스턴스 이름과 데이터베이스 이름이 서로 다른 호스트(DEV, QA, PROD)에 있는 경우 백업과 로그는 동일한 하위 디렉토리로 이동할 수 있습니다.</p>	<p>DEV, QA 및 PROD에 서로 다른 S3 버킷을 사용하고 혼동을 피하기 위해 호스트 이름을 하위 디렉터리 접두사로 추가하십시오.</p>
<p>같은 위치에 백업 이미지가 여러 개 있는 경우 복원할 때 다음과 같은 오류가 발생합니다.</p> <pre>SQL2522N More than one backup file matches the time stamp value provided for the backed up database image.</pre>	<p>restore명령에서 백업의 타임스탬프를 추가합니다.</p> <pre>db2 restore db sample from DB2REMOTE://DB2AWSS3/logshimig-db2/SAMPLE_backup taken at 20230628164042 replace existing</pre>

관련 리소스

- [서로 다른 운영 체제와 하드웨어 플랫폼 간의 Db2 백업 및 복원 작업](#)
- [Db2 스토리지 액세스 별칭 및 DB2REMOTE 설정](#)
- [Db2 롤포워드 명령](#)
- [Db2 보조 로그 아카이브 메서드](#)

고가용성 재해 복구 기능을 갖춘 Db2 for LUW를 Amazon EC2로 마이그레이션하세요.

작성자: Feng Cai(AWS), Aruna Gangireddy(AWS), Venkatesan Govindan(AWS)

환경: 프로덕션	소스: 온프레미스 LUW용 IBM Db2	대상: Amazon EC2 기반 Db2
R 유형: 리호스팅	워크로드: IBM	기술: 마이그레이션, 데이터베이스, 운영 체제

AWS 서비스: AWS Direct Connect, 아마존 EC2, 아마존 S3, AWS 사이트 간 VPN

요약

고객이 IBM Db2 LUW (리눅스, UNIX, 윈도우) 워크로드를 Amazon Web Services (AWS) 로 마이그레이션할 때는 보유 라이선스 사용 (BYOL) 모델과 함께 Amazon Elastic Compute Cloud (Amazon EC2) 를 사용하는 것이 가장 빠른 방법입니다. 그러나 온프레미스 Db2에서 AWS로 대량의 데이터를 마이그레이션하는 것은 어려울 수 있으며, 특히 운영 중단 기간이 짧은 경우에는 더욱 그렇습니다. 많은 고객이 운영 중단 기간을 30분 미만으로 설정하려고 하는데, 이로 인해 데이터베이스 자체에 사용할 시간이 거의 없습니다.

이 패턴은 Db2 고가용성 재해 복구(HADR)를 사용하여 운영 중단 기간이 짧은 Db2 마이그레이션을 수행하는 방법을 다룹니다. 이 접근 방식은 리틀 인디안 Linux 플랫폼에 있고 데이터 파티셔닝 기능(DPF) 을 사용하지 않는 Db2 데이터베이스에 적용됩니다.

사전 조건 및 제한 사항

사전 조건

- 활성 상태의 AWS 계정.
- 온프레미스 파일 시스템 레이아웃과 일치하는 Amazon EC2 인스턴스에서 실행되는 Db2 인스턴스
- EC2 인스턴스에서 액세스 가능한 Amazon Simple Storage Service(S3) 버킷
- Amazon S3를 프로그래밍 방식으로 호출하기 위한 AWS 자격 증명 및 액세스 관리 (IAM) 정책 및 역할

- Amazon EC2와 온프레미스 서버의 동기화된 시간대 및 시스템 시계
- [AWS Site-to-Site VPN](#) 또는 [AWS Direct Connect](#)를 통해 AWS에 연결된 온프레미스 네트워크
- HADR 포트를 통한 온프레미스 서버와 Amazon EC2 간의 통신

제한 사항

- Db2 온프레미스 인스턴스와 Amazon EC2는 동일한 [플랫폼 패밀리](#)에 있어야 합니다.
- HADR은 분할된 데이터베이스 환경에서는 지원되지 않습니다.
- HADR은 데이터베이스 로그 파일에 원시 I/O(직접 디스크 액세스) 사용을 지원하지 않습니다.
- HADR은 무한 로깅을 지원하지 않습니다.
- LOGINDEXBUILD은(는) YES(으)로 설정해야 하며, 이렇게 하면 인덱스 재구축을 위한 로그 사용량이 늘어납니다.
- Db2 온프레미스 워크로드를 기록해야 합니다. 기록되지 않은 모든 트랜잭션을 차단하도록 데이터베이스 구성에서 blocknonlogged=yes을(를) 설정합니다.

제품 버전

- LUW 버전 11.5.9 이상용 Db2

아키텍처

소스 기술 스택

- Db2 on Linux x86_64

대상 기술 스택

- Amazon EC2
- AWS Identity and Access Management(IAM)
- Amazon S3
- AWS Site-to-Site VPN

대상 아키텍처

다음 다이어그램에서는 Db2 온프레미스가 db2-server1에서 기본으로 실행되고 있습니다. 두 개의 HADR 대기 타겟이 있습니다. 하나의 스탠바이 타겟은 온프레미스이며 선택 사항입니다. 다른 스탠바이 타겟 db2-ec2은(는) Amazon EC2에 있습니다. 데이터베이스가 AWS로 db2-ec2 이전되면 기본 데이터베이스가 됩니다.

1. 로그는 기본 온프레미스 데이터베이스에서 스탠바이 온프레미스 데이터베이스로 스트리밍됩니다.
2. Db2 HADR을 사용하면 로그가 사이트 간 VPN을 통해 기본 온프레미스 데이터베이스에서 Amazon EC2의 Db2로 스트리밍됩니다.
3. Db2 백업 및 아카이브 로그는 기본 온프레미스 데이터베이스에서 AWS의 S3 버킷으로 전송됩니다.

도구

서비스

- [AWS Command Line Interface\(AWS CLI\)](#)는 명령줄 셸에서 명령을 사용하여 AWS 서비스와 상호 작용할 수 있는 오픈 소스 도구입니다.
- [Direct Connect](#)를 사용하면 표준 Ethernet 광섬유 케이블을 통해 내부 네트워크를 Direct Connect 위치에 연결할 수 있습니다. 이 연결을 구성하면, 네트워크 경로에서 인터넷 서비스 제공업체를 우회하여 퍼블릭 서비스에 직접 가상 인터페이스를 생성할 수 있습니다.
- [Amazon Elastic Compute Cloud\(Amazon EC2\)](#)는 AWS 클라우드에서 확장 가능한 컴퓨팅 용량을 제공합니다. 필요한 만큼 가상 서버를 시작하고 빠르게 스케일 업하거나 스케일 다운할 수 있습니다.
- [AWS Identity and Access Management\(IAM\)](#)를 사용하면 사용자에 대해 인증 및 권한 부여를 제어함으로써 AWS 리소스에 대한 액세스를 안전하게 관리할 수 있습니다.
- [Amazon Simple Storage Service\(S3\)](#)는 원하는 양의 데이터를 저장, 보호 및 검색하는 데 도움이 되는 클라우드 기반 객체 스토리지 서비스입니다.
- [AWS Site-to-Site VPN](#)을 사용하면 AWS에서 시작하는 인스턴스와 자체 원격 네트워크 간에 트래픽을 전달할 수 있습니다.

기타 도구

- [db2cli](#)는 Db2 대화형 CLI 명령입니다.

모범 사례

- 타겟 데이터베이스에서 [Amazon S3의 게이트웨이 엔드포인트](#)를 사용하여 Amazon S3의 데이터베이스 백업 이미지 및 로그 파일에 액세스합니다.
- 원본 데이터베이스에서 [Amazon PrivateLink S3용 AWS](#)를 사용하여 데이터베이스 백업 이미지와 로그 파일을 Amazon S3로 전송합니다.

에픽

환경 변수 설정

작업	설명	필요한 기술
환경 변수를 설정합니다.	<p>이 패턴은 다음 이름과 포트를 사용합니다.</p> <ol style="list-style-type: none"> 1. Db2 온프레미스 호스트 이름: db2-server1 2. HADR 대기 호스트 이름: db2-server2 (HADR이 현재 온프레미스에서 실행 중인 경우) 3. Amazon EC2 호스트 이름: db2-ec2 4. 인스턴스 이름: db2inst1 5. 데이터베이스 이름: SAMPLE 6. 하드 포트: <ul style="list-style-type: none"> • db2-server1: 50010 • db2-server2: 50011 • db2-ec2: 50012 <p>환경에 맞게 변경할 수 있습니다.</p>	DBA

온프레미스 Db2 서버 구성

작업	설명	필요한 기술
<p>AWS CLI를 설정합니다.</p>	<p>최신 버전의 AWS CLI를 다운로드하고 설치하려면 다음 명령을 실행합니다.</p> <pre data-bbox="592 499 1027 816"> \$ curl "https://awscli.amazonaws.com/awscli-exe-linux-x86_64.zip" -o "awscliv2.zip" unzip awscliv2.zip sudo ./aws/install </pre>	<p>리눅스 관리자</p>
<p>Db2 아카이브 로그의 로컬 대상을 설정합니다.</p>	<p>대량 업데이트 일괄 작업 및 네트워크 속도 저하와 같은 조건으로 인해 HADR 대기 서버가 지연될 수 있습니다. 이를 따라잡으려면 스탠바이 서버에 기본 서버의 트랜잭션 로그가 필요합니다. 로그를 요청하는 순서는 다음과 같습니다.</p> <ul data-bbox="592 1268 1027 1661" style="list-style-type: none"> • 기본 서버의 활성 로그 디렉터리 • 스탠바이 서버의 LOGARCHMETH1 또는 LOGARCHMETH2 위치 • 기본 서버의 LOGARCHMETH1 또는 LOGARCHMETH2 위치 <p>이 설정에서 /db2logs은(는) 소스의 LOGARCHMETH2 에 의해 스테이징 영역으로 설정됨</p>	<p>DBA</p>

작업	설명	필요한 기술
	<p>니다. 이 디렉터리에 보관된 로그는 Amazon S3에 동기화되고 Amazon EC2의 Db2에서 액세스할 수 있습니다. 이 패턴이 사용하는 LOGARCHMETH2 이유는 AWS CLI 명령으로 액세스할 수 없는 타사 공급업체 도구를 사용하도록 LOGARCHMETH1 구성되었을 수 있기 때문입니다.</p> <pre data-bbox="597 716 1024 911"> db2 connect to sample db2 update db cfg for SAMPLE using LOGARCHMETH2 disk:/db2logs </pre>	

<p>온라인 데이터베이스 백업을 실행합니다.</p>	<p>온라인 데이터베이스 백업을 실행하고 로컬 백업 파일 시스템에 저장합니다.</p> <pre data-bbox="597 1121 1024 1236"> db2 backup db sample online to /backup </pre>	<p>DBA</p>
------------------------------	--	------------

S3 버킷 및 IAM 정책 설정

작업	설명	필요한 기술
<p>S3 버킷을 생성합니다.</p>	<p>온프레미스 서버가 백업 Db2 이미지 및 로그 파일을 AWS에 전송할 S3 버킷을 생성합니다. Amazon EC2에서 버킷에 액세스할 수 있습니다.</p>	<p>AWS 관리자</p>

작업	설명	필요한 기술
	<pre>aws s3api create-bucket --bucket hadrmig-db2 --region us-east-1</pre>	

작업	설명	필요한 기술
IAM 정책을 생성합니다.	<p>이 db2bucket.json 파일에는 S3 버킷에 액세스하기 위한 IAM 정책이 포함되어 있습니다.</p> <pre data-bbox="597 443 1027 1770"> { "Version": "2012-10-17", "Statement": [{ "Effect": "Allow", "Action": ["kms:GenerateDataK ey", "kms:Decrypt", "s3:PutObject", "s3:GetObject", "s3:AbortMultipart Upload", "s3:ListBucket", "s3:DeleteObject", "s3:GetObjectVersi on", "s3:ListMultipartU ploadParts"], "Resource": [</pre>	AWS 관리자, AWS 시스템 관리자

작업	설명	필요한 기술
	<pre data-bbox="609 247 1015 619"> "arn:aws:s3:::hadr mig-db2/*", "arn:aws:s3:::hadr mig-db2"] }] } </pre> <p data-bbox="592 661 1015 745">정책을 생성하려면 다음 AWS CLI 명령을 사용하십시오.</p> <pre data-bbox="609 787 1015 1060"> aws iam create-policy \ --policy-name db2s3hapolicy \ --policy-document file://db2bucket.j son </pre> <p data-bbox="592 1102 1015 1270">JSON 출력에는 정책의 Amazon 리소스 이름 (ARN) 이 표시되며, <code>aws_account_id</code> 여기서 계정 ID를 나타냅니다.</p> <pre data-bbox="609 1312 1015 1501"> "Arn": "arn:aws: iam::aws_account_i d:policy/db2s3hapo licy" </pre>	

작업	설명	필요한 기술
<p>IAM 정책을 IAM 역할에 연결합니다.</p>	<p>일반적으로 Db2를 실행하는 EC2 인스턴스에는 시스템 관리자가 할당한 IAM 역할이 있습니다. IAM 역할이 할당되지 않은 경우 Amazon EC2 콘솔에서 IAM 역할 수정을 선택할 수 있습니다.</p> <p>IAM 정책을 EC2 인스턴스와 연결된 IAM 역할에 연결합니다. 정책이 연결되면 EC2 인스턴스는 S3 버킷에 액세스할 수 있습니다.</p> <pre data-bbox="597 856 1026 1129">aws iam attach-role-policy --policy-arn "arn:aws:iam::aws_account_id:policy/db2s3hapolicy" --role-name db2s3harole</pre>	

원본 데이터베이스 백업 이미지 및 로그 파일을 Amazon S3로 전송

작업	설명	필요한 기술
<p>온프레미스 Db2 서버에서 AWS CLI를 구성합니다.</p>	<p>이전에 생성한 and를 사용하여 AWS CLI를 Access Key ID 구성합니다. Secret Access Key</p> <pre data-bbox="597 1633 1026 1883">\$ aws configure AWS Access Key ID [None]: ***** AWS Secret Access Key [None]: ***** *****</pre>	<p>AWS 관리자, AWS 시스템 관리자</p>

작업	설명	필요한 기술
	<pre>Default region name [None]: us-east-1 Default output format [None]: json</pre>	
<p>Amazon S3에 백업 이미지를 보냅니다.</p>	<p>이전에는 온라인 데이터베이스 백업이 /backup 로컬 디렉터리에 저장되었습니다. 해당 백업 이미지를 S3 버킷으로 보내려면 다음 명령을 실행합니다.</p> <pre>aws s3 sync /backup s3://hadrmig-db2/S AMPLE_backup</pre>	<p>AWS 관리자, AWS 시스템 관리자</p>
<p>Db2 아카이브 로그를 Amazon S3에 보냅니다.</p>	<p>온프레미스 Db2 아카이브 로그를 Amazon EC2의 대상 Db2 인스턴스에서 액세스할 수 있는 Amazon S3 버킷과 동기화합니다.</p> <pre>aws s3 sync /db2logs s3://hadrmig-db2/S AMPLE_LOGS</pre> <p>cron 또는 기타 예약 도구를 사용하여 이 명령을 주기적으로 실행하세요. 빈도는 원본 데이터베이스가 트랜잭션 로그 파일을 보관하는 빈도에 따라 달라집니다.</p>	

Amazon EC2의 Db2를 Amazon S3에 연결하고 초기 데이터베이스 동기화를 시작합니다.

작업	설명	필요한 기술
PKCS12 키스토어를 생성하세요.	<p>Db2는 공개 키 암호화 표준 (PKCS) 암호화 키스토어를 사용하여 AWS 액세스 키를 안전하게 유지합니다. 키스토어를 생성하고 이를 사용하도록 원본 Db2를 구성합니다.</p> <pre>gsk8capicmd_64 -keydb -create -db "/home/db 2inst1/.keystore/d b2s3.p12" -pw "<password>" -type pkcs12 - stash db2 "update dbm cfg using keystore_ location /home/db2 inst1/.keystore/db 2s3.p12 keystore_type pkcs12"</pre>	DBA
Db2 스토리지 액세스 에이리어스를 생성합니다.	<p>Db2는 스토리지 액세스 에이리어스를 사용하여, INGEST, LOAD, BACKUP DATABASE, RESTORE DATABASE 명령으로 Amazon S3에 직접 액세스합니다.</p> <p>EC2 인스턴스에 IAM 역할을 할당했으므로 필수는 아니기 때문입니다. USER PASSWORD</p> <pre>db2 "catalog storage access alias <alias_name> vendor S3"</pre>	DBA

작업	설명	필요한 기술
	<pre>server <S3 endpoint> container '<bucket_ name>' "</pre> <p>예를 들어 스크립트는 다음과 같을 수 있습니다.</p> <pre>db2 "catalog storage access alias DB2AWSS3 vendor S3 server s3.us-east-1.amazo naws.com container 'hadrmig-db2' "</pre>	

작업	설명	필요한 기술
<p>스테이징 영역을 설정합니다.</p>	<p>데이터베이스 백업 및 복원을 위한 Amazon S3 스테이징 영역을 우회하려면 DB2_ENABLE_COS_SDK=ON DB2_OBJEC T_STORAGE_SETTINGS=EnableStreamingRestore , 및 awssdk 라이브러리 링크를 사용하는 것이 좋습니다.</p> <pre data-bbox="597 730 1026 1444"> #By root: cp -rp /home/db2inst1/ sqllib/lib64/awssdk/ RHEL/7.6/* /home/db2 inst1/sqllib/lib64/ #By db2 instance owner: db2set DB2_OBJEC T_STORAGE_LOCAL_ST AGING_PATH=/db2stage db2set DB2_ENABL E_COS_SDK=ON db2set DB2_OBJEC T_STORAGE_LOCAL_ST AGING_PATH=/db2stage db2stop db2start </pre>	DBA

작업	설명	필요한 기술
백업 이미지에서 데이터베이스를 복원합니다.	<p>S3 버킷의 백업 이미지로부터 Amazon EC2의 대상 데이터베이스를 복원합니다.</p> <pre>db2 create db sample on /data1 db2 restore db sample from DB2REMOTE:// DB2AWSS3/hadrmig-db2/ SAMPLE_backup replace existing</pre>	DBA

온프레미스에서 하드 드라이브 없이 HARD 설치

작업	설명	필요한 기술
온프레미스 Db2 서버를 기본 서버로 구성합니다.	<p>db2-server1 에서 HADR(온-프레미스 소스)의 데이터베이스 구성 설정을 기본으로 업데이트하세요. 트랜잭션 응답 시간이 가장 짧은 HADR_SYNCMODE SUPERASYNC 모드로 설정:</p> <pre>db2 update db cfg for sample using HADR_LOCAL_HOST db2-server1 HADR_LOCAL_SVC 50010 HADR_REMOTE_HOST db2-ec2 HADR_REMOTE_SVC 50012 HADR_REMOTE_INST db2inst1 HADR_SYNCMODE SUPERASYNC DB20000 I The UPDATE DATABASE</pre>	DBA

작업	설명	필요한 기술
	<p>CONFIGURATION command completed successfully</p> <p>온프레미스 데이터 센터와 AWS 간에 일부 네트워크 지연이 발생할 것으로 예상됩니다. (네트워크 안정성에 따라 다른 HADR_SYNCMODE 값을 설정할 수 있습니다. 자세한 내용은 관련 리소스 섹션을 참조하세요.</p>	
<p>타겟 데이터베이스 로그 아카이브 타겟을 변경합니다.</p>	<p>Amazon EC2 환경에 맞게 대상 데이터베이스 로그 아카이브 대상을 변경합니다.</p> <pre data-bbox="594 968 1027 1367"> db2 update db cfg for SAMPLE using LOGARCHME TH1 'DB2REMOTE://DB2AW SS3//SAMPLE_LOGS/' LOGARCHMETH2 OFF DB20000I The UPDATE DATABASE CONFIGURA TION command completed successfully </pre>	<p>DBA</p>

작업	설명	필요한 기술
<p>Amazon EC2 서버에서 Db2용 HADR을 구성합니다.</p>	<p>대기 모드로 설정된 HADR의 데이터베이스 구성을 업데이트 하십시오db2-ec2.</p> <pre>db2 update db cfg for sample using HADR_LOCAL_HOST db2-ec2 HADR_LOCA L_SVC 50012 HADR_REMO TE_HOST db2-server1 HADR_REMOTE_SVC 50010 HADR_REMOTE_INST db2inst1 HADR_SYNC MODE SUPERASYN C DB20000I The UPDATE DATABASE CONFIGURATION command completed successfu lly</pre>	<p>DBA</p>

작업	설명	필요한 기술
<p>HADR 설정을 확인합니다.</p>	<p>소스 및 타겟 Db2 서버의 HADR 파라미터를 확인합니다.</p> <p>설정이 db2-server1 켜져 있는지 확인하려면 다음 명령을 실행합니다.</p> <pre data-bbox="597 520 1024 1841"> db2 get db cfg for sample grep HADR HADR database role = PRIMARY HADR local host name (HADR_LOCAL_HOST) = db2-server1 HADR local service name (HADR_LOCAL_SVC) = 50010 HADR remote host name (HADR_REMOTE_HOST) = db2-ec2 HADR remote service name (HADR_REMOTE_SVC) = 50012 HADR instance name of remote server (HADR_REMOTE_INST) = db2inst1 HADR timeout value (HADR_TIMEOUT) = 120 HADR target list (HADR_TARGET_LIST) = HADR log write synchronization mode (HADR_SYNCMODE) = NEARSYNC </pre>	<p>DBA</p>

작업	설명	필요한 기술
	<pre>HADR spool log data limit (4KB) (HADR_SPOOL_LIMIT) = AUTOMATIC(52000) HADR log replay delay (seconds) (HADR_REP LAY_DELAY) = 0 HADR peer window duration (seconds) (HADR_PEER_WINDOW) = 0 HADR SSL certifica te label (HADR_SSL_LABEL) = HADR SSL Hostname Validation (HADR_SSL_HOST_VAL) = OFF</pre> <p>설정이 db2-ec2 켜져 있는지 확인하려면 다음 명령을 실행합니다.</p> <pre>db2 get db cfg for sample grep HADR HADR database role = STANDBY HADR local host name (HADR_LOCAL_HOST) = db2-ec2 HADR local service name (HADR_LOCAL_SVC) = 50012 HADR remote host name (HADR_REMOTE_HOST) = db2-serve r1 HADR remote service name</pre>	

작업	설명	필요한 기술
	<pre> (HADR_REMOTE_SVC) = 50010 HADR instance name of remote server (HADR_REMOTE_INST) = db2inst1 HADR timeout value (HADR_TIMEOUT) = 120 HADR target list (HADR_TAR GET_LIST) = HADR log write synchronization mode (HADR_SYNCMODE) = SUPERASYNC HADR spool log data limit (4KB) (HADR_SPOOL_LIMIT) = AUTOMATIC(52000) HADR log replay delay (seconds) (HADR_REP LAY_DELAY) = 0 HADR peer window duration (seconds) (HADR_PEER_WINDOW) = 0 HADR SSL certifica te label (HADR_SSL_LABEL) = HADR SSL Hostname Validation (HADR_SSL_HOST_VAL) = OFF HADR_LOCAL_HOST , HADR_LOCAL_SVC , HADR_REMOTE_HOST , HADR_REMOTE_SVC 파라미 터는 기본 HADR 설정과 대기 </pre>	

작업	설명	필요한 기술
	HADR 설정 하나를 나타냅니다.	
Db2 HADR 인스턴스를 시작합니다.	<p>먼저 스탠바이 서버에서 db2-ec2 Db2 HADR 인스턴스를 시작합니다.</p> <pre data-bbox="594 506 1027 785">db2 start hadr on db sample as standby DB20000I The START HADR ON DATABASE command completed successfully.</pre> <p>기본 (소스) 서버에서 Db2 HADR을 시작합니다. db2-server1</p> <pre data-bbox="594 989 1027 1268">db2 start hadr on db sample as primary DB20000I The START HADR ON DATABASE command completed successfully.</pre> <p>온프레미스에서 Db2와 Amazon EC2 사이의 HADR 연결이 이제 성공적으로 설정되었습니다. Db2 기본 서버 db2-server1 은(는) 트랜잭션 로그 기록을 db2-ec2(으)로 실시간으로 스트리밍하기 시작합니다.</p>	DBA

온프레미스에 HADR이 있는 경우 HADR을 설정하세요.

작업	설명	필요한 기술
<p>Amazon EC2의 Db2를 보조 예비 복제본으로 추가합니다.</p>	<p>HADR이 온프레미스 Db2 인스턴스에서 실행 중인 경우, 다음 명령을 HADR_TARGET_LIST 실행하여 Amazon EC2의 Db2를 보조 예비 인스턴스로 추가할 수 있습니다. db2-ec2</p> <pre>db2 update db cfg for sample using HADR_LOCAL_HOST db2-ec2 HADR_LOCA L_SVC 50012 HADR_REMO TE_HOST db2-server1 HADR_REMOTE_SVC 50010 HADR_REMOTE_INST db2inst1 HADR_SYNC MODE SUPERASYN C DB20000I The UPDATE DATABASE CONFIGURATION command completed successfu lly. db2 update db cfg for sample using HADR_TARGET_LIST "db2-server1:50010 db2-server2:50011 " DB20000I The UPDATE DATABASE CONFIGURATION command completed successfu lly.</pre>	<p>DBA</p>

작업	설명	필요한 기술
<p>온프레미스 서버에 보조 대기 정보를 추가합니다.</p>	<p>두 개의 온프레미스 서버(기본 및 대기)에서 HADR_TARGET_LIST 을(를) 업데이트하세요.</p> <p>이제 다음 코드를 db2-server1 실행합니다.</p> <pre>db2 update db cfg for sample using HADR_TARGET_LIST "db2-server2:50011 db2-ec2:50012" DB20000I</pre> <p>The UPDATE DATABASE CONFIGURATION command completed successfully. SQL1363W One or more of the parameters submitted for immediate modification were not changed dynamically. For these configuration parameters, the database must be shutdown and reactivated before the configuration parameter changes become effective.</p> <p>db2-server2 켜기, 다음 코드를 실행합니다.</p> <pre>db2 update db cfg for sample using HADR_TARGET_LIST "db2-server1:50011 db2-ec2:50012" DB20000I</pre>	DBA

작업	설명	필요한 기술
	<pre>ET_LIST "db2-server1:50010 db2-ec2:50012" DB2000I The UPDATE DATABASE CONFIGURATION command completed successfully. SQL1363W One or more of the parameters submitted for immediate modification were not changed dynamically. For these configuration parameters, the database must be shutdown and reactivated before the configuration parameter changes become effective.</pre>	

작업	설명	필요한 기술
HADR 설정을 확인합니다.	<p>소스 및 타겟 Db2 서버의 HADR 파라미터를 확인합니다.</p> <p>db2-server1 켜기, 다음 코드를 실행합니다.</p> <pre data-bbox="592 472 1031 1839"> db2 get db cfg for sample grep HADR HADR database role = PRIMARY HADR local host name (HADR_LOCAL_HOST) = db2-server1 HADR local service name (HADR_LOCAL_SVC) = 50010 HADR remote host name (HADR_REMOTE_HOST) = db2-server2 HADR remote service name (HADR_REMOTE_SVC) = 50011 HADR instance name of remote server (HADR_REMOTE_INST) = db2inst1 HADR timeout value (HADR_TIMEOUT) = 120 HADR target list (HADR_TARGET_LIST) = db2-server2:50011 db2-ec2:50012 HADR log write synchronization mode </pre>	

작업	설명	필요한 기술
	<pre> (HADR_SYNCMODE) = NEARSYNC HADR spool log data limit (4KB) (HADR_SPOOL_LIMIT) = AUTOMATIC(52000) HADR log replay delay (seconds) (HADR_REP LAY_DELAY) = 0 HADR peer window duration (seconds) (HADR_PEER_WINDOW) = 0 HADR SSL certifica te label (HADR_SSL_LABEL) = HADR SSL Hostname Validation (HADR_SSL_HOST_VAL) = OFF </pre> <p>db2-server2 켜기, 다음 코드를 실행합니다.</p> <pre> db2 get db cfg for sample grep HADR HADR database role = STANDBY HADR local host name (HADR_LOCAL_HOST) = db2-server2 HADR local service name (HADR_LOCAL_SVC) = 50011 HADR remote host name (HADR_REMOTE_HOST) = db2-server1 HADR remote service name </pre>	

작업	설명	필요한 기술
	<pre> (HADR_REMOTE_SVC) = 50010 HADR instance name of remote server (HADR_REMOTE_INST) = db2inst1 HADR timeout value (HADR_TIMEOUT) = 120 HADR target list (HADR_TAR GET_LIST) = db2-serve r1:50010 db2-ec2:5 0012 HADR log write synchronization mode (HADR_SYNCMODE) = NEARSYNC HADR spool log data limit (4KB) (HADR_SPOOL_LIMIT) = AUTOMATIC(52000) HADR log replay delay (seconds) (HADR_REP LAY_DELAY) = 0 HADR peer window duration (seconds) (HADR_PEER_WINDOW) = 0 HADR SSL certifica te label (HADR_SSL_LABEL) = HADR SSL Hostname Validation (HADR_SSL_HOST_VAL) = OFF </pre> <p>db2-ec2켜기, 다음 코드를 실행합니다.</p> <pre> db2 get db cfg for sample grep HADR </pre>	

작업	설명	필요한 기술
	<pre> HADR database role = STANDBY HADR local host name (HADR_LOCAL_HOST) = db2-ec2 HADR local service name (HADR_LOCAL_SVC) = 50012 HADR remote host name (HADR_REMOTE_HOST) = db2-server1 HADR remote service name (HADR_REMOTE_SVC) = 50010 HADR instance name of remote server (HADR_REMOTE_INST) = db2inst1 HADR timeout value (HADR_TIMEOUT) = 120 HADR target list (HADR_TARGET_LIST) = db2-server1:50010 db2-server2:50011 HADR log write synchronization mode (HADR_SYNCMODE) = SUPERASYNC HADR spool log data limit (4KB) (HADR_SPOOL_LIMIT) = AUTOMATIC(52000) HADR log replay delay (seconds) (HADR_REPLAY_DELAY) = 0 </pre>	

작업	설명	필요한 기술
	<pre>HADR peer window duration (seconds) (HADR_PEER_WINDOW) = 0 HADR SSL certifica te label (HADR_SSL_LABEL) = HADR SSL Hostname Validation (HADR_SSL_HOST_VAL) = OFF</pre> <p>HADR_LOCAL_HOST , HADR_LOCAL_SVC , HADR_REMOTE_HOST , HADR_REMOTE_SVC , HADR_TARGET_LIST 파라미 터는 기본 HADR 설정 1개와 스탠바이 HADR 설정 2개를 나 타냅니다.</p>	

작업	설명	필요한 기술
<p>Db2 HADR을 중지하고 시작합니다.</p>	<p>HADR_TARGET_LIST 이(가) 이제 세 서버 모두에 설치되었습니다. 각 Db2 서버는 나머지 두 서버를 인식합니다. 새 구성을 활용하려면 HADR을 중지했다가 다시 시작(잠시 중단)하세요.</p> <p>db2-server1 켜기에서 다음 명령을 실행합니다.</p> <pre>db2 stop hadr on db sample db2 deactivate db sample db2 activate db sample</pre> <p>db2-server2 켜기에서 다음 명령을 실행합니다.</p> <pre>db2 deactivate db sample db2 start hadr on db sample as standby SQL1766W The command completed successfully</pre> <p>db2-ec2 켜기에서 다음 명령을 실행합니다.</p> <pre>db2 start hadr on db sample as standby SQL1766W The command completed successfully</pre> <p>db2-server1 켜기에서 다음 명령을 실행합니다.</p>	<p>DBA</p>

작업	설명	필요한 기술
	<pre>db2 start hadr on db sample as primary SQL1766W The command completed successfully</pre> <p>온프레미스에서 Db2와 Amazon EC2 사이의 HADR 연결이 이제 성공적으로 설정되었습니다. Db2 기본 서버 db2-server1 은(는) 트랜잭션 로그 기록을 db2-server2 및 db2-ec2 모두에 실시간으로 스트리밍하기 시작합니다.</p>	

전환 기간 동안 Amazon EC2의 Db2를 기본 버전으로 설정합니다.

작업	설명	필요한 기술
<p>스탠바이 서버에 HADR 락이 없는지 확인하세요.</p>	<p>기본 서버 db2-server1 에서 HADR 상태를 확인합니다. HADR_STATE 이(가) REMOTE_CATCHUP 상태일 때는 놀라지 마세요. HADR_SYNC_MODE 이(가) SUPERASYNC (으)로 설정되어 있으면 정상입니다. PRIMARY_LOG_TIME 및 이 둘이 동기화되어 있음을 STANDBY_REPLAY_LOG_TIME 보여줍니다.</p> <pre>db2pd -hadr -db sample</pre> <p>HADR_ROLE = PRIMARY</p>	<p>DBA</p>

작업	설명	필요한 기술
	<pre> REPLAY_TYPE = PHYSICAL HADR_SYNCMODE = SUPERASYNC STANDBY_ID = 2 LOG_STREAM_ID = 0 HADR_STATE = REMOTE_CATCHUP PRIMARY_LOG_TIME = 10/26/2022 02:11:32. 000000 (1666750292) STANDBY_LOG_TIME = 10/26/2022 02:11:32. 000000 (1666750292) STANDBY_R EPLAY_LOG_TIME = 10/26/2022 02:11:32. 000000 (1666750292) </pre>	

작업	설명	필요한 기술
<p>HADR 테이크오버를 실행하세요.</p>	<p>마이그레이션을 완료하려면 HADR 테이크오버 명령을 실행하여 db2-ec2을(를) 기본 데이터베이스로 만드세요. 다음 명령을 db2pd 사용하여 HADR_ROLE 값을 확인합니다.</p> <pre> db2 TAKEOVER HADR ON DATABASE sample DB20000I The TAKEOVER HADR ON DATABASE command completed successfully. db2pd -hadr -db sample Database Member 0 -- Database SAMPLE -- Active -- Up 0 days 00:03:25 -- Date 2022-10-26-02.46.4 5.048988 HADR_ROLE = PRIMARY REPLAY_TYPE = PHYSICAL </pre> <p>AWS로 마이그레이션을 완료하려면 Amazon EC2의 Db2를 가리키도록 애플리케이션 연결을 지정하세요.</p>	

문제 해결

문제	Solution
<p>방화벽 및 보안상의 이유로 NAT를 사용하는 경우 호스트는 내부 IP 주소 하나와 외부 IP 주소 두 개를 가질 수 있으며, 이로 인해 HADR IP 주소 확인이 실패할 수 있습니다. 이 START HADR ON DATABASE 명령은 다음 메시지를 반환합니다.</p> <pre>HADR_LOCAL_HOST:HADR_LOCAL_SVC (-xx-xx-xx-xx.:50011 (xx.xx.xx .xx:50011)) on remote database is different from HADR_REMOTE_HOST:H ADR_REMOTE_SVC (xx-xx-xx- xx.:50011 (x.x.x.x:50011)) on local database.</pre>	<p>NAT 환경에서 HADR을 지원하려면 내부 및 외부 주소를 모두 사용하여 HADR_LOCAL_HOST 을(를) 구성할 수 있습니다. 예를 들어, Db2 서버에 내부 이름이 host1, 외부 이름이 host1E인 경우 HADR_LOCAL_HOST 은(는) HADR_LOCAL_HOST: "host1 host1E"일 수 있습니다.</p>

관련 리소스

- [서로 다른 운영 체제와 하드웨어 플랫폼 간의 Db2 백업 및 복원 작업](#)
- [Db2 스토리지 액세스 별칭 및 DB2REMOTE 설정](#)
- [Db2 고가용성 재해 복구](#)
- [hadr_syncmode - 피어 상태 구성 파라미터의 로그 쓰기를 위한 HADR 동기화 모드](#)

PowerCLI를 사용하여 HCX 자동화를 통해 VMware VM 마이그레이션

작성자: Giri Nadiminty(AWS), Hassan Adekoya(AWS), Naveen Deshwal

환경: 프로덕션	소스: 온프레미스 또는 클라우드 기반 VMware vCenter 또는 SDDC	대상: AWS의 VMware Cloud
R 유형: 리호스팅	워크로드: 기타 모든 워크로드	기술: 마이그레이션, 하이브리드 클라우드
AWS 서비스: AWS의 VMware Cloud		

요약

이 패턴은 VMware PowerCLI 스크립트로 구동되는 VMware Hybrid Cloud Extension(HCX) 자동화를 사용하여 VMware 온프레미스 가상 머신(VM)을 AWS의 VMware Cloud로 마이그레이션하는 방법을 설명합니다. [PowerCLI](#)는 윈도우에 구축된 명령줄 도구입니다. PowerShell VMware 소프트웨어를 관리할 수 있으며 인프라 및 마이그레이션 작업을 자동화합니다.

이 패턴을 vCenter, 소프트웨어 정의 데이터 센터(SDDCs) 및 클라우드 환경의 모든 조합 간 마이그레이션에 적용할 수 있습니다. 이 패턴에 포함된 PowerCLI 스크립트는 모든 VM 구성 및 예약 작업에 마우스 클릭 대신 자동화를 사용하므로 마이그레이션 작업에 소요되는 시간을 절약하고 인적 오류의 위험을 줄이는 데 도움이 됩니다.

사전 조건 및 제한 사항

사전 조건

- SDDC가 있는 AWS의 VMware Cloud 계정
- 기존 온프레미스 또는 클라우드 기반 vCenter 또는 SDDC
- 소스 및 대상 vCenter 또는 SDDC에 필요한 권한이 있는 사용자 계정
- 소스 및 대상 vCenter 또는 SDDC 간에 구성된 [HCX Network Extension\(HCX-NE\)](#)이 있는 [HCX 사이트 페어링](#)
- 선택한 서버에 설치된 [VMware PowerCLI](#)

제한 사항

- 소스 vCenter에서 크로스 vCenter NSX를 사용하는 경우 PowerCLI 모듈이 작동하지 않습니다. PowerCLI 대신 HCX API를 사용하는 스크립팅 방법(예: Python)을 사용하세요.
- 마이그레이션된 VM에 새 이름이나 IP 주소가 필요한 경우 HCX API를 사용하는 스크립팅 방법(예: Python)을 사용하세요.
- 이 패턴은 필수인 .csv 파일을 채우지 않습니다. VMware vRealize Network Insight(vRNI) 또는 다른 방법을 사용하여 파일을 채울 수 있습니다.

제품 버전

- VMware vSphere 버전 5 이상
- VMware HCX 버전 4.4 이상
- VMware PowerCLI 버전 12.7 이상

아키텍처

소스 기술 스택

- 온프레미스 또는 클라우드 기반 VMware

대상 기술 스택

- AWS의 VMware Cloud

대상 아키텍처

도구

서비스

- [AWS의 VMware Cloud](#)는 온프레미스 VMware vSphere 기반 환경을 AWS 클라우드로 마이그레이션하고 확장할 수 있도록 AWS와 VMware가 공동으로 설계한 서비스입니다.

기타 도구

- [VMware Hybrid Cloud Extension\(HCX\)](#)은 기본 플랫폼을 변경하지 않고 온프레미스 VMware 환경에서 AWS의 VMware Cloud로 워크로드를 마이그레이션하기 위한 유틸리티입니다. 참고: 이전에는 이 제품을 Hybrid Cloud Extension 및 NSX Hybrid Connect라고도 했습니다. 이 패턴은 VM 마이그레이션에 HCX를 사용합니다.
- [VMware PowerCLI](#)는 VMware vSphere 및 vCloud 관리를 자동화하기 위한 명령줄 도구입니다. Windows에서 cmdlet을 사용하여 PowerCLI 명령을 실행할 수 있습니다. PowerShell PowerShell 이 패턴은 PowerCLI를 사용하여 마이그레이션 명령을 실행합니다.

코드

간단한 독립형 스크립트

이 단일 시스템 스크립트를 초기 테스트에 사용하여 구성 옵션이 허용되고 예상대로 작동하는지 확인하는 것이 좋습니다. 지침은 [에픽](#) 섹션을 참조하세요.

```
<# Manual Variables #>
$HcxServer = "[enterValue]"
$SrcNetworkName = "[enterValue]"
$DstNetworkName = "[enterValue]"
$DstComputeName = "[enterValue]"
$DstDSName = "[enterValue]"
$DstFolderName = "[enterValue]"
$vmName = "[enterValue]"

<# Environment Setup #>
Connect-HCXServer -Server $HcxServer
$HcxDstSite = Get-HCXSite -Destination
$HcxSrcSite = Get-HCXSite -Source
$SrcNetwork = Get-HCXNetwork -Name $SrcNetworkName -Type VirtualWire -Site $HcxSrcSite
$DstNetwork = Get-HCXNetwork -Name $DstNetworkName -Type NsxtSegment -Site $HcxDstSite
$DstCompute = Get-HCXContainer -Name $DstComputeName -Site $HcxDstSite
$DstDS = Get-HCXDatastore -Name $DstDSName -Site $HcxDstSite
$DstFolder = Get-HCXContainer -name $DstFolderName -Site $HcxDstSite
$vm = Get-HCXVM -Name $vmName

<# Migration #>
$NetworkMapping = New-HCXNetworkMapping -SourceNetwork $SrcNetwork -DestinationNetwork
$DstNetwork
$NewMigration = New-HCXMigration -VM $vm -MigrationType vMotion -SourceSite $HcxSrcSite
-DestinationSite $HcxDstSite -Folder $DstFolder -TargetComputeContainer $DstCompute
-TargetDatastore $DstDS -NetworkMapping $NetworkMapping -DiskProvisionType Thin
```

```
-UpgradeVMTools $True -RemoveISOs $True -ForcePowerOffVm $True -RetainMac $True -
UpgradeHardware $True -RemoveSnapshots $True
```

모든 기능을 갖춘 .csv 기반 스크립트

테스트가 완료되면 프로덕션 환경에서 다음 스크립트를 사용할 수 있습니다. 지침은 [에픽](#) 섹션을 참조하세요.

```
<# Schedule #>
write-host("Getting Time for Scheduling")
$startTime = [DateTime]::Now.AddDays(12)
$endTime = [DateTime]::Now.AddDays(15)

<# Migration #>
Connect-HCXServer -Server [enterValue]
write-host("Getting Source Site")
$HcxSrcSite = Get-HCXSite
write-host("Getting Target Site")
$HcxDstSite = Get-HCXSite -Destination
$HCXVMS = Import-CSV .\Import_VM_list.csv
ForEach ($HCXVM in $HCXVMS) {
    $DstFolder = Get-HCXContainer $HCXVM.DESTINATION_VM_FOLDER -Site $HcxDstSite
    $DstCompute = Get-HCXContainer $HCXVM.DESTINATION_COMPUTE -Site $HcxDstSite
    $DstDatastore = Get-HCXDatastore $HCXVM.DESTINATION_DATASTORE -Site $HcxDstSite
    $SrcNetwork = Get-HCXNetwork $HCXVM.SOURCE_NETWORK -Type VirtualWire -Site
    $HcxSrcSite
    $DstNetwork = Get-HCXNetwork $HCXVM.DESTINATION_NETWORK -Type NsxtSegment -Site
    $HcxDstSite
    $NetworkMapping = New-HCXNetworkMapping -SourceNetwork $SrcNetwork -
    DestinationNetwork $DstNetwork
    $NewMigration = New-HCXMigration -VM (Get-HCXVM $HCXVM.VM_NAME) -MigrationType
    Bulk -SourceSite $HcxSrcSite -DestinationSite $HcxDstSite -Folder $DstFolder -
    TargetComputeContainer $DstCompute -TargetDatastore $DstDatastore -NetworkMapping
    $NetworkMapping -DiskProvisionType Thin -UpgradeVMTools $True -RemoveISOs $True -
    ForcePowerOffVm $True -RetainMac $True -UpgradeHardware $True -RemoveSnapshots $True -
    ScheduleStartTime $startTime -ScheduleEndTime $endTime
    Start-HCXMigration -Migration $NewMigration -Confirm:$false
}
```

에픽

수동 변수에 대한 정보 수집

작업	설명	필요한 기술
소스 및 대상 vCenter 및 SDDC 서버 이름을 찾습니다.	<p>PowerCLI 스크립트에는 이 에픽에 설명된 변수가 필요합니다. 스크립트를 쉽게 사용할 수 있도록 이 정보를 미리 수집할 수 있습니다.</p> <p>vSphere 콘솔의 HCX 섹션에서 인프라, 사이트 페어링을 선택합니다. 표시된 소스 및 대상 서버 이름을 기록해 둡니다.</p>	클라우드 아키텍트
소스 및 대상 HCX 이름을 찾습니다.	vSphere 콘솔의 HCX 섹션에서 시스템, 관리를 선택합니다. 표시된 소스 및 대상 HCX 이름을 기록해 둡니다.	클라우드 아키텍트
소스 및 대상 네트워크 이름을 찾습니다.	<p>vSphere 콘솔의 HCX 섹션에서 시스템, 네트워크 확장을 선택합니다. 소스 및 대상 네트워크 이름을 기록해 둡니다.</p> <p>참고: 대신 HCX 서버에 연결한 후 PowerCLI Get-HCXNetwork 및 Get-HCXNetwork-Destination 명령을 사용하여 소스 및 대상 네트워크 이름을 가져올 수 있습니다.</p>	클라우드 아키텍트
vSphere 콘솔에서 추가 정보를 수집합니다.	<p>vSphere 콘솔에서 다음 정보를 수집합니다.</p> <ul style="list-style-type: none"> • 마이그레이션하려는 VM의 이름 	클라우드 아키텍트

작업	설명	필요한 기술
	<ul style="list-style-type: none"> 대상 컴퓨팅 환경(클러스터/호스트) 대상 데이터 스토어 대상 VM 폴더 이름 	

마이그레이션 결정

작업	설명	필요한 기술
마이그레이션 옵션을 결정합니다.	<p>다음을 결정합니다.</p> <ul style="list-style-type: none"> MigrationType - HCX 지원 마이그레이션 유형으로는 vMotion, 대량, 콜드, RAV가 있습니다. 선택은 가동 중지 시간 요구 사항, 네트워크 대역폭, 마이그레이션 기간, 워크로드 유형에 따라 달라집니다. 자세한 내용은 Migrating Workloads to VMware Cloud on AWS with Hybrid Cloud Extension (HCX) AWS 블로그 게시물을 참조하세요. DiskProvisionType (Thin, Thick) UpgradeVMTools (\$True, \$False) RemoveISOs (\$True, \$False) ForcePowerOffVm (\$True, \$False) 	클라우드 아키텍트

작업	설명	필요한 기술
	<ul style="list-style-type: none"> • RetainMac (\$True, \$False) • UpgradeHardware (\$True, \$False) • RemoveSnapshots (\$True, \$False) <p>각 옵션에 대한 자세한 내용은 VMware 개발자 설명서를 참조하세요.</p>	

초기 테스트를 위한 간단한 스크립트 실행

작업	설명	필요한 기술
스크립트를 복사합니다.	<p>간단한 버전의 스크립트는 단일 파일로 독립되어 있습니다. 이를 사용하여 단일 시스템의 마이그레이션을 테스트할 수 있습니다.</p> <p>이 패턴의 코드 섹션에서 첫 번째 스크립트를 복사하여 VMware PowerCLI 모듈이 설치된 컴퓨터에 저장합니다. (PowerCLI를 설치하려면 VMware 설명서의 지침을 따르세요.)</p>	클라우드 아키텍트
스크립트 변수를 설정합니다.	스크립트의 Manual Variables 섹션에서 모든 변수를 설정합니다.	클라우드 아키텍트

작업	설명	필요한 기술
마이그레이션 변수를 설정합니다.	스크립트의 Migration 섹션에서 모든 New-HCXMigration 설정을 지정합니다.	클라우드 아키텍트
사이트를 지정합니다.	(선택 사항) 소스 또는 대상에 여러 사이트가 있는 경우 스크립트의 Environment Setup 섹션에서 사이트를 수동으로 지정합니다. 소스 및 대상에 단일 사이트가 있는 경우 스크립트는 자동으로 정보를 조회합니다.	클라우드 아키텍트
스크립트를 실행합니다.	PowerCLI가 설치된 서버의 관리자 PowerShell 창에서 스크립트를 실행하고 메시지가 표시되면 자격 증명을 입력합니다.	클라우드 아키텍트
스크립트를 검증합니다.	VM 마이그레이션이 시작되었는지 확인합니다.	클라우드 아키텍트

모든 기능을 갖춘 스크립트를 실행하여 여러 VM 마이그레이션

작업	설명	필요한 기술
.csv 파일을 생성하고 채웁니다.	컴퓨터에서 Import_VM_list.csv (이)라는 .csv 파일을 생성하고 다음 샘플 콘텐츠로 채웁니다. <pre>VM_NAME, DESTINATION_VM_FOLDER, DESTINATION_COMPUTE, DESTINATION_DATASTORE,</pre>	클라우드 아키텍트

작업	설명	필요한 기술
	<p>SOURCE_NETWORK, DESTINATION_NETWORK [enterValue], [enterValue], [enterValue], [enterValue], [enterValue], [enterValue], [enterValue], [enterValue]</p> <p>.csv 파일의 각 [enterValue] 을(를) 이전에 수집한 정보로 바꿉니다.</p> <p>참고: VMware vRealize Network Insight(vRNI) 또는 다른 방법을 사용하여.csv 파일을 채울 수 있습니다.</p>	
<p>스크립트를 복사합니다.</p>	<p>모든 기능을 갖춘 버전의 스크립트는 외부 .csv 파일의 정보를 사용하여 여러 VM을 자동으로 마이그레이션합니다.</p> <p>이 패턴의 코드 섹션에서 두 번째 스크립트를 복사하여 VMware PowerCLI 모듈이 설치된 컴퓨터의 .csv 파일과 동일한 폴더에 저장합니다.</p>	<p>클라우드 아키텍트</p>

작업	설명	필요한 기술
스크립트를 수정합니다.	<p>스크립트를 편집하여 다음과 같이 변경합니다.</p> <ul style="list-style-type: none"> • 7행: HCX 서버 변수 (Connect-HCXServer)를 설정합니다. • 12행:(선택 사항) .csv 파일 이름을 다르게 설정한 경우 업데이트합니다. • 3~4행:(선택 사항) 일정을 설정합니다. • 20행:(선택 사항) Migration 섹션에서 New-HCXMigration 설정을 지정합니다. • 9행 및 11행:(선택 사항) 소스 또는 대상에 여러 사이트가 포함된 경우 원하는 사이트를 수동으로 지정합니다. 	클라우드 아키텍트
스크립트를 실행합니다.	PowerCLI가 설치된 서버의 관리자 PowerShell 창에서 스크립트를 실행하고 메시지가 표시되면 자격 증명을 입력합니다.	클라우드 아키텍트
스크립트를 검증합니다.	VM 마이그레이션이 시작되었는지 확인합니다.	클라우드 아키텍트

문제 해결

문제	Solution
<p>스크립트가 실패하고 다음과 같은 오류 메시지가 표시됩니다.</p> <p>“All source networks are not mapped to target!”</p>	<p>소스 vCenter에서 크로스 vCenter NSX를 사용하는 경우 PowerCLI 모듈이 작동하지 않습니다. PowerCLI 대신 HCX API를 사용하는 스크립팅 방법(예: Python)을 사용하세요. 이는 PowerCLI 스크립트의 알려진 제한 사항입니다.</p>
<p>스크립트가 실패하고 다음과 같은 오류 메시지가 표시됩니다.</p> <p>“Connect-HCXServer Error: Unauthorized”</p>	<p>입력한 보안 인증 정보가 필요한 권한을 제공하지 않습니다.</p>

관련 리소스

- [Migrating Workloads to AWS의 VMware Cloud with Hybrid Cloud Extension \(HCX\)](#)(AWS 블로그 게시물)
- [AWS 클라우드로 VMware 애플리케이션 및 워크로드를 재배포하기 위한 마이그레이션 방식 선택](#)(AWS 권장 가이드)
- [VMware HCX를 사용하여 VMware SDDC를 AWS의 VMware Cloud로 마이그레이션](#)(AWS 권장 가이드)
- [Getting Started with the HCX Module](#)(VMware 블로그 게시물)

F5 BIG-IP 워크로드를 AWS 클라우드의 F5 BIG-IP VE로 마이그레이션

작성자: 월 바우어(AWS)

소스: F5 BIG-IP TMOS 13.1 이상	대상: AWS의 F5 BIG-IP VE	R 유형: 리호스팅
환경: 프로덕션	기술: 마이그레이션, 보안, 자격 증명, 규정 준수, 네트워킹	워크로드: 기타 모든 워크로드

AWS 서비스: 아마존 EC2, 아마존 VPC, AWS Transit Gateway, 아마존, CloudFront 아마존, AWS 글로벌 CloudWatch 액셀러레이터, AWS CloudFormation

요약

조직은 Amazon Web Services(AWS) 클라우드로 마이그레이션하여 민첩성과 복원력을 향상하길 원합니다. [F5 BIG-IP](#) 보안 및 트래픽 관리 솔루션을 AWS 클라우드로 마이그레이션한 후에는 엔터프라이즈 아키텍처 전반에서 민첩성과 고가치 운영 모델 채택에 집중할 수 있습니다.

이 패턴은 F5 BIG-IP 워크로드를 AWS 클라우드의 [F5 BIG-IP Virtual Edition\(VE\)](#) 워크로드로 마이그레이션하는 방법을 설명합니다. 기존 환경을 재호스팅하고 서비스 검색 및 API 통합과 같은 리플랫폼의 측면을 배포함으로써 워크로드를 마이그레이션합니다. [AWS CloudFormation 템플릿](#)은 워크로드를 AWS 클라우드로 마이그레이션하는 속도를 높입니다.

이 패턴은 F5 보안 및 트래픽 관리 솔루션을 마이그레이션하는 기술 엔지니어링 및 아키텍처 팀을 위한 것으로, AWS Precriptive Guidetion 웹사이트에 있는 [AWS 클라우드의 F5 BIG-IP에서 F5 BIG-IP VE로 마이그레이션](#) 가이드와 함께 제공됩니다.

사전 조건 및 제한 사항

사전 조건

- 기존 온프레미스 F5 BIG-IP 워크로드.

- BIG-IP VE 버전용 기존 F5 라이선스.
- 활성 상태의 AWS 계정
- NAT 게이트웨이 또는 엘라스틱 IP 주소를 통한 송신을 통해 구성되고 아마존 심플 스토리지 서비스 (Amazon S3), Amazon Elastic Compute Cloud (아마존 EC2), AWS 보안 토큰 서비스 (AWS STS), 아마존 등의 엔드포인트에 액세스할 수 있도록 구성된 기존 가상 사설 클라우드 (VPC). CloudWatch 또한 [확장 가능한 모듈식 VPC 아키텍처 Quick Start](#)를 배포를 위한 빌딩 블록으로 수정할 수 있습니다.
- 사용자의 요구 사항에 따라 하나 또는 두 개의 기존 가용 영역이 있습니다.
- 각 가용 영역에는 3개의 기존 프라이빗 서브넷이 있습니다.
- AWS CloudFormation 템플릿, [F5 GitHub 리포지토리에서 사용 가능](#).

마이그레이션 중에 사용자의 요구 사항에 따라 다음을 사용할 수도 있습니다.

- 엘라스틱 IP 주소 매핑, 보조 IP 매핑 및 라우팅 테이블 변경을 관리하기 위한 [F5 Cloud Failover Extension](#)입니다.
- 여러 가용 영역을 사용하는 경우, F5 Cloud Failover Extension을 사용하여 가상 서버에 대한 엘라스틱 IP 매핑을 처리해야 합니다.
- 구성을 관리하려면 [F5 Application Services 3\(AS3\)](#), [F5 Application Services Templates \(FAST\)](#) 또는 다른 코드형 인프라(IaC)를 사용하는 것을 고려해야 합니다. IaC 모델에서 구성을 준비하고 코드 리포지토리를 사용하면 마이그레이션과 지속적인 관리 작업에 도움이 됩니다.

전문성

- 이 패턴을 사용하려면 하나 이상의 VPC를 기존 데이터 센터에 연결하는 방법에 익숙해야 합니다. 이에 대한 자세한 내용은 Amazon VPC 설명서의 [네트워크와 Amazon VPC 간 연결 옵션](#)을 참조하세요.
- 또한 [Traffic Management Operating System\(TMOS\)](#), [Local Traffic Manager\(LTM\)](#), [Global Traffic Manager\(GTM\)](#), [Access Policy Manager\(APM\)](#), [Application Security Manager\(ASM\)](#), [Advanced Firewall Manager\(AFM\)](#), [BIG-IQ](#)를 포함한 F5 제품 및 모듈에 대한 지식이 필요합니다.

제품 버전

- 이 패턴은 F5 BIG-IP [버전 12.1](#) 이상을 지원하지만 F5 BIG-IP [버전 13.1](#) 이상을 사용하는 것이 좋습니다.

아키텍처

소스 기술 스택

- F5 BIG-IP 워크로드

대상 기술 스택

- 아마존 CloudFront
- 아마존 CloudWatch
- Amazon EC2
- Amazon S3
- Amazon VPC
- AWS Global Accelerator
- AWS STS
- AWS Transit Gateway
- F5 BIG-IP VE

대상 아키텍처

도구

- [AWS는 AWS CloudFormation](#) 리소스를 설정하고, 빠르고 일관되게 프로비저닝하고, AWS 계정 및 지역 전반의 수명 주기 전반에 걸쳐 리소스를 관리할 수 있도록 지원합니다.
- [Amazon은](#) 전 세계 데이터 센터 네트워크를 통해 웹 콘텐츠를 전송함으로써 웹 콘텐츠 배포 CloudFront 속도를 높여 지연 시간을 줄이고 성능을 개선합니다.
- [Amazon은](#) AWS 리소스의 지표와 AWS에서 실행하는 애플리케이션을 실시간으로 모니터링할 수 있도록 CloudWatch 도와줍니다.
- [Amazon Elastic Compute Cloud\(Amazon EC2\)](#)는 AWS 클라우드에서 확장 가능한 컴퓨팅 용량을 제공합니다. 필요한 만큼 가상 서버를 시작하고 빠르게 스케일 업하거나 스케일 다운할 수 있습니다.
- [AWS Identity and Access Management\(IAM\)](#)를 사용하면 사용자에 대해 인증 및 권한 부여를 제어함으로써 AWS 리소스에 대한 액세스를 안전하게 관리할 수 있습니다.

- [Amazon Simple Storage Service\(S3\)](#)는 원하는 양의 데이터를 저장, 보호 및 검색하는 데 도움이 되는 클라우드 기반 객체 스토리지 서비스입니다.
- [AWS Security Token Service\(AWS STS\)](#)를 사용하면 사용자를 위한 제한된 권한의 임시 보안 인증 정보를 요청할 수 있습니다.
- [AWS Transit Gateway](#)는 Virtual Private Cloud(VPC)와 온프레미스 네트워크를 연결하는 중앙 허브입니다.
- [Amazon Virtual Private Cloud\(VPC\)](#)를 이용하면 사용자가 정의한 가상 네트워크로 AWS 리소스를 시작할 수 있습니다. 이 가상 네트워크는 사용자의 자체 데이터 센터에서 운영하는 기존 네트워크와 유사하며 AWS의 확장 가능한 인프라를 사용한다는 이점이 있습니다.

에픽

검색 및 평가

작업	설명	필요한 기술
F5 BIG-IP의 성능을 평가합니다.	가상 서버에 있는 애플리케이션의 성능 지표와 마이그레이션할 시스템의 지표를 수집하고 기록합니다. 이렇게 하면 비용 최적화 개선을 위해 대상 AWS 인프라의 크기를 올바르게 조정하는 데 도움이 됩니다.	F5 아키텍트, 엔지니어 및 네트워크 아키텍트, 엔지니어
F5 BIG-IP 운영 체제 및 구성을 평가합니다.	마이그레이션할 객체와 VLAN과 같은 네트워크 구조를 유지 관리해야 하는지 평가합니다.	F5 아키텍트, 엔지니어
F5 라이선스 옵션을 평가합니다.	어떤 라이선스와 소비 모델이 필요한지 평가합니다. 이 평가는 F5 BIG-IP 운영 체제 및 구성에 대한 평가를 기반으로 해야 합니다.	F5 아키텍트, 엔지니어
퍼블릭 애플리케이션을 평가합니다.	퍼블릭 IP 주소가 필요한 애플리케이션을 결정합니다. 성능 및 서비스 수준에 관한 계약	F5 아키텍트, 클라우드 아키텍트, 네트워크 아키텍트, 엔지니어, 앱 팀

작업	설명	필요한 기술
	(SLA) 요구 사항을 충족하도록 해당 애플리케이션을 필요한 인스턴스 및 클러스터에 맞게 조정합니다.	
내부 애플리케이션을 평가합니다.	내부 사용자가 사용할 애플리케이션을 평가합니다. 조직 내 내부 사용자의 위치와 해당 환경이 AWS 클라우드에 연결되는 방식을 알아야 합니다. 또한 그러한 애플리케이션이 도메인 이름 시스템(DNS)을 기본 도메인의 일부로 사용할 수 있는지 확인해야 합니다.	F5 아키텍트, 클라우드 아키텍트, 네트워크 아키텍트, 엔지니어, 앱 팀
AMI를 마무리합니다.	모든 F5 BIG-IP 버전이 Amazon Machine Images(AMI)로 생성되지 않습니다. 필요한 특정 QFE(quick-fix engineering) 버전이 있는 경우 F5 BIG-IP 이미지 생성기 도구를 사용할 수 있습니다. 이 도구에 대한 자세한 내용은 “관련 리소스” 섹션을 참조하세요.	F5 아키텍트, 클라우드 아키텍트, 엔지니어
인스턴스 유형과 아키텍처를 마무리합니다.	인스턴스 유형, VPC 아키텍처, 상호 연결된 아키텍처를 결정합니다.	F5 아키텍트, 클라우드 아키텍트, 네트워크 아키텍트, 엔지니어

보안 및 규정 준수 관련 활동 완료

작업	설명	필요한 기술
기존 F5 보안 정책을 문서화합니다.	기존 F5 보안 정책을 수집하고 문서화합니다. 보안 코드 리포	F5 아키텍트, 엔지니어

작업	설명	필요한 기술
	지토리에 사본을 생성해야 합니다.	
AMI를 암호화합니다.	(선택 사항) 조직에서 저장 데이터의 암호화를 요구할 수 있습니다. 사용자 지정 기존 보유 라이선스 사용 (BYOL) 이미지 생성에 대한 자세한 내용은 “관련 리소스” 섹션을 참조하세요.	F5 아키텍트, 엔지니어 클라우드 아키텍트, 엔지니어
디바이스를 강화합니다.	이렇게 하면 잠재적 취약성으로부터 보호하는 데 도움이 됩니다.	F5 아키텍트, 엔지니어

새 AWS 환경 구성

작업	설명	필요한 기술
엣지 및 보안 계정을 생성합니다.	AWS Management Console에 로그인하여 엣지 및 보안 서비스를 제공하고 운영할 AWS 계정을 생성합니다. 이러한 계정은 공유 서비스 및 애플리케이션용 VPC를 운영하는 계정과 다를 수 있습니다. 이 단계는 랜딩 존의 일부로 완료할 수 있습니다.	클라우드 아키텍트, 엔지니어
엣지 및 보안 VPC를 배포합니다.	엣지 및 보안 서비스를 제공하는 데 필요한 VPC를 설정하고 구성합니다.	클라우드 아키텍트, 엔지니어
소스 데이터 센터에 연결합니다.	F5 BIG-IP 워크로드를 호스팅하는 소스 데이터 센터에 연결합니다.	클라우드 아키텍트, 네트워크 아키텍트, 엔지니어

작업	설명	필요한 기술
VPC 연결을 배포합니다.	엣지 및 보안 서비스 VPC를 애플리케이션 VPC에 연결합니다.	네트워크 아키텍트, 엔지니어
인스턴스를 배포합니다.	“관련 리소스” 섹션의 AWS CloudFormation 템플릿을 사용하여 인스턴스를 배포합니다.	F5 아키텍트, 엔지니어
인스턴스 장애 조치를 테스트하고 구성합니다.	AWS Advanced HA IApp 템플릿 또는 F5 Cloud Failover Extension 이 구성되어 올바르게 작동하고 있는지 확인합니다.	F5 아키텍트, 엔지니어

네트워킹 구성

작업	설명	필요한 기술
VPC 토폴로지를 준비합니다.	Amazon VPC 콘솔을 열고 VPC에 F5 BIG-IP VE 배포에 필요한 모든 서브넷과 보호 기능이 있는지 확인합니다.	네트워크 아키텍트, F5 아키텍트, 클라우드 아키텍트, 엔지니어
VPC 엔드포인트를 준비합니다.	F5 BIG-IP 워크로드가 TMM 인터페이스의 NAT 게이트웨이 또는 엘라스틱 IP 주소에 액세스할 수 없는 경우, Amazon EC2, Amazon S3 및 AWS STS를 위한 VPC 엔드포인트를 준비합니다.	클라우드 아키텍트, 엔지니어

데이터 마이그레이션

작업	설명	필요한 기술
구성을 마이그레이션합니다.	F5 BIG-IP 구성을 AWS 클라우드의 F5 BIG-IP VE로 마이그레이션합니다.	F5 아키텍트, 엔지니어
보조 IP를 연결합니다.	가상 서버 IP 주소는 인스턴스에 할당된 보조 IP 주소와 관계가 있습니다. 보조 IP 주소를 할당하고 “재매핑/재할당 허용”이 선택되어 있는지 확인합니다.	F5 아키텍트, 엔지니어

구성 테스트

작업	설명	필요한 기술
가상 서버 구성을 검증합니다.	가상 서버를 테스트합니다.	F5 아키텍트, 앱 팀

운영 마무리

작업	설명	필요한 기술
백업 전략을 생성합니다.	전체 스냅샷을 생성하려면 시스템을 종료해야 합니다. 자세한 내용은 “관련 리소스” 섹션의 “F5 BIG-IP 가상 시스템 업데이트”를 참조하세요.	F5 아키텍트, 클라우드 아키텍트, 엔지니어
클러스터 장애 조치 런북을 생성합니다.	장애 조치 런북 프로세스가 완료되었는지 확인합니다.	F5 아키텍트, 엔지니어
로깅을 설정 및 검증합니다.	필요한 대상으로 로그를 전송하도록 F5 텔레메트리 스트리밍을 구성합니다.	F5 아키텍트, 엔지니어

전환 완료

작업	설명	필요한 기술
새 배포로 전환합니다.		F5 아키텍트, 클라우드 아키텍트, 네트워크 아키텍트, 엔지니어, AppTeams

관련 리소스

마이그레이션 가이드

- [AWS 클라우드에서 F5 BIG-IP에서 F5 BIG-IP VE로 마이그레이션하기](#)

F5 리소스

- [F5 CloudFormation GitHub 리포지토리의 AWS 템플릿](#)
- [AWS Marketplace의 F5](#)
- [F5 BIG-IP VE 개요](#)
- [Quickstart 예제 - WAF\(LTM+ASM\)를 사용하는 BIG-IP Virtual Edition](#)
- [AWS 기반 F5 애플리케이션 서비스: 개요\(동영상\)](#)
- [F5 Application Services 3 Extension 사용 설명서](#)
- [F5 클라우드 설명서](#)
- [F5 iControl REST 위키](#)
- [F5 단일 구성 파일 개요\(11.x - 15.x\)](#)
- [F5 토폴로지 랩](#)
- [F5 백서](#)
- [F5 BIG-IP 이미지 생성 도구](#)
- [F5 BIG-IP VE 가상 머신 업데이트](#)
- [UCS 아카이브 “플랫폼 마이그레이션” 옵션 개요](#)

이진법을 사용하여 온프레미스 Go 웹 애플리케이션을 AWS Elastic Beanstalk로 마이그레이션

작성자: 수하스 바사바라즈(AWS) 및 슈마즈 무크타르 카지(AWS)

환경: PoC 또는 파일럿	소스: 애플리케이션	대상: Elastic Beanstalk
R 유형: 리호스팅	기술: 마이그레이션, 웹 및 모바일 앱	AWS 서비스: AWS Elastic Beanstalk

요약

이 패턴은 온프레미스 Go 웹 애플리케이션을 AWS Elastic Beanstalk로 마이그레이션하는 방법을 설명합니다. Elastic Beanstalk는 애플리케이션 마이그레이션 후 소스 번들용 이진수를 빌드하여 Amazon Elastic Compute Cloud(Amazon EC2) 인스턴스로 배포합니다.

리호스팅 마이그레이션 전략으로서 이 패턴의 접근 방식은 빠르고 코드 변경이 필요하지 않으므로 테스트 및 마이그레이션 시간이 단축됩니다.

사전 조건 및 제한 사항

사전 조건

- 활성 상태의 AWS 계정.
- 온프레미스 Go 웹 애플리케이션.
- Go 애플리케이션의 소스 코드가 들어 있는 GitHub 리포지토리입니다. 사용하지 GitHub 않는 경우 다른 방법으로 [Elastic Beanstalk용 애플리케이션 소스 번들을 생성할 수 있습니다](#).

제품 버전

- Elastic Beanstalk에서 지원하는 가장 최신 Go 버전입니다. 자세한 내용은 [Elastic Beanstalk 설명서](#)를 참조하세요.

아키텍처

소스 기술 스택

- 온프레미스 Go 웹 애플리케이션.

대상 기술 스택

- AWS Elastic Beanstalk
- 아마존 CloudWatch

대상 아키텍처

도구

- [AWS Elastic Beanstalk](#)를 사용하면 애플리케이션을 실행하는 인프라에 대해 자세히 알지 못해도 AWS 클라우드에서 애플리케이션을 신속하게 배포하고 관리할 수 있습니다. Elastic Beanstalk를 사용하면 선택 또는 제어에 대한 제한 없이 관리 복잡성을 줄일 수 있습니다.
- [GitHub](#) 오픈 소스 분산 버전 제어 시스템입니다.

에픽

Go 웹 애플리케이션 소스 번들 .zip 파일 생성

작업	설명	필요한 기술
Go 애플리케이션의 소스 번들을 생성하세요.	Go 애플리케이션의 소스 코드가 들어 있는 GitHub 리포지토리를 열고 소스 번들을 준비합니다. 소스 번들에는 Go 애플리케이션의 기본 패키지를 호스팅하는 루트 디렉터리의 <code>application.go</code> 소스 파일이 포함되어 있습니다. 사용하지 GitHub 않는 경우 이 패턴 앞부분의 사전 요구 사항 섹션에서 애플리케이션 소스 번들을 생성하는 다른 방법을 확인하세요.	시스템 관리자, 애플리케이션 개발자

작업	설명	필요한 기술
구성 파일을 생성합니다.	소스 번들에 .ebextensions 폴더를 생성한 다음 이 폴더 안에 options.config 파일을 생성합니다. 자세한 내용은 Elastic Beanstalk 설명서 를 참조하세요.	시스템 관리자, 애플리케이션 개발자
소스 번들 .zip 파일을 생성합니다.	<p>다음 명령을 실행합니다.</p> <pre>git archive -o ../godemo app.zip HEAD</pre> <p>그러면 소스 번들 .zip 파일이 생성됩니다. .zip 파일을 로컬 파일로 다운로드하여 저장합니다.</p> <p>중요: .zip 파일은 512MB를 초과할 수 없으며 상위 폴더 또는 최상위 디렉터리를 포함할 수 없습니다.</p>	시스템 관리자, 애플리케이션 개발자

Go 웹 애플리케이션을 Elastic Beanstalk로 마이그레이션

작업	설명	필요한 기술
Elastic Beanstalk 애플리케이션을 선택합니다.	<ol style="list-style-type: none"> 1. AWS Management Console에 로그인하고 Elastic Beanstalk console을 엽니다. 2. 리전 목록에서 해당 AWS 리전을 선택합니다. 3. 탐색 창에서 애플리케이션을 선택한 기존 Elastic 	시스템 관리자, 애플리케이션 개발자

작업	설명	필요한 기술
	<p>Beanstalk 애플리케이션을 선택하거나 애플리케이션을 생성합니다.</p> <p>Elastic Beanstalk 애플리케이션 생성에 관한 지침은 Elastic Beanstalk 설명서를 참조하세요.</p>	
Elastic Beanstalk 웹 서버 환경을 시작합니다.	<ol style="list-style-type: none"> 1. 애플리케이션 개요 페이지에서 새 환경 생성을 선택한 다음 웹 서버 환경을 선택합니다. 2. 환경 이름 및 도메인 이름 필드를 작성합니다. 3. 플랫폼 버전을 선택하고 Go를 플랫폼으로 선택합니다. 	시스템 관리자, 애플리케이션 개발자
소스 번들 .zip 파일을 Elastic Beanstalk에 업로드합니다.	<ol style="list-style-type: none"> 1. 애플리케이션 코드에서 코드 업로드를 선택한 다음 로컬 파일을 선택합니다. 2. 소스 번들이 들어 있는 .zip 파일을 선택합니다. 3. 버전 라벨에서 파일에 고유한 이름을 지정한 다음 환경 생성을 선택합니다. 	시스템 관리자, 애플리케이션 개발자

작업	설명	필요한 기술
배포된 Go 웹 애플리케이션을 테스트합니다.	Elastic Beanstalk 애플리케이션의 개요 페이지로 리디렉션됩니다. 개요 상단의 환경 ID 옆에서 elasticbeanstalk.com (으)로 끝나는 URL을 선택하여 애플리케이션으로 이동합니다. 애플리케이션은 구성 파일에서 이 이름을 환경 변수로 사용하고 웹 페이지에 표시해야 합니다.	시스템 관리자, 애플리케이션 개발자

문제 해결

문제	Solution
Application Load Balancer를 통해 애플리케이션에 액세스할 수 없습니다.	Elastic Beanstalk 애플리케이션을 포함하는 대상 그룹을 확인합니다. 비정상인 경우 Elastic Beanstalk 인스턴스에 로그인하고 nginx.conf 파일 구성을 확인하여 올바른 상태의 URL로 라우팅되는지 확인하세요. 대상 그룹 상태 확인 URL을 변경해야 할 수도 있습니다.

관련 리소스

- [Elastic Beanstalk 지원 플랫폼 버전으로 이동](#)
- [Elastic Beanstalk에서 구성 파일 사용](#)
- [Elastic Beanstalk에서 예제 애플리케이션 생성](#)

SFTP용 AWS Transfer를 사용하여 온프레미스 SFTP 서버를 AWS로 마이그레이션

작성자: Akash Kumar(AWS)

환경: 프로덕션	소스: 스토리지	대상: Amazon S3
R 유형: 리호스팅	기술: 마이그레이션, 스토리지 및 백업, 웹 및 모바일 앱	AWS 서비스: 아마존 S3, AWS Transfer Family, 아마존 CloudWatch 로그

요약

이 패턴은 AWS Transfer for SFTP 서비스를 사용하여 Secure Shell(SSH) File Transfer Protocol(SFTP)을 사용하는 온프레미스 파일 전송 솔루션을 Amazon Web Services(AWS) 클라우드로 마이그레이션하는 방법을 설명합니다. 사용자는 일반적으로 도메인 이름이나 고정 IP를 통해 SFTP 서버에 연결합니다. 이 패턴은 두 경우 모두에 적용됩니다.

SFTP용 AWS Transfer는 AWS Transfer Family의 일원입니다. 그것은은 SFTP에 대한 스토리지 서비스에 들어오고 나가도록 파일을 전송할 수 있는 보안 전송 서비스입니다. Amazon Simple Storage Service(S3) 나 Amazon Elastic File System(Amazon EFS)과 함께 SFTP용 AWS Transfer를 사용할 수 있습니다. 이 패턴은 저장용 Amazon S3을 사용합니다.

사전 조건 및 제한 사항

사전 조건

- 활성 상태의 AWS 계정.
- 기존 SFTP 도메인 이름 또는 고정된 SFTP IP.

제한 사항

- 한 번의 요청으로 전송할 수 있는 최대 객체는 현재 5GiB입니다. 100MiB보다 큰 파일의 경우 [Amazon S3 멀티파트 업로드](#) 사용을 고려해 보십시오.

아키텍처

소스 기술 스택

- 온프레미스 플랫폼 파일 또는 데이터베이스 덤프 파일.

대상 기술 스택

- AWS Transfer for SFTP
- Amazon S3
- Amazon Virtual Private Cloud(VPC)
- AWS Identity 및 Access Management(IAM) 역할 및 정책
- 탄력적 IP 주소
- 보안 그룹
- 아마존 CloudWatch 로그 (선택 사항)

대상 아키텍처

자동화 및 규모 조정

이 패턴의 대상 아키텍처를 자동화하려면 첨부된 AWS CloudFormation 템플릿을 사용하십시오.

- `amazon-vpc-subnets.yml`은 퍼블릭 서브넷 2개와 프라이빗 서브넷 2개를 사용해 Virtual Private Cloud(VPC)를 프로비저닝합니다.
- `amazon-sftp-server.yml`은 SFTP 서버를 프로비저닝합니다.
- `amazon-sftp-customer.yml`은 사용자를 추가합니다.

도구

서비스

- [Amazon CloudWatch Logs](#)를 사용하면 모든 시스템, 애플리케이션 및 AWS 서비스의 로그를 중앙 집중화하여 로그를 모니터링하고 안전하게 보관할 수 있습니다.
- [AWS Identity and Access Management\(IAM\)](#)를 사용하면 사용자에 대해 인증 및 권한 부여를 제어함으로써 AWS 리소스에 대한 액세스를 안전하게 관리할 수 있습니다.

- [Amazon Simple Storage Service\(S3\)](#)는 원하는 양의 데이터를 저장, 보호 및 검색하는 데 도움이 되는 클라우드 기반 객체 스토리지 서비스입니다. 이 패턴은 Amazon S3를 File Transfer를 위한 스토리지 시스템으로 사용합니다.
- [SFTP용 AWS Transfer](#)를 사용하면 SFTP 프로토콜을 통해 AWS 스토리지 서비스에서 파일을 주고 받을 수 있습니다.
- [Amazon Virtual Private Cloud\(VPC\)](#)를 이용하면 사용자가 정의한 가상 네트워크로 AWS 리소스를 시작할 수 있습니다. 이 가상 네트워크는 사용자의 자체 데이터 센터에서 운영하는 기존 네트워크와 유사하며 AWS의 확장 가능한 인프라를 사용한다는 이점이 있습니다.

에픽

VPC 생성

작업	설명	필요한 기술
서브넷이 있는 VPC를 생성합니다.	<p>https://console.aws.amazon.com/vpc/에서 Amazon VPC 콘솔을 여세요. 퍼블릭 서브넷이 두 개 있는 Virtual Private Cloud(VPC)를 생성하십시오. (두 번째 서브넷은고가용성을 제공합니다.)</p> <p>- 또는 -</p> <p>첨부된 CloudFormation 템플릿을 CloudFormation 콘솔에 배포하여 이 에픽의 작업을 자동화할 수 있습니다. amazon-vpc-subnets.yml</p>	개발자, 시스템 관리자
인터넷 게이트웨이를 추가하십시오.	인터넷 게이트웨이를 프로비저닝하고 VPC에 연결합니다.	개발자, 시스템 관리자
기존 IP를 마이그레이션하십시오.	탄력적 IP 주소에 기존 IP를 연결합니다. 주소 풀에서 탄력적	개발자, 시스템 관리자

작업	설명	필요한 기술
	IP 주소를 생성하여 사용하십시오.	

SFTP 서버 프로비저닝

작업	설명	필요한 기술
SFTP 서버를 생성하십시오.	<p>https://console.aws.amazon.com/transfer/에서 AWS Transfer Family 콘솔을 엽니다. AWS Transfer Family 설명서의 서버용 인터넷 연결 엔드포인트 생성의 지침에 따라 인터넷 연결 엔드포인트가 있는 SFTP 서버를 생성하십시오. 엔드포인트 유형에서 VPC 호스트를 선택합니다. 액세스에서 인터넷 연결을 선택합니다. VPC에서 이전 단계에서 방금 생성한 VPC를 선택합니다.</p> <p>- 또는 -</p> <p>첨부된 CloudFormation 템플릿을 CloudFormation 콘솔에 배포하여 이 에픽의 작업을 자동화할 수 있습니다. <code>amazon-sftp-server.yml</code></p>	개발자, 시스템 관리자
도메인 이름을 마이그레이션합니다.	기존 도메인 이름을 사용자 지정 호스트 이름에 연결합니다. 새 도메인 이름을 사용하는 경우 Amazon Route 53 DNS 별칭을 사용하십시오. 기존 도메인 이름의 경우 기타 DNS를 선택	개발자, 시스템 관리자

작업	설명	필요한 기술
	택합니다. 자세한 내용은 AWS Transfer Family 설명서의 사용자 지정 호스트 이름 사용 을 참조하십시오.	
CloudWatch 로깅 역할을 추가하세요.	(선택 사항) CloudWatch 로깅을 활성화하려면 CloudWatch Logs API 작업 <code>logs:CreateLogGroup</code> , <code>logs:CreateLogStream</code> , <code>logs:DescribeLogStreams</code> , 및 <code>logs:PutLogEvents</code> 를 사용하여 Transfer 역할을 생성합니다. 자세한 내용은 AWS Transfer Family 설명서의 로그 활동을 참조 하십시오. CloudWatch	개발자, 시스템 관리자
저장하고 제출하십시오.	저장을 선택합니다. 작업에서 시작을 선택하고 SFTP 서버가 온라인 상태로 생성될 때까지 기다립니다.	개발자, 시스템 관리자

탄력적 IP 주소를 SFTP 서버에 매핑

작업	설명	필요한 기술
설정을 수정할 수 있도록 서버를 중지하십시오.	AWS Transfer Family 콘솔 에서 서버를 선택한 다음, 생성한 SFTP 서버를 선택합니다. 작업에서 중지를 선택합니다. 서버가 오프라인 상태이면 편	개발자, 시스템 관리자

작업	설명	필요한 기술
	집을 선택하여 설정을 수정합니다.	
가용 영역 및 서브넷을 선택합니다.	가용 영역 섹션에서 VPC용 가용 영역 및 서브넷을 선택합니다.	개발자, 시스템 관리자
탄력적 IP 주소를 추가하십시오.	IPv4 주소의 경우 각 서브넷의 탄력적 IP 주소를 선택한 다음 저장을 선택합니다.	개발자, 시스템 관리자

사용자 추가

작업	설명	필요한 기술
S3 버킷에 액세스하기 위한 IAM 역할을 생성하십시오.	<p>Transfer IAM 역할을 생성하고 <code>s3:ListBucket</code> , <code>s3:GetBucketLocation</code> 및 <code>s3:PutObject</code> 를 리소스로 S3 버킷 이름과 함께 추가합니다. 자세한 내용은 AWS Transfer Family 설명서의 IAM 역할 및 정책 생성을 참조하십시오.</p> <p>- 또는 -</p> <p>첨부된 CloudFormation 템플릿을 CloudFormation 콘솔에 배포하여 이 에픽의 작업을 자동화할 수 있습니다. <code>amazon-sftp-customer.yml</code></p>	개발자, 시스템 관리자
S3 버킷을 생성합니다.	애플리케이션을 위한 S3 버킷을 생성합니다.	개발자, 시스템 관리자

작업	설명	필요한 기술
선택형 폴더를 만듭니다.	(선택 사항) 특정 Amazon S3 폴더에 사용자용 파일을 별도로 저장하려는 경우 폴더를 적절히 추가합니다.	개발자, 시스템 관리자
SSH 퍼블릭 키를 생성합니다.	SSH 키 쌍을 생성하려면 AWS Transfer Family 설명서의 SSH 키 생성 을 참조하십시오.	개발자, 시스템 관리자
사용자를 추가합니다.	AWS Transfer Family 콘솔 에서 서버를 선택한 다음, 생성한 SFTP 서버를 고른 다음 사용자 추가를 선택하십시오. 홈 디렉터리의 경우 이전에 생성한 S3 버킷을 선택합니다. SSH 퍼블릭 키에는 SSH 키 쌍의 SSH 퍼블릭 키 부분을 지정합니다. SFTP 서버의 사용자를 추가한 다음 추가를 선택합니다.	개발자, 시스템 관리자

SFTP 서버 테스트

작업	설명	필요한 기술
보안 그룹을 업데이트하십시오.	SFTP 서버의 보안 그룹 섹션에서 테스트 머신의 IP를 추가하여 SFTP 액세스 권한을 얻으십시오.	개발자
SFTP 클라이언트 유틸리티를 사용하여 서버를 테스트하십시오.	SFTP 클라이언트 유틸리티를 사용하여 File Transfer를 테스트합니다. 클라이언트 및 지침 목록은 AWS Transfer Family	개발자

작업	설명	필요한 기술
	설명서에서 클라이언트를 사용한 파일 전송 을 참조하십시오.	

관련 리소스

- [AWS Transfer Family 사용 설명서](#)
- [Amazon S3 사용 설명서](#)
- 자세한 내용은 Amazon EC2 설명서의 [탄력적 IP 주소](#)를 참조하십시오.

첨부

이 문서와 관련된 추가 콘텐츠에 액세스하려면 [attachment.zip](#) 파일의 압축을 풉니다.

AWS 애플리케이션 마이그레이션 서비스를 사용하여 온프레미스 VM을 Amazon EC2로 마이그레이션하기

작성자: Thanh Nguyen (AWS)

환경: 프로덕션	소스: 온프레미스 가상 머신	대상: Amazon ECS
R 유형: 리호스팅	기술: 마이그레이션	AWS 서비스: AWS Application Migration Service, Amazon EC2, Amazon EBS

요약

애플리케이션 마이그레이션과 관련하여 조직은 다양한 접근 방식을 사용하여 애플리케이션 서버를 온프레미스 환경에서 Amazon Web Services(AWS) Cloud로 리호스팅(리프트 앤 시프트) 할 수 있습니다. 한 가지 방법은 새 Amazon Elastic Compute Cloud (Amazon EC2) 인스턴스를 프로비저닝한 후 애플리케이션을 처음부터 설치하고 구성하는 것입니다. 또 다른 접근 방식은 타사 또는 AWS 네이티브 마이그레이션 서비스를 사용하여 여러 서버를 동시에 마이그레이션하는 것입니다.

이 패턴은 AWS 애플리케이션 마이그레이션 서비스를 사용하여 지원되는 가상 머신(VM)을 AWS 클라우드의 Amazon EC2 인스턴스로 마이그레이션하는 단계를 설명합니다. 이 패턴의 접근 방식을 사용하여 하나 이상의 가상 머신을 수동으로 마이그레이션하거나, 하나씩 마이그레이션하거나, 개략적인 단계에 따라 적절한 자동화 스크립트를 생성하여 자동으로 마이그레이션할 수 있습니다.

사전 조건 및 제한 사항

사전 조건

- 애플리케이션 마이그레이션 서비스를 지원하는 AWS 리전 중 하나에 있는 활성 AWS 계정
- AWS Direct Connect 또는 가상 사설망(VPN)을 사용하는 사설 네트워크 또는 인터넷을 통한 소스 서버와 대상 EC2 서버 간의 네트워크 연결

제한 사항

- 지원되는 리전의 최신 목록은 [지원되는 AWS 리전](#)을 참조하세요.
- 지원되는 운영 체제 목록은 [Amazon EC2 FAQ](#)의 [지원되는 운영 체제](#) 및 일반 섹션을 참조하세요.

아키텍처

소스 기술 스택

- Amazon EC2에서 지원하는 운영 체제를 실행하는 물리적, 가상 또는 클라우드 호스팅 서버

대상 기술 스택

- 소스 VM과 동일한 운영 체제에서 실행되는 Amazon EC2 인스턴스
- Amazon Elastic Block Store(Amazon EBS)

소스 및 대상 아키텍처

다음 다이어그램은 솔루션의 상위 수준 아키텍처와 주요 구성 요소를 보여줍니다. 온프레미스 데이터 센터에는 로컬 디스크가 있는 가상 시스템이 있습니다. AWS에는 복제 서버가 있는 스테이징 영역과 테스트 및 전환을 위한 EC2 인스턴스가 있는 마이그레이션된 리소스 영역이 있습니다. 두 서브넷 모두 EBS 볼륨을 포함합니다.

1. AWS Application Migration Service 초기화
2. 스테이징 영역 리소스를 비롯하여 스테이징 영역 서버 구성 및 보고를 설정합니다.
3. 소스 서버에 에이전트를 설치하고 지속적인 블록 수준 데이터 복제(압축 및 암호화)를 사용합니다.
4. 오케스트레이션 및 시스템 변환을 자동화하여 전환 기간을 단축합니다.

네트워크 아키텍처

다음 다이어그램은 온프레미스 데이터 센터와 AWS의 주요 구성 요소 간 통신에 필요한 프로토콜 및 포트를 포함하여 네트워킹 관점에서 솔루션의 상위 수준 아키텍처와 주요 구성 요소를 보여줍니다.

도구

- [AWS Application Migration Service](#)를 사용하면 변경 없이 다운타임을 최소화하면서 애플리케이션을 AWS 클라우드로 리호스팅(리프트 앤드 시프트)할 수 있습니다.

모범 사례

- 대상 EC2 인스턴스로의 전환이 완료될 때까지 소스 서버를 오프라인 상태로 전환하거나 재부팅하지 마세요.
- 사용자가 대상 서버에서 UAT(사용자 승인 테스트)를 수행하여 문제를 파악하고 해결할 수 있는 충분한 기회를 제공합니다. 이 테스트는 전환 최소 2주 전에 시작하는 것이 좋습니다.
- 애플리케이션 마이그레이션 서비스 콘솔에서 서버 복제 상태를 자주 모니터링하여 문제를 조기에 파악합니다.
- 에이전트 설치에는 영구 IAM 사용자 보안 인증 정보 대신 임시 AWS Identity 및 Access Management(IAM) 보안 인증 정보를 사용합니다.

에픽

AWS 보안 인증 생성

작업	설명	필요한 기술
AWS 복제 에이전트 IAM 역할을 생성합니다.	<p>관리자 권한으로 AWS 계정에 로그인합니다.</p> <p>AWS Identity 및 Access Management(IAM) 콘솔에서 IAM 역할을 만듭니다.</p> <ol style="list-style-type: none"> 1. IAM 콘솔에서 역할을 선택합니다. 2. Create role(역할 생성)을 선택합니다. 3. 신뢰할 수 있는 엔터티 선택 페이지의 신뢰할 수 있는 엔터티 유형 섹션에서 AWS 계정을 선택합니다. 4. AWS 계정 섹션에서 이 계정 (<account-id>)을 선택합니다. 5. 다음을 선택합니다. 	AWS 관리자, 마이그레이션 엔지니어

작업	설명	필요한 기술
	<ol style="list-style-type: none"> 6. 권한 추가 페이지에서 <code>AWSApplicationMigrationAgentInstallationPolicy</code> 정책을 검색한 다음 정책 이름 옆의 확인란을 선택합니다. 7. 다음을 선택합니다. 8. 역할 세부 정보 페이지에서 <code>MGN_Agent_Installation_Role</code>을 역할 이름으로 입력합니다. 9. 필드가 올바른지 확인한 다음 역할 생성을 선택합니다. 	

작업	설명	필요한 기술
임시 보안 인증 생성	<p>AWS Command Line Interface (AWS CLI)가 설치된 시스템에서 관리자 권한으로 로그인합니다. 또는 (지원되는 AWS 지역 내에서) AWS Management Console에서 관리자 권한으로 AWS 계정에 로그인하고 AWS를 열 수도 CloudShell 있습니다.</p> <p>다음 명령으로 임시 보안 인증 정보를 생성하고 <account-id> 을 AWS 계정 ID로 대체합니다.</p> <pre>aws sts assume-role --role-arn arn:aws:iam::<account-id>:role/MGN_Agent_Installation_Role -- role-session-name mgn_installation_session_role</pre> <p>명령의 출력에서 AccessKeyId , SecretAccessKey , SessionToken 의 값을 복사합니다. 나중에 사용할 수 있도록 안전한 장소에 보관하세요.</p> <p>중요: 이러한 임시 보안 인증 정보는 1시간 후에 만료됩니다. 1시간 후에 보안 인증 정보가 필요한 경우 이전 단계를 반복하세요.</p>	AWS 관리자, 마이그레이션 엔지니어

애플리케이션 마이그레이션 서비스 초기화 및 복제 설정 템플릿 생성

작업	설명	필요한 기술
서비스를 초기화합니다.	<p>콘솔에서 관리자 권한으로 AWS 계정에 로그인합니다.</p> <p>애플리케이션 마이그레이션 서비스를 선택한 다음 시작하기를 선택합니다.</p>	AWS 관리자, 마이그레이션 엔지니어
복제 설정 템플릿을 생성하고 구성합니다.	<ol style="list-style-type: none"> 1. 다음과 같은 구성 세부 정보를 제공하세요. <ol style="list-style-type: none"> a. 스테이징 영역 서브넷을 선택합니다. b. 복제 서버 인스턴스 유형(기본값은 t3.small)을 선택합니다. c. EBS 볼륨 유형(기본값은 gp3)을 선택합니다. d. EBS 암호화 옵션을 선택합니다. e. 항상 애플리케이션 마이그레이션 서비스 보안 그룹 사용 확인란이 선택되어 있는지 확인하세요. f. 온프레미스 환경과 AWS 간에 사설 네트워크 연결을 사용하는 경우 데이터 복제에 사설 IP 사용 (VPN DirectConnect, VPC 피어링) 확인란을 선택합니다. g. 애플리케이션 마이그레이션 서비스의 네트워크 대역폭을 제한하려면 네트 	AWS 관리자, 마이그레이션 엔지니어

작업	설명	필요한 기술
	<p>워크 대역폭 조절(서버당 - Mbps 단위) 확인란을 선택합니다.</p> <p>2. 템플릿 생성을 선택합니다.</p> <p>애플리케이션 마이그레이션 서비스는 데이터 복제를 용이하게 하고 마이그레이션된 서버를 시작하는 데 필요한 모든 IAM 역할을 자동으로 생성합니다.</p>	

소스 시스템에 AWS 복제 에이전트 설치

작업	설명	필요한 기술
필요한 AWS 보안 인증 정보를 준비하세요.	소스 서버에서 설치 프로그램 파일을 실행할 때는 <code>AccessKeyId</code> , <code>SecretAccessKey</code> , <code>SessionToken</code> 을 포함하여 이전에 생성한 임시 보안 인증 정보를 입력해야 합니다.	마이그레이션 엔지니어, AWS 관리자
Linux 서버의 경우 에이전트를 설치합니다.	설치 프로그램 명령을 복사하고 소스 시스템에 로그인한 다음 설치 프로그램을 실행합니다. 자세한 지침은 AWS 설명서 를 참조하세요.	AWS 관리자, 마이그레이션 엔지니어
Windows 서버의 경우 에이전트를 설치합니다.	설치 프로그램 파일을 각 서버에 다운로드한 다음 설치 프로그램 명령을 실행합니다. 자세한	AWS 관리자, 마이그레이션 엔지니어

작업	설명	필요한 기술
	한 지침은 AWS 설명서 를 참조하세요.	
초기 데이터 복제가 완료될 때까지 기다리세요.	에이전트가 설치되면 애플리케이션 마이그레이션 서비스 콘솔의 소스 서버 섹션에 소스 서버가 나타납니다. 서버가 초기 데이터 복제를 수행하는 동안 잠시 기다려 주세요.	AWS 관리자, 마이그레이션 엔지니어

시작 설정 구성

작업	설명	필요한 기술
서버 세부 정보를 지정합니다.	애플리케이션 마이그레이션 서비스 콘솔에서 소스 서버 섹션을 선택한 다음, 목록에서 서버 이름을 선택하여 서버 세부 정보에 액세스합니다.	AWS 관리자, 마이그레이션 엔지니어
시작 설정을 구성합니다.	시작 설정 탭을 선택합니다. 일반 시작 설정 및 EC2 시작 템플릿 설정을 비롯한 다양한 설정을 구성할 수 있습니다. 자세한 지침은 AWS 설명서 를 참조하세요.	AWS 관리자, 마이그레이션 엔지니어

테스트 수행

작업	설명	필요한 기술
소스 서버를 테스트합니다.	1. 애플리케이션 마이그레이션 서비스 콘솔의 소스 서버 섹션에서 소스 서버의 마이그	AWS 관리자, 마이그레이션 엔지니어

작업	설명	필요한 기술
	<p>레이션 수명 주기라 테스트 준비 완료이고 데이터 복제 상태가 정상인지 확인합니다.</p> <ol style="list-style-type: none"> 2. 각 소스 서버의 왼쪽에 있는 확인란을 선택합니다. 3. 테스트 및 전환을 선택한 다음, 테스트 인스턴스 시작을 선택합니다. 4. 메시지가 나타나면 시작을 선택합니다. <p>서버가 시작됩니다.</p>	
<p>테스트가 성공적으로 완료되었는지 확인합니다.</p>	<p>테스트 서버가 완전히 시작된 후, 페이지의 경고 상태가 각 서버에 대해 시작됨을 표시합니다.</p>	<p>AWS 관리자, 마이그레이션 엔지니어</p>
<p>서버를 테스트합니다.</p>	<p>테스트 서버를 대상으로 테스트를 수행하여 예상대로 작동하는지 확인합니다.</p>	<p>AWS 관리자, 마이그레이션 엔지니어</p>

전환 예약 및 수행

작업	설명	필요한 기술
<p>전환 기간을 예약합니다.</p>	<p>관련 팀과 함께 적절한 전환을 예약합니다.</p>	<p>AWS 관리자, 마이그레이션 엔지니어</p>
<p>전환을 수행합니다.</p>	<ol style="list-style-type: none"> 1. 애플리케이션 마이그레이션 콘솔의 소스 서버 페이지에서 각 소스 서버의 왼쪽에 있는 확인란을 선택합니다. 	<p>AWS 관리자, 마이그레이션 엔지니어</p>

작업	설명	필요한 기술
	<p>2. 테스트 및 전환을 선택하고 '전환 준비 완료'로 표시를 선택합니다.</p> <p>3. 각 소스 서버의 마이그레이션 수명 주기가 전환 준비 완료 상태인지 확인합니다.</p> <p>4. 테스트 및 전환을 선택한 다음, 전환 인스턴스 시작을 선택합니다.</p> <p>5. 메시지가 나타나면 시작을 선택합니다. 서버가 시작됩니다.</p> <p>소스 서버의 마이그레이션 수명 주기가 전환 진행 중으로 변경됩니다.</p>	
<p>전환이 성공적으로 완료되었는지 확인합니다.</p>	<p>전환 서버가 완전히 시작된 후, 소스 서버 페이지의 경고 상태가 각 서버에 대해 시작됨을 표시합니다.</p>	<p>AWS 관리자, 마이그레이션 엔지니어</p>
<p>서버를 테스트합니다.</p>	<p>전환 서버를 대상으로 테스트를 수행하여 예상대로 작동하는지 확인합니다.</p>	<p>AWS 관리자, 마이그레이션 엔지니어</p>
<p>전환을 마무리합니다.</p>	<p>테스트 및 전환을 선택한 다음, 전환 완료를 선택하여 마이그레이션 프로세스를 마무리합니다.</p>	<p>AWS 관리자, 마이그레이션 엔지니어</p>

관련 리소스

- [AWS Application Migration Service](#)

- [AWS Application Migration Service 사용 설명서](#)

AWS SFTP를 사용하여 온프레미스에서 Amazon S3로 소규모 데이터 세트를 마이그레이션합니다.

작성자: Charles Gibson(AWS)

R 타입: 리호스팅	소스: 스토리지	대상: Amazon S3
작성자: AWS	환경: 프로덕션	기술: 스토리지 및 백업, 마이그레이션
AWS 서비스: Amazon S3		

요약

이 패턴은 AWS Transfer for SFTP(AWS SFTP)를 사용하여 온프레미스 데이터 센터에서 Amazon Simple Storage Service(Amazon S3)로 소규모 데이터 세트(5TB 이하)를 마이그레이션하는 방법을 설명합니다. 데이터는 데이터베이스 덤프 또는 플랫 파일일 수 있습니다.

사전 조건 및 제한 사항

사전 조건

- 활성 상태의 AWS 계정
- 데이터 센터와 AWS 간에 설정된 AWS Direct Connect 링크

제한 사항

- 데이터 파일은 5TB 미만이어야 합니다. 5TB를 초과하는 파일의 경우 Amazon S3에 멀티파트 업로드를 수행하거나 다른 데이터 전송 방법을 선택할 수 있습니다.

아키텍처

소스 기술 스택

- 온프레미스 플랫 파일 또는 데이터베이스 덤프

대상 기술 스택

• Amazon S3

소스 및 대상 아키텍처

도구

- [AWS SFTP](#) - 보안 파일 전송 프로토콜(SFTP)을 사용하여 Amazon S3에서 직접 파일을 주고 받을 수 있습니다.
- [AWS Direct Connect](#) - 온프레미스 데이터 센터에서 AWS로 전용 네트워크 연결을 설정합니다.
- [VPC 엔드포인트](#) — 인터넷 게이트웨이, 네트워크 주소 변환 (NAT) 디바이스, VPN 연결 또는 AWS Direct Connect 연결 PrivateLink 없이 VPC를 AWS에서 제공하는 지원되는 AWS 서비스 및 VPC 엔드포인트 서비스에 비공개로 연결할 수 있습니다. VPC의 인스턴스는 서비스의 리소스와 통신하는데 퍼블릭 IP 주소를 필요로 하지 않습니다.

에픽

마이그레이션 준비

작업	설명	필요한 기술
현재 SFTP 요구 사항을 문서화합니다.		애플리케이션 소유자, SA
인증 요구 사항을 확인하세요.	요구 사항에는 키 기반 인증, 사용자 이름 또는 암호 또는 ID 제공업체(idP)가 포함될 수 있습니다.	애플리케이션 소유자, SA
애플리케이션 통합 요구 사항을 확인합니다.		애플리케이션 소유자
서비스가 필요한 사용자를 식별합니다.		애플리케이션 소유자
SFTP 서버 엔드포인트의 DNS 이름을 결정합니다.		네트워킹

작업	설명	필요한 기술
백업 전략을 결정합니다.		SA, DBA(데이터가 전송되는 경우)
애플리케이션 마이그레이션 또는 전환 전략을 식별합니다.		애플리케이션 소유자, SA, DBA

인프라 구성

작업	설명	필요한 기술
AWS 계정에서 하나 이상의 Virtual Private Cloud(VPC)와 서브넷을 생성합니다.		애플리케이션 소유자, AMS
보안 그룹 및 네트워크 액세스 제어 목록(ACL)을 생성합니다.		보안, 네트워킹, AMS
S3 버킷을 생성합니다.		애플리케이션 소유자, AMS
ID 및 액세스 관리(IAM) 역할을 생성합니다.	AWS SFTP가 S3 버킷에 액세스할 수 있도록 하는 권한이 포함된 IAM 정책을 생성합니다. 이 IAM 정책은 SFTP 사용자에게 제공하는 액세스 수준을 결정합니다. AWS SFTP와 신뢰 관계를 설정하는 또 다른 IAM 정책을 생성합니다.	보안, AMS
등록된 도메인 연결합니다(선택 사항).	자체 등록 도메인이 있는 경우 해당 도메인을 SFTP 서버와 연결할 수 있습니다. SFTP 트래픽을 도메인이나 하위 도메인에서 SFTP 서버 엔드포인트로 라우팅할 수 있습니다.	네트워킹, AMS

작업	설명	필요한 기술
SFTP 서버를 생성합니다.	서비스가 사용하는 자격 증명 공급자 유형을 지정해 사용자를 인증합니다.	애플리케이션 소유자, AMS
SFTP 클라이언트를 엽니다.	SFTP 클라이언트를 열고 사용할 SFTP 서버에 대해 SFTP 엔드포인트 호스트 이름을 사용하도록 연결을 구성합니다. AWS SFTP는 모든 표준 SFTP 클라이언트를 지원합니다. 일반적으로 사용되는 SFTP 클라이언트에는 OpenSSH, WinSCP, 사이버덕 등이 있습니다. FileZilla AWS SFTP 콘솔에서 SFTP 서버 호스트 이름을 가져올 수 있습니다.	애플리케이션 소유자, AMS

계획 및 테스트

작업	설명	필요한 기술
애플리케이션 마이그레이션을 계획합니다.	필요한 애플리케이션 구성 변경에 대한 계획을 세우고, 마이그레이션 날짜를 설정하고, 테스트 일정을 결정하세요.	애플리케이션 소유자, AMS
인프라를 테스트합니다.	비프로덕션 환경에서 테스트합니다.	애플리케이션 소유자, AMS

관련 리소스

참조

- [AWS Transfer for SFTP 사용 설명서](#)

- [AWS Direct Connect 리소스](#)
- [VPC 엔드포인트](#)

자습서 및 동영상

- [AWS Transfer for SFTP\(동영상\)](#)
- [AWS Transfer for SFTP 사용 설명서](#)
- [AWS SA Whiteboarding - Direct Connect\(동영상\)](#)

오라클에서 AWS Elastic GlassFish Beanstalk로 마이그레이션

작성자: Sandeep Bondugula(AWS)

R 유형: 리호스팅	소스: 애플리케이션 개발	대상: AWS Elastic Beanstalk
작성자: AWS	환경: PoC 또는 파일럿	기술: 컨테이너 및 마이크로서비스, 웹 및 모바일 앱, 마이그레이션
워크로드: 오픈 소스, Oracle	AWS 서비스: AWS Elastic Beanstalk	

요약

이 패턴은 온프레미스 Oracle GlassFish 서버에서 실행되는 Java 애플리케이션을 AWS 클라우드의 AWS Elastic Beanstalk로 마이그레이션하는 방법을 설명합니다.

AWS에서는 아마존 Elastic Compute Cloud (아마존 EC2) Auto Scaling 그룹에서 실행되는 AWS Elastic Beanstalk가 설치된 도커 GlassFish 서버에 자바 애플리케이션을 배포합니다.

기타 기능:

- Amazon Elastic Beanstalk는 여러 기본 리소스의 래퍼 역할을 합니다. 이는 (Amazon Route 53에서 들어오는 트래픽을 처리하는) Elastic Load Balancing을 설정하고, 트래픽을 하나 이상의 EC2 인스턴스로 분산하며, 배포 도구 역할도 합니다.
- Amazon Relational Database Service(RDS)로 온프레미스 데이터베이스를 마이그레이션하려면 데이터베이스 연결 세부 정보를 업데이트합니다. 백엔드 데이터베이스에서 Amazon RDS 다중 AZ 배포를 구성하고 데이터베이스 엔진 유형을 선택할 수 있습니다.
- Auto Scaling 그룹 및 규모 조정 정책과 함께 높은 가용성을 위한 다중 AZ 배포를 사용하여 복원력을 개선할 수 있습니다.
- Amazon CloudWatch 지표를 기반으로 조정 정책을 설정할 수 있습니다.
- AWS Elastic Beanstalk에서는 기본 Elastic Load Balancing 설정과 Amazon EC2 Auto Scaling을 구성할 수 있습니다.

사전 조건 및 제한 사항

사전 조건

- 활성 상태의 AWS 계정
- 에서 실행되는 온프레미스 Java 애플리케이션 GlassFish
- Java 웹 애플리케이션 리소스(WAR) 파일

제품 버전

- Oracle Glassfish 4.1.2 및 5.0
- 자바 7 4.0 GlassFish
- 자바 8 GlassFish 4.1 이상

아키텍처

소스 기술 스택

- 에서 개발된 애플리케이션 GlassFish

대상 기술 스택

- Elastic Beanstalk

대상 아키텍처

배포 워크플로

도구

- [Amazon Elastic Beanstalk](#)-Java, .NET, PHP, Node.js, Python, Ruby, Go, 및 Docker를 사용하여 개발된 웹 애플리케이션 및 서비스를 Apache, NGINX, Passenger, 및 IIS와 같은 서버에 배포하고 확장하기 위한 서비스입니다.

- [Amazon CloudWatch](#) — 애플리케이션을 모니터링하고, 시스템 전반의 성능 변화에 대응하고, 리소스 사용률을 최적화하고, 운영 상태를 통합적으로 파악하기 위한 데이터와 실행 가능한 통찰력을 제공합니다.
- [Docker](#)-소프트웨어를 표준화된 단위로 패키징하여 애플리케이션을 신속하게 구축, 테스트 및 배포하는 플랫폼입니다.
- [Java](#)-범용 프로그래밍 언어입니다. Java는 클래스 기반의 객체 지향적이며 구현 종속성이 적도록 설계되었습니다.

에픽

VPC를 설정

작업	설명	필요한 기술
필수 정보를 사용하여 Virtual Private Cloud(VPC) 인스턴스를 생성합니다.		SysAdmin
VPC 내에 2개 이상의 서브넷을 생성합니다.		SysAdmin
요구 사항에 따라 라우팅 테이블을 생성합니다.		SysAdmin

Amazon S3 설정

작업	설명	필요한 기술
Amazon Simple Storage Service(S3) 버킷을 생성합니다.		SysAdmin
WAR 파일을 S3 버킷에 복사하고 애플리케이션 코드를 업로드합니다.		SysAdmin

IAM 역할 생성

작업	설명	필요한 기술
AWS Identity 및 Access Management(IAM) 역할을 생성합니다.	기본 "aws-elasticbeanstalk-ec2-역할" 프로필을 사용하거나 Elastic Beanstalk가 자동으로 생성하도록 할 수 있습니다.	SysAdmin

AWS Elastic Beanstalk 설정

작업	설명	필요한 기술
Elastic Beanstalk 대시보드를 엽니다.		SysAdmin
새 애플리케이션을 생성하고 웹 서버 환경을 선택합니다.		SysAdmin
GlassFish Docker를 사전 구성된 플랫폼으로 선택합니다.		SysAdmin
코드를 업로드합니다.	로컬 시스템 파일의 S3 버킷 파일 URL 또는 ZIP 파일을 제공합니다.	SysAdmin
환경 유형을 선택합니다.	구성 용량 설정에서 단일 인스턴스 또는 로드 밸런서를 선택합니다.	SysAdmin
로드 밸런서를 구성합니다.	이전 단계에서 로드 밸런서를 선택한 경우 다중 AZ 배포를 구성합니다.	SysAdmin
구성 보안 설정에서 이전에 생성한 IAM 역할을 선택합니다.		SysAdmin

작업	설명	필요한 기술
구성 보안 설정에서 기존 키 페어가 있는 경우 해당 키 페어를 사용하거나 새 Amazon EC2 키 페어를 생성합니다.		SysAdmin
구성 모니터링 설정에서 Amazon을 구성합니다 CloudWatch.		SysAdmin
구성 보안 설정에서 이전에 생성한 VPC를 선택합니다.		SysAdmin
환경 생성을 선택합니다.		SysAdmin

애플리케이션 테스트

작업	설명	필요한 기술
생성된 환경에서 제공된 URL을 사용하여 애플리케이션을 테스트합니다.		
Amazon Route 53에서 도메인 이름 서비스(DNS) 변경 사항을 적용합니다.		

관련 리소스

- [오라클 GlassFish 설명서](#)
- [GlassFish 오픈소스 자바 EE 레퍼런스 구현](#)
- [AWS Elastic Beanstalk 설명서](#)
- [아마존에서 Elastic Beanstalk 사용하기 CloudWatch](#)
- [AWS Elastic Beanstalk 가격 책정](#)
- [EC2 Auto Scaling 그룹](#)

- [Auto Scaling 그룹의 크기 조정](#)
- [Amazon RDS 다중 AZ 배포](#)

온프레미스 Oracle 데이터베이스를 Amazon EC2의 Oracle로 마이그레이션

작성자: Baji Shaik(AWS) 및 Pankaj Choudhary(AWS)

환경: PoC 또는 파일럿	소스: 데이터베이스: 관계형	대상: Amazon EC2 기반 Oracle
R 유형: 리호스팅	워크로드: Oracle	기술: 마이그레이션, 데이터베이스
AWS 서비스: Amazon EC2		

요약

이 패턴은 온프레미스 Oracle 데이터베이스를 Amazon Elastic Compute Cloud(Amazon EC2) 인스턴스의 Oracle로 마이그레이션하는 단계를 안내합니다. 여기서는 두 가지 마이그레이션 옵션을 설명합니다. 하나는 AWS 데이터 마이그레이션 서비스 (AWS DMS) 를 사용하는 것이고 다른 하나는 RMAN, 데이터 펌프 가져오기/내보내기, 전송 가능한 테이블스페이스 및 Oracle과 같은 기본 Oracle 도구를 사용하는 것입니다. GoldenGate

사전 조건 및 제한 사항

사전 조건

- 활성 상태의 AWS 계정
- 온프레미스 데이터 센터의 소스 Oracle 데이터베이스

제한 사항

- 대상 운영 체제(OS)는 Amazon EC2에서 지원해야 합니다. 지원되는 시스템의 전체 목록은 [Amazon EC2 FAQ](#)를 참조하십시오.

제품 버전

- Enterprise, Standard, Standard One 및 Standard Two 버전용 Oracle 버전 10.2 이상(버전 10.x의 경우), 11g, 최대 12.2 및 18c. AWS DMS에서 지원하는 최신 버전 목록은 AWS DMS 설명서의 [데이터 마이그레이션 소스](#) '온프레미스 및 Amazon EC2 인스턴스 데이터베이스'를 참조하십시오.

아키텍처

소스 기술 스택

- 온프레미스 Oracle 데이터베이스

대상 기술 스택

- Amazon EC2의 Oracle 데이터베이스 인스턴스

대상 아키텍처

데이터 마이그레이션 아키텍처

AWS DMS 사용:

네이티브 Oracle 도구 사용:

도구

- AWS DMS - [AWS Database Migration Service\(AWS DMS\)](#)는 여러 소스 및 대상 데이터베이스를 지원합니다. 지원되는 데이터베이스 버전 및 에디션에 대한 자세한 내용은 [AWS DMS용 소스로 Oracle 데이터베이스 사용](#)을 참조하십시오. 가장 종합적인 버전 및 기능 지원을 위해 최신 버전의 AWS DMS를 사용하는 것을 권장합니다.
- 네이티브 오라클 도구 - RMAN, 데이터 펌프 가져오기/내보내기, 전송 가능한 테이블스페이스, Oracle GoldenGate

에픽

마이그레이션 계획

작업	설명	필요한 기술
소스 및 대상 데이터베이스의 버전을 확인합니다.		DBA
대상 OS의 버전을 식별합니다.		DBA, SysAdmin
Oracle 호환성 목록 및 용량 요구 사항을 기반으로 대상 서버 인스턴스의 하드웨어 요구 사항을 식별합니다.		DBA, SysAdmin
스토리지 요구 사항(스토리지 유형 및 용량)을 식별합니다.		DBA, SysAdmin
네트워크 요구 사항(지연 시간 및 대역폭) 파악.		DBA, SysAdmin
용량, 스토리지 기능, 네트워크 기능에 따라 적절한 인스턴스 유형을 선택합니다.		DBA, SysAdmin
소스 및 대상 데이터베이스의 네트워크 및 호스트 액세스 보안 요구 사항을 확인합니다.		DBA, SysAdmin
Oracle 소프트웨어 설치에 필요한 OS 사용자의 목록을 식별합니다.		DBA, SysAdmin
AWS Schema Conversion Tool(AWS SCT) 및 드라이버를 다운로드하십시오.		DBA

작업	설명	필요한 기술
워크로드용 AWS SCT 프로젝트를 생성하고 소스 데이터베이스에 연결합니다.		DBA
객체(테이블, 인덱스, 시퀀스 등) 생성을 위한 SQL 파일을 생성합니다.		DBA
백업 전략을 결정합니다.		DBA, SysAdmin
가용성 요구 사항을 결정합니다.		DBA
애플리케이션 마이그레이션/전환 전략을 파악합니다.		DBA SysAdmin, 앱 소유자

인프라 구성

작업	설명	필요한 기술
AWS 계정에 Virtual Private Cloud(VPC) 및 서브넷을 생성합니다.		SysAdmin
보안 그룹 및 네트워크 액세스 제어 목록(ACL)을 생성합니다.		SysAdmin
EC2 인스턴스를 구성하고 시작합니다.		SysAdmin

Oracle 소프트웨어 설치

작업	설명	필요한 기술
Oracle 소프트웨어에 필요한 OS 사용자 및 그룹을 생성합니다.		DBA, SysAdmin
필요한 버전의 Oracle 소프트웨어를 다운로드합니다.		
EC2 인스턴스에 Oracle 소프트웨어를 설치합니다.		DBA, SysAdmin
AWS SCT에서 생성한 스크립트를 사용하여 테이블, 기본 키, 보기, 시퀀스와 같은 객체를 생성합니다.		DBA

데이터 마이그레이션 - 옵션 1

작업	설명	필요한 기술
기본 Oracle 도구 또는 타사 도구를 사용하여 데이터베이스 개체 및 데이터를 마이그레이션하십시오.	Oracle 도구에는 데이터 펌프 가져오기/내보내기, RMAN, 전송 가능한 테이블스페이스 등이 포함됩니다. GoldenGate	DBA

데이터 마이그레이션 - 옵션 2

작업	설명	필요한 기술
마이그레이션 방법을 결정하십시오.		DBA

작업	설명	필요한 기술
AWS DMS 콘솔에 복제 인스턴스를 생성합니다.		DBA
소스 및 대상 엔드포인트를 생성합니다.		DBA
복제 작업을 생성합니다.		DBA
변경 데이터 캡처(CDC)를 활성화하여 연속 복제를 위한 변경 사항을 캡처합니다.		DBA
복제 작업을 실행하고 로그를 모니터링합니다.		DBA
전체 로드가 완료되면 인덱스 및 외래 키와 같은 보조 객체를 생성합니다.		DBA

애플리케이션 마이그레이션

작업	설명	필요한 기술
애플리케이션 마이그레이션 전략을 따릅니다.		SysAdminDBA, 앱 소유자

전환

작업	설명	필요한 기술
애플리케이션 전환/스위치오버 전략을 따릅니다.		DBA SysAdmin, 앱 소유자

프로젝트 닫기

작업	설명	필요한 기술
임시 AWS Secrets Manager 리소스를 종료하십시오.		DBA, SysAdmin
프로젝트 문서를 검토하고 검증하세요.		DBA SysAdmin, 앱 소유자
마이그레이션 시간, 수동 대비 도구 비율(%), 비용 절감 등에 대한 지표를 수집합니다.		DBA SysAdmin, 앱 소유자
프로젝트를 마무리하고 피드백을 제공하세요.		

관련 리소스

참조

- [Oracle 데이터베이스를 AWS로 마이그레이션하는 전략](#)
- [Oracle 데이터베이스를 AWS 클라우드로 마이그레이션하기](#)
- [Amazon EC2 웹 사이트](#)
- [AWS DMS 웹사이트](#)
- [AWS DMS 블로그 게시물](#)
- [Amazon EC2 요금](#)
- [클라우드 컴퓨팅 환경에서의 Oracle 소프트웨어 라이선스](#)

자습서 및 비디오

- [Amazon EC2 시작하기](#)
- [AWS DMS 시작하기](#)
- [Amazon EC2 소개 - AWS 기반 탄력적 클라우드 서버 및 호스팅\(동영상\)](#)

Oracle Data Pump를 사용하여 온프레미스 Oracle 데이터베이스를 Amazon EC2 로 마이그레이션

작성자: Navakanth Talluri

환경: PoC 또는 파일럿	소스: 온프레미스 Oracle 데이터베이스	대상: Amazon EC2의 Oracle 데이터베이스
R 유형: 리호스팅	워크로드: Oracle	기술: 마이그레이션, 데이터베이스
AWS 서비스: Amazon EC2, Direct Connect		

요약

데이터베이스를 마이그레이션할 때는 소스 및 대상 데이터베이스 엔진과 버전, 마이그레이션 도구 및 서비스, 허용 가능한 가동 중지 기간 등의 요소를 고려해야 합니다. 온프레미스 Oracle 데이터베이스를 Amazon Elastic Compute Cloud(Amazon EC2)로 마이그레이션하는 경우 Oracle Data Pump 및 Oracle Recovery Manager(RMAN)와 같은 Oracle 도구를 사용할 수 있습니다. 자세한 내용은 백서 [Oracle 데이터베이스를AWS 클라우드로 마이그레이션하기](#)를 참조하십시오.

Oracle Data Pump는 데이터베이스의 논리적이고 일관된 백업을 추출하여 대상 EC2 인스턴스로 복원하는 데 도움이 됩니다. 이 패턴은 Oracle Data Pump와 NETWORK_LINK 파라미터를 사용하여 가동 중지 시간을 최소화하면서 온프레미스 Oracle 데이터베이스를 EC2 인스턴스로 마이그레이션하는 방법을 설명합니다. NETWORK_LINK 파라미터는 데이터베이스 링크를 통해 가져오기를 시작합니다. 대상 EC2 인스턴스의 Oracle Data Pump Import(impdp) 클라이언트는 원본 데이터베이스에 연결하고, 원본 데이터베이스에서 데이터를 검색하며, 대상 인스턴스의 데이터베이스에 직접 데이터를 기록합니다. 이 솔루션에는 백업 또는 덤프 파일이 사용되지 않습니다.

사전 조건 및 제한 사항

사전 조건

- 활성 상태의 AWS 계정
- 온프레미스 Oracle 데이터베이스 다음과 같습니다.
 - Oracle Real Application Clusters(RAC) 데이터베이스가 아닙니다.

- Oracle Automatic Storage Management(Oracle ASM) 데이터베이스가 아닙니다.
- 읽기-쓰기 모드입니다.
- 온프레미스 데이터 센터와 AWS 간에 Direct Connect 링크를 생성했습니다. 자세한 내용은 [연결 생성](#)(Direct Connect 설명서)을 참조하십시오.

제품 버전

- Oracle Database 10g 릴리스 1 (10.1) 이상

아키텍처

소스 기술 스택

- 온프레미스 데이터 센터의 독립형(비 RAC 및 비 ASM) Oracle 데이터베이스 서버

대상 기술 스택

- Amazon EC2에서 실행되는 Oracle 데이터베이스

대상 아키텍처

Well-Architected Framework의 [신뢰성 원칙](#)에서는고가용성과 복원력을 제공하는 데 도움이 되는 데이터 백업을 생성할 것을 권장합니다. 자세한 내용은 AWS에서 Oracle 데이터베이스 실행에 관한 모범 사례에 나와 있는 [고가용성을 위한 아키텍처](#)를 참조하십시오. 이 패턴은 Oracle Active Data Guard를 사용하여 EC2 인스턴스에 기본 및 대기 데이터베이스를 설정합니다.고가용성을 위해 EC2 인스턴스는 서로 다른 가용 영역에 있어야 합니다. 하지만 가용 영역은 동일한 리전 또는 서로 다른 여러 리전에 있을 수 있습니다.

Active Data Guard는 물리적 대기 데이터베이스에 대한 읽기 전용 액세스를 제공하고 기본 데이터베이스의 재실행 변경 사항을 지속적으로 적용합니다. Recovery Point Objective(RPO) 및 Recovery Time Objective(RTO)에 기반하여 동기식 전송 재실행 옵션과 비동기식 전송 재실행 옵션 중에서 선택할 수 있습니다.

다음 이미지는 기본 및 대기 EC2 인스턴스가 서로 다른 여러 리전에 있는 경우의 대상 아키텍처를 보여줍니다.

데이터 마이그레이션 아키텍처

대상 아키텍처 설정을 완료한 후에는 Oracle Data Pump를 사용하여 온프레미스 데이터와 스키마를 기본 EC2 인스턴스로 마이그레이션합니다. 전환 중에는 애플리케이션이 온프레미스 데이터베이스 또는 대상 데이터베이스에 액세스할 수 없습니다. 기본 EC2 인스턴스의 새 대상 데이터베이스에 연결할 수 있을 때까지 이러한 애플리케이션을 종료합니다.

다음 이미지는 데이터 마이그레이션 중의 아키텍처를 보여줍니다. 이 샘플 아키텍처에서, 기본 및 대기 EC2 인스턴스는 서로 다른 여러 지역에 있습니다.

도구

서비스

- [Direct Connect](#)를 사용하면 표준 Ethernet 광섬유 케이블을 통해 내부 네트워크를 Direct Connect 위치에 연결할 수 있습니다. 이 연결을 구성하면, 네트워크 경로에서 인터넷 서비스 제공업체를 우회하여 퍼블릭 서비스에 직접 가상 인터페이스를 생성할 수 있습니다.
- [Amazon Elastic Compute Cloud\(Amazon EC2\)](#)는 AWS 클라우드에서 확장 가능한 컴퓨팅 용량을 제공합니다. 필요한 만큼 가상 서버를 시작하고 빠르게 스케일 업하거나 스케일 다운할 수 있습니다.

기타 도구 및 서비스

- [Oracle Active Data Guard](#)는 대기 데이터베이스를 생성, 유지, 관리, 모니터링하는 데 도움이 됩니다.
- [Oracle Data Pump](#)를 사용하면 한 데이터베이스에서 다른 데이터베이스로 데이터와 메타데이터를 빠른 속도로 이동할 수 있습니다.

모범 사례

- [Best Practices for Running Oracle Database on AWS](#)
- [NETWORK_LINK를 사용하여 데이터 가져오기](#)

에픽

EC2 인스턴스 설정

작업	설명	필요한 기술
<p>온프레미스 호스트의 소스 하드웨어 구성과 커널 파라미터를 식별합니다.</p>	<p>스토리지 크기, 초당 입출력 작업 처리량(IOPS), CPU를 포함한 온프레미스 구성을 검증합니다. 이는 CPU 코어를 기반으로 하는 Oracle 라이선싱에 중요합니다.</p>	<p>DBA, SysAdmin</p>
<p>AWS에서 인프라를 생성합니다.</p>	<p>Virtual Private Cloud(VPC), 프라이빗 서브넷, 보안 그룹, 네트워크 액세스 제어 목록(ACL), 라우팅 테이블, 인터넷 게이트웨이를 생성합니다. 자세한 내용은 다음 자료를 참조하세요.</p> <ul style="list-style-type: none"> • VPC 및 서브넷 • 자습서: 데이터베이스 인스턴스와 사용할 VPC 생성 	<p>DBA, AWS 시스템 관리자</p>
<p>Active Data Guard를 사용하여 EC2 인스턴스를 설정합니다.</p>	<p>Well-Architected Framework에 설명된 대로 Active Data Guard 구성을 사용하여 EC2 인스턴스를 구성합니다. 이 패턴은 논리적 백업을 사용하기 때문에 EC2 인스턴스의 Oracle 데이터베이스 버전은 온프레미스 버전과 다를 수 있습니다. 다음을 참고합니다.</p> <ul style="list-style-type: none"> • 대상 데이터베이스를 읽기-쓰기 모드로 설정합니다. • 대상 데이터베이스에서 원본 데이터베이스에 대 	<p>DBA, AWS 시스템 관리자</p>

작업	설명	필요한 기술
	<p>한 Transparent Network Substrate(TNS) 세부 정보를 제공합니다.</p> <p>자세한 내용은 다음을 참조하십시오.</p> <ul style="list-style-type: none"> • 데이터베이스 시작 (Oracle 설명서) • Oracle 데이터베이스의 생성 및 구성 (Oracle 설명서) 	

데이터베이스를 Amazon EC2로 마이그레이션

작업	설명	필요한 기술
EC2 인스턴스에서 온프레미스 데이터베이스에 대한 dblink를 생성합니다.	EC2 인스턴스의 Oracle 데이터베이스와 온프레미스 Oracle 데이터베이스 간에 데이터베이스 링크(dblink)를 생성합니다. 자세한 내용은 네트워크 링크 가져오기를 사용하여 데이터 이동하기 (Oracle 설명서)를 참조하십시오.	DBA
EC2 인스턴스와 온프레미스 호스트 간의 연결을 검증합니다.	dblink를 사용하여, EC2 인스턴스와 온프레미스 데이터베이스 간의 연결이 제대로 작동하는지 확인합니다. 지침은 데이터베이스 링크 생성 (Oracle 설명서)을 참조하십시오.	DBA
온프레미스 데이터베이스에 연결된 모든 애플리케이션을 중지합니다.	데이터베이스 가동 중지 시간이 승인된 후에는 온프레미스 데이터베이스에 연결되는 모든	DBA, 앱 개발자

작업	설명	필요한 기술
	<p>애플리케이션과 종속 작업을 종료합니다. 이는 애플리케이션에서 직접 수행하거나 cron 을 사용하여 데이터베이스에서 수행할 수 있습니다. 자세한 내용은 Oracle Linux에서 Crontab Utility를 사용하여 작업을 스케줄링하기를 참조하십시오.</p>	
<p>데이터 마이그레이션 작업을 스케줄링합니다.</p>	<p>대상 호스트에서, 명령 impdb(을)를 사용하여 Data Pump 가져오기를 스케줄링합니다. 그러면 대상 데이터베이스가 온프레미스 호스트에 연결되고 데이터 마이그레이션이 시작됩니다. 자세한 내용은 Data Pump 가져오기 및 NETWORK_LINK (Oracle 설명서)를 참조하십시오.</p>	<p>DBA</p>
<p>데이터 마이그레이션을 검증합니다.</p>	<p>데이터 검증은 중요한 단계입니다. 데이터 검증을 위해 사용자 지정 도구 또는 Oracle 도구 (예: dblink 및 SQL 쿼리의 조합)를 사용할 수 있습니다.</p>	<p>DBA</p>

전환

작업	설명	필요한 기술
<p>소스 데이터베이스를 읽기 전용 모드로 전환합니다.</p>	<p>응용 프로그램이 종료되고 소스 데이터베이스가 변경되지 않았는지 확인합니다. 소스 데이터베이스를 읽기 전용 모드</p>	<p>DBA, DevOps 엔지니어, 앱 개발자</p>

작업	설명	필요한 기술
	<p>로 엽니다. 이렇게 하면 트랜잭션 열림을 방지할 수 있습니다. 자세한 내용은 SQL 성명서(Oracle 설명서)의 ALTER DATABASE을(를) 참조하십시오.</p>	
<p>객체 수와 데이터를 검증합니다.</p>	<p>데이터와 객체를 검증하려면 사용자 지정 도구 또는 Oracle 도구(예: dblink 및 SQL 쿼리의 조합)를 사용합니다.</p>	<p>DBA, 앱 개발자</p>
<p>애플리케이션을 기본 EC2 인스턴스의 데이터베이스에 연결합니다.</p>	<p>기본 EC2 인스턴스에서 생성한 새 데이터베이스를 포인팅하도록 애플리케이션의 연결 속성을 변경합니다.</p>	<p>DBA, 앱 개발자</p>
<p>애플리케이션 성능을 검증합니다.</p>	<p>애플리케이션을 시작합니다. 자동 워크로드 리포지토리를 사용하여 애플리케이션의 기능과 성능을 검증합니다(Oracle 설명서).</p>	<p>앱 개발자, DevOps 엔지니어, DBA</p>

관련 리소스

AWS 참조

- [클라우드로 Oracle 데이터베이스를 마이그레이션하기](#)
- [Oracle용 Amazon EC2](#)
- [교차 플랫폼 환경을 위해 대형 Oracle 데이터베이스를 AWS로 마이그레이션하기](#)
- [VPC 및 서브넷](#)
- [자습서: 데이터베이스 인스턴스와 같이 사용할 VPC 생성](#)

Oracle 참고 문헌

- [Oracle Data Guard 구성](#)
- [Data Pump 가져오기](#)

온프레미스 SAP ASE 데이터베이스를 Amazon EC2로 마이그레이션

작성자: Sergey Dmitriev(AWS)

R 유형: 리호스팅	소스: 데이터베이스: 관계형	대상: Amazon EC2의 SAP Adaptive Server Enterprise
작성자: AWS	환경: PoC 또는 파일럿	기술: 데이터베이스, 마이그레이션
워크로드: SAP	AWS 서비스: Amazon EC2	

요약

이 패턴은 온프레미스 호스트에서 Amazon Elastic Compute Cloud(Amazon EC2) 인스턴스로 SAP Adaptive Server Enterprise(ASE) 데이터베이스를 마이그레이션하는 방법을 설명합니다. 이 패턴은 ASE Cockpit, ASE용 Sybase Central, 마이그레이션용 DBA Cockpit과 같은 AWS Database Migration Service(DMS) 또는 SAP ASE 네이티브 도구의 사용을 다룹니다.

사전 조건 및 제한 사항

사전 조건

- 활성 상태의 AWS 계정.
- 온프레미스 데이터 센터의 SAP ASE 소스 데이터베이스

제한 사항

- 소스 데이터베이스는 64TB 미만이어야 합니다.

제품 버전

- SAP ASE 버전 15.x 및 16.x 이상

아키텍처

소스 기술 스택

- 온프레미스 SAP ASE 데이터베이스

대상 기술 스택

- EC2 인스턴스의 SAP ASE 데이터베이스

데이터베이스 마이그레이션 아키텍처

DMS 사용:

네이티브 SAP ASE 도구 사용:

도구

- DMS - [Database Migration Service](#)(DMS)는 여러 소스 및 대상 데이터베이스를 지원합니다. 자세한 내용은 [데이터 마이그레이션용 소스](#) 및 [데이터 마이그레이션용 대상](#)을 참조하십시오. 가장 종합적인 버전 및 기능 지원을 위해 최신 버전의 AWS DMS를 사용하는 것을 권장합니다.
- SAP ASE - 네이티브 도구에는 ASE Cockpit, ASE용 Sybase Central, DBA Cockpit이 포함됩니다.

에픽

마이그레이션 분석

작업	설명	필요한 기술
소스 및 대상 데이터베이스 버전을 검증합니다.		DBA
대상 OS 버전을 식별합니다.		DBA, SysAdmin
SAP ASE 호환성 목록 및 용량 요구 사항을 기반으로 대상 서		DBA, SysAdmin

작업	설명	필요한 기술
버 인스턴스의 하드웨어 요구 사항을 식별합니다.		
스토리지 유형 및 용량에 대한 요구 사항을 확인합니다.		DBA, SysAdmin
지연 시간 및 대역폭을 포함한 네트워크 요구 사항을 확인합니다.		DBA, SysAdmin
적절한 인스턴스 유형, 용량, 스토리지 특성 및, 네트워크 특성을 선택합니다.		DBA, SysAdmin
소스 및 대상 데이터베이스의 네트워크 및 호스트 액세스 보안 요구 사항을 확인합니다.		DBA, SysAdmin
SAP ASE 소프트웨어 설치에 필요한 운영 체제 사용자의 목록을 식별합니다.		DBA, SysAdmin
백업 전략을 결정합니다.		DBA
가용성 요구 사항을 결정합니다.		DBA
애플리케이션 마이그레이션 또는 전환 전략을 확인합니다.		DBA SysAdmin, 앱 소유자

인프라 구성

작업	설명	필요한 기술
Virtual Private Cloud(VPC) 및 서브넷을 생성합니다.		SysAdmin

작업	설명	필요한 기술
보안 그룹 및 네트워크 액세스 제어 목록(ACL)을 생성합니다.		SysAdmin
EC2 인스턴스를 구성하고 시작합니다.		SysAdmin

소프트웨어를 설치합니다.

작업	설명	필요한 기술
SAP ASE 소프트웨어가 작동하는 데 필요한 OS 사용자 및 그룹을 생성합니다.		DBA, SysAdmin
필요한 SAP ASE 소프트웨어를 다운로드합니다.		DBA, SysAdmin
EC2 인스턴스에 SAP ASE 데이터베이스, 백업 서버 소프트웨어 및 복제 서버 소프트웨어를 설치한 다음 서버를 구성합니다.		DBA, SysAdmin

데이터 마이그레이션하기 - 옵션 1

작업	설명	필요한 기술
네이티브 SAP ASE 또는 타사 도구를 사용하여 데이터베이스 객체 및 데이터를 마이그레이션합니다.	SAP ASE 또는 타사 도구에 대한 설명서를 참조하십시오. 여기에는 ASE 및 DBA Cockpit용 ASE Cockpit, Sybase Central이 포함됩니다.	DBA

데이터 마이그레이션하기 - 옵션 2

작업	설명	필요한 기술
DMS를 사용하여 데이터를 마이그레이션합니다.		DBA

애플리케이션 마이그레이션

작업	설명	필요한 기술
애플리케이션 마이그레이션 전략을 따릅니다.		DBA SysAdmin, 앱 소유자

전환

작업	설명	필요한 기술
애플리케이션 컷오버 또는 전환 전략을 따릅니다.		DBA SysAdmin, 앱 소유자

프로젝트 닫기

작업	설명	필요한 기술
임시 AWS 리소스를 종료합니다.		DBA, SysAdmin
프로젝트 문서를 검토 및 유효성을 확인합니다.		DBA SysAdmin, 앱 소유자
마이그레이션 시간, 수동 대비 도구 비용 절감 비율 등에 대한 지표를 모읍니다.		DBA SysAdmin, 앱 소유자

작업	설명	필요한 기술
프로젝트를 종료하고 피드백을 제공하세요.		DBA SysAdmin, 앱 소유자

관련 리소스

참조

- [Amazon EC2](#)
- [DMS](#)
- [Amazon EC2 요금 정책](#)

자습서 및 동영상

- [Amazon EC2 시작하기](#)
- [AWS Database Migration Service 시작하기](#)
- [Database Migration Service\(동영상\)](#)
- [Amazon EC2 소개 - 탄력적 클라우드 서버 및 호스팅\(동영상\)](#)

온프레미스 Microsoft SQL Server 데이터베이스를 Amazon EC2로 마이그레이션

작성자: 마크 잘키에비치(AWS)

R 유형: 리호스팅	소스: 데이터베이스: 관계형	대상: Amazon EC2 기반 Microsoft SQL Server
작성자: AWS	환경: PoC 또는 파일럿	기술: 데이터베이스, 마이그레이션
워크로드: Microsoft	AWS 서비스: Amazon EC2	

요약

이 패턴은 온프레미스 Microsoft SQL Server 데이터베이스를 Amazon Elastic Compute Cloud(Amazon EC2) 인스턴스의 Microsoft SQL Server로 마이그레이션하는 방법을 설명합니다. 여기에는 두 가지 마이그레이션 옵션이 포함됩니다. 하나는 AWS Data Migration Service(AWS DMS)를 사용하는 것이고 다른 하나는 백업 및 복원, Copy Database Wizard 또는 데이터베이스 복사 및 연결과 같은 기본 Microsoft SQL Server 도구를 사용하는 것입니다.

사전 조건 및 제한 사항

사전 조건

- 활성 상태의 AWS 계정.
- Amazon EC2에서 지원하는 운영 체제(지원되는 운영 체제 버전의 전체 목록은 [Amazon EC2 FAQ](#)를 참조)
- 온프레미스 데이터 센터의 Microsoft SQL Server 소스 데이터베이스

제품 버전

- AWS DMS를 사용하는 경우, Enterprise, Standard, Workgroup 및 Developer 버전용 Microsoft SQL Server 버전 2005, 2008, 2008R2, 2012, 2014, 2016, 2017. Microsoft SQL Server 웹 또는 Express 에디션을 마이그레이션하려면 기본 도구 또는 타사 도구를 사용하세요. 지원되는 버전의 최신 목록은 [Microsoft SQL Server 데이터베이스를 AWS DMS의 대상으로 사용하기](#) 섹션을 참조하세요.

아키텍처

소스 기술 스택

- 온프레미스 Microsoft SQL Server 데이터베이스

대상 기술 스택

- EC2 인스턴스의 Microsoft SQL Server 데이터베이스

대상 아키텍처

데이터 마이그레이션 아키텍처

- AWS DMS 사용

- 기본 SQL 서버 도구 사용

도구

- AWS DMS - [AWS Data Migration Service](#)(AWS DMS)를 사용하면 Oracle, SQL Server, MySQL 및 PostgreSQL을 비롯하여 널리 사용되는 상용 및 오픈 소스 데이터베이스에서 데이터를 마이그레이션할 수 있습니다. AWS DMS를 사용하여 데이터를 온프레미스 인스턴스 간(AWS 클라우드 설정을 통해) 또는 클라우드와 온프레미스 데이터베이스 조합 간에 AWS Cloud로 마이그레이션할 수 있습니다.
- 기본 Microsoft SQL Server 도구 - 여기에는 백업 및 복원, Copy Database Wizard, 데이터베이스 복사 및 연결이 포함됩니다.

에픽

마이그레이션 계획

작업	설명	필요한 기술
소스 및 대상 데이터베이스 버전을 검증합니다.		DBA
대상 운영 체제 버전을 식별합니다.		DBA, SysAdmin
Microsoft SQL Server 호환성 목록 및 용량 요구 사항을 기반으로 대상 서버 인스턴스의 하드웨어 요구 사항을 확인합니다.		DBA, SysAdmin
유형 및 용량에 대한 스토리지 요구 사항을 확인합니다.		DBA, SysAdmin
지연 시간 및 대역폭을 포함한 네트워크 요구 사항을 확인합니다.		DBA, SysAdmin
용량, 스토리지 기능, 네트워크 기능에 따라 EC2 인스턴스 유형을 선택합니다.		DBA, SysAdmin
소스 및 대상 데이터베이스의 네트워크 및 호스트 액세스 보안 요구 사항을 확인합니다.		DBA, SysAdmin
Microsoft SQL Server 소프트웨어 설치에 필요한 사용자 목록을 확인합니다.		DBA, SysAdmin
백업 전략을 결정합니다.		DBA

작업	설명	필요한 기술
가용성 요구 사항을 결정합니다.		DBA
애플리케이션 마이그레이션 및 전환 전략을 확인합니다.		DBA, SysAdmin

인프라 구성

작업	설명	필요한 기술
Virtual Private Cloud(VPC) 및 서브넷을 생성합니다.		SysAdmin
보안 그룹 및 네트워크 액세스 제어 목록(ACL)을 생성합니다.		SysAdmin
EC2 인스턴스를 구성하고 시작합니다.		SysAdmin

소프트웨어를 설치합니다.

작업	설명	필요한 기술
Microsoft SQL Server 소프트웨어에 필요한 사용자 및 그룹을 생성합니다.		DBA, SysAdmin
Microsoft SQL Server 소프트웨어를 다운로드합니다.		DBA, SysAdmin
EC2 인스턴스에 Microsoft SQL Server 소프트웨어를 설치하고 서버를 구성합니다.		DBA, SysAdmin

데이터 마이그레이션하기 - 옵션 1

작업	설명	필요한 기술
기본 Microsoft SQL Server 도구 또는 타사 도구를 사용하여 데이터베이스 개체 및 데이터를 마이그레이션할 수 있습니다.	도구에는 백업 및 복원, Copy Database Wizard, 데이터베이스 복사 및 연결이 포함됩니다.	DBA

데이터 마이그레이션하기 - 옵션 2

작업	설명	필요한 기술
AWS DMS를 사용하여 데이터를 마이그레이션합니다.	AWS DMS 사용에 대한 자세한 내용은 참조 및 도움말 섹션의 링크를 참조하세요.	DBA

애플리케이션 마이그레이션

작업	설명	필요한 기술
애플리케이션 마이그레이션 전략을 따릅니다.	AWS Schema Conversion Tool(AWS SCT)을 사용하여 애플리케이션 소스 코드에 내장된 SQL 코드를 분석하고 수정할 수 있습니다.	DBA, 앱 소유자

전환

작업	설명	필요한 기술
애플리케이션 전환 전략을 따르세요.		DBA SysAdmin, 앱 소유자

프로젝트 닫기

작업	설명	필요한 기술
모든 임시 AWS 리소스를 종료합니다.	임시 리소스에는 AWS DMS 복제 인스턴스와 AWS SCT용 EC2 인스턴스가 포함됩니다.	DBA, SysAdmin
프로젝트 문서를 검토하고 검증하세요.		DBA SysAdmin, 앱 소유자
마이그레이션 시간, 수동 대비 도구 비용 절감 비율 등에 대한 지표를 모읍니다.		DBA SysAdmin, 앱 소유자
프로젝트를 종료하고 피드백을 제공합니다.		DBA SysAdmin, 앱 소유자

관련 리소스

참조

- [Amazon Web Services에 Microsoft SQL Server 배포](#)
- [Amazon EC2](#)
- [Amazon EC2 FAQ](#)
- [AWS Database Migration Service](#)
- [Amazon EC2 요금](#)
- [AWS의 Microsoft 제품](#)
- [AWS의 Microsoft 라이선싱](#)
- [AWS의 Microsoft SQL Server](#)

자습서 및 동영상

- [Amazon EC2 시작하기](#)
- [AWS Database Migration Service 시작하기](#)
- [디렉터리에 Amazon EC2 인스턴스 추가\(Simple AD 및 Microsoft AD\)](#)

- [AWS Database Migration Service\(동영상\)](#)
- [Amazon EC2 소개 - 엘라스틱 클라우드 서버 및 AWS로 호스팅하기\(동영상\)](#)

온프레미스 MySQL 데이터베이스를 Amazon EC2로 마이그레이션

작성자: Sergey Dmitriev(AWS)

R 유형: 리호스팅	소스: 데이터베이스: 관계형	대상: Amazon EC2의 MySQL
작성자: AWS	환경: PoC 또는 파일럿	기술: 데이터베이스, 마이그레이션
워크로드: 오픈 소스		

요약

이 패턴은 온프레미스 MySQL 데이터베이스를 Amazon Elastic Compute Cloud(Amazon EC2) 인스턴스의 MySQL 데이터베이스로 마이그레이션하기 위한 지침을 제공합니다. 이 패턴은 마이그레이션을 위해 AWS Database Migration Service(AWS DMS) 또는 네이티브 MySQL 도구(예: mysqldbcopy 및 mysqldump)를 사용하는 방법을 설명합니다.

사전 조건 및 제한 사항

사전 조건

- 활성 상태의 AWS 계정
- 온프레미스 데이터 센터의 MySQL 소스 데이터베이스

제품 버전

- MySQL 버전 5.5, 5.6 및 5.7
- Amazon EC2에서 지원하는 대상 운영 체제 목록은 [Amazon EC2 FAQ](#)를 참조하십시오.

아키텍처

소스 기술 스택

- 온프레미스 MySQL 데이터베이스

대상 기술 스택

- Amazon EC2의 MySQL 데이터베이스 인스턴스

데이터 마이그레이션 방법

- DMS
- 네이티브 MySQL 도구(mysqldbcop, mysqldump)

대상 아키텍처

AWS 데이터 마이그레이션 아키텍처

DMS 사용:

네이티브 MySQL 도구 사용:

도구

- AWS DMS - [AWS Database Migration Service\(DMS\)](#)는 여러 소스 및 대상 데이터베이스를 지원합니다. DMS에서 지원하는 MySQL 소스 및 대상 데이터베이스에 대한 내용은 [MySQL 호환 데이터베이스를 AWS로 마이그레이션하기](#) 항목을 참조하십시오. DMS에서 소스 데이터베이스를 지원하지 않는 경우 데이터를 마이그레이션하기 위한 다른 방법을 선택해야 합니다.
- 네이티브 MySQL 도구 - mysqldbcop 및 mysqldump

에픽

마이그레이션 계획

작업	설명	필요한 기술
소스 및 대상 데이터베이스 버전을 검증합니다.		DBA

작업	설명	필요한 기술
대상 운영 체제 버전을 식별합니다.		DBA, SysAdmin
MySQL 호환성 목록 및 용량 요구 사항을 기반으로 대상 서버 인스턴스의 하드웨어 요구 사항을 식별합니다.		DBA, SysAdmin
스토리지 요구 사항(스토리지 유형 및 용량)을 식별합니다.		DBA, SysAdmin
지연 시간 및 대역폭과 같은 네트워크 요구 사항을 식별합니다.		DBA, SysAdmin
용량, 스토리지 기능, 네트워크 기능에 따라 적절한 인스턴스 유형을 선택합니다.		DBA, SysAdmin
소스 및 대상 데이터베이스의 네트워크 또는 호스트 액세스 보안 요구 사항을 식별합니다.		DBA, SysAdmin
MySQL 소프트웨어 설치에 필요한 운영 체제 사용자의 목록을 식별합니다.		DBA, SysAdmin
백업 전략을 결정합니다.		DBA
가용성 요구 사항을 결정합니다.		DBA
애플리케이션 마이그레이션 또는 전환 전략을 식별합니다.		DBA, SysAdmin

인프라 구성

작업	설명	필요한 기술
Virtual Private Cloud(VPC) 및 서브넷을 생성합니다.		SysAdmin
보안 그룹 및 네트워크 액세스 제어 목록(ACL)을 생성합니다.		SysAdmin
EC2 인스턴스를 구성하고 시작합니다.		SysAdmin

MySQL 소프트웨어 설치

작업	설명	필요한 기술
MySQL 소프트웨어가 작동하는 데 필요한 운영 체제 사용자 및 그룹을 생성합니다.		DBA, SysAdmin
필요한 버전의 MySQL 소프트웨어를 다운로드합니다.		DBA, SysAdmin
EC2 인스턴스에 MySQL 소프트웨어를 설치하고 서버를 구성합니다.		DBA, SysAdmin

데이터 마이그레이션 - 옵션 1

작업	설명	필요한 기술
네이티브 MySQL 도구 또는 타사 도구를 사용하여 데이터베이스 객체 및 데이터를 마이그레이션합니다.	이러한 도구에는 mysqldbcopy 및 mysqldump가 포함됩니다.	DBA

데이터 마이그레이션 - 옵션 2

작업	설명	필요한 기술
AWS DMS를 사용하여 데이터를 마이그레이션합니다.		DBA

애플리케이션 마이그레이션

작업	설명	필요한 기술
애플리케이션 마이그레이션 전략을 따릅니다.		DBA SysAdmin, 앱 소유자

전환

작업	설명	필요한 기술
애플리케이션 컷오버 또는 전환 전략을 따릅니다.		DBA SysAdmin, 앱 소유자

프로젝트 닫기

작업	설명	필요한 기술
임시 AWS 리소스를 종료합니다.	DMS 복제 인스턴스를 종료합니다.	DBA, SysAdmin
프로젝트 문서를 검토하고 검증하세요.		DBA SysAdmin, 앱 소유자
마이그레이션 시간, 수동 대비 도구 비율(%), 비용 절감 등에 대한 지표를 수집합니다.		DBA SysAdmin, 앱 소유자

작업	설명	필요한 기술
프로젝트를 마무리하고 피드백을 제공하세요.		DBA SysAdmin, 앱 소유자

관련 리소스

참조

- [Amazon EC2 웹 사이트](#)
- [AWS DMS 웹사이트](#)
- [Amazon EC2 요금](#)
- [AWS DMS 단계별 안내](#)

자습서 및 동영상

- [DMS 시작](#)
- [Amazon EC2 소개 - AWS 기반 탄력적 클라우드 서버 및 호스팅\(동영상\)](#)

Application Migration Service를 사용하여 동종 SAP 마이그레이션 전환 시간 단축

작성자: 파벨 루빈(AWS), 디에고 발베르데(AWS), 수닐 야다브(AWS)

환경: 프로덕션	소스: 온프레미스 SAP ASE 데이터베이스	대상: Amazon EC2의 SAP 데이터베이스
R 유형: 리호스팅	워크로드: SAP	기술: 마이그레이션, 데이터베이스

AWS 서비스: AWS Application Migration Service, Amazon EBS

요약

이 패턴은 AWS Application Migration Service를 사용하여 SAP 워크로드를 마이그레이션하는 단계를 설명합니다. Application Migration Service는 블록 레벨 복제를 사용하여 소스와 지속적으로 동기화되는 복제 볼륨을 유지하여 전환이 쉽도록 합니다.

SAP 워크로드에는 애플리케이션 SAP CRM(Customer Relationship Management), SAP ERP(Enterprise Resource Planning) 및 SAP BW(Business Warehouse)가 포함됩니다.

사전 조건 및 제한 사항

사전 조건

- 소스 SAP 서버와 AWS의 대상 Virtual Private Cloud(VPC) 간에 안정적 네트워크 연결이 가능한 활성 AWS 계정
- 온프레미스 데이터 센터의 Linux 또는 Windows용 SAP Adaptive Server Enterprise(ASE) 소스 데이터베이스

제한 사항

- Amazon Elastic Compute Cloud(Amazon EC2)가 대상 운영 체제를 지원해야 합니다. 자세한 내용은 [Amazon EC2 FAQ](#)를 참조하세요.

아키텍처

소스 기술 스택

- SAP ASE 데이터베이스

대상 기술 스택

- Amazon EC2
- Amazon Elastic Block Store(Amazon EBS)

소스 및 대상 아키텍처

다음 다이어그램은 온프레미스 서버에서 Replication Agent를 통해 Application Migration Service 엔드 포인트로 마이그레이션하는 과정을 보여줍니다. Amazon Simple Storage Service(Amazon S3) 엔드 포인트는 설치 및 구성 파일에 액세스하는 데 사용됩니다. 스테이징 영역의 서브넷과 마이그레이션된 리소스에는 EBS 볼륨에 데이터를 저장하는 EC2 인스턴스가 포함되어 있습니다. 포트 TCP 443은 소스 시스템 네트워크를 Application Migration Service에 연결하고 스테이징 영역 서브넷을 Application Migration Service, Amazon EC2 및 Amazon S3 리전 엔드포인트에 연결하는 데 사용됩니다. 포트 TCP 1500은 로컬 네트워크와 스테이징 영역 간의 데이터 복제에 사용됩니다.

도구

- [AWS 애플리케이션 마이그레이션 서비스](#)를 사용하면 변경 없이 다운타임을 최소화하면서 애플리케이션을 AWS 클라우드로 재호스팅 (lift-and-shift) 할 수 있습니다.
- [Amazon Elastic Block Store\(Amazon EBS\)](#)는 Amazon Elastic Compute Cloud(Amazon EC2) 인스턴스와 함께 사용할 수 있는 블록 스토리지 볼륨을 제공합니다.
- [Amazon Elastic Compute Cloud\(Amazon EC2\)](#)는 AWS 클라우드에서 확장 가능한 컴퓨팅 용량을 제공합니다. 필요한 만큼 가상 서버를 시작하고 빠르게 스케일 업하거나 스케일 다운할 수 있습니다.
- [Amazon Simple Storage Service\(S3\)](#)는 원하는 양의 데이터를 저장, 보호 및 검색하는 데 도움이 되는 클라우드 기반 객체 스토리지 서비스입니다.
- [AWS Security Token Service\(AWS STS\)](#)를 사용하면 사용자를 위한 제한된 권한의 임시 보안 인증 정보를 요청할 수 있습니다.

에픽

Application Migration Service 초기화

작업	설명	필요한 기술
Application Migration Service를 초기화합니다.	SAP ASE 데이터베이스를 배포할 리전에서 Application Migration Service를 초기화합니다. AWS는 각 리전의 Application Migration Service 페이지로 처음 이동할 때 자동 설정을 제공합니다.	AWS 관리자
서비스 역할을 수동으로 생성합니다.	(선택 사항) 자동화(예: AWS Control Tower)를 사용하여 계정을 설정하려는 경우 설치, 복제 및 시작에 필요한 6개의 AWS Identity and Access Management(IAM) 역할을 수동으로 생성할 수 있습니다. 지침은 AWS 설명서 를 참조하세요.	AWS 관리자
복제 설정 템플릿을 생성합니다.	복제 설정 템플릿은 서브넷, 인스턴스 유형, Amazon EBS 암호화 및 데이터 라우팅 방식을 정의합니다. 자세한 설정 정보는 AWS 설명서 를 참조하세요.	일반 AWS

에이전트 설치를 위한 보안 인증 생성

작업	설명	필요한 기술
새 IAM 역할을 생성합니다.	IAM 콘솔에서 역할로 이동한 후 역할 생성을 선택합니다.	AWS 시스템 관리자

작업	설명	필요한 기술
	신뢰할 수 있는 엔터티 유형의 경우 AWS 계정을 선택한 후 다음을 선택합니다.	
IAM AWSApplicationMigrationAgentPolicy 역할에 연결하세요.	AWS 관리형 AWSApplicationMigrationAgentPolicy 정책에는 Application Migration Service Agent 설치를 수행하는 데 필요한 권한이 포함되어 있습니다. 정책을 연결한 후 다음을 선택합니다.	AWS 시스템 관리자
역할 생성을 완료합니다.	친숙한 이름을 지정하고 역할 생성을 선택합니다.	AWS 시스템 관리자
임시 보안 인증 정보를 생성합니다.	액세스 키 ID, 비밀 액세스 키 및 세션 토큰을 생성하려면 AWS STS 설명서 의 지침을 따르세요. 이러한 보안 인증은 에이전트 설치 중에 사용됩니다.	AWS 시스템 관리자

SAP 소스 시스템에 Application Migration Service Agent를 설치합니다.

작업	설명	필요한 기술
SAP 소스 시스템에 에이전트 설치 프로그램을 다운로드합니다.	소스 운영 체제에 적합한 에이전트 설치 프로그램(Windows 또는 Linux)을 다운로드합니다.	앱 소유자
AWS Replication Agent를 설치합니다.	소스 시스템에서 에이전트 설치 관리자 파일을 실행하면 먼저 액세스 키, 비밀 액세스 키, 세션 토큰, 복제할 리전을 입력하라는 메시지가 표시됩니다.	앱 소유자

작업	설명	필요한 기술
	이전에 생성한 IAM 역할의 임시 보안 인증 정보와 초기화 중에 구성된 동일한 리전을 사용합니다.	
초기 데이터 복제를 기다립니다.	에이전트가 설치되면 Application Migration Service 콘솔의 시스템 탭에 소스 시스템이 나타납니다.	앱 소유자

대상 머신의 Launch 템플릿을 구성합니다.

작업	설명	필요한 기술
소스 서버의 Launch 템플릿을 업데이트합니다.	각 소스 서버는 대상 EC2 서버의 구성을 알려주는 고유한 EC2 Launch 템플릿을 사용합니다. 마이그레이션된 서버의 Amazon EC2 구성을 사용자 지정하려는 경우 이 템플릿을 편집할 수 있습니다.	일반 AWS
기본 Launch 템플릿 버전을 설정합니다.	Launch 템플릿을 필요에 따라 변경한 후 이 업데이트된 버전을 기본 Launch 템플릿으로 사용하도록 지정합니다. 자세한 내용은 AWS 설명서 를 참조하세요.	일반 AWS
인스턴스 유형 적정 크기 조절을 끕니다.	(선택 사항) 인스턴스 유형 적정 크기 조정 은 소스 SAP 서버의 구성을 기반으로 자동 인스턴스 유형 권장 사항을 제공합니다. Launch 템플릿에서 사용자 지정 인스턴스 유형을 지정	일반 AWS

작업	설명	필요한 기술
	할 수 있도록 이 설정을 끄는 것이 좋습니다.	

테스트 수행

작업	설명	필요한 기술
테스트 시작을 시작하세요.	Application Migration Service 콘솔에서 서버를 하나 이상 선택한 다음 테스트 및 전환에서 테스트 인스턴스 시작을 선택합니다.	일반 AWS, 마이그레이션 엔지니어, 마이그레이션 책임자
전환 및 시작 프로세스가 완료 될 때까지 기다립니다.	시작 이력 탭에서 시작 프로세스를 검토할 수 있습니다. 시스템이 EC2 인스턴스를 성공적으로 시작하면 알림 탭이 시작됨으로 업데이트됩니다.	
테스트가 성공적으로 완료되었는지 확인합니다.	RDP(원격 데스크톱 프로토콜) 또는 SSH(보안 셸)을 통해 시작된 인스턴스에 연결하고 적절한 애플리케이션 검사를 수행합니다. 예를 들어, SAP 인터페이스에 로그인하여 기능을 검증합니다.	마이그레이션 엔지니어, 앱 소유자
소스 수명 주기를 업데이트합니다.	테스트가 성공적이면 테스트 및 전환 탭에서 소스 시스템 수명 주기를 “전환 준비 완료”로 표시로 업데이트합니다.	마이그레이션 엔지니어, 마이그레이션 책임자

Amazon EC2 대상에 대한 전환 예약 및 수행

작업	설명	필요한 기술
전환 기간을 예약합니다.		전환 리드, 마이그레이션 책임자, 앱 소유자
전환 시작을 착수합니다.	하나 이상의 서버를 선택합니다. Application Migration Service 콘솔의 테스트 및 전환 탭에서 테스트 및 전환의 전환 인스턴스 시작을 선택합니다.	마이그레이션 엔지니어
전환 및 시작 프로세스가 완료 될 때까지 기다립니다.	시작 이력 탭에서 시작 프로세스를 검토할 수 있습니다. 시스템이 EC2 인스턴스를 성공적으로 시작하면 알림 탭이 시작됨으로 업데이트됩니다.	
전환이 성공적으로 완료되었는지 확인합니다.	RDP 또는 SSH를 통해 시작된 인스턴스에 연결하고 적절한 애플리케이션 검사를 수행합니다.	앱 소유자, 마이그레이션 엔지니어
소스 수명 주기를 업데이트합니다.	전환이 성공적이면 테스트 및 전환 탭에서 전환 마무리를 선택하여 소스 시스템 수명 주기를 업데이트합니다.	마이그레이션 엔지니어

관련 리소스

참조

- [AWS Application Migration Service](#)
- [AWS Application Migration FAQ](#)

동영상

- [AWS Application Migration Service 아키텍처](#)

AWS 클라우드의 온프레미스 워크로드 리호스팅: 마이그레이션 체크리스트

작성자: Srikanth Rangavajhala(AWS)

환경: PoC 또는 파일럿	소스: 온프레미스 워크로드	대상: AWS 클라우드
R 유형: 리호스팅	기술: 마이그레이션	AWS 서비스: AWS 애플리케이션 마이그레이션 서비스

요약

Amazon Web Services(AWS) 클라우드에서 온프레미스 워크로드를 재호스팅하려면 계획, 사전 검색, 검색, 구축, 테스트 및 전환과 같은 마이그레이션 단계가 포함됩니다. 이 패턴은 단계 및 관련 작업을 간략하게 설명합니다. 작업은 개괄적으로 설명되며 전체 애플리케이션 워크로드의 약 75% 를 지원합니다. 애자일 스프린트 주기로 2~3주에 걸쳐 이러한 작업을 구현할 수 있습니다.

마이그레이션 팀 및 컨설턴트와 함께 이러한 작업을 검토하고 검토해야 합니다. 검토 후에는 의견을 수집하고, 요구 사항을 충족하는 데 필요한 작업을 제거하거나 재평가하고, 포트폴리오에 있는 애플리케이션 워크로드의 최소 75%를 지원하도록 기타 작업을 수정할 수 있습니다. 그런 다음 Atlassian Jira 또는 Rally Software와 같은 애자일 프로젝트 관리 도구를 사용하여 작업을 가져오고, 리소스에 할당하고, 마이그레이션 활동을 추적할 수 있습니다.

이 패턴은 [AWS 클라우드 마이그레이션 팩토리](#)를 사용하여 워크로드를 리호스팅한다고 가정하지만 원하는 마이그레이션 도구를 사용할 수 있습니다.

사전 조건 및 제한 사항

사전 조건

- 마이그레이션 작업을 추적하기 위한 프로젝트 관리 도구(예를 들어, Atlassian Jira 또는 Rally 소프트웨어)
- AWS에서 워크로드를 리호스팅하기 위한 마이그레이션 도구(예를 들어, [클라우드 마이그레이션 팩토리](#))

아키텍처

소스 플랫폼

- 온프레미스 소스 스택(기술, 애플리케이션, 데이터베이스, 인프라 포함)

대상 플랫폼

- AWS 클라우드 대상 스택(기술, 애플리케이션, 데이터베이스, 인프라 포함)

아키텍처

다음 다이어그램은 클라우드 마이그레이션 팩토리와 AWS 애플리케이션 마이그레이션 서비스를 사용하여 서버를 리호스팅(온프레미스 소스 환경에서 AWS로 서버를 검색하고 마이그레이션)하는 것을 보여줍니다.

도구

- 원하는 마이그레이션 및 프로젝트 관리 도구를 사용할 수 있습니다.

에픽

계획 단계

작업	설명	필요한 기술
사전 검색 백로그를 정리하십시오.	부서장 및 애플리케이션 소유자와 함께 사전 검색 백로그 정리 작업 세션을 진행하십시오.	프로젝트 매니저, 애자일 스크럼 리더
스프린트 계획 작업 세션을 진행하십시오.	범위 지정 연습으로, 마이그레이션하려는 애플리케이션을 스프린트와 웨이브에 배포하십시오.	프로젝트 매니저, 애자일 스크럼 리더

사전 검색 단계

작업	설명	필요한 기술
애플리케이션 지식을 확인하십시오.	애플리케이션 소유자와 애플리케이션에 대한 지식을 확인하고 문서화하십시오. 기술적인 질문을 담당할 담당자가 더 있는지 확인하십시오.	마이그레이션 전문가(면접관)
애플리케이션 규정 준수 요구 사항을 결정하십시오.	애플리케이션이 결제 카드 산업 데이터 보안 표준 (PCI DSS), Sarbanes-Oxley Act(SOX), 개인 식별 정보(PII) 또는 기타 표준의 요구 사항을 준수할 필요가 있는지 애플리케이션 소유자에게 확인하십시오. 규정 준수 요구 사항이 있는 경우 팀은 마이그레이션할 서버에서 규정 준수 검사를 완료해야 합니다.	마이그레이션 전문가(면접관)
프로덕션 릴리스 요구 사항을 확인하십시오.	마이그레이션된 애플리케이션을 프로덕션으로 릴리스하기 위한 요구 사항 (릴리스 날짜 및 다운타임 기간 등)은 애플리케이션 소유자 또는 기술 담당자에게 확인하십시오.	마이그레이션 전문가(면접관)
서버 목록 가져오기	대상 애플리케이션과 연결된 서버 목록을 가져옵니다.	마이그레이션 전문가(면접관)
현재 상태를 보여주는 논리적 다이어그램을 구하십시오.	엔터프라이즈 아키텍트나 애플리케이션 소유자로부터 애플리케이션의 현재 상태 다이어그램을 구하십시오.	마이그레이션 전문가(면접관)

작업	설명	필요한 기술
대상 상태를 보여주는 논리적 다이어그램을 만드십시오.	AWS의 대상 아키텍처를 보여주는 애플리케이션의 논리적 다이어그램을 작성하십시오. 이 다이어그램은 서버, 연결 및 매핑 요소를 설명해야 합니다.	엔터프라이즈 아키텍트, 비즈니스 오너
서버 정보 가져오기	구성 세부 정보를 포함하여 애플리케이션과 연결된 서버에 대한 정보를 수집합니다.	마이그레이션 전문가(면접관)
검색 템플릿에 서버 정보를 추가합니다.	애플리케이션 검색 템플릿에 자세한 서버 정보를 추가합니다 (이 패턴의 첨부 파일 <code>mobilize-application-questionnaire.xlsx</code> 참조). 이 템플릿에는 모든 애플리케이션 관련 보안, 인프라, 운영 체제 및 네트워킹 세부 정보가 포함됩니다.	마이그레이션 전문가(면접관)
애플리케이션 검색 템플릿을 게시합니다.	애플리케이션 검색 템플릿을 애플리케이션 소유자 및 마이그레이션 팀과 공유하여 공통적으로 액세스하고 사용할 수 있도록 하십시오.	마이그레이션 전문가(면접관)

검색 단계

작업	설명	필요한 기술
서버 목록을 확인하십시오.	애플리케이션 소유자 또는 기술 책임자에게 서버 목록과 각 서버의 용도를 확인하십시오.	마이그레이션 체크리스트

작업	설명	필요한 기술
<p>서버 그룹을 식별하고 추가합니다.</p>	<p>웹 서버 또는 애플리케이션 서버와 같은 서버 그룹을 식별하고 이 정보를 애플리케이션 검색 템플릿에 추가합니다. 각 서버가 속해야 하는 애플리케이션(웹, 애플리케이션, 데이터베이스)의 계층을 선택합니다.</p>	<p>마이그레이션 체크리스트</p>
<p>애플리케이션 검색 템플릿을 작성합니다.</p>	<p>마이그레이션 팀, 애플리케이션 팀, AWS의 도움을 받아 애플리케이션 검색 템플릿의 세부 정보를 작성하십시오.</p>	<p>마이그레이션 체크리스트</p>
<p>누락된 서버 세부 정보(미들웨어 및 OS 팀)를 추가합니다.</p>	<p>미들웨어 및 운영 체제 (OS) 팀에 애플리케이션 검색 템플릿을 검토하고 데이터베이스 정보를 포함하여 누락된 서버 세부 정보를 추가하도록 요청하십시오.</p>	<p>마이그레이션 체크리스트</p>
<p>인바운드 및 아웃바운드 트래픽 규칙을 구합니다(네트워크 팀).</p>	<p>네트워크 팀에 요청하여 소스 및 대상 서버의 인바운드/아웃바운드 트래픽 규칙을 구하십시오. 또한 네트워크 팀은 기존 방화벽 규칙을 추가하고, 보안 그룹 형식으로 내보내고, 기존 로드 밸런서를 애플리케이션 검색 템플릿에 추가해야 합니다.</p>	<p>마이그레이션 체크리스트</p>
<p>필요한 태깅을 식별하십시오.</p>	<p>애플리케이션의 태깅 요구 사항을 결정하십시오.</p>	<p>마이그레이션 체크리스트</p>
<p>방화벽 요청 세부 정보를 생성합니다.</p>	<p>애플리케이션과 통신하는 데 필요한 방화벽 규칙을 캡처하고 필터링합니다.</p>	<p>마이그레이션 전문가, 솔루션 아키텍트, 네트워크 책임자</p>

작업	설명	필요한 기술
EC2 인스턴스 유형을 업데이트합니다.	인프라 및 서버 요구 사항에 따라 대상 환경에서 사용할 Amazon Elastic Compute Cloud(Amazon EC2) 인스턴스 유형을 업데이트합니다.	마이그레이션 전문가, 솔루션 아키텍트, 네트워크 책임자
현재 상태 다이어그램을 식별하십시오.	애플리케이션의 현재 상태를 보여주는 다이어그램을 식별하거나 생성하십시오. 이 다이어그램은 정보 보안 (InfoSec) 요청에 사용됩니다.	마이그레이션 전문가, 솔루션 아키텍트
미래 상태 다이어그램을 완성하십시오.	애플리케이션의 미래(대상) 상태를 보여주는 다이어그램을 완성하십시오. 이 다이어그램은 InfoSec 요청에도 사용됩니다.	마이그레이션 전문가, 솔루션 아키텍트
방화벽 또는 보안 그룹 서비스 요청을 생성합니다.	방화벽 또는 보안 그룹 서비스 요청(개발 및 QA, 사전 제작 및 프로덕션용)을 생성합니다. 클라우드 마이그레이션 팩토리를 사용하는 경우 복제 전용 포트가 아직 열려 있지 않다면 해당 포트를 포함하십시오.	마이그레이션 전문가, 솔루션 아키텍트, 네트워크 책임자
방화벽 또는 보안 그룹 요청 (InfoSec 팀) 을 검토하십시오.	이 단계에서 InfoSec 팀은 이전 단계에서 만든 방화벽 또는 보안 그룹 요청을 검토하고 승인합니다.	InfoSec 엔지니어, 마이그레이션 전문가
방화벽 보안 그룹 요청을 구현합니다(네트워크 팀).	InfoSec 팀이 방화벽 요청을 승인한 후 네트워크 팀은 필요한 인바운드/아웃바운드 방화벽 규칙을 구현합니다.	마이그레이션 전문가, 솔루션 아키텍트, 네트워크 책임자

구축 단계(개발 및 QA, 사전 프로덕션 및 프로덕션 환경에서 반복)

작업	설명	필요한 기술
애플리케이션 및 서버 데이터를 가져옵니다.	<ol style="list-style-type: none"> 범위 내 소스 서버에서 로컬 관리자 권한을 가진 도메인 사용자로 마이그레이션 실행 서버에 로그인했는지 확인하십시오. 마이그레이션 접속 양식을 사용하여 범위 내 소스 서버의 속성을 가져옵니다. 자세한 내용은 Cloud Migration Factory 구현 설명서를 참조하십시오. <p>클라우드 마이그레이션 팩토리를 사용하지 않는 경우 마이그레이션 도구 설정 지침을 따르십시오.</p>	마이그레이션 전문가, 클라우드 관리자
소스 서버의 사전 조건을 확인합니다.	범위 내 소스 서버에 연결하여 TCP 포트 1500, TCP 포트 443, 루트 볼륨 여유 스페이스, .NET 프레임워크 버전 및 기타 파라미터와 같은 사전 조건을 확인합니다. 이는 복제에 필요합니다. 자세한 내용은 Cloud Migration Factory 구현 설명서 를 참조하십시오.	마이그레이션 전문가, 클라우드 관리자
복제 에이전트를 설치하기 위한 서비스 요청을 생성합니다.	개발 및 QA, 사전 프로덕션 또는 프로덕션을 위해 범위 내 서버에 복제 에이전트를 설치하기 위한 서비스 요청을 생성합니다.	마이그레이션 전문가, 클라우드 관리자

작업	설명	필요한 기술
복제 에이전트를 설치합니다.	개발 및 QA, 사전 프로덕션 또는 프로덕션 머신의 범위 내 소스 서버에 복제 에이전트를 설치합니다. 자세한 내용은 Cloud Migration Factory 구현 설명서 를 참조하십시오.	마이그레이션 전문가, 클라우드 관리자
시작 후 스크립트를 푸시합니다.	애플리케이션 마이그레이션 서비스는 시작 후 스크립트를 지원하여 대상 인스턴스를 시작한 후 소프트웨어 설치 또는 제거와 같은 OS 수준 작업을 자동화하는 데 도움이 됩니다. 이 단계에서는 마이그레이션을 위해 식별된 서버에 따라 실행 후 스크립트를 Windows 또는 Linux 머신에 푸시합니다. 자세한 지침은 Cloud Migration Factory 구현 설명서 를 참고하십시오.	마이그레이션 전문가, 클라우드 관리자
복제 상태를 확인하십시오.	제공된 스크립트를 사용하여 범위 내 소스 서버의 복제 상태를 자동으로 확인합니다. 스크립트는 지정된 웨이브의 모든 소스 서버 상태가 정상으로 변경될 때까지 5분마다 반복됩니다. 자세한 지침은 Cloud Migration Factory 구현 설명서 를 참고하십시오.	마이그레이션 전문가, 클라우드 관리자

작업	설명	필요한 기술
관리자 사용자를 생성합니다.	범위 내 소스 서버에서 AWS로 마이그레이션을 전환한 후 문제를 해결하려면 소스 머신의 로컬 관리자 또는 sudo 사용자가 필요할 수 있습니다. 마이그레이션 팀은 인증 서버(예를 들어, DC 또는 LDAP 서버)에 연결할 수 없을 때 이 사용자를 사용하여 대상 서버에 로그인합니다. 이 단계에 대한 자세한 지침은 Cloud Migration Factory 구현 설명서 를 참고하십시오.	마이그레이션 전문가, 클라우드 관리자
시작 템플릿을 검증합니다.	서버 메타데이터를 검증하여 제대로 작동하고 잘못된 데이터가 없는지 확인합니다. 이 단계에서는 메타데이터의 테스트와 전환을 모두 검증합니다. 자세한 지침은 Cloud Migration Factory 구현 설명서 를 참고하십시오.	마이그레이션 전문가, 클라우드 관리자

테스트 단계(개발 및 QA, 사전 프로덕션 및 프로덕션 환경에서 반복)

작업	설명	필요한 기술
서비스 요청을 생성하십시오.	인프라 팀 및 기타 팀이 개발 및 QA, 사전 프로덕션 또는 프로덕션 인스턴스로 애플리케이션 전환을 수행할 수 있도록 서비스 요청을 생성합니다.	마이그레이션 전문가, 클라우드 관리자
로드 밸런서를 구성합니다(선택 사항).	Application Load Balancer 또는 iRules가 있는 F5 로드 밸런	마이그레이션 전문가, 클라우드 관리자

작업	설명	필요한 기술
	<p>서와 같은 필수 로드 밸런서를 구성합니다.</p>	
<p>테스트를 위해 인스턴스를 시작합니다.</p>	<p>테스트 모드에서 애플리케이션 마이그레이션 서비스에서 지정된 웨이브에 대한 모든 대상 머신을 실행합니다. 자세한 내용은 Cloud Migration Factory 구현 설명서를 참조하십시오.</p>	<p>마이그레이션 전문가, 클라우드 관리자</p>
<p>대상 인스턴스 상태를 확인합니다.</p>	<p>범위 내 모든 소스 서버의 부팅 프로세스를 동일한 웨이브로 확인하여 대상 인스턴스의 상태를 확인합니다. 대상 인스턴스가 부팅되는 데는 최대 30분이 소요될 수 있습니다. Amazon EC2 콘솔에 로그인하여 서버 이름을 검색하고 열의 상태 확인을 통해 상태를 수동으로 확인할 수 있습니다. 2/2 검사 통과 상태는 인프라 관점에서 인스턴스가 정상임을 나타냅니다.</p>	<p>마이그레이션 전문가, 클라우드 관리자</p>

작업	설명	필요한 기술
DNS 항목을 수정하십시오.	<p>도메인 이름 시스템(DNS) 항목을 수정합니다. (resolv.conf 또는 host.conf 를 Microsoft Windows 환경용으로 사용하십시오.) 각 EC2 인스턴스가 이 호스트의 새 IP 주소를 가리키도록 구성합니다.</p> <p>참고: 온프레미스와 AWS 클라우드 서버 간에 DNS 충돌이 없는지 확인하십시오. 서버가 호스팅되는 환경에 따라 이 단계와 다음 단계는 선택 사항입니다.</p>	마이그레이션 전문가, 클라우드 관리자
EC2 인스턴스에서 백엔드 호스트에 대한 연결을 테스트합니다.	마이그레이션된 서버의 도메인 보안 인증을 사용하여 로그인을 확인합니다.	마이그레이션 전문가, 클라우드 관리자
DNS A 레코드를 업데이트합니다.	새 Amazon EC2 프라이빗 IP 주소를 가리키도록 각 호스트의 DNS A 레코드를 업데이트합니다.	마이그레이션 전문가, 클라우드 관리자
DNS CNAME 레코드를 업데이트합니다.	웹 및 애플리케이션 서버의 클러스터를 가리키도록 가상 IP(로드 밸런서 이름)에 대한 DNS CNAME 레코드를 업데이트합니다.	마이그레이션 전문가, 클라우드 관리자
적용 가능한 환경에서 애플리케이션을 테스트하십시오.	새 EC2 인스턴스에 로그인하고 개발 및 QA, 사전 프로덕션 및 프로덕션 환경에서 애플리케이션을 테스트합니다.	마이그레이션 전문가, 클라우드 관리자

작업	설명	필요한 기술
전환 준비가 된 것으로 표시하십시오.	테스트가 완료되면 소스 서버의 상태를 전환할 준비가 되었음을 나타내도록 변경하여 사용자가 전환 인스턴스를 시작할 수 있도록 하십시오. 자세한 지침은 Cloud Migration Factory 구현 설명서 를 참고하십시오.	마이그레이션 전문가, 클라우드 관리자

전환 단계

작업	설명	필요한 기술
프로덕션 배포 계획을 세우십시오.	프로덕션 배포 계획(백아웃 계획 포함)을 만드십시오.	마이그레이션 전문가, 클라우드 관리자
운영 팀에 가동 중지 시간을 알리십시오.	해당 운영 팀에 서버 가동 중지 일정을 알리십시오. 일부 팀에서는 이 알림을 받기 위해 변경 요청 또는 서비스 요청(CR/SR) 티켓이 필요할 수 있습니다.	마이그레이션 전문가, 클라우드 관리자
프로덕션 머신을 복제하십시오.	애플리케이션 마이그레이션 서비스 또는 다른 마이그레이션 도구를 사용하여 프로덕션 머신을 복제합니다.	마이그레이션 전문가, 클라우드 관리자
범위 내 소스 서버를 종료하십시오.	소스 서버의 복제 상태를 확인했으면 소스 서버를 종료하여 클라이언트 애플리케이션에서 서버로의 트랜잭션을 중지할 수 있습니다. 전환 창에서 소스 서버를 종료할 수 있습니다. 자세한 내용은 Cloud Migration	클라우드 관리자

작업	설명	필요한 기술
	Factory 구현 설명서 를 참조하십시오.	
전환을 위해 인스턴스를 시작합니다.	전환 모드에서 애플리케이션 마이그레이션 서비스에서 지정된 웨이브에 대한 모든 대상 머신을 실행합니다. 자세한 내용은 Cloud Migration Factory 구현 설명서 를 참조하십시오.	마이그레이션 전문가, 클라우드 관리자
대상 인스턴스 IP를 검색합니다.	대상 인스턴스의 IP를 검색합니다. DNS 업데이트가 환경에서 수동 프로세스인 경우 모든 대상 인스턴스에 대해 새 IP 주소를 가져와야 합니다. 자세한 내용은 Cloud Migration Factory 구현 설명서 를 참조하십시오.	마이그레이션 전문가, 클라우드 관리자
대상 서버 연결을 확인합니다.	DNS 레코드를 업데이트한 후 호스트 이름을 사용하여 대상 인스턴스에 연결하여 연결을 확인합니다. 자세한 내용은 Cloud Migration Factory 구현 설명서 를 참조하십시오.	마이그레이션 전문가, 클라우드 관리자

관련 리소스

참조

- [마이그레이션 방법](#)
- [AWS Cloud Migration Factory 구현 설명서](#)
- [클라우드 마이그레이션 팩토리를 통한 대규모 서버 마이그레이션 자동화](#)
- [AWS Application Migration Service 사용 설명서](#)
- [AWS Migration Acceleration Program](#)

첨부

이 문서와 관련된 추가 콘텐츠에 액세스하려면 [attachment.zip](#) 파일의 압축을 풉니다.

Amazon FSx를 사용하여 SQL Server Always On FCI용 다중 AZ 인프라 설정

작성자: Manish Garg(AWS), T.V.R.L.Phani Kumar Dadi(AWS), Nishad Mankar(AWS), RAJNEESH TYAGI(AWS)

코드 리포지토리: - 자동화 aws-windows-failover-cluster	환경: PoC 또는 파일럿	소스: 온프레미스 SQL Server 데이터베이스
대상: EC2 기반 Microsoft SQL Server	R 유형: 리호스팅	워크로드: Microsoft
기술: 마이그레이션, 인프라, DevOps	AWS 서비스: AWS Managed Microsoft AD, Amazon EC2, Amazon FSx, AWS Systems Manager	

요약

많은 Microsoft SQL Server Always On 장애 조치 클러스터 인스턴스(FCI)를 빠르게 마이그레이션해야 하는 경우 이 패턴을 이용하면 프로비저닝 시간을 최소화할 수 있습니다. 자동화 및 Amazon FSx for Windows File Server를 이용하면 많은 클러스터를 배포하는 데 필요한 수동 작업, 사람이 저지르는 오류 및 시간을 줄일 수 있습니다.

이 패턴은 Amazon Web Services(AWS)의 다중 가용 영역(다중 AZ) 배포에서 SQL Server FCI의 인프라를 설정합니다. 이 인프라에 필요한 AWS 서비스 프로비저닝은 [AWS CloudFormation](#) 템플릿을 사용하여 자동화됩니다. [Amazon Elastic Compute Cloud \(Amazon EC2\)](#) 인스턴스에서의 SQL Server 설치 및 클러스터 노드 생성은 명령을 사용하여 수행됩니다. PowerShell

이 솔루션은 가용성이 높은 다중 AZ [Amazon FSx for Windows](#) 파일 시스템을 SQL Server 데이터베이스 파일을 저장하기 위한 공용 감시 시스템으로 사용합니다. SQL Server를 호스팅하는 Amazon FSx 파일 시스템 및 EC2 Windows 인스턴스는 동일한 AWS Directory Service for Microsoft Active Directory(AWS Managed Microsoft AD) 도메인에 조인됩니다.

사전 조건 및 제한 사항

사전 조건

- 활성 상태의 AWS 계정
- AWS CloudFormation 템플릿을 사용하여 리소스를 프로비저닝할 수 있는 충분한 권한을 가진 AWS 사용자
- Microsoft Active Directory용 AWS Directory Service
- 키-값 페어로 AWS Managed Microsoft AD에 인증하기 위한 AWS Secrets Manager의 보안 인증 정보:
 - ADDomainName: <도메인 이름>
 - ADDomainJoinUserName: <도메인 사용자 이름>
 - ADDomainJoinPassword: <도메인 사용자 비밀번호>
 - TargetOU: <대상 OU 값>

참고: AWS Managed Microsoft AD 조인 활동에는 AWS Systems Manager Automation과 동일한 키 이름을 사용하게 됩니다.

- SQL Server 설치용 SQL Server 미디어 파일과 생성된 Windows 서비스 또는 도메인 계정(클러스터 생성 시 사용됨)
- 별도의 가용 영역에 있는 두 공개 서브넷, 가용 영역에 있는 두 비공개 서브넷, 인터넷 게이트웨이, NAT 게이트웨이, 라우팅 테이블 연결, 점프 서버를 포함하는 Virtual Private Cloud(VPC)

제품 버전

- Windows Server 2012 R2와 Microsoft SQL Server 2016

아키텍처

소스 기술 스택

- 공용 드라이브를 사용하는 온프레미스 SQL Server

대상 기술 스택

- AWS EC2 인스턴스
- Amazon FSx for Windows File Server
- AWS Systems Manager Automation 런북
- 네트워크 구성(VPC, 서브넷, 인터넷 게이트웨이, NAT 게이트웨이, 점프 서버, 보안 그룹)

- AWS Secrets Manager
- AWS Managed Microsoft AD
- 아마존 EventBridge
- AWS Identity and Access Management(IAM)

대상 아키텍처

다음 다이어그램은 가용 영역 두 개, NAT 게이트웨이가 있는 공개 서브넷 두 개, 첫 번째 공개 서브넷의 점프 서버 한 개, 비공개 서브넷 두 개(각각 노드 보안 그룹의 SQL Server 노드용 EC2 인스턴스 포함), 각 SQL Server 노드에 연결되는 Amazon FSx 파일 시스템을 포함하는 VPC가 있는 단일 AWS 리전의 AWS 계정을 보여줍니다. AWS 디렉토리 서비스, 아마존 EventBridge, AWS Secrets Manager 및 AWS Systems Manager도 포함됩니다.

자동화 및 규모 조정

- AWS Systems Manager를 사용하여 AWS Managed Microsoft AD에 조인하고 SQL 서버 설치를 수행할 수 있습니다.

도구

서비스

- [AWS는 AWS CloudFormation](#) 리소스를 설정하고, 빠르고 일관되게 프로비저닝하고, AWS 계정 및 지역 전반의 수명 주기 전반에 걸쳐 리소스를 관리할 수 있도록 지원합니다.
- [AWS Directory Service](#)는 Amazon Elastic Compute Cloud(Amazon EC2), Amazon Relational Database Service(RDS) for SQL Server, Amazon FSx for Windows File Server 등의 기타 AWS Service와 함께 Microsoft Active Directory(AD)를 사용할 수 있는 다양한 방법을 제공합니다.
- [Amazon Elastic Compute Cloud\(Amazon EC2\)](#)는 AWS 클라우드에서 확장 가능한 컴퓨팅 용량을 제공합니다. 필요한 만큼 가상 서버를 시작하고 빠르게 스케일 업하거나 스케일 다운할 수 있습니다.
- [EventBridgeAmazon](#)은 다양한 소스의 실시간 데이터와 애플리케이션을 연결하는 데 도움이 되는 서버리스 이벤트 버스 서비스입니다. AWS Lambda 함수, API 대상을 사용하는 HTTP 간접 호출 엔드포인트 또는 다른 AWS 계정의 이벤트 버스를 예로 들 수 있습니다.
- [AWS Identity and Access Management\(IAM\)](#)를 이용하면 사용자에 대해 인증 및 권한 부여를 제어함으로써 AWS 리소스에 대한 액세스를 안전하게 관리할 수 있습니다.

- [AWS Secrets Manager](#)를 사용하면 코드에 하드코딩된 보안 인증 정보(암호 등)를 Secrets Manager에 대한 API 직접 호출로 바꾸어 프로그래밍 방식으로 보안 암호를 검색할 수 있습니다.
- [AWS Systems Manager](#)는 AWS 클라우드에서 실행되는 애플리케이션과 인프라를 관리하는 데 도움이 됩니다. 애플리케이션 및 리소스 관리를 간소화하고, 운영 문제의 감지 및 해결 시간을 단축하며, AWS 리소스를 규모에 따라 안전하게 관리하는 데 도움이 됩니다.

기타 도구

- [PowerShell](#) Windows, Linux 및 macOS에서 실행되는 Microsoft 자동화 및 구성 관리 프로그램입니다. 이 패턴은 PowerShell 스크립트를 사용합니다.

코드 리포지토리

이 패턴의 코드는 GitHub [aws-windows-failover-cluster-automation](#) 리포지토리에서 사용할 수 있습니다.

모범 사례

- 이 솔루션을 배포하는 데 사용되는 IAM 역할은 최소 권한 원칙을 준수해야 합니다. 자세한 내용은 [IAM 설명서](#)를 참조하세요.
- [AWS CloudFormation 모범 사례](#)를 따르십시오.

에픽

인프라 배포

작업	설명	필요한 기술
Systems Manager CloudFormation 스택을 배포합니다.	<ol style="list-style-type: none"> 1. AWS 계정에 로그인하고 AWS Management Console을 엽니다. 2. CloudFormation 콘솔로 이동하여 <code>ssm.yaml</code> 템플릿을 업로드하여 Systems Manager CloudFormation 스택을 생성합니다. 다음 파라미터 값을 입력합니다. 	AWS DevOps, DevOps 엔지니어

작업	설명	필요한 기술
	<ul style="list-style-type: none"> • StateUnJoinAssociationLoggingBucketName— 템플릿이 로깅을 위해 생성할 S3 버킷의 이름을 입력합니다. • SSM AssociationADUnjoinName — 리소스의 이름을 제공합니다. AWS::SSM::Association • SSM AutomationDocumentName — Systems Manager 자동화 런북의 이름을 제공합니다. • EventBridgeName— EventBridge 이벤트 버스의 이름을 입력합니다. <p>3. ssm.yaml CloudFormation 템플릿을 실행하여 Systems Manager CloudFormation 스택을 배포합니다. 템플릿은 태그 ADJoined:FSXADD가 있는 새 EC2 인스턴스가 시작될 때 시작되는 Systems Manager Automation 런북을 생성합니다. Automation 런북은 AWS Managed Microsoft AD 디렉터리에 인스턴스를 추가합니다.</p>	

작업	설명	필요한 기술
인프라 스택을 배포합니다.	<p>Systems Manager 스택을 성공적으로 배포한 후 EC2 인스턴스 노드, 보안 그룹, Amazon FSx for Windows File Server 파일 시스템 및 IAM 역할을 포함하는 infra 스택을 생성합니다.</p> <p>1. CloudFormation 콘솔로 이동하여 infra-cf.yaml 템플릿을 실행합니다. 이 스택을 배포하려면 다음의 파라미터가 필요합니다.</p> <ul style="list-style-type: none"> • ActiveDirectoryId - AWS Managed Microsoft AD의 ID • ADDnsIpAddresses1 - AWS Managed Microsoft AD의 기본 DNS IP 주소 • ADDnsIpAddresses2 - AWS Managed Microsoft AD의 보조 DNS IP 주소 • FSxSecurityGroupName - Amazon FSx 보안 그룹의 이름 • FSxWindowsFileSystemName - Amazon FSx 드라이브 이름 • ImageID-SQL 서버 인스턴스 노드를 생성하는 데 사용된 기본 Windows 2012 R2 이미 	AWS DevOps, DevOps 엔지니어

작업	설명	필요한 기술
	<p>지 또는 Amazon Machine Image(AMI)의 ID</p> <ul style="list-style-type: none"> • KeyPairName -액세스를 위해 EC2 인스턴스 노드에 연결할 키-값 페어 • Node1SecurityGroupName -첫 번째 노드 보안 그룹의 이름 • Node2SecurityGroupName -두 번째 노드 보안 그룹의 이름 • OUSecretName -AWS Managed Microsoft AD 정보가 들어 있는 보안 암호의 이름 • PrivateSubnet1 -첫 번째 비공개 서브넷의 ID • PrivateSubnet2 -두 번째 비공개 서브넷의 ID • SqlFSxFCIName -기본 및 보조 노드와 Amazon FSx에 적용되는 태그의 이름. • SqlFSxServerNetBIOSName1 -기본 EC2 인스턴스 노드의 이름(최대 15 자) • SqlFSxServerNetBIOSName2 -보조 EC2 인스턴스 노드의 이름(최대 15 자) • VPC-VPC ID 	

작업	설명	필요한 기술
	<ul style="list-style-type: none"> • WorkloadInstanceType -EC2 인스턴스 유형 <p>infra 스택을 배포합니다. 스택은 Windows SQL Server FCI를 설정하는 데 필요한 모든 인프라 구성 요소를 생성합니다.</p> <p>2. EC2 인스턴스 노드가 시작되면 Systems Manager Automation 문서를 간접적으로 호출하여 이러한 인스턴스를 AWS Managed Microsoft AD에 조인합니다. Systems Manager 콘솔 자동화 페이지에서 진행 상황을 추적할 수 있습니다.</p>	

Windows SQL Server Always On FCI 설정

작업	설명	필요한 기술
<p>윈도우 도구를 설치합니다.</p>	<p>1. 기본 EC2 인스턴스(노드 1)에 로그인합니다. Windows 기능 (액티브 디렉터리 및 FCI 도구) 을 설치하려면 다음 PowerShell 스크립트를 실행합니다.</p> <pre data-bbox="630 1619 1029 1833">Install-WindowsFeature -Name RSAT-AD-Powershell,Failover-Clustering -IncludeManagementTools</pre>	<p>AWS DevOps, DevOps 엔지니어, DBA</p>

작업	설명	필요한 기술
	<pre>Install-WindowsFeature -Name RSAT-Clustering,RSAT-ADDS-Tools,RSAT-AD-Powershell,RSAT-DHCP,RSAT-DNS-Server</pre> <p>2. 보조 EC2 인스턴스(노드 2)에 로그인하고 동일한 스크립트를 실행하여 노드 2에서 기능을 활성화합니다.</p>	
Active Directory Domain Services에서 클러스터 컴퓨터 객체를 사전 준비합니다.	Active Directory Domain Services(AD DS)에서 클러스터 이름 개체(CNO)를 사전 준비하고 클러스터된 역할을 위해 가상 컴퓨터 개체(VCO)를 사전 준비하려면 Windows Server 설명서 의 지침을 따릅니다.	AWS DevOps, DBA, 엔지니어 DevOps

작업	설명	필요한 기술
<p>WSFC를 생성합니다.</p>	<p>Windows Server Failover Clustering(WSFC) 클러스터를 만들려면 다음과 같이 합니다.</p> <ol style="list-style-type: none"> 1. 기본 EC2 인스턴스(노드 1)에 로그인합니다. Amazon FSx 파일 공유를 생성하고 나열된 AD 서비스 계정에 대한 전체 액세스 권한을 부여하려면 다음 코드를 실행합니다. <div data-bbox="630 758 1029 1675" style="border: 1px solid #ccc; border-radius: 10px; padding: 10px; margin: 10px 0;"> <pre>Invoke-Command - ComputerName "<FSx Windows Remote PowerShell Endpoint> " -ConfigurationName FSxRemoteAdmin - scriptblock { New-FSxSmbShare -Name "SQLDB" -Path "D: \share" -Descript ion "SQL Databases Share" -Continuo uslyAvailable \$true -FolderEnumeration Mode AccessBased - EncryptData \$true grant-fsx smb shareaccess -name SQLDB -AccountName "<domain\user>" - accessRight Full }</pre> </div> <p>또한 이 명령을 실행하면 지속적으로 사용 가능한(CA) 파일 공유가 만들어지는데,</p>	<p>AWS DevOps, DBA, 엔지니어 DevOps</p>

작업	설명	필요한 기술
	<p>이 공유는 Microsoft SQL Server에서 사용하도록 최적화되어 있습니다.</p> <p>2. 기본 인스턴스(노드 1)에 장애 조치 클러스터를 만들려면 다음 명령을 실행합니다.</p> <pre data-bbox="633 529 1029 848">New-Cluster -Name <CNO Name> -Node <Node1 Name>, <Node2 Name> -StaticAddress <Node1 Secondary Private IP>, <Node2 Secondary Private IP></pre> <p>명령은 다음 파라미터를 필요로 합니다.</p> <ul data-bbox="630 991 1010 1331" style="list-style-type: none"> • Name-클러스터의 이름 (CNO) • Node-기본 노드와 보조 노드의 이름 각각 • StaticAddress -기본 노드와 보조 노드의 보조 IP 주소 각각 <p>중요: Windows Server Failover Clustering(WSFC) 클러스터를 만들려면 도메인 관리자 또는 일반 사용자에게 두 노드 모두에 대한 관리자 권한이 있어야 합니다. 그렇지 않으면 이전 명령이 실패하고 You do not have administrator privilege on</p>	

작업	설명	필요한 기술
	<p>servers라는 메시지가 반환됩니다.</p> <p>3. 클러스터가 생성된 후 다음 명령을 실행하여 파일 공유 감시를 연결합니다.</p> <pre>Set-ClusterQuorum - FileShareWitness \ <FSx Windows Remote PowerShell Endpoint> \share\witness</pre>	

작업	설명	필요한 기술
<p>SQL Server 장애 조치 클러스터를 설치합니다.</p>	<p>WSFC 클러스터를 설정한 후 기본 인스턴스(노드1)에 SQL Server 클러스터를 설치합니다.</p> <ol style="list-style-type: none"> 1. 두 노드의 T 드라이브에 tempdb 및 log 폴더를 생성합니다. 폴더는 PowerShell 명령에 사용됩니다. 2. SQL Server 설치용 SQL Server 미디어 파일을 두 노드에 복사한 후 노드 1에서 다음 PowerShell 명령을 실행하여 노드 1에 SQL Server를 설치합니다. <pre data-bbox="597 1050 1026 1854"> D:\setup.exe /Q ` /ACTION=InstallF ailoverCluster ` /IACCEPTSQLSERVE RLICENSETERMS ` /FEATURES="SQL,I S,BC,Conn" ` /INSTALLSHAREDDIR="C: \Program Files\Mic rosoft SQL Server" ` /INSTALLSHAREDWO WDIR="C:\Program Files (x86)\Microsoft SQL Server" ` /RSINSTALLMODE=" FilesOnlyMode" ` /INSTANCEID="MSS QLSERVER" ` /INSTANCENAME="M SSQLSERVER" ` </pre>	<p>AWS DevOps, DBA, 엔지니어 DevOps</p>

작업	설명	필요한 기술
	<pre> /FAILOVERCLUSTER GROUP="SQL Server (MSSQLSERVER)" ` /FAILOVERCLUSTER IPADDRESSES="IPv4; <2nd Sec Private Ip node1>;Cluster Network 1;<subnet mask>" ` /FAILOVERCLUSTER NETWORKNAME="<Fail over cluster Network Name>" ` /INSTANCEDIR="C: \Program Files\Mic rosoft SQL Server" ` /ENU="True" ` /ERRORREPORTING=0 ` /SQMREPORTING=0 ` /SAPWD="<Domain User password>" ` /SQLCOLLATION="S QL_Latin1_General_ CP1_CI_AS" ` /SQLSYSADMINACCO UNTS="<domain\user name>" ` /SQLSVCACCOUNT=" <domain\username>" /SQLSVCPASSWORD="< Domain User password>" ` /AGTSVCACCOUNT=" <domain\username>" /AGTSVCPASSWORD="< Domain User password>" ` /ISSVCACCOUNT="<domain \username>" /ISSVCPAS SWORD="<Domain User password>" ` </pre>	

작업	설명	필요한 기술
	<pre> /FTSVCAccount="NT Service\MSSQLFDLau ncher" ` /INSTALLSQLDATADIR="\ \<fsx dns="" name="">\sha re\Program Files\Mic rosoft SQL Server" ` /SQLUSERDBDIR="\\<FSX DNS name>\share\data" ` /SQLUSERDBLOGDIR="\ \<fsx dns="" name="">\share \log" ` /SQLTEMPDBDIR="T: \tempdb" ` /SQLTEMPDBLOGDIR="T: \log" ` /SQLBACKUPDIR="\\<FSX DNS name>\share\SQLBac kup" ` /SkipRules=Clust er_VerifyForErrors ` /INDICATEPROGRESS </fsx></fsx></pre>	

작업	설명	필요한 기술
클러스터에 보조 노드를 추가합니다.	<p>보조 노드 (노드 2) 에 SQL Server를 추가하려면 다음 PowerShell 명령을 실행합니다.</p> <pre data-bbox="592 443 1029 1806"> D:\setup.exe /Q ` /ACTION=AddNode ` /IACCEPTSQLSERVE RLICENSETERMS ` /INSTANCENAME="M SSQLSERVER" ` /FAILOVERCLUSTER GROUP="SQL Server (MSSQLSERVER)" ` /FAILOVERCLUSTER IPADDRESSES="IPv4; <2nd Sec Private Ip node2>;Cluster Network 2;<subnet mask>" ` /FAILOVERCLUSTER NETWORKNAME="<Fail over cluster Network Name>" ` /CONFIRMIPDEPEND ENCYCHANGE=1 ` /SQLSVCACCOUNT=" <domain\username>" /SQLSVCPASSWORD="< Domain User password>" /AGTSVCACCOUNT="domain \username>" /AGTSVCPA SSWORD="<Domain User password>" ` /FTSVCACCOUNT="NT Service\MSSQLFDLau ncher" ` /SkipRules=Clust er_VerifyForErrors ` </pre>	AWS DevOps, DBA, 엔지니어 DevOps

작업	설명	필요한 기술
	/INDICATEPROGRESS	
<p>SQL 서버 FCI를 테스트합니다.</p>	<ol style="list-style-type: none"> 1. 노드 중 하나에 대한 Windows 인스턴스의 관리 도구에서 장애 조치 클러스터 관리자를 시작합니다. 2. 노드로 이동하여 노드 상태가 실행 상태인지 확인합니다. 3. 역할을 선택하고 SQL Serve(MSSQLSERVER)의 컨텍스트(마우스 오른쪽 버튼 클릭) 메뉴를 연 다음 이동 및 노드 선택을 선택합니다. 4. 노드를 선택한 후에는 다른 노드에서 SQL Server가 실행되고 있어야 합니다. 	<p>DBA, 엔지니어 DevOps</p>

리소스 정리

작업	설명	필요한 기술
<p>리소스를 정리하십시오.</p>	<p>리소스를 정리하려면 AWS CloudFormation 스택 삭제 프로세스를 사용하십시오.</p> <ol style="list-style-type: none"> 1. AWS CloudFormation 콘솔을 엽니다. 2. 스택 페이지에서 infra 스택을 선택합니다. 스택이 현재 실행 중이어야 합니다. 3. 스택 세부 정보 창에서 삭제를 선택합니다. 	<p>AWS DevOps, DBA, 엔지니어 DevOps</p>

작업	설명	필요한 기술
	<p>4. 메시지가 나타나면 스택 삭제를 선택하세요.</p> <p>5. ssm 스택에 대해 2~4단계를 반복합니다.</p> <p>스택 삭제가 완료되면 스택의 상태가 DELETE_COMPLETE 으로 바뀝니다. DELETE_COMPLETE 상태에 있는 스택은 기본적으로 CloudFormation 콘솔에 표시되지 않습니다. 삭제된 스택을 표시하려면 AWS CloudFormation 콘솔에서 삭제된 스택 보기에 설명된 대로 스택 보기 필터를 변경해야 합니다.</p> <p>삭제에 실패하면 스택이 DELETE_FAILED 상태가 됩니다. 해결 방법은 설명서의 스택 삭제 실패를 CloudFormation 참조하십시오.</p>	

문제 해결

문제	Solution
<p>AWS CloudFormation 템플릿 실패</p>	<p>배포 중에 CloudFormation 템플릿이 실패하는 경우 다음을 수행하십시오.</p> <ol style="list-style-type: none"> 1. AWS CloudFormation 콘솔을 엽니다. 2. CloudFormation 콘솔의 스택 페이지에서 스택을 선택합니다.

문제	Solution
AWS Managed Microsoft AD 조인 실패	<p>3. 이벤트를 선택하고 스택 상태를 확인합니다.</p> <p>조인 문제를 해결하려면 다음 단계를 수행하세요.</p> <ol style="list-style-type: none"> 1. Systems Manager 콘솔을 엽니다. 2. 배포 리전을 선택합니다. 3. 왼쪽 창에서 자동화를 선택하고 실패한 자동화 런북을 찾습니다. 4. 자동화 런북을 열고 실행 상태 및 실행 단계를 확인합니다. 5. 실패한 단계의 세부 정보를 조사하여 정확한 오류 또는 실패를 확인합니다.

관련 리소스

- [Amazon FSx for Windows File Server를 사용하여 Microsoft SQL Server 높은 가용성 배포 단순화](#)
- [Microsoft SQL Server가 포함된 FSx for Windows File Server 사용](#)

BMC Discovery 쿼리를 사용하여 마이그레이션 계획을 위한 마이그레이션 데이터 추출

작성자: Ben Taylor-Hamblin(AWS), Simon Cunningham(AWS), Emma Baldry(AWS) 및 Shabnam Khan(AWS)

환경: 프로덕션	소스: : BMC Discovery	대상: Migration Plan
R 유형: 리호스팅	워크로드: 기타 모든 워크로드	기술: 마이그레이션, 관리 및 거버넌스, 네트워킹, 하이브리드 클라우드
AWS 서비스: AWS Migration Hub		

요약

이 가이드는 BMC Discovery를 사용하여 온프레미스 인프라 및 애플리케이션에서 데이터를 추출하는 데 도움이 되는 쿼리 예제와 단계를 제공합니다. 이 패턴은 BMC Discovery 쿼리를 사용하여 인프라를 스캔하고 소프트웨어, 서비스 및 종속성 정보를 추출하는 방법을 보여줍니다. 추출된 데이터는 Amazon Web Services(AWS) 클라우드로의 대규모 마이그레이션을 평가하고 동원하는 단계를 수행하는 데 필요합니다. 이 데이터를 사용하여 마이그레이션 계획의 일환으로 어떤 애플리케이션을 함께 마이그레이션할지에 대한 중요한 결정을 내릴 수 있습니다.

사전 조건 및 제한 사항

사전 조건

- BMC Discovery(구 BMC ADDM) 또는 BMC Helix Discovery의 서비스형 소프트웨어(SaaS) 버전에 대한 라이선스
- 온프레미스 또는 SaaS 버전의 BMC Discovery가 [설치되어 있음](#) (참고: 온프레미스 버전의 BMC Discovery의 경우 여러 데이터 센터 간에 마이그레이션할 수 있는 범위 내에 있는 모든 네트워킹 및 서버 장치에 액세스할 수 있는 클라이언트 네트워크에 애플리케이션을 설치해야 합니다. 클라이언트 네트워크에 대한 액세스는 애플리케이션 설치 지침에 따라 제공되어야 합니다. Windows Server 정보 검사가 필요한 경우 네트워크에 Windows 프록시 관리자 장치를 설정해야 합니다.)
- BMC Helix Discovery를 사용하는 경우 애플리케이션이 데이터 센터의 여러 디바이스를 스캔할 수 있도록 하는 [네트워킹 액세스](#)

제품 버전

- BMC Discovery 22.2(12.5)
- BMC Discovery 22.1(12.4)
- BMC Discovery 21.3(12.3)
- BMC Discovery 21.05(12.2)
- BMC Discovery 20.08(12.1)
- BMC Discovery 20.02(12.0)
- BMC Discovery 11.3
- BMC Discovery 11.2
- BMC Discovery 11.1
- BMC Discovery 11.0
- BMC Atrium Discovery 10.2
- BMC Atrium Discovery 10.1
- BMC Atrium Discovery 10.0

아키텍처

다음 다이어그램은 자산 관리자가 BMC Discovery 쿼리를 사용하여 SaaS 및 온프레미스 환경 모두에서 BMC 모델링 애플리케이션을 스캔하는 방법을 보여줍니다.

다이어그램은 다음 워크플로를 보여줍니다. 자산 관리자는 BMC Discovery 또는 BMC Helix Discovery를 사용하여 여러 물리적 서버에 호스팅된 가상 서버에서 실행되는 데이터베이스 및 소프트웨어 인스턴스를 스캔합니다. 이 도구는 여러 가상 및 물리적 서버에 걸친 구성 요소를 사용하여 애플리케이션을 모델링할 수 있습니다.

기술 스택

- BMC Discovery
- BMC Helix Discovery

도구

- [BMC Discovery](#)는 데이터 센터를 자동으로 검색하는 데 도움이 되는 데이터 센터 검색 도구입니다.

- [BMC Helix Discovery](#)는 데이터 자산과 종속성을 동적으로 모델링하는 데 도움이 되는 SaaS 기반 검색 및 종속성 모델링 시스템입니다.

모범 사례

클라우드에 마이그레이션할 때 애플리케이션, 종속성 및 인프라 데이터를 매핑하는 것이 가장 좋습니다. 매핑은 현재 환경의 복잡성과 다양한 구성 요소 간의 종속성을 이해하는 데 도움이 됩니다.

이러한 쿼리가 제공하는 자산 정보는 다음과 같은 여러 가지 이유로 중요합니다.

1. 계획 – 구성 요소 간의 종속성을 이해하면 마이그레이션 프로세스를 보다 효과적으로 계획하는 데 도움이 됩니다. 예를 들어, 다른 구성 요소를 성공적으로 마이그레이션하려면 먼저 특정 구성 요소를 마이그레이션해야 할 수 있습니다.
2. 위험 평가 – 구성 요소 간의 종속성을 매핑하면 마이그레이션 프로세스 중에 발생할 수 있는 잠재적 위험이나 문제를 식별하는 데 도움이 될 수 있습니다. 예를 들어, 특정 구성 요소가 구식이거나 지원되지 않는 기술에 의존하여 클라우드에서 문제를 일으킬 수 있다는 사실을 발견할 수 있습니다.
3. 클라우드 아키텍처 – 애플리케이션과 인프라 데이터를 매핑하면 조직의 요구 사항에 맞는 적절한 클라우드 아키텍처를 설계하는 데도 도움이 될 수 있습니다. 예를 들어 고가용성 또는 확장성 요구 사항을 지원하는 다중 계층 아키텍처를 설계해야 할 수 있습니다.

전반적으로 애플리케이션, 종속성 및 인프라 데이터를 매핑하는 것은 클라우드 마이그레이션 프로세스의 중요한 단계입니다. 매핑 연습을 통해 현재 환경을 더 잘 이해하고, 잠재적 문제나 위험을 식별하며, 적합한 클라우드 아키텍처를 설계할 수 있습니다.

에픽

검색 툴링 식별 및 평가

작업	설명	필요한 기술
ITSM 소유자를 식별합니다.	일반적으로 운영 지원 팀에 문의하여 IT Service Management(ITSM) 소유자를 식별합니다.	마이그레이션 책임자
CMDB를 확인합니다.	자산 정보가 들어 있는 구성 관리 데이터베이스(CMDB)의 수	마이그레이션 책임자

작업	설명	필요한 기술
	를 식별한 다음 해당 정보의 출처를 식별합니다.	
<p>검색 도구를 식별하고 BMC Discovery의 사용 여부를 확인합니다.</p>	<p>조직에서 BMC Discovery를 사용하여 환경에 대한 데이터를 CMDB 도구로 보내는 경우 스캔 범위와 적용 범위를 확인합니다. 예를 들어 BMC Discovery가 모든 데이터 센터를 스캔하고 있는지, 액세스 서버가 경계 구역에 있는지 확인합니다.</p>	<p>마이그레이션 책임자</p>
<p>애플리케이션 모델링 수준을 확인합니다.</p>	<p>애플리케이션이 BMC Discovery에서 모델링되었는지 확인합니다. 그렇지 않은 경우 BMC Discovery 도구를 사용하여 실행 중인 소프트웨어 인스턴스가 애플리케이션 및 비즈니스 서비스를 제공하는지 모델링할 것을 권장합니다.</p>	<p>마이그레이션 엔지니어, 마이그레이션 책임자</p>

인프라 데이터 추출

작업	설명	필요한 기술
<p>물리적 및 가상 서버에서 데이터를 추출합니다.</p>	<p>BMC Discovery에서 스캔한 물리적 및 가상 서버의 데이터를 추출하려면 Query Builder를 사용하여 다음 쿼리를 실행합니다.</p> <pre data-bbox="594 1709 1029 1883"> search Host show key as 'Serverid ', virtual, name as 'HOSTNAME', os_type </pre>	<p>마이그레이션 엔지니어, 마이그레이션 책임자</p>

작업	설명	필요한 기술
	<pre> as 'osName', os_versio n as 'OS Version', num_logical_proces sors as 'Logical Processor Counts', cores_per_processo r as 'Cores per Processor', logical_r am as 'Logical RAM', #Consumer:StorageU se:Provider:DiskDr ive.size as 'Size' </pre> <p>참고: 추출된 데이터를 사용하여 마이그레이션에 적합한 인스턴스 크기를 결정할 수 있습니다.</p>	

작업	설명	필요한 기술
<p>모델링된 애플리케이션에서 데이터를 추출합니다.</p>	<p>애플리케이션이 BMC Discovery에서 모델링된 경우 애플리케이션 소프트웨어를 실행하는 서버에 대한 데이터를 추출할 수 있습니다. 서버 이름을 가져오려면 Query Builder를 사용하여 다음 쿼리를 실행합니다.</p> <pre data-bbox="597 632 1027 951"> search SoftwareInstance show key as 'ApplicationID', #RunningSoftware:HostedSoftware:Host:Host.key as 'ReferenceID', type, name </pre> <p>참고: 애플리케이션은 실행 중인 소프트웨어 인스턴스 모음을 기반으로 BMC Discovery에서 모델링됩니다. 애플리케이션은 애플리케이션 소프트웨어를 실행하는 모든 서버에 종속됩니다.</p>	<p>BMC Discovery 애플리케이션 소유자</p>

작업	설명	필요한 기술
<p>데이터베이스에서 데이터를 추출합니다.</p>	<p>스캔한 모든 데이터베이스와 해당 데이터베이스가 실행되고 있는 서버의 목록을 가져오려면 Query Builder를 사용하여 다음 쿼리를 실행합니다.</p> <pre data-bbox="594 489 1029 1402"> search Database show key as 'Key', name, type as 'Source Engine Type', #Detail:D etail:ElementWithD etail:SoftwareInst ance.name as 'Software Instance', #Detail:D etail:ElementWithD etail:SoftwareInst ance.product_version as 'Product Version', #Detail:Detail:Ele mentWithDetail:Sof twareInstance.edit ion as 'Edition', #Detail:Detail:Ele mentWithDetail:Sof twareInstance.#Run ningSoftware:Hoste dSoftware:Host:Hos t.key as 'ServerID' </pre>	<p>앱 소유자</p>

작업	설명	필요한 기술
<p>서버 통신에서 데이터를 추출합니다.</p>	<p>BMC Discovery가 이전 네트워크 통신 로그에서 수집한 서버 간 모든 네트워크 통신에 대한 정보를 가져오려면 Query Builder를 사용하여 다음 쿼리를 실행합니다.</p> <pre data-bbox="597 537 1026 1171"> search Host TRVERSE InferredElement:Inference:Associate:DiscoveryAccess TRVERSE DiscoveryAccess:DiscoveryAccessResult:DiscoveryResult:NetworkConnectionList TRVERSE List:List:Member:DiscoveredNetworkConnection PROCESS WITH networkConnectionInfo </pre>	<p>BMC Discovery 애플리케이션 소유자</p>
<p>애플리케이션 검색에서 데이터를 추출합니다.</p>	<p>애플리케이션 종속성에 대한 정보를 얻으려면 Query Builder를 사용하여 다음 쿼리를 실행합니다.</p> <pre data-bbox="597 1428 1026 1743"> search SoftwareInstance show key as 'SRC App ID', #Dependant:Dependency:DependedUpon:SoftwareInstance.key as 'DEST App ID' </pre>	<p>BMC Discovery 애플리케이션 소유자</p>

작업	설명	필요한 기술
비즈니스 서비스에 대한 데이터를 추출합니다.	호스트가 제공하는 비즈니스 서비스에 대한 데이터를 추출하려면 Query Builder 를 사용하여 다음 쿼리를 실행합니다. <pre data-bbox="597 443 1027 680">search Host show name, #Host:HostedSoftware:AggregateSoftware:BusinessService .name as 'Name'</pre>	BMC Discovery 애플리케이션 소유자

문제 해결

문제	Solution
쿼리가 실행되지 않거나 채워지지 않은 열이 있습니다.	BMC Discovery에서 자산 레코드를 검토하고 필요한 필드를 결정하십시오. 그런 다음 Query Builder 를 사용하여 쿼리의 이러한 필드를 바꾸십시오.
종속 자산의 세부 정보는 채워지지 않습니다.	이는 액세스 권한 또는 네트워크 연결 때문일 수 있습니다. 검색 도구에는 특정 자산에 액세스하는 데 필요한 권한이 없을 수 있습니다. 특히 자산이 다른 네트워크나 환경에 있는 경우에는 더욱 그렇습니다. 모든 관련 자산을 식별할 수 있도록 검색 주제 전문가와 긴밀히 협력하는 것이 좋습니다.

관련 리소스

참조

- [BMC Discovery 라이선싱 자격](#)(BMC 설명서)

- [BMC Discovery 기능 및 구성 요소](#)(BMC 설명서)
- [BMC Discovery 사용 설명서](#)(BMC 설명서)
- [데이터 검색\(BMC Discovery\)](#)(BMC 설명서)
- [마이그레이션을 위한 포트폴리오 검색 및 분석](#)(AWS 권장 가이드)

튜토리얼 및 동영상

- [BMC 디스커버리: 웨비나 - 쿼리 보고 모범 사례 \(1부\) \(\)](#) YouTube

재배치

주제

- [지속적인 복제를 위해 AWS DMS를 사용하여 Amazon RDS for Oracle 데이터베이스를 다른 AWS 계정 및 AWS 리전으로 마이그레이션](#)
- [VMware HCX를 사용하여 VMware SDDC를 AWS의 VMware Cloud로 마이그레이션](#)
- [Amazon RDS DB 인스턴스를 다른 VPC 또는 계정으로 마이그레이션](#)
- [Amazon RDS for Oracle DB 인스턴스를 다른 VPC로 마이그레이션](#)
- [Amazon Redshift 클러스터를 중국의 AWS 리전으로 마이그레이션](#)
- [VMware HCX를 사용하여 워크로드를 AWS의 VMware Cloud로 마이그레이션합니다.](#)
- [pg_transport를 사용하여 두 Amazon RDS DB 인스턴스 간에 PostgreSQL 데이터베이스 전송](#)

지속적인 복제를 위해 AWS DMS를 사용하여 Amazon RDS for Oracle 데이터베이스를 다른 AWS 계정 및 AWS 리전으로 마이그레이션

두르가 프라사드 치푸리 (AWS) 와 에두아르도 발렌티 (AWS) 이 제작했습니다.

환경: PoC 또는 파일럿	소스: 데이터베이스: 관계형	대상: Amazon RDS for Oracle
R 유형: 재배치	워크로드: Oracle	기술: 마이그레이션, 데이터베이스
AWS 서비스: Amazon RDS		

요약

경고: IAM 사용자는 장기 자격 증명을 보유하므로 보안상 위험이 있습니다. 이 위험을 줄이려면 이러한 사용자에게 작업을 수행하는 데 필요한 권한만 제공하고 더 이상 필요하지 않을 경우 이러한 사용자를 제거하는 것이 좋습니다.

이 패턴은 Oracle 소스 데이터베이스를 위한 Amazon Relational Database Service(Amazon RDS)를 다른 AWS 계정 및 AWS 리전으로 마이그레이션하는 절차를 안내합니다. 이 패턴은 한 번의 전체 데이터 로드와 DB 스냅샷을 사용하고, 지속적인 복제를 위해 AWS Database Migration Service(AWS DMS)를 활성화합니다.

사전 조건 및 제한 사항

사전 조건

- 기본이 아닌 AWS Key Management Service(AWS KMS) 키를 사용하여 암호화된 Amazon RDS for Oracle 소스 데이터베이스를 포함하는 활성 상태의 AWS 계정
- 대상 Amazon RDS for Oracle 데이터베이스에 사용할 소스 데이터베이스와 다른 AWS 리전에 있는 활성 상태의 AWS 계정
- 소스 및 대상 VPC 간 Virtual Private Cloud(VPC) 피어링
- [Oracle 데이터베이스를 AWS DMS의 소스로 사용](#)하는 방법 속지
- [Oracle 데이터베이스를 AWS DMS의 대상으로 사용](#)하는 방법 속지

제품 버전

- Oracle 버전 11g(버전 11.2.0.3.v1 이상) 및 최대 12.2 및 18c. 지원되는 버전 및 에디션의 최신 목록은 AWS 설명서의 [Oracle 데이터베이스를 AWS DMS의 소스로 사용](#) 및 [Oracle 데이터베이스를 AWS DMS의 대상으로 사용](#)을 참조하세요. Amazon RDS에서 지원하는 Oracle 버전에 대해서는 [Amazon RDS의 Oracle](#)을 참조하세요.

아키텍처

소스 및 대상 기술 스택

- Amazon RDS for Oracle DB 인스턴스

지속적인 복제 아키텍처

도구

전체 데이터를 한 번에 로드하는 데 사용되는 도구:

- Amazon RDS DB 스냅샷 - Amazon RDS는 개별 데이터베이스만이 아닌 전체 DB 인스턴스를 백업하는 DB 인스턴스의 스토리지 볼륨 스냅샷을 생성합니다. DB 스냅샷을 생성할 때는 백업할 DB 인스턴스를 구분한 다음 나중에 복구할 수 있도록 DB 스냅샷을 명명해야 합니다. 스냅샷을 생성하는 데 걸리는 시간은 데이터베이스 크기에 따라 다릅니다. 스냅샷에는 전체 스토리지 볼륨이 포함되기 때문에 임시 파일 같은 파일들의 크기가 스냅샷을 생성하는 데 걸리는 시간에 영향을 미치기도 합니다. DB 스냅샷 사용 방법에 대한 자세한 내용은 Amazon RDS 설명서의 [DB 스냅샷 생성](#)을 참조하세요.
- Amazon RDS 암호화를 위한 KMS 키- 암호화된 DB 인스턴스를 생성할 때 암호화 키에 대한 KMS 키 식별자를 제공할 수도 있습니다. KMS 키 식별자를 지정하지 않으면 Amazon RDS가 새 DB 인스턴스에 대해 기본 암호화 키를 사용합니다. AWS KMS는 사용자의 AWS 계정을 위한 기본 암호화 키를 생성합니다. AWS 계정에는 AWS 리전마다 다른 기본 암호화 키가 있습니다. 이 패턴의 경우 기본이 아닌 KMS 키를 사용하여 Amazon RDS DB 인스턴스를 암호화해야 합니다. Amazon RDS 암호화에 KMS 키를 사용하는 방법에 대한 자세한 내용은 Amazon RDS 설명서의 [Amazon RDS 리소스 암호화](#)를 참조하세요.

지속적인 복제에 사용되는 도구:

- AWS DMS - 이 패턴은 AWS DMS를 사용하여 진행 중인 변경 사항을 복제하고 소스 및 대상 데이터베이스의 동기화 상태를 유지합니다. 지속적인 복제를 위해 AWS DMS를 사용하는 방법에 대한 자세한 내용은 AWS DMS 설명서의 [AWS DMS 복제 인스턴스를 사용한 작업을 참조하세요](#).

모범 사례

< 작성자: 참고 사항 삭제: 사용자가 이 패턴을 보다 효과적으로 구현하는 데 도움이 될 수 있는 지침 및 권장 사항 목록을 제공하십시오. >

에픽

소스 AWS 계정 구성

작업	설명	필요한 기술
소스 Oracle DB 인스턴스를 준비합니다.	Amazon RDS for Oracle DB 인스턴스를 ARCHIVELOG 모드에서 실행하도록 하고 보존 기간을 설정합니다. 자세한 내용은 https://docs.aws.amazon.com/dms/latest/userguide/CHAP_Source.Oracle.html#CHAP_Source.Oracle.Amazon-Managed 를 참조하세요.	DBA
소스 Oracle DB 인스턴스에 대한 보충 로깅을 설정합니다.	Amazon RDS Oracle DB 인스턴스에 대한 데이터베이스 및 테이블 수준의 보충 로깅을 설정합니다. 자세한 내용은 https://docs.aws.amazon.com/dms/latest/userguide/CHAP_Source.Oracle.html#CHAP_Source.Oracle.Amazon-Managed 를 참조하세요.	DBA

작업	설명	필요한 기술
소스 계정의 KMS 키 정책을 업데이트합니다.	대상 AWS 계정이 암호화된 Amazon RDS KMS 키를 사용할 수 있도록 소스 AWS 계정의 KMS 키 정책을 업데이트합니다. 자세한 내용은 https://docs.aws.amazon.com/kms/latest/developerguide/key-policy-modifying-key-policy-modifying-external.html#accounts 를 참조하십시오.	SysAdmin
소스 DB 인스턴스의 수동 Amazon RDS DB 스냅샷을 생성합니다.		AWS IAM 사용자
암호화된 수동 Amazon RDS 스냅샷을 대상 AWS 계정과 공유합니다.	자세한 내용은 https://docs.aws.amazon.com/AmazonRDS/latest/UserGuide/ShareSnapshot 을 참조하십시오.	AWS IAM 사용자

대상 AWS 계정 구성

작업	설명	필요한 기술
정책을 연결합니다.	대상 AWS 계정에서 AWS Identity and Access Management(IAM) 정책을 루트 IAM 사용자에게 연결하여 IAM 사용자가 공유된 AWS KMS 키를 사용하여 암호화된 DB 스냅샷을 복사할 수 있게 합니다.	SysAdmin

작업	설명	필요한 기술
소스 AWS 리전으로 전환합니다.		AWS IAM 사용자
공유된 스냅샷을 복사합니다.	Amazon RDS 콘솔의 스냅샷 창에서 Shared with Me를 선택하고 공유된 스냅샷을 선택합니다. 소스 데이터베이스에서 사용하는 KMS 키의 Amazon 리소스 이름(ARN)을 사용하여 소스 데이터베이스와 동일한 AWS 리전에 스냅샷을 복사합니다. 자세한 내용은 https://docs.aws.amazon.com/AmazonRDS/latest/UserGuide/USER_CopySnapshot 을 참조하십시오.	AWS IAM 사용자
대상 AWS 리전으로 전환하고 새 KMS 키를 생성합니다.		AWS IAM 사용자
스냅샷을 복사합니다.	소스 AWS 리전으로 전환합니다. Amazon RDS 콘솔의 스냅샷 창에서 Owned by Me를 선택하고 복사된 스냅샷을 선택합니다. 새 대상 AWS 리전의 KMS 키를 사용하여 스냅샷을 대상 AWS 리전에 복사합니다.	AWS IAM 사용자

작업	설명	필요한 기술
스냅샷을 복원합니다.	대상 AWS 리전으로 전환합니다. Amazon RDS 콘솔의 스냅샷 창에서 Owned by Me를 선택합니다. 복사된 스냅샷을 선택한 후 Amazon RDS for Oracle DB 인스턴스로 복원합니다. 자세한 내용은 https://docs.aws.amazon.com/AmazonRDS/latest/UserGuide/USER_RestoreFromSnapshot	AWS IAM 사용자

지속적인 복제를 위해 소스 데이터베이스 준비

작업	설명	필요한 기술
적절한 권한을 가진 Oracle 사용자를 생성합니다.	AWS DMS의 소스로서 Oracle에 대한 필수 권한을 가진 Oracle 사용자를 생성합니다. 자세한 내용은 https://docs.aws.amazon.com/dms/latest/userguide/CHAP_SourceOracle.html 을 참조하세요.	DBA
오라클 LogMiner 또는 오라클 바이너리 리더용 소스 데이터베이스를 구성하십시오.		DBA

지속적인 복제를 위해 대상 데이터베이스 준비

작업	설명	필요한 기술
적절한 권한을 가진 Oracle 사용자를 생성합니다.	AWS DMS의 대상으로서 Oracle에 대한 필수 권한을 가진 Oracle 사용자를 생성합니다. 자세한 내용은 https://docs.aws.amazon.com/dms/latest/userguide/CHAP_Target.Oracle.html#CHAP_Target.Oracle.Privileges 를 참조하세요.	DBA

AWS DMS 구성 요소 생성

작업	설명	필요한 기술
대상 AWS 리전에 복제 인스턴스를 생성합니다.	대상 AWS 리전의 VPC에 복제 인스턴스를 생성합니다. 자세한 내용은 https://docs.aws.amazon.com/dms/latest/userguide/CHAP_GettingStarted.html#CHAP_GettingStarted 을 참조하십시오. ReplicationInstance.	AWS IAM 사용자
필요한 암호화로 소스 및 대상 엔드포인트를 생성하고 연결을 테스트합니다.	자세한 내용은 https://docs.aws.amazon.com/dms/latest/userguide/CHAP_GettingStarted.html#CHAP_GettingStarted .엔드포인트를 참조하십시오.	DBA
복제 작업을 생성합니다.	마이그레이션 유형으로는 지속적인 복제를 선택합니다. 변경 데이터 캡처(CDC) 시작점으로는 전체 로드를 위해 Amazon	IAM 사용자

작업	설명	필요한 기술
	<p>RDS 스냅샷을 생성한 경우 Oracle 시스템 변경 번호(SCN)을 사용하고 전체 로드가 수행된 시점의 타임스탬프를 사용합니다. 그러려면 <code>DO_NOTHING</code>을 <code>TargetTablePrepMode</code> 선택하십시오. 작업에 대용량 이진 객체(LOB) 데이터 테이블이 있는 경우 제한된 LOB 모드를 선택하고 최대 LOB 크기를 테이블에 있는 LOB 데이터의 최대 크기로 설정합니다. 로깅을 활성화합니다. 키를 통해 관련된 테이블을 단일 작업으로 그룹화합니다. 많은 양의 LOB 데이터가 있는 테이블이 있고 테이블이 다른 테이블과 관계가 없는 경우, 앞서 설명한 LOB 설정을 사용하여 해당 테이블에 대한 별도의 작업을 생성합니다. 자세한 내용은 https://docs.aws.amazon.com/dms/latest/userguide/CHAP_GettingStarted.html#CHAP_GettingStarted를 참조하십시오. <code>GettingStarted</code></p>	
<p>작업을 시작하고 이를 모니터링합니다.</p>	<p>자세한 내용은 https://docs.aws.amazon.com/dms/latest/userguide/CHAP_Monitoring.html를 참조하세요.</p>	<p>AWS IAM 사용자</p>

작업	설명	필요한 기술
필요한 경우 작업에 대한 검증을 활성화합니다.	검증을 활성화하면 복제 성능에 영향을 미친다는 점에 유의하세요. 자세한 내용은 https://docs.aws.amazon.com/dms/latest/userguide./CHAP_Validating.html 을 참조하세요.	AWS IAM 사용자

관련 리소스

- [KMS 키 정책 변경](#)
- [수동 Amazon RDS DB 스냅샷 생성](#)
- [수동 Amazon RDS DB 스냅샷 공유](#)
- [스냅샷 복사](#)
- [Amazon RDS DB 스냅샷에서 복원](#)
- [AWS DMS 시작하기](#)
- [Oracle 데이터베이스를 AWS DMS의 소스로 사용](#)
- [Oracle 데이터베이스를 AWS DMS의 대상으로 사용](#)
- [VPC 피어링을 사용한 AWS DMS 설정](#)
- [수동 Amazon RDS DB 스냅샷 또는 DB 클러스터 스냅샷을 다른 AWS 계정과 공유하려면 어떻게 해야 하나요? \(AWS 지식 센터 문서\)](#)

VMware HCX를 사용하여 VMware SDDC를 AWS의 VMware Cloud로 마이그레이션

작성자: Deepak Kumar(AWS)

환경: PoC 또는 파일럿	소스: 네트워킹	대상: AWS의 VMware Cloud
R 유형: 재배치	기술: 마이그레이션, 인프라	

요약

이 패턴은 VMware Hybrid Cloud Extension(HCX)을 사용하여 온프레미스 가상 머신(VM) 및 애플리케이션을 Amazon Web Services(AWS)의 VMware Cloud로 마이그레이션하는 방법을 설명합니다. 마이그레이션에서는 AWS 클라우드에서 VMware 엔터프라이즈급 소프트웨어 정의 데이터 센터(SDDC) 소프트웨어를 사용하여 AWS 서비스에 대한 최적화된 액세스를 제공합니다.

AWS의 VMware Cloud는 컴퓨팅, 스토리지 및 네트워크 가상화 제품(vSphere, vSAN 및 VMware NSX)을 탄력적인 전용 베어 메탈 AWS 인프라에서 실행할 수 있도록 최적화된 VMware vCenter 서버 관리와 통합합니다. 그 결과 유지 관리가 거의 필요 없으며 단순한 하이퍼 컨버지드 인프라가 됩니다.

이 서비스를 통해 IT 팀은 친숙한 VMware 도구를 사용하여 클라우드 기반 리소스를 관리할 수 있습니다. 자세한 내용은 VMware 웹 사이트의 [AWS의 VMware Cloud](#)를 참조하세요.

VMware HCX는 다음 세 가지 유형의 클라우드 마이그레이션을 지원합니다.

- 하이브리드(데이터 센터 확장): 기존 온프레미스 VMware SDDC를 AWS로 확장하여 설치 공간 확장, 온디맨드 용량, 테스트/개발 환경 및 가상 데스크톱을 제공합니다.
- 클라우드 대피(데이터 센터 전체 인프라 새로 고침): 데이터 센터를 통합하고 AWS 클라우드로 완전히 이동합니다(데이터 센터 코로케이션 또는 임대 종료 처리 포함).
- 애플리케이션별 마이그레이션: 특정 비즈니스 요구 사항을 충족하기 위해 개별 애플리케이션을 AWS 클라우드로 이동합니다.

사전 조건 및 제한 사항

사전 조건

- AWS 계정에 가입합니다 (VMware Cloud SDDC 생성에 필요).
- My VMware 계정에 가입합니다. <https://my.vmware.com/web/vmware/>에서 등록하고 모든 필드를 작성합니다.
- vCenter 및 호스트의 버전을 확인하고 VM 수를 수집합니다. 가능하면 [RVTools](#) 내보내기를 요청하여 가상 환경에 대한 정보를 표시합니다. vCenter 버전 6.0 이상을 사용하는 것이 좋습니다.
- 데이터 센터 네트워크(L2)를 확장하거나, HCX를 사용하여 vMotion을 테스트하거나, vRealize Network Insight를 사용하여 애플리케이션 종속성을 분석하려면 분산 가상 스위치를 배포해야 합니다.
- 충돌하지 않는 온프레미스 현재 관리 서브넷 네트워크를 선택하여 AWS의 VMware Cloud에서 SDDC를 생성합니다.
- [VMware HCX 사용 설명서](#)에 제공된 사전 조건을 검토하여 HCX 요구 사항을 검증합니다.
- 마이그레이션 웨이브를 위한 VM을 식별하고 그룹화합니다. 테스트에 사용할 수 있는 VM을 확인합니다.
- 상대적 대역폭 사용량, WAN 압축 및 데이터 전송 속도에 대한 모든 데이터를 수집합니다.

참고

- 온프레미스에는 VMware NSX-V 또는 NSX-T가 필요하지 않습니다.
- HCX에는 추가 비용이 없습니다(AWS의 VMware Cloud에 포함됨).

아키텍처

다음 다이어그램은 여러 구성 요소 서비스에 구축된 HCX 솔루션을 보여 줍니다. 각 구성 요소는 HCX 솔루션의 특정 기능을 지원합니다. 각 HCX 구성 요소에 대한 자세한 내용은 [Migrating Workloads to VMware Cloud on AWS with Hybrid Cloud Extension \(HCX\)](#) 블로그 게시물을 참조하세요.

소스 기술 스택

- VMware vSphere에서 관리하는 온프레미스 VM 및 애플리케이션

대상 기술 스택

- AWS의 VMware Cloud

도구

- [VMware HCX](#) - VMware HCX는 데이터 센터 및 클라우드 환경 전반에서 애플리케이션과 워크로드를 마이그레이션하는 데 사용할 수 있는 도구입니다. AWS의 VMware Cloud에 포함되어 있습니다.

에픽

마이그레이션 계획

작업	설명	필요한 기술
마이그레이션 전략을 선택합니다.	데이터 센터를 확장(하이브리드)할지, 모든 데이터 센터를 이전(클라우드 대피)할지, 아니면 특정 애플리케이션을 AWS로 이전할지 결정합니다.	SysAdmin, 앱 소유자
HCX 요구 사항을 검증합니다.	마이그레이션 정보는 VMware HCX 사용 설명서 를 참조하세요.	SysAdmin, 앱 소유자

AWS의 VMware Cloud로 마이그레이션

작업	설명	필요한 기술
VM 또는 애플리케이션을 마이그레이션합니다.	자세한 내용은 VMware 설명서의 VMware HCX를 이용한 하이브리드 마이그레이션 을 참조하세요.	SysAdmin, 앱 소유자

관련 리소스

- [AWS의 VMware Cloud 시작](#)
- [VMware HCX를 이용한 하이브리드 마이그레이션](#)
- [VMware HCX 사용 설명서](#)

- [AWS의 VMware Cloud 가격](#)
- [AWS의 VMware Cloud 로드맵](#)

Amazon RDS DB 인스턴스를 다른 VPC 또는 계정으로 마이그레이션

작성자: Dhruvajyoti Mukherjee(AWS)

환경: PoC 또는 파일럿	소스: Amazon RDS	대상: Amazon RDS
R 유형: 재배치	기술: 마이그레이션, 데이터베이스	AWS 서비스: Amazon RDS, Amazon VPC

요약

이 패턴은 하나의 Virtual Private Cloud(VPC)에서 동일한 AWS 계정의 다른 Virtual Private Cloud(VPC)로 또는 한 AWS 계정에서 다른 AWS 계정으로 Amazon Relational Database Service(RDS) DB 인스턴스를 마이그레이션하기 위한 지침을 제공합니다.

이 패턴은 분리 또는 보안상의 이유로 Amazon RDS DB 인스턴스를 다른 VPC 또는 계정으로 마이그레이션하려는 경우 유용합니다(예를 들어, 애플리케이션 스택과 데이터베이스를 다른 VPC에 배치하고자 하는 경우).

DB 인스턴스를 다른 AWS 계정으로 마이그레이션하려면 수동 스냅샷을 생성하고, 공유하며 대상 계정에서 스냅샷을 복원하는 등의 단계가 필요합니다. 이 프로세스는 데이터베이스 변경 및 트랜잭션 속도에 따라 시간이 많이 걸릴 수 있습니다. 또한 데이터베이스 다운타임이 발생하므로 마이그레이션을 미리 계획하세요. 다운타임을 최소화하려면 블루/그린 배포 전략을 고려하세요. 또는 AWS Data Migration Service(AWS DMS)를 평가하여 변경으로 인한 다운타임을 최소화할 수 있습니다. 하지만 이 패턴에는 이 옵션이 포함되지 않습니다. 자세한 내용은 [AWS DMS 설명서](#)를 참조하세요.

사전 조건 및 제한 사항

사전 조건

- 활성 상태의 AWS 계정
- VPC, 서브넷 및 Amazon RDS 콘솔에 대해 AWS Identity and Management(IAM) 권한이 필요합니다.

제한 사항

- VPC를 변경하면 데이터베이스가 재부팅되어 애플리케이션이 중단됩니다. 사용량이 적은 시간에 마이그레이션하는 것이 좋습니다.

- Amazon RDS를 다른 VPC로 마이그레이션할 때의 제한 사항:
 - 마이그레이션하는 DB 인스턴스는 대기 인스턴스가 없는 단일 인스턴스여야 합니다. 클러스터의 멤버가 아니어야 합니다.
 - Amazon RDS가 여러 가용 영역에 속해서는 안 됩니다.
 - Amazon RDS에는 읽기 전용 복제본이 없어야 합니다.
 - 대상 VPC에 생성된 서브넷 그룹에는 소스 데이터베이스가 실행되는 가용 영역의 서브넷이 있어야 합니다.
- Amazon RDS를 다른 AWS 계정으로 마이그레이션할 때의 제한 사항:
 - Amazon RDS용 기본 서비스 키로 암호화된 스냅샷 공유는 현재 지원되지 않습니다.

아키텍처

동일한 AWS 계정의 VPC로 마이그레이션

다음 다이어그램은 Amazon RDS DB 인스턴스를 동일한 AWS 계정의 다른 VPC로 마이그레이션하는 워크플로우를 보여줍니다.

단계는 다음과 같이 구성되어 있습니다. 자세한 지침은 [에픽](#) 섹션을 참조하세요.

1. 대상 VPC에 DB 서브넷 그룹을 생성합니다. DB 서브넷 그룹은 DB 인스턴스를 만들 때 특정 VPC를 지정하는 데 사용할 수 있는 서브넷 모음입니다.
2. 새 DB 서브넷 그룹을 사용하도록 소스 VPC의 Amazon RDS DB 인스턴스를 구성합니다.
3. 변경 내용을 적용하여 Amazon RDS DB를 대상 VPC로 마이그레이션합니다.

다른 AWS 계정으로 마이그레이션

다음 다이어그램은 Amazon RDS DB 인스턴스를 다른 AWS 계정으로 마이그레이션하는 워크플로우를 보여줍니다.

단계는 다음과 같이 구성되어 있습니다. 자세한 지침은 [에픽](#) 섹션을 참조하세요.

1. 소스 AWS 계정에서 Amazon RDS DB 인스턴스에 액세스합니다.
2. 소스 AWS 계정에서 Amazon RDS 스냅샷을 생성합니다.
3. Amazon RDS 스냅샷을 대상 AWS 계정과 공유합니다.

4. 대상 AWS 계정에서 Amazon RDS 스냅샷에 액세스합니다.
5. 대상 AWS 계정에서 Amazon RDS DB 인스턴스를 생성합니다.

도구

서비스

- [Amazon Relational Database Service\(Amazon RDS\)](#)는 AWS 클라우드에서 관계형 데이터베이스를 설정, 운영 및 조정하는 데 도움이 됩니다.
- [Amazon Virtual Private Cloud\(VPC\)](#)를 이용하면 사용자가 정의한 가상 네트워크로 AWS 리소스를 시작할 수 있습니다. 이 가상 네트워크는 사용자의 자체 데이터 센터에서 운영하는 기존 네트워크와 유사하며 AWS의 확장 가능한 인프라를 사용한다는 이점이 있습니다.

모범 사례

- Amazon RDS DB 인스턴스를 다른 계정으로 마이그레이션할 때 데이터베이스 다운타임이 우려되는 경우 [AWS DMS](#)를 사용하는 것이 좋습니다. 이 서비스는 데이터 복제를 제공하므로 중단 시간이 5분 미만입니다.

에픽

동일한 AWS 계정의 다른 VPC로 마이그레이션

작업	설명	필요한 기술
새 VPC를 생성합니다.	Amazon VPC 콘솔 에서 원하는 속성과 IP 주소 범위가 있는 새 VPC와 서브넷을 생성합니다. 자세한 지침은 Amazon VPC 설명서 를 참조하세요.	관리자
DB 서브넷 그룹을 생성합니다.	Amazon RDS 콘솔 에서: <ol style="list-style-type: none"> 1. 서브넷 그룹, DB 서브넷 그룹 생성을 선택합니다. 2. 서브넷 그룹 이름, 설명 및 VPC ID를 입력합니다. 	관리자

작업	설명	필요한 기술
	<p>3. 서브넷 그룹에 속하는 서브넷을 추가합니다. 2개 이상의 가용 영역을 다루도록 서브넷을 추가합니다.</p> <p>4. 생성을 선택하세요.</p> <p>자세한 내용은 Amazon RDS 설명서를 참조하세요.</p>	

작업	설명	필요한 기술
<p>Amazon RDS DB 인스턴스가 새 서브넷 그룹을 사용하도록 수정합니다.</p>	<p>Amazon RDS 콘솔에서:</p> <ol style="list-style-type: none"> 1. 탐색 창에서 데이터베이스를 선택한 후 마이그레이션하려는 Amazon RDS DB 인스턴스를 선택합니다. 2. 연결 섹션에서 대상 VPC와 연결된 서브넷 그룹을 선택합니다. 3. 수정 예약 섹션에서 즉시 적용을 선택합니다. <p>대상 VPC로의 마이그레이션이 완료되면 대상 VPC의 기본 보안 그룹이 Amazon RDS DB 인스턴스에 할당됩니다. DB 인스턴스에 대한 필수 인바운드 및 아웃바운드 규칙으로 해당 VPC에 대한 새 보안 그룹을 구성할 수 있습니다.</p> <p>또는 AWS Command Line Interface(AWS CLI)를 사용하여 새 VPC 보안 그룹 ID를 명시적으로 제공함으로써 대상 VPC에 마이그레이션을 수행합니다. 예:</p> <pre>aws rds modify-db-instance \ --db-instance-identifier testrds \ --db-subnet-group-name new-vpc-subnet-group \</pre>	<p>관리자</p>

작업	설명	필요한 기술
	<pre>--vpc-security-group-ids sg-idxxxx \ --apply-immediately</pre>	

다른 AWS 계정으로 마이그레이션

작업	설명	필요한 기술
대상 AWS 계정에서 새 VPC 및 서브넷 그룹 생성	<ol style="list-style-type: none"> 1. Amazon VPC 콘솔에서 원하는 속성과 IP 주소 범위가 있는 새 VPC를 생성합니다. 자세한 지침은 Amazon VPC 설명서를 참조하세요. 2. Amazon VPC 설명서의 지침에 따라 새 VPC를 위한 서브넷을 생성합니다. 3. Amazon RDS 콘솔에서 DB 서브넷 그룹을 생성합니다. 자세한 지침은 Amazon RDS 설명서를 참조하세요. 	관리자
데이터베이스의 수동 스냅샷을 공유하고 대상 계정과 공유합니다.	<ol style="list-style-type: none"> 1. Amazon RDS 설명서의 지침에 따라 소스 데이터베이스의 수동 스냅샷을 생성합니다. 2. 대상 계정 ID를 제공하여 대상 AWS 계정과 스냅샷을 공유합니다. 지침은 DB 스냅샷을 다른 계정과 공유하는 방법에 대한 re:Post article를 참조하세요. 	관리자

작업	설명	필요한 기술
새 Amazon RDS DB 인스턴스를 시작합니다.	대상 AWS 계정의 공유 스냅샷에서 새 Amazon RDS DB 인스턴스를 시작합니다. 자세한 지침은 Amazon RDS 설명서 를 참조하세요.	관리자

관련 리소스

- [Amazon VPC 설명서](#)
- [Amazon RDS 설명서](#)
- [RDS DB 인스턴스의 VPC를 변경하려면 어떻게 해야 하나요? \(AWS re:Post article\)](#)
- [Amazon RDS 리소스의 소유권을 다른 AWS 계정으로 이전하려면 어떻게 해야 하나요? \(AWS re:Post article\)](#)
- [수동 Amazon RDS DB 스냅샷 또는 Aurora DB 클러스터 스냅샷을 다른 AWS 계정과 공유하려면 어떻게 해야 하나요? \(AWS re:Post article\)](#)
- [AWS DMS 설명서](#)

Amazon RDS for Oracle DB 인스턴스를 다른 VPC로 마이그레이션

작성자: Pinesh Singal(AWS)

환경: PoC 또는 파일럿	소스: 데이터베이스: 관계형	대상: Amazon RDS for Oracle
R 유형: 재배치	워크로드: Oracle	기술: 마이그레이션, 데이터베이스

AWS 서비스: Amazon RDS

요약

이 마이그레이션 패턴은 오라클 데이터베이스 (DB) 용 Amazon RDS (Amazon RDS) 인스턴스를 하나의 가상 사설 클라우드 (VPC) 에서 동일한 Amazon Web Services (AWS) 계정의 다른 VPC로 마이그레이션하기 위한 step-by-step 지침을 제공합니다. 예를 들어 데이터베이스와 Amazon Elastic Compute Cloud(Amazon EC2) 애플리케이션 서버가 동일한 VPC에 있어야 하는 경우 이 패턴을 사용할 수 있습니다.

이 패턴은 트랜잭션 수가 많은 수 테라바이트의 Oracle 소스 데이터베이스에 대해 다운타임이 거의 없는 온라인 수동 마이그레이션 전략을 설명합니다.

Amazon RDS for Oracle의 DB 인스턴스를 다른 VPC로 이동하려면 Amazon RDS 서브넷 그룹을 변경해야 합니다. 이 서브넷 그룹은 새 VPC와 필요한 서브넷으로 사전 구성되어야 합니다. VPC가 한 네트워크에서 다른 네트워크로 변경되는 동안 Amazon RDS 인스턴스가 재부팅되므로 이동이 진행되는 동안에는 데이터베이스에 액세스할 수 없습니다.

사전 조건 및 제한 사항

사전 조건

- 활성 상태의 AWS 계정
- 프라이빗 서브넷이 있는 VPC 2개
- 인바운드 및 아웃바운드 보안 그룹으로 구성된 Oracle용 Amazon RDS 데이터베이스 인스턴스(가동 및 실행 중)

제한 사항

- 여러 가용 영역(다중 AZ)에 걸친 DB 인스턴스는 지원되지 않습니다. 하지만 이 패턴은 이러한 제한을 우회할 수 있는 방법을 제공합니다.
- 읽기 전용 복제본이 켜져 있는 동안에는 DB 인스턴스를 마이그레이션할 수 없습니다.
- 새 VPC의 서브넷 그룹은 데이터베이스와 같은 가용 영역에 있어야 합니다.
- DB를 다른 VPC로 이동하면 데이터베이스가 재부팅되어 몇 분 동안 애플리케이션 중단이 발생하기 때문에 예약된 유지 관리 기간이나 트래픽이 적은 시간에 마이그레이션을 해야 합니다.

제품 버전

- Amazon RDS for Oracle DB 인스턴스, 12.1.0.2 및 이후

아키텍처

소스 기술 스택

- VPC에 있는 Amazon RDS for Oracle의 12.1.0.2 v22 DB 인스턴스
- 별도의 라우팅 테이블에 구성된 VPC
- VPC에 구성된 Amazon RDS 서브넷 그룹
- Amazon RDS 옵션 그룹(필요한 경우)

대상 기술 스택

- 또 다른 VPC에 있는 Amazon RDS for Oracle 데이터베이스 인스턴스 버전 12.1.0.2.v22
- 별도의 라우팅에 구성된 Amazon VPC
- 새 VPC에 구성된 Amazon RDS 서브넷 그룹
- Amazon RDS 옵션 그룹(필요한 경우)

소스 및 대상 아키텍처

다음 다이어그램은 콘솔을 사용하여 Amazon RDS for Oracle DB를 한 VPC의 프라이빗 서브넷에서 다른 VPC의 프라이빗 서브넷으로 이동하는 것을 보여줍니다.

1. 콘솔을 사용해서 Amazon RDS for Oracle DB 인스턴스를 수정합니다.

2. 대상 VPC에서 서브넷 그룹을 수정하고, 옵션 그룹을 사용하는 경우 수정합니다.

도구

- [Amazon RDS](#) - Amazon Relational Database Service(RDS)는 AWS 클라우드의 관계형 데이터베이스를 더 쉽게 설치, 운영 및 규모 조정할 수 있게 하는 웹 서비스입니다. 이 서비스는 관계형 데이터베이스를 위한 경제적이고 크기 조절이 가능한 용량을 제공하고 공통 데이터베이스 관리 작업을 관리합니다. 이 패턴은 Amazon RDS for Oracle을 사용합니다.

에픽

기존 VPC에서 Amazon RDS for Oracle 데이터베이스의 구성을 변경합니다.

작업	설명	필요한 기술
서브넷 그룹을 생성하십시오.	Amazon RDS에서 서브넷 그룹을 구성합니다.	일반 AWS
옵션 그룹을 생성합니다.	(선택 사항) Amazon RDS에서 옵션 그룹을 구성합니다.	일반 AWS
Amazon RDS for Oracle DB 인스턴스를 수정합니다.	서브넷 그룹과 옵션 그룹으로 데이터베이스를 수정합니다.	일반 AWS, DBA
필요한 경우 Oracle 데이터베이스를 업데이트하십시오.	소스 Amazon RDS for Oracle 용 데이터베이스를 마이그레이션하려면 다음과 같이 변경하십시오. <ul style="list-style-type: none"> • 읽기 전용 복제본이 있는 경우 제거하십시오. • 다중 AZ 기능이 켜져 있는 경우 해당 기능을 끄십시오. 	일반 AWS

대상 VPC에서 Amazon RDS for Oracle 데이터베이스 구성

작업	설명	필요한 기술
서브넷 그룹을 생성하십시오.	Amazon RDS에서 새 VPC의 서브넷과 데이터베이스의 가용 영역을 사용하여 서브넷 그룹을 구성합니다.	일반 AWS
옵션 그룹을 생성합니다.	(선택 사항) Amazon RDS에서 옵션 그룹을 구성합니다.	일반 AWS
Amazon RDS for Oracle 데이터베이스를 수정합니다.	<p>새 서브넷 그룹과 새 VPC의 옵션 그룹으로 데이터베이스를 수정합니다. 이러한 변경 사항은 즉시 적용하거나 유지 관리 기간에 적용할 수 있습니다.</p> <p>수정이 완료되는 데 몇 분 정도 걸릴 수 있습니다. 수정하는 동안 다음과 같은 상태 변경 사항을 확인할 수 있습니다.</p> <ul style="list-style-type: none"> • moving-to-vpc • C. onfiguring-enhanced-monitoring • 수정 중 • 사용 가능 <p>수정하면 새 VPC의 기본 보안 그룹이 연결됩니다. Amazon RDS for Oracle에서 필요한 대로 새 보안 그룹을 연결합니다.</p>	일반 AWS, DBA
필요한 경우 Amazon RDS for Oracle 데이터베이스를 업데이트하십시오.	새 VPC의 대상 Amazon RDS for Oracle용 데이터베이스로	일반 AWS

작업	설명	필요한 기술
	<p>마이그레이션한 후 필요한 경우 다음과 같이 수정하십시오.</p> <ul style="list-style-type: none"> 소스 데이터베이스에 읽기 전용 복제본이 있는 경우 이를 활성화하십시오. 소스 데이터베이스에서 다중 AZ 기능이 켜져 있는 경우 해당 기능을 켜십시오. 	
<p>애플리케이션 연결성을 테스트합니다.</p>	<p>아무 애플리케이션에서나 데이터베이스 연결 테스트를 수행하십시오. 새 VPC에 수정된 Amazon RDS for Oracle DB가 연결되어 있고 애플리케이션에서 액세스할 수 있는지 확인합니다.</p>	<p>앱 소유자</p>

관련 리소스

- [Amazon VPC 설명서](#)
- [VPC 및 서브넷](#)
- [VPC에서 DB 인스턴스를 사용한 작업](#)
- [Amazon RDS 설명서](#)
- [Amazon RDS의 Oracle](#)
- [Amazon RDS 콘솔](#)
- [Amazon RDS DB 인스턴스의 VPC를 변경하려면 어떻게 해야 하나요?](#)

Amazon Redshift 클러스터를 중국의 AWS 리전으로 마이그레이션

작성자: Jing Yan(AWS)

R 유형: 재배치	환경: 프로덕션	기술: 데이터베이스, 마이그레이션
워크로드: 기타 모든 워크로드	AWS 서비스: Amazon Redshift	소스: AWS Redshift
대상: AWS Redshift		

요약

이 패턴은 Amazon Redshift 클러스터를 다른 AWS 지역에서 중국의 AWS 지역으로 마이그레이션하는 step-by-step 접근 방식을 제공합니다.

이 패턴은 SQL 명령을 사용하여 모든 데이터베이스 객체를 다시 생성하고 UNLOAD 명령을 사용하여 Amazon Redshift에서 소스 리전의 Amazon Simple Storage Service(S3) 버킷으로 해당 데이터를 이동합니다. 그런 다음 데이터는 중국 AWS 리전의 S3 버킷으로 마이그레이션됩니다. COPY 명령은 S3 버킷에서 데이터를 로드하여 대상 Amazon Redshift 클러스터로 전송하는 데 사용됩니다.

Amazon Redshift는 현재 중국 AWS 리전으로의 스냅샷 복사와 같은 리전 간 기능을 지원하지 않습니다. 이 패턴은 이러한 제한을 우회할 수 있는 방법을 제공합니다. 또한 이 패턴의 단계를 반대로 하여 중국의 AWS 리전에서 다른 AWS 리전으로 데이터를 마이그레이션할 수 있습니다.

사전 조건 및 제한 사항

사전 조건

- 중국 리전 및 중국 외 AWS 리전 모두의 활성 AWS 계정
- 중국 리전과 중국 외 AWS 리전 모두에 있는 기존 Amazon Redshift 클러스터

제한 사항

- 이는 오프라인 마이그레이션이므로 소스 Amazon Redshift 클러스터는 마이그레이션 중에 쓰기 작업을 수행할 수 없습니다.

아키텍처

소스 기술 스택

- 중국 외 AWS 리전의 Amazon Redshift 클러스터

대상 기술 스택

- 중국 AWS 리전의 Amazon Redshift 클러스터

대상 아키텍처

도구

도구

- [Amazon S3](#) - Amazon Simple Storage Service(S3)는 확장성, 데이터 가용성, 보안 및 성능을 제공하는 객체 스토리지 서비스입니다. Amazon S3를 사용하여 Amazon Redshift의 데이터를 저장할 수 있으며, S3 버킷에서 Amazon Redshift로 데이터를 복사할 수 있습니다.
- [Amazon Redshift](#) - Amazon Redshift는 클라우드에서 완벽하게 관리되는 페타바이트 규모의 데이터 웨어하우스 서비스입니다.
- [psql](#) - psql은 PostgreSQL의 터미널 기반 프론트엔드입니다.

에픽

소스 리전에서 마이그레이션 준비

작업	설명	필요한 기술
소스 리전에서 EC2 인스턴스를 시작하고 구성합니다.	AWS Management Console에 로그인하고 Amazon Elastic Compute Cloud(Amazon EC2) 콘솔을 엽니다. 화면 상단의 탐색 모음에는 현재 리전이 표시됩니다. 이 리전은 중국의 AWS 리전이 될 수 없습니다.	DBA, 개발자

작업	설명	필요한 기술
	<p>Amazon EC2 콘솔 대시보드에서 '인스턴스 시작'을 선택하고 EC2 인스턴스를 생성 및 구성합니다. 중요: 인바운드 규칙의 EC2 보안 그룹이 소스 머신에서 TCP 포트 22에 대한 제한 없는 액세스를 허용하는지 확인하십시오. EC2 인스턴스를 시작하고 구성하는 방법에 대한 지침은 '관련 리소스' 섹션을 참조하십시오.</p>	
psql 도구를 설치합니다.	<p>PostgreSQL을 다운로드하여 설치합니다. Amazon Redshift는 psql 도구를 제공하지 않고, PostgreSQL과 함께 설치됩니다. psql 사용 및 PostgreSQL 도구 설치에 대한 자세한 내용은 '관련 리소스' 섹션을 참조하십시오.</p>	DBA

작업	설명	필요한 기술
Amazon Redshift 클러스터 세부 정보를 기록하십시오.	Amazon Redshift 콘솔을 열고 탐색 창에서 '클러스터'를 선택합니다. 그런 다음 목록에서 Amazon Redshift 클러스터 이름을 선택합니다. '속성' 탭의 '데이터베이스 구성' 섹션에서 '데이터베이스 이름' 및 '포트'를 기록합니다. '연결 세부 정보' 섹션을 열고 '엔드포인트:<port>/<databasename>' 형식의 '엔드포인트'를 기록합니다. 중요: 인바운드 규칙에 대한 Amazon Redshift 보안 그룹이 EC2 인스턴스에서 TCP 포트 5439에 대한 제한 없는 액세스를 허용하는지 확인하십시오.	DBA
psql을 Amazon Redshift 클러스터에 연결합니다.	명령 프롬프트에서 'psql -h <endpoint> -U <userid> -d <databasename> -p <port>' 명령을 실행하여 연결 정보를 지정합니다. psql 암호 프롬프트에서 '<userid>' 사용자의 암호를 입력합니다. 그러면 Amazon Redshift 클러스터에 연결되어 명령을 대화식으로 입력할 수 있습니다.	DBA
S3 버킷을 생성합니다.	Amazon S3 콘솔을 열고, Amazon Redshift에서 내보낸 파일을 보관할 S3 버킷을 생성합니다. S3 버킷을 생성하는 방법에 대한 지침은 '관련 리소스' 섹션을 참조하십시오.	DBA, 일반 AWS

작업	설명	필요한 기술
<p>데이터 언로드를 지원하는 IAM 정책을 생성합니다.</p>	<p>AWS Identity and Access Management(IAM) 콘솔을 열고 '정책'을 선택합니다. '정책 생성'을 선택한 후 'JSON' 탭을 선택합니다. '추가 정보' 섹션에서 데이터 언로드를 위한 IAM 정책을 복사하여 붙여 넣습니다. 중요: 's3_bucket_name'을 S3 버킷 이름으로 바꾸십시오. '정책 검토'를 선택하고 정책 이름과 설명을 입력합니다. '정책 생성'을 선택합니다.</p>	<p>DBA</p>
<p>IAM 역할을 생성하여 Amazon Redshift에서 UNLOAD 작업을 허용합니다.</p>	<p>IAM 콘솔을 열고 '역할'을 선택합니다. '역할 생성'을 선택하고 '신뢰할 수 있는 기관 유형 선택'에서 'AWS 서비스'를 선택합니다. 서비스에 대해 'Redshift'를 선택하고 'Redshift - 사용자 지정 가능'을 선택한 후 '다음'을 선택합니다. 이전에 생성한 '언로드' 정책을 선택하고 '다음'을 선택합니다. '역할 이름'을 입력하고 '역할 생성'을 선택합니다.</p>	<p>DBA</p>

작업	설명	필요한 기술
IAM 역할을 Amazon Redshift 클러스터에 연결하십시오.	Amazon Redshift 콘솔을 열고 'IAM 역할 관리'를 선택합니다. 드롭다운 메뉴에서 '사용 가능한 역할'을 선택하고 이전에 생성한 역할을 선택합니다. '변경 사항 적용'을 선택합니다. 'IAM 역할 관리'에서 IAM 역할의 '상태'가 '동기화 중'으로 표시되면 UNLOAD 명령을 실행할 수 있습니다.	DBA
Amazon Redshift 클러스터에 대한 쓰기 작업을 중지합니다.	마이그레이션이 완료될 때까지 소스 Amazon Redshift 클러스터에 대한 모든 쓰기 작업을 중지해야 합니다.	DBA

대상 리전에서 마이그레이션 준비

작업	설명	필요한 기술
대상 리전에서 EC2 인스턴스를 시작하고 구성합니다.	중국 리전(베이징 또는 닝샤)의 AWS Management Console에 로그인하십시오. Amazon EC2 콘솔에서 '인스턴스 시작'을 선택하고 EC2 인스턴스를 생성 및 구성합니다. 중요: 인바운드 규칙의 Amazon EC2 보안 그룹이 소스 머신에서 TCP 포트 22에 대한 무제한 액세스를 허용하는지 확인하십시오. EC2 인스턴스를 시작하고 구성하는 방법에 대한 자세한 지침은 '관련 리소스' 섹션을 참조하십시오.	DBA

작업	설명	필요한 기술
<p>Amazon Redshift 클러스터 세부 정보를 기록하십시오.</p>	<p>Amazon Redshift 콘솔을 열고 탐색 창에서 '클러스터'를 선택합니다. 그런 다음 목록에서 Amazon Redshift 클러스터 이름을 선택합니다. '속성' 탭의 '데이터베이스 구성' 섹션에서 '데이터베이스 이름' 및 '포트'를 기록합니다. '연결 세부 정보' 섹션을 열고 '엔드포인트:<port>/<databasename>' 형식의 '엔드포인트'를 기록합니다. 중요: 인바운드 규칙의 Amazon Redshift 보안 그룹이 EC2 인스턴스에서 TCP 포트 5439에 대한 무제한 액세스를 허용하는지 확인하십시오.</p>	<p>DBA</p>
<p>psql을 Amazon Redshift 클러스터에 연결합니다.</p>	<p>명령 프롬프트에서 'psql -h <endpoint> -U <userid> -d <databasename> -p <port>' 명령을 실행하여 연결 정보를 지정합니다. psql 암호 프롬프트에서 '<userid>' 사용자의 암호를 입력합니다. 그러면 Amazon Redshift 클러스터에 연결되어 명령을 대화식으로 입력할 수 있습니다.</p>	<p>DBA</p>
<p>S3 버킷을 생성합니다.</p>	<p>Amazon S3 콘솔을 열고, Amazon Redshift에서 내보낸 파일을 보관할 S3 버킷을 생성합니다. 이 스토리와 다른 스토리에 대한 도움이 필요하면 "관련 리소스" 섹션을 참조하십시오.</p>	<p>DBA</p>

작업	설명	필요한 기술
<p>데이터 복사를 지원하는 IAM 정책을 생성합니다.</p>	<p>IAM 콘솔을 열고 '정책'을 선택합니다. '정책 생성'을 선택한 후 'JSON' 탭을 선택합니다. '추가 정보' 섹션에서 데이터 복사를 위한 IAM 정책을 복사하여 붙여 넣습니다. 중요: 's3_bucket_name'을 S3 버킷 이름으로 바꾸십시오. '정책 검토'를 선택, 정책 이름과 설명을 입력합니다. '정책 생성'을 선택합니다.</p>	<p>DBA</p>
<p>IAM 역할을 생성하여 Amazon Redshift에서 COPY 작업을 허용합니다.</p>	<p>IAM 콘솔을 열고 '역할'을 선택합니다. '역할 생성'을 선택하고 '신뢰할 수 있는 기관 유형 선택'에서 'AWS 서비스'를 선택합니다. 서비스에 대해 'Redshift'를 선택하고 'Redshift - 사용자 지정 가능'을 선택한 후 '다음'을 선택합니다. 이전에 생성한 '복사' 정책을 선택하고 '다음'을 선택합니다. '역할 이름'을 입력하고 '역할 생성'을 선택합니다.</p>	<p>DBA</p>
<p>IAM 역할을 Amazon Redshift 클러스터에 연결하십시오.</p>	<p>Amazon Redshift 콘솔을 열고 'IAM 역할 관리'를 선택합니다. 드롭다운 메뉴에서 '사용 가능한 역할'을 선택하고 이전에 생성한 역할을 선택합니다. '변경 사항 적용'을 선택합니다. 'IAM 역할 관리'에서 IAM 역할의 '상태'가 '동기화 중'으로 표시되면 'COPY' 명령을 실행할 수 있습니다.</p>	<p>DBA</p>

마이그레이션을 시작하기 전에 소스 데이터와 객체 정보를 확인하십시오.

작업	설명	필요한 기술
소스 Amazon Redshift 테이블의 행을 확인하십시오.	‘추가 정보’ 섹션의 스크립트를 사용하여 소스 Amazon Redshift 테이블의 행 수를 확인하고 기록합니다. UNLOAD 스크립트와 COPY 스크립트의 데이터를 균등하게 분할해야 한다는 점을 기억하십시오. 이렇게 하면 각 스크립트에서 다루는 데이터 양이 균형을 이루기 때문에 데이터 언로드 및 로드 효율성이 향상됩니다.	DBA
소스 Amazon Redshift 클러스터의 데이터베이스 객체 수를 확인합니다.	‘추가 정보’ 섹션의 스크립트를 사용하여 소스 Amazon Redshift 클러스터의 데이터베이스, 사용자, 스키마, 테이블, 보기 및 사용자 정의 함수 (UDF) 수를 확인하고 기록할 수 있습니다.	DBA
마이그레이션하기 전에 SQL 문 결과를 확인하십시오.	데이터 검증을 위한 일부 SQL 명령문은 실제 비즈니스 및 데이터 상황에 따라 정렬해야 합니다. 이는 가져온 데이터가 일관되고 올바르게 표시되는지 확인하기 위한 것입니다.	DBA

데이터 및 객체를 대상 리전으로 마이그레이션

작업	설명	필요한 기술
Amazon Redshift DDL 스크립트를 생성합니다.	‘추가 정보’ 섹션의 ‘Amazon Redshift를 쿼리하기 위한 SQL	DBA

작업	설명	필요한 기술
	<p>명령문' 섹션의 링크를 사용하여 데이터 정의 언어(DDL) 스크립트를 생성합니다. 이러한 DDL 스크립트에는 '사용자 생성', '스키마 생성', '사용자에 대한 스키마 권한', '테이블/보기 생성', '객체에 대한 사용자 권한' 및 '함수 생성' 쿼리가 포함되어야 합니다.</p>	
<p>Amazon Redshift 클러스터에서 대상 리전의 객체를 생성합니다.</p>	<p>중국 AWS 리전에서 AWS Command Line Interface(AWS CLI)를 사용하여 DDL 스크립트를 실행합니다. 이 스크립트는 Amazon Redshift 클러스터에서 대상 리전의 객체를 생성합니다.</p>	<p>DBA</p>
<p>소스 Amazon Redshift 클러스터 데이터를 S3 버킷으로 언로드합니다.</p>	<p>UNLOAD 명령을 실행하여 소스 리전의 Amazon Redshift 클러스터에서 S3 버킷으로 데이터를 언로드합니다.</p>	<p>DBA, 개발자</p>
<p>소스 리전 S3 버킷 데이터를 대상 리전 S3 버킷으로 전송합니다.</p>	<p>소스 리전 S3 버킷의 데이터를 대상 S3 버킷으로 전송합니다. '\$ aws s3 sync' 명령은 사용할 수 없으므로 '관련 리소스' 섹션의 'AWS 리전에서 중국 AWS 리전으로 Amazon S3 데이터 전송' 문서에 설명된 프로세스를 사용해야 합니다.</p>	<p>개발자</p>

작업	설명	필요한 기술
대상 Amazon Redshift 클러스터로 데이터를 로드합니다.	대상 리전의 psql 도구에서 COPY 명령을 실행하여 S3 버킷의 데이터를 대상 Amazon Redshift 클러스터로 로드합니다.	DBA

마이그레이션 후 소스 및 대상 리전의 데이터 확인

작업	설명	필요한 기술
소스 및 대상 테이블의 행 수를 확인하고 비교합니다.	소스 및 대상 리전의 테이블 행 수를 확인하고 비교하여 모두 마이그레이션되었는지 확인하십시오.	DBA
소스 및 대상 데이터베이스 객체 수를 확인하고 비교합니다.	소스 및 대상 리전의 모든 데이터베이스 객체를 확인하고 비교하여 모두 마이그레이션되었는지 확인합니다.	DBA
소스 및 대상 리전의 SQL 스크립트 결과를 확인하고 비교합니다.	마이그레이션 전에 준비된 SQL 스크립트를 실행합니다. 데이터를 확인하고 비교하여 SQL 결과가 정확한지 확인하십시오.	DBA
대상 Amazon Redshift 클러스터에 있는 모든 사용자의 암호를 재설정합니다.	마이그레이션이 완료되고 모든 데이터가 확인되면 중국 AWS 리전의 Amazon Redshift 클러스터에 대한 모든 사용자 암호를 재설정해야 합니다.	DBA

관련 리소스

- [AWS 리전에서 중국의 AWS 리전으로 Amazon S3 데이터 전송](#)
- [S3 버킷 생성](#)
- [Amazon Redshift 사용자 비밀번호 재설정](#)
- [psql 설명서](#)

추가 정보

데이터 언로드를 위한 IAM 정책

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": ["s3:ListBucket"],
      "Resource": ["arn:aws:s3:::s3_bucket_name"]
    },
    {
      "Effect": "Allow",
      "Action": ["s3:GetObject", "s3:DeleteObject"],
      "Resource": ["arn:aws:s3:::s3_bucket_name/*"]
    }
  ]
}
```

데이터 복사를 위한 IAM 정책

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": ["s3:ListBucket"],
      "Resource": ["arn:aws:s3:::s3_bucket_name"]
    },
    {
      "Effect": "Allow",
      "Action": ["s3:GetObject"],
      "Resource": ["arn:aws:s3:::s3_bucket_name/*"]
    }
  ]
}
```

```
    }  
  ]  
}
```

Amazon Redshift를 쿼리하기 위한 SQL 명령문

```
##Database  
  
select * from pg_database where datdba>1;  
  
##User  
  
select * from pg_user where usesysid>1;  
  
##Schema  
  
SELECT n.nspname AS "Name",  
       pg_catalog.pg_get_userbyid(n.nspowner) AS "Owner"  
  
FROM pg_catalog.pg_namespace n  
  
WHERE n.nspname !~ '^pg_' AND n.nspname <> 'information_schema'  
  
ORDER BY 1;  
  
##Table  
  
select count(*) from pg_tables where schemaname not in  
('pg_catalog','information_schema');  
  
select schemaname,count(*) from pg_tables where schemaname not in  
('pg_catalog','information_schema') group by schemaname order by 1;  
  
##View  
  
SELECT  
  
       n.nspname AS schemaname,c.relname AS  
viewname,pg_catalog.pg_get_userbyid(c.relowner) as "Owner"  
  
FROM
```

```
pg_catalog.pg_class AS c

INNER JOIN

pg_catalog.pg_namespace AS n

ON c.relnamespace = n.oid

WHERE relkind = 'v' and n.nspname not in ('information_schema','pg_catalog');

##UDF

SELECT

n.nspname AS schemaname,

p.proname AS proname,

pg_catalog.pg_get_userbyid(p.proowner) as "Owner"

FROM pg_proc p

LEFT JOIN pg_namespace n on n.oid = p.pronamespace

WHERE p.proowner != 1;
```

DDL 문을 생성하는 SQL 스크립트

- [Get_schema_priv_by_user script](#)
- [Generate_tbl_ddl script](#)
- [Generate_view_ddl](#)
- [Generate_user_grant_revoke_ddl](#)
- [Generate_udf_ddl](#)

VMware HCX를 사용하여 워크로드를 AWS의 VMware Cloud로 마이그레이션합니다.

작성자: 디팍 쿠마르 (AWS), 데릭 콕스 (AWS), 히만슈 굽타 (AWS)

환경: 프로덕션	소스: 온프레미스 VMware 워크로드	대상: AWS의 VMware Cloud
R 유형: 재배치	워크로드: 기타 모든 워크로드	기술: 마이그레이션, 하이브리드 클라우드

AWS 서비스: AWS의 VMware Cloud, Amazon VPC

요약

이 패턴은 기본 플랫폼을 변경하지 않고 VMware 하이브리드 클라우드 확장(HCX)을 사용하여 온프레미스 VMware 환경에서 AWS의 VMware Cloud로 워크로드를 마이그레이션하는 방법을 설명합니다. VMware HCX는 마이그레이션을 간소화하고, 워크로드를 재조정하고, 데이터를 보호하고, 온프레미스 데이터 센터와 클라우드 서버 모두의 재해 복구 프로세스를 최적화합니다. 패턴은 HCX의 설치, 구성, 업그레이드 및 제거 단계를 설명합니다.

HCX는 다음 사항을 지원합니다.

- 이전 버전의 VMware vSphere - HCX를 사용하면 가상 머신(VM)을 이전 버전의 vSphere에서 AWS의 VMware Cloud로 마이그레이션할 수 있습니다. 호스트는 자동으로 업데이트되고 복구되므로 마이그레이션 준비 과정에서 시간이 많이 걸리는 업데이트를 생략할 수 있습니다.
- 대량 마이그레이션 - HCX와 WAN 최적화 서비스를 함께 사용하면 많은 수의 VM을 다운타임 없이 한 번에 마이그레이션하여 온프레미스 네트워크를 클라우드로 확장할 수 있습니다.
- 이중 네트워크 환경 - 현재 네트워크(예: vSphere, NSX, VXLAN 또는 NSX-T)에 따라 마이그레이션의 복잡성이 결정됩니다. HCX는 복잡한 절차 없이 네트워크 애플리케이션의 기초를 추출하여 현재 네트워크를 클라우드로 확장합니다.
- 느린 네트워크 속도 - 마이그레이션에는 일반적으로 250Mbps 이상의 연결 속도가 필요합니다. HCX는 약 100Mbps의 훨씬 낮은 속도로 워크로드를 마이그레이션할 수 있습니다.

HCX는 다음 세 가지 유형의 클라우드 마이그레이션을 지원합니다.

- 하이브리드(데이터 센터 확장) - 기존의 온프레미스 VMware 정의된 데이터 센터(SDDC)를 클라우드로 확장하여 설치 공간 확장, 주문형 용량, 테스트/개발 환경 및 가상 데스크톱을 제공합니다.
- 클라우드 대피(데이터 센터 전체 인프라 교체) - 데이터 센터를 통합하고 AWS 클라우드로 완전히 이동합니다(데이터 센터 공동 배치 또는 임대 종료 처리 포함).
- 애플리케이션별 마이그레이션 - 특정 비즈니스 요구 사항을 충족하기 위해 개별 애플리케이션을 AWS 클라우드로 이동합니다.

HCX를 사용하여 온프레미스 환경과 VMware Cloud on AWS 간에 양방향으로 워크로드를 마이그레이션할 수 있습니다. HCX는 소스 위치와 대상 위치 간에 워크로드를 마이그레이션할 수 있는 다양한 방법을 제공합니다.

- HCX 콜드 마이그레이션은 오프라인 상태인 VM을 마이그레이션합니다. 이 방법은 상당한 다운타임이 필요하므로 전원이 꺼진 VM에 적합합니다.
- HCX vMotion은 VMware vMotion 프로토콜을 사용하여 가상 머신을 이동합니다. HCX vMotion은 다운타임 없는 마이그레이션을 제공하지만 한 번에 하나의 VM만 마이그레이션할 수 있습니다.
- HCX Bulk Migration은 VMware vSphere 복제 프로토콜을 사용하여 VM을 대상으로 이동합니다. 여러 VM을 병렬로 마이그레이션하고 전환을 예약할 수 있습니다. 다운타임은 서버 재부팅과 동일하며 모든 VM의 전환이 병렬로 발생합니다.
- HCX Replication Assisted vMotion(RAV)은 HCX 대량 마이그레이션과 HCX vMotion의 조합입니다. 병렬 마이그레이션, 스케줄링 및 제로 다운타임을 제공합니다.
- HCX OS Assisted Migration을 사용하면 온프레미스에서 여러 하이퍼바이저 및 vSphere 이외의 VM을 사용할 때 여러 VM을 대량으로 마이그레이션할 수 있습니다. HCX OS 지원 마이그레이션은 온프레미스에서 AWS의 VMware Cloud로 마이그레이션하는 데 사용할 경우 무료이지만, 두 온프레미스 환경 간에 마이그레이션하거나 온프레미스에서 다른 클라우드 공급자로 마이그레이션하려는 경우 추가 라이선스가 필요합니다.

사전 조건 및 제한 사항

사전 조건

- [vmware.com](https://www.vmware.com)에서 VMware 콘솔에 액세스할 수 있는 VMware 계정.
- HCX에는 다음과 같은 방화벽 포트가 필요합니다.

소스

대상

Port

HCX 관리자 및 어플라이언스 IP 온프레미스	AWS의 VMware Cloud HCX 관리자 및 어플라이언스 IP	UDP 500, UDP 4500, ICMP
HCX 관리자 및 어플라이언스 IP 온프레미스	connect.hcx.vmware.com hybrid dity-depot.vmware.com	TCP 443
HCX 관리자 및 어플라이언스 IP 온프레미스	HCX 클라우드 URL	TCP 443

온프레미스 네트워크에 내부 방화벽이 있는 경우 데이터 센터 내에 로컬로 포트를 몇 개 더 허용해야 합니다. HCX의 전체 포트 요구 사항 목록은 [VMware HCX 설명서](#)를 참조하세요.

- HCX를 구성하려면 도메인 이름 시스템(DNS) IP, vCenter FQDN(정규화된 도메인 이름), NTP 서버 FQDN, AWS Single Sign-On(SSO) 사용자 및 유사한 정보가 필요합니다. 이러한 세부 정보를 미리 수집하여 배포가 지연되지 않도록 하세요.

제한 사항

네트워크 확장 어플라이언스를 사용하여 온프레미스 환경과 VMware Cloud on AWS 간에 최대 8개의 네트워크를 확장할 수 있습니다. HCX 서비스 제한의 전체 목록은 [VMware HCX 설명서](#)를 참조하세요.

아키텍처

소스 기술 스택

- 온프레미스 VMware 워크로드

대상 기술 스택

- AWS의 VMware Cloud

도구

도구

- [AWS의 VMware Cloud](#)는 온프레미스 VMware vSphere 기반 환경을 AWS 클라우드로 마이그레이션하고 확장하는 데 도움이 되도록 AWS와 VMware가 공동으로 설계한 서비스입니다.

- [VMware 하이브리드 클라우드 확장\(HCX\)](#)은 기본 플랫폼을 변경하지 않고 온프레미스 VMware 환경에서 AWS의 VMware CloudS로 워크로드를 마이그레이션할 수 있는 VMware 유틸리티입니다.

에픽

HCX 배포

작업	설명	필요한 기술
AWS 기반 VMware 클라우드에서 HCX 서비스를 활성화합니다.	<ol style="list-style-type: none"> 1. AWS의 VMware Cloud 콘솔에 로그인합니다. 2. SDCC로 이동하여 세부 정보 보기를 선택합니다. 3. 추가 기능 탭을 선택합니다. 4. HCX 열기를 선택합니다. 5. HCX 배포를 선택하고 확인합니다. HCX 배포가 시작됩니다. 	클라우드 관리자, 시스템 관리자
HCX 활성화 키를 생성합니다.	<ol style="list-style-type: none"> 1. AWS의 VMware Cloud 콘솔의 경우. 2. SDCC로 이동하여 세부 정보 보기를 선택합니다. 3. 추가 기능 탭을 선택합니다. 4. HCX 열기를 선택한 다음 활성화 키를 선택합니다. 5. 활성화 키 생성을 선택하고 키를 복사합니다. 	클라우드 관리자, 시스템 관리자
클라우드 SDCC에 HCX용 방화벽 규칙을 추가합니다.	HCX Manager를 배포한 후에는 온프레미스 환경과 SDCC 간의 통신을 가능하게 하는 방화벽 규칙을 구성해야 합니다. 두 개의 방화벽 규칙을 만들어야 합니다. 하나는 인바운드 통	클라우드 관리자, 시스템 관리자

작업	설명	필요한 기술
	<p>신용이고 다른 하나는 아웃바운드 통신용입니다.</p> <ol style="list-style-type: none"> 1. AWS의 VMware Cloud 콘솔에서 SDDC를 선택하고 네트워킹 및 보안으로 이동합니다. 2. 게이트웨이 방화벽을 선택한 다음 관리 게이트웨이 탭을 선택합니다. 3. 규칙 추가를 선택하고 아웃바운드 규칙을 생성합니다. <ol style="list-style-type: none"> a. 규칙 이름을 입력합니다. b. 소스를 편집하고 HCX를 선택합니다. c. 대상을 편집하고 HCX에 액세스할 수 있는 온프레미스 IP 및 서브넷을 제공합니다. d. 서비스에서 아무거나를 선택합니다. e. 작업에서 허가를 선택합니다. f. 게시를 선택합니다. 4. 규칙 추가를 선택하고 인바운드 규칙을 생성합니다. <ol style="list-style-type: none"> a. 규칙 이름을 입력합니다. b. 원본을 편집하고 HCX에 액세스할 수 있는 온프레미스 IP 및 서브넷을 제공합니다. c. 대상을 편집하고 HCX를 선택합니다. 	

작업	설명	필요한 기술
	<ul style="list-style-type: none"> d. 서비스에서 SSH, HTTPS, TCP(9443) 및 ICMP를 선택합니다. e. 작업에서 허가를 선택합니다. f. 게시를 선택합니다. 	
<p>온프레미스에 HCX Manager를 설치합니다.</p>	<ol style="list-style-type: none"> 1. 클라우드 vCenter에 로그인하고 메뉴에서 HCX로 이동합니다. 2. HCX 대시보드에서 관리, 시스템 업데이트를 선택합니다. 3. VMware HCX 커넥터의 다운로드 링크를 요청하고 온프레미스 OVA 파일을 다운로드합니다. 4. 온프레미스 vCenter에 로그인하고 다운로드한 OVA 파일을 사용하여 OVF 템플릿을 배포합니다. 5. 템플릿을 배포하는 동안 메시지가 표시되면 정적 IP, NTP, DNS, DNS 검색 목록 및 기타 세부 정보를 제공합니다. 6. 모든 세부 정보를 검증하여 HCX Manager 배포를 완료합니다. 	<p>클라우드 관리자, 시스템 관리자</p>

작업	설명	필요한 기술
<p>온프레미스에서 HCX Manager를 구성합니다.</p>	<ol style="list-style-type: none"> 1. <a href="https://<HCX_Manager_IP>:9433">https://<HCX_Manager_IP>:9433 브라우저에서 HCX 관리자를 엽니다. 2. 배포 중에 제공된 사용자 이름과 비밀번호를 사용하여 로그인합니다. 3. 이전에 생성한 활성화 키를 입력하고 활성화를 선택하여 HCX 인스턴스를 활성화합니다. 4. 확인을 선택하여 다음 단계로 이동합니다. 5. 온프레미스 데이터 센터의 위치를 선택한 다음 계속을 선택합니다. 6. 시스템 이름에 호스트 이름을 입력한 다음 계속을 선택하여 활성화를 완료합니다. 7. vCenter 연결을 구성하는 데 필요한 정보를 입력합니다. 8. 정보를 입력하여 SSO/PSC 세부 정보를 구성합니다. 9. 변경 사항이 적용되도록 재시작을 선택합니다. 	<p>클라우드 관리자, 시스템 관리자</p>

작업	설명	필요한 기술
<p>사이트 페어링을 구성합니다.</p>	<p>클라우드와 온프레미스에서 HCX를 구성한 후 다음 단계에 따라 둘 사이의 사이트 페어링을 구성합니다.</p> <ol style="list-style-type: none"> 1. 온프레미스 vCenter에 로그인하고 HCX 대시보드로 이동합니다. 2. 왼쪽 탐색 창에서 사이트 페어링을 선택한 다음 원격 사이트에 연결을 선택합니다. 3. 원격 사이트에 연결 대화 상자에서 HCX 클라우드 URL 및 보안 인증 정보를 추가한 다음 연결을 선택합니다. <p>사이트 페어링이 완료되면 사이트 페어링 대시보드에 온프레미스 및 클라우드 SDDC가 연결된 것으로 표시됩니다.</p>	<p>클라우드 관리자, 시스템 관리자</p>

작업	설명	필요한 기술
<p>네트워크 프로파일을 생성합니다.</p>	<p>네트워크 프로파일은 네트워크의 Layer 3 구성 요소를 추상화한 것입니다. 이 프로파일은 컴퓨팅 프로파일을 생성하기 위한 전제 조건입니다.</p> <ol style="list-style-type: none"> 1. 클라우드 vCenter에 로그인하고 HCX 대시보드로 이동합니다. 2. 상호 연결을 선택하고 네트워크 프로파일 탭을 선택한 다음 네트워크 프로파일 생성을 선택합니다. 3. 네트워크 프로파일을 구성합니다. <ol style="list-style-type: none"> a. vCenter 서버를 선택합니다. b. 네트워크 탭을 선택합니다. c. 프로필의 이름을 추가합니다. d. IP 풀, 접두사 길이, 게이트웨이, DND 및 MTU를 입력합니다. e. 생성을 선택하세요. 4. 동일한 프로세스를 거쳐 온프레미스 네트워크 프로파일을 생성합니다. 	<p>클라우드 관리자, 시스템 관리자</p>

작업	설명	필요한 기술
<p>컴퓨팅 프로파일을 생성합니다.</p>	<p>컴퓨팅 프로파일은 HCX의 네트워크, 스토리지 및 컴퓨팅 세부 정보로 구성됩니다. HCX는 서비스 메시지를 생성하는 동안 HCX 어플라이언스를 생성할 때 이러한 설정을 사용합니다.</p> <ol style="list-style-type: none"> 1. 온프레미스 vCenter에 로그인하고 HCX 대시보드로 이동합니다. 2. 상호 연결을 선택하고 컴퓨팅 프로파일 탭을 선택한 다음 컴퓨팅 프로파일 생성을 선택합니다. 3. 컴퓨팅 프로파일의 이름을 지정합니다. 4. 활성화하려는 HCX 서비스를 선택한 다음 계속을 선택합니다. 5. 서비스 리소스를 선택합니다. 클러스터가 여러 개 있는 경우 HCX 서비스를 활성화하려는 각 클러스터를 선택한 다음 계속을 선택하세요. 6. HCX 어플라이언스를 배포하기 위한 컴퓨팅 및 스토리지 리소스를 선택한 다음 계속을 선택합니다. 7. vCenter 및 ESXi 호스트의 관리 인터페이스에 연결하는데 사용할 수 있는 관리 네트워크 프로파일을 선택한 다음 계속을 선택합니다. 	<p>클라우드 관리자, 시스템 관리자</p>

작업	설명	필요한 기술
	<p>8. 원격 사이트의 상호 연결 장치에 연결하는 데 사용할 수 있고 원격 사이트 장치가 로컬 상호 연결 장치에 연결하는 데 사용할 수 있는 업링크 네트워크 프로파일을 선택한 다음 계속을 선택합니다.</p> <p>9. vMotion 네트워크 프로파일을 선택한 다음 계속을 선택합니다.</p> <p>10.vSphere 복제 네트워크 프로파일을 선택한 다음 계속을 선택합니다.</p> <p>11.네트워크 확장에 적합한 분산 스위치를 선택한 다음 계속을 선택합니다.</p> <p>12.WAN 및 LAN 연결에서 열어야 하는 모든 포트를 검토한 다음 계속을 선택합니다.</p> <p>13.프로젝트를 생성하려면 마침을 선택하세요.</p> <p>14.동일한 단계를 거쳐 클라우드 사이트에서 컴퓨팅 프로필을 생성합니다.</p>	

작업	설명	필요한 기술
서비스 메시지를 생성합니다.	<p>서비스 메시지는 온프레미스 사이트와 클라우드 사이트 모두에 대한 HCX 서비스 구성을 제공합니다. 서비스 메시지를 생성하면 두 사이트 모두에 HCX 상호 연결 가상 어플라이언스 배포가 시작됩니다. 상호 연결 서비스는 소스 사이트에서 생성되어야 합니다.</p> <ol style="list-style-type: none"> 1. 온프레미스 vCenter에 로그인하고 HCX 대시보드로 이동합니다. 2. 상호 연결을 선택하고 서비스 메시 탭을 선택한 다음 서비스 메시 생성을 선택합니다. 3. 서비스 메시지를 생성할 소스 및 대상 사이트를 선택한 다음 계속을 선택합니다. 4. 이전에 생성한 소스 및 대상 사이트의 컴퓨팅 프로필을 선택한 다음 계속을 선택합니다. 5. 활성화하려는 HCX 서비스를 선택한 다음 계속을 선택합니다. 6. 소스 및 대상 사이트 모두에 대한 업링크 프로필을 선택한 다음 계속을 선택합니다. 7. 리소스와 네트워크를 검토한 다음 계속을 선택합니다. 	클라우드 관리자, 시스템 관리자

작업	설명	필요한 기술
	<p>8. 서비스 메시의 이름을 입력한 다음 마침을 선택합니다.</p> <p>서비스 메시 배포가 시작됩니다. 서비스 메시의 태스크 탭에서 진행 상황을 추적할 수 있습니다. 배포가 완료되면 서비스 메시에 대해 활성화한 모든 HCX 서비스의 상태가 표시됩니다.</p>	

HCX를 사용하여 네트워킹 확장

작업	설명	필요한 기술
네트워크 확장을 생성합니다.	<p>HCX 네트워크 확장 기능을 사용하여 클라우드 SDDC HCX 사이트에서 L2 네트워크 확장을 생성하고 원격 네트워크와 소스 네트워크를 연결할 수 있습니다.</p> <p>이렇게 하면 동일한 IP 주소를 유지하면서 온프레미스에서 AWS의 VMware Cloud로 서버를 마이그레이션할 수 있습니다.</p> <ol style="list-style-type: none"> 1. 온프레미스 vCenter에 로그인하고 HCX 대시보드로 이동합니다. 2. 서비스, 네트워크 확장을 선택합니다. 	클라우드 관리자, 시스템 관리자

작업	설명	필요한 기술
	<ol style="list-style-type: none"> 3. 네트워크 확장 또는 네트워크 확장 생성을 선택합니다. 4. 적절한 서비스 메시, 분산 포트 그룹 또는 NSX 논리적 스위치를 선택합니다. 5. 게이트웨이 IP 주소를 입력한 다음 제출을 선택합니다. <p>네트워크 확장이 완료되면 시스템은 확장 완료를 표시합니다.</p>	

HCX를 사용하여 복제 작업 구성

작업	설명	필요한 기술
복제를 구성합니다.	<p>HCX를 사용하여 VM을 복제하는 방법:</p> <ol style="list-style-type: none"> 1. 온프레미스 vCenter에 로그인하고 HCX 대시보드로 이동합니다. 2. 마이그레이션을 선택한 다음 마이그레이션 탭을 선택합니다. 3. 모빌리티 그룹 이름을 제공하고 마이그레이션할 VM을 선택한 다음 추가를 선택합니다. 4. 대상 컴퓨팅 컨테이너, 스토리지 폴더, 마이그레이션 유형(cold, bulk, RAV, 	클라우드 관리자, 시스템 관리자

작업	설명	필요한 기술
	<p>vMotion) 및 전환 일정을 선택합니다.</p> <p>5. 검증을 선택하고 검증이 완료될 때까지 기다린 다음 Go를 선택하여 복제를 시작합니다.</p>	

HCX 업그레이드

작업	설명	필요한 기술
<p>권장 사항 및 단계를 검토합니다.</p>	<p>대규모 마이그레이션 프로젝트는 6개월에서 8개월까지 지속될 수 있고 때로는 더 오래 지속될 수 있으며 VMware는 소프트웨어 수정, 보안 업데이트 및 버그 수정으로 구성된 HCX 업데이트를 정기적으로 게시합니다. 보안 취약성을 제거하고 새로운 기능을 활용하려면 HCX와 어플라이언스를 최신 상태로 유지하는 것이 좋습니다.</p> <p>참고: 현재 HCX 버전이 최신 릴리스 또는 이전 버전보다 세 가지 버전 이상 뒤처진 경우 HCX를 업그레이드할 수 없으며 재배포해야 합니다.</p> <p>HCX 업그레이드는 다음과 같은 세 단계로 구성됩니다.</p> <p>1. 온프레미스와 클라우드에서 HCX Manager를 백업합니다.</p>	<p>클라우드 관리자, 시스템 관리자</p>

작업	설명	필요한 기술
	<p>2. 온프레미스와 클라우드에서 HCX Manager를 업그레이드합니다.</p> <p>3. 온프레미스와 클라우드에서 서비스 메시 어플라이언스를 업그레이드합니다.</p> <p>다음 이야기에서는 이러한 단계를 자세히 설명합니다.</p>	

작업	설명	필요한 기술
<p>HCX 클라우드 매니저를 백업합니다.</p>	<p>AWS의 VMware Cloud용 HCX 클라우드 관리자는 VMware에서 관리하므로 스냅샷을 찍을 수 없습니다. HCX Cloud Manager를 백업하려면 업그레이드가 실패하거나 이전 단계로 롤백해야 하는 경우에 대비하여 HCX 콘솔에서 백업을 다운로드하고 이 백업을 사용하여 HCX 구성을 복원해야 합니다.</p> <ol style="list-style-type: none"> 1. <a href="https://<HCX_cloud_manager_ip_or_fqdn>:9433">https://<HCX_cloud_manager_ip_or_fqdn>:9433 에서 HCX 클라우드 관리자에 로그인합니다. 2. 관리, 문제 해결, 백업 및 복원으로 이동합니다. 3. 백업 섹션에서 생성을 선택하여 백업 파일을 생성합니다. 4. 다운로드를 선택하여 백업 파일을 저장합니다. <p>HCX-IX, HCX-NE 및 HCX-WO와 같은 HCX 서비스 어플라이언스에는 개별 백업이 필요하지 않습니다.</p>	<p>클라우드 관리자, 시스템 관리자</p>

작업	설명	필요한 기술
<p>온프레미스에서 HCX Manager를 백업합니다.</p>	<p>VM 스냅샷을 생성하거나 구성 파일을 백업하는 두 가지 방법으로 온프레미스에서 HCX Manager를 백업할 수 있습니다.</p> <p>VM 스냅샷을 찍는 방법:</p> <ol style="list-style-type: none"> 1. 온프레미스 vCenter에 로그인합니다. 2. VM 및 템플릿으로 이동하여 HCX 관리자 VM으로 이동합니다. 3. 작업, 스냅샷, 스냅샷 촬영을 선택합니다. <p>구성 파일을 백업하는 방법:</p> <ol style="list-style-type: none"> 1. <code>https://<HCX_cloud_manager_ip_or_fqdn>:9433</code> 에서 HCX 클라우드 관리자에 로그인합니다. 2. 관리, 문제 해결, 백업 및 복원으로 이동합니다. 3. 백업 섹션에서 생성을 선택하여 백업 파일을 생성합니다. 4. 다운로드를 선택하여 백업 파일을 저장합니다. <p>HCX-IX, HCX-NE 및 HCX-WO와 같은 HCX 서비스 어플라이</p>	<p>클라우드 관리자, 시스템 관리자</p>

작업	설명	필요한 기술
	언스에는 개별 백업이 필요하지 않습니다.	

작업	설명	필요한 기술
<p>온프레미스와 클라우드에서 HCX Manager를 업그레이드합니다.</p>	<p>먼저 온프레미스에서 HCX Manager를 업그레이드한 다음 HCX 클라우드 관리자를 업그레이드해야 합니다.</p> <p>온프레미스에서 HCX Manager를 업그레이드하는 방법:</p> <ol style="list-style-type: none"> 1. vCenter에 로그인하고 HCX 대시보드로 이동합니다. 2. 시스템, 관리를 선택합니다. 3. 관리 페이지에서 시스템 업데이트 탭을 선택합니다. 사용 가능한 서비스 업데이트 버전 옆에는 보류 중인 업데이트가 표시됩니다. 4. 서비스 업데이트 선택, 다운로드를 선택하여 나중에 업그레이드할 수 있도록 업데이트를 다운로드하거나 다운로드 및 업그레이드를 선택하여 업데이트를 즉시 다운로드하고 배포할 수 있습니다. 다운로드를 선택한 경우 업그레이드를 선택하고 준비가 되면 업그레이드를 시작할지 확인합니다. 5. 업그레이드 완료 시: <ul style="list-style-type: none"> • HCX 관리자 관리 페이지에서 최신 HCX 버전이 표시되는지 확인합니다. • HCX 대시보드에서 사이트 페어링이 Up 상태인지 확인합니다. 	<p>클라우드 관리자, 시스템 관리자</p>

작업	설명	필요한 기술
	<ul style="list-style-type: none">인프라, 서비스 메시지를 선택하고 모든 HCX 서비스가 정상인지 확인합니다. <p>동일한 단계를 거쳐 HCX Cloud Manager를 업그레이드합니다.</p>	

작업	설명	필요한 기술
<p>서비스 메시 어플라이언스를 업그레이드합니다.</p>	<p>서비스 메시는 소스 사이트의 HCX Manager와 독립적으로 업데이트됩니다. 대상 사이트의 서비스 메시 어플라이언스는 자동으로 업데이트됩니다.</p> <p>소스 사이트에서 서비스 메시 어플라이언스를 업그레이드하는 방법:</p> <ol style="list-style-type: none"> 1. vCenter에 로그인하고 HCX 대시보드로 이동합니다. 2. 인프라를 선택한 다음 서비스 메시 탭을 선택합니다. 3. “서비스 메시 어플라이언스의 새 버전 사용 가능” 배너가 표시되는 경우입니다. 최신 버전으로 업그레이드하려면 어플라이언스 업데이트를 클릭하고 어플라이언스 업데이트를 선택합니다. 4. 기기를 표시하는 대화 상자에서 하나 이상의 기기를 선택한 다음 확인을 선택하여 업그레이드 프로세스를 시작합니다. (모든 서비스 메시 어플라이언스를 업데이트하는 것이 좋습니다.) 5. 업그레이드를 모니터링하려면 각 서비스 메시의 작업 보기를 선택하세요. 6. 업그레이드가 완료되면 각 어플라이언스 및 서비스에 	<p>클라우드 관리자, 시스템 관리자</p>

작업	설명	필요한 기술
	<p>성공적인 완료를 확인하는 배너가 나타납니다.</p> <p>7. 업그레이드 후 터널 상태를 검증합니다.</p> <ul style="list-style-type: none"> • 인프라, 서비스 메시, 어플라이언스 보기를 선택합니다. • 터널 상태 열이 Up으로 표시되고 화면에 사용 가능한 다른 어플라이언스 버전이 표시되지 않아야 합니다. 	

HCX 네트워크 확장 제거

작업	설명	필요한 기술
<p>확장되지 않은 네트워크.</p>	<p>이전 단계에서는 HCX 네트워크 확장 기능을 사용하여 L2 네트워크 확장을 생성하고 온프레미스에서 AWS의 VMware Cloud로 마이그레이션하는 동안 기존 IP를 유지하는 방법을 설명했습니다. 특정 VLAN의 모든 VM을 AWS의 VMware Cloud로 이동시킨 후에는 온프레미스 사이트와 클라우드 SDDC 간의 네트워크 확장을 취소하고 SDDC에서 네트워크를 라우팅할 수 있게 만들어야 합니다.</p> <p>지연 시간을 피하려면 모든 VM이 온프레미스에서 AWS의</p>	<p>클라우드 관리자, 시스템 관리자</p>

작업	설명	필요한 기술
	<p>VMware Cloud로 마이그레이션되는 즉시 확장 네트워크를 제거하는 것이 좋습니다.</p> <ol style="list-style-type: none"> 1. 온프레미스 vCenter에 로그인하고 HCX 대시보드로 이동합니다. 2. HCX 대시보드에서 서비스, 네트워크 확장을 선택합니다. 3. 확장 취소하려는 네트워크를 선택한 다음 네트워크 확장 취소를 선택합니다. 4. 확장을 취소한 후 클라우드 네트워크를 클라우드 에지 게이트웨이에 연결을 선택합니다. 이렇게 하면 클라우드 측의 네트워크가 활성화됩니다. 	

작업	설명	필요한 기술
클라우드 SDDC에서 이동한 네트워크를 라우팅합니다.	<ol style="list-style-type: none"> 1. VMC 포털에 로그인합니다. 2. SDCC로 이동한 다음 세부 정보 보기를 선택합니다. 3. 네트워킹 및 보안 탭을 선택합니다. 4. 네트워킹 및 보안 페이지의 경우: <ul style="list-style-type: none"> • 네트워킹, 세그먼트를 선택하고 최근에 확장되지 않은 서브넷이 라우팅 가능으로 표시되는지 확인합니다. • 인벤토리, 그룹을 선택하고 해당 서브넷을 그룹에 추가합니다. • 보안, 분산 방화벽을 선택하고 그룹이 의도한 방화벽 규칙에 속하는지 확인합니다. 	클라우드 관리자, 시스템 관리자

WSL 제거

작업	설명	필요한 기술
전제 조건을 확인합니다.	데이터 센터를 종료하는 경우 마이그레이션 프로젝트가 끝날 때 HCX를 제거하고 해당 구성 요소를 제거하는 것이 좋습니다. 그러나 여전히 온프레미스 설치 공간을 유지하고 있다면 HCX를 계속 운영하는 것이 좋습니다.	클라우드 관리자, 시스템 관리자

작업	설명	필요한 기술
	<p>HCX를 제거하기 전에 다음 사항을 확인하세요.</p> <ul style="list-style-type: none">• 활성 마이그레이션이 없습니다.• 모든 네트워크 확장이 제거되었습니다.	

작업	설명	필요한 기술
온프레미스에서 HCX를 제거합니다.	<ol style="list-style-type: none"> 1. 온프레미스 vCenter에 로그인하고 HCX 콘솔로 이동합니다. 2. 서비스, 마이그레이션을 선택하고 활성 마이그레이션이 없는지 확인합니다. 3. 서비스, 네트워크 확장을 선택하고 확장된 네트워크가 없는지 확인합니다. 4. 인프라, 사이트 페어링, 서비스 메시지를 선택합니다. 5. 서비스 메시지를 식별한 다음 삭제를 선택합니다. 6. 확인 프롬프트에서 다시 삭제를 선택합니다. 서비스 메시지 화면에 “서비스 메시지 제거” 배너가 나타납니다. 7. 가지고 있는 다른 서비스 메시지에 대해 5-6단계를 반복합니다. 8. 사이트 페어링을 제거하려면 인프라, 사이트 페어링을 선택한 다음 페어링된 모든 사이트의 연결을 해제하세요. 9. HCX 관리자 어플라이언스 제거: <ol style="list-style-type: none"> a. 온프레미스 vCenter에 로그인하고 HCX Manager 어플라이언스로 이동합니다. 	클라우드 관리자, 시스템 관리자

작업	설명	필요한 기술
	b. 작업, 전원, 전원 끄기를 선택합니다. c. 작업, 디스크에서 삭제를 선택합니다.	
온프레미스 vCenter 서버에서 HCX 플러그인 등록을 취소합니다.	<ol style="list-style-type: none"> 1. <code>https://<vc_fqdn>/mob</code> 에서 vCenter MOB UI 에 로그인합니다. 2. 속성 섹션의 값 열에 있는 내용을 선택합니다. 3. 콘텐츠 페이지에서 등록된 모든 플러그인을 보도록 선택합니다. Extension Manager 4. <code>com.vmware.hybridty ,com.vmware.e.hcsp.alarm , com.vmware.vca.marketing.ngc.ui</code> 로 시작하는 확장 프로그램을 기록합니다. 5. 다음 확장 프로그램을 제거합니다. <ul style="list-style-type: none"> • 방법 섹션에서 선택합니다 UnregisterExtension. • 4단계에서 기록해 둔 확장 키를 입력한 다음 간접 호출 메서드를 선택하여 확장을 제거합니다. <p>모든 확장이 제거되면 vSphere Web Client에서 HCX 플러그인이 사라집니다.</p>	클라우드 관리자, 시스템 관리자

작업	설명	필요한 기술
클라우드에서 HCX를 제거합니다.	<p>클라우드에서 HCX 서비스 메시 및 사이트 페어링을 제거하려면 앞서 온프레미스 HCX 제거에서 설명한 단계를 반복하세요. AWS의 VMware Cloud에서는 VMware에서 HCX Manager를 관리합니다. vCenter에서 삭제할 수는 없지만 VMC 관리 인터페이스에서는 배포를 취소할 수 있습니다.</p> <p>HCX Manager의 배포를 취소하는 방법:</p> <ol style="list-style-type: none"> 1. VMC 관리 인터페이스에 로그인합니다. 2. 조직 및 SDDC를 선택합니다. 3. HCX가 배포된 모든 SDDC를 표시하려면 추가 기능을 선택합니다. 4. HCX 배포 취소를 선택합니다. 	클라우드 관리자, 시스템 관리자

문제 해결

문제	Solution
HCX 대량 마이그레이션을 구성할 때는 마이그레이션할 서버를 선택할 수 없습니다.	<p>원인: 이러한 서버의 마이그레이션이 취소되었지만 정리 중에 HCX 데이터베이스가 업데이트되지 않았습니다. HCX는 데이터베이스 마이그레이션이 아직 진행 중인 것으로 간주하여 상태를 “전환 진행 중”으로 고정했습니다.</p>

문제	Solution
	<p>해결 방법: VMware 지원 팀에 문의하여 HCX 데이터베이스를 정리하세요.</p>
<p>전환은 실패하지만 강제 전원 끄기 옵션과 함께 사용할 수 있습니다.</p>	<p>원인: VMware Tools 버전이 HCX 대량 마이그레이션의 사전 조건을 충족하지 않아 HCX가 소스 VM을 종료하지 못했습니다.</p> <p>해결 방법: VMware 도구를 마이그레이션 유형에 맞는 권장 버전으로 업데이트하세요.</p>
<p>마이그레이션이 진행되는 동안 HCX 사이트 페어링 어플라이언스 업그레이드가 실패하고 “진행 중인 대량 마이그레이션에는 작업이 허용되지 않습니다”라는 오류가 발생합니다.</p>	<p>원인: 전환 후 HCX 데이터베이스가 업데이트되지 않았습니다.</p> <p>해결 방법: 진행 중인 마이그레이션이 없는지 확인하세요. 사이트 페어링 어플라이언스를 업그레이드할 때 강제 업그레이드를 선택합니다.</p>
<p>전환이 실패하고 “리소스 가용성 부족” 오류가 발생합니다.</p>	<p>원인: 호스트 VM의 스토리지가 부족합니다.</p> <p>해결 방법: 마이그레이션하기 전에 스토리지와 컴퓨팅 리소스를 확인하세요.</p>

관련 리소스

참조

- [AWS의 VMware Cloud 특성](#)
- [AWS의 VMware Cloud 개요 및 운영 모델\(AWS 권장 가이드\)](#)
- [VMware HCX를 사용하여 Migrate VMware SDDC에서 AWS의 VMware Cloud로 마이그레이션\(AWS 권장 가이드\)](#)
- [AWS의 VMware Cloud에서의 VMware HCX\(VMware 설명서\)](#)
- [HCX HCX 릴리스 노트\(VMware 설명서\)](#)
- [AWS 기반 SDDC 배포 및 모범 사례 가이드\(AWS 백서\)](#)

도구

- [PowerCLI를 사용한 AWS Automation의 VMware Cloud](#)(VMware Cloud Tech Zone)

파트너

- [AWS의 VMware Cloud 파트너 이니셔티브](#)

비디오

- AWS 기반 [VMware 클라우드](#) YouTube (동영상)

pg_transport를 사용하여 두 Amazon RDS DB 인스턴스 간에 PostgreSQL 데이터베이스 전송

작성자: Raunak Rishabh(AWS)와 Jitender Kumar(AWS)

환경: PoC 또는 파일럿	소스: 데이터베이스: 관계형	대상: Amazon RDS for PostgreSQL
R 유형: 재배치	워크로드: 오픈 소스	기술: 마이그레이션, 데이터베이스
AWS 서비스: Amazon RDS		

요약

이 패턴은 pg_transport 확장 프로그램을 사용하여 PostgreSQL DB 인스턴스용 두 Amazon Relational Database Service(Amazon RDS) 간에 매우 큰 데이터베이스를 마이그레이션하는 단계를 설명합니다. 이 확장 프로그램은 각 데이터베이스를 이동하기 위한 물리적 전송 메커니즘을 제공합니다. 최소한의 처리로 데이터베이스 파일을 스트리밍함으로써 가동 중단을 최소화하면서 DB 인스턴스 간에 대규모 데이터베이스를 매우 빠르게 마이그레이션할 수 있는 방법을 제공합니다. 이 확장 프로그램은 대상 DB 인스턴스가 소스 DB 인스턴스에서 데이터베이스를 가져오는 풀링 모델을 사용합니다.

사전 조건 및 제한 사항

사전 조건

- 두 DB 인스턴스 모두 동일한 메이저 버전의 PostgreSQL을 실행해야 합니다.
- 데이터베이스는 대상에 없어야 합니다. 그렇지 않으면 전송이 실패합니다.
- 소스 데이터베이스에서 pg_transport이외의 확장 프로그램을 활성화하지 않아야 합니다.
- 모든 소스 데이터베이스 객체는 기본 pg_default 테이블스페이스에 있어야 합니다.
- 소스 DB 인스턴스의 보안 그룹은 대상 DB 인스턴스에서 들어오는 트래픽을 허용해야 합니다.
- psql과 [같은](#) PostgreSQL 클라이언트를 [PgAdmin](#) 설치하거나 Amazon RDS PostgreSQL DB 인스턴스와 함께 작동하도록 설치하십시오. 클라이언트는 로컬 시스템에 설치하거나 Amazon Elastic Compute Cloud (Amazon EC2) 인스턴스를 사용할 수 있습니다. 이 패턴에서는 EC2 인스턴스에서 psql을 사용합니다.

제한 사항

- Amazon RDS for PostgreSQL의 서로 다른 주요 버전 간에는 데이터베이스를 전송할 수 없습니다.
- 소스 데이터베이스의 액세스 권한 및 소유권은 대상 데이터베이스로 전송되지 않습니다.
- 읽기 전용 복제본 또는 읽기 전용 복제본 또는 읽기 전용 복제본의 상위 인스턴스에서 전송 가능 데이터베이스를 사용할 수 없습니다.
- 이 방법으로 전송하려는 데이터베이스 테이블에서는 reg 데이터 유형을 사용할 수 없습니다.
- DB 인스턴스에서는 동시에 최대 32개의 전송(가져오기 및 내보내기 포함)을 실행할 수 있습니다.
- 테이블의 이름을 바꾸거나 테이블을 포함/제외할 수 없습니다. 모든 것이 그대로 마이그레이션됩니다.

주의

- 확장 프로그램을 제거하기 전에 백업을 생성합니다. 왜냐하면 확장 프로그램을 제거하면 데이터베이스 작동에 중요한 종속 객체 및 일부 데이터도 제거되기 때문입니다.
- pg_transport 작업자 수와 work_mem 값을 결정할 때는 소스 인스턴스의 다른 데이터베이스에서 실행되고 있는 인스턴스 클래스와 프로세스를 고려합니다.
- 전송이 시작되면 소스 데이터베이스의 모든 연결이 종료되고 데이터베이스가 읽기 전용 모드로 전환됩니다.

참고: 전송이 한 데이터베이스에서 실행되고 있는 경우 동일한 서버의 다른 데이터베이스에는 영향을 주지 않습니다.

제품 버전

- Amazon RDS for PostgreSQL 10.10 이상 및 Amazon RDS for PostgreSQL 11.5 이상입니다. 최신 버전 정보는 Amazon RDS 설명서의 [DB 인스턴스 간 PostgreSQL 데이터베이스 전송](#)을 참조하세요.

아키텍처

도구

- pg_transport은 각 데이터베이스를 이동하기 위한 물리적 전송 메커니즘을 제공합니다. 물리적 전송은 최소한의 처리로 데이터베이스 파일을 스트리밍함으로써 기존의 덤프 및 로드 프로세스보다 훨씬

신 빠르게 데이터를 이동하고 가동 중단을 최소화합니다. PostgreSQL 전송 가능 데이터베이스는 대상 DB 인스턴스가 소스 DB 인스턴스에서 데이터베이스를 가져오는 풀링 모델을 사용합니다. 이 패턴에 설명된 대로 소스 및 대상 환경을 준비할 때 이 확장 프로그램을 DB 인스턴스에 설치합니다.

- [psql](#)을 사용하면 PostgreSQL DB 인스턴스에 연결하여 작업할 수 있습니다. [시스템에 psql을 설치하려면 PostgreSQL 다운로드](#) 페이지를 참조하세요.

에픽

타겟 파라미터 그룹 생성

작업	설명	필요한 기술
대상 시스템에 대한 파라미터 그룹을 생성합니다.	대상 파라미터 그룹으로 식별되는 그룹 이름을 지정합니다. 예를 들면, <code>pgtarget-param-group</code> 입니다. 자세한 지침은 Amazon RDS 설명서 를 참조하세요.	DBA
파라미터 그룹의 파라미터를 수정합니다.	다음 파라미터를 설정합니다. <ol style="list-style-type: none"> 1. <code>pg_transport</code> 을 <code>shared_preload_libraries</code> 파라미터에 추가합니다. <pre>shared_preload_libraries = pg_stat_statements, pg_transport</pre> 2. <code>pg_transport.num_workers</code> 파라미터를 설정합니다. 함께 전송을 실행하려는 작업자 수를 선택합니다. 설정한 값에 따라 소스에서 생성될 <code>transport</code> 	DBA

작업	설명	필요한 기술
	<p><code>.send_file</code> 작업자 수가 결정됩니다.</p> <p>3. <code>max_worker_processes</code> 의 값을 <code>pg_transport.num_workers</code> 값의 3배 이상으로 증가합니다. 예를 들어 <code>pg_transport.num_workers</code> 의 값을 4로 설정하는 경우 <code>max_worker_processes</code> 값은 13 이상이어야 합니다. 실패할 경우 <code>pg_transport</code>는 최소값을 권장합니다.</p> <p>4. <code>pg_transport.timing</code> 을 1로 설정합니다. 이 설정을 사용하면 전송 중 타이밍 정보를 보고할 수 있습니다.</p> <p>5. <code>pg_transport.work_mem</code> 파라미터를 설정합니다. 이 파라미터는 각 작업자에게 할당할 최대 메모리를 지정합니다. 기본값은 128MB입니다.</p> <p>이 파라미터에 대한 자세한 내용은 Amazon RDS 설명서를 참조하세요.</p>	

소스 파라미터 그룹 생성

작업	설명	필요한 기술
소스 시스템에 대한 파라미터 그룹을 생성합니다.	소스 파라미터 그룹으로 식별되는 그룹 이름을 지정합니다. 예를 들면, <code>pgsource-param-group</code> 입니다. 자세한 지침은 Amazon RDS 설명서 를 참조하세요.	DBA
파라미터 그룹의 파라미터를 수정합니다.	다음 파라미터를 설정합니다. <ol style="list-style-type: none"> <code>pg_transport</code> 을 <code>shared_preload_libraries</code> 파라미터에 추가합니다. <div data-bbox="630 947 1029 1146" style="border: 1px solid #ccc; border-radius: 10px; padding: 10px; margin: 10px 0;"> <pre>shared_preload_libraries = pg_stat_statements, pg_transport</pre> </div> <code>pg_transport.num_workers</code> 파라미터를 설정합니다. 대상에 정의된 이 파라미터의 값에 따라 사용할 <code>transport.send_file</code> 작업자 수가 결정됩니다. 이 인스턴스에서 가져오기를 실행하는 경우 이 값을 증가하되 이미 실행 중인 작업자 수를 고려해야 합니다. 대상에서 <code>max_worker_processes</code> 의 값은 <code>pg_transport.num_workers</code> 값의 3배 이상으로 증가시킵니다. 예를 들 	DBA

작업	설명	필요한 기술
	<p>어 대상에서 <code>pg_transport.num_workers</code> 의 값을 4로 설정하는 경우 소스의 <code>max_workers_processes</code> 값은 13 이상이어야 합니다. 실패할 경우 <code>pg_transport</code>는 최소값을 권장합니다.</p> <p>4. <code>pg_transport.work_mem</code> 파라미터를 설정합니다. 이 파라미터는 각 작업자에게 할당할 최대 메모리를 지정합니다. 기본값은 128MB입니다.</p> <p>이 파라미터에 대한 자세한 내용은 Amazon RDS 설명서를 참조하세요.</p>	

대상 환경 준비

작업	설명	필요한 기술
<p>소스 데이터베이스를 전송할 신규 Amazon RDS for PostgreSQL DB 인스턴스를 생성합니다.</p>	<p>비즈니스 요구 사항에 따라 인스턴스 클래스와 PostgreSQL 버전을 결정합니다.</p>	<p>데이터베이스 관리자, 시스템 관리자, 데이터베이스 아키텍트</p>
<p>EC2 인스턴스로부터의 DB 인스턴스 포트 연결을 허용하도록 대상의 보안 그룹을 수정합니다.</p>	<p>기본적으로 PostgreSQL 인스턴스의 포트는 5432입니다. 다른 포트를 사용하는 경우 EC2 인스턴스에 대해 해당 포트에 대한 연결이 열려 있어야 합니다.</p>	<p>DBA, 시스템 관리자</p>

작업	설명	필요한 기술
인스턴스를 수정하고 새로운 타겟 파라미터 그룹을 할당합니다.	예를 들면 <code>pgtarget-param-group</code> 입니다.	DBA
대상 Amazon RDS DB 인스턴스를 다시 시작합니다.	<code>shared_preload_libraries</code> 과 <code>max_worker_processes</code> 파라미터는 정적 파라미터이므로 인스턴스를 재부팅해야 합니다.	DBA, 시스템 관리자
<code>psql</code> 을 사용하여 EC2 인스턴스에서 데이터베이스에 연결합니다.	다음 명령을 사용합니다. <pre>psql -h <rds_end_point> -p PORT -U username -d database -W</pre>	DBA
<code>pg_transport</code> 확장 프로그램을 생성합니다.	<code>rds_superuser</code> 역할이 있는 사용자로서 다음 쿼리를 실행합니다. <pre>create extension pg_transport;</pre>	DBA

소스 환경 준비

작업	설명	필요한 기술
Amazon EC2 인스턴스 및 대상 DB 인스턴스로부터의 DB 인스턴스 포트 연결을 허용하도록 소스의 보안 그룹을 수정	기본적으로 PostgreSQL 인스턴스의 포트는 5432입니다. 다른 포트를 사용하는 경우 EC2 인스턴스에 대해 해당 포트에 대한 연결이 열려 있어야 합니다.	DBA, 시스템 관리자

작업	설명	필요한 기술
인스턴스를 수정하고 새로운 소스 파라미터 그룹을 할당합니다.	예를 들면 <code>pgsource-param-group</code> 입니다.	DBA
소스 Amazon RDS DB 인스턴스를 다시 시작합니다.	<code>shared_preload_libraries</code> 과 <code>max_worker_processes</code> 파라미터는 정적 파라미터이므로 인스턴스를 재부팅해야 합니다.	DBA
<code>psql</code> 을 사용하여 EC2 인스턴스에서 데이터베이스에 연결합니다.	다음 명령을 사용합니다. <pre>psql -h <rds_end_point> -p PORT -U username -d database -W</pre>	DBA
<code>pg_transport</code> 확장 프로그램을 만들고 데이터베이스에서 전송할 다른 모든 확장 프로그램을 제거합니다.	소스 데이터베이스에 <code>pg_transport</code> 이외의 확장 프로그램이 설치되어 있는 경우 전송이 실패합니다. 이 명령은 해당 <code>rds_superuser</code> 역할을 가진 사용자가 실행해야 합니다.	DBA

전송 수행

작업	설명	필요한 기술
모의 실행을 수행합니다.	<code>transport.import_from_server</code> 함수를 사용하여 먼저 모의 실행을 다음과 같이 실시합니다. <pre>SELECT transport .import_from_serve</pre>	DBA

작업	설명	필요한 기술
	<pre data-bbox="609 212 1015 583">r('source-db-instan- ce-endpoint', source- db-instance-port, 'source-db-instance- user', 'source-user- password', 'source- database-name', 'destination-user- password', 'true');</pre> <p data-bbox="592 621 1003 751">이 함수의 마지막 파라미터 (true로 설정)는 모의 실행을 정의합니다.</p> <p data-bbox="592 793 1019 974">이 함수는 주요 전송을 실행할 때 나타날 수 있는 모든 오류를 표시합니다. 주요 전송을 실행하기 전에 오류를 해결합니다.</p>	

작업	설명	필요한 기술
<p>모의 실습이 성공하면 데이터베이스 전송을 시작합니다.</p>	<p><code>transport.import_from_server</code> 함수를 실행하여 전송을 실시합니다. 소스에 연결하고 데이터를 가져옵니다.</p> <pre data-bbox="597 491 1026 968">SELECT transport.import_from_server('source-db-instance-endpoint', source-db-instance-port, 'source-db-instance-user', 'source-user-password', 'source-database-name', 'destination-user-password', false);</pre> <p>이 함수의 마지막 파라미터 (<code>false</code>로 설정)는 모의 실습이 아님을 나타냅니다.</p>	DBA
<p>전송 후 단계를 실시합니다.</p>	<p>데이터베이스 전송이 완료된 후:</p> <ul data-bbox="597 1310 1026 1801" style="list-style-type: none"> • 대상 환경에서 데이터를 검증합니다. • 모든 역할과 권한을 대상에 추가합니다. • 필요한 경우 대상 및 소스에서 필요한 모든 확장 프로그램을 활성화합니다. • <code>max_worker_processes</code> 파라미터의 값을 되돌립니다. 	DBA

관련 리소스

- [Amazon RDS 설명서](#)
- [pg_transport 설명서](#)
- [RDS PostgreSQL 전송 가능한 데이터베이스를 사용한 데이터베이스 마이그레이션\(블로그 게시물\)](#)
- [PostgreSQL 다운로드](#)
- [psql 유틸리티](#)
- [DB 파라미터 그룹 생성](#)
- [파라미터 그룹의 파라미터를 수정하려면](#)
- [PostgreSQL 다운로드](#)

리플랫폼

주제

- [Oracle Database와 Aurora PostgreSQL-Compatible 간의 링크 구성](#)
- [AWS DMS를 사용하여 Microsoft SQL Server 데이터베이스를 Amazon S3로 내보내기](#)
- [AWS 개발자 도구를 SageMaker 사용하여 ML, 빌드, 교육 및 워크로드를 Amazon으로 마이그레이션, 배포](#)
- [OpenText TeamSite 워크로드를 AWS 클라우드로 마이그레이션](#)
- [AWS에서 PostgreSQL의 개별 행으로 Oracle CLOB 값을 마이그레이션](#)
- [데이터베이스 링크를 통한 직접 Oracle 데이터 펌프 가져오기를 사용하여 온프레미스 Oracle 데이터베이스를 Amazon RDS for Oracle로 마이그레이션](#)
- [Oracle E-Business Suite를 Amazon RDS Custom으로 마이그레이션](#)
- [PeopleSoft 오라클을 Amazon RDS 커스텀으로 마이그레이션](#)
- [Oracle ROWID 기능을 AWS 기반 PostgreSQL로 마이그레이션](#)
- [Oracle Database 오류 코드를 Amazon Aurora PostgreSQL Compatible 데이터베이스로 마이그레이션](#)
- [Redis 워크로드를 AWS의 Redis Enterprise Cloud로 마이그레이션](#)
- [AWS SCT 및 AWS DMS를 사용하여 SAP ASE에 있는 Amazon EC2를 Amazon Aurora PostgreSQL-Compatible로 마이그레이션하기](#)
- [ACM을 사용하여 Windows SSL 인증서를 Application Load Balancer로 마이그레이션](#)
- [메시지 대기열을 Microsoft Azure 서비스 버스에서 Amazon SQS로 마이그레이션](#)
- [오라클 데이터 펌프와 AWS DMS를 사용하여 오라클 JD 에드워즈 EnterpriseOne 데이터베이스를 AWS로 마이그레이션하십시오.](#)
- [AWS DMS를 사용하여 오라클 PeopleSoft 데이터베이스를 AWS로 마이그레이션합니다.](#)
- [온프레미스 MySQL 데이터베이스를 Amazon RDS for MySQL로 마이그레이션](#)
- [온프레미스 Microsoft SQL Server 데이터베이스를 Amazon RDS for SQL Server로 마이그레이션](#)
- [Rclone를 사용하여 Microsoft Azure Blob에서 Amazon S3로 데이터 마이그레이션하기](#)
- [카우치베이스 서버에서 AWS의 카우치베이스 카펠라로 마이그레이션](#)
- [IBM WebSphere 애플리케이션 서버에서 Amazon EC2의 아파치 톰캣으로 마이그레이션](#)
- [Auto Scaling을 사용하여 IBM WebSphere 애플리케이션 서버에서 Amazon EC2의 아파치 톰캣으로 마이그레이션하세요](#)
- [Microsoft Azure 앱 서비스의 .NET 애플리케이션을 AWS Elastic Beanstalk로 마이그레이션합니다.](#)

- [자체 호스팅된 MongoDB 환경을 AWS 클라우드의 MongoDB Atlas로 마이그레이션](#)
- [오라클에서 Amazon WebLogic ECS의 아파치 톰캣 \(TomEE\) 으로 마이그레이션](#)
- [AWS DMS를 사용하여 Amazon EC2에서 Amazon RDS for Oracle로 Oracle 데이터베이스 마이그레이션](#)
- [Logstash를 사용하여 온프레미스 Oracle 데이터베이스를 아마존 OpenSearch 서비스로 마이그레이션합니다.](#)
- [온프레미스 Oracle 데이터베이스를 Amazon RDS for Oracle로 마이그레이션](#)
- [Oracle Data Pump를 사용하여 온프레미스 Oracle 데이터베이스를 Amazon RDS for Oracle로 마이그레이션](#)
- [pglogical을 사용하여 Amazon EC2의 PostgreSQL에서 Amazon RDS for PostgreSQL로 마이그레이션합니다.](#)
- [온프레미스 PostgreSQL 데이터베이스를 Aurora PostgreSQL로 마이그레이션하기](#)
- [Linux가 실행되는 Amazon EC2의 Microsoft SQL Server로 온프레미스 Microsoft SQL Server 데이터베이스의 마이그레이션](#)
- [연결된 서버를 사용하여 온프레미스 Microsoft SQL Server 데이터베이스를 Amazon RDS for SQL Server로 마이그레이션](#)
- [기본 백업 및 복원 수단을 사용하여 온프레미스 Microsoft SQL Server 데이터베이스를 Amazon RDS for SQL Server로 마이그레이션](#)
- [AWS DMS와 AWS SCT를 사용하여 Microsoft SQL Server 데이터베이스를 Aurora MySQL로 마이그레이션](#)
- [온프레미스 MariaDB 데이터베이스를 기본 도구를 사용하여 Amazon RDS for MariaDB로 마이그레이션](#)
- [온프레미스 MySQL 데이터베이스를 Aurora MySQL로 마이그레이션](#)
- [페르코나, Amazon EFS 및 Amazon S3를 사용하여 온프레미스 MySQL 데이터베이스를 Aurora MySQL로 XtraBackup 마이그레이션합니다.](#)
- [AWS App2Container를 사용하여 온프레미스 Java 애플리케이션을 AWS로 마이그레이션](#)
- [AWS 대규모 마이그레이션에서 공유 파일 시스템 마이그레이션](#)
- [Oracle 플랫폼 파일 어댑터를 사용하여 GoldenGate Oracle 데이터베이스를 Amazon RDS for Oracle로 마이그레이션합니다.](#)
- [Microsoft SQL Server에서 Amazon Aurora PostgreSQL-Compatible Edition으로 데이터베이스 마이그레이션을 지원하도록 Python 및 Perl 애플리케이션 변경](#)

Oracle Database와 Aurora PostgreSQL-Compatible 간의 링크 구성

작성자: Jeevan Shetty(AWS), Bhanu Ganesh Gudivada(AWS), Sushant Deshmukh(AWS), Uttiya Gupta(AWS), Vikas Gupta(AWS)

환경: PoC 또는 파일럿	소스: Oracle Database	대상: Aurora PostgreSQL-Compatible
R 타입: 리플랫폼	워크로드: Oracle, 오픈 소스	기술: 마이그레이션, 데이터베이스

AWS 서비스: Amazon Aurora;
Amazon EC2 Auto Scaling;
Amazon Route 53

요약

Amazon Web Services(AWS) 클라우드로 마이그레이션하는 과정에서 클라우드 네이티브 데이터베이스를 사용하도록 애플리케이션을 현대화할 수 있습니다. Oracle Database에서 Amazon Aurora PostgreSQL 호환 버전으로 마이그레이션하는 것은 현대화를 향한 그러한 단계 중 하나입니다. 마이그레이션의 일환으로 네이티브 Oracle 데이터베이스 링크도 변환이 필요합니다.

데이터베이스 링크를 사용하면 한 데이터베이스가 다른 데이터베이스의 객체에 액세스할 수 있습니다. Oracle Database에서 Aurora PostgreSQL-Compatible로 마이그레이션한 후에는 Oracle Database 서버에서 다른 Oracle Database 서버로의 데이터베이스 링크를 PostgreSQL에서 Oracle Database로 가는 링크로 변환해야 합니다.

이 패턴은 Oracle Database 서버에서 Aurora PostgreSQL-Compatible 데이터베이스로 데이터베이스 링크를 설정하는 방법을 보여줍니다. 데이터베이스 링크는 단방향이므로 PostgreSQL 데이터베이스에서 Oracle Database로 데이터베이스 링크를 변환하는 패턴도 포함됩니다.

Oracle Database에서 Aurora PostgreSQL-Compatible 데이터베이스로 마이그레이션 및 변환한 후 데이터베이스 간 데이터베이스 링크를 설정하려면 다음 단계를 수행해야 합니다.

- Oracle Database를 소스로 사용하고 Aurora PostgreSQL-Compatible을 대상으로 하는 데이터베이스 링크를 설정하려면 이기종 데이터베이스 간의 통신을 위해 [Oracle Database Gateways](#)를 구성해야 합니다.

- Aurora PostgreSQL 호환 버전 12.6 및 이전 버전을 소스 데이터베이스로 설정하고 Oracle Database를 대상으로 하는 경우 `oracle_fdw` 확장 프로그램을 기본적으로 사용할 수 없습니다. 대신 Aurora PostgreSQL-Compatible 데이터베이스의 `postgres_fdw` 확장 프로그램을 사용하고 Amazon Elastic Compute Cloud(Amazon EC2)에서 만든 PostgreSQL 데이터베이스에서 `oracle_fdw`를 구성할 수 있습니다. 이 데이터베이스는 Aurora PostgreSQL-Compatible 데이터베이스와 Oracle Database 사이의 중개자 역할을 합니다. 이 패턴에는 Aurora PostgreSQL 12.6 및 이전 버전에서 데이터베이스 링크를 설정하는 두 가지 옵션이 포함되어 있습니다.
- Amazon Route 53의 내부 도메인 이름 시스템(DNS) 항목을 업데이트하는 Amazon EC2 시작 스크립트를 사용하여 Amazon EC2 Auto Scaling 그룹의 EC2 인스턴스를 구성합니다.
- 고가용성(HA)을 위한 Network Load Balancer를 사용하여 Amazon EC2 Auto Scaling 그룹의 EC2 인스턴스를 구성합니다.

Aurora PostgreSQL-Compatible 버전 12.7 이상 간에 데이터베이스 링크를 설정하는 경우 `oracle_fdw` 확장 프로그램을 사용할 수 있습니다.

사전 조건 및 제한 사항

사전 조건

- Virtual Private Cloud(VPC)의 Amazon Aurora PostgreSQL-Compatible 데이터베이스
- Oracle과 Aurora PostgreSQL-Compatible 데이터베이스 간의 네트워크 연결

제한 사항

- 현재는 Oracle용 Amazon Relational Database Service(Amazon RDS)를 소스 데이터베이스로 사용하고 Aurora PostgreSQL-Compatible을 대상 데이터베이스로 사용하여 데이터베이스 링크를 설정할 수 없습니다.

제품 버전

- Oracle Database 11g 이상
- Aurora PostgreSQL-Compatible 11 이상

아키텍처

소스 기술 스택

마이그레이션하기 전에 소스 Oracle 데이터베이스는 데이터베이스 링크를 사용하여 다른 Oracle 데이터베이스의 객체에 액세스할 수 있습니다. 이는 온프레미스 또는 AWS 클라우드에 있는 Oracle 데이터베이스 간에 기본적으로 작동합니다.

대상 기술 스택

옵션 1

- Amazon Aurora PostgreSQL 호환 에디션
- Amazon EC2 인스턴스의 PostgreSQL 데이터베이스
- Amazon EC2 Auto Scaling 그룹
- Amazon Route 53
- Amazon Simple Notification Service(Amazon SNS)
- AWS Identity and Access Management(IAM)
- AWS Direct Connect

옵션 2

- Amazon Aurora PostgreSQL 호환 에디션
- Amazon EC2 인스턴스의 PostgreSQL 데이터베이스
- Amazon EC2 Auto Scaling 그룹
- Network Load Balancer
- Amazon SNS
- Direct Connect

옵션 3

- Amazon Aurora PostgreSQL 호환 에디션
- Direct Connect

대상 아키텍처

옵션 1

다음 다이어그램은 Amazon EC2 Auto Scaling 그룹 및 Route 53에서 제공하는 HA와 함께 `oracle_fdw` 및 `postgres_fdw` 확장 프로그램을 사용한 데이터베이스 링크 설정을 보여줍니다.

1. `postgres_fdw` 확장 프로그램을 갖춘 Aurora PostgreSQL-Compatible 인스턴스는 Amazon EC2의 PostgreSQL 데이터베이스에 연결됩니다.
2. `oracle_fdw` 확장 프로그램이 있는 PostgreSQL 데이터베이스가 오토 스케일링 그룹에 속합니다.
3. Amazon EC2의 PostgreSQL 데이터베이스는 Direct Connect를 사용하여 온프레미스의 Oracle Database에 연결합니다.
4. Oracle Database는 Oracle Database에서 AWS의 PostgreSQL 데이터베이스로 연결할 수 있도록 Oracle Database Gateways로 구성되어 있습니다.
5. IAM은 Amazon EC2에 Route 53 레코드를 업데이트할 수 있는 권한을 부여합니다.
6. Amazon SNS는 자동 조정 작업에 대한 알림을 보냅니다.
7. Route 53에 구성된 도메인 이름은 PostgreSQL Amazon EC2 인스턴스 IP 주소를 가리킵니다.

옵션 2

다음 다이어그램은 Auto Scaling 그룹 및 Network Load Balancer에서 제공하는 HA와 함께 `oracle_fdw` 및 `postgres_fdw` 확장 프로그램을 사용한 데이터베이스 링크 설정을 보여줍니다.

1. `postgres_fdw` 확장 프로그램이 있는 Aurora PostgreSQL-Compatible 인스턴스는 네트워크 로드 밸런서에 연결됩니다.
2. Network Load Balancer는 Aurora PostgreSQL-Compatible 데이터베이스의 연결을 Amazon EC2의 PostgreSQL 데이터베이스로 분산합니다.
3. `oracle_fdw` 확장 프로그램이 있는 PostgreSQL 데이터베이스가 오토 스케일링 그룹에 속합니다.
4. Amazon EC2의 PostgreSQL 데이터베이스는 Direct Connect를 사용하여 온프레미스의 Oracle Database에 연결합니다.
5. Oracle Database는 Oracle Database에서 AWS의 PostgreSQL 데이터베이스로 연결할 수 있도록 Oracle Database Gateways로 구성되어 있습니다.
6. Amazon SNS는 자동 조정 작업에 대한 알림을 보냅니다.

옵션 3

다음 다이어그램은 Aurora PostgreSQL-Compatible 데이터베이스의 `oracle_fdw` 확장 프로그램을 사용한 데이터베이스 링크 설정을 보여줍니다.

1. `oracle_fdw` 확장 프로그램이 있는 Aurora PostgreSQL-Compatible 인스턴스는 Direct Connect를 사용하여 Oracle Database에 연결합니다.
2. Oracle 서버에 설정된 Oracle Database Gateways를 사용하면 Direct Connect를 통해 Aurora PostgreSQL-Compatible 데이터베이스에 연결할 수 있습니다.

도구

서비스

- [Amazon Aurora PostgreSQL 호환 버전](#)은 PostgreSQL 배포를 설정, 운영 및 확장할 수 있고 ACID를 준수하는 완전관리형 관계형 데이터베이스 엔진입니다.
- [AWS Direct Connect](#)를 사용하면 표준 이더넷 광섬유 케이블을 통해 내부 네트워크를 Direct Connection 위치에 연결할 수 있습니다. 이 연결을 구성하면, 네트워크 경로에서 인터넷 서비스 제공 업체를 우회하여 퍼블릭 서비스에 직접 가상 인터페이스를 생성할 수 있습니다.
- [Amazon Elastic Compute Cloud\(Amazon EC2\)](#)는 AWS 클라우드에서 확장 가능한 컴퓨팅 용량을 제공합니다. 필요한 만큼 가상 서버를 시작하고 빠르게 스케일 업하거나 스케일 다운할 수 있습니다. 이 패턴에서 옵션 1과 2는 EC2 인스턴스를 사용하여 PostgreSQL 데이터베이스를 호스팅합니다.
- [Amazon EC2 Auto Scaling](#)을 사용하면 애플리케이션 가용성을 유지하고 정의된 조건에 따라 Amazon EC2 인스턴스를 자동으로 추가하거나 제거할 수 있습니다.
- [AWS Identity and Access Management\(IAM\)](#)는 사용자에 대한 인증 및 권한 부여를 제어함으로써 AWS 리소스에 대한 액세스를 안전하게 제어할 수 있습니다.
- [Amazon Route 53](#)은 가용성과 확장성이 뛰어난 DNS 웹 서비스입니다.
- [Amazon Simple Notification Service\(Amazon SNS\)](#)를 사용하면 웹 서버 및 이메일 주소를 포함하여 게시자와 클라이언트 간의 메시지 교환을 조정하고 관리할 수 있습니다.
- [Elastic Load Balancing\(ELB\)](#)은 들어오는 애플리케이션 또는 네트워크 트래픽을 여러 대상에 분산합니다. 예를 들어 하나 이상의 가용 영역에 있는 Amazon Elastic Compute Cloud(Amazon EC2) 인스턴스, 컨테이너, IP 주소 전반에 걸쳐 트래픽을 분산할 수 있습니다. 이 패턴은 Network Load Balancer를 사용합니다.

기타 서비스

- [Oracle Database Gateways](#)는 비Oracle 시스템의 데이터에 액세스할 수 있는 기능을 Oracle Database에 제공합니다.

에픽

옵션 1과 옵션 2의 일반적인 설정 작업

작업	설명	필요한 기술
<p>EC2 인스턴스를 생성하고 oracle_fdw PostgreSQL 확장 프로그램을 구성합니다.</p>	<ol style="list-style-type: none"> 1. Amazon Linux 2 운영 체제로 EC2 인스턴스를 생성합니다. 2. PostgreSQL을 설치하려면 EC2 인스턴스에 ec2-user로 로그인하고 다음 명령을 실행합니다. <pre data-bbox="630 789 1029 1879"> sudo su - root sudo tee /etc/yum.repos.d/pgdg.repo< EOF [pgdg12] name=PostgreSQL 12 for RHEL/CentOS 7 - x86_64 baseurl=https://download.postgresql.org/pub/repos/yum/12/redhat/rhel-7-x86_64 enabled=1 gpgcheck=0 EOF sudo yum install -y postgresql12-server sudo yum install postgresql12-devel sudo /usr/pgsql-12/bin/postgresql-12-setup initdb sudo systemctl enable postgresql-12 </pre>	<p>클라우드 관리자, DBA</p>

작업	설명	필요한 기술
	<pre data-bbox="630 205 1026 306">sudo systemctl start postgresql-12</pre> <p data-bbox="591 319 1026 453">3. 에서 oracle_fdw 소스 코드를 GitHub 다운로드합니다.</p> <pre data-bbox="630 487 1026 961">mkdir -p /var/lib/ pgsql/oracle_fdw/ cd /var/lib/pgsql/ oracle_fdw/ wget https://g ithub.com/laurenz/ oracle_fdw/archive /refs/heads/master .zip unzip master.zip</pre> <p data-bbox="591 982 1026 1117">4. Oracle Instant Client를 설치하고 Oracle 환경 변수를 설정합니다.</p> <pre data-bbox="630 1150 1026 1507">yum install https://d ownload.oracle.com /otn_software/linu x/instantclient/19 12000/oracle-insta ntclient19.12-basi c-19.12.0.0.0-1.x8 6_64.rpm</pre> <pre data-bbox="630 1541 1026 1789">yum install https://d ownload.oracle.com /otn_software/linu x/instantclient/19 12000/oracle-insta ntclient19.12-deve</pre>	

작업	설명	필요한 기술
	<pre>1-19.12.0.0.0-1.x86_64.rpm</pre> <pre>export ORACLE_H0 ME=/usr/lib/oracle /19.12/client64exp ort LD_LIBRAR Y_PATH=/usr/lib/or acle/19.12/client6 4/lib:\$LD_LIBRARY_ PATH</pre> <p>5. pg_config 가 올바른 버전을 나타내도록 합니다.</p> <pre>which pg_config</pre> <p>6. oracle_fdw 를 컴파일합니다.</p> <pre>cd /var/lib/pgsql/oracle_fdw/oracle_fdw-master make make install</pre> <p>참고: oci.h가 누락되었다는 오류 메시지가 표시되면 Makefile에 다음을 추가합니다.</p> <ul style="list-style-type: none"> • PG_CPPFLAGS 에 -I/usr/include/oracle/19.12/client64 추가 • SHLIB_LINK 에 -L/usr/lib/oracle/ 	

작업	설명	필요한 기술
	<p>19.12/client64/ lib 추가</p> <p>자세한 내용은 oracle_fdw 리포지토리를 참조하세요.</p> <p>7. PostgreSQL 데이터베이스에 로그인하여 oracle_fdw 확장 프로그램을 생성합니다.</p> <pre data-bbox="630 659 1029 856">sudo su - postgres psql postgres create extension oracle_fdw;</pre> <p>8. 외부 테이블을 소유할 PostgreSQL 사용자를 생성합니다.</p> <pre data-bbox="630 1041 1029 1318">CREATE USER pguser WITH PASSWORD '<password>'; GRANT CONNECT ON DATABASE postgres TO pguser;</pre> <p>9. 외부 데이터 래퍼를 생성합니다. 다음 값을 Oracle Database 서버 세부 정보로 대체합니다.</p> <ul data-bbox="630 1537 1029 1738" style="list-style-type: none"> • <Oracle DB Server IP> • <Oracle DB Port> • <Oracle_SID> <pre data-bbox="630 1772 1029 1864">create server oradb foreign data wrapper</pre>	

작업	설명	필요한 기술
	<pre data-bbox="630 205 1026 541"> oracle_fdw options (dbserver '//<Oracle DB Server IP>:<Oracle DB Port>/<Oracle_SID>'); GRANT USAGE ON FOREIGN SERVER oradb TO pguser; </pre> <p data-bbox="591 558 1026 1071"> 10사용자 매핑과 Oracle 테이블에 매핑되는 외부 테이블을 생성하려면 PostgreSQL 데이터베이스에 pguser로 연결하고 다음 명령을 실행합니다. 예제 코드에서는 NAME_DATA 테이블이 포함된 Oracle 스키마로 DMS_SAMPLE 이 사용되며 dms_sample 은 암호입니다. 필요에 따라 교체합니다. </p> <pre data-bbox="630 1108 1026 1390"> create user mapping for pguser server oradb options (user 'DMS_SAMPLE', password 'dms_sample'); </pre> <p data-bbox="630 1423 1010 1843"> 참고: 다음 예제에서는 Oracle Database의 테이블에 대해 PostgreSQL에서 외부 테이블을 만듭니다. PostgreSQL 인스턴스에서 액세스해야 하는 모든 Oracle 테이블에 대해 유사한 외부 테이블을 생성해야 합니다. </p>	

작업	설명	필요한 기술
	<pre>CREATE FOREIGN TABLE name_data(name_type CHARACTER VARYING(1 5) NOT NULL, name CHARACTER VARYING(45) NOT NULL) SERVER oradb OPTIONS (schema 'DMS_SAMPLE', table 'NAME_DATA');</pre> <p>select count(*) from name_data;</p> <p>11 PostgreSQL 데이터베이스 를 시작하는 동안 Oracle 라 이브러리를 찾을 수 있도록 EC2 인스턴스에 PostgreSQ L 데이터베이스를 구성합니 다. 이는 oracle_fdw 확 장 프로그램에 필요합니다.</p> <pre>sudo systemctl stop postgresql-12</pre> <p>참고: systemctl 시작 프 로그램이 oracle_fdw 에 서 필요한 Oracle 라이브러 리를 찾을 수 있도록 환경 변 수를 포함하게 /usr/lib/ systemd/system/pos tgresql-12.service 파일을 편집합니다.</p> <pre># Oracle Environment Variables</pre>	

작업	설명	필요한 기술
	<pre>Environment=ORACLE_HOME=/u01/app/oracle/product/12.2.0.1/db_1 Environment=LD_LIBRARY_PATH=/u01/app/oracle/product/12.2.0.1/db_1/lib:/lib:/usr/lib sudo systemctl start postgresql-12</pre>	

옵션 1: oracle_fdw 및 postgres_fdw 확장 프로그램, Auto Scaling 그룹 및 Route 53을 사용하여 데이터베이스 링크를 설정합니다.

작업	설명	필요한 기술
<p>Amazon Route 53에서 프라이빗 호스팅 영역을 설정합니다.</p>	<ol style="list-style-type: none"> Amazon Route 53에서 프라이빗 호스팅 영역을 생성합니다. EC2 인스턴스와 연결될 도메인 이름을 기록해 둡니다. oracle_fdw PostgreSQL 확장 프로그램이 포함된 EC2 인스턴스 IP 주소로 확인되는 단순 라우팅 정책을 사용하여 "A" 레코드를 추가합니다. "A" 레코드를 저장한 후 1단계에서 입력한 도메인 이름의 호스팅 영역 ID를 기록해 둡니다. 이는 적절한 IAM 정책을 생성하는 데 사용됩니다. 	<p>DBA, 클라우드 관리자</p>

작업	설명	필요한 기술
<p>EC2 인스턴스에 연결할 IAM 역할을 생성합니다.</p>	<p>EC2 인스턴스에 연결할 IAM 역할을 생성하려면 다음 정책을 사용합니다. 이전 스토리에서 캡처한 정보로 <Hosted zone ID>를 대체합니다.</p> <pre data-bbox="597 491 1029 1717"> { "Version": "2012-10-17", "Statement": [{ "Sid": "VisualEditor0", "Effect": "Allow", "Action": "route53:ChangeResourceRecordSets", "Resource": "arn:aws:route53::hostedzone/<Hosted zone ID>" }, { "Sid": "VisualEditor1", "Effect": "Allow", "Action": "route53:ListHostedZones", "Resource": "*" }] } </pre>	<p>클라우드 관리자, DBA</p>

작업	설명	필요한 기술
<p>EC2 시작 템플릿을 생성합니다.</p>	<ol style="list-style-type: none"> 1. oracle_fdw PostgreSQL 확장 프로그램이 포함된 EC2 인스턴스의 AMI를 생성합니다. 2. AMI를 사용하여 EC2 시작 템플릿을 생성합니다. 3. Aurora PostgreSQL-Compatible 인스턴스에서 EC2 인스턴스의 PostgreSQL 데이터베이스로의 연결을 허용하려면 이전에 만든 IAM 역할을 연결하고 보안 그룹을 연결합니다. 4. 사용자 데이터 섹션에서 다음 명령을 추가하여 Hosted zone ID 및 Domain Name을 적절한 값으로 변경합니다. 시작 템플릿 생성을 선택합니다. <pre data-bbox="630 1186 1029 1879"> #!/bin/bash v_zone_id='Hosted zone ID' v_domain_name=' Domain Name' v_local_ipv4=\$(curl -s http://16 9.254.169.254/latest/meta-data/local- ipv4) aws route53 change-re source-record-sets --hosted-zone-id \$v_zone_id --change- batch '{"Change </pre>	<p>클라우드 관리자, DBA</p>

작업	설명	필요한 기술
	<pre>s":[{"Action":"UPS ERT","ResourceReco rdSet":{"Name":"'v_ v_domain_name',"T ype":"A","TTL":10, "ResourceRecords": [{"Value":"'v_loc al_ipv4'"}]}]}</pre>	

작업	설명	필요한 기술
<p>오토 스케일링 그룹을 설정합니다.</p>	<ol style="list-style-type: none"> 오토 스케일링 그룹을 설정하려면 이전 단계에서 생성한 시작 템플릿을 사용합니다. EC2 인스턴스를 시작하는데 사용할 적절한 VPC와 서브넷을 구성합니다. 옵션 1 설치에서는 Load Balancer를 사용하지 않습니다. 조정 정책에서 원하는 용량, 최소 용량, 최대 용량을 1로 설정합니다. 운영 팀에 알림을 보내려면 시작 또는 종료와 같은 이벤트에 대한 알림을 추가합니다. 구성을 검토하고 오토 스케일링 그룹 생성을 선택합니다. <p>완료되면 오토 스케일링 그룹은 Oracle Database에 연결되는 <code>oracle_fdw</code> PostgreSQL 확장 프로그램이 포함된 EC2 인스턴스를 시작합니다.</p> <p>참고: 새 Oracle 테이블에 액세스하거나 Oracle 테이블의 구조를 변경해야 하는 경우 해당 변경 사항을 PostgreSQL 외부 테이블에 반영해야 합니다. 변경 사항을 구현한 후에는 EC2 인스턴스의 새 AMI를 생성하고</p>	<p>클라우드 관리자, DBA</p>

작업	설명	필요한 기술
	이를 사용하여 시작 템플릿을 구성해야 합니다.	

작업	설명	필요한 기술
<p>Aurora PostgreSQL-Compatible 인스턴스에서 postgres_fdw 확장 프로그램을 구성합니다.</p>	<ol style="list-style-type: none"> 1. Aurora PostgreSQL-Compatible 인스턴스에서 postgres_fdw 를 구성합니다. 이는 Amazon EC2의 PostgreSQL 데이터베이스에 연결되며, 이 데이터베이스는 Aurora PostgreSQL-Compatible 인스턴스와 Oracle Database 사이의 중간 노드 역할을 합니다. 2. Aurora PostgreSQL-Compatible 인스턴스에 연결하고 다음 명령을 실행합니다. <pre data-bbox="633 928 1029 1816"> create extension postgres_fdw; CREATE SERVER pgoradb FOREIGN DATA WRAPPER postgres_fdw OPTIONS (dbname 'postgres', host 'Domain Name', port '5432'); CREATE USER MAPPING for postgres SERVER pgoradb OPTIONS (user 'pguser', password '<password>'); CREATE FOREIGN TABLE data_mart.name_data(name_type CHARACTER VARYING(15) NOT NULL,</pre>	<p>클라우드 관리자, DBA</p>

작업	설명	필요한 기술
	<pre data-bbox="630 212 1029 625"> name CHARACTER VARYING(45) NOT NULL) SERVER pgoradb OPTIONS (schema_name 'public', table_name 'name_data'); select count(*) from data_mart.name_data; </pre> <p data-bbox="591 695 1024 919">이로써 Aurora PostgreSQL-Compatible 데이터베이스 링크에서 Oracle Database로의 데이터베이스 링크 설정이 완료되었습니다.</p> <p data-bbox="591 961 1016 1570">이 솔루션은 PostgreSQL 데이터베이스를 호스팅하는 EC2 인스턴스에 장애가 발생하는 경우에 대비한 재해 복구(DR) 전략을 제공합니다. 오토 스케일링 그룹은 새 EC2 인스턴스를 시작하고 새 EC2 인스턴스의 IP 주소로 DNS를 업데이트합니다. 이렇게 하면 Aurora PostgreSQL-Compatible 인스턴스의 외부 테이블이 수동 개입 없이 Oracle 테이블에 액세스할 수 있습니다.</p>	

옵션 2: oracle_fdw 및 postgres_fdw 확장, 오토 스케일링 그룹 및 Network Load Balancer를 사용하여 데이터베이스 링크를 설정합니다.

작업	설명	필요한 기술
<p>EC2 시작 템플릿을 생성합니다.</p>	<ol style="list-style-type: none"> 1. oracle_fdw PostgreSQL 확장 프로그램이 포함된 EC2 인스턴스의 AMI를 생성합니다. 2. AMI를 사용하여 EC2 시작 템플릿을 생성합니다. 	<p>클라우드 관리자, DBA</p>
<p>대상 그룹, Network Load Balancer, 오토 스케일링 그룹을 설정합니다.</p>	<ol style="list-style-type: none"> 1. 대상 그룹을 만들려면 대상 유형으로 인스턴스를 선택합니다. 프로토콜에서 TCP를 선택하고 포트에서 5432를 선택합니다. 그런 다음 대상 그룹을 배치할 VPC를 선택하고 적절한 상태 확인을 선택합니다. 2. VPC에서 내부 Network Load Balancer를 생성합니다. 프로토콜:포트 TCP:5432에서 수신하도록 로드 밸런서를 구성합니다. 기본 작업을 전달 대상으로 설정하고 생성한 대상 그룹을 선택합니다. 3. 생성한 시작 템플릿을 사용하여 오토 스케일링 그룹을 생성합니다. 4. EC2 인스턴스를 시작하는데 사용할 적절한 VPC와 서브넷으로 오토 스케일링 그룹을 구성합니다. 	<p>클라우드 관리자, DBA</p>

작업	설명	필요한 기술
	<p>5. 로드 밸런싱 옵션의 경우 기존 로드 밸런서에 연결을 선택하고 생성한 대상 그룹을 선택합니다. 상태 확인에서 ELB를 선택합니다.</p> <p>6. 조정 정책에서 HA 부하를 지원하는 데 필요한 바에 따라서 원하는 용량과 최소 용량을 2로 설정하고 최대 용량을 더 큰 수로 설정합니다.</p> <p>7. 운영 팀에 알림을 보내려면 시작 또는 종료와 같은 이벤트에 대한 알림을 추가합니다.</p> <p>8. 구성을 검토하고 오토 스케일링 그룹 생성을 선택합니다.</p> <p>완료되면 오토 스케일링 그룹은 Oracle Database에 연결되는 <code>oracle_fdw</code> PostgreSQL 확장 프로그램을 포함하는 EC2 인스턴스를 원하는 수만큼 시작합니다.</p> <p>참고: 새 Oracle 테이블에 액세스하거나 Oracle 테이블의 구조를 변경해야 하는 경우 해당 변경 사항을 PostgreSQL 외부 테이블에 반영해야 합니다. 변경 사항을 구현한 후에는 EC2 인스턴스의 새 AMI를 생성하고 이를 사용하여 시작 템플릿을 구성해야 합니다.</p>	

작업	설명	필요한 기술
<p>Aurora PostgreSQL-Compatible 인스턴스에서 postgres_fdw 확장 프로그램을 구성합니다.</p>	<p>Aurora PostgreSQL-Compatible 인스턴스에서 postgres_fdw 를 구성합니다. 이는 Network Load Balancer를 통해 EC2의 PostgreSQL 데이터베이스에 연결됩니다. EC2의 PostgreSQL 인스턴스는 Aurora PostgreSQL-Compatible 인스턴스와 Oracle Database 사이의 중간 노드 역할을 합니다.</p> <p>Aurora PostgreSQL-Compatible 인스턴스에 연결하고 다음 명령을 실행합니다.</p> <pre data-bbox="592 951 1029 1877"> create extension postgres_fdw; CREATE SERVER pgoradb FOREIGN DATA WRAPPER postgres_fdw OPTIONS (dbname 'postgres ', host 'DNS name of Network Load Balancer' , port '5432'); CREATE USER MAPPING for postgres SERVER pgoradb OPTIONS (user 'pguser', password '<password>'); CREATE FOREIGN TABLE data_mart.name_data(name_type CHARACTER VARYING(15) NOT NULL, name CHARACTER VARYING(45) NOT NULL </pre>	<p>클라우드 관리자, DBA</p>

작업	설명	필요한 기술
	<pre data-bbox="613 212 1010 506">) SERVER pgoradb OPTIONS (schema_name 'public', table_name 'name_data'); select count(*) from data_mart.name_data; </pre> <p data-bbox="592 541 1003 720">이것으로 Aurora PostgreSQL-Compatible에서 Oracle Database로의 데이터베이스 링크 설정이 완료되었습니다.</p> <p data-bbox="592 764 1013 1465">PostgreSQL 데이터베이스를 호스팅하는 EC2에 장애가 발생하는 경우 Network Load Balancer는 장애를 식별하고 장애가 발생한 EC2 인스턴스로 향하는 트래픽을 중지합니다. 오토 스케일링 그룹은 새 EC2 인스턴스를 시작하고 이를 로드 밸런서에 등록합니다. 이렇게 하면 기존 EC2 인스턴스에 장애가 발생한 후 Aurora PostgreSQL-Compatible 인스턴스의 외부 테이블이 수동 개입 없이 Oracle 테이블에 액세스할 수 있습니다.</p>	

옵션 3: Aurora PostgreSQL-Compatible 데이터베이스에서 oracle_fdw 확장 프로그램을 사용하여 데이터베이스 링크를 설정합니다.

