
AWS 시작

WordPress 웹 사이트 구축



AWS 시작: WordPress 웹 사이트 구축

Copyright © 2017 Amazon Web Services, Inc. and/or its affiliates. All rights reserved.

Amazon's trademarks and trade dress may not be used in connection with any product or service that is not Amazon's, in any manner that is likely to cause confusion among customers, or in any manner that disparages or discredits Amazon. All other trademarks not owned by Amazon are the property of their respective owners, who may or may not be affiliated with, connected to, or sponsored by Amazon.

Table of Contents

| | |
|--|----|
| AWS 상에서 WordPress 구축 | 1 |
| 아키텍처 | 1 |
| 자습서 | 2 |
| 요금 | 2 |
| 1단계: 정적 자산을 위한 스토리지 준비 | 3 |
| AWS에 가입 | 3 |
| Amazon S3 버킷 만들기 | 3 |
| IAM 사용자 생성 | 4 |
| 2단계: 데이터베이스 생성 | 6 |
| 3단계: WordPress 다운로드 | 8 |
| 4단계: WordPress 배포 | 10 |
| 사전 조건 | 10 |
| Elastic Beanstalk를 사용하여 애플리케이션 서버 시작 | 10 |
| 구성 업데이트 | 12 |
| WordPress 설치 | 13 |
| Amazon S3에 자산을 저장하도록 WordPress 설정 | 14 |
| Amazon S3에 WordPress 자산 저장 | 16 |
| 사용자 지정 도메인 이름 사용 | 17 |
| 5단계: 애플리케이션 버전 업데이트 | 18 |
| 사전 조건 | 18 |
| Elastic Beanstalk를 사용하여 새 애플리케이션 버전 추가 | 18 |
| 6단계: 정리 | 20 |
| Elastic Beanstalk 리소스 삭제 | 20 |
| Amazon RDS 데이터베이스 삭제 | 20 |
| Amazon S3 버킷 삭제 | 21 |

Amazon Web Services 상에서 WordPress 구축

Amazon Web Services(AWS)는 종량 과금방식의 가격 정책과 함께 클라우드에서 주문형 컴퓨팅 리소스 및 서비스를 제공합니다.

AWS Elastic Beanstalk을 사용하면 몇 분 만에 AWS 상에 WordPress 애플리케이션을 배포할 수 있습니다. Elastic Beanstalk은 애플리케이션 서버와 같은 AWS 리소스 프로비저닝, 로드 밸런싱 구성, 크기 조정 및 모니터링 등의 호스팅 환경 세부 작업을 처리합니다.

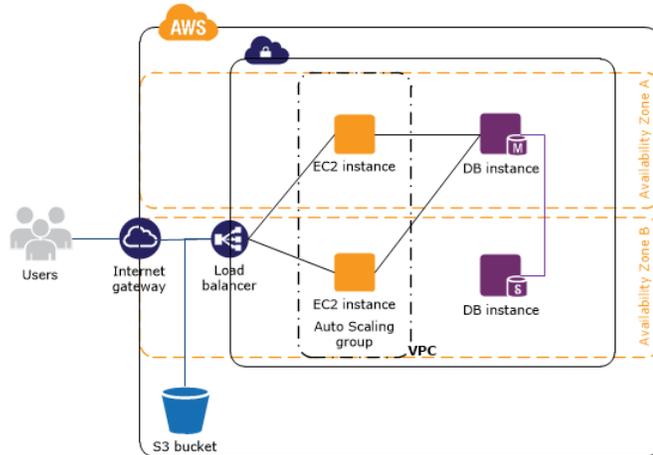
아키텍처

이 구현 작업에서는 다음 AWS 리소스를 사용합니다.

- 인스턴스라고 하는 Amazon Elastic Compute Cloud(Amazon EC2)의 애플리케이션 서버
- 버킷이라고 하는 Amazon Simple Storage Service(Amazon S3)의 스토리지 공간
- DB 인스턴스라고 하는 Amazon Relational Database Service(Amazon RDS)의 관리되는 관계형 데이터베이스
- 트래픽을 애플리케이션 서버로 분산하는 Elastic Load Balancing 서비스의 로드 밸런서
- 최소 개수의 사용 가능한 애플리케이션 서버를 보유하고, WordPress 웹 사이트 또는 블로그의 수요 변동에 따라 애플리케이션 서버를 추가하거나 제거할 수 있도록 해주는 Auto Scaling 서비스의 조정 서비스

WordPress가 콘텐츠 전송 네트워크(CDN)로 Amazon S3를 사용하도록 해주는 WordPress 플러그인도 설치할 수 있습니다.

다음 다이어그램은 이 자습서의 아키텍처를 보여 줍니다.



