FleetIQ 개발자 가이드

Amazon GameLift

버전

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

Amazon GameLift: FleetIQ 개발자 가이드

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

Amazon의 상표 및 브랜드 디자인은 Amazon 외 제품 또는 서비스와 함께, Amazon 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

Amazon GameLift FleetIQ란 무엇인가요? 1
- FleetIQ 작동 방식 2
  - Amazon GameLift FleetIQ 로직 2
  - 주요 리소스 및 구성 요소 5
- 게임 아키텍처 7
  - 온프레미스 호스팅 보완 7
- 수명 주기 및 상태 9
  - 스팟 밸런싱 프로세스 9
  - 게임 서버 그룹의 수명 11
  - 게임 서버의 수명 13
- 모범 사례 15
- Amazon GameLift FleetIQ 기능 17
- Amazon GameLift FleetIQ 요금 18

설정 19
- 지원 소프트웨어 19
- AWS 계정 설정 20
  - 생성 AWS 계정 20
  - Amazon GameLift FleetIQ에 대한 사용자 권한 관리 22
  - 교차 서비스 상호 작용을 위한 IAM 역할 생성 26

FleetIQ에 사용되는 게임 준비 33
- 통합 단계 33
- 게임 서버 그룹 관리 35
  - 게임 서버 그룹 생성 35
  - 게임 서버 그룹 업데이트 36
  - 게임 서버 그룹 인스턴스 추적 36
- 게임 서버 통합 37
  - 게임 서버 등록 37
  - 게임 서버 상태 업데이트 37
  - 게임 서버 등록 취소 38
- 게임 클라이언트 통합 38
  - Amazon GameLift FleetIQ가 게임 서버를 선택하도록 허용 39
  - 자체 게임 서버 선택 40

CloudWatch를 사용하여 모니터링 41

FleetIQ를 사용한 보안 44

Amazon GameLift FleetIQ 참조 45
서비스 API 참조(AWS SDK) 45
Amazon GameLift FleetIQ API 작업 45
사용 가능한 프로그래밍 언어 46
릴리스 정보 및 SDK 버전 47
모든 Amazon GameLift 가이드 47
AWS 용어집 48
..... xlix

Amazon GameLift FleetIQ란 무엇인가요?

Amazon GameLift FleetIQ는 클라우드 기반 게임 호스팅용 Amazon Elastic Compute Cloud(Amazon EC2) 스팟 인스턴스를 저렴한 비용으로 사용하도록 최적화합니다. Amazon GameLift FleetIQ를 사용하면 Amazon EC2 및 Amazon EC2 Auto Scaling의 호스팅 리소스를 직접 사용하는 동시에 Amazon GameLift의 최적화를 활용하여 플레이어에게 저렴하고 탄력적인 게임 호스팅을 제공할 수 있습니다. Amazon EC2 스팟 인스턴스는 대폭 할인된 가격으로 제공되기는 하지만 가용성이 변동하고 중단될 가능성이 있기 때문에 일반적으로 게임 호스팅에는 적합하지 않습니다. Amazon GameLift FleetIQ는 이러한 제한을 크게 완화하여 게임 호스팅에 저렴한 스팟 인스턴스를 사용할 수 있도록 합니다.

FleetIQ 최적화는 Amazon GameLift를 사용하여 게임 호스팅을 관리할 때도 사용할 수 있습니다. Amazon GameLift 호스팅 옵션에 대한 자세한 내용은 Amazon GameLift 개발자 가이드를 참조하세요.

Amazon GameLift FleetIQ 게임 호스팅 솔루션은 다음을 수행하려는 게임 개발자를 위해 설계되었습니다.

- 기존 AWS 배포가 있거나 완전 관리형 Amazon GameLift 서비스를 사용하지 않고 Amazon EC2를 직접 사용하고자 합니다. Amazon GameLift FleetIQ는 AWS 계정 사용자가 관리하는 EC2 오토 스케일링과 함께 작동하여 EC2 인스턴스 및 그룹에 대한 전체 액세스 권한을 제공합니다. Amazon Elastic Container Service(Amazon ECS), Amazon Elastic Kubernetes Service(Amazon EKS), AWS Shield Advanced를 포함한 기타 AWS 서비스와 통합할 수도 있습니다.
- 기존 온프레미스 게임 호스팅이 있고 클라우드로 용량을 확장하려는 고객. Amazon GameLift FleetIQ를 사용하면 온프레미스 용량을 활용하면서 필요에 따라 AWS 클라우드 용량을 점진적으로 추가할 수 있는 하이브리드 배포 시스템을 구축할 수 있습니다.

Amazon GameLift FleetIQ를 사용할 준비가 되셨나요?

- AWS 스킬 빌더의 게임 서버용 Amazon GameLift FleetIQ 사용을 수강하여 게임에 Amazon GameLift FleetIQ를 사용하는 방법에 대해 알아봅니다. 관련 과정에 대한 개요는 Game Tech 학습 계획을 참조하세요. 일부 코스는 다양한 언어로 사용할 수 있습니다.
- Amazon GameLift FleetIQ 통합 단계의 지침을 따르세요.

Amazon GameLift FleetIQ 작동 방식

Amazon GameLift FleetIQ 솔루션은 Amazon EC2 및 Auto Scaling으로 얻을 수 있는 전체 컴퓨팅 리소스 관리 도구 세트를 보완하는 게임 호스팅 계층입니다. Amazon GameLift FleetIQ는 게임 호스팅에 고유한 기능 목록을 제공하는 것 외에도 게임 호스팅에 저렴한 스팟 인스턴스를 사용할 수 있는 추가 로직 계층을 제공합니다. 이 솔루션을 사용하면 Amazon EC2 및 Auto Scaling 리소스를 직접 관리하고 필요에 따라 다른 AWS 서비스와 통합할 수 있습니다.

Amazon GameLift FleetIQ 를 사용하는 경우 게임 서버 소프트웨어로 Amazon Machine Image(AMI)를 만들고, Amazon EC2 시작 템플릿을 생성하고, Auto Scaling 그룹에 대한 구성 설정을 정의하는 등 Amazon EC2 인스턴스를 평소와 같이 시작할 준비를 합니다. 그러나 Auto Scaling 그룹을 직접 생성하는 대신 Amazon EC2 및 Auto Scaling 리소스 및 구성을 사용하여 Amazon GameLift FleetIQ 게임 서버 그룹을 생성합니다. 이 작업을 수행하면 Amazon GameLift FleetIQ가 게임 서버 그룹과 해당 Auto Scaling 그룹을 모두 생성하라는 메시지가 표시됩니다. 게임 서버 그룹은 오토 스케일링에 연결되어 있으며 오토 스케일링의 특정 부분을 관리합니다.

Auto Scaling 그룹이 생성되면 Amazon EC2 및 Auto Scaling 리소스에 대한 전체 액세스 권한을 갖게 됩니다. Auto Scaling 그룹의 구성을 변경하고, 다중 수준 조정 정책 또는 로드 밸런서를 추가하고, 다른 AWS 서비스와 통합할 수 있습니다. 그룹 내 인스턴스에 직접 연결할 수 있습니다. 최적화 로직의 일부로 Amazon GameLift FleetIQ는 특정 Auto Scaling 그룹 속성도 주기적으로 업데이트합니다. 오토 스케일링에서 배포한 모든 인스턴스의 가용성 상태를 추적할 수 있습니다.

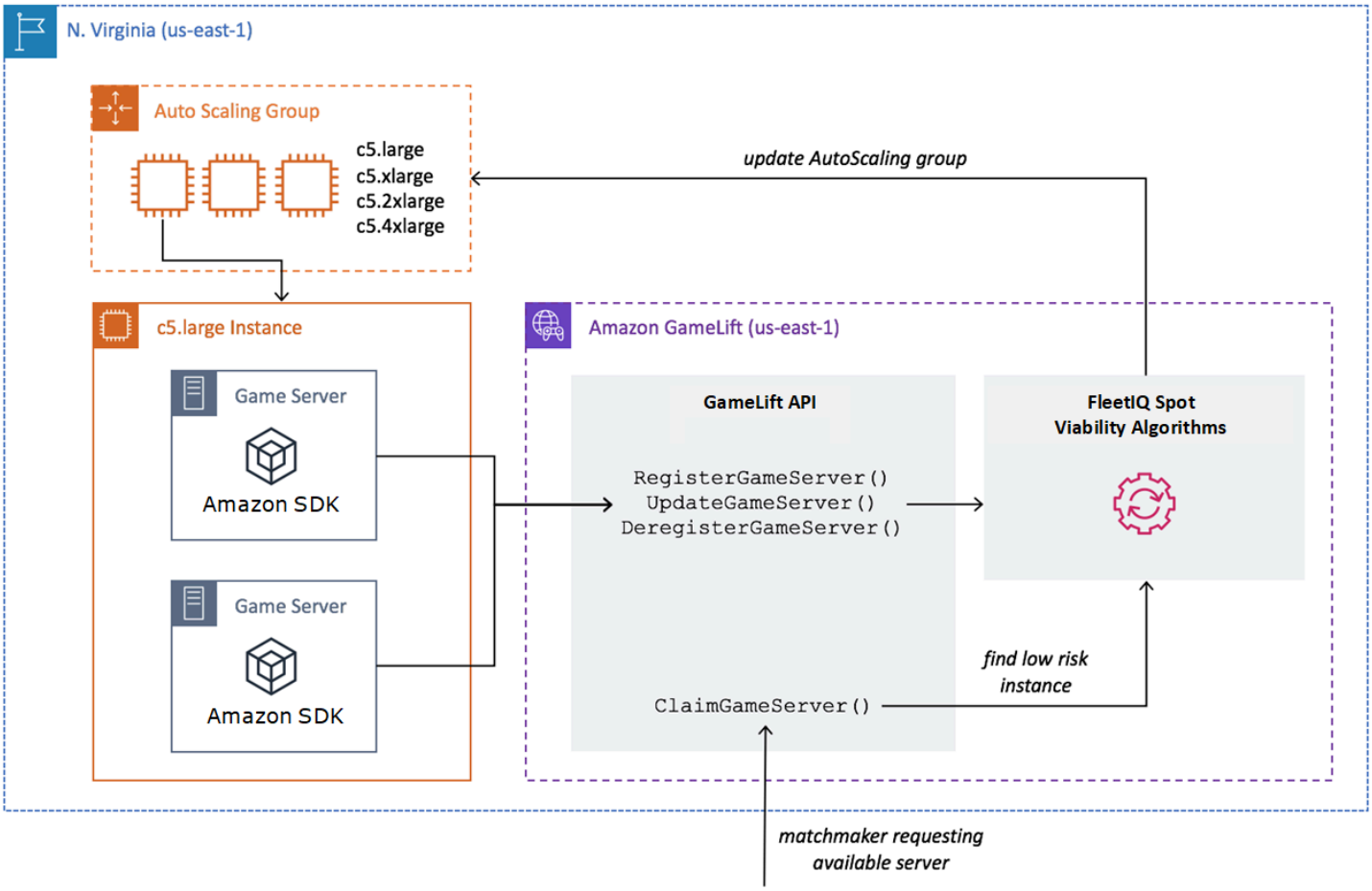
언제든지 게임 서버 그룹에 대한 Amazon GameLift FleetIQ 활동을 일시적으로 중지할 수 있습니다. 또한 게임 서버 그룹을 삭제하지만 해당 오토 스케일링은 유지할 수 있는 옵션도 있습니다.

주제

- Amazon GameLift FleetIQ 로직
- 주요 리소스 및 구성 요소

Amazon GameLift FleetIQ 로직

다음 다이어그램은 게임 호스팅을 EC2 위해 Amazon과 함께 작업할 때 Amazon GameLift FleetIQ의 역할을 보여줍니다. 주된 목표는 게임 세션을 호스팅하는 데 최고의 적정 게임 서버를 찾고 플레이어에게 최적의 게임 플레이 경험을 제공하는 것입니다. Amazon GameLift FleetIQ는 가장 저렴한 비용으로 가장 높은 게임 호스팅 실행 가능성을 제공하는 리소스로 최고의 리소스를 정의합니다. Amazon GameLift FleetIQ는 두 가지 주요 방식으로 이 목표에 접근합니다. 첫째는 Auto Scaling 그룹에서 실행 가능한 인스턴스 유형만 허용하고, 둘째는 그룹의 사용 가능한 리소스 전체에 새 게임 세션을 효과적으로 배치하는 것입니다.



오토 스케일링을 최적의 인스턴스 유형으로 채우기

오토 스케일링의 임무는 새 인스턴스를 시작하고 이전 인스턴스를 사용 중지하면서 호스팅 리소스 모음을 유지하고 플레이어 수요에 맞게 조정하는 것입니다. 이를 위해 오토 스케일링은 원하는 인스턴스 유형 목록을 사용합니다. Amazon GameLift FleetIQ의 작업은 이러한 원하는 인스턴스 유형의 실행 가능성을 지속적으로 확인하고 Auto Scaling 그룹의 목록을 업데이트하는 것입니다. 이 프로세스를 인스턴스 밸런싱이라고 합니다. 이를 통해 오토 스케일링의 인스턴스가 지속적으로 새로 고쳐져 현재 실행 가능한 인스턴스 유형만 항상 사용됩니다.

Amazon GameLift FleetIQ는 Auto Scaling 그룹이 다음과 같은 방식으로 최적의 인스턴스 유형을 선택하는 방식에 영향을 미칩니다.

- 스팟 및/또는 온디맨드 인스턴스의 사용량을 결정합니다. Amazon GameLift FleetIQ 게임 서버 그룹은 Auto Scaling 그룹이 스팟 및/또는 온디맨드 인스턴스를 사용하는 방식에 영향을 미치는 밸런싱 전략으로 구성됩니다. 스팟 인스턴스는 가용성 변동 및 잠재적 중단, Amazon GameLift FleetIQ가 게임 서버 호스팅에 대해 최소화하는 제한으로 인해 비용이 절감됩니다. 온디맨드 인스턴스는 비용이 더 많이 들지만 필요할 때 더 안정적인 가용성을 제공합니다.

- 새 인스턴스를 실행 가능한 인스턴스 유형에서만 시작할 수 있도록 제한합니다. Amazon GameLift FleetIQ 게임 서버 그룹은 원하는 인스턴스 유형의 마스터 목록을 유지합니다. 인스턴스 밸런싱 프로세스는 인스턴스 유형의 최근 가용성 및 중단율을 확인하는 예측 알고리즘을 사용하여 게임 호스팅 실행 가능성에 대해 목록에서 원하는 각 인스턴스 유형을 지속적으로 평가합니다. 이 평가 결과 Amazon GameLift FleetIQ는 현재 사용 가능한 인스턴스 유형만 포함하도록 Auto Scaling 그룹의 원하는 인스턴스 유형 목록을 지속적으로 업데이트합니다.
- 실행 불가능한 인스턴스 유형인 기존 인스턴스에 플래그를 지정합니다. Amazon GameLift FleetIQ는 Auto Scaling 그룹의 기존 인스턴스를 식별합니다. 이 인스턴스는 현재 생존할 수 없는 인스턴스 유형입니다. 이러한 인스턴스에는 드레이닝이라는 플래그가 지정되어 있습니다. 이는 해당 인스턴스가 종료되고 새 인스턴스로 교체됨을 의미합니다. 게임 서버 보호 기능이 설정되어 있는 인스턴스의 경우 모든 활성 게임 세션이 정상적으로 종료될 때까지 종료가 연기됩니다.