작업	설명	필요한 기술
<p>Aurora PostgreSQL-Compatible 인스턴스에서 oracle_fdw 확장 프로그램을 구성합니다.</p>	<p>Aurora PostgreSQL-Compatible 데이터베이스 버전 12.7 이상의 경우 oracle_fdw 확장 프로그램을 기본적으로 사용할 수 있습니다. 따라서 EC2 인스턴스에서 중간 PostgreSQL 데이터베이스를 생성할 필요가 없습니다. Aurora PostgreSQL-Compatible 인스턴스는 Oracle Database에 직접 연결할 수 있습니다.</p> <ol style="list-style-type: none"> oracle_fdw 확장 프로그램을 생성하려면 Aurora PostgreSQL-Compatible 인스턴스에 로그인하고 다음 명령을 실행합니다. <div data-bbox="630 1199 1029 1318" style="border: 1px solid #ccc; border-radius: 10px; padding: 10px; margin: 10px 0;"> <pre>create extension oracle_fdw;</pre> </div> 외부 데이터 래퍼를 생성합니다. 다음 값을 Oracle Database 서버 세부 정보로 대체합니다. <ul style="list-style-type: none"> <Oracle DB Server IP> <Oracle DB Port> <Oracle_SID> <div data-bbox="630 1770 1029 1869" style="border: 1px solid #ccc; border-radius: 10px; padding: 10px; margin: 10px 0;"> <pre>create server oradb foreign data wrapper</pre> </div> 	<p>클라우드 관리자, DBA</p>

작업	설명	필요한 기술
	<pre data-bbox="646 212 993 415">oracle_fdw options (dbserver '//<Orac le DB Server IP>:<Orac le DB Port>/<Or acle_SID>');</pre> <p data-bbox="592 443 1026 1094">3. 사용자 매핑과 Oracle 테이블에 매핑되는 외부 테이블을 생성하려면 다음 명령을 실행합니다. 예제 코드에서는 NAME_DATA 테이블이 포함된 Oracle 스키마로 DMS_SAMPLE 이 사용되며 dms_sample 은 암호입니다. 필요에 따라 교체합니다. 또한 다른 모든 Oracle 테이블에 액세스하려면 Aurora PostgreSQL-Compatible 인스턴스에서 외부 테이블을 생성해야 합니다.</p> <pre data-bbox="646 1157 993 1856">create user mapping for postgres server oradb options (user 'DMS_SAMPLE', password 'dms_samp le'); CREATE FOREIGN TABLE name_data(name_type character varying(1 5) OPTIONS (key 'true') NOT NULL, name character varying(45) OPTIONS (key 'true') NOT NULL)SERVER oradb OPTIONS (schema 'DMS_SAMP</pre>	

작업	설명	필요한 기술
	<pre data-bbox="630 205 1026 310">LE', table 'NAME_DATA');</pre> <p data-bbox="630 340 1026 571">PostgreSQL 인스턴스에서 액세스해야 하는 모든 Oracle 테이블에 대해 유사한 외부 테이블을 생성해야 합니다.</p>	

온프레미스 Oracle Database에서 Aurora PostgreSQL-Compatible로 연결하도록 Oracle Database Gateways를 설정합니다.

작업	설명	필요한 기술
<p data-bbox="116 896 545 1022">온프레미스 Oracle Database 서버에서 게이트웨이를 구성합니다.</p>	<ol data-bbox="594 896 1029 1022" style="list-style-type: none"> 루트 사용자로 최신 UnixODBC 드라이버 관리자를 설치합니다. <pre data-bbox="630 1066 1026 1180">sudo yum install unixODBC*</pre> <ol data-bbox="594 1201 1029 1327" style="list-style-type: none"> PostgreSQL ODBC 드라이버(psqlODBC)를 설치합니다. <pre data-bbox="630 1371 1026 1873">sudo wget https://download.postgresql.org/pub/repos/yum/reporepms/EL-7-x86_64/pgdg-redhat-repo-latest.noarch.rpm sudo yum install pgdg-redhat-repo-latest.noarch.rpm sudo yum install postgresql12-odbc</pre>	<p data-bbox="1068 896 1136 928">DBA</p>

작업	설명	필요한 기술
	<p>3. 드라이버의 ODBC 데이터 소스 이름(DSN)을 생성합니다.</p> <p>UnixODBC 드라이버 관리자는 드라이버를 구성하고 테스트하는 데 사용되는 <code>odbcinst</code>, <code>odbc_config</code>, <code>isql</code> 명령줄 유틸리티를 제공합니다. <code>odbcinst</code> 또는 <code>odbc_config</code> 유틸리티를 사용하여 DSN을 생성하는 데 필요한 드라이버 정보를 전달할 <code>unixODBC</code> 드라이버 관리자 파일을 찾을 수 있습니다.</p> <pre data-bbox="630 982 1029 1066">odbcinst -j</pre> <p>다음 코드는 출력의 예입니다.</p> <pre data-bbox="630 1220 1029 1871">unixODBC 2.3.1 DRIVERS.....: /etc/odbcinst.ini SYSTEM DATA SOURCES: /etc/odbc.ini FILE DATA SOURCES.. : /etc/ODBCDataSources USER DATA SOURCES.. : /root/.odbc.ini SQLULEN Size.....: 8 SQLLEN Size.....: 8</pre>	

작업	설명	필요한 기술
	<pre>SQLSETPOSIROW Size.: 8 odbc_config --odbcini --odbcinstini /etc/odbc.ini /etc/odbcinst.ini</pre> <p>예제 출력에서 odbcinst.ini 및 odbc.ini 파일을 볼 수 있습니다. 기본적으로 odbcinst.ini 는 환경의 ODBC 드라이버용 레지스트리 및 구성 파일이고 odbc.ini는 ODBC DSN의 레지스트리 및 구성 파일입니다. 드라이버를 활성화하려면 이 두 파일을 수정해야 합니다.</p> <p>4. ODBC 드라이버 파일 / etc/odbcinst.ini 에서 psq1ODBC 드라이버 라이브러리를 구성하고 파일 끝에 다음 줄을 추가합니다. 이 줄은 드라이버에 대한 항목을 만듭니다.</p> <pre>[PostgreSQL] Description = ODBC for PostgreSQL Driver = / usr/lib/psqlodbcw.so Setup = / usr/lib/libodbcps qlS.so</pre>	

작업	설명	필요한 기술
	<pre>Driver64 = /usr/lib64/psqlodbcw.so Setup64 = /usr/lib64/libodbcpsqlS.so FileUsage = 1</pre> <p>5. /etc/odbc.ini 파일에서 DSN을 생성합니다. 드라이버 관리자는 이 파일을 읽고 odbcinst.ini 에서 지정한 드라이버 세부 정보를 사용하여 데이터베이스에 연결하는 방법을 결정합니다. 다음 파라미터를 실제 값으로 변경합니다.</p> <ul style="list-style-type: none"> • <PostgreSQL Port> • <PostgreSQL Database Name> • <Aurora PostgreSQL Endpoint> • <PostgreSQL username> • <PostgreSQL password> <pre>[pgdsn] Driver=/usr/pgsql-12/lib/psqlodbc.so Description=PostgreSQL ODBC Driver Database=<PostgreSQL Database Name> Servername=<Aurora PostgreSQL Endpoint></pre>	

작업	설명	필요한 기술
	<pre> Username=<Postgre SQL username> Password=<PostgreSQL password> Port=<PostgreSQL Port> UseDeclareFetch=1 CommLog=/tmp/ pgodblink.log Debug=1 LowerCaseIde ntifier=1 </pre> <p>6. isql 유틸리티를 사용하여 생성한 PostgreSQL 데이터베이스 DSN에 대한 ODBC 연결(psqlODBC)을 테스트합니다.</p> <pre> isql -v pgdsn </pre> <p>다음 코드는 출력의 예입니다.</p> <pre> +-----+ Connected! sql-statement help [tablename] quit </pre>	

작업	설명	필요한 기술
	<pre data-bbox="630 212 1029 386">+-----+ -----+ quit</pre> <p data-bbox="591 401 1029 531">7. DSN을 사용하여 ODBC(HS) 서비스 핸들러용 게이트웨이를 생성합니다.</p> <p data-bbox="630 575 1013 898">oracle 사용자로 \$ORACLE_HOME/hs/admin 위치에 <code>initDSN.ora</code> 파일을 생성합니다. 이 경우 <code>pgdsn</code>은 DSN이므로 <code>initpgdsn.ora</code> 라는 파일을 만들어야 합니다.</p> <pre data-bbox="630 936 1029 1016">more initpgdsn.ora</pre> <p data-bbox="630 1052 1013 1136">다음 코드는 출력의 예입니다.</p> <pre data-bbox="630 1171 1029 1785"># This is a sample agent init file that contains the HS parameters that are # needed for the Database Gateway for ODBC # # HS init parameters # HS_FDS_CONNEC T_INFO=pgdsn HS_FDS_TRACE_L EVEL=OFF</pre>	

작업	설명	필요한 기술
	<pre> HS_FDS_TRACE_FILE_ NAME=/tmp/ora_hs_t race.log HS_FDS_SHAREABLE_N AME=/usr/lib64/lib odbc.so HS_NLS_NCHAR=UCS2 HS_LANGUAGE=AMERICA N_AMERICA.AL32UTF8 # # ODBC specific environment variables # set ODBCINI=/etc/ odbc.ini </pre> <p>8. SID_LIST_LISTENER 의 DSN 항목을 추가하여 리스너(\$ORACLE_HOME/network/admin/listener.ora)를 조정합니다.</p> <pre> more \$ORACLE_HOME/ network/admin/ listener.ora </pre> <p>다음 코드는 출력의 예입니다.</p> <pre> SID_LIST_LISTENER = (SID_LIST = (SID_DESC= (SID_NAME = pgdsn) (ORACLE_HOME = / u01/app/oracle/pr </pre>	

작업	설명	필요한 기술
	<pre data-bbox="646 212 993 646"> oduct/12.2.0.1/db_ 1) (ENVS="LD _LIBRARY_PATH=/lib 64:/usr/lib:/usr/l ib64:/u01/app/orac le/product/12.2.0. 1/db_1") (PROGRAM=dg4odbc))) </pre> <p data-bbox="591 680 1013 905">9. DSN 항목을 추가하여 tnsname(\$ORACLE_HOME/network/admin/tnsnames.ora)을 조정합니다.</p> <pre data-bbox="646 947 993 1100"> more \$ORACLE_HOME/ network/admin/ tnsnames.ora </pre> <p data-bbox="630 1136 1013 1220">다음 코드는 출력의 예입니다.</p> <pre data-bbox="646 1262 993 1528"> pgdsn=(DESCRIPTION =(ADDRESS=(PROTOCO L=tcp)(HOST=localh ost)(PORT=1521))(C ONNECT_DATA=(SID=p gdsn))(HS=OK)) </pre> <p data-bbox="591 1556 1013 1822">10.네트워킹 파일에 대한 DSN 관련 항목이 효과적일 수 있도록 Oracle 리스너를 재시작하고 <Listener Name>을 Oracle 리스너 이름으로 변경합니다.</p>	

작업	설명	필요한 기술
	<pre data-bbox="634 226 1003 394">lsnrctl stop <Listener Name> lsnrctl start <Listener Name></pre> <p data-bbox="630 447 1027 625">Oracle 리스너를 재시작하면 DSN 이름(pgdsn)을 사용하여 Oracle HS 핸들러가 생성됩니다.</p> <p data-bbox="594 646 1016 919">11.Oracle Database에 로그인해서 PostgreSQL 데이터베이스에 액세스할 수 있는 Oracle Database 링크를 DSN을 사용하여 생성합니다.</p> <pre data-bbox="634 972 1003 1192">create public database link pgdb connect to "postgres" identified by "postgres" using 'pgdsn';</pre> <p data-bbox="594 1213 1027 1339">12.생성된 Oracle 데이터베이스 링크를 사용하여 PostgreSQL 데이터에 액세스합니다.</p> <pre data-bbox="634 1392 1003 1497">select count(*) from "pg_tables"@pgdb;</pre>	

관련 리소스

- [Amazon Aurora PostgreSQL](#)
- [Amazon Elastic Compute Cloud\(Amazon EC2\)](#)

- [AWS Identity and Access Management\(IAM\)](#)
- [시작 템플릿에서 인스턴스 시작](#)
- [오토 스케일링 그룹](#)
- [Amazon Route 53](#)
- [Amazon Simple Notification Service\(SNS\)](#)
- [AWS Network Load Balancer](#)
- [Oracle Database Gateways](#)

추가 정보

oracle_fdw 확장 프로그램은 Aurora PostgreSQL-Compatible 버전 12.7 이상에서 사용할 수 있지만, 이 패턴에는 이전 버전의 Aurora PostgreSQL-Compatible 데이터베이스에 대한 솔루션도 포함됩니다. 많은 고객이 이전 버전의 Aurora PostgreSQL-Compatible 데이터베이스를 사용하고 데이터베이스를 업그레이드하려면 여러 수준의 애플리케이션 및 성능 테스트가 필요하기 때문입니다. 또한 데이터베이스 링크 기능이 광범위하게 사용되고 있으며, 이 문서의 목적은 모든 버전의 Aurora PostgreSQL-Compatible에 옵션을 제공하는 것입니다.

AWS DMS를 사용하여 Microsoft SQL Server 데이터베이스를 Amazon S3로 내보내기

작성자: Sweta Krishna(AWS)

환경: PoC 또는 파일럿	소스: Microsoft SQL Server	대상: Amazon S3
R 유형: 리플랫폼	워크로드: Microsoft	기술: 마이그레이션, 데이터베이스
AWS 서비스: AWS DMS, Amazon S3		

요약

조직은 데이터베이스 마이그레이션, 백업 및 복원, 데이터 보관, 데이터 분석을 위해 Amazon Simple Storage Service(S3)에 데이터베이스를 복사해야 하는 경우가 많습니다. 이 패턴은 Microsoft SQL Server 데이터베이스를 Amazon S3로 내보내는 방법을 설명합니다. 소스 데이터베이스는 온프레미스로 호스팅되거나 Amazon Elastic Compute Cloud(Amazon EC2) 또는 Amazon Web Services(AWS) 클라우드의 Amazon Relational Database Service(RDS) for Microsoft SQL Server에서 호스팅될 수 있습니다.

데이터는 AWS Database Migration Service(AWS DMS)를 사용하여 내보냅니다. AWS DMS는 기본적으로 쉘표로 구분된 값(.csv) 형식으로 전체 로드와 변경 데이터 캡처(CDC) 데이터를 작성합니다. 이 패턴은 더 작은 스토리지와 더 빠른 쿼리 옵션을 위해 Apache Parquet(.parquet) 형식 옵션을 사용합니다.

사전 조건 및 제한 사항

사전 조건

- 활성 상태의 AWS 계정
- 대상 S3 버킷에 대한 쓰기, 삭제 및 태그 액세스 권한이 있는 계정의 AWS Identity and Access Management(IAM) 역할 및 이 IAM 역할에 신뢰할 수 있는 개체로 AWS DMS(dms.amazonaws.com) 추가됨
- 온프레미스 Microsoft SQL Server 데이터베이스(또는 EC2 인스턴스의 Microsoft SQL Server 또는 Amazon RDS for SQL Server 데이터베이스)

- AWS의 Virtual Private Cloud(VPC) 및 AWS Direct Connect에서 제공하는 온프레미스 네트워크 또는 가상 프라이빗 네트워크(VPN) 간의 네트워크 연결

제한 사항

- VPC 지원(게이트웨이 VPC) S3 버킷은 현재 3.4.7 이전의 AWS DMS 버전에서 지원되지 않습니다.
- 전체 로드 중에 소스 테이블의 구조 변경은 지원되지 않습니다.
- AWS DMS 전체 대용량 이진 객체(LOB) 모드는 지원되지 않습니다.

제품 버전

- Enterprise, Standard, Workgroup 및 Developer 버전용 Microsoft SQL Server 버전 2005 이상
- 소스로서 Microsoft SQL Server 버전 2019에 대한 지원은 AWS DMS 버전 3.3.2 이상에서 사용할 수 있습니다.

아키텍처

소스 기술 스택

- 온프레미스 Microsoft SQL Server 데이터베이스(또는 EC2 인스턴스의 Microsoft SQL Server 또는 Amazon RDS for SQL Server 데이터베이스)

대상 기술 스택

- AWS Direct Connect
- DMS
- Amazon S3

대상 아키텍처

도구

- [AWS Database Migration Service\(AWS DMS\)](#)를 사용하면 데이터 스토어를 AWS 클라우드로 마이그레이션하거나 클라우드와 온프레미스 설정 조합 간에 데이터 스토어를 마이그레이션할 수 있습니다.

- [AWS Direct Connect](#)는 표준 이더넷 광섬유 케이블을 통해 내부 네트워크를 Direct Connect 위치에 연결합니다. 이 연결을 통해 네트워크 경로의 인터넷 서비스 공급자를 우회하여 퍼블릭 AWS 서비스에 직접 연결하는 가상 인터페이스를 생성할 수 있습니다.
- [Amazon Simple Storage Service\(S3\)](#)는 원하는 양의 데이터를 저장, 보호 및 검색하는 데 도움이 되는 클라우드 기반 객체 스토리지 서비스입니다.

에픽

마이그레이션 준비

작업	설명	필요한 기술
데이터베이스 버전을 검증합니다.	소스 데이터베이스 버전을 검증하고 AWS DMS에서 지원되는지 확인합니다. 지원되는 SQL Server 데이터베이스 버전에 대한 자세한 내용은 Microsoft SQL Server 데이터베이스를 AWS DMS용 소스로 사용 을 참조하세요.	DBA
VPC 및 보안 그룹을 생성합니다.	AWS 계정에서 VPC와 보안 그룹을 생성합니다. 자세한 내용은 Amazon VPC 설명서 를 참조하세요.	시스템 관리자
AWS DMS 작업을 위한 사용자를 생성합니다.	소스 데이터베이스에서 AWS DMS 사용자를 생성하고 읽기 권한을 부여합니다. 이 사용자는 AWS DMS에서 사용됩니다.	DBA
DB 연결을 테스트합니다.	AWS DMS 사용자로부터 SQL Server DB 인스턴스로의 연결을 테스트합니다.	DBA

작업	설명	필요한 기술
S3 버킷을 생성합니다.	대상 S3 버킷을 생성합니다. 이 버킷에는 마이그레이션된 테이블 데이터가 보관됩니다.	시스템 관리자
IAM 정책 및 역할을 생성합니다.	<ol style="list-style-type: none"> 버킷 권한이 포함된 IAM 정책을 생성하려면 추가 정보 섹션의 코드를 사용합니다. AWS DMS 역할을 생성하고 정책을 이 역할에 연결합니다. 	시스템 관리자

AWS DMS를 이용한 데이터 마이그레이션

작업	설명	필요한 기술
AWS DMS 복제 인스턴스를 생성합니다.	AWS Management Console에 로그인하고 AWS DMS 콘솔을 엽니다. 탐색 창에서 복제 인스턴스, 복제 인스턴스 생성을 선택합니다. 지침은 AWS DMS 설명서의 1단계 를 참조하세요.	DBA
소스 및 대상 DB 엔드포인트를 생성합니다.	소스 및 대상 DB 엔드포인트를 생성합니다. 복제 인스턴스에서 소스 및 대상 엔드포인트로의 연결을 테스트합니다. 지침은 AWS DMS 설명서의 2단계 를 참조하세요.	DBA
복제 작업을 생성합니다.	복제 작업을 생성하고 전체 로드 또는 변경 데이터 캡처(CDC)가 포함된 전체 로드를 선택하여 SQL Server에서 S3 버킷으로 데이터를 마이그레이	DBA

작업	설명	필요한 기술
	선택합니다. 지침은 AWS DMS 설명서의 3단계 를 참조하세요.	
데이터 복제를 시작합니다.	복제 작업을 시작하고 로그에 오류가 있는지 모니터링합니다.	DBA

데이터 유효성 검증

작업	설명	필요한 기술
마이그레이션된 데이터를 검증합니다.	콘솔에서 대상 S3 버킷을 탐색합니다. 소스 데이터베이스와 이름이 같은 하위 폴더를 엽니다. 폴더에 소스 데이터베이스에서 마이그레이션된 모든 테이블이 들어 있는지 확인합니다.	DBA

리소스 정리

작업	설명	필요한 기술
임시 AWS 리소스를 종료하고 삭제합니다.	데이터 마이그레이션을 위해 생성한 임시 AWS 리소스(예: AWS DMS 복제 인스턴스)를 종료하고 내보내기를 검증한 후 삭제합니다.	DBA

관련 리소스

- [AWS Database Migration Service 사용 설명서](#)
- [Microsoft SQL Server 데이터베이스를 AWS DMS 소스로 사용](#)

- [Amazon S3를 AWS Database Migration Service의 대상으로 사용](#)
- [S3 버킷을 AWS DMS 대상으로 사용\(AWS re:Post\)](#)

추가 정보

다음 코드를 사용하여 AWS DMS 역할에 대한 S3 버킷 권한이 있는 IAM 정책을 추가합니다. bucketname을 버킷의 이름으로 바꿉니다.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "s3:PutObject",
        "s3:DeleteObject"
      ],
      "Resource": [
        "arn:aws:s3:::bucketname*"
      ]
    },
    {
      "Effect": "Allow",
      "Action": [
        "s3:ListBucket"
      ],
      "Resource": [
        "arn:aws:s3:::bucketname*"
      ]
    }
  ]
}
```

AWS 개발자 도구를 SageMaker 사용하여 ML, 빌드, 교육 및 워크로드를 Amazon으로 마이그레이션, 배포

작성자: Scot Marvin(AWS)

R 유형: 리플랫폼	소스: 기계 학습	대상: 아마존 SageMaker
작성자: AWS	환경: PoC 또는 파일럿	기술: 기계 학습 및 AI DevOps, 마이그레이션

AWS 서비스: 아마존 SageMaker

요약

이 패턴은 Unix 또는 Linux 서버에서 실행되는 온프레미스 기계 학습 (ML) 애플리케이션을 Amazon을 사용하여 AWS에서 교육 및 배포하도록 마이그레이션하기 위한 지침을 제공합니다. SageMaker 이 배포는 지속적 통합 및 지속적 배포(CI/CD) 파이프라인을 사용합니다. 마이그레이션 패턴은 AWS CloudFormation 스택을 사용하여 배포됩니다.

사전 조건 및 제한 사항

사전 조건

- [AWS 랜딩 존](#)을 사용하는 활성 AWS 계정
- Unix 또는 Linux 서버에 [AWS Command Line Interface\(AWS CLI\)](#)가 설치 및 구성됩니다.
- AWS CodeCommit 또는 아마존 심플 스토리지 서비스 (Amazon S3) 에 있는 ML 소스 코드 리포지토리 GitHub

제한 사항

- 한 AWS 리전에 배포할 수 있는 개별 파이프라인은 300개뿐입니다.
- 이 패턴은 Python으로 train-and-deploy 코드를 사용하는 감독형 ML 워크로드를 위한 것입니다.

제품 버전

- 도커 버전 19.03.5, 빌드 633a0ea, Python 3.6x 사용

아키텍처

소스 기술 스택

- 로컬 파일 시스템 또는 관계형 데이터베이스의 데이터가 있는 온프레미스 Linux 컴퓨팅 인스턴스

소스 아키텍처

대상 기술 스택

- AWS는 데이터 스토리지용으로 Amazon S3와 함께, 파이프라인 실행 추적 또는 로깅을 위한 메타데이터 스토어로 Amazon DynamoDB와 함께 CodePipeline 배포되었습니다.

대상 아키텍처

애플리케이션 마이그레이션 아키텍처

- 기본 Python 패키지 및 AWS CodeCommit 리포지토리 (및 데이터베이스 인스턴스의 온프레미스 데이터 세트용 SQL 클라이언트)

도구

- Python
- Git
- AWS CLI — AWS [CLI는 AWS CloudFormation 스택을 배포하고](#) 데이터를 S3 버킷으로 이동합니다. S3 버킷은 차례로 대상으로 이어집니다.

에픽

마이그레이션 계획

작업	설명	필요한 기술
소스 코드 및 데이터 세트를 검증하십시오.		데이터 사이언티스트
대상 빌드, 교육, 배포 인스턴스 유형과 크기를 식별합니다.		데이터 엔지니어, 데이터 사이언티스트
기능 목록 및 용량 요구 사항을 작성합니다.		
네트워크 요구 사항을 확인합니다.		DBA, 시스템 관리자
소스 및 대상 데이터베이스의 네트워크 또는 호스트 액세스 보안 요구 사항을 확인합니다.		데이터 엔지니어, ML 엔지니어, 시스템 관리자
백업 전략을 결정합니다.		ML 엔지니어, 시스템 관리자
가용성 요구 사항을 결정합니다.		ML 엔지니어, 시스템 관리자
애플리케이션 마이그레이션 또는 전환 전략을 확인합니다.		데이터 사이언티스트, ML 엔지니어

인프라 구성

작업	설명	필요한 기술
Virtual Private Cloud(VPC)를 생성합니다.		ML 엔지니어, 시스템 관리자
보안 그룹 생성.		ML 엔지니어, 시스템 관리자

작업	설명	필요한 기술
ML 코드용 Amazon S3 버킷과 AWS CodeCommit 리포지토리 브랜치를 설정합니다.		ML 엔지니어

데이터 및 코드 업로드

작업	설명	필요한 기술
기본 MySQL 도구 또는 타사 도구를 사용하여 데이터 세트를 프로비저닝된 S3 버킷으로 마이그레이션, 교육, 검증 및 테스트할 수 있습니다.	이는 AWS CloudFormation 스택 배포에 필요합니다.	데이터 엔지니어, ML 엔지니어
ML 트레인과 호스팅 코드를 Python 패키지로 패키징하고 AWS CodeCommit 또는 의 프로비저닝된 리포지토리로 푸시합니다. GitHub	마이그레이션할 AWS CloudFormation 템플릿을 배포하려면 리포지토리의 브랜치 이름이 필요합니다.	데이터 사이언티스트, ML 엔지니어

애플리케이션 마이그레이션

작업	설명	필요한 기술
ML 워크로드 마이그레이션 전략을 따릅니다.		애플리케이션 소유자, ML 엔지니어
AWS CloudFormation 스택을 배포합니다.	AWS CLI를 사용하여 이 솔루션과 함께 제공된 YAML 템플릿에 선언된 스택을 생성합니다.	데이터 사이언티스트, ML 엔지니어

전환

작업	설명	필요한 기술
애플리케이션 클라이언트를 새 인프라로 전환합니다.		애플리케이션 소유자, 데이터 사이언티스트, ML 엔지니어

프로젝트 닫기

작업	설명	필요한 기술
임시 AWS 리소스를 종료합니다.	AWS CloudFormation 템플릿에서 모든 사용자 지정 리소스 (예: 사용되지 않는 모든 AWS Lambda 함수) 를 종료합니다.	데이터 사이언티스트, ML 엔지니어
프로젝트 문서를 검토하고 검증하세요.		애플리케이션 소유자, 데이터 사이언티스트
운영자와 함께 결과 및 ML 모델 평가 지표를 검증합니다.	모델 성능이 애플리케이션 사용자의 기대치와 일치하고 온프레미스 상태와 비슷한지 확인합니다.	애플리케이션 소유자, 데이터 사이언티스트
프로젝트를 마무리하고 피드백을 제공하세요.		애플리케이션 소유자, ML 엔지니어

관련 리소스

- [AWS CodePipeline](#)
- [AWS CodeBuild](#)
- [아마존 SageMaker](#)
- [Amazon S3](#)
- [Amazon DynamoDB](#)
- [AWS Lambda](#)

첨부

이 문서와 관련된 추가 콘텐츠에 액세스하려면 [attachment.zip](#) 파일의 압축을 풉니다.

OpenText TeamSite 워크로드를 AWS 클라우드로 마이그레이션

작성자: Battulga Purevragchaa(AWS), Michael Stewart 및 Carlos Marruenda Molina

환경: 프로덕션	소스: 온프레미스	대상: AWS
R 유형: 리플랫폼	워크로드: 기타 모든 워크로드	기술: 마이그레이션, 웹 및 모바일 앱

AWS 서비스: Amazon EC2,
Amazon RDS

요약

경고: 이 시나리오에서는 IAM 사용자에게 프로그래밍 액세스 권한과 장기 자격 증명이 필요하며, 이로 인해 보안상 위험이 따릅니다. 이 위험을 줄이려면 이러한 사용자에게 작업을 수행하는 데 필요한 권한만 제공하고 더 이상 필요하지 않을 경우 이러한 사용자를 제거하는 것이 좋습니다. 필요한 경우 액세스 키를 업데이트할 수 있습니다. 자세한 내용은 IAM [사용 설명서의 액세스 키 업데이트를](#) 참조하십시오.

많은 [OpenText Experience Platform](#) 인스턴스는 고정 용량 및 레거시 비용 모델을 사용하는 온프레미스 또는 기존 호스팅 솔루션에서 호스팅됩니다. OpenText Experience Platform 워크로드를 Amazon Web Services (AWS) 클라우드로 마이그레이션하면 전체 소유 비용을 줄이는 것 외에도 비즈니스 민첩성과 통합 기회를 높여 추가 기능과 가치를 얻을 수 있습니다.

이 패턴은 [OpenText TeamSite](#) 워크로드를 AWS 클라우드로 마이그레이션하기 위한 단계와 템플릿을 제공합니다. 이 패턴은 마이그레이션 프로세스를 안내하는 상세한 Epic 섹션을 제공하여 마이그레이션 프로젝트의 범위를 정하고 예산을 책정하는 방법을 이해하는 데 도움이 됩니다. OpenText TeamSite

이 패턴은 AWS와 AWS 파트너인 [TBSCG](#)가 개발했으며, AWS Precriptive Guidean 웹 사이트의 [AWS 클라우드로 워크로드 마이그레이션 OpenText TeamSite 및 미디어 관리](#) 가이드와 함께 제공됩니다.

사전 조건 및 제한 사항

사전 조건

- 활성 상태의 AWS 계정 하나 이상
- 온프레미스 데이터 센터 또는 다른 클라우드 공급자에서 호스팅되는 OpenText 워크로드
- 활성 라이선스 OpenText

마이그레이션 프로세스에는 다음 표에 설명된 역할과 책임도 필요합니다.

역할	책임
스폰서	내부 스폰서십
전송 관리자	마이그레이션 전송
솔루션 아키텍트	현재 아키텍처 및 새 아키텍처 정의
DevOps 엔지니어	DevOps 활동
QA 테스터	시스템 수준 테스트
제품 소유자	비즈니스 요구 사항에 따른 작업 우선 순위 지정
TeamSite 저자	마이그레이션 사용자 승인 테스트(UAT)
TeamSite 관리자	마이그레이션 UAT
OpenText 리드	OpenText 제품 전문가
OpenText 개발자	OpenText 제품 전문가
요금 전문가	AWS 및 OpenText 라이선스
IT 보안	보안 기준
타사 통합 개발자	기존 통합 재작업
프론트 엔드 개발자	마이그레이션된 프론트 엔드 코드 변경
데이터베이스 관리자	데이터베이스 구성

제한 사항

- 대상 운영 체제(OS)와의 호환성을 확인합니다. 마이그레이션하려는 제품 버전의 OpenText 제품 릴리스 노트에 있는 호환성 매트릭스를 사용할 수 있습니다.

아키텍처

소스 기술 스택

- OpenText 온프레미스 또는 다른 클라우드 제공업체에서 호스팅되는 고객 경험 솔루션:
 - OpenText TeamSite
 - OpenText LiveSite
 - OpenText 미디어 관리
 - OpenText MediaBin

대상 기술 스택

- AWS 클라우드에서 호스팅되고 다음 AWS 서비스를 사용하는 OpenText 고객 경험 플랫폼:
 - Amazon Elastic Compute Cloud(Amazon EC2)
 - Amazon Elastic Container Service(Amazon ECS)
 - 아마존 OpenSearch 서비스
 - Elastic Load Balancing
 - AWS Lambda
 - Amazon API Gateway
 - Amazon Relational Database Service(Amazon RDS)
 - Amazon Elastic Block Store(Amazon EBS)
 - Amazon Simple Storage Service(S3)

대상 아키텍처

도구

- [AWS Database Migration Service\(AWS DMS\)](#)는 관계형 데이터베이스, 데이터 웨어하우스, NoSQL 데이터베이스 및 기타 유형의 데이터 스토어를 쉽게 마이그레이션할 수 있게 해주는 클라우드 서비스입니다.

- [AWS Application Migration Service](#)는 AWS에서 기본적으로 실행되도록 소스 서버를 자동화합니다. 또한 기본 제공 및 사용자 지정 최적화 옵션을 통해 애플리케이션 현대화를 간소화합니다.

에픽

검색 및 평가

작업	설명	필요한 기술
검색 요구 사항에 대한 워크숍을 개최합니다.	비즈니스 및 기술 팀과 함께 워크숍을 개최하여 현재 환경을 파악하고, 요구 사항을 수집하며, 마이그레이션 전략을 검증합니다. 마이그레이션의 복잡성과 범위에 따라 조직에 여러 워크숍이 필요할 수 있습니다. 소요 시간: 2주	스폰서 (선택 사항), 배송 관리자, 솔루션 아키텍트, OpenText 책임자, 제품 소유자
솔루션 및 마이그레이션 요구 사항을 분석합니다.	계획된 솔루션 및 마이그레이션 프로세스의 설계에 영향을 미치는 비즈니스, 기능 및 기술 요구 사항을 분석하고 문서화합니다. 소요 시간: 1주	솔루션 아키텍트, OpenText 책임자, 제품 소유자
기존 OpenText 아키텍처를 문서화하십시오.	핵심 구성 요소와 모든 관련 애플리케이션 및 서비스를 포함한 기존 OpenText 아키텍처를 문서화하십시오. 소요 시간: 1주	솔루션 아키텍트, OpenText 책임자, 제품 소유자
계획된 AWS 아키텍처를 정의합니다.	식별된 구성 요소, 요구 사항 및 OpenText 호환성 매트릭스를 사용하여 계획된 AWS 아키텍처를 정의하십시오. OpenText 호환성 매트릭스는 사용 중인	솔루션 아키텍트, OpenText 책임자, 제품 소유자, IT 보안

작업	설명	필요한 기술
	<p>OpenText TeamSite 버전의 릴리스 노트에서 찾을 수 있습니다.</p> <p>소요 시간: 1주</p>	
<p>계획한 AWS 아키텍처의 규모를 평가합니다.</p>	<p>크기 요구 사항은 워크로드 및 기타 비기능 요구 사항에 따라 아키텍처 구성 요소마다 다릅니다.</p> <p>소요 시간: 2일</p>	<p>솔루션 아키텍트, OpenText 책임자</p>
<p>TCO를 계산합니다.</p>	<p>제안된 솔루션의 총 소유 비용(TCO)을 계산합니다.</p> <p>소요 시간: 2일</p>	<p>솔루션 아키텍트, 요금 전문가</p>
<p>각 구성 요소에 대한 마이그레이션 전략을 정의합니다.</p>	<p>AWS 클라우드로 마이그레이션해야 하는 각 코어 또는 추가 구성 요소에 사용할 7가지 일반적인 마이그레이션 전략(7R) 중 어떤 것을 정의하고 문서화합니다.</p> <p>소요 시간: 1주</p>	<p>솔루션 아키텍트, OpenText 책임자, 제품 소유자</p>
<p>구성 요소의 마이그레이션 프로세스를 정의합니다.</p>	<p>각 워크로드 구성 요소의 세부 마이그레이션 프로세스를 정의합니다.</p> <p>소요 시간: 1주</p>	<p>솔루션 아키텍트, OpenText 책임자, 제품 소유자, IT 보안</p>

작업	설명	필요한 기술
글로벌 마이그레이션 프로세스와 종속성을 정의합니다.	구성 요소, 종속성, 비즈니스 연속성에 대한 마이그레이션 세부 정보가 포함된 글로벌 마이그레이션 프로세스 및 달력을 생성합니다. 소요 시간: 3일	솔루션 아키텍트, OpenText 책임자, 제품 소유자, IT 보안

보안 및 규정 준수 활동

작업	설명	필요한 기술
보안 정책을 생성합니다.	AWS 계정에서 고객 관리형 보안 정책을 구성합니다. 여기에는 사용하지 않는 계정을 자동으로 비활성화하는 것 외에도 암호 복잡성 및 순환이 포함되어야 합니다. 고객 관리형 정책에 대한 자세한 내용은 AWS Identity and Access Management(IAM) 설명서에서 고객 관리형 정책 을 참조하십시오.	솔루션 아키텍트
두 IAM 사용자를 생성합니다.	AWS Management Console, AWS Command Line Interface (AWS CLI) 및 AWS SDK에 액세스해야 하는 IAM 사용자를 생성합니다. 자세한 내용은 IAM 설명서의 AWS 계정에서 IAM 사용자 생성 을 참조하십시오.	솔루션 아키텍트

작업	설명	필요한 기술
IAM 그룹을 생성합니다.	<p>필요한 IAM 사용자 그룹(예: 관리자 또는 개발자 그룹)을 생성하고 IAM 사용자를 해당 그룹에 추가합니다.</p> <p>IAM 사용자 그룹에 대한 자세한 내용은 IAM 사용 설명서의 IAM 사용자 그룹을 참조하십시오.</p>	솔루션 아키텍트
보안 정책을 연결합니다.	<p>IAM 그룹 또는 역할에 보안 정책을 연결합니다.</p> <p>자세한 내용은 IAM 설명서의 IAM 그룹에 정책 연결을 참조하십시오.</p>	솔루션 아키텍트
세부 결제를 활성화합니다.	<p>결제에 대한 자세한 내용은 AWS 과금 정보 및 비용 관리 설명서에서 사용량 및 비용 모니터링을 참조하십시오.</p>	솔루션 아키텍트
계정의 연락처 세부 정보를 확인하십시오.	<p>계정의 연락처 세부 정보가 최신 상태이고 조직 내 두 명 이상의 개인에게 매핑되는지 확인하십시오.</p> <p>자세한 내용은 AWS 과금 정보 및 비용 관리 설명서에서 AWS 계정 관리를 참조하십시오.</p>	솔루션 아키텍트, 제품 소유자

작업	설명	필요한 기술
<p>보안 연락처 정보를 추가합니다.</p>	<p>보안 연락처 정보로 연락처 정보를 구성하십시오.</p> <p>이에 대한 자세한 내용은 AWS 과금 정보 및 비용 관리 설명서에서 AWS 계정 관리를 참조하십시오.</p>	<p>솔루션 아키텍트, IT 보안</p>
<p>EC2 인스턴스용 IAM 역할을 설정합니다.</p>	<p>EC2 인스턴스의 IAM 역할을 구성합니다.</p> <p>이에 대한 자세한 내용은 Amazon EC2 사용 설명서의 Amazon EC2의 IAM 역할을 참조하십시오.</p>	<p>솔루션 아키텍트</p>
<p>AWS Support에 대한 액세스를 구성합니다.</p>	<p>AWS Support for Support Center에 액세스해야 하는 IAM 사용자에게 IAM 정책을 연결하고 지원 사례를 생성합니다.</p> <p>이에 대한 자세한 내용은 AWS Support 설명서에서 AWS Support에 대한 액세스 권한을 참조하십시오.</p>	<p>솔루션 아키텍트</p>
<p>CloudTrail활성화하세요.</p>	<p>모든 AWS CloudTrail 지역에서 AWS를 자동으로 활성화합니다.</p> <p>이에 대한 자세한 내용은 AWS CloudTrail 설명서의 사용을 create-trail 참조하십시오.</p>	<p>솔루션 아키텍트</p>

작업	설명	필요한 기술
<p>CloudTrail 로그 파일 검증을 활성화합니다.</p>	<p>CloudTrail 로그 파일 검증을 활성화합니다.</p> <p>이에 대한 자세한 내용은 AWS CloudTrail 설명서의 로그 파일 무결성 검증 활성화를 참조하십시오. CloudTrail</p>	<p>솔루션 아키텍트</p>
<p>CloudTrail 로그가 포함된 모든 S3 버킷에 대한 액세스를 제한하십시오.</p>	<p>로그 파일이 포함된 CloudTrail S3 버킷에 대한 액세스를 제한하는 버킷 정책을 적용합니다.</p> <p>이에 대한 자세한 내용은 AWS CloudTrail 설명서의 Amazon S3 버킷 정책을 참조하십시오. CloudTrail</p>	<p>솔루션 아키텍트</p>
<p>CloudTrail CloudWatch 로그와 통합</p>	<p>에서 생성한 트레일을 Amazon CloudTrail CloudWatch Logs 와 통합합니다.</p> <p>이에 대한 자세한 내용은 AWS CloudTrail 설명서의 CloudWatch 로그로 이벤트 전송을 참조하십시오.</p>	<p>솔루션 아키텍트</p>
<p>모든 필수 리전에서 AWS Config를 활성화합니다.</p>	<p>모든 필수 리전에서 AWS Config를 자동으로 활성화합니다.</p> <p>AWS CLI를 사용하여 AWS Config를 설정할 수 있습니다. 자세한 내용은 AWS Config 설명서에서 AWS CLI를 이용한 AWS Config 설정을 참조하십시오.</p>	<p>솔루션 아키텍트</p>

작업	설명	필요한 기술
<p>S3 버킷 액세스 로깅을 활성화합니다.</p>	<p>를 사용하여 S3 버킷 액세스 로깅을 자동화하십시오 CloudTrail.</p> <p>이에 대한 자세한 내용은 Amazon S3 설명서의 S3 버킷 및 객체에 대한 CloudTrail 이벤트 로깅 활성화를 참조하십시오.</p>	<p>솔루션 아키텍트</p>
<p>에 대한 AWS KMS 키 정책을 구성합니다. CloudTrail</p>	<p>에 대한 AWS 키 관리 서비스 (AWS KMS) 키 정책의 구성을 자동화합니다. CloudTrail</p> <p>이에 대한 자세한 내용은 AWS 설명서의 AWS KMS 키 정책 구성을 참조하십시오. CloudTrail CloudTrail</p>	<p>솔루션 아키텍트</p>
<p>저장된 CloudTrail 로그를 암호화하십시오.</p>	<p>AWS KMS에 보관된 고객 관리 키를 사용하여 서버 측 CloudTrail 로그 암호화를 구성합니다.</p> <p>이에 대한 자세한 내용은 AWS 설명서에서 AWS KMS 관리 키 (SSE-KMS) 를 사용한 CloudTrail 로그 파일 암호화를 참조하십시오. CloudTrail</p>	<p>솔루션 아키텍트</p>

작업	설명	필요한 기술
KMS 키를 자동으로 교체합니다.	<p>AWS KMS 키의 순환을 구성합니다.</p> <p>이에 대한 자세한 내용은 AWS KMS 설명서의 자동 키 순환을 활성화 및 비활성화하는 방법을 참조하십시오.</p>	솔루션 아키텍트
CloudWatch 경보를 구성하십시오.	<p>특정 이벤트에 의해 시작되는 Amazon CloudWatch 경보를 구성합니다. API에 대한 무단 요청 또는 루트 계정 사용을 예로 들 수 있습니다.</p> <p>이에 대한 자세한 내용은 AWS Security 블로그에서 AWS 계정의 루트 액세스 키가 사용될 때 알림 수신 방법을 참조하십시오.</p>	솔루션 아키텍트
보안 그룹을 구성합니다.	포트 22와 3389에서 무제한 인바운드 트래픽이 허용되지 않도록 보안 그룹을 구성합니다.	솔루션 아키텍트
VPC 흐름 로깅을 활성화합니다.	<p>가상 사설 클라우드 (VPC) 의 네트워크 인터페이스에서 들어 오고 나가는 거부된 IP 트래픽을 캡처하고 CloudWatch 이를 캡처하도록 구성합니다.</p> <p>이에 대한 자세한 내용은 Amazon VPC 설명서의 흐름 로그 생성을 참조하십시오.</p>	솔루션 아키텍트

작업	설명	필요한 기술
기본 보안 그룹을 수정하여 모든 트래픽을 제한합니다.	트래픽이 기본적으로 거부되고 보안 그룹을 통해 액세스 권한이 명시적으로 부여되도록 각 VPC의 기본 보안 그룹을 수정하십시오. 이에 대한 자세한 내용은 Amazon VPC 설명서의 VPC의 보안 그룹 을 참조하십시오.	솔루션 아키텍트
VPC 간 라우팅 테이블을 구성 하합니다.	최소한의 액세스 권한으로 VPC 피어링에 사용할 라우팅 테이블을 구성합니다. 자세한 내용은 Amazon VPC 설명서의 VPC 피어링 연결을 위한 라우팅 테이블 업데이트 를 참조하십시오.	솔루션 아키텍트

새 AWS 인프라를 위한 설정 활동

작업	설명	필요한 기술
AWS 인프라를 프로비저닝합니다.	AWS 계정과 리소스를 생성합니다. 소요 시간: 2주	DevOps 엔지니어, 솔루션 아키텍트
DevOps 도구 및 프로세스를 설정합니다.	지속적 통합 및 지속적 전달 (CI/CD) 파이프라인, 자동화된 테스트 프레임워크와 같은 DevOps 도구 및 절차를 설정합니다.	DevOps 엔지니어, 솔루션 아키텍트
핵심 구성 요소의 마이그레이션을 자동화합니다.	기존 템플릿 또는 스크립트를 사용하여 TeamSite LiveSite,	DevOps 엔지니어, 솔루션 아키텍트, OpenText 책임자

작업	설명	필요한 기술
	<p>OpenDeploy 및 를 비롯한 OpenText 제품의 설치 및 구성을 자동화합니다 MediaBin.</p> <p>소요 시간: 1주</p>	
<p>추가 구성 요소의 마이그레이션을 자동화합니다.</p>	<p>OpenText 핵심 구성 요소 (예: 추가 데이터베이스, 통신, 모니터링 또는 캐시 구성 요소)와 통합된 추가 애플리케이션의 마이그레이션을 분석하고 자동화합니다.</p> <p>소요 시간: 2주</p>	<p>DevOps 엔지니어, 솔루션 아키텍트, OpenText 책임자</p>
<p>핵심 구성 요소를 조정합니다.</p>	<p>OpenText 핵심 구성 요소의 사용자 지정 (예: 통합)을 필요에 따라 변경합니다.</p>	<p>솔루션스 아키텍트, OpenText 책임자, OpenText 개발자, 타사 통합 개발자, 프론트엔드 개발자</p>
<p>추가 서비스를 구현하고 구성합니다.</p>	<p>AWS Lambda 함수 또는 Amazon API Gateway와 같은 새로운 AWS 서비스를 프로비저닝, 구성 및 구현합니다.</p>	<p>DevOps 엔지니어, 솔루션 아키텍트, 타사 통합 개발자, 프론트엔드 개발자</p>
<p>다른 구성 요소를 마이그레이션하거나 리팩터링할 수 있습니다.</p>	<p>필요한 리팩터링을 포함한 추가 구성 요소를 마이그레이션합니다. 여기에는 맞춤형 보고 포털 또는 기존 API 통합 레이어와 같은 외부 애플리케이션이 포함됩니다.</p>	<p>DevOps 엔지니어, 솔루션스 아키텍트, 타사 통합 개발자, 프론트엔드 개발자</p>
<p>개발 환경에서 마이그레이션을 수행합니다.</p>	<p>시스템 프로비저닝, 데이터 마이그레이션, 애플리케이션 마이그레이션, 설치 및 구성을 포함한 개발 환경을 위한 자동화된 마이그레이션 활동.</p>	<p>DevOps 엔지니어</p>

작업	설명	필요한 기술
프로덕션 환경에서 마이그레이션을 수행합니다.	시스템 프로비저닝, 데이터 마이그레이션, 애플리케이션 마이그레이션, 설치 및 구성을 포함한 프로덕션 환경을 위한 자동화된 마이그레이션 활동.	DevOps 엔지니어

네트워킹 활동

작업	설명	필요한 기술
각 VPC의 CIDR 블록을 정의합니다.	기본이 아닌 각 VPC에 대해 Classless Inter-Domain Routing(CIDR) 블록(IP 범위 및 마스크)을 정의합니다. 기간: 1주 미만	DevOps 엔지니어, 솔루션 아키텍트
서브넷과 가용 영역을 정의합니다.	기본이 아닌 각 VPC에서 사용되는 서브넷과 가용 영역을 정의합니다. 기간: 1주 미만	DevOps 엔지니어, 솔루션 아키텍트
보안 그룹을 정의합니다.	AWS 리소스의 보안을 제어하기 위한 보안 그룹 및 보안 그룹 규칙을 정의합니다. 기간: 1주 미만	DevOps 엔지니어, 솔루션 아키텍트
네트워크 ACL을 정의합니다.	서브넷 경계에서의 보안을 제어하기 위한 네트워크 액세스 제어 목록(ACL)을 정의합니다. 기간: 1주 미만	DevOps 엔지니어, 솔루션 아키텍트

데이터베이스 마이그레이션

작업	설명	필요한 기술
소스 데이터베이스를 준비합니다.	AWS DMS를 사용하여 AWS 클라우드에 지속적으로 복제할 수 있도록 각 소스 데이터베이스를 준비할 수 있습니다.	DevOps 엔지니어, 솔루션 아키텍트
OpenText 핵심 구성 요소에 대한 데이터베이스를 생성합니다.	Opentext TeamSite LiveSite, 및 MediaBin 구성 요소에 필요한 데이터베이스를 생성합니다. OpenText 설치 설명서에 따라 사용자 및 액세스 권한이 올바르게 구성되었는지 확인하십시오.	솔루션 아키텍트, OpenText 책임자, OpenText 개발자
소스 데이터베이스 서버에서 데이터를 복사합니다.	OpenText 핵심 구성 요소의 데이터를 원본 데이터베이스 서버에서 대상 데이터베이스 서버로 복사하는 프로세스를 자동화합니다.	솔루션 아키텍트, OpenText 책임자, OpenText 개발자
데이터베이스 서버의 데이터를 동기화합니다.	소스 데이터베이스에서 대상 데이터베이스로 정기적인 데이터 동기화를 수행하는 프로세스를 자동화합니다.	OpenText 개발자

콘텐츠 마이그레이션 활동

작업	설명	필요한 기술
OpenText TeamSite 콘텐츠 스토어 복사.	원본 OpenText TeamSite 서버에서 대상 OpenText TeamSite 서버로 콘텐츠 저장소를 복사하는 프로세스를 자동화합니다.	솔루션 아키텍트, OpenText 책임자, OpenText 개발자

작업	설명	필요한 기술
사용자와 그룹을 매핑합니다.	내부 OpenText TeamSite 사용자 ID를 대상 시스템 ID에 대한 내부 매핑.	OpenText 리드
OpenText TeamSite 콘텐츠 저장소를 동기화하십시오.	소스 및 대상 콘텐츠 스토어의 정기적인 동기화 수행 프로세스를 자동화합니다. 이는 마이그레이션 및 QA 프로세스의 일부로 구현됩니다.	OpenText 개발자
웹 서버에서 데이터를 복사합니다.	소스 웹 서버에서 대상 웹 서버로 데이터를 복사하는 프로세스를 자동화합니다.	솔루션 아키텍트, OpenText 책임자, OpenText 개발자
웹 서버 데이터를 동기화합니다.	소스 및 대상 웹 서버 데이터를 정기적으로 동기화하는 프로세스를 자동화합니다.	OpenText 개발자
웹 서버 파일 시스템에서 데이터를 복사합니다.	소스 웹 서버 파일 시스템의 콘텐츠 및 기타 웹 자산을 대상 웹 서버로 복사하는 프로세스를 자동화합니다.	솔루션 아키텍트, OpenText 책임자, OpenText 개발자
웹 서버 파일 시스템을 동기화합니다.	소스 웹 서버 파일 시스템의 콘텐츠 및 기타 웹 자산을 대상 웹 서버에 정기적으로 동기화하는 프로세스를 자동화합니다.	OpenText 개발자
피드와 인덱스를 생성합니다.	웹 서버 콘텐츠나 웹 서버 콘텐츠를 데이터 원본으로 사용하는 OpenText TeamSite 피드 또는 기타 색인 (예: 웹 검색) 을 생성하는 프로세스를 자동화합니다.	솔루션스 아키텍트, OpenText 책임자, 개발자 OpenText

작업	설명	필요한 기술
피드 및 인덱스 생성을 동기화합니다.	데이터 동기화 후 피드 및 인덱스를 정기적으로 재생성하는 프로세스를 자동화합니다.	OpenText 개발자

테스트 및 QA 활동

작업	설명	필요한 기술
마이그레이션 QA를 수행합니다.	대상 AWS 환경, 애플리케이션 및 서비스를 테스트하여 자동화된 마이그레이션 프로세스가 올바르게 구축되고 구성되었는지 확인합니다.	DevOps 엔지니어, OpenText 리드, QA 테스터
성능 테스트를 수행합니다.	<p>특정 워크로드의 응답성 및 안정성 측면에서 성능을 테스트합니다. 확장성 및 신뢰성과 같은 대상 시스템의 기타 품질 속성을 조사, 측정, 검증 또는 검증합니다.</p> <p>이 테스트가 유용하려면 프로덕션 환경과 동일한 크기의 테스트 환경이 있어야 합니다.</p> <p>소요 시간: 1~2주</p>	DevOps 엔지니어, OpenText 리드
보안 테스트.	<p>데이터를 보호하고 필요에 따라 기능을 유지하는 애플리케이션의 보안 메커니즘에 잠재적인 결함을 찾아내기 위한 취약성 검사 및 침투 테스트입니다.</p> <p>이 테스트가 유용하려면 네트워크 및 보안 측면에서 프로덕</p>	DevOps 엔지니어, OpenText 리드

작업	설명	필요한 기술
	<p>선 환경과 동일한 테스트 환경이 있어야 합니다.</p> <p>소요 시간: 1~2주</p>	

운영 통합 활동

작업	설명	필요한 기술
운영 준비 상태를 확인합니다.	<p>현재 IT 운영을 어떻게 수행하고 있으며 AWS 클라우드에서 어떻게 운영할 것인지 이해하십시오. 클라우드 운영 모델을 정의하여 이러한 비즈니스 성과를 달성할 수 있습니다.</p> <p>소요 시간: 1주</p>	DevOps 엔지니어, OpenText 리드, 서비스 제공 관리자
운영 자동화에 투자합니다.	<p>자동화에 투자하여 AWS 운영 모델을 제공합니다.</p>	DevOps 엔지니어, OpenText 리드, 서비스 제공 관리자
운영 통합.	<p>기존 IT 도구를 계속 사용하고 AWS 클라우드로의 통합을 통해 이를 확장합니다.</p>	DevOps 엔지니어, OpenText 리드, 서비스 제공 관리자

전환 활동

작업	설명	필요한 기술
DNS를 전환합니다.	<p>도메인 이름 시스템(DNS)을 기존 호스트에서 AWS 클라우드 기반 호스트로 수동 전환합니다.</p> <p>소요 시간: 1시간</p>	DevOps 엔지니어, OpenText 리드

작업	설명	필요한 기술
재해 복구를 테스트합니다.	재해 복구, 백업 복원을 테스트하고 자동화된 테스트를 실행합니다. 소요 시간: 1일	DevOps 엔지니어, OpenText 리드, QA 테스터
모니터링 및 분석을 검증합니다.	모니터링 및 분석이 제대로 작동하는지 확인합니다. 소요 시간: 2시간	DevOps 엔지니어, OpenText 리드
이전 환경을 끄고 서버 종료를 요청하십시오.	소요 시간: 3일	DevOps 엔지니어, OpenText 리드

관련 리소스

- [고객 관리형 정책](#)
- [AWS 계정에서 IAM 사용자 생성](#)
- [IAM 사용자 그룹](#)
- [정책을 IAM 사용자 그룹에 연결](#)
- [사용량 및 비용 모니터링](#)
- [AWS 계정 관리](#)
- [Amazon EC2에 대한 IAM 역할](#)
- [AWS Support에 대한 액세스 권한](#)
- [create-trail 사용](#)
- [에 대한 로그 파일 무결성 검증 활성화 CloudTrail](#)
- [에 대한 Amazon S3 버킷 정책 CloudTrail](#)
- [CloudWatch 로그로 이벤트 전송](#)
- [AWS CLI를 사용하여 AWS Config 설정](#)
- [S3 버킷 및 객체에 대한 CloudTrail 이벤트 로깅 활성화](#)
- [에 대한 AWS KMS 키 정책을 구성합니다. CloudTrail](#)
- [AWS KMS 관리형 키 \(SSE-KMS\) 를 사용하여 CloudTrail 로그 파일 암호화](#)

- [자동 키 교체를 활성화하고 비활성화하는 방법](#)
- [AWS 계정의 루트 액세스 키가 사용될 때 알림 수신 방법](#)
- [흐름 로그 생성](#)
- [VPC의 보안 그룹](#)
- [VPC 피어링 연결을 위한 라우팅 테이블 업데이트](#)

AWS에서 PostgreSQL의 개별 행으로 Oracle CLOB 값을 마이그레이션

작성자: Sai Krishna Namburu(AWS) 및 Sindhusa Paturu(AWS)

환경: PoC 또는 파일럿	소스: Oracle Database	대상: Aurora PostgreSQL-Compatible 또는 Amazon RDS for PostgreSQL
R 유형: 리플랫폼	워크로드: Oracle, 오픈 소스	기술: 마이그레이션, 스토리지 및 백업, 데이터베이스

AWS 서비스: Amazon Aurora, AWS DMS, Amazon S3, Amazon RDS

요약

이 패턴은 Amazon Aurora PostgreSQL 호환 버전과 PostgreSQL용 Amazon Relational Database Service(Amazon RDS)에서 Oracle 캐릭터 라지 오브젝트(CLOB) 값을 개별 행으로 분할하는 방법을 설명합니다. PostgreSQL은 CLOB 데이터 형식을 지원하지 않습니다.

간격 파티션이 있는 테이블은 소스 Oracle 데이터베이스에서 식별되며 테이블 이름, 파티션 유형, 파티션 간격 및 기타 메타데이터가 캡처되어 대상 데이터베이스로 로드됩니다. AWS Database Migration Service(AWS DMS)를 사용하여 크기가 1GB 미만인 CLOB 데이터를 대상 테이블에 텍스트로 로드하거나, 데이터를 CSV 형식으로 내보내고 Amazon Simple Storage Service(S3) 버킷으로 로드한 다음 대상 PostgreSQL 데이터베이스로 마이그레이션할 수 있습니다.

마이그레이션 후에는 이 패턴과 함께 제공되는 사용자 지정 PostgreSQL 코드를 사용하여 새 줄 문자 식별자(CHR(10))를 기준으로 CLOB 데이터를 개별 행으로 분할하고 대상 테이블을 채울 수 있습니다.

사전 조건 및 제한 사항

사전 조건

- 간격 파티션과 CLOB 데이터 유형의 레코드가 있는 Oracle 데이터베이스 테이블.
- 소스 테이블과 유사한 테이블 구조(열 및 데이터 유형이 동일)를 보이는 Aurora PostgreSQL-Compatible 또는 Amazon RDS for PostgreSQL 데이터베이스.

제한 사항

- CLOB 값은 1GB를 초과할 수 없습니다.
- 대상 테이블의 각 행에는 새 줄 문자 식별자가 있어야 합니다.

제품 버전

- Oracle 12c
- Aurora Postgres 11.6

아키텍처

다음 다이어그램은 CLOB 데이터가 포함된 소스 Oracle 테이블과 Aurora PostgreSQL 호환 버전 11.6의 해당 PostgreSQL 테이블을 보여줍니다.

도구

서비스

- [Amazon Aurora PostgreSQL 호환 버전](#)은 PostgreSQL 배포를 설정, 운영 및 확장할 수 있고 ACID를 준수하는 완전관리형 관계형 데이터베이스 엔진입니다.
- [PostgreSQL용 Amazon Relational Database Service\(Amazon RDS\)](#)는 AWS 클라우드에서 PostgreSQL 관계형 데이터베이스(DB)를 설정, 운영 및 조정하는 데 도움이 됩니다.
- [AWS Database Migration Service\(AWS DMS\)](#)를 사용하면 데이터 스토어를 AWS 클라우드로 마이그레이션하거나 클라우드와 온프레미스 설정 간에 데이터 스토어를 마이그레이션할 수 있습니다.
- [Amazon Simple Storage Service\(S3\)](#)는 원하는 양의 데이터를 저장, 보호 및 검색하는 데 도움이 되는 클라우드 기반 객체 스토리지 서비스입니다.

기타 도구

다음 클라이언트 도구를 사용하여 Aurora PostgreSQL-Compatible 및 Amazon RDS for PostgreSQL 데이터베이스에 연결하고, 액세스하고, 관리할 수 있습니다. (이 패턴에서는 이러한 도구가 사용되지 않습니다.)

- [pgAdmin](#)은 PostgreSQL을 위한 오픈 소스 관리 도구입니다. 데이터베이스 객체를 생성, 유지 관리 및 사용하는 데 도움이 되는 그래픽 인터페이스를 제공합니다.

- [DBeaver](#)는 개발자와 데이터베이스 관리자를 위한 오픈 소스 데이터베이스 도구입니다. 이 도구를 사용하여 데이터를 조작, 모니터링, 분석, 관리 및 마이그레이션할 수 있습니다.

모범 사례

Oracle에서 PostgreSQL로 데이터베이스를 마이그레이션하는 모범 사례는 AWS 블로그 게시물 [Oracle 데이터베이스를 Amazon RDS PostgreSQL 또는 Amazon Aurora PostgreSQL로 마이그레이션하는 모범 사례: 마이그레이션 프로세스 및 인프라 고려 사항](#)을 참조하세요.

대용량 바이너리 객체를 마이그레이션하기 위한 AWS DMS 작업을 구성하는 모범 사례는 AWS DMS 설명서의 [대형 바이너리 객체\(LOB\) 마이그레이션](#)을 참조하세요.

에픽

CLOB 데이터 식별하기

작업	설명	필요한 기술
CLOB 데이터를 분석합니다.	<p>소스 Oracle 데이터베이스에서 CLOB 데이터를 분석하여 대상 테이블에 데이터를 로드하는 방법을 결정할 수 있도록 열 헤더가 포함되어 있는지 확인합니다.</p> <p>입력 데이터를 분석하려면 다음 쿼리를 사용합니다.</p> <pre>SELECT * FROM clobdata_or;</pre>	개발자
대상 데이터베이스에 CLOB 데이터를 로드합니다.	CLOB 데이터가 있는 테이블을 Aurora 또는 Amazon RDS 대상 데이터베이스의 중간 (스테이징) 테이블로 마이그레이션합니다. AWS DMS를 사용하거나 Amazon S3 버킷에 데이터를 CSV 파일로 업로드할 수 있습니다.	마이그레이션 엔지니어, DBA

작업	설명	필요한 기술
	<p>이 작업에 AWS DMS를 사용하는 방법에 대한 자세한 내용은 AWS DMS 설명서의 Oracle 데이터베이스를 소스로 사용 및 PostgreSQL 데이터베이스를 대상으로 사용을 참조하세요.</p> <p>이 작업에 Amazon S3를 사용하는 방법에 대한 자세한 내용은 AWS DMS 설명서의 Amazon S3를 대상으로 사용을 참조하세요.</p>	
<p>대상 PostgreSQL 테이블을 검증합니다.</p>	<p>대상 데이터베이스에서 다음 쿼리를 사용하여 소스 데이터와 비교하여 헤더를 포함한 대상 데이터를 검증합니다.</p> <pre data-bbox="594 1014 1029 1213">SELECT * FROM clobdata_ pg; SELECT * FROM clobdatat arget;</pre> <p>결과를 소스 데이터베이스의 쿼리 결과(첫 번째 단계)와 비교합니다.</p>	<p>개발자</p>
<p>CLOB 데이터를 별도의 행으로 분할합니다.</p>	<p>추가 정보 섹션에 제공된 사용자 지정 PostgreSQL 코드를 실행하여 CLOB 데이터를 분할하고 대상 PostgreSQL 테이블의 개별 행에 삽입합니다.</p>	<p>개발자</p>

데이터를 검증합니다.

작업	설명	필요한 기술
대상 테이블의 데이터를 검증합니다.	<p>다음 쿼리를 사용하여 대상 테이블에 삽입된 데이터를 검증합니다.</p> <pre>SELECT * FROM clobdata_pg; SELECT * FROM clobdatatarget;</pre>	개발자

관련 리소스

- [CLOB 데이터 유형](#)(Oracle 설명서)
- [데이터 유형](#)(PostgreSQL 설명서)

추가 정보

CLOB 데이터 분할을 위한 PostgreSQL 함수

```
do
$$
declare
totalstr varchar;
str1 varchar;
str2 varchar;
pos1 integer := 1;
pos2 integer ;
len integer;

begin
    select rawdata||chr(10) into totalstr from clobdata_pg;
    len := length(totalstr) ;
    raise notice 'Total length : %',len;
    raise notice 'totalstr : %',totalstr;
```

```

    raise notice 'Before while loop';

    while pos1 < len loop

        select position (chr(10) in totalstr) into pos2;
        raise notice '1st position of new line : %',pos2;

        str1 := substring (totalstr,pos1,pos2-1);
        raise notice 'str1 : %',str1;

        insert into clobdatatarget(data) values (str1);
        totalstr := substring(totalstr,pos2+1,len);
        raise notice 'new totalstr :%',totalstr;
        len := length(totalstr) ;

    end loop;
end
$$
LANGUAGE 'plpgsql' ;

```

입력 및 출력 예제

데이터를 마이그레이션하기 전에 다음 예제를 사용하여 PostgreSQL 코드를 시험해 볼 수 있습니다.

세 개의 입력 라인이 있는 Oracle 데이터베이스를 생성합니다.

```

CREATE TABLE clobdata_or (
id INTEGER GENERATED ALWAYS AS IDENTITY,
rawdata clob );

insert into clobdata_or(rawdata) values (to_clob('test line 1') || chr(10) ||
to_clob('test line 2') || chr(10) || to_clob('test line 3') || chr(10));
COMMIT;

SELECT * FROM clobdata_or;

```

그러면 다음 출력이 표시됩니다.

id	rawdata
1	테스트 라인 1, 테스트 라인 2, 테스트 라인 3

처리를 위해 PostgreSQL 스테이징 테이블(clobdata_pg)에 소스 데이터를 로드합니다.

```
SELECT * FROM clobdata_pg;

CREATE TEMP TABLE clobdatatarget (id1 SERIAL,data VARCHAR );

<Run the code in the additional information section.>

SELECT * FROM clobdatatarget;
```

그러면 다음 출력이 표시됩니다.

id1	data
1	테스트 라인 1
2	테스트 라인 2
3	테스트 라인 3

데이터베이스 링크를 통한 직접 Oracle 데이터 펌프 가져오기를 사용하여 온프레미스 Oracle 데이터베이스를 Amazon RDS for Oracle로 마이그레이션

작성자: Rizwan Wangde(AWS)

환경: 프로덕션	소스: 온프레미스 오라클 데이터베이스	대상: Amazon RDS for Oracle
R 유형: 리플렛포밍	워크로드: Oracle	기술: 마이그레이션; 데이터베이스

AWS 서비스: AWS DMS, AWS Direct Connect, 아마존 RDS

요약

대규모 Oracle 워크로드를 마이그레이션하는 데 선호되는 방법인 네이티브 Oracle 유틸리티인 Oracle Data Pump를 사용하여 온프레미스 Oracle 데이터베이스를 Amazon RDS for Oracle로 마이그레이션하는 방법은 다양합니다. 이러한 패턴에는 일반적으로 애플리케이션 스키마 또는 테이블을 덤프 파일로 내보내고, 덤프 파일을 Amazon RDS for Oracle의 데이터베이스 디렉터리로 전송한 다음, 덤프 파일에서 애플리케이션 스키마와 데이터를 가져오는 작업이 포함됩니다.

이 접근 방식을 사용하면 데이터 크기와 Amazon RDS 인스턴스로 덤프 파일을 전송하는 데 걸리는 시간에 따라 마이그레이션 시간이 더 오래 걸릴 수 있습니다. 또한 덤프 파일은 Amazon RDS 인스턴스의 Amazon Elastic Block Store(Amazon EBS) 볼륨에 있으며, 이 볼륨은 데이터베이스와 덤프 파일을 저장할 수 있을 만큼 충분히 커야 합니다. 가져온 후 덤프 파일을 삭제하면 빈 스페이스를 검색할 수 없으므로 사용하지 않은 스페이스에 대한 비용을 계속 지불해야 합니다.

이 패턴은 데이터베이스 링크를 통해 Oracle Data Pump API(DBMS_DATAPUMP)를 사용하여 Amazon RDS 인스턴스에서 직접 가져오기를 수행함으로써 이러한 문제를 완화합니다. 이 패턴은 소스 데이터베이스와 대상 데이터베이스 간의 동시 내보내기 및 가져오기 파이프라인을 시작합니다. 이 패턴은 덤프 파일이 생성되거나 볼륨에 저장되지 않으므로 덤프 파일의 EBS 볼륨 크기를 조정할 필요가 없습니다. 이 방법을 사용하면 사용하지 않는 디스크 스페이스의 월별 비용을 절감할 수 있습니다.

사전 조건 및 제한 사항

사전 조건

- 활성 Amazon Web Services(AWS) 계정.
- Amazon RDS 인스턴스에 네트워크 인프라를 제공하기 위해 최소 두 개의 가용 영역에 걸쳐 프라이빗 서브넷으로 구성된 Virtual Private Cloud(VPC).
- 온프레미스 데이터 센터의 Oracle 데이터베이스입니다.
- 단일 가용 영역에 있는 기존 [Amazon RDS Oracle](#) 인스턴스입니다. 단일 가용 영역을 사용하면 마이그레이션 중에 쓰기 성능이 향상됩니다. 전환 24~48시간 전에 다중 AZ 배포를 활성화할 수 있습니다.
- [AWS Direct Connect](#) (대규모 데이터베이스의 경우 권장).
- Amazon RDS 인스턴스에서 온프레미스 Oracle 데이터베이스로의 인바운드 연결을 허용하도록 구성된 온프레미스 네트워크 연결 및 방화벽 규칙.

제한 사항

- Amazon RDS for Oracle의 데이터베이스 크기 제한은 64TiB입니다(2022년 12월 기준).

제품 버전

- 소스 데이터베이스: 오라클 데이터베이스 버전 10g 릴리스 1 이상.
- 대상 데이터베이스: Amazon RDS에서 지원되는 최신 버전 및 에디션 목록은 AWS 설명서에서 [Amazon RDS for Oracle](#)을 참조하십시오.

아키텍처

소스 기술 스택

- 온프레미스 또는 클라우드에 있는 자체 관리형 Oracle 데이터베이스

대상 기술 스택

- Amazon RDS for Oracle

대상 아키텍처

다음 다이어그램은 단일 AZ 환경에서 온프레미스 Oracle 데이터베이스에서 Amazon RDS for Oracle로 마이그레이션하기 위한 아키텍처를 보여줍니다. 화살표 방향은 아키텍처의 데이터 흐름을 나타냅니다. 다이어그램에는 연결을 시작하는 구성 요소가 표시되지 않습니다.

1. Amazon RDS for Oracle은 온프레미스 소스 Oracle 데이터베이스에 연결하여 데이터베이스 링크를 통해 전체 로드 마이그레이션을 수행합니다.
2. AWS DMS는 온프레미스 소스 Oracle 데이터베이스에 연결하여 변경 데이터 캡처(CDC)를 사용하여 지속적인 복제를 수행합니다.
3. CDC 변경 사항은 Amazon RDS for Oracle 데이터베이스에도 적용됩니다.

도구

서비스

- [AWS Database Migration Service\(AWS DMS\)](#)는 데이터 스토어를 AWS 클라우드로 마이그레이션하거나 클라우드와 온프레미스 설정의 조합 간에 마이그레이션하는 데 도움이 됩니다. 이 패턴은 CDC 및 데이터 변경 내용 복제 전용 설정을 사용합니다.
- [AWS Direct Connect](#)는 표준 이더넷 광섬유 케이블을 통해 내부 네트워크를 Direct Connect 위치에 연결합니다. 이 연결을 통해 네트워크 경로의 인터넷 서비스 공급자를 우회하여 퍼블릭 AWS 서비스에 직접 연결되는 가상 인터페이스를 생성할 수 있습니다.
- [Amazon Relational Database Service\(RDS\)](#)는 AWS 클라우드에서 Oracle 관계형 데이터베이스를 설정하고, 운영하고, 규모를 조정하도록 도와줍니다.

기타 도구

- [Oracle Data Pump](#)를 사용하면 한 데이터베이스에서 다른 데이터베이스로 데이터와 메타데이터를 빠른 속도로 이동할 수 있습니다.
- [Oracle Instant Client 또는 SQL Developer와 같은 클라이언트](#) 도구는 데이터베이스에 연결하고 SQL 쿼리를 실행하는 데 사용됩니다.

모범 사례

[AWS Direct Connect](#)는 온프레미스 네트워크와 AWS 간에 전용 프라이빗 네트워크 연결을 사용하지만, 전송 중인 데이터에 대한 추가 보안 및 데이터 암호화를 위해 다음 옵션을 고려해 보십시오.

- [Amazon Site-to-Site VPN을 사용하거나 온프레미스 네트워크에서 AWS 네트워크로 연결되는 IPsec VPN 연결을 사용하는 가상 사설망 \(VPN\)](#)
- [온프레미스 Oracle 데이터베이스에 구성된 Oracle 데이터베이스 네이티브 네트워크 암호화](#)

- [TLS](#)를 사용한 암호화

에픽

온프레미스 소스 Oracle 데이터베이스 준비

작업	설명	필요한 기술
대상 데이터베이스에서 소스 데이터베이스로의 네트워크 연결을 설정합니다.	대상 Amazon RDS 인스턴스에서 온프레미스 소스 Oracle 데이터베이스로 들어오는 연결을 허용하도록 온프레미스 네트워크 및 방화벽을 구성합니다.	네트워크 관리자, 보안 엔지니어
적절한 권한을 가진 데이터베이스 사용자를 생성합니다.	Oracle Data Pump를 사용하여 소스와 대상 간에 데이터를 마이그레이션할 수 있는 권한을 가진 데이터베이스 사용자를 온프레미스 소스 Oracle 데이터베이스에 생성합니다. <pre>GRANT CONNECT to <migration_user>; GRANT DATAPUMP_ EXP_FULL_DATABASE to <migration_user>; GRANT SELECT ANY TABLE to <migration_user>;</pre>	DBA
AWS DMS CDC 마이그레이션을 위한 온프레미스 소스 데이터베이스를 준비합니다.	(선택 사항) Oracle 데이터 펌프 전체 로드를 완료한 후 AWS DMS CDC 마이그레이션을 위해 온프레미스 소스 Oracle 데이터베이스를 준비하십시오. 1. Oracle Data Pump 마이그레이션 중에 플래시백을 관리	DBA

작업	설명	필요한 기술
	<p>하는 데 필요한 추가 권한을 구성하십시오.</p> <pre data-bbox="630 327 1029 604">GRANT FLASHBACK ANY TABLE to <migratio n_user>; GRANT FLASHBACK ARCHIVE ADMINISTER to <migration_user>;</pre> <p>2. AWS DMS용 자체 관리형 Oracle 소스에 필요한 사용자 계정 권한을 구성하려면 AWS DMS 설명서를 참조하십시오.</p> <p>3. AWS DMS를 사용하여 CDC용 Oracle 자체 관리형 소스 데이터베이스를 준비하려면 AWS DMS 설명서를 참조하십시오.</p>	
SQL 개발자를 설치하고 구성합니다.	소스 및 대상 데이터베이스에서 SQL 쿼리를 연결하고 실행하도록 SQL Developer 를 설치하고 구성합니다.	DBA, 마이그레이션 엔지니어

작업	설명	필요한 기술
<p>스크립트를 생성하여 테이블스페이스를 생성합니다.</p>	<p>다음 예제 SQL 쿼리를 사용하여 소스 데이터베이스에서 스크립트를 생성합니다.</p> <pre data-bbox="592 394 1027 949">SELECT 'CREATE TABLESPACE E ' tablespace_name ' DATAFILE SIZE 1G AUTOEXTEND ON MAXSIZE UNLIMITED;' from dba_table spaces where tablespac e_name not in ('SYSTEM' , 'SYSAUX', 'TEMP', 'U NDOTBS1') order by 1;</pre> <p>스크립트는 대상 데이터베이스에 적용됩니다.</p>	DBA
<p>스크립트를 생성하여 사용자, 프로필, 역할 및 권한을 생성합니다.</p>	<p>데이터베이스 사용자, 프로필, 롤 및 권한을 생성하는 스크립트를 생성하려면 Oracle Support 문서 dbms_meta data.get_ddl을 사용하여 권한 및 롤을 포함한 사용자에 대한 DDL을 추출하는 방법 (문서 ID 2739952.1) (오라클 계정 필요)의 스크립트를 사용하십시오.</p> <p>스크립트는 대상 데이터베이스에 적용됩니다.</p>	DBA

대상 Amazon RDS for Oracle 인스턴스를 준비합니다.

작업	설명	필요한 기술
<p>소스 데이터베이스로 연결되는 데이터베이스 링크를 생성하고 연결을 확인합니다.</p>	<p>온프레미스 소스 데이터베이스로 연결되는 데이터베이스 링크를 만들려면 다음 예제 명령을 사용하면 됩니다.</p> <pre data-bbox="594 548 1027 1182">CREATE DATABASE LINK link2src CONNECT TO <migration_user_account> IDENTIFIED BY <password> USING '(DESCRIPTION=(ADDRESS=(PROTOCOL=TCP)(HOST=<dns or ip address of remote db>) (PORT=<listener port>))(CONNECT_DATA=(SID=<remote SID>)))';</pre> <p>연결을 확인하려면 다음 SQL 명령을 실행합니다.</p> <pre data-bbox="594 1346 1027 1461">select * from dual@link2src;</pre> <p>응답이 다음과 같으면 연결에 성공한 X 것입니다.</p>	<p>DBA</p>
<p>스크립트를 실행하여 대상 인스턴스를 준비합니다.</p>	<p>이전에 생성한 스크립트를 실행하여 Amazon RDS for Oracle 인스턴스로 전환할 대상 Amazon RDS for Oracle을 준비합니다.</p>	<p>DBA, 마이그레이션 엔지니어</p>

작업	설명	필요한 기술
	<ol style="list-style-type: none"> 1. 테이블스페이스 2. 프로파일 3. 역할 <p>이렇게 하면 Oracle Data Pump 마이그레이션에서 스키마와 해당 객체를 생성할 수 있습니다.</p>	

데이터베이스 링크를 통해 Oracle Data Pump Import를 사용하여 전체 로드 마이그레이션을 수행합니다.

작업	설명	필요한 기술
필수 스키마를 마이그레이션하십시오.	<p>필요한 스키마를 소스 온프레미스 데이터베이스에서 대상 Amazon RDS 인스턴스로 마이그레이션하려면 추가 정보 섹션의 코드를 사용하십시오.</p> <ul style="list-style-type: none"> • 단일 스키마를 마이그레이션하려면 추가 정보 섹션에서 코드 1을 실행하십시오. • 여러 스키마를 마이그레이션하려면 추가 정보 섹션에서 코드 2를 실행합니다. <p>마이그레이션 성능을 조정하려면 다음 명령을 실행하여 Parallel 프로세스 수를 조정할 수 있습니다.</p>	DBA

작업	설명	필요한 기술
	<pre>DBMS_DATAPUMP.SET_ PARALLEL (handle => v_hdn1, degree => 4);</pre>	
<p>스키마 통계를 수집하여 성능을 향상시킵니다.</p>	<p>스키마 통계 수집 명령은 데이터베이스 개체에 대해 수집된 Oracle 쿼리 최적화 프로그램 통계를 반환합니다. 최적화 프로그램은 이 정보를 사용하여 이러한 개체에 대한 쿼리에 가장 적합한 실행 계획을 선택할 수 있습니다.</p> <pre>EXECUTE DBMS_STAT S.GATHER_SCHEMA_ST ATS(ownname => '<schema_name>');</pre>	<p>DBA</p>

Oracle 데이터 펌프와 AWS DMS를 사용하여 전체 로드 마이그레이션 및 CDC 복제 수행

작업	설명	필요한 기술
<p>소스 온프레미스 Oracle 데이터베이스에서 SCN을 캡처합니다.</p>	<p>소스 온프레미스 Oracle 데이터베이스에서 시스템 변경 번호(SCN)를 캡처합니다. 전체 로드 가져오기에는 SCN을 사용하고 CDC 복제의 시작점으로 사용합니다.</p> <p>소스 데이터베이스에서 현재 SCN을 생성하려면 다음 SQL 문을 실행합니다.</p>	<p>DBA</p>

작업	설명	필요한 기술
	<pre>SELECT current_scn FROM V\$DATABASE;</pre>	

작업	설명	필요한 기술
스키마의 전체 로드 마이그레이션을 수행합니다.	<p>필요한 스키마(FULL LOAD)를 소스 온프레미스 데이터베이스에서 대상 Amazon RDS 인스턴스로 마이그레이션하려면 다음을 수행하십시오.</p> <ul style="list-style-type: none"> • 단일 스키마를 마이그레이션하려면 추가 정보 섹션에서 코드 3을 실행하십시오. • 여러 스키마를 마이그레이션하려면 추가 정보 섹션에서 코드 4를 실행합니다. <p>코드에서 소스 데이터베이스에서 캡처한 <CURRENT_SCN_VALUE_IN_SOURCE_DATABASE> SCN으로 대체합니다.</p> <pre>DBMS_DATAPUMP.SET_PARAMETER (handle => v_hdn1, name => 'FLASHBACK_SCN', value => <CURRENT_SCN_VALUE_IN_SOURCE_DATABASE>);</pre> <p>마이그레이션 성능을 조정하기 위해 Parallel 프로세스 수를 조정할 수 있습니다.</p> <pre>DBMS_DATAPUMP.SET_PARALLEL (handle => v_hdn1, degree => 4);</pre>	DBA

작업	설명	필요한 기술
<p>마이그레이션된 스키마에서 트리거를 비활성화합니다.</p>	<p>AWS DMS CDC 전용 작업을 시작하기 전에 마이그레이션된 스키마에서 TRIGGERS 해당 작업을 비활성화하십시오.</p>	<p>DBA</p>
<p>스키마 통계를 수집하여 성능을 향상시킵니다.</p>	<p>스키마 통계 수집 명령은 데이터베이스 개체에 대해 수집된 Oracle 쿼리 최적화 프로그램 통계를 반환합니다. 최적화 프로그램은 이 정보를 사용하여 이러한 개체에 대한 쿼리에 가장 적합한 실행 계획을 선택할 수 있습니다.</p> <pre data-bbox="597 856 1027 1056">EXECUTE DBMS_STAT S.GATHER_SCHEMA_STATS(ownname => '<schema_name>');</pre>	<p>DBA</p>
<p>AWS DMS를 사용하여 소스에서 대상으로 지속적인 복제를 수행합니다.</p>	<p>AWS DMS를 사용하여 소스 Oracle 데이터베이스에서 대상 Amazon RDS for Oracle 인스턴스로 지속적인 복제를 수행합니다.</p> <p>자세한 내용은 AWS DMS를 사용하여 지속적인 복제를 위한 작업 생성 및 AWS DMS에서 네이티브 CDC 지원을 활용하는 방법에 대한 블로그 게시물을 참조하십시오.</p>	<p>DBA, 마이그레이션 엔지니어</p>

Amazon RDS for Oracle로 전환

작업	설명	필요한 기술
전환 48시간 전에 인스턴스에서 다중 AZ를 활성화하십시오.	프로덕션 인스턴스인 경우 고가용성 (HA) 및 재해 복구 (DR)의 이점을 제공하려면 Amazon RDS 인스턴스에 다중 AZ 배포를 활성화하는 것이 좋습니다.	DBA, 마이그레이션 엔지니어
AWS DMS CDC 전용 작업을 중지합니다 (CDC가 켜져 있는 경우).	<ol style="list-style-type: none"> 1. AWS DMS 작업의 Amazon CloudWatch 지표에서 소스 지연 시간과 대상 지연 시간이 0초로 표시되는지 확인하십시오. 2. AWS DMS CDC 전용 작업을 중지하십시오. 	DBA
트리거를 활성화하십시오.	CDC 작업을 생성하기 전에 비활성화한 트리거를 활성화하십시오.	DBA

관련 리소스

AWS

- [AWS DMS를 사용하여 CDC용 오라클의 자체 관리형 소스 데이터베이스 준비](#)
- [AWS DMS를 사용하여 지속 복제를 위한 작업 생성](#)
- [고가용성을 위한 다중 AZ 배포](#)
- [AWS DMS에서 기본 CDC 지원을 활용하는 방법](#) (블로그 게시물)

오라클 설명서

- [DBMS_데이터펌프](#)

추가 정보

코드 1: 전체 로드 마이그레이션 전용, 단일 애플리케이션 스키마

```

DECLARE
    v_hdn1 NUMBER;
BEGIN
    v_hdn1 := DBMS_DATAPUMP.OPEN(operation => 'IMPORT', job_mode => 'SCHEMA',
remote_link => '<DB LINK Name to Source Database>', job_name => null);
    DBMS_DATAPUMP.ADD_FILE( handle => v_hdn1, filename => 'import_01.log', directory
=> 'DATA_PUMP_DIR', filetype => dbms_datapump.ku$_file_type_log_file);
    DBMS_DATAPUMP.METADATA_FILTER(v_hdn1, 'SCHEMA_EXPR', 'IN (''<schema_name>'')'); --
To migrate one selected schema
    DBMS_DATAPUMP.METADATA_FILTER (hdn1, 'EXCLUDE_PATH_EXPR', 'IN (''STATISTICS'')'); --
To prevent gathering Statistics during the import
    DBMS_DATAPUMP.SET_PARALLEL (handle => v_hdn1, degree => 4); -- Number of parallel
processes performing export and import
    DBMS_DATAPUMP.START_JOB(v_hdn1);
END;
/

```

코드 2: 전체 로드 마이그레이션 전용, 다중 애플리케이션 스키마

```

DECLARE
    v_hdn1 NUMBER;
BEGIN
    v_hdn1 := DBMS_DATAPUMP.OPEN(operation => 'IMPORT', job_mode => 'SCHEMA',
remote_link => '<DB LINK Name to Source Database>', job_name => null);
    DBMS_DATAPUMP.ADD_FILE( handle => v_hdn1, filename => 'import_01.log', directory
=> 'DATA_PUMP_DIR', filetype => dbms_datapump.ku$_file_type_log_file);
    DBMS_DATAPUMP.METADATA_FILTER (v_hdn1, 'SCHEMA_LIST',
''<SCHEMA_1>'', ''<SCHEMA_2>'', ''<SCHEMA_3>''); -- To migrate multiple schemas
    DBMS_DATAPUMP.METADATA_FILTER (v_hdn1, 'EXCLUDE_PATH_EXPR', 'IN (''STATISTICS'')');
-- To prevent gathering Statistics during the import
    DBMS_DATAPUMP.SET_PARALLEL (handle => v_hdn1, degree => 4); -- Number of parallel
processes performing export and import
    DBMS_DATAPUMP.START_JOB(v_hdn1);
END;
/

```

코드 3: CDC 전용 작업 이전의 전체 로드 마이그레이션, 단일 애플리케이션 스키마

```

DECLARE

```

```

v_hdn1 NUMBER;
BEGIN
  v_hdn1 := DBMS_DATAPUMP.OPEN(operation => 'IMPORT', job_mode => 'SCHEMA',
remote_link => '<DB LINK Name to Source Database>', job_name => null);
  DBMS_DATAPUMP.ADD_FILE( handle => v_hdn1, filename => 'import_01.log', directory
=> 'DATA_PUMP_DIR', filetype => dbms_datapump.ku$_file_type_log_file);
  DBMS_DATAPUMP.METADATA_FILTER(v_hdn1, 'SCHEMA_EXPR', 'IN ('<schema_name>')'); --
To migrate one selected schema
  DBMS_DATAPUMP.METADATA_FILTER (v_hdn1, 'EXCLUDE_PATH_EXPR', 'IN (''STATISTICS'')');
-- To prevent gathering Statistics during the import
  DBMS_DATAPUMP.SET_PARAMETER (handle => v_hdn1, name => 'FLASHBACK_SCN', value =>
<CURRENT_SCN_VALUE_IN_SOURCE_DATABASE>); -- SCN required for AWS DMS CDC only task.
  DBMS_DATAPUMP.SET_PARALLEL (handle => v_hdn1, degree => 4); -- Number of parallel
processes performing export and import
  DBMS_DATAPUMP.START_JOB(v_hdn1);
END;
/

```

코드 4: CDC 전용 작업 이전의 전체 로드 마이그레이션, 다중 애플리케이션 스키마

```

DECLARE
  v_hdn1 NUMBER;
BEGIN
  v_hdn1 := DBMS_DATAPUMP.OPEN (operation => 'IMPORT', job_mode => 'SCHEMA',
remote_link => '<DB LINK Name to Source Database>', job_name => null);
  DBMS_DATAPUMP.ADD_FILE (handle => v_hdn1, filename => 'import_01.log', directory
=> 'DATA_PUMP_DIR', filetype => dbms_datapump.ku$_file_type_log_file);
  DBMS_DATAPUMP.METADATA_FILTER (v_hdn1, 'SCHEMA_LIST',
''<SCHEMA_1>', ''<SCHEMA_2>', ''<SCHEMA_3>''); -- To migrate multiple schemas
  DBMS_DATAPUMP.METADATA_FILTER (v_hdn1, 'EXCLUDE_PATH_EXPR', 'IN (''STATISTICS'')');
-- To prevent gathering Statistics during the import
  DBMS_DATAPUMP.SET_PARAMETER (handle => v_hdn1, name => 'FLASHBACK_SCN', value =>
<CURRENT_SCN_VALUE_IN_SOURCE_DATABASE>); -- SCN required for AWS DMS CDC only task.
  DBMS_DATAPUMP.SET_PARALLEL (handle => v_hdn1, degree => 4); -- Number of parallel
processes performing export and import
  DBMS_DATAPUMP.START_JOB(v_hdn1);
END;
/

```

혼합 마이그레이션 접근 방식이 더 효과적일 수 있는 시나리오

드문 경우지만 소스 데이터베이스에 수백만 개의 행과 매우 큰 LOBSEGMENT 열로 구성된 테이블이 포함되어 있는 경우 이 패턴으로 인해 마이그레이션 속도가 느려질 수 있습니다. Oracle은 네트워

크 링크를 통해 LOBSegment를 한 번에 하나씩 마이그레이션합니다. 소스 테이블에서 단일 행(LOB 열 데이터 포함)을 추출하여 대상 테이블에 행을 삽입하고 모든 행이 마이그레이션될 때까지 프로세스를 반복합니다. 데이터베이스 링크를 통한 Oracle Data Pump는 LOBSegment에 대한 대량 로드나 직접 경로 로드 메커니즘을 지원하지 않습니다.

이 상황에서는 다음을 권장합니다.

- 다음 메타데이터 필터를 추가하여 Oracle Data Pump 마이그레이션 중에 식별된 테이블을 건너뛰십시오.

```
dbms_datapump.metadata_filter(handle =>h1, name=>'NAME_EXPR', value => 'NOT IN ('TABLE_1','TABLE_2'))');
```

- AWS DMS 작업(전체 로드 마이그레이션, 필요한 경우 CDC 복제 포함)을 사용하여 식별된 테이블을 마이그레이션합니다. AWS DMS는 소스 Oracle 데이터베이스에서 여러 행을 추출하여 대상 Amazon RDS 인스턴스에 일괄적으로 삽입하므로 성능이 향상됩니다.

Oracle E-Business Suite를 Amazon RDS Custom으로 마이그레이션

작성자: Simon Cunningham(AWS), Jaydeep Nandy(AWS), Nitin Saxena(AWS), Vishnu Vinnakota(AWS)

환경: 프로덕션	소스: Amazon EC2 또는 온프레미스	대상: Amazon RDS Custom
R 유형: 리플랫폼	워크로드: Oracle	기술: 마이그레이션, 데이터베이스, 인프라

AWS 서비스: Amazon EFS, Amazon RDS, AWS Secrets Manager

요약

Oracle E-Business Suite는 재무, 인사, 공급망, 제조 등 전사적 프로세스를 자동화하기 위한 전사적 자원 계획(ERP) 솔루션입니다. 클라이언트, 애플리케이션, 데이터베이스의 3계층 아키텍처를 갖추고 있습니다. 이전에는 자체 관리형 [Amazon Elastic Compute Cloud\(Amazon EC2\) 인스턴스](#)에서 Oracle E-Business Suite 데이터베이스를 실행해야 했지만 이제는 [Amazon Relational Database Service\(Amazon RDS\) Custom](#)을 활용할 수 있습니다.

[Amazon RDS Custom for Oracle](#)은 기본 운영 체제 및 데이터베이스 환경에 액세스해야 하는 레거시, 커스텀 및 패키지 애플리케이션을 위한 관리형 데이터베이스 서비스입니다. 데이터베이스 관리 작업 및 운영을 자동화하고 데이터베이스 관리자가 데이터베이스 환경 및 운영 체제에 액세스하고 사용자 정의할 수 있도록 합니다. Oracle 데이터베이스를 Amazon RDS Custom으로 마이그레이션하면 Amazon Web Services(AWS)가 백업 작업과 같은 무거운 작업을 처리하고고가용성을 보장하며, 사용자는 Oracle E-Business Suite 애플리케이션 및 기능을 유지 관리하는 데 집중할 수 있습니다. 마이그레이션에 대해 고려해야 할 주요 요소는 AWS 권장 가이드의 [Oracle 데이터베이스 마이그레이션 전략](#)을 참조하십시오.

이 패턴은 Oracle 복구 관리자(RMAN) 백업과 EC2 인스턴스와 Amazon RDS Custom 간의 [Amazon Elastic File System\(Amazon EFS\)](#) 공유 파일 시스템을 사용하여 Amazon EC2의 독립형 Oracle 데이터베이스를 Amazon RDS Custom으로 마이그레이션하는 단계를 중점적으로 다룹니다. 이 패턴은 RMAN 전체 백업(레벨 0 백업이라고도 함)을 사용합니다. 단순화를 위해 애플리케이션을 종료하고

데이터베이스가 마운트된 상태로 열려 있지 않은 콜드 백업을 사용합니다. (Oracle Data Guard 또는 RMAN 복제를 사용하여 백업할 수도 있습니다. 하지만 이 패턴에는 이러한 옵션이 포함되지 않습니다.)

고가용성 및 재해 복구를 위해 AWS에서 Oracle E-Business Suite를 설계하는 방법에 대한 자세한 내용은 [활성 대기 데이터베이스를 사용하여 Amazon RDS Custom에서 Oracle E-Business Suite를 위한 HA/DR 아키텍처 설정](#) 패턴을 참조하세요.

참고: 이 패턴은 Oracle 지원 정보로 연결되는 링크를 제공합니다. 이러한 문서에 액세스하려면 [Oracle Support](#) 계정이 필요합니다.

사전 조건 및 제한 사항

사전 조건

- Oracle Linux 7 또는 Red Hat Enterprise Linux(RHEL) 버전 7.x가 설치된 Amazon EC2에서 실행되는 Oracle 버전 12.1.0.2 또는 19c(최소 19.3) 소스 데이터베이스. 이 패턴은 소스 데이터베이스 이름이 VIS이고 Oracle 19c의 추가 컨테이너 데이터베이스 이름이 VISCD인 것으로 가정하지만 다른 이름을 사용할 수 있습니다.

참고: 온프레미스 네트워크와 [Amazon Virtual Private Cloud\(VPC\)](#) 간에 적절한 네트워크 연결이 있는 경우 온프레미스 Oracle 소스 데이터베이스에서도 이 패턴을 사용할 수 있습니다.

- Oracle E-Business Suite 버전 12.2.x 애플리케이션(비전 인스턴스). 이 절차는 버전 12.2.11에서 테스트되었습니다.
- 단일 Oracle E-Business Suite 애플리케이션 계층. 그러나 이 패턴을 여러 애플리케이션 계층에서 작동하도록 조정할 수 있습니다.
- Oracle 12.1.0.2의 경우 Amazon RDS Custom은 최소 16GB의 스왑 공간으로 구성되었습니다. 그렇지 않으면 12c 예제 CD에 경고가 표시됩니다. (이 문서의 뒷부분에서 설명하는 것처럼 Oracle 19c에는 예제 CD가 필요하지 않습니다.)

마이그레이션을 시작하기 전에 다음 단계를 완료해야 합니다.

- Amazon RDS 콘솔에서 데이터베이스 이름 VIS(또는 소스 데이터베이스 이름)를 사용하여 Oracle DB용 Amazon RDS Custom 인스턴스를 생성합니다. 지침은 AWS 설명서의 [Amazon RDS Custom 사용 및 Oracle용 Amazon RDS Custom - 데이터베이스 환경의 새로운 제어 기능](#) 블로그 게시물을 참조하십시오. 이렇게 하면 데이터베이스 이름이 소스 데이터베이스와 동일한 이름으로 설정됩니다. (비워 두면 EC2 인스턴스 및 데이터베이스 이름이 ORCL로 설정됩니다.) 최소한 소스에 적용된 패치를 사용하여 [사용자 지정 엔진 버전\(CEV\)](#)을 생성해야 합니다. 자세한 내용은 Amazon RDS 설명서의 [CEV 생성 준비](#)를 참조하세요.

Oracle 19c에 대한 참고 사항: 현재 Oracle 19c의 경우 Amazon RDS 컨테이너 데이터베이스 이름을 사용자 지정할 수 있습니다. 기본값은 RDSCDB입니다. 소스 EC2 인스턴스와 동일한 시스템 ID(SID)를 사용하여 RDS Custom Oracle 인스턴스를 생성해야 합니다. 예를 들어 이 패턴에서는 Oracle 19c SID가 소스 인스턴스에 VISCDDB로 있는 것으로 가정합니다. 따라서 Amazon RDS Custom 상의 대상 Oracle 19c SID도 VISCDDB여야 합니다.

2. Amazon EC2 소스 데이터베이스와 일치하도록 충분한 스토리지, vCPU 및 메모리를 갖춘 Amazon RDS Custom DB 인스턴스를 구성합니다. 이를 위해 vCPU 및 메모리를 기반으로 [Amazon EC2 인스턴스 유형](#)을 일치시킬 수 있습니다.
3. Amazon EFS 파일 시스템을 생성하고 Amazon EC2 및 Amazon RDS Custom 인스턴스에 탑재합니다. 자세한 지침은 [Oracle용 Amazon RDS Custom과 Amazon EFS 통합](#) 블로그 게시물을 참조하세요. 이 패턴은 소스 Amazon EC2와 대상 Amazon RDS Custom DB 인스턴스 모두의 /RMAN에 Amazon EFS 볼륨을 탑재했으며 소스와 대상 간에 네트워크 연결이 가능하다고 가정합니다. [Amazon FSx](#) 또는 다른 공유 드라이브를 사용하여 동일한 방법을 사용할 수도 있습니다.

가정

이 패턴은 애플리케이션과 데이터베이스가 논리적 호스트 이름을 사용한다고 가정하므로 마이그레이션 단계 수가 줄어듭니다. 물리적 호스트 이름을 사용하도록 이러한 단계를 조정할 수 있지만 논리적 호스트 이름을 사용하면 마이그레이션 프로세스의 복잡성이 줄어듭니다. 논리적 호스트 이름 사용의 이점에 대한 자세한 내용은 다음 지원 노트를 참조하세요.

- 12c의 경우 Oracle Support Note 2246690.1
- 19c의 경우 Oracle Support Note 2617788.1

이 패턴은 Oracle 12c에서 19c로 업그레이드하는 시나리오를 다루지 않으며, Amazon EC2에서 실행되는 동일한 버전의 Oracle 데이터베이스를 Oracle용 Amazon RDS Custom으로 마이그레이션하는 데 중점을 둡니다.

Oracle용 Amazon RDS Custom은 [Oracle Home 사용자 정의를 지원합니다](#). (Oracle Home은 Oracle 바이너리를 저장합니다.) /rdsdbbin/oracle 기본 경로를 지정한 경로(예: /d01/oracle/VIS/19c)로 변경할 수 있습니다. 단순화를 위해 이 패턴의 지침에서는 /rdsdbbin/oracle 기본 경로를 가정합니다.

제한 사항

이 패턴은 다음 기능 및 구성을 지원하지 않습니다.

- 데이터베이스 ARCHIVE_LAG_TARGET 파라미터를 60~7200 범위를 벗어난 값으로 설정
- DB 인스턴스 로그 모드 비활성화 (NOARCHIVELOG)
- EC2 인스턴스의 EBS-optimized 속성 끄기
- EC2 인스턴스에 연결된 원본 Amazon Elastic Block Store(Amazon EBS) 볼륨 수정
- 새 EBS 볼륨 추가 또는 볼륨 유형을 gp2에서 gp3로 변경
- TNS ifile 지원
- control_file 위치 및 이름 변경(VISDCDB가 CDB 이름인 경우 반드시 /rdsdbdata/db/VISDCDB_A/controlfile/control-01.ct1이어야 함)

이러한 구성 및 기타 지원되지 않는 구성에 대한 추가 정보는 Amazon RDS 설명서의 [지원되지 않는 구성 수정](#)을 참조하세요.

제품 버전

Amazon RDS Custom에서 지원하는 Oracle Database 버전 및 인스턴스 클래스에 대한 자세한 내용은 [Oracle용 Amazon RDS Custom의 가용성 및 요구 사항](#)을 참조하세요.

아키텍처

다음 아키텍처 다이어그램은 AWS의 단일 [가용 영역](#)에서 실행되는 Oracle E-Business Suite 시스템을 나타냅니다. 애플리케이션 계층은 [Application Load Balancer](#)를 통해 액세스되며, 애플리케이션과 데이터베이스는 모두 프라이빗 서브넷에 있고, Amazon RDS Custom 및 Amazon EC2 데이터베이스 계층은 Amazon EFS 공유 파일 시스템을 사용하여 RMAN 백업 파일을 저장하고 액세스합니다.

도구

서비스

- [Amazon RDS Custom for Oracle](#)은 기본 운영 체제 및 데이터베이스 환경에 액세스해야 하는 레거시, 커스텀 및 패키지 애플리케이션을 위한 관리형 데이터베이스 서비스입니다. 데이터베이스 관리 작업 및 운영을 자동화하고 데이터베이스 관리자가 데이터베이스 환경 및 운영 체제에 액세스하고 사용자 정의할 수 있도록 합니다.
- [Amazon Elastic File System\(Amazon EFS\)](#)은 관리나 프로비저닝이 필요 없이 파일을 추가하고 제거할 수 있는 간단하고 서버리스이며 탄력적인 파일 시스템입니다. 이 패턴은 Amazon EFS 공유 파일 시스템을 사용하여 RMAN 백업 파일을 저장하고 액세스합니다.

- [AWS Secrets Manager](#)는 데이터베이스 보안 인증, API 키, 기타 보안 암호를 손쉽게 교체, 관리, 검색할 수 있게 도와주는 AWS 관리형 서비스입니다. Amazon RDS Custom은 데이터베이스 생성 시 키 쌍과 데이터베이스 사용자 보안 인증 정보를 Secrets Manager에 저장합니다. 이 패턴에서는 Secrets Manager에서 데이터베이스 사용자 암호를 검색하여 RDSADMIN 및 ADMIN 사용자를 생성하고 시스템 및 시스템 암호를 변경합니다.

기타 도구

- RMAN은 Oracle 데이터베이스에 대한 백업 및 복구 지원을 제공하는 도구입니다. 이 패턴은 RMAN을 사용하여 Amazon RDS Custom에서 복원된 Amazon EC2의 소스 Oracle 데이터베이스에 대한 콜드 백업을 수행합니다.

모범 사례

- 논리적 호스트 이름을 사용합니다. 이렇게 하면 실행해야 하는 사후 복제 스크립트의 수가 크게 줄어듭니다. 자세한 내용은 Oracle Support 문서 2246690.1을 참조하세요.
- Amazon RDS Custom은 기본적으로 Oracle [자동 메모리 관리](#)(AMM)를 사용합니다. Hugesmem 커널을 사용하려는 경우, 자동 공유 메모리 관리(ASMM)를 대신 사용하도록 Amazon RDS Custom을 구성할 수 있습니다.
- `memory_max_target` 파라미터는 기본적으로 활성화되어 있습니다. 프레임워크는 백그라운드에서 이 파라미터를 사용하여 읽기 전용 복제본을 생성합니다.
- Oracle Flashback Database를 활성화합니다. 이 기능은 대기를 복원하기 위한 장애 조치(전환 아님) 테스트 시나리오에서 유용합니다.
- 데이터베이스 초기화 파라미터의 경우 Oracle 소스 데이터베이스의 SPFILE을 사용하는 대신 Oracle E-Business Suite용 Amazon RDS Custom DB 인스턴스에서 제공하는 표준 PFILE을 사용자 지정합니다. Amazon RDS Custom에서 읽기 전용 복제본을 생성할 때 스페이스와 주석으로 인해 문제가 발생하기 때문입니다. 데이터베이스 초기화 파라미터에 대한 자세한 내용은 Oracle Support 문서 396009.1을 참조하세요.

다음 에픽 섹션에서는 Oracle 12.1.0.2 및 19c에 대한 세부 정보가 다른 별도의 지침을 제공합니다.

에픽

소스 애플리케이션 종료

작업	설명	필요한 기술
애플리케이션을 종료합니다.	<p>소스 애플리케이션을 종료하려면 다음 명령을 사용합니다.</p> <pre>\$ su - applmgr \$ cd \$INST_TOP/admin/scripts \$./adstpall.sh</pre>	DBA
.zip 파일을 만듭니다.	<p>소스 애플리케이션 계층에서 appsutil.zip 파일을 생성합니다. 나중에 이 파일을 사용하여 Amazon RDS Custom 데이터베이스 노드를 구성합니다.</p> <pre>\$ perl \$AD_TOP/bin/admkappsutil.pl</pre>	DBA
.zip 파일을 Amazon EFS로 복사합니다.	<p>\$INST_TOP/admin/out 에서 appsutil.zip 을 공유 Amazon EFS 볼륨(/RMAN/appsutil)으로 복사합니다. 보안 복사(SCP) 또는 다른 전송 메커니즘을 사용하여 파일을 수동으로 전송할 수 있습니다.</p>	DBA

소스 데이터베이스 사전 복제

작업	설명	필요한 기술
Amazon EC2에서 데이터베이스 계층을 사전 복제합니다.	<p>Oracle 사용자로 로그인하고 다음을 실행합니다.</p> <pre>\$ cd \$ORACLE_HOME/appsu til/scripts/\$CONTE XT_NAME \$ perl adpreclone.pl dbTier</pre> <p>생성된 로그 파일을 확인하여 작업이 성공적으로 완료되었는지 확인합니다.</p>	DBA
appsutil.zip 파일을 Amazon EFS 파일 시스템에 복사합니다.	<p>tar 백업을 생성하고 공유 Amazon EFS 파일 시스템에 \$ORACLE_HOME/appsutil 을 복사합니다 (예: /RMAN/appsutil).</p> <pre>\$ cd \$ORACLE_HOME \$ tar cvf sourceapp sutil.tar appsutil \$ cp sourceapp sutil.tar /RMAN/app sutil</pre>	DBA

소스 Amazon EC2 데이터베이스의 콜드 RMAN 전체 백업 수행

작업	설명	필요한 기술
백업 스크립트를 생성합니다.	공유 Amazon EFS 파일 시스템에 소스 데이터베이스의 RMAN 전체 백업을 수행합니다.	DBA

작업	설명	필요한 기술
	<p>단순화를 위해 이 패턴은 콜드 RMAN 백업을 수행합니다. 하지만 Oracle Data Guard를 사용하여 핫 RMAN 백업을 수행하도록 수정하여 가동 중지 시간을 줄일 수 있습니다.</p> <p>1. 탑재 모드에서 소스 Amazon EC2 데이터베이스를 시작합니다.</p> <pre data-bbox="597 695 1027 894"> \$ sqlplus / as sysdba \$ SQL> shutdown immediate \$ SQL> startup mount </pre> <p>2. RMAN 백업 스크립트(사용 중인 Oracle 버전에 따라 다음 예 중 하나를 사용하거나 기존 RMAN 스크립트 중 하나를 실행)를 생성하여 탑재한 Amazon EFS 파일 시스템에 데이터베이스를 백업합니다(이 예에서는 /RMAN).</p> <p>Oracle 12.1.0.2의 경우:</p> <pre data-bbox="597 1419 1027 1871"> \$ vi FullRMANColdBackup .sh #!/bin/bash . /home/oracle/.bash _profile export ORACLE_SID=VIS export ORACLE_HOME=/ d01/oracle/VIS/12.1.0 export DATE=\$(date + %y-%m-%d_%H%M%S) </pre>	

작업	설명	필요한 기술
	<pre> rman target / log=/RMAN /VISDB_\${DATE}.log << EOF run { allocate channel ch1 device type disk format '/RMAN/visdb_full_ bkp_%u'; allocate channel ch2 device type disk format '/RMAN/visdb_full_ bkp_%u'; crosscheck backup; delete noprompt obsolete; BACKUP AS COMPRESSED BACKUPSET DATABASE PLUS ARCHIVELOG; backup archivelog all; release channel ch1; release channel ch2; } EOF </pre> <p>Oracle 19c의 경우:</p> <pre> \$ vi FullRMANColdBackup .sh #!/bin/bash . /home/oracle/.bash _profile export ORACLE_SI D=VISDCDB export ORACLE_HOME=/ d01/oracle/VIS/19c export DATE=\$(date + %y-%m-%d_%H%M%S) </pre>	

작업	설명	필요한 기술
	<pre> rman target / log=/RMAN /VISDB_\$(DATE).log << EOF run { allocate channel ch1 device type disk format '/RMAN/visdb_full_ bkp_%.u'; allocate channel ch2 device type disk format '/RMAN/visdb_full_ bkp_%.u'; crosscheck backup; delete noprompt obsolete; BACKUP AS COMPRESSED BACKUPSET DATABASE PLUS ARCHIVELOG; backup archivelog all; backup current controlfile format '/ RMAN/cntrl.bak'; release channel ch1; release channel ch2; } EOF </pre>	
<p>백업 스크립트를 실행합니다.</p>	<p>권한을 변경하고 Oracle 사용자로 로그인한 다음 스크립트를 실행합니다.</p> <pre> \$ chmod 755 FullRMANColdBackup.sh \$./FullRMANColdBackup.sh </pre>	<p>DBA</p>

작업	설명	필요한 기술
<p>오류를 확인하고 백업 파일의 이름을 기록해 둡니다.</p>	<p> RMAN 로그 파일에서 오류를 확인합니다. 모든 것이 정상인 것 같으면 제어 파일의 백업을 나열합니다. 출력 파일의 경로 및 이름을 기록합니다. </p> <p>Oracle 12.1.0.2의 경우:</p> <pre> RMAN> connect target / RMAN> list backup of controlfile; BS Key Type LV Size Device Type Elapsed Time Completion Time ----- ----- ----- 9 Full 1.11M DISK 00:00:04 23-APR-22 BP Key: 9 Status: AVAILABLE Compressed: YES Tag: TAG20220423T121011 Piece Name: / RMAN/visdb_full_b kp_100rlsbt Control File Included: Ckp SCN: 122045953 96727 Ckp time: 23- APR-22 </pre> <p> 나중에 Amazon RDS Custom에서 데이터베이스를 복원할 때 /RMAN/visdb_full_b </p>	<p>DBA</p>

작업	설명	필요한 기술
	<p>kp_100rlsbt 백업 파일을 사용하게 됩니다.</p> <p>Oracle 19c의 경우:</p> <pre> RMAN> connect target / RMAN> list backup of controlfile; BS Key Type LV Size Device Type Elapsed Time Completion Time ----- ----- ----- 38 Full 17.92M DISK 00:00:01 25-NOV-22 BP Key: 38 Status: AVAILABLE Compressed: NO Tag: TAG20221125T095014 Piece Name: / RMAN/cntrl.bak Control File Included: Ckp SCN: 122046201 88873 Ckp time: 23- NOV-22 </pre> <p>나중에 Amazon RDS Custom에서 데이터베이스를 복원할 때 /RMAN/cntrl.bak 백업 파일을 사용하게 됩니다.</p>	

대상 Amazon RDS Custom 데이터베이스 구성

작업	설명	필요한 기술
<p>호스트 파일을 변경하고 호스트 이름을 설정합니다.</p>	<p>참고: 이 섹션의 명령은 루트 사용자로 실행해야 합니다.</p> <p>1. Amazon RDS Custom DB 인스턴스에서 <code>/etc/hosts</code> 파일을 편집합니다. 이를 수행하는 간단한 방법은 소스 Amazon EC2 데이터베이스 호스트 파일에서 데이터베이스 및 애플리케이션 호스트 항목을 복사하는 것입니다.</p> <pre data-bbox="594 863 1027 1262"> <IP-address> OEBS-app01.localdomain OEBS-app01 OEBS-app01log.localdomain OEBS-app01log <IP-address> OEBS-db01.localdomain OEBS-db01 OEBS-db01log.localdomain OEBS-db01log </pre> <p>데이터베이스 노드 IP 주소가 <code><IP-address></code> 인 경우 이 주소를 Amazon RDS Custom IP 주소로 바꿔야 합니다. 논리적 호스트 이름에는 <code>*log</code>가 추가됩니다.</p> <p>2. <code>hostnamectl</code> 명령을 실행하여 데이터베이스 호스트 이름을 변경합니다.</p>	DBA

작업	설명	필요한 기술
	<pre data-bbox="594 212 1027 367">\$ sudo hostnamectl set-hostname --static persistent-hostname</pre> <p data-bbox="594 405 634 436">예:</p> <pre data-bbox="594 478 1027 634">\$ sudo hostnamectl set- hostname --static OEBS- db01log</pre> <p data-bbox="594 674 1000 804">자세한 내용은 정적 호스트 이름 할당에 관한 Knowledge Center 문서를 참조하세요.</p> <p data-bbox="594 852 1000 1073">3. Amazon RDS Custom DB 인스턴스를 다시 시작합니다. 이후 단계에서 데이터베이스가 삭제되므로 데이터베이스 종료는 걱정하지 마세요.</p> <pre data-bbox="594 1115 1027 1192">\$ reboot</pre> <p data-bbox="594 1234 1000 1404">4. Amazon RDS Custom DB 인스턴스가 다시 가동되면 로그인하여 호스트 이름이 변경되었는지 확인합니다.</p> <pre data-bbox="594 1446 1027 1564">\$ hostname oebs-db01</pre>	

작업	설명	필요한 기술
<p>Oracle E-Business Suite 소프트웨어를 설치합니다.</p>	<p>Oracle E-Business Suite 권장 RPM을 Amazon RDS Custom DB 인스턴스의 Oracle Home 위치에 설치합니다. 자세한 내용은 Oracle Support 문서 #1330701.1을 참조하세요. 다음은 목록의 일부입니다. RPM 목록은 각 릴리스마다 변경되므로 필요한 RPM이 모두 설치되어 있는지 확인합니다.</p> <p>루트 사용자로 다음을 실행합니다.</p> <pre data-bbox="597 856 1026 1291"> \$ sudo yum -y update \$ sudo yum install -y elfutils-libelf-devel* \$ sudo yum install -y libXp-1.0.2-2.1*.i686 \$ sudo yum install -y libXp-1.0.2-2.1* \$ sudo yum install -y compat-libstdc++-* </pre> <p>다음 단계를 진행하기 전에 필요한 패치가 모두 설치되었는지 확인합니다.</p>	<p>DBA</p>

작업	설명	필요한 기술
VNC 서버를 설치합니다.	<p>참고: 예제 CD는 더 이상 필요하지 않으므로 Oracle 19c의 경우 이 단계를 생략할 수 있습니다. Oracle Support 문서 2782085.1을 참조하세요.</p> <p>Oracle 12.1.0.2의 경우:</p> <p>VNC 서버 및 해당 종속 데스크톱 패키지를 설치합니다. 이는 다음 단계에서 12c 예제 CD를 설치하기 위한 요구 사항입니다.</p> <p>1. 루트 사용자로 다음을 실행합니다.</p> <pre data-bbox="594 968 1027 1245">\$ sudo yum install -y tigervnc-server \$ sudo yum install -y *kde* \$ sudo yum install -y *xorg*</pre> <p>2. rdsdb 사용자용 VNC 서버를 시작하고 VNC의 암호를 설정합니다.</p> <pre data-bbox="594 1451 1027 1612">\$ su - rdsdb \$ vncserver :1 \$ vncpassword</pre>	DBA

작업	설명	필요한 기술
12c 예제 CD를 설치합니다.	<p>참고: 예제 CD는 더 이상 필요하지 않으므로 Oracle 19c의 경우 이 단계를 생략할 수 있습니다. Oracle Support 문서 2782085.1을 참조하세요.</p> <p>Oracle 12.1.0.2의 경우:</p> <ol style="list-style-type: none"> 1. https://edelivery.oracle.com/에서 설치 파일을 다운로드합니다. Oracle E-Business Suite 12.2.11-Oracle Database 12c 릴리스 1(12.1.0.2)의 경우 Linux x86-64 V100102-01.zip 예제를 찾아보세요. 2. 예제 CD를 저장할 디렉토리를 생성합니다. <pre data-bbox="597 1108 1026 1228">\$ mkdir /RMAN/12c examples</pre> 3. 선택한 전송 메커니즘(예: SCP)을 사용하여 예제 CD .zip 파일을 이 디렉토리에 복사합니다. <pre data-bbox="597 1486 1026 1564">V100102-01.zip</pre> 4. 소유권을 rdsdb로 변경합니다. <pre data-bbox="597 1726 1026 1837">\$ chown -R rdsdb:rdsdb /RMAN/12cexamples</pre> 	DBA

작업	설명	필요한 기술
	<p>5. rdsdb 사용자로서 파일의 압축을 풉니다.</p> <pre data-bbox="597 331 1026 411">\$ unzip V10010201.zip</pre> <p>6. VNC 클라이언트 및 Amazon RDS Custom에 액세스할 수 있는 클라이언트에서 연결합니다. VNC에 대한 액세스를 허용하는 데 필요한 네트워크 연결 및 방화벽 포트가 열려 있는지 확인해야 합니다. 예를 들어, <code>display :1</code>에서 실행 중인 VNC 서버는 Amazon RDS Custom EC2 호스트와 연결된 보안 그룹에서 포트 5901 개방이 필요합니다.</p> <p>7. 예제 CD를 복사한 디렉터리로 변경합니다.</p> <pre data-bbox="597 1171 1026 1293">\$ cd /RMAN/12cexamples/examples</pre> <p>8. 설치 관리자를 실행합니다. <code>oraInst.loc</code> 파일 위치를 확인합니다.</p> <pre data-bbox="597 1499 1026 1696">./runInstaller - invPtrLoc /rdsdbbin /oracle.12.1.custo m.r1.EE.1/oraInst.loc</pre> <p>9. 예제 CD를 설치하는 동안 다음 파라미터를 사용합니다.</p>	

작업	설명	필요한 기술
	<div data-bbox="594 212 1027 604" style="border: 1px solid #ccc; border-radius: 10px; padding: 10px; margin-bottom: 10px;"> Skip Software Update Downloads Select Oracle Home 12.1.0.2 (Oracle Base = /rdsdbbin) (Software Location = /rdsdbbin/oracle/12.1.custom.r1.EE.1) </div> <p>10. 설치 프로그램에는 프롬프트가 있는 다섯 단계가 포함되어 있습니다. 설치가 완료될 때까지 각 단계를 따릅니다.</p>	

스타터 데이터베이스를 삭제하고 데이터베이스 파일을 저장할 디렉토리를 생성합니다.

작업	설명	필요한 기술
<p>자동화 모드를 일시 중지합니다.</p>	<p>자동화가 RMAN 활동을 방해하지 않도록 하려면 다음 단계를 진행하기 전에 Amazon RDS Custom DB 인스턴스에서 자동화 모드를 일시 중지해야 합니다.</p> <p>다음 AWS Command Line Interface(AWS CLI) 명령을 사용하여 자동화를 일시 중지합니다. (먼저 AWS CLI를 구성을 했는지 확인합니다.)</p> <div data-bbox="594 1696 1027 1873" style="border: 1px solid #ccc; border-radius: 10px; padding: 10px; margin-top: 10px;"> <pre>aws rds modify-db-instance \ --db-instance-id entifier VIS \</pre> </div>	<p>DBA</p>

작업	설명	필요한 기술
	<pre data-bbox="597 212 1026 466">--automation-mode all- paused \ --resume-full-au tomation-mode-minute 360 \ --region eu-west-1</pre> <p data-bbox="597 499 1026 772">일시 중지 기간을 지정할 때는 RMAN 복원을 위한 충분한 시간을 두고 있어야 합니다. 이는 소스 데이터베이스의 크기에 따라 달라지므로 360 값을 적절히 수정합니다.</p>	
스타터 데이터베이스를 삭제합니다.	<p data-bbox="597 821 1026 905">기존 Amazon RDS Custom 데이터베이스를 삭제합니다.</p> <p data-bbox="597 947 1026 1125">Oracle Home 사용자의 권한으로 다음 명령을 실행합니다. (사용자 정의하지 않은 경우 기본 사용자는 rdsdb입니다.)</p> <pre data-bbox="597 1167 1026 1556">\$ sqlplus / as sysdba SQL> shutdown immediate ; SQL> startup nomount restrict; SQL> alter database mount; SQL> drop database; SQL> exit</pre>	DBA

작업	설명	필요한 기술
<p>데이터베이스 파일을 저장할 디렉터리를 생성합니다.</p>	<p>Oracle 12.1.0.2의 경우:</p> <p>데이터베이스, 제어 파일, 데이터 파일 및 온라인 로그용 디렉터리를 생성합니다. 이전 명령에서 <code>control_files</code> 파라미터의 상위 디렉터리(이 경우 <code>VIS_A</code>)를 사용합니다. Oracle Home 사용자(기본값 <code>rdsdb</code>)로 다음 명령을 실행합니다.</p> <pre data-bbox="594 758 1029 1041"> \$ mkdir -p /rdsbdbdata/db/VIS_A/controlfile \$ mkdir -p /rdsbdbdata/db/VIS_A/datafile \$ mkdir -p /rdsbdbdata/db/VIS_A/onlineolog </pre> <p>Oracle 19c의 경우:</p> <p>데이터베이스, 제어 파일, 데이터 파일 및 온라인 로그용 디렉터리를 생성합니다. 이전 명령에서 <code>control_files</code> 파라미터의 상위 디렉터리(이 경우 <code>VISCDB_A</code>)를 사용합니다. Oracle Home 사용자(기본값 <code>rdsdb</code>)로 다음 명령을 실행합니다.</p> <pre data-bbox="594 1612 1029 1856"> \$ mkdir -p /rdsbdbdata/db/cdb/VISCDB_A/controlfile \$ mkdir -p /rdsbdbdata/db/cdb/VISCDB_A/datafile </pre>	DBA

작업	설명	필요한 기술
	<pre>\$ mkdir -p /rdsdbdata/db/cdb/VISCDB_A/onlineolog \$ mkdir -p /rdsdbdata/db/cdb/VISCDB_A/onlineolog/arch \$ mkdir /rdsdbdata/db/pdb/VISCDB_A</pre>	

작업	설명	필요한 기술
<p>Oracle E-Business Suite의 파라미터 파일을 생성하고 수정합니다.</p>	<p>이 단계에서는 소스 데이터베이스에서 서버 파라미터 파일 (SPFILE)을 복사하지 않습니다. 대신 Amazon RDS Custom DB 인스턴스로 만든 표준 파라미터 파일(PFILE)을 사용하고 Oracle E-Business Suite에 필요한 파라미터를 추가합니다.</p> <p>데이터베이스를 삭제하면 Amazon RDS 자동화가 Amazon RDS Custom 데이터베이스와 연결된 <code>init.ora</code> 파일의 백업을 생성합니다. 이 파일은 <code>oracle_pfile</code> 에서 호출되며 <code>/rdsdbdata/config</code> 위치에 있습니다.</p> <p>Oracle 12.1.0.2의 경우:</p> <ol style="list-style-type: none"> <code>/rdsdbdata/config/oracle_pfile</code> 를 <code>\$ORACLE_HOME</code> 에 복사합니다. <pre data-bbox="594 1352 1029 1514">\$ cp /rdsdbdata/config/oracle_pfile \$ORACLE_HOME/dbs/initVIS.ora</pre> <ol style="list-style-type: none"> Amazon RDS Custom DB 인스턴스에서 <code>initVIS.ora</code> 파일을 편집합니다. <code>thtm</code>의 모든 파라미터를 검증하고 필요에 따라 파라미터를 추가합니다. 자세한 내용은 Oracle 	<p>DBA</p>

작업	설명	필요한 기술
	<p>Support 문서 396009.1을 참조하세요.</p> <p>중요: 추가하는 파라미터에 설명이 없는지 확인합니다. 댓글로 인해 읽기 전용 복제본 생성 및 PITR (point-in-time 복구) 발행 등 자동화 관련 문제가 발생할 수 있습니다.</p> <p>3. 요구 사항에 따라 다음과 유사한 파라미터를 <code>initVIS.ora</code> 파일에 추가합니다.</p> <pre> *.workarea_size_policy='AUTO' *.plsql_code_type='INTERPRETED' *.cursor_sharing='EXACT' *._b_tree_bitmap_plans=FALSE *.session_cached_cursors=500 *.optimizer_adaptive_features=false *.optimizer_secure_view_merging=false *.SQL92_SECURITY=TRUE *.temp_undo_enabled=true *_system_trig_enabled = TRUE nls_language = american nls_territory = america nls_numeric_characters = "., " nls_comp = binary nls_sort = binary </pre>	

작업	설명	필요한 기술
	<pre> nls_date_format = DD- MON-RR nls_length_semantics = BYTE aq_tm_processes = 1 _sort_elimination_cost_ratio = 5 _like_with_bind _as_equality = TRUE _fast_full_scan_enabled = FALSE _b_tree_bitmap_plans = FALSE optimizer_secure_view_merging = FALSE _optimizer_autostats_job = FALSE parallel_max_servers = 8 parallel_min_servers = 0 parallel_degree_policy = MANUAL sec_case_sensitive_logon = FALSE compatible = 12.1.0 dictionary_accessibility = FALSE utl_file_dir = /tmp </pre> <p>4. 다음을 수정합니다. 값은 소스 시스템에 따라 달라지므로 현재 설정에 따라 수정합니다.</p> <pre> *.open_cursors=500 *.undo_tablespace ='APPS_UNDOTS1 </pre> <p>5. SPFILE 참조를 제거합니다.</p>	

작업	설명	필요한 기술
	<div data-bbox="597 212 1024 367" style="border: 1px solid #ccc; border-radius: 10px; padding: 10px; margin-bottom: 10px;"> <pre>*.spfile='/rdsdbbin/oracle/dbs/spfileVIS.ora'</pre> </div> <p>참고:</p> <ul style="list-style-type: none"> • control_files 및 db_unique_name 에 대해 Amazon RDS Custom PFILE에서 제공한 값을 변경하지 않습니다. Amazon RDS는 이러한 값을 예상합니다. 이러한 기준을 벗어나면 나중에 읽기 전용 복제본을 만들려고 할 때 문제가 발생할 수 있습니다. • Amazon RDS Custom은 기본적으로 자동 메모리 관리(AMM)를 사용합니다. Hugesmem을 사용하려는 경우 자동 공유 메모리 관리(ASMM)를 사용하도록 Amazon RDS Custom을 구성할 수 있습니다. • memory_max_target 파라미터는 기본적으로 활성화되어 있습니다. Amazon RDS 프레임워크는 백그라운드에서 이를 사용하여 읽기 전용 복제본을 생성합니다. <p>6. startup nomount 명령을 실행하여 initVIS.ora 파</p>	

작업	설명	필요한 기술
	<p>일에 문제가 없는지 확인합니다.</p> <pre>SQL> startup nomount pfile=/rdsdbbin/oracle/dbs/initVIS.ora; SQL> create spfile='/rdsdbdata/admin/VIS/pfile/spfileVIS.ora' from pfile; SQL> exit</pre> <p>7. SPFILE용 심볼릭 링크를 생성합니다.</p> <pre>\$ ln -s /rdsdbdata/admin/VIS/pfile/spfileVIS.ora \$ORACLE_HOME/dbs/</pre> <p>Oracle 19c의 경우:</p> <p>1. /rdsdbdata/config/oracle_pfile 를 \$ORACLE_HOME 에 복사합니다.</p> <pre>\$ cp /rdsdbdata/config/oracle_pfile \$ORACLE_HOME/dbs/initVIS.ora .ora</pre> <p>2. Amazon RDS Custom DB 인스턴스에서 initVIS.ora 파일을 편집합니다. thtm의 모든 파라미터를 검증하고 필요에 따라 파라미터</p>	

작업	설명	필요한 기술
	<p>를 추가합니다. 자세한 내용은 Oracle Support 문서 396009.1 을 참조하세요.</p> <p>중요: 추가하는 파라미터에 설명이 없는지 확인합니다. 댓글이 있으면 읽기 전용 복제본을 생성하고 point-in-time 복구 (PITR) 를 실행하는 등 자동화에 문제가 발생할 수 있습니다.</p> <p>3. 요구 사항에 따라 다음과 유사한 파라미터를 <code>initVISCD B.ora</code> 파일에 추가합니다.</p> <pre data-bbox="597 871 1026 1881"> *.instance_name=VI SCDB *.sec_case_sensitive_logon= FALSE *.result_cache_max_size = 600M *.optimizer_adaptive_plans =TRUE *.optimizer_adaptive_statistics = FALSE *.pga_aggregate_limit = 0 *.temp_undo_enabled = FALSE *._pdb_name_case_sensitive = TRUE *.event='10946 trace name context forever, level 8454144' *.workarea_size_policy='AUTO' *.plsql_code_type=' INTERPRETED' *.cursor_sharing=' EXACT'</pre>	

작업	설명	필요한 기술
	<pre> *_b_tree_bitmap_pla ns=FALSE *.session_cached_c ursors=500 *.optimizer_secu re_view_merging=false *.SQL92_SECURITY=TRUE _system_trig_enabled = TRUE nls_language = american nls_territory = america nls_numeric_charact ers = "., " nls_comp = binary nls_sort = binary nls_date_format = DD- MON-RR nls_length_semantics = BYTE aq_tm_processes = 1 _sort_eliminatio n_cost_ratio =5 _like_with_bind _as_equality = TRUE _fast_full_scan_enable d = FALSE _b_tree_bitmap_plans = FALSE optimizer_secure_view _merging = FALSE _optimizer_autostats_ job = FALSE parallel_max_servers = 8 parallel_min_servers = 0 parallel_degree_policy = MANUAL </pre>	

작업	설명	필요한 기술
	<p>4. 다음을 수정합니다. 값은 소스 시스템에 따라 달라지므로 현재 설정에 따라 수정합니다.</p> <pre data-bbox="597 380 1027 537">*.open_cursors=500 *.undo_tablespace ='UNDOTBS1'</pre> <p>5. SPFILE 참조를 제거합니다.</p> <pre data-bbox="597 646 1027 804">*.spfile='/rdsdbbin/oracle/dbs/spfileVISDB.ora'</pre> <p>참고:</p> <ul data-bbox="597 919 1027 1780" style="list-style-type: none"> • control_files 및 db_unique_name 에 대해 Amazon RDS Custom PFILE에서 제공한 값을 변경하지 않습니다. Amazon RDS는 이러한 값을 예상합니다. 이러한 기준을 벗어나면 나중에 읽기 전용 복제본을 만들려고 할 때 문제가 발생할 수 있습니다. • Amazon RDS Custom은 기본적으로 자동 메모리 관리(AMM)를 사용합니다. Hugesmem을 사용하려는 경우 자동 공유 메모리 관리(ASMM)를 사용하도록 Amazon RDS Custom을 구성할 수 있습니다. 	

작업	설명	필요한 기술
	<ul style="list-style-type: none"> • <code>memory_max_target</code> 파라미터는 기본적으로 활성화되어 있습니다. Amazon RDS 프레임워크는 백그라운드에서 이를 사용하여 읽기 전용 복제본을 생성합니다. <p>6. <code>startup nomount</code> 명령을 실행하여 <code>initVISCD.B.ora</code> 파일에 문제가 없는지 확인합니다.</p> <pre>SQL> startup nomount pfile=/rdsdbbin/oracle/dbs/initVISCD B.ora; SQL> create spfile='/rdsbdbdata/admin/VISCDB/pfile/spfileVISCDB.ora' from pfile; SQL> exit</pre> <p>7. SPFILE용 심볼릭 링크를 생성합니다.</p> <pre>\$ ln -s /rdsbdbdata/admin/VISCDB/pfile/spfileVISCDB.ora \$ORACLE_HOME/dbs/</pre>	

작업	설명	필요한 기술
<p>백업에서 Amazon RDS Custom 데이터베이스를 복원합니다.</p>	<p>Oracle 12.1.0.2의 경우:</p> <ol style="list-style-type: none"> 이전에 소스에서 캡처한 백업 파일을 사용하여 제어 파일을 복원합니다. <pre data-bbox="597 474 1027 1625"> RMAN> connect target / RMAN> RESTORE CONTROLFILE FROM '/RMAN/vi sdb_full_bkp_100r1 sbt'; Starting restore at 10- APR-22 using target database control file instead of recovery catalog allocated channel: ORA_DISK_1 channel ORA_DISK_ 1: SID=201 device type=DISK channel ORA_DISK_1: restoring control file channel ORA_DISK_ 1: restore complete, elapsed time: 00:00:01 output file name=/rds dbdata/db/VIS_A/co ntrolfile/control- 01.ctl Finished restore at 10- APR-22 </pre> <ol style="list-style-type: none"> RMAN restore를 발행할 수 있도록 백업 조각의 카탈로그를 작성합니다. 	<p>DBA</p>

작업	설명	필요한 기술
	<pre> RMAN> alter database mount; RMAN> catalog start with '/RMAN/visdb'; </pre> <p>3. 데이터베이스를 복원하기 위한 스크립트를 생성합니다.</p> <pre> \$ vi restore.sh rman target / log=/home /rdpdb/rman.log << EOF run { set newname for database to '/rdpdbdata/db/VIS _A/datafile/%b'; restore database; switch datafile all; switch tempfile all; } EOF </pre> <p>4. 소스를 대상 Amazon RDS Custom 데이터베이스로 복원합니다. 스크립트 실행을 허용하도록 스크립트 권한을 변경한 다음 restore.sh 스크립트를 실행하여 데이터베이스를 복원해야 합니다.</p> <pre> \$ chmod 755 restore.sh \$ nohup ./restore.sh & </pre> <p>Oracle 19c의 경우:</p>	

작업	설명	필요한 기술
	<p>1. 이전에 소스에서 캡처한 백업 파일을 사용하여 제어 파일을 복원합니다.</p> <pre data-bbox="597 380 1029 1409"> RMAN> connect target / RMAN> RESTORE CONTROLFILE FROM '/RMAN/controlfile/trl.bak'; Starting restore at 07-JUN-23 using target database control file instead of recovery catalog allocated channel: ORA_DISK_1 channel ORA_DISK_1: SID=201 device type=DISK channel ORA_DISK_1: restoring control file channel ORA_DISK_1: restore complete, elapsed time: 00:00:01 output file name=/rdsdbdata/db/cdb/VISCD_BA/controlfile/control-01.ctl Finished restore at 07-JUN-23 </pre> <p>2. RMAN restore를 발행할 수 있도록 백업 조각의 카탈로그를 작성합니다.</p> <pre data-bbox="597 1619 1029 1810"> RMAN> alter database mount; RMAN> catalog start with '/RMAN/visdb'; </pre>	

작업	설명	필요한 기술
	<p>start with 명령에 문제가 있는 경우 백업 조각을 개별적으로 추가할 수 있습니다. 예를 들면 다음과 같습니다.</p> <pre data-bbox="597 426 1027 583"> RMAN> catalog backuppie ce '/RMAN/visdb_full_ bkp_1d1e507m'; </pre> <p>그런 다음 각 백업 조각에 대해 명령을 반복합니다.</p> <p>3. 스크립트를 생성하여 데이터베이스를 복원합니다. 요구 사항에 따라 플러그 가능한 데이터베이스 이름을 수정합니다. 사용 가능한 vCPU 수에 따라 병렬 채널을 할당하여 복원 프로세스의 속도를 높입니다.</p> <pre data-bbox="597 1108 1027 1839"> \$ vi restore.sh rman target / log=/home /rdsdb/rmancdb.log << EOF run { allocate channel c1 type disk; allocate channel c2 type disk; allocate channel c<N> type disk; set newname for database to '/rdsbdbdata/db/cdb /VISCDB_A/datafile/ %b'; set newname for database root to '/rdsbdba </pre>	

작업	설명	필요한 기술
	<pre data-bbox="609 210 1015 1018"> ta/db/cdb/VISCDB_A/ datafile/%f_%b'; set newname for database "PDB\$SEED" to '/rdsbdbdata/db/cdb/ pdbseed/%f_%b'; set newname for pluggable database VIS to '/rdsbdbdata/db/pdb /VISCDB_A/%f_%b'; restore database; switch datafile all; switch tempfile all; release channel c1; release channel c2; release channel c3; release channel c<N>; } EOF </pre> <p data-bbox="592 1060 1031 1375">4. 소스를 대상 Amazon RDS Custom 데이터베이스로 복원합니다. 스크립트 실행을 허용하도록 스크립트 권한을 변경한 다음 restore.sh 스크립트를 실행하여 데이터베이스를 복원해야 합니다.</p> <pre data-bbox="609 1438 1015 1522"> \$ chmod 755 restore.sh \$ nohup ./restore.sh & </pre>	

작업	설명	필요한 기술
로그 파일에 문제가 있는지 확인합니다.	<p>Oracle 12.1.0.2의 경우:</p> <ol style="list-style-type: none"> 1. rman.log 파일을 검토하여 문제가 없는지 확인합니다. <pre data-bbox="597 428 1027 541">\$ cat /home/irdsdb/rman.log</pre> <ol style="list-style-type: none"> 2. 제어 파일에 등록된 로그 파일의 경로를 확인합니다. <pre data-bbox="597 709 1027 1297">SQL> select member from v\$logfile; MEMBER ----- ----- ----- ----- ----- /d01/oracle/VIS/data/log1.dbf /d01/oracle/VIS/data/log2.dbf /d01/oracle/VIS/data/log3.dbf</pre> <ol style="list-style-type: none"> 3. 대상의 파일 경로와 일치하도록 로그 파일 이름을 바꿉니다. 경로를 이전 단계의 출력과 일치하도록 바꿉니다. <pre data-bbox="597 1556 1027 1803">SQL> ALTER DATABASE RENAME FILE '/d01/oracle/VIS/data/log1.dbf' TO '/irdsdbdata/db/VIS_A/onlinelog/log1.dbf';</pre>	DBA

작업	설명	필요한 기술
	<pre>SQL> ALTER DATABASE RENAME FILE '/d01/oracle/VIS/data/log2.dbf' TO '/rdsdbdata/db/VIS_A/online/log2.dbf'; SQL> ALTER DATABASE RENAME FILE '/d01/oracle/VIS/data/log3.dbf' TO '/rdsdbdata/db/VIS_A/online/log3.dbf';</pre> <p>Oracle 19c의 경우:</p> <ol style="list-style-type: none"> 1. rmancdb.log 파일을 검토하여 문제가 없는지 확인합니다. <pre>\$ cat /home/rdsdb/rmancdb.log</pre> <ol style="list-style-type: none"> 2. 제어 파일에 등록된 로그 파일의 경로를 확인합니다. <pre>SQL> select member from v\$logfile; MEMBER ----- ----- ----- ----- ----- /d01/oracle/VIS/oradata/VIS_CDB/redo03.log /d01/oracle/VIS/oradata/VIS_CDB/redo02.log</pre>	

작업	설명	필요한 기술
	<pre data-bbox="609 210 1015 346">/d01/oracle/VIS/ oradata/VIS/redo01.log</pre> <p data-bbox="592 378 1031 556">3. 대상의 파일 경로와 일치하도록 로그 파일 이름을 바꿉니다. 경로를 이전 단계의 출력과 일치하도록 바꿉니다.</p> <pre data-bbox="609 619 1015 1459">SQL> ALTER DATABASE RENAME FILE '/d01/oracle/VIS/oradata/VIS/redo01.log' TO '/rdsdbdata/db/cdb/VIS/redo01.log'; SQL> ALTER DATABASE RENAME FILE '/d01/oracle/VIS/oradata/VIS/redo02.log' TO '/rdsdbdata/db/cdb/VIS/redo02.log'; SQL> ALTER DATABASE RENAME FILE '/d01/oracle/VIS/oradata/VIS/redo03.log' TO '/rdsdbdata/db/cdb/VIS/redo03.log';</pre> <p data-bbox="592 1512 1031 1638">4. 제어 파일에 등록된 경로, 로그 파일 상태 및 그룹 번호를 확인합니다.</p> <pre data-bbox="609 1690 1015 1764">SQL> column REDOLOG_F FILE_NAME format a50</pre>	

작업	설명	필요한 기술
	<pre> SQL> SELECT a.GROUP#, a.status, b.MEMBER AS REDOLOG_FILE_NAME, (a.BYTES/1024/1024) AS SIZE_MB FROM v\$log a JOIN v\$logfile b ON a.Group#=b.Group# ORDER BY a.GROUP#; GROUP# STATUS REDOLOG_F ILE_NAME SIZE_MB 1 CURRENT /rdsbdat a/db/cdb/VISCD_B_A/ onlineolog/log1.dbf 512 2 INACTIVE /rdsbdat a/db/cdb/VISCD_B_A/ onlineolog/log2.dbf 512 3 INACTIVE /rdsbdat a/db/cdb/VISCD_B_A/ onlineolog/log3.dbf 512 </pre>	

작업	설명	필요한 기술
<p>Amazon RDS Custom 데이터베이스를 열고 OMF 로그 파일을 생성할 수 있는지 확인합니다.</p>	<p>Oracle용 Amazon RDS Custom은 Oracle 관리 파일(OMF)을 사용하여 작업을 단순화합니다. 읽기 전용 복제본을 독립형 인스턴스로 승격할 수 있지만 먼저 OMF를 사용하여 로그 파일을 생성해야 합니다. 이는 인스턴스가 승격될 때 올바른 경로가 사용되도록 하기 위한 것입니다. 읽기 전용 복제본을 승격하는 방법에 대한 자세한 내용은 Amazon RDS 설명서를 참조하세요. OMF 파일을 사용하지 않으면 읽기 전용 복제본을 승격하려고 할 때 잠재적으로 문제가 발생할 수 있습니다.</p> <p>1. <code>resetlogs</code> 를 사용하여 데이터베이스를 엽니다.</p> <pre data-bbox="591 1188 1029 1310">SQL> alter database open resetlogs;</pre> <p>참고: ORA-00392: log xx of thread 1 is being cleared, operation not allowed 오류가 발생하는 경우 ORA-00392에 대한 문제 해결 섹션에 있는 단계를 따릅니다.</p> <p>2. 데이터베이스가 열려 있는지 확인합니다.</p> <pre data-bbox="591 1785 1029 1877">SQL> select open_mode from v\$database;</pre>	<p>DBA</p>

작업	설명	필요한 기술
	<pre> OPEN_MODE ----- READ WRITE </pre> <p>3. OMF 로그 파일을 생성합니다. 이전 로그파일 쿼리의 출력을 사용하여 요구 사항에 따라 그룹 번호, 그룹 수 및 크기를 변경합니다. 다음 예제는 단순화를 위해 그룹 4에서 시작하여 세 개의 그룹을 추가합니다.</p> <pre> SQL> alter database add logfile group 4 size 512M; Database altered. SQL> alter database add logfile group 5 size 512M; Database altered. SQL> alter database add logfile group 6 size 512M; Database altered. </pre> <p>4. OMF가 아닌 이전 파일은 삭제합니다. 다음은 이전 단계에서 요구 사항과 쿼리 결과를 기반으로 사용자 지정할 수 있는 예제입니다.</p> <pre> SQL> alter database drop logfile group 1; System altered. SQL> alter database drop logfile group 2; System altered. </pre>	

작업	설명	필요한 기술
	<pre>SQL> alter database drop logfile group 3; System altered.</pre> <p>참고: 로그 파일을 삭제하려고 할 때 ORA-01624 오류가 발생하는 경우 문제 해결 섹션을 참조하세요.</p> <p>5. 생성된 OMF 파일을 볼 수 있는지 확인합니다. (디렉터리 경로는 Oracle 12.1.0.2와 19c 이 서로 다르지만 개념은 동일합니다.)</p> <pre>SQL> select member from v\$logfile;</pre> <pre>MEMBER ----- ----- ----- /irdsdbdata/db/cdb/ VISCDB_A/onlineolog/ o1_mf_4_ksrbslny_.log /irdsdbdata/db/cdb/VIS CDB_A/onlineolog/o1 _mf_5_ksrchw0k_.log /irdsdbdata/db/cdb/ VISCDB_A/onlineolog/ o1_mf_6_ksrcn19v_.log</pre> <p>6. 데이터베이스를 재시작하고 인스턴스에서 SPFILE을 사용 중인지 확인합니다.</p> <pre>SQL> shutdown immediate SQL> startup</pre>	

작업	설명	필요한 기술
	<pre>SQL> show parameter spfile</pre> <p>Oracle 12.1.0.2의 경우 이 쿼리는 다음을 반환합니다.</p> <pre>spfile /rdsdbbin /oracle/dbs/spfile VIS.ora</pre> <p>Oracle 19c의 경우 쿼리는 다음을 반환합니다.</p> <pre>spfile /rdsdbbin /oracle/dbs/spfile VISCDB.ora</pre> <p>7. Oracle 19c의 경우에만 컨테이너 데이터베이스의 상태를 확인하고 필요한 경우 엽니다.</p> <pre>SQL> show pdbs CON_ID CON_NAME OPEN MODE RESTRICTED ----- - 2 PDB\$SEED READ ONLY NO 3 VIS MOUNTED NO</pre> <pre>SQL> alter session set container=VIS; Session altered.</pre> <pre>SQL> alter database open;</pre>	

작업	설명	필요한 기술
	<pre>Database altered. SQL> alter database save state; Database altered. SQL> show pdbs CON_ID CON_NAME OPEN MODE RESTRICTED ----- ----- ----- 3 VIS READ WRITE NO SQL> exit</pre> <p>8. PFILE을 사용하지 않으므로 \$ORACLE_HOME/dbs 에서 init.ora 파일을 삭제합니다.</p> <pre>\$ cd \$ORACLE_HOME/dbs</pre> <p>Oracle 12.1.0.2의 경우 다음 명령을 사용합니다.</p> <pre>\$ pwd /rdsdbbin/oracle/dbs \$ rm initVIS.ora</pre> <p>Oracle 19c의 경우 다음 명령을 사용합니다.</p> <pre>\$ pwd /rdsdbbin/oracle/dbs \$ rm initVISCDB.ora</pre>	

Secrets Manager에서 암호를 검색하고, 사용자를 생성하고, 암호를 변경합니다.

작업	설명	필요한 기술
Secrets Manager에서 암호를 검색합니다.	<p>콘솔에서 또는 AWS CLI를 사용하여 이러한 단계를 수행할 수 있습니다. 다음 단계는 콘솔에 대한 지침을 제공합니다.</p> <ol style="list-style-type: none"> 1. AWS 관리 콘솔에 로그인한 후 https://console.aws.amazon.com/rds/에서 Amazon RDS 콘솔을 엽니다. 2. 탐색 창에서 데이터베이스를 선택하고 Amazon RDS 데이터베이스를 선택합니다. 3. 구성을 선택하고 인스턴스의 리소스 ID(형식: db-WZ4WLC K6A0Q6TJGZKMGRCDI 3Y)를 기록합니다. 4. https://console.aws.amazon.com/secretsmanager/에서 AWS Secrets Manager 콘솔을 엽니다. 5. do-not-delete-custom-<resource_id> 와 같은 이름을 가진 암호를 선택합니다. 여기서 resource-id 는 3단계에서 기록해 둔 인스턴스의 ID를 나타냅니다. 6. 보안 암호 값 검색을 선택합니다. 	DBA

작업	설명	필요한 기술
RDSADMIN 사용자를 생성합니다.	<p>RDSADMIN은 Amazon RDS Custom DB 인스턴스의 모니터링 및 오케스트레이터 데이터베이스 사용자입니다. 시작 데이터베이스를 삭제하고 RMAN을 사용하여 대상 데이터베이스를 thtm에서 복원했으므로 복원 작업 후에 이 사용자를 다시 생성하여 Amazon RDS Custom 모니터링이 예상대로 작동하는지 확인해야 합니다. 또한 RDSADMIN 사용자를 위한 별도의 프로필과 테이블스페이스를 생성해야 합니다. Oracle 12.1.0.2와 19c 지침은 약간 다릅니다.</p> <p>Oracle 12.1.0.2의 경우:</p> <ol style="list-style-type: none"> 1. SQL 프롬프트에서 다음 명령을 입력합니다. <pre data-bbox="597 1222 1026 1852"> SQL> set echo on feedback on serverout on SQL> @?/rdbms/admin/utl pwmg.sql SQL> ALTER PROFILE DEFAULT LIMIT FAILED_LOGIN_ATTEMPTS UNLIMITED PASSWORD_LIFE_TIME UNLIMITED PASSWORD_VERIFY_F UNCTION NULL; </pre>	DBA

작업	설명	필요한 기술
	<p>2. RDSADMIN 프로필을 생성합니다.</p> <pre data-bbox="597 331 1024 1602"> SQL> create profile RDSADMIN LIMIT COMPOSITE_LIMIT UNLIMITED SESSIONS_PER_USER UNLIMITED CPU_PER_SESSION UNLIMITED CPU_PER_CALL UNLIMITED LOGICAL_READS_PER _SESSION UNLIMITED LOGICAL_READS_PER_CALL UNLIMITED IDLE_TIME UNLIMITED CONNECT_TIME UNLIMITED PRIVATE_SGA UNLIMITED FAILED_LOGIN_ATTEMPTS 10 PASSWORD_LIFE_TIME UNLIMITED PASSWORD_REUSE_TIME UNLIMITED PASSWORD_REUSE_MAX UNLIMITED PASSWORD_VERIFY_F UNCTION NULL PASSWORD_LOCK_TIME 86400/86400 PASSWORD_GRACE_TIME 604800/86400; </pre> <p>3. SYS, SYSTEM, DBSNMP 사용자 프로필을 RDSADMIN과 같이 설정합니다.</p>	

작업	설명	필요한 기술
	<pre>SQL> set echo on feedback on serverout on SQL> alter user SYS profile RDSADMIN; SQL> alter user SYSTEM profile RDSADMIN; SQL> alter user DBSNMP profile RDSADMIN;</pre> <p>4. RDSADMIN 테이블스페이스를 생성합니다.</p> <pre>SQL> create bigfile tablespace rdsadmin datafile size 7M autoextend on next 1m Logging online permanent blocksize 8192 extent managemen t local autoallocate default nocompress segment space managemen t auto;</pre> <p>5. RDSADMIN 사용자를 생성합니다. RDSADMIN 암호를 이전에 Secrets Manager에서 얻은 암호로 바꿉니다.</p> <pre>SQL> create user rdsadmin identified by xxxxxxxxxx Default tablespace rdsadmin Temporary tablespace temp profile rdsadmin ;</pre>	

작업	설명	필요한 기술
	<p>6. RDSADMIN에 권한을 부여합니다.</p> <pre> SQL> grant select on sys.v_\$instance to rdsadmin; SQL> grant select on sys.v_\$archived_log to rdsadmin; SQL> grant select on sys.v_\$database to rdsadmin; SQL> grant select on sys.v_\$database_in carnation to rdsadmin; SQL> grant select on dba_users to rdsadmin; SQL> grant alter system to rdsadmin; SQL> grant alter database to rdsadmin; SQL> grant connect to rdsadmin with admin option; SQL> grant resource to rdsadmin with admin option; SQL> alter user rdsadmin account unlock identified by xxxxxxxxxxxx; SQL> @?/rdbms/admin/use rlock.sql SQL> @?/rdbms/admin/utl rp.sql </pre> <p>Oracle 19c의 경우:</p> <ol style="list-style-type: none"> 1. SQL 프롬프트에서 다음 명령을 입력합니다. 	

작업	설명	필요한 기술
	<pre>SQL> set echo on feedback on serverout on SQL> @?/rdbms/admin/utl pwdmg.sql SQL> alter profile default LIMIT FAILED_LOGIN_ATTEMPTS UNLIMITED PASSWORD_LIFE_TIME UNLIMITED PASSWORD_VERIFY_F UNCTION NULL;</pre> <p>2. RDSADMIN 프로필을 생성합니다.</p> <p>참고: Oracle 19c에서 RDSADMIN에 접두사 C##이 붙습니다. 이는 데이터베이스 파라미터 common_user_prefix 가 C##으로 설정되었기 때문입니다. Oracle 12.1.0.2에서는 RDSADMIN에 접두사가 없습니다.</p> <pre>SQL> create profile C##RDSADMIN LIMIT COMPOSITE_LIMIT UNLIMITED SESSIONS_PER_USER UNLIMITED CPU_PER_SESSION UNLIMITED CPU_PER_CALL UNLIMITED</pre>	

작업	설명	필요한 기술
	<pre> LOGICAL_READS_PER _SESSION UNLIMITED LOGICAL_READS_PER_CALL UNLIMITED IDLE_TIME UNLIMITED CONNECT_TIME UNLIMITED PRIVATE_SGA UNLIMITED FAILED_LOGIN_ATTEMPTS 10 PASSWORD_LIFE_TIME UNLIMITED PASSWORD_REUSE_TIME UNLIMITED PASSWORD_REUSE_MAX UNLIMITED PASSWORD_VERIFY_F UNCTION NULL PASSWORD_LOCK_TIME 86400/86400 PASSWORD_GRACE_TIME 604800/86400; </pre> <p>3. SYS, SYSTEM, DBSNMP 사용자 프로필을 RDSADMIN과 같이 설정합니다.</p> <pre> SQL> alter user SYS profile C##RDSADMIN; SQL> alter user SYSTEM profile C##RDSADMIN; SQL> alter user DBSNMP profile C##RDSADMIN; </pre> <p>4. RDSADMIN 테이블스페이스를 생성합니다.</p> <pre> SQL> create bigfile tablespace rdsadmin datafile size 7M autoextend on next 1m </pre>	

작업	설명	필요한 기술
	<pre>Logging online permanent blocksize 8192 extent managemen t local autoallocate default nocompress segment space managemen t auto;</pre> <p>5. RDSADMIN 사용자를 생성합니다. RDSADMIN 암호를 이전에 Secrets Manager에서 얻은 암호로 바꿉니다.</p> <pre>SQL> create user C##rdsadmin identifie d by xxxxxxxxxxxx profile C##rdsadmin container=all;</pre> <p>6. RDSADMIN에 권한을 부여합니다.</p> <pre>SQL> grant select on sys.v_\$instance to c##rdsadmin; SQL> grant select on sys.v_\$archived_log to c##rdsadmin; SQL> grant select on sys.v_\$database to c##rdsadmin; SQL> grant select on sys.v_\$database_in carnation to c##rdsadm in; SQL> grant select on dba_users to c##rdsadm in;</pre>	

작업	설명	필요한 기술
	<pre> SQL> grant alter system to C##rdsadmin; SQL> grant alter database to C##rdsadm in; SQL> grant connect to C##rdsadmin with admin option; SQL> grant resource to C##rdsadmin with admin option; SQL> alter user C##rdsadmin account unlock identified by xxxxxxxxxxxx; SQL> @?/rdbms/admin/use rlock.sql SQL> @?/rdbms/admin/utl rp.sql </pre>	

작업	설명	필요한 기술
<p>마스터 사용자를 생성합니다.</p>	<p>시작 데이터베이스를 삭제하고 RMAN을 사용하여 소스 데이터베이스에서 대상 데이터베이스를 복원했으므로 마스터 사용자를 다시 만들어야 합니다. 이 예제에서 마스터 사용자 이름은 admin입니다.</p> <p>Oracle 12.1.0.2의 경우:</p> <pre>SQL> create user admin identified by <password>; SQL> grant dba to admin</pre> <p>Oracle 19c의 경우:</p> <pre>SQL> alter session set container=VIS; Session altered. SQL> create user admin identified by <password>; User created. SQL> grant dba to admin; Grant succeeded.</pre>	<p>DBA</p>

작업	설명	필요한 기술
수퍼 유저 암호를 변경합니다.	<p>1. Secrets Manager에서 검색한 암호를 사용하여 시스템 암호를 변경합니다.</p> <p>Oracle 12.1.0.2의 경우:</p> <pre>SQL> alter user sys identified by xxxxxxxxxxxx; SQL> alter user system identified by xxxxxxxxxxxx;</pre> <p>Oracle 19c의 경우:</p> <pre>SQL> alter user sys identified by xxxxxxxxxxxx container =all; SQL> alter user system identified by xxxxxxxxxxxx container =all;</pre> <p>1. EBS_SYSTEM 암호 변경</p> <p>Oracle 12.1.0.2의 경우:</p> <pre>SQL> alter user ebs_system identified by xxxxxxxxxxxx;</pre> <p>Oracle 19c의 경우:</p> <p>이 버전의 경우 컨테이너 데이터베이스에도 연결하여</p>	DBA

작업	설명	필요한 기술
	<p>EBS_SYSTEM 암호를 업데이트해야 합니다.</p> <pre data-bbox="597 331 1024 646">SQL> alter session set container=vis; SQL> alter user ebs_system identified by xxxxxxxxxx; SQL> exit;</pre> <p>이러한 암호를 변경하지 않으면 Amazon RDS Custom에 데이터베이스 모니터링 사용자 또는 사용자 보안 인증 정보가 변경되었다는 오류 메시지가 표시됩니다.</p>	

Oracle E-Business Suite용 디렉터리 생성, ETCC 설치, 자동 구성 실행

작업	설명	필요한 기술
<p>Oracle E-Business Suite에 필요한 디렉터리를 생성합니다.</p>	<p>1. Amazon RDS Custom Oracle 데이터베이스에서 Oracle Home 사용자로 다음 스크립트를 실행하여 9idata 디렉터를 \$ORACLE_HOME/nls/data/9idata 에 생성합니다. 이 디렉터리는 Oracle E-Business Suite에 필요합니다.</p> <pre data-bbox="597 1711 1024 1822">perl \$ORACLE_HOME/nls/data/old/cr9idata.pl</pre>	

작업	설명	필요한 기술
	<p>이후 단계에서 컨텍스트 지원 환경을 생성하므로 ORA_NLS10 메시지는 무시합니다.</p> <p>2. 공유 Amazon EFS 파일 시스템에서 이전에 생성한 appsutil.tar 파일을 복사하고 Amazon RDS Custom Oracle Home 디렉터리에 압축을 풉니다. 그러면 \$ORACLE_HOME 디렉터리에 appsutil 디렉터리가 생성됩니다.</p> <pre data-bbox="594 888 1027 1167"> \$ cd /RMAN/appsutil \$ cp sourceappsutil.tar \$ORACLE_HOME \$ cd \$ORACLE_HOME \$ tar xvf sourceappsutil.tar appsutil </pre> <p>3. 이전에 Amazon EFS 공유 파일 시스템에 저장한 appsutil.zip 파일을 복사합니다. 이 파일은 애플리케이션 계층에서 생성한 파일입니다.</p> <p>Amazon RDS Custom DB 인스턴스 rdsdb 사용자의 경우:</p> <pre data-bbox="594 1644 1027 1801"> \$ cp /RMAN/appsutil/appsutil.zip \$ORACLE_HOME \$ cd \$ORACLE_HOME </pre>	

작업	설명	필요한 기술
	<p>4. appsutil.zip 파일의 압축을 풀어 Oracle Home에 appsutil 디렉터리 및 하위 디렉터리를 생성합니다.</p> <div style="border: 1px solid #ccc; border-radius: 10px; padding: 5px; margin: 10px 0;"> <pre>\$ unzip -o appsutil.zip</pre> </div> <p>-o 옵션은 일부 파일을 덮어쓰게 됨을 의미합니다.</p>	

작업	설명	필요한 기술
<p>tsnames.ora 및 sqlnet.ora 파일을 구성합니다.</p>	<p>자동 구성 도구를 사용하여 데이터베이스에 연결할 수 있도록 tnsnames.ora 파일을 구성해야 합니다. 다음 예제에서는 tnsnames.ora 파일이 소프트웨어 링크되어 있지만 기본적으로 파일이 비어 있는 것을 볼 수 있습니다.</p> <pre data-bbox="597 632 1024 1507"> \$ cd \$ORACLE_HOME/network/admin \$ ls -ltr -rw-r--r-- 1 rdsdb database 373 Oct 31 2013 shrept.lst lrwxrwxrwx 1 rdsdb database 30 Feb 9 17:17 listener.ora - > /rdsbdbdata/config/ listener.ora lrwxrwxrwx 1 rdsdb database 28 Feb 9 17:17 sqlnet.ora - > /rdsbdbdata/config/ sqlnet.ora lrwxrwxrwx 1 rdsdb database 30 Feb 9 17:17 tnsnames.ora - > /rdsbdbdata/config/ tnsnames.ora </pre> <p>1. tnsnames.ora 항목을 생성합니다. Amazon RDS 자동화가 파일을 파싱하는 방식 때문에 항목에 스페이스, 주석 또는 추가 행이 없어야 합니다. 그렇지 않으면 -replica와 같은 일부 API를 사용할 때 문제가 발</p>	<p>DBA</p>

작업	설명	필요한 기술
	<p>생할 수 있습니다. create-db-instance-read 다음 템플릿을 예로 사용합니다.</p> <p>2. 요구 사항에 따라 포트, 호스트 및 SID를 교체합니다.</p> <pre data-bbox="597 506 1027 863">\$ vi tnsnames.ora VIS=(DESCRIPTION= (ADDRESS_LIST=(ADDRESS=(PROTOCOL=TCP) (PORT=1521)(HOST=xx.xx.xx.xx))) (CONNECT_DATA=(SID=VIS) (SERVER=DEDICATED)))</pre> <p>참고: 파일에는 추가 행이 없어야 합니다. 행을 제거하지 않으면 나중에 읽기 전용 복제본을 생성할 때 문제가 발생할 수 있습니다. 읽기 전용 복제본 생성이 실패하고 다음과 같은 오류 메시지가 표시될 수 있습니다. 작업에서 예외 발생:: 모든 호스트에서 RestrictReplication을 성공적으로 호출할 수 없습니다 HostManagerException.</p> <p>3. 데이터베이스에 연결할 수 있는지 확인합니다.</p> <pre data-bbox="597 1577 1027 1696">\$ tnsping vis OK (0 msec)</pre> <p>4. Oracle 19c의 경우에만 sqlnet.ora 파일을 업데이트합니다. 이렇게 하지 않</p>	

작업	설명	필요한 기술
	<p>면 데이터베이스에 연결하려고 할 때 ORA-01017: invalid username/password; logon denied 오류가 발생합니다. 다음과 일치하도록 \$ORACLE_HOME/network/admin 의 sqlnet.ora 를 편집합니다.</p> <pre data-bbox="597 569 1026 1045"> NAMES.DIRECTORY_PATH=(TNSNAMES, ONAMES, HOSTNAME) SQLNET.EXPIRE_TIME= 10 SQLNET.INBOUND_CONNECT_TIMEOUT =60 SQLNET.ALLOWED_LOGON_VERSION_SERVER=10 HTTPS_SSL_VERSION=undetermined </pre> <p>5. 연결 테스트</p> <pre data-bbox="597 1157 1026 1234"> \$ sqlplus apps/****@vis </pre>	

작업	설명	필요한 기술
<p>데이터베이스를 구성합니다.</p>	<p>이제 데이터베이스 연결을 테스트했으므로 appsutil 유틸리티를 사용하여 데이터베이스를 구성하여 컨텍스트 지원 환경을 만들 수 있습니다.</p> <p>Oracle 12.1.0.2의 경우:</p> <p>1. 다음 명령을 실행합니다.</p> <pre data-bbox="594 646 1029 1486"> \$ cd \$ORACLE_HOME/appsutil/bin \$ perl adbldxml.pl appsuser=apps Enter Hostname of Database server: oebs-db01 Enter Port of Database server: 1521 Enter SID of Database server: VIS Enter Database Service Name: VIS Enter the value for Display Variable: :1 The context file has been created at: /rdsdbbin/oracle/appsutil/VIS_oebs-db01.xml </pre> <p>2. 루트 사용자로 oraInst.loc 생성.</p> <pre data-bbox="594 1640 1029 1854"> \$ vi /etc/oraInst.loc inventory_loc=/rdsdbbin/oracle.12.1.custom.r1.EE.1/orainventory </pre>	<p>DBA</p>

작업	설명	필요한 기술
	<pre data-bbox="597 205 1024 268">inst_group=database</pre> <p data-bbox="597 304 1008 527">3. 이전 단계에서 생성한 컨텍스트 파일을 복제하여 논리적 호스트 이름을 설정합니다. rdsdb 사용자로 다음을 실행합니다.</p> <pre data-bbox="597 569 1024 961"> \$ cd \$ORACLE_HOME/appsu til/clone/bin \$ perl adclonctx.pl \ contextfile=[ORA CLE_HOME]/appsutil/ [current context file] \ template=[ORACLE _HOME]/appsutil/te mplate/adxdbctx.tmp </pre> <p data-bbox="597 1003 1008 1129">여기서 <code>oebs-db01log</code> 는 논리적 호스트 이름을 나타냅니다. 예:</p> <pre data-bbox="597 1171 1024 1858"> \$ perl adclonctx.pl \ contextfile=/rdsdbbin/ oracle.12.1.custom.r1 .EE.1/appsutil/VIS _oebs-db01.xml \ template=/rdsdbbin/ oracle/appsutil/ template/adxdbctx.tmp Target System Hostname (virtual or normal) [oebs-db01] : oebs- db01log Target System Base Directory : /rdsdbbin/ oracle Target Instance is RAC (y/n) [n] : n </pre>	

작업	설명	필요한 기술
	<pre> Target System Database SID : VIS Oracle OS User [irdsdb] : Oracle OS Group [irdsdb] : database Role separation is supported y/n [n] ? : n Target System utl_file_ dir Directory List : / tmp Number of DATA_TOP's on the Target System [1] : Target System DATA_TOP Directory 1 [/rdsdbbi n/oracle/data] : / rdsbdbdata/db/VIS_A/ datafile/ Target System RDBMS ORACLE_HOME Directory [/rdsdbbin/oracle/ 12.1.0] : /rdsdbbin/ oracle Do you want to preserve the Display [:1] (y/n) : y Do you want the target system to have the same port values as the source system (y/n) [y] ? : y The new database context file has been created : /rdsdbbin/oracle.1 2.1.custom.r1.EE.1/ </pre>	

작업	설명	필요한 기술
	<pre>appsutil/clone/bin/ VIS_oeps-db01log.xml contextfile=/rdsdbbin/ oracle.12.1.custom .r1.EE.1/appsutil/ clone/bin/VIS_oeps- db01log.xml</pre> <p>Oracle 19c의 경우:</p> <p>1. 다음 명령을 실행합니다.</p> <pre>\$ cd \$ORACLE_HOME/appsutil/bin \$ perl adbldxml.pl appsuser=apps Enter Hostname of Database server: oebs- db01 Enter Port of Database server: 1521 Enter SID of Database server: VIS Enter the database listener name:L_VI SCDB_001 Enter the value for Display Variable: :1 The context file has been created at: /rdsdbbin/oracle/ appsutil/VIS_oeps- db01.xml</pre> <p>2. 루트 사용자로 oraInst.loc 생성.</p> <pre>\$ vi /etc/oraInst.loc</pre>	

작업	설명	필요한 기술
	<pre data-bbox="609 210 1015 388">inventory_loc=/rdsdbbin/oracle/oraInventory inst_group=database</pre> <p data-bbox="592 420 1015 651">3. 이전 단계에서 생성한 컨텍스트 파일을 복제하여 논리적 호스트 이름을 설정합니다. rdsdb 사용자로 다음을 실행합니다.</p> <pre data-bbox="609 693 1015 1081">\$ cd \$ORACLE_HOME/appsutil/clone/bin \$ perl adclonctx.pl \ contextfile=[ORACLE_HOME]/appsutil/[current context file] \ template=[ORACLE_HOME]/appsutil/template/adxdbctx.tmp</pre> <p data-bbox="592 1123 1015 1249">여기서 <code>oebs-db01log</code> 는 논리적 호스트 이름을 나타냅니다. 예:</p> <pre data-bbox="609 1291 1015 1858">\$ perl adclonctx.pl \ contextfile=/rdsdbbin/oracle/appsutil/VIS_oebs-db01.xml \ template=/rdsdbbin/oracle/appsutil/template/adxdbctx.tmp Target System Hostname (virtual or normal) [oebs-db01] : oebs-db01log Target System Base Directory : /rdsdbbin/oracle</pre>	

작업	설명	필요한 기술
	<pre> Target Instance is RAC (y/n) [n] : n Target System CDB Name : VISCDB Target System PDB Name : VIS Oracle OS User [oracle] : rdsdb Oracle OS Group [dba] : database Role separation is supported y/n [n] ? : n Number of DATA_TOP's on the Target System [2] : Target System DATA_TOP Directory 1 [/d01/ oracle/VISCDB] : / rdsbdbdata/db/pdb/ VISCDB_A Target System DATA_TOP Directory 2 [/d01/ora cle/data] : /rdsbdbat a/db/pdb/VISCDB_A/ datafile Specify value for OSBACKUPDBA group [database] : Specify value for OSDGDBA group [database] : Specify value for OSKMDBA group [database] : Specify value for OSRACDBA group [database] : Target System RDBMS ORACLE_HOME Directory [/d01/oracle/19.0. 0] : /rdsdbbin/oracle </pre>	

작업	설명	필요한 기술
	<pre> Do you want to preserve the Display [:1] (y/n) : y Do you want the target system to have the same port values as the source system (y/n) [y] ? : y Validating if the source port numbers are available on the target system.. Complete port informati on available at / rdsdbbin/oracle/a ppsutil/clone/bin/ out/VIS_oebs-db01log/ portpool.lst New context path and file name [VIS_oebs -db01log.xml] : / rdsdbbin/oracle/a ppsutil/VIS_oebs-d b01log.xml Do you want to overwrite it (y/n) [n] ? : y Replacing /rdsdbbin /oracle/appsutil/V IS_oebs-db01log.xml file. The new database context file has been created : contextfile=/rdsdbbin/ oracle/appsutil/VIS_o ebs-db01log.xml Check Clone Context logfile /rdsdbbin/ oracle/appsutil/clone/ bin/CloneContext_06091 41428.log for details. </pre>	

작업	설명	필요한 기술
<p>ETCC를 설치하고 자동 구성을 실행합니다.</p>	<p>1. Oracle E-Business Suite 기술 코드 레벨 검사기(ETCC)를 설치합니다.</p> <p>My Oracle Support에서 패치 17537119를 다운로드하고 README.txt 의 지침을 따릅니다. \$ORACLE_HOME 디렉터리에 etcc라는 디렉터리를 만들고 패치의 압축을 풀어 checkMTpatch.sh 스크립트를 만든 다음 스크립트를 실행하여 패치 버전을 확인합니다.</p> <p>2. 자동 구성 유틸리티를 실행하고 새 논리적 호스트 이름 컨텍스트 파일을 전달합니다.</p> <p>Oracle 12.1.0.2의 경우:</p> <pre>cd \$ORACLE_HOME/appsu til/bin \$./adconfig.sh contextfile=/rdsdb bin/oracle.12.1.cu stom.r1.EE.1/appsu til/clone/bin/VIS_ oeb-db01log.xml</pre> <p>Oracle 19c의 경우:</p> <p>자동 구성에서는 리스너 이름이 CDBNAME과 일치할 것으로 예상합니다. 따라서 백업된 원본 리스너 구성 파일은</p>	<p>DBA</p>

작업	설명	필요한 기술
	<p>L_<CDBNAME>_001 을 일시적으로 사용합니다.</p> <pre> \$ lsnrctl stop L_VISCDB_001 \$ cp -rp /rdsdbdata/config/listener.ora /rdsdbdata/config/listener.ora_orig \$ vi /rdsdbdata/config/listener.ora :%s/L_VISCDB_001/VISCDB/g \$ lsnrctl start VISCDB \$ cd /rdsdbbin/oracle/appsutil \$. ./txkSetCfgCDB.env dboraclehome=/rdsdbbin/oracle.19.custom.r1.EE-CDB.1 Oracle Home being passed: /rdsdbbin/oracle \$ echo \$ORACLE_HOME /rdsdbbin/oracle.19.custom.r1.EE-CDB.1 \$ export ORACLE_SID=VISCDB \$ cd \$ORACLE_HOME/appsutil/bin \$ perl \$ORACLE_HOME/appsutil/bin/txkPostPDBCreationTasks.pl -dboraclehome=\$ORACLE_HOME -outdir=\$ORACLE_HOME/appsutil/log -cidsid=VISCDB -pidsid=VISCDB -appsuser </pre>	

작업	설명	필요한 기술
	<pre>=apps -dbport=1521 - servicetype=onpremise Enter the APPS Password: <apps password> Enter the CDB SYSTEM Password:<password from secrets manager></pre> <p>참고: 데이터베이스 디렉터리가 변경된 경우 Oracle Support 문서 2525754.1의 지침을 따릅니다.</p>	

Amazon RDS Custom 및 Oracle E-Business Suite에 대한 TNS 항목을 구성

작업	설명	필요한 기술
<p>Amazon RDS Custom 및 Oracle E-Business Suite에 대한 TNS 항목을 구성합니다.</p>	<p>자동 구성의 기본 위치에 TNS 파일을 생성합니다. Oracle 12.1.0.2(CDB가 아님)와 Oracle 19c PDB의 경우 기본 위치는 \$ORACLE_HOME/network/admin/\$<CONTEXT_NAME> 입니다. Oracle 19c용 CDB는 이전 단계에서 자동 구성을 실행할 때 생성된 환경 파일에 \$TNS_ADMIN 으로 정의된 기본값 \$ORACLE_HOME/network/admin/ 을 사용합니다.</p>	<p>DBA</p>

작업	설명	필요한 기술
	<p>Oracle 12.1.0.2 및 19c CDB의 경우 자동 구성으로 생성된 tnsnames.ora 및 listener.ora 파일이 공백이나 설명 없음과 같은 Amazon RDS 요구 사항을 준수하지 않기 때문에 이러한 기능을 사용하지 않습니다. 대신 Amazon RDS Custom 데이터베이스와 함께 제공된 일반 파일을 사용하여 시스템이 예상하는 내용을 준수하고 오류 범위를 줄일 수 있습니다.</p> <p>예를 들어 Amazon RDS Custom은 다음과 같은 이름 지정 형식을 예상합니다.</p> <div style="border: 1px solid #ccc; border-radius: 10px; padding: 5px; margin: 10px 0; text-align: center;">L_<INSTANCE_NAME>_001</div> <p>Oracle 12.1.0.2의 경우 다음과 같습니다.</p> <div style="border: 1px solid #ccc; border-radius: 10px; padding: 5px; margin: 10px 0; text-align: center;">L_VIS_001</div> <p>Oracle 19c의 경우 다음과 같습니다.</p> <div style="border: 1px solid #ccc; border-radius: 10px; padding: 5px; margin: 10px 0; text-align: center;">L_VISCDB_001</div> <p>사용할 listener.ora 파일의 예는 다음과 같습니다. 이는 Amazon RDS Custom 데이터베이스를 생성할 때 생성되었습니다. 지금은 이 파일을 변경</p>	

작업	설명	필요한 기술
	<p>하지 않았으므로 이 파일을 기본값으로 유지해야 합니다.</p> <p>Oracle 12.1.0.2의 경우:</p> <pre data-bbox="594 411 1029 1365"> \$ cd \$ORACLE_HOME/network/admin \$ cat listener.ora ADR_BASE_L_VIS_001=/rdsbdbdata/log/SID_LIST_L_VIS_001=(SID_LIST = (SID_DESC = (SID_NAME = VIS)(GLOBAL_DBNAME = VIS) (ORACLE_HOME = /rdsdbbin/oracle))) L_VIS_001=(DESCRIPTION_LIST = (DESCRIPTION = (ADDRESS = (PROTOCOL = TCP)(PORT = 1521) (HOST = xx.xx.xx.xx))) (DESCRIPTION = (ADDRESS = (PROTOCOL = TCP)(PORT = 1521)(HOST = 127.0.0.1)))) SUBSCRIBE_FOR_NODE_DOWN_EVENT_L_VIS_001=OFF </pre> <p>Oracle 19c의 경우: 리스너 이름 L_<INSTANCE_NAME>_001 을 사용하여 원본 listener.ora 파일을 복원합니다.</p> <pre data-bbox="594 1667 1029 1877"> \$ cd \$ORACLE_HOME/network/admin \$ cp -rp /rdsbdbdata/config/listener.ora / </pre>	

작업	설명	필요한 기술
	<pre> rdsdbdata/config/ listener.ora_autoc onfig \$ cp -rp /rdsdbdat a/config/listener. ora_orig /rdsdbdata/ config/listener.ora \$ cat listener.ora SUBSCRIBE_FOR_ NODE_DOWN_EVENT_L_ VISCDB_001=OFF ADR_BASE_L_VISCDB_001 =/rdsdbdata/log/ USE_SID_AS_SERVICE_ L_VISCDB_001=ON L_VISCDB_001=(DESCRI PTION_LIST = (DESCRIPT ION = (ADDRESS = (PROTOCOL = TCP)(PORT = 1521)(HOST = xx.xx.xx. xx))) (DESCRIPTION = (ADDRESS = (PROTOCOL = TCP)(PORT = 1521)(HOST = 127.0.0.1)))) SID_LIST_L_VISCDB_001= (SID_LIST = (SID_DESC = (SID_NAME = VISCDB)(G LOBAL_DBNAME = VISCDB) (ORACLE_HOME = / rdsdbbin/oracle))) </pre> <p>표준 Amazon RDS 작업을 위한 리스너 L_<INSTANCE_NAME>_001 을 시작합니다.</p> <pre> \$ lsnrctl stop \$ lsnrctl start L_VISCDB_001 </pre>	

작업	설명	필요한 기술
	<p>Oracle 12.1.0.2의 경우:</p> <p>Oracle E-Business Suite 환경 파일을 편집하여 Amazon RDS Custom 일반 TNS 파일을 사용할 \$TNS_ADMIN 경로를 변경합니다. 환경 파일은 이전에 자동 구성을 실행할 때 생성되었습니다. <CONTEXT_NAME> 접미사를 제거하여 TNS_ADMIN 변수를 편집합니다.</p> <p>참고: 19c의 기본 홈은 Amazon RDS Custom의 기본 홈과 동일한 \$ORACLE_HOME/network/admin 이므로 Oracle 12.1.0.2에서만 환경 파일을 편집해야 합니다.</p> <p>예를 들어 Oracle 12.1.0.2에서 다음과 같이 파일을 편집합니다.</p> <pre data-bbox="594 1287 1029 1404">\$ vi \$ORACLE_HOME/VIS_oebs-db01log.env</pre> <p>경로를 다음과 같이 변경합니다.</p> <pre data-bbox="594 1566 1029 1766">TNS_ADMIN="/rdsdbbin/oracle/network/admin/VIS_oebs-db01log" export TNS_ADMIN</pre> <p>변경 후:</p>	

작업	설명	필요한 기술
	<pre>TNS_ADMIN="/idsdbbin/oracle/network/admin" export TNS_ADMIN</pre> <p>참고: 자동 구성을 실행할 때마다 이 단계를 반복하여 올바른 TNS 파일이 사용되고 있는지 확인해야 합니다(12.1.0.2만 해당).</p> <p>Oracle 19c의 경우:</p> <ol style="list-style-type: none"> 1. 데이터베이스 계층 컨텍스트 변수 <code>s_cdb_tnsadmin</code> 의 값을 <code><ORACLE_HOME>/network/admin</code> 대신 <code><ORACLE_HOME>/network/admin/<CONTEXT_NAME></code> 으로 변경합니다. <p>참고: <code>s_db_tnsadmin</code> 컨텍스트 변수를 업데이트하지 않습니다. <code><ORACLE_HOME>/network/admin/<CONTEXT_NAME></code> 으로 놓습니다.</p> <pre>\$. \$ORACLE_HOME/VIS_obebs-db01log.env \$ vi \$CONTEXT_FILE</pre> <ol style="list-style-type: none"> 2. <code>s_cdb_tnsadmin</code> 값에 대한 변경 사항을 저장합니다. <p><code>s_db_tnsadmin</code> 및 <code>s_cdb_tnsadmin</code> 의 값은</p>	

작업	설명	필요한 기술
	<p>PDB 이름이 VIS이며 데이터베이스 노드 논리명이 oebs-db01log 인 것과 비슷해야 합니다.</p> <pre data-bbox="592 430 1031 982"> \$ grep -i tns_admin \$CONTEXT_FILE <TNS_ADMIN oa_var="s_db_tnsad min">/rdsdbbin/ora cle/network/admin/ VIS_oebs-db01log</ TNS_ADMIN> <CDB_TNS_ADMIN oa_var="s_cdb_tnsa dmin">/rdsdbbin/or acle/network/admin</ CDB_TNS_ADMIN> </pre> <p>3. 데이터베이스 계층에서 자동 구성을 실행합니다.</p> <pre data-bbox="592 1144 1031 1801"> \$. \$ORACLE_HOME/VISCD B_oebs-db01log.env \$ export ORACLE_PD B_SID=VIS \$ sqlplus "/ as sysdba" @\$ORACLE_HOME/apps util/admin/adgrant s.sql APPS \$ sqlplus "/ as sysdba" @\$ORACLE_HOME/rdms/ admin/utl1rp.sql \$. \$ORACLE_HOME/VIS_o ebs-db01log.env \$ echo \$ORACLE_SID VIS </pre>	

작업	설명	필요한 기술
	<pre>\$ cd \$ORACLE_HOME/appsu til/scripts/\$CONTE XT_NAME \$./adautocfg.sh</pre>	

작업	설명	필요한 기술
<p>rdpdb 사용자를 위한 환경을 설정합니다.</p>	<p>Oracle 19c 버전의 경우 이 단계를 건너뛰십시오.</p> <p>Oracle 12.1.0.2의 경우:</p> <p>이제 자동 구성 및 TNS 입력을 완료했으므로 rdpdb 사용자 프로필에서 환경 파일을 설정하여 로드해야 합니다.</p> <p>Oracle E-Business Suite 데이터베이스 .env 파일을 호출하도록 .bash_profile 을 업데이트합니다. 환경이 로드되도록 프로필을 업데이트해야 합니다. 이 환경 파일은 이전에 자동 구성을 실행할 때 생성되었습니다.</p> <p>다음 예제 환경 파일은 자동 구성을 실행할 때 생성됩니다.</p> <pre data-bbox="597 1192 1026 1310">. /rdpdbbin/oracle/V IS_oebs-db01log.env</pre> <p>rdpdb 사용자로 다음을 수행합니다.</p> <pre data-bbox="597 1472 1026 1839">cd \$HOME vi .bash_profile export LD_LIBRARY_PATH= \${ORACLE_HOME}/lib:\${ ORACLE_HOME}/ctx/lib export SHLIB_PATH= \${ORACLE_HOME}/lib export PATH=\$PATH: \${ORACLE_HOME}/bin</pre>	<p>DBA</p>

작업	설명	필요한 기술
	<pre data-bbox="597 212 1023 384">alias sql='rlwrap -c sqlplus / as sysdba' . \${ORACLE_HOME}/VIS _oebs-db01log.env</pre> <p data-bbox="597 422 1023 842">참고: Oracle 19c의 경우 .bash_profile 에서 CDB 환경을 로드할 필요가 없습니다. 이는 ORACLE_HOME 기본 값이 rdsdb(Oracle Home) 사용자의 기본 홈인 \$ORACLE_HOME/network/admin 기본 경로로 설정되어 있기 때문입니다.</p>	

작업	설명	필요한 기술
<p>Amazon RDS Custom을 위한 애플리케이션 및 데이터베이스를 구성합니다.</p>	<p>Oracle 12.1.0.2와 19c 모두에 대해 처음 두 단계를 완료합니다. 후속 단계는 각 버전마다 다릅니다.</p> <ol style="list-style-type: none"> 1. 애플리케이션 계층에서 /etc/hosts 를 편집하고 데이터베이스의 IP 주소를 Amazon RDS Custom IP 주소로 변경합니다. <div data-bbox="592 709 1027 911" style="border: 1px solid #ccc; border-radius: 10px; padding: 10px; margin: 10px 0;"> <pre>xx.xx.xx.xx OEBS-db01 .localdomain OEBS- db01 OEBS-db01log.local domain OEBS-db01log</pre> </div> <p>논리적 호스트 이름을 사용하기 때문에 데이터베이스 노드를 거의 순조롭게 교체할 수 있습니다.</p> <ol style="list-style-type: none"> 2. Amazon RDS Custom DB 인스턴스에서 Amazon RDS Custom DB 인스턴스를 반영하도록 소스 EC2 인스턴스에 할당된 보안 그룹을 추가하거나 수정하여 애플리케이션이 노드에 액세스할 수 있도록 합니다. <p>Oracle 12.1.0.2의 경우:</p> <ol style="list-style-type: none"> 3. 자동 구성을 실행합니다. 애플리케이션 소유자(예: applmgr)로 다음을 실행합니다. 	<p>DBA</p>

작업	설명	필요한 기술
	<pre data-bbox="592 220 1031 451">\$ cd \$INST_TOP/admin/scripts \$./adautocfg.sh AutoConfig completed successfully.</pre> <p data-bbox="592 483 1031 567">4. fnd_nodes 항목을 확인합니다.</p> <pre data-bbox="592 598 1031 1081">SQL> select node_name from apps.fnd_nodes NODE_NAME ----- ----- ----- ----- ----- AUTHENTICATION OEBS-APP01LOG OEBS-DB01LOG</pre> <p data-bbox="592 1113 1031 1249">5. 로그인할 수 있는지 확인하고 애플리케이션을 시작합니다.</p> <pre data-bbox="592 1281 1031 1365">\$./adstrtal.sh</pre> <p data-bbox="592 1396 1031 1438">Oracle 19c의 경우:</p> <p data-bbox="592 1480 1031 1564">1. PDB가 열려 있는지 확인하고 필요한 경우 엽니다.</p> <pre data-bbox="592 1638 1031 1806">SQL> show pdbs CON_ID CON_NAME OPEN MODE RESTRICTED</pre>	

작업	설명	필요한 기술
	<pre> ----- ----- ----- 2 PDB\$SEED READ ONLY NO 3 VIS MOUNTED SQL> alter session set container=vis; SQL> alter database open; SQL> alter database save state; </pre> <p>2. apps으로 연결을 테스트합니다.</p> <pre> SQL> sqlplus apps/**** @vis </pre> <p>3. 데이터베이스 계층에서 자동 구성을 실행합니다.</p> <pre> \$. \$ORACLE_HOME/VIS_o ebs-db01log.env \$ echo \$ORACLE_SID VIS \$ cd \$ORACLE_HOME/appsu til/scripts/\$CONTE XT_NAME \$./adautocfg.sh </pre> <p>4. 애플리케이션 소유자(예: applmgr)로 애플리케이션 계</p>	

작업	설명	필요한 기술
	<p>층에서 자동 구성을 실행합니다.</p> <pre data-bbox="594 327 1024 569"> \$ cd \$INST_TOP/admin/scripts \$./adautocfg.sh AutoConfig completed successfully. </pre> <p>5. <code>fnd_nodes</code> 항목을 확인합니다.</p> <pre data-bbox="594 726 1024 1203"> SQL> select node_name from apps.fnd_nodes NODE_NAME ----- ----- ----- ----- ----- AUTHENTICATION OEBS-APP01LOG OEBS-DB01LOG </pre> <p>6. 애플리케이션을 시작합니다.</p> <pre data-bbox="594 1314 1024 1392"> \$./adstrtal.sh </pre>	

마이그레이션 후 단계 수행

작업	설명	필요한 기술
<p>자동화를 재개하여 제대로 작동하는지 확인합니다.</p>	<p>다음 AWS CLI 명령을 사용하여 자동화를 재개합니다.</p> <pre data-bbox="594 1791 1024 1881"> aws rds modify-db-instance \ </pre>	<p>DBA</p>

작업	설명	필요한 기술
	<pre data-bbox="592 210 1031 388">--db-instance-identifier vis \ --automation-mode full \ 데이터베이스는 이제 Amazon RDS Custom에서 관리합니다. 예를 들어 리스너 또는 데이터베이스가 다운되면 Amazon RDS Custom 에이전트는 리스너 또는 데이터베이스를 다시 시작합니다. 이를 테스트하려면 다음과 같은 명령을 실행합니다. 리스너 중지 예제: <pre data-bbox="592 955 1031 1081">-bash-4.2\$ lsnrctl stop vis</pre> 데이터베이스 종료 예제: <pre data-bbox="592 1186 1031 1312">SQL> shutdown immediate ;</pre></pre>	

작업	설명	필요한 기술
스키마, 연결 및 유지 관리 작업을 검증합니다.	<p>마이그레이션을 완료하려면 최소한 다음 작업을 수행해야 합니다.</p> <ul style="list-style-type: none"> FS_CLONE을 실행하여 패치 파일 시스템을 동기화합니다. 스키마 통계를 수집합니다. 외부 인터페이스와 시스템이 새 Amazon RDS Custom 데이터베이스에 연결할 수 있는지 확인합니다. 백업 및 유지 관리 일정을 설정합니다. 전환을 실행하여 파일 시스템을 전환하여 AD 온라인 패치(ADOP)가 예상대로 작동하는지 확인합니다. 	DBA

문제 해결

문제	Solution
로그 파일을 삭제하려고 하면 ORA-01624 오류가 발생합니다.	<p>로그 파일을 삭제하려고 할 때 ORA-01624 오류가 발생하는 경우 다음 단계를 따릅니다.</p> <p>다음 명령을 실행하고 삭제하려는 로그 파일의 상태가 INACTIVE가 될 때까지 기다립니다. v\$log의 상태 코드에 대한 자세한 내용은 Oracle 설명서를 참조하세요. 다음은 예제 명령과 해당 출력입니다.</p> <pre>SQL> select group#, status from v\$log;</pre>

문제	Solution
	<pre> GROUP# STATUS ----- 1 ACTIVE 2 CURRENT 3 UNUSED 4 UNUSED 5 UNUSED 6 UNUSED 6 rows selected. </pre> <p>이 예제에서는 로그 파일 1이 ACTIVE이므로 이전에 추가한 첫 번째 새 로그 파일의 상태가 CURRENT가 되도록 로그 파일을 세 번 강제 전환해야 합니다.</p> <pre> SQL> alter system switch logfile; System altered. SQL> alter system switch logfile; System altered. SQL> alter system switch logfile; System altered. </pre> <p>다음 예제와 같이 삭제하려는 로그 파일이 모두 INACTIVE가 될 때까지 기다린 다음 DROP LOGFILE 명령을 실행합니다.</p> <pre> SQL> select group#, status from v\$log; GROUP# STATUS ----- 1 INACTIVE 2 INACTIVE 3 INACTIVE 4 CURRENT 5 UNUSED 6 UNUSED 6 rows selected. </pre>

문제	Solution
<p>resetlogs 로 데이터베이스를 열면 ORA-00392 오류가 발생합니다.</p>	<p>error ORA-00392: log xx of thread 1 is being cleared, operation not allowed 오류를 수신하면 다음 명령(xx를 로그 파일 번호로 바꾸기)을 실행한 다음 열린 resetlogs 를 다시 실행합니다.</p> <pre data-bbox="829 491 1507 646">SQL> alter database clear logfile group xx; SQL> alter database open resetlogs;</pre>

문제	Solution
<p>시스템 관리자 또는 애플리케이션 사용자로 애플리케이션에 연결하는 데 문제가 있습니다.</p>	<p>문제를 확인하려면 다음 SQL 쿼리를 실행합니다.</p> <pre data-bbox="829 346 1507 783"> SQL> select dbms_java.get_jdk_ version() from dual; select dbms_java.get_jdk_version() from dual ERROR at line 1: ORA-29548: Java system class reported: release of Java system classes in the database (19.0.0.0.220719 1.8) does not match that of the oracle executabl e (19.0.0.0.0 1.8) </pre> <p>근본 원인: 소스 데이터베이스에 여러 패치가 적용되었지만 Amazon RDS Custom DB_HOME이 새로 설치되었거나 CEV를 생성할 때 OJVM과 같은 필수 RSU 패치를 사용하지 않았으므로 CEV에 모든 패치가 포함되지 않았습니다. 이를 검증하려면 소스 패치 세부 정보가 \$ORACLE_HOME/sqlpatch , \$ORACLE_HOME/.patch_storage , 및 opatch -lsinventory 에 나열되어 있는지 확인합니다.</p> <p>참조: datapatch -verbose Fails with Error : "Patch xxxxxx: Archived Patch Directory Is Empty" (Doc ID 2235541.1)</p> <p>수정: 소스(\$ORACLE_HOME/sqlpatch/)에서 누락된 패치 관련 파일을 Amazon RDS Custom(\$ORACLE_HOME/sqlpatch/)으로 복사한 다음 ./datapatch -verbose 를 다시 실행합니다.</p> <p>예:</p>

문제	Solution
	<pre data-bbox="829 212 1507 365">-bash-4.2\$ cp -rp 18793246 20204035 20887355 22098146 22731026 \$ORACLE_H OME/sqlpatch/</pre> <p data-bbox="829 401 1507 485">또는 CDB 및 PDB에서 다음 명령을 실행하여 해결 방법을 사용할 수도 있습니다.</p> <pre data-bbox="829 520 1507 642">@?/javavm/install/update_javavm_db.s ql</pre> <p data-bbox="829 678 1507 716">그런 다음 PDB에서 다음 명령을 실행합니다.</p> <pre data-bbox="829 751 1507 911">sql> alter session set container=vis; @?/javavm/install/update_javav m_db.sql</pre> <p data-bbox="829 947 1507 984">이제 테스트를 다시 실행합니다.</p> <pre data-bbox="829 1020 1507 1142">SQL> select dbms_java.get_jdk_ version() from dual;</pre>

관련 리소스

- [Amazon RDS Custom 작업](#)(Amazon RDS 설명서)
- [Amazon RDS Custom for Oracle - 데이터베이스 환경의 새로운 제어 기능](#)(AWS 뉴스 블로그)
- [Oracle용 Amazon RDS Custom을 Amazon EFS와 통합](#)(AWS 데이터베이스 블로그)
- [AWS 기반 Oracle E-Business Suite 마이그레이션](#)(AWS 백서)
- [AWS 기반 Oracle E-Business Suite 아키텍처](#)(AWS 백서)
- [활성 대기 데이터베이스를 사용하여 Amazon RDS Custom에서 Oracle E-Business Suite를 위한 HA/DR 아키텍처 설정](#)(AWS 권장 가이드)

추가 정보

유지 관리 작업

Oracle E-Business Suite 데이터베이스 홈에 새로운 패치 적용

[빈 볼륨 \(/rdsdbbin\)](#) 은 업그레이드이므로 [out-of-place CEV 업그레이드](#) 중에 빈 볼륨의 콘텐츠가 삭제됩니다. 따라서 CEV를 사용하여 업그레이드를 수행하기 전에 appsutil 디렉터리 사본을 만들어야 합니다.

소스 Amazon RDS Custom 인스턴스에서 CEV를 업그레이드하기 전에 \$ORACLE_HOME/appsutil을 먼저 백업합니다.

참고: 이 예에서는 NFS 볼륨을 사용합니다. 하지만 Amazon Simple Storage Service(Amazon S3)에 대한 복사본을 대신 사용할 수 있습니다.

1. 소스 Amazon RDS Custom 인스턴스에 appsutil을 저장할 디렉터를 만듭니다.

```
$ mkdir /RMAN/appsutil.preupgrade
```

2. 압축하고 Amazon EFS 볼륨으로 복사합니다.

```
$ tar cvf /RMAN/appsutil.preupgrade appsutil
```

3. Tar 파일이 존재하는지 확인합니다.

```
$ bash-4.2$ ls -l /RMAN/appsutil.preupgrade
-rw-rw-r-- 1 rdsdb rdsdb 622981120 Feb  8 20:16 appsutil.tar
```

4. Amazon RDS 설명서의 [RDS Custom DB 인스턴스 업그레이드](#) 지침에 따라 최신 CEV(필수 CEV가 이미 생성됨)로 업그레이드합니다.

OPATCH를 사용하여 직접 패치할 수도 있습니다. Amazon RDS 설명서의 [Oracle용 RDS Custom 업그레이드 요구 사항 및 고려 사항](#) 섹션을 참조하세요.

참고: CEV 패치 프로세스 중에는 호스트 시스템의 IP 주소가 변경되지 않습니다. 이 프로세스는 out-of-place 업그레이드를 수행하고 시작 중에 새 bin 볼륨이 동일한 인스턴스에 연결됩니다.

PeopleSoft 오라클을 Amazon RDS 커스텀으로 마이그레이션

작성자: Gaurav Gupta(AWS)

환경: 프로덕션	소스: Amazon EC2	대상: Amazon RDS Custom
R 유형: 리플랫폼	워크로드: Oracle	기술: 마이그레이션; 인프라; 데이터베이스

AWS 서비스: Amazon RDS, Amazon S3, AWS Secrets Manager, Amazon EFS

요약

[PeopleSoftOracle](#)은 전사적 프로세스를 위한 전사적 자원 계획 (ERP) 솔루션입니다. PeopleSoft 클라이언트, 애플리케이션, 데이터베이스의 3계층 아키텍처를 갖추고 있습니다. PeopleSoft [아마존 관계형 데이터베이스 서비스 \(Amazon RDS\)](#) 에서 실행할 수 있습니다. 이제 기본 운영 체제에 PeopleSoft 대한 액세스를 제공하는 [Amazon RDS Custom에서도](#) 실행할 수 있습니다.

[Amazon RDS Custom for Oracle](#)은 기본 운영 체제 및 데이터베이스 환경에 액세스해야 하는 레거시, 커스텀 및 패키지 애플리케이션을 위한 관리형 데이터베이스 서비스입니다. Oracle 데이터베이스를 Amazon RDS Custom으로 마이그레이션하면 Amazon Web Services (AWS) 에서 백업 작업과 고가용성을 관리하고 사용자는 PeopleSoft 애플리케이션과 기능을 유지 관리하는 데 집중할 수 있습니다. 마이그레이션에 대해 고려해야 할 주요 요소는 AWS 권장 가이드의 [Oracle 데이터베이스 마이그레이션 전략](#)을 참조하십시오.

이 패턴은 Oracle 복구 관리자 (RMAN) 백업을 사용하여 Amazon Elastic Compute Cloud (Amazon EC2) 의 PeopleSoft 데이터베이스를 Amazon RDS Custom으로 마이그레이션하는 단계를 중점적으로 다룹니다. EC2 인스턴스와 Amazon RDS Custom 간에 [Amazon Elastic File System\(Amazon EFS\)](#) 공유 파일 시스템을 사용하지만, Amazon FSx 또는 다른 공유 드라이브도 사용할 수 있습니다. 이 패턴은 RMAN 전체 백업(레벨 0 백업이라고도 함)을 사용합니다.

사전 조건 및 제한 사항

사전 조건

- Oracle Linux 7, Oracle Linux 8, Red Hat Enterprise Linux (RHEL) 7 또는 RHEL 8이 설치된 Amazon EC2에서 실행되는 Oracle 버전 19C 소스 데이터베이스입니다. 이 패턴의 예제에서 소스 데이터베이스 이름은 FSDM092이지만 필수 사항은 아닙니다.

참고: 이 패턴은 온프레미스 Oracle 소스 데이터베이스에도 사용할 수 있습니다. 온프레미스 네트워크와 Virtual Private Cloud(VPC) 간에 적절한 네트워크 연결이 있어야 합니다.

- PeopleSoft 9.2 데모 인스턴스.
- 단일 PeopleSoft 애플리케이션 티어. 그러나 이 패턴을 여러 애플리케이션 계층에서 작동하도록 조정할 수 있습니다.
- Amazon RDS Custom은 최소 8GB의 스왑 공간으로 구성되었습니다.

제한 사항

이 패턴은 다음 구성을 지원하지 않습니다.

- 데이터베이스 ARCHIVE_LAG_TARGET 파라미터를 60~7200 범위를 벗어난 값으로 설정
- DB 인스턴스 로그 모드 비활성화(NOARCHIVELOG)
- EC2 인스턴스의 Amazon Elastic Block Store(Amazon EBS) 최적화 속성 끄기
- EC2 인스턴스에 연결된 소스 EBS 볼륨 수정
- 새 EBS 볼륨 추가 또는 볼륨 유형을 gp2에서 gp3으로 변경
- LOG_ARCHIVE_FORMAT 파라미터의 확장 형식 변경(*.arc필요)
- 제어 파일 위치 및 이름 다중화 또는 변경(/rdsdbdata/db/*DBNAME*/controlfile/control-01.ctl 필수)

이러한 구성 및 기타 지원되지 않는 구성에 대한 추가 정보는 [Amazon RDS 설명서](#)를 참조하십시오.

제품 버전

Amazon RDS Custom에서 지원하는 Oracle Database 버전 및 인스턴스 클래스에 대한 자세한 내용은 [Oracle용 Amazon RDS Custom의 요구 사항 및 제한 사항](#)을 참조하십시오.

아키텍처

대상 기술 스택

- Application Load Balancer

- Amazon EFS
- Amazon RDS Custom for Oracle
- AWS Secrets Manager
- Amazon Simple Storage Service(S3)

대상 아키텍처

다음 아키텍처 다이어그램은 AWS의 단일 [가용 영역에서](#) 실행되는 PeopleSoft 시스템을 나타냅니다. 애플리케이션 계층은 [Application Load Balancer](#)를 통해 액세스됩니다. 애플리케이션과 데이터베이스 모두 프라이빗 서브넷에 있으며, Amazon RDS Custom 및 Amazon EC2 데이터베이스 인스턴스는 Amazon EFS 공유 파일 시스템을 사용하여 RMAN 백업 파일을 저장하고 액세스합니다. Amazon S3는 사용자 지정 RDS Oracle 엔진을 생성하고 redo 로그 메타데이터를 저장하는 데 사용됩니다.

도구

도구

서비스

- [Oracle용 Amazon RDS Custom](#)은 기본 운영 체제 및 데이터베이스 환경에 액세스해야 하는 레거시, 커스텀 및 패키지 애플리케이션을 위한 관리형 데이터베이스 서비스입니다. 백업 및고가용성과 같은 데이터베이스 관리 작업을 자동화합니다.
- [Amazon Elastic File System\(Amazon EFS\)](#)은 AWS 클라우드에서 공유 파일 시스템을 생성하고 구성하는 데 도움이 됩니다. 이 패턴은 Amazon EFS 공유 파일 시스템을 사용하여 RMAN 백업 파일을 저장하고 액세스합니다.
- [AWS Secrets Manager](#)를 사용하면 코드에 하드코딩된 보안 인증 정보(암호 등)를 Secrets Manager에 대한 API 직접 호출로 바꾸어 프로그래밍 방식으로 보안 암호를 검색할 수 있습니다. 이 패턴에서는 Secrets Manager에서 데이터베이스 사용자 암호를 검색하여 RDSADMIN 및 ADMIN 사용자를 생성하고 sys 및 system 암호를 변경합니다.
- [Amazon Simple Storage Service\(S3\)](#)는 원하는 양의 데이터를 저장, 보호 및 검색하는 데 도움이 되는 클라우드 기반 객체 스토리지 서비스입니다.
- [Elastic Load Balancing\(ELB\)](#)은 들어오는 애플리케이션 또는 네트워크 트래픽을 여러 대상에 분산합니다. 예를 들어 하나 이상의 가용 영역에 있는 Amazon Elastic Compute Cloud(Amazon EC2) 인스턴스, 컨테이너, IP 주소 전반에 걸쳐 트래픽을 분산할 수 있습니다. 이 패턴에서는 Application Load Balancer를 사용합니다.

기타 도구

- Oracle Recovery Manager(RMAN)는 Oracle 데이터베이스에 대한 백업 및 복구 지원을 제공합니다. 이 패턴은 RMAN을 사용하여 Amazon RDS Custom에서 복원된 Amazon EC2의 Oracle 소스 데이터베이스의 핫 백업을 수행합니다.

모범 사례

- 데이터베이스 초기화 파라미터의 경우 Oracle 소스 데이터베이스의 spfile을 사용하는 PeopleSoft 대신 Amazon RDS 사용자 지정 DB 인스턴스에서 제공하는 표준 pfile을 사용자 지정하십시오. Amazon RDS Custom에서 읽기 전용 복제본을 생성할 때 스페이스와 주석으로 인해 문제가 발생하기 때문입니다. 데이터베이스 초기화 파라미터에 대한 자세한 내용은 Oracle 지원 노트 1100831.1([Oracle 지원](#) 계정 필요)을 참조하십시오.
- Amazon RDS Custom은 기본적으로 Oracle 자동 메모리 관리를 사용합니다. Hugesmem 커널을 사용하려는 경우 자동 공유 메모리 관리를 대신 사용하도록 Amazon RDS Custom을 구성할 수 있습니다.
- memory_max_target 파라미터는 기본적으로 활성화되어 있지 않습니다. 프레임워크는 백그라운드에서 이를 사용하여 읽기 전용 복제본을 생성합니다.
- Oracle Flashback Database를 활성화합니다. 이 기능은 장애 조치(전환 아님) 테스트 시나리오에서 대기를 복원할 때 유용합니다.

에픽

DB 인스턴스 및 파일 시스템 설정

작업	설명	필요한 기술
DB 인스턴스를 생성합니다.	Amazon RDS 콘솔에서 FSDMO92(또는 소스 데이터베이스 이름)이라는 DB 이름을 사용하여 오라클용 Amazon RDS Custom DB 인스턴스를 생성합니다. 지침은 AWS 설명서의 Amazon RDS Custom 사용	DBA

작업	설명	필요한 기술
	<p>및 Oracle용 Amazon RDS Custom - 데이터베이스 환경의 새로운 제어 기능 블로그 게시물을 참조하십시오. 이렇게 하면 데이터베이스 이름이 소스 데이터베이스와 동일한 이름으로 설정됩니다. (비워 두면 EC2 인스턴스 및 데이터베이스 이름이 ORCL로 설정됩니다.)</p>	

Amazon EC2 소스 데이터베이스의 RMAN 전체 백업 수행

작업	설명	필요한 기술
<p>백업 스크립트를 생성합니다.</p>	<p>RMAN 백업 스크립트를 생성하여 마운트한 Amazon EFS 파일 시스템에 데이터베이스를 백업합니다(다음 예제에서 /efs). 예제 코드를 사용하거나 기존 RMAN 스크립트 중 하나를 실행할 수 있습니다.</p> <pre data-bbox="592 1312 1031 1885"> #!/bin/bash Dt=`date +%Y%m%d-%H%M` BACKUP_LOG="rman-\${ORACLE_SID}-\${Dt}" export TAGDATE=`date +%Y%m%d%H%M`; LOGPATH=/u01/scripts/logs rman target / >> \$LOGPATH/rman-\${ORACLE_SID}-\${Dt} << EOF SQL "ALTER SYSTEM SWITCH LOGFILE"; </pre>	<p>DBA</p>

작업	설명	필요한 기술
	<pre> SQL "ALTER SESSION SET NLS_DATE_FORMAT='D D.MM.YYYY HH24:MI:S S''"; RUN { ALLOCATE CHANNEL ch11 TYPE DISK MAXPIECESIZE 5G; ALLOCATE CHANNEL ch12 TYPE DISK MAXPIECESIZE 5G; BACKUP AS COMPRESSED BACKUPSET FULL DATABASE FORMAT '/efs/ima n_backup/FSCM/%d_%T_ %s_%p_FULL' ; SQL "ALTER SYSTEM ARCHIVE LOG CURRENT"; BACKUP FORMAT '/efs/ rman_backup/FSCM/%d_ %T_%s_%p_ARCHIVE ' ARCHIVELOG ALL DELETE ALL INPUT ; BACKUP CURRENT CONTROLFILE FORMAT '/ efs/rman_backup/FSCM/ %d_%T_%s_%p_CONTROL' ; } EXIT; EOF </pre>	

작업	설명	필요한 기술
백업 스크립트를 실행합니다.	<p>RMAN 백업 스크립트를 실행하려면 Oracle 홈 사용자로 로그인하고 스크립트를 실행합니다.</p> <pre data-bbox="597 443 1026 604">\$ chmod a+x rman_backup.sh \$./rman_backup.sh &</pre>	DBA

작업	설명	필요한 기술
<p>오류를 확인하고 백업 파일의 이름을 기록해 둡니다.</p>	<p>RMAN 로그 파일에서 오류를 확인합니다. 모든 것이 정상인 것 같으면 다음 명령을 실행하여 제어 파일의 백업을 나열하십시오.</p> <pre data-bbox="594 489 1029 768"> RMAN> list backup of controlfile; using target database control file instead of recovery catalog </pre> <p>출력 파일의 이름을 기록합니다.</p> <pre data-bbox="594 926 1029 1854"> List of Backup Sets ===== BS Key Type LV Size Device Type Elapsed Time Completion Time ----- - ----- 12 Full 21.58M DISK 00:00:01 13-JUL-22 BP Key: 12 Status: AVAILABLE Compressed: NO Tag: TAG20220713T150155 Piece Name: / efs/rman_backup/F SCM/FSDM092_202207 13_12_1_CONTROL Control File Included: Ckp SCN: 165591599 </pre>	DBA

작업	설명	필요한 기술
	<p>85898 Ckp time: 13-JUL-22</p> <p>Amazon RDS Custom에서 데이터베이스를 복원할 때 백업 제어 파일 /efs/rman_backup/FSCM/FSDMO92_20220713_12_1_CONTROL 을 사용하게 됩니다.</p>	

소스 애플리케이션 계층 종료

작업	설명	필요한 기술
<p>애플리케이션을 종료합니다.</p>	<p>소스 응용프로그램 계층을 종료하려면 psadmin 유틸리티 또는 psadmin 명령줄 유틸리티를 사용합니다.</p> <p>1. 웹 서버를 종료하려면 다음 명령을 실행합니다.</p> <pre data-bbox="630 1255 1029 1415">psadmin -w shutdown -d "webserver domain name"</pre> <p>2. 애플리케이션 서버를 종료하려면 다음 명령을 실행합니다.</p> <pre data-bbox="630 1598 1029 1757">psadmin -c shutdown -d "application server domain name"</pre>	<p>DBA, 관리자 PeopleSoft</p>

작업	설명	필요한 기술
	<p>3. 프로세스 스케줄러를 종료하려면 다음 명령을 실행합니다.</p> <pre>psadmin -p stop -d "process scheduler domain name"</pre>	

대상 Amazon RDS Custom 데이터베이스 구성

작업	설명	필요한 기술
nfs-utils rpm 패키지를 설치합니다.	<p>이 nfs-utils rpm 패키지를 설치하려면 다음 명령을 실행합니다.</p> <pre>\$ yum install -y nfs- utils</pre>	DBA
EFS 스토리지를 마운트합니다.	<p>Amazon EFS 콘솔 페이지에서 Amazon EFS 마운트 명령을 가져옵니다. NFS(Network File System) 클라이언트를 사용하여 Amazon RDS 인스턴스에 EFS 파일 시스템을 마운트합니다.</p> <pre>sudo mount -t nfs4 -o nfsvers=4.1,rsiz= 1048576,wsiz=1048 576,hard,timeo=600 ,retrans=2,noresvp ort fs-xxxxxxxxx.efs. eu-west-1.amazonaw s.com:/ /efs</pre>	DBA

작업	설명	필요한 기술
	<pre>sudo mount -t nfs4 -o nfsvers=4.1,rsize= 1048576,wsiz=1048 576,hard,timeo=600 ,retrans=2,noresvp ort fs-xxxxxxxxx.efs. eu-west-1.amazonaw s.com:/ /efs</pre>	

스타터 데이터베이스를 삭제하고 데이터베이스 파일을 저장할 디렉토리를 생성합니다.

작업	설명	필요한 기술
<p>자동화 모드를 일시 중지합니다.</p>	<p>자동화가 RMAN 복원 작업을 방해하지 않도록 하려면 다음 단계를 진행하기 전에 Amazon RDS Custom DB 인스턴스에서 자동화 모드를 일시 중지해야 합니다.</p> <p>AWS 콘솔 또는 AWS Command Line Interface(AWS CLI) 명령을 사용하여 자동화를 일시 중지할 수 있습니다(먼저 AWS CLI를 구성했는지 확인).</p> <pre>aws rds modify-db- instance \ --db-instance-id entifier peoplesoft- fscm-92 \ --automation-mode all- paused \ --resume-full-au tomation-mode-minute 360 \</pre>	<p>DBA</p>

작업	설명	필요한 기술
	<pre data-bbox="597 205 1024 268">--region eu-west-1</pre> <p data-bbox="597 300 1024 583">일시 중지 기간을 지정할 때는 RMAN 복원을 위한 충분한 시간을 두고 있어야 합니다. 이는 소스 데이터베이스의 크기에 따라 달라지므로 360 값을 적절히 수정합니다.</p> <p data-bbox="597 615 1024 804">또한 자동화가 일시 중지된 총 시간이 데이터베이스의 백업 또는 유지 관리 기간과 겹치지 않는지 확인하십시오.</p>	

작업	설명	필요한 기술
<p>에 대한 매개변수 파일 생성 및 수정 PeopleSoft</p>	<p>pfile을 생성하고 수정하려면 Amazon RDS 사용자 지정 DB 인스턴스로 만든 표준 pfile을 사용하십시오. PeopleSoft 필요한 파라미터를 추가합니다. PeopleSoft</p> <ol style="list-style-type: none"> 다음 명령을 실행하여 rds user rdsdb로 전환합니다. <pre>\$ sudo su - rdsdb</pre> <ol style="list-style-type: none"> 시작 데이터베이스에서 SQL*Plus에 로그인하고 다음 명령을 실행하여 pfile을 생성합니다. <pre>SQL> create pfile from spfile;</pre> <p>그러면 \$ORACLE_HOME/dbs에 pfile이 생성됩니다.</p> <ol style="list-style-type: none"> 이 파일을 백업합니다. pfile을 편집하여 PeopleSoft 매개변수를 추가하거나 업데이트합니다. <pre>*._gby_hash_aggregation_enabled=false *._unnest_subquery=false *.nls_language='AMERICAN'</pre>	DBA

작업	설명	필요한 기술
	<pre> *.nls_length_semantics='CHAR' *.nls_territory='AMERICA' *.open_cursors=1000 *.db_files=1200 *.undo_tablespace='UNDOTBS1' </pre> <p>PeopleSoft 관련 매개변수는 오라클 지원 노트 1100831.1에서 확인할 수 있습니다.</p> <p>5. pfile에서 spfile 참조를 제거합니다.</p> <pre> *.spfile='/rdsdbbin/oracle/dbs/spfileFSDM092.ora' </pre>	
<p>스타터 데이터베이스를 삭제합니다.</p>	<p>기존 Amazon RDS Custom 데이터베이스를 삭제하려면 다음 코드를 사용합니다.</p> <pre> \$ sqlplus / as sysdba SQL> shutdown immediate ; SQL> startup mount exclusive restrict; SQL> drop database; SQL> exit </pre>	

작업	설명	필요한 기술
백업에서 Amazon RDS Custom 데이터베이스를 복원합니다.	<p>다음 스크립트를 사용하여 데이터베이스를 복원합니다. 스크립트는 먼저 제어 파일을 복원한 다음 EFS 마운트에 저장된 백업 조각에서 전체 데이터베이스를 복원합니다.</p> <pre data-bbox="597 537 1027 1862">#!/bin/bash Dt=`date +%Y%m%d-%H%M` BACKUP_LOG="rman-\${ORACLE_SID}-\${Dt}" export TAGDATE=`date +%Y%m%d%H%M`; LOGPATH=/rdsdbdata/scripts/logs rman target / >> \$LOGPATH/rman-\${ORACLE_SID}-\${Dt} << EOF restore controlfile from "/efs/rman_backup/FSCM/FSDM092_20220713_12_1_CONTROL"; alter database mount; run { set newname for database to '/rdsdbdata/db/FSDM092_A/datafile/%f_%b'; SET NEWNAME FOR TEMPFILE 1 TO '/rdsdbdata/db/FSDM092_A/datafile/%f_%b'; RESTORE DATABASE; SWITCH DATAFILE ALL; SWITCH TEMPFILE ALL; RECOVER DATABASE; }</pre>	DBA

작업	설명	필요한 기술
	<pre> EOF sqlplus / as sysdba >> \$LOGPATH/rman-#{OR ACLE_SID}-\$Dt<<-EOF ALTER DATABASE RENAME FILE '/u01/psoft/db/ oradata/FSDM092/redo0 1.log' TO '/rdsbdba ta/db/FSDM092_A/on lineolog/redo01.log'; ALTER DATABASE RENAME FILE '/u01/psoft/db/ oradata/FSDM092/redo0 2.log' TO '/rdsbdba ta/db/FSDM092_A/on lineolog/redo02.log'; ALTER DATABASE RENAME FILE '/u01/psoft/db/ oradata/FSDM092/redo0 3.log' TO '/rdsbdba ta/db/FSDM092_A/on lineolog/redo03.log'; alter database clear unarchived logfile group 1; alter database clear unarchived logfile group 2; alter database clear unarchived logfile group 3; alter database open resetlogs; EXIT EOF </pre>	

Secrets Manager에서 암호를 검색하고, 사용자를 생성하고, 암호를 변경합니다.

작업	설명	필요한 기술
Secrets Manager에서 암호를 검색합니다.	<p>AWS console 또는 AWS CLI를 사용하여 이 단계를 수행할 수 있습니다. 다음 단계에서는 콘솔에 대한 지침을 보여줍니다.</p> <ol style="list-style-type: none"> 1. AWS Management Console에 로그인하고 Amazon RDS 콘솔을 엽니다. 2. 탐색 창에서 데이터베이스를 선택한 후 Amazon RDS 데이터베이스를 선택합니다. 3. 구성 탭을 선택하고 인스턴스의 리소스 ID를 기록해 둡니다. 형식은 db-<code><ID></code>입니다(예: db-73GJNH LGDNZND0XNWXSECUW6LE). 4. Secrets Manager 콘솔을 엽니다. 5. do-not-delete-custom-<code><resource_id></code> 와 같은 이름을 가진 암호를 선택합니다. 여기서 resource-id 는 단계 3에서 기록해 둔 리소스 ID를 나타냅니다. 6. 보안 암호 값 검색을 선택합니다. 	DBA

작업	설명	필요한 기술
	이 암호는 sys, system, rdsadmin 및 admin사용자의 암호와 동일합니다.	

작업	설명	필요한 기술
RDSADMIN 사용자를 생성합니다.	<p>RDSADMIN은 Amazon RDS Custom DB 인스턴스를 모니터링하고 오케스트레이션하는 데이터베이스 사용자입니다. 시작 데이터베이스를 삭제하고 RMAN을 사용하여 대상 데이터베이스를 소스에서 복원했으므로 Amazon RDS Custom 모니터링이 예상대로 작동하도록 복원 작업 후에 이 사용자를 다시 생성해야 합니다. 또한 RDSADMIN 사용자를 위한 별도의 프로파일과 테이블스페이스를 생성해야 합니다.</p> <ol style="list-style-type: none"> 1. SQL 프롬프트에서 다음 명령을 입력합니다. <pre data-bbox="634 1050 1029 1644"> SQL> set echo on feedback on serverout on SQL> @?/rdbms/admin/utlpwdmg.sql SQL> ALTER PROFILE DEFAULT LIMIT FAILED_LOGIN_ ATTEMPTS UNLIMITED PASSWORD_LIFE_TIME UNLIMITED PASSWORD_VERIFY_F UNCTION NULL; </pre> <ol style="list-style-type: none"> 2. RDSADMIN 프로파일을 생성합니다. 	DBA

작업	설명	필요한 기술
	<pre> SQL> set echo on feedback on serverout on SQL> alter session set "_oracle_script"=t rue; SQL> CREATE PROFILE RDSADMIN LIMIT COMPOSITE_LIMIT UNLIMITED SESSIONS_PER_USER UNLIMITED CPU_PER_SESSION UNLIMITED CPU_PER_CALL UNLIMITED LOGICAL_READS_PER _SESSION UNLIMITED LOGICAL_READS_PER _CALL UNLIMITED IDLE_TIME UNLIMITED CONNECT_TIME UNLIMITED PRIVATE_SGA UNLIMITED FAILED_LOGIN_ATTE MPTS 10 PASSWORD_LIFE_TIME UNLIMITED PASSWORD_REUSE_TIME UNLIMITED PASSWORD_REUSE_MAX UNLIMITED PASSWORD_VERIFY_F UNCTION NULL PASSWORD_LOCK_TIME 86400/86400 PASSWORD_GRACE_TIME 604800/86400; </pre>	

작업	설명	필요한 기술
	<p>3. RDSADMIN 테이블스페이스를 생성합니다.</p> <pre data-bbox="634 331 1029 926">SQL> CREATE BIGFILE TABLESPACE rdsadmin '/rdsdbdata/db/FSD M092_A/datafile/rd sadmin.dbf' DATAFILE SIZE 7M AUTOEXTEND ON NEXT 1m LOGGING ONLINE PERMANENT BLOCKSIZE 8192 EXTENT MANAGEMEN T LOCAL AUTOALLOCATE DEFAULT NOCOMPRES S SEGMENT SPACE MANAGEMENT AUTO;</pre> <p>4. RDSADMIN 사용자를 생성합니다. RDSADMIN 암호를 이전에 Secrets Manager에서 얻은 암호로 바꿉니다.</p> <pre data-bbox="634 1157 1029 1514">SQL> CREATE USER rdsadmin IDENTIFIED BY xxxxxxxxxxxx DEFAULT TABLESPACE rdsadmin TEMPORARY TABLESPACE TEMP profile rdsadmin ;</pre> <p>5. RDSMADMIN에게 권한을 부여합니다.</p> <pre data-bbox="634 1650 1029 1776">SQL> GRANT "CONNECT" TO RDSADMIN WITH ADMIN OPTION;</pre>	

작업	설명	필요한 기술
	<pre>SQL> GRANT "RESOURCE " TO RDSADMIN WITH ADMIN OPTION; SQL> GRANT "DBA" TO RDSADMIN; SQL> GRANT "SELECT_C ATALOG_ROLE" TO RDSADMIN WITH ADMIN OPTION; SQL> GRANT ALTER SYSTEM TO RDSADMIN; SQL> GRANT UNLIMITED TABLESPACE TO RDSADMIN; SQL> GRANT SELECT ANY TABLE TO RDSADMIN; SQL> GRANT ALTER DATABASE TO RDSADMIN; SQL> GRANT ADMINISTER DATABASE TRIGGER TO RDSADMIN; SQL> GRANT ANY OBJECT PRIVILEGE TO RDSADMIN WITH ADMIN OPTION; SQL> GRANT INHERIT ANY PRIVILEGES TO RDSADMIN; SQL> ALTER USER RDSADMIN DEFAULT ROLE ALL;</pre> <p>6. Set the SYS, SYSTEM, and DBSNMP user profiles to RDSADMIN.</p> <pre>SQL> set echo on feedback on serverout on</pre>	

작업	설명	필요한 기술
<p>마스터 사용자를 생성합니다.</p>	<pre>SQL> alter user SYS profile RDSADMIN; SQL> alter user SYSTEM profile RDSADMIN; SQL> alter user DBSNMP profile RDSADMIN;</pre> <p>시작 데이터베이스를 삭제하고 RMAN을 사용하여 소스 데이터베이스에서 대상 데이터베이스를 복원했으므로 마스터 사용자를 다시 만들어야 합니다. 이 예제에서 마스터의 이름은 admin입니다.</p> <pre>SQL> create user admin identified by <password>; SQL> grant dba to admin</pre>	<p>DBA</p>
<p>시스템 암호를 변경합니다.</p>	<p>Secrets Manager에서 검색한 암호를 사용하여 시스템 암호를 변경합니다.</p> <pre>SQL> alter user sys identified by xxxxxxxxxx; SQL> alter user system identified by xxxxxxxxxx;</pre> <p>이러한 암호를 변경하지 않으면 Amazon RDS Custom이 “데이터베이스 모니터링 사용자 또는 사용자 보안 인증이 변경되었습니다.”라는 오류 메시지를 표시합니다.</p>	<p>DBA</p>

Amazon RDS 사용자 지정에 대한 TNS 항목을 구성하고 PeopleSoft

작업	설명	필요한 기술
<p>tnsnames 파일을 구성합니다.</p>	<p>애플리케이션 계층에서 데이터베이스에 연결하려면 애플리케이션 계층에서 데이터베이스에 연결할 수 있도록 tnsnames.ora 파일을 구성하십시오. 다음 예제에서는 tnsnames.ora 파일에 대한 소프트 링크가 있지만 기본적으로 파일이 비어 있는 것을 볼 수 있습니다.</p> <pre data-bbox="594 835 1027 1709"> \$ cd /rdsdbbin/oracle/network/admin \$ ls -ltr -rw-r--r-- 1 rdsdb database 1536 Feb 14 2018 shrept.lst lrwxrwxrwx 1 rdsdb database 30 Apr 5 13:19 listener.ora - > /rdsbdbdata/config/ listener.ora lrwxrwxrwx 1 rdsdb database 28 Apr 5 13:19 sqlnet.ora - > /rdsbdbdata/config/ sqlnet.ora lrwxrwxrwx 1 rdsdb database 30 Apr 5 13:19 tnsnames.ora - > /rdsbdbdata/config/ tnsnames.ora </pre> <p>1. tsnames.ora 항목을 생성합니다. Amazon RDS 자동화가 파일을 파싱하는 방</p>	DBA

작업	설명	필요한 기술
	<p>식 때문에 항목에 스페이스, 주석 또는 추가 행이 없어야 합니다. 그렇지 않으면 -replica와 같은 일부 API를 사용할 때 문제가 발생할 수 있습니다. create-db-instance-read</p> <p>2. 데이터베이스 요구 사항에 따라 포트, 호스트 및 SID를 교체합니다. PeopleSoft 다음 코드를 예로 사용합니다.</p> <pre data-bbox="634 772 1029 1245"> \$ vi tnsnames.ora FSDM092=(DESCRIPTION = (ADDRESS_LIST = (ADDRESS = (PROTOCOL = TCP)(HOST = x.x.x.x)(PORT = 1521))) (CONNECT_DATA = (SERVER = DEDICATED) (SID = FSDM092))) </pre> <p>3. PeopleSoft 데이터베이스에 연결할 수 있는지 확인하려면 다음 명령을 실행합니다.</p> <pre data-bbox="634 1430 1029 1879"> \$ tnsping FSDM092 TNS Ping Utility for Linux: Version 19.0.0.0.0 - Production on 14- JUL-2022 10:16:45 Copyright (c) 1997, 2021, Oracle. All rights reserved. </pre>	

작업	설명	필요한 기술
	<pre>Used parameter files: /rdsdbbin/oracle/network/admin/sqlnet.ora Used TNSNAMES adapter to resolve the alias Attempting to contact (DESCRIPTION = (ADDRESS_ LIST = (ADDRESS = (PROTOCOL = TCP)(HOST = x.x.x.x)(PORT = 1521))) (CONNECT_ DATA = (SERVER = DEDICATED) (SID = FSDM092))) OK (0 msec)</pre>	

spfile 소프트링크 생성

작업	설명	필요한 기술
<p>spfile 소프트링크를 생성합니다.</p>	<ol style="list-style-type: none"> 해당 위치 /rdsdbdata/admin/FSDM092/pfile 에 spfile을 생성하려면 다음 명령을 실행합니다. <pre>SQL> create spfile='/ rdsdbdata/admin/FS DM092/pfile/spfile FSDM092.ora' from pfile;</pre> \$ORACLE_HOME/dbs 로 이동하여 spfile에 대한 소프트 링크를 생성합니다. 	DBA

작업	설명	필요한 기술
	<pre data-bbox="630 210 1026 409">ln -s '/rdsdbdata/admin/FSDM092/pfile/spfileFSDM092.ora' spfileFSDM092.ora</pre> <p data-bbox="591 422 1016 598">3. 이 파일이 생성된 후 spfile 을 사용하여 데이터베이스를 종료하고 시작할 수 있습니다.</p>	

마이그레이션 후 단계 수행

작업	설명	필요한 기술
<p>스키마, 연결 및 유지 관리 작업을 검증합니다.</p>	<p>마이그레이션을 완료하려면 다음 작업을 수행합니다.</p> <ul data-bbox="591 1039 1026 1428" style="list-style-type: none"> • 스키마 통계를 수집합니다. • PeopleSoft 애플리케이션 tier가 새 Amazon RDS 사용자 지정 데이터베이스에 연결할 수 있는지 확인하십시오. • 백업 및 유지 관리 일정을 설정합니다. 	<p>DBA</p>

관련 리소스

- [Amazon RDS Custom 작업](#)
- [Oracle용 Amazon RDS Custom – 데이터베이스 환경의 새로운 제어 기능](#)(블로그 게시물)
- [Oracle용 Amazon RDS Custom을 Amazon EFS와 통합](#)(블로그 게시물)
- [Amazon RDS를 오라클 PeopleSoft 데이터베이스로 구성](#) (AWS 백서)

Oracle ROWID 기능을 AWS 기반 PostgreSQL로 마이그레이션

작성자: Rakesh Raghav(AWS) 및 Ramesh Pathuri(AWS)

환경: PoC 또는 파일럿	소스: Oracle Database	대상: AWS에서 PostgreSQL 데이터베이스
R 유형: 리플랫폼	워크로드: Oracle	기술: 마이그레이션, 데이터베이스

AWS 서비스: Amazon Aurora, Amazon RDS, AWS SCT, AWS CLI

요약

이 패턴은 Amazon Relational Database Service(Amazon RDS) for PostgreSQL, Amazon Aurora PostgreSQL-Compatible Edition 또는 Amazon Elastic Compute Cloud(Amazon EC2)에서 Oracle Database의 ROWID 가상 열 기능을 PostgreSQL 데이터베이스로 마이그레이션하는 옵션을 설명합니다.

Oracle 데이터베이스에서 ROWID 가상 열은 테이블 행의 물리적 주소입니다. 이 가상 열은 테이블에 프라이머리 키가 없더라도 행을 고유하게 식별하는 데 사용됩니다. PostgreSQL에는 ctid라는 유사한 가상 열이 있지만 ROWID로 사용할 수는 없습니다. [PostgreSQL 설명서](#)에 설명된 대로 업데이트되거나 VACUUM 프로세스가 끝날 때마다 ctid를 변경될 수 있습니다.

ROWID PostgreSQL에서 가상 열 기능을 생성할 수 있는 세 가지 방법이 있습니다.

- 테이블의 행을 식별하려면 ROWID 대신 프라이머리 키 열을 사용하십시오.
- 테이블의 논리적 프라이머리/고유 키(복합 키일 수 있음)를 사용하십시오.
- 자동 생성된 값이 있는 열을 추가하고 ROWID를 모방할 프라이머리/고유 키를 만듭니다.

이 패턴은 세 가지 구현을 모두 안내하고 각 옵션의 장단점을 설명합니다.

사전 조건 및 제한 사항

사전 조건

- 활성 상태의 AWS 계정.
- 절차적 언어/PostgreSQL(PL/PgSQL) 코딩 전문 지식
- Source Oracle Database
- Amazon RDS for PostgreSQL 또는 Aurora PostgreSQL-Compatible 클러스터 또는 PostgreSQL 데이터베이스를 호스팅하기 위한 EC2 인스턴스

제한 사항

- 이 패턴은 ROWID 기능에 대한 해결 방법을 제공합니다. PostgreSQL은 Oracle Database에서 ROWID와 동등한 기능을 제공하지 않습니다.

제품 버전

- PostgreSQL 11.9 이상

아키텍처

소스 기술 스택

- Oracle Database

대상 기술 스택

- Aurora PostgreSQL-Compatible, Amazon RDS for PostgreSQL 또는 PostgreSQL 데이터베이스가 있는 EC2 인스턴스

구현 옵션

테이블에 프라이머리 키 또는 고유 인덱스, 논리적 프라이머리 키 또는 자격 증명 속성이 있는지 여부에 따라 PostgreSQL에서 ROWID가 지원되지 않는 문제를 해결하는 세 가지 옵션이 있습니다. 프로젝트 일정, 현재 마이그레이션 단계, 애플리케이션 및 데이터베이스 코드에 대한 종속성 등에 따라 선택이 달라집니다.

옵션	설명	장점	단점
----	----	----	----

프라이머리 키 또는 고유 인덱스

Oracle 테이블에 프라이머리 키가 있는 경우 이 키의 속성을 사용하여 행을 고유하게 식별할 수 있습니다.

- 전용 데이터베이스 기능에 종속되지 않습니다.
- 프라이머리 키 필드가 인덱싱되므로 성능에 미치는 영향이 최소화됩니다.

- 자격 증명 속성으로 전환하는 데 ROWID에 의존하는 애플리케이션 및 데이터베이스 코드를 변경해야 합니다.

논리적 프라이머리/고유 키

Oracle 테이블에 논리적 프라이머리 키가 있는 경우 이 키의 속성을 사용하여 행을 고유하게 식별할 수 있습니다. 논리적 프라이머리 키는 행을 고유하게 식별할 수 있는 속성 또는 속성 집합으로 구성되지만 제약 조건을 통해 데이터베이스에 적용되지 않습니다.

- 전용 데이터베이스 기능에 종속되지 않습니다.

- 자격 증명 속성으로 전환하는 데 ROWID에 의존하는 애플리케이션 및 데이터베이스 코드를 변경해야 합니다.
- 논리적 프라이머리 키의 속성이 인덱싱되지 않을 경우 성능에 상당한 영향을 미칩니다. 그러나 고유 인덱스를 추가하여 성능 문제를 방지할 수 있습니다.

ID 속성

Oracle 테이블에 프라 이머리 키가 없는 경우 GENERATED ALWAYS AS IDENTITY로 추가 필드를 생성할 수 있습니다. 이 속성은 테이블에 데이터를 삽입할 때마다 고유한 값을 생성하므로 데이터 조작 언어(DML) 작업에서 행을 고유하게 식별하는 데 사용할 수 있습니다.

- 전용 데이터베이스 기능에 종속되지 않습니다.
- PostgreSQL 데이터베이스는 속성을 채우고 고유성을 유지합니다.

- 자격 증명 속성으로 전환하는 데 ROWID에 의존하는 애플리케이션 및 데이터베이스 코드를 변경해야 합니다.
- 추가 필드가 인덱싱되지 않을 경우 성능에 큰 영향을 미칩니다. 그러나 인덱스를 추가하여 성능 문제를 방지할 수 있습니다.

도구

- [Amazon Relational Database Service\(RDS\) for PostgreSQL](#)는 AWS Cloud에서 관계형 데이터베이스를 설정, 운영 및 규모를 조정하는 데 도움이 됩니다.
- [Amazon Aurora PostgreSQL-Compatible Edition](#)은 PostgreSQL 배포를 설정, 운영 및 규모를 조정할 수 있는 완전관리형의 ACID 준수 관계형 데이터베이스 엔진입니다.
- [AWS Command Line Interface\(AWS CLI\)](#)는 명령줄 셸에서 명령을 사용하여 AWS 서비스와 상호 작용할 수 있는 오픈 소스 도구입니다. 이 패턴에서는 AWS CLI를 사용하여 pgAdmin을 통해 SQL 명령을 실행할 수 있습니다.
- [pgAdmin](#)은 PostgreSQL을 위한 오픈 소스 관리 도구입니다. 데이터베이스 객체를 생성, 유지 관리 및 사용하는 데 도움이 되는 그래픽 인터페이스를 제공합니다.
- [AWS Schema Conversion Tool\(AWS SCT\)](#)은 소스 데이터베이스 스키마와 대부분의 사용자 지정 코드를 대상 데이터베이스와 호환되는 형식으로 자동 변환하여 이기종 데이터베이스 마이그레이션을 지원합니다.

에픽

소스 테이블 식별

작업	설명	필요한 기술
<p>ROWID 속성을 사용하는 Oracle 테이블을 식별하십시오.</p>	<p>AWS Schema Conversion Tool(AWS SCT)를 사용하여 ROWID 기능이 있는 Oracle 테이블을 식별할 수 있습니다. 자세한 내용은 AWS SCT 설명서를 참조하십시오.</p> <p>- 또는 -</p> <p>Oracle에서는 DBA_TAB_COLUMNS 뷰를 사용하여 ROWID 속성이 있는 테이블을 식별할 수 있습니다. 이러한 필드는 영숫자 10바이트 문자를 저장하는 데 사용될 수 있습니다. 사용량을 결정하고 필요한 경우 이를 VARCHAR 필드로 변환합니다.</p>	<p>DBA 또는 개발자</p>
<p>이러한 테이블을 참조하는 코드를 식별합니다.</p>	<p>AWS SCT를 사용하여 마이그레이션 평가 보고서를 생성하여 ROWID의 영향을 받는 절차를 식별합니다. 자세한 내용은 AWS SCT 설명서를 참조하십시오.</p> <p>- 또는 -</p> <p>소스 Oracle 데이터베이스에서 dba_source 테이블의 텍스트 필드를 사용하여 ROWID 기</p>	<p>DBA 또는 개발자</p>

작업	설명	필요한 기술
	능을 사용하는 객체를 식별합니다.	

프라이머리 키 사용 결정

작업	설명	필요한 기술
프라이머리 키가 없는 테이블을 식별합니다.	<p>소스 Oracle 데이터베이스에서 DBA_CONSTRAINTS 를 사용하여 프라이머리 키가 없는 테이블을 식별합니다. 이 정보는 각 테이블의 전략을 결정하는데 도움이 됩니다. 예:</p> <pre> select dt.* from dba_tables dt where not exists (select 1 from all_constraints ct where ct.owner = Dt.owner and ct.table_name = Dt.table_name and ct.constraint_type = 'P') and dt.owner = '{schema}' </pre>	DBA 또는 개발자

솔루션 식별 및 적용

작업	설명	필요한 기술
정의되거나 논리적인 프라이머리 키가 있는 테이블에 변경 사항을 적용합니다.	고유한 프라이머리 키를 사용하거나 테이블의 행을 식별하는 논리적 프라이머리 키를 사용하도록 추가 정보 섹션에 표시된 애플리케이션 및 데이터베이스 코드를 변경합니다.	DBA 또는 개발자
정의되거나 논리적인 프라이머리 키가 없는 테이블에 필드를 추가합니다.	GENERATED ALWAYS AS IDENTITY 유형의 속성을 추가합니다. 추가 정보 섹션에 표시된 애플리케이션 및 데이터베이스 코드를 변경합니다.	DBA 또는 개발자
필요한 경우 인덱스를 추가합니다.	추가 필드 또는 논리적 프라이머리 키에 인덱스를 추가하여 SQL 성능을 개선합니다.	DBA 또는 개발자

관련 리소스

- [PostgreSQL CTID](#)(PostgreSQL 설명서)
- [생성된 컬럼](#)(PostgreSQL 설명서)
- [ROWID 가상 열](#)(Oracle 설명서)

추가 정보

다음 섹션에서는 세 가지 접근 방식을 설명하는 Oracle 및 PostgreSQL 코드 예제를 제공합니다.

시나리오 1: 프라이머리 고유 키 사용

다음 예제에서는 emp_id를 프라이머리 키로 사용하여 testrowid_s1 테이블을 생성합니다.

Oracle 코드:

```

create table testrowid_s1 (emp_id integer, name varchar2(10), CONSTRAINT testrowid_pk
PRIMARY KEY (emp_id));
INSERT INTO testrowid_s1(emp_id,name) values (1,'empname1');
INSERT INTO testrowid_s1(emp_id,name) values (2,'empname2');
INSERT INTO testrowid_s1(emp_id,name) values (3,'empname3');
INSERT INTO testrowid_s1(emp_id,name) values (4,'empname4');
commit;

SELECT rowid,emp_id,name FROM testrowid_s1;
ROWID                EMP_ID NAME
-----
AAAF3pAAAAAAAAAM0AAA      1 empname1
AAAF3pAAAAAAAAAM0AAB      2 empname2
AAAF3pAAAAAAAAAM0AAC      3 empname3
AAAF3pAAAAAAAAAM0AAD      4 empname4

UPDATE testrowid_s1 SET name = 'Ramesh' WHERE rowid = 'AAAF3pAAAAAAAAAM0AAB' ;
commit;

```

```

SELECT rowid,emp_id,name FROM testrowid_s1;
ROWID                EMP_ID NAME
-----
AAAF3pAAAAAAAAAM0AAA      1 empname1
AAAF3pAAAAAAAAAM0AAB      2 Ramesh
AAAF3pAAAAAAAAAM0AAC      3 empname3
AAAF3pAAAAAAAAAM0AAD      4 empname4

```

PostgreSQL 코드:

```

CREATE TABLE public.testrowid_s1
(
    emp_id integer,
    name character varying,
    primary key (emp_id)
);

insert into public.testrowid_s1 (emp_id,name) values
(1,'empname1'),(2,'empname2'),(3,'empname3'),(4,'empname4');

select emp_id,name from testrowid_s1;
 emp_id |  name
-----+-----
       1 | empname1

```

```

2 | empname2
3 | empname3
4 | empname4

```

```
update testrowid_s1 set name = 'Ramesh' where emp_id = 2 ;
```

```
select emp_id,name from testrowid_s1;
```

```

emp_id | name
-----+-----
1 | empname1
3 | empname3
4 | empname4
2 | Ramesh

```

시나리오 2: 논리적 프라이머리 키 사용

다음 예제에서는 논리적 프라이머리 키를 emp_id로 사용하여 testrowid_s2 테이블을 생성합니다.

Oracle 코드:

```

create table testrowid_s2 (emp_id integer, name varchar2(10) );
INSERT INTO testrowid_s2(emp_id,name) values (1,'empname1');
INSERT INTO testrowid_s2(emp_id,name) values (2,'empname2');
INSERT INTO testrowid_s2(emp_id,name) values (3,'empname3');
INSERT INTO testrowid_s2(emp_id,name) values (4,'empname4');
commit;

```

```
SELECT rowid,emp_id,name FROM testrowid_s2;
```

```

ROWID          EMP_ID NAME
-----
AAAF3rAAAAAAAMeAAA      1 empname1
AAAF3rAAAAAAAMeAAB      2 empname2
AAAF3rAAAAAAAMeAAC      3 empname3
AAAF3rAAAAAAAMeAAD      4 empname4

```

```
UPDATE testrowid_s2 SET name = 'Ramesh' WHERE rowid = 'AAAF3rAAAAAAAMeAAB' ;
commit;
```

```
SELECT rowid,emp_id,name FROM testrowid_s2;
```

```

ROWID          EMP_ID NAME
-----
AAAF3rAAAAAAAMeAAA      1 empname1
AAAF3rAAAAAAAMeAAB      2 Ramesh
AAAF3rAAAAAAAMeAAC      3 empname3

```

AAAF3rAAAAAAMeAAD

4 empname4

PostgreSQL 코드:

```
CREATE TABLE public.testrowid_s2
(
    emp_id integer,
    name character varying
);

insert into public.testrowid_s2 (emp_id,name) values
(1, 'empname1'),(2, 'empname2'),(3, 'empname3'),(4, 'empname4');

select emp_id,name from testrowid_s2;
emp_id | name
-----+-----
      1 | empname1
      2 | empname2
      3 | empname3
      4 | empname4

update testrowid_s2 set name = 'Ramesh' where emp_id = 2 ;

select emp_id,name from testrowid_s2;
emp_id | name
-----+-----
      1 | empname1
      3 | empname3
      4 | empname4
      2 | Ramesh
```

시나리오 3: 자격 증명 속성 사용

다음 예제에서는 프라이머리 키가 없는 상태에서 자격 증명 속성을 사용하여 testrowid_s3 테이블을 생성합니다.

Oracle 코드:

```
create table testrowid_s3 (name varchar2(10));
INSERT INTO testrowid_s3(name) values ('empname1');
INSERT INTO testrowid_s3(name) values ('empname2');
INSERT INTO testrowid_s3(name) values ('empname3');
INSERT INTO testrowid_s3(name) values ('empname4');
```

```

commit;

SELECT rowid,name FROM testrowid_s3;
ROWID          NAME
-----
AAAF3sAAAAAAAMmAAA empname1
AAAF3sAAAAAAAMmAAB empname2
AAAF3sAAAAAAAMmAAC empname3
AAAF3sAAAAAAAMmAAD empname4

UPDATE testrowid_s3 SET name = 'Ramesh' WHERE rowid = 'AAAF3sAAAAAAAMmAAB' ;
commit;

SELECT rowid,name FROM testrowid_s3;
ROWID          NAME
-----
AAAF3sAAAAAAAMmAAA empname1
AAAF3sAAAAAAAMmAAB Ramesh
AAAF3sAAAAAAAMmAAC empname3
AAAF3sAAAAAAAMmAAD empname4

```

PostgreSQL 코드:

```

CREATE TABLE public.testrowid_s3
(
    rowid_seq bigint generated always as identity,
    name character varying
);

insert into public.testrowid_s3 (name) values
('empname1'),('empname2'),('empname3'),('empname4');

select rowid_seq,name from testrowid_s3;
 rowid_seq | name
-----+-----
          1 | empname1
          2 | empname2
          3 | empname3
          4 | empname4

update testrowid_s3 set name = 'Ramesh' where rowid_seq = 2 ;

select rowid_seq,name from testrowid_s3;

```

```
rowid_seq | name
-----+-----
      1 | empname1
      3 | empname3
      4 | empname4
      2 | Ramesh
```

Oracle Database 오류 코드를 Amazon Aurora PostgreSQL Compatible 데이터베이스로 마이그레이션

작성자: Sai Parthasaradh(AWS)와 Veeranjaneyulu Grandhi(AWS)

환경: PoC 또는 파일럿	소스: Oracle	대상: PostgreSQL
R 유형: 리플랫폼	워크로드: Oracle	기술: 마이그레이션, 데이터베이스
AWS 서비스: Amazon Aurora		

요약

이 패턴은 사전 정의된 메타데이터 테이블을 사용하여 Oracle Database 오류 코드를 [Amazon Aurora PostgreSQL-Compatible Edition](#) 데이터베이스로 마이그레이션하는 방법을 보여줍니다.

Oracle Database 오류 코드에 항상 해당 PostgreSQL 오류 코드가 있는 것은 아닙니다. 오류 코드의 이러한 차이로 인해 대상 PostgreSQL 아키텍처에서 프로시저 또는 함수의 처리 로직을 구성하기가 어려울 수 있습니다.

PL/PgSQL 프로그램에 의미 있는 소스 및 대상 데이터베이스 오류 코드를 메타데이터 테이블에 저장하여 프로세스를 단순화할 수 있습니다. 그런 다음 유효한 Oracle Database 오류 코드에 플래그를 지정하도록 테이블을 구성하고 해당하는 PostgreSQL에 매핑한 다음 나머지 프로세스 로직을 계속 진행합니다. Oracle Database 오류 코드가 메타데이터 테이블에 없는 경우 프로세스가 예외 처리되어 종료됩니다. 그런 다음, 오류 세부 정보를 수동으로 검토하고 프로그램에서 요구하는 경우 테이블에 새로운 오류 코드를 추가할 수 있습니다.

이 구성을 사용하면 Amazon Aurora PostgreSQL-Compatible 데이터베이스에서 소스 Oracle 데이터베이스와 동일한 방식으로 오류를 처리할 수 있습니다.

참고: Oracle Database 오류 코드를 올바르게 처리하도록 PostgreSQL 데이터베이스를 구성하려면 일반적으로 데이터베이스 및 애플리케이션 코드를 변경해야 합니다.

사전 조건 및 제한 사항

사전 조건

- 활성 상태의 AWS 계정
- 인스턴스 및 리스너 서비스가 가동 및 실행되고 있는 소스 Oracle Database
- 가동 및 실행 중인 Amazon Aurora PostgreSQL-Compatible 클러스터
- Oracle Database에 대한 지식
- PostgreSQL 데이터베이스에 대한 지식

아키텍처

다음 다이어그램은 데이터 오류 코드 검증 및 처리를 위한 Amazon Aurora PostgreSQL-Compatible 데이터베이스 워크플로우의 예를 보여줍니다.

이 다이어그램은 다음 워크플로우를 보여줍니다.

1. 테이블에는 Oracle Database 오류 코드 및 분류와 이에 해당하는 PostgreSQL 오류 코드 및 분류가 들어 있습니다. 테이블에는 미리 정의된 특정 오류 코드의 유효 여부를 분류하는 `valid_error` 열이 포함되어 있습니다.
2. PL/PgSQL 함수 (`func_processdata`)에서 예외가 발생하면 두 번째 PL/pgSQL 함수 (`error_validate`)가 호출됩니다.
3. `error_validate` 함수는 Oracle Database 오류 코드를 입력 인수로 받아들입니다. 그런 다음 함수는 테이블과 비교하여 들어오는 오류 코드를 확인하여 테이블에 오류가 포함되어 있는지 확인합니다.
4. Oracle Database 오류 코드가 테이블에 포함된 경우 `error_validate` 함수는 TRUE 값을 반환하고 프로세스 로직은 계속됩니다. 오류 코드가 테이블에 포함되지 않은 경우 함수는 FALSE 값을 반환하고 프로세스 로직은 예외 처리되어 종료됩니다.
5. 함수가 FALSE 값을 반환하면 애플리케이션의 기능 책임자가 수동으로 오류 세부 정보를 검토하여 유효성을 판단합니다.
6. 그러면 새로운 오류 코드가 테이블에 수동으로 추가되거나 추가되지 않습니다. 오류 코드가 유효하고 테이블에 추가된 경우 `error_validate` 함수는 다음에 예외가 발생할 때 TRUE 값을 반환합니다. 오류 코드가 유효하지 않고 예외가 발생했을 때 프로세스가 실패해야 하는 경우 오류 코드가 테이블에 추가되지 않습니다.

기술 스택

- Amazon Aurora PostgreSQL
- pgAdmin

- Oracle SQL Developer

도구

- [Amazon Aurora PostgreSQL 호환 버전](#)은 PostgreSQL 배포를 설정, 운영 및 확장할 수 있고 ACID를 준수하는 완전관리형 관계형 데이터베이스 엔진입니다.
- [pgAdmin](#)은 PostgreSQL의 오픈 소스 관리 및 개발 도구입니다. 데이터베이스 객체의 생성, 유지 관리 및 사용을 간소화하는 그래픽 인터페이스를 제공합니다.
- [Oracle SQL Developer](#)는 기존 배포와 클라우드 배포 모두에서 Oracle Database의 개발 및 관리를 간소화하는 무료 통합 개발 환경입니다.

에픽

Oracle Database 오류 코드를 Amazon Aurora PostgreSQL-Compatible 데이터베이스로 마이그레이션

작업	설명	필요한 기술
Amazon Aurora PostgreSQL-Compatible 데이터베이스에 테이블을 생성합니다.	<p>다음 PostgreSQL CREATE TABLE 명령을 실행합니다.</p> <pre>(source_error_code numeric NOT NULL, target_error_code character varying NOT NULL, valid_error character varying(1) NOT NULL);</pre>	PostgreSQL 개발자, Oracle, PostgreSQL용 RDS/Aurora
PostgreSQL 오류 코드와 해당 Oracle 오류 코드를 테이블에 추가합니다.	PostgreSQL INSERT 명령을 실행하여 필요한 오류 코드 값	PostgreSQL 개발자, Oracle, PostgreSQL용 RDS/Aurora

작업	설명	필요한 기술
	<p>을 error_codes 테이블에 추가합니다.</p> <p>PostgreSQL 오류 코드는 문자를 변경하는 데이터 유형 (SQLSTATE 값)을 사용해야 합니다. Oracle 오류 코드는 숫자 데이터 유형(SQLCODE 값)을 사용해야 합니다.</p> <p>Insert 문의 예:</p> <pre data-bbox="597 730 1026 1123">insert into error_codes values (-1817, '2007', 'Y'); insert into error_codes values (-1816, '2007', 'Y'); insert into error_codes values (-3114, '08006', 'N');</pre> <p>참고: Oracle 관련 Java 데이터베이스 연결 (JDBC) 예외를 발견한 경우 이러한 예외를 일반 데이터베이스 간 예외로 바꾸거나 PostgreSQL 관련 예외로 전환해야 합니다.</p>	

작업	설명	필요한 기술
<p>PL/PgSQL 함수를 생성하여 오류 코드를 검증합니다.</p>	<p>PostgreSQL CREATE FUNCTION 명령을 실행하여 PL/PgSQL 함수를 생성합니다. 함수가 다음을 수행하는지 확인합니다.</p> <ul style="list-style-type: none"> • 프로그램에서 발생한 Oracle 오류 코드를 승인합니다. • error_codes 테이블에 오류 코드가 있는지 확인합니다. • 오류 코드가 메타데이터 테이블에 있는지 여부에 따라 TRUE 또는 FALSE 값을 반환합니다. 	<p>PostgreSQL 개발자, Oracle, PostgreSQL용 RDS/Aurora</p>
<p>PL/PgSQL 함수에 기록된 새로운 오류 코드를 수동으로 검토합니다.</p>	<p>새로운 오류 코드를 수동으로 검토합니다.</p> <p>사용 사례에 적합한 새로운 오류 코드가 있는 경우 PostgreSQL INSERT 명령을 실행하여 error_codes 테이블에 추가합니다.</p> <p>-또는-</p> <p>새로운 오류 코드가 사용 사례에 적합하지 않은 경우 테이블에 추가하지 않습니다. 오류가 발생하면 프로세스 로직이 계속 실패하고 예외 처리되어 종료됩니다.</p>	<p>PostgreSQL 개발자, Oracle, PostgreSQL용 RDS/Aurora</p>

관련 리소스

[부록 A. PostgreSQL Error Codes](#) (PostgreSQL 설명서)

[데이터베이스 오류 메시지](#) (Oracle Database 설명서)

Redis 워크로드를 AWS의 Redis Enterprise Cloud로 마이그레이션

작성자: Antony Prasad Thevaraj(AWS) 및 Srinivas Pendyala(Redis)

환경: 프로덕션	소스: 온프레미스 (Redis 또는 기타) 데이터베이스	대상: AWS의 Redis 엔터프라이즈 클라우드
R 유형: 리플랫폼	워크로드: 오픈 소스	기술: 마이그레이션, 데이터베이스
AWS 서비스: AWS DMS, Amazon S3		

요약

이 패턴은 Amazon Web Services(AWS)의 Redis Enterprise Cloud로 Redis 워크로드를 마이그레이션하는 상위 수준의 프로세스를 설명합니다. 마이그레이션 단계를 설명하고, 사용 가능한 도구 선택에 대한 정보를 제공하며, 각 도구를 사용하기 위한 장점, 단점 및 단계를 설명합니다. 선택적으로 Redis에서 워크로드를 마이그레이션하는 데 추가 지원이 필요한 경우 Redis 전문 서비스에 문의할 수 있습니다.

온프레미스에서 Redis OSS 또는 Redis Enterprise Software를 실행하는 경우 데이터 센터에서 Redis 데이터베이스를 유지 관리하는 데 따르는 상당한 관리 오버헤드와 운영상의 복잡성에 익숙할 것입니다. 워크로드를 클라우드로 마이그레이션하면 이러한 운영 부담을 크게 줄이고 Redis에서 제공하는 완전히 호스팅되는 서비스형 데이터베이스(DBaaS)인 [Redis Enterprise Cloud](#)를 활용할 수 있습니다. 이 마이그레이션을 통해 99.999% 가용성, 아키텍처 단순성, 확장성과 같은 최신 Redis Enterprise Cloud on AWS 기능에 액세스하면서 비즈니스 민첩성을 높이고, 애플리케이션 안정성을 개선하고, 전체 비용을 절감할 수 있습니다.

금융 서비스, 소매, 의료, 게임 부문은 물론 사기 탐지, 실시간 인벤토리, 청구 처리 및 세션 관리를 위한 솔루션이 필요한 사용 사례에서도 Redis Enterprise Cloud를 사용할 수 있는 잠재적 애플리케이션이 있습니다. Redis Enterprise Cloud를 사용하여 AWS 리소스를 예를 들어 Amazon Elastic Compute Cloud(Amazon EC2) 인스턴스에서 실행되는 애플리케이션 서버 또는 AWS Lambda 서비스로 배포된 마이크로서비스에 연결할 수 있습니다.

사전 조건 및 제한 사항

가정

- 현재 클라우드로 마이그레이션할 온프레미스 데이터베이스 시스템을 운영하고 있습니다.
- 다음을 포함하여 워크로드에 대한 마이그레이션 요구 사항을 확인했습니다.
 - 데이터 일관성 요구 사항
 - 인프라 및 시스템 환경 요구 사항
 - 데이터 매핑 및 변환 요구 사항
 - 기능 테스트 요구 사항
 - 성능 테스트 요구 사항
 - 검증 요구 사항
 - 정의된 전환 전략
- 마이그레이션에 필요한 일정과 예상 비용을 평가했습니다.
- 요구 사항에는 작업 범위와 마이그레이션의 일부로 식별된 시스템 및 데이터베이스가 고려됩니다.
- RACI(책임, 권한, 자문, 정보 제공) 매트릭스를 통해 이해 관계자를 역할 및 책임과 함께 식별했습니다.
- 모든 이해 관계자로부터 필요한 계약 및 승인을 받았습니다.

비용

기존 소스 데이터베이스의 기술 사양(예: 메모리 크기, 처리량, 총 데이터 크기)에 따라 Redis 솔루션 아키텍트는 Redis Enterprise Cloud에서 대상 시스템의 크기를 조정할 수 있습니다. 일반 가격 정보는 Redis 웹사이트의 [Redis 요금](#)을 참조하십시오.

사람과 기술

마이그레이션 프로세스에는 다음과 같은 역할과 책임이 포함됩니다.

역할	설명	필요한 기술
마이그레이션 솔루션 아키텍트	마이그레이션 전략의 정의, 계획 및 구현에 대한 전문 지식을 갖춘 기술 아키텍트	소스 및 대상 시스템에 대한 기술 및 애플리케이션 수준의 이해, 클라우드 워크로드를 마이그레이션한 경험
데이터 아키텍트	다양한 데이터베이스를 위한 데이터 솔루션을 정의, 구현 및	정형 및 비정형 데이터를 위한 데이터 모델링, 기업용 데이터

	제공하는 데 폭넓은 경험을 갖춘 기술 아키텍트	베이스 구현에 대한 깊은 이해와 경험
Redis 솔루션 아키텍트	적절한 사용 사례에 맞게 최적의 크기의 Redis 클러스터를 설계하는 데 도움을 줄 수 있는 기술 아키텍트	다양한 사용 사례를 위한 Redis 솔루션 설계 및 배포에 대한 전문 지식
클라우드 솔루션 아키텍트	클라우드 솔루션, 특히 AWS에 대해 더 깊이 이해하고 있는 기술 아키텍트	클라우드용 솔루션 설계 전문 지식, 워크로드 마이그레이션 및 애플리케이션 현대화 경험
엔터프라이즈 아키텍트	조직의 기술 환경을 완벽하게 이해하고, 미래 로드맵에 대한 비전을 공유하고, 조직의 모든 팀 전반에서 표준화된 아키텍처 모범 사례를 실천하고 확립하는 기술 아키텍트	TOGAF, 기본 소프트웨어 엔지니어링 기술, 솔루션 아키텍처 및 엔터프라이즈 아키텍처 전문 지식과 같은 소프트웨어 아키텍처 인증
IT 또는 DevOps 엔지니어	인프라의 문제 모니터링, 유지 관리 작업 수행, 필요에 따른 업데이트 등 인프라 생성 및 유지 관리를 담당하는 엔지니어.	운영 체제, 네트워킹, 클라우드 컴퓨팅을 비롯한 다양한 기술에 대한 깊은 이해도, Python, Bash, Ruby와 같은 프로그래밍 언어와 Docker, Kubernetes, Ansible과 같은 도구에 익숙함

아키텍처

마이그레이션 옵션

다음 다이어그램은 온프레미스(Redis 기반 또는 기타) 데이터 소스를 AWS로 마이그레이션하기 위한 옵션을 보여줍니다. Redis 데이터베이스(RDB) 파일을 Amazon Simple Storage Service(S3)로 내보내거나, Redis 복제 기능을 사용하거나, AWS DMS를 사용하는 등 선택할 수 있는 여러 마이그레이션 도구를 보여줍니다.

1. 온프레미스 데이터 소스: MySQL, PostgreSQL, Oracle, SQL Server 또는 MariaDB와 같이 Redis를 기반으로 하지 않는 데이터베이스

2. 온프레미스 데이터 소스: Redis OSS 및 Redis Enterprise Software와 같은 Redis 프로토콜 기반 데이터베이스.
3. Redis 기반 데이터베이스에서 데이터를 마이그레이션하는 가장 간단한 방법은 RDB 파일을 내보내 AWS의 대상 Redis Enterprise Cloud로 가져오는 것입니다.
4. 또는 Redis의 복제 기능(ReplicaOf)을 사용하여 소스에서 대상으로 데이터를 마이그레이션할 수 있습니다.
5. 데이터 마이그레이션 요구 사항에 데이터 변환이 포함되는 경우 Redis 입력/출력 도구(RIOT)를 사용하여 데이터를 마이그레이션할 수 있습니다.
6. 또는 AWS Data Migration Service(AWS DMS)를 사용하여 SQL 기반 데이터베이스의 데이터를 마이그레이션할 수 있습니다.
7. 데이터를 AWS 기반 Redis Enterprise Cloud로 성공적으로 마이그레이션하려면 AWS DMS용 Virtual Private Cloud(VPC) 피어링을 사용해야 합니다.

대상 아키텍처

다음 다이어그램은 AWS 기반 Redis Enterprise Cloud의 일반적인 배포 아키텍처를 보여주고 주요 AWS 서비스와 함께 사용할 수 있는 방법을 보여줍니다.

1. AWS 기반 Redis Enterprise Cloud가 지원하는 비즈니스 애플리케이션에 연결할 수 있습니다.
2. 자신의 AWS 계정, 해당 계정 내 VPC에서 비즈니스 애플리케이션을 실행할 수 있습니다.
3. Redis Enterprise Cloud 데이터베이스 엔드포인트를 사용하여 애플리케이션에 연결할 수 있습니다. EC2 인스턴스에서 실행 중인 애플리케이션 서버, AWS Lambda 서비스로 배포된 마이크로서비스, Amazon Elastic Container Service(Amazon ECS) 애플리케이션 또는 Amazon Elastic Kubernetes Service(Amazon EKS) 애플리케이션이 그 예입니다.
4. VPC에서 실행되는 비즈니스 애플리케이션을 실행하려면 Redis Enterprise Cloud VPC에 대한 VPC 피어 연결이 필요합니다. 이를 통해 비즈니스 애플리케이션을 프라이빗 엔드포인트를 통해 안전하게 연결할 수 있습니다.
5. AWS 기반 Redis Enterprise Cloud는 AWS에서 DBaaS로 배포되는 인메모리 NoSQL 데이터베이스 플랫폼이며 Redis에서 완전히 관리합니다.
6. Redis Enterprise Cloud는 Redis에서 생성한 표준 AWS 계정의 VPC 내에 배포됩니다.
7. 보안상의 이유로 Redis Enterprise Cloud는 프라이빗 및 퍼블릭 엔드포인트 모두에서 액세스할 수 있는 프라이빗 서브넷에 배포됩니다. 클라이언트 애플리케이션을 프라이빗 엔드포인트의 Redis에

연결하는 것이 좋습니다. 퍼블릭 엔드포인트를 사용하려는 경우 [TLS를 사용](#)하여 클라이언트 애플리케이션과 Redis Enterprise Cloud 간의 데이터를 암호화하는 것이 좋습니다.

Redis 마이그레이션 방법론은 AWS 마이그레이션 방법론과 일치합니다. 이 방법론은 AWS 권장 가이드 웹사이트의 [대규모 마이그레이션을 가속화하기 위한 조직 동원에](#) 설명되어 있습니다.

자동화 및 규모 조정

마이그레이션을 위한 환경 설정 작업은 자동화 및 규모 조정을 위한 AWS 랜딩 존 및 코드형 인프라 (IaC) 템플릿을 통해 자동화할 수 있습니다. 이에 대해서는 이 패턴의 [애플리케이션](#) 섹션에서 설명합니다.

도구

데이터 마이그레이션 요구 사항에 따라 다양한 기술 옵션 중에서 선택하여 데이터를 AWS의 Redis Enterprise Cloud로 마이그레이션할 수 있습니다. 다음 표는 이러한 도구를 설명하고 비교합니다.

도구	설명	장점	단점
RDB 내보내기 및 가져오기	<p>소스(예: Redis OSS 또는 Redis Enterprise Software) 데이터베이스에서 RDB 파일 형식으로 데이터를 내보냅니다. Redis OSS 클러스터를 통해 데이터베이스를 제공하는 경우 각 마스터 샤드를 RDB로 내보냅니다.</p> <p>그런 다음 모든 RDB 파일을 한 번에 가져옵니다. 소스 데이터베이스가 OSS 클러스터를 기반으로 하지만 대상 데이터베이스가 OSS 클러스터 API를 사용하지 않는 경우 표준 Redis 클라이언트 라</p>	<ul style="list-style-type: none"> 간편함. RDB 형식의 데이터를 소스로 내보낼 수 있는 모든 Redis 기반 솔루션(Redis OSS 및 Redis Enterprise Software 포함)과 함께 작동합니다. 간단한 프로세스로 데이터 일관성을 유지합니다. 	<ul style="list-style-type: none"> 데이터 변환 요구 사항을 해결하거나 논리적 데이터베이스 병합을 지원하지 않습니다. 대규모 데이터 세트의 경우 시간이 많이 걸립니다. 델타 마이그레이션이 지원되지 않으면 다운타임이 길어질 수 있습니다.

이브러리를 사용하도록 애플리케이션 소스 코드를 변경해야 합니다.

데이터 변환 요구 사항이나 논리적 데이터베이스 병합에는 더 복잡한 프로세스가 필요하며, 이에 대해서는 이 표 뒷부분의 논리적 데이터베이스 병합에 설명되어 있습니다.

Redis 복제 기능(액티브-패시브)

Redis OSS, Enterprise Software 또는 Enterprise Cloud 데이터베이스의 데이터를 Redis Enterprise Cloud 데이터베이스로 지속적으로 복제할 수 있습니다. 초기 동기화 후에는 Redis 복제 기능(Replica0f)이 델타 마이그레이션을 수행하므로 애플리케이션 다운타임이 거의 발생하지 않습니다.

Redis 복제 기능은 액티브-패시브 방식으로 사용하기 위한 것입니다. 대상은 패시브로 간주되며 완전히 재동기화(소스 데이터베이스에서 플러시 및 동기화)됩니다. 따라서 소스와 대상 간의 전환은 다소 복잡합니다.

OSS 클러스터의 모든 마스터 샤드를 소스로 지정하여 Redis OSS 클러스터에서 표준 클러스터 Redis Enterprise Cloud 데이터베이스로 복제할 수 있습니다. 하지만 Redis 복제 기능을 사용하면 최대 32개의 소

- 연속 복제(초기 데이터 로드 후 델타)를 지원합니다.
- 다운타임이 거의 없습니다(복제 지연에 따라 다름).
- 데이터 일관성을 유지합니다.
- 한 사이트만 활성화 되도록 설계되었으므로 사이트 간 전환이 더 복잡합니다.
- OSS 클러스터에서 마이그레이션할 때 최대 32개의 마스터 샤드를 지원합니다.

스 데이터베이스를 사용할 수 있습니다.

DMS

AWS DMS를 사용하여 지원되는 모든 소스 데이터베이스의 데이터를 가동 중지 시간을 최소화하면서 대상 Redis 데이터 스토어로 마이그레이션할 수 있습니다. 자세한 내용은 AWS DMS 설명서에서 [Redis를 AWS DMS의 대상으로 사용하기](#) 섹션을 참조하십시오.

- NoSQL 및 SQL 데이터 소스의 마이그레이션을 모두 지원합니다.
- 다른 AWS 서비스와 잘 작동합니다.
- 라이브 마이그레이션 및 변경 데이터 캡처(CDC) 사용 사례를 지원합니다.
- Redis 키 값에는 %와 같은 특수 문자가 포함될 수 없습니다.
- 열 또는 필드 이름에 특수 문자가 있는 데이터는 마이그레이션할 수 없습니다.
- 대형 바이너리 객체(LOB) 모드를 지원하지 않습니다.

논리적 데이터베이스 병합

특별한 데이터베이스 병합 요구 사항에는 맞춤형 데이터 마이그레이션 솔루션이 필요할 수 있습니다. 예를 들어 Redis OSS에 4개의 논리적 데이터베이스(SELECT 0..3)가 있지만 데이터를 여러 Redis Enterprise Cloud 데이터베이스로 이동하는 대신 단일 데이터베이스 엔드포인트를 사용하는 것이 좋을 수 있습니다. Redis Enterprise는 선택 가능한 논리적 데이터베이스를 지원하지 않으므로 소스 데이터베이스의 물리적 데이터 모델을 변환해야 합니다. 예를 들어 각 데이터베이스 인덱스를 접두사에 매핑한 다음(0은 usr로, 1은 cmp로 등) 마이그레이션 스크립트 또는 추출, 전환, 적재(ETL) 도구를 사용하여 RDB 파일을 출력한 다음 대상 데이터베이스로 가져올 수 있습니다.

- 사용자 지정 스크립트를 사용하여 대상 시스템으로 마이그레이션하는 동안 데이터 세이핑을 세밀하게 제어할 수 있습니다.
- 마이그레이션을 완료하지 않기로 결정하면 롤백이 매우 어려울 수 있습니다. 특히 최신 데이터를 소스 시스템으로 롤백해야 하는 경우에는 더욱 그렇습니다.
- 일회성 마이그레이션을 위한 일회성 솔루션을 구축하는 것이 목표인 경우 구축 비용이 높을 수 있습니다.
- 마이그레이션 요구 사항이 자주 변경되면 코드, 인프라, 개발 시간 및 기타 영역에 대한 유지 관리 비용이 높을 수 있습니다.

또한 AWS에서 제공하는 다음 도구 및 서비스를 사용할 수 있습니다.

평가 및 검색 도구:

- [AWS Application Discovery Service](#)
- [마이그레이션 평가자](#)

애플리케이션 및 서버 마이그레이션 도구:

- [AWS Application Migration Service](#)

[데이터베이스 마이그레이션 도구:](#)

- [AWS Schema Conversion Tool\(AWS SCT\)](#)
- [AWS Database Migration Service\(AWS DMS\)](#)

[데이터 마이그레이션 도구:](#)

- [AWS Storage Gateway](#)
- [AWS DataSync](#)
- [AWS Direct Connect](#)
- [AWS Snowball](#)
- [Amazon Data Firehose](#)

마이그레이션 관리:

- [AWS Migration Hub](#)

AWS 파트너 솔루션

- [AWS 마이그레이션 컴피턴시 파트너](#)

에픽

검색 및 평가 작업 완료

작업	설명	필요한 기술
워크로드를 식별합니다.	<p>마이그레이션하려는 적합한 후보 워크로드를 식별합니다. 마이그레이션할 워크로드를 선택하기 전에 다음 사항을 고려하십시오.</p> <ul style="list-style-type: none"> 이 워크로드를 마이그레이션하거나 마이그레이션하지 않을 때 얻을 수 있는 비즈니스 가치는 무엇입니까? 이 워크로드가 대상 시스템으로 성공적으로 마이그레이션되지 않을 경우를 대비한 비상 계획이 있습니까? <p>비즈니스에 미치는 영향은 극대화하면서 관련된 위험은 최소화하는 워크로드를 선택하는 것이 가장 좋습니다. 전체 프로세스를 반복해서 진행하고 조금씩 마이그레이션하십시오.</p>	데이터 아키텍트, 비즈니스 챔피언, 마이그레이션 프로젝트 스폰서
데이터 소스 및 요구 사항 파악, 데이터 모델 설계.	Redis는 검색을 가속화하고 프로젝트의 마이그레이션 계획을 정의하기 위한 워크숍을 운영합니다. 이 워크숍의 일환으로 Redis 팀은 데이터 소스 및 소스 데이터 모델 요구 사항을 식별하고 Redis Enterprise Cloud에서 이러한 요구 사항을 어떻	Redis 솔루션 아키텍트

작업	설명	필요한 기술
	<p>게 리모델링할 수 있는지 분석합니다.</p> <p>Redis 마이그레이션 팀(전문 서비스)은 조직과 함께 상세한 데이터 모델 설계 연습을 수행합니다. 이 연습의 일환으로 Redis 팀은 다음을 수행합니다.</p> <ul style="list-style-type: none"> • 대상 Redis 데이터 구조를 식별합니다. • 데이터 매핑 전략을 정의합니다. • 마이그레이션 접근 방식 및 권장 사항을 문서화합니다. • 이해 관계자와 함께 데이터 모델을 검토하고 마무리합니다. 	

작업	설명	필요한 기술
<p>소스 데이터베이스의 특성을 식별합니다.</p>	<p>소스 및 대상 환경에서 사용되는 Redis 제품을 식별합니다.</p> <p>예:</p> <ul style="list-style-type: none"> • 소스 데이터베이스는 OSS 클러스터 데이터베이스입니까, 독립형 Redis 데이터베이스입니까? 아니면 Redis Enterprise 데이터베이스입니까? • 대상 데이터베이스는 Redis Enterprise 표준 데이터베이스입니까? 아니면 OSS 클러스터 호환 데이터베이스입니까? • 애플리케이션 소스 코드와 관련된 영향은 무엇입니까? 	<p>데이터 아키텍트</p>
<p>현재 시스템 SLA 및 기타 크기 측정 지표를 수집합니다.</p>	<p>처리량(초당 작업 수), 지연 시간, 데이터베이스당 전체 메모리 크기, 고가용성(HA) 요구 사항 등으로 표현된 현재 서비스 수준에 관한 계약(SLA)을 결정합니다.</p>	<p>데이터 아키텍트</p>

작업	설명	필요한 기술
대상 시스템의 특성을 식별합니다.	<p>다음 질문에 대한 답을 결정하십시오.</p> <ul style="list-style-type: none"> • 마이그레이션해야 하는 데이터의 양은 얼마나 됩니까? • 주어진 양의 데이터를 마이그레이션하는 데 얼마나 걸리나요? • 마이그레이션에 필요한 다운타임 요구 사항은 무엇입니까? 특정 기간 동안 서비스 또는 애플리케이션을 사용할 수 없어도 괜찮습니까? 그렇다면 얼마나 오래 걸립니까? • 마이그레이션된 데이터의 일관성은 어느 정도여야 합니까? 대상 데이터베이스가 약간 일관성이 없는 (오래된) 상태일 수 있습니까? • 대상 데이터베이스에 로드하기 전에 데이터를 변환해야 합니까? (예를 들어, 마이그레이션 전에 선택 가능한 DB 인덱스를 접두사로 변환하려고 할 수 있습니다.) • 대상 데이터베이스의 호스트 (예: 피어 VPC 또는 암호화를 사용하는 퍼블릭 엔드포인트)에서 소스 데이터베이스에 연결할 수 있습니까? • Redis 기술 아키텍트와 함께 데이터 크기 조정 및 Redis 클러스터 크기 조정 연습을 완료합니다. 	데이터 아키텍트, Redis 솔루션 아키텍트(선택 사항)

작업	설명	필요한 기술
	<ul style="list-style-type: none"> • 네트워킹 요구 사항, 인프라 요구 사항, 소프트웨어 버전 및 소프트웨어 라이선스를 식별하고 마이그레이션 전에 구성 요소를 조달하십시오. • 이 데이터 전송과 관련된 보안 문제가 있습니까? 	
<p>종속성을 파악합니다.</p>	<p>마이그레이션할 현재 시스템의 업스트림 및 다운스트림 종속성을 식별합니다. 마이그레이션 작업이 다른 종속 시스템 마이그레이션과 일치하는지 확인하십시오. 예를 들어, 다른 비즈니스 애플리케이션을 온프레미스에서 AWS 클라우드로 마이그레이션할 계획이라면 이러한 애플리케이션을 식별하고 프로젝트 목표, 일정 및 이해 관계자에 따라 조정합니다.</p>	<p>데이터 아키텍트, 엔터프라이즈 아키텍트</p>

작업	설명	필요한 기술
<p>마이그레이션 도구를 식별합니다.</p>	<p>데이터 마이그레이션 요구 사항(예: 소스 데이터 또는 다운타임 요구 사항)에 따라 앞서 도구 섹션에서 설명한 도구 중 하나를 사용할 수 있습니다. 또한 다음을 사용할 수 있습니다.</p> <ul style="list-style-type: none"> • CRDB 배포를 사용한 양방향 (액티브-액티브) 복제 • 사용자 지정 내보내기/가져오기 스크립트(예: DUMP/RESTORE 명령 사용) • RIOT, ECstats2 또는 ETL 도구와 같은 추가 내보내기/가져오기 도구 및 도우미 도구 • 테라폼 또는 AWS 템플릿과 같은 IaC 도구 CloudFormation 	<p>마이그레이션 솔루션 아키텍트, Redis 솔루션 아키텍트</p>
<p>비상 계획을 세웁니다.</p>	<p>마이그레이션 중에 문제가 발생할 경우를 대비하여 롤백할 비상 계획을 수립하십시오.</p>	<p>프로젝트 관리, 기술팀(아키텍트 포함)</p>

보안 및 규정 준수 작업 완료

작업	설명	필요한 기술
<p>Redis 관리 콘솔을 보호합니다.</p>	<p>관리 콘솔을 보호하려면 Redis 설명서의 지침을 따르십시오.</p>	<p>IT 인프라 관리자</p>
<p>Redis 데이터베이스를 보호합니다.</p>	<p>자세한 내용은 Redis 설명서의 다음 페이지를 참조하십시오.</p> <ul style="list-style-type: none"> • 역할 기반 액세스 제어 정의. 	

작업	설명	필요한 기술
	<ul style="list-style-type: none"> • 네트워크 보안 정의. • TLS 활성화. 	
안전한 Redis Cloud API.	<p>API를 활성화하면 Redis Cloud 계정의 모든 소유자에 대한 API 키를 관리할 수 있습니다. API의 보안 기능에 대한 개요는 Redis 웹사이트의 API 인증 설명서를 참조하십시오.</p>	IT 인프라 관리자

새 환경 설정

작업	설명	필요한 기술
AWS에 새 환경을 설정합니다.	<p>이 작업에는 다음이 포함됩니다.</p> <ul style="list-style-type: none"> • AWS 랜딩 존 설정 활동. 랜딩 존은 다음을 지원합니다. <ul style="list-style-type: none"> • 다중 계정 배포 • 최소 보안 기준 • 보안 기준 및 ISV 사전 요구 사항(네트워킹, 보안 구성 등)을 사용하여 새 계정을 자동으로 프로비저닝하는 방법 • 알림, 중앙 집중식 로깅 및 모니터링 • ISV 소프트웨어 구성 활동. 여기에는 제품 및 워크로드 설정과 변경 사항 등 마이그레이션에 포함해야 하는 구성이 포함됩니다. 	DevOps IT 또는 엔지니어

작업	설명	필요한 기술
	<ul style="list-style-type: none"> AWS 또는 Terraform 템플릿 구성 또는 사용자 지정과 같은 CloudFormation IaC 활동 	
<p>마이그레이션 아키텍처를 배포합니다.</p>	<ol style="list-style-type: none"> AWS에서 Redis Enterprise Cloud를 설정합니다. RIOT 또는 AWS DMS와 같은 마이그레이션 도구를 설치합니다. 사용 가능한 도구 목록은 도구 섹션을 참조하십시오. 애플리케이션, 마이그레이션 및 데이터베이스 계층 간의 연결을 설정합니다. 각 계층을 통과할 수 있는 샘플 워크로드를 만들고 소량의 샘플 데이터 세트를 마이그레이션합니다. <p>이제 실제 데이터 마이그레이션 파이프라인을 실행하고 테스트할 준비가 되었습니다.</p>	<p>IT 또는 DevOps 엔지니어</p>

네트워킹 설정

작업	설명	필요한 기술
<p>연결성 설정.</p>	<p>온프레미스 인프라와 AWS 클라우드 리소스 간의 연결을 설정합니다. 보안 그룹, AWS Direct Connect 및 기타 리소스를 사용하여 이 기능을 구현합니다. 자세한 내용은 AWS 웹사</p>	<p>IT 또는 DevOps 엔지니어</p>

작업	설명	필요한 기술
	이트에서 데이터 센터를 AWS에 연결 을 참조하십시오.	
VPC 피어링을 설정합니다.	비즈니스 애플리케이션을 실행하는 VPC(또는 마이그레이션 도구나 AWS DMS 복제 서버를 실행하는 EC2인스턴스)와 Redis Enterprise Cloud를 실행하는 VPC 간에 VPC 피어링을 설정합니다. 지침은 Amazon VPC 설명서의 Amazon VPC 시작하기 및 Redis 설명서의 VPC 피어링 활성화 를 참조하십시오.	IT 또는 DevOps 엔지니어

데이터 마이그레이션

작업	설명	필요한 기술
데이터 마이그레이션 도구를 선택합니다.	<p>도구 섹션의 테이블을 검토하여 이러한 도구에 대한 설명, 장점 및 단점을 확인하십시오.</p> <ul style="list-style-type: none"> • RDS 내보내기 및 가져오기 • Redis 복제 기능 (ReplicaOf) • DMS • 논리적 데이터베이스 병합 <p>다음 열은 각 도구와 관련된 데이터 마이그레이션 작업을 설명합니다.</p>	마이그레이션 솔루션 아키텍트

작업	설명	필요한 기술
<p>옵션 1: RDB 내보내기 및 가져오기 사용.</p>	<ol style="list-style-type: none"> 1. 소스 연결 해제: 소스 데이터베이스의 트래픽을 중지합니다(예: 비즈니스 애플리케이션 연결 해제). 2. 내보내기: 소스 데이터베이스의 데이터를 RDB 파일로 내보냅니다. 3. 단계: AWS의 Redis Enterprise Cloud 인스턴스에 액세스할 수 있는 위치에 데이터를 업로드합니다(예: 데이터를 S3 버킷 또는 FTP 서버에 업로드할 수 있음). 4. 가져오기: RDB 파일을 한 번에 모두 나열하여 Redis Enterprise Cloud 대상 데이터베이스로 가져옵니다. 5. 전환: 대상 데이터베이스로 이동합니다(예: 대상 데이터베이스에 애플리케이션 연결). <p>자세한 내용은 Redis 설명서를 참조하십시오.</p>	<p>마이그레이션 솔루션 아키텍트, Redis 솔루션 아키텍트</p>

작업	설명	필요한 기술
<p>옵션 2: Redis 복제 기능(액티브-패시브) 사용.</p>	<ol style="list-style-type: none"> 1. 데이터베이스 연결: 소스 및 대상 데이터베이스 간의 ReplicaOf 링크를 설정합니다. 2. 초기 동기화 실행: 소스 데이터베이스와 대상 데이터베이스 간의 초기 동기화가 완료될 때까지 기다립니다. 3. 소스 연결 해제: 소스 데이터베이스의 트래픽을 중지합니다(예: 애플리케이션 연결 해제). 4. 델타 복제 실행: 대상 데이터베이스에 델타가 복제될 때까지 기다립니다. 5. 전환: 대상 데이터베이스로 이동합니다(예: 애플리케이션을 대상 데이터베이스에 연결). 6. 삭제: 소스 및 대상 데이터베이스 간의 ReplicaOf 링크를 제거합니다. <p>자세한 내용은 Redis 설명서를 참조하십시오.</p>	<p>마이그레이션 솔루션 아키텍트, Redis 솔루션 아키텍트</p>

작업	설명	필요한 기술
<p>옵션 3: AWS DMS 사용.</p>	<ol style="list-style-type: none"> 1. AWS DMS 복제 인스턴스 설정: 이 인스턴스는 모든 마이그레이션 프로세스를 수행합니다. 지침: AWS DMS 설명서의 AWS DMS 복제 인스턴스 작업. 2. 소스 데이터베이스 정의: 소스 엔드포인트를 정의합니다. 소스 엔드포인트와 AWS DMS 복제 서버 간의 연결을 테스트합니다. 지침: AWS DMS 설명서의 소스 및 대상 엔드포인트 생성. 3. 대상 데이터베이스 설정: AWS 기반 Redis Enterprise Cloud를 설정하고 마이그레이션할 데이터베이스를 설정합니다. 4. 대상 데이터베이스 정의: 대상 엔드포인트를 정의합니다. AWS DMS가 실행되는 VPC와 AWS의 Redis Enterprise Cloud를 호스팅하는 VPC 간에 VPC 피어링이 설정되었는지 확인하십시오. AWS DMS 복제 서버와 대상 데이터베이스 간의 연결을 테스트합니다. 5. AWS DMS 작업 생성: 작업 또는 작업 세트를 생성하여 데이터를 마이그레이션하는 데 사용할 테이블과 복제 프로세스를 정의합니다. 지침 참조: AWS DMS 작업 사 	<p>마이그레이션 솔루션 아키텍트, Redis 솔루션 아키텍트</p>

작업	설명	필요한 기술
	<p>용은 AWS DMS 설명서를 참조하십시오.</p> <p>6. 마이그레이션: AWS DMS 작업을 실행하여 데이터를 마이그레이션합니다.</p> <p>7. 전환: 대상 데이터베이스로 이동합니다(예: 애플리케이션을 대상 데이터베이스에 연결).</p>	
<p>옵션 4: 논리적 데이터베이스 병합 사용.</p>	<p>이 옵션에는 소스 데이터베이스의 물리적 데이터 모델을 변환하고 RDB 파일을 생성할 수 있는 마이그레이션 스크립트 또는 ETL 도구를 사용하는 것이 포함됩니다. 필요한 경우 Redis 전문 서비스가 이 단계를 도와드릴 수 있습니다.</p>	<p>마이그레이션 솔루션 아키텍트, Redis 솔루션 아키텍트</p>

애플리케이션 마이그레이션

작업	설명	필요한 기술
<p>프로젝트 관리 일정과 목표를 조정합니다.</p>	<p>응용 계층의 마이그레이션 프로젝트 목표, 마일스톤, 일정을 Redis 데이터 마이그레이션 프로젝트의 목표와 일치시킵니다.</p>	<p>프로젝트 관리</p>
<p>테스트 활동을 조정합니다.</p>	<p>응용 계층을 AWS 클라우드로 마이그레이션 및 현대화한 후에는 새로 마이그레이션된 AWS 기반 Redis Enterprise</p>	<p>테스트</p>

작업	설명	필요한 기술
	Cloud에 응용 계층을 연결하여 테스트하십시오.	

테스트

작업	설명	필요한 기술
테스트 계획을 구현합니다.	구현 단계에서 개발된 데이터 마이그레이션 루틴과 스크립트를 테스트 요구 사항에 따라 사이트의 테스트 환경에서 실행합니다.	테스트
데이터 품질을 테스트합니다.	데이터를 마이그레이션한 후 데이터 품질을 테스트합니다.	테스트
기능을 테스트합니다.	데이터 쿼리와 응용 계층을 테스트하여 애플리케이션이 소스 시스템에서와 동일한 수준에서 작동하는지 확인합니다.	테스트

전환

작업	설명	필요한 기술
전환 결정을 내립니다.	모든 애플리케이션 수준 및 데이터베이스 수준 테스트를 완료한 후 경영진과 이해 관계자는 테스트 팀에서 확인한 최종 결과에 따라 AWS의 새 환경으로 전환할지 여부에 대한 최종 결정을 내립니다.	프로젝트 관리, 비즈니스 챔피언
AWS 클라우드로 전환합니다.	모든 것이 제자리에 있는지 확인했으면 애플리케이션 계층	IT 또는 DevOps 엔지니어, 데이터 아키텍트, 마이그레이션

작업	설명	필요한 기술
	은 새로 마이그레이션된 데이터를 가리키고 클라이언트는 AWS의 새로운 Redis Enterprise Cloud 시스템을 기반으로 실행되는 새 애플리케이션 계층을 가리킵니다.	솔루션 아키텍트, Redis 솔루션 아키텍트

관련 리소스

Redis 리소스

- [Redis Enterprise Cloud 설명서](#)
- [RIOT 툴 \(GitHub 리포지토리\)](#)
- [Terraform Provider\(다운로드\)](#)

AWS 리소스

- [데모 마이그레이션](#)
- [AWS 파트너 솔루션](#)
- [설명서](#)
- [블로그 게시물](#)
- [백서](#)
- [튜토리얼 및 동영상](#)
- [AWS 클라우드 마이그레이션](#)
- [AWS 권장 가이드](#)

추가 정보

Redis 워크로드를 AWS 클라우드로 마이그레이션하기 위한 표준 보안 요구 사항은 AWS 웹 사이트의 [보안, 자격 증명 및 규정 준수에 대한 모범 사례와](#) Redis 웹 사이트의 [Redis 신뢰 센터](#)를 참조하십시오.

AWS SCT 및 AWS DMS를 사용하여 SAP ASE에 있는 Amazon EC2를 Amazon Aurora PostgreSQL-Compatible로 마이그레이션하기

작성: Amit Kumar (AWS) 및 Ankit Gupta

환경: PoC 또는 파일럿	소스: SAP ASE	대상: Aurora PostgreSQL-Compatible
R 유형: 리플랫폼	워크로드: SAP	기술: 마이그레이션, 데이터베이스
AWS 서비스: AWS DMS, AWS SCT		

요약

이 패턴은 AWS Schema Conversion Tool(AWS SCT) 및 AWS Database Migration Service(AWS DMS)를 사용하여 Amazon Elastic Compute Cloud(Amazon EC2) 인스턴스에 호스팅되는 SAP Adaptive Server Enterprise(SAP ASE) 데이터베이스를 Amazon Aurora PostgreSQL-Compatible 에디션으로 마이그레이션하는 방법을 설명합니다. 이 패턴은 저장된 객체의 데이터 정의 언어(DDL) 변환과 데이터 마이그레이션 모두에 중점을 둡니다.

Aurora PostgreSQL-Compatible은 온라인 트랜잭션 프로세싱(OLTP) 워크로드를 지원합니다. 이 관리형 서비스는 온디맨드로 자동으로 확장되는 구성을 제공합니다. 애플리케이션의 요구 사항에 따라 데이터베이스를 자동으로 시작, 종료, 확장 또는 축소할 수 있습니다. 데이터베이스 인스턴스를 관리하지 않고도 클라우드에서 데이터베이스를 실행할 수 있습니다. Aurora PostgreSQL-Compatible은 빈도가 낮거나, 간헐적이거나, 예측할 수 없는 워크로드를 위한 비용 효율적인 옵션입니다.

마이그레이션 프로세스는 다음과 같은 두 가지 주요 단계로 구성됩니다.

- AWS SCT를 사용하여 데이터베이스 스키마 변환하기
- AWS DMS를 사용하여 데이터 마이그레이션하기

두 단계에 대한 자세한 지침은 에픽 섹션에 나와 있습니다. SAP ASE 데이터베이스와 함께 AWS DMS를 사용하는 것과 관련된 문제 해결에 대한 자세한 내용은 AWS DMS 설명서의 [SAP ASE 관련 문제 해결](#)을 참고하십시오.

사전 조건 및 제한 사항

사전 조건

- 활성 상태의 AWS 계정
- 서버, 데이터베이스, 리스너 서비스가 가동되고 실행되는 EC2 인스턴스의 소스 SAP ASE 데이터베이스
- 대상 Aurora PostgreSQL-Compatible 데이터베이스

제한 사항

- 연결용 포트 번호는 5432이어야 합니다.
- [huge_pages](#) 기능은 기본적으로 켜져 있지만 수정할 수 있습니다.
- Point-in-time 복구 (PITR) 세분성은 5분입니다.
- 리전 간 복제는 현재 사용할 수 없습니다.
- Aurora 데이터베이스의 최대 스토리지 크기는 128TiB입니다.
- 최대 15개의 읽기 복제본을 생성할 수 있습니다.
- 테이블 크기 제한은 Aurora 클러스터 볼륨 크기에 의해서만 제한되므로 Aurora PostgreSQL-Compatible DB 클러스터의 최대 테이블 크기는 32TiB입니다. 테이블 디자인 모범 사례(예: 대용량 테이블 분할)를 따르는 것이 좋습니다.

제품 버전

- 소스 데이터베이스: AWS DMS는 현재 SAP ASE 15, 15.5, 15.7 및 16.x를 지원합니다. SAP ASE 버전 지원에 대한 최신 정보는 [AWS DMS 사용 설명서](#)를 참고하십시오.
- 대상 데이터베이스: PostgreSQL 9.4 이상(버전 9.x용), 10.x, 11.x, 12.x, 13.x, 14.x 지원되는 최신 PostgreSQL 버전은 [AWS DMS 사용 설명서](#)를 참고하십시오.
- Amazon Aurora 1.x 이상입니다. 최신 정보는 Aurora 설명서에서 [Aurora PostgreSQL-Compatible 릴리스 및 엔진 버전](#)을 참고하십시오.

아키텍처

소스 기술 스택

- Amazon EC2에서 실행되는 SAP ASE 데이터베이스

대상 기술 스택

- Aurora PostgreSQL-Compatible 데이터베이스

마이그레이션 아키텍처

도구

- [Amazon Aurora PostgreSQL-Compatible Edition](#)은 PostgreSQL 배포를 설정, 운영 및 확장할 수 있는 완전 관리형 ACID 호환 관계형 데이터베이스 엔진입니다.
- [AWS Schema Conversion Tool\(AWS SCT\)](#)은 소스 데이터베이스 스키마와 대부분의 사용자 지정 코드를 대상 데이터베이스와 호환되는 형식으로 자동 변환하여 이기종 데이터베이스 마이그레이션을 지원합니다.
- [AWS DMS](#)는 여러 가지 소스 및 대상 데이터베이스를 지원합니다. 자세한 내용은 AWS DMS 설명서의 [데이터 마이그레이션용 소스](#) 및 [데이터 마이그레이션용 대상](#)을 참고하십시오. 가장 종합적인 버전 및 기능 지원을 위해 최신 버전의 AWS DMS를 사용하는 것을 권장합니다.

에픽

환경 설정

작업	설명	필요한 기술
소스 EC2 인스턴스에서 네트워크 액세스를 구성합니다.	소스 SAP ASE 데이터베이스를 호스팅하는 EC2 인스턴스에 보안 그룹을 설정합니다. 지침은 Amazon EC2 설명서의 Linux 인스턴스용 Amazon EC2 보안 그룹 을 참고하십시오.	시스템 관리자
대상 Aurora PostgreSQL-Compatible DB 클러스터를 생성합니다.	대상 데이터베이스를 위한 Aurora PostgreSQL-Compati	DBA

작업	설명	필요한 기술
	<p>ble 클러스터를 설치, 구성, 실행합니다.</p> <p>자세한 내용은 Aurora 설명서의 Amazon Aurora DB 클러스터 생성을 참고하십시오.</p>	
<p>대상 DB 클러스터에 대한 인증을 설정합니다.</p>	<p>대상 데이터베이스의 보안 그룹 및 방화벽을 설정합니다.</p> <p>자세한 내용은 Aurora 설명서의 Amazon Aurora DB 클러스터 생성을 참고하십시오.</p>	<p>DBA, 시스템 관리자</p>

AWS SCT로 데이터베이스 스키마 변환

작업	설명	필요한 기술
<p>AWS SCT를 시작합니다.</p>	<p>AWS SCT 설명서의 지침에 따라 AWS SCT를 시작합니다.</p> <p>AWS SCT는 SAP ASE 소스 데이터베이스의 데이터베이스 스키마를 대상 Aurora PostgreSQL-Compatible DB 인스턴스와 호환되는 형식으로 자동 변환할 수 있는 프로젝트 기반 사용자 인터페이스를 제공합니다.</p>	<p>DBA</p>
<p>AWS SCT 엔드포인트를 생성합니다.</p>	<p>소스 SAP ASE와 대상 PostgreSQL 데이터베이스의 엔드포인트를 생성합니다.</p> <p>자세한 지침은 AWS SCT 설명서를 참조하세요.</p>	<p>DBA</p>

작업	설명	필요한 기술
평가 보고서를 생성합니다.	<p>마이그레이션을 평가하고 호환되지 않는 객체 및 기능을 감지하려면 데이터베이스 마이그레이션 평가 보고서를 생성합니다.</p> <p>자세한 지침은 AWS SCT 설명서를 참조하세요.</p>	DBA
스키마를 변환합니다.	AWS SCT 설명서 의 지침에 따라 데이터베이스 스키마를 변환합니다.	DBA
데이터베이스 객체의 유효성을 검사합니다.	<p>AWS SCT는 데이터베이스 객체를 변환할 수 없는 경우 이름 및 기타 세부 정보를 파악합니다. 사용자는 이러한 객체를 수동으로 변환해야 합니다.</p> <p>이러한 불일치 사항을 파악하려면, AWS 블로그 게시물 SAP ASE에서 Amazon RDS for PostgreSQL 또는 Amazon Aurora PostgreSQL로 마이그레이션한 후 데이터베이스 객체 검증에 나와 있는 지침을 따르십시오.</p>	DBA

AWS DMS 마이그레이션 분석

작업	설명	필요한 기술
소스 및 대상 데이터베이스 버전을 검증합니다.	SAP ASE 데이터베이스 버전에서 AWS DMS와의 호환성을 확인합니다.	DBA

작업	설명	필요한 기술
	자세한 내용은 AWS DMS 설명서의 AWS DMS용 소스 및 AWS DMS용 대상 을 참고하십시오.	
스토리지 유형 및 용량에 대한 요구 사항을 확인하십시오.	소스 데이터베이스의 크기에 따라 대상 데이터베이스의 적절한 스토리지 용량을 선택합니다.	DBA, 시스템 관리자
복제 인스턴스의 인스턴스 유형, 용량 및 기타 기능을 선택합니다.	요구 사항에 맞는 인스턴스 유형, 용량, 스토리지 기능 및 네트워크 기능을 선택합니다. 지침은 AWS DMS 설명서의 마이그레이션에 적합한 AWS DMS 복제 인스턴스 선택 을 참고하십시오.	DBA, 시스템 관리자
네트워크 액세스 보안 요구 사항을 파악하십시오.	소스 및 대상 데이터베이스의 네트워크 액세스 보안 요구 사항을 식별합니다. AWS DMS 설명서의 복제 인스턴스용 네트워크 설정 지침을 따르십시오.	DBA, 시스템 관리자

데이터 마이그레이션

작업	설명	필요한 기술
AWS DMS에서 마이그레이션 작업을 생성하여 데이터를 마이그레이션합니다.	데이터를 마이그레이션하려면 AWS DMS 설명서 의 지침을 따르세요.	DBA

작업	설명	필요한 기술
	가장 종합적인 버전 및 기능 지원을 위해 최신 버전의 AWS DMS를 사용하는 것을 권장합니다.	
데이터를 검증합니다.	데이터가 소스 데이터베이스에서 대상 데이터베이스로 정확하게 마이그레이션되었는지 검증하려면 AWS DMS 설명서에 제공된 데이터 검증 지침 을 따르십시오.	DBA

애플리케이션 마이그레이션

작업	설명	필요한 기술
애플리케이션 마이그레이션 전략을 파악합니다.	애플리케이션을 클라우드로 마이그레이션하기 위한 7가지 전략(7R) 중 하나를 선택합니다.	DBA, 앱 소유자, 시스템 관리자
애플리케이션 마이그레이션 전략을 따릅니다.	대상 데이터베이스의 DNS 연결 세부 정보 업데이트 및 동적 쿼리 업데이트를 포함하여 애플리케이션 팀이 파악한 데이터베이스 작업을 완료합니다.	DBA, 앱 소유자, 시스템 관리자

대상 데이터베이스로 전환

작업	설명	필요한 기술
애플리케이션 클라이언트를 새 인프라로 전환합니다.	대상 데이터베이스에서 소스 데이터베이스로 연결을 전환합니다.	DBA, 앱 소유자, 시스템 관리자

작업	설명	필요한 기술
	자세한 내용은 관계형 데이터베이스의 마이그레이션 전략의 컷오버 섹션을 참고하십시오.	

프로젝트 닫기

작업	설명	필요한 기술
임시 AWS 리소스를 종료합니다.	모든 마이그레이션 작업, 복제 인스턴스, 엔드포인트, 기타 AWS SCT 및 AWS DMS 리소스를 종료합니다. 자세한 내용은 the AWS DMS 설명서 를 참조하십시오.	DBA, 시스템 관리자
프로젝트 문서를 검토하고 검증하세요.	프로젝트 문서의 모든 단계를 검증하여 모든 작업이 성공적으로 완료되었는지 확인합니다.	DBA, 앱 소유자, 시스템 관리자
프로젝트 닫기	마이그레이션 프로젝트를 종료하고 피드백을 제공합니다.	DBA, 앱 소유자, 시스템 관리자

관련 리소스

참조

- [Amazon RDS에서 PostgreSQL DB 인스턴스에 대한 암호화된 연결 활성화하기](#)(AWS 권장 가이드)
- [pg_transport를 사용하여 두 개의 Amazon RDS DB 인스턴스 간에 PostgreSQL 데이터베이스 전송하기](#)(AWS 권장 가이드)
- [Amazon Aurora 요금](#)
- [Amazon Aurora PostgreSQL-Compatible 에디션의 모범 사례](#)(Amazon Aurora 설명서)
- [AWS SCT 설명서](#)

- [AWS DMS 설명서](#)
- [SAP ASE 데이터베이스를 AWS DMS용 원본으로 사용](#)

자습서 및 동영상

- [AWS Database Migration Service 시작하기](#)
- [AWS Database Migration Service\(동영상\)](#)

ACM을 사용하여 Windows SSL 인증서를 Application Load Balancer로 마이그레이션

작성자: Chandra Sekhar Yaratha(AWS) 및 Igor Kovalchuk(AWS)

환경: 프로덕션	소스: Windows 웹 애플리케이션	타겟: Application Load Balancer
R 유형: 리플랫폼	워크로드: Microsoft	기술: 마이그레이션, 관리 및 거버넌스, 웹 및 모바일 앱
AWS 서비스: Elastic Load Balancing(ELB), Certificate Manager(ACM)		

요약

이 패턴은 Certificate Manager(ACM)를 사용하여, 온프레미스 서버에 호스팅된 웹 사이트 또는 Microsoft Internet Information Services(IIS)의 Amazon Elastic Compute Cloud(Amazon EC2) 인스턴스로부터 기존의 Secure Sockets Layer(SSL) 인증서를 마이그레이션하는 지침을 제공합니다. 그런 다음 Elastic Load Balancing과 함께 SSL 인증서를 사용할 수 있습니다.

SSL은 데이터를 보호하고, ID를 확인하며, 더 나은 검색 엔진 순위를 제공하고, 지불 카드 산업 데이터 보안 표준(PCI DSS) 요구 사항을 충족하는 데 도움이 되며, 고객 신뢰를 향상시킵니다. 이러한 워크로드를 관리하는 개발자와 IT 팀은 IIS 서버 및 Windows 서버를 비롯한 웹 애플리케이션과 인프라가 기본 정책에 계속 부합하기를 바랍니다.

이 패턴에는 Microsoft IIS에서 기존 SSL 인증서를 수동으로 내보내고, 이를 개인 정보 교환(PFX) 형식에서 ACM이 지원하는 Private Enhanced Mail(PEM) 형식으로 변환한 다음, 계정의 ACM으로 가져오는 작업이 포함됩니다. 또한 애플리케이션용 Application Load Balancer를 생성하고 가져온 인증서를 사용하도록 Application Load Balancer를 구성하는 방법도 설명합니다. 그러면 Application Load Balancer에서 HTTPS 연결이 종료되므로 웹 서버에서 추가 구성 오버헤드가 필요하지 않습니다. 자세한 내용은 [Application Load Balancer에 대한 HTTPS 리스너 생성](#)을 참조하세요.

Windows 서버는 .pfx 또는 .p12 파일을 사용하여 퍼블릭 키 파일(SSL 인증서)과 고유한 프라이빗 키 파일을 포함합니다. 인증 기관(CA)은 퍼블릭 키 파일을 제공합니다. 서버를 사용하여, 인증서 서명 요청(CSR)이 생성된 곳에서 관련 프라이빗 키 파일을 생성합니다.

사전 조건 및 제한 사항

사전 조건

- 활성 상태의 AWS 계정.
- 대상에서 사용하는 각 가용 영역에 하나 이상의 프라이빗 서브넷과 하나 이상의 퍼블릭 서브넷이 있는 Virtual Private Cloud(VPC)
- Windows 서버 2012 이상에서 실행되는 IIS 버전 8.0 이상
- IIS에서 실행되는 웹 애플리케이션
- IIS 서버에 대한 관리자 액세스

아키텍처

소스 기술 스택

- 데이터가 암호화된 연결(HTTPS)을 통해 안전하게 전송되도록 SSL을 사용한 IIS 웹 서버 구현

소스 아키텍처

대상 기술 스택

- 계정의 ACM 인증서
- 가져온 인증서를 사용하도록 구성된 Application Load Balancer
- 프라이빗 서브넷의 Windows 서버 인스턴스

대상 아키텍처

도구

- [Certificate Manager\(ACM\)](#)은 웹 사이트와 애플리케이션을 보호하는 퍼블릭 및 프라이빗 SSL/TLS X.509 인증서와 키를 생성하고, 저장하고, 갱신하는 데 도움을 줍니다.
- [Elastic Load Balancing\(ELB\)](#)은 들어오는 애플리케이션 또는 네트워크 트래픽을 여러 대상에 분산합니다. 예를 들어 하나 이상의 가용 영역에 있는 EC2 인스턴스, 컨테이너, IP 주소 전반적으로 트래픽을 분산할 수 있습니다.

모범 사례

- HTTP에서 HTTPS로 리디렉션을 적용합니다.
- 특정 포트로의 인바운드 트래픽만 허용하도록 Application Load Balancer의 보안 그룹을 적절하게 구성합니다.
- 각기 다른 가용 영역에서 EC2 인스턴스를 시작하여 고가용성을 보장합니다.
- 애플리케이션의 도메인이 IP 주소 대신 Application Load Balancer의 DNS 이름을 가리키도록 구성합니다.
- Application Load Balancer에 애플리케이션 계층 [상태 확인](#)이 구성되어 있는지 확인합니다.
- 상태 확인의 임계값을 구성합니다.
- [CloudWatchAmazon](#)을 사용하여 애플리케이션 로드 밸런서를 모니터링하십시오.

에픽

.pfx 파일 내보내기

작업	설명	필요한 기술
Windows 서버에서 .pfx 파일을 내보냅니다.	<p>Windows 서버의 온프레미스 IIS 관리자에서 .pfx 파일로서 SSL 인증서를 내보내려면:</p> <ol style="list-style-type: none"> 1. 시작, 관리, 인터넷 정보 서비스(IIS) 관리자를 선택합니다. 2. 서버 이름을 선택하고, 보안에서 서버 인증서를 두 번 클릭합니다. 3. 내보내려는 인증서를 선택한 다음 내보내기를 선택합니다. 4. 인증서 내보내기 상자에서 .pfx 파일의 위치, 경로, 이름을 선택합니다. 	시스템 관리자

작업	설명	필요한 기술
	<p>5. .pfx 파일의 암호를 지정하고 확인합니다.</p> <p>참고: .pfx 파일을 설치할 때 이 암호가 필요합니다.</p> <p>6. 확인을 선택합니다.</p> <p>이제 .pfx 파일을 지정 위치와 경로에 저장해야 합니다.</p>	

PFX로 인코딩된 인증서를 PEM 형식으로 변환합니다.

작업	설명	필요한 기술
OpenSSL 툴킷을 다운로드하여 설치합니다.	<ol style="list-style-type: none"> Shining Light Productions 웹사이트에서 Win32/Win64 OpenSSL을 다운로드하여 설치합니다. OpenSSL 바이너리의 위치를 시스템 PATH 변수에 추가하여 명령줄에서 바이너리를 사용할 수 있도록 합니다. 	시스템 관리자
PFX로 인코딩된 인증서를 PEM 형식으로 변환합니다.	<p>다음 절차는 PFX로 인코딩되고 서명된 인증서 파일을 PEM 형식의 3개 파일로 변환합니다.</p> <ul style="list-style-type: none"> cert-file.pem 에는 리소스의 SSL/TLS 인증서가 들어 있습니다. privatekey.pem 에는 암호로 보호하지 않는 인증서의 프라이빗 키가 들어 있습니다. 	시스템 관리자

작업	설명	필요한 기술
	<ul style="list-style-type: none"> ca-chain.pem 에는 CA의 루트 인증서가 들어 있습니다. <p>PFX로 인코딩된 인증서를 변환하려면:</p> <ol style="list-style-type: none"> 윈도우를 PowerShell 실행합니다. 다음 명령을 사용하여 PFX 파일로부터 인증서의 프라이빗 키를 추출합니다. 메시지가 표시되면 인증서 암호를 입력합니다. <div data-bbox="634 915 1029 1108" style="border: 1px solid #ccc; border-radius: 10px; padding: 10px; margin: 10px 0;"> <pre>openssl pkcs12 -in <filename>.pfx -nocerts -out withpw-privatekey.pem</pre> </div> <p>이 명령은 PEM으로 인코딩된 프라이빗 키 파일(이름: privatekey.pem)을 생성합니다. 메시지가 표시되면 프라이빗 키를 보호하는 패스프레이즈를 입력합니다.</p> 다음 명령을 실행하여 에 대한 패스프레이즈를 설정합니다. 메시지가 표시되면 2 단계에서 생성한 패스프레이즈를 입력합니다. <div data-bbox="634 1759 1029 1850" style="border: 1px solid #ccc; border-radius: 10px; padding: 10px; margin: 10px 0;"> <pre>openssl rsa -in withpw-privatekey.</pre> </div> 	

작업	설명	필요한 기술
	<pre data-bbox="630 205 1027 306">pem -out privatekey.pem</pre> <p data-bbox="630 338 1027 474">명령이 성공하면 “RSA 키를 작성”이라는 메시지가 표시됩니다.</p> <p data-bbox="589 495 1027 625">4. 다음 명령을 사용하여 PFX 파일의 인증서를 PEM 파일로 전송합니다.</p> <pre data-bbox="630 663 1027 863">openssl pkcs12 -in <file_name>.pfx -clcerts -nokeys -out cert-file.pem</pre> <p data-bbox="630 898 1027 1171">이 명령은 cert-file .pem 라는 이름의 PEM 인코딩된 인증서 파일을 생성합니다. 명령이 성공하면 “MAC 승인됨”이라는 메시지가 표시됩니다.</p> <p data-bbox="589 1192 1027 1423">5. PFX 파일에서 CA 체인 파일을 생성합니다. 다음 명령을 실행하여 ca-chain.pem 이라는 CA 체인 파일을 생성합니다.</p> <pre data-bbox="630 1461 1027 1661">openssl pkcs12 -in <file_name>.pfx -cacerts -nokeys -chain -out ca-chain.pem</pre> <p data-bbox="630 1696 1027 1822">명령이 성공하면 “MAC 승인됨”이라는 메시지가 표시됩니다.</p>	

인증서를 ACM으로 가져오기

작업	설명	필요한 기술
인증서 가져오기를 준비합니다.	ACM 콘솔 에서 인증서 가져오기를 선택합니다.	클라우드 관리자
인증서 본문을 입력합니다.	인증서 본문에서 가져오려는 PEM 인코딩된 인증서를 붙여넣습니다. 이 작업과 이 에픽의 다른 작업에 설명된 명령 및 절차에 대한 자세한 내용은 ACM 설명서에서 인증서 가져오기 를 참조하십시오.	클라우드 관리자
인증서 프라이빗 키를 입력합니다.	[Certificate private key]에 인증서의 퍼블릭 키와 일치하는 암호화되지 않은 PEM 인코딩 프라이빗 키를 붙여넣습니다.	클라우드 관리자
인증서 체인을 입력합니다.	인증서 체인의 경우 CertificateChain.pem 파일에 저장되어 있는 PEM 인코딩 인증서 체인을 붙여넣습니다.	클라우드 관리자
인증서 가져오기	[Review and import]를 선택합니다. 인증서에 대한 정보가 정확한지 확인한 다음 가져오기를 선택합니다.	클라우드 관리자

Application Load Balancer 생성

작업	설명	필요한 기술
로드 밸런서와 리스너를 생성하고 구성합니다.	Elastic Load Balancing 설명서 에 나와 있는 지침을 따라 대상 그룹을 구성하고, 대상을 등록하며, Application Load Balancer 및 리스너를 생성합니다. 포트 443의 경우 두 번째 리스너(HTTPS)를 추가합니다.	클라우드 관리자

문제 해결

문제	Solution
Windows에서는 OpenSSL 명령을 시스템 경로에 추가한 후에도 OpenSSL 명령을 인식하지 PowerShell 못합니다.	<p>\$env:path (을)를 점검하여, OpenSSL 바이너리의 위치가 포함되어 있는지 확인합니다.</p> <p>인식되지 않으면 다음 명령어를 다음 명령어를 실행하세요. PowerShell</p> <pre>\$env:path = \$env:path + ";C:\0penSSL-Win64\bin"</pre>

관련 리소스

ACM으로 인증서 가져오기

- [ACM 콘솔](#)
- [가져기할 인증서 및 키 형식](#)
- [인증서 가져오기](#)
- [Certificate Manager 사용 설명서](#)

Application Load Balancer 생성

- [Application Load Balancer 생성](#)
- [Application Load Balancer 사용 설명서](#)

메시지 대기열을 Microsoft Azure 서비스 버스에서 Amazon SQS로 마이그레이션

작성자: Nisha Gambhir(AWS)

R 타입: 리플랫폼	소스: Azure 서비스 버스 대기열을 사용하는 애플리케이션	대상: Amazon SQS
작성자: AWS	환경: PoC 또는 파일럿	기술: 웹 및 모바일 앱, 마이그레이션
워크로드: Microsoft	AWS 서비스: Amazon SQS	

요약

이 패턴은 Microsoft Azure Service Bus 큐 메시징 플랫폼을 사용하는 방식에서 .NET Framework 또는 .NET Core 웹 또는 콘솔 애플리케이션을 Amazon Simple Queue Service(Amazon SQS)로 마이그레이션하는 방법을 설명합니다.

애플리케이션은 메시징 서비스를 사용하여 다른 애플리케이션과 데이터를 주고 받습니다. 이러한 서비스는 클라우드에서 분리되고 확장성이 뛰어난 마이크로서비스, 분산 시스템 및 서버리스 애플리케이션을 구축하는 데 도움이 됩니다.

Azure Service Bus 대기열은 대기열 및 게시/구독 메시징을 지원하는 광범위한 Azure 메시징 인프라의 일부입니다.

Amazon SQS는 마이크로서비스와 분산 시스템, 서버리스 애플리케이션을 분리하거나 확장하기 쉽게 해 주는 완전 관리형 메시지 대기열 서비스입니다. Amazon SQS는 메시지 지향 미들웨어의 관리 및 운영과 관련된 복잡성과 오버헤드를 없애고 개발자가 작업을 차별화하는 데 집중할 수 있도록 합니다. Amazon SQS를 사용하면 메시지를 손실하거나 다른 서비스를 사용할 필요 없이 소프트웨어 구성 요소 간에 어떤 볼륨으로든 메시지를 전송, 저장 및 수신할 수 있습니다.

사전 조건 및 제한 사항

사전 조건

- 활성 상태의 AWS 계정

- Azure Service Bus 대기열을 사용하는 .NET 프레임워크 또는 .NET Core 웹 또는 콘솔 애플리케이션 (샘플 코드 첨부)

제품 버전

- .NET Framework 3.5 이상 또는 .NET Core 1.0.1, 2.0.0 이상

아키텍처

소스 기술 스택

- Azure Service Bus 대기열을 사용하여 메시지를 보내는 .NET 코어 또는 프레임워크) 웹 또는 콘솔 애플리케이션

대상 기술 스택

- Amazon SQS

도구

도구

- Microsoft Visual Studio

코드

Amazon SQS용 AWS Identity 및 Access Management(IAM) 정책을 생성하는 방법:

1. AWS Management Console에 로그인하여 <https://console.aws.amazon.com/iam/>에서 IAM 콘솔을 엽니다.
2. 왼쪽 탐색 창에서 정책을 선택한 후 정책 생성을 선택합니다.
3. JSON 탭을 선택하고 다음 코드를 붙여 넣습니다.

```
{
  "Version": "2012-10-17",
  "Statement": [
```

```

{
  "Sid": "VisualEditor0",
  "Effect": "Allow",
  "Action": [
    "sqs:DeleteMessage",
    "sqs:GetQueueUrl",
    "sqs:ChangeMessageVisibility",
    "sqs:SendMessageBatch",
    "sqs:ReceiveMessage",
    "sqs:SendMessage",
    "sqs:GetQueueAttributes",
    "sqs:ListQueueTags",
    "sqs:ListDeadLetterSourceQueues",
    "sqs:DeleteMessageBatch",
    "sqs:PurgeQueue",
    "sqs:DeleteQueue",
    "sqs:CreateQueue",
    "sqs:ChangeMessageVisibilityBatch",
    "sqs:SetQueueAttributes"
  ],
  "Resource": "arn:aws:sqs:*:<AccountId>:*"
},
{
  "Sid": "VisualEditor1",
  "Effect": "Allow",
  "Action": "sqs:ListQueues",
  "Resource": "*"
}
]
}

```

4. 정책 검토를 선택하고 이름을 입력한 다음 정책 생성을 선택합니다.

5. 새로 만든 정책을 기존 IAM 역할에 연결하거나 새 역할을 생성합니다.

에픽

AWS에서 Amazon SQS 설정

작업	설명	필요한 기술
Amazon SQS용 IAM 정책을 생성합니다.	Amazon SQS에 대한 액세스를 제공하는 IAM 정책을 생성합니다.	시스템 엔지니어

작업	설명	필요한 기술
	다. 샘플 정책에 대한 코드 섹션을 참조하세요.	
AWS 프로파일을 생성합니다.	PowerShell 명령 Set-용 AWS 도구를 실행하여 새 프로필을 생성합니다AWSCredential. 이 명령은 지정한 프로파일 이름 아래의 기본 보안 인증 파일에 액세스 키와 보안 키를 저장합니다. 이전에 생성한 Amazon SQS 정책을 이 계정에 연결합니다. AWS 액세스 키 ID 및 비밀번호 액세스 키를 유지합니다. 다음 단계에서 이 정보를 사용할 것입니다.	시스템 엔지니어
SQS 대기열을 생성합니다.	표준 대기열 또는 선입선출 (FIFO) 대기열을 만들 수 있습니다. 지침은 참조 섹션의 링크를 참조하세요.	시스템 엔지니어

.NET 애플리케이션 코드 수정

작업	설명	필요한 기술
AWS Toolkit for Visual Studio를 설치합니다.	이 툴킷은 Microsoft Visual Studio용 확장 프로그램입니다. 이를 통해 AWS에서 .NET 애플리케이션을 쉽게 구축하고 배포할 수 있습니다. 설치 및 사용 지침은 참조 섹션의 링크를 참조하세요.	애플리케이션 개발자
AWSSDK.SQS NuGet 패키지를 설치합니다.	Visual Studio에서 “NuGet 패키지 관리”를 선택하	애플리케이션 개발자

작업	설명	필요한 기술
	<p>거나 “Install-Package AWSSDK .SQS” 명령을 실행하여.SQS를 설치할 AWSSDK 수 있습니다.</p>	
<p>.NET 응용 프로그램에서 개체를 만듭니다. AWSCredentials</p>	<p>첨부 파일의 샘플 응용 프로그램에서는 에서 상속되는 Basic AWSCredentials 객체를 만드는 방법을 보여 줍니다. AWSCredentials 이전의 액세스 키 ID와 비밀 액세스 키를 사용하거나, 객체가 런타임에 사용자 프로필의 일부로.aws 폴더에서 이러한 키를 선택하도록 할 수 있습니다.</p>	<p>애플리케이션 개발자</p>
<p>SQS 클라이언트 객체를 생성합니다.</p>	<p>.NET 프레임워크용 SQS 클라이언트 객체(Amazon SQSClient)를 생성합니다. 이는 Amazon.SQS 네임스페이스의 일부입니다. Microsoft .Azure의 QueueClient 일부인 I 대신 이 개체가 필요합니다. ServiceBus 네임스페이스.</p>	<p>애플리케이션 개발자</p>
<p>SendMessageAsync 메서드를 호출하여 메시지를 SQS 대기열로 전송합니다.</p>	<p>를 사용하도록 메시지를 대기열로 보내는 코드를 변경하십시오. amazonSqsClient SendMessageAsync 메서드. 자세한 내용은 첨부된 코드 샘플을 참조하세요..</p>	<p>애플리케이션 개발자</p>

작업	설명	필요한 기술
ReceiveMessageAsync 메서드를 호출하여 SQS 대기열에서 메시지를 수신합니다.	메시지를 수신하는 코드를 변경하여 를 사용하십시오. amazonSqsClient ReceiveMessageAsync 메서드. 자세한 내용은 첨부된 코드 샘플을 참조하세요..	애플리케이션 개발자
DeleteMessageAsync 메서드를 호출하여 SQS 대기열에서 메시지를 삭제합니다.	메시지를 삭제하려면 QueueClient에서 코드를 변경하십시오. CompleteAsync 메서드를 로. amazonSqsClient DeleteMessageAsync 메소드. 자세한 내용은 첨부된 코드 샘플을 참조하세요..	애플리케이션 개발자

관련 리소스

- [.NET용 AWS SDK 개발자 안내서](#)
- [Amazon SQS를 사용한 메시징](#)
- [AWS SDK for .NET을 사용하여 Amazon SQS 대기열 생성 및 사용](#)
- [Amazon SQS 메시지 전송](#)
- [Amazon SQS Queue에서 메시지 수신](#)
- [Amazon SQS 대기열에서 메시지 삭제](#)
- [AWS Toolkit for Visual Studio](#)

추가 정보

이 패턴에는 두 개의 샘플 애플리케이션이 포함됩니다(첨부 파일 섹션 참조).

- AzureSbTestAppAzure 서비스 버스 대기열을 사용하는 코드를 포함합니다.
- AmazonSqsTestApp아마존 SQS를 사용합니다. 이 애플리케이션은 .NET Core 2.2를 사용하는 콘솔 애플리케이션으로, 메시지 송수신 예제가 포함되어 있습니다.

참고:

- QueueClient는 Microsoft.Azure의 QueueClient 일부인 I의 객체입니다. ServiceBus 네임스페이스 (Microsoft.Azure에 포함됨) ServiceBus NuGet 패키지).
- amazonSqsClient Amazon.SQS 네임스페이스 (.SQS 패키지에 포함) 의 일부인 AmazonSQSClient 의 객체입니다. AWSSDK NuGet
- 코드가 실행되는 위치(예: EC2에서 실행 중인지 여부)에 따라 역할에 SQS 대기열에 쓸 수 있는 권한 이 있어야 합니다.

첨부

이 문서와 관련된 추가 콘텐츠에 액세스하려면 [attachment.zip](#) 파일의 압축을 풉니다.

오라클 데이터 펌프와 AWS DMS를 사용하여 오라클 JD 에드워즈 EnterpriseOne 데이터베이스를 AWS로 마이그레이션하십시오.

작성자: Thanigaivel Thirumalai(AWS)

환경: 프로덕션	출처: 오라클 JD 에드워즈 EnterpriseOne	대상: Amazon RDS for Oracle
R 유형: 리플랫폼	워크로드: Oracle	기술: 마이그레이션, 데이터베이스
AWS 서비스: Amazon RDS; AWS DMS		

요약

JD Edwards EnterpriseOne 데이터베이스를 [Amazon 관계형 데이터베이스 서비스 \(Amazon RDS\)](#) 에서 마이그레이션하고 실행할 수 있습니다. 데이터베이스를 Amazon RDS로 마이그레이션하면 AWS에서 백업 작업과 고가용성 설정을 처리하므로 사용자는 EnterpriseOne 애플리케이션과 해당 기능을 유지 관리하는 데 집중할 수 있습니다. 마이그레이션 프로세스 중에 고려해야 할 주요 요소의 포괄적인 목록은 AWS 권장 가이드의 [Oracle 데이터베이스 마이그레이션 전략](#)을 참고하십시오.

EnterpriseOne 데이터베이스를 마이그레이션하는 방법은 다음과 같이 여러 가지가 있습니다.

- 스키마 및 테이블 생성에는 Oracle Universal Batch Engine (UBE) R98403 사용, 마이그레이션에는 AWS Database Migration Service(AWS DMS) 사용
- 스키마 및 테이블 생성에는 DB 네이티브 도구 사용, 마이그레이션에는 AWS DMS 사용
- 기존 데이터 마이그레이션(전체 로드)을 위한 DB 네이티브 도구 사용, 변경 데이터 캡처(CDC) 작업을 위한 AWS DMS 사용

이 패턴은 세 번째 옵션을 다룹니다. [AWS](#) DMS 및 CDC 기능과 함께 Oracle 데이터 펌프를 사용하여 온 프레미스 EnterpriseOne 데이터베이스를 Amazon RDS for Oracle로 마이그레이션하는 방법을 설명합니다.

[Oracle JD EnterpriseOne Edwards](#)는 제품 또는 물리적 자산을 제조, 건설, 배포, 서비스 또는 관리하는 조직을 위한 엔터프라이즈 리소스 계획 (ERP) 솔루션입니다. JD EnterpriseOne Edwards는 다양한 하드웨어, 운영 체제 및 데이터베이스 플랫폼을 지원합니다.

JD EnterpriseOne Edwards와 같은 중요한 ERP 애플리케이션을 마이그레이션할 때는 다운타임을 최소화하는 것이 중요합니다. AWS DMS는 소스 데이터베이스에서 대상 데이터베이스로의 전체 로드와 연속 복제를 모두 지원하여 가동 중지 시간을 최소화합니다. 또한, AWS DMS는 마이그레이션에 대한 실시간 모니터링 및 로깅을 제공하므로 가동 중지 시간을 유발할 수 있는 문제를 식별하고 해결하는 데 도움이 됩니다.

AWS DMS로 변경 내용을 복제할 때는 데이터베이스 로그에서 변경 내용을 읽기 위한 시작 지점으로 시간 또는 SCN(시스템 변경 번호)를 지정해야 합니다. AWS DMS가 이러한 변경 사항에 액세스할 수 있도록 하려면 지정된 시간(15일 권장) 동안 서버에서 이러한 로그에 액세스할 수 있도록 유지하는 것이 중요합니다.

사전 조건 및 제한 사항

사전 조건

- 대상 데이터베이스로 AWS 클라우드 환경에서 프로비저닝된 Amazon RDS for Oracle용 데이터베이스. 자세한 지침은 [Amazon RDS 설명서](#)를 참조하세요.
- 온프레미스 또는 AWS의 Amazon Elastic Compute Cloud (Amazon EC2) 인스턴스에서 실행되는 EnterpriseOne 데이터베이스입니다.

참고: 이 패턴은 온프레미스에서 AWS로 마이그레이션하기 위해 설계되었지만 EC2 인스턴스의 EnterpriseOne 데이터베이스를 사용하여 테스트되었습니다. 온프레미스 환경에서 마이그레이션하려는 경우 적절한 네트워크 연결을 구성해야 합니다.

- 스키마 세부 정보. 마이그레이션하려는 Oracle 데이터베이스 스키마 (예: DV920) 를 식별하십시오. EnterpriseOne 마이그레이션 프로세스를 시작하기 전에 스키마에 대한 다음과 같은 세부 정보를 수집합니다.
 - 스키마 크기
 - 객체 유형당 객체 수
 - 잘못된 객체 수

제한 사항

- 대상 Amazon RDS for Oracle 데이터베이스에서 원하는 스키마를 생성해야 합니다. AWS DMS는 이러한 스키마를 생성하지 않습니다. ([에픽](#) 섹션에서는 Data Pump를 사용하여 스키마를 내보내고 가

져오는 방법을 설명합니다.) 대상 Oracle 데이터베이스에 대한 스키마 이름이 이미 있어야 합니다. 원본 스키마의 테이블을 사용자 또는 스키마로 가져오게 되며, AWS DMS가 관리자 또는 시스템 계정을 사용하여 대상 인스턴스에 연결합니다. 여러 스키마를 마이그레이션하려면 다중 복제 작업을 생성하면 됩니다. 대상 인스턴스의 다른 스키마로 데이터를 마이그레이션할 수도 있습니다. 이를 위해서는 AWS DMS 테이블 매핑의 스키마 변환 규칙을 사용하십시오.

- 이 패턴은 데모 데이터 세트로 테스트되었습니다. 데이터 세트와 사용자 지정의 호환성을 검증하는 것을 권장합니다.
- 이 패턴은 Microsoft Windows에서 실행되는 EnterpriseOne 데이터베이스를 사용합니다. 하지만 AWS DMS에서 지원하는 다른 운영 체제에서도 동일한 프로세스를 사용할 수 있습니다.

아키텍처

다음 다이어그램은 Oracle EnterpriseOne 데이터베이스에서 원본 데이터베이스로, Amazon RDS for Oracle 데이터베이스에서 대상 데이터베이스로 실행되는 시스템을 보여줍니다. 데이터를 소스 Oracle 데이터베이스에서 내보내고, Oracle Data Pump를 사용하여 대상 Amazon RDS for Oracle 데이터베이스로 가져오고, AWS DMS를 사용하여 CDC 업데이트를 위해 복제합니다.

1. Oracle Data Pump는 소스 데이터베이스에서 데이터를 추출하고, 이 데이터는 Amazon RDS for Oracle 데이터베이스 대상으로 전송됩니다.
2. CDC 데이터는 소스 데이터베이스에서 AWS DMS의 소스 엔드포인트로 전송됩니다.
3. 소스 엔드포인트에서 데이터는 AWS DMS 복제 인스턴스로 전송되며, 여기서 복제 작업이 수행됩니다.
4. 복제 작업이 완료되면 데이터가 AWS DMS의 대상 엔드포인트로 전송됩니다.
5. 대상 엔드포인트에서 데이터는 Amazon RDS for Oracle용 데이터베이스 인스턴스로 전송됩니다.

도구

서비스

- [AWS Database Migration Service\(AWS DMS\)](#)를 사용하면 데이터 스토어를 AWS 클라우드로 마이그레이션하거나 클라우드와 온프레미스 설정 조합 간에 마이그레이션할 수 있습니다.
- [Oracle용 Amazon Relational Database Service\(RDS\)](#)는 AWS 클라우드에서 Oracle 관계형 데이터베이스를 설정, 운영 및 조정하는 데 도움이 됩니다.

기타 서비스

- [Oracle Data Pump](#)를 사용하면 한 데이터베이스에서 다른 데이터베이스로 데이터와 메타데이터를 빠른 속도로 이동할 수 있습니다.

모범 사례

LOB 마이그레이션

소스 데이터베이스에 대상 데이터베이스로 마이그레이션해야 하는 대형 바이너리 객체(LOB)가 포함된 경우, AWS DMS는 다음과 같은 옵션을 제공합니다.

- 전체 LOB 모드 - AWS DMS는 모든 LOB를 크기에 관계없이 소스에서 대상 데이터베이스로 마이그레이션합니다. 마이그레이션은 다른 모드보다 느리지만 데이터가 잘리지 않는다는 장점이 있습니다. 성능 향상을 위해 새 복제 인스턴스에서 별도의 작업을 생성하여 LOB가 몇 메가바이트보다 큰 테이블을 마이그레이션할 수 있습니다.
- 제한된 LOB 모드 — LOB 열 데이터의 최대 크기를 지정하면 AWS DMS가 리소스를 사전 할당하고 LOB를 대량으로 적용할 수 있습니다. LOB 열의 크기가 작업에 지정된 크기를 초과하는 경우 AWS DMS는 데이터를 자르고 AWS DMS 로그 파일에 경고를 보냅니다. LOB 데이터 크기가 제한된 LOB 크기 이내인 경우 제한된 LOB 모드를 사용하여 성능을 개선할 수 있습니다.
- 인라인 LOB 모드 - 작은 LOB와 큰 LOB를 모두 복제하여 데이터를 자르거나 작업 성능을 저하시키지 않고 LOB를 마이그레이션할 수 있습니다. 먼저 `InlineLobMaxSize` 매개변수 값을 지정합니다. 이 값은 전체 LOB 모드가 `true`로 설정된 경우에만 사용할 수 있습니다. AWS DMS 작업은 소규모 LOB를 인라인으로 전송하므로 더 효율적입니다. 그런 다음 AWS DMS는 소스 테이블에서 조회를 수행하여 대규모 LOB를 마이그레이션합니다. 하지만 인라인 LOB 모드는 전체 로드 단계에서만 작동합니다.

시퀀스 값 생성

AWS DMS CDC 프로세스 중에는 소스 데이터베이스에서 중복 시퀀스 번호가 복제되지 않습니다. 시퀀스 값의 불일치를 방지하려면 모든 시퀀스의 소스에서 가장 최근의 시퀀스 값을 생성하여 대상 Amazon RDS for Oracle 데이터베이스에 적용해야 합니다.

AWS Secrets Manager

보안 인증 정보를 관리하는 데 도움이 되도록 [AWS Secrets Manager를 사용하여 AWS DMS 엔드포인트 보안 인증 정보 관리하기](#) 블로그 게시물의 지침을 따르는 것을 권장합니다.

성능

- 복제 인스턴스 – 최적의 인스턴스 크기 선택에 대한 지침은 AWS DMS 설명서의 [복제 인스턴스에 가장 적합한 크기 선택하기](#)를 참고하십시오.
- 연결 옵션 – 지연 문제를 방지하려면 적절한 연결 옵션을 선택하는 것을 권장합니다. AWS Direct Connect는 기업 데이터 센터와 AWS 간의 전용 연결이기 때문에 AWS 리소스에 대해 최단 경로를 제공합니다. 전송 중에는 네트워크 트래픽은 AWS 글로벌 네트워크에 남아 있으며 인터넷을 통해 전송되지 않습니다. 이렇게 하면 VPN 또는 공용 인터넷을 사용할 때와 비교할 때 병목 현상이 발생하거나 예상치 못한 지연 시간 증가가 발생할 가능성이 줄어듭니다.
- 네트워크 대역폭 – 성능을 최적화하려면 네트워크 처리량이 빠르지 확인하십시오. 온프레미스 소스 데이터베이스와 AWS DMS 간에 VPN 터널을 사용하는 경우 대역폭이 워크로드에 충분한지 확인하십시오.
- 작업 병렬화 – 전체 로드 중에 여러 테이블을 병렬로 로드하여 데이터 복제 속도를 높일 수 있습니다. 이 패턴은 RDBMS 엔드포인트를 사용하므로 이 옵션은 전체 로드 프로세스에만 적용됩니다. 작업 병렬화는 병렬로 실행되는 전체 로드 하위 작업의 수를 결정하는 MaxFullLoadSubTasks 매개변수에 의해 제어됩니다. 기본적으로 이 매개변수는 8로 설정되어 있습니다. 즉, 전체 모드에서 8개 테이블(테이블 매핑에서 선택한 경우)이 함께 로드됩니다. 작업에 대한 JSON 스크립트의 전체 로드 작업 설정 섹션에서 이 매개변수를 조정할 수 있습니다.
- 테이블 병렬화 – AWS DMS를 사용하면 여러 병렬 스레드를 사용하여 하나의 대형 테이블을 로드할 수도 있습니다. 이는 수십억 개의 레코드뿐만 아니라 여러 파티션과 하위 파티션이 있는 Oracle 소스 테이블에 특히 유용합니다. 소스 테이블이 파티셔닝되지 않은 경우 병렬 로드에는 열 경계를 사용할 수 있습니다.
- 로드 분할 – 로드를 여러 작업 또는 AWS DMS 인스턴스로 분할하는 경우 변경 사항을 캡처할 때 트랜잭션 경계를 기억해 두십시오.

에픽

Oracle Data Pump를 사용하여 스키마를 내보낼 수 있습니다 EnterpriseOne .

작업	설명	필요한 기술
SCN을 생성합니다.	소스 데이터베이스가 활성 상태이고 EnterpriseOne 응용 프로그램에서 사용 중인 경우 Oracle Data Pump를 사용하여 데이터 내보내기를 시작합니다. 먼저 Oracle Data Pump를 사용하여 내보내는 동안 데	DBA

작업	설명	필요한 기술
	<p>이더 일관성을 유지하고 AWS DMS에서 CDC의 시작점으로 사용하기 위해 소스 데이터베이스에서 시스템 변경 번호 (SCN)를 생성해야 합니다.</p> <p>소스 데이터베이스에서 현재 SCN을 생성하려면 다음 SQL 문을 사용합니다.</p> <pre data-bbox="592 646 1029 926"> SQL> select current_scn from v\$database; CURRENT_SCN ----- 30009727 </pre> <p>생성된 SCN을 저장합니다. 데이터를 내보내고 AWS DMS 복제 작업을 생성할 때 SCN을 사용합니다.</p>	

작업	설명	필요한 기술
<p>파라미터 파일을 생성합니다.</p>	<p>스키마를 내보내기 위한 매개 변수 파일을 만들려면 다음 코드를 사용할 수 있습니다.</p> <pre data-bbox="597 394 1026 751"> directory=DMS_DATA _PUMP_DIR logfile=export_dms.log dumpfile=export_dms_data.dmp schemas=<schema name> flashback_scn=<SCN from previous command> </pre> <p>참고: 요구 사항에 따라 다음 명령을 사용하여 나만의 DATA_PUMP_DIR 을 정의할 수도 있습니다.</p> <pre data-bbox="597 1008 1026 1444"> SQL> CREATE OR REPLACE DIRECTORY DMS_DATA_ PUMP_DIR AS '<Directory for dump>'; Directory created. SQL> GRANT READ, WRITE ON DIRECTORY DMS_DATA_ PUMP_DIR TO SYSTEM; Grant succeeded. </pre>	<p>DBA</p>

작업	설명	필요한 기술
스키마를 내보냅니다.	<p>내보내기를 수행하려면 다음과 같이 expdp 유틸리티를 사용하십시오.</p> <pre data-bbox="592 394 1027 1877"> C:\Users\Administrator>expdp system/ *****@<DB Name> PARFILE='<Path to PAR file create above>' Export: Release 19.0.0.0.0 - Production on *** ** **.**. ** Version 19.3.0.0.0 Copyright (c) 1982, 2019, Oracle and/or its affiliates. All rights reserved. Connected to: Oracle Database 19c Standard Edition 2 Release 19.0.0.0.0 - Production n Starting "SYSTEM". "SYS_EXPORT_SCHEMA _02": system/** *****@<DB Name>PARF ILE='E:\exp_dms_da tapump.par' Processing object type SCHEMA_EXPORT/TABLE/ TABLE_DATA Processing object type SCHEMA_EXPORT/TABL E/INDEX/STATISTICS/ INDEX_STATISTICS Processing object type SCHEMA_EXPORT/TABL </pre>	DBA

작업	설명	필요한 기술
	<pre> E/STATISTICS/TABLE _STATISTICS Processing object type SCHEMA_EXPORT/STAT ISTICS/MARKER Processing object type SCHEMA_EXPORT/USER Processing object type SCHEMA_EXPORT/ROLE _GRANT Processing object type SCHEMA_EXPORT/DEFA ULT_ROLE Processing object type SCHEMA_EXPORT/TABL ESPACE_QUOTA Processing object type SCHEMA_EXPORT/PRE_ SCHEMA/PROCACT_SCHEMA Processing object type SCHEMA_EXPORT/TABLE/ TABLE Processing object type SCHEMA_EXPORT/TABL E/GRANT/OWNER_GRANT/ OBJECT_GRANT Processing object type SCHEMA_EXPORT/TABLE/ INDEX/INDEX Processing object type SCHEMA_EXPORT/TABLE/ CONSTRAINT/CONSTRAINT . . exported "<Schema Name>". "<Table Name>" 228.9 MB 496397 rows Master table "SYSTEM". "SYS_EXPORT_SCHEMA _02" successfully loaded/unloaded </pre>	

작업	설명	필요한 기술
	<pre> ***** ***** ***** ***** **** Dump file set for SYSTEM.SYS_EXPORT_ SCHEMA_02 is: E:\DMSDUMP\EXPORT_ DMS_DATA.DMP Job "SYSTEM"."SYS_EXPO RT_SCHEMA_02" successfully completed at *** ** * *:*.**.* **** elapsed 0 00:01:57 </pre>	

Oracle Data Pump를 사용하여 스키마를 가져올 수 있습니다 EnterpriseOne .

작업	설명	필요한 기술
<p>덤프 파일을 대상 인스턴스로 전송합니다.</p>	<p>DBMS_FILE_TRANSFER 유틸리티를 사용하여 파일을 전송하려면 소스 데이터베이스에서 Amazon RDS for Oracle 인스턴스로 데이터베이스 링크를 생성해야 합니다. 이 링크가 설정되면 유틸리티를 사용하여 데이터 펌프 파일을 Amazon RDS 인스턴스로 직접 전송할 수 있습니다.</p> <p>또는, 데이터 펌프 파일을 Amazon Simple Storage Service(Amazon S3) 로 전송한 다음 이 파일을 Amazon RDS for Oracle 인스턴스로 가져올 수 있습니다. 이 옵션에 대</p>	<p>DBA</p>

작업	설명	필요한 기술
	<p>한 자세한 내용은 추가 정보 섹션을 참고하십시오.</p> <p>대상 DB 인스턴스의 Amazon RDS 마스터 사용자에게 연결하는 데이터베이스 링크 ORARDSDB를 생성하려면 원본 데이터베이스에서 다음 명령을 실행하세요.</p> <pre data-bbox="592 646 1029 1856"> sqlplus / as sysdba SQL*Plus: Release 19.0.0.0.0 on *** *** ** **:**:** **** Version 19.3.0.0.0 Copyright (c) 1982, 2019, Oracle. All rights reserved. Connected to: Oracle Database 19c Standard Edition 2 Release 19.0.0.0.0 Version 19.3.0.0.0 SQL> create database link orardsdb connect to admin identifie d by "*****" using '(DESCRIPTION = (ADDRESS = (PROTOCOL = TCP)(HOST = orcl.**** **.us-east-1.rds.a mazonaws.com)(PORT = 1521))(CONNECT_DATA = (SERVER = DEDICATED) (SERVICE_NAME = orcl)))'; </pre>	

작업	설명	필요한 기술
	<pre>Database link created. SQL></pre>	
<p>데이터베이스 링크를 테스트합니다.</p>	<p>sqlplus를 사용하여 Amazon RDS for Oracle용 대상 데이터베이스에 연결할 수 있는지 확인할 수 있도록 데이터베이스 링크를 테스트합니다.</p> <pre>SQL> select name from v \$database@orardsdb; NAME ----- ORCL</pre>	<p>DBA</p>

작업	설명	필요한 기술
<p>덤프 파일을 대상 데이터베이스로 전송합니다.</p>	<p>덤프 파일을 Amazon RDS for Oracle 데이터베이스에 복사하기 위해 DATA_PUMP_DIR 기본 디렉터를 사용하거나 대상 Amazon RDS 인스턴스에서 실행해야 하는 다음과 같은 코드를 사용하여 나만의 디렉터를 생성할 수 있습니다.</p> <pre data-bbox="592 632 1027 1031"> exec rdsadmin.rdsadmin_util.create_directory(p_directory_name => 'DMS_TARGET_PUMP_DIR'); PL/SQL procedure successfully completed . </pre> <p>다음 스크립트는 orardsdb라는 데이터베이스 링크를 사용하여 소스 인스턴스에 있는 EXPORT_DMS_DATA.DMP 라는 덤프 파일을 대상 Amazon RDS for Oracle 데이터베이스로 복사합니다. 소스 데이터베이스 인스턴스에서 이 스크립트를 실행해야 합니다.</p> <pre data-bbox="592 1522 1027 1770"> BEGIN DBMS_FILE_TRANSFER.PUT_FILE(source_directory_object => 'DMS_DATA_PUMP_DIR', </pre>	<p>DBA</p>

작업	설명	필요한 기술
	<pre> source_file_name => 'EXPORT_DMS_DATA.D MP', destination_directory_ object => 'DMS_TARG ET_PUMP_DIR', destination_file_name => 'EXPORT_DMS_DATA.D MP', destination_database => 'orardsdb'); END; PL/SQL procedure successfully completed . </pre>	
<p>대상 데이터베이스에 있는 덤프 파일을 나열합니다.</p>	<p>PL/SQL 절차가 완료된 후 다음 코드를 사용하여 Amazon RDS for Oracle 데이터베이스에 데이터 덤프 파일을 나열할 수 있습니다.</p> <pre> select * from table (rdsadmin.rds_file _util.listdir(p_di rectory => 'DMS_TARG ET_PUMP_DIR')); </pre>	DBA

작업	설명	필요한 기술
<p>대상 인스턴스에서 JDE 전용 사용자를 생성합니다.</p>	<p>대상 인스턴스에서 다음 명령을 사용하여 JD Edwards 프로필 및 역할을 생성합니다.</p> <pre data-bbox="594 394 1026 991"> SQL> CREATE PROFILE "JDEPROFILE" LIMIT IDLE_TIME 15; Profile created. SQL> CREATE ROLE "JDE_ROLE"; Role created. SQL> CREATE ROLE "JDEADMIN"; CREATE ROLE "JDEUSER"; Role created. Role created. </pre> <p>역할에 필요한 권한 부여:</p> <pre data-bbox="594 1100 1026 1457"> SQL> GRANT CREATE ANY SEQUENCE TO JDE_ROLE; GRANT DROP ANY SEQUENCE TO JDE_ROLE; GRANT CREATE ANY TRIGGER TO JDE_ROLE; GRANT DROP ANY TRIGGER TO JDE_ROLE; </pre>	<p>DBA, JDE CNC</p>

작업	설명	필요한 기술
대상 인스턴스에 테이블스페이스를 생성합니다.	<p>이 마이그레이션과 관련된 스키마에 대해 다음 명령을 사용하여 대상 인스턴스에 필요한 테이블스페이스를 생성합니다.</p> <pre data-bbox="609 443 1027 842">SQL> CREATE TABLESPACE <Tablespace Name for Tables>; Tablespace created. SQL> CREATE TABLESPACE <Tablespace Name for Indexes>; Tablespace created.</pre>	DBA, JDE CNC

작업	설명	필요한 기술
<p>대상 데이터베이스에서 가져오기를 시작합니다.</p>	<p>가져오기 프로세스를 시작하기 전에 데이터 덤프 파일을 사용하여 대상 Amazon RDS for Oracle 데이터베이스에 역할, 스키마 및 테이블스페이스를 설정합니다.</p> <p>가져오기를 수행하려면 Amazon RDS 기본 사용자 계정으로 대상 데이터베이스에 액세스하고 tnsnames.ora 파일에서 Amazon RDS for Oracle 데이터베이스 tns-entry 를 포함하는 연결 문자열 이름을 사용합니다. 필요한 경우 다른 테이블스페이스나 다른 스키마 이름으로 데이터 덤프 파일을 가져오는 재매핑 옵션을 포함할 수 있습니다.</p> <p>가져오기를 시작하려면 다음 코드를 사용합니다.</p> <pre data-bbox="592 1270 1031 1507"> impdp admin@orardsdb directory=DMS_TARG ET_PUMP_DIR logfile=i mport.log dumpfile= EXPORT_DMS_DATA.DMP </pre> <p>가져오기에 성공하려면 가져오기 로그 파일에 오류가 있는지 확인하고 객체 수, 행 수, 잘못된 객체 등의 세부 정보를 검토하십시오. 잘못된 객체가 있는 경우 해당 객체를 다시 컴파일하십시오. 또한 소스 데이터베</p>	<p>DBA</p>

작업	설명	필요한 기술
	이스 객체와 대상 데이터베이스 객체를 비교하여 일치하는지 확인하십시오.	

소스 및 대상 엔드포인트로 AWS DMS 복제 인스턴스를 프로비저닝합니다.

작업	설명	필요한 기술
템플릿을 다운로드합니다.	AWS CloudFormation DMS_Instance.yaml 템플릿을 다운로드하여 AWS DMS 복제 인스턴스와 해당 소스 및 대상 엔드포인트를 프로비저닝합니다.	클라우드 관리자, DBA
스택 생성을 시작합니다.	<ol style="list-style-type: none"> 1. AWS 관리 콘솔에 로그인하고 https://console.aws.amazon.com/cloudformation 에서 AWS CloudFormation 콘솔을 엽니다. 2. 스택 생성을 선택합니다. 3. 템플릿 지정에서 템플릿 파일 업로드를 선택합니다. 4. 파일 선택을 선택합니다. 5. <code>DMS_instance.yaml</code> 파일을 선택합니다. 6. 다음을 선택합니다. 	클라우드 관리자, DBA
파라미터를 지정합니다.	<ol style="list-style-type: none"> 1. 스택 이름에 스택 이름을 입력합니다. 2. AWS DMS 인스턴스 파라미터에 다음과 같은 매개변수를 입력합니다. 	클라우드 관리자, DBA

작업	설명	필요한 기술
	<ul style="list-style-type: none"> • DMS InstanceType — 비즈니스 요구 사항에 따라 AWS DMS 복제 인스턴스에 필요한 인스턴스를 선택합니다. • DMS StorageSize — 마이그레이션 규모에 따라 AWS DMS 인스턴스의 스토리지 크기를 입력합니다. <p>3. 소스 Oracle 데이터베이스 구성에는 다음과 같은 매개변수를 입력합니다.</p> <ul style="list-style-type: none"> • SourceOracleEndpointID — 원본 Oracle 데이터베이스 서버 이름 • SourceOracleDatabaseName- 해당하는 경우 소스 데이터베이스 서비스 이름 또는 세션 ID (SID) • SourceOracleUsername— 소스 데이터베이스 사용자 이름 (system기본값) • SourceOracleDbPassword — 소스 데이터베이스 사용자 이름의 비밀번호입니다. • SourceOracleDBPort — 소스 데이터베이스 포트입니다. 	

작업	설명	필요한 기술
	<p>4. Oracle 데이터베이스용 대상 RDS 구성에는 다음 파라미터를 입력합니다.</p> <ul style="list-style-type: none"> • 대상 RDS OracleEndpoint ID — 대상 RDS 데이터베이스 엔드포인트 • 대상 RDS OracleDatabaseName — 대상 RDS 데이터베이스 이름 • 대상 RDS — 대상 RDS 사용자 이름 OracleUsername • TargetRDSOracleDBPassword — 대상 RDS 비밀번호 • TargetOracleDBPort — 대상 RDS 데이터베이스 포트입니다. <p>5. VPC, 서브넷 및 보안 그룹 구성의 경우 다음과 같은 매개변수를 입력합니다.</p> <ul style="list-style-type: none"> • VPCID — 복제 인스턴스를 위한 VPC • VPC SecurityGroupId — 복제 인스턴스의 VPC 보안 그룹 • DMSSubnet1 — 가용 영역 1을 위한 서브넷 • DMSSubnet2 — 가용 영역 2를 위한 서브넷 <p>6. 다음을 선택합니다.</p>	

작업	설명	필요한 기술
스택을 생성합니다.	<ol style="list-style-type: none"> 구성 스택 옵션 페이지에서 태그에 선택적 값을 입력합니다. 다음을 선택합니다. 검토 페이지에서 세부 정보를 확인한 다음, 제출을 선택합니다. <p>프로비저닝은 약 5~10분 내에 완료될 것입니다. AWS CloudFormation 스택 페이지에 CREATE_COMPLETE가 표시되면 작업이 완료됩니다.</p>	클라우드 관리자, DBA
엔드포인트를 설정합니다.	<ol style="list-style-type: none"> https://console.aws.amazon.com/dms/v2/에서 AWS DMS 콘솔을 엽니다. 리소스 관리의 경우 복제 인스턴스를 선택한 다음 복제 인스턴스를 검토하십시오. 리소스 관리의 경우 엔드포인트를 선택한 다음 엔드포인트를 검토하십시오. 	클라우드 관리자, DBA
연결을 테스트합니다.	소스 및 대상 엔드포인트의 상태가 활성화로 표시되면 연결을 테스트합니다. 각 엔드포인트(소스 및 대상)에 대해 테스트 실행을 선택하여 상태가 성공으로 표시되는지 확인합니다.	클라우드 관리자, DBA

실시간 복제를 위한 AWS DMS 복제 작업 생성

작업	설명	필요한 기술
복제 작업을 생성합니다.	<p>다음 단계를 사용하여 AWS DMS 복제 작업을 생성합니다.</p> <ol style="list-style-type: none"> 1. https://console.aws.amazon.com/dms/v2/에서 AWS DMS 콘솔을 엽니다. 2. 탐색 창의 데이터 마이그레이션 아래에서 데이터베이스 마이그레이션 작업을 선택합니다. 3. 작업 구성 항목의 작업 식별자에 작업 식별자를 입력합니다. 4. 복제 인스턴스에 대해 생성한 DMS 복제 인스턴스를 선택합니다. 5. 소스 데이터베이스 엔드포인트에는 소스 엔드포인트를 선택합니다. 6. 대상 데이터베이스 엔드포인트에는 대상 Amazon RDS for Oracle 데이터베이스를 선택합니다. 7. 마이그레이션 유형에는 데이터 변경 내용만 복제를 선택합니다. 추가 로깅을 켜야 한다는 메시지가 표시되면 문제 해결 섹션의 지침을 따르십시오. 8. 작업 설정 항목에서 로그 시퀀스 번호 지정을 선택합니다. 	클라우드 관리자, DBA

작업	설명	필요한 기술
	<p>9. 시스템 변경 번호에는 소스 Oracle 데이터베이스에서 생성한 Oracle 데이터베이스 SCN을 입력합니다.</p> <p>10.검증 활성화를 선택합니다.</p> <p>11[로그 활성화] 를 선택합니다. CloudWatch</p> <p>이 기능을 활성화하면 데이터와 Amazon CloudWatch 로그를 검증하여 AWS DMS 복제 인스턴스 로그를 검토할 수 있습니다.</p> <p>12.선택 규칙에서 다음 사항을 완료하십시오.</p> <ul style="list-style-type: none"> • 스키마에 대해 스키마 입력을 선택합니다. • 스키마 이름에는 JDE 스키마 이름(예: DV920)을 입력합니다. • 테이블 이름에 %를 입력합니다. • 작업에 포함을 선택합니다. <p>13.작업 생성을 선택합니다.</p> <p>작업을 생성하면 AWS DMS가 CDC 시작 모드에서 사용자가 제공한 SCN에서 Amazon RDS for Oracle 데이터베이스 인스턴스로 진행 중인 변경 사항을 마이그레이션합니다. 로그를 검토하여 마이그레이</p>	

작업	설명	필요한 기술
	이션을 확인할 수도 있습니다. CloudWatch	
복제 작업을 반복합니다.	이전 단계를 반복하여 마이그레이션의 일부인 다른 JD Edwards 스키마에 대한 복제 작업을 생성합니다.	클라우드 관리자, DBA, JDE CNC 관리자

대상 Amazon RDS for Oracle 데이터베이스에서 데이터베이스 스키마를 확인합니다.

작업	설명	필요한 기술
데이터 전송을 검증합니다.	AWS DMS 작업이 시작된 후 작업 페이지의 테이블 통계 탭을 확인하여 데이터에 대한 변경 사항을 확인할 수 있습니다. 콘솔의 데이터베이스 마이그레이션 작업 페이지에서 진행 중인 복제 상태를 모니터링할 수 있습니다. 자세한 내용은 AWS DMS 데이터 검증 섹션을 참조하세요.	클라우드 관리자, DBA

전환

작업	설명	필요한 기술
복제를 중지합니다.	복제 절차를 중단하고 소스 애플리케이션 서비스를 중지합니다.	클라우드 관리자, DBA
JD Edwards 애플리케이션을 실행합니다.	AWS에서 대상 JD Edwards 프레젠테이션 및 로직 티어 애플	DBA, JDE CNC 관리자

작업	설명	필요한 기술
	<p>리케이션을 시작하고 Amazon RDS for Oracle 데이터베이스로 전송합니다.</p> <p>애플리케이션에 액세스하면 이제 모든 연결이 Amazon RDS for Oracle 데이터베이스와 설정되었음을 알 수 있습니다.</p>	
<p>소스 데이터베이스를 끕니다.</p>	<p>더 이상 연결되어 있지 않은지 확인한 후 소스 데이터베이스를 끌 수 있습니다.</p>	<p>DBA</p>

문제 해결

문제	Solution
<p>진행 중인 복제를 위해 소스 데이터베이스에서 보충 로깅을 활성화하라는 경고 메시지가 나타납니다.</p>	<p>보충 로깅을 활성화하려면 다음 명령을 입력합니다.</p> <pre>SQL> ALTER DATABASE ADD SUPPLEMENTAL LOG DATA (ALL) COLUMNS; SQL> ALTER DATABASE ADD SUPPLEMENTAL LOG DATA (PRIMARY KEY) COLUMNS; SQL> ALTER DATABASE ADD SUPPLEMENTAL LOG DATA (UNIQUE) COLUMNS; SQL> ALTER DATABASE ADD SUPPLEMENTAL LOG DATA (FOREIGN KEY) COLUMNS; SQL> ALTER DATABASE ADD SUPPLEMENTAL LOG DATA (PRIMARY KEY) COLUMNS; SQL> ALTER DATABASE ADD SUPPLEMENTAL LOG DATA (UNIQUE) COLUMNS;</pre>
<p>AWS DMS는 보충 로깅을 해제했습니다.</p>	<p>보충 로깅은 AWS DMS에서 기본적으로 꺼져 있습니다. 소스 Oracle 엔드포인트에서 이를 활성화하려면,</p>

문제	Solution
	<ol style="list-style-type: none"> 1. AWS 관리 콘솔에 로그인하고 https://console.aws.amazon.com/dms/v2/ 에서 AWS DMS 콘솔을 엽니다. 2. 엔드포인트를 선택합니다. 3. 보충 로깅을 추가할 Oracle 원본 엔드포인트를 선택합니다. 4. 수정을 선택합니다. 5. 고급을 선택하고 나서 추가 연결 속성 텍스트 상자에 다음 코드를 추가합니다. <pre>addSupplementalLogging=Y</pre> 6. 수정을 선택합니다.
<p>CDB 수준에서는 보충 로깅이 활성화되지 않습니다.</p>	<ol style="list-style-type: none"> 1. 이 명령을 입력합니다. <pre>SQL> alter session set container = CDB\$ROOT; Session altered.</pre> 2. 보충 로깅을 활성화하는 단계를 반복합니다.
<p>“테스트 엔드포인트 실패: 애플리케이션 상태: 1020912, 애플리케이션 메시지: Oracle PDB 환경에서 지원되지 않음 엔드포인트 초기화 실패”라는 오류 메시지가 표시됩니다. LogMiner</p>	<p>이 오류 메시지가 나타나는 경우 대신 바이너리 리더를 사용할 수 있습니다. LogMiner 엔드포인트 설정에서 소스 데이터베이스의 추가 연결 속성에 다음 줄을 추가합니다.</p> <pre>useLogMinerReader=N;useBfile=Y;</pre>

관련 리소스

- [AWS Database Migration Service 시작하기](#)
- [AWS Database Migration Service의 모범 사례](#)

- [Oracle 데이터베이스를 AWS Cloud로 마이그레이션하기](#)
- [AWS용 AWS Database Migration Service 리소스 유형 참조 CloudFormation](#)
- [AWS Secrets Manager를 사용한 AWS DMS 엔드포인트 보안 인증 정보 관리](#)
- [AWS Database Migration Service의 마이그레이션 작업 문제 해결](#)
- [AWS Database Migration Service의 모범 사례](#)

추가 정보

Amazon S3를 사용하여 파일 전송하기

Amazon S3에 파일을 전송하려면 AWS CLI 또는 Amazon S3 콘솔을 사용하면 됩니다. Amazon S3로 파일을 전송한 후, Amazon RDS for Oracle 인스턴스를 사용하여 Amazon S3에서 데이터 펌프 파일을 가져올 수 있습니다.

Amazon S3 통합을 대체 방법으로 사용하여 덤프 파일을 전송하기로 선택한 경우 다음 단계를 수행합니다.

1. S3 버킷을 생성합니다.
2. Oracle Data Pump를 사용하여 원본 데이터베이스에서 데이터를 내보냅니다.
3. S3 버킷에 Data Pump 파일을 업로드합니다.
4. S3 버킷에서 대상 Amazon RDS for Oracle 데이터베이스로 데이터 펌프 파일을 다운로드합니다.
5. 데이터 펌프 파일을 사용하여 가져오기를 수행합니다.

참고: S3와 RDS 인스턴스 간에 대용량 데이터 파일을 전송하려면 [Amazon S3 Transfer Acceleration](#) 기능을 사용하는 것을 권장합니다.

AWS DMS를 사용하여 오라클 PeopleSoft 데이터베이스를 AWS로 마이그레이션합니다.

작성자: sampath kathirvel(AWS)

환경: 프로덕션	출처: 오라클 PeopleSoft	대상: Amazon RDS for Oracle
R 유형: 리플랫폼	워크로드: Oracle	기술: 마이그레이션, 데이터베이스
AWS 서비스: AWS DMS, Amazon RDS		

요약

[PeopleSoftOracle](#)은 전사적 프로세스를 위한 전사적 자원 관리 (ERP) 솔루션입니다. PeopleSoft 클라이언트, 애플리케이션, 데이터베이스의 3계층 아키텍처를 갖추고 있습니다. PeopleSoft [아마존 관계형 데이터베이스 서비스 \(Amazon RDS\)](#) 에서 실행할 수 있습니다.

Oracle 데이터베이스를 Amazon RDS로 마이그레이션하면 Amazon Web Services (AWS) 에서 백업 작업과고가용성을 처리하므로 사용자는 PeopleSoft 애플리케이션과 해당 기능을 유지 관리하는 데 집중할 수 있습니다. 마이그레이션 프로세스 중에 고려해야 할 주요 요소의 포괄적인 목록은 AWS 권장 가이드의 [Oracle 데이터베이스 마이그레이션 전략](#)을 참고하십시오.

이 패턴은 [AWS Database Migration Service\(AWS DMS\)](#) 및 변경 데이터 캡처(CDC) 기능을 갖춘 Oracle Data Pump를 사용하여 온프레미스 Oracle 데이터베이스를 Amazon RDS for Oracle로 마이그레이션하기 위한 솔루션을 제공합니다.

PeopleSoftOracle과 같은 중요한 ERP 애플리케이션을 마이그레이션할 때는 다운타임을 최소화하는 것이 중요합니다. AWS DMS는 원본 데이터베이스에서 대상 데이터베이스로의 전체 로드와 연속 복제를 모두 지원하여 가동 중지 시간을 최소화합니다. 또한, AWS DMS는 마이그레이션에 대한 실시간 모니터링 및 로깅을 제공하므로 가동 중지 시간을 유발할 수 있는 문제를 식별하고 해결하는 데 도움이 될 수 있습니다.

AWS DMS로 변경 내용을 복제할 때는 AWS DMS가 데이터베이스 로그에서 변경 내용을 읽기 위한 시작 지점으로 시간 또는 SCN(시스템 변경 번호)을 지정해야 합니다. AWS DMS가 이러한 변경 사항에 액세스할 수 있도록 하려면 지정된 시간 동안 서버에서 이러한 로그에 액세스할 수 있도록 유지하는 것이 중요합니다.

사전 조건 및 제한 사항

사전 조건

- 대상 데이터베이스로 AWS 클라우드 환경에서 프로비저닝된 Amazon RDS for Oracle 데이터베이스.
- 온프레미스 또는 AWS 클라우드의 Amazon Elastic Compute Cloud (Amazon EC2) 에서 실행되는 오라클 PeopleSoft 데이터베이스입니다.

참고: 이 패턴은 온프레미스에서 AWS로 마이그레이션하기 위해 설계되었지만 Amazon EC2 인스턴스의 Oracle Database를 사용하여 테스트되었습니다. 온프레미스에서 마이그레이션하려면 적절한 네트워크 연결을 구성해야 합니다.

- 스키마 세부 정보. Oracle PeopleSoft 애플리케이션을 Amazon RDS for Oracle로 마이그레이션할 때는 마이그레이션할 Oracle 데이터베이스 스키마 (예:SYSADM) 를 식별해야 합니다. 마이그레이션 프로세스를 시작하기 전에 스키마에 대한 다음과 같은 세부 정보를 수집합니다.
 - 크기
 - 객체 유형당 객체 수
 - 잘못된 객체 수

이 정보는 마이그레이션 프로세스에 도움이 됩니다.

제한 사항

- 이 시나리오는 PeopleSoft DEMO 데이터베이스에서만 테스트되었습니다. 대규모 데이터 세트로는 테스트되지 않았습니다.

아키텍처

다음 다이어그램은 Oracle 데이터베이스에서 EnterpriseOne을 원본 데이터베이스로 실행하고 Amazon RDS for Oracle 데이터베이스를 대상 데이터베이스로 실행하는 시스템을 보여줍니다. 데이터를 원본 Oracle 데이터베이스에서 내보내고, Oracle Data Pump를 사용하여 대상 Amazon RDS for Oracle 데이터베이스로 가져오고, AWS DMS를 사용하여 CDC 업데이트를 위해 복제합니다.

1. 초기 단계는 Oracle Data Pump를 사용하여 소스 데이터베이스에서 데이터를 추출한 다음 Amazon RDS for Oracle 데이터베이스 대상으로 데이터를 전송하는 것입니다.

2. CDC 데이터는 원본 데이터베이스에서 AWS DMS의 원본 엔드포인트로 전송됩니다.
3. 소스 엔드포인트에서 데이터는 AWS DMS 복제 인스턴스로 전송되며, 여기서 복제 작업이 수행됩니다.
4. 복제 작업이 완료되면 데이터가 AWS DMS의 대상 엔드포인트로 전송됩니다.
5. 대상 엔드포인트에서 데이터는 Amazon RDS for Oracle용 데이터베이스 인스턴스로 전송됩니다.

도구

서비스

- [AWS Database Migration Service\(AWS DMS\)](#)를 사용하면 데이터 스토어를 AWS 클라우드로 마이그레이션하거나 클라우드와 온프레미스 설정 조합 간에 마이그레이션할 수 있습니다.
- [Oracle용 Amazon Relational Database Service\(RDS\)](#)는 AWS 클라우드에서 Oracle 관계형 데이터베이스를 설정, 운영 및 조정하는 데 도움이 됩니다.

기타 서비스

- [Oracle Data Pump](#)를 사용하면 한 데이터베이스에서 다른 데이터베이스로 데이터와 메타데이터를 빠른 속도로 이동할 수 있습니다.

모범 사례

LOB 마이그레이션

소스 데이터베이스에 대상 데이터베이스로 마이그레이션해야 하는 대형 바이너리 객체(LOB)가 포함된 경우, AWS DMS는 다음과 같은 옵션을 제공합니다.

- 전체 LOB 모드 – AWS DMS는 모든 LOB를 크기에 관계없이 소스에서 대상 데이터베이스로 마이그레이션합니다. 마이그레이션이 더 느리지만 데이터가 잘리지 않는다는 장점이 있습니다. 성능 향상을 위해 새 복제 인스턴스에서 별도의 작업을 생성하여 LOB가 몇 메가바이트보다 큰 테이블을 마이그레이션할 수 있습니다.
- 제한된 LOB 모드 — LOB 열 데이터의 최대 크기를 지정하면 AWS DMS가 리소스를 사전 할당하고 LOB를 대량으로 적용할 수 있습니다. LOB 열의 크기가 작업에 지정된 크기를 초과하는 경우 AWS DMS는 데이터를 자르고 AWS DMS 로그 파일에 경고를 보냅니다. LOB 데이터 크기가 제한된 LOB 크기 이내인 경우 제한된 LOB 모드를 사용하여 성능을 개선할 수 있습니다.

- 인라인 LOB 모드 - 작은 LOB와 큰 LOB를 모두 복제하여 데이터를 자르거나 작업 성능을 저하시키지 않고 LOB를 마이그레이션할 수 있습니다. 먼저, Full LOB 모드가 true로 설정된 경우에만 사용할 수 있는 InlineLobMaxSize 매개변수 값을 지정합니다. AWS DMS 작업은 소규모 LOB를 인라인으로 전송하므로 더 효율적입니다. 그런 다음 AWS DMS는 소스 테이블에서 조회를 수행하여 대규모 LOB를 마이그레이션합니다. 하지만 인라인 LOB 모드는 전체 로드 단계에서만 작동합니다.

시퀀스 값 생성

AWS DMS를 사용한 변경 데이터 캡처 프로세스 중에는 원본 데이터베이스에서 중복 시퀀스 번호가 복제되지 않는다는 점에 유의하세요. 시퀀스 값의 불일치를 방지하려면 모든 시퀀스의 소스에서 가장 최근의 시퀀스 값을 생성하여 대상 Amazon RDS for Oracle 데이터베이스에 적용해야 합니다.

보안 인증 정보 관리

AWS 리소스를 보호하는 데 도움이 되도록 AWS IAM(Identity and Access Management) [모범 사례](#)를 따르는 것을 권장합니다.

에픽

소스 및 대상 엔드포인트로 AWS DMS 복제 인스턴스를 프로비저닝합니다.

작업	설명	필요한 기술
템플릿을 다운로드합니다.	DMS_Instance.yaml AWS CloudFormation 템플릿을 다운로드하여 AWS DMS 복제 인스턴스 와 해당 소스 및 대상 엔드포인트를 프로비저닝합니다.	클라우드 관리자, DBA
스택 생성을 시작합니다.	<ol style="list-style-type: none"> AWS 관리 콘솔에서 이를 선택합니다 CloudFormation. 스택 생성을 선택합니다. 템플릿 지정에서 템플릿 파일 업로드를 선택합니다. 파일 선택을 선택합니다. DMS_instance.yaml 파일을 선택합니다. 	클라우드 관리자, DBA

작업	설명	필요한 기술
	6. 다음을 선택합니다.	

작업	설명	필요한 기술
<p>파라미터를 지정합니다.</p>	<ol style="list-style-type: none"> 1. 스택 이름에 스택 이름을 입력합니다. 2. AWS DMS 인스턴스 파라미터에 다음과 같은 파라미터를 입력합니다. <ul style="list-style-type: none"> • DMS InstanceType — 비즈니스 요구 사항에 따라 AWS DMS 복제 인스턴스에 필요한 인스턴스를 선택합니다. • DMS StorageSize — 마이그레이션 규모에 따라 AWS DMS 인스턴스의 스토리지 크기를 입력합니다. 3. 원본 Oracle 데이터베이스 구성에는 다음과 같은 매개변수를 입력합니다. <ul style="list-style-type: none"> • SourceOracleEndpointID — 원본 Oracle 데이터베이스 서버 이름 • SourceOracleDatabaseName- 해당하는 경우 소스 데이터베이스 서버 이름 또는 세션 ID (SID) • SourceOracleUsername— 소스 데이터베이스 사용자 이름 (기본값은 system) • SourceOracleDbPassword — 소스 데이터베이스 	<p>클라우드 관리자, DBA</p>

작업	설명	필요한 기술
	<p>스 사용자 이름의 비밀번호입니다.</p> <ul style="list-style-type: none"> SourceOracleDBPort — 소스 데이터베이스 포트입니다. <p>4. 원본 Oracle 데이터베이스 구성에는 다음과 같은 매개 변수를 입력합니다.</p> <ul style="list-style-type: none"> 대상 RDS OracleEndpoint ID — 대상 RDS 데이터베이스 엔드포인트 대상 RDS OracleDatabaseName — 대상 RDS 데이터베이스 이름 대상 RDS — 대상 RDS 사용자 이름 OracleUsername TargetRDSOracleDBPassword — 대상 RDS 비밀번호 TargetOracleDBPort — 대상 RDS 데이터베이스 포트입니다. <p>5. VPC, 서브넷 및 보안 그룹 구성에서 다음 파라미터를 입력합니다.</p> <ul style="list-style-type: none"> VPCID — 복제 인스턴스를 위한 VPC VPC SecurityGroupID — 복제 인스턴스의 VPC 보안 그룹 DMSSubnet1 — 가용 영역 1을 위한 서브넷 	

작업	설명	필요한 기술
	<ul style="list-style-type: none"> DMSSubnet2 – 가용 영역 2를 위한 서브넷 <p>6. 다음을 선택합니다.</p>	
스택을 생성합니다.	<ol style="list-style-type: none"> 구성 스택 옵션 페이지에서 태그에 선택적 값을 입력합니다. 다음을 선택합니다. 검토 페이지에서 세부 정보를 확인한 다음, 제출을 선택합니다. <p>프로비저닝은 약 5~10분 내에 완료될 것입니다. AWS CloudFormation 스택 페이지에 CREATE_COMPLETE가 표시되면 작업이 완료됩니다.</p>	클라우드 관리자, DBA
엔드포인트를 설정합니다.	<ol style="list-style-type: none"> AWS 관리 콘솔에서 데이터베이스 마이그레이션 서비스를 선택합니다. 리소스 관리에서 복제 인스턴스를 선택합니다. 리소스 관리에서 엔드포인트를 선택합니다. 	클라우드 관리자, DBA
연결을 테스트합니다.	원본 및 대상 엔드포인트의 상태가 활성으로 표시되면 연결을 테스트합니다. 각 엔드포인트(소스 및 대상)에 대해 테스트 실행을 선택하여 상태가 성공으로 표시되는지 확인합니다.	클라우드 관리자, DBA

Oracle 데이터 펌프를 사용하여 온프레미스 Oracle 데이터베이스에서 PeopleSoft 스키마를 내보냅니다.

작업	설명	필요한 기술
<p>SCN을 생성합니다.</p>	<p>원본 데이터베이스가 활성 상태이고 EnterpriseOne 애플리케이션에서 사용 중인 경우 Oracle Data Pump를 사용하여 데이터 내보내기를 시작하세요. 먼저 Oracle Data Pump를 사용하여 내보내는 동안 데이터 일관성을 유지하고 AWS DMS에서 CDC의 시작점으로 사용하기 위해 원본 데이터베이스에서 시스템 변경 번호 (SCN)를 생성해야 합니다.</p> <p>원본 데이터베이스에서 현재 SCN을 생성하려면 다음 SQL 문을 사용합니다.</p> <pre data-bbox="594 1150 1027 1669"> SQL> select name from v \$database; SQL> select name from v \$database; NAME ----- PSFTDMO SQL> SELECT current_s cn FROM v\$database; CURRENT_SCN ----- 23792008 </pre> <p>데이터를 내보내고 AWS DMS 복제 작업을 생성할 때 사용할 생성된 SCN을 저장합니다.</p>	<p>DBA</p>

작업	설명	필요한 기술
<p>파라미터 파일을 생성합니다.</p>	<p>스키마를 내보내기 위한 매개 변수 파일을 만들려면 다음 코드를 사용할 수 있습니다.</p> <pre data-bbox="597 394 1024 869"> \$ cat exp_datapmp.par userid=system/***** directory=DATA_P UMP_DIR logfile=export_dms_ sample_user.log dumpfile=export_dms_ sample_data_%U.dmp schemas=SYSADM flashback_scn=237920 08 </pre> <p>참고: 요구 사항에 따라 다음 명령을 사용하여 나만의 DATA_PUMP_DIR 을 정의할 수도 있습니다.</p> <pre data-bbox="597 1129 1024 1814"> SQL> CREATE OR REPLACE DIRECTORY DATA_PUMP _DIR AS '/opt/oracle/ product/19c/dbhome_1/ dmsdump/'; Directory created. SQL> GRANT READ, WRITE ON DIRECTORY DATA_PUMP _DIR TO system; Grant succeeded. SQL> SQL> SELECT owner, directory_name, directory_path FROM dba_directories WHERE directory_name='DA TA_PUMP_DIR'; </pre>	<p>DBA</p>

작업	설명	필요한 기술
	<pre>OWNER DIRECTORY_NAME DIRECTORY_PATH ----- ----- ----- ----- ----- ----- ----- SYS DATA_PUMP_DIR /opt/ oracle/product/19c/dbh ome_1/dmsdump/</pre>	

작업	설명	필요한 기술
스키마를 내보냅니다.	<p>내보내기를 수행하려면 다음과 같이 expdp 유틸리티를 사용하세요.</p> <pre data-bbox="592 394 1027 1877"> \$ expdp parfile=e xp_datapmp.par Transferring the dump file with DBMS_FILE _TRANSFER to Target: . . exported "SYSADM". "PS_XML_TEMPLT_LNG" 6.320 KB 0 rows . . exported "SYSADM". "PS_XML_TEMPLT_LNK" 6.328 KB 0 rows . . exported "SYSADM". "PS_XML_XLATDEF_LNG" 6.320 KB 0 rows . . exported "SYSADM". "PS_XML_XLATITM_LNG" 7.171 KB 0 rows . . exported "SYSADM". "PS_XPQRYRUNCNTL" 7.601 KB 0 rows . . exported "SYSADM". "PS_XPQRYRUNPARM" 7.210 KB 0 rows . . exported "SYSADM". "PS_YE_AMOUNTS" 9.351 KB 0 rows . . exported "SYSADM". "PS_YE_DATA" 16.58 KB 0 rows . . exported "SYSADM". "PS_YE_EE" 6.75 KB 0 rows . . exported "SYSADM". "PS_YE_W2CP_AMOUNTS" 9.414 KB 0 rows </pre>	DBA

작업	설명	필요한 기술
	<pre> . . exported "SYSADM". "PS_YE_W2CP_DATA" 20.94 KB 0 rows . . exported "SYSADM". "PS_YE_W2C_AMOUNTS" 10.27 KB 0 rows . . exported "SYSADM". "PS_YE_W2C_DATA" 20.95 KB 0 rows . . exported "SYSADM". "PS_ZBD_JOBCODE_TBL" 14.60 KB 0 rows . . exported "SYSADM". "PTGRANTTBL" 5.468 KB 0 rows Master table "SYSTEM". "SYS_EXPORT_SCHEMA _01" successfully loaded/unloaded ** Dump file set for SYSTEM.SYS_EXPORT_ SCHEMA_01 is: /opt/oracle/pr oduct/19c/dbhome_1 /dmsdump/export_dm s_sample_data_01.dmp Job "SYSTEM"."SYS_EXPO RT_SCHEMA_01" successfully completed at Mon Dec 19 20:13:57 2022 elapsed 0 00:38:22 </pre>	

Oracle 데이터 펌프를 사용하여 Amazon RDS for Oracle용 데이터베이스로 PeopleSoft 스키마를 가져옵니다.

작업	설명	필요한 기술
<p>덤프 파일을 대상 인스턴스로 전송합니다.</p>	<p>DBMS_FILE_TRANSFER 유틸리티를 사용하여 파일을 전송하려면 원본 데이터베이스에서 Amazon RDS for Oracle 인스턴스로 데이터베이스 링크를 생성해야 합니다. 이 링크가 설정되면 유틸리티를 사용하여 데이터 펌프 파일을 RDS 인스턴스로 직접 전송할 수 있습니다.</p> <p>또는, 데이터 펌프 파일을 Amazon Simple Storage Service(Amazon S3) 로 전송한 다음 이 파일을 Amazon RDS for Oracle 인스턴스로 가져올 수 있습니다. 이 옵션에 대한 자세한 내용은 추가 정보 섹션을 참고하십시오.</p> <p>다음 명령은 대상 DB 인스턴스의 Amazon RDS 마스터 사용자에게 연결하는 ORARDSDB라는 데이터베이스 링크를 생성합니다.</p> <pre data-bbox="597 1549 1029 1881"> \$sqlplus / as sysdba \$ SQL> create database link orardsdb connect to admin identified by "*****" using '(DESCRIP TION = (ADDRESS = (PROTOCOL = TCP)(HOST = testpsft.*****.u </pre>	<p>DBA</p>

작업	설명	필요한 기술
	<pre>s-west-2.rds.amazons.com)(PORT = 1521))(CONNECT_DATA = (SERVER = DEDICATED) (SERVICE_NAME = orcl)))'; Database link created.</pre>	
<p>데이터베이스 링크를 테스트합니다.</p>	<p>sqlplus를 사용하여 Amazon RDS for Oracle용 대상 데이터베이스에 연결할 수 있는지 확인할 수 있도록 데이터베이스 링크를 테스트합니다.</p> <pre>SQL> SQL> select name from v \$database@orardsdb; NAME ----- ORCL SQL></pre>	DBA

작업	설명	필요한 기술
<p>덤프 파일을 대상 데이터베이스로 전송합니다.</p>	<p>덤프 파일을 Amazon RDS for Oracle 데이터베이스에 복사하기 위해 기본 DATA_PUMP_DIR 디렉터를 사용하거나 다음과 같은 코드를 사용하여 나만의 디렉터를 생성할 수 있습니다.</p> <pre data-bbox="592 583 1027 825"> exec rdsadmin.rdsadmin_ util.create_directory(p_directory_name => 'TARGET_PUMP_DIR') ; </pre> <p>다음 스크립트는 orardsdb라는 데이터베이스 링크를 사용하여 원본 인스턴스에 있는 export_dms_sample_data_01.dmp 라는 덤프 파일을 대상 Amazon RDS for Oracle 데이터베이스로 복사합니다.</p> <pre data-bbox="592 1270 1027 1833"> \$ sqlplus / as sysdba SQL> BEGIN DBMS_FILE_TRANSFER .PUT_FILE(source_directory _object => 'DATA_PUMP_DIR', source_file_name => 'export_dms_sample_data_01.dmp', destination_directory _object => 'TARGET_PUMP_DIR', </pre>	DBA

작업	설명	필요한 기술
	<pre> destination_file_name => 'export_dms_sample _data_01.dmp', destination_database => 'orardsdb'); END; / PL/SQL procedure successfully completed . </pre>	
<p>대상 데이터베이스에 있는 덤프 파일을 나열합니다.</p>	<p>PL/SQL 절차가 완료된 후 다음 코드를 사용하여 Amazon RDS for Oracle 데이터베이스에 데이터 덤프 파일을 나열할 수 있습니다.</p> <pre> SQL> select * from table (rdsadmin.rds_file _util.listdir(p_di rectory => 'TARGET_P UMP_DIR')); </pre>	DBA

작업	설명	필요한 기술
<p>대상 데이터베이스에서 가져오기를 시작합니다.</p>	<p>가져오기 프로세스를 시작하기 전에 데이터 덤프 파일을 사용하여 대상 Amazon RDS for Oracle 데이터베이스에 역할, 스키마 및 테이블스페이스를 설정합니다.</p> <p>가져오기를 수행하려면 Amazon RDS 마스터 사용자 계정으로 대상 데이터베이스에 액세스하고 tnsnames.ora 파일에서 Amazon RDS for Oracle 데이터베이스 tnsentry 를 포함하는 연결 문자열 이름을 사용합니다. 필요한 경우 다른 테이블스페이스나 다른 스키마 이름으로 데이터 덤프 파일을 가져오는 재매핑 옵션을 포함할 수 있습니다.</p> <p>가져오기를 시작하려면 다음 코드를 사용합니다.</p> <pre data-bbox="592 1270 1031 1549"> impdp admin@orardsdb directory=TARGET_P UMP_DIR logfile=i mport.log dumpfile= export_dms_sample_ data_01.dmp </pre> <p>가져오기에 성공하려면 가져오기 로그 파일에 오류가 있는지 확인하고 객체 수, 행 수, 잘못된 객체 등의 세부 정보를 검토하십시오. 잘못된 객체가 있는 경우 해당 객체를 다시 컴파일</p>	<p>DBA</p>

작업	설명	필요한 기술
	<p>하십시오. 또한 소스 데이터베이스 객체와 대상 데이터베이스 객체를 비교하여 일치하는지 확인하십시오.</p>	

실시간 복제를 수행하는 CDC를 사용하여 AWS DMS 복제 작업을 생성합니다.

작업	설명	필요한 기술
<p>복제 작업을 생성합니다.</p>	<p>다음 단계를 사용하여 AWS DMS 복제 작업을 생성합니다.</p> <ol style="list-style-type: none"> 1. AWS DMS 콘솔의 전환 및 마이그레이션에서 데이터베이스 마이그레이션 작업을 선택합니다. 2. 작업 구성 항목의 작업 식별자에 작업 식별자를 입력합니다. 3. 복제 인스턴스에는 생성한 DMS 복제 인스턴스를 선택합니다. 4. 소스 데이터베이스 엔드포인트에는 소스 엔드포인트를 선택합니다. 5. 대상 데이터베이스 엔드포인트에는 대상 Amazon RDS for Oracle 데이터베이스를 선택합니다. 6. 마이그레이션 유형에는 데이터 변경 내용만 복제를 선택합니다. 추가 로깅을 켜야 한다는 메시지가 표시되면 	<p>클라우드 관리자, DBA</p>

작업	설명	필요한 기술
	<p>문제 해결 섹션의 지침을 따르세요.</p> <p>7. 작업 설정 항목에서 로그 시퀀스 번호 지정을 선택합니다.</p> <p>8. 시스템 변경 번호에는 소스 Oracle 데이터베이스에서 생성한 Oracle 데이터베이스 SCN을 입력합니다.</p> <p>9. 검증 활성화를 선택합니다.</p> <p>10[CloudWatch 로그 활성화]를 선택합니다.</p> <p>이 기능을 활성화하면 데이터와 Amazon CloudWatch 로그를 검증하여 AWS DMS 복제 인스턴스 로그를 검토할 수 있습니다.</p> <p>11.선택 규칙에서 다음 사항을 완료하십시오.</p> <ul style="list-style-type: none"> • 스키마에 대해 스키마 입력을 선택합니다. • 스키마 이름에 SYSADM을 입력합니다. • 테이블 이름에 %를 입력합니다. • 작업에서 포함을 선택합니다. <p>12.변환 규칙에서 다음을 완료하십시오.</p> <ul style="list-style-type: none"> • 대상에서 테이블을 선택합니다. 	

작업	설명	필요한 기술
	<ul style="list-style-type: none"> 스키마 이름에서 스키마 입력을 선택합니다. 스키마 이름에 SYSADM 을 입력합니다. 작업에서 이름 바꾸기를 선택합니다. <p>13.작업 생성을 선택합니다.</p> <p>작업을 생성하면 CDC 시작 모드에서 사용자가 제공한 SCN에서 Amazon RDS for Oracle 데이터베이스 인스턴스로 CDC를 마이그레이션합니다. 로그를 검토하여 확인할 수도 있습니다. CloudWatch</p>	

대상 Amazon RDS for Oracle 데이터베이스에서 데이터베이스 스키마를 확인합니다.

작업	설명	필요한 기술
<p>데이터 전송을 검증합니다.</p>	<p>AWS DMS 작업이 시작된 후 작업 페이지의 테이블 통계 탭을 확인하여 데이터에 대한 변경 사항을 확인할 수 있습니다.</p> <p>콘솔의 데이터베이스 마이그레이션 작업 페이지에서 진행 중인 복제 상태를 모니터링할 수 있습니다.</p> <p>자세한 내용은 AWS DMS 데이터 검증 섹션을 참조하세요.</p>	<p>클라우드 관리자, DBA</p>

전환

작업	설명	필요한 기술
복제를 중지합니다.	복제 절차를 중단하고 원본 애플리케이션 서비스를 중지합니다.	클라우드 관리자, DBA
PeopleSoft 중간 계층을 실행하세요.	AWS에서 대상 PeopleSoft 미들 티어 애플리케이션을 시작하고 최근에 마이그레이션한 Amazon RDS for Oracle용 데이터베이스로 전송하십시오. 애플리케이션에 액세스하면 이제 모든 앱 연결이 Amazon RDS for Oracle 데이터베이스와 설정되었음을 알 수 있습니다.	DBA, 관리자 PeopleSoft
소스 데이터베이스를 끕니다.	더 이상 연결되어 있지 않은지 확인한 후 원본 데이터베이스를 끌 수 있습니다.	DBA

관련 리소스

- [AWS Database Migration Service 시작하기](#)
- [AWS Database Migration Service의 모범 사례](#)
- [Oracle 데이터베이스를 AWS Cloud로 마이그레이션하기](#)

추가 정보

Amazon S3를 사용하여 파일 전송하기

Amazon S3에 파일을 전송하려면 AWS CLI 또는 Amazon S3 콘솔을 사용하면 됩니다. Amazon S3로 파일을 전송한 후, Amazon RDS for Oracle 인스턴스를 사용하여 Amazon S3에서 데이터 펌프 파일을 가져올 수 있습니다.

Amazon S3 통합을 대체 방법으로 사용하여 덤프 파일을 전송하기로 선택한 경우 다음 단계를 수행합니다.

1. S3 버킷을 생성합니다.
2. Oracle Data Pump를 사용하여 원본 데이터베이스에서 데이터를 내보냅니다.
3. S3 버킷에 Data Pump 파일을 업로드합니다.
4. S3 버킷에서 대상 Amazon RDS for Oracle 데이터베이스로 데이터 펌프 파일을 다운로드합니다.
5. 데이터 펌프 파일을 사용하여 가져오기를 수행합니다.

참고: S3와 RDS 인스턴스 간에 대용량 데이터 파일을 전송하려면 Amazon S3 Transfer Acceleration 기능을 사용하는 것을 권장합니다.

추가 로깅 활성화

진행 중인 복제를 위해 원본 데이터베이스에서 [보충 로깅](#)을 활성화하라는 경고 메시지가 나타납니다.