자습서

본 자습서에서는 AWS 상에서 WordPress 애플리케이션을 배포하는 절차를 단계별로 살펴봅니다. AWS Management Console을 사용하여 AWS에 액세스합니다.

단계

1. 정적 자산을 위한 스토리지 준비 (p. 3)
2. 데이터베이스 생성 (p. 6)
3. WordPress 다운로드 (p. 8)
4. WordPress 배포 (p. 10)
5. 애플리케이션 버전 업데이트 (p. 18)
6. 정리 (p. 20)

요금

[AWS 월 사용량 계산기](#)를 이용해 AWS에서의 WordPress 호스팅 사용료를 예측할 수 있습니다.

AWS 계정을 생성한지 12개월이 지나지 않았다면 [AWS 프리 티어](#)에 해당됩니다.

이 자습서의 예측 비용에 대한 자세한 내용은 [WordPress 웹 사이트 구축: 사용된 서비스 및 비용](#)을 참조하십시오. AWS 요금에 대한 자세한 내용은 [요금](#)을 참조하십시오.

1단계: 정적 자산을 위한 스토리지 준비

미디어 라이브러리 및 테마 파일과 같은 WordPress 웹 사이트의 정적 자산을 Amazon S3에 저장할 수 있습니다. S3 버킷을 생성하고 S3 버킷에 자산을 저장할 수 있는 권한이 있는 AWS Identity and Access Management(IAM) 사용자를 생성해야 합니다.

AWS에 가입

Amazon Web Services(AWS)에 가입하면 AWS의 모든 서비스에 AWS 계정이 자동으로 등록되어 곧바로 사용할 수 있습니다. 사용한 서비스에 대해서만 청구됩니다.

AWS 계정을 생성하고 12개월이 지나지 않은 경우에는 AWS를 무료로 시작할 수 있습니다. 자세한 내용은 [AWS 프리 티어](#) 섹션을 참조하십시오.

AWS 계정이 아직 없는 경우에는 아래 단계를 수행하여 계정을 만드십시오.

AWS 계정을 만들려면 다음을 수행합니다.

1. <https://aws.amazon.com/>을 열고 [Create an AWS Account]를 선택합니다.
2. 온라인 지시 사항을 따릅니다.

등록 절차 중 전화를 받고 전화 키패드를 사용하여 PIN을 입력하는 과정이 있습니다.

Amazon S3 버킷 만들기

AWS Management Console을 사용하여 WordPress 웹 사이트의 정적 자산에 대해 버킷을 생성합니다.

创建 S3 存储桶

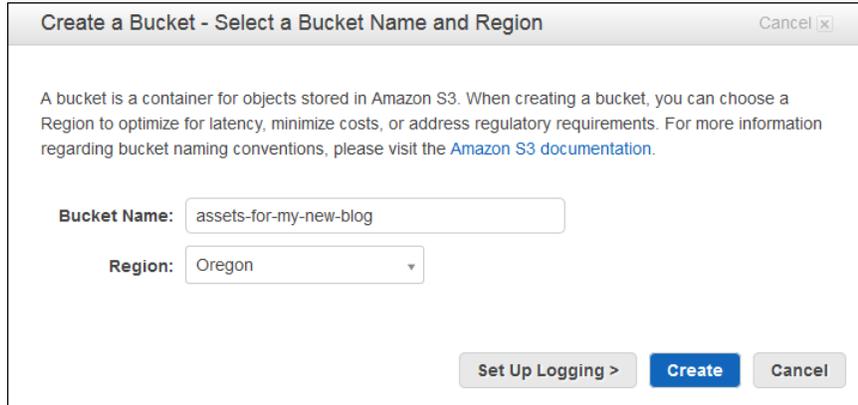
1. <https://console.aws.amazon.com/s3/>에서 Amazon S3 콘솔을 엽니다.
2. Create Bucket을 선택합니다.
3. Bucket Name에서 버킷 이름을 입력합니다.

4. Region에서, 가까운 리전을 선택합니다. 리전이 가까우면 지연 시간이 줄어들고 비용이 최소화됩니다.

Warning

WordPress가 CDN으로 Amazon S3를 사용할 수 있도록 하는 데 사용할 WordPress 플러그인은 2014년 1월 이후 도입된 리전은 지원하지 않습니다. 지원되지 않는 리전을 선택하는 경우 이 자습서를 마칠 수 없습니다. 자세한 내용은 [W3 Total Cache는 최신 S3 리전과 호환되지 않음](#)을 참조하십시오.

5. Create를 선택합니다.



IAM 사용자 생성

모든 S3 버킷을 나열하고 방금 만든 S3 버킷에 대한 전체 액세스 권한을 가질 수 있는 IAM 사용자를 생성합니다. 이 IAM 사용자에게 대한 자격 증명을 WordPress 웹 사이트의 정적 자산을 Amazon S3에 저장하도록 해주는 WordPress 플러그인에 제공할 것입니다.