저비용 스팟 인스턴스 유형의 가용성이 변동되더라도 오토 스케일링은 인스턴스를 시작하고 만료할 때 게임 호스팅에 최적화된 모음을 유지합니다. 밸런싱 활동은 활성 인스턴스가 있는 게임 서버 그룹에서만 이루어집니다. 이 프로세스의 작동 방식에 대해 스팟 밸런싱 프로세스에서 자세히 알아보십시오.

효과적으로 게임 세션 배치

Amazon GameLift FleetIQ는 게임 서버 그룹의 모든 활성 게임 서버를 추적하고 이 정보를 사용하여 새 게임 세션 및 플레이어에 가장 적합한 위치를 결정합니다.

Amazon GameLift FleetIQ가 게임 서버를 추적할 수 있도록 하려면 게임 서버 소프트웨어가 상태를 보고해야 합니다. 사용자 지정은 각 인스턴스에서 새 게임 서버 프로세스가 시작되고 중지되는 방식을 AMI 제어합니다. 새 게임 서버가 시작되면 Amazon GameLift FleetIQ 에 등록하여 게임 세션을 호스팅할 준비가 되었음을 나타냅니다. 등록 후 게임 서버는 주기적으로 자체 상태와 현재 게임 세션을 호스팅하고 있는지 여부를 보고합니다. 게임 서버가 종료되면 Amazon GameLift FleetIQ 에 대한 등록이 취소됩니다.

새 게임 세션을 시작하려면 게임 클라이언트(또는 매치메이커 또는 기타 클라이언트 서비스)가 Amazon GameLift FleetIQ 에 게임 서버에 대한 요청을 보냅니다. Amazon GameLift FleetIQ는 사용 가능한 게임 서버를 찾아서 새 게임 세션에 대해 클레임하고 게임 서버 ID 및 연결 정보로 응답합니다. 그런 다음 게임 서버의 상태가 업데이트되고 들어오는 플레이어를 위한 새 게임 세션이 시작됩니다.

새 게임 세션을 호스팅할 게임 서버를 선택할 때 Amazon GameLift FleetIQ는 다음과 같은 의사 결정 프로세스를 사용하여 실행 가능한 저비용 스팟 인스턴스를 사용하여 배치를 최적화합니다.

1. 가능하면 Amazon GameLift FleetIQ는 이미 다른 게임 세션을 호스팅하고 있는 인스턴스에 새 게임 세션을 배치합니다. 일부 인스턴스로 압축하고(과부하시키는 않음) 다른 인스턴스를 유휴 상태로 유

지함으로써 오토 스케일링이 필요 없을 때 유휴 인스턴스를 빠르게 축소할 수 있으므로 호스팅 비용을 절감할 수 있습니다.

2. Amazon GameLift FleetIQ는 드레이닝 으로 플래그가 지정된 인스턴스, 즉 게임 호스팅에 사용할 수 없는 인스턴스를 무시합니다. 이러한 인스턴스는 기존 게임 세션을 지원하기 위해서만 계속 실행됩니다. 이러한 인스턴스는 다른 게임 서버를 사용할 수 없으면 새 게임 세션에 사용할 수 없습니다.
3. Amazon GameLift FleetIQ는 실행 가능한 인스턴스에서 실행 중인 사용 가능한 모든 게임 서버를 식별합니다.

게임 서버 그룹에 대한 게임 세션 보호 기능을 설정하여 오토 스케일링이 활발하게 실행 중인 게임 세션이 있는 인스턴스를 종료하지 못하게 할 수 있습니다.

주요 리소스 및 구성 요소

Amazon GameLift FleetIQ 를 사용하여 게임 호스팅 리소스를 설정하기 전에 AWS 계정에 다음 리소스를 생성합니다. 게임 서버 그룹을 통해 사용하기 전에 이러한 리소스를 사용하여 게임 서버 배포를 개발하고 테스트하는 것이 좋습니다.

- Amazon Machine Image(AMI). AMI 는 Amazon EC2 인스턴스로 시작하려는 특정 소프트웨어 구성의 템플릿입니다. 게임 호스팅의 경우 에는 운영 체제, 게임 서버 바이너리 또는 컨테이너, 게임 서버에 필요한 기타 런타임 소프트웨어가 AMI 포함됩니다. 생성에 대한 자세한 내용은 Amazon 사용 설명서의 Amazon Machine 이미지를 AMI참조하세요. EC2 AMIs 는 리전별로 다릅니다. Amazon EC2 사용 설명서의 복사에 설명된 대로 한 리전AMI에서 다른 리전으로 를 복사AMIs할 수 있습니다.
- Amazon EC2 시작 템플릿. 시작 템플릿은 오토 스케일링에서 인스턴스를 시작하고 관리하는 것에 대한 지침을 제공합니다. 를 지정하고AMI, 적절한 인스턴스 유형 목록을 제공하며, 네트워크, 보안 및 기타 속성을 설정합니다. 시작 템플릿 생성에 대한 자세한 내용은 Amazon EC2 사용 설명서의 시작 템플릿에서 인스턴스 시작을 참조하세요. 시작 템플릿은 리전별로 고유합니다.
- AWS IAM 역할. IAM 역할은 AWS 리소스에 대한 제한된 액세스를 허용하는 권한 집합을 정의합니다. 다른 AWS 서비스와 같은 신뢰할 수 있는 엔터티는 역할을 수임하고 권한을 상속할 수 있습니다. Amazon GameLift FleetIQ 를 사용할 때는 Amazon GameLift FleetIQ가 AWS 계정에서 Auto Scaling 그룹 및 EC2 인스턴스 리소스를 생성하고 액세스할 수 있도록 허용하는 관리형 정책을 IAM 역할에 제공해야 합니다. IAM 역할은 리전별로 다릅니다.

Amazon GameLift FleetIQ는 다음 리소스를 직접 관리하고 이에 대한 직접적인 권한을 갖습니다.

- GameLift 게임 서버 그룹 . 게임 서버 그룹에는 Amazon GameLift FleetIQ가 해당 Auto Scaling 그룹과 협력하여 저렴한 게임 호스팅을 제공하는 방법을 정의하는 구성 설정이 포함되어 있습니다. 게임

서버 그룹은 리전별로 고유합니다. 리전에서 게임 서버 그룹을 생성하면 동일한 리전의 AWS 계정에 새 Auto Scaling 그룹이 자동으로 생성됩니다. 게임 서버 그룹은 Auto Scaling 그룹에 연결되며 (IAM 역할을 수임하여) 일부 설정을 관리하고 수정할 수 있습니다. 게임 서버 그룹은 수명이 긴 리소스이므로 개발자가 자주 생성할 필요가 없습니다. 게임 서버 그룹은 Auto Scaling 그룹의 인스턴스에서 호스팅되고 Amazon GameLift FleetIQ 에 등록된 게임 서버에 대한 기능 그룹화 리소스이기도 합니다.

- GameLift 게임 서버. 게임 서버 리소스는 Amazon GameLift FleetIQ 게임 서버 그룹과 연결된 인스턴스에서 실행 중인 게임 실행을 나타냅니다. 이 리소스는 게임 서버가 Amazon GameLift FleetIQ에 등록하고 게임 서버 그룹을 식별할 때 생성됩니다. Amazon GameLift FleetIQ는 등록된 각 게임 서버의 사용률 상태와 클레임 상태를 추적하여 게임 서버 가용성을 모니터링할 수 있습니다. 게임 서버는 리전별 게임 서버 그룹과 연결되어 있다는 점에서 리전별로 고유합니다. 게임에서 새 게임 서버를 요청할 때 게임 서버 그룹 및 리전을 지정합니다.

이러한 리소스는 Amazon GameLift FleetIQ 리소스를 통해 생성됩니다. 이 리소스는 AWS 계정에서 생성되며 사용자가 완벽하게 제어할 수 있습니다.

- Amazon EC2 Auto Scaling 그룹. Auto Scaling 그룹은 EC2 인스턴스 컬렉션을 시작하고 관리하며 그룹 용량을 자동으로 조정합니다. Amazon GameLift FleetIQ 에서는 게임 서버 그룹과 Auto Scaling 그룹 간에 one-to-one 관계가 있습니다. Auto Scaling 그룹의 모든 설정을 업데이트할 수 있지만 Amazon GameLift FleetIQ는 주기적으로 로직의 일부로 특정 설정을 재정의하고 업데이트하여 게임 호스팅 실행 가능성을 위한 스팟 인스턴스의 균형을 유지합니다. 자세한 내용은 Amazon Auto Scaling 사용 설명서 AutoScalingGroup의 섹션을 참조하세요. EC2 Auto Scaling 오토 스케일링은 리전별로 고유하며 게임 서버 그룹과 동일한 리전에 생성됩니다.
- Amazon EC2 인스턴스. 인스턴스는 클라우드의 가상 서버입니다. 인스턴스 유형에는 컴퓨팅, 메모리, 디스크 및 네트워크 리소스를 지정하는 특정 하드웨어 구성이 있습니다. 일반적으로 를 사용하여 Auto Scaling 그룹에서 시작합니다AMI. 인스턴스는 가용성에 따라 스팟 또는 온디맨드일 수 있습니다. Amazon GameLift FleetIQ 를 사용하면 인스턴스가 하나 이상의 게임 서버 프로세스를 실행하며, 각 프로세스는 여러 게임 세션을 호스팅할 수 있습니다. 인스턴스는 리전별 오토 스케일링과 연결되어 있다는 점에서 리전별로 고유합니다.

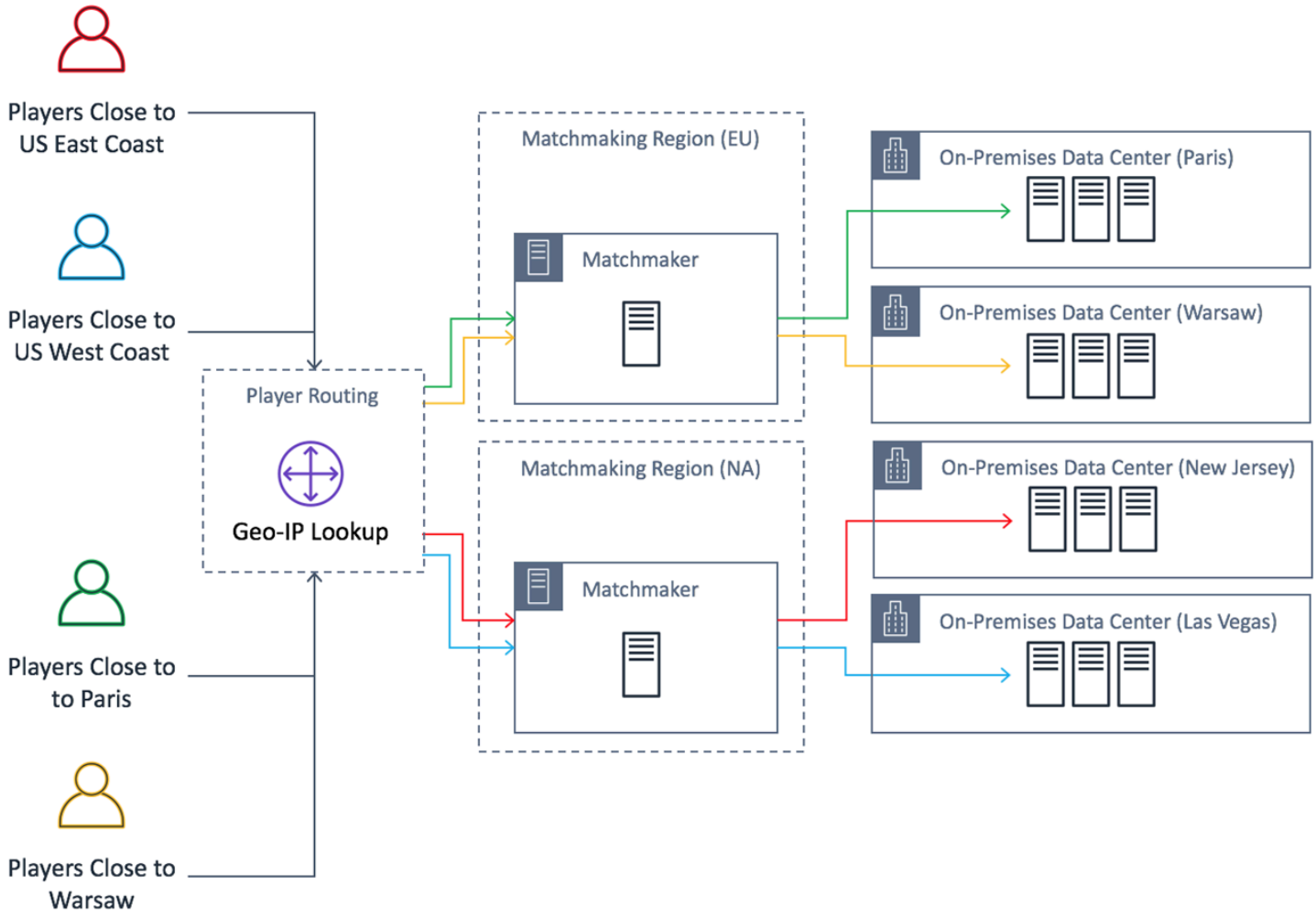
Amazon GameLift FleetIQ를 사용한 게임 아키텍처

온프레미스 호스팅 보완

Amazon GameLift FleetIQ는 이미 보유하고 있는 플레이어 지리 IP 라우팅, 매치메이킹, 로비 서비스를 비롯한 기존 게임 백엔드를 재사용할 수 있도록 설계되었습니다. 다음 예에서는 기존 온프레미스 배포에 Amazon GameLift FleetIQ를 어떻게 사용할 수 있는지 보여 줍니다.

Example

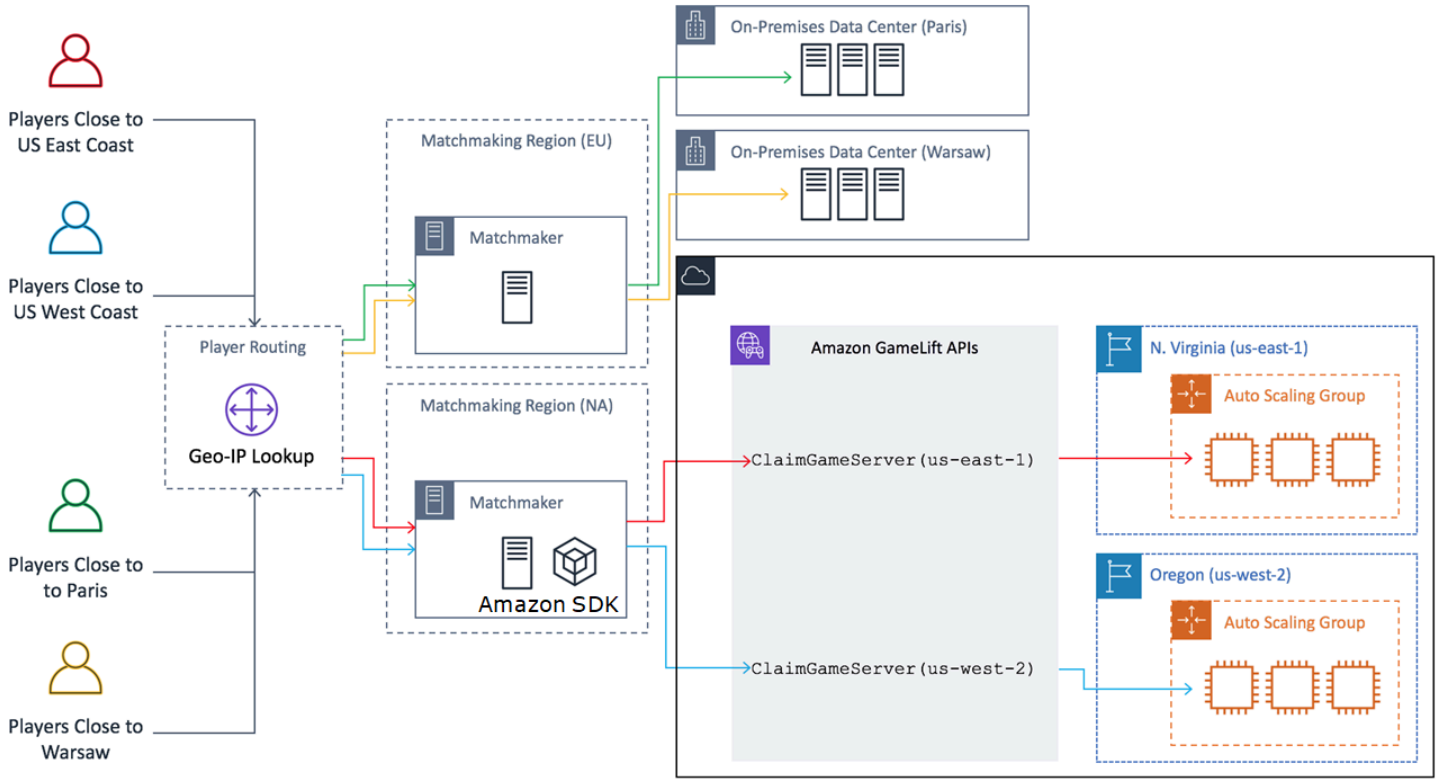
이 예에서는 북미와 유럽의 플레이어를 호스팅하도록 처음에 4개의 전용 데이터 센터에서 게임 호스팅을 처리합니다. 플레이어는 소재지에 따라 두 리전 매치메이커 중 하나로 라우팅됩니다. 매치메이커는 스킬과 지연 시간을 기준으로 플레이어를 그룹화하고 가까운 게임 서버에 배치하여 지연을 최소화합니다.