```
SQL> ALTER DATABASE ADD SUPPLEMENTAL LOG DATA (ALL) COLUMNS;  
SQL> ALTER DATABASE ADD SUPPLEMENTAL LOG DATA (PRIMARY KEY) COLUMNS;  
SQL> ALTER DATABASE ADD SUPPLEMENTAL LOG DATA (UNIQUE) COLUMNS;  
SQL> ALTER DATABASE ADD SUPPLEMENTAL LOG DATA (FOREIGN KEY) COLUMNS;  
SQL> ALTER DATABASE ADD SUPPLEMENTAL LOG DATA (PRIMARY KEY) COLUMNS  
SQL> ALTER DATABASE ADD SUPPLEMENTAL LOG DATA (UNIQUE) COLUMNS;
```

온프레미스 MySQL 데이터베이스를 Amazon RDS for MySQL로 마이그레이션

작성자: Lorenzo Mota(AWS)

환경: PoC 또는 파일럿	소스: 온프레미스 MySQL 데이터베이스	대상: Amazon RDS for MySQL
R 유형: 리플랫폼	워크로드: 오픈 소스	기술: 마이그레이션, 데이터베이스
AWS 서비스: Amazon RDS		

요약

이 패턴은 온프레미스 MySQL 데이터베이스를 MySQL용 Amazon Relational Database Service(RDS)로 마이그레이션하기 위한 지침을 제공합니다. 이 패턴은 전체 마이그레이션을 위해 AWS Database Migration Service(DMS) 또는 네이티브 MySQL 도구(예: mysqldbcopy 및 mysqldump)를 사용하는 방법을 설명합니다. 이 패턴은 주로 DBA와 솔루션스 아키텍트를 위한 것입니다. 소규모 또는 대규모 프로젝트에서 테스트 절차(최소 한 번의 테스트 주기를 권장함) 또는 최종 마이그레이션 절차로 사용할 수 있습니다.

사전 조건 및 제한 사항

사전 조건

- 활성 상태의 AWS 계정
- 온프레미스 데이터 센터의 MySQL 소스 데이터베이스

제한 사항

- 데이터베이스 크기 제한: 64TB

제품 버전

- MySQL 버전 5.5, 5.6, 5.7, 8.0. 지원되는 버전과 에디션의 최신 목록은 AWS 설명서의 [Amazon RDS의 MySQL](#)을 참조하세요. DMS를 사용하는 경우 현재 DMS에서 지원하는 MySQL 버전용 [DMS의 대상으로 MySQL 호환 데이터베이스 사용](#)도 참조하십시오.

아키텍처

소스 기술 스택

- 온프레미스 MySQL 데이터베이스

대상 기술 스택

- MySQL을 실행하는 Amazon RDS DB

대상 아키텍처

다음 다이어그램은 마이그레이션 후 대상 Amazon RDS for MySQL 구현을 보여줍니다.

AWS 데이터 마이그레이션 아키텍처

DMS 사용:

다음 다이어그램은 DMS를 사용하여 전환 때까지 전체 및 점증적 변경 사항을 전송할 때의 데이터 마이그레이션 아키텍처를 보여줍니다. 온프레미스에서 AWS로의 네트워크 연결은 요구 사항에 따라 달라지며 이 패턴의 범위를 벗어납니다.

네이티브 MySQL 도구 사용:

다음 다이어그램은 네이티브 MySQL 도구를 사용할 때의 데이터 마이그레이션 아키텍처를 보여줍니다. 내보내기 덤프 파일은 Amazon Simple Storage Service(S3)에 복사되고, 전환 전에 AWS의 Amazon RDS for MySQL 데이터베이스로 가져오기가 실행됩니다. 온프레미스에서 AWS로의 네트워크 연결은 요구 사항에 따라 달라지며 이 패턴의 범위를 벗어납니다.

참고:

- 가동 중지 요구 사항 및 데이터베이스 크기에 따라 DMS 또는 변경 데이터 캡처(CDC) 도구를 사용하면 전환 시간을 최소화할 수 있습니다. DMS는 새 대상으로의 전환 시간을 최소 수준(일반적으로 분)으로 줄이는 데 도움이 될 수 있습니다. 데이터베이스 크기와 네트워크 지연 시간이 짧을 경우 mysqldump 또는 mysqldbcopy를 사용한 오프라인 전략으로도 충분할 수 있습니다. (대략적인 시간을 확인하려면 테스트하는 것이 좋습니다.)
- 일반적으로 DMS와 같은 CDC 전략은 오프라인 옵션보다 더 많은 모니터링과 더 큰 복잡성이 필요합니다.

도구

- AWS 서비스: [AWS Database Migration Service \(DMS\)](#)를 사용하면 데이터 스토어를 클라우드로 마이그레이션하거나 클라우드와 온프레미스 설정 간에 데이터 스토어를 마이그레이션할 수 있습니다. DMS에서 지원하는 MySQL 소스 및 대상 데이터베이스에 대한 내용은 [MySQL 호환 데이터베이스를 AWS로 마이그레이션하기](#)를 참조하십시오. DMS에서 소스 데이터베이스를 지원하지 않는 경우 데이터를 마이그레이션하기 위한 다른 방법을 선택해야 합니다.
- 네이티브 MySQL 도구: [mysqldbcopy](#) 및 [mysqldump](#)
- 서드파티 도구: [페르코나 XtraBackup](#)

에픽

마이그레이션 계획

작업	설명	필요한 기술
데이터베이스 버전을 검증합니다.	소스 및 대상 데이터베이스 버전을 검증합니다.	DBA
하드웨어 요구 사항을 식별합니다.	대상 서버의 하드웨어 요구 사항을 식별합니다.	DBA, 시스템 관리자
스토리지 요구 사항을 식별합니다.	대상 데이터베이스에 관한 스토리지 요구 사항(예: 스토리지 유형 및 용량)을 식별합니다.	DBA, 시스템 관리자

작업	설명	필요한 기술
인스턴스 유형을 선택합니다.	용량, 스토리지 특성, 네트워킹 특성에 따라 타겟 인스턴스 유형을 선택합니다.	DBA, 시스템 관리자
네트워크 액세스 요구 사항을 식별합니다.	소스 및 대상 데이터베이스의 보안 요구 사항을 식별합니다.	DBA, 시스템 관리자
지원되지 않는 객체를 식별합니다.	지원되지 않는 객체(있는 경우)를 식별하고 마이그레이션 노력을 결정합니다.	DBA
종속성을 파악합니다.	원격 데이터베이스에 대한 종속성을 식별합니다.	DBA
애플리케이션 마이그레이션 전략을 결정합니다.	클라이언트 애플리케이션을 마이그레이션하기 위한 전략을 결정합니다.	DBA, 앱 소유자, 시스템 관리자

인프라 구성

작업	설명	필요한 기술
Virtual Private Cloud(VPC)를 생성합니다.	라우팅 테이블, 인터넷 게이트웨이, NAT 게이트웨이, 서브넷을 구성합니다. 자세한 내용은 Amazon RDS 문서에서 VPC 및 Amazon RDS 를 참조하십시오.	시스템 관리자
보안 그룹을 생성합니다.	요구 사항에 따라 포트 및 CIDR 범위 또는 특정 IP를 구성합니다. MySQL의 기본 포트는 3306입니다. 자세한 내용은 Amazon RDS 사용 설명서 의	시스템 관리자

작업	설명	필요한 기술
	보안 그룹으로 액세스 제어를 참조하세요.	
Amazon RDS for MySQL 인스턴스를 구성 및 시작합니다.	지침은 Amazon RDS 설명서의 Amazon RDS DB 인스턴스 생성 을 참고하십시오. 지원되는 버전을 확인합니다.	시스템 관리자

데이터 마이그레이션 - 옵션 1(네이티브 도구 사용)

작업	설명	필요한 기술
네이티브 MySQL 도구 또는 타사 도구를 사용하여 데이터베이스 객체 및 데이터를 마이그레이션합니다.	<p>자세한 지침은 mysqldbcopy, mysqldump 및 Percona (물리적 마이그레이션용) 와 같은 MySQL 도구에 대한 설명서를 참조하십시오. XtraBackup</p> <p>옵션에 대한 자세한 내용은 블로그 게시물 MySQL을 Amazon RDS for MySQL 또는 Amazon Aurora MySQL로 마이그레이션 하는 옵션을 참조하십시오.</p>	DBA

데이터 마이그레이션 - 옵션 2(DMS 사용)

작업	설명	필요한 기술
AWS DMS를 사용하여 데이터를 마이그레이션합니다.	자세한 지침은 AWS DMS 설명서 를 참조하세요.	DBA

전환 전에 예비 작업 수행

작업	설명	필요한 기술
객체 수 불일치를 수정합니다.	소스 데이터베이스와 새 대상 데이터베이스에서 객체 수를 수집합니다. 대상 데이터베이스에서 불일치를 수정합니다.	DBA
종속성을 확인합니다.	다른 데이터베이스와 주고받는 종속성(링크)이 유효하고 예상대로 작동하는지 확인합니다.	DBA
테스트를 수행합니다.	이것이 테스트 주기인 경우 쿼리 테스트를 수행하고, 지표를 수집하며, 문제를 해결합니다.	DBA

전환

작업	설명	필요한 기술
대상 데이터베이스로 전환합니다.	클라이언트 애플리케이션을 새 인프라로 전환합니다.	DBA, 앱 소유자, 시스템 관리자
테스트 지원을 제공합니다.	기능적 애플리케이션 테스트를 지원합니다.	DBA

프로젝트 닫기

작업	설명	필요한 기술
리소스 종료합니다.	마이그레이션을 위해 생성한 임시 AWS 리소스를 종료합니다.	DBA, 시스템 관리자

작업	설명	필요한 기술
프로젝트 문서를 검증합니다.	프로젝트 문서를 검토하고 검증하세요.	DBA, 앱 소유자, 시스템 관리자
지표를 수집합니다.	마이그레이션 시간, 수동 작업과 자동 작업의 퍼센티지, 비용 절감 등과 같은 지표를 수집합니다.	DBA, 앱 소유자, 시스템 관리자
프로젝트를 닫습니다.	프로젝트를 마무리하고 피드백을 제공하세요.	DBA, 앱 소유자, 시스템 관리자
소스 데이터베이스를 폐기합니다.	모든 마이그레이션 및 전환 작업이 완료되면 온프레미스 데이터베이스를 폐기합니다.	DBA, 시스템 관리자

관련 리소스

참조

- [Migration strategy for relational databases](#)
- [AWS DMS 웹사이트](#)
- [DMS 설명서](#)
- [Amazon RDS 설명서](#)
- [Amazon RDS 요금 책정](#)
- [VPC 및 Amazon RDS](#)
- [Amazon RDS 다중 AZ 배포](#)
- [페르코나, Amazon EFS 및 Amazon S3를 사용하여 온프레미스 MySQL 데이터베이스를 Aurora MySQL로 XtraBackup 마이그레이션합니다.](#)

Tutorials

- [DMS 시작하기](#)
- [Amazon RDS 시작](#)

온프레미스 Microsoft SQL Server 데이터베이스를 Amazon RDS for SQL Server로 마이그레이션

작성자: 엔리케 로바오(AWS), 조나단 페레이라 크루즈(AWS), 비샬 싱(AWS)

환경: PoC 또는 파일럿	소스: Microsoft SQL Server	대상: Amazon RDS for SQL Server
R 유형: 리플랫폼	워크로드: Microsoft	기술: 마이그레이션, 데이터베이스

AWS 서비스: Amazon RDS

요약

이 패턴은 온프레미스 Microsoft SQL Server 데이터베이스를 SQL Server용 Amazon Relational Database Service(RDS)로 마이그레이션하기 위한 지침을 제공합니다. 여기에는 두 가지 마이그레이션 옵션이 포함됩니다. 하나는 AWS Data Migration Service(AWS DMS)를 사용하는 것이고 다른 하나는 Copy Database Wizard와 같은 기본 Microsoft SQL Server 도구를 사용하는 것입니다.

사전 조건 및 제한 사항

사전 조건

- 활성 상태의 AWS 계정
- 온프레미스 데이터 센터의 Microsoft SQL Server 소스 데이터베이스

제한 사항

- 데이터베이스 크기 제한: 16TB

제품 버전

- SQL Server 2014-2019, Enterprise, Standard, Workgroup, 및 Developer 에디션입니다. 지원되는 버전 및 기능의 최신 목록은 AWS 설명서에서 [Amazon RDS의 Microsoft SQL Server](#)를 참조하세요. AWS DMS를 사용하는 경우, AWS DMS에서 지원하는 SQL 서버 버전용 [AWS DMS의 대상으로 Microsoft SQL 서버 데이터베이스를 사용하기](#)를 참조하세요.

아키텍처

소스 기술 스택

- 온프레미스 Microsoft SQL Server 데이터베이스

대상 기술 스택

- Amazon RDS for SQL Server DB 인스턴스

소스 및 대상 아키텍처

AWS DMS 사용:

기본 SQL 서버 도구 사용:

도구

- [AWS DMS](#)는 여러 유형의 소스 및 대상 데이터베이스를 지원합니다. 자세한 내용은 [AWS DMS 단계별 안내](#)를 참조하세요. AWS DMS가 소스 데이터베이스를 지원하지 않는 경우, 다른 데이터 마이그레이션 방법을 선택합니다.
- 기본 Microsoft SQL Server 도구에는 백업 및 복원, Copy Database Wizard, 데이터베이스 복사 및 연결이 포함됩니다.

에픽

마이그레이션 계획

작업	설명	필요한 기술
소스 및 대상 데이터베이스 버전과 엔진의 유효성을 확인합니다.		DBA

작업	설명	필요한 기술
대상 서버 인스턴스의 하드웨어 요구 사항을 확인합니다.		DBA, 시스템 관리자
스토리지 요구 사항(스토리지 유형 및 용량)을 확인합니다.		DBA, 시스템 관리자
용량, 스토리지 기능, 네트워크 기능에 따라 적절한 인스턴스 유형을 선택합니다.		DBA, 시스템 관리자
소스 및 대상 데이터베이스의 네트워크 액세스 보안 요구 사항을 확인합니다.		DBA, 시스템 관리자
애플리케이션 마이그레이션 전략을 파악합니다.		DBA, 시스템 관리자

인프라 구성

작업	설명	필요한 기술
Virtual Private Cloud(VPC)를 생성합니다.		시스템 관리자
보안 그룹을 생성합니다.		시스템 관리자
Amazon RDS DB 인스턴스를 구성하고 시작합니다.		DBA, 시스템 관리자

데이터 마이그레이션 - 옵션 1

작업	설명	필요한 기술
기본 SQL Server 도구 또는 타사 도구를 사용하여 데이터베		DBA

작업	설명	필요한 기술
이스 객체 및 데이터를 마이그레이션할 수 있습니다.		

데이터 마이그레이션 - 옵션 2

작업	설명	필요한 기술
AWS DMS를 사용하여 데이터를 마이그레이션합니다.		DBA

애플리케이션 마이그레이션

작업	설명	필요한 기술
애플리케이션 마이그레이션 전략을 따릅니다.		DBA, 앱 소유자, 시스템 관리자

전환

작업	설명	필요한 기술
애플리케이션 클라이언트를 새 인프라로 전환합니다.		DBA, 앱 소유자, 시스템 관리자

프로젝트 닫기

작업	설명	필요한 기술
임시 AWS 리소스를 종료합니다.		DBA, 시스템 관리자
프로젝트 문서를 검토하고 검증하세요.		DBA, 앱 소유자, 시스템 관리자

작업	설명	필요한 기술
마이그레이션 시간, 수동 작업과 자동 작업의 비율, 비용 절감과 같은 지표를 수집합니다.		DBA, 앱 소유자, 시스템 관리자
프로젝트를 마무리하고 피드백을 제공하세요.		DBA, 앱 소유자, 시스템 관리자

관련 리소스

참조

- [Amazon Web Services에 Microsoft SQL Server 배포](#)
- [AWS DMS 웹사이트](#)
- [Amazon RDS 요금](#)
- [AWS의 Microsoft 제품](#)
- [AWS의 Microsoft 라이선싱](#)
- [AWS의 Microsoft SQL Server](#)
- [Microsoft SQL Server DB 인스턴스를 통한 Windows 인증 사용](#)
- [Amazon RDS 다중 AZ 배포](#)

자습서 및 동영상

- [AWS DMS 시작하기](#)
- [Amazon RDS 시작](#)
- [DMS \(동영상\)](#)
- [Amazon RDS\(동영상\)](#)

Rclone를 사용하여 Microsoft Azure Blob에서 Amazon S3로 데이터 마이그레이션하기

작성자: Suhas Basavaraj (AWS), Aidan Keane (AWS), Corey Lane (AWS)

환경: PoC 또는 파일럿	소스: Microsoft Azure 스토리지 컨테이너	대상: Amazon S3 버킷
R 유형: 리플랫폼	워크로드: Microsoft	기술: 마이그레이션, 스토리지 및 백업
AWS 서비스: Amazon S3		

요약

이 패턴은 Microsoft Azure Blob 객체 스토리지에서 Amazon Simple Storage Service(Amazon S3) 버킷으로 데이터를 마이그레이션하기 위해 [Rclone](#)을 사용하는 방법을 설명합니다. 이 패턴을 사용하여 데이터의 일회성 마이그레이션 또는 지속적인 동기화를 수행할 수 있습니다. Rclone은 Go로 작성된 명령줄 프로그램으로, 클라우드 공급자의 다양한 스토리지 기술 간에 데이터를 이동하는 데 사용됩니다.

사전 조건 및 제한 사항

사전 조건

- 활성 상태의 AWS 계정.
- Azure Blob 컨테이너 서비스에 저장된 데이터

아키텍처

소스 기술 스택

- Azure Blob 스토리지 컨테이너

대상 기술 스택

- Amazon S3 버킷

- Amazon Elastic Compute Cloud(Amazon EC2) Linux 인스턴스

아키텍처

도구

- [Amazon Simple Storage Service\(S3\)](#)는 원하는 양의 데이터를 저장, 보호 및 검색하는 데 도움이 되는 클라우드 기반 객체 스토리지 서비스입니다.
- [Rclone](#)은 rsync에서 영감을 받은 오픈 소스 명령줄 프로그램입니다. 여러 클라우드 스토리지 플랫폼에서 파일을 관리하는 데 사용됩니다.

모범 사례

Azure에서 Amazon S3로 데이터를 마이그레이션할 때는 불필요한 비용이나 느린 전송 속도를 방지하기 위해 다음 고려 사항을 염두에 두십시오.

- Azure 스토리지 계정 및 Blob 컨테이너와 동일한 지리적 리전(예: AWS 리전 us-east-1(버지니아 북부) 및 Azure 리전 East US)에 AWS 인프라를 생성합니다.
- NAT 게이트웨이는 수신 대역폭과 송신 대역폭 모두에 대해 데이터 전송 요금이 발생하므로 가능하면 사용하지 마십시오.
- [Amazon S3용 VPC 게이트웨이 엔드포인트](#)를 사용하여 성능을 향상시키십시오.
- Intel x86 인스턴스보다 비용을 낮추고 성능을 높이려면 AWS Graviton2(ARM) 프로세서 기반 EC2 인스턴스를 사용해 보십시오. Rclone은 대부분 크로스 컴파일되며 사전 컴파일된 ARM 바이너리를 제공합니다.

에픽

AWS 및 Azure 클라우드 리소스 준비

작업	설명	필요한 기술
대상 S3 버킷을 준비합니다.	적절한 AWS 리전에 새 S3 버킷을 생성 하거나 마이그레이션	AWS 관리자

작업	설명	필요한 기술
	<p>하려는 데이터의 대상으로 기존 버킷을 선택합니다.</p>	
<p>Amazon EC2의 IAM 인스턴스 역할을 생성합니다.</p>	<p>Amazon EC2에 대해 새로운 AWS Identity 및 Access Management(IAM) 역할을 생성합니다. 이 역할은 EC2 인스턴스에 대상 S3 버킷에 대한 쓰기 액세스 권한을 부여합니다.</p>	<p>AWS 관리자</p>
<p>IAM 인스턴스 역할에 정책을 연결합니다.</p>	<p>IAM 콘솔 또는 AWS Command Line Interface(AWS CLI)를 사용하여 대상 S3 버킷에 대한 쓰기 액세스 권한을 허용하는 EC2 인스턴스 역할에 대한 인라인 정책을 생성합니다. 예제 정책은 추가 정보 섹션을 참조하세요.</p>	<p>AWS 관리자</p>
<p>EC2 인스턴스를 시작합니다.</p>	<p>새로 생성한 IAM 서비스 역할을 사용하도록 구성된 Amazon Linux 2 EC2 인스턴스를 시작합니다. 또한, 이 인스턴스는 인터넷을 통해 Azure 공개 API 엔드포인트에 액세스해야 합니다.</p> <p>참고: 비용을 낮추려면 AWS Graviton 기반 EC2 인스턴스를 사용해 보십시오. Rclone은 ARM으로 컴파일된 바이너리를 제공합니다.</p>	<p>AWS 관리자</p>

작업	설명	필요한 기술
<p>Azure AD 서비스 보안 주체를 생성합니다.</p>	<p>Azure CLI를 사용하여 소스 Azure Blob 저장소 컨테이너에 대한 읽기 전용 액세스 권한이 있는 Azure Active Directory (Azure AD) 서비스 보안 주체를 만들 수 있습니다. 지침은 추가 정보 섹션을 참조하세요. 이 보안 인증 정보를 ~/azure-principal.json 위치의 EC2 인스턴스에 저장하십시오.</p>	<p>클라우드 관리자, Azure</p>

Rclone 설치 및 구성

작업	설명	필요한 기술
<p>Rclone을 다운로드하고 설치합니다.</p>	<p>Rclone 명령줄 프로그램을 다운로드하고 설치합니다. 설치 지침은 Rclone 설치 설명서를 참조하세요.</p>	<p>일반 AWS, 클라우드 관리자</p>
<p>Rclone을 구성합니다.</p>	<p>다음 rclone.conf 샘플 파일을 복사합니다. AZStorage Account 를 Azure Storage 계정 이름으로 바꾸고 us-east-1 을 S3 버킷이 위치한 AWS 리전으로 바꾸십시오. 이 파일을 EC2 인스턴스의 ~/.config/rclone/rclone.conf 위치에 저장합니다.</p> <div style="border: 1px solid #ccc; padding: 5px; margin-top: 10px;"> <pre>[AZStorageAccount] type = azureblob</pre> </div>	<p>일반 AWS, 클라우드 관리자</p>

작업	설명	필요한 기술
	<pre>account = AZStorage Account service_principal_file = azure-principal.json [s3] type = s3 provider = AWS env_auth = true region = us-east-1</pre>	

작업	설명	필요한 기술
Rclone 구성을 확인하십시오.	<p>Rclone이 구성되어 있고 권한이 제대로 작동하는지 확인하려면, Rclone이 구성 파일을 파싱할 수 있고 Azure Blob 컨테이너 및 S3 버킷 내의 객체에 액세스할 수 있는지 확인하십시오. 검증 명령 예는 다음을 참조하십시오.</p> <ul style="list-style-type: none"> 구성 파일에 구성된 원격 항목을 나열합니다. 이렇게 하면 구성 파일이 올바르게 파싱될 수 있습니다. 출력을 검토하여 <code>rclone.conf</code> 파일과 일치하는지 확인하십시오. <pre data-bbox="625 997 1031 1155">rclone listremotes AZStorageAccount: s3:</pre> <ul style="list-style-type: none"> 구성된 계정의 Azure Blob 컨테이너를 나열합니다. <code>rclone.conf</code> 파일에 사용한 저장소 계정 이름으로 <code>AZStorageAccount</code> 를 바꿉니다. <pre data-bbox="625 1480 1031 1680">rclone lsd AZStorage Account: 2020-04-29 08:29:26 docs</pre> <ul style="list-style-type: none"> Azure Blob 컨테이너에 있는 파일을 나열합니다. 이 명령의 문서를 Azure 저장소 계 	일반 AWS, 클라우드 관리자

작업	설명	필요한 기술
	<p>정의 실제 Blob 컨테이너 이름으로 바꿉니다.</p> <pre data-bbox="625 331 1029 529">rclone ls AZStorage Account:docs 824884 administrator-en.a4.pdf</pre> <ul style="list-style-type: none"> AWS 계정의 버킷을 나열합니다. <pre data-bbox="625 667 1029 1138">[root@ip-10-0-20-157 ~]# rclone lsd s3: 2022-03-07 01:44:40 examplebu cket-01 2022-03-07 01:45:16 examplebu cket-02 2022-03-07 02:12:07 examplebu cket-03</pre> <ul style="list-style-type: none"> S3 버킷에 있는 파일을 나열합니다. <pre data-bbox="625 1276 1029 1520">[root@ip-10-0-20-157 ~]# rclone ls s3:examplebucket-01 template0.yaml template1.yaml</pre>	

Rclone을 사용하여 데이터 마이그레이션하기

작업	설명	필요한 기술
컨테이너에서 데이터를 마이그레이션합니다.	Rclone 복사 또는 동기화 명령을 실행합니다.	일반 AWS, 클라우드 관리자

작업	설명	필요한 기술
	<p>예: 복사</p> <p>이 명령은 소스 Azure Blob 컨테이너의 데이터를 대상 S3 버킷으로 복사합니다.</p> <pre style="border: 1px solid #ccc; border-radius: 10px; padding: 10px;">rclone copy AZStorage Account:blob-conta iner s3:exampl ebucket-01</pre> <p>예: 동기화</p> <p>이 명령은 소스 Azure Blob 컨테이너와 대상 S3 버킷 간에 데이터를 동기화합니다.</p> <pre style="border: 1px solid #ccc; border-radius: 10px; padding: 10px;">rclone sync AZStorage Account:blob-conta iner s3:exampl ebucket-01</pre> <p>중요: 동기화 명령을 사용하면 소스 컨테이너에 없는 데이터가 대상 S3 버킷에서 삭제됩니다.</p>	
<p>컨테이너를 동기화합니다.</p>	<p>초기 복사가 완료된 후, Rclone 동기화 명령을 실행하여 대상 S3 버킷에서 누락된 새 파일만 복사되도록 지속적인 마이그레이션을 수행합니다.</p>	<p>일반 AWS, 클라우드 관리자</p>
<p>데이터가 성공적으로 마이그레이션되었는지 확인하십시오.</p>	<p>데이터가 대상 S3 버킷에 성공적으로 복사되었는지 확인하려면 Rclone lsd 및 ls 명령을 실행합니다.</p>	<p>일반 AWS, 클라우드 관리자</p>

관련 리소스

- [Amazon S3 사용 설명서](#) (AWS 설명서)
- [Amazon EC2의 IAM 역할](#) (Amazon 설명서)
- [Microsoft Azure Blob 컨테이너 만들기](#) (Microsoft Azure 설명서)
- [Rclone 명령](#) (Rclone 설명서)

추가 정보

EC2 인스턴스의 역할 정책 예시

이 정책은 EC2 인스턴스에 사용자 계정에 있는 특정 버킷에 대한 읽기 및 쓰기 권한을 부여합니다. 버킷이 서버 측 암호화를 위해 고객 관리형 키를 사용하는 경우 해당 정책상 AWS Key Management Service(AWS KMS)에 대한 추가 액세스가 필요할 수 있습니다.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "s3:ListBucket",
        "s3:DeleteObject",
        "s3:GetObject",
        "s3:PutObject",
        "s3:PutObjectAcl"
      ],
      "Resource": [
        "arn:aws:s3:::BUCKET_NAME/*",
        "arn:aws:s3:::BUCKET_NAME"
      ]
    },
    {
      "Effect": "Allow",
      "Action": "s3:ListAllMyBuckets",
      "Resource": "arn:aws:s3:::*"
    }
  ]
}
```

읽기 전용 Azure AD 서비스 보안 주체 생성

Azure 서비스 보안 주체는 고객 애플리케이션, 서비스 및 자동화 도구가 특정 Azure 리소스에 액세스하는 데 사용하는 보안 ID입니다. 특정 역할이 있고 리소스에 액세스할 수 있도록 엄격하게 제어되는 권한을 가진 사용자 ID(로그인 및 암호 또는 인증서)라고 생각하면 됩니다. 최소 권한을 따르고 Azure의 데이터가 실수로 삭제되지 않도록 보호하는 읽기 전용 서비스 보안 주체를 만들려면 다음 단계를 따릅니다.

1. Microsoft Azure 클라우드 계정 포털에 로그인하여 워크스테이션에서 클라우드 셸을 PowerShell 시작하거나 Azure 명령줄 인터페이스 (CLI) 를 사용하십시오.
2. 서비스 보안 주체를 만들고 Azure Blob 저장소 계정에 대한 [읽기 전용](#) 액세스 권한으로 구성하세요. 이 명령의 JSON 출력을 `azure-principal.json`이라는 로컬 파일에 저장합니다. 파일이 EC2 인스턴스에 업로드됩니다. 중괄호({ 및 })로 표시된 자리 표시자 변수를 Azure 구독 ID, 리소스 그룹 이름 및 스토리지 계정 이름으로 바꿉니다.

```
az ad sp create-for-rbac `
--name AWS-Rclone-Reader `
--role "Storage Blob Data Reader" `
--scopes /subscriptions/{Subscription ID}/resourceGroups/{Resource Group Name}/
providers/Microsoft.Storage/storageAccounts/{Storage Account Name}
```

카우치베이스 서버에서 AWS의 카우치베이스 카펠라로 마이그레이션

제작: 바툴가 푸레브라차 (AWS), 마크 갬블, 사우라브 산바그 (AWS)

환경: 프로덕션	소스: 카우치베이스 서버	대상: 카우치베이스 카펠라
R 유형: 리플랫폼	워크로드: 기타 모든 워크로드	기술: 마이그레이션, 분석, 데이터베이스

요약

Couchbase Capella는 업무상 중요한 애플리케이션(예: 사용자 프로필, 온라인 카탈로그 및 인벤토리 관리)을 위한 완전 관리형 NoSQL 서비스형 데이터베이스 (DBaaS)입니다. 카우치베이스 카펠라는 카우치베이스에서 관리하는 Amazon Web Services(AWS) 계정에서 DBaaS 워크로드를 관리합니다. Capella를 사용하면 단일 인터페이스 내에서 여러 클러스터, 다중 AWS 리전, 멀티클라우드 및 하이브리드 클라우드 복제를 쉽게 실행하고 관리할 수 있습니다.

Couchbase Capella를 사용하면 Couchbase 서버 애플리케이션을 즉시 확장하여 몇 분 만에 다중 노드 클러스터를 생성할 수 있습니다. Couchbase Capella는 [SQL++](#), [전체 텍스트 검색](#), [이벤트 서비스](#) 및 [분석 서비스](#)를 포함한 모든 Couchbase 서버 기능을 지원합니다. 또한 설치, 업그레이드, 백업 및 일반 데이터베이스 유지 관리를 관리할 필요도 없습니다.

이 패턴은 자체 관리형 [Couchbase 서버](#) 환경을 AWS 클라우드로 마이그레이션하는 단계 및 모범 사례를 설명합니다. 이 패턴은 온프레미스 또는 클라우드에서 실행되는 Couchbase 서버 클러스터에서 Couchbase Capella로 데이터와 인덱스를 마이그레이션하는 반복 가능한 프로세스를 제공합니다. 이 단계를 사용하면 마이그레이션 중에 발생하는 문제를 방지하고 전체 마이그레이션 프로세스를 가속화할 수 있습니다.

이 패턴은 다음과 같은 두 가지 마이그레이션 옵션을 제공합니다.

- 마이그레이션할 인덱스가 50개 미만인 경우 옵션 1이 적합합니다.
- 마이그레이션할 인덱스가 50개 이상인 경우 옵션 2가 적합합니다.

자체 관리형 Couchbase Server에서 마이그레이션 가이드에 따라 [샘플 데이터를 설정](#)할 수도 있습니다.

마이그레이션 옵션 2를 선택하거나 기본값이 아닌 범위 또는 컬렉션을 사용하는 경우 추가 정보 섹션에 있는 예제 구성 파일을 사용해야 합니다.

사전 조건 및 제한 사항

사전 조건

- 기존 카우치베이스 카펠라 유료 계정. 또한 [AWS에서 Couchbase Capella](#) 계정을 생성하고 Couchbase Capella 무료 평가판을 사용한 다음 유료 계정으로 업그레이드하여 마이그레이션을 위한 클러스터를 구성할 수 있습니다. 평가판을 시작하려면 [Couchbase Capella 시작하기](#)에 나와 있는 지침을 따르십시오.
- 기존의 자체 관리형 Couchbase 서버 환경(온프레미스 또는 클라우드 서비스 제공업체에 배포됨).
- 마이그레이션 옵션 2의 경우 Couchbase 셸 및 구성 파일이 필요합니다. 구성 파일을 만들려면 추가 정보 섹션에 있는 예제 파일을 사용할 수 있습니다.
- 카우치베이스 서버 및 카우치베이스 카펠라 관리에 익숙해야 합니다.
- 명령줄 인터페이스(CLI)에서 TCP 포트를 열고 명령을 실행하는 데 익숙합니다.

마이그레이션 프로세스에는 다음 표에 설명된 역할과 전문 지식도 필요합니다.

역할	전문성	책임
카우치베이스 관리자	<ul style="list-style-type: none"> • 카우치베이스 서버 및 카우치베이스 카펠라에 대한 지식 • 기본적인 명령줄 지식은 도움이 되지만 필수는 아닙니다. 	<ul style="list-style-type: none"> • 카우치베이스 서버 및 카펠라 관련 작업
시스템 관리자, IT 관리자	<ul style="list-style-type: none"> • 자체 관리형 Couchbase 서버 시스템 환경 및 관리에 대한 지식 	<ul style="list-style-type: none"> • 자체 관리형 Couchbase 서버 클러스터 노드의 포트 열기 및 IP 주소 결정

제한 사항

- 이 패턴은 데이터, 인덱스 및 [Couchbase 전체 텍스트 검색](#) 인덱스를 Couchbase 서버에서 AWS의 Couchbase Capella로 마이그레이션하는 데 사용됩니다. 이 패턴은 [Couchbase 이벤트 서비스](#) 마이그레이션이나 [Couchbase Analytics](#)에는 적용되지 않습니다.
- 카우치베이스 카펠라는 여러 AWS 리전에서 사용할 수 있습니다. 카펠라가 지원하는 지역에 대한 자세한 up-to-date 내용은 카우치베이스 설명서의 [Amazon Web Services](#)를 참조하십시오.

제품 버전

- [카우치베이스 서버 \(커뮤니티 또는 엔터프라이즈\) 에디션 버전 5.x 이상](#)

아키텍처

소스 기술 스택

- 카우치베이스 Server

대상 기술 스택

- 카우치베이스 카펠라

대상 아키텍처

1. 카펠라 컨트롤 플레인을 사용하여 카우치베이스 카펠라에 액세스할 수 있습니다. Capella 컨트롤 플레인을 사용하여 다음을 수행할 수 있습니다.
 - 계정을 관리하고 모니터링하세요.
 - 클러스터 및 데이터, 인덱스, 사용자 및 그룹, 액세스 권한, 모니터링, 이벤트를 관리합니다.
2. 클러스터가 생성됩니다.
3. 카펠라 데이터 영역은 카우치베이스에서 관리하는 AWS 계정에 있습니다. 새 클러스터를 생성한 후, Couchbase Capella는 선택한 AWS 리전의 여러 가용 영역에 클러스터를 배포합니다.
4. AWS 계정의 VPC에서 Couchbase 애플리케이션을 개발하고 배포할 수 있습니다. 일반적으로 이 VPC는 [VPC 피어링](#)을 통해 카펠라 데이터 영역에 액세스합니다.

도구

- [Couchbase 데이터 센터 간 복제 \(XDCR\)](#)를 사용하면 여러 클라우드 공급자와 데이터 센터에 위치한 클러스터 간에 데이터를 복제할 수 있습니다. 자체 관리형 Couchbase 서버 클러스터에서 Couchbase Capella로 데이터를 마이그레이션하는 데 사용됩니다.

참고: XDCR은 카우치베이스 서버 커뮤니티 에디션과 함께 사용하여 카우치베이스 카펠라로 마이그레이션할 수 없습니다. 대신 [cbexport](#)를 사용할 수 있습니다. 자세한 내용은 커뮤니티 에디션에서 데이터 마이그레이션 에픽을 참조하십시오.

- [Couchbase Shell](#)은 Couchbase 서버와 Couchbase Capella가 로컬 및 원격 Couchbase 클러스터에 액세스할 수 있는 명령줄 셸입니다. 이 패턴에서는 Couchbase 셸을 사용하여 인덱스를 마이그레이션합니다.
- [cbexport](#)는 Couchbase 클러스터에서 데이터를 익스포트하기 위한 카우치베이스 유틸리티입니다. [카우치베이스 서버 CLI 도구](#)에 포함됩니다.

에픽

마이그레이션 준비

작업	설명	필요한 기술
자가 관리형 카우치베이스 Server 클러스터의 크기를 평가합니다.	<p>Couchbase 서버용 Couchbase 웹 콘솔에 로그인하여 자체 관리형 클러스터의 노드와 버킷을 평가하십시오.</p> <ol style="list-style-type: none"> 1. 클러스터 노드 목록을 표시하려면 탐색 표시줄에서 서버 탭을 선택합니다. 2. 노드 수를 기록한 다음 목록에서 각 노드를 선택하여 해당 속성을 표시합니다. 3. 각 개별 노드의 메모리와 스토리지를 기록합니다. 4. 탐색 표시줄에서 Buckets 탭을 선택한 다음 목록에서 	카우치베이스 관리자

작업	설명	필요한 기술
	<p>각 버킷을 선택하여 해당 속성을 표시합니다. 각 버킷의 RAM 할당량 및 충돌 해결 설정을 기록해 둡니다.</p> <p>자체 관리형 Couchbase 서버 클러스터 구성을 Couchbase Capella에서 대상 클러스터의 크기를 조정하고 구성하기 위한 일반적인 지침으로 사용하게 됩니다.</p> <p>더 자세한 카우치베이스 카펠라 사이징 연습에 대한 도움이 필요하면 Couchbase에 문의하세요.</p>	
<p>자체 관리형 카우치베이스 서버 클러스터에서 카우치베이스 서비스 배포를 기록하세요.</p>	<ol style="list-style-type: none"> 1. Couchbase 웹 콘솔에서 서버 탭을 선택하여 클러스터 노드 목록을 표시합니다. 2. 속성을 표시할 각 노드를 선택한 다음 각 노드 (데이터 서비스, 쿼리 서비스, 인덱스 서비스, 검색 서비스, 분석 서비스, 이벤트 서비스)에 대한 Couchbase 서비스 배포를 기록합니다. 	<p>카우치베이스 관리자</p>
<p>자체관리형 카우치베이스 Server 클러스터 노드의 IP 주소를 기록합니다.</p>	<p>(커뮤니티 에디션을 사용하는 경우 이 단계를 무시하세요.) 클러스터의 각 노드에 대한 IP 주소를 기록하십시오. 나중에 Couchbase Capella 클러스터의 허용 목록에 추가될 예정입니다.</p>	<p>카우치베이스 관리자, 시스템 관리자</p>

카우치베이스 카펠라에 리소스를 배포하고 구성하세요.

작업	설명	필요한 기술
<p>템플릿(Template)을 선택합니다.</p>	<ol style="list-style-type: none"> 1. Couchbase Capella 콘트를 플레인에 로그인하고 기본 탐색에서 대시보드 탭 또는 클러스터 탭을 선택한 다음 클러스터 생성을 선택합니다. 2. 자체 관리형 Couchbase 서버 클러스터 평가에서 기록한 정보를 사용하여 구성 요구 사항을 충족하는 클러스터 템플릿을 선택합니다. 적절한 템플릿을 찾을 수 없는 경우 클러스터 크기 조정 편집기에서 사용자 지정 템플릿을 선택합니다. 	<p>카우치베이스 관리자</p>
<p>노드를 선택하고 구성합니다.</p>	<p>노드 수, 서비스 배포, 컴퓨팅 또는 RAM, 스토리지를 포함하여 자체 관리형 Couchbase Server 클러스터 환경에 맞게 노드를 선택하고 구성하십시오.</p> <p>카우치베이스 카펠라는 다차원 규모 조정 모범 사례를 사용합니다. 서비스와 노드는 배포 모범 사례에 따라서만 선택할 수 있습니다. 따라서 자체 관리형 Couchbase Server 클러스터의 구성과 정확히 일치하지 않을 수 있습니다.</p>	<p>카우치베이스 관리자</p>

작업	설명	필요한 기술
클러스터를 배포합니다.	<p>지원 영역과 지원 패키지를 선택한 다음 클러스터를 배포합니다. 자세한 단계 및 지침은 Couchbase 설명서의 클러스터 생성을 참조하십시오.</p> <p>중요: Couchbase Capella 무료 평가판을 사용하는 경우 마이그레이션을 시작하기 전에 유료 계정으로 전환해야 합니다. 계정을 전환하려면 Couchbase Capella 컨트롤 플레인의 결제 섹션을 연 다음 활성화 ID 추가를 선택합니다. Couchbase Sales와 구매 계약을 완료한 후 또는 AWS Marketplace를 통해 구매한 후에 활성화 ID가 청구 연락처 이메일 주소로 전송됩니다.</p>	카우치베이스 관리자

작업	설명	필요한 기술
<p>데이터베이스 보안 인증 사용자 생성.</p>	<p>데이터베이스 보안 인증 사용자는 클러스터에만 해당되며 사용자 이름, 암호 및 일련의 버킷 권한으로 구성됩니다. 이 사용자는 버킷을 생성하고 버킷 데이터에 액세스하는 데 필요합니다.</p> <p>Couchbase Capella 컨트롤 플레인에서 Couchbase Capella 설명서의 데이터베이스 보안 인증 구성의 지침에 따라 새 클러스터의 데이터베이스 보안 인증을 생성합니다.</p> <p>참고: 조직 사용자가 원격으로 또는 Couchbase Capella UI를 통해 특정 클러스터의 버킷 데이터에 액세스하려는 경우 조직 역할 보안 인증이 할당되어야 합니다. 이는 앱과 통합에서 일반적으로 사용되는 데이터베이스 보안 인증과는 별개입니다. 조직 사용자를 생성하면 Couchbase Capella 클러스터에서 대상 버킷을 생성하고 관리할 수 있습니다.</p>	<p>카우치베이스 관리자</p>

작업	설명	필요한 기술
<p>마이그레이션 옵션 2를 사용하는 경우 카우치베이스 셸을 설치하세요.</p>	<p>자체 관리형 Couchbase 서버와 Couchbase Capella 클러스터 모두에 네트워크 액세스가 가능한 모든 시스템에 Couchbase Shell을 설치할 수 있습니다. 자세한 내용은 카우치베이스 셸 설명서의 카우치베이스 셸 버전 1.0.0-beta.5 설치를 참조하십시오.</p> <p>명령줄 터미널에서 자체 관리형 클러스터에 대한 연결을 테스트하여 Couchbase Shell이 설치되었는지 확인합니다.</p>	<p>카우치베이스 관리자, 시스템 관리자</p>

작업	설명	필요한 기술
IP 주소 허용.	<ol style="list-style-type: none"> 1. Couchbase Capella 콘트를 플레인에서 클러스터를 선택한 다음 대상 클러스터를 선택합니다. 2. 클러스터의 Connect 탭을 선택하고 허용된 IP 관리에 나열된 클러스터의 연결 엔드포인트를 기록합니다. 3. Couchbase Shell을 설치한 시스템의 IP 주소와 자체 관리형 Couchbase 서버 클러스터 인스턴스의 IP 주소를 허용된 IP 주소로 추가하려면 다음과 같이 하십시오. <ol style="list-style-type: none"> a. 광역 네트워크에서 허용된 IP 관리를 선택합니다. b. 허용된 IP 추가를 선택하고 Couchbase Shell을 설치한 시스템의 IP 주소를 입력한 다음 IP 추가를 선택합니다. c. 이전 단계를 반복하여 자체 관리형 Couchbase 서버 클러스터 인스턴스의 IP 주소를 추가합니다. <p>허용된 IP 주소에 대한 자세한 내용은 Couchbase 설명서에서 허용된 IP 주소 구성을 참조하십시오.</p>	카우치베이스 관리자, 시스템 관리자

작업	설명	필요한 기술
인증서를 구성합니다.	<ol style="list-style-type: none"> 1. 클러스터의 루트 인증서를 다운로드하려면 루트 인증서에서 다운로드를 선택합니다. 2. .pem 파일 확장자를 사용하여 Couchbase Shell을 실행할 시스템의 폴더에 루트 인증서를 저장합니다. 3. 그런 다음 자체 관리형 Couchbase Server 웹 콘솔에 로그인하고 왼쪽 탐색 표시줄에서 보안을 선택한 다음 인증서 탭을 선택합니다. 4. 자체 관리형 Couchbase 서버 클러스터의 루트 인증서를 복사하여 Couchbase Capella 클러스터의 루트 인증서 파일을 저장한 폴더와 동일한 폴더에 .pem 파일로 저장합니다. 루트 인증서에 대한 자세한 내용은 Couchbase Server 설명서의 루트 인증서를 참조하세요. 	카우치베이스 관리자, 시스템 관리자

작업	설명	필요한 기술
<p>카우치베이스 셸에 구성 파일을 생성합니다.</p>	<p>Couchbase Shell 설치의 홈 디렉터리(예: /<HOME_DIRECTORY>/ .cbsh/config)에 구성 도트파일을 생성합니다. 자세한 내용은 Couchbase 문서에서 구성 닷 파일을 참조하세요.</p> <p>소스 및 대상 클러스터의 연결 속성을 구성 파일에 추가합니다. 추가 정보 섹션에 있는 예제 구성 파일을 사용하여 클러스터의 설정을 편집할 수 있습니다.</p> <p>업데이트된 설정이 포함된 구성 파일을 .cbsh 폴더에 저장합니다 (예: /<HOME_DIRECTORY>/ .cbsh/config).</p>	<p>카우치베이스 관리자, 시스템 관리자</p>
<p>대상 버킷을 생성합니다.</p>	<p>Couchbase 설명서의 버킷 생성에 나와 있는 지침에 따라 각 소스 버킷에 대해 Couchbase Capella 클러스터에 대상 버킷 하나를 생성합니다.</p> <p>대상 버킷 구성은 자체 관리형 Couchbase Server 클러스터에 있는 버킷의 버킷 이름, 메모리 설정 및 충돌 해결 설정과 일치해야 합니다.</p>	<p>카우치베이스 관리자</p>

작업	설명	필요한 기술
<p>스코프와 컬렉션을 생성하세요.</p>	<p>모든 버킷에는 <code>_default</code>. <code>_default</code> 키스페이스와 함께 기본 범위와 컬렉션이 포함되어 있습니다. 범위 및 컬렉션에 다른 키스페이스를 사용하는 경우 대상 Capella 클러스터에 동일한 키스페이스를 생성해야 합니다.</p> <ol style="list-style-type: none"> Couchbase Shell을 설치한 시스템에서 명령줄 터미널을 엽니다. 카우치베이스 셸을 시작하려면 다음 명령을 실행합니다. <div data-bbox="630 961 1029 1041" style="border: 1px solid #ccc; border-radius: 10px; padding: 5px; margin: 10px 0;"> <pre>./cbsh</pre> </div> 마이그레이션하려는 각 버킷에 대해 다음 명령을 실행하여 Capella 클러스터에서 범위와 컬렉션을 생성합니다. <code><BUCKET_NAME></code> 마이그레이션할 버킷의 이름으로 바꿉니다. <div data-bbox="597 1451 1029 1864" style="border: 1px solid #ccc; border-radius: 10px; padding: 10px; margin: 10px 0;"> <pre>scopes --clusters "On-Prem-Cluster" --bucket <BUCKET_NAME> select scope where scope != "_default" each { it scopes create \$it.scope --clusters "Capella-Cluster" } collections --clusters "On-Prem-Cluster"</pre> </div> 	<p>카우치베이스 관리자</p>

작업	설명	필요한 기술
	<pre> --bucket <BUCKET_NAME> select scope collection where \$it.scope != "_default" where \$it.collection != "_default" each { it collections create \$it.collection --clusters "Capella-Cluster" -- bucket <BUCKET_NAME> -- scope \$it.scope } </pre>	

엔터프라이즈 에디션의 데이터 이전

작업	설명	필요한 기술
<p>자체 관리형 카우치베이스 서버 클러스터 노드에서 TCP 포트를 엽니다.</p>	<p>자체 관리형 Couchbase 서버 클러스터 노드에서 XDCR 통신을 위해 적절한 포트가 열려 있는지 확인하십시오. 자세한 내용은 Couchbase 서버 포트 문서를 참조하세요.</p>	<p>카우치베이스 관리자, 시스템 관리자</p>
<p>카우치베이스 서버 엔터프라이즈 에디션을 사용하는 경우 카우치베이스 XDCR을 설정하세요.</p>	<ol style="list-style-type: none"> 1. Couchbase Capella 컨트롤 플레인 기본 탐색에서 클러스터를 선택한 다음 마이그레이션할 대상 클러스터를 선택합니다. 2. 루트 인증서에서 복사를 선택합니다. 3. 자체 관리형 Couchbase 서버 웹 콘솔에 로그인하고 기본 탐색에서 XDCR을 선택합니다. 그런 다음 원격 추가를 선택합니다. 	<p>카우치베이스 관리자</p>

작업	설명	필요한 기술
	<p>4. 다음 설정을 입력합니다.</p> <ul style="list-style-type: none"> • 클러스터 이름 - Capella 클러스터 연결의 이름 • IP/호스트 이름 — 카우치베이스 카펠라 클러스터의 연결 엔드포인트 • 원격 클러스터의 사용자 이름 - Couchbase Capella 클러스터의 데이터베이스 사용자입니다. • 비밀번호 - Couchbase Capella 클러스터의 데이터베이스 사용자 비밀번호입니다. • 보안 연결 활성화 - 선택됨 • 전체 (TLS 암호 및 데이터 암호화) - 선택 <p>5. 이전에 복사한 Capella 클러스터 루트 인증서를 붙여넣은 다음 저장을 선택합니다.</p>	

작업	설명	필요한 기술
<p>카우치베이스 XDCR을 시작합니다.</p>	<ol style="list-style-type: none"> 1. 자체 관리형 Couchbase 서버 웹 콘솔의 기본 탐색 메뉴에서 XDCR을 선택한 다음 복제 추가를 선택합니다. 2. 다음 설정을 입력합니다. <ul style="list-style-type: none"> • 버킷에서 복제 - 마이그레이션할 원본 버킷을 선택합니다. • 원격 버킷 - 대상 버킷 이름을 입력합니다. • 원격 클러스터 - 이전에 생성한 대상 클러스터를 선택합니다. 3. 복제 저장을 선택합니다. 복제 프로세스는 몇 초 내에 시작되어야 합니다. 	<p>카우치베이스 관리자</p>

옵션 1을 사용하여 인덱스를 마이그레이션하세요.

작업	설명	필요한 기술
<p>자체 관리형 클러스터 인덱스를 카우치베이스 카펠라로 마이그레이션하세요.</p>	<p>중요: 마이그레이션할 인덱스가 50개 미만인 경우 이 프로세스를 사용하는 것이 좋습니다. 마이그레이션할 인덱스가 50개를 초과하는 경우 마이그레이션 옵션 2를 사용하는 것이 좋습니다.</p> <ol style="list-style-type: none"> 1. Couchbase 웹 콘솔에서 인덱스를 선택합니다. 2. 인덱스 목록에서 마이그레이션할 첫 번째 인덱스를 선택합니다. 	<p>카우치베이스 관리자, 시스템 관리자</p>

작업	설명	필요한 기술
	<p>택합니다. 그러면 인덱스 정의가 표시됩니다.</p> <p>3. CREATE 명령문을 사용하여 인덱스 정의를 복사하되 WITH { "defer_build":true } 을(를) 복사하지는 마십시오.</p> <p>예를 들어, 다음 예제 인덱스 정의에서는 CREATE INDEX `cityindex` ON `travel-sample`(`city`) 만 복사할 수 있습니다.</p> <pre>CREATE INDEX `cityindex` ON `travel-sample`(`city`) WITH { "defer_build":true }</pre> <p>4. Couchbase Capella 컨트롤 플레인에서 클러스터를 선택한 다음 대상 클러스터를 선택합니다.</p> <p>5. 도구 드롭다운 목록에서 쿼리 워크벤치를 선택합니다. 이전에 복사한 CREATE 명령문을 쿼리 편집기에 붙여넣은 다음 실행을 선택합니다. 그러면 인덱스가 생성되고 작성됩니다.</p> <p>6. 인덱스가 생성되었는지 확인하려면 도구 드롭다운 목록에서 인덱스를 선택합니다. 목록에는 인덱스가 생성</p>	

작업	설명	필요한 기술
	<p>되고 작성되었음을 알 수 있습니다.</p> <p>7. 마이그레이션할 각 인덱스에 대해 이 프로세스를 반복합니다.</p>	

옵션 2을 사용하여 인덱스를 마이그레이션하세요.

작업	설명	필요한 기술
<p>인덱스 정의를 마이그레이션하십시오.</p>	<p>중요: 마이그레이션할 인덱스가 50개 이상인 경우 이 프로세스를 사용하는 것이 좋습니다. 마이그레이션할 인덱스가 50개 미만인 경우 마이그레이션 옵션 1을 사용하는 것이 좋습니다.</p> <ol style="list-style-type: none"> 1. Couchbase Shell을 설치한 시스템에서 명령줄 터미널을 엽니다. 2. 카우치베이스 셸을 시작하려면 다음 명령을 실행합니다. <div data-bbox="630 1423 1029 1507" style="border: 1px solid #ccc; border-radius: 10px; padding: 5px; margin: 10px 0;"> <pre>./cbsh</pre> </div> 3. 자체관리형 Couchbase Server 클러스터에 연결하려면 다음 명령을 실행합니다. <div data-bbox="630 1688 1029 1806" style="border: 1px solid #ccc; border-radius: 10px; padding: 5px; margin: 10px 0;"> <pre>cb-env cluster On-Prem-Cluster</pre> </div> 	<p>카우치베이스 관리자, 시스템 관리자</p>

작업	설명	필요한 기술
	<p>4. 자체 관리형 Couchbase 서버 클러스터에서 Couchbase Capella 클러스터로 인덱스 정의를 마이그레이션하려면 마이그레이션하려는 각 버킷에 대해 다음 명령을 실행합니다. <BUCKET_NAME> 을(를) 마이그레이션하려는 인덱스에 해당하는 버킷 이름으로 바꿔야 합니다. 이 마이그레이션 옵션을 사용하려면 대상 버킷 이름이 원본 버킷 이름과 동일해야 합니다.</p> <pre data-bbox="630 903 1029 1222">query indexes -- definitions where bucket =~ <BUCKET_N AME> get definitio n each { it query \$it --clusters Capella-Cluster }</pre>	

작업	설명	필요한 기술
인덱스 정의를 빌드하세요.	<p>1. 컨텍스트를 Couchbase Capella 클러스터로 전환하려면 다음 명령을 실행합니다.</p> <pre data-bbox="630 443 1029 562">cb-env cluster Capella-Cluster</pre> <p>2. Couchbase Capella 클러스터로 마이그레이션된 인덱스 정의를 빌드하려면 <BUCKET_NAME> 을(를) 빌드하려는 인덱스에 해당하는 버킷 이름으로 바꾸고 다음 명령을 실행합니다.</p> <pre data-bbox="630 936 1029 1862">query 'SELECT RAW CONCAT("BUILD INDEX ON ", k , "(['", CONCAT2 ("','", inames), "'']);") FROM system:indexes AS s LET bid = CONCAT("`",s.bucket_id, "`"), sid = CONCAT("`", s.scope_id, "`"), kid = CONCAT("`", s.keyspace_id, "`"), k = NVL2(bid, CONCAT2(".", bid, sid, kid), kid) WHERE s.namespa ce_id = "default" AND s.bucket_id = "" GROUP BY k LETTING inames = ARRAY_AGG (s.name) FILTER (WHERE s.state = 'deferred') HAVING</pre>	카우치베이스 관리자, 시스템 관리자

작업	설명	필요한 기술
	<pre>ARRAY_LENGTH(inames) > 0;' each { it query \$it }</pre> <p>3. 각 버킷에 대해 반복합니다.</p>	

전체 텍스트 검색 인덱스 마이그레이션

작업	설명	필요한 기술
<p>자체 관리형 클러스터 전체 텍스트 검색 인덱스를 Couchbase Capella로 마이그레이션하십시오.</p>	<ol style="list-style-type: none"> 1. Couchbase 웹 콘솔에서 검색을 선택합니다. 2. 전체 텍스트 검색 (FTS) 인덱스 목록에서 마이그레이션하려는 첫 번째 FTS 인덱스를 선택하고 인덱스 정의 JSON 표시를 선택한 다음 클립보드로 복사를 선택합니다. 인덱스 이름과 해당 인덱스가 속한 버킷을 기록해 둡니다. 3. Couchbase Capella 컨트롤 플레인에서 클러스터를 선택한 다음 대상 클러스터를 선택합니다. 4. 도구 드롭다운 목록에서 전체 텍스트 검색을 선택합니다. 5. 인덱스 가져오기를 선택하고 FTS 인덱스 정의를 붙여 넣습니다. 6. 인덱스 이름을 입력하고 자체 관리형 클러스터에 명시 	<p>카우치베이스 관리자</p>

작업	설명	필요한 기술
	<p>된 대로 올바른 버킷을 선택한 다음 생성을 선택합니다.</p> <p>7. 마이그레이션할 각 FTS 인덱스에 대해 이 프로세스를 반복합니다.</p>	

카우치베이스 커뮤니티 에디션에서 데이터 이전

작업	설명	필요한 기술
<p>자체 관리형 Couchbase 서버 커뮤니티 에디션에서 데이터를 내보냅니다.</p>	<p>카우치베이스 커뮤니티 에디션에서는 암호화된 XDCR을 사용할 수 없습니다. 카우치베이스 커뮤니티 에디션에서 데이터를 내보낸 다음 카우치베이스 카펠라로 데이터를 수동으로 가져올 수 있습니다.</p> <p>원본 버킷에서 데이터를 내보내려면 명령줄에서 <code>cbexport</code>을(를) 사용하십시오.</p> <p>다음 명령이 예시로 제공됩니다.</p> <pre>cbexport json \ --cluster localhost \ --bucket <SOURCE BUCKET NAME> \ --format lines \ --username <USERNAME> \ --password <PASSWORD> \ --include-key cbkey \ --scope-field cbscope \</pre>	<p>카우치베이스 관리자</p>

작업	설명	필요한 기술
	<pre data-bbox="597 212 1026 386">--collection-field cbcoll \ --output cbexporte d_data.json</pre> <p data-bbox="597 422 1026 745">참고로 cbkey, cbscope, cbcoll, 및 cbexporte d_data.json 는 임의의 레이블입니다. 프로세스 후반부에 참조될 것이므로 이름을 다르게 지정하려면 이 내용을 메모해 두십시오.</p>	

작업	설명	필요한 기술
<p>카우치베이스 카펠라로 데이터를 가져옵니다.</p>	<ol style="list-style-type: none"> 1. Couchbase Capella 컨트롤 플레인에서 클러스터를 선택한 다음 대상 클러스터를 선택합니다. 2. 도구 드롭다운 목록에서 가져오기를 선택합니다. 그러면 다음 6단계가 포함된 마법사가 열립니다. <ol style="list-style-type: none"> a. 버킷 — 대상 버킷을 선택합니다. b. 파일 — JSON을 선택하고 라인을 선택한 다음 웹 브라우저 사용을 선택합니다. 데이터가 많은 경우 수동 옵션을 탐색할 수 있습니다. <code>cbexport</code>이(가) 만든 파일을 선택합니다. c. 컬렉션 — 사용자 지정 컬렉션 매핑을 선택합니다. <p>Community Edition 데이터베이스에서 범위나 컬렉션을 사용하지 않거나 <code>_default</code>만 사용하는 경우 단일 컬렉션 선택 옵션을 대신 선택할 수 있습니다.</p> <p>컬렉션 매핑 표현식의 경우 <code>%cbscope%</code>. <code>%cbcoll%</code> 을(를) 입력합니다. 이 표현식이 제대로 작동하는지 확인하기 위해 다음과 같은 예제 데</p> 	<p>카우치베이스 관리자</p>

작업	설명	필요한 기술
	<p>이터를 붙여넣을 수 있습니다.</p> <pre data-bbox="667 331 1029 569"> { "cbscope" :"inventory", "cbcoll":"landmark ", "cbkey":" landmark_3991" } </pre> <p>d. 키 — 고객 세대를 선택하세요. (가져오는 데이터의 키를 보존하는 데 신경 쓰지 않는 경우 자동으로 생성된 UUID를 대신 선택하고 5단계로 진행하면 됩니다.) 키 이름 생성기 표현식의 경우 %cbkey%를 입력합니다. 이 표현식이 제대로 작동하는지 확인하려면 몇 가지 예제 데이터를 붙여넣으십시오.</p> <p>e. 구성 — 필드 무시를 선택하고 cbscope, cbcoll, cbkey를 입력합니다. 이 필드에는 임포트 후 대상 버킷에 있지 않아도 되는 임시 정보가 포함됩니다. 기타 설정은 기본값을 유지합니다.</p> <p>f. 가져오기 - 검토하고 준비가 되면 가져오기를 선택합니다. 업로드 및 데이터 가져오기가 완료될 때까지 기다리세요.</p>	

작업	설명	필요한 기술
	대용량 파일의 경우, 카우치베이스 카펠라는 cURL을 사용한 커맨드 라인 임포트를 지원합니다. Couchbase Capella 설명서의 데이터 가져오기 에서 가져오기 옵션을 더 자세히 살펴볼 수 있습니다.	

마이그레이션 테스트 및 확인

작업	설명	필요한 기술
데이터 마이그레이션 확인	<ol style="list-style-type: none"> 1. Couchbase Capella 컨트롤 플레인에서 클러스터를 선택한 다음 클러스터 목록에서 대상 클러스터를 선택합니다. 2. 대상 클러스터의 버킷 탭을 선택합니다. 대상 버킷의 항목 (문서) 수가 원본 버킷의 항목 수와 일치하는지 확인하십시오. 3. 대상 클러스터의 도구 드롭다운 목록에서 문서를 선택합니다. 모든 문서가 마이그레이션되었는지 확인합니다. 4. (선택 사항) 모든 데이터를 마이그레이션한 후 데이터를 삭제하여 복제를 종료할 수 있습니다. 자세한 내용은 Couchbase 설명서의 복제 삭제를 참조하세요. 	카우치베이스 관리자

작업	설명	필요한 기술
인덱스 마이그레이션을 확인하세요.	Couchbase Capella 컨트롤 플레인의 대상 클러스터의 도구 드롭다운 목록에서 인덱스를 선택합니다. 인덱스가 마이그레이션되고 빌드되었는지 확인하세요.	카우치베이스 관리자
쿼리 결과 확인	<ol style="list-style-type: none"> 1. Couchbase Capella 컨트롤 플레인의 대상 클러스터의 도구 드롭다운 목록에서 쿼리 워크벤치를 선택합니다. 2. 샘플 N1QL 쿼리 또는 애플리케이션에서 사용되는 쿼리를 실행합니다. 자체 관리형 Couchbase Server 클러스터에서 쿼리를 실행할 때와 동일한 결과를 받는지 확인하세요. 	카우치베이스 관리자
전체 텍스트 검색 결과를 확인합니다 (FTS 인덱스를 마이그레이션한 경우 해당).	<ol style="list-style-type: none"> 1. Couchbase Capella 컨트롤 플레인의 대상 클러스터의 도구 드롭다운 목록에서 전체 텍스트 검색을 선택합니다. 2. 이름을 선택하여 FTS 인덱스를 선택합니다. 3. 검색을 선택합니다. 4. 샘플 검색 쿼리를 입력하고 검색을 선택합니다. 5. 결과가 자체 관리형 클러스터에서 검색을 실행할 때와 동일한지 확인합니다. 	카우치베이스 관리자

관련 리소스

마이그레이션 준비

- [카우치베이스 카펠라 무료 평가판으로 시작해 보세요.](#)
- [카우치베이스 카펠라의 클라우드 제공자 요구 사항](#)
- [카우치베이스 카펠라 사이징 가이드라인](#)

데이터 및 인덱스 마이그레이션

- [카우치베이스 XDCR](#)
- [카우치베이스 셸 문서](#)

카우치베이스 카펠라 SLA 및 지원

- [카우치베이스 카펠라 서비스 수준에 관한 계약\(SLA\)](#)
- [카우치베이스 카펠라 서비스 지원 정책](#)

추가 정보

다음 코드는 [Couchbase 셸의 구성 파일 예제](#)입니다.

```
Version = 1

[[clusters]]
identifier = "On-Prem-Cluster"
hostnames = ["<SELF_MANAGED_COUCHBASE_CLUSTER>"]
default-bucket = "travel-sample"
username = "<SELF_MANAGED_ADMIN>"
password = "<SELF_MANAGED_ADMIN_PWD>"
tls-cert-path = "/<ABSOLUTE_PATH_TO_SELF_MANAGED_ROOT_CERT>"
data-timeout = "2500ms"
connect-timeout = "7500ms"
query-timeout = "75s"

[[clusters]]
identifier = "Capella-Cluster"
hostnames = ["<COUCHBASE_CAPELLA_ENDPOINT>"]
default-bucket = "travel-sample"
```

```
username = "<CAPELLA_DATABASE_USER>"
password = "<CAPELLA_DATABASE_USER_PWD>"
tls-cert-path = "/<ABSOLUTE_PATH_TO_COUCHBASE_CAPELLA_ROOT_CERT>"
data-timeout = "2500ms"
connect-timeout = "7500ms"
query-timeout = "75s"
```

구성 파일을 저장하기 전에 다음 표를 참조하여 원본 및 대상 클러스터 정보를 추가했는지 확인하십시오.

<SELF_MANAGED_COUCHBASE_CLUSTER>	자체 관리형 카우치베이스 서버 클러스터의 IP 주소를 사용하세요.
<SELF_MANAGED_ADMIN>	자체 관리형 Couchbase 서버 클러스터에는 관리자 사용자를 사용하십시오.
<ABSOLUTE_PATH_TO_SELF_MANAGED_ROOT_CERT>	자체 관리형 Couchbase 서버 클러스터의 저장된 루트 인증서 파일의 절대 경로를 사용하십시오.
<COUCHBASE_CAPELLA_ENDPOINT>	카우치베이스 카펠라 클러스터의 연결 엔드포인트를 사용하세요.
<CAPELLA_DATABASE_USER>	Couchbase Capella 클러스터에는 데이터베이스 사용자를 사용하십시오.
<CAPELLA_DATABASE_USER_PWD>	Couchbase Capella 클러스터의 데이터베이스 사용자 비밀번호를 사용하십시오.
<ABSOLUTE_PATH_TO_COUCHBASE_CAPELLA_ROOT_CERT>	Couchbase Capella 클러스터의 저장된 루트 인증서 파일의 절대 경로를 사용하십시오.

IBM WebSphere 애플리케이션 서버에서 Amazon EC2의 아파치 톰캣으로 마이그레이션

작성자: Neal Ardeljan(AWS) 및 Afroz Khan(AWS)

환경: 프로덕션	소스: 애플리케이션	대상: Amazon EC2 인스턴스의 Apache Tomcat
R 유형: 리플랫폼	워크로드: IBM, 오픈 소스	기술: 마이그레이션, 웹 및 모바일 앱
AWS 서비스: Amazon EC2		

요약

이 패턴은 IBM WebSphere 애플리케이션 서버 (WAS) 를 실행하는 온프레미스 레드햇 엔터프라이즈 리눅스 (RHEL) 6.9 이상 시스템에서 아마존 Elastic Compute Cloud (Amazon EC2) 인스턴스에서 아파치 톰캣을 실행하는 RHEL 8로 마이그레이션하는 단계를 안내합니다.

이 패턴은 다음의 소스 및 대상 버전에 적용될 수 있습니다.

- WebSphere 애플리케이션 서버 7.x에서 아파치 톰캣 8로 (자바 7 이상)
- WebSphere 애플리케이션 서버 8.x에서 아파치 톰캣 8까지 (Java 7 이상)
- WebSphere 애플리케이션 서버 8.5.5.x에서 아파치 톰캣 9까지 (자바 8 이상)
- WebSphere 애플리케이션 서버 8.5.5.x에서 아파치 톰캣 10으로 (자바 8 이상)

사전 조건 및 제한 사항

필수 조건

- 활성 상태의 AWS 계정
- 소스 Java 코드(다음과 같이 가정)
 - Java 7 이상의 Java Depoment Kit(JDK) 버전을 사용합니다.
 - Spring 또는 Apache Struts 프레임워크를 사용합니다.
 - 엔터프라이즈 자바 빈즈 (EJB) 프레임워크 또는 Tomcat에서 쉽게 사용할 수 없는 기타 WebSphere 서버 기능을 사용하지 않습니다.

- 주로 서블릿이나 Java Server Page(JSP)를 사용합니다.
- Java Database Connectivity(JDBC) 커넥터를 사용하여 데이터베이스에 연결합니다.
- 소스: IBM WebSphere 애플리케이션 서버 버전 7.x 이상
- 대상: Apache Tomcat 버전 8.5 이상

아키텍처

소스 기술 스택

- Apache Struts Model-View-Controller(MVC) 프레임워크를 사용하여 구축된 웹 애플리케이션
- IBM 애플리케이션 서버 버전 7.x 또는 8.x에서 실행되는 웹 WebSphere 애플리케이션
- Lightweight Directory Access Protocol(LDAP) 커넥터를 사용하여 LDAP 디렉터리(iPlanet/eTrust)에 연결하는 웹 애플리케이션
- IBM Tivoli Access Manager(TAM) 연결을 사용하여 TAM 사용자 비밀번호를 업데이트하는 애플리케이션(현재 구현에서는 애플리케이션이 PD.jar 사용)

온프레미스 데이터베이스

- Oracle Database 21c(21.0.0.0)
- Oracle Database 19c(19.0.0.0)
- Oracle Database 12c 릴리스 2(12.2.0.1)
- Oracle Database 12c 릴리스 1(12.1.0.2)

대상 기술 스택

- EC2 인스턴스의 RHEL에서 실행되는 Apache Tomcat 버전 8(이상)
- Amazon Relational Database Service(RDS) for Oracle

Amazon RDS에서 지원하는 Oracle 버전에 대한 자세한 내용은 [Amazon RDS for Oracle](#) 웹 사이트를 참조하세요.

대상 아키텍처

도구

- 애플리케이션 티어: Java 애플리케이션을 WAR 파일로 재구축.
- 데이터베이스 티어: Oracle 기본 백업 및 복원.
- 자카르타 EE용 Apache Tomcat 마이그레이션 도구. 이 도구는 Apache Tomcat 9에서 실행되는 Java EE 8용으로 작성된 웹 애플리케이션을 가져와 자카르타 EE 9를 구현하는 Apache Tomcat 10에서 실행되도록 자동 변환합니다.

에픽

마이그레이션 계획

작업	설명	필요한 기술
애플리케이션 검색, 현재 상태 정보 및 성능 기준을 완료합니다.		BA, 마이그레이션 책임자
소스 및 대상 데이터베이스 버전을 검증합니다.		DBA
대상 서버 EC2 인스턴스의 하드웨어 요구 사항을 식별합니다.		DBA, SysAdmin
스토리지 요구 사항(스토리지 유형 및 용량)을 식별합니다.		DBA, SysAdmin
용량, 스토리지 기능, 네트워크 기능에 따라 적절한 EC2 인스턴스 유형을 선택합니다.		DBA, SysAdmin
소스 및 대상 데이터베이스의 네트워크 액세스 보안 요구 사항을 식별합니다.		DBA, SysAdmin
애플리케이션 마이그레이션 또는 툴링 식별합니다.		DBA, 마이그레이션 책임자

작업	설명	필요한 기술
애플리케이션에 대한 마이그레이션 설계 및 마이그레이션 가이드를 작성합니다.		빌드 책임자, 마이그레이션 책임자
애플리케이션 마이그레이션 런북을 완성합니다.		빌드 책임자, 전환 리드, 테스트 책임자, 마이그레이션 책임자

인프라 구성

작업	설명	필요한 기술
Virtual Private Cloud(VPC)를 생성합니다.		SysAdmin
보안 그룹을 생성합니다.		SysAdmin
Amazon RDS for Oracle을 구성하고 시작합니다.		DBA, SysAdmin

데이터 마이그레이션

작업	설명	필요한 기술
데이터베이스 백업 파일을 가져오기 위한 엔드포인트를 생성하거나 액세스 권한을 확보합니다.		DBA
기본 데이터베이스 엔진 또는 타사 도구를 사용하여 데이터베이스 개체 및 데이터를 마이그레이션합니다.	자세한 내용은 추가 정보 섹션의 "데이터베이스 개체 및 데이터 마이그레이션"을 참조하세요.	DBA

애플리케이션 마이그레이션

작업	설명	필요한 기술
마이그레이션을 위한 변경 요청(CR)을 제출합니다.		전환 리드
마이그레이션을 위한 CR 승인을 받습니다.		전환 리드
애플리케이션 마이그레이션 런북에 따른 애플리케이션 마이그레이션 전략을 따릅니다.	자세한 내용은 추가 정보 섹션의 "애플리케이션 계층 설정"을 참조하세요.	DBA, 마이그레이션 엔지니어, 애플리케이션 소유자
애플리케이션을 업그레이드합니다(필요한 경우).		DBA, 마이그레이션 엔지니어, 애플리케이션 소유자
기능, 비기능, 데이터 검증, SLA 및 성능 테스트를 완료합니다.		테스트 책임자, 앱 소유자, 앱 사용자

전환

작업	설명	필요한 기술
애플리케이션 소유자 또는 비즈니스 소유자로부터 사인오프를 받습니다.		전환 리드
애플리케이션 클라이언트를 새 인프라로 전환합니다.		DBA, 마이그레이션 엔지니어, 애플리케이션 소유자

프로젝트 닫기

작업	설명	필요한 기술
임시 AWS 리소스를 종료합니다.		DBA, 마이그레이션 엔지니어, SysAdmin
프로젝트 문서를 검토하고 검증하세요.		마이그레이션 책임자
마이그레이션 시간, 수동 작업과 자동 작업의 비율, 비용 절감과 같은 지표를 수집합니다.		마이그레이션 책임자
프로젝트를 마무리하고 피드백을 제공하세요.		마이그레이션 책임자, 앱 소유자

관련 리소스

참조

- [Apache Tomcat 10.0 설명서](#)
- [Apache Tomcat 9.0 설명서](#)
- [Apache Tomcat 8.0 설명서](#)
- [Apache Tomcat 8.0 설치 가이드](#)
- [Apache Tomcat JNDI 설명서](#)
- [Amazon RDS for Oracle 웹사이트](#)
- [Amazon RDS 요금](#)
- [Oracle 및 Amazon Web 서비스](#)
- [Amazon RDS의 Oracle](#)
- [Amazon RDS 다중 AZ 배포](#)

자습서 및 동영상

- [Amazon RDS 시작](#)

추가 정보

데이터베이스 객체 및 데이터 마이그레이션

예를 들어, 기본 Oracle 백업/복원 유틸리티를 사용하는 경우

1. 데이터베이스 백업 파일에 대한 Amazon Simple Storage Service(S3) 백업을 생성(선택 사항)합니다.
2. Oracle DB 데이터를 네트워크 공유 폴더에 백업합니다.
3. 마이그레이션 스테이징 서버에 로그인하여 네트워크 공유 폴더를 매핑합니다.
4. 네트워크 공유 폴더에서 S3 버킷으로 데이터를 복사합니다.
5. Amazon RDS 다중 AZ 배포를 요청합니다.
6. 온프레미스 데이터베이스 백업을 Amazon RDS for Oracle로 복원합니다.

애플리케이션 티어 설정

1. Apache Tomcat 웹 사이트에서 Tomcat 8(또는 9/10)을 설치합니다.
2. 애플리케이션 및 공유된 라이브러리를 WAR 파일로 패키징합니다.
3. Tomcat에 WAR 파일을 배포합니다.
4. 에서 Linux cat 누락된 공유 라이브러리의 시작 로그를 모니터링하십시오. WebSphere
5. WebSphere특정 배포 디스크립터 Linux cat 확장에 대한 시작 기록을 확인하세요.
6. 서버에서 누락된 종속 Java 라이브러리를 모두 수집하세요. WebSphere
7. WebSphere특정 배포 디스크립터 요소를 Tomcat과 호환되는 것으로 수정하십시오.
8. 종속 Java 라이브러리와 업데이트된 배포 설명자를 사용하여 WAR 파일을 다시 빌드합니다.
9. LDAP 구성, 데이터베이스 구성 및 테스트 연결을 업데이트(Apache Tomcat 설명서의 [영역 구성 방법](#) 및 [JNDI 데이터소스 사용 방법](#) 참조)합니다.
- 10.복원된 Amazon RDS for Oracle 데이터베이스에 대해 설치된 애플리케이션을 테스트합니다.
- 11.EC2 인스턴스에서 Linux용 Amazon Machine Image(AMI)를 생성합니다.
- 12.Application Load Balancer 및 오토 스케일링 그룹으로 완성된 아키텍처를 시작합니다.
- 13.Application Load Balancer를 가리키도록 URL을 업데이트합니다(WebSeal 접합 사용).
- 14.구성 관리 데이터베이스(CMDB)를 업데이트합니다.

Auto Scaling을 사용하여 IBM WebSphere 애플리케이션 서버에서 Amazon EC2의 아파치 톰캣으로 마이그레이션하세요

작성자: Kevin Yung(AWS)

R 유형: 리플랫폼	소스: 애플리케이션	대상: AutoScaling이 활성화된 Amazon EC2 인스턴스의 Apache Tomcat
작성자: AWS	환경: PoC 또는 파일럿	기술: 웹 및 모바일 앱, 마이그레이션
워크로드: 오픈 소스, IBM	AWS 서비스: Amazon EC2	

요약

이 패턴은 Amazon EC2 Auto Scaling이 활성화된 Amazon Elastic Compute Cloud (Amazon EC2) 인스턴스의 Java WebSphere 애플리케이션을 IBM 애플리케이션 서버에서 Apache Tomcat으로 마이그레이션하기 위한 지침을 제공합니다.

이 패턴을 사용하면 다음을 달성할 수 있습니다.

- IBM 라이선스 비용 절감
- 다중 AZ 배포를 사용하는 고가용성
- Amazon EC2 Auto Scaling으로 애플리케이션 복원력 향상

사전 조건 및 제한 사항

사전 조건

- Java 애플리케이션(버전 7).x 또는 8.x)는 LAMP 스택에서 개발되어야 합니다.
- 대상 상태는 Linux 호스트에서 Java 애플리케이션을 호스팅하는 것입니다. 이 패턴은 Red Hat Enterprise Linux(RHEL) 7 환경에서 성공적으로 구현되었습니다. 다른 Linux 배포판도 이 패턴을 따를 수 있지만 Apache Tomcat 배포판의 구성을 참조해야 합니다.
- Java 애플리케이션의 종속성을 이해해야 합니다.
- 변경하려면 Java 애플리케이션 소스 코드에 액세스할 수 있어야 합니다.

제한 및 변경 사항 리플랫폼

- 엔터프라이즈 아카이브(EAR) 구성 요소를 이해하고 모든 라이브러리가 웹 구성 요소 WAR 파일에 패키징되어 있는지 확인해야 합니다. [Apache Maven WAR 플러그인](#)을 구성하고 WAR 파일 아티팩트를 생성해야 합니다.
- Apache Tomcat 8을 사용할 때 `javax.servlet-api.jar` 및 응용 프로그램 패키지에 내장된 `javax.servlet.jar` 파일 간에 알려져 있는 충돌이 존재합니다. 이 문제를 해결하려면 응용 프로그램 패키지에서 `javax.servlet-api.jar`를 삭제합니다.
- [Apache Tomcat 구성](#)의 클래스 경로에 있는 WEB-INF/리소스를 구성해야 합니다. 기본적으로 JAR 라이브러리는 디렉토리에 로드되지 않습니다. 또는, `src/main/resources` 아래에 모든 리소스를 배포할 수도 있습니다.
- [Java 응용 프로그램 내에서 하드 코딩된 컨텍스트 루트가 있는지 확인하고 Apache Tomcat의 새 컨텍스트 루트를 업데이트하십시오.](#)
- JVM 런타임 옵션을 설정하려면 Apache Tomcat bin 폴더에 `setenv.sh` 구성 파일을 생성할 수 있습니다(예: `JAVA_OPTS`, `JAVA_HOME`, 등).
- 인증은 컨테이너 수준에서 구성되며 Apache Tomcat 구성에서 영역으로 설정됩니다. 인증은 다음 세 가지 영역 중 하나에 대해 설정됩니다.
 - [JDBC Database Realm](#)은 JDBC 드라이버에 의해 접근된 관계형 데이터베이스에서 사용자를 조회합니다.
 - [DataSource 데이터베이스 Realm](#)은 [JNDI에서](#) 액세스하는 데이터베이스에서 사용자를 조회합니다.
 - [JNDI Directory Realm](#)은 JNDI 제공자에 의해 접근된 LDAP(Lightweight Directory Access Protocol) 디렉토리에서 사용자를 조회합니다. 조회에는 다음이 필요합니다.
 - LDAP 연결 세부 정보: 사용자 검색 기준, 검색 필터, 역할 기반, 역할 필터
 - 키 JNDI Directory Realm: LDAP에 연결하고, 사용자를 인증하고, 사용자가 구성원인 모든 그룹을 검색합니다.
- 권한 부여: `web.xml` 권한 부여 제약 조건을 확인하는 역할 기반 권한 부여가 있는 컨테이너의 경우 웹 리소스를 정의하고 제약 조건에 정의된 역할과 비교해야 합니다. LDAP에 그룹 역할 매핑이 없는 경우, 그룹 역할 매핑을 구현하려면 `web.xml` 속성 `<security-role-ref>` 을 설정해야 합니다. 구성 문서의 예를 보려면 [Oracle 설명서](#)를 참조하세요.
- 데이터베이스 연결: Amazon Relational Database Service(RDS) 엔드포인트 URL 및 연결 세부 정보를 사용하여 Apache Tomcat에서 리소스 정의를 생성합니다. JNDI 조회를 사용하여 a를 참조하도록 애플리케이션 코드를 업데이트하십시오. DataSource 에 정의된 기존 DB 연결은 WebSphere 의 WebSphere JNDI 이름을 사용하므로 작동하지 않습니다. `<resource-ref>`JNDI 이름 및 DataSource

유형 정의가 포함된 항목을 web.xml 에 추가할 수 있습니다. 샘플 구성 문서를 보려면 [Apache Tomcat 설명서](#)를 참조하세요.

- 로깅: 기본적으로 Apache Tomcat은 콘솔이나 로그 파일에 기록합니다. logging.properties를 업데이트하여 영역 수준 추적을 활성화할 수 있습니다([Tomcat 내 로깅](#) 참조). Apache Log4j를 사용하여 파일에 로그를 추가하는 경우 tomcat-juli를 다운로드하여 클래스 경로에 추가해야 합니다.
- 세션 관리: 애플리케이션 로드 밸런싱 및 세션 관리를 위해 IBM WebSeal을 유지하는 경우에는 변경할 필요가 없습니다. [AWS 기반 Application Load Balancer 또는 Network Load Balancer를 사용하여 IBM WebSeal 구성 요소를 대체하는 경우, ElastiCache Amazon 인스턴스를 Memcached 클러스터와 함께 사용하여 세션 관리를 설정하고 오픈 소스 세션 관리를 사용하도록 Apache Tomcat을 설정해야 합니다.](#)
- IBM WebSEAL 전달 프록시를 사용하는 경우 AWS에 새 Network Load Balancer를 설정해야 합니다. WebSEAL 집합 구성에는 Network Load Balancer에서 제공하는 IP를 사용하세요.
- SSL 구성: 통신에는 보안 소켓 계층 (SSL) 을 사용하는 것이 좋습니다. end-to-end Apache Tomcat 에서 SSL 서버 구성을 설정하려면 [Apache Tomcat 설명서](#)의 지침을 따르세요.

아키텍처

소스 기술 스택

- IBM WebSphere 애플리케이션 서버

대상 기술 스택

- 이 아키텍처는 [Elastic Load Balancing\(버전 2\)](#)을 사용합니다. 식별 관리 및 로드 밸런싱을 위해 IBM WebSEAL을 사용하는 경우, AWS 기반 Network Load Balancer를 선택하여 IBM WebSEAL 역방향 프록시와 통합할 수 있습니다.
- Java 애플리케이션은 [Amazon EC2 Auto Scaling 그룹](#)의 EC2 인스턴스에서 실행되는 Apache Tomcat 애플리케이션 서버에 배포됩니다. CPU 사용률과 같은 Amazon CloudWatch 지표를 기반으로 [조정 정책을](#) 설정할 수 있습니다.
- 로드 밸런싱을 위한 IBM WebSeal 사용을 중단하려는 경우 [Amazon ElastiCache for Memcached](#)를 세션 관리에 사용할 수 있습니다.
- 백엔드 데이터베이스의 경우 Amazon RDS용 [High Availability \(Multi-AZ\) for Amazon RDS](#)를 배포하고 데이터베이스 엔진 유형을 선택할 수 있습니다.

대상 아키텍처

도구

- [AWS CloudFormation](#)
- [AWS Command Line Interface \(AWS CLI\)](#)
- Apache Tomcat(버전 7. x 또는 8.x)
- RHEL 7 또는 Centos 7
- [Amazon RDS 다중 AZ 배포](#)
- [ElastiCache Memcached용 Amazon \(선택 사항\)](#)

에픽

VPC 설정

작업	설명	필요한 기술
Virtual Private Cloud(VPC)를 생성합니다.		
서브넷 생성		
필요한 경우 라우팅 테이블을 생성하세요.		
네트워크 액세스 제어 목록 (ACL) 생성		
AWS Direct Connect 또는 기업 VPN 연결을 설정합니다.		

애플리케이션 리플랫폼

작업	설명	필요한 기술
애플리케이션 빌드 Maven 구성을 리팩토링하여 WAR 아티팩트를 생성합니다.		
Apache Tomcat에서 애플리케이션 종속성 데이터 소스를 리팩토링합니다.		
Apache Tomcat에서 JNDI 이름을 사용하도록 애플리케이션 소스 코드를 리팩토링합니다.		
WAR 아티팩트를 Apache Tomcat에 배포합니다.		
애플리케이션 검증 및 테스트를 완료하세요.		

네트워크 구성

작업	설명	필요한 기술
종속성 서비스에 연결할 수 있도록 회사 방화벽을 구성합니다.		
최종 사용자가 AWS의 Elastic Load Balancing에 액세스할 수 있도록 기업 방화벽을 구성합니다.		

애플리케이션 인프라 생성

작업	설명	필요한 기술
EC2 인스턴스에 애플리케이션을 생성하고 배포합니다.		
세션 관리를 위한 Amazon ElastiCache for Memcached 클러스터를 생성하십시오.		
백엔드 데이터베이스용 Amazon RDS 다중 AZ 인스턴스를 생성합니다.		
SNI 인증서를 생성하고 AWS Certificate Manager(ACM)에 가져옵니다.		
로드 밸런서에 SSL 인증서를 설치합니다.		
Apache Tomcat 서버용 SSL 인증서를 설치합니다.		
애플리케이션 검증 및 테스트를 완료하세요.		

전환

작업	설명	필요한 기술
기존 인프라를 종료합니다.		
프로덕션 환경에서 Amazon RDS로 데이터베이스를 복원합니다.		

작업	설명	필요한 기술
DNS를 변경하여 애플리케이션을 전환합니다.		

관련 리소스

참조

- [Apache Tomcat 7.0 설명서](#)
- [Apache Tomcat 7.0 설치 가이드](#)
- [Apache Tomcat JNDI 설명서](#)
- [Amazon RDS 다중 AZ 배포](#)
- [ElastiCache 메모캐시를 위한 Amazon](#)

자습서 및 동영상

- [Amazon RDS 시작](#)

Microsoft Azure 앱 서비스의 .NET 애플리케이션을 AWS Elastic Beanstalk로 마이그레이션합니다.

제작: Raghavender Madamshitti(AWS)

환경: PoC 또는 파일럿	소스: 애플리케이션	대상: AWS Elastic Beanstalk
R 유형: 리플랫폼	워크로드: Microsoft	기술: 마이그레이션, 웹 및 모바일 앱

요약

이 패턴은 Microsoft Azure 앱 서비스에 호스팅된 .NET 웹 애플리케이션을 AWS Elastic Beanstalk로 마이그레이션하는 방법을 설명합니다. 애플리케이션을 Elastic Beanstalk로 마이그레이션하는 방법에는 두 가지가 있습니다.

- Visual Studio용 AWS Toolkit 사용 - 이 Microsoft Visual Studio IDE용 플러그인은 사용자 지정 .NET 애플리케이션을 AWS에 배포하는 가장 쉽고 간단한 방법을 제공합니다. 이 접근 방식을 사용하면 .NET 코드를 AWS에 직접 배포하고 SQL 서버 데이터베이스용 Amazon Relational Database Service(RDS)와 같은 지원 리소스를 Visual Studio에서 직접 만들 수 있습니다.
- Elastic Beanstalk에 업로드 및 배포 - 각 Azure 앱 서비스에는 Kudu라는 백그라운드 서비스가 포함되어 있습니다. Kudu는 메모리 덤프 및 배포 로그를 캡처하고, 구성 파라미터를 보고, 배포 패키지에 액세스하는 데 유용합니다. Kudu 콘솔을 사용하여 Azure 앱 서비스 콘텐츠에 액세스하고, 배포 패키지를 추출한 다음, Elastic Beanstalk 콘솔의 업로드 및 배포 옵션을 사용하여 패키지를 Elastic Beanstalk에 업로드할 수 있습니다.

이 패턴은 두 번째 접근 방식(Kudu를 통해 Elastic Beanstalk에 애플리케이션을 업로드)을 설명합니다. 이 패턴은 또한 AWS Elastic Beanstalk, 아마존 가상 사설 클라우드 (Amazon VPC), 아마존, 아마존, 아마존 Elastic Compute Cloud (CloudWatch아마존 EC2) Auto Scaling, 아마존 심플 스토리지 서비스 (아마존 S3), 아마존 Route 53 등의 AWS 서비스를 사용합니다.

.NET 웹 애플리케이션은 Amazon EC2 Auto Scaling 그룹에서 실행되는 AWS Elastic Beanstalk에 배포됩니다. CPU 사용률과 같은 Amazon CloudWatch 지표를 기반으로 조정 정책을 설정할 수 있습니다. 데이터베이스의 경우 애플리케이션 및 비즈니스 요구 사항에 따라 다중 AZ 환경에서 Amazon RDS를 사용하거나 Amazon DynamoDB를 사용할 수 있습니다.

사전 조건 및 제한 사항

사전 조건

- 활성 상태의 AWS 계정.
- Azure 앱 서비스에서 실행되는 .NET 웹 애플리케이션
- Azure 앱 서비스 Kudu 콘솔을 사용할 수 있는 권한

제품 버전

- .NET Core(x64) 1.0.1, 2.0.0 이상 또는 .NET Framework 4.x, 3.5([Windows Server .NET 플랫폼 히스토리 참조](#))
- Windows Server 2012 이상에서 실행되는 IIS(인터넷 정보 서비스) 버전 8.0 이상
- .NET 2.0 또는 4.0 런타임

아키텍처

소스 기술 스택

- .NET 프레임워크 3.5 이상 또는 .NET Core 1.0.1, 2.0.0 이상을 사용하여 개발되고 Azure 앱 서비스에 호스팅된 애플리케이션(웹 앱 또는 API 앱)

대상 기술 스택

- Amazon EC2 Auto Scaling 그룹에서 실행되는 AWS Elastic Beanstalk

마이그레이션 아키텍처

배포 워크플로

도구

도구

- .NET 코어 또는 .NET 프레임워크
- C#
- IIS
- Kudu 콘솔

AWS 서비스 및 특성

- [AWS Elastic Beanstalk — Elastic Beanstalk](#)는 .NET 웹 애플리케이션을 배포하고 easy-to-use 확장하기 위한 서비스입니다. Elastic Beanstalk는 용량 프로비저닝, 로드 밸런싱 및 오토 스케일링을 자동으로 관리합니다.
- [Amazon EC2 Auto Scaling 그룹](#) - Elastic Beanstalk에는 환경에서 Amazon EC2 인스턴스를 관리하는 Auto Scaling 그룹이 포함됩니다. 단일 인스턴스 환경에서 Auto Scaling 그룹은 실행 중인 인스턴스가 항상 한 개가 있도록 보장합니다. 로드 밸런싱된 환경에서 사용자가 실행할 다양한 인스턴스로 이루어진 그룹을 구성하면 Amazon EC2 Auto Scaling에서는 로드를 기준으로 필요에 따라 인스턴스를 추가하거나 제거합니다.
- [Elastic Load Balancing](#) - AWS Elastic Beanstalk에서 로드 밸런싱을 활성화하면 환경의 EC2 인스턴스 간에 트래픽을 분산하는 로드 밸런서가 생성됩니다.
- [아마존 CloudWatch](#) — Elastic Beanstalk는 Amazon을 자동으로 CloudWatch 사용하여 애플리케이션 및 환경 리소스에 대한 정보를 제공합니다. CloudWatch Amazon은 표준 지표, 사용자 지정 지표 및 경보를 지원합니다.
- [Amazon Route 53](#) - Amazon Route 53은 가용성과 확장성이 뛰어난 도메인 이름 시스템(DNS) 웹 서비스입니다. Route 53 별칭 레코드를 사용하여 사용자 지정 도메인 이름을 AWS Elastic Beanstalk 환경에 매핑할 수 있습니다.

에픽

VPC를 설정

작업	설명	필요한 기술
Virtual Private Cloud(VPC).	AWS 계정에 필요한 정보가 포함된 VPC를 생성하세요.	시스템 관리자
서브넷을 생성합니다.	VPC에서 서브넷 2개 이상을 생성합니다.	시스템 관리자

작업	설명	필요한 기술
라우팅 테이블을 생성합니다.	요구 사항에 따라 라우팅 테이블을 생성합니다.	시스템 관리자

AWS Elastic Beanstalk 설정

작업	설명	필요한 기술
Azure 앱 서비스 Kudu 콘솔에 액세스하세요.	Azure 포털에서 앱 서비스 대시보드로 이동한 다음 고급 도구, 이동을 선택하여 Kudu에 액세스합니다. 또는 Azure 앱 서비스 URL을 다음과 같이 수정할 수 있습니다: <code>https://<appservicename>.scm.azurewebsites.net</code>	앱 개발자, 시스템 관리자
Kudu에서 배포 패키지를 다운로드하세요.	DebugConsole 옵션을 PowerShell 선택하여 Windows로 이동합니다. 그러면 Kudu 콘솔이 열립니다. <code>wwwroot</code> 폴더로 이동하여 다운로드하세요. 그러면 Azure 앱 서비스 배포 패키지가 zip 파일로 다운로드됩니다. 예제는 첨부 문서를 참조하세요.	앱 개발자, 시스템 관리자
Elastic Beanstalk를 위한 패키지를 만드세요.	Azure 앱 서비스에서 다운로드한 배포 패키지의 압축을 풉니다. <code>aws-windows-deployment-manifest.json</code> 이라는 JSON 파일을 생성합니다(이 파일은 .NET Core 애플리케이션에만 필요함). <code>aws-windows-deployment-</code>	앱 개발자, 시스템 관리자

작업	설명	필요한 기술
	manifest.json 및 Azure 앱 서비스 배포 패키지 파일을 포함하는 zip 파일을 생성합니다. 예제는 첨부 문서를 참조하세요.	
새 Elastic Beanstalk 애플리케이션을 생성하세요.	Elastic Beanstalk 콘솔을 엽니다. 기존 애플리케이션을 선택하거나 새 애플리케이션을 생성합니다.	앱 개발자, 시스템 관리자
환경을 생성합니다.	Elastic Beanstalk 콘솔 작업 메뉴에서 환경 생성을 선택합니다. 웹 서버 환경 및 .NET/IIS 플랫폼을 선택합니다. 애플리케이션 코드에서 코드 업로드를 선택합니다. Elastic Beanstalk용으로 준비한 zip 파일을 업로드한 다음 환경 생성을 선택합니다.	앱 개발자, 시스템 관리자
Amazon을 구성합니다 CloudWatch.	기본적으로 기본 CloudWatch 모니터링이 활성화됩니다. 구성을 변경하려면 Elastic Beanstalk 마법사에서 게시된 애플리케이션을 선택한 다음 모니터링을 선택합니다.	시스템 관리자
배포 패키지가 Amazon S3에 있는지 확인합니다.	애플리케이션 환경이 생성되면 S3 버킷에서 배포 패키지를 찾을 수 있습니다.	앱 개발자, 시스템 관리자
애플리케이션을 테스트합니다.	환경이 생성되면 Elastic Beanstalk 콘솔에서 제공된 URL을 사용하여 애플리케이션을 테스트합니다.	시스템 관리자

관련 리소스

- [AWS Elastic Beanstalk 개념](#)(Elastic Beanstalk 설명서)
- [Elastic Beanstalk에서 .NET 시작하기](#)(Elastic Beanstalk 설명서)
- [Kudu 콘솔 \(\)](#) GitHub
- [‘Kudu’를 사용한 Azure 웹 앱 관리](#)(GS Lab 기사)
- [사용자 지정 ASP.NET Core Elastic Beanstalk 배포](#)(Visual Studio용 AWS 툴킷 사용 설명서)
- [Elastic Load Balancing 설명서](#)
- [AWS Elastic Beanstalk 지원 플랫폼](#)(Elastic Beanstalk 설명서)
- [AWS에 웹 애플리케이션 배포](#)(C# 코너 기사)
- [오토 스케일링의 크기 조정](#)(Amazon EC2 설명서)
- [Amazon RDS를 위한 고가용성\(다중 AZ\)](#)(Amazon RDS 설명서)

추가 정보

참고

- 온프레미스 또는 Azure SQL Server 데이터베이스를 Amazon RDS로 마이그레이션하는 경우 데이터베이스 연결 세부 정보도 업데이트해야 합니다.
- 테스트 목적으로 샘플 데모 애플리케이션이 첨부되어 있습니다.

첨부

이 문서와 관련된 추가 콘텐츠에 액세스하려면 [attachment.zip](#) 파일의 압축을 풉니다.

자체 호스팅된 MongoDB 환경을 AWS 클라우드의 MongoDB Atlas로 마이그레이션

작성자: Suresh Veeragoni(AWS)

소스: MongoDB	대상: AWS 기반 MongoDB Atlas	R 유형: 리플랫폼
환경: 프로덕션	기술: 마이그레이션, 분석, 데이터베이스	워크로드: 기타 모든 워크로드
AWS 서비스: Amazon EC2, Amazon VPC		

요약

이 패턴은 자체 관리형 MongoDB 환경(MongoDB 커뮤니티 서버, 엔터프라이즈 서버, 엔터프라이즈 어드밴스드, mLab 또는 기타 관리형 MongoDB 클러스터 포함)에서 Amazon Web Services(AWS) 클라우드의 MongoDB Atlas로 마이그레이션하는 단계를 설명합니다. [Atlas Live Migration Service](#)를 사용하여 MongoDB에서 MongoDB Atlas로 데이터를 더욱 빠르게 마이그레이션합니다.

이 패턴은 AWS 권장 가이드 웹 사이트의 [MongoDB에서 AWS 클라우드의 MongoDB Atlas로 마이그레이션](#) 안내서와 함께 제공됩니다. 마이그레이션을 위한 구현 단계를 제공합니다.

이 패턴은 AWS Service Integrator Partners(SI Partners)와 AWS 사용자를 대상으로 합니다.

사전 조건 및 제한 사항

사전 조건

- MongoDB Atlas로 마이그레이션하기 위한 소스 MongoDB 환경

전문성

- 이 패턴을 사용하려면 MongoDB, MongoDB Atlas 및 AWS 서비스에 익숙해야 합니다. 자세한 내용은 AWS 권장 가이드 웹 사이트의 [MongoDB에서 AWS 클라우드 MongoDB Atlas로 마이그레이션](#) 안내서의 [역할 및 책임](#)을 참조하세요.

제품 버전

- MongoDB 버전 2.6 이상의 경우

아키텍처

다양한 사용 시나리오를 지원하는 MongoDB Atlas 참조 아키텍처에 대한 내용은 AWS 권장 가이드 웹 사이트의 MongoDB에서 AWS 클라우드 MongoDB Atlas로 마이그레이션 안내서에서 [AWS상의 MongoDB Atlas 참조 아키텍처](#)를 참조하세요.

도구

- [Atlas Live Migration Service](#) – 데이터베이스를 Atlas로 마이그레이션하는 데 도움이 되는 무료 MongoDB 유틸리티입니다. 이 서비스는 전환이 완료될 때까지 소스 데이터베이스를 대상 데이터베이스와 동기화된 상태로 유지합니다. 전환할 준비가 되면 애플리케이션 인스턴스를 중지하고 대상 Atlas 클러스터를 가리킨 다음 다시 시작합니다.

에픽

검색 및 평가

작업	설명	필요한 기술
클러스터 크기를 결정합니다.	전체 인덱스 공간에 대한 db.stats()의 정보를 사용하여 작업 집합 크기를 추정합니다. 데이터 공간 중 일정 비율에 자주 액세스한다고 가정합니다. 또는 자체 가정을 기반으로 메모리 요구량을 추정할 수도 있습니다. 이 작업은 약 1주일 정도 소요됩니다. 이 스토리와 이 에픽에 실린 다른 스토리에 대한 자세한 내용과 예제를 보려면 '관련 리소스' 섹션의 링크를 참조하세요.	MongoDB DBA, 애플리케이션 아키텍트

작업	설명	필요한 기술
네트워크 대역폭 요구 사항을 추정합니다.	네트워크 대역폭 요구 사항을 추정하려면 평균 문서 크기에 초당 제공되는 문서 수를 곱하세요. 클러스터의 모든 노드가 부담하는 최대 트래픽을 기준으로 고려하세요. 클러스터에서 클라이언트 애플리케이션으로의 다운스트림 데이터 전송 속도를 계산하려면 일정 기간 동안 반환된 총 문서의 합계를 사용하세요. 애플리케이션이 보조 노드에서 읽는 경우, 이 전체 문서 수를 읽기 작업을 수행할 수 있는 노드 수로 나누세요. 데이터베이스의 평균 문서 크기를 찾으려면 <code>db.stats()</code> 를 사용하십시오. <code>avgObjSize</code> 명령. 이 작업은 일반적으로 하루가 소요됩니다.	MongoDB DBA
Atlas 티어를 선택합니다.	MongoDB 설명서의 지침에 따라 올바른 Atlas 클러스터 티어를 선택하세요.	MongoDB DBA
애플리케이션 전환을 계획합니다.		MongoDB DBA, 애플리케이션 아키텍트

AWS에 새로운 MongoDB Atlas 환경 설정

작업	설명	필요한 기술
AWS에 새 MongoDB Atlas 클러스터를 생성하세요.	MongoDB Atlas에서 '클러스터 구축'을 선택하여 '새 클러스터 생성' 대화 상자를 표시합니다.	MongoDB DBA

작업	설명	필요한 기술
	AWS를 클라우드 공급자로 선택합니다.	
리전 및 글로벌 클러스터 구성을 선택합니다.	Atlas 클러스터에 사용 가능한 AWS 리전 목록에서 선택하세요. 필요한 경우 글로벌 클러스터를 구성하세요.	MongoDB DBA
클러스터 티어를 선택합니다.	선호하는 클러스터 티어를 선택합니다. 티어 선택에 따라 메모리, 스토리지, IOPS 사양과 같은 요소가 결정됩니다.	MongoDB DBA
추가 클러스터 설정을 구성합니다.	MongoDB 버전, 백업 및 암호화 옵션과 같은 추가 클러스터 설정을 구성합니다. 이러한 옵션에 대한 자세한 내용은 '관련 리소스' 섹션의 링크를 참조하세요.	MongoDB DBA

보안 및 규정 준수를 구성합니다.

작업	설명	필요한 기술
액세스 목록을 구성합니다.	Atlas 클러스터에 연결하려면 프로젝트의 액세스 목록에 항목을 추가해야 합니다. Atlas는 전송 계층 보안(TLS)/Secure Sockets Layer(SSL)를 사용하여 데이터베이스의 Virtual Private Cloud(VPC) 연결을 암호화합니다. 프로젝트에 대한 액세스 목록을 설정하고 이 에픽의 스토리에 대한 자세한 내	MongoDB DBA

작업	설명	필요한 기술
	용을 확인하려면 ‘관련 리소스’ 섹션의 링크를 참조하세요.	
사용자를 인증하고 권한을 부여합니다.	MongoDB Atlas 클러스터에 액세스할 데이터베이스 사용자를 생성하고 인증해야 합니다. 프로젝트의 클러스터에 액세스하려면 사용자는 해당 프로젝트에 속해야 하며 여러 프로젝트에 속할 수 있습니다.	MongoDB DBA
사용자 지정 역할을 생성합니다.	(선택 사항)Atlas는 내장된 Atlas 데이터베이스 사용자 권한이 원하는 권한 세트를 지원하지 않는 경우 활용할 수 있는 사용자 지정 역할 생성을 지원합니다.	MongoDB DBA
VPC 피어링을 설정합니다.	(선택 사항)Atlas는 다른 AWS, Azure 또는 구글 클라우드 플랫폼(GCP) VPC와의 VPC 피어링을 지원합니다.	MongoDB DBA
AWS PrivateLink 엔드포인트를 설정합니다.	(선택 사항) AWS를 사용하여 PrivateLink AWS에 프라이빗 엔드포인트를 설정할 수 있습니다.	MongoDB DBA
2단계 인증을 활성화합니다.	(선택 사항)Atlas는 사용자가 Atlas 계정에 대한 액세스를 제어할 수 있도록 2단계 인증(2FA)을 지원합니다.	MongoDB DBA

작업	설명	필요한 기술
LDAP을 사용하여 사용자 인증 및 권한 부여를 설정합니다.	(선택 사항)Atlas는 Lightweight Directory Access Protocol(LDAP)을 통한 사용자 인증 및 권한 부여를 지원합니다.	MongoDB DBA
통합 AWS 액세스를 설정합니다.	(선택 사항)Atlas Data Lake 및 고객 키 관리를 사용한 저장 중 암호화를 비롯한 일부 Atlas 기능은 AWS Identity and Access Management(AWS IAM) 역할을 인증에 사용합니다.	MongoDB DBA
AWS KMS를 사용하여 저장 중 암호화를 설정합니다.	(선택 사항)Atlas는 AWS Key Management System(AWS KMS)을 사용하여 스토리지 엔진 및 클라우드 공급자 백업을 암호화할 수 있도록 지원합니다.	MongoDB DBA
클라이언트 측 필드 수준 암호화를 설정합니다.	(선택 사항)Atlas는 필드 자동 암호화를 비롯한 클라이언트 측 필드 수준 암호화를 지원합니다.	MongoDB DBA

데이터 마이그레이션

작업	설명	필요한 기술
MongoDB Atlas에서 대상 복제본 세트를 시작합니다.	MongoDB Atlas에서 대상 복제본 세트를 시작합니다. Atlas Live Migration Service에서 '마이그레이션 준비 완료'를 선택합니다.	MongoDB DBA

작업	설명	필요한 기술
Atlas Live Migration Service를 AWS 소스 클러스터의 액세스 목록에 추가합니다.	이렇게 하면 소스 환경을 준비하여 대상 Atlas 클러스터에 연결할 수 있습니다.	MongoDB DBA
Atlas Live Migration Service를 통해 AWS 보안 인증 정보를 검증하세요.	‘마이그레이션 시작’을 선택합니다. ‘전환 준비’ 버튼이 녹색으로 바뀌면 전환을 수행하세요. Atlas 클러스터 성능 지표를 검토하세요.	MongoDB DBA

운영 통합 구성

작업	설명	필요한 기술
MongoDB Atlas 클러스터에 연결합니다.		애플리케이션 개발자
클러스터 데이터와 상호 작용합니다.		애플리케이션 개발자
클러스터를 모니터링합니다.		MongoDB DBA
클러스터 데이터를 백업하고 복원합니다.		MongoDB DBA

관련 리소스

마이그레이션 가이드

- [MongoDB에서 AWS 클라우드의 MongoDB Atlas로 마이그레이션](#)

검색 및 평가

- [메모리](#)
- [Atlas 샘플 데이터 세트를 사용한 크기 조정 예제](#)

- [모바일 애플리케이션의 크기 조정 예제](#)
- [네트워크 트래픽](#)
- [클러스터 Auto Scaling](#)
- [Atlas 크기 조정 템플릿](#)

보안 및 규정 준수 구성

- [IP 액세스 목록 항목 구성](#)
- [데이터베이스 사용자 구성](#)
- [Atlas 사용자 액세스](#)
- [사용자 지정 역할 구성](#)
- [기본 데이터베이스 사용자 권한](#)
- [네트워크 피어링 연결 설정](#)
- [프라이빗 엔드포인트 설정](#)
- [멀티 팩터 인증](#)
- [LDAP을 통한 사용자 인증 및 권한 부여 설정](#)
- [Atlas 데이터 레이크](#)
- [고객 키 관리를 사용한 저장 중 암호화](#)
- [IAM 역할 사용](#)
- [클라이언트측 필드 수준 암호화](#)
- [자동 클라이언트측 필드 레벨 암호화](#)
- [MongoDB Atlas 보안](#)
- [MongoDB 신뢰 센터](#)
- [보안 기능 및 설정](#)

AWS에 새로운 MongoDB Atlas 환경 설정

- [클라우드 공급자 및 리전](#)
- [글로벌 클러스터](#)
- [클러스터 티어](#)
- [추가 클러스터 설정](#)
- [Atlas로 시작](#)

- [Atlas 사용자 액세스](#)
- [클러스터](#)

데이터 마이그레이션

- [클러스터 모니터링](#)

운영 통합

- [클러스터에 연결](#)
- [Atlas에서 CRUD 작업 수행](#)
- [클러스터 모니터링](#)
- [클러스터 데이터 백업 및 복원](#)

오라클에서 Amazon WebLogic ECS의 아파치 톰캣 (TomEE) 으로 마이그레이션

작성자: Anya Epishcheva(AWS)

R 유형: 리플랫폼	소스: 컨테이너	대상: Amazon ECS의 Apache Tomcat(TomEE)
작성자: AWS	환경: PoC 또는 파일럿	기술: 컨테이너 및 마이크로서비스, 마이그레이션
워크로드: Oracle	AWS 서비스: Amazon ECS	

요약

이 패턴은 오라클을 실행하는 온프레미스 Oracle Solaris SPARC 시스템을 Amazon Elastic Container Service (Amazon ECS) 를 WebLogic 사용하여 [Apache Tomee \(컨테이너 지원이 추가된 Apache Tomcat\)](#) 를 실행하는 Docker 컨테이너 기반 설치로 마이그레이션하는 단계를 설명합니다.

Oracle에서 Tomcat으로 마이그레이션하는 애플리케이션과 관련된 데이터베이스를 마이그레이션하는 방법에 대한 자세한 내용은 이 카탈로그의 데이터베이스 마이그레이션 패턴을 참조하십시오.
WebLogic

모범 사례

Java 및 Java Enterprise Edition(Java EE) 웹 애플리케이션을 마이그레이션하는 단계는 애플리케이션에서 사용하는 컨테이너별 리소스 수에 따라 달라집니다. Spring 기반 애플리케이션은 배포 컨테이너에 대한 종속성 수가 적기 때문에 일반적으로 마이그레이션하기가 더 쉽습니다. 반면 엔터프라이즈 JavaBeans (EJB) 및 스레드 풀, Java 인증 및 권한 부여 서비스 (JAAS), 컨테이너 관리 지속성 (CMP) 과 같은 관리형 컨테이너 리소스를 사용하는 Java EE 응용 프로그램에는 더 많은 노력이 필요합니다.

Oracle 애플리케이션 서버용으로 개발된 애플리케이션은 Oracle Identity Management 제품군을 사용하는 경우가 많습니다. 오픈 소스 애플리케이션 서버로 마이그레이션하는 고객은 SAML 기반 페더레이션을 사용하여 ID 및 액세스 관리를 다시 구현하기로 선택하는 경우가 많습니다. Oracle ID 관리 제품군에서 마이그레이션할 수 없는 경우 Oracle HTTP Server Webgate를 사용하는 경우도 있습니다.

Java 및 Java EE 웹 애플리케이션은 AWS Fargate 및 Amazon ECS와 같은 Docker 기반 AWS 서비스에 배포하기에 적합합니다. 고객은 대상 애플리케이션 서버(예: TomEE)의 최신 버전과 Java 개발 키트

(JDK)가 사전 설치된 Docker 이미지를 선택하는 경우가 많습니다. 기본 Docker 이미지 위에 애플리케이션을 설치하고, Amazon Elastic Container Registry(Amazon ECR) 레지스트리에 게시하고, 이를 사용하여 AWS Fargate 또는 Amazon ECS에서 애플리케이션을 확장 가능한 방식으로 배포합니다.

이상적으로는 애플리케이션 배포가 탄력적입니다. 즉, 트래픽 또는 워크로드에 따라 애플리케이션 인스턴스 수를 확장하거나 축소할 수 있습니다. 즉, 수요에 맞게 용량을 조정하려면 애플리케이션 인스턴스를 온라인 상태로 전환하거나 종료해야 합니다.

Java 애플리케이션을 AWS로 이전할 때는 스테이트리스로 만드는 것을 고려해 보세요. 이는 컨테이너화를 사용하여 수평적 규모 조정을 가능하게 하는 AWS Well-Architected Framework의 주요 아키텍처 원칙입니다. 예를 들어, 대부분의 Java 기반 웹 애플리케이션은 사용자 세션 정보를 로컬에 저장합니다. Amazon Elastic Compute Cloud(Amazon EC2)의 자동 조정 또는 기타 이유로 인한 애플리케이션 인스턴스 종료에서 살아남으려면 웹 애플리케이션 사용자가 웹 애플리케이션에 다시 연결하거나 다시 로그인하지 않고도 원활하고 투명하게 작업을 계속할 수 있도록 사용자 세션 정보를 전역적으로 저장해야 합니다. 이 접근 방식에는 Amazon ElastiCache for Redis나 글로벌 데이터베이스에 세션 상태 저장 등 여러 가지 아키텍처 옵션이 있습니다. TomEE와 같은 애플리케이션 서버에는 Redis, 데이터베이스 및 기타 글로벌 데이터 스토어를 통해 세션을 저장하고 관리할 수 있는 플러그인이 있습니다.

Amazon 및 AWS X-Ray와 쉽게 통합되는 일반적인 중앙 집중식 로깅 CloudWatch 및 디버깅 도구를 사용하십시오. 마이그레이션으로 애플리케이션 수명 주기 기능을 개선할 수 있습니다. 예를 들어 지속적 통합 및 지속적 전달(CI/CD) 파이프라인을 사용하여 쉽게 변경할 수 있도록 빌드 프로세스를 자동화할 수 있습니다. 이렇게 하려면 가동 중지 시간 없이 배포할 수 있도록 애플리케이션을 변경해야 할 수 있습니다.

사전 조건 및 제한 사항

사전 조건

- 활성 상태의 AWS 계정.
- 소스: Java 코드 및 JDK
- Oracle로 구축한 소스 애플리케이션 WebLogic
- ID 및 액세스 관리를 위한 정의된 솔루션(SAML 또는 Oracle Webgate)
- 애플리케이션 세션 관리를 위한 정의된 솔루션 (이동 like-for-like 또는 ElastiCache Amazon으로 이동 또는 필요한 경우 애플리케이션을 상태 비저장으로 설정)
- 팀에서 Apache TomEE로의 이식성을 위해 J2EE 전용 라이브러리를 리팩토링해야 하는지 여부 이해(Apache 웹사이트의 [Java EE 7 구현 상태](#) 참조)
- 보안 요구 사항을 기반으로 강화된 TomEE 이미지
- 대상 TomEE가 사전 설치된 컨테이너 이미지

- 필요한 경우 애플리케이션 문제 해결 합의 및 구현(예: 디버그, 빌드 로깅 및 인증)

제품 버전

- 오라클 WebLogic OC4J, 9i, 10g
- Tomcat 7(Java 1.6 이상)

아키텍처

소스 기술 스택

- 오라클을 사용하여 구축한 웹 애플리케이션 WebLogic
- Oracle Webgate 또는 SAML 인증을 사용하는 웹 애플리케이션
- Oracle Database 버전 10g 이상에 연결된 웹 애플리케이션

대상 기술 스택

- Amazon ECS에서 실행되는 TomEE(추가 컨테이너 지원이 포함된 Apache Tomcat)([Java 웹 애플리케이션 배포](#) 및 [Amazon ECS의 Java 마이크로서비스](#) 참조)
- Amazon Relational Database Service(Amazon RDS) for Oracle. Amazon RDS에서 지원하는 Oracle 버전의 경우 [Amazon RDS for Oracle](#)을 참조하세요.

대상 아키텍처

도구

TomEE에서 작동하려면 Java 애플리케이션을 .war 파일로 다시 빌드해야 합니다. 경우에 따라 TomEE에서 애플리케이션을 작동하기 위해 애플리케이션 변경이 필요할 수 있습니다. 필요한 구성 옵션과 환경 속성이 올바르게 정의되었는지 확인해야 합니다.

또한 Java 네이밍 및 디렉터리 인터페이스 (JNDI) 조회 및 JavaServer 페이지 (JSP) 네임스페이스를 올바르게 정의해야 합니다. 내장 T 라이브러리와 이름 충돌을 피하려면 응용 프로그램에서 사용하는 파일 이름을 확인하는 것이 좋습니다. 예를 들어, persistence.xml은 Apache OpenJPA 프레임워크(TomEE의 OpenEJB와 함께 제공됨)에서 구성 목적으로 사용하는 파일 이름입니다. PUI의 persistence.xml 파일에는 스프링 프레임워크 빈 선언이 포함되어 있습니다.

TomEE 버전 7.0.3 이상(Tomcat 8.5.7 이상)에서는 특수 문자가 포함된 원시(인코딩되지 않은) URL에 대해 HTTP 400 응답(잘못된 요청)을 반환합니다. 서버 응답은 최종 사용자에게 빈 페이지로 표시됩니다. 이전 버전의 TomEE와 Tomcat에서는 URL에 인코딩되지 않은 특정 특수 문자를 사용할 수 있었지만 [CVE-2016-6816 웹사이트](#)에 명시된 바와 같이 안전하지 않은 것으로 간주됩니다. URL 인코딩 문제를 해결하려면 를 통해 브라우저에 직접 전달되는 URL을 원시 문자열로 사용하는 대신 `encodeURIComponent()` 메서드로 JavaScript 인코딩해야 합니다.

.war 파일을 TomEE에 배포한 후 Linux `cat`의 로그 시작에서 누락된 공유 라이브러리가 있는지 확인하고 Oracle 전용 확장 프로그램을 모니터링하여 Tomcat 라이브러리에서 누락된 구성 요소를 추가합니다.

일반 절차

- TomEE에서 애플리케이션을 구성합니다.
- 애플리케이션 서버별 구성 파일 및 리소스를 소스에서 대상 형식으로 식별하고 재구성합니다.
- JNDI 리소스를 식별하고 재구성합니다.
- 대상 애플리케이션 서버에 필요한 형식으로 EJB 네임스페이스와 검색을 조정합니다(해당하는 경우).
- JAAS 애플리케이션 컨테이너별 보안 역할 및 기본 매핑을 재구성합니다(해당하는 경우).
- 애플리케이션 및 공유 라이브러리를 .war 파일로 패키징합니다.
- 제공된 Docker 컨테이너를 사용하여 .war 파일을 TomEE에 배포합니다.
- 로그 시작을 모니터링하여 누락된 공유 라이브러리 및 배포 설명자 확장을 식별합니다. 발견된 항목이 있으면 첫 번째 작업으로 돌아갑니다.
- 복원된 Amazon RDS 데이터베이스에서 설치된 애플리케이션을 테스트합니다.
- [Docker 컨테이너 배포](#)의 지침에 따라 로드 밸런서와 Amazon ECS 클러스터를 사용하여 전체 아키텍처를 시작합니다.
- 로드 밸런서를 가리키도록 URL을 업데이트합니다.
- 구성 관리 데이터베이스(CMDB)를 업데이트합니다.

에픽

마이그레이션 계획

작업	설명	필요한 기술
애플리케이션 검색 수행(현재 상태가 차지하는 공간 및 성능 기준)		BA, 마이그레이션 책임자
소스 및 대상 데이터베이스 버전과 엔진을 검증합니다.		DBA
소스 및 대상 애플리케이션 디자인(ID 및 세션 관리)을 검증합니다.		DBA, 마이그레이션 엔지니어, 애플리케이션 소유자
대상 서버 인스턴스의 하드웨어 및 스토리지 요구 사항을 식별합니다.		DBA, SysAdmin
용량, 스토리지 기능, 네트워크 기능에 따라 적절한 인스턴스 유형을 선택합니다.		DBA, SysAdmin
소스 및 대상 데이터베이스의 네트워크 액세스 보안 요구 사항을 확인합니다.		DBA, SysAdmin
애플리케이션 마이그레이션 전략 및 도구를 식별합니다.		DBA, 마이그레이션 책임자
애플리케이션에 대한 마이그레이션 설계 및 마이그레이션 가이드를 작성합니다.		빌드 책임자, 마이그레이션 책임자
애플리케이션 마이그레이션 런북을 완성합니다.		빌드 책임자, 전환 리드, 테스트 책임자, 마이그레이션 책임자

인프라 구성

작업	설명	필요한 기술
Virtual Private Cloud(VPC)를 생성합니다.		SysAdmin
보안 그룹을 생성합니다.		SysAdmin
Amazon RDS DB 인스턴스를 구성하고 시작합니다.		DBA, SysAdmin
Amazon ECS 배포를 구성합니다.		SysAdmin
애플리케이션을 Docker 이미지로 패키징합니다.		SysAdmin
이미지를 Amazon ECR 레지스트리로 푸시하거나 이 단계를 건너뛰고 Amazon ECS 클러스터로 푸시합니다.		SysAdmin
애플리케이션 및 Amazon ECS 서비스 옵션에 대한 작업 정의를 구성합니다.		SysAdmin
클러스터를 구성하고, 보안 설정을 검토하고, AWS Identity and Access Management (IAM) 역할을 설정합니다.		SysAdmin
설정을 시작하고 애플리케이션 마이그레이션 런북에 따라 테스트를 실행합니다.		SysAdmin

데이터 마이그레이션

작업	설명	필요한 기술
프로덕션 데이터를 AWS로 이전하려면 보안 보증 팀의 허가를 받습니다.		DBA, 마이그레이션 엔지니어, 애플리케이션 소유자
데이터베이스 백업 파일을 가져오기 위한 엔드포인트를 생성하고 이에 대한 액세스 권한을 확보합니다.		DBA
네이티브 데이터베이스 엔진 또는 타사 도구를 사용하여 데이터베이스 객체 및 데이터를 마이그레이션합니다.		DBA
애플리케이션 마이그레이션 런북에서 필요한 테스트를 실행하여 성공적인 데이터 마이그레이션을 확인합니다.		DBA, 마이그레이션 엔지니어, 애플리케이션 소유자

애플리케이션 마이그레이션

작업	설명	필요한 기술
마이그레이션을 위한 변경 요청(CR)을 생성합니다.		전환 리드
마이그레이션을 위한 CR 승인을 받습니다.		전환 리드
애플리케이션 마이그레이션 런북의 애플리케이션 마이그레이션 전략을 따릅니다.		DBA, 마이그레이션 엔지니어, 애플리케이션 소유자

작업	설명	필요한 기술
애플리케이션을 업그레이드합니다(필요한 경우).		DBA, 마이그레이션 엔지니어, 애플리케이션 소유자
전체 기능, 비기능, 데이터 검증, SLA 및 성능 테스트		테스트 책임자, 앱 소유자, 앱 사용자

전환

작업	설명	필요한 기술
애플리케이션 또는 비즈니스 소유자로부터 승인을 받습니다.		전환 리드
테이블 주제 연습을 실행하여 전환 런북의 모든 단계를 살펴봅니다.		DBA, 마이그레이션 엔지니어, 애플리케이션 소유자
애플리케이션 클라이언트를 새 인프라로 전환합니다.		DBA, 마이그레이션 엔지니어, 애플리케이션 소유자

프로젝트 닫기

작업	설명	필요한 기술
임시 AWS 리소스를 종료합니다.		DBA, 마이그레이션 엔지니어, SysAdmin
프로젝트 문서를 검토하고 검증하세요.		마이그레이션 책임자
마이그레이션 시간, 수동 대비 도구 비율, 비용 절감 등에 대한 지표를 수집합니다.		마이그레이션 책임자

작업	설명	필요한 기술
프로젝트를 마무리하고 피드백을 제공하세요.		마이그레이션 책임자, 앱 소유자

관련 리소스

참조

- [Apache Tomcat 7.0 설명서](#)
- [Apache Tomcat 7.0 설치 가이드](#)
- [Apache Tomcat JNDI 설명서](#)
- [Apache TomEE 설명서](#)
- [Amazon RDS for Oracle](#)
- [Amazon RDS 요금](#)
- [Oracle과 AWS](#)
- [Amazon RDS의 Oracle 설명서](#)
- [Amazon RDS 다중 AZ 배포](#)
- [Amazon ECS 시작하기](#)
- [Amazon RDS 시작하기](#)

자습서 및 동영상

- [Amazon RDS에서 Oracle Database를 실행하는 모범 사례](#)(2018년 re:Invent 발표)

AWS DMS를 사용하여 Amazon EC2에서 Amazon RDS for Oracle로 Oracle 데이터베이스 마이그레이션

작성자: Chethan Gangadharaiah(AWS)

R 유형: 리플랫폼	소스: 데이터베이스: 관계형	대상: Amazon RDS for Oracle
작성자: AWS	환경: PoC 또는 파일럿	기술: 데이터베이스; 마이그레이션
워크로드: Oracle	AWS 서비스: Amazon EC2, Amazon RDS	

요약

이 패턴은 AWS Database Migration Service(AWS DMS)를 사용하여 Amazon Elastic Compute Cloud(Amazon EC2) 기반의 Oracle 데이터베이스에서 Amazon Relational Database Service(RDS) for Oracle로 마이그레이션하는 단계를 설명합니다. 또한 이 패턴은 Oracle SQL Developer 또는 SQL *Plus를 사용하여 Oracle DB 인스턴스에 연결하며, 일부 작업을 자동화하는 AWS CloudFormation 템플릿도 포함되어 있습니다.

Amazon RDS for Oracle로 마이그레이션하면 Amazon RDS에서 데이터베이스 프로비저닝, 백업 및 복구, 보안 패치, 버전 업그레이드, 스토리지 관리와 같은 데이터베이스 관리 작업을 처리하는 동안 비즈니스와 애플리케이션에만 집중할 수 있습니다.

사전 조건 및 제한 사항

사전 조건

- 활성 상태의 AWS 계정
- Amazon EC2 기반 Oracle 데이터베이스용 Amazon Machine Image(AMI)

제품 버전

- AWS DMS는 Enterprise, Standard, Standard One, Standard Two 에디션용 Amazon RDS 인스턴스 데이터베이스용 Oracle 버전 11g(버전 11.2.0.3.v1 이상), 12c 및 18c를 지원합니다. 지원되는 버전

에 대한 최신 정보는 AWS 설명서의 [AWS DMS용 대상으로 Oracle 데이터베이스 사용](#)을 참조하세요. (첨부된 AWS CloudFormation 템플릿은 Oracle 버전 12c를 원본 데이터베이스로 사용합니다.)

- Oracle SQL Developer 4.0.3

아키텍처

소스 아키텍처

- Amazon EC2 기반 Oracle Database

대상 아키텍처

- Amazon RDS for Oracle

마이그레이션 아키텍처

도구

- [AWS DMS](#) – AWS Database Migration Service(AWS DMS)를 사용하여 데이터베이스를 AWS로 빠르고 안전하게 마이그레이션할 수 있습니다. 동종 마이그레이션과 이기종 마이그레이션을 모두 지원합니다. 지원되는 Oracle 데이터베이스 버전 및 에디션에 대한 자세한 내용은 AWS 설명서의 [AWS DMS용 소스로 Oracle 데이터베이스 사용](#) 및 [AWS DMS용 대상으로 Oracle 데이터베이스 사용](#)을 참조하세요.
- Oracle SQL Developer 또는 SQL*Plus – 이러한 도구를 사용하여 Amazon RDS for Oracle DB 인스턴스에 연결할 수 있습니다.

에픽

대상 데이터베이스 설정

작업	설명	필요한 기술
Amazon RDS for Oracle DB 인스턴스를 생성합니다.	AWS Management Console에 로그인한 후 https://console.aws.amazon.com/rds/ 에서	개발자

작업	설명	필요한 기술
	<p>Amazon RDS 콘솔을 엽니다. 적절한 엔진, 템플릿, 데이터베이스 보안 인증 설정, 인스턴스 유형, 스토리지, 다중 AZ 설정, Virtual Private Cloud(VPC) 및 구성, 로그인 보안 인증 정보, Oracle 데이터베이스의 추가 설정을 선택하여 Oracle DB 인스턴스를 생성합니다. 지침은 '관련 리소스' 섹션의 링크를 참조하세요. 또는 첨부 파일의 AWS CloudFormation 템플릿(Create_RDS.yaml)을 사용하여 오라클용 Amazon RDS DB 인스턴스를 생성할 수 있습니다.</p>	
<p>Amazon RDS에 연결하여 Oracle 사용자에게 권한을 부여합니다.</p>	<p>보안 그룹을 수정하여 로컬 시스템과 AWS DMS 복제 인스턴스에서 연결할 적절한 포트를 엽니다. 연결을 구성할 때 VPC 외부에서 데이터베이스에 연결할 수 있도록 '공개 액세스' 옵션을 선택해야 합니다. 로그인 보안 인증 정보를 사용하여 Oracle SQL Developer 또는 SQL*Plus로 Amazon RDS에 연결하고, AWS DMS 사용자를 생성하며, AWS DMS 사용자에게 데이터베이스를 수정하는 데 필요한 권한을 제공합니다.</p>	<p>개발자</p>

소스 EC2 인스턴스의 보안 그룹 구성

작업	설명	필요한 기술
Oracle 데이터베이스가 가동 및 실행 중인지 확인합니다.	Secure Shell(SSH)를 사용하여 EC2 인스턴스에 연결하고 SQL*Plus를 사용하여 Oracle 데이터베이스에 연결해 봅니다.	개발자
보안 그룹을 수정합니다.	로컬 시스템과 AWS DMS 복제 인스턴스에서 연결할 수 있도록 EC2 인스턴스의 보안 그룹을 수정하여 적절한 포트를 엽니다.	개발자

AWS DMS 설정

작업	설명	필요한 기술
AWS DMS 복제 인스턴스를 생성합니다.	AWS DMS에서 Amazon RDS for Oracle DB 인스턴스와 동일한 VPC에 복제 인스턴스를 생성합니다. 복제 인스턴스의 이름과 설명을 지정하고, 인스턴스 클래스와 복제 엔진 버전(기본값 사용)을 선택하고, Amazon RDS DB 인스턴스를 생성한 VPC를 선택하고, 필요한 경우 다중 AZ 설정을 지정하고, 스토리지를 할당하고, 가용 영역을 지정하고, 추가 설정을 구성합니다. 또는 첨부 파일의 AWS CloudFormation 템플릿(DMS.yaml)을 사용하여 이 단계를 구현할 수도 있습니다.	DBA

작업	설명	필요한 기술
<p>소스 및 대상 데이터베이스 엔드포인트에 연결합니다.</p>	<p>엔드포인트 식별자, 엔진, 서버, 포트, 로그인 보안 인증 정보 및 추가 연결 속성을 지정하여 소스 및 대상 데이터베이스 엔드포인트를 생성합니다. 소스 서버의 경우 Oracle 데이터베이스를 호스팅하는 EC2 인스턴스의 퍼블릭 DNS를 사용합니다. 대상 서버의 경우 Amazon RDS for Oracle의 엔드포인트를 사용합니다. 테스트를 실행하여 소스 및 대상 연결이 작동하는지 확인합니다. 또는 첨부 파일의 AWS CloudFormation 템플릿 (DMS.yaml) 을 사용하여 이 단계를 구현할 수도 있습니다.</p>	<p>DBA</p>

작업	설명	필요한 기술
AWS DMS 작업을 생성합니다.	AWS DMS 작업을 생성하여 소스 엔드포인트에서 대상 엔드포인트로 데이터를 마이그레이션하거나, 소스 엔드포인트와 대상 엔드포인트 간 복제를 설정하거나, 둘 다 수행합니다. AWS DMS 작업을 생성할 때 복제 인스턴스, 소스 엔드포인트, 대상 엔드포인트, 마이그레이션 유형(데이터만, 복제만 또는 둘 다), 테이블 매핑 및 필터를 지정합니다. Amazon CloudWatch DMS 작업을 실행하고, 작업을 모니터링하고, 테이블 통계를 확인하고, 로그를 확인합니다. 또는 첨부 파일의 AWS CloudFormation 템플릿 (DMS.yaml) 을 사용하여 이 단계를 구현할 수도 있습니다.	DBA

관련 리소스

- [Amazon RDS DB 인스턴스 생성](#)
- [Oracle 데이터베이스 엔진을 실행하는 DB 인스턴스에 연결](#)
- [AWS DMS 설명서](#)
- [AWS DMS 단계별 안내](#)
- [Oracle 데이터베이스를 AWS 클라우드로 마이그레이션](#)

첨부

이 문서와 관련된 추가 콘텐츠에 액세스하려면 [attachment.zip](#) 파일의 압축을 풉니다.

Logstash를 사용하여 온프레미스 Oracle 데이터베이스를 아마존 OpenSearch 서비스로 마이그레이션합니다.

작성자: 아디티야 고테티(AWS)

환경: PoC 또는 파일럿	소스: Oracle Database	대상: 아마존 OpenSearch 서비스
R 유형: 리플랫폼	워크로드: Oracle	기술: 마이그레이션; 데이터베이스
AWS 서비스: 아마존 OpenSearch 서비스		

요약

이 패턴은 Logstash를 사용하여 온프레미스 Oracle 데이터베이스에서 Amazon OpenSearch Service로 데이터를 이동하는 방법을 설명합니다. 여기에는 아키텍처 고려 사항, 몇 가지 필수 기술 세트 및 권장 사항이 포함됩니다. 데이터는 전체 텍스트 검색을 수행해야 하는 단일 테이블 또는 다중 테이블에서 가져올 수 있습니다.

OpenSearch 서비스는 가상 사설 클라우드 (VPC) 내에서 구성하거나 IP 기반 제한을 적용하여 공개적으로 배치할 수 있습니다. 이 패턴은 VPC 내에서 OpenSearch 서비스가 구성된 시나리오를 설명합니다. Logstash는 Oracle 데이터베이스에서 데이터를 수집하여 JSON 형식으로 파싱한 다음 서비스에 데이터를 공급하는 데 사용됩니다. OpenSearch

사전 조건 및 제한 사항

사전 조건

- 활성 상태의 AWS 계정
- Java 8(Logstash 6.4.3에서 요구함)
- AWS Virtual Private Network(AWS VPN)을 사용하여 구축된 VPC의 온프레미스 데이터베이스 서버와 Amazon Elastic Compute Cloud (Amazon EC2) 인스턴스 간의 연결
- Service로 푸시하는 데 필요한 데이터를 데이터베이스에서 검색하기 위한 쿼리입니다. OpenSearch
- Oracle JDBC(Java Database Connectivity) 드라이버

제한 사항

- Logstash는 데이터베이스에서 하드 삭제된 레코드를 식별할 수 없음

제품 버전

- Oracle Database 12c
- OpenSearch 서비스 6.3
- Logstash 6.4.3

아키텍처

소스 기술 스택

- 온프레미스 Oracle 데이터베이스
- 온프레미스 AWS VPN

대상 기술 스택

- VPC
- EC2 인스턴스
- OpenSearch 서비스
- Logstash
- NAT 게이트웨이(EC2 인스턴스의 운영 체제 업데이트 및 Java 8, Logstash 및 플러그인 설치용)

데이터 마이그레이션 아키텍처

도구

- Logstash 6.4.3
- JDBC 입력 플러그인([다운로드 및 추가 정보](#))
- [로그스태시 출력 플러그인 \(_es\) logstash-output-amazon](#)
- Oracle JDBC 드라이버

에픽

마이그레이션 계획

작업	설명	필요한 기술
소스 데이터베이스의 크기를 확인합니다.	소스 데이터의 크기는 인덱스에 구성할 샤드 수를 결정하는데 사용하는 파라미터 중 하나입니다.	DBA, 데이터베이스 개발자
각 열의 데이터 유형과 해당 데이터를 분석합니다.	OpenSearch 서비스는 이전에 볼 수 없었던 필드가 문서에서 발견되면 데이터 유형을 동적으로 매핑합니다. 명시적으로 선언해야 하는 특정 데이터 유형이나 형식(예: 날짜 필드)이 있는 경우, 인덱스 생성 중에 필드를 식별하고 해당 필드에 대한 매핑을 정의하십시오.	앱 소유자, 개발자, 데이터베이스 개발자
프라이머리 키 또는 고유 키가 있는 열이 있는지 확인하세요.	업데이트 또는 삽입 중에 Amazon OpenSearch Service의 레코드 중복을 방지하려면 amazon_es 플러그인의 출력 섹션에서 document_id 설정을 구성해야 합니다 (예: document_id => "%{customer_id}" where customer_id a a primary key).	앱 소유자, 개발자
추가된 새 레코드 수와 빈도를 분석하고 레코드가 얼마나 자주 삭제되는지 확인하세요.	이 작업은 소스 데이터 증가율을 이해하는 데 필요합니다. 데이터를 집중적으로 읽고 삽입이 거의 없으면 단일 인덱스를 사용할 수 있습니다. 새 레코드를 자주 삽입하고 삭제가 없으	앱 소유자, 개발자

작업	설명	필요한 기술
	면 샤드 크기는 최대 권장 크기인 50GB를 쉽게 초과할 수 있습니다. 이 경우 Logstash에서 인덱스 패턴을 구성하고 별칭을 사용하여 액세스할 수 있는 코드에서 인덱스 패턴을 구성하여 동적으로 인덱스를 생성할 수 있습니다.	
필요한 복제본 수를 결정합니다.		앱 소유자, 개발자
인덱스에 구성할 샤드 수를 결정합니다.		앱 소유자, 개발자
전용 프라이머리 노드, 데이터 노드 및 EC2 인스턴스의 인스턴스 유형을 확인합니다.	자세한 내용은 관련 리소스 섹션을 참조하세요.	앱 소유자, 개발자
필요한 전용 프라이머리 노드와 데이터 노드 수를 결정합니다.	자세한 내용은 관련 리소스 섹션을 참조하세요.	

데이터 마이그레이션

작업	설명	필요한 기술
EC2 인스턴스를 시작합니다.	AWS VPN이 연결된 VPC 내에서 EC2 인스턴스를 시작합니다.	Amazon VPC 구성, AWS VPN
EC2 인스턴스에 Logstash를 설치합니다.		개발자
Logstash 플러그인을 설치합니다.	필수 Logstash 플러그인 jdbc-input 및	개발자

작업	설명	필요한 기술
	logstash-output-amazon_es 을(를) 설치합니다.	
Logstash를 구성합니다.	Logstash 키스토어를 생성하여 AWS Secrets Manager 키 및 데이터베이스 보안 인증 정보 같은 민감한 정보를 저장한 다음 참조를 Logstash 구성 파일에 배치합니다.	개발자
DLQ(Dead Letter Queue)와 영구 대기열을 구성합니다.	기본적으로 데이터에 매핑 오류나 기타 문제가 포함되어 있어 처리할 수 없는 이벤트가 Logstash에서 발견되면 Logstash 파이프라인이 중단되거나 실패한 이벤트를 삭제합니다. 이 상황에서 데이터 손실을 방지하기 위해 실패한 이벤트를 삭제하는 대신 DLQ(Dead Letter Queue)에 기록하도록 Logstash를 구성할 수 있습니다. 비정상적 종료 시 데이터 손실을 방지하기 위해 Logstash에는 메시지 대기열을 디스크에 저장하는 영구 대기열 기능이 있습니다. 영구 대기열은 Logstash의 데이터 내구성을 제공합니다.	개발자

작업	설명	필요한 기술
Amazon OpenSearch 서비스 도메인을 생성합니다.	AWS ID 및 액세스 관리 (IAM) 자격 증명을 사용하여 요청에 서명할 필요가 없는 액세스 정책을 사용하여 Amazon OpenSearch 서비스 도메인을 생성합니다. Amazon OpenSearch 서비스 도메인은 동일한 VPC 내에 생성되어야 합니다. 또한 인스턴스 유형을 선택하고 분석을 기반으로 전용 및 프라이머리 노드의 수를 설정해야 합니다.	개발자
필수 Amazon OpenSearch 서비스 로그를 구성합니다.	자세한 내용은 OpenSearch 서비스 설명서를 참조하십시오.	
인덱스를 생성합니다.		개발자
Logstash를 시작합니다.	Logstash를 백그라운드 서비스로 실행합니다. Logstash는 구성된 SQL 쿼리를 실행하고 데이터를 가져와서 JSON 형식으로 변환하고 서비스에 공급합니다. OpenSearch 초기 로드 의 경우, Logstash 구성 파일에서 스케줄러를 구성하지 마세요.	개발자

작업	설명	필요한 기술
문서를 확인합니다.	<p>인덱스에 있는 문서 수와 소스 데이터베이스에 모든 문서가 있는지 확인합니다. 초기 로드 중에 인덱스에 추가되어 Logstash를 중지하는 데 사용 됩니다.</p> <p>Logstash 구성을 변경하여 클라이언트 요구 사항에 따라 고정된 간격으로 실행되는 스케줄러를 추가하고 Logstash를 다시 시작합니다. Logstash는 마지막 실행 이후에 업데이트되거나 추가된 레코드만 선택하며, 마지막 실행 타임스탬프는 Logstash 구성 파일의 <code>last_run_metadata_path => "/usr/share/logstash/.logstash_jdbc_last_run"</code> 속성으로 구성된 파일에 저장 됩니다.</p>	개발자

관련 리소스

- [권장 CloudWatch 알람](#)
- [전용 Amazon OpenSearch 서비스 마스터 노드](#)
- [Amazon OpenSearch 서비스 도메인 크기 조정](#)
- [Logstash 설명서](#)
- [JDBC 입력 플러그인](#)
- [Logstash 출력 플러그인](#)
- [아마존 OpenSearch 서비스 웹사이트](#)

온프레미스 Oracle 데이터베이스를 Amazon RDS for Oracle로 마이그레이션

작성자: Baji Shaik(AWS) 및 Pavan Pusuluri(AWS)

환경: PoC 또는 파일럿	소스: 데이터베이스: 관계형	대상: Amazon RDS for Oracle
R 유형: 리플랫폼	워크로드: Oracle	기술: 마이그레이션, 데이터베이스
AWS 서비스: Amazon RDS; AWS DMS		

요약

이 패턴은 온프레미스 Oracle 데이터베이스를 Amazon Relational Database Service(RDS) for Oracle로 마이그레이션하는 단계를 설명합니다. 마이그레이션 프로세스의 일부로 마이그레이션 계획을 세우고 소스 데이터베이스를 기반으로 대상 데이터베이스 인프라의 중요한 요소를 고려합니다. 비즈니스 요구 사항 및 사용 사례에 따라 두 가지 마이그레이션 옵션 중 하나를 선택할 수 있습니다.

1. [AWS Database Migration Service\(AWS DMS\)](#) - AWS DMS를 사용하면 데이터베이스를 빠르고 안전하게 AWS Cloud로 마이그레이션할 수 있습니다. 소스 데이터베이스는 마이그레이션 중에도 완전히 작동하여 데이터베이스를 사용하는 애플리케이션의 가동 중지 시간을 최소화합니다. [변경 데이터 캡처\(CDC\)](#) 라는 프로세스를 통해 초기 전체 로드 마이그레이션을 완료한 후 AWS DMS를 사용하여 진행 중인 변경 사항을 캡처하는 작업을 생성하면 마이그레이션 시간을 줄일 수 있습니다. 자세한 내용은 AWS 설명서의 [AWS DMS를 사용하여 Oracle에서 Amazon RDS로 마이그레이션](#)을 참조하십시오.
2. [기본 Oracle 도구 — Oracle GoldenGate for CDC의 Oracle 및 데이터 펌프 내보내기 및 데이터 펌프 가져오기와 같은 기본 Oracle 도구를 사용하여 데이터베이스를 마이그레이션할 수 있습니다.](#) 또한 원래 [내보내기 유틸리티](#) 및 원래 [가져오기 유틸리티](#) 같은 네이티브 Oracle 도구를 사용하여 전체 로드 시간을 줄일 수 있습니다.

사전 조건 및 제한 사항

사전 조건

- 활성 상태의 AWS 계정
- 온프레미스 Oracle 데이터베이스
- Amazon RDS Oracle 데이터베이스(DB) 인스턴스

제한 사항

- 데이터베이스 크기 제한: 64TB

제품 버전

- 버전 11g(버전 11.2.0.3.v1 이상), 12.2 이하 및 18c 지원되는 버전과 에디션의 최신 목록은 AWS 설명서의 [Amazon RDS for Oracle](#)을 참조하십시오. AWS DMS에서 지원하는 Oracle 버전의 경우, AWS DMS 설명서에서 [Oracle 데이터베이스를 AWS DMS의 소스로 사용하기](#) 섹션을 참조하십시오.

아키텍처

소스 기술 스택

- 온프레미스 Oracle 데이터베이스

대상 기술 스택

- Amazon RDS for Oracle

소스 및 대상 아키텍처

다음 다이어그램은 AWS DMS를 사용하여 온프레미스 Oracle 데이터베이스를 Amazon RDS for Oracle로 마이그레이션하는 방법을 보여줍니다.

이 다이어그램은 다음 워크플로를 보여줍니다.

1. 기존 데이터베이스 사용자를 생성하거나 사용하고, 해당 사용자에게 필요한 [AWS DMS 권한](#)을 부여하고, [ARCHIVELOG 모드](#)를 켜 다음, [추가 로깅](#)을 설정합니다.
2. 온프레미스와 AWS 네트워크 간에 인터넷 게이트웨이를 구성합니다.
3. AWS DMS의 [소스 및 대상 엔드포인트](#)를 구성합니다.

4. 소스 데이터베이스에서 대상 데이터베이스로 데이터를 마이그레이션하도록 [AWS DMS 복제 작업](#)을 구성합니다.
5. 대상 데이터베이스에서 마이그레이션 후 활동을 완료하십시오.

다음 다이어그램은 네이티브 Oracle 도구를 사용하여 온프레미스 Oracle 데이터베이스를 Amazon RDS for Oracle로 마이그레이션하는 방법을 보여줍니다.

이 다이어그램은 다음 워크플로를 보여줍니다.

1. Oracle Export (exp) 및 Import (imp) 유틸리티를 사용하여 기존 데이터베이스 사용자를 생성하거나 사용하고 Oracle 데이터베이스를 백업하는 데 필요한 권한을 부여하십시오.
2. 온프레미스와 AWS 네트워크 간에 인터넷 게이트웨이를 구성합니다.
3. 백업 데이터베이스를 가져오도록 [Bastion](#) 호스트의 Oracle 클라이언트를 구성합니다.
4. 백업 데이터베이스를 Amazon Simple Storage Service(S3) 버킷에 업로드합니다.
5. Amazon S3에서 Amazon RDS for Oracle 데이터베이스로 데이터베이스 백업을 복원합니다.
6. GoldenGate CDC에 맞게 오라클을 구성하십시오.
7. 대상 데이터베이스에서 마이그레이션 후 활동을 완료하십시오.

도구

- [AWS Database Migration Service\(AWS DMS\)](#)는 데이터 스토어를 AWS 클라우드로 마이그레이션하거나 클라우드와 온프레미스 설정의 조합 간에 마이그레이션하는 데 도움이 됩니다.
- 기본 Oracle 도구를 사용하면 동종 마이그레이션을 수행할 수 있습니다. [Oracle Data Pump](#)를 사용하여 소스 데이터베이스와 대상 데이터베이스 간에 데이터를 마이그레이션할 수 있습니다. 이 패턴은 Oracle Data Pump를 사용하여 소스 데이터베이스에서 대상 데이터베이스로 전체 로드를 수행합니다.
- [GoldenGateOracle](#)은 둘 이상의 데이터베이스 간에 논리적 복제를 수행할 수 있도록 지원합니다. 이 패턴은 GoldenGate Oracle Data Pump를 사용하여 초기 로드 후 델타 변경 사항을 복제하는 데 사용됩니다.

에픽

마이그레이션 계획

작업	설명	필요한 기술
<p>프로젝트 문서를 작성하고 데이터베이스 세부 정보를 기록하십시오.</p>	<ol style="list-style-type: none"> 1. 마이그레이션 목표, 마이그레이션 요구 사항, 주요 프로젝트 이해관계자, 프로젝트 마일스톤, 프로젝트 마감일, 주요 지표, 마이그레이션 위험, 위험 완화 계획을 문서화하십시오. 2. RAM, IOPS, CPU 등 소스 데이터베이스에 대한 중요한 정보를 문서화하십시오. 나중에 이 정보를 사용하여 적절한 대상 DB 인스턴스를 결정하게 됩니다. 3. 소스 및 대상 데이터베이스 버전을 검증합니다. 	DBA
<p>스토리지 요구 사항을 식별합니다.</p>	<p>다음에 포함된 스토리지 요구 사항을 확인하고 문서화하십시오.</p> <ol style="list-style-type: none"> 1. 소스 DB 인스턴스에 할당된 스토리지를 계산합니다. 2. 소스 DB 인스턴스에서 기간별 성장 지표를 수집합니다. 3. 대상 DB 인스턴스의 미래 성장률을 예측합니다. <p>참고: 범용(gp2) SSD 볼륨의 경우 스토리지 1GB당 IOPS 3개를 얻을 수 있습니다. 소스 데</p>	DBA, SysAdmin

작업	설명	필요한 기술
<p>컴퓨팅 요구 사항에 따라 적절한 인스턴스 유형을 선택합니다.</p>	<p>이터베이스의 총 읽기 및 쓰기 IOPS 수를 계산하여 스토리지를 할당하십시오.</p> <ol style="list-style-type: none"> 1. 대상 DB 인스턴스의 컴퓨팅 요구 사항을 결정합니다. 2. 성능 문제를 식별합니다. 3. 적절한 인스턴스 유형을 결정하는 요인을 고려하십시오. <ul style="list-style-type: none"> • 소스 DB 인스턴스의 CPU 사용률 • 소스 DB 인스턴스의 IOPS(읽기 및 쓰기) • 소스 DB 인스턴스의 메모리 사용량 	SysAdmin
<p>네트워크 액세스 보안 요구 사항을 파악하십시오.</p>	<ol style="list-style-type: none"> 1. 소스 및 대상 데이터베이스의 네트워크 액세스 보안 요구 사항을 파악하고 문서화합니다. 2. 애플리케이션이 데이터베이스와 통신할 수 있도록 적절한 보안 그룹을 구성하십시오. 	DBA, SysAdmin
<p>애플리케이션 마이그레이션 전략을 파악합니다.</p>	<ol style="list-style-type: none"> 1. 마이그레이션 전환 전략을 결정하고 문서화하십시오. 2. 애플리케이션의 복구 시간 목표 (RTO)와 복구 시점 목표 (RPO)를 결정 및 문서화한 다음 그에 따라 컷오버를 계획하십시오. 	DBA, 앱 소유자 SysAdmin

작업	설명	필요한 기술
마이그레이션 위험을 식별하십시오.	<p>데이터베이스를 평가하고 마이그레이션 관련 위험 및 완화 방법을 문서화하십시오. 예:</p> <ul style="list-style-type: none"> 로깅 없는 테이블을 식별하고 복구 시 데이터 손실 위험을 강조합니다. 소스 데이터베이스 사용자 및 권한을 추출하고 Amazon RDS 권한과의 충돌을 강조하십시오. 알림 로그를 검토하여 Oracle별 오류 및 경고가 있는지 확인하십시오. 대상 DB 인스턴스에서 지원되는 기능과 지원되지 않는 기능을 식별하십시오. 대상 DB 버전 엔진의 더 이상 사용되지 않는 기능을 검토하십시오. 	DBA

인프라 구성

작업	설명	필요한 기술
VPC를 생성합니다.	대상 DB 인스턴스에 대해 Amazon Virtual Private Cloud(VPC) 를 새로 생성합니다.	SysAdmin
보안 그룹을 생성합니다.	새 VPC에 보안 그룹을 생성 하여 DB 인스턴스로의 인바운드 연결을 허용합니다.	SysAdmin

작업	설명	필요한 기술
Amazon RDS for Oracle DB 인스턴스를 생성합니다.	새 VPC와 보안 그룹을 사용하여 대상 DB 인스턴스 를 만든 다음 인스턴스를 시작합니다.	SysAdmin

(옵션 1) 네이티브 Oracle 또는 타사 도구를 사용하여 데이터 마이그레이션

작업	설명	필요한 기술
소스 데이터베이스를 준비합니다.	<ol style="list-style-type: none"> Data Pump 디렉터리를 생성하거나 기존 디렉터를 사용하십시오. 마이그레이션 사용자를 만들고 데이터 펌프 추출을 수행할 권한을 부여하십시오. 소스 데이터베이스에서 역할, 사용자 및 테이블스페이스를 SQL 스크립트로 추출합니다. 추출된 Data Pump 덤프를 대상 DB 인스턴스 data pump 디렉터리로 전송합니다. 	DBA, SysAdmin
대상 데이터베이스를 준비합니다.	<ol style="list-style-type: none"> 대상 Amazon RDS for Oracle DB 인스턴스에 모든 데이터베이스 옵션(예를 들어, 텍스트 및 Java)이 설치 또는 활성화되었는지 확인합니다. Data Pump 디렉터리를 생성하거나 기존 디렉터를 사용하십시오. 	DBA, SysAdmin

작업	설명	필요한 기술
	<ol style="list-style-type: none"> 3. 마이그레이션 사용자를 만들고 데이터 펌프 가져오기를 수행할 권한을 부여하십시오. 4. 대상 DB 인스턴스에서 필요한 테이블스페이스, 사용자 및 역할을 생성합니다. 5. 전송된 Data Pump 내보내기 덤프를 대상 데이터베이스로 가져옵니다. 6. 가져오기 또는 객체 생성 중에 제외된 모든 인덱스를 생성합니다. 7. 가져오기 중에 제외된 제약 조건을 모두 생성합니다. 8. 잘못된 객체를 검증하거나 재컴파일합니다. 9. 잘못된 인덱스를 다시 빌드합니다. 10. 소스 데이터베이스와 대상 데이터베이스 간의 데이터베이스 개체 수를 확인합니다. 11. 객체 수 간에 불일치가 발견되면 이를 해결합니다. 	

(옵션 2) AWS DMS를 사용하여 데이터를 마이그레이션합니다.

작업	설명	필요한 기술
데이터를 준비하십시오.	<ol style="list-style-type: none"> 1. 소스 데이터베이스의 데이터를 정리합니다. 	DBA

작업	설명	필요한 기술
	<ol style="list-style-type: none"> 복제 인스턴스 생성 소스 엔드포인트와 대상 엔드포인트를 생성하십시오. 마이그레이션할 테이블 및 객체 수를 식별하십시오. 	
데이터를 마이그레이션하십시오.	<ol style="list-style-type: none"> 대상 데이터베이스에 외래 키 제약 조건 및 트리거를 삭제합니다. 대상 데이터베이스에 보조 인덱스를 삭제합니다. 소스 데이터베이스에서 대상 데이터베이스로 AWS DDS 전체 로드 작업 설정을 구성합니다. 외래 키를 활성화합니다. AWS DMS CDC를 활성화하여 진행 중인 변경 사항을 복제합니다. 트리거를 활성화합니다. 시퀀스를 업데이트하십시오. 소스 및 대상 데이터를 검증합니다. 	DBA

대상 데이터베이스로 전환

작업	설명	필요한 기술
애플리케이션 클라이언트를 새 인프라로 전환하십시오.	<ol style="list-style-type: none"> Oracle을 가리키는 모든 애플리케이션 서비스 및 클라이언트 연결을 중지하십시오. 	DBA SysAdmin, 앱 소유자

작업	설명	필요한 기술
	<ol style="list-style-type: none"> 2. AWS DMS 작업을 실행합니다. 3. 롤백 작업(예를 들어, Amazon RDS 데이터베이스에서 온프레미스 Oracle 데이터베이스로 CDC 역방향을 설정합니다). 4. 데이터를 검증합니다. 5. Amazon Route 53을 새 Amazon RDS for Oracle DB 인스턴스로 구성하여 새 대상 데이터베이스에서 애플리케이션 서비스를 시작합니다. 6. 새 오라클 RDS용 Amazon RDS DB 인스턴스에 Amazon CloudWatch 모니터링을 추가합니다. 	

작업	설명	필요한 기술
롤백 계획을 구현하십시오.	<ol style="list-style-type: none"> 1. Amazon RDS for Oracle DB 인스턴스를 가리키는 애플리케이션 서비스를 모두 중지합니다. 2. AWS DMS 작업을 사용하여 소스 온프레미스 Oracle 데이터베이스의 변경 내용을 롤백합니다. 3. 온프레미스 Oracle 데이터베이스에서 Amazon RDS for Oracle 데이터베이스로 실행되는 AWS DMS 작업을 중지하십시오. 4. 소스 Oracle 데이터베이스에서 애플리케이션을 다시 구성하십시오. 5. 롤백 배포가 완료되었는지 확인합니다. 	DBA, 앱 소유자

마이그레이션 프로젝트 종료

작업	설명	필요한 기술
리소스를 정리하십시오.	AWS DMS 복제 인스턴스 및 S3 버킷과 같은 임시 AWS 리소스를 종료하거나 제거합니다.	DBA, SysAdmin
프로젝트 문서를 검토하십시오.	마이그레이션 계획 문서와 목표를 검토한 다음 필요한 마이그레이션 단계를 모두 완료했는지 확인하십시오.	DBA SysAdmin, 앱 소유자

작업	설명	필요한 기술
지표를 수집합니다.	마이그레이션을 완료하는 데 걸린 시간, 수동 작업 대 도구 기반 작업의 비율, 비용 절감, 기타 관련 지표 등 주요 마이그레이션 지표를 기록하십시오.	DBA SysAdmin, 앱 소유자
프로젝트를 닫습니다.	마이그레이션 프로젝트를 닫고 노력에 대한 피드백을 수집하십시오.	DBA SysAdmin, 앱 소유자

관련 리소스

참조

- [Oracle 데이터베이스를 AWS로 마이그레이션하기 위한 전략](#)(AWS 백서)
- [AWS Database Migration Service](#)(AWS DMS 설명서)
- [Amazon RDS 요금](#) (Amazon RDS 설명서)

자습서 및 비디오

- [AWS Database Migration Service 시작하기](#)(AWS DMS 설명서)
- [Amazon RDS 요금](#)(Amazon RDS 설명서)
- [AWS 데이터베이스 마이그레이션 서비스 \(DMS\) \(YouTube\)](#)

Oracle Data Pump를 사용하여 온프레미스 Oracle 데이터베이스를 Amazon RDS for Oracle로 마이그레이션

작성자: Mohan Annam(AWS) 및 Brian motzer(AWS)

환경: PoC 또는 파일럿	소스: 데이터베이스: 관계형	대상: Amazon RDS for Oracle
R 유형: 리플랫폼	워크로드: Oracle	기술: 마이그레이션, 데이터베이스
AWS 서비스: Amazon RDS		

요약

이 패턴은 Oracle Data Pump를 사용하여 Oracle 데이터베이스를 온프레미스 데이터 센터에서 Amazon Relational Database Service(RDS) for Oracle DB 인스턴스로 마이그레이션하는 방법을 설명합니다.

패턴에는 소스 데이터베이스에서 데이터 덤프 파일을 생성하고 Amazon Simple Storage Service(S3) 버킷에 파일을 저장한 후 Amazon RDS for Oracle DB 인스턴스로 데이터를 복원하는 작업이 포함됩니다. 이 패턴은 마이그레이션을 하기 위해 AWS Database Migration Service(AWS DMS)를 사용할 때 제한이 발생하는 경우 유용합니다.

사전 조건 및 제한 사항

사전 조건

- 활성 상태의 AWS 계정
- AWS Identity and Access Management(IAM)에서 역할을 생성하고 Amazon S3 멀티파트 업로드에 필요한 권한
- 소스 데이터베이스에서 데이터를 내보내는 데 필요한 권한
- AWS Command Line Interface(AWS CLI) [설치](#) 및 [구성됨](#)

제품 버전

- Oracle Data Pump는 Oracle 데이터베이스 10g 릴리스 1(10.1) 이상 버전에서만 사용할 수 있습니다.

아키텍처

소스 기술 스택

- 온프레미스 Oracle 데이터베이스

대상 기술 스택

- Amazon RDS for Oracle
- SQL 클라이언트 (Oracle SQL Developer)
- S3 버킷

소스 및 대상 아키텍처

도구

서비스

- [AWS Identity and Access Management\(IAM\)](#)은 누구에게 인증 및 사용 권한이 있는지 제어하여 AWS 리소스에 대한 액세스를 안전하게 관리할 수 있도록 도와줍니다. 이 패턴에서 IAM은 Amazon S3에서 Amazon RDS for Oracle로 데이터를 마이그레이션하는 데 필요한 역할과 정책을 만드는 데 사용됩니다.
- [Amazon Relational Database Service\(RDS\)](#)은 AWS 클라우드에서 Oracle 관계형 데이터베이스를 설정하고, 운영하고, 규모를 조정하도록 도와줍니다.
- [Amazon Simple Storage Service\(S3\)](#)은 원하는 양의 데이터를 저장, 보호 및 검색하는 데 도움이 되는 클라우드 기반 객체 스토리지 서비스입니다.

기타 도구

- [Oracle Data Pump](#)를 사용하면 한 데이터베이스에서 다른 데이터베이스로 데이터와 메타데이터를 빠른 속도로 이동할 수 있습니다. 이 패턴에서는 Oracle Data Pump를 사용하여 데이터 덤프(.dmp) 파일을 Oracle 서버로 내보내고 이 파일을 Amazon RDS for Oracle로 가져옵니다. 자세한 내용은 Amazon RDS 설명서의 [Amazon RDS 기반 Oracle로 데이터 가져오기](#)를 참조하십시오.

- [Oracle SQL Developer](#)는 기존 배포와 클라우드 기반 배포 모두에서 Oracle 데이터베이스의 개발 및 관리를 간소화하는 통합 개발 환경입니다. 온프레미스 Oracle 데이터베이스 및 Amazon RDS for Oracle과 상호 작용하여 데이터를 내보내고 가져오는 데 필요한 SQL 명령을 실행합니다.

에픽

S3 버킷 생성

작업	설명	필요한 기술
버킷을 생성합니다.	S3 버킷을 만들려면 AWS 설명서 에 있는 지침을 따르십시오.	AWS 시스템 관리자

IAM 역할 생성 및 정책 할당

작업	설명	필요한 기술
IAM 권한을 구성합니다.	권한을 구성하려면 AWS 설명서 의 지침을 따르십시오.	AWS 시스템 관리자

대상 Amazon RDS for Oracle DB 인스턴스를 생성하고 Amazon S3 통합 역할을 연결합니다.

작업	설명	필요한 기술
대상 Amazon RDS for Oracle DB 인스턴스를 생성합니다.	Amazon RDS for Oracle 인스턴스를 생성하려면 AWS 설명서 의 지침을 따르십시오.	AWS 시스템 관리자
역할을 DB 인스턴스와 연결합니다.	역할을 인스턴스와 연결하려면 AWS 설명서의 지침을 따르십시오.	DBA

대상 데이터베이스에서 데이터베이스 사용자 생성

작업	설명	필요한 기술
사용자를 생성합니다.	<p>Oracle SQL Developer 또는 SQL*Plus에서 대상 Amazon RDS for Oracle 데이터베이스에 연결하고 다음 SQL 명령을 실행하여 스키마를 가져올 사용자를 생성합니다.</p> <pre>create user SAMPLE_SC HEMA identified by <PASSWORD>; grant create session, resource to <USER NAME>; alter user <USER NAME> quota 100M on users;</pre>	DBA

소스 Oracle 데이터베이스에서 내보내기 파일을 생성합니다.

작업	설명	필요한 기술
데이터 덤프 파일 생성	<p>SAMPLE_SCHEMA 사용자를 내보낼 DATA_PUMP_DIR 디렉터리에 sample.dmp 로 이름이 지정된 덤프 파일을 만들려면 다음 스크립트를 사용합니다.</p> <pre>DECLARE hdn1 NUMBER; BEGIN hdn1 := dbms_data pump.open(operation => 'EXPORT',</pre>	DBA

작업	설명	필요한 기술
	<pre> job_mode => 'SCHEMA', job_name => NULL); dbms_datapump.add_ file(handle => hdn1, filename => 'sample.dmp', directory => 'DATA_PUMP_DIR', filetype => dbms_datapump.ku\$_ file_type_dump_file); dbms_datapump.add_ file(handle => hdn1, filename => 'export.log', directory => 'DATA_PUMP_DIR', filetype => dbms_datapump.ku\$_ file_type_log_file); dbms_datapump.meta data_filter(hdn1, 'SCHEMA_EXPR', 'IN ('SAMPLE_SCHEMA')'); dbms_datapump.star t_job(hdn1); END; </pre>	

작업	설명	필요한 기술
	<p>/</p> <p>로컬 DATA_PUMP_DIR 디렉터리에 있는 export.log 파일을 검토하여 내보내기 세부 정보를 검토하십시오.</p>	

덤프 파일을 S3 버킷에 업로드합니다.

작업	설명	필요한 기술
소스에서 S3 버킷으로 데이터 덤프 파일을 업로드합니다.	<p>Amazon CLI를 사용하여 다음 명령을 실행합니다.</p> <pre>aws s3 cp sample.dmp s3://<bucket_created_epic_1>/</pre>	DBA

S3 버킷에서 RDS 인스턴스로 내보내기 파일을 다운로드하십시오.

작업	설명	필요한 기술
Amazon RDS에 데이터 덤프 파일을 다운로드하십시오.	<p>덤프 파일을 S3 버킷 sample.dmp 에서 Amazon RDS for Oracle 데이터베이스로 복사하려면 다음 SQL 명령을 실행합니다. 이 예시에서는 sample.dmp 파일이 S3 버킷 my-s3-integration1 에서 Oracle 디렉터리 DATA_PUMP_DIR 로 다운로드됩니다. 데이터베이스와 내보내기 파일을 모두 수용할 수 있을 만큼 RDS 인스턴스</p>	AWS 시스템 관리자

작업	설명	필요한 기술
	<p>에 충분한 디스크 스페이스가 할당되었는지 확인하십시오.</p> <pre data-bbox="592 331 1031 1003"> -- If you want to download all the files in the S3 bucket remove the p_s3_prefix line. SELECT rdsadmin. rdsadmin_s3_tasks. download_from_s3(p_bucket_name => 'my-s3-in tegration', p_s3_prefix => 'sample.dmp', p_directory_name => 'DATA_PUMP_DIR') AS TASK_ID FROM DUAL; </pre> <p>이전 명령은 작업 ID를 출력합니다. 작업 ID의 데이터를 검토하여 다운로드 상태를 검토하려면 다음 명령을 실행합니다.</p> <pre data-bbox="592 1260 1031 1575"> SELECT text FROM table(rdsadmin.rds _file_util.read_text_file('BDUMP','d btask-<task_id>.log')); </pre> <p>DATA_PUMP_DIR 디렉터리에서 파일을 보려면 다음 명령을 실행합니다.</p> <pre data-bbox="592 1785 1031 1875"> SELECT filename, type,filesize/1024 </pre>	

작업	설명	필요한 기술
	<pre> /1024 size_megs ,to_char(mtime,'DD -MON-YY HH24:MI:SS') timestamp FROM TABLE(rdsadmin.rds _file_util.listdir (p_directory => upper('DATA_PUMP_D IR')))) order by 4; </pre>	

대상 데이터베이스에 있는 덤프 파일을 가져옵니다.

작업	설명	필요한 기술
스키마와 데이터를 Amazon RDS로 복원합니다.	<p>덤프 파일을 sample_schema 데이터베이스 스키마로 가져오려면, SQL Developer 또는 SQL*Plus에서 다음 SQL 명령을 실행합니다.</p> <pre> DECLARE hdn1 NUMBER; BEGIN hdn1 := DBMS_DATA PUMP.OPEN(operation => 'IMPORT', job_mode => 'SCHEMA', job_name= >null); DBMS_DATAPUMP.ADD_ FILE(handle => hdn1, filename => 'sample.d mp', directory => 'DATA_PUMP_DIR', filetype => dbms_data pump.ku\$_file_type _dump_file); </pre>	DBA

작업	설명	필요한 기술
	<pre> DBMS_DATAPUMP.ADD_FILE (handle => hdn1, filename => 'import.log', directory => 'DATA_PUMP_DIR', filetype => dbms_data pump.ku\$_file_type _log_file); DBMS_DATAPUMP. METADATA_FILTER(hd n1, 'SCHEMA_EXPR', ' IN ('SAMPLE_SCHEMA')'); DBMS_DATAPUMP.START_J OB(hdn1); END; / </pre> <p>가져오기에서 로그 파일을 보려면 다음 명령을 실행합니다.</p> <pre> SELECT text FROM table(rdsadmin.rds _file_util.read_t xt_file('DATA_PUM P_DIR', 'import.log')); </pre>	

DATA_PUMP_DIR 디렉터리에서 덤프 파일 제거

작업	설명	필요한 기술
내보내기 파일을 나열하고 정리합니다.	DATA_PUMP_DIR 디렉터리의 내보내기 파일을 나열하고 제	AWS 시스템 관리자

작업	설명	필요한 기술
	<p>거한 뒤, 다음 명령을 실행합니다.</p> <pre data-bbox="597 331 1024 846"> -- List the files SELECT filename, type,filesize/1024 /1024 size_megs ,to_char(mtime,'DD -MON-YY HH24:MI:S S') timestamp FROM TABLE(rdsadmin.rds _file_util.listdir (p_directory => upper('DATA_PUMP_D IR')))) order by 4; </pre> <pre data-bbox="597 877 1024 1192"> -- Remove the files EXEC UTL_FILE. REMOVE('DATA_PUMP _DIR','sample.dmp'); EXEC UTL_FILE.REMOVE(' DATA_PUMP_DIR','im port.log'); </pre>	

관련 리소스

- [Amazon S3 통합](#)
- [DB 인스턴스 생성](#)
- [Amazon RDS의 Oracle로 데이터 가져오기](#)
- [Amazon S3 설명서](#)
- [IAM 설명서](#)
- [Amazon RDS 설명서](#)
- [Oracle Data Pump 설명서](#)
- [Oracle SQL Developer](#)

pglogical을 사용하여 Amazon EC2의 PostgreSQL에서 Amazon RDS for PostgreSQL로 마이그레이션합니다.

작성자: Rajesh Madiwale (AWS)

환경: PoC 또는 파일럿	소스: Amazon EC2	대상: Amazon RDS for PostgreSQL
R 유형: 리플랫폼	워크로드: 오픈 소스	기술: 마이그레이션, 데이터베이스
AWS 서비스: Amazon RDS		

요약

이 패턴은 PostgreSQL pglogical 확장 프로그램을 사용하여 PostgreSQL 데이터베이스(버전 9.5 이상)를 Amazon Elastic Compute Cloud(Amazon EC2)에서 PostgreSQL용 Amazon Relational Database Service(Amazon RDS)로 마이그레이션하는 단계를 설명합니다. Amazon RDS는 이제 PostgreSQL 버전 10의 pglogical 확장을 지원합니다.

사전 조건 및 제한 사항

사전 조건

- 적절한 유형의 Amazon RDS 인스턴스를 선택합니다. 자세한 내용은 [Amazon RDS 인스턴스 유형](#)을 참조하세요.
- PostgreSQL의 소스 버전과 대상 버전이 동일한지 확인합니다.
- Amazon EC2에서 [pglogical 확장 프로그램을 PostgreSQL](#)과 함께 설치하고 통합합니다.

제품 버전

- Amazon RDS에서 지원되는 기능을 포함한 Amazon RDS의 PostgreSQL 버전 10 이상입니다(AWS 설명서의 [Amazon RDS에서의 PostgreSQL](#) 참조). 이 패턴은 Amazon RDS에서 PostgreSQL 9.5를 PostgreSQL 버전 10으로 마이그레이션하여 테스트했지만, Amazon RDS의 PostgreSQL 이후 버전에도 적용됩니다.

아키텍처

데이터 마이그레이션 아키텍처

도구

- [pglogical](#) 확장 프로그램
- PostgreSQL 네이티브 유틸리티: [pg_dump](#) 및 [pg_restore](#)

에픽

pglogical 확장 프로그램을 사용하여 데이터를 마이그레이션합니다.

작업	설명	필요한 기술
Amazon RDS PostgreSQL DB 인스턴스를 생성합니다.	Amazon RDS에서 PostgreSQL DB 인스턴스를 설정합니다. 자세한 지침은 Amazon RDS for PostgreSQL 설명서 를 참조하세요.	DBA
소스 PostgreSQL 데이터베이스에서 스키마 덤프를 가져와 대상 PostgreSQL 데이터베이스로 복원합니다.	<ol style="list-style-type: none"> 1. pg_dump 유틸리티를 <code>-s</code> 옵션과 함께 사용하면 소스 데이터베이스에서 스키마 파일을 생성할 수 있습니다. 2. psql 유틸리티를 <code>-f</code> 옵션과 함께 사용하면 대상 데이터베이스에 스키마를 로드할 수 있습니다. 	DBA
논리적 디코딩을 사용 설정합니다.	Amazon RDS DB 파라미터 그룹의 <code>rds.logical_replication</code> 정적 파라미터를 1로 설정합니다. 자세한 지침은 Amazon RDS 설명서 를 참조하세요.	DBA

작업	설명	필요한 기술
<p>소스 및 대상 데이터베이스에 pglogical 확장 프로그램을 생성합니다.</p>	<p>1. 소스 PostgreSQL 데이터베이스에 pglogical 확장 프로그램을 생성합니다.</p> <pre data-bbox="634 401 1029 667">psql -h <amazon-ec2-endpoint> -d target-dbname -U target-dbuser -c "create extension pglogical ;"</pre> <p>2. 대상 PostgreSQL 데이터베이스에 pglogical 확장 프로그램을 생성합니다.</p> <pre data-bbox="634 856 1029 1123">psql -h <amazon-rds-endpoint> -d source-dbname -U source-dbuser -c "create extension pglogical ;"</pre>	DBA
<p>소스 PostgreSQL 데이터베이스에 게시자를 생성합니다.</p>	<p>게시자를 만들려면 다음을 실행합니다.</p> <pre data-bbox="597 1293 1029 1770">psql -d dbname -p 5432 <<EOF SELECT pglogical .create_node(node_name := 'provider1', dsn := 'host=<ec2-endpoint> port=5432 dbname=source-dbname user=source-dbuser'); EOF</pre>	DBA

작업	설명	필요한 기술
<p>복제 세트를 만들고, 테이블과 시퀀스를 추가합니다.</p>	<p>소스 PostgreSQL 데이터베이스에 복제 세트를 생성하고 복제 세트에 테이블과 시퀀스를 추가하려면 다음을 실행합니다.</p> <pre data-bbox="597 491 1024 890">psql -d dbname -p 5432 <<EOF SELECT pglogical .replication_set_a dd_all_tables('default', '{public} '::text[],synchronize_data := true); EOF</pre>	<p>DBA</p>
<p>구독자를 생성합니다.</p>	<p>대상 PostgreSQL 데이터베이스에 구독자를 생성하려면 다음을 실행합니다.</p> <pre data-bbox="597 1094 1024 1688">psql -h <rds-endpoint> -d target-database - U target-database-user <<EOF SELECT pglogical .create_node(node_name := 'subscriber1', dsn := 'host=<rds-endpoint> port=5432 database=target-database-name password=postgres user=target-database-user'); EOF</pre>	<p>DBA</p>

작업	설명	필요한 기술
구독을 생성합니다.	<p>대상 PostgreSQL 데이터베이스에서 구독을 생성하려면 다음을 실행합니다.</p> <pre>psql -h <rds-endpoint> -d target -U postgres <<EOF SELECT pglogical .create_subscription(subscription_name := 'subscription1', replication_sets := array['default'], provider_dsn := 'host=<ec2-endpoint> port=5432 dbname=<source-database-name> password=<password> user=source-database-user');</pre>	DBA

데이터 검증

작업	설명	필요한 기술
소스 및 대상 데이터베이스를 검사합니다.	<p>소스 및 대상 데이터베이스를 검사하여 데이터가 성공적으로 복제되고 있는지 확인합니다. 소스 및 대상 테이블에서 <code>select count(1)</code>를 사용하여 기본 검증을 수행할 수 있습니다.</p>	DBA

관련 리소스

- [Amazon RDS](#)

- [Amazon RDS의 PostgreSQL에 대한 논리적 복제\(Amazon RDS 설명서\)](#)
- 로그 로지컬 (리포지토리) GitHub
- 로그 로지컬 (GitHub 리포지토리 README 파일) 의 한계
- [논리적 복제를 사용하여 PostgreSQL을 온프레미스 또는 Amazon EC2에서 Amazon RDS로 마이그레이션하기\(AWS 데이터베이스 블로그\)](#)

온프레미스 PostgreSQL 데이터베이스를 Aurora PostgreSQL로 마이그레이션하기

작성자: Baji Shaik(AWS) 및 Jitender Kumar(AWS)

환경: PoC 또는 파일럿	소스: 온프레미스 PostgreSQL 데이터베이스	대상: Aurora PostgreSQL-Compatible
R 유형: 리플랫폼	워크로드: 오픈 소스	기술: 마이그레이션, 데이터베이스
AWS 서비스: Amazon Aurora, AWS DMS		

요약

Amazon Aurora PostgreSQL 호환 에디션은 하이엔드 상용 데이터베이스의 성능 및 가용성과 오픈 소스 데이터베이스의 단순성 및 비용 효율성을 결합한 제품입니다. Aurora는 동일한 AWS 리전의 3개 가용 영역에 걸쳐 스토리지를 확장하여 이러한 이점을 제공하며, 읽기 워크로드를 확장하고 단일 리전 내에서 고가용성을 제공하기 위해 최대 15개의 읽기 전용 복제본 인스턴스를 지원합니다. Aurora 글로벌 데이터베이스를 사용하면 최대 5개 리전에 PostgreSQL 데이터베이스를 복제하여 리전 장애 발생 시 원격 읽기 액세스 및 재해 복구를 수행할 수 있습니다. 이 패턴은 온프레미스 PostgreSQL 소스 데이터베이스를 Aurora PostgreSQL 호환 데이터베이스로 마이그레이션하는 단계를 설명합니다. [패턴에는 두 가지 마이그레이션 옵션이 포함됩니다. 하나는 AWS 데이터 마이그레이션 서비스\(AWS DMS\)를 사용하는 것이고, 다른 하나는 네이티브 PostgreSQL 도구\(예: pg_dump, pg_restore, psql\) 또는 타사 도구를 사용하는 것입니다.](#)

이 패턴에 설명된 단계는 Amazon Relational Database Service(RDS) 및 Amazon Elastic Compute Cloud(Amazon EC2) 인스턴스상의 대상 PostgreSQL 데이터베이스에도 적용됩니다.

사전 조건 및 제한 사항

사전 조건

- 활성 상태의 AWS 계정
- 온프레미스 데이터 센터의 PostgreSQL 소스 데이터베이스
- [Aurora PostgreSQL 호환 DB 인스턴스 또는 Amazon RDS for PostgreSQL DB 인스턴스](#)

제한 사항

- 데이터베이스 크기 제한은 Amazon RDS for PostgreSQL의 경우 64TB, Aurora PostgreSQL과 호환되는 경우 128TB입니다.
- AWS DMS 마이그레이션 옵션을 사용하는 경우 [PostgreSQL 데이터베이스를 소스로 사용하는 것에 대한 AWS DMS 제한 사항을 검토하십시오.](#)

제품 버전

- Amazon RDS의 PostgreSQL 메이저 버전 및 마이너 버전 지원에 대한 내용은 Amazon RDS 설명서에서 [Amazon RDS for PostgreSQL](#) 업데이트를 참조하십시오.
- Aurora에서의 PostgreSQL 지원에 대한 내용은 Aurora 설명서에서 [Amazon Aurora PostgreSQL 업데이트](#)를 참조하십시오.
- AWS DMS 마이그레이션 옵션을 사용하는 경우, AWS DMS 설명서에서 [지원되는 PostgreSQL 버전](#)을 참조하십시오.

아키텍처

소스 기술 스택

- 온프레미스 PostgreSQL 데이터베이스

대상 기술 스택

- Aurora PostgreSQL 호환 DB 인스턴스

소스 아키텍처

대상 아키텍처

데이터 마이그레이션 아키텍처

AWS DMS 사용

네이티브 PostgreSQL 도구 사용

도구

- [AWS Database Migration Service\(AWS DMS\)](#)를 사용하면 데이터 스토어를 AWS 클라우드로 마이그레이션하거나 클라우드와 온프레미스 구성을 조합하여 마이그레이션할 수 있습니다. 이 서비스는 다양한 소스 및 대상 데이터베이스를 지원합니다. AWS DMS에서 사용할 수 있도록 지원되는 PostgreSQL 소스 및 대상 데이터베이스 버전과 에디션을 검증하는 방법에 대한 자세한 내용은 [PostgreSQL 데이터베이스를 AWS DMS 소스로 사용](#)을 참조하십시오. 가장 포괄적인 버전 및 기능 지원을 위해 최신 버전의 AWS DMS를 사용하는 것을 권장합니다.
- [네이티브 PostgreSQL 도구에는 pg_dump, pg_restore 및 psql이 포함됩니다.](#)

에픽

마이그레이션 분석

작업	설명	필요한 기술
소스 및 대상 데이터베이스 버전을 검증합니다.	AWS DMS를 사용하는 경우 PostgreSQL의 지원되는 버전을 사용하고 있는지 확인 합니다.	DBA
스토리지 유형 및 용량 요구 사항을 확인하십시오.	<ol style="list-style-type: none"> 1. 소스 데이터베이스 인스턴스에 할당된 스토리지를 계산합니다. 2. 소스 데이터베이스 인스턴스의 과거 증가 지표를 수집하십시오. 3. 대상 데이터베이스 인스턴스의 향후 성장 예측을 예측합니다. 4. 소스 데이터베이스의 총 읽기 및 쓰기 IOPS 수를 계산하여 스토리지를 할당합니다. 	DBA, 시스템 관리자

작업	설명	필요한 기술
<p>적절한 인스턴스 유형, 용량, 스토리지 기능 및 네트워크 기능을 선택합니다.</p>	<p>다. 범용 SSD (gp2) 볼륨은 스토리지 1GB당 3 IOPS를 제공합니다.</p> <p>대상 데이터베이스 인스턴스의 컴퓨팅 요구 사항을 결정합니다. 추가 주의가 필요할 수 있는 알려진 성능 문제를 검토하십시오. 다음 요소를 고려하여 적절한 인스턴스 유형을 결정하십시오.</p> <ul style="list-style-type: none"> 소스 데이터베이스 인스턴스의 CPU 사용률 소스 데이터베이스 인스턴스의 IOPS (읽기 및 쓰기 작업) 소스 데이터베이스 인스턴스의 메모리 사용량 <p>자세한 내용은 Aurora 설명서의 Aurora DB 인스턴스 클래스를 참조하십시오.</p>	<p>DBA, 시스템 관리자</p>
<p>소스 및 대상 데이터베이스의 네트워크 액세스 보안 요구 사항을 식별합니다.</p>	<p>애플리케이션이 데이터베이스와 통신할 수 있도록 하는 적절한 보안 그룹을 결정하십시오.</p>	<p>DBA, 시스템 관리자</p>

작업	설명	필요한 기술
애플리케이션 마이그레이션 전략을 파악합니다.	<ul style="list-style-type: none"> 애플리케이션의 복잡성에 따라 마이그레이션 전환 전략을 결정하십시오. 애플리케이션에 대한 Recovery Time Objective (RTO) 및 Recovery Point Objective(RPO)를 결정하고 이에 따라 전환을 계획합니다. 	DBA, 앱 소유자, 시스템 관리자

인프라 구성

작업	설명	필요한 기술
VPC를 생성합니다.	대상 데이터베이스 인스턴스에 대한 새 Virtual Private Cloud(VPC)를 생성합니다.	시스템 관리자
보안 그룹을 생성합니다.	이전 에픽에서 결정한 대로 VPC 내에 보안 그룹을 생성하여 데이터베이스 인스턴스로의 인바운드 연결을 허용합니다.	시스템 관리자
Aurora DB 클러스터를 구성하고 시작합니다.	새 VPC와 보안 그룹으로 대상 데이터베이스 인스턴스를 만들고 인스턴스를 시작합니다.	시스템 관리자

데이터 마이그레이션 – 옵션 1 (AWS DMS 사용)

작업	설명	필요한 기술
마이그레이션 전 단계를 완료하십시오.	<ol style="list-style-type: none"> 소스 데이터베이스의 데이터를 정리합니다. 복제 인스턴스 생성 	DBA

작업	설명	필요한 기술
	<ol style="list-style-type: none"> 3. 소스 및 대상 엔드포인트를 생성합니다. 4. 마이그레이션할 수 있는 테이블 및 객체 수를 식별하십시오. 	
<p>마이그레이션 단계를 완료합니다.</p>	<ol style="list-style-type: none"> 1. 대상 데이터베이스에 외래 키 제약 조건 및 트리거를 삭제합니다. 2. 대상 데이터베이스에 보조 인덱스를 삭제합니다. 3. 전체 로드 작업을 사용하여 소스에서 대상 데이터베이스로 데이터를 마이그레이션합니다. 4. 외래 키를 활성화합니다. 5. 플래시컷 마이그레이션을 사용하고 애플리케이션 가동 중지 시간을 최소화해야 하는 경우 변경 데이터 캡처 (CDC)를 활성화하여 진행 중인 변경 사항을 복제하십시오. 6. 트리거를 활성화합니다. 7. 업데이트 시퀀스. 8. 소스 및 대상 데이터를 검증합니다. 	DBA
<p>데이터를 검증합니다.</p>	<p>데이터가 소스에서 대상으로 정확하게 마이그레이션되었는지 확인하려면 AWS DMS 설명서의 데이터 검증 단계를 따르십시오.</p>	DBA

데이터 마이그레이션 – 옵션 2 (pg_dump 및 pg_restore를 사용합니다.)

작업	설명	필요한 기술
소스 데이터베이스를 준비합니다.	<ol style="list-style-type: none"> 1. pg_dump 백업이 아직 없는 경우 해당 디렉토리를 생성하여 저장합니다. 2. 데이터베이스 객체에서 pg_dump를 실행할 권한이 있는 마이그레이션 사용자를 생성하십시오. 3. EC2 인스턴스에 연결하고 pg_dump 백업을 실행합니다. <p>자세한 내용은 pg_dump 설명서 및 AWS DMS 설명서의 안내를 참조하십시오.</p>	DBA
대상 데이터베이스를 준비합니다.	<ol style="list-style-type: none"> 1. 데이터베이스 개체에 pg_restore를 사용할 권한이 있는 마이그레이션 사용자를 생성하십시오. 2. pg_restore를 사용하여 데이터베이스 덤프를 가져옵니다. <p>자세한 내용은 pg_restore 설명서 및 AWS DMS 설명서의 안내를 참조하십시오.</p>	DBA
데이터를 검증합니다.	<ol style="list-style-type: none"> 1. 소스 데이터베이스와 대상 데이터베이스 간의 데이터베이스 개체 수를 비교합니다. 	DBA

작업	설명	필요한 기술
	2. 객체 수 간에 불일치가 발견되면 이를 해결합니다.	

애플리케이션 마이그레이션

작업	설명	필요한 기술
애플리케이션 마이그레이션 전략을 따릅니다.	첫 번째 에픽에서 만든 애플리케이션 마이그레이션 전략을 구현하십시오.	DBA, 앱 소유자, 시스템 관리자

대상 데이터베이스로 전환

작업	설명	필요한 기술
애플리케이션 클라이언트를 새 인프라로 전환합니다.	<ol style="list-style-type: none"> 1. 온프레미스 PostgreSQL 데이터베이스를 가리키는 모든 애플리케이션 서비스 및 클라이언트 연결을 중지합니다. 2. AWS DMS 작업을 실행합니다. 3. 필요한 경우 롤백 작업 (Aurora PostgreSQL과 호환되는 CDC를 온프레미스 PostgreSQL 데이터베이스로 역방향)을 설정합니다. 4. 데이터를 검증합니다. 5. Amazon Route 53을 새로운 Aurora PostgreSQL 호환 DB 인스턴스로 구성하여 새 대상에서 애플리케이션 서비스를 시작하십시오. 	DBA, 앱 소유자, 시스템 관리자

작업	설명	필요한 기술
	6. 새로운 Aurora PostgreSQL 호환 DB 인스턴스에 Amazon CloudWatch 및 Performance Insights 모니터링을 추가하십시오.	
마이그레이션을 롤백해야 하는 경우.	<ol style="list-style-type: none"> 1. Aurora PostgreSQL 호환 데이터베이스를 가리키는 모든 애플리케이션 서비스를 중지합니다. 2. 이전 스토리에서 생성한 AWS DMS 작업을 사용하여 소스 온프레미스 PostgreSQL 데이터베이스의 변경 내용을 롤백합니다. 3. 온프레미스 PostgreSQL 데이터베이스에서 Aurora PostgreSQL 호환 데이터베이스로 실행되는 AWS DMS 작업을 중지하십시오. 4. 소스 온프레미스 PostgreSQL 데이터베이스를 가리키도록 애플리케이션을 구성합니다. 5. 모든 롤백 배포가 완료되었는지 확인합니다. 	DBA, 앱 소유자

프로젝트 닫기

작업	설명	필요한 기술
리소스를 종료합니다.	임시 AWS 리소스를 종료합니다.	DBA, 시스템 관리자

작업	설명	필요한 기술
문서를 검증합니다.	프로젝트 문서를 검토하고 검증하세요.	DBA, 앱 소유자, 시스템 관리자
지표를 수집합니다.	마이그레이션 시간, 수동 비용 대비 도구 비용 절감 비율 등에 대한 지표를 수집하십시오.	DBA, 앱 소유자, 시스템 관리자
프로젝트 닫기	프로젝트를 종료하고 피드백을 제공하세요.	DBA, 앱 소유자, 시스템 관리자

관련 리소스

참조

- [AWS Data Migration Service](#)
- [VPC와 Amazon Aurora](#)
- [Amazon Aurora 요금](#)
- [PostgreSQL 데이터베이스를 AWS DMS 소스로 사용](#)
- [AWS DMS 복제 인스턴스를 생성하는 방법](#)
- [AWS DMS를 사용하여 소스 및 대상 엔드포인트를 생성하는 방법](#)

추가 리소스

- [AWS DMS 시작하기](#)
- [데이터 마이그레이션 안내 step-by-step](#)
- [Amazon Aurora 리소스](#)

Linux가 실행되는 Amazon EC2의 Microsoft SQL Server로 온프레미스 Microsoft SQL Server 데이터베이스의 마이그레이션

작성자: Tirumala Rama Chandra Murty Dasari(AWS)

R 유형: 리플랫폼	소스: 데이터베이스: 관계형	대상: Microsoft SQL Server가 포함된 Amazon EC2 Linux
작성자: AWS	환경: PoC 또는 파일럿	기술: 데이터베이스, 마이그레이션
워크로드: Microsoft	AWS 서비스: Amazon EC2	

요약

이 패턴은 백업 및 복원 유틸리티를 사용하여 Microsoft Windows에서 실행되는 온프레미스 Microsoft SQL Server 데이터베이스에서 Amazon Elastic Compute Cloud(Amazon EC2) Linux 인스턴스에서 실행되는 Microsoft SQL Server로 마이그레이션하는 방법을 설명합니다.

사전 조건 및 제한 사항

사전 조건

- 활성 상태의 AWS 계정
- Microsoft SQL 서버를 사용하는 Amazon EC2 Linux AMI (Amazon Machine Image)
- 온프레미스 Windows와 Linux EC2 인스턴스의 Microsoft SQL Server 사이의 Direct Connect

아키텍처

소스 기술 스택

- 온프레미스 Microsoft SQL Server 데이터베이스

대상 기술 스택

- Microsoft SQL Server 데이터베이스가 포함된 Linux EC2 인스턴스

데이터베이스 마이그레이션 아키텍처

도구

- WinSCP - 이 도구를 사용하면 Windows 사용자가 Linux 사용자와 파일을 쉽게 공유할 수 있습니다.
- Sqlcmd - 이 명령줄 유틸리티를 사용하면 T-SQL 문 또는 배치를 SQL Server의 로컬 및 원격 인스턴스에 제출할 수 있습니다. 이 유틸리티는 배치 처리 또는 유닛 테스트와 같은 반복적인 데이터베이스 작업에 매우 유용합니다.

에픽

SQL Server를 사용하여 EC2 Linux 인스턴스 준비

작업	설명	필요한 기술
Linux 운영 체제를 제공하고 Microsoft SQL Server를 포함하는 AMI를 선택합니다.		Sysadmin
EC2 인스턴스를 생성하도록 AMI를 구성합니다.		Sysadmin
보안 그룹에 대한 인바운드 및 아웃바운드 규칙을 생성합니다.		Sysadmin
Microsoft SQL Server 데이터베이스에 대한 Linux EC2 인스턴스를 구성합니다.		DBA
소스 데이터베이스에서와 같이 사용자를 생성하고 권한을 제공합니다.		앱 소유자, DBA
SQL Server 도구와 sqlcmd 유틸리티를 Linux EC2 인스턴스에 설치합니다.		DBA

데이터베이스를 백업하고 백업 파일을 Linux EC2 인스턴스로 이동

작업	설명	필요한 기술
온프레미스 SQL 데이터베이스를 백업합니다.		DBA
Microsoft SQL Server에 WinSCP를 설치합니다.		DBA
Microsoft SQL Server가 실행되는 Linux EC2 인스턴스로 백업 파일을 이동합니다.		DBA

SQL Server가 실행되는 Linux EC2 인스턴스에 데이터베이스 복원

작업	설명	필요한 기술
sqlcmd 유틸리티를 사용하여 데이터베이스 백업 파일에서 데이터베이스를 복원합니다.		DBA
데이터베이스 객체 및 데이터를 검증합니다.		개발자, 테스트 엔지니어

Windows SQL Server에서 Linux EC2 인스턴스의 Windows SQL Server로 전환

작업	설명	필요한 기술
데이터베이스 객체 및 데이터를 검증합니다.		개발자, 테스트 엔지니어
온프레미스 Microsoft SQL Server 데이터베이스에서 Microsoft SQL Server가 실행되는 Linux EC2로 전환합니다.		DBA

관련 리소스

- [Amazon Linux 2 및 Ubuntu AMI에서 SQL Server 2017을 구성하는 방법](#)
- [Linux 인스턴스에 SQL 도구 설치](#)
- [온프레미스 Microsoft SQL Server 데이터베이스에서 Linux EC2 인스턴스의 Microsoft SQL Server로 백업 및 복원](#)

연결된 서버를 사용하여 온프레미스 Microsoft SQL Server 데이터베이스를 Amazon RDS for SQL Server로 마이그레이션

작성자: Kevin Yung(AWS)

R 유형: 리플랫폼	소스: 데이터베이스: 관계형	대상: Amazon RDS for Microsoft SQL Server
작성자: AWS	환경: 프로덕션	기술: 데이터베이스, 마이그레이션
워크로드: Microsoft	AWS 서비스: Amazon RDS	

요약

연결된 서버를 사용하면 Microsoft SQL Server에서 데이터베이스 서버의 다른 인스턴스에서 SQL 문을 실행할 수 있습니다. 이 패턴은 비용을 절감하고 가용성을 높이기 위해 온프레미스 Microsoft SQL Server 데이터베이스를 Amazon Relational Database Service(RDS)로 마이그레이션하는 방법을 설명합니다. 현재 Amazon RDS for Microsoft SQL Server는 Amazon Virtual Private Cloud(VPC) 네트워크 외부 연결을 지원하지 않습니다.

이 패턴을 사용하여 다음과 같은 목표를 달성할 수 있습니다.

- 연결된 서버 기능을 손상시키지 않고 Microsoft SQL Server를 Amazon RDS for Microsoft SQL Server로 마이그레이션할 수 있습니다.
- 서로 다른 방식으로 연결된 Microsoft SQL Server의 우선 순위를 지정하고 마이그레이션합니다.

사전 조건 및 제한 사항

사전 조건

- [Amazon RDS 기반 Microsoft SQL Server](#)가 필요한 기능을 지원하는지 확인하세요.
- [기본 데이터 정렬 또는 데이터베이스 수준에서 설정된 데이터 정렬과 함께 Amazon RDS for Microsoft SQL Server](#)를 사용할 수 있는지 확인하세요.

아키텍처

소스 기술 스택

- 온프레미스 데이터베이스(Microsoft SQL 서버)

대상 기술 스택

- Amazon RDS for SQL Server

소스 상태 아키텍처

대상 상태 아키텍처

대상 상태에서는 연결된 서버를 사용하여 Microsoft SQL Server를 Amazon RDS for Microsoft SQL Server로 마이그레이션합니다. 이 아키텍처는 Network Load Balancer를 사용하여 Amazon RDS for Microsoft SQL Server에서 Microsoft SQL Server를 실행하는 온프레미스 서버로 트래픽을 프록시합니다. 다음 다이어그램은 Network Load Balancer의 역방향 프록시 기능을 보여줍니다.

도구

- AWS CloudFormation
- Network Load Balancer
- 다중 가용 영역(Multi-AZs)의 Amazon RDS for SQL Server
- AWS Database Migration Service(AWS DMS)

에픽

랜딩 존 VPC 생성

작업	설명	필요한 기술
CIDR 할당을 생성합니다.		AWS SysAdmin
Virtual Private Cloud(VPC)를 생성합니다.		AWS SysAdmin
VPC 서브넷을 생성합니다.		AWS SysAdmin
서브넷 액세스 제어 목록(ACL)을 생성합니다.		AWS SysAdmin
서브넷 라우팅 테이블을 생성합니다.		AWS SysAdmin
AWS Direct Connect 또는 AWS 가상 프라이빗 네트워크 (VPN)를 사용하여 연결을 생성합니다.		AWS SysAdmin

데이터베이스를 Amazon RDS로 마이그레이션

작업	설명	필요한 기술
Amazon RDS for Microsoft SQL Server DB 인스턴스를 생성합니다.		AWS SysAdmin
AWS DMS 복제 인스턴스를 생성합니다.		AWS SysAdmin
AWS DMS의 소스와 대상 데이터베이스 엔드포인트를 생성합니다.		AWS SysAdmin

작업	설명	필요한 기술
마이그레이션 작업을 생성하고 전체 로드 후 연속 복제를 ON으로 설정합니다.		AWS SysAdmin
Amazon RDS for Microsoft SQL Server가 온프레미스 SQL Server 데이터베이스에 액세스할 수 있도록 방화벽 변경을 요청합니다.		AWS SysAdmin
Network Load Balancer를 생성합니다.		AWS SysAdmin
데이터 센터의 데이터베이스 서버를 대상으로 하는 대상 그룹을 생성합니다.	대상 설정에서 호스트 이름을 사용하여 데이터 센터(DC) 장애 조치 이벤트를 통합하는 것이 좋습니다.	AWS SysAdmin
연결된 서버 설정을 위한 SQL 문을 실행합니다.	Amazon RDS for Microsoft SQL Server DB 인스턴스에 대해 Microsoft SQL 관리 도구를 사용하여 연결된 서버를 추가하기 위한 SQL 문을 실행합니다. SQL 문에서 Network Load Balancer 호스트 이름을 사용하도록 @datasrc를 설정합니다. Amazon RDS for Microsoft SQL Server DB 인스턴스에 대해 Microsoft SQL 관리 도구를 사용하여 연결된 서버 로그인 보안 인증 정보를 추가합니다.	AWS SysAdmin
SQL Server 함수를 테스트하고 검증합니다.		AWS SysAdmin
전환을 생성합니다.		AWS SysAdmin

관련 리소스

- [Amazon RDS에서 Microsoft SQL Server에 대한 공통 관리 작업](#)
- [Microsoft SQL Server의 데이터 정렬 및 문자 집합](#)
- [Network Load Balancer 설명서](#)
- [Amazon RDS for Microsoft SQL Server에 연결 서버 구현\(블로그 게시물\)](#)

기본 백업 및 복원 수단을 사용하여 온프레미스 Microsoft SQL Server 데이터베이스를 Amazon RDS for SQL Server로 마이그레이션

작성자: 티루말라 다사리(AWS), 데이비드 케이로즈(AWS) 및 비샬 싱(AWS)

환경: PoC 또는 파일럿	소스: 온프레미스 SQL Server 데이터베이스	대상: Amazon RDS for SQL Server
R 유형: 리플랫폼	워크로드: Microsoft	기술: 마이그레이션, 데이터베이스, 운영 체제
AWS 서비스: Amazon RDS, Amazon S3		

요약

이 패턴은 온프레미스 Microsoft SQL Server 데이터베이스를 SQL Server DB 인스턴스용 Amazon Relational Database Service(Amazon RDS)로 마이그레이션하는 방법(동종 마이그레이션)을 설명합니다. 마이그레이션 프로세스는 기본 SQL Server 백업 및 복원 수단을 기반으로 합니다. SQL Server Management Studio(SSMS)를 사용하여 데이터베이스 백업 파일을 생성하고, Amazon Simple Storage Service(S3) 버킷을 사용하여 백업 파일을 저장한 다음 Amazon RDS for SQL Server에 복원합니다.

사전 조건 및 제한 사항

사전 조건

- 활성 상태의 AWS 계정
- S3 버킷과 Amazon RDS for SQL Server DB 인스턴스에 액세스하기 위한 AWS Identity and Access Management(IAM) 역할 정책입니다.

제한 사항

- 이 패턴에 설명된 프로세스는 데이터베이스만 마이그레이션합니다. 추가 단계가 필요한 SQL Server 에이전트 작업을 비롯한 SQL 로그인 또는 데이터베이스 사용자는 마이그레이션되지 않습니다.

제품 버전

- SQL Server 2012-2017. 지원되는 버전 및 기능의 최신 목록은 AWS 설명서에서 [Amazon RDS의 Microsoft SQL Server](#)를 참조하세요.

아키텍처

소스 기술 스택

- 온프레미스 Microsoft SQL Server 데이터베이스

대상 기술 스택

- Amazon RDS for SQL Server DB 인스턴스

데이터 마이그레이션 아키텍처

도구

- Microsoft SQL Server Management Studio(SSMS)는 SQL Server 인프라를 관리하기 위한 통합 환경입니다. SQL Server와 상호 작용하는 다양한 스크립트 편집기와 함께 사용자 인터페이스와 도구 그룹을 제공합니다.

에픽

Amazon RDS for SQL Server DB 인스턴스 생성

작업	설명	필요한 기술
Amazon RDS for SQL Server에서 SQL Server를 데이터베이스 엔진으로 선택합니다.		DBA
SQL Server Express Edition을 선택합니다.		DBA

작업	설명	필요한 기술
데이터베이스 세부 정보를 지정합니다.	DB 생성에 관한 자세한 내용은 Amazon RDS 설명서 를 참조하세요.	DBA, 앱 소유자

온프레미스 SQL Server 데이터베이스에서 백업 파일 생성

작업	설명	필요한 기술
SSMS를 통해 온프레미스 SQL Server 데이터베이스에 연결합니다.		DBA
데이터베이스의 백업을 생성합니다.	자세한 지침은 SSMS 설명서 를 참조하세요.	DBA, 앱 소유자

Amazon S3로 백업 파일을 업로드합니다.

작업	설명	필요한 기술
Amazon S3에서 버킷을 생성합니다.	자세한 내용은 Amazon S3 설명서 를 참조하십시오.	DBA
백업 파일을 S3 버킷에 업로드합니다.	자세한 내용은 Amazon S3 설명서 를 참조하십시오.	SysOps 관리자

Amazon RDS for SQL Server에 데이터베이스 복원

작업	설명	필요한 기술
Amazon RDS에 옵션 그룹을 추가합니다.	1. https://console.aws.amazon.com/rds/ 에서 Amazon RDS 콘솔을 엽니다.	SysOps 관리자

작업	설명	필요한 기술
	<ol style="list-style-type: none"> 2. 탐색 창에서 옵션 그룹, 그룹 생성을 선택합니다. 3. 옵션 그룹에 대한 정보 작성을 완료한 다음, 생성을 선택합니다. 4. SQLSERVER_BACKUP_RESTORE 옵션을 옵션 그룹에 추가한 다음, 옵션 추가를 선택합니다. <p>자세한 내용은 Amazon RDS 설명서를 참조하세요.</p>	
<p>데이터베이스를 복원합니다.</p>	<ol style="list-style-type: none"> 1. SSMS를 통해 Amazon RDS for SQL Server에 연결합니다. 2. 데이터베이스를 복원하려면 <code>msdb.dbo.rds_restore_database</code> 저장 프로시저를 호출합니다. 	<p>DBA</p>

대상 데이터베이스 검증

작업	설명	필요한 기술
<p>객체 및 데이터를 검증합니다.</p>	<p>소스 데이터베이스와 Amazon RDS for SQL Server 간의 객체와 데이터를 검증합니다.</p> <p>참고: 이 작업은 데이터베이스만 마이그레이션합니다. 로그인과 작업은 마이그레이션되지 않습니다.</p>	<p>앱 소유자, DBA</p>

전환

작업	설명	필요한 기술
애플리케이션 트래픽을 리디렉션합니다.	검증이 끝나면 애플리케이션 트래픽을 Amazon RDS for SQL Server DB 인스턴스로 리디렉션합니다.	앱 소유자, DBA

관련 리소스

- [Amazon S3 설명서](#)
- [Amazon RDS for SQL Server 설명서](#)
- [Microsoft SQL Server 데이터베이스 엔진의 옵션](#)

AWS DMS와 AWS SCT를 사용하여 Microsoft SQL Server 데이터베이스를 Aurora MySQL로 마이그레이션

작성자: Mark Szalkiewicz(AWS)

R 유형: 리플랫폼	소스: 데이터베이스: 관계형	대상: Amazon Aurora MySQL
작성자: AWS	환경: PoC 또는 파일럿	기술: 데이터베이스, 마이그레이션
워크로드: Microsoft	AWS 서비스: Amazon Aurora	

요약

이 패턴은 온프레미스 또는 Amazon Elastic Compute Cloud(Amazon EC2) 인스턴스의 Microsoft SQL Server 데이터베이스를 Amazon Aurora MySQL로 마이그레이션하는 방법을 설명합니다. 이 패턴은 데이터 마이그레이션 및 스키마 변환을 위해 AWS Database Migration Service(AWS DMS) 및 AWS Schema Conversion Tool(AWS SCT)을 사용합니다.

사전 조건 및 제한 사항

사전 조건

- 활성 상태의 AWS 계정.
- 온프레미스 데이터 센터 또는 EC2 인스턴스에 있는 Microsoft SQL Server 소스 데이터베이스
- AWS SCT가 설치된 로컬 시스템 또는 EC2 인스턴스에 설치된 AWS SCT 커넥터용 자바 데이터베이스 연결(JDBC) 드라이버

제한 사항

- 데이터베이스 크기 제한: 64TB

제품 버전

- Enterprise, Standard, Workgroup 및 Developer 버전용 Microsoft SQL Server 버전 2008, 2008R2, 2012, 2014, 2016, 2017. Web 및 Express 버전은 AWS DMS에서 지원하지 않습니다. 지원되는 버

전의 최신 목록은 [Microsoft SQL Server 데이터베이스를 AWS DMS용 소스로 사용하기](#) 섹션을 참조하세요. 가장 종합적인 버전 및 기능 지원을 위해 최신 버전의 AWS DMS를 사용하는 것을 권장합니다. AWS SCT에서 지원하는 Microsoft SQL Server 버전에 대한 자세한 내용은 [AWS SCT 설명서](#)를 참조하세요.

- MySQL 버전 5.5, 5.6 및 5.7. 지원되는 버전의 최신 목록은 [MySQL 호환 데이터베이스를 AWS DMS용 타겟으로 사용](#)을 참조하세요.

아키텍처

소스 기술 스택

다음 중 하나입니다.

- 온프레미스 Microsoft SQL Server 데이터베이스
- EC2 인스턴스의 Microsoft SQL Server 데이터베이스

대상 기술 스택

- Aurora MySQL

데이터 마이그레이션 아키텍처

- AWS 클라우드에서 실행되는 Microsoft SQL Server 데이터베이스에서
- 온프레미스 데이터 센터에서 실행되는 Microsoft SQL Server 데이터베이스에서

도구

- AWS DMS - [AWS Data Migration Service](#)(AWS DMS)를 사용하면 Oracle, SQL Server, MySQL 및 PostgreSQL을 비롯하여 널리 사용되는 상용 및 오픈 소스 데이터베이스에서 데이터를 마이그레이션할 수 있습니다. AWS DMS를 사용하여 데이터를 AWS 클라우드로, 온프레미스 인스턴스 간(AWS 클라우드 설정을 통해) 또는 클라우드와 온프레미스 설정 조합 간에 마이그레이션할 수 있습니다.

- AWS SCT - [AWS Schema Conversion Tool](#)(AWS SCT)을 사용하면 소스 데이터베이스 스키마와 대부분의 사용자 지정 코드를 대상 데이터베이스와 호환되는 형식으로 자동 변환하여 이기종 데이터베이스를 쉽게 마이그레이션할 수 있습니다.

에픽

마이그레이션 준비

작업	설명	필요한 기술
소스 및 대상 데이터베이스 버전과 엔진의 유효성을 확인합니다.		DBA
소스 및 대상 데이터베이스를 위한 아웃바운드 보안 그룹을 생성합니다.		SysAdmin
필요한 경우 AWS SCT용 EC2 인스턴스를 생성하고 구성합니다.		DBA
최신 버전의 AWS SCT 및 관련 드라이버를 다운로드합니다.		DBA
원본 데이터베이스에 필수 사용자 및 권한을 추가하고 검증합니다.		DBA
워크로드용 AWS SCT 프로젝트를 생성하고 소스 데이터베이스에 연결합니다.		DBA
평가 보고서를 작성하고 타당성을 평가하세요.		DBA

대상 데이터베이스를 준비합니다.

작업	설명	필요한 기술
Amazon Aurora를 데이터베이스 엔진으로 사용하여 대상 Amazon RDS DB 인스턴스를 생성합니다.		DBA
소스에서 사용자, 역할 및 권한 목록을 추출합니다.		DBA
기존 데이터베이스 사용자를 새 데이터베이스 사용자에게 매핑합니다.		앱 소유자
대상 데이터베이스에서 사용자를 생성합니다.		DBA
대상 데이터베이스에 이전 단계의 역할을 적용합니다.		DBA
원본 데이터베이스의 데이터베이스 옵션, 파라미터, 네트워크 파일 및 데이터베이스 링크를 검토한 다음 대상 데이터베이스에 적용할 수 있는지 평가하세요.		DBA
모든 관련 설정을 대상에 적용합니다.		DBA

객체 전송

작업	설명	필요한 기술
대상 데이터베이스에 대한 AWS SCT 연결을 구성합니다.		DBA

작업	설명	필요한 기술
AWS SCT를 사용하여 스키마를 변환합니다.	AWS SCT는 소스 데이터베이스 스키마와 대부분의 사용자 지정 코드를 대상 데이터베이스와 호환되는 형식으로 자동 변환합니다. 도구가 자동으로 변환할 수 없는 코드는 명확하게 표시되므로 직접 변환할 수 있습니다.	DBA
생성된 SQL 보고서를 검토하고 모든 오류 및 경고를 저장합니다.		DBA
자동화된 스키마 변경을 대상에 적용하거나.sql 파일로 저장합니다.		DBA
AWS SCT가 대상에 객체를 생성했는지 확인합니다.		DBA
자동 변환에 실패한 모든 항목을 수동으로 재작성, 거부 또는 재설계하세요.		DBA
생성된 역할 및 사용자 부여를 적용하고 예외를 검토하세요.		DBA

데이터 마이그레이션

작업	설명	필요한 기술
마이그레이션 방법을 결정하십시오.		DBA

작업	설명	필요한 기술
AWS 콘솔을 사용하여 복제 인스턴스를 생성하려면	AWS DMS 사용에 대한 자세한 내용은 '관련 리소스' 섹션의 링크를 참조하세요.	DBA
원본 및 대상 DB 엔드포인트		DBA
복제 작업을 생성합니다.		DBA
복제 작업을 시작하고 로그를 모니터링합니다.		DBA

애플리케이션 마이그레이션

작업	설명	필요한 기술
AWS SCT를 사용하여 애플리케이션 코드 내의 SQL 항목을 분석하고 변환합니다.	한 엔진에서 다른 엔진으로 데이터베이스 스키마를 변환할 때는 이전 데이터베이스 엔진 대신 새 데이터베이스 엔진과 상호 작용하도록 애플리케이션의 SQL 코드도 업데이트해야 합니다. 변환된 SQL 코드를 보고, 분석하고, 편집하고, 저장할 수 있습니다. AWS SCT 사용에 대한 자세한 내용은 '관련 리소스' 섹션의 링크를 참조하세요.	앱 소유자
AWS에 새 애플리케이션 서버를 생성합니다.		앱 소유자
애플리케이션 코드를 새 서버로 마이그레이션합니다.		앱 소유자

작업	설명	필요한 기술
대상 데이터베이스 및 드라이버에 맞게 애플리케이션 서버를 구성합니다.		앱 소유자
애플리케이션의 소스 데이터베이스 엔진 관련 코드를 모두 수정하세요.		앱 소유자
대상 엔진에 맞게 애플리케이션 코드를 최적화하세요.		앱 소유자

전환

작업	설명	필요한 기술
모든 신규 사용자, 권한 및 코드 변경 사항을 대상에 적용하세요.		DBA
변경 사항이 있을 경우 애플리케이션을 잠급니다.		앱 소유자
모든 변경 사항이 대상 데이터베이스에 전파되었는지 확인합니다.		DBA
새 애플리케이션 서버가 대상 데이터베이스를 가리키도록 합니다.		앱 소유자
모든 것을 다시 확인해 보세요.		앱 소유자
가동을 시작합니다.		앱 소유자

프로젝트 닫기

작업	설명	필요한 기술
임시 AWS 리소스(AWS DMS 복제 인스턴스 및 AWS SCT에 사용되는 EC2 인스턴스)를 종료합니다.		DBA, 앱 소유자
내부 팀을 위한 AWS DMS 프로세스에 대한 피드백을 업데이트합니다.		DBA, 앱 소유자
AWS DMS 프로세스를 수정하고 필요한 경우 템플릿을 개선합니다.		DBA, 앱 소유자
프로젝트 문서를 검토하고 검증하세요.		DBA, 앱 소유자
마이그레이션 시간, 수동 대비 도구 비용 절감 비율 등에 대한 지표를 수집하세요.		DBA, 앱 소유자
프로젝트를 종료하고 피드백을 제공하세요.		DBA, 앱 소유자

관련 리소스

참조

- [AWS DMS 사용 설명서](#)
- [AWS SCT 사용 설명서](#)
- [Amazon Aurora 요금](#)

자습서 및 동영상

- [AWS Database Migration Service 시작하기](#)

- [AWS Schema Conversion Tool 사용 시작하기](#)
- [Amazon RDS 리소스](#)
- [AWS DMS 단계별 안내](#)

온프레미스 MariaDB 데이터베이스를 기본 도구를 사용하여 Amazon RDS for MariaDB로 마이그레이션

작성자: Sergey Dmitriev(AWS)

환경: PoC 또는 파일럿	소스: 데이터베이스: 관계형	대상: Amazon RDS for MariaDB
R 유형: 리플랫폼	워크로드: 오픈 소스	기술: 마이그레이션, 데이터베이스

요약

이 패턴은 기본 도구를 사용하여 온프레미스 MariaDB 데이터베이스를 Amazon Relational Database Service(RDS)로 마이그레이션하기 위한 지침을 제공합니다. MySQL 도구가 설치되어 있으면 mysq 및 mysqldump를 사용할 수 있습니다. MariaDB 도구가 설치되어 있으면 mariadb 및 mariadb-dump를 사용할 수 있습니다. MySQL과 MariaDB 도구는 출처가 동일하지만 MariaDB 버전 10.6 이상에서는 약간의 차이가 있습니다.

사전 조건 및 제한 사항

사전 조건

- 활성 상태의 AWS 계정
- 온프레미스 데이터 센터의 MariaDB 소스 데이터베이스

제한 사항

- 데이터베이스 크기 제한: 64TB

제품 버전

- MariaDB 버전 10.0-10.6(지원되는 버전의 최신 목록은 AWS 설명서의 [MariaDB on Amazon RDS](#) 참조)

아키텍처

소스 기술 스택

- 온프레미스 데이터 센터의 MariaDB 데이터베이스

대상 기술 스택

- Amazon RDS for MariaDB DB instance

대상 아키텍처

데이터 마이그레이션 아키텍처

도구

- 도구 MySQL 도구: mysql 및 mysqldump
- 기본 MariaDB 도구: mariadb 및 mariadb-dump

에픽

마이그레이션 계획

작업	설명	필요한 기술
소스 및 대상 데이터베이스 버전과 엔진을 확인합니다.		DBA

작업	설명	필요한 기술
대상 서버 인스턴스의 하드웨어 요구 사항을 확인합니다.		DBA, 시스템 관리자
스토리지 요구 사항(스토리지 유형 및 용량)을 확인합니다.		DBA, 시스템 관리자
용량, 스토리지 기능, 네트워크 기능에 따라 적절한 인스턴스 유형을 선택합니다.		DBA, 시스템 관리자
소스 및 대상 데이터베이스의 네트워크 액세스 보안 요구 사항을 확인합니다.		DBA, 시스템 관리자
애플리케이션 마이그레이션 전략을 파악합니다.		DBA, 앱 소유자, 시스템 관리자

인프라 구성

작업	설명	필요한 기술
Virtual Private Cloud(VPC)를 생성합니다.		시스템 관리자
보안 그룹을 생성합니다.		시스템 관리자
MariaDB를 실행하는 Amazon RDS DB 인스턴스를 구성하고 시작합니다.		시스템 관리자

데이터 마이그레이션

작업	설명	필요한 기술
기본 도구를 사용하여 데이터베이스 객체 및 데이터를 마이그레이션합니다.	소스 데이터베이스에서 mysqldump 또는 mariadb-dump를 사용하여 데이터베이스 객체 및 데이터가 포함된 출력 파일을 생성합니다. 대상 데이터베이스에서 mysql 또는 mariadb를 사용하여 데이터를 복원합니다.	DBA
데이터를 검증합니다.	소스 및 대상 데이터베이스를 검사하여 데이터 마이그레이션이 성공했는지 확인합니다.	DBA

애플리케이션 마이그레이션

작업	설명	필요한 기술
애플리케이션 마이그레이션 전략을 따릅니다.		DBA, 앱 소유자, 시스템 관리자

전환

작업	설명	필요한 기술
애플리케이션 클라이언트를 새 인프라로 전환합니다.		DBA, 앱 소유자, 시스템 관리자

프로젝트 닫기

작업	설명	필요한 기술
임시 AWS 리소스를 종료합니다.		시스템 관리자
프로젝트 문서를 검토하고 검증하세요.		DBA, 앱 소유자, 시스템 관리자
마이그레이션 시간, 도구를 통한 비용 절감 등에 대한 지표를 수집하세요.		DBA, 앱 소유자, 시스템 관리자
프로젝트를 마무리하고 피드백을 제공하세요.		DBA, 앱 소유자, 시스템 관리자

관련 리소스

Amazon RDS 참조

- [Amazon RDS for MariaDB](#)
- [Amazon Virtual Private Cloud VPCs 및 Amazon RDS](#)
- [Amazon RDS 다중 AZ 배포](#)
- [Amazon RDS 요금](#)

MySQL 및 MariaDB 참조

- [mariadb-dump/mysqldump](#)
- [mysql 명령줄 클라이언트](#)

자습서 및 동영상

- [Amazon RDS 시작](#)

온프레미스 MySQL 데이터베이스를 Aurora MySQL로 마이그레이션

작성자: 비노드 쿠마르 사두 (AWS) 와 이고르 오브라도비치 (AWS)

환경: 프로덕션	소스: 온프레미스 MySQL 데이터베이스	대상: Amazon Aurora MySQL-Compatible Edition
R 유형: 리플랫폼	워크로드: 오픈 소스	기술: 마이그레이션, 데이터베이스
AWS 서비스: AWS DMS		

요약

이 패턴은 온프레미스 MySQL 원본 데이터베이스를 Amazon Aurora MySQL 호환 에디션으로 마이그레이션하는 방법을 설명합니다. 마이그레이션을 위한 두 가지 옵션, 즉 AWS Database Migration Service (AWS DMS) 를 사용하거나 mysqldbcopy 및 mysqldump와 같은 기본 MySQL 도구를 사용하는 방법에 대해 설명합니다.

사전 조건 및 제한 사항

사전 조건

- 활성 상태의 AWS 계정
- 온프레미스 데이터 센터의 MySQL 소스 데이터베이스

제한 사항

- 데이터베이스 크기 제한: 64TB

제품 버전

- MySQL 버전 5.7과 8.0입니다. 지원되는 버전의 최신 목록은 설명서에서 [Amazon Aurora 버전을 참조하십시오](#). AWS 를 사용하는 AWS DMS 경우 에서 지원하는 [MySQL 버전의 AWS DMS MySQL 호환 데이터베이스를 대상으로 사용을 참조하십시오](#). AWS DMS

아키텍처

소스 기술 스택

- 온프레미스 MySQL 데이터베이스

대상 기술 스택

- Amazon Aurora MySQL 호환 버전

대상 아키텍처

데이터 마이그레이션 아키텍처

사용: AWS DMS

네이티브 MySQL 도구 사용:

도구

- [AWS Database Migration Service\(AWS DMS\)](#) 는 여러 소스 및 대상 데이터베이스를 지원합니다. 에서 지원하는 MySQL 원본 및 대상 데이터베이스에 대한 자세한 내용은 [MySQL AWS DMS 호환 데이터베이스 마이그레이션](#)을 참조하십시오. AWS 가장 포괄적인 버전 및 기능 지원을 받으려면 최신 버전의 AWS DMS 을 사용하는 것이 좋습니다.
- [mysqldbcopy](#)는 MySQL 데이터베이스를 단일 서버 또는 서버 간에 복사하는 MySQL 유틸리티입니다.
- [mysqldump](#)는 MySQL 데이터베이스에서 백업 또는 마이그레이션 목적으로 [덤프](#) 파일을 생성하는 MySQL 유틸리티입니다.

에픽

마이그레이션 계획

작업	설명	필요한 기술
소스 및 대상 데이터베이스 버전과 엔진의 유효성을 확인합니다.		DBA
대상 서버 인스턴스의 하드웨어 요구 사항을 확인합니다.		DBA, 시스템 관리자
스토리지 요구 사항(스토리지 유형 및 용량)을 확인합니다.		DBA, 시스템 관리자
용량, 스토리지 기능, 네트워크 기능에 따라 적절한 인스턴스 유형을 선택합니다.		DBA, 시스템 관리자
소스 및 대상 데이터베이스의 네트워크 액세스 보안 요구 사항을 확인합니다.		DBA, 시스템 관리자
애플리케이션 마이그레이션 전략을 파악합니다.		DBA, 앱 소유자, 시스템 관리자

인프라 구성

작업	설명	필요한 기술
Virtual Private Cloud(VPC)를 생성합니다.		시스템 관리자
보안 그룹을 생성합니다.		시스템 관리자
Aurora MySQL 호환 DB 클러스터를 구성하고 시작합니다.		시스템 관리자

데이터 마이그레이션 - 옵션 1

작업	설명	필요한 기술
네이티브 MySQL 도구 또는 타사 도구를 사용하여 데이터베이스 객체 및 데이터를 마이그레이션합니다.	자세한 지침은 mysqldbcopy 및 mysqldump와 같은 MySQL 도구에 대한 설명서를 참조하십시오.	DBA

데이터 마이그레이션 - 옵션 2

작업	설명	필요한 기술
AWS DMS를 사용하여 데이터를 마이그레이션하십시오.	지침은 설명서에서 MySQL 호환 데이터베이스를 원본으로 사용 및 MySQL 호환 데이터베이스를 대상으로 사용을 참조하십시오. AWS DMS	DBA

애플리케이션 마이그레이션

작업	설명	필요한 기술
애플리케이션 마이그레이션 전략을 따릅니다.		DBA, 앱 소유자, 시스템 관리자

전환

작업	설명	필요한 기술
애플리케이션 클라이언트를 새 인프라로 전환합니다.		DBA, 앱 소유자, 시스템 관리자

프로젝트 닫기

작업	설명	필요한 기술
임시 AWS 리소스를 종료합니다.		DBA, 시스템 관리자
프로젝트 문서를 검토하고 검증하세요.		DBA, 앱 소유자, 시스템 관리자
마이그레이션 시간, 수동 대비 도구 비율(%), 비용 절감 등에 대한 지표를 수집합니다.		DBA, 앱 소유자, 시스템 관리자
프로젝트를 마무리하고 피드백을 제공하세요.		

관련 리소스

참조

- [데이터베이스를 Amazon Aurora로 마이그레이션하기](#)
- [AWS DMS 웹사이트](#)
- [DMS 설명서](#)
- [Amazon Aurora 요금 책정](#)
- [Aurora MySQL DB 클러스터 생성 및 연결](#)
- [Amazon Virtual Private Cloud VPC 및 Amazon RDS](#)
- [Amazon Aurora 설명서](#)

자습서 및 동영상

- [DMS 시작하기](#)
- [Amazon Aurora 시작하기](#)

페르코나, Amazon EFS 및 Amazon S3를 사용하여 온프레미스 MySQL 데이터베이스를 Aurora MySQL로 XtraBackup 마이그레이션합니다.

작성자: Rohan Jamadagni(AWS), sajith menon(AWS), 및 Udayasimha Theepireddy(AWS)

소스: 온프레미스	대상: Aurora MySQL	R 유형: 리플랫폼
환경: 프로덕션	기술: 데이터베이스, 마이그레이션	워크로드: 오픈 소스

AWS 서비스: Amazon S3,
Amazon Aurora, Amazon EFS

요약

이 패턴은 Percona를 사용하여 대규모 온프레미스 MySQL 데이터베이스를 Amazon Aurora MySQL로 효율적으로 마이그레이션하는 방법을 설명합니다. XtraBackup Percona는 MySQL 기반 서버를 위한 오픈 소스, 비차단 백업 유틸리티입니다. 이 패턴은 Amazon Elastic File System(Amazon EFS)을 사용하여 백업을 Amazon Simple Storage Service(Amazon S3)에 업로드하는 시간을 줄이고 백업을 Amazon Aurora MySQL로 복원하는 방법을 보여줍니다. 또한 이 패턴은 대상 Aurora MySQL 데이터베이스에 적용할 바이너리 로그의 수를 최소화하기 위해 증분 Percona 백업을 만드는 방법에 대한 세부 정보를 제공합니다.

사전 조건 및 제한 사항

사전 조건

- 활성 상태의 AWS 계정
- AWS Identity and Access Management(IAM) 역할 및 정책을 만들 권한
- 온프레미스 MySQL 데이터베이스와 AWS의 Virtual Private Cloud(VPC) 간 네트워크 연결

제한 사항

- 소스 서버는 Network File System(NFS) 클라이언트(nfs-utils/nfs-common)를 설치할 수 있는 Linux 기반 시스템이어야 합니다.
- 백업 파일 업로드에 사용되는 S3 버킷은 서버 측 암호화(SSE-S3/SSE-KMS)만 지원합니다.

- Amazon S3는 백업 파일 크기를 5TB로 제한합니다. 백업 파일이 5TB를 초과하면 파일을 여러 개의 작은 크기의 파일로 나눌 수 있습니다.
- S3 버킷에 업로드되는 소스 파일의 수는 100만 개를 초과할 수 없습니다.
- 이 패턴은 XtraBackup Percona 전체 백업 및 증분 백업만 지원합니다. --tables, --tables-exclude, --tables-file, --databases, --databases-exclude, 또는 --databases-file을 사용하는 부분 백업을 지원하지 않습니다.
- Aurora는 소스 MySQL 데이터베이스에서 사용자, 함수, 저장 프로시저 또는 시간대 정보를 복원하지 않습니다.

제품 버전

- 소스 데이터베이스는 MySQL 버전 5.5, 5.6, 5.7이어야 합니다.
- MySQL 5.7의 경우 페르코나 2.4를 사용해야 합니다. XtraBackup
- MySQL 5.6 및 5.6의 경우 XtraBackup 페르코나 2.3 또는 2.4를 사용해야 합니다.

아키텍처

소스 기술 스택

- Linux 기반 운영 체제
- MySQL 서버
- 페르코나 XtraBackup

대상 기술 스택

- Amazon Aurora
- Amazon S3
- Amazon EFS

대상 아키텍처

도구

서비스

- [Amazon Aurora](#)는 MySQL 배포를 간편하고 비용 효율적으로 설정, 운영 및 확장할 수 있게 해 준 완전 관리형 관계형 데이터베이스입니다. Aurora MySQL은 MySQL의 드롭인 대체품입니다.
- [Amazon Elastic File System\(Amazon EFS\)](#)은 AWS 클라우드에서 공유 파일 시스템을 생성하고 구성하는 데 도움이 됩니다.
- [Amazon Simple Storage Service\(Amazon S3\)](#)는 원하는 양의 데이터를 저장, 보호 및 검색하는 데 도움이 되는 클라우드 기반 객체 스토리지 서비스입니다.

기타 도구

- [XtraBackupPercona](#)는 데이터베이스를 방해하거나 차단하지 않고 MySQL 데이터베이스의 스트리밍, 압축 및 증분 백업을 수행하는 오픈 소스 유틸리티입니다.

에픽

Amazon EFS 파일 시스템 생성

작업	설명	필요한 기술
Amazon EFS 탑재 대상과 연결할 보안 그룹을 생성합니다.	AWS Transit Gateway를 통해 온프레미스 데이터베이스에 대한 VPN 연결로 구성된 VPC에 보안 그룹을 생성합니다. 이 스토리와 다른 스토리에 설명된 명령 및 단계에 대한 자세한 내용은 이 패턴 끝에 있는 "관련 리소스" 섹션의 링크를 참조하세요.	AWS DevOps /데이터베이스 관리자
보안 그룹 규칙을 편집합니다.	유형 NFS, 포트 2049, 온프레미스 데이터베이스 서버의 IP 범위를 소스로 사용하여 인바운드 규칙을 추가합니다. 기본적으로 이 아웃바운드 규칙은 모든 트래픽이 나가도록 허용합니다. 그렇지 않은 경우 아웃바운드 규칙을 추가하여 NFS	AWS DevOps /데이터베이스 관리자

작업	설명	필요한 기술
	포트에 대한 연결을 엽니다. 포트 2049\ (소스: 동일 보안 그룹의 보안 그룹 ID)와 포트 22(소스: EC2 인스턴스에 연결할 IP 범위)라는 두 개의 인바운드 규칙을 더 추가합니다.	
파일 시스템을 만듭니다.	탑재 대상에서는 이전 스토리에서 만든 VPC와 보안 그룹을 사용합니다. 온프레미스 데이터베이스의 I/O 요구 사항에 따라 처리량 모드와 성능을 선택합니다. 필요에 따라 저장되어 있는 데이터 암호화를 활성화합니다.	AWS DevOps /데이터베이스 관리자

파일 시스템을 탑재

작업	설명	필요한 기술
EC2 인스턴스와 연결할 IAM 인스턴스 프로파일 역할을 생성합니다.	Amazon S3에서 객체를 업로드하고 액세스할 권한이 있는 IAM 역할을 생성합니다. 백업이 정책 리소스로 저장되는 S3 버킷을 선택합니다.	AWS DevOps
EC2 인스턴스 생성합니다.	Linux 기반 EC2 인스턴스를 시작하고 이전 단계에서 생성한 IAM 인스턴스 프로파일 역할과 앞서 생성한 보안 그룹을 연결합니다.	AWS DevOps
NFS 클라이언트를 설치합니다.	온프레미스 데이터베이스 서버와 EC2 인스턴스에 NFS 클라이언트를 설치합니다. 설치 지	DevOps

작업	설명	필요한 기술
	침은 "추가 정보" 섹션을 단원을 참조하세요.	
Amazon EFS 파일 시스템을 탑재합니다.	Amazon EFS 파일 시스템을 온프레미스와 EC2 인스턴스에 탑재합니다. 각 서버에서 백업을 저장할 디렉토리를 만들고 탑재 대상 엔드포인트를 사용하여 파일 시스템을 탑재합니다. 예제를 보려면 "추가 정보" 섹션을 참조하세요.	DevOps

MySQL 소스 데이터베이스의 백업 생성

작업	설명	필요한 기술
페르코나를 XtraBackup 설치하세요.	온프레미스 데이터베이스 서버에 Percona XtraBackup 2.3 또는 2.4 (MySQL 데이터베이스 버전에 따라 다름) 를 설치합니다. 설치 링크는 "관련 리소스" 섹션을 참조하세요.	데이터베이스 관리자
소스 데이터베이스의 스키마와 테이블 수를 계수합니다.	소스 MySQL 데이터베이스의 스키마와 객체 수를 수집하고 기록합니다. 마이그레이션 후 이 수를 사용하여 Aurora MySQL 데이터베이스를 검증합니다.	데이터베이스 관리자
(선택 사항) 소스 데이터베이스의 최신 바이너리 로그 시퀀스를 기록해 둡니다.	소스 데이터베이스와 Aurora MySQL 간에 바이너리 로그 복제를 설정하여 가동 중지 시간을 최소화하려면 이 단계를 수행합니다. log-bin을 활성화해	데이터베이스 관리자

작업	설명	필요한 기술
	<p>야 하고 server_id는 고유해야 합니다. 백업을 시작하기 직전에 소스 데이터베이스의 현재 바이너리 로그 시퀀스를 기록해 둡니다. 전체 백업만 사용하려는 경우 전체 백업 직전에 이 단계를 수행합니다. 전체 백업 후 증분 백업을 수행하려는 경우 Aurora MySQL DB 인스턴스에서 복원할 최종 증분 백업 직전에 이 단계를 수행합니다.</p>	
<p>소스 MySQL 데이터베이스의 전체 백업을 시작합니다.</p>	<p>Percona를 사용하여 MySQL 소스 데이터베이스의 전체 백업을 수행합니다. XtraBackup 전체 및 증분 백업 명령의 예는 "추가 정보" 섹션을 참조하세요.</p>	<p>데이터베이스 관리자</p>

작업	설명	필요한 기술
<p>(선택 사항) Percona를 사용하여 증분 백업을 생성합니다. XtraBackup</p>	<p>증분 백업을 사용하면 소스 데이터베이스를 Aurora MySQL과 동기화하기 위해 적용해야 하는 바이너리 로그의 양을 줄일 수 있습니다. 용량이 크고 트랜잭션이 많은 데이터베이스는 백업 중에 대량의 바이너리 로그를 생성할 수 있습니다. 증분 백업을 수행하여 공유 Amazon EFS 파일 시스템에 저장하면 데이터베이스 백업 및 업로드 시간을 크게 줄일 수 있습니다. 자세한 내용은 "추가 정보" 섹션을 참조하세요. Aurora로 마이그레이션 프로세스를 시작할 준비가 될 때까지 증분 백업을 계속 수행합니다.</p>	<p>데이터베이스 관리자</p>
<p>백업을 준비합니다.</p>	<p>이 단계에서는 백업 중에 진행 중이었던 트랜잭션의 백업에 트랜잭션 로그를 적용합니다. 마지막 백업을 제외하고 각 증분 백업에 트랜잭션 로그 (--apply-log-only) 를 계속 적용하여 백업을 병합합니다. 예제를 보려면 "추가 정보" 섹션을 참조하세요. 이 단계 후에 전체 병합 백업은 ~/efs_mount_name/fullbackup에 저장됩니다.</p>	<p>데이터베이스 관리자</p>

작업	설명	필요한 기술
최종 병합 백업을 압축하고 분할합니다.	최종 병합 백업을 준비한 후에는 tar, zip 및 split 명령을 사용하여 백업의 더 작은 압축 파일을 생성합니다. 예제를 보려면 "추가 정보" 섹션을 참조하세요.	데이터베이스 관리자

Aurora MySQL DB 클러스터에 백업 복원

작업	설명	필요한 기술
Amazon S3로 백업을 업로드합니다.	백업 파일이 저장되는 Amazon EFS 파일 시스템은 온프레미스 데이터베이스와 EC2 인스턴스 모두에 탑재되므로 EC2 인스턴스에서 백업 파일을 쉽게 사용할 수 있습니다. Secure Shell (SSH)을 사용하여 EC2 인스턴스에 연결하고 압축된 백업 파일을 신규 또는 기존 S3 버킷에 업로드합니다 (예: <code>aws s3 sync ~/efs_mount_name/fullbackup s3://<bucket_name>/fullbackup</code>). 자세한 내용은 "관련 리소스" 섹션의 링크를 참조하세요.	AWS DevOps
Aurora가 Amazon S3에 액세스할 수 있도록 서비스 역할을 생성합니다.	"rds.amazonaws.com"을 신뢰하는 IAM 역할과 Aurora가 백업 파일이 저장된 S3 버킷에 액세스할 수 있도록 하는 정책을 생성합니다. 필요한 권한은 ListBucket GetObject, 및 GetObjectVersion 입니다.	AWS DevOps

작업	설명	필요한 기술
Aurora용 네트워킹 구성을 생성합니다.	두 개 이상의 가용 영역과 소스 데이터베이스로의 아웃바운드 연결을 허용하는 서브넷 라우팅 테이블 구성을 포함하는 클러스터 DB 서브넷 그룹을 생성합니다. 온프레미스 데이터베이스가 아웃바운드에 연결하는 것을 허용하고 관리자가 Aurora DB 클러스터에 연결할 수 있도록 허용하는 보안 그룹을 생성합니다. 자세한 내용은 "관련 리소스" 섹션의 링크를 참조하세요.	AWS DevOps /데이터베이스 관리자
백업을 Aurora MySQL DB 클러스터에 복원합니다.	Amazon S3에 업로드한 백업에서 데이터를 복원합니다. 소스 데이터베이스의 MySQL 버전을 지정하고, 백업 파일을 업로드한 S3 버킷 이름 및 폴더 경로 접두사(예: "추가 정보" 섹션의 예제에서는 "fullback up")를 제공하며, Aurora가 Amazon S3에 액세스할 수 있도록 권한을 부여하기 위해 생성한 IAM 역할을 제공합니다.	AWS DevOps /데이터베이스 관리자
Aurora MySQL 데이터베이스를 검증합니다.	복원된 Aurora DB 클러스터의 스키마 및 객체 수를 소스 데이터베이스에서 가져온 수와 비교하여 검증합니다.	데이터베이스 관리자

작업	설명	필요한 기술
빈로그 복제를 설정합니다.	Aurora DB 클러스터에 복원된 마지막 백업을 만들기 전에 앞서 기록해 둔 바이너리 로그 시퀀스를 사용합니다. 소스 데이터베이스에서 복제 사용자를 생성하고 "추가 정보" 섹션의 지침에 따라 적절한 권한을 제공하고, Aurora에서 복제를 활성화하고, 복제가 동기화되었는지 확인합니다.	AWS DevOps /데이터베이스 관리자

관련 리소스

Amazon Elastic 파일 시스템 생성

- [보안 그룹 생성](#)(Amazon VPC 설명서)
- [트랜짓 게이트웨이 VPN 연결](#)(Amazon VPC 설명서)
- [AWS Transit Gateway를 사용하여 VPN 처리량 규모 조정](#)(네트워킹 및 콘텐츠 전송 블로그)
- [Amazon EFS 파일 시스템 생성](#)(Amazon EFS 설명서)
- [탑재 대상 생성](#)(Amazon EFS 설명서)
- [저장 데이터 암호화](#)(Amazon EFS 설명서)

EFS 파일 시스템 탑재

- [Amazon EC2의 IAM 역할](#)(Amazon EC2 설명서)
- [Amazon EC2 Linux 인스턴스 시작](#)(Amazon EC2 설명서)
- [NFS 클라이언트 설치](#)(Amazon EFS 설명서)
- [온프레미스 클라이언트에 Amazon EFS 파일 시스템 탑재](#)(Amazon EFS 설명서)
- [EFS 파일 시스템 탑재](#)(Amazon EFS 설명서)

MySQL 소스 데이터베이스의 백업 생성

- [페르코나 XtraBackup 2.3 설치 \(페르코나 설명서\)](#) XtraBackup

- 페르코나 [2.4 설치 \(페르코나 설명서\) XtraBackup](#) XtraBackup
- [복제 마스터 구성 설정\(MySQL 설명서\)](#)
- [외부 MySQL 데이터베이스의 데이터를 Aurora MySQL DB 클러스터로 마이그레이션\(Aurora 설명서\)](#)
- [중분 백업 \(페르코나 설명서\) XtraBackup](#)

Amazon Aurora MySQL로 백업 복원

- [버킷 생성\(Amazon S3 설명서\)](#)
- [SSH를 사용하여 Linux 인스턴스에 연결\(Amazon Ec2 설명서\)](#)
- [AWS CLI 구성\(AWS CLI 설명서\)](#)
- [동기화 명령\(AWS CLI 명령 참조\)](#)
- [Amazon S3 리소스에 액세스할 수 있는 IAM 정책 생성\(Aurora 설명서\)](#)
- [DB 클러스터 사전 요구 사항\(Aurora 설명서\)](#)
- [DB 서브넷 그룹을 사용한 작업\(Aurora 설명서\)](#)
- [프라이빗 DB 인스턴스에 대한 VPC 보안 그룹 생성\(Aurora 설명서\)](#)
- [S3 버킷에서 Aurora MySQL DB 클러스터 복원\(Aurora 설명서\)](#)
- [MySQL 또는 다른 Aurora DB 클러스터를 사용한 복제 설정 \(Aurora 설명서\)](#)
- [mysql.rds_set_external_master procedure\(Amazon RDS SQL 기반 MySQL 참조\)](#)
- [mysql.rds_start_replication procedure\(Amazon RDS SQL 기반 MySQL 참조\)](#)

추가 참조

- [외부 MySQL 데이터베이스의 데이터를 Aurora MySQL DB 클러스터로 마이그레이션\(Aurora 설명서\)](#)
- [MySQL 서버 다운로드\(Oracle 웹사이트\)](#)

자습서 및 동영상

- [Amazon S3를 사용하여 MySQL 데이터를 Aurora MySQL DB 클러스터로 마이그레이션\(AWS 지식 센터\)](#)
- [Amazon EFS 설치 및 탑재\(동영상\)](#)

추가 정보

NFS 클라이언트 설치

- Red Hat 또는 유사한 Linux 운영 체제를 사용하는 경우 다음 명령을 사용합니다.

```
$ sudo yum -y install nfs-utils
```

- Ubuntu 또는 유사한 Linux 운영 체제를 사용하는 경우 다음 명령을 사용합니다.

```
$ sudo apt-get -y install nfs-common
```

자세한 내용은 Amazon EFS 설명서의 [안내](#)를 참조하세요.

Amazon EFS 파일 시스템 탑재

다음 명령을 사용합니다.

```
mkdir ~/<efs_mount_name>
$ sudo mount -t nfs -o
  nfsvers=4.1,rsize=1048576,wsiz=1048576,hard,timeo=600,retrans=2,noresvport mount-
target-IP:/ ~/<efs_mount_name>
```

자세한 내용은 Amazon EFS 설명서의 [안내](#)와 [EFS 파일 시스템 탑재](#)를 참조하세요.

MySQL 소스 데이터베이스의 백업 생성

전체 백업

다음과 같은 명령을 사용합니다. 이 명령은 백업을 가져와 압축하여 각각 1GB의 작은 청크로 분할합니다.

```
xtrabackup --backup --user=dbuser --password=<password> --binlog-info=AUTO --stream=tar
  --target-dir=~/<efs_mount_name>/fullbackup | gzip - | split -d --bytes=1024MB - ~/
<efs_mount_name>/fullbackup/backup.tar.gz &
```

전체 백업 후 후속 증분 백업을 생성하려는 경우 백업을 압축하여 분할하지 않습니다. 대신 다음과 유사한 명령을 사용합니다.

```
xtrabackup --backup --user=dbuser --password=<password> --target-dir=~/  
<efs_mount_name>/fullbackup/
```

중분 백업

--incremental-basedir 파라미터에는 전체 백업 경로를 사용합니다. 예:

```
xtrabackup --backup --user=dbuser --password=<password> --target-dir=~/  
<efs_mount_name>/incremental/backupdate --incremental-basedir=~/<efs_mount_name>/  
fullbackup
```

여기서 basedir은 전체 백업 및 xtrabackup_checkpoints 파일의 경로입니다.

백업 생성에 관한 자세한 내용은 Aurora 설명서의 [외부 MySQL 데이터베이스에서 Amazon Aurora MySQL DB 클러스터로 데이터 마이그레이션](#)을 참조하세요.

백업 준비

전체 백업을 준비하려면:

```
xtrabackup --prepare --apply-log-only --target-dir=~/<efs_mount_name>/fullbackup
```

중분 백업을 준비하려면:

```
xtrabackup --prepare --apply-log-only --target-dir=~/<efs_mount_name>/fullbackup --  
incremental-dir=~/<efs_mount_name>/incremental/06062020
```

최종 백업을 준비하려면:

```
xtrabackup --prepare --target-dir=~/<efs_mount_name>/fullbackup --incremental-dir=~/  
<efs_mount_name>/incremental/06072020
```

자세한 내용은 Percona [설명서의 중분 백업을](#) 참조하십시오. XtraBackup

병합된 백업의 압축 및 분할

~/<efs_mount_name>/fullbackup에서 병합된 백업을 압축하려면:

```
tar -zcvf <backupfilename.tar.gz> ~/<efs_mount_name>/fullbackup
```

백업을 분할하려면:

```
split -d -b1024M --verbose <backupfilename.tar.gz> <backupfilename.tar.gz>
```

빈로그 복제 설정

소스 데이터베이스에서 복제 사용자를 생성하고 적절한 권한을 제공하려면:

```
CREATE USER 'repl_user'@'' IDENTIFIED BY ''; GRANT REPLICATION CLIENT, REPLICATION  
SLAVE ON *.* TO 'repl_user'@'';
```

Aurora DB 클러스터에 연결하여 Aurora에서 복제를 활성화하려면 DB 클러스터 파라미터 그룹에서 바이너리 로그를 활성화합니다. `binlog_format = mixed`을 설정합니다(혼합 모드 권장됨). 이 변경 사항을 적용하려면 업데이트를 적용하기 위해 인스턴스를 다시 시작해야 합니다.

```
CALL mysql.rds_set_external_master ('sourcedbinstanceIP', sourcedbport, 'repl_user',  
'', 'binlog_file_name', binlog_file_position, 0); CALL mysql.rds_start_replication;
```

복제가 동기화되어 있는지 확인하려면:

```
SHOW Slave Status \G;
```

Seconds Behind Master 필드는 Aurora가 온프레미스 데이터베이스로부터 얼마나 뒤처져 있는지를 보여줍니다.

AWS App2Container를 사용하여 온프레미스 Java 애플리케이션을 AWS로 마이그레이션

작성자: Dhananjay Karanjkar(AWS)

소스: 애플리케이션	대상: Amazon ECS에 배포된 컨테이너식 애플리케이션	R 유형: 리플랫폼
환경: PoC 또는 파일럿	기술: 마이그레이션, 웹 및 모바일 앱	워크로드: 오픈 소스
AWS 서비스: 아마존 EC2 컨테이너 레지스트리, Amazon ECS		

요약

AWS App2Container(A2C)는 코드를 변경할 필요 없이 가상 시스템에서 실행 중인 기존 애플리케이션을 컨테이너로 변환하는 데 도움이 되는 명령줄 도구입니다. A2C는 서버에서 실행 중인 애플리케이션을 검색하고, Amazon Elastic Container Service(Amazon ECS) 및 Amazon Elastic Kubernetes Service(Amazon EKS)에 원활하게 배포할 수 있도록 관련 아티팩트를 생성합니다.

이 패턴은 작업자 시스템을 통해 App2Container를 사용하여 애플리케이션 서버에 배포된 온프레미스 Java 애플리케이션을 AWS Fargate 또는 Amazon EKS로 원격으로 마이그레이션하는 단계를 제공합니다.

작업자 시스템은 다음과 같은 사용 사례에 사용할 수 있습니다.

- Java 애플리케이션이 실행되는 애플리케이션 서버에는 도커 설치가 허용되지 않거나 사용할 수 없습니다.
- 서로 다른 물리적 또는 가상 서버에 배포된 여러 애플리케이션의 마이그레이션을 관리해야 합니다.

사전 조건 및 제한 사항

사전 조건

- Linux 서버에서 실행되는 Java 애플리케이션이 있는 애플리케이션 서버

- Linux 운영 체제를 사용하는 작업자 시스템
- 20GB 이상의 사용 가능한 디스크 공간이 있는 작업자 시스템

제한 사항

- 일부 애플리케이션은 지원되지 않습니다. 자세한 내용은 [Linux에서 지원되는 애플리케이션](#)을 참조하십시오.

아키텍처

소스 기술 스택

- Linux 서버에서 실행되는 Java 애플리케이션

대상 기술 스택

- AWS CodeBuild
- AWS CodeCommit
- AWS CodeDeploy
- AWS CodePipeline
- Amazon Elastic 컨테이너 레지스트리
- AWS Fargate

대상 아키텍처

도구

도구

- [AWS App2Container](#) – AWS App2Container(A2C)는 온프레미스 데이터 센터나 가상 머신에서 실행되는 애플리케이션을 리프트 앤 시프트하여 Amazon ECS 또는 Amazon EKS에서 관리하는 컨테이너에서 실행되도록 도와주는 명령줄 도구입니다.

- [AWS CodeBuild](#) — CodeBuild AWS는 클라우드의 완전 관리형 빌드 서비스입니다. CodeBuild 소스 코드를 컴파일하고, 단위 테스트를 실행하고, 배포할 준비가 된 아티팩트를 생성합니다.
- [AWS CodeCommit](#) — CodeCommit AWS는 Amazon Web Services에서 호스팅하는 버전 제어 서비스로, 클라우드의 자산 (예: 문서, 소스 코드, 바이너리 파일) 을 비공개로 저장하고 관리하는 데 사용할 수 있습니다.
- [AWS CodePipeline](#) — CodePipeline AWS는 소프트웨어 릴리스에 필요한 단계를 모델링, 시각화 및 자동화하는 데 사용할 수 있는 지속적 전송 서비스입니다.
- [Amazon ECS](#) – Amazon Elastic Container Service(Amazon ECS)는 클러스터에서 컨테이너를 실행, 중지 및 관리하기 위한 컨테이너 관리 서비스로서 확장성과 속도가 뛰어납니다.
- [Amazon ECR](#) – Amazon Elastic Container Registry(Amazon ECR)는 안전하고 확장 가능하고 신뢰할 수 있는 AWS 관리형 컨테이너 이미지 레지스트리 서비스입니다.
- [Amazon EKS](#) – Amazon Elastic Kubernetes Service(Amazon EKS)는 Kubernetes 컨트롤 플레인 또는 노드를 설치, 작동 및 유지 관리할 필요 없이 AWS에서 Kubernetes를 실행하는 데 사용할 수 있는 관리형 서비스입니다.
- [AWS Fargate](#) – AWS Fargate는 Amazon Elastic Compute Cloud(Amazon EC2) 인스턴스의 서버나 클러스터를 관리할 필요 없이 컨테이너를 실행하기 위해 Amazon ECS에 사용할 수 있는 기술입니다. Fargate를 사용하면 더 이상 컨테이너를 실행하기 위해 가상 머신의 클러스터를 프로비저닝, 구성 또는 조정할 필요가 없습니다.

에픽

보안 인증 설정

작업	설명	필요한 기술
애플리케이션 서버에 액세스하기 위한 암호를 생성하십시오.	작업자 시스템에서 원격으로 애플리케이션 서버에 액세스하려면 AWS Secrets Manager에서 암호를 생성합니다. 암호에는 SSH 개인 키 또는 인증서와 SSH 개인 키를 사용할 수 있습니다. 자세한 내용은 AWS App2Container에 대한 보안 암호 관리 를 참조하십시오.	DevOps, 개발자

작업자 시스템 설정

작업	설명	필요한 기술
tar 파일을 설치합니다.	<code>sudo yum install -y tar</code> 를 실행합니다.	DevOps, 개발자
AWS CLI를 설치합니다.	Amazon Command Line Interface(AWS CLI)를 설치하고 <code>curl "https://awscli.amazonaws.com/awscli-exe-linux-x86_64.zip" -o "awscliv2.zip"</code> 을 실행합니다. <code>awscliv2.zip</code> 의 압축을 풉니다. <code>sudo ./aws/install</code> 을 실행합니다.	DevOps, 개발자
App2Container를 설치합니다.	다음 명령을 실행합니다. <code>curl -o AWSApp2Container-installer-linux.tar.gz https://app2container-release-us-east-1.s3.us-east-1.amazonaws.com/latest/linux/AWSApp2Container-installer-linux.tar.gz</code> <code>sudo tar xvf AWSApp2Container-installer-linux.tar.gz</code>	DevOps, 개발자

작업	설명	필요한 기술
	<pre>sudo ./install.sh</pre>	
<p>프로파일을 구성합니다.</p>	<p>AWS 기본 프로파일을 구성하려면 <code>sudo aws configure</code> 를 실행합니다.</p> <p>명명된 AWS 기본 프로파일을 구성하려면 <code>sudo aws configure --profile <profile name></code> 을 실행합니다.</p>	<p>DevOps, 개발자</p>
<p>도커를 설치합니다.</p>	<p>다음 명령을 실행합니다.</p> <pre>sudo yum install -y docker</pre> <pre>sudo systemctl enable docker & sudo systemctl restart docker</pre>	

작업	설명	필요한 기술
App2Container를 초기화합니다.	<p>App2Container를 초기화하려면 다음 정보가 필요합니다.</p> <ul style="list-style-type: none"> • <code>workspace</code> : 애플리케이션 컨테이너화 아티팩트를 저장합니다. 최소 20GB의 디스크 여유 공간이 있는 디렉터리 경로를 제공하는 것이 좋습니다. • <code>awsProfile</code> : AWS 프로파일이 서버에 구성되어 있습니다. 이는 Amazon S3에 아티팩트를 업로드하고, <code>containerize</code> 명령을 실행하고, Amazon ECS 또는 Amazon EKS에 배포할 AWS Artifact를 생성하는 데 필요합니다. • <code>s3Bucket</code>: AWS Artifact를 추출하고 저장합니다. • <code>metricsReportPermission</code> : 보고된 지표를 수집 및 저장합니다. • <code>dockerContentTrust</code> : 도커 이미지에 서명합니다. <pre>sudo app2container init</pre> 를 실행합니다.	DevOps, 개발자

작업자 시스템 구성

작업	설명	필요한 기술
애플리케이션 서버에서 App2Container 명령을 원격으로 연결하고 실행하도록 작업자 시스템을 구성합니다.	<p>작업자 시스템을 구성하려면 다음 정보가 필요합니다.</p> <ul style="list-style-type: none"> • Server FQDN: 애플리케이션 서버의 정규화된 도메인 이름입니다. • Server IP address: 애플리케이션 서버의 IP 주소입니다. FQDN이나 IP 주소 중 하나면 충분합니다. • SecretARN : Secrets Manager에 저장되고 애플리케이션 서버에 연결하는데 사용되는 보안 암호의 Amazon 리소스 이름(ARN)입니다. • AuthMethod : key 또는 cert 인증 방법. <p>sudo app2container remote configure 를 실행합니다.</p>	DevOps, 개발자

작업자 시스템에서 애플리케이션을 검색, 분석 및 추출

작업	설명	필요한 기술
온프레미스 Java 애플리케이션을 살펴봅니다.	애플리케이션 서버에서 실행 중인 모든 애플리케이션을 원격으로 검색하려면 다음 명령을 실행합니다.	개발자, DevOps

작업	설명	필요한 기술
	<pre>sudo app2container remote inventory -- target <FQDN/IP of App server></pre> <p>이 명령은 inventory .json 에 배포된 애플리케이션 목록을 생성합니다.</p>	
<p>검색된 애플리케이션을 분석합니다.</p>	<p>인벤토리 단계에서 얻은 application-id를 사용하여 각 애플리케이션을 원격으로 분석하려면 다음 명령을 실행합니다.</p> <pre>sudo app2container remote analyze -- application-id <java- app-id> --target <FQDN/IP of App Server></pre> <p>그러면 작업 공간 위치에 analysis.json 파일이 생성됩니다. 이 파일이 생성된 후 필요에 따라 컨테이너화 파라미터를 변경할 수 있습니다.</p>	<p>개발자, DevOps</p>

작업	설명	필요한 기술
분석된 애플리케이션을 추출합니다.	<p>분석된 애플리케이션에 대한 애플리케이션 아카이브를 생성하려면 다음 명령을 원격으로 실행합니다. 그러면 작업 공간 위치에 tar 번들이 생성됩니다.</p> <pre>sudo app2container remote extract -- application-id <application id> -- target <FQDN/IP of App Server></pre> <p>로컬 작업자 시스템에서 추출된 아티팩트를 생성할 수 있습니다.</p>	개발자, DevOps

작업자 시스템에서 추출된 아티팩트를 컨테이너화합니다.

작업	설명	필요한 기술
추출한 아티팩트를 컨테이너화합니다.	<p>다음 명령을 실행하여 이전 단계에서 추출한 아티팩트를 컨테이너화합니다.</p> <pre>sudo app2container containerize --input- archive <tar bundle location on worker machine></pre>	개발자, DevOps
대상을 지정합니다.	<p>대상을 지정하려면 <code>containerize</code> 명령이 실행될 때 생성되는 <code>deployment.json</code> 을 엽니다. AWS</p>	개발자, DevOps

작업	설명	필요한 기술
	Fargate를 대상으로 지정하려면 createEcsArtifacts 를 true로 설정합니다. Amazon EKS를 대상으로 설정하려면 createEksArtifacts 를 true로 설정합니다.	

AWS Artifact 생성 및 프로비저닝

작업	설명	필요한 기술
작업자 시스템에서 AWS 배포 아티팩트를 생성합니다.	<p>배포 아티팩트 생성하려면 다음 명령을 실행합니다.</p> <pre>sudo app2container generate app-deplo yment --application- id <application id></pre> <p>그러면 워크스페이스에 ecs-master.yml AWS CloudFormation 템플릿이 생성됩니다.</p>	DevOps
아티팩트를 프로비저닝합니다.	<p>생성된 아티팩트를 추가로 프로비저닝하려면 다음 명령을 실행하여 AWS CloudFormation 템플릿을 배포하십시오.</p> <pre>aws cloudformation deploy --template- file <path to ecs- master.yml> --capabil ities CAPABILIT</pre>	DevOps

작업	설명	필요한 기술
	<code>Y_NAMED_IAM --stack-name <application id>-ECS</code>	
파이프라인을 생성합니다.	이전 스토리에서 만든 내용을 필요에 따라 pipeline.json 을 수정합니다. 그런 다음 generate pipeline 명령을 실행하여 파이프라인 배포 아티팩트를 생성합니다.	DevOps

관련 리소스

- [App2Container란 무엇입니까?](#)
- [AWS App2Container 블로그 게시물](#)
- [AWS CLI 구성 기본 사항](#)
- [Amazon ECS용 도커 기본 사항](#)
- [도커 명령](#)

AWS 대규모 마이그레이션에서 공유 파일 시스템 마이그레이션

작성자: Amit Rudraraju(AWS), Sam Apa(AWS), Bheemeswararao Balla(AWS), Wally Lu(AWS), Sanjeev Prakasam(AWS)

환경: 프로덕션	소스: 온프레미스 공유 파일 시스템	대상: Amazon EFS 또는 Amazon FSx
R 타입: 리플랫폼	워크로드: 기타 모든 워크로드	기술: 마이그레이션, 스토리지 및 백업
AWS 서비스: AWS DataSync, 아마존 EFS, 윈도우 파일 서버용 아마존 FSx, ONTAP용 아마존 FSx NetApp		

요약

300대 이상의 서버를 마이그레이션하는 것은 대규모 마이그레이션으로 간주됩니다. 대규모 마이그레이션의 목적은 기존 온프레미스 데이터 센터에서 AWS 클라우드로 워크로드를 마이그레이션하는 것이며, 이러한 프로젝트는 일반적으로 애플리케이션 및 데이터베이스 워크로드에 중점을 둡니다. 그러나 공유 파일 시스템에는 집중적인 주의와 별도의 마이그레이션 계획이 필요합니다. 이 패턴은 공유 파일 시스템의 마이그레이션 프로세스를 설명하고 대규모 마이그레이션 프로젝트의 일환으로 공유 파일 시스템을 성공적으로 마이그레이션하기 위한 모범 사례를 제공합니다.

네트워크 또는 클러스터링된 파일 시스템이라고도 하는 공유 파일 시스템(SFS)은 여러 서버에 마운트되는 파일 공유입니다. 공유 파일 시스템은 NFS(네트워크 파일 시스템), CIFS(공용 인터넷 파일 시스템) 또는 SMB(서버 메시지 블록)와 같은 프로토콜을 통해 액세스됩니다.

이러한 시스템은 마이그레이션되는 호스트 전용도 아니고 블록 디바이스로 표시되지도 않기 때문에 AWS Application Migration Service와 같은 표준 마이그레이션 도구를 사용하여 마이그레이션되지 않습니다. 대부분의 호스트 종속성은 투명하게 마이그레이션되지만 종속 파일 시스템의 조정 및 관리는 별도로 처리해야 합니다.

검색, 계획, 준비, 축소, 검증 등의 단계를 거쳐 공유 파일 시스템을 마이그레이션합니다. 이 패턴과 첨부된 통합 문서를 사용하여 공유 파일 시스템을 Amazon Elastic File System (Amazon EFS), NetApp ONTAP용 Amazon FSx 또는 Windows File Server용 Amazon FSx와 같은 AWS 스토리지 서비스로

마이그레이션합니다. 파일 시스템을 전송하려면 AWS DataSync 또는 타사 도구 (예:) 를 사용할 수 NetApp SnapMirror 있습니다.

참고: 이 패턴은 [AWS 클라우드로의 대규모 마이그레이션](#)에 관한 AWS 권장 가이드 시리즈의 일부입니다. 이 패턴에는 서버용 웨이브 플랜에 SFS를 통합하는 모범 사례와 지침이 포함되어 있습니다. [대규모 마이그레이션 프로젝트 외부에서 하나 이상의 공유 파일 시스템을 마이그레이션하는 경우 Amazon EFS, Windows File Server용 Amazon FSx, ONTAP용 Amazon FSx 설명서의 데이터 전송 지침을 참조하십시오. NetApp](#)

사전 조건 및 제한 사항

사전 조건

사전 조건은 소스 및 대상 공유 파일 시스템과 사용 사례에 따라 달라질 수 있습니다. 가장 일반적인 문제는 다음과 같습니다.

- 활성 상태의 AWS 계정
- 대규모 마이그레이션 프로젝트를 위한 애플리케이션 포트폴리오 검색을 완료하고 웨이브 플랜 개발을 시작했습니다. 자세한 내용은 [AWS 대규모 마이그레이션을 위한 포트폴리오 플레이북](#)을 참조하십시오.
- 온프레미스 데이터 센터와 AWS 환경 간의 유입 및 송신 트래픽을 허용하는 Virtual Private Cloud(VPC) 및 보안 그룹. [자세한 내용은 네트워크-Amazon VPC 연결 옵션 및 AWS 네트워크 요구 사항을 참조하십시오. DataSync](#)
- AWS CloudFormation 스택을 생성할 수 있는 권한 또는 Amazon EFS 또는 Amazon FSx 리소스를 생성할 수 있는 권한 자세한 내용은 [CloudFormation 설명서, Amazon EFS 설명서 또는 Amazon FSx 설명서](#)를 참조하십시오.
- DataSync AWS를 사용하여 마이그레이션을 수행하는 경우 다음 권한이 필요합니다.
 - AWS가 AWS Logs 로그 그룹에 로그를 DataSync 전송할 수 있는 권한. CloudWatch 자세한 내용은 로그 [그룹에 CloudWatch 로그 업로드 DataSync 허용](#)을 참조하십시오.
 - CloudWatch 로그 로그 그룹에 액세스할 수 있는 권한. 자세한 내용은 [CloudWatch Logs 리소스에 대한 액세스 권한 관리 개요](#)를 참조하십시오.
 - 에서 상담원 및 작업을 생성할 DataSync 수 있는 권한 자세한 내용은 [AWS DataSync 사용에 필요한 IAM 권한](#)을 참조하십시오.

제한 사항

- 이 패턴은 대규모 마이그레이션 프로젝트의 일환으로 SFS를 마이그레이션하도록 설계되었습니다. 여기에는 애플리케이션 마이그레이션을 위한 웨이브 플랜에 SFS를 통합하기 위한 모범 사례 및 지침이 포함되어 있습니다. [대규모 마이그레이션 프로젝트 외부에서 하나 이상의 공유 파일 시스템을 마이그레이션하는 경우 Amazon EFS, Windows File Server용 Amazon FSx, ONTAP용 Amazon FSx 설명서의 데이터 전송 지침을 참조하십시오. NetApp](#)
- 이 패턴은 일반적으로 사용되는 아키텍처, 서비스 및 마이그레이션 패턴을 기반으로 합니다. 그러나 대규모 마이그레이션 프로젝트와 전략은 조직마다 다를 수 있습니다. 요구 사항에 따라 이 솔루션 또는 제공된 통합 문서를 사용자 지정해야 할 수 있습니다.

아키텍처

소스 기술 스택

다음 중 한 개 이상을 수행할 수 있습니다.

- Linux(NFS) 파일 서버
- Windows (SMB) 파일 서버
- NetApp 스토리지 어레이
- Dell EMC Isilon 스토리지 어레이

대상 기술 스택

다음 중 한 개 이상을 수행할 수 있습니다.

- Amazon Elastic File System
- ONTAP용 아마존 NetApp FSx
- Amazon FSx for Windows File Server

대상 아키텍처

이 다이어그램은 다음 사항을 보여 줍니다.

1. AWS Direct Connect 또는 AWS Site-to-Site VPN과 같은 AWS 서비스를 사용하여 온프레미스 데이터 센터와 AWS 클라우드 간에 연결을 설정합니다.
2. 온프레미스 데이터 센터에 DataSync 에이전트를 설치합니다.

3. 웨이브 플랜에 따라 원본 공유 파일 시스템의 데이터를 대상 AWS 파일 공유로 복제하는 DataSync 데 사용합니다.

마이그레이션 단계

다음 이미지는 대규모 마이그레이션 프로젝트에서 SFS를 마이그레이션하기 위한 단계 및 상위 단계를 보여줍니다.

이 패턴의 [에픽](#) 섹션에는 마이그레이션을 완료하고 첨부된 워크북을 사용하는 방법에 대한 자세한 지침이 포함되어 있습니다. 다음 사항은 이 단계별 접근 방식의 단계를 개괄적으로 보여줍니다.

Phase(단계)	단계
찾기	<ol style="list-style-type: none"> 1. 검색 도구를 사용하여 서버, 마운트 지점, IP 주소 등 공유 파일 시스템에 대한 데이터를 수집합니다. 2. 구성 관리 데이터베이스(CMDB) 또는 마이그레이션 도구를 사용하여 마이그레이션 웨이브, 환경, 응용 프로그램 소유자, IT 서비스 관리 (ITSM) 서비스 이름, 조직 단위 및 애플리케이션 ID에 대한 정보를 포함하여 서버에 대한 세부 정보를 수집합니다.
계획	<ol style="list-style-type: none"> 3. SFS와 서버에 대해 수집된 정보를 사용하여 SFS 웨이브 플랜을 생성합니다. 4. 빌드 워크시트의 정보를 사용하여 각 SFS에 대해 대상 AWS 서비스와 마이그레이션 도구를 선택합니다.
준비	<ol style="list-style-type: none"> 5. Amazon EFS, ONTAP용 Amazon FSx 또는 Windows File NetApp Server용 Amazon FSx에서 대상 인프라를 설정합니다. 6. 데이터 전송 서비스 (예:) 를 설정한 다음 초기 DataSync 데이터 동기화를 시작합니다. 초기 동

기화가 완료되면 일정에 따라 실행되도록 반복 동기화를 설정할 수 있습니다.

전환

7. IP 주소 또는 경로와 같은 대상 파일 공유에 대한 정보로 SFS 웨이브 플랜을 업데이트합니다.

8. 소스 SFS에 적극적으로 액세스하는 애플리케이션을 중지합니다.

9. 데이터 전송 서비스에서 최종 데이터 동기화를 수행합니다.

10. 동기화가 완료되면 Logs의 로그 데이터를 검토하여 동기화가 완전히 성공적으로 완료되었는지 확인하세요 CloudWatch .

Validate

11. 서버에서 마운트 지점을 새 SFS 경로로 변경합니다.

12. 애플리케이션을 다시 시작하고 유효성을 검사합니다.

도구

서비스

- [Amazon CloudWatch Logs](#)를 사용하면 모든 시스템, 애플리케이션 및 AWS 서비스의 로그를 중앙 집중화하여 로그를 모니터링하고 안전하게 보관할 수 있습니다.
- [DataSyncAWS](#)는 파일 또는 객체 데이터를 AWS 스토리지 서비스 간에, AWS 스토리지 서비스 간에 이동하는 데 도움이 되는 온라인 데이터 전송 및 검색 서비스입니다.
- [Amazon Elastic File System\(Amazon EFS\)](#)은 AWS 클라우드에서 공유 파일 시스템을 생성하고 구성하는 데 도움이 됩니다.
- [Amazon FSx](#)는 업계 표준 연결 프로토콜을 지원하고 AWS 리전 전반에 걸쳐고가용성 및 복제를 제공하는 파일 시스템을 제공합니다.

기타 도구

- [SnapMirror](#) 지정된 소스 볼륨 또는 [qtree](#)에서 각각 대상 볼륨 또는 [qtree](#)로 데이터를 복제하는 데이터 복제 도구입니다. NetApp 이 도구를 사용하여 NetApp 소스 파일 시스템을 ONTAP용 Amazon FSx 로 마이그레이션할 수 있습니다.
- [Robocopy](#)는 Robust 파일 복사의 줄임말로, Windows용 명령줄 디렉토리 및 명령입니다. 이 도구를 사용하여 Windows 소스 파일 시스템을 Amazon FSx for Windows File Server로 마이그레이션할 수 있습니다.

모범 사례

웨이브 계획 접근법

대규모 마이그레이션 프로젝트를 위한 웨이브를 계획할 때는 지연 시간과 애플리케이션 성능을 고려해야 합니다. SFS와 종속 애플리케이션이 서로 다른 위치(예: 클라우드와 온프레미스 데이터 센터)에서 운영되는 경우 지연 시간이 증가하고 애플리케이션 성능에 영향을 미칠 수 있습니다. 다음 사항은 웨이브 플랜을 만들 때 사용 가능한 옵션입니다.

1. SFS와 모든 종속 서버를 동일한 웨이브 내에서 마이그레이션. - 이 접근 방식은 성능 문제를 예방하고 마운트 지점을 여러 번 재구성하는 등의 재작업을 최소화합니다. 애플리케이션과 SFS 사이의 지연 시간이 매우 짧아야 할 때 사용하는 것이 좋습니다. 그러나 웨이브 플래닝은 복잡하며, 종속성 그룹에서 변수를 추가하는 것이 아니라 종속성 그룹에서 변수를 제거하는 것이 목표입니다. 또한 여러 서버가 동일한 SFS에 액세스하는 경우에는 웨이브 규모가 너무 커지므로 이 방법을 사용하지 않는 것이 좋습니다.
2. 마지막 종속 서버를 마이그레이션한 후 SFS 마이그레이션- 예를 들어 여러 서버가 SFS에 액세스하고 해당 서버가 4, 6, 7번 웨이브에서 마이그레이션되도록 스케줄링하는 경우 웨이브 7에서 SFS가 마이그레이션되도록 스케줄을 잡으세요.

이 접근 방식은 대규모 마이그레이션의 경우 가장 논리적인 경우가 많으며 지연 시간에 민감한 애플리케이션에 권장됩니다. 데이터 전송과 관련된 비용을 줄일 수 있습니다. 또한 상위 계층 애플리케이션은 일반적으로 개발 및 QA 애플리케이션 이후에 마지막으로 마이그레이션되도록 예약되므로 SFS와 상위 계층(예: 프로덕션) 애플리케이션 간의 지연 시간을 최소화합니다.

그러나 이 접근 방식에는 여전히 검색, 계획 및 민첩성이 필요합니다. SFS를 더 일찍 마이그레이션해야 할 수도 있습니다. 애플리케이션이 첫 번째 종속 웨이브와 SFS를 포함하는 웨이브 사이의 시간 동안 추가 지연 시간을 견딜 수 있는지 확인합니다. 애플리케이션 소유자와 검색 세션을 진행하고 지연 시간에 가장 민감한 애플리케이션인 동일한 웨이브로 애플리케이션을 마이그레이션합니다. 종속 애플리케이션을 마이그레이션한 후 성능 문제가 발견되면 최대한 빨리 SFS를 마이그레이션할 수 있도록 신속하게 방향을 전환할 준비를 하세요.

3. 대규모 마이그레이션 프로젝트 종료 시 SFS 마이그레이션 - SFS의 데이터에 자주 액세스하지 않거나 애플리케이션 성능에 중요하지 않은 경우와 같이 지연 시간이 문제가 되지 않는 경우에는 이 방법을 사용하는 것이 좋습니다. 이 접근 방식은 마이그레이션을 간소화하고 컷오버 작업을 단순화합니다.

애플리케이션의 지연 시간 민감도에 따라 이러한 접근 방식을 혼합할 수 있습니다. 예를 들어 접근 방식 1 또는 2를 사용하여 지연 시간에 민감한 SFS를 마이그레이션한 다음 접근 방식 3을 사용하여 나머지 SFS를 마이그레이션할 수 있습니다.

AWS 파일 시스템 서비스 선택

AWS는 파일 스토리지를 위한 여러 클라우드 서비스를 제공합니다. 각각은 성능, 규모, 접근성, 통합, 규정 준수 및 비용 최적화에 대해 서로 다른 혜택과 제한을 제공합니다. 몇 가지 논리적 기본 옵션이 있습니다. 예를 들어 현재 온프레미스 파일 시스템이 Windows Server를 운영하고 있다면 Amazon FSx for Windows File Server가 기본 선택 사항입니다. 또는 온프레미스 파일 시스템이 ONTAP를 운영하는 경우 NetApp ONTAP용 NetApp Amazon FSx가 기본 선택 사항입니다. 그러나 애플리케이션의 요구 사항에 따라 대상 서비스를 선택하거나 다른 클라우드 운영상의 이점을 실현하기 위해 대상 서비스를 선택할 수 있습니다. 자세한 내용은 [배포에 적합한 AWS 파일 스토리지 서비스 선택](#)(AWS Summit 프레젠테이션)을 참조하세요.

마이그레이션 도구 선택

Amazon EFS와 Amazon FSx는 AWS를 사용하여 공유 파일 시스템을 DataSync AWS 클라우드로 마이그레이션할 수 있도록 지원합니다. 지원되는 스토리지 시스템 및 서비스, 이점, 사용 사례에 대한 자세한 내용은 [AWS란?](#) 을 참조하십시오 DataSync. 파일을 전송하는 DataSync 데 사용하는 프로세스의 개요는 [AWS DataSync 전송 작동 방식을](#) 참조하십시오.

또한 다음 사항을 포함하여 여러 서드파티 도구를 사용할 수 있습니다.

- NetApp ONTAP용 Amazon FSx를 선택하면 온프레미스 데이터 센터에서 클라우드로 파일을 마이그레이션하는 데 NetApp SnapMirror 사용할 수 있습니다. SnapMirror 블록 수준 복제를 사용하므로 데이터 전송 프로세스보다 DataSync 빠르며 지속 시간을 줄일 수 있습니다. 자세한 내용은 ONTAP용 [FSx로 마이그레이션을](#) 참조하십시오. NetApp SnapMirror
- Amazon FSx for Windows File Server를 선택하는 경우 Robocopy를 사용하여 파일을 클라우드로 마이그레이션할 수 있습니다. 자세한 내용은 [Robocopy를 사용하여 기존 파일을 FSx for Windows File Server로 마이그레이션을](#) 참조하세요.

에픽

찾기

작업	설명	필요한 기술
SFS 검색 워크북을 준비합니다.	<ol style="list-style-type: none"> 이 패턴의 첨부 파일 섹션에서 통합 문서를 다운로드합니다. 여기에는 SFS-Discovery-Workbook.xlsx 및 SFS-Wave-Plan-Workbook.xlsx와 같은 두 개의 파일이 포함되어 있습니다. Microsoft Excel에서 SFS-Discovery-Workbook 파일을 엽니다. 대시보드에서 다음 작업을 수행하세요. <ul style="list-style-type: none"> A열에서 환경 이름을 업데이트합니다. B열에서 환경 순서를 업데이트하여 우선 순위가 가장 낮은 (1)에서 가장 높은 우선 순위로 정렬합니다. D~E열에서 웨이브 스케줄을 업데이트합니다. C열과 K열에서 AWS 계정 이름을 업데이트합니다. L열에서 VPC ID를 업데이트합니다. M~O열에서 서브넷 ID를 업데이트합니다. 통합 문서 템플릿의 나머지 부분을 검토하고 조직 또는 	마이그레이션 엔지니어, 마이그레이션 책임자

작업	설명	필요한 기술
	사용 사례에 필요한 기타 모든 값을 업데이트하세요. 5. 워크북을 저장합니다.	

작업	설명	필요한 기술
<p>소스 SFS에 대한 정보를 수집합니다.</p>	<ol style="list-style-type: none"> 1. 선호하는 검색 도구를 사용하여 해당하는 모든 스토리지 디바이스, Linux 서버 및 Windows 서버의 모든 SFS 마운트를 식별합니다. 일반적으로 다음 정보를 수집해야 합니다. <ul style="list-style-type: none"> • 클라이언트 디바이스 • 클라이언트 IP 주소 • SFS 세부 정보 • 탑재 지점 <p>참고: 마이그레이션 후 SFS를 다시 마운트하기 위한 마운트 지점 세부 정보를 마이그레이션 런북에 추가할 수 있습니다.</p> 2. SFS-Discovery-Workbook 파일을 엽니다. 3. 웨이브-시트 워크시트에서 다음 사항을 수행하세요. <ul style="list-style-type: none"> • 수식의 서버 위치(D) 열에 있는 온프레미스 소스의 CIDR 범위 형식이 해당 범위에 맞는 지 확인합니다. 예를 들어 CIDR 범위가 10.0.0.0/8 이면 10.*.*.*을 입력하세요. • SFS 위치(E) 열의 공식에서 대상 VPC의 CIDR 범위 형식이 해당 범위에 맞는 지 확인합니다. 예를 들어 CIDR 범위가 	<p>마이그레이션 엔지니어, 마이그레이션 책임자</p>

작업	설명	필요한 기술
	<p>176.16.0.0/16 이면 176.16.*.* 을 입력하세요.</p> <p>4. SFS 데이터 워크시트에서 다음 사항을 수행합니다.</p> <ul style="list-style-type: none"> • 서버 이름(A) 열에 SFS가 마운트된 서버의 이름을 입력합니다. • SFS 경로(B) 열에 SFS의 이름을 입력합니다. • IP 주소(C) 열에 서버의 IP 주소를 입력합니다. • 검색 중에 수집한 기타 관련 정보(예: 마운트 지점 및 SFS 크기)를 추가합니다. 나중에 이 데이터를 사용하여 웨이브 계획 계산을 수정할 수 있습니다. <p>5. 워크북을 저장합니다.</p>	

작업	설명	필요한 기술
<p>서버에 대한 정보를 수집합니다.</p>	<ol style="list-style-type: none"> 1. CMDB 또는 마이그레이션 도구에 기록된 데이터를 사용하여 SFS 마운트가 있는 서버에 대한 다음 정보를 모두 식별하세요. <ul style="list-style-type: none"> • [서버 이름] • IP 주소 • 웨이브 • 조직 단위(OU) • 서버 환경(예: DEV, QA 또는 PROD) • 애플리케이션 이름 • 애플리케이션 소유자 및 연락처 정보 2. SFS-Discovery-Workbook 파일을 엽니다. 3. 서버-데이터 워크시트의 A~H 열에 원본 서버에 대해 수집한 정보를 입력합니다. <p>유념할 사항:</p> <ul style="list-style-type: none"> • Wave # (C) 열에 웨이브 이름 (예:), (Wave1) 또는 를 입력합니다. out-of-scope OOS Retire • 앱 소유자 연락처(H) 열에 있는 경우 이메일 주소가 정확한지 확인하세요. 이 이메일 주소는 앱 소유자(G) 열에 입력한 이름을 기반으로 자동으로 생성됩니다. 필요한 경우 올바른 이메일 주소를 반영하 	<p>마이그레이션 엔지니어, 마이그레이션 책임자</p>

작업	설명	필요한 기술
	<p>도록 값을 수동으로 업데이트하세요.</p> <ul style="list-style-type: none"> 공식이 포함된 I~J 열은 수정하지 마세요. <p>4. 워크북을 저장합니다.</p>	

계획

작업	설명	필요한 기술
<p>SFS 웨이브 플랜을 빌드합니다.</p>	<ol style="list-style-type: none"> SFS-Discovery-Workbook 파일을 엽니다. 검색 단계에서 수집된 모든 정보가 정확하고 최신인지 검증합니다. 웨이브-시트 워크시트에서 SFS 웨이브(K) 열을 1 값으로 필터링합니다. 첫 번째 웨이브의 모든 SFS 목록입니다. <p>참고: 이 열의 0 값이 0이면 SFS가 마이그레이션 범위를 벗어났음을 나타냅니다. 이는 SFS가 이미 AWS에 호스팅되어 있거나 공유에 액세스하는 서버가 마이그레이션 범위를 벗어났기 때문일 수 있습니다.</p> <ol style="list-style-type: none"> 이번 웨이브에서 이러한 SFS를 마이그레이션하고 싶은지 검증합니다. 웨이브에 SFS를 할당하는 방법에 대한 자세한 내용은 모범 사례 	<p>빌드 책임자, 전환 리드, 마이그레이션 엔지니어, 마이그레이션 책임자</p>

작업	설명	필요한 기술
	<p>섹션의 웨이브 계획 접근 방식을 참조하세요.</p> <ol style="list-style-type: none"> 5. 필터링된 값이 들어 있는 셀을 선택하고 복사합니다. 열 제목이 포함된 헤더 행은 복사하지 마세요. 6. 이전에 다운로드한 SFS-Wave-Plan-Workbook 파일을 엽니다. 7. Export-from-Discovery 워크시트에서 A2 셀을 선택합니다. 8. 복사한 데이터를 붙여넣습니다. 9. SFS-Discovery-Workbook 및 SFS-Wave-Plan-Workbook 파일을 저장합니다. 	

작업	설명	필요한 기술
대상 AWS 서비스 및 마이그레이션 도구를 선택합니다.	<ol style="list-style-type: none"> 1. SFS-Wave-Plan-Workbook 파일의 Exported-from-Discovery 워크시트에서 기존 경로(C) 열에 있는 값을 선택하고 복사합니다. 2. Build-Wave 워크시트에서 A2 셀을 선택합니다. 3. 복사한 데이터를 붙여넣습니다. 이 워크시트의 B~M 열은 이 경로와 관련된 다른 데이터를 반영하도록 자동으로 업데이트됩니다. 4. A열에서 중복된 값을 모두 제거합니다. 자세한 지침은 중복 값 제거(Microsoft 지원 웹사이트)를 참조하세요. 5. Target 패턴 또는 서비스(F) 열에서 권장 대상 AWS 서비스를 검토하고 필요에 따라 업데이트합니다. 자세한 내용은 이 패턴의 모범 사례 섹션에서 AWS 파일 시스템 서비스 선택을 참조하세요. 6. 마이그레이션 메서드(G) 열에서 권장 마이그레이션 도구를 검토하고 필요에 따라 업데이트합니다. 자세한 내용은 이 패턴의 모범 사례 섹션에서 마이그레이션 도구 선택을 참조하세요. 7. SFS-Discovery-Workbook 파일을 저장합니다. 이 웨이 	마이그레이션 엔지니어, 마이그레이션 책임자

작업	설명	필요한 기술
	<p>브에 대한 웨이브 플랜 작성이 완료되었습니다.</p> <p>8. 이 지침을 반복하여 각 웨이브에 대한 웨이브 계획을 준비합니다. 웨이브 플랜은 마이그레이션 중에 변경될 수 있으므로 미리 5개 이상의 웨이브를 계획하지 않는 것이 좋습니다.</p>	

준비

작업	설명	필요한 기술
대상 파일 시스템을 설정합니다.	<p>웨이브 플랜에 기록된 세부 정보에 따라 대상 AWS 계정, VPC 및 서브넷에 대상 파일 시스템을 설정합니다. 지침은 다음 AWS 설명서를 참조하세요.</p> <ul style="list-style-type: none"> • Amazon EFS • ONTAP용 아마존 NetApp FSx • Amazon FSx for Windows File Server 	마이그레이션 엔지니어, 마이그레이션 책임자, AWS 관리자
마이그레이션 도구를 설정하고 데이터를 전송합니다.	<ol style="list-style-type: none"> 1. DataSyncAWS를 사용하는 경우 DataSync 작업에 대한 로깅을 구성하십시오. 지침은 AWS DataSync 작업 활동 로깅을 참조하십시오. 2. 마이그레이션 도구를 설정하고 선택한 도구의 지침에 	AWS 관리자, 클라우드 관리자, 마이그레이션 엔지니어, 마이그레이션 책임자

작업	설명	필요한 기술
	<p>따라 초기 데이터 전송을 수행합니다.</p> <ul style="list-style-type: none"> • Amazon EFS의 경우 다음 사항을 참조하세요. <ul style="list-style-type: none"> • AWS를 사용하여 Amazon EFS로 파일 전송 DataSync • Amazon FSx for ONTAP의 경우 다음 사항을 참조하세요. <ul style="list-style-type: none"> • 를 사용하여 ONTAP용 FSx로 마이그레이션하는 중 NetApp SnapMirror • AWS를 사용하여 ONTAP용 FSx로 마이그레이션하기 DataSync • Amazon FSx for Windows File Server의 경우 다음 사항을 참조하세요. <ul style="list-style-type: none"> • AWS를 사용하여 기존 파일을 Windows File Server용 FSx로 마이그레이션하기 DataSync • Robocopy를 사용하여 기존 파일을 FSx for Windows File Server로 마이그레이션 <p>3. 초기 전송 중 또는 이후에 소스 SFS가 변경될 수 있습니다. 데이터를 동기화된 상태로 유지하려면 소스와 타겟</p>	

작업	설명	필요한 기술
	<p>파일 시스템 간에 반복 데이터 전송을 설정하세요.</p> <ul style="list-style-type: none">• 를 사용하는 DataSync 경우 AWS DataSync 작업 예약을 참조하십시오. DataSync 소스 SFS에서 수정된 파일이나 새 파일만 전송합니다.• 타사 도구를 사용하는 경우 선택한 도구의 설명서를 참조하세요.	

작업	설명	필요한 기술
웨이브 플랜을 업데이트합니다.	<ol style="list-style-type: none"> 1. 현재 웨이브에 대한 SFS-Wave-Plan-Workbook 파일을 엽니다. 2. Build-Wave 워크시트의 새 경로 IP 주소(N) 열에 대상 파일 시스템의 IP 주소를 입력합니다. 다음 중 하나를 수행하여 IP 주소를 찾습니다. <ul style="list-style-type: none"> • FSx for Windows File Server의 경우 Amazon FSx 콘솔에서 파일 시스템을 선택하고 파일 시스템을 선택한 다음 네트워크 및 보안을 확인하세요. • ONTAP용 FSx의 경우 블록 마운팅을 참조하세요. • Amazon EFS의 경우 IP 주소를 사용한 마운트를 참조하세요. 3. 새 경로(O) 열에 새 마운트 경로를 입력합니다. 마운트 경로는 파일 시스템의 DNS 이름입니다. 다음 중 하나를 수행하여 마운트 경로를 찾습니다. <ul style="list-style-type: none"> • FSx for Windows File Server의 경우 Amazon FSx 콘솔에서 파일 시스템을 선택하고 파일 시스템을 선택한 다음 연결을 선택합니다. • FSx for ONTAP의 경우 파일 시스템 세부 정보 페이지 	마이그레이션 엔지니어, 마이그레이션 책임자

작업	설명	필요한 기술
	<p>지를 참조하세요. 지침은 볼륨 마운트를 참조하세요.</p> <ul style="list-style-type: none"> • Amazon EFS의 경우 정보 수집을 참조하세요. <p>4. 재마운트-요약 워크시트에 새 경로 (C) 및 새 경로 IP 주소 (D) 열에 업데이트된 값이 반영되어 있는지 확인합니다.</p> <p>5. 조직에서 전환 후 Linux 및 Windows 파일 시스템을 다시 마운트하기 위한 런북을 준비했는지 확인합니다. 일반적인 지침은 다음 사항을 참조하세요.</p> <ul style="list-style-type: none"> • EFS 파일 시스템 탑재 • FSx for Windows File Server 파일 공유에 액세스 • ONTAP 볼륨용 FSx 마운트 <p>6. 종속 서버가 이 웨이브에 포함되지 않은 경우 App-Team-Communication 워크시트에 기록하세요. 표준 웨이브 통신에 포함되지 않을 수 있으므로 해당 애플리케이션 또는 서버 소유자에게 알려주세요.</p> <p>7. 웨이브 플랜을 완료한 후 웨이브에서 SFS가 제거되면</p>	

작업	설명	필요한 기술
전환	Descoped 워크시트에서 해당 내용을 추적하세요.	

작업	설명	필요한 기술
애플리케이션을 중지합니다.	<p>애플리케이션 또는 클라이언트가 소스 SFS에서 읽기 및 쓰기 작업을 활발히 수행하는 경우 최종 데이터 동기화를 수행하기 전에 중지하세요. 지침은 읽기 및 쓰기 작업 중지에 대한 애플리케이션 설명서 또는 내부 프로세스를 참조하세요. 예를 들어 웹 서버 시작 또는 중지(IIS 8)(Microsoft 설명서) 또는 systemctl을 사용한 시스템 서비스 관리(Red Hat 문서)를 참조하세요.</p>	앱 소유자, 앱 개발자
최종 데이터 전송을 수행합니다.	<ol style="list-style-type: none"> 1. 마이그레이션 도구에서 최종 데이터 전송 작업 또는 작업을 수동으로 실행하여 대상 파일 시스템을 소스 SFS와 동기화합니다. 지침은 DataSync 작업 시작을 참조하거나 선택한 타사 마이그레이션 도구의 설명서를 참조하십시오. 2. 데이터 전송 작업이 완료될 때까지 기다립니다. 자세한 내용은 Amazon을 통한 AWS DataSync 활동 모니터 	마이그레이션 엔지니어, 마이그레이션 책임자

작업	설명	필요한 기술
	링 CloudWatch 및 명령줄에서 DataSync 작업 모니터링 을 참조하십시오.	

작업	설명	필요한 기술
<p>데이터 전송을 검증합니다.</p>	<p>AWS를 사용하는 경우 DataSync, 다음을 수행하여 최종 데이터 전송이 성공적으로 완료되었는지 확인하십시오.</p> <ol style="list-style-type: none"> 1. AWS DataSync 콘솔에서 작업 및 실행 ID (예:) 를 기록해 둡니다 task-0000-exec-1111 . 2. 작업의 작업 로깅 섹션으로 DataSync 이동합니다. 3. CloudWatch 로그 그룹 링크를 선택합니다. 4. 로그에서 작업 및 실행 ID를 검색합니다. 5. 모든 전송 오류를 기록합니다. 자세한 내용은 DataSync 설명서의 일반적인 오류를 참조하십시오. 6. 다음 사항을 검증합니다. <ul style="list-style-type: none"> • 소스 및 타겟 SFS의 파일 목록을 비교하여 모든 데이터가 전송되었는지 확인 • 소스 및 타겟 SFS의 파일 액세스 권한을 비교합니다. <p>타사 도구를 사용하는 경우 선택한 마이그레이션 도구의 설명서에서 데이터 전송 검증 지침을 참조하세요.</p>	<p>마이그레이션 엔지니어, 마이그레이션 책임자</p>

검증

작업	설명	필요한 기술
<p>파일 시스템을 다시 마운트하고 애플리케이션 기능 및 성능을 검증합니다.</p>	<ol style="list-style-type: none"> 1. 이 웨이브에서 종속 서버를 마이그레이션한 경우 SFS-Wave-Plan-Workbook 파일의 리마운트 요약 워크시트에 있는 새 서버 IP 주소(F) 열에 서버의 새 IP 주소를 입력하세요. 2. 모든 서버에서 파일 시스템의 마운트 지점을 이전 경로에서 새 경로로 업데이트합니다. 이전에 준비 단계에서 설명한 재마운트할 때는 조직의 런북을 사용하세요. 3. 마운트를 확인하고 파일이 있는지 확인하여 파일 시스템이 제대로 마운트되고 액세스할 수 있는지 확인합니다. 인프라 팀은 일반적으로 이러한 활동을 수행합니다. 4. 애플리케이션을 재시작하고 애플리케이션 소유자 또는 QA 팀에 문의하여 애플리케이션에 필요한 경우 애플리케이션에 대한 기능 및 성능 테스트를 완료하세요. 	<p>AWS 시스템 관리자, 앱 소유자</p>

문제 해결

문제	Solution
Microsoft Excel의 셀 값은 업데이트되지 않습니다.	채우기 핸들을 드래그하여 샘플 행의 수식을 복사합니다. 자세한 내용은 Windows 또는 Mac용 지침 (Microsoft 지원 웹사이트)을 참조하세요.

관련 리소스

AWS 설명서

- [AWS DataSync 설명서](#)
- [Amazon EFS 설명서](#)
- [Amazon FSx 설명서](#)
- [AWS 클라우드로의 대규모 마이그레이션](#)
 - [AWS 대규모 마이그레이션 가이드](#)
 - [AWS 대규모 마이그레이션을 위한 포트폴리오 플레이북](#)

문제 해결

- [AWS DataSync 문제 해결](#)
- [Amazon EFS 문제 해결](#)
- [Amazon FSx for Windows File Server 문제 해결](#)
- [ONTAP용 Amazon NetApp FSx 문제 해결](#)

첨부

이 문서와 관련된 추가 콘텐츠에 액세스하려면 [attachment.zip](#) 파일의 압축을 풉니다.

Oracle 플랫폼 파일 어댑터를 사용하여 GoldenGate Oracle 데이터베이스를 Amazon RDS for Oracle로 마이그레이션합니다.

작성자: Dhairya Jindani(AWS) 및 Baji Shaik(AWS)

환경: PoC 또는 파일럿	소스: Oracle 데이터베이스(온프레미스 또는 EC2 인스턴스)	대상: Amazon RDS for Oracle
R 유형: 리플랫폼	워크로드: Oracle	기술: 마이그레이션, 분석, 데이터베이스
AWS 서비스: Amazon RDS		

요약

GoldenGate Oracle은 이기종 데이터베이스 및 IT 환경을 위한 실시간 데이터 캡처 및 복제 서비스입니다. 하지만 이 서비스는 현재 Amazon Relational Database Service(RDS) for Oracle을 지원하지 않습니다. 지원되는 데이터베이스 목록은 [이기종 GoldenGate 데이터베이스용 Oracle \(Oracle 설명서\)](#) 을 참조하십시오. 이 패턴은 Oracle GoldenGate 및 Oracle GoldenGate 플랫폼 파일 어댑터를 사용하여 소스 Oracle 데이터베이스에서 플랫폼 파일을 생성하는 방법을 설명합니다. 이 데이터베이스는 온프레미스 또는 Amazon Elastic Compute Cloud (Amazon EC2) 인스턴스에 있을 수 있습니다. 그런 다음 이러한 플랫폼 파일을 Amazon RDS for Oracle용 데이터베이스 인스턴스로 가져올 수 있습니다.

이 패턴에서는 GoldenGate Oracle을 사용하여 원본 Oracle 데이터베이스에서 트레일 파일을 추출합니다. 데이터 펌프는 트레일 파일을 통합 서버, 즉 EC2 인스턴스로 복사합니다. 통합 서버에서 GoldenGate Oracle은 플랫폼 파일 어댑터를 사용하여 트레일 파일의 트랜잭션 데이터 캡처를 기반으로 일련의 순차 플랫폼 파일을 생성합니다. Oracle은 데이터를 구분자로 구분된 값이나 길이로 구분된 값으로 GoldenGate 포맷합니다. 그런 다음 Oracle SQL *Loader를 사용하여 플랫폼 파일을 대상 Amazon RDS for Oracle 데이터베이스 인스턴스로 가져옵니다.

대상 고객

이 패턴은 Oracle의 기본 구성 요소에 대한 경험과 지식이 있는 사용자를 대상으로 합니다. GoldenGate 자세한 내용은 [Oracle GoldenGate 아키텍처 개요 \(Oracle 설명서\)](#) 를 참조하십시오.

사전 조건 및 제한 사항

사전 조건

- 활성 Amazon Web Services(AWS) 계정.
- 오라클 GoldenGate 라이선스.
- Oracle GoldenGate 어댑터에 대한 별도의 라이선스.
- 온프레미스 또는 EC2 인스턴스에서 실행되는 소스 Oracle 데이터베이스.
- 통합 서버로 사용되는 EC2 Linux 인스턴스. 자세한 내용은 [Amazon EC2 Linux 인스턴스 시 작](#)(Amazon EC2 설명서)를 참조하십시오.
- 대상 Amazon RDS for Oracle 데이터베이스 인스턴스. 자세한 내용은 [Oracle DB 인스턴스 생 설](#)(Amazon RDS 설명서)을 참조하십시오.

제품 버전

- Oracle Database Enterprise Edition 버전 10g, 11g, 12c 이상
- 오라클 GoldenGate 버전 12.2.0.1.1 이상

아키텍처

소스 기술 스택

Oracle 데이터베이스(온프레미스 또는 EC2 인스턴스)

대상 기술 스택

Amazon RDS for Oracle

소스 및 대상 아키텍처

1. Oracle은 원본 데이터베이스 로그에서 트레일을 GoldenGate 추출합니다.
2. 데이터 펌프는 트레일을 추출하여 통합 서버로 마이그레이션합니다.
3. Oracle GoldenGate 플랫폼 파일 어댑터는 트레일, 소스 정의 및 추출 매개변수를 읽습니다.
4. 추출을 종료하면 제어 파일과 플랫폼 데이터 파일이 생성됩니다.
5. 플랫폼 데이터 파일을 AWS 클라우드의 Amazon RDS for Oracle 데이터베이스 인스턴스로 마이그레이션합니다.

도구

서비스

- [Amazon Elastic Compute Cloud\(Amazon EC2\)](#)는 AWS 클라우드에서 확장 가능한 컴퓨팅 용량을 제공합니다. 필요한 만큼 가상 서버를 시작하고 빠르게 스케일 업하거나 스케일 다운할 수 있습니다.
- [Amazon Relational Database Service\(RDS\)](#)는 AWS 클라우드에서 Oracle 관계형 데이터베이스를 설정하고, 운영하고, 규모를 조정하도록 도와줍니다.

기타 서비스

- [GoldenGateOracle](#)은 한 데이터베이스의 데이터를 다른 이기종 데이터베이스 또는 다른 대상 토폴로지 (예: 플랫폼 파일) 로 복제, 필터링 및 변환하는 데 도움이 되는 서비스입니다.
- [Oracle GoldenGate 애플리케이션 어댑터를](#) 사용하면 GoldenGate Oracle은 소스 데이터베이스의 트레일 파일에 캡처된 트랜잭션 데이터로부터 일련의 순차적 플랫폼 파일 및 제어 파일을 생성할 수 있습니다. 이러한 어댑터는 데이터 웨어하우스 애플리케이션과 전용 또는 레거시 애플리케이션에서 추출, 전환, 적재(ETL) 작업에 널리 사용됩니다. GoldenGate Oracle은 이 캡처를 수행하여 이기종 데이터베이스, 플랫폼 및 운영 체제에 거의 실시간으로 적용합니다. 어댑터는 CSV 또는 Apache Parquet과 같은 다양한 출력 파일 형식을 지원합니다. 데이터를 다른 이기종 데이터베이스에 로드하기 위해 이렇게 생성된 파일을 로드할 수 있습니다.

에픽

원본 데이터베이스 GoldenGate 서버에 Oracle을 설정합니다.

작업	설명	필요한 기술
오라클을 다운로드하십시오 GoldenGate.	소스 데이터베이스 서버에 서 Oracle GoldenGate 버전 12.2.0.1.1 이상을 다운로드하십시오. 자세한 지침은 Oracle 다운로드 GoldenGate (Oracle 설명서) 를 참조하십시오.	DBA
오라클을 설치합니다 GoldenGate.	자세한 지침은 Oracle 설치 GoldenGate (Oracle 설명서) 를 참조하십시오.	DBA

작업	설명	필요한 기술
오라클을 설정하십시오 GoldenGate.	자세한 지침은 Oracle용 데이터베이스 준비 GoldenGate (Oracle 설명서) 를 참조하십시오.	DBA

통합 GoldenGate 서버에 Oracle을 설정합니다.

작업	설명	필요한 기술
오라클을 다운로드하십시오 GoldenGate.	통합 서버에서 오라클 GoldenGate 버전 12.2.0.1.1 이상을 다운로드하십시오. 자세한 지침은 Oracle 다운로드 GoldenGate (Oracle 설명서) 를 참조하십시오.	DBA
오라클을 설치합니다 GoldenGate.	디렉토리를 생성하고, 관리자 프로세스를 설정하고, 이 기종 환경을 위한 defgen 파일을 생성합니다. 자세한 지침은 Oracle 설치 GoldenGate (Oracle 설명서) 를 참조하십시오.	DBA

Oracle GoldenGate 데이터 캡처 구성을 변경하십시오.

작업	설명	필요한 기술
Oracle GoldenGate 어댑터를 준비하십시오.	통합 서버에서 Oracle GoldenGate 어댑터 소프트웨어를 설정합니다. 다음을 따릅니다.	DBA

작업	설명	필요한 기술
	<ol style="list-style-type: none"> 1. Oracle 소프트웨어 딜리버리 클라우드에서 ggs_Adapters_Linux_x64.zip 파일을 다운로드하십시오. 2. ggs_Adapters_Linux_x64.zip 압축을 풉니다. 3. 다음 명령을 실행하여 어댑터를 설치합니다. <pre style="border: 1px solid #ccc; border-radius: 10px; padding: 10px; margin: 10px 0;">tar -xvf ggs_Adapters_Linux_x64.tar</pre>	
<p>데이터 펌프를 구성하십시오.</p>	<p>소스 서버에서 데이터 펌프를 구성하여 소스 서버에서 통합 서버로 트레일 파일을 전송합니다. 데이터 펌프 파라미터 파일 및 트레일 파일 디렉토리를 생성합니다. 지침은 플랫 파일 어댑터 구성(Oracle 설명서)을 참조하십시오.</p>	<p>DBA</p>

플랫 파일 생성 및 마이그레이션

작업	설명	필요한 기술
<p>플랫 파일을 생성합니다.</p>	<p>추출 파일 및 제어 파일을 생성한 다음 통합 서버에서 추출 프로세스를 시작합니다. 그러면 데이터베이스 변경 내용이 추출되고 소스 데이터베이스가 플랫 파일에 기록됩니다. 지침은 플랫 파일 어댑터 사용(Oracle 설명서)을 참조하십시오.</p>	<p>DBA</p>

작업	설명	필요한 기술
플랫 파일을 대상 데이터베이스로 로드합니다.	플랫 파일을 대상 Amazon RDS for Oracle 데이터베이스 인스턴스로 가져옵니다. 자세한 내용은 Oracle SQL*Loader를 사용하여 가져오기 (Amazon RDS 설명서)를 참조하십시오.	DBA

문제 해결

문제	Solution
Oracle GoldenGate 플랫 파일 어댑터에서 오류가 발생합니다.	어댑터 오류에 대한 설명은 오류 메시지 찾기 (Oracle 설명서)를 참조하십시오. 문제 해결 지침은 플랫 파일 어댑터 문제 해결 (Oracle 설명서)을 참조하십시오.

관련 리소스

- [오라클 설치 GoldenGate](#) (오라클 설명서)
- [오라클 구성 GoldenGate](#) (오라클 설명서)
- [오라클 GoldenGate 어댑터 이해](#) (오라클 설명서)
- [플랫 파일 어댑터 구성](#)(Oracle 설명서)

Microsoft SQL Server에서 Amazon Aurora PostgreSQL-Compatible Edition으로 데이터베이스 마이그레이션을 지원하도록 Python 및 Perl 애플리케이션 변경

작성자: Dwarika Patra(AWS) 및 Deepesh Jayaprakash(AWS)

환경: PoC 또는 파일럿	소스: SQL Server	대상: Aurora PostgreSQL-Compatible
R 유형: 리플랫폼	워크로드: Microsoft, 오픈 소스	기술: 마이그레이션, 데이터베이스
AWS 서비스: Amazon Aurora		

요약

이 패턴은 Microsoft SQL Server에서 Amazon Aurora PostgreSQL-Compatible Edition으로 데이터베이스를 마이그레이션할 때 필요할 수 있는 애플리케이션 리포지토리의 변경 사항을 설명합니다. 이 패턴은 이러한 애플리케이션이 Python 기반 또는 Perl 기반이라고 가정하며 이러한 스크립팅 언어에 대한 별도의 지침을 제공합니다.

SQL Server 데이터베이스를 Aurora PostgreSQL-Compatible로 마이그레이션하는 데에는 스키마 변환, 데이터베이스 객체 변환, 데이터 마이그레이션 및 데이터 로드와 필요합니다. PostgreSQL과 SQL Server 간의 차이(데이터 유형, 연결 객체, 구문 및 로직 관련)로 인해 가장 어려운 마이그레이션 작업은 PostgreSQL에서 올바르게 작동하도록 코드 베이스를 필요에 따라 변경하는 것입니다.

Python 기반 애플리케이션의 경우 연결 객체와 클래스는 시스템 전체에 분산되어 있습니다. 또한 Python 코드 베이스는 여러 라이브러리를 사용하여 데이터베이스에 연결할 수 있습니다. 데이터베이스 연결 인터페이스를 변경하면 애플리케이션의 인라인 쿼리를 실행하는 객체도 변경해야 합니다.

Perl 기반 애플리케이션의 경우 변경 사항에는 연결 객체, 데이터베이스 연결 드라이버, 정적 및 동적 인라인 SQL 문, 애플리케이션이 복잡한 동적 DML 쿼리와 결과 세트를 처리하는 방식이 포함됩니다.

애플리케이션을 마이그레이션할 때 FTP 서버를 Amazon Simple Storage Service(S3) 액세스로 교체하는 등 AWS에서 가능한 개선 사항을 고려할 수도 있습니다.

애플리케이션 마이그레이션 프로세스에는 다음과 같은 문제가 포함됩니다.

- 연결 객체. 연결 객체가 여러 라이브러리 및 함수 호출을 포함하는 코드에 분산되어 있는 경우 PostgreSQL을 지원하도록 변경하는 일반적인 방법을 찾아야 할 수 있습니다.
- 레코드 검색 또는 업데이트 중 오류 또는 예외 처리. 데이터베이스에서 변수, 결과 세트 또는 데이터 프레임을 반환하는 조건부 생성, 읽기, 업데이트, 삭제(CRUD) 작업을 수행하는 경우 오류나 예외로 인해 애플리케이션 오류와 연쇄적인 효과가 발생할 수 있습니다. 이러한 오류는 적절한 검증과 저장 시점을 통해 신중하게 처리해야 합니다. 이러한 저장 시점 중 하나는 BEGIN...EXCEPTION...END 블록 내에서 대규모 인라인 SQL 쿼리 또는 데이터베이스 객체를 호출하는 것입니다.
- 트랜잭션 제어 및 유효성 검사. 여기에는 수동 및 자동 커밋과 롤백이 포함됩니다. Perl용 PostgreSQL 드라이버를 사용하려면 항상 자동 커밋 속성을 명시적으로 설정해야 합니다.
- 동적 SQL 쿼리 처리. 이를 위해서는 쿼리 로직에 대한 깊은 이해와 쿼리가 예상대로 작동하는지 확인하기 위한 반복 테스트가 필요합니다.
- 성능. 코드 변경으로 인해 애플리케이션 성능이 저하되지 않도록 해야 합니다.

이 패턴은 변환 프로세스를 자세히 설명합니다.

사전 조건 및 제한 사항

사전 조건

- Python 및 Perl 구문에 대한 실무 지식.
- SQL Server 및 PostgreSQL의 기본 기술.
- 기존 애플리케이션 아키텍처에 대한 이해.
- 애플리케이션 코드, SQL Server 데이터베이스 및 PostgreSQL 데이터베이스에 대한 액세스 권한.
- 애플리케이션 변경 사항을 개발, 테스트 및 검증하기 위한 보안 인증 정보를 통한 Windows 또는 Linux(또는 기타 Unix) 개발 환경에 대한 액세스 권한.
- Python 기반 애플리케이션의 경우 데이터 프레임을 처리하기 위한 Pandas, 데이터베이스 연결을 위한 psycopg2 또는 SQLAlchemy와 같은 애플리케이션에 필요할 수 있는 표준 Python 라이브러리 필요.
- Perl 기반 애플리케이션의 경우 종속 라이브러리 또는 모듈을 포함하는 Perl 패키지 필요. Comprehensive Perl Archive Network(CPAN) 모듈은 대부분의 애플리케이션 요구 사항을 지원할 수 있습니다.
- 필요한 모든 종속 사용자 지정 라이브러리 또는 모듈.
- SQL Server에 대한 읽기 권한 및 Aurora에 대한 읽기/쓰기 권한을 위한 데이터베이스 보안 인증 정보.
- 서비스 및 사용자를 통해 애플리케이션 변경 사항을 검증하고 디버깅하기 위한 PostgreSQL.

- Visual Studio Code, Sublime Text 또는 pgAdmin과 같은 애플리케이션 마이그레이션 중 개발 도구에 대한 액세스 권한.

제한 사항

- 일부 Python 또는 Perl 버전, 모듈, 라이브러리 및 패키지는 클라우드 환경과 호환되지 않습니다.
- SQL Server에 사용되는 일부 타사 라이브러리 및 프레임워크는 PostgreSQL 마이그레이션을 지원하지하도록 대체할 수 없습니다.
- 성능 변화에 따라 애플리케이션, 인라인 Transact-SQL(T-SQL) 쿼리, 데이터베이스 함수 및 저장 프로시저를 변경해야 할 수도 있습니다.
- PostgreSQL은 테이블 이름, 열 이름 및 기타 데이터베이스 객체에 소문자 이름을 지원합니다.
- UUID 열과 같은 일부 데이터 유형은 소문자로만 저장됩니다. Python 및 Perl 애플리케이션은 이러한 대소문자 차이를 처리해야 합니다.
- 문자 인코딩 차이는 PostgreSQL 데이터베이스의 해당 텍스트 열에 대해 올바른 데이터 유형으로 처리해야 합니다.

제품 버전

- Python 3.6 이상(사용자 운영 체제를 지원하는 버전 사용)
- Perl 5.8.3 이상(사용자 운영 체제를 지원하는 버전 사용)
- Aurora PostgreSQL-Compatible Edition 4.2 이상([세부 정보 참조](#))

아키텍처

소스 기술 스택

- 스크립팅(애플리케이션 프로그래밍) 언어: Python 2.7 이상 또는 Perl 5.8
- 데이터베이스: Microsoft SQL Server 버전 13
- 운영 체제: Red Hat Enterprise Linux(RHEL) 7

대상 기술 스택

- 스크립팅(애플리케이션 프로그래밍) 언어: Python 3.6 이상 또는 Perl 5.8 이상
- 데이터베이스: Aurora PostgreSQL-Compatible 4.2
- 운영 체제: RHEL 7

마이그레이션 아키텍처

도구

AWS 서비스 및 도구

- [Aurora PostgreSQL-Compatible Edition](#)은 하이엔드 상용 데이터베이스의 속도와 안정성에 오픈 소스 데이터베이스의 비용 효율성을 결합한 완전 관리형 PostgreSQL 호환 및 ACID 호환 관계형 데이터베이스 엔진입니다. Aurora PostgreSQL은 PostgreSQL을 대체하는 기능으로 신규 및 기존 PostgreSQL 배포를 간편하고 비용 효율적으로 설정, 운영 및 확장할 수 있습니다.
- [AWS Command Line Interface\(AWS CLI\)](#)는 명령줄 셸의 명령을 사용하여 AWS 서비스와 상호 작용할 수 있는 오픈 소스 도구입니다.

기타 도구

- [psycopg2](#) 및 [SQLAlchemy](#)와 같은 [Python](#) 및 PostgreSQL 데이터베이스 연결 라이브러리
- [Perl](#) 및 해당 [DBI 모듈](#)
- [PostgreSQL 대화식 터미널\(psql\)](#)

에픽

애플리케이션 리포지토리를 PostgreSQL로 마이그레이션-고급 단계

작업	설명	필요한 기술
다음 코드 변환 단계에 따라 애플리케이션을 PostgreSQL로 마이그레이션합니다.	<ol style="list-style-type: none"> 1. PostgreSQL용 데이터베이스 관련 ODBC 드라이버 및 라이브러리를 설정합니다. 예를 들어 Perl의 CPAN 모듈 및 Python의 pyodbc, psycopg2 또는 SQLAlchemy 중 하나를 사용할 수 있습니다. 2. 이러한 라이브러리를 사용하여 Aurora PostgreSQL-Compatible에 연결함으로써 	앱 개발자

작업	설명	필요한 기술
	<p>데이터베이스 객체를 변환합니다.</p> <ol style="list-style-type: none"> 3. 기존 애플리케이션 모듈에서 코드 변경 사항을 적용하여 호환되는 T-SQL 문을 가져옵니다. 4. 애플리케이션 코드에 데이터베이스 관련 함수 호출 및 저장 프로시저를 다시 작성합니다. 5. 인라인 SQL 쿼리에 사용되는 애플리케이션 변수 및 데이터 유형에 대한 변경 사항을 처리합니다. 6. 호환되지 않는 데이터베이스 관련 함수를 처리합니다. 7. 데이터베이스 마이그레이션을 위해 변환된 애플리케이션 코드의 전체 end-to-end 테스트 8. Microsoft SQL Server의 결과를 PostgreSQL로 마이그레이션한 애플리케이션과 비교합니다. 9. Microsoft SQL Server와 PostgreSQL 간에 애플리케이션 성능 벤치마킹을 수행합니다. 10. 애플리케이션에서 호출한 저장 프로시저 또는 인라인 T-SQL 문을 수정하여 성능을 개선합니다. 	

작업	설명	필요한 기술
	다음 에픽은 Python 및 Perl 애플리케이션의 이러한 변환 작업 중 일부에 대한 자세한 지침을 제공합니다.	

작업	설명	필요한 기술
<p>마이그레이션 각 단계마다 체크리스트를 사용합니다.</p>	<p>최종 단계를 포함하여 애플리케이션 마이그레이션의 각 단계에 대한 체크리스트에 다음을 추가합니다.</p> <ul style="list-style-type: none"> • PostgreSQL 설명서를 검토하여 모든 변경 사항이 PostgreSQL 표준과 호환되는지 확인합니다. • 열에 정수 및 부동소수점 값이 있는지 확인합니다. • 열 이름 및 날짜/시간 스탬프와 함께 삽입, 업데이트 및 추출된 행 수를 식별합니다. diff 유틸리티를 사용하거나 스크립트를 작성하여 이러한 검사를 자동화할 수 있습니다. • 대규모 인라인 SQL 문에 대한 성능 검사를 완료하고 애플리케이션의 전체 성능을 확인합니다. • try/catch 블록을 여러 개 사용하여 데이터베이스 작업에 대한 오류 처리가 올바른지 확인하고 프로그램이 정상적으로 종료되는지 확인합니다. • 로깅 프로세스가 제대로 되어 있는지 확인합니다. 	<p>앱 개발자</p>

애플리케이션 분석 및 업데이트-Python 코드 베이스

작업	설명	필요한 기술
<p>기존 Python 코드 베이스를 분석합니다.</p>	<p>분석에는 애플리케이션 마이그레이션 프로세스를 용이하게 하는 다음 내용이 포함되어야 합니다.</p> <ul style="list-style-type: none"> • 코드의 모든 연결 객체를 식별합니다. • 호환되지 않는 모든 인라인 SQL 쿼리(예: T-SQL 문 및 저장 프로시저)를 식별하고 필요한 변경 사항을 분석합니다. • 코드 설명서를 검토하고 제어 흐름을 추적하여 코드 기능을 이해합니다. 나중에 애플리케이션의 성능 또는 로드 비교를 테스트할 때 유용합니다. • 데이터베이스 변환 후 효과적으로 테스트할 수 있도록 애플리케이션의 용도를 이해합니다. 데이터베이스 마이그레이션을 통한 변환 대상이 되는 대부분의 Python 애플리케이션은 다른 소스에서 데이터베이스 테이블로 데이터를 로드하는 피드이거나, 또는 테이블에서 데이터를 검색하여 보고서를 생성하거나 검증을 수행하기 위한 API 호출에 적합한 다양한 출력 형식(예: CSV, JSON 	<p>앱 개발자</p>

작업	설명	필요한 기술
	또는 플랫폼 파일)으로 변환하는 추출기입니다.	

작업	설명	필요한 기술
<p>PostgreSQL을 지원하도록 데이터베이스 연결을 변환합니다.</p>	<p>대부분의 Python 애플리케이션은 다음과 같이 pyodbc 라이브러리를 사용하여 SQL Server 데이터베이스에 연결합니다.</p> <pre data-bbox="597 443 1027 1356"> import pyodbc try: conn_string = "Driver=ODBC Driver 17 for SQL Server;UID={};PWD= {};Server={};Datab ase={}".format (conn_user, conn_pass word, conn_server, conn_database) conn = pyodbc.co nnect(conn_string) cur = conn.cursor() result = cur.execu te(query_string) for row in result: print (row) except Exception as e: print(str(e)) </pre> <p>다음과 같이 PostgreSQL을 지원하도록 데이터베이스 연결을 변환합니다.</p> <pre data-bbox="597 1562 1027 1854"> import pyodbc import psycopg2 try: conn_string = 'postgresql+psycop g2://'+ </pre>	<p>앱 개발자</p>

작업	설명	필요한 기술
	<pre> conn_user+' ':'+conn _password+'@'+conn _server+'/' +conn_d atabase conn = pyodbc.co nnect(conn_string, connect_args={'opt ions': '-csearch_pa th=dbo'}) cur = conn.cursor() result = cur.execu te(query_string) for row in result: print (row) except Exception as e: print(str(e)) </pre>	

작업	설명	필요한 기술
<p>인라인 SQL 쿼리를 PostgreSQL로 변경합니다.</p>	<p>인라인 SQL 쿼리를 PostgreSQL 호환 형식으로 변환합니다. 예를 들어 다음 SQL Server 쿼리는 테이블에서 문자열을 검색합니다.</p> <pre data-bbox="594 489 1027 1360">dtype = "type1" stm = '''SELECT TOP 1 searchcode FROM TypesTable (NOLOCK) WHERE code=''' + ''' + str(dtype) + ''' # For Microsoft SQL Server Database Connection engine = create_en gine('mssql+pyodbc :///odbc_connect=%s' % urllib.parse.quote _plus(conn_string) , connect_args={'con nect_timeout':logi n_timeout}) conn = engine_connect() rs = conn.execute(stm) for row in rs: print(row)</pre> <p>변환 후 PostgreSQL 호환 인라인 SQL 쿼리는 다음과 같습니다.</p> <pre data-bbox="594 1570 1027 1860">dtype = "type1" stm = '''SELECT searchcode FROM TypesTable WHERE code=''' + ''' + str(dtype) + ''' LIMIT 1'''</pre>	<p>앱 개발자</p>

작업	설명	필요한 기술
	<pre># For PostgreSQL Database Connection engine = create_engine('postgres+psycopy2://%s' %conn_string, connect_args={'connect_timeout':login_timeout}) conn = engine.connect() rs = conn.execute(stm) for row in rs: print(row)</pre>	

작업	설명	필요한 기술
<p>동적 SQL 쿼리를 처리합니다.</p>	<p>동적 SQL은 하나의 스크립트 또는 여러 Python 스크립트에 존재할 수 있습니다. 이전 예제에서는 Python의 문자열 대체 함수를 사용하여 동적 SQL 쿼리를 구성하기 위한 변수를 삽입하는 방법을 보여 주었습니다. 또 다른 접근 방식은 해당하는 경우 쿼리 문자열에 변수를 추가하는 것입니다.</p> <p>다음 예제에서는 함수가 반환한 값에 따라 쿼리 문자열을 즉시 구성합니다.</p> <pre data-bbox="597 905 1024 1220"> query = "SELECT id from equity e join issues i on e.permId=i.permId where e.id" query += get_id_filter(ids) + " e.id is NOT NULL </pre> <p>이러한 유형의 동적 쿼리는 애플리케이션 마이그레이션 중에 매우 일반적입니다. 동적 쿼리를 처리하려면 다음 단계를 따르세요.</p> <ul data-bbox="597 1528 1024 1816" style="list-style-type: none"> • 전체 구문(예: JOIN 절이 있는 SELECT 문의 구문)을 확인합니다. • 쿼리에 사용된 모든 변수 또는 열 이름(예: i 및 id)을 확인합니다. 	<p>앱 개발자</p>

작업	설명	필요한 기술
	<ul style="list-style-type: none">• 쿼리에 사용된 함수, 인수 및 반환 값(예: <code>get_id_filter</code> 및 그 인수 <code>ids</code>)을 확인합니다.	