1. <https://console.aws.amazon.com/iam/>에서 IAM 콘솔을 엽니다.
2. 탐색 창에서 Policies를 선택한 후 Create Policy를 선택합니다.
3. Create Your Own Policy에서 Select 버튼을 선택합니다.
4. Policy Name에서, 정책 이름을 입력합니다(예: AllowWordPressBucketAccess). Policy Document에서, 다음 정책 문서를 복사하여 붙여 넣고, assets-for-my-new-blog를 S3 버킷 이름으로 바꾼 다음, Create Policy를 선택합니다.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": ["s3:ListAllMyBuckets"],
      "Resource": "*"
    },
    {
      "Effect": "Allow",
      "Action": ["s3:*"],
      "Resource": [
        "arn:aws:s3:::assets-for-my-new-blog",
        "arn:aws:s3:::assets-for-my-new-blog/*"
      ]
    }
  ]
}
```

5. 탐색 창에서 Users와 Add user를 차례로 선택합니다.

6. 사용자 이름(예: wp-s3-user)을 입력하고, Programmatic access를 선택한 다음, Next: Permissions를 선택합니다.
7. Attach existing policies directly를 선택하고, 생성한 정책의 확인란을 선택한 다음, Next: Review를 선택합니다.
8. Create user를 선택합니다.
9. 지침을 따라 이 사용자의 액세스 키를 안전한 위치에 보관합니다.
10. Close를 선택합니다.

2단계: 데이터베이스 생성

Amazon RDS는 관리되는 관계형 데이터베이스를 제공합니다. 기본 구성 요소는 DB 인스턴스이며, AWS 클라우드의 격리된 데이터베이스 환경입니다. 한 DB 인스턴스에 여러 개의 데이터베이스가 포함될 수 있습니다.

이 단계에서는 Multi-AZ DB instance를 시작합니다. 다중 AZ 배포에서 Amazon RDS는 자동으로 서로 다른 가용 영역에 동기식 예비 복제본을 프로비저닝하고 유지합니다. 기본 DB 인스턴스는 가용 영역에서 예비 복제본으로 동기식으로 복제되어 데이터 이중화를 제공하고 I/O 중지를 제거하며 시스템 백업 시 지연 시간 스파이크를 최소화합니다. DB 인스턴스에 계획되거나 계획되지 않은 중단이 발생하면 Amazon RDS는 자동으로 예비 복제본으로 전환합니다.

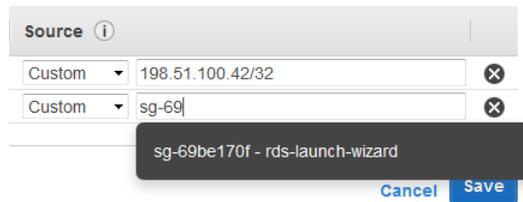
DB 인스턴스를 시작하려면

1. <https://console.aws.amazon.com/rds/>에서 Amazon RDS 콘솔을 엽니다.
2. 탐색 모음에서 S3 버킷을 생성하는 데 사용한 것과 동일한 리전을 선택합니다.
3. 탐색 창에서 Instances를 선택합니다.
4. Launch DB Instance를 선택합니다.
5. Select Engine 페이지의 MySQL 탭에서, Select를 선택합니다.
6. Do you plan to use this database for production purposes 페이지의 Production에서 MySQL을 선택한 다음 Next Step을 선택합니다.
7. Specify DB Details 페이지에서 다음 작업을 수행합니다.
 - a. (선택 사항) DB Instance Class에서 생산 워크로드의 요구 사항을 만족할 만큼 충분히 큰 인스턴스 크기를 선택합니다. 처음에는 db.r3.large로 시작하고 로드 테스트를 수행하여 이 크기가 요구 사항을 만족하는지 여부를 확인하는 것이 좋습니다.
 - b. Multi-AZ Deployment에는 Yes를 선택하는 것이 좋습니다.
 - c. DB Instance Identifier에, 인스턴스의 고유 이름을 입력합니다(예: `my-db-instance-for-wordpress`).
 - d. Master Username에, 마스터 사용자의 로그인을 입력합니다(예: `master_user`).
 - e. Master Password 및 Confirm Password에서 마스터 사용자의 암호를 입력합니다. 암호를 안전한 곳에 기록해 둡니다.
 - f. Next Step(다음 단계)을 클릭합니다.
8. Configure Advanced Settings 페이지에서 다음 작업을 수행하십시오.
 - a. 기본 Virtual Private Cloud(VPC) 및 기본 서브넷 그룹을 선택된 상태 그대로 둡니다.
 - b. Publicly Accessible에서 No를 선택합니다.
 - c. Database Name에, 최초 데이터베이스 이름을 입력합니다(예: `my_wordpress_database`).

- d. Enhanced Monitoring에서, No를 선택합니다.
 - e. Launch DB Instance를 선택합니다.
9. 몇 분 후 시작 프로세스가 완료됩니다. 인스턴스가 생성되고 있다는 메시지가 표시되면 View Your DB Instances를 선택합니다.

처음에는 DB 인스턴스의 상태가 `creating`으로 되어 있습니다. `available` 상태로 변경된 이후에 DB 인스턴스를 사용할 수 있습니다.

10. Endpoint를 끝부분의 포트 정보를 빼고 복사하여(예: :3306 생략), 이 자습서의 뒷부분에서 사용할 수 있도록 저장해 둡니다.
11. Details 탭의 Security and Network에서, 보안 그룹을 보려는 링크를 Amazon EC2 콘솔에서 엽니다. Inbound 탭에서 Edit를 선택합니다. 기본적으로 선택한 DB 엔진에 따른 규칙이 있습니다. 동일한 Type의 또 다른 규칙을 추가합니다. Source에, 보안 그룹 자체의 ID를 지정합니다. ID 입력을 시작한 후 목록에서 보안 그룹을 선택하면 됩니다. 이렇게 하면 보안 그룹과 연결된 리소스가 데이터베이스 포트에서 데이터베이스로부터 트래픽을 받을 수 있습니다. Save를 선택합니다.



3단계: WordPress 다운로드

AWS Elastic Beanstalk을 사용하여 WordPress 배포를 준비하려면, WordPress 파일을 컴퓨터에 복사하고 일부 구성 정보를 제공해야 합니다. AWS Elastic Beanstalk은 ZIP 또는 WAR 파일 형식의 소스 번들만 지원합니다.

WordPress를 다운로드하고 소스 번들을 생성하려면

1. <http://wordpress.org/download/>를 엽니다.
2. 최신 릴리스를 다운로드합니다.
3. 다운로드의 파일을 로컬 컴퓨터의 폴더에 압축 해제합니다.
4. wp-config-sample.php 파일의 복사본을 생성하고 이름을 wp-config.php로 지정합니다. 이를 통해 새 구성 파일을 생성하고 원본 샘플 파일을 이전 상태 그대로 백업으로 보존할 수 있습니다.
5. 텍스트 편집기에서 wp-config.php 파일을 엽니다.
6. WordPress에서 Amazon RDS 데이터베이스를 사용하도록 구성하려면, 다음 데이터베이스 설정을 바꿉니다.