게임 개발자는 북미 게임 서버를 Amazon GameLift FleetIQ에서 제공하는 서버로 교체하려고 합니다. 개발자는 우선 Amazon GameLift FleetIQ를 사용할 수 있도록 게임 서버를 약간 업데이트한 다음 Amazon Machine Image(AMI)를 생성합니다. 이 이미지는 게임용으로 배포된 모든 EC2 인스턴스에 설치됩니다. 이미지에는 게임 서버, 종속성 및 플레이어를 위한 게임 세션을 실행하는 데 필요한 모든 것이 포함되어 있습니다.

AMI가 준비되면 개발자는AWS 북미 리전(`us-east-1` 및 `us-west-2`))별로 하나씩 Amazon GameLift FleetIQ 게임 서버 그룹 두 개를 생성합니다. 개발자는 시작 템플릿(AMI 제공), 필요한 인스턴스 유형 목록 및 그룹에 대한 기타 구성 설정을 전달합니다. 필요한 인스턴스 유형 목록은 게임 호스팅에 대해 실행 가능한 스팟 인스턴스를 확인할 때 사용할 유형을 Amazon GameLift FleetIQ에 알려줍니다.

마지막으로, 개발자는 AWS SDK를 Amazon GameLift FleetIQ와 북미 지역의 매치메이커에 통합합니다. 이 매치메이커는 새로운 플레이어 그룹이 게임 세션을 위해 서버 용량이 필요할 때 Amazon GameLift FleetIQ를 호출합니다. Amazon GameLift FleetIQ는 사용 가능한 게임 서버가 있는 스팟 인스턴스를 찾아 플레이어용으로 예약하고 서버 연결 정보를 제공합니다. 플레이어는 서버에 연결하여 게임을 플레이하고 연결을 끊습니다. 플레이어가 새 게임을 시작하려고 하면 매치메이킹이 다시 시작되어 Amazon GameLift FleetIQ에서 사용 가능한 게임 서버를 다시 찾습니다. 새 게임 요청이 있을 때마다 Amazon GameLift FleetIQ는 중단 가능성이 낮은 게임 서버를 검색하고 선택합니다. 즉, 스팟 인스턴스 가용성이 시간에 따라 변동되더라도 Amazon GameLift FleetIQ는 항상 게임 호스팅에 대해 실행 가능하지 않은 게임 서버에서 멀리 플레이어를 리디렉션합니다.



Amazon GameLift FleetIQ 수명 주기 및 상태

스팟 밸런싱 프로세스

Amazon GameLift FleetIQ는 스팟 인스턴스가 있는 Auto Scaling 그룹의 인스턴스의 균형을 주기적으로 조정합니다. 이 프로세스는 ON_DEMAND_ONLY 밸런싱 전략을 사용하거나 활성 인스턴스가 없는 게임 서버 그룹에서는 활성화되지 않습니다.

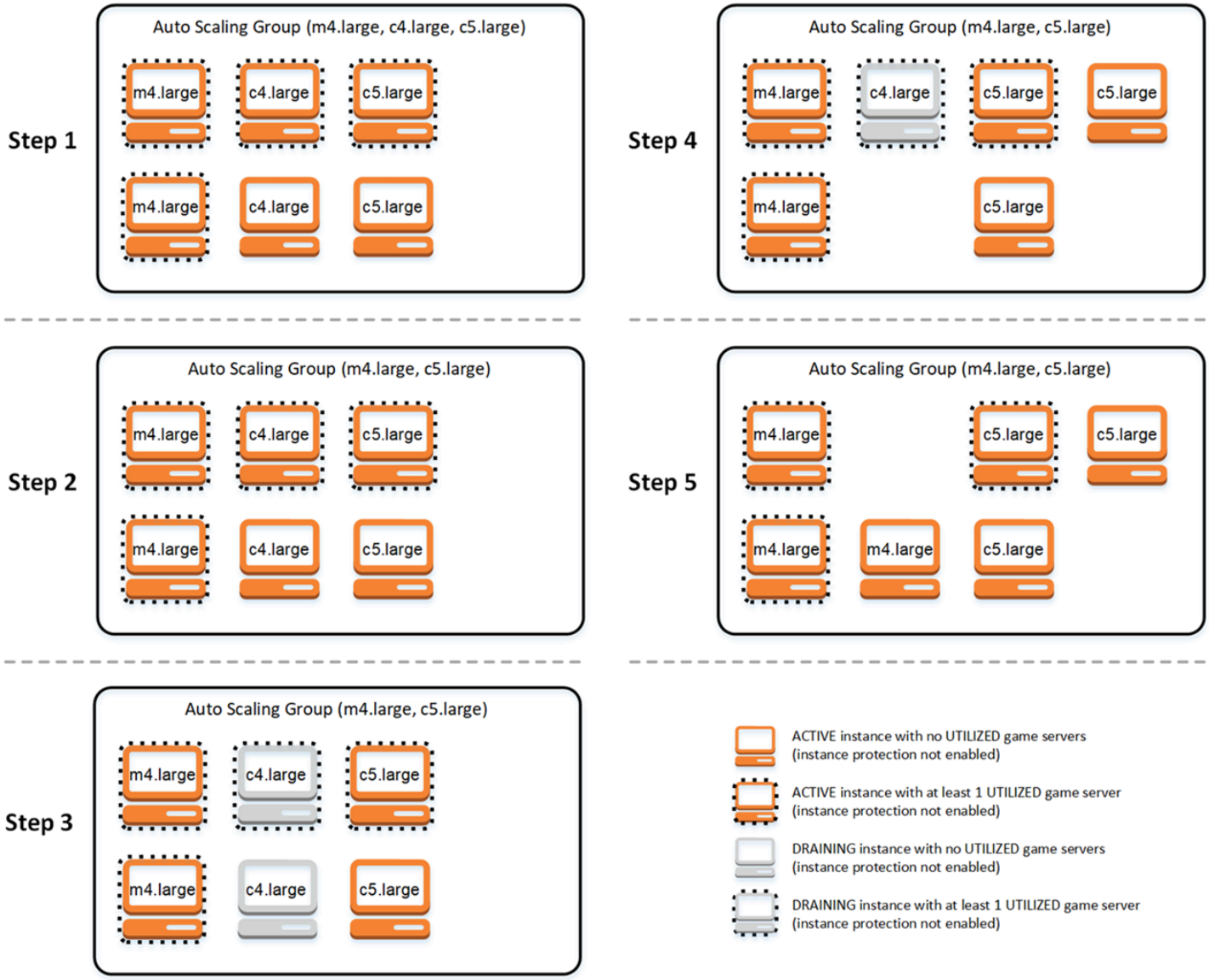
스팟 밸런싱에는 두 가지 주요 목표가 있습니다.

- 게임 호스팅에 대해 실행 가능한 스팟 인스턴스 유형만 사용하여 그룹을 지속적으로 새로 고치기 위해.
- 예기치 않은 게임 서버 중단의 영향을 줄일 목적으로 실행 가능한 여러 인스턴스 유형(가능한 경우)을 사용하기 위해

Amazon GameLift FleetIQ는 그룹의 인스턴스 유형을 평가하고 게임 서버 중단을 초래할 가능성이 더 높은 인스턴스를 제거하여 균형을 맞춥니다. 밸런싱 중에 활성 게임 플레이가 있는 인스턴스가 종료되지 않도록 프로덕션 환경의 게임 서버 그룹에 대해 게임 서버 보호 기능을 설정하는 것이 가장 좋습니다.

Example

다음 예제에서는 오토 스케일링의 인스턴스가 어떻게 스팟 밸런싱의 영향을 받는지 보여 줍니다.



- 단계 1. 게임 서버 그룹을 통해 연결된 오토 스케일링이 게임 서버 보호가 활성화된 상태로 m4.large, c4.large 및 c5.large 유형의 인스턴스를 시작하도록 설정됩니다. 오토 스케일링은 각 유형의 두 스팟 인스턴스로 구성된 밸런싱된 컬렉션을 시작했습니다. 4개의 인스턴스에는 UTILIZED 상태(파선 테두리로 표시됨)의 게임 서버가 하나 이상 있는 반면, 2개의 인스턴스는 현재 게임 플레이를 지원하지 않습니다.
- 단계 2. Amazon GameLift FleetIQ는 세 가지 인스턴스 유형 모두의 현재 게임 호스팅 실행 가능성을 평가합니다. 평가 결과 c4.large 인스턴스 유형에는 게임 서버 중단 가능성이 용납할 수 없는 것으로 확인되었습니다. Amazon GameLift FleetIQ는 Auto Scaling 그룹 구성을 즉시 업데이트하여 인스턴

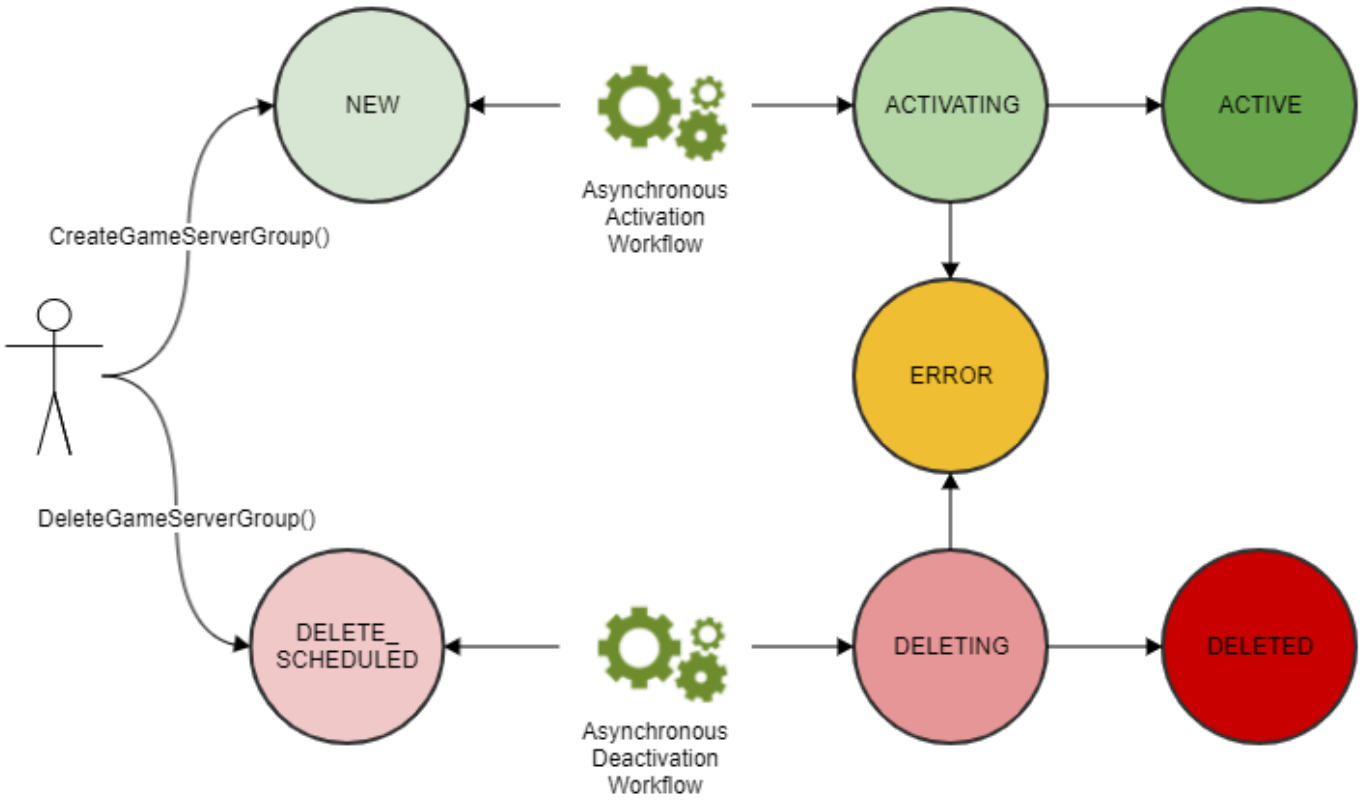
스 유형 목록에서 c4.large를 일시적으로 제거하여 추가 c4.large 인스턴스가 시작되지 않도록 합니다.

- 단계 3. Amazon GameLift FleetIQ는 c4.large 유형의 기존 인스턴스를 식별하고 그룹에서 제거하도록 조치를 취합니다. 첫 번째 단계로 c4.large 인스턴스에서 실행 중인 모든 게임 서버에 드레이닝 플래그가 지정됩니다. 드레이닝 인스턴스의 게임 서버는 다른 게임 서버를 사용할 수 없을 때 최후의 수단으로만 클레임될 수 있습니다. 또한 드레이닝 인스턴스가 있는 오토 스케일링이 트리거되어 새 인스턴스를 시작함으로써 드레이닝 인스턴스를 교체합니다.
- 단계 4. 실행 가능한 새 인스턴스가 온라인 상태가 되면 오토 스케일링이 드레이닝 인스턴스를 종료합니다. 이 교체를 통해 그룹의 원하는 용량을 유지할 수 있습니다. 종료되는 첫 번째 인스턴스는 사용되는 게임 서버가 없고 게임 서버 보호가 비활성화된 c4.large 인스턴스입니다. 이는 새 c5.large 인스턴스로 교체됩니다.
- 단계 5. 게임 서버 보호가 활성화된 드레이닝 인스턴스는 게임 서버가 게임 플레이를 지원하는 동안 계속 실행됩니다. 게임 플레이가 끝나면 새 m4.large 인스턴스가 시작되고 남은 c4.large 인스턴스가 종료됩니다.