작업	설명	필요한 기술
<p>결과 세트, 변수 및 데이터 프레임 처리합니다.</p>	<p>Microsoft SQL Server의 경우 <code>fetchone()</code> 또는 <code>fetchall()</code> 과 같은 Python 메서드를 사용하여 데이터베이스에서 결과 세트를 검색합니다. <code>fetchmany(size)</code> 를 사용하여 결과 세트에서 반환할 레코드 수를 지정할 수도 있습니다. 이를 위해 다음 예제와 같이 <code>pyodbc</code> 연결 객체를 사용할 수 있습니다.</p> <p><code>pyodbc</code> (Microsoft SQL Server)</p> <pre data-bbox="592 903 1031 1873"> import pyodbc server = 'tcp:myserver.database.windows.net' database = 'exampledb' username = 'exampleuser' password = 'examplepassword' conn = pyodbc.connect('DRIVER={ODBC Driver 17 for SQL Server};SERVER='+server+';DATABASE='+database+';UID='+username+';PWD='+password) cursor = conn.cursor() cursor.execute("SELECT * FROM ITEMS") row = cursor.fetchone() while row: print(row[0]) </pre>	<p>앱 개발자</p>

작업	설명	필요한 기술
	<pre data-bbox="592 210 1031 304">row = cursor.fetchone()</pre> <p data-bbox="592 346 1031 808">Aurora에서는 PostgreSQL에 연결하고 결과 세트를 가져오는 것과 같은 유사한 작업을 수행하기 위해 psycopg2 또는 SQLAlchemy를 사용할 수 있습니다. 이러한 Python 라이브러리는 다음 예제와 같이 PostgreSQL 데이터베이스 레코드를 탐색할 수 있는 연결 모듈과 커서 객체를 제공합니다.</p> <p data-bbox="592 850 1031 934">psycopg2 (Aurora PostgreSQL-Compatible)</p> <pre data-bbox="592 976 1031 1848">import psycopg2 query = "SELECT * FROM ITEMS;" //Initialize variables host=dbname=user=password=port=sslmode=connect_timeout="" connstring = "host='{host}' dbname='{dbname}' user='{user}' \ password='{password}' port='{port}' ".format(host=host, dbname=dbname, user=user, password=password, port=port) conn = psycopg2.connect(connstring) cursor = conn.cursor()</pre>	

작업	설명	필요한 기술
	<pre> cursor.execute(query) column_names = [column[0] for column in cursor.description] print("Column Names: ", column_names) print("Column values: " for row in cursor: print("itemid :", row[0]) print("itemdescript ion :", row[1]) print("it emprice :", row[3])) </pre> <p>SQLAlchemy (Aurora PostgreSQL-Compatible)</p> <pre> from sqlalchemy import create_engine from pandas import DataFrame conn_string = 'postgres ql://core:database @localhost:5432/ex ampledatabase' engine = create_en gine(conn_string) conn = engine.co nnect() dataid = 1001 result = conn.exec ute("SELECT * FROM ITEMS") df = DataFrame (result.fetchall()) df.columns = result.ke ys() df = pd.DataFrame() engine.connect() </pre>	

작업	설명	필요한 기술
	<pre>df = pd.read_sql_query(sql_query, engine, coerce_float=False) print("df=", df)</pre>	
<p>마이그레이션 도중과 마이그레이션 후에 애플리케이션을 테스트합니다.</p>	<p>마이그레이션된 Python 애플리케이션을 테스트하는 것은 지속적인 프로세스입니다. 마이그레이션에는 연결 객체 변경 (psycopg2 또는 SQLAlchemy), 오류 처리, 새 기능(데이터 프레임), 인라인 SQL 변경, 대량 복사 기능(COPY 대신 bcp) 및 유사한 변경 사항이 포함되므로 애플리케이션 마이그레이션 도중 및 후에 신중하게 테스트해야 합니다. 다음을 확인합니다.</p> <ul style="list-style-type: none"> 오류 조건 및 처리 마이그레이션 후 레코드 불일치 레코드 업데이트 또는 삭제 애플리케이션 실행에 필요한 시간 	<p>앱 개발자</p>

애플리케이션 분석 및 업데이트-Perl 코드 베이스

작업	설명	필요한 기술
<p>기존 Perl 코드베이스를 분석합니다.</p>	<p>분석에는 애플리케이션 마이그레이션 프로세스를 용이하게 하는 다음 내용이 포함되어야</p>	<p>앱 개발자</p>

작업	설명	필요한 기술
	<p>합니다. 다음을 식별해야 합니다.</p> <ul style="list-style-type: none"> • 모든 INI 또는 구성 기반 코드 • 데이터베이스 관련 표준 Open Database Connectivity(ODBC) Perl 드라이버 또는 기타 사용자 지정 드라이버 • 인라인 및 T-SQL 쿼리에 필요한 코드 변경 • 다양한 Perl 모듈 간 상호 작용(예: 여러 기능 구성 요소에서 호출하거나 사용하는 단일 Perl ODBC 연결 객체) • 데이터 세트 및 결과 세트 처리 • 외부, 종속 Perl 라이브러리 • 애플리케이션에서 사용되는 모든 API • Aurora PostgreSQL-Compatible과 호환되는 Perl 버전 호환성 및 드라이버 호환성 	

작업	설명	필요한 기술
<p>PostgreSQL을 지원하도록 Perl 애플리케이션과 DBI 모듈의 연결을 변환합니다.</p>	<p>Perl 기반 애플리케이션은 일반적으로 Perl 프로그래밍 언어의 표준 데이터베이스 액세스 모듈인 Perl DBI 모듈을 사용합니다. SQL Server 및 PostgreSQL의 드라이버가 다른 동일한 DBI 모듈을 사용할 수 있습니다.</p> <p>필수 Perl 모듈, 설치 및 기타 지침에 대한 자세한 내용은 DBD::Pg 설명서를 참조하세요. 다음 예제는 <code>exampletest-aurorapg-database.cluster-sampleclusture.us-east-.rds.amazonaws.com</code> 에서 Aurora PostgreSQL-Compatible에 연결합니다.</p> <pre data-bbox="597 1144 1027 1833"> #!/usr/bin/perl use DBI; use strict; my \$driver = "Pg"; my \$hostname = "exampletest-aurorapg-database-sampleclusture.us-east.rds.amazonaws.com" my \$dsn = "DBI:\$driver:dbname = \$hostname;host = 127.0.0.1;port = 5432"; my \$username = "postgres"; my \$password = "pass123"; </pre>	<p>앱 개발자</p>

작업	설명	필요한 기술
	<pre data-bbox="597 205 1026 665">\$dbh = DBI->connect("dbi:Pg:dbname=\$hostname;host=\$hostname;port=\$port;options=\$options", \$username, \$password, {AutoCommit => 0, RaiseError => 1, PrintError => 0});</pre>	

작업	설명	필요한 기술
<p>인라인 SQL 쿼리를 PostgreSQL로 변경합니다.</p>	<p>애플리케이션에 SELECT, DELETE, UPDATE, 및 PostgreSQL에서 지원하지 않는 쿼리 절을 포함하는 유사한 문을 포함하는 인라인 SQL 쿼리가 있을 수 있습니다. 예를 들어 TOP 및 NOLOCK과 같은 쿼리 키워드는 PostgreSQL에서 지원되지 않습니다. 다음 예는 TOP, NOLOCK 및 부울 변수를 처리할 수 있는 방법을 보여 줍니다.</p> <p>SQL Server에서:</p> <pre data-bbox="594 905 1029 1381"> \$sqlStr = \$sqlStr . "WHERE a.student _id in (SELECT TOP \$numofRecords c_student_id \ FROM active_student_rec ord b WITH (NOLOCK) \ INNER JOIN student_c ontributor c WITH (NOLOCK) on c.contrib utor_id = b.c_st) </pre> <p>PostgreSQL의 경우 다음과 같이 변환합니다.</p> <pre data-bbox="594 1541 1029 1824"> \$sqlStr = \$sqlStr . "WHERE a.student _id in (SELECT TOP \$numofRecords c_student_id \ FROM active_student_rec ord b INNER JOIN </pre>	<p>앱 개발자</p>

작업	설명	필요한 기술
	<pre>student_contributor c \ on c.contributor_id = b.c_student_contr_id WHERE b_current_1 is true \ LIMIT \$numofRecords)"</pre>	

작업	설명	필요한 기술
<p>동적 SQL 쿼리와 Perl 변수를 처리합니다.</p>	<p>동적 SQL 쿼리는 애플리케이션 런타임 시 빌드되는 SQL 문입니다. 이러한 쿼리는 특정 조건에 따라 애플리케이션이 실행 중일 때 동적으로 구성되므로 런타임까지 쿼리의 전체 텍스트를 알 수 없습니다. 예를 들면 상위 10개 주식을 매일 분석하는 재무 분석 애플리케이션이 있으며, 이러한 주식은 매일 바뀝니다. SQL 테이블은 최고 성과자에 따라 생성되며 런타임까지 값을 알 수 없습니다.</p> <p>이 예제의 인라인 SQL 쿼리를 래퍼 함수에 전달하여 변수에 결과 세트를 가져온 다음 변수가 조건을 사용하여 테이블이 존재하는지 여부를 판별한다고 가정해 보겠습니다.</p> <ul style="list-style-type: none"> 테이블이 존재한다면 생성하지 않고 몇 가지 처리를 수행합니다. 테이블이 존재하지 않는 경우 테이블을 생성하고 몇 가지 처리도 수행합니다. <p>다음은 변수 처리의 예와 이 사용 사례에 대한 SQL Server 및 PostgreSQL 쿼리입니다.</p> <pre>my \$tableexists = db_read(arg 1,</pre>	<p>앱 개발자</p>

작업	설명	필요한 기술
	<pre> \$sql_qry, undef, 'writer'); my \$table_already_exists = \$tableexists->[0]{table_exists}; if (\$table_already_exists){ # do some thing } else { # do something else } </pre> <p>SQL Server:</p> <pre> my \$sql_qry = "SELECT OBJECT_ID('\$backen dTable', 'U') table_exists", undef, 'writer') "; </pre> <p>PostgreSQL:</p> <pre> my \$sql_qry = "SELECT TO_REGCLASS('\$back endTable', 'U') table_exists", undef, 'writer')"; </pre> <p>다음 예제에서는 JOIN과 함께 SELECT 문을 실행하여 테이블의 프라이머리 키와 키 열의 위치를 가져 오는 인라인 SQL의 Perl 변수를 사용합니다.</p> <p>SQL Server:</p> <pre> my \$sql_qry = "SELECT column_name', </pre>	

작업	설명	필요한 기술
	<pre> character_maximum_length \ FROM INFORMATION_SCHEMA .COLUMNS \ WHERE TABLE_SCHEMA= '\$example_schema' \ AND TABLE_NAME='\$example_table' \ AND DATA_TYPE IN ('varchar','nvarchar');"; </pre> <p>PostgreSQL:</p> <pre> my \$sql_qry = "SELECT c1.column_name, c1.ordinal_position \ FROM information_schema .key_column_usage AS c LEFT \ JOIN information_schema .table_constraints AS t1 \ ON t1.constraint_name = c1.constraint_name \ WHERE t1.table_name = \$example_schemaInf o.'\$example_table' \ AND t1.constraint_type = 'PRIMARY KEY' ";"; </pre>	

PostgreSQL을 지원하도록 Perl 기반 또는 Python 기반 애플리케이션 추가 변경

작업	설명	필요한 기술
<p>추가 SQL Server 구문을 PostgreSQL로 변환합니다.</p>	<p>다음 변경 사항은 프로그래밍 언어에 상관없이 모든 애플리케이션에 적용됩니다.</p> <ul style="list-style-type: none"> • 애플리케이션에서 사용하는 데이터베이스 객체를 적절한 새 스키마 이름으로 한정합니다. • PostgreSQL의 비교 기능과 대소문자 구분 일치성을 위해 LIKE 연산자를 처리합니다. • DATEDIFF, DATEADD, GETDATE, CONVERT, CAST 연산자 등 지원되지 않는 데이터베이스 관련 함수를 처리합니다. 동등한 PostgreSQL 호환 함수는 추가 정보 섹션의 기본 또는 내장 SQL 함수를 참조하세요. • 비교문에서 부울 값을 처리합니다. • 함수의 반환값을 처리합니다. 이러한 값은 레코드 세트, 데이터 프레임, 변수, 부울 값일 수 있습니다. 애플리케이션의 요구 사항에 따라 이를 처리하고 PostgreSQL을 지원합니다. • 새로운 사용자 정의 PostgreSQL 함수를 사용하여 익명 블록(예:BEGIN TRAN)을 처리합니다. 	<p>앱 개발자</p>

작업	설명	필요한 기술
	<ul style="list-style-type: none"> • 행에 대한 대량 삽입을 변환합니다. 애플리케이션 내에서 호출되는 SQL Server 대량 복사(bcp) 유틸리티와 동일한 PostgreSQL 유틸리티는 COPY입니다. • 열 연결 연산자를 변환합니다. SQL Server는 문자열 연결에 +를 사용하지만 PostgreSQL은 을 사용합니다. 	

성능 개선

작업	설명	필요한 기술
<p>AWS 서비스를 활용하여 성능을 개선합니다.</p>	<p>AWS 클라우드로 마이그레이션하면 애플리케이션 및 데이터베이스 설계를 개선하여 AWS 서비스를 활용할 수 있습니다. 예를 들어 Aurora PostgreSQL-Compatible 데이터베이스 서버에 연결된 Python 애플리케이션의 쿼리가 원래 Microsoft SQL Server 쿼리보다 시간이 더 많이 걸리는 경우 Aurora 서버에서 Amazon Simple Storage Service(S3) 버킷으로 직접 기록 데이터 피드를 생성하고 Amazon Athena 기반 SQL 쿼리를 사용하여 보고서를 생성하고 사용자 대시보드에 대한 데이터 쿼리를 분</p>	<p>앱 개발자, 클라우드 아키텍트</p>

작업	설명	필요한 기술
	석하는 것을 고려할 수 있습니다.	

관련 리소스

- [Perl](#)
- [Perl DBI 모듈](#)
- [Python](#)
- [psycopg2](#)
- [SQLAlchemy](#)
- [대량 복사-PostgreSQL](#)
- [대량 복사-Microsoft SQL Server](#)
- [PostgreSQL](#)
- [Amazon Aurora PostgreSQL 작업](#)

추가 정보

Microsoft SQL Server와 Aurora PostgreSQL-Compatible은 모두 ANSI SQL 규격입니다. 하지만 Python 또는 Perl 애플리케이션을 SQL Server에서 PostgreSQL로 마이그레이션할 때는 여전히 구문, 열 데이터 유형, 기본 데이터베이스 관련 함수, 대량 삽입 및 대/소문자 구분에서 호환되지 않는 문제를 알고 있어야 합니다.

다음 섹션에서는 발생할 수 있는 불일치에 대한 자세한 내용을 제공합니다.

데이터 유형 비교

SQL Server에서 PostgreSQL로 데이터 유형을 변경하면 애플리케이션이 작동하는 결과 데이터가 크게 달라질 수 있습니다. 데이터 유형을 비교하려면 [Sqlines 웹 사이트](#)의 표를 참조하세요.

기본 또는 내장 SQL 함수

일부 함수의 동작은 SQL Server와 PostgreSQL 데이터베이스 간에 다릅니다. 다음 표에 비교가 나와 있습니다.

Microsoft SQL Server	설명	PostgreSQL
CAST	값을 한 데이터 형식에서 다른 형식으로 변환합니다.	PostgreSQL type :: operator
GETDATE()	현재 데이터베이스 시스템 날짜 및 시간을 YYYY-MM-DD hh:mm:ss.mmm 형식으로 반환합니다.	CLOCK_TIMESTAMP
DATEADD	날짜에 시간/날짜 간격을 추가합니다.	INTERVAL 표현
CONVERT	값을 특정 데이터 형식으로 변환합니다.	TO_CHAR
DATEDIFF	두 날짜의 차이를 반환합니다.	DATE_PART
TOP	SELECT 결과 세트에서 행의 수를 제한합니다.	LIMIT/FETCH

익명 블록

구조화된 SQL 쿼리는 선언, 실행 파일, 예외 처리와 같은 섹션으로 구성됩니다. 다음 표는 간단한 익명 블록의 Microsoft SQL Server와 PostgreSQL 버전을 비교합니다. 복잡한 익명 블록의 경우 애플리케이션 내에서 사용자 지정 데이터베이스 함수를 호출하는 것이 좋습니다.

Microsoft SQL Server

```
my $sql_qry1=
my $sql_qry2 =
my $sqlqry = "BEGIN TRAN
$sql_qry1 $sql_qry2
if @@error !=0 ROLLBACK
TRAN
else COMIT TRAN";
```

PostgreSQL

```
my $sql_qry1=
my $sql_qry2 =
my $sql_qry = " DO \$$
BEGIN
$header_sql $content_sql
END
\$$";
```

기타 차이점

- 행 대량 삽입: [Microsoft SQL Server bcp 유틸리티](#)에 해당하는 PostgreSQL 유틸리티는 [COPY](#)입니다.
- 대소문자 구분: PostgreSQL에서는 열 이름이 대소문자를 구분하므로 SQL Server 열 이름을 소문자나 대문자로 변환해야 합니다. 이는 데이터를 추출 또는 비교하거나 결과 세트 또는 변수에 열 이름을 배치할 때 요인이 됩니다. 다음 예제에서는 값이 대문자 또는 소문자로 저장될 수 있는 열을 식별합니다.

```
my $sql_qry = "SELECT $record_id FROM $exampleTable WHERE LOWER($record_name) = \failed transaction\"";
```

- 연결: SQL Server는 문자열 연결에 +를 연산자로 사용하는 반면 PostgreSQL은 ||을 사용합니다.
- 검증: 인라인 SQL 쿼리 및 함수를 PostgreSQL용 애플리케이션 코드에서 사용하기 전에 먼저 테스트하고 검증해야 합니다.
- ORM 라이브러리 포함: 기존 데이터베이스 연결 라이브러리를 [SQLAlchemy](#) 및 [PynomoDB](#)와 같은 Python ORM 라이브러리로 포함하거나 대체할 수도 있습니다. 이를 통해 객체 지향 패러다임을 사용하여 데이터베이스의 데이터를 쉽게 쿼리하고 조작할 수 있습니다.

워크로드별 마이그레이션 패턴

주제

- [IBM](#)
- [Microsoft](#)
- [해당 사항 없음](#)
- [오픈 소스](#)
- [Oracle](#)
- [SAP](#)

IBM

- [AWS DMS를 사용하여 Db2 데이터베이스를 Amazon EC2에서 Aurora MySQL과 호환되는 Aurora로 마이그레이션](#)
- [로그 전송을 사용하여 운영 중단 시간을 줄임으로써 Db2 for LUW를 Amazon EC2로 마이그레이션합니다.](#)
- [고가용성 재해 복구 기능을 갖춘 Db2 for LUW를 Amazon EC2로 마이그레이션하세요.](#)
- [AWS DMS 및 AWS SCT를 사용하여 Amazon EC2의 IBM Db2에서 PostgreSQL과 호환되는 Aurora PostgreSQL로 마이그레이션하십시오.](#)
- [IBM WebSphere 애플리케이션 서버에서 Amazon EC2의 아파치 톰캣으로 마이그레이션](#)

Microsoft

- [Microsoft 워크로드의 검색 및 AWS로의 마이그레이션을 가속화합니다.](#)
- [Microsoft SQL Server에서 Amazon Aurora PostgreSQL-Compatible Edition으로 데이터베이스 마이그레이션을 지원하도록 Python 및 Perl 애플리케이션 변경](#)
- [마이크로소프트 엑셀과 Python을 사용하여 AWS DMS 작업을 위한 AWS CloudFormation 템플릿을 생성합니다.](#)
- [AWS DMS를 사용하여 Microsoft SQL Server 데이터베이스를 Amazon S3로 내보내기](#)
- [EC2 Windows 인스턴스를 수집하여 AWS Managed Services 계정으로 마이그레이션합니다.](#)
- [메시지 대기열을 Microsoft Azure 서비스 버스에서 Amazon SQS로 마이그레이션](#)
- [AWS DMS를 사용하여 Microsoft SQL 서버 데이터베이스를 Amazon EC2에서 Amazon DocumentDB로 마이그레이션](#)
- [AWS DMS와 AWS SCT를 사용하여 Microsoft SQL Server 데이터베이스를 Aurora MySQL로 마이그레이션](#)
- [Microsoft Azure 앱 서비스의 .NET 애플리케이션을 AWS Elastic Beanstalk로 마이그레이션합니다.](#)
- [온프레미스 Microsoft SQL Server 데이터베이스를 Amazon EC2로 마이그레이션](#)
- [온프레미스 Microsoft SQL Server 데이터베이스를 Amazon RDS for SQL Server로 마이그레이션](#)
- [연결된 서버를 사용하여 온프레미스 Microsoft SQL Server 데이터베이스를 Amazon RDS for SQL Server로 마이그레이션](#)
- [기본 백업 및 복원 수단을 사용하여 온프레미스 Microsoft SQL Server 데이터베이스를 Amazon RDS for SQL Server로 마이그레이션](#)
- [AWS DMS를 사용하여 온프레미스 Microsoft SQL Server 데이터베이스를 Amazon Redshift로 마이그레이션](#)
- [SCT 데이터 추출 에이전트를 사용하여 온프레미스 Microsoft SQL Server 데이터베이스를 Amazon Redshift로 마이그레이션](#)
- [Linux가 실행되는 Amazon EC2의 Microsoft SQL Server로 온프레미스 Microsoft SQL Server 데이터베이스의 마이그레이션](#)
- [Rclone를 사용하여 Microsoft Azure Blob에서 Amazon S3로 데이터 마이그레이션하기](#)
- [ACM을 사용하여 Windows SSL 인증서를 Application Load Balancer로 마이그레이션](#)
- [Amazon FSx를 사용하여 SQL Server Always On FCI용 다중 AZ 인프라 설정](#)

해당 사항 없음

- [AWS로 리호스팅 마이그레이션하는 동안 방화벽 요청에 대한 승인 프로세스를 생성](#)

오픈 소스

- [Aurora PostgreSQL 호환에서 애플리케이션 사용자 및 역할을 생성](#)
- [온프레미스 MariaDB 데이터베이스를 기본 도구를 사용하여 Amazon RDS for MariaDB로 마이그레이션](#)
- [온프레미스 MySQL 데이터베이스를 Amazon EC2로 마이그레이션](#)
- [온프레미스 MySQL 데이터베이스를 Amazon RDS for MySQL로 마이그레이션](#)
- [온프레미스 MySQL 데이터베이스를 Aurora MySQL로 마이그레이션](#)
- [온프레미스 PostgreSQL 데이터베이스를 Aurora PostgreSQL로 마이그레이션하기](#)
- [Auto Scaling을 사용하여 IBM WebSphere 애플리케이션 서버에서 Amazon EC2의 아파치 톰캣으로 마이그레이션하세요](#)
- [오라클에서 AWS Elastic GlassFish Beanstalk로 마이그레이션](#)
- [pglogical을 사용하여 Amazon EC2의 PostgreSQL에서 Amazon RDS for PostgreSQL로 마이그레이션합니다.](#)
- [AWS App2Container를 사용하여 온프레미스 Java 애플리케이션을 AWS로 마이그레이션](#)
- [페르코나, Amazon EFS 및 Amazon S3를 사용하여 온프레미스 MySQL 데이터베이스를 Aurora MySQL로 XtraBackup 마이그레이션합니다.](#)
- [Oracle 외부 테이블을 Amazon Aurora PostgreSQL 호환으로 마이그레이션](#)
- [Redis 워크로드를 AWS의 Redis Enterprise Cloud로 마이그레이션](#)
- [RHEL 소스 서버를 재부팅한 후 SELinux를 비활성화하지 않고 Replication Agent를 자동으로 다시 시작](#)
- [pg_transport를 사용하여 두 Amazon RDS DB 인스턴스 간에 PostgreSQL 데이터베이스 전송](#)

Oracle

- [Oracle Database와 Aurora PostgreSQL-Compatible 간의 링크 구성](#)
- [Oracle의 VARCHAR2\(1\) 데이터 유형을 Amazon Aurora PostgreSQL의 부울 데이터 유형으로 변환](#)
- [PostgreSQL-compatible Aurora 글로벌 데이터베이스를 사용하여 Oracle DR 에뮬레이션하기](#)
- [Oracle SQL Developer 및 AWS SCT를 사용하여 Amazon RDS for Oracle에서 Amazon RDS for PostgreSQL로 점진적으로 마이그레이션](#)
- [Aurora PostgreSQL-Compatible에서 파일 인코딩을 사용하여 BLOB 파일을 TEXT에 로드](#)
- [AWS DMS를 사용하여 SSL 모드에서 Amazon RDS for Oracle를 Amazon RDS for PostgreSQL로 마이그레이션](#)
- [AWS CLI와 AWS를 사용하여 AWS SCT 및 AWS DMS를 사용하여 오라클용 Amazon RDS를 PostgreSQL용 Amazon RDS로 마이그레이션합니다. CloudFormation](#)
- [지속적인 복제를 위해 AWS DMS를 사용하여 Amazon RDS for Oracle 데이터베이스를 다른 AWS 계정 및 AWS 리전으로 마이그레이션](#)
- [Amazon RDS for Oracle DB 인스턴스를 다른 VPC로 마이그레이션](#)
- [Oracle Data Pump를 사용하여 온프레미스 Oracle 데이터베이스를 Amazon EC2 로 마이그레이션](#)
- [Logstash를 사용하여 온프레미스 Oracle 데이터베이스를 아마존 OpenSearch 서비스로 마이그레이션합니다.](#)
- [DMS 및 SCT를 사용하여 온프레미스 Oracle 데이터베이스를 Amazon RDS for MySQL로 마이그레이션](#)
- [온프레미스 Oracle 데이터베이스를 Amazon RDS for Oracle로 마이그레이션](#)
- [데이터베이스 링크를 통한 직접 Oracle 데이터 펌프 가져오기를 사용하여 온프레미스 Oracle 데이터베이스를 Amazon RDS for Oracle로 마이그레이션](#)
- [Oracle Data Pump를 사용하여 온프레미스 Oracle 데이터베이스를 Amazon RDS for Oracle로 마이그레이션](#)
- [Oracle bystander 및 AWS DMS를 사용하여 온프레미스 Oracle 데이터베이스를 Amazon RDS for PostgreSQL로 마이그레이션](#)
- [온프레미스 Oracle 데이터베이스를 Amazon EC2의 Oracle로 마이그레이션](#)
- [AWS DMS 및 AWS SCT를 사용하여 Amazon EC2에서 Amazon RDS for MariaDB로 Oracle 데이터베이스 마이그레이션](#)
- [AWS DMS를 사용하여 Amazon EC2에서 Amazon RDS for Oracle로 Oracle 데이터베이스 마이그레이션](#)
- [AWS DMS를 사용하여 Amazon DynamoDB로 Oracle 데이터베이스 마이그레이션](#)

- [Oracle 플랫폼 파일 어댑터를 사용하여 GoldenGate Oracle 데이터베이스를 Amazon RDS for Oracle로 마이그레이션합니다.](#)
- [AWS DMS 및 AWS SCT를 사용하여 Amazon Redshift로 Oracle 데이터베이스 마이그레이션](#)
- [AWS DMS 및 AWS SCT를 사용하여 Aurora PostgreSQL로 Oracle 데이터베이스를 마이그레이션하기](#)
- [오라클 데이터 펌프와 AWS DMS를 사용하여 오라클 JD 에드워즈 EnterpriseOne 데이터베이스를 AWS로 마이그레이션하십시오.](#)
- [AWS DMS를 사용하여 Oracle 파티션형 테이블을 PostgreSQL로 마이그레이션하기](#)
- [AWS DMS를 사용하여 오라클 PeopleSoft 데이터베이스를 AWS로 마이그레이션합니다.](#)
- [Aurora PostgreSQL로 온프레미스 Oracle 데이터베이스의 데이터를 마이그레이션하기](#)
- [Amazon RDS for Oracle에서 Amazon RDS for MySQL로 마이그레이션](#)
- [구체화된 뷰와 AWS DMS를 사용하여 Oracle 8i 또는 9i에서 Amazon RDS for PostgreSQL로 마이그레이션](#)
- [AWS DMS를 사용하여 Oracle 8i 또는 9i에서 PostgreSQL용 아마존 RDS로 마이그레이션하십시오. SharePlex](#)
- [오라클을 사용하여 Oracle 데이터베이스에서 PostgreSQL용 Amazon RDS로 마이그레이션하십시오. GoldenGate](#)
- [AWS DMS 및 AWS SCT를 사용하여 Amazon EC2의 오라클에서 Amazon RDS for MySQL로 마이그레이션하십시오.](#)
- [AWS DMS를 사용하여 Oracle에서 Amazon DocumentDB로 마이그레이션](#)
- [오라클에서 Amazon WebLogic ECS의 아파치 톰캣 \(TomEE\) 으로 마이그레이션](#)
- [합수 기반 인덱스를 Oracle에서 PostgreSQL로 마이그레이션](#)
- [레거시 애플리케이션을 Oracle Pro*C에서 ECPG로 마이그레이션](#)
- [AWS에서 PostgreSQL의 개별 행으로 Oracle CLOB 값을 마이그레이션](#)
- [Oracle Database 오류 코드를 Amazon Aurora PostgreSQL Compatible 데이터베이스로 마이그레이션](#)
- [Oracle E-Business Suite를 Amazon RDS Custom으로 마이그레이션](#)
- [확장 기능을 사용하여 Oracle 네이티브 함수를 PostgreSQL로 마이그레이션](#)
- [PeopleSoft 오라클을 Amazon RDS 커스텀으로 마이그레이션](#)
- [Oracle ROWID 기능을 AWS 기반 PostgreSQL로 마이그레이션](#)
- [Oracle SERIALLY_REUSABLE 프라그마 패키지를 PostgreSQL로 마이그레이션](#)
- [가상으로 생성된 열을 오라클에서 PostgreSQL로 마이그레이션](#)

- [Aurora PostgreSQL 호환에서 Oracle UTL_FILE 기능 설정](#)
- [오라클에서 Amazon Aurora PostgreSQL로 마이그레이션한 다음 데이터베이스 객체를 검증합니다.](#)

SAP

- [온프레미스 SAP ASE 데이터베이스를 Amazon EC2로 마이그레이션](#)
- [AWS DMS를 사용하여 SAP ASE에서 Amazon RDS for SQL Server로 마이그레이션](#)
- [AWS SCT 및 AWS DMS를 사용하여 SAP ASE에 있는 Amazon EC2를 Amazon Aurora PostgreSQL-Compatible로 마이그레이션하기](#)
- [Application Migration Service를 사용하여 동종 SAP 마이그레이션 전환 시간 단축](#)

패턴 더 보기

- [CAST Highlight를 사용하여 AWS 클라우드로 마이그레이션하기 위한 애플리케이션 준비 상태 평가](#)
- [SQL Server 데이터베이스를 AWS의 MongoDB Atlas로 마이그레이션하기 위한 쿼리 성능 평가](#)
- [AWS 클라우드에서 고급 메인프레임 파일 뷰어 구축](#)
- [하이브리드 연결 모드를 사용하여 AWS의 VMware Cloud로의 데이터 센터 확장 구성](#)
- [프라이빗 네트워크를 통해 Application Migration Service 데이터 및 컨트롤 플레인에 연결](#)
- [Blu Age로 현대화된 메인프레임 워크로드 컨테이너화](#)
- [JSON Oracle 쿼리를 PostgreSQL 데이터베이스 SQL로 변환](#)
- [Teradata NORMALIZE 임시 기능을 Amazon Redshift SQL로 변환](#)
- [Teradata RESET WHEN 기능을 Amazon Redshift SQL로 변환](#)
- [AWS Backup을 사용하여 계정 전반적으로 Amazon DynamoDB 테이블의 복사](#)
- [AWS CLI를 사용하여 S3 버킷에서 다른 계정 및 리전으로 데이터 복사](#)
- [프라이빗 고정 IP를 사용하여 Amazon EC2에 Cassandra 클러스터를 배포하여 리밸런싱 방지](#)
- [AWS CDK를 사용하여 다중 스택 애플리케이션을 배포할 수 있습니다. TypeScript](#)
- [Aurora PostgreSQL의 사용자 지정 엔드포인트를 사용하여 Oracle RAC 워크로드 에뮬레이션하기](#)
- [AWR 보고서를 사용하여 오라클 데이터베이스의 Amazon RDS 엔진 크기를 추정합니다.](#)
- [Aurora PostgreSQL의 동적 SQL 명령문에서 익명 블록 처리](#)
- [Aurora PostgreSQL-Compatible에서 오버로드된 Oracle 함수 처리](#)
- [VMware vRealize 네트워크 인사이트를 VMware Cloud on AWS와 통합](#)
- [Amazon RDS for Oracle DB 인스턴스를 AMS를 사용하는 다른 계정으로 마이그레이션](#)
- [를 사용하여 온프레미스 Apache Kafka 클러스터를 Amazon MSK로 마이그레이션합니다. MirrorMaker](#)
- [AWS Glue를 사용하여 아파치 카산드라 워크로드를 Amazon Keyspaces로 마이그레이션하십시오.](#)
- [AWS DMS를 사용하여 오라클 8i 또는 9i에서 오라클용 아마존 RDS로 마이그레이션할 수 있습니다. SharePlex](#)
- [WanDisco 마이그레이션기를 사용하여 하둡 데이터를 Amazon S3로 마이그레이션합니다. LiveData](#)
- [100개 이상의 인수가 있는 Oracle 함수 및 프로시저를 PostgreSQL로 마이그레이션](#)
- [Oracle OUT 바인드 변수를 PostgreSQL 데이터베이스로 마이그레이션](#)
- [AWS MGN을 사용하여 RHEL BYOL 시스템을 AWS 라이선스가 포함된 인스턴스로 마이그레이션하기](#)

- [동일한 호스트 이름을 가진 SAP HSR을 사용하여 SAP HANA를 AWS로 마이그레이션](#)
- [분산된 가용성 그룹을 사용하여 SQL Server를 AWS로 마이그레이션](#)
- [HCX OS 지원 마이그레이션을 사용하여 VM을 AWS의 VMware Cloud로 마이그레이션](#)
- [Micro Focus Enterprise Server 및 LRS VPSX/MFI를 사용하여 AWS에서 메인프레임 온라인 인쇄 워크로드를 현대화](#)
- [OpenText 마이크로 포커스 엔터프라이즈 서버 및 LRS X를 사용하여 AWS의 메인프레임 출력 관리를 현대화합니다. PageCenter](#)
- [F5에서 AWS의 Application Load Balancer로 마이그레이션할 때 HTTP 헤더를 수정](#)
- [Microsoft SQL Server를 AWS 클라우드로 마이그레이션한 후 연결 오류 해결](#)
- [VMware Aria Operations for Logs를 사용하여 VMware Cloud on AWS에서 스포링크로 로그를 전송합니다.](#)
- [EnterpriseOne AWS Elastic 재해 복구를 사용하여 Oracle JD Edwards의 재해 복구 설정](#)
- [AWS 프라이빗 CA와 AWS RAM을 사용하여 프라이빗 인증서 관리를 간소화합니다.](#)
- [대규모 Db2 z/OS 데이터를 CSV 파일로 Amazon S3에 전송](#)

현대화

주제

- [CAST Imaging의 소프트웨어 아키텍처 분석 및 시각화](#)
- [CAST Highlight를 사용하여 AWS 클라우드로 마이그레이션하기 위한 애플리케이션 준비 상태 평가](#)
- [DynamoDB TTL을 사용하여 Amazon S3에 항목 자동으로 보관](#)
- [Amazon EC2 Auto Scaling 및 Systems Manager를 사용하여 Micro Focus Enterprise Server PAC 구축하기](#)
- [Amazon Service에서 멀티테넌트 서버리스 아키텍처 구축 OpenSearch](#)
- [AWS CDK를 사용하여 다중 스택 애플리케이션을 배포할 수 있습니다. TypeScript](#)
- [AWS SAM을 사용하여 중첩된 애플리케이션 자동 배포](#)
- [AWS Lambda 토큰 벤딩 머신을 사용하여 Amazon S3에 대한 SaaS 테넌트 격리를 구현합니다.](#)
- [AWS Step Functions을 사용하여 서버리스 사가 패턴 구현](#)
- [AWS CDK로 Amazon ECS Anywhere를 설정하여 온프레미스 컨테이너 애플리케이션을 관리](#)
- [AWS에서 ASP.NET Web Forms 애플리케이션 현대화](#)
- [AWS Fargate를 사용하여 이벤트 기반 및 예약된 워크로드를 대규모로 실행](#)
- [C# 및 AWS CDK를 사용한 사일로 모델을 위한 SaaS 아키텍처의 테넌트 온보딩](#)
- [CQRS 및 이벤트 소싱을 사용하여 모놀리식 유형을 마이크로서비스로 분해하기](#)
- [패턴 더 보기](#)

CAST Imaging의 소프트웨어 아키텍처 분석 및 시각화

제작: Arpita Sinha(Cast Software)와 James Hurrell(Cast Software)

환경: 프로덕션

기술: 현대화

워크로드: 기타 모든 워크로드

요약

이 패턴은 CAST Imaging을 사용하여 복잡한 소프트웨어 시스템을 시각적으로 탐색하고 소프트웨어 구조를 정밀하게 분석하는 방법을 보여줍니다. 이러한 방식으로 CAST Imaging을 활용하면 특히 현대화를 목적으로 할 때 애플리케이션 아키텍처에 대해 정보에 입각한 결정을 내릴 수 있습니다.

CAST Imaging에서 애플리케이션 아키텍처를 보려면 먼저 CAST Console을 통해 애플리케이션의 소스 코드를 온보딩해야 합니다. 그러면 콘솔이 애플리케이션 데이터를 CAST Imaging에 게시합니다. 여기서 애플리케이션 아키텍처를 계층별로 시각화하고 탐색할 수 있습니다.

사전 조건 및 제한 사항

사전 조건

- 활성 상태의 AWS 계정
- [CAST Imaging용 Amazon Machine Image\(AMI\)](#)
- 다음을 포함하는 Amazon Elastic Compute Cloud(Amazon EC2) 인스턴스(메모리에 최적화된 r5.xlarge Amazon EC2 인스턴스 권장):
 - 4 vCPU
 - 32GB RAM
 - 최소 500GB의 범용 솔리드 스테이트 드라이브(SSD)(gp3) 볼륨
- CAST Console 및 CAST Imaging 라이선스 키(필요한 라이선스 키를 받으려면 aws.contact-me@castsoftware.com으로 CAST에 문의)
- 압축된(.zip) 형식으로 분석하려는 애플리케이션의 전체 소스 코드
- Microsoft Edge, Mozilla Firefox, Google Chrome 중 하나

아키텍처

다음 다이어그램은 CAST Console을 통해 애플리케이션의 소스 코드를 온보딩한 다음 CAST Imaging에서 확인하는 예제 워크플로를 보여줍니다.

이 다이어그램은 다음 워크플로를 보여줍니다.

1. CAST는 프론트 엔드, 미들웨어 및 백엔드 코드를 리버스 엔지니어링하여 애플리케이션 소스 코드 메타데이터를 생성합니다.
2. CAST에서 생성된 애플리케이션 데이터는 자동으로 CAST Imaging으로 가져와서 시각화하고 분석할 수 있습니다.

다음은 이 프로세스의 작동 방식에 대한 간략한 설명입니다.

도구

- [CAST Imaging](#)은 소프트웨어 시스템을 시각적으로 보고 탐색할 수 있도록 도와주는 브라우저 기반 애플리케이션으로, 아키텍처에 대해 정보에 입각한 결정을 내릴 수 있습니다.
- [CAST Console](#)은 CAST AIP 분석을 구성, 실행 및 관리하는 데 도움이 되는 브라우저 기반 애플리케이션입니다.

참고: CAST Imaging과 CAST Console은 AMI for CAST Imaging에 포함되어 있습니다.

에픽

CAST Imaging 환경 설정

작업	설명	필요한 기술
초기 CAST Console 구성을 실행합니다.	1. 웹 브라우저를 열고 URL <code>http://localhost:8081</code> 을 입력하여 CAST Console에 연결합니다.	소프트웨어 설계자, 개발자, 기술 리더

작업	설명	필요한 기술
	<ol style="list-style-type: none"> 2. 메시지가 표시되면 CAST Console 라이선스 키를 입력합니다. 그리고 다음을 선택합니다. 3. 구성 설정입니다. 변경할 필요가 없는 경우 저장 후 종료를 선택합니다. 	
초기 CAST Imaging 구성을 실행합니다.	<ol style="list-style-type: none"> 1. 웹 브라우저를 열고 URL <code>http://localhost:8083</code>을 입력하여 CAST Imaging에 연결합니다. 2. 메시지가 표시되면 사용자 이름과 암호 모두에 <code>admin</code>을 입력하여 로그인합니다. 3. 메시지가 표시되면 CAST Imaging 라이선스 키를 입력합니다. 그런 다음 업데이트를 선택하여 키를 저장합니다. 	소프트웨어 설계자, 개발자, 기술 리더

작업	설명	필요한 기술
CAST Extend 로컬 서버 구성.	<p>(선택 사항)기본적으로 CAST Extend 로컬 서버는 오프라인 모드에서 작동하도록 구성되어 있습니다. 이 설정이 허용되는 경우 추가 구성이 필요하지 않습니다. 그러나 CAST Extend 에 직접 연결하여 온라인/프록시 모드에서 CAST Extend 로컬 서버를 구성하려면 다음 단계를 따르세요.</p> <p>참고: CAST Extend 보안 인증 정보에 대해서는 CAST Extend 등록 페이지를 참조하세요.</p> <ol style="list-style-type: none"> 1. 데스크탑의 CAST Extend 관리 센터 바로가기기를 사용하여 웹 브라우저를 로드하고 CAST Extend 로컬 서버에 연결할 수 있습니다. 2. 온라인 옵션을 선택합니다. 3. CAST Extend 보안 인증 정보(이메일 및 비밀번호)를 입력하고 저장을 선택하여 프로세스를 완료합니다. 	소프트웨어 설계자, 개발자, 기술 리더

애플리케이션을 CAST Imaging에 온보딩

작업	설명	필요한 기술
애플리케이션의 소스 코드를 준비하세요.	애플리케이션의 소스 코드를 .zip 파일 하나로 압축하여 저장합니다.	소프트웨어 설계자, 개발자, 기술 리더

작업	설명	필요한 기술
CAST Console에 애플리케이션을 추가하세요.	<ol style="list-style-type: none"> 1. 웹 브라우저를 열고 URL <code>http://localhost:8081</code>을 입력하여 CAST Console에 연결합니다. 2. 메시지가 표시되면 사용자 이름과 암호 모두에 <code>admin</code>을 입력하여 로그인합니다. 3. 애플리케이션 추가를 선택합니다. 그런 다음 애플리케이션 이름을 입력하고 추가를 선택합니다. 	소프트웨어 설계자, 개발자, 기술 리더
소스 코드 전달 마법사를 엽니다.	CAST Console에서 만든 애플리케이션을 찾으세요. 그런 다음 버전 추가를 선택합니다.	소프트웨어 설계자, 개발자, 기술 리더
애플리케이션의 소스 코드를 업로드하세요.	<p>다음 중 하나를 수행합니다.</p> <ul style="list-style-type: none"> • 애플리케이션의 소스 코드가 포함된 .zip 파일을 소스 코드 전송 마법사로 드래그 앤 드롭합니다. • 혹은 클라우드 업로드 아이콘을 선택합니다. 그런 다음 애플리케이션의 소스 코드가 들어 있는 .zip 파일을 엽니다. 	소프트웨어 설계자, 개발자, 기술 리더

작업	설명	필요한 기술
<p>분석 프로세스를 시작하세요.</p>	<ol style="list-style-type: none"> 1. 전송 마법사에서 버전 세부 정보를 제공하고 구성 옵션을 지정합니다. 자세한 내용은 CAST Imaging 설명서의 CAST Imaging용 표준 온도를 참조하세요. 2. CAST Imaging에 게시 옵션이 선택되어 있는지 확인하세요. 그런 다음 진행을 선택합니다. <p>참고: 진행을 선택하면 소스 코드에 대한 분석 프로세스가 시작됩니다. CAST Console의 진행률 창에는 분석 프로세스의 각 단계가 표시되고 분석이 완료되면 알림이 표시됩니다.</p>	<p>소프트웨어 설계자, 개발자, 기술 리더</p>

CAST Imaging에 게시된 분석 결과 및 데이터를 확인

작업	설명	필요한 기술
<p>상태 및 로그를 확인하세요.</p>	<p>모든 분석 작업이 완료되면 진행 창에 성공 메시지가 있는지 확인하세요.</p> <p>참고: 분석 작업이 완료된 후 즉시 각 분석 작업에 대한 개별 로그를 확인할 수 있습니다. 특정 작업에 대한 로그를 보려면 진행 창에서 로그 보기를 선택하세요.</p>	<p>소프트웨어 설계자, 개발자, 기술 리더</p>

작업	설명	필요한 기술
애플리케이션 세부 정보를 확인하세요.	애플리케이션 세부 정보 패널 에서 분석 결과에 대한 세부 정보를 검토하세요. 발견된 기술과 소스 코드 구성을 꼭 살펴보세요.	소프트웨어 설계자, 개발자, 기술 리더
CAST Imaging의 확인 및 액세스.	<ol style="list-style-type: none"> 1. CAST Console의 애플리케이션 관리 창에서 애플리케이션의 버전 상태가 이미징 처리됨으로 되어 있는지 확인합니다. CAST Imaging 아이콘이 나타납니다. 2. CAST Imaging 아이콘을 선택하여 CAST Imaging의 애플리케이션 데이터를 직접 탐색할 수 있습니다. <p>참고: 이미징 처리 상태는 소스 코드가 분석되어 CAST Imaging 인스턴스에 업로드되었음을 의미합니다.</p>	소프트웨어 설계자, 개발자, 기술 리더

CAST Imaging으로 애플리케이션 분석을 시작

작업	설명	필요한 기술
CAST Imaging에 로그인하세요.	CAST Imaging을 열고 기본 관리자 보안 인증 정보(admin/admin)를 입력합니다. 애플리케이션 데이터가 표시됩니다.	소프트웨어 설계자, 개발자, 기술 리더
CAST Imaging에서 애플리케이션 데이터를 탐색하세요.	CAST Imaging 기능을 사용하여 소프트웨어 아키텍처 보기를 시작하세요.	소프트웨어 설계자, 개발자, 기술 리더

작업	설명	필요한 기술
	<p>CAST Imaging 기능을 사용하는 방법에 대한 간단한 튜토리얼을 보려면 도움말 아이콘을 선택하여 CAST Imaging 도우미를 표시하세요.</p> <p>자세한 내용은 CAST Imaging 사용 설명서를 참조하세요.</p>	

관련 리소스

CAST Console 설명서

- [로그인](#)
- [CAST Console을 통한 옵션 구성](#)

CAST Imaging 설명서

- [CAST Imaging의 애플리케이션 온보딩 - 사전 조건](#)
- [CAST Imaging에 새 애플리케이션 추가](#)
- [CAST Imaging의 표준 온보딩 - 결과 확인](#)
- [로그인](#)
- [구성 옵션 - 관리 센터 GUI](#)

AWS 기반 CAST Imaging에 대한 추가 리소스

- [CAST를 통한 AWS로의 애플리케이션 현대화 가속화 — 기술](#) (AWS PartnerCast 웨비나, 무료 계정 필요)
- [CAST 및 AWS Migration Hub Refactor Spaces를 사용한 레거시 애플리케이션 현대화](#)(AWS 블로그 게시물)
- [CAST Imaging을 사용하여 애플리케이션을 AWS 아키텍처로 현대화](#)(AWS 워크숍)
- [AWS Marketplace: CAST Imaging](#)
- [AWS 기반 CAST의 모든 리소스](#)

CAST Highlight를 사용하여 AWS 클라우드로 마이그레이션하기 위한 애플리케이션 준비 상태 평가

작성자: 그렉 리베라(Cast Software)

환경: 프로덕션	소스 : 레거시 애플리케이션 소스 코드	대상: AWS에서 리팩터링된 애플리케이션 코드
R 유형: 리아키텍트	워크로드: IBM, Microsoft, 오픈 소스, Oracle	기술: 현대화, 마이그레이션, 컨테이너 및 마이크로서비스

AWS 서비스: Amazon RDS, Amazon S3

요약

CAST Highlight는 신속한 애플리케이션 포트폴리오 분석을 수행하기 위한 서비스형 소프트웨어 (SaaS) 솔루션입니다. 이 패턴은 CAST Highlight를 구성 및 사용하여 조직의 IT 포트폴리오 전반에서 사용자 지정 소프트웨어 애플리케이션의 클라우드 준비 상태를 평가하고 Amazon Web Services(AWS) 클라우드로의 현대화 또는 마이그레이션을 계획하는 방법을 설명합니다.

CAST Highlight는 애플리케이션의 클라우드 준비 상태에 대한 인사이트를 생성하고, 마이그레이션 전에 제거해야 하는 코드 차단 요인을 식별하고, 이러한 차단 요인을 제거하기 위한 작업량을 추정하고, 개별 애플리케이션이 마이그레이션 후에 사용할 수 있는 AWS 서비스를 권장합니다.

이 패턴은 CAST Highlight를 설정하고 사용하는 절차를 설명하며, 이 절차는 신규 사용자 설정, 애플리케이션 관리, 캠페인 관리, 소스 코드 분석, 결과 분석의 5단계로 구성됩니다. 성공적인 애플리케이션 스캔 및 분석을 위해서는 이 패턴의 에픽 섹션에 있는 모든 단계를 완료해야 합니다.

사전 조건 및 제한 사항

사전 조건

- Portfolio Manager 권한이 있는 활성 CAST Highlight 계정이 있어야 합니다.
- CAST Highlight Local Agent를 설치하려면 로컬 컴퓨터에 최소 300MB의 여유 디스크 공간과 4GB 메모리가 있어야 합니다.
- Microsoft Windows 8 이상의 버전이어야 합니다.

- 애플리케이션 소스 코드는 Local Agent가 설치된 시스템에서 액세스할 수 있는 텍스트 파일에 저장해야 합니다. 소스 코드는 프레미스를 떠나지 않으며 모든 코드는 로컬에서 스캔됩니다.

아키텍처

다음 다이어그램은 CAST Highlight를 사용하는 워크플로우를 보여줍니다.

이 워크플로우는 다음 단계로 구성됩니다.

1. CAST Highlight 포털에 로그인하고 Local Agent를 다운로드한 다음 로컬 컴퓨터에 설치합니다. Amazon Simple Storage Service(Amazon S3)는 Local Agent 설치 패키지를 저장합니다.
2. 소스 코드 파일을 스캔하고 결과 파일을 생성합니다.
3. 결과 파일을 CAST Highlight 포털에 업로드합니다. 중요: 결과 파일에는 소스 코드가 포함되어 있지 않습니다.
4. 스캔한 각 애플리케이션에 대한 설문조사 질문에 답변합니다.
5. CAST Highlight 포털에서 사용할 수 있는 대시보드 및 보고서를 확인합니다. Amazon Relational Database Service(Amazon RDS)는 코드 스캔, 분석 결과 및 CAST Highlight 소프트웨어 데이터를 저장합니다.

기술 스택

CAST Highlight는 애플리케이션 클라우드 준비 상태를 분석하기 위해 다음 기술을 지원합니다.

- Java
- COBOL
- C#
- C++
- Clojure
- PHP
- JavaScript
- TypeScript
- Python
- Microsoft Transact-SQL

- VB.net
- Kotlin
- Scala
- Swift

자동화 및 규모 조정

- [CLI 분석기](#)를 사용하여 CAST Highlight 분석 프로세스를 자동화할 수 있습니다.

도구

모든 사전 조건이 충족되면 이 패턴에는 도구가 필요하지 않습니다. 하지만 소스 코드 관리(SCM) 유틸리티, 코드 추출기 또는 소스 코드 파일을 관리하기 위한 기타 도구와 같은 선택적 도구 사용을 선택할 수 있습니다.

에픽

새 사용자 설정

작업	설명	필요한 기술
CAST Highlight 계정을 활성화하고 암호를 선택합니다.	CAST Highlight를 처음 사용하는 모든 사용자는 계정 활성화 이메일을 받게 됩니다. 활성화 링크를 따라 CAST Highlight 계정을 활성화하고 암호를 입력하여 활성화 프로세스를 완료합니다.	N/A
CAST Highlight 포털에 로그인합니다.	새 암호를 입력하면 CAST Highlight 홈페이지가 나타납니다. 보안 인증 정보로 CAST Highlight 포털에 로그인합니다.	N/A

애플리케이션 관리

작업	설명	필요한 기술
애플리케이션 레코드를 생성합니다.	CAST Highlight 포털에서 포트폴리오 관리 섹션의 애플리케이션 관리 탭으로 이동합니다. 화면 상단의 애플리케이션 타일에서 추가를 선택합니다.	N/A
애플리케이션 이름을 선택합니다.	애플리케이션의 이름을 입력한 후 저장을 선택합니다. 이 이름은 CAST Highlight의 애플리케이션 레코드에 사용됩니다.	N/A
모든 애플리케이션에 대해 이 단계를 반복합니다.	스캔할 각 애플리케이션에 대해 이러한 단계를 반복합니다.	N/A

캠페인 관리

작업	설명	필요한 기술
캠페인을 생성합니다.	CAST Highlight는 “캠페인”을 사용하여 특정 시점에 분석되는 일련의 애플리케이션을 설명합니다. CAST Highlight 포털에서 포트폴리오 관리 섹션의 캠페인 관리 탭으로 이동합니다. 캠페인 생성을 선택하여 캠페인 생성 화면을 실행합니다.	N/A
이름을 입력하고 캠페인 마감일을 선택합니다.	캠페인의 이름을 입력하고 그 캠페인의 마감일을 선택합니다.	N/A

작업	설명	필요한 기술
	중요: 기고자는 캠페인 마감일 이후에는 애플리케이션 분석 결과를 제출할 수 없습니다.	
소스 코드 스캔, 설문조사 답변, 도메인 및 애플리케이션 범위를 포함하기로 결정합니다.	소스 코드 분석 데이터를 질적 정보로 향상시키는 데 사용되는 표준 설문조사 중 하나 이상을 선택합니다. 설문 조사 범주는 비즈니스 영향, 소프트웨어 유지 관리 노력 CloudReady, 애플리케이션 속성 및 친환경 영향입니다. 캠페인 중에 분석되는 도메인과 애플리케이션을 선택합니다. 중요: 캠페인을 시작하기 전에 애플리케이션 관리 섹션에서 스캔하려는 애플리케이션을 모두 추가해야 합니다.	N/A
시작 메시지를 사용자 지정합니다.	캠페인의 애플리케이션과 관련된 모든 기고자에게 이메일로 전송될 시작 메시지를 사용자 지정합니다.	N/A
캠페인을 시작합니다.	완료를 선택하여 캠페인을 시작합니다.	N/A

소스 코드 분석

작업	설명	필요한 기술
CAST Highlight Local Agent를 다운로드합니다.	CAST Highlight 포털에서 애플리케이션 스캔을 선택하고	N/A

작업	설명	필요한 기술
	Local Agent를 로컬 컴퓨터에 다운로드합니다.	
Local Agent를 설치합니다.	CAST HighlightSetup .exe 설치 프로그램을 시작하고 나타나는 설치 지침을 따르십시오. Local Agent가 설치되면 애플리케이션을 분석할 준비가 된 것입니다.	N/A
Local Agent 코드 스캔 범위를 정의합니다.	<p>코드 분석은 파일 수준에서 수행되며 파일 간의 논리 링크나 종속성은 고려하지 않습니다. 모든 파일은 동일한 것으로 간주되며 애플리케이션의 일부입니다.</p> <p>정확하고 일관된 결과를 제공하려면 Local Agent에서 사용할 수 있는 파일 또는 폴더 제외 기능을 사용하여 코드 스캔 범위를 준비합니다.</p>	N/A
오픈 소스 또는 COTS 패키지를 포함합니다.	(선택 사항) 오픈 소스 또는 상업용 off-the-shelf (COTS) 패키지를 포함하려면 스캔하려는 폴더에 해당 패키지가 포함되어 있는지 확인하십시오. 일반적으로 외부 라이브러리는 “제3자” 또는 이와 유사한 하위 폴더에 그룹화되며 기본 코드는 “src/main” 파일 폴더에 있는 경우가 많습니다.	N/A

작업	설명	필요한 기술
테스트 클래스는 제외합니다.	테스트 클래스는 일반적으로 컴파일된 애플리케이션의 일부가 아니기 때문에 보통 소스 코드 분석에서 제외됩니다. 그러나 필요한 경우 스캔에 포함하도록 선택할 수 있습니다.	N/A
SCM, 빌드 및 배포 폴더는 제외합니다.	보다 일관된 결과를 얻으려면 SCM, 빌드 또는 배포 폴더(예: .git 또는 .svn 파일)를 스캔에 포함하지 않아야 합니다.	N/A
종속성 파일을 포함합니다.	물리적 파일이 스캔 중인 폴더의 일부가 아닌 프레임워크 및 종속 항목에 대한 인사이트를 얻으려면 종속성 파일(예: pom.xml, build.gradle, package.json 또는 .vcsproj 파일)을 포함해야 합니다.	N/A
Local Agent를 호출합니다.	로컬 Windows 시스템에서 Local Agent를 실행합니다.	N/A
소스 코드가 들어 있는 폴더를 선택합니다.	소스 코드가 들어 있는 폴더를 선택합니다. Local Agent가 검색할 폴더를 여러 개 추가할 수 있습니다. Local Agent는 네트워크 경로를 통한 소스 검색을 지원하지만 소스 폴더가 로컬 시스템에 있는지 확인해야 합니다. 중요: 소스 폴더에 10,000개 이상의 파일이 있는 경우 여러 번 스캔을 실행하는 것이 좋습니다.	N/A

작업	설명	필요한 기술
파일 검색을 시작합니다.	<p>Local Agent 대시보드에서 파일 검색을 선택합니다. Local Agent는 폴더와 하위 폴더에서 파일을 검색하고 해당 기술을 탐지합니다. 취소 버튼을 선택하여 언제든지 검색을 취소할 수 있습니다.</p> <p>파일 검색이 끝나면 Local Agent는 찾은 폴더와 파일을 나열합니다. 기술 열에는 관련 기술 및 파일 수가 표시됩니다. 경로 열에는 폴더 및 파일의 위치가 표시됩니다.</p>	N/A
소스 코드 스캔 구성을 수정합니다.	<p>(선택 사항) Local Agent 스캔을 세분화하려면 특정 폴더 또는 파일에 대해 하나 이상의 기술을 비활성화할 수 있습니다. 모든 기술이 비활성화된 경우 폴더 또는 파일은 스캔 범위에서 제외됩니다.</p> <p>기술을 비활성화하려면 비활성화하려는 기술의 노랑 레이블을 선택합니다. 파일이나 폴더 위에 커서를 놓을 때 필터 아이콘을 선택하여 기술을 특정 파일 또는 폴더에 연결할 수도 있습니다. 이러한 설정은 저장되며 폴더 또는 파일의 검색 프로세스를 가속합니다.</p>	N/A

작업	설명	필요한 기술
소스 코드 스캔을 시작합니다.	스캔을 구성한 후 “파일 스캔”을 선택하여 스캔 프로세스를 시작합니다.	N/A
녹색 또는 회색 레이블이 있는지 확인합니다.	<p>소스 코드 스캔이 완료되면 폴더 및 파일 수준에서 상태 레이블이 표시됩니다.</p> <p>녹색 레이블은 관련 기술로 파일이 올바르게 스캔되었음을 의미합니다.</p> <p>회색 레이블은 파일이 스캔되지 않았으며 제외되었음을 의미합니다. 각 파일의 레이블에 커서를 올리면면 제외 이유가 표시됩니다. 파일이 제외되는 이유로는 바이너리 파일, 읽을 수 없는 파일, 누락된 파일, 외부 라이브러리, 인코딩된 파일, 생성된 파일, 구문 오류, 예상 언어에 맞지 않는 콘텐츠, 충분한 분석 기준을 준수하지 않는 코드, 크기 제한(10MB)을 초과하는 파일, 시간 초과 문제 또는 분석기 사용 불가 등이 있습니다.</p>	N/A
스캔 구성을 수정하고 코드를 다시 스캔합니다.	(선택 사항) 스캔 구성 설정을 수정하고 파일 스캔을 선택하여 파일을 다시 스캔할 수 있습니다.	N/A
스캔 결과를 확인합니다.	스캔 결과가 요구 사항을 충족하는 경우 결과 확인을 선택합니다.	N/A

작업	설명	필요한 기술
<p>Local Agent에서 찾은 프레임워크와 소프트웨어 라이브러리를 확인합니다.</p>	<p>애플리케이션에서 사용하거나 참조하고 코드 스캔 중에 Local Agent가 발견한 프레임워크와 소프트웨어 라이브러리를 확인합니다. 개별 스위치 버튼을 선택하여 이러한 목록의 요소를 유지하거나 무시할 수 있습니다.</p> <p>계속하려면 종속성 확인을 선택합니다.</p> <p>중요: 프레임워크가 꺼져 있는 경우 해당 프레임워크는 CAST Highlight 포털에 나열되지 않거나 애플리케이션에 연결되지 않습니다.</p>	N/A
<p>코드 스캔 결과를 저장합니다.</p>	<p>Local Agent는 기술별로 그룹화된 코드 스캔 결과 요약 표시합니다. 저장을 선택하고 결과를 저장할 폴더를 지정합니다. Local Agent는 모든 분석 결과를 포함하는 스캔당 하나의.zip 파일을 생성합니다.</p> <p>개별 기술 및 루트 소스 폴더의 수에 따라 로컬 에이전트는.technology.date.csv 이름 지정 구조를 사용하여 하나 이상의.csv 파일을 자동으로 생성합니다.</p> <p>FolderName</p>	N/A

작업	설명	필요한 기술
코드 스캔 결과를 CAST Highlight 포털에 업로드합니다.	CAST Highlight 포털의 애플리케이션 스캔 섹션에서 분석한 애플리케이션을 선택합니다. 결과 업로드를 선택하고 .csv 파일을 선택합니다. .csv 파일을 개별적으로 업로드할 수도 있습니다. 각 파일이 업로드되면 업로드 기록이 화면에 나타납니다.	N/A
필요한 경우 분석 결과 파일을 삭제합니다.	(선택 사항) 업로드 프로세스 중에 휴지통 아이콘을 선택하여 분석 결과 파일을 언제든지 삭제할 수 있습니다. 중요: Portfolio Manager 권한이 있는 사용자 또는 결과를 업로드한 기고자만 결과를 삭제할 수 있습니다.	N/A
애플리케이션 설문조사에 응답합니다.	설문조사가 필요한 애플리케이션에는 설문조사 버튼이 표시됩니다. 설문조사를 선택하고 설문조사의 각 섹션에 대한 질문에 답하고 모두 마친 후 제출을 선택합니다. 설문조사 진행 상황은 화면 상단에 표시됩니다. 모든 필수 정보가 제출된 후에 결과를 제출할 수 있습니다. 하지만 모든 질문에 답하여 조직의 CAST Highlight 인스턴스에 있는 데이터를 보강할 수 있습니다.	N/A

작업	설명	필요한 기술
코드 스캔 결과를 제출합니다.	애플리케이션에 대한 .csv 결과 파일을 모두 업로드하고 설문 조사 질문을 완료한 후, 애플리케이션 스캔 섹션에서 제출을 선택합니다. 이 단계는 프로세스를 완료하고 CAST Highlight 포털에서 결과를 사용할 수 있도록 하는 데 필요합니다.	N/A

결과 분석

작업	설명	필요한 기술
CAST Highlight 포털 홈페이지를 확인합니다.	CAST Highlight 포털 홈페이지에는 소프트웨어 상태, 전체 포트폴리오의 오픈 소스 안전 점수 등 애플리케이션 포트폴리오에 대한 수준 높은 정보가 있는 타일이 포함되어 있습니다. CloudReady 홈페이지에는 온보딩된 애플리케이션 수도 나와 있습니다. CAST Highlight 지표 정의 및 측정 방법론에 대한 자세한 내용은 CAST Highlight — 지표 및 방법론 (Microsoft PowerPoint 프레젠테이션) 을 참조하십시오.	N/A
대시보드 보기. CloudReady	CloudReady 타일을 선택하여 CloudReady 대시보드를 엽니다. 이 대시보드는 애플리케이션의 클라우드 준비 상태를 평가하기 위한 기본 포트폴리오 수준 대시보드입니다. 이는 클	N/A

작업	설명	필요한 기술
	라우드 마이그레이션을 위한 포트폴리오 로드맵을 계획하고 개발하는 데 도움이 됩니다.	

작업	설명	필요한 기술
<p>Portfolio Advisor for Cloud 대시보드를 확인합니다.</p>	<p>Portfolio Advisor for Cloud 대시보드는 애플리케이션을 권장 마이그레이션 범주로 자동 분류합니다. 세분화는 각 애플리케이션의 기술적 특성을 기반으로 합니다. 요인에는 소스 코드 분석(클라우드 준비 상태 평가, 소프트웨어 복원력 등)과 설문조사에서 나온 비즈니스 영향이 포함됩니다. 오른쪽 상단에서 컴퓨팅을 선택하여 초기 세분화 권장 사항을 생성합니다.</p> <p>대시보드 상단의 차트에 있는 버블은 포트폴리오의 각 애플리케이션을 나타내며 권장 세분화별로 정리되어 있습니다. 또한 각 애플리케이션은 각 애플리케이션에 대한 관련 지표를 포함하여 차트 아래의 데이터 표에 나열되어 있습니다.</p> <p>권장되는 세그먼트는 다음과 같습니다.</p> <ul style="list-style-type: none"> • 리호스팅 - 서비스형 인프라 (IaaS) 솔루션을 사용하여 애플리케이션을 클라우드로 리프트 앤드 시프트하기 위해 애플리케이션의 인프라 구성을 변경하라는 권장 사항입니다. • 리팩터링 - 컨테이너형 서비스(CssS) 또는 서비스형 플 	<p>N/A</p>

작업	설명	필요한 기술
	<p>랫폼(PaaS) 솔루션을 사용하여 마이그레이션할 수 있도록 아키텍처나 기능을 변경하지 않고 애플리케이션 코드를 약간 수정하라는 권장 사항입니다.</p> <ul style="list-style-type: none"> • 리아키텍트 - 애플리케이션 코드를 대폭 수정하여 애플리케이션 상태를 개선하고 PaaS 솔루션을 사용하여 마이그레이션을 준비하거나 서비스형 기능(FaaS) 솔루션을 사용하여 서버리스 애플리케이션으로 배포하는 권장 사항입니다. • 리빌드 - 애플리케이션 코드를 삭제하고 PaaS 솔루션을 사용하여 클라우드에서 다시 개발하거나 FaaS 솔루션을 사용하여 서버리스 애플리케이션으로 다시 개발하는 권장 사항입니다. • 사용 중지 - 애플리케이션을 완전히 폐기하거나 잠재적으로 상용 서비스형 소프트웨어(SaaS) 대안으로 대체하라는 권장 사항입니다. 	

작업	설명	필요한 기술
<p>세분화 권장 사항을 수정합니다.</p>	<p>경우에 따라 CAST Highlight에서 권장하는 세그먼트를 변경할 수도 있습니다. 데이터 표에서 애플리케이션을 탐색하고 애플리케이션 이름 옆의 드롭다운 목록에서 다른 세그먼트를 선택하여 이 작업을 수행할 수 있습니다. 그런 다음 오른쪽 상단에서 저장을 선택해 변경 사항을 저장합니다.</p> <p>오른쪽 상단에서 내보내기를 선택하여 언제든지 이 데이터를 내보내기할 수도 있습니다.</p>	N/A
<p>분석할 애플리케이션을 선택합니다.</p>	<p>Portfolio Advisor for Cloud 대시보드에서 해당 애플리케이션을 분석할 애플리케이션 버블을 선택합니다. 표에서 버블 차트 다음에 있는 애플리케이션 이름을 선택하면 심층 분석을 시작할 수 있습니다.</p> <p>Code Insights(소프트웨어 상태 패턴), Trends, 및 소 Software Composition(오픈 소스 위험) 등 개별 애플리케이션을 분석할 수 있는 다양한 대시보드를 사용할 수 있습니다.</p>	N/A

작업	설명	필요한 기술
<p>개별 애플리케이션의 CloudReady 결과를 분석하십시오.</p>	<p>애플리케이션의 전체 CloudReady 점수를 보여주는 CloudReady 탭을 선택합니다. 이 점수는 CloudReady 설문조사 답변과 CloudReady 코드 스캔의 조합을 기반으로 한 가중 평균입니다. 설문조사 질문에 대한 답변은 타일 아래에 있는 표에 나와 있습니다.</p> <p>CloudReady 코드 스캔 결과를 보려면 코드 스캔을 선택합니다. 애플리케이션 코드를 스캔한 CloudReady 패턴 목록이 있습니다. 이 목록에는 다음 열이 포함됩니다.</p> <ul style="list-style-type: none"> 클라우드 요구 사항은 특정 코드 패턴입니다. 기술은 패턴의 프로그래밍 언어입니다. “영향”은 패턴이 애플리케이션에 미치는 영향입니다(C = 코드, F = 프레임워크, A = 아키텍처). 중요도는 마이그레이션 전에 이 패턴 해결에 대한 중요도 수준입니다. 기여도는 이 패턴이 전체 CloudReady 점수에 미치는 영향을 나타냅니다. 패턴이 녹색이면 부스터 패턴이 되어 CloudReady 점수가 올라갑니다. 패턴이 빨간색이면 차단제가 되어 점수를 낮춥니다. 	<p>N/A</p>

작업	설명	필요한 기술
	<p>니다. CloudReady 패턴에 색상이 없으면 감지되지 않은 차단제이므로 점수가 증가합니다. CloudReady</p> <ul style="list-style-type: none"> 로드블록은 차단 요인이 개별적으로 발생하는 횟수입니다. 패턴이 탐지된 소스 코드 파일 목록을 표시하려면 로드블록 번호를 선택합니다. Est. 작업량은 각 행의 로드블록을 해결하는 데 걸리는 예상 일수입니다. 	
<p>데이터를 Microsoft Excel로 내보냅니다.</p>	<p>(선택 사항) 추가 분석을 위해 데이터를 내보내려면 Excel로 내보내기를 선택합니다. 애플리케이션 분석 결과 데이터를 사용하여 애플리케이션의 클라우드 준비 상태를 추가로 분석하고 마이그레이션 전에 업데이트해야 할 코드를 결정할 수 있습니다.</p>	<p>N/A</p>
<p>권장 사항을 확인합니다.</p>	<p>CloudReady 코드 스캔 옆의 권장 사항을 선택하면 클라우드 서비스 권장 사항 화면을 볼 수 있습니다. 이는 애플리케이션의 특성에 따라 애플리케이션이 채택할 수 있는 AWS 서비스를 식별합니다.</p> <p>이 단계를 반복하여 분석한 모든 애플리케이션에 대한 권장 사항을 확인합니다.</p>	<p>N/A</p>

관련 리소스

캠페인 관리

- [CAST Highlight Foundation 자격증 교육 섹션 3: 포트폴리오 구성 \(동영상\)](#)

소스 코드 분석

- [CAST Highlight Foundation 자격증 교육 섹션 4: 애플리케이션 분석 \(동영상\)](#)

기타 리소스

- [AWS Marketplace의 CAST Highlight](#)
- [AWS 및 CAST: 애플리케이션 현대화 가속화](#)
- [CAST Highlight - 설명서, 제품 자습서 및 타사 도구](#)
- [CAST Highlight - 클라우드 준비 상태 평가 제품 데모 \(동영상\)](#)
- [CAST Highlight를 통한 애플리케이션 포트폴리오 현대화 \(AWS 워크숍\)](#)

DynamoDB TTL을 사용하여 Amazon S3에 항목 자동으로 보관

작성자: Tabby Ward(AWS)

코드 리포지토리: DynamoDB TTL을 사용하여 S3에 항목을 보관	환경: PoC 또는 파일럿	기술: 현대화, 데이터베이스, 서버리스, 스토리지 및 백업, 비용 관리
워크로드: 오픈 소스	AWS 서비스: Amazon S3, Amazon DynamoDB, Amazon Kinesis, AWS Lambda	

요약

이 패턴은 Amazon DynamoDB 테이블에서 오래된 데이터를 제거하고 여러 플릿의 서버를 관리할 필요 없이 Amazon Web Services(AWS)의 Amazon Simple Storage Service(S3) 버킷에 보관하는 단계를 제공합니다.

이 패턴은 Amazon DynamoDB Time to Live(TTL)를 사용하여 오래된 항목을 자동으로 삭제하고 Amazon DynamoDB Streams를 사용하여 TTL 만료 항목을 캡처합니다. 그런 다음 DynamoDB 스트림을 AWS Lambda에 연결하고, AWS Lambda는 서버를 프로비저닝하거나 관리하지 않고도 코드를 실행합니다.

새 항목이 DynamoDB 스트림에 추가되면 Lambda 함수가 시작되고 Amazon Data Firehose 전송 스트림에 데이터를 씁니다. Firehose는 Amazon S3에 데이터를 아카이브로 로드하는 간단한 완전 관리형 솔루션을 제공합니다.

DynamoDB는 센서 및 연결된 디바이스의 웹 페이지 클릭 스트림 데이터 또는 사물 인터넷(IoT) 데이터와 같은 시계열 데이터를 저장하는 데 주로 사용됩니다. 액세스 빈도가 낮은 항목을 삭제하는 대신 감사 목적으로 보관하려는 고객이 많습니다. TTL은 타임스탬프 속성에 따라 항목을 자동으로 삭제하여 보관을 간소화합니다.

TTL에 의해 삭제된 항목은 DynamoDB Streams에서 식별할 수 있으며, 여기에서는 항목 수준 수정을 캡처하고 최대 24시간 동안 로그에 저장합니다. 이 데이터는 Lambda 함수에서 사용하고 Amazon S3 버킷에 보관하여 스토리지 비용을 줄일 수 있습니다. 비용을 더욱 절감하기 위해 [Amazon S3 수명 주기 규칙](#)을 생성하여 데이터를 생성하자마자 S3 Glacier Instant Retrieval 또는 S3 Glacier Flexier

Flexival와 같이 가장 비용이 저렴한 [스토리지 클래스](#) 또는 장기 보관을 위한 Amazon S3 Glacier Deep Archive로 자동 전환할 수 있습니다.

사전 조건 및 제한 사항

사전 조건

- 활성 상태의 AWS 계정.
- [AWS Command Line Interface\(AWS CLI\) 버전 1.7 이상](#) macOS, Linux 또는 Windows에 설치 및 구성.
- [Python 3.7](#) 이상.
- [Boto3](#), 설치 및 구성됨. Boto3가 설치되지 않은 경우에는 다음 `python -m pip install boto3` 명령을 실행하여 설치합니다.

아키텍처

기술 스택

- Amazon DynamoDB
- Amazon DynamoDB Streams
- 아마존 데이터 파이어호스
- AWS Lambda
- Amazon S3

1. 항목은 TTL에 의해 삭제됩니다.
2. DynamoDB 스트림 트리거는 Lambda 스트림 프로세서 함수를 간접 호출합니다.
3. Lambda 함수는 Firehose 전송 스트림에 레코드를 배치 형식으로 저장합니다.
4. 데이터 레코드는 S3 버킷에 보관됩니다.

도구

- [AWS CLI](#) - AWS Command Line Interface(AWS CLI)는 AWS 서비스를 관리하는 통합 도구입니다.

- [Amazon DynamoDB](#) - Amazon DynamoDB는 모든 규모에서 10밀리초 미만의 성능을 제공하는 카스트 및 도큐먼트 데이터베이스입니다.
- [Amazon DynamoDB Time to Live\(TTL\)](#) - Amazon DynamoDB TTL을 사용하면 항목별 타임스탬프를 정의하여 항목이 더 이상 필요하지 않은 시점을 결정할 수 있습니다.
- [Amazon DynamoDB Streams](#) - Amazon DynamoDB Streams는 DynamoDB 테이블에서 시간 순서에 따라 항목 수준 수정을 캡처하고 이 정보를 최대 24시간 동안 로그에 저장합니다.
- [Amazon Data Firehose](#) — Amazon Data Firehose는 스트리밍 데이터를 데이터 레이크, 데이터 스토어 및 분석 서비스에 안정적으로 로드할 수 있는 가장 쉬운 방법입니다.
- [AWS Lambda](#) - AWS Lambda는 서버를 프로비저닝하거나 관리할 필요 없이 코드를 실행합니다. 사용한 컴퓨팅 시간에 대해서만 비용을 지불하면 됩니다.
- [Amazon S3](#) - Amazon Simple Storage Service(S3)는 업계 최고의 확장성, 데이터 가용성, 보안 및 성능을 제공하는 객체 스토리지 서비스입니다.

코드

이 패턴의 코드는 [DynamoDB TTL 리포지토리를 사용하여 S3에 항목 GitHub 보관에서](#) 사용할 수 있습니다.

에픽

DynamoDB 테이블, TTL 및 DynamoDB 스트림을 설정합니다.

작업	설명	필요한 기술
DynamoDB 테이블을 생성합니다.	AWS CLI를 사용하여 DynamoDB에 Reservation이라는 테이블을 생성합니다. 읽기 용량 유닛(RCU)과 쓰기 용량 유닛(WCU)을 선택하고 테이블에 두 가지 속성 ReservationID 및 ReservationDate를 지정합니다.	클라우드 아키텍트, 앱 개발자

```
aws dynamodb create-table \
```

작업	설명	필요한 기술
	<pre data-bbox="609 210 1015 976"> --table-name Reservati on \ --attribute-defi nitions Attribute Name=ReservationID ,AttributeType=S AttributeName=Rese rvationDate,Attrib uteType=N \ --key-schema Attribute Name=ReservationID ,KeyType=HASH AttributeName=Rese rvationDate,KeyTyp e=RANGE \ --provisioned-th roughput ReadCapac ityUnits=100,Write CapacityUnits=100 </pre> <p data-bbox="609 1018 1015 1144">ReservationDate 는 TTL을 켜는 데 사용되는 에포크 타임 스탬프입니다.</p>	
DynamoDB TTL을 켭니다.	<p data-bbox="609 1197 1015 1375">AWS CLI를 사용하여 ReservationDate 속성에 대해 DynamoDB TTL을 활성화합니다.</p> <pre data-bbox="609 1417 1015 1753"> aws dynamodb update-ti me-to-live \ --table-name Reservati on\ --time-to-live-spe cification Enabled=t rue,AttributeName= ReservationDate </pre>	클라우드 아키텍트, 앱 개발자

작업	설명	필요한 기술
DynamoDB 스트림을 켭니다.	<p>AWS CLI를 사용하면 <code>NEW_AND_OLD_IMAGES</code> 스트림 유형을 사용하여 <code>Reservation</code> 테이블에 대한 DynamoDB 스트림을 활성화할 수 있습니다.</p> <pre data-bbox="594 537 1029 936">aws dynamodb update-table \ --table-name Reservati on \ --stream-specifica tion StreamEna bled=true,StreamVi ewType=NEW_AND_OLD _IMAGES</pre> <p>이 스트림에는 새 항목, 업데이트된 항목, 삭제된 항목, TTL에 의해 삭제된 항목에 대한 레코드가 포함됩니다. TTL로 삭제된 항목의 레코드에는 수동으로 삭제된 항목과 구분하기 위한 추가 메타데이터 속성이 포함되어 있습니다. TTL 삭제 <code>userIdentity</code> 필드는 DynamoDB 서비스가 삭제 작업을 수행했음을 나타냅니다.</p> <p>이 패턴에서는 TTL에 의해 삭제된 항목만 보관되지만, <code>eventName</code> 이 <code>REMOVE</code>이고 <code>userIdentity</code> 가 <code>dynamodb.amazonaws.com</code> 에 상응하는 <code>principalId</code> 를 포함하</p>	클라우드 아키텍트, 앱 개발자

작업	설명	필요한 기술
	는 레코드만 보관할 수 있습니다.	

S3 버킷 생성 및 구성

작업	설명	필요한 기술
S3 버킷을 생성합니다.	<p>AWS CLI를 사용하여 AWS 리전에 대상 S3 버킷을 생성하고 해당 리전으로 us-east-1 을 대체하십시오.</p> <pre data-bbox="594 804 1027 1083">aws s3api create-bucket \ --bucket reservati onfirehosedestinat ionbucket \ --region us-east-1</pre> <p>모든 AWS 계정이 네임스페이스를 공유하므로 S3 버킷 이름은 전역 수준에서 고유해야 합니다.</p>	클라우드 아키텍트, 앱 개발자
S3 버킷에 대한 30일 수명 주기 정책을 생성합니다.	<ol style="list-style-type: none"> 1. AWS Management Console에 로그인한 후 Amazon S3 콘솔을 엽니다. 2. Firehose의 데이터가 포함된 S3 버킷을 선택합니다. 3. S3 버킷에서 관리 탭을 선택하고 수명 주기 규칙 추가를 선택합니다. 4. 수명 주기 규칙 대화 상자에 규칙 이름을 입력하고, 버킷 	클라우드 아키텍트, 앱 개발자

작업	설명	필요한 기술
	의 30일 수명 주기 규칙을 구성합니다.	

Firehose 전송 스트림 생성

작업	설명	필요한 기술
Firehose 전송 스트림을 만들고 구성합니다.	<p>GitHub 리포지토리에서 <code>CreateFireHoseToS3.py</code> 코드 예제를 다운로드하고 편집합니다.</p> <p>이 코드는 Python으로 작성되었으며 Firehose 전송 스트림과 AWS Identity 및 Access Management (IAM) 역할을 생성하는 방법을 보여줍니다. IAM 역할에는 Firehose에서 대상 S3 버킷에 쓰는 데 사용할 수 있는 정책이 있습니다.</p> <p>스크립트를 실행하려면 다음 명령과 명령줄 인수를 사용하십시오.</p> <p>인수 1=<Your_S3_bucket_ARN> , 이는 앞서 생성한 버킷의 Amazon 리소스 이름(ARN)입니다.</p> <p>인수 2= 사용자의 Firehose 이름 (이 파일럿이 <code>firehose_to_s3_stream</code> 사용하고 있습니다.)</p>	클라우드 아키텍트, 앱 개발자

작업	설명	필요한 기술
	<p>인수 3= IAM 역할 이름(이 파일럿에서 firehose_to_s3 사용)</p> <pre data-bbox="594 380 1027 619">python CreateFireHoseToS3.py <Your_S3_Bucket_ARN> firehose_to_s3_stream firehose_to_s3</pre> <p>지정된 IAM 역할이 없는 경우 스크립트는 신뢰할 수 있는 관계 정책과 충분한 Amazon S3 권한을 부여하는 정책을 사용하여 수입 역할을 생성합니다. 이러한 정책의 예는 추가 정보 섹션을 참조하십시오.</p>	
<p>Firehose 전송 스트림을 확인합니다.</p>	<p>전송 스트림이 성공적으로 생성되었는지 확인하기 위해 AWS CLI를 사용하여 Firehose 전송 스트림을 설명하십시오.</p> <pre data-bbox="594 1241 1027 1480">aws firehose describe-delivery-stream --delivery-stream-name firehose_to_s3_stream</pre>	<p>클라우드 아키텍트, 앱 개발자</p>

Firehose 전송 스트림을 처리하기 위한 Lambda 함수를 생성합니다.

작업	설명	필요한 기술
<p>Lambda 함수에 대해 신뢰 정책을 생성합니다.</p>	<p>다음 정보로 신뢰 정책 파일을 생성합니다.</p>	<p>클라우드 아키텍트, 앱 개발자</p>

작업	설명	필요한 기술
	<pre data-bbox="597 212 1024 961"> { "Version": "2012-10-17", "Statement": [{ "Effect": "Allow", "Principal": { "Service" : "lambda.amazonaws. com" }, "Action": "sts:AssumeRole" }] } </pre> <p data-bbox="597 1003 1024 1136">이렇게 하면 함수에 AWS 리소스에 액세스할 수 있는 권한이 부여됩니다.</p>	
<p data-bbox="110 1178 513 1262">Lambda 함수용 실행 역할을 생성합니다.</p>	<p data-bbox="597 1178 1024 1262">실행 역할을 생성하려면 다음 코드를 실행합니다.</p> <pre data-bbox="597 1297 1024 1535"> aws iam create-role --role-name lambda- ex --assume-role-poli cy-document file://Tr ustPolicy.json </pre>	<p data-bbox="1073 1178 1487 1209">클라우드 아키텍트, 앱 개발자</p>

작업	설명	필요한 기술
<p>역할에 권한을 추가하십시오.</p>	<p>역할에 권한을 추가하려면 <code>attach-policy-to-role</code> 명령을 사용합니다.</p> <pre data-bbox="594 394 1026 1423"> aws iam attach-role-policy --role-name lambda-ex --policy-arn arn:aws:iam::aws:policy/service-role/AWSLambdaBasicExecutionRole aws iam attach-role-policy --role-name lambda-ex --policy-arn arn:aws:iam::aws:policy/service-role/AWSLambdaDynamoDBExecutionRole aws iam attach-role-policy --role-name lambda-ex --policy-arn arn:aws:iam::aws:policy/AmazonKinesisFirehoseFullAccess aws iam attach-role-policy --role-name lambda-ex --policy-arn arn:aws:iam::aws:policy/IAMFullAccess </pre>	<p>클라우드 아키텍트, 앱 개발자</p>

작업	설명	필요한 기술
<p>Lambda 함수를 생성합니다.</p>	<p>다음 명령어를 실행하여 코드 리포지토리의 <code>LambdaStreamProcessor.py</code> 파일을 압축합니다.</p> <pre data-bbox="602 443 1027 604">zip function.zip LambdaStreamProcessor.py</pre> <p>Lambda 함수를 생성할 때, Lambda 실행 역할 ARN이 필요합니다. ARN을 가져오려면 다음 코드를 실행합니다.</p> <pre data-bbox="602 856 1027 974">aws iam get-role \ --role-name lambda-ex</pre> <p>Lambda 함수를 생성하려면 다음 코드를 실행합니다.</p> <pre data-bbox="602 1136 1027 1854">aws lambda create-function --function-name LambdaStreamProcessor \ --zip-file fileb://function.zip --handler LambdaStreamProcessor.handler --runtime python3.8 \ --role {Your Lambda Execution Role ARN} \ --environment Variables="{firehose_name=firehose_t o_s3_stream,bucket_arn = arn:aws:s 3::reservationfir ehosedestinationbu</pre>	<p>클라우드 아키텍트, 앱 개발자</p>

작업	설명	필요한 기술
<p>Lambda 함수 트리거를 구성합니다.</p>	<pre data-bbox="597 205 1023 346">cket, iam_role_name = firehose_to_s3, batch_size=400}"</pre> <p data-bbox="597 388 1023 661">AWS CLI를 사용하여 Lambda 함수를 간접 호출하는 트리거 (DynamoDB 스트림)를 구성합니다. 배치 크기 400은 Lambda 동시성 문제가 발생하지 않도록 하기 위한 것입니다.</p> <pre data-bbox="597 693 1023 1123">aws lambda create-event-source-mapping --function-name LambdaStreamProcessor \ --batch-size 400 --starting-position LATEST \ --event-source-arn <Your Latest Stream ARN From DynamoDB Console></pre>	<p>클라우드 아키텍트, 앱 개발자</p>

기능 테스트

작업	설명	필요한 기술
<p>타임스탬프가 만료된 항목을 예약 테이블에 추가합니다.</p>	<p data-bbox="597 1417 1023 1690">기능을 테스트하려면 에포크 타임스탬프가 만료된 항목을 Reservation 테이블에 추가하십시오. TTL은 타임스탬프를 기반으로 항목을 자동으로 삭제합니다.</p> <p data-bbox="597 1732 1023 1858">Lambda 함수는 DynamoDB 스트림 활동 시 시작되며 이벤트를 필터링하여 REMOVE 활동 또</p>	<p>클라우드 아키텍트</p>

작업	설명	필요한 기술
	<p>는 삭제된 항목을 식별합니다. 그런 다음 Firehose 전송 스트림에 레코드를 배치 형식으로 저장합니다.</p> <p>Firehose 전송 스트림은 접두사가 지정된 대상 S3 버킷으로 항목을 전송합니다. <code>firehose-to-s3-example/year=current year/month=current month/day=current day/hour=current hour/</code>.</p> <p>중요: 데이터 검색을 최적화하려면 추가 정보 섹션에 자세히 설명된 Prefix 및 <code>ErrorOutputPrefix</code> 를 사용하여 Amazon S3를 구성하십시오.</p>	

리소스 정리

작업	설명	필요한 기술
모든 리소스를 삭제합니다.	모든 리소스를 삭제하여 사용하지 않는 서비스에 대해 요금이 청구되지 않도록 합니다.	클라우드 아키텍트, 앱 개발자

관련 리소스

- [스토리지 수명 주기 관리](#)
- [Amazon S3 스토리지 클래스](#)
- [AWS SDK for Python\(Boto3\) 설명서](#)

추가 정보

Firehose 전송 스트림 생성 및 구성 — 정책 예제

Firehose 신뢰할 수 있는 관계 정책 예제 문서

```
firehose_assume_role = {
  'Version': '2012-10-17',
  'Statement': [
    {
      'Sid': '',
      'Effect': 'Allow',
      'Principal': {
        'Service': 'firehose.amazonaws.com'
      },
      'Action': 'sts:AssumeRole'
    }
  ]
}
```

S3 권한 정책 예

```
s3_access = {
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "",
      "Effect": "Allow",
      "Action": [
        "s3:AbortMultipartUpload",
        "s3:GetBucketLocation",
        "s3:GetObject",
        "s3:ListBucket",
        "s3:ListBucketMultipartUploads",
        "s3:PutObject"
      ],
      "Resource": [
        "{your s3_bucket ARN}/*",
        "{Your s3 bucket ARN}"
      ]
    }
  ]
}
```

```
}
```

기능 테스트 - Amazon S3 구성

다음과 Prefix 및 ErrorOutputPrefix의 Amazon S3 구성이 데이터 검색을 최적화하도록 선택되었습니다.

접두사

```
firehose-3example/year=! {timestamp: yyyy}/month=! {timestamp:MM}/day=!
{timestamp:dd}/hour=!{timestamp:HH}/
```

Firehose는 먼저 S3 버킷 firehose-3example 바로 아래에 호출되는 기본 폴더를 만듭니다. 그런 다음 Java [DateTimeFormatter](#) 형식을 사용하여 표현식 `!{timestamp:yyyy} !{timestamp:MM} !{timestamp:dd}`, 및 `!{timestamp:HH}` 연도, 월, 일, 시간을 평가합니다.

예를 들어, Unix 에포크 타임의 대략적인 도착 타임스탬프 1604683577은 `year=2020`, `month=11`, `day=06` 및 `hour=05`로 평가됩니다. 따라서 데이터 레코드가 전송되는 Amazon S3의 위치는 `firehose-3example/year=2020/month=11/day=06/hour=05/`로 평가됩니다.

ErrorOutputPrefix

```
firehose-3erroroutputbase/!{firehose:random-string}/!{firehose:error-output-type}/!
{timestamp:yyyy/MM/dd}/
```

ErrorOutputPrefix는 S3 버킷 바로 아래에 firehose-3erroroutputbase가 직접 호출되는 기본 폴더를 만듭니다. 표현식 `!{firehose:random-string}`은 `ztWxkdg3Thg`와 같은 11자 임의 문자열로 평가됩니다. 실패한 레코드가 전송된 Amazon S3 객체의 위치를 `firehose-3erroroutputbase/ztWxkdg3Thg/processing-failed/2020/11/06/`로 평가할 수 있습니다.

Amazon EC2 Auto Scaling 및 Systems Manager를 사용하여 Micro Focus Enterprise Server PAC 구축하기

작성자: Kevin Yung (AWS), Peter Woods (Micro Focus), Abraham Rondon (Micro Focus), Krithika Palani Selvam (AWS)

환경: 프로덕션

기술: 현대화, 클라우드 네이티브, 인프라 DevOps

요약

이 패턴은 [스케일 아웃 성능 및 가용성 클러스터\(PAC\)의 Micro Focus Enterprise Server](#)와 Amazon Web Services(AWS)의 Amazon Elastic Compute Cloud(Amazon EC2) Auto Scaling 그룹을 사용하는 메인프레임 애플리케이션을 위한 확장 가능한 아키텍처를 도입합니다. 이 솔루션은 AWS Systems Manager 및 Amazon EC2 Auto Scaling 수명 주기 후크를 사용하여 완전히 자동화됩니다. 이 패턴을 사용하면, 용량 수요에 따라 자동으로 규모를 확장 및 축소하여 높은 복원력을 달성하도록 메인프레임 온라인 및 배치 애플리케이션을 설정할 수 있습니다.

사전 조건 및 제한 사항

사전 조건

- 활성 상태의 AWS 계정.
- Micro Focus Enterprise Server 소프트웨어 및 라이선스. 자세한 내용은 [Micro Focus 영업팀](#)에 문의하십시오.
- Micro Focus Enterprise Server에서 실행할 메인프레임 애플리케이션을 재구축하고 제공하는 개념에 대한 이해입니다. 높은 수준의 개요는 [Micro Focus Enterprise Server 데이터시트](#)를 참고하십시오.
- Micro Focus Enterprise Server 스케일 아웃 성능 및 가용성 클러스터의 개념에 대한 이해. 자세한 내용은 [Micro Focus Enterprise Server 설명서](#)를 참조하세요.
- 지속적 통합 (CI)을 DevOps 통한 메인프레임 애플리케이션의 전체 개념에 대한 이해. AWS와 Micro Focus에서 개발한 AWS Preciptive Guidance 패턴에 대해서는 [메인프레임 현대화: DevOps Micro Focus를 사용한 AWS 기반 메인프레임 현대화를](#) 참조하십시오.

제한 사항

- Micro Focus Enterprise Server 데이터시트에서 지원하는 플랫폼 목록은 [Micro Focus Enterprise Server 데이터 시트](#)를 참고하십시오.
- 이 패턴에 사용된 스크립트와 테스트는 Amazon EC2 Windows Server 2019를 기반으로 합니다. 다른 Windows Server 버전 및 운영 체제에서는 이 패턴에 대해 테스트되지 않았습니다.
- 이 패턴은 Windows용 Micro Focus Enterprise Server 6.0을 기반으로 합니다. 이전 또는 이후 릴리스는 이 패턴 개발 과정에서 테스트되지 않았습니다.

제품 버전

- Micro Focus Enterprise Server 6.0
- Windows Server 2019

아키텍처

기존 메인프레임 환경에서는 애플리케이션과 기업 데이터를 호스팅할 하드웨어를 프로비저닝해야 합니다. 계절별, 월별, 분기별 또는 예상치 못하거나 전혀 없는 수요 급증에 대응하기 위해, 메인프레임 사용자는 스토리지와 컴퓨팅 파워를 추가로 구매하여 규모를 확장해야 합니다. 스토리지 및 컴퓨팅 용량 리소스의 수를 늘리면 전반적인 성능이 향상되지만 이러한 확장이 선형적인 것은 아닙니다.

Amazon EC2 Auto Scaling 및 Micro Focus Enterprise 서버를 사용하여 AWS에서 온디맨드 소비 모델을 채택하기 시작하는 경우에는 그렇지 않습니다. 다음 섹션에서는 Amazon EC2 Auto Scaling 그룹과 함께 Micro Focus Enterprise Server 스케일 아웃 성능 및 가용성 클러스터(PAC)를 사용하여 완전히 자동화되고 확장 가능한 메인프레임 애플리케이션 아키텍처를 구축하는 방법을 자세히 설명합니다.

Micro Focus Enterprise Server 자동 확장 아키텍처

먼저 Micro Focus Enterprise Server의 기본 개념을 이해하는 것이 중요합니다. 이 환경은 기존에 IBM 메인프레임에서 실행되던 애플리케이션을 위한 메인프레임과 호환되는 x86 배포 환경을 제공합니다. 온라인 및 배치 실행과 다음을 지원하는 트랜잭션 환경을 모두 제공합니다.

- IBM COBOL
- IBM PL/I
- IBM JCL 배치 작업
- IBM CICS 및 IMS TM 트랜잭션
- 웹 서비스
- SORT를 포함한 일반 배치 유틸리티

Micro Focus Enterprise Server를 사용하면 메인프레임 애플리케이션을 최소한의 변경으로 실행할 수 있습니다. 기존 메인프레임 워크로드를 x86 플랫폼으로 이동하고 현대화하여 AWS 클라우드 네이티브 확장을 활용함으로써 새로운 시장이나 지역으로 빠르게 확장할 수 있습니다.

AWS Preciptive Guidance 패턴 [메인프레임 현대화: DevOps AWS 기반 Micro Focus](#)에서는 [Micro Focus 엔터프라이즈 개발자 및 AWS와](#) AWS의 엔터프라이즈 테스트 서버를 사용하여 AWS에서 메인프레임 애플리케이션의 개발 및 테스트를 가속화하는 아키텍처를 도입했습니다. CodePipeline CodeBuild 이 패턴은 고가용성과 복원력을 달성하기 위해 메인프레임 애플리케이션을 AWS 프로덕션 환경에 배포하는 데 중점을 둡니다.

메인프레임 프로덕션 환경에서는 고성능 및 고가용성을 달성하기 위해 메인프레임에 IBM Parallel Sysplex를 설치했을 수 있습니다. Sysplex와 유사한 스케일 아웃 아키텍처를 생성하기 위해 Micro Focus는 엔터프라이즈 서버에 성능 및 가용성 클러스터(PAC)를 도입했습니다. PAC는 단일 이미지로 관리되고 Amazon EC2 인스턴스에서 확장되는 여러 엔터프라이즈 서버 지역에 메인프레임 애플리케이션을 배포할 수 있도록 지원합니다. 또한 PAC는 예측 가능한 애플리케이션 성능과 온디맨드 시스템 처리량을 지원합니다.

PAC에서는 여러 엔터프라이즈 서버 인스턴스가 하나의 논리적 엔터티로 함께 작동합니다. 따라서 용량이 다른 지역과 공유되고 Amazon EC2 Auto Scaling 그룹과 같은 업계 표준 기능을 사용하여 새 인스턴스가 자동으로 시작되므로 어떤 Enterprise Server 인스턴스에 장애가 발생해도 비즈니스 연속성이 중단되지 않습니다. 이렇게 하면 단일 장애 지점이 제거되어 하드웨어, 네트워크 및 애플리케이션 문제에 대한 복원력이 향상됩니다. ESCWA(엔터프라이즈 서버 공용 웹 관리) API를 사용하여 확장된 엔터프라이즈 서버 인스턴스를 운영 및 관리할 수 있으므로 엔터프라이즈 서버의 운영 유지 관리 및 서비스 가능성을 간소화할 수 있습니다.

참고: Micro Focus는 엔터프라이즈 서버 지역에 장애가 발생하거나 유지 관리가 필요한 경우 가용성이 저하되지 않도록 [성능 및 가용성 클러스터\(PAC\)](#)를 3개 이상의 엔터프라이즈 서버 지역으로 구성할 것을 권장합니다.

PAC 구성에는 지역 데이터베이스, 지역 간 데이터베이스 및 선택적 데이터 저장소 데이터베이스를 관리하기 위해 지원되는 관계형 데이터베이스 관리 서비스(RDBMS)가 필요합니다. 가용성과 확장성을 높이려면 Micro Focus Database File Handler 지원을 사용하여 VSAM(가상 저장소 액세스 방법) 파일을 관리하는 데 데이터 저장소 데이터베이스를 사용해야 합니다. 지원되는 RDBMS에는 다음이 포함됩니다.

- Microsoft SQL Server 2009 R2 이상
- PostgreSQL 10.x (Amazon Aurora PostgreSQL-Compatible Edition 포함)
- DB2 10.4 이상

지원되는 RDBMS 및 PAC 요구 사항에 대한 자세한 내용은 [Micro Focus Enterprise Server - 사전 조건](#) 및 [Micro Focus Enterprise Server - PAC 권장 구성](#)을 참고하십시오.

다음 다이어그램은 Micro Focus PAC의 일반적인 AWS 아키텍처 설정을 보여줍니다.

	구성 요소	설명
1	Enterprise Server 인스턴스 오토 스케일링 그룹	PAC에서 Enterprise Server 인스턴스와 함께 배포된 오토 스케일링 그룹을 설정합니다. Amazon CloudWatch 경보를 통해 지표를 사용하여 인스턴스 수를 확장하거나 시작할 수 있습니다. CloudWatch
2	Enterprise Server 인스턴스 오토 스케일링 그룹	Enterprise Server Common Web Administration(ESCWA)와 함께 배포된 오토 스케일링 그룹을 설정합니다. ESCWA는 클러스터 관리 API를 제공합니다. ESCWA 서버는 Enterprise Server 인스턴스 오토 스케일링 이벤트 중에 Enterprise Server를 추가 또는 제거하고 PAC에서 Enterprise Servers 지역을 시작 또는 중지하는 컨트롤 플레인 역할을 합니다. ESCWA 인스턴스는 PAC 관리에만 사용되기 때문에 트래픽 패턴을 예측할 수 있으며 원하는 용량 오토 스케일링 요건을 1로 설정할 수 있습니다.
3	다중 AZ 설정의 Amazon Aurora 인스턴스	Enterprise Server 인스턴스에서 공유할 사용자 및 시스템 데이터 파일을 모두 호스팅하도

		록 관계형 데이터베이스 관리 시스템(RDBMS)을 설정합니다.
4	Amazon ElastiCache Redis용 인스턴스 및 복제본	ElastiCache Redis 기본 인스턴스와 하나 이상의 복제본을 설정하여 사용자 데이터를 호스팅하고 엔터프라이즈 서버 인스턴스의 스케일 아웃 리포지토리 (SOR) 역할을 하십시오. 특정 유형의 사용자 데이터를 저장하도록 하나 이상의 스케일 아웃 리포지토리 를 구성할 수 있습니다. Enterprise Server는 Redis NoSQL 데이터베이스를 SOR로 사용하며, 이는 PAC 무결성을 유지하기 위한 요구 사항 입니다.
5	Network Load Balancer	애플리케이션이 Enterprise Server 인스턴스에서 제공하는 서비스에 연결할 수 있도록 호스트 이름을 제공하여 로드 밸런서를 설정합니다(예: 3270 에 물레이터를 통해 애플리케이션에 액세스).

이러한 구성 요소는 Micro Focus Enterprise Server PAC 클러스터의 최소 요구 사항을 구성합니다. 다음 섹션에서는 클러스터 관리 자동화에 대해 다룹니다.

AWS Systems Manager Automation을 사용하여 확장하기

PAC 클러스터를 AWS에 배포한 후, PAC는 Enterprise Server Common Web Administration(ESCWA) API를 통해 관리됩니다.

자동 조정 이벤트 중에 클러스터 관리 작업을 자동화하려면 Systems Manager Automation 런북 및 Amazon EC2 Auto Scaling을 사용할 수 있습니다. EventBridge 이러한 자동화의 아키텍처는 다음 다이어그램에 나와 있습니다.

	구성 요소	설명
1	오토 스케일링 수명 주기 후크	자동 조정 수명 주기 후크를 설정하고 자동 조정 그룹에서 새 인스턴스가 시작되고 기존 인스턴스가 EventBridge 종료되면 Amazon에 알림을 보냅니다.
2	아마존 EventBridge	자동 조정 이벤트를 Systems Manager Automation 런북 대상으로 라우팅하도록 Amazon EventBridge 규칙을 설정합니다.
3	Automation 런북	Systems Manager 자동화 런북을 설정하여 Windows PowerShell 스크립트를 실행하고 ESCWA API를 호출하여 PAC를 관리합니다. 예제를 보려면 추가 정보 섹션을 참조하세요.
4	자동 조정 그룹의 Enterprise Server ESCWA 인스턴스	자동 조정 그룹의 Enterprise Server ESCWA 인스턴스를 설정합니다. ESCWA 인스턴스는 PAC를 관리하기 위한 API를 제공합니다.

도구

- [Micro Focus Enterprise Server](#) – Micro Focus Enterprise Server는 Enterprise Developer의 통합 개발 환경(IDE)의 모든 변형으로 만든 애플리케이션을 위한 실행 환경을 제공합니다.

- [Amazon EC2 Auto Scaling](#) – Amazon EC2 Auto Scaling을 사용하면 애플리케이션의 로드를 처리할 수 있는 정확한 수의 Amazon EC2 인스턴스를 유지할 수 있습니다. Auto Scaling 그룹이라는 EC2 인스턴스 모음을 생성하고 최소 및 최대 인스턴스 수를 지정합니다.
- [Amazon ElastiCache for ElastiCache Redis—Amazon](#)은 클라우드에서 분산된 인메모리 데이터 스토어 또는 캐시 환경을 설정, 관리 및 확장하기 위한 웹 서비스입니다. 확장 가능하고 비용 효율적인 고성능 캐싱 솔루션을 제공합니다.
- [Amazon RDS](#) - Amazon Relational Database Service(RDS)는 AWS 클라우드의 관계형 데이터베이스를 더 쉽게 설치, 운영 및 확장할 수 있게 하는 웹 서비스입니다. 이 서비스는 관계형 데이터베이스를 위한 경제적이고 크기 조절이 가능한 용량을 제공하고 공통 데이터베이스 관리 작업을 관리합니다.
- [AWS Systems Manager](#) – AWS Systems Manager는 AWS에서 인프라를 보고 제어하기 위해 사용할 수 있는 AWS 서비스입니다. Systems Manager 콘솔을 사용하여 여러 AWS 서비스의 운영 데이터를 보고 AWS 리소스에서 운영 태스크를 자동화할 수 있습니다. Systems Manager는 관리형 인스턴스를 검사하고 탐지된 정책 위반을 보고하거나 시정 조치를 취해서 보안 및 규정 준수를 유지하는데 도움이 됩니다.

에픽

Amazon Aurora 인스턴스 생성

작업	설명	필요한 기술
Amazon Aurora 인스턴스용 AWS CloudFormation 템플릿을 생성합니다.	AWS 예제 코드 스니펫 을 사용하여 Amazon Aurora CloudFormation PostgreSQL 호환 에디션 인스턴스를 생성하는 템플릿을 만드십시오.	클라우드 아키텍트
CloudFormation 스택을 배포하여 Amazon Aurora 인스턴스를 생성합니다.	CloudFormation 템플릿을 사용하여 프로덕션 워크로드에 다중 AZ 복제가 활성화된 Aurora PostgreSQL 호환 인스턴스를 만들 수 있습니다.	클라우드 아키텍트
Enterprise Server의 데이터베이스 연결 설정을 구성합니다.	Micro Focus 설명서 의 지침에 따라 Micro Focus Enterprise	데이터 엔지니어, 엔지니어 DevOps

작업	설명	필요한 기술
	Server의 연결 문자열 및 데이터베이스 구성을 준비하십시오.	

Redis 인스턴스용 Amazon ElastiCache 클러스터를 생성합니다.

작업	설명	필요한 기술
Redis 인스턴스용 Amazon ElastiCache 클러스터용 CloudFormation 템플릿을 생성합니다.	AWS 예제 코드 스니펫 을 사용하여 Redis 인스턴스용 Amazon ElastiCache 클러스터를 생성하는 CloudFormation 템플릿을 만드십시오.	클라우드 아키텍트
CloudFormation 스택을 배포하여 Redis 인스턴스용 Amazon ElastiCache 클러스터를 생성합니다.	프로덕션 워크로드에 다중 AZ 복제가 활성화된 Redis 인스턴스용 Amazon ElastiCache 클러스터를 생성합니다.	클라우드 아키텍트
Enterprise Server PSOR 연결 설정을 구성합니다.	Micro Focus 설명서 의 지침에 따라 Micro Focus Enterprise Server PAC의 PAC 스케일 아웃 리포지토리(PSOR) 연결 구성을 준비합니다.	DevOps 엔지니어

Micro Focus Enterprise Server ESCWA 오토 스케일링 그룹 생성

작업	설명	필요한 기술
Micro Focus Enterprise Server AMI를 생성합니다.	Amazon EC2 Windows Server 인스턴스를 생성하고 EC2 인스턴스에 Micro Focus Enterprise Server 바이너리를 설치합니다. EC2 인스턴스의	클라우드 아키텍트

작업	설명	필요한 기술
	<p>Amazon Machine Image(AMI)를 생성합니다. 자세한 내용은 Enterprise Server 설치 설명서를 참조하세요.</p>	
<p>엔터프라이즈 서버 ESCWA용 CloudFormation 템플릿을 생성합니다.</p>	<p>AWS 예제 코드 스니펫을 사용하여 오토 스케일링 그룹에서 Enterprise Server ESCWA의 사용자 지정 스택을 생성하기 위한 템플릿을 만듭니다.</p>	<p>클라우드 아키텍트</p>
<p>CloudFormation 스택을 배포하여 엔터프라이즈 서버 ESCWA용 Amazon EC2 스케일링 그룹을 생성합니다.</p>	<p>CloudFormation 템플릿을 사용하여 이전 스토리에서 만든 Micro Focus 엔터프라이즈 서버 ESCWA AMI를 사용하여 자동 스케일링 그룹을 배포하십시오.</p>	<p>클라우드 아키텍트</p>

AWS Systems Manager Automation 런북 생성

작업	설명	필요한 기술
<p>Systems Manager 자동화 런북용 CloudFormation 템플릿을 생성합니다.</p>	<p>추가 정보 섹션의 예제 코드 스니펫을 사용하여 PAC 생성, 엔터프라이즈 서버 스케일 인 및 엔터프라이즈 서버 스케일 아웃을 자동화하는 Systems Manager Automation 런북을 생성하는 CloudFormation 템플릿을 만드십시오.</p>	<p>클라우드 아키텍트</p>
<p>Systems Manager 자동화 런북이 포함된 CloudFormation 스택을 배포합니다.</p>	<p>CloudFormation 템플릿을 사용하여 PAC 생성, 엔터프라이즈 서버 스케일 인 및 엔터프라이즈 서버 스케일아웃을 위한 자</p>	<p>클라우드 아키텍트</p>

작업	설명	필요한 기술
	동화 런북이 포함된 스택을 배포하십시오.	

Micro Focus Enterprise Server용 자동 조정 그룹 생성

작업	설명	필요한 기술
Micro Focus 엔터프라이즈 서버의 자동 스케일링 그룹을 설정하기 위한 CloudFormation 템플릿을 생성합니다.	<p>AWS 예제 코드 스니펫을 사용하여 자동 조정 그룹을 생성할 CloudFormation 템플릿을 만드십시오. 이 템플릿은 Micro Focus Enterprise Server ESCWA 인스턴스용으로 생성된 것과 동일한 AMI를 재사용합니다.</p> <p>그런 다음 AWS 예제 코드 스니펫을 사용하여 자동 조정 수명 주기 이벤트를 생성하고 동일한 템플릿에서 확장 및 축소 이벤트를 EventBridge 필터링 하도록 Amazon을 설정합니다. CloudFormation</p>	클라우드 아키텍트
Micro Focus 엔터프라이즈 서버의 자동 크기 조정 그룹용 CloudFormation 스택을 배포하십시오.	Micro Focus 엔터프라이즈 서버용 자동 스케일링 그룹이 포함된 CloudFormation 스택을 배포하십시오.	클라우드 아키텍트

관련 리소스

- [Micro Focus Enterprise Server 성능 및 가용성 클러스터\(PAC\)](#)
- [Amazon EC2 Auto Scaling 수명 주기 후크](#)
- [를 사용하여 트리거를 사용하여 자동화를 실행합니다. EventBridge](#)

추가 정보

PAC 클러스터를 스케일 인하거나 스케일 아웃하려면 다음 시나리오를 자동화해야 합니다.

PAC 시작 또는 재생성을 위한 자동화

PAC 클러스터를 시작할 때 Enterprise Server는 ESCWA가 API를 호출하여 PAC 구성을 생성해야 합니다. 그러면 Enterprise Server 지역이 시작되고 PAC에 Enterprise Server 영역이 추가됩니다. PAC를 생성 또는 재생성하려면 다음 단계를 따릅니다.

1. 지정된 이름을 사용하여 ESCWA에서 [PAC 스케일 아웃 리포지토리 PSOR](#)를 구성합니다.

```
POST /server/v1/config/groups/sors
```

2. 지정된 이름으로 PAC를 생성하고 여기에 PSOR을 연결합니다.

```
POST /server/v1/config/groups/pacs
```

3. PAC를 처음 설정하는 경우 지역 데이터베이스와 지역 간 데이터베이스를 구성합니다.

참고: 이 단계에서는 SQL 쿼리와 Micro Focus Enterprise Suite 명령줄 dbfhadmin 도구를 사용하여 데이터베이스를 만들고 초기 데이터를 가져옵니다.

4. PAC 정의를 Enterprise Server 지역에 설치합니다.

```
POST /server/v1/config/mfds
POST /native/v1/config/groups/pacs/${pac_uid}/install
```

5. PAC에서 Enterprise Server 지역을 시작합니다.

```
POST /native/v1/regions/${host_ip}/${port}/${region_name}/start
```

Windows PowerShell 스크립트를 사용하여 이전 단계를 구현할 수 있습니다.

다음 단계에서는 Windows PowerShell 스크립트를 재사용하여 PAC 생성을 자동화하는 방법을 설명합니다.

1. 부트스트랩 프로세스의 일부로 Windows PowerShell 스크립트를 다운로드하거나 생성하는 Amazon EC2 시작 템플릿을 생성합니다. 예를 들면, EC2 사용자 데이터를 사용하여 Amazon Simple Storage Service(Amazon S3) 버킷에서 스크립트를 다운로드할 수 있습니다.

2. AWS Systems Manager Automation 런북을 생성하여 Windows PowerShell 스크립트를 호출하십시오.
3. 인스턴스 태그를 사용하여 런북을 ESCWA 인스턴스에 연결합니다.
4. 시작 템플릿을 사용하여 ESCWA 자동 조정 그룹을 생성합니다.

다음 예제 AWS CloudFormation 스니펫을 사용하여 자동화 런북을 생성할 수 있습니다.

PAC 생성에 사용되는 Systems Manager 자동화 런북의 예제 CloudFormation 스니펫

```
PACInitDocument:
  Type: AWS::SSM::Document
  Properties:
    DocumentType: Command
    Content:
      schemaVersion: '2.2'
      description: Operation Runbook to create Enterprise Server PAC
      mainSteps:
        - action: aws:runPowerShellScript
          name: CreatePAC
          inputs:
            onFailure: Abort
            timeoutSeconds: "1200"
            runCommand:
              - |
                C:\Scripts\PAC-Init.ps1
PacInitAutomation:
  Type: AWS::SSM::Document
  Properties:
    DocumentType: Automation
    Content:
      description: Prepare Micro Focus PAC Cluster via ESCWA Server
      schemaVersion: '0.3'
      assumeRole: !GetAtt SsmAssumeRole.Arn
      mainSteps:
        - name: RunPACInitDocument
          action: aws:runCommand
          timeoutSeconds: 300
          onFailure: Abort
          inputs:
            DocumentName: !Ref PACInitDocument
          Targets:
            - Key: tag:Enterprise Server - ESCWA
```

```

      Values:
        - "true"
PacInitDocumentAssociation:
  Type: AWS::SSM::Association
  Properties:
    DocumentVersion: "$LATEST"
    Name: !Ref PACInitDocument
    Targets:
      - Key: tag:Enterprise Server - ESCWA
        Values:
          - "true"

```

자세한 내용은 [Micro Focus Enterprise Server - PAC 구성](#)을 참고하십시오.

새 Enterprise Server 인스턴스로 스케일 아웃하기 위한 자동화

Enterprise Server 인스턴스를 스케일 아웃할 때는 해당 Enterprise Server 지역을 PAC에 추가해야 합니다. 다음 단계는 ESCWA API를 호출하고 Enterprise Server 지역을 PAC에 추가하는 방법을 설명합니다.

1. PAC 정의를 Enterprise Server 지역에 설치합니다.

```

POST '/server/v1/config/mfds'
POST /native/v1/config/groups/pacs/${pac_uid}/install

```

2. PAC에서 해당 지역을 웹 스타트합니다.

```

POST /native/v1/regions/${host_ip}/${port}/${region_name}/start

```

3. 자동 스케일링 그룹을 로드 밸런서에 연결하여 Enterprise Server 인스턴스를 로드 밸런서에 추가합니다.

Windows 스크립트를 사용하여 이전 단계를 구현할 수 있습니다. PowerShell 자세한 내용은 [Micro Focus Enterprise Server - PAC 구성](#)을 참고하십시오.

다음 단계는 Windows PowerShell 스크립트를 재사용하여 새로 시작된 Enterprise Server 인스턴스를 PAC에 추가하는 이벤트 기반 자동화를 구축하는 데 사용할 수 있습니다.

1. 부트스트랩 중에 Enterprise Server Region을 프로비저닝하는 Enterprise Server 인스턴스용 Amazon EC2 시작 템플릿을 생성합니다. 예를 들어 Micro Focus Enterprise Server 명령 mfd

를 사용하여 지역 구성을 가져올 수 있습니다. 이 명령에 사용할 수 있는 자세한 내용 및 옵션은 [Enterprise Server Reference](#)를 참고하십시오.

2. 이전 단계에서 생성된 시작 템플릿을 사용하는 Enterprise Server 오토 스케일링 그룹을 생성합니다.
3. Systems Manager 자동화 런북을 생성하여 Windows PowerShell 스크립트를 호출합니다.
4. 인스턴스 태그를 사용하여 런북을 ESCWA 인스턴스에 연결합니다.
5. 엔터프라이즈 서버 자동 조정 그룹의 EC2 인스턴스 시작 성공 이벤트를 필터링하는 Amazon EventBridge 규칙을 생성하고 자동화 런북을 사용할 대상을 생성합니다.

다음 예제 CloudFormation 스니펫을 사용하여 자동화 런북과 규칙을 생성할 수 있습니다. EventBridge 엔터프라이즈 서버 인스턴스를 확장하는 데 사용되는 Systems Manager의 예제 CloudFormation 스니펫

```
ScaleOutDocument:
  Type: AWS::SSM::Document
  Properties:
    DocumentType: Command
    Content:
      schemaVersion: '2.2'
      description: Operation Runbook to Adding MFDS Server into an existing PAC
      parameters:
        MfdsPort:
          type: String
        InstanceIpAddress:
          type: String
          default: "Not-Available"
        InstanceId:
          type: String
          default: "Not-Available"
      mainSteps:
        - action: aws:runPowerShellScript
          name: Add_MFDS
          inputs:
            onFailure: Abort
            timeoutSeconds: "300"
            runCommand:
              - |
                $ip = "{{InstanceIpAddress}}"
                if ( ${ip} -eq "Not-Available" ) {
```

```

    $ip = aws ec2 describe-instances --instance-id {{InstanceId}} --output
text --query "Reservations[0].Instances[0].PrivateIpAddress"
  }
  C:\Scripts\Scale-Out.ps1 -host_ip ${ip} -port {{MfdsPort}}

```

PacScaleOutAutomation:

Type: AWS::SSM::Document

Properties:

DocumentType: Automation

Content:

parameters:

MfdsPort:

type: String

InstanceIpAddress:

type: String

default: "Not-Available"

InstanceId:

type: String

default: "Not-Available"

description: Scale Out 1 New Server in Micro Focus PAC Cluster via ESCWA

Server

schemaVersion: '0.3'

assumeRole: !GetAtt SsmAssumeRole.Arn

mainSteps:

- name: RunScaleOutCommand

action: aws:runCommand

timeoutSeconds: 300

onFailure: Abort

inputs:

DocumentName: !Ref ScaleOutDocument

Parameters:

InstanceIpAddress: "{{InstanceIpAddress}}"

InstanceId: "{{InstanceId}}"

MfdsPort: "{{MfdsPort}}"

Targets:

- Key: tag:Enterprise Server - ESCWA

Values:

- "true"

Enterprise Server 인스턴스의 스케일 인을 위한 자동화

스케일 아웃과 마찬가지로, Enterprise Server 인스턴스가 스케일 인되면, EC2 인스턴스 종료 라이프 사이클 작업 이벤트가 시작되고 PAC에서 Micro Focus Enterprise Server 인스턴스를 제거하려면 다음 프로세스와 API 호출이 필요합니다.

1. 종료되는 Enterprise Server 인스턴스에서 해당 지역을 중지하십시오.

```
POST "/native/v1/regions/${host_ip}/${port}/${region_name}/stop"
```

2. PAC에서 Enterprise Server 인스턴스를 제거합니다.

```
DELETE "/server/v1/config/mfds/${uid}"
```

3. 신호를 보내 Enterprise Server 인스턴스를 계속 종료합니다.

이전 단계는 Windows PowerShell 스크립트에서 구현할 수 있습니다. 이 프로세스에 대한 자세한 내용은 [Micro Focus Enterprise Server 문서 - PAC 관리](#)를 참고하십시오.

다음 단계는 Windows 스크립트를 재사용하여 PAC에서 Enterprise Server 인스턴스를 종료하는 이벤트 기반 자동화를 구축하는 방법을 설명합니다. PowerShell

1. Systems Manager 자동화 런북을 생성하여 Windows PowerShell 스크립트를 호출합니다.
2. 인스턴스 태그를 사용하여 런북을 ESCWA 인스턴스에 연결합니다.
3. EC2 인스턴스 종료를 위한 오토 스케일링 그룹 라이프사이클 후크를 생성합니다.
4. Enterprise Server 자동 스케일링 그룹에 대한 EC2 인스턴스 종료 라이프사이클 작업 이벤트를 필터링하는 Amazon EventBridge 규칙을 생성하고 자동화 런북을 사용할 대상을 생성합니다.

다음 예제 CloudFormation 템플릿을 사용하여 Systems Manager Automation 런북, 라이프사이클 후크 및 EventBridge 규칙을 생성할 수 있습니다.

엔터프라이즈 서버 인스턴스의 확장에 사용되는 Systems Manager 자동화 런북의 예제 CloudFormation 스택

```
ScaleInDocument:
  Type: AWS::SSM::Document
  Properties:
    DocumentType: Command
    Content:
      schemaVersion: '2.2'
      description: Operation Runbook to Remove MFDS Server from PAC
```

```

parameters:
  MfdsPort:
    type: String
  InstanceIpAddress:
    type: String
    default: "Not-Available"
  InstanceId:
    type: String
    default: "Not-Available"
mainSteps:
- action: aws:runPowerShellScript
  name: Remove_MFDS
  inputs:
    onFailure: Abort
    runCommand:
      - |
        $ip = "{{InstanceIpAddress}}"
        if ( ${ip} -eq "Not-Available" ) {
          $ip = aws ec2 describe-instances --instance-id {{InstanceId}} --output
text --query "Reservations[0].Instances[0].PrivateIpAddress"
        }
        C:\Scripts\Scale-In.ps1 -host_ip ${ip} -port {{MfdsPort}}

PacScaleInAutomation:
  Type: AWS::SSM::Document
  Properties:
    DocumentType: Automation
    Content:
      parameters:
        MfdsPort:
          type: String
        InstanceIpAddress:
          type: String
          default: "Not-Available"
        InstanceId:
          type: String
          default: "Not-Available"
      description: Scale In 1 New Server in Micro Focus PAC Cluster via ESCWA Server
      schemaVersion: '0.3'
      assumeRole: !GetAtt SsmAssumeRole.Arn
      mainSteps:
        - name: RunScaleInCommand
          action: aws:runCommand
          timeoutSeconds: "600"

```

```

onFailure: Abort
inputs:
  DocumentName: !Ref ScaleInDocument
  Parameters:
    InstanceIpAddress: "{{InstanceIpAddress}}"
    MfdsPort: "{{MfdsPort}}"
    InstanceId: "{{InstanceId}}"
  Targets:
    - Key: tag:Enterprise Server - ESCWA
      Values:
        - "true"
- name: TerminateTheInstance
  action: aws:executeAwsApi
  inputs:
    Service: autoscaling
    Api: CompleteLifecycleAction
    AutoScalingGroupName: !Ref AutoScalingGroup
    InstanceId: "{{ InstanceId }}"
    LifecycleActionResult: CONTINUE
    LifecycleHookName: !Ref ScaleInLifeCycleHook

```

Amazon EC2 오토 스케일링 트리거 자동화

Enterprise Server 인스턴스에 대한 스케일링 정책을 설정하는 프로세스에는 애플리케이션 동작에 대한 이해가 필요합니다. 대부분의 경우, 대상 추적 조정 정책을 설정할 수 있습니다. 예를 들어 평균 CPU 사용률을 Amazon CloudWatch 지표로 사용하여 자동 조정 정책을 설정할 수 있습니다. 자세한 설명은 [Amazon EC2 Auto Scaling에 대한 대상 추적 조정 정책](#)을 참조하세요. 트래픽 패턴이 규칙적인 애플리케이션의 경우 예측 스케일링 정책을 사용하는 것을 고려해 보십시오. 자세한 정보는 [Amazon EC2 Auto Scaling의 예측 조정](#)을 참조하세요.

Amazon Service에서 멀티테넌트 서버리스 아키텍처 구축 OpenSearch

태비 워드(AWS)와 니샤 감비르(AWS)가 제작했습니다.

환경: PoC 또는 파일럿

기술: 현대화, SaaS, 서버리스

워크로드: 오픈 소스

AWS 서비스: 아마존
OpenSearch 서비스, AWS 람
다, 아마존 S3, 아마존 API 게
이트웨이

요약

Amazon OpenSearch Service는 널리 사용되는 오픈 소스 검색 및 분석 엔진인 Elasticsearch를 쉽게 배포, 운영 및 확장할 수 있는 관리형 서비스입니다. Amazon OpenSearch Service는 로그 및 지표와 같은 스트리밍 데이터에 대한 거의 실시간 수집 및 대시보딩뿐 아니라 자유 텍스트 검색도 제공합니다.

SaaS (Software as a OpenSearch Service) 제공업체는 Amazon Service를 자주 사용하여 확장 가능하고 안전한 방식으로 고객 통찰력을 확보하는 동시에 복잡성과 다운타임을 줄이는 등 광범위한 사용 사례를 해결합니다.

멀티 테넌트 환경에서 Amazon OpenSearch Service를 사용하면 SaaS 솔루션의 파티셔닝, 격리, 배포 및 관리에 영향을 미치는 일련의 고려 사항이 적용됩니다. SaaS 공급자는 지속적으로 변화하는 워크로드에 맞춰 Elasticsearch 클러스터를 효과적으로 확장하는 방법을 고려해야 합니다. 또한 계층화 및 소음이 많은 이웃 조건이 파티셔닝 모델에 어떤 영향을 미칠 수 있는지도 고려해야 합니다.

이 패턴은 Elasticsearch 구조로 테넌트 데이터를 표현하고 분리하는 데 사용되는 모델을 검토합니다. 또한 이 패턴은 간단한 서버리스 참조 아키텍처를 예로 들어 멀티 테넌트 환경에서 Amazon OpenSearch Service를 사용한 인덱싱 및 검색을 보여줍니다. 테넌트의 데이터 격리를 유지하면서 모든 테넌트 간에 동일한 인덱스를 공유하는 풀 데이터 파티셔닝 모델을 구현합니다. 이 패턴은 아마존 API 게이트웨이, AWS Lambda, 아마존 심플 스토리지 서비스 (Amazon S3), 아마존 서비스와 같은 아마존 웹 서비스 (AWS) 서비스를 사용합니다. OpenSearch

풀 모델 및 기타 데이터 파티셔닝 모델에 대한 자세한 내용은 [추가 정보](#) 섹션을 참조하십시오.

사전 조건 및 제한 사항

사전 조건

- 활성 상태의 AWS 계정.
- [AWS 명령줄 인터페이스 \(AWS CLI\) 버전 2.x](#), macOS, 리눅스 또는 윈도우에 설치 및 구성
- [Python 버전 3.7](#)
- [pip3](#) — Python 소스 코드는 Lambda 함수에 배포할 수 있는.zip 파일로 제공됩니다. 코드를 로컬에서 사용하거나 사용자 지정하려면 다음 단계에 따라 소스 코드를 개발하고 다시 컴파일하십시오.
 1. Python 스크립트와 동일한 디렉터리에서 `pip3 freeze > requirements.txt` 명령을 실행하여 `requirements.txt` 파일을 생성합니다.
 2. 종속성을 설치합니다: `pip3 install -r requirements.txt`

제한 사항

- 이 코드는 Python에서 실행되며 현재 다른 프로그래밍 언어를 지원하지 않습니다.
- 샘플 애플리케이션에는 AWS 크로스 리전 또는 재해 복구 (DR) 지원이 포함되어 있지 않습니다.
- 이 패턴은 데모용으로만 제공됩니다. 프로덕션 환경에서 사용하기 위한 것이 아닙니다.

아키텍처

다음 다이어그램은 이 패턴의 높은 수준의 아키텍처를 보여줍니다. 이 아키텍처에는 다음 이벤트가 포함됩니다.

- 콘텐츠를 인덱싱하고 쿼리하는 AWS Lambda
- 검색을 수행하는 Amazon OpenSearch 서비스
- 사용자와의 API 상호 작용을 제공하는 Amazon API Gateway
- 원시 (인덱싱되지 않은) 데이터를 저장하는 Amazon S3
- Amazon은 CloudWatch 로그를 모니터링합니다
- 테넌트 역할 및 정책을 생성하는 AWS Identity and Access Management (IAM)

자동화 및 규모 조정

단순화를 위해 패턴은 AWS CLI를 사용하여 인프라를 프로비저닝하고 샘플 코드를 배포합니다. AWS CloudFormation 템플릿 또는 AWS Cloud Development Kit (AWS CDK) 스크립트를 생성하여 패턴을 자동화할 수 있습니다.

도구

서비스

- [AWS CLI](#) — AWS Command Line Interface(AWS CLI)는 명령줄 셸에서 명령을 사용하여 AWS 서비스와 리소스를 관리하는 통합 도구입니다.
- [AWS Lambda](#) - AWS Lambda는 서버를 프로비저닝하거나 관리하지 않고도 코드를 실행할 수 있게 해주는 컴퓨팅 서비스입니다. Lambda는 필요 시에만 코드를 실행하며, 일일 몇 개의 요청에서 초당 수천 개의 요청까지 자동으로 규모를 조정합니다.
- [Amazon API Gateway](#) — Amazon API Gateway는 모든 규모에서 REST, HTTP 및 WebSocket API를 생성, 게시, 유지 관리, 모니터링 및 보호하는 데 사용되는 AWS 서비스입니다.
- [Amazon S3](#) - Amazon Simple Storage Service(S3)는 웹에서 언제 어디서나 원하는 양의 정보를 저장하고 검색할 수 있는 개체 스토리지 서비스입니다.
- [Amazon OpenSearch Service](#) — Amazon OpenSearch Service는 Elasticsearch를 대규모로 비용 효율적으로 쉽게 배포, 보호 및 실행할 수 있게 해주는 완전 관리형 서비스입니다.

코드

첨부 파일은 이 패턴에 대한 샘플 파일을 제공합니다. 다음이 포함됩니다.

- `index_lambda_package.zip`— 풀 모델을 사용하여 OpenSearch Amazon 서비스의 데이터를 인덱싱하는 Lambda 함수.
- `search_lambda_package.zip`— Amazon 서비스에서 데이터를 검색하기 위한 Lambda 함수. OpenSearch
- `Tenant-1-data`— 테넌트-1의 원시 (인덱싱되지 않은) 데이터 샘플.
- `Tenant-2-data`— 테넌트-2의 원시 (인덱싱되지 않은) 데이터 샘플.

중요: 이 패턴의 스토리에는 Unix, Linux 및 macOS용으로 포맷된 CLI 명령 예제가 포함됩니다. Windows의 경우 각 줄의 끝에 있는 백슬래시(\) Unix 연속 문자를 캐럿(^)으로 바꿉니다.

에픽

S3 버킷 생성 및 구성

작업	설명	필요한 기술
S3 버킷을 생성합니다.	<p>AWS 리전에 S3 버킷을 생성합니다. 이 버킷에는 샘플 애플리케이션의 인덱싱되지 않은 테넌트 데이터가 보관됩니다. 모든 AWS 계정이 네임스페이스를 공유하므로 S3 버킷 이름은 전역 수준에서 고유해야 합니다.</p> <p>S3 버킷을 생성하려면 다음과 같이 AWS CLI create-bucket 명령을 사용할 수 있습니다.</p> <pre>aws s3api create-bucket \ --bucket tenantraw data \ --region <your-AWS-Region></pre> <p>tenantrawdata 은 S3 버킷 이름입니다. (버킷 이름 지정 지침을 따르는 모든 고유한 이름을 사용할 수 있습니다.)</p>	클라우드 아키텍트, 클라우드 관리자

Elasticsearch 클러스터 생성 및 구성

작업	설명	필요한 기술
Amazon OpenSearch 서비스 도메인을 생성합니다.	AWS CLI create-elasticsearch-domain 명령을 실행하여 아마	클라우드 아키텍트, 클라우드 관리자

작업	설명	필요한 기술
	<p>존 OpenSearch 서비스 도메인을 생성합니다.</p> <pre>aws es create-elasticsearch-domain \ --domain-name vpc-cli-example \ --elasticsearch-version 7.10 \ --elasticsearch-cluster-config InstanceType=t3.medium.elasticsearch,InstanceCount=1 \ --ebs-options EBSEnabled=true,VolumeType=gp2,VolumeSize=10 \ --domain-endpoint-options "{\"EnforceHTTPS\": true}" \ --encryption-at-rest-options "{\"Enabled\": true}" \ --node-to-node-encryption-options "{\"Enabled\": true}" \ --advanced-security-options "{\"Enabled\": true, \"InternalUserDatabaseEnabled\": true, \"MasterUserOptions\": {\"MasterUserName\": \"KibanaUser\", \"MasterUserPassword\": \"NewKibanaPassword@123\"}}" \ --vpc-options "{\"SubnetIds\":</pre>	

작업	설명	필요한 기술
	<pre data-bbox="609 210 1015 934"> [\ "<subnet-id>\"], \"SecurityGroupIds\": [\ "<sg-id>\"]} \" \ --access-policies {\"Version\": \"2012-10-17\", \"Statement\": [{ \"Effect\": \"Allow\", \ \"Principal\": {\"AWS\": \"*\"}, \"Action\": \"es:*\", \ \"Resource\": \"arn:aws:es:regio n:account-id:domain /vpc-cli-example/* \" }] }\" </pre> <p data-bbox="592 976 1031 1354"> 도메인은 테스트용이므로 인스턴스 수는 1로 설정되어 있습니다. 도메인을 생성한 후에는 세부 정보를 변경할 수 없으므로 advanced-security-options 파라미터를 사용하여 세밀한 액세스 제어를 활성화해야 합니다. </p> <p data-bbox="592 1396 1031 1575"> 이 명령은 Kibana 콘솔에 로그인하는 데 사용할 수 있는 마스터 사용자 이름(KibanaUser)과 암호를 생성합니다. </p> <p data-bbox="592 1617 1031 1837"> 도메인은 가상 사설 클라우드 (VPC) 의 일부이므로 사용할 액세스 정책을 지정하여 Elasticsearch 인스턴스에 연결할 수 있는지 확인해야 합니다. </p>	

작업	설명	필요한 기술
	자세한 내용은 AWS 설명서의 VPC를 사용하여 Amazon OpenSearch 서비스 도메인 시작 을 참조하십시오.	

작업	설명	필요한 기술
Bastion Host 설정합니다.	<p>Amazon Elastic Compute Cloud (Amazon EC2) Windows 인스턴스를 Kibana 콘솔에 액세스할 수 있는 접속 bastion host 로 설정합니다. Elasticsearch 보안 그룹은 Amazon EC2 보안 그룹으로부터의 트래픽을 허용해야 합니다. 지침은 Bastion Server를 사용하여 EC2 인스턴스에 대한 네트워크 액세스 제어 블로그 게시물을 참조하십시오.</p> <p>배스천 호스트가 설정되고 인스턴스와 연결된 보안 그룹을 사용할 수 있게 되면, AWS authorize-security-group-ingress CLI 명령을 사용하여 Elasticsearch 보안 그룹에 권한을 추가하여 Amazon EC2 (배스천 호스트) 보안 그룹에서 포트 443을 허용할 수 있습니다.</p> <pre data-bbox="597 1333 1026 1808">aws ec2 authorize-security-group-ingress \ --group-id <SecurityGroupIdElasticSearch> \ --protocol tcp \ --port 443 \ --source-group <SecurityGroupIdBastionHostEC2></pre>	클라우드 아키텍트, 클라우드 관리자

Lambda 인덱스 함수 생성 및 구성하기

작업	설명	필요한 기술
<p>Lambda 실행 역할을 만듭니다.</p>	<p>AWS CLI create-role 명령을 실행하여 Lambda 인덱스 함수에 AWS 서비스 및 리소스에 대한 액세스 권한을 부여합니다.</p> <pre data-bbox="594 533 1029 814">aws iam create-role \ --role-name index-lambda-role \ --assume-role-policy-document file://lambda_assume_role.json</pre> <p>lambda_assume_role.json 은(는) 다음과 같이 Lambda 함수에 AssumeRole 권한을 부여하는 현재 폴더의 JSON 문서입니다.</p> <pre data-bbox="594 1115 1029 1864">{ "Version": "2012-10-17", "Statement": [{ "Effect": "Allow", "Principal": { "Service": "lambda.amazonaws.com" }, "Action": "sts:AssumeRole" }] }</pre>	<p>클라우드 아키텍트, 클라우드 관리자</p>

작업	설명	필요한 기술
<p>Lambda 역할에 관리되는 정책을 첨부합니다.</p>	<p>AWS CLI attach-role-policy 명령을 실행하여 이전 단계에서 생성한 역할에 관리형 정책을 연결합니다. 이 두 정책은 역할에 Elastic Network 인터페이스를 생성하고 로그에 로그를 쓸 수 있는 권한을 부여합니다.</p> <p>CloudWatch</p> <pre>aws iam attach-role-policy \ --role-name index-lambda-role \ --policy-arn arn:aws:iam::aws:policy/service-role/AWSLambdaBasicExecutionRole aws iam attach-role-policy \ --role-name index-lambda-role \ --policy-arn arn:aws:iam::aws:policy/service-role/AWSLambdaVPCLessExecutionRole</pre>	<p>클라우드 아키텍트, 클라우드 관리자</p>

작업	설명	필요한 기술
<p>Lambda 인덱스 함수에 S3 객체를 읽을 수 있는 권한을 부여하는 정책을 생성합니다.</p>	<p>AWS CLI create-policy 명령을 실행하여 Lambda 인덱스 함수에 S3 버킷의 객체를 읽을 수 있는 <code>s3:GetObject</code> 권한을 부여합니다.</p> <pre data-bbox="592 489 1027 730">aws iam create-policy \ --policy-name s3- permission-policy \ --policy-document file://s3-policy.json</pre> <p>파일 <code>s3-policy.json</code> 은 (는) S3 객체에 대한 읽기 액세스를 허용하는 <code>s3:GetObject</code> 권한을 부여하는 현재 폴더의 JSON 문서입니다. S3 버킷을 생성할 때 다른 이름을 사용한 경우 다음 <code>Resource</code> 섹션에 올바른 버킷 이름을 제공하십시오.</p> <pre data-bbox="592 1220 1027 1856">{ "Version": "2012-10-17", "Statement": [{ "Effect": "Allow", "Action": "s3:GetObject", "Resource": "arn:aws:s3:::tena ntrawdata/*" }] }</pre>	<p>클라우드 아키텍트, 클라우드 관리자</p>

작업	설명	필요한 기술
<p>Amazon S3 권한 정책을 Lambda 실행 역할에 연결합니다.</p>	<p>AWS CLI attach-role-policy 명령을 실행하여 이전 단계에서 생성한 Amazon S3 권한 정책을 Lambda 실행 역할에 연결합니다.</p> <pre data-bbox="597 491 1024 768">aws iam attach-role-policy \ --role-name index-lambda-role \ --policy-arn <PolicyARN></pre> <p>PolicyARN 은(는) Amazon S3 권한 정책의 Amazon 리소스 이름(ARN)입니다. 이 값은 이전 명령의 출력에서 가져올 수 있습니다.</p>	<p>클라우드 아키텍트, 클라우드 관리자</p>

작업	설명	필요한 기술
<p>Lambda 인덱스 함수를 생성합니다.</p>	<p>AWS CLI create-function 명령을 실행하여 아마존 서비스에 액세스할 Lambda 인덱스 함수를 생성합니다. OpenSearch</p> <pre data-bbox="597 443 1026 1314"> aws lambda create-function \ --function-name \ index-lambda-function \ --zip-file fileb:// \ index_lambda_package.zip \ --handler lambda_index.lambda_handler \ --runtime python3.7 \ --role "arn:aws:iam::account-id:role/index-lambda-role" \ --timeout 30 \ --vpc-config \ "{\"SubnetIds\": \ [\"<subnet-id1>\", \ \"<subnet-id2>\"], \ \"SecurityGroupIds \ \": [\"<sg-1>\"]}" </pre>	<p>클라우드 아키텍트, 클라우드 관리자</p>

작업	설명	필요한 기술
<p>Amazon S3가 Lambda 인덱스 함수를 호출하도록 허용합니다.</p>	<p>AWS CLI 추가 권한 명령을 실행하여 Amazon S3에 Lambda 인덱스 함수를 호출할 권한을 부여합니다.</p> <pre data-bbox="594 443 1027 1115"> aws lambda add-permission \ --function-name \ index-lambda-function \ --statement-id s3-permissions \ --action lambda:InvokeFunction \ --principal s3.amazonaws.com \ --source-arn \ "arn:aws:s3:::tenantrawdata" \ --source-account \ "<account-id>" </pre>	<p>클라우드 아키텍트, 클라우드 관리자</p>

작업	설명	필요한 기술
Amazon S3 이벤트에 Lambda 트리거를 추가합니다.	<p>Amazon S3 이벤트가 감지되면 AWS CLI put-bucket-notification-configuration 명령을 실행하여 Lambda 인덱스 함수에 알림을 보냅니다. ObjectCreated 인덱스 함수는 객체가 S3 버킷에 업로드될 때마다 실행됩니다.</p> <pre>aws s3api put-bucket-notification-configuration \ --bucket tenantraw-data \ --notification-configuration file://s3-trigger.json</pre> <p>파일 s3-trigger.json 은 (는) Amazon S3 ObjectCreated 이벤트가 발생할 때 Lambda 함수에 리소스 정책을 추가하는 현재 폴더의 JSON 문서입니다.</p>	클라우드 아키텍트, 클라우드 관리자

Lambda 검색 기능 생성 및 구성하기

작업	설명	필요한 기술
Lambda 실행 역할을 만듭니다.	<p>AWS CLI create-role 명령을 실행하여 Lambda 검색 함수에 AWS 서비스 및 리소스에 대한 액세스 권한을 부여합니다.</p> <pre>aws iam create-role \</pre>	클라우드 아키텍트, 클라우드 관리자

작업	설명	필요한 기술
	<pre data-bbox="609 212 1015 426"> --role-name search-lambda-role \ --assume-role-policy-document file://lambda_assume_role.json </pre> <p data-bbox="592 464 1019 688"> lambda_assume_role.json 은(는) 다음과 같이 Lambda 함수에 AssumeRole 권한을 부여하는 현재 폴더의 JSON 문서입니다. </p> <pre data-bbox="609 730 1015 1486"> { "Version": "2012-10-17", "Statement": [{ "Effect": "Allow", "Principal": { "Service": "lambda.amazonaws.com" }, "Action": "sts:AssumeRole" }] } </pre>	

작업	설명	필요한 기술
<p>Lambda 역할에 관리되는 정책을 첨부합니다.</p>	<p>AWS CLI attach-role-policy 명령을 실행하여 이전 단계에서 생성한 역할에 관리형 정책을 연결합니다. 이 두 정책은 역할에 Elastic Network 인터페이스를 생성하고 로그에 로그를 쓸 수 있는 권한을 부여합니다.</p> <p>CloudWatch</p> <pre>aws iam attach-role-policy \ --role-name search-lambda-role \ --policy-arn arn:aws:iam::aws:policy/service-role/AWSLambdaBasicExecutionRole aws iam attach-role-policy \ --role-name search-lambda-role \ --policy-arn arn:aws:iam::aws:policy/service-role/AWSLambdaVPCLessExecutionRole</pre>	<p>클라우드 아키텍트, 클라우드 관리자</p>

작업	설명	필요한 기술
<p>Lambda 함수를 생성합니다.</p>	<p>AWS CLI create-function 명령을 실행하여 아마존 서비스에 액세스하는 Lambda 검색 함수를 생성합니다. OpenSearch</p> <pre data-bbox="592 441 1031 1312">aws lambda create-function \ --function-name search-lambda-function \ --zip-file fileb://search_lambda_package.zip \ --handler lambda_search.lambda_handler \ --runtime python3.7 \ --role "arn:aws:iam::account-id:role/search-lambda-role" \ --timeout 30 \ --vpc-config '{"SubnetIds":["<subnet-id1>","<subnet-id2>"],"SecurityGroupIds":["<sg-1>"]}'</pre>	<p>클라우드 아키텍트, 클라우드 관리자</p>

테넌트 역할 생성 및 구성

작업	설명	필요한 기술
<p>테넌트 IAM 역할 생성</p>	<p>AWS CLI create-role 명령을 실행하여 검색 기능을 테스트하는 데 사용할 두 개의 테넌트 역할을 생성합니다.</p> <pre data-bbox="592 1816 1031 1869">aws iam create-role \</pre>	<p>클라우드 아키텍트, 클라우드 관리자</p>

작업	설명	필요한 기술
	<pre data-bbox="609 212 1015 415"> --role-name Tenant-1- role \ --assume-role-poli cy-document file://as sume-role-policy.json aws iam create-role \ --role-name Tenant-2- role \ --assume-role-poli cy-document file://as sume-role-policy.json </pre> <p data-bbox="592 772 1015 997">파일 <code>assume-role-policy.json</code> 은(는) Lambda 실행 역할에 <code>AssumeRole</code> 권한을 부여하는 현재 폴더의 JSON 문서입니다.</p> <pre data-bbox="609 1056 1015 1837"> { "Version": "2012-10-17", "Statement": [{ "Effect": "Allow", "Principa 1": { "AWS": "<Lambda execution role for index function>", "AWS": "<Lambda execution role for search function>" }, "Action": "sts:AssumeRole" }] } </pre>	

작업	설명	필요한 기술
	}	

작업	설명	필요한 기술
<p>테넌트 IAM 정책을 생성합니다.</p>	<p>AWS CLI create-policy 명령을 실행하여 Elasticsearch 작업에 대한 액세스 권한을 부여하는 테넌트 정책을 생성합니다.</p> <pre data-bbox="597 443 1027 680">aws iam create-policy \ --policy-name tenant-policy \ --policy-document file://policy.json</pre> <p>파일 <code>policy.json</code> 은(는) Elasticsearch에서 권한을 부여하는 현재 폴더의 JSON 문서입니다.</p> <pre data-bbox="597 936 1027 1860">{ "Version": "2012-10-17", "Statement": [{ "Effect": "Allow", "Action": ["es:ESHttpDelete", "es:ESHttpGet", "es:ESHttpHead", "es:ESHttpPost", "es:ESHttpPut", "es:ESHttpPatch"], "Resource": [</pre>	<p>클라우드 아키텍트, 클라우드 관리자</p>

작업	설명	필요한 기술
<p>테넌트 IAM 정책을 테넌트 역할에 연결합니다.</p>	<pre data-bbox="592 205 1027 514"> "<ARN of Elasticsearch domain created earlier>"] }] } </pre> <p>AWS CLI attach-role-policy 명령을 실행하여 이전 단계에서 생성한 두 개의 테넌트 역할에 테넌트 IAM 정책을 연결합니다.</p> <pre data-bbox="592 808 1027 1522"> aws iam attach-role-policy \ --policy-arn arn:aws:iam::account-id:policy/tenant-policy \ --role-name Tenant-1-role aws iam attach-role-policy \ --policy-arn arn:aws:iam::account-id:policy/tenant-policy \ --role-name Tenant-2-role </pre> <p>정책 ARN은 이전 단계의 출력에서 가져온 것입니다.</p>	<p>클라우드 아키텍트, 클라우드 관리자</p>

작업	설명	필요한 기술
<p>Lambda에 역할을 수임할 권한을 부여하는 IAM 정책을 생성합니다.</p>	<p>AWS CLI create-policy 명령을 실행하여 Lambda가 테넌트 역할을 맡도록 정책을 생성합니다.</p> <pre data-bbox="594 443 1027 720">aws iam create-policy \ --policy-name assume-tenant-role-policy \ --policy-document file://lambda_policy.json</pre> <p>파일 <code>lambda_policy.json</code> 은(는) <code>AssumeRole</code> 에 권한을 부여하는 현재 폴더의 JSON 문서입니다.</p> <pre data-bbox="594 978 1027 1612">{ "Version": "2012-10-17", "Statement": [{ "Effect": "Allow", "Action": "sts:AssumeRole", "Resource": "<ARN of tenant role created earlier>" }] }</pre> <p><code>Resource</code>의 경우, 와일드카드 문자를 사용하여 각 테넌트에 대해 새 정책을 만들지 않아도 됩니다.</p>	<p>클라우드 아키텍트, 클라우드 관리자</p>

작업	설명	필요한 기술
<p>Amazon S3에 액세스할 수 있는 권한을 Lambda 인덱스 역할에 부여하는 IAM 정책을 생성합니다.</p>	<p>AWS CLI create-policy 명령을 실행하여 Lambda 인덱스 역할에 S3 버킷의 객체에 액세스할 수 있는 권한을 부여합니다.</p> <pre data-bbox="592 441 1027 720">aws iam create-policy \ --policy-name s3- permission-policy \ --policy-document file://s3_lambda_p olicy.json</pre> <p>파일 <code>s3_lambda_policy.json</code> 은(는) 현재 폴더의 다음 JSON 정책 문서입니다.</p> <pre data-bbox="592 926 1027 1560">{ "Version": "2012-10-17", "Statement": [{ "Effect": "Allow", "Action": "s3:GetObject", "Resource ": "arn:aws:s3:::tena ntrawdata/*" }] }</pre>	<p>클라우드 아키텍트, 클라우드 관리자</p>

작업	설명	필요한 기술
<p>정책을 Lambda 실행 역할에 첨부합니다.</p>	<p>AWS CLI attach-role-policy 명령을 실행하여 이전 단계에서 생성한 정책을 이전에 생성한 Lambda 인덱스 및 검색 실행 역할에 연결합니다.</p> <pre data-bbox="597 491 1024 1562"> aws iam attach-role-policy \ --policy-arn arn:aws:iam::account-id:policy/assume-tenant-role-policy \ --role-name index-lambda-role aws iam attach-role-policy \ --policy-arn arn:aws:iam::account-id:policy/assume-tenant-role-policy \ --role-name search-lambda-role aws iam attach-role-policy \ --policy-arn arn:aws:iam::account-id:policy/s3-permission-policy \ --role-name index-lambda-role </pre> <p>정책 ARN은 이전 단계의 출력에서 가져온 것입니다.</p>	<p>클라우드 아키텍트, 클라우드 관리자</p>

검색 API 생성 및 구성

작업	설명	필요한 기술
<p>API 게이트웨이에서 REST API를 만듭니다.</p>	<p>CLI create-rest-api 명령을 실행하여 REST API 리소스를 생성합니다.</p> <pre data-bbox="594 499 1024 779">aws apigateway create-rest-api \ --name Test-API \ --endpoint-configuration "{ \"types\": [\"REGIONAL\"] }"</pre> <p>엔드포인트 구성 유형의 경우 특정 AWS 리전 EDGE 대신 엣지 REGIONAL 로케이션을 사용하도록 지정할 수 있습니다.</p> <p>명령 출력에서 id 필드 값을 적어 둡니다. 이는 후속 명령에서 사용할 API ID입니다.</p>	<p>클라우드 아키텍트, 클라우드 관리자</p>
<p>검색 API용 리소스를 만드세요.</p>	<p>검색 API 리소스는 리소스 이름 search을(를) 사용하여 Lambda 검색 함수를 시작합니다. (객체가 S3 버킷에 업로드될 때 자동으로 실행되므로 Lambda 인덱스 함수용 API를 생성할 필요가 없습니다.)</p> <ol style="list-style-type: none"> 1. AWS CLI get-resources 명령을 실행하여 루트 경로의 상위 ID를 가져옵니다. <pre data-bbox="630 1749 997 1843">aws apigateway get-resources \</pre>	<p>클라우드 아키텍트, 클라우드 관리자</p>

작업	설명	필요한 기술
	<pre data-bbox="634 205 1027 306">--rest-api-id <API-ID></pre> <p data-bbox="630 342 1016 474">ID 필드의 값을 기록합니다. 다음 명령에서 이 부모 ID를 사용합니다.</p> <pre data-bbox="634 510 1027 947"> { "items": [{ "id": "zpsri964ck", "path": "/" }] } </pre> <p data-bbox="591 961 1008 1236">2. AWS CLI create-resource 명령을 실행하여 검색 API 용 리소스를 생성합니다. parent-id 의 경우, 이전 명령에서 확인한 ID를 지정합니다.</p> <pre data-bbox="634 1272 1027 1591"> aws apigateway create-resource \ --rest-api-id <API-ID> \ --parent-id <Parent-ID> \ --path-part search </pre>	

작업	설명	필요한 기술
검색 API용 GET 메서드를 만드세요.	<p>AWS CLI put-method 명령을 실행하여 검색 API에 대한 GET 메서드를 생성합니다.</p> <pre data-bbox="594 394 1029 911">aws apigateway put-method \ --rest-api-id <API-ID> \ --resource-id <ID from the previous command output> \ --http-method GET \ --authorization-type "NONE" \ --no-api-key-required</pre> <p>resource-id 의 경우, create-resource 명령 출력에서 ID를 지정합니다.</p>	클라우드 아키텍트, 클라우드 관리자

작업	설명	필요한 기술
<p>검색 API에 대한 메서드 응답을 생성합니다.</p>	<p>AWS CLI put-method-response 명령을 실행하여 검색 API에 대한 메서드 응답을 추가합니다.</p> <pre data-bbox="597 443 1027 997">aws apigateway put-method-response \ --rest-api-id <API-ID> \ --resource-id <ID from the create-resource command output> \ --http-method GET \ --status-code 200 \ --response-models '{"application/json": {"Empty"}}'</pre> <p><code>resource-id</code> 의 경우, 이전 명령의 출력에서 ID를 지정합니다. <code>create-resource</code></p>	<p>클라우드 아키텍트, 클라우드 관리자</p>

작업	설명	필요한 기술
<p>검색 API에 대한 프록시 Lambda 통합을 설정합니다.</p>	<p>AWS CLI 명령 put-integration 명령을 실행하여 Lambda 검색 함수와의 통합을 설정합니다.</p> <pre data-bbox="594 394 1029 1230">aws apigateway put-integration \ --rest-api-id <API-ID> \ --resource-id <ID from the create-resource command output> \ --http-method GET \ --type AWS_PROXY \ --integration-http-method GET \ --uri arn:aws:apigateway:region:lambda:path/2015-03-31/functions/arn:aws:lambda:<region>:<account-id>:function:<function-name>/invocations</pre> <p>resource-id 의 경우, 이전 명령의 출력에서 ID를 지정합니다. create-resource</p>	<p>클라우드 아키텍트, 클라우드 관리자</p>

작업	설명	필요한 기술
<p>Lambda 검색 기능을 호출할 수 있는 API 게이트웨이 권한을 부여합니다.</p>	<p>AWS CLI 추가 권한 명령을 실행하여 API Gateway에 검색 기능을 사용할 권한을 부여합니다.</p> <pre data-bbox="597 443 1024 1079">aws lambda add-permission \ --function-name \ <function-name> \ --statement-id \ apigateway-get \ --action lambda:InvokeFunction \ --principal apigateway.amazonaws.com \ --source-arn \ "arn:aws:execute-api:<region>:<account-id>:api-id/*/GET/search</pre> <p>대신 다른 API 리소스 이름을 사용한 경우 source-arn 경로를 변경하십시오. search</p>	<p>클라우드 아키텍트, 클라우드 관리자</p>

작업	설명	필요한 기술
검색 API를 배포하세요.	<p>AWS CLI 생성-배포 명령을 실행하여 다음과 같은 이름의 스테이지 리소스를 생성합니다.</p> <p>dev</p> <pre>aws apigateway create-deployment \ --rest-api-id <API-ID> \ --stage-name dev</pre> <p>API를 업데이트하면 동일한 CLI 명령을 사용하여 동일한 단계에 다시 배포할 수 있습니다.</p>	클라우드 아키텍트, 클라우드 관리자

Kibana 역할 생성 및 구성

작업	설명	필요한 기술
Kibana 콘솔에 로그인합니다.	<ol style="list-style-type: none"> Amazon OpenSearch 서비스 콘솔의 도메인 대시보드에서 Kibana로 연결되는 링크를 찾을 수 있습니다. URL 형식은 다음과 같습니다 <domain-endpoint>/_plugin/kibana/ . 첫 번째 에픽에서 구성한 bastion host 를 사용하여 Kibana 콘솔에 액세스하세요. Amazon OpenSearch Service 도메인을 생성할 때 이전 단계에서 설명한 마스터 사용자 이름과 암호를 사 	클라우드 아키텍트, 클라우드 관리자

작업	설명	필요한 기술
	<p>용하여 Kibana 콘솔에 로그인합니다.</p> <p>4. 테넌트를 선택하라는 메시지가 표시되면 [Private] 를 선택합니다.</p>	

작업	설명	필요한 기술
Kibana 역할을 만들고 구성하세요.	<p>데이터 격리를 제공하고 한 테넌트가 다른 테넌트의 데이터를 검색할 수 없도록 하려면 테넌트가 테넌트 ID가 포함된 문서에만 액세스할 수 있도록 하는 문서 보안을 사용해야 합니다.</p> <ol style="list-style-type: none"> 1. Kibana 콘솔의 탐색 창에서 보안, 역할을 선택합니다. 2. 새 테넌트 역할을 생성합니다. 3. 클러스터 권한을 로 <code>indices_all</code> 설정하면 Amazon OpenSearch 서비스 인덱스에 대한 생성, 읽기, 업데이트 및 삭제 (CRUD) 권한이 부여됩니다. 4. 인덱스 권한을 인덱스로 제한하십시오. <code>tenant-data</code> (인덱스 이름은 Lambda 검색 및 인덱스 함수의 이름과 일치해야 합니다.) 5. 인덱스 권한을 로 설정하여 <code>indices_all</code> 사용자가 모든 인덱스 관련 작업을 수행할 수 있도록 하십시오. (요구 사항에 따라 보다 세분화된 액세스를 위해 작업을 제한할 수 있습니다.) 6. 문서 수준 보안을 위해 다음 정책을 사용하여 테넌트 ID 	클라우드 아키텍트, 클라우드 관리자

작업	설명	필요한 기술
	<p>별로 문서를 필터링하여 공유 인덱스의 테넌트에 데이터를 격리하십시오.</p> <pre data-bbox="630 380 1027 814"> { "bool": { "must": { "match": { "TenantId": "Tenant-1" } } } } </pre> <p>색인 이름, 속성 및 값은 대/소문자를 구분합니다.</p>	

작업	설명	필요한 기술
<p>사용자를 역할에 매핑합니다.</p>	<ol style="list-style-type: none"> 1. 역할에 대해 [매핑된 사용자] 탭을 선택한 다음 [사용자 매핑] 을 선택합니다. 2. 백엔드 역할 섹션에서 이전에 생성한 IAM 테넌트 역할의 ARN을 지정한 다음 Map 을 선택합니다. 이렇게 하면 IAM 테넌트 역할이 Kibana 역할에 매핑되므로 테넌트 별 검색에서 해당 테넌트에 대한 데이터만 반환됩니다. 예를 들어 테넌트-1의 IAM 역할 이름이 인 경우 테넌트-1 Kibana 역할의 백엔드 역할 상자에서 Tenant-1-Role (테넌트 역할 생성 및 구성 에픽에서) 의 ARN 을 지정합니다. Tenant-1-Role 3. 테넌트-2에 대해 1단계와 2 단계를 반복합니다. <p>테넌트 온보딩 시 테넌트 및 Kibana 역할 생성을 자동화하는 것이 좋습니다.</p>	<p>클라우드 아키텍트, 클라우드 관리자</p>

작업	설명	필요한 기술
<p>테넌트 데이터 인덱스를 생성합니다.</p>	<p>탐색 창의 관리에서 Dev Tools를 선택하고 다음 명령을 실행합니다. 이 명령은 TenantId 속성에 대한 매핑을 정의하는 tenant-data 색인을 만듭니다.</p> <pre data-bbox="597 537 1029 936"> PUT /tenant-data { "mappings": { "properties": { "TenantId": { "type": "keyword"} } } } </pre>	<p>클라우드 아키텍트, 클라우드 관리자</p>

Amazon S3와 AWS STS를 위한 VPC 엔드포인트 생성

작업	설명	필요한 기술
<p>Amazon S3용 VPC 엔드포인트를 만듭니다.</p>	<p>AWS CLI create-vpc-endpoint 명령을 실행하여 Amazon S3용 VPC 엔드포인트를 생성합니다. 엔드포인트를 사용하면 VPC의 Lambda 인덱스 함수가 Amazon S3 서비스에 액세스할 수 있습니다.</p> <pre data-bbox="597 1581 1029 1837"> aws ec2 create-vpc-endpoint \ --vpc-id <VPC-ID> \ --service-name com.amazonaws.us-east-1.s3 \ </pre>	<p>클라우드 아키텍트, 클라우드 관리자</p>

작업	설명	필요한 기술
	<pre data-bbox="592 205 1031 310">--route-table-ids <route-table-ID></pre> <p data-bbox="592 342 1031 766">의 경우 <code>vpc-id</code>, Lambda 인덱스 함수에 사용할 VPC를 지정하십시오. <code>service-name</code> 경우 Amazon S3 엔드포인트에 올바른 URL을 사용하십시오. <code>route-table-ids</code>, VPC 엔드포인트와 연결된 라우팅 테이블을 지정합니다.</p>	

작업	설명	필요한 기술
<p>AWS STS용 VPC 엔드포인트를 생성합니다.</p>	<p>AWS CLI create-vpc-endpoint 명령을 실행하여 AWS 보안 토큰 서비스 (AWS STS) 용 VPC 엔드포인트를 생성합니다. 엔드포인트를 사용하면 VPC의 Lambda 인덱스 및 검색 기능이 AWS STS 서비스에 액세스할 수 있습니다. 함수는 IAM 역할을 맡을 때 AWS STS를 사용합니다.</p> <pre data-bbox="597 730 1026 1243"> aws ec2 create-vpc-endpoint \ --vpc-id <VPC-ID> \ --vpc-endpoint-type Interface \ --service-name com.amazonaws.us-east-1.sts \ --subnet-id <subnet-ID> \ --security-group-id <security-group-ID> </pre> <p>의 경우 <code>vpc-id</code>, Lambda 인덱스 및 검색 함수에 사용할 VPC를 지정하십시오. 의 경우 <code>subnet-id</code> , 이 엔드포인트를 생성해야 하는 서브넷을 제공하십시오. 의 경우 <code>security-group-id</code> , 이 엔드포인트를 연결할 보안 그룹을 지정하십시오. (Lambda가 사용하는 보안 그룹과 같을 수 있습니다.)</p>	<p>클라우드 아키텍트, 클라우드 관리자</p>

테스트 멀티테넌시 및 데이터 격리

작업	설명	필요한 기술
<p>색인 및 검색 함수에 대한 Python 파일을 업데이트하십시오.</p>	<ol style="list-style-type: none"> 1. <code>index_lambda_package.zip</code> 파일에서 파일을 편집하여 AWS 계정 ID, AWS 리전 및 Elasticsearch 엔드포인트 정보를 업데이트합니다. <code>lambda_index.py</code> 2. <code>search_lambda_package.zip</code> 파일에서 파일을 편집하여 AWS 계정 ID, AWS 리전 및 Elasticsearch 엔드포인트 정보를 업데이트합니다. <code>lambda_search.py</code> <p>아마존 OpenSearch 서비스 콘솔의 개요 탭에서 Elasticsearch 엔드포인트를 가져올 수 있습니다. 이 리전은 <code><AWS-Region>.es.amazonaws.com</code> 의 형태로 되어 있습니다.</p>	<p>클라우드 아키텍트, 앱 개발자</p>
<p>Lambda 코드를 업데이트합니다.</p>	<p>AWS CLI update-function-code 명령을 사용하여 Python 파일의 변경 내용으로 Lambda 코드를 업데이트하십시오.</p> <pre data-bbox="592 1661 1027 1873">aws lambda update-function-code \ --function-name index-lambda-function \</pre>	<p>클라우드 아키텍트, 앱 개발자</p>

작업	설명	필요한 기술
<p>S3 버킷에 원시 데이터를 업로드합니다.</p>	<pre data-bbox="597 205 1024 709"> --zip-file fileb:// index_lambda_packag e.zip aws lambda update-fu nction-code \ --function-name search-lambda-func tion \ --zip-file fileb:// search_lambda_packa ge.zip </pre> <p data-bbox="597 741 1024 1024">AWS CLI cp 명령을 사용하여 테넌트-1 및 테넌트-2 객체의 데이터를 <code>tenantrawdata</code> 버킷에 업로드합니다 (이 용도로 생성한 S3 버킷의 이름 지정).</p> <pre data-bbox="597 1056 1024 1255"> aws s3 cp tenant-1-data s3://tenantrawdata aws s3 cp tenant-2-data s3://tenantrawdata </pre> <p data-bbox="597 1287 1024 1518">S3 버킷은 데이터가 업로드될 때마다 Lambda 인덱스 함수를 실행하도록 설정되어 있어 문서가 Elasticsearch에서 인덱싱됩니다.</p>	<p>클라우드 아키텍트, 클라우드 관리자</p>

작업	설명	필요한 기술
Kibana 콘솔에서 데이터를 검색하세요.	<p>Kibana 콘솔에서 다음 쿼리를 실행합니다.</p> <pre data-bbox="597 346 1027 466">GET tenant-data/_search</pre> <p>이 쿼리는 Elasticsearch에서 인덱싱된 모든 문서를 표시합니다. 이 경우 테넌트-1과 테넌트-2에 대한 별도의 문서 두 개가 보일 것입니다.</p>	클라우드 아키텍트, 클라우드 관리자

작업	설명	필요한 기술
<p>API Gateway에서 검색 API를 테스트합니다.</p>	<ol style="list-style-type: none"> 1. API Gateway 콘솔에서 검색 API를 열고 검색 리소스 내에서 GET 메서드를 선택한 다음 Test를 선택합니다. 2. 테스트 창에서 테넌트 ID에 대한 다음 쿼리 문자열 (대소 문자 구분) 을 제공한 다음 [Test] 를 선택합니다. <div data-bbox="630 642 1029 722" style="border: 1px solid #ccc; border-radius: 10px; padding: 5px; text-align: center; margin: 10px 0;"> TenantId=Tenant-1 </div> <p>Lambda 함수는 OpenSearch Amazon Service에 쿼리를 전송하여 문서 수준 보안을 기반으로 테넌트 문서를 필터링합니다. 메서드는 테넌트-1에 속하는 문서를 반환합니다.</p> 3. 쿼리 문자열을 다음과 같이 변경합니다. <div data-bbox="630 1222 1029 1302" style="border: 1px solid #ccc; border-radius: 10px; padding: 5px; text-align: center; margin: 10px 0;"> TenantId=Tenant-2 </div> <p>이 쿼리는 Tenant-2에 속하는 문서를 반환합니다.</p> <p>화면 그림은 추가 정보 섹션을 참조하십시오.</p>	<p>클라우드 아키텍트, 앱 개발자</p>

관련 리소스

- [AWS SDK for Python\(Boto3\)](#)
- [AWS Lambda 설명서](#)

- [Amazon API 게이트웨이 설명서](#)
- [Amazon S3 설명서](#)
- [아마존 OpenSearch 서비스 설명서](#)
 - [Amazon 서비스의 세밀한 액세스 제어 OpenSearch](#)
 - [Amazon OpenSearch 서비스를 사용하여 검색 애플리케이션 생성](#)
 - [VPC 내에서 아마존 OpenSearch 서비스 도메인 시작](#)

추가 정보

데이터 파티셔닝 모델

멀티테넌트 시스템에서 사용되는 일반적인 데이터 파티셔닝 모델은 사일로, 풀, 하이브리드의 세 가지입니다. 선택하는 모델은 환경의 규정 준수, 잡음이 많은 이웃, 운영 및 격리 요구 사항에 따라 달라집니다.

사일로 모델

사일로 모델에서 각 테넌트의 데이터는 테넌트 데이터가 섞이지 않는 별개의 스토리지 영역에 저장됩니다. Amazon OpenSearch Service에서 사일로 모델을 구현하려면 테넌트당 도메인과 테넌트당 인덱싱이라는 두 가지 접근 방식을 사용할 수 있습니다.

- 테넌트별 도메인 — 테넌트별로 별도의 Amazon OpenSearch 서비스 도메인 (Elasticsearch 클러스터와 동의어) 을 사용할 수 있습니다. 각 테넌트를 자체 도메인에 배치하면 데이터를 독립형 구조로 보유하는 것과 관련된 모든 이점을 얻을 수 있습니다. 그러나 이 접근 방식은 관리 및 민첩성 문제를 야기합니다. 분산된 특성 때문에 임차인의 운영 상태 및 활동을 집계하고 평가하기가 더 어렵습니다. 이 옵션은 비용이 많이 드는 옵션이므로 각 Amazon OpenSearch Service 도메인에 최소한 프로덕션 워크로드를 위한 마스터 노드 3개와 데이터 노드 2개가 있어야 합니다.
- 테넌트별 인덱스 — Amazon OpenSearch Service 클러스터 내의 개별 인덱스에 테넌트 데이터를 배치할 수 있습니다. 이 접근 방식을 사용하면 인덱스 이름 앞에 테넌트 식별자를 추가하여 인덱스를 생성하고 이름을 지정할 때 테넌트 식별자를 사용합니다. 테넌트당 인덱스 접근 방식을 사용하면 각 테넌트에 대해 완전히 분리된 클러스터를 도입하지 않고도 사일로 목표를 달성할 수 있습니다. 그러나 인덱스 수가 증가하면 더 많은 샤드가 필요하고 마스터 노드가 더 많은 할당과 재조정을 처리해야 하기 때문에 메모리 부족이 발생할 수 있습니다.

사일로 모델에서의 격리 - 사일로 모델에서는 IAM 정책을 사용하여 각 테넌트의 데이터를 보관하는 도메인이나 인덱스를 분리합니다. 이러한 정책은 한 테넌트가 다른 테넌트의 데이터에 액세스하는 것을 방지합니다. 사일로 격리 모델을 구현하기 위해 테넌트 리소스에 대한 액세스를 제어하는 리소스 기반 정책을 만들 수 있습니다. 이는 주로 주체가 Elasticsearch 인덱스 및 API를 비롯한 도메인의 하위 리소스에서 수행할 수 있는 작업을 지정하는 도메인 액세스 정책입니다. IAM 자격 증명 기반 정책을 사용하면 Amazon Service 내의 도메인, 인덱스 또는 API에 대한 허용 또는 거부 작업을 지정할 수 있습니다. OpenSearch IAM 정책의 Action 요소는 정책에 따라 허용되거나 거부되는 특정 작업에 대해 설명하며 Principal 요소는 영향을 받는 계정, 사용자 또는 역할을 지정합니다.

다음 샘플 정책은 Tenant-1에 도메인의 하위 리소스에만 전체 액세스 권한 (에서 지정한 대로 `es:*`) 을 부여합니다. `tenant-1 Resource` 요소의 뒤에 오는 `/*`는 이 정책이 도메인 자체에 적용되는 것이 아니라 도메인의 하위 리소스에 적용됨을 나타냅니다. 이 정책이 시행되면 테넌트는 새 도메인을 만들거나 기존 도메인의 설정을 수정할 수 없습니다.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Principal": {
        "AWS": "arn:aws:iam::aws-account-id:user/Tenant-1"
      },
      "Action": "es:*",
      "Resource": "arn:aws:es:Region:account-id:domain/tenant-1/*"
    }
  ]
}
```

인덱스별 테넌트 사일로 모델을 구현하려면 인덱스 이름을 지정하여 테넌트-1을 지정된 인덱스 또는 인덱스로 추가로 제한하도록 이 샘플 정책을 수정해야 합니다. 다음 샘플 정책은 Tenant-1을 인덱스로 제한합니다. `tenant-index-1`

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Principal": {
        "AWS": "arn:aws:iam::123456789012:user/Tenant-1"
      }
    }
  ]
}
```

```

    },
    "Action": "es:*",
    "Resource": "arn:aws:es:Region:account-id:domain/test-domain/tenant-index-1/*"
  }
]
}

```

폴 모델

폴 모델에서는 모든 테넌트 데이터가 동일한 도메인 내의 인덱스에 저장됩니다. 테넌트 식별자는 데이터(문서)에 포함되어 파티션 키로 사용되므로 어떤 데이터가 어떤 테넌트에 속하는지 확인할 수 있습니다. 이 모델은 관리 오버헤드를 줄여줍니다. 풀링된 인덱스를 운영하고 관리하는 것이 여러 인덱스를 관리하는 것보다 쉽고 효율적입니다. 그러나 테넌트 데이터가 동일한 인덱스 내에서 혼합되므로 사일로 모델이 제공하는 자연스러운 테넌트 격리가 손실됩니다. 이 접근 방식은 노이즈 인접 효과 때문에 성능을 저하시킬 수도 있습니다.

폴 모델에서의 테넌트 격리 - 일반적으로 테넌트 격리는 폴 모델에서 구현하기가 어렵습니다. 사일로 모델에 사용되는 IAM 메커니즘으로는 문서에 저장된 테넌트 ID를 기반으로 격리를 설명할 수 없습니다.

또 다른 접근 방식은 Elasticsearch용 오픈 배포판에서 제공하는 [세분화된 액세스 제어 \(FGAC\)](#) 지원을 사용하는 것입니다. FGAC를 사용하면 색인, 문서 또는 필드 수준에서 권한을 제어할 수 있습니다. FGAC는 요청이 있을 때마다 사용자 자격 증명을 평가하여 사용자를 인증하거나 액세스를 거부합니다. FGAC가 사용자를 인증하면 해당 사용자에게 매핑된 모든 역할을 가져오고 전체 권한 집합을 사용하여 요청을 처리할 방법을 결정합니다.

풀링 모델에서 필요한 격리를 달성하려면 문서 수준 보안을 사용할 수 있습니다. [문서 수준 보안](#)을 사용하면 역할을 인덱스에 있는 문서의 하위 집합으로 제한할 수 있습니다. 다음 샘플 역할은 쿼리를 Tenant-1로 제한합니다. 이 역할을 테넌트-1에 적용하면 필요한 격리를 달성할 수 있습니다.

```

{
  "bool": {
    "must": {
      "match": {
        "tenantId": "Tenant-1"
      }
    }
  }
}

```

하이브리드 모델

하이브리드 모델은 동일한 환경에서 사일로 및 풀 모델을 조합하여 각 테넌트 계층 (예: 무료, 표준 및 프리미엄 계층) 에 고유한 경험을 제공합니다. 각 계층은 풀 모델에서 사용된 것과 동일한 보안 프로필을 따릅니다.

하이브리드 모델의 테넌트 격리 - 하이브리드 모델에서는 풀 모델과 동일한 보안 프로필을 따르며, 문서 수준에서 FGAC 보안 모델을 사용하면 테넌트 격리가 제공됩니다. 이 전략은 클러스터 관리를 단순화하고 민첩성을 제공하지만 아키텍처의 다른 측면을 복잡하게 만듭니다. 예를 들어, 각 테넌트와 관련된 모델을 결정하려면 코드가 더 복잡해야 합니다. 또한 단일 테넌트 쿼리가 전체 도메인을 포화시켜 다른 테넌트의 경험을 저하시키지 않도록 해야 합니다.

API Gateway에서의 테스트

테넌트-1 쿼리의 테스트 창

테넌트-2 쿼리의 테스트 창

첨부

이 문서와 관련된 추가 콘텐츠에 액세스하려면 [attachment.zip](#) 파일의 압축을 풉니다.

AWS CDK를 사용하여 다중 스택 애플리케이션을 배포할 수 있습니다. TypeScript

작성자: Dr. Rahul Sharad Gaikwad(AWS)

환경: 프로덕션

기술: 현대화, 마이그레이션,
DevOps

워크로드: 기타 모든 워크로드

AWS 서비스: Amazon API
Gateway; AWS Lambda;
Amazon Kinesis

요약

이 패턴은 AWS Cloud Development Kit (AWS CDK) 와 함께 TypeScript 사용하여 Amazon Web Services (AWS) 에 애플리케이션을 배포하기 위한 step-by-step 접근 방식을 제공합니다. 예를 들어, 이 패턴은 서버리스 실시간 분석 애플리케이션을 배포합니다.

이 패턴은 중첩된 스택 애플리케이션을 빌드하고 배포합니다. 상위 AWS CloudFormation 스택은 하위 또는 중첩된 스택을 호출합니다. 각 하위 스택은 CloudFormation 스택에 정의된 AWS 리소스를 빌드하고 배포합니다. 명령줄 인터페이스 (CLI) cdk 명령인 AWS CDK Toolkit은 스택의 기본 인터페이스입니다. CloudFormation

사전 조건 및 제한 사항

사전 조건

- 활성 상태의 AWS 계정.
- 기존 Virtual Private Cloud(VPC) 및 서브넷
- AWS CDK Toolkit이 설치 및 구성됨
- 관리자 권한이 있는 사용자 및 액세스 키 세트.
- Node.js
- AWS Command Line Interface(AWS CLI)

제한 사항

- AWS CDK는 AWS를 CloudFormation 사용하기 때문에 AWS CDK 애플리케이션에는 CloudFormation 서비스 할당량이 적용됩니다. 자세한 내용은 [AWS CloudFormation 할당량을 참조](#) 하십시오.

제품 버전

이 패턴은 다음 도구 및 버전을 사용하여 구축 및 테스트되었습니다.

- AWS CDK Toolkit 1.83.0
- Node.js 14.13.0
- npm 7.0.14

이 패턴은 모든 버전의 AWS CDK 또는 npm에서 작동해야 합니다. 참고로 Node.js 버전 13.0.0부터 13.6.0까지는 AWS CDK와 호환되지 않습니다.

아키텍처

대상 기술 스택

- AWS Amplify Console
- Amazon API Gateway
- AWS CDK
- 아마존 CloudFront
- Amazon Cognito
- Amazon DynamoDB
- 아마존 데이터 파이어호스
- Amazon Kinesis Data Streams
- AWS Lambda
- Amazon Simple Storage Service (S3)

대상 아키텍처

다음 다이어그램은 AWS CDK를 사용한 다중 스택 애플리케이션 배포를 보여줍니다. TypeScript

다음 다이어그램은 예제 서버리스 실시간 애플리케이션의 아키텍처를 나타냅니다.

도구

도구

- [AWS Amplify 콘솔](#)은 AWS에서의 풀스택 웹 및 모바일 애플리케이션 배포를 위한 제어 센터입니다. Amplify Console 호스팅은 지속적인 배포로 풀스택 서버리스 웹 앱을 호스팅하기 위한 Git 기반 워크플로를 제공합니다. 관리 UI는 프론트엔드 웹 및 모바일 개발자가 AWS Console 외부에서 앱 백엔드를 만들고 관리할 수 있는 시각적 인터페이스입니다.
- [Amazon API Gateway](#)는 모든 규모에서 REST, HTTP 및 WebSocket API를 생성, 게시, 유지 관리, 모니터링 및 보호하는 AWS 서비스입니다.
- [AWS Cloud Development Kit\(AWS CDK\)](#)는 AWS 클라우드 인프라를 코드로 정의하고 프로비저닝하는 데 도움이 되는 소프트웨어 개발 프레임워크입니다.
- [AWS CDK 툴킷](#)은 AWS CDK 앱과 상호 작용하는 데 도움이 되는 명령줄 클라우드 개발 키트입니다. cdk CLI 명령은 AWS CDK 앱과 상호 작용하는 기본 도구입니다. 앱을 실행하고, 정의한 애플리케이션 모델을 조사하고, AWS CDK에서 생성된 AWS CloudFormation 템플릿을 생성 및 배포합니다.
- [CloudFrontAmazon](#)은 .html, .css, .js 및 이미지 파일과 같은 정적 및 동적 웹 콘텐츠의 배포 속도를 높이는 웹 서비스입니다. CloudFront 지연 시간을 줄이고 성능을 향상시키기 위해 엣지 로케이션이라고 하는 전 세계 데이터 센터 네트워크를 통해 콘텐츠를 전송합니다.
- [Amazon Cognito](#)는 웹 및 모바일 앱에 대한 인증, 권한 부여 및 사용자 관리를 제공합니다. 사용자가 직접 로그인하거나 타사를 통해 로그인할 수 있습니다.
- [Amazon DynamoDB](#)는 완전관리형 NoSQL 데이터베이스 서비스로서 원활한 확장성과 함께 빠르고 예측 가능한 성능을 제공합니다.
- [Amazon Data Firehose](#)는 Amazon S3, Amazon Redshift, Amazon 서비스, Splunk 및 지원되는 타사 서비스 공급자가 소유한 사용자 지정 HTTP 엔드포인트 또는 HTTP 엔드포인트와 같은 대상에 실시간 [스트리밍 데이터](#)를 전송하는 완전 관리형 서비스입니다. OpenSearch
- [Amazon Kinesis Data Streams](#)는 대규모 데이터 레코드 스트림을 실시간으로 수집하고 처리하는 서비스입니다.
- [AWS Lambda](#)는 서버를 프로비저닝하거나 관리하지 않고도 코드를 실행할 수 있게 해주는 컴퓨팅 서비스입니다. Lambda는 필요 시에만 코드를 실행하며, 일일 몇 개의 요청에서 초당 수천 개의 요청

까지 자동으로 규모를 조정합니다. 사용한 컴퓨팅 시간만큼만 비용을 지불하고, 코드가 실행되지 않을 때는 요금이 부과되지 않습니다.

- [Amazon Simple Storage Service\(S3\)](#)는 원하는 양의 데이터를 저장, 보호 및 검색하는 데 도움이 되는 클라우드 기반 객체 스토리지 서비스입니다.

코드

이 패턴의 코드가 첨부되어 있습니다.

에픽

AWS CDK Toolkit 설치

작업	설명	필요한 기술
AWS CDK Toolkit을 설치합니다.	AWS CDK Toolkit을 전역적으로 설치하려면 다음 명령을 실행합니다. <code>npm install -g aws-cdk</code>	DevOps
버전을 확인합니다.	AWS CDK Toolkit 버전을 확인하려면 다음 명령을 실행하세요. <code>cdk --version</code>	DevOps

AWS 보안 인증 설정

작업	설명	필요한 기술
보안 인증을 설정합니다.	보안 인증 정보를 설정하려면 <code>aws configure</code> 명령을 실행하고 프롬프트를 따르세요. <pre>\$aws configure AWS Access Key ID [None]:</pre>	DevOps

작업	설명	필요한 기술
	<pre>AWS Secret Access Key [None]: your_secret_access_key Default region name [None]: Default output format [None]:</pre>	

프로젝트 코드 다운로드

작업	설명	필요한 기술
첨부된 프로젝트 코드를 다운로드합니다.	디렉토리 및 파일 구조에 대한 자세한 내용은 추가 정보 섹션을 참고하십시오.	DevOps

AWS CDK 환경 부트스트랩

작업	설명	필요한 기술
환경을 부트스트랩합니다.	<p>사용하려는 계정 및 AWS 지역에 AWS CloudFormation 템플릿을 배포하려면 다음 명령을 실행합니다.</p> <pre>cdk bootstrap <account>/<Region></pre> <p>자세한 내용은 AWS 설명서를 참조하십시오.</p>	DevOps

프로젝트 구축 및 배포

작업	설명	필요한 기술
프로젝트를 빌드합니다.	프로젝트 코드를 빌드하려면 <code>npm run build</code> 명령을 실행하세요.	DevOps
프로젝트를 배포합니다.	프로젝트 코드를 배포하려면 <code>cdk deploy</code> 명령을 실행하세요.	

출력 확인

작업	설명	필요한 기술
스택 생성을 확인합니다.	AWS 관리 콘솔에서 을 선택합니다 CloudFormation. 프로젝트의 스택에서 상위 스택과 두 개의 하위 스택이 생성되었는지 확인합니다.	DevOps

애플리케이션 테스트

작업	설명	필요한 기술
Kinesis Data Streams로 데이터를 전송합니다.	Amazon Kinesis Data Generator(KDG)를 사용하여 Kinesis Data Streams으로 데이터를 전송하도록 AWS 계정을 구성합니다. 자세한 내용은 Amazon Kinesis Data Generator 를 참조하세요.	DevOps
Amazon Cognito 사용자를 생성합니다.	Amazon Cognito 사용자를 생성하려면 Kinesis 데이터 생성	DevOps

작업	설명	필요한 기술
	<p>기 도움말 페이지의 Amazon Cognito 사용자 생성 섹션에서 cognito-setup.json CloudFormation 템플릿을 다운로드 하십시오. 템플릿을 시작한 다음 Amazon Cognito 사용자 이름과 암호를 입력합니다.</p> <p>출력 탭에는 Kinesis Data Generator URL이 나열됩니다.</p>	
Kinesis Data Generator에 로그인	KDG에 로그인하려면 제공한 Amazon Cognito 보안 인증 정보와 Kinesis Data Generator URL을 사용하십시오.	DevOps
애플리케이션을 테스트합니다.	KDG의 레코드 템플릿, 템플릿 1에서 추가 정보 섹션의 테스트 코드를 붙여넣고 데이터 보내기를 선택합니다.	DevOps
API Gateway를 테스트합니다.	데이터를 수집한 후에는 GET 메서드를 사용하여 데이터를 검색하여 API Gateway를 테스트합니다.	DevOps

관련 리소스

참조

- [AWS Cloud Development Kit](#)
- [AWS CDK 기반 GitHub](#)
- [중첩 스택 작업](#)
- [AWS 샘플 예제 — 서버리스 실시간 분석](#)

추가 정보

디렉터리 및 파일 세부 정보

이 패턴은 다음 세 개의 스택을 설정합니다.

- `parent-cdk-stack.ts` — 이 스택은 상위 스택 역할을 하며 두 하위 애플리케이션을 중첩된 스택으로 호출합니다.
- `real-time-analytics-poc-stack.ts` — 이 중첩된 스택에는 인프라 및 애플리케이션 코드가 포함됩니다.
- `real-time-analytics-web-stack.ts` — 이 중첩된 스택에는 정적 웹 애플리케이션 코드만 포함됩니다.

중요 파일 및 해당 기능

- `bin/real-time-analytics-poc.ts` — AWS CDK 애플리케이션의 엔트리 포인트. `lib/`에서 정의된 모든 스택을 로드합니다.
- `lib/real-time-analytics-poc-stack.ts` - AWS CDK 애플리케이션 스택의 정의(`real-time-analytics-poc`).
- `lib/real-time-analytics-web-stack.ts` - AWS CDK 애플리케이션 스택의 정의(`real-time-analytics-web-stack`).
- `lib/parent-cdk-stack.ts` - AWS CDK 애플리케이션 스택의 정의(`parent-cdk`).
- `package.json` — 애플리케이션 이름, 버전 및 종속성을 포함하는 npm 모듈 매니페스트.
- `package-lock.json` — npm에서 유지 관리합니다.
- `cdk.json` — 애플리케이션 실행을 위한 툴킷.
- `tsconfig.json` — 프로젝트 TypeScript 구성.
- `.gitignore` — Git이 소스 제어에서 제외해야 하는 파일 목록입니다.
- `node_modules` — npm에서 유지 관리합니다. 프로젝트의 종속성을 포함합니다.

상위 스택의 다음 코드 섹션은 하위 애플리케이션을 중첩된 AWS CDK 스택으로 호출합니다.

```
import * as cdk from '@aws-cdk/core';
import { Construct, Stack, StackProps } from '@aws-cdk/core';
import { RealTimeAnalyticsPocStack } from './real-time-analytics-poc-stack';
import { RealTimeAnalyticsWebStack } from './real-time-analytics-web-stack';
```

```

export class CdkParentStack extends Stack {
  constructor(scope: Construct, id: string, props?: StackProps) {
    super(scope, id, props);

    new RealTimeAnalyticsPocStack(this, 'RealTimeAnalyticsPocStack');
    new RealTimeAnalyticsWebStack(this, 'RealTimeAnalyticsWebStack');
  }
}

```

테스트용 코드

```

session={{date.now('YYYYMMDD')}}|sequence={{date.now('x')}}|
reception={{date.now('x')}}|instrument={{random.number(9)}}|
l={{random.number(20)}}|price_0={{random.number({"min":10000,
"max":30000})}}|price_1={{random.number({"min":10000, "max":30000})}}|
price_2={{random.number({"min":10000, "max":30000})}}|
price_3={{random.number({"min":10000, "max":30000})}}|
price_4={{random.number({"min":10000, "max":30000})}}|
price_5={{random.number({"min":10000, "max":30000})}}|
price_6={{random.number({"min":10000, "max":30000})}}|
price_7={{random.number({"min":10000, "max":30000})}}|
price_8={{random.number({"min":10000, "max":30000})}}|

```

API Gateway 테스트

API Gateway 콘솔에서 GET 메서드를 사용하여 API Gateway 콘솔을 테스트합니다.

첨부

이 문서와 관련된 추가 콘텐츠에 액세스하려면 [attachment.zip](#) 파일의 압축을 풉니다.

AWS SAM을 사용하여 중첩된 애플리케이션 자동 배포

작성자: Dr. Rahul Sharad Gaikwad(AWS), Dmitry Gulin (AWS), Ishwar Chauthaiwale (AWS), 및 Tabby Ward (AWS)

코드 리포지토리: -샘플 aws-sam-nested-stack	환경: PoC 또는 파일럿	기술: 현대화, 서버리스, DevOps
워크로드: 기타 모든 워크로드	AWS 서비스: AWS Serverless Application Repository	

요약

Amazon Web Services(AWS)에서 AWS Serverless Application Model(AWS SAM)은 함수, API, 데이터베이스 및 이벤트 소스 매핑을 표현하는 간편 구문을 제공하는 오픈 소스 프레임워크입니다. 각 리소스에 대해 단 몇 줄이면 원하는 애플리케이션을 정의하고 YAML을 사용하여 이를 모델링할 수 있습니다. 배포 중에 SAM은 SAM 구문을 서버리스 애플리케이션을 더 빠르게 구축하는 데 사용할 수 있는 AWS CloudFormation 구문으로 변환 및 확장합니다.

AWS SAM은 AWS 플랫폼에서 서버리스 애플리케이션의 개발, 배포 및 관리를 간소화합니다. 표준화된 프레임워크, 더 빠른 배포, 로컬 테스트 기능, 리소스 관리, 개발 도구와의 원활한 통합, 지원 커뮤니티를 제공합니다. 이러한 기능을 통해 서버리스 애플리케이션을 효율적이고 효과적으로 구축하는 데 유용한 도구가 됩니다.

이 패턴은 AWS SAM 템플릿을 사용하여 중첩된 애플리케이션 배포를 자동화합니다. 중첩 애플리케이션은 다른 애플리케이션 내에 있는 애플리케이션입니다. 상위 애플리케이션은 하위 애플리케이션을 호출합니다. 이들은 서버리스 아키텍처의 느슨하게 결합된 구성 요소입니다.

중첩된 애플리케이션을 사용하면 독립적으로 작성 및 유지 관리되지만 AWS SAM과 Serverless Application Repository를 사용하여 구성된 서비스 또는 구성 요소를 재사용하여 매우 정교한 서버리스 아키텍처를 빠르게 구축할 수 있습니다. 중첩된 애플리케이션을 사용하면 더 강력한 애플리케이션을 구축하고, 중복 작업을 방지하고, 팀과 조직 전체에서 일관성과 모범 사례를 보장할 수 있습니다. 중첩된 애플리케이션을 시연하기 위해 패턴은 [예제 AWS 서버리스 쇼핑 카트 애플리케이션](#)을 배포합니다.

사전 조건 및 제한 사항

사전 조건

- 활성 상태의 AWS 계정
- 기존 Virtual Private Cloud(VPC) 및 서브넷
- AWS Cloud9 또는 Visual Studio Code와 같은 통합 개발 환경(자세한 내용은 [AWS 기반 구축 도구 참조](#))
- Pip 설치 유틸을 사용하여 Python 유틸 라이브러리 설치됨(아직 설치되지 않은 경우)

제한 사항

- 서버리스 애플리케이션에 중첩될 수 있는 최대 애플리케이션 수는 200개입니다.
- 중첩된 애플리케이션의 최대 파라미터 수는 60개일 수 있습니다.

제품 버전

- 이 솔루션은 AWS SAM 명령줄 인터페이스(AWS SAM CLI) 버전 1.21.1을 기반으로 구축되었지만, 이 아키텍처는 이후 AWS SAM CLI 버전에서도 작동해야 합니다.

아키텍처

대상 기술 스택

- Amazon API Gateway
- AWS SAM
- Amazon Cognito
- Amazon DynamoDB
- AWS Lambda
- Amazon Simple Queue Service(Amazon SQS) 대기열

대상 아키텍처

다음 다이어그램은 API를 호출하여 쇼핑 서비스에 사용자 요청을 전송하는 방법을 보여줍니다. 필요한 모든 정보를 포함한 사용자 요청은 Amazon API Gateway와 Amazon Cognito 권한 부여자로 전송되며, 권한 부여자는 API에 대한 인증 및 권한 부여 메커니즘을 수행합니다.

DynamoDB에서 항목이 추가, 삭제 또는 업데이트되면 이벤트가 DynamoDB Stream에 전달되면 Lambda 함수가 시작됩니다. 동기식 워크플로우의 일부로 오래된 항목이 즉시 삭제되는 것을 방지하기 위해 메시지가 SQS 대기열로 전송됩니다. 이는 메시지를 삭제하는 워커 함수를 시작합니다.

이 솔루션 설정에서 AWS SAM CLI는 AWS 스택의 인터페이스 역할을 합니다. CloudFormation AWS SAM 템플릿은 중첩된 애플리케이션을 자동으로 배포합니다. 상위 SAM 템플릿은 하위 템플릿을 호출하고 상위 CloudFormation 스택은 하위 스택을 배포합니다. 각 하위 스택은 AWS SAM CloudFormation 템플릿에 정의된 AWS 리소스를 구축합니다.

1. 스택을 빌드하여 배포합니다.
2. 인증 CloudFormation 스택에는 Amazon Cognito가 포함되어 있습니다.
3. 제품 CloudFormation 스택에는 Lambda 함수와 Amazon API Gateway가 포함되어 있습니다.
4. 쇼핑 CloudFormation 스택에는 Lambda 함수, Amazon API Gateway, SQS 대기열, 아마존 DynamoDB 데이터베이스가 포함되어 있습니다.

도구

도구

- [Amazon API Gateway](#)를 사용하면 어떤 규모에서든 REST, HTTP 및 WebSocket API를 생성, 게시, 유지 관리, 모니터링 및 보호할 수 있습니다.
- [AWS는 AWS CloudFormation](#) 리소스를 설정하고, 빠르고 일관되게 프로비저닝하고, AWS 계정 및 지역 전반의 수명 주기 전반에 걸쳐 리소스를 관리할 수 있도록 지원합니다.
- [Amazon Cognito](#)는 웹 및 모바일 앱에 대한 인증, 권한 부여 및 사용자 관리를 제공합니다.
- [Amazon DynamoDB](#)는 빠르고 예측 가능하고 확장 가능한 성능을 제공하는 완전 관리형 NoSQL 데이터베이스 서비스입니다.
- [AWS Lambda](#)는 서버를 프로비저닝하거나 관리할 필요 없이 코드를 실행하는 데 도움이 되는 컴퓨팅 서비스입니다. 필요할 때만 코드를 실행하며 자동으로 확장이 가능하므로 사용한 컴퓨팅 시간만큼만 비용을 지불합니다.
- [AWS Serverless Application Model\(AWS SAM\)](#)은 AWS 클라우드에서 서버리스 애플리케이션을 빌드하는 데 사용할 수 있는 오픈 소스 프레임워크입니다.
- [Amazon Simple Queue Service\(Amazon SQS\)](#)는 내구력 있고 가용성이 뛰어난 보안 호스팅 대기열을 제공하며 이를 통해 분산 소프트웨어 시스템과 구성 요소를 통합 및 분리할 수 있습니다.

code

이 패턴의 코드는 GitHub [AWS SAM 중첩 스택 샘플](#) 리포지토리에서 사용할 수 있습니다.

에픽

AWS SAM CLI 설치

작업	설명	필요한 기술
AWS SAM CLI를 설치합니다.	AWS SAM CLI를 설치하려면 AWS SAM 설명서 의 지침을 참조하세요.	DevOps 엔지니어
AWS 보안 인증을 설정합니다.	<p>AWS SAM CLI가 사용자를 대신하여 AWS 서비스를 호출할 수 있도록 AWS 보안 인증을 설정하려면 <code>aws configure</code> 명령을 실행하고 프롬프트를 따릅니다.</p> <pre> \$aws configure AWS Access Key ID [None]: <your_access_key_id> AWS Secret Access Key [None]: your_secret_access_key Default region name [None]: Default output format [None]: </pre> <p>보안 인증 설정에 대한 자세한 내용은 인증 및 액세스 보안 인증을 참조하세요.</p>	DevOps 엔지니어

AWS SAM 프로젝트 초기화

작업	설명	필요한 기술
AWS SAM 코드 리포지토리를 복제합니다.	<p>1. 다음 명령을 입력하여 이 패턴의 aws sam 중첩 스택 샘플 리포지토리를 복제합니다.</p> <pre>git clone https://github.com/aws-samples/aws-sam-nested-stack-sample.git</pre> <p>2. 다음 명령을 입력하여 복제된 디렉터리로 이동합니다.</p> <pre>cd aws-sam-nested-stack-sample</pre>	DevOps 엔지니어
템플릿을 배포하여 프로젝트를 초기화합니다.	<p>프로젝트를 초기화하려면 SAM init 명령을 실행합니다. 템플릿 소스를 선택하라는 메시지가 표시되면 Custom Template Location을(를) 선택합니다.</p>	DevOps 엔지니어

SAM 템플릿 코드 컴파일 및 빌드

작업	설명	필요한 기술
AWS SAM 애플리케이션 템플릿을 검토합니다.	<p>중첩된 애플리케이션의 템플릿을 검토합니다. 이 예제에서는 다음과 같은 중첩된 애플리케이션 템플릿을 사용합니다.</p> <ul style="list-style-type: none"> auth.yaml - 이 템플릿은 Amazon Cognito 및 	DevOps 엔지니어

작업	설명	필요한 기술
	<p>AWS Systems Manager Parameter Store와 같은 인증 관련 리소스를 설정합니다.</p> <ul style="list-style-type: none"> • <code>product-mock.yaml</code> - 이 템플릿은 Lambda 함수 및 Amazon API Gateway와 같은 제품 관련 리소스를 배포합니다. • <code>shoppingcart-service.yaml</code> - 이 템플릿은 AWS Identity and Access Management(IAM), DynamoDB 테이블 및 Lambda 함수와 같은 장바구니 관련 리소스 쇼핑을 설정합니다. 	
<p>상위 템플릿을 검토합니다.</p>	<p>중첩된 애플리케이션 템플릿을 호출할 템플릿을 검토합니다. 이 예제에서 상위 템플릿은 <code>template.yml</code> 입니다. 모든 개별 애플리케이션은 단일 상위 <code>template.yml</code> 템플릿에 중첩됩니다.</p>	<p>DevOps 엔지니어</p>
<p>AWS SAM 템플릿 코드를 컴파일 및 빌드합니다.</p>	<p>AWS SAM CLI를 사용하여 다음 명령을 실행합니다.</p> <div style="border: 1px solid #ccc; border-radius: 15px; padding: 10px; margin: 10px 0;"> <pre>sam build</pre> </div>	<p>DevOps 엔지니어</p>

AWS SAM 템플릿 배포

작업	설명	필요한 기술
애플리케이션을 배포합니다.	<p>중첩된 애플리케이션 CloudFormation 스택을 생성하고 AWS 환경에 코드를 배포하는 SAM 템플릿 코드를 실행하려면 다음 명령을 실행합니다.</p> <pre>sam deploy --guided --stack-name shopping-cart-nested-stack --capabilities CAPABILITY_IAM CAPABILITY_AUTO_EXPAND</pre> <p>명령을 실행하면 몇 가지 질문이 표시됩니다. 모든 질문에 y(으)로 답변합니다.</p>	DevOps 엔지니어

배포 확인

작업	설명	필요한 기술
스택을 확인합니다.	<p>AWS SAM 템플릿에 정의된 AWS CloudFormation 스택과 AWS 리소스를 검토하려면 다음과 같이 하십시오.</p> <ol style="list-style-type: none"> 1. AWS 관리 콘솔에 로그인하고 콘솔로 이동합니다 CloudFormation. 2. 상위 스택과 하위 스택이 나열되어 있는지 확인하세요. 	DevOps 엔지니어

작업	설명	필요한 기술
	<p>이 예제에서 sam-shopping-cart 은(는) 중첩된 Auth, Product 및 Shopping 스택을 호출하는 상위 스택입니다.</p> <p>제품 스택은 Product API Gateway URL 링크를 출력으로 제공합니다.</p>	

관련 리소스

참조

- [AWS Serverless Application Model\(AWS SAM\)](#)
- [AWS SAM 커짐 GitHub](#)
- [Serverless Shopping Cart Microservice](#) (AWS 예제 애플리케이션)

자습서 및 동영상

- [서버리스 앱 구축](#)
- [AWS 온라인 테크 토크: AWS SAM을 사용한 서버리스 애플리케이션 구축 및 배포](#)

추가 정보

코드가 모두 준비되면 예제의 디렉터리 구조는 다음과 같습니다.

- [sam_stacks](#) - 이 폴더에는 shared.py 계층이 포함되어 있습니다. 계층은 라이브러리, 사용자 지정 런타임 또는 기타 종속 항목을 포함하는 파일 아카이브입니다. 배포 패키지에 라이브러리를 포함시킬 필요 없이 계층을 통해 함수에서 라이브러리를 사용할 수 있습니다.
- product-mock-service— 이 폴더에는 모든 제품 관련 Lambda 함수 및 파일이 들어 있습니다.
- shopping-cart-service— 이 폴더에는 쇼핑과 관련된 모든 Lambda 함수 및 파일이 들어 있습니다.

AWS Lambda 토큰 벤딩 머신을 사용하여 Amazon S3에 대한 SaaS 테넌트 격리를 구현합니다.

작성자: Tabby Ward(AWS), Sravan Periyathambi(AWS), Thomas Davis(AWS)

환경: PoC 또는 파일럿

기술: 현대화; SaaS

AWS 서비스: AWS Identity 및 Access Management, AWS Lambda, Amazon S3, AWS STS

요약

멀티테넌트 SaaS 애플리케이션은 테넌트 격리가 유지되도록 시스템을 구현해야 합니다. 동일한 Amazon Web Services(AWS) 리소스에 테넌트 데이터를 저장하는 경우(예: 여러 테넌트가 동일한 Amazon Simple Storage Service(S3) 버킷에 데이터를 저장하는 경우) 테넌트 간 액세스가 발생하지 않도록 해야 합니다. 토큰 자판기(TVM)는 테넌트 데이터 격리를 제공하는 한 가지 방법입니다. 이러한 머신은 토큰 생성 방식의 복잡성을 추상화하는 동시에 토큰을 획득할 수 있는 메커니즘을 제공합니다. 개발자는 토큰 생성 방법에 대한 자세한 지식 없이도 TVM을 사용할 수 있습니다.

이 패턴은 AWS Lambda를 사용하여 TVM을 구현합니다. TVM은 S3 버킷의 단일 SaaS 테넌트 데이터에 대한 액세스를 제한하는 임시 보안 토큰 서비스(STS) 보안 인증 정보로 구성된 토큰을 생성합니다.

TVM과 이 패턴으로 제공되는 코드는 일반적으로 JSON 웹 토큰(JWT)에서 파생된 클레임과 함께 사용되어 AWS 리소스에 대한 요청을 테넌트 범위의 AWS Identity 및 Access Management(IAM) 정책과 연결합니다. 기본적으로 이 패턴의 코드를 사용하여 JWT 토큰에 제공되는 클레임을 기반으로 범위가 지정된 임시 STS 보안 인증 정보를 생성하는 SaaS 애플리케이션을 구현할 수 있습니다.

사전 조건 및 제한 사항

사전 조건

- 활성 상태의 AWS 계정.
- macOS, Linux 또는 Windows에 설치 및 구성된 AWS Command Line Interface(AWS CLI) 버전 [1.19.0 이상](#) 또는 AWS CLI [버전 2.1 이상](#)을 사용할 수 있습니다.

제한 사항

- 이 코드는 Java에서 실행되며 현재 다른 프로그래밍 언어를 지원하지 않습니다.
- 샘플 애플리케이션에는 AWS 교차 리전 또는 재해 복구(DR) 지원이 포함되어 있지 않습니다.
- 이 패턴은 SaaS 애플리케이션용 Lambda TVM이 범위 지정 테넌트 액세스를 제공하는 방법을 보여줍니다. 프로덕션 환경에서 사용하기 위한 것이 아닙니다.

아키텍처

대상 기술 스택

- AWS Lambda
- Amazon S3
- IAM
- AWS Security Token Service (AWS STS)

대상 아키텍처

도구

서비스

- [AWS Command Line Interface\(AWS CLI\)](#)는 명령줄 셸에서 명령을 사용하여 AWS 서비스와 상호 작용할 수 있는 오픈 소스 도구입니다.
- [AWS Identity and Access Management\(IAM\)](#)는 사용자에게 대한 인증 및 권한 부여를 제어함으로써 AWS 리소스에 대한 액세스를 안전하게 관리할 수 있습니다.
- [AWS Lambda](#)는 서버를 프로비저닝하거나 관리할 필요 없이 코드를 실행하는 데 도움이 되는 컴퓨팅 서비스입니다. 필요할 때만 코드를 실행하며 자동으로 확장이 가능하므로 사용한 컴퓨팅 시간만큼만 비용을 지불합니다.
- [AWS Security Token Service\(AWS STS\)](#)를 사용하면 사용자를 위한 제한된 권한의 임시 보안 인증 정보를 요청할 수 있습니다.
- [Amazon Simple Storage Service\(S3\)](#)는 원하는 양의 데이터를 저장, 보호 및 검색하는 데 도움이 되는 클라우드 기반 객체 스토리지 서비스입니다.

코드

이 패턴의 소스 코드는 첨부 파일로 제공되며 다음과 같은 파일이 포함됩니다.

- s3UploadSample.jar은 JSON 문서를 S3 버킷에 업로드하는 Lambda 함수의 소스 코드를 제공합니다.
- tvmlayer.zip(은)는 Lambda 함수가 S3 버킷에 액세스하고 JSON 문서를 업로드할 수 있도록 토큰(STS 임시 보안 인증 정보)을 제공하는 재사용 가능한 Java 라이브러리를 제공합니다.
- token-vending-machine-sample-app.zip은 이러한 아티팩트와 컴파일 지침을 생성하는 데 사용되는 소스 코드를 제공합니다.

이러한 파일을 사용하려면 다음 섹션의 지침을 따르세요.

에픽

변수 값 결정

작업	설명	필요한 기술
변수 값을 결정합니다.	<p>이 패턴의 구현에는 일관되게 사용해야 하는 여러 변수 이름이 포함됩니다. 각 변수에 사용해야 하는 값을 결정하고 이후 단계에서 요청할 경우 해당 값을 제공하세요.</p> <p><AWS 계정 ID> – 이 패턴을 구현하려는 AWS 계정과 연결된 12자리 계정 ID. AWS 계정 ID를 찾는 방법에 대한 자세한 내용은 IAM 설명서에서 AWS 계정 ID 및 별칭을 참조하세요.</p> <p><AWS 리전> – 이 패턴을 구현하고 있는 AWS 리전. AWS 리전에 대한 자세한 내용은 AWS 웹사이트의 리전 및 가용 영역을 참조하세요.</p>	클라우드 관리자

작업	설명	필요한 기술
	<p>< sample-tenant-name > – 애플리케이션에서 사용할 테넌트의 이름. 단순화를 위해 이 값에는 영숫자만 사용하는 것이 좋지만 S3 객체 키에는 어떤 유효한 이름이라도 사용할 수 있습니다.</p> <p>< sample-tvm-role-name > – TVM 및 샘플 애플리케이션을 실행하는 Lambda 함수에 연결된 IAM 역할의 이름. 역할 이름은 공백 없이 대문자 및 소문자 영숫자로 구성된 문자열입니다. 밑줄(_), 더하기 기호(+), 등호(=), 쉼표(,), 마침표(.), 골뱅이 기호(@) 및 하이픈(-) 문자를 포함할 수도 있습니다. 역할 이름은 계정 내에서 고유해야 합니다.</p> <p>< sample-app-role-name > – 범위가 지정된 임시 STS 자격 증명을 생성할 때 Lambda 함수가 맡는 IAM 역할의 이름입니다. 역할 이름은 공백 없이 대문자 및 소문자 영숫자로 구성된 문자열입니다. 밑줄(_), 더하기 기호(+), 등호(=), 쉼표(,), 마침표(.), 골뱅이 기호(@) 및 하이픈(-) 문자를 포함할 수도 있습니다. 역할 이름은 계정 내에서 고유해야 합니다.</p> <p>< sample-app-function-name > – Lambda 함수의 이름. 이 문</p>	

작업	설명	필요한 기술
	<p>자열은 길이가 최대 64자입니다.</p> <p>< sample-app-bucket-name > - 특정 테넌트로 범위가 지정된 권한으로 액세스해야 하는 S3 버킷의 이름. S3 버킷 이름:</p> <ul style="list-style-type: none"> • 3~63자 이내로 작성합니다. • 소문자, 숫자, 마침표(.) 및 하이픈(-)만 포함해야 합니다. • 문자나 숫자로 시작하고 끝나야 합니다. • IP 주소 형식(예: 192.168.5.4)을 사용할 수 없습니다. • 파티션 내에서 고유해야 합니다. 파티션은 리전의 그룹입니다. AWS는 현재 aws (표준 지역), (중국 지역) 및 aws-cn aws-us-gov (AWS GovCloud [미국] 지역) 이라는 세 개의 파티션을 가지고 있습니다. 	

S3 버킷 생성

작업	설명	필요한 기술
샘플 애플리케이션을 위한 S3 버킷을 생성합니다.	다음 AWS CLI 명령을 사용하여 S3 버킷을 생성합니다. 코드 스니펫에 < sample-app-bucket-name > 값을 입력하십시오.	클라우드 관리자

작업	설명	필요한 기술
	<pre>aws s3api create-bucket --bucket <sample-app- bucket-name></pre> <p>Lambda 샘플 애플리케이션은 JSON 파일을 이 버킷에 업로드합니다.</p>	

IAM TVM 역할 및 정책 생성

작업	설명	필요한 기술
TVM 역할을 생성합니다.	<p>다음 AWS CLI 명령 중 하나를 사용하여 IAM 역할을 생성합니다. 명령에 < sample-tvm-role-name > 값을 입력합니다.</p> <p>macOS 또는 Linux 셸의 경우:</p> <pre>aws iam create-role \ --role-name <sample-t vm-role-name> \ --assume-role-policy- document '{ "Version": "2012-10-17", "Statement": [{ "Effect": "Allow", "Principa l": { "Service": "lambda.a mazonaws.com" }, "Action": "sts:AssumeRole"</pre>	클라우드 관리자

작업	설명	필요한 기술
	<pre data-bbox="594 205 1027 310"> }]}' </pre> <p data-bbox="594 342 941 380">Windows 명령줄의 경우:</p> <pre data-bbox="594 415 1027 1014"> aws iam create-role ^ --role-name <sample-t vm-role-name> ^ --assume-role-policy- document "{\"Versi on\": \"2012-10 -17\", \"Statement \": [{\"Effect\": \"Allow\", \"Princip al\": {\"Service\": \"lambda.amazonaws .com\"}, \"Action\": \"sts:AssumeRole\" }]}" </pre> <p data-bbox="594 1045 1027 1367">Lambda 샘플 애플리케이션은 애플리케이션이 호출될 때 이 역할을 말합니다. 범위 지정 정책으로 애플리케이션 역할을 맡을 수 있는 기능은 코드에 S3 버킷에 액세스할 수 있는 더 넓은 권한을 부여합니다.</p>	

작업	설명	필요한 기술
<p>인라인 TVM 역할 정책을 생성합니다.</p>	<p>다음 AWS CLI 명령 중 하나를 사용하여 IAM 정책을 생성합니다. <AWS Account ID>명령에 < sample-tvm-role-name >, 및 < sample-app-role-name > 값을 입력합니다.</p> <p>macOS 또는 Linux 셸의 경우:</p> <pre>aws iam put-role-policy \ --role-name <sample-tvm-role-name> \ --policy-name assume-app-role \ --policy-document '{ "Version": "2012-10-17", "Statement": [{ "Effect": "Allow", "Action": "sts:AssumeRole", "Resource": "arn:aws:iam::<AWS Account ID>:role/<sample-app-role-name>" }]}'</pre> <p>Windows 명령줄의 경우:</p> <pre>aws iam put-role-policy ^ --role-name <sample-tvm-role-name> ^</pre>	<p>클라우드 관리자</p>

작업	설명	필요한 기술
	<pre data-bbox="597 212 1024 779"> --policy-name assume-app-role ^ --policy-document '{"Version": "2012-10-17", "Statement": [{"Effect": "Allow", "Action": "sts:AssumeRole", "Resource": "arn:aws:iam::<AWS Account ID>:role/<sample-app-role-name>"}]}' </pre> <p data-bbox="597 821 1024 1136">이 정책은 TVM 역할에 연결되어 있습니다. 이는 코드에 애플리케이션 역할을 맡을 수 있는 권한을 부여하며, 애플리케이션 역할은 S3 버킷에 액세스할 수 있는 더 넓은 권한을 가집니다.</p>	

작업	설명	필요한 기술
<p>관리형 Lambda 정책을 연결합니다.</p>	<p>다음 AWS CLI 명령을 사용하여 AWSLambdaBasicExecutionRole IAM 정책에 연결합니다. 명령에 < sample-tvm-role-name > 값을 입력합니다.</p> <pre data-bbox="594 537 1029 898">aws iam attach-role-policy \ --role-name <sample-tvm-role-name> \ --policy-arn arn:aws:iam::aws:policy/service-role/AWSLambdaBasicExecutionRole</pre> <p>Windows 명령줄의 경우:</p> <pre data-bbox="594 1010 1029 1371">aws iam attach-role-policy ^ --role-name <sample-tvm-role-name> ^ --policy-arn arn:aws:iam::aws:policy/service-role/AWSLambdaBasicExecutionRole</pre> <p>이 관리형 정책은 Lambda가 Amazon에 로그를 전송할 수 있도록 TVM 역할에 연결됩니다. CloudWatch</p>	<p>클라우드 관리자</p>

IAM 애플리케이션 역할 및 정책 생성

작업	설명	필요한 기술
<p>애플리케이션 역할을 생성합니다.</p>	<p>다음 AWS CLI 명령 중 하나를 사용하여 IAM 역할을 생성합니다. <AWS Account ID>명령에 < sample-app-role-name >, 및 < sample-tvm-role-name > 값을 입력합니다.</p> <p>macOS 또는 Linux 셸의 경우:</p> <pre>aws iam create-role \ --role-name <sample-app-role-name> \ --assume-role-policy-document '{ "Version": "2012-10-17", "Statement": [{ "Effect": "Allow", "Principal": { "AWS": "arn:aws:iam::<AWS Account ID>:role/<sample-tvm-role-name>" }, "Action": "sts:AssumeRole" }]}'</pre> <p>Windows 명령줄의 경우:</p> <pre>aws iam create-role ^</pre>	<p>클라우드 관리자</p>

작업	설명	필요한 기술
	<pre data-bbox="609 212 1015 781"> --role-name <sample-app-role-name> ^ --assume-role-policy-document "{\"Version\": \"2012-10-17\", \"Statement\": [{\"Effect\": \"Allow\", \"Principal\": {\"AWS\": \"arn:aws:iam::<AWS Account ID>:role/<sample-tvm-role-name>\"}, \"Action\": \"sts:AssumeRole\"}]}" </pre> <p data-bbox="592 821 1031 997">Lambda 샘플 애플리케이션은 S3 버킷에 대한 테넌트 기반 액세스를 얻기 위해 범위 지정 정책을 통해 이 역할을 맡습니다.</p>	

작업	설명	필요한 기술
<p>인라인 애플리케이션 역할 정책을 생성합니다.</p>	<p>다음 AWS CLI 명령 중 하나를 사용하여 IAM 정책을 생성합니다. 명령에 < sample-app-role-name > 및 < sample-app-bucket-name > 값을 입력합니다.</p> <p>macOS 또는 Linux 셸의 경우:</p> <pre data-bbox="597 617 1029 1862">aws iam put-role-policy \ --role-name <sample-app-role-name> \ --policy-name s3-bucket-access \ --policy-document '{ "Version": "2012-10-17", "Statement": [{ "Effect": "Allow", "Action": ["s3:PutObject", "s3:GetObject", "s3:DeleteObject"], "Resource": "arn:aws:s3:::<sample-app-bucket-name>/*" }, { "Effect": "Allow", "Action": ["s3:ListBucket"],</pre>	<p>클라우드 관리자</p>

작업	설명	필요한 기술
	<pre data-bbox="597 205 1024 426"> "Resource ": "arn:aws:s3:::<sample-app-bucket-name>" }]]'</pre> <p data-bbox="597 457 943 499">Windows 명령줄의 경우:</p> <pre data-bbox="597 531 1024 1570"> aws iam put-role-policy ^ --role-name <sample-app-role-name> ^ --policy-name s3-bucket-access ^ --policy-document "{\"Version\": \"2012-10-17\", \"Statement\": [{\"Effect\": \"Allow\", \"Action\": [\"s3:PutObject\", \"s3:GetObject\", \"s3:DeleteObject\"], \"Resource\": \"arn:aws:s3:::<sample-app-bucket-name>/*\"}, {\"Effect\": \"Allow\", \"Action\": [\"s3:ListBucket\"], \"Resource\": \"arn:aws:s3:::<sample-app-bucket-name>\"}]}"</pre> <p data-bbox="597 1602 1024 1837">이 정책은 애플리케이션 역할에 연결되어 있습니다. S3 버킷에 있는 객체에 대한 폭넓은 액세스를 제공합니다. 샘플 애플리케이션이 역할을 맡으면</p>	

작업	설명	필요한 기술
	TVM의 동적으로 생성되는 정책에 따라 이러한 권한의 범위가 특정 테넌트로 제한됩니다.	

TVM을 사용하여 Lambda 샘플 애플리케이션 생성

작업	설명	필요한 기술
컴파일된 소스 파일을 다운로드합니다.	첨부 파일로 포함된 s3UploadSample.jar 및 tvm-layer.zip 파일을 다운로드합니다. 이러한 아티팩트를 만드는 데 사용된 소스 코드와 컴파일 지침은 token-vending-machine-sample-app.zip 에 제공됩니다.	클라우드 관리자
Lambda 레이어를 생성합니다.	다음 AWS CLI 명령을 사용하여 Lambda 계층을 생성하면 Lambda가 TVM에 액세스할 수 있습니다. 참고: tvm-layer.zip 을 다운로드한 위치에서 이 명령을 실행하지 않는 경우 --zip-file 파라미터의 tvm-layer.zip 에 올바른 경로를 제공하세요.	클라우드 관리자, 앱 개발자

```
aws lambda publish-layer-version \
  --layer-name sample-token-vending-machine \
  --compatible-runtimes java11 \
```

작업	설명	필요한 기술
	<pre data-bbox="597 205 1026 310">--zip-file fileb://t vm-layer.zip</pre> <p data-bbox="597 340 941 382">Windows 명령줄의 경우:</p> <pre data-bbox="597 415 1026 772">aws lambda publish-l ayer-version ^ --layer-name sample-to ken-vending-machine ^ --compatible-runtimes java11 ^ --zip-file fileb://t vm-layer.zip</pre> <p data-bbox="597 806 993 940">이 명령은 재사용 가능한 TVM 라이브러리를 포함하는 Lambda 계층을 생성합니다.</p>	

작업	설명	필요한 기술
<p>Lambda 함수를 생성합니다.</p>	<p>다음 AWS CLI 명령을 사용하여 Lambda 함수를 생성합니다. <AWS Account ID><AWS Region>명령에 < sample-app-function-name >, < sample-tvm-role-name >, < sample-app-bucket-name >, < sample-app-role-name > 및 < > 값을 입력합니다.</p> <p>참고: s3UploadSample.jar 를 다운로드한 위치에서 이 명령을 실행하지 않는 경우 --zip-file 파라미터의 s3UploadSample.jar 에 올바른 경로를 입력하세요.</p> <pre data-bbox="592 1050 1031 1850">aws lambda create-function \ --function-name <sample-app-function-name> \ --timeout 30 \ --memory-size 256 \ --runtime java11 \ --role arn:aws:iam::<AWS Account ID>:role/<sample-tvm-role-name> \ --handler com.amazonaws.s3UploadSample.App \ --zip-file fileb://s3UploadSample.jar \ --layers arn:aws:lambda:<AWS Region>:<AWS Account ID>:layer</pre>	<p>클라우드 관리자, 앱 개발자</p>

작업	설명	필요한 기술
	<pre data-bbox="609 210 998 577">:sample-token-vending-machine:1 \ --environment "Variables={S3_BUCKET=<sample-app-bucket-name>, ROLE=arn:aws:iam::<AWS Account ID>:role/<sample-app-role-name>}"</pre> <p data-bbox="592 619 941 661">Windows 명령줄의 경우:</p> <pre data-bbox="609 714 998 1774">aws lambda create-function ^ --function-name <sample-app-function-name> ^ --timeout 30 ^ --memory-size 256 ^ --runtime java11 ^ --role arn:aws:iam::<AWS Account ID>:role/<sample-tvm-role-name> ^ --handler com.amazonaws.s3UploadSample.App ^ --zip-file fileb://s3UploadSample.jar ^ --layers arn:aws:lambda:<AWS Region>:<AWS Account ID>:layer:sample-token-vending-machine:1 ^ --environment "Variables={S3_BUCKET=<sample-app-bucket-name>,ROLE=arn:aws:iam::<AWS Account</pre>	

작업	설명	필요한 기술
	<pre data-bbox="594 205 1024 306">ID>:role/<sample-app-role-name>}"</pre> <p data-bbox="594 342 1024 758">이 명령은 샘플 애플리케이션 코드와 TVM 계층이 연결된 Lambda 함수를 생성합니다. 또한 두 개의 환경 변수인 S3_BUCKET 및 ROLE을 설정합니다. 샘플 애플리케이션은 이러한 변수를 사용하여 맡을 역할과 JSON 문서를 업로드할 S3 버킷을 결정합니다.</p>	

샘플 애플리케이션 및 TVM 테스트

작업	설명	필요한 기술
<p data-bbox="115 1056 535 1136">Lambda 샘플 애플리케이션을 호출합니다.</p>	<p data-bbox="594 1056 1024 1377">다음 AWS CLI 명령 중 하나를 사용하여 예상 페이로드로 Lambda 샘플 애플리케이션을 시작합니다. 명령에 < sample-app-function-name > 및 < sample-tenant-name > 값을 입력합니다.</p> <p data-bbox="594 1419 979 1455">macOS 및 Linux 셸의 경우:</p> <pre data-bbox="594 1493 1024 1822">aws lambda invoke \ --function <sample-app-function-name> \ --invocation-type RequestResponse \ --payload '{"tenant": "<sample-tenant-name>"}' \</pre>	<p data-bbox="1073 1056 1455 1092">클라우드 관리자, 앱 개발자</p>

작업	설명	필요한 기술
	<pre data-bbox="592 210 1031 346">--cli-binary-format raw-in-base64-out response.json</pre> <p data-bbox="592 378 1031 420">Windows 명령줄의 경우:</p> <pre data-bbox="592 451 1031 934">aws lambda invoke ^ --function <sample-app-function-name> ^ --invocation-type RequestResponse ^ --payload "{\"tenant \": \"<sample-tenant-name>\"}" ^ --cli-binary-format raw-in-base64-out response.json</pre> <p data-bbox="592 966 1031 1386">이 명령은 Lambda 함수를 호출하고 결과를 <code>response.json</code> 문서에 반환합니다. 대부분의 UNIX 기반 시스템에서는 다른 파일을 생성하지 않고도 결과를 셸에 직접 출력하도록 <code>response.json</code> 을 <code>/dev/stdout</code> 으로 변경할 수 있습니다.</p> <p data-bbox="592 1428 1031 1659">참고: 이후에 이 Lambda 함수를 호출할 때 <code><sample-tenant-name></code> 값을 변경하면 JSON 문서의 위치와 토큰이 제공하는 권한이 변경됩니다.</p>	

작업	설명	필요한 기술
S3 버킷을 보면 생성된 객체를 확인할 수 있습니다.	이전에 생성한 S3 버킷 (< sample-app-bucket-name >) 을 찾아보십시오. 이 버킷에는 값이 < sample-tenant-name > 인 S3 객체 접두사가 포함되어 있습니다. 이 접두사 아래에는 UUID로 이름이 지정된 JSON 문서가 있습니다. 샘플 애플리케이션을 여러 번 호출하면 더 많은 JSON 문서가 추가됩니다.	클라우드 관리자

작업	설명	필요한 기술
<p>샘플 애플리케이션의 Cloudwatch 로그를 볼 수 있습니다.</p>	<p>이름이 < >인 Lambda 함수와 관련된 클라우드워치 로그를 확인하십시오. sample-app-function-name 지침은 AWS Lambda 설명서에서 AWS Lambda의 아마존 CloudWatch 로그에 액세스하기를 참조 하십시오. 이 로그에서 TVM이 생성한 테넌트 범위 정책을 확인할 수 있습니다. 이 테넌트 범위 정책은 샘플 애플리케이션에 대한 권한을 Amazon S3 PutObject, GetObject DeleteObject, 및 ListBucketAPI에 부여하지만 < >와 관련된 객체 접두사에 대해서만 권한을 부여합니다. sample-tenant-name 이후 샘플 애플리케이션을 호출할 때 < sample-tenant-name >를 변경하면 TVM은 호출 페이로드에 제공된 테넌트에 대응하도록 범위 지정 정책을 업데이트합니다. 동적으로 생성되는 이 정책은 SaaS 애플리케이션에서 TVM을 통해 테넌트 범위 액세스를 유지하는 방법을 보여줍니다.</p> <p>TVM 기능은 Lambda 계층에서 제공되므로 코드를 복제하지 않고도 애플리케이션에서 사용하는 다른 Lambda 함수에 연결할 수 있습니다.</p>	<p>클라우드 관리자</p>

작업	설명	필요한 기술
	동적으로 생성되는 정책에 대한 설명은 추가 정보 섹션을 참조하세요.	

관련 리소스

- [동적으로 생성된 IAM 정책으로 테넌트 격리](#)(블로그 게시물)
- [SaaS 환경에서 동적으로 생성된 격리 정책 적용](#)(블로그 게시물)
- [AWS SaaS Boost](#)(SaaS 제품을 AWS로 이전하는 데 도움이 되는 오픈 소스 참조 환경)

추가 정보

다음 Amazon CloudWatch Log는 TVM 코드에 따라 이러한 패턴으로 생성된 동적으로 생성된 정책을 보여줍니다. 이 스크린샷에서 < >는 이고 < sample-app-bucket-name >는 입니다 **DOC-EXAMPLE-BUCKET**. sample-tenant-name test-tenant-1 지정된 이 범위 정책에서 반환된 STS 보안 인증은 객체 키 접두사 test-tenant-1와 연결된 객체를 제외하고 S3 버킷의 객체에 대해 어떠한 작업도 수행할 수 없습니다.

첨부

이 문서와 관련된 추가 콘텐츠에 액세스하려면 [attachment.zip](#) 파일의 압축을 풉니다.

AWS Step Functions을 사용하여 서버리스 사가 패턴 구현

작성자: Tabby Ward(AWS), Rohan Mehta(AWS), Rimpay Tewani(AWS)

환경: PoC 또는 파일럿

기술: 현대화, 서버리스, 클라우드는 네이티브

워크로드: 오픈 소스

AWS 서비스: Amazon API Gateway, Amazon DynamoDB, AWS Lambda, Amazon SNS, AWS Step Functions

요약

마이크로서비스 아키텍처의 주요 목표는 분리되고 독립적인 구성 요소를 구축하여 애플리케이션의 민첩성, 유연성을 높이고 출시 시간을 단축하는 것입니다. 디커플링의 결과로 각 마이크로서비스 구성 요소에는 자체 데이터 지속성 계층이 있습니다. 분산 아키텍처에서는 비즈니스 트랜잭션이 여러 마이크로서비스에 걸쳐 있을 수 있습니다. 이러한 마이크로서비스는 단일 원자성, 일관성, 격리성, 내구성(ACID) 트랜잭션을 사용할 수 없으므로 부분 트랜잭션으로 끝날 수 있습니다. 이 경우 이미 처리된 트랜잭션을 취소하려면 일부 제어 로직이 필요합니다. 분산 사가 패턴은 일반적으로 이 목적에 사용됩니다.

사가 패턴은 분산 애플리케이션의 일관성을 유지하고 여러 마이크로서비스 간의 트랜잭션을 조정하여 데이터 일관성을 유지하는 데 도움이 되는 장애 관리 패턴입니다. 사가 패턴을 사용하면 트랜잭션을 수행하는 모든 서비스가 이벤트를 게시하여 후속 서비스가 체인에서 다음 트랜잭션을 수행하도록 트리거합니다. 이는 체인의 마지막 거리가 완료될 때까지 계속됩니다. 비즈니스 트랜잭션이 실패할 경우, saga는 이전 트랜잭션에서 이루어진 변경 사항을 취소하는 일련의 보상 트랜잭션을 조정합니다.

이 패턴은 AWS Step Functions, AWS Lambda 및 Amazon DynamoDB와 같은 서버리스 기술을 사용하여 (여행 예약을 처리하는) 샘플 애플리케이션의 설정 및 배포를 자동화하는 방법을 보여줍니다. 또한 샘플 애플리케이션에서는 사가 실행 코디네이터를 구현하기 위해 Amazon API Gateway와 Amazon Simple Notification Service(SNS)를 사용합니다. 패턴은 AWS Cloud Development Kit(AWS CDK), AWS Serverless Application Model(AWS SAM) 또는 Terraform과 같은 코드형 인프라(IaC) 프레임워크와 함께 배포할 수 있습니다.

사가 패턴 및 기타 데이터 지속성 패턴에 대한 자세한 내용은 AWS 권장 가이드 웹사이트의 [마이크로 서비스에서 데이터 지속성 활성화](#) 가이드를 참조하세요.

사전 조건 및 제한 사항

사전 조건

- 활성 상태의 AWS 계정.
- AWS CloudFormation 스택을 생성할 수 있는 권한. 자세한 내용은 CloudFormation 설명서의 [액세스 제어](#)를 참조하십시오.
- 원하는 IaC 프레임워크(AWS CDK, AWS SAM 또는 Terraform)를 AWS 계정으로 구성하면 프레임 워크 CLI를 사용하여 애플리케이션을 배포할 수 있습니다.
- NodeJS는 애플리케이션을 빌드하고 로컬에서 실행하는 데 사용됩니다.
- 원하는 코드 편집기(예: 비주얼 스튜디오 코드, 서브라임 또는 아톰)

제품 버전

- [NodeJS 버전 14](#)
- [AWS CDK 버전 2.37.1](#)
- [AWS SAM 버전 1.71.0](#)
- [Terraform 버전 1.3.7](#)

제한 사항

이벤트 소싱은 모든 구성 요소가 느슨하게 결합되어 있고 서로에 대한 직접적인 지식이 없는 마이크로 서비스 아키텍처에서 사가 오케스트레이션 패턴을 구현하는 자연스러운 방법입니다. 트랜잭션이 적은 단계(3~5개)를 포함하는 경우 사가 패턴이 매우 적합할 수 있습니다. 그러나 마이크로서비스 수와 단계 수에 따라 복잡성이 증가합니다.

이 디자인을 사용하면 트랜잭션 패턴을 시뮬레이션하기 위해 모든 서비스를 실행해야 하므로 테스트 및 디버깅이 어려울 수 있습니다.

아키텍처

대상 아키텍처

제안된 아키텍처는 AWS Step Functions를 사용하여 항공편을 예약하고, 렌터카를 예약하고, 휴가에 대한 결제를 처리하는 사가 패턴을 구축합니다.

다음 워크플로 다이어그램은 여행 예약 시스템의 일반적인 흐름을 보여줍니다. 워크플로는 항공 여행 예약 (“ReserveFlight”), 자동차 예약 (“ReserveCarRental”), 결제 처리 (“ProcessPayment”), 항공편 예약 확인 (“ConfirmFlight”), 렌터카 확인 (“ConfirmCarRental”) 으로 구성되며, 이 단계가 완료되면 성공 알림이 표시됩니다. 그러나 시스템에서 이러한 트랜잭션을 실행하는 중에 오류가 발생하면 역방향으로 실패하기 시작합니다. 예를 들어 결제 처리 중 오류 (“ProcessPayment”) 가 발생하면 환불 (“RefundPayment”) 이 시작되고, 그러면 렌터카 및 항공편 (“CancelRentalReservation” 및 “CancelFlightReservation”) 이 취소되고, 실패 메시지와 함께 전체 거래가 종료됩니다.

이 패턴은 다이어그램에 강조 표시된 각 작업에 대해 별도의 Lambda 함수를 배포하고 항공편, 렌터카 및 결제를 위한 3개의 DynamoDB 테이블을 배포합니다. 각 Lambda 함수는 트랜잭션이 확인되었는지 또는 롤백되었는지에 따라 각 DynamoDB 테이블에서 행을 생성, 업데이트 또는 삭제합니다. 이 패턴은 Amazon SNS를 사용하여 구독자에게 트랜잭션 실패 또는 성공을 알리는 문자(SMS) 메시지를 전송합니다.

자동화 및 규모 조정

IaC 프레임워크 중 하나를 사용하여 이 아키텍처의 구성을 생성할 수 있습니다. 원하는 IaC에 대한 다음 링크 중 하나를 사용합니다.

- [AWS CDK로 배포하기](#)
- [AWS SAM으로 배포하기](#)
- [Terraform으로 배포하기](#)

도구

서비스

- [AWS Step Functions](#)는 AWS Lambda 함수와 기타 AWS 서비스를 결합할 수 있는 서버리스 오케스트레이션 서비스로, 비즈니스 크리티컬 애플리케이션을 구축합니다. Step Functions 그래픽 콘솔을 통해 애플리케이션의 워크플로를 일련의 이벤트 기반 단계로 볼 수 있습니다.
- [Amazon DynamoDB](#)는 원활한 확장성과 함께 빠르고 예측 가능한 성능을 제공하는 완전 관리형 NoSQL 데이터베이스 서비스입니다. DynamoDB를 사용하여 데이터 규모에 관계없이 데이터를 저장 및 검색하고, 어떤 수준의 요청 트래픽이라도 처리할 수 있는 데이터베이스 테이블을 생성할 수 있습니다.

- [AWS Lambda](#)는 서버를 프로비저닝하거나 관리하지 않고도 코드를 실행할 수 있게 해주는 컴퓨팅 서비스입니다. Lambda는 필요 시에만 코드를 실행하며, 일일 몇 개의 요청에서 초당 수천 개의 요청까지 자동으로 규모를 조정합니다.
- [Amazon API Gateway](#)는 모든 규모에서 REST, HTTP 및 WebSocket API를 생성, 게시, 유지 관리, 모니터링 및 보호하는 AWS 서비스입니다.
- [Amazon Simple Notification Service\(SNS\)](#)는 게시자에서 구독자에게 메시지를 전송하는 관리형 서비스입니다.
- [AWS Cloud Development Kit \(AWS CDK\)](#) 는 Python, Java TypeScript JavaScript, C#.Net 같은 친숙한 프로그래밍 언어를 사용하여 클라우드 애플리케이션 리소스를 정의하기 위한 소프트웨어 개발 프레임워크입니다.
- [AWS Serverless Application Model\(AWS SAM\)](#)은 서버리스 애플리케이션을 빌드하는 데 사용할 수 있는 오픈소스 프레임워크입니다. 함수, API, 데이터베이스 및 이벤트 소스 매핑을 표현하는 속기 구문을 제공합니다.

코드

IaC 템플릿(AWS CDK, AWS SAM 또는 Terraform), Lambda 함수 및 DynamoDB 테이블을 포함하여 사가 패턴을 보여주는 샘플 애플리케이션의 코드는 다음 링크에서 찾을 수 있습니다. 이를 설치하려면 첫 번째 에픽의 지침을 따르세요.

- [AWS CDK로 배포하기](#)
- [AWS SAM으로 배포하기](#)
- [Terraform으로 배포하기](#)

에픽

패키지, 컴파일, 빌드 설치

작업	설명	필요한 기술
NPM 패키지를 설치합니다.	새 디렉토리를 생성하고 터미널에서 해당 디렉토리로 이동한 다음 이 패턴 앞부분의 코드 섹션에서 원하는 GitHub 리포지토리를 복제합니다.	개발자, 클라우드 아키텍트

작업	설명	필요한 기술
	<p>package.json 파일이 있는 루트 폴더에서 다음 명령을 실행하여 모든 Node Package Manager(NPM) 패키지를 다운로드하고 설치합니다.</p> <pre data-bbox="594 474 1027 554">npm install</pre>	
스크립트를 컴파일합니다.	<p>루트 폴더에서 다음 명령을 실행하여 TypeScript 트랜스파일러가 필요한 모든 파일을 만들도록 지시합니다. JavaScript</p> <pre data-bbox="594 810 1027 890">npm run build</pre>	개발자, 클라우드 아키텍트
변경 사항을 확인하고 다시 컴파일하세요.	<p>루트 폴더에서 별도의 터미널 창에서 다음 명령을 실행하여 코드 변경을 관찰하고 변경 사항이 감지되면 코드를 컴파일합니다.</p> <pre data-bbox="594 1188 1027 1268">npm run watch</pre>	개발자, 클라우드 아키텍트
유닛 테스트를 실행합니다 (AWS CDK만 해당).	<p>AWS CDK를 사용하는 경우 루트 폴더에서 다음 명령을 실행하여 Jest 유닛 테스트를 수행하세요.</p> <pre data-bbox="594 1524 1027 1604">npm run test</pre>	개발자, 클라우드 아키텍트

대상 AWS 계정에 리소스 배포

작업	설명	필요한 기술
<p>데모 스택을 AWS에 배포합니다.</p>	<p>중요: 애플리케이션은 AWS 리전에 구애받지 않습니다. 프로파일을 사용하는 경우 AWS Command Line Interface(AWS CLI) 프로파일 또는 AWS CLI 환경 변수를 통해 리전을 명시적으로 선언해야 합니다.</p> <p>루트 폴더에서 다음 명령을 실행하여 배포 어셈블리를 생성하고 이를 기본 AWS 계정 및 리전에 배포합니다.</p> <p>AWS CDK:</p> <pre>cdk bootstrap cdk deploy</pre> <p>AWS SAM:</p> <pre>sam build sam deploy --guided</pre> <p>Terraform:</p> <pre>terraform init terraform apply</pre> <p>이 단계를 완료하는 데 몇 분 정도 걸릴 수 있습니다. 이 명령은 AWS CLI용으로 구성된 기본 보안 인증 정보를 사용합니다.</p>	<p>개발자, 클라우드 아키텍트</p>

작업	설명	필요한 기술
	배포가 완료된 후 콘솔에 표시되는 API Gateway URL을 기록합니다. 사가 실행 흐름을 테스트하려면 이 정보가 필요합니다.	
배포된 스택을 현재 상태와 비교합니다.	루트 폴더에서 다음 명령을 실행하여 소스 코드를 변경한 후 배포된 스택을 현재 상태와 비교합니다. AWS CDK: <pre>cdk diff</pre> AWS SAM: <pre>sam deploy</pre> Terraform: <pre>terraform plan</pre>	개발자, 클라우드 아키텍트

실행 흐름 테스트

작업	설명	필요한 기술
사가 실행 흐름을 테스트합니다.	이전 단계에서 스택을 배포할 때 기록해 둔 API Gateway URL로 이동합니다. 이 URL은 상태 머신 시작을 트리거합니다. 다양한 URL 파라미터를 전달하여 상태 머신의 흐름을 조작하는 방법에 대한 자세한 내	개발자, 클라우드 아키텍트

작업	설명	필요한 기술
	<p>용은 추가 정보 섹션을 참조하세요.</p> <p>결과를 보려면 AWS Management Console에 로그인 후 Step Functions 콘솔로 이동합니다. 여기에서 사가 스테이트 머신의 모든 단계를 볼 수 있습니다. 또한 DynamoDB 테이블을 보고 삽입, 업데이트 또는 삭제된 레코드를 확인할 수 있습니다. 화면을 자주 새로고침하면 트랜잭션 상태가 pending에서 confirmed 로 변경되는 것을 볼 수 있습니다.</p> <p>예약 성공 또는 실패 시 SMS 메시지를 수신하도록 휴대폰 번호로 stateMachine.ts 파일의 코드를 업데이트하여 SNS 주제를 구독할 수 있습니다. 자세한 내용은 추가 정보 섹션의 Amazon SNS를 참조하세요.</p>	

정리

작업	설명	필요한 기술
리소스를 정리합니다.	<p>이 응용 프로그램에 배포된 리소스를 정리하려면 다음 명령 중 하나를 사용하시면 됩니다.</p> <p>AWS CDK:</p>	앱 개발자, 클라우드 아키텍트

작업	설명	필요한 기술
	<div data-bbox="597 212 1029 289" style="border: 1px solid #ccc; border-radius: 10px; padding: 5px; margin-bottom: 10px;"><code>cdk destroy</code></div> <p data-bbox="597 327 751 359">AWS SAM:</p> <div data-bbox="597 401 1029 478" style="border: 1px solid #ccc; border-radius: 10px; padding: 5px; margin-bottom: 10px;"><code>sam delete</code></div> <p data-bbox="597 516 740 548">Terraform:</p> <div data-bbox="597 590 1029 667" style="border: 1px solid #ccc; border-radius: 10px; padding: 5px;"><code>terraform destroy</code></div>	

관련 리소스

기술 문서

- [AWS에서 마이크로서비스 구현](#)
- [서버리스 애플리케이션 렌즈](#)
- [마이크로서비스의 데이터 지속성 지원](#)

AWS 서비스 설명서

- [AWS SDK 시작하기](#)
- [AWS SAM으로 시작하기](#)
- [AWS Step Functions](#)
- [Amazon DynamoDB](#)
- [AWS Lambda](#)
- [Amazon API Gateway](#)
- [Amazon SNS](#)

자습서

- [서버리스 컴퓨팅 실습 워크숍](#)

추가 정보

코드

테스트 목적으로 이 패턴은 API Gateway와 Step Functions 상태 머신을 트리거하는 테스트 Lambda 함수를 배포합니다. Step Functions를 사용하면 “,” “,” “”,” “” ReserveFlightReserveCarRental, “” 등의 오류를 모방하는 run_type 매개변수를 전달하여 여행 예약 시스템의 기능을 제어할 수 ConfirmCarRental 있습니다. ProcessPayment ConfirmFlight

saga Lambda 함수(sagaLambda.ts)는 API Gateway URL의 쿼리 파라미터에서 입력을 받아 다음 JSON 객체를 생성하고 실행을 위해 Step Functions에 전달합니다.

```
let input = {
  "trip_id": tripID, // value taken from query parameter, default is AWS request ID
  "depart_city": "Detroit",
  "depart_time": "2021-07-07T06:00:00.000Z",
  "arrive_city": "Frankfurt",
  "arrive_time": "2021-07-09T08:00:00.000Z",
  "rental": "BMW",
  "rental_from": "2021-07-09T00:00:00.000Z",
  "rental_to": "2021-07-17T00:00:00.000Z",
  "run_type": runType // value taken from query parameter, default is "success"
};
```

다음 URL 파라미터를 전달하여 Step Functions 상태 머신의 다양한 흐름을 시험해 볼 수 있습니다.

- 성공적인 실행 – `https://{api gateway url}`
- 항공편 예약 실패 – `https://{api 게이트웨이 url}? 런타임= failFlightsReservation`
- 비행 실패 확인 – `https://{api 게이트웨이 URL}? 런타임= failFlightsConfirmation`
- 렌터카 예약 실패 – `https://{api 게이트웨이 URL}? 실행 유형= 예약 failCarRental`
- 렌터카 실패 확인 – `https://{api 게이트웨이 URL}? 실행 유형= 확인 failCarRental`
- 결제 처리 실패 – `https://{api gateway url}?runType=failPayment`
- 트립 ID 전달하기 – `https://{api gateway url}?tripID={by default, trip ID will be the AWS request ID}`

IaC 템플릿

연결된 리포지토리에는 전체 샘플 여행 예약 애플리케이션을 생성하는 데 사용할 수 있는 IaC 템플릿이 포함되어 있습니다.

- [AWS CDK로 배포하기](#)
- [AWS SAM으로 배포하기](#)
- [Terraform으로 배포하기](#)

DynamoDB 테이블

항공편, 렌터카, 결제 테이블의 데이터 모델은 다음과 같습니다.

Flight Data Model:

```
var params = {
  TableName: process.env.TABLE_NAME,
  Item: {
    'pk' : {S: event.trip_id},
    'sk' : {S: flightReservationID},
    'trip_id' : {S: event.trip_id},
    'id': {S: flightReservationID},
    'depart_city' : {S: event.depart_city},
    'depart_time': {S: event.depart_time},
    'arrive_city': {S: event.arrive_city},
    'arrive_time': {S: event.arrive_time},
    'transaction_status': {S: 'pending'}
  }
};
```

Car Rental Data Model:

```
var params = {
  TableName: process.env.TABLE_NAME,
  Item: {
    'pk' : {S: event.trip_id},
    'sk' : {S: carRentalReservationID},
    'trip_id' : {S: event.trip_id},
    'id': {S: carRentalReservationID},
    'rental': {S: event.rental},
    'rental_from': {S: event.rental_from},
    'rental_to': {S: event.rental_to},
    'transaction_status': {S: 'pending'}
  }
};
```

Payment Data Model:

```
var params = {
  TableName: process.env.TABLE_NAME,
```

```

    Item: {
      'pk' : {S: event.trip_id},
      'sk' : {S: paymentID},
      'trip_id' : {S: event.trip_id},
      'id': {S: paymentID},
      'amount': {S: "750.00"}, // hard coded for simplicity as implementing any
      monetary transaction functionality is beyond the scope of this pattern
      'currency': {S: "USD"},
      'transaction_status': {S: "confirmed"}
    }
  };

```

Lambda 함수

Step Functions에서 상태 머신 플로우 및 실행을 지원하기 위해 다음 함수가 생성됩니다.

- 항공편 예약: 항공편을 예약하기 위해 DynamoDB 항공편 테이블에 pending의 transaction_status에 대한 레코드를 삽입합니다.
- 항공편 확인: DynamoDB Flights 테이블의 기록을 업데이트하여 transaction_status에서 confirmed으로 설정하고 항공편을 확인합니다.
- 항공편 예약 취소: DynamoDB 항공편 테이블에서 기록을 삭제하여 보류 중인 항공편을 취소합니다.
- 렌터카 예약: 렌터카를 예약하기 위해 레코드를 CarRentals DynamoDB 테이블에 +와 함께 transaction_status 삽입합니다. pending
- 렌터카 확인: CarRentals DynamoDB 테이블의 레코드를 업데이트하여 로 **transaction_status** 설정하고 **confirmed** 렌터카를 확인합니다.
- 렌터카 예약 취소: CarRentals DynamoDB 테이블에서 레코드를 삭제하여 보류 중인 렌터카를 취소합니다.
- 결제 처리: 결제를 위해 DynamoDB 결제 테이블에 레코드를 삽입합니다.
- 결제 취소: DynamoDB 결제 테이블에서 결제에 대한 레코드를 삭제합니다.

Amazon SNS

샘플 애플리케이션은 SMS 메시지를 전송하고 고객에게 예약 성공 또는 실패를 알리기 위한 다음 주제 및 구독을 생성합니다. 샘플 애플리케이션을 테스트하는 동안 문자 메시지를 수신하려면 상태 시스템 정의 파일에 있는 유효한 전화 번호로 SMS 구독을 업데이트하세요.

AWS CDK 스니펫(다음 코드의 두 번째 줄에 전화번호 추가):

```

const topic = new sns.Topic(this, 'Topic');
topic.addSubscription(new subscriptions.SmsSubscription('+1111111111'));
const snsNotificationFailure = new tasks.SnsPublish(this, 'SendingSMSFailure', {
  topic:topic,
  integrationPattern: sfn.IntegrationPattern.REQUEST_RESPONSE,
  message: sfn.TaskInput.fromText('Your Travel Reservation Failed'),
});

const snsNotificationSuccess = new tasks.SnsPublish(this, 'SendingSMSSuccess', {
  topic:topic,
  integrationPattern: sfn.IntegrationPattern.REQUEST_RESPONSE,
  message: sfn.TaskInput.fromText('Your Travel Reservation is Successful'),
});

```

AWS SAM 스니펫(+1111111111 문자열을 유효한 전화번호로 대체):

```

StateMachineTopic1111111111:
  Type: 'AWS::SNS::Subscription'
  Properties:
    Protocol: sms
    TopicArn:
      Ref: StateMachineTopic
    Endpoint: '+1111111111'
  Metadata:
    'aws:sam:path': SamServerlessSagaStack/StateMachine/Topic/+1111111111/Resource

```

Terraform 스니펫(+1111111111 문자열을 유효한 전화번호로 대체):

```

resource "aws_sns_topic_subscription" "sms-target" {
  topic_arn = aws_sns_topic.topic.arn
  protocol  = "sms"
  endpoint  = "+1111111111"
}

```

성공적인 예약

다음 흐름은 “,”, “” ReserveFlight, “” 다음에 “ReserveCarRental” 및 “” ProcessPayment 가 붙은 성공적인 예약을 보여줍니다. ConfirmFlight ConfirmCarRental SNS 주제 구독자에게 전송되는 SMS 메시지를 통해 예약 성공 여부를 고객에게 알립니다.

예약 실패

이 흐름은 사가 패턴의 실패 사례입니다. 항공권 및 렌터카를 예약한 후 "ProcessPayment" 이 (가) 실패하면 단계가 역순으로 취소됩니다. 예약이 취소되고 SNS 주제 구독자에게 전송되는 SMS 메시지를 통해 고객에게 실패 사실을 알립니다.

AWS CDK로 Amazon ECS Anywhere를 설정하여 온프레미스 컨테이너 애플리케이션을 관리

작성자: Dr. Rahul Sharad Gaikwad(AWS)

코드 [amazon-ecs-anywhere-cdk](#) 리포지토리: -샘플

환경: PoC 또는 파일럿

기술: 현대화, 컨테이너 및 마이크로서비스, 하이브리드 클라우드 DevOps, 인프라

워크로드: 기타 모든 워크로드

AWS 서비스: AWS CDK, Amazon ECS, AWS Identity and Access Management

요약

[Amazon ECS Anywhere](#)는 Amazon Elastic Container Service(Amazon ECS)의 확장입니다. ECS Anywhere를 사용하여 온프레미스 또는 고객 관리형 환경에 네이티브 Amazon ECS 작업을 배포할 수 있습니다. 이 기능은 비용을 절감하고 복잡한 로컬 컨테이너 오케스트레이션 및 운영을 완화하는 데 도움이 됩니다. ECS Anywhere를 사용하여 온프레미스와 클라우드 환경 모두에서 컨테이너 애플리케이션을 배포하고 실행할 수 있습니다. 따라서 팀이 여러 도메인과 기술을 배우거나 복잡한 소프트웨어를 자체적으로 관리할 필요가 없습니다.

이 패턴은 AWS Cloud Development Kit([AWS CDK](#)) 스택을 사용하여 ECS Anywhere를 설정하는 단계를 보여줍니다.

사전 조건 및 제한 사항

사전 조건

- 활성 상태의 AWS 계정.
- AWS Command Line Interface(AWS CLI), 설치 및 구성됨. (AWS CLI 문서에서 [AWS CLI 설치, 업데이트, 제거](#) 참조)
- AWS CDK Toolkit, 설치 및 구성됨. (AWS CDK 설명서의 [AWS CDK 툴킷](#)을 참조하고 지침에 따라 최신 버전을 글로벌로 설치하세요.)
- AWS CDK용으로 설치 및 구성된 노드 패키지 관리자 (npm) TypeScript (npm 문서에서 [Node.js 및 npm 다운로드 및 설치](#) 참조)

제한 사항

- 제한 사항 및 고려 사항은 Amazon ECS 설명서의 [외부 인스턴스\(Amazon ECS Anywhere\)](#)를 참조하세요.

제품 버전

- AWS CDK Toolkit 버전 1.116.0 이상
- npm 버전 7.20.3 이상
- Node.js 버전 16.6.1 이상

아키텍처

대상 기술 스택

- AWS CloudFormation
- AWS CDK
- Amazon ECS Anywhere
- AWS Identity and Access Management(IAM)

대상 아키텍처

다음 다이어그램은 이 패턴으로 구현된 AWS CDK를 사용하여 ECS Anywhere를 설정하는 상위 수준 시스템 아키텍처를 보여줍니다. TypeScript

- AWS CDK 스택을 배포하면 AWS에 CloudFormation 스택이 생성됩니다.
- CloudFormation 스택은 Amazon ECS 클러스터 및 관련 AWS 리소스를 프로비저닝합니다.
- Amazon ECS 클러스터에 외부 인스턴스를 등록하려면 가상 머신(VM)에 AWS Systems Manager Agent(SSM Agent)를 설치하고 해당 VM을 AWS Systems Manager 관리형 인스턴스로 등록해야 합니다.
- 또한 VM을 Amazon ECS 클러스터에 외부 인스턴스로 등록하려면 VM에 Amazon ECS 컨테이너 에이전트와 Docker를 설치해야 합니다.
- Amazon ECS 클러스터에 외부 인스턴스를 등록하고 구성하면, 외부 인스턴스로 등록된 VM에서 여러 컨테이너를 실행할 수 있습니다.

자동화 및 규모 조정

이 패턴으로 제공되는 [GitHub 리포지토리](#)는 AWS CDK를 코드형 인프라 (IaC) 도구로 사용하여 이 아키텍처에 대한 구성을 생성합니다. AWS CDK를 사용하면 리소스를 오케스트레이션하고 ECS Anywhere를 설정할 수 있습니다.

도구

- [AWS Cloud Development Kit\(AWS CDK\)](#)는 AWS 클라우드 인프라를 코드로 정의하고 프로비저닝하는 데 도움이 되는 소프트웨어 개발 프레임워크입니다.
- [AWS Command Line Interface\(AWS CLI\)](#)는 명령줄 셸에서 명령을 사용하여 AWS 서비스와 상호 작용할 수 있는 오픈 소스 도구입니다.

코드

이 패턴의 소스 코드는 [Amazon ECS Anywhere CDK](#) 샘플 리포지토리에서 사용할 수 있습니다. GitHub 리포지토리를 복제하여 사용하려면 다음 섹션의 지침을 따르세요.

에픽

AWS CDK 구성 확인

작업	설명	필요한 기술
AWS CDK 버전 확인	<p>다음 명령을 실행하여 AWS CDK Toolkit 버전을 확인하세요.</p> <pre>cdk --version</pre> <p>이 패턴을 사용하려면 버전 1.116.0 이상이 필요합니다. 이전 버전의 AWS CDK를 사용 중이면 AWS CDK 문서에 있는 지침에 따라 업데이트하세요.</p>	DevOps 엔지니어

작업	설명	필요한 기술
AWS 보안 인증 정보를 설정합니다.	<p>보안 인증 정보를 설정하려면 <code>aws configure</code> 명령을 실행하고 프롬프트를 따릅니다.</p> <pre>\$aws configure AWS Access Key ID [None]: <your-access-key-ID> AWS Secret Access Key [None]: <your-secret-access-key> Default region name [None]: <your-Region-name> Default output format [None]:</pre>	DevOps 엔지니어

AWS CDK 환경 부트스트랩

작업	설명	필요한 기술
AWS CDK 코드 리포지토리를 복제합니다.	<p>다음 명령을 사용하여 이 패턴의 GitHub 코드 저장소를 복제합니다.</p> <pre>git clone https://github.com/aws-samples/amazon-ecs-anywhere-cdk-samples.git</pre>	DevOps 엔지니어
환경을 부트스트랩합니다.	<p>사용하려는 계정 및 AWS 지역에 AWS CloudFormation 템플릿을 배포하려면 다음 명령을 실행합니다.</p>	DevOps 엔지니어

작업	설명	필요한 기술
	<pre>cdk bootstrap <account-number>/<Region></pre> <p>자세한 내용은 AWS CDK 설명서의 부트스트래핑을 참조하세요.</p>	

프로젝트 구축 및 배포

작업	설명	필요한 기술
<p>패키지 종속 항목을 설치하고 파일을 컴파일합니다 TypeScript .</p>	<p>다음 명령을 실행하여 패키지 종속성을 설치하고 TypeScript 파일을 컴파일합니다.</p> <pre>\$cd amazon-ecs-anywhere-cdk-samples \$npm install \$npm fund</pre> <p>이 명령은 샘플 저장소의 모든 패키지를 설치합니다.</p> <p>중요: 누락된 패키지와 관련된 오류가 발생하면 다음 명령 중 하나를 사용하세요.</p> <pre>\$npm ci</pre> <p>—또는—</p> <pre>\$npm install -g @aws-cdk/<package_name></pre>	<p>DevOps 엔지니어</p>

작업	설명	필요한 기술
	<p>자세한 내용은 npm 설명서에 있는 npm ci와 npm install을 참조하세요.</p>	
프로젝트를 빌드합니다.	<p>다음 명령을 실행하여, 프로젝트를 구축합니다.</p> <pre>npm run build</pre> <p>프로젝트 구축 및 배포에 대한 자세한 내용은 AWS CDK 설명서의 첫 AWS CDK 앱을 참조하세요.</p>	DevOps 엔지니어
프로젝트를 배포합니다.	<p>프로젝트 코드를 배포하려면 다음 명령을 실행합니다.</p> <pre>cdk deploy</pre>	DevOps 엔지니어
스택 생성 및 출력을 확인합니다.	<p>https://console.aws.amazon.com/cloudformation 에서 AWS CloudFormation 콘솔을 열고 EcsAnywhereStack 스택을 선택합니다. Outputs 탭에는 외부 VM에서 실행할 명령이 표시됩니다.</p>	DevOps 엔지니어

온프레미스 머신 설정

작업	설명	필요한 기술
Vagrant를 사용하여 VM을 설정하세요.	<p>데모를 위해 HashiCorp Vagrant를 사용하여 VM을 만들 수 있습니다. Vagrant는 휴대용 가상 소프트웨어 개발 환</p>	DevOps 엔지니어

작업	설명	필요한 기술
	<p>경을 구축하고 유지 관리하기 위한 오픈 소스 유틸리티입니다. Vagrantfile이 있는 루트 디렉터리에서 <code>vagrant up</code> 명령을 실행하여 Vagrant VM을 생성합니다. 자세한 내용은 Vagrant 설명서를 참조하세요.</p>	

작업	설명	필요한 기술
<p>VM을 외부 인스턴스로 등록합니다.</p>	<p>1. <code>vagrant ssh</code> 명령을 사용하여 Vagrant VM에 로그인합니다. 자세한 내용은 Vagrant 설명서를 참조하세요.</p> <p>2. AWS Systems Manager에 VM을 등록하고 외부 인스턴스를 활성화하는 데 사용할 수 있는 활성화 코드와 ID를 생성합니다. 이 명령의 출력에는 다음과 같이 <code>ActivationId</code> 및 <code>ActivationCode</code> 값이 포함됩니다.</p> <pre>aws ssm create-activation --iam-role EcsAnywhereInstanceRole tee ssm-activation.json</pre> <p>3. 활성화 ID 및 코드 값 보내기:</p> <pre>export ACTIVATION_ID=<activation-ID> export ACTIVATION_CODE=<activation-code></pre> <p>4. 온프레미스 서버 또는 VM에서 설치 스크립트를 다운로드합니다.</p> <pre>curl -o "ecs-anywhere-install.sh" "https://amazon-ecs-agent.s3.amazonaws.com/ec</pre>	<p>DevOps 엔지니어</p>

작업	설명	필요한 기술
	<pre>s-anywhere-install -latest.sh" && sudo chmod +x ecs-anywhere- install.sh</pre> <p>5. 온프레미스 서버 또는 VM에 서 설치 스크립트를 실행합니 다.</p> <pre>sudo ./ecs-anywhere-ins tall.sh \ --cluster test-ecs- anywhere \ --activation-id \$ACTIVATION_ID \ --activation-code \$ACTIVATION_CODE \ --region <Region></pre> <p>VM 설정 및 등록에 대한 자세 한 내용은 Amazon ECS 설명 서의 클러스터에 외부 인스턴 스 등록을 참조하세요.</p>	
<p>ECS Anywhere와 외부 VM의 상태를 확인합니다.</p>	<p>가상 박스가 Amazon ECS 컨 트를 플레인에 연결되어 실행 중인지 확인하려면 다음 명령 을 사용하세요.</p> <pre>aws ssm describe- instance-information aws ecs list-container- instances --cluster \$CLUSTER_NAME</pre>	<p>DevOps 엔지니어</p>

정리

작업	설명	필요한 기술
리소스를 정리하고 삭제합니다.	<p>이 패턴을 수행한 다음에는 추가 요금이 발생하지 않도록 생성한 리소스를 제거해야 합니다. 정리하려면 다음 명령을 실행합니다.</p> <pre>cdk destroy</pre>	DevOps 엔지니어

관련 리소스

- [Amazon ECS Anywhere 설명서](#)
- [Amazon ECS Anywhere 데모](#)
- [Amazon ECS Anywhere 워크숍 샘플](#)

AWS에서 ASP.NET Web Forms 애플리케이션 현대화

작성자: Vijai Anand Ramalingam(AWS) 및 Sreelaxmi Pai(AWS)

환경: PoC 또는 파일럿

기술: 현대화, 컨테이너 및 마이크로서비스, 소프트웨어 개발 및 테스트, 웹 및 모바일 앱

워크로드: Microsoft

AWS 서비스: 아마존 CloudWatch, 아마존 ECS, AWS Systems Manager

요약

이 패턴은 레거시 모놀리스 ASP.NET 웹 양식 애플리케이션을 AWS의 ASP.NET Core로 이식하여 현대화하는 단계를 설명합니다.

ASP.NET Web Forms 애플리케이션을 ASP.NET Core로 포팅하면 Linux의 성능, 비용 절감 및 강력한 에코시스템을 활용할 수 있습니다. 하지만 이 작업에는 상당한 수작업이 필요할 수 있습니다. 이 패턴에서는 단계적 접근 방식을 사용하여 레거시 애플리케이션을 점진적으로 현대화한 다음 AWS 클라우드에서 컨테이너화합니다.

쇼핑 카트에 사용할 레거시 모놀리스 애플리케이션을 생각해 보세요. ASP.NET Web Forms 애플리케이션으로 만들어졌으며 코드 숨김(asp.x.cs) 파일이 있는 .aspx 페이지로 구성되어 있다고 가정해 보겠습니다. 현대화 프로세스는 세 단계로 구성됩니다.

1. 적절한 분해 패턴을 사용하여 모놀리스를 마이크로서비스로 분리하세요. 자세한 내용은 AWS 권장 가이드 웹 사이트에서 [모놀리스를 마이크로서비스로 분해](#) 가이드를 참조하세요.
2. 레거시 ASP.NET Web Forms(.NET 프레임워크) 애플리케이션을 .NET 5 이상의 ASP.NET Core로 포팅합니다. 이 패턴에서는 Porting Assistant for .NET를 사용하여 ASP.NET Web Forms 애플리케이션을 스캔하고 ASP.NET Core와 호환되지 않는지 식별합니다. 이렇게 하면 수동 포팅 작업이 줄어듭니다.
3. React를 사용하여 Web Forms UI 레이어를 재개발하세요. 이 패턴에는 UI 재개발이 포함되지 않습니다. 자세한 지침은 React 설명서에서 [새 React 앱 만들기](#)를 참조하세요.
4. Web Forms 코드 숨김 파일(비즈니스 인터페이스)을 ASP.NET Core 웹 API로 재개발하세요. 이 패턴은 NDepend 보고서를 사용하여 필수 파일 및 종속성을 파악하는 데 도움이 됩니다.

5. Porting Assistant for .NET를 사용하여 기존 애플리케이션의 공유/공통 프로젝트(예: 비즈니스 로직 및 데이터 액세스)를 .NET 5 이상으로 업그레이드하세요.
6. 애플리케이션을 보완하기 위해 AWS 서비스를 추가하세요. 예를 들어 [Amazon Logs를 사용하여 애플리케이션 CloudWatch 로그를](#) 모니터링, 저장 및 액세스하고, [AWS Systems Manager를](#) 사용하여 애플리케이션 설정을 저장할 수 있습니다.
7. 현대화된 ASP.NET Core 애플리케이션을 컨테이너화합니다. 이 패턴은 Visual Studio에서 Linux를 대상으로 하는 Docker 파일을 만들고, 이를 로컬에서 테스트하기 위해 Docker Desktop을 사용합니다. 이 단계에서는 레거시 애플리케이션이 이미 온프레미스 또는 Amazon Elastic Compute Cloud(Amazon EC2) Windows 인스턴스에서 실행 중이라고 가정합니다. 자세한 내용은 [Amazon EC2 Linux 인스턴스에서 ASP.NET Core 웹 API Docker 컨테이너 실행](#) 패턴을 참조하세요.
8. 현대화된 ASP.NET Core 애플리케이션을 Amazon Elastic Container Service(Amazon ECS)에 배포합니다. 이 패턴은 배포 단계를 다루지 않습니다. 지침은 [Amazon ECS 워크숍](#)을 참조하세요.

참고: 이 패턴에는 UI 개발, 데이터베이스 현대화 또는 컨테이너 배포 단계가 포함되지 않습니다.

사전 조건 및 제한 사항

사전 조건

- [Visual Studio](#) 또는 [Visual Studio Code](#), 다운로드 및 설치됨.
- AWS Management Console 및 AWS Command Line Interface(AWS CLI) 버전 2를 사용하여 AWS 계정에 액세스합니다. ([AWS CLI 구성 지침](#)을 참조하세요.)
- AWS Toolkit for Visual Studio([설정 지침](#)을 참조).
- Docker Desktop, [다운로드](#) 및 설치됨.
- .NET SDK, [다운로드](#) 및 설치됨.
- NDepend 도구, [다운로드](#) 및 설치됨. Visual Studio용 NDepend 확장 프로그램을 설치하려면 NDepend.VisualStudioExtension.Installer를 실행합니다([지침 참조](#)). 필요에 따라 Visual Studio 2019 또는 2022를 선택할 수 있습니다.
- Porting Assistant for .NET, [다운로드](#) 및 설치됨.

아키텍처

쇼핑 카트 애플리케이션 현대화

다음 다이어그램은 기존 ASP.NET 쇼핑 카트 애플리케이션의 현대화 프로세스를 보여줍니다.

대상 아키텍처

다음 다이어그램은 AWS에서 현대화된 쇼핑 카트 애플리케이션의 아키텍처를 보여 줍니다. ASP.NET Core 웹 API는 Amazon ECS 클러스터에 배포됩니다. 로깅 및 구성 서비스는 Amazon CloudWatch Logs 및 AWS Systems Manager에서 제공합니다.

도구

서비스

- [Amazon ECS](#) – Amazon Elastic Container Service(Amazon ECS)는 클러스터에서 컨테이너를 실행, 중지 및 관리하기 위한 컨테이너 관리 서비스로서 확장성과 속도가 뛰어납니다. AWS Fargate에서 관리하는 서버리스 인프라에서 작업 및 서비스를 실행할 수 있습니다. 또는 인프라에 대한 더 세부적인 제어를 위해 관리하는 EC2 인스턴스의 클러스터에서 작업과 서비스를 실행할 수 있습니다.
- [Amazon CloudWatch Logs](#) — Amazon Logs는 사용하는 모든 시스템, 애플리케이션 및 AWS 서비스의 CloudWatch 로그를 중앙 집중화합니다. 그런 다음 로그를 보고, 특정 오류 코드 또는 패턴이 있는지 검색하고, 특정 필드를 기반으로 필터링하거나, 향후 분석을 위해 안전하게 보관할 수 있습니다.
- [AWS Systems Manager](#) – AWS Systems Manager는 AWS에서 인프라를 보고 제어하기 위해 사용할 수 있는 AWS 서비스입니다. Systems Manager 콘솔을 사용하여 여러 AWS 서비스의 운영 데이터를 보고 AWS 리소스에서 운영 작업을 자동화할 수 있습니다. Systems Manager는 관리형 인스턴스를 검사하고 탐지된 정책 위반을 보고(또는 시정 조치)함으로써 보안 및 규정 준수를 유지하는 데 도움이 됩니다.

도구

- [Visual Studio](#) 또는 [Visual Studio Code](#) – .NET 애플리케이션, 웹 API 및 기타 프로그램을 구축하기 위한 도구.
- [AWS Toolkit for Visual Studio](#) – AWS 서비스를 사용하는 .NET 애플리케이션을 개발, 디버깅 및 배포하는 데 도움이 되는 Visual Studio용 확장 프로그램.
- [Docker Desktop](#) – 컨테이너식 애플리케이션의 구축 및 배포를 간소화하는 도구.
- [NDepend](#) – .NET 코드를 모니터링하여 종속성, 품질 문제 및 코드 변경을 모니터링하는 분석기.
- [Porting Assistant for .NET](#) – .NET 코드를 스캔하여 .NET Core와의 비호환성을 식별하고 마이그레이션 노력을 추정하는 분석 도구.

에픽

레거시 애플리케이션을 .NET 5 이상 버전으로 포팅

작업	설명	필요한 기술
<p>.NET 프레임워크 레거시 애플리케이션을 .NET 5로 업그레이드하세요.</p>	<p>Porting Assistant for .NET를 사용하여 기존 ASP.NET Web Forms 애플리케이션을 .NET 5 이상으로 변환할 수 있습니다. Porting Assistant for .NET의 지침을 따르세요.</p>	<p>앱 개발자</p>
<p>NDepend 보고서를 생성하세요.</p>	<p>ASP.NET Web Forms 애플리케이션을 마이크로서비스로 분해하여 현대화하는 경우, 기존 애플리케이션의 일부 .cs 파일이 필요하지 않을 수 있습니다. NDepend를 사용하여 모든 코드 숨김(.cs) 파일에 대한 보고서를 생성하여 모든 호출자와 수신자를 가져올 수 있습니다. 이 보고서를 사용하면 마이크로서비스에서 필요한 파일만 식별하여 사용할 수 있습니다.</p> <p>NDepend를 설치한 후(필수 구성 요소 섹션 참조) Visual Studio에서 기존 애플리케이션의 솔루션(.sln 파일)을 열고 다음 단계를 따르세요.</p> <ol style="list-style-type: none"> 1. Visual Studio에서 레거시 애플리케이션을 구축하세요. 2. Visual Studio 메뉴 표시줄에서 NDepend, 새 NDepend 프로젝트를 현재 VS 솔루션에 연결을 선택합니다. 	<p>앱 개발자</p>

작업	설명	필요한 기술
	<ol style="list-style-type: none"> 3. .NET 어셈블리 분석을 선택합니다. 4. 분석이 완료되면 Solution Explorer의 해당 프로젝트로 이동합니다. 보고서를 생성하려는 코드 숨김 파일(예: listproducts.aspx.cs)을 마우스 오른쪽 단추로 클릭한 다음 종속성 그래프에 표시를 선택합니다. 5. 탐색 막대에서 발신자 및 수신자를 선택한 다음 코드 쿼리 편집을 선택합니다. 6. 쿼리 및 규칙 편집 창에서 다운로드 화살표를 선택한 다음 Excel로 내보내기를 선택합니다. <p>이 프로세스는 모든 발신자와 수신자를 나열하는 코드 숨김 파일에 대한 보고서를 생성합니다. 종속성 그래프에 대한 자세한 내용은 NDepend 설명서를 참조하세요.</p>	

작업	설명	필요한 기술
<p>새 .NET 5 솔루션을 생성합니다.</p>	<p>현대화된 ASP.NET Core 웹 API를 위한 새 .NET 5(또는 그 이상) 구조를 만들려면 다음을 따르세요.</p> <ol style="list-style-type: none"> 1. Visual Studio를 엽니다. 2. 비어 있는 새 솔루션을 만드세요. 3. 기존 애플리케이션을 기반으로 .NET 5(또는 그 이상)를 대상으로 하는 새 프로젝트를 만드세요. 장바구니 애플리케이션을 위한 기존 프로젝트와 새 프로젝트의 예제는 추가 정보 섹션을 참조하세요. 4. 이전 단계의 NDepend 보고서를 사용하여 모든 필수 파일을 파악하세요. 이전에 업그레이드한 애플리케이션에서 이러한 파일을 복사하여 새 솔루션에 추가합니다. 5. 솔루션을 구축하고 모든 문제를 해결하세요. <p>프로젝트 및 솔루션 만들기에 대한 자세한 내용은 Visual Studio 설명서를 참조하세요.</p> <p>참고 솔루션을 구축하고 기능을 확인하면서 NDepend가 식별한 파일 외에도 솔루션에 추가할 몇 가지 추가 파일을 식별할 수 있습니다.</p>	<p>앱 개발자</p>

애플리케이션 코드를 업데이트

작업	설명	필요한 기술
<p>ASP.NET Core를 사용하여 웹 API 구현.</p>	<p>기존 모놀리스 쇼핑 카트 애플리케이션에서 식별한 마이크로서비스 중 하나가 제품이라고 가정해 보겠습니다. 이전에 에픽에서 새 제품을 위한 새 ASP.NET Core 웹 API 프로젝트를 만들었습니다. 이 단계에서는 제품과 관련된 모든 웹 양식(.aspx 페이지)을 식별하고 현대화합니다. 앞서 아키텍처 섹션에서 설명한 것처럼 제품이 네 개의 웹 양식으로 구성되어 있다고 가정해 보겠습니다.</p> <ul style="list-style-type: none"> • 제품 목록 • 제품 보기 • 제품 추가/편집 • 제품 삭제 <p>각 웹 양식을 분석하고, 데이터베이스로 전송된 모든 요청을 식별하여 로직을 수행하고, 응답을 받아야 합니다. 각 요청을 웹 API 엔드포인트로 구현할 수 있습니다. 웹 양식이 주어진다면 제품에는 다음과 같은 엔드포인트가 있을 수 있습니다.</p> <ul style="list-style-type: none"> • /api/products • /api/products/{id} • /api/products/add 	<p>앱 개발자</p>

작업	설명	필요한 기술
	<ul style="list-style-type: none"> • /api/products/update/{id} • /api/products/delete/{id} <p>앞서 언급한 것처럼 비즈니스 로직, 데이터 액세스, 공유/공통 프로젝트를 포함하여 .NET 5로 업그레이드한 다른 모든 프로젝트도 재사용할 수 있습니다.</p>	
<p>Amazon CloudWatch 로그를 구성합니다.</p>	<p>Amazon Logs를 사용하여 애플리케이션 CloudWatch 로그를 모니터링, 저장 및 액세스할 수 있습니다. AWS SDK를 사용하여 Amazon CloudWatch Logs에 데이터를 기록할 수 있습니다. 또한 NLog, Log4Net 및 ASP.NET Core 로깅 프레임워크와 CloudWatch 같은 인기 있는 .NET 로깅 프레임워크를 사용하여 .NET 애플리케이션을 로그와 통합할 수 있습니다.</p> <p>이 단계에 대한 자세한 내용은 블로그 게시물 Amazon CloudWatch Logs 및 .NET 로깅 프레임워크를 참조하십시오.</p>	<p>앱 개발자</p>

작업	설명	필요한 기술
<p>AWS Systems Manager Parameter Store를 구성합니다.</p>	<p>AWS Systems Manager Parameter Store를 사용하여 연결 문자열과 같은 애플리케이션 설정을 애플리케이션 코드와는 별도로 저장할 수 있습니다. Amazon. NuGet Extensions.Configuration 패키지. SystemsManager 애플리케이션이 이러한 설정을 AWS Systems Manager Parameter Store에서 .NET Core 구성 시스템으로 로드하는 방법을 간소화합니다.</p> <p>이 단계에 대한 자세한 내용은 블로그 게시물 AWS Systems Manager용 .NET Core 구성 공 급자를 참조하세요.</p>	<p>앱 개발자</p>

인증 및 권한 부여를 추가

작업	설명	필요한 기술
<p>인증을 위해 공유 쿠키를 사용하세요.</p>	<p>레거시 모놀리스 애플리케이션을 현대화하는 것은 반복적인 프로세스이므로 모놀리스와 현대화된 버전이 공존해야 합니다. 공유 쿠키를 사용하여 두 버전 간에 원활한 인증을 수행할 수 있습니다. 현대화된 ASP.NET Core 애플리케이션이 쿠키의 유효성을 검사하는 동안 기존 ASP.NET 애플리케이션은 계속해서 사용자 보안</p>	<p>앱 개발자</p>

작업	설명	필요한 기술
	<p>인증 정보를 확인하고 쿠키를 발행합니다.</p> <p>지침과 샘플 코드는 샘플 프로젝트를 참조하십시오. GitHub</p>	

로컬에서 컨테이너 구축 및 실행

작업	설명	필요한 기술
Visual Studio를 사용하여 도커 이미지를 생성합니다.	<p>이 단계에서는 .NET Core 웹 API용 Visual Studio를 사용하여 Docker 파일을 만듭니다.</p> <ol style="list-style-type: none"> 1. Visual Studio를 엽니다. 2. Solution Explorer의 프로젝트 컨텍스트 메뉴(마우스 오른쪽 버튼 클릭)에서 추가, Docker Support를 선택합니다. 3. 대상 운영 체제로 Linux를 선택합니다. <p>Visual Studio가 프로젝트에 사용할 Docker 파일을 만듭니다. 샘플 Docker 파일은 Microsoft 웹 사이트의 Docker용 Visual Studio 컨테이너 도구를 참조하세요.</p>	앱 개발자
Docker Desktop을 사용하여 컨테이너를 구축하고 실행합니다.	이제 Docker Desktop에서 컨테이너를 구축, 생성 및 실행할 수 있습니다.	앱 개발자

작업	설명	필요한 기술
	<p>1. 명령 프롬프트 창을 엽니다. Docker 파일이 있는 솔루션 폴더로 이동합니다. 도커 이미지를 생성하려면 다음 명령을 실행합니다.</p> <pre data-bbox="634 474 1027 632">docker build -t aspnetcorewebapiimage -f Dockerfile .</pre> <p>2. 모든 도커 이미지를 보려면 다음 명령을 실행합니다.</p> <pre data-bbox="634 768 1027 846">docker images</pre> <p>3. 컨테이너를 생성하고 실행하려면 다음 명령을 실행합니다.</p> <pre data-bbox="634 1031 1027 1266">docker run -d -p 8080:80 --name aspnetcorewebapicontainer aspnetcorewebapiimage</pre> <p>4. Docker Desktop을 열고 컨테이너/앱을 선택합니다. <code>aspnetcorewebapicontainer running</code>이라는 새 컨테이너를 볼 수 있습니다.</p>	

관련 리소스

- [Amazon EC2 Linux 인스턴스에서 ASP.NET Core 웹 API Docker 컨테이너 실행\(AWS 권장 가이드\)](#)
- [Amazon ECS 워크숍](#)

- [AWS를 사용하여 ECS 블루/그린 배포 수행 \(CodeDeploy AWS CloudFormation 설명서\)](#)
CloudFormation
- [NDepend 시작하기\(NDepend 설명서\)](#)
- [Porting Assistant for .NET](#)

추가 정보

다음 테이블에는 기존 쇼핑 카트 애플리케이션의 샘플 프로젝트 예제와 현대화된 ASP.NET Core 애플리케이션의 해당 프로젝트가 나와 있습니다.

레거시 솔루션:

프로젝트 이름	프로젝트 템플릿	대상 프레임워크
비즈니스 인터페이스	클래스 라이브러리	.NET Framework
BusinessLogic	클래스 라이브러리	.NET Framework
WebApplication	ASP.NET 프레임워크 웹 애플리케이션	.NET Framework
UnitTests	NUnit 테스트 프로젝트	.NET Framework
공유 ->공통	클래스 라이브러리	.NET Framework
공유 ->프레임워크	클래스 라이브러리	.NET Framework

새 솔루션:

프로젝트 이름	프로젝트 템플릿	대상 프레임워크
BusinessLogic	클래스 라이브러리	.NET 5.0
<WebAPI>	ASP.NET 코어 웹 API	.NET 5.0
<WebAPI>. UnitTests	NUnit 3 테스트 프로젝트	.NET 5.0
공유 ->공통	클래스 라이브러리	.NET 5.0

공유 ->프레임워크

클래스 라이브러리

.NET 5.0

AWS Fargate를 사용하여 이벤트 기반 및 예약된 워크로드를 대규모로 실행

작성자: HARI OHM PRASATH RAJAGOPAL(AWS)

환경: PoC 또는 파일럿

기술: 현대화, 서버리스, 운영

워크로드: 오픈 소스

AWS 서비스: 아마존 EC2 컨테이너 레지스트리, 아마존 ECS, AWS, AWS Fargate, CodeCommit AWS Lambda, 아마존 SNS

요약

이 패턴은 AWS Fargate를 사용하여 Amazon Web Services(AWS) 클라우드에서 스케줄링된 워크로드와 이벤트 기반 워크로드를 대규모로 실행하는 방법을 설명합니다.

이 패턴이 설정한 사용 사례에서는 풀 요청이 제출될 때마다 AWS 계정 번호 및 보안 인증과 같은 AWS 민감 정보가 있는지 코드를 스캔합니다. 풀 요청은 Lambda 함수를 시작합니다. Lambda 함수는 코드 스캔을 처리하는 Fargate 작업을 간접 호출합니다. 새 풀 리퀘스트가 발생할 때마다 Lambda가 시작됩니다. 스캔 결과 민감한 정보가 발견되면 Amazon Simple Notification Service(SNS)에서 스캔 결과를 이메일 메시지로 전송합니다.

이 패턴은 다음과 같은 경우에 유용합니다.

- 기업에서 런타임(15분 제한) 또는 메모리 제한으로 인해 AWS Lambda에서 실행할 수 없는 많은 예약 및 이벤트 기반 워크로드를 실행해야 하는 경우
- AWS에서 이러한 워크로드에 프로비저닝된 인스턴스를 관리하도록 하려는 경우

이 패턴을 사용할 때는 Virtual Private Cloud(VPC)를 새로 생성할 수 있습니다.

사전 조건 및 제한 사항

사전 조건

- 활성 상태의 AWS 계정.

- 코드 베이스 호스팅 및 풀 리퀘스트 생성을 CodeCommit 위한 AWS
- AWS Command Line Interface(AWS CLI) 버전 1.7 이상, macOS, Linux 또는 Windows에 설치 및 구성
- 컨테이너에서 실행되는 워크로드
- 클래스 경로에 설정된 Apache Maven 실행 파일

아키텍처

전체 흐름에는 다음 단계가 포함됩니다.

1. 새 풀 리퀘스트가 제출될 때마다 Lambda 함수가 시작됩니다. CodeCommit Lambda 함수는 Amazon을 통해 이벤트를 수신합니다. CodeCommit Pull Request State Change EventBridge
2. Lambda 함수는 코드를 체크아웃하고 스캔하기 위한 다음 환경 파라미터와 함께 새로운 Fargate 작업을 제출합니다.

```
RUNNER # <<TaskARN>>
SNS_TOPIC # <<SNSTopicARN>>
SUBNET # <<Subnet in which Fargate task gets launched>>
```

스캔 결과 코드에서 민감한 정보가 발견되면 Fargate는 Amazon SNS 주제로 새 메시지를 푸시합니다.

3. SNS 구독자는 주제의 메시지를 읽고 이메일 메시지를 보냅니다.

기술

- AWS CodeCommit
- Amazon Elastic Container Registry(Amazon ECR)
- Amazon Elastic Container Service(Amazon ECS)
- 아마존 EventBridge
- AWS Fargate
- AWS Lambda
- Amazon SNS

- Docker

도구

도구

- [AWS CLI](#) - AWS Command Line Interface(CLI)는 AWS 서비스를 관리하는 통합 도구입니다.
- [AWS CodeCommit](#) — CodeCommit AWS는 안전한 Git 기반 리포지토리를 호스팅하는 완전 관리형 소스 제어 서비스입니다. 이를 사용하여 CodeCommit 팀은 안전하고 확장성이 뛰어난 환경에서 코드 작업을 공동으로 수행할 수 있습니다.
- [Amazon ECR](#) - Amazon Elastic Container Registry(Amazon ECR)는 Docker 컨테이너 이미지를 저장, 관리 및 배포하는 데 사용할 수 있는 완전 관리형 레지스트리입니다.
- [Amazon ECS](#) - Amazon Elastic Container Service(Amazon ECS)는 확장성이 뛰어나고 빠른 컨테이너 관리 서비스입니다. Amazon ECS를 사용하여 클러스터에서 컨테이너를 실행, 중지 및 관리할 수 있습니다.
- [AWS Fargate](#) - AWS Fargate는 Amazon EC2 인스턴스의 서버나 클러스터를 관리할 필요 없이 컨테이너를 실행하기 위해 Amazon ECS에 사용할 수 있는 기술입니다.
- [AWS Lambda](#) - AWS Lambda는 서버를 프로비저닝하거나 관리하지 않고도 코드를 실행하도록 지원하는 컴퓨팅 서비스입니다. Lambda는 필요 시에만 코드를 실행하며, 일일 몇 개의 요청에서 초당 수천 개의 요청까지 자동으로 규모를 조정합니다.
- [Amazon SNS](#) - Amazon Simple Notification Service(SNS)는 게시자에서 구독자(생산자 및 소비자라고도 함)로 메시지를 전송하는 관리형 서비스입니다. 게시자는 논리적 액세스 지점 및 커뮤니케이션 채널인 주제에 메시지를 전송하여 구독자와 비동기식으로 통신합니다. SNS 주제를 구독하는 클라이언트는 Lambda, 이메일, 모바일 푸시 알림 및 모바일 문자 메시지(SMS)와 같이 지원되는 프로토콜을 사용하여 게시된 메시지를 수신합니다.
- [Docker](#) - Docker는 컨테이너라는 패키지로 애플리케이션을 빌드, 테스트 및 제공할 수 있도록 도와줍니다.
- [Git 클라이언트](#) - 필요한 아티팩트를 체크아웃하기 위한 명령줄 또는 데스크톱 도구
- [Maven](#) - Apache Maven은 프로젝트의 구축, 보고 및 문서를 중앙에서 관리하기 위한 프로젝트 관리 도구입니다.

에픽

로컬 리포지토리 설정

작업	설명	필요한 기술
코드를 다운로드하십시오.	첨부 섹션에서 .zip 파일을 다운로드하고 파일을 추출합니다.	개발자, AWS 시스템 관리자
리포지토리를 설정합니다.	루트 폴더에서 <code>mvn clean install</code> 을 실행합니다.	개발자, AWS 시스템 관리자

Amazon ECR 이미지를 생성하고 이미지를 푸시합니다.

작업	설명	필요한 기술
Amazon ECR 리포지토리를 생성하고 로그인합니다.	Amazon ECR 콘솔을 엽니다. 탐색 창에서 리포지토리를 선택하고 리포지토리 생성을 선택합니다. 이 이야기와 다른 이야기에 대한 도움이 필요하면 관련 리소스 섹션을 참조하십시오.	개발자, AWS 시스템 관리자
컨테이너 이미지를 푸시하십시오.	리포지토리를 열고 푸시 명령 보기를 선택한 다음 Docker에 로그인합니다. 로그인한 후 추가 정보 섹션의 컨테이너 이미지 푸시에 있는 명령을 필수 대체 항목과 함께 실행합니다. 그러면 코드 스캔을 수행하는 데 사용되는 Docker 컨테이너 이미지가 업로드됩니다. 업로드가 완료되면 Amazon ECR 리포지토리에 있는 최신 빌드의 URL을 복사합니다.	개발자, AWS 시스템 관리자

CodeCommit 리포지토리 만들기

작업	설명	필요한 기술
CodeCommit 리포지토리를 생성합니다.	새 AWS CodeCommit 리포지토리를 생성하려면 추가 정보 섹션의 CodeCommit 리포지토리 생성에서 명령을 실행합니다.	개발자, AWS 시스템 관리자

VPC 생성(선택 사항)

작업	설명	필요한 기술
VPC를 생성합니다.	기존 VPC 대신 새 VPC를 사용하려면 추가 정보 섹션의 VPC 생성에 있는 명령을 실행합니다. AWS Cloud Development Kit(AWS CDK) 스크립트는 생성된 VPC와 서브넷의 ID를 출력합니다.	개발자, AWS 시스템 관리자

Amazon ECS 클러스터 및 Fargate 작업 생성

작업	설명	필요한 기술
클러스터와 작업을 생성합니다.	Amazon ECS 클러스터와 Fargate 작업 정의를 생성하려면 추가 정보 섹션의 클러스터 및 작업 생성에 있는 명령을 실행합니다. 셸 스크립트를 실행하는 동안 올바른 VPC ID와 Amazon ECR 리포지토리 URI가 파라미터로 전달되었는지 확인하십시오. 스크립트는 도	개발자, AWS 시스템 관리자

작업	설명	필요한 기술
	커 이미지(스캔 담당)를 가리키는 Fargate 작업 정의를 생성합니다. 그러면 스크립트가 작업 및 관련 실행 역할을 생성합니다.	
Amazon ECS 클러스터를 확인합니다.	Amazon ECS 콘솔을 엽니다. 탐색 창에서 클러스터를 선택하고 Fargate-Job-Cluster라는 새로 생성된 Amazon ECS 클러스터를 선택합니다. 그런 다음 탐색 창에서 작업 정의를 선택하고 접두어 awscdkfargateecsTaskDef 가 포함된 새 작업 정의가 있는지 확인합니다.	개발자, AWS 시스템 관리자

SNS 주제 및 구독자 생성

작업	설명	필요한 기술
SNS 주제를 생성합니다.	SNS 주제를 만들려면 추가 정보 섹션의 SNS 주제 만들기에 있는 명령을 실행합니다. 생성이 성공하면 다음 단계에서 사용되는 SNS ARN을 기록해 둡니다.	개발자, AWS 시스템 관리자
SNS 구독자를 생성하십시오.	SNS 주제에 대한 이메일 구독자를 만들려면 추가 정보 섹션의 SNS 구독자 만들기에 있는 명령을 실행합니다. 반드시 CLI 명령에서 TopicARN과 Email address를 교체하여 사용하십시오. 이메일 알림을 받으려	개발자, AWS 시스템 관리자

작업	설명	필요한 기술
	면 구독자로 사용되는 이메일 주소를 확인해야 합니다.	

Lambda 함수 및 트리거 생성 CodeCommit

작업	설명	필요한 기술
함수와 트리거를 생성합니다.	트리거가 있는 Lambda 함수를 생성하려면 Lambda 함수에서 명령을 실행하고 추가 정보 섹션에서 트리거하십시오. CodeCommit CodeCommit 명령을 실행하기 전에 파라미터를 해당 값으로 바꿔야 합니다. 스크립트는 Lambda 함수를 생성하고 새 풀 요청이 생성될 때 간접 호출되도록 구성합니다.	개발자, AWS 시스템 관리자

애플리케이션 테스트

작업	설명	필요한 기술
애플리케이션을 테스트합니다.	AWS의 민감한 정보를 CodeCommit 리포지토리에 체크인하는 경우 Lambda 함수가 시작되어야 합니다. Lambda 함수는 코드를 스캔하고 스캔 결과를 이메일 알림으로 보내는 Fargate 작업을 시작합니다.	개발자, AWS 시스템 관리자

관련 리소스

- [Amazon ECT 리포지토리 생성](#)

- [도커 이미지를 Amazon ECR로 푸시](#)

추가 정보

컨테이너 이미지 푸시

```
> cd 1-ecr-image-push
> ./run.sh <<ecr-repository>>
```

리포지토리 생성 CodeCommit

```
aws codecommit create-repository --repository-name test-repo --repository-description
"My Test repository"
```

VPC 생성

```
> cd 2-create-vpc
> ./run.sh
```

출력

```
aws-batch-cdk-vpc-efs-launch-template.privatesubnet = subnet-<<id>>
aws-batch-cdk-vpc-efs-launch-template.publicsubnet = subnet-<<id>>
aws-batch-cdk-vpc-efs-launch-template.vpcid = vpc-<<id>>
```

클러스터 및 작업 생성

```
> export CDK_DEFAULT_ACCOUNT = <<aws_account_id>>
> export CDK_DEFAULT_REGION = <<aws_region>>
> cd 3-create-ecs-task
> ./run.sh <<vpc-id>> <<ecr-repo-uri>>
```

출력

```
aws-cdk-fargate-ecs.CLUSTERNAME = Fargate-Job-Cluster
aws-cdk-fargate-ecs.ClusterARN = <<cluster_arn>>
aws-cdk-fargate-ecs.ContainerARN = Fargate-Container
aws-cdk-fargate-ecs.TaskARN = <<task_arn>>
```

```
aws-cdk-fargate-ecs.TaskExecutionRole = <<execution_role_arn>>
aws-cdk-fargate-ecs.TaskRole = <<task_role_arn>>
```

SNS 주제 생성

```
aws sns create-topic --name code-commit-topic
```

SNS 구독자 생성

```
aws sns subscribe \
  --topic-arn <<topic_arn>> \
  --protocol email \
  --notification-endpoint <<email_address>>
```

Lambda 함수 및 트리거 CodeCommit

```
> export CDK_DEFAULT_ACCOUNT = <<aws_account_id>>
> export CDK_DEFAULT_REGION = <<aws_region>>
> cd 5-Lambda-CodeCommit-Trigger
> ./run.sh <<taskarn>> <<snstopicarn>> subnet-<<id>> <<codecommitarn>>
```

출력

```
aws-cdk-fargate-lambda-event.Cloudwatchrule = <<cloudwatchrule>>
aws-cdk-fargate-lambda-event.CodeCommitLambda = AWS-Code-Scanner-Function
aws-cdk-fargate-lambda-event.LambdaRole = <<lambdaiamrole>>
```

첨부

이 문서와 관련된 추가 콘텐츠에 액세스하려면 [attachment.zip](#) 파일의 압축을 풉니다.

C# 및 AWS CDK를 사용한 사일로 모델을 위한 SaaS 아키텍처의 테넌트 온보딩

작성자: Tabby Ward(AWS), Susmitha Reddy Gankidi(AWS), Vijai Anand Ramalingam(AWS)

코드 리포지토리: 테넌트 온보딩 사일로	환경: PoC 또는 파일럿	기술: 현대화, 클라우드 네이티브, SaaS, DevOps
워크로드: 오픈 소스	AWS 서비스: AWS, 아마존 다이나모DB CloudFormation, 아마존 다이나모DB 스트림, AWS Lambda, 아마존 API Gateway	

요약

서비스형 소프트웨어(SaaS) 애플리케이션은 다양한 아키텍처 모델을 사용하여 구축할 수 있습니다. 사일로 모델은 테넌트에게 전용 리소스가 제공되는 아키텍처를 말합니다.

SaaS 애플리케이션은 마찰 없는 모델을 사용하여 환경에 새로운 테넌트를 도입합니다. 이를 위해서는 새 테넌트를 생성하는 데 필요한 모든 요소를 성공적으로 프로비저닝하고 구성하기 위해 여러 구성 요소를 오케스트레이션해야 하는 경우가 많습니다. SaaS 아키텍처에서는 이 프로세스를 테넌트 온보딩이라고 합니다. 온보딩 프로세스에서 인프라를 코드로 활용하여 모든 SaaS 환경에서 온보딩을 완전히 자동화해야 합니다.

이 패턴은 Amazon Web Services(AWS)에서 테넌트를 생성하고 테넌트를 위한 기본 인프라를 프로비저닝하는 예를 안내합니다. 패턴은 C#과 AWS Cloud Development Kit(AWS CDK)를 사용합니다.

이 패턴은 결제 경보를 생성하므로 스택을 미국 동부(버지니아 북부) 또는 us-east-1 AWS 리전에 배포하는 것이 좋습니다. 자세한 내용은 [AWS 설명서](#)를 참조하세요.

사전 조건 및 제한 사항

사전 조건

- 활성 상태의 [AWS 계정](#).

- IAM 액세스 권한이 충분하여 이 패턴에 대한 AWS 리소스를 생성할 수 있는 AWS Identity and Access Management(IAM) 주체. 자세한 내용은 [IAM 역할](#)을 참조하세요.
- [Amazon Command Line Interface\(AWS CLI\)를 설치하고](#) AWS CDK 배포를 수행하도록 [AWS CLI를 구성합니다.](#)
- [Visual Studio 2022](#)를 다운로드 및 설치하거나 [Visual Studio 코드](#)를 다운로드하여 설치합니다.
- [AWS Toolkit for Visual Studio](#) 설정.
- [.NET Core 3.1 이상](#)(C# AWS CDK 애플리케이션에 필요)
- [Amazon.Lambda.Tools](#) 설치

제한 사항

- AWS CDK는 [AWS를 CloudFormation](#) 사용하므로 AWS CDK 애플리케이션에는 CloudFormation 서비스 할당량이 적용됩니다. 자세한 내용은 [AWS CloudFormation 할당량](#)을 참조하십시오.
- 테넌트 CloudFormation 스택은 작업 (sns* 및 sqs*) 에 와일드카드 문자가 infra-cloudformation-role 있는 CloudFormation 서비스 역할과 함께 생성되지만 리소스는 접두사로 고정됩니다. tenant-cluster 프로덕션 사용 사례의 경우 이 설정을 평가하고 이 서비스 역할에 필요한 액세스 권한만 제공합니다. 또한 InfrastructureProvision Lambda 함수는 와일드카드 문자 cloudformation* () 를 사용하여 스택을 프로비저닝하지만 리소스는 CloudFormation 접두사로 고정됩니다. tenant-cluster
- 이 예제 코드의 docker 빌드는 linux/amd64 기반 이미지를 강제 실행하는 데 --platform=linux/amd64를 사용합니다. 이는 최종 이미지 아티팩트가 기본적으로 x86-64 아키텍처를 사용하는 Lambda에 적합하도록 하기 위한 것입니다. 대상 Lambda 아키텍처를 변경해야 하는 경우 Dockerfile과 AWS CDK 코드를 모두 변경해야 합니다. 자세한 내용은 [AWS Lambda 함수를 ARM 기반 AWS Graviton2 프로세서로 마이그레이션하기](#) 블로그 게시물을 참조하세요.
- 스택 삭제 프로세스는 스택에서 생성된 CloudWatch 로그 (로그 그룹 및 로그) 를 정리하지 않습니다. AWS 관리 콘솔 Amazon CloudWatch 콘솔 또는 API를 통해 수동으로 로그를 정리해야 합니다.

이 패턴은 예시로 설정되었습니다. 프로덕션 용도로 사용하려면 다음 설정을 평가하고 비즈니스 요구 사항에 따라 변경합니다.

- 이 예제의 [AWS Simple Storage Service\(Amazon S3\)](#) 버킷에는 간편성을 위해 버전 관리 기능이 제공되지 않습니다. 필요에 따라 설정을 평가하고 업데이트합니다.
- 이 예제에서는 간편성을 위해 인증, 권한 부여 또는 스토어링 없이 [Amazon API Gateway](#) REST API 엔드포인트를 설정합니다. 프로덕션 용도로는 시스템을 비즈니스 보안 인프라와 통합하는 것이 좋습니다. 이 설정을 평가하고 필요에 따라 필요한 보안 설정을 추가합니다.

- 이 테넌트 인프라 예제의 경우 [Amazon Simple Notification Service\(Amazon SNS\)](#) 및 [Amazon Simple Queue Service\(Amazon SQS\)](#)는 최소 설정만 사용합니다. 각 테넌트의 [AWS 키 관리 서비스\(AWS KMS\)](#)는 [AWS KMS](#) 키 정책에 따라 계정 내 [Amazon CloudWatch](#) 및 Amazon SNS 서비스를 사용할 수 있도록 개방합니다. 설정은 플레이스홀더 예제일 뿐입니다. 비즈니스 사용 사례의 필요에 따라 설정을 조정합니다.
- CloudFormationAWS를 사용한 API 엔드포인트와 백엔드 테넌트 프로비저닝 및 삭제를 포함하나 이에 국한되지 않는 전체 설정에서는 기본적인 해피 패스 사례만 다룹니다. 비즈니스 요구 사항에 따라 필요한 재시도 로직, 추가 오류 처리 로직, 보안 로직으로 설정을 평가하고 업데이트합니다.
- 예제 코드는 본 작성 시점의 정책을 확인하기 위해 up-to-date [cdk-nag](#)로 테스트되었습니다. 향후 새로운 정책이 시행될 수 있습니다. 이러한 새 정책에 따라 스택을 배포하기 전에 권장 사항에 맞춰 스택을 수동으로 수정해야 할 수도 있습니다. 기존 코드를 검토하여 비즈니스 요구 사항에 맞는지 확인합니다.
- 코드는 생성된 리소스 대부분에 대해 정적으로 할당된 물리적 이름을 사용하는 대신 AWS CDK를 사용하여 임의의 접미사를 생성합니다. 이 설정은 이러한 리소스가 고유하고 다른 스택과 충돌하지 않도록 하기 위한 것입니다. 자세한 내용은 [AWS CDK 설명서](#)를 참조하세요. 비즈니스 요구 사항에 따라 이를 조정합니다.
- 이 예제 코드는 .NET Lambda 아티팩트를 Docker 기반 이미지로 패키징하고 Lambda가 제공한 [컨테이너 이미지 런타임](#)으로 실행합니다. 컨테이너 이미지 런타임은 표준 전송 및 저장 메커니즘(컨테이너 레지스트리)과 보다 정확한 로컬 테스트 환경(컨테이너 이미지를 통한)의 이점을 제공합니다. [Lambda에서 제공한 .NET 런타임](#)을 사용하도록 프로젝트를 전환하여 Docker 이미지의 빌드 시간을 줄일 수 있지만, 그런 다음 전송 및 저장 메커니즘을 설정하고 로컬 설정이 Lambda 설정과 일치하는지 확인해야 합니다. 사용자의 비즈니스 요구 사항에 맞게 코드를 조정합니다.

제품 버전

- AWS CDK 버전 2.45.0 이상
- Visual Studio 2022

아키텍처

기술 스택

- Amazon API Gateway
- AWS CloudFormation
- 아마존 CloudWatch

- Amazon DynamoDB
- AWS Identity and Access Management(IAM)
- AWS KMS
- AWS Lambda
- Amazon S3
- Amazon SNS
- Amazon SQS

아키텍처

다음 다이어그램은 테넌트 스택 생성 흐름을 나타냅니다. 컨트롤 플레인 및 테넌트 기술 스택에 대한 자세한 내용은 추가 정보 섹션을 참조하세요.

테넌트 스택 생성 흐름

1. 사용자가 JSON으로 새 테넌트 페이로드(테넌트 이름, 테넌트 설명)가 포함된 POST API 요청을 Amazon API Gateway에서 호스팅하는 REST API에 보냅니다. API Gateway는 요청을 처리하여 백엔드 Lambda 테넌트 온보딩 함수로 전달합니다. 이 예시에서는 권한 부여 또는 인증이 없습니다. 프로덕션 설정에서는 이 API를 SaaS 인프라 보안 시스템과 통합해야 합니다.
2. 테넌트 온보딩 기능은 요청을 확인합니다. 그런 다음 테넌트 이름, 생성된 테넌트 범용 고유 식별자(UUID) 및 테넌트 설명이 포함된 테넌트 레코드를 Amazon DynamoDB 테넌트 온보딩 테이블에 저장하려고 시도합니다.
3. DynamoDB가 레코드를 저장한 후 DynamoDB 스트림은 다운스트림 Lambda 테넌트 인프라 함수를 시작합니다.
4. 테넌트 인프라 Lambda 함수는 수신된 DynamoDB 스트림을 기반으로 작동합니다. 스트림이 INSERT 이벤트용 스트림인 경우 함수는 스트림의 NewImage 섹션(최신 업데이트 레코드, 테넌트 이름 필드)을 사용하여 S3 버킷에 저장된 템플릿을 사용하여 새 테넌트 인프라를 CloudFormation 호출합니다. CloudFormation 템플릿에는 테넌트 이름 파라미터가 필요합니다.
5. AWS는 CloudFormation 템플릿과 입력 파라미터를 기반으로 테넌트 인프라를 CloudFormation 생성합니다.
6. 각 테넌트 인프라 설정에는 CloudWatch 경고, 청구 경고 및 경고 이벤트가 있습니다.
7. 알람 이벤트는 테넌트의 AWS KMS 키로 암호화된 SNS 주제에 대한 메시지가 됩니다.

8. SNS 주제는 수신된 경보 메시지를 SQS 대기열로 전달하며, 이 대기열은 테넌트의 AWS KMS에 의해 암호화 키용으로 암호화됩니다.

다른 시스템을 Amazon SQS와 통합하여 대기열에 있는 메시지를 기반으로 작업을 수행할 수 있습니다. 이 예시에서는 코드를 일반화하기 위해 수신 메시지가 대기열에 남아 있으므로 수동으로 삭제해야 합니다.

테넌트 스택 삭제 흐름

1. 사용자가 JSON으로 새 테넌트 페이로드(테넌트 이름, 테넌트 설명)가 포함된 DELETE API 요청을 Amazon API Gateway에서 호스팅하는 REST API로 전송하면 Amazon API Gateway가 요청을 처리하고 테넌트 온보딩 기능으로 전달합니다. 이 예시에서는 권한 부여 또는 인증이 없습니다. 프로덕션 설정에서 이 API는 SaaS 인프라 보안 시스템과 통합됩니다.
2. 테넌트 온보딩 기능은 요청을 확인한 다음 테넌트 온보딩 테이블에서 테넌트 레코드(테넌트 이름)를 삭제하려고 시도합니다.
3. DynamoDB가 레코드를 성공적으로 삭제하면(레코드가 테이블에 존재하고 삭제됨), DynamoDB 스트림은 다운스트림 Lambda 테넌트 인프라 함수를 시작합니다.
4. 테넌트 인프라 Lambda 함수는 수신된 DynamoDB 스트림 레코드를 기반으로 작동합니다. 스트림이 REMOVE 이벤트용인 경우 함수는 레코드 OldImage 섹션 (레코드 정보 및 테넌트 이름 필드, 최근 변경 이전의 테넌트 이름 필드, 삭제) 을 사용하여 해당 레코드 정보를 기반으로 기존 스택의 삭제를 시작합니다.
5. AWS는 입력에 따라 대상 테넌트 스택을 CloudFormation 삭제합니다.

도구

서비스

- [Amazon API Gateway](#)를 사용하면 어떤 규모에서든 REST, HTTP 및 WebSocket API를 생성, 게시, 유지 관리, 모니터링 및 보호할 수 있습니다.
- [AWS Cloud Development Kit\(AWS CDK\)](#)는 AWS 클라우드 인프라를 코드로 정의하고 프로비저닝하는 데 도움이 되는 소프트웨어 개발 프레임워크입니다.
- [AWS CDK Toolkit](#)은 AWS Cloud Development Kit(AWS CDK) 앱과 상호 작용하는 데 도움이 되는 명령줄 클라우드 개발 키트입니다.
- [AWS Command Line Interface \(AWS CLI\)](#)는 명령줄 셸에서 명령을 사용하여 AWS 서비스와 상호 작용할 수 있는 오픈 소스 도구입니다.

- [AWS는 AWS CloudFormation](#) 리소스를 설정하고, 빠르고 일관되게 프로비저닝하고, 전체 AWS 계정 및 지역의 수명 주기 전반에 걸쳐 리소스를 관리할 수 있도록 지원합니다.
- [Amazon DynamoDB](#)는 빠르고 예측 가능하고 확장 가능한 성능을 제공하는 완전 관리형 NoSQL 데이터베이스 서비스입니다.
- [AWS Identity and Access Management\(IAM\)](#)는 사용자에게 대한 인증 및 권한 부여를 제어함으로써 AWS 리소스에 대한 액세스를 안전하게 제어할 수 있습니다.
- [AWS Key Management Service\(AWS KMS\)](#)를 사용하면 암호화 키를 생성하고 제어하여 데이터를 보호할 수 있습니다.
- [Lambda](#)는 서버를 프로비저닝하거나 관리할 필요 없이 코드를 실행할 수 있도록 도와주는 컴퓨팅 서비스입니다. 필요할 때만 코드를 실행하며 자동으로 확장이 가능하므로 사용한 컴퓨팅 시간만큼만 비용을 지불합니다.
- [Amazon Simple Storage Service\(Amazon S3\)](#)는 원하는 양의 데이터를 저장, 보호 및 검색하는 데 도움이 되는 클라우드 기반 객체 스토리지 서비스입니다.
- [Amazon Simple Notification Service\(Amazon SNS\)](#)를 사용하면 웹 서버 및 이메일 주소를 포함하여 게시자와 클라이언트 간의 메시지 교환을 조정하고 관리할 수 있습니다.
- [Amazon SQS](#)는 내구력 있고 가용성이 뛰어난 보안 호스팅 대기열을 제공하며 이를 통해 분산 소프트웨어 시스템과 구성 요소를 통합 및 분리할 수 있습니다.
- [Visual Studio용 AWS Toolkit](#)은 Visual Studio 통합 개발 환경(IDE)을 위한 플러그인입니다. Visual Studio용 Toolkit은 AWS 서비스를 사용하는 .NET 애플리케이션의 개발, 디버깅 및 배포를 지원합니다.

기타 도구

- [Visual Studio](#)는 컴파일러, 코드 완성 도구, 그래픽 디자이너 및 소프트웨어 개발을 지원하는 기타 기능을 포함하는 IDE입니다.

코드

이 패턴의 코드는 [사일로 모델 APG용 SaaS 아키텍처의 테넌트 온보딩 예제](#) 리포지토리에 있습니다.

에픽

AWS CDK 설정

작업	설명	필요한 기술
Node.js 설치를 확인합니다.	로컬 시스템에 Node.js가 설치되었는지 확인하려면 다음 명령을 실행합니다. <pre>node --version</pre>	AWS 관리자, AWS DevOps
AWS CDK Toolkit을 설치합니다.	로컬 시스템에 AWS CDK Toolkit을 설치하려면 다음 명령을 실행합니다. <pre>npm install -g aws-cdk</pre> npm이 설치되지 않은 경우 Node.js 사이트 에서 설치할 수 있습니다.	AWS 관리자, AWS DevOps
AWS CDK Toolkit 버전을 확인합니다.	AWS CDK Toolkit 버전이 시스템에 제대로 설치되었는지 확인하려면 다음 명령을 실행합니다. <pre>cdk --version</pre>	AWS 관리자, AWS DevOps

테넌트 온보딩 컨트롤 플레인의 코드를 검토합니다.

작업	설명	필요한 기술
리포지토리를 복제합니다.	리포지토리 를 복제하고 <code>\tenant-onboarding-in-saas-architect</code>	AWS 관리자, AWS DevOps

작업	설명	필요한 기술
	<p>ure-for-silo-model-apg-example 폴더로 이동합니다.</p> <p>Visual Studio 2022에서 \src\TenantOnboardingInfra.sln 솔루션 파일을 엽니다. TenantOnboardingInfraStack.cs 파일을 열고 코드를 검토합니다.</p> <p>이 스택의 일부로 생성되는 리소스는 다음과 같습니다.</p> <ul style="list-style-type: none"> • DynamoDB 테이블 • S3 버킷 (CloudFormation 템플릿을 S3 버킷에 업로드) • Lambda 실행 역할 • Lambda 함수 • API Gateway API • Lambda 함수에 대한 이벤트 소스 	

작업	설명	필요한 기술
<p>CloudFormation 템플릿을 검토하십시오.</p>	<p><code>\tenant-onboarding-in-saas-architecture-for-silo-model-apg-example\template</code> 폴더에서 CloudFormation 템플릿을 열고 <code>infra.yaml</code> 검토합니다. 이 템플릿은 테넌트 온보딩 DynamoDB 테이블에서 검색된 테넌트 이름으로 하이드레이트됩니다.</p> <p>템플릿은 테넌트별 인프라를 제공합니다. 이 예시에서는 AWS KMS 키, 아마존 SNS, 아마존 SQS 및 경보를 프로비저닝합니다. CloudWatch</p>	<p>앱 개발자, AWS DevOps</p>

작업	설명	필요한 기술
<p>테넌트 온보딩 기능을 검토합니다.</p>	<p>Function.cs 를 열고 .NET 6(컨테이너 이미지) 청사진이 포함된 Visual Studio AWS Lambda 프로젝트(.NET Core-C#) 템플릿으로 생성된 테넌트 온보딩 함수의 코드를 검토합니다.</p> <p>Dockerfile 을 열고 코드를 검토합니다. Dockerfile 은 Lambda 컨테이너 이미지를 빌드하기 위한 지침으로 구성된 텍스트 파일입니다.</p> <p>다음 NuGet 패키지가 TenantOnboardingFunction 프로젝트에 종속성으로 추가된다는 점에 유의하십시오.</p> <ul style="list-style-type: none"> • Amazon.Lambda.APIGatewayEvents • AWSSDK.DynamoDBv2 • Newtonsoft.Json 	<p>앱 개발자, AWS DevOps</p>

작업	설명	필요한 기술
<p>테넌트 InfraProvisioning 기능을 검토하십시오.</p>	<p><code>\tenant-onboarding-in-saas-architecture-for-silo-model-apg-example\src\InfraProvisioningFunction</code> 로 이동합니다.</p> <p><code>Function.cs</code> 를 열고 .NET 6(컨테이너 이미지) 청사진이 포함된 Visual Studio AWS Lambda 프로젝트(.NET Core-C#) 템플릿으로 생성된 테넌트 인프라 프로비저닝 함수의 코드를 검토합니다.</p> <p><code>Dockerfile</code> 을 열고 코드를 검토합니다.</p> <p>다음 NuGet 패키지가 <code>InfraProvisioningFunction</code> 프로젝트에 종속 항목으로 추가된다는 점에 유의하십시오.</p> <ul style="list-style-type: none"> • <code>Amazon.Lambda.DynamoDBEvents</code> • <code>AWSSDK.DynamoDBv2</code> • <code>AWSSDK.Cloudformation</code> 	<p>앱 개발자, AWS DevOps</p>

AWS 리소스 배포

작업	설명	필요한 기술
<p>솔루션을 빌드합니다.</p>	<p>솔루션을 구축하려면 다음 단계를 수행합니다.</p> <ol style="list-style-type: none"> 1. Visual Studio 2022에서 <code>\tenant-onboarding-in-saas-architecture-for-silo-model-apg-example\src\TenantOnboardingInfra.sln</code> 솔루션 파일을 엽니다. 2. 솔루션의 컨텍스트 메뉴(마우스 오른쪽 클릭)를 열고 솔루션 빌드를 선택합니다. <p>참고: 솔루션을 빌드하기 전에 <code>\tenant-onboarding-in-saas-architecture-for-silo-model-apg-example\src\TenantOnboardingInfra</code> 프로젝트에서 <code>Amazon.CDK.Lib</code> NuGet 패키지를 최신 버전으로 업데이트해야 합니다.</p>	<p>앱 개발자</p>
<p>AWS CDK 환경을 부트스트랩합니다.</p>	<p>Windows 명령 프롬프트를 열고 <code>cdk.json</code> 파일을 사용할 수 있는 AWS CDK 앱 루트 폴더로 이동합니다(<code>\tenant-onboarding-in-saas-architecture-for-silo-model-apg-exam</code></p>	<p>AWS 관리자, AWS DevOps</p>

작업	설명	필요한 기술
	<p>ple). 부트스트래핑을 위해 다음 명령을 실행합니다.</p> <pre>cdk bootstrap</pre> <p>보안 인증 정보에 대한 AWS 프로필을 생성한 경우 프로필과 함께 명령을 사용합니다.</p> <pre>cdk bootstrap --profile <profile name></pre>	
<p>AWS CDK 스택을 나열합니다.</p>	<p>이 프로젝트의 일부로 생성할 스택을 모두 나열하려면 다음 명령을 실행합니다.</p> <pre>cdk ls cdk ls --profile <profile name></pre> <p>보안 인증 정보에 대한 AWS 프로필을 생성한 경우 프로필과 함께 명령을 사용합니다.</p> <pre>cdk ls --profile <profile name></pre>	<p>AWS 관리자, AWS DevOps</p>

작업	설명	필요한 기술
생성될 AWS 리소스를 검토합니다.	<p>이 프로젝트의 일부로 생성될 모든 AWS 리소스를 검토하려면 다음 명령을 실행합니다.</p> <pre>cdk diff</pre> <p>보안 인증 정보에 대한 AWS 프로필을 생성한 경우 프로필과 함께 명령을 사용합니다.</p> <pre>cdk diff --profile <profile name></pre>	AWS 관리자, AWS DevOps

작업	설명	필요한 기술
AWS CDK를 사용하여 모든 AWS 리소스를 배포합니다.	<p>모든 AWS 리소스를 배포하려면 다음 명령을 실행합니다.</p> <pre>cdk deploy --all --require-approval never</pre> <p>보안 인증 정보에 대한 AWS 프로필을 생성한 경우 프로필과 함께 명령을 사용합니다.</p> <pre>cdk deploy --all --require-approval never --profile <profile name></pre> <p>배포가 완료되면 다음 예와 같이 명령 프롬프트의 출력 섹션에서 API URL을 복사합니다.</p> <pre>Outputs: TenantOnboardingInfraStack.TenantOnboardingAPIEndpoint 42E526D7 = https://j2qmp8ds21i1i.execute-api.us-west-2.amazonaws.com/prod/</pre>	AWS 관리자, AWS DevOps

기능성 확인

작업	설명	필요한 기술
새 테넌트 생성.	새 테넌트를 만들려면 다음 curl 요청을 보냅니다.	앱 개발자, AWS 관리자, AWS DevOps

작업	설명	필요한 기술
	<pre>curl -X POST <TenantOnboardingAPIEndpoint* from CDK Output>tenant -d '{"Name":"Tenant123", "Description":"Stack for Tenant123"}'</pre> <p>다음 예제와 같이 자리 표시자 <TenantOnboardingAPIEndpoint* from CDK Output>를 AWS CDK의 실제 값으로 변경합니다.</p> <pre>curl -X POST https://j2qmp8ds21i1i.execute-api.us-west-2.amazonaws.com/prod/tenant -d '{"Name":"Tenant123", "Description":"test12"}'</pre> <p>다음은 출력을 보여주는 예제입니다.</p> <pre>{"message": "A new tenant added - 5/4/2022 7:11:30 AM"}</pre>	

작업	설명	필요한 기술
<p>DynamoDB에서 새로 생성한 테넌트 세부 정보를 확인합니다.</p>	<p>DynamoDB에서 새로 생성된 테넌트 세부 정보를 확인하려면 다음 단계를 수행합니다.</p> <ol style="list-style-type: none"> 1. AWS Management Console을 열고 Amazon DynamoDB 서비스로 이동합니다. 2. 왼쪽 탐색 창에서 항목 탐색을 선택하고 TenantOnboarding 테이블을 선택합니다. <p>참고: 테넌트 이름 앞에 <code>tenantcluster-</code> 가 추가됩니다. 자세한 내용은 추가 정보 섹션을 참조하세요.</p> <ol style="list-style-type: none"> 3. 테넌트 세부 정보로 새 항목이 생성되었는지 확인합니다. 	<p>앱 개발자, AWS 관리자, AWS DevOps</p>

작업	설명	필요한 기술
<p>새 테넌트의 스택 생성을 확인합니다.</p>	<p>새 스택이 성공적으로 생성되고 템플릿에 따라 새로 생성된 테넌트를 위한 인프라로 프로비저닝되었는지 확인하십시오.</p> <p>CloudFormation</p> <ol style="list-style-type: none"> 1. 콘솔을 엽니다. CloudFormation 2. 왼쪽 탐색창에서 스택을 선택하고 테넌트 이름이 포함된 스택이 성공적으로 생성되었는지 확인합니다. 3. 새로 생성된 테넌트 스택을 선택한 다음 리소스 탭을 선택합니다. 알람 리소스와 Amazon SQS 리소스를 기록해 둡니다. 4. AWS 보안 인증 정보가 구성된 새 터미널을 열고 올바른 리전을 가리킵니다. 테스트 경보를 발생시키려면 다음 코드를 입력하고 <alarm resource name>을 3단계에서 기록해 둔 경보 리소스 이름으로 대체합니다. <pre>aws cloudwatch set-alarm-state --alarm-name <alarm resource name> --state-value ALARM --state-reason 'Test setup'</pre>	<p>앱 개발자, AWS 관리자, AWS DevOps</p>

작업	설명	필요한 기술
	<p>다음 예에서는 경보 리소스 이름이 있는 코드를 보여줍니다.</p> <pre data-bbox="630 380 1029 695">aws cloudwatch set-alarm-state --alarm-name tenantcluster-tenant123-alarm --state-value ALARM --state-reason 'Test setup'</pre> <p>5. 콘솔을 열고 Amazon SQS 콘솔로 이동합니다. 3단계에서 식별한 Amazon SQS 리소스 이름을 선택합니다. AWS 설명서 지침에 따라 4단계에서 발생한 경보에서 테스트 메시지를 수신하고 삭제합니다.</p>	

작업	설명	필요한 기술
<p>테넌트 스택을 삭제합니다.</p>	<p>테넌트 스택을 삭제하려면 다음 curl 요청을 보냅니다.</p> <pre data-bbox="597 348 1027 583">curl -X DELETE <TenantOnboardingAPIEndpoint* from CDK Output>tenant/<Tenant Name from previous step></pre> <p>다음 예와 같이 플레이스홀더 <TenantOnboardingAPIEndpoint* from CDK Output>를 AWS CDK의 실제 값으로 변경하고 <Tenant Name from previous step>을 이전 테넌트 생성 단계의 실제 값으로 변경합니다.</p> <pre data-bbox="597 1031 1027 1266">curl -X DELETE https://j2qmp8ds21i1i.execute-api.us-west-2.amazonaws.com/prod/tenant/Tenant123</pre> <p>다음은 출력을 보여주는 예제입니다.</p> <pre data-bbox="597 1430 1027 1583">{"message": "Tenant destroyed - 5/4/2022 7:14:48 AM"}</pre>	<p>앱 개발자, AWS DevOps, AWS 관리자</p>

작업	설명	필요한 기술
기존 테넌트의 스택 삭제를 확인합니다.	<p>기존 테넌트 스택이 삭제되었는지 확인하려면 다음 단계를 수행합니다.</p> <ol style="list-style-type: none"> 1. 콘솔을 열고 CloudFormation 콘솔로 이동합니다. 2. 왼쪽 탐색창에서 테넌트 이름이 있는 기존 스택이 더 이상 콘솔에 없거나 (CloudFormation 콘솔이 활성 스택만 표시하도록 설정된 경우) 삭제 중인지 확인합니다. 스택이 콘솔에 더 이상 없는 경우 드롭다운 목록을 사용하여 CloudFormation 콘솔 설정을 Active에서 Deleted (삭제됨)로 변경하여 삭제된 스택을 확인하고 스택이 성공적으로 삭제되었는지 확인합니다. 	앱 개발자, AWS 관리자, AWS DevOps

정리

작업	설명	필요한 기술
환경 파괴.	<p>스택을 정리하기 전에 다음을 확인합니다.</p> <ul style="list-style-type: none"> • DynamoDB의 모든 레코드가 이전 테넌트 삭제 작업이나 DynamoDB 콘솔 또는 API를 통해 제거되었습니다. 각 테넌트 레코드 삭제는 해당 	AWS 관리자, AWS DevOps

작업	설명	필요한 기술
	<p>AWS CloudFormation 대응 물의 정리를 시작합니다.</p> <ul style="list-style-type: none"> 모든 테넌트 기반 AWS CloudFormation 스택은 AWS 콘솔에서 정리됩니다 (DynamoDB 트리거 정리 로직이 실패할 경우). CloudFormation <p>테스트가 완료되면 AWS CDK 를 사용하여 다음 명령을 실행 하여 모든 스택과 관련 리소스를 제거할 수 있습니다.</p> <pre>cdk destroy --all;</pre> <p>보안 인증에 대한 AWS 프로필을 생성한 경우 해당 프로필을 사용합니다.</p> <p>스택 삭제 프롬프트를 확인하여 스택을 삭제합니다.</p>	
Amazon CloudWatch 로그를 정리하십시오.	스택 삭제 프로세스는 스택에서 생성된 CloudWatch 로그 (로그 그룹 및 로그) 를 정리하지 않습니다. CloudWatch 콘솔이나 API를 사용하여 CloudWatch 리소스를 수동으로 정리합니다.	앱 개발자, AWS DevOps, AWS 관리자

관련 리소스

- [AWS CDK .NET 워크숍](#)

- [C#에서 AWS CDK 사용하기](#)
- [CDK .NET 참조](#)

추가 정보

컨트롤 플레인 기술 스택

.NET으로 작성된 CDK 코드는 다음 리소스로 구성된 컨트롤 플레인 인프라를 프로비저닝하는 데 사용됩니다.

1. API Gateway

컨트롤 플레인 스택의 REST API 진입점 역할을 합니다.

2. 테넌트 온보딩 Lambda 함수

이 Lambda 함수는 API Gateway에서 m 메서드를 사용하여 시작합니다.

POST 메서드 API 요청으로 인해 DynamoDB Tenant Onboarding 테이블에 (tenant name 및 tenant description)이 삽입됩니다.

이 코드 예제에서 테넌트 이름은 테넌트 스택 이름 및 해당 스택 내 리소스 이름의 일부로도 사용됩니다. 이는 이러한 리소스를 더 쉽게 식별할 수 있도록 하기 위한 것입니다. 충돌이나 오류를 방지하려면 설정에서 이 테넌트 이름이 고유해야 합니다. 자세한 입력 검증 설정은 [IAM 역할](#) 설명서 및 제한 섹션에 설명되어 있습니다.

DynamoDB 테이블에 대한 지속성 프로세스는 테넌트 이름이 테이블의 다른 레코드에서 사용되지 않는 경우에만 성공합니다.

파티션 키만 PutItem 조건 표현식으로 사용할 수 있기 때문에 이 경우 테넌트 이름이 해당 테이블의 파티션 키입니다.

테넌트 이름이 이전에 기록된 적이 없는 경우 레코드가 테이블에 성공적으로 저장됩니다.

그러나 테이블의 기존 레코드에서 테넌트 이름을 이미 사용하고 있는 경우 작업이 실패하고 DynamoDB ConditionalCheckFailedException 예외가 시작됩니다. 이 예외는 테넌트 이름이 이미 존재함을 나타내는 실패 메시지(HTTP BadRequest)를 반환하는 데 사용됩니다.

DELETE 메서드 API 요청은 Tenant Onboarding 테이블에서 특정 테넌트 이름에 대한 레코드를 제거합니다.

이 예제의 DynamoDB 레코드 삭제는 레코드가 없더라도 성공적으로 삭제됩니다.

대상 레코드가 존재하고 삭제되면 DynamoDB 스트림 레코드가 생성됩니다. 그렇지 않으면 다운스트림 레코드가 생성되지 않습니다.

3. 테넌트 온보딩 DynamoDB(Amazon DynamoDB 스트림 사용)

이렇게 하면 테넌트 메타데이터 정보가 기록되며, 모든 레코드 저장 또는 삭제가 Tenant Infrastructure Lambda 함수로 다운스트림을 전송합니다.

4. 테넌트 인프라 Lambda 함수

이 Lambda 함수는 이전 단계의 DynamoDB 스트림 레코드에 의해 시작됩니다. INSERT 이벤트용 레코드의 경우 AWS를 CloudFormation 호출하여 S3 버킷에 저장된 CloudFormation 템플릿으로 새 테넌트 인프라를 생성합니다. REMOVE를 위한 레코드의 경우 스트림 레코드의 Tenant Name 필드를 기반으로 기존 스택의 삭제를 시작합니다.

5. S3 버킷

이는 CloudFormation 템플릿을 저장하기 위한 것입니다.

6. 각 Lambda 함수의 IAM 역할 및 다음에 대한 서비스 역할 CloudFormation

각 Lambda 함수에는 작업을 수행하기 위한 [최소 권한](#)을 가진 고유한 IAM 역할이 있습니다. 예를 들어, Tenant On-boarding Lambda 함수는 DynamoDB에 대한 읽기/쓰기 액세스 권한을 가지고 있으며 Tenant Infrastructure Lambda 함수는 DynamoDB 스트림을 읽기만 할 수 있습니다.

테넌트 스택 프로비저닝을 위한 사용자 지정 CloudFormation 서비스 역할이 생성됩니다. 이 서비스 역할에는 CloudFormation 스택 프로비저닝에 대한 추가 권한 (예: AWS KMS 키) 이 포함되어 있습니다. 이렇게 하면 Lambda 간에 역할이 나뉘어 단일 역할 (인프라 Lambda CloudFormation 역할)에 대한 모든 권한을 피할 수 있습니다.

강력한 작업 (예: CloudFormation 스택 생성 및 삭제) 을 허용하는 권한은 잠겨 있으며 로 시작하는 리소스에만 허용됩니다. `tenantcluster-` 단, AWS KMS는 리소스 명명 규칙 때문에 예외입니다. API에서 수집된 테넌트 이름은 다른 검증 검사와 함께 앞에 `tenantcluster-`가 추가됩니다(대시가 포함된 영숫자, 대부분의 AWS 리소스 이름 지정에 맞게 30자 미만으로 제한). 이렇게 하면 테넌트 이름으로 인해 실수로 핵심 인프라 스택이나 리소스가 중단되는 일이 발생하지 않습니다.

테넌트 기술 스택

CloudFormation 템플릿은 S3 버킷에 저장됩니다. [템플릿은 테넌트별 AWS KMS 키, CloudWatch 경보, SNS 주제, SQS 대기열, SQS 정책을 제공합니다.](#)

AWS KMS 키는 Amazon SNS와 Amazon SQS에서 메시지의 데이터를 암호화하는 데 사용됩니다. [AwsSolutions-SNS2 및 AwsSolutions -SQS2의 보안 관행에서는 암호화를 사용하여 Amazon SNS와 Amazon SQS를 설정하는 것이 좋습니다.](#) 하지만 AWS 관리 키를 사용할 때는 Amazon SNS에서 CloudWatch 경보가 작동하지 않으므로 이 경우에는 고객 관리 키를 사용해야 합니다. 자세한 정보는 [AWS 지식 센터](#)를 참조하세요.

Amazon SQS 대기열에서는 생성된 SNS 주제가 메시지를 대기열에 전송할 수 있도록 허용하는 데 SQS 정책이 사용됩니다. SQS 정책이 없으면 액세스가 거부됩니다. 자세한 내용은 [Amazon SNS 설명서](#)를 참조하세요.

CQRS 및 이벤트 소싱을 사용하여 모놀리식 유형을 마이크로서비스로 분해하기

작성자: Rodolfo Jr. Cerrada(AWS), Dmitry Gulin(AWS), Tabby Ward(AWS)

환경: PoC 또는 파일럿	소스: 모놀리식 CRUD 모델	대상: 마이크로서비스
R 유형: 리아키텍트	워크로드: 오픈 소스	기술: 현대화, 메시징 및 통신, 서버리스
AWS 서비스: Amazon DynamoDB, AWS Lambda, Amazon SNS		

요약

이 패턴은 명령 쿼리 책임 분리(CQRS) 패턴과 이벤트 소싱 패턴을 모두 사용하는 두 가지 패턴을 결합합니다. CQRS 패턴은 명령 모델과 쿼리 모델의 책임을 구분합니다. 이벤트 소싱 패턴은 비동기 이벤트 기반 통신을 활용하여 전반적인 사용자 경험을 개선합니다.

CQRS 및 Amazon Web Services(AWS) 서비스를 사용하여 각 데이터 모델을 독립적으로 유지 관리하고 규모를 조정하는 동시에 모놀리식 애플리케이션을 마이크로서비스 아키텍처로 리팩터링할 수 있습니다. 그런 다음 이벤트 소싱 패턴을 사용하여 명령 데이터베이스의 데이터를 쿼리 데이터베이스와 동기화할 수 있습니다.

이 패턴은 최신 버전의 Visual Studio를 사용하여 열 수 있는 솔루션(*.sln) 파일이 포함된 예제 코드를 사용합니다. 이 예제에는 AWS 서버리스 및 기존 또는 온프레미스 애플리케이션에서 CQRS 및 이벤트 소싱이 어떻게 작동하는지 보여주는 Reward API 코드가 포함되어 있습니다.

CQRS 및 이벤트 소싱에 대해 자세히 알아보려면 [추가 정보](#) 섹션을 참조하십시오.

사전 조건 및 제한 사항

사전 조건

- 활성 상태의 AWS 계정.

- 아마존 CloudWatch
- Amazon DynamoDB 테이블
- Amazon DynamoDB Streams
- AWS Identity and Access Management(IAM) 액세스 키 및 보안 키에 관한 자세한 내용은 관련 리소스 섹션의 동영상을 참조하십시오.
- AWS Lambda
- Visual Studio에 대한 지식
- AWS Toolkit for Visual Studio에 대한 지식, 자세한 내용은 관련 리소스 섹션의 AWS Toolkit for Visual Studio 데모 비디오를 참조하십시오.

제품 버전

- [Visual Studio 2019 Community Edition.](#)
- [AWS Toolkit for Visual Studio 2019.](#)
- .NET Core 3.1. 이 구성 요소는 Visual Studio 설치 시 사용할 수 있는 옵션입니다. 설치 중에 .NET Core를 포함하려면 NET Core 교차 플랫폼 개발을 선택합니다.

제한 사항

- 기존 온프레미스 애플리케이션(ASP.NET Core Web API 및 데이터 액세스 개체)의 예제 코드는 데이터베이스와 함께 제공되지 않습니다. 하지만 여기에는 모의 데이터베이스 역할을 하는 `CustomerData` 인 메모리 객체가 함께 제공됩니다. 제공된 코드는 패턴을 테스트하기에 충분합니다.

아키텍처

소스 기술 스택

- ASP.NET Core Web API 프로젝트
- IIS 웹 서버
- 데이터 액세스 객체
- CRUD 모델

소스 아키텍처

소스 아키텍처에서 CRUD 모델은 하나의 애플리케이션에 명령 인터페이스와 쿼리 인터페이스를 모두 포함합니다. 예제 코드는 CustomerDAO.cs(첨부)를 참조하십시오.

대상 기술 스택

- Amazon DynamoDB
- Amazon DynamoDB Streams
- AWS Lambda
- (선택 사항) Amazon API Gateway
- (선택 사항) Amazon Simple Notification Service(SNS)

대상 아키텍처

대상 아키텍처에서는 명령 인터페이스와 쿼리 인터페이스가 분리되어 있습니다. 다음 다이어그램에 표시된 아키텍처는 API Gateway와 Amazon SNS를 사용하여 확장할 수 있습니다. 자세한 내용은 [추가 정보](#) 섹션을 참조하세요.

1. 명령 Lambda 함수는 데이터베이스에서 생성, 업데이트 또는 삭제와 같은 쓰기 작업을 수행합니다.
2. 쿼리 Lambda 함수는 데이터베이스에서 가져오기 또는 선택과 같은 읽기 작업을 수행합니다.
3. 이 Lambda 함수는 명령 데이터베이스의 DynamoDB 스트림을 처리하고 변경 사항에 맞게 쿼리 데이터베이스를 업데이트합니다.

도구

도구

- [Amazon DynamoDB](#)는 완전 관리형 NoSQL 데이터베이스 서비스로서 원활한 확장성과 함께 빠르고 예측 가능한 성능을 제공합니다.
- [Amazon DynamoDB Streams](#) - DynamoDB Streams는 어떤 DynamoDB 테이블이든 시간 순서에 따라 항목 수준 수정을 캡처합니다. 그런 다음 이 정보를 최대 24시간 동안 로그에 저장합니다. 유틸리티 암호화는 DynamoDB Streams의 데이터를 암호화합니다.
- [AWS Lambda](#) - AWS Lambda는 서버를 프로비저닝하거나 관리하지 않고도 코드를 실행하도록 지원하는 컴퓨팅 서비스입니다. Lambda는 필요 시에만 코드를 실행하며, 일일 몇 개의 요청에서 초당

수천 개의 요청까지 자동으로 규모를 조정합니다. 사용한 컴퓨팅 시간만큼만 비용을 지불하고, 코드가 실행되지 않을 때는 요금이 부과되지 않습니다.

- [AWS Management Console](#) - AWS Management Console은 AWS 서비스 관리를 위한 다양한 서비스 콘솔의 모음을 구성하는 웹 애플리케이션입니다.
- [Visual Studio 2019 Community Edition](#) - Visual Studio 2019는 통합 개발 환경(IDE)입니다. Community Edition은 오픈 소스 기여자에게 무료로 제공됩니다. 이 패턴에서는 Visual Studio 2019 Community Edition을 사용하여 예제 코드를 열고 컴파일하고 실행합니다. 보기 전용으로 모든 텍스트 편집기 또는 [Visual Studio Code](#)를 사용할 수 있습니다.
- [AWS Toolkit for Visual Studio](#) - AWS Toolkit for Visual Studio는 Visual Studio IDE용 플러그인입니다. AWS Toolkit for Visual Studio는 AWS 서비스를 사용하는 .NET 애플리케이션을 더 쉽게 개발, 디버깅 및 배포할 수 있도록 도와줍니다.

code

예제 코드가 첨부되어 있습니다. 예제 코드 배포에 대한 지침은 에픽 섹션을 참조하십시오.

에픽

솔루션 열기 및 빌드

작업	설명	필요한 기술
섹션을 여십시오.	<ol style="list-style-type: none"> 1. 첨부 섹션에서 예제 소스 코드(CQRS-ES Code.zip)를 다운로드하고 파일을 추출합니다. 2. Visual Studio IDE에서 파일, 열기, 프로젝트 솔루션을 선택하고 소스 코드를 추출한 폴더로 이동합니다. 3. AWS.APG.CQRSES.sln을 선택한 다음, 열기를 선택합니다. 전체 솔루션이 Visual Studio에 로드됩니다. 	앱 개발자
솔루션을 빌드합니다.	솔루션의 컨텍스트 메뉴(마우스 오른쪽 버튼 클릭)를 연다	앱 개발자

작업	설명	필요한 기술
	<p>음 솔루션 빌드를 선택합니다. 그러면 솔루션의 모든 프로젝트가 빌드되고 컴파일됩니다. 성공적으로 컴파일될 것입니다.</p> <p>Visual Studio 솔루션 탐색기에 디렉터리 구조가 표시되어야 합니다.</p> <ul style="list-style-type: none"> • CQRS On-Premises Code Sample에는 온프레미스에서 CQRS를 사용하는 예제가 포함되어 있습니다. • CQRS AWS Serverless에는 AWS 서버리스 서비스를 사용하는 모든 CQRS 및 이벤트 소싱 예제 코드가 포함되어 있습니다. 	

DynamoDB 테이블 빌드

작업	설명	필요한 기술
보안 인증을 제공합니다.	<p>아직 액세스 키가 없는 경우 관련 리소스 섹션의 비디오를 참조하십시오.</p> <ol style="list-style-type: none"> 1. 솔루션 탐색기에서 CQRS AWS 서버리스를 확장한 다음 빌드 솔루션 폴더를 확장합니다. 2. Aws.APG.CQRSES.Build 프로젝트를 확장하고 	앱 개발자, 데이터 엔지니어, DBA

작업	설명	필요한 기술
	<p>Program.cs 파일을 확인합니다.</p> <p>3. Program.cs 의 맨 위로 스크롤하여 Program() 을 검색합니다.</p> <p>4. YOUR ACCESS KEY를 계정 액세스 키로 바꾸고 YOUR SECRET KEY를 계정 비밀번호로 바꾸십시오. 프로덕션 환경에서는 키를 하드코딩하지 않는다는 점에 유의하십시오. 대신 AWS Secrets Manager를 사용하여 보안 인증을 저장하고 검색할 수 있습니다.</p>	
프로젝트를 빌드합니다.	프로젝트를 구축하려면 AwS.APG.CQRSES.Build 프로젝트의 컨텍스트 메뉴(마우스 오른쪽 버튼 클릭)를 연 다음, 빌드를 선택합니다.	앱 개발자, 데이터 엔지니어, DBA
테이블을 빌드하고 채웁니다.	테이블을 빌드하고 시드 데이터로 채우려면 AwS.APG.CQRSES.Build 프로젝트의 컨텍스트 메뉴(마우스 오른쪽 버튼 클릭)를 열고 디버그, 새 인스턴스 시작을 선택합니다.	앱 개발자, 데이터 엔지니어, DBA
테이블 구성과 데이터를 확인합니다.	확인하려면 AWS 탐색기로 이동하여 Amazon DynamoDB를 확장하십시오. 테이블이 표시되어야 합니다. 각 테이블을 열어 예제 데이터를 표시합니다.	앱 개발자, 데이터 엔지니어, DBA

로컬 테스트 실행

작업	설명	필요한 기술
CQRS 프로젝트를 빌드합니다.	<ol style="list-style-type: none"> 1. 솔루션을 열고 CQRS AWS 서비스/CQRS/테스트 솔루션 폴더로 이동합니다. 2. AWS.apg.cqrse.cqrslambda.tests 프로젝트에서.cs를 열고 생성한 IAM 키로 와 를 교체합니다. BaseFunctionTest AccessKeySecretKey 3. 변경 사항을 저장합니다. 4. 프로젝트를 빌드하려면 프로젝트의 컨텍스트 메뉴(마우스 오른쪽 버튼 클릭)를 연 다음, 빌드를 선택합니다. 	앱 개발자, 테스트 엔지니어
이벤트 소싱 프로젝트를 빌드하십시오.	<ol style="list-style-type: none"> 1. CQRS AWS 서비스/이벤트 소스/테스트 솔루션 폴더로 이동합니다. 2. AWS.APG.CQRSES에서 EventSourceLambda.Tests BaseFunctionTest 프로젝트에서.cs를 열고 생성한 IAM 키로 와 를 AccessKey교체합니다. SecretKey 3. 변경 사항을 저장합니다. 4. 프로젝트를 빌드하려면 프로젝트의 컨텍스트 메뉴(마우스 오른쪽 버튼 클릭)를 연 다음, 빌드를 선택합니다. 	앱 개발자, 테스트 엔지니어
테스트를 실행합니다.	모든 테스트를 실행하려면 보기, 테스트 탐색기를 선택한	앱 개발자, 테스트 엔지니어

작업	설명	필요한 기술
	<p>다음 보기에서 모든 테스트 실행을 선택합니다. 모든 테스트를 통과해야 하며, 이는 녹색 확인 표시 아이콘으로 표시됩니다.</p>	

CQRS Lambda 함수를 AWS에 게시하십시오.

작업	설명	필요한 기술
<p>첫 번째 Lambda 함수를 게시하십시오.</p>	<ol style="list-style-type: none"> 1. 솔루션 탐색기에서 AWS.APG.CQRSES의 컨텍스트 (마우스 오른쪽 버튼 클릭) 메뉴를 엽니다. CommandCreateLambda 프로젝트를 선택한 다음 AWS Lambda에 게시를 선택합니다. 2. 사용하려는 프로필, Lambda 함수를 배포하려는 AWS 리전 및 함수 이름을 선택합니다. 3. 나머지 필드의 기본값을 유지하고 다음을 선택합니다. 4. 역할 이름 드롭다운 목록에서 <code>AWSLambdaFullAccess</code>를 선택합니다. 5. 계정 키를 제공하려면 추가를 선택하고 변수로 <code>AcessKey</code>를 입력하고 액세스 키를 값으로 입력합니다. 그런 다음 추가를 다시 선택하고, 변수로 <code>SecretKey</code> 	<p>앱 개발자, DevOps 엔지니어</p>

작업	설명	필요한 기술
	<p>를 그리고 값으로 비밀 키를 입력합니다.</p> <p>6. 나머지 필드의 경우 기본값을 유지하고 업로드를 선택합니다. Lambda 테스트 함수가 업로드되면 Visual Studio에 자동으로 표시됩니다.</p> <p>7. 다음 프로젝트에 대해 1~6 단계를 반복합니다.</p> <ul style="list-style-type: none"> • AWS.APG.CQRSES.CommandDeleteLambda • AWS.APG.CQRSES.CommandUpdateLambda • AWS.APG.CQRSES.CommandAddRewardLambda • AWS.APG.CQRSES.CommandRedeemRewardLambda • AWS.APG.CQRSES.QueryCustomerListLambda • AWS.APG.CQRSES.QueryRewardLambda 	
함수 업로드를 확인합니다.	(선택 사항) AWS 탐색기로 이동하여 AWS Lambda를 확장하여 함수가 성공적으로 로드되었는지 확인할 수 있습니다. 테스트 창을 열려면 Lambda 함수를 선택합니다(두 번 클릭).	앱 개발자, DevOps 엔지니어

작업	설명	필요한 기술
Lambda 함수를 테스트합니다.	<ol style="list-style-type: none"> 요청 데이터를 입력하거나 추가 정보 섹션의 테스트 데이터에서 예제 요청 데이터를 복사합니다. 테스트 중인 함수에 대한 데이터를 선택했는지 확인하십시오. 테스트를 실행하려면 Invoke를 선택합니다. 응답 및 오류는 응답 텍스트 상자에 표시되고 로그는 로그 텍스트 상자 또는 로그에 CloudWatch 표시됩니다. 데이터를 확인하려면 AWS 탐색기에서 DynamoDB 테이블을 선택합니다(두 번 클릭). <p>모든 CQRS Lambda 프로젝트는 CQRS AWS Serverless\CQRS\Command Microservice 및 CQRS AWS Serverless\CQRS\Command Microservice 솔루션 폴더 아래에 있습니다. 솔루션 디렉터리 및 프로젝트는 추가 정보 섹션의 소스 코드 디렉터리를 참조하십시오.</p>	앱 개발자, DevOps 엔지니어

작업	설명	필요한 기술
나머지 함수를 게시합니다.	<p>다음 프로젝트에 대해 이전 단계를 반복합니다.</p> <ul style="list-style-type: none"> • AWS.APG.CQRSES. CommandDeleteLambda • AWS.APG.CQRSES. CommandUpdateLambda • AWS.APG.CQRSES. CommandAddRewardLambda • AWS.APG.CQRSES. CommandRedeemRewardLambda • AWS.APG.CQRSES. QueryCustomerListLambda • AWS.APG.CQRSES. QueryRewardLambda 	앱 개발자, DevOps 엔지니어

Lambda 함수를 이벤트 리스너로 설정

작업	설명	필요한 기술
고객 및 보상 Lambda 이벤트 핸들러를 게시합니다.	<p>각 이벤트 핸들러를 게시하려면 이전 에픽의 단계를 따르십시오.</p> <p>프로젝트는 CQRS AWS Serverless\Event Source\Customer Event 및 CQRS AWS Serverless\Event Source\Reward Event 솔루션 폴더 아래에 있습니다. 자세한 내용은 추가</p>	앱 개발자

작업	설명	필요한 기술
	정보 섹션의 소스 코드 디렉토리를 참조하십시오.	

작업	설명	필요한 기술
이벤트 소싱 Lambda 이벤트 리스너를 연결합니다.	<ol style="list-style-type: none"> 1. Lambda 프로젝트를 게시할 때 사용한 것과 동일한 계정을 사용하여 AWS Management Console에 로그인합니다. 2. 리전의 경우 미국 동부 1 또는 이전 에픽에서 Lambda 함수를 배포한 리전을 선택합니다. 3. Lambda 서비스로 이동합니다. 4. EventSourceCustomer Lambda 함수를 선택합니다. 5. 트리거 추가를 선택합니다. 6. 트리거 구성 드롭다운 목록에서 DynamoDB를 선택합니다. 7. DynamoDB 테이블 드롭다운 목록에서 <code>cqrses-customer-cmd</code>를 선택합니다. 8. 시작 위치 드롭다운 목록에서 수평 트리밍을 선택합니다. 수평 트리밍은 DynamoDB 트리거가 샤드에서 가장 오래된 레코드인 마지막(트리밍되지 않은) 스트림 레코드에서 읽기를 시작하는 것을 의미합니다. 9. 트리거 활성화 확인란을 선택합니다. 	앱 개발자

작업	설명	필요한 기술
	<p>10나머지 필드의 경우 기본값을 유지하고 추가를 선택합니다.</p> <p>리스너가 DynamoDB 테이블에 성공적으로 연결되면 Lambda 디자이너 페이지에 리스너가 표시됩니다.</p>	
EventSourceReward Lambda 함수를 게시하고 연결합니다.	EventSourceReward Lambda 함수를 게시하고 연결하려면 DynamoDB 테이블 드롭다운 목록에서 선택하여 이전 두 스토리의 단계를 반복합니다. <code>cqrses-reward-cmd</code>	앱 개발자

DynamoDB 스트림과 Lambda 트리거를 테스트하고 검증합니다.

작업	설명	필요한 기술
스트림과 Lambda 트리거를 테스트합니다.	<ol style="list-style-type: none"> 1. Visual Studio에서 AWS 탐색기로 이동합니다. 2. AWS Lambda를 확장하고 함수를 선택합니다 (CommandRedeemReward 두 번 클릭). 함수 창이 열리면 함수를 테스트할 수 있습니다. 3. 요청 텍스트 상자에 요청 데이터를 JavaScript 객체 표기법 (JSON) 형식으로 입력합니다. 예제 요청은 추가 정보 	앱 개발자

작업	설명	필요한 기술
	<p>섹션의 테스트 데이터를 참조하십시오.</p> <p>4. Invoke를 선택합니다.</p>	
DynamoDB 보상 쿼리 테이블을 사용하여 검증합니다.	<ol style="list-style-type: none"> 테이블을 엽니다. <code>cqrse-reward-query</code> 리워드를 사용한 고객의 포인트를 확인하십시오. 사용한 포인트는 고객의 총 누적 포인트에서 차감해야 합니다. 	앱 개발자
CloudWatch 로그를 사용하여 검증하십시오.	<ol style="list-style-type: none"> 로그 그룹으로 CloudWatch 이동하여 선택합니다. <code>/aws/lambda/EventSourceReward</code> 로그 그룹에는 트리거에 대한 로그가 포함되어 있습니다. <code>EventSourceReward Lambda</code> 코드에서 <code>context.Logger.LogLine</code> 및 <code>Console.WriteLine</code>에 입력한 메시지를 포함하여 모든 Lambda 직접 호출이 기록됩니다. 	앱 개발자
트리거의 <code>EventSourceCustomer</code> 유효성을 검사합니다.	<p><code>EventSourceCustomer</code> 트리거를 검증하려면 트리거의 해당 고객 테이블과 CloudWatch 로그를 사용하여 이 에픽의 단계를 반복하세요.</p> <p><code>EventSourceCustomer</code></p>	앱 개발자

관련 리소스

참조

- [Visual Studio 2019 Community Edition 다운로드](#)
- [AWS Toolkit for Visual Studio 다운로드](#)
- [AWS Toolkit for Visual Studio 사용 설명서](#)
- [AWS의 서버리스](#)
- [DynamoDB 사용 사례 및 디자인 패턴](#)
- [마틴 파울러 CQRS](#)
- [마틴 파울러 이벤트 소싱](#)

비디오

- [AWS Toolkit for Visual Studio 데모](#)
- [새 IAM 사용자의 액세스 키 ID를 생성하려면 어떻게 해야 하나요?](#)

추가 정보

CQRS 및 이벤트 소싱

CQRS

CQRS 패턴은 데이터 액세스 객체 단일 CRUD(생성, 읽기, 업데이트, 삭제) 모델과 같은 단일 개념적 운영 모델을 명령 및 쿼리 운영 모델로 구분합니다. 명령 모델은 생성, 업데이트 또는 삭제와 같이 상태를 변경하는 모든 작업을 말합니다. 쿼리 모델은 값을 반환하는 모든 작업을 가리킵니다.

1. 고객 CRUD 모델에는 다음과 같은 인터페이스가 포함됩니다.

- CreateCustomer()
- UpdateCustomer()
- DeleteCustomer()
- AddPoints()
- RedeemPoints()
- GetVIPCustomers()

- `GetCustomerList()`
- `GetCustomerPoints()`

요구 사항이 더욱 복잡해지면 이 단일 모델 접근 방식을 벗어나도 됩니다. CQRS는 명령 모델과 쿼리 모델을 사용하여 데이터 쓰기 및 읽기 책임을 분리합니다. 이렇게 하면 데이터를 독립적으로 유지 관리하고 관리할 수 있습니다. 책임을 명확히 구분하면 각 모델의 개선이 다른 모델에는 영향을 미치지 않습니다. 이렇게 분리하면 유지 관리 및 성능이 향상되고 애플리케이션이 성장함에 따라 복잡성이 줄어듭니다.

1. 고객 명령 모델의 인터페이스.

- `Create Customer()`
- `UpdateCustomer()`
- `DeleteCustomer()`
- `AddPoints()`
- `RedeemPoints()`

2. 고객 쿼리 모델의 인터페이스.

- `GetVIPCustomers()`
- `GetCustomerList()`
- `GetCustomerPoints()`
- `GetMonthlyStatement()`

예제 코드는 소스 코드 디렉토리를 참조하십시오.

그런 다음 CQRS 패턴은 데이터베이스를 분리합니다. 이러한 디커플링은 각 서비스의 완전한 독립성으로 이어지며, 이는 마이크로서비스 아키텍처의 주요 요소입니다.

AWS 클라우드에서 CQRS를 사용하면 각 서비스를 추가로 최적화할 수 있습니다. 예를 들어, 다양한 컴퓨팅 설정을 지정하거나 서버리스 또는 컨테이너 기반 마이크로서비스 중에서 선택할 수 있습니다. 온프레미스 캐싱을 Amazon으로 대체할 수 있습니다. ElastiCache 온프레미스 게시/구독 메시징이 있는 경우 Amazon Simple Notification Service(SNS)로 대체할 수 있습니다. 또한 pay-as-you-go 요금은 물론 사용한 만큼만 지불하는 다양한 AWS 서비스를 이용할 수 있습니다.

CQRS에는 다음과 같은 이점이 있습니다.

- 독립적 규모 조정 - 각 모델은 서비스의 요구 사항 및 수요를 충족하도록 확장 전략을 조정할 수 있습니다. 고성능 애플리케이션과 마찬가지로 읽기와 쓰기를 분리하면 모델을 독립적으로 규모를 조정하여 각 요구 사항을 충족할 수 있습니다. 또한 컴퓨팅 리소스를 추가하거나 줄여 다른 모델에 영향을 주지 않으면서 한 모델의 확장성 요구를 해결할 수 있습니다.
- 독립적 유지 관리 - 쿼리와 명령 모델을 분리하면 모델의 유지 관리 용이성이 향상됩니다. 다른 모델에는 영향을 주지 않으면서 한 모델의 코드를 변경하고 개선할 수 있습니다.
- 보안 - 읽기 및 쓰기를 위해 별도의 모델에 권한과 정책을 적용하는 것이 더 쉽습니다.
- 읽기 최적화 - 쿼리에 최적화된 스키마를 정의할 수 있습니다. 예를 들어, 집계된 데이터에 대한 스키마를 정의하고 팩트 테이블에 대해 별도의 스키마를 정의할 수 있습니다.
- 통합 - CQRS는 이벤트 기반 프로그래밍 모델에 적합합니다.
- 복잡성 관리 - 쿼리와 명령 모델로의 분리는 복잡한 도메인에 적합합니다.

CQRS를 사용할 때는 다음 주의 사항을 명심하십시오.

- CQRS 패턴은 애플리케이션의 특정 부분에만 적용되며 전체 애플리케이션에는 적용되지 않습니다. 패턴에 맞지 않는 도메인에 구현하면 생산성이 저하되고 위험이 증가하며 복잡성이 생길 수 있습니다.
- 이 패턴은 읽기 및 쓰기 작업이 불균형하고 자주 사용되는 모델에 가장 적합합니다.
- 처리하는 데 시간이 걸리는 대용량 보고서와 같이 읽기 중심의 애플리케이션의 경우 CQRS는 적절한 데이터베이스를 선택하고 집계된 데이터를 저장할 스키마를 생성할 수 있는 옵션을 제공합니다. 이렇게 하면 보고서 데이터를 한 번만 처리하고 집계된 테이블에 덤프하므로 보고서를 읽고 보는 응답 시간이 향상됩니다.
- 쓰기가 많은 애플리케이션의 경우 쓰기 작업을 위해 데이터베이스를 구성하고 쓰기 수요가 증가할 때 명령 마이크로서비스가 독립적으로 규모가 조정되도록 할 수 있습니다. 예제는 `AWS.APG.CQRSES.CommandRedeemRewardLambda` 및 `AWS.APG.CQRSES.CommandAddRewardLambda` 마이크로서비스를 참조하십시오.

이벤트 소싱

다음 단계는 명령 실행 시 이벤트 소싱을 사용하여 쿼리 데이터베이스를 동기화하는 것입니다. 예를 들어 다음 이벤트를 고려합니다.

- 고객 보상 포인트가 추가되어 쿼리 데이터베이스의 고객 총 보상 포인트 또는 집계된 보상 포인트를 업데이트해야 합니다.

- 명령 데이터베이스에서 고객의 성이 업데이트되므로 쿼리 데이터베이스의 대리 고객 정보를 업데이트해야 합니다.

기존 CRUD 모델에서는 트랜잭션이 완료될 때까지 데이터를 잠가서 데이터의 일관성을 보장합니다. 이벤트 소싱에서는 구독자가 해당 데이터를 업데이트하는 데 사용하는 일련의 이벤트 게시를 통해 데이터가 동기화됩니다.

이벤트 소싱 패턴은 데이터에 대해 취해진 일련의 전체 조치를 보장 및 기록하고 일련의 이벤트를 통해 데이터를 게시합니다. 이러한 이벤트는 해당 이벤트의 구독자가 기록을 최신 상태로 유지하기 위해 처리해야 하는 일련의 데이터 변경 사항을 나타냅니다. 이러한 이벤트는 구독자가 소비하여 구독자 데이터베이스의 데이터를 동기화합니다. 이 경우에는 쿼리 데이터베이스가 바로 그것입니다.

다음 다이어그램은 AWS 기반 CQRS에서 사용되는 이벤트 소싱을 보여줍니다.

1. 명령 Lambda 함수는 데이터베이스에서 생성, 업데이트 또는 삭제와 같은 쓰기 작업을 수행합니다.
2. 쿼리 Lambda 함수는 데이터베이스에서 가져오기 또는 선택과 같은 읽기 작업을 수행합니다.
3. 이 Lambda 함수는 명령 데이터베이스의 DynamoDB 스트림을 처리하고 변경 사항에 맞게 쿼리 데이터베이스를 업데이트합니다. 또한 이 함수를 사용하여 Amazon SNS에 메시지를 게시하여 구독자가 데이터를 처리할 수 있도록 할 수도 있습니다.
4. (선택 사항) Lambda 이벤트 구독자는 Amazon SNS에서 게시한 메시지를 처리하고 쿼리 데이터베이스를 업데이트합니다.
5. (선택 사항) Amazon SNS는 쓰기 작업에 대한 이메일 알림을 보냅니다.

AWS에서는 DynamoDB 스트림으로 쿼리 데이터베이스를 동기화할 수 있습니다. DynamoDB는 DynamoDB 테이블에서 시간 순서에 따라 항목 수준 수정을 거의 실시간으로 캡처하고 24시간 내에 정보를 안정적으로 저장합니다.

DynamoDB 스트림을 활성화하면 데이터베이스에서 이벤트 소싱 패턴을 가능하게 하는 일련의 이벤트를 게시할 수 있습니다. 이벤트 소싱 패턴은 이벤트 구독자를 추가합니다. 이벤트 구독자 애플리케이션은 이벤트를 소비하고 구독자의 책임에 따라 이벤트를 처리합니다. 이전 다이어그램에서 이벤트 구독자는 변경 내용을 쿼리 DynamoDB 데이터베이스에 푸시하여 데이터를 동기화된 상태로 유지합니다. Amazon SNS, 메시지 브로커 및 이벤트 구독자 애플리케이션을 사용하면 아키텍처가 분리된 상태로 유지됩니다.