```
define('DB_NAME', 'database_name_here');
define('DB_USER', 'username_here');
define('DB_PASSWORD', 'password_here');
define('DB_HOST', 'localhost');
```

데이터베이스 설정을 복사하여 붙여 넣습니다. 이 설정에서 사용하는 환경 변수는 Amazon RDS 데이터베이스의 연결 정보를 WordPress에 제공하도록 설정될 것입니다.

```
define('DB_NAME', $_SERVER["RDS_DB_NAME"]);
define('DB_USER', $_SERVER["RDS_USERNAME"]);
define('DB_PASSWORD', $_SERVER["RDS_PASSWORD"]);
define('DB_HOST', $_SERVER["RDS_HOSTNAME"]);
```

7. WordPress 사이트의 보안을 강화하려면, 다음 인증 설정을 바꿉니다.

```
define('AUTH_KEY', 'put your unique phrase here');
define('SECURE_AUTH_KEY', 'put your unique phrase here');
define('LOGGED_IN_KEY', 'put your unique phrase here');
define('NONCE_KEY', 'put your unique phrase here');
define('AUTH_SALT', 'put your unique phrase here');
define('SECURE_AUTH_SALT', 'put your unique phrase here');
define('LOGGED_IN_SALT', 'put your unique phrase here');
define('NONCE_SALT', 'put your unique phrase here');
```

다음 인증 설정을 복사하여 붙여 넣습니다. 이 설정에서 사용하는 환경 변수는 정보를 WordPress에 제공하도록 설정될 것입니다.

```
define('AUTH_KEY',          $_SERVER["AUTH_KEY"]);
define('SECURE_AUTH_KEY',   $_SERVER["SECURE_AUTH_KEY"]);
define('LOGGED_IN_KEY',     $_SERVER["LOGGED_IN_KEY"]);
define('NONCE_KEY',         $_SERVER["NONCE_KEY"]);
define('AUTH_SALT',         $_SERVER["AUTH_SALT"]);
define('SECURE_AUTH_SALT',  $_SERVER["SECURE_AUTH_SALT"]);
define('LOGGED_IN_SALT',    $_SERVER["LOGGED_IN_SALT"]);
define('NONCE_SALT',        $_SERVER["NONCE_SALT"]);
```

8. wp-config.php 파일을 저장하고 텍스트 편집기를 닫습니다.
9. 운영 체제에 따라 다음 방법 중 하나를 사용하여 WordPress 폴더(상위 디렉터리 아님)의 파일 및 폴더에서 ZIP 파일을 만듭니다.
 - Windows - Windows 탐색기에서, 파일 및 폴더를 선택하고, 마우스 오른쪽 버튼을 누른 다음, Send to, Compressed (zipped) Folder를 선택합니다. 파일 이름을 wordpress-x.y.z.zip으로 지정합니다. 여기서, x.y.z는 WordPress의 버전입니다.
 - Mac OS X 및 Linux - 다음 명령을 사용합니다. 여기서, x.y.z는 WordPress의 버전입니다.