이 프로세스의 결과로 그룹이 3가지에서 2가지 인스턴스 유형을 사용하는 것으로 밸런싱되고 오토 스케일링에서는 원하는 용량이 유지됩니다. Amazon GameLift FleetIQ는 게임 호스팅 실행 가능성에 대한 인스턴스 유형의 원래 목록을 계속 평가합니다. c4.large가 다시 실행 가능한 인스턴스 유형으로 간주되면 세 가지 인스턴스 유형을 모두 포함하도록 오토 스케일링이 업데이트됩니다. 이 그룹은 시간이 지남에 따라 자동으로 밸런싱됩니다.

게임 서버 그룹의 수명

게임 서버 그룹은 프로비저닝 및 상태 업데이트를 포함한 다음과 같은 수명 주기를 거칩니다. 게임 서버 그룹은 수명이 긴 리소스로 예상됩니다.



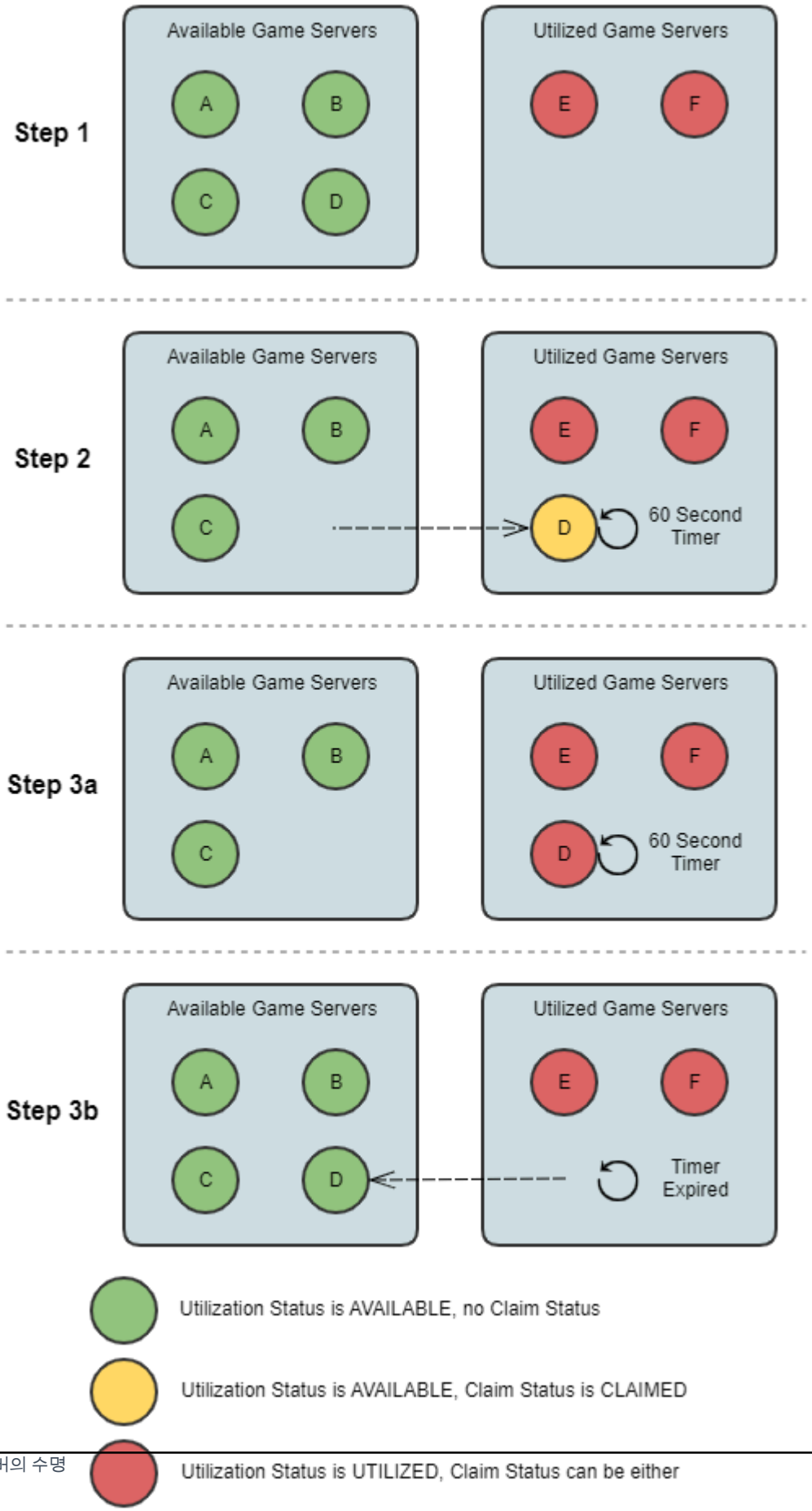
- Amazon GameLift API을 호출`CreateGameServerGroup()`하고 EC2 시작 템플릿 및 구성 설정을 전달하여 게임 서버 그룹을 생성합니다. 호출에 대한 응답으로 새 게임 서버 그룹이 생성되어 상태에 배치됩니다NEW.
- Amazon GameLift FleetIQ는 비동기 활성화 워크플로를 활성화하여 게임 서버 그룹 상태를 로 전환합니다ACTIVATING. 워크플로는 Amazon EC2 Auto Scaling 그룹 및 제공된 가 있는 EC2 인스턴스를 포함한 기본 리소스 생성을 시작합니다AMI.
  - 어떤 이유로든 프로비저닝이 실패하면 게임 서버 그룹이 상태로 전환됩니다ERROR. 오류 원인을 디버깅하는 데 도움이 되는 추가 오류 정보를 얻으려면 오류 상태의 게임 서버 그룹에서 `DescribeGameServerGroup()`을 호출합니다.
  - 프로비저닝에 성공하면 게임 서버 그룹이 상태로 전환됩니다ACTIVE. 이때 Amazon GameLift FleetIQ 에 등록하는 게임 서버로 인스턴스가 시작됩니다. 그룹의 인스턴스 유형은 게임 호스팅 실행 가능성에 대해 주기적으로 평가되고 필요에 따라 균형을 맞춥니다. Amazon GameLift FleetIQ 는 또한 그룹의 활성 게임 서버의 상태를 추적하고 게임 서버에 대한 요청에 응답합니다.
- 게임 서버 그룹을 제거하려면 그룹 식별자를 사용하여 `DeleteGameServerGroup()`을 호출합니다. 이 작업은 게임 서버 그룹을 DELETE_ 상태로 전환합니다SCHEDULED. ACTIVE 또는 ERROR 상태의 게임 서버 그룹만 삭제를 예약할 수 있습니다.

- Amazon GameLift FleetIQ는 DELETE_SCHEDULED 상태에 대한 응답으로 비동기 비활성화 워크플로를 활성화하여 게임 서버 그룹 상태를 로 전환합니다DELETING. 게임 서버 그룹만 삭제하거나 게임 서버 그룹 및 연결된 오토 스케일링을 모두 삭제할 수 있습니다.
  - 어떤 이유로든 비활성화에 실패하면 게임 서버 그룹이 상태로 전환됩니다ERROR. 오류 원인을 디버깅하는 데 도움이 되는 추가 오류 정보를 얻으려면 오류 상태의 게임 서버 그룹에서 `DescribeGameServerGroup()`을 호출합니다.
  - 비활성화에 성공하면 게임 서버 그룹이 상태로 전환됩니다DELETED.

게임 서버의 수명

Amazon GameLift FleetIQ 를 사용하면 게임 서버가 프로비저닝 및 상태 업데이트를 포함하여 다음 수명 주기를 거칩니다. 게임 서버는 수명이 짧은 리소스입니다. 게임 세션이 끝난 후 다른 게임 세션에 다시 사용하지 말고 게임 서버의 등록을 취소하는 것이 가장 좋습니다. 이렇게 하면 항상 게임 호스팅에 실행 가능한 가장 저렴한 리소스에서 사용 가능한 게임 서버가 실행되고 있는지 확인할 수 있습니다.

- Amazon GameLift FleetIQ 연결 Auto Scaling 그룹의 인스턴스에서 실행되는 게임 서버 프로세스가 Amazon GameLift API GameLift FleetIQ`RegisterGameServer()`에 플레이어와 게임 플레이를 호스팅할 준비가 되었음을 알릴 때 게임 서버 리소스가 생성됩니다. 게임 서버의 두 가지 상태를 통해 현재 가용성이 추적됩니다.
  - 사용률 상태는 게임 서버가 현재 게임 플레이를 지원하는지 여부를 추적합니다. 이 상태는 처음에로 설정되어 새 게임 플레이를 수락할 준비가 되었음을 AVAILABLE나타냅니다. 게임 서버가 게임 플레이에 점유되면 이 상태가 로 설정됩니다UTILIZED.
  - 클레임 상태는 게임 서버가 임박한 게임 플레이에 클레임되었는지 여부를 추적합니다. CLAIMED 상태인 게임 서버는 게임 클라이언트(또는 매치메이커와 같은 게임 서비스)가 일시적으로 예약했음을 나타냅니다. 이 상태는 Amazon GameLift FleetIQ가 여러 요청자에게 동일한 게임 서버를 제공하지 못하도록 합니다. 빈 클레임 상태의 게임 서버는 클레임 가능합니다.
- 다음 다이어그램은 게임 서버의 사용률 상태 및 클레임 상태가 수명 기간 동안 어떻게 변하는지를 보여줍니다.


Step 1
Available Game Servers
A
B
C
D
Utilized Game Servers
E
F
Step 2
Available Game Servers
A
B
C
Utilized Game Servers
E
F
D
60 Second Timer
Step 3a
Available Game Servers
A
B
C
Utilized Game Servers
E
F
D
60 Second Timer
Step 3b
Available Game Servers
A
B
C
D
Utilized Game Servers
E
F
Timer Expired
Utilization Status is AVAILABLE, no Claim Status
Utilization Status is AVAILABLE, Claim Status is CLAIMED
Utilization Status is UTILIZED, Claim Status can be either

- 단계 1. 게임 서버 그룹에 6개의 게임 서버가 등록되어 있습니다. 4개는 사용률 상태 AVAILABLE(A, B, C, D)이고 2개는 현재UTILIZED(E 및 F)입니다.
- 단계 2. 게임 클라이언트 또는 매치메이킹 시스템이 Amazon GameLift API을 호출`ClaimGameServer()`하여 새 게임 서버를 요청합니다. 이 요청은 Amazon GameLift FleetIQ에 사용 가능한 게임 서버(D)를 검색하고 클레임 상태를 60초 CLAIMED 동안 로 설정하라는 메시지를 표시합니다. Amazon GameLift FleetIQ는 게임 서버에 대한 연결 정보(IP 주소 및 포트)와 기타 선택적 게임별 데이터로 요청에 응답합니다. 게임 서버에서 게임 플레이가 아직 시작되지 않았기 때문에 사용률 상태는 로 유지AVAILABLE되지만 다른 요청으로는 요청할 수 없습니다.
- 단계 3a. 제공된 연결 정보를 사용하여 게임 클라이언트가 게임 서버에 연결하고 게임 플레이를 시작할 수 있습니다. Amazon GameLift API UTILIZED를 호출하여 사용률을 로 변경하려면 게임 서버(D)를 60초 이내에 트리거해야 합니다`UpdateGameServer()`.
- 단계 3b. 게임 서버의 사용률 상태가 60초 이내에 업데이트되지 않으면 클레임 타이머가 만료되고 클레임 상태가 빈 클레임으로 재설정됩니다. 게임 서버(D)가 사용 가능하고 클레임되지 않은 게임 서버 풀로 반환됩니다.

- 게임 서버의 게임 플레이가 완료되고 플레이어 연결이 끊어지면 게임 서버 리소스가 제거됩니다. 종료하기 전에 게임 서버 프로세스는 Amazon GameLift API을 호출`DeregisterGameServer()`하여 Amazon GameLift FleetIQ에 게임 서버 그룹의 게임 서버 풀에서 이탈을 알립니다.

Amazon GameLift FleetIQ 모범 사례

Amazon GameLift FleetIQ는 게임 호스팅을 위한 Amazon EC2 리소스를 관리하는 데 도움이 되는 저수준 로직 계층입니다. 특히 Amazon GameLift FleetIQ는 게임 세션이 중단될 가능성을 최소화하여 게임 호스팅에 적합한 스팟 인스턴스의 사용을 최적화합니다. 사용 가능한 게임 서버를 추적하고 실행 가능성이 높은 저렴한 게임 서버로 게임 플레이를 라우팅하는 기본적인 게임 호스팅 기능을 제공합니다.

독립형 기능으로 사용되는 Amazon GameLift FleetIQ는 호스팅 비용 최소화를 위해 FleetIQ도 사용하는 전체 Amazon GameLift 관리형 솔루션과 함께 제공되는 고급 기능은 제공하지 않습니다. 매치메이킹, 지연 시간 기반 플레이어 라우팅, 게임 세션 및 플레이어 세션 관리, 버전 관리 등의 기능이 필요한 경우 Amazon GameLift 솔루션을 살펴봅니다.

다음은 Amazon GameLift FleetIQ의 이점을 최대한 활용하는 데 도움이 될 수 있는 몇 가지 모범 사례입니다.

- 세션 기반 게임에 대해 Amazon GameLift FleetIQ를 사용합니다. Amazon GameLift FleetIQ는 게임 세션이 중단될 가능성이 가장 적은 인스턴스로 플레이어를 지속적으로 유도할 때 가장 효과적입니다. 장기간 세션을 유지하면 Amazon GameLift FleetIQ 밸런싱 프로세스가 방해되어 게임 세션이 중

단될 가능성이 높아집니다. 이상적인 워크플로는 플레이어가 매치메이킹(또는 서버 선택)에서 게임 플레이로 이동하는 것입니다. 게임이 종료되면 플레이어는 매치메이킹으로 돌아가고 새 인스턴스의 다른 게임 서버로 라우팅됩니다. 세션이 두 시간 미만인 게임에 Amazon GameLift FleetIQ를 사용하는 것이 좋습니다.