이벤트 소싱에는 다음과 같은 이점이 있습니다.

- 트랜잭션 데이터의 일관성
- 데이터에서 취해진 조치를 모니터링하는 데 사용 가능한 신뢰할 수 있는 감사 추적 및 작업 기록
- 마이크로서비스와 같은 분산 애플리케이션에 환경 전반에서 걸쳐 데이터 동기화 허용
- 상태가 변경될 때마다 신뢰할 수 있는 이벤트 게시
- 과거 상태 재구성 또는 재생
- 모놀리식 애플리케이션에서 마이크로서비스로의 마이그레이션을 위해 이벤트를 교환하는 느슨하게 결합된 엔터티
- 동시 업데이트로 인한 충돌 감소, 이벤트 소싱 사용으로 데이터 스토어에서 객체 직접 업데이트 불필요
- 작업과 이벤트를 분리하는 데 따른 유연성 및 확장성
- 외부 시스템 업데이트
- 단일 이벤트에서 여러 작업 관리

이벤트 소싱을 사용할 때는 다음 주의 사항을 명심하십시오.

- 소스 구독자 데이터베이스 간에 데이터를 업데이트하는 데 약간의 지연이 있기 때문에 변경을 취소할 수 있는 유일한 방법은 이벤트 저장소에 보상 이벤트를 추가하는 것입니다.
- 이벤트 소싱은 프로그래밍 스타일이 다르기 때문에 구현을 익히는 데 시간이 많이 걸립니다.

테스트 데이터

배포 성공 후 다음 테스트 데이터를 사용하여 Lambda 함수를 테스트하십시오.

CommandCreate 고객

```
{ "Id":1501, "Firstname":"John", "Lastname":"Done", "CompanyName":"AnyCompany",
  "Address": "USA", "VIP":true }
```

CommandUpdate 고객

```
{ "Id":1501, "Firstname":"John", "Lastname":"Doe", "CompanyName":"Example Corp.",
  "Address": "Seattle, USA", "VIP":true }
```

CommandDelete 고객

고객 ID를 요청 데이터로 입력합니다. 예를 들어 고객 ID가 151인 경우 요청 데이터로 151을 입력합니다.

```
151
```

QueryCustomerList

이것은 비어 있습니다. 간접 호출되면 모든 고객이 반환됩니다.

CommandAddReward

그러면 ID 1(리처드)인 고객에게 40포인트가 추가됩니다.

```
{
  "Id":10101,
  "CustomerId":1,
  "Points":40
}
```

CommandRedeemReward

그러면 ID 1(리처드)인 고객에게 15포인트가 차감됩니다.

```
{
  "Id":10110,
  "CustomerId":1,
  "Points":15
}
```

QueryReward

고객 ID를 입력합니다. 예를 들어 리처드의 경우 1, Arnav의 경우 2, 설리의 경우 3을 입력합니다.

```
2
```

소스 코드 디렉터리

다음 테이블을 Visual Studio 솔루션의 디렉터리 구조에 대한 지침으로 사용하십시오.

CQRS 온프레미스 코드 샘플 솔루션 디렉터리

고객 CRUD 모델

CQRS On-Premises Code Sample\CRUD Model\AWS.APG.CQRSES.DAL 프로젝트

고객 CRUD 모델의 CQRS 버전

- 고객 명령: CQRS On-Premises Code Sample\CQRS Model\Command Microservice \AWS.APG.CQRSES.Command 프로젝트
- 고객 쿼리: CQRS On-Premises Code Sample\CQRS Model\Query Microservice \AWS.APG.CQRSES.Query 프로젝트

명령 및 쿼리 마이크로서비스

Command 마이크로서비스는 솔루션 폴더 CQRS On-Premises Code Sample\CQRS Model \Command Microservice 아래에 있습니다.

- AWS.APG.CQRSES.CommandMicroservice ASP.NET Core API 프로젝트는 소비자가 서비스와 상호 작용하는 진입점 역할을 합니다.
- AWS.APG.CQRSES.Command .NET Core 프로젝트는 명령 관련 객체 및 인터페이스를 호스팅하는 객체입니다.

쿼리 마이크로서비스는 솔루션 폴더 CQRS On-Premises Code Sample\CQRS Model\Query Microservice 아래에 있습니다.

- AWS.APG.CQRSES.QueryMicroservice ASP.NET Core API 프로젝트는 소비자가 서비스와 상호 작용하는 진입점 역할을 합니다.
- AWS.APG.CQRSES.Query .NET Core 프로젝트는 쿼리 관련 객체 및 인터페이스를 호스팅하는 객체입니다.

CQRS AWS 서버리스 코드 솔루션 디렉터리

이 코드는 AWS 서버리스 서비스를 사용하는 온프레미스 코드의 AWS 버전입니다.

C# .NET Core에서 각 Lambda 함수는 하나의 .NET 코어 프로젝트로 표현됩니다. 이 패턴의 예제 코드에는 명령 및 쿼리 모델의 각 인터페이스에 대해 별도의 프로젝트가 있습니다.

AWS services를 사용한 CQRS

AWS 서버리스 서비스를 사용하는 CQRS용 루트 솔루션 디렉터리는 CQRS AWS Serverless\CQRS 폴더에서 찾을 수 있습니다. 이 예제에는 고객과 보상이라는 두 가지 모델이 포함됩니다.

고객 및 보상에 대한 명령 Lambda 함수는 CQRS\Command Microservice\Customer 및 CQRS\Command Microservice\Reward 폴더 아래에 있습니다. 여기에는 다음과 같은 Lambda 프로젝트가 포함됩니다.

- 고객 명령: CommandCreateLambda, CommandDeleteLambda 및 CommandUpdateLambda
- 보상 명령: CommandAddRewardLambda 및 CommandRedeemRewardLambda

고객 및 보상에 대한 쿼리 Lambda 함수는 CQRS\Query Microservice\Customer 및 CQRS\QueryMicroservice\Reward 폴더 아래에 있습니다. 여기에는 QueryCustomerListLambda 및 QueryRewardLambda Lambda 프로젝트가 포함됩니다.

CQRS 테스트 프로젝트

테스트 프로젝트는 CQRS\Tests 폴더 아래에 있습니다. 이 프로젝트에는 CQRS Lambda 함수 테스트를 자동화하는 테스트 스크립트가 포함되어 있습니다.

AWS 서비스를 사용한 이벤트 소싱

다음 Lambda 이벤트 핸들러는 고객 및 보상 DynamoDB 스트림에 의해 시작되어 쿼리 테이블의 데이터를 처리하고 동기화합니다.

- EventSourceCustomer Lambda 함수는 고객 테이블(cqrses-customer-cmd) DynamoDB 스트림에 매핑됩니다.
- EventSourceReward Lambda 함수는 보상 테이블(cqrses-reward-cmd) DynamoDB 스트림에 매핑됩니다.

첨부

이 문서와 관련된 추가 콘텐츠에 액세스하려면 [attachment.zip](#) 파일의 압축을 풉니다.

패턴 더 보기

- [PrivateLink AWS와 Network Load Balancer를 사용하여 Amazon EKS에서 비공개로 컨테이너 애플리케이션에 액세스합니다.](#)
- [AWS Systems Manager를 사용하여 Windows 레지스트리 항목의 추가 또는 업데이트 자동화](#)
- [를 사용하여 마이그레이션 전략 식별 및 계획을 자동화합니다. AppScore](#)
- [CI/CD 파이프라인을 사용하여 Amazon EKS에 Java 애플리케이션 자동 구축 및 배포](#)
- [AWS CDK를 사용하여 마이크로서비스용 CI/CD 파이프라인 및 Amazon ECS 클러스터 자동으로 구축](#)
- [BMC AMI 클라우드 데이터를 사용하여 메인프레임 데이터를 Amazon S3에 백업하고 보관합니다.](#)
- [서버리스 접근 방식을 사용하여 AWS 서비스를 함께 연결](#)
- [Blu Age로 현대화된 메인프레임 워크로드 컨테이너화](#)
- [AWS 리포지토리에서 최신 AWS Amplify 웹 애플리케이션을 지속적으로 배포하십시오. CodeCommit](#)
- [Python을 사용하여 AWS에서 EBCDIC 데이터를 ASCII로 변환 및 압축 해제](#)
- [Micro Focus를 사용하여 복잡한 레코드 레이아웃이 있는 메인프레임 데이터 파일 변환](#)
- [AWS CLI를 사용하여 S3 버킷에서 다른 계정 및 리전으로 데이터 복사](#)
- [HashiCorp 패커를 사용하여 CodePipeline 파이프라인 및 AMI 생성](#)
- [파이프라인을 생성하고 다음을 사용하여 온프레미스 EC2 인스턴스에 아티팩트 업데이트를 배포합니다. CodePipeline](#)
- [Amazon EKS 클러스터의 배포 및 디버깅](#)
- [Elastic Beanstalk를 사용하여 컨테이너 배포](#)
- [PostgreSQL-compatible Aurora 글로벌 데이터베이스를 사용하여 Oracle DR 에뮬레이션하기](#)
- [Oracle SQL Developer 및 AWS SCT를 사용하여 Amazon RDS for Oracle에서 Amazon RDS for PostgreSQL로 점진적으로 마이그레이션](#)
- [Stonebranch 유니버설 컨트롤러를 AWS Mainframe Modernization과 통합](#)
- [여러 AWS 계정 및 AWS 리전의 AWS Service Catalog 제품을 관리](#)
- [AWS Organizations의 AWS 멤버 계정을 AWS Control Tower로 마이그레이션](#)
- [Connect from Precisely를 사용하여 VSAM 파일을 Amazon RDS 또는 Amazon MSK로 마이그레이션하고 복제하기](#)
- [AWS DMS를 사용하여 SAP ASE에서 Amazon RDS for SQL Server로 마이그레이션](#)
- [Oracle 외부 테이블을 Amazon Aurora PostgreSQL 호환으로 마이그레이션](#)

- [Micro Focus Enterprise Server 및 LRS VPSX/MFI를 사용하여 AWS에서 메인프레임 배치 인쇄 워크로드를 현대화](#)
- [Micro Focus Enterprise Server 및 LRS VPSX/MFI를 사용하여 AWS에서 메인프레임 온라인 인쇄 워크로드를 현대화](#)
- [OpenText 마이크로 포커스 엔터프라이즈 서버 및 LRS X를 사용하여 AWS의 메인프레임 출력 관리를 현대화합니다. PageCenter](#)
- [Transfer Family를 사용하여 메인프레임 파일을 Amazon S3로 직접 이동](#)
- [AWS App2Container 생성 도커 이미지 최적화](#)
- [Precisely Connect를 사용하여 메인프레임 데이터베이스를 AWS에 복제하기](#)
- [Amazon ECS Anywhere를 WorkSpaces 사용하여 Amazon에서 Amazon ECS 작업을 실행합니다.](#)
- [Amazon S3에서 Helm v3 차트 리포지토리 설정](#)
- [다중 지역, 다중 계정 조직에서 AWS CloudFormation 드리프트 감지 설정](#)
- [AWS Lambda를 사용하여 육각형 아키텍처로 Python 프로젝트 구조화](#)
- [SAP Pacemaker 클러스터를 ENSA1에서 ENSA2 클러스터로 업그레이드](#)
- [온프레미스 데이터베이스의 재해 CloudEndure 복구에 사용](#)
- [Account Factory에 대한 테라폼\(AFT\) 코드를 로컬에서 검증합니다.](#)

네트워킹

주제

- [AWS Transit Gateway를 사용하여 리전 간 피어링 설정을 자동화하십시오.](#)
- [AWS Transit Gateway를 사용하여 네트워크 연결 중앙 집중화](#)
- [Application Load Balancer를 사용하여 Oracle JD Edwards EnterpriseOne on WebLogic Oracle에 대한 HTTPS 암호화를 구성합니다.](#)
- [프라이빗 네트워크를 통해 Application Migration Service 데이터 및 컨트롤 플레인에 연결](#)
- [AWS CloudFormation 사용자 지정 리소스와 Amazon SNS를 사용하여 인포블록 객체를 생성합니다.](#)
- [AWS 네트워크 방화벽에 대한 Amazon CloudWatch 알림을 사용자 지정합니다.](#)
- [DNS 레코드를 Amazon Route 53 프라이빗 호스팅 영역으로 대량 마이그레이션합니다.](#)
- [F5에서 AWS의 Application Load Balancer로 마이그레이션할 때 HTTP 헤더를 수정](#)
- [다중 VPC에서 중앙 AWS 서비스 엔드포인트에 비공개로 액세스](#)
- [여러 AWS 계정에서 인바운드 인터넷 액세스에 대한 Network Access Analyzer 조사 결과 보고서 생성](#)
- [AWS Organizations를 사용하여 Transit Gateway Attachment에 자동으로 태그 지정](#)
- [ELB 로드 밸런서에 TLS 터미널이 필요한지 검증합니다](#)
- [Splunk를 사용하여 AWS 네트워크 방화벽 로그 및 지표 보기](#)
- [더 많은 패턴](#)

AWS Transit Gateway를 사용하여 리전 간 피어링 설정을 자동화하십시오.

작성자: Ram Kandaswamy(AWS)

환경: 프로덕션

기술: 네트워킹, 하이브리드 클라우드

AWS 서비스: AWS Transit Gateway, AWS Step Functions, AWS Lambda

요약

AWS Transit Gateway는 중앙 허브를 통해 Virtual Private Cloud(VPC)와 온프레미스 네트워크를 연결합니다. Transit Gateway 트래픽은 항상 글로벌 Amazon Web Services(AWS) 백본에 머물며 공용 인터넷을 통과하지 않으므로 일반적인 악용 및 분산 서비스 거부 (DDoS) 공격과 같은 위협 벡터가 줄어듭니다.

두 개 이상의 AWS 리전 간에 통신해야 하는 경우, 리전 간 Transit Gateway 피어링을 사용하여 서로 다른 리전의 전송 게이트웨이 간에 피어링 연결을 설정할 수 있습니다. 하지만 Transit Gateway를 사용하여 리전 간 피어링을 수동으로 구성하려면 여러 단계를 거쳐야 하므로 시간이 많이 걸릴 수 있습니다. 이 패턴은 코드를 사용하여 피어링을 수행함으로써 이러한 수동 단계를 제거하는 자동화된 프로세스를 제공합니다. 다중 리전 조직 설정 중에 여러 리전 및 AWS 계정을 반복적으로 구성해야 하는 경우 이 방법을 사용할 수 있습니다.

이 패턴은 AWS CloudFormation Step Functions 워크플로, AWS Lambda 함수, AWS Identity 및 Access Management (IAM) 역할, Amazon Logs의 로그 그룹을 포함하는 AWS 스택을 사용합니다. CloudWatch 그런 다음 Step Functions 실행을 시작하고 트랜짓 게이트웨이를 위한 리전 간 피어링 연결을 생성할 수 있습니다.

사전 조건 및 제한 사항

사전 조건

- 활성 상태의 AWS 계정.
- 기존 Amazon Simple Storage Service(S3) 버킷.

- 요청자 리전 및 수락자 리전에서 생성 및 구성된 트랜짓 게이트웨이. 요청자 리전은 피어링 요청이 시작되는 곳이고 수락자 영역은 피어링 요청을 수락합니다. 이에 대한 자세한 내용은 Amazon VPC 설명서의 [VPC 피어링 연결 생성 및 수락](#)을 참조하십시오.
- 수락자 및 요청자 리전에 설치 및 구성된 VPC. VPC를 생성하는 단계는 Amazon VPC 설명서의 [Amazon VPC 시작하기에서 VPC 생성](#)을 참조하십시오.
- VPC는 태그와 값을 사용해야 합니다. `addToTransitGateway true`
- VPC의 보안 그룹 및 네트워크 액세스 제어 목록(ACL)이 요구 사항에 따라 구성됩니다. 이에 대한 자세한 내용은 Amazon [VPC 설명서에서 VPC의 보안 그룹 및 네트워크 ACL](#)을 참조하십시오.

AWS 리전 및 제한

- 리전 내 피어링은 특정 AWS 리전에서만 지원합니다. 리전 간 피어링을 지원하는 리전의 전체 목록은 [AWS Transit Gateway FAQ](#)를 참조하십시오.
- 첨부된 샘플 코드에서는 요청자 리전을 가정하고 수락자 리전으로 가정했습니다. `us-east-2 us-west-2` 다른 리전을 구성하려면 모든 Python 파일에서 이러한 값을 편집해야 합니다. 두 개 이상의 리전을 포함하는 더 복잡한 설정을 구현하려면 리전을 Lambda 함수에 파라미터로 전달하도록 Step 함수를 변경하고 각 조합에 대해 함수를 실행할 수 있습니다.

아키텍처

이 다이어그램은 다음 단계의 워크플로를 보여줍니다.

1. 사용자가 AWS CloudFormation 스택을 생성합니다.
2. AWS는 Lambda 함수를 사용하는 Step Functions 상태 머신을 CloudFormation 생성합니다. 이에 대한 자세한 내용은 AWS [Step Functions 설명서에서 Lambda를 사용하는 Step Functions 상태 머신 생성](#)을 참조하십시오.
3. Step Functions는 피어링을 위해 Lambda 함수를 직접 호출합니다.
4. Lambda 함수는 Transit Gateway 간에 피어링 연결을 생성합니다.
5. Step Functions는 라우팅 테이블 수정을 위해 Lambda 함수를 직접 호출합니다.
6. Lambda 함수는 VPC의 Classless Inter-Domain Routing (CIDR) 블록을 추가하여 라우팅 테이블을 수정합니다.

Step Functions 워크플로우

이 다이어그램은 다음 단계 함수 워크플로를 보여줍니다.

1. Step Functions 워크플로는 Transit Gateway 피어링에 대해 Lambda 함수를 직접 호출합니다.
2. 1분 동안 대기해야 하는 타이머 직접 호출이 있습니다.
3. 피어링 상태가 검색되어 조건 블록으로 전송됩니다. 블록은 루핑을 담당합니다.
4. 성공 조건이 충족되지 않으면 워크플로가 타이머 단계에 들어가도록 코딩됩니다.
5. 성공 조건이 충족되면 Lambda 함수가 직접 호출되어 라우팅 테이블을 수정합니다. 이 직접 호출이 끝나면 Step Functions 워크플로가 종료됩니다.

도구

- [AWS CloudFormation](#) — CloudFormation AWS는 AWS 리소스를 모델링하고 설정하는 데 도움이 되는 서비스입니다.
- [Amazon CloudWatch CloudWatch Logs](#) — Logs는 사용하는 모든 시스템, 애플리케이션 및 AWS 서비스의 로그를 중앙 집중화하는 데 도움이 됩니다.
- [Identity and Access Management\(IAM\)](#) – IAM은 AWS 리소스에 대한 사용자의 액세스를 안전하게 제어할 수 있게 지원하는 웹 서비스입니다.
- [AWS Lambda](#) — Lambda는 고가용성 컴퓨팅 인프라에서 코드를 실행하고 모든 컴퓨팅 리소스 관리를 수행합니다.
- [AWS Step Functions](#) – Step Functions를 사용하면 시각적 워크플로우에서 분산 애플리케이션의 구성 요소를 일련의 단계로 쉽게 조정할 수 있습니다.

에픽

피어링 자동화

작업	설명	필요한 기술
첨부 파일을 S3 버킷에 업로드합니다.	AWS Management Console에 로그인하고 Amazon S3 콘솔을 연 다음 modify-tr	일반 AWS

작업	설명	필요한 기술
	ansit-gateway-routes.zip , peer-transit-gateway.zip , get-transit-gateway-peering-status.zip 파일(첨부)을 S3 버킷에 업로드합니다.	

작업	설명	필요한 기술
<p>AWS CloudFormation 스택을 생성합니다.</p>	<p>다음 명령을 실행하여 transit-gateway-peering.json 파일 (첨부) CloudFormation 을 사용하여 AWS 스택을 생성합니다.</p> <pre>aws cloudformation create-stack --stack-name myteststack --template-body file://sampletemplate.json</pre> <p>AWS CloudFormation 스택은 Step Functions 워크플로, Lambda 함수, IAM 역할 및 로그 그룹을 생성합니다. CloudWatch</p> <p>AWS CloudFormation 템플릿이 이전에 업로드한 파일이 포함된 S3 버킷을 참조하는지 확인하십시오.</p> <p>참고: AWS CloudFormation 콘솔을 사용하여 스택을 생성할 수도 있습니다. 이에 대한 자세한 내용은 AWS CloudFormation 설명서의 AWS CloudFormation 콘솔에 스택 생성을 참조하십시오.</p>	<p>DevOps 엔지니어</p>

작업	설명	필요한 기술
<p>Step Functions에서 새 실행을 시작합니다.</p>	<p>Step Functions 콘솔을 열고 새 실행을 시작합니다.</p> <p>Step Functions는 Lambda 함수를 직접 호출하고 Transit Gateway를 위한 피어링 연결을 생성합니다. 입력 JSON 파일은 필요하지 않습니다. 첨부 파일을 사용할 수 있고 연결 유형이 피어링인지 확인하십시오.</p> <p>이에 대한 자세한 내용은 AWS Steps Functions 설명서의 AWS Step Functions 시작하기에서 새 실행 시작을 참조하십시오.</p>	<p>DevOps 엔지니어, 일반 AWS</p>
<p>라우팅 테이블의 경로를 확인합니다.</p>	<p>Transit Gateway 간에 리전 간 피어링이 설정됩니다. 라우팅 테이블은 피어 리전 VPC의 IPv4 CIDR 블록 범위로 업데이트됩니다.</p> <p>Amazon VPC 콘솔을 열고 라우팅 테이블에서 Transit Gateway Attachment에 해당하는 Associations 탭을 선택합니다. 피어링된 리전의 VPC CIDR 블록 범위를 확인합니다.</p> <p>자세한 단계 및 지침은 Amazon VPC 설명서의 Transit Gateway 라우팅 테이블 연결을 참조하십시오.</p>	<p>네트워크 관리자</p>

관련 리소스

- [Step Functions에서의 실행](#)
- [Transit Gateway 피어링 연결](#)
- [AWS Transit Gateway를 사용하여 여러 AWS 리전의 VPC를 상호 연결 - 데모 \(비디오\)](#)

첨부

이 문서와 관련된 추가 콘텐츠에 액세스하려면 [attachment.zip](#) 파일의 압축을 풉니다.

AWS Transit Gateway를 사용하여 네트워크 연결 중앙 집중화

작성자: 마이딜리 팔라구미(AWS)와 니킴 마라푸(AWS)

환경: 프로덕션

기술: 네트워킹

AWS 서비스: AWS Transit Gateway; Amazon VPC

요약

이 패턴은 AWS Transit Gateway를 사용하여 AWS 리전 내 여러 AWS 계정의 Virtual Private Cloud(VPC)에 온프레미스 네트워크를 연결할 수 있는 가장 간단한 구성을 설명합니다. 이 설정을 사용하면 한 리전의 여러 VPC 네트워크와 온프레미스 네트워크를 연결하는 하이브리드 네트워크를 구축할 수 있습니다. 이는 전송 게이트웨이와 온프레미스 네트워크에 대한 가상 프라이빗 네트워크(VPN) 연결을 사용하여 수행됩니다.

사전 조건 및 제한 사항

사전 조건

- AWS Organizations에서 조직의 회원 계정으로 관리되는 네트워크 서비스를 호스트하기 위한 계정
- Classless Inter-Domain Routing(CIDR) 블록이 겹치지 않는 여러 AWS 계정의 VPC

제한 사항

이 패턴은 특정 VPC 또는 온프레미스 네트워크 간의 트래픽 격리를 지원하지 않습니다. 전송 게이트웨이에 연결된 모든 네트워크는 서로 연결할 수 있습니다. 트래픽을 격리하려면 전송 게이트웨이에서 사용자 지정 라우팅 테이블을 사용해야 합니다. 이 패턴은 가장 간단한 구성인 단일 기본 전송 게이트웨이 라우팅 테이블을 사용하여 VPC와 온프레미스 네트워크만 연결합니다.

아키텍처

대상 기술 스택

- AWS Transit Gateway
- AWS Site-to-Site VPN

- VPC
- AWS Resource Access Manager(AWS RAM)

대상 아키텍처

도구

서비스

- [AWS Resource Access Manager\(AWS RAM\)](#)를 사용하면 AWS 계정, 조직 구성 단위 또는 AWS Organizations의 전체 조직에서 리소스를 안전하게 공유할 수 있습니다.
- [AWS Transit Gateway](#)는 Virtual Private Cloud(VPC)와 온프레미스 네트워크를 연결하는 중앙 허브입니다.

에픽

네트워크 서비스 계정에서 전송 게이트웨이 생성

작업	설명	필요한 기술
전송 게이트웨이를 생성합니다.	<p>네트워크 서비스를 호스트하려는 AWS 계정에서 대상 AWS 리전에 전송 게이트웨이를 생성합니다. 지침은 전송 게이트웨이 생성을 참조하세요. 다음을 참고합니다.</p> <ul style="list-style-type: none"> • 기본 라우팅 테이블 연결을 선택합니다. • 기본 라우팅 테이블 전파를 선택합니다. 	네트워크 관리자

전송 게이트웨이를 온프레미스 네트워크에 연결

작업	설명	필요한 기술
VPN 연결의 고객 게이트웨이 디바이스를 설정합니다.	고객 게이트웨이 디바이스는 전송 게이트웨이와 온프레미스 네트워크 간 Site-to-Site VPN 연결의 온프레미스 측에 연결됩니다. 자세한 내용은 AWS Site-to-Site VPN 설명서의 고객 게이트웨이 디바이스 를 참조하세요. 지원되는 온프레미스 고객 디바이스를 식별하거나 시작하고 해당 공용 IP 주소를 기록해 둡니다. VPN 구성은 이 에픽의 후반부에서 완료됩니다.	네트워크 관리자
네트워크 서비스 계정에서 전송 게이트웨이에 대한 VPN 연결을 생성합니다.	연결을 설정하려면 전송 게이트웨이에 대한 VPN 연결을 생성합니다. 지침은 전송 게이트웨이 VPN 연결 을 참조하세요.	네트워크 관리자
온프레미스 네트워크의 고객 게이트웨이 디바이스에서 VPN을 구성합니다.	전송 게이트웨이와 연결된 Site-to-Site VPN 연결의 구성 파일을 다운로드하고 고객 게이트웨이 디바이스에서 VPN 설정을 구성합니다. 지침은 구성 파일 다운로드 를 참조하세요.	네트워크 관리자

네트워크 서비스 계정의 전송 게이트웨이를 다른 AWS 계정 또는 조직과 공유

작업	설명	필요한 기술
AWS Organizations 관리 계정에서 공유를 활성화합니다.	전송 게이트웨이를 조직 또는 특정 조직 단위와 공유하려면	AWS 시스템 관리자

작업	설명	필요한 기술
	<p>AWS Organizations에서 공유를 켭니다. 그렇지 않으면 각 계정에서 전송 게이트웨이를 개별적으로 공유해야 합니다. 지침은 AWS Organizations 내 리소스 공유 활성화를 참조하세요.</p>	
<p>네트워크 서비스 계정에서 전송 게이트웨이 리소스 공유를 생성합니다.</p>	<p>조직 내 다른 AWS 계정의 VPC가 전송 게이트웨이에 연결되도록 하려면 네트워크 서비스 계정에서 AWS RAM 콘솔을 사용하여 전송 게이트웨이 리소스를 공유합니다. 지침은 리소스 공유 생성을 참조하세요.</p>	<p>AWS 시스템 관리자</p>

VPC를 전송 게이트웨이에 연결

작업	설명	필요한 기술
<p>개별 계정에서 VPC 연결을 생성합니다.</p>	<p>전송 게이트웨이가 공유된 계정에서 전송 게이트웨이 VPC 연결을 생성합니다. 지침은 VPC에 대한 Transit Gateway Attachment 생성을 참조하세요.</p>	<p>네트워크 관리자</p>
<p>VPC 연결 요청을 수락합니다.</p>	<p>네트워크 서비스 계정에서 전송 게이트웨이 VPC 연결 요청을 수락합니다. 지침은 공유 연결 수락을 참조하세요.</p>	<p>네트워크 관리자</p>

라우팅 구성

작업	설명	필요한 기술
<p>개별 계정 VPC에서 경로를 구성합니다.</p>	<p>각 개별 계정 VPC에서 전송 게이트웨이를 대상으로 사용하여 온프레미스 네트워크 및 다른 VPC 네트워크에 대한 경로를 추가합니다. 지침은 라우팅 테이블에서 경로 추가 및 제거를 참조하세요.</p>	<p>네트워크 관리자</p>
<p>전송 게이트웨이 라우팅 테이블의 경로를 구성합니다.</p>	<p>VPC의 경로와 VPN 연결은 전파되어야 하며 전송 게이트웨이 기본 라우팅 테이블에 표시되어야 합니다. 필요한 경우 전송 게이트웨이 기본 라우팅 테이블에 정적 경로(한 예로 정적 VPN 연결을 위한 정적 경로)를 생성합니다. 지침은 정적 경로 생성을 참조하세요.</p>	<p>네트워크 관리자</p>
<p>보안 그룹 및 네트워크 액세스 제어 목록(ACL) 규칙을 추가합니다.</p>	<p>VPC의 EC2 인스턴스 및 기타 리소스의 경우, 보안 그룹 규칙과 네트워크 ACL 규칙이 VPC와 온프레미스 네트워크 간의 트래픽을 허용하는지 확인합니다. 지침은 보안 그룹을 사용하여 리소스에 대한 트래픽 제어 및 ACL에서 규칙 추가 및 삭제를 참조하세요.</p>	<p>네트워크 관리자</p>

연결 테스트

작업	설명	필요한 기술
VPC 간 연결성을 테스트합니다.	네트워크 ACL 및 보안 그룹이 인터넷 제어 메시지 프로토콜 (ICMP) 트래픽을 허용하는지 확인한 다음 VPC의 인스턴스에서 전송 게이트웨이에 연결된 다른 VPC로 핑을 실행합니다.	네트워크 관리자
VPC와 온프레미스 네트워크 간의 연결을 테스트합니다.	네트워크 ACL 규칙, 보안 그룹 규칙, 방화벽이 ICMP 트래픽을 허용하는지 확인한 다음 온프레미스 네트워크와 VPC의 EC2 인스턴스 간에 ping을 실행합니다. VPN 연결을 UP 상태로 유지하려면 먼저 온프레미스 네트워크에서 네트워크 통신을 시작해야 합니다.	네트워크 관리자

관련 리소스

- [확장 가능하고 안전한 다중 VPC AWS 네트워크 인프라 구축](#) (AWS 백서)
- [공유 리소스로 작업하기](#) (AWS RAM 설명서)
- [트랜짓 게이트웨이 사용](#) (AWS Transit Gateway 설명서)

Application Load Balancer를 사용하여 Oracle JD Edwards EnterpriseOne on WebLogic Oracle에 대한 HTTPS 암호화를 구성합니다.

작성자: Thanigaivel Thirumalai(AWS)

환경: 프로덕션

기술: 네트워킹; 보안, 자격 증명, 규정 준수

워크로드: Oracle

AWS 서비스: AWS Certificate Manager(ACM); Elastic Load Balancing(ELB); Amazon Route 53

요약

이 패턴은 오라클 워크로드의 Oracle JD Edwards에서 SSL 오프로딩을 위한 HTTPS 암호화를 구성하는 방법을 설명합니다. EnterpriseOne WebLogic 이 접근 방식은 사용자 브라우저와 로드 밸런서 간의 트래픽을 암호화하여 서버의 암호화 부담을 제거합니다. EnterpriseOne

많은 사용자가 [AWS Application Load Balancer](#)를 사용하여 JVM (EnterpriseOne Java 가상 머신) 티어를 수평적으로 확장합니다. 로드 밸런서는 클라이언트에 대해 단일 접점의 역할을 하며 여러 JVM 간에 수신 트래픽을 분산합니다. 선택적으로 로드 밸런서는 트래픽을 여러 가용 영역에 분산하고 가용성을 높일 수 있습니다. EnterpriseOne

이 패턴에 설명된 프로세스는 로드 밸런서와 JVM 간의 트래픽을 암호화하는 대신 브라우저와 로드 밸런서 간의 암호화를 구성합니다. EnterpriseOne 이 접근 방식을 SSL 오프로딩이라고 합니다. EnterpriseOne 웹 또는 애플리케이션 서버에서 Application Load Balancer로 SSL 복호화 프로세스를 오프로드하면 애플리케이션 측의 부담이 줄어듭니다. 로드 밸런서에서 SSL이 종료되면 암호화되지 않은 트래픽이 AWS의 애플리케이션으로 라우팅됩니다.

[Oracle JD EnterpriseOne Edwards](#)는 제품 또는 물리적 자산을 제조, 건설, 유통, 서비스 또는 관리하는 조직을 위한 전사적 자원 관리 (ERP) 솔루션입니다. JD EnterpriseOne Edwards는 다양한 하드웨어, 운영 체제 및 데이터베이스 플랫폼을 지원합니다.

사전 조건 및 제한 사항

사전 조건

- 활성 상태의 AWS 계정
- AWS 서비스를 호출하고 AWS 리소스를 관리할 권한이 있는 AWS ID 및 액세스 관리 (IAM) 역할
- SSL 인증서

제품 버전

- 이 패턴은 Oracle WebLogic 12c에서 테스트되었지만 다른 버전도 사용할 수 있습니다.

아키텍처

SSL 오프로드를 수행하는 방법은 여러 가지가 있습니다. 이 패턴은 다음 다이어그램과 같이 Application Load Balancer와 Oracle HTTP Server(OHS)를 사용합니다.

다음 다이어그램은 JD Edwards EnterpriseOne, Application Load Balancer 및 JAS (자바 애플리케이션 서버) JVM 레이아웃을 보여줍니다.

도구

서비스

- [Application Load Balance](#)는 여러 가용 영역에서 Amazon Elastic Compute Cloud(Amazon EC2) 인스턴스 같은 여러 대상에 수신 애플리케이션 트래픽을 분산합니다.
- [AWS Certificate Manager\(ACM\)](#)는 AWS 웹사이트와 애플리케이션을 보호하는 퍼블릭 및 프라이빗 SSL/TLS X.509 인증서와 키를 만들고, 저장하고, 갱신하는 데 도움을 줍니다.
- [Amazon Route 53](#)은 가용성과 확장성이 뛰어난 DNS 웹 서비스입니다.

모범 사례

- ACM 모범 사례는 [ACM 설명서](#)를 참조하세요.

에픽

설정 및 OHS WebLogic

작업	설명	필요한 기술
<p>Oracle 구성 요소를 설치하고 구성합니다.</p>	<ol style="list-style-type: none"> 표준 설치 프로세스에 따라 퓨전 미들웨어 인프라를 설치합니다. 이 프로그램은 WebLogic 도메인을 설치하고 구성하는 데 도움이 됩니다. 자세한 지침은 Oracle 설명서를 참조하세요. 표준 설치 프로세스에 따라 OHS를 설치합니다. 자세한 지침은 Oracle 설명서를 참조하세요. 설치가 완료되면 구성 마법사(config.sh 파일)를 시작하여 OHS를 구성합니다. <ul style="list-style-type: none"> 기존 도메인을 업데이트하거나 새로운 도메인을 생성할 수 있습니다. 이 패턴은 기존 도메인을 업데이트한다고 가정합니다. 사용 가능한 템플릿의 경우 Oracle Enterprise Manager의 제한된 JRF와 Oracle HTTP Server(제한된 JRF)를 선택합니다. 이러한 Java 필수 파일(JRF) 옵션을 선택하면 외부 데이터베이스에 연결할 수 없습니다. 관리 대상 서버, 클러스터, 서버 템플릿, Coherence 	<p>JDE CNC, 관리자 WebLogic</p>

작업	설명	필요한 기술
	<p>클러스터, 머신, 머신에 서버 할당, 가상 대상 및 파티션의 경우 기본 구성 값을 그대로 사용하고 다음을 선택하여 다음 범주로 이동합니다.</p> <ul style="list-style-type: none"> OHS 인스턴스의 구성 세부 정보(예: 관리자 호스트 및 포트, 수신 주소 및 포트, 서버 이름)를 완료합니다(예: ohs1). 	
<p>도메인 수준에서 WebLogic 플러그인을 활성화합니다.</p>	<p>WebLogic 플러그인은 로드 밸런싱에 필요합니다. 플러그인을 활성화하려면:</p> <ol style="list-style-type: none"> 다음 링크를 사용하여 WebLogic 관리 콘솔에 로그인합니다. <p><code>http://<WeblogicServer>:<Adminport>/console</code></p> <ol style="list-style-type: none"> 잠금 및 편집을 선택한 다음 구성, 웹 애플리케이션을 선택합니다. WebLogic 플러그인 활성화 (확인란 또는 드롭다운 옵션)를 선택합니다. 변경 내용 저장 및 활성화를 선택합니다. 	<p>JDE CNC, 관리자 WebLogic</p>

작업	설명	필요한 기술
구성 파일을 편집합니다.	<p>이 <code>mod_wl_ohs.conf</code> 파일은 OHS에서의 프록시 요청을 구성합니다. WebLogic</p> <ol style="list-style-type: none"> 파일을 편집합니다. 위치는 다음과 같습니다. <p>\$ORACLE_HOME/user_projects/domains/</p> <p>예:</p> <pre>/home/oracle/Oracl e/Middleware/Oracl e_Home/user_projec ts/domains/base_do main/config/fmwcon fig/components/OHS /instances/ohs1</pre> WebLogic 호스트 (WebLogicHost) 및 포트 (WebLogicPort) 값을 추가합니다. (이 패턴은 로컬 호스트와 포트 8000을 가정합니다.) 다음과 같이 WLPProxySSL 및 WLPProxySSLPassThrough 값을 추가합니다. <pre><VirtualHost *:8000> <Location /jde> WLSRequest On SetHandler weblogic- handler</pre>	JDE CNC, 관리자 WebLogic

작업	설명	필요한 기술
	<pre>WebLogicHost localhost WebLogicPort 8000 WLProxySSL On WLProxySSLPassthrough On </Location> </VirtualHost></pre>	

작업	설명	필요한 기술
<p>Enterprise Manager를 사용하여 OHS를 시작합니다.</p>	<ol style="list-style-type: none"> 1. 다음 링크를 사용하여 Enterprise Manager 퓨전 미들웨어에 로그인합니다. <code>http://<WeblogicServer>:<Adminport>/em/</code> 2. 대상 탐색의 HTTP Server에서 OHS 인스턴스(예: ohs1)를 선택합니다. 3. 종료 및 시작을 선택하여 OHS 인스턴스를 다시 시작합니다. 4. OHS 설정이 완료되면 서버 호스트 이름 대신 포트 8000이 있는 HTTP 서버 호스트 이름을 사용하여 EnterpriseOne HTML 클라이언트에 연결할 수 있습니다. EnterpriseOne <ul style="list-style-type: none"> • 이전 링크: <code>http://<Webserver>:80/jde/owhtml</code> • 새로운 링크: <code>http://<HTTP server or web server>:8000/jde/owhtml</code> <p>기본 Oracle HTTP 포트 이외의 포트를 사용하는 경우 <code>httpd.conf</code> 파일을 편집하여 다음 두 위치에 해당 포트에 대한 리스너를 추가합니다.</p> 	<p>JDE CNC, 관리자 WebLogic</p>

작업	설명	필요한 기술
	<pre>#[Listen] OHS_LISTEN PORT Listen 8000</pre> <p>및</p> <pre># ServerName <Weblogic Server1>:8000</pre>	

Application Load Balancer를 구성합니다.

작업	설명	필요한 기술
대상 그룹을 설정합니다.	<ol style="list-style-type: none"> 1. HTTP 서버 포트 8000에 대한 대상 그룹을 생성합니다. 2. 동일한 포트를 사용하여 대상 그룹에 대상을 등록합니다. 3. 대상의 상태를 확인하여 정상인지 확인합니다. 4. 상태 확인 설정을 필요에 따라 구성합니다. <p>자세한 지침은 Elastic Load Balancing 설명서를 참조하세요.</p>	AWS 관리자
로드 밸런서를 설정합니다.	<ol style="list-style-type: none"> 1. 기본 속성과 필요한 Virtual Private Cloud(VPC), 보안 그룹 및 서브넷을 사용하여 Application Load Balancer를 생성합니다. 자세한 내용 	AWS 관리자

작업	설명	필요한 기술
	<p>은 탄력성 로드 밸런싱 설명서를 참조하세요.</p> <p>2. HTTPS 443에 대한 리스너 항목을 추가하고 이전 단계에서 생성한 대상 그룹에 전달합니다. 자세한 내용은 탄력성 로드 밸런싱 설명서를 참조하세요. HTTPS 리스너에는 SSL 인증서가 필요합니다. ACM에서 인증서를 선택하거나 업로드할 수 있습니다.</p> <p>3. 두 리스너 모두 Elastic Load Balancing 설명서의 지침을 따라 고정성을 활성화합니다.</p>	
<p>Route 53(DNS) 레코드 추가</p>	<p>(선택 사항) 하위 도메인에 대한 Amazon Route 53 DNS 레코드를 추가할 수 있습니다. 이 레코드는 Application Load Balancer를 가리킵니다. 자세한 지침은 Route 53 설명서를 참조하세요.</p>	<p>AWS 관리자</p>

문제 해결

문제	Solution
<p>HTTP 서버가 표시되지 않습니다.</p>	<p>Enterprise Manager 콘솔의 대상 탐색 목록에 HTTP Server가 나타나지 않는 경우 다음 단계를 수행합니다.</p>

문제	Solution
	<ol style="list-style-type: none"> 1. WebLogic 도메인, 관리에서 OHS 인스턴스를 선택합니다. 2. 생성을 선택하여 새 OHS 인스턴스를 생성합니다. 3. 인스턴스 이름을 입력한 다음 OK를 선택하여 인스턴스를 생성합니다. <p>인스턴스가 생성되고 변경 사항이 활성화되면 대상 탐색 패널에서 HTTP 서버를 볼 수 있습니다.</p>

관련 리소스

AWS 설명서

- [Application Load Balancers](#)
- [퍼블릭 호스팅 영역 작업](#)
- [프라이빗 호스팅 영역 사용](#)

Oracle 설명서:

- [Oracle WebLogic 서버 프록시 플러그인 개요](#)
- [인프라 설치 프로그램을 사용하여 WebLogic 서버 설치](#)
- [Oracle HTTP Server 설치 및 구성](#)

프라이빗 네트워크를 통해 Application Migration Service 데이터 및 컨트롤 플레인에 연결

작성자: Dipin Jain(AWS) 및 Mike Kuznetsov(AWS)

환경: PoC 또는 파일럿

기술: 네트워킹, 마이그레이션

AWS 서비스: AWS Application Migration Service, Amazon EC2, Amazon VPC, Amazon S3

요약

이 패턴은 인터페이스 VPC 엔드포인트를 사용하여 보안 프라이빗 네트워크의 AWS Application Migration Service (AWS MGN) 데이터 영역 및 컨트롤 플레인에 연결하는 방법을 설명합니다.

애플리케이션 마이그레이션 서비스는 AWS로 애플리케이션을 마이그레이션하는 작업을 간소화하고 속도를 높이며 비용을 절감하는 고도로 자동화된 lift-and-shift (재호스팅) 솔루션입니다. 이를 통해 기업은 호환성 문제, 성능 중단 또는 긴 전환 기간 없이 많은 수의 물리적, 가상 또는 클라우드 서버를 리호스팅할 수 있습니다. AWS Management Console에서 Application Migration Service를 이용할 수 있습니다. 이를 통해 AWS, Amazon CloudTrail CloudWatch, AWS ID 및 액세스 관리 (IAM) 와 같은 다른 AWS 서비스와 원활하게 통합할 수 있습니다.

AWS VPN 서비스, AWS Direct Connect 또는 Application Migration Service의 VPC 피어링을 사용하여 프라이빗 연결을 통해 소스 데이터 센터에서 데이터 영역(즉, 대상 VPC에서 데이터 복제를 위한 스테이징 영역 역할을 하는 서브넷)에 연결할 수 있습니다. 또한 AWS에서 제공하는 [인터페이스 VPC 엔드포인트](#)를 사용하여 프라이빗 네트워크를 통해 애플리케이션 마이그레이션 서비스 컨트롤 플레인에 PrivateLink 연결할 수 있습니다.

사전 조건 및 제한 사항

사전 조건

- 스테이징 영역 서브넷 – Application Migration Service를 설정하기 전에 소스 서버에서 AWS로 복제되는 데이터(즉, 데이터 영역)의 스테이징 영역으로 사용할 서브넷을 생성합니다. Application Migration Service 콘솔에 처음 액세스할 때 [복제 설정 템플릿](#)에서 이 서브넷을 지정해야 합니다. 복

제 설정 템플릿에서 특정 소스 서버에 대해 이 서브넷을 재정의할 수 있습니다. AWS 계정의 기존 서브넷을 사용할 수 있지만 이를 위해 전용 서브넷을 새로 생성하는 것이 좋습니다.

- 네트워크 요구 사항 – 스테이징 영역 서브넷에서 Application Migration Service가 시작하는 복제 서버는 복제 대상 AWS 리전의 코드인 <region>이 있는 <https://mgn.<region>.amazonaws.com/>의 Application Migration Service API 엔드포인트로 데이터를 전송할 수 있어야 합니다(예: https://mgn.us-east-1.amazonaws.com). Amazon Simple Storage Service(S3) 서비스 URL은 Application Migration Service 소프트웨어를 다운로드하는 데 필요합니다.
- AWS Replication Agent 설치 관리자는 Application Migration Service와 함께 사용 중인 AWS 리전의 S3 버킷 URL에 액세스할 수 있어야 합니다.
- 스테이징 영역 서브넷은 Amazon S3에 액세스할 수 있어야 합니다.
- AWS Replication Agent가 설치된 소스 서버는 스테이징 영역 서브넷의 복제 서버와 <https://mgn.<region>.amazonaws.com/>의 Application Migration Service API 엔드포인트로 데이터를 전송할 수 있어야 합니다.

다음 표에는 필수 포트가 나열되어 있습니다.

소스(Source)	대상	포트	자세한 내용은 단원을 참조하십시오.
소스 데이터 센터	Amazon S3 서비스 URL	443(TCP)	TCP 포트 443을 통한 통신
소스 데이터 센터	Application Migration Service를 위한 AWS 리전별 콘솔 주소	443(TCP)	TCP 포트 443을 통한 소스 서버와 Application Migration Service 간 통신
소스 데이터 센터	스테이징 영역 서브넷	1500(TCP)	TCP 포트 1500을 통한 소스 서버와 스테이징 영역 서브넷 간 통신
스테이징 영역 서브넷	Application Migration Service를 위한 AWS 리전별 콘솔 주소	443(TCP)	TCP 포트 443을 통한 스테이징 영역 서브넷과 Application

			Migration Service 간 통신
스태이징 영역 서브넷	Amazon S3 서비스 URL	443(TCP)	TCP 포트 443을 통한 통신
스태이징 영역 서브넷	서브넷 AWS 리전의 Amazon EC2 엔드포인트	443(TCP)	TCP 포트 443을 통한 통신

제한 사항

Application Migration Service는 현재 모든 AWS 리전 및 운영 체제에서 사용할 수 있는 것은 아닙니다.

- [지원되는 AWS 리전](#)
- [지원되는 운영 체제](#)

아키텍처

다음 다이어그램은 일반적인 마이그레이션에 대한 네트워크 아키텍처를 보여 줍니다. 이 아키텍처에 대한 자세한 내용은 [Application Migration Service 설명서](#)와 [Application Migration Service 아키텍처 및 네트워크 아키텍처 동영상](#)을 참조하십시오.

다음 세부 보기는 Amazon S3와 Application Migration Service를 연결하기 위한 스테이징 영역 VPC의 인터페이스 VPC 엔드포인트 구성을 보여줍니다.

도구

- [AWS Application Migration Service](#)는 AWS에서 애플리케이션을 리호스팅하는 작업을 간소화, 가속화하고 비용을 절감하는 AWS 서비스입니다.
- [인터페이스 VPC 엔드포인트](#)를 사용하면 인터넷 게이트웨이, NAT 디바이스, VPN 연결 또는 AWS Direct Connect 연결 PrivateLink 없이 AWS에서 제공하는 서비스에 연결할 수 있습니다. VPC의 인스턴스는 서비스의 리소스와 통신하는 데 퍼블릭 IP 주소를 필요로 하지 않습니다. VPC와 기타 서비스 간의 트래픽은 Amazon 네트워크를 벗어나지 않습니다.

에픽

Application Migration Service, Amazon EC2 및 Amazon S3용 엔드포인트 생성

작업	설명	필요한 기술
<p>Application Migration Service를 위한 인터페이스 엔드포인트를 구성합니다.</p>	<p>소스 데이터 센터와 스테이징 영역 VPC는 대상 스테이징 영역 VPC에서 생성한 인터페이스 엔드포인트를 통해 Application Migration Service 컨트롤 플레인에 비공개로 연결됩니다.</p> <p>엔드포인트 생성:</p> <ol style="list-style-type: none"> 1. https://console.aws.amazon.com/vpc/에서 Amazon VPC 콘솔을 엽니다. 2. 탐색 창에서 엔드포인트와 엔드포인트 생성을 선택합니다. 3. 서비스 범주에서 AWS services를 선택합니다. 4. 서비스 이름에 <code>com.amazonaws.<region>.mgn</code>를 입력합니다. 유형에서 인터페이스를 선택합니다. 5. VPC에서 엔드포인트를 생성할 대상 스테이징 영역 VPC를 선택합니다. 6. 서브넷에서 엔드포인트 네트워크 인터페이스를 생성할 서브넷(가용 영역)을 선택합니다. 7. 인터페이스 엔드포인트에 대한 프라이빗 DNS를 활성화하려면 추가 설정 섹션에 	<p>마이그레이션 책임자</p>

작업	설명	필요한 기술
	<p>서 DNS 이름 활성화를 선택합니다.</p> <p>8. TCP 443을 통해 스테이징 영역 VPC 서브넷으로부터의 수신을 허용하는 보안 그룹을 선택합니다.</p> <p>9. Create endpoint(엔드포인트 생성)을 선택합니다.</p> <p>자세한 내용은 Amazon VPC 설명서의 인터페이스 VPC 엔드포인트를 참조하십시오.</p>	
<p>Amazon EC2의 인터페이스 엔드포인트를 구성합니다.</p>	<p>스테이징 영역 VPC는 대상 스테이징 영역 VPC에 생성한 인터페이스 엔드포인트를 통해 Amazon EC2 API에 비공개로 연결됩니다. 엔드포인트를 생성하려면 이전 스토리에 제공된 지침을 따르십시오.</p> <ul style="list-style-type: none"> 서비스 이름에 <code>com.amazonaws.<region>.ec2</code> 를 입력합니다. 유형에서 인터페이스를 선택합니다. 보안 그룹은 포트 443을 통한 스테이징 영역 VPC 서브넷의 인바운드 HTTPS 트래픽을 허용해야 합니다. 추가 설정 섹션에서 DNS 이름 활성화를 선택합니다. 	<p>마이그레이션 책임자</p>

작업	설명	필요한 기술
<p>Amazon S3의 인터페이스 엔드포인트를 구성합니다.</p>	<p>소스 데이터 센터와 스테이징 영역 VPC는 대상 스테이징 영역 VPC에서 생성한 인터페이스 엔드포인트를 통해 Amazon S3 API에 비공개로 연결됩니다. 엔드포인트를 생성하려면 첫 번째 스토리에 제공된 지침을 따르십시오.</p> <ul style="list-style-type: none"> 서비스 이름에 <code>com.amazonaws.<region>.s3</code> 를 입력합니다. 유형에서 인터페이스를 선택합니다. VPC 보안 그룹은 포트 443을 통한 스테이징 영역 VPC 서브넷으로부터의 인바운드 HTTPS 트래픽을 허용해야 합니다. 추가 설정 섹션에서 DNS 이름 활성화를 선택 해제합니다. Amazon S3 인터페이스 엔드포인트는 프라이빗 DNS 이름을 지원하지 않습니다. <p>참고: 게이트웨이 엔드포인트 연결은 VPC 외부로 확장할 수 없으므로 인터페이스 엔드포인트를 사용합니다. (자세한 내용은 Amazon VPC 설명서를 참조하십시오.)</p>	<p>마이그레이션 책임자</p>

작업	설명	필요한 기술
<p>Amazon S3 게이트웨이 엔드포인트를 구성합니다.</p>	<p>구성 단계에서 복제 서버는 S3 버킷에 연결하여 AWS Replication 서버의 소프트웨어 업데이트를 다운로드해야 합니다. 하지만 Amazon S3 인터페이스 엔드포인트는 프라이빗 DNS 이름을 지원하지 않으며, Amazon S3 엔드포인트 DNS 이름을 복제 서버에 제공할 방법이 없습니다.</p> <p>이 문제를 완화하려면 스테이징 영역 서브넷이 속한 VPC에 Amazon S3 게이트웨이 엔드포인트를 생성하고 스테이징 서브넷의 라우팅 테이블을 관련 경로로 업데이트합니다. 자세한 내용은 AWS PrivateLink 설명서의 게이트웨이 엔드포인트 생성을 참조하십시오.</p>	<p>클라우드 관리자</p>

작업	설명	필요한 기술
<p>엔드포인트의 프라이빗 DNS 이름을 확인하도록 온프레미스 DNS를 구성합니다.</p>	<p>Application Migration Service 와 Amazon EC2의 인터페이스 엔드포인트에는 VPC에서 확인할 수 있는 프라이빗 DNS 이름이 있습니다. 하지만 이러한 인터페이스 엔드포인트의 프라이빗 DNS 이름을 확인하도록 온프레미스 서버를 구성해야 합니다.</p> <p>이러한 서버를 구성하는 방법은 여러 가지가 있습니다. 이 패턴에서는 온프레미스 DNS 쿼리를 스테이징 영역 VPC의 Amazon Route 53 Resolver 인바운드 엔드포인트로 전달하여 이 기능을 테스트했습니다. 자세한 내용은 Route 53 설명서에서 VPC와 네트워크 간 DNS 쿼리 해결을 참조하십시오.</p>	<p>마이그레이션 엔지니어</p>

전용 링크를 통해 Application Migration Service 컨트롤 플레인에 연결

작업	설명	필요한 기술
<p>AWS를 사용하여 AWS 복제 에이전트를 설치합니다 PrivateLink.</p>	<ol style="list-style-type: none"> 1. 대상 리전의 프라이빗 S3 버킷에 AWS 복제 에이전트를 다운로드합니다. 2. 마이그레이션할 소스 서버에 로그인합니다. AWS Replication Agent 설치 프로그램은 Application Migration Service 및 Amazon S3 엔드포인트에 	<p>마이그레이션 엔지니어</p>

작업	설명	필요한 기술
	<p>대한 네트워크 액세스 권한이 필요합니다. 온 프레미스 네트워크는 애플리케이션 마이그레이션 서비스 및 Amazon S3 퍼블릭 엔드포인트에 개방되어 있지 않으므로 AWS를 사용하여 이전 단계에서 생성한 인터페이스 엔드포인트를 사용하여 에이전트를 설치해야 합니다. PrivateLink</p> <p>다음은 Linux에 대한 예입니다.</p> <p>1. 명령을 사용하여 에이전트를 다운로드할 수 있습니다.</p> <pre data-bbox="594 997 1029 1436">wget -O ./aws-replication-installer-init.py \ https://aws-application-migration-service-<aws_region>.bucket.<s3-endpoint-DNS-name>/latest/linux/aws-replication-installer-init.py</pre> <p>참고: bucket는 Amazon S3 인터페이스 엔드포인트 DNS 이름 앞에 추가해야 하는 정적 키워드입니다. 자세한 내용은 Amazon S3 설명서를 참조하십시오.</p> <p>예를 들어 Amazon S3 인터페이스 엔드포인트의 DNS 이름</p>	

작업	설명	필요한 기술
	<p>이 <code>vpce-009c8b07adb052a11-qgf8q50y.s3.us-west-1.vpce.amazonaws.com</code> 이고 AWS 리전이 <code>us-west-1</code> 인 경우 다음 명령을 사용합니다.</p> <pre>wget -O ./aws-replication-installer-init.py \ https://aws-application-migration-service-us-west-1.bucket.vpce-009c8b07adb052a11-qgf8q50y.s3.us-west-1.vpce.amazonaws.com/latest/linux/aws-replication-installer-init.py</pre> <p>2. 에이전트 설치:</p> <ul style="list-style-type: none"> Application Migration Service를 위한 인터페이스 엔드포인트를 생성할 때 DNS 이름 활성화를 선택한 경우 다음 명령을 실행합니다. <pre>sudo python3 aws-replication-installer-init.py \ --region <aws_region> \ --aws-access-key-id <access-key> \</pre>	

작업	설명	필요한 기술
	<pre data-bbox="613 212 1008 426"> --aws-secret-access-key <secret-key> \ --no-prompt \ --s3-endpoint <s3-endpoint-DNS-name> </pre> <ul data-bbox="594 464 1019 737" style="list-style-type: none"> • Application Migration Service를 위한 인터페이스 엔드포인트를 생성할 때 DNS 이름 활성화를 선택하지 않은 경우 다음 명령을 실행합니다. <pre data-bbox="613 810 1008 1402"> sudo python3 aws-replication-installer-init.py \ --region <aws_region> \ --aws-access-key-id <access-key> \ --aws-secret-access-key <secret-key> \ --no-prompt \ --s3-endpoint <s3-endpoint-DNS-name> \ --endpoint <mgn-endpoint-DNS-name> </pre> <p data-bbox="594 1444 1019 1623"> 자세한 내용은 Application Migration Service 설명서의 AWS Replication 에이전트 설치 지침을 참조하십시오. </p> <p data-bbox="594 1671 1019 1848"> Application Migration Service에 연결하고 AWS Replication Agent를 설치한 후, Application Migration Service 설명서의 </p>	

작업	설명	필요한 기술
	지침에 따라 소스 서버를 대상 VPC 및 서브넷으로 마이그레이션하십시오.	

관련 리소스

Application Migration 서비스 설명서

- [개념](#)
- [마이그레이션 워크플로](#)
- [빠른 시작 설명서](#)
- [FAQ](#)
- [문제 해결](#)

추가 리소스

- [AWS Application Migration Service - 기술 소개](#)(AWS Training and Certification 안내)
- [AWS Application Migration Service 아키텍처 및 네트워크 아키텍처](#)(동영상)

추가 정보

Linux 서버의 AWS Replication 에이전트 설치 문제 해결

Amazon Linux 서버에서 gcc 오류가 발생하는 경우 패키지 리포지토리를 구성하고 다음 명령을 사용하십시오.

```
## sudo yum groupinstall "Development Tools"
```

AWS CloudFormation 사용자 지정 리소스와 Amazon SNS를 사용하여 인포블록 객체를 생성합니다.

작성자: Tim Sutton(AWS)

환경: PoC 또는 파일럿

기술: 네트워킹

워크로드: 기타 모든 워크로드

AWS 서비스: 아마존 SNS, AWS, AWS KMS CloudFormation, AWS Lambda, AWS Organizations

요약

Infoblox 도메인 이름 시스템(DNS), Dynamic Host Configuration Protocol(DHCP) 및 IP 주소 관리 ([Infoblox DDI](#))를 사용하면 복잡한 하이브리드 환경을 중앙 집중화하고 효율적으로 제어할 수 있습니다. Infoblox DDI를 사용하면 동일한 어플라이언스를 사용하여 온프레미스와 Amazon Web Services(AWS) 클라우드의 DNS를 관리하는 것 외에도 하나의 신뢰할 수 있는 IP 주소 관리자(IPAM) 데이터베이스에서 모든 네트워크 자산을 검색하고 기록할 수 있습니다.

이 패턴은 Infoblox WAPI API를 호출하여 AWS CloudFormation 사용자 지정 리소스를 사용하여 Infoblox 객체 (예: DNS 레코드 또는 IPAM 객체) 를 생성하는 방법을 설명합니다. Infoblox WAPI에 대한 자세한 내용은 Infoblox 설명서의 [WAPI 설명서](#)를 참조하세요.

이 패턴의 접근 방식을 사용하면 레코드를 생성하고 네트워크를 프로비저닝하는 수동 프로세스를 제거하는 것 외에도 AWS 및 온프레미스 환경의 DNS 레코드 및 IPAM 구성을 통합적으로 볼 수 있습니다. 다음과 같은 사용 사례에서 이 패턴의 접근 방식을 사용할 수 있습니다.

- Amazon Elastic Compute Cloud(Amazon EC2) 인스턴스를 생성한 후 A 레코드 추가
- Application Load Balancer를 생성한 후 CNAME 레코드 추가
- Virtual Private Cloud(VPC) 생성 후 네트워크 객체 추가
- 다음 네트워크 범위를 제공하고 해당 범위를 사용하여 서브넷을 생성

또한 이 패턴을 확장하여 다른 DNS 레코드 유형 추가 또는 Infoblox vDiscovery 구성과 같은 다른 Infoblox 장치 기능을 사용할 수 있습니다.

이 패턴은 허브가 AWS 클라우드 또는 온프레미스의 Infoblox 어플라이언스에 연결되어야 하고 AWS Lambda를 사용하여 Infoblox API를 호출하는 hub-and-spoke 설계를 사용합니다. 스포크는 AWS Organizations의 동일한 조직 내 동일하거나 다른 계정에 있으며, AWS 사용자 지정 리소스를 사용하여 Lambda 함수를 호출합니다. CloudFormation

사전 조건 및 제한 사항

사전 조건

- AWS 클라우드, 온프레미스 또는 둘 다에 설치되고 IPAM 및 DNS 작업을 관리할 수 있는 관리자 사용자로 구성된 기존 Infoblox 어플라이언스 또는 그리드. 자세한 내용은 Infoblox 설명서의 [관리자 계정 정보](#)를 참조하세요.
- Infoblox 어플라이언스에 레코드를 추가하려는 기존 DNS 권한 영역. 이에 대한 자세한 내용은 Infoblox 설명서의 [신뢰할 수 있는 영역 구성](#)을 참조하세요.
- AWS Organizations의 활성 AWS 계정 2개. 한 계정은 허브 계정이고 다른 계정은 스포크 계정이어야 합니다.
- CMK는 동일한 AWS 계정 및 리전에 있어야 합니다.
- 허브 계정의 VPC는 Infoblox 어플라이언스에 연결해야 합니다.(예: AWS Transit Gateway 또는 VPC 피어링 사용)
- [AWS 서버리스 애플리케이션 모델 \(AWS SAM\), 로컬에 설치되고 AWS Cloud9 또는 AWS로 구성되어 있습니다.](#) CloudShell
- AWS SAM이 포함된 로컬 환경에 다운로드된 Infoblox-Hub.zip 및 ClientTest.yaml 파일(첨부).

제한 사항

- AWS CloudFormation 사용자 지정 리소스의 서비스 토큰은 스택이 생성된 지역과 동일한 지역의 것이어야 합니다. 한 리전에서 Amazon Simple Notification Service(SNS) 주제를 생성하고 다른 리전에서 Lambda 함수를 직접 호출하는 대신 각 리전에서 허브 계정을 사용하는 것이 좋습니다.

제품 버전

- Infoblox API 버전 2.7

아키텍처

다음 다이어그램은 이 워크플로를 보여 줍니다.

이 패턴 솔루션의 다음 구성 요소를 보여주는 다이어그램입니다.

1. AWS CloudFormation 사용자 지정 리소스를 사용하면 스택을 생성, 업데이트 또는 삭제할 때 AWS에서 CloudFormation 실행하는 템플릿에 사용자 지정 프로비저닝 로직을 작성할 수 있습니다. 스택을 생성하면 AWS는 EC2 인스턴스에서 실행되는 애플리케이션이 모니터링하는 SNS 주제에 create 요청을 CloudFormation 보냅니다.
2. AWS CloudFormation 사용자 지정 리소스의 Amazon SNS 알림은 특정 AWS KMS (키 관리 서비스) 키를 통해 암호화되며, Organizations에 있는 조직의 계정으로만 액세스가 제한됩니다. SNS 주제는 Infoblox WAPI API를 직접 호출하는 Lambda 리소스를 시작합니다.
3. Amazon SNS는 Infoblox의 WAPI URL, 사용자 이름, 암호 AWS Secrets Manager Amazon 리소스 이름(ARN)을 환경 변수로 사용하는 다음과 같은 Lambda 함수를 간접적으로 호출합니다.
 - `dnsapi.lambda_handler`— AWS CloudFormation 사용자 지정 리소스에서 `DNSNameDNSType`, 및 `DNSValue` 값을 수신하고 이를 사용하여 DNS A 레코드 및 CNAME을 생성합니다.
 - `ipaddr.lambda_handler`— AWS CloudFormation 사용자 지정 리소스에서 `VPCCIDRTypeSubnetPrefix`, 및 `Network Name` 값을 수신하고 이를 사용하여 네트워크 데이터를 Infoblox IPAM 데이터베이스에 추가하거나 새 서브넷을 생성하는 데 사용할 수 있는 다음 가용 네트워크를 사용자 지정 리소스에 제공합니다.
 - `describeprefixes.lambda_handler` - "com.amazonaws."+Region+".s3" 필터를 사용하여 필요한 prefix ID을(를) 검색하고 `describe_managed_prefix_lists` AWS API를 직접 호출합니다.

중요: 이러한 Lambda 함수는 Python으로 작성되었으며 서로 비슷하지만 서로 다른 API를 직접 호출합니다.

4. Infoblox 그리드를 물리적, 가상 또는 클라우드 기반 네트워크 어플라이언스로 배포할 수 있습니다. 온프레미스로 배포하거나 VMware ESXi, Microsoft Hyper-V, Linux KVM 및 Xen을 비롯한 다양한 하이퍼바이저를 사용하여 가상 어플라이언스로 배포할 수 있습니다. Amazon Machine Image(AMI)를 사용하여 AWS 클라우드에 Infoblox 그리드를 배포할 수도 있습니다.
5. 다음 다이어그램은 AWS 클라우드와 온프레미스의 리소스에 DNS와 IPAM을 제공하는 Infoblox 그리드용 하이브리드 솔루션을 보여줍니다.

기술 스택

- AWS CloudFormation
- IAM
- AWS KMS
- AWS Lambda
- AWS SAM
- AWS Secrets Manager
- Amazon SNS
- Amazon VPC

도구

- [AWS는 AWS CloudFormation](#) 리소스를 설정하고, 빠르고 일관되게 프로비저닝하고, AWS 계정 및 지역 전반의 수명 주기 전반에 걸쳐 리소스를 관리할 수 있도록 지원합니다.
- [AWS Identity and Access Management\(IAM\)](#)은 누구에게 인증 및 사용 권한이 있는지 제어하여 AWS 리소스에 대한 액세스를 안전하게 관리할 수 있도록 도와줍니다.
- [AWS Key Management Service\(AWS KMS\)](#)를 사용하면 암호화 키를 생성하고 제어하여 데이터를 보호할 수 있습니다.
- [Lambda](#)는 서버를 프로비저닝하거나 관리할 필요 없이 코드를 실행할 수 있도록 도와주는 컴퓨팅 서비스입니다. 필요할 때만 코드를 실행하며 자동으로 확장이 가능하므로 사용한 컴퓨팅 시간만큼만 비용을 지불합니다.
- [AWS Organizations](#)는 사용자가 생성하고 중앙에서 관리하는 조직으로 여러 AWS 계정을 통합할 수 있는 계정 관리 서비스입니다.
- [AWS Secrets Manager](#)를 사용하면 코드에 하드코딩된 보안 인증 정보(암호 등)를 Secrets Manager에 대한 API 직접 호출로 바꾸어 프로그래밍 방식으로 보안 암호를 검색할 수 있습니다.
- [AWS Serverless Application Model\(AWS SAM\)](#)은 AWS 클라우드에서 서버리스 애플리케이션을 구축하는 데 사용할 수 있는 오픈소스 프레임워크입니다.
- [Amazon Simple Notification Service\(Amazon SNS\)](#)를 사용하면 웹 서버 및 이메일 주소를 포함하여 게시자와 클라이언트 간의 메시지 교환을 조정하고 관리할 수 있습니다.
- [Amazon Virtual Private Cloud\(VPC\)](#)를 이용하면 사용자가 정의한 가상 네트워크로 AWS 리소스를 시작할 수 있습니다. 이 가상 네트워크는 사용자의 자체 데이터 센터에서 운영하는 기존 네트워크와 유사하며 AWS의 확장 가능한 인프라를 사용한다는 이점이 있습니다.

코드

ClientTest.yaml 샘플 AWS CloudFormation 템플릿 (첨부) 을 사용하여 Infoblox 허브를 테스트할 수 있습니다. 다음 표의 사용자 지정 리소스를 포함하도록 AWS CloudFormation 템플릿을 사용자 지정할 수 있습니다.

Infoblox 스포크 사용자 지정 리소스를 사용하여 A 레코드를 생성

반환 값:

infobloxref – Infoblox 참조

리소스 예제:

```
ARECORDCustomResource:

  Type: "Custom::InfobloxAPI"

  Properties:

    ServiceToken: !Sub arn:aws:sns:
${AWS::Region}:${HubAccountID}:Ru
nInfobloxDNSFunction

    DNSName: 'arecordtest.compa
ny.com'

    DNSType: 'ARecord'

    DNSValue: '10.0.0.1'
```

Infoblox 스포크 사용자 지정 리소스를 사용하여 CNAME 레코드 생성

반환 값:

infobloxref – Infoblox 참조

리소스 예제:

```
CNAMECustomResource:

  Type: "Custom::InfobloxAPI"

  Properties:
```

```

ServiceToken: !Sub arn:aws:sns:
${AWS::Region}:${HubAccountID}:Ru
nInfoblox

DNSFunction

DNSName: 'cnametest.company.com'

DNSType: 'cname'

DNSValue: 'aws.amazon.com'

```

Infoblox 스포크 사용자 지정 리소스를 사용하여
네트워크 개체 생성

반환 값:

`infobloxref` – Infoblox 참조

`network` – 네트워크 범위(VPCCIDR와 동일)

리소스 예제:

```

VPCCustomResource:

  Type: 'Custom::InfobloxAPI'

  Properties:

    ServiceToken: !Sub arn:aws:sns:
${AWS::Region}:${HubAccountID}:Ru
nInfobloxNextSubnetFunction

    VPCCIDR: !Ref VpcCIDR

    Type: VPC

    NetworkName: My-VPC

```

Infoblox 스포크 사용자 지정 리소스를 사용하여 다음의 사용 가능한 서브넷을 검색

반환 값:

infobloxref - Infoblox 참조

network - 서브넷의 네트워크 범위

리소스 예제:

```
Subnet1CustomResource:
  Type: 'Custom::InfobloxAPI'
  DependsOn: VPCCustomResource
  Properties:
    ServiceToken: !Sub arn:aws:sns:
    ${AWS::Region}:${HubAccountID}:Ru
    nInfobloxNextSubnetFunction
    VPCCIDR: !Ref VpcCIDR
    Type: Subnet
    SubnetPrefix: !Ref SubnetPrefix
  NetworkName: My-Subnet
```

에픽

허브 계정의 VPC 생성 및 구성

작업	설명	필요한 기술
Infoblox 어플라이언스에 연결하여 VPC를 생성합니다.	허브 계정을 위한 AWS Management Console에 로그인하고 AWS Quick Start의 AWS 클라우드 킷 스타트 참조 배포의 Amazon VPC 에 있는	네트워크 관리자, 시스템 관리자

작업	설명	필요한 기술
	<p>단계에 따라 VPC를 생성합니다.</p> <p>중요: VPC는 Infoblox 어플라이언스에 HTTPS로 연결되어 있어야 하며 이 연결에는 프라이빗 서브넷을 사용하는 것이 좋습니다.</p>	
<p>(선택 사항) 프라이빗 서브넷을 위한 VPC 엔드포인트를 생성하세요.</p>	<p>VPC 엔드포인트는 프라이빗 서브넷에 퍼블릭 서비스에 대한 연결을 제공합니다. 다음 엔드포인트는 필수 엔드포인트입니다.</p> <ul style="list-style-type: none"> • Lambda가 AWS와 통신할 수 있도록 해주는 Amazon Simple Storage 서비스 (Amazon S3) 의 게이트웨이 엔드포인트 CloudFormation • Secrets Manager와의 연결을 지원하는 Secrets Manager의 인터페이스 엔드포인트 • SNS 주제 및 Secrets Manager 비밀의 암호화를 허용하는 AWS KMS용 인터페이스 엔드포인트 <p>프라이빗 서브넷용 엔드포인트를 생성하는 방법에 대한 자세한 내용은 Amazon VPC 설명서의 VPC 엔드포인트 섹션을 참조하세요.</p>	<p>네트워크 관리자, 시스템 관리자</p>

Infoblox 허브 배포하기

작업	설명	필요한 기술
<p>AWS SAM 템플릿을 구축하세요.</p>	<ol style="list-style-type: none"> 1. AWS SAM이 포함된 환경에서 <code>unzip Infoblox-Hub.zip</code> 명령을 실행합니다. 2. <code>cd Hub/</code> 명령을 실행하여 사용자 디렉터리를 Hub 디렉터리로 변경합니다. 3. <code>sam build</code> 명령을 실행하여 AWS SAM 템플릿 파일, 애플리케이션 코드, 모든 언어별 파일 및 종속성을 처리합니다. 또한 <code>sam build</code> 명령은 다음 스토리에 예상되는 형식과 위치로 빌드 아티팩트를 복사합니다. 	<p>개발자, 시스템 관리자</p>
<p>AWS SAM 템플릿을 배포하세요.</p>	<p>이 <code>sam deploy</code> 명령은 필수 파라미터를 가져와 <code>samconfig.toml</code> 파일에 저장하고, AWS CloudFormation 템플릿과 Lambda 함수를 S3 버킷에 저장한 다음, CloudFormation AWS 템플릿을 허브 계정에 배포합니다.</p> <p>다음 샘플 코드는 AWS SAM 템플릿을 배포하는 방법을 보여줍니다.</p> <pre data-bbox="597 1671 1026 1881"> \$ sam deploy --guided Configuring SAM deploy ===== == </pre>	<p>개발자, 시스템 관리자</p>

작업	설명	필요한 기술
	<pre> Looking for config file [samconfi g.toml] : Found Reading default arguments : Success Setting default arguments for 'sam deploy' ===== ===== ===== Stack Name [Infoblox-Hub]: AWS Region [eu- west-1]: Parameter InfobloxUsername: Parameter InfobloxPassword: Parameter InfobloxIPAddress [xxx.xxx.xx.xxx]: Parameter AWSOrganisationID [o- xxxxxxxxxx]: Parameter VPCID [vpc-xxxxxxxxxx]: Parameter VPCCIDR [xxx.xxx. xxx.xxx/16]: Parameter VPCSubnetID1 [subnet-x xx]: Parameter VPCSubnetID2 [subnet-x xx]: Parameter VPCSubnetID3 [subnet-x xx]: Parameter VPCSubnetID4 []: </pre>	

작업	설명	필요한 기술
	<pre data-bbox="592 205 1031 1218"> #Shows you resources changes to be deployed and require a 'Y' to initiate deploy Confirm changes before deploy [Y/n]: y #SAM needs permission to be able to create roles to connect to the resources in your template Allow SAM CLI IAM role creation [Y/n]: n Capabilities [['CAPABI LITY_NAMED_IAM']]: Save arguments to configuration file [Y/n]: y SAM configura tion file [samconfi g.toml]: SAM configura tion environment [default]: </pre> <p data-bbox="592 1249 1031 1491"> 중요: Infoblox 로그인 보안 인 증 정보가 samconfig.toml 파일에 저장되지 않으므로 매 번 --guided 옵션을 사용해야 합니다. </p>	

관련 리소스

- [Postman을 사용하여 WAPI 시작하기](#)(Infoblox 블로그)
- [BYOL 모델을 사용한 AWS용 vNIOS 프로비저닝](#)(Infoblox 설명서)
- [quickstart-aws-vpc](#)(리포지토리) GitHub
- [describe_managed_prefix_lists](#) (AWS SDK for Python 설명서)

첨부

이 문서와 관련된 추가 콘텐츠에 액세스하려면 [attachment.zip](#) 파일의 압축을 풉니다.

AWS 네트워크 방화벽에 대한 Amazon CloudWatch 알림을 사용자 지정합니다.

작성자: Jason Owens(AWS)

환경: PoC 또는 파일럿

기술: 네트워킹; 보안, 자격 증명, 규정 준수

워크로드: 오픈 소스

AWS 서비스: 아마존 CloudWatch 로그, AWS 네트워크 방화벽, AWS CLI

요약

이 패턴은 Amazon Web Services (AWS) 네트워크 방화벽에서 생성되는 Amazon CloudWatch 알림을 사용자 지정하는 데 도움이 됩니다. 사전 정의된 규칙을 사용하거나 알림의 메시지, 메타데이터 및 심각도를 결정하는 사용자 지정 규칙을 생성할 수 있습니다. 그런 다음 이러한 경고에 따라 조치를 취하거나 Amazon과 같은 다른 Amazon 서비스의 응답을 자동화할 수 EventBridge 있습니다.

이 패턴에서는 Suricata와 호환되는 방화벽 규칙을 생성합니다. [Suricata](#)는 오픈 소스 위협 탐지 엔진입니다. 먼저 간단한 규칙을 만든 다음 테스트하여 CloudWatch 알림이 생성되고 기록되는지 확인합니다. 규칙을 성공적으로 테스트한 후에는 규칙을 수정하여 사용자 지정 메시지, 메타데이터 및 심각도를 정의한 다음 다시 한 번 테스트하여 업데이트를 확인합니다.

사전 조건 및 제한 사항

사전 조건

- 활성 상태의 AWS 계정.
- Linux, macOS 또는 Windows 워크스테이션에 AWS Command Line Interface(AWS CLI)를 설치하고 구성합니다. 자세한 내용은 [최신 버전의 AWS CLI 설치 또는 업데이트](#)를 참조하세요.
- AWS Network Firewall은 CloudWatch 로그를 사용하도록 설치 및 구성되었습니다. 자세한 내용은 [AWS Network Firewall에서 네트워크 트래픽 로깅](#)을 참조하세요.
- Network Firewall을 통해 보호되는 Virtual Private Cloud(VPC)의 프라이빗 서브넷에 있는 Amazon Elastic Compute Cloud(Amazon EC2) 인스턴스입니다.

제품 버전

- AWS CLI의 버전 1의 경우 1.18.180 이상을 사용하세요. AWS CLI의 버전 2의 경우 2.1.2 이상을 사용하세요.
- Suricata 버전 5.0.2의 classification.config 파일입니다. 이 구성 파일의 사본은 [추가 정보](#) 섹션을 참조하세요.

아키텍처

대상 기술 스택

- Network Firewall
- 아마존 CloudWatch 로그

대상 아키텍처

다이어그램은 다음 아키텍처를 보여줍니다.

1. 프라이빗 서브넷의 EC2 인스턴스는 [curl](#) 또는 [Wget](#)을 사용하여 요청을 보냅니다.
2. Network Firewall은 트래픽을 처리하고 알림을 생성합니다.
3. Network Firewall은 CloudWatch 로그에 기록된 경고를 보냅니다.

도구

서비스

- [Amazon](#)은 AWS 리소스의 지표와 AWS에서 실행하는 애플리케이션을 실시간으로 모니터링할 수 있도록 CloudWatch 도와줍니다.
- [Amazon CloudWatch Logs](#)를 사용하면 모든 시스템, 애플리케이션 및 AWS 서비스의 로그를 중앙 집중화하여 로그를 모니터링하고 안전하게 보관할 수 있습니다.
- [AWS Command Line Interface\(AWS CLI\)](#)는 명령줄 셸에서 명령을 사용하여 AWS 서비스와 상호 작용할 수 있는 오픈 소스 도구입니다.
- [AWS Network Firewall](#)은 AWS 클라우드에 있는 Virtual Private Cloud(VPC)를 위한 상태 저장형, 관리형, Network Firewall 및 침입 탐지 및 방지 서비스입니다.

기타 도구 및 서비스

- [curl](#) – curl은 오픈 소스 명령줄 도구 및 라이브러리입니다.
- [Wget](#) – GNU Wget은 무료 명령줄 도구입니다.

에픽

방화벽 규칙 및 규칙 그룹 생성

작업	설명	필요한 기술
규칙을 생성합니다.	<p>1. 텍스트 편집기에서 방화벽에 추가할 규칙 목록을 만듭니다. 각 규칙은 별도의 줄에 있어야 합니다. classtype 파라미터의 값은 기본 Suricata 분류 구성 파일에서 가져온 것입니다. 전체 구성 파일 내용은 추가 정보 섹션을 참조하세요. 다음은 규칙의 두 가지 예제입니다.</p> <pre> alert http any any -> any any (content:"badstuff"; classtype:misc-activity; sid:3; rev:1;) alert http any any -> any any (content:"morebadstuff"; classtype:bad-unknown; sid:4; rev:1;) </pre> <p>2. custom.rules 라는 파일에 규칙을 저장합니다.</p>	AWS 시스템 관리자, 네트워크 관리자

작업	설명	필요한 기술
<p>규칙 그룹을 생성합니다.</p>	<p>AWS CLI에서 다음 명령을 입력합니다. 이렇게 하면 규칙 그룹이 생성됩니다.</p> <pre data-bbox="597 394 1024 869"># aws network-firewall create-rule-group \ --rule-group- name custom --type STATEFUL \ --capacity 10 --rules file://cu stom.rules \ --tags Key=envir onment,Value=devel opment</pre> <p>다음은 예시 출력입니다. 이후 단계에서 필요할 RuleGroup Arn 을 메모합니다.</p> <pre data-bbox="597 1079 1024 1835">{ "UpdateToken": "4f998d72-973c-490a- bed2-fc3460547e23", "RuleGroupResponse ": { "RuleGroupArn": "arn:aws:network-f irewall:us-east-2: 1234567890:stateful- rulegroup/custom", "RuleGrou pName": "custom", "RuleGroupId": "238a8259-9eaf-48b b-90af-5e690cf8c48b", "Type": "STATEFUL", "Capacity": 10,</pre>	<p>AWS 시스템 관리자</p>

작업	설명	필요한 기술
	<pre> "RuleGroupStatus": "ACTIVE", "Tags": [{ "Key": "environment", "Value": "development" }] } </pre>	

방화벽 정책 업데이트

작업	설명	필요한 기술
방화벽 정책의 ARN 획득.	<p>AWS CLI에서 다음 명령을 입력합니다. 이는 방화벽 정책의 Amazon 리소스 이름(ARN)을 반환합니다. 나중에 이 패턴에서 사용할 수 있도록 ARN을 기록합니다.</p> <pre> # aws network-firewall describe-firewall \ --firewall-name aws-network-firewall- anfw \ --query 'Firewall .FirewallPolicyArn' </pre> <p>다음은 이 명령을 통해 반환된 ARN 예시입니다.</p> <pre> "arn:aws:network-f irewall:us-east-2: 1234567890:firewal </pre>	AWS 시스템 관리자

작업	설명	필요한 기술
	l-policy/firewall-policy-anfw"	

작업	설명	필요한 기술
<p>방화벽 정책을 업데이트합니다.</p>	<p>텍스트 편집기에서 다음 코드를 붙여 넣습니다. <RuleGroupArn> 을 이전 에픽에서 기록한 값으로 바꾸세요. 파일을 firewall-policy-anfw.json (으)로 저장합니다.</p> <pre data-bbox="594 537 1029 1331"> { "StatelessDefaultActions": ["aws:forward_to_sfe"], "StatelessFragmentDefaultActions": ["aws:forward_to_sfe"], "StatefulRuleGroupReferences": [{ "ResourceArn": "<RuleGroupArn>" }] } </pre> <p>AWS CLI에서 다음 명령을 입력합니다. 이 명령을 사용해 새 규칙을 추가하려면 업데이트 토큰이 필요합니다. 토큰은 정책을 마지막으로 검색한 이후 정책이 변경되지 않았음을 확인하는 데 사용됩니다.</p> <pre data-bbox="594 1730 1029 1822"> UPDATETOKEN=(`aws network-firewall </pre>	<p>AWS 시스템 관리자</p>

작업	설명	필요한 기술
	<pre> describe-firewall- policy \ -- firewall-policy-name firewall-policy-anfw \ --output text --query UpdateTok en`) aws network-firewall update-firewall-po licy \ --update-token \$UPDATETOKEN \ --firewall-policy- name firewall-policy- anfw \ --firewall-policy file://firewall-po licy-anfw.json </pre>	

작업	설명	필요한 기술
<p>정책 업데이트를 확인합니다.</p>	<p>(선택 사항)규칙이 추가되었는지 확인하고 정책 형식을 보려면 AWS CLI에서 다음 명령을 입력합니다.</p> <pre data-bbox="597 443 1026 800"># aws network-firewall describe-firewall- policy \ --firewall-policy- name firewall-policy- anfw \ --query FirewallP olicy</pre> <p>다음은 예시 출력입니다.</p> <pre data-bbox="597 911 1026 1864">{ "StatelessDefaultA ctions": ["aws:forw ard_to_sfe"], "StatelessFragment DefaultActions": ["aws:forw ard_to_sfe"], "StatefulRuleGroup References": [{ "Resource Arn": "arn:aws: network-firewall:u s-east-2:123456789 0:stateful-rulegroup/ custom" }] }</pre>	<p>AWS 시스템 관리자</p>

테스트 경고 기능

작업	설명	필요한 기술
테스트용 경고 생성.	<ol style="list-style-type: none"> 1. 방화벽 서브넷 내의 테스트 워크스테이션에 로그인합니다. 2. 경고를 생성하는 명령을 입력합니다. 예를 들어, wget 또는 curl를 사용할 수 있습니다. <pre data-bbox="630 695 1029 856">wget -U "badstuff" http://www.amazon. com -o /dev/null</pre> <pre data-bbox="630 884 1029 1087">curl -A "morebads tuff" http://ww w.amazon.com -o / dev/null</pre>	AWS 시스템 관리자
경고가 로그되었는지 확인합니다.	<ol style="list-style-type: none"> 1. https://console.aws.amazon.com/cloudwatch/에서 CloudWatch 콘솔을 엽니다. 2. 올바른 로그 그룹 및 스트림으로 이동합니다. 자세한 내용은 로그로 전송된 로그 데이터 보기 (CloudWatch CloudWatch 로그 문서)를 참조하십시오. 3. 로그 이벤트가 다음 예제와 비슷한지 확인하세요. 예제는 경고 관련 부분만 보여줍니다. <p>예 1</p>	AWS 시스템 관리자

작업	설명	필요한 기술
	<pre> "alert": { "action": "allowed", "signatur e_id": 3, "rev": 1, "signatur e": "", "category ": "Misc activity", "severity ": 3 } </pre> <p>예제 2</p> <pre> "alert": { "action": "allowed", "signatur e_id": 4, "rev": 1, "signatur e": "", "category ": "Potentially Bad Traffic", "severity ": 2 } </pre>	

방화벽 규칙 및 규칙 그룹 업데이트

작업	설명	필요한 기술
방화벽 규칙 업데이트.	<ol style="list-style-type: none"> 1. 텍스트 편집기에서 <code>custom.rules</code> 파일을 엽니다. 2. 다음과 유사하게 첫 번째 규칙을 변경합니다. 이 규칙은 파일에서 한 줄에 입력해야 합니다. <div data-bbox="630 699 1029 1178" style="border: 1px solid #ccc; border-radius: 10px; padding: 10px; margin: 10px 0;"> <pre>alert http any any -> any any (msg:"Watch out - Bad Stuff!!"; content:"badstuff" ; classtype:misc- activity; priority: 2; sid:3; rev:2; metadata:custom- field-2 Danger!, custom-field More Info;)</pre> </div> <p>이렇게 하면 규칙이 다음과 같이 변경됩니다.</p> <ul style="list-style-type: none"> • 서명 또는 경고에 대한 텍스트 정보를 제공하는 msg(Suricata 웹사이트) 문자열을 추가합니다. 생성된 경고에서 이는 서명에 매핑됩니다. • 기본 <code>misc-activity</code> 의 우선순위(Suricata 웹사이트)를 3에서 2로 조정합니다. 다양한 <code>classtypes</code> 의 기 	AWS 시스템 관리자

작업	설명	필요한 기술
	<p>본값은 추가 정보 섹션을 참조하세요.</p> <ul style="list-style-type: none"> 경고에 사용자 지정 메타 데이터(Suricata 웹 사이트)를 추가합니다. 이는 서명에 추가되는 추가 정보입니다. 키값 페어를 사용하는 것이 좋습니다. rev(Suricata 웹사이트)를 1에서 2로 변경합니다. 이는 서명 버전을 나타냅니다. 	

작업	설명	필요한 기술
<p>규칙 그룹을 업데이트합니다.</p>	<p>AWS CLI에서 다음 명령을 사용합니다. 방화벽 정책의 ARN을 사용하세요. 이 명령은 업데이트 토큰을 얻고 규칙 그룹을 규칙 변경으로 업데이트합니다.</p> <pre data-bbox="597 537 1024 1014"> # UPDATETOKEN=(`aws network-firewall \ describe-rule-group \ --rule-group-arn arn:aws:network-fi rewall:us-east-2:1 23457890:stateful- rulegroup/custom \ --output text --query UpdateToken`) </pre> <pre data-bbox="597 1045 1024 1522"> # aws network-firewall update-rule-group \ --rule-group-arn arn:aws:network-fi rewall:us-east-2:1 234567890:stateful- rulegroup/custom \ --rules file://cu stom.rules \ --update-token \$UPDATETOKEN </pre> <p>다음은 예시 출력입니다.</p> <pre data-bbox="597 1633 1024 1803"> { "UpdateToken": "7536939f-6a1d-414 c-96d1-bb28110996ed", </pre>	<p>AWS 시스템 관리자</p>

작업	설명	필요한 기술
	<pre> "RuleGroupResponse ": { "RuleGroupArn": "arn:aws:network-f irewall:us-east-2: 1234567890:stateful- rulegroup/custom", "RuleGrou pName": "custom", "RuleGroupId": "238a8259-9eaf-48b b-90af-5e690cf8c48b", "Type": "STATEFUL", "Capacity": 10, "RuleGrou pStatus": "ACTIVE", "Tags": [{ "Key": "environment", "Value": "development" }] } </pre>	

업데이트된 경고 기능 테스트

작업	설명	필요한 기술
<p>테스트용 경고를 생성합니다.</p>	<ol style="list-style-type: none"> 방화벽 서브넷 내의 테스트 워크스테이션에 로그인합니다. 경고를 생성하는 명령을 입력합니다. 예를 들어 <code>curl</code>를 사용할 수 있습니다. 	<p>AWS 시스템 관리자</p>

작업	설명	필요한 기술
	<pre>curl -A "badstuff" http://www.amazon. com -o /dev/null</pre>	

작업	설명	필요한 기술
경고가 변경되었는지 확인합니다.	<ol style="list-style-type: none"> 1. https://console.aws.amazon.com/cloudwatch/ 에서 CloudWatch 콘솔을 엽니다. 2. 올바른 로그 그룹 및 스트림으로 이동합니다. 3. 로그 이벤트가 다음 예제와 유사한지 확인합니다. 예제는 경고 관련 부분만 보여줍니다. <pre data-bbox="634 699 1029 1692"> "alert": { "action": "allowed", "signature_id": 3, "rev": 2, "signature": "Watch out - Bad Stuff!!", "category": "Misc activity", "severity": 2, "metadata": { "custom-f ield": ["More Info"], "custom-f ield-2": ["Danger!"] } } </pre>	AWS 시스템 관리자

관련 리소스

참조

- [AWS Network Firewall에서 Slack 채널로 경고 전송](#)(AWS 권장 가이드)
- [AWS에서 Suricata를 이용한 위협 방지 규모 조정](#)(AWS 블로그 게시물)
- [AWS Network Firewall의 배포 모델](#)(AWS 블로그 게시물)
- [Suricata 메타 키웍스](#)(Suricata 설명서)

자습서 및 동영상

- [AWS Network Firewall 워크숍](#)

추가 정보

다음은 Suricata 5.0.2의 분류 구성 파일입니다. 이러한 분류는 방화벽 규칙을 만들 때 사용됩니다.

```
# config classification:shortname,short description,priority

config classification: not-suspicious,Not Suspicious Traffic,3
config classification: unknown,Unknown Traffic,3
config classification: bad-unknown,Potentially Bad Traffic, 2
config classification: attempted-recon,Attempted Information Leak,2
config classification: successful-recon-limited,Information Leak,2
config classification: successful-recon-largescale,Large Scale Information Leak,2
config classification: attempted-dos,Attempted Denial of Service,2
config classification: successful-dos,Denial of Service,2
config classification: attempted-user,Attempted User Privilege Gain,1
config classification: unsuccessful-user,Unsuccessful User Privilege Gain,1
config classification: successful-user,Successful User Privilege Gain,1
config classification: attempted-admin,Attempted Administrator Privilege Gain,1
config classification: successful-admin,Successful Administrator Privilege Gain,1

# NEW CLASSIFICATIONS
config classification: rpc-portmap-decode,Decode of an RPC Query,2
config classification: shellcode-detect,Executable code was detected,1
config classification: string-detect,A suspicious string was detected,3
config classification: suspicious-filename-detect,A suspicious filename was detected,2
config classification: suspicious-login,An attempted login using a suspicious username
was detected,2
```

```
config classification: system-call-detect,A system call was detected,2
config classification: tcp-connection,A TCP connection was detected,4
config classification: trojan-activity,A Network Trojan was detected, 1
config classification: unusual-client-port-connection,A client was using an unusual
  port,2
config classification: network-scan,Detection of a Network Scan,3
config classification: denial-of-service,Detection of a Denial of Service Attack,2
config classification: non-standard-protocol,Detection of a non-standard protocol or
  event,2
config classification: protocol-command-decode,Generic Protocol Command Decode,3
config classification: web-application-activity,access to a potentially vulnerable web
  application,2
config classification: web-application-attack,Web Application Attack,1
config classification: misc-activity,Misc activity,3
config classification: misc-attack,Misc Attack,2
config classification: icmp-event,Generic ICMP event,3
config classification: inappropriate-content,Inappropriate Content was Detected,1
config classification: policy-violation,Potential Corporate Privacy Violation,1
config classification: default-login-attempt,Attempt to login by a default username and
  password,2

# Update
config classification: targeted-activity,Targeted Malicious Activity was Detected,1
config classification: exploit-kit,Exploit Kit Activity Detected,1
config classification: external-ip-check,Device Retrieving External IP Address
  Detected,2
config classification: domain-c2,Domain Observed Used for C2 Detected,1
config classification: pup-activity,Possibly Unwanted Program Detected,2
config classification: credential-theft,Successful Credential Theft Detected,1
config classification: social-engineering,Possible Social Engineering Attempted,2
config classification: coin-mining,Crypto Currency Mining Activity Detected,2
config classification: command-and-control,Malware Command and Control Activity
  Detected,1
```

DNS 레코드를 Amazon Route 53 프라이빗 호스팅 영역으로 대량 마이그레이션합니다.

작성자: Ram Kandaswamy(AWS)

환경: 프로덕션

기술: 네트워킹, 클라우드 네이티브, 인프라 DevOps

AWS 서비스: AWS Cloud9, Amazon Route 53, Amazon S3

요약

네트워크 엔지니어와 클라우드 관리자는 Amazon Route 53의 프라이빗 호스팅 영역에 도메인 이름 시스템(DNS) 레코드를 추가할 수 있는 효율적이고 간단한 방법이 필요합니다. 수동 접근 방식을 사용하여 Microsoft Excel 워크시트의 항목을 Route 53 콘솔의 적절한 위치로 복사하는 것은 번거롭고 오류가 발생하기 쉽습니다. 이 패턴은 여러 레코드를 추가하는 데 필요한 시간과 노력을 줄이는 자동화된 접근 방식을 설명합니다. 또한 여러 호스팅 영역을 생성할 수 있는 반복 가능한 일련의 단계를 제공합니다.

이 패턴은 개발 및 테스트에는 AWS Cloud9 통합 개발 환경(IDE)을 사용하고 레코드를 저장하기 위해 Amazon Simple Storage Service(S3)를 사용합니다. 데이터를 효율적으로 처리하기 위해 패턴은 단순하고 Python 사전 (dict 데이터 유형) 을 지원하는 기능 때문에 JSON 형식을 사용합니다.

참고: 시스템에서 영역 파일을 생성할 수 있다면 [Route 53 가져오기 기능을](#) 대신 사용해 보십시오.

사전 조건 및 제한 사항

사전 조건

- 프라이빗 호스팅 영역 레코드가 포함된 Excel 워크시트
- A 레코드, NAPTR (이름 기관 포인터) 레코드, SRV 레코드와 같은 다양한 유형의 DNS 레코드에 대한 지식 ([지원되는 DNS 레코드 유형](#) 참조)
- Python 언어 및 해당 라이브러리에 대한 지식

제한 사항

- 이 패턴은 모든 사용 사례 시나리오에 대한 광범위한 범위를 제공하지는 않습니다. 예를 들어 [change_resource_record_sets](#) 호출은 API의 사용 가능한 모든 속성을 사용하지 않습니다.

- Excel 워크시트에서는 각 행의 값이 고유한 것으로 간주됩니다. 각 FQDN(Fully Qualified Domain Name)에 대한 여러 값이 동일한 행에 표시될 것으로 예상됩니다. 그렇지 않은 경우 이 패턴에 제공된 코드를 수정하여 필요한 연결을 수행해야 합니다.
- 이 패턴은 Python용 AWS SDK(Boto3)를 사용하여 Route 53 서비스를 직접 호출합니다. `create_stack` 및 `update_stack` 명령에 AWS CloudFormation 래퍼를 사용하고 JSON 값을 사용하여 템플릿 리소스를 채우도록 코드를 개선할 수 있습니다.

아키텍처

기술 스택

- 트래픽 라우팅을 위한 Route 53 프라이빗 호스팅 영역
- 개발 및 테스트를 위한 AWS Cloud9 IDE
- 출력 JSON 파일을 저장하기 위한 Amazon S3

워크플로는 이전 다이어그램에 나와 있고 에픽 섹션에서 설명한 대로 다음 단계로 구성됩니다.

1. 레코드 세트 정보가 있는 Excel 워크시트를 S3 버킷에 업로드합니다.
2. Excel 데이터를 JSON 형식으로 변환하는 Python 스크립트를 만들고 실행합니다.
3. S3 버킷에서 레코드를 읽고 데이터를 정리합니다.
4. 프라이빗 호스팅 영역에서 레코드 세트를 생성합니다.

도구

- [Route 53](#) — Amazon Route 53은 도메인 등록, DNS 라우팅 및 상태 확인을 처리하는 가용성과 확장성이 뛰어난 DNS 웹 서비스입니다.
- [AWS Cloud9](#) – AWS Cloud9은 여러 프로그래밍 언어와 런타임 디버거, 내장 터미널을 지원하여 풍부한 코드 편집 환경을 제공하는 IDE입니다. 소프트웨어를 코딩, 빌드, 실행, 테스트 및 디버깅하기 위한 도구 모음을 갖추고 있으며 클라우드에 소프트웨어를 릴리스하는 데에도 도움이 됩니다.
- [Amazon S3](#)-Amazon Simple Storage Service(S3)는 객체 스토리지 서비스입니다. Amazon S3를 사용하면 인터넷을 통해 언제 어디서든 원하는 양의 데이터를 저장하고 검색할 수 있습니다.

에픽

자동화를 위한 데이터 준비

작업	설명	필요한 기술
<p>기록을 위한 Excel 파일을 만드세요.</p>	<p>현재 시스템에서 내보낸 레코드를 사용하여 FQDN (정규화된 도메인 이름), 레코드 유형, TTL (TTL), 값 등 레코드에 필요한 열이 있는 Excel 워크시트를 만드십시오. NAPTR 및 SRV 레코드의 경우 값은 여러 속성의 조합이므로 Excel의 concat 방법을 사용하여 이러한 속성을 결합하십시오.</p> <pre data-bbox="613 976 998 1234"> Fqdn\ Record\ Value\ TTL ----- somet A 1.1.1.1 900 .exam org </pre>	<p>데이터 엔지니어, 엑셀 스킬</p>
<p>작업 환경을 확인하세요.</p>	<p>AWS Cloud9 IDE에서 Python 파일을 생성하여 엑셀 입력 워크시트를 JSON 형식으로 변환합니다. (AWS Cloud9 대신 SageMaker Amazon 노트북을 사용하여 Python 코드로 작업할 수도 있습니다.)</p> <p>사용 중인 Python 버전이 버전 3.7 이상인지 확인하십시오.</p> <pre data-bbox="597 1787 1027 1856" style="border: 1px solid #ccc; border-radius: 10px; padding: 5px;"> python3 --version </pre>	<p>일반 AWS</p>

작업	설명	필요한 기술
	<p>pandas 패키지를 설치합니다.</p> <pre>pip3 install pandas --user</pre>	
엑셀 워크시트 데이터를 JSON으로 변환합니다.	<p>Excel에서 JSON으로 변환하는 다음 코드가 포함된 Python 파일을 생성합니다.</p> <pre>import pandas as pd data=pd.read_excel('./Book1.xls') data.to_json(path_or_buf='my.json', orient='records')</pre> <p>여기서 Book1은(는) Excel 워크시트의 이름이고 my.json은(는) 출력 JSON 파일의 이름입니다.</p>	데이터 엔지니어, Python 스킬
이 JSON 파일을 S3 버킷에 업로드합니다.	<p>my.json 파일을 S3 버킷에 업로드합니다. 자세한 내용은 Amazon S3 설명서의 버킷 생성을 참조하세요.</p>	앱 개발자

레코드 삽입

작업	설명	필요한 기술
프라이빗 호스팅 영역을 생성합니다.	<p>create_hosted_zone API 와 다음 Python 샘플 코드를 사용하여 프라이빗 호스팅 영역을 생성합니다. 파라미터 hostedZoneName ,</p>	클라우드 아키텍트, 네트워크 관리자, Python 스킬

작업	설명	필요한 기술
	<p>vpcRegion 및 vpcId를 사용자의 값으로 바꿉니다.</p> <pre data-bbox="594 331 1027 1724"> import boto3 import random hostedZoneName = "xxx" vpcRegion = "us-east-1" vpcId="vpc-xxxx" route53_client = boto3.client('route53') response = route53_client.create_hosted_zone(Name= hostedZoneName, VPC={ 'VPCRegion': vpcRegion, 'VPCId': vpcId }, CallerReference=str(random.random()*1000000), HostedZoneConfig={ 'Comment': "private hosted zone created by automation", 'PrivateZone': True }) print(response) </pre> <p>또한 CloudFormation AWS와 같은 IaC (코드형 인프라) 도구</p>	

작업	설명	필요한 기술
	<p>를 사용하여 이러한 단계를 적절한 리소스 및 속성으로 스택을 생성하는 템플릿으로 대체할 수 있습니다.</p>	
<p>Amazon S3에서 세부 정보를 사전으로 검색합니다.</p>	<p>다음 코드를 사용하여 S3 버킷에서 읽고 JSON 값을 Python 사전으로 가져옵니다.</p> <pre data-bbox="597 604 1026 1199"> fileobj = s3_client .get_object(Bucket=bu cket_name, Key='my.json') filedata = fileobj[' Body'].read() contents = filedata. decode('utf-8') json_content=json. loads(contents) print(json_content) </pre> <p>json_content 은(는) Python 사전이 들어 있습니다.</p>	<p>앱 개발자, Python 기술</p>

작업	설명	필요한 기술
공백 및 유니코드 문자의 데이터 값을 정리합니다.	<p>데이터의 정확성을 보장하기 위한 안전 조치로 다음 코드를 사용하여 <code>json_content</code> 의 값에 대해 제거 작업을 수행하십시오. 이 코드는 각 문자열의 앞과 끝에 있는 공백 문자를 제거합니다. 또한 <code>replace</code> 메서드를 사용하여 고정된 (끊어지지 않는) 공백 (<code>\xa0</code> 문자) 을 제거합니다.</p> <pre data-bbox="597 730 1026 1444"> for item in json_content: fqdn_name = unicodedata.normalize("NFKD", item["FqdnName"]).replace("u", "").replace('\xa0', '').strip() rec_type = item["RecordType"].replace('\xa0', '').strip() res_rec = { 'Value': item["Value"].replace('\xa0', '').strip() } </pre>	앱 개발자, Python 기술

작업	설명	필요한 기술
레코드 삽입	<p>이전 for 루프의 일부로 다음 코드를 사용합니다.</p> <pre data-bbox="597 348 1024 1734"> change_response = route53_client.change_resource_record_sets(HostedZoneId="xxxxxxxx", ChangeBatch={ 'Comment': 'Created by automation', 'Changes': [{ 'Action': 'UPSERT', 'ResourceRecordSet': { 'Name': fqdn_name, 'Type': rec_type, 'TTL': item["TTL"], 'ResourceRecords': res_rec } }] }) </pre>	앱 개발자, Python 기술

작업	설명	필요한 기술
	이 에픽의 첫 번째 단계에 있는 호스팅 영역 xxxxxxxx ID는 어디에 있습니까?	

관련 리소스

참조

- [영역 파일을 가져와서 레코드 생성](#) (Amazon Route 53 설명서)
- [create_hosted_zone 메서드](#) (Boto3 설명서)
- [변경 리소스 레코드 세트 메서드](#) (Boto3 설명서)

자습서 및 동영상

- [파이썬 튜토리얼](#) (파이썬 문서)
- [Amazon Route 53을 사용한 DNS 설계](#) (YouTube 동영상, AWS 온라인 테크 토크)

F5에서 AWS의 Application Load Balancer로 마이그레이션할 때 HTTP 헤더를 수정

작성자: Sachin Trivedi(AWS)

환경: PoC 또는 파일럿	소스: 온프레미스	대상: AWS 클라우드
R 유형: 리플랫폼	워크로드: 기타 모든 워크로드	기술: 네트워킹, 하이브리드 클라우드, 마이그레이션

AWS 서비스: 아마존 CloudFront, 엘라스틱 로드 밸런싱 (ELB), AWS Lambda

요약

F5 로드 밸런서를 사용하는 애플리케이션을 Amazon Web Services(AWS)로 마이그레이션하고 AWS에서 Application Load Balancer를 사용하려는 경우, 헤더 수정을 위한 F5 규칙을 마이그레이션하는 것은 일반적으로 발생하는 문제입니다. Application Load Balancer는 헤더 수정을 지원하지 않지만 Amazon을 CDN (콘텐츠 전송 네트워크) CloudFront 으로 사용하고 Lambda @Edge 를 사용하여 헤더를 수정할 수 있습니다.

이 패턴은 필요한 통합을 설명하고 AWS CloudFront 및 Lambda @Edge 를 사용하여 헤더를 수정하기 위한 샘플 코드를 제공합니다.

사전 조건 및 제한 사항

사전 조건

- `if`, `else`를 사용하여 HTTP 헤더 값을 대체하는 구성을 갖춘 F5 로드 밸런서를 사용하는 온프레미스 애플리케이션입니다. 이 구성에 대한 자세한 내용은 F5 제품 설명서의 [HTTP::header](#)를 참조하세요.

제한 사항

- 이 패턴은 F5 로드 밸런서 헤더 사용자 지정에 적용됩니다. 다른 타사 로드 밸런서의 경우 해당 로드 밸런서 설명서에서 지원 정보를 확인하세요.

- Lambda@Edge에 사용하는 Lambda 함수는 미국 동부(버지니아 북부) 리전에 있어야 합니다.

아키텍처

다음 다이어그램은 CDN과 다른 AWS 구성 요소 간의 통합 흐름을 포함하여 AWS의 아키텍처를 보여 줍니다.

도구

서비스

- [Application Load Balancer](#) – Application Load Balancer는 오픈 시스템 상호 연결(OSI) 모델의 일곱 번째 계층에서 작동하는 AWS의 완전 관리형 로드 밸런싱 서비스입니다. 여러 대상에 걸쳐 트래픽을 분산하고 HTTP 헤더 및 메서드, 쿼리 문자열, 호스트 기반 또는 경로 기반 라우팅을 기반으로 고급 라우팅 요청을 지원합니다.
- [Amazon CloudFront](#) — CloudFront Amazon은.html, .css, .js, 이미지 파일과 같은 정적 및 동적 웹 콘텐츠를 사용자에게 빠르게 배포하는 웹 서비스입니다. CloudFront 지연 시간을 줄이고 성능을 향상 시키기 위해 엣지 로케이션이라고 하는 전 세계 데이터 센터 네트워크를 통해 콘텐츠를 전송합니다.
- [Lambda @Edge](#) – Lambda @Edge 는 함수를 실행하여 제공하는 콘텐츠를 사용자 지정할 수 있게 해주는 AWS Lambda의 확장 프로그램입니다. CloudFront 미국 동부 (버지니아 북부) 지역에서 함수를 작성한 다음 이 함수를 CloudFront 배포와 연결하여 서버를 프로비저닝하거나 관리하지 않고도 코드를 전 세계에 자동으로 복제할 수 있습니다. 이렇게 하면 지연 시간이 줄어들고 사용자 경험이 향상됩니다.

코드

다음 샘플 코드는 응답 헤더를 수정하기 위한 청사진을 제공합니다. CloudFront 에픽 섹션의 지침에 따라 코드를 배포하세요.

```
exports.handler = async (event, context) => {
  const response = event.Records[0].cf.response;
  const headers = response.headers;

  const headerNameSrc = 'content-security-policy';
  const headerNameValue = '*.xyz.com';
```

```

if (headers[headerNameSrc.toLowerCase()]) {
  headers[headerNameSrc.toLowerCase()] = [{
    key: headerNameSrc,
    value: headerNameValue,
  }];
  console.log(`Response header "${headerNameSrc}" was set to ` +
    `${headers[headerNameSrc.toLowerCase()][0].value}`);
}
else {
  headers[headerNameSrc.toLowerCase()] = [{
    key: headerNameSrc,
    value: headerNameValue,
  }];
}
return response;
};

```

에픽

CDN 배포 만들기

작업	설명	필요한 기술
<p>웹 배포를 만드세요. CloudFront</p>	<p>이 단계에서는 CloudFront 배포를 생성하여 콘텐츠를 전송할 CloudFront 위치를 지정하고 콘텐츠 전달을 추적하고 관리하는 방법에 대한 세부 정보를 제공합니다.</p> <p>콘솔을 사용하여 배포를 생성하려면 AWS Management Console에 로그인하고 CloudFront 콘솔을 연 다음 CloudFront 설명서의 단계를 따르십시오.</p>	클라우드 관리자

Lambda@Edge 함수 생성 및 배포

작업	설명	필요한 기술
<p>Lambda@Edge 함수를 생성하고 배포합니다.</p>	<p>응답 헤더를 수정하기 위한 블루프린트를 사용하여 Lambda@Edge 함수를 생성할 수 있습니다. CloudFront (사용 사례별로 다른 블루프린트를 사용할 수 있습니다. 자세한 내용은 설명서의 Lambda @Edge 예제 함수를 참조하십시오.)</p> <p>CloudFront</p> <p>Lambda@Edge 함수를 생성하려면:</p> <ol style="list-style-type: none"> 1. AWS Management Console에 로그인하여 https://console.aws.amazon.com/lambda/에서 AWS Lambda 콘솔을 엽니다. 2. 미국 동부 (버지니아 북부) 지역에 있는지 확인하십시오. CloudFront 블루프린트는 이 지역에서만 사용할 수 있습니다. 3. 함수 생성을 선택합니다. 4. 청사진 사용을 선택한 다음 청사진 검색 필드에 cloudfront를 입력합니다. 5. cloudfront-modify-response-header블루프린트를 선택한 다음 구성을 선택합니다. 6. 기본 정보 페이지에서 다음 정보를 입력합니다. 	<p>AWS 관리자</p>

작업	설명	필요한 기술
	<ol style="list-style-type: none"> a. 함수 이름을 입력합니다. b. 실행 역할에서 AWS 정책 템플릿에서 새 역할 생성을 선택합니다. c. 필수 AWS Identity and Access Management(IAM) 역할 이름을 연결합니다. <ol style="list-style-type: none"> 7. 함수 생성을 선택합니다. 8. 해당 페이지의 디자이너 섹션에서 함수 이름을 선택합니다. 9. 함수 코드 섹션에서 템플릿 코드를 코드 섹션에 있는 이전에 이 패턴에 제공된 샘플 코드로 교체합니다. 10. 샘플 코드에서 xyz.com을 사용자 도메인 이름으로 바꿉니다. 11. 저장을 선택합니다. 	
Lambda@Edge 함수를 배포합니다.	Amazon 설명서의 4단계 : CloudFront Amazon 설명서에 있는 간단한 Lambda@Edge 함수 생성의 지침에 따라 트리거를 구성하고 함수를 CloudFront 배포하십시오.	AWS 관리자

관련 리소스

CloudFront 설명서

- [사용자 지정 오리진에 대한 요청 및 응답 동작](#)

- [배포 작업](#)
- [Lambda@Edge 예제 함수](#)
- [Lambda@Edge를 사용하여 엣지에서 사용자 지정](#)
- [자습서: 간단한 Lambda@Edge 함수 생성](#)

다중 VPC에서 중앙 AWS 서비스 엔드포인트에 비공개로 액세스

작성자: 마틴 쿼트너(AWS)와 사무엘 고든(AWS)

코드 리포지토리: [VPC 엔드포인트 공유](#)

환경: 프로덕션

기술: 네트워킹, 인프라

AWS 서비스: AWS RAM, 아마존 Route 53, 아마존 SNS, AWS Transit Gateway, 아마존 VPC

요약

사용자 환경의 보안 및 규정 준수 요구 사항에는 Amazon Web Services(AWS) 서비스 또는 엔드포인트로 향하는 트래픽이 퍼블릭 인터넷을 통과해서는 안 된다고 명시되어 있을 수도 있습니다. 이 패턴은 중앙 허브 VPC가 여러 분산 스포크 VPC에 연결되는 hub-and-spoke 토폴로지에 맞게 설계된 솔루션입니다. 이 솔루션에서는 PrivateLink AWS를 사용하여 허브 계정에서 AWS 서비스에 대한 인터페이스 VPC 엔드포인트를 생성합니다. 그런 다음 전송 게이트웨이와 분산 도메인 이름 시스템(DNS) 규칙을 사용하여 연결된 VPC에 걸쳐 엔드포인트의 프라이빗 IP 주소에 대한 요청을 해결합니다.

이 패턴은 연결된 VPC의 리소스에서 DNS 쿼리를 해결하기 위해 AWS Transit Gateway, 인바운드 Amazon Route 53 Resolver 엔드포인트, 공유된 Route 53 전달 규칙을 사용하는 방법을 설명합니다. 허브 계정에서 엔드포인트, 전송 게이트웨이, Resolver 및 전달 규칙을 생성합니다. 그런 다음 AWS Resource Access Manager(AWS RAM)를 사용하여 전송 게이트웨이 및 전달 규칙을 스포크 VPC와 공유합니다. 제공된 AWS CloudFormation 템플릿은 허브 VPC 및 스포크 VPC에서 리소스를 배포하고 구성하는 데 도움이 됩니다.

사전 조건 및 제한 사항

사전 조건

- AWS Organizations의 동일한 조직에서 관리되는 허브 계정과 하나 이상의 스포크 계정입니다. 자세한 내용은 [조직 생성 및 관리](#)를 참조하세요.
- AWS Resource Access Manager(AWS RAM)는 AWS Organizations에서 신뢰할 수 있는 서비스로 구성되어 있습니다. 자세한 내용은 [기타 AWS 서비스와 AWS Organizations 사용](#)을 참조하세요.

- 허브 및 스포크 VPC에서 DNS 확인을 활성화해야 합니다. 자세한 내용은 [VPC의 DNS 속성](#)(Amazon Virtual Private Cloud 설명서)를 참조하세요.

제한 사항

- 이 패턴은 동일한 AWS 리전의 허브 계정과 스포크 계정을 연결합니다. 다중 리전 배포의 경우 각 리전에 대해 이 패턴을 반복해야 합니다.
- AWS 서비스는 인터페이스 VPC PrivateLink 엔드포인트로 통합되어야 합니다. 전체 목록은 [AWS와 통합되는 AWS 서비스 PrivateLink \(PrivateLink 설명서\)](#) 를 참조하십시오.
- 가용 영역 친화성은 보장되지 않습니다. 예를 들어 가용 영역 A의 쿼리는 가용 영역 B의 IP 주소로 응답할 수 있습니다.
- VPC 엔드포인트와 연결된 Elastic Network 인터페이스의 쿼리는 초당 10,000개로 제한됩니다.

아키텍처

대상 기술 스택

- 허브 AWS 계정의 허브 VPC
- 스포크 AWS 계정에 있는 하나 이상의 스포크 VPC
- 허브 계정에 있는 하나 이상의 인터페이스 VPC 엔드포인트
- 허브 계정의 인바운드 및 아웃바운드 Route 53 Resolver
- 허브 계정에 배포되고 스포크 계정과 공유되는 Route 53 Resolver 전달 규칙
- 허브 계정에 배포되고 스포크 계정과 공유되는 전송 게이트웨이
- 허브와 스포크 VPC를 연결하는 AWS Transit Gateway

대상 아키텍처

다음 이미지는 이 솔루션의 샘플 아키텍처를 보여줍니다. 이 아키텍처에서 허브 계정의 Route 53 Resolver 전달 규칙은 다른 아키텍처 구성 요소와 다음과 같은 관계를 갖습니다.

1. 전달 규칙은 AWS RAM을 사용하여 스포크 VPC와 공유됩니다.
2. 전달 규칙은 허브 VPC의 아웃바운드 Resolver와 연결됩니다.
3. 전달 규칙은 허브 VPC의 인바운드 Resolver를 대상으로 합니다.

다음 이미지는 샘플 아키텍처를 통한 트래픽 흐름을 보여줍니다.

1. 스포크 VPC에 있는 Amazon Elastic Compute Cloud(Amazon EC2) 인스턴스 같은 리소스가 `<service>.<region>.amazonaws.com`으로 DNS 요청을 보냅니다. 요청은 Amazon DNS Resolver에 의해 수신됩니다.
2. 허브 계정에서 공유되고 스포크 VPC와 연결된 Route 53 전달 규칙이 요청을 가로챍니다.
3. 허브 VPC에서 아웃바운드 Resolver는 전달 규칙을 사용하여 요청을 인바운드 Resolver로 전달합니다.
4. 인바운드 Resolver는 허브 VPC Amazon DNS Resolver를 사용하여 VPC 엔드포인트의 프라이빗 IP 주소로 `<service>.<region>.amazonaws.com`의 IP 주소를 확인합니다. VPC 엔드포인트가 없는 경우 퍼블릭 IP 주소로 확인합니다.

도구

AWS 도구 및 서비스

- [AWS는 AWS CloudFormation](#) 리소스를 설정하고, 빠르고 일관되게 프로비저닝하고, AWS 계정 및 지역 전반의 수명 주기 전반에 걸쳐 리소스를 관리할 수 있도록 지원합니다.
- [Amazon Elastic Compute Cloud\(Amazon EC2\)](#)는 클라우드에서 확장 가능한 컴퓨팅 용량을 제공합니다. 필요한 만큼 많은 가상 서버를 시작하고 빠르게 규모를 확장하거나 축소할 수 있습니다.
- [AWS Identity and Access Management\(IAM\)](#)를 사용하면 AWS 리소스를 사용하도록 인증받고 권한이 부여된 사용자를 통제함으로써 AWS 리소스에 대한 액세스를 안전하게 관리할 수 있습니다.
- [AWS Resource Access Manager\(AWS RAM\)](#)는 AWS 계정에 걸쳐 리소스를 안전하게 공유하여 운영 오버헤드를 줄이고 가시성과 감사 가능성을 제공하도록 도와줍니다.
- [Amazon Route 53](#)는 가용성과 확장성이 뛰어난 DNS(도메인 이름 시스템) 웹 서비스입니다.
- [AWS Systems Manager](#)는 AWS 클라우드에서 실행되는 애플리케이션과 인프라를 관리하는 데 도움이 됩니다. 애플리케이션 및 리소스 관리를 간소화하고, 운영 문제의 감지 및 해결 시간을 단축하며, AWS 리소스를 규모에 따라 안전하게 관리하는 데 도움이 됩니다.
- [AWS Transit Gateway](#)는 VPC와 온프레미스 네트워크를 연결하는 중앙 허브입니다.
- [Amazon Virtual Private Cloud\(VPC\)](#)를 이용하면 사용자가 정의한 가상 네트워크로 AWS 리소스를 시작할 수 있습니다. 이 가상 네트워크는 사용자의 자체 데이터 센터에서 운영하는 기존 네트워크와 유사하며 AWS의 확장 가능한 인프라를 사용한다는 이점이 있습니다.

기타 도구 및 서비스

- [nslookup](#)은 DNS 레코드를 쿼리하는 데 사용되는 명령줄 도구입니다. 이 패턴에서는 이 도구를 사용하여 솔루션을 테스트합니다.

코드 리포지토리

이 패턴의 코드는 [vpc-endpoint-sharing](#) 리포지토리에서 GitHub 사용할 수 있습니다. 이 패턴은 두 개의 AWS CloudFormation 템플릿을 제공합니다.

- 허브 계정에 다음 리소스를 배포하기 위한 템플릿:
 - rSecurityGroupEndpoints - VPC 엔드포인트에 대한 액세스를 제어하는 보안 그룹입니다.
 - rSecurityGroupResolvers - Route 53 Resolver에 대한 액세스를 제어하는 보안 그룹입니다.
 - rKMSEndpoint, rSSMMessagesEndpoint, rSSMEndpoint 및 rEC2MessagesEndpoint - 허브 계정의 인터페이스 VPC 엔드포인트 예제입니다. 사용 사례에 맞게 이러한 엔드포인트를 사용자 지정합니다.
 - rInboundResolver - 허브 Amazon DNS Resolver에 대한 DNS 쿼리를 확인하는 Route 53 Resolver입니다.
 - rOutboundResolver - 쿼리를 인바운드 Resolver로 전달하는 아웃바운드 Route 53 Resolver입니다.
 - rAWSApiResolverRule - 모든 스포크 VPC와 공유되는 Route 53 Resolver 전달 규칙입니다.
 - rRamShareAWSResolverRule - 스포크 VPC가 rAWSApiResolverRule 전달 규칙을 사용하도록 허용하는 AWS RAM 공유입니다.
 - *rVPC - 공유 서비스를 모델링하는 데 사용되는 허브 VPC입니다.
 - *rSubnet1 - 허브 리소스를 보관하는 데 사용되는 프라이빗 서브넷입니다.
 - *rRouteTable1 - 허브 VPC의 라우팅 테이블입니다.
 - *rRouteTableAssociation1 - 허브 VPC의 rRouteTable1 라우팅 테이블의 경우, 프라이빗 서브넷에 대한 연결입니다.
 - *rRouteSpoke - 허브 VPC에서 스포크 VPC로의 경로입니다.
 - *rTgw - 모든 스포크 VPC와 공유되는 전송 게이트웨이입니다.
 - *rTgwAttach - 허브 VPC가 트래픽을 rTgw 전송 게이트웨이로 라우팅할 수 있도록 하는 연결입니다.
 - *rTgwShare - 스포크 계정이 rTgw 전송 게이트웨이를 사용할 수 있도록 허용하는 AWS RAM 공유입니다.

- 스포크 계정에 다음 리소스를 배포하기 위한 템플릿:
 - `rAWSApiResolverRuleAssociation` - 스포크 VPC가 허브 계정의 공유 전달 규칙을 사용할 수 있도록 하는 연결입니다.
 - `*rVPC` - 스포크 VPC입니다.
 - `*rSubnet1, rSubnet2, rSubnet3` - 스포크 프라이빗 리소스를 수용하는 데 사용되는 각 가용 영역의 서브넷입니다.
 - `*rTgwAttach` - 스포크 VPC가 트래픽을 `rTgw` 전송 게이트웨이로 라우팅할 수 있도록 하는 연결입니다.
 - `*rRouteTable1` - 스포크 VPC의 라우팅 테이블입니다.
 - `*rRouteEndpoints` - 스포크 VPC의 리소스에서 전송 게이트웨이까지의 경로입니다.
 - `*rRouteTableAssociation1/2/3` - 스포크 VPC의 `rRouteTable1` 라우팅 테이블의 경우 프라이빗 서브넷에 대한 연결입니다.
 - `*rInstanceRole` - 솔루션을 테스트하는 데 사용되는 IAM 역할입니다.
 - `*rInstancePolicy` - 솔루션 테스트에 사용되는 IAM 정책입니다.
 - `*rInstanceSg` - 솔루션을 테스트하는 데 사용되는 보안 그룹입니다.
 - `*rInstanceProfile` - 솔루션을 테스트하는 데 사용되는 IAM 인스턴스 프로파일입니다.
 - `*rInstance` - AWS Systems Manager를 통해 액세스할 수 있도록 사전 구성된 EC2 인스턴스입니다. 이 인스턴스를 사용하여 솔루션을 테스트합니다.

* 이러한 리소스는 샘플 아키텍처를 지원하며 기존 랜딩 존에서 이 패턴을 구현할 때는 필요하지 않을 수 있습니다.

에픽

CloudFormation 템플릿 준비

작업	설명	필요한 기술
코드 리포지토리를 복제합니다.	<ol style="list-style-type: none"> 1. 명령줄 인터페이스에서 작업 디렉터리를 샘플 파일을 저장하고자 하는 위치로 변경합니다. 2. 다음 명령을 입력합니다. 	네트워크 관리자, 클라우드 아키텍트

작업	설명	필요한 기술
	<pre>git clone https://github.com/aws-samples/vpc-endpoint-sharing.git</pre>	
<p>템플릿을 수정합니다.</p>	<ol style="list-style-type: none"> 복제된 리포지토리에서 hub.yml 및 spoke.yml 파일을 엽니다. 이러한 템플릿으로 생성한 리소스를 검토하고 환경에 맞게 필요한 만큼 템플릿을 조정합니다. 전체 목록은 도구의 코드 리포지토리 섹션을 참조하세요. 계정에 이러한 리소스 중 일부가 이미 있는 경우 CloudFormation 템플릿에서 해당 리소스를 제거하세요. 자세한 내용은 템플릿 작업 (CloudFormation 문서) 을 참조하십시오. hub.yml 및 spoke.yml 파일을 저장하고 닫습니다. 	<p>네트워크 관리자, 클라우드 아키텍트</p>

대상 계정에 리소스 배포

작업	설명	필요한 기술
<p>허브 리소스를 배포합니다.</p>	<p>hub.yml 템플릿을 사용하여 스택을 생성합니다. CloudFormation 메시지가 표시되면 템플릿에 파라미터 값을 제공합니다. 자세한 내용은 스택 만</p>	<p>클라우드 아키텍트, 네트워크 관리자</p>

작업	설명	필요한 기술
	<p>들기 (문서) 를 참조하십시오.</p> <p>CloudFormation</p>	
스포크 리소스를 배포합니다.	<p>spoke.yml 템플릿을 사용하여 스택을 생성합니다.</p> <p>CloudFormation 메시지가 표시되면 템플릿에 파라미터 값을 제공합니다. 자세한 내용은 스택 만들기 (문서) 를 참조하십시오. CloudFormation</p>	클라우드 아키텍트, 네트워크 관리자

솔루션 테스트

작업	설명	필요한 기술
AWS 서비스에 대한 프라이빗 DNS 쿼리를 테스트합니다.	<ol style="list-style-type: none"> AWS Systems Manager의 기능인 Session Manager 를 사용하여 rInstance EC2 인스턴스에 연결합니다. 자세한 내용은 Session Manager를 사용하여 Linux 인스턴스에 연결(Amazon EC2 설명서)을 참조하세요. 허브 계정에 VPC 엔드포인트가 있는 AWS 서비스의 경우, nslookup을(를) 사용하여 인바운드 Route 53 Resolver의 프라이빗 IP 주소가 반환되는지 확인합니다. <p>다음은 nslookup을(를) 사용하여 Amazon Systems Manager 엔드포인트에 도달하는 예입니다.</p>	네트워크 관리자

작업	설명	필요한 기술
	<pre data-bbox="630 212 1029 327">nslookup ssm.<region>.amazonaws.com</pre> <p data-bbox="591 344 1029 709">3. AWS Command Line Interface(AWS CLI)에 변경 사항이 서비스 기능에 영향을 미치지 않았는지 확인하는 데 도움이 되는 명령을 입력합니다. 명령 목록은 AWS CLI 명령 참조를 참조하세요.</p> <p data-bbox="630 751 1016 932">예를 들어, 다음 명령은 Amazon Systems Manager 문서 목록을 반환해야 합니다.</p> <pre data-bbox="630 974 1029 1089">aws ssm list-documents</pre>	

작업	설명	필요한 기술
AWS 서비스에 대한 퍼블릭 DNS 쿼리를 테스트합니다.	<p>1. 허브 계정에 VPC 엔드포인트가 없는 AWS 서비스의 경우 nslookup을(를) 퍼블릭 IP 주소가 반환되는지 확인하는 데 사용합니다. 다음은 nslookup을(를) 사용하여 Amazon Simple Notification Service(Amazon SNS) 엔드포인트에 도달하는 예입니다.</p> <pre>nslookup sns.<region>.amazonaws.com</pre> <p>2. AWS CLI에서 변경 사항이 서비스 기능에 영향을 미치지 않았는지 확인하는 데 도움이 되는 명령을 입력합니다. 명령 목록은 AWS CLI 명령 참조를 참조하세요.</p> <p>예를 들어, 허브 계정에 Amazon SNS 주제가 있는 경우 다음 명령은 주제 목록을 반환해야 합니다.</p> <pre>aws sns list-topics</pre>	네트워크 관리자

관련 리소스

- [확장 가능하고 안전한 다중 VPC AWS 네트워크 인프라 구축](#) (AWS 백서)
- [공유 리소스로 작업하기](#) (AWS RAM 설명서)
- [트랜짓 게이트웨이 사용](#) (AWS Transit Gateway 설명서)

여러 AWS 계정에서 인바운드 인터넷 액세스에 대한 Network Access Analyzer 조사 결과 보고서 생성

작성자: Mike Virgilio

코드 리포지토리: [네트워크 액세스 분석기 다중 계정 분석](#)

환경: 프로덕션

기술: 네트워킹; 보안, 자격 증명, 규정 준수

AWS 서비스: AWS CloudFormation, 아마존 S3, 아마존 VPC, AWS 보안 허브

요약

AWS 리소스에 대한 의도하지 않은 인바운드 인터넷 액세스는 조직의 데이터 경계에 위험을 초래할 수 있습니다. [Network Access Analyzer](#)는 Amazon Web Services(AWS)의 리소스에 대한 의도하지 않은 네트워크 액세스를 식별하는 데 도움이 되는 Amazon Virtual Private Cloud(VPC)입니다. Network Access Analyzer를 사용하여 네트워크 액세스 요구 사항을 지정하고 지정된 요구 사항을 충족하지 않는 잠재적 네트워크 경로를 식별할 수 있습니다. Network Access Analyzer를 사용하여 다음을 수행할 수 있습니다.

1. 인터넷 게이트웨이를 통해 인터넷에 액세스할 수 있는 AWS 리소스를 식별합니다.
2. 프로덕션 및 개발 환경을 격리하고 트랜잭션 워크로드를 분리하는 등 virtual private cloud(VPC)가 적절하게 세그먼트화되어 있는지 확인합니다.

네트워크 액세스 분석기는 단일 구성 요소뿐만 아니라 end-to-end 네트워크 도달 가능성 조건을 분석합니다. Network Access Analyzer는 리소스가 인터넷에 액세스할 수 있는지 여부를 확인하기 위해 인터넷 게이트웨이, VPC 라우팅 테이블, 네트워크 액세스 제어 목록(ACL), 탄력적 네트워크 인터페이스의 퍼블릭 IP 주소, 보안 그룹을 평가합니다. 이러한 구성 요소 중 하나라도 인터넷 액세스를 방해하는 경우 Network Access Analyzer는 조사 결과를 생성하지 않습니다. 예를 들어 Amazon Elastic Compute Cloud(Amazon EC2) 인스턴스에 0/0(으)로부터 트래픽을 허용하는 개방형 보안 그룹이 있지만 해당 인스턴스가 인터넷 게이트웨이에서 라우팅할 수 없는 프라이빗 서브넷에 있는 경우 Network Access Analyzer는 조사 결과를 생성하지 않습니다. 이는 충실도가 높은 결과를 제공하므로 인터넷에서 실제로 액세스할 수 있는 리소스를 식별할 수 있습니다.

Network Access Analyzer를 실행할 때는 [네트워크 액세스 범위](#)를 사용하여 네트워크 액세스 요구 사항을 지정합니다. 이 솔루션은 인터넷 게이트웨이와 탄력적 네트워크 인터페이스 간의 네트워크 경로를 식별합니다. 이 패턴에서는 AWS Organizations에서 관리하는 조직의 중앙 집중식 AWS 계정에 솔루션을 배포하고, AWS 리전에 관계없이 조직 내 모든 계정을 분석합니다.

이 솔루션은 다음 사항을 염두에 두고 설계되었습니다.

- AWS CloudFormation 템플릿을 사용하면 이러한 패턴으로 AWS 리소스를 배포하는 데 필요한 노력을 줄일 수 있습니다.
- 배포 시 CloudFormation 템플릿과 naa-script.sh 스크립트의 파라미터를 조정하여 환경에 맞게 사용자 지정할 수 있습니다.
- Bash 스크립팅은 여러 계정의 네트워크 액세스 범위를 병렬로 자동으로 프로비저닝하고 분석합니다.
- Python 스크립트는 조사 결과를 처리하고 데이터를 추출한 다음 결과를 통합합니다. Network Access Analyzer 조사 결과에 대한 통합 보고서를 CSV 형식으로 검토하거나 Security Hub에서 검토할 수 있습니다. CSV 보고서의 예제는 이 패턴의 [추가 정보](#) 섹션에서 확인할 수 있습니다.
- 조사 결과의 문제를 해결하거나, naa-exclusions.csv 파일에 추가하여 향후 분석에서 제외할 수 있습니다.

사전 조건 및 제한 사항

사전 조건

- AWS Organizations에서 조직의 회원 계정으로 관리되는 보안 서비스와 도구를 호스팅하기 위한 AWS 계정 이 패턴에서는 이 계정을 보안 계정이라고 합니다.
- 보안 계정에는 아웃바운드 인터넷 액세스가 가능한 프라이빗 서브넷이 있어야 합니다. 지침은 Amazon VPC 설명서의 [서브넷 생성](#)을 참조하십시오. [NAT 게이트웨이](#) 또는 [인터페이스 VPC 엔드포인트](#)를 사용하여 인터넷 액세스를 설정할 수 있습니다.
- AWS Organizations 관리 계정 또는 관리자 권한을 위임받은 계정에 대한 CloudFormation 액세스 지침은 설명서의 [위임 관리자 등록](#)을 참조하십시오. CloudFormation
- AWS Organizations 및 간에 신뢰할 수 있는 액세스를 지원합니다 CloudFormation. 지침은 설명서에서 [AWS Organizations를 통한 신뢰할 수 있는 액세스 활성화](#)를 CloudFormation 참조하십시오.
- 검색 결과를 Security Hub에 업로드하는 경우 EC2 인스턴스가 프로비저닝되는 계정 및 AWS 리전에서 Security Hub를 활성화해야 합니다. 더 자세한 정보는 [AWS Security Hub 설정](#)을 참조하십시오.

제한 사항

- Network Access Analyzer 기능의 한계로 인해 교차 계정 네트워크 경로는 현재 분석되지 않습니다.
- 대상 AWS 계정은 AWS Organizations의 조직으로 관리되어야 합니다. AWS Organizations를 사용하지 않는 경우 해당 환경에 맞게 `naa-execrole.yaml` CloudFormation 템플릿과 `naa-script.sh` 스크립트를 업데이트할 수 있습니다. 대신에 스크립트를 실행할 곳의 AWS 계정 ID와 리전의 목록을 제공합니다.
- CloudFormation 템플릿은 아웃바운드 인터넷 액세스가 가능한 프라이빗 서브넷에 EC2 인스턴스를 배포하도록 설계되었습니다. Systems Manager Agent(SSM Agent)는 Systems Manager 서비스 엔드포인트에 도달하기 위한 아웃바운드 액세스가 필요하며, 코드 리포지토리를 복제하고 종속성을 설치하려면 아웃바운드 액세스가 필요합니다. 퍼블릭 서브넷을 사용하려면 `naa-resources.yaml` 템플릿을 수정하여 [탄력적 IP 주소](#)를 EC2 인스턴스에 연결해야 합니다.

아키텍처

대상 기술 스택

- Network Access Analyzer
- Amazon EC2 인스턴스
- Identity and Access Management(IAM) 역할
- Amazon Simple Storage Service(S3) 버킷
- Amazon Simple Notification Service(Amazon SNS) 주제
- AWS Security Hub(옵션 2만 해당)

대상 아키텍처

옵션 1: Amazon S3 버킷의 조사 결과에 액세스

이 다이어그램은 다음 프로세스를 보여줍니다.

1. 솔루션을 수동으로 실행하는 경우 사용자는 세션 관리자를 사용하여 EC2 인스턴스를 인증한 다음 `naa-script.sh` 스크립트를 실행합니다. 이 셸 스크립트는 2~7단계를 수행합니다.

솔루션을 자동으로 실행하는 경우 cron 표현식에서 정의한 일정에 따라 naa-script.sh 스크립트가 자동으로 시작됩니다. 이 셸 스크립트는 2~7단계를 수행합니다. 자세한 내용은 이 섹션 끝부분에 나와 있는 자동화 및 규모 조정을 참조하십시오.

2. EC2 인스턴스는 S3 버킷에서 최신 naa-exception.csv 파일을 다운로드합니다. 이 파일은 Python 스크립트가 제외를 처리할 때 프로세스 후반부에 사용됩니다.
3. EC2 인스턴스는 S3 버킷에 액세스하고 조직의 다른 계정에서 NAAExecRole IAM 역할을 떠맡는 권한을 부여하는 NAAEC2Role IAM 역할을 떠맡습니다.
4. EC2 인스턴스는 조직의 관리 계정에서 NAAExecRole IAM 역할을 떠맡고 조직의 계정 목록을 생성합니다.
5. EC2 인스턴스는 조직의 멤버 계정(아키텍처 다이어그램에서는 워크로드 계정이라고 함)에서 NAAExecRole IAM 역할을 맡아 각 계정에서 보안 평가를 수행합니다. 조사 결과는 EC2 인스턴스에 JSON 파일로 저장됩니다.
6. EC2 인스턴스는 Python 스크립트를 사용하여 JSON 파일을 처리하고, 데이터 필드를 추출하며, CSV 보고서를 생성합니다.
7. EC2 인스턴스는 CSV 파일을 S3 버킷에 업로드합니다.
8. Amazon EventBridge 규칙은 파일 업로드를 탐지하고 Amazon SNS 주제를 사용하여 사용자에게 보고서 작성을 알리는 이메일을 보냅니다.
9. 사용자는 S3 버킷에서 CSV 파일을 다운로드합니다. 사용자는 결과를 Excel 템플릿으로 가져와서 검토합니다.

옵션 2: AWS Security Hub의 조사 결과에 액세스

이 다이어그램은 다음 프로세스를 보여줍니다.

1. 솔루션을 수동으로 실행하는 경우 사용자는 세션 관리자를 사용하여 EC2 인스턴스를 인증한 다음 naa-script.sh 스크립트를 실행합니다. 이 셸 스크립트는 2~7단계를 수행합니다.

솔루션을 자동으로 실행하는 경우 cron 표현식에서 정의한 일정에 따라 naa-script.sh 스크립트가 자동으로 시작됩니다. 이 셸 스크립트는 2~7단계를 수행합니다. 자세한 내용은 이 섹션 끝부분에 나와 있는 자동화 및 규모 조정을 참조하십시오.

2. EC2 인스턴스는 S3 버킷에서 최신 naa-exception.csv 파일을 다운로드합니다. 이 파일은 Python 스크립트가 제외를 처리할 때 프로세스 후반부에 사용됩니다.

3. EC2 인스턴스는 S3 버킷에 액세스하고 조직의 다른 계정에서 NAAExecRole IAM 역할을 떠맡는 권한을 부여하는 NAAEC2Role IAM 역할을 떠맡습니다.
4. EC2 인스턴스는 조직의 관리 계정에서 NAAExecRole IAM 역할을 떠맡고 조직의 계정 목록을 생성합니다.
5. EC2 인스턴스는 조직의 멤버 계정(아키텍처 다이어그램에서는 워크로드 계정이라고 함)에서 NAAExecRole IAM 역할을 맡아 각 계정에서 보안 평가를 수행합니다. 조사 결과는 EC2 인스턴스에 JSON 파일로 저장됩니다.
6. EC2 인스턴스는 Python 스크립트를 사용하여 JSON 파일을 처리하고, Security Hub으로 가져오기 위한 데이터 필드를 추출합니다.
7. EC2 인스턴스는 Network Access Analyzer 조사 결과를 Security Hub에 가져옵니다.
8. Amazon EventBridge 규칙은 가져오기를 탐지하고 Amazon SNS 주제를 사용하여 사용자에게 프로세스가 완료되었음을 알리는 이메일을 보냅니다.
9. 사용자는 Security Hub에서 조사 결과를 확인합니다.

자동화 및 규모 조정

사용자 지정 일정에 따라 naa-script.sh 스크립트가 자동으로 실행되도록 이 솔루션을 예약할 수 있습니다. 사용자 지정 일정을 설정하려면 CloudFormation naa-resources.yaml 템플릿에서 파라미터를 수정하십시오. CronScheduleExpression 예를 들어 0 0 * * 0 기본값은 매주 일요일 자정에 솔루션을 실행합니다. 0 0 * 1-12 0 값은 매월 첫 번째 일요일 자정에 솔루션을 실행합니다. Cron 및 rate 표현식에 대한 자세한 내용은 Systems Manager 설명서의 [cron 및 rate 표현식](#)을 참조하십시오.

NAA-Resources 스택을 배포한 후 일정을 조정하려면 /etc/cron.d/naa-schedule에서 cron 일정을 수동으로 편집할 수 있습니다.

도구

서비스

- [Amazon Elastic Compute Cloud\(Amazon EC2\)](#)는 AWS 클라우드에서 확장 가능한 컴퓨팅 용량을 제공합니다. 필요한 만큼 가상 서버를 시작하고 빠르게 스케일 업하거나 스케일 다운할 수 있습니다.
- [EventBridgeAmazon](#)은 다양한 소스의 실시간 데이터와 애플리케이션을 연결하는 데 도움이 되는 서버리스 이벤트 버스 서비스입니다. AWS Lambda 함수, API 대상을 사용하는 HTTP 간접 호출 엔드포인트 또는 다른 AWS 계정의 이벤트 버스를 예로 들 수 있습니다.
- [Identity and Access Management\(IAM\)](#)는 사용자에 대한 인증 및 권한 부여를 제어함으로써 AWS 리소스에 대한 액세스를 안전하게 관리할 수 있습니다.

- [Organizations](#)는 사용자가 생성하고 중앙에서 관리하는 조직으로 여러 AWS 계정을 통합할 수 있는 계정 관리 서비스입니다.
- [AWS Security Hub](#)에서는 AWS의 보안 상태에 대한 포괄적인 보기가 제공됩니다. 이를 통해 AWS 환경에서 보안 업계 표준 및 모범 사례를 준수하는지 확인할 수도 있습니다.
- [Amazon Simple Notification Service\(Amazon SNS\)](#)를 사용하면 웹 서버 및 이메일 주소를 포함하여 게시자와 클라이언트 간의 메시지 교환을 조정하고 관리할 수 있습니다.
- [Amazon Simple Storage Service\(S3\)](#)는 원하는 양의 데이터를 저장, 보호 및 검색하는 데 도움이 되는 클라우드 기반 객체 스토리지 서비스입니다.
- [AWS Systems Manager](#)는 AWS 클라우드에서 실행되는 애플리케이션과 인프라를 관리하는 데 도움이 됩니다. 애플리케이션 및 리소스 관리를 간소화하고, 운영 문제의 감지 및 해결 시간을 단축하며, AWS 리소스를 규모에 따라 안전하게 관리하는 데 도움이 됩니다. 이 패턴은 Systems Manager의 기능인 Session Manger를 사용합니다.

코드 리포지토리

이 패턴의 코드는 GitHub [네트워크 액세스 분석기 다중 계정](#) 분석 리포지토리에서 사용할 수 있습니다. 코드 리포지토리에는 다음 파일이 포함되어 있습니다.

- `naa-script.sh` — 이 bash 스크립트는 여러 AWS 계정의 Network Access Analyzer 분석을 병렬로 시작하는 데 사용됩니다. `naa-resources.yaml` CloudFormation 템플릿에 정의된 대로 이 스크립트는 EC2 인스턴스의 폴더에 자동으로 배포됩니다. `/usr/local/naa`
- `naa-resources.yaml` — 이 CloudFormation 템플릿을 사용하여 조직의 보안 계정에 스택을 생성합니다. 이 템플릿은 솔루션을 지원하기 위해 이 계정에 필요한 모든 리소스를 배포합니다. 이 스택은 `naa-execrole.yaml` 템플릿보다 먼저 배포되어야 합니다.

참고: 이 스택을 삭제하고 재배포하는 경우 IAM 역할 간의 교차 계정 종속성을 다시 빌드하려면 `NAAExecRole` 스택 세트를 다시 빌드해야 합니다.

- `naa-execrole.yaml` — 이 CloudFormation 템플릿을 사용하여 관리 계정을 포함하여 조직의 모든 계정에 IAM 역할을 배포하는 스택 세트를 생성합니다. `NAAExecRole`
- `naa-processfindings.py` — `naa-script.sh` 스크립트는 이 Python 스크립트를 자동으로 호출하여 Network Access Analyzer JSON 출력을 처리하고 `naa-exclusions.csv` 파일에서 정상 작동이 확인된 리소스를 제외한 다음 통합 결과의 CSV 파일을 생성하거나 결과를 Security Hub로 가져옵니다.

에픽

배포 준비

작업	설명	필요한 기술
코드 리포지토리를 복제합니다.	<ol style="list-style-type: none"> 명령줄 인터페이스에서 작업 디렉터리를 샘플 파일을 저장하고자 하는 위치로 변경합니다. 다음 명령을 입력합니다. <pre>git clone https://github.com/aws-samples/network-access-analyzer-multi-account-analysis.git</pre> 	AWS DevOps
템플릿을 검토합니다.	<ol style="list-style-type: none"> 복제된 리포지토리에서 <code>naa-resources.yaml</code> 및 <code>naa-execrole.yaml</code> 파일을 엽니다. 이러한 템플릿으로 생성한 리소스를 검토하고 환경에 맞게 필요한 만큼 템플릿을 조정합니다. 자세한 내용은 CloudFormation 설명서의 템플릿 사용을 참조하십시오. <code>naa-resources.yaml</code> 및 <code>naa-execrole.yaml</code> 파일을 저장하고 닫습니다. 	AWS DevOps

CloudFormation 스택 생성

작업	설명	필요한 기술
<p>보안 계정에서 리소스를 프로 비저닝합니다.</p>	<p>naa-resources.yaml 템플릿 을 사용하여 보안 계정에 필 요한 모든 리소스를 배포하 는 CloudFormation 스택을 생 성합니다. 지침은 설명서의 스택 생성을 참조하십시오.</p> <p>CloudFormation 이 템플릿을 배포할 때는 다음 사항을 유념 합니다.</p> <ol style="list-style-type: none"> 1. 템플릿 지정 페이지에서 템 플릿 준비 완료를 선택한 다 음 naa-resources.yaml 파일 을 업로드합니다. 2. 스택 세부 정보 지정 페이지 의 스택 이름 상자에 NAA- Resources 를 입력합니 다. 3. Parameters(파라미터) 섹션 에서 다음을 입력합니다. <ul style="list-style-type: none"> • VPCId — 계정에서 VPC 를 선택합니다. • SubnetId — 인터넷에 액세스할 수 있는 프라이빗 서브넷을 선택합니다. <p>참고: 퍼블릭 서브넷을 선택하면 CloudFormation 템플릿이 기본적으로 엘라스틱 IP 주소를 프로 비저닝 및 연결하지 않기 때 문에 EC2 인스턴스에 퍼</p> 	<p>AWS DevOps</p>

작업	설명	필요한 기술
	<p>블릭 IP 주소가 할당되지 않을 수 있습니다.</p> <ul style="list-style-type: none"> • InstanceType — 기본 인스턴스 유형을 그대로 둡니다. • InstanceImageId - 기본값을 그대로 둡니다. • KeyPairName — 액세스에 SSH를 사용하는 경우 기존 키 페어의 이름을 지정합니다. • PermittedSSHInbound — 액세스에 SSH를 사용하는 경우 허용되는 CIDR 블록을 지정합니다. SSH를 사용하지 않는 경우 기본값인 127.0.0.1을 유지합니다. • BucketName - 기본값은 naa-<accountID>-<region> 입니다. 필요에 따라 이를 수정할 수 있습니다. 사용자 지정 값을 지정하는 경우 계정 ID와 리전이 지정된 값에 자동으로 추가됩니다. • EmailAddress — 분석 완료 시 Amazon SNS 알림을 받을 이메일 주소를 지정합니다. <p>참고: Amazon SNS 구독 구성은 분석을 완료하기 전에 확인해야 합니다. 그</p>	

작업	설명	필요한 기술
	<p>렇지 않으면 알림이 전송되지 않습니다.</p> <ul style="list-style-type: none"> • NAAEC2Role — 명명 규칙에 따라 이 IAM 역할에 다른 이름이 필요한 경우가 아니면 기본값을 유지합니다. • NAAExecRole — naa-execrole.yaml을 배포할 때 다른 이름을 사용하지 않는 한 기본값을 유지합니다. • Parallelism — 수행할 병렬 평가 수를 지정합니다. • Regions — 분석하려는 AWS 리전을 지정합니다. • ScopeNameValue — 범위에 할당할 태그를 지정합니다. 이 태그는 네트워크 액세스 범위를 결정하는 데 사용됩니다. • ExclusionFile — 제외 파일 이름을 지정합니다. 이 파일의 항목은 조사 결과에서 제외됩니다. • FindingsToCSV — 조사 결과를 CSV로 출력해야 할지 여부를 지정합니다. 허용되는 값은 true 및 false입니다. • FindingsToSecurityHub — 조사 결과를 	

작업	설명	필요한 기술
	<p>Security Hub로 가져와야 할지 여부를 지정합니다. 허용되는 값은 true 및 false입니다.</p> <ul style="list-style-type: none"> • EmailNotificationsForSecurityHub — 조사 결과를 Security Hub로 가져올 때 이메일 알림을 생성해야 할지 여부를 지정합니다. 허용되는 값은 true 및 false입니다. • ScheduledAnalysis — 솔루션이 일정에 따라 자동으로 실행되도록 하려면 true(을)를 입력한 다음 CronScheduleExpression 매개 변수에서 일정을 사용자 지정합니다. 솔루션을 자동으로 실행하지 않으려면 false(을)를 입력합니다. • CronScheduleExpression — 솔루션을 자동으로 실행하는 경우 cron 표현식을 입력하여 일정을 정의합니다. 자세한 내용은 이 패턴의 아키텍처 섹션의 자동화 및 규모 조정을 참조하십시오. <p>1. 검토 페이지에서 다음 리소스에 기능이 필요함: [AWS::IAM::Role] 를 선택한</p>	

작업	설명	필요한 기술
	<p>다음 [Create Stack] 을 선택합니다.</p> <p>2. 스택이 성공적으로 생성되면 CloudFormation 콘솔의 출력 탭에서 NAAEC2Role Amazon 리소스 이름 (ARN) 을 복사합니다. 나중에 naa-execrole.yaml 파일을 배포할 때 이 ARN을 사용합니다.</p>	

작업	설명	필요한 기술
<p>멤버 계정에서 IAM 역할을 프로비저닝합니다.</p>	<p>AWS Organizations 관리 계정 또는 관리자 권한이 위임된 계정에서 <code>naa-execrole.yaml</code> 템플릿을 사용하여 스택 세트를 생성합니다. CloudFormation CloudFormation 그러면 스택 세트는 조직의 모든 멤버 계정에서 <code>NAAExecRole</code> IAM 역할을 배포합니다. 지침은 설명서의 서비스 관리 권한이 포함된 스택 세트 생성을 참조하십시오. CloudFormation 이 템플릿을 배포할 때는 다음 사항을 유념합니다.</p> <ol style="list-style-type: none"> 1. 템플릿 준비에서 템플릿 준비 완료를 선택한 다음 <code>naa-execrole.yaml</code> 파일을 업로드합니다. 2. StackSet 세부 정보 지정 페이지에서 스택 세트의 이름을 지정합니다. <code>NAA-ExecRole</code> 3. Parameters(파라미터) 섹션에서 다음을 입력합니다. <ul style="list-style-type: none"> • <code>AuthorizedARN</code> — <code>NAA-Resources</code> 스택을 생성할 때 복사한 <code>NAAEC2Role</code> ARN을 입력합니다. • <code>NAARoleName</code> — <code>naa-resources.yaml</code> 파일을 배포할 때 다른 이름을 사용하지 않는 한 <code>NAAExecRo</code> 	<p>AWS DevOps</p>

작업	설명	필요한 기술
	<p>1e 기본값을 유지합니다.</p> <p>4. 권한에서 서비스 관리형 권한을 선택합니다.</p> <p>5. Set 배포 옵션 페이지에서, 배포 타겟에서, 조직에 배포하고 모든 기본값을 수용합니다.</p> <p>참고: 스택을 모든 멤버 계정에 동시에 배포하려면 최대 동시 계정 및 내결함성을 높은 값(예: 100)으로 설정합니다.</p> <p>6. 배포 리전에서 Network Access Analyzer용 EC2 인스턴스가 배포되는 리전을 선택합니다. IAM 리소스는 리전이 아닌 글로벌이므로 이를 통해 모든 활성 리전에 IAM 역할을 배포합니다.</p> <p>7. 검토 페이지에서 AWS가 사용자 지정 이름으로 IAM 리소스를 생성할 CloudFormation 수 있음을 인정합니다를 선택한 다음 [Create StackSet] 를 선택합니다.</p> <p>8. 스택 인스턴스 탭(개별 계정 상태)과 운영 탭(전체 상태)을 모니터링하여 배포가 완료되는 시점을 확인합니다.</p>	

작업	설명	필요한 기술
<p>관리 계정에서 IAM 역할을 프로비저닝합니다.</p>	<p>naa-execrole.yaml 템플릿을 사용하여 조직의 관리 계정에 NAAExecRole IAM 역할을 배포하는 CloudFormation 스택을 생성합니다. 이전에 생성한 스택 세트는 관리 계정에 IAM 역할을 배포하지 않습니다. 지침은 설명서의 스택 생성을 참조하십시오. CloudFormation 이 템플릿을 배포할 때는 다음 사항을 유념합니다.</p> <ol style="list-style-type: none"> 1. 템플릿 지정 페이지에서 템플릿 준비 완료를 선택한 다음 naa-resources.yaml 파일을 업로드합니다. 2. 스택 세부 정보 지정 페이지의 스택 이름 상자에 NAA-ExecRole 를 입력합니다. 3. Parameters(파라미터) 섹션에서 다음을 입력합니다. <ul style="list-style-type: none"> • AuthorizedARN — NAA-Resources 스택을 생성할 때 복사한 NAAEC2Role ARN을 입력합니다. • NAARoleName — naa-resources.yaml 파일을 배포할 때 다른 이름을 사용하지 않는 한 NAAExecRole 기본값을 유지합니다. 4. 검토 페이지에서 다음 리소스에 필요한 기능: 	<p>AWS DevOps</p>

작업	설명	필요한 기술
	[AWS::IAM::Role] 를 선택한 다음 [Create Stack] 을 선택합니다.	

분석 수행

작업	설명	필요한 기술
셸 스크립트를 사용자 지정합니다.	<ol style="list-style-type: none"> 조직의 보안 계정에 로그인합니다. 세션 관리자를 사용하여 이전에 프로비저닝한 Network Access Analyzer용 EC2 인스턴스에 연결합니다. 자세한 지침은 Session Manager를 사용하여 Linux 인스턴스에 연결을 참조하십시오. 연결할 수 없는 경우 이 패턴의 문제 해결 섹션을 참조하십시오. 다음 명령을 입력하여 편집할 naa-script.sh 파일을 엽니다. <div style="border: 1px solid #ccc; border-radius: 10px; padding: 10px; margin: 10px 0;"> <pre>sudo -i cd /usr/local/naa vi naa-script.sh</pre> </div> 환경에 맞게 필요한 만큼 이 스크립트의 조정 가능한 파라미터와 변수를 검토하고 수정합니다. 사용자 지정 옵션에 대한 자세한 내용은 스크립트 시작 부분에 있는 설명을 참조하십시오. 	AWS DevOps

작업	설명	필요한 기술
	<p>예를 들어 관리 계정에서 조직의 모든 멤버 계정 목록을 가져오는 대신 스크립트를 수정하여 스캔하려는 AWS 계정 ID 또는 AWS 리전을 지정하거나 이러한 파라미터가 들어 있는 외부 파일을 참조할 수 있습니다.</p> <p>5. <code>naa-script.sh</code> 파일을 저장하고 닫습니다.</p>	

작업	설명	필요한 기술
대상 계정을 분석합니다.	<p>1. 다음 명령을 입력합니다. 그러면 naa-script.sh 스크립트가 실행됩니다.</p> <pre data-bbox="630 394 1029 594">sudo -i cd /usr/local/naa screen ./naa-script.sh</pre> <p>유념할 사항:</p> <ul style="list-style-type: none"> • screen 명령을 사용하면 연결 시간이 초과되거나 콘솔에 액세스할 수 없는 경우에도 스크립트가 계속 실행되도록 할 수 있습니다. • 스캔이 시작된 후 Ctrl+A D를 눌러 화면을 강제로 분리할 수 있습니다. 화면이 분리되면 분석이 진행되는 동안 인스턴스 연결을 닫을 수 있습니다. • 분리된 세션을 재개하려면 인스턴스에 연결하고 sudo -i(을)를 입력한 다음 screen -r(을)를 입력합니다. <p>2. 출력에 오류가 있는지 모니터링하여 스크립트가 제대로 작동하는지 확인합니다. 샘플 출력이 경우, 이 패턴의 추가 정보 섹션을 참조합니다.</p>	AWS DevOps

작업	설명	필요한 기술
	<p>3. 분석이 완료될 때까지 기다립니다. 이메일 알림을 구성한 경우 결과가 S3 버킷에 업로드되거나 Security Hub로 가져오기될 때 이메일을 받게 됩니다.</p>	
<p>옵션 1 — S3 버킷에서 결과를 검색합니다.</p>	<ol style="list-style-type: none"> 1. CSV 파일을 <code>naa-<accountID>-<region></code> 버킷에서 다운로드합니다. 자세한 지침은 Amazon S3 설명서의 객체 다운로드를 참조하십시오. 2. S3 버킷에서 CSV 파일을 삭제합니다. 이는 비용 최적화를 위한 모범 사례입니다. 자세한 지침은 Amazon S3 설명서의 객체 삭제를 참조하십시오. 	<p>AWS DevOps</p>

작업	설명	필요한 기술
옵션 2 - Security Hub에서 결과를 검토합니다.	<ol style="list-style-type: none"> https://console.aws.amazon.com/securityhub/에서 Security Hub 콘솔을 엽니다. 탐색 창에서 Findings(조사 결과)를 선택합니다. Network Access Analyzer 조사 결과를 검토합니다. 지침은 Security Hub 설명서에서 조사 결과 목록 및 세부 정보 보기를 참조하십시오. <p>참고: 필터로 제목 시작을 추가하고 Network Access Analyzer(을)를 입력하여 조사 결과를 검색할 수 있습니다.</p>	AWS DevOps

조사 결과의 문제 해결 및 조사 결과 제외

작업	설명	필요한 기술
조사 결과의 문제를 해결합니다.	해결하려는 모든 조사 결과의 문제를 해결합니다. AWS ID, 리소스, 네트워크를 중심으로 경계를 생성하는 방법에 대한 자세한 내용 및 모범 사례는 AWS에서 데이터 경계 빌드 (AWS 백서)를 참조하십시오.	AWS DevOps
정상 작동이 확인된 네트워크 경로를 가진 리소스는 제외합니다.	Network Access Analyzer가 인터넷에서 액세스할 수 있어야 하는 리소스에 대한 조사 결과	AWS DevOps

작업	설명	필요한 기술
	<p>를 생성하는 경우 해당 리소스를 제외 목록에 추가할 수 있습니다. 다음에 Network Access Analyzer를 실행할 때는 해당 리소스에 대한 조사 결과가 생성되지 않습니다.</p> <ol style="list-style-type: none"> 1. <code>/usr/local/naa</code> (으)로 이동한 다음 <code>naa-script.sh</code> 스크립트를 엽니다. <code>S3_EXCLUSION_FILE</code> 변수의 값을 기록해 둡니다. 2. <code>S3_EXCLUSION_FILE</code> 변수 값이 <code>true</code>인 경우 <code>naa-<accountID>-<region></code> 버킷에서 <code>naa-exclusions.csv</code> 파일을 다운로드합니다. 자세한 지침은 Amazon S3 설명서의 객체 다운로드를 참조하십시오. <p><code>S3_EXCLUSION_FILE</code> 변수 값이 <code>false</code>인 경우 <code>/usr/local/naa</code> (으)로 이동한 다음 <code>naa-exclusions.csv</code> 파일을 엽니다.</p> <p>참고: <code>S3_EXCLUSION_FILE</code> 변수 값이 <code>false</code>인 경우 스크립트는 로컬 버전의 제외 파일을 사용합니다. 나중에 값을 <code>true</code>(으)로 변경하면 스크립트가 로컬 버전을 S3 버킷의 파일로 덮어씁니다.</p>	

작업	설명	필요한 기술
	<p>3. naa-exclusions.csv 파일에서 제외하려는 리소스를 입력합니다. 각 줄에 리소스를 하나씩 입력하고 다음 형식을 사용합니다.</p> <pre><resource_id>,<sec_group_id>,<sgrule_cidr>,<sgrule_port_range>,<sgrule_protocol></pre> <p>다음은 예제 리소스입니다.</p> <pre>eni-1111aaaaa2222bbb,sg-3333cccc4444ddd,0.0.0.0/0,80 to 80,tcp</pre> <p>4. naa-exclusions.csv 파일을 저장하고 닫습니다.</p> <p>5. S3 버킷에서 naa-exclusions.csv 파일을 다운로드한 경우 새 버전을 업로드합니다. 이에 관한 지침은 Amazon S3 설명서의 객체 업로드를 참조하세요.</p>	

(선택 사항) naa-script.sh 스크립트 업데이트

작업	설명	필요한 기술
<p>(선택 사항) naa-script.sh 스크립트 업데이트</p>	<p>naa-script.sh 스크립트를 리포지토리의 최신 버전으로 업데이트하려면 다음을 수행합니다.</p>	<p>AWS DevOps</p>

작업	설명	필요한 기술
	<p>1. 세션 관리자를 사용하여 EC2 인스턴스에 연결합니다. 자세한 지침은 Session Manager를 사용하여 Linux 인스턴스에 연결을 참조하십시오.</p> <p>2. 다음 명령을 입력합니다.</p> <pre data-bbox="630 577 1029 659">sudo -i</pre> <p>3. naa-script.sh 스크립트 디렉터리로 이동합니다.</p> <pre data-bbox="630 793 1029 875">cd /usr/local/naa</pre> <p>4. 사용자 지정 변경 사항을 최신 버전에 병합할 수 있도록 다음 명령을 입력하여 로컬 스크립트를 stash합니다.</p> <pre data-bbox="630 1102 1029 1184">git stash</pre> <p>5. 다음 명령을 입력하여 최신 버전의 스크립트를 다운로드합니다.</p> <pre data-bbox="630 1367 1029 1449">git pull</pre> <p>6. 다음 명령을 입력하여 사용자 지정 스크립트를 최신 버전의 스크립트와 통합합니다.</p> <pre data-bbox="630 1675 1029 1757">git stash pop</pre>	

(선택 사항)정리

작업	설명	필요한 기술
<p>배포된 모든 리소스를 삭제합니다.</p>	<p>리소스는 계정에 배포된 상태로 둘 수 있습니다.</p> <p>모든 리소스의 프로비저닝을 해제하려면 다음을 수행합니다.</p> <ol style="list-style-type: none"> 1. 관리 계정에 프로비저닝된 NAA-ExecRole 스택을 삭제합니다. 지침은 설명서의 스택 삭제를 CloudFormation 참조하십시오. 2. 조직의 관리 계정 또는 위임된 관리자 계정에 프로비저닝된 NAA-ExecRole 스택 세트를 삭제합니다. 지침은 설명서의 스택 세트 삭제를 CloudFormation 참조하십시오. 3. naa-<code><accountID></code>-<code><region></code> S3 버킷의 모든 객체를 삭제합니다. 자세한 지침은 Amazon S3 설명서의 객체 삭제를 참조하십시오. 4. 관리 계정에 프로비저닝된 NAA-Resources 스택을 삭제합니다. 지침은 설명서의 스택 삭제를 CloudFormation 참조하십시오. 	<p>AWS DevOps</p>

문제 해결

문제	Solution
세션 관리자를 사용하여 EC2 인스턴스에 연결합니다.	SSM 에이전트는 Systems Manager 엔드포인트와 통신할 수 있어야 합니다. 다음을 따릅니다. <ol style="list-style-type: none"> 1. EC2 인스턴스가 배포된 서브넷이 인터넷에 액세스할 수 있는지 확인합니다. 2. EC2 인스턴스를 재부팅합니다.
스택 세트를 배포할 때 CloudFormation 콘솔에 배포하라는 메시지가 표시됩니다. Enable trusted access with AWS Organizations to use service-managed permissions	이는 AWS Organizations와 간에 신뢰할 수 있는 액세스가 활성화되지 않았음을 나타냅니다. CloudFormation. 서비스 관리형 스택 세트를 배포하려면 신뢰할 수 있는 액세스가 필요합니다. 신뢰할 수 있는 액세스를 활성화하는 버튼을 선택합니다. 자세한 내용은 CloudFormation 설명서의 신뢰할 수 있는 액세스 활성화를 참조하십시오 .

관련 리소스

- [신규 - Amazon VPC Network Access Analyzer](#)(AWS 블로그 게시물)
- [AWS re:Inforce 2022 - AWS에서 유효한 네트워크 액세스 제어를 확인\(NIS202\)](#)(동영상)
- [데모 - Network Access Analyzer를 사용한 조직 전반의 인터넷 수신 데이터 경로 분석](#)(동영상)

추가 정보

콘솔 출력 예제

다음 샘플은 대상 계정 목록을 생성하고 대상 계정을 분석한 결과물을 보여줍니다.

```
[root@ip-10-10-43-82 naa]# ./naa-script.sh
download: s3://naa-<account ID>-us-east-1/naa-exclusions.csv to ./naa-exclusions.csv

AWS Management Account: <Management account ID>
```

```
AWS Accounts being processed...
<Account ID 1> <Account ID 2> <Account ID 3>

Assessing AWS Account: <Account ID 1>, using Role: NAAExecRole
Assessing AWS Account: <Account ID 2>, using Role: NAAExecRole
Assessing AWS Account: <Account ID 3>, using Role: NAAExecRole
Processing account: <Account ID 1> / Region: us-east-1
Account: <Account ID 1> / Region: us-east-1 - Detecting Network Analyzer scope...
Processing account: <Account ID 2> / Region: us-east-1
Account: <Account ID 2> / Region: us-east-1 - Detecting Network Analyzer scope...
Processing account: <Account ID 3> / Region: us-east-1
Account: <Account ID 3> / Region: us-east-1 - Detecting Network Analyzer scope...
Account: <Account ID 1> / Region: us-east-1 - Network Access Analyzer scope detected.
Account: <Account ID 1> / Region: us-east-1 - Continuing analyses with Scope ID.
  Accounts with many resources may take up to one hour
Account: <Account ID 2> / Region: us-east-1 - Network Access Analyzer scope detected.
Account: <Account ID 2> / Region: us-east-1 - Continuing analyses with Scope ID.
  Accounts with many resources may take up to one hour
Account: <Account ID 3> / Region: us-east-1 - Network Access Analyzer scope detected.
Account: <Account ID 3> / Region: us-east-1 - Continuing analyses with Scope ID.
  Accounts with many resources may take up to one hour
```

CSV 보고서 예제

다음 이미지는 CSV 출력의 예제입니다.

AWS Organizations를 사용하여 Transit Gateway Attachment에 자동으로 태그 지정

작성자: Richard Milner-Watts(AWS), Haris Bin Ayub(AWS) 및 John Capps(AWS)

코드 리포지토리: [Transit Gateway 첨부 태거](#)

환경: 프로덕션

기술: 네트워킹, 인프라, 관리 및 거버넌스, 운영

AWS 서비스: AWS Step Functions, AWS Transit Gateway, Amazon VPC, AWS Lambda

요약

Amazon Web Services(AWS)에서는 [AWS 리소스 액세스 관리자](#)를 사용하여 AWS 계정 경계를 넘어 [AWS Transit Gateway](#)를 공유할 수 있습니다. 그러나 계정 경계를 넘어 Transit Gateway attachment를 생성할 경우 첨부 파일은 이름 태그 없이 생성됩니다. 따라서 첨부 파일을 식별하는 데 시간이 많이 걸릴 수 있습니다.

이 솔루션은 [AWS Organizations](#)에서 관리하는 조직 내 계정의 각 Transit Gateway Attachment에 대한 정보를 수집하는 자동화된 메커니즘을 제공합니다. 이 프로세스에는 Transit Gateway 라우팅 테이블에서 [Classless Inter-Domain Routing\(CIDR\)](#) 범위를 조회하는 작업이 포함됩니다. 그러면 솔루션은 트랜짓 게이트웨이를 보유한 계정 내의 첨부 파일에 이름 태그를 <CIDR-range>-<AccountName>의 형태로 적용합니다.

이 솔루션은 AWS Solutions Library의 [서버리스 트랜짓 네트워크 오케스트레이터](#)와 같은 솔루션과 함께 사용할 수 있습니다. 서버리스 트랜짓 네트워크 오케스트레이터를 사용하면 대규모로 Transit Gateway Attachment를 자동으로 생성할 수 있습니다.

사전 조건 및 제한 사항

사전 조건

- 활성 상태의 AWS 계정
- 모든 관련 계정을 포함하는 AWS Organizations 조직

- 조직의 루트 아래에 있는 조직 관리 계정에 액세스하여 필요한 AWS Identity and Access Management(IAM) 역할 생성
- 조직과 공유되고 첨부 파일이 있는 하나 이상의 트랜짓 게이트웨이를 포함하는 공유 네트워킹 멤버 계정

아키텍처

AWS Management Console의 다음 스크린샷은 연결된 이름 태그가 없는 Transit Gateway Attachment 및 이 솔루션에서 생성된 이름 태그가 있는 Transit Gateway Attachment 두 개의 예를 보여줍니다. 생성된 이름 태그의 구조는 <CIDR-range>-<AccountName>입니다.

이 솔루션은 [AWS를 CloudFormation](#) 사용하여 구성된 모든 지역의 Transit Gateway 이름 태그 생성을 관리하는 [AWS Step Functions](#) 워크플로를 배포합니다. 워크플로는 기본 작업을 수행하는 [AWS Lambda](#) 함수를 간접적으로 호출합니다.

솔루션이 AWS Organizations로부터 계정 이름을 가져오면 Step Functions 상태 머신은 모든 Transit Gateway Attachment ID를 가져옵니다. 이는 AWS 리전에서 병렬로 처리됩니다. 이 처리에는 각 첨부 파일의 CIDR 범위 검색이 포함됩니다. CIDR 범위는 리전 내의 Transit Gateway 라우팅 테이블에서 일치하는 Transit Gateway Attachment ID를 검색하여 얻을 수 있습니다. 필요한 정보를 모두 사용할 수 있는 경우 솔루션은 이름 태그를 첨부 파일에 적용합니다. 솔루션은 기존 이름 태그를 덮어쓰지 않습니다.

솔루션은 [Amazon EventBridge](#) 이벤트에서 제어하는 일정에 따라 실행됩니다. 이벤트는 매일 오전 6시 (UTC)에 솔루션을 시작합니다.

대상 기술 스택

- 아마존 EventBridge
- AWS Lambda
- AWS Organizations
- AWS Transit Gateway
- Amazon Virtual Private Cloud(VPC)
- AWS X-Ray

대상 아키텍처

솔루션 아키텍처 및 워크플로는 다음 다이어그램에 나와 있습니다.

1. 예약된 이벤트가 규칙을 시작합니다.
2. EventBridge 규칙은 Step Functions 상태 머신을 시작합니다.
3. 상태 머신은 `tgw-tagger-organizations-account-query` Lambda 함수를 호출합니다.
4. `tgw-tagger-organizations-account-query` Lambda 함수는 조직 관리 계정에서 역할을 맡습니다.
5. `tgw-tagger-organizations-account-query` Lambda 함수는 Organizations API를 호출하여 AWS 계정 메타데이터를 반환합니다.
6. 상태 머신은 `tgw-tagger-attachment-query` Lambda 함수를 호출합니다.
7. 병렬로 각 리전에 대해 상태 머신은 `tgw-tagger-rtb-query` Lambda 함수를 간접적으로 호출하여 각 첨부 파일의 CIDR 범위를 읽습니다.
8. 병렬로 각 리전에 대해 상태 머신은 Lambda 함수를 `tgw-tagger-attachment-tagger` 간접적으로 호출합니다.
9. 공유 네트워킹 계정의 Transit Gateway Attachment에 대한 이름 태그가 생성됩니다.

자동화 및 규모 조정

솔루션은 각 리전을 병렬로 처리하여 총 실행 시간을 단축합니다.

도구

서비스

- [AWS CloudFormation](#) — CloudFormation AWS는 인프라를 코드로 취급하여 관련 AWS 및 타사 리소스 모음을 모델링하고, 빠르고 일관되게 프로비저닝하고, 수명 주기 전반에 걸쳐 관리할 수 있는 방법을 제공합니다.
- [Amazon EventBridge](#) — EventBridge Amazon은 애플리케이션을 다양한 소스의 데이터와 연결하는데 사용할 수 있는 서버리스 이벤트 버스 서비스입니다. EventBridge 환경 변화를 나타내는 이벤트를 수신하고 규칙을 적용하여 이벤트를 대상으로 라우팅합니다. 규칙은 이벤트 패턴이라고 하는 이벤트 구조 또는 일정에 따라 이벤트를 대상과 일치시킵니다.
- [AWS Lambda](#) - AWS Lambda는 서버를 프로비저닝하거나 관리하지 않고도 코드를 실행할 수 있게 해주는 컴퓨팅 서비스입니다. Lambda는 필요 시에만 코드를 실행하며, 일일 몇 개의 요청에서 초당

수천 개의 요청까지 자동으로 확장이 가능합니다. 사용한 컴퓨팅 시간에 대해서만 비용을 지불하면 됩니다. 코드가 실행되지 않을 때는 비용이 부과되지 않습니다.

- [AWS Organizations](#) – AWS Organizations는 AWS 리소스 규모 조정 및 확장 시 중앙에서 환경을 관리하고 통제하는 데 도움이 됩니다. AWS Organizations를 사용하면 프로그래밍 방식으로 새 AWS 계정을 만들고 리소스를 할당하며, 계정을 그룹화하여 워크플로를 구성하고, 거버넌스를 위해 계정 또는 그룹에 정책을 적용하고, 모든 계정에 대해 단일 결제 방법을 사용하여 청구를 간소화할 수 있습니다.
- [AWS Step Functions](#) – AWS Step Functions는 AWS 서비스를 오케스트레이션하고, 비즈니스 프로세스를 자동화하며, 서버리스 애플리케이션을 구축하는 데 사용되는 코드가 적은 시각적 워크플로 서비스입니다. 워크플로는 장애, 재시도, 병렬화, 서비스 통합 및 관찰성을 관리하므로 개발자는 더 중요한 비즈니스 로직에 집중할 수 있습니다.
- [AWS Transit Gateway](#) – AWS Transit Gateway는 중앙 허브를 통해 VPC와 온프레미스 네트워크를 연결합니다. 이렇게 하면 네트워크가 단순화되고 복잡한 피어링 관계가 없어집니다. 클라우드 라우터 역할을 하므로 새로 연결할 때마다 한 번만 이루어집니다.
- [Amazon VPC](#) – Amazon Virtual Private Cloud(VPC)는 사용자가 정의하는 논리적으로 격리된 가상 네트워크에서 AWS 리소스를 시작할 수 있는 서비스입니다.
- [AWS X-Ray](#) – AWS X-Ray는 애플리케이션이 처리하는 요청에 대한 데이터를 수집하는 웹 서비스로, 해당 데이터를 보고, 필터링하고, 통찰을 얻어 문제와 최적화 기회를 식별할 수 있는 도구를 제공합니다.

code

이 솔루션의 소스 코드는 [Transit Gateway 첨부 태거](#) GitHub 리포지토리에서 사용할 수 있습니다. 리포지토리에는 다음 파일이 포함됩니다.

- `tgw-attachment-tagger-main-stack.yaml`은 공유 네트워킹 계정 내에 이 솔루션을 지원하는 모든 리소스를 생성합니다.
- `tgw-attachment-tagger-organizations-stack.yaml`은 조직의 관리 계정에 역할을 생성합니다.

에픽

기본 솔루션 스택 배포

작업	설명	필요한 기술
필수 사전 조건 정보를 수집하십시오.	<p>Lambda 함수에서 AWS Organizations API로의 크로스 계정 액세스를 구성하려면 조직의 관리 계정에 대한 계정 ID가 필요합니다.</p> <p>참고: 두 CloudFormation 스택이 생성되는 순서가 중요합니다. 먼저 공유 네트워킹 계정에 리소스를 배포해야 합니다. 조직의 관리 계정에 리소스를 배포하기 전에 공유 네트워킹 계정의 역할이 이미 존재해야 합니다. 자세한 내용은 AWS 설명서를 참조하십시오.</p>	DevOps 엔지니어
기본 솔루션 스택용 CloudFormation 템플릿을 실행합니다.	<p>기본 솔루션 스택용 템플릿은 IAM 역할, Step Functions 워크플로, Lambda 함수 및 이벤트를 배포합니다. CloudWatch</p> <p>공유 네트워킹 계정용 AWS Management 콘솔을 연 다음 CloudFormation 콘솔을 엽니다. <code>tgw-attachment-tagger-main-stack.yaml</code> 1 템플릿과 다음 값을 사용하여 스택을 생성합니다.</p> <ul style="list-style-type: none"> 스택 이름 — <code>tgw-attachment-tagger-main-stack</code> 	DevOps 엔지니어

작업	설명	필요한 기술
	<ul style="list-style-type: none"> • awsOrganizationsRoleAccountId— 조직 관리 계정의 계정 ID • TGWRegions 파라미터- 솔루션의 AWS 리전, 쉼표로 구분된 문자열로 입력 • TGWlist 파라미터 - 솔루션에서 제외할 트랜짓 게이트웨이 ID, 쉼표로 구분된 문자열로 입력 <p>CloudFormation 스택 시작에 대한 자세한 내용은 AWS 설명서를 참조하십시오.</p>	
<p>솔루션이 성공적으로 시작되었는지 확인하십시오.</p>	<p>CloudFormation 스택이 CREATE_COMPLETE 상태에 도달할 때까지 기다리십시오. 1 분 내에 완료됩니다.</p> <p>Step Functions 콘솔을 열고 새 상태 머신이 tgw-attachment-tagger-state-machine이라는 이름으로 생성되었는지 확인합니다.</p>	<p>DevOps 엔지니어</p>

AWS Organizations 스택 배포

작업	설명	필요한 기술
<p>필수 사전 조건 정보를 수집하십시오.</p>	<p>Lambda 함수에서 AWS Organizations API로의 크로스 계정 액세스를 구성하려면 공</p>	<p>DevOps 엔지니어</p>

작업	설명	필요한 기술
	<p>유 네트워킹 계정의 계정 ID가 필요합니다.</p>	
<p>Organizations 스택용 CloudFormation 템플릿을 실행합니다.</p>	<p>AWS Organizations 스택용 템플릿이 조직의 관리 계정에 IAM 역할을 배포 합니다.</p> <p>조직의 관리 계정에 대한 AWS 콘솔에 액세스한 다음 콘솔을 CloudFormation 엽니다.</p> <p>tgw-attachment-tagger-organizations-stack.yaml 템플릿과 다음 값을 사용하여 스택을 생성합니다.</p> <ul style="list-style-type: none"> • 스택 이름 — -stack tgw-attachment-tagger-organizations • NetworkingAccountId파라미터 — 공유 네트워킹 계정의 계정 ID <p>다른 스택 생성 옵션의 경우 기본값을 사용합니다.</p>	<p>DevOps 엔지니어</p>

작업	설명	필요한 기술
솔루션이 성공적으로 시작되었는지 확인하십시오.	<p>CloudFormation 스택이 CREATE_COMPLETE 상태에 도달할 때까지 기다리십시오. 1분 내에 완료됩니다.</p> <p>ID 및 액세스 관리 (IAM) 콘솔을 열고 -query-role이라는 이름으로 새 역할이 생성되었는지 확인합니다. tgw-attachment-tagger-organization</p>	DevOps 엔지니어

솔루션 확인

작업	설명	필요한 기술
상태 머신을 실행합니다.	<p>공유 네트워킹 계정의 Step Functions 콘솔을 열고 탐색 창에서 상태 머신을 선택합니다.</p> <p>스테이트 머신 tgw-attachment-tagger-state-machine을 선택하고 실행 시작을 선택합니다.</p> <p>이 상태 머신에 대한 입력은 솔루션에서 사용되지 않으므로 기본값을 사용할 수 있습니다.</p> <pre> { "Comment": "Insert your JSON here" } </pre> <p>실행 시작을 선택합니다.</p>	DevOps 엔지니어

작업	설명	필요한 기술
<p>완료될 때까지 상태 머신을 주시하십시오.</p>	<p>새 페이지가 열리면 상태 머신이 실행되는 것을 볼 수 있습니다. 기간은 처리할 Transit Gateway Attachment 수에 따라 달라집니다.</p> <p>이 페이지에서 상태 머신의 각 단계를 검토할 수 있습니다. 상태 머신 내의 다양한 작업을 보고 Lambda 함수의 CloudWatch 로그 링크를 따라갈 수 있습니다. 맵 내에서 병렬로 실행되는 작업의 경우 인덱스 드롭다운 목록을 사용하여 각 리전의 구체적인 구현을 확인할 수 있습니다.</p>	<p>DevOps 엔지니어</p>
<p>Transit Gateway Attachment 태그를 확인합니다.</p>	<p>공유 네트워킹 계정의 VPC 콘솔을 열고 Transit Gateway Attachment 첨부 파일을 선택합니다. 콘솔에서는 조건을 충족하는 첨부 파일에 대해 이름 태그가 제공됩니다(첨부 파일은 Transit Gateway 라우팅 테이블에 전파되고 리소스 소유자는 조직의 구성원임).</p>	<p>DevOps 엔지니어</p>

작업	설명	필요한 기술
<p>CloudWatch 이벤트 시작을 확인하세요.</p>	<p>CloudWatch 이벤트가 시작될 때까지 기다리세요. 이 일정은 06:00(UTC)으로 예정되어 있습니다.</p> <p>그런 다음 공유 네트워킹 계정의 Step Functions 콘솔을 열고 탐색 창에서 상태 머신을 선택합니다.</p> <p>스테이트 머신 <code>tgw-attachment-tagger-state-machine</code> 을 선택합니다. 솔루션이 06:00(UTC)에 실행되었는지 확인합니다.</p>	<p>DevOps 엔지니어</p>

관련 리소스

- [AWS Organizations](#)
- [AWS Resource Access Manager](#)
- [서버리스 트랜짓 네트워크 오케스트레이터](#)
- [IAM 역할 생성](#)
- [AWS CloudFormation 콘솔에 스택 생성](#)

ELB 로드 밸런서에 TLS 터미널이 필요한지 검증합니다

작성자: Priyanka Chaudhary(AWS)

환경: 프로덕션

기술: 네트워킹; 보안, 자격 증명, 규정 준수

AWS 서비스: 아마존 CloudWatch 이벤트, 엘라스틱 로드 밸런싱 (ELB), AWS Lambda

요약

Amazon Web Services(AWS) 클라우드에서Elastic Load Balancing(ELB)은 들어오는 애플리케이션 트래픽을 Amazon Elastic Compute Cloud(Amazon EC2) 인스턴스, 컨테이너, IP 주소, AWS Lambda 함수 등 여러 대상에 자동으로 분산합니다. 로드 밸런서는 리스너를 사용하여 로드 밸런서가 사용자의 트래픽을 수락하는 데 사용하는 포트와 프로토콜을 정의합니다. 애플리케이션 로드 밸런서는 애플리케이션 계층에서 라우팅을 결정하고 HTTP/HTTPS 프로토콜을 사용합니다. 클래식 로드 밸런서는 TCP 또는 Secure Sockets Layer(SSL) 프로토콜을 사용하여 전송 계층에서 또는 HTTP/HTTPS를 사용함으로써 애플리케이션 계층에서 라우팅 결정을 내립니다.

이 패턴은 Application Load Balancer 및 Classic Load Balancer에 대한 여러 이벤트 유형을 검사하는 보안 제어를 제공합니다. 함수가 호출되면 AWS Lambda는 이벤트를 검사하고 로드 밸런서가 규정을 준수하는지 확인합니다.

함수는 [CreateLoadBalancer](#),,,,,,

[CreateLoadBalancerListenersDeleteLoadBalancerListenersCreateLoadBalancerPolicySetLoadBalancerPolicy](#)

및 [ModifyListener](#)API 호출에서 Amazon CloudWatch Events 이벤트를 시작합니다. 이벤트가 이러한 API 중 하나를 감지하면 Python 스크립트를 실행하는 AWS Lambda를 호출합니다. Python 스크립트는 리스너에 SSL 인증서가 포함되어 있는지, 적용된 정책이 전송 계층 보안(TLS)을 사용하는지 여부를 평가합니다. SSL 정책이 TLS가 아닌 것으로 확인되면 함수는 관련 정보와 함께 Amazon Simple Notification Service(SNS) 알림을 사용자에게 보냅니다.

사전 조건 및 제한 사항

사전 조건

- 활성 상태의 AWS 계정.

제한 사항

- 이 보안 제어는 로드 밸런서 리스너를 업데이트하지 않는 한 기존 로드 밸런서를 확인하지 않습니다.
- 이 보안 제어는 리전별로 적용됩니다. 모니터링하려는 각 AWS 리전에 이를 배포해야 합니다.

아키텍처

대상 아키텍처

자동화 및 규모 조정

- [AWS Organizations](#)를 사용하는 경우, [AWS StackSets Cloudformation](#)을 사용하여 모니터링하려는 여러 계정에 이 템플릿을 배포할 수 있습니다.

도구

서비스

- [AWS CloudFormation](#) — AWS는 AWS CloudFormation 리소스를 모델링 및 설정하고, 빠르고 일관되게 프로비저닝하고, 수명 주기 전반에 걸쳐 관리할 수 있도록 지원합니다. 템플릿을 사용하여 리소스와 해당 종속성을 설명하고, 리소스를 개별적으로 관리하는 대신 스택으로 함께 시작하고 구성할 수 있습니다.
- [Amazon CloudWatch Events](#) — Amazon CloudWatch Events는 AWS 리소스의 변경 사항을 설명하는 시스템 이벤트의 스트림을 거의 실시간으로 제공합니다.
- [AWS Lambda](#) - AWS Lambda는 서버를 프로비저닝하거나 관리하지 않고도 코드를 실행할 수 있도록 지원하는 컴퓨팅 서비스입니다.
- [Amazon S3](#)-Amazon Simple Storage Service(S3)는 웹 사이트, 모바일 애플리케이션, 백업, 데이터 레이크 등 다양한 스토리지 솔루션에 사용할 수 있는 확장성이 뛰어난 객체 스토리지 서비스입니다.
- [Amazon SNS](#) – Amazon Simple Notification Service(SNS)는 웹 서버와 이메일 주소를 포함하여 게시자와 클라이언트 간에 메시지를 전달 또는 전송하는 것을 조정하고 관리합니다. 구독자는 구독하는 주제에 게시된 모든 메시지를 수신하며 주제에 대한 모든 구독자는 동일한 메시지를 수신합니다.

코드

이 패턴에는 다음과 같은 첨부 파일이 포함됩니다.

- ELBRequirestlstermination.zip - 보안 제어를 위한 Lambda 코드입니다.
- ELBRequirestlstermination.yml— 이벤트 및 Lambda 함수를 설정하는 CloudFormation 템플릿.

에픽

S3 버킷 선택

작업	설명	필요한 기술
S3 버킷을 정의합니다.	Amazon S3 콘솔 에서 Lambda 코드 .zip 파일을 호스팅할 S3 버킷을 선택하거나 생성합니다. 이 S3 버킷은 평가하려는 로드 밸런서와 동일한 AWS 리전에 있어야 합니다. S3 버킷 이름은 전역 수준에서 고유하며, 네임스페이스는 모든 AWS 계정이 공유합니다. S3 버킷 이름에는 선행 슬래시를 포함할 수 없습니다.	클라우드 아키텍트
Lambda 코드를 업로드합니다.	첨부 파일 섹션에 제공된 Lambda 코드(ELBRequirestlstermination.zip 파일)를 S3 버킷에 업로드합니다.	클라우드 아키텍트

템플릿을 배포하십시오. CloudFormation

작업	설명	필요한 기술
AWS CloudFormation 템플릿을 실행합니다.	S3 버킷과 동일한 AWS 리전에서 AWS CloudFormation 콘솔 을 열고 첨부된 템플릿을 ELBRequirestlsterm	클라우드 아키텍트

작업	설명	필요한 기술
	<p>ination.yml 배포합니다. AWS CloudFormation 템플릿 배포에 대한 자세한 내용은 CloudFormation 설명서의 AWS CloudFormation 콘솔에 스택 생성을 참조하십시오.</p>	

작업	설명	필요한 기술
<p>템플릿에서 파라미터를 작성합니다.</p>	<p>템플릿을 시작하면 다음 정보를 입력하라는 메시지가 표시됩니다.</p> <ul style="list-style-type: none"> • S3 버킷: 첫 번째 에픽에서 생성하거나 선택한 버킷을 지정합니다. 여기에 첨부된 Lambda 코드(ELBRequirestlstermination.zip 파일)를 업로드했습니다. • S3 키: S3 버킷에 있는 Lambda .zip 파일의 위치를 지정합니다(예: ELBRequirestlstermination.zip 또는 controls/ELBRequirestlstermination.zip). 선행 슬래시를 포함하지 마세요. • 알림 이메일: Amazon SNS 알림을 받으려는 활성 이메일 주소를 입력합니다. • Lambda 로깅 수준: Lambda 함수의 로깅 수준 및 빈도를 지정합니다. 정보를 사용하여 진행 상황에 대한 자세한 정보 메시지를 기록하고, 배포를 계속할 수 있는 오류 이벤트의 경우 오류를 기록하고, 잠재적으로 유해한 상황에 대한 경고를 기록할 수 있습니다. 	<p>클라우드 아키텍트</p>

구독 확인

작업	설명	필요한 기술
구독을 확인합니다.	CloudFormation 템플릿이 성공적으로 배포되면 제공한 이메일 주소로 구독 이메일이 전송됩니다. 위반 알림을 받기 시작하려면 이 이메일 구독을 확인해야 합니다.	클라우드 아키텍트

관련 리소스

- [AWS CloudFormation 콘솔에서 스택 생성](#) (AWS CloudFormation 설명서)
- [AWS Lambda란 무엇입니까?](#) (AWS Lambda 설명서)
- [Classic Load Balancer란 무엇인가요?](#) (ELB 설명서)
- [Application Load Balancer란 무엇인가요?](#) (ELB 설명서)

첨부

이 문서와 관련된 추가 콘텐츠에 액세스하려면 [attachment.zip](#) 파일의 압축을 풉니다.

Splunk를 사용하여 AWS 네트워크 방화벽 로그 및 지표 보기

작성자: Ivo Pinto

환경: PoC 또는 파일럿

기술: 네트워킹, 클라우드 네이티브, 콘텐츠 전송, 운영, 보안, ID, 규정 준수

워크로드: 기타 모든 워크로드

AWS 서비스: 아마존 CloudWatch, 아마존 CloudWatch 로그, AWS 네트워크 방화벽

요약

많은 조직에서 [Splunk Enterprise](#)를 다양한 소스의 로그 및 지표에 대한 중앙 집중식 집계 및 시각화 도구로 사용합니다. 이 패턴은 AWS용 Splunk 애드온을 사용하여 [Amazon CloudWatch](#) Logs에서 [AWS 네트워크 방화벽](#) 로그와 지표를 가져오도록 Splunk를 구성하는 데 도움이 됩니다.

이를 위해서는 읽기 전용 AWS ID 및 액세스 관리 (IAM) 역할을 생성해야 합니다. AWS용 Splunk 애드온은 이 역할을 사용하여 액세스합니다. CloudWatch AWS용 Splunk 애드온이 지표와 로그를 가져오도록 구성합니다. CloudWatch 마지막으로 검색된 로그 데이터 및 지표로 Splunk에서 시각화를 생성합니다.

사전 조건 및 제한 사항

사전 조건

- [Splunk 계정](#)
- 스플링크 엔터프라이즈 인스턴스, 버전 8.2.2 이상
- 활성 상태의 AWS 계정.
- Network Firewall, 로그를 로그로 전송하도록 [설정](#) 및 [구성](#) CloudWatch

제한 사항

- Splunk Enterprise는 AWS 클라우드에서 Amazon Elastic Compute Cloud (Amazon EC2) 인스턴스의 클러스터로 배포해야 합니다.
- Amazon EC2에 대해 자동으로 검색된 IAM 역할을 사용하여 데이터를 수집하는 것은 AWS 중국 지역에서 지원되지 않습니다.

아키텍처

다이어그램은 다음을 보여 줍니다.

1. Network Firewall은 로그를 로그에 CloudWatch 게시합니다.
2. Splunk Enterprise는 에서 메트릭과 로그를 검색합니다. CloudWatch

이 아키텍처에 예제 메트릭과 로그를 채우기 위해 워크로드는 Network Firewall 엔드포인트를 통과하여 인터넷으로 이동하는 트래픽을 생성합니다. 이는 [라우팅 테이블](#)을 사용하여 달성됩니다. 이 패턴은 단일 Amazon EC2 인스턴스를 워크로드로 사용하지만, 이 패턴은 로그를 Logs로 전송하도록 Network Firewall이 구성되어 있는 한 모든 아키텍처에 적용할 수 있습니다. CloudWatch

또한 이 아키텍처는 다른 가상 사설 클라우드 (VPC) 의 Splunk Enterprise 인스턴스를 사용합니다. 하지만 Splunk 인스턴스는 API에 CloudWatch 도달할 수 있는 한 다른 위치 (예: 워크로드와 동일한 VPC) 에 있을 수 있습니다.

도구

서비스

- [Amazon CloudWatch Logs](#)를 사용하면 모든 시스템, 애플리케이션 및 AWS 서비스의 로그를 중앙 집중화하여 로그를 모니터링하고 안전하게 보관할 수 있습니다.
- [Amazon Elastic Compute Cloud\(Amazon EC2\)](#)는 AWS 클라우드에서 확장 가능한 컴퓨팅 용량을 제공합니다. 필요한 만큼 가상 서버를 시작하고 빠르게 스케일 업하거나 스케일 다운할 수 있습니다.
- [AWS Network Firewall](#)은 AWS Cloud를 위한 상태 저장형, 관리형 네트워크 방화벽이자, 침입 탐지 및 방지 서비스입니다.

기타 도구

- [Splunk](#)는 로그 데이터를 모니터링, 시각화 및 분석하는 데 도움이 됩니다.

에픽

IAM 역할 생성

작업	설명	필요한 기술
IAM 정책을 생성합니다.	<p>JSON 편집기를 사용한 정책 생성의 지침에 따라 CloudWatch 로그 데이터 및 지표에 대한 읽기 전용 액세스 권한을 부여하는 IAM 정책을 생성하십시오. CloudWatch 다음 정책을 JSON 편집기에 붙여넣습니다.</p> <pre data-bbox="591 835 1029 1806"> { "Statement": [{ "Action": ["cloudwatch:List*", "cloudwatch:Get*", "network-firewall:List*", "logs:Describe*", "logs:Get*", "logs:List*", "logs:StartQuery", "logs:StopQuery", "logs:TestMetricFilter", </pre>	AWS 관리자

작업	설명	필요한 기술
	<pre> "logs:FilterLogEvents", "network-firewall:Describe*"], "Effect": "Allow", "Resource": "*" }], "Version": "2012-10-17" } </pre>	
<p>새 IAM 역할을 생성합니다.</p>	<p>AWS 서비스에 권한을 위임할 역할 생성의 지침에 따라 AWS용 Splunk 애드온이 액세스하는 데 사용하는 IAM 역할을 생성합니다. CloudWatch 권한 정책의 경우 이전에 생성한 정책을 선택합니다.</p>	<p>AWS 관리자</p>

작업	설명	필요한 기술
Splunk 클러스터의 EC2 인스턴스에 IAM 역할을 할당합니다.	<ol style="list-style-type: none"> https://console.aws.amazon.com/ec2/에서 Amazon EC2 콘솔을 엽니다. 탐색 창에서 인스턴스를 선택합니다. Splunk 클러스터에서 EC2 인스턴스를 선택합니다. 작업, 보안, IAM 역할 수정을 차례로 선택합니다. 이전에 생성한 IAM 역할을 선택한 다음 [Save] 를 선택합니다. 	AWS 관리자

AWS용 스플링크 애드온 설치

작업	설명	필요한 기술
애드온을 설치합니다.	<ol style="list-style-type: none"> Splunk 대시보드에서 Splunk 앱으로 이동합니다. Amazon Web Services용 스플링크 애드온을 검색하십시오. 설치를 선택합니다. Splunk 자격 증명을 제공하십시오. 	스플링크 관리자
AWS 자격 증명을 구성합니다.	<ol style="list-style-type: none"> Splunk 대시보드에서 AWS용 스플링크 애드온으로 이동합니다. Configuration(구성)을 선택합니다. 	스플링크 관리자

작업	설명	필요한 기술
	<p>3. 자동 검색된 IAM 역할 열에서 이전에 생성한 IAM 역할을 선택합니다.</p> <p>자세한 내용은 Splunk 설명서의 Splunk 플랫폼 인스턴스 내에서 IAM 역할 찾기를 참조하십시오.</p>	

Splunk 액세스를 다음으로 구성하십시오. CloudWatch

작업	설명	필요한 기술
<p>로그에서 CloudWatch Network Firewall 로그를 검색하도록 구성합니다.</p>	<ol style="list-style-type: none"> 1. Splunk 대시보드에서 AWS 용 스플링크 애드온으로 이동합니다. 2. 입력을 선택합니다. 3. 새 입력 생성을 선택합니다. 4. 목록에서 사용자 지정 데이터 유형을 선택한 다음 CloudWatch 로그를 선택합니다. 5. Network Firewall 로그의 이름, AWS 계정, AWS 지역, 로그 그룹을 입력합니다. 6. 저장을 선택합니다. <p>기본적으로 Splunk는 10분마다 로그 데이터를 가져옵니다. 이는 고급 설정에서 구성할 수 있는 매개변수입니다. 자세한 내용은 Splunk 설명서의 Splunk Web을 사용한</p>	<p>Splunk 관리자</p>

작업	설명	필요한 기술
	CloudWatch 로그 입력 구성을 참조하십시오.	

작업	설명	필요한 기술
<p>에서 CloudWatch Network Firewall 메트릭의 검색을 구성합니다.</p>	<ol style="list-style-type: none"> 1. Splunk 대시보드에서 AWS 용 스플링크 애드온으로 이동합니다. 2. 입력을 선택합니다. 3. 새 입력 생성을 선택합니다. 4. 목록에서 선택합니다 CloudWatch. 5. Network Firewall 지표에 대한 이름, AWS 계정, AWS 지역을 제공하십시오. 6. 지표 구성 옆의 고급 모드에서 편집을 선택합니다. 7. (선택 사항) 사전 구성된 네임스페이스를 모두 삭제합니다. 8. [네임스페이스 추가] 를 선택한 다음 이름을 AWS/로 지정합니다. NetworkFirewall 9. 차원 값에 다음을 추가합니다. <div data-bbox="630 1283 1027 1480" style="border: 1px solid #ccc; border-radius: 10px; padding: 10px; margin: 10px 0;"> <pre>[{"AvailabilityZone":[".*"],"Engine":[".*"],"FirewallName":[".*"]}]]</pre> </div> 10.지표의 경우 모두를 선택합니다. 11.지표 통계의 경우 합계를 선택합니다. 12.확인을 선택합니다. 13.저장을 선택합니다. 	<p>Splunk 관리자</p>

작업	설명	필요한 기술
	기본적으로 Splunk는 5분마다 지포 데이터를 가져옵니다. 이는 고급 설정에서 구성할 수 있는 매개변수입니다. 자세한 내용은 Splunk 설명서의 Splunk Web을 사용한 CloudWatch 입력 구성을 참조하십시오.	

쿼리를 사용하여 Splunk 시각화를 생성합니다.

작업	설명	필요한 기술
상위 소스 IP 주소 보기	<ol style="list-style-type: none"> Splunk 대시보드에서 검색 및 보고로 이동합니다. 여기에 검색 입력 상자에 다음을 입력합니다. <div style="border: 1px solid #ccc; border-radius: 10px; padding: 10px; margin: 10px 0;"> <pre>sourcetype="aws:cloudwatchlogs" top event.src_ip</pre> </div> <p>이 쿼리는 트래픽이 가장 많은 소스 IP 주소 표를 내림차순으로 표시합니다.</p> <ol style="list-style-type: none"> 그래픽으로 표시하려면 시각화를 선택하십시오. 	Splunk 관리자
패킷 통계 보기.	<ol style="list-style-type: none"> Splunk 대시보드에서 검색 및 보고로 이동합니다. 여기에 검색 입력 상자에 다음을 입력합니다. <div style="border: 1px solid #ccc; border-radius: 10px; padding: 10px; margin: 10px 0;"> <pre>sourcetype="aws:cloudwatch" timechart</pre> </div>	Splunk 관리자

작업	설명	필요한 기술
	<pre>sum(Sum) by metric_name</pre> <p>이 쿼리에는 측정치 DroppedPackets 및 ReceivedPackets 분당 테이블이 표시됩니다. PassedPackets</p> <p>3. 그래픽으로 표시하려면 시각화를 선택하세요.</p>	
가장 많이 사용되는 소스 포트 보기	<ol style="list-style-type: none"> Splunk 대시보드에서 검색 및 보고로 이동합니다. 여기에 검색 입력 상자에 다음을 입력합니다. <pre>sourcetype="aws:cloudwatchlogs" top event.dest_port</pre> <p>이 쿼리는 트래픽이 가장 많은 소스 포트 표를 내림차순으로 표시합니다.</p> <p>3. 그래픽으로 표시하려면 시각화를 선택하십시오.</p>	Splunk 관리자

관련 리소스

AWS 설명서

- [AWS 서비스에 권한을 위임하기 위한 역할 생성 \(IAM 설명서\)](#)
- [IAM 정책 생성\(IAM 설명서\)](#)
- [AWS Network Firewall에서의 로깅 및 모니터링 \(네트워크 방화벽 설명서\)](#)
- [AWS 네트워크 방화벽의 라우팅 테이블 구성 \(네트워크 방화벽 설명서\)](#)

AWS 블로그 게시물

- [AWS 네트워크 방화벽 배포 모델](#)

AWS Marketplace

- [스플링크 엔터프라이즈 아마존 머신 이미지 \(AMI\)](#)

더 많은 패턴

- [Session Manager 및 Amazon EC2 인스턴스 연결을 사용한 Bastion Host 액세스](#)
- [AWS Fargate, PrivateLink AWS 및 Network Load Balancer를 사용하여 Amazon ECS에서 비공개로 컨테이너 애플리케이션에 액세스할 수 있습니다.](#)
- [PrivateLink AWS와 Network Load Balancer를 사용하여 Amazon ECS에서 비공개로 컨테이너 애플리케이션에 액세스할 수 있습니다.](#)
- [AWS Managed Microsoft AD와 온프레미스 Microsoft Active Directory를 사용하여 DNS 확인 중앙 집중화](#)
- [IPv4 및 IPv6용 보안 그룹 수신 규칙에서 단일 호스트 네트워크 항목 확인](#)
- [AWS Network Firewall과 AWS Transit Gateway를 사용하여 방화벽 배포](#)
- [프라이빗 엔드포인트와 Application Load Balancer 사용하여 내부 웹 사이트에 Amazon API Gateway API 배포](#)
- [Amazon RDS에서 PostgreSQL DB 인스턴스에 대한 암호화된 연결 활성화하기](#)
- [AWS Transit Gateway Connect를 사용하여 VRF를 AWS로 확장](#)
- [ABAC\(속성 기반 액세스 제어\)를 사용하여 퍼블릭 서브넷 보호](#)
- [F5 BIG-IP 워크로드를 AWS 클라우드의 F5 BIG-IP VE로 마이그레이션](#)
- [비 워크로드 서브넷을 위한 다중 계정 VPC 설계에서 라우팅 가능한 IP 공간 보존](#)
- [AWS Network Firewall에서 Slack 채널로 알림 전송](#)
- [Amazon을 사용하여 VPC를 통해 Amazon S3 버킷의 정적 콘텐츠를 제공합니다. CloudFront](#)
- [EnterpriseOne AWS Elastic 재해 복구를 사용하여 Oracle JD Edwards의 재해 복구 설정](#)
- [다중 계정 AWS 환경에서 하이브리드 네트워크를 위한 DNS 확인 설정](#)
- [BMC Discovery 쿼리를 사용하여 마이그레이션 계획을 위한 마이그레이션 데이터 추출](#)
- [Network Firewall을 사용하여 아웃바운드 트래픽의 Server Name Indication\(SNI\)에서 DNS 도메인 이름 캡처](#)

운영 체제

주제

- [AWS MGN을 사용하여 RHEL BYOL 시스템을 AWS 라이선스가 포함된 인스턴스로 마이그레이션하기](#)
- [Microsoft SQL Server를 AWS 클라우드로 마이그레이션한 후 연결 오류 해결](#)
- [패턴 더 보기](#)

AWS MGN을 사용하여 RHEL BYOL 시스템을 AWS 라이선스가 포함된 인스턴스로 마이그레이션하기

작성자: Mike Kuznetsov (AWS)

환경: 프로덕션	소스: RHEL BYOL 인스턴스 (온프레미스 또는 기타 클라우드 환경)	대상: AWS 라이선스가 포함된 RHEL 인스턴스
R 유형: 리호스트	워크로드: 기타 모든 워크로드	기술: 운영 체제, 인프라, 마이그레이션
AWS 서비스: AWS 애플리케이션 마이그레이션 서비스		

요약

AWS Application Migration Service(AWS MGN)를 사용하여 워크로드를 AWS로 마이그레이션하는 경우, 마이그레이션 중에 Red Hat Enterprise Linux(RHEL) 인스턴스를 리프트 앤드 시프트(리호스팅)하고 라이선스를 기본 보유 라이선스 사용(BYOL) 모델에서 AWS 라이선스가 포함된(LI) 모델로 변경해야 할 수 있습니다. AWS MGN은 Amazon Machine Image(AMI) ID를 사용하는 확장 가능한 접근 방식을 지원합니다. 이 패턴은 대규모 리호스트 마이그레이션 중에 RHEL 서버에서 라이선스 변경을 수행하는 방법을 설명합니다. 또한 Amazon Elastic Compute Cloud(Amazon EC2)에서 이미 실행 중인 RHEL 시스템의 라이선스를 변경하는 방법도 설명합니다.

사전 조건 및 제한 사항

사전 조건

- 대상 AWS 계정에 대한 액세스
- 마이그레이션을 위해 대상 AWS 계정 및 리전에서 AWS MGN이 초기화됨 (온프레미스 시스템에서 AWS로 이미 마이그레이션한 경우 필요 없음)
- 유효한 RHEL 라이선스가 있는 소스 RHEL 서버

아키텍처

이 패턴은 다음 두 가지 시나리오를 다룹니다.

- AWS MGN을 사용하여 온프레미스에서 AWS LI 인스턴스로 직접 시스템을 마이그레이션합니다. 이 시나리오의 경우 첫 번째 에픽(LI 인스턴스로 마이그레이션 - 옵션 1)과 세 번째 에픽의 지침을 따르십시오.
- Amazon EC2에서 이미 실행 중인 이전에 마이그레이션된 RHEL 시스템의 라이선스 모델을 BYOL에서 LI로 변경합니다. 이 시나리오의 경우 첫 번째 에픽(LI 인스턴스로 마이그레이션 - 옵션 2)과 세 번째 에픽의 지침을 따르세요.

참고: 마지막 에픽은 AWS에서 제공하는 Red Hat Update Infrastructure (RHUI) 서버를 사용하도록 새 RHEL 인스턴스를 재구성하는 것입니다. 이 프로세스는 두 시나리오에서 동일합니다.

도구

서비스

- [AWS Application Migration Service \(AWS MGN\)](#)를 사용하면 변경 없이 다운타임을 최소화하면서 애플리케이션을 AWS 클라우드로 리호스팅(리프트 앤드 시프트) 할 수 있습니다.

에픽

LI 인스턴스로 마이그레이션 - 옵션 1 (온프레미스 RHEL 시스템용)

작업	설명	필요한 기술
대상 리전에서 RHEL AWS LI 인스턴스의 AMI ID를 찾습니다.	AWS Marketplace 를 방문하거나 Amazon EC2 콘솔 을 사용하여 RHEL 소스 시스템 버전(예: RHEL-7.7)과 일치하는 RHEL AMI ID를 찾아 AMI ID를 기록해 둡니다. Amazon EC2 콘솔에서는 다음 검색어 중 하나를 사용하여 AMI를 필터링할 수 있습니다.	클라우드 관리자

작업	설명	필요한 기술
	<ul style="list-style-type: none">• Description = Provided by Red Hat, Inc.• AMI name = RHEL-7.7	

작업	설명	필요한 기술
<p>AWS MGN 시작 설정을 구성합니다.</p>	<ol style="list-style-type: none"> 1. AWS MGN 콘솔에서 소스 RHEL 시스템을 추가합니다. AWS MGN 설명서의 지침에 따라 AWS 복제 에이전트를 설치하고 소스 서버를 추가합니다. 2. 소스 서버 페이지에서 소스 RHEL 시스템을 선택한 다음 시작 설정 탭을 선택합니다. 3. 일반 시작 설정 섹션에서 편집을 선택합니다. 자동 선택을 비활성화하고 대상 인스턴스 유형을 수동으로 지정하려면 인스턴스 유형 적정 크기 조정을 없애므로 변경한 다음 설정 저장을 선택합니다. 이렇게 하면 Amazon EC2 시작 템플릿에서 구성된 인스턴스 유형을 사용할 수 있습니다. 자세한 내용은 AWS MGN 설명서를 참조하세요. 4. EC2 시작 템플릿 섹션에서 수정을 선택합니다. EC2 시작 템플릿 수정 정보 대화 상자에서 수정을 다시 선택합니다. 그러면 Amazon EC2 콘솔이 열리고 이 인스턴스의 템플릿을 변경할 수 있습니다. 5. AWS MGN 설명서의 주요 고려 사항을 검토하십시오. 	<p>클라우드 관리자</p>

작업	설명	필요한 기술
	<p>참고: AMI를 직접 선택하라는 경고는 무시해도 됩니다.</p> <p>6. Amazon EC2 콘솔의 새 시작 템플릿에서 다음을 수정합니다.</p> <ul style="list-style-type: none"> AMI에는 이전에 파악한 AMI ID를 지정하거나 RHEL-x를 검색하고 필요한 버전(예: RHEL-7.7)을 지정합니다. 인스턴스 유형에는 원하는 대상 인스턴스 유형을 설정합니다. 키 쌍(로그인), 네트워크 설정(대상 서브넷 및 보안 그룹을 지정하려는 경우 제외), 스토리지, 리소스 태그(태그를 추가하거나 수정하려는 경우 제외) 섹션은 변경하지 말고 그대로 두십시오. (선택 사항) 향후 AWS Systems Manager에서 관리하는 데 필요한 경우 고급 세부 정보 섹션에서 IAM 인스턴스 프로파일 역할을 지정합니다. <p>7. 템플릿 버전 생성을 선택한 다음, 성공 메시지의 링크를 선택하여 시작 템플릿을 확인합니다.</p> <p>8. 작업, 기본 버전 설정을 선택합니다. 템플릿 버전의 경우</p>	

작업	설명	필요한 기술
	<p>최신 버전(새 시스템의 경우 버전 2)을 선택한 다음 기본 버전으로 설정을 선택합니다.</p> <p>이제 AWS MGN이 이 버전의 시작 템플릿을 사용하여 테스트 또는 컷오버 인스턴스를 시작합니다. 자세한 내용은 AWS MGN 설명서를 참조하세요.</p>	
<p>설정을 확인합니다.</p>	<ol style="list-style-type: none"> 1. AWS MGN 콘솔의 소스 서버 페이지에서 소스 서버를 선택한 다음 시작 설정 탭을 선택합니다. 2. EC2 시작 템플릿 섹션에서 인스턴스 유형, 서브넷 및 보안 그룹 매개변수가 올바르게 설정되었는지 확인합니다. <p>참고: 이 섹션에는 선택한 AMI ID가 표시되지 않습니다. 이 ID를 보려면 Amazon EC2 콘솔을 열고, 시작 템플릿 보기를 열고, 이 섹션에 표시된 템플릿 ID를 검색하면 됩니다.</p>	<p>클라우드 관리자</p>

작업	설명	필요한 기술
새 NI 인스턴스를 시작합니다.	<ol style="list-style-type: none"> 1. 초기 동기화가 완료되면 AWS MGN 콘솔 소스 서버 페이지에 있는 서버의 마이그레이션 수명 주기 열이 테스트 준비 완료됨으로 변경됩니다. 새 테스트 인스턴스를 시작하려면 소스 서버를 선택하고 테스트 및 컷오버 메뉴를 연 다음 테스트 인스턴스 시작을 선택합니다. 작업 세부 정보 보기를 선택하여 시작 작업의 상태를 모니터링합니다. 자세한 내용은 AWS MGN 설명서를 참조하세요. 2. 시작 작업이 완료될 때까지 기다린 다음, 시작된 EC2 인스턴스 세부 정보 페이지를 엽니다. 세부 정보 탭을 선택하고 인스턴스 세부 정보 섹션에 다음 내용이 포함되어 있는지 확인합니다. <ul style="list-style-type: none"> • 플랫폼 세부 정보: “Red Hat Enterprise Linux” • AMI 이름: EC2 시작 템플릿에서 지정한 AMI 이름 3. AWS MGN 설명서의 지침에 따라 새 NI 인스턴스로 컷오버합니다. 4. 마지막 에픽의 단계에 따라 AWS에서 제공한 RHUI 서버를 사용하도록 새 인스턴스를 재구성합니다. 	클라우드 관리자

LI 인스턴스로 마이그레이션 - 옵션 2 (RHEL BYOL EC2 인스턴스용)

작업	설명	필요한 기술
<p>RHEL BYOL EC2 인스턴스를 AWS LI 인스턴스로 마이그레이션합니다.</p>	<p>디스크(Amazon Elastic Block Store 볼륨)를 이동하고 새 LI 인스턴스에 연결하여 이전에 BYOL로 AWS로 마이그레이션한 RHEL 시스템을 AWS LI 인스턴스로 전환할 수 있습니다. 전환하려면 다음 단계를 따르십시오.</p> <ol style="list-style-type: none"> 1. RHEL LI AMI에서 새 대상 RHEL 인스턴스를 시작합니다. (현재 RHEL 인스턴스와 동일한 RHEL 버전을 사용하십시오.) 2. 두 인스턴스(새 LI 인스턴스와 원본 소스 인스턴스)를 모두 중지합니다. 3. 새 LI 인스턴스에서 모든 EBS 볼륨(루트 디스크 포함)을 분리하고 삭제합니다. 4. 이전 소스 인스턴스에서 모든 EBS 볼륨(루트 디스크 포함)을 분리하여 새 LI 인스턴스에 연결합니다. 디바이스에 대한 볼륨 매핑을 동일하게 유지하십시오. (예를 들어, 이전에 /dev/sda 드라이브에 연결된 EBS 볼륨을 새 인스턴스에 /dev/sda로 연결해야 합니다.) 5. 소스(현재는 디스크가 없는) 인스턴스를 삭제합니다. 	<p>클라우드 관리자</p>

작업	설명	필요한 기술
	6. 새 EC2 인스턴스를 시작합니다. 인스턴스에 로그인하고 다음 에픽의 단계에 따라 AWS에서 제공하는 RHUI 서버를 사용하도록 재구성합니다.	

AWS에서 제공하는 RHUI를 사용하도록 RHEL OS를 재구성하기 — 두 옵션 모두

작업	설명	필요한 기술
Red Hat 서브스크립션 및 라이선스에서 OS 등록을 취소합니다.	<p>마이그레이션과 성공적인 컷오버 후에는 Red Hat 라이선스 사용을 중단하고 이중 청구를 방지하려면 Red Hat 구독에서 RHEL 시스템을 제거해야 합니다.</p> <p>Red Hat 구독에서 RHEL OS를 제거하려면 Red Hat 구독 관리 (RHSM) 설명서에 설명된 프로세스를 따르십시오. CLI 명령을 사용합니다.</p> <pre>subscription-manager unregister</pre> <p>구독 관리자 플러그인을 비활성화하여 모든 yum 호출에서 구독 상태 확인을 중지할 수도 있습니다. 이렇게 하려면 구성 파일 <code>/etc/yum/pluginconf.d/subscription-manager.conf</code> 을 편집하고 매개변수 <code>enabled=1</code> 을</p>	Linux 또는 시스템 관리자

작업	설명	필요한 기술
	enabled=0 으로 변경하십시오.	

작업	설명	필요한 기술
<p>이전 업데이트 구성(RHUI, Red Hat Satellite 네트워크, yum 리포지토리)을 AWS에서 제공하는 RHUI로 교체합니다.</p>	<p>AWS에서 제공한 RHUI 서버를 사용하도록 마이그레이션된 RHEL 시스템을 재구성해야 합니다. 이렇게 하면 외부 업데이트 인프라 없이도 AWS 리전 내의 RHUI 서버에 액세스할 수 있습니다. 이 변경 사항에는 다음 프로세스가 포함됩니다.</p> <ol style="list-style-type: none"> 1. 기존 yum 구성을 백업합니다. 2. 이전 RHUI(yum 리포지토리) 구성 및 패키지를 제거합니다. 3. AWS에서 제공하는 새 RHUI 구성 및 인증서 패키지를 추가합니다. 이러한 구성 패키지는 AWS에서 제공한 RHUI 서버에서만 사용할 수 있으므로 AWS의 다른 RHEL 인스턴스에서 검색해야 합니다. <p>자세한 단계 및 명령은 다음과 같습니다.</p> <ol style="list-style-type: none"> 1. 모든 <code>/etc/yum*</code> 및 <code>/etc/pki/*</code> 폴더를 백업 위치에 복사하여 기존 yum 구성 및 인증서를 백업합니다. 예: <pre data-bbox="634 1738 1029 1871">mkdir yum-backup cp -ra /etc/yum* /etc/pki ./yum-backup</pre>	<p>Linux 또는 시스템 관리자</p>

작업	설명	필요한 기술
	<pre data-bbox="630 205 1026 304">tar czf yum-backup.p.tgz ./yum-backup</pre> <p data-bbox="591 321 1019 405">2. 다음과 같은 이전 RHUI 구성 및 패키지를 제거합니다.</p> <p data-bbox="630 426 1026 510">a. 다음과 같은 설치된 모든 RHUI 패키지를 찾습니다.</p> <pre data-bbox="669 548 1003 667">sudo rpm -qa grep rhui</pre> <p data-bbox="630 678 1019 762">b. 다음과 같은 패키지를 삭제합니다.</p> <pre data-bbox="669 800 1003 961">sudo yum remove \$(rpm -qa grep rhui)</pre> <p data-bbox="630 972 1026 1150">c. <code>/etc/yum/vars/releasever</code> 파일이 있으면 이 파일을 제거합니다.</p> <p data-bbox="591 1171 1026 1591">3. AWS에서 제공하는 RHUI 및 인증서 패키지를 추가합니다. 이들은 AWS의 다른 RHEL 인스턴스에서 검색해야 합니다. 이를 달성하는 데는 몇 가지 방법이 있습니다. 예를 들어, Red Hat 지식베이스 문서에 제공된 지침을 따르면 됩니다.</p> <p data-bbox="630 1612 1019 1749">a. AWS 마켓플레이스에서 다른 RHEL(RHEL-EC2) 인스턴스를 시작합니다.</p> <p data-bbox="630 1770 1019 1854">b. 이 인스턴스에서 두 개의 패키지, 즉 최신 RHUI 클</p>	

작업	설명	필요한 기술
	<p>라이언트 구성 패키지와 인증 기관(CA) 인증서를 다운로드합니다. 예를 들어, 데스크톱에서 다음 명령을 실행합니다.</p> <pre data-bbox="667 474 1027 705">ssh RHEL-EC2 "sudo yumdownloader ca-certificates rh-amazon-rhui-client"</pre> <p>c. RHEL-EC2 인스턴스에서 마이그레이션된 새 시스템으로 패키지를 복사합니다. 예:</p> <pre data-bbox="667 947 1027 1457">scp RHEL-EC2:rh-amazon-rhui-client* RHEL-EC2:ca-certificates* . ssh <migrated-instance> "mkdir /tmp/amazon" scp rh-amazon-rhui-client* ca-certificates* <migrated-instance>:/tmp/amazon</pre> <p>d. 마이그레이션된 인스턴스에 새 RHUI 및 CA 구성 패키지를 설치합니다.</p> <pre data-bbox="667 1646 1027 1839">ssh <migrated-instance> "sudo rpm -Uhv /tmp/amazon/*"</pre>	

작업	설명	필요한 기술
구성을 확인합니다.	<p>마이그레이션된 대상 인스턴스에서 새 구성이 올바른지 확인합니다.</p> <pre data-bbox="597 394 1026 512">sudo yum clean all sudo yum repolist</pre>	Linux 또는 시스템 관리자

관련 리소스

- [AWS Application Migration Service\(AWS MGN\) 사용 설명서](#)
- [IMDSv2를 지원하는 AWS RHUI 클라이언트 패키지 받기](#)(Red Hat 지식베이스 문서)
- [Amazon EC2 시작 템플릿](#)(Amazon EC2 설명서)

Microsoft SQL Server를 AWS 클라우드로 마이그레이션한 후 연결 오류 해결

작성자: Premkumar Chelladurai

환경: 프로덕션

기술: 운영 체제, 마이그레이션

워크로드: Microsoft

AWS 서비스: Amazon EC2

요약

Windows Server 2008 R2, 2012 또는 2012 R2에서 실행되는 Microsoft SQL Server를 Amazon Web Services(AWS) 클라우드의 Amazon Elastic Compute Cloud(Amazon EC2) 인스턴스로 마이그레이션 하면 SQL Server로의 연결이 실패하고 다음과 같은 오류가 나타납니다.

- [Microsoft][ODBC SQL Server Driver][DBNETLIB] General Network error
- ERROR [08S01] [Microsoft][SQL Native Client]Communication link failure. System.Data.SqlClient.SqlException: A transport-level error has occurred when sending the request to the server. (provider: TCP Provider, error: 0 - An existing connection was forcibly closed by the remote host.)
- TCP Provider: The semaphore timeout period has expired

이 패턴은 Windows Server 2008 R2, 2012 또는 2012 R2에서 실행되는 SQL Server의 운영 체제(OS) 및 네트워크 인터페이스 수준에서 Windows Scalable Networking Pack(SNP) 기능을 꺼서 이러한 오류를 해결하는 방법을 설명합니다.

사전 조건 및 제한 사항

사전 조건

- Windows 서버에 대한 관리자 권한.
- 애플리케이션 마이그레이션 서비스를 마이그레이션 도구로 사용한 경우 다음 Windows 서버 버전 중 하나가 필요합니다.
 - Windows Server 2008 R2 서비스 팩 1, 2012, 또는 2012 R2

- CloudEndure 마이그레이션을 마이그레이션 도구로 사용한 경우 다음 Windows Server 버전 중 하나가 필요합니다.
 - Windows Server 2003 R2 서비스 팩 3, 2008, 2008 R2 Service Pack 1, 2012, 또는 2012 R2

도구

- [Amazon EC2](#) – Amazon Elastic Compute Cloud(Amazon EC2)는 AWS 클라우드에서 확장 가능한 컴퓨팅 용량을 제공합니다. Amazon EC2를 사용하여 필요한 만큼 많거나 적은 수의 가상 서버를 시작하고 스케일 아웃하거나 스케일 인할 수 있습니다.
- [Windows Server](#) — Windows Server는 연결된 애플리케이션, 네트워크, 웹 서비스의 인프라를 빌드하기 위한 플랫폼입니다.

에픽

OS 및 Elastic Network 인터페이스 수준에서 SNP 기능을 끕니다.

작업	설명	필요한 기술
OS 수준에서 SNP 기능을 끕니다.	<ol style="list-style-type: none"> 1. 관리자 권한으로 Windows 서버에 로그인하고 명령 프롬프트를 엽니다. 2. <code>netsh int tcp show global</code> 명령을 실행합니다. 3. 출력에서 Receive-Side Scaling 또는 Chimney Offload(이)가 enabled 모드에 있는지 확인합니다. 둘 중에 하나가 enabled인 경우 다음과 같은 명령을 실행합니다. <ul style="list-style-type: none"> • <code>netsh int tcp set global chimney=disabled</code> 	AWS 관리자, AWS 시스템 관리자, 마이그레이션 엔지니어, 클라우드 관리자

작업	설명	필요한 기술
	<ul style="list-style-type: none"> • netsh int tcp set global rss=disabled 	
탄력적 네트워크 인터페이스 수준에서 SNP 기능을 끕니다.	<ol style="list-style-type: none"> 1. 시작을 선택하고, ncpa.cpl을(를)입력, 그 다음 에키를 누릅니다. 2. Elastic Network Adapter를 마우스 오른쪽 버튼으로 클릭합니다. 3. 팝업 메뉴에서 속성을 선택합니다. 4. Ethernet Adapter 속성 창에서 구성을 선택합니다. 5. Amazon Elastic Network Adapter 속성 팝업 창에서 고급 탭을 선택합니다. 6. 속성 섹션에서 모든 오프로드 및 RSS를 끕니다. 	AWS 관리자, 클라우드 관리자, AWS 시스템 관리자

관련 리소스

- [RSS 및 NetDMA와 같은 고급 네트워크 성능 기능의 문제를 해결하는 방법](#)

패턴 더 보기

- [AWS Cloud의 Stromasys Charon-SSP 에뮬레이터에서 Sun SPARC 서버 백업하기](#)
- [기본 백업 및 복원 수단을 사용하여 온프레미스 Microsoft SQL Server 데이터베이스를 Amazon RDS for SQL Server로 마이그레이션](#)
- [고가용성 재해 복구 기능을 갖춘 Db2 for LUW를 Amazon EC2로 마이그레이션하세요.](#)
- [AWS 서비스를 사용하여 SAP RHEL 페이스메이커 클러스터를 모니터링합니다.](#)
- [RHEL 소스 서버를 재부팅한 후 SELinux를 비활성화하지 않고 Replication Agent를 자동으로 다시 시작](#)

운영

주제

- [Python을 사용하여 AMS에서 자동으로 RFC 생성](#)
- [클라우드 운영 모델을 위한 RACI 또는 RASCI 지표 생성](#)
- [기본 암호화가 적용된 Amazon EBS 볼륨을 사용하는 AWS Cloud9 IDE를 생성](#)
- [태그 기반 Amazon CloudWatch 대시보드를 자동으로 생성](#)
- [AWS Config 고급 쿼리를 사용하여 생성 날짜를 기준으로 AWS 리소스 검색](#)
- [계정 또는 조직의 EBS 스냅샷 세부 정보 보기](#)
- [패턴 더 보기](#)

Python을 사용하여 AMS에서 자동으로 RFC 생성

작성자: Gnanasekaran Kailasam(AWS)

환경: 프로덕션

기술: 운영, 클라우드 네이티브

AWS 서비스: AWS Managed Services

요약

AWS Managed Services(AMS)는 Amazon Web Services(AWS) 인프라를 지속적으로 관리하여 클라우드 기반 인프라를 보다 효율적이고 안전하게 운영할 수 있도록 지원합니다. 관리형 환경을 변경하려면 특정 운영 또는 작업에 대한 변경 유형(CT) ID가 포함된 새 변경 요청(RFC)을 생성하여 제출해야 합니다.

하지만 RFC를 수동으로 생성하는 데는 약 5분이 소요될 수 있으며 조직의 팀이 매일 여러 개의 RFC를 제출해야 할 수도 있습니다. 이 패턴을 사용하면 RFC 생성 프로세스를 자동화하고, 각 RFC의 생성 시간을 줄이고, 수동 오류를 제거하는 데 도움이 됩니다.

이 패턴은 Python 코드를 사용하여 AMS 계정에서 Amazon Elastic Compute Cloud(Amazon EC2) 인스턴스를 중지하는 Stop EC2 instance RFC를 자동으로 생성하는 방법을 설명합니다. 그런 다음 이 패턴의 접근 방식과 Python 자동화를 다른 RFC 유형에 적용할 수 있습니다.

사전 조건 및 제한 사항

사전 조건

- AMS 고급 계정. 이에 대한 자세한 내용은 AWS Managed Services 설명서의 [AMS 운영 계획](#)을 참조하십시오.
- AMS 계정에 기존 EC2 인스턴스가 하나 이상 있어야 합니다.
- AMS에서 RFC를 생성하고 제출하는 방법에 대한 이해.
- Python에 대해 숙지.

제한 사항

- RFC는 AMS 계정 변경 시에만 사용할 수 있습니다. AWS 계정은 유사한 변경 사항에 대해 서로 다른 프로세스를 사용합니다.

아키텍처

기술 스택

- AMS
- AWS Command Line Interface(AWS CLI)
- AWS SDK for Python(Boto3)
- Python과 그 필수 패키지(JSON과 Boto3)

자동화 및 규모 조정

이 패턴은 Stop EC2 instance RFC를 자동화하는 샘플 코드를 제공하지만 이 패턴의 샘플 코드와 접근 방식을 다른 RFC에도 사용할 수 있습니다.

도구

- [AWS Managed Services\(AMS\)](#) - AMS는 AWS 인프라를 보다 효율적이고 안전하게 운영하는 데 도움이 됩니다.
- [AWS CLI](#) - AWS Command Line Interface(AWS CLI)는 AWS 서비스를 관리하는 통합 도구입니다. AMS에서 변경 관리 API는 RFC를 생성하고 관리하는 작업을 제공합니다.
- [AWS SDK for Python\(Boto3\)](#) - Python용 SDK를 사용하면 Python 애플리케이션, 라이브러리 또는 스크립트를 AWS 서비스와 쉽게 통합할 수 있습니다.

코드

AMS Stop EC2 Instance.zip 파일(첨부)에는 Stop EC2 instance RFC를 만들기 위한 Python 코드가 들어 있습니다. 여러 EC2 인스턴스에 대해 단일 RFC를 제출하도록 이 코드를 구성할 수도 있습니다.

에픽

옵션 1 - MacOS 또는 Linux용 환경 설정

작업	설명	필요한 기술
Python을 설치하고 유효성을 검사합니다.	<ol style="list-style-type: none"> 1. 터미널 창을 열고 <code>brew install python3</code> 명령을 실행합니다. 2. <code>python --version</code> 명령을 실행하여 Python이 제대로 설치되었는지 확인합니다. 3. <code>pip --version</code> 명령을 실행하여 pip이 제대로 설치되었는지 확인합니다. 	AWS 시스템 관리자
AWS CLI를 설치합니다.	<code>pip install awscli --upgrade -user</code> 명령을 실행하여 AWS CLI를 설치합니다.	AWS 시스템 관리자
Boto3을 설치합니다.	<code>pip install boto3</code> 명령을 실행하여 Boto3을 설치합니다.	AWS 시스템 관리자
JSON을 설치합니다.	<code>pip install json</code> 명령을 실행하여 JSON을 설치합니다.	AWS 시스템 관리자
AMS CLI를 설정합니다.	<p>AWS Management Console에 로그인하여 AMS 콘솔을 열고 설명서를 선택합니다. AMS CLI가 포함된.zip 파일을 다운로드하고 압축을 해제한 다음 로컬 머신에 설치합니다.</p> <p>AMS CLI를 설치한 후 <code>aws amscm help</code> 명령을 실행합니다. 출력은 AMS 변경 관리 프</p>	AWS 시스템 관리자

작업	설명	필요한 기술
	로세스에 대한 정보를 제공합니다.	

옵션 2 - Windows용 환경 설정

작업	설명	필요한 기술
Python을 설치하고 유효성을 검사합니다.	<ol style="list-style-type: none"> Windows용 Python 릴리스 페이지를 열고 최신 버전을 다운로드한 다음 Python을 설치합니다. <code>python --version</code> 명령을 실행하여 Python이 제대로 설치되었는지 확인합니다. <code>pip --version</code> 명령을 실행하여 pip이 제대로 설치되었는지 확인합니다. 	AWS 시스템 관리자
AWS CLI를 설치합니다.	<code>pip install awscli --upgrade -user</code> 명령을 실행하여 AWS CLI를 설치합니다.	AWS 시스템 관리자
Boto3을 설치합니다.	<code>pip install boto3</code> 명령을 실행하여 Boto3을 설치합니다.	AWS 시스템 관리자
JSON을 설치합니다.	<code>pip install json</code> 명령을 실행하여 JSON을 설치합니다.	AWS 시스템 관리자
AMS CLI를 설정합니다.	AWS Management Console에 로그인하여 AMS 콘솔을 열고 설명서를 선택합니다. AMS CLI가 포함된.zip 파일을 다운로드하고 압축을 해제한 다음 로컬 머신에 설치합니다.	AWS 시스템 관리자

작업	설명	필요한 기술
	AMS CLI를 설치한 후 <code>aws amscm help</code> 명령을 실행합니다. 출력은 AMS 변경 관리 프로세스에 대한 정보를 제공합니다.	

RFC의 CT ID 및 실행 파라미터를 추출합니다.

작업	설명	필요한 기술
RFC의 CT ID, 버전 및 실행 파라미터를 추출합니다.	<p>각 RFC에는 서로 다른 CT ID, 버전 및 실행 파라미터가 있습니다. 다음 옵션 중 하나를 사용하여 이 정보를 추출할 수 있습니다.</p> <ol style="list-style-type: none"> 1. AWS Managed Services 설명서의 RFC 사용 예제에서 CLI를 사용한 변경 요청 찾기(RFC) 섹션의 지침을 따르십시오. 2. AMS 콘솔을 통해 유사한 유형의 기존 RFC를 열거나 새 RFC를 생성하여 테스트하십시오. RFC의 CT ID 및 실행 파라미터를 사용하십시오. 이에 대한 자세한 내용은 AWS Managed Services 설명서에서 콘솔을 사용한 RFC 찾기를 참조하십시오. <p>참고: 이 패턴의 Python 자동화를 다른 RFC에 적용하려면 AMS Stop EC2</p>	AWS 시스템 관리자

작업	설명	필요한 기술
	Instance.zip 파일(첨부)의 <code>ams_stop_ec2_instance</code> Python 코드 파일에 있는 CT 유형 및 파라미터 값을 추출한 값으로 바꾸십시오.	

Python 자동화 실행

작업	설명	필요한 기술
Python 자동화를 실행합니다.	<ol style="list-style-type: none"> AMS Stop EC2 Instance.zip 파일(첨부)을 로컬 머신에 다운로드 하고 파일을 추출합니다. EC2 인스턴스 정보로 <code>input_instances</code> 를 업데이트하십시오. 터미널을 열고 추출된 코드의 경로로 이동합니다. <code>pythonams_stop_ec2_instance.py</code> 명령을 실행합니다. 	AWS 시스템 관리자

관련 리소스

- [변경 유형이란 무엇입니까?](#)
- [CLI 자습서: 고가용성 2계층 스택\(Linux/RHEL\)](#)

첨부

이 문서와 관련된 추가 콘텐츠에 액세스하려면 [attachment.zip](#) 파일의 압축을 풉니다.

클라우드 운영 모델을 위한 RACI 또는 RASCI 지표 생성

작성자: Teddy Germade(AWS), Jerome Descreux(AWS), Josselin LE MINEUR(AWS), Florian Leroux(AWS)

환경: 프로덕션

기술: 운영, 관리 및 거버넌스

요약

클라우드 혁신 센터(CCoE) 또는 CEE(클라우드 지원 엔진)는 클라우드의 운영 준비에 중점을 두고 권한을 부여하고 책임을 다하는 팀입니다. 주요 중점 사항은 정보 IT 조직을 온프레미스 운영 모델에서 클라우드 운영 모델로 전환하는 것입니다. CCoE는 인프라, 애플리케이션, 운영 및 보안 분야를 대표하는 부서 간 팀이어야 합니다.

클라우드 운영 모델의 주요 구성 요소 중 하나는 RACI 매트릭스 또는 RASCI 매트릭스입니다. 이는 마이그레이션 활동 및 클라우드 운영과 관련된 모든 당사자의 역할과 책임을 정의하는 데 사용됩니다. 매트릭스 이름은 매트릭스에 정의된 책임 유형인 책무(R), 책임(A), 지원(S), 상담(C), 정보 제공(I)에서 파생됩니다. 지원 유형은 선택사항입니다. 이를 포함하면 RASCI 지표라고 하고, 제외하면 RACI 지표라고 합니다.

CCoE 팀은 첨부된 템플릿을 사용하여 조직에 맞는 RACI 또는 RASCI 지표를 생성할 수 있습니다. 템플릿에는 클라우드 운영 모델에서 흔히 사용되는 팀, 역할 및 작업이 포함되어 있습니다. 이 매트릭스의 기본은 운영 통합 및 CCoE 기능과 관련된 작업입니다. 그러나 조직의 구조 및 사용 사례의 요구 사항에 맞게 이 템플릿을 사용자 지정할 수 있습니다.

RACI 매트릭스 구현에는 제한이 없습니다. 이 접근 방식은 대규모 조직, 신생 기업 및 그 사이에 있는 모든 기업에 적합합니다. 소규모 조직의 경우 동일한 리소스가 여러 역할을 수행할 수 있습니다.

에픽

지표 생성

작업	설명	필요한 기술
주요 이해관계자를 파악합니다.	클라우드 운영 모델의 전략적 목표와 관련된 주요 서비스 및 팀 관리자를 식별합니다.	프로젝트 관리자

작업	설명	필요한 기술
<p>지표 템플릿을 사용자 지정합니다.</p>	<p>첨부 파일 섹션에서 템플릿을 다운로드한 다음 다음과 같이 RACI 또는 RASCI 매트릭스를 업데이트합니다.</p> <ul style="list-style-type: none"> • 클라우드 팀 워크시트에서 조직의 필요에 따라 CCoE 스트림 이름, 팀 이름, 팀 설명을 업데이트합니다. • 클라우드 역할 워크시트에서 조직의 필요에 따라 역할, 팀 이름, 역할 설명을 업데이트합니다. • RASCI 워크시트에서 조직의 필요에 따라 다음 사항을 업데이트합니다. <ul style="list-style-type: none"> • 1행과 A열에서 CCoE 스트림을 업데이트합니다. • 2행에서 팀 이름을 업데이트합니다. • 3행에서 역할 이름을 업데이트합니다. • D열과 E열에서 RASCI 차트에 포함하려는 일반 필드와 활동을 업데이트합니다. 	<p>프로젝트 관리자</p>
<p>회의를 계획합니다.</p>	<ol style="list-style-type: none"> 1. RASCI 목표를 모든 이해 관계자에게 전달합니다. 2. 각 팀의 권한이 있는 담당자가 참석할 수 있도록 회의를 한 번 이상 계획하세요. 	<p>프로젝트 관리자</p>

작업	설명	필요한 기술
지표를 작성합니다.	<p>모든 이해 관계자와의 회의에서 다음 작업을 수행합니다.</p> <ol style="list-style-type: none"> 1. 각 팀의 담당자가 참석하는지 확인합니다. 각 작업에 책임 유형을 정확하게 지정할 수 있으려면 팀 참여가 필수입니다. 2. 참가자와 함께 RASCI 매트릭스가 무엇인지, 목표를 검토하세요. 3. 참가자와 함께 공동 책임 모델을 검토하여 참가자가 클라우드 보안에 대한 조직의 책임 범위를 이해할 수 있도록 하세요. 4. RASCI 워크시트의 각 작업 또는 활동에 대해 F~AN 열을 작성하여 다음 책임 유형을 지정합니다. <ul style="list-style-type: none"> • 책무(R) - 이 역할은 작업을 완료하기 위한 작업 수행을 담당합니다. • 책임(A) - 이 역할은 작업이 완료되었는지 확인하는 역할을 담당합니다. 또한 이 역할은 사전 조건이 충족되었는지 확인하고 책임자에게 작업을 위임하는 역할도 담당합니다. • 지원(S) - 이 역할은 책임자가 작업을 완료하는 데 도움이 됩니다. 이 책임 유형은 선택 사항이며, 보다 	프로젝트 관리자

작업	설명	필요한 기술
	<p>전통적인 RACI 매트릭스를 생성하기 위해 제외하도록 선택할 수 있습니다.</p> <ul style="list-style-type: none"> • 상담(C) - 작업에 대한 의견이나 전문 지식이 필요한 경우 이 역할을 참조해야 합니다. 작업에 따라 이 책임 유형이 필요하지 않을 수도 있습니다. • 알림(I) - 이 역할은 작업 진행 상황을 최신 상태로 유지하고 작업이 완료되면 알림을 받아야 합니다. • 공백 - 이 역할은 활동이나 작업에 관여하지 않습니다. 	
<p>RASCI 매트릭스를 공유합니다.</p>	<p>RACI 또는 RASCI 지표가 완성되면 경영진의 승인을 받습니다. 모든 이해 관계자가 액세스할 수 있는 공유 리포지토리 또는 중앙 위치에 저장합니다. 표준 문서 관리 프로세스를 사용하여 매트릭스의 수정 내용을 기록하고 승인하는 것이 좋습니다.</p>	<p>프로젝트 관리자</p>

관련 리소스

- [AWS Shared Responsibility Model](#)

첨부

이 문서와 관련된 추가 콘텐츠에 액세스하려면 [attachment.zip](#) 파일의 압축을 풉니다.

기본 암호화가 적용된 Amazon EBS 볼륨을 사용하는 AWS Cloud9 IDE를 생성

자나르단 말리알라 (AWS) 와 드루바쥬티 무케르지 (AWS) 가 제작했습니다.

환경: 프로덕션	기술: 운영	워크로드: 기타 모든 워크로드
AWS 서비스: AWS Cloud9, AWS KMS		

요약

[암호화를 기본으로](#) 사용하여 Amazon Web Services(AWS) 클라우드에서 Amazon Elastic Block Store(Amazon EBS) 볼륨과 스냅샷 복사본을 암호화할 수 있습니다.

기본적으로 암호화된 EBS 볼륨을 사용하는 AWS Cloud9 통합 개발 환경(IDE)을 생성할 수 있습니다. 하지만 AWS Identity and Access Management(IAM) [서비스 연결 역할](#)을 AWS Cloud9에 사용하려면 이러한 EBS 볼륨의 AWS Key Management Service(AWS KMS) 키에 대한 액세스 권한이 필요합니다. 이 액세스 권한이 제공되지 않으면 AWS Cloud9 IDE가 시작되지 않고 디버깅이 어려울 수 있습니다.

이 패턴은 EBS 볼륨에서 사용되는 AWS KMS 키에 AWS Cloud9의 서비스 연결 역할을 추가하는 단계를 제공합니다. 이 패턴이 설명하는 설정은 기본적으로 암호화가 적용된 EBS 볼륨을 사용하는 IDE를 성공적으로 생성하고 시작하는 데 도움이 됩니다.

사전 조건 및 제한 사항

사전 조건

- 활성 상태의 AWS 계정.
- 기본 암호화가 켜져 있는 EBS 볼륨. 기본 암호화에 대한 자세한 내용은 Amazon Elastic Compute Cloud(Amazon EC2) 설명서의 [Amazon EBS 암호화](#)를 참조하세요.
- EBS 볼륨을 암호화하기 위한 기존 [고객 관리형 KMS 키](#).

참고: AWS Cloud9의 서비스 연결 역할을 생성할 필요가 없습니다. AWS Cloud9 개발 환경을 생성할 때 이 서비스 연결 역할을 자동으로 생성합니다.

아키텍처

기술 스택

- AWS Cloud9
- IAM
- AWS KMS

도구

- [AWS Cloud9](#)는 소프트웨어를 코딩, 구축, 실행, 테스트 및 디버깅할 수 있게 해주는 통합 개발 환경(IDE)입니다. 또한, 소프트웨어를 AWS 클라우드로 릴리스하는 데도 도움이 됩니다.
- [Amazon Elastic Block Store\(Amazon EBS\)](#)는 Amazon Elastic Compute Cloud(Amazon EC2) 인스턴스에 사용할 수 있는 블록 스토리지 볼륨을 제공합니다.
- [AWS Identity and Access Management\(IAM\)](#)를 사용하면 사용자에 대해 인증 및 권한 부여를 제어함으로써 AWS 리소스에 대한 액세스를 안전하게 관리할 수 있습니다.
- [AWS Key Management Service\(AWS KMS\)](#)를 사용하면 암호화 키를 생성하고 제어하여 데이터를 보호할 수 있습니다.

에픽

기본 암호화 키값 찾기

작업	설명	필요한 기술
EBS 볼륨의 기본 암호화 키값을 기록합니다.	AWS Management Console에 로그인한 후 Amazon EC2 콘솔을 엽니다. EC2 대시보드를 선택한 다음 계정 속성에서 데이터 보호 및 보안을 선택합니다. EBS 암호화 섹션에서 기본 암호화 키의 값을 복사하여 기록합니다.	클라우드 아키텍트, 엔지니어 DevOps

AWS KMS 키에 대한 액세스 제공

작업	설명	필요한 기술
<p>AWS Cloud9에 EBS 볼륨용 KMS 키에 대한 액세스 권한을 제공하세요.</p>	<ol style="list-style-type: none"> 1. AWS KMS 콘솔을 열고 고객 관리형 키를 선택합니다. Amazon EBS 암호화에 사용되는 AWS KMS 키를 선택한 다음 키 보기를 선택합니다. 2. 키 정책 탭에서 키 정책의 텍스트 양식을 볼 수 있는지 확인합니다. 텍스트 양식이 보이지 않으면 정책 보기로 전환을 선택합니다. 3. 편집을 선택합니다. 추가 정보 섹션의 코드를 정책에 추가한 다음 변경 내용 저장을 선택합니다. 정책 변경으로 AWS Cloud9, AWSServiceRoleForAWSCloud9의 서비스 연결 역할이 키에 액세스할 수 있게 되었습니다. <p>키 정책 업데이트에 대한 자세한 내용은 키 정책 변경 방법(AWS KMS 설명서)을 참조하세요.</p> <p>중요: AWS Cloud9의 서비스 연결 역할은 첫 IDE를 시작할 때 자동으로 생성됩니다. 자세한 내용은 AWS Cloud9 설명서의 서비스 연결 역할 생성 섹션을 참조하세요.</p>	<p>클라우드 아키텍트, DevOps 엔지니어</p>

IDE 생성 및 시작

작업	설명	필요한 기술
AWS Cloud9 IDE를 생성하고 실행합니다.	AWS Cloud9 콘솔을 열고 환경 생성을 선택합니다. AWS Cloud9 설명서의 EC2 환경 생성 단계에 따라 필요한 IDE를 구성합니다.	클라우드 아키텍트, DevOps 엔지니어

관련 리소스

- [AWS Cloud9에서 사용하는 EBS 볼륨 암호화](#)
- [AWS Cloud9에 대한 서비스 연결 역할 생성](#)
- [AWS Cloud9에서 EC2 환경 생성](#)

추가 정보

AWS KMS 키 정책 업데이트

<aws_accountid>를 AWS 계정 ID로 바꿉니다.

```
{
  "Sid": "Allow use of the key",
  "Effect": "Allow",
  "Principal": {
    "AWS": "arn:aws:iam::<aws_accountid>:role/aws-service-role/cloud9.amazonaws.com/AWSServiceRoleForAWSCloud9"
  },
  "Action": [
    "kms:Encrypt",
    "kms:Decrypt",
    "kms:ReEncrypt*",
    "kms:GenerateDataKey*",
    "kms:DescribeKey"
  ],
  "Resource": "*"
},
```

```

    {
      "Sid": "Allow attachment of persistent resources",
      "Effect": "Allow",
      "Principal": {
        "AWS": "arn:aws:iam::<aws_accountid>:role/aws-service-role/
cloud9.amazonaws.com/AWSServiceRoleForAWSCloud9"
      },
      "Action": [
        "kms:CreateGrant",
        "kms:ListGrants",
        "kms:RevokeGrant"
      ],
      "Resource": "*",
      "Condition": {
        "Bool": {
          "kms:GrantIsForAWSResource": "true"
        }
      }
    }
  }
}

```

크로스 어카운트 키 사용

교차 계정 KMS 키를 사용하려면 KMS 키 정책과 함께 권한 부여를 사용해야 합니다. 이렇게 하면 키에 대한 교차 계정 액세스가 가능해집니다. Cloud9 환경을 만들 때 사용한 것과 동일한 계정에서 터미널에서 다음 명령을 실행합니다.

```

aws kms create-grant \
  --region <Region where Cloud9 environment is created> \
  --key-id <The cross-account KMS key ARN> \
  --grantee-principal arn:aws:iam::<The account where Cloud9 environment is
created>:role/aws-service-role/cloud9.amazonaws.com/AWSServiceRoleForAWSCloud9 \
  --operations "Encrypt" "Decrypt" "ReEncryptFrom" "ReEncryptTo" "GenerateDataKey"
"GenerateDataKeyWithoutPlaintext" "DescribeKey" "CreateGrant"

```

이 명령을 실행한 후 다른 계정의 키로 EBS 암호화를 사용하여 Cloud9 환경을 만들 수 있습니다.

태그 기반 Amazon CloudWatch 대시보드를 자동으로 생성

작성: 자낙 바다리아 (AWS), 라즈네시 야기 (AWS), 빈도쿠마르 만달라푸 (AWS)

코드 [리포지토리](#): 골든시그널

환경: 프로덕션

기술: 운영, 클라우드 네이티브

AWS 서비스: AWS CDK,
아마존 CloudWatch, AWS
CodeBuild, AWS CodePipeline

요약

다양한 Amazon CloudWatch 대시보드를 수동으로 생성하는 것은 시간이 많이 걸릴 수 있습니다. 특히 환경을 자동으로 확장하기 위해 여러 리소스를 생성하고 업데이트해야 하는 경우에는 더욱 그렇습니다. CloudWatch 대시보드를 자동으로 생성하고 업데이트하는 솔루션을 사용하면 시간을 절약할 수 있습니다. 이 패턴을 사용하면 태그 변경 이벤트를 기반으로 AWS 리소스에 대한 CloudWatch 대시보드를 생성 및 업데이트하여 Golden Signals 메트릭을 표시하는 완전 자동화된 AWS Cloud Development Kit (AWS CDK) 파이프라인을 배포할 수 있습니다.

사이트 신뢰성 엔지니어링 (SRE) 에서 Golden Signals는 사용자 또는 소비자 관점에서 서비스를 폭넓게 볼 수 있는 포괄적인 지표 세트를 말합니다. 이러한 지표는 지연 시간, 트래픽, 오류 및 포화 상태로 구성됩니다. 자세한 내용은 [사이트 신뢰성 엔지니어링 \(SRE\) 이란?](#) 을 참조하십시오. AWS 웹 사이트에서.

이 패턴에서 제공하는 솔루션은 이벤트 기반입니다. 배포된 후에는 태그 변경 이벤트를 지속적으로 모니터링하고 CloudWatch 대시보드와 경보를 자동으로 업데이트합니다.

사전 조건 및 제한 사항

사전 요구 사항

- 액티브 AWS 계정
- AWS Command Line Interface (AWS CLI), [설치 및 구성](#)
- v2를 위한 [사전 요구 사항](#) AWS CDK
- [부트스트랩된 환경](#) AWS
- [파이썬 버전 3](#)

- [AWS 파이썬용 SDK \(Boto3\), 설치](#)
- [Node.js 버전 18 이상](#)
- 다음을 위해 [설치 및 구성된](#) 노드 패키지 관리자 (npm) AWS CDK
- 및 에 대한 중간 정도 (레벨 200) 의 지식 AWS CDK AWS CodePipeline

제한 사항

이 솔루션은 현재 다음 AWS 서비스에 대해서만 자동화된 대시보드를 생성합니다.

- [Amazon Relational Database Service\(RDS\)](#)
- [AWS Auto Scaling](#)
- [Amazon Simple Notification Service \(Amazon SNS\)](#)
- [Amazon DynamoDB](#)
- [AWS Lambda](#)

아키텍처

대상 기술 스택

- [CloudWatch 대시보드](#)
- [CloudWatch 알람](#)

대상 아키텍처

1. 구성된 애플리케이션 AWS 태그 또는 코드 변경에 대한 태그 변경 이벤트가 발생하면 업데이트된 대시보드를 구축하고 AWS CodePipeline 배포하기 위한 파이프라인이 시작됩니다. CloudWatch
2. AWS CodeBuild Python 스크립트를 실행하여 태그가 구성된 리소스를 찾고 리소스 ID를 CodeBuild 환경의 로컬 파일에 저장합니다.
3. CodeBuild cdk synth를 실행하여 CloudWatch 대시보드와 경보를 배포하는 AWS CloudFormation 템플릿을 생성합니다.
4. CodePipeline AWS CloudFormation 템플릿을 지정된 지역과 지역에 배포합니다. AWS 계정
5. AWS CloudFormation 스택이 성공적으로 배포되면 CloudWatch 대시보드와 경보를 볼 수 있습니다.

자동화 및 규모 조정

이 솔루션은 를 사용하여 자동화되었습니다. AWS CDK [Amazon CloudWatch 리포지토리의 GitHub 골든 시그널 대시보드에서](#) 코드를 찾을 수 있습니다. 추가 규모 조정 및 사용자 지정 대시보드 생성을 위해 여러 태그 키와 값을 구성할 수 있습니다.

도구

Amazon 서비스

- [EventBridgeAmazon](#)은 AWS Lambda 함수, API 대상을 사용하는 HTTP 호출 엔드포인트 또는 기타 이벤트 버스를 비롯한 다양한 소스의 실시간 데이터와 애플리케이션을 연결할 수 있도록 지원하는 서버리스 이벤트 버스 서비스입니다. AWS 계정
- [AWS CodePipeline](#) 소프트웨어 릴리스의 여러 단계를 신속하게 모델링 및 구성하고 소프트웨어 변경 사항을 지속적으로 릴리스하는 데 필요한 단계를 자동화할 수 있도록 지원합니다.
- [AWS CodeBuild](#) 소스 코드를 컴파일하고, 단위 테스트를 실행하고, 배포할 준비가 된 아티팩트를 생성하는 데 도움이 되는 완전 관리형 빌드 서비스입니다.
- [AWS CodeCommit](#) 자체 소스 제어 시스템을 관리할 필요 없이 Git 리포지토리를 비공개로 저장하고 관리할 수 있는 버전 제어 서비스입니다.
- [AWS Command Line Interface \(AWS CLI\)](#) 는 명령줄 셸의 명령을 통해 AWS 서비스와 상호 작용하는 데 도움이 되는 오픈 소스 도구입니다.
- [AWS Identity and Access Management \(IAM\)](#) 을 사용하면 리소스 인증 및 사용 권한을 부여받은 사용자를 제어하여 AWS 리소스에 대한 액세스를 안전하게 관리할 수 있습니다.
- [Amazon Simple Storage Service\(S3\)](#)는 원하는 양의 데이터를 저장, 보호 및 검색하는 데 도움이 되는 클라우드 기반 객체 스토리지 서비스입니다.

모범 사례

보안 모범 사례로, 파이프라인에 연결되는 소스 리포지토리에 암호화 및 인증을 사용할 수 있습니다. 추가 모범 사례는 설명서의 [CodePipeline 모범 사례 및 사용 사례를](#) 참조하십시오. CodePipeline

에픽

샘플 애플리케이션 구성 및 배포

작업	설명	필요한 기술
<p>샘플 애플리케이션을 구성하고 배포합니다.</p>	<ol style="list-style-type: none"> 다음 명령을 사용하여 GitHub 샘플 코드 리포지토리를 복제합니다. <pre data-bbox="630 600 1029 840">git clone https://github.com/aws-samples/golden-signals-dashboards-sample-app</pre> 컴퓨터의 복제된 리포지토리로 이동한 다음 원하는 편집기로 <code>src/project-settings.ts</code> 파일을 엽니다. AWS 리소스 태그와 응용 프로그램 매핑에 따라 <code>projectSettings</code> 상수 값을 변경하십시오. <code>AWS_ACCOUNT</code>, <code>AWS_REGION</code>, <code>GS_DASHBOARD_INSTANCE</code> 환경 변수를 설정합니다. <ul style="list-style-type: none"> 계정의 계정 <code>AWS_ACCOUNT ID</code>로 설정합니다. 샘플 애플리케이션을 배포할 지역으로 설정합니다. 	<p>AWS DevOps</p>

작업	설명	필요한 기술
	<ul style="list-style-type: none"> 개발 환경에 따라 dev testprod, 또는 GS_DASHBOARD_INSTANCE 로 설정합니다. (이 test 패턴에 설명된 테스트 절차에 권장됩니다.) <p>5. AWS 자격 AWS CLI 증명으로 설정하십시오. 자세한 내용은 AWS CLI 설명서의 명령을 사용하여 구성 설정 설정 및 보기를 참조하십시오.</p> <p>6. 다음 명령을 실행하여 Golden Signals 대시보드 샘플 애플리케이션을 배포하십시오.</p> <div style="border: 1px solid #ccc; border-radius: 10px; padding: 5px; margin: 10px auto; width: fit-content;"> <pre>sh deploy.sh</pre> </div>	

작업	설명	필요한 기술
<p>대시보드와 경보를 자동으로 생성합니다.</p>	<p>샘플 애플리케이션을 배포한 후에는 이 솔루션이 지원하는 모든 리소스를 예상 태그 값으로 만들 수 있습니다. 그러면 지정된 대시보드와 경보가 자동으로 생성됩니다.</p> <p>이 솔루션을 테스트하려면 함수를 AWS Lambda 생성하십시오.</p> <ol style="list-style-type: none"> 1. 샘플 애플리케이션을 배포한 AWS 리전 위치에 로그인합니다. AWS Management Console 2. https://console.aws.amazon.com/lambda/에서 Lambda 콘솔을 엽니다. 3. 함수 생성을 선택한 다음 함수 이름을 입력합니다. 4. 고급 설정 창에서 태그 활성화를 선택한 다음 새 태그 추가를 선택합니다. 다음 키와 값을 입력합니다. <ul style="list-style-type: none"> • 키: AutoDashboard • 값: True 5. 함수 생성을 선택합니다. <p>Lambda 함수는 코드 파이프라인을 즉시 시작하여 특정 Lambda 함수에 대한 대시보드와 경보를 자동으로 생성합니다.</p>	<p>AWS DevOps</p>

작업	설명	필요한 기술
	<p>6. 자동화된 대시보드와 경보를 보려면 https://console.aws.amazon.com/cloudwatch/ 에서 콘솔을 여십시오. CloudWatch projectSettings 상수에 지정한 함수 (기본적으로 App1-Lambda) 에 대한 사용자 지정 대시보드 및 경보를 볼 수 있습니다.</p> <p>7. Lambda 함수의 대시보드를 선택하면 이 솔루션의 일부로 생성된 추가 자동 대시보드를 볼 수 있습니다.</p> <p>8. Amazon RDS, Amazon SNS AWS Auto Scaling, DynamoDB 등의 다른 서비스에 대해서도 이 단계를 반복하여 관련 대시보드를 생성합니다. Amazon RDS의 예를 보려면 추가 정보 섹션을 참조하십시오.</p>	

샘플 애플리케이션 제거

작업	설명	필요한 기술
<p>golden-signals-dashboard 구문을 제거합니다.</p>	<p>1. 샘플 애플리케이션에서 만든 모든 AWS CloudFormation 스택을 제거하려면,, GS_DASHBOARD_INSTANCE 환경 변수를 다시 구성해야 합니다. AWS_ACCOUNT AWS_REGION</p>	<p>AWS DevOps</p>

작업	설명	필요한 기술
	<p>destroy.sh 명령에는 다음과 같은 구성이 필요합니다.</p> <ul style="list-style-type: none"> • AWS_ACCOUNT 계정의 계정 ID입니다. AWS • AWS_REGION 샘플 애플리케이션을 배포한 지역입니다. • GS_DASHBOARD_INSTANCE 이전 설정을 기반으로 한 dev testprod, 또는 입니다. <ol style="list-style-type: none"> 2. AWS 자격 AWS CLI 증명으로 설정하세요. 3. 다음 명령을 실행하여 샘플 애플리케이션 및 모든 관련 AWS CloudFormation 스택을 제거합니다. <div style="border: 1px solid #ccc; border-radius: 10px; padding: 5px; margin-top: 10px; text-align: center;"> <pre>sh destroy.sh</pre> </div>	

문제 해결

문제	Solution
<p>Python 명령을 찾을 수 없습니다 (8행 findresources.sh 참조).</p>	<p>설치된 Python 버전을 확인하십시오. Python 버전 3을 설치한 경우 resources.sh 파일의 8번째 python 줄로 python3 바꾸고 sh deploy.sh 명령을 다시 실행하여 솔루션을 배포하십시오.</p>

관련 리소스

- [부트스트래핑 \(설명서\)](#) AWS CDK
- [이름이 지정된 프로필 사용 \(문서\)](#) AWS CLI
- [AWS CDK 워크숍](#)

추가 정보

다음 그림은 이 솔루션의 일부로 생성된 Amazon RDS용 샘플 대시보드를 보여줍니다.

AWS Config 고급 쿼리를 사용하여 생성 날짜를 기준으로 AWS 리소스 검색

작성자: Inna Saman(AWS)

환경: 프로덕션

기술: 운영, 보안, 자격 증명, 규정 준수

AWS 서비스: AWS Config, 아마존 EBS, 아마존 EC2, 아마존 S3, AWS Lambda

요약

이 패턴은 AWS [Config 고급 쿼리 기능](#)을 사용하여 생성 날짜를 기준으로 AWS 리소스를 찾는 방법을 보여줍니다.

AWS Config 고급 쿼리는 SQL의 하위 집합을 사용하여 인벤토리 관리, 운영 인텔리전스, 보안 및 규정 준수를 위한 AWS 리소스의 구성 상태를 쿼리합니다. 이러한 쿼리를 사용하여 단일 AWS 계정과 AWS 리전 또는 여러 계정 및 리전에서 AWS 리소스를 찾을 수 있습니다. resourceCreationTime속성을 사용하는 쿼리를 실행하면 특정 생성 날짜를 기준으로 AWS 리소스 목록을 반환할 수 있습니다. AWS Config 고급 쿼리는 다음과 같은 방법으로 실행할 수 있습니다.

- AWS Config 콘솔의 AWS Config 쿼리 편집기
- AWS Command Line Interface(AWS CLI)

이 패턴의 추가 정보 섹션에 있는 예제 쿼리는 특정 60일 기간 내에 생성된 AWS 리소스 목록을 반환합니다. 쿼리 출력에는 식별된 각 리소스에 대한 다음 정보가 포함됩니다.

- 계정 ID
- 리전
- 리소스 이름
- 리소스 ID
- 리소스 유형
- Tags
- 생성 시간

또한 예제 쿼리는 "WHERE ...IN" 명령문을 사용하여 인벤토리 목록의 범위를 특정 리소스 유형으로 지정하는 방법도 IN" 보여줍니다. 유사한 쿼리를 사용하여 태그와 함께 작동하는 다른 AWS 리소스 유형을 찾을 수 있습니다.

참고: 여러 AWS 계정 및 리전 또는 AWS Organizations 조직 전체에서 리소스를 쿼리하려면 AWS Config 애그리게이터를 사용해야 합니다. 자세한 내용은 AWS Config 개발자 가이드에서 [다중 계정 다중 리전 데이터 집계](#)를 참조하십시오. 글로벌 리소스는 해당 리소스의 홈 리전에만 기록됩니다. 예를 들어, AWS Identity 및 액세스 관리 (IAM) 는 글로벌 리소스이며 us-east-1(북부 버지니아 리전)에 기록됩니다.

사전 조건 및 제한 사항

사전 조건

- 지원되는 모든 리소스 유형을 기록하도록 AWS Config가 활성화된 하나 이상의 활성 AWS 계정([기본 구성](#))
- (다중 계정, 다중 리전 쿼리의 경우) 활성화된 AWS Config 애그리게이터

제한 사항

- AWS Config 고급 쿼리 결과는 페이지로 구분됩니다. 내보내기를 선택하면 AWS Management Console에서 최대 500개의 결과를 내보냅니다. 또한 API를 사용하여 한 번에 최대 100개의 페이지로 구성된 결과를 검색할 수 있습니다.
- AWS Config 고급 쿼리는 자체 구문 제한이 있는 SQL의 하위 집합을 사용합니다. 자세한 내용은 AWS Config 개발자 가이드에서 AWS 리소스의 현재 구성 상태 쿼리의 [제한 사항](#)을 참조하십시오.

도구

도구

- [AWS Config](#)는 사용자의 AWS 계정의 리소스와 구성 방법을 상세하게 볼 수 있도록 합니다. 리소스가 서로 관련되는 방식과 리소스의 구성이 시간이 지남에 따라 변경된 방식을 식별하는 데 도움이 됩니다.
- [AWS Command Line Interface\(AWS CLI\)](#)는 명령줄 셸에서 명령을 사용하여 AWS 서비스와 상호 작용할 수 있는 오픈 소스 도구입니다.

에픽

AWS Config 고급 쿼리 실행

작업	설명	필요한 기술
<p>쿼리하려는 리소스가 AWS Config에서 지원되는지 확인합니다.</p>	<p>AWS Config가 지원하는 AWS 리소스의 전체 목록은 AWS Config 개발자 가이드의 지원되는 리소스 유형을 참조하십시오.</p>	<p>클라우드 관리자</p>
<p>구성 레코더가 생성되어 실행 중인지 확인합니다.</p>	<p>AWS Config 개발자 가이드의 구성 레코더 관리에 나와 있는 지침을 따르십시오.</p> <p>참고: AWS Config는 기본 구성 레코더를 자동으로 생성한 다음 시작합니다.</p>	<p>클라우드 관리자</p>
<p>쿼리를 실행합니다.</p>	<p>AWS Config 개발자 가이드에서 SQL 쿼리 편집기를 사용한 쿼리 (콘솔) 또는 SQL 쿼리 편집기를 사용한 쿼리(AWS CLI)의 지침을 따르십시오.</p> <p>참고: AWS CLI 명령을 실행할 때 오류가 발생하는 경우, 최신 버전의 AWS CLI를 사용하고 있는지 확인합니다.</p> <p>단일 AWS 계정 및 리전 쿼리용</p> <p>쿼리 편집기 페이지의 쿼리 범위 섹션에서 이 계정 및 리전만을 선택했는지 확인합니다.</p> <p>다중 계정 및 다중 리전 쿼리용</p>	<p>클라우드 관리자</p>

작업	설명	필요한 기술
	<p>쿼리 편집기 페이지의 쿼리 범위 섹션에서 AWS Config 애그리게이터를 생성하고 선택했는지 확인합니다. 자세한 내용은 AWS Config 개발자 가이드에서 다중 계정 다중 리전 데이터 집계를 참조하십시오.</p> <p>여러 계정 또는 리전에 대한 쿼리가 작동하지 않는 경우, AWS Config 개발자 가이드의 다중 계정 다중 리전 데이터 집계 문제 해결의 지침을 따르십시오.</p> <p>참고: 리소스 유형에 따라 쿼리 범위를 수정하려면 WHERE resourceType IN (...) 구성을 사용하십시오. 예제 쿼리는 추가 정보 섹션의 AWS Config 고급 쿼리 예제를 참조하십시오.</p>	

추가 정보

AWS Config 고급 쿼리 예제

다음 예제 쿼리는 특정 60일 기간 내에 생성된 AWS 리소스 목록을 반환합니다. 더 많은 AWS Config 고급 쿼리 예제는 AWS Config 개발자 가이드의 [쿼리 예제](#)를 참조하십시오.

```
SELECT
  accountId,
  awsRegion,
  resourceName,
  resourceId,
  resourceType,
  resourceCreationTime,
  tags
WHERE
```

```

resourceType IN (
  'AWS::CloudFormation::Stack',
  'AWS::EC2::VPC',
  'AWS::EC2::Volume',
  'AWS::EC2::Instance',
  'AWS::RDS::DBInstance',
  'AWS::ElasticLoadBalancingV2::LoadBalancer',
  'AWS::ServiceCatalog::CloudFormationProvisionedProduct',
  'AWS::EC2::NetworkInterface',
  'AWS::EC2::Subnet',
  'AWS::EC2::SecurityGroup',
  'AWS::AutoScaling::AutoScalingGroup',
  'AWS::Lambda::Function',
  'AWS::DynamoDB::Table',
  'AWS::S3::Bucket'
)
AND resourceCreationTime BETWEEN '2022-05-23T00:00:00.000Z' AND
'2022-07-23T17:59:51.000Z'
ORDER BY
  accountId ASC,
  resourceType ASC

```

데이터 프라이버시 및 보호

AWS Config는 각 AWS 리전에서 개별적으로 활성화됩니다. 규제 요구 사항을 준수하려면 별도의 리전 애그리게이터를 생성하는 등 특별한 고려 사항을 적용해야 합니다. 자세한 내용은 AWS Config 개발자 가이드의 [AWS Config의 데이터 보호](#)를 참조하십시오.

IAM 권한

[ConfigRoleAWS Config 고급 쿼리를 실행하려면 AWS_](#) AWS 관리형 정책이 최소 권한 집합으로 필요합니다. 자세한 내용은 AWS Config 개발자 가이드의 AWS Config에 할당된 IAM 역할에 대한 권한에서 [구성 세부 정보를 가져오는 IAM 역할 정책](#)을 참조하십시오.

계정 또는 조직의 EBS 스냅샷 세부 정보 보기

작성자: Arun Chandapillai(AWS) 및 Parag Nagwekar(AWS)

환경: 프로덕션

기술: 운영, 저장 및 백업

AWS 서비스: Amazon EBS

요약

이 패턴은 Organizations 내 Amazon Web Services(AWS) 또는 조직 단위(OU)에서 모든 Amazon Elastic Block Store(Amazon EBS) 스냅샷의 온디맨드 보고서를 자동으로 생성하는 방법을 설명합니다.

Amazon EBS는 Amazon Elastic Compute Cloud (Amazon EC2) 용으로 설계된 확장 가능한 고성능 블록 스토리지 서비스입니다. easy-to-use EBS 볼륨은 EC2 인스턴스에 연결할 수 있는 내구성을 갖춘 영구 스토리지를 제공합니다. EBS 볼륨을 데이터의 기본 스토리지로 사용하고 스냅샷을 생성하여 EBS 볼륨을 point-in-time 백업할 수 있습니다. Management Console 또는 Command Line Interface(AWS CLI)를 사용하여 특정 EBS 스냅샷의 세부 정보를 볼 수 있습니다. 이 패턴은 계정 또는 OU의 모든 EBS 스냅샷에 대한 정보를 검색하는 프로그래밍 방식을 제공합니다.

이 패턴에서 제공하는 스크립트를 사용하여 계정 ID, 스냅샷 ID, 볼륨 ID 및 크기, 스냅샷을 촬영한 날짜, 인스턴스 ID, 설명 등 각 스냅샷에 대한 정보가 포함된 CSV(쉼표로 구분된 값) 파일을 생성할 수 있습니다. EBS 스냅샷에 태그가 지정되는 경우 보고서에는 소유자 및 팀 속성도 포함합니다.

사전 조건 및 제한 사항

사전 조건

- 활성 상태의 AWS 계정.
- CLI 버전 2의 [설치](#) 및 [구성](#)
- 적절한 권한이 있는 Identity and Access Management(IAM) 역할(Organizations에서 스크립트를 실행하려는 경우 특정 계정 또는 OU 내 모든 계정에 대한 액세스 권한)

아키텍처

다음 다이어그램은 OU의 여러 AWS 계정 전반적으로 분산된 EBS 스냅샷의 온디맨드 보고서를 생성하는 스크립트 워크플로를 보여줍니다.

도구

서비스

- [명령줄 인터페이스\(CLI\)](#)는 명령줄 셸에서 명령을 사용하여 서비스와 상호 작용할 수 있는 오픈 소스 도구입니다.
- [Amazon Elastic Block Store\(Amazon EBS\)](#)는 EC2 인스턴스에 사용할 수 있는 블록 수준 스토리지 볼륨을 제공합니다.
- [Identity and Access Management \(IAM\)](#)를 사용하여 누가 인증을 받고 권한을 받는지를 제어하여 리소스에 대한 액세스를 안전하게 제어할 수 있습니다.
- [Organizations](#)는 여러 계정을 사용자가 생성하고 중앙에서 관리하는 단일 조직으로 통합할 수 있는 계정 관리 서비스입니다.

code

이 패턴에 사용된 샘플 애플리케이션의 코드는 저장소의 에서 GitHub 사용할 수 있습니다. [aws-eks-snapshots-awsorganizations](#) 샘플 파일을 사용하려면 다음 섹션의 지침을 따르십시오.

에픽

스크립트 다운로드

작업	설명	필요한 기술
Python 스크립트를 다운로드합니다.	GitHub 리포지토리에서 GetSnapshotDetailsAllAccounts OU.py 스크립트를 다운로드합니다.	일반 AWS

AWS 계정에 대한 EBS 스냅샷 세부 정보 가져오기

작업	설명	필요한 기술
Python 스크립트를 실행합니다.	명령을 실행합니다.	일반 AWS

작업	설명	필요한 기술
	<pre data-bbox="592 220 1027 409">python3 getsnapsh otinfo.py --file <output-file>.csv -- region <region-name></pre> <p data-bbox="592 441 1027 724">여기서 <output-file> (은)는 EBS 스냅샷에 대한 정보를 배치하려는 CSV 출력 파일을 의미하고, <region-name> (은)는 스냅샷이 저장되는 AWS 리전입니다. 예:</p> <pre data-bbox="592 756 1027 955">python3 getsnapsh otinfo.py --file snapshots.csv --region us-east-1</pre>	

조직에 대한 EBS 스냅샷 세부 정보 가져오기

작업	설명	필요한 기술
Python 스크립트를 실행합니다.	<p data-bbox="592 1239 1027 1281">명령을 실행합니다.</p> <pre data-bbox="592 1312 1027 1554">python3 getsnapsh otinfo.py --file <output-file>.csv --role <IAM-role> -- region <region-name></pre> <p data-bbox="592 1585 1027 1858">여기서 <output-file> (은)는 EBS 스냅샷에 대한 정보를 배치하려는 CSV 출력 파일을 의미하고, <IAM-role> (은)는 Organizations에 액세스하는 권한을 부여하는 역할이고,</p>	일반 AWS

작업	설명	필요한 기술
	<p><region-name> (은)는 스냅샷이 저장되는 AWS 리전입니다. 예:</p> <pre data-bbox="597 380 1024 617">python3 getsnapsh otinfo.py --file snapshots.csv --role <IAM role> --region us- west-2</pre>	

관련 리소스

- [Amazon EBS 설명서](#)
- [Amazon EBS 작업](#)
- [Amazon EBS API 참조](#)
- [Amazon EBS 성능 개선](#)
- [Amazon EBS 리소스](#)
- [EBS 스냅샷 가격 책정](#)

추가 정보

EBS 스냅샷 유형

Amazon EBS는 소유권과 액세스를 기준으로 세 가지 유형의 스냅샷을 제공합니다.

- 사용자에 의한 소유 – 기본적으로 사용자만 자기 소유의 스냅샷에서 볼륨을 생성할 수 있습니다.
- 퍼블릭 스냅샷 — 스냅샷을 다른 모든 계정과 공개적으로 공유할 수 있습니다. 퍼블릭 스냅샷을 생성하려면, 퍼블릭 스냅샷 권한을 수정하여 그것을 지정한 계정과 공유합니다. 그런다음, 권한을 부여 받은 사용자는 공유하는 스냅샷을 사용하여 자신의 EBS 볼륨을 생성할 수 있으며, 원본 스냅샷은 영향을 받지 않습니다. 암호화되지 않은 스냅샷을 모든 사용자에게 공개되어 사용할 수 있게 할 수 있습니다. 그러나, 암호화된 스냅샷은 보안 이유로 공개적으로 사용 가능하게 만들 수 없습니다. 퍼블릭 스냅샷은 개인 데이터와 민감한 데이터가 노출될 수 있기 때문에 심각한 보안 위험을 초래합니다. EBS 스냅샷을 모든 AWS 계정과 공유하지 않는 것이 좋습니다. 스냅샷 공유에 대한 자세한 내용은 [설명서](#)를 참조하십시오.

- 프라이빗 스냅샷 — 지정한 개별 계정과 비공개로 스냅샷을 공유할 수 있습니다. 스냅샷을 특정 계정과 비공개로 공유하려면 설명서의 [지침](#)을 따르고 권한 설정에 대해 비공개를 선택합니다. 권한을 부여받은 사용자는 공유 스냅샷을 사용하여 자체 EBS 볼륨을 생성할 수 있습니다. 원본 스냅샷은 영향을 받지 않습니다.

개요 및 절차

다음 표에는 사용하지 않는 스냅샷을 찾아 삭제하여 EBS 볼륨 비용을 절감하고, 자주 검색하거나 빠르게 검색할 필요가 없는 거의 액세스되지 않는 스냅샷을 아카이브하는 방법을 포함하여 EBS 스냅샷에 관한 자세한 정보에 대한 링크가 나와 있습니다.

자세한 내용은

참조하세요

스냅샷, 기능 및 제한

[Create Amazon EBS snapshots](#)

스냅샷을 생성하는 방법

콘솔: [스냅샷 생성](#)

CLI: [create-snapshot 명령](#)

예:

```
aws ec2 create-snapshot --volume-id
vol-1234567890abcdef0 --description
" volume snapshot"
```

스냅샷 삭제(일반 정보)

[Amazon EBS 스냅샷 삭제](#)

스냅샷을 삭제하는 방법

콘솔: [스냅샷 삭제](#)

CLI: [delete-snapshot 명령](#)

예:

```
aws ec2 delete-snapshot --snapshot-id
snap-1234567890abcdef0
```

스냅샷 아카이브(일반 정보)

[Amazon EBS 스냅샷 아카이브하기](#)

[Amazon EBS 스냅샷 아카이브\(블로그 포스트\)](#)

스냅샷을 아카이브하는 방법

콘솔: [스냅샷 아카이브](#)

[AWS CLI: 명령 modify-snapshot-tier](#)

아카이브된 스냅샷을 검색하는 방법

Console: [아카이브된 스냅샷 복원](#)

[AWS CLI: 명령 restore-snapshot-tier](#)

Snapshot 요금 책정

[Amazon EBS 요금 책정](#)

FAQ

최소 아카이브 기간이란 무엇입니까?

최소 아카이브 기간은 90일입니다.

아카이브된 스냅샷을 복원하는 데 시간이 얼마나 걸립니까?

스냅샷의 크기에 따라 아카이브 계층에서 표준 계층으로 아카이빙된 스냅샷을 복원하는 데 최대 72시간이 걸릴 수 있습니다.

아카이브된 스냅샷은 전체 스냅샷입니까?

아카이브된 스냅샷은 항상 전체 스냅샷입니다.

사용자가 아카이빙할 수 있는 스냅샷은 무엇입니까?

계정에서 소유한 스냅샷만 아카이빙할 수 있습니다.

등록된 Amazon Machine Image(AMI)의 루트 디바이스 볼륨의 스냅샷을 아카이빙할 수 있습니까?

아니오, 등록된 AMI의 루트 디바이스 볼륨의 스냅샷을 아카이빙할 수 없습니다.

스냅샷 공유 시 보안 고려 사항은 무엇입니까?

스냅샷을 공유하면 다른 사람들이 해당 스냅샷의 모든 데이터에 액세스할 수 있게 됩니다. 신뢰할 수 있는 사용자하고만 데이터를 공유하세요.

스냅샷을 다른 리전과 공유하려면 어떻게 해야 합니까?

스냅샷은 생성된 리전으로 제한됩니다. 스냅샷을 다른 리전과 공유하려면 스냅샷을 해당 리전에 복사한 다음 복사본을 공유합니다.

암호화된 스냅샷을 공유할 수 있습니까?

기본 관리형 키로 암호화된 스냅샷은 공유할 수 없습니다. 고객 관리형 키로 암호화된 스냅샷만을 공유할 수 있습니다. 암호화된 스냅샷을 공유할 때는 해당 스냅샷을 암호화하는 데 사용되었던 고객 관리형 키도 공유해야 합니다.

암호화되지 않은 스냅샷은 어떻습니까?

암호화되지 않은 스냅샷은 공개적으로 공유할 수 있습니다.

패턴 더 보기

- [AMS 계정의 S3 버킷에 대한 EC2 인스턴스 쓰기 액세스 허용](#)
- [AWS 리소스 평가 자동화](#)
- [Amazon Inspector와 Security Hub를 사용하여 교차 계정 워크로드에 대한 보안 스캔 자동화](#)
- [AWS Config에서 사용자 지정 수정 규칙을 CloudTrail 사용하여 AWS를 자동으로 다시 활성화합니다.](#)
- [SageMaker Amazon과 Azure를 사용하여 MLOps 워크플로를 구축하세요. DevOps](#)
- [Amazon CloudWatch 오피저빌리티 액세스 관리자를 사용하여 모니터링을 중앙 집중화합니다.](#)
- [AWS IoT 환경에서 보안 이벤트에 대한 로깅 및 모니터링을 구성합니다.](#)
- [Session Manager를 사용하여 Amazon EC2 인스턴스에 연결](#)
- [Amazon CloudWatch 이상 탐지를 사용하여 사용자 지정 지표에 대한 경보 생성](#)
- [AWS CloudFormation 템플릿을 사용하여 Amazon을 GuardDuty 조건부로 활성화합니다.](#)
- [AWS CDK를 사용하여 여러 AWS 지역, 계정 및 DevOps OU에서 Amazon Guru를 활성화하여 운영 성능을 개선하십시오.](#)
- [EC2 Windows 인스턴스를 수집하여 AWS Managed Services 계정으로 마이그레이션합니다.](#)
- [다음을 사용하여 Amazon EKS 작업자 노드에 SSM CloudWatch 에이전트 및 에이전트를 설치합니다. preBootstrapCommands](#)
- [Stonebranch 유니버설 컨트롤러를 AWS Mainframe Modernization과 통합](#)
- [Step Functions와 Lambda 프록시 함수를 사용하여 여러 AWS 계정에서 CodeBuild 프로젝트를 시작합니다.](#)
- [예정된 AWS KMS 키 삭제 모니터링 및 문제 해결](#)
- [여러 AWS 계정에 공유된 Amazon Machine Image의 사용을 모니터링](#)
- [AWS Step Functions에서 AWS Systems Manager Automation 작업을 동기적으로 실행](#)
- [AWS Fargate를 사용하여 이벤트 기반 및 예약된 워크로드를 대규모로 실행](#)
- [다중 지역, 다중 계정 조직에서 AWS CloudFormation 드리프트 감지 설정](#)
- [AWS 기반 IBM Db2에서 SAP를 위한 재해 복구 설정](#)
- [AWS Organizations를 사용하여 Transit Gateway Attachment에 자동으로 태그 지정](#)
- [Splunk를 사용하여 AWS 네트워크 방화벽 로그 및 지표 보기](#)

SaaS

주제

- [단일 컨트롤 플레인에서 여러 SaaS 제품의 테넌트 관리](#)
- [더 많은 패턴](#)

단일 컨트롤 플레인에서 여러 SaaS 제품의 테넌트 관리

작성자: Ramanna Avanch(AWS), Jenifer Pascal(AWS), Kishan Kavala(AWS), Anusha Mandava(AWS)

환경: PoC 또는 파일럿	기술: SaaS	AWS 서비스: Amazon API Gateway, Amazon Cognito, AWS Lambda, AWS Step Functions, Amazon DynamoDB
----------------	----------	--

요약

이 패턴은 AWS 클라우드의 단일 컨트롤 플레인에서 여러 서비스형 소프트웨어(SaaS) 제품의 테넌트 수명 주기를 관리하는 방법을 보여줍니다. 제공된 참조 아키텍처는 조직이 개별 SaaS 제품에서 중복 및 공유된 기능의 구현을 줄이고 규모에 맞는 거버넌스 효율성을 제공하는 데 도움이 될 수 있습니다.

대기업은 대체로 다양한 사업부에서 여러 SaaS 제품을 보유하고 있습니다. 이러한 제품은 외부 테넌트가 다양한 구독 수준에서 사용할 수 있도록 프로비저닝해야 하는 경우가 많습니다. 공통 테넌트 솔루션이 없으면 IT 관리자는 핵심 제품 기능 개발에 집중하는 대신 여러 SaaS API에서 차별화되지 않은 기능을 관리하는 데 시간을 할애해야 합니다.

이 패턴으로 제공되는 공통 테넌트 솔루션은 다음을 포함하여 조직의 여러 공유 SaaS 제품 기능을 중앙 집중식으로 관리하는 데 도움이 될 수 있습니다.

- 보안
- 테넌트 프로비저닝
- 테넌트 데이터 스토리지
- 테넌트 커뮤니케이션
- 제품 관리
- 지표 로깅 및 모니터링

사전 조건 및 제한 사항

사전 조건

- 활성 상태의 AWS 계정.
- Amazon Cognito 또는 타사 ID 제공업체(IdP)에 대한 지식
- Amazon API Gateway에 대한 지식
- AWS Lambda에 대한 지식
- Amazon DynamoDB에 대한 지식
- AWS Identity and Access Management(IAM)에 대한 지식
- AWS Step Functions에 대한 지식
- CloudTrail AWS와 아마존에 대한 지식 CloudWatch
- Python 라이브러리 및 코드에 대한 지식
- 다양한 유형의 사용자(조직, 테넌트, 관리자, 애플리케이션 사용자), 구독 모델, 테넌트 격리 모델을 비롯한 SaaS API에 대한 지식
- 조직의 다중 제품 SaaS 요구 사항 및 멀티테넌트 구독에 대한 지식

제한 사항

- 공통 테넌트 솔루션과 개별 SaaS 제품 간의 통합은 이 패턴에서 다루지 않습니다.
- 이 패턴은 Amazon Cognito 서비스를 단일 AWS 리전에만 배포합니다.

아키텍처

대상 기술 스택

- Amazon API Gateway
- Amazon Cognito
- AWS CloudTrail
- 아마존 CloudWatch
- Amazon DynamoDB
- IAM
- AWS Lambda
- Amazon Simple Storage Service (S3)
- Amazon Simple Notification Service(SNS)
- AWS Step Functions

대상 아키텍처

다음 다이어그램은 AWS 클라우드의 단일 컨트롤 플레인에서 여러 SaaS 제품의 테넌트 수명 주기를 관리하는 워크플로 예제를 보여줍니다.

이 다이어그램은 다음 워크플로를 보여줍니다.

1. AWS 사용자는 API Gateway 엔드포인트를 직접적으로 호출하여 테넌트 프로비저닝, 제품 프로비저닝 또는 관리 관련 작업을 시작합니다.
2. 이 사용자는 Amazon Cognito 사용자 풀 또는 다른 IdP에서 검색된 액세스 토큰으로 인증됩니다.
3. 개별 프로비저닝 또는 관리 작업은 API Gateway API 엔드포인트와 통합된 Lambda 함수에 의해 실행됩니다.
4. 일반 테넌트 솔루션용(테넌트, 제품 및 사용자용) 관리 API는 필요한 입력 파라미터, 헤더 및 토큰을 모두 수집합니다. 그런 다음 관리 API가 관련 Lambda 함수를 간접적으로 호출합니다.
5. 관리 API와 Lambda 함수 모두에 대한 IAM 권한은 IAM 서비스에 의해 검증됩니다.
6. Lambda 함수는 DynamoDB 및 Amazon S3의 카탈로그(테넌트, 제품 및 사용자용)를 저장하고 데이터를 검색합니다.
7. 권한이 검증되면 AWS Step Functions 워크플로가 간접적으로 호출되어 특정 업무를 수행합니다. 다이어그램의 예제는 테넌트 프로비저닝 워크플로를 보여줍니다.
8. 개별 AWS Step Functions 워크플로 업무는 미리 정해진 워크플로(상태 머신)에서 실행됩니다.
9. 각 워크플로 업무와 관련된 Lambda 함수를 실행하는 데 필요한 모든 필수 데이터는 DynamoDB 또는 Amazon S3에서 검색됩니다. 다른 AWS 리소스는 AWS 템플릿을 사용하여 프로비저닝해야 할 수 있습니다. CloudFormation
- 10.필요한 경우 워크플로는 특정 SaaS 제품에 대한 추가 AWS 리소스를 해당 제품의 AWS 계정에 프로비저닝하라는 요청을 보냅니다.
- 11.이 요청이 성공하거나 실패하면 워크플로는 상태 업데이트를 Amazon SNS 주제에 메시지로 게시합니다.
- 12.Amazon SNS는 Step Functions 워크플로의 Amazon SNS 주제를 구독하고 있습니다.
- 13.그러면 Amazon SNS에서는 AWS 사용자에게 워크플로 상태 업데이트를 다시 보냅니다.
- 14.API 호출의 감사 추적을 비롯한 각 AWS 서비스의 작업 로그가 로 전송됩니다 CloudWatch. 각 사용 사례에 CloudWatch 대해 특정 규칙 및 경보를 구성할 수 있습니다.
- 15.로그는 감사 목적으로 Amazon S3 버킷에 보관됩니다.

자동화 및 규모 조정

이 패턴은 CloudFormation 템플릿을 사용하여 공통 테넌트 솔루션의 배포를 자동화하는 데 도움이 됩니다. 또한 템플릿을 사용하면 관련 리소스를 빠르게 확장하거나 축소할 수 있습니다.

자세한 내용은 AWS CloudFormation 사용 설명서의 [AWS CloudFormation 템플릿](#) 사용을 참조하십시오.

도구

도구

- [Amazon API Gateway](#)를 사용하면 어떤 규모에서든 REST, HTTP 및 WebSocket API를 생성, 게시, 유지 관리, 모니터링 및 보호할 수 있습니다.
- [Amazon Cognito](#)는 웹 및 모바일 앱에 대한 인증, 권한 부여 및 사용자 관리를 제공합니다.
- [AWS는 AWS](#) 계정의 거버넌스, 규정 준수 및 운영 위험을 감사할 수 있도록 CloudTrail 지원합니다.
- [Amazon은](#) AWS 리소스의 지표와 AWS에서 실행하는 애플리케이션을 실시간으로 모니터링할 수 있도록 CloudWatch 도와줍니다.
- [Amazon DynamoDB](#)는 빠르고 예측 가능하고 확장 가능한 성능을 제공하는 완전 관리형 NoSQL 데이터베이스 서비스입니다.
- [AWS Identity and Access Management\(IAM\)](#)를 사용하면 사용자에 대해 인증 및 권한 부여를 제어함으로써 AWS 리소스에 대한 액세스를 안전하게 관리할 수 있습니다.
- [AWS Lambda](#)는 서버를 프로비저닝하거나 관리할 필요 없이 코드를 실행하는 데 도움이 되는 컴퓨팅 서비스입니다. 필요할 때만 코드를 실행하며 자동으로 확장이 가능하므로 사용한 컴퓨팅 시간만큼만 비용을 지불합니다.
- [Amazon Simple Storage Service\(Amazon S3\)](#)는 원하는 양의 데이터를 저장, 보호 및 검색하는 데 도움이 되는 클라우드 기반 객체 스토리지 서비스입니다.
- [Amazon Simple Notification Service\(Amazon SNS\)](#)를 사용하면 웹 서버 및 이메일 주소를 포함하여 게시자와 클라이언트 간의 메시지 교환을 조정하고 관리할 수 있습니다.
- [AWS Step Functions](#)는 AWS Lambda 함수와 기타 AWS 서비스를 결합할 수 있는 서버리스 오케스트레이션 서비스로, 비즈니스 크리티컬 애플리케이션을 구축합니다.

모범 사례

이 패턴의 솔루션은 단일 컨트롤 플레인을 사용하여 여러 테넌트의 온보딩을 관리하고, 여러 SaaS 제품에 대한 액세스를 프로비저닝합니다. 컨트롤 플레인은 관리 사용자가 다음 네 가지 기능별 영역을 관리하는 데 도움이 됩니다.

- 보안 영역
- 워크플로 영역
- 통신 영역
- 로깅 및 모니터링 영역

에픽

보안 플레인 구성

작업	설명	필요한 기술
멀티테넌트 SaaS 플랫폼에 대한 요구 사항을 설정합니다.	다음에 대한 세부 요구 사항을 설정하세요. <ul style="list-style-type: none"> • 테넌트 • 사용자 • 역할 • SaaS 제품 • 구독 • 프로필 교환 	클라우드 아키텍트, AWS 시스템 관리자
Amazon Cognito 서비스를 설정합니다.	Amazon Cognito 개발자 안내서의 Amazon Cognito 시작하기 에 나와 있는 지침을 따르세요.	클라우드 아키텍트
필수 IAM 정책을 구성합니다.	사용 사례에 필요한 IAM 정책을 생성합니다. 그런 다음 정책을 Amazon Cognito의 IAM 역할에 매핑합니다.	클라우드 관리자, 클라우드 아키텍트, AWS IAM 보안

작업	설명	필요한 기술
	<p>자세한 내용은 Amazon Cognito 개발자 안내서의 정책을 사용한 액세스 관리와 역할 기반 액세스 제어를 참조하세요.</p>	
<p>필요한 API 권한을 구성합니다.</p>	<p>IAM 역할 및 정책, Lambda 권한 부여자를 사용하여 API Gateway 액세스 권한을 설정합니다.</p> <p>지침은 Amazon API Gateway 개발자 안내서의 다음 섹션을 참조하세요.</p> <ul style="list-style-type: none"> • IAM 권한을 사용하여 API에 대한 액세스 제어 • API Gateway Lambda 권한 부여자 사용 	<p>클라우드 관리자, 클라우드 아키텍트</p>

데이터 영역 구성

작업	설명	필요한 기술
<p>필요한 데이터 카탈로그를 만드세요.</p>	<ol style="list-style-type: none"> 1. 사용자 카탈로그의 데이터를 저장할 DynamoDB 테이블을 생성합니다. 사용자 속성 및 역할을 포함해야 합니다. 또한 카탈로그 테이블에서 데이터 모델링을 수행하여 각 사용자 및 역할에 대한 필수 및 선택적 속성을 유지해야 합니다. 2. 제품 카탈로그의 데이터를 저장할 DynamoDB 테이블 	<p>DBA</p>

작업	설명	필요한 기술
	<p>을 생성합니다. SaaS 제품의 특정 사용 사례를 모델링해야 합니다.</p> <p>3. 테넌트 카탈로그의 데이터를 저장할 DynamoDB 테이블을 생성합니다. 다중 SaaS 구독을 위한 테넌트, 제품, 라이선스 및 태그에 대한 구독 모델을 설정해야 합니다.</p> <p>자세한 내용을 알아보려면 Amazon DynamoDB 개발자 안내서의 DynamoDB란 무엇인가요?를 참조하세요.</p>	

컨트롤 플레인 설정

작업	설명	필요한 기술
<p>Lambda 함수 및 API Gateway API를 생성하여 필요한 컨트롤 플레인 작업을 실행합니다.</p>	<p>별도의 Lambda 함수와 API Gateway API를 생성하여 다음을 추가, 삭제 및 관리합니다.</p> <ul style="list-style-type: none"> • 사용자 • 테넌트 • 제품 <p>자세한 내용은 AWS Lambda 개발자 안내서의 Amazon RDS와 함께 AWS Lambda 사용을 참조하세요.</p>	<p>앱 개발자</p>

워크플로 영역 구성

작업	설명	필요한 기술
<p>AWS Step Functions 워크플로에서 실행해야 할 작업을 파악합니다.</p>	<p>다음에 대한 AWS Step Functions 워크플로 요구 사항을 상세하게 파악하고 문서화하세요.</p> <ul style="list-style-type: none"> • 사용자 • 테넌트 • 제품 <p>중요: 주요 이해 관계자가 요구 사항을 승인했는지 확인하세요.</p>	<p>앱 소유자</p>
<p>필요한 AWS Step Functions 워크플로를 생성합니다.</p>	<ol style="list-style-type: none"> 1. AWS Step Functions에서 사용자, 테넌트 및 제품에 필요한 워크플로를 생성합니다. 자세한 내용은 AWS Step Functions 개발자 안내서를 참조하세요. 2. 재시도 및 오류 처리 메커니즘을 식별하세요. 자세한 내용은 AWS 블로그의 오류 처리, 재시도 및 Step Function State Machine에 경고 추가를 참조하세요. 3. Lambda 함수를 사용하여 워크플로 단계를 구현하세요. 지침은 AWS Step Functions 개발자 가이드의 Lambda를 사용하는 Step Functions 상태 시스템 생성을 참조하세요. 	<p>앱 개발자, 빌드 리더</p>

작업	설명	필요한 기술
	<p>4. 필요에 따라 원하는 외부 서비스를 AWS Step Functions 와 통합합니다.</p> <p>5. DynamoDB 테이블에서 각 워크플로의 상태를 유지하고 Amazon SNS를 사용하여 각 워크플로의 상태를 전달합니다.</p>	

통신 영역 구성

작업	설명	필요한 기술
Amazon SNS 주제를 생성합니다.	<p>Amazon SNS 주제를 생성하여 다음에 대한 알림을 받아봅니다.</p> <ul style="list-style-type: none"> • 워크플로 상태 • Errors • 재시도 <p>자세한 정보는 Amazon SNS 개발자 안내서의 SNS 주제 생성을 참조하세요.</p>	앱 소유자, 클라우드 아키텍트
각 Amazon SNS 주제에 엔드포인트 구독을 선택합니다.	<p>Amazon SNS 주제에 게시된 메시지를 수신하려면 엔드포인트에서 해당 주제를 구독해야 합니다.</p> <p>자세한 정보는 Amazon SNS 개발자 안내서에서 Amazon SNS 주제 구독을 참조하세요.</p>	앱 개발자, 클라우드 아키텍트

로깅 및 모니터링 영역 구성

작업	설명	필요한 기술
<p>공통 테넌트 솔루션의 각 구성 요소에 대한 로깅을 활성화합니다.</p>	<p>생성한 공통 테넌트 솔루션의 각 리소스에 대해 구성 요소 수준에서 로깅을 활성화합니다.</p> <p>지침은 다음을 참조하세요.</p> <ul style="list-style-type: none"> • API Gateway REST WebSocket API 또는 API의 문제를 해결하기 위해 CloudWatch 로그를 켜려면 어떻게 해야 하나요? (AWS 지식 센터) • CloudWatch 로그를 사용한 로깅 (AWS Step Functions 개발자 가이드) • Python에서 AWS Lambda 함수 로깅(AWS Lambda 개발자 안내서) • Amazon Cognito에서 로깅 및 모니터링(Amazon Cognito 개발자 가이드) • Amazon을 통한 모니터링 CloudWatch (Amazon DynamoDB 개발자 가이드) <p>참고: IAM 정책을 사용하여 각 리소스의 로그를 중앙 집중식 로깅 계정으로 통합할 수 있습니다. 자세한 내용은 중앙 집중식 로깅 및 다중 계정 보안 가드레일을 참조하세요.</p>	<p>AWS 시스템 관리자, 클라우드 관리자, DevOps 엔지니어</p>

공통 테넌트 솔루션 프로비저닝 및 배포

작업	설명	필요한 기술
템플릿 생성 CloudFormation .	<p>CloudFormation 템플릿을 사용하여 전체 공통 테넌트 솔루션과 모든 구성 요소의 배포 및 유지 관리를 자동화합니다.</p> <p>자세한 내용은 AWS CloudFormation 사용 설명서를 참조하십시오.</p>	앱 개발자, DevOps 엔지니어, CloudFormation 개발자

관련 리소스

- [Amazon Cognito 사용자 풀을 권한 부여자로 사용하여 REST API에 대한 액세스 제어](#)(API Gateway 개발자 가이드)
- [API Gateway Lambda 권한 부여자 사용](#)(Amazon API Gateway 개발자 가이드)
- [Amazon Cognito 사용자 풀](#)(Amazon Cognito 개발자 안내서)
- [크로스 어카운트 리전 CloudWatch 콘솔](#) (Amazon CloudWatch 사용 설명서)

더 많은 패턴

- [를 사용하여 마이그레이션 전략 식별 및 계획을 자동화합니다. AppScore](#)
- [AWS를 사용하여 AppStream 2.0 리소스 생성을 자동화합니다. CloudFormation](#)
- [Amazon Service에서 멀티테넌트 서버리스 아키텍처 구축 OpenSearch](#)
- [AWS Lambda 토큰 벤딩 머신을 사용하여 Amazon S3에 대한 SaaS 테넌트 격리를 구현합니다.](#)
- [Stonebranch 유니버설 컨트롤러를 AWS Mainframe Modernization과 통합](#)
- [C# 및 AWS CDK를 사용한 사일로 모델을 위한 SaaS 아키텍처의 테넌트 온보딩](#)

보안, 자격 증명, 규정 준수

주제

- [Amazon Cognito 자격 증명 풀을 사용하여 ASP.NET Core 앱에서 AWS 서비스에 액세스](#)
- [AWS Directory Service를 사용하여 Amazon EC2에서 Microsoft SQL Server 인증하기](#)
- [인시던트 응답 및 포렌식 자동화](#)
- [AWS Security Hub 표준 조사 결과에 대한 문제 해결 자동화](#)
- [Amazon Inspector와 Security Hub를 사용하여 교차 계정 워크로드에 대한 보안 스캔 자동화](#)
- [AWS Config에서 사용자 지정 수정 규칙을 CloudTrail 사용하여 AWS를 자동으로 다시 활성화합니다.](#)
- [암호화되지 않은 Amazon RDS DB 인스턴스 및 클러스터를 자동으로 수정하기](#)
- [AWS Organizations 및 AWS Secrets Manager를 사용하여 대규모 IAM 사용자 액세스 키 자동 교체](#)
- [IAM 액세스 분석기 및 AWS 매크로를 사용하여 CodePipeline AWS 계정의 IAM 정책 및 역할을 자동으로 검증하고 배포합니다. CloudFormation](#)
- [AWS Security Hub를 Jira 소프트웨어와 양방향으로 통합하기](#)
- [EC2 Image Builder와 Terraform을 사용하여 강화된 컨테이너 이미지를 위한 파이프라인을 구축하십시오.](#)
- [Terraform을 사용하여 AWS Organizations의 IAM 액세스 키 관리를 중앙 집중화하십시오.](#)
- [중앙 집중식 로깅 및 다중 계정 보안 가드레일](#)
- [Amazon CloudFront 배포판에서 액세스 로깅, HTTPS 및 TLS 버전을 확인하십시오.](#)
- [IPv4 및 IPv6용 보안 그룹 수신 규칙에서 단일 호스트 네트워크 항목 확인](#)
- [엔터프라이즈 애플리케이션을 위한 Amazon Cognito 인증 흐름 선택](#)
- [AWS 가드 정책을 사용하여 AWS Config 사용자 지정 규칙을 생성합니다. CloudFormation](#)
- [여러 AWS 계정의 Prowler 보안 조사 결과에 대한 통합 보고서 생성](#)
- [AWS Config 및 AWS Systems Manager로 사용하지 않는 Amazon Elastic Block Store\(Amazon EBS\) 볼륨 삭제](#)
- [AWS CDK 및 AWS를 사용하여 AWS 컨트롤 타워 컨트롤을 배포하고 관리합니다. CloudFormation](#)
- [Terraform을 사용하여 AWS Control Tower 컨트롤을 배포하고 관리합니다.](#)
- [여러 코드 결과물에서 보안 문제를 동시에 탐지하는 파이프라인을 배포하십시오.](#)
- [Terraform을 사용하여 AWS WAF의 보안 자동화 솔루션 배포](#)
- [Step Functions를 사용하여 IAM Access Analyzer로 IAM 정책을 동적으로 생성하기](#)

- [AWS Amplify에서 호스팅하는 웹 애플리케이션에 대해 AWS WAF 활성화하기](#)
- [AWS CloudFormation 템플릿을 사용하여 Amazon을 GuardDuty 조건부로 활성화합니다.](#)
- [Amazon RDS for SQL Server에서 투명한 데이터 암호화 활성화하기](#)
- [AWS CloudFormation 스택이 승인된 S3 버킷에서 시작되었는지 확인](#)
- [AWS 로드 밸런서가 보안 리스너 프로토콜\(HTTPS, SSL/TLS\)을 사용하는지 확인](#)
- [시작 시 Amazon EMR 저장 데이터에 대한 암호화가 활성화되었는지 확인](#)
- [IAM 프로파일이 EC2 인스턴스와 연결되었는지 확인](#)
- [Amazon Redshift 클러스터를 생성할 때 암호화되었는지 확인합니다.](#)
- [를 사용하여 AWS IAM ID 센터 자격 증명 및 할당에 대한 보고서를 내보냅니다. PowerShell](#)
- [예정된 AWS KMS 키 삭제 모니터링 및 문제 해결](#)
- [ABAC\(속성 기반 액세스 제어\)를 사용하여 퍼블릭 서브넷 보호](#)
- [Security Hub를 사용하여 AWS Organizations의 퍼블릭 S3 버킷 식별](#)
- [Okta를 AWS IAM Identity Center와 통합하여 사용자, 역할 및 다중 계정 액세스 관리](#)
- [AWS를 사용하여 AWS IAM ID 센터 권한 세트를 코드로 관리합니다. CodePipeline](#)
- [AWS Secrets Manager를 사용한 보안 인증 정보 관리](#)
- [시작 시 전송 중 암호화가 있는지 Amazon EMR 클러스터 모니터링](#)
- [Amazon ElastiCache 클러스터의 미사용 암호화 모니터링](#)
- [AWS Config를 사용하여 EC2 인스턴스 키 페어를 모니터링](#)
- [보안 그룹의 ElastiCache 클러스터 모니터링](#)
- [IAM 루트 사용자 활동 모니터링](#)
- [IAM 사용자 생성 시 알림 전송](#)
- [git-secrets를 사용하여 Git 리포지토리에서 민감한 정보 및 보안 문제 검사하기](#)
- [AWS Network Firewall에서 Slack 채널로 알림 전송](#)
- [AWS 프라이빗 CA와 AWS RAM을 사용하여 프라이빗 인증서 관리를 간소화합니다.](#)
- [다중 계정 환경에서 모든 Security Hub 멤버 계정의 보안 표준 제어를 끕니다.](#)
- [를 사용하여 AWS IAM 자격 증명 센터의 AWS CLI 자격 증명을 업데이트합니다. PowerShell](#)
- [AWS Config를 사용하여 Amazon Redshift 보안 구성 모니터링](#)
- [Network Firewall을 사용하여 아웃바운드 트래픽의 Server Name Indication\(SNI\)에서 DNS 도메인 이름 캡처](#)
- [Terraform을 사용하여 조직에서 GuardDuty Amazon을 자동으로 활성화하십시오.](#)
- [새 Amazon Redshift 클러스터에 필수 SSL 엔드포인트가 있는지 확인](#)

- [새 Amazon Redshift 클러스터가 VPC에서 시작되는지 확인](#)
- [패턴 더 보기](#)

Amazon Cognito 자격 증명 풀을 사용하여 ASP.NET Core 앱에서 AWS 서비스에 액세스

제작: Bibhuti Sahu(AWS)와 Marcelo Barbosa(AWS)

환경: PoC 또는 파일럿

기술: 보안, ID, 규정 준수, 웹
및 모바일 앱

AWS 서비스: Amazon Cognito

요약

이 패턴은 Amazon Cognito 사용자 풀과 자격 증명 풀을 구성한 다음 인증에 성공한 후 ASP.NET Core 앱이 AWS 리소스에 액세스할 수 있도록 하는 방법을 설명합니다.

Amazon Cognito는 웹 및 모바일 애플리케이션에 대한 인증, 권한 부여 및 사용자 관리를 제공합니다. Amazon Cognito의 두 가지 주요 구성 요소는 사용자 풀과 자격 증명 풀입니다.

사용자 풀은 Amazon Cognito의 사용자 디렉터리입니다. 사용자 풀이 있으면 사용자는 Amazon Cognito를 통해 웹 또는 모바일 앱에 로그인할 수 있습니다. 또한 사용자는 Google, Facebook, Amazon 또는 Apple과 같은 소셜 자격 증명 공급자를 통해서 그리고 SAML 자격 증명 공급자를 통해 로그인할 수 있습니다.

Amazon Cognito 자격 증명 풀(페더레이션 자격 증명)을 사용하여 사용자의 고유한 자격 증명을 만들고 자격 증명 공급자와 페더레이션할 수 있습니다. 자격 증명 풀로, 권한이 제한된 임시 AWS 보안 인증 정보를 얻어 다른 AWS 서비스에 액세스할 수 있습니다. 새 Amazon Cognito 자격 증명 풀을 사용하기 전에 하나 이상의 AWS Identity and Access Management(IAM) 역할을 할당하여 애플리케이션 사용자가 AWS 리소스에 액세스할 수 있는 수준을 결정해야 합니다. 자격 증명 풀에서는 인증 및 미인증의 두 가지 자격 증명 유형을 정의합니다. IAM에서 각 ID 유형에 고유한 역할을 할당할 수 있습니다. 인증 자격 증명은 퍼블릭 로그인 공급자(Amazon Cognito 사용자 풀, Facebook, Google, SAML 또는 OpenID Connect 공급자) 또는 개발자 공급자(자체 백엔드 인증 프로세스)에 속하지만, 인증되지 않은 ID는 일반적으로 게스트 사용자에게 속합니다. Amazon Cognito가 사용자 요청을 받으면 서비스는 요청의 인증 여부와 해당 인증 유형과 연결된 역할을 확인한 다음, 해당 역할에 연결된 정책을 사용하여 요청에 응답합니다.

사전 조건 및 제한 사항

사전 조건

- Amazon Cognito 및 IAM 권한이 있는 AWS 계정
- 사용하려는 AWS 리소스에 대한 액세스
- ASP.NET Core 2.0.0 이상

아키텍처

기술 스택

- Amazon Cognito
- ASP.NET Core

대상 아키텍처

도구

도구, SDK 및 AWS 서비스

- Visual Studio 또는 Visual Studio Code
- [아마존. AspNetCore.Identity.Cognito](#) (1.0.4) — 패키지 NuGet
- [AWSSDK.S3 \(3.3.110.32\)](#) — 패키지 NuGet
- [Amazon Cognito](#)

코드

첨부된 .zip 파일에는 다음을 설명하는 샘플 파일이 포함되어 있습니다.

- 로그인한 사용자의 액세스 토큰을 검색하는 방법
- 액세스 토큰을 AWS 보안 인증 정보로 교환하는 방법
- AWS 보안 인증 정보로 Amazon Simple Storage Service(S3)에 액세스하는 방법

인증된 자격 증명을 위한 IAM 역할

```
{  
  "Version": "2012-10-17",
```

```

"Statement": [
  {
    "Effect": "Allow",
    "Action": [
      "mobileanalytics:PutEvents",
      "cognito-sync:*",
      "cognito-identity:*",
      "s3:ListAllMyBuckets*"
    ],
    "Resource": [
      "*"
    ]
  }
]
}

```

에픽

Amazon Cognito 사용자 풀을 생성

작업	설명	필요한 기술
사용자 풀을 생성합니다.	<ol style="list-style-type: none"> 1. AWS Management Console에 로그인하고 https://console.aws.amazon.com/cognito/home에서 Amazon Cognito 콘솔을 엽니다. 2. 사용자 풀 관리를 선택합니다. 3. 페이지 오른쪽 상단에서 사용자 풀 생성을 선택합니다. 4. 사용자 풀의 이름을 입력하고 기본값 검토를 선택한 다음 풀 생성을 선택합니다. 5. 풀 ID를 기록합니다. 	개발자
앱 클라이언트를 추가합니다.	앱을 생성하여 사용자 가입 및 로그인에 대해 내장 웹 페이지를 사용할 수 있습니다.	개발자

작업	설명	필요한 기술
	<ol style="list-style-type: none"> 1. 사용자 풀 페이지 왼쪽 탐색 모음에서 일반 설정에서 앱 클라이언트를 선택하고 나서 앱 클라이언트 추가를 선택합니다. 2. 앱 이름을 지정한 다음 앱 클라이언트 생성을 선택합니다. 3. 앱 클라이언트 ID와 클라이언트 암호를 기록해 둡니다. (클라이언트 암호를 보려면 세부 정보 보기를 선택하세요). 	

Amazon Cognito 자격 증명 풀 생성

작업	설명	필요한 기술
<p>자격 증명 풀을 생성합니다.</p>	<ol style="list-style-type: none"> 1. Amazon Cognito 콘솔에 로그인하고 자격 증명 풀 관리를 선택한 다음 새 자격 증명 풀 생성을 선택합니다. 2. 자격 증명 풀의 이름을 입력합니다. 3. 인증되지 않은 자격 증명을 활성화하려면 인증되지 않은 자격 증명 섹션에서 해당 옵션을 선택합니다. 4. 인증 제공자 섹션에서 사용자 풀 ID 및 앱 클라이언트 ID를 설정하여 Cognito 자격 증명 풀을 구성한 다음 풀 생성을 선택합니다. 	<p>개발자</p>

작업	설명	필요한 기술
<p>자격 증명 풀에 대한 IAM 역할을 할당합니다.</p>	<p>인증된 사용자 및 인증되지 않은 사용자의 IAM 역할을 편집하거나 기본값을 유지하고 허용을 선택할 수 있습니다. 이 패턴의 경우 인증된 IAM 역할을 편집하고 <code>s3:ListAllMyBuckets</code>에 대한 액세스 권한을 제공할 것입니다. 샘플 코드는 도구 섹션 앞부분에 제공된 IAM 역할을 참조하세요.</p>	<p>개발자</p>
<p>자격 증명 풀 ID를 복사합니다.</p>	<p>이전 단계에서 허용을 선택하면 Amazon Cognito 시작하기 페이지가 표시됩니다. 이 페이지에서 AWS 보안 인증 정보 가져오기 섹션에서 자격 증명 풀 ID를 복사하거나 오른쪽 상단에서 자격 증명 풀 편집을 선택하고 표시된 화면에서 자격 증명 풀 ID를 복사할 수 있습니다.</p>	<p>개발자</p>

샘플 앱 구성

작업	설명	필요한 기술
<p>샘플 ASP.NET Core 웹 앱을 복제합니다.</p>	<ol style="list-style-type: none"> 1. https://github.com/aws/provider.git에서 샘플 <code>.NET 코어 웹 앱을 복제하십시오. aws-aspnet-cognito-identity</code> 2. <code>samples</code> 폴더로 이동하여 솔루션을 엽니다. 이 프로젝트에서는 <code>appsettings.json</code> 파일을 구성하고 	<p>개발자</p>

작업	설명	필요한 기술
	로그인에 성공한 후 모든 S3 버킷을 렌더링하는 새 페이지를 추가합니다.	
종속성을 추가합니다.	ASP.NET Core 애플리케이션에 NuGet 대한 종속성을 추가하십시오. Amazon.AspNetCore.Identity.Cognito	개발자
appsettings.json에 구성 키와 값을 추가합니다.	appsettings.json 파일에 첨부된 appsettings.json 파일의 코드를 포함한 다음, 자리 표시자를 이전 단계의 값으로 바꿉니다.	개발자
새 사용자를 생성하고 로그인합니다.	Amazon Cognito 사용자 풀에 새 사용자를 생성하고 해당 사용자가 사용자 풀의 사용자 및 그룹에 있는지 확인합니다.	개발자
MyS3Buckets이라는 새 Razor Page를 생성합니다.	샘플 앱에 새 ASP.NET Core Razor Page를 추가하고 첨부된 샘플의 MyS3Bucket.cshtml 및 MyS3Bucket.cshtml.cs 의 콘텐츠를 교체합니다. _Layout.cshtml 페이지의 탐색 아래에 새 MyS3Bucket 페이지를 추가합니다.	개발자

문제 해결

문제	Solution
<p>GitHub 리포지토리에서 샘플 응용 프로그램을 연 후 샘플 프로젝트에 NuGet 패키지를 추가하려고 하면 오류가 발생합니다.</p>	<p>src 폴더에서 Amazon.AspNetCore.Identity.Cognito 프로젝트에 대한 참조를 Samples.sln 파일에서 제거했는지 확인하세요. 그러면 아무 문제 없이 Samples 프로젝트에 NuGet 패키지를 추가할 수 있습니다.</p>

관련 리소스

- [Amazon Cognito](#)
- [Amazon Cognito 사용자 풀](#)
- [Amazon Cognito 자격 증명 풀](#)
- [액세스 정책 예제](#)
- [GitHub — AWS ASP.NET 코그니토 자격 증명 공급자](#)

첨부

이 문서와 관련된 추가 콘텐츠에 액세스하려면 [attachment.zip](#) 파일의 압축을 풉니다.

AWS Directory Service를 사용하여 Amazon EC2에서 Microsoft SQL Server 인증하기

작성자: 자가디쉬 칸투부가타(AWS) 및 올루다훈 바데 아지다훈(AWS)

환경: PoC 또는 파일럿	소스: Active Directory	대상: AWS Directory Service
R 유형: 해당 없음	워크로드: Microsoft	기술: 보안, 자격 증명, 규정 준수, 분석, 데이터베이스
AWS 서비스: AWS Directory Service		

요약

이 패턴은 AWS Directory Service 디렉터리를 생성하고 이를 사용하여 Amazon Elastic Compute Cloud(Amazon EC2) 인스턴스에서 Microsoft SQL Server를 인증하는 방법을 설명합니다.

AWS Directory Service는 Amazon Cloud Directory 및 Microsoft Active Directory(AD)를 다른 AWS 서비스와 함께 사용할 수 있는 몇 가지 방법을 제공합니다. 디렉터리에는 사용자, 그룹 및 디바이스에 대한 정보가 저장되며 관리자는 이를 사용하여 정보 및 리소스에 대한 액세스를 관리합니다. AWS Directory Service는 클라우드에서 기존 Microsoft AD 또는 LDAP(Lightweight Directory Access Protocol) 인식 애플리케이션 사용을 원하는 사용자에게 다양한 디렉터리 옵션을 제공합니다. 또한 사용자, 그룹, 디바이스 및 액세스 권한을 관리하기 위해 디렉터리가 필요한 개발자에게도 동일한 선택 옵션을 제공합니다.

사전 조건 및 제한 사항

사전 조건

- 활성 상태의 AWS 계정
- 최소 2개의 프라이빗 서브넷과 2개의 퍼블릭 서브넷이 있는 Virtual Private Cloud(VPC)
- 서버를 도메인에 연결하기 위한 AWS Identity and Access Management(IAM) 역할

아키텍처

소스 기술 스택

- 소스는 온프레미스 Active Directory일 수 있습니다.

대상 기술 스택

- AWS Directory Service for Microsoft Active Directory(AWS Managed Microsoft AD)

대상 아키텍처

도구

- SQL Server Management Studio(SSMS)는 SQL Server 구성 요소에 대한 액세스, 구성 및 관리를 포함하여 Microsoft SQL Server를 관리하는 도구입니다.

에픽

디렉터리 설정

작업	설명	필요한 기술
디렉터리 유형으로 AWS Managed Microsoft AD를 선택합니다.	AWS Directory Service 콘솔 에 서디렉터리, 디렉터리 설정, AWS 관리형 Microsoft AD, 다음을 선택합니다.	DevOps
에디션을 선택하세요.	AWS Managed Microsoft AD에 이용 가능한 에디션에서 Standard Edition을 선택합니다.	DevOps
디렉터리 DNS 이름을 지정합니다.	정규화된 도메인 이름을 사용합니다. 이 이름은 VPC 내에서만 해석됩니다. 공개적으로 해	DevOps

작업	설명	필요한 기술
	석 가능해야 할 필요는 없습니다.	
관리자 암호를 설정합니다.	이름이 Admin인 기본 관리 사용자의 암호를 설정합니다.	DevOps
VPC 및 서브넷을 선택합니다.	디렉터리를 포함할 VPC와 도메인 컨트롤러의 서브넷을 선택합니다. 서브넷이 2개 이상 있는 VPC가 없으면 서브넷을 생성해야 합니다.	DevOps
디렉터리를 검토하고 시작합니다.	디렉터리의 에디션 및 가격 정보를 검토한 다음 디렉터리 생성을 선택합니다.	DevOps

도메인에서 SQL Server용 EC2 인스턴스 시작

작업	설명	필요한 기술
SQL 서버용 AMI를 선택합니다.	이 에픽의 단계는 AWS Managed Microsoft AD 디렉터리에 Windows EC2 인스턴스를 원활하게 연결합니다. Amazon EC2 콘솔 에서 인스턴스 시작을 선택한 다음 SQL Server에 적합한 Amazon Machine Image(AMI)를 선택합니다.	DevOps, DBA
인스턴스 세부 정보를 구성합니다.	SQL Server의 요구 사항을 충족하도록 Windows 인스턴스를 구성하세요.	DevOps, DBA

작업	설명	필요한 기술
키 페어 이름을 선택합니다.	키 페어를 선택한 다음 해당 인스턴스를 시작합니다.	DevOps, DBA
네트워크를 추가합니다.	디렉터리가 생성된 VPC를 선택할 수 있습니다.	DevOps, DBA
IAM 역할을 선택합니다.	고급 설정에서 AWS 관리형 정책 AmazonSSMManagedInstanceCore 및 AmazonSSMDirectoryServiceAccess 이(가) 있고 거기에 연결된 IAM 프로파일을 선택합니다.	DevOps, DBA
서브넷을 추가합니다.	VPC에서 퍼블릭 서브넷 중 하나를 선택하세요. 선택한 서브넷에서는 인터넷 게이트웨이로 모든 외부 트래픽이 라우팅되어야 합니다. 그렇지 않으면 인스턴스를 원격으로 연결할 수 없게 됩니다.	DevOps, DBA
도메인을 선택합니다.	도메인 조인디렉터리 목록에서 생성한 도메인을 선택합니다.	DevOps, DBA
인스턴스를 시작합니다.	인스턴스 시작을 선택합니다.	DBA

Directory Service를 사용하여 SQL 서버 인증하기

작업	설명	필요한 기술
Windows 관리자로 로그인합니다.	Windows 관리자 보안 인증 정보를 사용하여 Windows EC2 인스턴스에 로그인합니다.	DBA

작업	설명	필요한 기술
SQL Server에 로그인합니다.	SQL Server Management Studio(SSMS)를 시작하고 Windows 인증 방법을 사용하여 SQL Server에 로그인합니다.	DBA
디렉터리 사용자의 로그인을 생성합니다.	SSMS에서 보안을 선택한 다음 새 로그인을 선택합니다.	DBA
로그인 이름을 검색합니다.	로그인 텍스트 상자 옆에 있는 검색 버튼을 선택합니다.	DBA
위치를 선택합니다.	사용자 또는 그룹 선택 대화 상자에서 위치를 선택합니다.	DBA
네트워크 보안 인증 정보를 입력합니다.	디렉터리 서비스를 생성할 때 사용한 정규화된 네트워크 보안 인증 정보를 입력합니다. 예제: test.com\admin	DBA
디렉터리를 선택합니다.	AWS 디렉터리 이름을 선택하고 확인을 선택합니다.	DBA
객체 이름을 선택합니다.	로그인을 생성하려는 사용자를 선택합니다. 위치를 선택하고, 전체 디렉터를 선택하고, 사용자를 검색하고, 로그인을 추가합니다.	DBA
SQL Server 인스턴스에 로그인합니다.	도메인 보안 인증 정보를 사용하여 SQL Server용 Windows EC2 인스턴스에 로그인합니다.	DBA
도메인 사용자로 SQL Server에 로그인합니다.	SSMS를 시작하고 Windows 인증 수단을 사용하여 데이터베이스 엔진에 연결합니다.	DBA

관련 리소스

- [AWS Directory Service 설명서](#) (AWS 웹사이트)
- [AWS Managed Microsoft AD directory 생성](#) (AWS Directory Service 설명서)
- [Windows EC2 인스턴스에 원활하게 연결](#) (AWS Directory Service 설명서)
- [AWS의 Microsoft SQL Server](#) (AWS 웹사이트)
- [SSMS 설명서](#) (Microsoft 웹사이트)
- [SQL 서버에서 로그인 생성](#) (SQL Server 설명서)

인시던트 응답 및 포렌식 자동화

작성자: Lucas Kauffman 및 Tomek Jakubowski

코드 리포지토리: [aws-automated-incident-response-및-포렌식](#)

환경: 프로덕션

기술: 보안, 자격 증명 및 규정 준수

AWS 서비스: Amazon EC2, AWS Lambda, Amazon S3, AWS Security Hub, AWS Identity and Access Management

요약

이 패턴은 AWS Lambda 함수를 사용하여 다음을 제공하는 프로세스 집합을 배포합니다.

- 최소한의 지식으로 인시던트 응답 프로세스를 시작하는 방법
- 보안 인시던트 응답 가이드에 따른 자동화되고 반복 가능한 프로세스
- 자동화 절차 운영, 아티팩트 저장, 포렌식 환경 생성을 위한 계정의 분리

자동화된 인시던트 응답 및 포렌식 프레임워크는 다음 단계로 구성된 표준 디지털 포렌식 프로세스를 따릅니다.

1. 제한
2. 인수
3. 검사
4. 분석

정적 데이터(예: 획득한 메모리 또는 디스크 이미지) 및 분리된 시스템에 있는 라이브 동적 데이터에 대해 조사를 수행할 수 있습니다.

자세한 내용은 [추가 정보](#) 섹션을 참조하세요.

사전 조건 및 제한 사항

사전 조건

- 다음 2개의 계정:
 - 보안 계정: 기존 계정일 수 있지만 새 계정인 것이 좋습니다.
 - 포렌식 계정: 새 계정인 것이 좋습니다.
- AWS Organizations 설정
- Organizations 멤버 계정에서:
 - Amazon Elastic Compute Cloud(Amazon EC2) 역할은 Amazon Simple Storage Service(S3)에 대한 Get 및 List 액세스 권한이 있어야 하며 AWS Systems Manager가 액세스할 수 있어야 합니다. AmazonSSMManagedInstanceCore 관리형 역할을 사용하는 것이 좋습니다. 인시던트 응답이 시작되면 이 역할이 EC2 인스턴스에 자동으로 연결된다는 점을 유념합니다. 응답이 완료되면 Identity and Access Management(IAM)은 인스턴스에 대한 모든 권한을 제거합니다.
 - 멤버 계정과 인시던트 응답 및 분석 VPC의 Virtual Private Cloud(VPC) 엔드포인트. 이러한 엔드포인트는 S3 게이트웨이, EC2 메시지, SSM, SSM 메시지입니다.
- EC2 인스턴스에 설치된 Command Line Interface(CLI). EC2 인스턴스에 CLI가 설치되어 있지 않은 경우 디스크 스냅샷과 메모리 획득이 작동하려면 인터넷 액세스가 필요합니다. 이 경우 스크립트는 인터넷에 접속하여 CLI 설치 파일을 다운로드하고 인스턴스에 설치합니다.

제한 사항

- 이 프레임워크는 전자 증거로 간주되어 법원에 제출할 수 있는 아티팩트를 생성하려는 의도가 없습니다.
- 현재 이 패턴은 x86 아키텍처에서 실행되는 Linux 기반 인스턴스만을 지원합니다.

아키텍처

대상 기술 스택

- AWS CloudFormation
- AWS CloudTrail
- AWS Config
- IAM
- Lambda

- Amazon S3
- Key Management System (KMS)
- AWS Security Hub
- Amazon Simple Notification Service(SNS)
- AWS Step Functions

대상 아키텍처

대상 환경은 멤버 계정 외에도 두 개의 기본 계정, 즉 보안 계정과 포렌식 계정으로 구성됩니다. 두 개의 계정이 사용되는 이유는 다음과 같습니다.

- 포렌식 분석 실패 시 영향 범위를 줄이기 위해 다른 고객 계정과 분리하기 위해
- 분석 대상 아티팩트의 격리 및 무결성 보호를 보장하는 데 도움이 되기 위해
- 조사를 기밀로 유지하기 위해
- 위협 행위자가 Service Quotas를 초과하여 Amazon EC2 인스턴스를 인스턴스화해서 조사를 수행하지 못하게 함으로써 손상된 AWS 계정에서 즉시 사용할 수 있는 모든 리소스를 사용했을 수 있는 상황을 방지하기 위해.

또한 별도의 보안 및 포렌식 계정을 보유하면 증거 수집을 위한 응답자와 증거 분석을 위한 조사자 등 별도의 역할을 생성할 수 있습니다. 각 역할은 별도의 계정에 액세스할 수 있습니다.

다음 다이어그램은 계정 간의 상호 작용만 보여줍니다. 각 계정의 세부 정보는 후속 다이어그램에 표시되며 전체 다이어그램이 첨부되어 있습니다.

다음 다이어그램은 멤버 계정을 보여줍니다.

1. 이벤트는 Slack Amazon SNS 주제로 전송됩니다.

다음 다이어그램은 보안 계정을 보여줍니다.

2. 보안 계정의 SNS 주제는 포렌식 이벤트를 시작합니다.

다음 다이어그램은 포렌식 계정을 보여줍니다.

보안 계정은 메모리 및 디스크 이미지 획득을 위해 두 개의 기본 Step Functions 워크플로를 생성하는 곳입니다. 워크플로가 실행되고 나면 인시던트와 관련된 EC2 인스턴스가 있는 멤버 계정에 액세스하여 메모리 덤프 또는 디스크 덤프를 수집하는 Lambda 함수 집합을 시작합니다. 그러면 해당 아티팩트가 포렌식 계정에 저장됩니다.

포렌식 계정은 Step Functions 워크플로에서 수집한 아티팩트를 분석 아티팩트 S3 버킷에 보관합니다. 포렌식 계정에는 포렌식 인스턴스의 Amazon Machine Image(AMI)를 빌드하는 EC2 Image Builder 파이프라인도 있습니다. 현재, 이미지는 SANS SIFT 워크스테이션을 기반으로 합니다.

빌드 프로세스는 인터넷에 연결된 Maintenance VPC를 사용합니다. 이 이미지는 나중에 Analysis VPC에서 수집된 아티팩트를 분석하기 위해 EC2 인스턴스를 실행하는 데 사용할 수 있습니다.

Analysis VPC가 인터넷에 연결되어 있지 않습니다. 기본적으로 이 패턴은 3개의 프라이빗 분석 서브넷을 생성합니다. VPC의 서브넷 수에 대한 할당량인 최대 200개의 서브넷을 생성할 수 있지만, AWS Systems Manager Sessions Manager가 해당 서브넷에서 명령 실행을 자동화하려면 VPC 엔드포인트에 해당 서브넷을 추가해야 합니다.

모범 사례 관점에서 볼 때 CloudTrail AWS와 AWS Config를 사용하여 다음을 수행하는 것이 좋습니다.

- 포렌식 계정의 변경 내용 추적
- 저장 및 분석된 아티팩트의 액세스 및 무결성을 모니터링합니다.

워크플로

다음 다이어그램은 인스턴스가 손상된 시점부터 분석 및 억제되는 시점까지의 프로세스 및 의사 결정 트리를 포함하는 워크플로의 주요 단계를 보여줍니다.

1. SecurityIncidentStatus태그가 Analyze 값으로 설정되었습니까? 설정되었다면 다음을 수행합니다.
 - a. Systems Manager 및 Amazon S3에 대한 올바른 IAM 프로파일을 연결합니다.
 - b. Slack의 Amazon SNS 대기열로 Amazon SNS 메시지를 전송합니다.
 - c. Amazon SNS 메시지를 SecurityIncident 대기열로 전송합니다.
 - d. 메모리 및 디스크 획득 상태 시스템을 호출합니다.
2. 메모리와 디스크를 획득했습니까? 아니라면 오류가 발생한 것입니다.
3. Contain 태그로 EC2 인스턴스를 지정합니다.

4. IAM 역할 및 보안 그룹을 연결하여 인스턴스를 완전히 격리합니다.

자동화 및 규모 조정

이 패턴의 목적은 단일 Organizations 조직 내 여러 계정 전반적으로 인시던트 응답 및 포렌식을 수행할 수 있도록 확장 가능한 솔루션을 제공하는 것입니다.

도구

AWS 서비스

- [AWS는 AWS CloudFormation](#) 리소스를 설정하고, 빠르고 일관되게 프로비저닝하고, AWS 계정 및 지역 전반의 수명 주기 전반에 걸쳐 리소스를 관리할 수 있도록 지원합니다.
- [Command Line Interface\(CLI\)](#)는 명령줄 쉘에서 명령을 사용하여 서비스와 상호 작용할 수 있는 오픈 소스 도구입니다.
- [Identity and Access Management\(IAM\)](#)는 사용자에 대한 인증 및 권한 부여를 제어함으로써 AWS 리소스에 대한 액세스를 안전하게 제어할 수 있습니다.
- [Key Management Service\(KMS\)](#)를 사용하면 암호화 키를 생성하고 제어하여 데이터를 보호할 수 있습니다.
- [Lambda](#)는 서버를 프로비저닝하거나 관리할 필요 없이 코드를 실행하는 데 도움이 되는 컴퓨팅 서비스입니다. 필요할 때만 코드를 실행하며 자동으로 확장이 가능하므로 사용한 컴퓨팅 시간만큼만 비용을 지불합니다.
- [Amazon Simple Storage Service\(Amazon S3\)](#)는 원하는 양의 데이터를 저장, 보호 및 검색하는 데 도움이 되는 클라우드 기반 객체 스토리지 서비스입니다.
- [AWS Security Hub](#)에서는 AWS의 보안 상태에 대한 포괄적인 보기가 제공됩니다. 이를 통해 AWS 환경에서 보안 업계 표준 및 모범 사례를 준수하는지 확인할 수도 있습니다.
- [Amazon Simple Notification Service\(Amazon SNS\)](#)를 사용하면 웹 서버 및 이메일 주소를 포함하여 게시자와 클라이언트 간의 메시지 교환을 조정하고 관리할 수 있습니다.
- [AWS Step Functions](#)는 Lambda 함수와 기타 AWS 서비스를 결합할 수 있는 서버리스 오케스트레이션 서비스로서 비즈니스 크리티컬 애플리케이션을 구축합니다.
- [AWS Systems Manager](#)는 AWS 클라우드에서 실행되는 애플리케이션과 인프라를 관리하는 데 도움이 됩니다. 애플리케이션 및 리소스 관리를 간소화하고, 운영 문제의 감지 및 해결 시간을 단축하며, AWS 리소스를 규모에 따라 안전하게 관리하는 데 도움이 됩니다.

코드

코드와 구체적인 구현 및 사용 지침은 GitHub [자동 사고 대응 및 포렌식](#) 프레임워크 리포지토리를 참조하십시오.

에픽

템플릿 배포 CloudFormation

작업	설명	필요한 기술
CloudFormation 템플릿 배포.	<p>CloudFormation 템플릿은 1~7로 표시되어 있으며, 스크립트 이름의 첫 단어는 템플릿을 배포해야 하는 계정을 나타냅니다. CloudFormation 템플릿 실행 순서가 중요하다는 점에 유의하세요.</p> <ul style="list-style-type: none"> 1-forensic-AnalysisVPCnS3Buckets.yam 1 : 포렌식 계정에 배포되었습니다. S3 버킷과 분석 VPC를 생성하고 활성화합니다. CloudTrail 2-forensic-MaintenanceVPCnEC2ImageBuilderPipeline.yaml : SANS SIFT를 기반으로 유지 관리 VPC 및 이미지 빌더 파이프라인을 배포합니다. 3-security_IR-Disk_Mem_automation.yaml : 디스크 및 메모리 획득을 활성화하는 함수를 보안 계정에 배포합니다. 	AWS 관리자

작업	설명	필요한 기술
	<ul style="list-style-type: none"> • 4-security_LiME_Volatility_Factory.yaml : 빌드 함수를 시작하여 지정 AMI ID를 기반으로 메모리 모듈 생성을 시작합니다. AMI ID는 리전마다 다르다는 점을 유념합니다. 새 메모리 모듈이 필요할 때마다 새 AMI ID를 사용하여 이 스크립트를 다시 실행할 수 있습니다. 이를 골든 이미지 AMI 빌더 파이프라인(사용자 환경에서 사용하는 경우)과 통합하는 것을 고려합니다. • 5-member-IR-automation.yaml : 인시던트 응답 프로세스를 시작하는 멤버 인시던트 응답 자동화 함수를 생성합니다. 이를 통해 계정 전반적으로 Amazon Elastic Block Store(Amazon EBS) 볼륨을 공유하고, 인시던트 응답 프로세스 중에 Slack 채널에 자동으로 게시하며, 포렌식 프로세스를 시작하고, 프로세스 완료 후 인스턴스를 격리할 수 있습니다. • 6-forensic-artifact-s3-policies.yaml : 모든 스크립트가 배포된 후 이 스크립트는 모든 교차 계정 상호 작용에 필요한 권한을 수정합니다. 	

작업	설명	필요한 기술
	<ul style="list-style-type: none"> 7-security-IR-vpc.yaml : 인시던트 응답 볼륨 처리에 사용되는 VPC를 구성합니다. <p>특정 EC2 인스턴스에 대한 인시던트 응답 프레임워크를 시작하려면 키 SecurityIncidentStatus 및 값 Analyze(이)가 포함된 태그를 생성합니다. 그러면 격리 및 메모리는 물론 디스크 획득을 자동으로 시작하는 멤버 Lambda 함수가 시작됩니다.</p>	
프레임워크를 운영합니다.	<p>또한 Lambda 함수는 종료(또는 실패) 시 Contain 태그로 자산을 다시 지정합니다. 그러면 제한이 시작되어 인바운드/아웃바운드 없는 보안 그룹 및 모든 액세스를 허용하지 않는 IAM 역할을 가진 인스턴스가 완전히 격리됩니다.</p> <p>리포지토리의 단계를 따르십시오. GitHub</p>	AWS 관리자

사용자 지정 Security Hub 작업 배포

작업	설명	필요한 기술
CloudFormation 템플릿을 사용하여 사용자 지정 Security Hub 작업을 배포합니다.	Security Hub의 드롭다운 목록을 사용할 수 있도록 사용자 지정 작업을 생성하려면	AWS 관리자

작업	설명	필요한 기술
	<p>Modules/SecurityHub Custom Actions/SecurityHubCustomActions.yaml CloudFormation 템플릿을 배포하십시오. 그런 다음 각 멤버 계정에서 IRAutomation 역할을 수정하여 작업을 실행하는 Lambda 함수가 IRAutomation 역할을 떠날 수 있도록 합니다. 자세한 내용은 GitHub 리포지토리를 참조하십시오.</p>	

관련 리소스

- [보안 인시던트 응답 가이드](#)

추가 정보

이 환경을 사용하여 보안 운영 센터(SOC) 팀은 다음을 통해 보안 인시던트 응답 프로세스를 개선할 수 있습니다.

- 분리된 환경에서 포렌식을 수행하여 프로덕션 리소스의 우발적인 손상을 방지할 수 있는 능력 확보
- 제한 및 분석을 수행할 수 있는 표준화되고 반복 가능하며 자동화된 프로세스 확보.
- 모든 계정 소유자 또는 관리자에게 태그 사용 방법에 대한 최소한의 지식으로 인시던트 응답 프로세스를 시작할 수 있는 능력을 제공
- 대규모 환경의 소음 없이 인시던트 분석 및 포렌식을 수행할 수 있는 표준화되고 깨끗한 환경 확보
- 여러 분석 환경을 병렬로 생성할 수 있는 능력 확보
- SOC 리소스를 클라우드 포렌식 환경의 유지 관리 및 문서화 대신 인시던트 응답에 집중
- 수동 프로세스에서 자동화된 프로세스로 전환하여 확장성 달성
- 일관성을 유지하고 반복 가능한 작업을 피하기 위해 CloudFormation 템플릿 사용

또한 영구적 인프라를 사용하지 않고, 필요할 때 리소스에 대한 비용을 지불합니다.

첨부

이 문서와 관련된 추가 콘텐츠에 액세스하려면 [attachment.zip](#) 파일의 압축을 풉니다.

AWS Security Hub 표준 조사 결과에 대한 문제 해결 자동화

작성자: Chandini Penmetsa 및 Aromal Raj Jayarajan

환경: 프로덕션

기술: 보안, 자격 증명, 규정 준수

워크로드: 기타 모든 워크로드

AWS 서비스: AWS CloudFormation, 아마존 CloudWatch, AWS 람다, AWS 보안 허브, 아마존 SNS

요약

AWS Security Hub를 사용하면 다음과 같은 표준 모범 사례를 확인할 수 있습니다.

- Foundational Security Best Practices
- CIS 기반 벤치마크
- Payment Card Industry Data Security Standard(PCI DSS)

이러한 각 표준에는 사전 정의된 제어 항목이 있습니다. Security Hub는 지정된 계정의 제어 항목을 확인하고 조사 결과를 보고합니다.

AWS Security Hub는 EventBridge 기본적으로 모든 조사 결과를 아마존에 전송합니다. 이 패턴은 AWS 기본 보안 모범 사례 표준 결과를 식별하는 EventBridge 규칙을 배포하는 보안 제어를 제공합니다. 이 규칙은 기반적인 보안 모범 사례 표준의 자동 조건, Virtual Private Cloud(VPC), Amazon Elastic Block Store(Amazon EBS), Amazon Relational Database Service(Amazon RDS)에 대한 다음과 같은 조사 결과를 식별합니다.

- [AutoScaling.1] 로드 밸런서와 연결된 Auto Scaling 그룹은 로드 밸런서 상태 점검을 사용해야 합니다.
- [EC2.2] VPC 기본 보안 그룹은 인바운드 및 아웃바운드 트래픽을 허용하지 않아야 합니다.
- [EC2.6] VPC 플로 로깅은 모든 VPC에서 활성화되어야 합니다.
- [EC2.7] EBS 기본 암호화를 활성화해야 합니다.
- [RDS.1] RDS 스냅샷은 비공개 상태여야 합니다.

- [RDS.6] RDS DB 인스턴스 및 클러스터에 대한 향상된 모니터링을 구성해야 합니다.
- [RDS.7] RDS 클러스터에는 삭제 방지 기능이 활성화되어 있어야 합니다.

EventBridge 규칙은 이러한 결과를 AWS Lambda 함수로 전달하여 검색 결과를 수정합니다. 그러면 Lambda 함수가 문제 해결 정보가 포함된 알림을 Amazon Simple Notification Service(Amazon SNS) 주제에 보냅니다.

사전 조건 및 제한 사항

사전 조건

- 활성 상태의 AWS 계정
- 문제 해결 알림을 수신하려는 이메일 주소
- 제어 기능을 배포하려는 리전에서 활성화된 Security Hub 및 AWS Config
- Lambda 코드를 업로드하기 위한 제어와 동일한 리전에 있는 Amazon Simple Storage Service(S3) 버킷

제한 사항

- 이 보안 제어는 보안 제어 배포 후 보고된 새로운 조사 결과의 문제를 자동으로 해결합니다. 기존 조사 결과의 문제를 해결하려면 Security Hub 콘솔에서 조사 결과를 수동으로 선택합니다. 그런 다음 작업에서 AWS에서 배포의 일부로 생성한 AFSBpremedy 사용자 지정 작업을 선택합니다. CloudFormation
- 이 보안 제어는 리전과 관련이 있으므로 모니터링하고자 하는 AWS 리전에 배포해야 합니다.
- EC2.6 해결 방안의 경우 VPC 흐름 CloudWatch 로그를 활성화하기 위해 Amazon Logs 로그 그룹 이/vpc_id 형식으로 생성됩니다. VpcFlowLogs 이름이 동일한 로그 그룹이 있다면 기존 로그 그룹을 사용합니다.
- EC2.7 문제 해결의 경우 Amazon EBS 기본 암호화를 활성화하려면 기본 Key Management Service(KMS) 키를 사용합니다. 이 변경으로 인해 암호화를 지원하지 않는 특정 인스턴스를 사용할 수 없게 됩니다.

아키텍처

대상 기술 스택

- Lambda 함수

- Amazon SNS 주제
- EventBridge 규칙
- Lambda 함수, VPC 흐름 로그, Amazon Relational Database Service(Amazon RDS) 확장 모니터링을 위한 Identity and Access Management(IAM) 역할

대상 아키텍처

자동화 및 규모 조정

AWS Organizations를 사용하는 경우, [CloudFormation StackSetsAWS](#)를 사용하여 모니터링하려는 여러 계정에 이 템플릿을 배포할 수 있습니다.

도구

도구

- [AWS CloudFormation](#) — CloudFormation AWS는 인프라를 코드로 사용하여 AWS 리소스를 모델링하고 설정하는 데 도움이 되는 서비스입니다.
- [EventBridge](#)— EventBridge Amazon은 자체 애플리케이션, 서비스형 소프트웨어 (SaaS) 애플리케이션 및 AWS 서비스로부터 실시간 데이터 스트림을 전달하여 해당 데이터를 Lambda 함수와 같은 대상으로 라우팅합니다.
- [Lambda](#) - Lambda는 서버를 프로비저닝하거나 관리하지 않고도 코드를 실행할 수 있게 해주는 컴퓨팅 서비스입니다.
- [Amazon S3](#) — Amazon Simple Storage Service(S3)는 웹 사이트, 모바일 애플리케이션, 백업 및 데이터 레이크를 포함하여 다양한 스토리지 솔루션에 사용할 수 있는 확장성이 뛰어난 객체 스토리지 서비스입니다.
- [Amazon SNS](#) – Amazon Simple Notification Service(SNS)는 웹 서버와 이메일 주소를 포함하여 게시자와 클라이언트 간에 메시지를 전달 또는 전송하는 것을 조정하고 관리합니다. 구독자는 구독하는 주제에 게시된 모든 메시지를 수신하며 주제에 대한 모든 구독자는 동일한 메시지를 수신합니다.

모범 사례

- [9 개의 AWS Security Hub 모범 사례](#)
- [Foundational Security Best Practices 표준](#)

에픽

보안 제어의 배포

작업	설명	필요한 기술
S3 버킷을 정의합니다.	Amazon S3 콘솔에서 선행 슬래시를 포함하지 않는 고유한 이름을 가진 S3 버킷을 선택하거나 생성합니다. S3 버킷 이름은 전역 수준에서 고유하며, 네임스페이스는 모든 AWS 계정이 공유합니다. S3 버킷은 평가 중인 Security Hub 조사 결과와 동일한 리전에 있어야 합니다.	클라우드 아키텍트
Lambda 코드를 S3 버킷에 업로드	“첨부 파일” 섹션에 나와 있는 Lambda 코드.zip 파일을 정의된 S3 버킷에 업로드합니다.	클라우드 아키텍트
AWS CloudFormation 템플릿을 배포합니다.	이 패턴에 첨부하여 제공된 AWS CloudFormation 템플릿을 배포하십시오. 다음 에픽에서는 파라미터 값을 제공하세요.	클라우드 아키텍트

AWS CloudFormation 템플릿에서 파라미터 완성

작업	설명	필요한 기술
S3 버킷 이름을 입력합니다.	첫번째 에픽에서 생성한 S3 버킷의 이름을 입력합니다.	클라우드 아키텍트
Amazon S3 접두사를 입력합니다.	S3 버킷에 있는 Lambda 코드 .zip 파일의 위치를 앞에 슬래시 없이 입력합니다(예: <directory>/<file-name>.zip).	클라우드 아키텍트

작업	설명	필요한 기술
SNS 주제 ARN을 입력합니다.	문제 해결 알림에 기존 SNS 주제를 사용하려는 경우 SNS 주제 Amazon 리소스 이름(ARN)을 입력합니다. 새 SNS 주제를 사용하려면 값을 “없음”(기본 값)으로 유지합니다.	클라우드 아키텍트
이메일 주소를 입력합니다.	수정 알림을 받을 이메일 주소를 입력합니다 (AWS에서 SNS 주제를 CloudFormation 생성하도록 하려는 경우에만 필요).	클라우드 아키텍트
로깅 수준을 정의합니다.	Lambda 함수의 로깅 수준 및 빈도를 정의합니다. ‘정보’는 애플리케이션 진행 상황에 대한 자세한 정보 메시지를 의미합니다. ‘오류’는 여전히 애플리케이션을 계속 실행할 수 있는 오류 이벤트를 의미합니다. ‘경고’는 잠재적으로 위험한 상황을 의미합니다.	클라우드 아키텍트
VPC 플로 로그 IAM 역할 ARN을 입력합니다.	VPC 플로 로그로 사용될 IAM 역할 ARN을 입력합니다. (입력으로 “없음”을 입력하면 AWS는 IAM 역할을 CloudFormation 생성하여 사용합니다.)	클라우드 아키텍트
RDS 확장 모니터링 IAM 역할 ARN을 입력합니다.	RDS 확장 모니터링에 사용될 IAM 역할 ARN을 제공합니다. (“없음”을 입력하면 AWS는 IAM 역할을 CloudFormation 생성하여 사용합니다.)	클라우드 아키텍트

구독 확인

작업	설명	필요한 기술
Amazon SNS 구독을 확인합니다.	템플릿이 성공적으로 배포될 때, 새 SNS 주제가 생성되면 제공한 이메일 주소로 구독 메시지가 전송됩니다. 문제 해결 알림을 수신하려면 이 구독 이메일 메시지를 확인해야 합니다.	클라우드 아키텍트

관련 리소스

- [AWS CloudFormation 콘솔에 스택 생성](#)
- [AWS Lambda](#)
- [AWS Security Hub](#)

첨부

이 문서와 관련된 추가 콘텐츠에 액세스하려면 [attachment.zip](#) 파일의 압축을 풉니다.

Amazon Inspector와 Security Hub를 사용하여 교차 계정 워크로드에 대한 보안 스캔 자동화

작성자: Ramya Pulipaka 및 Mikesh Khanal

환경: 프로덕션

기술: 보안, 자격 증명, 규정 준수, DevOps

AWS 서비스: 아마존 인스펙터, 아마존 SNS, AWS 램다, AWS 보안 허브, 아마존 CloudWatch

요약

이 패턴은 Amazon Web Services(AWS) 클라우드의 교차 계정 워크로드에 대한 취약성을 자동으로 스캔하는 방법을 설명합니다.

이 패턴은 태그별로 그룹화된 Amazon Elastic Compute Cloud(Amazon EC2) 인스턴스의 호스트 기반 스캔 또는 네트워크 기반 Amazon Inspector 스캔에 대한 일정을 생성하는 데 도움이 됩니다. AWS CloudFormation 스택은 필요한 모든 AWS 리소스 및 서비스를 AWS 계정에 배포합니다.

Amazon Inspector의 조사 결과는 AWS Security Hub로 보내져 계정, AWS 리전, Virtual Private Cloud(VPC), EC2 인스턴스 전반의 취약성에 대한 통찰력을 제공합니다. 이러한 조사 결과를 이메일로 받거나, HTTP 엔드포인트를 사용하여 티켓팅 도구, SIEM(보안 정보 및 이벤트 관리) 소프트웨어 또는 기타 타사 보안 솔루션으로 조사 결과를 전송하는 Amazon Simple Notification Service(SNS) 주제를 생성할 수 있습니다.

사전 조건 및 제한 사항

사전 조건

- Amazon SNS로부터 이메일 알림을 수신하기 위한 기존 이메일 주소.
- 티켓팅 도구, SIEM 소프트웨어 또는 기타 타사 보안 솔루션에서 사용하는 기존 HTTP 엔드포인트.
- 중앙 감사 계정을 포함하여 교차 계정 워크로드를 호스팅하는 활성 AWS 계정.
- Security Hub, 활성화되고 및 구성됨. 이 패턴은 Security Hub 없이 사용할 수 있지만, Security Hub를 통해 생성되는 인사이트 때문에 Security Hub를 사용하는 것이 좋습니다. 자세한 내용은 AWS Security Hub 설명서에서 [Security Hub 설정](#)을 참조하십시오.

- 스캔하려는 각 EC2 인스턴스에 Amazon Inspector 에이전트를 설치해야 합니다. [AWS Systems Manager Run Command](#)를 사용하여 다중 EC2 인스턴스에 Amazon Inspector 에이전트를 설치할 수 있습니다.

기술

- AWS에서 스택 세트를 self-managed 사용하고 service-managed 권한을 부여한 경험이 CloudFormation 있습니다. self-managed 권한을 사용하여 특정 리전의 특정 계정에 스택 인스턴스를 배포하려면 필요한 Identity and Access Management(IAM) 역할을 생성해야 합니다. service-managed 권한을 사용하여 특정 리전의 Organizations에 의해 스택 인스턴스를 배포하려면 필요한 Identity and Access Management(IAM) 역할을 생성해야 합니다. 자세한 내용은 AWS CloudFormation 설명서의 [스택 세트 생성](#)을 참조하십시오.

제한 사항

- 계정의 EC2 인스턴스에 태그가 적용되지 않은 경우 Amazon Inspector는 해당 계정의 모든 EC2 인스턴스를 스캔합니다.
- AWS CloudFormation 스택 세트와 onboard-audit-account .yaml 파일 (첨부) 은 동일한 지역에 배포되어야 합니다.
- 기본적으로 [Amazon Inspector Classic](#)은 집계된 조사 결과를 지원하지 않습니다. Security Hub는 여러 계정 또는 AWS 리전에 대한 평가를 보는 데 권장되는 솔루션입니다.
- 이 패턴의 접근 방식은 리전별로 제한 값이 다를지라도 미국 동부(버지니아 북부) 리전(us-east-1)의 SNS 주제에 대한 30,000 초당 트랜잭션(TPS)의 게시 할당량에 따라 확장할 수 있습니다. 더욱 효과적으로 확장하고 데이터 손실을 방지하기 위해 SNS 주제 앞에 Amazon Simple Queue Service(Amazon SQS)를 사용하는 것이 좋습니다.

아키텍처

다음 다이어그램은 EC2 인스턴스를 자동으로 스캔하는 워크플로를 보여줍니다.

이 워크플로우는 다음 단계로 구성됩니다.

1. 아마존 EventBridge 규칙은 크론 표현식을 사용하여 특정 일정에 따라 자체 시작하고 Amazon Inspector를 시작합니다.

2. Amazon Inspector는 계정에서 태그가 지정된 EC2 인스턴스를 스캔합니다.
3. Amazon Inspector는 조사 결과를 Security Hub로 전송하고, Security Hub는 워크플로, 우선 순위 지정, 문제 해결에 대한 통찰력을 생성합니다.
4. 또한 Amazon Inspector는 평가 상태를 감사 계정의 SNS 주제로 전송합니다. findings reported 이벤트가 SNS 주제에 게시되면 Lambda 함수가 호출됩니다.
5. Lambda 함수는 조사 결과를 가져와서 형식을 지정하고 감사 계정의 다른 SNS 주제로 전송합니다.
6. 조사 결과는 SNS 주제를 구독하는 이메일 주소로 전송됩니다. 전체 세부 정보 및 권장 사항은 JSON 형식으로 구독된 HTTP 엔드포인트에 전송됩니다.

기술 스택

- AWS Control Tower
- EventBridge
- IAM
- Amazon Inspector
- Lambda
- Security Hub
- Amazon SNS

도구

- [AWS CloudFormation](#) — AWS를 CloudFormation 사용하면 AWS 리소스를 모델링하고 설정하여 리소스를 관리하는 시간을 줄이고 애플리케이션에 더 많은 시간을 집중할 수 있습니다.
- [AWS CloudFormation StackSets](#) — AWS는 단일 작업으로 여러 계정 및 지역에서 스택을 생성, 업데이트 또는 삭제할 수 있도록 하여 스택의 기능을 CloudFormation StackSets 확장합니다.
- [Control Tower](#) — Control Tower는 Organizations를 비롯한 여러 다른 AWS 서비스의 기능을 결합하고 통합하는 추상화 또는 오케스트레이션 계층을 생성합니다.
- [Amazon EventBridge](#) — EventBridge 다양한 소스의 데이터에 애플리케이션을 쉽게 연결할 수 있게 해주는 서버리스 이벤트 버스 서비스입니다.
- [Lambda](#) - Lambda는 서버를 프로비저닝하거나 관리하지 않고도 코드를 실행할 수 있게 해주는 컴퓨팅 서비스입니다.
- [AWS Security Hub](#) – Security Hub에서는 AWS에서 보안 상태를 포괄적으로 파악할 수 있으며 보안 업계 표준 및 모범 사례와 대한 환경을 확인할 수 있습니다.

- [Amazon SNS](#) – Amazon Simple Notification Service(Amazon SNS)는 게시자에서 구독자로 메시지를 전송을 제공하는 관리형 서비스입니다.