```
zip -r ../wordpress-x.y.z.zip .
```

4단계: WordPress 배포

AWS Elastic Beanstalk은 AWS에서 WordPress 사이트를 쉽게 배포, 관리 및 확장하도록 해줍니다.

Note

이 자습서에서는 퍼블릭 엔드포인트에 HTTP를 사용하여 WordPress를 설치합니다. 보안을 강화하려면, 로컬 서버에 먼저 WordPress를 설치한 다음, 관리자 암호를 설정한 후에 Elastic Beanstalk를 사용하여 배포합니다.

사전 조건

- Linux 인스턴스용 Amazon EC2 사용 설명서의 [키 페어 만들기](#) 단원에 설명된 대로 키 페어를 만듭니다. 애플리케이션 서버를 구성할 경우 키 페어를 지정해야 하며, 키 페어를 지정하지 않으면 연결할 수 없습니다.
- [2단계: 데이터베이스 생성 \(p. 6\)](#) 단원의 지침대로 Amazon RDS를 사용하여 DB 인스턴스를 만듭니다. Elastic Beanstalk가 데이터베이스 연결 정보를 WordPress에 전달할 때 사용할 환경 속성은 사용자가 구성합니다.
- [3단계: WordPress 다운로드 \(p. 8\)](#) 단원의 설명대로 WordPress용 ZIP 파일을 만듭니다. 애플리케이션 서버를 시작할 때 이 파일을 지정해야 합니다.

Elastic Beanstalk를 사용하여 애플리케이션 서버 시작

애플리케이션 버전 정보를 Elastic Beanstalk에 제공합니다. Elastic Beanstalk는 EC2 인스턴스와 같은 WordPress를 실행하는 데 필요한 AWS 리소스를 프로비저닝합니다.

Important

새 환경 마법사에 다음 절차에서 설명하는 화면이 표시되지 않는 경우 새 환경 마법사의 미리 보기를 사용하고 있는 것입니다. 미리 보기 마법사 사용 지침은 이 절차 다음에 나오는 절차에 설명되어 있습니다.

Elastic Beanstalk를 사용하여 WordPress를 시작하려면

1. <https://console.aws.amazon.com/elasticbeanstalk/>에서 Elastic Beanstalk 콘솔을 엽니다.

2. 탐색 모음에서 S3 버킷을 생성하는 데 사용한 것과 동일한 리전을 선택합니다.
3. Create New Application을 선택합니다.
4. 애플리케이션의 이름과 설명을 입력하고 Next를 선택합니다.
5. New Environment 페이지에서 Create web server를 선택합니다.
6. Predefined configuration에서, PHP를 선택합니다. Environment type에서, Load balancing, auto scaling을 그대로 둡니다. 로드 밸런싱 및 자동 조정을 사용하면 웹 사이트로 들어오는 웹 트래픽이 가상 서버군으로 라우팅되도록 해주므로 웹 사이트 수요 변동에 따라 자동으로 조정이 이루어집니다. Next를 선택합니다.
7. 최초 애플리케이션 버전의 소스로 ZIP 파일을 업로드한 다음 Next를 선택합니다.
8. 환경의 이름과 URL을 입력합니다. Check availability를 선택하여 이 이름을 사용할 수 있는지 확인한 다음, Next를 선택합니다.
9. Additional Resources 페이지에서 Next를 선택합니다.

Warning

프로덕션 WordPress 웹 사이트에서는, Elastic Beanstalk를 사용하여 DB 인스턴스를 만들지 않는 것이 좋습니다. Elastic Beanstalk가 DB 인스턴스를 만들면 환경을 종료할 때 DB 인스턴스가 삭제되며 데이터를 잃게 됩니다.

10. Configuration details 페이지에서 다음을 수행합니다.
 - a. Instance type에서, t2.large를 선택합니다.
 - b. EC2 key pair에서, 키 페어를 선택합니다.
 - c. Application health check URL에서, /readme.html을 입력하여 포트 80에서 HTTP를 사용하여 상태 확인 요청을 애플리케이션 루트 디렉터리의 해당 파일로 보내도록 로드 밸런서를 구성합니다.
 - d. Next를 선택합니다.