- 다양한 인스턴스 유형을 선택할 수 있도록 제공. 게임 서버 그룹을 설정할 때 사용할 인스턴스 유형 목록을 제공합니다. 더 많은 인스턴스 유형을 포함할수록 Amazon GameLift FleetIQ에서 게임 호스팅에 실행 가능성이 높은 스팟 인스턴스를 사용하는 유연성도 증가합니다. 예를 들어, 동일한 인스턴스 패밀리(c5.large, c5.xlarge, c5.2xlarge, c5.4xlarge) 내 여러 크기를 나열할 수도 있습니다. 인스턴스가 클수록 각 인스턴스에서 더 많은 게임 서버를 실행할 수 있으므로 잠재적으로 비용을 절감할 수 있습니다. 인스턴스가 작을수록 Auto Scaling은 플레이어 수요의 변화에 더 빠르게 대응할 수 있습니다. 원하는 인스턴스 유형 목록은 우선 순위가 정해지지 않는다는 점을 유의하고, 오토 스케일링은 실행 가능한 인스턴스 유형을 균형 있게 사용하여 그룹의 복원력을 유지합니다.
- 모든 인스턴스 유형에서 게임을 테스트합니다. 게임 서버 그룹에 대해 구성하는 모든 인스턴스 유형에서 게임 서버가 제대로 실행되는지 확인합니다.
- 인스턴스 용량 가중치를 사용합니다. 다양한 인스턴스 크기(예: c5.2xlarge, c5.4xlarge, c5.12xlarge)를 사용하도록 게임 서버 그룹을 구성하는 경우 각 인스턴스 유형에 대한 용량 가중치 정보를 포함합니다. 자세한 정보는 Amazon EC2 Auto Scaling 사용 설명서의 Amazon EC2 Auto Scaling에 대해 인스턴스 가중치 구성을 참조하세요.
- Amazon GameLift FleetIQ를 사용하여 게임 세션을 배치합니다. 게임 서버가 있는 플레이어 그룹을 배치할 때는 Amazon GameLift API `ClaimGameServer()`를 사용합니다. Amazon GameLift FleetIQ는 게임 세션이 중단될 가능성이 높은 인스턴스에 플레이어를 배치하지 않습니다.
- Amazon GameLift FleetIQ에 게임 서버 상태를 보고합니다. Amazon GameLift API `UpdateGameServer()`를 사용하여 서버 상태 및 사용률 상태를 주기적으로 보고합니다. 정확한 게임 서버 상태를 유지하면 Amazon GameLift FleetIQ에서 게임 플레이를 보다 효율적으로 배치할 수 있을 뿐만 아니라, 스팟 밸런싱 활동 중에 활성 게임 플레이가 있는 인스턴스가 종료되는 일을 방지할 수 있습니다.
- AutoScaling 정책 설정. 플레이어 사용률 및 예상 수요에 따라 호스팅 용량을 유지하는 대상 추적 조정 정책을 생성할 수 있습니다. Amazon GameLift FleetIQ 지표 `PercentUtilizedGameServers`는 현재 사용 중인 호스팅 용량의 크기를 측정한 것입니다. 대부분의 게임은 새 플레이어가 게임에 빠르게 참가할 수 있도록 사용되지 않는 게임 서버의 버퍼를 유지하려고 합니다. 플레이어 수요 변동에 따라 인스턴스를 추가하거나 제거하여 특정 버퍼 크기를 유지하는 조정 정책을 생성할 수 있습니다. 자세한 내용을 알아보려면 Amazon EC2 Auto Scaling 사용 설명서의 대상 추적 조정 정책을 참조하세요.
- 개발과 프로덕션 환경에 서로 다른 AWS 계정을 사용합니다. 전체 계정에서 개발과 프로덕션 구성을 분리하면 잘못된 구성이 라이브 플레이어에게 영향을 미치는 위험을 줄일 수 있습니다.

- 프로덕션 환경에서 게임 서버 그룹에 대한 게임 세션 보호를 활성화합니다. 플레이어를 보호하려면 게임 세션 보호 기능을 설정하고 조정 또는 밸런싱 활동으로 인해 활성 게임 세션이 조기에 종료되는 것을 방지합니다.
- Amazon GameLift FleetIQ와 통합하기 전에 EC2에서 게임을 테스트합니다. 먼저 EC2에서 게임을 시작 및 실행하고 구성을 미세 조정하는 것이 좋습니다. 그런 다음 동일한 시작 템플릿 및 AMI를 사용하여 게임 서버 그룹을 생성할 수 있습니다.

  Kubernetes를 사용 중인 경우 먼저 Kubernetes 클러스터에 표준 EC2 인스턴스를 추가한 다음 Kubernetes 클러스터의 작업자 노드에 대해 생성한 시작 템플릿을 사용하여 게임 서버 그룹을 생성하는 것이 좋습니다. EKS를 사용 중인 경우 EKS 클러스터 및 게임 서버 그룹을 별도로 생성합니다. 게임 서버 그룹의 경우 EKS 최적화 AMI를 EKS 통합에 사용되는 적절한 사용자 데이터 및 시작 템플릿 구성과 함께 사용합니다. EKS 작업자 노드 및 EKS 최적화 AMI에 대한 자세한 내용은 Amazon EKS 최적화 Linux AMI 안내서를 참조하십시오.
- 안정적인 게임 서버 가용성을 위해 게임 서버 그룹 밸런싱 전략 **ON_DEMAND_ONLY**를 사용합니다. 이 밸런싱 전략을 적용하면 스팟 인스턴스가 사용되지 않습니다. 이는 기능 출시 또는 기타 특별 이벤트와 같이 가장 필요할 때 서버 가용성을 보장하는 데 유용한 도구입니다. 필요에 따라 게임 서버 그룹을 스팟에서 온디맨드 전략으로 전환할 수 있습니다.

또한 다음 AWS 모범 사례를 검토합니다.

- Amazon EC2 모범 사례
- Amazon EC2 Auto Scaling 모범 사례

Amazon GameLift FleetIQ 기능

- 최적화된 스팟 밸런싱. Amazon GameLift FleetIQ는 인스턴스 유형을 주기적으로 평가하여 게임 세션 중단 가능성이 높아 실행 가능한 것으로 간주되지 않는 스팟 인스턴스를 대체합니다. EC2 오토 스케일링이 이전 인스턴스를 사용 중지하고 새 인스턴스를 시작하면 해당 그룹은 현재 게임 호스팅에 적합한 인스턴스 유형으로 항상 새로 고쳐집니다.
- 최적의 플레이어 라우팅. Amazon GameLift FleetIQ API는 새 게임 세션을 게임 세션이 중단될 가능성이 가장 적으면서도 가장 복원력이 뛰어난 스팟 인스턴스로 보냅니다. 또한 게임 세션을 더 적은 수의 인스턴스로 압축하여 EC2 오토 스케일링이 불필요한 리소스를 축소하고 호스팅 비용을 절감하는 기능을 개선합니다.

- 플레이어 사용량에 따른 Auto Scaling. Amazon GameLift FleetIQ는 게임 서버 사용률 데이터를 Amazon CloudWatch 지표로 내보냅니다. 이러한 지표를 바탕으로 사용 가능한 호스팅 리소스를 자동으로 조정하여 실제 플레이어 수요에 맞추고 호스팅 비용을 줄일 수 있습니다.
- Amazon EC2 인스턴스의 직접 관리. AWS 계정에서 EC2 인스턴스 및 EC2 오토 스케일링을 완벽하게 제어할 수 있습니다. 즉, 인스턴스 시작 템플릿을 설정하고, EC2 오토 스케일링 구성을 유지하며, 다른 AWS 서비스와 통합할 수 있습니다. 스팟 밸런싱 활동의 일환으로 Amazon GameLift FleetIQ는 일부 EC2 오토 스케일링 속성을 주기적으로 업데이트합니다. 필요에 따라 일시적으로 이러한 설정을 재정의하거나 Amazon GameLift FleetIQ 활동을 일시 중단할 수 있습니다.
- 여러 게임 서버 실행 파일 형식의 지원. Amazon GameLift FleetIQ는 Windows, Linux, 컨테이너 및 Kubernetes를 포함하여 현재 Amazon EC2에서 실행되는 모든 형식을 지원합니다. 지원되는 운영 체제 및 런타임 목록은 Amazon EC2 FAQ를 참조하세요.
- 여러 유형의 호스팅 리소스. Amazon GameLift FleetIQ를 사용하면 게임 서버 호스팅을 위한 다양한 인스턴스 유형에 액세스할 수 있습니다. (AWS 리전별로 가용성이 다릅니다.) 따라서 게임 서버에 적절한 조합의 CPU, 메모리, 스토리지 및 네트워킹 용량을 사용하여 플레이어에게 최상의 게임 경험을 제공할 수 있습니다.
- 전 세계 이용 가능. Amazon GameLift FleetIQ는 중국을 포함한 15개 리전에서 사용할 수 있습니다. 따라서 플레이어가 어디에 있든 최소한의 지연으로 게임 서버에 액세스할 수 있습니다. 리전의 전체 목록은 AWS 일반 참조의 Amazon GameLift 엔드포인트 및 할당량을 참조하세요.

Amazon GameLift FleetIQ 요금

Amazon GameLift는 사용 기간별로 인스턴스에 대해 요금을 부과하고, 전송된 데이터 양을 기준으로 대역폭을 청구합니다. Amazon GameLift에 관련된 전체적인 요금 및 가격 목록은 Amazon GameLift 요금을 참조하세요.

Amazon GameLift를 통한 게임 호스팅 또는 매치메이킹 비용 계산에 대한 자세한 내용은 AWS Pricing Calculator를 사용 방법을 설명하는 Amazon GameLift의 추정 요금 산출 섹션을 참조하세요.

Amazon GameLift FleetIQ 설정

이 섹션의 주제에서는 Amazon GameLift FleetIQ 서비스에서 사용할 AWS 계정을 설정하는 방법을 비롯한 설정 작업에 도움이 됩니다.

주제

- 아마존 GameLift FleetiQ 지원 소프트웨어
- Amazon GameLift FleetIQ AWS 계정 설정

아마존 GameLift FleetiQ 지원 소프트웨어

Amazon GameLift FleetiQ는 Amazon에서 호스팅하기 위한 64비트 멀티플레이어 게임 서버, 클라이언트 및 게임 서비스를 배포하는 데 사용됩니다. EC2 이 솔루션은 다음 환경을 지원합니다.

게임 서버용 운영 체제

에서 지원하는 모든 운영 체제에서 실행되는 게임 서버와 함께 Amazon GameLift FleetiQ를 사용할 수 있습니다. EC2 여기에는 아마존 리눅스, 우분투, 윈도우 서버, 레드햇 엔터프라이즈 리눅스, 리눅스 엔터프라이즈 서버, 페도라, 데비안, CentOS, 젠투 SUSE 리눅스, 오라클 리눅스, 프리 등이 포함됩니다. BSD Amazon EC2 기능에서 현재 EC2 기능 및 지원을 확인하십시오.

컨테이너의 사용

게임 서버가 컨테이너를 사용하는 경우 Amazon GameLift FleetiQ는 쿠버네티스, 아마존 엘라스틱 컨테이너 서비스 (Amazon) 및 아마존 Elastic Kubernetes Service () 와의 통합을 지원합니다. ECS EKS 자세한 내용은 AWS의 컨테이너를 참조하세요.

게임 개발 환경

게임 클라이언트 및 서버가 Amazon GameLift FleetiQ 서비스와 통신하려면 일부 통합이 필요합니다. 게임은 를 API 호출합니다. AWS SDK Amazon GameLift API 참조 설명서를 AWS SDK 다운로드하거나 확인하십시오.

GameLift Amazon에 대한 AWS SDK with 지원은 다음 언어로 제공됩니다. 개발 환경 지원에 대한 자세한 내용은 각 언어로 된 설명서를 참조하세요.

- C++ (SDK문서) (아마존) GameLift
- 자바 (SDK문서) (아마존 GameLift)

- . NET(SDK문서) (아마존 GameLift)
- Go (SDK문서) (아마존 GameLift)
- Python (SDK문서) (아마존 GameLift)
- 루비 (SDK문서) (아마존) GameLift
- PHP(SDK문서) (아마존 GameLift)
- JavaScript/Node.js (SDK문서) (아마존 GameLift)

Amazon GameLift FleetIQ AWS 계정 설정

Amazon GameLift FleetIQ를 Amazon EC2, Auto Scaling 및 기타 AWS 서비스와 함께 사용하려면 필요한 액세스 권한이 있는 를 AWS 계정 설정해야 합니다. 다음 단계를 완료합니다.

- Amazon GameLift FleetIQ 에 사용할 계정이 아직 없는 경우 새 AWS 계정을 생성합니다. 생성 AWS 계정을 참조하세요.
- 사용자 및 사용자 그룹에 대한 Amazon GameLift FleetIQ별 권한을 설정합니다. Amazon GameLift FleetIQ에 대한 사용자 권한 관리을 참조하세요.
- Amazon GameLift 과 Amazon EC2 리소스가 상호 작용할 수 있도록 IAM 역할을 생성합니다. 교차 서비스 상호 작용을 위한 IAM 역할 생성을 참조하세요.

생성 AWS 계정

Amazon GameLift FleetIQ 와 함께 사용할 AWS 계정 를 생성하고 설정합니다. AWS 계정를 만드는 데는 비용이 들지 않습니다.

주제

- 에 가입 AWS 계정
- 관리자 액세스 권한이 있는 사용자 생성

에 가입 AWS 계정

가 없는 경우 다음 단계를 AWS 계정완료하여 를 생성합니다.

에 가입하려면 AWS 계정

1. https://portal.aws.amazon.com/billing/가입 을 엽니다.

2. 온라인 지시 사항을 따릅니다.

   등록 절차 중 전화를 받고 전화 키패드로 확인 코드를 입력하는 과정이 있습니다.

   에 가입하면 AWS 계정AWS 계정 루트 사용자가 생성됩니다. 루트 사용자에게는 계정의 모든 AWS 서비스 및 리소스에 액세스할 권한이 있습니다. 보안 모범 사례는 사용자에게 관리 액세스 권한을 할당하고, 루트 사용자만 사용하여 루트 사용자 액세스 권한이 필요한 작업을 수행하는 것입니다.

AWS 는 가입 프로세스가 완료된 후 확인 이메일을 보냅니다. 언제든지 https://aws.amazon.com/로 이동하여 내 계정 을 선택하여 현재 계정 활동을 보고 계정을 관리할 수 있습니다.

관리자 액세스 권한이 있는 사용자 생성

에 가입한 후 일상적인 작업에 루트 사용자를 사용하지 않도록 AWS 계정를 보호하고, 를 AWS 계정 루트 사용자활성화하고 AWS IAM Identity Center, 관리 사용자를 생성합니다.

보안 AWS 계정 루트 사용자

1. 루트 사용자를 선택하고 AWS 계정 이메일 주소를 입력하여 계정 소유자AWS Management Console로 에 로그인합니다. 다음 페이지에서 비밀번호를 입력합니다.

   루트 사용자를 사용하여 로그인하는 데 도움이 필요하면 AWS 로그인 User Guide의 루트 사용자로 로그인을 참조하십시오.
2. 루트 사용자에 대해 다중 인증(MFA)을 켭니다.

   지침은 IAM 사용 설명서의 AWS 계정 루트 사용자(콘솔)에 대한 가상 MFA 디바이스 활성화를 참조하세요.

관리자 액세스 권한이 있는 사용자 생성

1. IAM Identity Center를 활성화합니다.

   지침은 AWS IAM Identity Center 사용 설명서의 AWS IAM Identity Center설정을 참조하세요.
2. IAM Identity Center에서 사용자에게 관리 액세스 권한을 부여합니다.

   를 자격 증명 소스 IAM Identity Center 디렉터리 로 사용하는 방법에 대한 자습서는 AWS IAM Identity Center 사용 설명서의 기본값으로 사용자 액세스 구성을 IAM Identity Center 디렉터리 참조하세요.

관리 액세스 권한이 있는 사용자로 로그인

- IAM Identity Center 사용자로 로그인하려면 IAM Identity Center 사용자를 생성할 때 이메일 주소로 전송URL된 로그인을 사용합니다.

  IAM Identity Center 사용자를 사용하여 로그인하는 방법에 대한 자세한 내용은 AWS 로그인 사용 설명서의 AWS 액세스 포털에 로그인을 참조하세요.

추가 사용자에게 액세스 권한 할당

1. IAM Identity Center에서 최소 권한 적용 모범 사례를 따르는 권한 세트를 생성합니다.

   지침은AWS IAM Identity Center 사용 설명서의 Create a permission set를 참조하세요.
2. 사용자를 그룹에 할당하고, 그룹에 Single Sign-On 액세스 권한을 할당합니다.

   지침은AWS IAM Identity Center 사용 설명서의 Add groups를 참조하세요.

Amazon GameLift FleetIQ에 대한 사용자 권한 관리

필요에 따라 추가 사용자를 생성하거나 기존 사용자에게 Amazon GameLift FleetIQ 액세스 권한을 확장합니다. Amazon GameLift FleetIQ 게임 서버 그룹 및 관련 Amazon EC2 및 Auto Scaling 서비스를 사용하는 사용자는 이러한 서비스에 액세스할 수 있는 권한이 있어야 합니다.