에픽

AWS CloudFormation 템플릿 배포

작업	설명	필요한 기술
<p>감사 계정에 AWS CloudFormation 템플릿을 배포합니다.</p>	<p>onboard-audit-account.yaml 파일(첨부됨)을 다운로드하여 컴퓨터의 로컬 경로에 저장합니다.</p> <p>감사 계정의 AWS Management 콘솔에 로그인하고 AWS CloudFormation 콘솔을 연 다음 [Create stack] 을 선택합니다.</p> <p>사전 조건섹션에서 템플릿 준비를 선택하고, 템플릿이 준비 완료을 선택합니다. 템플릿 지정 섹션에서 템플릿 소스를 선택하고 템플릿 준비 완료를 선택하십시오. onboard-audit-account.yaml 파일을 업로드한 다음 요구 사항에 따라 나머지 옵션을 구성합니다.</p> <p>중요: 다음 입력 파라미터를 구성해야 합니다.</p> <ul style="list-style-type: none"> • DestinationEmailAddress — 조사 결과를 받을 이메일 주소를 입력합니다. 	<p>개발자, 보안 엔지니어</p>

작업	설명	필요한 기술
	<ul style="list-style-type: none"> • HTTPEndpoint — 티켓팅 또는 SIEM 도구에 대한 HTTP 엔드포인트를 제공합니다. <p>또한 AWS 명령줄 인터페이스 (AWS CLI) 를 사용하여 AWS CloudFormation 템플릿을 배포할 수 있습니다. 이에 대한 자세한 내용은 AWS CloudFormation 설명서의 스택 생성을 참조하십시오.</p>	
Amazon SNS 구독을 확인합니다.	이메일 받은 편지함을 확인하고 Amazon SNS로부터 받은 이메일에서 구독 확인을 선택합니다. 그러면 웹 브라우저 창이 열리고 구독 확인이 표시됩니다.	개발자, 보안 엔지니어

AWS CloudFormation 스택 세트를 생성하여 Amazon Inspector 스캔 일정을 자동화하십시오.

작업	설명	필요한 기술
감사 계정에서 스택 세트를 생성합니다.	<p>vulnerability-management-program.yaml 파일을(첨부됨)을 컴퓨터의 로컬 경로에 다운로드합니다.</p> <p>AWS CloudFormation 콘솔에서 [스택 세트 보기] 를 선택한 다음 [Create] 를 선택합니다. StackSet Choose 템플릿 준비 완료를 선택하고, 템플릿 파일 업로드를 선택한</p>	개발자, 보안 엔지니어

작업	설명	필요한 기술
	<p>후, vulnerability-management-program.yaml 파일을 업로딩합니다.</p> <p>self-managed 권한을 사용하려면 AWS CloudFormation 설명서의 자체 관리 권한으로 스택 세트 생성의 지침을 따르십시오. 이렇게 하면 개별 계정에 스택 세트가 생성됩니다.</p> <p>service-managed 권한을 사용하려면 AWS CloudFormation 설명서의 서비스 관리 권한으로 스택 세트 생성의 지침을 따르십시오. 이를 통해 스택 세트를 전체 조직 또는 지정된 조직 단위(OU) 안에 생성합니다.</p> <p>중요: 다음 입력 파라미터가 구성되었는지 확인합니다.</p> <ul style="list-style-type: none"> • AssessmentSchedule <ul style="list-style-type: none"> — cron 표현식 EventBridge 사용 일정. • Duration – Amazon Inspector 평가 실행 기간은 초 단위입니다. • CentralSNSTopicArn <ul style="list-style-type: none"> – 중앙 SNS 주제에 대한 Amazon Resource Name (ARN) • Tagkey — 리소스 그룹과 연결된 태그 키. 	

작업	설명	필요한 기술
	<ul style="list-style-type: none"> • Tagvalue — 리소스 그룹과 연결된 태그 값. <p>감사 계정에서 EC2 인스턴스를 스캔하려면 감사 계정에서 vulnerability-management-program.yaml 파일을 AWS CloudFormation 스택으로 실행해야 합니다.</p>	
<p>솔루션을 검증합니다.</p>	<p>Amazon Inspector에 대해 지정한 일정에 따라 이메일 또는 HTTP 엔드포인트를 통해 조사 결과를 수신하는지 확인합니다.</p>	<p>개발자, 보안 엔지니어</p>

관련 리소스

- [Amazon Inspector를 사용하여 보안 취약성 테스트를 확장합니다.](#)
- [Amazon Inspector 보안 조사 결과의 문제를 자동으로 해결합니다.](#)
- [Amazon EC2, Systems Manager, Amazon Inspector를 사용하여 보안 평가 설정을 간소화하는 방법](#)

첨부

이 문서와 관련된 추가 콘텐츠에 액세스하려면 [attachment.zip](#) 파일의 압축을 풉니다.

AWS Config에서 사용자 지정 수정 규칙을 CloudTrail 사용하여 AWS를 자동으로 다시 활성화합니다.

작성자: Manigandan Shri (AWS)

환경: 프로덕션

기술: 인프라, 운영, 보안, 자격 증명, 규정 준수

AWS 서비스: 아마존 S3, AWS Config, AWS KMS, AWS 자격 증명 및 액세스 관리, AWS Systems Manager, AWS CloudTrail

요약

Amazon Web Services(AWS) 계정의 활동에 대한 가시성은 중요한 보안 및 운영 모범 사례입니다. AWS는 계정의 거버넌스, 규정 준수, 운영 및 위험 감사를 CloudTrail 지원합니다.

계정에서 활성화된 CloudTrail 상태를 유지할 수 있도록 AWS Config는 *cloudtrail-enabled* 관리형 규칙을 제공합니다. CloudTrail 이 기능을 끄면 [자동](#) 수정을 사용하여 *cloudtrail-enabled* 규칙이 자동으로 다시 활성화됩니다.

하지만 자동 수정을 사용하는 CloudTrail 경우 [보안 모범 사례](#)를 반드시 준수해야 합니다. 이러한 모범 사례에는 모든 AWS CloudTrail 지역에서의 활성화, 읽기 및 쓰기 워크로드 로깅, 인사이트 활성화, AWS KMS ([AWS KMS](#)) 관리 키 (SSE-KMS) 를 사용한 서버 측 암호화로 로그 파일 암호화 등이 포함됩니다.

이 패턴은 계정에서 자동으로 다시 활성화되도록 사용자 지정 수정 조치를 제공함으로써 이러한 보안 모범 사례를 따르는 데 도움이 됩니다. CloudTrail

중요: 변조를 방지하려면 [서비스 제어 정책 \(SCP\)](#) 을 사용하는 것이 좋습니다. CloudTrail 이에 대한 자세한 내용은 AWS 보안 블로그에서 [AWS Organizations를 사용하여 대규모로 보안을 간소화하는 방법](#) 의 AWS 변조 방지 CloudTrail 섹션을 참조하십시오.

사전 조건 및 제한 사항

사전 조건

- 활성화 상태의 AWS 계정.

- AWS Systems Manager Automation 런북 생성 권한
- 사용자 계정의 기존 트레일

제한 사항

이 패턴은 다음 동작을 지원하지 않습니다.

- 스토리지 위치에 대한 Amazon Simple Storage Service(Amazon S3) 접두사 키 설정
- Amazon Simple Notification Service(SNS) 항목에 게시
- CloudWatch 로그를 모니터링하도록 Amazon CloudTrail Logs 구성

아키텍처

기술 스택

- AWS Config
- CloudTrail
- Systems Manager
- Systems Manager Automation

도구

- [AWS Config](#)는 사용자의 계정에서 AWS 리소스의 구성을 상세하게 볼 수 있도록 합니다.
- [CloudTrailAWS](#)는 계정의 거버넌스, 규정 준수, 운영 및 위험 감사를 가능하게 합니다.
- [AWS Key Management Service\(AWS KMS\)](#)는 암호화 및 키 관리 서비스입니다.
- [AWS Systems Manager](#)는 AWS에서 인프라를 확인하고 제어할 수 있도록 지원합니다.
- [AWS Systems Manager Automation](#)은 Amazon Elastic Compute Cloud(Amazon EC2) 인스턴스 및 기타 AWS 리소스의 일반적인 유지 관리 및 배포 작업을 간소화합니다.
- [Amazon Simple Storage Service\(S3\)](#)는 원하는 양의 데이터를 저장, 보호 및 검색하는 데 도움이 되는 클라우드 기반 객체 스토리지 서비스입니다.

코드

cloudtrail-remediation-action.yml 파일 (첨부) 을 사용하면 Systems Manager 자동화 런북을 생성하여 보안 모범 사례를 CloudTrail 사용하여 설정하고 다시 활성화할 수 있습니다.

에픽

구성: CloudTrail

작업	설명	필요한 기술
S3 버킷을 생성합니다.	AWS 관리 콘솔에 로그인하고 Amazon S3 콘솔을 연 다음 CloudTrail 로그를 저장할 S3 버킷을 생성합니다. 자세한 내용은 Amazon S3 설명서의 S3 버킷 생성 을 참조하세요.	시스템 관리자
로그 파일을 S3 CloudTrail 버킷으로 전송할 수 있도록 허용하는 버킷 정책을 추가합니다.	CloudTrail 로그 파일을 S3 버킷으로 전송하는 데 필요한 권한이 있어야 합니다. Amazon S3 콘솔에서 앞서 생성한 S3 버킷을 선택한 후에 권한을 선택합니다. CloudTrail 설명서의 Amazon S3 버킷 정책을 사용하여 S3 버킷 정책을 생성 합니다. CloudTrail S3 버킷에 정책을 추가하는 방법에 대한 단계는 Amazon S3 설명서의 Amazon S3 콘솔을 사용하여 버킷 정책 추가 를 참고하세요. 중요: 트레일을 생성할 때 접두사를 지정한 경우 해당 접두사를 S3 버킷 정책에 포함해야 합니다. CloudTrail 접두사는 S3 버킷 안에 폴더 같은 조직을 생성한 S3 객체 키에 선택적으로	시스템 관리자

작업	설명	필요한 기술
	<p>추가할 수 있습니다. 이에 대한 자세한 내용은 CloudTrail 설명서의 트레일 생성을 참조하십시오.</p>	
<p>KMS 키를 생성합니다.</p>	<p>객체를 S3 버킷에 추가하기 전에 객체를 암호화하는 CloudTrail 데 사용할 AWS KMS 키를 생성합니다. 이 이야기에 대한 도움이 필요하면 설명서에서 AWS KMS 관리 키 (SSE-KMS) 를 사용한 CloudTrail 로그 파일 암호화를 참조하십시오. CloudTrail</p>	<p>시스템 관리자</p>
<p>KMS 키에 키 정책을 추가합니다.</p>	<p>KMS 키 정책을 첨부하여 KMS 키 사용을 CloudTrail 허용하십시오. 이 이야기에 대한 도움이 필요하면 설명서에서 AWS KMS 관리 키 (SSE-KMS) 를 사용한 CloudTrail 로그 파일 암호화를 참조하십시오. CloudTrail</p> <p>중요: 권한이 필요하지 않습니다. CloudTrail Decrypt</p>	<p>시스템 관리자</p>
<p>Systems Manager AssumeRole 런북용으로 생성</p>	<p>런북을 실행하려면 Systems Manager Automation용 AssumeRole 을 생성합니다. 이에 대한 지침과 자세한 내용은 Systems Manager 설명서의 자동화 설정을 참조하세요.</p>	<p>시스템 관리자</p>

Systems Manager Automation 런북 생성 및 테스트

작업	설명	필요한 기술
Systems Manager Automation 런북을 생성합니다.	cloudtrail-remediation-action.yml 파일 (첨부됨)을 사용하여 Systems Manager Automation 런북을 생성합니다. 자세한 내용은 Systems Manager 설명서의 Systems Manager 문서 생성 을 참고하세요.	시스템 관리자
런북을 테스트합니다.	Systems Manager 콘솔에서 이전에 만든 Systems Manager Automation 런북을 테스트합니다. 이에 대한 자세한 내용은 Systems Manager 설명서의 간단한 자동화 실행 을 참조하세요.	시스템 관리자

AWS Config에서 자동 수정 규칙을 설정합니다.

작업	설명	필요한 기술
CloudTrail-enabled 규칙을 추가합니다.	AWS Config 콘솔에서 규칙을 선택한 다음 규칙 추가를 선택합니다. 규칙 추가 페이지에서 사용자 지정 규칙 추가를 선택합니다. 규칙 구성 페이지에서 이름과 설명을 입력하고 cloudtrail-enabled 규칙을 추가합니다. 자세한 내용은 AWS Config 설명서의 AWS Config 규칙 관리 를 참고하세요.	시스템 관리자

작업	설명	필요한 기술
자동 수정 작업을 추가합니다.	<p>작업 드롭다운 목록에서 수정 관리를 선택합니다. 자동 수정을 선택한 후에 이전에 만든 Systems Manager 런북을 선택합니다.</p> <p>에 필요한 입력 매개변수는 다음과 같습니다. CloudTrail</p> <ul style="list-style-type: none"> • CloudTrailName • CloudTrailS3Bucket Name • CloudTrailKmsKeyId • AssumeRole (선택 사항) <p>다음과 같은 입력 파라미터는 기본적으로 true로 설정됩니다.</p> <ul style="list-style-type: none"> • IsMultiRegionTrail • IsOrganizationTrail • IncludeGlobalServiceEvents • EnableLogFileValidation <p>속도 제한 파라미터 및 리소스 ID 파라미터의 기본값을 유지합니다. 저장을 선택합니다.</p> <p>자세한 내용은 AWS Config 설명서의 AWS Config 규칙을 사용하여 규정을 준수하지 않는</p>	시스템 관리자

작업	설명	필요한 기술
	AWS 리소스 문제 해결 을 참고하세요.	
<p>자동 수정 규칙을 테스트합니다.</p>	<p>자동 수정 규칙을 테스트하려면 CloudTrail 콘솔을 열고 Trails를 선택한 다음 트레일을 선택합니다. 트레일에 대한 로깅을 끄려면 로깅 중지를 선택합니다. 확인하라는 메시지가 표시되면 로깅 중지를 선택합니다. CloudTrail 해당 트레일에 대한 활동 기록을 중지합니다.</p> <p>AWS Config 설명서의 리소스 평가 지침에 따라 자동으로 다시 CloudTrail 활성화되었는지 확인하십시오.</p>	<p>시스템 관리자</p>

관련 리소스

구성: CloudTrail

- [S3 버킷 생성](#)
- [에 대한 Amazon S3 버킷 정책 CloudTrail](#)
- [Amazon S3 콘솔을 사용하여 버킷 정책 추가](#)
- [추적 생성](#)
- [자동화 설정](#)
- [AWS KMS 관리형 키 \(SSE-KMS\) 를 사용하여 CloudTrail 로그 파일 암호화](#)

AWS Systems Manager Automation 런북 생성

- [Systems Manager 문서 생성](#)
- [단순한 자동화 실행](#)

AWS Config에서 자동 수정 규칙 설정

- [AWS Config 규칙 관리](#)
- [AWS Config 규칙에 따른 미준수 AWS 리소스 문제 해결](#)

추가 리소스

- [AWS CloudTrail — 보안 모범 사례](#)
- [AWS Systems Manager 시작하기](#)
- [AWS Config 시작하기](#)
- [AWS 시작하기 CloudTrail](#)

첨부

이 문서와 관련된 추가 콘텐츠에 액세스하려면 [attachment.zip](#) 파일의 압축을 풉니다.

암호화되지 않은 Amazon RDS DB 인스턴스 및 클러스터를 자동으로 수정하기

작성자: Ajay Rawat (AWS) 및 Josh Joy (AWS)

환경: PoC 또는 파일럿

기술: 보안, 자격 증명, 규정 준수, 분석, 데이터베이스

AWS 서비스: AWS Config; AWS KMS, AWS Identity and Access Management, AWS Systems Manager, Amazon RDS

요약

이 패턴은 AWS Config, AWS Systems Manager 런북 및 AWS KMS(키 관리 서비스) 키를 사용하여 Amazon Web Services(AWS)에서 암호화되지 않은 Amazon Relational Database Service(Amazon RDS) DB 인스턴스 및 클러스터를 자동으로 수정하는 방법을 설명합니다.

암호화된 RDS DB 인스턴스는 기본 스토리지에 대한 무단 액세스로부터 데이터의 보안을 유지해 주며 계층의 데이터 보호를 제공합니다. AWS 클라우드에 배포된 애플리케이션의 데이터 보호를 강화하고 저장된 데이터 암호화를 위한 규정 준수 요구 사항을 만족하기 위해 Amazon RDS 암호화를 사용할 수 있습니다. RDS DB 인스턴스를 생성할 때 암호화를 사용하도록 설정할 수 있지만, 생성된 후에는 사용할 수 없습니다. 하지만 DB 인스턴스의 스냅샷을 만든 후 해당 스냅샷의 암호화된 사본을 만들어 암호화되지 않은 RDS DB 인스턴스에 암호화를 추가할 수 있습니다. 그런 다음 암호화된 스냅샷에서 DB 인스턴스를 복원하여 원본 DB 인스턴스의 암호화된 사본을 얻을 수 있습니다.

이 패턴은 AWS Config 규칙을 사용하여 RDS DB 인스턴스 및 클러스터를 평가합니다. 규정 미준수 Amazon RDS 리소스에서 수행할 작업을 정의하는 AWS Systems Manager 런북과 DB 스냅샷을 암호화할 AWS KMS 키를 사용하여 문제 해결을 적용합니다. 그런 다음 서비스 제어 정책(SCP)을 적용하여 암호화 없이 새로운 DB 인스턴스 및 클러스터를 생성하지 못하도록 합니다.

이 패턴의 코드는 [여기](#)에 제공됩니다.

사전 조건 및 제한 사항

사전 조건

- 활성 상태의 AWS 계정
- 이 패턴에 대한 [GitHub 소스 코드 저장소의](#) 파일이 컴퓨터에 다운로드됩니다.
- 암호화되지 않은 RDS DB 인스턴스 또는 클러스터
- RDS DB 인스턴스 및 클러스터를 암호화하기 위한 기존 AWS KMS 키
- KMS 키 리소스 정책을 업데이트하기 위한 액세스
- AWS 계정에서 활성화된 AWS Config(AWS 설명서의 [AWS Config 시작하기](#) 참고).

제한 사항

- RDS DB 인스턴스에 대한 암호화는 DB 인스턴스를 생성할 때에만 활성화할 수 있으며 DB 인스턴스가 생성된 후에는 불가능합니다.
- 암호화되지 않은 DB 인스턴스의 암호화된 읽기 전용 복제본이나 암호화된 DB 인스턴스의 암호화되지 않은 읽기 전용 복제본은 보유할 수 없습니다.
- 암호화되지 않은 백업 또는 스냅샷을 암호화된 DB 인스턴스로 복원할 수 없습니다.
- Amazon RDS 암호화는 대부분의 DB 인스턴스 클래스에서 사용 가능합니다. 예외 목록은 Amazon RDS 설명서의 [Amazon RDS 리소스 암호화](#)를 참고하십시오.
- 암호화된 스냅샷을 한 AWS 리전에서 다른 리전으로 복사하려면 대상 AWS 리전에 KMS 키를 지정해야 합니다. 이는 KMS 키가 생성된 AWS 리전에만 해당하기 때문입니다.
- 소스 스냅샷은 복사 프로세스 전체에서 암호화를 유지합니다. Amazon RDS는 봉투 암호화를 사용하여 복사 프로세스 중에 데이터를 보호합니다. 자세한 내용은 AWS KMS 설명서의 [봉투 암호화](#)를 참조하세요.
- 암호화된 DB 인스턴스의 암호화를 해제할 수 없습니다. 하지만 암호화된 DB 인스턴스에서 데이터를 내보내고 암호화되지 않은 DB 인스턴스로 해당 데이터를 가져올 수 있습니다.
- 더 이상 사용할 필요가 없다고 확신하는 경우에만 KMS 키를 삭제해야 합니다. 확실하지 않은 경우에는 삭제하는 대신 [KMS 키를 비활성화](#)하는 방법을 고려하세요. 비활성화한 KMS 키는 나중에 다시 사용해야 할 때 재활성화할 수 있지만 삭제한 KMS 키는 복구할 수 없습니다.
- 자동 백업을 보존하도록 선택하지 않으면 DB 인스턴스와 동일한 AWS 리전의 자동 백업이 삭제됩니다. DB 인스턴스를 삭제한 후에는 복구할 수 없습니다.
- 자동 백업은 삭제 시점에 DB 인스턴스에 설정된 보존 기간 동안 보존됩니다. 설정된 이 보존 기간은 최종 DB 스냅샷을 생성할지 여부와 상관없이 적용됩니다.
- 자동 수정이 활성화된 경우 이 솔루션은 동일한 KMS 키를 가진 모든 데이터베이스를 암호화합니다.

아키텍처

다음 다이어그램은 AWS CloudFormation 구현의 아키텍처를 보여줍니다. AWS Cloud Development Kit(AWS CDK)를 사용하여 이 패턴을 구현할 수도 있다는 점을 유의해 주십시오.

도구

도구

- [AWS는 AWS](#) 리소스를 자동으로 설정할 수 있도록 CloudFormation 도와줍니다. 이를 통해 템플릿 파일을 사용하여 리소스 모음을 단일 유닛으로 생성 및 구성할 수 있습니다.
- [AWS Cloud Development Kit\(AWS CDK\)](#)는 익숙한 프로그래밍 언어를 사용하여 클라우드 인프라를 코드로 정의하고 프로비저닝하기 위한 소프트웨어 개발 프레임워크입니다.

AWS 서비스 및 특성

- [AWS Config](#)는 AWS 리소스의 구성 및 다른 리소스와의 관계를 추적합니다. 또한, 해당 AWS 리소스의 규정 준수를 평가할 수 있습니다. 이 서비스는 원하는 구성과 비교하여 AWS 리소스를 평가하도록 구성할 수 있는 규칙을 사용합니다. 일반적인 규정 준수 시나리오에 대해 AWS Config 관리형 규칙 세트를 사용하거나 사용자 지정 시나리오에 대한 자체 규칙을 생성할 수 있습니다. AWS 리소스가 규정을 준수하지 않는 것으로 확인되면 AWS Systems Manager 런북을 통해 수정 조치를 지정하고 선택적으로 Amazon Simple Notification Service(Amazon SNS) 주제를 통해 알림을 보낼 수 있습니다. 즉, 수정 작업을 AWS Config 규칙과 연결하고 수동 개입 없이 규정을 준수하지 않는 리소스를 처리하는 것을 자동으로 실행하도록 선택할 수 있습니다. 자동 문제 해결 후에도 리소스가 계속 규정을 준수하지 않는 경우, 자동 문제 해결을 다시 시도하도록 규칙을 설정할 수 있습니다.
- [Amazon Relational Database Service\(RDS\)](#)를 사용하여 클라우드에서 더 쉽게 관계형 데이터베이스를 설정, 운영 및 확장할 수 있습니다. Amazon RDS의 기본 빌딩 블록은 AWS 클라우드에 있는 격리된 데이터베이스 환경인 DB 인스턴스입니다. Amazon RDS는 다양한 관계형 데이터베이스 사용 사례에 맞게 최적화된 [다양한 인스턴스 유형](#)을 제공합니다. 인스턴스 유형은 CPU, 메모리, 스토리지, 네트워킹 용량의 다양한 조합으로 구성되며 애플리케이션에 적합한 리소스 조합을 선택할 수 있는 유연성을 제공합니다. 각 인스턴스 유형에는 하나 이상의 인스턴스 크기가 포함되므로 대상 워크로드의 요구 사항에 맞게 리소스를 확장할 수 있습니다.
- [AWS Key Management Service\(AWS KMS\)](#)는 데이터를 암호화하는 AWS KMS 키를 쉽게 생성하고 제어할 수 있게 해주는 관리형 서비스입니다. KMS 키는 루트 키를 논리적으로 표현한 것입니다. KMS 키에는 키 ID, 생성 날짜, 설명 및 키 상태와 같은 메타데이터가 포함됩니다.

- [AWS Identity and Access Management\(IAM\)](#)는 사용자에 대한 인증 및 권한 부여를 제어함으로써 AWS 리소스에 대한 액세스를 안전하게 제어할 수 있습니다.
- [SCP\(서비스 제어 정책\)](#)는 조직의 모든 계정에 사용 가능한 최대 권한을 중앙에서 제어합니다. SCP를 사용하면 조직의 액세스 제어 지침에 따라 계정을 유지할 수 있습니다. SCP는 관리 계정의 사용자 또는 역할에 영향을 미치지 않습니다. 조직의 멤버 계정에만 영향을 줍니다. 정책이 계정에 미치는 영향을 철저히 검증하지 않았다면 SCP를 조직의 루트에 연결하지 않는 것을 적극 권장합니다. 대신 한 번에 하나씩, 또는 소량 단위로 계정을 옮길 수 있는 조직 단위(OU)를 만들어 사용자가 주요 서비스를 이용하지 못하는 일이 없게 하세요.

코드

이 패턴의 소스 코드와 템플릿은 [GitHub 리포지토리에서](#) 사용할 수 있습니다. 패턴은 두 가지 구현 옵션을 제공합니다. AWS CloudFormation 템플릿을 배포하여 RDS DB 인스턴스 및 클러스터를 암호화하는 수정 역할을 생성하거나 AWS CDK를 사용할 수 있습니다. 리포지토리에는 이 두 옵션에 대한 별도의 폴더가 있습니다.

Epic 섹션에서는 템플릿 배포 step-by-step 지침을 제공합니다. CloudFormation AWS CDK를 사용하려면 리포지토리의 README.md 파일에 있는 지침을 따르십시오. GitHub

모범 사례

- 저장 및 전송 중에 데이터 암호화를 활성화합니다.
- 모든 계정과 AWS 리전에서 AWS Config를 활성화합니다.
- 모든 리소스 유형에 대한 구성 변경 사항을 기록합니다.
- IAM 자격 증명을 정기적으로 순환합니다.
- 에 AWS Config 대한 태그 지정을 활용하여 리소스를 더욱 쉽게 관리, 검색 및 필터링할 수 있습니다.

에픽

IAM 수정 역할 및 AWS Systems Manager 런북 생성

작업	설명	필요한 기술
템플릿을 다운로드하십시오. CloudFormation	GitHub 저장소에서 unencrypted-to-encrypted-rds.templat	DevOps 엔지니어

작업	설명	필요한 기술
	<p>e.json 파일을 다운로드합니다.</p>	
<p>CloudFormation 스택을 만드세요.</p>	<ol style="list-style-type: none"> 1. AWS 관리 콘솔에 로그인하고 https://console.aws.amazon.com/cloudformation/ 에서 CloudFormation 콘솔을 엽니다. 2. unencrypted-to-encrypted-rds.template.json 템플릿을 실행하여 새 스택을 생성합니다. <p>템플릿 배포에 대한 자세한 내용은 AWS CloudFormation 설명서를 참조하십시오.</p>	<p>DevOps 엔지니어</p>
<p>CloudFormation 매개변수와 값을 검토하십시오.</p>	<ol style="list-style-type: none"> 1. 스택 세부 정보를 검토하고 환경 요구 사항에 따라 값을 업데이트합니다. 2. 스택 생성을 선택하여 템플릿을 배포합니다. 	<p>DevOps 엔지니어</p>
<p>리소스를 검토합니다.</p>	<p>스택이 생성되면 상태가 CREATE_COMPLETE로 바뀝니다. 콘솔에서 생성된 리소스 (IAM 역할, AWS Systems Manager 런북) 를 CloudFormation 검토하십시오.</p>	<p>DevOps 엔지니어</p>

AWS KMS 키 정책 업데이트

작업	설명	필요한 기술
KMS 키 정책을 업데이트합니다.	<ol style="list-style-type: none"> 키 별칭 alias/RDS EncryptionAtRestKMSAlias 가 존재하는지 확인합니다. 주요 정책 설명에는 IAM 수정 역할이 포함되어야 합니다. (이전 에픽에서 배포한 CloudFormation 템플릿으로 만든 리소스를 확인하세요.) 다음 주요 정책에서 굵게 표시된 부분을 계정 및 생성된 IAM 역할과 일치하도록 업데이트하십시오. <pre data-bbox="594 1031 1029 1835"> { "Sid": "Allow access through RDS for all principals in the account that are authorized to use RDS", "Effect": "Allow", "Principal": { "AWS": "arn:aws: iam:: <your-AWS- account-ID>:role/ <your-IAM-remediation- role>" }, "Action": ["kms:Encrypt", "kms:Decrypt", "kms:ReEn crypt*" </pre>	DevOps 엔지니어

작업	설명	필요한 기술
	<pre> "kms:GenerateDataKey*", "kms:CreateGrant", "kms:ListGrants", "kms:DescribeKey"], "Resource": "*", "Condition": { "StringEquals": { "kms:ViaService": "rds.us-east-1.amazonaws.com", "kms:CallerAccount": "<your-AWS-account-ID>" } } } } } </pre>	

규정을 준수하지 않는 리소스를 찾아 수정합니다.

작업	설명	필요한 기술
<p>규정을 준수하지 않는 리소스를 확인합니다.</p>	<ol style="list-style-type: none"> 1. 규정을 준수하지 않는 리소스 목록을 보려면 https://console.aws.amazon.com/config/에서 AWS Config 콘솔을 엽니다. 2. 탐색 창에서 규칙을 선택한 후 rds-storage-encrypted 규칙을 선택합니다. 	<p>DevOps 엔지니어</p>

작업	설명	필요한 기술
	<p>AWS Config 콘솔에 나열된 비준수 리소스는 클러스터가 아닌 인스턴스입니다. 수정 자동화는 인스턴스와 클러스터를 암호화하고 새로 암호화된 인스턴스 또는 새로 생성된 클러스터를 생성합니다. 그러나 동일한 클러스터에 속하는 여러 인스턴스에 동시에 업데이트를 적용하지 않도록 주의하십시오.</p> <p>RDS DB 인스턴스 또는 볼륨에 업데이트를 적용하기 전에 RDS DB 인스턴스를 사용하고 있지 않아야 합니다. 스냅샷을 생성하는 동안 쓰기 작업이 발생하지 않는지 확인하여 스냅샷에 원본 데이터가 포함되어 있는지 확인하십시오. 수정 작업이 실행되는 동안 유지 관리 기간을 적용하는 것을 고려해 보십시오.</p>	

작업	설명	필요한 기술
<p>규정을 준수하지 않는 리소스를 찾아 수정합니다.</p>	<ol style="list-style-type: none"> 1. 사용자가 준비를 마친 상태이고 유지 관리 기간이 적용되면 수정할 리소스를 선택한 다음 해결을 선택합니다. <p>이제 작업 상태 열에 작업 실행 대기 중으로 표시되어야 합니다.</p> <ol style="list-style-type: none"> 2. Systems Manager에서 수정 진행 상황과 상태를 확인합니다. https://console.aws.amazon.com/systems-manager/에서 AWS Systems Manager 콘솔을 엽니다. 탐색 창에서 자동화를 선택한 후, 해당 자동화의 실행 ID를 선택하면 자세한 내용을 볼 수 있습니다. 	<p>DevOps 엔지니어</p>
<p>RDS DB 인스턴스를 사용할 수 있는지 확인합니다.</p>	<p>자동화가 완료되면 새로 암호화된 RDS DB 인스턴스를 사용할 수 있게 됩니다. 암호화된 RDS DB 인스턴스에는 접두사 encrypted 뒤에 원래 이름이 붙습니다. 예를 들어 암호화되지 않은 RDS DB 인스턴스 이름이 database-1 인 경우 새로 암호화된 RDS DB 인스턴스는 encrypted-database-1 이 됩니다.</p>	<p>DevOps 엔지니어</p>

작업	설명	필요한 기술
암호화되지 않은 인스턴스를 종료합니다.	수정이 완료되고 새로 암호화된 리소스가 검증되면, 암호화되지 않은 인스턴스를 종료할 수 있습니다. 리소스를 종료하기 전에 새로 암호화된 리소스가 암호화되지 않은 리소스와 일치하는지 확인하십시오.	DevOps 엔지니어

SCP 집행

작업	설명	필요한 기술
SCP를 적용합니다.	SCP를 적용하여 향후에 암호화 없이 DB 인스턴스와 클러스터를 생성하는 것을 방지합니다. GitHub 리포지토리에 제공된 <code>rds_encrypted.json</code> 파일을 이 용도로 사용하고 AWS 설명서의 지침을 따르십시오.	보안 엔지니어

관련 리소스

참조

- [AWS Config 설정](#)
- [AWS Config 사용자 지정 규칙](#)
- [AWS KMS 개념](#)
- [AWS Systems Manager 문서](#)
- [서비스 제어 정책](#)

도구

- [AWS CloudFormation](#)
- [AWS Cloud Development Kit\(AWS CDK\)](#)

가이드 및 패턴

- [AWS Config에서 사용자 지정 수정 규칙을 CloudTrail 사용하여 AWS를 자동으로 다시 활성화합니다.](#)

추가 정보

FAQ

질문: AWS Config는 어떻게 작동합니까?

A. AWS Config를 켜면 먼저 계정에 있는 지원되는 AWS 리소스를 찾고, 각 리소스에 대한 [구성 항목](#)을 생성합니다. 또한 는 리소스의 구성이 변경되면 구성 항목을 생성하고, 구성 레코더를 시작할 때부터 리소스의 구성 항목에 대한 기록 내역을 유지합니다. 기본적 AWS Config 으로 는 해당 AWS Region 지원되는 모든 리소스에 대한 구성 항목을 만듭니다. 에서 지원되는 모든 리소스에 대한 구성 항목을 만들지 않도록 하려면, 추적하도록 할 리소스 유형을 지정하면 됩니다.

질문 AWS Config 및 AWS Config 규칙은 AWS Security Hub와 어떤 관련이 있습니까?

답변 AWS Security Hub는 보안 및 규정 준수 상태 관리를 서비스로 제공하는 보안 및 규정 준수 서비스입니다. AWS Config 및 AWS Config 규칙을 기본 메커니즘으로 사용하여 AWS 리소스의 구성을 평가합니다. AWS Config 규칙을 사용하여 리소스 구성을 직접 평가할 수도 있습니다. Config 규칙은 AWS Control Tower 및 AWS Firewall Manager와 같은 다른 AWS 서비스에서도 사용됩니다.

AWS Organizations 및 AWS Secrets Manager를 사용하여 대규모 IAM 사용자 액세스 키 자동 교체

작성자: Tracy Hickey(AWS), Gaurav Verma(AWS), Laura Seletos(AWS), Michael Davie(AWS), Arvind Patel(AWS)

환경: PoC 또는 파일럿

기술: 보안, 자격 증명, 규정 준수

AWS 서비스: AWS CloudFormation, 아마존 CloudWatch 이벤트, AWS ID 및 액세스 관리, AWS Lambda, AWS Organizations, 아마존 S3, 아마존 SES, AWS Secrets Manager

요약

중요: AWS는 액세스 키와 같은 장기 보안 인증을 갖는 IAM 사용자 대신 AWS Identity and Access Management(IAM) 역할을 사용할 것을 [모범 사례](#)로 권장합니다. 이 패턴에 나와 있는 접근 방식은 수명이 긴 AWS API 보안 인증이 필요한 레거시 구현에만 사용됩니다. 이러한 구현의 경우에도 [Amazon Elastic Compute Cloud\(Amazon EC2\) 인스턴스 프로파일](#) 또는 [IAM Roles Anywhere](#)와 같은 단기 보안 인증을 사용하는 옵션을 고려하는 것이 좋습니다. 이 문서의 접근 방식은 단기 보안 인증을 즉시 사용하도록 변경할 수 없으며 일정에 따라 장기 보안 인증을 교체해야 하는 경우에만 해당됩니다. 이 접근 방식을 사용할 때는 교체된 API 보안 인증을 사용하도록 레거시 애플리케이션 코드나 구성을 정기적으로 업데이트해야 합니다.

[액세스 키](#)는 IAM 사용자에게 대한 장기 보안 인증입니다. IAM 보안 인증을 정기적으로 교체하면 손상된 IAM 액세스 키 세트가 AWS 계정의 구성 요소에 액세스하는 것을 방지할 수 있습니다. IAM 보안 인증을 교체하는 것도 [IAM의 보안 모범 사례](#)에서 중요한 부분입니다.

이 패턴은 IAM 키 [교체 리포지토리](#)에서 제공되는 [AWS CloudFormation 템플릿을 사용하여 GitHub IAM](#) 액세스 키를 자동으로 교체하는 데 도움이 됩니다.

이 패턴은 단일 계정 또는 여러 계정에서의 배포를 지원합니다. AWS Organizations를 사용하는 경우 이 솔루션은 조직 내 모든 AWS 계정 ID를 식별하여 계정이 제거되거나 새 계정이 생성되면 동적으로

규모를 조정합니다. 중앙 집중식 AWS Lambda 함수는 위임된 IAM 역할을 사용하여 선택한 여러 계정에서 교체 기능을 로컬로 실행합니다.

- 기존 액세스 키가 90일이 경과하면 새 IAM 액세스 키가 생성됩니다.
- 새 액세스 키는 AWS Secrets Manager에 보안 암호로 저장됩니다. 리소스 기반 정책은 지정된 [IAM 보안 주체](#)만 보안 암호에 액세스하고 검색할 수 있도록 허용합니다. 관리 계정에 키를 저장하기로 선택하면 모든 계정의 키가 관리 계정에 저장됩니다.
- 새 액세스 키가 생성된 AWS 계정 소유자에게 할당된 이메일 주소로 알림이 전송됩니다.
- 이전 액세스 키는 100일이 경과하면 비활성화되며 110일이 경과하면 삭제됩니다.
- AWS 계정 소유자에게 중앙 집중식 이메일 알림이 전송됩니다.

Lambda 함수와 CloudWatch Amazon은 이러한 작업을 자동으로 수행합니다. 그런 다음 새 액세스 키 페어를 검색하여 코드나 애플리케이션에서 바꿀 수 있습니다. 교체, 삭제 및 비활성화 기간을 사용자 지정할 수 있습니다.

사전 조건 및 제한 사항

- 활성 상태의 AWS 계정 하나 이상.
- 구성 및 설정된 AWS Organizations([자습서](#) 참조).
- 관리 계정에서 AWS Organizations를 쿼리할 수 있는 권한. 지침은 AWS Organizations 설명서의 [서비스 제어 정책 연결 및 분리](#)를 참조하세요.
- AWS CloudFormation 템플릿 및 관련 리소스를 시작할 권한이 있는 IAM 보안 주체. 자세한 내용은 AWS CloudFormation 설명서의 [자체 관리 권한 부여](#)를 참조하십시오.
- 리소스를 배포하기 위한 기존 Amazon Simple Storage Service(S3) 버킷.
- Amazon Simple Email Service(Amazon SES)는 샌드박스에서 벗어났습니다. 자세한 내용은 Amazon SES 설명서의 [Amazon SES 샌드박스 환경에서 나가기](#)를 참조하십시오..
- 가상 사설 클라우드 (VPC) 에서 Lambda를 실행하기로 선택한 경우 기본 템플릿을 실행하기 전에 다음 리소스를 생성해야 합니다. CloudFormation
 - VPC
 - 서브넷.
 - Amazon SES, AWS Systems Manager, AWS Security Token Service(AWS STS), Amazon S3, AWS Secrets Manager에 대한 엔드포인트. (GitHub [IAM 키 로테이션](#) 리포지토리에 제공된 엔드포인트 템플릿을 실행하여 이러한 엔드포인트를 생성할 수 있습니다.)

- AWS Systems Manager 파라미터(SSM 파라미터)에 저장된 Simple Mail Transfer Protocol(SMTP) 사용자 및 암호. 파라미터는 기본 CloudFormation 템플릿 파라미터와 일치해야 합니다.

아키텍처

기술 스택

- 아마존 CloudWatch
- 아마존 EventBridge
- IAM
- AWS Lambda
- AWS Organizations
- Amazon S3

아키텍처

다음 다이어그램은 이 패턴의 구성 요소 및 워크플로를 보여 줍니다. 이 솔루션은 보안 인증을 저장하는 두 가지 시나리오, 즉 멤버 계정과 관리 계정을 지원합니다.

옵션 1: 멤버 계정에 보안 인증 저장

옵션 2: 관리 계정에 보안 인증 저장

이 다이어그램은 다음 워크플로를 보여 줍니다.

1. EventBridge 이벤트는 `account_inventory` 24시간마다 Lambda 함수를 시작합니다.
2. 이 Lambda 함수는 모든 AWS 계정 ID, 계정 이름 및 계정 이메일 목록을 AWS Organizations에 쿼리합니다.
3. `account_inventory` Lambda 함수는 각 AWS 계정 ID에 대해 `access_key_auto_rotation` Lambda 함수를 시작하고 추가 처리를 위해 해당 함수에 메타데이터를 전달합니다.
4. `access_key_auto_rotation` Lambda 함수는 위임된 IAM 역할을 사용하여 AWS 계정 ID에 액세스합니다. Lambda 스크립트는 계정의 모든 사용자 및 IAM 액세스 키에 대해 감사를 실행합니다.
5. IAM 액세스 키의 수명이 모범 사례 임계값을 초과하지 않은 경우 Lambda 함수는 추가 작업을 수행하지 않습니다.

6. IAM 액세스 키의 수명이 모범 사례 임계값을 초과한 경우 `access_key_auto_rotation` Lambda 함수는 수행할 교체 작업을 결정합니다.
7. 작업이 필요한 경우 `access_key_auto_rotation` Lambda 함수는 새 키가 생성되면 AWS Secrets Manager에서 보안 암호를 생성하고 업데이트합니다. 지정된 IAM 보안 주체만 보안 암호에 액세스하고 검색할 수 있도록 허용하는 리소스 기반 정책도 생성됩니다. 옵션 1의 경우 보안 인증은 해당 계정의 Secrets Manager에 저장됩니다. 옵션 2의 경우(`StoreSecretsInCentralAccount` 플러그가 참으로 설정된 경우) 보안 인증은 관리 계정의 Secrets Manager에 저장됩니다.
8. 계정 소유자에게 교체 활동을 알리기 위해 `notifier` Lambda 함수가 시작됩니다. 이 함수는 AWS 계정 ID, 계정 이름, 계정 이메일 및 수행된 교체 작업을 수신합니다.
9. `notifier` Lambda 함수는 이메일 템플릿에 대한 배포 S3 버킷을 쿼리하고 관련 활동 메타데이터로 버킷을 동적으로 업데이트합니다. 그러면 이메일이 계정 소유자의 이메일 주소로 전송됩니다.

참고:

- 이 솔루션은 여러 가용 영역에서 복원력을 지원합니다. 하지만 여러 AWS 리전에서 복원력은 지원하지 않습니다. 여러 지역에서 지원하려면 두 번째 지역에 솔루션을 배포하고 키 순환 EventBridge 규칙을 비활성화한 상태로 유지할 수 있습니다. 그런 다음 두 번째 리전에서 솔루션을 실행하려는 경우 규칙을 활성화할 수 있습니다.
- 감사 모드에서 이 솔루션을 실행할 수 있습니다. 감사 모드에서는 IAM 액세스 키가 수정되지 않지만 사용자에게 알리기 위해 이메일이 전송됩니다. 감사 모드에서 솔루션을 실행하려면 키 교체 템플릿을 실행할 때 또는 `access_key_auto_rotation` Lambda 함수의 환경 변수에서 `DryRunFlag` 플러그를 참으로 설정하십시오.

자동화 및 규모 조정

이 솔루션을 자동화하는 CloudFormation 템플릿은 GitHub [IAM 키 로테이션](#) 리포지토리에서 제공되며 코드 섹션에 나열되어 있습니다. AWS Organizations에서는 솔루션을 각 멤버 계정에 개별적으로 [CloudFormation StackSets](#) 배포하는 대신 여러 계정에 `ASA-iam-key-auto-rotation-iam-assumed-roles.yaml` CloudFormation 템플릿을 배포하는 데 사용할 수 있습니다.

도구

서비스

- [Amazon](#)은 AWS 리소스의 지표와 AWS에서 실행하는 애플리케이션을 실시간으로 모니터링할 수 있도록 CloudWatch 도와줍니다.

- [AWS Identity and Access Management\(IAM\)](#)를 사용하면 사용자에 대해 인증 및 권한 부여를 제어함으로써 AWS 리소스에 대한 액세스를 안전하게 관리할 수 있습니다.
- [AWS Lambda](#)는 서버를 프로비저닝하거나 관리할 필요 없이 코드를 실행하는 데 도움이 되는 컴퓨팅 서비스입니다. 필요할 때만 코드를 실행하며 자동으로 확장이 가능하므로 사용한 컴퓨팅 시간만큼만 비용을 지불합니다.
- [AWS Organizations](#)는 사용자가 생성하고 중앙에서 관리하는 조직으로 여러 AWS 계정을 통합할 수 있는 계정 관리 서비스입니다.
- [AWS Secrets Manager](#)를 사용하면 코드에 하드코딩된 보안 인증 정보(암호 등)를 Secrets Manager에 대한 API 직접 호출로 바꾸어 프로그래밍 방식으로 보안 암호를 검색할 수 있습니다.
- [Amazon Simple Storage Service\(S3\)](#)는 원하는 양의 데이터를 저장, 보호 및 검색할 수 있는 클라우드 기반 객체 스토리지 서비스입니다.
- [Amazon Simple Email Service\(Amazon SES\)](#)를 사용하면 자신의 이메일 주소와 도메인을 사용하여 이메일을 보내고 받을 수 있습니다.
- [Amazon Simple Notification Service\(Amazon SNS\)](#)를 사용하면 웹 서버 및 이메일 주소를 포함하여 게시자와 클라이언트 간의 메시지 교환을 조정하고 관리할 수 있습니다.
- [Amazon Virtual Private Cloud\(VPC\)](#)를 이용하면 사용자가 정의한 가상 네트워크로 AWS 리소스를 시작할 수 있습니다. 이 가상 네트워크는 사용자의 자체 데이터 센터에서 운영하는 기존 네트워크와 유사하며 AWS의 확장 가능한 인프라를 사용한다는 이점이 있습니다.
- [Amazon VPC 엔드포인트](#)는 많은 AWS 서비스를 포함하여 AWS PrivateLink 기반 서비스에 연결할 수 있는 인터페이스를 제공합니다. VPC에서 지정하는 각 서브넷에 대해 엔드포인트 네트워크 인터페이스가 서브넷에 생성되며 서브넷 주소 범위의 프라이빗 IP 주소가 할당됩니다.

코드

필수 AWS CloudFormation 템플릿, Python 스크립트 및 Runbook 설명서는 GitHub [IAM 키 로테이션](#) 리포지토리에서 제공됩니다. 템플릿은 다음과 같이 배포됩니다.

템플릿	배포 위치	참고
ASA-iam-key-auto-rotation-and-notifier-solution.yaml	배포 계정	솔루션의 기본 템플릿입니다.

ASA-iam-key-auto-rotation-iam-assumed-roles.yaml	보안 인증을 교체하려는 단일 또는 여러 멤버 계정	CloudFormation 스택 세트를 사용하여 이 템플릿을 여러 계정에 배포할 수 있습니다.
ASA-iam-key-auto-rotation-list-accounts-role.yaml	중앙/관리 계정	이 템플릿을 사용하여 AWS Organizations에 계정 인벤토리를 보관합니다.
ASA-iam-key-auto-rotation-vpc-endpoints.yaml	배포 계정	VPC에서 Lambda 함수를 실행하려는 경우에만 이 템플릿을 사용하여 엔드포인트 생성을 자동화합니다(기본 템플릿에서 RunLambdaInVPC 파라미터를 참으로 설정).

에픽

솔루션 설정

작업	설명	필요한 기술
배포 S3 버킷을 선택합니다.	계정의 AWS Management Console에 로그인하여 Amazon S3 콘솔 을 연 다음 배포할 S3 버킷을 선택합니다. AWS Organizations에서 여러 계정을 위한 솔루션을 구현하려면 조직의 관리 계정으로 로그인합니다.	클라우드 아키텍트
리포지토리를 복제합니다.	GitHub IAM 키 순환 리포지토리를 로컬 데스크톱에 복제합니다.	클라우드 아키텍트
파일을 S3 버킷에 업로드합니다.	복제한 파일을 S3 버킷에 업로드합니다. asa/asa-iam-rotation 기본 폴더 구조를	클라우드 아키텍트

작업	설명	필요한 기술
	<p>사용하여 복제한 모든 파일 및 디렉터리를 복사하여 붙여넣습니다.</p> <p>참고: CloudFormation 템플릿에서 이 폴더 구조를 사용자 지정할 수 있습니다.</p>	
이메일 템플릿을 수정합니다.	<p>요구 사항에 따라 iam-auto-key-rotation-enforcement.html 이메일 템플릿(template 폴더에 위치)을 수정합니다. 템플릿 끝의 [Department Name Here]를 부서 이름으로 바꿉니다.</p>	클라우드 아키텍트

솔루션 배포

작업	설명	필요한 기술
키 로테이션을 위한 CloudFormation 템플릿을 실행합니다.	<ol style="list-style-type: none"> 배포 계정에서 ASA-iam-key-auto-rotation-and-notifier-solution.yaml 템플릿을 실행합니다. 자세한 내용은 CloudFormation 설명서의 스택 템플릿 선택을 참조하십시오. 다음 사항을 포함한 파라미터 값을 지정합니다. <ul style="list-style-type: none"> CloudFormation S3 버킷 이름 (S3BucketName) - 	클라우드 아키텍트

작업	설명	필요한 기술
	<p>Lambda 코드가 포함된 배포 S3 버킷의 이름입니다.</p> <ul style="list-style-type: none"> • CloudFormation S3 버킷 접두사 (S3BucketPrefix) — S3 버킷의 접두사입니다. • 위임된 IAM 역할 이름(IAMRoleName)-key-rotation Lambda 함수가 키를 교체하기 위해 수임할 역할 이름입니다. • IAM 실행 역할 이름(ExecutionRoleName)-key-rotation Lambda 함수에서 사용하는 IAM 실행 역할의 이름입니다. • 인벤토리 실행 역할 이름(InventoryExecutionRoleName)-account_inventory Lambda 함수에서 사용하는 IAM 실행 역할의 이름입니다. • 모의 실행 플래그(감사 모드)(DryRunFlag)-감사 모드를 켜려면 참으로 설정합니다(기본값). 적용 모드를 켜려면 거짓으로 설정합니다. • 조직 계정을 나열할 계정(OrgListAccount)- 	

작업	설명	필요한 기술
	<p>조직의 계정을 나열하는 데 사용할 중앙/관리 계정의 계정 ID입니다.</p> <ul style="list-style-type: none"> • 나열 계정 역할 이름(OrgListRole)-조직의 계정을 나열하는 데 사용할 역할 이름입니다. • 중앙 계정의 보안 암호 저장소 플래그(StoreSecretsInCentralAccount)-중앙 계정에 보안 암호를 저장하려면 참으로 설정합니다. 각 계정에 보안 암호를 저장하려면 거짓으로 설정합니다. • 보안 인증을 복제할 리전(CredentialRegion)-보안 인증을 복제하려는 AWS 리전으로(Secrets Manager) 심포로 구분합니다(예: us-east-2, us-west-1, us-west-2). 스택을 생성하는 리전은 건너뜁니다. • VPC에서 Lambda 실행(RunLambdaInVpc)-지정된 VPC에서 Lambda 함수를 실행하려면 참으로 설정합니다. VPC 엔드포인트를 생성하고 Lambda 함수를 포함하는 	

작업	설명	필요한 기술
	<p>서브넷에 NAT 게이트웨이를 연결해야 합니다. 자세한 내용은 이 옵션을 다루는 re:Post 문서를 참조하십시오.</p> <ul style="list-style-type: none"> • Lambda 함수의 VPC ID(<code>VpcId</code>), 보안 그룹 규칙의 VPC CIDR(<code>VpcCidr</code>) 및 Lambda 함수의 서브넷 ID(<code>SubnetId</code>)-<code>RunLambdaInVpc</code> 를 참으로 설정하면 VPC, CIDR 및 서브넷에 대한 정보를 제공합니다. • 관리자 이메일 주소(<code>AdminEmailAddress</code>)-알림을 보낼 유효한 이메일 주소입니다. • AWS Organizations ID(<code>AWSOrgID</code>)-조직의 고유 ID입니다. 이 ID는 o-로 시작하며 그 뒤에 10~32개의 소문자 또는 숫자가 옵니다. • 이메일 템플릿 파일 이름 [감사 모드](<code>EmailTemplateAudit</code>) 및 [적용 모드](<code>EmailTemplateEnforce</code>)-감사 모드 및 적용 모드에 대해 <code>notifier</code> 모듈에서 전송 	

작업	설명	필요한 기술
	<p>하는 이메일 HTML 템플릿의 파일 이름입니다.</p> <ul style="list-style-type: none"> SMTP 사용자 SSM 파라미터 이름(SMTPUserName) 및 SMTP 암호 SSM 파라미터 이름(SMTPPasswordParamName) - Simple Mail Transfer Protocol(SMTP)에 대한 사용자 및 암호 정보입니다. 	

작업	설명	필요한 기술
<p>수입된 역할을 위한 CloudFormation 템플릿을 실행합니다.</p>	<ol style="list-style-type: none"> 1. AWS CloudFormation 콘솔에서 키를 교체하려는 각 계정의 ASA-iam-key-auto-rotation-iam-assumed-roles.yaml 템플릿을 실행합니다. 계정이 두 개 이상인 경우 관리 계정의 기본 CloudFormation 템플릿을 스택으로 배포하고 CloudFormation 스택 세트가 포함된 ASA-iam-key-auto-rotation-iam-assumed-roles.yaml 템플릿을 필요한 모든 계정에 배포할 수 있습니다. 자세한 내용은 CloudFormation 설명서의 AWS CloudFormation StackSets 사용을 참조하십시오. 2. 다음 파라미터의 값을 지정합니다. <ul style="list-style-type: none"> • 위임된 IAM 역할 이름(IAMRoleName)-Lambda access_key_auto_rotation 합수가 수입할 IAM 역할 이름입니다. 기본값을 유지할 수 있습니다. • IAM 실행 역할 이름(ExecutionRoleName)-Lambda 함수를 실행하는 하위 계정 	<p>클라우드 아키텍트</p>

작업	설명	필요한 기술
	<p>역할을 수입할 IAM 역할입니다.</p> <ul style="list-style-type: none"> • 기본 AWS 계정 ID(PrimaryAccountID)-기본 템플릿을 배포할 AWS 계정 ID입니다. • IAM 면제 그룹(IAMExemptionGroup)-자동 키 교체에서 제외하려는 IAM 계정을 지원하는 데 사용되는 IAM 그룹 이름입니다. 	

작업	설명	필요한 기술
계정 인벤토리 CloudFormation 템플릿을 실행하십시오.	<ol style="list-style-type: none"> 1. 관리/중앙 계정에서 ASA-iam-key-auto-rotation-list-accounts-role.yaml 템플릿 실행 2. 다음 파라미터의 값을 지정합니다. <ul style="list-style-type: none"> • 위임된 IAM 역할 이름(IAMRoleName)- Lambda access_key_auto_rotation 함수가 수임할 IAM 역할 이름입니다. • Lambda 계정의 IAM 실행 역할 이름(AccountExecutionRoleName)- Lambda notifier 함수가 수임할 IAM 역할의 이름입니다. • Lambda 교체의 IAM 실행 역할 이름(RotationExecutionRoleName)- Lambda access_key_auto_rotation 함수가 수임할 IAM 역할의 이름입니다. • 기본 AWS 계정 ID(PrimaryAccountID)-기본 템플릿을 배포할 AWS 계정 ID입니다. 	클라우드 아키텍트

작업	설명	필요한 기술
VPC CloudFormation 엔드포인트용 템플릿을 실행합니다.	<p>이 작업은 선택 사항입니다.</p> <ol style="list-style-type: none"> 1. 배포 계정에서 <code>ASA-iam-key-auto-rotation-vpc-endpoints.yaml</code> 템플릿을 실행합니다. 2. 다음 파라미터의 값을 지정합니다. <ul style="list-style-type: none"> • VPC ID(<code>pVpcId</code>), 서브넷 ID(<code>pSubnetId</code>) 및 VPC의 CIDR 범위(<code>pVPCcidr</code>)-VPC, CIDR 및 서브넷에 대한 정보를 제공합니다. • 각 VPC 엔드포인트의 파라미터를 참으로 설정합니다. 이미 엔드포인트가 있는 경우 거짓을 선택할 수 있습니다. 	클라우드 아키텍트

관련 리소스

- [IAM의 보안 모범 사례](#)(IAM 설명서)
- [AWS Organizations 및 서비스 연결 역할](#)(AWS Organizations 설명서)
- [스택 템플릿 선택 \(설명서\)](#) CloudFormation
- [AWS 사용 CloudFormation StackSets](#) (CloudFormation 설명서)

IAM 액세스 분석기 및 AWS 매크로를 사용하여 CodePipeline AWS 계정의 IAM 정책 및 역할을 자동으로 검증하고 배포합니다. CloudFormation

작성자: Helton Henrique Ribeiro(AWS) 및 Guilherme Simoes(AWS)

코드 리포지토리: [IAM 역할 파이프라인](#)

환경: PoC 또는 파일럿

기술: 보안, ID, 규정 준수
DevOps

AWS 서비스: AWS, AWS
CloudFormation, AWS
CodeBuild, AWS CodeCommit
CodePipeline, AWS Lambda,
AWS SAM

요약

이 패턴은 단계를 설명하고 개발 팀이 Amazon Web Services(AWS) 계정에서 AWS Identity and Access Management(IAM) 정책 및 역할을 생성할 수 있도록 배포 파이프라인을 생성하는 코드를 제공합니다. 이 접근 방식은 조직이 운영 팀의 오버헤드를 줄이고 배포 프로세스를 가속화하는 데 도움이 됩니다. 또한 개발자가 기존 거버넌스 및 보안 제어와 호환되는 IAM 역할 및 정책을 만들 수 있습니다.

이 패턴의 접근 방식은 [AWS Identity 및 Access Management Access Analyzer](#)를 사용하여 IAM 역할에 연결하려는 IAM 정책을 검증하고 CloudFormation AWS를 사용하여 IAM 역할을 배포합니다. 하지만 개발 팀은 AWS CloudFormation 템플릿 파일을 직접 편집하는 대신 JSON 형식의 IAM 정책 및 역할을 생성합니다. AWS CloudFormation 매크로는 배포를 시작하기 전에 이러한 JSON 형식의 정책 파일을 AWS CloudFormation IAM 리소스 유형으로 변환합니다.

배포 파이프라인(RolesPipeline)에는 소스, 검증 및 배포 스테이지가 있습니다. 소스 단계에서 개발 팀은 IAM 역할 및 정책 정의가 포함된 JSON 파일을 AWS 리포지토리로 푸시합니다. CodeCommit CodeBuild 그런 다음 AWS는 스크립트를 실행하여 해당 파일을 검증하고 Amazon Simple Storage Service (Amazon S3) 버킷에 복사합니다. 개발 팀은 별도의 S3 버킷에 저장된 AWS CloudFormation 템플릿 파일에 직접 액세스할 수 없으므로 JSON 파일 생성 및 검증 프로세스를 따라야 합니다.

마지막으로, 배포 단계에서 CodeDeploy AWS는 AWS CloudFormation 스택을 사용하여 계정의 IAM 정책 및 역할을 업데이트하거나 삭제합니다.

중요: 이 패턴의 워크플로는 개념 증명(POC)이므로 테스트 환경에서만 사용하는 것이 좋습니다. 프로덕션 환경에서 이 패턴의 접근 방식을 사용하려면 IAM 설명서에서 [IAM의 보안 모범 사례](#)를 참조하고 IAM 역할 및 AWS 서비스에 필요한 사항을 변경하십시오.

사전 조건 및 제한 사항

사전 조건

- 활성 상태의 AWS 계정.
- RolesPipeline 파이프라인의 새 S3 버킷 또는 기존 S3 버킷. 사용 중인 액세스 보안 인증에 이 버킷에 객체를 업로드할 권한이 있는지 확인하십시오.
- AWS Command Line Interface(AWS CLI), 설치 및 구성됨. 이에 관한 자세한 내용은 AWS CLI 설명서의 [AWS CLI 설치, 업데이트 및 제거](#)를 참조하십시오.
- AWS Serverless Application Model(AWS SAM) CLI, 설치 및 구성됨. 이에 관한 자세한 내용은 AWS SAM 설명서의 [AWS SAM CLI 설치](#)를 참조하십시오.
- Python 3 로컬 머신에 설치. 자세한 내용은 [Python 설명서](#)를 참조하십시오.
- Git 클라이언트, 설치 및 구성됨.
- 로컬 시스템에 복제된 GitHub IAM roles pipeline 리포지토리.
- 기존 JSON 형식의 IAM 정책 및 역할. 이에 대한 자세한 내용은 Github IAM roles pipeline 리포지토리의 [ReadMe](#) 파일을 참조하십시오.
- 개발자 팀은 이 솔루션의 AWS CodePipeline CodeBuild, 및 CodeDeploy 리소스를 편집할 권한이 없어야 합니다.

제한 사항

- 이 패턴의 워크플로는 개념 증명(POC)이므로 테스트 환경에서만 사용하는 것이 좋습니다. 프로덕션 환경에서 이 패턴의 접근 방식을 사용하려면 IAM 설명서에서 [IAM의 보안 모범 사례](#)를 참조하고 IAM 역할 및 AWS 서비스에 필요한 사항을 변경하십시오.

아키텍처

다음 다이어그램은 IAM Access Analyzer 및 AWS 매크로를 사용하여 CodePipeline IAM 역할 및 정책을 자동으로 검증하고 계정에 배포하는 방법을 보여줍니다. CloudFormation

이 다이어그램은 다음 워크플로를 보여줍니다.

1. 개발자는 IAM 정책 및 역할에 대한 정의가 포함된 JSON 파일을 작성합니다. 개발자는 CodeCommit 리포지토리에 코드를 푸시한 다음 파이프라인을 시작합니다. CodePipeline RolesPipeline
2. CodeBuild IAM 액세스 분석기를 사용하여 JSON 파일을 검증합니다. 보안 또는 오류 관련 발견이 발견되면 배포 프로세스가 중지됩니다.
3. 보안 또는 오류 관련 발견이 없는 경우 JSON 파일이 RolesBucket S3 버킷으로 전송됩니다.
4. AWS Lambda 함수로 구현된 AWS CloudFormation 매크로는 버킷에서 RolesBucket JSON 파일을 읽고 이를 AWS IAM 리소스 유형으로 변환합니다. CloudFormation
5. 사전 정의된 AWS CloudFormation 스택은 계정에서 IAM 정책 및 역할을 설치, 업데이트 또는 삭제합니다.

자동화 및 규모 조정

이 패턴을 자동으로 배포하는 AWS CloudFormation 템플릿은 GitHub [IAM 역할 파이프라인](#) 리포지토리에서 제공됩니다.

도구

- [AWS Command Line Interface\(AWS CLI\)](#)는 명령줄 셸에서 명령을 사용하여 AWS 서비스와 상호 작용할 수 있는 오픈 소스 도구입니다.
- [AWS Identity and Access Management\(IAM\)](#)은 누구에게 인증 및 사용 권한이 있는지 제어하여 AWS 리소스에 대한 액세스를 안전하게 관리할 수 있도록 도와줍니다.
- [IAM Access Analyzer](#)를 사용하면 외부 엔터티와 공유되는 조직 및 계정 내 리소스(예: S3 버킷 또는 IAM 역할)를 식별할 수 있습니다. 이를 통해 리소스 및 데이터에 대한 의도치 않은 액세스를 식별할 수 있습니다.
- [AWS Serverless Application Model\(AWS SAM\)](#)은 AWS 클라우드에서 서버리스 애플리케이션을 빌드하는 데 사용할 수 있는 오픈 소스 프레임워크입니다.

code

이 패턴의 소스 코드와 템플릿은 GitHub [IAM 역할 파이프라인](#) 리포지토리에서 제공됩니다.

에픽

리포지토리 복제

작업	설명	필요한 기술
샘플 리포지토리를 복제합니다.	GitHub IAM 역할 파이프라인 리포지토리를 로컬 시스템에 복제합니다.	앱 개발자, 일반 AWS

파이프라인을 배포하세요. RolesPipeline

작업	설명	필요한 기술
파이프라인을 배포합니다.	<ol style="list-style-type: none"> 복제된 리포지토리가 들어 있는 디렉터리로 이동합니다. make deploy bucket=<bucket_name> 명령을 실행합니다. 중요: 기존 S3 버킷의 버킷 이름으로 <bucket_name> 를 바꿔야 합니다. aws codepipeline get-pipeline -name RolesPipeline 명령을 실행하여 배포가 성공했는지 확인합니다. 	앱 개발자, 일반 AWS
파이프라인의 리포지토리를 복제합니다.	<ol style="list-style-type: none"> RolesPipeline AWS CloudFormation 스택은 roles-pipeline-repo CodeCommit 리포지토리를 생성합니다. AWS Management 콘솔에 로그인하고 AWS 	앱 개발자, 일반 AWS

작업	설명	필요한 기술
	<p>CodeCommit 콘솔을 연 다음 CodeCommit 리포지토리의 URL을 복사하여 로컬 시스템에 복제합니다. 이에 대한 자세한 내용은 AWS CodeCommit 설명서의 AWS CodeCommit 리포지토리에 연결을 참조하십시오.</p>	

RolesPipeline 파이프라인 테스트

작업	설명	필요한 기술
<p>유효한 IAM 정책 및 역할로 RolesPipeline 파이프라인을 테스트합니다.</p>	<ol style="list-style-type: none"> 1. IAM 정책 및 역할에 대한 JSON 파일을 생성합니다. <code>role-example GitHub IAM roles pipeline</code> 리포지토리의 디렉터리에 있는 샘플을 사용할 수 있습니다. 2. 필요한 구성으로 IAM 정책 및 역할을 정의합니다. 중요: GitHub IAM <code>roles pipeline</code> 저장소의 ReadMe 파일에 설명된 형식을 따라야 합니다. 3. 수정 내용을 <code>roles-pipeline-repo</code> CodeCommit 리포지토리로 푸시하십시오. 4. RolesPipeline 파이프라인 구현을 확인합니다. 	<p>앱 개발자, 일반 AWS</p>

작업	설명	필요한 기술
	5. IAM 정책 및 역할이 계정에 올바르게 배포되었는지 확인하십시오. 6. IAM 정책 또는 역할과 관련된 권한 경계가 있는지 확인하십시오. 자세한 내용은 IAM 사용 설명서의 IAM 엔터티에 대한 권한 경계 를 참조하십시오.	
잘못된 IAM 정책 및 역할로 RolesPipeline 파이프라인을 테스트하십시오.	1. roles-pipeline-repo CodeCommit 리포지토리를 수정하고 잘못된 IAM 역할 또는 정책을 포함하세요. 예를 들어 존재하지 않는 작업이나 잘못된 IAM 정책 버전을 사용할 수 있습니다. 2. 파이프라인 구현을 확인합니다. IAM Access Analyzer는 잘못된 IAM 정책 또는 역할을 탐지할 경우 검증 스테이지에서 파이프라인을 중지합니다.	앱 개발자, 일반 AWS

리소스 정리

작업	설명	필요한 기술
정리를 준비하십시오.	S3 버킷을 비운 다음 destroy 명령을 실행합니다.	앱 개발자, 일반 AWS
RolesStack 스택을 삭제합니다.	1. RolesPipeline 파이프라인은 IAM 정책 및 역할을 배포하는 RolesStack	앱 개발자, 일반 AWS

작업	설명	필요한 기술
	<p>AWS CloudFormation 스택을 생성합니다. RolesPipeline 파이프라인을 삭제하기 전에 이 스택을 삭제해야 합니다.</p> <p>2. AWS Management 콘솔에 로그인하고 AWS 콘솔을 연 다음 CloudFormation RolesStack 스택을 선택하고 [Delete] 를 선택합니다.</p>	
RolesPipeline 스택을 삭제합니다.	RolesPipeline AWS CloudFormation 스택을 삭제하려면 Github IAM roles pipeline 리포지토리의 ReadMe 파일에 있는 지침을 따르십시오.	앱 개발자, 일반 AWS

관련 리소스

- [IAM Access Analyzer - 정책 검증](#)(AWS 뉴스 블로그)
- [AWS CloudFormation 매크로를 사용하여 템플릿에 대한 사용자 지정 처리 수행](#) (AWS CloudFormation 설명서)
- [Python을 사용한 Lambda 함수 구축](#)(AWS Lambda 설명서)

AWS Security Hub를 Jira 소프트웨어와 양방향으로 통합하기

작성자: Joaquin Manuel Rinaudo (AWS)

코드 리포지토리: Security Hub에서 JIRA로의 통합	환경: PoC 또는 파일럿	기술: 보안, 자격 증명, 규정 준수
워크로드: 기타 모든 워크로드	AWS 서비스: AWS 람다, AWS 보안 허브, 아마존 CloudWatch	

요약

이 솔루션은 AWS Security Hub와 Jira 간의 양방향 통합을 지원합니다. 이 솔루션을 사용하면 Security Hub 조사 결과에서 JIRA 티켓을 자동 및 수동으로 만들고 업데이트할 수 있습니다. 보안팀은 이 통합을 사용하여 조치가 필요한 심각한 보안 조사사 결과를 개발자 팀에 알릴 수 있습니다.

이 솔루션을 통해 다음을 수행할 수 있습니다.

- Jira에서 티켓을 자동으로 만들거나 업데이트하는 Security Hub 컨트롤을 선택합니다.
- Security Hub 콘솔에서 Security Hub 사용자 지정 작업을 사용하여 Jira에서 티켓을 수동으로 에스컬레이션합니다.
- AWS Organizations에 정의된 AWS 계정 태그를 기반으로 Jira에서 티켓을 자동으로 배정합니다. 이 태그가 정의되지 않은 경우 기본 담당자가 사용됩니다.
- Jira에서 오탐 또는 허용된 위협으로 표시된 Security Hub 조사 결과를 자동으로 숨깁니다.
- 관련 조사 결과가 Security Hub에 보관되면 Jira 티켓을 자동으로 종료합니다.
- Security Hub 조사사 결과가 다시 발생하면 Jira 티켓을 다시 엽니다.

Jira 워크플로우

이 솔루션은 개발자가 위협을 관리하고 문서화할 수 있는 사용자 지정 Jira 워크플로우를 사용합니다. 문제가 워크플로우를 통해 이동함에 따라 양방향 통합을 통해 Jira 티켓 상태 및 Security Hub 조사 결과가 두 서비스의 워크플로우 전반에서 동기화됩니다. [이 워크플로는 CC BY 4.0에 따라 라이선스를 받은 디니스 크루즈의 SecDevOps 리스크 워크플로우에서 파생된 것입니다.](#) 보안 팀 구성원만 티켓 상태를 변경할 수 있도록 Jira 워크플로 조건을 추가하는 것을 권장합니다.

이 솔루션으로 자동으로 생성되는 Jira 티켓의 예는 이 패턴의 [추가 정보](#) 섹션을 참고하십시오.

사전 조건 및 제한 사항

사전 조건

- 다중 계정 AWS 환경에 이 솔루션을 배포하려는 경우:
 - 다중 계정 환경은 활성화되어 있으며 AWS Organizations에서 관리합니다.
 - Security Hub는 AWS 계정에서 활성화되어 있습니다.
 - AWS Organizations에서 사용자가 Security Hub 관리자 계정을 지정했습니다.
 - 사용자는 AWS Organizations 관리 계정에 대해 AWSOrganizationsReadOnlyAccess 권한이 있는 계정 간 IAM 역할을 가지고 있습니다.
 - (선택 사항) 사용자가 AWS 계정에 SecurityContactID로 태그를 지정했습니다. 이 태그는 정의된 보안 연락처에 Jira 티켓을 할당하는 데 사용됩니다.
- 단일 AWS 계정 내에 이 솔루션을 배포하려는 경우:
 - 활성화 AWS; 계정이 있습니다.
 - Security Hub는 AWS 계정에서 활성화되어 있습니다.
- Jira 서버 인스턴스

중요: 이 솔루션은 Jira Cloud 사용을 지원합니다. 하지만, Jira Cloud는 XML 워크플로우 가져오기를 지원하지 않으므로 Jira에서 워크플로우를 수동으로 다시 만들어야 합니다.

- Jira의 관리자 권한
- 다음 Jira 토큰 중 하나:
 - Jira Enterprise의 경우 개인용 액세스 토큰(PAT). 자세한 내용은 [개인용 액세스 토큰 사용](#)(Atlassian 지원)을 참고하십시오.
 - Jira Cloud의 경우 Jira API 토큰입니다. 자세한 내용은 [API 토큰 관리](#)(Atlassian 지원)를 참고하십시오.

아키텍처

이 섹션에서는 개발자와 보안 엔지니어가 위험을 허용하거나 문제를 해결하기로 결정하는 경우와 같은 다양한 시나리오에서의 솔루션 아키텍처를 보여줍니다.

시나리오 1: 개발자가 문제를 해결합니다.

1. Security Hub가 [AWS 기본 보안 모범 사례 표준](#)과 같은 특정 보안 제어에 대한 조사 결과를 생성합니다.
2. 검색 결과 및 CreateJIRA 작업과 관련된 Amazon CloudWatch 이벤트는 AWS Lambda 함수를 시작합니다.
3. Lambda 함수는 구성 파일과 조사 결과의 GeneratorId 필드를 사용하여 조사 결과를 에스컬레이션해야 하는지 여부를 평가합니다.
4. Lambda 함수가 조사 결과를 에스컬레이션해야 한다고 판단하고, AWS 관리 계정의 AWS Organizations로부터 SecurityContactID 계정 태그를 가져옵니다. 이 ID는 개발자와 연결되며 Jira 티켓의 담당자 ID로 사용됩니다.
5. Lambda 함수는 AWS Secrets Manager에 저장된 보안 인증 정보를 사용하여 Jira에서 티켓을 생성합니다. Jira가 개발자에게 알립니다.
6. 개발자가 기본 보안 조사 결과를 해결하고 Jira에서 티켓 상태를 TEST FIX로 변경합니다.
7. Security Hub가 조사 결과를 ARCHIVED로 업데이트하고 새 이벤트를 생성합니다. 이 이벤트로 인해 Lambda 함수가 Jira 티켓을 자동으로 종료합니다.

시나리오 2: 개발자가 위험을 허용하기로 결정합니다.

1. Security Hub가 [AWS 기본 보안 모범 사례 표준](#)과 같은 특정 보안 제어에 대한 조사 결과를 생성합니다.
2. 검색 결과 및 CreateJIRA 작업과 관련된 CloudWatch 이벤트는 Lambda 함수를 시작합니다.
3. Lambda 함수는 구성 파일과 조사 결과의 GeneratorId 필드를 사용하여 조사 결과를 에스컬레이션해야 하는지 여부를 평가합니다.
4. Lambda 함수가 조사 결과를 에스컬레이션해야 한다고 판단하고, AWS 관리 계정의 AWS Organizations로부터 SecurityContactID 계정 태그를 가져옵니다. 이 ID는 개발자와 연결되며 Jira 티켓의 담당자 ID로 사용됩니다.
5. Lambda 함수는 AWS Secrets Manager에 저장된 보안 인증 정보를 사용하여 Jira에서 티켓을 생성합니다. Jira가 개발자에게 알립니다.
6. 개발자가 위험을 허용하기로 결정하고 Jira에서 티켓 상태를 AWAITING RISK ACCEPTANCE로 변경합니다.
7. 보안 엔지니어가 요청을 검토하고 적절한 비즈니스 근거를 찾습니다. 보안 엔지니어가 Jira 티켓의 상태를 ACCEPTED RISK로 변경합니다. 이렇게 하면 Jira 티켓이 종료됩니다.

8. CloudWatch 일일 이벤트는 종료된 JIRA 티켓을 식별하고 관련 Security Hub 결과를 다음과 같이 업데이트하는 Lambda 새로 고침 함수를 시작합니다. SUPPRESSED

도구

- [AWS는 AWS CloudFormation](#) 리소스를 설정하고, 빠르고 일관되게 프로비저닝하고, 전체 AWS 계정 및 지역의 수명 주기 전반에 걸쳐 리소스를 관리할 수 있도록 지원합니다.
- [Amazon CloudWatch Events](#)는 규칙을 사용하여 이벤트를 매칭하고 함수 또는 스트림으로 라우팅함으로써 AWS 리소스의 시스템 이벤트를 모니터링할 수 있도록 지원합니다.
- [AWS Lambda](#)는 서버를 프로비저닝하거나 관리할 필요 없이 코드를 실행하는 데 도움이 되는 컴퓨팅 서비스입니다. 필요할 때만 코드를 실행하며 자동으로 확장이 가능하므로 사용한 컴퓨팅 시간만큼만 비용을 지불합니다.
- [AWS Organizations](#)는 사용자가 생성하고 중앙에서 관리하는 조직으로 여러 AWS 계정을 통합할 수 있는 계정 관리 서비스입니다.
- [AWS Secrets Manager](#)는 코드의 암호를 포함해 하드 코딩된 보안 인증 정보를 Secrets Manager에서 프로그래밍 방식으로 보안 암호를 검색하는 API 호출로 바꿀 수 있습니다.
- [AWS Security Hub](#)에서는 AWS의 보안 상태에 대한 포괄적인 보기가 제공됩니다. 이를 통해 AWS 환경에서 보안 업계 표준 및 모범 사례를 준수하는지 확인할 수도 있습니다.

코드 리포지토리

이 패턴의 코드는 [aws-securityhub-jira-software-integration](#) 리포지토리에서 사용할 수 있습니다. GitHub 여기에는 이 솔루션의 샘플 코드와 Jira 워크플로우가 포함됩니다.

에픽

Jira 구성

작업	설명	필요한 기술
워크플로우를 가져옵니다.	Jira의 관리자로서 <code>issue-workflow.xml</code> 파일을 Jira Server 인스턴스로 가져옵니다. 이 파일은 의 aws-secur	Jira 관리자

작업	설명	필요한 기술
	<p>ityhub-jira-software-integration 리포지토리에서 찾을 수 있습니다. GitHub 지침은 XML을 사용하여 워크플로우 만들기(Jira 설명서)를 참고하십시오.</p>	
워크플로우를 활성화하고 할당합니다.	<p>워크플로우는 사용자가 이를 워크플로우 체계에 할당할 때까지 비활성화됩니다. 그런 다음 프로젝트에 워크플로우 체계를 할당합니다.</p> <ol style="list-style-type: none"> 1. 프로젝트의 경우 프로젝트의 문제 유형 체계를 파악했는지 확인합니다. 새 문제 유형을 생성하거나 기존의 문제 유형(예: Bug)에서 선택할 수 있습니다. 2. 워크플로우 활성화(Jira 설명서)의 지침에 따라 가져온 워크플로우를 워크플로우 체계에 할당합니다. 3. 워크플로우 스키마를 프로젝트에 연결 지침(Jira 설명서)에 따라 프로젝트에 워크플로우 스키마를 할당합니다. 	Jira 관리자

솔루션 매개변수 설정

작업	설명	필요한 기술
솔루션 매개변수를 구성합니다.	1. 구성 폴더에서 <code>params_prod.shfile</code> 을 엽니다.	AWS 시스템 관리자

작업	설명	필요한 기술
	<p>2. 다음 파라미터 값을 입력합니다.</p> <ul style="list-style-type: none"> • ORG_ACCOUNT_ID - AWS Organizations 관리 계정의 계정 ID. 솔루션은 계정 태그를 읽고 해당 AWS 계정 태그에 정의된 특정 보안 연락처에 티켓을 배정합니다. • ORG_ROLE - AWS 조직 관리 계정에 액세스하는데 사용되는 IAM 역할의 이름입니다. 이 역할에는 OrganizationsReadOnlyAccess 권한이 있어야 합니다. • EXTERNAL_ID - ORG_ROLE에 정의된 IAM 역할을 위임하기 위해 외부 ID를 사용하는 경우 선택할 수 있는 매개변수입니다. 자세한 내용은 외부 ID 사용 방법(IAM 설명서)을 참고하십시오. • JIRA_DEFAULT_ASSIGNEE - 모든 보안 문제의 기본 담당자를 위한 Jira ID입니다. 위임된 이 기본 값은 계정에 제대로 태그가 지정되지 않았거나 역할을 맡을 수 없는 경우에 사용됩니다. • JIRA_INSTANCE - Jira 서버의 HTTPS 주소는 	

작업	설명	필요한 기술
	<p>team-<team-id>.atllassian.net/ 형식으로 표시됩니다.</p> <ul style="list-style-type: none"> • JIRA_PROJECT_KEY – 티켓을 만드는 데 사용된 Jira 프로젝트 키의 이름 (예: SEC 또는 TEST). 이 프로젝트는 Jira에 이미 존재해야 합니다. • ISSUE_TYPE – Jira에서 프로젝트에 할당된 문제 유형 체계의 이름(예: Bug 또는 Security Issue). • REGIONS – 이 솔루션을 배포하려는 AWS 리전 코드 목록(예:eu-west-1). <p>3. 솔루션 매개변수 파일을 저장하고 닫습니다.</p>	

작업	설명	필요한 기술
<p>자동화하려는 조사 결과를 파악합니다.</p>	<ol style="list-style-type: none"> 1. https://console.aws.amazon.com/securityhub/에서 Security Hub 콘솔을 엽니다. 2. Security Hub 탐색 창에서 조사 결과를 선택합니다. 3. 조사 결과 제목을 선택합니다. 4. 조사 결과 ID를 선택합니다. 그러면 조사사 결과에 대한 전체 JSON이 표시됩니다. 5. JSON에서 GeneratorId 필드의 문자열을 복사합니다. 이 값은 AWS 보안 조사 결과 형식(ASFF)입니다. 예를 들어, aws-foundational-security-best-practices/v/1.0.0/S3.1 은 보안 제어 S3.1의 조사 결과에 해당하며 S3 공개 액세스 차단 설정을 활성화해야 합니다. 6. 자동화하려는 조사 결과에 대한 모든 GeneratorID 값을 복사할 때까지 이 단계를 반복합니다. 	

작업	설명	필요한 기술
<p>구성 파일에 조사 결과를 추가합니다.</p>	<ol style="list-style-type: none"> 1. src/code에서 config.js onconfig 파일을 엽니다. 2. 이전 스토리에서 검색한 GeneratorID 값을 default 매개 변수에 붙여 넣고 쉼표를 사용하여 각 ID를 구분합니다. 3. 구성 파일을 저장하고 닫습니다. <p>다음의 코드 예제는 aws-foundational-security-best-practices/v/1.0.0/SNS.1 및 aws-foundational-security-best-practices/v/1.0.0/S3.1 조사 결과를 자동화하는 방법을 보여줍니다.</p> <pre data-bbox="592 1197 1031 1845"> { "Controls" : { "eu-west-1": ["arn:aws:securityhub:::ruleset/cis-aws-foundations-benchmark/v/1.2.0/rule/1.22"], "default": [aws-foundational-security-best-practices/v/1.0.0/SNS.1, aws-foundational-security-best-practices/v/1.0.0/S3.1] } } </pre>	<p>AWS 시스템 관리자</p>

작업	설명	필요한 기술
	<pre data-bbox="592 205 1029 346">] } } </pre> <p data-bbox="592 380 1029 701">참고: 각 AWS 리전에 대해 서로 다른 조사 결과를 자동화하도록 선택할 수 있습니다. 조사 결과 중복을 방지하는 데 도움이 되는 좋은 방법은 IAM 관련 제어 생성을 자동화할 단일 리전을 선택하는 것입니다.</p>	

통합 배포

작업	설명	필요한 기술
통합을 배포합니다.	<p data-bbox="592 995 1029 1079">명령줄 터미널에 다음 명령을 입력합니다.</p> <pre data-bbox="592 1119 1029 1199"> ./deploy.sh prod </pre>	AWS 시스템 관리자
AWS Secrets Manager에 Jira 보안 인증 업로드	<ol data-bbox="592 1234 1029 1871" style="list-style-type: none"> 1. https://console.aws.amazon.com/secretsmanager/에서 Secrets Manager 콘솔을 엽니다. 2. 보안 암호에서 새 보안 암호 저장을 선택합니다. 3. 보안 암호 유형(Secret type)에서 다른 유형의 보안 암호(Other type of secret)를 선택합니다. 4. Jira Enterprise를 사용하는 경우 키/값 쌍에 대해 다음을 수행하십시오. 	AWS 시스템 관리자

작업	설명	필요한 기술
	<ul style="list-style-type: none"> • 첫 번째 행에서 키 상자에 auth를 입력한 다음, 값 상자에 token_auth 를 입력합니다. • 두 번째 행을 추가하고 키 상자에 token를 입력한 다음, 값 상자에 개인용 액세스 토큰을 입력합니다. <p>Jira Cloud를 사용하는 경우 키/값 쌍에 대해 다음을 수행하세요.</p> <ul style="list-style-type: none"> • 첫 번째 행에서 키 상자에 auth를 입력한 다음, 값 상자에 basic_auth 를 입력합니다. • 두 번째 행을 추가하고 키 상자에 token을 입력한 다음, 값 상자에 API 토큰을 입력합니다. • 세 번째 행을 추가하고 키 상자에 email을 입력한 다음, 값 상자에 이메일 주소를 입력합니다. <p>5. 다음을 선택합니다.</p> <p>6. 보안 암호 이름에 Jira-Token 을 입력하고 페이지 하단에서 다음을 선택합니다.</p> <p>7. 보안 암호 교체 페이지에서 자동 교체 사용 중지를 그대로 사용하고 페이지 하단에서 다음을 선택합니다.</p>	

작업	설명	필요한 기술
	<p>8. 검토 페이지에서 보안 암호 세부 정보를 검토한 후 저장을 선택합니다.</p>	
<p>Security Hub 사용자 지정 작업을 생성합니다.</p>	<p>1. 각 AWS 지역에 대해 AWS 명령줄 인터페이스 (AWS CLI) Line AWS CLI에서 명령을 사용하여 create-action-target 이름이 지정된 Security Hub 사용자 지정 작업을 생성합니다.</p> <p>CreateJiraIssue</p> <pre>aws securityhub create-action-target --name "CreateJiraIssue" \ --description "Create ticket in JIRA" \ --id "CreateJiraIssue" --region \$<aws-region></pre> <p>2. https://console.aws.amazon.com/securityhub/에서 Security Hub 콘솔을 엽니다.</p> <p>3. Security Hub 탐색 창에서 조사 결과를 선택합니다.</p> <p>4. 조사 결과 목록에서 에스컬레이션하려는 조사 결과를 선택합니다.</p> <p>5. 작업 메뉴에서 CreateJiraIssue 를 선택합니다.</p>	<p>AWS 시스템 관리자</p>

관련 리소스

- [AWS Service Management Connector for Jira Service Management](#)
- [AWS 기반 보안 모범 사례 표준](#)

추가 정보

Jira 티켓의 예

지정된 Security Hub 조사 결과가 발생하면 이 솔루션은 Jira 티켓을 자동으로 생성합니다. 티켓에 포함되는 정보는 다음과 같습니다.

- 제목 — 제목은 다음과 같은 형식으로 보안 문제를 식별합니다.

```
AWS Security Issue :: <AWS account ID> :: <Security Hub finding title>
```

- 설명 — 티켓의 설명 섹션은 조사 결과와 관련된 보안 제어를 설명하고, Security Hub 콘솔의 조사 결과에 대한 링크를 포함하며, Jira 워크플로우에서 보안 문제를 처리하는 방법에 대해 간략한 설명을 제공합니다.

다음은 자동으로 생성된 Jira 티켓의 예입니다.

제목	AWS Security Issue :: 012345678912 :: Lambda.1 Lambda 함수 정책은 공개 액세스를 금지해야 합니다.
설명	어떤 문제가 있습니까? 귀하가 알고 있는 AWS 계정 012345678912에서 보안 감지 조사 결과를 발견했습니다. 이 제어 기능은 Lambda 리소스에 연결된 Lambda 함수 정책이 퍼블릭 액세스를 금지하는 지 여부를 확인합니다. Lambda 함수 정책에서 공개 액세스를 허용하는 경우 제어가 실패합니 다. <Security Hub 조사 결과 연결>

티켓을 가지고 어떻게 해야 하나요?

- 계정에 접속하여 구성을 확인합니다. “수정하기 위해 할당됨”으로 이동하여 티켓 작업을 확인합니다. 수정이 완료되면 보안 팀에서 문제가 해결되었는지 확인할 수 있도록 테스트 수정으로 이동합니다.
- 위험을 허용해야 한다고 생각되면 “위험 허용 대기 중”으로 이동합니다. 이를 위해서는 보안 엔지니어의 검토가 필요합니다.
- 오탐이라고 생각되면 “오탐으로 표시”로 전환합니다. 보안 엔지니어가 검토한 후 상황에 맞게 재개/종료됩니다.

EC2 Image Builder와 Terraform을 사용하여 강화된 컨테이너 이미지를 위한 파이프라인을 구축하십시오.

마이크 세인크로스(AWS)와 앤드류 레인즈(AWS)가 제작했습니다.

코드 리포지토리: Terraform EC2 Image Builder 컨테이너 강화 파이프라인	환경: 프로덕션	소스: 패커, 셰프 또는 퓨어 앤서블
대상: EC2 Image Builder	R 유형: 리아키텍트	워크로드: 오픈 소스
기술: 보안, ID, 규정 준수 DevOps	AWS 서비스: Amazon EC2 컨테이너 레지스트리, Amazon EC2 Image Builder	

요약

이 패턴은 강화된 [Amazon Linux 2](#) 기본 컨테이너 이미지를 생성하는 [EC2 Image Builder 파이프라인](#)을 구축합니다. Terraform은 강화된 컨테이너 이미지를 생성하는 데 사용되는 인프라를 구성하고 프로비저닝하기 위한 코드형 인프라(IaC) 도구로 사용됩니다. 이 레시피는 레드햇 엔터프라이즈 리눅스(RHEL) 7 STIG 버전 3 릴리스 7 – 미디엄에 따라 강화된 도커 기반 Amazon 리눅스 2 컨테이너 이미지를 배포하는 데 도움이 됩니다. (EC2 Image [Builder설명서의 Linux STIG 구성 요소 섹션에서 STIG-빌드-리눅스 미디엄 버전 2022.2.1](#)을 참조하십시오.) 이를 골든 컨테이너 이미지라고 합니다.

빌드에는 두 개의 [Amazon EventBridge 규칙](#)이 포함되어 있습니다. 한 가지 규칙은 [Amazon Inspector의 조사 결과](#)가 높음 또는 중요일 때 컨테이너 이미지 파이프라인을 시작하여 비보안 이미지가 대체되도록 하는 것입니다. 이 규칙을 사용하려면 Amazon Inspector와 Amazon Elastic Container Registry(Amazon ECR)의 [향상된 스캔 기능](#)을 모두 활성화해야 합니다. 또 다른 규칙은 Amazon ECR 리포지토리로 이미지 푸시가 성공한 후 Amazon Simple Queue Service(Amazon SQS) [대기열](#)에 알림을 전송하여 최신 컨테이너 이미지를 사용할 수 있도록 합니다.

사전 조건 및 제한 사항

사전 조건

- 인프라를 배포할 수 있는 [AWS 계정](#).

- 로컬 배포를 위한 AWS 보안 인증을 설정하기 위해 [AWS Command Line Interface\(AWS CLI\)](#)가 설치되었습니다.
- Terraform 설명서의 [지침](#)에 따라 Terraform을 [다운로드](#)하고 설정했습니다.
- [Git](#) (로컬 시스템에서 프로비저닝하는 경우)
- AWS 리소스를 생성하는 데 사용할 수 있는 AWS 계정 내 [역할](#).
- 모든 변수는 [.tfvars](#) 파일에 정의되어 있습니다. 또는 Terraform 구성을 적용할 때 모든 변수를 정의할 수 있습니다.

제한 사항

- 이 솔루션은 프라이빗 서브넷에서의 인터넷 연결을 위한 [NAT 게이트웨이](#)와 [인터넷 게이트웨이](#)를 포함하는 Amazon Virtual Private Cloud(VPC) 인프라를 생성합니다. [VPC 엔드포인트](#)는 사용할 수 없습니다. [AWS 태스크 오케스트레이터 및 Executor \(\)](#)의 부트스트랩 프로세스가 [인터넷에서 AWSTOE AWS CLI 버전 2를 설치하기](#) 때문입니다.

제품 버전

- Amazon Linux 2
- AWS CLI 버전 1.1 이상

아키텍처

대상 기술 스택

이 패턴은 다음을 포함하여 43개의 리소스를 생성합니다.

- Amazon Simple Storage Service(S3) [버킷](#) 2개: 하나는 파이프라인 구성 요소 파일용, 다른 하나는 서버 액세스 및 Amazon VPC 흐름 로그용
- [Amazon ECR 리포지토리](#)
- 퍼블릭 서브넷, 프라이빗 서브넷, 라우팅 테이블, NAT 게이트웨이 및 인터넷 게이트웨이가 포함된 VPC
- EC2 Image Builder 파이프라인, 레시피, 구성 요소
- 컨테이너 이미지
- 이미지 암호화를 위한 AWS Key Management Service (AWS KMS) [키](#)
- SQS 대기열

- 역할 3개: EC2 Image Builder 파이프라인을 실행하는 역할 1개, EC2 Image Builder용 인스턴스 프로필 1개, 규칙을 위한 역할 1개 EventBridge
- 두 가지 규칙 EventBridge

Terraform 모듈 구조

소스 코드는 GitHub 리포지토리 [Terraform EC2 Image Builder](#) 컨테이너 강화 파이프라인을 참조하십시오.

```

### components.tf
### config.tf
### dist-config.tf
### files
#   ###assumption-policy.json
### hardening-pipeline.tfvars
### image.tf
### infr-config.tf
### infra-network-config.tf
### kms-key.tf
### main.tf
### outputs.tf
### pipeline.tf
### recipes.tf
### roles.tf
### sec-groups.tf
### trigger-build.tf
### variables.tf

```

모듈 세부 정보

- components.tf/files 디렉터리의 콘텐츠를 업로드하기 위한 Amazon S3 업로드 리소스를 포함합니다. 여기에 사용자 지정 구성 요소 YAML 파일을 모듈식으로 추가할 수도 있습니다.
- /files에는 components.tf에서 사용되는 구성 요소를 정의하는 .yaml 파일이 들어 있습니다.
- image.tf은(는) 기본 이미지 Operations Management 체제에 대한 정의를 포함합니다. 여기에서 다른 기본 이미지 파이프라인의 정의를 수정할 수 있습니다.
- infr-config.tf 및 dist-config.tf은(는) 이미지를 가동하고 배포하는 데 필요한 최소 AWS 인프라를 위한 리소스를 포함합니다.
- infra-network-config.tf은(는) 컨테이너 이미지를 배포하기 위한 최소 VPC 인프라를 포함합니다.

- `hardening-pipeline.tfvars`은(는) 적용 시 사용할 Terraform 변수를 포함합니다.
- `pipeline.tf`은(는) Terraform에서 EC2 Image Builder 파이프라인을 생성하고 관리합니다.
- `recipes.tf`은(는) 다양한 구성 요소 조합을 지정하여 컨테이너 레시피를 생성할 수 있는 곳입니다.
- `roles.tf`에는 Amazon Elastic Compute Cloud (Amazon EC2) 인스턴스 프로파일 및 파이프라인 배포 역할에 대한 AWS Identity 및 Access Management (IAM) 정책 정의가 포함되어 있습니다.
- `trigger-build.tf` EventBridge 규칙 및 SQS 대기열 리소스가 포함되어 있습니다.

대상 아키텍처

다이어그램은 다음 워크플로를 보여줍니다:

1. EC2 Image Builder는 정의된 레시피를 사용하여 컨테이너 이미지를 구축합니다. 이 레시피는 운영 체제 업데이트를 설치하고 Amazon Linux 2 기본 이미지에 RHEL Medium STIG를 적용합니다.
2. 강화된 이미지는 프라이빗 Amazon ECR 레지스트리에 게시되며, 이미지가 성공적으로 게시되면 EventBridge 규칙에 따라 SQS 대기열로 메시지가 전송됩니다.
3. Amazon Inspector가 향상된 스캔을 위해 구성된 경우 Amazon ECR 레지스트리를 스캔합니다.
4. Amazon Inspector에서 이미지에 대한 심각도 또는 높음 심각도를 생성하는 경우, EventBridge 규칙에 따라 EC2 Image Builder 파이프라인이 다시 실행되고 새로 강화된 이미지를 게시하도록 트리거합니다.

자동화 및 규모 조정

- 이 패턴은 컴퓨터에 인프라를 프로비저닝하고 파이프라인을 구축하는 방법을 설명합니다. 하지만 대규모로 사용하기 위한 것입니다. Terraform 모듈을 로컬에 배포하는 대신 [Terraform용 Account Factory](#)가 있는 [AWS Control Tower](#) 환경과 같은 다중 계정 환경에서 사용할 수 있습니다. 이 경우 구성 상태를 로컬에서 관리하는 대신 [백엔드 상태 S3 버킷](#)을 사용하여 Terraform 상태 파일을 관리해야 합니다.
- 확장된 사용을 위해 Control Tower 또는 랜딩 존 계정 모델에서 Shared Services 또는 Common Services 계정과 같은 하나의 중앙 계정에 솔루션을 배포하고 소비자 계정에 Amazon ECR 리포지토리와 AWS KMS 키에 액세스할 수 있는 권한을 부여하십시오. 설정에 대한 자세한 내용은 [re:Post](#) 문서를 참조하십시오. Amazon ECR [이미지 리포지토리에서 보조 계정이 이미지를 푸시하거나 가져오도록 허용하려면 어떻게 해야 합니까?](#) 예를 들어 [계정 자동 판매기](#) 또는 Account Factory for Terraform에서 각 계정 기준 또는 계정 사용자 지정 기준에 권한을 추가하여 해당 Amazon ECR 리포지토리 및 암호화 키에 대한 액세스를 제공합니다.

- 컨테이너 이미지 파이프라인이 배포된 후에는 [구성 요소](#)와 같은 EC2 Image Builder 기능을 사용하여 수정할 수 있으며, 이를 통해 더 많은 구성 요소를 Docker 빌드에 패키징할 수 있습니다.
- 컨테이너 이미지를 암호화하는 데 사용되는 AWS KMS 키는 이미지를 사용할 계정 간에 공유해야 합니다.
- 전체 Terraform 모듈을 복제하고 다음 `recipes.tf` 속성을 수정하여 다른 이미지에 대한 지원을 추가할 수 있습니다.
 - `parent_image = "amazonlinux:latest"`을(를) 다른 이미지 유형으로 수정하십시오.
 - `repository_name`이(가) 기존 Amazon ECR 리포지토리를 가리키도록 수정하십시오. 그러면 기존 Amazon ECR 리포지토리에 다른 상위 이미지 유형을 배포하는 또 다른 파이프라인이 생성됩니다.

도구

도구

- Terraform (IaC 프로비저닝)
- Git (로컬에서 프로비저닝하는 경우)
- AWS CLI 버전 1 또는 버전 2 (로컬에서 프로비저닝하는 경우)

코드

이 패턴의 코드는 GitHub 리포지토리 [Terraform EC2 Image Builder](#) 컨테이너 강화 파이프라인에 있습니다. 샘플 코드를 사용하려면 다음 섹션의 지침을 따르세요.

에픽

인프라 프로비저닝

작업	설명	필요한 기술
로컬 보안 인증을 설정합니다.	<p>AWS 임시 보안 인증을 설정합니다.</p> <ol style="list-style-type: none"> 1. AWS CLI가 설치되었는지 확인합니다. <pre>\$ aws --version</pre>	AWS DevOps

작업	설명	필요한 기술
	<pre>aws-cli/1.16.249 Python/3.6.8...</pre> <ul style="list-style-type: none"> • AWS CLI 버전은 1.1 이상이어야 합니다. • 명령을 찾을 수 없는 경우, AWS CLI를 설치하십시오. <p>2. aws configure 을(를) 실행하고 다음 값을 제공하십시오.</p> <pre>\$ aws configure AWS Access Key ID [*****]: <Your AWS access key ID> AWS Secret Access Key [*****x]: <Your AWS secret access key> Default region name: [us-east-1]: <Your desired Region for deployment> Default output format [None]: <Your desired output format></pre>	

작업	설명	필요한 기술
리포지토리를 복제합니다.	<p>1. 이 패턴과 함께 제공된 리포지토리를 복제하십시오. HTTPS 또는 Secure Shell(SSH)을 사용할 수 있습니다.</p> <p>HTTPS:</p> <pre>git clone https://github.com/aws-samples/terraform-ec2-image-builder-container-hardening-pipeline</pre> <p>SSH:</p> <pre>git clone git@github.com:aws-samples/terraform-ec2-image-builder-container-hardening-pipeline.git</pre> <p>2. 다음 솔루션이 포함된 로컬 디렉터리로 이동합니다.</p> <pre>cd terraform-ec2-image-builder-container-hardening-pipeline</pre>	AWS DevOps

작업	설명	필요한 기술
변수를 업데이트합니다.	<p>환경 및 원하는 구성에 맞게 <code>hardening-pipeline.tfvars</code> 파일의 변수를 업데이트하십시오. <code>account_id</code> 을(를) 제공해야 합니다. 하지만 나머지 변수도 원하는 배포에 맞게 수정해야 합니다. 모든 변수가 필요합니다.</p> <pre data-bbox="592 634 1027 1839"> account_id = "<DEPLOYMENT-ACCOUNT-ID>" aws_region = "us-east-1" vpc_name = "example-hardening-pipeline-vpc" kms_key_alias = "image-builder-container-key" ec2_iam_role_name = "example-hardening-instance-role" hardening_pipeline_role_name = "example-hardening-pipeline-role" aws_s3_ami_resources_bucket = "example-hardening-ami-resources-bucket-0123" image_name = "example-hardening-al2-container-image" ecr_name = "example-hardening-container-repo" recipe_version = "1.0.0" </pre>	AWS DevOps

작업	설명	필요한 기술
	<pre>ebs_root_vol_size = 10</pre> <p>다음은 각 변수에 대한 설명입니다.</p> <ul style="list-style-type: none"> • <code>account_id</code> – 솔루션을 배포하려는 AWS 계정 번호. • <code>aws_region</code> – 솔루션을 배포할 AWS 리전입니다. • <code>vpc_name</code> – VPC 인프라의 이름. • <code>kms_key_alias</code> – EC2 Image Builder 인프라 구성에서 사용할 AWS KMS 키 이름입니다. • <code>ec2_iam_role_name</code> – EC2 인스턴스 프로파일로 사용될 역할의 이름. • <code>hardening_pipeline_role_name</code> – 강화 파이프라인을 배포하는 데 사용할 역할의 이름. • <code>aws_s3_ami_resources_bucket</code> – 파이프라인 및 컨테이너 이미지를 빌드하는 데 필요한 모든 파일을 호스팅할 S3 버킷의 이름. • <code>image_name</code> – 컨테이너 이미지 이름. 이 값은 3~50자 사이여야 하며 영숫자와 하이픈만 포함해야 합니다. 	

작업	설명	필요한 기술
	<ul style="list-style-type: none"> • <code>ecr_name</code> – 컨테이너 이미지를 저장할 Amazon ECR 레지스트리의 이름. • <code>recipe_version</code> – 이미지 레시피의 의미 체계 버전입니다. 기본값은 1.0.0입니다. • <code>ebs_root_vol_size</code> – Amazon Elastic Block Store(Amazon EBS) 루트 볼륨의 크기 (기가바이트). 기본값은 10GB입니다. 	
Terraform을 초기화합니다.	<p>변수 값을 업데이트한 후 Terraform 구성 디렉터리를 초기화할 수 있습니다. 구성 디렉터리를 초기화하면 구성에 정의된 AWS 공급자가 다운로드 되고 설치됩니다.</p> <pre data-bbox="597 1150 1026 1226">terraform init</pre> <p>Terraform이 성공적으로 초기화되었으며 설치된 제공자의 버전을 식별한다는 메시지가 표시되어야 합니다.</p>	AWS DevOps

작업	설명	필요한 기술
<p>인프라를 배포하고 컨테이너 이미지를 생성합니다.</p>	<p>다음 명령을 사용하여 .tfvars 파일에 정의된 변수를 사용하여 Terraform 모듈을 초기화, 검증 및 환경에 적용하십시오.</p> <pre data-bbox="591 489 1024 728"> terraform init && terraform validate && terraform apply -var-file *.tfvars -auto-approve </pre>	<p>AWS DevOps</p>
<p>컨테이너를 사용자 지정합니다.</p>	<p>EC2 Image Builder에서 파이프라인과 초기 레시피를 배포한 후 새 버전의 컨테이너 레시피를 생성할 수 있습니다.</p> <p>EC2 Image Builder에서 사용 가능한 31개 이상의 구성 요소를 추가하여 컨테이너 빌드를 사용자 지정할 수 있습니다. 자세한 내용은 EC2 Image Builder 설명서에서 새 버전의 컨테이너 레시피 생성의 구성 요소 섹션을 참조하십시오.</p>	<p>AWS 관리자</p>

리소스 검증

작업	설명	필요한 기술
<p>AWS 인프라 프로비저닝을 검증합니다.</p>	<p>첫 번째 Terraform apply 명령을 성공적으로 완료한 후 로컬에서 프로비저닝하는 경우 로컬 시스템의 터미널에서 다음 스니펫을 확인할 수 있습니다.</p>	<p>AWS DevOps</p>

작업	설명	필요한 기술
	<pre>Apply complete! Resources: 43 added, 0 changed, 0 destroyed.</pre>	
<p>개별 AWS 인프라 리소스를 검증합니다.</p>	<p>로컬에서 프로비저닝하려면 다음 명령을 실행하면 배포된 개별 리소스의 유효성을 검사할 수 있습니다.</p> <pre>terraform state list</pre> <p>이 명령은 43개 리소스 목록을 반환합니다.</p>	<p>AWS DevOps</p>

리소스 제거

작업	설명	필요한 기술
<p>인프라 및 컨테이너 이미지를 제거합니다.</p>	<p>Terraform 구성 작업을 마치면 다음 명령을 실행하여 리소스를 제거할 수 있습니다.</p> <pre>terraform init && terraform validate && terraform destroy -var- file *.tfvars -auto-app rove</pre>	<p>AWS DevOps</p>

문제 해결

문제	Solution
<p>공급자 보안 인증을 검증하는 중 오류가 발생했습니다.</p>	<p>로컬 시스템에서 Terraform apply 또는 destroy 명령을 실행할 때 다음과 유사한 오류가 발생할 수 있습니다.</p> <pre data-bbox="829 506 1507 947"> Error: configuring Terraform AWS Provider: error validating provider credentials: error calling sts:GetCallerIdentity: operation error STS: GetCallerIdentity, https response error StatusCode: 403, RequestID: 123456a9-fbc1-40ed-b8d8-513d0133ba7f, api error InvalidClientTokenId: The security token included in the request is invalid. </pre> <p>이 오류는 로컬 시스템 구성에 사용된 보안 인증 정보의 보안 토큰이 만료되어 발생합니다.</p> <p>오류를 해결하려면 AWS CLI 설명서의 구성 설정 설정 및 보기를 참조하십시오.</p>

관련 리소스

- [Terraform EC2 Image Builder 컨테이너 강화](#) 파이프라인 (리포지토리) GitHub
- [EC2 Image Builder 설명서](#)
- [Terraform용 AWS Control Tower 어카운트 팩토리](#) (AWS 블로그 게시물)
- [백엔드 상태 S3 버킷](#) (Terraform 설명서)
- [AWS CLI의 최신 버전 설치 또는 업데이트](#) (AWS CLI 설명서)
- [Terraform 다운로드](#)

Terraform을 사용하여 AWS Organizations의 IAM 액세스 키 관리를 중앙 집중화하십시오.

제작: 아르티 라즈푸트 (AWS), 친타마니 아팔레 (AWS), T.V.R.L. 파니 쿠마르 다디 (AWS), 프라딕 쿠마르 판디 (AWS), 마유리 신데 (AWS), 프라탑 쿠마르 난다 (AWS)

환경: 프로덕션

기술: 보안, 자격 증명, 규정 준수, 분석, 인프라

AWS 서비스: 아마존, AWS Lambda EventBridge, AWS Organizations, AWS Secrets Manager, 아마존 SES

요약

키와 암호에 대한 보안 규칙을 적용하는 것은 모든 조직의 필수 작업입니다. 한 가지 중요한 규칙은 보안을 강화하기 위해 AWS Identity 및 Access Management (IAM) 키를 정기적으로 교체하는 것입니다. AWS 액세스 키는 일반적으로 팀이 AWS 명령줄 인터페이스 (AWS CLI) 또는 AWS 외부 애플리케이션에서 AWS에 액세스하고자 할 때마다 로컬에서 생성 및 구성됩니다. 조직 전체에서 강력한 보안을 유지하려면 요구 사항이 충족된 후 또는 정기적으로 이전 보안 키를 변경하거나 삭제해야 합니다. 조직 내 여러 계정의 키 순환을 관리하는 프로세스는 시간이 많이 걸리고 지루합니다. 이 패턴은 테라폼용 Account Factory (AFT) 및 AWS 서비스를 사용하여 로테이션 프로세스를 자동화하는 데 도움이 됩니다.

패턴은 다음과 같은 이점을 제공합니다.

- 중앙 위치에서 조직 내 모든 계정의 액세스 키 ID와 보안 액세스 키를 관리합니다.
- `AWS_ACCESS_KEY_ID` 및 `AWS_SECRET_ACCESS_KEY` 환경 변수를 자동으로 교체합니다.
- 사용자 자격 증명이 손상된 경우 갱신을 적용합니다.

이 패턴은 Terraform을 사용하여 AWS Lambda 함수, EventBridge Amazon 규칙 및 IAM 역할을 배포합니다. EventBridge 규칙은 정기적으로 실행되며 생성 시기를 기준으로 모든 사용자 액세스 키를 나열하는 Lambda 함수를 호출합니다. 추가 Lambda 함수는 이전 키가 사용자가 정의한 순환 기간 (예: 45일) 보다 오래된 경우 새 액세스 키 ID 및 보안 액세스 키를 생성하고 Amazon Simple Notification Service (Amazon SNS) 및 Amazon Simple Email Service (Amazon SES) 를 사용하여 보안 관리자에게 알립니다. AWS Secrets Manager에서 해당 사용자에게 대한 암호가 생성되고, 이전 보안 액세스 키는 Secrets Manager에 저장되며, 이전 키에 액세스할 수 있는 권한이 구성됩니다. 이전 액세스 키는 더

이상 사용되지 않도록 비활성 기간 (예: 60일, 이 예에서는 키가 교체된 지 15일 후) 이 지나면 비활성화됩니다. 비활성 버퍼 기간 (예: 90일 또는 예시에서는 키가 교체된 후 45일) 이 지나면 AWS Secrets Manager에서 이전 액세스 키가 삭제됩니다. 자세한 아키텍처 및 워크플로우는 [아키텍처](#) 섹션을 참조하십시오.

사전 조건 및 제한 사항

- [AWS Control Tower](#) (버전 3.1 이상) 를 사용하여 구축한 조직의 랜딩 존
- 세 개의 계정으로 구성된 [Account Factory for Terraform \(AFT\)](#):
 - [조직 관리 계정](#)은 중앙 위치에서 전체 조직을 관리합니다.
 - [AFT 관리 계정](#)은 Terraform 파이프라인을 호스팅하고 인프라를 배포 계정에 배포합니다.
 - [배포 계정](#)은 이 완전한 솔루션을 배포하고 중앙 위치에서 IAM 키를 관리합니다.
- Terraform 버전 0.15.0 이상은 배포 계정에서 인프라를 프로비저닝하기 위한 것입니다.
- [아마존 심플 이메일 서비스 \(Amazon SES\)](#) 에서 구성된 이메일 주소입니다.
- (권장) 보안을 강화하려면 이 솔루션을 [가상 사설 클라우드 \(VPC\) 내의 사설 서브넷](#) (배포 계정) 에 배포하십시오. 변수를 사용자 지정할 때 VPC와 서브넷의 세부 정보를 제공할 수 있습니다 ([Epic](#) 섹션의 코드 파이프라인의 매개변수 사용자 지정 참조).

아키텍처

AFT 리포지토리

이 패턴은 Account Factory for Terraform (AFT) 을 사용하여 필요한 모든 AWS 리소스를 생성하고 코드 파이프라인을 사용하여 배포 계정에 리소스를 배포합니다. 코드 파이프라인은 두 개의 리포지토리에서 실행됩니다.

- 글로벌 사용자 지정에는 AFT에 등록된 모든 계정에서 실행되는 Terraform 코드가 포함되어 있습니다.
- 계정 사용자 지정에는 배포 계정에서 실행할 Terraform 코드가 포함됩니다.

리소스 세부 정보

AWS CodePipeline Jobs는 배포 계정에서 다음과 같은 리소스를 생성합니다.

- AWS EventBridge 규칙 및 구성된 규칙
- `account-inventory` 램다 함수

- IAM-access-key-rotation 램다 함수
- Notification 램다 함수
- 이메일 템플릿이 포함된 Amazon Simple Storage Service (Amazon S3) 버킷
- 필수 IAM 정책

아키텍처

다이어그램은 다음을 보여 줍니다.

1. EventBridge 규칙은 24시간마다 account-inventory Lambda 함수를 호출합니다.
2. account-inventoryLambda 함수는 AWS Organizations에 모든 AWS 계정 ID, 계정 이름 및 계정 이메일의 목록을 쿼리합니다.
3. account-inventoryLambda 함수는 각 AWS 계정에 대해 IAM-access-key-auto-rotation Lambda 함수를 시작하고 추가 처리를 위해 해당 계정에 메타데이터를 전달합니다.
4. IAM-access-key-auto-rotationLambda 함수는 위임된 IAM 역할을 사용하여 AWS 계정에 액세스합니다. Lambda 스크립트는 계정의 모든 사용자 및 IAM 액세스 키에 대해 감사를 실행합니다.
5. IAM 키 교체 임계값 (회전 기간) 은 IAM-access-key-auto-rotation Lambda 함수가 배포될 때 환경 변수로 구성됩니다. 순환 기간이 수정되면 IAM-access-key-auto-rotation Lambda 함수가 업데이트된 환경 변수와 함께 재배포됩니다. [회전 기간, 이전 키의 비활성 기간, 이전 키가 삭제되는 시점의 비활성 버퍼를 설정하도록 파라미터를 구성할 수 있습니다 \(Epic 섹션의 코드 파이프라인에 대한 파라미터 사용자 지정 참조\).](#)
6. IAM-access-key-auto-rotationLambda 함수는 구성을 기반으로 액세스 키의 수명을 검증합니다. IAM 액세스 키의 사용 기간이 정의한 순환 기간을 초과하지 않은 경우 Lambda 함수는 추가 조치를 취하지 않습니다.
7. IAM 액세스 키의 사용 기간이 사용자가 정의한 교체 기간을 초과한 경우 IAM-access-key-auto-rotation Lambda 함수는 새 키를 생성하고 기존 키를 교체합니다.
8. Lambda 함수는 이전 키를 Secrets Manager에 저장하고 액세스 키가 보안 표준에서 벗어난 사용자에게만 권한을 제한합니다. 또한 Lambda 함수는 지정된 IAM 보안 주체만 암호에 액세스하고 검색할 수 있도록 허용하는 리소스 기반 정책을 생성합니다.
9. IAM-access-key-rotationLambda 함수는 Lambda 함수를 호출합니다. Notification
- 10 NotificationLambda 함수는 S3 버킷에 이메일 템플릿을 쿼리하고 관련 활동 메타데이터가 포함된 이메일 메시지를 동적으로 생성합니다.
- 11 NotificationLambda 함수는 추가 작업을 위해 Amazon SES를 호출합니다.

12. Amazon SES는 계정 소유자의 이메일 주소로 관련 정보가 포함된 이메일을 보냅니다.

도구

서비스

- [AWS Identity and Access Management\(IAM\)](#)를 사용하면 사용자에게 대해 인증 및 권한 부여를 제어함으로써 AWS 리소스에 대한 액세스를 안전하게 관리할 수 있습니다. 이 패턴에는 IAM 역할 및 권한이 필요합니다.
- [AWS Lambda](#)는 서버를 프로비저닝하거나 관리할 필요 없이 코드를 실행하는 데 도움이 되는 컴퓨팅 서비스입니다. 필요할 때만 코드를 실행하며 자동으로 확장이 가능하므로 사용한 컴퓨팅 시간만큼만 비용을 지불합니다.
- [AWS Secrets Manager](#)는 코드의 암호를 포함해 하드 코딩된 보안 인증 정보를 Secrets Manager에서 프로그래밍 방식으로 보안 암호를 검색하는 API 호출로 바꿀 수 있습니다.
- [Amazon Simple Email Service\(Amazon SES\)](#)를 사용하면 자신의 이메일 주소와 도메인을 사용하여 이메일을 보내고 받을 수 있습니다.

기타 도구

- [Terraform](#)은 클라우드 및 온프레미스 리소스를 생성하고 관리하는 데 도움이 HashiCorp 되는 코드형 인프라 (IaC) 도구입니다.

코드 리포지토리

이 패턴에 대한 지침과 코드는 GitHub [IAM](#) 액세스 키 로테이션 리포지토리에서 확인할 수 있습니다. AWS Control Tower 중앙 배포 계정에 코드를 배포하여 중앙 위치에서 키 교체를 관리할 수 있습니다.

모범 사례

- IAM의 경우 IAM [설명서의 보안 모범 사례](#)를 참조하십시오.
- 키 교체에 대해서는 IAM 설명서의 [액세스 키 업데이트 가이드라인](#)을 참조하십시오.

에픽

소스 파일 설정

작업	설명	필요한 기술
리포지토리를 복제합니다.	<ol style="list-style-type: none"> IAM 액세스 키 로테이션 GitHub 리포지토리를 복제하십시오. <pre>\$ git clone https://github.com/aws-samples/centralized-iam-key-management-aws-organizations-terraform.git</pre> <ol style="list-style-type: none"> 리포지토리의 로컬 사본에 세 개의 폴더가 있는지 확인하십시오. <pre>\$ cd Iam-Access-keys-Rotation \$ ls org-account-customization global-account-customization account-customization</pre>	DevOps 엔지니어

계정 구성

작업	설명	필요한 기술
부트스트래핑 계정을 설정합니다.	AFT 부트스트래핑 프로세스의 일환으로 로컬 <code>aft-bootstrap</code> 컴퓨터에 호출된 폴더가 있어야 합니다.	DevOps 엔지니어

작업	설명	필요한 기술
	<ol style="list-style-type: none"> 모든 Terraform 파일을 로컬 GitHub org-account-customization 폴더에서 폴더로 수동으로 복사합니다. aft-bootstrap Terraform 명령을 실행하여 AWS Control Tower 관리 계정에서 글로벌 교차 계정 역할을 구성합니다. <pre style="border: 1px solid #ccc; border-radius: 10px; padding: 10px; margin: 10px 0;">\$ cd aft-bootstrap \$ terraform init \$ terraform apply - auto-approve</pre>	
<p>글로벌 커스터마이징을 구성합니다.</p>	<p>AFT 폴더 설정의 일부로 로컬 aft-global-customizations 컴퓨터에 호출된 폴더가 있어야 합니다.</p> <ol style="list-style-type: none"> 로컬 GitHub global-account-customization 폴더의 모든 Terraform 파일을 수동으로 폴더로 복사합니다. aft-global-customizations/terraform 코드를 AWS로 CodeCommit 푸시하십시오. <pre style="border: 1px solid #ccc; border-radius: 10px; padding: 10px; margin: 10px 0;">\$ git add * \$ git commit -m "message" \$ git push</pre>	DevOps 엔지니어

작업	설명	필요한 기술
<p>계정 사용자 지정을 구성합니다.</p>	<p>AFT 폴더 설정의 일부로 로컬 <code>aft-account-customizations</code> 컴퓨터에서 폴더를 호출할 수 있게 되었습니다.</p> <ol style="list-style-type: none"> 1. 벤더 계정 번호로 폴더를 만드세요. 2. 로컬 GitHub 계정 사용자 지정 폴더의 모든 Terraform 파일을 수동으로 폴더로 복사하십시오. <code>aft-account-customizations/<vended account>/terraform</code> 3. 코드를 AWS로 CodeCommit 푸시하십시오. <div style="border: 1px solid #ccc; border-radius: 10px; padding: 10px; margin-top: 10px;"> <pre>\$ git add * \$ git commit -m "message" \$ git push</pre> </div>	<p>DevOps 엔지니어</p>

코드 파이프라인의 파라미터를 사용자 지정하세요.

작업	설명	필요한 기술
<p>모든 계정의 비 Terraform 코드 파이프라인 매개변수를 사용자 지정합니다.</p>	<p><code>input.auto.tfvars aft-global-customizations/terraform/</code> 폴더에 호출된 파일을 만들고 필요한 입력 데이터를 제공합니다. 기본값은 GitHub 리포지토리의 파일을 참조하십시오.</p>	<p>DevOps 엔지니어</p>

작업	설명	필요한 기술
<p>배포 계정의 코드 파이프라인 매개변수를 사용자 지정합니다.</p>	<p><code>input.auto.tfvars aft-account-customizations/<AccountName>/terraform/</code> 폴더에서 라는 파일을 생성하고 코드를 AWS로 CodeCommit 푸시합니다. 코드를 AWS로 푸시하면 코드 파이프라인이 CodeCommit 자동으로 시작됩니다.</p> <p>다음을 포함하여 조직의 요구 사항에 따라 파라미터 값을 지정하십시오 (기본값은 Github 리포지토리의 파일 참조).</p> <ul style="list-style-type: none"> • <code>s3_bucket_name</code> — 이메일 템플릿의 고유한 버킷 이름. • <code>s3_bucket_prefix</code> — S3 버킷 내 폴더 이름. • <code>admin_email_addresses</code> — 알림을 받아야 하는 관리자의 이메일 주소입니다. • <code>org_list_account</code> — 관리 계정의 계정 번호. • <code>rotation_period</code> — 키를 활성에서 비활성으로 교체해야 하는 기간 (일). • <code>inactive_period</code> — 회전된 키를 비활성화해야 하는 기간 (일). 이 값은 의 값보다 커야 합니다. <code>rotation_period</code> 	<p>DevOps 엔지니어</p>

작업	설명	필요한 기술
	<ul style="list-style-type: none"> • <code>inactive_buffer</code> — 키 회전과 비활성화 사이의 유예 기간. • <code>recovery_grace_period</code> — 키 비활성화와 삭제 사이의 유예 기간. • <code>dry_run_flag</code> — 테스트 목적으로 키를 교체하지 않고 관리자에게 알림을 보내려면 <code>true</code>로 설정하십시오. • <code>store_secrets_in_central_account</code> — 배포 계정에 암호를 저장하려면 <code>true</code>로 설정하십시오. 변수를 <code>false</code> (기본값)로 설정하면 비밀이 멤버 계정에 저장됩니다. • <code>credential_replication_region</code> — Lambda 함수를 배포하려는 AWS 리전과 이메일 템플릿용 S3 버킷 • <code>run_lambda_in_vpc</code> — VPC 내에서 Lambda 함수를 실행하려면 <code>true</code>로 설정합니다. • <code>vpc_id</code>— VPC 내에서 Lambda 함수를 실행하려는 경우 배포 계정의 VPC ID. • <code>vpc_cidr</code>— 배포 계정의 CIDR 범위. • <code>subnet_id</code> — 배포 계정의 서브넷 ID. 	

작업	설명	필요한 기술
	<ul style="list-style-type: none"> • <code>create_smtp_endpoint</code> — 이메일 엔드포인트를 활성화하려면 <code>true</code>로 설정합니다. 	

키 로테이션 확인

작업	설명	필요한 기술
솔루션을 검증합니다.	<ol style="list-style-type: none"> 1. AWS 관리 콘솔에서 배포 계정에 로그인합니다. 2. IAM 콘솔을 열고 사용자 자격 증명 (액세스 키 ID 및 비밀 키) 이 지정된 대로 교체되고 있는지 확인합니다. 3. IAM 키를 교체한 후 다음을 확인하십시오. <ul style="list-style-type: none"> • 이전 값은 AWS Secrets Manager에 저장됩니다. • 시크릿 이름의 형식은 다음과 같습니다 <code>다Account_<account ID>_User_<username>_AccessKey</code> . • <code>admin_email_addresses</code> 파라미터에 지정한 사용자는 키 교체에 대한 이메일 알림을 받습니다. 	DevOps 엔지니어

솔루션 확장

작업	설명	필요한 기술
이메일 알림 날짜를 사용자 지정합니다.	<p>액세스 키를 비활성화하기 전 특정 날짜에 이메일 알림을 보내려는 경우 다음과 같은 변경 사항으로 IAM-access-key-auto-rotation Lambda 함수를 업데이트할 수 있습니다.</p> <ol style="list-style-type: none"> 라는 변수를 정의하십시오. notify-period 키를 비활성화하기 main.py 전에 if 조건을 추가하세요. <pre> If (keyage>rotation-period-notify-period){ send_to_notifier(context, aws_account_id, account_name, resource_owner, resource_actions[resource_owner], dryrun, config.emailTemplateAudit) } </pre>	DevOps 엔지니어

문제 해결

문제	Solution
계정을 account-inventory 나열하는 동안 Lambda 작업이 실패합니다AccessDenied .	이 문제가 발생하면 권한을 검증해야 합니다.

문제	Solution
	<ol style="list-style-type: none"> 1. 새로 벤딩된 계정에 로그인하고 Amazon CloudWatch 콘솔을 연 다음 CloudWatch 로그 그룹을 <code>/aws/lambda/account-inventory-lambda</code> 확인합니다. 2. 최신 CloudWatch 로그에서 액세스 거부 문제를 일으킨 계정 번호를 확인하십시오. 3. AWS Control Tower 관리 계정에 로그인하여 역할이 <code>allow-list-account</code> 생성되었는지 확인합니다. 4. 역할이 없는 경우 명령을 사용하여 Terraform 코드를 다시 실행하십시오. <code>terraform apply</code> 5. 신뢰할 수 있는 계정 탭을 선택하고 동일한 계정이 신뢰할 수 있는지 확인합니다.

관련 리소스

- [테라폼 권장 사례](#) (테라폼 설명서)
- [IAM의 보안 모범 사례](#)(IAM 설명서)
- [키 로테이션 모범 사례](#) (IAM 설명서)

중앙 집중식 로깅 및 다중 계정 보안 가드레일

작성자: Ankush Verma(AWS), Tracy(Pierce) Hickey(AWS)

환경: 프로덕션

기술: 보안, 자격 증명, 규정 준수, 관리 및 거버넌스

AWS 서비스: AWS CloudFormation, AWS Config, 아마존, AWS, 아마존, CloudWatch, AWS CodePipeline, Lambda, GuardDuty, 아마존 마시, AWS 보안 허브, 아마존 S3

요약

이 패턴에서 다루는 접근 방식은 AWS Organizations의 Amazon Web Services(AWS) 계정을 여러 개 보유하고 있으며 AWS Control Tower, 랜딩 존 또는 자동 판매기 서비스를 사용하여 계정에 기본 가드레일을 설정하는 데 어려움을 겪고 있는 고객에게 적합합니다.

이 패턴은 간소화된 다중 계정 아키텍처를 사용하여 체계적인 방식으로 중앙 집중식 로깅과 표준화된 보안 제어를 설정하는 방법을 보여줍니다. AWS CloudFormation 템플릿, CodePipeline, AWS 및 자동화 스크립트의 도움으로 이 설정은 조직에 속한 모든 계정에 배포됩니다.

다중 계정 아키텍처에는 다음과 같은 계정이 포함됩니다.

- 중앙 집중식 로깅 계정 — 모든 가상 사설 클라우드 (VPC) 흐름 로그, AWS CloudTrail 로그, AWS Config 로그 및 다른 모든 계정의 CloudWatch Amazon Logs (구독 사용) 의 모든 로그가 저장되는 계정입니다.
- 상위 보안 계정-여러 계정을 관리하는 다음 보안 서비스의 상위 계정 역할을 하는 계정입니다.
 - 아마존 GuardDuty
 - AWS Security Hub
 - Amazon Macie
 - Amazon Detective
- 하위 계정-조직 내 다른 계정입니다. 이러한 계정은 모든 유용한 로그를 중앙 집중식 로깅 계정에 저장합니다. 하위 계정은 상위 보안 계정에 보안 서비스의 구성원으로 가입합니다.

CloudFormation 템플릿 (첨부) 을 실행하면 중앙 집중식 로깅 계정에 Amazon Simple Storage Service (Amazon S3) 버킷 3개가 프로비저닝됩니다. 하나의 버킷은 모든 계정의 모든 AWS 관련 로그 (예: VPC 흐름 로그의 로그 및 AWS Config) 를 저장하는 데 사용됩니다. CloudTrail 두 번째 버킷은 모든 계정의 CloudFormation 템플릿을 저장하는 데 사용됩니다. 세 번째 버킷은 Amazon S3 액세스 로그를 저장하기 위한 것입니다.

별도의 CloudFormation 템플릿이 AWS를 사용하는 파이프라인을 생성합니다 CodeCommit. 업데이트 된 코드가 CodeCommit 리포지토리로 푸시되면 모든 계정에서 리소스를 시작하고 보안 서비스를 설정합니다. CodeCommit 리포지토리에 업로드될 파일의 파일 구조에 대한 자세한 내용은 README.md 파일 (첨부) 을 참조하십시오.

사전 조건 및 제한 사항

사전 조건

- 모든 계정이 동일한 조직에 가입되어 있는 AWS Organizations 조직 ID.
- Amazon Simple Notification Service(SNS) 알림을 수신하기 위한 활성 이메일 주소.
- 각 계정의 Amazon Simple Storage Service(S3) 버킷에 대한 할당량 확인. 기본적으로 각 계정에는 100개의 S3 버킷이 있습니다. 추가 버킷이 필요한 경우 이 솔루션을 배포하기 전에 할당량 증가를 요청하십시오.

제한 사항

모든 계정은 동일한 조직에 속해야 합니다. AWS Organizations를 사용하지 않는 경우 각 계정의 AWS Identity and Access Management(IAM) 역할에서 액세스할 수 있도록 S3 버킷 정책과 같은 특정 정책을 수정해야 합니다.

참고: 솔루션을 배포하는 동안 Amazon SNS 구독을 확인해야 합니다. 배포 프로세스 중에 제공한 이메일 주소로 확인 메시지가 전송됩니다. 그러면 계정에서 IAM 역할 정책을 생성하거나 수정할 때마다 이러한 경보가 시작되므로 이 이메일 주소로 이메일 알림 메시지가 몇 개 전송됩니다. 배포 프로세스 중에는 이러한 알림 메시지를 무시해도 됩니다.

아키텍처

대상 기술 스택

- 아마존 CloudWatch 알람 및 로그
- AWS CodeCommit 리포지토리

- AWS CodePipeline
- AWS Config
- Amazon Detective
- 아마존 GuardDuty
- IAM 역할 및 권한
- Amazon Macie
- S3 버킷
- AWS Security Hub
- Amazon SNS

대상 아키텍처

1. 보안 서비스를 위해 상위 보안 계정의 하위 계정으로 등록된 기타 계정
2. 상위 계정을 포함한 모든 하위 계정의 보안 조사 결과

리소스

업데이트된 코드가 각 계정 및 AWS 지역의 CodeCommit 리포지토리에 푸시되면 다음 리소스가 자동으로 프로비저닝됩니다.

CloudFormation 스택 1 — 부모 스택 로깅

- 중첩 스택 1-표준 IAM 역할 및 정책
- 중첩 스택 2-계정의 AWS Config 설정
- 중첩 스택 3 — 알람 CloudWatch
 - SecurityGroupChangesAlarm
 - UnauthorizedAttemptAlarm
 - RootActivityAlarm
 - NetworkAclChangesAlarm

- IAM UserManagementAlarm
- IAM PolicyChangesAlarm
- CloudTrailChangeAlarm
- IAM CreateAccessKeyAlarm
- CloudTrail 로그에서 메트릭을 생성하고 이를 경보에 사용하기 위한 메트릭 필터
- SNS 주제

CloudFormation 스택 2 — 부모 가드레일 스택

- 중첩 스택 1-계정 암호 정책 설정을 위한 AWS Lambda 함수
- 중첩 스택 2-기본 AWS Config 규칙
 - CIS- SecurityGroupsMustRestrictSshTraffic
 - OpenSecurityGroupRuleCheck 보안 그룹 규칙 평가를 위한 Lambda 함수와 함께 사용
 - checkec2- for-required-tag
 - check-for-unrestricted-ports

CloudFormation 스택 3 — 로그 내보내기 CloudWatch

- Amazon Kinesis 구독을 사용하여 CloudWatch 로그 그룹에서 Amazon S3로 로그 내보내기

도구

- [AWS CloudFormation](#) — CloudFormation AWS는 템플릿을 사용하여 모든 AWS 지역 및 계정에서 애플리케이션에 필요한 모든 리소스를 자동화되고 안전한 방식으로 모델링하고 프로비저닝합니다.
- [Amazon CloudWatch](#) — Amazon은 AWS 리소스와 AWS에서 실행하는 애플리케이션을 실시간으로 CloudWatch 모니터링합니다. 리소스와 애플리케이션에 대해 측정할 수 있는 변수인 지표를 수집하고 추적하는 데 사용할 CloudWatch 수 있습니다.
- [AWS CodeCommit](#) — CodeCommit AWS는 AWS에서 호스팅하는 버전 제어 서비스입니다. 를 CodeCommit 사용하여 클라우드에서 자산 (예: 문서, 소스 코드, 바이너리 파일) 을 비공개로 저장하고 관리할 수 있습니다.
- [AWS CodePipeline](#) — CodePipeline AWS는 소프트웨어 릴리스에 필요한 단계를 모델링, 시각화 및 자동화하는 데 사용할 수 있는 지속적 전송 서비스입니다.

- [AWS Config](#) - AWS Config는 사용자의 AWS 계정에서 AWS 리소스의 구성을 상세하게 볼 수 있도록 합니다. 이러한 보기에는 리소스 간에 어떤 관계가 있는지와 리소스가 과거에 어떻게 구성되었는지도 포함되므로, 시간이 지나면서 구성과 관계가 어떻게 변하는지 확인할 수 있습니다.
- [Amazon Detective](#)-Amazon Detective는 사용자가 보안 조사 결과 또는 의심스러운 활동의 근본 원인을 분석 및 조사하고 신속하게 파악하는 데 도움이 됩니다. Detective는 AWS 리소스에서 로그 데이터를 자동으로 수집합니다. 그런 다음 기계 학습, 통계 분석, 그래프 이론을 사용하여 더 빠르고 효율적으로 보안 조사를 수행할 수 있도록 시각화를 지원합니다.
- [Amazon GuardDuty](#) — GuardDuty Amazon은 흐름 로그, CloudTrail 관리 이벤트 로그, CloudTrail 데이터 이벤트 로그 및 도메인 이름 시스템 (DNS) 로그를 분석하고 처리하는 지속적인 보안 모니터링 서비스입니다. 악성 IP 주소 및 도메인 목록 등 위협 인텔리전스 피드와 기계 학습을 사용하여 AWS 환경에서 예기치 않게 발생하는 잠재적 무단 활동과 악의적 활동을 찾아냅니다.
- [AWS Identity and Access Management](#)-AWS Identity and Access Management(IAM)는 AWS 리소스에 대한 사용자의 액세스를 안전하게 제어하는 데 도움이 되는 웹 서비스입니다. IAM을 사용하여 리소스를 사용하도록 인증(로그인) 및 권한 부여(권한 있음)된 대상을 제어합니다.
- [Amazon Macie](#)-Amazon Macie는 개인 식별 정보(PII) 및 재무 데이터와 같은 민감한 데이터의 검색을 자동화하여 조직이 Amazon S3에 저장하는 데이터를 더 잘 이해할 수 있도록 합니다.
- [Amazon S3](#)-Amazon Simple Storage Service(S3)는 웹 사이트, 모바일 애플리케이션, 백업, 데이터 레이크 등 다양한 스토리지 솔루션에 사용할 수 있는 확장성이 뛰어난 객체 스토리지 서비스입니다.
- [AWS Security Hub](#)-AWS Security Hub는 AWS의 보안 상태를 포괄적으로 파악하고 보안 표준 및 모범 사례와 비교하여 사용자의 환경을 확인하는 데 도움이 됩니다.
- [Amazon SNS](#)-Amazon Simple Notification Service(SNS)는 게시자에서 구독자(생산자 및 소비자라고도 함)로 메시지를 전송하는 관리형 서비스입니다.

에픽

1단계: 모든 계정에서 IAM 역할 설정

작업	설명	필요한 기술
childAccount_IAM_Role_All_Accounts.yaml 템플릿을 실행하여 us-east-1 리전에 CloudFormation IAM 역할을 생성하십시오.	필요한 IAM 역할과 권한을 생성하려면 us-east-1 리전의 각 계정(중앙 집중식 로깅 계정, 상위 보안 계정 및 조직의 다른 모든 AWS 계정)에서 이 템플릿을 하나씩 수동으로 실행	클라우드 아키텍트

작업	설명	필요한 기술
	<p>행해야 합니다. Childaccount_IAM_role_All_Accounts.yaml 템플릿은 패키지의 /templates/initial_deployment_templates 디렉터리에 있습니다. IAM 역할은 나머지 아키텍처를 프로비저닝하고 설정하기 위해 API를 호출할 때 사용됩니다. 파라미터로 전달되는 IAM 역할의 이름이 모든 계정에서 일치하는지 확인합니다.</p>	
<p>템플릿 파라미터에서 IAM 역할의 이름을 입력합니다.</p>	<p>부모 보안 계정에서 다른 모든 자녀 계정에서 맡을 수 있는 IAM 역할을 제공하십시오. CodeBuild 기본 역할 이름은 security_execute_child_stack_role 입니다.</p>	클라우드 아키텍트
<p>파라미터에 상위 보안 계정의 계정 ID를 입력합니다.</p>	<p>부모 보안 계정은 CodeBuild 실행되는 계정입니다.</p>	클라우드 아키텍트

2단계: 중앙 집중식 로깅 계정에서 S3 버킷 설정

작업	설명	필요한 기술
<p>us-east-1의 중앙 집중식 로깅 계정에서 S3Bucket 중앙 집중식 .yaml 템플릿을 실행합니다. LoggingAccount CloudFormation</p>	<p>중앙 집중식 로깅 계정에서 S3 버킷을 생성하려면 S3Buckets-Centralized-LoggingAccount .yaml 을 실행합니다. 템플릿은 패키지의 /templates/initial_deploye</p>	클라우드 아키텍트

작업	설명	필요한 기술
	<p>nt_templates 디렉터리에 있습니다. S3 버킷은 모든 로그, 템플릿 및 Amazon S3 액세스 로그를 저장합니다. 다음 단계에서 파라미터 파일을 수정하는 데 사용할 모든 S3 버킷 이름을 기록해 둡니다.</p>	
<p>템플릿 파라미터에 AWS 로그 스토리지용 S3 버킷의 이름을 입력합니다.</p>	<p>S3 Bucket Name for Centralized Logging in Logging Account 파라미터의 이름을 입력합니다. 이 버킷은 모든 계정의 흐름 로그 및 로그와 같은 AWS CloudTrail 로그를 저장하는 중앙 위치 역할을 합니다. 버킷 이름과 Amazon 리소스 이름 (ARN)을 기록해 둡니다.</p>	클라우드 아키텍트
<p>액세스 로그 저장을 위한 S3 버킷의 이름을 입력합니다.</p>	<p>S3 Bucket Name for Access Logs in Logging Account 파라미터에 S3 버킷 이름을 입력합니다. 이 S3 버킷은 Amazon S3에 대한 액세스 로그를 저장합니다.</p>	클라우드 아키텍트
<p>템플릿 저장을 위한 S3 버킷의 이름을 입력합니다.</p>	<p>S3 Bucket Name for CloudFormation Template storage in Logging Account 파라미터에 S3 버킷 이름을 입력합니다.</p>	클라우드 아키텍트

작업	설명	필요한 기술
조직 ID를 입력합니다.	조직에서 S3 버킷에 대한 액세스를 제공하려면 Organization Id for Non-AMS accounts 파라미터에 조직의 ID를 입력합니다.	클라우드 아키텍트

3단계: 상위 보안 계정에 CI/CD 인프라 배포

작업	설명	필요한 기술
security-guard-rails-codepipeline-중앙 SecurityAccount CloudFormation 집중식-.yml 템플릿을 실행합니다.	CI/CD 파이프라인을 배포하려면 us-east-1의 상위 보안 계정에서 security-guard-rails-codepipeline-Centralized-SecurityAccount.yml 템플릿을 수동으로 실행합니다. 템플릿은 패키지의 /templates/initial_deployment_templates 디렉터리에 있습니다. 이 파이프라인은 모든 하위 계정에 모든 인프라를 배포합니다.	클라우드 아키텍트
중앙 집중식 로깅 계정에 템플릿을 저장할 S3 버킷의 이름을 입력합니다.	2단계에서 S3 Bucket Name for the CloudFormation Template storage in Logging Account 파라미터에 입력한 S3 버킷의 이름을 입력합니다.	클라우드 아키텍트
하위 계정에서 사용할 IAM 역할의 이름을 입력합니다.	1단계에서 Name of the IAM role 파라미터에 입력한 이름을 입력합니다.	클라우드 아키텍트

작업	설명	필요한 기술
실패 알림을 받을 CodePipeline 수 있는 활성 이메일 주소를 제공하십시오.	CodePipeline 실패 알림 및 기타 CloudWatch 알람 관련 알림을 수신하는 데 사용할 이메일 주소를 입력합니다.	클라우드 아키텍트

4단계: 계정 정보를 포함하도록 파일 업데이트

작업	설명	필요한 기술
Accountlist.json을 수정합니다.	패키지의 최상위 수준에 있는 Accountlist.json 파일에 상위 보안 계정 번호와 하위 계정 번호를 추가합니다. ChildAccountList 필드에는 상위 보안 계정 번호도 포함된다는 점에 유의하십시오. 패키지의 deployment-instructions.md 파일에 있는 예를 참조하십시오.	클라우드 아키텍트
accounts.csv를 수정합니다.	패키지의 최상위 수준에 있는 accounts.csv 파일에서 계정에 등록된 이메일과 함께 모든 하위 계정을 추가합니다. deployment-instructions.md 파일의 예제를 참조하십시오.	클라우드 아키텍트
parameters.config를 수정합니다.	/templates 폴더에 있는 parameters.config 파일에서 다음과 같은 6개의 파라미터를 업데이트합니다.	클라우드 아키텍트

작업	설명	필요한 기술
	<ul style="list-style-type: none"> • pNotifyEmail : 파이프라인을 설정할 때 입력한 이메일 주소(3단계 참조) • pstackNameLogging : 중앙 로깅을 위한 CloudFormation 스택 이름 • pS3LogsBucket : 모든 계정의 로그가 저장되는 S3 버킷의 이름(2단계 참조) • pBucketName : 로그를 저장하는 데 사용하는 S3 버킷의 ARN • pTemplateBucketName : 템플릿을 저장할 S3 버킷의 이름(2단계 참조) • pAllowedAccounts : 상위 계정 및 하위 계정의 계정 ID <p>다른 파라미터의 경우 기본값을 유지할 수 있습니다. 예제는 패키지의 deployment-instructions.md 파일을 참조하십시오.</p>	

5단계: CodeCommit 리포지토리에 액세스하여 업데이트된 파일을 푸시합니다.

작업	설명	필요한 기술
<p>3단계에서 만든 CodeCommit 리포지토리에 액세스합니다.</p>	<p>CI/CD 인프라 CloudFormation 스택 (3단계에서 시작) 의 출력 섹션에서 리포지토리 URL의 CodeCommit 이름을 기록해 둡</p>	<p>클라우드 아키텍트</p>

작업	설명	필요한 기술
	<p>니다. 모든 대상 계정에 인프라를 배포하기 위해 파일을 푸시할 수 있도록 리포지토리에 대한 액세스 권한을 생성합니다. 자세한 내용은 내용은 AWS 설정을 참조하십시오 CodeCommit.</p>	
<p>파일을 CodeCommit 리포지토리로 푸시하십시오.</p>	<p>시스템에 Git을 설치합니다. 그런 다음 Git 명령을 실행하여 빈 리포지토리를 복제하고 노트북에서 리포지토리 폴더로 파일을 복사한 다음 아티팩트를 리포지토리로 푸시합니다. 패키지의 deployment-instructions.md 파일에서 샘플 Git 명령을 확인합니다. 기본 Git 명령은 관련 리소스 섹션을 참조하십시오.</p>	<p>클라우드 아키텍트</p>

6단계: 확인 CodePipeline 및 CodeBuild 상태

작업	설명	필요한 기술
<p>CodePipeline 및 상태를 확인합니다 CodeBuild.</p>	<p>아티팩트를 CodeCommit 리포지토리로 푸시한 후 3단계에서 만든 CodePipeline 파이프라인이 시작되었는지 확인합니다. 그런 다음 CodeBuild 로그를 확인하여 상태 또는 오류를 확인합니다.</p>	<p>클라우드 아키텍트</p>

관련 리소스

- [AWS CloudFormation 템플릿 배포](#)

- [AWS를 사용하기 위해 설정하기 CodeCommit](#)
- [S3 버킷에 파일 업로드](#)
- [기본 Git 명령](#)

첨부

이 문서와 관련된 추가 콘텐츠에 액세스하려면 [attachment.zip](#) 파일의 압축을 풉니다.

Amazon CloudFront 배포판에서 액세스 로깅, HTTPS 및 TLS 버전을 확인하십시오.

작성자: SaiJeevan 데비레디 (AWS)

환경: 프로덕션	기술: 콘텐츠 전송, 보안, 자격 증명, 규정 준수	워크로드: 기타 모든 워크로드
AWS 서비스: 아마존 SNS, AWS, 아마존 CloudFormation CloudWatch, AWS Lambda		

요약

이 패턴은 Amazon CloudFront 배포를 검사하여 HTTPS를 사용하고, 전송 계층 보안 (TLS) 버전 1.2 이상을 사용하며, 액세스 로깅이 활성화되어 있는지 확인합니다. CloudFront Amazon Web Services (AWS) 에서 제공하는 서비스로서, .html, .css, .js, 이미지 파일 등의 정적 및 동적 웹 콘텐츠를 사용자에게 빠르게 배포할 수 있도록 합니다. CloudFront 엣지 로케이션이라고 하는 전 세계 데이터 센터 네트워크를 통해 콘텐츠를 전송합니다. 서비스를 제공하는 콘텐츠를 사용자가 요청하면 지연 시간 (지연 시간) 이 가장 낮은 엣지 로케이션으로 CloudFront 요청이 라우팅되므로 콘텐츠가 최상의 성능으로 전달됩니다.

이 패턴은 CloudWatch Amazon Events에서 API 호출, 또는 을 CloudFront 탐지할 때 시작되는 AWS Lambda 함수를 제공합니다. [CreateDistributionCreateDistributionWithTagsUpdateDistribution](#) Lambda 함수의 사용자 지정 로직은 AWS 계정에서 생성되거나 업데이트된 CloudFront 모든 배포를 평가합니다. 다음 위반이 감지되면 Amazon Simple Notification Service(SNS)를 사용하여 위반 알림을 보냅니다.

- 글로벌 검사:
 - 사용자 지정 인증서가 TLS 버전 1.2를 사용하지 않습니다.
 - 배포 시 로깅이 비활성화되어 있습니다.
- 원본 검사:
 - 원본이 TLS 버전 1.2로 구성되어 있지 않습니다.
 - 원본과의 통신이 HTTPS 이외의 프로토콜에서 허용됩니다.

- 동작 검사:
 - 기본 동작 통신이 HTTPS 이외의 프로토콜에서 허용됩니다.
 - 사용자 지정 동작 통신이 HTTPS 이외의 프로토콜에서 허용됩니다.

사전 조건 및 제한 사항

사전 조건

- 활성 상태의 AWS 계정
- 위반 알림을 받으려는 이메일 주소

제한 사항

- 이 보안 제어는 배포를 업데이트하지 않는 한 기존 Cloudfront 배포를 확인하지 않습니다.
- CloudFront 글로벌 서비스로 간주되며 특정 AWS 지역에 국한되지 않습니다. 하지만 글로벌 서비스를 위한 Amazon CloudWatch Logs와 AWS Cloudtrail API 로깅은 미국 동부 (버지니아 북부) 지역 ()에서 발생합니다. us-east-1 따라서 이 보안 제어 양식을 배포하고 유지 CloudFront 관리해야 합니다. us-east-1 이 단일 배포는 에 대한 CloudFront 모든 배포를 모니터링합니다. 다른 AWS 리전에 보안 제어를 배포하지 마세요. (다른 지역에 배포하면 CloudWatch Events 및 Lambda 함수를 시작하지 못하고 SNS 알림이 표시되지 않습니다.)
- 이 솔루션은 CloudFront 웹 콘텐츠 배포를 통한 광범위한 테스트를 거쳤습니다. 실시간 메시징 프로토콜(RTMP) 스트리밍 배포는 다루지 않습니다.

아키텍처

대상 기술 스택

- Lambda 함수
- SNS 주제
- 아마존 EventBridge 룰

대상 아키텍처

자동화 및 규모 조정

- AWS Organizations를 사용하는 경우, [AWS StackSets Cloudformation](#)을 사용하여 모니터링하려는 여러 계정에 첨부된 템플릿을 배포할 수 있습니다.

도구

서비스

- [AWS CloudFormation](#) — CloudFormation 인프라를 코드로 사용하여 AWS 리소스를 모델링하고 설정하는 데 도움이 되는 서비스입니다.
- [Amazon EventBridge](#) — 자체 애플리케이션, 서비스형 소프트웨어 (SaaS) 애플리케이션 및 AWS 서비스로부터 실시간 데이터 스트림을 EventBridge 전달하여 해당 데이터를 Lambda 함수와 같은 대상으로 라우팅합니다.
- [AWS Lambda](#)-Lambda는 서버를 프로비저닝하거나 관리하지 않고도 코드를 실행할 수 있습니다.
- [Amazon S3](#)-Amazon Simple Storage Service(S3)는 웹 사이트, 모바일 애플리케이션, 백업, 데이터 레이크 등 다양한 스토리지 솔루션에 사용할 수 있는 확장성이 뛰어난 객체 스토리지 서비스입니다.
- [Amazon SNS](#)-Amazon SNS는 게시자와 클라이언트 간에 웹 서버와 이메일 주소를 포함한 메시지 전달 또는 전송을 조정하고 관리합니다. 구독자는 구독하는 주제에 게시된 모든 메시지를 수신하며 주제에 대한 모든 구독자는 동일한 메시지를 수신합니다.

코드

첨부된 코드에는 다음이 포함됩니다.

- Lambda 코드(index.py)를 포함한 .zip 파일
- Lambda 코드를 배포하기 위해 실행하는 CloudFormation 템플릿 (.yml 파일)

에픽

보안 제어 업로드

작업	설명	필요한 기술
Lambda 코드용 S3 버킷을 생성합니다.	Amazon S3 콘솔에서 앞에 슬래시를 포함하지 않는 고유한 이름을 가진 S3 버킷을 생성함	클라우드 아키텍트

작업	설명	필요한 기술
	니다. S3 버킷 이름은 전역 수준에서 고유하며, 네임스페이스는 모든 AWS 계정이 공유합니다. S3 버킷은 Lambda 코드를 배포하려는 리전에 있어야 합니다.	
Lambda 코드를 S3 버킷에 업로드	첨부 파일 섹션에 제공된 Lambda 코드(<code>cloudfront_ssl_log_lambda.zip</code> 파일)를 이전 단계에서 생성한 S3 버킷에 업로드합니다.	클라우드 아키텍트

템플릿 CloudFormation 배포

작업	설명	필요한 기술
CloudFormation 템플릿을 배포합니다.	AWS CloudFormation 콘솔의 S3 버킷과 동일한 AWS 리전에 첨부 파일 섹션에 제공된 CloudFormation 템플릿(<code>cloudfront-ssl-logging.yml</code>)을 배포합니다.	클라우드 아키텍트
S3 버킷 이름을 지정합니다.	S3 Bucket 파라미터에 대해 첫 번째 에픽에서 생성한 S3 버킷의 이름을 지정합니다.	클라우드 아키텍트
Lambda 파일의 Amazon S3 키 이름을 지정합니다.	S3 키 파라미터의 경우 S3 버킷에 있는 Lambda 코드 .zip 파일의 Amazon S3 위치를 지정합니다. 선행 슬래시를 포함하지 마세요(예를 들어 <code>lambda.zip</code> 또는 <code>controls/lambda.zip</code> 은 입력 가능합니다).	클라우드 아키텍트

작업	설명	필요한 기술
알림 이메일 주소를 제공합니다.	알림 이메일 파라미터의 경우 위반 알림을 받고자 하는 이메일 주소를 제공합니다.	클라우드 아키텍트
로깅 수준을 정의합니다.	<p>Lambda 로깅 수준 파라미터의 경우 Lambda 함수의 로깅 수준을 정의합니다. 다음 값 중 하나를 선택합니다.</p> <ul style="list-style-type: none"> • INFO 애플리케이션 진행 상황에 대한 자세한 정보 메시지를 받을 수 있습니다. • ERROR 여전히 애플리케이션을 계속 실행할 수 있는 오류 이벤트에 대한 정보를 받을 수 있습니다. • WARNING 잠재적으로 위험한 상황에 대한 정보를 받을 수 있습니다. 	클라우드 아키텍트

구독 확인

작업	설명	필요한 기술
구독을 확인합니다.	CloudFormation 템플릿이 성공적으로 배포되면 새 SNS 주제가 생성되고 입력한 이메일 주소로 구독 메시지가 전송됩니다. 위반 알림을 받으려면 이 이메일 구독을 확인해야 합니다.	클라우드 아키텍트

관련 리소스

- [AWS CloudFormation 정보](#)
- [AWS CloudFormation 콘솔에서 스택 생성](#) (CloudFormation 설명서)
- [CloudFront 로깅](#) (CloudFront 설명서)
- [Amazon S3 정보](#)
- [AWS Lambda 정보](#)

첨부

이 문서와 관련된 추가 콘텐츠에 액세스하려면 [attachment.zip](#) 파일의 압축을 풉니다.

IPv4 및 IPv6용 보안 그룹 수신 규칙에서 단일 호스트 네트워크 항목 확인

작성자: SaiJeevan 데비레디 (AWS), 가네쉬 쿠마르 (AWS), 존 레이놀즈 (AWS)

환경: 프로덕션

기술: 네트워킹; 보안, 자격 증명, 규정 준수

AWS 서비스: 아마존 SNS, AWS, 아마존 CloudFormation CloudWatch, AWS Lambda, 아마존 VPC

요약

이 패턴은 Amazon Web Services(AWS) 리소스가 사양을 충족하지 않을 경우 알려주는 보안 제어 기능을 제공합니다. Internet Protocol 버전 4(IPv4) 및 IPv6 보안 그룹 소스 주소 필드 모두에서 단일 호스트 네트워크 항목을 찾는 AWS Lambda 함수를 제공합니다. Lambda 함수는 아마존 CloudWatch 이벤트가 아마존 Elastic Compute Cloud (아마존 EC2) API 호출을 감지할 때 시작됩니다. [AuthorizeSecurityGroupIngress](#) Lambda 함수의 사용자 지정 로직은 보안 그룹 수신 규칙의 CIDR 블록의 서브넷 마스크를 평가합니다. 서브넷 마스크가 /32(IPv4) 또는 /128(IPv6) 이외의 것으로 확인되면 Lambda 함수는 Amazon Simple Notification Service(Amazon SNS)를 사용하여 위반 알림을 보냅니다.

사전 조건 및 제한 사항

사전 조건

- 활성 상태의 AWS 계정
- 위반 알림을 받으려는 이메일 주소

제한 사항

- 이 보안 모니터링 솔루션은 리전과 관련이 있으므로 모니터링하고자 하는 각 AWS 리전에 배포해야 합니다.

아키텍처

대상 기술 스택

- Lambda 함수
- SNS 주제
- 아마존 EventBridge 룰

대상 아키텍처

자동화 및 규모 조정

- AWS Organizations를 사용하는 경우, [AWS StackSets Cloudformation](#)을 사용하여 모니터링하려는 여러 계정에 이 템플릿을 배포할 수 있습니다.

도구

서비스

- [CloudFormation](#)은 인프라를 코드로 사용하여 AWS 리소스를 모델링하고 설정하는 데 도움이 되는 서비스입니다.
- [EventBridge](#)은 자체 애플리케이션, 서비스형 소프트웨어 (SaaS) 애플리케이션 및 AWS 서비스로부터 실시간 데이터 스트림을 전달하고 이 데이터를 Lambda 함수와 같은 대상으로 라우팅합니다.
- [AWS Lambda](#)는 서버를 프로비저닝하거나 관리하지 않고도 코드를 실행할 수 있도록 지원합니다.
- [Amazon S3](#)-Amazon Simple Storage Service(S3)는 웹 사이트, 모바일 애플리케이션, 백업, 데이터 레이크 등 다양한 스토리지 솔루션에 사용할 수 있는 확장성이 뛰어난 객체 스토리지 서비스입니다.
- [Amazon SNS](#)는 웹 서버 및 이메일 주소를 포함하여 게시자와 클라이언트 간에 메시지 전달 또는 전송을 조정 및 관리합니다. 구독자는 구독하는 주제에 게시된 모든 메시지를 수신하며 주제에 대한 모든 구독자는 동일한 메시지를 수신합니다.

코드

첨부된 코드에는 다음이 포함됩니다.

- Lambda 보안 제어 코드(index.py)가 포함된 .zip 파일
- Lambda 코드를 배포하기 위해 실행하는 CloudFormation 템플릿 (security-control.yml파일)

에픽

보안 제어 업로드

작업	설명	필요한 기술
Lambda 코드용 S3 버킷을 생성합니다.	Amazon S3 콘솔 에서 앞에 슬래시를 포함하지 않는 고유한 이름을 가진 S3 버킷을 생성합니다. S3 버킷 이름은 전역 수준에서 고유하며, 네임스페이스는 모든 AWS 계정이 공유합니다. S3 버킷은 보안 그룹 수신 확인을 배포하고자 하는 AWS 리전에 있어야 합니다.	클라우드 아키텍트
Lambda 코드를 S3 버킷에 업로드	Attachments 섹션에 제공된 Lambda 코드(security-control-lambda.zip 파일)를 이전 단계에서 생성한 S3 버킷에 업로드합니다.	클라우드 아키텍트

템플릿 배포 CloudFormation

작업	설명	필요한 기술
Python 버전을 변경합니다.	첨부 파일 섹션에 제공된 CloudFormation 템플릿 (security-control.yml) 을 다운로드합니다. 파일을 열고 Lambda에서 지원하는 최신 버전(현재 Python 3.9)을	클라우드 아키텍트

작업	설명	필요한 기술
	<p>반영하도록 Python 버전을 수정합니다.</p> <p>예를 들어, 코드에서 python 을 검색하고 Runtime의 값을 python3.6 에서 python3.9 로 변경할 수 있습니다.</p> <p>Python 런타임 버전 지원에 대한 최신 정보는 AWS Lambda 설명서를 참조하세요.</p>	
AWS CloudFormation 템플릿을 배포합니다.	AWS CloudFormation 콘솔의 S3 버킷과 동일한 AWS 리전에 CloudFormation 템플릿 (security-control.yml)을 배포합니다.	클라우드 아키텍트
S3 버킷 이름을 지정합니다.	S3 Bucket 파라미터에 대해 첫 번째 에픽에서 생성한 S3 버킷의 이름을 지정합니다.	클라우드 아키텍트
Lambda 파일의 Amazon S3 키 이름을 지정합니다.	S3 Key 파라미터의 경우, S3 버킷에 있는 Lambda 코드 .zip 파일의 Amazon S3 위치를 지정합니다. 앞에 슬래시를 포함하지 않아야 합니다. (예: lambda.zip 또는 controls/lambda.zip 를 입력할 수 있음)	클라우드 아키텍트
알림 이메일 주소를 제공합니다.	알림 이메일 파라미터의 경우 위반 알림을 받고자 하는 이메일 주소를 제공합니다.	클라우드 아키텍트

작업	설명	필요한 기술
로깅 수준을 정의합니다.	<p>Lambda 로깅 수준 파라미터의 경우 Lambda 함수의 로깅 수준을 정의합니다. 다음 값 중 하나를 선택합니다.</p> <ul style="list-style-type: none"> • INFO 애플리케이션 진행 상황에 대한 자세한 정보 메시지를 받을 수 있습니다. • ERROR 여전히 애플리케이션을 계속 실행할 수 있는 오류 이벤트에 대한 정보를 받을 수 있습니다. • WARNING 잠재적으로 위험한 상황에 대한 정보를 받을 수 있습니다. 	클라우드 아키텍트

구독 확인

작업	설명	필요한 기술
구독을 확인합니다.	<p>CloudFormation 템플릿이 성공적으로 배포되면 새 SNS 주제가 생성되고 입력한 이메일 주소로 구독 메시지가 전송됩니다. 위반 알림을 받으려면 이 이메일 구독을 확인해야 합니다.</p>	클라우드 아키텍트

관련 리소스

- [AWS CloudFormation 정보](#)
- [AWS CloudFormation 콘솔에서 스택 생성](#) (AWS CloudFormation 설명서)
- [VPC의 보안 그룹](#)(Amazon VPC 설명서)
- [Amazon S3 정보](#)

- [AWS Lambda 정보](#)

첨부

이 문서와 관련된 추가 콘텐츠에 액세스하려면 [attachment.zip](#) 파일의 압축을 풉니다.

엔터프라이즈 애플리케이션을 위한 Amazon Cognito 인증 흐름 선택

작성자: 마이클 다허트 (AWS) 와 파비안 안케 (AWS)

환경: 프로덕션

기술: 보안, 자격 증명, 규정 준수

AWS 서비스: Amazon Cognito

요약

[Amazon Cognito](#)는 웹 및 모바일 애플리케이션을 위한 인증, 권한 부여 및 사용자 관리를 제공합니다. 페더레이션 ID 인증에 유용한 기능을 제공합니다. 이를 설치하고 실행하려면 기술 설계자가 해당 기능을 어떻게 사용할지 결정해야 합니다.

Amazon Cognito는 인증 요청을 위한 여러 흐름을 지원합니다. 이러한 흐름은 사용자가 자신의 ID를 확인할 수 있는 방법을 정의합니다. 사용할 인증 흐름은 애플리케이션의 특정 요구 사항에 따라 결정되며 복잡해질 수 있습니다. 이 패턴은 엔터프라이즈 애플리케이션에 가장 적합한 인증 흐름을 결정하는 데 도움이 됩니다. 여기서는 사용자가 Amazon Cognito, OpenID Connect (OIDC) 및 페더레이션에 대한 기본 지식을 이미 알고 있다고 가정하고 다양한 페더레이션된 인증 흐름에 대한 세부 정보를 안내합니다.

이 솔루션은 기술 의사 결정권자를 대상으로 합니다. 다양한 인증 흐름을 이해하고 이를 애플리케이션 요구 사항에 매핑하는 데 도움이 됩니다. 기술 책임자는 Amazon Cognito 통합을 시작하는 데 필요한 통찰력을 수집해야 합니다. 엔터프라이즈 조직은 주로 SAML 페더레이션에 중점을 두기 때문에 이 패턴에는 SAML 페더레이션을 사용하는 [Amazon Cognito 사용자](#) 풀에 대한 설명이 포함됩니다.

사전 조건 및 제한 사항

사전 조건

- 활성 상태의 AWS 계정.
- AWS 자격 증명 및 액세스 관리 (IAM) 역할 및 권한 (Amazon Cognito에 대한 전체 액세스 권한 포함)
- (선택 사항) Microsoft Entra ID, 액티브 디렉터리 페더레이션 서비스 (ADFS) 또는 Okta와 같은 ID 공급자 (IdP) 에 대한 액세스
- 애플리케이션에 대한 높은 수준의 전문 지식
- 아마존 코그니토, OpenID Connect (OIDC) 및 페더레이션에 대한 기본 지식

제한 사항

- 이 패턴은 Amazon Cognito 사용자 풀과 자격 증명 공급자에 초점을 맞춥니다. Amazon Cognito 자격 증명 풀에 대한 자세한 내용은 [추가 정보](#) 섹션을 참조하십시오.

아키텍처

다음 표를 참조하면 인증 흐름을 선택하는 데 도움이 됩니다. 각 흐름에 대한 자세한 내용은 이 섹션에 나와 있습니다.

machine-to-machine 인증이 필요한가요?	앱이 서버에서 프론트엔드를 렌더링하는 웹 기반 애플리케이션인가요?	앱이 SPA (단일 페이지 애플리케이션)인가요 아니면 모바일 기반 프론트엔드 애플리케이션인가요?	애플리케이션에 '로그인 사용자' 기능을 사용하려면 새로 고침 토큰이 필요한가요?	프론트엔드는 브라우저 기반 리디렉션 메커니즘을 제공하나요?	권장 아마존 Cognito 플로우
예	아니요	아니요	아니요	아니요	클라이언트 자격 증명 흐름
아니요	예	아니요	예	예	인증 코드 흐름
아니요	아니요	예	예	예	PKCE (코드 교환용 증명 키 포함) 인증 코드 흐름
아니요	아니요	아니요	아니요	아니요	리소스 소유자 암호 흐름*

* 리소스 소유자 암호 흐름은 꼭 필요한 경우에만 사용해야 합니다. 자세한 내용은 이 패턴의 리소스 소유자 비밀번호 흐름 섹션을 참조하십시오.

클라이언트 자격 증명 흐름

클라이언트 자격 증명 흐름은 Amazon Cognito 흐름 중 가장 짧은 흐름입니다. 시스템 또는 서비스가 사용자 상호 작용 없이 서로 통신하는 경우 사용해야 합니다. 요청 시스템은 클라이언트 ID와 클라이언트 암호를 사용하여 액세스 토큰을 검색합니다. 두 시스템 모두 사용자 개입 없이 작동하므로 추가 동의 단계가 필요하지 않습니다.

다이어그램은 다음을 보여 줍니다.

1. 애플리케이션 1이 클라이언트 ID 및 클라이언트 암호와 함께 Amazon Cognito 엔드포인트로 인증 요청을 보내고 액세스 토큰을 검색합니다.
2. 애플리케이션 1은 이후에 애플리케이션 2를 호출할 때마다 이 액세스 토큰을 사용합니다.
3. 애플리케이션 2는 Amazon Cognito를 사용하여 액세스 토큰을 검증합니다.

다음 흐름을 사용해야 합니다.

- 사용자 상호 작용이 없는 애플리케이션 간 통신용

이 흐름은 사용해서는 안 됩니다.

- 사용자 상호 작용이 가능한 모든 커뮤니케이션용

인증 코드 흐름

인증 코드 흐름은 클래식 웹 기반 인증을 위한 것입니다. 이 흐름에서는 백엔드가 모든 토큰 교환 및 저장을 처리합니다. 브라우저 기반 클라이언트는 실제 토큰을 볼 수 없습니다. 이 솔루션은 .NET Core, Jakarta Faces 또는 JSP (자카르타 서버 페이지) 와 같은 프레임워크로 작성된 애플리케이션에 사용됩니다.

인증 코드 흐름은 리디렉션 기반 흐름입니다. 클라이언트는 웹 브라우저 또는 유사한 클라이언트와 상호 작용할 수 있어야 합니다. 클라이언트는 인증 서버로 리디렉션되고 이 서버에 대해 인증됩니다. 클라이언트가 성공적으로 인증되면 다시 서버로 리디렉션됩니다.

다이어그램은 다음을 보여 줍니다.

1. 클라이언트가 웹 서버에 요청을 보냅니다.

2. 웹 서버는 HTTP 302 상태 코드를 사용하여 클라이언트를 Amazon Cognito로 리디렉션합니다. 클라이언트는 이 리디렉션을 자동으로 따라 구성된 IdP 로그인으로 이동합니다.
3. IdP는 IdP 측에서 기존 브라우저 세션을 확인합니다. 존재하지 않는 경우 사용자는 사용자 이름과 비밀번호를 제공하여 인증을 요청합니다. IdP는 SAML 토큰으로 아마존 코그니토에 응답합니다.
4. Amazon Cognito는 JSON 웹 토큰 (JWT), 특히 코드 토큰을 사용하여 성공을 거두었습니다. 웹 서버는 /oauth2/token을 호출하여 코드 토큰을 액세스 토큰으로 교환합니다. 웹 서버는 검증을 위해 클라이언트 ID와 클라이언트 암호를 Amazon Cognito로 전송합니다.
5. 액세스 토큰은 이후에 다른 애플리케이션을 호출할 때마다 사용됩니다.
6. 다른 애플리케이션은 Amazon Cognito로 액세스 토큰을 검증합니다.

다음 흐름을 사용해야 합니다.

- 사용자가 웹 브라우저 또는 클라이언트와 상호 작용할 수 있는 경우 브라우저에 비밀이 노출되지 않도록 애플리케이션 코드가 서버에서 실행되고 렌더링됩니다.

이 흐름은 사용해서는 안 됩니다.

- 단일 페이지 애플리케이션 (SPA) 또는 모바일 앱은 클라이언트에서 렌더링되므로 클라이언트 암호를 사용해서는 안 되기 때문입니다.

PKCE를 사용한 인증 코드 흐름

단일 페이지 애플리케이션과 모바일 애플리케이션에는 PKCE (코드 교환용 증명 키)가 포함된 인증 코드 흐름을 사용해야 합니다. 암시적 흐름의 후속 단계이며 PKCE를 사용하기 때문에 더 안전합니다. PKCE는 퍼블릭 클라이언트를 위한 OAuth 2.0 인증 코드 부여의 확장입니다. PKCE는 도용된 인증 코드의 사용을 방지합니다.

다이어그램은 다음을 보여 줍니다.

1. 애플리케이션은 코드 검증기와 코드 챌린지를 생성합니다. 이러한 값은 잘 정의되고 고유한 값이며 나중에 참조할 수 있도록 Amazon Cognito로 전송됩니다.
2. 애플리케이션은 Amazon Cognito의 /oauth2/권한 부여 엔드포인트를 호출합니다. 구성된 IdP 로그인으로 사용자를 자동으로 리디렉션합니다.
3. IdP는 기존 세션을 확인합니다. 존재하지 않는 경우 사용자는 사용자 이름과 비밀번호를 제공하여 인증을 요청하는 메시지를 받습니다. IdP는 SAML 토큰으로 아마존 코그니토에 응답합니다.

4. Amazon Cognito가 코드 토큰으로 성공을 반환하면 웹 서버는 /oauth2/token을 호출하여 코드 토큰을 액세스 토큰으로 교환합니다.
5. 액세스 토큰은 이후에 다른 애플리케이션을 호출할 때마다 사용됩니다.
6. 다른 애플리케이션은 Amazon Cognito로 액세스 토큰을 검증합니다.

다음 흐름을 사용해야 합니다.

- SPA 또는 모바일 애플리케이션용

이 흐름은 사용해서는 안 됩니다.

- 애플리케이션 백엔드가 인증을 처리하는 경우

리소스 소유자 암호 흐름

리소스 소유자 암호 흐름은 리디렉션 기능이 없는 애플리케이션을 위한 것입니다. 자체 애플리케이션에서 로그인 양식을 생성하여 구축됩니다. 리디렉션 흐름에 의존하지 않고 Amazon Cognito에서 CLI 또는 SDK 호출을 통해 로그인을 확인합니다. 페더레이션에는 브라우저 기반 리디렉션이 필요하므로 이 인증 흐름에서는 페더레이션이 불가능합니다.

다이어그램은 다음을 보여 줍니다.

1. 사용자는 애플리케이션에서 제공하는 로그인 양식에 자격 증명을 입력합니다.
2. AWS 명령줄 인터페이스 (AWS CLI) 는 아마존 [admin-initiated-auth](#) 코그니토를 호출합니다.

참고: 또는 AWS CLI 대신 AWS SDK를 사용할 수도 있습니다.

3. Amazon Cognito는 액세스 토큰을 반환합니다.
4. 액세스 토큰은 이후에 다른 애플리케이션을 호출할 때마다 사용됩니다.
5. 다른 애플리케이션은 Amazon Cognito로 액세스 토큰을 검증합니다.

다음 흐름을 사용해야 합니다.

- 저장된 자격 증명을 액세스 토큰으로 변환하여 직접 인증 로직 (예: 기본 액세스 인증 또는 디지스트 액세스 인증) 을 사용하는 기존 클라이언트를 OAuth로 마이그레이션하는 경우

이 흐름은 사용해서는 안 됩니다.

- 페더레이션 ID를 사용하려는 경우
- 애플리케이션이 리디렉션을 지원하는 경우

도구

서비스

- [Amazon Cognito](#)는 웹 및 모바일 앱에 대한 인증, 권한 부여 및 사용자 관리를 제공합니다.

기타 도구

- [JSON 웹 토큰 \(JWT\) 디버거](#)는 웹 기반 JWT 검증 도구입니다.

에픽

애플리케이션 평가

작업	설명	필요한 기술
인증 요구 사항을 정의하십시오.	특정 인증 요구 사항에 따라 애플리케이션을 평가하십시오.	앱 개발자, 앱 아키텍트
요구 사항을 인증 흐름에 맞게 조정하십시오.	아키텍처 섹션에서 각 흐름의 의사 결정표와 설명을 사용하여 Amazon Cognito 인증 흐름을 선택합니다.	앱 개발자, 일반 AWS, 앱 아키텍트

아마존 Cognito 사용자 풀 설정

작업	설명	필요한 기술
사용자 풀을 생성합니다.	1. AWS 관리 콘솔에 로그인한 다음 Amazon Cognito 콘솔을 엽니다.	일반 AWS

작업	설명	필요한 기술
	<ol style="list-style-type: none"> 2. 새 Cognito 사용자 풀을 생성합니다. 지침은 Amazon Cognito 사용자 풀을 참조하십시오. 3. 필요에 따라 사용자 풀 설정과 속성을 업데이트하십시오. 예를 들어, 사용자 풀에 대한 암호 정책을 설정합니다. 아직 앱 클라이언트를 만들지 마세요. 	
<p>(선택 사항) ID 공급자를 구성합니다.</p>	<ol style="list-style-type: none"> 1. Amazon Cognito 사용자 풀에서 SAML 자격 증명 공급자를 생성합니다. 지침은 사용자 풀에 SAML 자격 증명 공급자 추가 및 관리를 참조하십시오. 2. Amazon Cognito 사용자 풀의 페더레이션을 사용하도록 타사 SAML 자격 증명 공급자를 구성하십시오. 자세한 내용은 타사 SAML ID 공급자 구성을 참조하십시오. ADFS를 사용하는 경우 Amazon Cognito 사용자 풀을 사용하여 웹 앱을 위한 ADFS 페더레이션 구축 (AWS 블로그 게시물)을 참조하십시오. 	<p>일반 AWS, 페더레이션 관리자</p>

작업	설명	필요한 기술
앱 클라이언트 생성	<ol style="list-style-type: none"> 1. 사용자 풀용 앱 클라이언트를 생성합니다. 지침은 앱 클라이언트 만들기를 참조하십시오. 유념할 사항: <ul style="list-style-type: none"> • 필요에 따라 설정을 변경하십시오 (예: 토큰 만료). • 인증 흐름에 클라이언트 암호가 필요하지 않은 경우 클라이언트 암호 생성 확인란의 선택을 취소하십시오. 2. 앱 클라이언트 설정을 선택하여 통합을 사용자 풀 로그인 (사용자 이름 및 암호) 또는 SAML 기반 IdP를 통한 페더레이션 로그인으로 변경합니다. 3. URL을 정의하고 필요에 따라 OAuth 흐름 또는 범위를 정의하여 IdP를 활성화하세요. 	일반 AWS

애플리케이션을 Amazon Cognito와 통합

작업	설명	필요한 기술
Amazon Cognito 통합 세부 정보를 교환하십시오.	인증 흐름에 따라 사용자 풀 ID 및 앱 클라이언트 ID와 같은 Amazon Cognito 정보를 애플리케이션과 공유하십시오.	앱 개발자, 일반 AWS
Amazon Cognito 인증을 구현하십시오.	이는 선택한 인증 흐름, 프로그래밍 언어, 사용 중인 프레임워크	앱 개발자

작업	설명	필요한 기술
	크에 따라 달라집니다. 시작하기 위한 일부 링크는 관련 리소스 섹션을 참조하십시오.	

관련 리소스

AWS 설명서

- [사용자 풀 인증 흐름](#)
- [JSON 웹 토큰 확인](#)
- [Amazon Cognito 자격 증명 풀을 사용하여 ASP.NET Core 앱에서 AWS 서비스에 액세스할 수 있습니다.](#)
- 프레임워크 및 SDK:
 - [아마존 Amplify 인증](#)
 - [Amazon Cognito 자격 증명 공급자 예제 \(Java 2.x용 AWS SDK 설명서\)](#)
 - [Amazon Cognito를 사용한 사용자 인증 \(.NET용 AWS SDK 설명서\)](#)

AWS 블로그 게시물

- [쿠키를 사용한 권한 부여 @Edge: 인증되지 않은 사용자가 Amazon CloudFront 콘텐츠를 다운로드하지 못하도록 보호합니다.](#)
- [Amazon Cognito 사용자 풀을 사용하여 웹 앱을 위한 ADFS 페더레이션 구축](#)

구현 파트너

- [인증 솔루션을 위한 AWS 파트너](#)

추가 정보

FAQ

암시적 흐름이 더 이상 사용되지 않는 이유는 무엇입니까?

[OAuth 2.1 프레임워크](#)가 출시된 이후, 암시적 흐름은 보안상의 이유로 더 이상 사용되지 않는 것으로 표시되었습니다. [대안으로 아키텍처 섹션에 설명된 PKCE의 인증 코드 흐름을 사용하세요.](#)

Amazon Cognito에서 필요한 일부 기능을 제공하지 않으면 어떻게 됩니까?

AWS 파트너는 인증 및 권한 부여 솔루션을 위한 다양한 통합을 제공합니다. 자세한 [내용은 인증 솔루션 AWS 파트너](#)를 참조하십시오.

Amazon Cognito 자격 증명 풀 흐름은 어떻게 됩니까?

Amazon Cognito 사용자 풀과 페더레이션 자격 증명은 인증을 위한 것입니다. Amazon Cognito 자격 증명 풀은 임시 AWS 자격 증명을 요청하여 AWS 리소스 액세스를 승인하는 데 사용됩니다. 자격 증명 풀을 위한 ID 토큰 및 액세스 토큰 교환은 이 패턴에서 설명되지 않습니다. 자세한 내용은 Amazon Cognito [사용자 풀과 자격 증명 풀의 차이점과 일반적인 Amazon Cognito](#) 시나리오를 참조하십시오.

다음 단계

이 패턴은 Amazon Cognito 인증 흐름의 개요를 제공합니다. 다음 단계로 애플리케이션 프로그래밍 언어에 대한 세부 구현을 선택해야 합니다. 여러 언어가 Amazon Cognito와 함께 사용할 수 있는 SDK 및 프레임워크를 제공합니다. 유용한 참조는 [관련](#) 리소스 섹션을 참조하십시오.

AWS 가드 정책을 사용하여 AWS Config 사용자 지정 규칙을 생성합니다. CloudFormation

작성자: 앤드류 록 (AWS), 카일라쉬 하빌다르 (AWS), 니콜 브라운 (AWS), 타냐 하웰 (AWS)

코드 [aws-config-custom-rule리포지토리](#): -클라우드포메이션-가드

환경: PoC 또는 파일럿

기술: 보안, 자격 증명, 규정 준수, 관리 및 거버넌스

AWS 서비스: AWS CloudFormation, AWS Config

요약

[AWS Config](#) 규칙은 AWS 리소스와 대상 구성 상태를 평가하는 데 도움이 됩니다. AWS Config 규칙에는 관리형 규칙과 사용자 지정이라는 두 가지 유형이 있습니다. AWS Lambda 함수 또는 언어인 [CloudFormation AWS Guard GitHub \(\)](#) 를 사용하여 사용자 지정 규칙을 생성할 수 있습니다. `policy-as-code`

Guard로 생성한 규칙은 관리형 규칙보다 더 세밀한 제어를 제공하며, 일반적으로 완전 사용자 지정 Lambda 규칙보다 구성하기가 더 쉽습니다. 이 접근 방식을 통해 엔지니어와 아키텍트는 Lambda를 통해 사용자 지정 규칙을 배포하는 데 필요한 Python, NodeJS 또는 Java에 대해 몰라도 규칙을 구축할 수 있습니다.

이 패턴은 Guard를 통해 사용자 지정 규칙을 채택하는 데 도움이 되는 실행 가능한 템플릿, 코드 샘플 및 배포 접근 방식을 제공합니다. 관리자는 이 패턴을 사용하여 AWS Config를 사용하여 [구성 항목](#) 속성이 있는 사용자 지정 규정 준수 규칙을 구축할 수 있습니다. 예를 들어, 개발자는 AWS Config 구성 항목에 대한 Guard 정책을 사용하여 배포된 AWS 및 비 AWS 리소스의 상태를 지속적으로 모니터링하고, 규칙 위반을 탐지하고, 자동으로 해결을 시작할 수 있습니다.

목표

이 패턴을 읽고 나면 다음을 할 수 있을 것입니다.

- Guard 정책 코드가 AWS Config 서비스와 어떻게 상호 작용하는지 이해하십시오.

- 시나리오 1을 배포하십시오. 시나리오 1은 Guard 구문을 사용하여 암호화된 볼륨의 규정 준수를 검증하는 AWS Config 사용자 지정 규칙입니다. [이 규칙은 드라이브가 사용 중인지 확인하고 드라이브 유형이 gp3인지 확인합니다.](#)
- 시나리오 2를 배포하십시오. 시나리오 2는 Guard 구문을 사용하여 Amazon GuardDuty 규정 준수를 검증하는 AWS Config 사용자 지정 규칙입니다. 이 규칙은 GuardDuty 레코더에 [Amazon S3 보호 및 Amazon EKS](#) 보호가 활성화되어 있는지 확인합니다.

사전 조건 및 제한 사항

사전 조건

- 활성 상태의 AWS 계정
- AWS Config, 사용자의 AWS [계정에서 설정](#)

제한 사항

- Guard 사용자 지정 규칙은 대상 구성 항목 JSON 레코드의 키-값 쌍만 쿼리할 수 있습니다.

아키텍처

Guard 구문을 AWS Config 규칙에 사용자 지정 정책으로 적용합니다. AWS Config는 지정된 각 리소스의 계층적 JSON을 캡처합니다. AWS Config 구성 항목의 JSON에는 키-값 쌍이 포함되어 있습니다. 이러한 속성은 Guard 구문에서 해당 값에 할당되는 변수로 사용됩니다.

다음은 Guard 구문에 대한 설명입니다. 구성 항목 JSON의 변수가 사용되며 앞에 문자가 추가됩니다.
%

```
# declare variable
let <variable name> = <'value'>

# create rule and assign condition and policy
rule <rule name> when
  <CI json key> == <"CI json value"> {
    <top level CI json key>.<next level CI json key> == %<variable name>
  }
```

시나리오 1: 아마존 EBS 볼륨

시나리오 1은 Guard 구문을 사용하여 암호화된 볼륨의 규정 준수를 검증하는 AWS Config 사용자 지정 규칙을 배포합니다. 이 규칙은 드라이브가 사용 중인지 확인하고 드라이브 유형이 gp3인지 확인합니다.

다음은 시나리오 1에 대한 AWS Config 구성 항목의 예시입니다. 이 구성 항목에는 Guard 정책에서 변수로 사용되는 세 개의 키값 쌍이 있습니다: `volumestatus`, `volumeencryptionstatus`, `volumetype` 또한 `resourceType` 키는 Guard 정책에서 필터로 사용됩니다.

```
{
  "version": "1.3",
  "accountId": "111111111111",
  "configurationItemCaptureTime": "2023-01-15T19:04:45.402Z",
  "configurationItemStatus": "ResourceDiscovered",
  "configurationStateId": "4444444444444444",
  "configurationItemMD5Hash": "",
  "arn": "arn:aws:ec2:us-west-2:111111111111:volume/vol-222222222222",
  "resourceType": "AWS::EC2::Volume",
  "resourceId": "vol-222222222222",
  "awsRegion": "us-west-2",
  "availabilityZone": "us-west-2b",
  "resourceCreationTime": "2023-01-15T19:03:22.247Z",
  "tags": {},
  "relatedEvents": [],
  "relationships": [
    {
      "resourceType": "AWS::EC2::Instance",
      "resourceId": "i-3333333333333333",
      "relationshipName": "Is attached to Instance"
    }
  ],
  "configuration": {
    "attachments": [
      {
        "attachTime": "2023-01-15T19:03:22.000Z",
        "device": "/dev/xvda",
        "instanceId": "i-3333333333333333",
        "state": "attached",
        "volumeId": "vol-222222222222",
        "deleteOnTermination": true,
        "associatedResource": null,
        "instanceOwningService": null
      }
    ]
  }
}
```

```

    "availabilityZone": "us-west-2b",
    "createTime": "2023-01-15T19:03:22.247Z",
    "encrypted": false,
    "kmsKeyId": null,
    "outpostArn": null,
    "size": 8,
    "snapshotId": "snap-5555555555555555",
    "state": "in-use",
    "volumeId": "vol-222222222222",
    "iops": 100,
    "tags": [],
    "volumeType": "gp2",
    "fastRestored": null,
    "multiAttachEnabled": false,
    "throughput": null,
    "sseType": null
  },
  "supplementaryConfiguration": {}
}

```

다음은 Guard 구문을 사용하여 시나리오 1의 변수와 규칙을 정의하는 예제입니다. 이 예에서 다음과 같이 합니다.

- 처음 세 줄은 `let` 명령을 사용하여 변수를 정의합니다. 구성 항목의 속성에서 파생된 이름과 값이 할당됩니다.
- `compliancecheck` 규칙 블록은 일치하는 `resourceType` 키-값 쌍을 찾는 `when` 조건부 종속성을 추가합니다. `AWS::EC2::Volume` 일치하는 항목이 발견되면 규칙은 나머지 JSON 속성을 살펴보고 다음 세 가지 조건, 및 에서 일치하는 항목을 찾습니다. `state encrypted volumeType`

```

let volumestatus = 'available'
let volumetype = 'gp3'
let volumeencryptionstatus = true

rule compliancecheck when
  resourceType == "AWS::EC2::Volume" {
    configuration.state == %volumestatus
    configuration.encrypted == %volumeencryptionstatus
    configuration.volumeType == %volumetype
  }

```

이 사용자 지정 규칙을 구현하는 전체 CloudFormation Guard 사용자 지정 정책은 코드 [awsconfig-guard-cft저장소의.yaml](#) 또는 [awsconfig-guard-tf-ec2vol.json](#)을 참조하십시오. GitHub [이 사용자 지정 정책을 Guard에 배포하는 HashiCorp Terraform 코드의 경우 코드 저장소의.json](#)을 참조하십시오. [CloudFormation awsconfig-guard-tf-example](#)

시나리오 2: 규정 준수 GuardDuty

시나리오 2는 Guard 구문을 사용하여 Amazon GuardDuty 규정 준수를 검증하는 AWS Config 사용자 지정 규칙을 배포합니다. 이 규칙은 GuardDuty 레코더에 Amazon S3 보호 및 Amazon EKS 보호가 활성화되어 있는지 확인합니다. 또한 GuardDuty 조사 결과가 15분마다 게시되는지 확인합니다. 이 시나리오는 조직 (AWS Organizations) 의 모든 AWS 계정과 AWS 지역에 배포할 수 있습니다.

다음은 시나리오 2에 대한 AWS Config 구성 항목의 예시입니다. 이 구성 항목에는 Guard 정책에 서 변수로 사용되는 세 개의 키값 쌍이 있습니다: FindingPublishingFrequency, S3Logs. Kubernetes 또한 resourceType 키는 정책에서 필터로 사용됩니다.

```
{
  "version": "1.3",
  "accountId": "111111111111",
  "configurationItemCaptureTime": "2023-11-27T13:34:28.888Z",
  "configurationItemStatus": "OK",
  "configurationStateId": "77777777777777",
  "configurationItemMD5Hash": "",
  "arn": "arn:aws:guardduty:us-west-2:111111111111:detector/66666666666666666666666666666666",
  "resourceType": "AWS::GuardDuty::Detector",
  "resourceId": "66666666666666666666666666666666",
  "resourceName": "66666666666666666666666666666666",
  "awsRegion": "us-west-2",
  "availabilityZone": "Regional",
  "resourceCreationTime": "2020-02-17T02:48:04.511Z",
  "tags": {},
  "relatedEvents": [],
  "relationships": [],
  "configuration": {
    "Enable": true,
    "FindingPublishingFrequency": "FIFTEEN_MINUTES",
    "DataSources": {
      "S3Logs": {
        "Enable": true
      },
      "Kubernetes": {
```

```

    "AuditLogs": {
      "Enable": true
    }
  },
  "Id": "66666666666666666666666666666666",
  "Tags": [],
},
"supplementaryConfiguration": {
  "CreatedAt": "2020-02-17T02:48:04.511Z"
}
}

```

다음은 시나리오 2에서 Guard 구문을 사용하여 변수와 규칙을 정의하는 예제입니다. 이 예에서 다음과 같이 합니다.

- 처음 세 줄은 let 명령을 사용하여 변수를 정의합니다. 구성 항목의 속성에서 파생된 이름과 값이 할당됩니다.
- compliancecheck 규칙 블록은 일치하는 resourceType 키-값 쌍을 찾는 when 조건부 종속성을 추가합니다. AWS::GuardDuty::Detector 일치하는 항목이 발견되면 규칙은 나머지 JSON 속성을 살펴보고 다음 세 가지 조건, 및 에서 일치하는 항목을 찾습니다. S3Logs.Enable Kubernetes.AuditLogs.Enable FindingPublishingFrequency

```

let s3protection = true
let kubernetesprotection = true
let publishfrequency = 'FIFTEEN_MINUTES'

rule compliancecheck when
  resourceType == "AWS::GuardDuty::Detector" {
    configuration.DataSources.S3Logs.Enable == %s3protection
    configuration.DataSources.Kubernetes.AuditLogs.Enable ==
%kubernetesprotection
    configuration.FindingPublishingFrequency == %publishfrequency
  }

```

이 사용자 지정 규칙을 구현하는 전체 CloudFormation Guard 사용자 지정 정책은 코드 [awsconfig-guard-cft-gd저장소의.yaml](#)을 참조하십시오. GitHub [이 사용자 지정 정책을 CloudFormation Guard에 배포하는 HashiCorp Terraform 코드의 경우 코드 저장소의.json](#)을 참조하십시오awsconfig-guard-tf-gd.

도구

서비스

- [AWS는 AWS CloudFormation](#) 리소스를 설정하고, 빠르고 일관되게 프로비저닝하고, AWS 계정 및 지역 전반의 수명 주기 전반에 걸쳐 리소스를 관리할 수 있도록 지원합니다.
- [Config](#)는 사용자의 AWS 계정에 있는 리소스와 그 구성 방식을 자세히 보여줍니다. 리소스가 서로 관련되는 방식과 리소스의 구성이 시간이 지남에 따라 변경된 방식을 식별하는 데 도움이 됩니다.

기타 도구

- [HashiCorp Terraform](#)은 코드를 사용하여 클라우드 인프라 및 리소스를 프로비저닝하고 관리하는 데 도움이 되는 오픈 소스 IaC (코드형 인프라) 도구입니다.

코드 리포지토리

이 패턴의 코드는 GitHub [AWS Config with CloudFormation Guard](#) 리포지토리에서 사용할 수 있습니다. 이 코드 리포지토리에는 이 패턴에 설명된 두 시나리오 모두에 대한 샘플이 들어 있습니다.

에픽

AWS Config 사용자 지정 규칙 생성

작업	설명	필요한 기술
(선택 사항) 규칙의 키-값 쌍을 선택합니다.	<p>사용자 지정 가드 정책을 정의하는 경우 이 단계를 완료하세요. 시나리오 1 또는 2의 샘플 정책 중 하나를 사용하는 경우 이 단계를 건너뛰십시오.</p> <ol style="list-style-type: none"> 1. AWS Management Console에 로그인하여 https://console.aws.amazon.com/config/에서 AWS Config 콘솔을 엽니다. 2. 왼쪽 탐색창에서 [Resources] 를 선택합니다. 	AWS 관리자, 보안 엔지니어

작업	설명	필요한 기술
	<ol style="list-style-type: none"> 3. 리소스 인벤토리에서 AWS Config 사용자 지정 규칙을 생성할 대상 리소스 유형을 선택합니다. 4. 세부 정보 보기를 선택합니다. 5. 구성 항목 보기 (JSON) 를 선택합니다. 이 섹션은 구성 항목을 JSON 형식으로 표시하도록 확장됩니다. 6. AWS Config 사용자 지정 규칙을 구축하려는 키-값 쌍을 식별하십시오. 	
사용자 지정 규칙 생성.	이전에 식별한 키-값 쌍을 사용하거나 제공된 샘플 가드 정책 중 하나를 사용하여 AWS Config 사용자 지정 정책 규칙 생성의 지침에 따라 사용자 지정 규칙을 생성합니다.	AWS 관리자, 보안 엔지니어

작업	설명	필요한 기술
<p>사용자 지정 규칙을 확인합니다.</p>	<p>다음 중 하나를 수행하여 사용자 지정 Guard 규칙을 검증하십시오.</p> <ul style="list-style-type: none"> AWS 명령줄 인터페이스 (AWS CLI) 에 다음 명령을 입력합니다. <pre data-bbox="625 569 1029 768">cfn-guard validate -r guard-s3.guard -d s3bucket-prod-pass.json</pre> <ul style="list-style-type: none"> AWS Config 규칙을 사용한 리소스 평가의 Detective 모드 지침에 따라 규칙을 AWS Config에 배포하십시오. Guard 구문이 대상 계정 또는 파일의 해당 리소스와 올바르게 일치하는지 확인하십시오. 	<p>AWS 관리자, 보안 엔지니어</p>

문제 해결

문제	Solution
<p>AWS Config 외부에서 CloudFormation 가드 정책을 테스트해 보십시오.</p>	<p>단위 테스트는 로컬 디바이스 또는 AWS Cloud9 IDE와 같은 통합 개발 환경 (IDE) 에서 수행할 수 있습니다. 단위 테스트를 수행하려면 다음을 수행하십시오.</p> <ol style="list-style-type: none"> AWS CloudFormation Guard CLI와 해당 종속 항목을 설치합니다. JSON 형식의 CI 샘플을 워크스테이션에.json 파일로 저장합니다.

문제	Solution
	<p>3. GuardDuty 정책을 워크스테이션에.guard 파일로 저장합니다.</p> <p>4. Guard CLI에서 다음 명령을 입력하여 Guard 정책을 사용하여 샘플 JSON 파일을 검증합니다.</p> <pre data-bbox="868 483 1507 640">cfn-guard validate \ -r guard-s3.guard \ -d s3bucket-prod-pass.json</pre>
AWS Config 사용자 지정 규칙을 디버깅합니다.	가드 정책에서 EnableDebugLogDelivery 값을 변경하십시오. true 기본 값은 false입니다. 로그 메시지는 Amazon에 저장됩니다 CloudWatch.

관련 리소스

AWS 설명서

- [AWS Config 사용자 지정 정책 규칙 생성](#) (AWS Config 설명서)
- [AWS CloudFormation 가드 규칙 작성](#) (CloudFormation 가드 설명서)

AWS 블로그 게시물 및 워크샵

- [AWS CloudFormation 가드 2.0 소개](#) (AWS 블로그 게시물)

기타 리소스

- [AWS CloudFormation 가드](#) (GitHub)
- [CloudFormation 가드 CLI 설명서](#) () GitHub

여러 AWS 계정의 Prowler 보안 조사 결과에 대한 통합 보고서 생성

작성자: Mike Virgilio(AWS), Andrea Di Fabio(AWS), Cameron Covington (AWS), 및 Jay Durga(AWS)

코드 리포지토리: multi-account-security-assessment-via-prowler	환경: 프로덕션	기술: 보안, 자격 증명, 규정 준수
워크로드: 오픈소스	AWS 서비스: AWS CloudFormation, 아마존 EC2, AWS 자격 증명 및 액세스 관리	

요약

[Prowler](#) (GitHub) 는 Amazon Web Services (AWS) 계정을 평가, 감사 및 모니터링하여 보안 모범 사례를 준수하는지 확인하는 데 도움이 되는 오픈 소스 명령줄 도구입니다. 이 패턴에서는 에서 관리하는 중앙 집중식 조직에 Prowler를 배포한 다음 Prowler를 사용하여 조직 AWS 계정 내 모든 계정에 대한 보안 평가를 수행합니다. AWS Organizations

평가에 Prowler를 배포하고 활용하는 방법은 다양하지만 이 솔루션은 신속한 배포, 조직의 모든 계정 또는 정의된 대상 계정에 대한 전체 분석, 보안 조사 결과에 대한 액세스 가능한 보고를 위해 설계되었습니다. 이 솔루션에서는 Prowler가 조직의 모든 계정에 대한 보안 평가를 완료하면 결과를 통합합니다. 또한 Prowler가 AWS Control Tower을 통해 프로비저닝된 계정에서 Amazon Simple Storage Service(Amazon S3) 버킷을 스캔하지 못하게 하는 제한과 관련된 오류와 같은 모든 예상 오류 메시지를 필터링합니다. 필터링된 통합 결과는 이 패턴에 포함된 Microsoft Excel 템플릿에 보고됩니다. 이 보고서를 사용하여 조직의 보안 제어에 대한 잠재적 개선 사항을 식별할 수 있습니다.

이 솔루션은 다음 사항을 염두에 두고 설계되었습니다.

- AWS CloudFormation 템플릿을 사용하면 이러한 패턴으로 AWS 리소스를 배포하는 데 필요한 노력을 줄일 수 있습니다.
- 배포 시 CloudFormation 템플릿과 prowler_scan.sh 스크립트의 매개 변수를 조정하여 환경에 맞게 템플릿을 사용자 지정할 수 있습니다.
- Prowler 평가 및 보고 속도는 병렬 처리 AWS 계정, 집계된 결과, 권장 수정 사항이 포함된 통합 보고, 자동 생성된 시각화를 통해 최적화됩니다.

- 사용자는 스캔 진행 상황을 모니터링할 필요가 없습니다. 평가가 완료되면 보고서를 검색할 수 있도록 Amazon Simple Service(Amazon SNS) 주제를 통해 사용자에게 알립니다.
- 보고서 템플릿을 사용하면 조직 전체에 관련된 결과만 읽고 평가하는 데 도움을 줍니다.

사전 조건 및 제한 사항

사전 조건

- 에서 조직의 AWS 계정 구성원 계정으로 관리되는 보안 서비스 및 도구를 호스팅하는 데 사용됩니다. AWS Organizations 이 패턴에서는 이 계정을 보안 계정이라고 합니다.
- 보안 계정에는 아웃바운드 인터넷 액세스가 가능한 프라이빗 서브넷이 있어야 합니다. 자세한 지침은 Amazon Virtual Private Cloud(Amazon VPC) 설명서의 [프라이빗 서브넷에 서버가 있는 VPC 및 NAT](#)를 참조하세요. 퍼블릭 서브넷에 프로비저닝된 [NAT 게이트웨이](#)를 사용하여 인터넷 액세스를 설정할 수 있습니다.
- AWS Organizations 관리 계정 또는 관리자 권한을 위임받은 계정에 대한 CloudFormation 액세스 지침은 설명서의 [위임된 관리자 등록](#)을 참조하십시오. CloudFormation
- AWS Organizations 와 CloudFormation 사이에 신뢰할 수 있는 액세스를 활성화하십시오. 지침은 설명서의 [신뢰할 수 있는 AWS Organizations 액세스 활성화](#)를 CloudFormation 참조하십시오.

제한 사항

- 대상은 에서 조직으로 AWS 계정 관리해야 AWS Organizations 합니다. 를 사용하지 AWS Organizations 않는 경우 IAM- ProwlerExecRole .yaml CloudFormation 템플릿과 prowler_scan.sh 스크립트를 환경에 맞게 업데이트할 수 있습니다. 대신 스크립트를 실행하려는 AWS 계정 ID와 지역 목록을 제공합니다.
- CloudFormation 템플릿은 아웃바운드 인터넷 액세스가 가능한 프라이빗 서브넷에 Amazon Elastic Compute Cloud (Amazon EC2) 인스턴스를 배포하도록 설계되었습니다. AWS Systems Manager 에이전트 (SSM 에이전트) 가 AWS Systems Manager 서비스 엔드포인트에 도달하려면 아웃바운드 액세스가 필요하며, 코드 리포지토리를 복제하고 종속성을 설치하려면 아웃바운드 액세스가 필요합니다. 퍼블릭 서브넷을 사용하고자 한다면 prowler-resources.yaml 템플릿을 수정하여 [Elastic IP 주소](#)를 EC2 인스턴스와 연결해야 합니다.

제품 버전

- Prowler 버전 3.0 이상

아키텍처

이 다이어그램은 다음 프로세스를 보여줍니다.

1. 사용자는 기능인 세션 관리자를 사용하여 EC2 인스턴스를 인증하고 prowler_scan.sh 스크립트를 실행합니다. AWS Systems Manager 이 셸 스크립트는 2~8단계를 수행합니다.
2. EC2 인스턴스는 S3 버킷에 액세스하고 조직의 다른 계정에서 ProwlerEC2Role IAM 역할을 떠맡는 권한을 부여하는 ProwlerExecRole IAM 역할을 떠맡습니다.
3. EC2 인스턴스는 조직의 관리 계정에서 ProwlerExecRole IAM 역할을 떠맡고 조직의 계정 목록을 생성합니다.
4. EC2 인스턴스는 조직의 멤버 계정(아키텍처 다이어그램에서는 워크로드 계정이라고 함)에서 ProwlerExecRole IAM 역할을 맡아 각 계정에서 보안 평가를 수행합니다. 조사 결과는 EC2 인스턴스에 CSV 및 HTML 파일로 저장됩니다.

참고: HTML 파일은 Prowler 평가의 결과입니다. HTML의 특성상 이 패턴에서는 직접 연결되거나 처리되거나 사용되지 않습니다. 하지만 이는 개별 계정 보고서를 검토하는 데 유용할 수 있습니다.

5. EC2 인스턴스는 모든 CSV 파일을 처리하여 알려진 예상 오류를 제거하고 나머지 조사 결과를 단일 CSV 파일로 통합합니다.
6. EC2 인스턴스는 generateVisualizations.py 스크립트를 실행합니다. 이 스크립트는 집계된 조사 결과의 CSV 파일을 처리하고 결과를 이해하고 보고하는 데 도움을 주는 그래프와 차트로 구성된 PNG 파일을 생성합니다. 또한 스캔 및 PNG 파일에 대한 정보가 들어 있는 HTML 파일도 만듭니다.
7. EC2 인스턴스는 개별 계정 결과, 집계된 결과 및 생성된 시각화를 zip 파일로 패키징합니다.
8. EC2 인스턴스는 이 zip 파일을 S3 버킷에 업로드합니다.
9. EventBridge 규칙은 파일 업로드를 탐지하고 Amazon SNS 주제를 사용하여 사용자에게 평가 완료를 알리는 이메일을 보냅니다.
10. 사용자가 S3 버킷에서 zip 파일을 다운로드합니다. 사용자는 결과를 Excel 템플릿으로 가져와서 검토합니다.

도구

서비스

- [Amazon Elastic Compute Cloud\(Amazon EC2\)](#)는 AWS 클라우드에서 확장 가능한 컴퓨팅 용량을 제공합니다. 필요한 만큼 가상 서버를 시작하고 빠르게 스케일 업하거나 스케일 다운할 수 있습니다.
- [EventBridgeAmazon](#)은 다양한 소스의 실시간 데이터와 애플리케이션을 연결하는 데 도움이 되는 서버리스 이벤트 버스 서비스입니다. AWS Lambda 함수, API 대상을 사용하는 HTTP 호출 엔드포인트 또는 기타 이벤트 버스를 예로 들 수 있습니다. AWS 계정
- [AWS Identity and Access Management \(IAM\)](#) 을 사용하면 리소스 인증 및 사용 권한을 부여받은 사용자를 제어하여 AWS 리소스에 대한 액세스를 안전하게 관리할 수 있습니다.
- [AWS Organizations](#)여러 조직을 사용자가 만들고 중앙에서 관리할 수 있는 하나의 AWS 계정 조직으로 통합하는 데 도움이 되는 계정 관리 서비스입니다.
- [Amazon Simple Notification Service\(Amazon SNS\)](#)를 사용하면 웹 서버 및 이메일 주소를 포함하여 게시자와 클라이언트 간의 메시지 교환을 조정하고 관리할 수 있습니다.
- [Amazon Simple Storage Service\(S3\)](#)는 원하는 양의 데이터를 저장, 보호 및 검색하는 데 도움이 되는 클라우드 기반 객체 스토리지 서비스입니다.
- [AWS Systems Manager](#)은 AWS 클라우드에서 실행되는 애플리케이션 및 인프라를 관리하는 데 도움을 줍니다. 애플리케이션 및 리소스 관리를 간소화하고, 운영 문제를 감지하고 해결하는 시간을 단축하며, AWS 리소스를 대규모로 안전하게 관리할 수 있도록 지원합니다. 이 패턴은 Systems Manager의 기능인 Session Manger를 사용합니다.

기타 도구

- [Prowler](#)는 계정이 AWS 보안 모범 사례와 기타 보안 프레임워크 및 표준을 준수하는지 평가, 감사 및 모니터링할 수 있도록 도와주는 오픈 소스 명령줄 도구입니다.

코드 리포지토리

[이 패턴의 코드는 Prowler 리포지토리를 통한 다중 계정 보안 평가에서 확인할 수 있습니다. GitHub 코드 리포지토리에는 다음 파일이 포함되어 있습니다.](#)

- `prowler_scan.sh` — 이 bash 스크립트는 여러 개의 AWS 계정 Prowler 보안 평가를 병렬로 시작하는 데 사용됩니다. `Prowler-resources.yaml`에 정의된 대로 이 스크립트는 EC2 인스턴스의 폴더에 `CloudFormationtemplate` 자동으로 배포됩니다. `usr/local/prowler`
- `Prowler-resources.yaml` — 이 템플릿을 사용하여 조직의 보안 계정에 스택을 생성합니다. CloudFormation 이 템플릿은 솔루션을 지원하기 위해 이 계정에 필요한 모든 리소스를 배포합니다. 이 스택은 `IAM-.yaml` 템플릿보다 먼저 배포해야 합니다. `ProwlerExecRole` 중요한 프로덕션 워크로드를 호스팅하는 계정에는 이러한 리소스를 배포하지 않는 것이 좋습니다.

참고: 이 스택을 삭제하고 재배포하는 경우 IAM 역할 간의 교차 계정 종속성을 다시 빌드하려면 ProwlerExecRole 스택 세트를 다시 빌드해야 합니다.

- IAM- ProwlerExecRole .yaml — 이 CloudFormation 템플릿을 사용하여 관리 계정을 포함하여 조직의 모든 계정에 ProwlerExecRole IAM 역할을 배포하는 스택 세트를 생성합니다.
- generateVisualizations.py-prowler_scan.sh 스크립트는 이 Python 스크립트를 자동으로 호출하여 집계된 조사 결과를 기반으로 시각화를 생성하고 이를 S3 버킷에 저장된.zip 파일에 포함합니다. 이 스크립트는 다음 파일을 생성합니다.
 - FailuresByAccount-<date>.png-각 계정의 실패한 Prowler 검사를 보여 주는 막대형 차트
 - FailuresByService-<date>.png— 각각에 대한 실패한 Prowler 검사를 보여 주는 막대형 차트 AWS 서비스
 - ProcessedResultsByFailureSeverityCount-<date>.png- 각 심각도 수준(심각, 높음, 중간, 낮음, 정보용)에 대한 실패한 Prowler 검사의 분포를 보여 주는 막대형 차트
 - ResultsByFail-<date>.png- 심각도별 실패한 Prowler 검사의 파이형 차트
 - ResultsBySeverity-<date>.png- 심각도별 모든 Prowler 검사(합격 및 불합격)의 파이형 차트
 - ProwlerReport.html- 모든 이미지가 포함된 단일 HTML 파일
- prowler3-report-template.xlsm-이 Excel 템플릿을 사용하여 Prowler 조사 결과를 처리합니다. 보고서의 피벗 테이블은 검색 기능, 차트 및 통합 조사 결과를 제공합니다.

에픽

배포 준비

작업	설명	필요한 기술
코드 리포지토리를 복제합니다.	<ol style="list-style-type: none"> 1. 명령줄 인터페이스에서 작업 디렉터리를 샘플 파일을 저장하고자 하는 위치로 변경합니다. 2. 다음 명령을 입력합니다. <pre>git clone https://github.com/aws-samp</pre>	AWS DevOps

작업	설명	필요한 기술
	<p>les/multi-account-security-assessment-via-prowler.git</p>	
<p>템플릿을 검토합니다.</p>	<ol style="list-style-type: none"> 1. 복제된 리포지토리에서 프로러-리소스.yaml 및 IAM-.yaml 파일을 엽니다. ProwlerExecRole 2. 이러한 템플릿으로 생성한 리소스를 검토하고 환경에 맞게 필요한 만큼 템플릿을 조정합니다. 자세한 내용은 설명서의 템플릿 작업을 참조하십시오. CloudFormation 3. Prowler-resources.yaml 및 IAM-.yaml 파일을 저장하고 닫습니다. ProwlerExecRole 	<p>AWS DevOps</p>

CloudFormation 스택 생성

작업	설명	필요한 기술
<p>보안 계정에서 리소스를 프로 비저닝합니다.</p>	<p>prowler-resources.yaml 템플릿을 사용하여 보안 계정에 필요한 모든 리소스를 배포하는 CloudFormation 스택을 생성합니다. 지침은 설명서의 스택 생성을 참조하십시오. CloudFormation 이 템플릿을 배포할 때는 다음 사항을 유념합니다.</p>	<p>AWS DevOps</p>

작업	설명	필요한 기술
	<ol style="list-style-type: none"> 1. 템플릿 지정 페이지에서 템플릿 준비 완료를 선택한 다음 <code>prowler-resources.yaml</code> 파일을 업로드합니다. 2. 스택 세부 정보 지정 페이지의 스택 이름 상자에 <code>Prowler-Resources</code> 를 입력합니다. 3. Parameters(파라미터) 섹션에서 다음을 입력합니다. <ul style="list-style-type: none"> • <code>VPCId</code> — 계정에서 VPC 를 선택합니다. • <code>SubnetId</code> — 인터넷에 액세스할 수 있는 프라이빗 서브넷을 선택합니다. <p>참고: 퍼블릭 서브넷을 선택하면 CloudFormation 템플릿이 기본적으로 엘라스틱 IP 주소를 프로비저닝 및 연결하지 않기 때문에 EC2 인스턴스에 퍼블릭 IP 주소가 할당되지 않습니다.</p> <ul style="list-style-type: none"> • <code>InstanceType</code> -병렬 평가 수를 기준으로 인스턴스 크기를 선택합니다. <ul style="list-style-type: none"> • 10개의 경우 <code>r6i.large</code> 를 선택합니다. • 12개의 경우 <code>r6i.xlarge</code> 를 선택합니다. 	

작업	설명	필요한 기술
	<ul style="list-style-type: none"> • 14~18의 경우 r6i.2xlarge 를 선택합니다. • InstanceImageId - Amazon Linux의 기본값은 그대로 둡니다. • KeyPairName — 액세스에 SSH를 사용하는 경우 기존 키 페어의 이름을 지정합니다. • PermittedSSHInbound — 액세스에 SSH를 사용하는 경우 허용되는 CIDR 블록을 지정합니다. SSH를 사용하지 않는 경우 기본값인 127.0.0.1 을 유지합니다. • BucketName - 기본값은 prowler-output- <accountID>- <region> 입니다. 필요에 따라 이를 수정할 수 있습니다. 사용자 지정 값을 지정하는 경우 계정 ID와 리전이 지정된 값에 자동으로 추가됩니다. • EmailAddress - Prowler가 평가를 완료하고.zip 파일을 S3 버킷에 업로드할 때 Amazon SNS 알림을 받을 이메일 주소를 지정합니다. 	

작업	설명	필요한 기술
	<p>참고: Prowler가 평가를 완료하기 전에 SNS 구독 구성을 확인해야 합니다. 그렇지 않으면 알림이 전송되지 않습니다.</p> <ul style="list-style-type: none"> • IAMProwlerEC2Role — 명명 규칙에 따라 이 IAM 역할에 다른 이름이 필요한 경우가 아니면 기본값을 유지합니다. • IAMProwlerExecRole — IAM - .yaml 파일을 배포할 때 다른 이름을 사용하지 않는 한 기본값을 유지하십시오. ProwlerExecRole • Parallelism — 수행할 병렬 평가 수를 지정합니다. InstanceType 파라미터의 값이 이 수의 병렬 평가를 지원하는지 확인합니다. • FindingOutput -합격 결과를 제외하려면 FailOnly을 선택합니다. 이렇게 하면 출력 크기가 크게 줄어들고 해결해야 할 검사에 초점을 맞출 수 있습니다. 합격 결과를 포함하려면 FailAndPass 을 선택합니다. <p>4. 검토 페이지에서 다음 리소스에 기능이 필요함:</p>	

작업	설명	필요한 기술
	<p>[AWS::IAM::Role] 를 선택한 다음 [Create Stack] 을 선택합니다.</p> <p>5. 스택이 성공적으로 생성되면 CloudFormation 콘솔의 출력 탭에서 ProwlerEC2Role Amazon 리소스 이름 (ARN) 을 복사합니다. 이 ARN은 나중에 IAM - .yaml 파일을 배포할 때 사용됩니다. ProwlerExecRole</p>	

작업	설명	필요한 기술
<p>멤버 계정에서 IAM 역할을 프로비저닝합니다.</p>	<p>AWS Organizations 관리 계정 또는 관리자 권한이 위임된 계정에서 IAM- ProwlerExecRole .yaml 템플릿을 사용하여 스택 세트를 생성합니다. CloudFormation CloudFormation 그러면 스택 세트는 조직의 모든 멤버 계정에서 ProwlerExecRole IAM 역할을 배포합니다. 지침은 설명서의 서비스 관리 권한이 포함된 스택 세트 생성을 참조하십시오. CloudFormation 이 템플릿을 배포할 때는 다음 사항을 유념합니다.</p> <ol style="list-style-type: none"> 1. 템플릿 준비에서 [템플릿 준비 완료] 를 선택한 다음 IAM- ProwlerExecRole .yaml 파일을 업로드합니다. 2. StackSet 세부 정보 지정 페이지에서 스택 세트의 이름을 지정합니다. IAM-ProwlerExecRole 3. Parameters(파라미터) 섹션에서 다음을 입력합니다. <ul style="list-style-type: none"> • AuthorizedARN — ProwlerEC2Role 스택을 생성할 때 복사한 Prowler-Resources ARN을 입력합니다. • ProwlerExecRoleName -Prowler-resources. 	<p>AWS DevOps</p>

작업	설명	필요한 기술
	<p>yaml 파일을 배포할 때 다른 이름을 사용하지 않는 기본값을 ProwlerExecRole 로 유지합니다.</p> <ol style="list-style-type: none"> 권한에서 서비스 관리형 권한을 선택합니다. 배포 옵션 설정 페이지의 배포 대상에서 조직에 배포를 선택하고 모든 기본값을 수락합니다. <p>참고: 스택을 모든 멤버 계정에 동시에 배포하려면 최대 동시 계정 및 예를 들면 100와 같은 장애 허용 범위를 높은 값으로 설정합니다.</p> <ol style="list-style-type: none"> 배포 지역에서 Prowler용 EC2 인스턴스를 배포할 AWS 리전 위치를 선택합니다. IAM 리소스는 리전이 아닌 글로벌이므로 이를 통해 모든 활성 리전에 IAM 역할을 배포합니다. 검토 페이지에서 사용자 지정 이름으로 IAM 리소스를 생성할 AWS CloudFormation 수 있음을 확인합니다. 를 선택한 다음 [Create] 를 선택합니다. StackSet 스택 인스턴스 탭(개별 계정 상태)과 운영 탭(전체 상태)을 모니터링하여 배포가 완료되는 시점을 확인합니다. 	

작업	설명	필요한 기술
<p>관리 계정에서 IAM 역할을 프로비저닝합니다.</p>	<p>IAM- ProwlerExecRole .yaml 템플릿을 사용하여 조직의 관리 계정에 ProwlerExecRole IAM 역할을 배포하는 CloudFormation 스택을 생성합니다. 이전에 생성한 스택 세트는 관리 계정에 IAM 역할을 배포하지 않습니다. 지침은 설명서의 스택 생성을 참조하십시오. CloudFormation 이 템플릿을 배포할 때는 다음 사항을 유념합니다.</p> <ol style="list-style-type: none"> 1. 템플릿 지정 페이지에서 Template is ready (Template is ready) 를 선택한 다음 IAM- ProwlerExecRole .yaml 파일을 업로드합니다. 2. 스택 세부 정보 지정 페이지의 스택 이름 상자에 IAM- ProwlerExecRole 를 입력합니다. 3. Parameters(파라미터) 섹션에서 다음을 입력합니다. <ul style="list-style-type: none"> • AuthorizedARN — ProwlerEC2Role 스택을 생성할 때 복사한 Prowler-Resources ARN을 입력합니다. • ProwlerExecRoleName -Prowler-resources. yaml 파일을 배포할 때 다른 이름을 사용하지 않는 	<p>AWS DevOps</p>

작업	설명	필요한 기술
	<p>한 기본값을 ProwlerExecRole 로 유지합니다.</p> <p>4. 검토 페이지에서 다음 리소스에 기능이 필요함: [AWS::IAM::Role] 를 선택한 다음 [Create Stack] 을 선택합니다.</p>	

Prowler 보안 평가 수행

작업	설명	필요한 기술
스캔을 실행합니다.	<ol style="list-style-type: none"> 1. 조직의 보안 계정에 로그인합니다. 2. Session Manager를 사용하여 이전에 프로비저닝한 Prowler용 EC2 인스턴스에 연결합니다. 자세한 지침은 Session Manager를 사용하여 Linux 인스턴스에 연결을 참조하세요. 연결할 수 없는 경우 이 패턴의 문제 해결 섹션을 참조하세요. 3. <code>usr/local/prowler</code> 로 이동한 다음 <code>prowler_scan.sh</code> 파일을 엽니다. 4. 환경에 맞게 필요한 만큼 이 스크립트의 조정 가능한 파라미터와 변수를 검토하고 수정합니다. 사용자 지정 옵션에 대한 자세한 내용은 스크립트 시작 부분에 있는 설명을 참조하십시오. 	AWS 관리자

작업	설명	필요한 기술
	<p>예를 들어 관리 계정에서 조직의 모든 구성원 계정 목록을 가져오는 대신 스크립트를 수정하여 스캔할 AWS 계정 ID 또는 AWS 리전 ID를 지정하거나 이러한 파라미터가 포함된 외부 파일을 참조할 수 있습니다.</p> <p>5. prowler_scan.sh 파일을 저장하고 닫습니다.</p> <p>6. 다음 명령을 입력합니다. 이를 통해 prowler_scan.sh 스크립트를 실행합니다.</p> <pre data-bbox="630 871 1029 1115"> sudo -i screen cd /usr/local/ prowler ./prowler_scan.sh </pre> <p>다음을 참조하세요.</p> <ul style="list-style-type: none"> • screen 명령을 사용하면 연결 시간이 초과되거나 콘솔에 액세스할 수 없는 경우에도 스크립트를 계속 실행할 수 있습니다. • 스캔이 시작된 후 Ctrl+AD를 눌러 화면을 강제로 분리할 수 있습니다. 화면이 분리되면 인스턴스 연결을 닫고 평가를 계속할 수 있습니다. • 분리된 세션을 재개하려면 인스턴스에 연결하고 	

작업	설명	필요한 기술
	<p>sudo -i(을)를 입력한 다음 screen -r(을)를 입력합니다.</p> <ul style="list-style-type: none"> • 개별 계정 평가의 진행 상황을 모니터링하려면 <code>usr/local/prowler</code> 디렉터리로 이동하여 <code>tail -f output/stdout-<account-id></code> 명령을 입력하면 됩니다. <p>7. Prowler가 모든 계정에서 스캔을 완료할 때까지 기다립니다. 이 스크립트는 동시에 여러 계정을 평가합니다. 모든 계정에서 평가가 완료되면 <code>Prowler-Resources.yaml</code> 파일을 배포할 때 이메일 주소를 지정한 경우 알림을 받게 됩니다.</p>	

작업	설명	필요한 기술
<p>Prowler 조사 결과를 검색합니다.</p>	<ol style="list-style-type: none"> 1. prowler-output- <assessDate>.zip 버킷에서 prowler-output-<accountID>-<region> 파일을 다운로드합니다. 자세한 지침은 Amazon S3 설명서의 객체 다운로드를 참조하세요. 2. 다운로드한 파일을 포함하여 버킷에 있는 모든 객체를 삭제합니다. 이는 비용 최적화를 위한 모범 사례이며 언제든지 Prowler-Resources CloudFormation 스택을 삭제할 수 있도록 하기 위한 것입니다. 자세한 지침은 Amazon S3 설명서의 객체 삭제를 참조하세요. 	<p>일반 AWS</p>
<p>EC2 인스턴스를 중지합니다.</p>	<p>인스턴스가 유휴 상태일 때 해제되지 않도록 하려면 Prowler를 실행하는 EC2 인스턴스를 중지합니다. 자세한 지침은 Amazon EC2 설명서의 인스턴스 중지 및 시작을 참조하세요.</p>	<p>AWS DevOps</p>

조사 결과 보고서 생성

작업	설명	필요한 기술
<p>조사 결과를 가져옵니다.</p>	<ol style="list-style-type: none"> 1. prowler-report-templateExcel에서.xlsx 파일을 	<p>일반 AWS</p>

작업	설명	필요한 기술
	<p>연 다음 프로러 CSV 워크시트를 선택합니다.</p> <ol style="list-style-type: none"> 2. 헤더 행을 포함한 모든 샘플 데이터를 삭제합니다. 제거되는 데이터와 관련된 쿼리를 삭제할지 여부를 묻는 메시지가 표시되면 아니오를 선택합니다. 쿼리를 삭제하면 Excel 템플릿의 피벗 테이블 기능에 영향을 미칠 수 있습니다. 3. S3 버킷에서 다운로드한 zip 파일의 압축을 풉니다. 4. Excel에서는 .txt를 엽니다. prowler-fullorgresults-accessdeniedfiltered 리소스 스캔 시도와 관련된 오류와 같이 Access Denied 가장 흔하고 실행할 수 없는 오류가 이미 제거되었으므로 이 파일을 사용하는 것이 좋습니다. AWS Control Tower 조사 결과를 필터링하지 않으려면 prowler-fullorgresults.txt 파일을 대신 엽니다. 5. A열을 선택합니다. 6. Windows를 사용하는 경우 Ctrl+C를 입력하고, MacOS를 사용하는 경우 Cmd+C를 입력합니다. 이렇게 하면 모든 데이터가 클립보드에 복사됩니다. 	

작업	설명	필요한 기술
	<p>7. Excel 보고서 템플릿의 Prowler CSV 워크시트에서 A1 셀을 선택합니다.</p> <p>8. Windows를 사용하는 경우 Ctrl+V를 입력하고, MacOS를 사용하는 경우 Cmd+V를 입력합니다. 그러면 조사 결과가 보고서에 붙여넣어집니다.</p> <p>9. 붙여넣은 데이터가 들어 있는 모든 셀이 선택되었는지 확인합니다. 그렇지 않은 경우 A열을 선택합니다.</p> <p>10. 데이터 탭에서 텍스트를 열로를 선택합니다.</p> <p>11. 마법사에서 다음을 수행합니다.</p> <ul style="list-style-type: none"> • 1단계에서는 구분을 선택합니다. • 2단계에서는 구분 기호로 세미콜론을 선택합니다. 데이터 미리 보기 창에서 데이터가 열로 구분되는지 확인합니다. • 3단계에서는 마침을 선택합니다. <p>12. 텍스트 데이터가 여러 열로 구분되었는지 확인합니다.</p> <p>13. 새로운 이름으로 Excel 보고서를 저장합니다.</p> <p>14. 조사 결과에서 Access Denied 오류가 있으면 검색하고 삭제합니다. 프로그래</p>	

작업	설명	필요한 기술
	<p>밍 방식으로 오류를 제거하는 방법에 대한 자세한 지침은 추가 정보 섹션의 프로그래밍 방식 오류 제거를 참조하세요.</p>	

작업	설명	필요한 기술
보고서를 마무리합니다.	<ol style="list-style-type: none"> Findings 워크시트를 선택한 다음 A17 셀을 선택합니다. 이 셀은 피벗 테이블의 헤더입니다. 리본의 PivotTable 도구에서 분석을 선택한 다음 새로 고침에서 모두 새로 고침을 선택합니다. 이를 통해 피벗 테이블을 새로운 데이터 세트 로 업데이트합니다. 기본적으로 Excel에서는 AWS 계정 숫자가 제대로 표시되지 않습니다. 숫자 서식을 수정하려면 다음을 수행합니다. <ul style="list-style-type: none"> Findings 워크시트에서 열 A에 대한 컨텍스트(오른쪽 클릭) 메뉴를 연 다음 Format Cells을 선택합니다. 숫자를 선택하고 소수점 이하 자릿수로 0을 입력합니다. 확인을 선택합니다. <p>참고: AWS 계정 숫자가 하나 이상의 0으로 시작하는 경우 Excel은 자동으로 0을 제거합니다. 보고서에서 12 자리 미만의 계정 번호가 표시되는 경우 번호 앞에 있는 누락된 자릿수는 0입니다.</p> (선택 사항) 조사 결과를 더욱 쉽게 읽을 수 있도록 필드 	일반 AWS

작업	설명	필요한 기술
	<p>를 축소할 수 있습니다. 다음을 수행합니다.</p> <ul style="list-style-type: none"> • Findings 워크시트에서 커서를 18행과 19행 사이의 선(크리티컬 헤더와 첫 번째 조사 결과 사이의 스페이스)으로 이동하면 커서 아이콘이 아래쪽을 가리키는 작은 화살표로 변경됩니다. • 모든 조사 결과 필드를 선택하려면 클릭합니다. • 컨텍스트(오른쪽 클릭) 메뉴를 열고 확장/축소를 찾은 다음 축소를 선택합니다. <p>5. 평가에 대한 자세한 내용은 Findings, Severity 및 Pass Fail 워크시트를 검토합니다.</p> <p>6. zip 파일의 Results-Visualization-<date-of-scan> 폴더에서 자동으로 생성된 그래프와 차트를 검토하여 시각화를 통해 보고서를 개선하는 데 사용할 수 있습니다.</p>	

(선택 사항) 코드 리포지토리의 Prowler 또는 리소스 업데이트

작업	설명	필요한 기술
Prowler를 업데이트합니다.	<p>Prowler를 최신 버전으로 업데이트하려면 다음을 수행합니다.</p> <ol style="list-style-type: none"> 1. Session Manager를 사용하여 Prowler용 EC2 인스턴스에 연결합니다. 자세한 지침은 Session Manager를 사용하여 Linux 인스턴스에 연결을 참조하세요. 2. 다음 명령을 입력합니다. <pre style="border: 1px solid #ccc; border-radius: 10px; padding: 10px; margin: 10px 0;">sudo -i pip3 install --upgrade prowler</pre>	일반 AWS
prowler_scan.sh 스크립트를 업데이트합니다.	<p>prowler_scan.sh 스크립트를 리포지토리의 최신 버전으로 업데이트하고자 한다면 다음을 수행합니다.</p> <ol style="list-style-type: none"> 1. Session Manager를 사용하여 Prowler용 EC2 인스턴스에 연결합니다. 자세한 지침은 Session Manager를 사용하여 Linux 인스턴스에 연결을 참조하세요. 2. 다음 명령을 입력합니다. <pre style="border: 1px solid #ccc; border-radius: 10px; padding: 10px; margin: 10px 0;">sudo -i</pre> <ol style="list-style-type: none"> 3. Prowler 스크립트 디렉터리로 이동합니다. 	일반 AWS

작업	설명	필요한 기술
	<pre>cd /usr/local/prowler</pre> <p>4. 사용자 지정 변경 사항을 최신 버전에 병합할 수 있도록 다음 명령을 입력하여 로컬 스크립트를 stash합니다.</p> <pre>git stash</pre> <p>5. 다음 명령을 입력하여 최신 버전의 스크립트를 다운로드합니다.</p> <pre>git pull</pre> <p>6. 다음 명령을 입력하여 사용자 지정 스크립트를 최신 버전의 스크립트와 통합합니다.</p> <pre>git stash pop</pre> <p>참고: GitHub 리포지토리에 없는 로컬에서 생성된 파일과 관련된 경고 (예: 보고서 찾기) 가 표시될 수 있습니다. 로컬에서 숨긴 변경 내용이 다시 병합된 것으로 <code>prowler_scan.sh</code>에 표시되는 한 이러한 내용은 무시해도 됩니다.</p>	

(선택 사항)정리

작업	설명	필요한 기술
<p>배포된 모든 리소스를 삭제합니다.</p>	<p>리소스는 계정에 배포된 상태로 둘 수 있습니다. 사용하지 않을 때 EC2 인스턴스를 종료하고 S3 버킷을 비워 두면 향후 스캔을 위해 리소스를 유지 관리하는 데 드는 비용을 줄일 수 있습니다.</p> <p>모든 리소스의 프로비저닝을 해제하려면 다음을 수행합니다.</p> <ol style="list-style-type: none"> 1. 관리 계정에 프로비저닝된 IAM-ProwlerExecRole 스택을 삭제합니다. 지침은 CloudFormation 설명서의 스택 삭제를 참조하십시오. 2. 조직의 관리 계정 또는 위임된 관리자 계정에 프로비저닝된 IAM-ProwlerExecRole 스택 세트를 삭제합니다. 지침은 설명서의 스택 세트 삭제를 CloudFormation 참조하십시오. 3. prowler-output S3 버킷의 모든 객체를 삭제합니다. 자세한 지침은 Amazon S3 설명서의 객체 삭제를 참조하세요. 4. 관리 계정에 프로비저닝된 Prowler-Resources 	<p>AWS DevOps</p>

작업	설명	필요한 기술
	스택을 삭제합니다. 지침은 설명서의 스택 삭제 를 CloudFormation 참조하십시오.	

문제 해결

문제	Solution
세션 관리자를 사용하여 EC2 인스턴스에 연결합니다.	SSM 에이전트는 Systems Manager 엔드포인트와 통신할 수 있어야 합니다. 다음을 따릅니다. <ol style="list-style-type: none"> 1. EC2 인스턴스가 배포된 서브넷이 인터넷에 액세스할 수 있는지 확인합니다. 2. EC2 인스턴스를 재부팅합니다.
스택 세트를 배포할 때 CloudFormation 콘솔에 배포하라는 메시지가 표시됩니다. Enable trusted access with AWS Organizations to use service-managed permissions	이는 과 사이에 AWS Organizations 신뢰할 수 있는 액세스가 활성화되지 않았음을 나타냅니다. CloudFormation 서비스 관리형 스택 세트를 배포하려면 신뢰할 수 있는 액세스가 필요합니다. 신뢰할 수 있는 액세스를 활성화하는 버튼을 선택합니다. 자세한 내용은 CloudFormation 설명서의 신뢰할 수 있는 액세스 활성화를 참조하십시오.

관련 리소스

AWS 설명서

- AWS(AWS 규범적 지침) [에 대한 보안 통제 구현](#)

기타 리소스

- [프롤러 \(\)](#) GitHub

추가 정보

프로그래밍 방식으로 오류 제거

결과에 Access Denied 오류가 있는 경우 조사 결과에서 오류를 제거해야 합니다. 이러한 오류는 일반적으로 Prowler가 특정 리소스를 평가하지 못하게 하는 외부 영향 권한 때문입니다. 예를 들어, 프로비저닝된 S3 버킷을 검토할 때 일부 검사가 실패합니다. AWS Control Tower 프로그래밍 방식으로 이러한 결과를 추출하고 필터링된 결과를 새로운 파일로 저장할 수 있습니다.

다음 명령은 단일 텍스트 문자열(패턴) 이 포함된 행을 제거한 다음 결과를 새로운 파일로 출력합니다.

- Linux 또는 MacOS(Grep)용

```
grep -v -i "Access Denied getting bucket" myoutput.csv > myoutput_modified.csv
```

- 윈도우용 () PowerShell

```
Select-String -Path myoutput.csv -Pattern 'Access Denied getting bucket' -NotMatch > myoutput_modified.csv
```

다음 명령은 1개 이상의 텍스트 문자열과 일치하는 행을 제거한 다음 결과를 새로운 파일로 출력합니다.

- Linux 또는 MacOS용(문자열 사이에 이스케이프 처리된 파이프 사용)

```
grep -v -i 'Access Denied getting bucket\|Access Denied Trying to Get' myoutput.csv > myoutput_modified.csv
```

- Windows용(문자열 사이에 쉼표 사용)

```
Select-String -Path myoutput.csv -Pattern 'Access Denied getting bucket', 'Access Denied Trying to Get' -NotMatch > myoutput_modified.csv
```

보고서 예

다음 이미지는 통합된 Prowler 결과 보고서에 있는 Findings 워크시트의 예입니다.

다음 이미지는 통합 Prowler 조사 결과 보고서에 있는 Pass Fail 워크시트의 예입니다. (기본적으로 합격 결과는 출력에서 제외됩니다.)

다음 이미지는 통합 Prowler 조사 결과 보고서의 Severity 워크시트 예시입니다.

AWS Config 및 AWS Systems Manager로 사용하지 않는 Amazon Elastic Block Store(Amazon EBS) 볼륨 삭제

작성자: 상카르 산구보틀라(AWS)

환경: PoC 또는 파일럿

기술: 보안, 자격 증명, 규정 준수, 관리 및 거버넌스, 비용 관리

AWS 서비스: AWS Config, AWS Systems Manager

요약

Amazon Elastic Block Store(Amazon EBS) 볼륨의 수명 주기는 일반적으로 볼륨이 연결된 Amazon Elastic Compute Cloud(Amazon EC2) 인스턴스의 수명 주기와는 별개입니다. 시작 시 종료 시 삭제 옵션을 선택하지 않는 한, EC2 인스턴스를 종료하면 EBS 볼륨이 분리되지만 삭제되지는 않습니다. 특히 EC2 인스턴스를 시작하고 종료하는 것이 일반적인 개발 및 테스트 환경에서는 이로 인해 많은 수의 EBS 볼륨을 사용하지 않을 수 있습니다. EBS 볼륨은 사용 여부에 관계없이 Amazon Web Services(AWS) 계정에 요금이 누적됩니다. 이러한 볼륨을 삭제하면 AWS 계정의 비용을 최적화하는데 도움이 될 수 있습니다. 또한 미사용 EBS 볼륨을 삭제하는 것은 해당 볼륨에서 사용되지 않는 잠재적으로 민감한 데이터에 대한 액세스를 방지하는 보안 모범 사례입니다.

AWS Config는 규정을 준수하지 않는 리소스를 수동 또는 자동으로 문제 해결을 하는 데 도움이 될 수 있습니다. 이 패턴은 계정에서 사용하지 않는 Amazon EBS 볼륨을 삭제하는 AWS Config 규칙 및 자동 문제 해결 작업을 구성하는 방법을 설명합니다. 문제 해결 작업은 AWS Systems Manager의 기능인 사전 정의된 Automation 런북입니다. 볼륨을 삭제하기 전에 볼륨의 스냅샷을 생성하도록 런북을 구성할 수 있습니다.

사전 조건 및 제한 사항

사전 조건

- 활성 상태의 AWS 계정
- AWS Systems Manager 기능인 AWSConfigRemediation-DeleteUnusedEBSVolume Automation 런북을 실행할 수 있는 AWS Identity and Access Management(IAM) 권한. 자세한 내용은 [AWSConfigRemediation-DeleteUnused EBSVolume](#)의 필수 IAM 권한을 참조하십시오.
- 사용하지 않는 하나 또는 하나 이상의 Amazon EBS 볼륨.

제한 사항

- 사용하지 않는 Amazon EBS 볼륨은 available 상태에 있어야 합니다.

아키텍처

기술 스택

- AWS Config
- Amazon EBS
- Systems Manager
- Systems Manager Automation

대상 아키텍처

1. AWS Config 규칙은 EBS 볼륨을 평가합니다.
2. 규칙은 규정 준수 및 미준수 리소스 목록을 반환합니다. available 상태에 있는 EBS 볼륨(미사용 볼륨)은 규정을 준수하지 않는 것으로 판단됩니다.
3. AWS Config는 Automation 런북을 자동으로 시작합니다.
4. 구성된 경우 Systems Manager는 사용하지 않는 볼륨을 삭제하기 전에 해당 볼륨의 스냅샷을 생성합니다.
5. Systems Manager는 사용하지 않는 EBS 볼륨을 삭제합니다.

자동화 및 규모 조정

이 솔루션은 조직의 모든 계정에 적용할 수 있습니다. 자세한 내용은 AWS Config 설명서의 [조직 내 모든 계정에 걸쳐 규칙 관리](#)를 참조하세요.

도구

- [AWS Config](#)는 사용자의 AWS 계정의 리소스와 구성 방법을 상세하게 볼 수 있도록 합니다. 리소스가 서로 관련되는 방식과 리소스의 구성이 시간이 지남에 따라 변경된 방식을 식별하는 데 도움이 됩니다.

- [AWS Systems Manager](#)는 AWS 클라우드에서 실행되는 애플리케이션과 인프라를 관리하는 데 도움이 됩니다. 애플리케이션 및 리소스 관리를 간소화하고, 운영 문제의 감지 및 해결 시간을 단축하며, AWS 리소스를 규모에 따라 안전하게 관리하는 데 도움이 됩니다.
- [AWS Systems Manager Automation](#)은 여러 AWS 서비스의 일반적인 유지 관리, 배포 및 개선 작업을 간소화합니다.

에픽

AWS Config 규칙 구성

작업	설명	필요한 기술
Automation 런북의 역할을 생성합니다.	AssumeRole (이)라고 하는 역할을 생성합니다. Systems Manager Automation은 이 역할을 사용하여 런북을 실행합니다. 지침은 Systems Manager 설명서의 자동화에 대한 서비스 역할(수입 역할) 액세스 구성 을 참조하세요.	AWS 시스템 관리자
AWS Config 레코더를 켭니다.	AWS Config 설명서의 콘솔을 이용한 AWS Config 설정 지침을 따라서 AWS Config가 실행 중이고 Amazon EBS 볼륨을 기록하도록 구성되어 있는지 확인하십시오.	AWS 시스템 관리자
규칙을 실행합니다.	<ol style="list-style-type: none"> 1. ec2-volume-inuse-check 규칙을 실행하려면 AWS Config 설명서의 리소스 평가에 있는 지침을 따르십시오. 평가가 완료될 때까지 기다립니다. 2. 규칙 페이지에서 ec2-volume-inuse-check 규칙을 선택한 다음 범위 내 	AWS 시스템 관리자

작업	설명	필요한 기술
	<p>리소스에서 미준수를 선택합니다.</p> <p>3. 평가 결과에 미사용 Amazon EBS 볼륨이 하나 이상 있는지 확인합니다.</p>	

사용하지 않는 Amazon EBS 볼륨의 자동 문제 해결 구성

작업	설명	필요한 기술
자동 수정 작업을 추가합니다.	<ol style="list-style-type: none"> 규칙 페이지에서 ec2-volume-inuse-check 규칙을 선택합니다. AWS Config 설명서의 자동 문제 해결 설정 지침을 따르세요. 다음을 참고합니다. 문제 해결 작업 세부 정보 섹션에서 AWSConfig Remediation-Delete UnusedEBSVolume 을 (를) 선택합니다. <ul style="list-style-type: none"> 리소스 ID 파라미터를 선택한 다음 목록에서 선택합니다. Volumeld 런타임 시 이 파라미터는 미준수 EBS 볼륨의 ID로 대체됩니다. 파라미터 섹션에서 다음 파라미터 값을 입력합니다. <ul style="list-style-type: none"> CreateSnapshot - (선택 사항) true(으)로 설정하면 삭제 전에 자 	AWS 시스템 관리자

작업	설명	필요한 기술
	<p>동화가 EBS 볼륨의 스냅샷을 생성합니다.</p> <ul style="list-style-type: none"> AutomationAssumeRole - 이전에 생성한 AssumeRole 서비스 역할의 Amazon 리소스 이름(ARN)을 입력합니다. 	
<p>AWS Config 규칙의 자동 문제 해결을 테스트합니다.</p>	<ol style="list-style-type: none"> AWS Config 콘솔의 규칙페이지에서 ec2-volume-inuse-check 규칙을 선택합니다. 작업 메뉴에서 재평가를 선택합니다. 규칙이 규정을 준수하지 않는 리소스를 평가하도록 허용한 다음 사용하지 않는 Amazon EBS 볼륨이 삭제되었는지 확인합니다. 	<p>AWS 시스템 관리자</p>

문제 해결

문제	Solution
<p>AWS Config는 리소스 상태를 정확하게 반영하지 않습니다.</p>	<p>AWS Config가 리소스 상태를 업데이트하지 않는 경우가 있습니다. AWS Config 설정 페이지에서 레코더를 꺾다가 다시 꺾습니다. 레코더는 리소스 상태를 캡처합니다. 새로 만들거나 삭제한 리소스의 경우, 레코더에 현재 상태가 반영되는 데 시간이 걸릴 수 있습니다. EBS 볼륨 상태에 대</p>

문제	Solution
	한 자세한 내용은 Amazon EC2 설명서의 블록 상태 를 참조하세요.

관련 리소스

- [AWSConfigRemediation- DeleteUnused EBS 볼륨 런북](#)
- [ec2 - 규칙 volume-inuse-check](#)
- [AWS Config 규칙에 따른 미준수 AWS 리소스 문제 해결](#)

AWS CDK 및 AWS를 사용하여 AWS 컨트롤 타워 컨트롤을 배포하고 관리합니다. CloudFormation

작성자: Iker Reina Fuente와 Ivan Girardi

코드 리포지토리: [aws-control-tower-controls-cdk](#)

환경: 프로덕션

기술: 보안, 자격 증명, 규정 준수, 클라우드 네이티브, 관리 및 거버넌스

AWS 서비스: AWS CloudFormation, AWS 컨트롤 타워, AWS Organizations, AWS CDK

요약

이 패턴은 AWS CloudFormation 및 AWS Cloud Development Kit (AWS CDK) 를 사용하여 예방적, 탐지적, 사전 예방적 AWS Control Tower 제어를 코드형 인프라 (IaC) 로 구현하고 관리하는 방법을 설명합니다. [제어](#) (가드레일로 알려지기도 함)는 전반적인 AWS 환경에 대한 지속적인 거버넌스를 제공하는 상위 수준 규칙입니다. 예를 들어 컨트롤을 사용하여 AWS 계정에 대한 로깅을 요구한 다음 특정 보안 관련 이벤트가 발생할 경우 자동 알림을 구성할 수 있습니다.

AWS Control Tower를 사용하면 AWS 리소스를 관리하고 여러 AWS 계정의 규정 준수를 모니터링하는 예방적, 탐지적, 사전 예방적 제어를 구현할 수 있습니다. 각 컨트롤은 단일 규칙을 적용합니다. 이 패턴에서는 제공된 IaC 템플릿을 사용하여 환경에 배포할 컨트롤을 지정합니다.

AWS Control Tower 규제 항목은 전체 [조직 유닛\(OU\)](#)에 적용되며, 제어는 OU 내의 모든 AWS 계정에 영향을 미칩니다. 따라서 사용자가 랜딩 존의 어떤 계정에서든 작업을 수행하는 경우 해당 작업에는 OU를 제어하는 컨트롤이 적용됩니다.

AWS Control Tower 제어를 구현하면 AWS 랜딩 존을 위한 강력한 보안 기반을 구축하는 데 도움이 됩니다. 이 패턴을 사용하여 AWS CDK를 통해 CloudFormation IaC로 컨트롤을 배포하면 랜딩 존의 컨트롤을 표준화하고 더 효율적으로 배포 및 관리할 수 있습니다. 이 솔루션은 배포 중에 [cdk_nag](#)를 사용하여 CDK 애플리케이션을 스캔합니다. 이 도구는 애플리케이션이 모범 사례를 준수하는지 확인합니다.

AWS 컨트롤 타워 컨트롤을 IaC로 배포하려면 AWS CDK 대신 HashiCorp Terraform을 사용할 수도 있습니다. 자세한 내용은 [Terraform을 사용한 Control Tower 제어 배포 및 관리](#)를 참조하십시오.

대상 청중

이 패턴은 AWS 컨트롤 타워 CloudFormation, AWS CDK 및 AWS Organizations를 사용해 본 경험이 있는 사용자에게 권장됩니다.

사전 조건 및 제한 사항

사전 조건

- AWS Organizations와 Control Tower 랜딩 존에서 하나의 조직으로 관리되는 활성 AWS 계정. 지침은 [계정 구조 생성](#)(Well-Architected Labs)을 참조하십시오.
- AWS Command Line Interface(AWS CLI), [설치](#) 및 [구성됨](#)
- 노드 패키지 관리자(npm), CDK용으로 [설치 및 구성됨](#).
- CDK를 위한 [사전 조건](#).
- 배치 계정에서 Identity and Management (IAM) 역할을 맡을 수 있는 권한입니다.
- CDK를 부트스트랩하는 데 사용할 수 있는 조직의 관리 계정에서 IAM 역할을 맡을 수 있는 권한. 역할에는 CloudFormation 리소스를 수정하고 배포할 수 있는 권한이 있어야 합니다. 자세한 내용은 AWS CDK 설명서의 [부트스트래핑](#)을 참조하십시오.
- 조직의 관리 계정에서 IAM 역할과 정책을 생성할 수 있는 권한. 자세한 내용은 IAM 사용 설명서에서 [Permissions required to access IAM 리소스에 액세스하는 데 필요한 권한](#)을 참조하십시오.
- 식별자CT.CLOUDFORMATION.PR.1을 사용하여 서비스 제어 정책(SCP) 기반 제어를 적용합니다. 사전 예방적 제어를 배포하려면 이 SCP를 활성화해야 합니다. 지침은 [AWS CloudFormation 레지스트리 내 리소스 유형, 모듈 및 후크 관리 금지](#) 섹션을 참조하십시오.

제한 사항

- 이 패턴은 배포 계정에서 조직의 관리 계정에 이르기까지 AWS 계정 전반적으로 이 솔루션을 배포하기 위한 지침을 제공합니다. 테스트 목적으로 이 솔루션을 관리 계정에 직접 배포할 수 있지만 이 구성에 대한 지침은 명시적으로 제공되지 않습니다.

제품 버전

- Python 버전 3.9 이상.
- npm 버전 8.9.0 이상

아키텍처

대상 아키텍처

이 섹션에서는 이 솔루션과 샘플 코드를 통해 설정된 아키텍처에 대한 종합적 개요를 제공합니다. 다음 다이어그램은 OU의 다양한 계정 전반적으로 배포된 제어를 보여줍니다.

AWS Control Tower 컨트롤은 행동 및 지침에 따라 분류됩니다.

제어 동작에는 세 가지 기본 유형이 있습니다.

1. 예방 제어는 조치가 발생하지 않도록 설계되었습니다. 이는 AWS Organizations의 [서비스 제어 정책 \(SCP\)](#)을 통해 구현됩니다. 예방적 제어의 상태는 적용 또는 활성화되지 않음입니다. 예방적 제어는 모든 AWS 리전에서 지원됩니다.
2. Detective Controls는 특정 이벤트가 발생할 때 이를 탐지하고 해당 작업을 기록하도록 설계되었습니다. CloudTrail 이는 [AWS Config](#) 규칙을 사용하여 구현됩니다. 탐지적 제어의 상태는 정상, 위반 또는 활성화되지 않음입니다. 탐지 제어는 AWS Control Tower에서 지원하는 AWS 리전에만 적용됩니다.
3. 사전 예방적 제어는 AWS에서 프로비저닝할 리소스를 스캔하고 해당 리소스가 회사 정책 CloudFormation 및 목표를 준수하는지 확인합니다. 규정을 준수하지 않는 리소스는 프로비저닝되지 않습니다. 이들은 [AWS CloudFormation 후크로](#) 구현됩니다. 선제적 제어의 상태는 합격, 불합격 또는 건너뛰기입니다.

제어 지침에는 각 제어를 OU에 적용하는 방법에 대한 권장 사례를 참조합니다. Control Tower에서는 세 가지 범주 지침, 즉 필수, 적극 권장 및 선택을 제공합니다. 컨트롤의 지침은 동작과 무관합니다. 자세한 내용은 [제어 동작 및 지침](#)을 참조하십시오.

도구

서비스

- [AWS Cloud Development Kit\(AWS CDK\)](#)는 AWS 클라우드 인프라를 코드로 정의하고 프로비저닝하는 데 도움이 되는 소프트웨어 개발 프레임워크입니다. [CDK Toolkit](#)은 CDK 앱과 상호 작용하는 기본 도구입니다.
- [AWS는 AWS CloudFormation](#) 리소스를 설정하고, 빠르고 일관되게 프로비저닝하고, 전체 AWS 계정 및 지역의 수명 주기 전반에 걸쳐 리소스를 관리할 수 있도록 지원합니다.

- [Config](#)는 사용자의 AWS 계정에 있는 리소스와 그 구성 방식을 자세히 보여줍니다. 리소스가 서로 관련되는 방식과 리소스의 구성이 시간이 지남에 따라 변경된 방식을 식별하는 데 도움이 됩니다.
- [AWS Control Tower](#)는 권장 모범 사례에 따라 AWS 다중 계정 환경을 설정하고 관리할 수 있도록 지원합니다.
- [Organizations](#)는 사용자가 생성하고 중앙에서 관리하는 조직으로 여러 AWS 계정을 통합할 수 있는 계정 관리 서비스입니다.

기타 도구

- [cdk_nag](#)는 규칙 팩의 조합을 사용하여 Cloud Development Kit(CDK) 애플리케이션이 모범 사례를 준수하는지 확인하는 오픈 소스 도구입니다.
- [npm](#)은 Node.js 환경에서 실행되는 소프트웨어 레지스트리로, 패키지를 공유 또는 대여하고 개인 패키지의 배포를 관리하는 데 사용됩니다.
- [Python](#)은 범용 컴퓨터 프로그래밍 언어입니다.

코드 리포지토리

이 패턴의 코드는 AWS [CDK 리포지토리를 사용한 AWS Control Tower GitHub 배포 컨트롤에서](#) 사용할 수 있습니다. cdk.json 파일을 사용하여 CDK 앱과 상호 작용하고, package.json 파일을 사용하여 npm 패키지를 설치합니다.

모범 사례

- [최소 권한 원칙](#) (IAM 설명서)을 따르세요. 이 패턴에서 제공되는 IAM 정책 및 신뢰 정책 샘플에는 필요한 최소 권한이 포함되며, 관리 계정에서 생성되는 AWS CDK 스택은 이러한 권한으로 제한됩니다.
- [Control Tower 관리자를 위한 모범 사례](#)(Control Tower 설명서)를 따르십시오.
- [CDK를 사용한 클라우드 인프라 개발 및 배포 모범 사례](#)(CDK 설명서)를 따르십시오.
- CDK를 부트스트래핑할 때는 부트스트랩 템플릿을 사용자 지정하여 관리 계정의 모든 리소스를 읽고 쓸 수 있어야 하는 신뢰할 수 있는 계정과 정책을 정의합니다. 자세한 내용은 [부트스트래핑 사용자 지정하기](#)를 참조하십시오.
- [cfn_nag](#)와 같은 코드 분석 도구를 사용하여 생성된 템플릿을 스캔합니다. CloudFormation cfn-nag 도구는 CloudFormation 템플릿에서 인프라가 안전하지 않음을 나타낼 수 있는 패턴을 찾습니다. [클라우드포메이션-include 모듈을 사용하여 cdk-nag를 사용하여 템플릿을 확인할 수도 있습니다.](#) [CloudFormation](#)

에픽

제어를 활성화할 준비를 합니다.

작업	설명	필요한 기술
관리 계정에서 IAM 역할을 생성합니다.	<p>1. 추가 정보 섹션의 IAM 정책에 정의된 권한을 사용하여 관리 계정에서 IAM 정책을 생성합니다. 지침은 IAM 설명서의 IAM 정책 생성을 참조하십시오. 정책의 Amazon 리소스 이름(ARN)에 유의합니다. 다음은 ARN 예제입니다.</p> <pre>arn:aws:iam::<MANAGEMENT-ACCOUNT-ID>:policy/<POLICY-NAME></pre> <p>2. 관리 계정에서 IAM 역할을 생성하고, 이전 단계에서 생성한 IAM 권한 정책을 연결하며, 추가 정보 섹션의 신뢰 정책에서 사용자 지정 신뢰 정책을 연결합니다. 지침은 IAM 설명서에서 사용자 지정 신뢰 정책을 사용한 역할 생성을 참조하십시오. 다음은 새로운 역할에 대한 ARN의 예입니다.</p> <pre>arn:aws:iam::<MANAGEMENT-ACCOUNT-ID>:role/<ROLE-NAME></pre>	DevOps 엔지니어, 일반 AWS

작업	설명	필요한 기술
CDK를 부트스트랩합니다.	<ol style="list-style-type: none"> 1. 관리 계정에서 CDK를 부트스트랩할 수 있는 권한이 있는 역할을 떠맡습니다. 2. 다음 명령을 입력하여 다음 값을 바꿉니다. <ul style="list-style-type: none"> • <MANAGEMENT-ACCOUNT-ID> 은(는) 조직의 관리 계정의 ID입니다. • <AWS-CONTROL-TOWER-REGION> (은)는 Control Tower가 배포된 리전입니다. 완전한 리전 코드 목록은 일반 참조의 리전 엔드포인트를 참조하십시오. • <DEPLOYMENT-ACCOUNT-ID> (은)는 배포 계정의 ID입니다. • <DEPLOYMENT-ROLE-NAME> (은)는 배포 계정에서 사용하는 IAM 역할의 이름입니다. • <POLICY-NAME> (은)는 관리 계정에서 생성한 정책의 이름입니다. <pre data-bbox="633 1507 1029 1837"> \$ npx cdk bootstrap aws://<MANAGEMENT-ACCOUNT-ID>/<AWS-CONTROL-TOWER-REGION> \ --trust arn:aws:iam::<DEPLOYMENT-ACCOUNT-ID>:role/<DE </pre>	DevOps 엔지니어, 일반 AWS, Python

작업	설명	필요한 기술
<p>리포지토리를 복제합니다.</p>	<pre>PLOYMENT-ROLE-NAME> \ --cloudformation- execution-policies arn:aws:iam::<MANA GEMENT-ACCOUNT-ID> :policy/<POLICY-NA ME></pre> <p>bash 셸에서 다음 명령을 입력합니다. 이렇게 하면 AWS CDK 리포지토리를 사용하여 AWS Control Tower 배포 컨트롤이 복제됩니다. GitHub</p> <pre>git clone https://g ithub.com/aws-samp les/aws-control-to wer-controls-cdk.git</pre>	<p>DevOps 엔지니어, 일반 AWS</p>

작업	설명	필요한 기술
CDK 구성 파일을 편집합니다.	<ol style="list-style-type: none"> 1. 복제된 리포지토리에서 constants.py 파일을 엽니다. 2. ACCOUNT_ID 파라미터에서 관리 계정의 ID를 입력합니다. 3. <AWS-CONTROL-TOWER-REGION> 파라미터에서 Control Tower가 배포된 AWS 리전을 입력합니다. 4. ROLE_ARN 파라미터에서, 관리 계정에서 생성한 역할의 ARN을 입력합니다. 5. GUARDRAILS_CONFIGURATION 섹션의 EnableControl 파라미터에서 제어 API 식별자를 입력합니다. 식별자를 큰따옴표로 입력하고 여러 식별자를 쉼표로 구분합니다. 각 컨트롤에는 AWS Control Tower를 사용할 수 있는 각 리전에 대한 고유한 API 식별자가 있습니다. 컨트롤 식별자를 찾으려면 다음을 수행합니다. <ol style="list-style-type: none"> a. 제어 메타데이터의 테이블에서 활성화하려는 제어를 찾습니다. b. 제어 API 식별자의 리전별 열에서 API를 직접 호출하려는 지역의 API 식별자(예arn:aws:controltower:us-east-1::control/ 	

작업	설명	필요한 기술
	<p>AWS-GR_ENCRYPTED_VOLUMES)를 찾습니다.</p> <p>c. 리전의 식별자에서 제어 식별자를 추출합니다(예AWS-GR_ENCRYPTED_VOLUMES).</p> <p>6. GUARDRAILS_CONFIGURATION 섹션의 OrganizationalUnitIds 파라미터에서 제어 (예: ou-1111-11111111)를 활성화하려는 조직 단위의 ID를 입력합니다. ID를 큰 따옴표로 입력하고 여러 ID는 쉼표로 구분합니다. OU ID를 검색하는 방법에 대한 자세한 내용은 OU 세부 정보 보기를 참조하십시오.</p> <p>7. constants.py 파일을 저장하고 닫습니다. 업데이트된 constants.py 파일의 예는 이 패턴의 추가 정보를 참조하십시오.</p>	

관리 계정의 제어 활성화

작업	설명	필요한 기술
<p>배포 계정에서 IAM 역할을 수 있습니다.</p>	<p>배포 계정에서, 관리 계정의 CDK 스택을 배포할 수 있는 권한이 있는 IAM 역할을 떠맡습니다. AWS CLI에서 IAM 역할을 맡는 방법에 대한 자세한 내</p>	<p>DevOps 엔지니어, 일반 AWS</p>

작업	설명	필요한 기술
	<p>용은 AWS CLI에서의 IAM 역할 사용을 참조하십시오.</p>	
<p>환경을 활성화합니다.</p>	<p>Linux 또는 MacOS를 사용하는 경우:</p> <ol style="list-style-type: none"> 다음 명령을 입력하여 가상 환경을 생성합니다. <pre data-bbox="630 583 1029 701">\$ python3 -m venv .venv</pre> <ol style="list-style-type: none"> 가상 환경이 생성된 후에, 다음 명령을 입력하여 활성화합니다. <pre data-bbox="630 888 1029 1005">\$ source .venv/bin/ activate</pre> <p>Windows를 사용하는 경우:</p> <ol style="list-style-type: none"> 다음 명령을 입력하여 가상 환경을 활성화합니다. <pre data-bbox="630 1276 1029 1394">% .venv\Scripts\acti vate.bat</pre>	<p>DevOps 엔지니어, 일반 AWS</p>
<p>종속성을 설치합니다.</p>	<p>가상 환경이 활성화된 후, 다음 명령을 입력하여 install_deps.sh 스크립트를 실행합니다. 이 스크립트는 필수 종속성을 설치합니다.</p> <pre data-bbox="594 1696 1029 1814">\$./scripts/install_ deps.sh</pre>	<p>DevOps 엔지니어, 일반 AWS, Python</p>

작업	설명	필요한 기술
스택을 배포합니다.	다음 명령을 입력하여 CloudFormation 스택을 합성하고 배포합니다. <pre>\$ npx cdk synth \$ npx cdk deploy</pre>	DevOps 엔지니어, 일반 AWS, Python

관련 리소스

AWS 설명서

- [제어에 대한 정보](#)(Control Tower 설명서)
- [제어 라이브러리](#)(Control Tower 설명서)
- [CDK Toolkit 명령](#)(CDK 설명서)
- [Terraform을 사용한 제어 배포 및 관리](#) (권장 가이드)

기타 리소스

- [Python](#)

추가 정보

예제 constants.py 파일

다음은 업데이트된 constants.py 파일의 예입니다.

```
ACCOUNT_ID = 111122223333
AWS_CONTROL_TOWER_REGION = us-east-2
ROLE_ARN = "arn:aws:iam::111122223333:role/CT-Controls-Role"
GUARDRAILS_CONFIGURATION = [
    {
        "Enable-Control": {
            "AWS-GR_ENCRYPTED_VOLUMES",
            ...
        },
    },
]
```

```

    "OrganizationalUnitIds": ["ou-1111-11111111", "ou-2222-22222222"...],
  },
  {
    "Enable-Control": {
      "AWS-GR_SUBNET_AUTO_ASSIGN_PUBLIC_IP_DISABLED",
      ...
    },
    "OrganizationalUnitIds": ["ou-2222-22222222"...],
  },
]

```

IAM 정책

다음 샘플 정책은 배포 계정에서 관리 계정으로 AWS CDK 스택을 배포할 때 Control Tower 제어를 활성화 또는 비활성화하는 데 필요한 최소 작업을 허용합니다.

```

{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "controltower:EnableControl",
        "controltower:DisableControl",
        "controltower:GetControlOperation",
        "controltower:ListEnabledControls",
        "organizations:AttachPolicy",
        "organizations:CreatePolicy",
        "organizations>DeletePolicy",
        "organizations:DescribeOrganization",
        "organizations:DescribeOrganizationalUnit",
        "organizations:DetachPolicy",
        "organizations:ListAccounts",
        "organizations:ListAWSServiceAccessForOrganization",
        "organizations:ListChildren",
        "organizations:ListOrganizationalUnitsForParent",
        "organizations:ListParents",
        "organizations:ListPoliciesForTarget",
        "organizations:ListRoots",
        "organizations:UpdatePolicy",
        "ssm:GetParameters"
      ],
      "Resource": "*"
    }
  ]
}

```

```

    }
  ]
}

```

신뢰 정책

다음 사용자 지정 신뢰 정책은 배포 계정의 특정 IAM 역할이 관리 계정의 IAM 역할을 떠맡을 수 있도록 허용합니다. 다음을 바꿉니다.

- <DEPLOYMENT-ACCOUNT-ID>(은)는 배포 계정의 ID입니다.
- <DEPLOYMENT-ROLE-NAME>(은)는 관리 계정에서 역할을 맡을 수 있도록 허용되는 배포 계정의 역할 이름입니다.

```

{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Principal": {
        "AWS": "arn:aws:iam::<DEPLOYMENT-ACCOUNT-ID>:role/<DEPLOYMENT-ROLE-NAME>"
      },
      "Action": "sts:AssumeRole",
      "Condition": {}
    }
  ]
}

```

Terraform을 사용하여 AWS Control Tower 컨트롤을 배포하고 관리합니다.

작성자: Iker Reina Fuente와 Ivan Girardi

<p>코드 리포지토리: Terraform을 사용하여 AWS Control Tower 컨트롤을 배포하고 관리합니다.</p>	<p>환경: 프로덕션</p>	<p>기술: 보안, 자격 증명, 규정 준수, 클라우드 네이티브, 인프라, 관리 및 거버넌스</p>
<p>워크로드: 오픈 소스</p>	<p>AWS 서비스: AWS Control Tower; AWS Organizations</p>	

요약

이 패턴은 AWS Control Tower 제어, HashiCorp Terraform 및 코드형 인프라 (IaC) 를 사용하여 예방, 탐지 및 사전 예방 보안 제어를 구현하고 관리하는 방법을 설명합니다. [제어](#) (가드레일로 알려지기도 함)는 전반적인 AWS 환경에 대한 지속적인 거버넌스를 제공하는 상위 수준 규칙입니다. 예를 들어 컨트롤을 사용하여 AWS 계정에 대한 로깅을 요구한 다음 특정 보안 관련 이벤트가 발생할 경우 자동 알림을 구성할 수 있습니다.

AWS Control Tower를 사용하면 AWS 리소스를 관리하고 여러 AWS 계정의 규정 준수를 모니터링하는 예방적, 탐지적, 사전 예방적 제어를 구현할 수 있습니다. 각 컨트롤은 단일 규칙을 적용합니다. 이 패턴에서는 제공된 IaC 템플릿을 사용하여 환경에 배포할 컨트롤을 지정합니다.

AWS Control Tower 규제 항목은 전체 [조직 유닛\(OU\)](#)에 적용되며, 제어는 OU 내의 모든 AWS 계정에 영향을 미칩니다. 따라서 사용자가 랜딩 존의 어떤 계정에서든 작업을 수행하는 경우 해당 작업에는 OU를 제어하는 컨트롤이 적용됩니다.

AWS Control Tower 제어를 구현하면 AWS 랜딩 존을 위한 강력한 보안 기반을 구축하는 데 도움이 됩니다. 이 패턴을 사용하여 Terraform을 통해 IaC로 컨트롤을 배포하면 랜딩 존의 컨트롤을 표준화하고 더 효율적으로 배포 및 관리할 수 있습니다.

AWS Control Tower 컨트롤을 IaC로 배포하려면 Terraform 대신 AWS Cloud Development Kit(AWS CDK)를 사용할 수도 있습니다. 자세한 내용은 AWS [CDK 및 AWS를 사용한 AWS 컨트롤 타워 제어 배포 및 관리](#)를 참조하십시오. CloudFormation

대상 고객

이 패턴은 AWS Control Tower, Terraform 및 AWS Organizations를 사용해 본 경험이 있는 사용자에게 권장됩니다.

사전 조건 및 제한 사항

사전 조건

- AWS Organizations와 Control Tower 랜딩 존에서 하나의 조직으로 관리되는 활성 AWS 계정. 지침은 [계정 구조 생성](#)(Well-Architected Labs)을 참조하십시오.
- AWS Command Line Interface(AWS CLI), [설치](#) 및 [구성됨](#)
- 이 패턴을 배포할 권한이 있는 관리 계정의 AWS Identity and Access Management(IAM) 역할. 필요한 권한 및 샘플 정책에 대한 자세한 내용은 이 패턴의 [추가 정보](#) 섹션에서 IAM 역할에 대한 최소 권한 권한을 참조하십시오.
- 관리 계정에서 IAM 역할을 맡을 수 있는 권한.
- 식별자 CT.CLOUDFORMATION.PR.1을 사용하여 서비스 제어 정책(SCP) 기반 제어를 적용합니다. 사전 예방적 제어를 배포하려면 이 SCP를 활성화해야 합니다. 지침은 [AWS CloudFormation 레지스트리 내 리소스 유형, 모듈 및 후크 관리 금지](#) 섹션을 참조하십시오.
- Terraform CLI, [설치됨](#)(Terraform 설명서).
- Terraform AWS Provider, [구성됨](#)(Terraform 설명서)
- Terraform 백엔드, [구성됨](#)(Terraform 설명서)

제품 버전

- AWS Control Tower 버전 3.0 이상
- Terraform 버전 1.5 이상:
- Terraform AWS Provider 버전 4.67 이상

아키텍처

대상 아키텍처

이 섹션에서는 이 솔루션과 샘플 코드를 통해 설정된 아키텍처에 대한 종합적 개요를 제공합니다. 다음 다이어그램은 OU의 다양한 계정 전반적으로 배포된 제어를 보여줍니다.

AWS Control Tower 컨트롤은 행동 및 지침에 따라 분류됩니다.

제어 동작에는 세 가지 기본 유형이 있습니다.

1. 예방 제어는 조치가 발생하지 않도록 설계되었습니다. 이는 AWS Organizations의 [서비스 제어 정책 \(SCP\)](#)을 통해 구현됩니다. 예방적 제어의 상태는 적용 또는 활성화되지 않음입니다. 예방적 제어는 모든 AWS 리전에서 지원됩니다.
2. Detective Controls는 특정 이벤트가 발생할 때 이를 탐지하고 해당 작업을 기록하도록 설계되었습니다. CloudTrail 이는 [AWS Config](#) 규칙을 사용하여 구현됩니다. 탐지적 제어의 상태는 정상, 위반 또는 활성화되지 않음입니다. 탐지 제어는 AWS Control Tower에서 지원하는 AWS 리전에만 적용됩니다.
3. 사전 예방적 제어는 AWS에서 프로비저닝할 리소스를 스캔하고 해당 리소스가 회사 정책 CloudFormation 및 목표를 준수하는지 확인합니다. 규정을 준수하지 않는 리소스는 프로비저닝되지 않습니다. 이들은 [AWS CloudFormation 후크로](#) 구현됩니다. 선제적 제어의 상태는 합격, 불합격 또는 건너뛰기입니다.

제어 지침은 각 제어를 OU에 적용하는 방법에 대한 권장 사항입니다. AWS Control Tower에서는 세 가지 범주 지침, 즉 필수, 적극 권장 및 선택을 제공합니다. 컨트롤의 지침은 동작과 무관합니다. 자세한 내용은 [제어 동작 및 지침](#)을 참조하십시오.

도구

서비스

- [AWS는 AWS CloudFormation](#) 리소스를 설정하고, 빠르고 일관되게 프로비저닝하고, 전체 AWS 계정 및 지역의 수명 주기 전반에 걸쳐 리소스를 관리할 수 있도록 지원합니다.
- [Config](#)는 사용자의 AWS 계정에 있는 리소스와 그 구성 방식을 자세히 보여줍니다. 리소스가 서로 관련되는 방식과 리소스의 구성이 시간이 지남에 따라 변경된 방식을 식별하는 데 도움이 됩니다.
- [AWS Control Tower](#)는 권장 모범 사례에 따라 AWS 다중 계정 환경을 설정하고 관리할 수 있도록 지원합니다.
- [Organizations](#)는 사용자가 생성하고 중앙에서 관리하는 조직으로 여러 AWS 계정을 통합할 수 있는 계정 관리 서비스입니다.

기타 도구

- [HashiCorp Terraform](#)은 코드를 사용하여 클라우드 인프라 및 리소스를 프로비저닝하고 관리하는 데 도움이 되는 오픈 소스 IaC (코드형 인프라) 도구입니다.

코드 리포지토리

이 패턴의 코드는 [Terraform 리포지토리를 사용한 AWS Control Tower 컨트롤 GitHub 배포 및 관리에](#)서 사용할 수 있습니다.

모범 사례

- 이 솔루션을 배포하는 데 사용되는 IAM 역할은 [최소 권한 원칙](#)(IAM 설명서)을 준수해야 합니다.
- [AWS Control Tower 관리자를 위한 모범 사례](#)(AWS Control Tower 설명서)를 따르십시오.

에픽

계정 관리 제어 활성화

작업	설명	필요한 기술
리포지토리를 복제합니다.	<p>bash 셸에서 다음 명령을 입력합니다. 이렇게 하면 Terraform 리포지토리를 사용하여 AWS Control Tower 배포 및 관리 컨트롤을 복제할 수 있습니다.</p> <p>GitHub</p> <pre>git clone https://github.com/aws-samples/aws-control-tower-controls-terraform.git</pre>	DevOps 엔지니어
Terraform 백엔드 구성 파일을 편집합니다.	<ol style="list-style-type: none"> 1. 복제된 리포지토리에서 백엔드.tf 파일을 엽니다. 2. 파일을 편집하여 Terraform 백엔드 구성을 설정합니다. 이 파일에서 정의하는 구성 	DevOps 엔지니어, 테라폼

작업	설명	필요한 기술
	<p>은 환경에 따라 다릅니다. 자세한 내용은 백엔드 구성(Terraform 설명서)을 참조하십시오.</p> <p>3. 백엔드.tf 파일을 저장하고 닫습니다.</p>	
<p>Terraform 공급자 구성 파일을 편집합니다.</p>	<ol style="list-style-type: none"> 복제된 리포지토리에서 provider.tf 파일을 엽니다. 파일을 편집하여 Terraform 공급자 구성을 설정합니다. 자세한 내용은 구성 공급자(Terraform 설명서)를 참조하십시오. AWS 리전을 AWS Control Tower API를 사용할 수 있는 리전으로 설정합니다. provider.tf 파일을 저장하고 닫습니다. 	<p>DevOps 엔지니어, 테라폼</p>

작업	설명	필요한 기술
구성 파일을 편집합니다.	<ol style="list-style-type: none"> 1. 복제된 리포지토리에서 <code>variables.tfvars</code> 파일을 엽니다. 2. <code>controls</code> 섹션의 <code>control_names</code> 파라미터에 제어 API 식별자를 입력합니다. 각 컨트롤에는 AWS Control Tower를 사용할 수 있는 각 리전에 대한 고유한 API 식별자가 있습니다. 컨트롤 식별자를 찾으려면 다음을 수행합니다. <ol style="list-style-type: none"> a. 제어 메타데이터의 테이블에서 활성화하려는 제어를 찾습니다. b. 제어 API 식별자의 리전별 열에서 API를 직접 호출하려는 지역의 API 식별자(예 <code>arn:aws:controltower:us-east-1::control/AWS-GR_AUDIT_BUCKET_ENCRYPTION_ENABLED</code>)를 찾습니다. c. 리전의 식별자에서 제어 식별자를 추출합니다(예 <code>AWS-GR_AUDIT_BUCKET_ENCRYPTION_ENABLED</code>). 3. <code>controls</code> 섹션의 <code>organizational_uni</code> 	DevOps 엔지니어, 일반 AWS, 테라폼

작업	설명	필요한 기술
	<p>t_ids 파라미터에서 제어 (예: ou-1111-11111111) 를 활성화하려는 조직 단위의 ID를 입력합니다. ID를 큰 따옴표로 입력하고 여러 ID 는 쉼표로 구분합니다. OU ID를 검색하는 방법에 대한 자세한 내용은 OU 세부 정보 보기를 참조하십시오.</p> <p>4. variables.tfvars 파일을 저장하고 닫습니다. 업데이트된 variables.tfvars 파일의 예는 이 패턴의 추가 정보 섹션을 참조하십시오.</p>	
<p>관리 계정에서 IAM 역할을 수 있습니다.</p>	<p>관리 계정에서 Terraform 구성 파일을 배포할 권한이 있는 IAM 역할을 맡으십시오. 필요한 권한 및 샘플 정책에 대한 자세한 내용은 추가 정보 섹션의 IAM 역할에 대한 최소 권한 권한을 참조하십시오. AWS CLI 에서 IAM 역할을 맡는 방법에 대한 자세한 내용은 AWS CLI 에서의 IAM 역할 사용을 참조하십시오.</p>	<p>DevOps 엔지니어, 일반 AWS</p>

작업	설명	필요한 기술
구성 파일을 배포합니다.	<ol style="list-style-type: none"> 다음 명령을 입력하여 Terraform을 초기화합니다. <pre>\$ terraform init - upgrade</pre> 다음 명령을 입력하여 현재 상태와 비교하여 변경 내용을 미리 확인합니다. <pre>\$ terraform plan - var-file="variables.tfvars"</pre> Terraform 계획의 구성 변경 사항을 검토하고 조직에서 이러한 변경 사항을 구현할지 확인합니다. 다음 명령을 입력하여 리소스를 배포합니다. <pre>\$ terraform apply - var-file="variables.tfvars"</pre> 	DevOps 엔지니어, 일반 AWS, 테라폼

(선택 사항) AWS Control Tower 관리 계정에서 컨트롤을 비활성화합니다.

작업	설명	필요한 기술
파괴 명령을 실행합니다.	다음 명령을 입력하여 이 패턴으로 배포된 리소스를 제거합니다.	DevOps 엔지니어, 일반 AWS, 테라폼

작업	설명	필요한 기술
	<pre>\$ terraform destroy -var-file="variables.tfvars"</pre>	

문제 해결

문제	Solution
<p>Error: creating ControlTower Control ValidationException: Guardrail <control ID> is already enabled on organizational unit <OU ID> 오류</p>	<p>활성화하려는 제어는 대상 OU에서 이미 활성화되어 있습니다. 이 오류는 사용자가 AWS Management Console, AWS Control Tower 또는 AWS Organizations를 통해 수동으로 제어를 활성화한 경우 발생할 수 있습니다. Terraform 구성 파일을 배포하려면 다음 옵션 중 하나를 사용할 수 있습니다.</p> <p>옵션 1: Terraform 현재 상태 파일 업데이트</p> <p>리소스를 Terraform 현재 상태 파일로 가져올 수 있습니다. apply 명령을 다시 실행하면 Terraform은 이 리소스를 건너뛰게 됩니다. 리소스를 현재 Terraform 상태로 가져오려면 다음을 수행하십시오.</p> <ol style="list-style-type: none"> 1. AWS Control Tower 관리 계정에서 다음 명령을 입력하여 OU의 Amazon 리소스 이름 (ARN) 목록을 검색합니다. 여기서 <root-ID> 는 조직 루트입니다. 이 ID를 검색하는 방법에 대한 자세한 내용은 루트 세부 정보 보기를 참조하십시오. <pre>aws organizations list-organizational-units-for-parent --parent-id <root-ID></pre>

문제	Solution
	<p>2. 이전 단계에서 반환된 각 OU에 대해 다음 명령을 입력합니다. 여기서 <OU-ARN>은 OU의 ARN입니다.</p> <div style="border: 1px solid #ccc; border-radius: 10px; padding: 10px; margin: 10px 0;"> <pre>aws controltower list-enabled-controls --target-identifier <OU-ARN></pre> </div> <p>3. ARN을 복사하고 필수 모듈에서 Terraform 가져오기를 수행하여 Terraform 상태에 포함되도록 합니다. 자세한 지침은 가져오기(Terraform 설명서)를 참조하십시오.</p> <p>4. 에픽 섹션에서 구성 배포 단계를 반복하십시오.</p> <p>옵션 2: 컨트롤 비활성화</p> <p>비프로덕션 환경에서 작업하는 경우 콘솔에서 컨트롤을 비활성화할 수 있습니다. 에픽 섹션의 구성 배포 단계를 반복하여 다시 활성화하십시오. 일정 기간 동안 컨트롤이 비활성화되므로 프로덕션 환경에서는 이 방법을 사용하지 않는 것이 좋습니다. 프로덕션 환경에서 이 옵션을 사용하려는 경우 AWS Organizations에서 SCP를 일시적으로 적용하는 것과 같은 임시 제어를 구현할 수 있습니다.</p>

관련 리소스

AWS 설명서

- [제어에 대한 정보](#)(Control Tower 설명서)
- [제어 라이브러리](#)(Control Tower 설명서)
- [AWS CDK 및 AWS CloudFormation \(AWS 규범적 지침\) 를 사용하여 AWS 컨트롤 타워 컨트롤을 배포하고 관리합니다.](#)

기타 리소스

- [Terraform](#)
- [Terraform CLI 설명서](#)

추가 정보

variables.tfvars 파일

다음은 업데이트된 variables.tfvars 파일의 예입니다.

```
controls = [
  {
    control_names = [
      "AWS-GR_ENCRYPTED_VOLUMES",
      ...
    ],
    organizational_unit_ids = ["ou-1111-11111111", "ou-2222-22222222"...],
  },
  {
    control_names = [
      "AWS-GR_SUBNET_AUTO_ASSIGN_PUBLIC_IP_DISABLED",
      ...
    ],
    organizational_unit_ids = ["ou-1111-11111111"...],
  },
]
```

IAM 역할에 대한 최소 권한

이 APG 패턴을 사용하려면 관리 계정에서 IAM 역할을 수임해야 합니다. 가장 좋은 방법은 임시 권한이 있는 역할을 수임하고 최소 권한의 원칙에 따라 권한을 제한하는 것입니다. 다음 샘플 정책은 AWS Control Tower 제어를 활성화 또는 비활성화하는 데 필요한 최소 조치를 허용합니다.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "controltower:EnableControl",

```

```
        "controltower:DisableControl",
        "controltower:GetControlOperation",
        "controltower:ListEnabledControls",
        "organizations:AttachPolicy",
        "organizations:CreatePolicy",
        "organizations>DeletePolicy",
        "organizations:DescribeOrganization",
        "organizations:DetachPolicy",
        "organizations:ListAccounts",
        "organizations:ListAWSServiceAccessForOrganization",
        "organizations:ListChildren",
        "organizations:ListOrganizationalUnitsForParent",
        "organizations:ListParents",
        "organizations:ListPoliciesForTarget",
        "organizations:ListRoots",
        "organizations:UpdatePolicy"
    ],
    "Resource": "*"
}
]
}
```

여러 코드 결과물에서 보안 문제를 동시에 탐지하는 파이프라인을 배포하십시오.

작성자: 벤자민 모리스 (AWS), 디나 오돔 (AWS), 이사야 쉬슬러 (AWS), 사펙쉬 마단 (AWS), 팀 한 (AWS)

코드 리포지토리: [단순 코드 스캐닝 파이프라인](#)

환경: PoC 또는 파일럿

기술: 보안, ID, 규정 준수 DevOps

AWS 서비스: AWS CloudFormation, AWS CodeBuild, AWS CodeCommit, AWS CodePipeline

요약

[SCSP \(Simple Code Scanning Pipeline\)](#) 는 업계 표준 오픈 소스 보안 도구를 병렬로 실행하는 코드 분석 파이프라인을 두 번의 클릭으로 생성할 수 있는 기능을 제공합니다. 이를 통해 개발자는 도구를 설치하거나 실행 방법을 이해하지 않고도 코드의 품질과 보안을 확인할 수 있습니다. 이를 통해 코드 결과물의 취약성과 구성 오류를 줄일 수 있습니다. 또한 조직이 보안 도구를 설치, 조사 및 구성하는 데 소비하는 시간도 줄어듭니다.

SCSP 이전에는 개발자가 이 특정 도구 세트를 사용하여 코드를 스캔하려면 소프트웨어 분석 도구를 찾아 수동으로 설치하고 구성해야 했습니다. ASH (자동 보안 도우미) 와 같은 all-in-one 도구를 로컬에 설치하더라도 실행하려면 Docker 컨테이너를 구성해야 합니다. 하지만 SCSP를 사용하면 업계 표준 코드 분석 도구 제품군이 에서 자동으로 실행됩니다. AWS 클라우드 이 솔루션에서는 Git을 사용하여 코드 결과물을 푸시하면 실패한 보안 검사에 at-a-glance 대한 통찰력이 포함된 시각적 출력을 받게 됩니다.

사전 조건 및 제한 사항

- 액티브 AWS 계정
- 보안 문제를 확인하기 위해 스캔하려는 하나 이상의 코드 결과물
- AWS Command Line Interface ([AWS CLI](#)), [설치 및 구성](#)
- [Python 버전 3.0 이상 및 pip 버전 9.0.3 이상, 설치됨](#)

- Git, [설치됨](#)
- 로컬 [git-remote-codecommit](#) 워크스테이션에 설치

아키텍처

대상 기술 스택

- AWS CodeCommit 리포지토리
- AWS CodeBuild 프로젝트
- AWS CodePipeline 파이프라인
- Amazon Simple Storage Service(S3) 버킷
- AWS CloudFormation 템플릿

대상 아키텍처

정적 코드 분석을 위한 SCSP는 전달 가능한 코드에 대한 보안 피드백을 제공하도록 설계된 DevOps 프로젝트입니다.

1. 에서 AWS Management Console 대상에 AWS 계정 로그인합니다. 파이프라인을 배포하려는 AWS 리전 위치에 있는지 확인하세요.
2. 코드 리포지토리의 CloudFormation 템플릿을 사용하여 SCSP 스택을 배포합니다. 그러면 새 CodeCommit 리포지토리와 CodeBuild 프로젝트가 만들어집니다.

참고: 대체 배포 옵션으로, 스택 배포 중에 리포지토리의 Amazon 리소스 이름 (ARN) 을 파라미터로 CodeCommit 제공하여 기존 배포 옵션을 사용할 수 있습니다.

3. 리포지토리를 로컬 워크스테이션에 복제한 다음 복제된 리포지토리의 해당 폴더에 파일을 추가합니다.
4. Git을 사용하여 파일을 추가하고 커밋하고 리포지토리에 푸시합니다. CodeCommit
5. CodeCommit 리포지토리로 푸시하면 작업이 시작됩니다. CodeBuild CodeBuild 프로젝트는 보안 도구를 사용하여 코드 결과물을 스캔합니다.
6. 파이프라인 결과를 검토하세요. 오류 수준의 문제를 발견한 보안 도구를 사용하면 파이프라인에서 작업이 실패할 수 있습니다. 이러한 오류를 수정하거나 오탐으로 표시하지 않도록 하세요. 파이프라

인의 S3 버킷에 CodePipeline 있거나 파이프라인의 S3 버킷에 있는 작업 세부 정보에서 도구 출력의 세부 정보를 검토하세요.

도구

AWS 서비스

- [AWS CloudFormation](#) AWS 리소스를 설정하고, 리소스를 빠르고 일관되게 프로비저닝하고, 라이프 사이클 AWS 계정 전반과 지역에 걸쳐 리소스를 관리할 수 있도록 도와줍니다.
- [AWS CodeBuild](#) 소스 코드를 컴파일하고, 단위 테스트를 실행하고, 배포할 준비가 된 아티팩트를 생성하는 데 도움이 되는 완전 관리형 빌드 서비스입니다.
- [AWS CodeCommit](#) 자체 소스 제어 시스템을 관리할 필요 없이 Git 리포지토리를 비공개로 저장하고 관리할 수 있는 버전 제어 서비스입니다.

기타 도구

[SCSP가 코드 결과물을 스캔하는 데 사용하는 도구의 전체 목록은 의 SCSP readme를 참조하십시오.](#)

GitHub

코드 리포지토리

이 패턴의 코드는 의 [단순 코드 스캔 파이프라인 \(SCSP\)](#) 저장소에서 사용할 수 있습니다. GitHub

에픽

SCSP를 배포하십시오.

작업	설명	필요한 기술
CloudFormation 스택을 생성하세요.	<ol style="list-style-type: none"> 1. AWS Management Console에 로그인합니다. 2. 콘솔에서 솔루션을 배포하려는 대상 지역에 있는지 확인합니다. 자세한 내용은 지역 선택을 참조하십시오. 3. 다음 링크를 선택합니다. 그러면 빠른 스택 생성 마법 	AWS DevOps, AWS 관리자

작업	설명	필요한 기술
	<p>사가 열립니다 CloudFormation.</p> <p>https://console.aws.amazon.com/cloudformation/home?#/stacks/create/review?templateURL=https://proservetools.s3.amazonaws.com/cft/scsp-pipeline-stack.template.json&stackName=SimpleCodeScanPipeline</p> <ol style="list-style-type: none"> 4. 빠른 스택 생성 마법사에서 스택의 매개변수 설정을 검토하고 사용 사례에 맞게 필요에 따라 수정하십시오. 5. AWS에서 IAM 리소스를 생성할 CloudFormation 수 있음을 인정함을 선택한 다음 스택 생성을 선택합니다. <p>그러면 CodeCommit 리포지토리, CodePipeline 파이프라인, 여러 CodeBuild 작업 정의, S3 버킷이 생성됩니다. 빌드 실행 및 스캔 결과가 이 버킷에 복사됩니다. CloudFormation 스택이 완전히 배포되면 SCSP를 사용할 수 있습니다.</p>	

파이프라인 사용

작업	설명	필요한 기술
스캔 결과를 살펴보세요.	<ol style="list-style-type: none"> 1. Amazon S3 콘솔의 버킷에서 리소스 파이프라인-삭제 파이프라인/reso 버킷을 실행하는 간단한 코드를 선택합니다. 2. scan_results 디렉토리를 선택한 다음 가장 최근 스캔 날짜 스탬프가 있는 폴더를 선택합니다. 3. 이 폴더의 로그 파일을 검토하여 파이프라인에서 사용되는 보안 도구에서 탐지된 문제를 검토하십시오. 오류 수준의 문제를 발견한 보안 도구는 파이프라인에서 failed 조치를 취하게 됩니다. 오탐인 경우 이를 수정하거나 억제해야 합니다. <p>참고: CodePipeline 콘솔의 작업 세부 정보 섹션에서 도구 출력의 세부 정보 (통과 및 실패 스캔 모두) 를 볼 수도 있습니다.</p>	앱 개발자, AWS DevOps

문제 해결

문제	Solution
HashiCorp 테라폼 또는 AWS CloudFormation 파일은 스캔되지 않습니다.	Terraform (.tf) 및 CloudFormation (.yaml, .yml 또는 .json) 파일이 복제된 저장소의 적절한 폴더에 있는지 확인하십시오. CodeCommit
git clone 명령이 실패했습니다.	설치했는지, CLI가 git-remote-codecommit 리포지토리를 읽을 수 있는 권한이 있는 AWS 자격 증명에 액세스할 수 있는지 확인하십시오. CodeCommit
동시성 오류 (예:) Project-level concurrent build limit cannot exceed the account-level concurrent build limit of 1	콘솔에서 변경 릴리스 버튼을 선택하여 파이프라인을 다시 실행합니다. CodePipeline 이는 알려진 문제로, 파이프라인이 처음 몇 번 실행될 때 가장 흔하게 발생하는 것으로 보입니다.

관련 리소스

SCSP 프로젝트에 대한 [피드백을 제공하세요](#).

추가 정보

FAQ

SCSP 프로젝트가 ASH (자동 보안 도우미) 와 동일한가요?

아니요. 컨테이너를 사용하여 코드 스캔 도구를 실행하는 CLI 도구가 필요한 경우 ASH를 사용하십시오. [ASH \(자동 보안 도우미\)](#) 는 새 코드, 인프라 또는 IAM 리소스 구성에서 보안 위반이 발생할 가능성을 줄이기 위해 설계된 도구입니다. ASH는 로컬에서 실행할 수 있는 명령줄 유틸리티입니다. 로컬에서 사용하려면 시스템에 컨테이너 환경을 설치하고 운영해야 합니다.

ASH보다 쉬운 설정 파이프라인을 원할 때는 SCSP를 사용하세요. SCSP는 로컬 설치가 필요하지 않습니다. SCSP는 파이프라인에서 개별적으로 검사를 실행하고 도구별로 결과를 표시하도록 설계되었습니다. 또한 SCSP는 Docker를 설정하는 데 따른 오버헤드를 크게 피할 수 있으며 운영 체제 (OS) 에 구애받지 않습니다.

SCSP는 보안 팀에만 적합한가요?

아니요. 누구나 파이프라인을 배포하여 코드의 어느 부분이 보안 검사에 실패했는지 확인할 수 있습니다. 예를 들어, 보안이 아닌 사용자는 보안 팀과 검토하기 전에 SCSP를 사용하여 코드를 확인할 수 있습니다.

„ 또는 Bitbucket과 같은 GitLab 다른 유형의 저장소를 사용하는 경우에도 SCSP를 사용할 수 있습니까? GitHub

로컬 git 리포지토리가 서로 다른 두 개의 원격 리포지토리를 가리키도록 구성할 수 있습니다. 예를 들어 기존 GitLab 리포지토리를 복제하고 SCSP 인스턴스 (필요한 경우 Terraform 및 AWS Config AWS RDK) 폴더 지정 CloudFormation) 를 만든 다음 로컬 리포지토리가 SCSP 리포지토리를 git remote add upstream <SCSPGitLink> 가리키도록 하는 데 사용할 수도 있습니다. CodeCommit 이렇게 하면 코드 변경 내용을 먼저 SCSP로 전송하여 유효성을 검사한 다음, 결과를 해결하기 위한 추가 업데이트가 이루어진 후, 또는 Bitbucket 리포지토리로 푸시할 수 있습니다. GitLab GitHub 다중 리모트에 대한 자세한 내용은 [추가 Git 리포지토리로 커밋 푸시 \(AWS 블로그 게시물\)](#) 를 참조하십시오.

참고: 웹 인터페이스를 통해 변경하지 않는 등 드리프트에 주의하세요.

기여 및 자체 작업 추가

SCSP 설정은 SCSP 응용 프로그램의 소스 코드를 포함하는 GitHub 프로젝트로 유지 관리됩니다. AWS Cloud Development Kit (AWS CDK) 파이프라인에 추가 검사를 추가하려면 AWS CDK 응용 프로그램을 AWS 계정 업데이트한 다음 파이프라인이 실행될 대상으로 합성하거나 배포해야 합니다. 이렇게 하려면 먼저 SCSP [GitHub 프로젝트를](#) 복제한 다음 폴더에서 스택 정의 파일을 찾으십시오. lib

추가하려는 추가 검사가 있는 경우 AWS CDK 코드의 StandardizedCodeBuildProject 클래스를 사용하면 매우 간단하게 작업을 추가할 수 있습니다. 이름, 설명 및/또는 명령을 입력합니다. install build AWS CDK 적절한 기본값을 사용하여 CodeBuild 프로젝트를 만듭니다. 빌드 프로젝트를 만드는 것 외에도 빌드 단계의 CodePipeline 작업에도 프로젝트를 추가해야 합니다. 새 검사를 디자인할 때 검사 도구가 문제를 발견하거나 실행에 실패하면 조치를 취해야 FAIL 합니다. 스캔 도구에서 문제가 발견되지 않으면 조치를 취해야 PASS 합니다. 도구 구성의 예를 보려면 Bandit 작업 코드를 검토하세요.

예상 입력 및 출력에 대한 자세한 내용은 [리포지토리 설명서를](#) 참조하십시오.

사용자 지정 작업을 추가하는 경우 cdk deploy 또는 cdk synth + CloudFormation deploy 를 사용하여 SCSP를 배포해야 합니다. 빠른 생성 스택 CloudFormation 템플릿은 리포지토리 소유자가 유지 관리하기 때문입니다.

Terraform을 사용하여 AWS WAF의 보안 자동화 솔루션 배포

작성자: Dr. Rahul Sharad Gaikwad(AWS) 및 Tamilselvan P(AWS)

코드 리포지토리: - 샘플 aws-waf-automation-terraform	환경: PoC 또는 파일럿	기술: 보안, ID, 규정 준수, 인프라, 콘텐츠 제공, DevOps
워크로드: 기타 모든 워크로드	AWS 서비스: AWS WAF	

요약

AWS WAF는 웹 액세스 제어 목록(ACL)에서 정의하고 배포하는 사용자 지정 가능한 규칙을 사용하여 일반적인 공격으로부터 애플리케이션을 보호하는 데 도움이 되는 웹 애플리케이션 방화벽입니다. AWS WAF 규칙을 구성하는 것은 어려울 수 있으며, 특히 전담 보안 팀이 없는 조직의 경우 더욱 그렇습니다. 이 프로세스를 단순화하기 위해 Amazon Web Services(AWS)는 [AWS WAF의 보안 자동화](#) 솔루션을 제공합니다. 이 솔루션은 웹 기반 공격을 필터링하는 일련의 AWS WAF 규칙을 사용하여 단일 웹 ACL을 자동으로 배포합니다. Terraform 배포 중에 포함할 보호 기능을 지정할 수 있습니다. 이 솔루션을 배포한 후 AWS WAF는 기존 Amazon 배포 또는 애플리케이션 로드 밸런서에 CloudFront 대한 웹 요청을 검사하고 규칙과 일치하지 않는 모든 요청을 차단합니다.

AWS WAF용 보안 자동화 솔루션은 AWS WAF용 [보안 자동화](#) 구현 가이드의 지침에 따라 CloudFormation AWS를 사용하여 배포할 수 있습니다. 이 패턴은 HashiCorp Terraform을 선호하는 IaC(코드형 인프라) 도구로 사용하여 클라우드 인프라를 프로비저닝하고 관리하는 조직에 대체 배포 옵션을 제공합니다. 이 솔루션을 배포할 때 Terraform은 클라우드에 변경 사항을 자동으로 적용하고 AWS WAF 설정 및 보호 기능을 배포하고 구성합니다.

사전 조건 및 제한 사항

사전 조건

- 활성 상태의 AWS 계정.
- AWS Command Line Interface(AWS CLI)가 설치되고 필요한 권한으로 구성되었습니다. 자세한 내용은 [시작하기](#)(AWS CLI 설명서)를 참조하세요.
- Terraform이 설치 및 구성되어 있습니다. 자세한 내용은 [Terraform 설치](#)(Terraform 설명서)를 참조하세요.

제품 버전

- AWS CLI 버전 2.4.25 이상
- Terraform 버전 1.1.9 이상

아키텍처

대상 아키텍처

이 패턴은 AWS WAF의 보안 자동화 솔루션을 배포합니다. 대상 아키텍처에 대한 자세한 내용은 AWS WAF의 보안 자동화 구현 가이드의 [아키텍처 개요](#)를 참조하세요. 이 배포에서의 AWS Lambda 자동화, 애플리케이션 로그 파서, AWS WAF 로그 파서, IP 목록 파서, 액세스 핸들러에 대한 자세한 내용은 AWS WAF의 보안 자동화 구현 가이드의 [구성 요소 세부 정보](#)를 참조하세요.

Terraform 배포

terraform apply를 실행하면 Terraform이 다음과 같은 작업을 수행합니다.

1. Terraform은 testing.tfvars 파일의 입력을 기반으로 IAM 역할 및 Lambda 함수를 생성합니다.
2. Terraform은 testing.tfvars 파일의 입력을 기반으로 AWS WAF ACL 규칙 및 IP 세트를 생성합니다.
3. Terraform은 testing.tfvars 파일의 입력을 기반으로 아마존 심플 스토리지 서비스 (Amazon S3) 버킷 EventBridge , 아마존 규칙, AWS Glue 데이터베이스 테이블, 아마존 아테나 작업 그룹을 생성합니다.
4. Terraform은 AWS CloudFormation 스택을 배포하여 사용자 지정 리소스를 프로비저닝합니다.
5. Terraform은 testing.tfvars 파일의 지정된 입력을 기반으로 Amazon API Gateway 리소스를 생성합니다.

자동화 및 규모 조정

이 패턴을 사용하여 여러 AWS 계정 및 AWS 리전에 대한 AWS WAF 규칙을 생성하여 AWS 클라우드 환경 전체에 AWS WAF의 보안 자동화 솔루션을 배포할 수 있습니다.

도구

서비스

- [AWS Command Line Interface\(AWS CLI\)](#)는 명령줄 쉘에서 명령을 사용하여 AWS 서비스와 상호 작용할 수 있는 오픈 소스 도구입니다.

- [AWS WAF](#)는 보호되는 웹 애플리케이션 리소스로 전달되는 HTTP 및 HTTPS 요청을 모니터링할 수 있게 해주는 웹 애플리케이션 방화벽입니다.

기타 서비스

- [Git](#)은 오픈 소스 분산 버전 제어 시스템입니다.
- [HashiCorp Terraform](#)은 코드를 사용하여 클라우드 인프라 및 리소스를 프로비저닝하고 관리하는 데 도움이 되는 명령줄 인터페이스 애플리케이션입니다.

코드 리포지토리

이 패턴의 코드는 [Terraform을 사용한 GitHub AWS WAF 자동화](#) 리포지토리에서 사용할 수 있습니다.

모범 사례

- 정적 파일을 별도의 S3 버킷에 넣습니다.
- 변수를 하드 코딩하지 않습니다.
- 사용자 지정 스크립트의 사용을 제한합니다.
- 명명 규칙을 채택합니다.

에픽

로컬 워크스테이션 설정

작업	설명	필요한 기술
Git을 설치합니다.	시작하기 (Git 웹 사이트)의 지침에 따라 로컬 워크스테이션에 Git을 설치합니다.	DevOps 엔지니어
리포지토리를 복제합니다.	로컬 워크스테이션에서 코드 리포지토리를 복제하려면 다음 명령을 입력합니다. 리포지토리 URL을 포함한 전체 명령을 복사하려면 이 패턴의 추가 정보 섹션을 참조하세요.	DevOps 엔지니어

작업	설명	필요한 기술
	<pre>git clone <repo-URL> .git</pre>	
변수를 업데이트합니다.	<ol style="list-style-type: none"> 다음 명령을 입력하여 복제된 디렉터리로 이동합니다. <pre>cd terraform-aws-waf-automation</pre> 아무 텍스트 편집기에서 <code>testing.tfvars</code> 파일을 엽니다. <code>testing.tfvars</code> 파일의 변수 값을 업데이트합니다. 파일을 저장하고 닫습니다. 	DevOps 엔지니어

Terraform을 사용하여 대상 아키텍처 프로비저닝

작업	설명	필요한 기술
Terraform 구성을 초기화합니다.	<p>다음 명령을 입력하여 Terraform 구성 파일이 포함된 작업 디렉토리를 초기화합니다.</p> <pre>terraform init</pre>	DevOps 엔지니어
Terraform 계획을 미리 보기합니다.	<p>다음 명령을 입력합니다. Terraform은 구성 파일을 평가하여 선언된 리소스의 목표 상태를 결정합니다. 그런 다음 목표 상태를 현재 상태와 비교하고 계획을 구성합니다.</p>	DevOps 엔지니어

작업	설명	필요한 기술
	<pre>terraform plan -var-file ="testing.tfvars"</pre>	
계획을 확인합니다.	계획을 검토하고 대상 AWS 계정에 필요한 아키텍처가 구성되어 있는지 확인합니다.	DevOps 엔지니어
솔루션을 배포합니다.	<ol style="list-style-type: none"> 다음 명령을 입력하여 계획을 적용합니다. <pre>terraform apply - var-file="testing .tfvars"</pre> yes를 입력하여 확인합니다. Terraform은 구성 파일에 선언된 목표 상태를 달성하기 위해 인프라를 생성, 업데이트 또는 파괴합니다. 순서에 대한 자세한 내용은 이 패턴의 아키텍처 섹션에서 Terraform 배포를 참조하세요. 	DevOps 엔지니어

확인 및 정리

작업	설명	필요한 기술
변경 사항을 확인합니다.	<ol style="list-style-type: none"> Terraform 콘솔에서 출력이 예상 결과와 일치하는지 확인합니다. AWS Management Console에 로그인합니다. 	DevOps 엔지니어

작업	설명	필요한 기술
	3. Terraform 콘솔의 출력이 AWS 계정에 성공적으로 배포되었는지 확인합니다.	
(선택 사항) 인프라를 정리합니다.	이 솔루션으로 수행한 모든 리소스 및 구성의 변경 내용을 제거하려면 다음을 수행합니다. 1. Terraform 콘솔에서 다음 명령을 입력합니다. <div style="border: 1px solid #ccc; border-radius: 10px; padding: 10px; margin: 10px 0;"> <pre>terraform destroy - var-file="testing .tfvars"</pre> </div> 2. yes를 입력하여 확인합니다.	DevOps 엔지니어

문제 해결

문제	Solution
WAFV2 IPSet: WAFOptimisticLockException 오류	terraform destroy 명령을 실행할 때 이 오류가 발생하면 IP 세트를 수동으로 삭제해야 합니다. 지침은 IP 세트 삭제 (AWS WAF 설명서)를 참조하세요.

관련 리소스

AWS 참조

- [AWS WAF의 보안 자동화 구현 가이드](#)
- [AWS WAF의 보안 자동화](#)(AWS Solutions Library)
- [AWS WAF의 보안 자동화 FAQ](#)

Terraform 참조

- [Terraform 백엔드 구성](#)
- [Terraform AWS 공급자-설명서 및 사용](#)
- [테라폼 AWS 공급자](#) (GitHub 리포지토리)

추가 정보

다음 명령은 이 패턴의 GitHub 리포지토리를 복제합니다.

```
git clone https://github.com/aws-samples/aws-waf-automation-terraform-samples.git
```

Step Functions를 사용하여 IAM Access Analyzer로 IAM 정책을 동적으로 생성하기

작성: Thomas Scott (AWS), Adil El Kanabi (AWS), Koen van Blijderveen (AWS), and Rafal Pawlaszek (AWS)

코드 리포지토리: [자동화된 IAM 액세스 분석기 역할 정책 생성기](#)

환경: PoC 또는 파일럿

기술: 보안, 자격 증명, 규정 준수, 서버리스

AWS 서비스: AWS IAM Access Analyzer, AWS Lambda, AWS Step Functions, AWS Identity and Access Management

요약

최소 권한은 작업을 수행하는 데 필요한 최소 권한을 부여하는 보안 모범 사례입니다. 이미 활성화된 Amazon Web 서비스(AWS) 계정에서 최소 권한 액세스를 구현하는 것은 어려울 수 있습니다. 권한을 변경하여 의도치 않게 사용자가 직무 수행을 하지 못하도록 차단하는 것을 원하지 않기 때문입니다. AWS IAM(Identity and Access Management) 정책 변경 사항을 구현하기 전에 계정 사용자가 수행하는 작업 및 리소스를 이해해야 합니다.

이 패턴은 팀 생산성을 막거나 저하시키지 않으면서 최소 권한 액세스 원칙을 적용할 수 있도록 설계되었습니다. IAM 액세스 분석기 및 AWS Step Functions를 사용하여 계정에서 현재 수행 중인 작업을 기반으로 역할에 대한 up-to-date IAM 정책을 동적으로 생성하는 방법을 설명합니다. 새 정책은 현재 활동을 허용하되 불필요한 권한이나 상승된 권한을 제거하도록 설계되었습니다. 허용 및 거부 규칙을 정의하여 생성된 정책을 사용자 지정할 수 있으며, 이 솔루션은 사용자 지정 규칙을 통합합니다.

이 패턴에는 AWS Cloud Development Kit (AWS CDK) 또는 테라폼용 CDK (HashiCorp CDKTF) 를 사용하여 솔루션을 구현하는 옵션이 포함되어 있습니다. 그런 다음 지속적 통합 및 지속적 전달(CI/CD) 파이프라인을 사용하여 새 정책을 역할에 연결할 수 있습니다. 다중 계정 아키텍처를 사용하는 경우 역할에 대해 업데이트된 IAM 정책을 생성하려는 모든 계정에 이 솔루션을 배포하여 전체 AWS 클라우드 환경의 보안을 강화할 수 있습니다.

사전 조건 및 제한 사항

사전 조건

- CloudTrail 트레일이 활성화된 활성 AWS 계정
- 다음 항목에 대한 IAM 권한:
 - Step Functions 워크플로우를 생성하고 배포합니다. 자세한 내용은 [AWS Step Functions의 작업, 리소스 및 조건 키](#)(Step Functions 설명서)를 참고하십시오.
 - AWS Lambda 함수 생성 자세한 내용은 [실행 역할 및 사용자 권한](#)(Lambda 설명서)을 참조하세요.
 - IAM 역할을 생성합니다. 자세한 내용은 [IAM 사용자에게 권한을 위임하기 위한 역할 생성](#)(IAM 설명서)을 참조하십시오.
- npm이 설치되었습니다. 자세한 내용은 [Node.js 및 npm 다운로드 및 설치](#)(npm 설명서)를 참고하십시오.
- AWS CDK(옵션 1)를 사용하여 이 솔루션을 배포하는 경우:
 - AWS CDK Toolkit, 설치 및 구성됨. 자세한 내용은 [AWS CDK 설치](#)(AWS CDK 설명서)를 참조하세요.
- CDKTF(옵션 2)를 사용하여 이 솔루션을 배포하는 경우:
 - CDKTF, 설치 및 구성됨. 자세한 내용은 [Terraform용 CDK 설치](#)(CDKTF 설명서)를 참조하세요.
 - Terraform, 설치 및 구성됨 자세한 내용은 [시작하기](#)(Terraform 설명서)를 참조하세요.
- AWS 계정에 대해 로컬 위치에 설치 및 구성된 AWS Command Line Interface(AWS CLI). 자세한 내용은 [최신 버전의 AWS CLI 설치 또는 업데이트](#)(AWS CLI 설명서)를 참조하세요.

제한 사항

- 이 패턴은 새 IAM 정책을 역할에 적용하지 않습니다. 이 솔루션이 끝나면 새 IAM 정책이 CodeCommit 리포지토리에 저장됩니다. CI/CD 파이프라인을 사용하여 계정 내 역할에 정책을 적용할 수 있습니다.

아키텍처

대상 아키텍처

1. 정기적으로 예약된 Amazon EventBridge 이벤트 규칙에 따라 Step Functions 워크플로가 시작됩니다. 이 솔루션을 설정하는 과정에서 이 재생성 예약을 정의합니다.
2. Step Functions 워크플로에서 Lambda 함수는 로그의 계정 활동을 분석할 때 사용할 날짜 범위를 생성합니다. CloudTrail
3. 다음 워크플로우 단계에서는 IAM Access Analyzer API를 호출하여 정책 생성을 시작합니다.
4. IAM Access Analyzer는 설정 시 지정한 역할의 Amazon 리소스 이름 (ARN) 을 사용하여 지정된 날짜 내의 활동이 있는지 CloudTrail 로그를 분석합니다. IAM Access Analyzer는 활동을 기반으로 지정된 날짜 범위 동안 역할이 사용하는 작업 및 서비스만 허용하는 IAM 정책을 생성합니다. 이 단계가 완료되면 이 단계에서 작업 ID가 생성됩니다.
5. 다음 워크플로우 단계에서는 30초마다 작업 ID를 확인합니다. 작업 ID가 감지되면 이 단계에서는 작업 ID를 사용하여 IAM Access Analyzer API를 호출하고 새 IAM 정책을 검색합니다. IAM Access Analyzer는 정책을 JSON 파일로 반환합니다.
6. 다음 워크플로우 단계에서는 <IAM role name>/policy.json 파일을 Amazon Simple Storage Service(Amazon S3) 버킷에 넣습니다. 이 솔루션을 설정하는 과정에서 이 S3 버킷을 정의합니다.
7. Amazon S3 이벤트 알림이 Lambda 함수를 시작합니다.
8. Lambda 함수는 S3 버킷에서 정책을 검색하고 allow.json 및 deny.json 파일에 정의한 사용자 지정 규칙을 통합한 다음 업데이트된 정책을 푸시합니다. CodeCommit 이 솔루션 설정의 일환으로 CodeCommit 리포지토리, 브랜치 및 폴더 경로를 정의합니다.

도구

서비스

- [AWS Cloud Development Kit\(AWS CDK\)](#) 는 AWS 클라우드 인프라를 코드로 정의하고 프로비저닝하는 데 도움이 되는 소프트웨어 개발 프레임워크입니다.
- [AWS CDK Toolkit](#)은 AWS Cloud Development Kit(AWS CDK) 앱과 상호 작용하는 데 도움이 되는 명령줄 클라우드 개발 키트입니다.
- [AWS는 AWS](#) 계정의 거버넌스, 규정 준수 및 운영 위험을 감사할 수 있도록 CloudTrail 지원합니다.
- [CodeCommitAWS](#)는 자체 소스 제어 시스템을 관리할 필요 없이 Git 리포지토리를 비공개로 저장하고 관리할 수 있는 버전 제어 서비스입니다.
- [AWS Command Line Interface\(AWS CLI\)](#)는 명령줄 셸에서 명령을 사용하여 AWS 서비스와 상호 작용할 수 있는 오픈 소스 도구입니다.
- [AWS Identity and Access Management\(IAM\)](#)를 사용하면 사용자에 대해 인증 및 권한 부여를 제어함으로써 AWS 리소스에 대한 액세스를 안전하게 관리할 수 있습니다. 이 패턴은 [IAM의 기능인 IAM](#)

[Access Analyzer](#)를 사용하여 CloudTrail 로그를 분석하여 IAM 개체 (사용자 또는 역할) 가 사용한 작업 및 서비스를 식별한 다음 해당 활동을 기반으로 IAM 정책을 생성합니다.

- [AWS Lambda](#)는 서버를 프로비저닝하거나 관리할 필요 없이 코드를 실행하는 데 도움이 되는 컴퓨팅 서비스입니다. 필요할 때만 코드를 실행하며 자동으로 확장이 가능하므로 사용한 컴퓨팅 시간만큼만 비용을 지불합니다.
- [Amazon Simple Storage Service\(Amazon S3\)](#)는 원하는 양의 데이터를 저장, 보호 및 검색하는 데 도움이 되는 클라우드 기반 객체 스토리지 서비스입니다.
- [AWS Step Functions](#)는 AWS Lambda 함수와 기타 AWS 서비스를 결합할 수 있는 서버리스 오케스트레이션 서비스로, 비즈니스 크리티컬 애플리케이션을 구축합니다. 이 패턴에서는 Step Functions의 [AWS SDK 서비스 통합](#)을 사용하여 워크플로우에서 서비스 API 작업을 호출합니다.

기타 도구

- [Terraform\(CDKTF\)용 CDK](#)를 사용하면 Python 및 Typescript와 같은 일반적인 프로그래밍 언어를 사용하여 코드형 인프라(IaC)로 정의할 수 있습니다.
- [Lerna](#)는 동일한 저장소에서 여러 패키지 JavaScript 또는 TypeScript 패키지를 관리하고 게시하기 위한 빌드 시스템입니다.
- [Node.js](#)는 확장 가능한 네트워크 애플리케이션을 구축하기 위해 설계된 이벤트 기반 JavaScript 런타임 환경입니다.
- [npm](#)은 Node.js 환경에서 실행되는 소프트웨어 레지스트리로, 패키지를 공유 또는 대여하고 개인 패키지의 배포를 관리하는 데 사용됩니다.

코드 리포지토리

이 패턴의 코드는 GitHub [자동화된 IAM 액세스 분석기 역할](#) 정책 생성기 리포지토리에서 사용할 수 있습니다.

에픽

배포 준비

작업	설명	필요한 기술
리포지토리를 복제합니다.	다음 명령은 자동화된 IAM 액세스 분석기 역할 정책 생성기	앱 개발자

작업	설명	필요한 기술
	<p>() 리포지토리를 복제합니다.</p> <p>GitHub</p> <pre>git clone https://github.com/aws-samples/automated-iam-access-analyzer.git</pre>	
Lerna를 설치합니다.	<p>다음 명령은 Lerna를 설치합니다.</p> <pre>npm i -g lerna</pre>	앱 개발자
종속성을 설정합니다.	<p>다음 명령은 리포지토리의 종속성을 설치합니다.</p> <pre>cd automated-iam-access-advisor/ npm install && npm run bootstrap</pre>	앱 개발자
코드를 빌드합니다.	<p>다음 명령은 Lambda 함수의 zip 패키지를 테스트, 구축 및 준비합니다.</p> <pre>npm run test:code npm run build:code npm run pack:code</pre>	앱 개발자
생성자를 구축합니다.	<p>다음 명령은 AWS CDK와 CDKTF 모두에 대해 애플리케이션을 합성하는 인프라를 구축합니다.</p> <pre>npm run build:infra</pre>	

작업	설명	필요한 기술
모든 사용자 지정 권한을 구성합니다.	복제된 리포지토리의 복제 폴더에서 allow.json 및 deny.json 파일을 편집하여 역할에 대한 사용자 지정 권한을 정의합니다. allow.json 및 deny.json 파일에 동일한 권한이 포함된 경우 거부 권한이 적용됩니다.	AWS 관리자, 앱 개발자

옵션 1 — AWS CDK를 사용하여 솔루션 배포

작업	설명	필요한 기술
AWS CDK 스택을 배포합니다.	<p>다음 명령은 CloudFormation AWS를 통해 인프라를 배포합니다. 다음 파라미터를 정의합니다.</p> <ul style="list-style-type: none"> • <NAME_OF_ROLE> - 새 정책을 생성하는 데 사용할 IAM 역할의 ARN입니다. • <TRAIL_ARN> - 역할 활동이 저장되는 CloudTrail 트레일의 ARN입니다. • <CRON_EXPRESSION_TO_RUN_SOLUTION> - 정책의 재생성 예약을 정의하는 Cron 표현식입니다. Step Functions 워크플로우는 이 일정에 따라 실행됩니다. • <TRAIL_LOOKBACK> - 역할 권한을 평가할 때 추적에서 되돌아볼 기간(일) 	앱 개발자

작업	설명	필요한 기술
	<pre>cd infra/cdk cdk deploy --parameters roleArn=<NAME_OF_ROLE> \ --parameters trailArn= <TRAIL_ARN> \ --parameters schedule= <CRON_EXPRESSION_T O_RUN_SOLUTION> \ [--parameters trailLookBack=<TRAIL_LOOKBACK>]</pre> <p>참고 - 대괄호는 선택적 매개변수를 나타냅니다.</p>	
(선택 사항) 새 정책이 적용될 때까지 기다리십시오.	추적에 역할에 대한 합당한 양의 과거 활동이 포함되어 있지 않은 경우, IAM Access Analyzer에서 정확한 정책을 생성하기에 충분한 기록 활동이 있다고 확신할 수 있을 때까지 기다리십시오. 해당 계정에서 충분한 기간 동안 역할이 활성화된 경우에는 이 대기 시간이 필요하지 않을 수 있습니다.	AWS 관리자
생성된 정책 내용을 수동으로 검토합니다.	CodeCommit 리포지토리에서 생성된.json <ROLE_ARN>파일을 검토하여 허용 및 거부 권한이 역할에 적합한지 확인하십시오.	AWS 관리자

옵션 2 — CDKTF를 사용하여 솔루션 배포

작업	설명	필요한 기술
<p>Terraform 템플릿을 합성합니다.</p>	<p>다음 명령은 Terraform 템플릿을 합성합니다.</p> <pre>lerna exec cdktf synth --scope @aiaa/tfm</pre>	<p>앱 개발자</p>
<p>Terraform 템플릿을 배포합니다.</p>	<p>다음 명령은 CDKTF-정의형 인프라가 포함된 디렉토리로 이동합니다.</p> <pre>cd infra/cdktf</pre> <p>다음 명령은 대상 AWS 계정에 인프라를 배포합니다. 다음 파라미터를 정의합니다.</p> <ul style="list-style-type: none"> • <account_ID> – 대상 계정의 ID입니다. • <region> – 대상 AWS 리전입니다. • <selected_role_ARN> – 새 정책을 생성하는 데 사용할 IAM 역할의 ARN입니다. • <trail_ARN> — 역할 활동이 저장되는 CloudTrail 트레일의 ARN입니다. • <schedule_expression> – 정책의 재생성 예약을 정의하는 Cron 표현식입니다. Step Functions 워크플 	<p>앱 개발자</p>

작업	설명	필요한 기술
	<p>로우는 이 일정에 따라 실행됩니다.</p> <ul style="list-style-type: none"> <trail_look_back> - 역할 권한을 평가할 때 추적에서 되돌아볼 기간(일) <pre>TF_VAR_accountId=<account_ID> \ TF_VAR_region=<region> \ TF_VAR_roleArns=<selected_role_ARN> \ TF_VAR_trailArn=<trail_ARN> \ TF_VAR_schedule=<schedule_expression> \ [TF_VAR_trailLookBack=<trail_look_back>] \ cdktf deploy</pre> <p>참고 - 대괄호는 선택적 매개변수를 나타냅니다.</p>	
<p>(선택 사항) 새 정책이 적용될 때까지 기다리십시오.</p>	<p>추적에 역할에 대한 합당한 양의 과거 활동이 포함되어 있지 않은 경우, IAM Access Analyzer에서 정확한 정책을 생성하기에 충분한 기록 활동이 있다고 확신할 수 있을 때까지 기다리십시오. 해당 계정에서 충분한 기간 동안 역할이 활성화된 경우에는 이 대기 시간이 필요하지 않을 수 있습니다.</p>	<p>AWS 관리자</p>

작업	설명	필요한 기술
생성된 정책 내용을 수동으로 검토합니다.	CodeCommit 리포지토리에서 생성된.json <ROLE_ARN>파일을 검토하여 허용 및 거부 권한이 역할에 적합한지 확인하십시오.	AWS 관리자

관련 리소스

AWS 리소스

- [IAM Access Analyzer 엔드포인트 및 할당량](#)
- [AWS CLI 구성](#)
- [AWS CDK 시작하기](#)
- [최소 권한](#)

기타 리소스

- [Terraform용 CDK](#) (Terraform 웹사이트)

AWS Amplify에서 호스팅하는 웹 애플리케이션에 대해 AWS WAF 활성화하기

작성자: Karan Shah (AWS) 및 Abhinath Kumar (AWS)

코드 리포지토리: [aws-cdk-amplify-with-waf](#)

환경: 프로덕션

기술: 보안, ID, 규정 준수, 웹 및 모바일 앱

AWS 서비스: AWS WAF, 아마존 CloudFront

요약

Amplify 및 AWS WAF가 직접 통합되지 않기 때문에 AWS Amplify에서 호스팅하는 많은 웹 애플리케이션에는 방화벽이 연결되어 있지 않습니다. 하지만 인바운드 웹 애플리케이션 트래픽을 AWS WAF를 통해 라우팅하여 악용 및 봇으로부터 웹 애플리케이션을 보호할 수 있습니다. AWS WAF 웹 액세스 제어 목록(웹 ACL)은 규칙을 사용하여 애플리케이션이 응답하는 웹 요청을 제어합니다.

이 패턴에서는 기존 또는 제공된 웹 ACL을 Amazon CloudFront 배포에 연결합니다. 그런 다음 AWS WAF를 통해 트래픽을 라우팅하는 배포를 통해 CloudFront Amplify 웹 애플리케이션의 인바운드 트래픽을 전송합니다. 이 프로세스를 체인이라고 합니다. CloudFront 또한 Amplify 관리형 엔드포인트(<https://<branch>.<app-id>.amplifyapp.com>)를 통해 공개 액세스를 거부하도록 Amplify 웹 애플리케이션을 구성하면 사용자가 AWS WAF를 우회하는 것을 방지할 수 있습니다. 애플리케이션으로 향하는 모든 인바운드 트래픽은 새 배포를 통해 라우팅됩니다. CloudFront Amplify 애플리케이션에 액세스하기 위해 단일 보안 인증 정보가 사용되며, 이러한 보안 인증 정보는 AWS Secrets Manager에 저장됩니다. 이 솔루션을 배포하면 사용자가 애플리케이션에 원활하게 액세스할 수 있도록 CloudFront 배포에 이러한 자격 증명이 Authorization 헤더에 추가됩니다.

이 패턴의 코드 리포지토리에는 있는 그대로 사용하거나 기존 Amplify 웹 애플리케이션에 필요한 대로 수정할 수 있는 독립형 샘플 AWS Cloud Development Kit(AWS CDK) 구성이 포함되어 있습니다. 또한 이 코드를 사용하면 Amplify 웹 애플리케이션에 새 코드를 CloudFront 배포할 때마다 새로 생성된 배포의 캐시 무효화를 자동화할 수 있습니다. 자세한 내용은 설명서의 [파일 무효화](#)를 참조하십시오.

CloudFront

사전 조건 및 제한 사항

사전 조건

- 활성 Amazon Web Services(AWS) 계정
- Amplify에서 호스팅하는 기존 웹 애플리케이션
- AWS Command Line Interface(AWS CLI), [설치](#) 및 [구성됨](#)
- AWS Cloud Development Kit(AWS CDK) v2 Toolkit, [설치](#) 및 [구성됨](#)
- Python 3.8 이상, [설치 및 구성됨](#)
- Python에서 AWS CDK 사용에 대한 지식
- AWS CLI 사용에 대한 지식

제한 사항

- Amplify에서는 더 이상 사용자 지정 도메인을 사용할 수 없지만 예는 사용자 지정 도메인을 사용해야 합니다. CloudFront
- 자동화된 암호 교체는 활성화되지 않습니다. Amplify 웹 애플리케이션에서 기본 인증을 위한 새 암호를 생성하려면 스택을 재배포해야 합니다.
- AWS WAF 보호가 필요한 각 Amplify 애플리케이션 및 브랜치에 대해 스택의 새 인스턴스를 배포해야 합니다.

제품 버전

- AWS WAFV2 API
- AWS CDK v2(이 패턴은 버전 2.43.1에서 테스트됨)
- Python 버전 3.8 이상

아키텍처

대상 기술 스택

- AWS Amplify
- 아마존 CloudFront
- 아마존 EventBridge
- AWS Lambda
- AWS Secrets Manager
- AWS WAF

대상 아키텍처

이 다이어그램은 다음 프로세스를 보여 줍니다.

1. 사용자가 Amplify 웹 애플리케이션에 대한 액세스를 요청합니다.
2. AWS WAF 웹 ACL은 Amplify 웹 애플리케이션의 진입점에 대한 네트워크 액세스를 제어하고 제한합니다. 허용된 요청을 배포에 전달합니다. CloudFront
3. CloudFront 배포는 요청에 Authorization 헤더를 삽입합니다. 이 헤더에는 AWS Secrets Manager에 저장된 보안 인증 정보의 Base64 인코딩이 포함되어 있습니다. 그런 다음 CloudFront 배포는 요청을 Amplify 웹 애플리케이션의 사용자 지정 CloudFront 엔드포인트 () 로 전달합니다. `<distribution-ID>.cloudfront.net`
4. Amplify는 일반적인 사용자 인증 단계를 우회하고 Authorization 헤더의 정보를 사용합니다. 사용자가 Amplify 웹 애플리케이션에 대한 액세스를 요청합니다.
5. Amplify가 사전 정의된 브랜치에서 애플리케이션을 성공적으로 배포하면 기본 EventBridge 이벤트 버스의 Amazon EventBridge 규칙이 캐시 무효화 요청을 생성하는 Lambda 함수를 시작합니다. CloudFront

중요: Amplify 엔드포인트의 URL 또는 보안 인증 정보를 공유하지 마십시오. 이렇게 하면 사용자가 이를 애플리케이션의 진입점으로 사용하고 AWS WAF를 우회할 수 없습니다. 그러나 사용자가 엔드포인트 URL을 얻더라도 계속 진행하려면 Amplify 보안 인증 정보를 알아야 합니다.

자동화 및 규모 조정

이 패턴의 아키텍처는 AWS CDK 앱을 통해 배포됩니다. CDK Pipelines를 사용하여 AWS 환경에서 이 솔루션을 자동화하고 확장할 수 있습니다. CDK Pipelines은 AWS CDK 앱을 지속적으로 제공하기 위한 구성 라이브러리 모듈입니다. AWS CDK 앱의 소스 코드를 지원되는 리포지토리에 체크인하면 CDK Pipelines가 자동으로 새 버전을 빌드, 테스트 및 배포할 수 있습니다. 자세한 내용은 AWS CDK 설명서의 [CDK Pipeline을 사용한 지속적 통합 및 전송\(CI/CD\)](#)을 참고하십시오.

참고: 이 AWS CDK 앱은 기존 Amplify 웹 애플리케이션에 구성을 추가합니다. Amplify 애플리케이션에서 수행되는 다른 IaC(코드형 인프라) 작업이 있는 경우 이 CDK 앱은 해당 작업에 영향을 주지 않아야 합니다.

도구

서비스

- [AWS Amplify](#)는 프론트엔드 웹 및 모바일 개발자가 AWS에서 풀스택 애플리케이션을 빠르게 구축할 수 있도록 특별히 제작된 도구 및 기능 세트입니다.
- [AWS Cloud9](#)는 소프트웨어를 코딩, 구축, 실행, 테스트 및 디버깅할 수 있게 해주는 통합 개발 환경(IDE)입니다. 또한, 소프트웨어를 AWS 클라우드로 릴리스하는 데도 도움이 됩니다. 이 패턴으로 AWS Cloud9를 사용하는 것을 권장하지만, Visual Studio Code 또는 IntelliJ IDEA와 같은 다른 IDE를 사용할 수도 있습니다.
- [AWS Cloud Development Kit\(AWS CDK\)](#)는 AWS Cloud 인프라를 코드로 정의하고 프로비저닝하는 데 도움이 되는 소프트웨어 개발 프레임워크입니다.
- [Amazon](#)은 전 세계 데이터 센터 네트워크를 통해 웹 콘텐츠를 전송함으로써 웹 콘텐츠 배포 CloudFront 속도를 높여 지연 시간을 줄이고 성능을 개선합니다.
- [AWS Command Line Interface\(AWS CLI\)](#)는 명령줄 셸에서 명령을 사용하여 AWS 서비스와 상호 작용할 수 있는 오픈 소스 도구입니다.
- [EventBridgeAmazon](#)은 다양한 소스의 실시간 데이터와 애플리케이션을 연결하는 데 도움이 되는 서버리스 이벤트 버스 서비스입니다.
- [AWS Lambda](#)는 서버를 프로비저닝하거나 관리할 필요 없이 코드를 실행하는 데 도움이 되는 컴퓨팅 서비스입니다. 필요할 때만 코드를 실행하며 자동으로 확장이 가능하므로 사용한 컴퓨팅 시간만큼만 비용을 지불합니다.
- [AWS Secrets Manager](#)는 코드의 암호를 포함해 하드 코딩된 보안 인증 정보를 Secrets Manager에서 프로그래밍 방식으로 보안 암호를 검색하는 API 호출로 바꿀 수 있습니다.
- [AWS WAF](#)는 보호되는 웹 애플리케이션 리소스로 전달되는 HTTP 및 HTTPS 요청을 모니터링할 수 있게 해주는 웹 애플리케이션 방화벽입니다.

기타 도구

- [Python](#)은 범용 컴퓨터 프로그래밍 언어입니다.

코드 리포지토리

이 패턴의 코드는 [Amplify 호스팅 웹 응용 프로그램용 WAF GitHub 활성화](#) 리포지토리에서 사용할 수 있습니다. 제공된 AWS CDK 코드를 사용하여 Amplify 웹 애플리케이션이 포함된 AWS 계정에 대상 아키텍처를 배포합니다.

참고: 제공된 AWS CDK 코드를 사용하지 않고도 AWS WAF를 배포할 수 있습니다. 배포에 연결할 수 있는 경우 기존 AWS WAF 인스턴스를 사용할 수 있습니다. CloudFront

에픽

배포 준비

작업	설명	필요한 기술
리포지토리를 복제하고 가상 환경을 설정합니다.	<ol style="list-style-type: none"> 다음 명령을 입력하여 Amplify가 호스팅하는 웹 애플리케이션의 WAF 활성화하기 리포지토리를 IDE에 복제합니다. <pre>git clone https://github.com/aws-samples/aws-cdk-amplify-with-waf.git</pre> 디렉토리를 복제된 리포지토리로 변경합니다. MacOS 또는 Linux를 사용하는 경우 다음 명령을 입력하여 가상 환경을 수동으로 만들고 활성화합니다. <pre>python3 -m venv .venv source .venv/bin/activate</pre> <p>Windows를 사용하는 경우 다음 명령을 입력하여 가상 환경을 수동으로 만들고 활성화합니다.</p> <pre>python -m venv .venv</pre> 	AWS DevOps

작업	설명	필요한 기술
	<pre data-bbox="630 205 1026 310">.venv\Scripts \activate.bat</pre> <p data-bbox="591 319 1026 596">4. Windows를 사용하는 경우 cdk.json 파일에서 "app": "python3 app.py"를 "app": "python app.py"로 변경합니다.</p> <p data-bbox="591 613 1026 697">5. requirements.txt 파일에 정의된 종속성을 설치합니다.</p> <pre data-bbox="630 739 1026 856">pip install -r requirements.txt</pre>	

작업	설명	필요한 기술
<p>AWS CDK를 부트스트랩합니다.</p>	<ol style="list-style-type: none"> 1. 스택을 배포하려는 계정에 대해 올바른 AWS CLI 보안 인증 정보가 구성되어 있는지 확인하십시오. 자세한 내용은 AWS CLI 설정을 참조하세요. 2. 다음 명령을 입력하여 대상 AWS 환경에서 AWS CDK를 부트스트랩합니다. 여기서 <account-id> 는 Amplify 애플리케이션이 호스팅되는 계정의 ID이고 <app-region> 는 AWS 리전입니다. <div data-bbox="630 877 1029 1121" style="border: 1px solid #ccc; border-radius: 10px; padding: 10px; margin: 10px 0;"> <pre>cdk bootstrap aws://<account-id>/us-east-1 aws://<account-id>/<app-region></pre> </div> <p>부트스트래핑 및 대체 명령에 대한 자세한 내용은 부트스트랩하는 방법을 참고하십시오.</p>	<p>AWS DevOps</p>

작업	설명	필요한 기술
<p>웹 ACL을 생성합니다.</p>	<ul style="list-style-type: none"> • 기존 웹 ACL을 사용하려면 웹 ACL의 Amazon 리소스 이름(ARN)을 검색하십시오. 나중에 패턴에서 이 값을 제공해야 합니다. • 이 패턴에 포함된 웹 ACL을 사용하려면 다음 명령을 입력하여 CustomWebAc1Stack AWS CDK 스택을 생성합니다. 그러면 미리 정의된 AWS WAF용 AWS 관리형 규칙 세트를 포함하는 웹 ACL이 생성됩니다. 출력에서 웹 ACL의 ARN을 기록해 둡니다. 나중에 패턴에서 이 값을 제공해야 합니다. <div data-bbox="623 1024 1029 1146" style="border: 1px solid #ccc; border-radius: 10px; padding: 10px; margin-top: 10px;"> <pre>cdk deploy CustomWebAc1Stack</pre> </div>	<p>AWS DevOps</p>

작업	설명	필요한 기술
AWS CDK 스택 파라미터를 구성합니다.	<p>cdk.json 파일에서 다음 매개변수를 업데이트하십시오.</p> <ul style="list-style-type: none"> • <code>app_id</code> – 기존 Amplify 애플리케이션의 ID입니다. 애플리케이션 ID는 애플리케이션 ARN의 마지막 부분이며 다음과 같은 형식으로 되어 있습니다. Amplify 콘솔에서 ARN을 볼 수 있습니다. <pre>arn:<partition>:amplify:<region>:<account-id>:apps/<app-id></pre> <ul style="list-style-type: none"> • <code>branch_name</code> – Amplify 애플리케이션의 대상 브랜치입니다. • <code>web_acl_arn</code> – Amplify 애플리케이션과 연결할 웹 ACL의 ARN입니다. 	AWS DevOps

솔루션 배포

작업	설명	필요한 기술
Amplify용 사용자 지정 배포 스택을 배포합니다.	<p>다음과 같은 명령을 입력하여 Amplify 웹 애플리케이션에 대한 AWS WAF 보호를 활성화하는 AWS CDK 스택을 배포합니다.</p> <pre>cdk deploy CustomAmplifyDistributionStack</pre>	AWS DevOps

작업	설명	필요한 기술
배포를 확인합니다.	<ol style="list-style-type: none"> 1. CustomAmplifyDistributionStack AWS CDK 스택의 출력을 사용하여 웹 애플리케이션을 테스트합니다. CloudFront URL을 통해 애플리케이션에 액세스할 수 있어야 합니다. 2. AWS 웹 애플리케이션의 직접 엔드포인트에 액세스해 보십시오. 보안 인증을 입력하라는 메시지가 표시됩니다. 3. Amplify 웹 애플리케이션을 업데이트하고 변경 사항을 커밋합니다. Amplify가 웹 애플리케이션을 성공적으로 배포하고 사용자 지정 CloudFront 배포가 자동으로 무효화되는지 확인합니다. 	AWS DevOps

관련 리소스

- [CloudFront 도메인 사용자 지정](#) (CloudFront 설명서)
- [AWS CDK 개념](#) (AWS CDK 설명서)
- [AWS WAF에 대한 AWS 관리형 규칙](#) (AWS WAF 설명서)
- [AWS 관리형 규칙 규칙 그룹 목록](#) (AWS WAF 설명서)

AWS CloudFormation 템플릿을 사용하여 Amazon을 GuardDuty 조건부로 활성화합니다.

작성자: Ram Kandaswamy(AWS)

환경: 프로덕션

기술: 보안, ID, 규정 준수, 운영
DevOps

AWS 서비스: AWS, 아마존
CloudFormation GuardDuty,
AWS Lambda, AWS 자격 증명
및 액세스 관리

요약

AWS CloudFormation 템플릿을 사용하여 Amazon Web Services (AWS) GuardDuty 계정에서 Amazon을 활성화할 수 있습니다. 기본적으로, 활성화하는 데 CloudFormation 사용하려고 할 때 이미 GuardDuty 활성화되어 있으면 스택 배포가 실패합니다. 하지만 CloudFormation 템플릿의 조건을 사용하여 이미 GuardDuty 활성화되었는지 여부를 확인할 수 있습니다. CloudFormation 정적 값을 비교하는 조건의 사용을 지원하며, 동일한 템플릿 내에서 다른 리소스 속성의 출력을 사용하는 것은 지원하지 않습니다. 자세한 내용은 CloudFormation 사용 설명서의 [조건을](#) 참조하십시오.

이 패턴에서는 AWS Lambda 함수가 지원하는 CloudFormation 사용자 지정 리소스를 사용하여 아직 활성화되지 않은 경우 조건부로 GuardDuty 활성화합니다. GuardDuty 가 활성화된 경우 스택은 상태를 캡처하여 스택의 출력 섹션에 기록합니다. GuardDuty 가 활성화되지 않은 경우 스택에서 활성화합니다.

사전 조건 및 제한 사항

사전 조건

- 활성 상태의 AWS 계정
- 스택을 생성, 업데이트 및 삭제할 CloudFormation 권한이 있는 AWS Identity 및 Access Management (IAM) 역할

제한 사항

- AWS 계정 또는 지역에 대해 수동으로 비활성화한 경우 GuardDuty , 이 패턴은 해당 대상 계정 또는 지역에 GuardDuty 대해 활성화되지 않습니다.

아키텍처

대상 기술 스택

이 패턴은 코드형 인프라 (IaC) CloudFormation 에 사용됩니다. Lambda 함수가 지원하는 CloudFormation 사용자 지정 리소스를 사용하여 동적 서비스 지원 기능을 구현합니다.

대상 아키텍처

다음 상위 레벨 아키텍처 다이어그램은 템플릿을 배포하여 활성화하는 프로세스를 보여줍니다.

GuardDuty CloudFormation

1. CloudFormation 템플릿을 배포하여 스택을 생성합니다. CloudFormation
2. 스택은 IAM 역할과 Lambda 함수를 생성합니다.
3. Lambda 함수는 IAM 역할을 맡습니다.
4. 대상 AWS 계정에서 이 (GuardDuty 가) 아직 활성화되지 않은 경우 Lambda 함수가 이를 활성화합니다.

자동화 및 규모 조정

AWS CloudFormation StackSet 기능을 사용하여 이 솔루션을 여러 AWS 계정 및 AWS 지역으로 확장할 수 있습니다. 자세한 내용은 CloudFormation 사용 [설명서의 AWS CloudFormation StackSets](#) 사용을 참조하십시오.

도구

- [AWS Command Line Interface\(AWS CLI\)](#)는 명령줄 셸에서 명령을 사용하여 AWS 서비스와 상호 작용할 수 있는 오픈 소스 도구입니다.
- [AWS는 AWS CloudFormation](#) 리소스를 설정하고, 빠르고 일관되게 프로비저닝하고, AWS 계정 및 지역 전반의 수명 주기 전반에 걸쳐 리소스를 관리할 수 있도록 지원합니다.
- [GuardDutyAmazon](#)은 로그를 분석하고 처리하여 AWS 환경에서 예상치 못한 잠재적 무단 활동을 식별하는 지속적 보안 모니터링 서비스입니다.
- [AWS Identity and Access Management\(IAM\)](#)를 사용하면 사용자에 대해 인증 및 권한 부여를 제어함으로써 AWS 리소스에 대한 액세스를 안전하게 관리할 수 있습니다.

- [AWS Lambda](#)는 서버를 프로비저닝하거나 관리할 필요 없이 코드를 실행하는 데 도움이 되는 컴퓨팅 서비스입니다. 필요할 때만 코드를 실행하며 자동으로 확장이 가능하므로 사용한 컴퓨팅 시간만큼만 비용을 지불합니다.

에픽

CloudFormation 템플릿을 생성하고 스택을 배포합니다.

작업	설명	필요한 기술
CloudFormation 템플릿을 생성합니다.	<ol style="list-style-type: none"> 1. 추가 정보 섹션에서 CloudFormation 템플릿의 코드를 복사합니다. 2. 텍스트 편집기에 코드를 붙여넣습니다. 3. 워크스테이션에서 파일을 <code>sample.yaml</code> 로 저장합니다. 	AWS DevOps
CloudFormation 스택 생성.	<ol style="list-style-type: none"> 1. AWS CLI에서 다음 명령을 입력합니다. 그러면 <code>sample.yaml</code> 파일을 사용하여 새 CloudFormation 스택이 생성됩니다. 자세한 내용은 CloudFormation 사용 설명서의 스택 만들기를 참조하십시오. <div style="border: 1px solid #ccc; border-radius: 10px; padding: 10px; margin: 10px 0;"> <pre>aws cloudformation create-stack \ --stack-name guardduty-cf-stack \ --template-body file://sample.yaml</pre> </div> 2. 스택이 성공적으로 생성되었음을 나타내는 다음 값이 AWS CLI에 나타나는지 확 	AWS DevOps

작업	설명	필요한 기술
	<p>인합니다. 스택을 생성하는데 필요한 시간은 다를 수 있습니다.</p> <pre data-bbox="630 380 1029 499">"StackStatus": "CREATE_COMPLETE",</pre>	
<p>AWS 계정에 GuardDuty 활성화되어 있는지 확인합니다.</p>	<ol style="list-style-type: none"> 1. AWS 관리 콘솔에 로그인하고 https://console.aws.amazon.com/guardduty/에서 GuardDuty 콘솔을 엽니다. 2. GuardDuty 서비스가 활성화되었는지 확인하십시오. 	<p>클라우드 관리자, AWS 관리자</p>
<p>추가 계정 또는 AWS 리전을 구성합니다.</p>	<p>사용 사례에 따라 AWS CloudFormation StackSet 기능을 사용하여 이 솔루션을 여러 AWS 계정 및 AWS 지역으로 확장하십시오. 자세한 내용은 CloudFormation 사용 설명서의 AWS CloudFormation StackSets 사용을 참조하십시오.</p>	<p>클라우드 관리자, AWS 관리자</p>

관련 리소스

참조

- [AWS CloudFormation 설명서](#)
- [AWS::Lambda 리소스 유형 참조](#)
- [CloudFormation 리소스 유형: AWS::IAM::Role](#)
- [CloudFormation 리소스 유형: AWS::GuardDuty::Detector](#)
- [AWS를 사용하여 AWS 서비스 속성을 검색하는 네 가지 방법 CloudFormation \(블로그\)](#)

튜토리얼 및 동영상

- [AWS를 사용한 인프라 관리 간소화 CloudFormation](#) (자습서)
- [GuardDuty Amazon과 AWS Security Hub를 사용하여 여러 계정을 보호하십시오](#) (AWS re:Invent 2020).
- [AWS 작성 모범 사례 CloudFormation](#) (AWS re:Invent 2019)
- [AWS에서의 위협 탐지: 아마존 소개 GuardDuty](#) (AWS re:Force 2019)

추가 정보

CloudFormation 템플릿

```

AWSTemplateFormatVersion: 2010-09-09
Resources:
  rLambdaLogGroup:
    Type: 'AWS::Logs::LogGroup'
    DeletionPolicy: Delete
    Properties:
      RetentionInDays: 7
      LogGroupName: /aws/lambda/resource-checker
  rLambdaCheckerLambdaRole:
    Type: 'AWS::IAM::Role'
    Properties:
      RoleName: !Sub 'resource-checker-lambda-role-${AWS::Region}'
      AssumeRolePolicyDocument:
        Version: 2012-10-17
        Statement:
          - Effect: Allow
            Principal:
              Service: lambda.amazonaws.com
            Action: 'sts:AssumeRole'
    Path: /
    Policies:
      - PolicyName: !Sub 'resource-checker-lambda-policy-${AWS::Region}'
        PolicyDocument:
          Version: 2012-10-17
          Statement:
            - Sid: CreateLogGroup
              Effect: Allow
              Action:
                - 'logs:CreateLogGroup'

```

```

        - 'logs:CreateLogStream'
        - 'logs:PutLogEvents'
        - 'iam:CreateServiceLinkedRole'
        - 'cloudformation:CreateStack'
        - 'cloudformation>DeleteStack'
        - 'cloudformation:Desc*'
        - 'guardduty:CreateDetector'
        - 'guardduty:ListDetectors'
        - 'guardduty>DeleteDetector'
    Resource: '*'
resourceCheckerLambda:
  Type: 'AWS::Lambda::Function'
  Properties:
    Description: Checks for resource type enabled and possibly name to exist
    FunctionName: resource-checker
    Handler: index.lambda_handler
    Role: !GetAtt
      - rLambdaCheckerLambdaRole
      - Arn
    Runtime: python3.8
    MemorySize: 128
    Timeout: 180
    Code:
      ZipFile: |
        import boto3
        import os
        import json
        from botocore.exceptions import ClientError
        import cfnresponse

        guardduty=boto3.client('guardduty')
        cfn=boto3.client('cloudformation')

        def lambda_handler(event, context):
            print('Event: ', event)
            if 'RequestType' in event:
                if event['RequestType'] in ["Create","Update"]:
                    enabled=False
                    try:
                        response=guardduty.list_detectors()
                        if "DetectorIds" in response and len(response["DetectorIds"])>0:
                            enabled="AlreadyEnabled"

```

```

        elif "DetectorIds" in response and
len(response["DetectorIds"])==0:
            cfn_response=cfn.create_stack(
                StackName='guardduty-cfn-stack',
                TemplateBody='{ "AWSTemplateFormatVersion": "2010-09-09",
"Description": "A sample template",    "Resources": { "IRWorkshopGuardDutyDetector": {
"Type": "AWS::GuardDuty::Detector",    "Properties": {    "Enable": true  }  } } }'
            )
            enabled="True"
        except Exception as e:
            print("Exception: ",e)
        responseData = {}
        responseData['status'] = enabled
        cfnresponse.send(event, context, cfnresponse.SUCCESS, responseData,
"CustomResourcePhysicalID" )
        elif event['RequestType'] == "Delete":
            cfn_response=cfn.delete_stack(
                StackName='guardduty-cfn-stack')
            cfnresponse.send(event, context, cfnresponse.SUCCESS, {})

CheckResourceExist:
  Type: 'Custom::LambdaCustomResource'
  Properties:
    ServiceToken: !GetAtt
      - resourceCheckerLambda
      - Arn
Outputs:
  status:
    Value: !GetAtt
      - CheckResourceExist
      - status

```

Lambda 리소스의 대체 코드 옵션

제공된 CloudFormation 템플릿은 더 쉽게 참조하고 안내할 수 있도록 인라인 코드를 사용하여 Lambda 리소스를 참조합니다. 또는 Amazon Simple Storage Service (Amazon S3) 버킷에 Lambda 코드를 배치하고 템플릿에서 참조할 수 있습니다. CloudFormation 인라인 코드는 패키지 종속성 또는 라이브러리를 지원하지 않습니다. Lambda 코드를 S3 버킷에 배치하고 템플릿에서 참조하여 이를 지원할 수 있습니다. CloudFormation

아래의 코드 행을 바꿉니다.

Code:

```
ZipFile: |
```

다음 코드 행 사용:

Code:

```
S3Bucket: <bucket name>  
S3Key: <python file name>  
S3ObjectVersion: <version>
```

S3 버킷에서 버전 관리를 사용하지 않는 경우 S3ObjectVersion 속성을 생략할 수 있습니다. 자세한 내용은 Amazon S3 사용 설명서의 [S3 버킷에서 버전 관리 사용](#)을 참조하세요.

Amazon RDS for SQL Server에서 투명한 데이터 암호화 활성화하기

작성자: Ranga Cherukuri (AWS)

환경: PoC 또는 파일럿

기술: 보안, 자격 증명, 규정 준수, 분석, 데이터베이스

워크로드: Microsoft

AWS 서비스: Amazon RDS

요약

이 패턴에서는 Amazon Relational Database Service(Amazon RDS)에서 투명한 데이터 암호화(TDE)를 구현하여 SQL Server가 저장 데이터를 암호화하는 방법을 설명합니다.

사전 조건 및 제한 사항

사전 조건

- 활성 상태의 AWS 계정.
- Amazon RDS for SQL Server DB 인스턴스

제품 버전

Amazon RDS는 현재 다음 SQL Server 버전과 에디션에 TDE를 지원합니다.

- SQL Server 2012 Enterprise Edition
- SQL Server 2014 Enterprise Edition
- SQL Server 2016 Enterprise Edition
- SQL Server 2017 Enterprise Edition
- SQL Server 2019 Standard Edition 및 Enterprise Edition

지원되는 버전 및 에디션에 대한 최신 정보는 Amazon RDS 설명서의 [SQL Server의 투명한 데이터 암호화 지원](#)을 참고하십시오.

아키텍처

기술 스택

- Amazon RDS for SQL Server

아키텍처

도구

도구

- Microsoft SQL Server Management Studio(SSMS)는 SQL Server 인프라를 관리하기 위한 통합 환경입니다. SQL Server와 상호 작용하는 다양한 스크립트 편집기와 함께 사용자 인터페이스와 도구 그룹을 제공합니다.

에픽

Amazon RDS 콘솔에서 옵션 그룹 생성

작업	설명	필요한 기술
Amazon RDS 콘솔을 엽니다.	AWS Management Console에 로그인하고 Amazon RDS 콘솔 을 엽니다.	개발자, DBA
옵션 그룹을 생성합니다.	탐색 창에서 옵션 그룹, 그룹 생성을 선택합니다. sqlserver-ee를 DB 엔진으로 선택한 다음, 엔진 버전을 선택합니다.	개발자, DBA
TRANSPARENT_DATA_ENCRYPTION 옵션을 추가합니다.	생성한 옵션 그룹을 편집하고 TRANSPARENT_DATA_ENCRYPTION 이라는 옵션을 추가합니다.	개발자, DBA

옵션 그룹을 DB 인스턴스에 연동시킵니다.

작업	설명	필요한 기술
DB 인스턴스를 선택합니다.	Amazon RDS 콘솔의 탐색 창에서 데이터베이스를 선택한 후, 옵션 그룹과 연결할 DB 인스턴스를 선택합니다.	개발자, DBA
옵션 그룹을 DB 인스턴스에 연결합니다.	수정을 선택한 다음, 옵션 그룹 설정을 사용하여 SQL Server DB 인스턴스를 이전에 만든 옵션 그룹과 연결합니다.	개발자, DBA
변경 사항을 적용합니다.	변경 사항을 즉시 적용하거나 원한다면 다음 유지 관리 기간에 적용합니다.	개발자, DBA
인증서 이름을 가져옵니다.	다음 쿼리를 사용하여 기본 인증서 이름을 가져옵니다. <pre>USE [master] GO SELECT name FROM sys.certificates WHERE name LIKE 'RDSTDECertificate%' GO</pre>	개발자, DBA

데이터베이스 암호화 키 생성

작업	설명	필요한 기술
SSMS를 사용하여 Amazon RDS for SQL Server DB 인스턴스에 연결합니다.	지침은 Microsoft 설명서의 SSMS 사용 을 참조하세요.	개발자, DBA

작업	설명	필요한 기술
<p>기본 인증서를 사용하여 데이터베이스 암호화 키를 생성합니다.</p>	<p>앞서 받은 기본 인증서 이름을 사용하여 데이터베이스 암호화 키를 생성합니다. 다음 T-SQL 쿼리를 사용하여 데이터베이스 암호화 키를 생성합니다. AES_256 대신 AES_256 알고리즘을 지정할 수 있습니다.</p> <pre data-bbox="597 632 1024 1031"> USE [Databasename] GO CREATE DATABASE ENCRYPTION KEY WITH ALGORITHM = AES_128 ENCRYPTION BY SERVER CERTIFICATE [certific atename] GO </pre>	<p>개발자, DBA</p>
<p>데이터베이스의 암호화를 활성화합니다.</p>	<p>다음 T-SQL 쿼리를 사용하여 데이터베이스 암호화를 활성화합니다.</p> <pre data-bbox="597 1234 1024 1434"> ALTER DATABASE [Database Name] SET ENCRYPTION ON GO </pre>	<p>개발자, DBA</p>

작업	설명	필요한 기술
암호화 상태를 확인합니다.	<p>다음 T-SQL 쿼리를 사용하여 암호화 상태를 확인합니다.</p> <pre data-bbox="597 348 1024 663">SELECT DB_NAME(database_id) AS DatabaseName, encryption_state, percent_complete FROM sys.dm_database_encryption_keys</pre>	개발자, DBA

관련 리소스

- [Amazon RDS 설명서의 SQL Server의 투명한 데이터 암호화 지원](#)을 참조하세요.
- [옵션 그룹 사용](#) (Amazon RDS 설명서)
- [Amazon RDS DB 인스턴스 수정하기](#)(Amazon RDS 설명서)
- Amazon RDS 설명서의 [SQL Server의 투명한 데이터 암호화 지원](#)을 참조하세요.
- [SSMS 사용](#) (Microsoft 설명서)

AWS CloudFormation 스택이 승인된 S3 버킷에서 시작되었는지 확인

작성자: Chandini Penmetsa(AWS)

환경: 프로덕션

기술: 보안, 자격 증명, 규정 준수

워크로드: 기타 모든 워크로드

AWS 서비스: 아마존 SNS, AWS, 아마존 CloudFormation CloudWatch, AWS Lambda, 아마존 S3

요약

AWS CloudFormation 템플릿을 사용하여 Amazon Web Services (AWS) 리소스를 프로그래밍 방식으로 설정할 수 있으므로 리소스를 관리하는 시간을 줄이고 AWS에서 실행되는 애플리케이션에 더 많은 시간을 집중할 수 있습니다. 이 패턴은 특정 Amazon Simple Storage Service (Amazon S3) 버킷에 저장된 템플릿에서만 AWS CloudFormation 스택이 생성되었는지 확인하는 방법을 제공합니다. 이 검사는 허용 목록에 있는 S3 버킷에 저장된 템플릿을 사용하도록 요구하는 보안 또는 규정 준수 요구 사항이 있는 경우에 유용합니다.

이 보안 제어는 AWS CloudFormation [CreateStack](#) 및 [UpdateStack](#) API 호출을 모니터링하고 호출에 사용된 템플릿이 승인된 S3 버킷에서 가져온 것인지 확인하는 AWS Lambda 함수를 호출합니다. 템플릿을 승인되지 않은 버킷에서 가져온 경우, Lambda 함수는 관련 정보와 함께 사용자에게 Amazon Simple Notification Service(SNS) 이메일 알림을 트리거합니다.

사전 조건 및 제한 사항

사전 조건

- 위반 알림을 받을 활성 이메일 주소
- 제공된 Lambda 코드를 업로드하기 위한 S3 버킷
- 승인된 S3 버킷 이름 목록

제한 사항

- [UpdateStack](#) Amazon EventBridge 이벤트에서 S3 버킷의 URL을 사용할 수 없기 때문에 승인되지 않은 S3 버킷의 기존 템플릿을 사용하는 API 호출은 추가 위반을 생성하지 않습니다. 원래 [CreateStack](#) 위반 알림을 받은 후에는 승인되지 않은 S3 버킷에서 기존 템플릿을 삭제하는 것이 좋습니다.
- 이 보안 제어는 템플릿이 처음 배포된 후 업데이트를 처리하므로 다음 AWS CloudFormation 이벤트를 모니터링하지 않습니다: [CreateChangeSet](#), [CreateStackSet](#), [UpdateStackSet](#).
- 모니터링할 모든 AWS 리전에 이 보안 제어를 배포해야 합니다.

아키텍처

대상 기술 스택

- AWS Lambda
- Amazon SNS
- 아마존 EventBridge 를

대상 아키텍처

자동화 및 규모 조정

[AWS Organizations](#)를 사용하는 경우, [CloudFormation StackSets](#)를 사용하여 모니터링하려는 여러 계정에 이 템플릿을 배포할 수 있습니다.

도구

- [AWS 클라우드포메이션](#) — 모델을 사용하여 AWS 리소스를 모델링하고 설정하는 데 도움이 됩니다. infrastructure-as-code
- [Amazon EventBridge](#) — 자체 애플리케이션, SaaS software-as-a-service (애플리케이션) 및 AWS 서비스의 실시간 데이터 스트림을 제공하고 해당 데이터를 AWS Lambda와 같은 대상으로 라우팅합니다.
- [AWS Lambda](#) – 서버를 프로비저닝하거나 관리하지 않고도 코드를 실행할 수 있습니다.
- [Amazon SNS](#)-게시자에서 구독자에게 메시지를 전송합니다. 구독자는 구독하는 주제에 게시된 모든 메시지를 수신하며 주제에 대한 모든 구독자는 동일한 메시지를 수신합니다.
- [Amazon S3](#)-인터넷을 통해 언제 어디서든 원하는 양의 데이터를 저장하고 검색할 수 있습니다.

에픽

보안 제어의 배포

작업	설명	필요한 기술
Amazon S3로 Lambda 코드를 업로드합니다.	"첨부 파일" 섹션에 나와 있는 Lambda 코드가 포함된 .zip 파일을 신규 또는 기존 S3 버킷에 업로드합니다. 이 버킷은 평가할 리소스와 동일한 AWS 리전에 있어야 합니다.	클라우드 아키텍트
AWS CloudFormation 템플릿을 배포합니다.	S3 버킷과 동일한 리전에서 AWS CloudFormation 콘솔을 열고 "첨부 파일" 섹션에 제공된 템플릿을 배포합니다. 파라미터 값을 제공합니다. 이 값은 "추가 정보" 섹션에 설명되어 있습니다.	클라우드 아키텍트

구독 확인

작업	설명	필요한 기술
Amazon SNS 주제에 대한 구독을 확인합니다.	AWS CloudFormation 템플릿이 성공적으로 배포되면 제공한 이메일 주소로 구독 이메일이 전송됩니다. 알림을 받기 시작하려면 이메일 구독을 확인해야 합니다.	클라우드 아키텍트

관련 리소스

- [AWS CloudFormation 템플릿 배포](#)
- [아마존 EventBridge](#)

- [AWS Lambda](#)
- [Amazon S3](#)

추가 정보

이 패턴으로 제공된 AWS CloudFormation 템플릿을 배포하면 다음 정보를 입력하라는 메시지가 표시됩니다.

- S3 버킷: 첨부된 Lambda 코드(.zip 파일)를 업로드한 버킷을 지정합니다. 새 버킷을 만들거나 기존 버킷을 지정할 수 있습니다.
- S3 키: S3 버킷에 있는 Lambda .zip 파일의 위치를 지정합니다(예: 파일 이름.zip 또는 컨트롤/파일 이름.zip). 선행 슬래시 마크는 사용하면 안 됩니다.
- 알림 이메일: 위반 알림을 보내야 하는 활성 이메일 주소를 입력합니다.
- Lambda 로깅 수준: Lambda 함수의 로깅 수준을 지정합니다. 정보를 사용하여 진행 상황에 대한 자세한 정보 메시지를 기록하고, 배포를 계속할 수 있는 오류 이벤트의 경우 오류를 기록하고, 잠재적으로 유해한 상황에 대한 경고를 기록할 수 있습니다.
- 승인된 버킷: 선택으로 구분된 승인된 S3 버킷 목록을 제공합니다.

첨부

이 문서와 관련된 추가 콘텐츠에 액세스하려면 [attachment.zip](#) 파일의 압축을 풉니다.

AWS 로드 밸런서가 보안 리스너 프로토콜(HTTPS, SSL/TLS)을 사용하는지 확인

작성자: Chandini Penmetsa(AWS) 및 Purushotham G K(AWS)

환경: 프로덕션

기술: 보안, 자격 증명, 규정 준수

워크로드: 기타 모든 워크로드

AWS 서비스: 아마존 SNS, AWS, 아마존 CloudFormation, AWS Lambda CloudWatch, Elastic Load Balancing (ELB)

요약

Amazon 웹 서비스(AWS) 클라우드에서 Elastic Load Balancing은 들어오는 애플리케이션 트래픽을 Amazon Elastic Compute Cloud(Amazon EC2) 인스턴스, 컨테이너, IP 주소, AWS Lambda 함수 등 여러 타겟에 자동으로 분산합니다. 로드 밸런서는 리스너를 사용하여 로드 밸런서가 사용자의 트래픽을 수락하는 데 사용하는 포트와 프로토콜을 정의합니다. Application Load Balancer는 애플리케이션 계층에서 라우팅을 결정하고 HTTP/HTTPS 프로토콜을 사용합니다. Network Load Balancer는 전송 계층에서 라우팅을 결정하며 TCP(전송 제어 프로토콜), 전송 계층 보안(TLS), UDP(사용자 데이터그램 프로토콜) 또는 TCP_UDP 프로토콜을 사용합니다. Classic Load Balancer는 TCP 또는 SSL(보안 소켓 계층) 프로토콜을 사용하여 전송 계층에서 또는 HTTP/HTTPS를 사용하여 애플리케이션 계층에서 라우팅 결정을 내립니다.

조직에는 로드 밸런서가 HTTPS 또는 SSL/TLS와 같은 보안 프로토콜에서만 사용자의 트래픽을 수락해야 한다는 보안 또는 규정 준수 요구 사항이 있을 수 있습니다.

이 패턴은 Amazon EventBridge 규칙을 사용하여 애플리케이션 로드 밸런서와 네트워크 로드 밸런서에 대한 CreateListener 및 ModifyListener API 호출과 클래식 로드 밸런서에 대한 CreateLoadBalancerListeners 및 CreateLoadBalancer API 호출을 모니터링하는 보안 제어를 제공합니다. HTTP, TCP/UDP 또는 TCP_UDP가 로드 밸런서의 리스너 프로토콜에 사용되는 경우 제어는 Lambda 함수를 호출합니다. Lambda 함수는 Amazon Simple Notification Service(SNS) 주제에 메시지를 게시하여 로드 밸런서 세부 정보가 포함된 알림을 전송합니다.

사전 조건 및 제한 사항

사전 조건

- 활성 상태의 AWS 계정.
- 위반 알림을 수신할 이메일 주소
- Lambda 코드 .zip 파일을 저장하기 위한 Amazon Simple Storage Service(S3) 버킷

제한 사항

- 이 보안 제어는 로드 밸런서 리스너를 업데이트하지 않는 한 기존 로드 밸런서를 확인하지 않습니다.
- 이 보안 제어는 리전과 관련이 있으므로 모니터링하고자 하는 AWS 리전에 배포해야 합니다.

아키텍처

대상 기술 스택

- Lambda 함수
- Amazon SNS 주제
- EventBridge 규칙

대상 아키텍처

자동화 및 규모 조정

- AWS Organizations를 사용하는 경우, [AWS StackSets Cloudformation](#)을 사용하여 모니터링하려는 여러 계정에 이 템플릿을 배포할 수 있습니다.

도구

- [AWS CloudFormation](#) — CloudFormation AWS는 인프라를 코드로 사용하여 AWS 리소스를 모델링하고 설정하는 데 도움이 되는 서비스입니다.
- [Amazon EventBridge](#) — EventBridge Amazon은 자체 애플리케이션, 서비스형 소프트웨어 (SaaS) 애플리케이션 및 AWS 서비스로부터 실시간 데이터 스트림을 전달하여 해당 데이터를 Lambda 함수와 같은 대상으로 라우팅합니다.

- [AWS Lambda](#) - Lambda는 서버를 프로비저닝하거나 관리하지 않고도 코드를 실행할 수 있습니다.
- [Amazon S3](#)-Amazon Simple Storage Service(S3)는 웹 사이트, 모바일 애플리케이션, 백업, 데이터 레이크 등 다양한 스토리지 솔루션에 사용할 수 있는 확장성이 뛰어난 객체 스토리지 서비스입니다.
- [Amazon SNS](#) - Amazon Simple Notification Service(SNS)는 웹 서버와 이메일 주소를 포함하여 게시자와 클라이언트 간에 메시지를 전달 또는 전송하는 것을 조정하고 관리합니다. 구독자는 구독하는 주제에 게시된 모든 메시지를 수신하며 주제에 대한 모든 구독자는 동일한 메시지를 수신합니다.

모범 사례

사용된 SNS 주제에 공개적으로 액세스할 수 없도록 하세요. 자세한 내용은 [AWS 설명서](#)를 참조하세요.

에픽

Lambda 코드 업로드

작업	설명	필요한 기술
S3 버킷을 정의합니다.	Amazon S3 콘솔에서 선행 슬라이드를 포함하지 않는 고유한 이름을 가진 S3 버킷을 선택하거나 생성합니다. S3 버킷 이름은 전역 수준에서 고유하며, 네임스페이스는 모든 AWS 계정이 공유합니다. S3 버킷은 평가 중인 로드 밸런서와 같은 리전에 있어야 합니다.	클라우드 아키텍트
Lambda 코드를 S3 버킷에 업로드	“첨부 파일” 섹션에 나와 있는 Lambda 코드.zip 파일을 정의된 S3 버킷에 업로드합니다.	클라우드 아키텍트
AWS CloudFormation 템플릿을 배포합니다.	AWS CloudFormation 콘솔의 S3 버킷과 동일한 AWS 리전에 “첨부 파일” 섹션에 제공된 템플릿을 배포합니다. 다음 에픽	클라우드 아키텍트

작업	설명	필요한 기술
	에서는 파라미터 값을 제공하세요.	

CloudFormation 파라미터

작업	설명	필요한 기술
S3 버킷에 이름을 지정합니다.	첫 번째 에픽에서 생성한 S3 버킷의 이름을 입력합니다.	클라우드 아키텍트
Amazon S3 접두사를 입력합니다.	S3 버킷의 Lambda 코드 .zip 파일 위치를 선행 슬래시 없이 입력합니다(예: <directory>/<file-name>.zip).	클라우드 아키텍트
SNS 주제 ARN을 입력합니다.	위반 알림에 기존 SNS 주제를 사용하려는 경우 SNS 주제 Amazon 리소스 이름(ARN)을 입력합니다. 새 SNS 주제를 생성하려면 값을 None (기본값)으로 유지하세요.	클라우드 아키텍트
이메일 주소를 입력합니다.	Amazon SNS 알림을 수신할 활성 이메일 주소를 입력합니다.	클라우드 아키텍트
로깅 수준을 정의합니다.	Lambda 함수의 로깅 수준 및 빈도를 정의합니다. Info는 애플리케이션 진행 상황에 대한 자세한 정보 메시지를 지정합니다. Error는 애플리케이션을 계속 실행할 수 있게 해주는 오류 이벤트를 지정합니다. Warning은 잠재적으로 유해한 상황을 지정합니다.	클라우드 아키텍트

CloudFormation 템플릿 배포

작업	설명	필요한 기술
템플릿을 다운로드합니다.	첨부 파일 섹션에 제공된 CloudFormation 템플릿을 다운로드합니다.	클라우드 아키텍트
스택을 생성합니다.	S3 버킷과 동일한 지역에서 CloudFormation 서비스 콘솔로 이동하여 다운로드한 템플릿을 배포합니다. 파라미터 세부 정보는 이전 서사시를 참조하세요.	클라우드 아키텍트
리소스를 확인합니다.	스택이 완전히 생성되면 리소스 탭으로 이동하여 리소스를 확인합니다. 템플릿은 다음 리소스를 만듭니다. <ul style="list-style-type: none"> • EventBridge 규칙 • Lambda 함수 • Lambda 실행 역할 • Lambda 호출 권한 	클라우드 아키텍트

구독 확인

작업	설명	필요한 기술
구독을 확인합니다.	템플릿이 성공적으로 배포되면 새 SNS 주제가 생성되면 파라미터에 제공된 이메일 주소로 구독 이메일 메시지가 전송됩니다. 위반 알림을 받으려면 이 이메일 구독을 확인해야 합니다.	클라우드 아키텍트

문제 해결

문제	Solution
스택 생성이 실패했습니다. 중에 오류가 발생했습니다 GetObject. S3 오류 코드: Permanent Redirect. S3 오류 메시지: 버킷이 xx-xxxx-1 리전에 있습니다. 이 리전을 사용하여 요청을 다시 시도하세요.	S3 버킷 리전과 스택이 배포되는 리전이 동일한지 확인하세요.
스택 생성이 실패했습니다. python3.6의 런타임 파라미터는 AWS Lambda 함수를 생성 또는 업데이트하는 데 더 이상 지원되지 않습니다.	186행에서 다운로드한 템플릿을 Python 버전 3.6에서 3.9로 업데이트합니다.

관련 리소스

- [AWS CloudFormation 콘솔에 스택 생성](#)
- [AWS Lambda](#)
- [Classic Load Balancer란 무엇인가요?](#)
- [Application Load Balancer란 무엇인가요?](#)
- [Network Load Balancer란 무엇인가요?](#)
- [AWS Lambda 함수 사용에 대한 모범 사례](#)
- [AWS CloudFormation 모범 사례](#)

첨부

이 문서와 관련된 추가 콘텐츠에 액세스하려면 [attachment.zip](#) 파일의 압축을 풉니다.

시작 시 Amazon EMR 저장 데이터에 대한 암호화가 활성화되었는지 확인

작성자: Priyanka Chaudhary(AWS)

환경: 프로덕션

기술: 보안, 자격 증명, 규정 준수, 분석

워크로드: 오픈 소스

AWS 서비스: 아마존 EMR, 아마존 SNS, AWS KMS, AWS, CloudFormation AWS Lambda, 아마존 S3

요약

이 패턴은 Amazon Web Services(AWS)에서 Amazon EMR 클러스터의 암호화를 모니터링하기 위한 보안 제어를 제공합니다.

데이터 암호화는 권한 없는 사용자가 클러스터 및 관련 데이터 스토리지 시스템에서 데이터를 읽는 것을 방지하는 데 도움이 됩니다. 여기에는 네트워크를 이동하는 동안 가로챌 수 있는 데이터인 전송 중 데이터와 영구 미디어에 저장된 데이터인 저장 데이터가 포함됩니다. Amazon Simple Storage Service(S3)에 저장된 데이터는 두 가지 방법으로 암호화할 수 있습니다.

- Amazon S3 관리형 키를 사용한 서버 측 암호화(SSE-S3)
- AWS Key Management Service(AWS KMS) 키를 사용한 서버 측 암호화(SSE-KMS), Amazon EMR에 적합한 정책으로 설정됩니다.

이 보안 제어는 API 호출을 모니터링하고 에서 [RunJobFlow](#) Amazon CloudWatch Events 이벤트를 시작합니다. 이 트리거는 Python 스크립트를 실행하는 AWS Lambda를 호출합니다. 함수는 이벤트 JSON 입력에서 EMR 클러스터 ID를 검색하고 다음 검사를 수행하여 보안 위반 여부를 확인합니다.

1. EMR 클러스터가 Amazon EMR 전용 보안 구성과 연결되어 있는지 확인합니다.
2. Amazon EMR 전용 보안 구성이 EMR 클러스터와 연결된 경우 저장 시 암호화가 켜져 있는지 확인합니다.

3. 저장 시 암호화가 켜져 있지 않은 경우 EMR 클러스터 이름, 위반 세부 정보, AWS 리전, AWS 계정, 그리고 이 알림의 출처가 되는 Lambda Amazon 리소스 이름(ARN)이 포함된 Amazon Simple Notification Service(SNS) 알림을 보냅니다.

사전 조건 및 제한 사항

사전 조건

- 활성 상태의 AWS 계정
- Lambda 코드 .zip 파일을 위한 S3 버킷
- 위반 알림을 수신할 이메일 주소
- 모든 API 로그를 검색할 수 있도록 Amazon EMR 로깅 해제

제한 사항

- 이 보안 제어는 리전과 관련이 있으므로 모니터링하고자 하는 AWS 리전에 배포해야 합니다.

제품 버전

- Amazon EMR 릴리스 4.8.0 이상

아키텍처

대상 기술 스택

- Amazon EMR
- 아마존 CloudWatch 이벤트 이벤트
- Lambda 함수
- Amazon SNS

대상 아키텍처

자동화 및 규모 조정

- AWS Organizations를 사용하는 경우, [AWS StackSets Cloudformation](#)을 사용하여 모니터링하려는 여러 계정에 이 템플릿을 배포할 수 있습니다.

도구

도구

- [AWS CloudFormation](#) — CloudFormation AWS는 인프라를 코드로 사용하여 AWS 리소스를 모델링하고 설정하는 데 도움이 되는 서비스입니다.
- [Amazon CloudWatch Events](#) — Amazon CloudWatch Events는 AWS 리소스의 변경 사항을 설명하는 시스템 이벤트의 스트림을 거의 실시간으로 제공합니다.
- [Amazon EMR](#) - Amazon EMR은 빅 데이터 프레임워크 실행을 간소화하는 관리형 클러스터 플랫폼입니다.
- [AWS Lambda](#)-AWS Lambda는 서버를 프로비저닝하거나 관리하지 않고도 코드를 실행할 수 있습니다.
- [Amazon S3](#)-Amazon S3는 웹 사이트, 모바일 애플리케이션, 백업, 및 데이터 레이크 등 다양한 스토리지 솔루션에 사용할 수 있는 확장성이 뛰어난 객체 스토리지 서비스입니다.
- [Amazon SNS](#)-Amazon SNS는 게시자와 클라이언트 간에 웹 서버와 이메일 주소를 포함한 메시지 전달 또는 전송을 조정하고 관리합니다. 구독자는 구독하는 주제에 게시된 모든 메시지를 수신하며 주제에 대한 모든 구독자는 동일한 메시지를 수신합니다.

code

- 이 프로젝트의 EMR EncryptionAtRest .zip 및 EMR EncryptionAtRest .yml 파일은 첨부 파일로 제공됩니다.

에픽

S3 버킷 정의

작업	설명	필요한 기술
S3 버킷을 정의합니다.	Amazon S3 콘솔에서 선행 스크립트를 포함하지 않는 고유한	클라우드 아키텍트

작업	설명	필요한 기술
	이름을 가진 S3 버킷을 선택하거나 생성합니다. S3 버킷 이름은 전역 수준에서 고유하며, 네임스페이스는 모든 AWS 계정이 공유합니다. S3 버킷은 평가 중인 Amazon EMR 클러스터와 동일한 리전에 있어야 합니다.	

Lambda 코드를 S3 버킷에 업로드

작업	설명	필요한 기술
Lambda 코드를 S3 버킷에 업로드	“첨부 파일” 섹션에 나와 있는 Lambda 코드.zip 파일을 정의된 S3 버킷에 업로드합니다.	클라우드 아키텍트

AWS CloudFormation 템플릿 배포

작업	설명	필요한 기술
AWS CloudFormation 템플릿을 배포합니다.	AWS CloudFormation 콘솔의 S3 버킷과 동일한 리전에 이 패턴의 첨부 파일로 제공된 AWS CloudFormation 템플릿을 배포합니다. 다음 예픽에서 파라미터에 대한 값을 입력합니다. AWS CloudFormation 템플릿 배포에 대한 자세한 내용은 “관련 리소스” 섹션을 참조하십시오.	클라우드 아키텍트

AWS CloudFormation 템플릿에서 파라미터 완성

작업	설명	필요한 기술
S3 버킷에 이름을 지정합니다.	첫 번째 에픽에서 생성한 S3 버킷의 이름을 입력합니다.	클라우드 아키텍트
Amazon S3 키를 입력합니다.	S3 버킷에 있는 Lambda 코드 .zip 파일의 위치를 앞에 슬래시 없이 입력합니다(예: <directory>/<file-name>.zip).	클라우드 아키텍트
이메일 주소를 입력합니다.	Amazon SNS 알림을 수신할 활성 이메일 주소를 입력합니다.	클라우드 아키텍트
로깅 수준을 정의합니다.	Lambda 함수의 로깅 수준 및 빈도를 정의합니다. '정보'는 애플리케이션 진행 상황에 대한 자세한 정보 메시지를 의미합니다. '오류'는 여전히 애플리케이션을 계속 실행할 수 있는 오류 이벤트를 의미합니다. '경고'는 잠재적으로 위험한 상황을 의미합니다.	클라우드 아키텍트

구독 확인

작업	설명	필요한 기술
구독을 확인합니다.	템플릿이 성공적으로 배포되면 입력한 이메일 주소로 구독 이메일 메시지가 전송됩니다. 위반 알림을 받으려면 이 이메일 구독을 확인해야 합니다.	클라우드 아키텍트

관련 리소스

- [AWS CloudFormation 콘솔에 스택 생성](#)
- [AWS Lambda](#)
- [Amazon EMR 암호화 옵션](#)

첨부 파일

이 문서와 관련된 추가 콘텐츠에 액세스하려면 [attachment.zip](#) 파일의 압축을 풉니다.

IAM 프로파일이 EC2 인스턴스와 연결되었는지 확인

작성자: Mansi Suratwala(AWS)

환경: 프로덕션

기술: 인프라, 보안, 자격증명, 규정 준수

AWS 서비스: 아마존 EC2, AWS 자격 증명 및 액세스 관리, 아마존, AWS Lambda CloudWatch, 아마존 SNS

요약

이 패턴은 Amazon Elastic Compute Cloud (Amazon EC2) 인스턴스에 대해 AWS ID 및 액세스 관리 (IAM) 프로파일 위반이 발생할 경우 자동 알림을 설정하는 AWS CloudFormation 보안 제어 템플릿을 제공합니다.

인스턴스 프로파일은 IAM 역할을 위한 컨테이너로서 인스턴스 시작 시 EC2 인스턴스에 역할 정보를 전달하는 데 사용됩니다.

Amazon CloudWatch Events는 AWS가AssociateIamInstanceProfile, ReplaceIamInstanceProfileAssociation 및 작업을 기반으로 RunInstances Amazon EC2 API 호출을 CloudTrail 기록할 때 이 검사를 시작합니다. 트리거는 CloudWatch Amazon 이벤트 이벤트를 사용하여 IAM 프로파일을 확인하는 AWS Lambda 함수를 호출합니다.

IAM 프로파일이 없는 경우 Lambda 함수는 Amazon Web Services(AWS) 계정 ID 및 AWS 리전이 포함된 Amazon Simple Notification Service(SNS) 이메일 알림을 시작합니다.

IAM 프로파일이 있는 경우 Lambda 함수는 정책 문서에 와일드카드 항목이 있는지 확인합니다. 와일드카드 항목이 있는 경우 Amazon SNS 위반 알림을 시작하여 보안을 강화하는 데 도움이 됩니다. 알림에는 IAM 프로파일 이름, 이벤트, EC2 인스턴스 ID, 관리형 정책 이름, 위반, 계정 ID, 및 리전이 포함됩니다.

사전 조건 및 제한 사항

사전 조건

- 활성 상태의 계정

- Lambda 코드 .zip 파일을 위한 Amazon Simple Storage Service(S3) 버킷

제한 사항

- AWS CloudFormation 템플릿은 RunInstancesAssociateIamInstanceProfile, 및 ReplaceIamInstanceProfileAssociation 작업에 대해서만 배포해야 합니다.
- 보안 제어는 IAM 프로파일의 분리를 모니터링하지 않습니다.
- 보안 제어는 EC2 인스턴스 IAM 프로파일에 연결된 IAM 정책의 수정 여부를 확인하지 않습니다.
- "Resource":*를 사용해야 하는 [지원되지 않는 리소스 수준 권한](#)은 보안 제어에 고려되지 않습니다.

아키텍처

대상 기술 스택

- Amazon EC2
- AWS CloudTrail
- 아마존 CloudWatch
- AWS Lambda
- Amazon S3
- Amazon SNS

대상 아키텍처

자동화 및 규모 조정

여러 AWS 지역 및 계정에 대해 AWS CloudFormation 템플릿을 여러 번 사용할 수 있습니다. 템플릿은 각 계정 또는 리전에서 한 번만 실행하면 됩니다.

도구

도구

- [Amazon EC2](#)-Amazon EC2는 AWS Cloud에서 확장 가능한 컴퓨팅 용량(가상 서버)을 제공합니다.

- [AWS CloudTrail](#) — CloudTrail AWS는 AWS 계정의 거버넌스, 규정 준수, 운영 및 위험 감사를 가능하게 합니다. 사용자, 역할 또는 AWS 서비스가 수행한 작업은 에서 이벤트로 기록됩니다 CloudTrail.
- [Amazon CloudWatch Events](#) — Amazon CloudWatch Events는 AWS 리소스의 변경 사항을 설명하는 시스템 이벤트의 스트림을 거의 실시간으로 제공합니다.
- [AWS Lambda](#)-AWS Lambda는 서버를 프로비저닝하거나 관리하지 않고도 코드를 실행할 수 있게 해주는 컴퓨팅 서비스입니다. Lambda는 필요 시에만 코드를 실행하며, 일일 몇 개의 요청에서 초당 수천 개의 요청까지 자동으로 규모를 조정합니다.
- [Amazon S3](#)-Amazon S3는 웹 사이트, 모바일 애플리케이션, 백업, 및 데이터 레이크 등 다양한 스토리지 솔루션에 사용할 수 있는 확장성이 뛰어난 객체 스토리지를 제공합니다.
- [Amazon SNS](#)-Amazon SNS는 애플리케이션과 디바이스가 클라우드에서 알림을 전송 및 수신할 수 있게 해줍니다.

코드

- 프로젝트의 .zip 파일은 첨부 파일로 제공됩니다.

에픽

S3 버킷 정의

작업	설명	필요한 기술
S3 버킷을 정의합니다.	Lambda 코드 .zip 파일을 호스트하려면, 앞에 슬래시가 없는 고유한 이름을 가진 S3 버킷을 선택하거나 생성합니다. S3 버킷 이름은 전역 수준에서 고유하며, 네임스페이스는 모든 AWS 계정이 공유합니다. S3 버킷은 평가 중인 EC2 인스턴스와 동일한 리전에 있어야 합니다.	클라우드 아키텍트

Lambda 코드를 S3 버킷에 업로드

작업	설명	필요한 기술
Lambda 코드를 S3 버킷에 업로드	첨부 파일 섹션에 나와 있는 Lambda 코드를 S3 버킷에 업로드합니다. S3 버킷은 평가 중인 EC2 인스턴스와 동일한 리전에 있어야 합니다.	클라우드 아키텍트

AWS CloudFormation 템플릿 배포

작업	설명	필요한 기술
AWS CloudFormation 템플릿을 배포합니다.	이 패턴에 첨부하여 제공된 AWS CloudFormation 템플릿을 배포하십시오. 다음 에픽에서는 파라미터 값을 제공하세요.	클라우드 아키텍트

AWS CloudFormation 템플릿에서 파라미터 완성

작업	설명	필요한 기술
S3 버킷에 이름을 지정합니다.	첫 번째 에픽에서 생성한 S3 버킷의 이름을 입력합니다.	클라우드 아키텍트
S3 키를 입력합니다.	S3 버킷의 Lambda 코드 .zip 파일 위치를 선행 슬래시 없이 입력합니다(예: <directory>/<file-name>.zip).	클라우드 아키텍트
이메일 주소를 입력합니다.	Amazon SNS 알림을 수신할 활성 이메일 주소를 입력합니다.	클라우드 아키텍트

작업	설명	필요한 기술
로깅 수준을 정의합니다.	Lambda 함수의 로깅 수준 및 빈도를 정의합니다. Info는 애플리케이션 진행 상황에 대한 자세한 정보 메시지를 지정합니다. Error는 애플리케이션을 계속 실행할 수 있게 해주는 오류 이벤트를 지정합니다. Warning은 잠재적으로 유해한 상황을 지정합니다.	클라우드 아키텍트

구독 확인

작업	설명	필요한 기술
구독을 확인합니다.	템플릿이 성공적으로 배포되면 입력한 이메일 주소로 구독 이메일 메시지가 전송됩니다. 위반 알림을 받으려면 이 이메일 구독을 확인해야 합니다.	클라우드 아키텍트

관련 리소스

- [S3 버킷 생성](#)
- [S3 버킷에 파일 업로드](#)
- [인스턴스 프로파일 사용](#)
- [AWS를 사용하여 AWS API 호출에서 트리거되는 CloudWatch 이벤트 규칙 생성 CloudTrail](#)

첨부

이 문서와 관련된 추가 콘텐츠에 액세스하려면 [attachment.zip](#) 파일의 압축을 풉니다.

Amazon Redshift 클러스터를 생성할 때 암호화되었는지 확인합니다.

작성자: Mansi Suratwala(AWS)

환경: 프로덕션

기술: 분석, 데이터 레이크, 보안, 자격 증명, 규정 준수

워크로드: 기타 모든 워크로드

AWS 서비스: 아마존 레드시프트, 아마존 SNS, AWS, 아마존, AWS CloudTrail Lambda CloudWatch, 아마존 S3

요약

이 패턴은 새 Amazon Redshift 클러스터가 암호화 없이 생성될 때 자동 알림을 제공하는 AWS CloudFormation 템플릿을 제공합니다.

AWS CloudFormation 템플릿은 아마존 CloudWatch 이벤트 이벤트와 AWS Lambda 함수를 생성합니다. 이 이벤트는 AWS를 통해 스냅샷에서 생성되거나 복원되는 Amazon Redshift 클러스터를 감시합니다. CloudTrail AWS 계정에서 AWS 키 관리 서비스 (AWS KMS) 또는 클라우드 하드웨어 보안 모델 (HSM) 암호화 없이 클러스터를 생성한 경우, Lambda 함수를 시작하여 CloudWatch 위반 사실을 알리는 Amazon Simple Notification Service (Amazon SNS) 알림을 보냅니다.

사전 조건 및 제한 사항

사전 조건

- 활성 상태의 AWS 계정.
- 클러스터 서브넷 그룹이 있는 Virtual Private Cloud(VPC) 및 연결된 보안 그룹.

제한 사항

- AWS CloudFormation 템플릿은 CreateCluster 및 RestoreFromClusterSnapshot 작업용으로만 배포할 수 있습니다.

아키텍처

대상 기술 스택

- Amazon Redshift
- AWS CloudTrail
- 아마존 CloudWatch
- AWS Lambda
- Amazon Simple Storage Service (S3)
- Amazon SNS

대상 아키텍처

자동화 및 규모 조정

여러 AWS 지역 및 계정에 대해 AWS CloudFormation 템플릿을 여러 번 사용할 수 있습니다. 각 리전 또는 계정에서 한 번만 실행해야 합니다.

도구

도구

- [Amazon Redshift](#) - Amazon Redshift는 클라우드에서 완벽하게 관리되는 페타바이트 규모의 데이터 웨어하우스 서비스입니다. Amazon Redshift는 데이터 레이크와 통합되므로 데이터를 사용하여 비즈니스 및 고객에 대한 새로운 인사이트를 얻을 수 있습니다.
- [AWS CloudTrail](#) — CloudTrail AWS는 AWS 계정의 거버넌스, 규정 준수, 운영 및 위험 감사를 구현하는 데 도움이 되는 AWS 서비스입니다. 사용자, 역할 또는 AWS 서비스가 수행한 작업은 에서 이벤트 트로 기록됩니다 CloudTrail.
- [Amazon CloudWatch Events](#) — Amazon CloudWatch Events는 AWS 리소스의 변경 사항을 설명하는 시스템 이벤트의 스트림을 거의 실시간으로 제공합니다.
- [AWS Lambda](#) - AWS Lambda는 서버를 프로비저닝하거나 관리하지 않고도 코드를 실행할 수 있습니다. AWS Lambda는 필요 시에만 코드를 실행하며, 일일 몇 개의 요청에서 초당 수천 개의 요청까지 자동으로 규모를 조정합니다.
- [Amazon S3](#) - Amazon S3는 웹 사이트, 모바일 애플리케이션, 백업, 데이터 레이크 등 다양한 스토리지 솔루션에 사용할 수 있는 확장성이 뛰어난 객체 스토리지 서비스입니다.

- [Amazon SNS](#) – Amazon Simple Notification Service(SNS)는 웹 서버와 이메일 주소를 포함하여 게시자와 클라이언트 간에 메시지를 전달 또는 전송하는 것을 조정하고 관리합니다.

코드

- 프로젝트의 .zip 파일은 첨부 파일로 제공됩니다.

에픽

S3 버킷 정의

작업	설명	필요한 기술
S3 버킷을 정의합니다.	Amazon S3 콘솔에서 S3 버킷을 선택하거나 생성합니다. 이 S3 버킷은 Lambda 코드 .zip 파일을 호스팅합니다. S3 버킷은 평가 중인 Amazon Redshift 클러스터와 동일한 리전에 있어야 합니다. S3 버킷 이름에는 선행 슬래시를 포함할 수 없습니다.	클라우드 아키텍트

Lambda 코드를 S3 버킷에 업로드

작업	설명	필요한 기술
Lambda 코드를 S3 버킷에 업로드	첨부 파일 섹션에 나와 있는 Lambda 코드를 S3 버킷에 업로드합니다. S3 버킷은 평가 중인 Amazon Redshift 클러스터와 동일한 리전에 있어야 합니다.	클라우드 아키텍트

AWS CloudFormation 템플릿 배포

작업	설명	필요한 기술
AWS CloudFormation 템플릿을 배포합니다.	이 패턴에 첨부하여 제공된 AWS CloudFormation 템플릿을 배포하십시오. 다음 에픽에서는 파라미터 값을 제공하세요.	클라우드 아키텍트

AWS CloudFormation 템플릿에서 파라미터 완성

작업	설명	필요한 기술
S3 버킷에 이름을 지정합니다.	첫 번째 에픽에서 생성한 S3 버킷의 이름을 입력합니다.	클라우드 아키텍트
S3 키를 입력합니다.	S3 버킷의 Lambda 코드 .zip 파일 위치를 선행 슬래시 없이 입력합니다(예: <directory>/<file-name>.zip).	클라우드 아키텍트
이메일 주소를 입력합니다.	Amazon SNS 알림을 수신할 활성 이메일 주소를 입력합니다.	클라우드 아키텍트
로깅 수준을 정의합니다.	Lambda 함수의 로깅 수준 및 빈도를 정의합니다. Info는 애플리케이션 진행 상황에 대한 자세한 정보 메시지를 지정합니다. Error는 애플리케이션을 계속 실행할 수 있게 해주는 오류 이벤트를 지정합니다. Warning은 잠재적으로 유해한 상황을 지정합니다.	클라우드 아키텍트

구독 확인

작업	설명	필요한 기술
구독을 확인합니다.	템플릿이 성공적으로 배포되면 입력한 이메일 주소로 구독 이메일 메시지가 전송됩니다. 위반 알림을 받으려면 이 이메일 구독을 확인해야 합니다.	클라우드 아키텍트

관련 리소스

- [S3 버킷 생성](#)
- [S3 버킷에 파일 업로드](#)
- [AWS를 사용하여 AWS API 호출에서 트리거되는 CloudWatch 이벤트 규칙 생성 CloudTrail](#)
- [Amazon Redshift 클러스터 생성](#)

첨부

이 문서와 관련된 추가 콘텐츠에 액세스하려면 [attachment.zip](#) 파일의 압축을 풉니다.

를 사용하여 AWS IAM ID 센터 자격 증명 및 할당에 대한 보고서를 내보냅니다. PowerShell

작성자: Jorge Pava(AWS), Chad Miles(AWS), Frank Allotta(AWS) 및 Manideep Reddy Gillela(AWS)

환경: 프로덕션

기술: 보안, 자격 증명, 규정 준수, 관리 및 거버넌스

워크로드: Microsoft

AWS 서비스: IAM 자격 증명 센터, AWS 도구 PowerShell

요약

AWS IAM Identity Center(AWS Single Sign-On의 후속)를 사용하여 모든 Amazon Web Services(AWS) 계정과 클라우드 애플리케이션에 대한 AWS Single Sign-On(SSO) 액세스를 중앙에서 관리하는 경우, AWS Management Console을 통해 이러한 할당을 보고하고 감사하는 것은 지루하고 시간이 많이 걸릴 수 있습니다. 수십 또는 수백 개의 AWS 계정에 대한 사용자 또는 그룹의 권한을 보고하는 경우 특히 그렇습니다.

대부분의 경우 이 정보를 보는 데 이상적인 도구는 Microsoft Excel과 같은 스프레드시트 애플리케이션입니다. 이를 통해 AWS Organizations에서 관리하는 전체 조직의 데이터를 필터링, 검색 및 시각화할 수 있습니다.

이 패턴은 AWS Tools를 사용하여 IAM ID 센터에서 SSO 자격 증명 구성 보고서를 생성하는 방법을 설명합니다. PowerShell 보고서는 CSV 파일 형식이며, 여기에는 ID 이름(주체), ID 유형(사용자 또는 그룹), ID가 액세스할 수 있는 계정, 권한 집합이 포함됩니다. 이 보고서를 생성한 후 원하는 애플리케이션에서 열어 필요에 따라 데이터를 검색, 필터링 및 감사할 수 있습니다. 다음 이미지는 스프레드시트 애플리케이션의 샘플 데이터를 보여줍니다.

중요: 이 보고서에는 민감한 정보가 포함되어 있으므로 안전하게 저장하고 need-to-know 기준으로만 공유하는 것이 좋습니다.

사전 조건 및 제한 사항

사전 조건

- 구성 및 활성화된 IAM Identity Center 및 AWS Organizations
- PowerShell, 설치 및 구성. 자세한 내용은 [설치 PowerShell](#) (Microsoft 설명서) 를 참조하십시오.
- 설치 및 구성을 위한 PowerShell AWS 도구. 성능상의 이유로, 라는 AWS Tools를 위한 모듈화된 버전의 AWS Tools를 설치하는 것이 좋습니다. PowerShell AWS.Tools 각 AWS 서비스는 개별적인 소형 모듈에서 지원됩니다. PowerShell 셸에서 다음 명령을 입력하여 이 패턴에 필요한 모듈을 설치합니다.AWS.Tools.Installer,, OrganizationsSSOAdmin,. IdentityStore

```
Install-Module AWS.Tools.Installer
Install-AWSToolsModule -Name Organizations, SSOAdmin, IdentityStore
```

자세한 내용은 [Windows에 AWS.Tools 설치 또는 리눅스 또는 macOS에 AWS.Tools 설치 \(설명서용 AWS 도구\)](#) 를 참조하십시오. PowerShell 모듈을 설치할 때 오류가 발생하는 경우 이 패턴의 [문제 해결](#) 섹션을 참조하세요.

- 다음 중 하나를 수행하여 사전에 AWS Command Line Interface(AWS CLI) 또는 AWS SDK를 작업 가능한 보안 인증 정보로 구성해야 합니다.
 - AWS CLI `aws configure`를 사용합니다. 자세한 내용은 [빠른 구성](#)(AWS CLI 설명서)을 참조하세요.
 - AWS Identity and Access Management(IAM) 역할을 통해 임시로 액세스할 수 있도록 AWS CLI 또는 AWS Cloud Development Kit(AWS CDK)를 구성합니다. 자세한 내용은 [CLI 액세스를 위한 IAM 역할 보안 인증 정보 가져오기](#)(IAM Identity Center 설명서)를 참조하세요.
- 다음과 같은 IAM 주체의 보안 인증 정보를 저장한 AWS CLI의 명명된 프로필
 - AWS Organizations 관리 계정 또는 IAM Identity Center의 위임된 관리자 계정에 대한 액세스 권한이 있음.
 - AWSSS0ReadOnly 및 AWSSS0DirectoryReadOnly AWS 관리형 정책이 적용됨.

자세한 내용은 [명명된 프로필 사용](#)(AWS CLI 설명서) 및 [AWS 관리형 정책](#)(IAM 설명서)를 참조하세요.

제한 사항

- 대상 AWS 계정은 AWS Organizations의 조직으로 관리되어야 합니다.

제품 버전

- [모든 운영 체제에서는 버전 7.0 이상을 사용하는 PowerShell 것이 좋습니다.](#)

아키텍처

대상 아키텍처

1. 사용자가 PowerShell 명령줄에서 스크립트를 실행합니다.
2. 스크립트가 AWS CLI의 명명된 프로필을 취합니다. 이를 통해 IAM Identity Center에 대한 액세스 권한이 부여됩니다.
3. 스크립트가 IAM Identity Center에서 SSO ID 구성을 검색합니다.
4. 스크립트가 저장된 로컬 워크스테이션의 동일한 디렉터리에 스트립트가 CSV 파일을 생성합니다.

도구

서비스

- [AWS Command Line Interface \(AWS CLI\)](#)는 명령줄 셸에서 명령을 사용하여 AWS 서비스와 상호 작용할 수 있는 오픈 소스 도구입니다.
- [AWS IAM Identity Center](#)를 사용하면 모든 AWS 계정과 클라우드 애플리케이션에 대한 AWS Single Sign-On(SSO) 액세스를 중앙에서 관리할 수 있습니다.
- [AWS Tools for PowerShell](#)는 PowerShell 명령줄에서 AWS 리소스의 작업을 스크립팅하는 데 도움이 되는 PowerShell 모듈 세트입니다.

기타 도구

- [PowerShell](#) Windows, Linux 및 macOS에서 실행되는 Microsoft 자동화 및 구성 관리 프로그램입니다.

에픽

보고서를 생성합니다.

작업	설명	필요한 기술
스크립트를 준비합니다.	<ol style="list-style-type: none"> 이 패턴의 추가 정보 섹션에 있는 PowerShell 스크립트를 복사하십시오. Param 섹션에서 AWS 환경에 대해 다음 변수의 값을 정의합니다. <ul style="list-style-type: none"> OutputFile -보고서의 파일 이름입니다. ProfileName -보고서 생성에 사용하려는 AWS CLI의 명명된 프로파일입니다. Region-IAM Identity Center가 배포된 AWS 리전입니다. 전체 리전 및 코드 목록은 리전별 엔드포인트를 참조하세요. SS0-Report.ps1 이름으로 스크립트 파일을 저장합니다. 	클라우드 관리자
스크립트를 실행합니다.	<p>다음 명령을 사용하여 PowerShell 셸에서 사용자 지정 스크립트를 실행하는 것이 좋습니다.</p> <pre>.\SS0-Report.ps1</pre>	클라우드 관리자

작업	설명	필요한 기술
	<p>다음 명령을 입력하여 다른 셸에서 스크립트를 실행할 수도 있습니다.</p> <pre data-bbox="597 380 1026 457">pwsh .\SSO-Report.ps1</pre> <p>스크립트는 스크립트 파일과 같은 디렉터리에 CSV 파일을 생성합니다.</p>	
<p>보고서 데이터를 분석합니다.</p>	<p>출력 CSV 파일에는 헤더 AccountNamePermissionSet, 주체 및 유형이 있습니다. 원하는 스프레드시트 애플리케이션에서 이 파일을 엽니다. 데이터 테이블을 만들어 결과를 필터링하고 정렬할 수 있습니다.</p>	<p>클라우드 관리자</p>

문제 해결

문제	Solution
<p>The term 'Get-<parameter>' is not recognized as the name of a cmdlet, function, script file, or operable program. 오류</p>	<p>PowerShell 또는 해당 모듈용 AWS Tools 가 설치되지 않았습니니다. PowerShell 셸에서 다음 명령을 입력하여 AWS Tools를 설치하고 이 패턴에 필요한 모듈을 설치합니다 (AWS.Tools.Installer , Organizations SSOAdmin, 및)IdentityStore . PowerShell</p> <pre data-bbox="831 1665 1507 1864">Install-Module AWS.Tools.Installer Install-AWSToolsModule -Name Organizations, SSOAdmin, IdentityS tore</pre>

문제	Solution
<p>No credentials specified or obtained from persisted/shell defaults 오류</p>	<p>에픽 섹션의 스크립트 준비에서 ProfileName 및 Region 변수를 올바르게 입력했는지 확인합니다. 명명된 프로필의 설정 및 보안 인증 정보에 IAM Identity Center를 관리할 수 있는 충분한 권한이 있는지 확인합니다.</p>
<p>AWS.tools 모듈을 설치할 때 발생하는 Authenticode Issuer ... 오류.</p>	<p>-SkipPublisherCheck 파라미터를 Install-AWSToolsModule 명령 끝에 추가합니다.</p>
<p>Get-ORGAccountList : Assembly AWSSDK.SSO could not be found or loaded. 오류</p>	<p>이 오류는 명명된 AWS CLI 프로필이 지정되고, AWS CLI가 IAM Identity Center를 통해 사용자를 인증하도록 구성되고, AWS CLI가 새로 고쳐진 인증 토큰을 자동으로 검색하도록 구성된 경우 발생할 수 있습니다. 이 오류를 해결하려면 다음을 수행합니다.</p> <ol style="list-style-type: none"> 1. 다음 명령을 입력하여 SSO 및 SSO0IDC 모듈이 설치되었는지 확인합니다. <div data-bbox="868 1113 1507 1192" style="border: 1px solid #ccc; border-radius: 10px; padding: 5px; margin: 10px 0;"> <pre>Install-AWSToolsModule SSO, SSO0IDC</pre> </div> 2. 다음 라인을 스크립트의 param() 블록 아래에 삽입합니다. <div data-bbox="868 1333 1507 1413" style="border: 1px solid #ccc; border-radius: 10px; padding: 5px; margin: 10px 0;"> <pre>Import-Module AWS.Tools.SSO</pre> </div> <div data-bbox="868 1444 1507 1524" style="border: 1px solid #ccc; border-radius: 10px; padding: 5px; margin: 10px 0;"> <pre>Import-Module AWS.Tools.SSO0IDC</pre> </div>

관련 리소스

- [구성 설정이 저장되는 장소는 어딘가요? \(AWS CLI 설명서\)](#)
- [AWS IAM Identity Center를 사용하도록 AWS CLI 구성\(AWS CLI 설명서\)](#)
- [명명된 프로필 사용\(AWS CLI 설명서\)](#)

추가 정보

다음 스크립트에서 다음 파라미터의 값을 업데이트해야 하는지 여부를 결정합니다.

- AWS CLI의 명명된 프로필을 사용하여 IAM Identity Center가 구성된 계정에 액세스하는 경우 \$ProfileName 값을 업데이트합니다.
- IAM Identity Center를 AWS CLI 또는 AWS SDK 구성의 기본 리전과 다른 AWS 리전에 배포하는 경우, IAM Identity Center가 배포된 리전을 사용하도록 \$Region 값을 업데이트합니다.
- 이러한 상황 중 어느 것도 해당되지 않는 경우에는 스크립트 업데이트가 필요하지 않습니다.