11. (선택 사항) Environment Tags 페이지에서, 환경을 태그하거나 다음 단계를 계속할 수 있습니다. 태그를 추가하려면 해당 키 및 값을 지정합니다. 마쳤으면 Next를 선택합니다.
12. Permissions 페이지에서 Next를 선택합니다.
13. Review 페이지에서 Launch를 선택합니다.
14. Elastic Beanstalk는 AWS 리소스를 만들면서 해당 정보를 Recent Events 아래에 추가합니다.

시작 프로세스가 완료되면 다음 단계를 계속합니다.

15. 탐색 창에서 Configuration을 선택합니다.
16. Scaling에서 기어 모양 아이콘을 선택합니다.
17. 환경 업데이트가 완료되면 다음 절차를 계속합니다.

미리 보기 마법사를 사용하고 있는 경우 다음 절차를 사용합니다.

Elastic Beanstalk를 사용하여 WordPress를 시작하려면(미리 보기 마법사)

1. <https://console.aws.amazon.com/elasticbeanstalk/>에서 Elastic Beanstalk 콘솔을 엽니다.
2. 탐색 모음에서 S3 버킷을 생성하는 데 사용한 것과 동일한 리전을 선택합니다.
3. Create New Application을 선택합니다.
4. 애플리케이션의 이름과 설명을 입력하고 Create를 선택합니다.
5. Actions 메뉴에서 Create New Environment를 선택합니다.
6. Web server environment를 선택한 후 Select를 선택합니다.
7. Create a new environment 페이지에서 다음을 수행합니다.
 - a. Platform에서 PHP를 선택합니다.
 - b. Application code에서, Upload your code를 선택한 다음 Upload를 선택합니다.
 - c. Browse를 선택하고, 사이트의 소스로 생성한 ZIP 파일을 선택한 다음, Upload를 선택합니다.

- d. Configure more options를 선택합니다.
8. 다음과 같이 구성 설정을 지정합니다.
 - a. Configuration presets에서, High availability를 선택합니다.
 - b. Environment settings에서, Modify를 선택하여 Environment settings 패널을 엽니다. Name에, 환경 이름을 입력합니다. Save를 선택합니다.
 - c. Instances 패널을 엽니다. Instance type에서, t2.large를 선택합니다. Save를 선택합니다.
 - d. Capacity 패널을 엽니다. Instances에서, Min을 2로 늘리고 Max는 4를 그대로 유지합니다. Save를 선택합니다.
 - e. Security 패널을 엽니다. EC2 key pair에서, 키 페어를 선택합니다. Save를 선택합니다.
 - f. Monitoring 패널을 엽니다. Health check path에서, /readme.html을 입력하여 포트 80에서 HTTP를 사용하여 상태 확인 요청을 애플리케이션 루트 디렉터리의 해당 파일로 보내도록 로드 밸런서를 구성합니다. Save를 선택합니다.
 - g. Network 패널을 엽니다. VPC에서, 기본 VPC를 선택합니다. Load balancer subnets에서 및 Instance subnets에서 모든 서브넷을 선택합니다. 이렇게 하면 사이트의 가용성이 향상됩니다. Instance security groups에서 보안 그룹을 선택합니다. Save를 선택합니다.
9. Create environment를 선택합니다.
10. Elastic Beanstalk는 AWS 리소스를 만들면서 해당 정보를 상태 창에 추가합니다. 환경이 성공적으로 시작되었는지 확인하십시오. 경고나 오류가 있는 경우, 보고된 문제점을 수정하십시오. 시작 프로세스가 성공하면 다음 단계를 계속합니다.

구성 업데이트

Elastic Beanstalk가 시작되고, DB 인스턴스에 대한 정보를 제공하고, WordPress에 대한 인증 설정을 제공하는 EC2 인스턴스와 DB 인스턴스의 보안 그룹을 연결합니다.

필수 조건