모범 사례(의 보안 모범 사례IAM)로 모든 사용자에게 최소 권한 권한을 적용합니다. 개별 사용자 또는 사용자 그룹에 대한 권한을 설정하고 서비스, 작업 또는 리소스별로 사용자 액세스를 제한할 수 있습니다.

다음 지침을 사용하여 AWS 계정에서 사용자를 관리하는 방법에 따라 사용자 권한을 설정합니다. IAM 사용자를 사용하는 경우 모범 사례로 개별 사용자가 아닌 역할 또는 사용자 그룹에 항상 권한을 연결합니다.

- 사용자의 구문 권한
- 와 함께 사용할 추가 권한 구문 AWS CloudFormation

액세스 권한을 제공하려면 사용자, 그룹 또는 역할에 권한을 추가하세요:

- 의 사용자 및 그룹 AWS IAM Identity Center:

권한 세트를 생성합니다. AWS IAM Identity Center 사용 설명서의 권한 세트 생성의 지침을 따릅니다.

- 자격 증명 공급자를 IAM 통해 에서 관리되는 사용자:

  ID 페더레이션을 위한 역할을 생성합니다. IAM 사용 설명서의 타사 자격 증명 공급자(페더레이션)에 대한 역할 생성의 지침을 따릅니다.

- IAM 사용자:
  - 사용자가 맡을 수 있는 역할을 생성합니다. IAM 사용 설명서의 IAM 사용자 역할 생성의 지침을 따릅니다.
  - (권장되지 않음)정책을 사용자에게 직접 연결하거나 사용자를 사용자 그룹에 추가합니다. IAM 사용 설명서의 사용자(콘솔)에 권한 추가의 지침을 따릅니다.

참조: Amazon GameLift FleetIQ _policy

다음은 참조용 Amazon GameLift FleetIQ _policy의 예입니다.

```
{
  "Version": "2012-10-17",
  "Statement":
  [
    {
      "Action":
      [
        "iam:PassRole"
      ],
      "Effect": "Allow",
      "Resource": "*",
      "Condition":
      {
        "StringEquals":
        {
          "iam:PassedToService": "gamelift.amazonaws.com"
        }
      }
    },
    {
      "Action":
      [
        "iam:CreateServiceLinkedRole"
```

```
      ],
      "Effect": "Allow",
      "Resource": "arn:*:iam::*:role/aws-service-role/autoscaling.amazonaws.com/
AWSServiceRoleForAutoScaling"
    },
    {
      "Action":
      [
        "autoscaling:CreateAutoScalingGroup",
        "autoscaling:CreateOrUpdateTags",
        "autoscaling:DescribeAutoScalingGroups",
        "autoscaling:ExitStandby",
        "autoscaling:PutLifecycleHook",
        "autoscaling:PutScalingPolicy",
        "autoscaling:ResumeProcesses",
        "autoscaling:SetInstanceProtection",
        "autoscaling:UpdateAutoScalingGroup",
        "autoscaling:DeleteAutoScalingGroup"
      ],
      "Effect": "Allow",
      "Resource": "*"
    },
    {
      "Action":
      [
        "ec2:DescribeAvailabilityZones",
        "ec2:DescribeSubnets",
        "ec2:RunInstances",
        "ec2:CreateTags"
      ],
      "Effect": "Allow",
      "Resource": "*"
    },
    {
      "Action":
      [
        "events:PutRule",
        "events:PutTargets"
      ],
      "Effect": "Allow",
      "Resource": "*"
    }
  ]
```

```
}
```

에 대한 추가 권한 AWS CloudFormation

AWS CloudFormation를 사용하여 게임 호스팅 리소스를 관리하는 경우 정책 구문에 AWS CloudFormation 권한을 추가합니다.

```
    {
      "Action": [
        "autoscaling:DescribeLifecycleHooks",
        "autoscaling:DescribeNotificationConfigurations",
        "ec2:DescribeLaunchTemplateVersions"
      ]
      "Effect": "Allow",
      "Resource": "*"
    }
```

사용자의 프로그래밍 방식 액세스 설정

사용자는 AWS 외부에서 와 상호 작용하려는 경우 프로그래밍 방식으로 액세스해야 합니다 AWS Management Console. 프로그래밍 방식 액세스를 부여하는 방법은 에 액세스하는 사용자 유형에 따라 다릅니다 AWS.

사용자에게 프로그래밍 방식 액세스 권한을 부여하려면 다음 옵션 중 하나를 선택합니다.

프로그래밍 방식 액세스가 필요한 사용자는 누구인가요?	To	액세스 권한을 부여하는 사용자
작업 인력 ID (IAMID 센터에서 관리하는 사용자)	임시 자격 증명을 사용하여 AWS CLI AWS SDKs, 또는 에 대한 프로그래밍 요청에 서명합니다 AWS APIs.	사용하고자 하는 인터페이스에 대한 지침을 따릅니다. • 의 경우 AWS Command Line Interface 사용 설명서의 AWS CLI 를 사용하도록 구성을 AWS IAM Identity Center AWS CLI참조하세요. • AWS SDKs, 도구 및 의 경우 AWS SDKs 및 도구 참조 가이드의 IAM Identity Center

프로그래밍 방식 액세스가 필요한 사용자는 누구인가요?	To	액세스 권한을 부여하는 사용자
		인증을 AWS APIs참조하세요.
IAM	임시 자격 증명을 사용하여 AWS CLI AWS SDKs, 또는 에 대한 프로그래밍 요청에 서명합니다 AWS APIs.	IAM 사용 설명서의 AWS 리소스와 함께 임시 자격 증명 사용의 지침을 따릅니다.
IAM	(권장되지 않음) 장기 보안 인증 정보를 사용하여 AWS CLI AWS SDKs, 또는 에 대한 프로그래밍 요청에 서명합니다 AWS APIs.	사용하고자 하는 인터페이스에 대한 지침을 따릅니다. • 의 경우 AWS Command Line Interface 사용 설명서의 IAM 사용자 자격 증명을 사용하여 인증을 AWS CLI참조하세요. • 및 도구에 대한 AWS SDKs 자세한 내용은 AWS SDKs 및 도구 참조 가이드의 장기 자격 증명을 사용하여 인증을 참조하세요. • 의 경우 IAM 사용 설명서의 IAM 사용자에 대한 액세스 키 관리를 AWS APIs참조하세요.

액세스 키를 사용하는 경우 AWS 액세스 키 관리 모범 사례 를 참조하세요.

교차 서비스 상호 작용을 위한 IAM 역할 생성

Amazon GameLift FleetIQ가 Amazon EC2 인스턴스 및 Auto Scaling 그룹과 함께 작동하려면 서비스가 서로 상호 작용하도록 허용해야 합니다. 이는 계정 AWS 에서 IAM 역할을 생성하고 제한된 권한 집합을 할당하여 수행됩니다. 또한 각 역할은 역할을 수행할 수 있는 서비스를 지정합니다.

다음 역할을 설정합니다.

- Amazon GameLift FleetIQ에 대한 역할 생성 Amazon EC2 리소스를 업데이트합니다.
- Amazon에 대한 역할 생성 EC2 Amazon GameLift FleetIQ 와 통신하기 위한 리소스입니다.

Amazon GameLift FleetIQ에 대한 역할 생성

이 역할을 통해 Amazon GameLift FleetIQ는 스팟 밸런싱 및 자동 조정 활동의 일환으로 Amazon EC2 인스턴스, Auto Scaling 그룹 및 수명 주기 후크에 액세스하고 수정할 수 있습니다.

IAM 콘솔 또는 AWS CLI를 사용하여 Amazon GameLift FleetIQ에 대한 역할을 생성하고 필요한 권한이 있는 관리형 정책을 연결합니다. IAM 역할 및 관리형 정책에 대한 자세한 내용은 AWS 서비스 및 관리형 정책의 역할 생성을 참조하세요. AWS

Console

다음 단계에서는 를 사용하여 Amazon GameLift에 대한 관리형 정책을 사용하여 서비스 역할을 생성하는 방법을 설명합니다 AWS Management Console.

1. IAM 콘솔을 열고 역할: 역할 생성 을 선택합니다.
2. 신뢰할 수 있는 엔터티 유형 선택(Select type of trusted entity)에서 AWS 서비스(service)를 선택합니다.
3. 사용 사례 선택 GameLift에서 서비스 목록에서 를 선택합니다. 사용 사례 선택에서 적절한 Amazon GameLift 사용 사례가 자동으로 선택됩니다. 계속하려면 다음: 권한을 선택합니다.
4. 연결된 권한 정책 목록에는 하나의 정책인 가 포함되어야 합니다GameLiftGameServerGroupPolicy . 이 정책이 표시되지 않으면 필터를 확인하거나 검색 기능을 사용하여 역할에 추가합니다. 정책의 구문을 볼 수 있지만(확장하려면 ▸ 아이콘 선택) 구문을 변경할 수는 없습니다. 역할이 생성되면 역할을 업데이트하고 추가 정책을 연결하여 권한을 추가/제거할 수 있습니다.

   권한 경계 설정의 경우 기본 설정(권한 경계 없이 역할 생성)을 유지합니다. 이는 필요하지 않은 고급 설정입니다. 계속하려면 다음: 태그를 선택합니다.
5. 태그 추가는 리소스 관리를 위한 선택적 설정입니다. 예를 들어 이 역할에 태그를 추가하여 역할별로 프로젝트별 리소스 사용을 추적할 수 있습니다. IAM 역할 및 기타 용도의 태그 지정에 대한 자세한 내용은 자세히 알아보기 링크를 참조하세요. 계속하려면 다음: 검토를 선택합니다.
6. 검토 페이지에서 필요에 따라 다음과 같이 변경합니다.

   - 역할 이름을 입력하고 선택적으로 설명을 업데이트합니다.

- 다음을 확인합니다.
  - 신뢰할 수 있는 엔터티가 'AWS service: gamelift.amazonaws.com'으로 설정되어 있습니다. 역할이 생성되면 이 값을 업데이트해야 합니다.
  - 정책에는 이 포함됩니다 GameLiftGameServerGroupPolicy.

작업을 완료하려면 역할 생성을 선택합니다.

7. 새 역할이 생성되면 역할의 신뢰 관계를 수동으로 업데이트해야 합니다. 역할 페이지로 이동하여 새 역할 이름을 선택하여 요약 페이지를 엽니다. 신뢰 관계 탭을 열고 신뢰 관계 편집을 선택합니다. 정책 문서에서 `autoscaling.amazonaws.com`을 포함하도록 `Service` 속성을 업데이트합니다. 수정된 `Service` 속성은 다음과 같아야 합니다.

```
        "Service": [
          "gamelift.amazonaws.com",
          "autoscaling.amazonaws.com"
        ]
```

변경 사항을 저장하려면 신뢰 정책 업데이트를 선택합니다.

이제 역할을 사용할 준비가 되었습니다. 역할의 요약 페이지 상단에 표시되는 역할의 ARN 값을 기록해 둡니다. Amazon GameLift FleetIQ 게임 서버 그룹을 설정할 때 이 정보가 필요합니다.

AWS CLI

다음 단계에서는 를 사용하여 Amazon GameLift에 대한 관리형 정책을 사용하여 서비스 역할을 생성하는 방법을 설명합니다 AWS CLI.

1. 다음 JSON 구문으로 신뢰 정책 파일(예: `FleetIQtrustpolicyGameLift.json`)을 생성합니다.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Principal": {
        "Service": [
          "gamelift.amazonaws.com",
          "autoscaling.amazonaws.com"
        ]
```

```
      },
      "Action": "sts:AssumeRole"
    }
  ]
}
```

2. iam create-role을 사용하여 새 IAM 역할을 생성하고 방금 생성한 신뢰 정책 JSON 파일과 연결합니다.

   Windows:

   ```
   AWS iam create-role --role-name FleetIQ-role-for-GameLift --assume-role-policy-
   document file://C:\policies\FleetIQtrustpolicyGameLift.json
   ```

   Linux:

   ```
   AWS iam create-role --role-name FleetIQ-role-for-GameLift --assume-role-policy-
   document file://policies/FleetIQtrustpolicyGameLift.json
   ```

   요청이 성공하면 새로 생성된 역할의 속성이 응답에 포함됩니다. ARN 값을 기록해 둡니다. Amazon GameLift FleetIQ 게임 서버 그룹을 설정할 때 이 정보가 필요합니다.

3. iam attach-role-policy을 사용하여 관리형 권한 정책 “GameLiftGameServerGroupPolicy”을 연결합니다.

   ```
   AWS iam attach-role-policy --role-name FleetIQ-role-for-GameLift --policy-arn
    arn:aws:iam::aws:policy/GameLiftGameServerGroupPolicy
   ```

   권한 정책이 연결되어 있는지 확인하려면 새 역할 이름으로 iam list-attached-role-policies을 호출합니다.

이제 역할을 사용할 준비가 되었습니다. `role-arn` 속성을 새 IAM 역할 ARN 값으로 설정하여 gamelift create-game-server-group를 호출하여 역할이 올바르게 구성되었는지 확인할 수 있습니다. 가 ACTIVE 상태가 `GameServerGroup` 되면 Amazon GameLift FleetIQ가 예상대로 계정의 Amazon EC2 및 Auto Scaling 리소스를 수정할 수 있음을 나타냅니다.

Amazon에 대한 역할 생성 EC2

이 역할을 사용하면 Amazon EC2 리소스가 Amazon GameLift FleetIQ 와 통신할 수 있습니다. 예를 들어 Amazon EC2 인스턴스에서 실행 중인 게임 서버는 상태 상태를 보고할 수 있어야 합니다. Amazon GameLift FleetIQ 게임 서버 그룹을 생성할 때 Amazon EC2 시작 템플릿과 함께 IAM 인스턴스 프로파일에 이 역할을 포함합니다.

를 AWS CLI 사용하여 Amazon 에 대한 역할을 생성하고EC2, 필요한 권한이 있는 사용자 지정 정책을 연결하고, 역할을 인스턴스 프로파일에 연결합니다. 자세한 내용은 AWS 서비스에 대한 역할 생성을 참조하세요.

AWS CLI

이 단계에서는 를 EC2 사용하여 Amazon에 대한 사용자 지정 Amazon GameLift 권한을 사용하여 서비스 역할을 생성하는 방법을 설명합니다 AWS CLI.

1. 다음 JSON 구문으로 신뢰 정책 파일(예: `FleetIQtrustpolicyEC2.json`)을 생성합니다.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Principal": {
        "Service": "ec2.amazonaws.com"
      },
      "Action": "sts:AssumeRole"
    }
  ]
}
```

2. iam create-role을 사용하여 새 IAM 역할을 생성하고 방금 생성한 신뢰 정책 JSON 파일과 연결합니다.

   Windows:

```
AWS iam create-role --role-name FleetIQ-role-for-EC2 --assume-role-policy-
document file://C:\policies\FleetIQtrustpolicyEC2.json
```

   Linux:

```
AWS iam create-role --role-name FleetIQ-role-for-EC2 --assume-role-policy-
document file://policies/FleetIQtrustpolicyEC2.json
```

요청이 성공하면 새로 생성된 역할의 속성이 응답에 포함됩니다. ARN 값을 기록해 둡니다. Amazon EC2 시작 템플릿을 설정할 때 이 정보가 필요합니다.