```
param (
    # The name of the output CSV file
    [String] $OutputFile = "SSO-Assignments.csv",
    # The AWS CLI named profile
    [String] $ProfileName = "",
    # The AWS Region in which IAM Identity Center is configured
    [String] $Region      = ""
)
$Start = Get-Date; $OrgParams = @{}
If ($Region){ $OrgParams.Region = $Region}
if ($ProfileName){$OrgParams.ProfileName = $ProfileName}
$SSOParams = $OrgParams.Clone(); $IdsParams = $OrgParams.Clone()
$AccountList = Get-ORGAccountList @OrgParams | Select-Object Id, Name
$SSOinstance = Get-SSOADMNInstanceList @OrgParams
$SSOParams['InstanceArn'] = $SSOinstance.InstanceArn
$IdsParams['IdentityStoreId'] = $SSOinstance.IdentityStoreId
$PSsets = @{}; $Principals = @{}
$Assignments = @(); $AccountCount = 1; Write-Host ""
foreach ($Account in $AccountList) {
    $Duration = New-Timespan -Start $Start -End (Get-Date) | ForEach-Object
    {[Timespan]::New($_.Days, $_.Hours, $_.Minutes, $_.Seconds)}
    Write-Host "`r$Duration - Account $AccountCount of $($AccountList.Count)
    (Assignments:$($Assignments.Count))" -NoNewline
    $AccountCount++
    foreach ($PS in Get-SSOADMNPermissionSetsProvisionedToAccountList -AccountId
    $Account.Id @SSOParams) {
        if (-not $PSsets[$PS]) {$PSsets[$PS] = (Get-SSOADMNPermissionSet @SSOParams -
        PermissionSetArn $PS).Name;$APICalls++}
        $AssignmentsResponse = Get-SSOADMNAccountAssignmentList @SSOParams -
        PermissionSetArn $PS -AccountId $Account.Id
```

```

    if ($AssignmentsResponse.NextToken) {$AccountAssignments =
$AssignmentsResponse.AccountAssignments}
    else {$AccountAssignments = $AssignmentsResponse}
    While ($AssignmentsResponse.NextToken) {
        $AssignmentsResponse = Get-SSOADMNAccountAssignmentList @SSOParams -
PermissionSetArn $PS -AccountId $Account.Id -NextToken $AssignmentsResponse.NextToken
        $AccountAssignments += $AssignmentsResponse.AccountAssignments}
    foreach ($Assignment in $AccountAssignments) {
        if (-not $Principals[$Assignment.PrincipalId]) {
            $AssignmentType = $Assignment.PrincipalType.Value
            $Expression      = "Get-IDS"+$AssignmentType+" @IdsParams -"+
$AssignmentType+"Id "+$Assignment.PrincipalId
            $Principal       = Invoke-Expression $Expression
            if ($Assignment.PrincipalType.Value -eq "GROUP")
{ $Principals[$Assignment.PrincipalId] = $Principal.DisplayName }
            else { $Principals[$Assignment.PrincipalId] = $Principal.UserName }
        }
        $Assignments += [PSCustomObject]@{
            AccountName      = $Account.Name
            PermissionSet    = $PSsets[$PS]
            Principal        = $Principals[$Assignment.PrincipalId]
            Type              = $Assignment.PrincipalType.Value}
    }
}
}
$Duration = New-Timespan -Start $Start -End (Get-Date) | ForEach-Object
{[Timespan]::New($_.Days, $_.Hours, $_.Minutes, $_.Seconds)}
Write-Host "`r$($AccountList.Count) accounts done in $Duration. Outputting result to
$OutputFile"
$Assignments | Sort-Object Account | Export-CSV -Path $OutputFile -Force

```

예정된 AWS KMS 키 삭제 모니터링 및 문제 해결

작성자: Mikesh Khanal(AWS), Ramya Pulipaka(AWS)

환경: 프로덕션	기술: 보안, 자격 증명, 규정 준수, DevOps	AWS 서비스: 아마존 SNS, AWS CloudTrail, 아마존 CloudWatch
----------	------------------------------	--

요약

Amazon Web Services(AWS) 클라우드에서 AWS Key Management Service(AWS KMS) 키를 삭제하면 데이터가 손실될 수 있습니다. 삭제하면 AWS KMS 키와 연결된 키 자료 및 모든 메타데이터가 제거되며 이 작업은 되돌릴 수 없습니다. AWS KMS 키가 삭제되면 더 이상 해당 AWS KMS 키로 암호화된 데이터를 해독할 수 없으므로 데이터를 복구할 수 없습니다.

이 패턴은 애플리케이션 또는 사용자가 AWS KMS 키를 삭제하도록 예약할 때 알림을 통해 모니터링을 설정합니다. 알림을 받는 경우 AWS KMS 키 삭제를 취소하고 삭제 결정을 재고해 볼 수 있습니다. [이 패턴은 AWS Systems Manager 자동화 AWSConfigRemediation CancelKeyDeletion 런북을 사용하여 AWS KMS 키 삭제를 쉽게 취소할 수 있습니다.](#)

참고: AWS KMS 키 삭제를 모니터링하려는 모든 AWS 지역에 패턴 CloudFormation 템플릿을 배포해야 합니다.

사전 조건 및 제한 사항

사전 조건

- 활성 상태의 AWS 계정
- 다음과 같은 AWS 서비스에 대한 이해:
 - 아마존 EventBridge
 - AWS KMS
 - Amazon Simple Notification Service(Amazon SNS)
 - AWS Systems Manager

제한 사항

- 솔루션을 사용자 지정하려면 이 패턴에 사용되는 AWS CloudFormation 템플릿과 AWS 서비스에 대한 지식이 필요합니다.
- 현재 이 솔루션은 기본 이벤트 버스를 사용하며 요구 사항에 따라 사용자 지정할 수 있습니다. 사용자 지정 이벤트 버스에 대한 자세한 내용은 [AWS 설명서](#)를 참조하세요.

아키텍처

대상 기술 스택

- 아마존 EventBridge
- AWS KMS
- Amazon SNS
- AWS Systems Manager
- 다음을 사용하는 자동화:
 - AWS Command Line Interface(AWS CLI) 또는 AWS SDK
 - AWS CloudFormation 스택

대상 아키텍처

1. AWS KMS 키 삭제가 예정되어 있습니다.
2. 예약 삭제 이벤트는 규칙에 따라 평가됩니다. EventBridge
3. 이 EventBridge 규칙은 Amazon SNS 주제를 다룹니다.
4. 이 EventBridge 규칙은 Systems Manager 자동화 및 런북을 시작합니다.
5. 런북은 삭제를 취소합니다.

자동화 및 규모 조정

CloudFormation 스택은 이 솔루션이 작동하는 데 필요한 모든 리소스와 서비스를 배포합니다. 패턴은 단일 계정에서 독립적으로 실행하거나 여러 독립 계정 또는 조직에 CloudFormation StackSets 대해 AWS를 사용하여 실행할 수 있습니다.

```
aws cloudformation create-stack --stack-name <stack-name>\
```

```
--template-body file://<Full-Path-of-file> \
--parameters ParameterKey=,ParameterValue= \
--capabilities CAPABILITY_NAMED_IAM
```

도구

도구

- [AWS CloudFormation](#) — CloudFormation AWS는 Amazon Web Services 리소스를 모델링하고 설정하여 리소스를 관리하는 시간을 줄이고 AWS에서 실행되는 애플리케이션에 더 많은 시간을 집중할 수 있도록 지원하는 서비스입니다. CloudFormation 템플릿을 사용하여 AWS 지역의 AWS 계정에 스택을 생성할 수 있습니다. 템플릿은 사용자가 원하는 모든 AWS 리소스를 설명하고 해당 리소스를 자동으로 CloudFormation 프로비저닝 및 구성합니다.
- [AWS CLI](#) – AWS Command Line Interface(AWS CLI)는 명령줄 셸에서 명령을 사용하여 AWS 서비스와 상호 작용할 수 있는 오픈 소스 도구입니다.
- [Amazon EventBridge](#) — EventBridge Amazon은 애플리케이션을 다양한 소스의 데이터와 연결하는 서버리스 이벤트 버스 서비스입니다. EventBridge 자체 애플리케이션 및 AWS 서비스의 실시간 데이터 스트림을 제공하고 해당 데이터를 AWS Lambda와 같은 대상으로 라우팅합니다. EventBridge 이벤트 기반 아키텍처 구축 프로세스를 간소화합니다.
- [AWS KMS](#) - AWS Key Management Service(AWS KMS)는 데이터 암호화에 사용하는 암호화 키인 AWS KMS 키를 생성하고 제어하는 관리형 서비스입니다.
- [AWS SDK](#) - AWS 도구에는 원하는 프로그래밍 언어로 AWS에서 애플리케이션을 개발하고 관리할 수 있는 SDK가 포함되어 있습니다.
- [Amazon SNS](#) – Amazon Simple Notification Service(SNS)는 게시자에서 구독자(생산자 및 소비자라고도 함)로 메시지를 전송하는 관리형 서비스입니다. 게시자는 논리적 액세스 지점 및 커뮤니케이션 채널인 주제에 메시지를 전송하여 구독자와 비동기식으로 통신합니다.
- [AWS Systems Manager](#) – AWS Systems Manager는 AWS에서 인프라를 보고 제어하기 위해 사용할 수 있는 AWS 서비스입니다. Systems Manager 콘솔을 사용하여 AWS 리소스에서 운영 태스크를 자동화할 수 있습니다. Systems Manager는 관리형 인스턴스를 검사하고 탐지된 정책 위반을 보고하거나 시정 조치를 취해서 보안 및 규정 준수를 유지하는 데 도움이 됩니다.

code

- 프로젝트 alerting_ct_logs.yaml CloudFormation 템플릿이 첨부되었습니다.

에픽

AWS 계정 준비

작업	설명	필요한 기술
AWS CLI를 설치하고 구성합니다.	<p>AWS CLI 버전 2를 설치합니다. 그런 다음 자격 증명, 기본 출력 형식 및 AWS CLI가 AWS와 상호 작용하는 데 사용하는 기본 AWS 리전에 대한 보안 보안 인증 정보 설정을 구성합니다.</p> <p>자격 증명에는 작업을 수행하는 데 필요한 권한이 있어야 합니다.</p>	개발자, 보안 엔지니어

AWS CloudFormation 템플릿 배포

작업	설명	필요한 기술
CloudFormation 템플릿을 다운로드하세요.	첨부 파일을 컴퓨터의 로컬 경로에 다운로드하고 <code>alerting_ct_logs.yaml</code> 템플릿 파일을 추출합니다.	개발자, 보안 엔지니어
템플릿을 배포합니다.	<p>AWS 계정 프로필이 구성된 터미널 창에서 다음 명령을 실행합니다.</p> <pre>aws cloudformation create-stack --stack-name <stack_name> \ --capabilities <Value> \</pre>	개발자, 보안 엔지니어

작업	설명	필요한 기술
	<pre data-bbox="597 205 1024 898"> --template-body file:// <Full_Path> \ --parameters Parameter Key=DestinationEma ilAddress,Paramete rValue=<Value> \ ParameterKey=SNS TopicName,Paramete rValue=<Value> \ ParameterKey=Ena bleRemedi ation ,Paramete rValue=<Value> \ ParameterKey=Aut omationAssumeRole, ParameterValue=<Va lue> </pre> <p data-bbox="597 940 1024 1024">다음 단계에서 템플릿 파라미터 값을 입력합니다.</p>	

작업	설명	필요한 기술
<p>템플릿 파라미터를 작성합니다.</p>	<p>파라미터에 필요한 값을 입력합니다.</p> <ul style="list-style-type: none"> • <code>DestinationEmailAddress</code> - AWS KMS 키가 삭제될 예정인 경우 알림을 받을 이메일 주소입니다. • <code>SNSTopicName</code> - Amazon SNS 주제의 이름. • <code>EnableRemediation</code> - Systems Manager 런북을 사용하여 예약된 키 삭제를 취소합니다. 허용되는 값은 <code>true</code> 및 <code>false</code>입니다. • <code>AutomationAssumeRole</code> - Systems Manager 자동화가 사용자를 대신하여 작업을 수행하도록 허용하는 역할의 Amazon 리소스 이름 (ARN)입니다. 자세한 내용은 AWSConfigRemediationCancelKeyDeletion 설명서의 필수 IAM 권한 섹션을 참조하십시오. • <code>Capabilities</code> — AWS에서 CloudFormation 스택을 생성하려면 스택 템플릿에 특정 기능이 포함되어 있음을 명시적으로 인정해야 합니다. 	<p>개발자, 보안 엔지니어</p>

구독 확인

작업	설명	필요한 기술
구독을 확인합니다.	이메일 받은 편지함을 확인하고 Amazon SNS에서 받은 이메일 메시지에서 구독 확인을 선택합니다. 웹 브라우저 창이 열리고 구독 확인과 구독 ID가 표시됩니다.	개발자, 보안 엔지니어

관련 리소스

참조

- [AWS 서비스를 위한 규칙 생성](#)
- [삭제 대기 중인 AWS KMS 키의 사용을 탐지하기 위한 Amazon CloudWatch 경보 생성](#)

튜토리얼 및 동영상

- [아마존을 시작하는 방법 EventBridge](#)
- [아마존 심층 분석 EventBridge \(AWS 온라인 테크 토크\)](#)

AWS 워크숍

- [EventBridge 규칙 다루기](#)

추가 정보

다음 코드는 AWS 서비스의 변경 사항을 모니터링하고 알리도록 솔루션을 확장하는 예제를 제공합니다. 예제에는 사전 정의된 패턴과 사용자 지정 패턴이 포함됩니다. 자세한 내용은 [의 이벤트 및 이벤트 패턴](#)을 참조하십시오 EventBridge.

```
EventPattern:
  source:
    - aws.kms
```

```
detail-type:  
- AWS API Call via CloudTrail  
detail:  
  eventSource:  
  - kms.amazonaws.com  
  eventName:  
  - ScheduleKeyDeletion
```

첨부

이 문서와 관련된 추가 콘텐츠에 액세스하려면 [attachment.zip](#) 파일의 압축을 풉니다.

ABAC(속성 기반 액세스 제어)를 사용하여 퍼블릭 서브넷 보호

작성자: Joel Alfredo Nunez Gonzalez(AWS) 및 Samuel Ortega Sancho(AWS)

환경: PoC 또는 파일럿

기술: 보안, 자격 증명, 규정 준수, 네트워킹, 콘텐츠 전송

AWS 서비스: AWS Organizations, AWS Identity and Access Management

요약

중앙 집중식 네트워크 아키텍처에서 검사 및 엣지 Virtual Private Cloud(VPC)는 인터넷으로 들어오고 나가는 트래픽과 같은 모든 인바운드 및 아웃바운드 트래픽을 집중시킵니다. 하지만 이로 인해 병목 현상이 발생하거나 AWS service quotas 한도에 도달할 수 있습니다. VPC의 워크로드와 함께 네트워크 엣지 보안을 배포하면 일반적인 중앙 집중식 접근 방식에 비해 전례 없는 확장성을 제공합니다. 이를 분산 엣지 아키텍처라고 합니다.

워크로드 계정에 퍼블릭 서브넷을 배포하면 이점을 얻을 수 있지만 공격 표면이 증가하므로 새로운 보안 위험도 발생합니다. 이러한 VPC의 퍼블릭 서브넷에는 Application Load Balancer 또는 NAT 게이트웨이와 같은 Elastic Load Balancing(ELB) 리소스만 배포하는 것이 좋습니다. 전용 퍼블릭 서브넷에서 로드 밸런서와 NAT 게이트웨이를 사용하면 인바운드 및 아웃바운드 트래픽을 세밀하게 제어할 수 있습니다.

ABAC(속성 기반 액세스 제어)는 부서, 직무, 팀 이름 등의 사용자 속성을 기반으로 세분화된 권한을 생성하는 방식입니다. 자세한 내용은 [AWS용 ABAC](#)를 참조하십시오. ABAC는 워크로드 계정의 퍼블릭 서브넷을 위한 가드레일을 제공할 수 있습니다. 이를 통해 애플리케이션 팀은 인프라 보안을 손상시키지 않으면서 민첩성을 유지할 수 있습니다.

이 패턴은 AWS Organizations의 [서비스 제어 정책\(SCP\)](#) 및 AWS Identity and Access Management(IAM)의 [정책](#)을 통해 ABAC를 구현함으로써 퍼블릭 서브넷을 보호하는 방법을 설명합니다. SCP를 조직의 구성원 계정이나 조직 단위(OU)에 적용합니다. 이러한 ABAC 정책은 사용자가 대상 서브넷에 NAT 게이트웨이를 배포하도록 허용하고 EC2 인스턴스 및 탄력적 네트워크 인터페이스(ENI)와 같은 다른 Amazon Elastic Compute Cloud(Amazon EC2) 리소스를 배포하지 못하도록 합니다.

사전 조건 및 제한 사항

사전 조건

- AWS Organizations의 조직
- AWS Organizations 루트 계정에 대한 관리 액세스
- 조직에서 SCP 테스트를 위한 활성 구성원 계정 또는 OU

제한 사항

- 이 솔루션의 SCP는 서비스 연결 역할을 사용하는 AWS 서비스가 대상 서브넷에 리소스를 배포하는 것을 막지 않습니다. 이러한 서비스의 예로는 Elastic Load Balancing(ELB), Amazon Elastic Container Service(Amazon ECS) 및 Amazon Relational Database Service(RDS)가 있습니다. 자세한 내용은 AWS Organizations 설명서의 [SCP가 권한에 미치는 영향](#)을 참조하십시오. 보안 제어를 구현하여 이러한 예외를 탐지하십시오.

아키텍처

대상 기술 스택

- AWS Organizations의 AWS 계정 또는 OU에 SCP 적용
- IAM 역할은 다음과 같습니다.
 - AutomationAdminRole – SCP 구현 후 서브넷 태그를 수정하고 VPC 리소스를 생성하는 데 사용
 - TestAdminRole – SCP가 관리 액세스 권한이 있는 사용자를 비롯한 다른 IAM 보안 주체가 AutomationAdminRole 전용 작업을 수행하는 것을 막고 있는지 테스트하는 데 사용

대상 아키텍처

1. 대상 계정에서 AutomationAdminRole IAM 역할을 생성합니다. 이 역할에는 네트워킹 리소스를 관리할 권한이 있습니다. 이 역할에만 적용되는 다음 권한에 유의하십시오.
 - 이 역할은 VPC와 퍼블릭 서브넷을 생성할 수 있습니다.
 - 이 역할은 대상 서브넷의 태그 할당을 수정할 수 있습니다.
 - 이 역할은 자체 권한을 관리할 수 있습니다.
2. AWS Organizations에서는 대상 AWS 계정 또는 OU에 SCP를 적용합니다. 샘플 정책은 이 패턴의 [추가 정보](#)를 참조하십시오.

3. CI/CD 파이프라인의 사용자 또는 도구가 대상 서브넷에 SubnetType 태그를 적용하는 AutomationAdminRole 역할을 맡을 수 있습니다.
4. 조직의 승인된 IAM 보안 주체는 다른 IAM 역할을 맡아 대상 서브넷의 NAT 게이트웨이와 AWS 계정의 기타 허용된 네트워킹 리소스(예: 라우팅 테이블)를 관리할 수 있습니다. IAM 정책을 사용하여 이러한 권한을 부여할 수 있습니다. 자세한 내용은 [Amazon VPC의 자격 증명 및 액세스 관리](#)를 참조하십시오.

자동화 및 규모 조정

퍼블릭 서브넷을 보호하려면 해당 [AWS 태그](#)를 적용해야 합니다. SCP를 적용한 후 NAT 게이트웨이는 승인된 사용자가 SubnetType:IFA 태그가 있는 서브넷에서 생성할 수 있는 유일한 종류의 Amazon EC2 리소스입니다. (IFA는 인터넷 연결 자산을 의미합니다.) SCP는 인스턴스 및 ENI와 같은 다른 Amazon EC2 리소스의 생성을 방지합니다. VPC 리소스를 생성하는 AutomationAdminRole 역할을 맡는 CI/CD 파이프라인을 사용하여 이러한 태그가 퍼블릭 서브넷에 제대로 적용되도록 하는 것이 좋습니다.

도구

서비스

- [AWS Identity and Access Management\(IAM\)](#)를 사용하면 사용자에 대해 인증 및 권한 부여를 제어함으로써 AWS 리소스에 대한 액세스를 안전하게 관리할 수 있습니다.
- [AWS Organizations](#)는 사용자가 생성하고 중앙에서 관리하는 조직으로 여러 AWS 계정을 통합할 수 있는 계정 관리 서비스입니다. AWS Organizations에서 조직의 권한을 관리하는 데 사용할 수 있는 정책 유형인 [서비스 제어 정책\(SCP\)](#)을 구현할 수 있습니다.
- [Amazon Virtual Private Cloud\(VPC\)](#)를 이용하면 사용자가 정의한 가상 네트워크로 AWS 리소스를 시작할 수 있습니다. 이 가상 네트워크는 사용자의 자체 데이터 센터에서 운영하는 기존 네트워크와 유사하며 AWS의 확장 가능한 인프라를 사용한다는 이점이 있습니다.

에픽

SCP 적용

작업	설명	필요한 기술
테스트 관리자 역할을 생성합니다.	관리자 계정에서 TestAdmin Role 이라는 이름의 IAM 역할	AWS 관리자

작업	설명	필요한 기술
	<p>할을 생성합니다. AdministratorAccessAWS 관리형 IAM 정책을 새 역할에 연결합니다. 지침은 IAM 설명서의 IAM 사용자에게 권한을 위임하기 위한 역할 생성을 참조하십시오.</p>	

작업	설명	필요한 기술
<p>자동화 관리자 역할을 생성합니다.</p>	<ol style="list-style-type: none"> 관리자 계정에서 AutomationAdminRole 이라는 이름의 IAM 역할을 생성합니다. AdministratorAccessAWS 관리형 IAM 정책을 새 역할에 연결합니다. <p>다음은 00000000000000000000 계정에서 역할을 테스트하는 데 사용할 수 있는 신뢰 정책의 예제입니다.</p> <pre data-bbox="597 850 1024 1766"> { "Version": "2012-10-17", "Statement": [{ "Effect": "Allow", "Principal": { "AWS": ["arn:aws:iam::000000000000:root"] }, "Action": "sts:AssumeRole", "Condition": {} }] } </pre>	<p>AWS 관리자</p>

작업	설명	필요한 기술
SCP를 생성하고 연결합니다.	<ol style="list-style-type: none"> 추가 정보 섹션에 제공된 샘플 코드를 사용하여 보안 제어 정책을 생성합니다. 지침은 AWS Organizations 설명서의 SCP 생성을 참조하십시오. 대상 AWS 계정 또는 OU에 SCP를 연결합니다. 지침은 AWS Organizations 설명서의 서비스 제어 정책 연결 및 분리를 참조하세요. 	AWS 관리자

SCP 테스트

작업	설명	필요한 기술
VPC 또는 서브넷을 생성합니다.	<ol style="list-style-type: none"> 대상 AWS 계정에서 TestAdminRole 역할을 말합니다. 기존 VPC에 VPC 또는 새 퍼블릭 서브넷을 생성해 보십시오. 지침은 Amazon VPC 설명서에서 VPC, 서브넷 및 기타 VPC 리소스 생성을 참조하십시오. 이러한 리소스를 생성할 수 없어야 합니다. AutomationAdminRole 역할을 맡고 이전 단계를 다시 시도하십시오. 이제 네트워킹 리소스를 생성할 수 있을 것입니다. 	AWS 관리자

작업	설명	필요한 기술
태그를 관리합니다.	<ol style="list-style-type: none"> 1. 대상 AWS 계정에서 TestAdminRole 역할을 말합니다. 2. 사용 가능한 퍼블릭 서브넷에 SubnetType:IFA 태그를 추가합니다. 이 태그를 추가할 수 있어야 합니다. AWS Command Line Interface(AWS CLI)를 통해 태그를 추가하는 방법에 대한 지침은 AWS CLI 명령 참조의 태그 생성을 참조하십시오. 3. 보안 인증을 변경하지 말고 이 서브넷에 할당된 SubnetType:IFA 태그를 수정해 보십시오. 이 태그는 수정할 수 없어야 합니다. 4. AutomationAdminRole 역할을 맡고 이전 단계를 다시 시도하십시오. 이 역할은 이 태그를 추가하고 수정할 수 있어야 합니다. 	AWS 관리자

작업	설명	필요한 기술
<p>대상 서브넷에 리소스를 배포합니다.</p>	<ol style="list-style-type: none"> 1. TestAdminRole 역할을 말합니다. 2. SubnetType:IFA 태그가 있는 퍼블릭 서브넷의 경우 EC2 인스턴스를 생성해 보십시오. 지침은 Amazon EC2 설명서의 인스턴스 시작을 참조하세요. 이 서브넷에서는 NAT 게이트웨이를 제외한 Amazon EC2 리소스를 생성, 수정 또는 삭제할 수 없습니다. 3. 동일한 서브넷에서 NAT 게이트웨이를 생성합니다. 지침은 Amazon VPC 설명서의 NAT 게이트웨이 생성을 참조하십시오. 이 서브넷에서 NAT 게이트웨이를 생성, 수정 또는 삭제할 수 있어야 합니다. 	<p>AWS 관리자</p>
<p>AutomationAdminRole 역할을 관리합니다.</p>	<ol style="list-style-type: none"> 1. TestAdminRole 역할을 말합니다. 2. AutomationAdminRole 역할을 수정해 보십시오. 지침은 IAM 설명서의 역할 수정을 참조하십시오. 이 역할을 수정할 수 없어야 합니다. 3. AutomationAdminRole 역할을 맡고 이전 단계를 다시 시도하십시오. 이제 역할을 수정할 수 있을 것입니다. 	<p>AWS 관리자</p>

정리

작업	설명	필요한 기술
배포된 리소스를 정리합니다.	<ol style="list-style-type: none"> 1. AWS 계정 또는 OU에서 SCP를 분리합니다. 지침은 AWS Organizations 설명서의 SCP 분리를 참조하십시오. 2. SCP를 삭제합니다. 지침은 SCP 삭제(AWS Organizations 설명서)를 참조하십시오. 3. AutomationAdminRole 역할과 TestAdminRole 역할을 삭제합니다. 지침은 IAM 설명서의 역할 삭제를 참조하십시오. 4. 이 솔루션용으로 생성한 모든 네트워킹 리소스(예: VPC 및 서브넷)를 삭제합니다. 	AWS 관리자

관련 리소스

AWS 설명서

- [SCP 생성, 업데이트 및 삭제](#)
- [SCP 연결 및 분리](#)
- [서비스 승인 참조](#)
- [AWS용 ABAC란 무엇입니까?](#)
- [AWS 리소스에 태그 지정](#)
- [탐지 제어](#)

추가 AWS 참조

- [AWS Organizations의 서비스 제어 정책을 사용하여 권한 부여에 사용되는 리소스 태그 보호](#)(AWS 블로그 게시물)

추가 정보

다음 서비스 제어 정책은 조직에서 이 접근 방식을 테스트하는 데 사용할 수 있는 예제입니다.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "DenyVPCActions",
      "Effect": "Deny",
      "Action": [
        "ec2:CreateVPC",
        "ec2:CreateRoute",
        "ec2:CreateSubnet",
        "ec2:CreateInternetGateway",
        "ec2>DeleteVPC",
        "ec2>DeleteRoute",
        "ec2>DeleteSubnet",
        "ec2>DeleteInternetGateway"
      ],
      "Resource": [
        "arn:aws:ec2:*:*:*"
      ],
      "Condition": {
        "StringNotLike": {
          "aws:PrincipalARN": ["arn:aws:iam:*:*:role/AutomationAdminRole"]
        }
      }
    },
    {
      "Sid": "AllowNATGWOnIFASubnet",
      "Effect": "Deny",
      "NotAction": [
        "ec2:CreateNatGateway",
        "ec2>DeleteNatGateway"
      ],
      "Resource": [
        "arn:aws:ec2:*:*:subnet/*"
      ]
    }
  ]
}
```

```

    "Condition": {
      "ForAnyValue:StringEqualsIfExists": {
        "aws:ResourceTag/SubnetType": "IFA"
      },
      "StringNotLike": {
        "aws:PrincipalARN": ["arn:aws:iam::*:role/AutomationAdminRole"]
      }
    }
  },
  {
    "Sid": "DenyChangesToAdminRole",
    "Effect": "Deny",
    "NotAction": [
      "iam:GetContextKeysForPrincipalPolicy",
      "iam:GetRole",
      "iam:GetRolePolicy",
      "iam:ListAttachedRolePolicies",
      "iam:ListInstanceProfilesForRole",
      "iam:ListRolePolicies",
      "iam:ListRoleTags"
    ],
    "Resource": [
      "arn:aws:iam::*:role/AutomationAdminRole"
    ],
    "Condition": {
      "StringNotLike": {
        "aws:PrincipalARN": ["arn:aws:iam::*:role/AutomationAdminRole"]
      }
    }
  },
  {
    "Sid": "allowbydefault",
    "Effect": "Allow",
    "Action": "*",
    "Resource": "*"
  }
]
}

```

Security Hub를 사용하여 AWS Organizations의 퍼블릭 S3 버킷 식별

작성자: Mourad Cherfaoui(AWS), Arun Chandapillai(AWS) 및 Parag Nagwekar(AWS)

환경: 프로덕션

기술: 보안, 자격 증명, 규정 준수, 스토리지 및 백업

워크로드: 기타 모든 워크로드

AWS 서비스: 아마존 EventBridge, AWS 보안 허브, 아마존 SNS

요약

이 패턴은 AWS Organizations 계정에서 퍼블릭 Amazon Simple Storage Service(S3) 버킷을 식별하는 메커니즘을 구축하는 방법을 보여줍니다. 이 메커니즘은 AWS Security Hub의 [Foundational Security Best Practices\(FSBP\) 표준](#)의 제어를 사용하여 S3 버킷을 모니터링하는 방식으로 작동합니다. EventBridge Amazon을 사용하여 Security Hub [조사 결과를](#) 처리한 다음 이 결과를 Amazon Simple Notification Service (Amazon SNS) 주제에 게시할 수 있습니다. 조직의 이해 관계자가 주제를 구독하면 조사 결과에 대한 즉각적인 이메일 알림을 받을 수 있습니다.

기본적으로 새 S3 버킷 및 객체는 퍼블릭 액세스를 허용하지 않습니다. 조직의 요구 사항에 따라 기본 Amazon S3 구성을 수정해야 하는 시나리오에서 이 패턴을 사용할 수 있습니다. 예를 들어 인터넷상의 모든 사용자가 S3 버킷에서 읽을 수 있어야 하는 공개 웹사이트 또는 파일을 호스팅하는 S3 버킷이 있는 경우를 예로 들 수 있습니다.

Security Hub는 보안 표준 및 규정 준수 요구 사항과 관련된 조사 결과를 포함하여 모든 보안 결과를 통합하기 위한 중앙 서비스로 배포되는 경우가 가끔 있습니다. 퍼블릭 S3 버킷을 탐지하는 데 사용할 수 있는 다른 AWS 서비스도 있지만 이 패턴은 최소 구성의 기존 Security Hub 배포를 사용합니다.

사전 조건 및 제한 사항

사전 조건

- 전용 [Security Hub 관리자 계정](#)을 사용하는 AWS 다중 계정 설정
- Security Hub 및 AWS Config, 모니터링하려는 AWS 리전에서 활성화(참고: 단일 [집계 리전에서 여러 리전을 모니터링하려면 Security Hub에서 크로스 리전 집계 활성화](#)를 활성화해야 합니다.)

- Security Hub 관리자 계정에 액세스하고 업데이트하기 위한 사용자 권한, 조직 내 모든 S3 버킷에 대한 읽기 액세스, 퍼블릭 액세스를 해제하기 위한 권한(필요한 경우)

아키텍처

기술 스택

- AWS Security Hub
- 아마존 EventBridge
- Amazon Simple Notification Service(Amazon SNS)
- Amazon Simple Storage Service(S3)

대상 아키텍처

다음 다이어그램은 Security Hub를 사용하여 퍼블릭 S3 버킷을 식별하는 아키텍처를 보여줍니다.

이 다이어그램은 다음 워크플로를 보여 줍니다.

1. Security Hub는 FSBP 보안 표준의 S3.2 및 S3.3 제어를 사용하여 모든 AWS Organizations 계정(관리자 계정 포함)의 S3 버킷 구성을 모니터링하고 버킷이 퍼블릭으로 구성되어 있는지 여부를 감지합니다.
2. Security Hub 관리자 계정은 모든 구성원 계정의 조사 결과(S3.2 및 S3.3의 결과 포함)에 액세스합니다.
3. Security Hub는 모든 새 검색 결과 및 기존 검색 결과에 대한 모든 업데이트를 Security Hub 검색 결과 - 가져온 이벤트로 자동 전송합니다. EventBridge 여기에는 관리자 및 구성원 계정 모두의 조사 결과에 대한 이벤트가 포함됩니다.
4. EventBridge 규칙은 a ComplianceStatusFAILED, 워크플로 상태 및 a를 가진 S3.2 및 S3.3의 검색 결과를 기준으로 필터링합니다. NEW RecordState ACTIVE
5. 규칙은 이벤트 패턴을 사용하여 이벤트를 식별하고 일치하는 이벤트를 SNS 주제로 보냅니다.
6. SNS 주제는 구독자에게 이벤트를 전송합니다(예: 이메일 전송).
7. 이메일 알림을 수신하도록 지정된 보안 분석가가 해당 S3 버킷을 검토합니다.
8. 버킷의 퍼블릭 액세스가 승인되면 보안 분석가는 Security Hub에서 해당 조사 결과의 워크플로 상태를 SUPPRESSED로 설정합니다. 그렇지 않으면 분석가가 상태를 NOTIFIED로 설정합니다. 이렇게 하면 S3 버킷에 대한 향후 알림이 제거되고 알림 노이즈가 줄어듭니다.

9. 워크플로 상태가 NOTIFIED로 설정된 경우 보안 분석가는 버킷 소유자와 함께 조사 결과를 검토하여 퍼블릭 액세스가 정당하고 개인 정보 보호 및 데이터 보호 요구 사항을 준수하는지 확인합니다. 조사 결과 버킷에 대한 퍼블릭 액세스가 제거되거나 퍼블릭 액세스가 승인됩니다. 후자의 경우 보안 분석가는 워크플로 상태를 SUPPRESSED로 설정합니다.

참고: 아키텍처 다이어그램은 단일 리전 및 크로스 리전 집계 활성화 배포 모두에 적용됩니다. 다이어그램의 계정 A, B, C에서 Security Hub는 관리자 계정과 동일한 리전에 속하거나 크로스 리전 집계 활성화가 활성화된 경우 다른 리전에 속할 수 있습니다.

도구

도구

- [EventBridgeAmazon](#)은 다양한 소스의 실시간 데이터와 애플리케이션을 연결하는 데 도움이 되는 서버리스 이벤트 버스 서비스입니다. EventBridge 자체 애플리케이션, 서비스형 소프트웨어 (SaaS) 애플리케이션 및 AWS 서비스의 실시간 데이터 스트림을 제공합니다. EventBridge 데이터가 사용자 정의 규칙과 일치하는 경우 해당 데이터를 SNS 주제 및 AWS Lambda 함수와 같은 대상으로 라우팅합니다.
- [Amazon Simple Notification Service\(Amazon SNS\)](#)를 사용하면 웹 서버 및 이메일 주소를 포함하여 게시자와 클라이언트 간의 메시지 교환을 조정하고 관리할 수 있습니다. 구독자는 구독하는 주제에 게시된 모든 메시지를 수신하며 주제에 대한 모든 구독자는 동일한 메시지를 수신합니다.
- [Amazon Simple Storage Service\(S3\)](#)는 원하는 양의 데이터를 저장, 보호 및 검색하는 데 도움이 되는 클라우드 기반 객체 스토리지 서비스입니다.
- [AWS Security Hub](#)는 AWS의 보안 상태에 대한 포괄적인 보기를 제공합니다. Security Hub를 사용하면 사용자의 AWS 환경이 보안 업계 표준 및 모범 사례를 준수하는지 확인할 수도 있습니다. Security Hub는 AWS 계정, 서비스 및 지원되는 타사 파트너 제품 전반에 걸쳐 보안 데이터를 수집하여 보안 동향을 분석하고 우선순위가 가장 높은 보안 문제를 파악할 수 있도록 지원합니다.

에픽

Security Hub 계정 구성

작업	설명	필요한 기술
AWS Organizations 계정에서 Security Hub를 활성화합니다.	S3 버킷을 모니터링하려는 조직 계정에서 Security Hub를 활	AWS 관리자

작업	설명	필요한 기술
	성화하려면 AWS Security Hub 사용 설명서의 Designating a Security Hub 관리자 계정(콘솔) 및 조직에 속한 구성원 계정 관리 의 지침을 참조하십시오.	
(선택 사항) 크로스 리전 집계 활성화를 활성화합니다.	단일 리전의 여러 리전에 있는 S3 버킷을 모니터링하려면 크로스 리전 집계 활성화 를 설정하십시오.	AWS 관리자
FSBP 보안 표준에 대해 S3.2 및 S3.3 제어를 활성화합니다.	FSBP 보안 표준에 대해 S3.2 및 S3.3 제어를 활성화해야 합니다. <ol style="list-style-type: none"> S3.2 제어를 활성화하려면 AWS Security Hub 사용 설명서의 [S3.2] S3 버킷이 퍼블릭 읽기 액세스를 금지해야 한다는 지침을 따르십시오. S3.3 제어를 활성화하려면 AWS Security Hub 사용 설명서의 [3] S3 버킷이 퍼블릭 쓰기 액세스를 금지해야 한다는 지침을 따르십시오. 	AWS 관리자

환경 설정

작업	설명	필요한 기술
SNS 주제 및 이메일 구독을 구성합니다.	1. AWS Management Console에 로그인한 후 Amazon SNS 콘솔 을 엽니다.	AWS 관리자

작업	설명	필요한 기술
	<ol style="list-style-type: none"> 2. 탐색 창에서 토픽을 선택한 다음, 토픽 생성을 선택합니다. 3. 유형에서 표준을 선택합니다. 4. 이름에 주제 이름(예: public-s3-buckets)을 입력합니다. 5. 주제 생성을 선택합니다. 6. 주제의 구독 탭에서 구독 생성을 선택합니다. 7. 프로토콜에서 이메일을 선택합니다. 8. 엔드포인트에 알림을 수신하는 데 사용할 이메일 주소를 입력합니다. AWS 관리자, IT 전문가 또는 Infosec 전문가의 이메일 주소를 사용할 수 있습니다. 9. 구독 생성을 선택합니다. 추가 이메일 구독을 생성하려면 필요에 따라 단계 6~8을 반복합니다. 	

작업	설명	필요한 기술
규칙을 구성합니다. EventBridge	<ol style="list-style-type: none"> 1. EventBridge 콘솔을 엽니다. 2. 시작하기 섹션에서 규칙을 선택한 다음 EventBridge 규칙 생성을 선택합니다. 3. 규칙 세부 정보 정의 페이지에서 이름에 규칙 이름(예: public-s3-buckets)을 입력합니다. 다음을 선택합니다. 4. 이벤트 패턴 섹션에서 패턴 편집을 선택합니다. 5. 다음 코드를 복사하여 이벤트 패턴 코드 편집기에 붙여 넣고 다음을 선택합니다. <pre data-bbox="597 947 1027 1873"> { "source": ["aws.securityhub"], "detail-type": ["Security Hub Findings - Imported"], "detail": { "findings": { "Compliance": { "Status": ["FAILED"] }, "RecordState": ["ACTIVE"], "Workflow": { "Status": ["NEW"] }, "ProductFields": { "ControlId": ["S3.2", "S3.3"] } } } } </pre>	AWS 관리자

작업	설명	필요한 기술
	<pre data-bbox="594 205 1024 348"> } } } </pre> <p data-bbox="594 380 1008 464">그런 다음 다음과 같이 하십시오.</p> <ol data-bbox="594 506 1008 842" style="list-style-type: none"> 1. 대상 선택 페이지에서 대상 선택에 대해 SNS 주제를 대상으로 선택한 다음 이전에 만든 주제를 선택합니다. 2. 다음을 선택하고 다음을 다시 선택한 다음 규칙 생성을 선택합니다. 	

문제 해결

문제	Solution
<p>퍼블릭 액세스가 활성화된 S3 버킷이 있는데 이에 대한 이메일 알림을 받지 못합니다.</p>	<p>이는 버킷이 다른 리전에서 생성되었고 Security Hub 관리자 계정에서 크로스 리전 집계 활성화가 활성화되지 않았기 때문일 수 있습니다. 이 문제를 해결하려면 크로스 리전 집계 활성화를 활성화하거나 현재 S3 버킷이 있는 리전에 이 패턴의 솔루션을 구현하십시오.</p>

관련 리소스

- [AWS Security Hub란 무엇입니까?](#) (Security Hub 설명서)
- [AWS Foundational Security Best Practices\(FSBP\) 표준](#)(Security Hub 설명서)
- [AWS Security Hub 다중 계정 활성화 스크립트](#)(AWS 랩)
- [Amazon S3의 보안 모범 사례](#)(Amazon S3 설명서)

추가 정보

퍼블릭 S3 버킷을 모니터링하기 위한 워크플로

다음 워크플로는 조직의 퍼블릭 S3 버킷을 모니터링하는 방법을 보여줍니다. 워크플로는 이 패턴의 SNS 구성 주제 및 이메일 구독 스토리의 단계를 완료했다고 가정합니다.

1. S3 버킷이 퍼블릭 액세스로 구성되면 이메일 알림을 수신합니다.
 - 버킷의 퍼블릭 액세스가 승인되면 Security Hub 관리자 계정에서 해당 조사 결과의 워크플로 상태를 SUPPRESSED로 설정합니다. 이렇게 하면 Security Hub가 이 버킷에 대해 추가 알림을 발행하지 못하게 되며 중복 알림을 제거할 수 있습니다.
 - 버킷의 퍼블릭 액세스가 승인되지 않으면 Security Hub 관리자 계정에서 해당 조사 결과의 워크플로 상태를 NOTIFIED로 설정합니다. 이렇게 하면 Security Hub에서 이 버킷에 대해 Security Hub에서 추가 알림을 발행하지 않으므로 노이즈를 제거할 수 있습니다.
2. 버킷에 민감한 데이터가 포함되어 있을 수 있는 경우 검토가 완료될 때까지 퍼블릭 액세스를 즉시 끄십시오. 퍼블릭 액세스를 끄면 Security Hub는 워크플로 상태를 RESOLVED로 변경합니다. 그러면 버킷에 대한 이메일 알림이 중지됩니다.
3. 버킷을 퍼블릭으로 구성한 사용자 (예: AWS 사용 CloudTrail) 를 찾아 검토를 시작합니다. 검토 결과 버킷에 대한 퍼블릭 액세스가 제거되거나 퍼블릭 액세스가 승인됩니다. 퍼블릭 액세스가 승인되면 해당 결과의 워크플로 상태를 SUPPRESSED로 설정합니다.

Okta를 AWS IAM Identity Center와 통합하여 사용자, 역할 및 다중 계정 액세스 관리

작성자: Dhananjay Karanjkar(AWS), Shrikant Patil(AWS)

환경: 프로덕션

기술: 보안, 자격 증명, 규정 준수

워크로드: 기타 모든 워크로드

AWS 서비스: AWS IAM Identity Center

요약

AWS IAM Identity Center(AWS Single Sign-On 후속)를 사용하면 모든 AWS 계정과 클라우드 애플리케이션에 대한 AWS Single Sign-On(SSO) 액세스를 중앙에서 관리할 수 있습니다. Okta는 엔터프라이즈 사용자에게 모든 온프레미스 및 클라우드 리소스에 대한 SSO 환경을 제공할 수 있는 ID 관리 서비스입니다. IAM Identity Center는 도메인 간 ID 관리 시스템(SCIM) 2.0 프로토콜을 사용하여 Okta의 사용자 및 그룹 정보를 IAM Identity Center로 자동 프로비저닝(일명 동기화라고도 함)할 수 있도록 지원합니다. SCIM 동기화가 구성되면 Okta의 사용자 속성이 IAM Identity Center의 명명된 속성에 매핑됩니다. 이로 인해 IAM Identity Center와 Okta 간에 예상 속성이 일치하게 됩니다.

사전 조건 및 제한 사항

사전 조건

- 개발자 또는 라이선스를 받은 Okta 계정
- Okta의 사용자 및 권한을 관리하기 위한 관리 액세스
- 활성 상태의 AWS 계정
- IAM Identity Center에 대한 관리 액세스

아키텍처

대상 기술 스택

- IAM Identity Center

- Okta

대상 아키텍처

1. SCIM 프로토콜을 사용하면 Okta 사용자 및 그룹이 IAM Identity Center와 동기화됩니다.
2. 사용자는 Okta를 통해 IAM 아이덴티티 센터에 로그인합니다.
3. IAM Identity Center는 Security Assertion Markup Language(SAML) 토큰을 통해 역할을 맡습니다.
4. IAM Identity Center에서 사용자는 AWS Management Console을 시작합니다.

도구

서비스

- [AWS IAM Identity Center](#)를 사용하면 모든 AWS 계정 및 클라우드 애플리케이션에 대한 SSO 액세스를 중앙에서 관리할 수 있습니다.

기타 도구

- [Okta](#)는 엔터프라이즈 사용자에게 모든 온프레미스 및 클라우드 리소스에 대한 SSO 환경을 제공할 수 있는 ID 관리 서비스입니다.

에픽

Okta와 IAM Identity Center 연결

작업	설명	필요한 기술
Okta를 외부 자격 증명 공급자로 사용하도록 IAM Identity Center를 구성합니다.	<ol style="list-style-type: none"> 1. AWS Management Console에 로그인하여 IAM Identity Center 콘솔을 엽니다. 2. 탐색 창에서 설정을 선택합니다. 3. 설정 페이지에서 자격 증명 소스 탭을 선택합니다. 	AWS 관리자

작업	설명	필요한 기술
	<ol style="list-style-type: none"> 4. 작업 메뉴에서 ID 소스 변경을 선택합니다. 5. ID 소스 선택에서 외부 ID 공급자를 선택하고 다음을 선택합니다. 6. 서비스 공급자 메타데이터에서 다음 값을 기록해 둡니다. <ul style="list-style-type: none"> • IAM Identity Center Assertion Consumer Service(ACS) URL • IAM Identity Center 발급자 URL <p>이 값은 나중에 이 패턴에서 사용합니다. 이 페이지를 벗어나려고 하지 마세요.</p> 	
IAM Identity Center Okta 애플리케이션을 생성합니다.	<ol style="list-style-type: none"> 1. 새 브라우저 탭을 열고 Okta 콘솔에 로그인합니다. 2. 탐색 창에서 애플리케이션을 선택합니다. 3. 애플리케이션 페이지에서 앱 카탈로그 찾아보기를 선택합니다. 4. AWS IAM Identity Center를 찾습니다. 5. 통합 추가를 선택한 다음 완료 선택합니다. 이렇게 하면 IAM Identity Center 애플리케이션이 Okta에 추가됩니다. 	Okta 관리자

작업	설명	필요한 기술
Okta에서 IAM Identity Center 애플리케이션을 구성합니다.	<ol style="list-style-type: none"> 1. Okta 콘솔에서 로그인 탭을 선택합니다. 2. 편집을 선택합니다. 이렇게 하면 SAML 설정을 편집할 수 있습니다. 3. 고급 로그인 설정에서 IAM Identity Center 콘솔의 다음 값을 복사하여 붙여넣습니다. <ul style="list-style-type: none"> • IAM Identity Center ACS URL • IAM Identity Center 발급자 URL 4. 저장을 선택합니다. 	AWS 관리자, Okta 관리자

작업	설명	필요한 기술
Okta에서 IAM Identity Center 로 SAML 메타데이터를 업로드합니다.	<ol style="list-style-type: none"> 1. Okta 콘솔에서 로그인 탭을 선택합니다. 2. SAML 서명 인증서에서 작업을 선택한 다음 IdP 메타데이터 보기를 선택합니다. 3. 파일을 idp-saml.xml (으)로 저장합니다. 4. IAM ID 센터 콘솔에서 ID 소스 변경 페이지에 있는지 확인합니다. 5. ID 제공자 메타데이터에서 IdP SAML 메타데이터에 대해 파일 선택을 선택합니다. 6. idp-saml.xml 파일을 업로드한 후 다음을 선택합니다. 7. 변경 사항을 검토하고 확인합니다. 	AWS 관리자, Okta 관리자

Okta에서 SCIM 2.0 테스트 앱(OAuth Bearer 토큰) 애플리케이션을 구성합니다.

작업	설명	필요한 기술
SCIM을 활성화하여 사용자와 그룹을 동기합니다.	<ol style="list-style-type: none"> 1. IAM Identity Center 콘솔의 탐색 창에서 설정을 선택합니다. 2. 자동 프로비저닝에서 활성화를 선택합니다. 3. SCIM 엔드포인트와 액세스 토큰을 텍스트 편집기에 복사합니다. 이 값은 나중에 이 패턴에서 사용합니다. 	AWS 관리자

작업	설명	필요한 기술
<p>사용자 및 그룹을 IAM ID 센터로 푸시하는 SCIM 애플리케이션을 생성합니다.</p>	<ol style="list-style-type: none"> 1. Okta 콘솔의 애플리케이션 페이지에서 앱 카탈로그 찾아보기를 선택하세요. 2. SCIM 2.0 Test App (OAuth Bearer Token)를 찾습니다. 3. SCIM 2.0 테스트 앱(OAuth Bearer 토큰)을 선택한 다음 통합 추가를 선택합니다. 4. 일반 설정 페이지에서 다음 작업을 수행하세요. <ul style="list-style-type: none"> • 애플리케이션 레이블에 SCIM 2.0 Test App (OAuth Bearer Token)을 입력합니다. • 사용자에게 애플리케이션 아이콘을 표시하지 않음을 선택합니다. • Okta 모바일 앱에서 애플리케이션 아이콘을 표시하지 않음을 선택합니다. • 다음을 선택한 다음 완료를 선택합니다. 5. 프로비저닝 탭에서 API 통합 구성을 선택합니다. 6. API 통합 활성화를 선택합니다. 7. SCIM 2.0 기본 URL의 경우 이전에 복사한 SCIM 엔드포인트를 입력합니다. 	<p>AWS 관리자, Okta 관리자</p>

작업	설명	필요한 기술
	<p>8. OAuth Bearer 토큰에 이전에 복사한 액세스 토큰을 입력합니다.</p> <p>9. 저장을 선택한 다음 편집을 선택합니다.</p> <p>10. 사용자 생성 활성화를 선택한 다음 저장을 선택합니다.</p>	

작업	설명	필요한 기술
<p>사용자와 그룹을 IAM Identity Center로 푸시하는 규칙을 생성합니다.</p>	<ol style="list-style-type: none"> 1. Okta 콘솔에서 푸시 그룹 탭을 선택합니다. 2. 푸시 그룹 메뉴에서 규칙별 그룹 찾기를 선택합니다. 3. AWS SSO rule 규칙 이름을 지정합니다. 4. 로 시작하는 그룹 이름의 경우 awssso를 입력합니다. 아무 접두사나 사용할 수 있습니다. 5. 이 규칙으로 검색된 그룹을 즉시 푸시하기를 선택한 다음 규칙 생성을 선택합니다. 6. 탐색 창에서 기본 > 그룹을 선택합니다. 7. Add Group(그룹 추가)을 선택합니다. 8. 그룹 추가 대화 상자의 이름에 AWS Users를 입력한 다음 저장을 선택합니다. 9. 그룹 페이지에서 AWS 사용자 그룹을 선택합니다. 10. 애플리케이션 탭에서 애플리케이션 할당을 선택합니다. AWS IAM Identity Center 애플리케이션을 선택한 다음 할당을 선택합니다. 사용자가 AWS Management Console에서 실행할 애플리케이션입니다. 	<p>Okta 관리자</p>

작업	설명	필요한 기술
	<p>11.SCIM 2.0 테스트 앱(OAuth Bearer 토큰) 애플리케이션의 경우 할당을 선택합니다. 저장하고 돌아가기(Save and Go Back)를 선택합니다. 사용자는 이 애플리케이션과 상호 작용하지 않지만 이 애플리케이션은 사용자의 계정과 그룹이 IAM Identity Center에 프로비저닝되도록 합니다.</p> <p>12.완료를 선택합니다.</p>	

Okta 사용자 및 그룹 생성 및 매핑

작업	설명	필요한 기술
Okta에서 새 그룹을 생성합니다.	<ol style="list-style-type: none"> 1. 탐색 창의 Okta 콘솔에서 디렉터리 > 그룹을 선택합니다. 2. Add Group(그룹 추가)을 선택합니다. 3. 그룹 추가 대화 상자의 이름에 awsssoPowerUsers 를 입력한 다음 저장을 선택합니다. 4. 상단 탐색 메뉴에서 애플리케이션을 선택합니다. 5. SCIM 2.0 테스트 앱(OAuth Bearer 토큰) 애플리케이션을 연 다음 푸시 그룹을 선택합니다. 목록에 awsssoPowerUsers 표시되고 활성으로 	AWS 관리자, Okta 관리자

작업	설명	필요한 기술
	<p>표시된 것을 볼 수 있을 것입니다.</p> <p>6. IAM Identity Center 콘솔의 탐색 창에서 그룹을 선택합니다. 사용자 없음으로 표시된 <code>awsssoPowerUsers</code>가 보일 것입니다.</p>	
IAM Identity Center에서 권한 세트를 할당합니다.	<ol style="list-style-type: none"> 1. IAM Identity Center 콘솔의 탐색 창에서 AWS 계정을 선택한 다음 조직을 선택합니다. 2. 조직의 모든 계정을 선택합니다. 운영팀에 속한 사용자는 모든 계정에 액세스할 수 있습니다. 3. 사용자 또는 그룹 할당을 선택합니다. 4. 사용자 및 그룹 할당 페이지에서 그룹을 선택합니다. 5. 선택한 <code>awsssoPowerUsers</code> 후 다음을 선택합니다. 6. 권한 집합 선택 페이지에서 <code>AWSPowerUsersAccess</code>를 선택합니다. 7. 마침을 클릭합니다. 8. Proceed to AWS accounts(AWS 계정으로 진행)를 선택합니다. 	AWS 관리자

작업	설명	필요한 기술
Okta 포털에서 사용자를 생성합니다.	<ol style="list-style-type: none"> 1. Okta 콘솔의 상단 내비게이션 바에서 디렉토리를 선택한 다음 사람을 선택합니다. 2. 사람 추가(Add Person)를 선택합니다. 3. 인원 추가 페이지에서 다음 정보를 입력합니다. <ul style="list-style-type: none"> • 사용자 유형에서 사용자를 선택합니다. • 이름에는 사용자의 이름을 입력합니다. • 성에 사용자의 성을 입력합니다. • 사용자 이름에서 사용자의 이메일 주소를 입력합니다. • 기본 이메일의 경우 사용자의 이메일 주소를 입력합니다. • 그룹에는 awsssoPowerUsers 및 AWS 사용자를 입력합니다. • 암호에서 관리자별 설정을 선택한 다음 암호를 입력합니다. 4. 사용자가 처음 로그인할 때 비밀번호를 변경해야 함을 선택 취소한 다음 저장을 선택합니다. 5. IAM Identity Center 콘솔의 탐색 창에서 사용자를 선택 	AWS 관리자, Okta 관리자

작업	설명	필요한 기술
	<p>합니다. 목록에 생성한 사용자가 표시되어야 합니다.</p> <p>6. 사용자 이름을 선택하여 일반 정보를 엽니다.</p> <p>7. 사용자가 작성자: SCIM으로 표시되는지 확인합니다.</p>	

통합 검증

작업	설명	필요한 기술
새 사용자가 인증되었고 액세스 권한이 있는지 검증합니다.	<ol style="list-style-type: none"> 1. IAM Identity Center 콘솔의 탐색 창에서 대시보드를 선택합니다. 2. 사용자 포털 URL을 복사합니다. 3. 브라우저 창을 비공개 또는 시크릿 모드로 연 다음 사용자 포털 URL을 주소 표시줄에 붙여넣습니다. 브라우저가 Okta 로그인 페이지로 리디렉션되어야 합니다. 4. 다음 정보를 입력합니다. <ul style="list-style-type: none"> • 사용자이름의 경우 사용자의 이메일 주소를 입력합니다. • 비밀번호에는 이전에 만든 사용자 비밀번호를 입력합니다. 5. Okta 구성을 기반으로 추가 보안 질문을 입력합니다. 로그인하면 IAM Identity 	AWS 관리자

작업	설명	필요한 기술
	Center 콘솔로 돌아와야 합니다.	
새 사용자에게 AWS 계정에 대한 액세스 권한이 있는지 검증합니다.	<ol style="list-style-type: none"> 1. IAM Identity Center 콘솔에서 AWS 계정을 선택합니다. 2. 조직의 관리 계정을 선택합니다. 3. AWSPowerUsersAccess라인에서 관리 콘솔을 선택합니다. 4. AWS Management Console이 시작되는지 확인합니다. 	AWS 관리자

관련 리소스

AWS 설명서

- [자동 프로비저닝](#)(IAM Identity Center 설명서)
- [외부 ID 공급자에 연결](#)(IAM Identity Center 설명서)

AWS Marketplace

- [Okta 자격 증명 플랫폼](#)

Okta 리소스

- [Okta 콘솔](#)

AWS를 사용하여 AWS IAM ID 센터 권한 세트를 코드로 관리합니다. CodePipeline

작성자: Andre Cavalcante(AWS), Claison Amorim(AWS)

[코드 리포지토리: - 파이프라인 aws-iam-identity-center](#)

환경: 프로덕션

기술: 보안, ID, 규정 준수 DevOps

AWS 서비스: AWS, AWS CodeBuild, AWS CodeCommit CodePipeline, AWS IAM 아이덴티티 센터

요약

AWS IAM Identity Center(AWS Single Sign-On 후속)를 사용하면 모든 AWS 계정과 애플리케이션에 대한 AWS Single Sign-On(SSO) 액세스를 중앙에서 관리할 수 있습니다. IAM Identity Center에서 사용자 자격 증명을 생성하고 관리하거나 Microsoft Active Directory 도메인 또는 외부 ID 제공업체(idP)와 같은 기존 자격 증명 소스를 연결할 수 있습니다. IAM Identity Center는 [권한 세트](#)를 사용하여 AWS 환경에 대한 세분화된 액세스를 정의, 사용자 지정 및 할당할 수 있는 통합 관리 환경을 제공합니다. 권한 세트는 AWS IAM Identity Center ID 스토어 또는 외부 IdP의 페더레이션 사용자 및 그룹에 적용됩니다.

이 패턴은 AWS Organizations에서 조직으로 관리되는 다중 계정 환경에서 IAM Identity Center 권한 세트를 코드로 관리하는 데 도움이 됩니다. 이 패턴을 사용하면 다음을 달성할 수 있습니다.

- 권한 집합을 만들고, 삭제하고, 업데이트
- 대상 AWS 계정, 조직 단위(OU) 또는 조직 루트에 대한 권한 집합 할당을 생성, 업데이트 또는 삭제합니다.

IAM ID 센터 권한 및 할당을 코드로 관리하기 위해 이 솔루션은 AWS, CodeCommit AWS 및 AWS를 사용하는 지속적 통합 및 지속적 전송 (CI/CD) 파이프라인을 배포합니다. CodeBuild CodePipeline 리포지토리에 저장하는 JSON 템플릿에서 권한 세트와 할당을 관리합니다. CodeCommit Amazon EventBridge 규칙이 리포지토리 변경을 감지하거나 대상 OU의 계정 수정을 감지하면 AWS Lambda

함수를 시작합니다. Lambda 함수는 IAM Identity Center의 권한 집합과 할당을 업데이트하는 CI/CD 파이프라인을 시작합니다.

사전 조건 및 제한 사항

사전 조건

- AWS Organizations에서 하나의 조직으로 관리되는 다중 계정 환경. 자세한 내용은 [조직 생성](#)을 참조하세요.
- 자격 증명 소스로 활성화 및 구성된 IAM Identity Center. 자세한 내용은 IAM Identity Center 설명서의 [시작하기](#)를 참조하세요.
- IAM Identity Center의 위임 관리자로 등록된 멤버 계정. 지침은 IAM Identity Center 설명서의 [멤버 계정 등록](#)을 참조하세요.
- IAM Identity Center 위임 관리자 계정 및 조직의 관리 계정에 AWS CloudFormation 스택을 배포할 수 있는 권한. 자세한 내용은 설명서의 [액세스 제어](#)를 참조하십시오. CloudFormation
- Identity Center의 Amazon Simple Storage Service(S3) 버킷은 관리자에게 아티팩트 코드를 업로드 하도록 위임했습니다. 지침은 [버킷 생성](#)을 참조하세요.
- 조직의 관리 계정의 계정 ID. 지침은 [AWS 계정 ID 찾기](#)를 참조하세요.

제한 사항

- 이 패턴은 단일 계정 환경 또는 AWS Organizations에서 조직으로 관리되지 않는 계정에 대한 권한 세트를 관리하거나 할당하는 데 사용할 수 없습니다.
- 권한 세트 이름, 할당 ID, IAM Identity Center 보안 주체 유형 및 ID는 배포 후 수정할 수 없습니다.
- 이 패턴은 [사용자 지정 권한](#)을 생성하고 관리하는 데 도움이 됩니다. 이 패턴을 사용하여 [미리 정의된 권한](#)을 관리하거나 할당할 수 없습니다.
- 이 패턴은 조직의 관리 계정에 대한 권한 집합을 관리하는 데 사용할 수 없습니다.

아키텍처

기술 스택

- AWS CodeBuild
- AWS CodeCommit
- AWS CodePipeline

- 아마존 EventBridge
- AWS Identity Center
- AWS Lambda
- AWS Organizations

대상 아키텍처

이 다이어그램은 다음 워크플로를 보여줍니다.

1. 사용자는 다음 중 하나를 변경합니다.
 - a. 리포지토리에 하나 이상의 변경 사항을 커밋합니다. CodeCommit
 - b. AWS Organizations의 조직 단위(OU)에서 계정을 수정
2. 사용자가 CodeCommit 리포지토리에 변경 사항을 커밋하면 CodeChange EventBridge 규칙이 변경을 감지하고 IAM Identity Center의 위임 관리자 계정에서 Lambda 함수를 시작합니다. 규칙은 스토어에 있는 특정 파일(예: README.md 파일)의 변경에는 반응하지 않습니다.

사용자가 조직 단위의 계정을 수정한 경우 MoveAccount EventBridge 규칙은 변경을 감지하고 조직의 관리 계정에서 Lambda 함수를 시작합니다.

3. 시작된 Lambda 함수는 CI/CD 파이프라인을 시작합니다. CodePipeline
4. CodePipeline CodebuildTemplateValidation CodeBuild 프로젝트를 시작합니다.
5. CodebuildTemplateValidation CodeBuild 프로젝트는 CodeCommit 저장소의 Python 스크립트를 사용하여 권한 집합 템플릿의 유효성을 검사합니다. CodeBuild 다음을 검증합니다.
 - 권한 집합 이름은 고유합니다.
 - 할당문 ID(Sid)는 고유합니다.
 - CustomPolicy 파라미터의 정책 정의가 유효합니다. (이 확인에는 AWS Identity and Access Management Access Analyzer를 사용합니다)
 - 관리형 정책의 Amazon 리소스 이름(ARN)이 유효합니다.
6. 이 CodebuildPermissionSet CodeBuild 프로젝트는 Python용 AWS SDK (Boto3) 를 사용하여 IAM ID 센터의 권한 세트를 삭제, 생성 또는 업데이트합니다. SSOPipeline:true 태그가 있는 권한 집합만 영향을 받습니다. 이 파이프라인을 통해 관리되는 모든 권한 집합에는 이 태그가 있습니다.

7. CodebuildAssignments CodeBuild 프로젝트는 Terraform을 사용하여 IAM ID 센터에서 할당을 삭제, 생성 또는 업데이트합니다. Terraform 백엔드 상태 파일은 동일한 계정의 S3 버킷에 저장됩니다.
8. CodeBuild 조직의 관리 계정에서 lookup IAM 역할을 말합니다. 권한을 부여하거나 취소하는 데 필요한 리소스를 나열하기 위해 조직 및 [identitystore](#) API를 직접적으로 호출합니다.
9. CodeBuild IAM ID 센터의 권한 집합 및 할당을 업데이트합니다.

자동화 및 규모 조정

다중 계정 환경의 모든 새 계정은 AWS Organizations의 특정 조직 단위로 이동되므로 이 솔루션이 자동으로 실행되어 할당 템플릿에서 지정한 모든 계정에 필요한 권한 집합을 부여합니다. 추가 자동화 또는 규모 조정 작업이 필요하지 않습니다.

대규모 환경에서는 IAM Identity Center에 대한 API 요청 수가 많기 때문에 이 솔루션이 더 느리게 실행될 수 있습니다. Terraform과 Boto3는 자동으로 제한을 관리하여 성능 저하를 최소화합니다.

도구

서비스

- [AWS는 AWS CloudFormation](#) 리소스를 설정하고, 빠르고 일관되게 프로비저닝하고, AWS 계정 및 지역 전반의 수명 주기 전반에 걸쳐 리소스를 관리할 수 있도록 지원합니다.
- [CodeBuildAWS](#)는 소스 코드를 컴파일하고, 단위 테스트를 실행하고, 배포할 준비가 된 아티팩트를 생성하는 데 도움이 되는 완전관리형 빌드 서비스입니다.
- [CodeCommitAWS](#)는 자체 소스 제어 시스템을 관리할 필요 없이 Git 리포지토리를 비공개로 저장하고 관리할 수 있는 버전 제어 서비스입니다.
- [AWS](#)는 소프트웨어 릴리스의 여러 단계를 신속하게 모델링 및 구성하고 소프트웨어 변경 사항을 지속적으로 릴리스하는 CodePipeline 데 필요한 단계를 자동화하도록 지원합니다.
- [EventBridgeAmazon](#)은 다양한 소스의 실시간 데이터와 애플리케이션을 연결하는 데 도움이 되는 서버리스 이벤트 버스 서비스입니다. AWS Lambda 함수, API 대상을 사용하는 HTTP 간접 호출 엔드포인트 또는 다른 AWS 계정의 이벤트 버스를 예로 들 수 있습니다.
- [AWS IAM Identity Center](#)를 사용하면 모든 AWS 계정과 클라우드 애플리케이션에 대한 AWS Single Sign-On(SSO) 액세스를 중앙에서 관리할 수 있습니다.
- [AWS Organizations](#)는 사용자가 생성하고 중앙에서 관리하는 조직으로 여러 AWS 계정을 통합할 수 있는 계정 관리 서비스입니다.

- [AWS SDK for Python\(Boto3\)](#)는 Python 애플리케이션, 라이브러리 또는 스크립트를 AWS 서비스와 통합하는 데 도움이 되는 소프트웨어 개발 키트입니다.
- [Amazon Simple Storage Service\(S3\)](#)는 원하는 양의 데이터를 저장, 보호 및 검색하는 데 도움이 되는 클라우드 기반 객체 스토리지 서비스입니다.

코드 리포지토리

이 패턴의 코드는 [aws-iam-identity-center-pipeline](#) 리포지토리에서 사용할 수 있습니다. 리포지토리의 템플릿 폴더에는 권한 집합과 할당에 대한 샘플 템플릿이 포함되어 있습니다. 또한 대상 계정에 CI/CD 파이프라인 및 AWS 리소스를 배포하기 위한 AWS CloudFormation 템플릿도 포함되어 있습니다.

모범 사례

- 권한 집합과 할당 템플릿을 수정하기 전에 조직의 권한 세트를 계획하는 것이 좋습니다. 어떤 권한이 있어야 하는지, 권한 집합을 어떤 계정 또는 OU에 적용해야 하는지, 권한 집합의 영향을 받아야 하는 IAM Identity Center 보안 주체(사용자 또는 그룹)를 고려하세요. 권한 집합 이름, 연결 ID, IAM Identity Center 보안 주체 유형 및 ID는 배포 후 수정할 수 없습니다.
- 최소 권한 원칙을 준수하고 작업을 수행하는 데 필요한 최소 권한을 부여하세요. 자세한 내용은 IAM 설명서의 [최소 권한 부여](#) 및 [보안 모범 사례](#)를 참조하세요.

에픽

권한 설정 및 할당 계획

작업	설명	필요한 기술
리포지토리를 복제합니다.	<p>bash 셸에서 다음 명령을 입력합니다. 이렇게 하면 -pipeline aws-iam-identity-center 리포지토리가 복제됩니다. GitHub</p> <pre>git clone https://github.com/aws-samples/aws-iam-identity-center-pipeline.git</pre>	DevOps 엔지니어

작업	설명	필요한 기술
권한 세트를 정의합니다.	<ol style="list-style-type: none"> 복제된 리포지토리에서 <code>templates/permissions</code> 폴더로 이동한 다음 사용 가능한 템플릿 중 하나를 엽니다. Name 파라미터에 권한 집합의 이름을 입력합니다. 이 값은 고유해야 하며 배포 후에는 변경할 수 없습니다. Description 파라미터에서 사용 사례와 같은 권한 집합에 대한 간략한 설명을 추가하세요. SessionDuration 파라미터에 사용자가 AWS 계정에 로그인할 수 있는 시간을 지정합니다. ISO-8601 기간 형식(Wikipedia)을 사용하세요. (예: PT4H는 4시간) 정의된 값이 없는 경우 IAM Identity Center의 기본값은 1시간입니다. 권한 집합의 정책을 사용자 지정합니다. 다음 파라미터는 모두 선택 사항이며 배포 후 수정할 수 있습니다. 권한 집합의 정책을 정의하려면 다음 파라미터 중 하나 이상을 사용해야 합니다. <ul style="list-style-type: none"> ManagedPolicies 파라미터에 할당하려는 AWS 관리형 정책의 ARN을 입력합니다. 	DevOps 엔지니어

작업	설명	필요한 기술
	<ul style="list-style-type: none"> • CustomerManagedPolicies 파라미터에 할당하려는 고객 관리형 정책의 이름을 입력합니다. ARN을 사용하지 마세요. • PermissionBoundary 파라미터에서 다음을 수행하여 권한 경계를 할당하세요. • AWS 관리형 정책을 권한 경계로 사용하는 경우 PolicyType 에서는 AWS를 입력하고, Policy에서는 해당 정책의 ARN을 입력합니다. • 고객 관리형 정책을 권한 경계로 사용하는 경우 PolicyType 에서는 Customer를 입력하고, Policy에서는 해당 정책의 이름을 입력합니다. ARN을 사용하지 마세요. • CustomPolicy 파라미터에서 할당하려는 JSON 형식의 사용자 지정 정책을 정의합니다. JSON 정책 문서의 구조 및 내용에 대한 자세한 정보는 JSON 정책 개요를 참조하세요. <p>6. 권한 집합 템플릿을 저장하고 닫습니다. 권한 집합의 이</p>	

작업	설명	필요한 기술
	<p>름과 동일한 이름으로 파일을 저장하는 것이 좋습니다.</p> <p>7. 이 프로세스를 반복하여 조직에 필요한 수만큼 권한 집합을 만들고 필요하지 않은 샘플 템플릿은 삭제하세요.</p>	

작업	설명	필요한 기술
할당을 정의하세요.	<ol style="list-style-type: none"> 복제된 리포지토리에서 <code>templates/assignments</code> 폴더로 이동한 다음 <code>iam-identitycenter-assignments.json</code> 을 엽니다. 이 파일은 AWS 계정 또는 OU에 권한 세트를 할당하는 방법을 설명합니다. SID 파라미터에 할당을 위한 식별자를 입력합니다. 이 값은 고유해야 하며 배포 후에는 수정할 수 없습니다. Target 파라미터에서 권한 집합을 적용할 계정 또는 조직을 정의합니다. 유효한 값은 계정 ID, OU ID, OU 이름 또는 <code>root</code>입니다. <code>root</code> 관리 계정을 제외한 조직의 모든 멤버 계정에 사용 권한 세트를 할당합니다. 값을 큰따옴표로 입력하고 여러 값은 쉼표로 구분합니다. ID를 찾는 방법에 대한 지침은 계정 세부 정보 보기 또는 OU 세부 정보 보기를 참조하세요. PrincipalType 파라미터에 권한 설정의 영향을 받을 IAM Identity Center 보안 주체의 유형을 입력합니다. 유효한 값은 USER 또는 GROUP입니다. 이 값은 배포 후 수정할 수 없습니다. 	DevOps 엔지니어

작업	설명	필요한 기술
	<ol style="list-style-type: none"> 5. PrincipalID 파라미터에 권한 설정의 영향을 받을 IAM Identity Center ID 스토어의 사용자 또는 그룹 이름을 입력합니다. 이 값은 배포 후 수정할 수 없습니다. 6. PermissionSetName 파라미터에 할당하려는 권한 집합의 이름을 입력합니다. 7. 2~6단계를 반복하여 이 파일에 필요한 만큼 할당을 생성합니다. 일반적으로 각 권한 집합에는 하나의 할당이 있습니다. 필요하지 않은 샘플 할당은 모두 삭제하세요. 8. iam-identitycenter-assigments.json 파일을 저장하고 닫습니다. 	

권한 집합 및 할당 배포

작업	설명	필요한 기술
<p>파일을 S3 버킷에 업로드합니다.</p>	<ol style="list-style-type: none"> 1. 복제된 리포지토리를 .zip 파일로 압축합니다. 2. IAM Identity Center 위임 관리자 계정으로 로그인합니다. 3. https://console.aws.amazon.com/s3/ 에서 Amazon S3 콘솔을 엽니다. 4. 왼쪽 탐색 창에서 버킷을 선택합니다. 	<p>DevOps 엔지니어</p>

작업	설명	필요한 기술
	5. 이 솔루션을 배포하는 데 사용할 버킷을 선택합니다. 6. .zip 파일을 S3 버킷에 업로드합니다. 지침은 객체 업로드 를 참조하세요.	
IAM Identity Center 위임 관리자 계정에 리소스를 배포하세요.	1. IAM 아이덴티티 센터 위임 관리자 계정에서 https://console.aws.amazon.com/cloudformation/ 에서 CloudFormation 콘솔을 엽니다. 2. iam-identitycenter-pipeline.yaml 템플릿을 배포합니다. 스택에 명확하고 설명이 포함된 이름을 지정하고, 지침에 따라 파라미터를 업데이트하세요. 지침은 설명서의 스택 생성 을 참조하십시오. CloudFormation	DevOps 엔지니어

작업	설명	필요한 기술
<p>AWS Organizations 관리 계정에 리소스를 배포합니다.</p>	<ol style="list-style-type: none"> 1. 조직의 관리 계정에 로그인합니다. 2. https://console.aws.amazon.com/cloudformation/ 에서 CloudFormation 콘솔을 엽니다. 3. 탐색 모음에서 현재 표시된 AWS 리전의 이름을 선택합니다. us-east-1 리전을 선택합니다. 이 지역은 MoveAccount EventBridge 규칙이 조직 변경과 관련된 AWS CloudTrail 이벤트를 감지할 수 있도록 하기 위해 필요합니다. 4. iam-identitycenter-organization 템플릿을 배포합니다. 스택에 명확하고 설명이 포함된 이름을 지정하고, 지침에 따라 파라미터를 업데이트하세요. 지침은 설명서의 스택 생성을 CloudFormation 참조하십시오. 	<p>DevOps 엔지니어</p>

권한 집합 및 할당 업데이트

작업	설명	필요한 기술
<p>권한 집합 및 할당을 업데이트하세요.</p>	<p>MoveAccount Amazon EventBridge 규칙이 조직 내 계정에 대한 수정 사항을 감지하면 CI/CD 파이프라인이 자동</p>	<p>DevOps 엔지니어</p>

작업	설명	필요한 기술
	<p>으로 시작하여 권한 집합을 업데이트합니다. 예를 들어, 할당 JSON 파일에 지정된 OU에 계정을 추가하면 CI/CD 파이프라인은 권한 세트를 새 계정에 적용합니다.</p> <p>배포된 권한 집합과 할당을 수정하려면 JSON 파일을 업데이트한 다음 IAM Identity Center 위임 관리자 계정의 CodeCommit 리포지토리에 커밋하십시오. 지침은 설명서의 커밋 생성을 참조하십시오. CodeCommit</p> <p>CI/CD 파이프라인을 사용하여 이전에 배포한 권한 집합과 연결을 관리할 때는 다음 사항에 유의하세요.</p> <ul style="list-style-type: none"> • 권한 집합의 이름을 변경하면 CI/CD 파이프라인은 기존 권한 집합을 삭제하고 새 권한 집합을 만듭니다. • 이 파이프라인은 <code>SSOPipeline:true</code> 태그가 있는 권한 집합만 관리합니다. • 리포지토리의 같은 폴더에 여러 권한 집합과 할당 템플릿을 둘 수 있습니다. • 템플릿을 삭제하면 파이프라인이 해당 할당 또는 권한 집합을 삭제합니다. 	

작업	설명	필요한 기술
	<ul style="list-style-type: none"> • 전체 할당 JSON 블록을 삭제하면 파이프라인이 IAM Identity Center에서 할당을 삭제합니다. • AWS 계정에 할당된 권한 집합은 삭제할 수 없습니다. 우선 권한 집합의 할당을 취소해야 합니다. 	

문제 해결

문제	Solution
<p>액세스 거부 오류</p>	<p>CloudFormation 템플릿과 템플릿 내에 정의된 리소스를 배포하는 데 필요한 권한이 있는지 확인하십시오. 자세한 내용은 CloudFormation 설명서의 액세스 제어를 참조하십시오.</p>
<p>검증 단계의 파이프라인 오류</p>	<p>이 오류는 권한 집합이나 할당 템플릿에 오류가 있는 경우 나타납니다.</p> <ol style="list-style-type: none"> 1. 에서 CodeBuild 빌드 세부 정보를 확인하세요. 2. 구축 로그에서 구축 실패 원인에 대한 자세한 정보를 제공하는 유효성 검사 오류를 찾아보세요. 3. 권한 집합 또는 할당 템플릿을 업데이트한 다음 리포지토리에 커밋합니다. 4. CI/CD 파이프라인이 프로젝트를 다시 시작합니다. CodeBuild 상태를 모니터링하여 유효성 검사 오류가 해결되었는지 확인합니다.

관련 리소스

- [권한 세트](#)(IAM Identity Center 설명서)

AWS Secrets Manager를 사용한 보안 인증 정보 관리

작성자: Durga Prasad Cheepuri(AWS)

작성자: AWS

환경: PoC 또는 파일럿

기술: 데이터베이스, 보안, 자격 증명, 규정 준수

AWS 서비스: AWS Secrets Manager

요약

이 패턴은 AWS Secrets Manager를 사용하여 Java Spring 애플리케이션의 데이터베이스 보안 인증 정보를 동적으로 가져오는 과정을 안내합니다.

과거에는 데이터베이스에서 정보를 검색하는 사용자 지정 애플리케이션을 생성하면 일반적으로 데이터에 액세스하기 위한 보안 인증 정보(보안 암호)를 애플리케이션에 직접 포함시켜야 했습니다. 보안 인증 정보를 교체할 때는 시간을 투자하여 새 보안 인증 정보를 사용하도록 애플리케이션을 업데이트한 다음, 업데이트된 애플리케이션을 배포해야 했습니다. 보안 인증 정보를 공유하는 애플리케이션이 여러 개 있는데 이 중 하나를 업데이트하지 못한 경우 해당 애플리케이션이 침해될 수 있었습니다. 이러한 위험 때문에 많은 고객들은 정기적으로 보안 인증 정보를 교체하지 않기로 결정하며, 이 위험 대신 다른 위험에 직면하게 됩니다.

Secrets Manager는 코드의 암호를 포함해 하드 코딩된 보안 인증 정보를 프로그래밍 방식으로 보안 암호를 검색하도록 하는 API 직접 호출로 바꿀 수 있습니다. 이렇게 하면 보안 암호가 해당 위치에 있지 않기 때문에 여러분의 코드를 검사하는 누군가에 의해 보안 암호가 손상되지 않도록 방지할 수 있습니다. 또한 사용자가 지정한 일정에 따라 Secrets Manager가 자동으로 보안 암호를 교체하도록 구성할 수 있습니다. 따라서 단기 보안 암호로 장기 보안 암호를 교체할 수 있어 손상 위험이 크게 줄어듭니다. 자세한 내용은 [AWS Secrets Manager 설명서](#)를 참조하세요.

사전 조건 및 제한 사항

사전 조건

- Secrets Manager에 대한 액세스 권한이 있는 AWS 계정
- Java Spring 애플리케이션

아키텍처

소스 기술 스택

- application.properties 파일에서 관리되는 DB 보안 인증 정보를 사용하여 데이터베이스에 액세스하는 코드를 포함하는 Java Spring 애플리케이션입니다.

대상 기술 스택

- Secrets Manager에서 관리되는 DB 보안 인증 정보를 사용하여 데이터베이스에 액세스하는 코드를 포함하는 Java Spring 애플리케이션입니다. application.properties 파일에 Secrets Manager의 비밀이 보관되어 있습니다.

Secrets Manager를 애플리케이션과 통합

도구

- Secrets Manager – [AWS Secrets Manager](#)는 비밀을 보다 쉽게 관리할 수 있게 해주는 AWS 서비스입니다. 보안 암호는 데이터베이스 보안 인증 정보, 암호, 타사 API 키 및 임의 텍스트가 될 수 있습니다. Secrets Manager 콘솔, Secrets Manager 명령줄 인터페이스(CLI) 또는 Secrets Manager API 및 SDK를 사용하여 중앙에서 이러한 암호를 저장하고 해당 암호에 대한 액세스를 제어할 수 있습니다.

에픽

암호는 Secrets Manager에 저장

작업	설명	필요한 기술
DB 보안 인증 정보를 Secrets Manager에 암호로 저장합니다.	Secrets Manager 설명서의 암호 생성 에 나와 있는 단계에 따라 Secrets Manager에 Amazon Relational Database Service(RDS) 또는 기타 DB 보안 인증 정보를 암호로 저장하세요.	시스템 관리자

작업	설명	필요한 기술
Spring 애플리케이션이 Secrets Manager에 액세스할 수 있도록 권한을 설정합니다.	Java Spring 애플리케이션이 Secrets Manager를 사용하는 방식에 따라 적절한 권한을 설정합니다. 비밀에 대한 액세스를 제어하려면 Secrets Manager 설명서의 ID 기반 정책(IAM 정책) 및 Secrets Manager용 ABAC 사용 및 Secrets Manager의 리소스 기반 정책 사용 섹션에 제공된 정보를 기반으로 정책을 생성하세요. Secrets Manager 설명서의 암호 값 검색 섹션의 단계를 따르세요.	시스템 관리자

Spring 애플리케이션 업데이트

작업	설명	필요한 기술
Secrets Manager를 사용하려면 JAR 종속성을 추가하세요.	자세한 내용은 추가 정보 섹션을 참조하세요.	Java 개발자
Spring 애플리케이션에 암호의 세부 정보를 추가합니다.	application.properties 파일을 비밀 이름, 엔드포인트, AWS 리전으로 업데이트합니다. 예제를 보려면 추가 정보 섹션을 참조하세요.	Java 개발자
Java에서 DB 보안 인증 정보 검색 코드를 업데이트하세요.	애플리케이션에서 DB 보안 인증 정보를 가져오는 Java 코드를 업데이트하여 Secrets Manager에서 해당 세부 정보를 가져옵니다. 예제 코드는 추가 정보 섹션을 참조하세요.	Java 개발자

관련 리소스

- [AWS Secrets Manager 설명서](#)
- [ID 기반 정책\(IAM 정책\) 및 Secrets Manager용 ABAC 사용](#)
- [Secrets Manager를 위한 리소스 기반 정책 사용](#)
- [샘플 코드](#)

추가 정보

Secrets Manager 사용을 위한 JAR 종속성 추가

Maven:

```
<groupId>com.amazonaws</groupId>
  <artifactId>aws-java-sdk-secretsmanager</artifactId>
  <version>1.11.355 </version>
```

Gradle:

```
compile group: 'com.amazonaws', name: 'aws-java-sdk-secretsmanager', version:
'1.11.355'
```

application.properties 파일에 암호의 세부 정보 업데이트

```
spring.aws.secretsmanager.secretName=postgres-local
spring.aws.secretsmanager.endpoint=secretsmanager.us-east-1.amazonaws.com
spring.aws.secretsmanager.region=us-east-1
```

Java에서 DB 보안 인증 정보 검색 코드 업데이트

```
String secretName = env.getProperty("spring.aws.secretsmanager.secretName");
String endpoints = env.getProperty("spring.aws.secretsmanager.endpoint");
String AWS Region = env.getProperty("spring.aws.secretsmanager.region");
AwsClientBuilder.EndpointConfiguration config = new
    AwsClientBuilder.EndpointConfiguration(endpoints, AWS Region);
AWSecretsManagerClientBuilder clientBuilder =
    AWSecretsManagerClientBuilder.standard();
clientBuilder.setEndpointConfiguration(config);
AWSecretsManager client = clientBuilder.build();
```

```
ObjectMapper objectMapper = new ObjectMapper();

JsonNode secretsJson = null;

ByteBuffer binarySecretData;

GetSecretValueRequest getSecretValueRequest = new
    GetSecretValueRequest().withSecretId(secretName);

GetSecretValueResult getSecretValueResponse = null;

try {
    getSecretValueResponse = client.getSecretValue(getSecretValueRequest);
}

catch (ResourceNotFoundException e) {
    log.error("The requested secret " + secretName + " was not found");
}

catch (InvalidRequestException e) {
    log.error("The request was invalid due to: " + e.getMessage());
}

catch (InvalidParameterException e) {
    log.error("The request had invalid params: " + e.getMessage());
}

if (getSecretValueResponse == null) {
    return null;
} // Decrypted secret using the associated KMS key // Depending on whether the
secret was a string or binary, one of these fields will be populated

String secret = getSecretValueResponse.getSecretString();

if (secret != null) {
    try {
        secretsJson = objectMapper.readTree(secret);
    }

    catch (IOException e) {
        log.error("Exception while retrieving secret values: " +
            e.getMessage());
    }
}
```

```
}  
  
else {  
    log.error("The Secret String returned is null");  
  
    return null;  
  
}  
String host = secretsJson.get("host").textValue();  
String port = secretsJson.get("port").textValue();  
String dbname = secretsJson.get("dbname").textValue();  
String username = secretsJson.get("username").textValue();  
String password = secretsJson.get("password").textValue();  
}
```

시작 시 전송 중 암호화가 있는지 Amazon EMR 클러스터 모니터링

작성자: Susanne Kangnoh(AWS)

환경: 프로덕션

기술: 분석, 빅 데이터, 클라우
드 네이티브, 보안, 자격 증명,
규정 준수

워크로드: 오픈 소스

AWS 서비스: 아마존 EMR, 아
마존 SNS, AWS CloudTrail, 아
마존 CloudWatch

요약

이 패턴은 시작 시 Amazon EMR 클러스터를 모니터링하고 전송 중 암호화가 활성화되지 않은 경우 알림을 보내는 보안 제어를 제공합니다.

Amazon EMR은 Apache Hadoop과 같은 빅 데이터 프레임워크를 쉽게 실행하여 데이터를 처리하고 분석할 수 있게 해주는 웹 서비스입니다. Amazon EMR을 사용하면 매핑과 축소 단계를 동시에 실행하여 비용 효율적인 방식으로 방대한 양의 데이터를 처리할 수 있습니다.

데이터 암호화는 권한이 없는 사용자가 저장 데이터 또는 전송 중 데이터에 액세스하거나 읽는 것을 방지합니다. 저장 데이터는 Amazon Simple Storage Service(S3)를 통해 각 노드의 로컬 파일 시스템, Hadoop 분산 파일 시스템(HDFS) 또는 EMR 파일 시스템(EMRFS)과 같은 미디어에 저장되는 데이터를 말합니다. 전송 중인 데이터는 네트워크를 통해 이동하며 작업 간에 이동하는 데이터를 말합니다. 전송 중 암호화는 Apache Spark, Apache TEZ, Apache Hadoop, Apache HBase 및 Presto에 대한 오픈 소스 암호화 기능을 지원합니다. AWS Command Line Interface(AWS CLI) 콘솔 또는 AWS SDK에서 보안 구성을 생성하고 데이터 암호화 설정을 지정하여 암호화를 활성화합니다. 다음 두 가지 방법으로 전송 중 암호화를 위한 암호화 아티팩트를 제공할 수 있습니다.

- Amazon S3로 인증서 압축 파일을 업로드합니다.
- 암호화 아티팩트를 제공하는 사용자 정의 Java 클래스를 참조합니다.

이 패턴에 포함된 보안 제어는 API 호출을 모니터링하고 RunJobFlow작업에 대해 Amazon CloudWatch Events 이벤트를 생성합니다. 이벤트는 Python 스크립트를 실행하는 AWS Lambda 함수

를 직접 호출합니다. 함수는 이벤트 JSON 입력에서 EMR 클러스터 ID를 가져오고 다음 검사를 수행하여 보안 위반 여부를 확인합니다.

- EMR 클러스터에 Amazon EMR 전용 보안 구성이 있는지 확인합니다.
- 클러스터에 보안 구성이 있는 경우 전송 중 암호화가 활성화되어 있는지 확인하세요.
- 클러스터에 보안 구성이 없는 경우 Amazon Simple Notification Service(SNS)를 사용하여 사용자가 제공한 이메일 주소로 알림을 보내세요. 알림은 EMR 클러스터 이름, 위반 세부 정보, AWS 리전 및 계정 정보, 알림의 출처가 되는 AWS Lambda ARN(Amazon 리소스 이름)을 지정합니다.

사전 조건 및 제한 사항

사전 조건

- 활성 상태의 AWS 계정
- 이 패턴과 함께 제공되는 Lambda 코드를 업로드하기 위한 S3 버킷입니다.
- 위반 알림을 받으려는 이메일 주소입니다.
- 모든 API 로그에 액세스할 수 있도록 Amazon EMR 로깅이 활성화되었습니다.

제한 사항

- 이 탐지 제어는 리전과 관련이 있으므로 모니터링하려는 각 AWS 리전에 배포해야 합니다.

제품 버전

- Amazon EMR 릴리스 4.8.0 이상입니다.

아키텍처

워크플로 아키텍처

자동화 및 규모 조정

- AWS Organizations를 사용하는 경우, [AWS StackSets Cloudformation](#)을 사용하여 모니터링하려는 여러 계정에 템플릿을 배포할 수 있습니다.

도구

서비스

- [Amazon EMR](#) – Amazon EMR은 AWS에서 [Apache Hadoop](#) 및 [Apache Spark](#)와 같은 빅 데이터 프레임워크 실행을 단순화하여 방대한 양의 데이터를 처리하고 분석하는 관리형 클러스터 플랫폼입니다. 이러한 프레임워크 및 관련 오픈 소스 프로젝트를 사용하면 분석 목적 및 비즈니스 인텔리전스 워크로드용 데이터를 처리할 수 있습니다. 또한 Amazon EMR을 사용하여 Amazon S3 및 Amazon DynamoDB와 같은 기타 AWS 데이터 스토어 및 데이터베이스에서 많은 양의 데이터를 양방향으로 변환하고 이동할 수 있습니다.
- [AWS CloudFormation](#) — AWS는 AWS 리소스를 모델링 및 설정하고, 빠르고 일관되게 프로비저닝하고, 수명 주기 전반에 걸쳐 관리할 수 있도록 CloudFormation 지원합니다. 템플릿을 사용하여 리소스와 해당 종속성을 설명하고, 리소스를 개별적으로 관리하는 대신 스택으로 함께 시작하고 구성할 수 있습니다. 여러 AWS 계정 및 AWS 리전에서 스택을 관리하고 프로비저닝할 수 있습니다.
- [AWS Cloudwatch Events](#) — Amazon CloudWatch Events는 AWS 리소스의 변경 사항을 설명하는 시스템 이벤트의 스트림을 거의 실시간으로 제공합니다. CloudWatch 이벤트는 운영 변경이 발생하는 즉시 이를 인지하고 필요에 따라 수정 조치를 취하여 환경에 대응하고, 기능을 활성화하고, 변경하고, 상태 정보를 캡처합니다.
- [AWS Lambda](#) - AWS Lambda는 서버를 프로비저닝하거나 관리하지 않고도 코드를 실행할 수 있도록 지원하는 컴퓨팅 서비스입니다. Lambda는 필요 시에만 코드를 실행하며, 일일 몇 개의 요청에서 초당 수천 개의 요청까지 자동으로 확장이 가능합니다. 사용한 컴퓨팅 시간만큼만 비용을 지불하고, 코드가 실행되지 않을 때는 요금이 부과되지 않습니다.
- [AWS SNS](#) – Amazon Simple Notification Service(SNS)는 게시자와 클라이언트 간에 웹 서버와 이메일 주소를 비롯한 메시지 전송을 조정 및 관리합니다. 구독자는 구독하는 주제에 게시된 모든 메시지를 수신하며 주제에 대한 모든 구독자는 동일한 메시지를 수신합니다.

코드

이 패턴에는 두 개의 파일이 포함된 첨부 파일이 포함됩니다.

- `EMRInTransitEncryption.zip`은(는) 보안 제어(Lambda 코드)가 포함된 압축 파일입니다.
- `EMRInTransitEncryption.yml`보안 제어를 배포하는 CloudFormation 템플릿입니다.

이러한 파일을 사용하는 방법에 대한 자세한 내용은 에픽 섹션을 참조하세요.

에픽

보안 제어의 배포

작업	설명	필요한 기술
코드를 S3 버킷에 업로드합니다.	새 S3 버킷을 생성하거나 기존 S3 버킷을 사용하여 첨부된 EMRInTransitEncryption.zip 파일(Lambda 코드)을 업로드합니다. 이 버킷은 평가하려는 CloudFormation 템플릿 및 리소스와 동일한 AWS 리전에 있어야 합니다.	클라우드 아키텍트
CloudFormation 템플릿을 배포하십시오.	S3 버킷과 동일한 AWS 리전에서 Cloudformation 콘솔을 열고 첨부 파일에 제공된 EMRInTransitEncryption.yml 파일을 배포합니다. 다음 에픽에서 템플릿 파라미터에 대한 값을 입력합니다.	클라우드 아키텍트,

CloudFormation 템플릿의 파라미터를 완성하세요.

작업	설명	필요한 기술
S3 버킷 이름을 제공합니다.	첫 번째 에픽에서 생성하거나 선택한 S3 버킷의 이름을 입력합니다. 이 S3 버킷에는 Lambda 코드용.zip 파일이 들어 있으며, 평가될 템플릿 및 리소스와 동일한 CloudFormation AWS 리전에 있어야 합니다.	클라우드 아키텍트
S3 키를 입력합니다.	S3 버킷에 있는 Lambda 코드 .zip 파일의 위치를 앞에	클라우드 아키텍트

작업	설명	필요한 기술
	슬래시 없이 지정합니다(예: EMRInTransitEncryption.zip 또는 controls/EMRInTransitEncryption.zip).	
이메일 주소를 입력합니다.	위반 알림을 받을 사용 중인 이메일 주소를 지정합니다.	클라우드 아키텍트
로깅 수준을 지정합니다.	Lambda 로그에 대한 로깅 수준 및 상세 정보를 지정합니다. Info는 애플리케이션 진행 상황에 대한 자세한 정보 메시지를 지정하며 디버깅용으로만 사용해야 합니다. Error는 애플리케이션을 계속 실행할 수 있는 오류 이벤트를 지정합니다. Warning은 잠재적으로 유해한 상황을 지정합니다.	클라우드 아키텍트

구독 확인

작업	설명	필요한 기술
이메일 구독을 확인합니다.	CloudFormation 템플릿이 성공적으로 배포되면 제공한 이메일 주소로 구독 이메일 메시지가 전송됩니다. 알림을 받기 시작하려면 이 이메일 구독을 확인해야 합니다.	클라우드 아키텍트

관련 리소스

- [AWS CloudFormation 콘솔에서 스택 생성](#) (AWS CloudFormation 설명서)
- [암호화 옵션](#) (Amazon EMR 설명서)

첨부

이 문서와 관련된 추가 콘텐츠에 액세스하려면 [attachment.zip](#) 파일의 압축을 풉니다.

Amazon ElastiCache 클러스터의 미사용 암호화 모니터링

작성자: Susanne Kangnoh(AWS)

환경: 프로덕션

기술: 보안, 자격 증명, 규정 준수, 데이터베이스, 인프라, 클라우드 네이티브

워크로드: 오픈 소스

AWS 서비스: 아마존 SNS, 아마존 CloudWatch, 아마존 ElastiCache

요약

ElastiCache Amazon은 클라우드에 인메모리 데이터 스토어 또는 캐시 환경을 배포하기 위한 확장 가능하고 비용 효율적인 고성능 캐싱 솔루션을 제공하는 Amazon Web Services (AWS) 서비스입니다. 처리량이 높고 지연 시간이 짧은 인메모리 데이터 스토어에서 데이터를 검색합니다. 이 기능은 캐싱, 세션 스토어, 게임, 지리 공간 서비스, 실시간 분석 및 대기열과 같은 실시간 사용 사례에 널리 사용됩니다. ElastiCache 밀리초 미만의 응답 시간을 제공하는 Redis 및 Memcached 데이터 스토어를 제공합니다.

데이터 암호화는 권한 없는 사용자가 Redis 클러스터 및 관련 캐시 스토리지 시스템에서 사용 가능한 민감한 데이터를 읽는 것을 방지하는 데 도움이 됩니다. 여기에는 영구 미디어에 저장된 데이터인 저장 데이터와 클라이언트와 캐시 서버 사이의 네트워크를 통해 이동하는 동안 가로챌 수 있는 데이터인 전송 중 데이터가 포함됩니다.

복제 그룹을 생성할 때 파라미터를 true로 설정하여 ElastiCache Redis용 미사용 암호화를 활성화할 수 있습니다. AtRestEncryptionEnabled 이 파라미터를 활성화하면 동기화, 백업 및 스왑 작업 중에 디스크를 암호화하고 Amazon Simple Storage Service(S3)에 저장된 백업을 암호화합니다. 기존 복제 그룹에 대해서는 미사용 데이터 암호화를 설정할 수 없습니다. 복제 그룹을 생성할 때 다음 두 가지 방법으로 저장 데이터 암호화를 활성화할 수 있습니다.

- 저장된 서비스 관리형 암호화를 사용하는 기본 옵션을 선택합니다.
- 고객 관리형 키를 사용하고 AWS Key Management Service(AWS KMS)의 키 ID 또는 Amazon 리소스 이름(ARN)을 제공합니다.

이 패턴은 API 호출을 모니터링하고 CreateReplicationGroup 작업에서 Amazon CloudWatch Events 이벤트를 생성하는 보안 제어를 제공합니다. 이 이벤트는 Python 스크립트를 실행하는 AWS Lambda 함수를 직접 호출합니다. 함수는 이벤트 JSON 입력에서 복제 그룹 ID를 가져오고 다음 검사를 수행하여 보안 위반 여부를 확인합니다.

- AtRestEncryptionEnabled 키가 존재하는지 확인합니다.
- AtRestEncryptionEnabled 존재하는 경우 값을 검사하여 값이 참인지 확인합니다.
- AtRestEncryptionEnabled 값이 false로 설정된 경우, Amazon Simple Notification Service (Amazon SNS) 알림을 사용하여 위반을 추적하고 사용자가 제공한 이메일 주소로 위반 메시지를 보내는 변수를 설정합니다.

사전 조건 및 제한 사항

사전 조건

- 활성 상태의 AWS 계정
- 제공된 Lambda 코드를 업로드하기 위한 S3 버킷입니다.
- 위반 알림을 받으려는 이메일 주소입니다.
- ElastiCache 모든 API 로그에 액세스할 수 있도록 로깅이 활성화되었습니다.

제한 사항

- 이 탐지 제어는 리전별이므로 모니터링하려는 각 AWS 리전에 배포해야 합니다.
- 이 컨트롤은 Virtual Private Cloud(VPC)에서 실행 중인 복제 그룹을 지원합니다.
- 컨트롤은 다음 노드 유형을 실행하는 복제 그룹을 지원합니다.
 - R5, R4, R3
 - M5, M4, M3
 - T3, T2

제품 버전

- ElastiCache Redis 버전 3.2.6 이상의 경우

아키텍처

워크플로 아키텍처

자동화 및 규모 조정

- AWS Organizations를 사용하는 경우, [AWS StackSets Cloudformation](#)을 사용하여 모니터링하려는 여러 계정에 이 템플릿을 배포할 수 있습니다.

도구

서비스

- [Amazon ElastiCache](#) — Amazon을 ElastiCache 사용하면 AWS 클라우드에서 분산된 인메모리 캐시 환경을 쉽게 설정, 관리 및 확장할 수 있습니다. 분산 캐시 환경 배포 및 관리와 관련된 복잡성을 제거하면서 크기 조정이 가능하고 비용 효율적인 고성능 인메모리 캐시를 제공합니다. ElastiCache Redis 및 Memcached 엔진 모두에서 작동합니다.
- [AWS CloudFormation](#) — AWS는 AWS CloudFormation 리소스를 모델링 및 설정하고, 빠르고 일관되게 프로비저닝하고, 수명 주기 전반에 걸쳐 관리할 수 있도록 지원합니다. 템플릿을 사용하여 리소스와 해당 종속성을 설명하고, 리소스를 개별적으로 관리하는 대신 스택으로 함께 시작하고 구성할 수 있습니다. 여러 AWS 계정 및 AWS 리전에서 스택을 관리하고 프로비저닝할 수 있습니다.
- [AWS Cloudwatch Events](#) — Amazon CloudWatch Events는 AWS 리소스의 변경 사항을 설명하는 시스템 이벤트의 스트림을 거의 실시간으로 제공합니다. CloudWatch 이벤트는 운영 변경이 발생하는 즉시 이를 인지하고 필요에 따라 수정 조치를 취하여 환경에 대응하고, 기능을 활성화하고, 변경하고, 상태 정보를 캡처합니다.
- [AWS Lambda](#) - AWS Lambda는 서버를 프로비저닝하거나 관리하지 않고도 코드를 실행할 수 있도록 지원하는 컴퓨팅 서비스입니다. Lambda는 필요 시에만 코드를 실행하며, 일일 몇 개의 요청에서 초당 수천 개의 요청까지 자동으로 확장이 가능합니다. 사용한 컴퓨팅 시간만큼만 비용을 지불하고, 코드가 실행되지 않을 때는 요금이 부과되지 않습니다.
- [Amazon SNS](#) – Amazon Simple Notification Service(SNS) 는 게시자와 클라이언트 간에 웹 서버와 이메일 주소를 비롯한 메시지 전송을 조정 및 관리합니다. 구독자는 구독하는 주제에 게시된 모든 메시지를 수신하며 주제에 대한 모든 구독자는 동일한 메시지를 수신합니다.

코드

이 패턴에는 두 개의 파일이 포함된 첨부 파일이 포함됩니다.

- ElasticCache-EncryptionAtRest.zip은(는) 보안 제어(Lambda 코드)가 포함된 압축 파일입니다.
- elasticache_encryption_at_rest.yml보안 제어를 배포하는 CloudFormation 템플릿입니다.

이러한 파일을 사용하는 방법에 대한 자세한 내용은 에픽 섹션을 참조하세요.

에픽

보안 제어의 배포

작업	설명	필요한 기술
코드를 S3 버킷에 업로드합니다.	새 S3 버킷을 생성하거나 기존 S3 버킷을 사용하여 첨부된 ElasticCache-EncryptionAtRest.zip 파일 (Lambda 코드)을 업로드합니다. 이 버킷은 평가하려는 리소스와 동일한 AWS 리전에 있어야 합니다.	클라우드 아키텍트
CloudFormation 템플릿을 배포합니다.	S3 버킷과 동일한 AWS 리전에서 Cloudformation 콘솔을 열고 첨부 파일에 제공된 elasticache_encryption_at_rest.yml 파일을 배포합니다. 다음 에픽에서 템플릿 파라미터에 대한 값을 입력합니다.	클라우드 아키텍트

CloudFormation 템플릿의 파라미터를 완성하세요.

작업	설명	필요한 기술
S3 버킷 이름을 제공합니다.	첫 번째 에픽에서 생성하거나 선택한 S3 버킷의 이름을 입력합니다. 이 S3 버킷에는 Lambda 코드용.zip 파일이 들어 있으며, 평가될 템플릿 및 리소스와 동일한 CloudFormation AWS 리전에 있어야 합니다.	클라우드 아키텍트
S3 키를 입력합니다.	S3 버킷에 있는 Lambda 코드 .zip 파일의 위치를 앞에 슬래시 없이 입력합니다(예: ElasticCache-EncryptionAtRest.zip 또는 controls/ElasticCache-EncryptionAtRest.zip).	클라우드 아키텍트
이메일 주소를 입력합니다.	위반 알림을 받을 사용 중인 이메일 주소를 입력합니다.	클라우드 아키텍트
로깅 수준을 지정합니다.	로깅 수준 및 상세 정보를 지정합니다. Info는 애플리케이션 진행 상황에 대한 자세한 정보 메시지를 지정하며 디버깅용으로만 사용해야 합니다. Error는 애플리케이션을 계속 실행할 수 있는 오류 이벤트를 지정합니다. Warning은 잠재적으로 유해한 상황을 지정합니다.	클라우드 아키텍트

구독 확인

작업	설명	필요한 기술
이메일 구독을 확인합니다.	CloudFormation 템플릿이 성공적으로 배포되면 제공한 이메일 주소로 구독 이메일 메시지가 전송됩니다. 알림을 받기 시작하려면 이 이메일 구독을 확인해야 합니다.	클라우드 아키텍트

관련 리소스

- [AWS CloudFormation 콘솔에서 스택 생성](#) (AWS CloudFormation 설명서)
- [ElastiCache Redis용 저장 중 암호화](#) (Amazon ElastiCache 설명서)

첨부

이 문서와 관련된 추가 콘텐츠에 액세스하려면 [attachment.zip](#) 파일의 압축을 풉니다.

AWS Config를 사용하여 EC2 인스턴스 키 페어를 모니터링

제작: Wassim Benhallam(AWS)과 Vikrant Telkar(AWS)

환경: 프로덕션

기술: 보안, 자격 증명 및 규정 준수

AWS 서비스: Amazon SNS, AWS Config, AWS Lambda

요약

Amazon Web Services(AWS) 클라우드에서 Amazon Elastic Compute Cloud(Amazon EC2) 인스턴스를 시작하는 경우 가장 좋은 방법은 인스턴스에 연결하는 데 기존 키 페어를 사용하거나 생성하는 것입니다. 인스턴스에 저장된 퍼블릭 키와 사용자에게 제공되는 프라이빗 키로 구성된 키 페어를 사용하면 Secure Shell(SSH)를 통해 인스턴스에 안전하게 액세스할 수 있으며 암호 사용을 피할 수 있습니다. 하지만 사용자가 키 페어를 연결하지 않고 실수로 인스턴스를 시작하는 경우가 있습니다. 키 페어는 인스턴스 시작 시에만 할당될 수 있으므로 키 페어 없이 시작되는 모든 인스턴스를 신속하게 식별하여 비준수 상태로 플래그를 지정하는 것이 중요합니다. 이는 인스턴스 액세스에 반드시 키 페어를 사용해야 하는 계정이나 환경에서 작업할 때 특히 유용합니다.

이 패턴은 AWS Config에서 EC2 인스턴스 키 페어를 모니터링하기 위한 사용자 지정 규칙을 생성하는 방법을 설명합니다. 인스턴스가 규정을 준수하지 않는 것으로 확인되면 Amazon 이벤트를 통해 시작된 Amazon Simple Notification Service (Amazon SNS) 알림을 사용하여 알림이 전송됩니다. EventBridge

사전 조건 및 제한 사항

사전 조건

- 활성 상태의 AWS 계정.
- 모니터링하려는 AWS 리전에 대해 AWS Config가 활성화되고 모든 AWS 리소스를 기록하도록 구성됨

제한 사항

- 이 솔루션은 리전별로 다릅니다. 이러한 모든 리소스를 동일한 AWS 리전 내에 생성해야 합니다.

아키텍처

대상 기술 스택

- AWS Config
- 아마존 EventBridge
- AWS Lambda
- Amazon SNS

대상 아키텍처

1. AWS Config가 규칙을 시작합니다.
2. 이 규칙은 Lambda 함수를 간접적으로 호출하여 EC2 인스턴스의 규정 준수를 평가합니다.
3. Lambda 함수는 업데이트된 규정 준수 상태를 AWS Config에 전송합니다.
4. AWS Config는 이 이벤트를 전송합니다. EventBridge
5. EventBridge 규정 준수 변경 알림을 SNS 주제에 게시합니다.
6. Amazon SNS는 이메일로 경고를 보냅니다.

자동화 및 규모 조정

이 솔루션은 리전 내 EC2 인스턴스를 원하는 수만큼 모니터링할 수 있습니다.

도구

도구

- [AWS Config](#) – AWS Config는 AWS 리소스 구성을 측정, 감사 및 평가할 수 있도록 지원합니다. 를 사용하여 AWS 리소스 구성을 측정, 감사 및 평가할 수 있습니다. 는 AWS 리소스 구성을 지속적으로 모니터링 및 기록하여 사용자가 원하는 구성을 기준으로 기록된 구성을 자동으로 평가할 수 있습니다.
- [Amazon EventBridge](#) — EventBridge Amazon은 애플리케이션을 다양한 소스의 데이터와 연결하는 서버리스 이벤트 버스 서비스입니다.

- [AWS Lambda](#) – AWS Lambda는 서버를 프로비저닝하거나 관리하지 않고도 코드를 실행하고, 워크로드를 인식하는 클러스터 확장 로직을 생성하고, 이벤트 통합을 유지 관리하거나 런타임을 관리할 수 있도록 지원하는 서버리스 컴퓨팅 서비스입니다.
- [아마존 SNS](#) — 아마존 심플 알림 서비스 (Amazon SNS) 는 (A2A) application-to-person 및 (A2P) 통신 application-to-application 모두를 위한 완전 관리형 메시징 서비스입니다.

코드

Lambda 함수에 대한 코드가 첨부되어 있습니다.

에픽

Amazon EC2 규정 준수를 평가하는 Lambda 함수를 생성합니다.

작업	설명	필요한 기술
Amazon EC2에 대해 새로운 AWS Identity 및 Access Management(IAM) 역할을 생성합니다.	AWS Management Console에서 IAM을 선택한 다음, Lambda를 신뢰할 수 있는 엔터티로 사용하고 AmazonEventBridgeFullAccess 및 AWSConfigRulesExecutionRole 권한을 추가하여 역할을 생성합니다. 자세한 내용은 AWS 설명서 를 참조하세요.	DevOps
Lambda 함수를 생성하고 배포합니다.	<ol style="list-style-type: none"> 1. Lambda 콘솔에서 새로 작성하여 Python 3.6을 런타임으로 사용하고 이전에 생성된 IAM 역할을 사용하여 함수를 생성합니다. Amazon 리소스 이름(ARN)을 적어 둡니다. 2. 코드 탭에서 lambda_function.py 을(를) 선택 	DevOps

작업	설명	필요한 기술
	<p>하여 이 패턴에 첨부된 코드를 붙여넣습니다.</p> <p>3. 변경 사항을 저장하려면 배포를 선택합니다.</p>	

사용자 지정 AWS Config 규칙 생성

작업	설명	필요한 기술
사용자 지정 AWS Config 규칙을 추가합니다.	<p>AWS Config 콘솔에서 다음 설정을 사용하여 사용자 지정 규칙을 추가합니다.</p> <ul style="list-style-type: none"> • ARN – 이전에 생성된 Lambda 함수의 ARN • 트리거 유형 – 구성 변경 • 변경 범위 – 리소스 • 리소스 유형 – Amazon EC2 인스턴스 <p>자세한 내용은 AWS 설명서를 참조하세요.</p>	DevOps

규정 준수 변경 이벤트가 감지될 때 이메일 알림을 구성

작업	설명	필요한 기술
SNS 주제 및 구독자를 생성합니다.	Amazon SNS 콘솔에서 표준을 유형으로 사용하여 주제를 생성한 다음 이메일을 프로토콜로 사용하여 구독을 생성합니다.	DevOps

작업	설명	필요한 기술
	<p>이메일을 수신하면 이메일에서 구독 확인 링크를 선택합니다.</p> <p>자세한 내용은 AWS 설명서를 참조하세요.</p>	
Amazon SNS 알림을 시작하는 EventBridge 규칙을 생성하십시오.	<p>EventBridge 콘솔에서 다음 설정을 사용하여 규칙을 생성합니다.</p> <ul style="list-style-type: none"> 서비스 이름 – AWS Config 이벤트 유형 – Config 규칙 규정 준수 변경 메시지 유형 — 특정 메시지 유형, ComplianceChangeNotification 특정 규칙 이름 – 이전에 생성한 AWS Config 규칙의 이름 대상 – SNS 주제, 이전에 생성한 주제 <p>자세한 내용은 AWS 설명서를 참조하세요.</p>	DevOps

규칙 및 알림 확인

작업	설명	필요한 기술
EC2 인스턴스를 생성합니다.	모든 유형의 EC2 인스턴스 2개를 생성하고 키 페어를 연결한 다음, 키 페어가 없는 EC2 인스턴스 1개를 생성합니다.	DevOps

작업	설명	필요한 기술
규칙을 확인합니다.	<ol style="list-style-type: none"> 1. AWS Config 콘솔의 규칙 페이지에서 규칙을 선택합니다. 2. 규정 준수 및 비준수 EC2 인스턴스를 보려면 리소스 범위를 모두로 변경하세요. 두 개의 인스턴스가 규정 준수로 나열되고 하나의 인스턴스가 비준수로 나열되어 있는지 확인하세요. 3. EC2 인스턴스의 규정 준수 상태에 관한 Amazon SNS 이메일 알림을 받을 때까지 기다리세요. 	DevOps

관련 리소스

- [AWS 서비스에 대한 권한을 위임할 역할 생성](#)
- [AWS Config에서 사용자 지정 규칙 생성](#)
- [Amazon SNS 주제 생성](#)
- [Amazon SNS 주제 구독](#)
- [아마존에서 규칙 생성 EventBridge](#)

첨부

이 문서와 관련된 추가 콘텐츠에 액세스하려면 [attachment.zip](#) 파일의 압축을 풉니다.

보안 그룹의 ElastiCache 클러스터 모니터링

작성자: Susanne Kangnoh(AWS)와 Archit Mathur(AWS)

환경: 프로덕션	기술: 보안, 자격 증명, 규정 준수, 데이터베이스, 인프라, 클라우드 네이티브	AWS 서비스: 아마존 SNS, AWS CloudTrail, 아마존 CloudWatch, 아마존 ElastiCache
----------	--	---

요약

ElastiCache Amazon은 클라우드에 인메모리 데이터 스토어 또는 캐시 환경을 배포하기 위한 확장 가능하고 비용 효율적인 고성능 캐싱 솔루션을 제공하는 Amazon Web Services (AWS) 서비스입니다. 처리량이 높고 지연 시간이 짧은 인메모리 데이터 스토어에서 데이터를 검색합니다. 이 기능은 캐싱, 세션 스토어, 게임, 지리 공간 서비스, 실시간 분석 및 대기열과 같은 실시간 사용 사례에 널리 사용됩니다. ElastiCache 밀리초 미만의 응답 시간을 제공하는 Redis 및 Memcached 데이터 스토어를 제공합니다.

보안 그룹은 인바운드 및 아웃바운드 트래픽을 제어하여 ElastiCache 인스턴스의 가상 방화벽 역할을 합니다. 보안 그룹은 서브넷 수준이 아닌 인스턴스 수준에서 작동합니다. 각 보안 그룹에 대해 인스턴스에 대한 인바운드 트래픽을 제어하는 규칙과 아웃바운드 트래픽을 제어하는 별도의 규칙 세트를 추가합니다. 허용 규칙을 지정할 수 있지만 거부 규칙은 지정할 수 없습니다.

이 패턴은 API 호출을 모니터링하고 CreateReplicationGroup, CreateCacheClusterModifyCacheCluster, 및 ModifyReplicationGroup 작업에서 Amazon CloudWatch Events 이벤트를 생성하는 보안 제어를 제공합니다. 이 이벤트는 Python 스크립트를 실행하는 AWS Lambda 함수를 직접 호출합니다. 함수는 이벤트 JSON 입력에서 복제 그룹 ID를 가져오고 다음 검사를 수행하여 보안 위반 여부를 확인합니다.

- 클러스터의 보안 그룹이 Lambda 함수에 구성된 보안 그룹과 일치하는지 확인합니다.
- 클러스터의 보안 그룹이 일치하지 않는 경우 함수는 Amazon Simple Notification Service(SNS) 알림을 사용하여 사용자가 제공한 이메일 주소로 위반 메시지를 보냅니다.

사전 조건 및 제한 사항

사전 조건

- 활성 상태의 AWS 계정
- 제공된 Lambda 코드를 업로드하기 위한 S3 버킷입니다.
- 위반 알림을 받으려는 이메일 주소입니다.
- ElastiCache 모든 API 로그에 액세스할 수 있도록 로깅이 활성화되었습니다.

제한 사항

- 이 탐지 제어는 리전별이므로 모니터링하려는 각 AWS 리전에 배포해야 합니다.
- 이 컨트롤은 Virtual Private Cloud(VPC)에서 실행 중인 복제 그룹을 지원합니다.

아키텍처

워크플로 아키텍처

자동화 및 규모 조정

- AWS Organizations를 사용하는 경우, [AWS StackSets Cloudformation](#)을 사용하여 모니터링하려는 여러 계정에 이 템플릿을 배포할 수 있습니다.

도구

서비스

- [Amazon을 ElastiCache](#) 사용하면 AWS 클라우드에서 분산된 인메모리 캐시 환경을 쉽게 설정, 관리 및 확장할 수 있습니다. 분산 캐시 환경 배포 및 관리와 관련된 복잡성을 제거하면서 크기 조정이 가능하고 비용 효율적인 고성능 인메모리 캐시를 제공합니다. ElastiCache Redis 및 Memcached 엔진 모두에서 작동합니다.
- [AWS는 AWS CloudFormation](#) 리소스를 모델링 및 설정하고, 빠르고 일관되게 프로비저닝하고, 수명 주기 전반에 걸쳐 관리할 수 있도록 지원합니다. 템플릿을 사용하여 리소스와 해당 종속성을 설명하고, 리소스를 개별적으로 관리하는 대신 스택으로 함께 시작하고 구성할 수 있습니다. 여러 AWS 계정 및 AWS 리전에서 스택을 관리하고 프로비저닝할 수 있습니다.
- [AWS Cloudwatch Events](#)는 AWS 리소스의 변경 사항을 설명하는 시스템 이벤트의 스트림을 거의 실시간으로 제공합니다. CloudWatch 이벤트는 운영 변경이 발생하는 즉시 이를 인지하고 필요에 따라 수정 조치를 취하여 환경에 대응하고, 기능을 활성화하고, 변경하고, 상태 정보를 캡처합니다.

- [AWS Lambda](#)는 서버를 프로비저닝하거나 관리하지 않고도 코드를 실행할 수 있게 해주는 컴퓨팅 서비스입니다. Lambda는 필요 시에만 코드를 실행하며, 일일 몇 개의 요청에서 초당 수천 개의 요청까지 자동으로 확장이 가능합니다. 사용한 컴퓨팅 시간만큼만 비용을 지불하고, 코드가 실행되지 않을 때는 요금이 부과되지 않습니다.
- [Amazon Simple Notification Service\(SNS\)](#)는 게시자와 클라이언트 간에 웹 서버와 이메일 주소를 비롯한 메시지 전송을 조정 및 관리합니다. 구독자는 구독하는 주제에 게시된 모든 메시지를 수신하며 주제에 대한 모든 구독자는 동일한 메시지를 수신합니다.

코드

이 패턴에는 두 개의 파일이 포함된 첨부 파일이 포함됩니다.

- `ElastiCacheAllowedSecurityGroup.zip`은(는) 보안 제어(Lambda 코드)가 포함된 압축 파일입니다.
- `ElastiCacheAllowedSecurityGroup.yml`보안 제어를 배포하는 CloudFormation 템플릿입니다.

이러한 파일을 사용하는 방법에 대한 자세한 내용은 에픽 섹션을 참조하세요.

에픽

보안 제어의 배포

작업	설명	필요한 기술
코드를 S3 버킷에 업로드합니다.	새 S3 버킷을 생성하거나 기존 S3 버킷을 사용하여 첨부된 <code>ElastiCacheAllowedSecurityGroup.zip</code> 파일(Lambda 코드)을 업로드합니다. 이 버킷은 평가하려는 리소스와 동일한 AWS 리전에 있어야 합니다.	클라우드 아키텍트
CloudFormation 템플릿을 배포합니다.	S3 버킷과 동일한 AWS 리전에서 Cloudformation 콘솔을 열고 첨부 파일에 제공된	클라우드 아키텍트

작업	설명	필요한 기술
	ElastiCacheAllowedSecurityControl.yml 파일을 배포합니다. 다음 에픽에서 템플릿 파라미터에 대한 값을 입력합니다.	

CloudFormation 템플릿의 파라미터를 완성하세요.

작업	설명	필요한 기술
S3 버킷 이름을 제공합니다.	첫 번째 에픽에서 생성하거나 선택한 S3 버킷의 이름을 입력합니다. 이 S3 버킷에는 Lambda 코드용.zip 파일이 들어 있으며, 평가될 템플릿 및 리소스와 동일한 CloudFormation AWS 리전에 있어야 합니다.	클라우드 아키텍트
S3 키를 입력합니다.	S3 버킷에 있는 Lambda 코드 .zip 파일의 위치를 앞에 슬래시 없이 입력합니다(예: ElasticCacheAllowedSecurityGroup.zip 또는 controls/ElasticCacheAllowedSecurityGroup.zip).	클라우드 아키텍트
이메일 주소를 입력합니다.	위반 알림을 받을 사용 중인 이메일 주소를 입력합니다.	클라우드 아키텍트
로깅 수준을 지정합니다.	로깅 수준 및 상세 정보를 지정합니다. Info은(는) 애플리케이션 진행 상황에 대한 자세한 정보 메시지를 지정하며 디버깅용으로만 사용해야 합니다.	클라우드 아키텍트

작업	설명	필요한 기술
	Error은(는) 애플리케이션을 계속 실행할 수 있는 오류 이벤트를 지정합니다. Warning은(는) 잠재적으로 유해한 상황을 지정합니다.	

구독 확인

작업	설명	필요한 기술
이메일 구독을 확인합니다.	CloudFormation 템플릿이 성공적으로 배포되면 제공한 이메일 주소로 구독 이메일 메시지가 전송됩니다. 알림을 받기 시작하려면 이 이메일 구독을 확인해야 합니다.	클라우드 아키텍트

관련 리소스

- [AWS CloudFormation 콘솔에서 스택 생성](#) (AWS CloudFormation 설명서)
- [아마존 VPC 및 ElastiCache 보안](#) (ElastiCache 레디스용 아마존 설명서)

첨부

이 문서와 관련된 추가 콘텐츠에 액세스하려면 [attachment.zip](#) 파일의 압축을 풉니다.

IAM 루트 사용자 활동 모니터링

작성자: Mostefa Brougui(AWS)

코드 리포지토리: aws-iam-root-user-활동 모니터	환경: PoC 또는 파일럿	기술: 보안, 자격 증명, 규정 준수, 관리 및 거버넌스
워크로드: 기타 모든 워크로드	AWS 서비스: 아마존 EventBridge, AWS Lambda, 아마존 SNS, AWS ID 및 액세스 관리	

요약

모든 Amazon Web Services(AWS) 계정에는 루트 사용자가 있습니다. AWS Identity and Access Management(IAM)의 [보안 모범 사례](#)로서, 루트 사용자만 수행할 수 있는 작업을 완료하는 데 루트 사용자를 사용하는 것이 좋습니다. 이러한 작업의 전체 목록은 일반 참조의 [루트 사용자 보안 인증이 필요한 작업](#)을 참조하세요. 루트 사용자는 모든 AWS 리소스 및 결제 정보에 대한 전체 액세스 권한을 가지므로 이 계정을 사용하지 말고 루트 사용자 보안 인증 정보가 손상되었음을 나타내는 활동이 있는지 모니터링하는 것이 좋습니다.

이 패턴을 사용하여 IAM 루트 사용자를 모니터링하는 [이벤트 기반 아키텍처](#)를 설정합니다. 이 패턴은 여러 AWS 계정인 스포크 계정을 모니터링하고 단일 계정인 허브 계정에서 관리 및 보고를 중앙 집중화하는 hub-and-spoke 솔루션을 설정합니다.

IAM 루트 사용자 자격 증명을 사용하면 CloudWatch Amazon과 AWS는 각각 로그와 트레일에 활동을 CloudTrail 기록합니다. 스포크 계정에서 Amazon EventBridge 규칙은 이벤트를 허브 계정의 중앙 [이벤트 버스로](#) 보냅니다. 허브 계정에서 EventBridge 규칙은 이벤트를 AWS Lambda 함수로 보냅니다. 이 함수는 루트 사용자 활동을 알려주는 Amazon Simple Notification Service(SNS) 주제를 사용합니다.

이 패턴에서는 AWS CloudFormation 템플릿을 사용하여 스포크 계정에 모니터링 및 이벤트 처리 서비스를 배포합니다. HashiCorp Terraform 템플릿을 사용하여 허브 계정에 이벤트 관리 및 알림 서비스를 배포합니다.

사전 조건 및 제한 사항

사전 조건

1. AWS 환경에 AWS 리소스를 배포할 수 있는 권한.
2. 스택 세트를 배포할 수 있는 권한. CloudFormation 자세한 [내용은 스택 세트 작업의 사전 요구 사항](#) (CloudFormation 문서) 을 참조하십시오.
3. Terraform 설치 및 사용할 준비 완료. 자세한 내용은 [시작하기 – AWS](#)(Terraform 설명서)를 참조하십시오.
4. 각 스포크 계정의 기존 트레일. 자세한 내용은 [AWS 시작하기 CloudTrail](#) (CloudTrail 설명서) 를 참조하십시오.
5. 트레일은 CloudWatch Logs로 이벤트를 전송하도록 구성되어 있습니다. 자세한 내용은 [CloudWatch 로그로 이벤트 전송](#) (CloudTrail 문서) 을 참조하십시오.
6. 허브 및 스포크 계정은 AWS Organizations에서 관리해야 합니다.

아키텍처

다음 다이어그램은 구현의 구성 요소를 보여 줍니다.

1. IAM 루트 사용자 자격 증명을 사용하는 경우 CloudWatch 로그와 트레일에 각각 활동을 CloudTrail 기록하십시오.
2. 스포크 계정에서 EventBridge 규칙은 허브 계정의 중앙 [이벤트 버스로 이벤트를](#) 전송합니다.
3. 허브 계정에서 EventBridge 규칙은 Lambda 함수로 이벤트를 전송합니다.
4. Lambda 함수는 루트 사용자 활동을 알려주는 Amazon SNS 주제를 사용합니다.

도구

서비스

- [AWS는 AWS CloudFormation](#) 리소스를 설정하고, 빠르고 일관되게 프로비저닝하고, AWS 계정 및 지역 전반의 수명 주기 전반에 걸쳐 리소스를 관리할 수 있도록 지원합니다.
- [AWS는 AWS](#) 계정의 거버넌스, 규정 준수 및 운영 위험을 감사할 수 있도록 CloudTrail 지원합니다.
- [Amazon CloudWatch Logs](#)를 사용하면 모든 시스템, 애플리케이션 및 AWS 서비스의 로그를 중앙 집중화하여 로그를 모니터링하고 안전하게 보관할 수 있습니다.
- [EventBridgeAmazon](#)은 다양한 소스의 실시간 데이터와 애플리케이션을 연결하는 데 도움이 되는 서버리스 이벤트 버스 서비스입니다. AWS Lambda 함수, API 대상을 사용하는 HTTP 간접 호출 엔드 포인트 또는 다른 AWS 계정의 이벤트 버스를 예로 들 수 있습니다.

- [AWS Identity and Access Management\(IAM\)](#)를 사용하면 사용자에 대해 인증 및 권한 부여를 제어함으로써 AWS 리소스에 대한 액세스를 안전하게 관리할 수 있습니다.
- [AWS Lambda](#)는 서버를 프로비저닝하거나 관리할 필요 없이 코드를 실행하는 데 도움이 되는 컴퓨팅 서비스입니다. 필요할 때만 코드를 실행하며 자동으로 확장이 가능하므로 사용한 컴퓨팅 시간만큼만 비용을 지불합니다.
- [Amazon Simple Notification Service\(Amazon SNS\)](#)를 사용하면 웹 서버 및 이메일 주소를 포함하여 게시자와 클라이언트 간의 메시지 교환을 조정하고 관리할 수 있습니다.

기타 도구 및 서비스

- [Terraform](#)은 구성 파일 형태의 코드를 사용하여 클라우드 인프라 및 리소스를 프로비저닝하고 관리하는 CLI 애플리케이션입니다.

코드 리포지토리

이 패턴의 소스 코드와 템플릿은 [GitHub 리포지토리에서](#) 사용할 수 있습니다. 이 패턴은 두 개의 템플릿을 제공합니다.

- 허브 계정에 배포하는 리소스가 포함된 Terraform 템플릿
- 스포크 계정에서 스택 세트 인스턴스로 배포하는 CloudFormation 템플릿입니다.

리포지토리의 전체 구조는 다음과 같습니다.

```

.
|__README.md
|__spoke-stackset.yaml
|__hub.tf
|__root-activity-monitor-module
    |__main.tf # contains Terraform code to deploy resources in the Hub account
    |__iam     # contains IAM policies JSON files
        |__ lambda-assume-policy.json          # contains trust policy of the IAM role
used by the Lambda function
        |__ lambda-policy.json                # contains the IAM policy attached to
the IAM role used by the Lambda function
    |__outputs # contains Lambda function zip code

```

에픽 섹션에서는 템플릿 배포 step-by-step 지침을 제공합니다.

에픽

허브 계정에 리소스 배포

작업	설명	필요한 기술
<p>샘플 코드 리포지토리를 복제합니다.</p>	<ol style="list-style-type: none"> 1. AWS IAM 루트 사용자 활동 모니터 리포지토리를 엽니다. 2. 코드 탭의 파일 목록 위에서 코드를 선택한 다음 HTTPS URL을 복사합니다. 3. 명령줄 인터페이스에서 작업 디렉터리를 샘플 파일을 저장하고자 하는 위치로 변경합니다. 4. 다음 명령을 입력합니다. <pre>git clone <repoURL></pre>	<p>일반 AWS</p>
<p>TerraForm 템플릿을 업데이트합니다.</p>	<ol style="list-style-type: none"> 1. 조직 ID를 검색합니다. 지침을 보려면 관리 계정에서 조직 세부 정보 보기(AWS Organizations 설명서)를 참조하세요. 2. 복제된 리포지토리에서 <code>hub.tf</code>(를) 엽니다. 3. 환경에 적합한 값으로 다음을 업데이트하세요. <ul style="list-style-type: none"> • <code>OrganizationId</code> - 조직 ID를 추가합니다. • <code>SNSTopicName</code> - Amazon SNS 주제의 이름을 추가합니다. 	<p>일반 AWS</p>

작업	설명	필요한 기술
	<ul style="list-style-type: none"> • SNSSubscriptions - Amazon SNS 알림을 전송해야 하는 이메일을 추가합니다. • Region - 리소스를 배포하려는 AWS 리전 코드를 추가합니다. 예를 들어 eu-west-1 입니다. • Tags - 태그를 추가합니다. 태그에 대한 자세한 내용은 AWS 리소스 태깅(일반 참조 안내서)을 참조하세요. <p>4. hub.tf 파일을 저장하고 닫습니다.</p>	
<p>리소스를 AWS 허브 계정에 배포합니다.</p>	<ol style="list-style-type: none"> 1. Terraform 명령줄 인터페이스에서 복제된 리포지토리의 루트 폴더로 이동한 후 다음 명령을 입력합니다. <div data-bbox="630 1213 1029 1335" style="border: 1px solid #ccc; border-radius: 10px; padding: 10px; margin: 10px 0;"> <pre>terraform init && terraform plan</pre> </div> <ol style="list-style-type: none"> 2. 출력을 검토하고 설명된 리소스를 생성할지 확인합니다. 3. 다음 명령을 입력합니다. <div data-bbox="630 1570 1029 1654" style="border: 1px solid #ccc; border-radius: 10px; padding: 10px; margin: 10px 0;"> <pre>terraform apply</pre> </div> <ol style="list-style-type: none"> 4. 메시지가 표시되면 yes를 입력하여 배포를 확인합니다. 	<p>일반 AWS</p>

스포크 계정에 리소스 배포

작업	설명	필요한 기술
<p>템플릿을 배포하세요. CloudFormation</p>	<ol style="list-style-type: none"> 1. AWS Management Console에 로그인하고 CloudFormation 콘솔을 엽니다. 2. 탐색 창에서 StackSets를 선택합니다. 3. StackSets페이지 상단에서 [Create] 를 선택합니다 StackSet. 4. 권한에서 서비스 관리 권한을 선택합니다. CloudFormation AWS Organizations에서 관리하는 대상 계정에 배포하는 데 필요한 권한을 자동으로 구성합니다. 5. 사전 조건 - 템플릿 준비에서 템플릿 준비 완료를 선택합니다. 6. 템플릿 지정에서 템플릿 파일 업로드를 선택합니다. 7. 파일 선택을 선택한 다음 복제된 리포지토리에서 spoke-stackset.yaml 1 을 선택합니다. 8. 다음을 선택합니다. 9. StackSet 세부 정보 지정 페이지에서 스택 세트의 이름을 입력합니다. 10. 파라미터에서 허브 계정의 계정 ID를 입력하고 다음을 선택합니다. 	<p>일반 AWS</p>

작업	설명	필요한 기술
	<p>11.StackSet 옵션 구성 페이지의 태그에 태그를 추가합니다.</p> <p>12.실행 구성에서 비활성을 선택한 후 다음을 선택합니다.</p> <p>13.배포 옵션 설정 페이지에서 스택 세트를 배포할 조직 단위 및 리전을 지정하고 다음을 선택합니다.</p> <p>14.검토 페이지에서 AWS가 IAM 리소스를 생성할 CloudFormation 수 있음을 인정합니까를 선택한 다음 제출을 선택합니다. CloudFormation 스택 세트 배포를 시작합니다.</p> <p>자세한 내용 및 지침은 스택 세트 생성 (CloudFormation 설명서) 을 참조하십시오.</p>	

(선택 사항)알림 테스트

작업	설명	필요한 기술
<p>루트 사용자 보안 인증을 사용합니다.</p>	<ol style="list-style-type: none"> 루트 사용자 보안 인증 정보를 사용하여 스포크 계정 또는 허브 계정에 로그인합니다. 지정한 이메일 계정이 Amazon SNS 알림을 수신하는지 확인합니다. 	<p>일반 AWS</p>

관련 리소스

- [보안 모범 사례\(IAM 설명서\)](#)
- [CloudFormation 사용 StackSets \(설명서\)](#)
- [시작하기\(Terraform 설명서\)](#)

추가 정보

[GuardDutyAmazon](#)은 로그를 분석하고 처리하여 AWS 환경에서 예상치 못한 잠재적 무단 활동을 식별하는 지속적 보안 모니터링 서비스입니다. 이 솔루션의 대안으로 GuardDuty, 활성화한 경우 루트 사용자 자격 증명이 사용되면 알림을 받을 수 있습니다. GuardDuty 결과는 이며Policy:IAMUser/RootCredentialUsage, 기본 심각도는 낮음입니다. 자세한 내용은 [Amazon GuardDuty 조사 결과 관리](#)를 참조하십시오.

IAM 사용자 생성 시 알림 전송

작성자: Mansi Suratwala(AWS) 및 Sergiy Shevchenko(AWS)

환경: 프로덕션

기술: 보안, 자격 증명, 규정 준수, 분석, 인프라

워크로드: 기타 모든 워크로드

AWS 서비스: 아마존 SNS, AWS 자격 증명 및 액세스 관리, AWS Lambda, 아마존 CloudWatch

요약

Amazon Web Services (AWS)에서는 이 패턴을 사용하여 AWS Identity 및 Access Management (IAM) 사용자가 생성될 때 알림을 자동으로 수신하도록 AWS CloudFormation 템플릿을 배포할 수 있습니다.

IAM을 사용하면 AWS 서비스 및 리소스에 대한 액세스를 안전하게 관리할 수 있습니다. AWS 사용자 및 그룹을 만들고 관리하며 권한을 사용하여 AWS 리소스에 대한 해당 사용자 및 그룹의 액세스를 허용 및 거부할 수 있습니다.

CloudFormation 템플릿은 Amazon CloudWatch 이벤트 이벤트와 AWS Lambda 함수를 생성합니다. 이 이벤트는 CloudTrail AWS를 사용하여 AWS 계정에서 생성되는 IAM 사용자를 모니터링합니다. 사용자가 생성되면 CloudWatch Events 이벤트는 Lambda 함수를 시작합니다. 이 함수는 새 사용자 생성 이벤트를 알려주는 Amazon Simple Notification Service (Amazon SNS) 알림을 보냅니다.

사전 조건 및 제한 사항

사전 조건

- 활성 상태의 AWS 계정
- 생성 및 배포된 AWS CloudTrail 트레일

제한 사항

- AWS CloudFormation 템플릿은 CreateUser 배포용으로만 사용해야 합니다.

아키텍처

대상 기술 스택

- IAM
- AWS CloudTrail
- 아마존 CloudWatch 이벤트
- AWS Lambda
- Amazon Simple Storage Service (S3)
- Amazon SNS

대상 아키텍처

자동화 및 규모 조정

여러 AWS 지역 및 계정에 대해 AWS CloudFormation 템플릿을 여러 번 사용할 수 있습니다. 각 리전 또는 계정에서 한 번만 실행해야 합니다. 여러 계정에 배포를 자동화하려면 [AWS](#)를 사용하십시오 CloudFormation StackSets. CloudFormation 템플릿은 각 계정에 필요한 모든 리소스를 배포할 수 있습니다.

도구

도구

- [IAM](#)-AWS Identity 및 Access Management(IAM)는 AWS 리소스에 대한 사용자의 액세스를 안전하게 제어할 수 있게 지원하는 웹 서비스입니다. IAM을 사용하여 리소스를 사용하도록 인증(로그인) 및 권한 부여(권한 있음)된 대상을 제어합니다.
- [AWS CloudFormation](#) — AWS는 CloudFormation Amazon Web Services 리소스를 모델링하고 설정하도록 지원하므로 리소스를 관리하는 시간을 줄이고 AWS에서 실행되는 애플리케이션에 더 많은 시간을 집중할 수 있습니다. 원하는 모든 AWS 리소스를 설명하는 템플릿을 만들고, 해당 리소스를 프로비저닝하고 구성해 CloudFormation 줍니다.
- [AWS CloudTrail](#) — AWS는 AWS 계정의 거버넌스, 규정 준수, 운영 및 위험 감사를 관리할 수 있도록 CloudTrail 지원합니다. 사용자, 역할 또는 AWS 서비스가 수행한 작업은 에서 이벤트로 기록됩니다.

다 CloudTrail. 이벤트에는 AWS Management Console, Command Line Interface, AWS SDK 및 API 에서 수행되는 작업이 포함됩니다.

- [Amazon CloudWatch Events](#) — Amazon CloudWatch Events는 AWS 리소스의 변경 사항을 설명하는 시스템 이벤트 near-real-time 스트림을 제공합니다.
- [AWS Lambda](#) – AWS Lambda는 서버를 프로비저닝하거나 관리하지 않고도 코드를 실행하도록 지원하는 컴퓨팅 서비스입니다. Lambda는 필요 시에만 코드를 실행하며, 일일 몇 개의 요청에서 초당 수천 개의 요청까지 자동으로 규모를 조정합니다.
- [Amazon S3](#) – Amazon Simple Storage Service(S3)는 인터넷에 대한 스토리지입니다. Amazon S3를 사용하면 인터넷을 통해 언제 어디서든 원하는 양의 데이터를 저장하고 검색할 수 있습니다.
- [Amazon SNS](#)-Amazon Simple Notification Service(SNS)은 Lambda, HTTP, 이메일, 모바일 푸시 알림 및 모바일 문자 메시지(SMS)를 사용하여 메시지를 전송하는 관리형 서비스입니다.

코드

프로젝트의 .zip 파일은 첨부 파일로 제공됩니다.

에픽

Lambda 스크립트용 S3 버킷을 생성

작업	설명	필요한 기술
S3 버킷을 정의합니다.	Amazon S3 콘솔을 열고 S3 버킷을 선택하거나 생성합니다. 이 S3 버킷은 Lambda 코드 .zip 파일을 호스팅합니다. S3 버킷 이름에는 선행 슬래시를 포함할 수 없습니다.	클라우드 아키텍트

Lambda 코드를 S3 버킷에 업로드

작업	설명	필요한 기술
Lambda 코드를 업로드합니다.	첨부 파일 섹션에 제공된 Lambda 코드 .zip 파일을 사용	클라우드 아키텍트

작업	설명	필요한 기술
	자가 정의한 S3 버킷에 업로드합니다.	

CloudFormation 템플릿을 배포하십시오.

작업	설명	필요한 기술
CloudFormation 템플릿을 배포합니다.	CloudFormation 콘솔에서 이 패턴의 첨부 파일로 제공된 CloudFormation createIAM user.yaml 템플릿을 배포합니다. 다음 에픽에서 템플릿 파라미터에 대한 값을 입력합니다.	클라우드 아키텍트

CloudFormation 템플릿의 파라미터를 완성하세요.

작업	설명	필요한 기술
S3 버킷 이름을 제공합니다.	첫 번째 에픽에서 생성하거나 선택한 S3 버킷의 이름을 입력합니다.	클라우드 아키텍트
S3 키를 입력합니다.	S3 버킷의 Lambda 코드 .zip 파일 위치를 선행 슬래시 없이 입력합니다(예: <directory>/<file-name>.zip).	클라우드 아키텍트
이메일 주소를 입력합니다.	Amazon SNS 알림을 수신할 활성 이메일 주소를 입력합니다.	클라우드 아키텍트
로깅 수준을 정의합니다.	Lambda 함수의 로깅 수준 및 빈도를 정의합니다. Info는 애플리케이션 진행 상황에 대한	클라우드 아키텍트

작업	설명	필요한 기술
	자세한 정보 메시지를 지정합니다. Error는 애플리케이션을 계속 실행할 수 있게 해주는 오류 이벤트를 지정합니다. Warning은 잠재적으로 유해한 상황을 지정합니다.	

구독 확인

작업	설명	필요한 기술
구독을 확인합니다.	템플릿이 성공적으로 배포되면 입력한 이메일 주소로 구독 이메일 메시지가 전송됩니다. 알림을 받기 시작하려면 이 이메일 구독을 확인해야 합니다.	클라우드 아키텍트

관련 리소스

- [추적 생성](#)
- [S3 버킷 생성](#)
- [S3 버킷에 파일 업로드](#)
- [템플릿 배포 CloudFormation](#)
- [IAM 사용자 생성](#)
- [AWS를 사용하여 AWS API 호출에서 트리거되는 CloudWatch 이벤트 규칙 생성 CloudTrail](#)

첨부

이 문서와 관련된 추가 콘텐츠에 액세스하려면 [attachment.zip](#) 파일의 압축을 풉니다.

git-secrets를 사용하여 Git 리포지토리에서 민감한 정보 및 보안 문제 검사하기

작성자: Saurabh Singh (AWS)

환경: 프로덕션

기술: 보안, 자격 증명, 규정 준수

워크로드: 오픈소스

요약

이 패턴은 AWS Labs의 오픈 소스 [git-secret](#) 도구를 사용하여 Git 소스 리포지토리를 스캔하고 사용자 암호 또는 AWS 액세스 키와 같은 민감한 정보를 포함할 수 있거나 기타 보안 문제가 있는 코드를 찾는 방법을 설명합니다.

git-secrets는 커밋, 커밋 메시지, 병합을 검사하여 보안 암호와 같은 민감한 정보가 Git 리포지토리에 추가되지 않도록 합니다. 예를 들어 커밋, 커밋 메시지 또는 병합 기록의 커밋이 구성되거나 금지된 정규 표현식 패턴 중 하나와 일치하면 커밋이 거부됩니다.

사전 조건 및 제한 사항

사전 조건

- 활성 상태의 AWS 계정
- 보안 스캔이 필요한 Git 리포지토리
- Git 클라이언트(버전 2.37.1 이상)가 설치됨

아키텍처

대상 아키텍처

- Git
- git-secrets

도구

- [git-secrets](#)는 민감한 정보를 Git 리포지토리에 커밋하지 못하도록 하는 도구입니다.
- [Git](#)은 오픈 소스 분산 버전 제어 시스템입니다.

모범 사례

- 모든 수정 내용을 포함하여 항상 Git 저장소를 검사합니다.

```
git secrets --scan-history
```

에픽

Amazon EC2 인스턴스에 연결

작업	설명	필요한 기술
SSH를 사용하여 EC2 인스턴스에 연결합니다.	SSH와 키 쌍 파일을 사용하여 Amazon Elastic Compute Cloud(Amazon EC2) 인스턴스와 연결합니다. 로컬 시스템에서 리포지토리에 대해 검사하는 경우 이 단계를 건너뛸 수 있습니다.	일반 AWS

Git 설치

작업	설명	필요한 기술
Git을 설치합니다.	다음 명령을 사용하여 Git을 설치합니다. <pre>yum install git -y</pre>	일반 AWS

작업	설명	필요한 기술
	로컬 컴퓨터를 사용하는 경우, 특정 OS 버전용 Git 클라이언트를 설치할 수 있습니다. 자세한 내용은 Git 웹 사이트 를 참조하세요.	

소스 리포지토리를 복제하고 git-secrets를 설치합니다.

작업	설명	필요한 기술
Git 소스 리포지토리를 복제합니다.	검사하려는 Git 리포지토리를 복제하려면 홈 디렉터리에서 Git 복제 명령을 선택합니다.	일반 AWS
git 시크릿을 복제합니다.	<p>git-secrets Git 리포지토리를 복제합니다.</p> <pre data-bbox="597 1031 1026 1188">git clone https://github.com/awslabs/git-secrets.git</pre> <p>git-secrets 실행 시 Git이 git-secrets 을 픽업하도록 PATH 내부 어딘가에 배치합니다.</p>	일반 AWS
git-secrets을 설치합니다.	Unix 및 기타 변종(Linux/macOS)의 경우: (git-secrets 리포지토리에 제공된) Makefile의 install 대상을 사용하여 도구를 설치할 수 있습니다. PREFIX 및 MANPREFIX 변수	일반 AWS

작업	설명	필요한 기술
	<p>를 사용하여 설치 경로를 사용자 지정할 수 있습니다.</p> <pre>make install</pre> <p>Windows의 경우:</p> <p>git-secrets 리포지토리에 제공된 PowerShell <code>install.ps1</code> 스크립트를 실행합니다. 이 스크립트는 설치 파일을 설치 디렉터리(기본값은 <code>%USERPROFILE%/.git-secrets</code>)에 복사하고 디렉터리를 현재 사용자 PATH에게 추가합니다.</p> <pre>PS > ./install.ps1</pre> <p>Homebrew (macOS 사용자)의 경우:</p> <p>실행합니다.</p> <pre>brew install git-secrets</pre> <p>자세한 내용은 관련 리소스 섹션을 참조하세요.</p>	

git 코드 리포지토리 검사

작업	설명	필요한 기술
소스 리포지토리로 이동합니다.	<p>검사하려는 Git 리포지토리의 디렉터리로 전환합니다.</p> <pre>cd my-git-repository</pre>	일반 AWS
AWS 규칙 세트(Git 후크)를 등록합니다.	<p>커밋할 때마다 Git 리포지토리를 검사하도록 git-secrets 을 구성하려면 다음 명령을 실행합니다.</p> <pre>git secrets --register-aws</pre>	일반 AWS
리포지토리를 검사합니다.	<p>리포지토리 검사를 시작하려면 다음 명령을 실행합니다.</p> <pre>git secrets --scan</pre>	일반 AWS
출력 파일을 검토합니다.	<p>이 도구는 Git 리포지토리에서 취약점을 발견하면 출력 파일을 생성합니다. 예:</p> <pre>example.sh:4:AWS_SECRET_ACCESS_KEY = ***** [ERROR] Matched one or more prohibited patterns Possible mitigations: - Mark false positives as allowed using: git config --add secrets.allowed ...</pre>	일반 AWS

작업	설명	필요한 기술
	<ul style="list-style-type: none"> - Mark false positives as allowed by adding regular expressions to .gitallowed at repository's root directory - List your configured patterns: <code>git config --get-all secrets.patterns</code> - List your configured allowed patterns: <code>git config --get-all secrets.allowed</code> - List your configured allowed patterns in .gitallowed at repository's root directory - Use <code>--no-verify</code> if this is a one-time false positive 	

관련 리소스

- [AWS 서비스를 사용한 Git 웹후크 \(AWS Quick Start\)](#)
- [git-secrets 도구](#)
- [Git 리포지토리를 AWS로 마이그레이션 \(AWS 실습 자습서\)](#)
- [AWS CodeCommit API 레퍼런스](#)

AWS Network Firewall에서 Slack 채널로 알림 전송

작성자: Venki Srivatsav(AWS) 및 Aromal Raj Jayarajan(AWS)

코드 리포지토리: NfwSlackIntegration	환경: PoC 또는 파일럿	기술: 보안, 자격 증명, 규정 준수, 네트워킹
AWS 서비스: AWS Lambda, AWS Network Firewall, Amazon S3		

요약

이 패턴은 분산 배포 모델로 Amazon Web Services(AWS) Network Firewall을 사용하여 방화벽을 배포하는 방법과 AWS Network Firewall에서 생성된 알림을 구성 가능한 Slack 채널로 전파하는 방법을 설명합니다.

지불 카드 산업 데이터 보안 표준(PCI DSS) 같은 규정 준수 표준은 고객 데이터를 보호하기 위해 방화벽을 설치하고 유지 관리할 것을 요구합니다. AWS 클라우드에서 Virtual Private Cloud(VPC)는 이러한 규정 준수 요구 사항의 맥락에서 물리적 네트워크와 동일하게 간주됩니다. Network Firewall을 사용하여 VPC 간 네트워크 트래픽을 모니터링하고 규정 준수 표준이 적용되는 VPC에서 실행되는 워크로드를 보호할 수 있습니다. Network Firewall은 동일한 계정 내 다른 VPC의 무단 액세스 감지 시 액세스를 차단하거나 알림을 생성합니다. 하지만 Network Firewall은 알림을 전달하는 대상 수를 제한적으로 지원합니다. 이러한 대상에는 Amazon Simple Storage Service (Amazon S3) 버킷, CloudWatch Amazon 로그 그룹 및 Amazon Data Firehose 전송 스트림이 포함됩니다. 이러한 알림에 대한 추가 조치를 취하려면 Amazon Athena 또는 Amazon Kinesis를 사용한 오프라인 분석이 필요합니다.

이 패턴은 Network Firewall에서 생성된 알림을 구성 가능한 Slack 채널로 전파하여 추가 조치를 거의 실시간으로 수행하는 방법을 제공합니다. 또한 Jira 및 이메일과 같은 PagerDuty 다른 알림 메커니즘으로 기능을 확장할 수 있습니다. (이러한 사용자 지정은 이 패턴의 범위를 벗어납니다.)

사전 조건 및 제한 사항

사전 조건

- Slack 채널(Slack [도움말 센터에서 시작하기](#) 참조)
- 채널에 메시지를 보내는 데 필요한 권한

- API 토큰이 포함된 Slack 엔드포인트 URL([앱을 선택하고](#) 수신 웹후크를 선택하면 해당 URL이 표시됩니다. 자세한 내용은 Slack API 설명서의 [수신 웹후크 만들기](#) 참조)
- 워크로드 서브넷의 Amazon Elastic Compute Cloud(Amazon EC2) 테스트 인스턴스
- Network Firewall의 규칙 테스트
- 테스트 규칙을 트리거하기 위한 실제 또는 시뮬레이션된 트래픽
- 배포할 소스 파일을 보관하는 S3 버킷

제한 사항

- 현재 이 솔루션은 단일 CIDR (클래스 없는 도메인 간 라우팅) 범위만 소스 및 대상 IP에 대한 필터로 지원합니다.

아키텍처

대상 기술 스택

- VPC 한 개
- 서브넷 네 개(방화벽용 둘, 워크로드용 둘)
- 인터넷 게이트웨이
- 규칙이 포함된 라우팅 테이블 네 개
- 알림 대상으로 사용되는 S3 버킷, Lambda 함수를 실행하기 위한 버킷 정책 및 이벤트 설정으로 구성됨
- Slack 알림 전송을 위한 실행 역할의 Lambda 함수(Slack 알림 전송)
- Slack URL 저장을 위한 AWS Secrets Manager 보안 암호
- 알림 구성이 포함된 네트워크 방화벽
- Slack 채널

[Slack 채널을 제외한 모든 구성 요소는 이 패턴과 함께 제공되는 CloudFormation 템플릿 및 Lambda 함수를 통해 프로비저닝됩니다 \(코드 섹션 참조\).](#)

대상 아키텍처

이 패턴은 Slack 통합이 포함된 분산형 네트워크 방화벽을 설정합니다. 이 아키텍처는 VPC와 가용 영역 두 개로 구성되어 있습니다. VPC는 보호된 서브넷 두 개와 네트워크 방화벽 엔드포인트가 있는 방화벽 서브넷 두 개를 포함합니다. [방화벽 정책 및 규칙을 생성하여](#) 보호된 서브넷으로 들어오고 나가는

모든 트래픽을 모니터링할 수 있습니다. 네트워크 방화벽은 S3 버킷에서 모든 알림을 배치하도록 구성되어 있습니다. 이 S3 버킷은 put 이벤트를 수신할 때 Lambda 함수를 직접적으로 호출하도록 구성되어 있습니다. Lambda 함수는 Secrets Manager에서 구성된 슬랙 URL을 가져와 슬랙 워크스페이스로 알림 메시지를 보냅니다.

이 아키텍처에 대한 자세한 내용은 AWS 블로그 게시물 [AWS Network Firewall의 배포 모델](#)을 참조하세요.

도구

서비스

- [AWS Network Firewall](#)은 AWS Cloud를 위한 상태 저장형, 관리형 네트워크 방화벽이자, 침입 탐지 및 방지 서비스입니다. Network Firewall을 이용하면 VPC 경계에서 트래픽을 필터링하고 AWS에서 워크로드를 보호할 수 있습니다.
- [AWS Secrets Manager](#)는 자격 증명 저장 및 검색을 위한 서비스입니다. 코드의 암호를 포함해 하드 코딩된 자격 증명을 Secrets Manager에서 프로그래밍 방식으로 보안 암호를 검색하도록 하는 API 직접 호출로 바꿀 수 있습니다. 이 패턴은 Secrets Manager를 사용하여 Slack URL을 저장합니다.
- [Amazon Simple Storage Service\(S3\)](#)는 객체 스토리지 서비스입니다. Amazon S3를 사용하면 인터넷을 통해 언제 어디서든 원하는 양의 데이터를 저장하고 검색할 수 있습니다. 이 패턴은 Amazon S3를 사용하여 Lambda 함수의 CloudFormation 템플릿과 Python 스크립트를 저장합니다. 또한 S3 버킷을 네트워크 방화벽 알림 대상으로 사용합니다.
- [AWS는 AWS CloudFormation](#) 리소스를 모델링 및 설정하고, 빠르고 일관되게 프로비저닝하고, 수명 주기 전반에 걸쳐 관리할 수 있도록 지원합니다. 템플릿을 사용하여 리소스와 해당 종속성을 설명하고, 리소스를 개별적으로 관리하는 대신 스택으로 함께 시작하고 구성할 수 있습니다. 이 패턴은 AWS를 CloudFormation 사용하여 Firewall Manager용 분산 아키텍처를 자동으로 배포합니다.

code

이 패턴의 코드는 [Network Firewall Slack 통합](#) 리포지토리의 에서 GitHub 확인할 수 있습니다. 리포지토리 src 폴더에는 다음을 볼 수 있습니다.

- YAML 형식의 CloudFormation 파일 세트. 이러한 템플릿을 사용하여 이 패턴의 구성 요소를 프로비저닝할 수 있습니다.
- Lambda 함수를 생성하기 위한 Python 소스 파일(slack-lambda.py).
- .zip 아카이브 배포 패키지(slack-lambda.py.zip)를 통해 Lambda 함수 코드를 업로드합니다.

이러한 파일을 사용하려면 다음 섹션의 지침을 따르세요.

에픽

S3 버킷 선택

작업	설명	필요한 기술
S3 버킷을 생성합니다.	<ol style="list-style-type: none"> 1. AWS Management Console에 로그인하고 https://console.aws.amazon.com/s3/에서 Amazon S3 콘솔을 엽니다. 2. 코드를 호스팅할 S3 버킷을 선택하거나 생성합니다. S3 버킷 이름은 전역 수준에서 고유하며, 네임스페이스는 모든 AWS 계정이 공유합니다. S3 버킷 이름에는 선행 슬래시를 포함할 수 없습니다. 접두사를 사용하여 이 패턴의 코드를 구성하는 것이 좋습니다. <p>자세한 내용은 Amazon S3 설명서의 버킷 생성을 참조하세요.</p>	앱 개발자, 앱 소유자, 클라우드 관리자
CloudFormation 템플릿과 Lambda 코드를 업로드합니다.	<ol style="list-style-type: none"> 1. 이 패턴에 대한 다음 파일을 GitHub 리포지토리에서 다운로드하십시오. <ul style="list-style-type: none"> • base.yml • igw-ingress-route.yml • slack-lambda.py • slackLambda.yml 	앱 개발자, 앱 소유자, 클라우드 관리자

작업	설명	필요한 기술
	<ul style="list-style-type: none"> decentralized-deployment.yml protected-subnet-route.yml slack-lambda.py.zip <p>2. 파일을 방금 생성한 S3 버킷으로 업로드합니다.</p>	

CloudFormation 템플릿을 배포하십시오.

작업	설명	필요한 기술
CloudFormation 템플릿을 실행합니다.	<p>S3 버킷과 동일한 AWS 리전에서 AWS CloudFormation 콘솔을 열고 템플릿을 base.yml 배포합니다. 이 템플릿은 Slack 채널로 알림을 전송하는 데 필요한 AWS 리소스 및 Lambda 함수를 생성합니다.</p> <p>CloudFormation 템플릿 배포에 대한 자세한 내용은 CloudFormation 설명서의 AWS CloudFormation 콘솔에 스택 생성을 참조하십시오.</p>	앱 개발자, 앱 소유자, 클라우드 관리자
템플릿에서 파라미터를 작성합니다.	<p>스택 이름을 지정하고 파라미터 값을 구성합니다. 파라미터 목록, 설명 및 기본값은 추가 정보 섹션의 CloudFormation 파라미터를 참조하십시오.</p>	앱 개발자, 앱 소유자, 클라우드 관리자

작업	설명	필요한 기술
스택을 생성합니다.	<ol style="list-style-type: none"> 스택 세부 정보를 검토하고 환경 요구 사항에 따라 값을 업데이트합니다. 스택 생성을 선택하여 템플릿을 배포합니다. 	앱 개발자, 앱 소유자, 클라우드 관리자

솔루션 확인

작업	설명	필요한 기술
배포를 테스트합니다.	<p>AWS CloudFormation 콘솔 또는 AWS 명령줄 인터페이스 (AWS CLI) 를 사용하여 대상 기술 스택 섹션에 나열된 리소스가 생성되었는지 확인하십시오.</p> <p>CloudFormation 템플릿이 성공적으로 배포되지 않는 경우, pAvailabilityZone1 및 pAvailabilityZone2 파라미터에 대해 제공한 값을 확인하십시오. 이는 솔루션을 배포하려는 AWS 리전에 적합해야 합니다. 리전의 가용 영역 목록은 Amazon EC2 설명서의 리전 및 영역을 참조하세요.</p>	앱 개발자, 앱 소유자, 클라우드 관리자
기능을 테스트합니다.	<ol style="list-style-type: none"> https://console.aws.amazon.com/ec2/에서 Amazon EC2 콘솔을 엽니다. 보호된 서브넷 중 하나에 EC2 인스턴스를 생성합니다. HTTPS 서버로 사용할 	앱 개발자, 앱 소유자, 클라우드 관리자

작업	설명	필요한 기술
	<p>Amazon Linux 2 AMI(HVM)를 선택합니다. 지침은 Amazon EC2 설명서의 인스턴스 시작을 참조하세요.</p> <p>3. 다음의 사용자 데이터를 사용하여 EC2 인스턴스에 웹 서버를 설치합니다.</p> <pre data-bbox="597 604 1026 995">#!/bin/bash yum install httpd -y systemctl start httpd systemctl stop firewalld cd /var/www/html echo "Hello!! this is a NFW alert test page, 200 OK" > index.html</pre> <p>4. 다음의 네트워크 방화벽 규칙을 생성하세요.</p> <p>스테이트리스 규칙:</p> <pre data-bbox="597 1234 1026 1474">Source: 0.0.0.0/0 Destination 10.0.3.65 /32 (private IP of the EC2 instance) Action: Forward</pre> <p>스테이트풀 규칙:</p> <pre data-bbox="597 1583 1026 1822">Protocol: HTTP Source ip/port: Any / Any Destination ip/port: Any /Any</pre>	

작업	설명	필요한 기술
	<p>5. 3단계에서 생성한 웹 서버의 공개 IP를 가져옵니다.</p> <p>6. 브라우저에서 공개 IP에 액세스합니다. 브라우저에 다음 메시지가 표시됩니다.</p> <div data-bbox="597 506 1027 663" style="border: 1px solid #ccc; border-radius: 10px; padding: 10px; margin: 10px 0;"> <pre>Hello!! this is a NFW alert test page, 200 OK</pre> </div> <p>그러면 Slack 채널에서도 알림을 받게 됩니다. 메시지 크기에 따라 알림이 지연될 수 있습니다. 테스트 목적으로 너무 좁지 않은 CIDR 필터를 제공하는 것이 좋습니다(예를 들어, CIDR 값이 /32이면 너무 좁은 것으로 간주되고 /8은 너무 넓은 것으로 간주됨). 자세한 내용은 추가 정보의 필터 동작 섹션을 참조하세요.</p>	

관련 리소스

- [AWS Network Firewall의 배포 모델](#)(AWS 블로그 게시물)
- [AWS Network Firewall 정책](#)(AWS 설명서)
- [네트워크 방화벽 슬랙 통합](#) (GitHub 리포지토리)
- [Slack 워크스페이스 생성](#)(Slack 도움말 센터)

추가 정보

CloudFormation 파라미터

파라미터	설명	기본값 또는 샘플 값
pVpcName	생성할 VPC의 이름입니다.	점검
pVpcCidr	VPC가 생성할 CIDR 범위입니다.	10.0.0.0/16
pVpcInstanceTenancy	EC2 인스턴스가 물리적 하드웨어에 분산되는 방식. 옵션은 default (공용 테넌시) 또는 dedicated (단일 테넌시)입니다.	기본값
pAvailabilityZone1	인프라의 첫 번째 가용 영역.	us-east-2a
pAvailabilityZone2	인프라의 두 번째 가용 영역.	us-east-2b
pNetworkFirewallSubnet1Cidr	첫 번째 방화벽 서브넷의 CIDR 범위(최소 /28).	10.0.1.0/24
pNetworkFirewallSubnet2Cidr	두 번째 방화벽 서브넷의 CIDR 범위(최소 /28).	10.0.2.0/24
pProtectedSubnet1Cidr	첫 번째 보호(워크로드) 서브넷의 CIDR 범위.	10.0.3.0/24
pProtectedSubnet2Cidr	두 번째 보호(워크로드) 서브넷의 CIDR 범위.	10.0.4.0/24
pS3BucketName	Lambda 소스 코드를 업로드한 기존 S3 버킷의 이름.	us-w2- yourname-lambda-functions
pS3KeyPrefix	Lambda 소스 코드를 업로드한 S3 버킷의 접두사.	aod-test
pAWSSecretName4Slack	Slack URL을 포함하는 보안 암호의 이름.	SlackEndpoint-Cfn
pSlackChannelName	생성한 Slack 채널의 이름.	somename-notifications

pSlackUserName	Slack 사용자 이름.	Slack 사용자
pSecretKey	아무 키나 사용할 수 있습니다. 기본값을 그대로 사용하는 것이 좋습니다.	webhookUrl
pWebHookUrl	Slack URL의 값.	https://hooks.slack.com/services/T????9T??/A031885JRM7/9D4Y??????
pAlertS3Bucket	네트워크 방화벽 알림 대상으로 사용할 S3 버킷의 이름. 이 버킷은 자동으로 생성됩니다.	us-w2- yourname-security-aod-alerts
pSecretTagName	보안 암호의 태그 이름.	AppName
pSecretTagValue	지정된 태그 이름의 태그 값.	LambdaSlackIntegration
pdestCidr	대상 CIDR 범위를 위한 필터. 자세한 정보는 다음 섹션인 필터 동작을 참조하세요.	10.0.0.0/16
pdestCondition	대상 일치 항목을 제외할지 또는 포함할지 여부를 나타내는 플래그. 자세한 정보는 다음 섹션을 참조하세요. 유효 값은 include 및 exclude입니다.	포함
psrcCidr	알림을 보낼 소스 CIDR 범위의 필터. 자세한 정보는 다음 섹션을 참조하세요.	118.2.0.0/16
psrcCondition	소스 일치 항목을 제외하거나 포함하는 플래그. 자세한 정보는 다음 섹션을 참조하세요.	포함

필터 동작

AWS Lambda에서 필터를 구성하지 않은 경우 생성된 모든 알림이 Slack 채널로 전송됩니다. 생성된 알림의 소스 및 대상 IP는 템플릿을 배포할 때 구성한 CIDR 범위와 일치합니다. CloudFormation 이때 일치하는 부분이 발견되면 조건이 적용됩니다. 소스 또는 대상이 구성된 CIDR 범위에 속하고 둘 중 하나 이상이 조건 include에 맞게 구성된 경우 알림이 생성됩니다. 다음 표에는 CIDR 값, 조건 및 결과의 예가 나와 있습니다.

	구성된 CIDR	알림 IP	구성됨	Alert
소스	10.0.0.0/16	10.0.0.25	포함	예
대상	100.0.0.0/16	202.0.0.13	포함	
	구성된 CIDR	알림 IP	구성됨	Alert
소스	10.0.0.0/16	10.0.0.25	제외	아니요
대상	100.0.0.0/16	202.0.0.13	포함	
	구성된 CIDR	알림 IP	구성됨	Alert
소스	10.0.0.0/16	10.0.0.25	포함	예
대상	100.0.0.0/16	100.0.0.13	포함	
	구성된 CIDR	알림 IP	구성됨	Alert
소스	10.0.0.0/16	90.0.0.25	포함	예
대상	Null	202.0.0.13	포함	
	구성된 CIDR	알림 IP	구성됨	Alert
소스	10.0.0.0/16	90.0.0.25	포함	아니요

대상	100.0.0.0/16	202.0.0.13	포함
----	--------------	------------	----

AWS 프라이빗 CA와 AWS RAM을 사용하여 프라이빗 인증서 관리를 간소화합니다.

작성자: Everett Hinckley(AWS) 및 Vivek Goyal(AWS)

코드 리포지토리: [ACMPCA 계층](#)

환경: 프로덕션

기술: 보안, 자격 증명, 규정 준수, 인프라, 마이그레이션

AWS 서비스: AWS Certificate Manager(ACM), AWS Organizations, AWS RAM

요약

AWS Private Certificate Authority(AWS 프라이빗 CA)를 사용하여 내부 리소스를 인증하고 컴퓨터 코드에 서명하기 위한 프라이빗 인증서를 발급할 수 있습니다. 이 패턴은 다단계 CA 계층 구조의 신속한 배포와 일관된 프로비저닝 경험을 위한 AWS CloudFormation 템플릿을 제공합니다. 선택적으로, AWS Resource Access Manager(AWS RAM)를 사용하여 조직 또는 AWS Organizations의 조직 단위(OU) 내에서 CA를 안전하게 공유하고, AWS RAM을 사용하여 권한을 관리하면서 CA를 중앙 집중화할 수 있습니다. 모든 계정에 프라이빗 CA가 필요하지 않으므로 이 방법을 사용하면 비용을 절감할 수 있습니다. 또한 Amazon Simple Storage Service(S3)를 사용하여 인증서 취소 목록(CRL) 및 액세스 로그를 저장할 수 있습니다.

이 구현에서는 다음과 같은 기능 및 이점을 제공합니다.

- AWS 프라이빗 CA를 사용하여 프라이빗 CA 계층 구조의 관리를 중앙 집중화하고 간소화합니다.
- 인증서와 키를 AWS 및 온프레미스의 고객이 관리하는 디바이스로 내보냅니다.
- 신속한 배포와 일관된 프로비저닝 경험을 위해 AWS CloudFormation 템플릿을 사용합니다.
- 1, 2, 3 또는 4개의 하위 CA 계층이 있는 프라이빗 루트 CA를 생성합니다.
- 선택적으로, AWS RAM을 사용하여 최종 엔티티의 하위 CA를 조직 또는 OU 수준의 다른 계정과 공유할 수 있습니다.
- AWS RAM을 사용하면 모든 계정에 프라이빗 CA가 필요하지 않으므로 비용을 절감할 수 있습니다.
- CRL용 선택적 S3 버킷을 생성합니다.
- CRL 액세스 로그용 선택적 S3 버킷을 생성합니다.

사전 조건 및 제한 사항

사전 조건

AWS Organizations 구조 내에서 CA를 공유하려면 다음을 식별하거나 설정합니다.

- CA 계층 구조를 생성하고 공유하기 위한 보안 계정.
- 테스트를 위한 별도의 OU 또는 계정.
- AWS Organizations 관리 계정 내에서 공유가 활성화됩니다. 자세한 내용은 AWS RAM 설명서의 [AWS Organizations 내에서 리소스 공유 활성화](#)를 참조하세요.

제한 사항

- CA는 리전의 리소스입니다. 모든 CA는 단일 AWS 계정과 단일 AWS 리전에 위치합니다.
- 사용자가 생성한 인증서 및 키는 지원되지 않습니다. 이 사용 사례에서는 외부 루트 CA를 사용하도록 이 솔루션을 사용자 지정하는 것이 좋습니다.
- 퍼블릭 CRL 버킷은 지원되지 않습니다. CRL을 비공개로 유지하는 것이 좋습니다. CRL에 대한 인터넷 액세스가 필요한 경우, AWS 사설 CA 설명서의 [S3 블록 퍼블릭 액세스 \(BPA\) 기능 활성화에서 CloudFront Amazon](#)을 사용하여 CRL을 제공하는 방법에 대한 섹션을 참조하십시오.
- 이 패턴은 단일 리전 접근 방식을 구현합니다. 멀티 리전 인증 기관이 필요한 경우 보조 AWS 리전 또는 온프레미스에서 하위 인증서를 구현할 수 있습니다. 해당 구현은 특정 사용 사례, 워크로드 볼륨, 종속성, 요구 사항에 따라 달라지기 때문에 복잡성이 이 패턴의 범위를 벗어납니다.

아키텍처

대상 기술 스택

- AWS 프라이빗 CA
- AWS RAM
- Amazon S3
- AWS Organizations
- AWS CloudFormation

대상 아키텍처

이 패턴은 AWS Organizations에 공유할 수 있는 두 가지 옵션을 제공합니다.

옵션 1 – 조직 수준에서 공유를 생성합니다. 조직의 모든 계정은 다음 다이어그램과 같이 공유 CA를 사용하여 프라이빗 인증서를 발급할 수 있습니다.

옵션 2 – 조직 단위(OU) 수준에서 공유를 생성합니다. 지정된 OU의 계정만 공유 CA를 사용하여 프라이빗 인증서를 발급할 수 있습니다. 예를 들어 다음 다이어그램에서 샌드박스 OU 수준에서 공유를 생성하면 개발자 1과 개발자 2 모두 공유 CA를 사용하여 프라이빗 인증서를 발급할 수 있습니다.

도구

서비스

- [AWS 프라이빗 CA](#)–AWS 프라이빗 인증 기관(AWS 프라이빗 CA)은 프라이빗 디지털 인증서를 발급 및 취소하기 위한 호스팅된 프라이빗 CA 서비스입니다. 온프레미스 CA를 운영하는 데 드는 투자 및 유지 관리 비용 없이 루트 및 하위 CA를 비롯한 프라이빗 CA 계층을 생성할 수 있습니다.
- [AWS RAM](#)–AWS Resource Access Manager(AWS RAM)를 사용하면 AWS 계정 전체와 AWS Organizations의 조직 또는 OU 내에서 리소스를 안전하게 공유할 수 있습니다. 다중 계정 환경에서 운영 오버헤드를 줄이려면 리소스를 생성하고 AWS RAM을 사용하여 계정 간에 해당 리소스를 공유할 수 있습니다.
- [AWS Organizations](#)–AWS Organizations는 사용자가 생성해 중앙 관리하는 단일 조직으로 여러 AWS 계정을 통합할 수 있는 계정 관리 서비스입니다.
- [Amazon S3](#)–Amazon Simple Storage Service(S3)는 객체 스토리지 서비스입니다. Amazon S3를 사용하면 인터넷을 통해 언제 어디서든 원하는 양의 데이터를 저장하고 검색할 수 있습니다. 이 패턴에서는 Amazon S3를 사용하여 인증서 취소 목록(CRL)과 액세스 로그를 저장합니다.
- [AWS CloudFormation](#) — AWS는 AWS CloudFormation 리소스를 모델링 및 설정하고, 빠르고 일관되게 프로비저닝하고, 수명 주기 전반에 걸쳐 관리할 수 있도록 지원합니다. 템플릿을 사용하여 리소스와 해당 종속성을 설명하고, 리소스를 개별적으로 관리하는 대신 스택으로 함께 시작하고 구성할 수 있습니다. 이 패턴은 AWS를 CloudFormation 사용하여 다단계 CA 계층 구조를 자동으로 배포합니다.

code

이 패턴의 소스 코드는 [AWS 사설 CA 계층 구조](#) 리포지토리에서 확인할 수 있습니다. GitHub 리포지토리에는 다음이 포함됩니다.

- AWS CloudFormation ACMPCA-RootCASubCA.yaml 템플릿입니다. 이 템플릿을 사용하여 이 구현의 CA 계층 구조를 배포할 수 있습니다.
- 인증서 요청, 내보내기, 설명 및 삭제와 같은 사용 사례에 대해 파일을 테스트합니다.

이러한 파일을 사용하려면 에픽 섹션의 지침을 따르세요.

에픽

CA 계층 구조 설계

작업	설명	필요한 기술
인증서 주체 정보를 수집합니다.	인증서 소유자에 대한 인증서 주체 정보(조직 이름, 조직 단위, 국가, 주, 지역, 일반 이름)를 수집합니다.	클라우드 아키텍트, 보안 아키텍트, PKI 엔지니어
AWS Organizations에 대한 선택적 정보를 수집합니다.	CA가 AWS Organizations 구조의 일부이고 해당 구조 내에서 CA 계층 구조를 공유하려는 경우 관리 계정 번호, 조직 ID 및 선택 사항으로 OU ID(CA 계층 구조를 특정 OU와만 공유하려는 경우)를 수집합니다. 또한 해당하는 경우, CA를 공유하려는 AWS Organizations 계정 또는 OU를 결정합니다.	클라우드 아키텍트, 보안 아키텍트, PKI 엔지니어
CA 계층 구조를 설계합니다.	루트 및 하위 CA를 보관할 계정을 결정합니다. 루트 인증서와 최종 엔티티 인증서 간에 계층 구조가 필요로 하는 하위 수준 수를 결정합니다. 자세한 내용은 AWS 프라이빗 CA 설명서의 CA 계층 구조 설계 를 참조하세요.	클라우드 아키텍트, 보안 아키텍트, PKI 엔지니어

작업	설명	필요한 기술
CA 계층 구조의 이름 지정 및 태그 지정 규칙을 결정합니다.	AWS 리소스의 이름(루트 CA와 각 하위 CA)을 결정합니다. 각 CA에 할당해야 하는 태그를 결정합니다.	클라우드 아키텍트, 보안 아키텍트, PKI 엔지니어
필요한 암호화 및 서명 알고리즘을 결정합니다.	<p>다음을 결정합니다.</p> <ul style="list-style-type: none"> • CA가 인증서를 발급할 때 사용하는 공개 키에 대한 조직의 암호화 알고리즘 요구 사항. 기본값은 RSA_2048입니다. • CA가 인증서 서명에 사용하는 키 알고리즘. 기본값은 SHA256WITHRSA입니다. 	클라우드 아키텍트, 보안 아키텍트, PKI 엔지니어
CA 계층 구조에 대한 인증서 취소 요구 사항을 결정합니다.	인증서 취소 기능이 필요한 경우 인증서 취소 목록(CRL)이 포함된 S3 버킷의 명명 규칙을 설정합니다.	클라우드 아키텍트, 보안 아키텍트, PKI 엔지니어
CA 계층 구조의 로깅 요구 사항을 결정합니다.	액세스 로깅 기능이 필요한 경우 액세스 로그가 포함된 S3 버킷의 명명 규칙을 설정합니다.	클라우드 아키텍트, 보안 아키텍트, PKI 엔지니어

작업	설명	필요한 기술
인증서 만료 기간을 결정합니다.	다음에 대한 만료 날짜를 결정합니다. 루트 인증서(기본값은 10년), 최종 엔티티 인증서(기본값은 13개월) 및 하위 CA 인증서(기본값은 3년). 하위 CA 인증서는 계층 구조의 상위 수준에 있는 CA 인증서보다 먼저 만료되어야 합니다. 자세한 내용은 AWS 프라이빗 CA 설명서의 프라이빗 CA 수명 주기 관리 를 참조하세요.	클라우드 아키텍트, 보안 아키텍트, PKI 엔지니어

CA 계층 구조 배포

작업	설명	필요한 기술
사전 조건 완료	이 패턴의 사전 요구 사항 섹션에 있는 단계를 완료합니다.	클라우드 관리자, 보안 엔지니어, PKI 엔지니어
다양한 페르소나에 대한 CA 역할을 생성합니다.	<ol style="list-style-type: none"> 루트 CA 관리자, 하위 CA 관리자 등과 같은 CA 계층 구조의 다양한 수준을 관리하는 데 필요한 AWS IAM ID 센터 (AWS Single Sign-On의 후속) 에서 AWS ID 및 액세스 관리 (IAM) 역할 또는 사용자의 유형을 결정합니다. CertificateConsumer 직무를 구분하는 데 필요한 정책의 세분성을 결정합니다. CA 계층 구조가 있는 계정의 IAM Identity Center에서 필 	클라우드 관리자, 보안 엔지니어, PKI 엔지니어

작업	설명	필요한 기술
	요한 IAM 역할 또는 사용자를 생성합니다.	
스택을 CloudFormation 배포하십시오.	<ol style="list-style-type: none"> 이 패턴에 대한 GitHub 리포지토리에서 AWSPCA - rootcaSubca.yaml 템플릿을 다운로드하십시오. AWS CloudFormation 콘솔 또는 AWS 명령줄 인터페이스 (AWS CLI) 에서 템플릿을 배포합니다. 자세한 내용은 설명서의 스택 사용을 참조하십시오. CloudFormation 템플릿에서 조직 이름, OU 이름, 키 알고리즘, 서명 알고리즘 및 기타 옵션을 포함하여 파라미터를 완성합니다. 	클라우드 관리자, 보안 엔지니어, PKI 엔지니어

작업	설명	필요한 기술
<p>사용자 관리 리소스에서 사용하는 인증서를 업데이트하기 위한 솔루션을 설계합니다.</p>	<p>Elastic Load Balancing과 같은 통합 AWS 서비스의 리소스는 만료 전에 인증서를 자동으로 업데이트합니다. 하지만 Amazon Elastic Compute Cloud(Amazon EC2) 인스턴스에서 실행되는 웹 서버와 같이 사용자가 관리하는 리소스는 다른 메커니즘이 필요합니다.</p> <ol style="list-style-type: none"> 1. 사용자가 관리하는 리소스 중 프라이빗 CA의 최종 엔티티 인증서가 필요한 리소스를 결정합니다. 2. 사용자가 관리하는 리소스 및 인증서의 만료에 대한 알림을 받을 프로세스를 계획합니다. 예를 들어 다음을 참조하세요. <ul style="list-style-type: none"> • AWS Config 관리형 규칙 사용 • CloudWatch 아마존과 아마존 사용 EventBridge 3. 사용자 지정 스크립트를 작성하여 사용자가 관리하는 리소스의 인증서를 업데이트하고 이를 AWS 서비스와 통합하여 업데이트를 자동화합니다. 통합 AWS 서비스에 대한 자세한 내용은 ACM 설명서의 AWS Certificate Manager와 통합된 서비스를 참조하세요. 	<p>클라우드 관리자, 보안 엔지니어, PKI 엔지니어</p>

CA 계층 구조 검증 및 문서화

작업	설명	필요한 기술
선택적 AWS RAM 공유를 검증합니다.	CA 계층 구조가 AWS Organizations의 다른 계정과 공유되는 경우, AWS Management Console에서 해당 계정 중 하나에 로그인하고, AWS 프라이빗 CA 콘솔로 이동하여 새로 만든 CA가 이 계정과 공유되는지 확인합니다. 계층 구조에서 가장 낮은 수준의 CA만 표시되는데, 이는 최종 엔티티 인증서를 생성하는 CA이기 때문입니다. CA가 공유되는 계정의 샘플링을 위해 이 단계를 반복합니다.	클라우드 관리자, 보안 엔지니어, PKI 엔지니어
인증서 수명 주기 테스트로 CA 계층 구조를 검증합니다.	이 패턴의 GitHub 리포지토리 에서 라이프사이클 테스트를 찾으십시오. AWS CLI에서 테스트를 실행하여 인증서를 요청하고, 인증서를 내보내고, 인증서를 설명하고, 인증서를 삭제합니다.	클라우드 관리자, 보안 엔지니어, PKI 엔지니어
인증서 체인을 트러스트 스토어로 가져옵니다.	브라우저 및 기타 애플리케이션이 인증서를 신뢰하려면 인증서 발급자가 신뢰할 수 있는 CA 목록인 브라우저의 트러스트 스토어에 포함되어야 합니다. 새 CA 계층 구조의 인증서 체인을 브라우저 및 애플리케이션의 트러스트 스토어에 추가합니다. 최종 엔티티 인증서	클라우드 관리자, 보안 엔지니어, PKI 엔지니어

작업	설명	필요한 기술
	를 신뢰할 수 있는지 확인합니다.	
런북을 생성하여 CA 계층 구조를 문서화합니다.	CA 계층 구조의 아키텍처, 최종 엔터티 인증서를 요청할 수 있는 계정 구조, 빌드 프로세스, 최종 엔터티 인증서 발급(자녀 계정별 셀프 서비스를 허용하려는 경우 제외)과 같은 기본 관리 작업, 사용 및 추적에 대해 설명하는 런북 문서를 생성합니다.	클라우드 관리자, 보안 엔지니어, PKI 엔지니어

관련 리소스

- [CA 계층 구조 설계](#)(AWS 프라이빗 CA 설명서)
- [프라이빗 CA 생성](#)(AWS 프라이빗 CA 설명서)
- [AWS RAM을 사용하여 AWS 프라이빗 CA 교차 계정을 공유하는 방법](#)(AWS 블로그 게시물)
- [AWS 프라이빗 CA 모범 사례](#)(AWS 블로그 게시물)
- [AWS Organizations 내에서 리소스 공유 활성화](#)(AWS RAM 설명서)
- [프라이빗 CA 수명 주기 관리](#)(AWS 프라이빗 CA 설명서)
- [acm-certificate-expiration-check AWS 구성의 경우 \(AWS 구성 설명서\)](#)
- [AWS Certificate Manager는 이제 아마존을 통해 인증서 만료 모니터링을 제공합니다 CloudWatch \(AWS 발표\)](#)
- [AWS Certificate Manager와 통합된 서비스](#)(ACM 설명서)

추가 정보

인증서를 내보낼 때는 암호학적으로 강력하며 조직의 데이터 손실 방지 전략에 부합하는 암호를 사용합니다.

다중 계정 환경에서 모든 Security Hub 멤버 계정의 보안 표준 제어를 끕니다.

작성자: Michael Fuellbier (AWS) 및 Ahmed Bakry (AWS)

환경: 프로덕션

기술: 서버리스 보안, 자격 증명, 규정 준수

AWS 서비스: 아마존 DynamoDB, 아마존, AWS Lambda, AWS Security Hub EventBridge, AWS Step Functions

요약

중요: AWS Security Hub는 이제 계정 전반에 걸쳐 보안 표준 및 제어를 위한 중앙 구성을 지원합니다. 이 새로운 기능은 이 APG 패턴의 솔루션에서 다루는 많은 시나리오를 해결합니다. 이 패턴으로 솔루션을 배포하기 전에 [Security Hub의 중앙 구성](#)을 참조하십시오.

Amazon Web Services (AWS) 클라우드에서는 [CIS AWS Foundations Benchmark](#) 또는 [AWS Foundational Security Best Practices](#)와 같은 AWS Security Hub 표준 제어를 단일 AWS 계정 내에서 수동으로만 해제(비활성화)할 수 있습니다. 다중 계정 환경에서는 “한 번의 클릭”(즉, 한 번의 API 호출)으로 여러 Security Hub 멤버 계정의 제어를 해제할 수 없습니다. 이 패턴은 Security Hub 관리자 계정으로 관리하는 모든 Security Hub 멤버 계정에서 한 번의 클릭으로 Security Hub 표준 제어를 해제하는 방법을 보여줍니다.

사전 조건 및 제한 사항

사전 조건

- 여러 멤버 계정을 관리하는 Security Hub 관리자 계정으로 구성된 다중 계정 환경
- AWS Command Line Interface (AWS CLI) 버전 2, [설치됨](#)
- [AWS Serverless Application Model Command Line Interface \(AWS SAM CLI\)](#), [설치됨](#)

제한 사항

- 이 패턴은 단일 Security Hub 관리자 계정이 여러 멤버 계정을 관리하는 다중 계정 환경에서만 작동합니다.
- 매우 짧은 시간 내에 많은 제어를 변경하는 경우 이벤트 시작 시 여러 개의 병렬 간접 호출이 발생합니다. 이로 인해 API 제한이 발생하여 간접 호출이 실패할 수 있습니다. 예를 들어 [Security Hub 제어 CLI](#)를 사용하여 많은 제어를 프로그래밍 방식으로 변경하는 경우 이 시나리오가 발생할 수 있습니다.

아키텍처

대상 기술 스택

- Amazon DynamoDB
- 아마존 EventBridge
- AWS CLI
- AWS Lambda
- AWS SAM CLI
- AWS Security Hub
- AWS Step Functions

대상 아키텍처

다음 다이어그램은 여러 Security Hub 멤버 계정(Security Hub 관리자 계정에서 볼 때)에서 Security Hub 표준 제어를 해제하는 Step Functions 워크플로우의 예를 보여줍니다.

이 다이어그램은 다음 워크플로우를 포함합니다.

1. EventBridge 규칙은 일일 일정에 따라 시작되며 상태 머신을 호출합니다. AWS CloudFormation 템플릿에서 Schedule 파라미터를 업데이트하여 규칙의 타이밍을 수정할 수 있습니다.
2. EventBridge 규칙은 Security Hub 관리자 계정에서 컨트롤을 켜거나 끌 때마다 시작됩니다.
3. Step Functions 상태 머신은 Security Hub 관리자 계정의 보안 표준 제어(즉, 켜거나 끈 제어)의 상태를 멤버 계정으로 전파합니다.
4. 계정 간 AWS Identity and Access Management(IAM) 역할이 각 멤버 계정에 배포되며 상태 머신에서 말합니다. 상태 머신은 각 멤버 계정에서 제어 기능을 켜거나 끕니다.

5. DynamoDB 테이블은 특정 계정에서 어떤 제어 기능을 켜거나 끄는지에 대한 예외 사항 및 정보를 포함합니다. 이 정보는 해당 멤버 계정에 대하여 Security Hub 관리자 계정에서 가져온 구성을 재정의합니다.

참고: 예약된 EventBridge 규칙의 목적은 새로 추가된 Security Hub 구성원 계정이 기존 계정과 동일한 제어 상태를 갖도록 하는 것입니다.

도구

- [Amazon DynamoDB](#)는 빠르고 예측 가능하고 확장 가능한 성능을 제공하는 완전 관리형 NoSQL 데이터베이스 서비스입니다.
- [EventBridgeAmazon](#)은 다양한 소스의 실시간 데이터와 애플리케이션을 연결하는 데 도움이 되는 서버리스 이벤트 버스 서비스입니다. AWS Lambda 함수, API 대상을 사용하는 HTTP 간접 호출 엔드 포인트 또는 다른 AWS 계정의 이벤트 버스를 예로 들 수 있습니다.
- [AWS Command Line Interface \(AWS CLI\)](#)는 명령줄 셸에서 명령을 사용하여 AWS 서비스와 상호 작용할 수 있는 오픈 소스 도구입니다.
- [AWS Lambda](#)는 서버를 프로비저닝하거나 관리할 필요 없이 코드를 실행하는 데 도움이 되는 컴퓨팅 서비스입니다. 필요할 때만 코드를 실행하며 자동으로 확장이 가능하므로 사용한 컴퓨팅 시간만큼만 비용을 지불합니다.
- [AWS Serverless Application Model\(AWS SAM\)](#)은 AWS 클라우드에서 서버리스 애플리케이션을 구축하는 데 사용할 수 있는 오픈 소스 프레임워크입니다.
- [AWS Security Hub](#)는 AWS의 보안 상태에 대한 포괄적인 보기를 제공합니다. 이를 사용하면 사용자의 AWS 환경이 보안 업계 표준 및 모범 사례를 준수하는지 확인할 수 있습니다.
- [AWS Step Functions](#)는 AWS Lambda 함수와 기타 AWS 서비스를 결합할 수 있는 서버리스 오케스트레이션 서비스로, 비즈니스 크리티컬 애플리케이션을 구축합니다.

코드

이 패턴의 코드는 GitHub [AWS Security Hub 교차 계정 제어 비활성화](#) 리포지토리에서 사용할 수 있습니다. 코드 리포지토리에는 다음 파일과 폴더가 포함되어 있습니다.

- UpdateMembers/template.yaml— 이 파일에는 Step Functions 상태 시스템 및 EventBridge 규칙을 포함하여 Security Hub 관리자 계정에 배포된 구성 요소가 들어 있습니다.
- member-iam-role/template.yaml— 이 파일에는 멤버 계정에 교차 계정 IAM 역할을 배포하기 위한 코드가 들어 있습니다.

- stateMachine.json – 이 파일은 상태 머신의 워크플로우를 정의합니다.
- GetMembers/index.py— 이 파일에는 GetMembers상태 머신의 코드가 들어 있습니다. 스크립트가 기존의 모든 Security Hub 멤버 계정에서 보안 표준 제어 상태를 검색합니다.
- UpdateMember/index.py–이 파일에는 각 멤버 계정의 제어 상태를 업데이트하는 스크립트가 들어 있습니다.
- CheckResult/index.py – 이 파일에는 워크플로우 간접 호출 상태(수락 또는 실패)를 확인하는 스크립트가 들어 있습니다.

에픽

Security Hub 멤버 계정에 교차 계정 IAM 역할 배포

작업	설명	필요한 기술
Security Hub 관리자 계정의 계정 ID를 식별합니다.	Security Hub 관리자 계정 을 설정한 다음 관리자 계정의 계정 ID를 기록해 둡니다.	클라우드 아키텍트
회원 계정에 교차 계정 IAM 역할을 포함하는 CloudFormation 템플릿을 배포하십시오.	Security Hub 관리자 계정으로 관리하는 모든 멤버 계정에 member-iam-role/template.yaml 템플릿을 배포하려면 다음 명령을 실행합니다. <pre data-bbox="597 1367 1024 1841">aws cloudformation deploy --template- file member-iam-role/ template.yaml -- capabilities CAPABILIT Y_NAMED_IAM --stack-n ame <your-stack-name> --parameter-overri des SecurityHubAdminAc countId=<your-acco unt-ID></pre>	AWS DevOps

작업	설명	필요한 기술
	SecurityHubAdminAccountId 파라미터가 앞서 기록한 Security Hub 관리자 계정 ID와 일치해야 합니다.	

Security Hub 관리자 계정에 상태 머신을 배포합니다.

작업	설명	필요한 기술
상태 머신이 포함된 CloudFormation 템플릿을 AWS SAM으로 패키징합니다.	<p>Security Hub 관리자 계정에서 UpdateMembers/template.yaml 템플릿을 패키징하려면 다음 명령을 실행합니다.</p> <pre>sam package --template-file UpdateMembers/template.yaml --output-template-file UpdateMembers/template-out.yaml --s3-bucket <your-s3-bucket-name></pre> <p>참고: Amazon Simple Storage 서비스 (Amazon S3) 버킷은 템플릿을 CloudFormation 배포하는 동일한 AWS 지역에 있어야 합니다.</p>	AWS DevOps
Security Hub 관리자 계정에 패키지 CloudFormation 템플릿을 배포합니다.	Security Hub 관리자 계정에서 CloudFormation 템플릿을 배포하려면 다음 명령을 실행합니다.	AWS DevOps

작업	설명	필요한 기술
	<pre>aws cloudformation deploy --template- file UpdateMembers/ template-out.yaml -- capabilities CAPABILIT Y_IAM --stack-name <your-stack-name></pre> <p>member-iam-role/te mplate.yaml 템플릿에서 MemberIM RolePath 매개 변 수는 IAM 매개 변수와 일치해 야 하고 RolePathRoleNameM emberIAM은 IAM과 일치해야 합니다. RoleName</p> <p>참고: Security Hub가 지역적 인 서비스이므로 템플릿을 각 AWS 리전에 개별적으로 배포 해야 합니다. 먼저 솔루션을 각 리전의 S3 버킷에 패키징해야 합니다.</p>	

관련 리소스

- [Security Hub 관리자 계정 지정](#)(AWS Security Hub 설명서)
- [Step Function 상태 머신 실행에 오류 관리, 재시도, 알림 추가 다루기](#)(AWS 블로그 게시물)

를 사용하여 AWS IAM 자격 증명 센터의 AWS CLI 자격 증명을 업데이트합니다. PowerShell

작성자: Chad Miles(AWS) 및 Andy Bowen(AWS)

환경: 프로덕션

기술: 보안, 자격 증명, 규정 준수, 클라우드 네이티브

워크로드: 오픈 소스

AWS 서비스: AWS 도구
PowerShell, AWS IAM 자격 증명 센터

요약

AWS Command Line Interface(AWS CLI), AWS SDK 또는 AWS Cloud Development Kit(AWS CDK)와 함께 AWS IAM Identity Center(AWS Single Sign-On의 후속) 보안 인증 정보를 사용하려면 일반적으로 IAM Identity Center 콘솔의 보안 인증 정보를 복사하여 명령줄 인터페이스에 붙여 넣어야 합니다. 이 프로세스는 상당한 시간이 소요될 수 있으며 액세스가 필요한 각 계정에 대해 반복해야 합니다.

일반적인 해결 방법 중 하나는 AWS CLI `aws sso configure` 명령을 사용하는 것입니다. 이 명령은 IAM Identity Center 지원 프로필을 AWS CLI 또는 AWS SDK에 추가합니다. 하지만 이 솔루션의 단점은 이러한 방식으로 구성한 각 AWS CLI 프로필 또는 계정에 대해 `aws sso login` 명령을 실행해야 한다는 것입니다.

대체 솔루션으로서 이 패턴은 AWS CLI [명명된 프로필과](#) AWS Tools를 사용하여 단일 IAM Identity Center 인스턴스에서 여러 계정의 자격 증명을 동시에 저장하고 새로 고치는 방법을 설명합니다. PowerShell 또한 이 스크립트는 IAM Identity Center에 다시 로그인하지 않고도 보안 인증 정보를 새로 고칠 수 있도록 IAM Identity Center 세션 데이터를 메모리에 저장합니다.

사전 조건 및 제한 사항

사전 조건

- PowerShell, 설치 및 구성. 자세한 내용은 [설치 PowerShell](#) (Microsoft 설명서) 를 참조하십시오.
- 설치 및 구성을 위한 PowerShell AWS 도구. 성능상의 이유로, 라는 AWS Tools를 위한 모듈화된 버전의 AWS Tools를 설치하는 것이 좋습니다. PowerShell AWS.Tools 각 AWS 서비스는 개별적인

소형 모듈에서 지원됩니다. PowerShell 프롬프트에 다음 명령을 입력하여 이 패턴에 필요한 모듈을 설치합니다. `AWS.Tools.Installer,SS0, SS0IDC`

```
Install-Module AWS.Tools.Installer
Install-AWSToolsModule SS0, SS0IDC
```

자세한 내용은 [Windows에 AWS.Tools 설치](#) 또는 [Linux 또는 macOS에 AWS.Tools 설치](#)를 참조하세요.

- 다음 중 하나를 수행하여 AWS CLI 또는 AWS SDK를 유효한 보안 인증 정보로 미리 구성해야 합니다.
 - AWS CLI `aws configure` 명령을 사용합니다. 자세한 내용을 알아보려면 [빠른 구성](#)(AWS CLI 설명서)을 참조하세요.
 - IAM 역할을 통해 일시적으로 액세스할 수 있도록 AWS CLI 또는 AWS CDK를 구성합니다. 자세한 내용은 [CLI 액세스를 위한 IAM 역할 보안 인증 정보 가져오기](#)(IAM Identity Center 설명서)를 참조하세요.

제한 사항

- 이 스크립트는 파이프라인 또는 완전 자동화된 솔루션에서 사용할 수 없습니다. 이 스크립트를 배포할 때는 IAM Identity Center에서 수동으로 액세스를 승인해야 합니다. 그러면 스크립트가 자동으로 계속됩니다.

제품 버전

- 모든 운영 체제에서는 [PowerShell 버전 7.0](#) 이상을 사용하는 것이 좋습니다.

아키텍처

이 패턴의 스크립트를 사용하여 여러 IAM Identity Center 보안 인증 정보를 동시에 새로 고치고, AWS CLI, AWS SDK 또는 AWS CDK와 함께 사용할 보안 인증 정보 파일을 생성할 수 있습니다.

도구

서비스

- [AWS Command Line Interface \(AWS CLI\)](#)는 명령줄 셸에서 명령을 사용하여 AWS 서비스와 상호 작용할 수 있는 오픈 소스 도구입니다.
- [AWS IAM Identity Center](#)를 사용하면 모든 AWS 계정과 클라우드 애플리케이션에 대한 AWS Single Sign-On(SSO) 액세스를 중앙에서 관리할 수 있습니다.
- [AWS Tools for PowerShell](#)는 PowerShell 명령줄에서 AWS 리소스의 작업을 스크립팅하는 데 도움이 되는 PowerShell 모듈 세트입니다.

기타 도구

- [PowerShell](#) Windows, Linux 및 macOS에서 실행되는 Microsoft 자동화 및 구성 관리 프로그램입니다.

모범 사례

각 IAM Identity Center 인스턴스마다 이 스크립트 사본을 하나씩 보관합니다. 여러 인스턴스에 하나의 스크립트를 사용하는 것은 지원되지 않습니다.

에픽

SSO 스크립트 실행

작업	설명	필요한 기술
SSO 스크립트를 사용자 지정합니다.	<ol style="list-style-type: none"> 1. 추가 정보 섹션에서 SSO 스크립트를 복사합니다. 2. Param 섹션에서 AWS 환경에 대해 다음 변수의 값을 정의합니다. <ul style="list-style-type: none"> • DefaultRoleName -기본적으로 사용하도록 설정된 IAM 역할 또는 권한입니다. • Region-IAM Identity Center가 배포된 AWS 리전입니다. 전체 리전 및 코 	클라우드 관리자

작업	설명	필요한 기술
	<p>드 목록은 리전별 엔드포인트를 참조하세요.</p> <ul style="list-style-type: none"> • StartUrl-IAM Identity Center 로그인 페이지에 액세스하는 데 사용되는 URL입니다. 스크립트의 예제 값과 동일한 형식을 사용합니다. • EnvironmentName -이 스크립트 사본을 참조하기 위한 짧은 이름이며 동일한 세션에서 여러 스크립트 사본을 실행할 때 사용합니다. <p>3. # Add your Account Information 에 관한 내용인 열 번째 줄에서 환경을 반영하도록 해시 테이블의 다음 값을 편집합니다.</p> <ul style="list-style-type: none"> • Profile-임시 자격 증명을 저장할 AWS CLI 프로필 이름. • AccountId -보안 인증 정보를 검색하는 AWS 계정의 ID. • RoleName-사용하려는 IAM Identity Center 역할 또는 권한 집합의 이름. Param 섹션에서 정의한 것과 동일한 역할을 사용하려는 경우 \$DefaultRoleName 으로 그대로 둘 수 있습니다. 	

작업	설명	필요한 기술
	<p>해시 테이블의 각 줄은 마지막 줄을 제외하고 쉼표로 끝나야 합니다.</p>	
SSO 스크립트 실행.	<p>다음 명령을 사용하여 PowerShell 셸에서 사용자 지정 스크립트를 실행하는 것이 좋습니다.</p> <pre data-bbox="597 636 1027 751">./Set-AwsCliSsoCredentials.ps1</pre> <p>다음 명령을 입력하여 다른 셸에서 스크립트를 실행할 수도 있습니다.</p> <pre data-bbox="597 961 1027 1077">pwsh Set-AwsCliSsoCredentials.ps1</pre>	클라우드 관리자

문제 해결

문제	Solution
No Access 오류	<p>사용 중인 IAM 역할에는 RoleName 파라미터에 정의한 역할 또는 권한 세트에 액세스할 수 있는 권한이 없습니다. 사용 중인 역할의 권한을 업데이트하거나 스크립트에서 다른 역할 또는 권한 세트를 정의합니다.</p>

관련 리소스

- [구성 설정이 저장되는 장소는 어디가요? \(AWS CLI 설명서\)](#)
- [AWS IAM Identity Center를 사용하도록 AWS CLI 구성\(AWS CLI 설명서\)](#)

- [명명된 프로필 사용](#)(AWS CLI 설명서)

추가 정보

SSO 스크립트

다음 스크립트에서 꺾쇠 괄호(<>) 안의 자리 표시자를 사용자 고유의 정보로 바꾸고 꺾쇠 괄호를 제거합니다.

```
Set-AwsCliSsoCredentials.ps1
Param(
    $DefaultRoleName = '<AWSAdministratorAccess>',
    $Region          = '<us-west-2>',
    $StartUrl        = "<https://d-12345abcde.awsapps.com/start/>",
    $EnvironmentName = "<CompanyName>"
)
Try {$SsoAwsAccounts = (Get-Variable -name "$($EnvironmentName)SsoAwsAccounts" -Scope
    Global -ErrorAction 'SilentlyContinue').Value.Clone()}
Catch {$SsoAwsAccounts = $False}
if (-not $SsoAwsAccounts) { $SsoAwsAccounts = @(
# Add your account information in the list of hash tables below, expand as necessary,
and do not forget the commas
    @{Profile = "<Account1>"           ; AccountId = "<012345678901 >"; RoleName =
$DefaultRoleName },
    @{Profile = "<Account2>"           ; AccountId = "<123456789012>"; RoleName =
"<AWSReadOnlyAccess>" }
)}
$errorActionPreference = "Stop"
if (-not (Test-Path ~\.aws))      { New-Item ~\.aws -type Directory }
if (-not (Test-Path ~\.aws\credentials)) { New-Item ~\.aws\credentials -type File }
$CredentialFile = Resolve-Path ~\.aws\credentials
$PseudoCreds    = @{AccessKey =
'AKAEXAMPLE123ACCESS'; SecretKey='PsuedoS3cret4cceSSKey123PsuedoS3cretKey'} # Pseudo
Creds, do not edit.
Try {$SSOTokenExpire = (Get-Variable -Scope Global -Name
"$($EnvironmentName)SSOTokenExpire" -ErrorAction 'SilentlyContinue').Value} Catch
{$SSOTokenExpire = $False}
Try {$SSOToken       = (Get-Variable -Scope Global -Name "$($EnvironmentName)SSOToken"
-ErrorAction 'SilentlyContinue').Value }      Catch {$SSOToken       = $False}
if ( $SSOTokenExpire -lt (Get-Date) ) {
    $SSOToken = $Null
}
```

```

$Client = Register-SSO0IDCClient -ClientName cli-sso-client -ClientType public -
Region $Region @PsuedoCreds
$Device = $Client | Start-SSO0IDCDeviceAuthorization -StartUrl $StartUrl -Region
$Region @PsuedoCreds
Write-Host "A Browser window should open. Please login there and click ALLOW." -
NoNewline
Start-Process $Device.VerificationUriComplete
While (-Not $SSOToken){
    Try {$SSOToken = $Client | New-SSO0IDCToken -DeviceCode $Device.DeviceCode -
GrantType "urn:ietf:params:oauth:grant-type:device_code" -Region $Region @PsuedoCreds}
    Catch {If ($_.Exception.Message -notlike "*AuthorizationPendingException*")}
{Write-Error $_.Exception} ; Start-Sleep 1}
}
$SSOTokenExpire = (Get-Date).AddSeconds($SSOToken.ExpiresIn)
Set-Variable -Name "$($EnvironmentName)SSOToken" -Value $SSOToken -Scope Global
Set-Variable -Name "$($EnvironmentName)SSOTokenExpire" -Value $SSOTokenExpire -
Scope Global
}
$CredsTime = $SSOTokenExpire - (Get-Date)
$CredsTimeText = ('{0:D2}:{1:D2}:{2:D2} left on SSO Token' -f $CredsTime.Hours,
$CredsTime.Minutes, $CredsTime.Seconds).TrimStart("0 :")
for ($i = 0; $i -lt $SsoAwsAccounts.Count; $i++) {
    if (([DateTimeOffset]::FromUnixTimeSeconds($SsoAwsAccounts[$i].CredsExpiration /
1000)).DateTime -lt (Get-Date).ToUniversalTime()) {
        Write-host "`r
`rRegistering Profile $($SsoAwsAccounts[$i].Profile)" -NoNewline
        $TempCreds = $SSOToken | Get-SSORoleCredential -AccountId
$SsoAwsAccounts[$i].AccountId -RoleName $SsoAwsAccounts[$i].RoleName -Region $Region
@PsuedoCreds
        [PSCustomObject]@{AccessKey = $TempCreds.AccessKeyId; SecretKey =
$TempCreds.SecretAccessKey; SessionToken = $TempCreds.SessionToken
        } | Set-AWSCredential -StoreAs $SsoAwsAccounts[$i].Profile -ProfileLocation
$CredentialFile
        $SsoAwsAccounts[$i].CredsExpiration = $TempCreds.Expiration
    }
}
Set-Variable -name "$($EnvironmentName)SsoAwsAccounts" -Value $SsoAwsAccounts.Clone() -
Scope Global
Write-Host "`r$($SsoAwsAccounts.Profile) Profiles registered, $CredsTimeText"

```

AWS Config를 사용하여 Amazon Redshift 보안 구성 모니터링

작성자: Lucas Kauffman(AWS) 및 abhishek sengar(AWS)

코드 리포지토리: [awslabs/
aws-config-rules](https://github.com/aws-labs/aws-config-rules)

환경: 프로덕션

기술: 보안, 자격 증명, 규정 준수

AWS 서비스: AWS Config,
Amazon Redshift, AWS
Lambda

요약

AWS Config를 사용하여 AWS 리소스의 보안 구성을 평가할 수 있습니다. AWS Config는 리소스를 모니터링할 수 있으며, 구성 설정이 정의된 규칙을 위반하는 경우 AWS Config는 해당 리소스를 비준수로 표시합니다.

AWS Config를 사용하여 Amazon Redshift 클러스터와 데이터베이스를 평가하고 모니터링할 수 있습니다. 보안 권장 사항 및 기능에 대한 자세한 내용은 [Amazon Redshift의 보안](#)을 참조하십시오. 이 패턴에는 AWS Config에 대한 사용자 지정 AWS Lambda 규칙이 포함됩니다. 계정에 이러한 규칙을 배포하여 Amazon Redshift 클러스터 및 데이터베이스의 보안 구성을 모니터링할 수 있습니다. 이 패턴의 규칙은 AWS Config를 사용하여 다음을 확인하는 데 도움이 됩니다.

- Amazon Redshift 클러스터의 데이터베이스에 대해 감사 로깅 활성화
- Amazon Redshift 클러스터에 연결하려면 SSL 필요
- Federal Information Processing Standards(FIPS) 사용됨
- Amazon Redshift 클러스터의 데이터베이스는 암호화됨
- 사용자 활동 모니터링 활성화

사전 조건 및 제한 사항

사전 조건

- 활성 상태의 AWS 계정

- AWS Config가 AWS 계정에서 활성화되어 있어야 합니다. 자세한 내용은 [콘솔을 사용한 AWS Config 설정](#) 또는 [AWS CLI를 사용한 AWS Config 설정](#)을 참조하십시오.
- Python 버전 3.9 이상을 AWS Lambda 핸들러에 사용해야 합니다. 자세한 내용은 [Python을 사용한 작업](#)(AWS Lambda 설명서)을 참조하십시오.

제품 버전

- Python 버전 3.9 이상.

아키텍처

대상 기술 스택

- AWS Config

대상 아키텍처

1. AWS Config는 사용자 지정 규칙을 주기적으로 실행합니다.
2. 사용자 지정 규칙은 Lambda 함수를 간접적으로 호출합니다.
3. Lambda 함수는 Amazon Redshift 클러스터에서 규정을 준수하지 않는 구성이 있는지 확인합니다.
4. Lambda 함수는 각 Amazon Redshift 클러스터의 규정 준수 상태를 AWS Config에 보고합니다.

자동화 및 규모 조정

AWS Config 사용자 지정 규칙은 계정 내 모든 Amazon Redshift 클러스터를 평가하도록 규모를 조정합니다. 이 솔루션의 규모를 조정하기 위한 추가 조치는 필요하지 않습니다.

도구

서비스

- [AWS Config](#)는 사용자의 AWS 계정에 있는 리소스와 그 구성 방식을 자세히 보여줍니다. 리소스가 서로 관련되는 방식과 리소스의 구성이 시간이 지남에 따라 변경된 방식을 식별하는 데 도움이 됩니다.

- [AWS Identity and Access Management\(IAM\)](#)를 사용하면 AWS 리소스를 사용하도록 인증받고 권한이 부여된 사용자를 통제함으로써 AWS 리소스에 대한 액세스를 안전하게 관리할 수 있습니다.
- [AWS Lambda](#)는 서버를 프로비저닝하거나 관리할 필요 없이 코드를 실행하는 데 도움이 되는 컴퓨팅 서비스입니다. 필요할 때만 코드를 실행하며 자동으로 확장이 가능하므로 사용한 컴퓨팅 시간만큼만 비용을 지불합니다.
- [Amazon Redshift](#)는 AWS 클라우드에서 관리되는 페타바이트급 데이터 웨어하우스 서비스입니다.

코드 리포지토리

이 패턴의 코드는 리포지토리에서 사용할 수 있습니다. GitHub [aws-config-rules](#) 이 리포지토리의 사용자 지정 규칙은 Python 프로그래밍 언어의 Lambda 규칙입니다. 이 리포지토리에는 AWS Config에 대한 많은 사용자 지정 규칙이 포함되어 있습니다. 이 패턴에는 다음 규칙만 사용됩니다.

- REDSHIFT_AUDIT_ENABLED – Amazon Redshift 클러스터에서 감사 로깅이 활성화되어 있는지 확인합니다. 사용자 활동 모니터링이 활성화되어 있는지도 확인하려면 REDSHIFT_USER_ACTIVITY_MONITORING_ENABLED 규칙을 대신 배포하십시오.
- REDSHIFT_SSL_REQUIRED – Amazon Redshift 클러스터에 연결하는 데 SSL이 필요한지 확인합니다. 연방 정보 처리 표준(FIPS) 암호도 사용 중인지 확인하려면 REDSHIFT_FIPS_REQUIRED 규칙을 대신 배포하십시오.
- REDSHIFT_FIPS_REQUIRED – SSL이 필요하고 FIPS 암호가 사용 중인지 확인하십시오.
- REDSHIFT_DB_ENCRYPTED – Amazon Redshift 클러스터의 데이터베이스가 암호화되어 있는지 확인합니다.
- REDSHIFT_USER_ACTIVITY_MONITORING_ENABLED – 감사 로깅 및 사용자 활동 모니터링이 활성화되어 있는지 확인합니다.

에픽

규칙 배포 준비

작업	설명	필요한 기술
IAM 정책을 구성합니다.	1. Lambda 실행 역할이 Amazon Redshift 클러스터 구성을 읽을 수 있도록 허용하는 사용자 지정 IAM 자격	AWS 관리자

작업	설명	필요한 기술
	<p>증명 기반 정책을 생성합니다. 자세한 내용은 리소스에 대한 액세스 관리(Amazon Redshift 설명서) 및 IAM 정책 생성(IAM 설명서)을 참조하십시오.</p> <pre data-bbox="630 520 1029 1806"> { "Version": "2012-10-17", "Statement": [{ "Effect": "Allow", "Action": ["redshift :DescribeClusterPa rameterGroups", "redshift :DescribeClusterPa rameters", "redshift :DescribeClusters", "redshift :DescribeClusterSe curityGroups", "redshift :DescribeClusterSn apshots", "redshift :DescribeClusterSu bnetGroups", "redshift :DescribeEventSubs criptions", "redshift :DescribeLoggingSt atus"], }], } </pre>	

작업	설명	필요한 기술
	<pre> "Resource": "*" }] } </pre> <p>2. AWSLambdaExecute 및 AWSConfigRulesExecutionRole 관리형 정책을 Lambda 실행 역할에 대한 권한 정책으로 할당합니다. 지침은 IAM 자격 증명 권한 추가(IAM 설명서)를 참조하십시오.</p>	
<p>리포지토리를 복제합니다.</p>	<p>Bash 셸에서 다음 명령을 실행합니다. 이렇게 하면 리포지토리가 복제됩니다 aws-config-rules. GitHub</p> <pre> git clone https://github.com/aws-labs/aws-config-rules.git </pre>	<p>일반 AWS</p>

AWS Config에 규칙을 배포합니다.

작업	설명	필요한 기술
<p>AWS Config에서 규칙을 배포합니다.</p>	<p>사용자 지정 Lambda 규칙 생성(AWS Config 설명서)의 지침에 따라 계정에 다음 규칙 중 하나 이상을 배포합니다.</p> <ul style="list-style-type: none"> • REDSHIFT_AUDIT_ENABLED 	<p>AWS 관리자</p>

작업	설명	필요한 기술
	<ul style="list-style-type: none"> • REDSHIFT_SSL_REQUIRED • REDSHIFT_FIPS_REQUIRED • REDSHIFT_DB_ENCRYPTED • REDSHIFT_USER_ACTIVITY_MONITORING_ENABLED 	
<p>규칙이 제대로 작동하는지 확인합니다.</p>	<p>규칙을 배포한 후에는 리소스 평가(AWS Config 설명서)의 지침에 따라 AWS Config가 Amazon Redshift 리소스를 올바르게 평가하고 있는지 확인합니다.</p>	<p>일반 AWS</p>

관련 리소스

AWS 서비스 설명서

- [Amazon Redshift의 보안](#)(Amazon Redshift 설명서)
- [데이터베이스 보안 관리](#)(Amazon Redshift 설명서)
- [AWS Config 사용자 지정 규칙](#)(AWS Config 설명서)

AWS 권장 가이드

- [새 Amazon Redshift 클러스터에 필수 SSL 엔드포인트가 있는지 확인](#)
- [Amazon Redshift 클러스터는 생성 시 암호화되었는지 확인](#)

추가 정보

AWS Config에서 다음과 같은 AWS 관리형 규칙을 사용하여 Amazon Redshift의 다음과 같은 보안 구성을 확인할 수 있습니다.

- [redshift-cluster-configuration-check](#)— 이 규칙을 사용하여 Amazon Redshift 클러스터의 데이터베이스에 감사 로깅이 활성화되었는지 확인하고 데이터베이스가 암호화되었는지 확인합니다.
- [redshift-require-tls-ssl](#)— 이 규칙을 사용하여 Amazon Redshift 클러스터에 연결하는 데 SSL이 필요한지 확인할 수 있습니다.

Network Firewall을 사용하여 아웃바운드 트래픽의 Server Name Indication(SNI)에서 DNS 도메인 이름 캡처

작성자: Kirankumar Chandrashekar(AWS)

환경: PoC 또는 파일럿

기술: 보안, ID, 규정 준수, 네트워킹, 웹 및 모바일 앱

워크로드: 기타 모든 워크로드

AWS 서비스: AWS Lambda, AWS 네트워크 방화벽, 아마존 VPC, 아마존 로그 CloudWatch

요약

이 패턴은 Amazon Web Services(AWS) Network Firewall을 사용하여 아웃바운드 네트워크 트래픽의 HTTPS 헤더에 있는 Server Name Indication(SNI)에서 제공하는 DNS 도메인 이름을 수집하는 방법을 보여줍니다. Network Firewall은 Amazon Virtual Private Cloud(VPC)에 대한 중요한 네트워크 보호 기능을 쉽게 배포할 수 있게 해주는 관리형 서비스입니다. 여기에는 특정 보안 요구 사항을 충족하지 못하는 패킷을 차단하는 방화벽으로 아웃바운드 트래픽을 보호하는 기능이 포함됩니다. 특정 DNS 도메인 이름에 대한 아웃바운드 트래픽을 보호하는 것을 이그레스 필터링이라고 하며, 이는 한 네트워크에서 다른 네트워크로의 아웃바운드 정보 흐름을 모니터링하고 잠재적으로 제한하는 방법입니다.

네트워크 방화벽을 통과하는 SNI 데이터를 캡처한 후에는 Amazon CloudWatch Logs 및 AWS Lambda를 사용하여 이메일 알림을 생성하는 Amazon Simple Notification Service (Amazon SNS) 주제를 게시할 수 있습니다. 이메일 알림에는 서버 이름 및 기타 관련 SNI 정보가 포함됩니다. 또한 이 패턴의 출력을 사용하여 방화벽 규칙을 사용하여 SNI에서 도메인 이름별로 아웃바운드 트래픽을 허용하거나 제한할 수 있습니다. 자세한 내용은 Network Firewall 설명서의 [AWS Network Firewall의 상태 저장 규칙 그룹 사용](#)을 참조하십시오.

사전 조건 및 제한 사항

사전 조건

- 활성 상태의 AWS 계정.
- [AWS Command Line Interface\(AWS CLI\)](#) 버전 2, Linux, macOS, 또는 Windows에 설치 및 구성됨

- Amazon VPC에서 설정 및 구성되었으며 아웃바운드 트래픽을 검사하는 데 사용 중인 [Network Firewall](#)

참고: Network Firewall은 다음 VPC 구성 중 하나를 사용할 수 있습니다.

- [인터넷 게이트웨이가 있는 간단한 단일 영역 아키텍처](#)
- [인터넷 게이트웨이가 있는 다중 영역 아키텍처](#)
- [인터넷 게이트웨이와 NAT 게이트웨이가 있는 아키텍처](#)

아키텍처

다음 다이어그램은 Network Firewall을 사용하여 아웃바운드 네트워크 트래픽에서 SNI 데이터를 수집한 다음 CloudWatch Logs and Lambda를 사용하여 해당 데이터를 SNS 주제에 게시하는 방법을 보여줍니다.

이 다이어그램은 다음 워크플로를 보여줍니다.

1. Network Firewall은 아웃바운드 네트워크 트래픽의 HTTPS 헤더에 있는 SNI 데이터에서 도메인 이름을 수집합니다.
2. CloudWatch 로그는 SNI 데이터를 모니터링하고 아웃바운드 네트워크 트래픽이 Network Firewall을 통과할 때마다 Lambda 함수를 호출합니다.
3. Lambda 함수는 Logs에서 CloudWatch 캡처한 SNI 데이터를 읽은 다음 해당 데이터를 SNS 주제에 게시합니다.
4. SNS 주제는 SNI 데이터가 포함된 이메일 알림을 보냅니다.

자동화 및 규모 조정

- [AWS를 CloudFormation](#) 사용하여 [인프라를 코드로](#) 사용하여 이 패턴을 생성할 수 있습니다.

기술 스택

- 아마존 CloudWatch 로그
- Amazon SNS
- Amazon VPC

- AWS Lambda
- AWS Network Firewall

도구

서비스

- [Amazon CloudWatch Logs](#) — Amazon CloudWatch Logs를 사용하여 Amazon Elastic Compute Cloud (Amazon EC2) 인스턴스, CloudTrail AWS, Amazon Route 53 및 기타 소스에서 로그 파일을 모니터링, 저장 및 액세스할 수 있습니다.
- [Amazon SNS](#) – Amazon Simple Notification Service(Amazon SNS)는 게시자에서 구독자(생산자 및 소비자라고도 함)로 메시지를 전송하는 관리형 서비스입니다.
- [Amazon VPC](#) – Amazon Virtual Private Cloud(VPC)를 사용하면 정의한 가상 네트워크에서 AWS 리소스를 시작할 수 있도록 클라우드의 논리적으로 격리된 섹션을 프로비저닝할 수 있습니다. 이 가상 네트워크는 AWS의 확장 가능한 인프라를 사용한다는 이점과 함께 고객의 자체 데이터 센터에서 운영하는 기존 네트워크와 매우 유사합니다.
- [AWS Lambda](#) - AWS Lambda는 서버를 프로비저닝하거나 관리하지 않고도 코드를 실행할 수 있게 해주는 컴퓨팅 서비스입니다.
- [AWS Network Firewall](#) – AWS Network Firewall은 모든 Amazon VPC에 필수 네트워크 보호 기능을 손쉽게 배포할 수 있도록 해주는 관리형 서비스입니다.

에픽

Network Firewall에 대한 CloudWatch 로그 그룹 생성

작업	설명	필요한 기술
CloudWatch 로그 그룹을 생성합니다.	<ol style="list-style-type: none"> 1. AWS 관리 콘솔에 로그인하고 CloudWatch 콘솔을 엽니다. 2. 탐색 창에서 로그 그룹을 선택합니다. 3. 작업을 선택한 후 로그 그룹 생성을 선택합니다. 	클라우드 관리자

작업	설명	필요한 기술
	<p>4. 로그 그룹의 이름을 입력한 다음 로그 그룹 생성을 선택합니다.</p> <p>자세한 내용은 CloudWatch 설명서의 로그 그룹 및 로그 스트림 사용을 참조하십시오.</p>	

SNS 주제 생성 및 구독

작업	설명	필요한 기술
SNS 주제를 생성합니다.	SNS 주제를 생성하려면 Amazon SNS 설명서 의 지침을 따르십시오.	클라우드 관리자
SNS 주제에 엔드포인트를 구독합니다.	생성한 SNS 주제에 대한 엔드포인트로 이메일 주소를 구독하려면 Amazon SNS 설명서 의 지침을 따르십시오. 프로토콜에서 Email/Email-JSON 을 선택합니다. 참고: 요구 사항에 따라 다른 엔드포인트를 선택할 수도 있습니다.	클라우드 관리자

Network Firewall 로그인 설정

작업	설명	필요한 기술
방화벽 로깅을 활성화합니다.	1. AWS Management Console에 로그인한 후 Amazon VPC 콘솔 을 엽니다.	클라우드 관리자

작업	설명	필요한 기술
	<ol style="list-style-type: none"> 탐색 창의 NETEWORK FIREWALL에서 방화벽을 선택합니다. 방화벽 섹션에서 아웃바운드 트래픽에 대해 SNI에서 서버 이름을 캡처하려는 방화벽을 선택합니다. 방화벽 세부 정보 탭을 선택한 다음 로깅 섹션에서 편집을 선택합니다. 로그 유형에서는 알림을 선택합니다. 경고의 로그 대상으로 CloudWatch 로그 그룹을 선택합니다. CloudWatch 로그 그룹의 경우 이전에 만든 로그 그룹을 검색하여 선택한 다음 [Save] 를 선택합니다. <p>Network Firewall의 CloudWatch 로그 대상으로 Logs를 사용하는 방법에 대한 자세한 내용은 Network Firewall 설명서의 Amazon CloudWatch Logs를 참조하십시오.</p>	

Network Firewall에서 상태 저장 규칙 설정

작업	설명	필요한 기술
상태 저장 규칙을 생성합니다.	1. AWS Management Console에 로그인한 후 Amazon VPC 콘솔 을 엽니다.	클라우드 관리자

작업	설명	필요한 기술
	<ol style="list-style-type: none"> 2. 탐색 창의 NETWORK FIREWALL에서 Network Firewall 규칙 그룹을 선택합니다. 3. Network Firewall 규칙 그룹 생성을 선택합니다. 4. Network Firewall 규칙 그룹 생성 페이지에서 규칙 그룹 유형으로 상태 저장 규칙 그룹을 선택합니다. 참고: 자세한 내용은 AWS Network Firewall에서 상태 저장 규칙 그룹 작업을 참조하십시오. 5. 상태 저장 규칙 그룹 섹션에서 규칙 그룹의 이름과 규칙 그룹의 설명을 입력합니다. 6. 용량에서 상태 저장 규칙 그룹에 허용하려는 최대 용량 (최대 30,000개)을 설정합니다. 참고: 규칙 그룹을 생성한 후에는 이 설정을 변경할 수 없습니다. 용량을 계산하는 방법에 대한 자세한 내용은 AWS Network Firewall에서 규칙 그룹 용량 설정을 참조하십시오. 최대 설정에 대한 자세한 내용은 AWS Network Firewall 할당량을 참조하십시오. 7. 상태 저장 규칙 그룹 옵션에서 5-tuple을 선택합니다. 8. 상태 저장 규칙 순서 섹션에서 기본값을 선택합니다. 	

작업	설명	필요한 기술
	<p>9. 규칙 변수 섹션에서 기본값을 유지합니다.</p> <p>10. 규칙 추가 섹션에서 프로토콜용 TLS를 선택합니다. 소스에서 모두를 선택합니다. 소스 포트의에서 모든 포트를 선택합니다. 대상에서 모두를 선택합니다. 대상 포트에서 모든 포트를 선택합니다. 트래픽 방향에서 전달을 선택합니다. 작업에서 알림을 선택합니다. 규칙 추가를 선택합니다.</p> <p>11. 상태 저장 규칙 그룹 생성을 선택합니다.</p>	
<p>상태 저장 규칙을 Network Firewall에 연결합니다.</p>	<ol style="list-style-type: none"> 1. AWS Management Console에 로그인한 후 Amazon VPC 콘솔을 엽니다. 2. 탐색 창의 NETWORK FIREWALL에서 방화벽을 선택합니다. 3. 아웃바운드 트래픽의 SNI에서 서버 이름을 캡처하려는 방화벽을 선택합니다. 4. 상태 저장 규칙 그룹 섹션에서 작업을 선택한 다음 비관리형 상태 저장 규칙 그룹 추가를 선택합니다. 5. 비관리형 상태 저장 규칙 그룹 추가 페이지에서 이전에 생성한 상태 저장 규칙 그룹을 선택한 다음 상태 저장 규칙 그룹 추가를 선택합니다. 	<p>클라우드 관리자</p>

Lambda 함수를 생성하여 로그를 읽습니다.

작업	설명	필요한 기술
<p>Lambda 함수의 코드를 생성합니다.</p>	<p>아웃바운드 트래픽에 대해 Network Firewall의 CloudWatch Logs 이벤트를 읽을 수 있는 통합 개발 환경 (IDE) 에서 다음 Python 3 코드를 붙여넣고 원하는 값으로 <SNS-topic-ARN> 바꿉니다.</p> <pre data-bbox="592 674 1027 1885"> import json import gzip import base64 import boto3 sns_client = boto3.client('sns') def lambda_handler(event, context): decoded_event = json.loads(gzip.decompress(base64.b64decode(event['aws logs']['data']))) body = ''' {filtermatch} '''.format(loggroup= decoded_event['log Group'], logstream =decoded_event['logStream'], filtermatch=decoded_event['logEvents'][0]['message'],) print(body) filterMatch = json.loads(body) </pre>	<p>앱 개발자</p>

작업	설명	필요한 기술
	<pre> data = [] if 'http' in filterMatch['event']: data.append(filterMatch['event']['http']['hostname']) elif 'tls' in filterMatch['event']: data.append(filterMatch['event']['tls']['sni']) result = 'Domain accessed ' + 1* ' ' + (data[0]) + 1* ' ' 'via AWS Network Firewall ' + 1* ' ' + (filterMatch['firewall_name']) print(result) message = {'ServerName': result} send_to_sns = sns_client.publish(TargetArn=<SNS- topic-ARN>, #Replace with the SNS topic ARN Message=json.dumps({'default': json.dumps(message), 'sms': json.dumps(message), 'email': json.dumps(message)}), Subject='Server Name passed through the Network Firewall', MessageStructure='json' </pre>	

작업	설명	필요한 기술
	<p>)</p> <p>이 코드 샘플은 CloudWatch 로그 콘텐츠를 파싱하고 SNI가 HTTPS 헤더에 제공한 서버 이름을 캡처합니다.</p>	
Lambda 함수를 생성합니다.	Lambda 함수를 생성하려면 Lambda 설명서 의 지침을 따르고 런타임을 위한 Python 3.9를 선택하십시오.	클라우드 관리자
Lambda 함수에 코드를 추가합니다.	이전에 생성한 Lambda 함수에 Python 코드를 추가하려면 Lambda 설명서 의 지침을 따르십시오.	클라우드 관리자

작업	설명	필요한 기술
<p>Lambda 함수에 CloudWatch 로그를 트리거로 추가합니다.</p>	<ol style="list-style-type: none"> 1. AWS Management Console에 로그인하고 Lambda 콘솔을 엽니다. 2. 탐색 창에서 함수를 선택한 후, 앞서 생성한 함수를 선택합니다. 3. 함수 개요 섹션에서 트리거 추가를 선택합니다. 4. 트리거 추가 페이지의 트리거 구성 섹션에서 [CloudWatch Logs] 를 선택한 다음 [Add] 를 선택합니다. 5. 로그 그룹의 경우 이전에 만든 CloudWatch 로그 그룹을 선택합니다. 6. 필터 이름에 필터 이름을 입력합니다. 7. 추가를 선택합니다. 8. 함수 페이지의 구성 탭에 있는 트리거 섹션에서 방금 추가한 트리거를 선택한 다음 활성화를 선택합니다. <p>자세한 내용은 Lambda 설명서의 CloudWatch 로그와 함께 Lambda 사용을 참조하십시오.</p>	<p>클라우드 관리자</p>

작업	설명	필요한 기술
<p>SNS 게시 권한을 추가합니다.</p>	<p>Lambda가 API 직접 호출을 통해 메시지를 SNS에 게시할 수 있도록 Lambda 실행 역할에 <code>sns:Publish</code> 권한을 추가합니다.</p> <ol style="list-style-type: none"> 1. 앞서 생성한 Lambda 함수의 실행 역할을 찾으십시오. 2. AWS Identity and Access Management(IAM) 역할에 다음 정책 추가: <pre data-bbox="592 808 1031 1837"> { "Version": "2012-10-17", "Statement": [{ "Sid": "AllowSNSPublish", "Effect": "Allow", "Action": ["sns:GetTopicAttributes", "sns:Subscribe", "sns:Unsubscribe", "sns:Publish"], "Resource": "*" }] }</pre>	<p>클라우드 관리자</p>

SNS 알림의 기능을 테스트합니다.

작업	설명	필요한 기술
Network Firewall을 통해 트래픽을 전송합니다.	<ol style="list-style-type: none"> 1. HTTPS 트래픽이 Network Firewall을 통과할 때까지 전송하거나 기다립니다. 2. 트래픽이 Network Firewall을 통과할 때 AWS에서 받는 SNS 알림 이메일을 확인합니다. 이메일에는 아웃바운드 트래픽에 대한 SNI 세부 정보가 포함되어 있습니다. 예를 들어, 액세스한 도메인 이름이 <code>https://aws.amazon.com</code>이고 구독 프로토콜이 EMAIL-JSON인 경우 위의 Lambda 코드에서 생성된 이메일에는 다음 콘텐츠가 포함됩니다. <pre data-bbox="592 1155 1031 1879"> { "Type": "Notification", "MessageId": "<messageID>", "TopicArn": "arn:aws:sns:us-west-2:123456789:testSNSTopic", "Subject": "Server Name passed through the Network Firewall", "Message": "{\"ServerName\": \"Domain 'aws.amazon.com' accessed via AWS Network Firewall 'AWS-Network-Firew</pre> 	테스트 엔지니어

작업	설명	필요한 기술
	<pre>all-Multi-AZ-firewall \}]", "Timestamp": "2022-03-22T04:10: 04.217Z", "SignatureVersion" : "1", "Signature": "<Signature>", "SigningCertURL": "<SigningCertUrl>", "UnsubscribeURL": "<UnsubscribeURL>" }</pre> <p>그런 다음 Amazon CloudWatch 설명서의 지침에 CloudWatch 따라 Amazon의 Network Firewall 알림 로그를 확인하십시오. 알림 로그는 다음 오류를 보여줍니다.</p> <pre>{ "firewall_name": "AWS-Network-Firewall- all-Multi-AZ-firewall", "availability_zone": "us-east-2b", "event_timestamp": "<event timestamp>", "event": { "timestamp": "2021-03-22T04:10: 04.214222+0000", "flow_id": <flow ID>, "event_type": "alert",</pre>	

작업	설명	필요한 기술
	<pre> "src_ip": "10.1.3.76", "src_port": 22761, "dest_ip": "99.86.59.73", "dest_port": 443, "proto": "TCP", "alert": { "action": "allowed", "signature_id": 2, "rev": 0, "signature": "", "category": "", "severity": 3 }, "tls": { "subject": "CN=aws.amazon.com", "issuerdn": "C=US, O=Amazon, OU=Server CA 1B, CN=Amazon", "serial": "<serial number>", "fingerprint": "<fingerprint ID>", "sni": "aws.amazon.com", "version": "TLS 1.2", "notbefore": "2020-09-30T00:00: 00", </pre>	

작업	설명	필요한 기술
	<pre> "notafter": "2021-09-23T12:00:00", "ja3": {}, "ja3s": {} }, "app_proto": "tls" } }</pre>	

Terraform을 사용하여 조직에서 GuardDuty Amazon을 자동으로 활성화하십시오.

작성자: Aarthi Kannan(AWS)

코드 amazon-guardduty-for-aws리포지토리 : - organizations-with-terraform	환경: 프로덕션	기술: 보안, ID, 규정 준수, 클라우드 네이티브, DevOps
워크로드: 기타 모든 워크로드	AWS 서비스: 아마존 GuardDuty, AWS 조직	

요약

Amazon은 Amazon Web Services (AWS) 계정을 GuardDuty 지속적으로 모니터링하고 위협 인텔리전스를 사용하여 AWS 환경 내에서 예상치 못한 잠재적 악의적 활동을 식별합니다. 여러 AWS 지역에 걸쳐 또는 AWS Management Console을 통해 여러 계정이나 조직을 수동으로 GuardDuty 활성화하는 것은 번거로울 수 있습니다. 클라우드에서 다중 계정, 다중 리전 서비스 및 리소스를 프로비저닝하고 관리할 수 있는 Terraform과 같은 코드형 인프라(IaC) 도구를 사용하여 프로세스를 자동화할 수 있습니다.

AWS는 AWS Organizations를 사용하여 여러 계정을 설정하고 관리할 것을 권장합니다 GuardDuty. 이 패턴은 해당 권장 사항을 준수합니다. 이 접근 방식의 한 가지 이점은 새 계정을 만들거나 조직에 추가할 때 수동 개입 없이 지원되는 모든 지역의 해당 계정에서 자동으로 활성화된다는 GuardDuty 것입니다.

이 패턴은 HashiCorp Terraform을 사용하여 조직의 3개 이상의 Amazon Web Services (AWS) 계정에 GuardDuty 대해 Amazon을 활성화하는 방법을 보여줍니다. 이 패턴과 함께 제공된 샘플 코드는 다음 사항을 수행합니다.

- AWS GuardDuty Organizations에서 대상 조직의 현재 구성원인 모든 AWS 계정에 사용할 수 있습니다.
- 에서 GuardDuty자동 활성화 기능을 활성화합니다. 이 기능은 향후 GuardDuty 대상 조직에 추가되는 모든 계정을 자동으로 활성화합니다.
- 활성화하려는 지역을 선택할 수 있습니다. GuardDuty
- 조직의 보안 계정을 GuardDuty 위임된 관리자로 사용합니다.

- 로깅 계정에 Amazon Simple Storage Service (Amazon S3) 버킷을 생성하고 이 버킷의 모든 계정에서 집계된 결과를 게시하도록 GuardDuty 구성합니다.
- 기본적으로 365일 후에 S3 버킷에서 Amazon S3 Glacier Flexible Retrieval 스토리지로 조사 결과를 전환하는 수명 주기 정책을 할당합니다.

이 샘플 코드를 수동으로 실행할 수도 있고 지속적 통합 및 지속적 전달 (CI/CD) 파이프라인에 통합할 수도 있습니다.

대상 오디언스

이 패턴은 Terraform GuardDuty, Python 및 AWS Organizations를 사용해 본 경험이 있는 사용자에게 권장됩니다.

사전 조건 및 제한 사항

사전 조건

- 활성 상태의 AWS 계정
- 조직은 AWS Organizations에서 설정하며, 조직에는 최소한 다음과 같은 세 개의 계정이 포함됩니다.
 - 관리 계정 - Terraform 코드를 독립 실행형 또는 CI/CD 파이프라인의 일부로 배포하는 계정입니다. Terraform 상태도 이 계정에 저장됩니다.
 - 보안 계정 — 이 계정은 GuardDuty 위임된 관리자로 사용됩니다. 자세한 내용은 [GuardDuty 위임된 관리자를 위한 중요 고려 사항](#) (GuardDuty 설명서) 을 참조하십시오.
 - 로깅 계정 - 이 계정에는 모든 회원 계정의 집계된 결과를 GuardDuty 게시하는 S3 버킷이 포함되어 있습니다.

필수 구성으로 조직을 설정하는 방법에 대한 자세한 내용은 [계정 구조 생성](#)(AWS Well-Architected Labs)을 참조하세요.

- 관리 계정에 Terraform의 상태를 저장하는 원격 백엔드 역할을 하는 Amazon S3 버킷 및 Amazon DynamoDB 테이블. Terraform 상태에 원격 백엔드를 사용하는 방법에 대한 자세한 내용은 [S3 백엔드](#)(Terraform 설명서)를 참조하세요. S3 백엔드로 원격 상태 관리를 설정하는 코드 샘플은 [remote-state-s3-백엔드](#) (Terraform 레지스트리) 를 참조하십시오. 다음과 같은 요구 사항을 확인합니다.
 - S3 버킷과 DynamoDB 테이블은 동일한 리전에 있어야 합니다.
 - DynamoDB 테이블을 생성할 때 파티션 키는 LockID(대소문자 구분)여야 하고 파티션 키 유형은 문자열이어야 합니다. 기타 모든 테이블 설정은 기본값이어야 합니다. 자세한 내용은 [프라이머리 키 정보 및 테이블 생성](#)(DynamoDB 설명서)을 참조하세요.

- 결과를 게시할 S3 버킷에 대한 액세스 로그를 저장하는 데 사용되는 S3 버킷입니다. GuardDuty 자세한 내용은 [Amazon S3 서버 액세스 로깅 활성화](#)(Amazon S3 설명서) 참조하세요. AWS Control Tower 랜딩 존에 배포하는 경우 로그 아카이브 계정의 S3 버킷을 이 용도로 재사용할 수 있습니다.
- Terraform 버전 0.14.6 이상이 설치 및 구성되어 있습니다. 자세한 내용은 [시작하기 - AWS](#)(Terraform 설명서)를 참조하세요.
- Python 버전 3.9.6 이상이 설치 및 구성되었습니다. 자세한 내용은 [소스 릴리스](#)(Python 웹사이트)를 참조하세요.
- AWS SDK for Python(Boto3)이 설치되어 있습니다. 자세한 내용은 [설치](#)(Boto3 설명서)를 참조하세요.
- jq가 설치 및 구성되었습니다. 자세한 내용은 [jq 다운로드](#)(jq 설명서)를 참조하세요.

제한 사항

- 이 패턴은 macOS 및 Amazon Linux 2 운영 체제를 지원합니다. 이 패턴은 Windows 운영 체제에서 사용할 수 있도록 테스트되지 않았습니다.
- GuardDuty 대상 지역의 어떤 계정에서도 이미 활성화되지 않았어야 합니다.
- 이 패턴의 IaC 솔루션은 사전 조건을 배포하지 않습니다.
- 이 패턴은 다음 모범 사례를 준수하는 AWS 랜딩 존을 위해 설계되었습니다.
 - 랜딩 존은 AWS Control Tower를 사용하여 생성되었습니다.
 - 보안 및 로깅에는 별도의 AWS 계정이 사용됩니다.

제품 버전

- Terraform 버전 0.14.6 이상. 샘플 코드는 버전 1.2.8에서 테스트됩니다.
- Python 버전 3.9.6 이상.

아키텍처

이 섹션에서는 이 솔루션과 샘플 코드로 설정된 아키텍처를 개략적으로 설명합니다. 다음 다이어그램은 단일 AWS 리전 내에서 조직의 다양한 계정에 배포된 리소스를 보여줍니다.

1. Terraform은 보안 계정과 로깅 계정에서 GuardDutyTerraformOrgRoleAWS ID 및 액세스 관리 (IAM) 역할을 생성합니다.

2. Terraform은 로깅 계정의 기본 AWS 리전에 S3 버킷을 생성합니다. 이 버킷은 모든 지역 및 조직 내 모든 계정의 모든 GuardDuty 결과를 집계하는 게시 대상으로 사용됩니다. 또한 Terraform은 S3 버킷의 결과를 암호화하는 데 사용되는 AWS Key Management Service(AWS KMS) 키를 보안 계정에 생성하고 S3 버킷의 결과를 S3 Glacier Flexible Retrieval 스토리지로 자동 보관하도록 구성합니다.
3. Terraform은 관리 계정에서 보안 계정을 위임 관리자로 지정합니다. GuardDuty 즉, 이제 보안 계정이 관리 계정을 포함한 모든 회원 계정의 GuardDuty 서비스를 관리합니다. 개별 회원 계정은 GuardDuty 혼자서 일시 중지하거나 비활성화할 수 없습니다.
4. Terraform은 GuardDuty 위임된 관리자를 위해 보안 계정에 GuardDuty 탐지기를 생성합니다.
5. 아직 활성화되지 않은 경우 Terraform은 S3 보호를 활성화합니다. GuardDuty 자세한 내용은 [Amazon에서의 Amazon S3 보호 GuardDuty](#) (GuardDuty 설명서) 를 참조하십시오.
6. Terraform은 조직의 현재 활성 회원 계정을 모두 회원으로 등록합니다. GuardDuty
7. Terraform은 GuardDuty 위임된 관리자가 모든 회원 계정의 집계된 결과를 로깅 계정의 S3 버킷에 게시하도록 구성합니다.
8. Terraform은 선택한 각 AWS 리전에 대해 3~7단계를 반복합니다.

자동화 및 규모 조정

제공된 샘플 코드는 자동화된 배포를 위해 CI/CD 파이프라인에 통합할 수 있도록 모듈화되어 있습니다.

도구

서비스

- [Amazon DynamoDB](#)는 빠르고 예측 가능하고 확장 가능한 성능을 제공하는 완전 관리형 NoSQL 데이터베이스 서비스입니다.
- [GuardDutyAmazon](#)은 로그를 분석하고 처리하여 AWS 환경에서 예상치 못한 잠재적 무단 활동을 식별하는 지속적 보안 모니터링 서비스입니다.
- [Identity and Access Management\(IAM\)](#)를 사용하여 리소스에 대한 액세스를 안전하게 제어할 수 있습니다.
- [AWS Key Management Service\(AWS KMS\)](#)를 사용하면 암호화 키를 생성하고 제어하여 데이터를 보호할 수 있습니다.
- [AWS Organizations](#)은 사용자가 생성하고 중앙에서 관리하는 조직으로 여러 AWS 계정을 통합할 수 있는 계정 관리 서비스입니다.

- [Amazon Simple Storage Service\(S3\)](#)는 원하는 양의 데이터를 저장, 보호 및 검색하는 데 도움이 되는 클라우드 기반 객체 스토리지 서비스입니다.
- [AWS SDK for Python\(Boto3\)](#)는 Python 애플리케이션, 라이브러리 또는 스크립트를 AWS 서비스와 통합하는 데 도움이 되는 소프트웨어 개발 키트입니다.

기타 도구 및 서비스

- [HashiCorp Terraform](#)은 코드를 사용하여 클라우드 인프라 및 리소스를 프로비저닝하고 관리하는 데 도움이 되는 명령줄 인터페이스 애플리케이션입니다.
- [Python](#)은 범용 프로그래밍 언어입니다.
- [jq](#)는 JSON 파일 작업을 도와주는 명령줄 프로세서입니다.

코드 리포지토리

[이 패턴의 코드는 - 저장소에서 GitHub 사용할 수 있습니다. amazon-guardduty-for-aws organizations-with-terraform](#)

에픽

조직 GuardDuty 내에서 활성화하십시오.

작업	설명	필요한 기술
리포지토리를 복제합니다.	Bash 셸에서 다음 명령을 실행합니다. 추가 정보 섹션의 리포지토리 복제에서 리포지토리의 URL이 포함된 전체 명령을 복사할 수 있습니다. GitHub 이렇게 하면 amazon-guardduty-for-awsorganizations-with-terraform-리포지토리 가 복제됩니다 GitHub. <div style="border: 1px solid #ccc; border-radius: 10px; padding: 10px; margin-top: 10px;"> <pre>git clone <github-repository-url></pre> </div>	DevOps 엔지니어

작업	설명	필요한 기술
Terraform 구성 파일을 편집합니다.	<p>1. 복제된 리포지토리의 root 폴더에서 다음 명령을 실행하여 configuration.json.sample 파일을 복제합니다.</p> <pre data-bbox="634 443 1029 600">cp configuration.json.sample configuration.json</pre> <p>2. 새 configuration.json 파일을 편집하고 다음 각 변수의 값을 정의합니다.</p> <ul data-bbox="634 772 1029 1860" style="list-style-type: none"> • management_acc_id - 관리 계정의 계정 ID입니다. • delegated_admin_acc_id - 보안 계정의 계정 ID입니다. • logging_acc_id - 로깅 계정의 계정 ID입니다. • target_regions — 활성화하려는 AWS 지역의 심포로 구분된 목록입니다. GuardDuty • organization_id — 활성화하려는 조직의 AWS 조직 ID GuardDuty. • default_region - Terraform 상태가 관리 계정에 저장되는 리전. 이 지역은 Terraform 백엔드용 S3 버킷 및 DynamoDB 테이블을 배포한 리전과 동일합니다. 	DevOps 엔지니어, 일반 AWS, 테라폼, Python

작업	설명	필요한 기술
	<ul style="list-style-type: none"> • <code>role_to_assume_for_role_creation</code> - 보안 및 로깅 계정의 새 IAM 역할에 할당하려는 이름. 다음 이야기에서 이 새 역할을 생성합니다. Terraform은 이 역할을 맡아 보안 및 로깅 계정에 <code>GuardDutyTerraformOrgRole</code> IAM 역할을 생성합니다. • <code>finding_publishing_frequency</code> — S3 버킷에 결과를 <code>GuardDuty</code> 게시하는 빈도. • <code>guardduty_findings_bucket_region</code> - 게시된 결과를 위한 S3 버킷을 생성하려는 선호 리전. • <code>logging_acc_s3_bucket_name</code> - 버킷에 게시되는 S3 버킷의 기본 이름입니다. • <code>security_acc_kms_key_alias</code> — 결과를 암호화하는 데 사용되는 키의 AWS KMS 별칭. <code>GuardDuty</code> • <code>s3_access_log_bucket_name</code> — 검색 결과에 사용된 S3 버킷에 대한 액세스 로그를 수집하려는 기존 S3 버킷의 이 	

작업	설명	필요한 기술
	<p>름. GuardDuty 이 버킷은 GuardDuty 검색 결과 버킷과 동일한 AWS 리전에 있어야 합니다.</p> <ul style="list-style-type: none"> • <code>tfm_state_backend_s3_bucket</code> - Terraform 원격 백엔드 상태를 저장할 기존 S3 버킷의 이름. • <code>tfm_state_backend_dynamodb_table</code> - Terraform 상태를 잠그기 위한 기존 DynamoDB 테이블의 이름입니다. <p>3. 구성 파일을 저장하고 닫습니다.</p>	

작업	설명	필요한 기술
<p>새 IAM CloudFormation 역할을 위한 템플릿을 생성합니다.</p>	<p>이 패턴에는 두 개의 CloudFormation 템플릿을 생성할 수 있는 IaC 솔루션이 포함됩니다. 이러한 템플릿은 Terraform이 설정 프로세스 중에 사용하는 두 개의 IAM 역할을 생성합니다. 이러한 템플릿은 최소 권한 승인의 보안 모범 사례를 준수합니다.</p> <ol style="list-style-type: none"> 1. Bash 셸의 리포지토리 root 폴더에서 <code>cf-n-templates/</code> 로 이동합니다. 이 폴더에는 스텝이 있는 CloudFormation 템플릿 파일이 들어 있습니다. 2. 다음 명령을 실행합니다. 이렇게 하면 스텝이 <code>configuration.json</code> 파일에 입력한 값으로 대체됩니다. <div data-bbox="630 1199 1029 1360" style="border: 1px solid #ccc; border-radius: 10px; padding: 10px; margin: 10px 0;"> <pre>bash scripts/replace_config_stubs.sh</pre> </div> 3. <code>cf-n-templates/</code> 폴더에 다음 CloudFormation 템플릿이 생성되었는지 확인합니다. <ul style="list-style-type: none"> • <code>management-account-role.yaml</code> — 이 파일에는 관리 계정의 IAM 역할에 대한 역할 정의 및 관련 권한이 들어 있습니다. IAM 역할에는 이 패턴을 완료 	<p>DevOps 엔지니어, 일반 AWS</p>

작업	설명	필요한 기술
	<p>하는 데 필요한 최소 권한이 있습니다.</p> <ul style="list-style-type: none"> role-to-assume-for-role-creation.yaml — 이 파일에는 보안 및 로깅 계정의 IAM 역할에 대한 역할 정의와 관련 권한이 들어 있습니다. Terraform은 이러한 계정에서 역할을 생성하기 위해 이 역할을 말합니다. GuardDuty TerraformOrgRole 	
<p>IAM 역할을 생성합니다.</p>	<p>스택 생성 (CloudFormation 설명서)의 지침에 따라 다음을 수행하십시오.</p> <ol style="list-style-type: none"> 보안 계정과 role-to-assume-for로깅 계정 모두에 -role-creation.yaml 스택을 배포하십시오. .yaml 스택을 관리 계정에 배포합니다. management-account-role 스택을 성공적으로 생성하고 CREATE_COMPLETE 스택 상태를 확인하면 출력에서 이 새 역할의 Amazon 리소스 이름(ARN)을 기록해 둡니다. 	<p>DevOps 엔지니어, 일반 AWS</p>

작업	설명	필요한 기술
관리 계정에서 IAM 역할을 수 입합니다.	보안 모범 사례로서, 진행하기 전에 새 management-account -role IAM 역할을 맡는 것이 좋 습니다. AWS Command Line Interface(AWS CLI)의 추가 정 보 섹션에서 관리 계정 IAM 역 할 맡기에 명령을 입력합니다.	DevOps 엔지니어, 일반 AWS

작업	설명	필요한 기술
설치 스크립트를 실행합니다.	<p>리포지토리 root 폴더에서 다음 명령을 실행하여 설치 스크립트를 시작합니다.</p> <pre data-bbox="597 394 1026 512">bash scripts/full-setup .sh</pre> <p>full-setup.sh 스크립트는 다음 작업을 수행합니다.</p> <ul data-bbox="597 680 1026 1722" style="list-style-type: none"> • 모든 구성 값을 환경 변수로 내보내기 • 각 Terraform 모듈에 대한 backend.tf 및 terraform.tfvars 코드 파일을 생성합니다. • AWS CLI를 통해 조직 GuardDuty 내에서 신뢰할 수 있는 액세스를 가능하게 합니다. • 조직 상태를 Terraform 상태로 가져옵니다. • 로깅 계정에 결과를 게시하기 위한 S3 버킷 생성 • 보안 계정의 탐지 결과를 암호화하기 위한 AWS KMS 키 생성 • 아키텍처 섹션에 설명된 대로 선택한 모든 지역에서 조직 GuardDuty 전체에서 활성화할 수 있습니다. 	DevOps 엔지니어, Python

(선택 사항) 조직 GuardDuty 내 비활성화

작업	설명	필요한 기술
정리 스크립트를 실행합니다.	<p>이 패턴을 사용하여 조직에서 GuardDuty 활성화한 후 사용하지 않도록 설정하려면 GuardDuty 리포지토리 root 폴더에서 다음 명령을 실행하여 cleanup-gd.sh 스크립트를 시작합니다.</p> <pre data-bbox="592 688 1027 810">bash scripts/cleanup-gd.sh</pre> <p>이 스크립트는 대상 조직에서 사용하지 않도록 설정하고 GuardDuty , 배포된 리소스를 모두 제거하고, Terraform을 사용하여 활성화하기 전의 이전 상태로 조직을 복원합니다. GuardDuty</p> <p>참고 이 스크립트는 로컬 및 원격 백엔드에서 Terraform 상태 파일이나 잠금 파일을 제거하지 않습니다. 이 작업을 수행해야 하는 경우 이러한 작업을 수동으로 수행해야 합니다. 또한 이 스크립트는 가져온 조직이나 해당 조직에서 관리하는 계정을 삭제하지 않습니다. 에 대한 신뢰할 수 있는 액세스는 정리 스크립트의 일부로 GuardDuty 비활성화되지 않습니다.</p>	DevOps 엔지니어, 일반 AWS, 테라폼, Python

작업	설명	필요한 기술
IAM 역할을 제거합니다.	role-to-assume-for-role-creation.yaml 및 .yaml 템플릿을 사용하여 생성된 스택을 삭제합니다. management-account-role CloudFormation 자세한 내용은 스택 삭제 (문서) 를 참조하십시오. CloudFormation	DevOps 엔지니어, 일반 AWS

관련 리소스

AWS 설명서

- [여러 계정 관리](#) (GuardDuty 설명서)
- [최소 권한 부여](#)(IAM 설명서)

AWS 마케팅

- [아마존 GuardDuty](#)
- [AWS Organizations](#)

기타 리소스

- [Terraform](#)
- [Terraform CLI 설명서](#)

추가 정보

리포지토리 복제

다음 명령을 실행하여 GitHub 리포지토리를 복제합니다.

```
git clone https://github.com/aws-samples/amazon-guardduty-for-aws-organizations-with-terraform
```

관리 계정 IAM 역할 수입

관리 계정의 IAM 역할을 수행하려면 다음 명령을 실행하세요. <IAM role ARN>를 IAM 역할의 ARN으로 바꿉니다.

```
export ROLE_CREDENTIALS=$(aws sts assume-role --role-arn <IAM role ARN> --role-session-name AWSCLI-Session --output json)
export AWS_ACCESS_KEY_ID=$(echo $ROLE_CREDENTIALS | jq .Credentials.AccessKeyId | sed 's/"//g')
export AWS_SECRET_ACCESS_KEY=$(echo $ROLE_CREDENTIALS | jq .Credentials.SecretAccessKey | sed 's/"//g')
export AWS_SESSION_TOKEN=$(echo $ROLE_CREDENTIALS | jq .Credentials.SessionToken | sed 's/"//g')
```

새 Amazon Redshift 클러스터에 필수 SSL 엔드포인트가 있는지 확인

작성자: Priyanka Chaudhary(AWS)

환경: 프로덕션

기술: 보안, 자격 증명, 규정 준수, 분석, 데이터 레이크

AWS 서비스: AWS, 아마존 CloudWatch 이벤트 CloudTrail, 아마존 Redshift, 아마존 SNS, AWS Lambda

요약

이 패턴은 SSL (보안 소켓 계층) 엔드포인트 없이 새 Amazon Redshift 클러스터가 시작되면 자동으로 알려주는 Amazon Web Services (AWS) CloudFormation 템플릿을 제공합니다.

Amazon Redshift는 완전 관리형 페타바이트 규모 클라우드 기반 데이터 웨어하우스 서비스입니다. 대규모 데이터 세트 저장 및 분석을 위해 설계되었습니다. 또한 대규모 데이터베이스 마이그레이션을 수행하는 데에도 사용됩니다. 보안을 위해 Amazon Redshift는 사용자의 SQL Server 클라이언트 애플리케이션과 Amazon Redshift 클러스터 간의 연결을 암호화하는 SSL을 지원합니다. 클러스터가 SSL 연결을 요구하도록 구성하려면 시작 중에 클러스터와 연결된 파라미터 그룹에서 `require_ssl` 파라미터를 `true`(로) 설정해야 합니다.

이 패턴과 함께 제공되는 보안 제어는 AWS CloudTrail 로그의 Amazon Redshift API 호출을 모니터링하고 [CreateCluster](#), [ModifyClusterRestoreFromClusterSnapshotCreateClusterParameterGroup](#), 및 API에 대한 Amazon CloudWatch Events 이벤트를 시작합니다. [ModifyClusterParameterGroup](#) 이벤트가 이러한 API 중 하나를 감지하면 Python 스크립트를 실행하는 AWS Lambda를 호출합니다. Python 함수는 나열된 CloudWatch 이벤트에 대한 CloudTrail 이벤트를 분석합니다. Amazon Redshift 클러스터가 생성, 수정 또는 기존 스냅샷에서 복원되거나, 클러스터에 대한 새 파라미터 그룹이 생성되거나, 기존 파라미터 그룹이 수정되면 함수가 클러스터의 `require_ssl` 파라미터를 확인합니다. 파라미터 값이 `false`인 경우 함수는 관련 정보(Amazon Redshift 클러스터 이름, AWS 리전, AWS 계정, 알림의 출처가 되는 Lambda의 Amazon 리소스 이름(ARN))와 함께 Amazon Simple Notification Service(Amazon SNS) 알림을 사용자에게 보냅니다.

사전 조건 및 제한 사항

사전 조건

- 활성 상태의 AWS 계정.
- 클러스터 서브넷 그룹이 있는 Virtual Private Cloud(VPC) 및 연결된 보안 그룹.

제한 사항

- 이 보안 제어는 리전별로 적용됩니다. 모니터링하려는 각 AWS 리전에 이를 배포해야 합니다.

아키텍처

대상 아키텍처

자동화 및 규모 조정

- [AWS Organizations](#)를 사용하는 경우, [AWS StackSets Cloudformation](#)을 사용하여 모니터링하려는 여러 계정에 이 템플릿을 배포할 수 있습니다.

도구

서비스

- [AWS CloudFormation](#) — AWS는 AWS CloudFormation 리소스를 모델링 및 설정하고, 빠르고 일관되게 프로비저닝하고, 수명 주기 전반에 걸쳐 관리할 수 있도록 지원합니다. 템플릿을 사용하여 리소스와 해당 종속성을 설명하고, 리소스를 개별적으로 관리하는 대신 스택으로 함께 시작하고 구성할 수 있습니다.
- [Amazon CloudWatch Events](#) — Amazon CloudWatch Events는 AWS 리소스의 변경 사항을 설명하는 시스템 이벤트의 스트림을 거의 실시간으로 제공합니다.
- [AWS Lambda](#) - AWS Lambda는 서버를 프로비저닝하거나 관리하지 않고도 코드를 실행할 수 있도록 지원하는 컴퓨팅 서비스입니다.
- [Amazon Redshift](#) - Amazon Redshift는 클라우드의 완전 관리형 페타바이트 규모 데이터 웨어하우스 서비스입니다.
- [Amazon S3](#)-Amazon Simple Storage Service(S3)는 객체 스토리지 서비스입니다. Amazon S3를 사용하면 인터넷을 통해 언제 어디서든 원하는 양의 데이터를 저장하고 검색할 수 있습니다.

- [Amazon SNS](#) – Amazon Simple Notification Service(SNS)는 웹 서버와 이메일 주소를 포함하여 게시자와 클라이언트 간에 메시지를 전달 또는 전송하는 것을 조정하고 관리합니다. 구독자는 구독하는 주제에 게시된 모든 메시지를 수신하며 주제에 대한 모든 구독자는 동일한 메시지를 수신합니다.

코드

이 패턴에는 다음과 같은 첨부 파일이 포함됩니다.

- RedshiftSSLEndpointsRequired.zip - 보안 제어를 위한 Lambda 코드입니다.
- RedshiftSSLEndpointsRequired.yml— 이벤트 및 Lambda 함수를 설정하는 CloudFormation 템플릿.

에픽

S3 버킷 선택

작업	설명	필요한 기술
S3 버킷을 정의합니다.	Amazon S3 콘솔 에서 Lambda 코드 .zip 파일을 호스팅할 S3 버킷을 선택하거나 생성합니다. 이 S3 버킷은 모니터링하려는 Amazon Redshift 클러스터와 동일한 AWS 리전에 있어야 합니다. S3 버킷 이름은 전역 수준에서 고유하며, 네임스페이스는 모든 AWS 계정이 공유합니다. S3 버킷 이름에는 선행 슬래시를 포함할 수 없습니다.	클라우드 아키텍트
Lambda 코드를 업로드합니다.	첨부 파일 섹션에 제공된 Lambda 코드 .zip 파일을 S3 버킷에 업로드합니다.	클라우드 아키텍트

템플릿을 배포하십시오. CloudFormation

작업	설명	필요한 기술
<p>AWS CloudFormation 템플릿을 실행합니다.</p>	<p>S3 버킷과 동일한 AWS 리전에서 AWS CloudFormation 콘솔을 열고 첨부된 템플릿을 RedshiftSSLEndpointsRequired.yml 배포합니다. AWS CloudFormation 템플릿 배포에 대한 자세한 내용은 CloudFormation 설명서의 AWS CloudFormation 콘솔에 스택 생성을 참조하십시오.</p>	<p>클라우드 아키텍트</p>
<p>템플릿에서 파라미터를 작성합니다.</p>	<p>템플릿을 시작하면 다음 정보를 입력하라는 메시지가 표시됩니다.</p> <ul style="list-style-type: none"> • S3 버킷: 첫 번째 에픽에서 생성하거나 선택한 버킷을 지정합니다. 첨부된 Lambda 코드(.zip 파일)를 업로드한 위치입니다. • S3 키: S3 버킷에 있는 Lambda .zip 파일의 위치를 지정합니다(예: filename.zip 또는 controls/filename.zip). 선행 슬래시를 포함하지 마세요. • 알림 이메일: Amazon SNS 알림을 받으려는 활성 이메일 주소를 입력합니다. • Lambda 로깅 수준: Lambda 함수의 로깅 수준 및 빈도를 지정합니다. 정보를 사용하여 진행 상황에 대한 자세한 	<p>클라우드 아키텍트</p>

작업	설명	필요한 기술
	정보 메시지를 기록하고, 배포를 계속할 수 있는 오류 이벤트의 경우 오류를 기록하고, 잠재적으로 유해한 상황에 대한 경고를 기록할 수 있습니다.	

구독 확인

작업	설명	필요한 기술
구독을 확인합니다.	CloudFormation 템플릿이 성공적으로 배포되면 제공한 이메일 주소로 구독 이메일이 전송됩니다. 위반 알림을 받기 시작하려면 이 이메일 구독을 확인해야 합니다.	클라우드 아키텍트

관련 리소스

- [S3 버킷 생성](#)(Amazon S3 설명서)
- [S3 버킷에 파일 업로드](#)(Amazon S3 설명서)
- [AWS CloudFormation 콘솔에서 스택 생성](#) (AWS CloudFormation 설명서)
- [AWS를 사용하여 AWS API 호출에서 트리거하는 CloudWatch 이벤트 규칙 생성 CloudTrail](#) (AWS CloudTrail 설명서)
- [Amazon Redshift 클러스터 생성](#)(Amazon Redshift 설명서)
- [연결을 위한 보안 옵션 구성](#)(Amazon Redshift 설명서)

첨부

이 문서와 관련된 추가 콘텐츠에 액세스하려면 [attachment.zip](#) 파일의 압축을 풉니다.

새 Amazon Redshift 클러스터가 VPC에서 시작되는지 확인

작성자: Priyanka Chaudhary(AWS)

환경: 프로덕션

기술: 보안, 자격 증명, 규정 준수, 분석, 데이터베이스

AWS 서비스: 아마존 CloudWatch, AWS Lambda, 아마존 Redshift

요약

이 패턴은 Amazon Redshift 클러스터가 가상 사설 클라우드 (VPC) 외부에서 시작되면 자동으로 알려주는 Amazon Web Services (AWS) CloudFormation 템플릿을 제공합니다.

Amazon Redshift는 완전 관리형 페타바이트 규모 클라우드 기반 데이터 웨어하우스 제품입니다. 대규모 데이터 세트 저장 및 분석을 위해 설계되었습니다. 또한 대규모 데이터베이스 마이그레이션을 수행하는 데에도 사용됩니다. Amazon Virtual Private Cloud(Amazon VPC)를 사용하면 정의한 가상 네트워크에서 AWS 리소스(예: Amazon Redshift 클러스터)를 시작할 수 있는 AWS 클라우드의 논리적으로 격리된 섹션을 프로비저닝할 수 있습니다.

이 패턴과 함께 제공되는 보안 제어는 AWS CloudTrail 로그의 Amazon Redshift API 호출을 모니터링하고 및 API에 대한 Amazon CloudWatch Events 이벤트를 시작합니다.

[CreateClusterRestoreFromClusterSnapshot](#) 이벤트가 이러한 API 중 하나를 감지하면 Python 스크립트를 실행하는 AWS Lambda를 호출합니다. Python 함수는 CloudWatch 이벤트를 분석합니다.

Amazon Redshift 클러스터가 스냅샷에서 생성되거나 복원되어 Amazon VPC 네트워크 외부에 나타나는 경우, 함수는 관련 정보(Amazon Redshift 클러스터 이름, AWS 리전, AWS 계정, 알림의 출처가 되는 Lambda의 Amazon 리소스 이름(ARN))와 함께 Amazon Simple Notification Service(Amazon SNS) 알림을 사용자에게 보냅니다.

사전 조건 및 제한 사항

사전 조건

- 활성 상태의 AWS 계정.
- 클러스터 서브넷 그룹이 있는 VPC 및 연결된 보안 그룹.

제한 사항

- AWS CloudFormation 템플릿은 [CreateCluster](#) 및 [RestoreFromClusterSnapshot](#) 작업 (새 클러스터) 만 지원합니다. VPC 외부에서 생성된 기존 Amazon Redshift 클러스터는 감지하지 않습니다.
- 이 보안 제어는 리전별로 적용됩니다. 모니터링하려는 각 AWS 리전에 이를 배포해야 합니다.

아키텍처

대상 아키텍처

자동화 및 규모 조정

[AWS Organizations](#)를 사용하는 경우, [AWS StackSets Cloudformation](#)을 사용하여 모니터링하려는 여러 계정에 이 템플릿을 배포할 수 있습니다.

도구

서비스

- [AWS CloudFormation](#) — AWS는 AWS CloudFormation 리소스를 모델링 및 설정하고, 빠르고 일관되게 프로비저닝하고, 수명 주기 전반에 걸쳐 관리할 수 있도록 지원합니다. 템플릿을 사용하여 리소스와 해당 종속성을 설명하고, 리소스를 개별적으로 관리하는 대신 스택으로 함께 시작하고 구성할 수 있습니다.
- [AWS CloudTrail](#) — AWS는 AWS 계정의 거버넌스, 규정 준수, 운영 및 위험 감사를 구현하도록 CloudTrail 지원합니다. 사용자, 역할 또는 AWS 서비스가 수행한 작업은 에서 이벤트로 기록됩니다 CloudTrail.
- [Amazon CloudWatch Events](#) — Amazon CloudWatch Events는 AWS 리소스의 변경 사항을 설명하는 시스템 이벤트의 스트림을 거의 실시간으로 제공합니다.
- [AWS Lambda](#) – AWS Lambda는 서버를 프로비저닝하거나 관리하지 않고도 코드를 실행하도록 지원하는 컴퓨팅 서비스입니다. AWS Lambda는 필요 시에만 코드를 실행하며, 일일 몇 개의 요청에서 초당 수천 개의 요청까지 자동으로 규모를 조정합니다.
- [Amazon Redshift](#) - Amazon Redshift는 클라우드의 완전 관리형 페타바이트 규모 데이터 웨어하우스 서비스입니다. Amazon Redshift는 데이터 레이크와 통합되므로 데이터를 사용하여 비즈니스 및 고객에 대한 새로운 인사이트를 얻을 수 있습니다.

- [Amazon S3](#) - Amazon Simple Storage Service(Amazon S3)는 웹 사이트, 모바일 애플리케이션, 백업 및 데이터 레이크 등 다양한 스토리지 솔루션에 사용할 수 있는 확장성이 뛰어난 객체 스토리지 서비스입니다.
- [Amazon SNS](#) – Amazon Simple Notification Service(SNS)는 웹 서버와 이메일 주소를 포함하여 게시자와 클라이언트 간에 메시지를 전달 또는 전송하는 것을 조정하고 관리합니다.

코드

이 패턴에는 다음과 같은 첨부 파일이 포함됩니다.

- RedshiftMustBeInVPC.zip - 보안 제어를 위한 Lambda 코드입니다.
- RedshiftMustBeInVPC.yml— 이벤트 및 Lambda 함수를 설정하는 CloudFormation 템플릿.

이러한 파일을 사용하려면 다음 섹션의 지침을 따르세요.

에픽

S3 버킷 선택

작업	설명	필요한 기술
S3 버킷을 정의합니다.	Amazon S3 콘솔 에서 Lambda 코드 .zip 파일을 호스팅할 S3 버킷을 선택하거나 생성합니다. 이 S3 버킷은 모니터링하려는 Amazon Redshift 클러스터와 동일한 AWS 리전에 있어야 합니다. S3 버킷 이름은 전역 수준에서 고유하며, 네임스페이스는 모든 AWS 계정이 공유합니다. S3 버킷 이름에는 선행 슬래시를 포함할 수 없습니다.	클라우드 아키텍트
Lambda 코드를 업로드합니다.	첨부 파일 섹션에 제공된 Lambda 코드(RedshiftMustBeInVPC.zip 파일)를 S3 버킷에 업로드합니다.	클라우드 아키텍트

템플릿을 배포하십시오. CloudFormation

작업	설명	필요한 기술
CloudFormation 템플릿을 실행합니다.	S3 버킷과 동일한 AWS 리전에서 AWS CloudFormation 콘솔 을 열고 첨부된 템플릿 (RedshiftMustBeInVP C.yml)을 배포합니다. AWS CloudFormation 템플릿 배포에 대한 자세한 내용은 CloudFormation 설명서의 AWS CloudFormation 콘솔에 스택 생성 을 참조하십시오.	클라우드 아키텍트
템플릿에서 파라미터를 작성합니다.	<p>템플릿을 시작하면 다음 정보를 입력하라는 메시지가 표시됩니다.</p> <ul style="list-style-type: none"> • S3 버킷: 첫 번째 에픽에서 생성하거나 선택한 버킷을 지정합니다. 첨부된 Lambda 코드(.zip 파일)를 업로드한 위치입니다. • S3 키: S3 버킷에 있는 Lambda .zip 파일의 위치를 지정합니다(예: filename.zip 또는 controls/filename.zip). 선행 슬래시를 포함하지 마세요. • 알림 이메일: Amazon SNS 알림을 받으려는 활성 이메일 주소를 입력합니다. • Lambda 로깅 수준: Lambda 함수의 로깅 수준 및 빈도를 지정합니다. 정보를 사용하여 진행 상황에 대한 자세한 	클라우드 아키텍트

작업	설명	필요한 기술
	정보 메시지를 기록하고, 배포를 계속할 수 있는 오류 이벤트의 경우 오류를 기록하고, 잠재적으로 유해한 상황에 대한 경고를 기록할 수 있습니다.	

구독 확인

작업	설명	필요한 기술
구독을 확인합니다.	CloudFormation 템플릿이 성공적으로 배포되면 제공한 이메일 주소로 구독 이메일이 전송됩니다. 위반 알림을 받기 시작하려면 이 이메일 구독을 확인해야 합니다.	클라우드 아키텍트

관련 리소스

- [S3 버킷 생성](#)(Amazon S3 설명서)
- [S3 버킷에 파일 업로드](#)(Amazon S3 설명서)
- [AWS CloudFormation 콘솔에서 스택 생성](#) (AWS CloudFormation 설명서)
- [AWS를 사용하여 AWS API 호출에서 트리거하는 CloudWatch 이벤트 규칙 생성 CloudTrail](#) (AWS CloudTrail 설명서)
- [Amazon Redshift 클러스터 생성](#)(Amazon Redshift 설명서)

첨부

이 문서와 관련된 추가 콘텐츠에 액세스하려면 [attachment.zip](#) 파일의 압축을 풉니다.

패턴 더 보기

- [Session Manager 및 Amazon EC2 인스턴스 연결을 사용한 Bastion Host 액세스](#)
- [AWS Fargate, PrivateLink AWS 및 Network Load Balancer를 사용하여 Amazon ECS에서 비공개로 컨테이너 애플리케이션에 액세스할 수 있습니다.](#)
- [PrivateLink AWS와 Network Load Balancer를 사용하여 Amazon ECS에서 비공개로 컨테이너 애플리케이션에 액세스할 수 있습니다.](#)
- [PrivateLink AWS와 Network Load Balancer를 사용하여 Amazon EKS에서 비공개로 컨테이너 애플리케이션에 액세스합니다.](#)
- [AMS 계정의 S3 버킷에 대한 EC2 인스턴스 쓰기 액세스 허용](#)
- [한 AWS 계정의 AWS CodeCommit 리포지토리를 다른 계정의 SageMaker Studio와 연결](#)
- [AWS Systems Manager를 사용하여 Windows 레지스트리 항목의 추가 또는 업데이트 자동화](#)
- [AWS CloudFormation 템플릿을 사용하여 AWS Glue에서 암호화 적용을 자동화합니다.](#)
- [Cloud Custodian 및 AWS CDK를 사용하여 Systems Manager용 AWS 관리형 정책을 EC2 인스턴스 프로파일에 자동으로 연결](#)
- [기존 및 새 Amazon EBS 볼륨을 자동으로 암호화하기](#)
- [Cloud Custodian을 사용하여 Amazon RDS에 대한 공개 액세스 차단하기](#)
- [cdk-nag 규칙 팩을 사용하여 AWS CDK 애플리케이션 또는 CloudFormation 템플릿에서 모범 사례를 확인하십시오.](#)
- [시작 시 EC2 인스턴스에 필수 태그가 있는지 확인](#)
- [Amazon DynamoDB에 대한 크로스 계정 액세스 구성](#)
- [Application Load Balancer를 사용하여 Oracle JD Edwards EnterpriseOne on WebLogic Oracle에 대한 HTTPS 암호화를 구성합니다.](#)
- [AWS IoT 환경에서 보안 이벤트에 대한 로깅 및 모니터링을 구성합니다.](#)
- [Amazon EKS에서 실행되는 애플리케이션에 대한 상호 TLS 인증을 구성합니다.](#)
- [pgAdmin에서 SSH 터널을 사용하여 연결](#)
- [AWS CLI를 사용하여 S3 버킷에서 다른 계정 및 리전으로 데이터 복사](#)
- [Amplify를 사용한 React 앱 생성과 Amazon Cognito를 사용한 인증 추가](#)
- [여러 AWS 계정에서 인바운드 인터넷 액세스에 대한 Network Access Analyzer 조사 결과 보고서 생성](#)
- [AWS 네트워크 방화벽에 대한 Amazon CloudWatch 알림을 사용자 지정합니다.](#)
- [AWS Network Firewall과 AWS Transit Gateway를 사용하여 방화벽 배포](#)

- [AWS 랜딩 존 설계를 문서화하십시오](#)
- [Amazon RDS에서 PostgreSQL DB 인스턴스에 대한 암호화된 연결 활성화하기](#)
- [기존 Amazon RDS for PostgreSQL DB 인스턴스 암호화하기](#)
- [시작 시 Amazon RDS 데이터베이스의 자동 태그 지정 적용](#)
- [시작 시 Amazon EMR 클러스터에 태그 지정 적용](#)
- [시작 시 Amazon S3에 Amazon EMR 로깅이 활성화되었는지 확인](#)
- [AWS Config 고급 쿼리를 사용하여 생성 날짜를 기준으로 AWS 리소스 검색](#)
- [트로포스피어를 사용하여 AWS Config 관리형 규칙이 포함된 AWS CloudFormation 템플릿을 생성합니다.](#)
- [AWS KMS 키의 키 상태가 변경될 때 Amazon SNS 알림 받기](#)
- [DynamoDB 태깅 적용 지원](#)
- [Amazon Data Firehose 리소스가 AWS KMS 키로 암호화되지 않은 경우 식별 및 알림](#)
- [AWS CDK를 사용하여 여러 AWS 지역, 계정 및 DevOps OU에서 Amazon Guru를 활성화하여 운영 성능을 개선하십시오.](#)
- [EC2 Windows 인스턴스를 수집하여 AWS Managed Services 계정으로 마이그레이션합니다.](#)
- [AWS DMS를 사용하여 SSL 모드에서 Amazon RDS for Oracle를 Amazon RDS for PostgreSQL로 마이그레이션](#)
- [ELK 스택을 AWS 기반 Elastic 클라우드로 마이그레이션](#)
- [F5 BIG-IP 워크로드를 AWS 클라우드의 F5 BIG-IP VE로 마이그레이션](#)
- [암호화를 사용하지 않는 인스턴스가 있는지 Amazon Aurora를 모니터링](#)
- [컨테이너를 다시 시작하지 않고 데이터베이스 보안 인증 교체](#)
- [신뢰할 수 있는 컨텍스트를 사용하여 AWS에서 Db2 페더레이션 데이터베이스의 사용자 액세스 보호 및 간소화](#)
- [AWS Firewall Manager와 Amazon Data Firehose를 사용하여 AWS WAF 로그를 스프링클에 전송](#)
- [Amazon을 사용하여 VPC를 통해 Amazon S3 버킷의 정적 콘텐츠를 제공합니다. CloudFront](#)
- [인증서 관리자 및 Let's Encrypt를 사용하여 Amazon EKS의 애플리케이션에 대한 end-to-end 암호화를 설정합니다.](#)
- [ELB 로드 밸런서에 TLS 터미널이 필요한지 검증합니다](#)
- [Splunk를 사용하여 AWS 네트워크 방화벽 로그 및 지표 보기](#)
- [Amazon을 사용하여 모든 AWS 계정에 대한 IAM 자격 증명 보고서를 시각화합니다. QuickSight](#)

Serverless

주제

- [AWS Amplify를 사용하여 서버리스 리액트 네이티브 모바일 앱을 구축하십시오.](#)
- [Kinesis Data Streams를 사용하고 AWS CDK를 사용하는 Amazon Data Firehose를 사용하여 Amazon S3에 DynamoDB 레코드를 제공합니다.](#)
- [Amazon API Gateway를 Amazon SQS와 통합하여 비동기식 REST API를 처리합니다.](#)
- [AWS Step Functions에서 AWS Systems Manager Automation 작업을 동기적으로 실행](#)
- [AWS Lambda 함수에서 Python을 사용하여 S3 객체의 병렬 읽기 실행](#)
- [VPC 엔드포인트를 통해 Amazon S3 버킷에 대한 프라이빗 액세스를 설정합니다.](#)
- [서버리스 접근 방식을 사용하여 AWS 서비스를 함께 연결](#)
- [패턴 더 보기](#)

AWS Amplify를 사용하여 서버리스 리액트 네이티브 모바일 앱을 구축하십시오.

작성자: 데크시톨루 펜타코타 (AWS)

코드 리포지토리: aws-amplify-react-native - ios-todo-app	환경: 프로덕션	소스: NA
대상: AWS Amplify AppSync, AWS, 아마존 Cognito, 아마존 DynamoDB	R 유형: 리아키텍트	워크로드: 오픈 소스
기술: 서버리스, 웹 및 모바일 앱	AWS 서비스: AWS Amplify, AWS, 아마존 Cognito AppSync, 아마존 DynamoDB	

요약

이 패턴은 AWS Amplify 및 다음 AWS 서비스를 사용하여 React Native 모바일 앱을 위한 서버리스 백엔드를 생성하는 방법을 보여줍니다.

- AWS AppSync
- Amazon Cognito
- Amazon DynamoDB

Amplify를 사용하여 앱의 백엔드를 구성하고 배포한 후 Amazon Cognito는 앱 사용자를 인증하고 앱 액세스를 승인합니다. AppSync 그런 다음 AWS는 프런트엔드 앱 및 백엔드 DynamoDB 테이블과 상호 작용하여 데이터를 생성하고 가져옵니다.

참고: 이 패턴은 간단한 “ToDoList” 앱을 예로 사용하지만 유사한 절차를 사용하여 모든 React Native 모바일 앱을 만들 수 있습니다.

사전 조건 및 제한 사항

사전 조건

- 활성 상태의 AWS 계정.
- [Amplify 명령줄 인터페이스 \(Amplify CLI\)](#), 설치 및 구성
- XCode (모든 버전)
- Microsoft Visual Studio (모든 버전, 모든 코드 편집기, 모든 텍스트 편집기)
- Amplify에 대한 지식
- Amazon Cognito에 대한 지식
- AWS에 대한 지식 AppSync
- DynamoDB에 대한 지식
- Node.js에 대한 지식
- npm에 대한 지식
- 리액트와 리액트 네이티브에 대한 지식
- ECMAScript 6 JavaScript (ES6) 에 대한 지식
- GraphQL에 대한 지식

아키텍처

다음 다이어그램은 AWS 클라우드에서 React Native 모바일 앱의 백엔드를 실행하기 위한 예제 아키텍처를 보여줍니다.

다이어그램은 다음 아키텍처를 보여줍니다:

1. Amazon Cognito는 앱 사용자를 인증하고 앱 액세스를 승인합니다.
2. 데이터를 생성하고 가져오기 위해 AppSync AWS는 GraphQL API를 사용하여 프론트엔드 앱 및 백엔드 DynamoDB 테이블과 상호 작용합니다.

도구

서비스

- [AWS Amplify](#)는 프론트엔드 웹 및 모바일 개발자가 에서 풀스택 애플리케이션을 빠르고 쉽게 구축할 수 있도록 특별히 제작된 도구 및 기능 세트입니다.

- [AppSyncAWS](#)는 애플리케이션 개발자가 Amazon DynamoDB, AWS Lambda 및 HTTP API를 비롯한 여러 소스의 데이터를 결합할 수 있도록 지원하는 확장 가능한 GraphQL 인터페이스를 제공합니다.
- [Amazon Cognito](#)는 웹 및 모바일 앱에 대한 인증, 권한 부여 및 사용자 관리를 제공합니다.
- [Amazon DynamoDB](#)는 빠르고 예측 가능하고 확장 가능한 성능을 제공하는 완전 관리형 NoSQL 데이터베이스 서비스입니다.

코드

이 패턴에 사용되는 샘플 애플리케이션의 코드는 - 리포지토리에서 사용할 수 있습니다. [GitHub aws-amplify-react-native ios-todo-app](#) 샘플 파일을 사용하려면 이 패턴의 에픽 섹션에 있는 지침을 따르세요.

에픽

리액트 네이티브 앱을 만들고 실행하세요.

작업	설명	필요한 기술
리액트 네이티브 개발 환경을 설정합니다.	자세한 지침은 React Native 문서의 개발 환경 설정 을 참조하십시오.	앱 개발자
iOS 시뮬레이터에서 ToDoList React Native 모바일 앱을 만들고 실행합니다.	<ol style="list-style-type: none"> 1. 새 터미널 창에서 다음 명령을 실행하여 로컬 환경에서 새 React Native 모바일 앱 프로젝트 디렉터리를 생성하세요. <pre>npx react-native init ToDoListAmplify</pre> 2. 다음 명령어를 실행하여 프로젝트의 루트 디렉터리로 이동합니다. <pre>cd ToDoListAmplify</pre> 	앱 개발자

작업	설명	필요한 기술
	<p>3. 다음 명령을 실행하여 앱을 실행합니다.</p> <pre>npx react-native run-ios</pre>	

앱의 새 백엔드 환경을 초기화합니다.

작업	설명	필요한 기술
<p>Amplify에서 앱을 지원하는 데 필요한 백엔드 서비스를 생성합니다.</p>	<ol style="list-style-type: none"> 1. 로컬 환경에서 프로젝트의 루트 디렉토리 (ToDoListAmplify) 에서 다음 명령을 실행합니다. <pre>amplify init</pre> 2. 앱에 대한 정보를 제공하라는 메시지가 나타납니다. 사용 사례를 바탕으로 사용 사례를 바탕으로 필요한 정보를 입력합니다. 그런 다음 Enter 키를 누릅니다. <p>이 패턴에서 사용되는 ToDoList 앱 설정의 경우 다음 예제 구성을 적용하십시오.</p> <p>리액트 네이티브 Amplify 앱 구성 설정 예시</p> <div style="border: 1px solid #ccc; border-radius: 10px; padding: 10px; margin-top: 10px;"> <pre>? Name: ToDoListAmplify ? Environment: dev ? Default editor: Visual Studio Code</pre> </div>	<p>앱 개발자</p>

작업	설명	필요한 기술
	<pre> ? App type: javascript ? Javascript framework : react-native ? Source Directory Path: src ? Distribution Directory Path: / ? Build Command: npm run-script build ? Start Command: npm run-script start ? Select the authentic ation method you want to use: AWS profile ? Please choose the profile you want to use: default </pre> <p>자세한 내용은 Amplify 개발 센터 설명서에서 새 Amplify 백엔드 만들기를 참조하십시오.</p> <p>참고: 이 amplify init 명령은 AWS를 사용하여 다음 리소스를 프로비저닝합니다 CloudFormation.</p> <ul style="list-style-type: none"> 인증된 사용자 및 인증되지 않은 사용자를 위한 AWS Identity 및 Access 	

작업	설명	필요한 기술
	<p>Management (IAM) 역할 (인증 역할 및 비인증 역할)</p> <ul style="list-style-type: none"> • 배포용 Amazon Simple Storage Service(S3) 버킷 (이 패턴의 예제 앱인 Amplify-Meta.json의 경우) • Amplify 호스팅의 백엔드 환경 	

Amplify React 네이티브 앱에 Amazon Cognito 인증을 추가합니다.

작업	설명	필요한 기술
Amazon Cognito 인증 서비스를 생성합니다.	<ol style="list-style-type: none"> 1. 로컬 환경에서는 프로젝트의 루트 디렉터리 (ToDoListAmplify) 에서 다음 명령을 실행합니다. <p><code>amplify add auth</code></p> 2. 인증 서비스의 구성 설정에 대한 정보를 제공하라는 메시지가 나타납니다. 사용 사례를 바탕으로 사용 사례를 바탕으로 필요한 정보를 입력합니다. 그런 다음 Enter 키를 누릅니다. <p>이 패턴에서 사용되는 ToDoList 앱 설정의 경우 다음 예제 구성을 적용하십시오.</p> <p>인증 서비스 구성 설정 예시</p>	앱 개발자

작업	설명	필요한 기술
	<pre> ? Do you want to use the default authentication and security configura tion? \ Default configuration ? How do you want users to be able to sign in? \ Username ? Do you want to configure advanced settings? \ No, I am done </pre> <p>참고: 이 <code>amplify add auth</code> 명령은 프로젝트의 루트 디렉터리 내 로컬 폴더 (<code>amplify</code>) 에 필요한 폴더, 파일 및 종속성 파일을 만듭니다. 이 패턴에서 사용되는 <code>ToDoList</code> 앱 설정의 경우 이 용도로 <code>aws-exports.js</code> 가 생성됩니다.</p>	

작업	설명	필요한 기술
<p>Amazon Cognito 서비스를 AWS 클라우드에 배포합니다.</p>	<ol style="list-style-type: none"> 1. 프로젝트의 루트 디렉터리에서 다음 Amplify CLI 명령을 실행합니다. <code>amplify push</code> 2. 배포를 확인하는 메시지가 나타납니다. Yes를 입력합니다. 그런 다음 Enter 키를 누릅니다. <p>참고: 프로젝트에 배포된 서비스를 보려면 다음 명령을 실행하여 Amplify 콘솔로 이동합니다. <code>amplify console</code></p>	<p>앱 개발자</p>

작업	설명	필요한 기술
<p>React Native에 필요한 Amplify 라이브러리와 CocoaPods iOS용 종속성을 설치합니다.</p>	<ol style="list-style-type: none"> 1. 프로젝트의 루트 디렉터리에서 다음 명령을 실행하여 필요한 Amplify 오픈 소스 클라이언트 라이브러리를 설치합니다. <pre>npm install aws-amplify aws-amplify-react-native \ amazon-cognito-identity-js @react-native-community/ netinfo \ @react-native-async-storage/async-storage</pre> 2. 다음 명령을 실행하여 iOS에 필요한 CocoaPods 종속 항목을 설치합니다. <pre>npx pod-install</pre> 	<p>앱 개발자</p>

작업	설명	필요한 기술
<p>Amplify 서비스를 가져오고 구성합니다.</p>	<p>앱의 진입점 파일 (예: App.js) 에서 다음 코드 줄을 입력하여 Amplify 서비스의 구성 파일을 가져오고 로드합니다.</p> <pre data-bbox="597 443 1027 720">import Amplify from 'aws-amplify' import config from './ src/aws-exports' Amplify.configure e(config)</pre> <p>참고: 앱의 진입점 파일에서 Amplify 서비스를 가져온 후 오류가 발생하면 앱을 중지하십시오. 그런 다음 XCode를 열고 프로젝트의 iOS ToDoListAmplify폴더에서.xcworkspace를 선택한 다음 앱을 실행합니다.</p>	<p>앱 개발자</p>

작업	설명	필요한 기술
<p>With AuthAuthenticator 고차 구성 요소 (HOC) 를 사용하도록 앱의 진입점 파일을 업데이트하십시오.</p>	<p>참고: withAuthenticator HOC는 단 몇 줄의 코드만으로 앱의 로그인, 가입 및 비밀번호 분실 워크플로를 제공합니다. 자세한 내용은 Amplify 개발 센터의 옵션 1: 사전 빌드 UI 구성 요소 사용을 참조하십시오. 또한 React 설명서에 있는 고차 컴포넌트도 있습니다.</p> <ol style="list-style-type: none"> 1. 앱의 진입점 파일 (예: App.js) 에서 다음 코드 줄을 입력하여 withAuthenticator HOC를 가져옵니다. <pre>import { withAuthenticator } from 'aws-amplify-react-native'</pre> 2. 다음 코드를 입력하여 인증자 HOC를 사용하여 내보냅니다. <pre>export default withAuthenticator(App)</pre> <p>인증자 HOC 코드 예제 포함</p> <pre>import Amplify from 'aws-amplify' import config from './src/aws-exports' Amplify.configure(config)</pre>	<p>앱 개발자</p>

작업	설명	필요한 기술
	<pre>import { withAuthenticator } from 'aws-amplify-react-native'; const App = () => { return null; }; export default withAuthenticator(App);</pre> <p>참고: iOS 시뮬레이터에서 앱은 Amazon Cognito 서비스에서 제공하는 로그인 화면을 표시합니다.</p>	

작업	설명	필요한 기술
인증 서비스 설정을 테스트하십시오.	<p>iOS 시뮬레이터에서 다음을 수행합니다.</p> <ol style="list-style-type: none"> 실제 이메일 주소를 사용하여 앱에서 새 계정을 만드세요. 그러면 등록된 이메일로 인증 코드가 전송됩니다. 확인 이메일에서 받은 코드를 사용하여 계정 설정을 확인합니다. 생성한 사용자 아이디와 비밀번호를 입력합니다. 그런 다음 로그인을 선택합니다. 시작 화면이 나타납니다. <p>참고: Amazon Cognito 콘솔을 열고 자격 증명 풀에 새 사용자가 생성되었는지 여부를 확인할 수도 있습니다.</p>	앱 개발자

AWS AppSync API와 DynamoDB 데이터베이스를 앱에 연결

작업	설명	필요한 기술
AWS AppSync API 및 DynamoDB 데이터베이스를 생성합니다.	<ol style="list-style-type: none"> 프로젝트에 AWS AppSync API를 추가하고 프로젝트의 루트 디렉터리에서 다음 Amplify CLI 명령을 실행하여 DynamoDB 데이터베이스를 자동으로 프로비저닝합니다. <pre>amplify add api</pre>	앱 개발자

작업	설명	필요한 기술
	<p>2. API 및 데이터베이스 구성 설정에 대한 정보를 제공하도록 요청하는 프롬프트가 나타납니다. 사용 사례를 바탕으로 사용 사례를 바탕으로 필요한 정보를 입력합니다. 그런 다음 Enter 키를 누릅니다. Amplify CLI는 텍스트 편집기에서 GraphQL 스키마 파일을 엽니다.</p> <p>이 패턴에서 사용되는 ToDoList 앱 설정의 경우 다음 예제 구성을 적용하십시오.</p> <p>API 및 데이터베이스 구성 설정 예시</p> <pre data-bbox="597 1045 1026 1854"> ? Please select from one of the below mentioned services: \ GraphQL ? Provide API name: todolistamplify ? Choose the default authorization type for the API \ Amazon Cognito User Pool Do you want to use the default authentication and security configura tion ? Default configuration How do you want users </pre>	

작업	설명	필요한 기술
	<pre> to be able to sign in? \ Username Do you want to configure advanced settings? \ No, I am done. ? Do you want to configure advanced settings for the GraphQL API \ No, I am done. ? Do you have an annotated GraphQL schema? \ No ? Choose a schema template: \ Single object with fields (e.g., "Todo" with ID, name, description) ? Do you want to edit the schema now? \ Yes </pre> <p>예제 GraphQL 스키마</p> <pre> type Todo @model { id: ID! name: String! description: String } </pre>	

작업	설명	필요한 기술
<p>AWS AppSync API를 배포하십시오.</p>	<ol style="list-style-type: none"> 1. 프로젝트의 루트 디렉터리에서 다음 Amplify CLI 명령을 실행합니다. <p style="text-align: center;"><code>amplify push</code></p> 2. API 및 데이터베이스 구성 설정에 대한 추가 정보를 제공하라는 프롬프트가 나타납니다. 사용 사례를 바탕으로 사용 사례를 바탕으로 필요한 정보를 입력합니다. 그런 다음 Enter 키를 누릅니다. 이제 앱이 AWS AppSync API와 상호 작용할 수 있습니다. <p>이 패턴에서 사용되는 ToDoList 앱 설정의 경우 다음 예제 구성을 적용하십시오.</p> <p>AWS AppSync API 구성 설정 예시</p> <p>참고: 다음 구성은 AppSync AWS에서 GraphQL API를 생성하고 Dynamo DB에 Todo 테이블을 생성합니다.</p> <div style="border: 1px solid #ccc; border-radius: 10px; padding: 10px; margin-top: 10px;"> <pre>? Are you sure you want to continue? Yes ? Do you want to generate code for your newly created GraphQL API Yes</pre> </div>	<p>앱 개발자</p>

작업	설명	필요한 기술
	<pre>? Choose the code generation language target javascript ? Enter the file name pattern of graphql queries, mutations and subscriptions src/ graphql/**/*.js ? Do you want to generate/update all possible GraphQL operations - \ queries, mutations and subscriptions Yes ? Enter maximum statement depth \ [increase from default if your schema is deeply nested] 2</pre>	

작업	설명	필요한 기술
<p>앱의 프론트엔드를 AWS AppSync API에 연결합니다.</p>	<p>이 패턴으로 제공된 예제 ToDoList 앱을 사용하려면 aws-amplify-react-native- ios-todo-app GitHub 리포지토리의 App.js 파일에서 코드를 복사하십시오. 그런 다음 예제 코드를 로컬 환경에 통합하세요.</p> <p>리포지토리의 App.js 파일에 제공된 예제 코드는 다음을 수행합니다.</p> <ul style="list-style-type: none"> • 제목 및 설명 필드가 있는 ToDo 항목을 만들기 위한 양식을 표시합니다. • 할 일 항목 목록 (제목 및 설명)을 표시합니다. • 메서드를 사용하여 데이터를 게시하고 가져옵니다. <code>aws-amplify</code> 	<p>앱 개발자</p>

관련 리소스

- [AWS Amplify](#)
- [Amazon Cognito](#)
- [AWS AppSync](#)
- [Amazon DynamoDB](#)
- [리액트](#) (리액트 문서)

Kinesis Data Streams를 사용하고 AWS CDK를 사용하는 Amazon Data Firehose를 사용하여 Amazon S3에 DynamoDB 레코드를 제공합니다.

작성자: 샤샨크 슈리바스타바(AWS) 및 다니엘 마투키 다 쿠냐(AWS)

코드 리포지토리: [아마존 S3로의 아마존 DynamoDB 통합](#)

환경: PoC 또는 파일럿

기술: 서버리스, 데이터 레이크, 데이터베이스, 스토리지 및 백업

AWS 서비스: AWS CDK; Amazon DynamoDB; Amazon Kinesis Data Firehose; Amazon Kinesis Data Streams; AWS Lambda; Amazon S3

요약

이 패턴은 Amazon Kinesis Data Streams와 Amazon Data Firehose를 사용하여 Amazon DynamoDB에서 아마존 심플 스토리지 서비스 (Amazon S3) 로 레코드를 전송하기 위한 샘플 코드와 애플리케이션을 제공합니다. 이 패턴의 접근 방식은 [AWS Cloud Development Kit\(AWS CDK\) L3 구성](#)을 사용하며, Amazon Web Services (AWS) 클라우드의 대상 S3 버킷으로 데이터를 전송하기 전에 AWS Lambda를 사용하여 데이터 변환을 수행하는 방법에 대한 예제를 포함합니다.

Kinesis Data Streams는 DynamoDB 테이블에서 항목 수준 수정 사항을 기록하여 필수 Kinesis 데이터 스트림에 복제합니다. 애플리케이션에서는 Kinesis 데이터 스트림에 액세스하고 항목 수준 변경 사항을 거의 실시간으로 볼 수 있습니다. 또한 Kinesis Data Streams는 Firehose와 아파치 플링크용 아마존 매니지드 서비스와 같은 다른 Amazon Kinesis 서비스에 대한 액세스를 제공합니다. 즉, 실시간 대시보드를 제공하고, 알림을 생성하고, 동적 요금 및 광고를 구현하고, 정교한 데이터 분석을 수행하는 애플리케이션을 빌드할 수 있습니다.

이 패턴을 데이터 통합 사용 사례에 사용할 수 있습니다. 예를 들어, 운송 차량 또는 산업용 장비가 대량의 데이터를 DynamoDB 테이블로 전송할 수 있습니다. 그런 다음 이 데이터를 변환하여 Amazon S3에 호스팅된 데이터 레이크에 저장할 수 있습니다. 그런 다음 Amazon Athena, Amazon Redshift

Spectrum, Amazon Rekognition 및 AWS Glue와 같은 서버리스 서비스를 사용하여 데이터를 쿼리 및 처리하고 잠재적 결합을 예측할 수 있습니다.

사전 조건 및 제한 사항

사전 조건

- 활성 상태의 AWS 계정.
- AWS Command Line Interface(AWS CLI), 설치 및 구성됨 자세한 내용은 AWS CLI 설명서에서 [AWS CLI 시작하기](#) 섹션을 참조하세요.
- Node.js (18.x+) 및 npm, 설치 및 구성. 자세한 내용은 npm 설명서의 [Node.js 및 npm 다운로드 및 설치](#)를 참조하세요.
- aws-cdk (2.x+), 설치 및 구성. 자세한 내용은 AWS CDK 설명서의 [AWS CDK 시작하기](#)를 참조하세요.
- 로컬 시스템에 GitHub [aws-dynamodb-kinesisfirehose-s복제 및 구성된 3-통합](#) 리포지토리.
- DynamoDB 테이블의 기존 샘플 데이터입니다. 데이터는 {"SourceDataId": {"S": "123"}, "MessageData":{"S": "Hello World"}} 형식을 사용해야 합니다.

아키텍처

다음 다이어그램은 Kinesis Data Streams 및 Firehose를 사용하여 DynamoDB에서 Amazon S3로 레코드를 전송하는 예제 워크플로를 보여줍니다.

이 다이어그램은 다음 워크플로를 보여줍니다.

1. Amazon API Gateway를 사용하여 DynamoDB용 프록시로 데이터를 수집합니다. 다른 소스를 사용하여 DynamoDB로 데이터를 수집할 수도 있습니다.
2. 항목 수준 변경은 Kinesis Data Streams에서 거의 실시간으로 생성되어 Amazon S3로 전송됩니다.
3. Kinesis Data Streams는 변환 및 전송을 위해 Firehose에 레코드를 전송합니다.
4. Lambda 함수는 DynamoDB 레코드 형식의 레코드를 레코드 항목 속성 이름 및 값만 포함하는 JSON 형식으로 변환합니다.

도구

- [AWS Cloud Development Kit\(AWS CDK\)](#)는 AWS 클라우드 인프라를 코드로 정의하고 프로비저닝하는 데 도움이 되는 소프트웨어 개발 프레임워크입니다.
- [AWS CDK Toolkit](#)은 AWS Cloud Development Kit(AWS CDK) 앱과 상호 작용하는 데 도움이 되는 명령줄 클라우드 개발 키트입니다.
- [AWS Command Line Interface \(AWS CLI\)](#)는 명령줄 셸에서 명령을 사용하여 AWS 서비스와 상호 작용할 수 있는 오픈 소스 도구입니다.
- [AWS는 AWS CloudFormation](#) 리소스를 설정하고, 빠르고 일관되게 프로비저닝하고, 전체 AWS 계정 및 지역의 수명 주기 전반에 걸쳐 리소스를 관리할 수 있도록 지원합니다.

code

이 패턴의 코드는 GitHub [aws-dynamodb-kinesisfirehose-s3-통합](#) 리포지토리에서 사용할 수 있습니다.

에픽

샘플 코드 설정 및 구성

작업	설명	필요한 기술
종속성을 설치합니다.	로컬 시스템에서 다음 명령을 실행하여 <code>pattern/aws-dynamodb-kinesisstreams-s3</code> 및 <code>sample-application</code> 디렉터리의 <code>package.json</code> 파일에서 종속성을 설치합니다. <pre>cd <project_root>/pattern/aws-dynamodb-kinesisstreams-s3</pre> <pre>npm install && npm run build</pre>	앱 개발자, 일반 AWS

작업	설명	필요한 기술
	<pre>cd <project_root>/sample-application/</pre> <pre>npm install && npm run build</pre>	
AWS CloudFormation 템플릿을 생성합니다.	<ol style="list-style-type: none"> 1. <code>cd <project_root>/sample-application/</code> 명령을 실행합니다. 2. <code>cdk synth</code> 명령을 실행하여 AWS CloudFormation 템플릿을 생성합니다. 3. <code>cdk.out</code> 디렉터리에 <code>AwsDynamodbKinesisFirehoseS3IngestionStack.template.json</code> 출력이 저장됩니다. 4. AWS CDK 또는 AWS 관리 콘솔을 사용하여 CloudFormation AWS에서 템플릿을 처리하십시오. 	앱 개발자, 일반 AWS, AWS DevOps

리소스 배포

작업	설명	필요한 기술
리소스를 확인하고 배포하세요.	1. <code>cdk diff</code> 명령을 실행하여 AWS CDK 구성으로 생성된 리소스 유형을 식별합니다.	앱 개발자, 일반 AWS, AWS DevOps

작업	설명	필요한 기술
	2. cdk deploy 명령을 실행하여 리소스를 배포합니다.	

DynamoDB 테이블로 데이터를 수집하여 솔루션 테스트

작업	설명	필요한 기술
샘플 데이터를 DynamoDB 테이블로 수집합니다.	<p>1. AWS CLI에서 다음 명령을 실행하여 DynamoDB 테이블에 요청을 전송합니다.</p> <pre>aws dynamodb put-item --table-name <your_table_name> --item '{"<table_partition_key>": {"S": "<partition_key_ID>"}, "MessageData": {"S": "<data>"}}'</pre> <p>예:</p> <pre>aws dynamodb put-item --table-name SourceData_table --item '{"SourceDataId": {"S": "123"}, "MessageData": {"S": "Hello World"}}'</pre> <p>기본적으로 put-item은(는) 작업이 성공해도 출력으로 어떤 값도 반환하지 않습니다.</p>	앱 개발자

작업	설명	필요한 기술
	<p>작업이 실패하면 오류가 반환됩니다. 데이터는 DynamoDB에 저장된 다음 Kinesis Data Streams 및 Firehose로 전송됩니다.</p> <p>참고: DynamoDB 테이블에 데이터를 추가할 때는 다양한 접근 방식을 사용합니다. 자세한 내용은 Amazon DynamoDB 설명서의 테이블에 데이터 로드를 참조하세요.</p>	
S3 버킷에 새 객체가 생성되었는지 확인합니다.	<p>AWS Management Console에 로그인하고 S3 버킷을 모니터링하여 전송한 데이터로 새 객체가 생성되었는지 확인합니다.</p> <p>자세한 내용은 Amazon S3 API 참조 설명서의 <code>get-object</code>를 참조하세요.</p>	앱 개발자, 일반 AWS

리소스 정리

작업	설명	필요한 기술
리소스를 정리합니다.	cdk destroy 명령을 실행하여 이 패턴에 사용되는 모든 리소스를 삭제합니다.	앱 개발자, 일반 AWS

관련 리소스

- [s3-.ts \(static-site-stack리포지토리\)](#) GitHub

- [aws-apigateway-dynamodb 모듈](#) (GitHub 리포지토리)
- [aws-kinesisstreams-kinesisfirehose-s3 모듈](#) (GitHub 리포지토리)
- [DynamoDB Streams에 대한 변경 데이터 캡처](#) (Amazon DynamoDB 설명서)
- [Kinesis Data Streams을 사용하여 DynamoDB에 대한 변경 사항 캡처](#) (Amazon DynamoDB 설명서)

Amazon API Gateway를 Amazon SQS와 통합하여 비동기식 REST API를 처리합니다.

제작: 나탈리아 콜란토니오 파베로 (AWS) 와 구스타보 마르팀 (AWS)

환경: PoC 또는 파일럿

기술: 서버리스, 메시징 및 통신

AWS 서비스: 아마존 SQS, 아마존 API 게이트웨이

요약

REST API를 배포할 때 클라이언트 애플리케이션이 게시할 수 있는 메시지 대기열을 노출해야 하는 경우가 있습니다. 예를 들어, 타사 API의 지연 시간 및 응답 지연 문제가 있거나, 데이터베이스 쿼리의 응답 시간을 피하거나 동시 API가 많을 때 서버를 확장하지 않는 것이 좋습니다. 이러한 시나리오에서 대기열에 게시하는 클라이언트 애플리케이션은 API가 데이터를 수신했다는 사실만 알면 되며 데이터를 받은 후에 어떤 일이 발생하는지는 알 수 없습니다.

이 패턴은 [Amazon API Gateway를 사용하여 아마존 심플 큐 서비스 \(Amazon SQS\)](#) 에 메시지를 전송함으로써 REST API 엔드포인트를 생성합니다. 그러면 두 서비스가 easy-to-implement 통합되므로 SQS 대기열에 직접 액세스할 수 없습니다.

사전 조건 및 제한 사항

- [활성 계정 AWS](#)

아키텍처

다이어그램은 다음 단계를 보여줍니다.

1. Postman, 다른 API 또는 기타 기술과 같은 도구를 사용하여 POST REST API 엔드포인트를 요청하십시오.
2. API Gateway는 요청 본문에서 수신한 메시지를 대기열에 게시합니다.
3. Amazon SQS는 메시지를 수신하고 성공 또는 실패 코드가 포함된 답변을 API Gateway에 보냅니다.

도구

- [Amazon API Gateway](#)를 사용하면 어떤 규모에서든 REST, HTTP 및 WebSocket API를 생성, 게시, 유지 관리, 모니터링 및 보호할 수 있습니다.
- [AWS Identity and Access Management \(IAM\)](#) 을 사용하면 누가 인증 및 사용 권한을 받았는지 제어 하여 AWS 리소스에 대한 액세스를 안전하게 관리할 수 있습니다.
- [Amazon Simple Queue Service\(Amazon SQS\)](#)는 분산 소프트웨어 시스템과 구성 요소를 통합하고 분리하는 데 도움이 되는 안전하고 내구성이 뛰어나며 가용성이 높은 호스팅 대기열을 제공합니다.

에픽

SQS 대기열 생성

작업	설명	필요한 기술
대기열을 생성합니다.	<p>REST API로부터 메시지를 수신하는 SQS 대기열을 만들려면:</p> <ol style="list-style-type: none"> 1. AWS 계정에 로그인합니다. 2. https://console.aws.amazon.com/sqs/에서 Amazon SQS 콘솔을 엽니다. 3. 대기열 생성을 선택합니다. 4. 대기열 생성 페이지의 지역 드롭다운 AWS 리전 목록에서 올바른 대기열을 선택합니다. 5. 유형의 경우 기본 설정 (표준) 을 유지합니다. 6. 대기열의 이름을 입력합니다. 7. 다른 모든 설정의 기본값은 그대로 유지합니다. 8. 대기열 생성을 선택합니다. 	앱 개발자

Amazon SQS에 대한 액세스 제공

작업	설명	필요한 기술
IAM 역할을 생성합니다.	<p>이 IAM 역할은 API Gateway 리소스에 Amazon SQS에 대한 전체 액세스 권한을 부여합니다.</p> <ol style="list-style-type: none"> 1. https://console.aws.amazon.com/iam/에서 IAM 콘솔을 엽니다. 2. 탐색 창에서 역할(Roles), 역할 생성(Create role)을 선택합니다. 3. 신뢰할 수 있는 엔터티 유형에 AWS 서비스를 선택합니다. 4. 사용 사례의 경우 드롭다운 목록에서 API Gateway를 선택한 후 다음, 다음을 선택합니다. 5. 역할 이름에 설명 (선택 사항) 을 AWSGatewayRoleForSQS입력하고 역할 생성을 선택합니다. 6. 역할 창에서 해당 확인란을 검색하여 선택합니다. AWSGatewayRoleForSQS 7. 권한 정책 섹션에서 권한 추가, 정책 연결을 차례로 선택합니다. 8. AmazonSQS를 FullAccess 검색하고 선택합니다. 9. 권한 추가를 선택합니다. 	앱 개발자, AWS 관리자

작업	설명	필요한 기술
	10의 AWSGatewayRoleForSQS요약 섹션에서 Amazon 리소스 번호 (ARN) 를 복사합니다. 이 ID는 이후 단계에서 사용하게 됩니다.	

REST API 생성

작업	설명	필요한 기술
REST API를 생성하십시오.	<p>이것은 HTTP 요청이 전송되는 REST API입니다.</p> <ol style="list-style-type: none"> https://console.aws.amazon.com/apigateway/에서 Amazon API Gateway 콘솔을 엽니다. REST API 섹션에서 빌드를 선택합니다. API 이름에 API의 이름과 선택적 설명을 입력하고 다른 모든 기본 설정을 유지한 다음 API 생성을 선택합니다. 	앱 개발자
API Gateway를 아마존 SQS에 연결합니다.	<p>이 단계를 통해 메시지가 HTTP 요청 본문 내에서 Amazon SQS로 전달될 수 있습니다.</p> <ol style="list-style-type: none"> API Gateway 콘솔에서 생성한 API를 선택합니다. 리소스 페이지의 방법 섹션에서 메서드 생성을 선택합니다. 	앱 개발자

작업	설명	필요한 기술
	<ol style="list-style-type: none"> 3. 메서드 유형에서 POST를 선택합니다. 4. 통합 유형에서 선택합니다 AWS 서비스. 5. 에서 SQS 대기열을 생성한 지역을 선택합니다. AWS 리전 6. 의 AWS 서비스경우 단순 대기열 서비스 (SQS) 를 선택합니다. 7. HTTP 메서드의 경우 POST 를 선택합니다. 8. 작업 유형에서 경로 재정의 사용을 선택합니다. 9. 경로 재정의에 <name of SQS queue>/를 <AWS account ID>입력합니다. 10.실행 역할에는 이전에 생성한 역할의 ARN을 붙여넣습니다. 11.메서드 생성을 선택합니다. 	

REST API를 테스트하세요.

작업	설명	필요한 기술
<p>REST API를 테스트해 보세요.</p>	<p>테스트를 실행하여 누락된 구성이 있는지 확인합니다.</p> <ol style="list-style-type: none"> 1. API Gateway 콘솔에서 생성한 REST API를 선택합니다. 2. 리소스 창에서 POST 메서드를 선택합니다. 	<p>앱 개발자</p>

작업	설명	필요한 기술
	<p>3. 테스트 탭을 선택합니다. (탭이 표시되지 않으면 오른쪽 화살표를 사용하십시오.)</p> <p>4. 요청 본문에 다음 JSON 코드를 붙여넣습니다.</p> <pre data-bbox="630 483 1029 682"> { "message": "lorem ipsum" } </pre> <p>5. 테스트를 선택합니다.</p> <p>다음과 비슷한 오류가 표시됩니다.</p> <pre data-bbox="630 898 1029 1016"> <UnknownOperationE xception/> </pre>	

작업	설명	필요한 기술
요청을 Amazon SQS에 제대로 전달하도록 API 통합을 변경하십시오.	<p>구성을 완료하여 통합 오류를 수정하십시오.</p> <ol style="list-style-type: none"> 1. API Gateway 콘솔에서 생성한 API를 선택한 다음 POST를 선택합니다. 2. 메서드 실행 섹션은 API Gateway와 Amazon SQS 간의 시각적 매핑을 보여줍니다. 이 섹션에서 통합 요청을 선택한 다음 편집을 선택합니다. 3. HTTP 헤더 섹션을 확장한 다음 요청 헤더 파라미터 추가를 선택합니다. <ul style="list-style-type: none"> • 이름에는 콘텐츠 유형을 지정합니다. • 매핑 원본에 '애플리케이션/'을 입력합니다. x-www-form-urlencoded • 캐싱 체크박스를 선택합니다. 4. 매핑 템플릿 섹션을 확장합니다. <ul style="list-style-type: none"> • 매핑 템플릿 추가(Add mapping template)를 선택합니다. • 콘텐츠 유형에는 애플리케이션/json을 입력합니다. • 템플릿 본문에 다음 코드를 붙여넣습니다. 	앱 개발자

작업	설명	필요한 기술
	<pre>Action=SendMessage &MessageBody=\$input.body</pre> <ul style="list-style-type: none">• 저장을 선택합니다.	