<https://api.wordpress.org/secret-key/1.1/salt/>를 엽니다. 이 사이트는 wp-config.php 파일에서 인증 설정에 대해 지정한 환경 변수에 사용할 수 있는 설정을 생성합니다. Elastic Beanstalk 콘솔을 사용하여 이 절차를 완료할 때까지 이 설정을 잘 간직해 두십시오.

구성을 업데이트하려면

1. 탐색 창에서 Configuration을 선택합니다.
2. Instances에서 기어 모양 아이콘을 선택합니다.
3. EC2 security groups에서, 현재 텍스트의 끝으로 이동하고, 심표를 입력한 다음, DB 인스턴스의 보안 그룹 이름(예: rds-launch-wizard)을 입력합니다. Apply를 선택합니다. 경고를 읽은 다음, Save를 선택합니다.

환경 업데이트가 완료되면 다음 단계를 계속합니다.

4. 탐색 창에서 Configuration을 선택합니다.
5. Software Configuration에서 기어 모양 아이콘을 선택합니다.
6. Environment Properties에서, 다음 속성을 정의하여 Elastic Beanstalk가 DB 인스턴스에 액세스할 수 있도록 합니다.

- RDS_DB_NAME - 데이터베이스의 이름
- RDS_USERNAME - 마스터 사용자의 이름
- RDS_PASSWORD - 마스터 사용자의 암호
- RDS_HOSTNAME - DB 인스턴스의 엔드포인트(Amazon RDS 콘솔에서)
- RDS_PORT - DB 인스턴스의 포트(예: 3306)

다음 인증 설정을 정의합니다.

- AUTH_KEY - AUTH_KEY에 대해 생성된 값입니다.
 - SECURE_AUTH_KEY - SECURE_AUTH_KEY에 대해 생성된 값입니다.
 - LOGGED_IN_KEY - LOGGED_IN_KEY에 대해 생성된 값입니다.
 - NONCE_KEY - NONCE_KEY에 대해 생성된 값입니다.
 - AUTH_SALT - AUTH_SALT에 대해 생성된 값입니다.
 - SECURE_AUTH_SALT - SECURE_AUTH_SALT에 대해 생성된 값입니다.
 - LOGGED_IN_SALT - LOGGED_IN_SALT에 대해 생성된 값입니다.
 - NONCE_SALT - NONCE_SALT에 대해 생성된 값입니다.
7. Apply를 선택합니다. 환경 업데이트가 완료되면 다음 절차를 계속합니다.

WordPress 설치

애플리케이션 서버가 시작되면 WordPress를 설치할 수 있습니다.

WordPress를 설치하려면

1. 환경의 URL(예: mywordpressblog-env.us-west-2.elasticbeanstalk.com)을 선택합니다. 모든 것이 잘 작동하는 경우 브라우저에 WordPress 설치 페이지가 표시됩니다.
2. 언어를 선택한 후 Continue를 선택합니다.
3. 사이트 제목, 사용자 이름 및 이메일 주소를 입력합니다. 암호는 안전한 장소에 보관하도록 하고, Install WordPress를 선택합니다.

Welcome

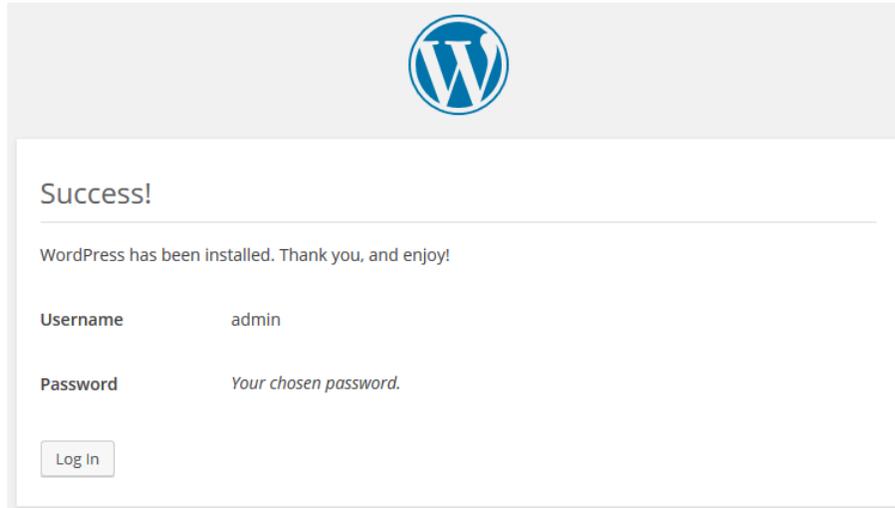
Welcome to the famous five-minute WordPress installation process! Just fill in the information below and you'll be on your way to using the most extendable and powerful personal publishing platform in the world.

Information needed

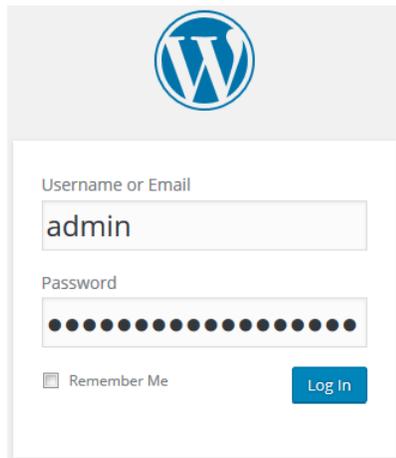
Please provide the following information. Don't worry, you can always change these settings later.

| | |
|--------------------------|---|
| Site Title | <input type="text" value="My WordPress Blog"/> |
| Username | <input type="text" value="admin"/> <small>Username can have only alphanumeric characters, spaces, underscores, hyphens, periods, and the @ symbol.</small> |