3. 다음 JSON 구문으로 권한 정책 파일(예: `FleetIQpermissionsEC2.json`)을 생성합니다.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": "gamelift:*",
            "Resource": "*"
        }
    ]
}
```

4. iam put-role-policy을 사용하여 방금 생성한 권한 정책 JSON 파일을 새 역할에 연결합니다.

   Windows:

```
AWS iam put-role-policy --role-name FleetIQ-role-for-EC2 --policy-
name FleetIQ-permissions-for-EC2 --policy-document file://C:\policies
\FleetIQpermissionsEC2.json
```

   Linux:

```
AWS iam put-role-policy --role-name FleetIQ-role-for-EC2 --policy-name FleetIQ-
permissions-for-EC2 --policy-document file://policies/FleetIQpermissionsEC2.json
```

   권한 정책이 연결되어 있는지 확인하려면 새 역할 이름으로 iam list-role-policies을 호출합니다.

5. Amazon 에서 사용할 새 역할이 있는 iam create-instance-profile을 사용하여 인스턴스 프로파일을 생성합니다EC2. 자세한 내용은 인스턴스 프로파일 관리를 참조하세요.

```
AWS iam create-instance-profile --instance-profile-name FleetIQ-role-for-EC2
```

요청이 성공하면 새로 생성된 인스턴스 프로파일의 속성이 응답에 포함됩니다.

6. iam add-role-to-instance-profile을 사용하여 역할을 인스턴스 프로파일에 연결합니다.

```
 AWS iam add-role-to-instance-profile --role-name FleetIQ-role-for-EC2 --
instance-profile-name FleetIQ-role-for-EC2
```

이제 Amazon EC2 시작 템플릿과 함께 역할 및 프로필을 사용할 준비가 되었습니다.

Amazon GameLift FleetIQ에 사용되는 게임 준비

이 섹션에서는 Amazon GameLift FleetIQ를 사용하여 Amazon EC2에서 게임을 호스팅하기 위한 설계를 구현하는 방법을 다룹니다. 멀티플레이어 게임을 시작하고 실행하려면 다음 작업을 수행해야 합니다.

- Amazon GameLift FleetIQ와 통신하도록 게임 서버를 조정합니다.
- FleetIQ 게임 서버 그룹을 생성하여 게임 서버를 배포합니다.
- 사용 가능한 게임 서버를 요청하는 기능을 게임 클라이언트 서비스에 추가합니다.

이 섹션의 주제에서는 이 작업을 수행하는 방법을 자세히 설명합니다. 시작하려면 자세한 단계별 가이드를 제공하는 통합 계획을 참조하세요.

주제

- Amazon GameLift FleetIQ 통합 단계
- Amazon GameLift FleetIQ 게임 서버 그룹 관리
- Amazon GameLift FleetIQ를 게임 서버에 통합
- Amazon GameLift FleetIQ를 게임 클라이언트에 통합

Amazon GameLift FleetIQ 통합 단계

이 통합 계획은 Amazon GameLift FleetIQ를 사용하여 Amazon EC2 인스턴스에서 멀티플레이어 게임을 시작하고 실행하기 위한 주요 단계를 설명합니다. 더 많은 게임 호스팅 프로세스를 자동화하는 Amazon GameLift 관리형 호스팅 서비스를 찾고 있다면 Amazon GameLift 개발자 가이드를 참조하세요.

Amazon GameLift FleetIQ 사용을 시작하려면 온프레미스 또는 Amazon EC2 환경에서 실행되는 작동 게임 서버가 있어야 합니다. 게임 서버는 하나 또는 여러 게임 세션을 관리하거나, 하위 프로세스를 생성하거나, 컨테이너 내에서 실행하는 단일 프로세스일 수 있습니다.

1. AWS 계정을 만들고 Amazon GameLift FleetIQ 액세스 권한을 가진 사용자를 설정합니다.

   Amazon GameLift FleetIQ에서 사용할 새 AWS 계정를 만들거나 기존 계정을 선택합니다. Amazon EC2, Auto Scaling 및 게임에 사용되는 기타 AWS 리소스를 관리할 권한을 가진 사용자를 설정합니다. 자세한 지침은 Amazon GameLift FleetIQ AWS 계정 설정 섹션을 참조하세요.
2. IAM 역할을 생성합니다.

Amazon GameLift FleetIQ, Amazon EC2 및 Auto Scaling 리소스가 서로 통신할 수 있도록 하는 역할을 생성합니다. 자세한 내용은 교차 서비스 상호 작용을 위한 IAM 역할 생성 섹션을 참조하세요.

3. Amazon GameLift FleetIQ 기능을 사용하여 AWS SDK 및 AWS CLI를 가져옵니다.
   - AWS SDK의 최신 버전을 다운로드합니다.
   - Amazon GameLift API 참조 문서를 봅니다.
4. Amazon GameLift FleetIQ와 함께 사용할 게임 서버를 준비합니다.

   게임 서버 프로젝트에 AWS SDK를 추가하고 Amazon GameLift FleetIQ가 게임 서버의 현재 상태 및 사용량에 맞게 업데이트되도록 하는 코드를 추가합니다. 추가 지침 및 예제는 the section called "게임 서버 통합" 섹션을 참조하세요. Amazon GameLift FleetIQ는 이 정보를 사용하여 사용 가능하고 점유되지 않은 게임 서버 목록을 매치메이킹 시스템에 제공하고 밸런싱 중에 현재 플레이어를 호스팅하고 있는 인스턴스가 종료되지 않도록 합니다.
5. 게임 서버로 Amazon EC2 Amazon Machine Image(AMI)를 생성합니다.

   게임 서버 소프트웨어 및 기타 런타임 자산 또는 구성 설정을 사용하여 AMI를 생성합니다. 도움이 필요하면 Amazon EC2 사용 설명서의 Amazon Machine Image(AMI)를 참조하세요.
6. Amazon EC2 시작 템플릿을 생성합니다.

   사용자 지정 AMI를 사용하고 호스팅 리소스에 대한 네트워크 및 보안 설정을 정의하는 Amazon EC2 시작 템플릿을 빌드합니다. 시작 템플릿은 게임 서버가 Amazon GameLift FleetIQ와 통신할 수 있도록 허용하는 권한을 생성한 인스턴스 프로파일(2단계 참조)을 참조해야 합니다. 이 작업은 나중에 수행되므로 시작 템플릿에 인스턴스 유형을 포함할 필요는 없습니다. 도움이 필요하면 Amazon EC2 사용 설명서의 시작 템플릿 생성을 참조하세요.

   > **Note**
   >
   > Amazon GameLift FleetIQ에서 시작 템플릿을 사용하려면 먼저 템플릿 구성과 AMI가 제대로 배포되고 있는지 확인하기 위해 오토 스케일링을 설정하는 것이 좋습니다.
7. Amazon GameLift FleetIQ 호스팅 리소스를 설정합니다.

   게임 서버를 배포할 각 리전에서 CreateGameServerGroup()을 호출하여 게임 서버 그룹을 생성합니다. 시작 템플릿(사용자 지정 AMI, 네트워크 및 보안 설정 포함), IAM 역할 및 게임을 실행할 수 있는 인스턴스 유형 목록을 전달합니다. 이 작업을 수행하면 Amazon GameLift FleetIQ가 수정할 수 있는 오토 스케일링이 사용자 AWS 계정에 설정됩니다. 추가 지침 및 예제는 Amazon GameLift FleetIQ 게임 서버 그룹 관리 섹션을 참조하세요.

8. Amazon GameLift FleetIQ를 게임 클라이언트에 통합합니다.

   게임 클라이언트, 매치메이커 또는 게임 서버 용량을 할당하는 기타 백엔드 구성 요소에 AWS SDK를 추가합니다. 게임 유형에 따라 매치메이커가 ListGameServers() 또는 ClaimGameServer()를 호출하여 서버 용량을 확보하고 사용 가능한 게임 서버를 예약할 수 있습니다. 추가 지침 및 예제는 Amazon GameLift FleetIQ를 게임 클라이언트에 통합 섹션을 참조하세요.

9. 오토 스케일링을 확대합니다.

   인스턴스가 오토 스케일링에서 프로비저닝되면 게임 서버가 시작됩니다. 그런 다음 각 게임 서버는 Amazon GameLift FleetIQ에 사용 가능한 용량으로 등록되어 나중에 매치메이커에 의해 나열되거나 클레임됩니다.

10. 게임을 테스트합니다.

    매치메이커를 호출하고 `ClaimGameServer`를 호출하여 서버 용량을 요청합니다. 게임 클라이언트가 게임 서버에 연결할 수 있도록 결과 IP 및 포트를 게임 클라이언트에 다시 전달합니다.

Amazon GameLift FleetIQ 게임 서버 그룹 관리

이 주제에서는 Amazon GameLift FleetIQ 게임 서버 그룹을 설정하는 데 필요한 작업을 설명합니다. 게임 서버 그룹을 생성하면 게임 호스팅을 위한 Amazon GameLift FleetIQ 최적화를 관리하는 구성과 함께 필요한 모든 구성 설정이 포함된 EC2 Auto Scaling 그룹이 생성됩니다.

게임 서버 그룹을 생성하려면 최소한 다음과 같은 리소스를 준비해야 합니다.

- 게임 서버 빌드로 Amazon EC2 인스턴스를 EC2 시작하는 방법을 지정하는 Amazon 시작 템플릿입니다. 자세한 내용은 Amazon EC2 사용 설명서의 시작 템플릿에서 인스턴스 시작을 참조하세요.
- FleetIQ가 Auto Scaling 그룹을 생성하고 상호 작용할 수 있도록 AWS GameLift 계정에 대한 제한된 액세스를 확장하는 IAM 역할입니다. 자세한 내용은 교차 서비스 상호 작용을 위한 IAM 역할 생성 단원을 참조하십시오.

게임 서버 그룹 생성

게임 서버 그룹을 생성하려면 CreateGameServerGroup()를 호출합니다. 이 작업은 Amazon GameLift FleetIQ 게임 서버 그룹과 해당 Auto Scaling 그룹을 모두 생성합니다. 게임 서버 그룹을 생성할 때 밸런싱 전략 및 인스턴스 유형 정의를 포함하여 Amazon GameLift FleetIQ 에 대한 게임별 설정을 제공합니다. 오토 스케일링의 초기 속성 설정도 제공합니다.

다음 예제에서는 c4.large 및 c5.large 인스턴스 유형을 지정하고 그룹을 스팟 인스턴스로만 제한하는 `GameServerGroup`의 생성을 트리거하고, 인스턴스 배포에 지정된 시작 템플릿을 사용하며 대상 추적 자동 크기 조정 정책을 사용하여 최소 및 최대 설정 내에서 그룹 용량을 관리하는 오토 스케일링을 트리거합니다. 프로비저닝 기간이 짧으면 `AutoScalingGroup` 리소스가 생성되고 가 ACTIVE 상태가 `GameServerGroup` 됩니다.

```
AWS gamelift create-game-server-group \
    --game-server-group-name MyLiveGroup \
    --role-arn arn:aws:iam::123456789012:role/GameLiftGSGRole \
    --min-size 1 \
    --max-size 10 \
    --game-server-protection-policy FULL_PROTECTION \
    --balancing-strategy SPOT_ONLY \
    --launch-template LaunchTemplateId=lt-012ab345cde6789ff \
    --instance-definitions '[{"InstanceType": "c4.large"}, {"InstanceType":
 "c5.large"}]' \
    --auto-scaling-policy '{"TargetTrackingConfiguration": {"TargetValue": 66}}'
```

게임 서버 그룹 업데이트

Amazon GameLift FleetIQ가 리소스 유형 최적화를 포함하여 게임 서버의 호스팅을 관리하는 방식에 영향을 미치는 게임 서버 그룹 속성을 업데이트할 수 있습니다. 이러한 속성을 업데이트하려면 UpdateGameServerGroup()를 호출합니다. 게임 서버 그룹에 대한 변경 사항이 적용된 후 Amazon GameLift FleetIQ는 Auto Scaling 그룹의 특정 속성을 덮어쓸 수 있습니다.

`MinSize`, `MaxSize`, `LaunchTemplate` 등의 다른 모든 오토 스케일링 속성의 경우 오토 스케일링에서 직접 수정할 수 있습니다.

아래 예에서 인스턴스 유형 정의는 c4.xlarge 및 c5.xlarge 인스턴스 유형으로 전환하도록 업데이트되었습니다.

```
AWS gamelift update-game-server-group \
    --game-server-group-name MyLiveGroup \
    --instance-definitions '[{"InstanceType": "c4.xlarge"}, {"InstanceType":
 "c5.xlarge"}]'
```

게임 서버 그룹 인스턴스 추적

게임 서버 그룹 및 Auto Scaling 그룹에 인스턴스를 생성하고 배포한 후 DescribeGameServerInstances()를 호출하여 게임 서버 인스턴스의 상태를 추적할 수 있습니다. 이 작

업을 사용하여 인스턴스 상태를 추적할 수 있습니다. 게임 서버 그룹 상태에 대한 자세한 내용은 게임 서버 그룹의 수명 섹션을 참조하세요.

게임 서버 그룹 에서 Amazon GameLift 콘솔 를 사용하여 게임 서버 그룹의 상태를 모니터링할 수도 있습니다.

Amazon GameLift FleetIQ를 게임 서버에 통합

이 주제에서는 Amazon GameLift FleetIQ 와 통신하도록 게임 서버 프로젝트를 준비하는 데 필요한 작업을 설명합니다. 추가 지침은 Amazon GameLift FleetIQ 모범 사례 섹션을 참조하세요.

게임 서버 등록

게임 서버 프로세스가 시작되고 라이브 게임 플레이를 호스팅할 준비가 되면 RegisterGameServer()를 호출하여 Amazon GameLift FleetIQ에 등록해야 합니다. 등록을 통해 Amazon GameLift FleetIQ는 서버 용량에 대한 정보를 요청하거나 게임 서버를 청구할 때 매치메이킹 시스템 또는 기타 클라이언트 서비스에 응답할 수 있습니다. 등록할 때 게임 서버는 Amazon GameLift FleetIQ에 인바운드 클라이언트 연결에 사용하는 포트 및 IP 주소를 비롯한 관련 게임 서버 데이터 및 연결 정보를 제공할 수 있습니다.

```
AWS gamelift register-game-server \
    --game-server-id UniqueId-1234 \
    --game-server-group-name MyLiveGroup \
    --instance-id i-1234567890 \
    --connection-info "1.2.3.4:123" \
    --game-server-data "{\"key\": \"value\"}"
```

게임 서버 상태 업데이트

게임 서버가 등록되면 Amazon GameLift FleetIQ 에서 서버 용량 상태를 동기화 상태로 유지하기 위해 정기적으로 상태 및 사용률을 보고해야 합니다. UpdateGameServer()를 호출하여 상태 및 사용률을 보고합니다. 아래 예에서는 게임 서버가 정상 상태이며 플레이어나 게임 플레이를 현재 호스팅하여 점유되지 않았음을 보고합니다.