작업	설명	필요한 기술
<p>Amazon SQS에서 메시지를 테스트하고 검증합니다.</p>	<p>테스트를 실행하여 테스트가 성공적으로 완료되었는지 확인합니다.</p> <ol style="list-style-type: none"> 1. API Gateway 콘솔에서 생성한 REST API를 선택합니다. 2. 리소스 창에서 POST 메서드를 선택합니다. 3. 테스트 탭을 선택합니다. (탭이 표시되지 않으면 오른쪽 화살표를 사용하십시오.) 4. 요청 본문에 다음 JSON 코드를 붙여넣습니다. <pre data-bbox="630 884 1029 1083"> { "message": "lorem ipsum" }</pre> 5. 테스트를 선택합니다. 6. Amazon SQS 콘솔을 엽니다. 7. 탐색 창에서 [Queues] 를 선택한 다음 대기열을 선택합니다. 8. [메시지 전송 및 수신(Send and receive messages)]을 선택합니다. 9. 메시지 폴링을 선택합니다. 10. 메시지를 선택합니다. 다음과 같이 표시되어야 합니다. <pre data-bbox="630 1745 1029 1864"> Body { "message": "lorem ipsum" }</pre> 	<p>앱 개발자</p>

작업	설명	필요한 기술
<p>특수 문자를 사용하여 API Gateway를 테스트합니다.</p>	<p>메시지에 허용되지 않는 특수 문자 (예: &) 가 포함된 테스트를 실행합니다.</p> <ol style="list-style-type: none"> 1. API Gateway 콘솔에서 API를 선택합니다. 2. 다음 JSON 코드를 사용하여 이전 단계의 테스트를 반복합니다. <pre data-bbox="630 674 1029 873"> { "message": "lorem ipsum &" } </pre> <ol style="list-style-type: none"> 3. 테스트를 선택합니다. <p>다음과 같은 오류가 표시됩니다.</p> <pre data-bbox="630 1087 1029 1837"> { "Error": { "Code": "AccessDenied", "Message": "Access to the resource https://s qs.us-east-2.amazo naws.com/976166761 794/Apg2 is denied.", "Type": "Sender" }, "RequestId": "e83c9c67-bcf6-5e9 a-91e9-c737094b17a b" } </pre>	<p>앱 개발자</p>

작업	설명	필요한 기술
	<p>이는 메시지 본문에서 특수 문자가 기본적으로 지원되지 않기 때문입니다. 다음 단계에서는 특수 문자를 지원하도록 API Gateway를 구성합니다. 콘텐츠 유형 변환에 대한 자세한 내용은 API Gateway 설명서를 참조하십시오.</p>	

작업	설명	필요한 기술
<p>특수 문자를 지원하도록 API 구성을 변경하십시오.</p>	<p>메시지에 특수 문자를 허용하도록 구성을 조정하십시오.</p> <ol style="list-style-type: none"> 1. API Gateway 콘솔에서 생성한 API를 선택한 다음 POST를 선택합니다. 2. 통합 요청을 선택한 다음 편집을 선택합니다. 3. 콘텐츠 처리를 텍스트로 변환하도록 변경합니다. 4. 매핑 템플릿 섹션에서: <ul style="list-style-type: none"> • 콘텐츠 유형에는 애플리케이션/json을 입력합니다. • 템플릿 본문의 경우 다음을 지정합니다. <pre data-bbox="662 1045 1029 1247">Action=SendMessage &MessageBody=\${util .urlEncode(\$input. body)</pre> <ul style="list-style-type: none"> • 저장을 선택합니다. 5. 테스트 탭을 선택합니다. 6. 요청 본문에는 이전의 JSON 코드를 입력합니다. <pre data-bbox="630 1495 1029 1654">{ " message": "lorem ipsum &" }</pre> 7. 테스트를 선택합니다. 8. Amazon SQS 콘솔을 엽니다. 	<p>앱 개발자</p>

작업	설명	필요한 기술
	<p>9. 대기열을 선택한 다음 메시지 보내기 및 받기, 메시지 설문 조사, 이전과 같은 메시지를 선택합니다.</p> <p>새 메시지에는 특수 문자가 포함되어야 합니다.</p>	

REST API를 배포하세요.

작업	설명	필요한 기술
API를 배포합니다.	<p>REST API를 배포하려면:</p> <ol style="list-style-type: none"> 1. API Gateway 콘솔을 엽니다. 2. API를 선택합니다. 3. Deploy API(API 배포)를 선택합니다. 이 단계에 대한 자세한 내용은 API Gateway 설명서를 참조하십시오. 	앱 개발자
외부 도구를 사용하여 테스트하세요.	<p>외부 도구를 사용하여 테스트를 실행하여 메시지가 성공적으로 수신되었는지 확인합니다.</p> <ol style="list-style-type: none"> 1. 포스트맨, 인섬니아 또는 cURL과 같은 도구를 엽니다. 2. API를 실행하세요. 3. Amazon SQS 콘솔을 엽니다. 	앱 개발자

작업	설명	필요한 기술
	4. 대기열을 선택합니다. 5. 메시지를 로드하면 새 메시지를 볼 수 있습니다.	

정리

작업	설명	필요한 기술
API를 삭제합니다.	API Gateway 콘솔에서 생성한 API를 선택한 다음 삭제를 선택합니다.	앱 개발자
IAM 역할을 삭제합니다.	IAM 콘솔의 역할 창에서 을 선택한 다음 삭제를 선택합니다 AWSGatewayRoleForSQS.	앱 개발자
SQS 대기열을 삭제합니다.	Amazon SQS 콘솔의 [대기열] 창에서 생성한 SQS 대기열을 선택한 다음 [Delete] 를 선택합니다.	앱 개발자

관련 리소스

- [SQS- SendMessage](#) (API 게이트웨이 설명서)
- [API Gateway에서의 콘텐츠 유형 변환](#) (API Gateway 설명서)
- [\\$util 변수](#) (API Gateway 설명서)
- [API Gateway REST API를 Amazon SQS와 통합하고 일반적인 오류를 해결하려면 어떻게 해야 합니까?](#) (기사AWS 재게시)

AWS Step Functions에서 AWS Systems Manager Automation 작업을 동기적으로 실행

작성자: Elie El khoury(AWS)

코드 리포지토리: [amazon-stepfunctions-ssm-waitfortask-token](#)

환경: 프로덕션

기술: 서버리스, 운영 DevOps, 최종 사용자 컴퓨팅

워크로드: 기타 모든 워크로드

AWS 서비스: AWS Systems Manager, AWS Step Functions

요약

이 패턴은 Amazon Web Services(AWS) Step Functions를 AWS Systems Manager와 통합하는 방법을 설명합니다. AWS SDK 서비스 통합을 사용하여 상태 머신 워크플로의 작업 토큰으로 AWS Systems Manager startAutomationExecutionAPI를 호출하고 성공 또는 실패 호출과 함께 토큰이 반환될 때까지 일시 중지합니다. 통합을 시연하기 위해 이 패턴은 AWS-RunShellScript 문서 주변에 자동화 문서(런북) 래퍼를 구현하고 .waitForTaskToken을 사용하여 AWS-RunShellScript를 동기적으로 직접 호출합니다. Step Functions의 AWS SDK 서비스 통합에 대한 자세한 내용은 [AWS Step Functions 개발자 안내서](#) 참조하십시오.

AWS Step Functions는 코드가 적은 시각적 워크플로 서비스로 AWS 서비스를 사용하여 분산 애플리케이션을 구축하고, IT 및 비즈니스 프로세스를 자동화하고, 데이터 및 기계 학습 파이프라인을 구축하는 데 사용할 수 있습니다. 워크플로는 장애, 재시도, 병렬화, 서비스 통합 및 관찰성을 관리하므로 더 중요한 비즈니스 로직에 집중할 수 있습니다.

AWS Systems Manager의 기능인 Automation은 Amazon Elastic Compute Cloud(Amazon EC2), Amazon Relational Database Service(RDS), Amazon Redshift 및 Amazon Simple Storage Service(S3)와 같은 AWS 서비스에 대한 일반적인 유지 관리, 배포 및 수정 작업을 단순화합니다. Automation을 사용하면 자동화의 동시성을 세부적으로 제어할 수 있습니다 예를 들어 동시에 대상으로 지정할 리소스 수와 자동화가 중지되기 전에 발생할 수 있는 오류 수를 지정할 수 있습니다.

런북 단계, 파라미터 및 예제를 포함한 구현 세부 정보는 [추가 정보](#) 섹션을 참조하십시오.

사전 조건 및 제한 사항

사전 조건

- 활성 상태의 AWS 계정
- AWS Step Functions 및 AWS Systems Manager에 액세스할 수 있는 AWS Identity and Access Management(IAM) 권한
- 인스턴스에 AWS Systems Manager Agent(SSM Agent)가 [설치된](#) EC2 인스턴스
- 런북을 실행하려는 인스턴스에 연결된 [Systems Manager용 IAM 인스턴스 프로파일](#)
- 다음과 같은 IAM 권한을 가진 Step Functions 역할(최소 권한 원칙 준수).

```
{
    "Effect": "Allow",
    "Action": "ssm:StartAutomationExecution",
    "Resource": "*"
}
```

제품 버전

- SSM 문서 스키마 버전 0.3 이상
- SSM Agent 버전 2.3.672.0 이상

아키텍처

대상 기술 스택

- AWS Step Functions
- AWS Systems Manager Automation

대상 아키텍처

자동화 및 규모 조정

- 이 패턴은 여러 인스턴스에 런북을 배포하는 데 사용할 수 있는 CloudFormation 템플릿을 제공합니다. (Step GitHub [Functions 및 Systems Manager 구현](#) 리포지토리를 참조하십시오.)

도구

서비스

- [AWS는 AWS CloudFormation](#) 리소스를 설정하고, 빠르고 일관되게 프로비저닝하고, AWS 계정 및 지역 전반의 수명 주기 전반에 걸쳐 리소스를 관리할 수 있도록 지원합니다.
- [AWS Identity and Access Management\(IAM\)](#)는 누구에게 인증 및 사용 권한이 있는지 제어하여 AWS 리소스에 대한 액세스를 안전하게 관리할 수 있도록 도와줍니다.
- [AWS Step Functions](#)은 Lambda 함수와 기타 AWS 서비스를 결합할 수 있는 서버리스 오케스트레이션 서비스로, 비즈니스 크리티컬 애플리케이션을 구축합니다.
- [AWS Systems Manager](#)는 AWS 클라우드에서 실행되는 애플리케이션과 인프라를 관리하는 데 도움이 됩니다. 애플리케이션 및 리소스 관리를 간소화하고, 운영 문제의 감지 및 해결 시간을 단축하며, AWS 리소스를 규모에 따라 안전하게 관리하는 데 도움이 됩니다.

코드

이 패턴의 코드는 Step GitHub [Functions 및 Systems Manager 구현](#) 리포지토리에서 사용할 수 있습니다.

에픽

런북 생성

작업	설명	필요한 기술
CloudFormation 템플릿을 다운로드하십시오.	GitHub 리포지토리 <code>cloudformation</code> 폴더에서 <code>ssm-automation-documents.cfn.json</code> 템플릿을 다운로드합니다.	AWS DevOps
런북을 생성하십시오.	AWS Management 콘솔에 로그인하고, AWS CloudFormation 콘솔 을 열고, 템플릿을 배포합니다. AWS CloudFormation 템플릿 배포에 대한 자세한 내용은 CloudFormation 설명서의 AWS CloudFormation	AWS DevOps

작업	설명	필요한 기술
	<p>콘솔에 스택 생성을 참조하십시오. CloudFormation 템플릿은 세 가지 리소스를 배포합니다.</p> <ul style="list-style-type: none"> • SfnRunCommandByInstanceIds - 인스턴스 ID를 사용하여 AWS-RunShellScript 를 실행할 수 있는 런북 • SfnRunCommandByTargets - 대상을 사용하여 AWS-RunShellScript 를 실행할 수 있는 런북 • SSMSyncRole - 런북에서 맡는 IAM 역할 	

샘플 상태 머신 생성

작업	설명	필요한 기술
<p>테스트 상태 머신을 생성하십시오.</p>	<p>AWS Step Functions 개발자 안내서의 지침에 따라 상태 머신을 생성하고 실행하십시오. 정의를 위해 다음 코드를 사용하십시오. 계정에 있는 유효한 Systems Manager 활성화 인스턴스의 ID로 InstanceIds 값을 업데이트해야 합니다.</p> <pre data-bbox="594 1671 1029 1843"> { "Comment": "A description of my state machine", </pre>	<p>AWS DevOps</p>

작업	설명	필요한 기술
	<pre> "StartAt": "StartAutomationWaitForCallback", "States": { "StartAutomationWaitForCallback": { "Type": "Task", "Resource": "arn:aws:states:::aws-sdk:ssm:startAutomationExecution.waitForTaskToken", "Parameters": { "DocumentName": "SfnRunCommandByInstanceIds", "Parameters": { "InstanceIds": ["i-1234567890abcdef0"], "taskToken.\$": "States.Array(\$.Task.Token)", "workingDirectory": ["/home/ssm-user/"], "Commands": ["echo \"This is a test running automation waitForTaskToken\" >> automation.log", "sleep 100"] } }, "End": true } } } </pre>	

작업	설명	필요한 기술
	<pre data-bbox="592 210 1031 304"> } } </pre> <p data-bbox="592 336 1031 997"> 이 코드는 런북을 직접 호출하여 Systems Manager Automation에 대한 <code>waitForTaskToken</code> 직접 호출을 보여주는 두 개의 명령을 실행합니다. 태스크는 <code>/home/ssm-user/automation.log</code> 파일에 “이것은 자동화 <code>waitForTask</code> 토큰을 실행하는 테스트입니다”라고 쓴 다음 100초 동안 휴면 상태가 된 후 작업 토큰으로 응답하여 워크플로우에서 다음 작업을 릴리스합니다. </p> <p data-bbox="592 1039 1031 1270"> 대신 <code>SfnRunCommandByTargets</code> 런북을 직접 호출하려면 이전 코드의 <code>Parameters</code> 섹션을 다음과 같이 바꾸십시오. </p> <pre data-bbox="592 1312 1031 1869"> "Parameters": { "Targets": [{ "Key": "InstanceIds", "Values": ["i-02573cafcfEXAMPLE", "i-0471e04240EXAMPLE"] }] } </pre>	

작업	설명	필요한 기술
	<pre>],</pre>	
<p>상태 머신의 IAM 역할을 업데이트합니다.</p>	<p>이전 단계에서는 상태 머신을 위한 전용 IAM 역할을 자동으로 생성합니다. 하지만 런북 직접 호출 권한은 부여하지 않습니다. 다음 권한을 추가하여 역할을 업데이트합니다.</p> <pre>{ "Effect": "Allow", "Action": "ssm:StartAutomationExecution", "Resource": "*" }</pre>	<p>AWS DevOps</p>
<p>동기 직접 호출을 검증합니다.</p>	<p>상태 머신을 실행하여 Step Functions와 Systems Manager Automation 간의 동기 직접 호출을 검증합니다.</p> <p>샘플 출력은 추가 정보 섹션을 참조하십시오.</p>	<p>AWS DevOps</p>

관련 리소스

- [AWS Step Functions 시작하기](#) (AWS Step Functions 개발자 안내서)
- [작업 토큰을 통한 콜백 대기](#) (AWS Step Functions 개발자 안내서, 서비스 통합 패턴)
- [send_task_success](#) 및 [send_task_failure](#) API 직접 호출 (Boto3 설명서)
- [AWS Systems Manager Automation](#) (AWS Systems Manager 사용 설명서)

추가 정보

구현 세부 정보

이 패턴은 두 개의 Systems Manager 런북을 배포하는 AWS CloudFormation 템플릿을 제공합니다.

- `SfnRunCommandByInstanceIds` 가 인스턴스 ID를 사용하여 AWS-RunShellScript 명령을 실행합니다.
- `SfnRunCommandByTargets` 이 대상을 사용하여 AWS-RunShellScript 명령을 실행합니다.

각 런북은 Step Functions의 `.waitForTaskToken` 옵션을 사용할 때 동기 직접 호출을 달성하기 위한 세 단계를 구현합니다.

단계	작업	설명
1	RunCommand	RunShellScript 명령을 실행합니다.
2	SendTaskFailure	1단계가 중단되거나 취소될 때 실행합니다. Step Functions send_task_failure API를 직접 호출하며, 이 API는 상태 머신이 전달한 토큰, 실패 오류, 실패 원인에 대한 설명이라는 세 가지 파라미터를 입력으로 받아들입니다.
3	SendTaskSuccess	1단계가 성공하면 실행됩니다. 상태 머신이 전달한 토큰을 입력으로 받아들이는 Step Functions send_task_success API를 직접 호출합니다.

런북 파라미터

SfnRunCommandByInstanceIds 런북

파라미터 이름	Type	선택 또는 필수	설명
<code>execution Timeout</code>	정수	선택 사항	명령이 실패로 간주되기 전에 완료해야

			할 시간(초). 기본값은 3,600(1시간)입니다. 최댓값은 172,800(48시간)입니다.
workingDirectory	String	선택 사항	인스턴스 상의 작업 디렉터리에 대한 경로.
Commands	StringList	필수	실행할 셸 스크립트 또는 명령.
InstanceIds	StringList	필수	명령을 실행하려고 하는 인스턴스의 ID.
taskToken	String	필수	콜백 응답에 사용할 작업 토큰.

SfnRunCommandByTargets 런북

이름	Type	선택 또는 필수	설명
executionTimeout	정수	선택 사항	명령이 실패로 간주되기 전에 완료해야 할 시간(초). 기본값은 3,600(1시간)입니다. 최댓값은 172,800(48시간)입니다.
workingDirectory	String	선택 사항	인스턴스 상의 작업 디렉터리에 대한 경로.
Commands	StringList	필수	실행할 셸 스크립트 또는 명령.
Targets	MapList	필수	지정한 키,값 쌍을 사용하여 인스턴스를 식별하는 검색 기준 배열. 예: [{"Key": "

```
InstanceIds", "Values":
["i-02573cafcf
EXAMPLE",
"i-0471e0
4240EXAMP
LE"]}]
```

taskToken String 필수 콜백 응답에 사용할 작업 토큰.

샘플 출력

다음 테이블에는 step 함수의 샘플 출력이 나와 있습니다. 이는 5단계(TaskSubmitted)와 6단계 (TaskSucceeded) 사이의 총 실행 시간이 100초 이상임을 보여줍니다. 이는 step 함수가 워크플로의 다음 작업으로 이동하기 전에 'sleep 100' 명령이 완료될 때까지 기다렸다는 것을 보여줍니다.

ID	Type	단계	리소스	경과 시간(밀리 초)	타임스탬프
1	Execution Started		-	0	2022년 3월 11일 오후 02:50:34. 303
2	TaskState Entered	StartAutomationWaitForCallback	-	40	2022년 3월 11일 오후 02:50:34. 343
3	TaskScheduled	StartAutomationWaitForCallback	-	40	2022년 3월 11일 오후 02:50:34. 343
4	TaskStarted	StartAutomationWait	-	15.4	2022년 3월 11일 오후 02:50:34. 457

		tForCallB ack			
5	TaskSubmi tted	StartAuto mationWai tForCallB ack	-	657	2022년 3월 11일 오후 02:50:34. 960
6	TaskSucce eded	StartAuto mationWai tForCallB ack	-	103835	2022년 3월 11일 오후 02:52:18. 138
7	TaskState Exited	StartAuto mationWai tForCallB ack	-	103860	2022년 3월 11일 오후 02:52:18. 163
8	Execution Succeeded		-	103897	2022년 3월 11일 오후 02:52:18. 200

AWS Lambda 함수에서 Python을 사용하여 S3 객체의 병렬 읽기 실행

제작: 에두아르도 보르톨루찌

코드 리포지토리: [aws-lambda-parallel-download](#)

환경: PoC 또는 파일럿

기술: 서버리스

AWS 서비스: AWS 람다, 아마존 S3, AWS Step Functions

요약

이 패턴을 사용하여 Amazon Simple Storage Service (Amazon S3) 버킷에서 실시간으로 문서 목록을 검색하고 요약할 수 있습니다. 이 패턴은 Amazon Web Services (AWS) 의 S3 버킷에서 객체를 병렬로 읽기 위한 예제 코드를 제공합니다. 이 패턴은 Python을 사용하여 AWS Lambda 함수로 I/O 바운드 작업을 효율적으로 실행하는 방법을 보여줍니다.

한 금융 회사는 대화형 솔루션에서 이 패턴을 사용하여 상호 관련된 금융 거래를 실시간으로 수동으로 승인하거나 거부했습니다. 금융 거래 문서는 시장과 관련된 S3 버킷에 저장되었습니다. 운영자는 S3 버킷에서 문서 목록을 선택하고 솔루션이 계산한 총 거래 금액을 분석한 후 선택한 일괄 처리를 승인하거나 거부하기로 결정했습니다.

I/O 바인딩된 작업은 다중 스레드를 지원합니다. 이 예제 코드에서는 [컨커런티퓨처스입니다](#).

[ThreadPoolExecutor](#) 최대 1,000개의 동시 스레드와 함께 사용됩니다. Lambda 함수는 최대 1,024개의 스레드를 지원하며, 이 스레드 중 하나가 기본 프로세스입니다. 또한 모든 스레드가 S3 객체 다운로드를 동시에 수행할 수 있도록 최대 풀 연결을 늘려야 합니다.

예제 코드는 S3 버킷에서 JSON 데이터가 포함된 8.3KB 객체 하나를 사용합니다. 객체를 여러 번 읽습니다. Lambda 함수가 객체를 읽은 후 JSON 데이터는 Python 객체로 디코딩됩니다. 이 예제를 실행한 결과 2,048MB의 메모리로 구성된 Lambda 함수를 사용하여 2.3초 만에 1,000건의 읽기가 처리되고 26초 만에 10,000건의 읽기가 처리되었습니다. Lambda 메모리를 늘려도 작업 실행 시간을 줄이는 데 도움이 되지 않았습니다.

[AWS Lambda Power Tuning](#) 도구는 다양한 Lambda 메모리 구성을 테스트하고 작업에 가장 적합한 비율을 확인하는 데 사용되었습니다. performance-to-cost 테스트 결과는 추가 정보 섹션을 참조하십시오.

사전 조건 및 제한 사항

사전 조건

- 활성 상태의 AWS 계정.
- Python 개발에 대한 숙련도

제한 사항

- Lambda 함수는 [최대 1,024개의](#) 실행 프로세스 또는 스레드를 포함할 수 있습니다.
- 새 AWS 계정의 Lambda 메모리 한도는 3,008MB입니다. AWS Lambda 파워 튜닝 도구를 적절히 조정하십시오. 자세한 내용은 [문제 해결](#) 섹션을 참조하십시오.
- Python 버전 3.8은 스레드 [실행 풀에서 스레드 재사용](#)을 도입했기 때문에 최소 권장 버전입니다.
- Amazon S3는 분할된 [접두사당 초당 GET/HEAD 요청 5,500개로](#) 제한됩니다.

제품 버전

- Python 3.8 이상
- AWS 클라우드 개발 키트 (AWS CDK) v2
- AWS Command Line Interface(AWS CLI) 버전 2
- AWS Lambda 파워 튜닝 4.3.3 (선택 사항)

아키텍처

대상 기술 스택

- AWS Lambda
- Amazon S3
- AWS Step Functions (AWS Lambda 파워 튜닝이 배포된 경우)

대상 아키텍처

다음 다이어그램은 S3 버킷에서 객체를 병렬로 읽는 Lambda 함수를 보여줍니다. 다이어그램에는 Lambda 함수 메모리를 미세 조정할 수 있는 AWS Lambda 파워 튜닝 도구를 위한 Step Functions 위

크플로도 포함되어 있습니다. 이러한 미세 조정은 비용과 성능 간에 적절한 균형을 이루는 데 도움이 됩니다.

자동화 및 규모 조정

Lambda 함수는 필요할 때 빠르게 확장됩니다. 수요가 많을 때 Amazon S3에서 발생하는 503 Slow Down 오류를 방지하려면 규모 조정에 몇 가지 제한을 두는 것이 좋습니다.

도구

서비스

- [AWS Cloud Development Kit \(AWS CDK\) v2](#)는 AWS 클라우드 인프라를 코드로 정의하고 프로비저닝하는 데 도움이 되는 소프트웨어 개발 프레임워크입니다. 예제 인프라는 AWS CDK로 배포하기 위해 만들어졌습니다.
- [AWS 명령줄 인터페이스 \(AWS CLI\)](#) 는 명령줄 셸의 명령을 통해 AWS 서비스와 상호 작용하는 데 도움이 되는 오픈 소스 도구입니다. 이 패턴에서는 AWS CLI 버전 2를 사용하여 예제 JSON 파일을 업로드합니다.
- [AWS Lambda](#)는 서버를 프로비저닝하거나 관리할 필요 없이 코드를 실행하는 데 도움이 되는 컴퓨팅 서비스입니다. 필요할 때만 코드를 실행하며 자동으로 확장이 가능하므로 사용한 컴퓨팅 시간만큼만 비용을 지불합니다.
- [Amazon Simple Storage Service\(Amazon S3\)](#)는 원하는 양의 데이터를 저장, 보호 및 검색하는 데 도움이 되는 클라우드 기반 객체 스토리지 서비스입니다.
- [AWS Step Functions](#)는 AWS Lambda 함수와 기타 AWS 서비스를 결합할 수 있는 서버리스 오케스트레이션 서비스로서 비즈니스 크리티컬 애플리케이션을 구축합니다.

기타 도구

- [Python](#)은 범용 컴퓨터 프로그래밍 언어입니다. 유틸 작업자 스레드의 재사용은 Python 버전 3.8에서 도입되었으며 이 패턴의 Lambda 함수 코드가 이 버전을 위해 만들어졌습니다.

코드 리포지토리

이 패턴의 코드는 리포지토리에서 사용할 수 있습니다. [aws-lambda-parallel-download](#) GitHub

모범 사례

- 이 AWS CDK 구조는 AWS 계정의 사용자 권한을 기반으로 인프라를 배포합니다. [AWS CDK Pipelines 또는 계정 간 배포를 사용할 계획이라면 스택 신시사이저를 참조하십시오.](#)
- 이 예제 애플리케이션에는 S3 버킷에서 액세스 로그가 활성화되어 있지 않습니다. 프로덕션 코드에서 액세스 로그를 활성화하는 것이 가장 좋습니다.

에픽

개발 환경 준비

작업	설명	필요한 기술
Python이 설치된 버전을 확인하십시오.	<p>제공된 코드는 Python 3.8 이상에서 작성 및 테스트되었습니다. 설치된 Python 버전을 확인하려면 <code>python3 --help</code>를 실행하십시오. 필요한 경우 새 버전을 다운로드하여 설치하세요.</p> <p>필수 모듈이 설치되었는지 확인하려면 <code>python3 -m pip install --help</code>를 실행하십시오. 모듈이 설치된 경우 오류가 반환되지 않습니다.</p>	클라우드 아키텍트
AWS CDK를 설치하고 구성합니다.	AWS CDK를 설치하고 아직 구성되지 않은 경우 이를 부트스트랩하려면 AWS CDK 시작하기 의 지침을 따르십시오. 설치된 AWS CDK 버전이 2.0 이상인지 확인하려면 다음을 실행하십시오 <code>cdk --version</code> .	클라우드 아키텍트

작업	설명	필요한 기술
	<p>부트스트래핑할 때 파라미터를 로 전달하십시오. <code>--cloudformation-execution-policies "arn:aws:iam::aws:policy/job-function/ViewOnlyAccess"</code> <code>cdk bootstrap</code></p> <p>이 예제에서는 정의된 역할을 사용하여 스택을 배포하지 않으며, 이 매개 변수를 사용하면 배포의 보안을 강화할 수 있습니다.</p>	

예제 리포지토리를 복제하십시오.

작업	설명	필요한 기술
리포지토리를 복제합니다.	<p>최신 버전의 리포지토리를 복제하려면 다음 명령을 실행합니다.</p> <pre>git clone --depth 1 --branch v1.1.2 \ git@github.com:aws-samples/aws-lambda-parallel-download.git</pre>	클라우드 아키텍트
작업 디렉터리를 복제된 리포지토리로 변경합니다.	<p>다음 명령을 실행합니다:</p> <pre>cd aws-lambda-parallel-download</pre>	클라우드 아키텍트
Python 가상 환경을 만듭니다.	<p>Python 가상 환경을 만들려면 다음 명령을 실행합니다.</p>	클라우드 아키텍트

작업	설명	필요한 기술
	<pre>python3 -m venv .venv</pre>	
가상 환경을 활성화합니다.	가상 환경을 활성화하려면 다음 명령을 실행합니다. <pre>source .venv/bin/activate</pre>	클라우드 아키텍트
종속성을 설치합니다.	Python 종속성을 설치하려면 다음 pip 명령을 실행합니다. <pre>pip install -r requirements.txt</pre>	클라우드 아키텍트
코드를 찾아보십시오.	(선택 사항) S3 버킷에서 객체를 다운로드하는 예제 코드에는 <code>resources/parallel.py</code> 있습니다. 인프라 코드는 <code>parallel_download</code> 폴더에 있습니다.	클라우드 아키텍트

앱 배포 및 테스트

작업	설명	필요한 기술
앱을 배포합니다.	<code>cdk deploy</code> 를 실행합니다. AWS CDK 결과를 기록해 두십시오. • <code>ParallelDownloadStack.LambdaFunctionARN</code>	클라우드 아키텍트

작업	설명	필요한 기술
<p>예제 JSON 파일을 업로드하십시오.</p>	<ul style="list-style-type: none"> ParallelDownloadStack.SampleS3BucketName ParallelDownloadStack.StateMachineARN <p>리포지토리에는 약 9KB의 예제 JSON 파일이 들어 있습니다. 생성된 스택의 S3 버킷에 파일을 업로드하려면 다음 명령을 실행합니다.</p> <pre>aws s3 cp sample.json s3://<ParallelDownloadStack.SampleS3BucketName></pre> <p>AWS CDK 출력의 해당 값으로 <ParallelDownloadStack.SampleS3BucketName> 대체합니다.</p>	클라우드 아키텍트

작업	설명	필요한 기술
<p>앱을 실행합니다.</p>	<ol style="list-style-type: none"> 1. AWS 관리 콘솔에 로그인하고 Lambda 콘솔로 이동하여 AWS CDK 출력의 ARN이 있는 Lambda 함수를 찾습니다. <code>ParallelDownloadStack.LambdaFunctionARN</code> 2. 테스트 탭에서 이벤트 JSON을 다음과 같이 변경합니다. <div style="border: 1px solid #ccc; border-radius: 10px; padding: 10px; margin: 10px 0;"> <pre>{"objectKey": "sample.json"}</pre> </div> 3. 테스트를 선택합니다. 4. 결과를 보려면 세부 정보를 선택합니다. 세부 정보에는 Parallel 다운로드 통계, 실행 정보 및 로그가 표시됩니다. 	<p>클라우드 아키텍트</p>
<p>다운로드 횟수를 추가합니다.</p>	<p>(선택 사항) 1,500개의 get 객체 호출을 실행하려면 파라미터의 Event JSON에서 다음 JSON을 사용하십시오. Test</p> <div style="border: 1px solid #ccc; border-radius: 10px; padding: 10px; margin: 10px 0;"> <pre>{"repeat": 1500, "objectKey": "sample.json"}</pre> </div>	<p>클라우드 아키텍트</p>

선택 사항: AWS Lambda 파워 튜닝 실행

작업	설명	필요한 기술
<p>AWS Lambda 파워 튜닝 도구를 실행합니다.</p>	<ol style="list-style-type: none"> 1. 콘솔에 로그인하고 Step Functions로 이동합니다. 	<p>클라우드 아키텍트</p>

작업	설명	필요한 기술
	<p>2. AWS CDK 출력의 ARN으로 스테이트 머신을 찾습니다. ParallelDownloadStack.StateMachineARN</p> <p>3. [실행 시작] 을 선택하고 다음 JSON을 붙여넣습니다.</p> <pre data-bbox="630 579 1029 1058"> { "lambdaARN": "<ParallelDownloadStack.LambdaFunctionARN>", "num": 5, "payload": {"repeat": 2000, "objectKey": "sample.json"} } </pre> <p>CDK 출력의 <ParallelDownloadStack.LambdaFunctionARN> 값으로 바뀌어야 한다는 점을 잊지 마세요.</p> <p>실행이 끝나면 실행 입력 및 출력 탭에 결과가 표시됩니다.</p>	
<p>AWS Lambda 파워 튜닝 결과를 그래프로 확인하십시오.</p>	<p>실행 입력 및 출력 탭에서 visualization 속성 링크를 복사하여 새 브라우저 탭에 붙여넣습니다.</p>	<p>클라우드 아키텍트</p>

정리

작업	설명	필요한 기술
<p>S3 버킷에서 객체를 제거합니다.</p>	<p>배포된 리소스를 삭제하기 전에 S3 버킷에서 모든 객체를 제거합니다.</p> <pre data-bbox="594 485 1026 724">aws s3 rm s3://<ParallelDownloadStack.SampleS3BucketName> \ --recursive</pre> <p>AWS CDK 출력의 값으로 <ParallelDownloadStack.SampleS3BucketName> 대체해야 합니다.</p>	<p>클라우드 아키텍트</p>
<p>리소스를 삭제하십시오.</p>	<p>이 파일럿을 위해 만든 모든 리소스를 삭제하려면 다음 명령을 실행합니다.</p> <pre data-bbox="594 1157 1026 1234">cdk destroy</pre>	<p>클라우드 아키텍트</p>

문제 해결

문제	Solution
<p>'MemorySize' value failed to satisfy constraint: Member must have value less than or equal to 3008</p>	<p>새 계정의 경우 Lambda 함수에서 3,008MB 이상을 구성하지 못할 수 있습니다. AWS Lambda 파워 튜닝을 사용하여 테스트하려면 Step Functions 실행을 시작할 때 입력 JSON에 다음 속성을 추가하십시오.</p> <pre data-bbox="834 1787 1507 1881">"powerValues": [512,</pre>

문제	Solution
	<pre> 1024, 1536, 2048, 2560, 3008] </pre>

관련 리소스

- [Python — 컨커런트.퓨처스. ThreadPoolExecutor](#)
- [Lambda 할당량 — 함수 구성, 배포 및 실행](#)
- [Python에서 AWS CDK로 작업하기](#)
- [AWS Lambda 파워 튜닝을 사용한 프로파일링 함수](#)

추가 정보

코드

다음 코드 스니펫은 병렬 I/O 처리를 수행합니다.

```

with ThreadPoolExecutor(max_workers=MAX_WORKERS) as executor:
    for result in executor.map(a_function, (the_arguments)):
        ...

```

는 스레드를 사용할 수 있게 되면 해당 스레드를 ThreadPoolExecutor 재사용합니다.

테스트 및 결과

첫 번째 테스트에서는 2,500개의 객체 읽기를 처리했으며 결과는 다음과 같습니다.

3,009MB부터 시작하여 메모리가 증가해도 처리 시간 수준은 동일하게 유지되었지만 메모리 크기가 커질수록 비용이 증가했습니다.

또 다른 테스트에서는 256MB의 배수인 값을 사용하고 10,000개의 객체 읽기를 처리하여 1,536MB에서 3,072MB 사이의 메모리 범위를 조사했으며 결과는 다음과 같습니다.

가장 좋은 performance-to-cost 비율은 2,048MB 메모리 Lambda 구성이었습니다.

비교하자면, 2,500개의 객체 읽기를 순차적으로 처리하는 데 40초가 걸렸습니다. 2,048MB Lambda 구성을 사용한 병렬 프로세스에는 5.8초가 걸렸으며, 이는 85% 단축된 수치입니다.

VPC 엔드포인트를 통해 Amazon S3 버킷에 대한 프라이빗 액세스를 설정합니다.

작성자: 마틴 마리치 (AWS), 가브리엘 로드리게스 가르시아 (AWS), 슈크라트 코자예프 (AWS), 니콜라스 제이콥 베어 (AWS), 모한 고우다 푸루쇼타마 (AWS), 호아킨 리나우도 (AWS)

코드 리포지토리: [프라이빗 S3 VPC](#)

환경: 프로덕션

기술: 서버리스

AWS 서비스: 아마존 API 게이트웨이, 아마존 S3, 아마존 VPC, 엘라스틱 로드 밸런싱 (ELB)

요약

Amazon Simple Storage Service (Amazon S3)에서는 미리 서명된 URL을 사용하여 임의 크기의 파일을 대상 사용자와 공유할 수 있습니다. 기본적으로 Amazon S3의 사전 서명된 URL은 만료 기간 내에 인터넷에서 액세스할 수 있으므로 편리하게 사용할 수 있습니다. 하지만 기업 환경에서는 Amazon S3의 사전 서명된 URL에 대한 액세스를 프라이빗 네트워크로만 제한해야 하는 경우가 많습니다.

이 패턴은 인터넷 통과 없이 사설 네트워크에서 미리 서명된 URL을 사용하여 S3 객체와 안전하게 상호 작용할 수 있는 서버리스 솔루션을 제공합니다. 아키텍처에서 사용자는 내부 도메인 이름을 통해 Application Load Balancer에 액세스합니다. 트래픽은 Amazon API Gateway와 S3 버킷의 가상 사설 클라우드 (VPC) 엔드포인트를 통해 내부적으로 라우팅됩니다. 이 AWS Lambda 함수는 프라이빗 VPC 엔드포인트를 통해 파일 다운로드를 위한 미리 서명된 URL을 생성하므로 민감한 데이터의 보안 및 개인 정보 보호를 강화하는 데 도움이 됩니다.

사전 조건 및 제한 사항

사전 조건

- 기업 네트워크에 연결된 (예: 를 통해) 에 배포된 서브넷을 포함하는 VPC입니다. AWS 계정 AWS Direct Connect

제한 사항

- S3 버킷은 도메인과 이름이 같아야 하므로 [Amazon S3 버킷 이름 지정 규칙](#)을 확인하는 것이 좋습니다.
- 이 샘플 아키텍처에는 배포된 인프라에 대한 모니터링 기능이 포함되어 있지 않습니다. 사용 사례에 모니터링이 필요한 경우 [AWS 모니터링 서비스](#)를 추가해 보세요.
- 이 샘플 아키텍처에는 입력 검증이 포함되지 않습니다. 사용 사례에 입력 검증과 보안 수준 강화가 필요한 경우 [API 보호를 AWS WAF 위해 사용을 고려해 보세요](#).
- 이 샘플 아키텍처에는 Application Load Balancer를 사용한 액세스 로깅이 포함되어 있지 않습니다. 사용 사례에 액세스 로깅이 필요한 경우 [로드 밸런서 액세스](#) 로그를 활성화해 보세요.

버전

- 파이썬 버전 3.11 이상
- 테라폼 버전 1.6 이상

아키텍처

대상 기술 스택

대상 기술 스택에는 다음과 같은 AWS 서비스가 사용됩니다.

- Amazon S3는 파일을 안전하게 업로드, 다운로드 및 저장하는 데 사용되는 핵심 스토리지 서비스입니다.
- Amazon API Gateway는 S3 버킷과 상호 작용하기 위한 리소스와 엔드포인트를 노출합니다. 이 서비스는 데이터 다운로드 또는 업로드를 위해 미리 서명된 URL을 생성하는 역할을 합니다.
- AWS Lambda Amazon S3에서 파일을 다운로드하기 위해 미리 서명된 URL을 생성합니다. Lambda 함수는 API Gateway에 의해 호출됩니다.
- Amazon VPC는 VPC 내에 리소스를 배포하여 네트워크 격리를 제공합니다. VPC에는 트래픽 흐름을 제어하는 서브넷과 라우팅 테이블이 포함되어 있습니다.
- 애플리케이션 로드 밸런서는 들어오는 트래픽을 API Gateway 또는 S3 버킷의 VPC 엔드포인트로 라우팅합니다. 이를 통해 기업 네트워크의 사용자가 내부적으로 리소스에 액세스할 수 있습니다.
- Amazon S3용 VPC 엔드포인트를 사용하면 퍼블릭 인터넷을 거치지 않고도 VPC와 Amazon S3의 리소스 간에 직접 비공개로 통신할 수 있습니다.
- AWS Identity and Access Management (IAM) 은 리소스에 대한 액세스를 제어합니다. AWS API 및 기타 서비스와의 안전한 상호 작용을 보장하기 위해 권한이 설정됩니다.

대상 아키텍처

다이어그램은 다음을 보여 줍니다.

1. 기업 네트워크의 사용자는 내부 도메인 이름을 통해 Application Load Balancer에 액세스할 수 있습니다. 예를 들어 연결을 통해 회사 네트워크와 인트라넷 서브넷 사이에 연결이 있다고 가정합니다. AWS 계정 AWS Direct Connect
2. Application Load Balancer는 들어오는 트래픽을 API Gateway로 라우팅하여 데이터를 Amazon S3에 다운로드하거나 업로드하기 위한 사전 서명된 URL을 생성하거나 S3 버킷의 VPC 엔드포인트로 라우팅합니다. 두 시나리오 모두 요청이 내부적으로 라우팅되므로 인터넷을 통과할 필요가 없습니다.
3. API Gateway는 S3 버킷과 상호 작용할 수 있는 리소스와 엔드포인트를 노출합니다. 이 예시에서는 S3 버킷에서 파일을 다운로드할 수 있는 엔드포인트를 제공하지만, 이를 확장하여 업로드 기능도 제공할 수 있습니다.
4. Lambda 함수는 퍼블릭 Amazon S3 도메인 대신 Application Load Balancer의 도메인 이름을 사용하여 Amazon S3에서 파일을 다운로드하기 위해 미리 서명된 URL을 생성합니다.
5. 사용자는 미리 서명된 URL을 수신하고 이 URL을 사용하여 Application Load Balancer를 사용하여 Amazon S3에서 파일을 다운로드합니다. 로드 밸런서에는 API에 적합하지 않은 트래픽을 S3 버킷의 VPC 엔드포인트로 보내는 기본 경로가 포함되어 있습니다.
6. VPC 엔드포인트는 사용자 지정 도메인 이름이 포함된 사전 서명된 URL을 S3 버킷으로 라우팅합니다. S3 버킷의 이름은 도메인과 같아야 합니다.

자동화 및 규모 조정

이 패턴은 Terraform을 사용하여 코드 리포지토리의 인프라를 a로 배포합니다. AWS 계정

도구

도구

- [Python](#)은 범용 컴퓨터 프로그래밍 언어입니다.
- [Terraform](#)은 클라우드 및 온프레미스 리소스를 생성하고 관리하는 데 도움이 HashiCorp 되는 코드 형 인프라 (IaC) 도구입니다.
- [AWS Command Line Interface \(AWS CLI\)](#) 는 명령줄 셸의 명령을 통해 AWS 서비스와 상호 작용하는 데 도움이 되는 오픈 소스 도구입니다.

코드 리포지토리

이 패턴의 코드는 <https://github.com/aws-samples/private-s3-vpce> GitHub 저장소에서 사용할 수 있습니다.

모범 사례

이 패턴의 샘플 아키텍처는 [IAM 권한을](#) 사용하여 API에 대한 액세스를 제어합니다. 유효한 IAM 자격 증명이 있는 사람은 누구나 API를 호출할 수 있습니다. 사용 사례에 더 복잡한 권한 부여 모델이 필요한 경우 [다른 액세스 제어 메커니즘을 사용하는 것이 좋습니다.](#)

에픽

솔루션을 다음 위치에 배포하십시오. AWS 계정

작업	설명	필요한 기술
AWS 자격 증명 취득.	AWS 자격 증명과 계정 액세스 권한을 검토하십시오. 지침은 설명서의 구성 및 자격 증명 파일 설정을 AWS CLI 참조하십시오.	AWS DevOps, 일반 AWS
리포지토리를 복제합니다.	다음 패턴으로 제공된 GitHub 리포지토리를 복제하십시오. <pre>git clone https://github.com/aws-samples/private-s3-vpce</pre>	AWS DevOps, 일반 AWS
변수를 구성합니다.	1. 컴퓨터의 GitHub 리포지토리에서 terraform 폴더를 엽니다. <pre>cd terraform</pre> 2. <code>example.tfvars</code> 파일을 열고 필요에 따라 매개 변수를 사용자 지정합니다.	AWS DevOps, 일반 AWS

작업	설명	필요한 기술
솔루션 배포.	<ol style="list-style-type: none"> terraform 폴더에서 Terraform을 실행하고 사용자 지정한 변수를 전달합니다. <pre>terraform apply -var-file="example.tfvars"</pre> <ol style="list-style-type: none"> 아키텍처 다이어그램에 표시된 리소스가 성공적으로 배포되었는지 확인합니다. 	AWS DevOps, 일반 AWS

솔루션 테스트

작업	설명	필요한 기술
테스트 파일 생성.	<p>Amazon S3에 파일을 업로드 하여 파일 다운로드를 위한 테스트 시나리오를 생성합니다.</p> <p>Amazon S3 콘솔 또는 다음 AWS CLI 명령을 사용할 수 있습니다.</p> <pre>aws s3 cp /path/to/testfile s3://your-bucket-name/testfile</pre>	AWS DevOps, 일반 AWS
사전 서명된 URL 기능을 테스트하십시오.	<ol style="list-style-type: none"> awscurl을 사용하여 Application Load Balancer에 요청을 보내 테스트 파일의 미리 서명된 URL을 생성하십시오. 	AWS DevOps, 일반 AWS

작업	설명	필요한 기술
	<pre>awsurl https://your-domain-name/api/get_url?key=testfile</pre> <p>이 단계는 자격 증명으로부터 유효한 서명을 생성하며, API Gateway에서 이를 검증합니다.</p> <p>2. 이전 단계에서 받은 응답에서 링크를 구문 분석하고 미리 서명된 URL을 열어 파일을 다운로드합니다.</p>	
정리	<p>리소스가 더 이상 필요하지 않을 때는 해당 리소스를 제거해야 합니다.</p> <pre>terraform destroy</pre>	AWS DevOps, 일반 AWS

문제 해결

문제	Solution
<p>숫자 기호 (#) 와 같은 특수 문자가 포함된 S3 객체 키 이름이 URL 파라미터를 깨뜨리고 오류가 발생합니다.</p>	<p>URL 파라미터를 올바르게 인코딩하고, S3 객체 키 이름이 Amazon S3 지침을 따르는지 확인하십시오.</p>

관련 리소스

아마존 S3:

- [미리 서명된 URL로 객체 공유](#)
- [버킷 정책으로 VPC 엔드포인트의 액세스 제어](#)

아마존 API 게이트웨이:

- [API Gateway의 프라이빗 API에 VPC 엔드포인트 정책 사용](#)

애플리케이션 로드 밸런서:

- [ALB, S3 및 PrivateLink \(AWS 블로그 게시물\) 를 사용하여 내부 HTTPS 정적 웹 사이트 호스팅](#)

서버리스 접근 방식을 사용하여 AWS 서비스를 함께 연결

작성자: Aniket Braganza(AWS)

환경: 프로덕션

기술: 서버리스, 클라우드 네이티브, 소프트웨어 개발 및 테스트, 현대화, 인프라 DevOps

AWS 서비스: Amazon S3, Amazon SNS, Amazon SQS, AWS Lambda

요약

이 패턴은 Amazon Simple Storage Service(S3), Amazon Simple Notification Service(SNS), Amazon Simple Queue Service(Amazon SQS), AWS Lambda를 함께 연결하여 업로드된 파일을 처리하는 확장 가능한 서버리스 접근 방식을 설명합니다. 업로드된 파일 예제는 데모용입니다. 서버리스 접근 방식을 사용하면 비즈니스 목표를 달성하는 데 필요한 AWS 서비스를 결합하여 다른 작업을 완료할 수 있습니다. 서버리스 접근 방식은 이벤트 기반 알림, 복원력이 뛰어난 스토리지, 서비스형 기능(FaaS) 컴퓨팅 등을 기반으로 요청을 처리하는 비동기 워크플로를 사용합니다. 서버리스 접근 방식을 사용하면 비용을 최소화하면서 수요에 맞게 확장할 수 있습니다.

참고: 서버리스 접근 방식을 통해 AWS 서비스를 결합하는 몇 가지 옵션이 있습니다. 예를 들어 Amazon SNS와 Amazon SQS 대신 Lambda와 Amazon S3를 결합하는 접근 방식을 사용할 수 있습니다. 하지만 이 패턴은 Amazon SNS와 Amazon SQS를 사용합니다. 이 접근 방식을 사용하면 이벤트 알림 중에 Lambda 호출 프로세스에 여러 통합 지점을 추가하고 처리 오버헤드를 최소화하면서 서버리스 오케스트레이션에 여러 리스너를 포함하도록 구현을 확장할 수 있기 때문입니다.

사전 조건 및 제한 사항

사전 조건

- 활성 상태의 AWS 계정
- AWS 계정에 대한 프로그래밍 방식 액세스 자세한 내용은 다음을 참조하십시오.
 - AWS Cloud Development Kit(AWS CDK) 설명서의 [사전 조건](#)
 - AWS Command Line Interface(AWS CLI) 설명서의 [사전 조건](#)
- AWS CDK, [설치됨](#)
- AWS CLI, [설치](#) 및 [구성됨](#)
- [Python 3.9](#)

제품 버전

- AWS CDK 2.x
- Python 3.9

아키텍처

다음 다이어그램은 사용자가 결합된 AWS 서비스로 파일을 S3 버킷에 업로드하여 처리하는 방법을 보여줍니다.

이 다이어그램은 다음 워크플로를 보여줍니다.

1. 사용자가 S3 버킷에 파일을 업로드합니다.
2. 해당 업로드는 SNS 주제에 메시지를 게시하는 S3 이벤트를 시작합니다. 이 메시지는 S3 이벤트의 세부 정보를 포함합니다.
3. SNS 주제에 게시된 메시지는 SQS 대기열에 삽입되며, SQS 대기열은 해당 주제에 대한 알림을 구독하고 수신합니다.
4. Lambda 함수는 SQS 대기열(이벤트 소스)을 폴링하고 메시지가 처리되기를 기다립니다.
5. Lambda 함수가 SQS 대기열에서 메시지를 수신하면 메시지를 처리하고 해당 메시지의 수신을 확인합니다.
6. Lambda에서 메시지를 처리하지 않는 경우 해당 메시지는 SQS 대기열로 반환되고 최종적으로 [SQS DLQ\(Dead Letter Queue\)](#)로 전송됩니다.

기술 스택

- Amazon S3
- Amazon SNS
- Amazon SQS
- AWS Lambda

도구

AWS 서비스

- [Amazon Simple Storage Service\(S3\)](#)는 원하는 양의 데이터를 저장, 보호 및 검색하는 데 도움이 되는 클라우드 기반 객체 스토리지 서비스입니다.
- [Amazon Simple Notification Service\(Amazon SNS\)](#)를 사용하면 웹 서버 및 이메일 주소를 포함하여 게시자와 클라이언트 간의 메시지 교환을 조정하고 관리할 수 있습니다.
- [Amazon Simple Queue Service\(Amazon SQS\)](#)는 내구력 있고 가용성이 뛰어난 보안 호스팅 대기열을 제공하며 이를 통해 분산 소프트웨어 시스템과 구성 요소를 통합 및 분리할 수 있습니다.
- [AWS Lambda](#)는 서버를 프로비저닝하거나 관리할 필요 없이 코드를 실행하는 데 도움이 되는 컴퓨팅 서비스입니다. 필요할 때만 코드를 실행하며 자동으로 확장이 가능하므로 사용한 컴퓨팅 시간만큼만 비용을 지불합니다.

기타 도구

- [AWS Cloud Development Kit\(AWS CDK\)](#)는 AWS CDK 앱과 상호 작용하기 위한 기본 도구입니다. 앱을 실행하고, 정의한 애플리케이션 모델을 조사하고, AWS CDK에서 생성된 CloudFormation AWS 템플릿을 생성 및 배포합니다.
- [AWS Command Line Interface\(AWS CLI\)](#)는 명령줄 셸에서 명령을 사용하여 AWS 서비스와 상호 작용할 수 있는 오픈 소스 도구입니다.
- [Python](#)은 높은 수준의 해석된 범용 프로그래밍 언어입니다.

코드

이 패턴의 코드는 S3에서 [SNS로, SQS, Lambda 리포지토리로 연결되는 GitHub 체인에서](#) 사용할 수 있습니다.

에픽

서버리스 환경을 개발하세요.

작업	설명	필요한 기술
리포지토리를 복제합니다.	리포지토리 를 복제하고 해당 python/s3-sns-sqs-lambda-chain 폴더로 이동합니다.	앱 개발자

작업	설명	필요한 기술
가상 환경을 설정합니다.	<ol style="list-style-type: none"> 1. AWS CDK에서 <code>python3 -m venv .venv</code> 명령을 실행합니다. 2. MacOS/Linux에서 <code>source .venv/bin/activate</code> 명령을 실행하거나 Windows에서 <code>.venv\Scripts\activate.bat</code> 명령을 실행합니다. 	앱 개발자
종속 항목 설치	<code>pip install -r requirements.txt</code> 명령을 실행합니다.	앱 개발자

스택 테스트 CloudFormation

작업	설명	필요한 기술
유닛 테스트를 실행합니다.	<ol style="list-style-type: none"> 1. <code>pip install -r requirements-dev.txt</code> 명령을 실행합니다. 2. (선택 사항) <code>cdk synth --no-staging > template.yml</code> 명령을 실행하여 CloudFormation 스택을 생성합니다. 중요: 스택을 검사할 수 있지만 스테이징된 리소스 및 아티팩트는 생성하지 마세요. 3. <code>pytest</code> 명령을 실행하여 모든 유닛 테스트를 실행합니다. 	앱 개발자, 테스트 엔지니어

작업	설명	필요한 기술
	4. (선택 사항) <code>pytest tests/unit/<test_filename></code> 명령을 실행하여 특정 파일에 대한 테스트를 실행합니다.	

CloudFormation 스택 배포

작업	설명	필요한 기술
부트스트랩 환경을 설정합니다.	AWS 설명서의 부트스트래핑 지침에 따라 CloudFormation 스택이 배포될 각 AWS 지역에서 AWS CDK 배포 환경을 부트스트랩하십시오. 참고: 이 단계를 수행하려면 프로그래밍 방식으로 액세스할 수 있는 보안 인증 정보가 필요합니다.	앱 개발자, 엔지니어, 데이터 엔지니어 DevOps
CloudFormation 스택을 배포하세요.	<code>cdk deploy</code> 명령을 실행하여 스택을 빌드하고 AWS 계정에 배포합니다.	앱 개발자, DevOps 엔지니어, AWS DevOps

환경 리소스 정리

작업	설명	필요한 기술
CloudFormation 스택을 삭제하고 관련 리소스를 제거합니다.	생성된 CloudFormation 스택을 삭제하고 관련 리소스를 모두 제거하려면 <code>run cdk destroy</code> 명령을 실행합니다.	앱 개발자

패턴 더 보기

- [Athena를 사용한 Amazon DynamoDB 테이블 액세스, 쿼리 및 조인](#)
- [Athena의 ML 예측을 위한 Amazon DynamoDB의 데이터 집계](#)
- [AWS 리소스 평가 자동화](#)
- [aws-nyk를 사용하여 AWS 리소스 자동 삭제](#)
- [AWS SAM을 사용하여 중첩된 애플리케이션 자동 배포](#)
- [AWS 계정 전반에 걸쳐 Amazon RDS 인스턴스의 복제를 자동화](#)
- [DynamoDB TTL을 사용하여 Amazon S3에 항목 자동으로 보관](#)
- [변경 사항을 자동으로 감지하고 단일 저장소에 대한 다양한 CodePipeline 파이프라인을 시작합니다. CodeCommit](#)
- [DevOps 프랙티스와 AWS Cloud9를 사용하여 마이크로서비스로 느슨하게 연결된 아키텍처를 구축 하십시오.](#)
- [Amazon Service에서 멀티테넌트 서버리스 아키텍처 구축 OpenSearch](#)
- [AWS 클라우드에서 고급 메인프레임 파일 뷰어 구축](#)
- [AWS 서비스를 사용하여 위험 가치\(VaR\) 계산](#)
- [여러 AWS 계정 및 AWS 리전에 걸쳐 AWS Service Catalog 제품 복사](#)
- [AWS CLI를 사용하여 S3 버킷에서 다른 계정 및 리전으로 데이터 복사](#)
- [Java 및 Python 프로젝트를 위한 동적 CI 파이프라인을 자동으로 생성](#)
- [CQRS 및 이벤트 소싱을 사용하여 모놀리식 유형을 마이크로서비스로 분해하기](#)
- [Amazon S3에 React 기반 단일 페이지 애플리케이션을 배포하고 CloudFront](#)
- [프라이빗 엔드포인트와 Application Load Balancer 사용하여 내부 웹 사이트에 Amazon API Gateway API 배포](#)
- [Amazon EKS 클러스터의 배포 및 디버깅](#)
- [AWS 클라우드에서 인프라를 코드로 사용하여 서버리스 데이터 레이크 배포와 관리](#)
- [컨테이너 이미지로 Lambda 함수 배포](#)
- [Amazon Bedrock 에이전트 및 지식 베이스를 사용하여 완전 자동화된 채팅 기반 도우미를 개발하십시오.](#)
- [RAG 및 프롬프트를 사용하여 고급 제너레이티브 AI 채팅 기반 어시스턴트 개발 ReAct](#)
- [Step Functions를 사용하여 IAM Access Analyzer로 IAM 정책을 동적으로 생성하기](#)
- [시작 시 Amazon S3에 Amazon EMR 로깅이 활성화되었는지 확인](#)

- [온디맨드 용량에 대한 DynamoDB 테이블의 비용 추정](#)
- [Amazon Personalize를 사용하여 개인화되고 순위가 다시 매겨진 추천 생성](#)
- [AWS Glue 작업과 Python을 사용하여 테스트 데이터 생성](#)
- [AWS Step Functions을 사용하여 서버리스 사가 패턴 구현](#)
- [AWS CDK를 사용하여 여러 AWS 지역, 계정 및 DevOps OU에서 Amazon Guru를 활성화하여 운영 성능을 개선하십시오.](#)
- [Step Functions와 Lambda 프록시 함수를 사용하여 여러 AWS 계정에서 CodeBuild 프로젝트를 시작합니다.](#)
- [AWS Glue를 사용하여 아파치 카산드라 워크로드를 Amazon Keyspaces로 마이그레이션하십시오.](#)
- [여러 AWS 계정에 공유된 Amazon Machine Image의 사용을 모니터링](#)
- [AWS Step Functions를 사용하여 검증, 변환 및 파티셔닝을 통해 ETL 파이프라인 오케스트레이션](#)
- [AWS Fargate를 사용하여 이벤트 기반 및 예약된 워크로드를 대규모로 실행](#)
- [Amazon을 사용하여 VPC를 통해 Amazon S3 버킷의 정적 콘텐츠를 제공합니다. CloudFront](#)
- [AWS Lambda를 사용하여 육각형 아키텍처로 Python 프로젝트 구조화](#)
- [다중 계정 환경에서 모든 Security Hub 멤버 계정의 보안 표준 제어를 끕니다.](#)

소프트웨어 개발 및 테스트

주제

- [Python 애플리케이션을 사용하여 Amazon DynamoDB용 PynamoDB 모델 및 CRUD 함수를 자동으로 생성합니다.](#)
- [Green Boost를 통한 풀스택 클라우드 네이티브 웹 애플리케이션 개발 살펴보기](#)
- [AWS를 GitHub 사용하여 Node.js 애플리케이션에 대한 단위 테스트 실행 CodeBuild](#)
- [AWS Lambda를 사용하여 육각형 아키텍처로 Python 프로젝트 구조화](#)
- [패턴 더 보기](#)

Python 애플리케이션을 사용하여 Amazon DynamoDB용 PynamoDB 모델 및 CRUD 함수를 자동으로 생성합니다.

작성자: 비지트 바시슈타 (AWS), 데라즈 알림찬다니 (AWS), 다난제이 카란카르 (AWS)

코드 리포지토리: amazon-reverse-engineer-dynamodb	환경: PoC 또는 파일럿	기술: 소프트웨어 개발 및 테스트, 데이터베이스, DevOps
워크로드: 오픈 소스	AWS 서비스: Amazon DynamoDB	

요약

Amazon DynamoDB 데이터베이스 작업을 효율적으로 수행하려면 일반적으로 엔티티와 생성, 읽기, 업데이트 및 삭제 (CRUD) 작업 함수가 필요합니다. PynamoDB는 파이썬 3을 지원하는 파이썬 기반 인터페이스입니다. 또한 Amazon DynamoDB 트랜잭션 지원, 자동 속성 값 직렬화 및 역직렬화, Flask 및 Django와 같은 일반적인 Python 프레임워크와의 호환성 등의 기능을 제공합니다. 이 패턴은 PynamoDB 모델 및 CRUD 작업 함수의 자동 생성을 간소화하는 라이브러리를 제공하여 개발자가 Python과 DynamoDB를 사용하는 데 도움이 됩니다. 데이터베이스 테이블을 위한 필수 CRUD 함수를 생성하는 동시에 Amazon DynamoDB 테이블에서 PynamoDB 모델 및 CRUD 함수를 리버스 엔지니어링할 수도 있습니다. 이 패턴은 Python 기반 응용 프로그램을 사용하여 데이터베이스 작업을 단순화하도록 설계되었습니다.

이 솔루션의 주요 기능은 다음과 같습니다.

- JSON 스키마를 PynamoDB 모델로 변환 — JSON 스키마 파일을 가져와서 Python에서 PynamoDB 모델을 자동으로 생성합니다.
- CRUD 함수 생성 — DynamoDB 테이블에서 CRUD 작업을 수행하는 함수를 자동으로 생성합니다.
- DynamoDB에서의 리버스 엔지니어링 — PynamoDB 객체 관계형 매핑 (ORM) 을 사용하여 기존 Amazon DynamoDB 테이블의 PynamoDB 모델 및 CRUD 함수를 리버스 엔지니어링할 수 있습니다.

사전 조건 및 제한 사항

사전 조건

- 활성 상태의 AWS 계정
- Python 버전 3.8 이상, [다운로드](#) 및 설치
- [Jinja2 버전 3.1.2 이상, 다운로드 및 설치](#)
- ORM을 생성하려는 Amazon DynamoDB 테이블
- AWS Command Line Interface(AWS CLI), [설치](#) 및 [구성됨](#)
- [PynamoDB 버전 5.4.1 이상, 설치됨](#)

아키텍처

대상 기술 스택

- JSON 스크립트
- Python 애플리케이션
- PynamoDB 모델
- Amazon DynamoDB 데이터베이스 인스턴스

대상 아키텍처

1. 입력 JSON 스키마 파일을 생성합니다. 이 JSON 스키마 파일은 PynamoDB 모델 및 CRUD 함수를 생성하려는 각 DynamoDB 테이블의 속성을 나타냅니다. 여기에는 다음과 같은 세 가지 중요한 키가 포함되어 있습니다.
 - `name`—대상 DynamoDB 테이블의 이름.
 - `region`— 테이블이 호스팅되는 AWS 지역
 - `attributes`— [파티션 키 \(해시 속성이라고도 함\), 정렬 키, 로컬 보조 인덱스, 글로벌 보조 인덱스 및 키가 아닌 모든 속성 등 대상 테이블에 속하는 속성](#). 이 도구는 애플리케이션이 대상 테이블에서 직접 키 속성을 가져올 때 입력 스키마가 키가 아닌 속성만 제공할 것으로 예상합니다. JSON 스키마 파일에서 속성을 지정하는 방법의 예는 이 패턴의 [추가 정보](#) 섹션을 참조하십시오.
2. Python 애플리케이션을 실행하고 JSON 스키마 파일을 입력으로 제공합니다.
3. Python 응용 프로그램은 JSON 스키마 파일을 읽습니다.
4. Python 애플리케이션은 DynamoDB 테이블에 연결하여 스키마와 데이터 형식을 도출합니다. 애플리케이션은 [describe_table](#) 작업을 실행하고 [테이블의](#) 키 및 인덱스 속성을 가져옵니다.

5. Python 애플리케이션은 JSON 스키마 파일과 DynamoDB 테이블의 속성을 결합합니다. Jinja 템플릿 엔진을 사용하여 PynamoDB 모델 및 해당 CRUD 함수를 생성합니다.
6. PynamoDB 모델에 액세스하여 DynamoDB 테이블에서 CRUD 작업을 수행할 수 있습니다.

도구

서비스

- [Amazon DynamoDB](#)는 빠르고 예측 가능하고 확장 가능한 성능을 제공하는 완전 관리형 NoSQL 데이터베이스 서비스입니다.

기타 도구

- [Jinja](#)는 템플릿을 최적화된 Python 코드로 컴파일하는 확장 가능한 템플릿 엔진입니다. 이 패턴은 Jinja를 사용하여 템플릿 내에 플레이스홀더와 로직을 삽입하여 동적 콘텐츠를 생성합니다.
- [PynamoDB](#)는 [아마존 DynamoDB](#)를 위한 파이썬 기반 인터페이스입니다.
- [Python](#)은 범용 컴퓨터 프로그래밍 언어입니다.

코드 리포지토리

이 패턴의 코드는 GitHub [자동 생성 PynamoDB](#) 모델 및 CRUD 함수 저장소에서 사용할 수 있습니다. 리포지토리는 컨트롤러 패키지와 템플릿이라는 두 개의 주요 부분으로 나뉩니다.

컨트롤러 패키지

컨트롤러 Python 패키지에는 PynamoDB 모델 및 CRUD 함수를 생성하는 데 도움이 되는 기본 애플리케이션 로직이 포함되어 있습니다. 이는 다음을 포함합니다.

- `input_json_validator.py`— 이 Python 스크립트는 입력 JSON 스키마 파일의 유효성을 검사하고 대상 DynamoDB 테이블 목록과 각 테이블에 필요한 속성을 포함하는 Python 객체를 생성합니다.
- `dynamo_connection.py`— 이 스크립트는 DynamoDB 테이블에 대한 연결을 설정하고 작업을 사용하여 `describe_table` PynamoDB 모델을 생성하는 데 필요한 속성을 추출합니다.
- `generate_model.py`— 이 스크립트에는 입력 JSON 스키마 파일 및 작업을 기반으로 PynamoDB 모델을 만드는 Python 클래스가 `GenerateModel` 포함되어 있습니다. `describe_table`
- `generate_crud.py`— JSON 스키마 파일에 정의된 DynamoDB 테이블의 경우 이 스크립트는 작업을 사용하여 Python `GenerateCrud` 클래스를 생성합니다.

템플릿

이 Python 디렉토리에는 다음과 같은 Jinja 템플릿이 포함되어 있습니다.

- `model.jinja`— 이 Jinja 템플릿에는 PynamoDB 모델 스크립트를 생성하기 위한 템플릿 표현식이 포함되어 있습니다.
- `crud.jinja`— 이 Jinja 템플릿에는 CRUD 함수 스크립트를 생성하기 위한 템플릿 표현식이 포함되어 있습니다.

에픽

환경 설정

작업	설명	필요한 기술
리포지토리를 복제합니다.	<p>다음 명령을 입력하여 자동 생성 PynamoDB 모델 및 CRUD 함수 저장소를 복제합니다.</p> <pre>git clone https://github.com/aws-samples/amazon-reverse-engineer-dynamodb.git</pre>	앱 개발자
Python 환경을 설정합니다.	<ol style="list-style-type: none"> 복제된 리포지토리의 최상위 디렉터리로 이동합니다. <pre>cd amazon-reverse-engineer-dynamodb</pre> 다음 명령을 입력하여 필수 라이브러리 및 패키지를 설치합니다. <pre>pip install -r requirements.txt</pre> 	앱 개발자

PynamoDB 모델 및 CRUD 함수 생성

작업	설명	필요한 기술
<p>JSON 스키마 파일을 수정하십시오.</p>	<ol style="list-style-type: none"> 1. 복제된 리포지토리의 최상위 디렉터리로 이동합니다. <div style="border: 1px solid #ccc; border-radius: 10px; padding: 10px; margin: 10px 0;"> <pre>cd amazon-reverse-engineer-dynamodb</pre> </div> 2. 원하는 편집기에서 <code>test.json</code> 파일을 엽니다. 이 파일을 참조로 사용하여 고유한 JSON 스키마 파일을 만들거나 이 파일의 값을 환경에 맞게 업데이트할 수 있습니다. 3. 대상 DynamoDB 테이블의 이름 AWS 리전, 및 속성 값을 수정합니다. <p>참고: JSON 스키마 파일에 없는 테이블을 정의하는 경우 이 솔루션은 해당 테이블에 대한 모델이나 CRUD 함수를 생성하지 않습니다.</p> 4. <code>test.json</code> 파일을 저장하고 닫습니다. 이 파일은 새 이름으로 저장하는 것이 좋습니다. 	<p>앱 개발자</p>
<p>Python 애플리케이션을 실행합니다.</p>	<p>다음 명령을 입력하여 PynamoDB 모델 및 CRUD 함수를 생성합니다. <code><input_schema.json></code> 여기서는 JSON 스키마 파일의 이름입니다.</p>	<p>앱 개발자</p>

작업	설명	필요한 기술
	<pre>python main.py --file <input_schema.json></pre>	

PynamoDB 모델 및 CRUD 함수를 검증하십시오.

작업	설명	필요한 기술
생성된 PynamoDB 모델을 확인합니다.	<ol style="list-style-type: none"> 복제된 리포지토리의 최상위 디렉터리에서 다음 명령을 입력하여 리포지토리로 이동합니다. <code>models</code> <pre>cd models</pre> <ol style="list-style-type: none"> 기본적으로 이 솔루션은 PynamoDB 모델 파일의 이름을 지정합니다. <code>demo_model.py</code> 이 파일이 있는지 확인하십시오. 	앱 개발자
생성된 CRUD 함수를 확인합니다.	<ol style="list-style-type: none"> 복제된 리포지토리의 최상위 디렉터리에서 다음 명령을 입력하여 리포지토리로 이동합니다. <code>crud</code> <pre>cd crud</pre> <ol style="list-style-type: none"> 기본적으로 이 솔루션은 스크립트 이름을 지정합니다. <code>demo_crud.py</code> 이 파일이 있는지 확인하십시오. <code>demo_crud.py</code> 파일의 Python 클래스를 사용하여 대상 DynamoDB 테이블에서 CRUD 작업을 수행합니다. 	앱 개발자

작업	설명	필요한 기술
	다. 작업이 성공적으로 완료되었는지 확인합니다.	

관련 리소스

- [Amazon DynamoDB의 핵심 구성 요소](#) (DynamoDB 설명서)
- [보조 인덱스를 통한 데이터 액세스 개선](#) (DynamoDB 설명서)

추가 정보

JSON 스키마 파일의 샘플 속성

```
[
  {
    "name": "test_table",
    "region": "ap-south-1",
    "attributes": [
      {
        "name": "id",
        "type": "UnicodeAttribute"
      },
      {
        "name": "name",
        "type": "UnicodeAttribute"
      },
      {
        "name": "age",
        "type": "NumberAttribute"
      }
    ]
  }
]
```

Green Boost를 통한 풀스택 클라우드 네이티브 웹 애플리케이션 개발 살펴보기

작성자: Ben Stickley(AWS), Amiin Samatar(AWS)

환경: PoC 또는 파일럿

기술: 소프트웨어 개발 및 테스트, 웹 및 모바일 앱, 클라우드 네이티브

워크로드: 오픈 소스

AWS 서비스: 아마존 오로라, AWS CDK, 아마존, AWS CloudFront Lambda, AWS WAF

요약

개발자의 진화하는 요구에 부응하기 위해 Amazon Web Services(AWS)는 클라우드 네이티브 웹 애플리케이션 개발의 효율적인 접근 방식에 대한 중요한 요구를 인식하고 있습니다. AWS는 AWS 클라우드에 웹 앱을 배포할 때 발생하는 일반적인 장애물을 극복할 수 있도록 지원하는 데 중점을 두고 있습니다. 이 패턴은 AWS Cloud Development Kit (TypeScriptAWS CDK), React 및 Node.js 같은 최신 기술의 기능을 활용하여 개발 프로세스를 간소화하고 가속화하는 것을 목표로 합니다.

Green Boost(GB) 툴킷을 기반으로 하는 이 패턴은 AWS의 광범위한 기능을 완전히 사용하는 웹 애플리케이션 구성에 필요한 실용적인 가이드를 제공합니다. 이는 포괄적인 로드맵 역할을 하며 Amazon Aurora PostgreSQL 호환 에디션과 통합된 기본 CRUD(생성, 읽기, 업데이트, 삭제) 웹 애플리케이션을 배포하는 프로세스를 안내합니다. 이는 Green Boost 명령줄 인터페이스(CLI)를 사용하고 로컬 개발 환경을 구축함으로써 이루어집니다.

애플리케이션이 성공적으로 배포되면 이 패턴이 인프라 설계, 백엔드 및 프론트엔드 개발, 시각화를 위한 cdk-dia와 같은 필수 도구를 포함하여 웹 앱의 주요 구성 요소를 심층적으로 분석하여 효율적인 프로젝트 관리를 촉진합니다.

사전 조건 및 제한 사항

사전 조건

- [Git](#) 설치

- [Visual Studio Code\(VS Code\) 설치](#)
- [AWS Command Line Interface\(AWS CLI\) 설치](#)
- [AWS CDK 툴킷 설치](#)
- [Node.js 18 설치](#) 또는 [pnpm이 설치된 Node.js 18 활성화](#)
- [pnpm](#) 설치(Node.js 설치의 일부가 아닌 경우)
- AWS CDK, Node.js TypeScript, React에 대한 기본 지식
- [활성 상태의 AWS 계정](#)
- [부트스트랩한 AWS 계정](#)(us-east-1에서 AWS CDK를 사용) Amazon CloudFront Lambda @Edge 함수를 지원하려면 us-east-1 AWS 리전이 필요합니다.
- [AWS 보안 인증 정보](#)(터미널 환경에 올바르게 구성된 AWS_ACCESS_KEY_ID 포함)
- Windows 사용자의 경우 관리자 모드의 터미널(pnpm이 노드 모듈을 처리하는 방식을 수용하기 위함)

제품 버전

- JavaScript 버전 3용 AWS SDK
- AWS CDK 버전 2
- AWS CLI 버전 2.2
- Node.js 버전 18
- React 버전 18

아키텍처

대상 기술 스택

- Amazon Aurora PostgreSQL 호환 에디션
- 아마존 CloudFront
- 아마존 CloudWatch
- Amazon Elastic Compute Cloud(Amazon EC2)
- AWS Lambda
- AWS Secrets Manager
- Amazon Simple Notification Service(SNS)

- Amazon Simple Storage Service(S3)
- AWS WAF

대상 아키텍처

다음 다이어그램은 사용자 요청이 Amazon CloudFront, AWS WAF 및 AWS Lambda를 거쳐 S3 버킷, Aurora 데이터베이스, EC2 인스턴스와 상호 작용하여 궁극적으로 개발자에게 도달한다는 것을 보여줍니다. 반면 관리자는 Amazon SNS와 Amazon을 알림 및 모니터링 목적으로 사용합니다. CloudWatch

배포 후 애플리케이션을 더 자세히 살펴보려면 다음 예와 같이 [cdk-dia](#)를 사용하여 다이어그램을 만들 수 있습니다.

이 다이어그램은 웹 애플리케이션 아키텍처를 두 가지 각도에서 보여줍니다. cdk-dia 다이어그램은 Amazon Aurora PostgreSQL 호환 및 AWS Lambda와 같은 특정 AWS 서비스를 강조하여 AWS CDK 인프라에 대한 상세한 기술적 관점을 제공합니다. 반면, 다른 다이어그램은 더 넓은 관점에서 데이터의 논리적 흐름과 사용자 상호 작용을 강조합니다. 주요 차이점은 세부 수준에 있습니다. cdk-dia는 기술적 복잡성을 상세히 다루는 반면 첫 번째 다이어그램은 사용자 중심적인 보기를 제공합니다.

cdk-dia 다이어그램 생성은 AWS CDK를 사용하여 앱 인프라 이해하기 에픽에서 다룹니다.

도구

서비스

- [Amazon Aurora PostgreSQL 호환 에디션](#)은 PostgreSQL 배포를 설정, 운영 및 확장할 수 있는 완전 관리형의 ACID 준수 관계형 데이터베이스 엔진입니다.
- [AWS Cloud Development Kit\(AWS CDK\)](#)는 AWS 클라우드 인프라를 코드로 정의하고 프로비저닝하는 데 도움이 되는 소프트웨어 개발 프레임워크입니다.
- [AWS Command Line Interface \(AWS CLI\)](#)는 명령줄 셸에서 명령을 사용하여 AWS 서비스와 상호 작용할 수 있는 오픈 소스 도구입니다.
- [Amazon](#)은 전 세계 데이터 센터 네트워크를 통해 웹 콘텐츠를 전송함으로써 웹 콘텐츠 배포 CloudFront 속도를 높여 지연 시간을 줄이고 성능을 개선합니다.
- [Amazon](#)은 AWS 리소스의 지표와 AWS에서 실행하는 애플리케이션을 실시간으로 모니터링할 수 있도록 CloudWatch 도와줍니다.

- [Amazon Elastic Compute Cloud\(Amazon EC2\)](#)는 AWS 클라우드에서 확장 가능한 컴퓨팅 용량을 제공합니다. 필요한 만큼 가상 서버를 시작하고 빠르게 스케일 업하거나 스케일 다운할 수 있습니다.
- [AWS Lambda](#)는 서버를 프로비저닝하거나 관리할 필요 없이 코드를 실행하는 데 도움이 되는 컴퓨팅 서비스입니다. 필요할 때만 코드를 실행하며 자동으로 확장이 가능하므로 사용한 컴퓨팅 시간만큼만 비용을 지불합니다.
- [AWS Secrets Manager](#)를 사용하면 코드에 하드코딩된 보안 인증 정보(암호 등)를 Secrets Manager에 대한 API 직접 호출로 바꾸어 프로그래밍 방식으로 보안 암호를 검색할 수 있습니다.
- [AWS Systems Manager](#)는 AWS 클라우드에서 실행되는 애플리케이션과 인프라를 관리하는 데 도움이 됩니다. 애플리케이션 및 리소스 관리를 간소화하고, 운영 문제의 감지 및 해결 시간을 단축하며, AWS 리소스를 규모에 따라 안전하게 관리하는 데 도움이 됩니다. 이 패턴은 AWS Systems Manager Session Manager를 사용합니다.
- [Amazon Simple Storage Service\(S3\)](#)는 원하는 양의 데이터를 저장, 보호 및 검색할 수 있게 해주는 클라우드 기반 객체 스토리지 서비스입니다. [Amazon Simple Notification Service\(SNS\)](#)를 사용하면 웹 서버 및 이메일 주소를 포함하여 게시자와 클라이언트 간의 메시지 교환을 조정하고 관리할 수 있습니다.
- [AWS WAF](#)는 보호되는 웹 애플리케이션 리소스로 전달되는 HTTP 및 HTTPS 요청을 모니터링할 수 있게 해주는 웹 애플리케이션 방화벽입니다.

기타 도구

- [Git](#)은 오픈 소스 분산 버전 제어 시스템입니다.
- [Green Boost](#)는 AWS에서 웹 앱을 구축하기 위한 툴킷입니다.
- [Next.js](#)는 특성 추가 및 최적화를 위한 React 프레임워크입니다.
- [Node.js](#) 는 확장 가능한 네트워크 애플리케이션을 구축하도록 설계된 이벤트 기반 JavaScript 런타임 환경입니다.
- [pgAdmin](#)은 PostgreSQL을 위한 오픈 소스 관리 도구입니다. 데이터베이스 객체를 생성, 유지 관리 및 사용하는 데 도움이 되는 그래픽 인터페이스를 제공합니다.
- [pnpm](#)은 Node.js 프로젝트 종속성을 위한 패키지 관리자입니다.

모범 사례

다음 권장 사항에 대한 자세한 내용은 [에픽](#) 섹션을 참조하세요.

- Amazon CloudWatch 대시보드 및 경보를 사용하여 인프라를 모니터링합니다.

- cdk-nag를 사용하여 정적 코드형 인프라(IaC) 분석을 실행하고 AWS 모범 사례를 적용합니다.
- Systems Manager Session Manager를 사용하여 SSH(보안 셸) 터널링을 통해 DB 포트 전달을 설정합니다. 이렇게 하면 공개적으로 노출된 IP 주소를 사용하는 것보다 더 안전합니다.
- pnpm audit를 실행하여 취약성을 관리합니다.
- [ESLint를 사용하여 정적 TypeScript 코드 분석을 수행하고 Prettier를 사용하여 코드 형식을 표준화하여 모범 사례를 적용하십시오.](#)

에픽

Aurora PostgreSQL-Compatible을 사용하여 CRUD 웹 앱 배포

작업	설명	필요한 기술
Green Boost CLI를 설치합니다.	<p>다음 명령을 실행하여 Green Boost CLI를 설치합니다.</p> <pre>pnpm add -g gboost</pre>	앱 개발자
GB 앱을 생성합니다.	<ol style="list-style-type: none"> 1. Green Boost를 사용하여 앱을 생성하려면 gboost create 명령을 실행합니다. 2. CRUD App with Aurora PostgreSQL 템플릿을 선택합니다. 	앱 개발자
종속 항목을 설치하고 앱을 배포합니다.	<ol style="list-style-type: none"> 1. 프로젝트 디렉터리 cd <your directory> 로 이동합니다. 2. pnpm i 명령을 실행하여 종속 항목을 설치합니다. 3. cd infra 인프라 디렉터리로 이동합니다. 4. 앱을 로컬에 배포하려면 pnpm deploy:local 명령을 실행합니다. 	앱 개발자

작업	설명	필요한 기술
	<p>cdk deploy ... 명령 (infra/package.json 에 정의됨)의 별칭입니다.</p> <p>배포가 완료될 때까지 기다리세요(약 20분). 기다리는 동안 CloudFormation 콘솔에서 AWS CloudFormation 스택을 모니터링하십시오. 코드에 정의된 구성이 배포된 리소스에 어떻게 매핑되는지 확인합니다. 콘솔에서 CDK 구성 트리 보기를 검토하십시오.</p> <p>CloudFormation</p>	

작업	설명	필요한 기술
<p>앱에 액세스합니다.</p>	<p>GB 앱을 로컬에 배포한 후 URL을 사용하여 액세스할 수 있습니다. CloudFront URL은 터미널 출력에 인쇄되지만 찾기가 다소 어려울 수 있습니다. 더 빠르게 찾으려면 다음 단계를 따르세요.</p> <ol style="list-style-type: none"> 1. <code>pnpm deploy:local</code> 명령을 실행한 터미널을 엽니다. 2. 터미널 출력에서 다음 텍스트와 비슷한 섹션을 찾습니다. <pre data-bbox="634 915 1029 1150">myapp5stickbui9C39 A55A.CloudFrontDomainName = d1q16n5pof924c.cloudfront.net</pre> <p>URL은 배포별로 고유합니다.</p> <p>Amazon CloudFront 콘솔에 액세스하여 CloudFront URL을 찾을 수도 있습니다.</p> <ol style="list-style-type: none"> 1. AWS 관리 콘솔에 로그인하고 CloudFront 서비스를 탐색하십시오. 2. 목록에서 가장 최근에 배포된 배포판을 찾아보세요. 	<p>앱 개발자</p>

작업	설명	필요한 기술
	배포와 관련된 도메인 이름을 복사합니다. <code>your-unique-id.cloudfront.net</code> 와 (과) 유사합니다.	

Amazon을 사용하여 모니터링하기 CloudWatch

작업	설명	필요한 기술
CloudWatch 대시보드 보기.	<ol style="list-style-type: none"> 1. CloudWatch 콘솔을 열고 대시보드를 선택합니다. 2. 이름이 <code><appId>-<stageName>-dashboard</code>인 대시보드를 선택합니다. 3. 대시보드를 검토합니다. 모니터링되는 리소스는 무엇인가요? 어떤 지표가 기록되고 있나요? 이 대시보드는 오픈 소스 구조로 만들어졌습니다. cdk-monitoring-constructs 	앱 개발자
알림을 활성화합니다.	<p>CloudWatch 대시보드를 사용하면 웹 앱을 능동적으로 모니터링할 수 있습니다. 알림을 활성화하여 웹 앱을 수동적으로 모니터링할 수 있습니다.</p> <ol style="list-style-type: none"> 1. 모니터 스택을 정의하는 <code>/infra/src/app/stateless/monitor-stack.ts</code> 으로 이동합니다. 	앱 개발자

작업	설명	필요한 기술
	<p>2. 다음 줄의 주석을 제거하고 <code>admin@example.com</code> 을 이메일 주소로 바꿉니다.</p> <pre>onAlarmTopic.addSubscription(new EmailSubscription("admin@example.com "));</pre> <p>3. 다음 중요 정보를 파일 상단에 추가합니다.</p> <pre>import { EmailSubscription } from "aws-cdk-lib/aws-sns-subscriptions";</pre> <p>4. <code>infra/</code>에서 다음 명령을 실행합니다.</p> <pre>cdk deploy "*/monitor" --exclusively.</pre> <p>5. 모니터링 경보가 시작될 때 시작되는 SNS 주제에 대한 구독을 확인하려면 이메일 메시지의 링크를 선택하세요.</p>	

AWS CDK를 사용하여 앱 인프라 이해

작업	설명	필요한 기술
아키텍처 다이어그램을 생성합니다.	<code>cdk-dia</code> 를 사용하여 웹 앱의 아키텍처 다이어그램을 생성합니다. 아키텍처를 시각화하면 팀	앱 개발자

작업	설명	필요한 기술
	<p>원 간의 이해와 의사소통을 개선하는 데 도움이 됩니다. 시스템의 구성 요소와 그 관계에 대한 명확한 개요를 확인할 수 있습니다.</p> <ol style="list-style-type: none">1. Graphviz를 설치합니다.2. <code>infra/</code>에서 <code>pnpm cdk-dia</code> 명령을 실행합니다.3. <code>infra/diagram.png</code> 를 확인합니다.	

작업	설명	필요한 기술
<p>cdk-nag를 사용하여 모범 사례를 적용합니다.</p>	<p>cdk-nag를 사용하면 모범 사례를 적용하고 보안 취약성 및 잘못된 구성의 위험을 줄임으로써 인프라의 보안과 규정 준수를 유지할 수 있습니다.</p> <ol style="list-style-type: none"> 1. 규칙 섹션을 통해 AWS Solutions Library의 규칙 팩에서 확인할 수 있는 사항을 포함하여 cdk-nag의 모범 사례 적용을 살펴보세요. 2. cdk-nag가 규칙을 어떻게 적용하는지 알아보려면 코드를 변경하세요. 예를 들면 <code>infra/src/app/stateful/data-stacks.ts</code> 에서 <code>storageEncrypted: true</code>를 <code>storageEncrypted: false</code> 로 변경합니다. 3. <code>infra/</code>에서 <code>cdk synth */data</code> 명령을 실행합니다. 합성 중에 규칙 위반을 나타내는 빌드 오류가 발생합니다. <pre>AwsSolutions-RDS2: The RDS instance or Aurora DB cluster does not have storage encryption enabled.</pre>	<p>앱 개발자</p>

작업	설명	필요한 기술
	<p>이 오류는 cdk-nag가 인프라 모범 사례를 적용하고 보안 구성 오류를 방지하기 위한 보안 메커니즘이라는 것을 보여줍니다.</p> <p>4. 필요한 경우 다양한 범위에서 규칙을 숨길 수도 있습니다. 예를 들어 AwsSolutions-RDS2를 억제하려면 인스턴스화 아래에 다음 코드를 추가하십시오. DbIamCluster</p> <pre data-bbox="634 821 1029 1528"> NagSuppressions.addResourceSuppressions(cluster.node.findChild("Resource"), [{ id: "AwsSolutions-RDS2", reason: "Customer requirement necessitates having unencrypted DB storage", },],); </pre> <p>5. 숨긴 후 <code>cdk synth "**/data"</code> 명령을 다시 실행합니다. 이제 AWS CDK 앱이 성공적으로 합성됩니다. 숨겨진 모든 규칙은 <code>infra/cdk.out/asse</code></p>	

작업	설명	필요한 기술
	mbly-<appId>-<stageName>/AwsSolutions-<appId>-<stageName>-\${stackId}-NagReport.csv 에서 찾을 수 있습니다.	

데이터베이스 구성 및 스키마 평가

작업	설명	필요한 기술
환경 변수를 가져옵니다.	<p>다음 단계를 사용하여 필요한 환경 변수를 얻을 수 있습니다.</p> <ol style="list-style-type: none"> 1. DB_BASTION_ID 를 찾으려면 콘솔에 로그인한 다음 EC2 콘솔로 이동합니다. 인스턴스 (실행 중) 를 선택하고 - 이름이 포함된 행을 찾습니다. ssm-db-bastion <stageName> 인스턴스 ID 는 i-로 시작합니다. 2. DB_ENDPOINT 를 찾으려면 Amazon Relational Database Service(RDS) 콘솔에서 DB 인스턴스를 선택하고 <appId>-<stageName>-data-로 시작하는 DB 식별자가 있는 리전 클러스터를 선택합니다. rds.amazonaws.com으로 끝나는 라이터 인스턴스 엔드포인트를 찾습니다. 	앱 개발자

작업	설명	필요한 기술
포트 전달을 설정합니다.	<p>다음 단계에 따라 포트 전달을 설정하세요.</p> <ol style="list-style-type: none"> 1. AWS Systems Manager Session Manager 플러그인을 설치합니다. 2. <code>pnpm db:connect</code> 의 실행 이후(<code>core/</code>에서) Bastion Host를 통해 보안 연결을 설정하여 포트 전달을 시작합니다. 3. 터미널에 <code>Waiting for connections...</code>, 텍스트가 표시되면 EC2 Bastion Host를 통해 로컬 시스템과 Aurora 서버 사이에 SSH 터미널이 성공적으로 설정된 것입니다. 	앱 개발자
Systems Manager Session Manager의 제한 시간을 조정합니다.	(선택 사항) 기본 20분 세션 제한 시간이 너무 짧으면 Systems Manager 콘솔에서 세션 관리자, 기본 설정, 편집, 유틸 세션 제한 시간을 선택하여 최대 60분까지 늘릴 수 있습니다.	앱 개발자

작업	설명	필요한 기술
<p>데이터베이스를 시각화합니다.</p>	<p>pgAdmin은 PostgreSQL 데이터베이스를 관리하기 위한 사용자 친화적인 오픈 소스 도구입니다. 데이터베이스 작업을 단순화하여 데이터베이스를 효율적으로 생성, 관리, 최적화할 수 있습니다. 이 섹션에서는 pgAdmin을 설치하고 PostgreSQL 데이터베이스 관리에 pgAdmin 특성을 사용하는 방법을 안내합니다.</p> <ol style="list-style-type: none"> 1. 객체 탐색기에서 서버의 컨텍스트(마우스 오른쪽 버튼 클릭) 메뉴를 열고 등록, 서버를 선택합니다. 2. 일반 탭에서 <appId>-<stageName> 값을 이름 필드에 입력합니다. 3. DB 암호를 가져오려면 AWS Secrets Manager 콘솔을 열고 스택에 대해 CDK에서 생성: <appId>-<stageName>-data 설명이 있는 보안 암호를 선택하고 보안 암호 값을 선택합니다. 보안 암호 값을 선택하고 암호 키와 함께 보안 암호 값을 복사합니다. 4. 연결 탭에서 0.0.0을 호스트 이름/주소 필드에 입력하고, <appId>_admin을 사용자 이름 필드에 입력합니다. 암호 필드에는 이전에 가져온 보 	<p>앱 개발자</p>

작업	설명	필요한 기술
	<p>안 암호를 사용합니다. 암호를 저장하시겠습니까?에서 예를 선택합니다.</p> <p>5. 저장을 선택합니다.</p> <p>6. 테이블을 보려면 <appId>-<stageName>, 데이터베이스, <appId>_db, 스키마, <appId>, 표로 이동합니다.</p> <p>7. 항목 표의 컨텍스트 (마우스 오른쪽 버튼 클릭) 메뉴를 열고 데이터 보기 및 편집, 모든 행을 선택합니다.</p> <p>8. 표를 탐색합니다.</p>	

Node.js로 디버깅

작업	설명	필요한 기술
항목 생성 사용 사례를 디버깅합니다.	<p>다음 절차에 따라 항목 생성 사용 사례를 디버깅합니다.</p> <p>1. core/src/modules/item/create-item.use-case.ts 파일을 열고 다음 코드를 삽입합니다.</p> <pre>import { fileURLToPath } from "node:url"; // existing create-item.use-case.ts code here</pre>	앱 개발자

작업	설명	필요한 기술
	<pre data-bbox="646 212 977 682"> if (process.argv[1] === fileURLTo Path(import.meta.u rl)) { createItemUseCase({ description: "Item 1's Descripti on", name: "Item 1", }); } </pre> <p data-bbox="592 720 1026 1039">2. 이전 단계에서 추가된 코드는 이 모듈이 직접 실행될 때 createItemUseCase 함수가 호출되게 합니다. 이 코드 블록 내에서 디버깅을 line-by-line 시작하려는 줄에 중단점을 설정합니다.</p> <p data-bbox="592 1119 1026 1770">1. VS Code JavaScript 디버그 터미널을 연 다음 pnpm tsx core/src/modules/item/create-item.use-case.ts 실행하여 디버깅과 함께 코드를 실행합니다. line-by-line 또는 console.log 명령문을 사용할 수 있지만 복잡한 비즈니스 로직으로 작업할 경우 인쇄 명령문이 적합하지 않을 수 있습니다. Line-by-line 디버깅은 더 많은 컨텍스트를 제공합니다.</p>	

프론트엔드 개발

작업	설명	필요한 기술
개발 서버를 설정합니다.	<ol style="list-style-type: none"> 1. <code>ui/</code> 위치로 이동하고 <code>pnpm dev</code>을 실행하여 Next.js 개발 서버를 시작합니다. 2. <code>http://localhost:3000</code> 에서 로컬로 웹 앱에 액세스할 수 있습니다. Next.js 개발 서버는 React 구성 요소의 편집 내용에 대해 빠른 새로 고침으로 즉각적인 피드백을 제공하도록 설정되어 있습니다. 3. 앱 바 색상을 사용자 지정해 보세요. <code>ui/src/components/theme/theme.tsx</code> 파일을 열고 앱 바의 테마를 정의하는 섹션을 찾습니다. <code>colorSchemes.light.palette.primary</code> 섹션에서 기본 값을 <code>colors.lagoon</code> 에서 <code>colors.carrot</code> 으로 업데이트합니다. 이렇게 변경한 후에는 파일을 저장하고 브라우저에서 업데이트를 관찰합니다. 4. 텍스트, 구성 요소를 수정하고 새 페이지를 추가해 보세요. 	앱 개발자

Green Boost를 사용한 툴링

작업	설명	필요한 기술
<p>monorepo 및 pnpm 패키지 관리자를 설정합니다.</p>	<ol style="list-style-type: none"> 1. GB 리포지토리의 루트에서 <code>pnpm-workspace.yaml</code> 를 살펴보고 워크스페이스가 어떻게 정의되어 있는지 확인합니다. 워크스페이스에 대한 자세한 내용은 pnpm 설명서를 참조하세요. 2. <code>ui/package.json</code> 에서 검토 후 <code>core/</code>에서(패키지 이름 "<code><appId>/core</code>": "<code>workspace:^</code>",) 워크스페이스를 어떻게 참조하는지 확인합니다. 3. 내에 정의된 유틸리티 패키지에서 ESLint 구성이 어떻게 TypeScript 중앙 집중화되는지 살펴보세요. <code>packages/</code> 이 구성은 <code>core/</code>, <code>infra/</code>, <code>ui/</code> 같은 애플리케이션 패키지에서 사용됩니다. 이는 앱을 확장하고, 구성 코드를 복제하지 않고도 유틸리티 패키지를 참조할 수 있는 애플리케이션 패키지를 추가로 정의할 때 유용합니다. 	<p>앱 개발자</p>
<p>pnpm 스크립트를 실행합니다.</p>	<p>리포지토리의 루트에서 다음 명령을 실행합니다.</p> <ol style="list-style-type: none"> 1. <code>pnpm lint</code>를 실행합니다. 이 명령은 ESLint를 사용하 	<p>앱 개발자</p>

작업	설명	필요한 기술
	<p>여 정적 코드 분석을 실행합니다.</p> <p>2. <code>pnpm typecheck</code> 를 실행합니다. 이 명령은 TypeScript 컴파일러를 실행하여 코드 유형을 확인합니다.</p> <p>3. <code>pnpm test</code>를 실행합니다. 이 명령은 Vitest를 실행하여 유닛 테스트를 실행합니다.</p> <p>모든 워크스페이스에서 이 명령이 어떻게 실행되는지 확인해 보세요. 명령은 각 워크스페이스의 <code>package.json#scripts</code> 필드에 정의되어 있습니다.</p>	

작업	설명	필요한 기술
정적 코드 분석에 ESLint를 사용합니다.	<p>다음을 수행하여 ESLint의 정적 코드 분석 기능을 테스트합니다.</p> <ol style="list-style-type: none"> 1. 먼저 VS Code ESLint 확장 프로그램(ID: dbaeumer.vscode-eslint)이 설치되어 있는지 확인합니다. 인라인으로 오류를 확인하려면 VS Code Error Lens(ID: usernamehw.errorlens)도 설치하는 것이 좋습니다. 2. 다음 예제와 같이 코드에 <code>eval()</code> 함수를 사용하는 코드 줄을 의도적으로 포함합니다. <div data-bbox="630 1056 1029 1415" style="border: 1px solid #ccc; border-radius: 10px; padding: 10px; margin: 10px 0;"> <pre>const userInput = "import("fs").then ((fs) => console.l og(fs.readFileSync ("/etc/passwd", { encoding: "utf8" })))"; eval(userInput);</pre> </div> <p>중요: 이는 테스트 목적으로만 사용됩니다. <code>eval()</code> 사용은 잠재적으로 위험한 것으로 간주되므로 보안 위험 때문에 피해야 합니다.</p> 3. <code>eval()</code> 줄을 포함한 후 코드 편집기를 열어 ESLint가 빨간색 물결선을 사용하여 	앱 개발자

작업	설명	필요한 기술
	<p>코드 스멜을 표시했는지 확인합니다.</p> <p>4. <code>packages/eslint-config-{node,next}/.eslintrc.cjs</code> 에서 ESLint 플러그인 및 구성을 검토합니다.</p>	
<p>종속성과 취약성을 관리합니다.</p>	<ol style="list-style-type: none"> 1. 일반적인 취약성 및 노출 (CVE)을 식별하려면 리포지토리의 루트에서 <code>pnpm audit</code>를 실행합니다. <p>알려진 취약성 찾을 수 없음이라는 메시지가 표시됩니다.</p> <ol style="list-style-type: none"> 2. <code>core/</code>에서 <code>pnpm add minimist@0.2.3</code> , <code>pnpm audit</code>를 차례로 실행하여 의도적으로 취약한 패키지를 설치합니다. 보고된 취약성을 확인합니다. 3. <code>core/</code>에서 <code>pnpm remove minimist</code>를 실행하여 취약한 패키지를 제거합니다. 	<p>앱 개발자</p>

작업	설명	필요한 기술
<p>Husky를 사용하여 후크를 사전 커밋합니다.</p>	<ol style="list-style-type: none"> 리포지토리 전체에서 TypeScript 파일을 약간만 변경하세요. 변경 작업은 설명 추가와 같은 기본적인 작업입니다. <code>git add -A</code> 및 <code>git commit -m "test husky"</code>의 사용으로 이러한 변경 내용을 스테이징하고 커밋합니다. <p><code>.husky/pre-commit</code>에 정의된 Husky 사전 커밋 후크 트리거는 <code>pnpm lint-staged</code> 명령을 실행합니다.</p> lint-staged가 Git에서 스테이징된 파일의 리포지토리 전체에서 <code>*/.lintstagedrc.js</code> 파일에 지정된 명령을 실행하는 방법을 관찰하세요. <p>이러한 도구는 잘못된 코드가 애플리케이션에 유입되는 것을 방지하는 데 도움이 되는 메커니즘입니다.</p>	<p>앱 개발자</p>

인프라 해체

작업	설명	필요한 기술
계정에서 배포를 제거합니다.	<ol style="list-style-type: none"> 첫 번째 에픽에서 프로비저닝한 인프라를 해체하려면 <code>pnpm destroy:local</code> 을 <code>infra/</code>에서 실행합니다. <code>pnpm destroy:local</code> 완료 후 15분 정도 기다린 다음 Lambda 콘솔에서 앱 ID 를 검색하여 보관된 Lambda @Edge 함수를 삭제합니다. Lambda @Edge 함수는 복제되므로 삭제하기가 어렵습니다. Lambda @Edge 함수 삭제에 대한 자세한 내용은 설명서를 참조하십시오. CloudFront 	앱 개발자

문제 해결

문제	Solution
포트 전달을 설정할 수 없음	<p>AWS 보안 인증 정보가 적절하게 구성되어 있고 필요한 권한이 있는지 확인합니다.</p> <p>Bastion Host ID(<code>DB_BASTION_ID</code>) 및 데이터베이스 엔드포인트(<code>DB_ENDPOINT</code>) 환경 변수가 올바르게 설정되었는지 다시 확인합니다.</p> <p>여전히 문제가 발생하는 경우 AWS 설명서에서 SSH 연결 및 세션 관리자 문제 해결을 참조하세요.</p>

문제	Solution
<p>웹사이트가 localhost:3000 에서 로드되지 않음</p>	<p>터미널 출력에 전송 주소를 포함하여 포트 전달이 성공했다고 표시되는지 확인합니다.</p> <p>로컬 시스템에서 포트 3000을 사용하여 충돌하는 프로세스가 없는지 확인합니다.</p> <p>Green Boost 애플리케이션이 제대로 구성되고 예상 포트(3000)에서 실행되고 있는지 확인합니다.</p> <p>웹 브라우저에서 로컬 연결을 차단할 수 있는 보안 확장 또는 설정이 있는지 확인합니다.</p>
<p>로컬 배포 중 오류 메시지(pnpm deploy:local)</p>	<p>오류 메시지를 주의 깊게 검토하여 문제의 원인을 파악합니다.</p> <p>필요한 환경 변수와 구성 파일이 올바르게 설정되었는지 확인합니다.</p>

관련 리소스

- [AWS CDK 설명서](#)
- [Green Boost 설명서](#)
- [Next.js 설명서](#)
- [Node.js 설명서](#)
- [React 설명서](#)
- [TypeScript 설명서](#)

AWS를 GitHub 사용하여 Node.js 애플리케이션에 대한 단위 테스트 실행 CodeBuild

작성자: Thomas Scott(AWS) 및 Jean-Baptiste Guillois(AWS)

코드 리포지토리: [노드 JS 테스트 샘플](#)

환경: 프로덕션

기술: 소프트웨어 개발 및 테스트

AWS 서비스: AWS CodeBuild

요약

이 패턴은 Node.js 게임 API의 샘플 소스 코드와 키 유닛 테스트 구성 요소를 제공합니다. 또한 지속적 통합 및 지속적 전달 (CI/CD) 워크플로의 일환으로 CodeBuild AWS를 사용하여 GitHub 리포지토리에 서 이러한 단위 테스트를 실행하기 위한 지침도 포함되어 있습니다.

유닛 테스트는 유닛이라고 하는 애플리케이션의 여러 부분이 올바르게 작동하는지 개별적으로 독립적으로 테스트하는 소프트웨어 개발 프로세스입니다. 테스트를 통해 코드의 품질을 검증하고 예상대로 작동하는지 확인합니다. 다른 개발자들도 테스트를 참조하여 코드 베이스에 쉽게 익숙해질 수 있습니다. 유닛 테스트는 향후 리팩토링 시간을 줄이고, 엔지니어가 코드 베이스를 더 빠르게 파악하도록 지원하며, 예상되는 동작에 대한 확신을 심어줍니다.

유닛 테스트에는 AWS Lambda 함수를 비롯한 개별 함수를 테스트하는 작업이 포함됩니다. 유닛 테스트를 생성하려면 테스트 프레임워크와 테스트(어설션)를 검증하는 방법이 필요합니다. 이 패턴의 코드 예제는 [Mocha](#) 테스트 프레임워크와 [Chai 어설션 라이브러리](#)를 사용합니다.

유닛 테스트 및 테스트 구성 요소 예제에 대한 자세한 내용은 [추가 정보](#) 섹션을 참조하십시오.

사전 조건 및 제한 사항

- 올바른 CodeBuild 권한을 가진 활성 AWS 계정
- GitHub 계정 ([가입 지침](#) 참조)
- Git([설치 지침](#) 참조)
- 변경하고 코드를 푸시하기 위한 코드 편집기 GitHub (예: [AWS Cloud9](#) 사용 가능)

아키텍처

다음 다이어그램은 이 패턴이 구축하는 아키텍처를 보여줍니다.

도구

도구

- [Git](#) - Git은 코드 개발에 사용할 수 있는 버전 관리 시스템입니다.
- [AWS Cloud9](#) - AWS Cloud9은 여러 프로그래밍 언어와 런타임 디버거, 내장 터미널을 지원하여 풍부한 코드 편집 환경을 제공하는 통합 개발 환경(IDE)입니다. 소프트웨어를 코딩, 빌드, 실행, 테스트 및 디버깅하기 위한 도구 모음을 갖추고 있으며 클라우드에 소프트웨어를 릴리스하는 데에도 도움이 됩니다. 웹 브라우저를 통해 AWS Cloud9 IDE에 액세스합니다.
- [AWS CodeBuild](#) - CodeBuild AWS는 소스 코드를 컴파일하고, 테스트를 실행하고, 배포 준비가 완료된 소프트웨어 패키지를 생성하는 완전관리형 지속적 통합 서비스입니다. 사용하면 자체 빌드 서버를 프로비저닝, 관리 및 확장할 필요가 없습니다. CodeBuild CodeBuild 지속적으로 확장하고 여러 빌드를 동시에 처리하므로 빌드가 대기열에 남아 있지 않아도 됩니다. 사전 패키징된 빌드 환경을 사용하면 신속하게 시작할 수 있으며 혹은 자체 빌드 도구를 사용하는 사용자 지정 빌드 환경을 만들 수 있습니다. 를 사용하면 사용한 CodeBuild 컴퓨팅 리소스에 대해 분 단위로 요금이 청구됩니다.

코드

이 패턴의 소스 코드는 [샘플 게임 유닛 테스트 애플리케이션](#) 리포지토리에서 확인할 수 있습니다. GitHub 이 샘플에서 자체 GitHub 리포지토리를 만들거나 (옵션 1) 이 패턴에 샘플 리포지토리를 직접 사용 (옵션 2) 할 수 있습니다. 다음 단원의 각 옵션에 대한 지침을 참조하십시오. 선택한 옵션은 사용 사례에 따라 다릅니다.

에픽

옵션 1 - 다음을 사용하여 개인 GitHub 리포지토리에서 단위 테스트를 실행합니다. CodeBuild

작업	설명	필요한 기술
샘플 프로젝트를 기반으로 자체 GitHub 리포지토리를 생성합니다.	1. 에 GitHub 로그인하세요.	앱 개발자, AWS 관리자, AWS DevOps

작업	설명	필요한 기술
	<ol style="list-style-type: none"><li data-bbox="591 212 1029 342">2. 새 리포지토리를 생성합니다. 지침은 GitHub 설명서를 참조하십시오.<li data-bbox="591 365 1029 495">3. 샘플 리포지토리를 계정의 새 리포지토리에 복제하고 푸시합니다.	

작업	설명	필요한 기술
새 CodeBuild 프로젝트 생성.	<ol style="list-style-type: none"> 1. AWS 관리 콘솔에 로그인하고 https://console.aws.amazon.com/codesuite/codebuild/home 에서 CodeBuild 콘솔을 엽니다. 2. 빌드 프로젝트 생성을 선택합니다. 3. 프로젝트 구성 섹션에서 프로젝트 이름에 aws-tests-sample-node-js를 입력합니다. 4. 소스 섹션의 소스 공급자에서 선택합니다 GitHub. 5. 리포지토리의 경우 내 GitHub 계정의 리포지토리를 선택한 다음 새로 만든 GitHub 리포지토리에 URL을 붙여넣습니다. 6. 기본 소스 웹훅 이벤트 섹션에서 코드 변경이 이 리포지토리로 푸시될 때마다 다시 빌드를 선택합니다. 7. 이벤트 유형은 푸시를 선택합니다. 8. 환경 섹션에서 관리 이미지, Amazon Linux 2, 최신 이미지를 선택합니다. 9. 기타 모든 옵션에 대해 기본 설정을 사용하고 빌드 프로젝트 생성을 선택합니다. 	앱 개발자, AWS 관리자, AWS DevOps
빌드를 시작합니다.	검토 페이지에서 빌드 시작을 선택하여 빌드를 실행합니다.	앱 개발자, AWS 관리자, AWS DevOps

옵션 2 - 다음을 사용하여 공개 리포지토리에서 단위 테스트 실행 CodeBuild

작업	설명	필요한 기술
<p>새 CodeBuild 빌드 프로젝트를 만드세요.</p>	<ol style="list-style-type: none"> 1. AWS 관리 콘솔에 로그인하고 https://console.aws.amazon.com/codesuite/codebuild/home 에서 CodeBuild 콘솔을 엽니다. 2. 빌드 프로젝트 생성을 선택합니다. 3. 프로젝트 구성 섹션에서 프로젝트 이름에 aws-tests-sample-node-js를 입력합니다. 4. 소스 섹션의 소스 공급자에서 선택합니다 GitHub. 5. 리포지토리의 경우 공개 리포지토리를 선택한 다음 URL https://github.com/aws-samples/ 을 붙여넣습니다 node-js-tests-sample. 6. 환경 섹션에서 관리 이미지, Amazon Linux 2, 최신 이미지를 선택합니다. 7. 기타 모든 옵션에 대해 기본 설정을 사용하고 빌드 프로젝트 생성을 선택합니다. 	<p>앱 개발자, AWS 관리자, AWS DevOps</p>
<p>빌드를 시작합니다.</p>	<p>검토 페이지에서 빌드 시작을 선택하여 빌드를 실행합니다.</p>	<p>앱 개발자, AWS 관리자, AWS DevOps</p>

유닛 테스트 분석

작업	설명	필요한 기술
테스트 결과를 보십시오.	CodeBuild 콘솔에서 CodeBuild 작업의 유닛 테스트 결과를 검토하십시오. 추가 정보 섹션에 표시된 결과와 일치해야 합니다. 이 결과는 GitHub 리포지토리와의 통합을 CodeBuild 검증합니다.	앱 개발자, AWS 관리자, AWS DevOps
웹후크를 적용합니다.	이제 웹후크를 적용할 수 있으므로 코드 변경 사항을 리포지토리의 기본 브랜치에 푸시할 때마다 자동으로 빌드를 시작할 수 있습니다. 지침은 CodeBuild 설명서를 참조하십시오.	앱 개발자, AWS 관리자, AWS DevOps

관련 리소스

- [샘플 게임 유닛 테스트 애플리케이션](#) (샘플 코드가 있는 GitHub 리포지토리)
- [AWS CodeBuild 설명서](#)
- [GitHub 웹후크 이벤트](#) (CodeBuild 설명서)
- [새 리포지토리 만들기](#) (GitHub 문서)

추가 정보

유닛 테스트 결과

프로젝트가 성공적으로 빌드되면 CodeBuild 콘솔에서 다음 테스트 결과를 확인할 수 있습니다.

유닛 테스트 컴포넌트 예시

이 섹션에서는 유닛 테스트에 사용되는 네 가지 유형의 테스트 구성 요소인 어설션, 스파이, 스텝, 모의 객체에 대해 설명합니다. 여기에는 각 구성 요소에 대한 간략한 설명과 코드 예제가 포함되어 있습니다.

어설션

어설션은 예상 결과를 확인하는 데 사용됩니다. 이는 주어진 함수의 예상 응답을 검증하기 때문에 중요한 테스트 구성 요소입니다. 다음 샘플 어설션은 새 게임을 초기화할 때 반환된 ID가 0에서 1000 사이인지 확인합니다.

```
const { expect } = require('chai');
const { Game } = require('../src/index');

describe('Game Function Group', () => {
  it('Check that the Game ID is between 0 and 1000', function() {
    const game = new Game();
    expect(game.id).is.above(0).but.below(1000)
  });
});
```

스파이

스파이는 함수가 실행될 때 어떤 일이 일어나는지 관찰하는 데 사용됩니다. 예를 들어, 함수가 올바르게 직접 호출되었는지 확인할 수 있습니다. 다음 예제는 Game 클래스 객체에서 시작 및 중지 메서드가 직접 호출되는 것을 보여줍니다.

```
const { expect } = require('chai');
const { spy } = require('sinon');

const { Game } = require('../src/index');

describe('Game Function Group', () => {
  it('should verify that the correct function is called', () => {
    const spyStart = spy(Game.prototype, "start");
    const spyStop = spy(Game.prototype, "stop");

    const game = new Game();
    game.start();
    game.stop();
  });
});
```

```

    expect(spyStart.called).to.be.true
    expect(spyStop.called).to.be.true
  });
});

```

스텝

스텝은 함수의 기본 응답을 재정의하는 데 사용됩니다. 이는 함수가 외부 요청을 할 때 특히 유용한데, 왜냐하면 유닛 테스트에서 외부 요청을 하지 않도록 하기 위해서입니다. (외부 요청은 서로 다른 구성 요소 간의 요청을 물리적으로 테스트할 수 있는 통합 테스트에 더 적합합니다.) 다음 예제에서 스텝은 `getId` 함수에서 반환 ID를 강제로 반환합니다.

```

const { expect } = require('chai');
const { stub } = require('sinon');

const { Game } = require('../src/index');

describe('Game Function Group', () => {
  it('Check that the Game ID is between 0 and 1000', function() {
    let generateIdStub = stub(Game.prototype, 'getId').returns(999999);

    const game = new Game();

    expect(game.getId).is.equal(999999);

    generateIdStub.restore();
  });
});

```

모의 객체

모의 객체는 다양한 시나리오를 테스트하기 위해 동작이 미리 프로그래밍된 가짜 메서드입니다. 모의 객체는 스텝의 확장된 형태로 간주될 수 있으며 여러 작업을 동시에 수행할 수 있습니다. 다음 예제에서는 모의 객체를 사용하여 세 가지 시나리오를 검증합니다.

- 함수가 직접 호출됩니다.
- 함수가 인수와 함께 직접 호출됩니다.
- 함수가 정수 9를 반환합니다.

```

const { expect } = require('chai');

```

```
const { .mock } = require('sinon');

const { Game } = require('../src/index');

describe('Game Function Group', () => {
  it('Check that the Game ID is between 0 and 1000', function() {
    let mock = mock(Game.prototype).expects('getId').withArgs().returns(9);

    const game = new Game();
    const id = game.getId();

    mock.verify();
    expect(id).is.equal(9);
  });
});
```

AWS Lambda를 사용하여 육각형 아키텍처로 Python 프로젝트 구조화

작성자: Furkan Oruc(AWS), Dominik Goby(AWS), Darius Kunce(AWS), Michal Ploski(AWS)

환경: PoC 또는 파일럿	기술: 소프트웨어 개발 및 테스트, 클라우드 네이티브, 컨테이너 및 마이크로서비스, 서버리스, 현대화	AWS 서비스: Amazon DynamoDB; AWS Lambda; Amazon API Gateway
----------------	--	--

요약

이 패턴은 AWS Lambda를 사용하여 Python 프로젝트를 육각형 아키텍처로 구성하는 방법을 보여줍니다. 이 패턴은 AWS Cloud Development Kit(AWS CDK)를 코드형 인프라(IaC) 도구로, Amazon API Gateway를 REST API로, Amazon DynamoDB를 지속성 계층으로 사용합니다. 육각형 아키텍처는 도메인 기반 설계 원칙을 따릅니다. 육각형 아키텍처에서 소프트웨어는 도메인, 포트, 어댑터라는 세 가지 구성 요소로 구성됩니다. 육각형 아키텍처와 그 이점에 대한 자세한 내용은 [AWS 기반 육각형 아키텍처 구축](#) 가이드를 참조하세요.

사전 조건 및 제한 사항

사전 조건

- 활성 상태의 AWS 계정.
- Python 경험
- AWS Lambda, AWS CDK, Amazon API Gateway 및 DynamoDB에 대한 지식
- GitHub 계정 ([가입 지침](#) 참조)
- Git([설치 지침](#) 참조)
- 변경하고 코드를 푸시하기 위한 코드 편집기 GitHub (예: [AWS Cloud9](#), [비주얼 스튜디오 코드](#) 또는) [JetBrains PyCharm](#)
- Docker가 설치되고 Docker 데몬이 가동되어 실행 중입니다.

제품 버전

- 버전 2.24.3 이상
- Python, 버전 3.7 이상.
- AWS CDK v2
- Poetry 버전 1.1.13 이상.
- AWS Lambda Powertools for Python 버전 1.25.6 이상
- pytest 버전 7.1.1 이상
- Moto 버전 3.1.9 이상
- pydantic 버전 1.9.0 이상
- Boto3 버전 1.22.4 이상
- mypy-boto3-dynamodb 버전 1.24.0 이상

아키텍처

대상 기술 스택

대상 기술 스택은 API Gateway, Lambda 및 DynamoDB를 사용하는 Python 서비스로 구성됩니다. 이 서비스는 DynamoDB 어댑터를 사용하여 데이터를 유지합니다. Lambda를 진입점으로 사용하는 함수를 제공합니다. 이 서비스는 Amazon API Gateway를 사용하여 REST API를 노출합니다. API는 AWS Identity and Access Management(IAM)를 [클라이언트 인증](#)에 사용합니다.

대상 아키텍처

구현을 설명하기 위해 이 패턴은 서버리스 대상 아키텍처를 배포합니다. 클라이언트는 API Gateway 엔드포인트에 요청을 보낼 수 있습니다. API Gateway는 육각형 아키텍처 패턴을 구현하는 대상 Lambda 함수에 요청을 전달합니다. Lambda 함수는 DynamoDB 테이블에서 생성, 읽기, 업데이트 및 삭제(CRUD) 작업을 수행합니다.

중요: 이 패턴은 PoC 환경에서 테스트되었습니다. 아키텍처를 프로덕션 환경에 배포하기 전에 보안 검토를 수행하여 위협 모델을 식별하고 보안 코드 베이스를 만들어야 합니다.

API는 제품 엔티티에 대한 다섯 가지 작업을 지원합니다.

- GET /products는 모든 제품을 돌려 보냅니다.
- POST /products는 새로운 제품을 생성합니다.

- GET /products/{id}는 특정 제품을 돌려 보냅니다.
- PUT /products/{id}는 특정 제품을 업데이트합니다.
- DELETE /products/{id}는 특정 제품을 삭제합니다.

다음 폴더 구조를 사용하여 육각형 아키텍처 패턴을 따르도록 프로젝트를 구성할 수 있습니다.

```

app/ # application code
|--- adapters/ # implementation of the ports defined in the domain
|   |--- tests/ # adapter unit tests
|--- entrypoints/ # primary adapters, entry points
|   |--- api/ # api entry point
|       |--- model/ # api model
|       |--- tests/ # end to end api tests
|--- domain/ # domain to implement business logic using hexagonal architecture
|   |--- command_handlers/ # handlers used to execute commands on the domain
|   |--- commands/ # commands on the domain
|   |--- events/ # events triggered via the domain
|   |--- exceptions/ # exceptions defined on the domain
|   |--- model/ # domain model
|   |--- ports/ # abstractions used for external communication
|   |--- tests/ # domain tests
|--- libraries/ # List of 3rd party libraries used by the Lambda function
infra/ # infrastructure code
simple-crud-app.py # AWS CDK v2 app

```

도구

서비스

- [Amazon API Gateway](#)는 어떤 규모에서든 개발자가 API를 손쉽게 생성, 게시, 유지 관리, 모니터링 및 보호할 수 있도록 지원하는 완전관리형 서비스입니다.
- [Amazon DynamoDB](#)는 모든 규모에서 고성능 애플리케이션을 실행하도록 설계된 완전 관리형 서버리스 키 값 NoSQL 데이터베이스입니다.
- [AWS Lambda](#)는 서버를 프로비저닝하거나 관리하지 않고도 사실상 모든 유형의 애플리케이션이나 백엔드 서비스에 대한 코드를 실행할 수 있는 서버리스 이벤트 기반 컴퓨팅 서비스입니다. 200개 이상의 AWS 서비스와 서비스형 소프트웨어(SaaS) 애플리케이션에서 Lambda 함수를 실행할 수 있으며 사용한 만큼만 비용을 지불하면 됩니다.

도구

- [Git](#)은 이 패턴의 코드 개발을 위한 버전 제어 시스템으로 사용됩니다.
- [Python](#)은 이 패턴의 프로그래밍 언어로 사용됩니다. Python은 높은 수준의 데이터 구조와 객체 지향 프로그래밍에 대한 접근 방식을 제공합니다. AWS Lambda는 Python 서비스의 운영을 간소화하는 내장 Python 런타임을 제공합니다.
- [Visual Studio Code](#)는 이 패턴의 개발 및 테스트를 위한 IDE로 사용됩니다. Python 개발을 지원하는 모든 IDE (예: [AWS Cloud9](#) 또는) 를 사용할 수 있습니다. [PyCharm](#)
- [AWS Cloud Development Kit\(AWS CDK\)](#)는 익숙한 프로그래밍 언어를 사용하여 클라우드 애플리케이션 리소스를 정의할 수 있는 오픈 소스 소프트웨어 개발 프레임워크입니다. 이 패턴은 CDK를 사용하여 클라우드 인프라를 코드로 작성하고 배포합니다.
- [Poetry](#)는 이 패턴에서 종속성을 관리하는 데 사용됩니다.
- [Docker](#)는 AWS CDK에서 Lambda 패키지 및 계층을 구축하는 데 사용됩니다.

code

이 패턴의 코드는 GitHub [Lambda 육각형](#) 아키텍처 샘플 리포지토리에서 사용할 수 있습니다.

모범 사례

프로덕션 환경에서 이 패턴을 사용하려면 다음 모범 사례를 따릅니다.

- AWS KMS (키 관리 서비스) 의 고객 관리 키를 사용하여 아마존 [로그 그룹](#)과 [CloudWatch Amazon DynamoDB](#) 테이블을 암호화할 수 있습니다.
- [Amazon API Gateway](#)용 [AWS WAF](#)를 구성하여 조직의 네트워크에서만 액세스할 수 있도록합니다.
- IAM이 요구 사항을 충족하지 못하는 경우 API Gateway 인증을 위한 다른 옵션을 고려합니다. 예를 들어 [Amazon Cognito 사용자 풀](#) 또는 [API Gateway Lambda 권한 부여자](#)를 사용할 수 있습니다.
- [DynamoDB 백업](#)을 사용합니다.
- [Virtual Private Cloud\(VPC\) 배포](#)로 Lambda 함수를 구성하여 네트워크 트래픽을 클라우드 내에 유지합니다.
- [Cross-origin resource sharing\(CORS\) 프리플라이트](#)용 허용된 소스 구성을 업데이트하여 요청한 소스 도메인으로만 액세스를 제한합니다.
- [cdk-nag](#)를 사용하여 AWS CDK 코드에서 보안 모범 사례를 확인합니다.
- 코드 스캔 도구를 사용하여 코드에서 일반적인 보안 문제를 찾아봅니다. 예를 들어 [Bandit](#)은 Python 코드에서 일반적인 보안 문제를 찾도록 설계된 도구입니다. [PIP-Audit](#)은 Python 환경에서 알려진 취약점이 있는 패키지를 검색합니다.

이 패턴은 [AWS X-Ray](#)를 사용하여 애플리케이션의 진입점, 도메인 및 어댑터를 통해 요청을 추적합니다. AWS X-Ray는 개발자가 병목 현상을 식별하고 높은 지연 시간을 파악하여 애플리케이션 성능을 개선할 수 있도록 지원합니다.

에픽

프로젝트 초기화

작업	설명	필요한 기술
자체 리포지토리를 생성합니다.	<ol style="list-style-type: none"> 에 로그인하십시오. GitHub 새 리포지토리를 생성합니다. 지침은 GitHub 설명서를 참조하십시오. 이 패턴의 샘플 리포지토리를 복제하여 계정의 새 리포지토리에 푸시합니다. 	앱 개발자
종속 항목 설치	<ol style="list-style-type: none"> Poetry를 설치합니다. <pre>pip install poetry</pre> 루트 디렉터리에서 패키지를 설치합니다. 다음 명령은 애플리케이션과 AWS CDK 패키지를 설치합니다. 또한 유닛 테스트를 실행하는 데 필요한 개발 패키지도 설치합니다. 설치된 모든 패키지는 새 가상 환경에 배치됩니다. <pre>poetry install</pre> 설치된 패키지를 그래픽으로 보려면 다음 명령을 실행합니다. 	앱 개발자

작업	설명	필요한 기술
	<pre>poetry show --tree</pre> <p>4. 모든 종속 항목을 업데이트 합니다.</p> <pre>poetry update</pre> <p>5. 새로 만든 가상 환경 내에서 새 셸을 엽니다. 여기에는 설치된 모든 종속성이 포함됩니다.</p> <pre>poetry shell</pre>	

작업	설명	필요한 기술
<p>IDE를 구성합니다.</p>	<p>Visual Studio Code를 사용하는 것이 좋지만 Python을 지원하는 모든 IDE를 사용할 수 있습니다. 다음 단계는 Visual Studio Code용입니다.</p> <ol style="list-style-type: none"> .vscode/settings 파일을 업데이트합니다. <pre data-bbox="630 617 1029 1493"> { "python.testing.pytestArgs": ["app/adapters/tests", "app/entrypoints/api/tests", "app/domain/tests"], "python.testing.unittestEnabled": false, "python.testing.pytestEnabled": true, "python.envFile": "\${workspaceFolder}/.env", } </pre> <ol style="list-style-type: none"> 프로젝트의 루트 디렉터리에서 .env 파일을 생성합니다. 이렇게 하면 프로젝트의 루트 디렉터리가 PYTHONPATH 에 포함되어 pytest가 프로젝트를 찾고 모든 패키지를 제대로 검색할 수 있습니다. 	<p>앱 개발자</p>

작업	설명	필요한 기술
	<div style="border: 1px solid #ccc; border-radius: 10px; padding: 5px; display: inline-block;">PYTHONPATH=.</div>	
유닛 테스트 실행 옵션 1: Visual Studio Code	<ol style="list-style-type: none"> Poetry에서 관리하는 가상 환경의 Python 인터프리터를 선택합니다. 테스트 탐색기에서 테스트를 실행합니다. 	앱 개발자
유닛 테스트 실행 옵션 2: 셸 명령어 사용.	<ol style="list-style-type: none"> 가상 환경 내에서 새 셸을 시작합니다. <div style="border: 1px solid #ccc; border-radius: 10px; padding: 5px; display: inline-block; margin: 10px 0;">poetry shell</div> 루트 디렉터리에서 pytest 명령을 실행합니다. <div style="border: 1px solid #ccc; border-radius: 10px; padding: 5px; display: inline-block; margin: 10px 0;">python -m pytest</div> <p>또는 Poetry에서 직접 명령을 실행할 수도 있습니다.</p> <div style="border: 1px solid #ccc; border-radius: 10px; padding: 5px; display: inline-block; margin: 10px 0;">poetry run python -m pytest</div> 	앱 개발자

애플리케이션 배포 및 테스트

작업	설명	필요한 기술
임시 보안 인증 정보 요청	<p>cdk deploy 실행 시 셸에서 AWS 보안 인증을 사용하려면, AWS IAM Identity Center(AWS Single Sign-On 후속)를 사용하여 임시 보안 인증 정보를 생성합니다. 지침은 AWS IAM</p>	앱 개발자, AWS DevOps

작업	설명	필요한 기술
	<p>Identity Center에서 CLI를 사용하기 위한 단기 보안 인증 정보를 검색하는 방법 블로그 게시물을 참조하세요.</p>	
<p>애플리케이션을 배포합니다.</p>	<ol style="list-style-type: none"> AWS CDK v2를 설치합니다. <pre>npm install -g aws-cdk</pre> <p>자세한 내용은 AWS CDK 설명서를 참조하세요.</p> AWS CDK를 계정 및 리전에 부트스트랩합니다. <pre>cdk bootstrap aws://12345678900/ us-east-1 --profile aws-profile-name</pre> AWS 프로필을 사용하여 애플리케이션을 AWS CloudFormation 스택으로 배포합니다. <pre>cdk deploy --profile aws-profile-name</pre> 	<p>앱 개발자, AWS DevOps</p>
<p>API 테스트 옵션 1: 콘솔 사용.</p>	<p>API Gateway 콘솔을 사용하여 API를 테스트합니다. API 작업 및 요청/응답 메시지에 대한 자세한 내용은 리포지토리에 있는 readme 파일의 API 사용 섹션을 참조하십시오. GitHub</p>	<p>앱 개발자, AWS DevOps</p>

작업	설명	필요한 기술
API 테스트 옵션 2: Postman 사용.	<p>Postman과 같은 도구를 사용하는 경우.</p> <ol style="list-style-type: none"> 1. 독립 실행형 애플리케이션 또는 브라우저 확장 프로그램으로 Postman을 설치합니다. 2. API Gateway의 엔드포인트 URL을 복사합니다. 형식은 다음과 같습니다. <pre>https://{api-id}.execute-api.{region}.amazonaws.com/{stage}/{path}</pre> <ol style="list-style-type: none"> 3. 권한 부여 탭에서 AWS 서명을 구성합니다. 지침은 API Gateway REST API에 대한 IAM 인증 활성화에 관한 AWS re:Post 문서를 참조하세요. 4. Postman을 사용하여 API 엔드포인트로 요청을 보냅니다. 	앱 개발자, AWS DevOps

서비스 개발

작업	설명	필요한 기술
비즈니스 도메인에 대한 유닛 테스트를 작성합니다.	<ol style="list-style-type: none"> 1. test_ 파일 이름 접두사를 사용하여 app/domain/tests 폴더에 Python 파일을 만듭니다. 	앱 개발자

작업	설명	필요한 기술
	<p>2. 다음 예제를 사용하여 새 비즈니스 로직을 테스트할 수 있는 새 테스트 방법을 생성합니다.</p> <pre data-bbox="630 424 1029 1499">def test_create_product_should_store_in_repository(): # Arrange command = create_product_command.CreateProductCommand(name="Test Product", description="Test Description",) # Act create_product_command_handler.handle_create_product_command(command=command, unit_of_work=mock_unit_of_work) # Assert</pre> <p>3. app/domain/commands 폴더에 명령 클래스를 생성합니다.</p> <p>4. 기능이 새로 추가된 경우 app/domain/command_handlers 폴더에 명령</p>	

작업	설명	필요한 기술
	<p>처리기용 스텝을 생성합니다.</p> <p>5. 아직 비즈니스 로직이 없기 때문에 유닛 테스트를 실행하여 실패하는지 확인합니다.</p> <pre data-bbox="630 529 1029 613">python -m pytest</pre>	

작업	설명	필요한 기술
<p>명령 및 명령 처리기를 구현합니다.</p>	<ol style="list-style-type: none"> 1. 새로 만든 명령 처리기 파일에 비즈니스 로직을 구현합니다. 2. 외부 시스템과 상호 작용하는 모든 종속성에 대해 <code>app/domain/ports</code> 폴더에서 추상 클래스를 선언합니다. <pre data-bbox="634 600 1029 1745"> class ProductsRepository(ABC): @abstractmethod def add(self, product: product.Product) -> None: ... class UnitOfWork(ABC): products: ProductsRepository @abstractmethod def commit(self) -> None: ... @abstractmethod def __enter__(self) -> typing.Any: ... @abstractmethod def __exit__(self, *args) -> None: ... </pre> 3. 추상 포트 클래스를 형식 주석으로 사용하여 새로 선언 	<p>앱 개발자</p>

작업	설명	필요한 기술
	<p>된 종속성을 허용하도록 명령 처리기 서명을 업데이트 합니다.</p> <pre data-bbox="630 380 1027 856">def handle_create_product_command(command: create_product_command.CreateProductCommand, unit_of_work: unit_of_work.UnitOfWork,) -> str: ...</pre> <p>4. 유닛 테스트를 업데이트하여 명령 처리기에 대해 선언된 모든 종속성의 동작을 시뮬레이션합니다.</p> <pre data-bbox="630 1087 1027 1797"># Arrange mock_unit_of_work = unittest.mock.create_autospec(spec=unit_of_work.UnitOfWork, instance=True) mock_unit_of_work.products = unittest.mock.create_autospec(spec=unit_of_work.ProductsRepository, instance=True)</pre>	

작업	설명	필요한 기술
	<p>5. 테스트에서 어설션 로직을 업데이트하여 예상되는 종속성 호출을 확인합니다.</p> <pre data-bbox="630 380 1029 1129"> # Assert mock_unit _of_work.commit.assert_called_once() product = mock_unit_of_work.products.add.call_args.args[0] assertpy.assert_that(product.name).is_equal_to("Test Product") assertpy.assert_that(product.description).is_equal_to("Test Description") </pre> <p>6. 유닛 테스트를 실행하여 성공했는지 확인합니다.</p> <pre data-bbox="630 1272 1029 1346">python -m pytest </pre>	

작업	설명	필요한 기술
<p>보조 어댑터에 대한 통합 테스트를 작성합니다.</p>	<ol style="list-style-type: none"> 1. test_를 파일 이름 접두사로 사용하여 app/adapters/tests 폴더에 테스트 파일을 생성합니다. 2. Moto 라이브러리를 사용하여 AWS 서비스를 모의합니다. <pre data-bbox="634 596 1029 953"> @pytest.fixture def mock_dynamodb(): with moto.mock_dynamodb(): yield boto3.resource("dynamodb", region_name="eu-central-1") </pre> <ol style="list-style-type: none"> 3. 어댑터의 통합 테스트를 위한 새 테스트 방법을 생성합니다. <pre data-bbox="634 1136 1029 1864"> def test_add_and_commit_should_store_product(mock_dynamodb): # Arrange unit_of_work = dynamodb_unit_of_work.DynamoDBUnitOfWork(table_name=TEST_TABLE_NAME, dynamodb_client=mock_dynamodb.meta.client) current_time = datetime.datetime.now(datetime.timezone </pre>	<p>앱 개발자</p>

작업	설명	필요한 기술
	<pre data-bbox="646 210 977 1396"> one.utc).isoformat () new_product_id = str(uuid.uuid4()) new_product = product.Product(id=new_pr oduct_id, name="test- name", descripti on="test-descripti on", createDat e=current_time, lastUpdat eDate=current_time,) # Act with unit_of_w ork: unit_of_w ork.products.add(n ew_product) unit_of_w ork.commit() # Assert </pre> <p data-bbox="592 1438 1031 1759"> 4. app/adapters 폴더에 어댑터 클래스를 생성합니다. 포트 폴더의 추상 클래스를 기본 클래스로 사용합니다. 5. 아직 로직이 없으므로 유닛 테스트를 실행하여 실패하는지 확인합니다. </p>	

작업	설명	필요한 기술
	<pre>python -m pytest</pre>	

작업	설명	필요한 기술
보조 어댑터를 구현합니다.	<ol style="list-style-type: none"> 1. 새로 만든 어댑터 파일에 로직을 구현합니다. 2. 테스트 어설션을 업데이트합니다. <pre data-bbox="630 451 1027 1759"> # Assert with unit_of_work_readonly: product_from_db = unit_of_work_readonly.products.get(new_product_id) assertpy.assert_that(product_from_db).is_not_none() assertpy.assert_that(product_from_db.dict()).is_equal_to({ "id": new_product_id, "name": "test-name", "description": "test-description", "createDate": current_time, "lastUpdateDate": current_time, }) </pre> <ol style="list-style-type: none"> 3. 유닛 테스트를 실행하여 성공했는지 확인합니다. 	앱 개발자

작업	설명	필요한 기술
	<pre>python -m pytest</pre>	

작업	설명	필요한 기술
end-to-end 테스트 작성.	<ol style="list-style-type: none"> 1. test_를 파일 이름 접두사로 사용하여 app/entry points/api/tests 폴더에 테스트 파일을 생성합니다. 2. 테스트에서 Lambda를 호출하는 데 사용할 Lambda 컨텍스트 픽스처를 생성합니다. <pre data-bbox="630 688 1029 1646"> @pytest.fixture def lambda_context(): @dataclass class LambdaContext: function_name: str = "test" memory_limit_in_mb: int = 128 invoked_function_arn: str = "arn:aws:lambda:eu-west-1:809313241:function:test" aws_request_id: str = "52fdcf07-2182-154f-163f-5f0f9a621d72" return LambdaContext() </pre> <ol style="list-style-type: none"> 3. API 호출을 위한 테스트 방법을 생성합니다. 	앱 개발자

작업	설명	필요한 기술
	<pre> def test_create_product(lambda_context): # Arrange name = "TestName" description = "Test description" request = api_model.CreateProductRequest(name=name, description=description) minimal_event = api_gateway_proxy_event.APIGatewayProxyEvent({ "path": "/products", "httpMethod": "POST", "requestContext": { # correlation ID "requestId": "c6af9ac6-7b61-11e6-9a41-93e8deadbeef" }, "body": json.dumps(request.dict()) }) create_product_func_mock = unittest.mock.create_autospec(</pre>	

작업	설명	필요한 기술
	<pre> spec=create_product_command_handler.handle_create_product_command) handler.create_product_command_handler.handle_create_product_command = (create_product_func_mock) # Act handler.handle_event(minimal_event, lambda_context) </pre> <p>4. 아직 로직이 없으므로 유닛 테스트를 실행하여 실패하는지 확인합니다.</p> <pre>python -m pytest</pre>	

작업	설명	필요한 기술
기본 어댑터를 구현합니다.	<p>1. API 비즈니스 로직용 함수를 만들고 이를 API 리소스로 선언합니다.</p> <pre data-bbox="630 394 1029 1150"> @tracer.capture_method @app.post("/products") @utils.parse_event(model=api_model.CreateProductRequest, app_context=app) def create_product(request: api_model.CreateProductRequest,) -> api_model.CreateProductResponse: """Creates a product.""" ... </pre> <p>참고: 표시되는 모든 데코레이터는 AWS Lambda Powertools for Python 라이브러리의 기능입니다. 자세한 내용은 AWS Lambda Powertools for Python 웹사이트를 참조하세요.</p> <p>2. API 로직을 구현합니다.</p> <pre data-bbox="630 1604 1029 1856"> id=create_product_command_handler.handle_create_product_command(command=create_product_comm </pre>	앱 개발자

작업	설명	필요한 기술
	<pre> and.CreateProductCommand(name=request.name, description=request.description, unit_of_work=unit_of_work,), response = api_model.CreateProductResponse(id=id) return response.dict() </pre> <p>3. 유닛 테스트를 실행하여 성공했는지 확인합니다.</p> <pre>python -m pytest</pre>	

관련 리소스

APG 가이드

- [AWS에서 육각형 아키텍처 구축](#)

AWS 참조

- [AWS Lambda 설명서](#)
- [AWS CDK 문서](#)
 - [첫 번째 AWS CDK 앱](#)
- [API Gateway 설명서](#)
 - [IAM 권한을 사용하여 API에 대한 액세스 제어](#)

- [API Gateway 콘솔을 사용하여 REST API 방법 테스트](#)
- [Amazon DynamoDB 설명서](#)

도구

- [git-scm.com 웹사이트](#)
- [Git 설치](#)
- [새 GitHub 리포지토리 만들기](#)
- [Python 웹사이트](#)
- [Python용 AWS Lambda Powertools](#)
- [Postman 웹사이트](#)
- [Python 모의 객체 라이브러리](#)
- [Poetry 웹사이트](#)

IDE

- [Visual Studio Code 웹사이트](#)
- [AWS Cloud9 설명서](#)
- [PyCharm 웹 사이트](#)

패턴 더 보기

- [aws-nuke를 사용하여 AWS 리소스 자동 삭제](#)
- [AWS CodePipeline 및 AWS를 사용하여 스택 세트 배포를 자동화합니다. CodeBuild](#)
- [Cloud Custodian 및 AWS CDK를 사용하여 Systems Manager용 AWS 관리형 정책을 EC2 인스턴스 프로파일에 자동으로 연결](#)
- [Amazon Kinesis Video Streams 및 AWS Fargate를 사용하여 비디오 처리 파이프라인 구축하기](#)
- [서버리스 접근 방식을 사용하여 AWS 서비스를 함께 연결](#)
- [Oracle의 VARCHAR2\(1\) 데이터 유형을 Amazon Aurora PostgreSQL의 부울 데이터 유형으로 변환](#)
- [AWS Copilot을 사용하여 클러스터링된 애플리케이션을 Amazon ECS에 배포](#)
- [테라폼을 사용하여 CloudWatch Synthetics 카나리아를 배포하세요](#)
- [컨테이너 이미지로 Lambda 함수 배포](#)
- [Lambda 함수, Amazon VPC 및 서버리스 아키텍처를 사용하여 정적 아웃바운드 IP 주소 생성](#)
- [AWS Glue 작업과 Python을 사용하여 테스트 데이터 생성](#)
- [다중 계정 환경을 위한 Gitflow 브랜칭 전략 구현 DevOps](#)
- [다중 DevOps 계정 GitHub 환경을 위한 Flow 분기 전략 구현](#)
- [다중 DevOps 계정 환경을 위한 트렁크 브랜칭 전략 구현](#)
- [AWS에서 ASP.NET Web Forms 애플리케이션 현대화](#)
- [Amazon EC2 리눅스 인스턴스에서 ASP.NET 코어 웹 API Docker 컨테이너를 실행합니다.](#)
- [pytest 프레임워크를 사용하여 AWS Glue에서 Python ETL 작업에 대한 유닛 테스트 실행](#)
- [대규모 Db2 z/OS 데이터를 CSV 파일로 Amazon S3에 전송](#)
- [Account Factory에 대한 테라폼\(AFT\) 코드를 로컬에서 검증합니다.](#)

스토리지 및 백업

주제

- [AMS 계정의 S3 버킷에 대한 EC2 인스턴스 쓰기 액세스 허용](#)
- [Snowflake Snowpipe, Amazon S3, Amazon SNS 및 Amazon Data Firehose를 사용하여 Snowflake 데이터베이스에 데이터 스트림을 자동으로 수집합니다.](#)
- [기존 및 새 Amazon EBS 볼륨을 자동으로 암호화하기](#)
- [AWS Cloud의 Stromasys Charon-SSP 에뮬레이터에서 Sun SPARC 서버 백업하기](#)
- [Veeam Backup & Replication을 사용하여 Amazon S3에 데이터를 백업 및 보관](#)
- [AWS에서 VMware 클라우드를 NetBackup 사용하도록 베리타스를 구성합니다.](#)
- [Amazon S3용 DistCp PrivateLink AWS를 사용하여 온 프레미스 하둡 환경에서 Amazon S3로 데이터를 마이그레이션합니다.](#)
- [온프레미스 데이터베이스의 재해 CloudEndure 복구에 사용](#)
- [패턴 더 보기](#)

AMS 계정의 S3 버킷에 대한 EC2 인스턴스 쓰기 액세스 허용

작성자: Mansi Suratwala(AWS)

환경: 프로덕션

기술: 스토리지 및 백업, 데이터베이스, 보안, 자격 증명, 규정 준수, 운영

워크로드: 기타 모든 워크로드

AWS 서비스: Amazon S3,
AWS 관리형 서비스

요약

AWS Managed Services(AMS)를 사용하면 Amazon Web Services(AWS) 인프라를 보다 효율적이고 안전하게 운영할 수 있습니다. AMS 계정에는 AWS 리소스의 표준화된 관리를 위한 보안 가드레일이 있습니다. 한 가지 문제는 기본 Amazon Elastic Compute Cloud(Amazon EC2) 인스턴스 프로파일에서는 Amazon Simple Storage Service(S3) 버킷에 대한 쓰기 액세스를 허용하지 않는다는 것입니다. 하지만 조직에 S3 버킷이 여러 개 있을 수 있으며 EC2 인스턴스의 액세스 제어를 강화해야 할 수도 있습니다. 예를 들어 S3 버킷에 EC2 인스턴스의 데이터베이스 백업을 저장하려고 할 수 있습니다.

이 패턴은 변경 요청(RFC)을 사용하여 EC2 인스턴스에 AMS 계정의 S3 버킷에 대한 쓰기 액세스를 허용하는 방법을 설명합니다. RFC는 관리형 환경을 변경하기 위해 사용자 또는 AMS가 생성하는 요청으로, 여기에는 특정 작업에 대한 [변경 유형](#)(CT) ID가 포함됩니다.

사전 조건 및 제한 사항

사전 조건

- AMS 고급 계정. 이에 대한 자세한 내용은 AWS Managed Services 설명서의 [AMS 운영 계획](#)을 참조하십시오.
- customer-mc-user-role AWS 자격 증명 및 액세스 관리 (IAM) 역할에 액세스하여 RFC를 제출하십시오.
- AMS 계정의 EC2 인스턴스에 설치 및 구성된 AWS Command Line Interface(AWS CLI).
- AMS에서 RFC를 생성하고 제출하는 방법에 대한 이해. 이에 대한 자세한 내용은 AWS 관리형 서비스 설명서의 [AMS 변경 유형이란 무엇입니까?](#)를 참조하십시오.

- 수동 및 자동 변경 유형(CT)에 대한 이해 이에 대한 자세한 내용은 AWS 관리형 서비스 설명서의 [자동 및 수동 CT](#)를 참조하세요.

아키텍처

기술 스택

- AMS
- AWS CLI
- Amazon EC2
- Amazon S3
- IAM

도구

- [AWS Command Line Interface\(AWS CLI\)](#)는 명령줄 셸에서 명령을 사용하여 AWS 서비스와 상호 작용할 수 있는 오픈 소스 도구입니다.
- [AWS Identity and Access Management\(IAM\)](#)는 사용자에 대한 인증 및 권한 부여를 제어함으로써 AWS 리소스에 대한 액세스를 안전하게 관리할 수 있습니다.
- [AWS Managed Services\(AMS\)](#)는 AWS 인프라를 보다 효율적이고 안전하게 운영하는 데 도움이 됩니다.
- [Amazon Simple Storage Service\(S3\)](#)는 원하는 양의 데이터를 저장, 보호 및 검색하는 데 도움이 되는 클라우드 기반 객체 스토리지 서비스입니다.
- [Amazon Elastic Compute Cloud\(Amazon EC2\)](#)는 AWS 클라우드에서 규모를 조정할 수 있는 컴퓨팅 용량을 제공합니다. 필요한 만큼 가상 서버를 시작하고 빠르게 스케일 업하거나 스케일 다운할 수 있습니다.

에픽

RFC를 사용하여 S3 버킷을 생성합니다.

작업	설명	필요한 기술
자동화된 RFC를 사용하여 S3 버킷을 생성합니다.	1. AMS 계정에 로그인하고, 변경 유형 선택 페이지를 선택	AWS 시스템 관리자, AWS 개발자

작업	설명	필요한 기술
	<p>하고, RFC를 선택한 다음 RFC 생성을 선택합니다.</p> <p>2. S3 버킷 생성 자동 RFC를 제출합니다.</p> <p>참고: S3 버킷의 이름을 기록해 두어야 합니다.</p>	

IAM 인스턴스 프로파일을 생성하고 EC2 인스턴스에 연결합니다.

작업	설명	필요한 기술
<p>수동 RFC를 제출하여 IAM 역할을 생성하세요.</p>	<p>AMS 계정이 온보딩되면 기본 customer-mc-ec 2개 인스턴스 프로파일 IAM 인스턴스 프로파일 생성되고 AMS 계정의 각 EC2 인스턴스에 연결됩니다. 그러나 인스턴스 프로파일에는 S3 버킷에 대한 쓰기 권한이 없습니다.</p> <p>쓰기 권한을 추가하려면 IAM 리소스 생성 매뉴얼 RFC를 제출하여 customer_ec2_instance_, customer_deny_policy, and customer_ec2_s3_integration_policy와 같은 세 가지 정책을 포함하는 IAM 역할을 생성하세요.</p> <p>중요: customer_ec2_instance_ 및 customer_deny_policy가 이미 AMS 계정에 이미 있습니다. 그러나 다음 샘플 정책을</p>	<p>AWS 시스템 관리자, AWS 개발자</p>

작업	설명	필요한 기술
	<p>사용하여 customer_ec2_s3_integration_policy 정책을 생성해야 합니다.</p> <pre data-bbox="592 380 1029 1862"> { "Version": "2012-10-17", "Statement": [{ "Sid": "", "Effect": "Allow", "Principal": { "Service": "ec2.amazonaws.com" }, "Action": "sts:AssumeRole" }] } Role Permissions: { "Version": "2012-10-17", "Statement": [{ "Action": ["s3:ListBucket", "s3:GetBucketLocation"], "Resource": ": "arn:aws:s3:::", "Effect": "Allow" }] } </pre>	

작업	설명	필요한 기술
	<pre> "Action": ["s3:GetObject", "s3:PutObject", "s3:ListMultipartU ploadParts", "s3:AbortMultipart Upload"], "Resource ": "arn:aws:s3::/*", "Effect": "Allow" }] } </pre>	
수동 RFC를 제출하여 IAM 인스턴스 프로파일을 대체하세요.	수동 RFC를 제출하여 대상 EC2 인스턴스를 새 IAM 인스턴스 프로파일과 연결하세요.	AWS 시스템 관리자, AWS 개발자
S3 버킷으로의 복사 작업을 테스트합니다.	AWS CLI인 <code>aws s3 cp test.txt s3://<S3 Bucket>/test2.txt</code> 에서 다음 명령을 실행하여 S3 버킷으로의 복사 작업을 테스트합니다.	AWS 시스템 관리자, AWS 개발자

관련 리소스

- [Amazon EC2 인스턴스에 대한 IAM 인스턴스 프로파일 생성](#)
- [S3 버킷 생성\(Amazon S3 콘솔, AWS SDK 또는 AWS CLI 사용\)](#)

Snowflake Snowpipe, Amazon S3, Amazon SNS 및 Amazon Data Firehose를 사용하여 Snowflake 데이터베이스에 데이터 스트림을 자동으로 수집합니다.

작성자: 비카쉬 찬드라 라우트(AWS)

환경: PoC 또는 파일럿

기술: 스토리지 및 백업

요약

이 패턴은 Amazon Web Services(AWS) 클라우드 서비스를 사용하여 지속적인 데이터 스트림을 처리하고 이를 Snowflake 데이터베이스에 로드하는 방법을 설명합니다. 이 패턴은 Amazon Data Firehose를 사용하여 Amazon 심플 스토리지 서비스 (Amazon S3) 에 데이터를 전달하고, Amazon Simple Notification Service (Amazon SNS) 에 데이터를 전달하여 새 데이터가 수신되면 알림을 보내고, Snowflake Snowpipe는 데이터를 Snowflake 데이터베이스에 로드합니다.

이 패턴을 따르면 지속해서 생성된 데이터를 몇 초 만에 분석에 사용할 수 있고, 여러 번의 수동 COPY 명령을 피할 수 있으며, 로드 시 반정형 데이터를 완벽하게 지원할 수 있습니다.

사전 조건 및 제한 사항

사전 조건

- 활성 상태의 AWS 계정.
- Firehose 전송 스트림으로 데이터를 지속적으로 전송하는 데이터 소스입니다.
- Firehose 전송 스트림에서 데이터를 수신하는 기존 S3 버킷입니다.
- 활성 상태의 Snowflake 계정.

제한 사항

- 눈송이 스노우파이프는 Firehose에 직접 연결되지 않습니다.

아키텍처

기술 스택

- 아마존 데이터 파이어호스
- Amazon SNS
- Amazon S3
- Snowflake Snowpipe
- Snowflake 데이터베이스

도구

- [Firehose](#) — Amazon Data Firehose는 Amazon S3, Amazon Redshift, Amazon 서비스, Splunk 및 지원되는 타사 OpenSearch 서비스 공급자가 소유한 모든 사용자 지정 HTTP 엔드포인트 또는 HTTP 엔드포인트와 같은 대상에 실시간 스트리밍 데이터를 전송하는 완전 관리형 서비스입니다.
- [Amazon S3](#) — Amazon Simple Storage Service(S3)는 인터넷에 대한 스토리지입니다.
- [Amazon SNS](#) — Amazon Simple Notification Service(Amazon SNS)는 구독 중인 엔드포인트 또는 클라이언트에 메시지를 전달 또는 전송하는 것을 조정하고 관리합니다.
- [스노우플레이크](#) — 스노우플레이크는 software-as-a S-Service (SaaS) 로 제공되는 분석 데이터 웨어하우스입니다.
- [Snowflake Snowpipe](#) - Snowpipe는 Snowflake 스테이지에서 파일이 제공되는 즉시 파일에서 데이터를 로드합니다.

에픽

Snowflake Snowpipe 설치

작업	설명	필요한 기술
Snowflake에서 CSV 파일을 생성합니다.	Snowflake에 로그인하고 “CREATE FILE FORMAT” 명령을 실행하여 지정된 필드 구분 기호가 포함된 CSV 파일을 생성합니다. 이 명령과 다른 Snowflake 명령에 대한 자세한	개발자

작업	설명	필요한 기술
	내용은 “추가 정보” 섹션을 참조하세요.	
외부 Snowflake 스테이지를 생성합니다.	“CREATE STAGE” 명령을 실행하여 이전에 생성한 CSV 파일을 참조하는 외부 Snowflake 스테이지를 생성합니다. 중요: S3 버킷의 URL, AWS 액세스 키, AWS 비밀 액세스 키가 필요합니다. “SHOW STAGES” 명령을 실행하여 Snowflake 스테이지가 생성되었는지 확인합니다.	개발자
Snowflake 대상 테이블을 생성합니다.	“CREATE TABLE” 명령을 실행하여 Snowflake 테이블을 생성합니다.	개발자
파이프를 생성합니다.	“CREATE PIPE” 명령을 실행하고 명령에 “auto_ingest=true”가 맞는지 확인합니다. “SHOW PIPES” 명령을 실행하여 파이프가 생성되었는지 확인합니다. “notification_channel” 열 값을 복사하여 저장합니다. 이 값은 Amazon S3 이벤트 알림을 구성하는 데 사용됩니다.	개발자

S3 버킷 구성

작업	설명	필요한 기술
S3 버킷에 대한 30일 수명 주기 정책을 생성합니다.	AWS Management Console에 로그인한 후 Amazon S3 콘솔을 엽니다. Firehose의 데이터	시스템 관리자, 개발자

작업	설명	필요한 기술
	<p>가 포함된 S3 버킷을 선택합니다. 그런 다음 S3 버킷에서 “관리” 탭을 선택하고, “수명 주기 규칙 추가”를 선택합니다. “수명 주기 규칙” 대화 상자에 규칙 이름을 입력하고 버킷의 30일 수명 주기 규칙을 구성합니다. 이 스토리와 다른 스토리에 대한 도움이 필요하다면 “관련 리소스” 섹션을 참조하십시오.</p>	
<p>S3 버킷에 대한 IAM 정책을 생성합니다.</p>	<p>AWS Identity and Access Management(IAM) 콘솔을 열고 ‘정책’을 선택합니다. “정책 생성”을 선택한 후 “JSON” 탭을 선택합니다. “추가 정보” 섹션의 정책을 복사하여 JSON 필드에 붙여넣습니다. 이 정책은 “PutObject” 및 “DeleteObject” 권한과 “,” GetObject GetObjectVersion, “ListBucket” 권한을 부여합니다. “정책 검토”를 선택하고 정책 이름을 입력한 후 “정책 생성”을 선택합니다.</p>	<p>시스템 관리자, 개발자</p>

작업	설명	필요한 기술
IAM 역할에 정책을 할당합니다.	IAM 콘솔을 열고 “역할”을 선택한 후 “역할 생성”을 선택합니다. 신뢰할 수 있는 엔터티로 “다른 AWS 계정”을 선택합니다. AWS 계정 ID를 입력하고 “외부 ID 필요”를 선택합니다. 나중에 변경할 자리 표시자 ID를 입력합니다. “다음”을 선택하고 이전에 생성한 IAM 정책을 할당합니다. 그런 다음 IAM 역할을 생성합니다.	시스템 관리자, 개발자
IAM 역할의 Amazon 리소스 이름(ARN)을 복사합니다.	IAM 콘솔을 열고 “역할”을 선택합니다. 이전에 생성한 IAM 역할을 선택한 다음 “Role ARN”을 복사하여 저장합니다.	시스템 관리자, 개발자

Snowflake에서 스토리지 통합 설정

작업	설명	필요한 기술
Snowflake에서 스토리지 통합을 생성합니다.	Snowflake에 로그인하고 “CREATE STORAGE INTEGRATION” 명령을 실행합니다. 그러면 신뢰할 수 있는 관계를 수정되고, Snowflake에 대한 액세스 권한이 부여되고, Snowflake 스테이지에 대한 외부 ID가 제공됩니다.	시스템 관리자, 개발자
Snowflake 계정에 대한 IAM 역할을 검색합니다.	“DESC INTEGRATION” 명령을 실행하여 IAM 역할에 대한 ARN을 검색합니다. 중요: <integration_name>은 이전에	시스템 관리자, 개발자

작업	설명	필요한 기술
	생성한 Snowflake 스토리지 통합의 이름입니다.	
두 개의 열 값을 기록합니다.	“storage_aws_iam_user_arn” 및 “storage_aws_external_id” 열의 값을 복사하여 저장합니다.	시스템 관리자, 개발자

Snowflake Snowpipe가 S3 버킷에 액세스하도록 허용

작업	설명	필요한 기술
IAM 역할 정책을 수정합니다.	IAM 콘솔을 열고 “역할”을 선택합니다. 이전에 생성한 IAM 역할을 선택하고 “신뢰 관계” 탭을 선택합니다. “신뢰 관계 편집”을 선택합니다. “snowflake_external_id”를 이전에 복사한 “storage_aws_external_id” 값으로 변경합니다. “snowflake_user_arn”을 이전에 복사한 “storage_aws_iam_user_arn” 값으로 변경합니다. 그런 다음 “신뢰 정책 업데이트”를 선택합니다.	시스템 관리자, 개발자

S3 버킷에 대한 SNS 알림을 켜고 구성합니다.

작업	설명	필요한 기술
S3 버킷에 대한 이벤트 알림을 켭니다.	Amazon S3 콘솔을 열고 버킷을 선택합니다. “속성”을 선택하고 “고급 설정”에서 “이벤트”를 선택합니다. “알림 추	시스템 관리자, 개발자

작업	설명	필요한 기술
	가”를 선택하고 이 이벤트의 이름을 입력합니다. 이름을 입력하지 않으면 전역 고유 식별자 (GUID)가 사용됩니다.	
S3 버킷에 대한 Amazon SNS 알림을 구성합니다.	“이벤트”에서 “ObjectCreate (모두)”를 선택한 다음 “전송 대상” 드롭다운 목록에서 “SQS Queue”를 선택합니다. “SNS” 목록에서 “SQS 대기열 ARN 추가”를 선택하고 이전에 복사한 “notification_channel” 값을 붙여넣습니다. 그런 다음 “저장”을 선택합니다.	시스템 관리자, 개발자
SNS 주제에 대한 Snowflake SQS 대기열을 구독합니다.	생성한 SNS 주제에 대한 Snowflake SQS 대기열을 구독합니다. 이 단계에 대한 도움이 필요하면 “관련 리소스” 섹션을 참조하세요.	시스템 관리자, 개발자

Snowflake 스테이지 통합 확인

작업	설명	필요한 기술
Snowpipe를 확인하고 테스트합니다.	Snowflake에 로그인하고 Snowflake 스테이지를 엽니다. S3 버킷에 파일을 드롭하고 Snowflake 테이블에 파일이 로드되는지 확인합니다. Amazon S3는 S3 버킷에 새 객체가 나타나면 Snowpipe에 SNS 알림을 전송합니다.	시스템 관리자, 개발자

관련 리소스

- [S3 버킷의 수명 주기 정책 생성](#)
- [Amazon SNS 주제에 대한 SQS 대기열 구독](#)

추가 정보

파일 형식 생성:

```
CREATE FILE FORMAT <name>
TYPE = 'CSV'
FIELD_DELIMITER = '|'
SKIP_HEADER = 1;
```

외부 스테이지 생성:

```
externalStageParams (for Amazon S3) ::=
  URL = 's3://[//]

  [ { STORAGE_INTEGRATION = } | { CREDENTIALS = ( { { AWS_KEY_ID = `` AWS_SECRET_KEY
= `` [ AWS_TOKEN = `` ] } | AWS_ROLE = `` } ) ) } ` ]
  [ ENCRYPTION = ( [ TYPE = 'AWS_CSE' ] [ MASTER_KEY = '' ] |
                   [ TYPE = 'AWS_SSE_S3' ] |
                   [ TYPE = 'AWS_SSE_KMS' [ KMS_KEY_ID = '' ] |
                   [ TYPE = NONE ] )
```

테이블 생성:

```
CREATE [ OR REPLACE ] [ { [ LOCAL | GLOBAL ] TEMP[ORARY] | VOLATILE } | TRANSIENT ]
TABLE [ IF NOT EXISTS ]
<table_name>
( <col_name> <col_type> [ { DEFAULT <expr>
                          | { AUTOINCREMENT | IDENTITY } [ ( <start_num> ,
<step_num> ) | START <num> INCREMENT <num> ] } ]
/* AUTOINCREMENT / IDENTITY supported only for numeric
data types (NUMBER, INT, etc.) */
  [ inlineConstraint ]
[ , <col_name> <col_type> ... ]
[ , outoflineConstraint ]
[ , ... ] )
```

```
[ CLUSTER BY ( <expr> [ , <expr> , ... ] ) ]
[ STAGE_FILE_FORMAT = ( { FORMAT_NAME = '<file_format_name>'
                        | TYPE = { CSV | JSON | AVRO | ORC | PARQUET | XML }
[ formatTypeOptions ] } ) ]
[ STAGE_COPY_OPTIONS = ( copyOptions ) ]
[ DATA_RETENTION_TIME_IN_DAYS = <num> ]
[ COPY GRANTS ]
[ COMMENT = '<string_literal>' ]
```

스테이지 보기:

```
SHOW STAGES;
```

파이프 생성:

```
CREATE [ OR REPLACE ] PIPE [ IF NOT EXISTS ]
[ AUTO_INGEST = [ TRUE | FALSE ] ]
[ AWS_SNS_TOPIC = ]
[ INTEGRATION = '' ]
[ COMMENT = '' ]
AS
```

파이프 보기:

```
SHOW PIPES [ LIKE '<pattern>' ]
[ IN { ACCOUNT | [ DATABASE ] <db_name> | [ SCHEMA ] <schema_name> } ]
```

스토리지 통합 생성:

```
CREATE STORAGE INTEGRATION <integration_name>
TYPE = EXTERNAL_STAGE
STORAGE_PROVIDER = S3
ENABLED = TRUE
STORAGE_AWS_ROLE_ARN = '<iam_role>'
STORAGE_ALLOWED_LOCATIONS = ('s3://<bucket>/<path>/', 's3://<bucket>/<path>/')
[ STORAGE_BLOCKED_LOCATIONS = ('s3://<bucket>/<path>/', 's3://<bucket>/<path>/') ]
```

예제

```
create storage integration s3_int
type = external_stage
```

```

storage_provider = s3
enabled = true
storage_aws_role_arn = 'arn:aws:iam::001234567890:role/myrole'
storage_allowed_locations = ('s3://mybucket1/mypath1/', 's3://mybucket2/mypath2/')
storage_blocked_locations = ('s3://mybucket1/mypath1/sensitivedata/', 's3://
mybucket2/mypath2/sensitivedata/');

```

이 단계에 관한 자세한 내용은 Snowflake 설명서에서 [Amazon S3에 액세스하기 위한 Snowflake 스토리지 통합 구성](#)을 참조하세요.

통합 설명:

```
DESC INTEGRATION <integration_name>;
```

S3 버킷 정책:

```

{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "s3:PutObject",
        "s3:GetObject",
        "s3:GetObjectVersion",
        "s3:DeleteObject",
        "s3:DeleteObjectVersion"
      ],
      "Resource": "arn:aws:s3:::/*"
    },
    {
      "Effect": "Allow",
      "Action": "s3:ListBucket",
      "Resource": "arn:aws:s3:::",
      "Condition": {
        "StringLike": {
          "s3:prefix": [
            "/*"
          ]
        }
      }
    }
  ]
}

```

```
}
```

기존 및 새 Amazon EBS 볼륨을 자동으로 암호화하기

토니 DeMarco (AWS) 와 조쉬 조이 (AWS) 가 제작했습니다.

코드 리포지토리: <https://github.com/aws-samples/aws-system-manager-automation/tree/main/ebs-unencrypted-to-encrypted-resources>

환경: 프로덕션

기술: 스토리지 및 백업, 보안, ID, 규정 준수, 관리 및 거버넌스

AWS 서비스: AWS Config, 아마존 EBS, AWS KMS, AWS Organizations, AWS Systems Manager

요약

Amazon Elastic Block Store(Amazon EBS) 볼륨의 암호화는 조직의 데이터 보호 전략에 중요합니다. 이는 잘 설계된 환경을 구축하는 데 있어 중요한 단계입니다. 암호화되지 않은 기존 EBS 볼륨 또는 스냅샷을 암호화하는 직접적인 방법은 없지만 새 볼륨 또는 스냅샷을 생성하여 암호화할 수 있습니다. 자세한 내용은 Amazon EC2 설명서의 [EBS 리소스 암호화](#)를 참조하세요. 이 패턴은 신규 및 기존의 EBS 볼륨 모두를 암호화하기 위한 예방 및 감지 제어를 제공합니다. 이 패턴에서는 계정 설정을 구성하고, 자동화된 수정 프로세스를 만들고, 액세스 제어를 구현합니다.

사전 조건 및 제한 사항

사전 조건

- 활성 Amazon Web Services(AWS) 계정
- [AWS Command Line Interface\(AWS CLI\)](#), macOS, Linux 또는 Windows에 설치 및 구성됨
- [jq](#), macOS, Linux 또는 Windows에 설치 및 구성됨
- AWS ID 및 액세스 관리 (IAM) 권한은 AWS, 아마존 엘라스틱 컴퓨트 클라우드 (CloudFormationAmazon EC2), AWS 시스템 관리자, AWS Config 및 AWS KMS (키 관리 서비스)에 대한 읽기 및 쓰기 액세스 권한을 갖도록 프로비저닝됩니다.
- AWS Organizations는 모든 기능을 활성화하도록 구성되어 있으며, 이는 서비스 제어 정책의 요구 사항입니다.

- AWS Config는 대상 계정에서 활성화되어 있습니다.

제한 사항

- 대상 AWS 계정에는 암호화된 볼륨이라는 AWS Config 규칙이 없어야 합니다. 이 솔루션은 이 이름을 가진 규칙을 배포합니다. 이 이름을 가진 기존 규칙을 사용하면 배포가 실패하고, 동일한 규칙을 두 번 이상 처리하는 것과 관련된 불필요한 요금이 부과될 수 있습니다.
- 이 솔루션은 모든 EBS 볼륨을 동일한 AWS KMS 키로 암호화합니다.
- 계정에 대해 EBS 볼륨 암호화를 활성화하는 경우 이 설정은 리전별로 다릅니다. 특정 기능에 대해 이 기능을 활성화하면 AWS 리전의 개별 볼륨 또는 스냅샷에 대해 비활성화할 수 없습니다. 자세한 내용은 Amazon EC2 설명서의 [기본적으로 암호화](#)를 참조하세요.
- 암호화되지 않은 기존 EBS 볼륨을 수정할 때는 EC2 인스턴스를 사용하고 있지 않아야 합니다. 이 자동화는 암호화되지 않은 볼륨을 분리하고 암호화된 볼륨을 연결하기 위해 인스턴스를 종료합니다. 수정이 진행되는 동안에는 다운타임이 있습니다. 이것이 조직의 중요한 인프라인 경우, 인스턴스에서 실행되는 애플리케이션의 가용성에 영향을 미치지 않도록 [수동](#) 또는 [자동](#) 고가용성 구성을 마련해야 합니다. 중요한 리소스는 표준 유지 관리 기간에만 수정하는 것이 좋습니다.

아키텍처

자동화 워크플로

1. AWS Config는 암호화되지 않은 EBS 볼륨을 감지합니다.
2. 관리자는 AWS Config를 사용하여 Systems Manager에 수정 명령을 보냅니다.
3. Systems Manager 자동화는 암호화되지 않은 EBS 볼륨의 스냅샷을 생성합니다.
4. Systems Manager 자동화는 AWS KMS를 사용하여 스냅샷의 암호화된 사본을 생성합니다.
5. Systems Manager 자동화는 다음 작업을 수행합니다.
 - a. 영향을 받은 EC2 인스턴스가 실행 중인 경우 해당 인스턴스를 중지합니다.
 - b. 암호화된 새 볼륨 사본을 EC2 인스턴스에 연결합니다.
 - c. EC2 인스턴스를 원래 상태로 되돌립니다.

도구

서비스

- [AWS CLI](#) — AWS Command Line Interface(AWS CLI)는 AWS 서비스의 퍼블릭 애플리케이션 프로그래밍 인터페이스(API)에 직접 액세스할 수 있는 기능을 제공합니다. AWS CLI를 사용하여 서비스의 기능을 살펴보고 리소스를 관리할 셸 스크립트를 개발할 수 있습니다. 하위 수준의 API에 상응하는 명령에 더해 여러 AWS 서비스를 AWS CLI에 맞게 사용자 지정할 수 있습니다. 사용자 지정에는 복잡한 API와 서비스의 사용을 간소화하는 상위 수준 명령이 포함될 수 있습니다.
- [AWS CloudFormation](#) — CloudFormation AWS는 AWS 리소스를 모델링하고 설정하는 데 도움이 되는 서비스입니다. 원하는 모든 AWS 리소스 (예: Amazon EC2 인스턴스) 를 설명하는 템플릿을 만들고 해당 리소스를 CloudFormation 프로비저닝 및 구성합니다.
- [AWS Config](#) - AWS Config는 사용자의 AWS 계정에서 AWS 리소스의 구성을 상세하게 볼 수 있도록 합니다. 이러한 보기에는 리소스 간에 어떤 관계가 있는지와 리소스가 과거에 어떻게 구성되었는지도 포함되므로, 시간이 지나면서 구성과 관계가 어떻게 변하는지 확인할 수 있습니다.
- [Amazon EC2](#) – Amazon Elastic Compute Cloud(Amazon EC2)는 소프트웨어 시스템을 구축하고 호스팅하는 데 사용하는 크기 조절이 가능한 컴퓨팅 용량을 제공하는 웹 서비스입니다.
- [AWS KMS](#) – AWS Key Management Service(AWS KMS)는 클라우드에 맞게 규모를 조정된 암호화 및 키 관리 서비스입니다. AWS KMS 키와 기능은 다른 AWS 서비스에서 사용되며, 사용자는 이 KMS 키와 기능을 사용하여 AWS 환경에서 데이터를 보호할 수 있습니다.
- [AWS Organizations](#) – AWS Organizations는 사용자가 생성해 중앙 관리하는 단일 조직으로 여러 AWS 계정을 통합할 수 있는 계정 관리 서비스입니다.
- [AWS Systems Manager Automation](#) – Systems Manager Automation은 Amazon EC2 인스턴스와 기타 AWS 리소스의 일반적인 유지 관리 및 배포 작업을 간소화합니다.

기타 서비스

- [jq](#) — jq는 가볍고 유연한 명령줄 JSON 프로세서입니다. 이 도구를 사용하여 AWS CLI 출력에서 주요 정보를 추출합니다.

코드

- 이 패턴의 코드는 고객 KMS 키를 사용하여 [암호화되지 않은 EBS 볼륨 GitHub 자동 수정 리포지토리에서](#) 사용할 수 있습니다.

에픽

암호화되지 않은 볼륨의 자동 수정

작업	설명	필요한 기술
<p>스크립트와 템플릿을 다운로드 하십시오. CloudFormation</p>	<p>고객 KMS 키 리포지토리를 사용하여 암호화되지 않은 EBS 볼륨의 GitHub 자동 업데이트 적용에서 셸 스크립트, JSON 파일 및 CloudFormation 템플릿을 다운로드합니다.</p>	<p>AWS 관리자, 일반 AWS</p>
<p>AWS KMS 키의 관리자를 파악합니다.</p>	<ol style="list-style-type: none"> 1. AWS Management Console에 로그인하여 https://console.aws.amazon.com/iam/에서 IAM 콘솔을 엽니다. 2. AWS KMS 키 관리자가 될 사용자 또는 역할을 파악합니다. 이러한 목적으로 새 사용자 또는 역할을 생성해야 하는 경우 지금 생성하십시오. 자세한 내용은 IAM 설명서의 IAM 자격 증명을 참조하세요. 이 자동화는 새 AWS KMS 키를 생성합니다. 3. 파악이 완료되면 사용자 또는 역할의 Amazon 리소스 이름(ARN)을 복사합니다. 자세한 내용은 IAM 설명서의 IAM ARN을 참조하세요. 다음 단계에서 이 ARN을 사용합니다. 	<p>AWS 관리자, 일반 AWS</p>

작업	설명	필요한 기술
Stack1 템플릿을 배포하십시오. CloudFormation	<ol style="list-style-type: none"> 1. https://console.aws.amazon.com/cloudformation/ 에서 AWS CloudFormation 콘솔을 엽니다. 2. 에서 CloudFormation Stack1.yaml 템플릿을 배포합니다. 아래의 배포 세부 정보에 주의합니다. <ul style="list-style-type: none"> • 스택에 명확한 서술식 이름을 지정합니다. 이 값은 다음 단계에서 필요하므로 스택 이름을 적어 둡니다. • 키 관리자의 ARN을 Stack1의 유일한 매개변수 필드에 붙여넣습니다. 이 사용자 또는 역할은 스택에서 생성된 AWS KMS 키의 관리자가 됩니다. <p>CloudFormation 템플릿 배포에 대한 자세한 내용은 CloudFormation 설명서의 AWS CloudFormation 템플릿 사용을 참조하십시오.</p>	AWS 관리자, 일반 AWS

작업	설명	필요한 기술
<p>CloudFormation Stack2 템플릿을 배포하십시오.</p>	<p>에서 CloudFormation 템플릿을 배포합니다. Stack2.yaml 아래의 배포 세부 정보에 주의합니다.</p> <ul style="list-style-type: none"> • 스택에 명확한 서술식 이름을 지정합니다. • Stack2의 유일한 매개변수에는 이전 단계에서 생성한 스택의 이름을 입력합니다. 이를 통해 Stack2는 이전 단계에서 스택에 배포된 새 AWS KMS 키와 역할을 참조할 수 있습니다. 	<p>AWS 관리자, 일반 AWS</p>
<p>테스트용으로 암호화되지 않은 볼륨을 생성합니다.</p>	<p>암호화되지 않은 EBS 볼륨이 포함된 EC2 인스턴스를 생성합니다. 지침은 Amazon EC2 설명서의 Amazon EBS 볼륨 생성을 참고하세요. 인스턴스 유형은 중요하지 않으며 인스턴스에 액세스할 필요가 없습니다. t2.micro 인스턴스를 생성하여 프리 티어를 유지할 수 있으며, 키 쌍을 생성할 필요가 없습니다.</p>	<p>AWS 관리자, 일반 AWS</p>

작업	설명	필요한 기술
<p>AWS Config 규칙을 테스트합니다.</p>	<ol style="list-style-type: none"> 1. https://console.aws.amazon.com/config/에서 AWS KMS 콘솔을 엽니다. 규칙 페이지에서 암호화된 볼륨 규칙을 선택합니다. 2. 암호화되지 않은 새 테스트 인스턴스가 비준수 리소스 목록에 나타나는지 확인합니다. 볼륨이 즉시 표시되지 않는 경우, 몇 분 더 기다렸다가 결과를 새로 고칩니다. AWS Config 규칙은 인스턴스와 볼륨이 생성된 직후 리소스 변경을 감지합니다. 3. 리소스를 선택한 다음 해결을 선택합니다. <p>Systems Manager에서 다음과 같이 수정 진행 상황과 상태를 볼 수 있습니다.</p> <ol style="list-style-type: none"> 1. https://console.aws.amazon.com/systems-manager/에서 AWS Systems Manager 콘솔을 엽니다. 2. 왼쪽 탐색 창에서 Automation을 선택합니다. 3. 실행 ID 링크를 선택하면 단계와 상태를 볼 수 있습니다. 	<p>AWS 관리자, 일반 AWS</p>
<p>추가 계정 또는 AWS 리전을 구성합니다.</p>	<p>사용 사례에 따라 추가 계정 또는 AWS 리전에 대해 이 에픽을 반복합니다.</p>	<p>AWS 관리자, 일반 AWS</p>

EBS 볼륨의 계정 수준 암호화를 활성화합니다.

작업	설명	필요한 기술
<p>Enable 스크립트를 실행합니다.</p>	<ol style="list-style-type: none"> 1. bash 셸에서 cd 명령을 사용하여 복제된 리포지토리로 이동합니다. 2. 다음 명령을 입력하여 enable-ebs-encryption-for-account 스크립트를 실행합니다. <div style="border: 1px solid #ccc; border-radius: 10px; padding: 10px; margin: 10px 0;"> <pre>./Bash/enable-ebs-encryption-for-account.sh</pre> </div>	<p>AWS 관리자, 일반 AWS, bash</p>
<p>설정이 업데이트되었는지 확인합니다.</p>	<ol style="list-style-type: none"> 1. https://console.aws.amazon.com/ec2/에서 Amazon EC2 콘솔을 엽니다. 2. 화면 오른쪽의 설정에서 데이터 보호 및 보안을 선택합니다. 3. EBS 암호화 섹션에서 새 EBS 볼륨 상시 암호화가 켜져 있고 기본 암호화 키가 이전에 지정한 ARN으로 설정되어 있는지 확인합니다. <p>참고: 새 EBS 볼륨 항상 암호화 설정이 꺼져 있거나 키가 여전히 alias/aws/ebs로 설정되어 있는 경우, 셸 스크립트를 실행한 AWS 지역과 동일한 계정에 로그인했는</p>	<p>AWS 관리자, 일반 AWS</p>

작업	설명	필요한 기술
	지 확인하고 셸에서 오류 메시지를 확인하십시오.	
추가 계정 또는 AWS 리전을 구성합니다.	사용 사례에 따라 추가 계정 또는 AWS 리전에 대해 이 에픽을 반복합니다.	AWS 관리자, 일반 AWS

암호화되지 않은 인스턴스 생성 방지

작업	설명	필요한 기술
서비스 제어 정책을 생성합니다.	<ol style="list-style-type: none"> https://console.aws.amazon.com/organizations/v2/에서 AWS Organizations 콘솔을 엽니다. 새 서비스 제어 정책을 생성합니다. 자세한 내용은 AWS Organizations 설명서의 서비스 제어 정책 생성을 참고하십시오. DenyUnencryptedEC2.json 내용을 정책에 추가하고 저장합니다. 첫 번째 에픽의 리포지토리에서 이 JSON 파일을 다운로드했습니다. GitHub 이 정책을 조직 루트 또는 필요한 조직 유닛(OU)에 연결합니다. 지침은 AWS Organizations 설명서의 서비스 제어 정책 연결 및 분리를 참조하세요. 	AWS 관리자, 일반 AWS

관련 리소스

AWS 서비스 설명서

- [AWS CLI](#)
- [AWS Config](#)
- [AWS CloudFormation](#)
- [Amazon EC2](#)
- [AWS KMS](#)
- [AWS Organizations](#)
- [AWS Systems Manager Automation](#)

기타 리소스

- [jq 매뉴얼](#) (jq 웹사이트)
- [jq 다운로드](#) () GitHub

AWS Cloud의 Stromasys Charon-SSP 에뮬레이터에서 Sun SPARC 서버 백업하기

작성: Kevin Yung (AWS), Luis Ramos (Stromasys), and Rohit Darji (AWS)

환경: 프로덕션

기술: 스토리지 및 백업, 운영 체제, DevOps

워크로드: Oracle

AWS 서비스: Amazon EFS, Amazon S3, AWS Storage Gateway, AWS Systems Manager, Amazon EC2

요약

이 패턴은 온프레미스 환경에서 Amazon Web Services(AWS) 클라우드로 마이그레이션한 후 Sun Microsystems SPARC 서버를 백업하기 위한 네 가지 옵션을 제공합니다. 이러한 백업 옵션은 조직의 복구 시점 목표(RPO) 및 복구 시간 목표(RTO)를 충족하고, 자동화된 접근 방식을 사용하며, 전체 운영 비용을 절감하는 백업 계획을 구현하는 데 도움이 됩니다. 이 패턴은 네 가지 백업 옵션과 이를 구현하는 단계에 대한 개요를 제공합니다.

[Stromasys Charon-SSP 에뮬레이터](#)에서 게스트로 호스팅되는 Sun SPARC 서버를 사용하는 경우 다음 세 가지 백업 옵션 중 하나를 사용할 수 있습니다.

- 백업 옵션 1: Stromasys 가상 테이프- Charon-SSP 가상 테이프 기능을 사용하여 Sun SPARC 서버에 백업 시설을 설정하고 [AWS Systems Manager Automation](#)을 사용하여 백업 파일을 [Amazon Simple Storage Service\(Amazon S3\)](#) 및 [Amazon Simple Storage Service Glacier](#)에 아카이브합니다.
- 백업 옵션 2: Stromasys 스냅샷- Charon-SSP 스냅샷 기능을 사용하여 Charon-SSP에 있는 Sun SPARC 게스트 서버의 백업 기능을 설정합니다.
- 백업 옵션 3: Amazon Elastic Block Store(Amazon EBS) 볼륨 스냅샷- Charon-SSP 에뮬레이터를 Amazon Elastic Compute Cloud(Amazon EC2)에서 호스팅하는 경우, [Amazon EBS 볼륨 스냅샷](#)을 사용하여 Sun SPARC 파일 시스템의 백업을 생성할 수 있습니다.

하드웨어에서는 게스트, Amazon EC2에서는 Charon-SSP로 호스팅되는 Sun SPARC 서버를 사용하는 경우 다음 백업 옵션을 사용할 수 있습니다.

- 백업 옵션 4: AWS Storage Gateway 가상 테이프 라이브러리(VTL)- [Storage Gateway VTL 테이프 게이트웨이](#)와 함께 백업 애플리케이션을 사용하여 Sun SPARC 서버를 백업합니다.

Sun SPARC 서버에서 브랜딩된 영역으로 호스팅되는 Sun SPARC 서버를 사용하는 경우 백업 옵션 1, 2, 4를 사용할 수 있습니다.

[Stromasys](#)는 레거시 SPARC, Alpha, VAX 및 PA-RISC 중요 시스템을 에뮬레이션하기 위한 소프트웨어와 서비스를 제공합니다. Stromasys 에뮬레이션을 사용하여 AWS 클라우드로 마이그레이션하는 방법에 대한 자세한 내용은 AWS 블로그에서 [Stromasys를 사용하여 SPARC, Alpha 또는 기타 레거시 시스템을 AWS로 리호스팅하기](#)를 참고하십시오.

사전 조건 및 제한 사항

사전 조건

- 활성 상태의 AWS 계정.
- 기존 Sun SPARC 서버.
- Charon-SSP의 기존 라이선스. Charon-SSP 라이선스는 AWS Marketplace에서, Stromasys 가상 환경(VE) 라이선스는 Stromasys에서 구입할 수 있습니다. 자세한 내용은 [Stromasys 영업팀](#)에 문의하십시오.
- Sun SPARC 서버 및 Linux 백업에 익숙해야 합니다.
- Charon-SSP 에뮬레이션 기술에 익숙해야 합니다. 이에 대한 자세한 내용은 Stromasys 설명서의 [Stromasys 레거시 서버 에뮬레이션](#)을 참고하십시오.
- Sun SPARC 서버 파일 시스템에 가상 테이프 시설 또는 백업 애플리케이션을 사용하려면, Sun SPARC 서버 파일 시스템의 백업 시설을 만들고 구성해야 합니다.
- RPO 및 RTO에 대한 이해 이에 대한 자세한 내용은 AWS Well-Architected Framework 설명서의 [신뢰성 요소](#) 백서에서 [재해 복구 목표](#)를 참조하십시오.
- 백업 옵션 4를 사용하려면 다음이 있어야 합니다.
 - Storage Gateway VTL 테이프 게이트웨이를 지원하는 소프트웨어 기반 백업 애플리케이션입니다. 이에 대한 자세한 내용은 AWS Storage Gateway 설명서의 [VTL 디바이스 사용](#)을 참고하십시오.
 - Bacula Director 또는 유사한 백업 애플리케이션 (설치 및 구성) 이에 대한 자세한 내용은 [Bacula Director](#) 설명서를 참조하세요.

다음 표는 이 패턴의 네 가지 백업 옵션에 대한 정보를 제공합니다.

백업 옵션	충돌 일관성 달성?	애플리케이션 일관성 달성?	가상 백업 어플라이언스 솔루션?	일반적인 사용 사례
<p>옵션 1 — Stromasys 가상 테이프</p>	<p>예</p> <p>Sun SPARC 파일 시스템 스냅샷을 자동화하여 가상 테이프의 데이터를 백업할 수 있습니다. 예를 들어 UFS 또는 ZFS 스냅샷을 사용할 수 있습니다.</p>	<p>예</p> <p>이 백업 옵션을 사용하려면 인플라이트랜잭션을 플러시하거나, 파일 시스템 스냅샷 중에 읽기 전용 또는 임시 오프라인 모드를 구성하거나, 애플리케이션 데이터 덤프를 생성하는 자동화된 스크립트가 필요합니다. 애플리케이션 다운타임 또는 읽기 전용 모드가 필요할 수도 있습니다.</p>	<p>예</p>	<p>.tar 또는 .zip 파일을 사용한 Sun SPARC 서버 파일 시스템 백업</p> <p>애플리케이션 데이터 백업</p>
<p>옵션 2 — Stromasys 스냅샷</p>	<p>예</p> <p>이 기능을 활성화하려면 Charon-SSP Manager를 구성하거나 명령줄 시작 인수를 사용해야 합니다.</p> <p>또한, Linux 명령을 실행하여 Sun SPARC 게스트 서버 상태를 스냅</p>	<p>예</p> <p>이 백업 옵션은 가상 디스크와 메모리 덤프를 포함하여 에뮬레이팅된 게스트 서버의 스냅샷을 생성합니다.</p> <p>중요: 스냅샷 생성 중에는 Sun SPARC 게스트</p>	<p>아니요</p>	<p>Sun SPARC 서버 스냅샷</p> <p>애플리케이션 데이터 백업</p>

샷 파일에 저장하도록 Charon-SSP 에뮬레이터에 요청해야 합니다.

중요: Sun SPARC 게스트 서버를 종료해야 합니다.

옵션 3 — Amazon EBS 볼륨 스냅샷

예
AWS Backup을 사용하여 Amazon EBS 스냅샷을 자동화할 수 있습니다.

예
이 백업 옵션을 사용하려면 인플라이트 트랜잭션을 플러시하고 Amazon EBS 볼륨 스냅샷 중에 EC2 인스턴스를 읽기 전용 또는 일시적으로 중지하도록 구성하는 자동화된 스크립트가 필요합니다.

중요: 이 백업 옵션을 사용하려면 애플리케이션 일관성을 유지하기 위해 애플리케이션 다운타임 또는 읽기 전용 모드가 필요할 수 있습니다.

아니요

Sun SPARC 서버 파일 시스템 스냅샷
애플리케이션 데이터 백업

<p>옵션 4 — AWS Storage Gateway VTL</p>	<p>예</p> <p>백업 에이전트를 사용하여 Sun SPARC 파일 시스템 백업 데이터를 VTL에 자동으로 백업할 수 있습니다.</p>	<p>예</p> <p>이 백업 옵션을 사용하려면 사용 중에 트랜잭션을 플러시하고 파일 시스템 스냅샷 또는 애플리케이션 데이터 덤프 중에 읽기 전용 또는 임시 오프라인 모드를 구성하는 자동 스크립트가 필요합니다.</p>	<p>예</p> <p>대량의 Sun SPARC 서버 파일 시스템 백업 애플리케이션 데이터 백업</p> <p>중요: 이 백업 옵션을 사용하려면 애플리케이션 다운타임 또는 읽기 전용 모드가 필요할 수 있습니다.</p>
---------------------------------------	--	---	---

제한 사항

- 이 패턴의 접근 방식을 사용하여 개별 Sun SPARC 서버를 백업할 수 있지만, 클러스터에서 실행되는 애플리케이션이 있는 경우 공유 데이터에도 이러한 백업 옵션을 사용할 수 있습니다.

도구

백업 옵션 1 — Stromasys 가상 테이프

- [Stromasys Charon-SSP 에뮬레이터](#) - Charon-SSP 에뮬레이터는 표준 64-bit x86 호환 컴퓨터 시스템 내에 원본 SPARC 하드웨어의 가상 복제본을 생성합니다. SunOS 또는 Solaris와 같은 운영 체제(OS), 계층화된 제품 및 애플리케이션을 포함한 원본 SPARC 바이너리 코드를 실행합니다.
- [Amazon EC2](#) - Amazon Elastic Compute Cloud(Amazon EC2)는 소프트웨어 시스템을 구축하고 호스팅하는 데 사용하는 크기 조절이 가능한 컴퓨팅 용량을 제공하는 웹 서비스입니다.

- [Amazon EFS](#) — Amazon Elastic File System (Amazon EFS) 은 AWS 클라우드 서비스 및 온프레미스 리소스와 함께 사용할 수 있는 간단하고 set-and-forget 탄력적인 서버리스 파일 시스템을 제공합니다.
- [Amazon S3](#) – Amazon Simple Storage Service(S3)는 인터넷에 대한 스토리지입니다.
- [Amazon S3 Glacier](#) – Amazon Simple Storage Service Glacier는 데이터 보관 및 장기 백업을 위한 안전하고 안정적이며 극히 저렴한 Amazon S3 스토리지 클래스입니다.
- [AWS Systems Manager Automation](#) – 자동화는 AWS Systems Manager Manager의 기능으로서 EC2 인스턴스와 기타 AWS 리소스스의 일반적인 유지 관리 및 배포 작업을 간소화합니다.

백업 옵션 2: Stomasys 스냅샷

- [Stomasys Charon-SSP 에뮬레이터](#) - Charon-SSP 에뮬레이터는 표준 64-bit x86 호환 컴퓨터 시스템 내에 원본 SPARC 하드웨어의 가상 복제본을 생성합니다. SunOS 또는 Solaris와 같은 OS, 계층화된 제품, 애플리케이션을 포함한 원본 SPARC 바이너리 코드를 실행합니다.
- [Amazon EC2](#) – Amazon Elastic Compute Cloud(Amazon EC2)는 소프트웨어 시스템을 구축하고 호스팅하는 데 사용하는 크기 조절이 가능한 컴퓨팅 용량을 제공하는 웹 서비스입니다.
- [Amazon EFS](#) — Amazon Elastic File System (Amazon EFS) 은 AWS 클라우드 서비스 및 온프레미스 리소스와 함께 사용할 수 있는 간단하고 set-and-forget 탄력적인 서버리스 파일 시스템을 제공합니다.
- [Amazon S3](#) – Amazon Simple Storage Service(S3)는 인터넷에 대한 스토리지입니다.
- [Amazon S3 Glacier](#) – Amazon Simple Storage Service Glacier는 데이터 보관 및 장기 백업을 위한 안전하고 안정적이며 극히 저렴한 Amazon S3 스토리지 클래스입니다.
- [AWS Systems Manager Automation](#) – 자동화는 AWS Systems Manager Manager의 기능으로서 EC2 인스턴스와 기타 AWS 리소스스의 일반적인 유지 관리 및 배포 작업을 간소화합니다.

백업 옵션 3: Amazon EBS 볼륨 스냅샷

- [Stomasys Charon-SSP 에뮬레이터](#) - Charon-SSP 에뮬레이터는 표준 64-bit x86 호환 컴퓨터 시스템 내에 원본 SPARC 하드웨어의 가상 복제본을 생성합니다. SunOS 또는 Solaris와 같은 OS, 계층화된 제품, 애플리케이션을 포함한 원본 SPARC 바이너리 코드를 실행합니다.
- [AWS Backup](#) — AWS Backup은 AWS 서비스, 클라우드, 온프레미스 전반적으로 손쉽게 중앙 집중화하고 자동화할 수 있게 해주는 완전 관리형 데이터 보호 서비스입니다.

- [Amazon EBS](#) – Amazon Elastic Block Store(Amazon EBS)는 EC2 인스턴스에 사용할 수 있는 블록 스토리지 볼륨을 제공합니다.
- [Amazon EC2](#) – Amazon Elastic Compute Cloud(Amazon EC2)는 소프트웨어 시스템을 구축하고 호스팅하는 데 사용하는 크기 조절이 가능한 컴퓨팅 용량을 제공하는 웹 서비스입니다.

백업 옵션 4: AWS Storage Gateway VTL

- [Stromasys Charon-SSP 에뮬레이터](#) - Charon-SSP 에뮬레이터는 표준 64-bit x86 호환 컴퓨터 시스템 내에 원본 SPARC 하드웨어의 가상 복제본을 생성합니다. SunOS 또는 Solaris와 같은 OS, 계층화된 제품, 애플리케이션을 포함한 원본 SPARC 바이너리 코드를 실행합니다.
- [Bacula](#) – Bacula는 오픈소스 엔터프라이즈급 컴퓨터 백업 시스템입니다. 기존 백업 애플리케이션이 테이프 게이트웨이를 지원하는지 여부에 대한 자세한 내용은 AWS Storage Gateway 설명서에서 [테이프 게이트웨이에 지원되는 타사 백업 애플리케이션](#)을 참고하십시오.
- [Amazon EC2](#) – Amazon Elastic Compute Cloud(Amazon EC2)는 소프트웨어 시스템을 구축하고 호스팅하는 데 사용하는 크기 조절이 가능한 컴퓨팅 용량을 제공하는 웹 서비스입니다.
- [Amazon RDS for MySQL](#) - Amazon Relational Database Service(Amazon RDS)는 여러 MySQL 버전을 실행하는 DB 인스턴스를 지원합니다.
- [Amazon S3](#) – Amazon Simple Storage Service(S3)는 인터넷에 대한 스토리지입니다.
- [Amazon S3 Glacier](#) – Amazon Simple Storage Service Glacier는 데이터 보관 및 장기 백업을 위한 안전하고 안정적이며 극히 저렴한 Amazon S3 스토리지 클래스입니다.
- [AWS Storage Gateway](#) – Storage Gateway는 온프레미스 소프트웨어 어플라이언스를 클라우드 기반 스토리지에 연결하여 데이터 보안 기능으로 온프레미스 IT 환경과 AWS 스토리지 인프라 사이에 원활한 통합이 이루어지도록 지원합니다.

에픽

백업 옵션 1 — Stromasys 가상 테이프

작업	설명	필요한 기술
가상 테이프 파일 스토리지용 Amazon EFS 공유 파일 시스템을 생성합니다.	AWS Management Console에 로그인하거나 AWS CLI를 사용	클라우드 아키텍트

작업	설명	필요한 기술
	<p>하여 Amazon EFS 파일 시스템을 생성합니다.</p> <p>이에 대한 자세한 내용은 Amazon EFS 설명서의 Amazon EFS 파일 시스템 생성을 참고하십시오.</p>	
<p>공유 파일 시스템을 마운트하도록 Linux 호스트를 구성합니다.</p>	<p>Amazon EC2 Linux 인스턴스에 Amazon EFS 드라이버를 설치하고 시작 중에 Amazon EFS 공유 파일 시스템을 마운트하도록 Linux OS를 구성합니다.</p> <p>이에 대한 자세한 내용은 Amazon EFS 설명서의 EFS 마운트 도우미를 사용한 파일 시스템 마운트를 참고하십시오.</p>	DevOps 엔지니어
<p>Charon-SSP 에뮬레이터를 설치합니다.</p>	<p>Amazon EC2 Linux 인스턴스에 Charon-SSP 에뮬레이터를 설치합니다.</p> <p>이에 대한 자세한 내용은 Stomasys 설명서의 Charon-SSP용 AWS 클라우드 인스턴스 설정을 참고하십시오.</p>	DevOps 엔지니어
<p>각 Sun SPARC 게스트 서버의 공유 파일 시스템에 가상 테이프 파일 컨테이너를 생성합니다.</p>	<p><code>touch <vtape-container-name></code> 명령을 실행하여 Charon-SSP 에뮬레이터에 배포된 각 Sun SPARC 게스트 서버의 공유 파일 시스템에 가상 테이프 파일 컨테이너를 생성합니다.</p>	DevOps 엔지니어

작업	설명	필요한 기술
<p>Sun SPARC 게스트 서버용 가상 테이프 디바이스를 생성하도록 Charon-SSP 관리자를 구성합니다.</p>	<p>Charon-SSP 관리자에 로그인하여 가상 테이프 장치를 만들고 각 Sun SPARC 게스트 서버의 가상 테이프 컨테이너 파일을 사용하도록 구성합니다.</p> <p>이에 대한 자세한 내용은 Stromasys 설명서의 Linux용 Charon-SSP 5.2 사용 설명서를 참고하십시오.</p>	DevOps 엔지니어
<p>Sun SPARC 게스트 서버에서 가상 테이프 장치를 사용할 수 있는지 확인합니다.</p>	<p>각 Sun SPARC 게스트 서버에 로그인하고 <code>mt -f /dev/rmt/1</code> 명령을 실행하여 가상 테이프 장치가 OS에 구성되어 있는지 확인합니다.</p>	DevOps 엔지니어
<p>Systems Manager Automation 런북 및 자동화를 개발합니다.</p>	<p>Systems Manager Automation 런북을 개발하고 Systems Manager에서 유지 관리 기간과 연결을 설정하여 백업 프로세스를 예약합니다.</p> <p>이에 대한 자세한 내용은 AWS Systems Manager 설명서의 자동화 안내 및 유지 관리 기간 설정을 참고하십시오.</p>	클라우드 아키텍트
<p>순환된 가상 테이프 컨테이너 파일을 보관하도록 Systems Manager Automation을 구성합니다.</p>	<p>추가 정보 섹션의 뒤로 옵션 1에 있는 코드 샘플을 사용하여 순환된 가상 테이프 컨테이너 파일을 Amazon S3 및 Amazon S3 Glacier에 보관하는 Systems Manager Automation 런북을 개발할 수 있습니다.</p>	클라우드 아키텍트

작업	설명	필요한 기술
보관 및 예약을 위해 Systems Manager Automation 런북을 배포합니다.	Systems Manager Automation 런북을 배포하고 Systems Manager에서 자동으로 실행되도록 예약합니다. 이에 대한 자세한 내용은 Systems Manager 설명서의 자동화 안내 를 참조하세요.	클라우드 아키텍트

백업 옵션 2 — Strosasys 스냅샷 생성

작업	설명	필요한 기술
가상 테이프 파일 스토리지용 Amazon EFS 공유 파일 시스템을 생성합니다.	AWS Management Console에 로그인하거나 AWS CLI를 사용하여 Amazon EFS 파일 시스템을 생성합니다. 이에 대한 자세한 내용은 Amazon EFS 설명서의 Amazon EFS 파일 시스템 생성 을 참고하십시오.	클라우드 아키텍트
공유 파일 시스템을 마운트하도록 Linux 호스트를 구성합니다.	Amazon EC2 Linux 인스턴스에 Amazon EFS 드라이버를 설치하고 시작 중에 Amazon EFS 공유 파일 시스템을 마운트하도록 Linux OS를 구성합니다. 이에 대한 자세한 내용은 Amazon EFS 설명서의 EFS 마운트 도우미를 사용한 파일 시스템 마운트 를 참고하십시오.	DevOps 엔지니어

작업	설명	필요한 기술
<p>Charon-SSP 에뮬레이터를 설치합니다.</p>	<p>Amazon EC2 Linux 인스턴스에 Charon-SSP 에뮬레이터를 설치합니다.</p> <p>이에 대한 자세한 내용은 Stromasys 설명서의 Charon-SSP용 AWS 클라우드 인스턴스 설정을 참고하십시오.</p>	<p>DevOps 엔지니어</p>
<p>스냅샷 옵션으로 시작하도록 Sun SPARC 게스트 서버를 구성합니다.</p>	<p>Charon-SSP 관리자를 사용하여 각 Sun SPARC 게스트 서버의 스냅샷 옵션을 설정합니다.</p> <p>이에 대한 자세한 내용은 Stromasys 설명서의 Linux용 Charon-SSP 5.2 사용 설명서를 참고하십시오.</p>	<p>DevOps 엔지니어</p>
<p>Systems Manager Automation 런북을 개발합니다.</p>	<p>추가 정보 섹션의 백업 옵션 2에 있는 코드 샘플을 사용하여 유지 관리 기간 중에 Sun SPARC 게스트 서버에서 스냅샷 명령을 원격으로 실행하는 Systems Manager Automation 런북을 개발할 수 있습니다.</p>	<p>클라우드 아키텍트</p>
<p>Systems Manager Automation 런북을 배포하고 Amazon EC2 Linux 호스트와의 연결을 설정합니다.</p>	<p>Systems Manager Automation 런북을 배포하고 Systems Manager에서 유지 관리 기간과 연결을 설정하여 백업 프로세스를 예약합니다.</p> <p>이에 대한 자세한 내용은 AWS Systems Manager 설명서의 자동화 안내 및 유지 관리 기간 설정을 참조하세요.</p>	<p>클라우드 아키텍트</p>

작업	설명	필요한 기술
스냅샷을 장기 스토리지에 보관합니다.	추가 정보 섹션의 런북 샘플 코드를 사용하여 스냅샷 파일을 Amazon S3 및 Amazon S3 Glacier에 보관하는 Systems Manager Automation 런북을 개발해 보세요.	클라우드 아키텍트

백업 옵션 3 – Amazon EBS 볼륨 스냅샷 생성

작업	설명	필요한 기술
Charon-SSP 에뮬레이터를 설치합니다.	Amazon EC2 Linux 인스턴스에 Charon-SSP 에뮬레이터를 설치합니다. 이에 대한 자세한 내용은 Stomasys 설명서의 Charon-SSP용 AWS 클라우드 인스턴스 설정 을 참고하십시오.	DevOps 엔지니어
Sun SPRAC 게스트 서버용 EBS 볼륨을 생성합니다.	AWS 관리 콘솔에 로그인하고 Amazon EBS 콘솔을 연 다음 Sun SPRAC 게스트 서버용 EBS 볼륨을 생성합니다. 이에 대한 자세한 내용은 Stomasys 설명서의 Charon-SSP용 AWS 클라우드 인스턴스 설정 을 참고하십시오.	클라우드 아키텍트
EBS 볼륨을 Amazon EC2 Linux 인스턴스에 연결합니다.	Amazon EC2 콘솔에서 EBS 볼륨을 새 인스턴스에 다시 연결합니다. 이에 대한 자세한 내용은 Amazon EC2 설명서의 인스턴스	AWS DevOps

작업	설명	필요한 기술
	<p>스에 Amazon EBS 볼륨 연결하기를 참조하세요.</p>	
<p>Charon-SSP 에뮬레이터에서 EBS 볼륨을 SCSI 드라이브로 매핑합니다.</p>	<p>EBS 볼륨을 Sun SPARC 게스트 서버의 SCSI 드라이브로 매핑하도록 Charon-SSP 관리자를 구성합니다.</p> <p>이에 대한 자세한 내용은 Stromasys 설명서의 Linux용 Charon-SSP V5.2 가이드의 SCSI 스토리지 구성 단원을 참조하세요.</p>	<p>AWS DevOps</p>
<p>EBS 볼륨 스냅샷 생성을 위한 AWS Backup 일정을 구성합니다.</p>	<p>EBS 볼륨의 스냅샷을 생성하기 위한 AWS Backup 정책 및 일정을 설정합니다.</p> <p>이에 대한 자세한 내용은 AWS 개발자 센터 설명서의 AWS Backup을 사용한 Amazon EBS 백업 및 복원 자습서를 참고하십시오.</p>	<p>AWS DevOps</p>

백업 옵션 4 - AWS Storage Gateway VTL 생성

작업	설명	필요한 기술
<p>테이프 게이트웨이 디바이스를 생성합니다.</p>	<p>AWS 관리 콘솔에 로그인하고, AWS Storage Gateway 콘솔을 연 다음, VPC에서 테이프 게이트웨이 디바이스를 생성합니다.</p> <p>이에 대한 자세한 내용은 AWS Storage Gateway 설명서의 게</p>	<p>클라우드 아키텍트</p>

작업	설명	필요한 기술
	<p>이트웨이 생성하기를 참고하십시오.</p>	
<p>Bacula 카탈로그를 위한 Amazon RDS DB 인스턴스를 생성합니다.</p>	<p>Amazon RDS 콘솔을 열고 Amazon RDS for MySQL DB 인스턴스를 생성합니다.</p> <p>이에 대한 자세한 내용은 Amazon RDS 설명서에서 MySQL DB 인스턴스 생성 및 MySQL DB 인스턴스의 데이터베이스에 연결을 참고하십시오.</p>	클라우드 아키텍트
<p>VPC에 백업 애플리케이션 컨트롤러를 배포합니다.</p>	<p>EC2 인스턴스에 Bacula를 설치하고 백업 애플리케이션 컨트롤러를 배포한 다음 테이프 게이트웨이 디바이스와 연결 되도록 백업 스토리지를 구성합니다. Bacula-storage-daemon-config.txt 파일(첨부됨)에서 샘플 Bacula Director 스토리지 데몬 구성을 사용할 수 있습니다.</p> <p>이에 대한 자세한 내용은 Bacula 설명서를 참조하세요.</p>	AWS DevOps
<p>Sun SPARC 게스트 서버에 백업 애플리케이션을 설정합니다.</p>	<p>SUN-SPARC-Guest-Bacula-Config.txt 파일(첨부됨)의 샘플 Bacula 구성을 사용하여 Sun SPARC 게스트 서버에 백업 애플리케이션을 설치하고 설정하는 두 번째 클라이언트를 설정합니다.</p>	DevOps 엔지니어

작업	설명	필요한 기술
백업 구성 및 일정을 설정합니다.	<p>Bacula-Directory-C onfig.txt 파일(첨부됨)에 있는 샘플 Bacula Director 구성을 사용하여 백업 애플리케이션 컨트롤러에서 백업 구성 및 일정을 설정합니다.</p> <p>이에 대한 자세한 내용은 Bacula 설명서를 참조하세요.</p>	DevOps 엔지니어
백업 구성 및 예약이 올바른지 확인하십시오.	<p>Bacula 설명서의 지침에 따라 Sun SPARC 게스트 서버의 설정에 대한 검증 및 백업 테스트를 수행합니다.</p> <p>예를 들어, 다음 명령을 사용하여 구성 파일의 유효성을 검사할 수 있습니다.</p> <ul style="list-style-type: none"> • bacula-dir -t -c bacula-dir.conf • bacula-fd -t -c bacula-fd.conf • bacula-sd -t -c bacula-sd.conf 	DevOps 엔지니어

관련 리소스

- [VE 라이선스가 포함된 Charon virtual SPARC](#)
- [Charon virtual SPARC](#)
- [Bacula Enterprise Edition을 통한 클라우드 서비스 및 객체 스토리지 사용](#)
- [재해 복구\(DR\) 목표](#)
- [Charon 레거시 시스템 에뮬레이션 솔루션](#)

추가 정보

백업 옵션 1 — Stromasys 가상 테이프 생성

다음 샘플 Systems Manager Automation 런북 코드를 사용하여 백업을 자동으로 시작한 다음 테이프를 교체할 수 있습니다.

```
...
# example backup script saved in SUN SPARC Server
#!/usr/bin/bash
mt -f rewind
tar -cvf
mt -f offline
...
    mainSteps:
    - action: aws:runShellScript
      name:
      inputs:
        onFailure: Abort
        timeoutSeconds: "1200"
        runCommand:
        - |
          # Validate tape backup container file exists
          if [ ! -f {{TapeBackupContainerFile}} ]; then
            logger -s -p local3.warning "Tape backup container file is not exists
- {{TapeBackupContainerFile}}, create a new one"
            touch {{TapeBackupContainerFile}}
          fi
    - action: aws:runShellScript
      name: startBackup
      inputs:
        onFailure: Abort
        timeoutSeconds: "1200"
        runCommand:
        - |
          user={{BACKUP_USER}}
          keypair={{KEYPAIR_PATH}}
          server={{SUN_SPARC_IP}}
          backup_script={{BACKUP_SCRIPT}}
          ssh -i $keypair $user@$server -c "/usr/bin/bash $backup_script"
    - action: aws:runShellScript
      name: swapVirtualDiskContainer
      inputs:
```

```

    onFailure: Abort
    timeoutSeconds: "1200"
    runCommand:
    - |
      mv {{TapeBackupContainerFile}} {{TapeBackupContainerFile}}.$(date +%s)
      touch {{TapeBackupContainerFile}}
- action: aws:runShellScript
  name: uploadBackupArchiveToS3
  inputs:
    onFailure: Abort
    timeoutSeconds: "1200"
    runCommand:
    - |
      aws s3 cp {{TapeBackupContainerFile}} s3://{{BACKUP_BUCKET}}/
      {{SUN_SPARC_IP}}/$(date '+%Y-%m-%d')/
  ...

```

백업 옵션 2 — Stomasys 스냅샷

다음 샘플 Systems Manager Automation 런북 코드를 사용하여 백업 프로세스를 자동화할 수 있습니다.

```

...

mainSteps:
- action: aws:runShellScript
  name: startSnapshot
  inputs:
    onFailure: Abort
    timeoutSeconds: "1200"
    runCommand:
    - |
      # You may consider some graceful stop of the application before taking a
      snapshot

      # Query SSP PID by configuration file
      # Example: ps ax | grep ssp-4 | grep Solaris10.cfg | awk '{print $1"
"$5}' | grep ssp4 | cut -f1 -d" "
      pid=`ps ax | grep ssp-4 | grep {{SSP_GUEST_CONFIG_FILE}} | awk '{print
$1" "$5}' | grep ssp4 | cut -f1 -d" "`
      if [ -n "${pid}" ]; then
        kill -SIGTSTP ${pid}
      else

```

```

        echo "No PID found for SPARC guest with config
{{SSP_GUEST_CONFIG_FILE}}"
        exit 1
    fi
- action: aws:runShellScript
  name: startBackup
  inputs:
    onFailure: Abort
    timeoutSeconds: "1200"
    runCommand:
      - |
        # upload snapshot and virtual disk files into S3
        aws s3 sync {{SNAPSHOT_FOLDER}} s3://{{BACKUP_BUCKET}}/$(date '+%Y-%m-%d')/
        aws s3 cp {{VIRTUAL_DISK_FILE}} s3://{{BACKUP_BUCKET}}/$(date '+%Y-%m-%d')/
- action: aws:runShellScript
  name: restratSPARCGuest
  inputs:
    onFailure: Abort
    timeoutSeconds: "1200"
    runCommand:
      - |
        /opt/charon-ssp/ssp-4u/ssp4u -f {{SSP_GUEST_CONFIG_FILE}} -d -a
        {{SPARC_GUEST_NAME}} --snapshot {{SNAPSHOT_FOLDER}}
    ...

```

백업 옵션 4: AWS Storage Gateway VTL

Solaris 비글로벌 영역을 사용하여 가상화된 레거시 Sun SPARC 서버를 실행하는 경우, 백업 애플리케이션 접근 방식을 Sun SPARC 서버에서 실행되는 비글로벌 영역에 적용할 수 있습니다(예: 백업 클라이언트는 비글로벌 영역 내에서 실행될 수 있음). 하지만 백업 클라이언트는 Solaris 호스트에서 실행되어 비글로벌 영역의 스냅샷을 찍을 수도 있습니다. 그러면 스냅샷을 테이프에 백업할 수 있습니다.

다음 샘플 구성은 Solaris 비글로벌 영역을 호스트하는 파일 시스템을 Solaris 호스트의 백업 구성에 추가합니다.

```

FileSet {
  Name = "Branded Zones"
  Include {
    Options {
      signature = MD5
    }
  }
}

```

```
    }  
    File = /zones  
  }  
}
```

첨부

이 문서와 관련된 추가 콘텐츠에 액세스하려면 [attachment.zip](#) 파일의 압축을 풉니다.

Veeam Backup & Replication을 사용하여 Amazon S3에 데이터를 백업 및 보관

작성자: Jeanna James(AWS), Anthony Fiore(AWS), William Quigley

환경: 프로덕션

기술: 스토리지 및 백업

AWS 서비스: Amazon EC2;
Amazon S3; Amazon S3
Glacier

요약

이 패턴은 Veeam 스케일 아웃 백업 리포지토리 기능을 사용하여 Veeam Backup & Replication에서 생성한 백업을 지원되는 Amazon Simple Storage Service(S3) 객체 스토리지 클래스로 보내는 프로세스를 자세히 설명합니다.

Veeam은 특정 요구 사항에 적합한 여러 Amazon S3 스토리지 클래스를 지원합니다. 백업 또는 아카이브 데이터의 데이터 액세스, 복원력, 비용 요구 사항에 따라 스토리지 유형을 선택할 수 있습니다. 예를 들어, 30일 이상 사용할 계획이 없는 데이터를 Amazon S3 Inrequent Access(IA)에 저렴한 비용으로 저장할 수 있습니다. 90일 이상 데이터를 보관할 계획이라면 Veeam의 아카이브 계층과 함께 Amazon Simple Storage Service Glacier(Amazon S3 Glacier) Flexible Retrieval 또는 S3 Glacier Deep Archive를 사용할 수 있습니다. 또한 S3 객체 잠금을 사용하여 Amazon S3 내에서 백업을 변경 불가능하게 만들 수 있습니다.

이 패턴은 AWS Storage Gateway에서 Tape Gateway를 사용하여 Veeam Backup & Replication을 설정하는 방법을 다루지 않습니다. 해당 주제에 대한 자세한 내용은 Veeam 웹 사이트의 [AWS VTL Gateway를 사용한 Veeam Backup & Replication-배포 가이드](#)를 참조하세요.

경고: 이 시나리오에서는 IAM 사용자에게 프로그래밍 액세스 권한 및 장기 자격 증명이 필요하며, 이는 보안 위험을 초래합니다. 이러한 위험을 줄이려면 이러한 사용자에게 작업을 수행하는 데 필요한 권한만 제공하고 더 이상 필요하지 않을 때는 해당 사용자를 제거하는 것이 좋습니다. 필요한 경우 액세스 키를 업데이트할 수 있습니다. 자세한 내용은 IAM 사용 설명서의 [액세스 키 업데이트](#)를 참조하십시오.

사전 조건 및 제한 사항

사전 조건

- Veeam Availability Suite 또는 Veeam Backup Essentials를 포함한 Veeam Backup & Replication 설치([무료 평가판](#)에 등록할 수 있음)
- Enterprise 또는 Enterprise Plus 기능이 포함된 Veeam Backup & Replication 라이선스, Veeam Universal License(VUL) 포함
- Amazon S3 버킷에 액세스할 수 있는 활성 AWS Identity and Access Management(IAM) 사용자
- Amazon Elastic Compute Cloud(Amazon EC2) 및 Amazon Virtual Private Cloud(VPC)에 액세스할 수 있는 활성 IAM 사용자(아카이브 계층을 사용하는 경우)
- 퍼블릭 인터넷 연결 또는 AWS Direct Connect 퍼블릭 가상 인터페이스(VIF)를 통해 트래픽을 백업 및 복원하기 위한 가용 대역폭을 갖춘 온프레미스에서 AWS 서비스로의 네트워크 연결
- 객체 스토리지 리포지토리와 적절한 통신을 보장하기 위해 다음과 같은 네트워크 포트 및 엔드포인트가 열려있습니다.
 - Amazon S3 스토리지-TCP-포트 443: Amazon S3 스토리지와의 통신하는 데 사용됩니다.
 - Amazon S3 스토리지 — 클라우드 엔드포인트 — AWS 지역 및 GovCloud AWS (미국) 지역의 경우 *.amazonaws.com, 중국 지역의 경우 *.amazonaws.com.cn: Amazon S3 스토리지와 통신하는 데 사용됩니다. 연결 엔드포인트의 전체 목록은 AWS 설명서의 [Amazon S3 엔드포인트](#)를 참조하세요.
 - Amazon S3 스토리지-TCP HTTP-포트 80: 인증서 상태를 확인하는 데 사용됩니다. 인증서 취소 목록(CRL) URL 및 온라인 인증서 상태 프로토콜(OCSP) 서버와 같은 인증서 확인 엔드포인트는 변경될 수 있습니다. 실제 주소 목록은 인증서 자체에서 찾을 수 있습니다.
 - Amazon S3 스토리지-인증서 확인 엔드포인트-*.amazontrust.com: 인증서 상태를 확인하는 데 사용됩니다. 인증서 확인 엔드포인트(CRL URL 및 OCSP 서버)는 변경될 수 있습니다. 실제 주소 목록은 인증서 자체에서 찾을 수 있습니다.

제한 사항

- Veeam은 Veeam 객체 스토리지 리포지토리로 사용되는 S3 버킷에 대한 S3 수명 주기 정책을 지원하지 않습니다. Amazon S3 스토리지 클래스 전환 정책과 S3 수명 주기 만료 규칙이 포함됩니다. Veeam은 이러한 객체를 관리하는 유일한 개체여야 합니다. S3 수명 주기 정책을 활성화하면 데이터 손실을 비롯한 예상치 못한 결과가 발생할 수 있습니다.

제품 버전

- Veeam Backup & Replication v9.5 업데이트 4 이상(백업 전용 또는 용량 계층)
- Veeam Backup & Replication v10 이상(백업 또는 용량 계층 및 S3 객체 잠금)
- Veeam Backup & Replication v11 이상(백업 또는 용량 계층, 아카이브 또는 아카이브 계층, S3 객체 잠금)
- Veeam Backup & Replication v12 이상(성능 계층, 백업 또는 용량 계층, 아카이브 또는 아카이브 계층, S3 객체 잠금)
- S3 Standard
- S3 Standard-IA
- S3 One Zone-IA
- S3 Glacier Flexible Retrieval(v11 이상만 해당)
- S3 Glacier Deep Archive(v11 이상만 해당)
- S3 Glacier Instant Retrieval(v12 이상만 해당)

아키텍처

소스 기술 스택

- Veeam 백업 서버 또는 Veeam 게이트웨이 서버에서 Amazon S3로 연결되는 온프레미스 Veeam Backup & Replication 설치

대상 기술 스택

- Amazon S3
- Amazon VPC 및 Amazon EC2(아카이브 계층을 사용할 경우)

대상 아키텍처: SOBR

다음 다이어그램은 스케일 아웃 백업 리포지토리(SOBR) 아키텍처를 보여줍니다.

Veeam Backup and Replication 소프트웨어는 시스템 장애, 애플리케이션 오류 또는 실수로 인한 삭제와 같은 논리적 오류로부터 데이터를 보호합니다. 이 다이어그램에서는 백업이 온프레미스에서 먼저 실행되고 보조 사본이 Amazon S3로 직접 전송됩니다. 백업은 데이터 사본을 나타냅니다. point-in-time

워크플로는 백업을 계층화하거나 Amazon S3로 복사하는 데 필요한 세 가지 기본 구성 요소와 하나의 선택적 구성 요소로 구성됩니다.

- Veeam Backup & Replication(1)-백업 인프라, 설정, 작업, 복구 작업 및 기타 프로세스를 조정, 제어, 관리하는 백업 서버입니다.
- Veeam 게이트웨이 서버(다이어그램에는 표시되지 않음)-Veeam 백업 서버가 Amazon S3에 대한 아웃바운드 연결이 없는 경우 필요한 선택적 온프레미스 게이트웨이 서버입니다.
- 스케일 아웃 백업 리포지토리(2)-수평 확장으로 데이터의 다중 계층 스토리지를 지원하는 리포지토리 시스템입니다. 스케일 아웃 백업 리포지토리는 데이터에 빠르게 액세스할 수 있는 하나 이상의 백업 리포지토리로 구성되며 장기 스토리지(용량 계층) 및 보관(아카이브 계층)을 위해 Amazon S3 객체 스토리지 리포지토리로 확장할 수 있습니다. Veeam은 스케일 아웃 백업 리포지토리를 사용하여 로컬(성능 계층) 과 Amazon S3 객체 스토리지(용량 및 아카이브 계층) 간에 데이터를 자동으로 계층화합니다.
- Amazon S3(3)-확장성, 데이터 가용성, 보안, 성능을 제공하는 AWS 객체 스토리지 서비스입니다.

대상 아키텍처: DTO

다음 다이어그램은 direct-to-object (DTO) 아키텍처를 보여줍니다.

이 다이어그램에서 백업 데이터는 온프레미스에 먼저 저장되지 않고 Amazon S3로 직접 이동합니다. 보조 사본은 S3 Glacier에 저장할 수 있습니다.

자동화 및 규모 조정

[리포지토리에 제공된 AWS CloudFormation 템플릿을 사용하여 IAM 리소스 및 S3 버킷 생성을 자동화할 수 있습니다. VeeamHub GitHub](#) 템플릿에는 표준 옵션과 변경 불가 옵션이 둘 다 포함되어 있습니다.

도구

도구 및 AWS 서비스

- [Veeam Backup & Replication](#)은 가상 및 물리적 워크로드를 보호, 백업, 복제, 복원하는 Veeam의 솔루션입니다.
- [AWS는 AWS CloudFormation](#) 리소스를 모델링 및 설정하고, 빠르고 일관되게 프로비저닝하고, 수명 주기 전반에 걸쳐 관리할 수 있도록 지원합니다. 템플릿을 사용하여 리소스와 해당 종속성을 설명하

고, 리소스를 개별적으로 관리하는 대신 스택으로 함께 시작하고 구성할 수 있습니다. 여러 AWS 계정 및 AWS 리전에서 스택을 관리하고 프로비저닝할 수 있습니다.

- [Amazon Elastic Compute Cloud\(Amazon EC2\)](#)는 AWS 클라우드에서 확장 가능한 컴퓨팅 용량을 제공합니다. Amazon EC2를 사용하여 필요에 따라 많거나 적은 수의 가상 서버를 시작하고 스케일 아웃 또는 스케일 인할 수 있습니다.
- [AWS Identity and Access Management\(IAM\)](#)는 AWS 서비스에 대한 액세스를 안전하게 제어하는 웹 서비스입니다. IAM을 통해 사용자, 액세스 키와 같은 보안 인증 정보, 사용자와 애플리케이션이 액세스할 수 있는 AWS 리소스를 제어하는 권한을 한 곳에서 관리할 수 있습니다.
- [Amazon Simple Storage Service\(S3\)](#)는 객체 스토리지 서비스입니다. Amazon S3를 사용하면 인터넷을 통해 언제 어디서든 원하는 양의 데이터를 저장하고 검색할 수 있습니다.
- [Amazon S3 Glacier\(S3 Glacier\)](#)는 저렴한 비용의 데이터 보관 및 장기 백업을 위한 안전하고 내구성이 뛰어난 서비스입니다.
- [Amazon Virtual Private Cloud\(VPC\)](#)는 정의한 가상 네트워크에서 AWS 리소스를 시작할 수 있도록 AWS 클라우드의 논리적으로 격리된 섹션을 프로비저닝합니다. 이 가상 네트워크는 AWS의 확장 가능한 인프라를 사용한다는 이점과 함께 고객의 자체 데이터 센터에서 운영하는 기존 네트워크와 매우 유사합니다.

code

[VeeamHub GitHub 리포지토리에](#) 제공된 CloudFormation 템플릿을 사용하여 이 패턴에 맞는 IAM 리소스와 S3 버킷을 자동으로 생성합니다. 이러한 리소스를 수동으로 생성하려면 에픽 섹션의 단계를 따르세요.

모범 사례

- IAM 모범 사례에 따라 Amazon S3에 Veeam Backup & Replication 백업을 작성할 때 사용하는 IAM 사용자와 같은 장기 IAM 사용자 보안 인증 정보를 정기적으로 교체하는 것이 좋습니다. 자세한 내용은 IAM 설명서의 [보안 모범 사례](#)를 참조하세요.

에픽

계정에서 Amazon S3 스토리지 구성

작업	설명	필요한 기술
IAM 사용자를 생성합니다.	<p>IAM 설명서의 지침에 따라 IAM 사용자를 생성합니다. 이 사용자는 AWS Console 액세스 권한이 없어야 하며 이 사용자에 대한 액세스 키를 생성해야 합니다. Veeam은 이 개체를 사용하여 AWS에 인증하고 S3 버킷을 읽고 씁니다. 사용자가 필요 이상의 권한을 갖지 않도록 최소 권한(즉 작업 수행에 필요한 권한만 부여)을 부여해야 합니다. Veeam IAM 사용자에게 연결할 IAM 정책의 예는 추가 정보 섹션을 참조하세요.</p> <p>참고 또는 VeeamHub GitHub 리포지토리에 제공된 CloudFormation 템플릿을 사용하여 이 패턴에 대한 IAM 사용자 및 S3 버킷을 생성할 수 있습니다.</p>	AWS 관리자
S3 버킷을 생성합니다.	<ol style="list-style-type: none"> 1. AWS Management Console에 로그인하고 https://console.aws.amazon.com/s3/에서 Amazon S3 콘솔을 엽니다. 2. 대상 스토리지로 사용할 기존 S3 버킷이 아직 없는 경우 버킷 생성을 선택하고 버 	AWS 관리자

작업	설명	필요한 기술
	<p>킷 이름, AWS 리전, 버킷 설정을 지정합니다.</p> <ul style="list-style-type: none"> • S3 버킷에 대해 퍼블릭 액세스 차단 옵션을 활성화하고 조직의 요구 사항에 맞게 액세스 및 사용자 권한 정책을 설정하는 것이 좋습니다. 예는 Amazon S3 설명서를 참조하세요. • S3 객체 잠금을 바로 사용할 생각이 없더라도 활성화하는 것이 좋습니다. 이 설정은 S3 버킷 생성 시에만 활성화할 수 있습니다. <p>자세한 내용은 Amazon S3 설명서의 버킷 생성을 참조하세요.</p>	

Veeam Backup & Replication에 Amazon S3 및 S3 Glacier Flexible Retrieval(또는 S3 Glacier Deep Archive) 추가

작업	설명	필요한 기술
<p>새 객체 리포지토리 마법사를 실행합니다.</p>	<p>Veeam에서 객체 스토리지 및 스케일 아웃 백업 리포지토리를 설정하기 전에 용량 및 아카이브 계층에 사용할 Amazon S3 및 Amazon S3 Glacier 스토리지 리포지토리를 추가해야 합니다. 다음 에픽에서는 이러한 스토리지 리포지토리를 스케일 아웃 백업 리포지토리에 연결해 보겠습니다.</p>	<p>AWS 관리자, 앱 소유자</p>

작업	설명	필요한 기술
	<ol style="list-style-type: none"><li data-bbox="592 212 1015 296">1. Veeam 콘솔에서 백업 인포라 보기를 엽니다.<li data-bbox="592 317 1015 495">2. 인벤토리 창에서 백업 리포지토리 노드를 선택한 다음 리포지토리 추가를 선택합니다.<li data-bbox="592 516 1015 642">3. 백업 리포지토리 추가 대화 상자에서 객체 스토리지, Amazon S3를 선택합니다.	

작업	설명	필요한 기술
<p>용량 계층에 Amazon S3 스토리지를 추가합니다.</p>	<ol style="list-style-type: none"> 1. Amazon Cloud Storage Services 대화 상자에서 Amazon S3를 선택합니다. 2. 마법사의 이름 단계에서 객체 스토리지 이름과 간단한 설명(예: 생성자 및 생성 날짜)을 지정합니다. 3. 마법사의 계정 단계에서 객체 스토리지 계정을 지정합니다. <ul style="list-style-type: none"> • 보안 인증 정보에서는 첫 번째 에픽에서 생성한 IAM 사용자를 선택하여 Amazon S3 객체 스토리지에 액세스합니다. • AWS 리전에서는 Amazon S3 버킷이 위치한 AWS 리전을 선택합니다. 4. 마법사의 버킷 단계에서 객체 스토리지 설정을 지정합니다. <ul style="list-style-type: none"> • 데이터 센터 리전에서는 Amazon S3 버킷이 위치한 AWS 리전을 선택합니다. • 버킷에서는 첫 번째 에픽에서 생성한 S3 버킷을 선택합니다. • 폴더에서는 객체 스토리지 리포지토리를 매핑할 클라우드 폴더를 생성하거나 선택합니다. 	<p>AWS 관리자, 앱 소유자</p>

작업	설명	필요한 기술
	<ul style="list-style-type: none"> 불변성을 활성화하려면 최근 백업을 X일 동안 변경할 수 없음을 선택하고 백업을 잠글 기간을 설정합니다. 참고로 불변성을 활성화하면 Veeam에서 Amazon S3에 대한 API 직접 호출 수가 증가하기 때문에 비용이 상승합니다. <p>5. 마법사의 요약 단계에서 구성 정보를 검토한 다음 완료 버튼을 선택합니다.</p>	

작업	설명	필요한 기술
<p>아카이브 계층에 S3 Glacier 스토리지를 추가합니다.</p>	<p>아카이브 계층을 생성하려면 IAM 권한(추가 정보 섹션에서 자세히 설명)을 사용합니다.</p> <ol style="list-style-type: none"> 1. 앞서 설명한 대로 새 객체 리포지토리 마법사를 실행합니다. 2. Amazon Cloud Storage Services 대화 상자에서 Amazon S3 Glacier를 선택합니다. 3. 마법사의 이름 단계에서 객체 스토리지 이름과 간단한 설명(예: 생성자 및 생성 날짜)을 지정합니다. 4. 마법사의 계정 단계에서 객체 스토리지 계정을 지정합니다. <ul style="list-style-type: none"> • 보안 인증 정보에서는 첫 번째 에픽에서 생성한 IAM 사용자를 선택하여 Amazon S3 Glacier 객체 스토리지에 액세스합니다. • AWS 리전에서는 Amazon S3 버킷이 위치한 AWS 리전을 선택합니다. 5. 마법사의 버킷 단계에서 객체 스토리지 설정을 지정합니다. <ul style="list-style-type: none"> • 데이터 센터 리전에서는 AWS 리전을 선택합니다. 	<p>AWS 관리자, 앱 소유자</p>

작업	설명	필요한 기술
	<ul style="list-style-type: none"> • 버킷에서는 백업 데이터를 저장할 S3 버킷을 선택합니다. 이 버킷은 용량 계층에 사용한 버킷과 동일할 수 있습니다. • 폴더에서는 객체 스토리지 리포지토리를 매핑할 클라우드 폴더를 생성하거나 선택합니다. • 불변성을 활성화하려면 보존 정책의 전체 기간에 최근 백업을 변경할 수 없음을 선택합니다. 참고로 불변성을 활성화하면 Veeam에서 Amazon S3에 대한 API 직접 호출 수가 증가하기 때문에 비용이 상승합니다. • S3 Glacier Deep Archive를 아카이브 스토리지 클래스로 사용하려면 Deep Archive 스토리지 클래스 사용을 선택합니다. <p>6. 마법사의 프록시 어플라이언스 단계에서 Amazon S3에서 Amazon S3 Glacier로 데이터를 전송하는 데 사용하는 보조 인스턴스를 구성합니다. 기본 설정을 사용하거나 각 설정을 수동으로 구성할 수 있습니다. 설정을 수동으로 구성하려면 다음 단계를 따릅니다.</p>	

작업	설명	필요한 기술
	<ul style="list-style-type: none"> • 사용자 지정을 선택합니다. • EC2 인스턴스 유형에서는 백업 파일을 스케일 아웃 백업 리포지토리의 아카이브 계층으로 전송하는 데 필요한 속도와 비용 요구 사항에 따라 프록시 어플라이언스의 인스턴스 유형을 선택합니다. • Amazon VPC에서는 대상 인스턴스의 VPC를 선택합니다. • 서브넷에서는 프록시 어플라이언스의 서브넷을 선택합니다. • 보안 그룹에서는 프록시 어플라이언스와 연결할 보안 그룹을 선택합니다. • 리디렉터 포트에서는 프록시 어플라이언스와 백업 인프라 구성 요소 간에 요청을 라우팅하기 위한 TCP 포트를 지정합니다. • 확인을 선택하여 설정을 확인합니다. <p>7. 마법사의 요약 단계에서 구성 정보를 검토한 다음 완료를 선택합니다.</p>	

스케일 아웃 백업 리포지토리 추가

작업	설명	필요한 기술
<p>새 스케일 아웃 백업 리포지토리 마법사를 실행합니다.</p>	<ol style="list-style-type: none"> 1. Veeam 콘솔에서 백업 인프라 보기를 엽니다. 2. 인벤토리 창에서 스케일 아웃 리포지토리를 선택한 다음 스케일 아웃 리포지토리 추가를 선택합니다. 	<p>앱 소유자, AWS 시스템 관리자</p>
<p>스케일 아웃 백업 리포지토리를 추가하고 용량 및 아카이브 계층을 구성합니다.</p>	<ol style="list-style-type: none"> 1. 마법사의 이름 단계에서 스케일 아웃 백업 리포지토리의 이름과 간략한 설명을 지정합니다. 2. 필요한 경우 성능 범위를 추가합니다. 기존 Veeam 로컬 백업 리포지토리를 성능 계층으로 사용할 수도 있습니다. Veeam 버전 12부터 로컬 성능 등급을 우회하여 direct-to-object (DTO) 백업의 성능 범위로 S3 버킷을 추가할 수 있습니다. 3. 고급을 선택하고 스케일 아웃 백업 리포지토리에 대한 추가 옵션을 지정합니다. <ul style="list-style-type: none"> • 시스템별 백업 파일 사용을 선택하여 각 시스템에 대해 별도의 백업 파일을 만들고 이 파일을 여러 스트림의 백업 리포지토리에 동시에 씁니다. 이 옵션은 더 나은 스토리지 및 컴퓨팅 리소스 활용을 위해 권장됩니다. 	<p>앱 소유자, AWS 시스템 관리자</p>

작업	설명	필요한 기술
	<ul style="list-style-type: none"> • 증분 백업의 복원 지점이 포함된 익스텐트가 오프라인 상태가 되는 경우 전체 백업 파일을 생성하려면 필요한 익스텐트가 오프라인일 때 전체 백업 수행을 선택합니다. 이 옵션을 사용하려면 스케일 아웃 백업 리포지토리에 전체 백업 파일을 호스팅할 수 있는 여유 공간이 필요합니다. <p>4. 마법사의 정책 단계에서 리포지토리의 백업 배치 정책을 지정합니다.</p> <ul style="list-style-type: none"> • 동일한 체인에 속하는 전체 및 증분 백업 파일을 동일한 성능 범위로 함께 저장하려면 데이터 로컬리티를 선택합니다. (중복 제거 스토리지 어플라이언스를 성능 범위로 사용하지 않는 한) 새 백업 체인에 속하는 파일을 동일한 성능 범위 또는 다른 성능 범위에 저장할 수 있습니다. • 전체 및 증분 백업 파일을 다양한 성능 범위로 저장하려면 성능을 선택합니다. 이 옵션을 사용하려면 빠르고 안정적인 네트워크 연결이 필요합니다. 성능을 선택하면 각 성능 범위에서 저장할 백업 파일 	

작업	설명	필요한 기술
	<p>유형을 제한할 수 있습니다. 예를 들어 한 범위에는 전체 백업 파일을 저장하고 다른 범위에는 증분 백업 파일을 저장할 수 있습니다. 파일 유형을 선택하려면 다음 단계를 따릅니다.</p> <ul style="list-style-type: none"> • 사용자 지정을 선택합니다. • 백업 배치 설정 대화 상자에서 성능 범위를 선택한 다음 편집을 선택합니다. • 해당 범위에 저장할 백업 파일 유형을 선택합니다. <p>5. 마법사의 용량 계층 단계에서 스케일 아웃 백업 리포지토리에 연결할 장기 스토리지 계층을 구성합니다.</p> <ul style="list-style-type: none"> • 객체 스토리지로 스케일 아웃 백업 리포지토리 용량 확장을 선택합니다. 객체 스토리지 리포지토리의 경우 이전 에픽에서 추가한 용량 계층에 해당하는 Amazon S3 스토리지를 선택합니다. • 기간을 선택하여 데이터 이동 또는 복사를 위한 기간을 선택합니다. 	

작업	설명	필요한 기술
	<ul style="list-style-type: none"> • 백업을 생성하는 즉시 객체 스토리지에 복사를 선택하여 전체 또는 최근에 생성된 백업 파일만 용량 범위까지 복사합니다. • 백업이 운영 복원 기간을 벗어나면 객체 스토리지로 이동을 선택하여 비활성 백업 체인을 용량 범위까지 전송합니다. X일보다 오래된 백업 파일 이동 필드에 백업 파일을 오프로드하기 전까지의 기간을 지정합니다. (비활성 백업 체인을 생성된 날에 오프로드하려면 0일로 지정합니다.) 스케일 아웃 백업 리포지토리가 지정한 임계값에 도달한 경우 재정을 선택하여 백업 파일을 더 빨리 이동할 수 있습니다. • 객체 스토리지에 업로드된 데이터 암호화를 선택하고 암호를 지정하여 오프로드할 모든 데이터와 해당 메타데이터를 암호화합니다. 암호 추가 또는 암호 관리를 선택하여 새 암호를 지정합니다. <p>6. 마법사의 아카이브 계층 단계에서 스케일 아웃 백업 리포지토리에 연결할 아카이브 스토리지 계층을 구성합</p>	

작업	설명	필요한 기술
	<p>니다. (Amazon S3 Glacier 스토리지 추가를 건너뛰었다면 이 단계가 나타나지 않습니다.)</p> <ul style="list-style-type: none"> GFS 전체 백업을 객체 스토리지에 아카이브를 선택합니다. 객체 스토리지 리포지토리의 경우 이전 에픽에서 추가한 Amazon S3 Glacier 스토리지를 선택합니다. N일보다 오래된 GFS 백업 아카이브의 경우 파일을 아카이브 범위로 이동할 기간을 선택합니다. (비활성 백업 체인을 생성된 날에 아카이브하려면 0일로 지정합니다.) <p>7. 마법사의 요약 단계에서 스케일 아웃 백업 리포지토리의 구성을 검토한 다음 완료 버튼을 선택합니다.</p>	

관련 리소스

- [AWS 계정에서 IAM 사용자 생성](#)(IAM 설명서)
- [버킷 생성](#)(Amazon S3 설명서)
- [Amazon S3 스토리지에 대한 퍼블릭 액세스 차단](#)(Amazon S3 설명서)
- [S3 객체 잠금 사용](#)(Amazon S3 설명서)
- [Veeam 기술 설명서](#)
- [S3 객체 스토리지 연결을 위한 안전한 IAM 정책을 생성하는 방법](#)(Veeam 설명서)

추가 정보

다음 섹션에서는 이 패턴의 [에픽](#) 섹션에서 IAM 사용자를 생성할 때 사용할 수 있는 샘플 IAM 정책을 설명합니다.

용량 계층에 대한 IAM 정책

참고 예제 정책의 S3 버킷 이름을 <yourbucketname>에서 Veeam 용량 계층 백업에 사용할 S3 버킷 이름으로 변경하세요.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "VisualEditor0",
      "Effect": "Allow",
      "Action": [
        "s3:GetObjectVersion",
        "s3:ListBucketVersions",
        "s3:ListBucket",
        "s3:PutObjectLegalHold",
        "s3:GetBucketVersioning",
        "s3:GetObjectLegalHold",
        "s3:GetBucketObjectLockConfiguration",
        "s3:PutObject*",
        "s3:GetObject*",
        "s3:GetEncryptionConfiguration",
        "s3:PutObjectRetention",
        "s3:PutBucketObjectLockConfiguration",
        "s3:DeleteObject*",
        "s3:DeleteObjectVersion",
        "s3:GetBucketLocation"
      ],
      "Resource": [
        "arn:aws:s3::/*",
        "arn:aws:s3:::"
      ]
    },
    {
      "Sid": "VisualEditor1",
      "Effect": "Allow",
      "Action": [
```

```

        "s3:ListAllMyBuckets",
        "s3:ListBucket"
    ],
    "Resource": "*"
}
]
}

```

아카이브 계층에 대한 IAM 정책

참고 예제 정책의 S3 버킷 이름을 <yourbucketname>에서 Veeam 아카이브 계층 백업에 사용할 S3 버킷 이름으로 변경하세요.

기존 VPC, 서브넷, 보안 그룹을 사용하려면:

```

{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "VisualEditor0",
      "Effect": "Allow",
      "Action": [
        "s3:DeleteObject",
        "s3:PutObject",
        "s3:GetObject",
        "s3:RestoreObject",
        "s3:ListBucket",
        "s3:AbortMultipartUpload",
        "s3:GetBucketVersioning",
        "s3:ListAllMyBuckets",
        "s3:GetBucketLocation",
        "s3:GetBucketObjectLockConfiguration",
        "s3:PutObjectRetention",
        "s3:GetObjectVersion",
        "s3:PutObjectLegalHold",
        "s3:GetObjectRetention",
        "s3:DeleteObjectVersion",
        "s3:ListBucketVersions",
        "ec2:DescribeInstances",
        "ec2:CreateKeyPair",
        "ec2:DescribeKeyPairs",
        "ec2:RunInstances",
        "ec2>DeleteKeyPair",

```

```

    "ec2:DescribeVpcAttribute",
    "ec2:CreateTags",
    "ec2:DescribeSubnets",
    "ec2:TerminateInstances",
    "ec2:DescribeSecurityGroups",
    "ec2:DescribeImages",
    "ec2:DescribeVpcs"
  ],
  "Resource": "*"
}
]
}

```

새 VPC, 서브넷, 보안 그룹을 생성하려면:

```

{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "VisualEditor0",
      "Effect": "Allow",
      "Action": [
        "s3:DeleteObject",
        "s3:PutObject",
        "s3:GetObject",
        "s3:RestoreObject",
        "s3:ListBucket",
        "s3:AbortMultipartUpload",
        "s3:GetBucketVersioning",
        "s3:ListAllMyBuckets",
        "s3:GetBucketLocation",
        "s3:GetBucketObjectLockConfiguration",
        "s3:PutObjectRetention",
        "s3:GetObjectVersion",
        "s3:PutObjectLegalHold",
        "s3:GetObjectRetention",
        "s3:DeleteObjectVersion",
        "s3:ListBucketVersions",
        "ec2:DescribeInstances",
        "ec2:CreateKeyPair",
        "ec2:DescribeKeyPairs",
        "ec2:RunInstances",
        "ec2>DeleteKeyPair",

```

```
    "ec2:DescribeVpcAttribute",
    "ec2:CreateTags",
    "ec2:DescribeSubnets",
    "ec2:TerminateInstances",
    "ec2:DescribeSecurityGroups",
    "ec2:DescribeImages",
    "ec2:DescribeVpcs",
    "ec2:CreateVpc",
    "ec2:CreateSubnet",
    "ec2:DescribeAvailabilityZones",
    "ec2:CreateRoute",
    "ec2:CreateInternetGateway",
    "ec2:AttachInternetGateway",
    "ec2:ModifyVpcAttribute",
    "ec2:CreateSecurityGroup",
    "ec2>DeleteSecurityGroup",
    "ec2:AuthorizeSecurityGroupIngress",
    "ec2:AuthorizeSecurityGroupEgress",
    "ec2:DescribeRouteTables",
    "ec2:DescribeInstanceTypes"
  ],
  "Resource": "*"
}
]
```

AWS에서 VMware 클라우드를 NetBackup 사용하도록 베리타스를 구성합니다.

작성자: Shubham Salani

환경: 프로덕션

기술: 클라우드 네이티브 스토리지 및 백업

워크로드: 기타 모든 워크로드

AWS 서비스: Amazon S3, AWS Transit Gateway, Amazon VPC, Amazon EBS

요약

많은 기업이 온프레미스 VMware vSphere 기반 워크로드를 위한 백업 및 복구 솔루션으로 NetBackup 베리타스를 사용합니다. 기업이 Amazon Web Services (AWS) 기반 VMware 클라우드 인프라의 소프트웨어 정의 데이터 센터 (SDDC) 로 워크로드를 마이그레이션한 후에는 통합을 위한 명확한 절차가 없습니다. lift-and-shift NetBackup 이 패턴은 AWS NetBackup 계정에서 베리타스를 설정하고 VMware SDDC에서 워크로드를 백업하도록 구성하는 방법을 설명합니다.

이 패턴은 워크로드 마이그레이션 지침을 포함하지 않습니다. 자세한 내용은 [VMware HCX를 사용하여 VMware SDDC를 AWS의 VMware Cloud에 마이그레이션](#)을 참조하세요. AWS의 VMware Cloud에 워크로드를 설정할 때는 [확장 클러스터](#)(VMware 설명서)를 사용합니다. 이 구성에서는 클러스터가 단일 리전 내의 AWS 가용 영역 두 개에 걸쳐 있습니다. 이는 가용 영역 중 하나를 사용할 수 없게 되는 경우에도 높은 가용성과 탄력성을 제공합니다. [Elastic DRS](#) 및 [vSAN 감시 호스트](#)(VMware 설명서)는 데이터를 장애 도메인이라고 하는 세 번째 가용 영역에 원활하게 복사합니다. 이 패리티 솔루션은 오류 발생 시 데이터를 복구하는 데 도움이 될 수 있습니다. 이 접근 방식에는 세 개의 가용 영역이 필요하므로 VMware Cloud 환경을 위한 AWS 리전을 선택할 때 가용 영역이 세 개 이상인지 확인합니다. 자세한 내용은 [리전 및 가용 영역](#)을 참조하세요.

이 패턴에서는 각 SDDC에 프록시 서버인 백업 호스트가 있습니다. Amazon Elastic Compute Cloud (Amazon EC2) 인스턴스를 사용하여 SDDC별로 하나씩, 별도의 가상 사설 클라우드 (VPC) 에 마스터 및 미디어 서버를 설정합니다 NetBackup . 엘라스틱 네트워크 인터페이스는 높은 대역폭과 짧은 지연 시간을 제공하므로, 이를 사용하여 백업 호스트와 해당 NetBackup 마스터 및 미디어 서버 간의 연결을 구성할 수 있습니다. EC2 인스턴스는 백업의 첫 번째 지점인 Amazon Elastic Block Store(Amazon

EBS) 볼륨으로 백업을 보냅니다. DataSync AWS를 사용하여 SDDC용 EBS 볼륨을 동기화된 상태로 유지할 수 있습니다.

또한 AWS Transit Gateway와 인터페이스 VPC 엔드포인트를 사용하여 EBS 볼륨을 Amazon Simple Storage Service(S3)와 같은 다른 스토리지 서비스에 연결할 수 있습니다. 보존 정책에 따라 S3 Intelligent-Tiering S3 Glacier 스토리지 클래스를 사용하여 스토리지 비용을 최적화할 수 있습니다. 자세한 내용은 [Amazon S3 스토리지 클래스 사용](#)(Amazon S3 설명서)을 참조하세요.

사전 조건 및 제한 사항

사전 조건

- AWS의 VMware Cloud 환경은 두 가용 영역에 걸친 확장 클러스터를 사용합니다.
- 백업 호스트는 VMware Virtual Machine Disk File(VMDK) 파일이 배포된 데이터 스토어에 액세스할 수 있는 AWS의 VMware Cloud SDDC에 있어야 합니다.
- HotAdd 가상 머신 (VM) 을 백업하고 복원하려면 NetBackup 클라이언트에서 전송 모드를 활성화해야 하며 사용자 지정 파일 및 폴더에서의 복원을 허용해야 합니다.

제한 사항

- NetBackup 마스터 서버는 SDDC에 있는 vCenter 백업 호스트의 프라이빗 IP 주소에 대한 DNS 확인을 사용해야 합니다.
- NetBackup 마스터 서버 및 백업 호스트의 호스트 파일에는 다음이 포함되어야 합니다.
 - 마스터 서버의 프라이빗 IP 주소 및 프라이빗 DNS 이름
 - 백업 호스트의 프라이빗 IP 주소 및 프라이빗 DNS 이름
- 인터페이스 VPC 엔드포인트를 S3 버킷으로 구성하는 경우, Classless Inter-Domain Routing(CIDR) 블록 소스에서 HTTPS를 허용하도록 SDDC Compute Gateway 방화벽을 구성해야 합니다. 자세한 내용은 [S3 엔드포인트를 사용한 S3 버킷 액세스](#)(VMware 설명서)를 참조하세요.
- VMware Cloud on AWS는 다음과 같은 기능을 NetBackup 지원하지 않습니다.
 - VM 템플릿 백업 또는 복원
 - NetBackup vSphere 클라이언트 사용 (HTML5 플러그인)
 - 백업 또는 복원을 위한 VM 잠금 및 잠금 해제
 - 백업은 vSAN 데이터 스토어에 저장할 수 없습니다.
 - 네트워크 블록 디바이스(NBD), NBDSSL 및 SAN 전송 모드

제품 버전

- AWS의 VMware Cloud SDDC 버전 1.0 이상
- 베리타스 NetBackup 버전 8.1.2 이상
- Linux 버전 6.8 이상
- VMware vSphere 버전 6.0 이상

아키텍처

다음 다이어그램은 VMware Cloud on AWS의 NetBackup 구성을 보여줍니다. NetBackup 마스터 및 미디어 서버는 별도의 VPC에 배포되며 엘라스틱 네트워크 인터페이스를 통해 SDDC의 백업 호스트에 연결됩니다. NetBackup 마스터 및 미디어 서버는 Amazon EBS 볼륨에 백업을 저장합니다. 선택적으로 AWS Transit Gateway와 AWS PrivateLink 인터페이스 VPC 엔드포인트를 사용하여 Amazon S3 버킷에 추가 스토리지를 구성할 수 있습니다.

도구

AWS 서비스 및 도구

- [Amazon Elastic Block Store\(Amazon EBS\)](#)는 Amazon Elastic Compute Cloud(Amazon EC2) 인스턴스에 사용할 수 있는 블록 수준 스토리지 볼륨을 제공합니다.
- [AWS](#)는 가상 사설 클라우드 (VPC) 에서 VPC 외부 서비스로의 단방향 프라이빗 연결을 생성할 수 있도록 PrivateLink 지원합니다.
- [Amazon Simple Storage Service\(S3\)](#)는 원하는 양의 데이터를 저장, 보호 및 검색하는 데 도움이 되는 클라우드 기반 객체 스토리지 서비스입니다.
- [Amazon Virtual Private Cloud\(VPC\)](#)를 이용하면 사용자가 정의한 가상 네트워크로 AWS 리소스를 시작할 수 있습니다. 이 가상 네트워크는 사용자의 자체 데이터 센터에서 운영하는 기존 네트워크와 유사하며 AWS의 확장 가능한 인프라를 사용한다는 이점이 있습니다.

기타 서비스

- [AWS의 VMware Cloud](#)는 Amazon Web Services(AWS)와 VMware가 공동으로 개발한 통합 클라우드 서비스입니다.
- [NetBackup VMware의 경우 VMware ESXi](#) 호스트에서 실행되는 VMware 가상 머신을 백업 및 복원합니다.

에픽

NetBackup 서버를 구성합니다.

작업	설명	필요한 기술
방화벽 규칙 업데이트.	<p>방화벽 규칙을 업데이트하여 VMware Cloud on AWS SDDC 와 마스터 및 NetBackup 미디어 서버 간의 연결을 설정합니다. 다음을 따릅니다.</p> <ol style="list-style-type: none"> 1. https://vmc.vmware.com/에서 AWS의 VMware Cloud에 로그인합니다. 2. 네트워킹 및 보안 탭에서 게이트웨이 방화벽을 선택합니다. 3. 게이트웨이 방화벽 페이지에서 게이트웨이 컴퓨팅을 선택합니다. 4. 규칙 추가를 선택한 다음 필요한 방화벽 포트 설정이 포함된 새 규칙을 생성합니다. 자세한 내용은 NetBackup 방화벽 포트 요구 사항 (베리타스 설명서) 을 참조하십시오. 	네트워크 관리자, 클라우드 관리자
NetBackup 마스터 및 미디어 서버를 시작합니다.	<ol style="list-style-type: none"> 1. AWS Management Console 에 로그인한 다음 https://console.aws.amazon.com/ec2/에서 Amazon EC2 콘솔을 엽니다. 2. EC2 인스턴스를 시작(Amazon EC2 설명서)하 	클라우드 관리자, 백업 관리자

작업	설명	필요한 기술
	<p>고 다음 구성 세부 정보를 사용합니다.</p> <p>a. NetBackup 마스터 및 미디어 서버의 경우 NBU-Linux-GA-8-1-2-Setup-f032d23e-881b-4dee-ba70-b9ca3e915910-ami-072509a7ffc156938.4 Amazon 머신 이미지 (AMI) 를 선택합니다. 사전 구성된 AMI는 AWS Marketplace 통해 제공됩니다.</p> <p>b. 인스턴스 유형을 선택합니다. NetBackup 마스터 및 미디어 서버에 권장됩니다. m5.2xlarge</p>	

작업	설명	필요한 기술
에 대한 백업 호스트를 구성합니다 NetBackup.	<ol style="list-style-type: none"> https://vmc.vmware.com/에서 AWS의 VMware Cloud에 로그인합니다. SDDC를 선택합니다. VCENTER 열기 탭을 선택합니다. 그러면 SDDC vCenter가 열립니다. 백업 호스트의 전체 주소도 메인 이름(FQDN)을 기록해 둡니다. NetBackup 관리 콘솔에 로그인합니다. 자세한 내용은 NetBackup 관리 콘솔에 로그인 (베리타스 설명서) 을 참조하십시오. 마스터 및 미디어 서버를 선택한 다음 VMware 액세스 호스트를 선택합니다. 백업 호스트의 FQDN을 추가합니다. 적용을 선택하고 확인을 선택합니다. 	클라우드 관리자, 백업 관리자

(선택 사항) Amazon S3 스토리지 설정

작업	설명	필요한 기술
Amazon S3에서 스토리지를 구성합니다.	<ol style="list-style-type: none"> Amazon S3 클라우드 스토리지 옵션(Veritas 설명서)을 검토하고 요구 사항에 적합한 스토리지 클래스를 선택합니다. 	클라우드 관리자, 일반 AWS

작업	설명	필요한 기술
	2. NetBackup(Veritas 설명서)의 클라우드 스토리지 구성의 지침에 따라 Amazon S3를 클라우드 스토리지에 사용하도록 구성합니다 NetBackup .	

관련 리소스

AWS 설명서

- [인터페이스 VPC 엔드포인트 생성](#) (AWS PrivateLink 설명서)

Veritas 설명서

- [NetBackup 방화벽 포트 요구 사항](#)

VMware 설명서

- [컨텐츠 라이브러리의 OVF 템플릿에서 VM 배포](#)
- [AWS의 VMware Cloud 데이터 전송 요금: 작동 원리](#) (VMware 블로그 게시물)
- [AWS의 VMware Cloud: 확장 클러스터](#)

Amazon S3용 DistCp PrivateLink AWS를 사용하여 온프레미스 하둡 환경에서 Amazon S3로 데이터를 마이그레이션합니다.

작성자: Jason Owens (AWS), Andres Cantor (AWS), Jeff Klopfenstein (AWS), Bruno Rocha Oliveira, Samuel Schmidt (AWS)

환경: 프로덕션	소스: Hadoop	대상: 전부
R 유형: 리플랫폼	워크로드: 오픈 소스	기술: 스토리지 및 백업, 분석
AWS 서비스: Amazon S3, Amazon EMR		

요약

이 패턴은 Amazon Simple Storage Service (Amazon S3) PrivateLink 용 AWS와 함께 Apache 오픈 소스 도구를 사용하여 온프레미스 Apache 하둡 환경에서 Amazon Web Services ([DistCpAWS](#)) 클라우드 로 거의 모든 양의 데이터를 마이그레이션하는 방법을 보여줍니다. 퍼블릭 인터넷 또는 프록시 솔루션을 사용하여 데이터를 마이그레이션하는 대신, Amazon [PrivateLink S3용 AWS를 사용하여 온프레미스 데이터 센터와 Amazon VPC \(Virtual Private Cloud\)](#) 간의 프라이빗 네트워크 연결을 통해 Amazon S3로 데이터를 마이그레이션할 수 있습니다. Amazon Route 53에서 DNS 항목을 사용하거나 온프레미스 Hadoop 클러스터의 모든 노드에 있는 /etc/hosts 파일에 항목을 추가하면 자동으로 올바른 인터페이스 엔드포인트로 이동합니다.

이 안내서는 데이터를 AWS 클라우드로 마이그레이션하는 DistCp 데 사용하는 지침을 제공합니다. DistCp 가장 일반적으로 사용되는 도구이지만 다른 마이그레이션 도구도 사용할 수 있습니다. [예를 들어 AWS Snowball 또는 AWS Snowmobile과 같은 오프라인 AWS도구나 AWS StorageGateway 또는 AWS와 같은 온라인 AWS 도구를 사용할 수 있습니다. DataSync 또한 Apache와 같은 다른 오픈 소스 도구도 사용할 수 있습니다. NiFi](#)

사전 조건 및 제한 사항

사전 조건

- 온프레미스 데이터 센터와 AWS 클라우드 간에 개인 네트워크로 연결된 활성 AWS 계정
- [Hadoop은 다음과 같이 온프레미스에 설치됩니다. DistCp](#)

- Hadoop 분산형 파일 시스템(HDFS)에서 마이그레이션 데이터에 액세스할 수 있는 Hadoop 사용자
- AWS Command Line Interface(AWS CLI), [설치](#) 및 [구성됨](#)
- 객체를 S3 버킷에 넣을 수 있는 [권한](#)

제한 사항

가상 사설 클라우드 (VPC) 제한은 Amazon PrivateLink S3용 AWS에 적용됩니다. 자세한 내용은 [인터페이스 엔드포인트 속성 및 제한 및 AWS PrivateLink 할당량 \(AWS PrivateLink 설명서\)](#) 을 참조하십시오.

Amazon PrivateLink S3용 AWS는 다음을 지원하지 않습니다.

- [Federal Information Processing Standard\(FIPS\) 엔드포인트](#)
- [웹 사이트 엔드포인트](#)
- [레거시 글로벌 엔드포인트](#)

아키텍처

소스 기술 스택

- 하둡 클러스터가 설치된 경우 DistCp

대상 기술 스택

- Amazon S3
- Amazon VPC

대상 아키텍처

다이어그램은 DistCp 하둡 관리자가 AWS Direct Connect와 같은 프라이빗 네트워크 연결을 통해 온 프레미스 환경에서 Amazon S3 인터페이스 엔드포인트를 통해 Amazon S3로 데이터를 복사하는 방법을 보여줍니다.

도구

서비스

- [AWS Identity and Access Management\(IAM\)](#)를 사용하여 AWS 리소스에 대한 액세스를 안전하게 제어할 수 있습니다.
- [Amazon Simple Storage Service\(S3\)](#)는 원하는 양의 데이터를 저장, 보호 및 검색하는 데 도움이 되는 클라우드 기반 객체 스토리지 서비스입니다.
- [Amazon Virtual Private Cloud\(VPC\)](#)를 이용하면 사용자가 정의한 가상 네트워크로 AWS 리소스를 시작할 수 있습니다. 이 가상 네트워크는 사용자의 자체 데이터 센터에서 운영하는 기존 네트워크와 유사하며 AWS의 확장 가능한 인프라를 사용한다는 이점이 있습니다.

기타 도구

- [Apache Hadoop DistCp](#) (분산 복사본) 은 대규모 클러스터 간 및 클러스터 내 복사에 사용되는 도구입니다. DistCp 배포, 오류 처리 및 복구, MapReduce 보고에 Apache를 사용합니다.

에픽

데이터를 AWS 클라우드로 마이그레이션

작업	설명	필요한 기술
Amazon S3용 PrivateLink AWS용 엔드포인트를 생성합니다.	<ol style="list-style-type: none"> 1. AWS Management Console에 로그인한 후 Amazon VPC 콘솔을 엽니다. 2. 탐색 창에서 엔드포인트를 선택하고 엔드포인트 생성을 선택합니다. 3. 서비스 범주에서 AWS services를 선택합니다. 4. 검색 상자에 s3을 입력하고 Enter를 누릅니다. 5. 검색 결과에서 com.amazonaws를 선택합니다. < your-aws-region >.s3 서비스 이름. 여기서 유형 열의 값은 인터페이스입니다. 	AWS 관리자

작업	설명	필요한 기술
	<ol style="list-style-type: none"> 6. VPC에서 VPC를 선택합니다. 서브넷에는 사용자의 서브넷을 선택합니다. 7. 보안 그룹에서 TCP 443을 허용하는 보안 그룹을 선택하거나 새로 만듭니다. 8. 요구 사항에 따라 태그를 추가한 다음, 엔드포인트 생성을 선택합니다. 	
<p>엔드포인트를 확인하고 DNS 항목을 찾습니다.</p>	<ol style="list-style-type: none"> 1. Amazon VPC 콘솔을 열고 엔드포인트를 선택한 다음, 이전에 만든 엔드포인트를 선택합니다. 2. 세부 정보 탭에서 DNS 이름의 첫 번째 DNS 항목을 찾습니다. 이 항목은 지역 DNS 항목입니다. 이 DNS 이름을 사용하면 가용 영역과 관련된 DNS 항목 간에 요청이 번갈아 나타납니다. 3. 서브넷 탭을 선택합니다. 각 가용 영역에서 엔드포인트의 탄력적 네트워크 인터페이스 주소를 찾을 수 있습니다. 	AWS 관리자

작업	설명	필요한 기술
<p>방화벽 규칙 및 라우팅 구성을 확인합니다.</p>	<p>방화벽 규칙이 열려 있고 네트워크 구성이 올바르게 설정되었는지 확인하려면 텔넷을 사용하여 포트 443에서 엔드포인트를 테스트하십시오. 예:</p> <pre data-bbox="592 489 1027 1562"> \$ telnet vpce-<your-VPC-endpoint-ID> .s3.us-east-2.vpce .amazonaws.com 443 Trying 10.104.88.6... Connected to vpce-<your-VPC-endpoint-ID> .s3.us-east-2.vpce .amazonaws.com. ... \$ telnet vpce-<your-VPC-endpoint-ID> .s3.us-east-2.vpce .amazonaws.com 443 Trying 10.104.71 .141... Connected to vpce-<your-VPC-endpoint-ID> .s3.us-east-2.vpce .amazonaws.com. </pre> <p>참고: 리전 항목을 사용하는 경우 테스트에 성공하면 Amazon VPC 콘솔에서 선택한 엔드포인트의 서브넷 탭에서 볼 수 있는 두 개의 IP 주소 사이에서</p>	<p>네트워크 관리자, AWS 관리자</p>

작업	설명	필요한 기술
	DNS가 번갈아 나타나는 것을 볼 수 있습니다.	

작업	설명	필요한 기술
이름 풀이를 구성합니다.	<p>Hadoop이 Amazon S3 인터페이스 엔드포인트에 액세스할 수 있도록 이름 풀이를 구성해야 합니다. 엔드포인트 이름 자체는 사용할 수 없습니다. 대신, <your-bucket-name>.s3.<your-aws-region>.amazonaws.com 또는 *.s3.<your-aws-region>.amazonaws.com 을 해결해야 합니다. 이 이름 지정 제한 사항에 대한 자세한 내용은 Hadoop S3A 클라이언트 소개(Hadoop 웹사이트)를 참고하십시오.</p> <p>다음 구성 옵션 중 하나를 선택합니다.</p> <ul style="list-style-type: none"> • 온프레미스 DNS를 사용하여 엔드포인트의 개인 IP 주소를 확인합니다. 모든 버킷 또는 선택한 버킷의 동작을 재정의할 수 있습니다. 자세한 내용은 AWS를 사용한 Amazon S3에 대한 보안 하이브리드 액세스 (PrivateLinkAWS 블로그 게시물)의 “옵션 2: 도메인 이름 시스템 응답 정책 영역 (DNS RPZ)을 사용하여 Amazon S3에 액세스”를 참조하십시오. • 트래픽을 VPC의 리졸버 인바운드 엔드포인트에 조건부로 전달하도록 온프레미 	AWS 관리자

작업	설명	필요한 기술
	<p>스 DNS를 구성합니다. 트래픽은 Route 53으로 전달됩니다. 자세한 내용은 AWS를 사용한 Amazon S3에 대한 보안 하이브리드 액세스 (PrivateLinkAWS 블로그 게시물)의 “옵션 3: Amazon Route 53 Resolver 인바운드 엔드포인트를 사용하여 온프레미스에서 DNS 요청 전달하기”를 참조하십시오.</p> <ul style="list-style-type: none"> Hadoop 클러스터의 모든 노드에서 /etc/hosts 파일을 편집합니다. 이는 테스트용 임시 솔루션이므로 프로덕션에는 권장되지 않습니다. /etc/hosts 파일을 편집하려면 <code><your-bucket-name>.s3.<your-aws-region>.amazonaws.com</code> 또는 <code>s3.<your-aws-region>.amazonaws.com</code>에 대한 항목을 추가하십시오. /etc/hosts 파일은 한 항목에 대해 여러 IP 주소를 가질 수 없습니다. 가용 영역 중 하나에서 단일 IP 주소를 선택해야 하며, 이 경우 단일 장애 지점이 됩니다. 	

작업	설명	필요한 기술
<p>Amazon S3에 대한 인증을 구성합니다.</p>	<p>Hadoop을 통해 Amazon S3에 인증하려면 임시 역할 보안 인증 정보를 Hadoop 환경으로 내보내는 것을 권장합니다. 자세한 내용은 S3를 사용한 인증(Hadoop 웹사이트)을 참고하십시오. 장기 실행 작업의 경우, 사용자를 생성하고 S3 버킷에만 데이터를 넣을 권한이 있는 정책을 할당할 수 있습니다. 액세스 키와 비밀 키는 하둡에 저장할 수 있으며, DistCp 작업 자체와 하둡 관리자만 액세스할 수 있습니다. 암호 저장에 대한 자세한 내용은 Hadoop 보안 인증 정보 공급자를 통한 암호 저장(Hadoop 웹사이트)을 참고하십시오. 다른 인증 방법에 대한 자세한 내용은 AWS IAM Identity Center(AWS Single Sign-On 후속)용 설명서에서 AWS 계정에 대한 CLI 액세스와 함께 사용할 IAM 역할의 자격 증명을 얻는 방법을 참고하십시오.</p> <p>임시 보안 인증 정보를 사용하려면 보안 인증 정보 파일에 임시 보안 인증 정보를 추가하거나 다음 명령을 실행하여 보안 인증 정보를 사용자 환경으로 내보내십시오.</p>	<p>AWS 관리자</p>

작업	설명	필요한 기술
	<pre>export AWS_SESSION_TOKEN=SECRET-SESSION-TOKEN export AWS_ACCESS_KEY_ID=SESSION-ACCESS-KEY export AWS_SECRET_ACCESS_KEY=SESSION-SECRET-KEY</pre> <p>기존 액세스 키와 비밀 키 조합을 사용하는 경우 다음 명령을 실행하십시오.</p> <pre>export AWS_ACCESS_KEY_ID=my.aws.key export AWS_SECRET_ACCESS_KEY=my.secret.key</pre> <p>참고: 액세스 키와 비밀 키 조합을 사용하는 경우 명령의 자격 증명 공급자를 <code>org.apache.hadoop.fs.s3a.TemporaryAWSCredentialsProvider</code>로 변경하십시오 <code>org.apache.hadoop.fs.s3a.SimpleAWSCredentialsProvider</code></p>	

작업	설명	필요한 기술
<p>를 사용하여 데이터를 DistCp 전송합니다.</p>	<p>데이터를 전송하는 DistCp 데 사용하려면 다음 명령을 실행합니다.</p> <pre data-bbox="597 394 1024 1507"> hadoop distcp -Dfs.s3a. aws.credentials.pr vider=\ "org.apache.hadoop .fs.s3a.TemporaryA WSCredentialsProvi der" \ -Dfs.s3a.access. key="\${AWS_ACCESS_ KEY_ID}" \ -Dfs.s3a.secret. key="\${AWS_SECRET_ ACCESS_KEY}" \ -Dfs.s3a.session .token="\${AWS_SESS ION_TOKEN}" \ -Dfs.s3a.path.st yle.access=true \ -Dfs.s3a.connect ion.ssl.enabled=true \ -Dfs.s3a.endpoin t=s3.<your-aws-reg ion>.amazonaws.com \ hdfs:///user/root/ s3a://<your-bucket- name> </pre> <p>참고: Amazon PrivateLink S3 용 AWS에서 DistCp 명령을 사 용할 때 엔드포인트의 AWS 리 전은 자동으로 검색되지 않습 니다. Hadoop 3.3.2 이상 버전 에서는 S3 버킷의 AWS 리전 을 명시적으로 설정하는 옵션</p>	<p>마이그레이션 엔지니어, AWS 관리자</p>

작업	설명	필요한 기술
	<p>을 활성화하여 이 문제를 해결합니다. 자세한 내용은 AWS 리전을 설정하기 위한 fs.s3a.endpoint.region 옵션을 추가하는 S3A를 참고하십시오(Hadoop 웹사이트).</p> <p>추가 S3A 공급자에 대한 자세한 내용은 일반 S3A 클라이언트 구성(Hadoop 웹사이트)을 참고하십시오. 예를 들어, 암호화를 사용하는 경우 암호화 유형에 따라 위의 명령 시리즈에 다음 옵션을 추가할 수 있습니다.</p> <div style="border: 1px solid #ccc; border-radius: 10px; padding: 10px; margin: 10px 0;"> <pre>-Dfs.s3a.server-side-encryption-algorithm=AES-256 [or SSE-C or SSE-KMS]</pre> </div> <p>참고: S3A와 함께 인터페이스 엔드포인트를 사용하려면, 인터페이스 엔드포인트에 S3 리전 이름(예: s3.<your-aws-region>.amazonaws.com)의 DNS 별칭 항목을 생성해야 합니다. 지침은 Amazon S3에 대한 인증 구성 섹션을 참고하십시오. 이 해결 방법은 Hadoop 3.3.2 이하 버전에 필요합니다. S3A의 이후 버전에서는 이 해결 방법이 필요하지 않습니다.</p>	

작업	설명	필요한 기술
	<p>Amazon S3에 서명 문제가 있는 경우, Signature Version 4(SigV4) 서명을 사용하는 옵션을 추가합니다.</p> <pre data-bbox="597 428 1026 625">-Dmapreduce.map.java.opts="-Dcom.amazonaws.services.s3.enableV4=true"</pre>	

온프레미스 데이터베이스의 재해 CloudEndure 복구에 사용

작성자: Nishant Jain(AWS) 및 Anuraag Deekonda(AWS)

환경: PoC 또는 파일럿

기술: 스토리지 및 백업, 현대화, 데이터베이스

요약

경고: IAM 사용자는 장기 자격 증명을 보유하므로 보안 위험이 발생할 수 있습니다. 이러한 위험을 줄이려면 이러한 사용자에게 작업을 수행하는 데 필요한 권한만 제공하고 더 이상 필요하지 않을 때는 해당 사용자를 제거하는 것이 좋습니다.

이 패턴은 CloudEndure 재해 복구 (DR) 에 재해 복구 및 CloudEndure 장애 복구 클라이언트를 사용합니다. Amazon Elastic Compute Cloud(Amazon EC2) 인스턴스를 사용하여 온프레미스 데이터 센터 호스트에 DR을 설정합니다.

비클라우드 또는 기타 클라우드 인프라에서 Amazon Web Services (AWS) 클라우드로 복제하려면 CloudEndure 장애 복구 클라이언트를 사용해야 합니다. 재해 이벤트가 끝난 후에는 시스템을 페일백하는 것이 좋습니다. CloudEndure 타겟 시스템에서 소스 시스템으로 데이터를 복제하는 방향을 역방향으로 전환하여 페일백에 대비합니다. CloudEndure 사용자 콘솔은 현재 시작된 대상 시스템을 소스 시스템으로 취급합니다. 복제는 선택한 대상 시스템에서 원래 소스 인프라로 되돌아갑니다.

중요: 2021년 11월, AWS는 [AWS Elastic 재해 복구를](#) 시작했으며, 이 서비스는 현재 AWS에서의 재해 복구를 위한 권장 서비스입니다.

Elastic Disaster Recovery가 성공적으로 출시되면 AWS는 AWS GovCloud (미국) 지역을 포함한 모든 AWS 지역에서 CloudEndure 재해 복구의 가용성을 제한하기 시작합니다 (AWS 중국 지역은 계속 지원됨). 이는 다음 일정에 따라 진행됩니다.

1. 2023년 9월 1일 — 고객은 더 이상 모든 AWS 지역 (AWS 중국 지역 제외) 에서 새 CloudEndure DR 계정을 등록할 수 없습니다.
2. 2023년 12월 1일 — 새로운 CloudEndure DR 에이전트 설치에 더 이상 모든 AWS 지역 (AWS 중국 지역 제외) 에서 지원되지 않습니다. 기존 에이전트의 업그레이드가 지원됩니다.

3. 2024년 3월 31일 — 모든 AWS 지역 (AWS 중국 지역 제외) 에서 CloudEndure DR이 중단될 예정입니다.
4. CloudEndure [재해 복구 EOL의 업데이트된 일정은 설명서를 참조하십시오. CloudEndure](#)

이 간행물은 2024년 3월 31일에 삭제될 예정입니다. 진행 중인 마이그레이션 프로젝트에 필요한 경우 이 페이지의 제목 아래에 있는 PDF 링크를 사용하여 PDF 파일을 다운로드하고 저장하십시오.

사전 조건 및 제한 사항

사전 조건

- 활성 상태의 AWS 계정.
- 온프레미스 데이터베이스.

아키텍처

소스 기술 스택

- 온프레미스 데이터 센터의 데이터베이스

대상 기술 스택

- EC2 인스턴스의 데이터베이스(지원되는 운영 체제 버전의 전체 목록은 [Amazon EC2 FAQ](#) 참조)

소스 및 대상 네트워크 아키텍처

도구

- [CloudEndure 재해 복구](#) — CloudEndure 재해 복구는 물리적, 가상 및 클라우드 기반 서버를 AWS로 빠르고 안정적으로 복구하여 다운타임과 데이터 손실을 줄입니다. CloudEndure 재해 복구는 시스템 (운영 체제, 시스템 상태 구성, 데이터베이스, 애플리케이션 및 파일 포함) 을 대상 AWS 계정 및 선호 지역의 저렴한 스테이징 영역에 지속적으로 복제합니다. 재해가 발생한 경우 Disaster Recovery에서 몇 분 만에 완전히 프로비저닝된 상태로 수천 대의 시스템을 자동으로 시작하도록 CloudEndure 지시할 수 있습니다.

에픽

재해 복구 구독하기 CloudEndure

작업	설명	필요한 기술
CloudEndure 재해 복구 구독하기.	CloudEndure 재해 복구는 AWS Marketplace에서 이용할 수 있습니다.	일반 AWS
CloudEndure 계정 생성.	계정을 CloudEndure 등록하고 생성하세요. 그런 다음 이메일에서 구독을 확인합니다.	일반 AWS
계정 암호를 설정하고 이용 약관에 동의합니다.	암호는 8자 이상이어야 하며, 대문자, 소문자, 숫자 및 특수 문자를 포함해야 합니다.	일반 AWS

CloudEndure 프로젝트 생성

작업	설명	필요한 기술
CloudEndure 사용자 콘솔에 로그인합니다.	CloudEndure 사용자 콘솔에서 이전 단계에서 생성한 자격 증명으로 로그인합니다.	CloudEndure 관리자
새 프로젝트를 생성합니다	콘솔의 왼쪽 상단에서 더하기 (+) 버튼을 선택하여 프로젝트를 생성합니다. 프로젝트 유형으로 재해 복구를 선택합니다. AWS Marketplace를 통해 라이선스를 취득할 수 있습니다.	CloudEndure 관리자

AWS 보안 인증 생성 및 사용

작업	설명	필요한 기술
CloudEndure 솔루션에 대한 IAM 정책을 생성합니다.	CloudEndure 솔루션을 실행하기 위해 생성해야 하는 AWS Identity 및 Access Management (IAM) 정책은 사전 정의된 CloudEndure 정책을 기반으로 합니다. 이 CloudEndure 정책에는 AWS를 대상 인프라로 사용하는 데 필요한 권한이 포함되어 있습니다.	AWS 시스템 관리자
새 IAM 사용자를 생성하고 AWS 보안 인증을 생성합니다.	CloudEndure 사용자 콘솔에 필요한 AWS 자격 증명을 생성하려면 IAM 사용자를 한 명 이상 생성하고 이 사용자에게 CloudEndure 권한 정책을 할당하십시오. 콘솔은 액세스 키 ID와 비밀 액세스 키 가 필요합니다. AWS 액세스 키 관리 모범 사례를 따르려면 주기적으로 IAM 키를 교체 해야 합니다. IAM 키를 변경하면 복제 서버가 재시작되어 일시적인 지연이 발생합니다.	AWS 시스템 관리자
스테이징 영역 계정 보안 인증을 설정합니다.	CloudEndure 사용자 콘솔 에 로그인하고 마이그레이션 프로젝트를 선택합니다. 설정 및 정보 탭에서 AWS 보안 인증을 탐색하여 AWS 액세스	AWS 시스템 관리자

작업	설명	필요한 기술
	키 ID 및 비밀 액세스 키 ID를 제공합니다.	

애플리케이션 설정 구성

작업	설명	필요한 기술
복제 서버를 정의합니다.	자세한 내용은 CloudEndure 설 명서 를 참조하십시오.	CloudEndure 관리자

소스 시스템에 CloudEndure 에이전트 설치

작업	설명	필요한 기술
에이전트 설치 토큰을 찾습니다.	CloudEndure 사용자 콘솔에서 컴퓨터, 컴퓨터 작업, 컴퓨터 추가로 이동합니다. 소스 시스템에서 설치 프로그램 파일을 실행하면 먼저 설치 토큰을 입력하라는 메시지가 표시됩니다. 토큰은 CloudEndure 계정이 활성화될 때 자동으로 생성되는 고유한 문자열입니다. 하나의 설치 토큰을 사용하여 프로젝트가 허용하는 한 많은 소스 시스템에 에이전트를 설치할 수 있습니다.	CloudEndure 관리자
Linux 시스템에서는 설치 프로그램 실행합니다.	Linux 시스템의 경우 설치 프로그램 명령을 복사하고 소스 시스템에 로그인한 다음 설치 프로그램을 실행합니다.	CloudEndure 관리자

작업	설명	필요한 기술
	자세한 지침은 CloudEndure 설 명서를 참조하십시오.	
Windows 시스템에서는 설치 프로그램을 실행합니다.	Windows 시스템의 경우 설치 프로그램 파일을 각 시스템에 다운로드한 다음 설치 프로그램 명령을 실행합니다. 자세한 지침은 CloudEndure 설 명서를 참조하십시오.	CloudEndure 관리자
데이터를 복제합니다.	에이전트가 설치되면 소스 시스템 복제가 CloudEndure 시작되고 스테이징 영역으로 소스 시스템이 복제됩니다. 초기 동기화가 완료되면 컴퓨터가 CloudEndure 사용자 콘솔의 시스템 탭에 나타납니다.	CloudEndure 관리자

대상 시스템의 청사진 구성

작업	설명	필요한 기술
청사진의 소스 시스템을 선택합니다.	CloudEndure 사용자 콘솔의 시스템 탭에서 소스 컴퓨터를 선택하여 컴퓨터 세부 정보 창에 액세스합니다.	CloudEndure 관리자
대상 시스템의 청사진을 구성합니다.	청사진 탭에서 요구사항에 따라 대상 시스템의 설정을 구성합니다. 자세한 지침은 CloudEndure 설명서를 참조하십시오.	CloudEndure 관리자

DR 솔루션 테스트

작업	설명	필요한 기술
테스트 모드를 사용하여 솔루션을 테스트합니다.	테스트 모드 및 테스트 컷오버 확인에 대한 자세한 지침은 설명서를 참조하십시오. CloudEndure	CloudEndure 관리자
Amazon EC2 서버에서 시작된 대상 인스턴스를 테스트합니다.	각 대상 시스템을 테스트하려면 시스템 이름을 선택합니다. 그런 다음 대상 탭을 열고 새 IP 주소를 복사한 다음 Amazon EC2 인스턴스에서 새로 시작한 서버에 로그인합니다.	CloudEndure 관리자

다음을 사용하여 페일오버를 수행합니다. CloudEndure

작업	설명	필요한 기술
소스 시스템 상태를 확인합니다.	CloudEndure 사용자 콘솔 컴퓨터 페이지에서 페일오버하려는 소스 컴퓨터에 다음과 같은 상태 표시가 있는지 확인합니다. <ul style="list-style-type: none"> 데이터 복제 진행 – 지속적인 데이터 보호 상태 – 대상 시스템을 시작할 수 있음을 나타내는 로켓 아이콘 Disaster Recovery Lifecycle – 최근 테스트 완료 	CloudEndure 관리자
전환을 시작합니다.	1. 시스템 페이지에서 소스 시스템을 선택합니다.	CloudEndure 관리자

작업	설명	필요한 기술
	<ol style="list-style-type: none"> 2. 대상 시스템 시작 탭에서 복구 모드를 선택합니다. 3. 대상 시스템의 복구 지점을 선택합니다. 시스템은 장애 조치를 위한 새 대상 시스템을 시작할 때 복구 지점을 사용합니다. 최신 복구 지점을 사용하거나 목록에서 이전 복구 지점을 선택할 수 있습니다. 4. Launch 계속 진행을 선택합니다. 	

작업 진행 상황 및 완료 상태를 확인합니다.

작업 진행률 창에는 대상 시스템 시작 프로세스에 대한 세부 정보가 표시됩니다.

CloudEndure 관리자

페일오버가 완료되면 CloudEndure 사용자 콘솔의 재해 복구 수명 주기 상태가 페일오버로 변경되어 성공적으로 완료되었음을 나타냅니다.

페일백 클라이언트를 사용하여 CloudEndure 페일백 수행

작업	설명	필요한 기술
<p>CloudEndure 페일백 클라이언트 요구 사항을 검토하십시오.</p>	<p>CloudEndure 장애 복구 클라이언트를 사용하여 온프레미스 또는 기타 클라우드 인프라에서 AWS로 복제할 수 있습니다. CloudEndure 페일백 클라이언트의 요구 사항은 다음과 같습니다.</p>	<p>CloudEndure 관리자</p>

작업	설명	필요한 기술
	<ul style="list-style-type: none"> • MBR 부팅을 지원하는 BIOS 모드에서 부팅하도록 시스템을 구성해야 합니다. UEFI 모드로 부팅하도록 구성된 시스템(GPT 부팅만 지원)은 지원되지 않습니다. • CloudEndure 페일백 클라이언트에는 최소 4GB의 전용 RAM이 필요합니다. 	
<p>페일백을 준비합니다.</p>	<p>페일백 준비 작업을 시작하려면 먼저 모든 소스 시스템이 테스트 모드 또는 복구 모드에서 대상 시스템을 시작해야 합니다.</p> <p>프로젝트 작업 메뉴에서 페일백 준비를 선택한 다음 계속을 선택합니다. CloudEndure 에이전트와 페일백 클라이언트 페어링이 표시되면 시스템이 페일백할 준비가 된 것입니다.</p>	<p>CloudEndure 관리자</p>

작업	설명	필요한 기술
<p>온-프레미스 환경에서 CloudEndure 페일백 클라이언트를 다운로드하십시오.</p>	<p>CloudEndure 페일백 클라이언트를 소스 환경에 다운로드하려면 다음과 같이 하십시오.</p> <ol style="list-style-type: none"> 1. DR 프로젝트에서 설정 및 정보를 선택합니다. 2. 복제 설정 페이지에서 “기타 인프라”로 페일백하는 방법 알아보기 링크를 선택합니다. 3. 미확인 클라우드/기타 인프라로 페일백 대화 상자에서 여기에서 다운로드를 선택합니다. <p>파일이 자동으로 다운로드됩니다.</p>	<p>CloudEndure 관리자</p>

작업	설명	필요한 기술
<p>온프레미스 시스템의 복제를 시작합니다.</p>	<p>소스 시스템의 복제를 시작하려면 타겟 시스템을 CloudEndure 페일백 클라이언트 이미지 () 로 부팅해야 합니다. failback_client.iso 클라이언트가 Dynamic Host Configuration Protocol(DHCP)을 사용하여 네트워킹 설정을 가져올 수 없는 경우 설정을 수동으로 입력합니다.</p> <p>CloudEndure 페일백 클라이언트는 TCP 포트 443을 통해 console.clouendure.com에 연결하고 입력하라는 메시지가 표시된 자격 증명을 사용하여 인증합니다. CloudEndure</p>	<p>CloudEndure 관리자</p>
<p>지침에 따라 필요한 세부 정보를 제공합니다.</p>	<p>다음 세부 정보를 제공합니다.</p> <ul style="list-style-type: none"> • 설치 토큰 • 소스 시스템의 시스템 ID • 소스와 대상 간의 디스크 매핑 <p>CloudEndure 페일백 클라이언트가 공용 또는 사설 IP 주소를 통해 CloudEndure 사용자 콘솔 및 대상 컴퓨터에 연결되어 있는지 확인하십시오.</p>	<p>CloudEndure 관리자</p>
<p>소스 시스템 ID를 찾습니다.</p>	<p>소스 시스템 ID를 찾으려면 시스템 탭에서 시스템 이름을 선택하고 소스 탭에서 ID를 복사합니다.</p>	<p>CloudEndure 관리자</p>

작업	설명	필요한 기술
<p>소스 시스템을 대상 시스템에 연결합니다.</p>	<p>온프레미스 서버(대상 시스템)의 소스 시스템 ID(이제 AWS의 서버가 페일백의 소스임)를 입력합니다. AWS 시스템(소스)은 TCP 포트 1500의 온프레미스 서버(대상)에 연결하여 복제를 시작합니다.</p> <p>초기 복제가 완료되면 CloudEndure 사용자 콘솔에 복제가 지속적 데이터 보호 모드에 있다고 표시됩니다.</p>	<p>CloudEndure 관리자</p>
<p>필요한 경우 페일백 설정을 편집합니다.</p>	<p>페일백 설정을 편집하려면 시스템 이름을 선택한 다음 페일백 설정 탭을 선택합니다.</p>	<p>CloudEndure 관리자</p>

작업	설명	필요한 기술
<p>대상 시스템을 실행합니다.</p>	<p>대상 시스템을 시작하려면 다음을 수행합니다.</p> <p>각 시스템 이름 왼쪽에 있는 확인란을 선택하고 x 대상 시스템 시작 을 선택한 다음 복구 모드를 선택합니다.</p> <p>대화 상자에서 다음을 선택합니다.</p> <p>최신 복구 지점을 선택한 다음 Launch 계속을 선택합니다.</p> <p>시작 프로세스가 완료되면 CloudEndure 사용자 콘솔에 [데이터 복제 진행] 아래에 [CloudEndure 에이전트와 복제 서버 페어링] 상태가 표시됩니다.</p>	<p>CloudEndure 관리자</p>
<p>시스템을 정상 작동 상태로 되돌립니다.</p>	<p>이제 온프레미스 시스템이 소스가 되고 AWS 시스템이 대상이 되도록 데이터 복제 방향을 변경하십시오. 프로젝트를 작업을 선택한 다음 정상으로 돌아가기 및 계속 을 선택합니다.</p> <p>데이터 복제 방향이 바뀌고 시스템이 초기 동기화 프로세스를 거칩니다. 데이터 복제 진행률 옆에 모든 시스템의 지속적인 데이터 보호 상태가 표시되면 페일백 프로세스가 완료됩니다.</p>	<p>CloudEndure 관리자</p>

관련 리소스

AWS Marketplace

- [CloudEndure 재해 복구](#)

CloudEndure 설명서

- [콘솔에 로그인](#)
- [프로젝트 생성](#)
- [보안 인증 생성 및 사용](#)
- [복제 설정 구성](#)
- [CloudEndure 에이전트 설치](#)
- [Disaster Recovery 장애 조치 수행](#)

튜토리얼 및 동영상

- [CloudEndure 문제 해결 플레이북](#)
- [CloudEndure 동영상](#)
- [AWS로의 Disaster Recovery 데모](#)

패턴 더 보기

- [및 이벤트를 사용하여 CodeBuild Amazon CodeCommit S3로 이벤트 기반 백업을 자동화합니다. CloudWatch](#)
- [DynamoDB TTL을 사용하여 Amazon S3에 항목 자동으로 보관](#)
- [Systems Manager를 사용하여 SAP HANA 데이터베이스를 자동으로 백업하고 EventBridge](#)
- [BMC AMI 클라우드 데이터를 사용하여 메인프레임 데이터를 Amazon S3에 백업하고 보관합니다.](#)
- [AWS Glue를 사용하여 Amazon S3에서 Amazon Redshift로 데이터를 점차 늘려 로딩하기 위한 ETL 서비스 파이프라인 빌드](#)
- [Python을 사용하여 AWS에서 EBCDIC 데이터를 ASCII로 변환 및 압축 해제](#)
- [Oracle의 VARCHAR2\(1\) 데이터 유형을 Amazon Aurora PostgreSQL의 부울 데이터 유형으로 변환](#)
- [AWS CLI를 사용하여 S3 버킷에서 다른 계정 및 리전으로 데이터 복사](#)
- [Amazon ECS 작업 정의를 생성하고 Amazon EFS를 사용하여 EC2 인스턴스에 파일 시스템을 마운트](#)
- [Kinesis Data Streams를 사용하고 AWS CDK를 사용하는 Amazon Data Firehose를 사용하여 Amazon S3에 DynamoDB 레코드를 제공합니다.](#)
- [Amazon DynamoDB 테이블의 스토리지 비용 추정](#)
- [Security Hub를 사용하여 AWS Organizations의 퍼블릭 S3 버킷 식별](#)
- [Amazon RDS for Oracle DB 인스턴스를 AMS를 사용하는 다른 계정으로 마이그레이션](#)
- [SFTP용 AWS Transfer를 사용하여 온프레미스 SFTP 서버를 AWS로 마이그레이션](#)
- [AWS DMS를 사용하여 Oracle 파티션형 테이블을 PostgreSQL로 마이그레이션하기](#)
- [Rclone를 사용하여 Microsoft Azure Blob에서 Amazon S3로 데이터 마이그레이션하기](#)
- [AWS에서 PostgreSQL의 개별 행으로 Oracle CLOB 값을 마이그레이션](#)
- [AWS 대규모 마이그레이션에서 공유 파일 시스템 마이그레이션](#)
- [AWS SFTP를 사용하여 온프레미스에서 Amazon S3로 소규모 데이터 세트를 마이그레이션합니다.](#)
- [암호화를 사용하지 않는 인스턴스가 있는지 Amazon Aurora를 모니터링](#)
- [Transfer Family를 사용하여 메인프레임 파일을 Amazon S3로 직접 이동](#)
- [AWS Fargate와 함께 Amazon EKS에서 Amazon EFS를 사용하여 영구 데이터 스토리지로 스테이트풀 워크로드를 실행하십시오.](#)
- [S3 버킷을 AWS CloudFormation 스택으로 성공적으로 가져오기](#)
- [계정 또는 조직의 EBS 스냅샷 세부 정보 보기](#)

웹 및 모바일 앱

주제

- [AWS 리포지토리에서 최신 AWS Amplify 웹 애플리케이션을 지속적으로 배포하십시오. CodeCommit](#)
- [Amplify를 사용한 React 앱 생성과 Amazon Cognito를 사용한 인증 추가](#)
- [Amazon S3에 React 기반 단일 페이지 애플리케이션을 배포하고 CloudFront](#)
- [프라이빗 엔드포인트와 Application Load Balancer 사용하여 내부 웹 사이트에 Amazon API Gateway API 배포](#)
- [로컬 Angular QuickSight 애플리케이션에 Amazon 대시보드 내장](#)
- [패턴 더 보기](#)

AWS 리포지토리에서 최신 AWS Amplify 웹 애플리케이션을 지속적으로 배포하십시오. CodeCommit

작성자: Deekshitulu Pentakota(AWS), Sai Katakam(AWS)

환경: PoC 또는 파일럿

기술: 웹 및 모바일 앱, 현대화
DevOps

AWS 서비스: AWS Amplify,
AWS CodeCommit

요약

[현대식 웹 애플리케이션](#)은 모든 애플리케이션 구성 요소를 정적 파일로 패키징하는 단일 페이지 애플리케이션(SPA)으로 구성됩니다. AWS Amplify Hosting을 사용하면 Git 기반 리포지토리에서 관리되는 최신 웹 애플리케이션을 구축, 배포 및 호스팅하는 지속적 통합 및 지속적 배포(CI/CD) 파이프라인을 구축할 수 있습니다. Amplify Hosting을 코드 리포지토리에 연결하면 각 커밋에서 애플리케이션 프론트엔드와 백엔드를 배포하는 단일 워크플로가 시작됩니다. 이 접근 방식의 이점은 배포가 성공적으로 완료된 후에만 웹 애플리케이션이 업데이트되므로 프론트엔드와 백엔드 간의 불일치가 방지된다는 것입니다.

이 패턴에서는 AWS CodeCommit 리포지토리를 사용하여 최신 웹 애플리케이션을 관리합니다. 이 지침의 샘플 웹 애플리케이션은 React SPA 프레임워크를 사용합니다. 그러나 Amplify Hosting은 Angular, Vue, Next.js 같은 다른 많은 SPA 프레임워크를 지원하며 Gatsby, Hugo, Jekyll.과 같은 단일 사이트 생성기도 지원합니다.

이 패턴은 다음 서비스 및 개념을 사용해 본 경험이 있는 AWS Builder를 대상으로 합니다.

- AWS CodeCommit
- AWS Amplify Hosting
- React
- JavaScript
- Node.js
- npm
- Git

사전 조건 및 제한 사항

사전 조건

- 활성 상태의 AWS 계정
- Amplify 및 에서 리소스를 생성할 수 있는 권한. CodeCommit 자세한 내용은 [Amplify용 ID 및 액세스 관리 및 AWS용 ID 및 액세스 관리를](#) 참조하십시오. CodeCommit
- AWS Command Line Interface(AWS CLI), [설치](#) 및 [구성됨](#)
- 텍스트 편집기 또는 코드 편집기.
- CodeCommit, [Git 자격 증명을 사용하는 HTTPS 사용자를 위해 설정합니다.](#)
- Amplify의 [IAM 서비스 역할](#)입니다.
- npm 및 Node.js, [설치됨](#)(npm 설명서).

제한 사항

- 이 패턴에서는 API, 인증 또는 데이터베이스와 같은 Amplify 애플리케이션용 백엔드의 개발 및 통합에 대해서는 설명하지 않습니다. 백엔드에 대한 자세한 내용은 Amplify 설명서의 [백엔드 생성](#)을 참조하세요.

제품 버전

- AWS CLI 버전 2.0
- Node.js 16.x 이상

아키텍처

대상 기술 스택

- React SPA가 포함된 AWS CodeCommit 리포지토리
- AWS Amplify Hosting 워크플로

대상 아키텍처

도구

서비스

- [AWS Amplify Hosting](#)은 지속적인 배포로 풀스택 서버리스 웹 애플리케이션을 호스팅하기 위한 Git 기반 워크플로를 제공합니다.
- [CodeCommitAWS](#)는 자체 소스 제어 시스템을 관리할 필요 없이 Git 리포지토리를 비공개로 저장하고 관리할 수 있는 버전 제어 서비스입니다.
- [AWS Identity and Access Management\(IAM\)](#)는 누구에게 인증 및 사용 권한이 있는지 제어하여 AWS 리소스에 대한 액세스를 안전하게 관리할 수 있도록 도와줍니다.

기타 도구

- [Node.js 는 확장](#) 가능한 네트워크 애플리케이션을 구축하도록 설계된 이벤트 기반 JavaScript 런타임 환경입니다.
- [npm](#)은 Node.js 환경에서 실행되는 소프트웨어 레지스트리로, 패키지를 공유 또는 대여하고 개인 패키지의 배포를 관리하는 데 사용됩니다.

에픽

리포지토리 생성 CodeCommit

작업	설명	필요한 기술
리포지토리를 생성합니다.	지침은 설명서의 AWS CodeCommit 리포지토리 생성 을 CodeCommit 참조하십시오.	AWS DevOps
리포지토리를 복제합니다.	자세한 지침은 CodeCommit 설명서에서 리포지토리를 복제하여 CodeCommit 리포지토리에 연결 을 참조하십시오. 메시지가 표시되는 경우 Git 보안 인증 정보를 제공합니다.	앱 개발자

React 애플리케이션 생성

작업	설명	필요한 기술
<p>새 React 애플리케이션을 생성합니다.</p>	<ol style="list-style-type: none"> 다음 명령을 입력하여 복제된 리포지토리를 탐색합니다. CodeCommit 리포지토리 <repo name> 이름으로 바꾸십시오. <pre data-bbox="630 594 1029 674">\$ cd <repo name></pre> 다음 명령을 입력하여 복제된 리포지토리에 새 React 애플리케이션을 생성합니다. <pre data-bbox="630 905 1029 1024">\$ npx create-react-app .</pre> 애플리케이션을 코딩한 후 다음 명령을 입력하여 시작합니다. <pre data-bbox="630 1209 1029 1289">\$ npm start</pre> <p>사용자 지정 React 애플리케이션을 생성하는 방법에 대한 자세한 내용은 React 앱 생성 설명서의 React 앱 생성 지침을 참조하세요. Amplify 설명서의 프론트엔드 배포에 나와 있는 지침에 따라 샘플 React 애플리케이션을 Amplify 계정에 배포할 수도 있습니다.</p>	<p>앱 개발자</p>

작업	설명	필요한 기술
<p>브랜치를 생성하고 코드를 푸시합니다.</p>	<ol style="list-style-type: none"> 다음 명령을 입력하여 로컬에서 새 브랜치를 생성합니다. 여기에서 <branch>는 새 브랜치에 할당할 이름입니다. <div style="border: 1px solid #ccc; border-radius: 10px; padding: 10px; margin: 10px 0;"> <pre>\$ git checkout -b <branch></pre> </div> <ol style="list-style-type: none"> 다음 명령을 입력하여 브랜치를 CodeCommit 리포지토리로 푸시합니다. 여기서 <branch> 는 이전 단계에서 할당한 이름입니다. 자세한 내용은 커밋으로 작업을 참조하세요. <div style="border: 1px solid #ccc; border-radius: 10px; padding: 10px; margin: 10px 0;"> <pre>\$ git push --set-upstream origin <branch></pre> </div>	<p>앱 개발자</p>

AWS Amplify 호스팅에 애플리케이션을 배포합니다.

작업	설명	필요한 기술
<p>Amplify를 리포지토리에 연결합니다.</p>	<p>지침은 Amplify 호스팅 설명서의 리포지토리 연결을 참조하세요. CodeCommit AWS와 이전에 생성한 리포지토리 및 브랜치를 선택합니다.</p>	<p>앱 개발자</p>
<p>프론트엔드 빌드 설정을 정의합니다.</p>	<p>자세한 지침은 Amplify 호스팅 설명서에서 프론트엔드의 빌드 설정 확인을 참조하세요. 기본값을 그대로 사용하거나 다음 사항을 입력합니다.</p>	<p>앱 개발자</p>

작업	설명	필요한 기술
	<pre> Build settings: version: 0.1 frontend: phases: preBuild: commands: - npm ci build: commands: - npm run build artifacts: baseDirectory: build files: - '**/*' cache: paths: - node_modules/ **/* </pre>	
<p>검토하고 배포합니다.</p>	<p>자세한 지침은 Amplify 호스팅 설명서의 저장 및 배포를 참조하세요. 배포 프로세스가 완료 될 때까지 기다리세요.</p>	<p>앱 개발자</p>

지속적 배포를 검증합니다.

작업	설명	필요한 기술
<p>초기 배포를 검증합니다.</p>	<p>배포 프로세스가 완료되면 도 메인에서 링크를 선택합니다. 애플리케이션이 예상대로 작동 하는지 검증합니다.</p>	<p>앱 개발자</p>
<p>코드 리포지토리에 변경 사항을 푸시합니다.</p>	<p>로컬 워크스테이션에서 코드를 편집하고 변경 내용을 CodeCommit 리포지토리에 푸</p>	<p>앱 개발자</p>

작업	설명	필요한 기술
	시합니다. Amplify Hosting은 리포지토리의 변경 사항을 감지하고 자동으로 빌드 및 배포 프로세스를 시작합니다. 애플리케이션 업데이트가 도메인에 표시되는지 확인합니다.	

관련 리소스

AWS CodeCommit 설명서

- [AWS를 사용하기 위한 설정 CodeCommit](#)
 - [Git 보안 인증 정보를 사용하는 HTTPS 사용자를 위한 설정](#)
 - [AWS CLI 자격 증명 도우미를 사용하여 리눅스, macOS 또는 유닉스의 AWS CodeCommit 리포지토리에 HTTPS 연결을 설정하기 위한 단계 설정](#)
- [AWS 시작하기 CodeCommit](#)

AWS Amplify Hosting 설명서

- [기존 코드로 시작하기](#)
- [사용자 지정 도메인 설정](#)

React 리소스

- [리액트 앱 웹사이트 생성](#)
- [리액트 앱 설명서 생성](#)
- [React 앱 리포지토리 생성 \(GitHub\)](#)

Amplify를 사용한 React 앱 생성과 Amazon Cognito를 사용한 인증 추가

작성자: Rishi Singla

환경: PoC 또는 파일럿

기술: 웹 및 모바일 앱, 보안, ID, 규정 준수

워크로드: 기타 모든 워크로드

AWS 서비스: Amplify, Amazon Cognito

요약

이 패턴은 Amplify를 사용하여 React 기반 앱을 생성하는 방법과 Amazon Cognito를 사용하여 프론트 엔드에 인증을 추가하는 방법을 보여줍니다. Amplify는 AWS상의 모바일 및 웹 앱 개발을 가속화하는 도구 세트(오픈 소스 프레임워크, 시각적 개발 환경, 콘솔)와 서비스(웹 앱 및 정적 웹 사이트 호스팅)로 구성되어 있습니다.

사전 조건 및 제한 사항

사전 조건

- 활성 상태의 AWS 계정
- 시스템에 설치된 [Node.js](#) 및 [npm](#)

제품 버전

- Node.js 버전 10.x 이상(버전을 확인하려면 터미널 창에서 `node -v`(을)를 실행)
- npm version 6.x 이상(버전을 확인하려면 터미널 창에서 `npm -v`(을)를 실행)

아키텍처

대상 기술 스택

- AWS Amplify

- Amazon Cognito

도구

- [Amplify 명령줄 인터페이스\(CLI\)](#)
- [Amplify Libraries](#)(오픈 소스 클라이언트 라이브러리)
- [Amplify Studio](#)(시각적 인터페이스)

에픽

Amplify CLI 설치

작업	설명	필요한 기술
Amplify CLI를 설치합니다.	<p>Amplify CLI는 React 앱을 AWS 클라우드 서비스 생성을 위한 통합 툴체인입니다. Amplify CLI를 설치하려면, 다음을 실행합니다.</p> <pre>npm install -g @aws-amplify/cli</pre> <p>새 주요 버전이 출시되면 npm에서 알려줍니다. 그 경우 다음 명령을 사용하여 npm 버전을 업그레이드합니다.</p> <pre>npm install -g npm@9.8.0</pre> <p>여기서 9.8.0은 설치하려는 버전을 의미합니다.</p>	앱 개발자

React App 생성

작업	설명	필요한 기술
<p>React App을 생성합니다.</p>	<p>새 AMI를 생성하려면 명령을 사용합니다.</p> <pre data-bbox="594 453 1027 611">npx create-react-app amplify-react-application</pre> <p>여기서 <code>amplify-react-application</code> (은)는 앱 이름입니다.</p> <p>앱이 성공적으로 생성되면 다음과 같은 메시지가 표시됩니다.</p> <pre data-bbox="594 993 1027 1150">Success! Created amplify-react-application</pre> <p>다양한 하위 폴더가 있는 디렉터리가 React 앱용으로 생성됩니다.</p>	<p>앱 개발자</p>
<p>로컬 시스템에서 앱을 실행합니다.</p>	<p>이전 단계에서 생성한 <code>amplify-react-application</code> 디렉터리로 이동하여 다음 명령을 실행합니다.</p> <pre data-bbox="594 1581 1027 1696">amplify-react-application% npm start</pre> <p>그러면 로컬 시스템에서 React 앱이 실행됩니다.</p>	<p>앱 개발자</p>

Amplify CLI 구성

작업	설명	필요한 기술
<p>AWS 계정에 연결되도록 Amplify를 구성합니다.</p>	<p>다음 명령을 실행하여 Amplify를 구성합니다.</p> <pre data-bbox="594 453 1027 611">amplify-react-application % amplify configure</pre> <p>Amplify CLI는 다음 절차에 따라 AWS 계정에 대한 액세스 권한을 설정하도록 요청합니다.</p> <ol style="list-style-type: none"> 1. 관리자 계정을 사용하여 로그인합니다. 2. — 분석하려는 AWS 리전을 지정합니다. 3. 프로그래밍 방식으로 액세스할 수 있는 Identity and Access Management(IAM) 사용자를 생성하고 AdministratorAccess-Amplify 권한 정책을 사용자에게 연결합니다. 4. 액세스 키 ID 및 보안 액세스 키를 생성하고 복사합니다. 5. 터미널에 이러한 세부 정보를 입력합니다. 6. 프로파일 이름을 생성하거나 기본 프로파일을 사용합니다. <p>경고: 이 시나리오에서는 IAM 사용자에게 프로그래밍 액세스</p>	<p>일반 AWS, 앱 개발자</p>

작업	설명	필요한 기술
	<p>스 권한 및 장기 자격 증명에 필요한 권한이 부족하며, 이로 인해 보안상 위험이 따릅니다. 이 위험을 줄이려면 이러한 사용자에게 작업을 수행하는 데 필요한 권한만 제공하고 더 이상 필요하지 않을 경우 이러한 사용자를 제거하는 것이 좋습니다. 필요한 경우 액세스 키를 업데이트할 수 있습니다. 자세한 내용은 IAM 사용자 설명서의 액세스 키 업데이트를 참조하세요.</p> <p>이 절차는 터미널에 다음과 같이 표시됩니다.</p> <pre data-bbox="592 934 1031 1856"> Follow these steps to set up access to your AWS account: Sign in to your AWS administrator account: https://console.aws.amazon.com/ Press Enter to continue Specify the AWS Region ? region: us-east-1 Follow the instructions at https://docs.amazoncloudwatch.amazonaws.com/iamv2/home#/users/create Press Enter to continue </pre>	

작업	설명	필요한 기술
	<pre> Enter the access key of the newly created user: ? accessKeyId: ***** ? secretAccessKey: ***** ***** **** This would update/create the AWS Profile in your local machine ? Profile Name: new Successfully set up the new user. </pre> <p>이 절차에 대한 자세한 내용은 Amplify Dev Center의 설명서를 참조하십시오.</p>	

Amplify 초기화

작업	설명	필요한 기술
<p>Amplify 초기화</p>	<ol style="list-style-type: none"> 새 디렉터리에서 Amplify를 초기화하려면 다음을 실행합니다. <pre> amplify init </pre> <p>Amplify는 프로젝트 이름 및 구성 파라미터를 입력하는 프롬프트를 표시합니다.</p> <ol style="list-style-type: none"> 모든 파라미터를 지정한 다음 Y를 눌러 지정된 구성으 	<p>앱 개발자, 일반 AWS</p>

작업	설명	필요한 기술
	<p>로 프로젝트를 초기화합니다.</p> <pre> Project information Name: amplifyre actproject Environment: dev Default editor: Visual Studio Code App type: javascript Javascript framework: react Source Directory Path: src Distribution Directory Path: build Build Command: npm run-script build Start Command: npm run-script start </pre> <p>3. 이전 단계에서 생성한 프로 파일을 선택합니다. 리소스는 생성한 Amplify 프로젝트의 dev 환경에 배포됩니다.</p> <p>4. 리소스가 생성되었는지 확인하려면 AWS Amplify 콘솔을 열고 리소스를 생성하는데 사용된 AWS CloudForm</p>	

작업	설명	필요한 기술
	<p>ation 템플릿과 세부 정보를 확인할 수 있습니다.</p> <pre> Deploying root stack amplifyreactproject [===== ===== ----] 2/4 amplify-amplif yreactproject-d... AWS::CloudFormatio n::Stack CREATE_IN_PROGRESS UnauthRole AWS::IAM: :Role CREATE_COMPLETE DeploymentBucket AWS::S3:: Bucket CREATE_IN_PROGRESS AuthRole AWS::IAM: :Role CREATE_COMPLETE </pre>	

프론트엔드에 인증 추가

작업	설명	필요한 기술
인증 추가하기	<p>amplify add <category > 명령을 사용하여 사용자 로</p>	<p>앱 개발자, 일반 AWS</p>

작업	설명	필요한 기술
	<p>그인 또는 백엔드 API와 같은 기능을 추가할 수 있습니다. 이 단계에서는 명령을 사용하여 인증을 추가합니다.</p> <p>Amplify는 Amazon Cognito, 프론트엔드 라이브러리, 드롭인 인증자 UI 구성 요소를 포함한 백엔드 인증 서비스를 제공합니다. 기능에는 사용자 가입, 사용자 로그인, 다중 인증, 사용자 로그아웃, 암호 없는 로그인이 포함됩니다. Amazon, Google 과 Facebook 같은 페더레이션된 자격 증명 공급자와 통합함으로써 사용자를 인증할 수도 있습니다. Amplify 인증 범주는 API, 분석 장치, 스토리지와 같은 다른 Amplify 범주와 원활하게 통합되므로 인증된 사용자 및 인증되지 않은 사용자에 대한 권한 부여 규칙을 정의할 수 있습니다.</p> <p>1. React 앱에 대한 인증을 구성하려면 다음 명령을 실행합니다.</p> <pre data-bbox="630 1486 1029 1650">amplify-react-application1 % amplify add auth</pre> <p>그러면 다음과 같은 정보와 프롬프트가 표시됩니다. 비즈니스 및 보안 요구 사항에</p>	

작업	설명	필요한 기술
	<p>따라 적절한 구성을 선택할 수 있습니다.</p> <pre>Using service: Cognito, provided by: awscloudformation The current configured provider is Amazon Cognito. Do you want to use the default authentication and security configuration? (Use arrow keys) # Default configuration Default configuration with Social Provider (Federation) Manual configuration I want to learn more.</pre> <p>2. 간단한 예를 들어, 기본 구성을 선택한 다음 사용자의 로그인 메커니즘(이 경우 이메일)을 선택합니다.</p> <pre>How do you want users to be able to sign in? Username # Email</pre>	

작업	설명	필요한 기술
	<div data-bbox="630 210 1029 541" style="border: 1px solid #ccc; border-radius: 10px; padding: 10px; margin-bottom: 10px;"> <p>Phone Number</p> <p>Email or Phone Number</p> <p>I want to learn more.</p> </div> <p data-bbox="591 558 1010 638">3. 고급 설정을 우회하여 인증 리소스 추가를 완료합니다.</p> <div data-bbox="630 680 1029 1075" style="border: 1px solid #ccc; border-radius: 10px; padding: 10px; margin-bottom: 10px;"> <p>Do you want to configure advanced settings? (Use arrow keys)</p> <p># No, I am done.</p> <p>Yes, I want to make some additional changes.</p> </div> <p data-bbox="591 1092 1010 1222">4. 로컬 백엔드 리소스를 빌드하고 클라우드에서 프로비저닝합니다.</p> <div data-bbox="630 1264 1029 1419" style="border: 1px solid #ccc; border-radius: 10px; padding: 10px; margin-bottom: 10px;"> <pre data-bbox="649 1285 941 1394">amplify-react-application1 % amplify push</pre> </div> <p data-bbox="630 1457 1010 1587">이 명령은 계정의 Congito 사용자 풀을 적절하게 변경합니다.</p> <p data-bbox="591 1608 1029 1738">5. 를 사용하여 auth CloudFormation 리소스를 구성하려면 Y를 누르십시오.</p> <p data-bbox="630 1780 1010 1869">그러면 다음과 같은 리소스가 구성됩니다.</p>	

작업	설명	필요한 기술
	<pre> UserPool AWS::Cogn ito::UserPool CREATE_COMPLETE UserPoolClientWeb AWS::Cogn ito::UserPoolClient CREATE_COMPLETE UserPoolClientWeb AWS::Cogn ito::UserPoolClient CREATE_COMPLETE UserPoolClientRole AWS::IAM: :Role CREATE_COMPLETE UserPoolClientLambda AWS::Lamb da::Function CREATE_COMPLETE UserPoolClientLam bdaPolicy AWS::IAM::Policy CREATE_CO Mplete UserPoolClientLog Policy AWS::IAM::Policy CREATE_IN _Progress </pre> <p>또한 Cognito 콘솔을 사용하여 이러한 리소스를 볼 수도</p>	

작업	설명	필요한 기술
	<p>있습니다(Cognito 사용자 풀 및 자격 증명 풀 검색).</p> <p>이 단계는 Cognito 사용자 풀 및 자격 증명 풀 구성으로 React 앱의 src 폴더에 있는 aws-exports.js 파일을 업데이트합니다.</p>	

App.js 파일 변경

작업	설명	필요한 기술
App.js 파일 변경	<p>src 폴더에서 App.js 파일을 열고 수정합니다. 수정된 파일은 다음과 같아야 합니다.</p> <pre> { App.Js File after modifications: import React from 'react'; import logo from './ logo.svg'; import './App.css'; import { Amplify } from 'aws-amplify'; import { withAuthenticator, Button, Heading } from '@aws- amplify/ui-react'; import awsconfig from './aws-exports'; Amplify.configure(a wsconfig); function App({ signOut }) { return (<div> </pre>	앱 개발자

작업	설명	필요한 기술
	<pre> <h1>Thankyou for doing verification</ h1> <h2>My Content</ h2> <button onClick={ signOut}>Sign out</ button> </div>); } export default withAuthenticator(App); </pre>	
React 패키지를 가져옵니다.	<p>App.js 파일은 두 개의 React 패키지를 가져옵니다. 다음 명령을 사용하여 이 패키지를 설치합니다.</p> <pre> amplify-react-app1 ication1 % npm install --save aws-amplify @aws-amplify/ui-react </pre>	앱 개발자

React 앱을 실행하고 인증을 확인합니다.

작업	설명	필요한 기술
앱을 실행합니다.	<p>로컬 머신에서 React 앱을 시작합니다.</p> <pre> amplify-react-app1 ication1 % npm start </pre>	앱 개발자, 일반 AWS
인증 확인합니다.	<p>앱에 인증 파라미터를 입력하라는 프롬프트가 표시되는지</p>	앱 개발자, 일반 AWS

작업	설명	필요한 기술
	<p>확인합니다. (이 예제에서는 이메일을 로그인 방법으로 구성했습니다.)</p> <p>프론트엔드 UI에 로그인 자격 증명을 입력하라는 프롬프트가 표시되고 계정을 생성할 수 있는 옵션이 있어야 합니다.</p> <p>또한 Amplify 빌드 프로세스를 구성하여 지속적 배포 워크플로의 부분으로서 백엔드를 추가할 수 있습니다. 하지만 이 패턴에서는 그러한 옵션을 다루지 않습니다.</p>	

관련 리소스

- [시작하기](#)(npm 설명서)
- [독립 실행형 계정 생성](#)(계정 관리 설명서)
- [Amplify 설명서](#)
- [Amazon Cognito 설명서](#)

Amazon S3에 React 기반 단일 페이지 애플리케이션을 배포하고 CloudFront

작성자: 장 밥티스트 길로이스(AWS)

코드 리포지토리: React 기반 CORS 단일 페이지 애플리케이션	환경: 프로덕션	기술: 웹 및 모바일 앱, 클라우드 네이티브, 서버리스
워크로드: 기타 모든 워크로드	AWS 서비스: 아마존 CloudFront, 아마존 S3, 아마존 API 게이트웨이	

요약

단일 페이지 애플리케이션 (SPA) 은 API를 사용하여 표시된 웹 페이지의 콘텐츠를 동적으로 업데이트 하는 웹 사이트 또는 웹 애플리케이션입니다. JavaScript 이 접근 방식은 서버에서 전체 웹 페이지를 다시 로드하는 대신 새 데이터만 업데이트하므로 웹사이트의 사용자 경험과 성능을 향상시킵니다.

이 패턴은 Amazon Simple Storage Service (Amazon S3) 및 Amazon에서 React로 작성된 SPA를 코딩하고 호스팅하는 step-by-step 접근 방식을 제공합니다. CloudFront 이 패턴의 SPA는 Amazon API Gateway에서 제공하는 REST API를 사용하며 [교차 출처 리소스 공유\(CORS\)](#)에 대한 모범 사례를 보여줍니다.

사전 조건 및 제한 사항

사전 조건

- 활성 상태의 AWS 계정.
- 통합 개발 환경(IDE)(예: [AWS Cloud9](#))
- Node.js 및 npm, 설치 및 구성됨. 자세한 내용은 Node.js 설명서의 [다운로드](#) 섹션을 참조하세요.
- Yarn, 설치 및 구성됨. 자세한 내용은 [Yarn 설명서](#)를 참조하세요.
- Git, 설치 및 구성됨. 자세한 내용은 [Git 설명서](#)를 참조하세요.

아키텍처

이 아키텍처는 AWS CloudFormation (코드형 인프라) 를 사용하여 자동으로 배포됩니다. Amazon S3와 같은 리전 서비스를 사용하여 정적 자산을 저장하고 Amazon API Gateway를 사용하여 리전 API(REST) 엔드포인트를 노출합니다. 애플리케이션 로그는 Amazon을 사용하여 수집됩니다 CloudWatch. 모든 AWS API 호출은 CloudTrail AWS에서 감사됩니다. 모든 보안 구성(예: ID 및 권한)은 Amazon Identity and Access Management(IAM)에서 관리됩니다. 정적 콘텐츠는 아마존 CDN (CloudFront 콘텐츠 전송 네트워크) 을 통해 전송되며, DNS 쿼리는 Amazon Route 53에서 처리됩니다.

기술 스택

- Amazon API Gateway
- 아마존 CloudFront
- Amazon Route 53
- Amazon S3
- IAM
- 아마존 CloudWatch
- AWS CloudTrail
- AWS CloudFormation

도구

서비스

- [Amazon API Gateway](#)를 사용하면 어떤 규모에서든 REST, HTTP 및 WebSocket API를 생성, 게시, 유지 관리, 모니터링 및 보호할 수 있습니다.
- [AWS Cloud9](#)는 소프트웨어를 코딩, 빌드, 실행, 테스트 및 디버그하는 데 도움이 되는 IDE입니다. 또한, 소프트웨어를 AWS 클라우드로 릴리스하는 데도 도움이 됩니다.
- [AWS는 AWS CloudFormation](#) 리소스를 설정하고, 빠르고 일관되게 프로비저닝하고, AWS 계정 및 지역 전반의 수명 주기 전반에 걸쳐 리소스를 관리할 수 있도록 지원합니다.
- [Amazon](#)은 전 세계 데이터 센터 네트워크를 통해 웹 콘텐츠를 전송함으로써 웹 콘텐츠 배포 CloudFront 속도를 높여 지연 시간을 줄이고 성능을 개선합니다.
- [AWS는 AWS](#) 계정의 거버넌스, 규정 준수 및 운영 위험을 감사할 수 있도록 CloudTrail 지원합니다.

- [Amazon](#)은 AWS 리소스의 지표와 AWS에서 실행하는 애플리케이션을 실시간으로 모니터링할 수 있도록 CloudWatch 도와줍니다.
- [AWS Identity and Access Management\(IAM\)](#)는 사용자에 대한 인증 및 권한 부여를 제어함으로써 AWS 리소스에 대한 액세스를 안전하게 제어할 수 있습니다.
- [Amazon Route 53](#)은 가용성과 확장성이 뛰어난 DNS 웹 서비스입니다.
- [Amazon Simple Storage Service\(S3\)](#)는 원하는 양의 데이터를 저장, 보호 및 검색하는 데 도움이 되는 클라우드 기반 객체 스토리지 서비스입니다.

코드

이 패턴의 샘플 애플리케이션 코드는 GitHub [React 기반 CORS](#) 단일 페이지 애플리케이션 리포지토리에서 사용할 수 있습니다.

에픽

애플리케이션 코드 로컬에서 빌드 및 배포

작업	설명	필요한 기술
리포지토리를 복제합니다.	<p>이 패턴의 IDE로는 AWS Cloud9를 사용하는 것이 좋지만, 다른 IDE(예: Visual Studio Code 또는 IntelliJ IDEA)를 사용할 수도 있습니다.</p> <p>다음 명령을 실행하여 샘플 애플리케이션의 리포지토리를 IDE에 복제합니다.</p> <pre>git clone https://github.com/aws-samples/react-cors-spa react-cors-spa && cd react-cors-spa</pre>	앱 개발자, AWS DevOps
애플리케이션을 로컬 방식으로 배포합니다.	1. 프로젝트 디렉터리에서 <code>npm install</code> 명령을 실행하여	앱 개발자, AWS DevOps

작업	설명	필요한 기술
	<p>애플리케이션 종속성을 시작합니다.</p> <p>2. <code>yarn start</code> 명령을 실행하여 애플리케이션을 로컬로 시작합니다.</p>	
애플리케이션에 로컬로 액세스합니다.	브라우저 창을 열고 <code>http://localhost:3000</code> URL을 입력하여 애플리케이션에 액세스합니다.	앱 개발자, AWS DevOps

애플리케이션 배포

작업	설명	필요한 기술
AWS CloudFormation 템플릿을 배포합니다.	<ol style="list-style-type: none"> 1. AWS 관리 콘솔에 로그인한 다음 AWS CloudFormation 콘솔을 엽니다. 2. 스택 생성을 선택한 다음 새 리소스 사용(표준)을 선택합니다. 3. 템플릿 파일 업로드를 선택합니다. 4. 파일 선택을 선택하고 복제된 레포지토리에서 <code>react-cors-spa-stack.yaml</code> 파일을 선택하고 다음을 선택합니다. 5. 스택에 대해 이름을 입력한 후 다음을 선택합니다. 6. 기본 옵션을 유지한 다음에 다음을 선택합니다. 	앱 개발자, AWS DevOps

작업	설명	필요한 기술
	7. 스택의 최종 설정을 검토한 다음 스택 생성을 선택합니다.	
애플리케이션 소스 파일을 사용자 지정합니다.	<ol style="list-style-type: none"> 1. 스택을 배포한 후 출력 탭을 열고 APIEndpoint URL, Bucket 이름, 그리고 CFDistributionURL 을 (를) 식별합니다. 2. API 엔드포인트 URL을 복사합니다. 3. <project_root>/src /App.js 으(로) 이동한 다음 App.js 파일의 라인 26에 있는 APIEndPoint 변수 값에 URL을 붙여 넣습니다. 	앱 개발자
애플리케이션 패키지를 빌드합니다.	프로젝트 디렉터리에서 yarn build 명령을 실행하여 애플리케이션 패키지를 빌드합니다.	앱 개발자

작업	설명	필요한 기술
애플리케이션 패키지를 배포합니다.	<ol style="list-style-type: none"> 1. Amazon S3 콘솔을 엽니다. 2. 이전에 생성한 S3 버킷을 식별하고 선택합니다. 3. 업로드를 선택한 후 파일 추가를 선택합니다. 4. 빌드 폴더의 콘텐츠를 선택합니다. 5. 폴더 추가를 선택한 다음 정적 디렉터리를 선택합니다. 중요: 콘텐츠가 아닌 디렉터리를 선택하세요. 6. 업로드를 선택하여 파일 및 디렉터리를 S3 버킷에 업로드합니다. 	앱 개발자, AWS DevOps

애플리케이션 테스트

작업	설명	필요한 기술
애플리케이션을 액세스하고 테스트합니다.	브라우저 창을 열고 URL (이전에 배포한 CloudFormation 스택의 CFDistributionURL 출력) 을 붙여넣어 애플리케이션에 액세스합니다.	앱 개발자, AWS DevOps

리소스 정리

작업	설명	필요한 기술
S3 버킷 콘텐츠를 삭제합니다.	1. Amazon S3 콘솔을 열고 스택에서 이전에 생성한 버킷(이름이 react-cors-	AWS DevOps, 앱 개발자

작업	설명	필요한 기술
	<ol style="list-style-type: none"> spa-으(로) 시작하는 첫 버킷을 선택합니다. 비우기를 선택하여 버킷의 콘텐츠를 삭제합니다. 스택에서 이전에 생성한 두 번째 버킷(이름이 react-cors-spa- 으(로) 시작하고 -logs으(로) 끝나는 두 번째 버킷)을 선택합니다. 비우기를 선택하여 버킷의 콘텐츠를 삭제합니다. 	
AWS CloudFormation 스택을 삭제합니다.	<ol style="list-style-type: none"> AWS CloudFormation 콘솔을 열고 이전에 생성한 스택을 선택합니다. 스택과 모든 관련 리소스를 삭제하려면 삭제를 선택합니다. 	AWS DevOps, 앱 개발자

추가 정보

웹 애플리케이션을 배포 및 호스팅하려면 지속적인 배포로 풀스택 서버리스 웹 앱을 호스팅하기 위한 Git 기반 워크플로우를 제공하는 [AWS Amplify Hosting](#)을 사용할 수도 있습니다. Amplify Hosting는 [AWS Amplify](#)의 일부로, 프론트엔드 웹 및 모바일 개발자가 AWS에서 풀스택 애플리케이션을 빠르고 쉽게 구축할 수 있도록 특별히 제작된 도구 및 기능 세트입니다.

프라이빗 엔드포인트와 Application Load Balancer 사용하여 내부 웹 사이트에 Amazon API Gateway API 배포

작성자: Saurabh Kothari(AWS)

환경: 프로덕션

기술: 웹 및 모바일 앱, 네트워크, 서버리스, 인프라

AWS 서비스: Amazon API Gateway, Amazon Route 53, AWS Certificate Manager(ACM)

요약

이 패턴은 온프레미스 네트워크에서 액세스할 수 있는 내부 웹 사이트에 Amazon API Gateway API를 배포하는 방법을 보여줍니다. 프라이빗 엔드포인트, Application Load Balancer, PrivateLink AWS 및 Amazon Route 53으로 설계된 아키텍처를 사용하여 프라이빗 API의 사용자 지정 도메인 이름을 생성하는 방법을 알아봅니다. 이 아키텍처는 API에서의 도메인 기반 라우팅을 지원하기 위해 사용자 지정 도메인 이름 및 프록시 서버를 사용할 때 발생하는 의도하지 않은 결과를 방지합니다. 예를 들어, 라우팅이 불가능한 서브넷에 Virtual Private Cloud(VPC) 엔드포인트를 배포하면 네트워크가 API Gateway에 도달할 수 없습니다. 일반적인 해결 방법은 사용자 지정 도메인 이름을 사용한 다음 라우팅 가능한 서브넷에 API를 배포하는 것이지만, 프록시 구성이 트래픽(execute-api.{region}.vpce.amazonaws.com)을 AWS Direct Connect로 전달할 때 다른 내부 사이트가 손상될 수 있습니다. 마지막으로, 이 패턴은 인터넷으로는 연결할 수 없는 프라이빗 API와 사용자 지정 도메인 이름을 사용하기 위한 조직의 요구 사항을 충족하는 데 도움이 될 수 있습니다.

사전 조건 및 제한 사항

사전 조건

- 활성 상태의 AWS 계정
- 웹 사이트 및 API에 대한 서버 이름 표시(SNI) 인증서
- 온프레미스 환경에서 AWS Direct Connect 또는 AWS Site-to-Site VPN을 사용하여 설정된 AWS 계정으로의 연결
- 온프레미스 네트워크에서 확인되고 DNS 쿼리를 Route 53으로 전달하는 해당 도메인(예: domain.com)이 있는 [프라이빗 호스팅 영역](#)
- 온프레미스 네트워크에서 연결할 수 있는 라우팅 가능한 프라이빗 서브넷

제한 사항

로드 밸런서, 규칙 및 기타 리소스의 할당량(이전에는 제한이라고 함)에 대한 자세한 내용은 Elastic 로드 밸런서 설명서에서 [Application Load Balancer의 할당량](#)을 참조하십시오.

아키텍처

기술 스택

- Amazon API Gateway
- Amazon Route 53
- Application Load Balancer
- AWS Certificate Manager
- AWS PrivateLink

대상 아키텍처

다음 다이어그램은 Application Load Balancer 리스너 규칙에 따라 웹 트래픽을 웹 사이트 대상 그룹 또는 API Gateway 대상 그룹으로 보내는 VPC에 Application Load Balancer를 배포하는 방법을 보여줍니다. API Gateway 대상 그룹은 API Gateway의 VPC 엔드포인트에 대한 IP 주소 목록입니다. API Gateway는 리소스 정책을 사용하여 API를 비공개로 설정하도록 구성되어 있습니다. 이 정책은 특정 VPC 엔드포인트가 아닌 모든 직접 호출을 거부합니다. API Gateway의 사용자 지정 도메인 이름이 API 및 해당 스테이지에 `api.domain.com`을 사용하도록 업데이트되었습니다. Application Load Balancer 규칙이 추가되어 호스트 이름을 기반으로 트래픽을 라우팅합니다.

이 다이어그램은 다음 워크플로를 보여줍니다.

1. 온프레미스 네트워크의 사용자가 내부 웹 사이트에 액세스하려고 합니다. 요청은 `ui.domain.com`과 `api.domain.com`으로 전송됩니다. 그러면 요청이 라우팅 가능한 프라이빗 서브넷의 내부 Application Load Balancer로 전달됩니다. SSL은 `ui.domain.com` 및 `api.domain.com`의 Application Load Balancer에서 종료됩니다.
2. Application Load Balancer에 구성된 리스너 규칙은 호스트 헤더를 확인합니다.
 - a. 호스트 헤더가 `api.domain.com`인 경우 요청은 API Gateway 대상 그룹으로 전달됩니다. Application Load Balancer는 포트 443을 통해 API Gateway에 대한 새 연결을 시작합니다.
 - b. 호스트 헤더가 `api.domain.com`인 경우 요청은 웹사이트 대상 그룹으로 전달됩니다.

3. 요청이 API Gateway에 도달하면 API Gateway에 구성된 사용자 지정 도메인 매핑이 호스트 이름과 실행할 API를 결정합니다.

자동화 및 규모 조정

이 패턴의 단계는 AWS CloudFormation 또는 AWS 클라우드 개발 키트 (AWS CDK) 를 사용하여 자동화할 수 있습니다. API Gateway 직접 호출의 대상 그룹을 구성하려면 사용자 지정 리소스를 사용하여 VPC 엔드포인트의 IP 주소를 검색해야 합니다. API는 IP 주소 [describe-vpc-endpoints](#) 및 보안 그룹을 호출하고 [describe-network-interfaces](#) 반환하며, 이를 사용하여 IP 주소의 API 대상 그룹을 생성할 수 있습니다.

도구

- [Amazon API Gateway](#)를 사용하면 어떤 규모에서든 REST, HTTP 및 WebSocket API를 생성, 게시, 유지 관리, 모니터링 및 보호할 수 있습니다.
- [Amazon Route 53](#)은 가용성과 확장성이 뛰어난 DNS 웹 서비스입니다.
- [AWS Certificate Manager\(ACM\)](#)는 AWS 웹 사이트와 애플리케이션을 보호하는 퍼블릭 및 프라이빗 SSL/TLS X.509 인증서와 키를 만들고, 저장하고, 갱신하는 데 도움을 줍니다.
- [AWS Cloud Development Kit\(AWS CDK\)](#)는 AWS 클라우드 인프라를 코드로 정의하고 프로비저닝하는 데 도움이 되는 소프트웨어 개발 프레임워크입니다.
- [AWS](#)는 VPC에서 VPC 외부의 서비스로의 단방향 프라이빗 연결을 생성할 수 있도록 PrivateLink 지원합니다.

에픽

SNI 인증서 생성

작업	설명	필요한 기술
SNI 인증서를 생성하고 인증서를 ACM으로 가져옵니다.	1. ui.domain.com 및 api.domain.com용 SNI 인증서를 생성하십시오. 자세한 내용은 Amazon CloudFront 설명서의 CloudFront HTTPS 요청 처리 방법 선택을 참조하십시오.	네트워크 관리자

작업	설명	필요한 기술
	2. AWS Certificate Manager(ACM)에 SNI 인증서를 가져옵니다. 자세한 내용은 ACM 설명서의 AWS Certificate Manager로 인증서 가져오기 를 참조하십시오.	

라우팅할 수 없는 프라이빗 서브넷에 VPC 엔드포인트 배포

작업	설명	필요한 기술
API Gateway용 인터페이스 VPC 엔드포인트 생성	인터페이스 VPC 엔드포인트를 생성하려면 Amazon Virtual Private Cloud(VPC) 설명서에서 인터페이스 VPC 엔드포인트를 사용하여 AWS 서비스 액세스의 지침을 따르십시오.	클라우드 관리자

Application Load Balancer를 구성합니다.

작업	설명	필요한 기술
애플리케이션에 대상 그룹 생성	애플리케이션의 UI 리소스에 대해 대상 그룹을 생성합니다.	클라우드 관리자
API 게이트트웨이 엔드포인트에 대한 대상 그룹을 생성합니다.	<ol style="list-style-type: none"> IP 주소 유형으로 대상 그룹을 생성한 다음, API Gateway 엔드포인트에 대한 VPC 엔드포인트의 IP 주소를 대상 그룹에 추가합니다. 성공 코드 200 및 403을 사용하여 대상 그룹의 상태 확인을 구성합니다. API가 인증을 사용하고 403 응답을 	클라우드 관리자

작업	설명	필요한 기술
	반환할 수 있으므로 403이 필요합니다.	
Application Load Balancer을 생성합니다.	<ol style="list-style-type: none"> 라우팅 가능한 프라이빗 서브넷에서 Application Load Balancer (내부)를 생성합니다. Application Load Balancer에 443 리스너를 추가한 다음 ACM에서 인증서를 선택합니다. 	클라우드 관리자
리스너 규칙을 생성합니다.	<p>리스너 규칙을 생성하여 다음을 수행하십시오.</p> <ol style="list-style-type: none"> 호스트 api.domain.com을 API Gateway 대상 그룹에 포워딩합니다. 호스트 ui.domain.com을 UI 리소스의 대상 그룹에 전달합니다. 	클라우드 관리자

Route 53 구성

작업	설명	필요한 기술
프라이빗 호스팅 영역을 생성합니다.	프라이빗 호스팅 영역을 생성 하십시오.	클라우드 관리자
도메인 레코드를 생성합니다.	<p>CNAME 레코드 생성에서 다음을 수행합니다.</p> <ul style="list-style-type: none"> 값이 Application Load Balancer의 DNS 이름으로 설정된 API 	클라우드 관리자

작업	설명	필요한 기술
	<ul style="list-style-type: none"> 값이 Application Load Balancer의 DNS 이름으로 설정된 UI 	

API Gateway에서 프라이빗 API 엔드포인트 생성

작업	설명	필요한 기술
<p>프라이빗 API 엔드포인트를 생성하고 구성합니다.</p>	<ol style="list-style-type: none"> 프라이빗 API 엔드포인트를 생성하려면 API Gateway 설명서의 Amazon API Gateway에서 프라이빗 API 생성 지침을 따르십시오. VPC 엔드포인트에서 API에 대한 직접 호출만 허용하도록 리소스 정책을 구성합니다. 자세한 내용은 API Gateway 설명서에서 API Gateway 리소스 정책을 사용하여 API에 대한 액세스 제어를 참조하십시오. 	<p>앱 개발자, 클라우드 관리자</p>
<p>사용자 지정 도메인 이름을 사용하십시오.</p>	<ol style="list-style-type: none"> api.domain.com에 대한 사용자 지정 도메인 이름을 생성합니다. 자세한 내용은 API Gateway 설명서에서 REST API에 대한 사용자 지정 도메인 이름 설정을 참조하십시오. 생성된 API와 스테이지를 선택합니다. 자세한 내용은 API Gateway 설명서의 REST API용 API 매핑 작업을 참조하십시오. 	<p>클라우드 관리자</p>

관련 리소스

- [Amazon API Gateway](#)
- [Amazon Route 53](#)
- [Application Load Balancer](#)
- [AWS PrivateLink](#)
- [AWS Certificate Manager](#)

로컬 Angular QuickSight 애플리케이션에 Amazon 대시보드 내장

작성자: Sean Griffin (AWS) 및 Milena Godau (AWS)

환경: PoC 또는 파일럿

기술: 웹 및 모바일 앱, 분석

AWS 서비스: AWS Lambda, QuickSight 아마존, 아마존 API Gateway

요약

이 패턴은 개발 또는 테스트를 위해 Amazon QuickSight 대시보드를 로컬에 호스팅된 Angular 애플리케이션에 내장하기 위한 지침을 제공합니다. [내장된 분석 기능은](#) 이 기능을 기본적으로 QuickSight 지원하지 않습니다. 기존 대시보드가 있는 QuickSight 계정과 Angular에 대한 지식이 필요합니다.

내장된 QuickSight 대시보드를 사용할 때는 일반적으로 웹 서버에서 애플리케이션을 호스팅해야 대시보드를 볼 수 있습니다. 이렇게 하면 모든 것이 올바르게 작동하는지 확인하기 위해 변경 내용을 웹 서버에 지속적으로 푸시해야 하기 때문에 개발이 더 어려워집니다. 이 패턴은 로컬에 호스팅된 서버를 실행하고 QuickSight 내장된 분석을 사용하여 개발 프로세스를 더 쉽고 간소화하는 방법을 보여줍니다.

사전 조건 및 제한 사항

사전 조건

- [활성 Amazon Web Services\(AWS\) 계정](#)
- [세션 용량 QuickSight 요금이 부과되는 활성 계정](#)
- [QuickSight 임베디드 SDK 설치](#)
- [Angular CLI 설치됨](#)
- [Angular에 대한 지식](#)
- [mkcert 설치됨](#)

제한 사항

- 이 패턴은 ANONYMOUS (공개적으로 액세스할 수 있는) 인증 유형을 사용하여 QuickSight 대시보드를 내장하는 방법에 대한 지침을 제공합니다. 내장된 대시보드를 통한 AWS Identity 및 Access Management (IAM) 또는 QuickSight 인증을 사용하는 경우 제공된 코드는 적용되지 않습니다. 하지만 [에픽](#) 섹션에서 Angular 애플리케이션을 호스팅하는 단계는 여전히 유효합니다.

- ANONYMOUS 자격 증명 유형과 함께 GetDashboardEmbedUrlAPI를 사용하려면 QuickSight 용량 요금제가 필요합니다.

버전

- [Angular CLI 버전 13.3.4](#)
- [QuickSight SDK 버전 2.3.1 임베딩](#)

아키텍처

기술 스택

- Angular 프론트엔드
- AWS Lambda 및 Amazon API Gateway 백엔드

아키텍처

이 아키텍처에서 API Gateway의 HTTP API를 사용하면 로컬 Angular 애플리케이션이 Lambda 함수를 호출할 수 있습니다. Lambda 함수는 대시보드를 내장하기 위한 URL을 반환합니다. QuickSight

자동화 및 규모 조정

AWS CloudFormation 또는 AWS 서버리스 애플리케이션 모델 (AWS SAM) 을 사용하여 백엔드 배포를 자동화할 수 있습니다.

도구

도구

- [Angular CLI](#) 는 명령 셸에서 직접 Angular 애플리케이션을 초기화, 개발, 스캐폴딩 및 유지 관리하는 데 사용하는 명령줄 인터페이스 도구입니다.
- [QuickSight 임베딩 SDK는 HTML에](#) 대시보드를 내장하는 데 사용됩니다. QuickSight
- [mkcert](#)는 로컬에서 신뢰할 수 있는 개발 인증서를 생성할 수 있는 간단한 도구입니다. 구성이 필요하지 않습니다. 대시보드 임베딩을 위한 HTTPS 요청만 허용하므로 QuickSight mkcert가 필요합니다.

서비스

- [Amazon API Gateway](#)는 모든 규모에서 REST, HTTP 및 WebSocket API를 생성, 게시, 유지 관리, 모니터링 및 보호하는 AWS 서비스입니다.
- [AWS Lambda](#)는 서버를 프로비저닝하거나 관리하지 않고도 코드를 실행할 수 있게 해주는 컴퓨팅 서비스입니다. Lambda는 필요 시에만 코드를 실행하며, 일일 몇 개의 요청에서 초당 수천 개의 요청까지 자동으로 규모를 조정합니다. 사용한 컴퓨팅 시간만큼만 비용을 지불하고, 코드가 실행되지 않을 때는 요금이 부과되지 않습니다.
- [QuickSightAmazon](#)은 데이터에서 시각화를 구축하고, 임시 분석을 수행하고, 비즈니스 통찰력을 얻을 수 있는 비즈니스 분석 서비스입니다.

에픽

EmbedURL 생성

작업	설명	필요한 기술
정책을 생성하십시오. EmbedUrl	다음 속성을 가진 이름을 QuicksightGetDashboardEmbedUrl가진 IAM 정책을 생성합니다. <pre data-bbox="591 1108 1029 1881"> { "Version": "2012-10-17", "Statement": [{ "Effect": "Allow", "Action": ["quicksight:GetDashboardEmbedUrl", "quicksight:GetAnonymousUserEmbedUrl"], "Resource": "*" }] } </pre>	AWS 관리자

작업	설명	필요한 기술
	<pre>] }</pre>	

작업	설명	필요한 기술
Lambda 함수를 생성합니다.	<ol style="list-style-type: none"> 1. Lambda 콘솔에서 함수 페이지를 엽니다. 2. 함수 생성을 선택합니다. 3. 새로 작성을 선택합니다. 4. [함수 이름]에 get-qs-embed-url 을 입력합니다. 5. Runtime(런타임)에서 Python 3.9를 선택합니다. 6. 함수 생성을 선택합니다. 7. 코드 탭에서 Lambda 함수에 다음 코드를 복사합니다. <pre data-bbox="597 968 1027 1856"> import json import boto3 from botocore.exceptions import ClientError import time from os import environ qs = boto3.client('quicksight', region_name='us-east-1') sts = boto3.client('sts') ACCOUNT_ID = boto3.client('sts').get_caller_identity().get('Account') DASHBOARD_ID = environ['DASHBOARD_ID'] </pre>	앱 개발자

작업	설명	필요한 기술
	<pre> def getDashboardURL(accountId, dashboardId, quicksightNamespace, resetDisabled, undoRedoDisabled): try: response = qs.get_dashboard_embed_url(AwsAccountId = accountId, DashboardId = dashboardId, Namespace = quicksightNamespace, IdentityType = 'ANONYMOUS', SessionLifetimeInMinutes = 600, UndoRedoDisabled = undoRedoDisabled, ResetDisabled = resetDisabled) return response except ClientError as e: print(e) return "Error generating embeddedURL: " + str(e) def lambda_handler(event, context): url = getDashboardURL(ACCOUNT_ID, DASHBOARD_ID, "default", True, True) return {'EmbedUrl': url} </pre>	

작업	설명	필요한 기술
	<pre data-bbox="594 205 1024 384">'statusCode': 200, 'url': url }</pre> <p data-bbox="594 422 899 457">8. 배포를 선택합니다.</p>	
<p data-bbox="115 506 532 583">대시보드 ID를 환경 변수로 추가합니다.</p>	<p data-bbox="594 506 1016 632">Lambda 함수에 DASHBOARD_ID 를 환경 변수로 추가합니다.</p> <ol data-bbox="594 680 1024 1604" style="list-style-type: none"> <li data-bbox="594 680 1024 806">1. 구성 탭에서 환경 변수, 편집, 환경 변수 추가를 선택합니다. <li data-bbox="594 833 1024 911">2. 키 DASHBOARD_ID 을 가진 환경 변수를 추가합니다. <li data-bbox="594 938 1024 1541">3. 의 DASHBOARD_ID 값을 가져오려면 대시보드로 이동하여 QuickSight 브라우저의 URL 끝에 있는 UUID를 복사하십시오. 예를 들어 URL이 <code>https://us-east-1.quicksight.aws.amazon.com/session/dashboards/<dashboard-id></code> 인 경우 URL의 <code><dashboard-id></code> 일부를 키 값으로 지정하십시오. <li data-bbox="594 1568 899 1604">4. 저장을 선택합니다. 	<p data-bbox="1070 506 1203 541">앱 개발자</p>

작업	설명	필요한 기술
Lambda 함수에 대한 권한 추가	<p>Lambda 함수의 실행 역할을 수정하고 여기에 정책을 추가합니다 QuicksightGetDashboardEmbedUrl.</p> <ol style="list-style-type: none"> 구성 탭에서 권한을 선택한 다음, 역할 이름을 선택합니다. 정책 연결을 선택하고, QuicksightGetDashboardEmbedUrl 을 검색하고, 해당 확인란을 선택한 다음, 정책 연결을 선택합니다. 	앱 개발자

작업	설명	필요한 기술
<p>Lambda 함수를 테스트합니다.</p>	<p>테스트 이벤트를 생성하고 실행합니다. 함수가 테스트 이벤트의 데이터를 전혀 사용하지 않으므로 “Hello World” 템플릿을 사용할 수 있습니다.</p> <ol style="list-style-type: none"> 1. 테스트 탭을 선택합니다. 2. 테스트 이벤트 이름을 지정한 다음 저장을 선택합니다. 3. Lambda 함수를 테스트하려면, 테스트를 선택합니다. 응답은 다음과 같습니다. <pre data-bbox="592 863 1027 1262"> { "statusCode": 200, "url": "\"https://us-east-1.quicksight.aws.amazon.com/embed/f1acc0786687783b9a4543a05ba929b3a/dashboards/... </pre> <p>참고: 사전 요구 사항 및 제한 섹션에서 언급한 바와 같이, QuickSight 계정에는 세션 용량 요금 플랜이 적용되어야 합니다. 그렇지 않으면 이 단계에서 오류 메시지가 표시됩니다.</p>	<p>앱 개발자</p>

작업	설명	필요한 기술
<p>API Gateway에서 API를 생성합니다.</p>	<ol style="list-style-type: none"> 1. API 게이트웨이 콘솔에서 API 생성을 선택한 다음, REST API, 구축을 선택합니다. <ul style="list-style-type: none"> • API 이름에 <code>qs-embed-api</code> 를 입력합니다. • API 생성(Create API)을 선택합니다. 2. 작업에서 메서드 생성을 선택합니다. <ul style="list-style-type: none"> • 받기를 선택하고 체크 표시를 선택하여 확인합니다. • Lambda 함수를 통합 유형으로 선택합니다. • Lambda 함수에 <code>get-qs-embed-url</code> 을 입력합니다. • 저장을 선택합니다. • Lambda 함수에 대한 권한 추가 상자에서 확인을 선택합니다. 3. CORS를 활성화합니다. <ul style="list-style-type: none"> • 작업에서 CORS 활성화를 선택합니다. • Access-Control-Allow-Origin에는 <code>'https://my-qs-app.net:4200'</code> 을 입력합니다. 	<p>앱 개발자</p>

작업	설명	필요한 기술
	<ul style="list-style-type: none"> CORS 활성화 및 기존의 CORS 헤더 대체를 선택하고 확인합니다. <p>4. API를 배포합니다.</p> <ul style="list-style-type: none"> 작업 및 API 배포를 선택합니다. 배포 단계에서 [새 단계]를 선택합니다. 단계 이름에 dev를 입력합니다. [배포]를 선택합니다. URL 호출을 복사합니다. <p>참고: my-qs-app.net 은 어떠한 도메인이라도 될 수 있습니다. 다른 도메인 이름을 사용하려면 3단계에서 Access-Control-Allow-Origin 정보를 업데이트하고 이후 단계에서 my-qs-app.net 을 변경해야 합니다.</p>	

Angular 애플리케이션을 생성합니다.

작업	설명	필요한 기술
Angular CLI를 사용하여 애플리케이션을 생성합니다.	<p>1. 애플리케이션을 생성합니다.</p> <pre data-bbox="630 1686 1029 1885">ng new quicksight-app --defaults cd quicksight-app/src /app</pre>	앱 개발자

작업	설명	필요한 기술
	<p>2. 대시보드 구성 요소를 생성합니다.</p> <pre>ng g c dashboard</pre> <p>3. <code>src/environments/environment.ts</code> 파일로 이동하여 환경 개체에 <code>apiUrl: '<Invoke URL from previous steps>'</code>를 추가합니다.</p> <pre>export const environment = { production: false, apiUrl: '<Invoke URL from previous steps>', };</pre>	
QuickSight 임베딩 SDK를 추가하세요.	<p>1. 프로젝트의 루트 폴더에서 다음 명령어를 실행하여 QuickSight 임베딩 SDK를 설치합니다.</p> <pre>npm i amazon-quicksight-embedding-sdk</pre> <p>2. <code>src</code> 폴더 안에 다음 내용으로 새 <code>decl.d.ts</code> 파일을 만듭니다.</p> <pre>declare module 'amazon-quicksight-embedding-sdk';</pre>	앱 개발자

작업	설명	필요한 기술
<p>dashboard.component.ts 파일에 코드를 추가합니다.</p>	<pre>import { Component, OnInit } from '@angular /core'; import { HttpClient } from '@angular/common/ http'; import * as Quicksigh tEmbedding from 'amazon-quicksight- embedding-sdk'; import { environme nt } from "../..en vironments/envirom ent"; import { take } from 'rxjs'; import { Embedding Context } from 'amazon- quicksight-embedding- sdk/dist/types'; import { createEmb eddingContext } from 'amazon-quicksight- embedding-sdk'; @Component({ selector: 'app-dash board', templateUrl: './ dashboard.compo nent.html', styleUrls: ['./dashb oard.component.scss'] }) export class Dashboard Component implements OnInit { constructor(private http: HttpClient) { }</pre>	<p>앱 개발자</p>

작업	설명	필요한 기술
	<pre> loadingError = false; dashboard: any; ngOnInit() { this.GetDashboardURL(); } public GetDashboardURL() { this.http.get(environment.apiUrl) .pipe(take(1),) .subscribe((data: any) => this.Dashboard(data.url)); } public async Dashboard(embeddedURL: any) { var containerDiv = document.getElementById("dashboardContainer") ''; const frameOptions = { url: embeddedURL, container: containerDiv, height: "850px", width: "100%", resizeModeOnSizeChangeEvent: true, } const embeddingContext: EmbeddingContext = await createEmbeddingContext(); </pre>	

작업	설명	필요한 기술
	<pre> this.dashboard = embeddingContext.e mbedDashboard(fram eOptions); } } </pre>	
<p>dashboard.component.html 파일에 코드를 추가합니다.</p>	<p>다음 코드를 src/app/dashboard/dashboard.component.html 파일에 추가합니다.</p> <pre> <div id="dashboardConta iner"></div> </pre>	<p>앱 개발자</p>
<p>app.component.html 파일을 수정하여 대시보드 구성 요소를 로드하십시오.</p>	<ol style="list-style-type: none"> 1. src/app/app.component.html 파일 내용을 삭제합니다. 2. 다음을 추가합니다. <pre> <app-dashboard></a pp-dashboard> </pre>	<p>앱 개발자</p>
<p>HttpClientModule app.module.ts 파일로 가져옵니다.</p>	<ol style="list-style-type: none"> 1. src/app/app.module.ts 파일 상단에 다음을 추가합니다. <pre> import { HttpClien tModule } from '@angular/common/h ttp'; </pre> <ol style="list-style-type: none"> 2. AppModule에 대해 imports에서 HttpClientModule을 추가합니다. 	<p>앱 개발자</p>

Angular 애플리케이션 호스팅

작업	설명	필요한 기술
mkcert를 구성합니다.	<p>참고: 다음 명령은 Unix 또는 MacOS 시스템용입니다. Windows를 사용하는 경우 해당 echo 명령에 대한 추가 정보 섹션을 참고하십시오.</p> <ol style="list-style-type: none"> 1. 컴퓨터에 로컬 인증 기관 (CA)을 생성합니다. <pre>mkcert -install</pre> <ol style="list-style-type: none"> 2. 항상 로컬 PC로 리디렉션되도록 my-qs-app.net 을 구성합니다. <pre>echo "127.0.0.1 my-qs-app.net" sudo tee -a /private/etc/hosts</pre> <ol style="list-style-type: none"> 3. 현재 위치가 Angular 프로젝트의 src 디렉터리에 있는지 확인합니다. <pre>mkcert my-qs-app.net 127.0.0.1</pre>	앱 개발자
도메인을 허용하도록 QuickSight 구성하세요.	<ol style="list-style-type: none"> 1. 에서 QuickSight 오른쪽 상단에서 이름을 선택한 다음 Quicksight 관리를 선택합니다. 2. 도메인 및 임베딩으로 이동합니다. 	AWS 관리자

작업	설명	필요한 기술
	<p>3. 허용된 도메인으로 <code>https://my-qs-app.net:4200</code> 을 추가합니다.</p>	
<p>솔루션을 테스트합니다.</p>	<p>다음 명령을 실행하여 로컬 Angular 개발 서버를 시작합니다.</p> <pre data-bbox="594 554 1027 833">ng serve --host my-qs-app.net --port 4200 --ssl --ssl-key "./src/my-qs-app.net-key.pem" --ssl-cert "./src/my-qs-app.net.pem" -o</pre> <p>이렇게 하면 이전에 생성한 사용자 지정 인증서로 SSL(Secure Sockets Layer)을 사용할 수 있습니다.</p> <p>빌드가 완료되면 브라우저 창이 열리고 Angular에서 로컬로 호스팅되는 임베디드 QuickSight 대시보드를 볼 수 있습니다.</p>	<p>앱 개발자</p>

관련 리소스

- [Angular 웹사이트](#)
- [익명 \(미등록\) 사용자를 위한 QuickSight 데이터 대시보드 내장 \(문서\)](#) QuickSight
- [QuickSight SDK 임베딩](#)
- [mkcert 도구](#)

추가 정보

Windows를 사용하는 경우, 관리자 권한으로 명령 프롬프트 창을 실행하고 다음 명령을 사용하여 항상 로컬 PC로 리디렉션되도록 `my-qs-app.net`을 구성합니다.

```
echo 127.0.0.1 my-qs-app.net >> %WINDIR%\System32\Drivers\Etc\Hosts
```

패턴 더 보기

- [Amazon Cognito 자격 증명 풀을 사용하여 ASP.NET Core 앱에서 AWS 서비스에 액세스](#)
- [AWS Fargate, PrivateLink AWS 및 Network Load Balancer를 사용하여 Amazon ECS에서 비공개로 컨테이너 애플리케이션에 액세스할 수 있습니다.](#)
- [PrivateLink AWS와 Network Load Balancer를 사용하여 Amazon ECS에서 비공개로 컨테이너 애플리케이션에 액세스할 수 있습니다.](#)
- [를 사용하여 마이그레이션 전략 식별 및 계획을 자동화합니다. AppScore](#)
- [DevOps 프랙티스와 AWS Cloud9를 사용하여 마이크로서비스로 느슨하게 연결된 아키텍처를 구축하십시오.](#)
- [AWS Amplify를 사용하여 서버리스 리액트 네이티브 모바일 앱을 구축하십시오.](#)
- [AWS CodeCommit, AWS 및 AWS Device Farm을 사용하여 iOS 앱을 구축하고 테스트할 수 있습니다. CodePipeline](#)
- [CloudWatch NLog를 사용하여 Amazon Logs에서.NET 애플리케이션에 대한 로깅을 구성합니다.](#)
- [HashiCorp 패커를 사용하여 CodePipeline 파이프라인 및 AMI 생성](#)
- [파이프라인을 생성하고 다음을 사용하여 온프레미스 EC2 인스턴스에 아티팩트 업데이트를 배포합니다. CodePipeline](#)
- [Amazon ECS 작업 정의를 생성하고 Amazon EFS를 사용하여 EC2 인스턴스에 파일 시스템을 마운트](#)
- [Amazon EKS 클러스터에 gRPC 기반 애플리케이션을 배포하고 Application Load Balancer를 사용하여 액세스하기](#)
- [테라폼을 사용하여 CloudWatch Synthetics 카나리아를 배포하세요](#)
- [Amazon ECR과 AWS Fargate를 사용하여 Amazon ECS에 Java 마이크로서비스 배포](#)
- [Amazon ECR 및 로드 밸런싱을 사용하여 Amazon ECS에 Java 마이크로서비스 배포](#)
- [Fargate를 사용하여 Amazon ECS에 Java 마이크로서비스 배포](#)
- [AWS Amplify에서 호스팅하는 웹 애플리케이션에 대해 AWS WAF 활성화하기](#)
- [Green Boost를 통한 풀스택 클라우드 네이티브 웹 애플리케이션 개발 살펴보기](#)
- [메시지 대기열을 Microsoft Azure 서비스 버스에서 Amazon SQS로 마이그레이션](#)
- [Microsoft Azure 앱 서비스의 .NET 애플리케이션을 AWS Elastic Beanstalk로 마이그레이션합니다.](#)
- [이진법을 사용하여 온프레미스 Go 웹 애플리케이션을 AWS Elastic Beanstalk로 마이그레이션](#)
- [SFTP용 AWS Transfer를 사용하여 온프레미스 SFTP 서버를 AWS로 마이그레이션](#)
- [IBM WebSphere 애플리케이션 서버에서 Amazon EC2의 아파치 톰캣으로 마이그레이션](#)

- [Auto Scaling을 사용하여 IBM WebSphere 애플리케이션 서버에서 Amazon EC2의 아파치 톰캣으로 마이그레이션하세요](#)
- [오라클에서 AWS Elastic GlassFish Beanstalk로 마이그레이션](#)
- [AWS App2Container를 사용하여 온프레미스 Java 애플리케이션을 AWS로 마이그레이션](#)
- [OpenText TeamSite 워크로드를 AWS 클라우드로 마이그레이션](#)
- [ACM을 사용하여 Windows SSL 인증서를 Application Load Balancer로 마이그레이션](#)
- [AWS에서 ASP.NET Web Forms 애플리케이션 현대화](#)
- [Amazon EC2 리눅스 인스턴스에서 ASP.NET 코어 웹 API Docker 컨테이너를 실행합니다.](#)
- [Amazon을 사용하여 VPC를 통해 Amazon S3 버킷의 정적 콘텐츠를 제공합니다. CloudFront](#)
- [AWS에서 고가용성 PeopleSoft 아키텍처 설정](#)
- [Network Firewall을 사용하여 아웃바운드 트래픽의 Server Name Indication\(SNI\)에서 DNS 도메인 이름 캡처](#)
- [Flask와 Elastic Beanstalk를 사용하여 AI/ML 모델 결과를 시각화합니다.](#)

기계 번역으로 제공되는 번역입니다. 제공된 번역과 원본 영어의 내용이 상충하는 경우에는 영어 버전이 우선합니다.