```
AWS gamelift update-game-server \
    --game-server-group-name MyLiveGroup \
    --game-server-id UniqueId-1234 \
    --health-check HEALTHY \
    --utilization-status AVAILABLE
```

상태 확인

게임 서버에 상태 추적 메커니즘이 있는 경우 이 메커니즘을 사용하여 Amazon GameLift FleetIQ 에 대한 게임 서버 상태 업데이트를 트리거할 수 있습니다.

사용률 상태

게임 서버 사용률 상태를 보고하면 Amazon GameLift FleetIQ가 현재 어떤 게임 서버가 이상적이며 새 게임 세션에 사용할 수 있는지 알 수 있습니다. 게임 서버에는 Amazon GameLift FleetIQ 에 대한 사용률 상태 업데이트를 트리거하는 메커니즘이 있어야 합니다. 예를 들어 플레이어가 게임 서버에 연결하거나 게임 세션이 시작될 때 업데이트를 트리거할 수 있습니다.

게임 세션을 시작할 때 클라이언트 또는 매치메이킹 서비스는 사용 가능한 게임 서버 (ClaimGameServer()를 호출하여)를 청구하고, 플레이어에게 게임 서버에 연결하도록 지시하고, 게임 서버를 트리거하여 게임 플레이를 시작합니다. 이 프로세스는 Amazon GameLift FleetIQ를 게임 클라이언트에 통합에 설명되어 있습니다. 게임 서버 “클레임”은 60초 동안 유효하며, 게임 서버는 이 창에서 사용 상태를 업데이트할 수 있어야 합니다. 사용률이 업데이트되지 않으면 Amazon GameLift FleetIQ는 클레임을 제거하고 게임 서버를 사용할 수 있다고 가정하며 다른 클라이언트 클레임 요청에 대해 게임 서버를 예약할 수 있습니다.

```
AWS gamelift update-game-server \
    --game-server-group-name MyLiveGroup \
    --game-server-id UniqueId-1234 \
    --health-check HEALTHY \
    --utilization-status UTILIZED
```

게임 서버 등록 취소

게임이 종료되면 게임 서버는 DeregisterGameServer()를 사용하여 Amazon GameLift FleetIQ에서 등록을 취소해야 합니다.

```
AWS gamelift deregister-game-server \
    --game-server-group-name MyLiveGroup \
    --game-server-id UniqueId-1234
```

Amazon GameLift FleetIQ를 게임 클라이언트에 통합

이 주제에서는 게임 클라이언트 또는 매치메이킹 서비스가 게임 세션을 호스팅하기 위해 게임 서버를 획득하기 위해 Amazon GameLift FleetIQ와 통신하도록 준비하는 데 필요한 작업을 설명합니다.

게임 클라이언트 또는 매치메이커가 플레이어를 위한 게임 서버 리소스를 요청할 수 있는 방법을 생성합니다. 이 작업을 수행하는 방법에 대한 몇 가지 옵션이 있습니다.

- Amazon GameLift FleetIQ가 사용 가능한 게임 서버를 선택하도록 합니다. 이 옵션은 Amazon GameLift FleetIQ 최적화를 활용하여 저렴한 스팟 인스턴스를 사용하고 자동 조정을 위해 사용합니다.
- 사용 가능한 모든 게임 서버를 요청하고 사용할 게임 서버 하나를 선택합니다("나열 및 선택"이라고도 함).

주제

- Amazon GameLift FleetIQ가 게임 서버를 선택하도록 허용
- 자체 게임 서버 선택

Amazon GameLift FleetIQ가 게임 서버를 선택하도록 허용

Amazon GameLift FleetIQ가 사용 가능한 게임 서버를 선택하도록 하려면 게임 서버 ID를 지정하지 않고 ClaimGameServer()를 호출합니다. 이 시나리오에서 Amazon GameLift FleetIQ는 로직을 사용하여 게임 호스팅에 실행 가능하고 자동 조정에 최적화된 인스턴스에서 게임 서버를 찾습니다.

```
AWS gamelift claim-game-server \
    --game-server-group-name MyLiveGroup
```

클레임 요청에 대한 응답으로 Amazon GameLift FleetIQ는 클라이언트가 게임 서버에 연결하는 데 사용할 수 있는 `GameServer` 리소스, 연결 정보 및 게임 데이터를 식별합니다. 게임 서버의 클레임 상태는 60초 CLAIMED 동안 로 설정됩니다. 플레이어가 연결하거나 게임 플레이를 시작한 후 게임 서버 또는 클라이언트 서비스가 Amazon GameLift FleetIQ에서 게임 서버의 상태를 업데이트해야 합니다. 이렇게 하면 Amazon GameLift FleetIQ가 후속 게임 서버 용량 요청에 대한 응답으로 이 게임 서버를 제공하지 않습니다. UpdateGameServer()를 호출하여 게임 서버 상태를 업데이트합니다.

```
AWS gamelift update-game-server \
    --game-server-group-name MyLiveGroup \
    --game-server-id UniqueId-1234 \
    --health-check HEALTHY \
    --utilization-status UTILIZED
```

자체 게임 서버 선택

“목록 및 선택” 메서드를 사용하면 게임 클라이언트 또는 매치메이커가 ListGameServers()를 호출하여 사용 가능한 게임 서버 목록을 요청합니다. 게임 서버 데이터를 사용하여 플레이어 또는 매치메이커가 게임 서버를 선택할 때 사용할 수 있는 추가 정보를 제공할 수 있습니다. 결과가 반환되는 방식을 제어하려면 페이지가 매겨진 결과를 요청하고 등록일별로 게임 서버를 정렬할 수 있습니다. 다음 요청은 지정된 게임 서버 그룹에서 사용 가능한 활성 게임 서버 20개를 반환하고, 등록 시간을 기준으로 정렬하여 최신 게임 서버를 먼저 나열합니다.

```
AWS gamelift list-game-servers \
    --game-server-group-name MyLiveGroup \
    --limit 20 \
    --sort-order DESCENDING
```

클라이언트 또는 매치메이킹 서비스는 사용 가능한 게임 서버 목록을 기반으로 게임 서버를 선택하고 특정 게임 서버 ID로 ClaimGameServer()를 호출하여 청구합니다. 이 시나리오에서 Amazon GameLift FleetIQ는 에 설명된 대로 인스턴스 유형 최적화 로직을 실행하지 않습니다Amazon GameLift FleetIQ 로직.

```
AWS gamelift claim-game-server \
    --game-server-group-name MyLiveGroup \
    --game-server-id UniqueId-1234
```

Amazon CloudWatch를 사용한 Amazon GameLift FleetIQ 모니터링

Amazon CloudWatch 지표를 사용하여 인스턴스 용량을 조정하고, 운영 대시보드를 구축하고, 경보를 트리거하는 데 도움이 됩니다. 독립형 솔루션인 Amazon GameLift FleetIQ는 Amazon CloudWatch 지표 세트를 AWS 계정으로 내보냅니다. 자세한 내용은 Amazon EC2 Auto Scaling 사용 설명서의 Amazon CloudWatch를 사용하여 오토 스케일링 및 인스턴스 모니터링을 참조하세요.

FleetIQ 지표는 여기에 나열됩니다. Amazon GameLift 지표에서 Amazon GameLift에 대한 전체 Amazon CloudWatch 지표 정보를 확인합니다.

지표	설명
`AvailableGameServers`	게임을 실행하는 데 사용할 수 있고 현재 게임 플레이로 점유되지 않은 게임 서버 수입니다. 여기에는 클레임되었지만 아직 AVAILABLE 상태인 게임 서버도 포함됩니다. 단위: 개수 관련 CloudWatch 통계: Sum 차원: GameServerGroup
`UtilizedGameServers`	현재 게임 플레이로 점유된 게임 서버 수입니다. 이 숫자에는 UTILIZED 상태인 게임 서버가 포함됩니다. 단위: 개수 관련 CloudWatch 통계: Sum 차원: GameServerGroup
`DrainingAvailableGameServers`	현재 게임 플레이를 지원하지 않으며 종료 예정된 인스턴스의 게임 서버 수입니다. 이러한 게임 서버는 새 클레임 요청에 대한 응답으로 클레임되는 우선 순위가 가장 낮습니다. 단위: 개수

지표	설명
	관련 CloudWatch 통계: Sum 차원: GameServerGroup
`DrainingUtilizedGameServers`	현재 게임 플레이를 지원하며 종료 예정된 인스턴스의 게임 서버 수입니다. 단위: 개수 관련 CloudWatch 통계: Sum 차원: GameServerGroup
`PercentUtilizedGameServers`	현재 게임 실행을 지원하는 게임 서버의 비율입니다. 이 지표는 현재 사용 중인 게임 서버 용량을 나타냅니다. 플레이어 수요에 맞게 인스턴스를 동적으로 추가 및 제거할 수 있는 Auto Scaling 정책을 실행하는 데 유용합니다. 단위: 백분율 상대적 CloudWatch 통계: Average, Minimum, Maximum 차원: GameServerGroup
`GameServerInterruptions`	제한된 스팟 가용성으로 인해 중단된 스팟 인스턴스의 게임 서버입니다. 단위: 개수 관련 CloudWatch 통계: Sum 차원: GameServerGroup, InstanceType

지표	설명
`InstanceInterruptions`	제한된 가용성으로 인해 중단된 스팟 인스턴스입니다. 단위: 개수 관련 CloudWatch 통계: Sum 차원: GameServerGroup, InstanceType

아마존 GameLift FleetiQ를 통한 보안

Amazon GameLift FleetiQ를 Amazon EC2에서 독립형 기능으로 사용하는 경우, Amazon EC2 사용 설명서의 Amazon EC2의 보안을 참조하십시오.

클라우드 보안은 최우선 과제입니다. AWS AWS 고객은 보안에 가장 보안에 민감한 조직의 요구 사항에 부합하도록 구축된 데이터 센터 및 네트워크 아키텍처의 혜택을 누릴 수 있습니다.

보안은 사용자와 사용자 AWS 간의 공동 책임입니다. Amazon GameLift FleetiQ를 사용할 때 공동 책임 모델을 적용하는 방법에 대한 자세한 내용은 Amazon의 보안을 참조하십시오. GameLift

Amazon GameLift FleetIQ 참조 가이드

이 섹션에는 Amazon GameLift FleetIQ 와 함께 사용하기 위한 참조 설명서가 포함되어 있습니다.

주제

- Amazon GameLift FleetIQ 서비스 API 참조(AWS SDK)
- Amazon GameLift FleetIQ 릴리스 정보 및 SDK 버전
- Amazon GameLift 개발자 리소스

Amazon GameLift FleetIQ 서비스 API 참조(AWS SDK)

이 주제에서는 Amazon GameLift FleetIQ 에 대한 작업 기반 API 작업 목록을 제공합니다. Amazon GameLift FleetIQ 서비스는 `aws.gamelift` 네임스페이스의 AWS SDK에 패키징API됩니다. 를 다운로드 AWS SDK하거나 Amazon GameLift API 참조 설명서 를 봅니다.

Amazon GameLift FleetIQ는 Amazon 를 통한 클라우드 기반 게임 호스팅을 위해 저렴한 스팟 인스턴스 사용을 최적화합니다EC2. 다른 Amazon 호스팅 옵션에 대한 자세한 내용은 Amazon GameLift 개발자 안내서를 참조하세요. GameLift

주제

- Amazon GameLift FleetIQ API 작업
- 사용 가능한 프로그래밍 언어

Amazon GameLift FleetIQ API 작업

다음 작업을 통해 Amazon EC2 및 Auto Scaling 그룹과 함께 게임 서버 그룹 및 게임 서버를 포함한 Amazon GameLift FleetIQ 리소스를 관리할 수 있습니다.

게임 서버 그룹 관리

이러한 작업을 사용하여 FleetIQ 최적화로 게임 서버 배포를 관리할 수 있습니다. 게임 서버 그룹은 Amazon EC2 인스턴스에서 게임 서버 프로세스가 시작되는 방식을 제어하고, 및 Auto Scaling 그룹을 설정하고, FleetIQ 최적화를 적용하는 방법을 정의합니다.

- CreateGameServerGroup - 새 게임 서버 그룹과 해당 Auto Scaling 그룹을 생성하고 게임 서버를 호스팅하기 위한 인스턴스 시작을 시작합니다. CLI 명령: create-game-server-group

- ListGameServerGroups - Amazon GameLift 리전의 모든 게임 서버 그룹 목록을 가져옵니다. CLI 명령: list-game-server-groups
- DescribeGameServerGroup - 게임 서버 그룹의 메타데이터를 검색합니다. CLI 명령: describe-game-server-group
- UpdateGameServerGroup - 게임 서버 그룹 메타데이터를 변경합니다. CLI 명령: update-game-server-group
- DeleteGameServerGroup - 게임 서버 그룹을 영구적으로 제거하고 연결된 호스팅 리소스에 대한 FleetIQ 활동을 종료합니다. CLI 명령: delete-game-server-group
- ResumeGameServerGroup – 게임 서버 그룹에 대해 일시 중지된 FleetIQ 활동을 복원합니다. CLI 명령: resume-game-server-group
- SuspendGameServerGroup – 게임 서버 그룹에 대한 FleetIQ 활동을 일시적으로 중지합니다. CLI 명령: suspend-game-server-group

게임 서버 관리

이러한 작업을 사용하여 FleetIQ 최적화로 게임 서버 배포를 관리할 수 있습니다. 게임 서버 그룹은 Amazon EC2 인스턴스에서 게임 서버 프로세스가 시작되는 방식을 제어하고, 및 Auto Scaling 그룹을 설정하고, FleetIQ 최적화를 적용하는 방법을 정의합니다.

- RegisterGameServer - 새 게임 서버에서 를 호출하여 게임 서버가 게임 플레이를 호스팅할 준비가 되었음을 Amazon GameLift FleetIQ에 알립니다. CLI 명령: register-game-server-group
- ListGameServers - 게임 클라이언트 서비스에서 를 호출하여 현재 게임 서버 그룹에서 실행 중인 모든 게임 서버 목록을 가져옵니다. CLI 명령: list-game-servers
- ClaimGameServer - 게임 클라이언트 서비스에서 를 호출하여 새 게임 세션을 호스팅할 게임 서버를 찾아 예약합니다. CLI 명령: claim-game-server
- DescribeGameServer - 게임 서버의 메타데이터를 검색합니다. CLI 명령: describe-game-server
- UpdateGameServer - 게임 서버 메타데이터, 상태 또는 사용률 상태를 변경합니다. CLI 명령: update-game-server
- DeregisterGameServer - 종료 게임 서버에서 를 호출하여 Amazon GameLift FleetIQ에 게임 서버 그룹에서 게임 서버를 제거하라는 메시지를 표시합니다. CLI 명령: deregister-game-server

사용 가능한 프로그래밍 언어

Amazon을 AWS SDK 지원하는 GameLift 는 다음 언어로 제공됩니다. 개발 환경 지원에 대한 자세한 내용은 각 언어로 된 설명서를 참조하세요.

- C++(SDK 문서)(Amazon GameLift)
- Java(SDK 문서)(Amazon GameLift)
- .NET (SDK 문서) (Amazon GameLift)
- Go(SDK 문서)(Amazon GameLift)
- Python(SDK 문서)(Amazon GameLift)
- Ruby(SDK 문서)(Amazon GameLift)
- PHP (SDK 문서) (Amazon GameLift)
- JavaScript/Node.js(SDK 문서)(Amazon GameLift)

Amazon GameLift FleetIQ 릴리스 정보 및 SDK 버전

Amazon GameLift 릴리스 정보에서는 서비스와 관련된 새로운 FleetIQ 기능, 업데이트 및 수정 사항에 대한 세부 정보를 제공합니다. 이 페이지에는 Amazon GameLift SDK 버전 기록도 포함되어 있습니다.

Amazon GameLift 개발자 리소스

모든 Amazon GameLift 설명서 및 개발자 리소스를 보려면 Amazon GameLift 설명서 홈 페이지를 참조하세요.

AWS 용어집

최신 AWS 용어는 AWS 용어집 참조서의 AWS 용어집을 참조하세요.

기계 번역으로 제공되는 번역입니다. 제공된 번역과 원본 영어의 내용이 상충하는 경우에는 영어 버전이 우선합니다.