| Password | <input type="password" value="nteCz6UIOb2x5oQlp\$"/> <input type="button" value="Hide"/> Strong <small>Important: You will need this password to log in. Please store it in a secure location.</small> |
| Your Email | <input type="text" value="my@email.com"/> <small>Double-check your email address before continuing.</small> |
| Search Engine Visibility | <input checked="" type="checkbox"/> Discourage search engines from indexing this site <small>It is up to search engines to honor this request.</small> |

- 설치를 마치면 성공 메시지가 표시됩니다. Log In을 선택합니다.



- 사용자 이름 및 암호를 사용하여 로그인합니다.

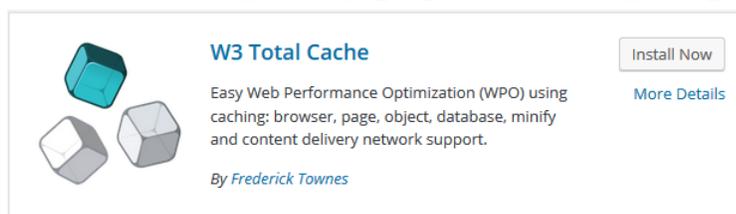


Amazon S3에 자산을 저장하도록 WordPress 설정

W3 Total Cache 플러그인을 설치하여 WordPress가 Amazon S3에 자산을 저장하도록 설정합니다.

W3 Total Cache 플러그인을 설치하려면

- 탐색 창의 WordPress 대시보드에서 Plugins를 선택합니다.
- Plugins 페이지에서 Add New를 선택합니다.
- W3 Total Cache라는 플러그인을 검색합니다. Install Now를 선택합니다.

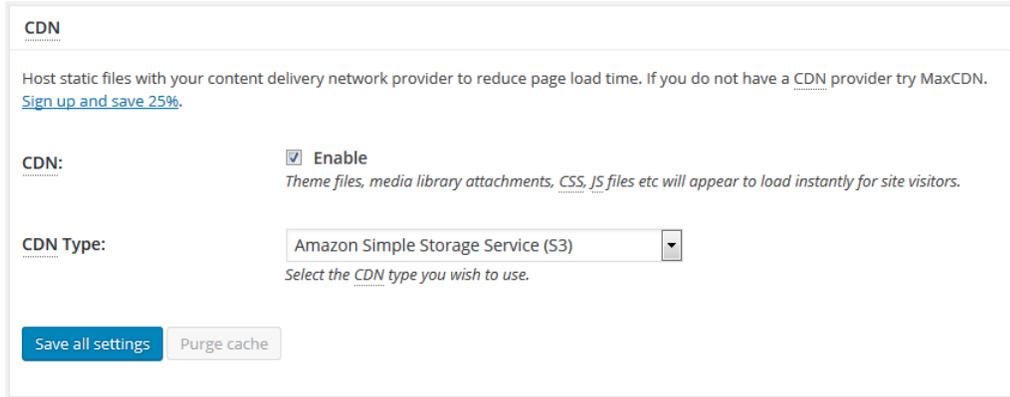


- 플러그인이 설치되면, Activate Plugin을 선택합니다.

5. Plugins 페이지의 W3 Total Cache 플러그인에서, Settings를 선택합니다.



6. CDN 창으로 스크롤합니다. CDN에서, Enable을 선택합니다. CDN type에서, Amazon Simple Storage Service (S3)를 선택합니다. Save all settings를 선택합니다.



7. 탐색 창에서 CDN을 선택합니다.
8. Configuration 창으로 스크롤하고 다음을 수행합니다.
- Access key ID에서, 생성한 IAM 사용자의 액세스 키 ID를 복사하여 붙여 넣습니다.
 - Secret key에서, 생성한 IAM 사용자의 비밀 키를 복사하여 붙여 넣습니다.
 - Bucket에서, 생성한 S3 버킷의 이름을 입력합니다.
 - Save all settings를 선택합니다.

Configuration

We recommend that you use [IAM](#) to create a new policy for AWS services that have limited permissions. A helpful tool: [AWS Policy Generator](#)

Access key ID: AKIAICIXT2IMLEXAMPLE

Secret key:

Bucket: assets-for-my-new-blog

SSL support: Auto (determine connection type automatically)
Some [CDN](#) providers may or may not support SSL, contact your vendor for more information.

Replace site's hostname with: <bucket>.s3.amazonaws.com or CNAME:
1.

If you have already added a [CNAME](#) to your [DNS](#) Zone, enter it here.

9. (선택 사항) 필요한 경우 다른 플러그인 설정을 변경할 수 있습니다. 자세한 내용은 [W3 Total Cache 설치 페이지](#)를 참조하십시오.

Amazon S3에 WordPress 자산 저장

W3 Total Cache 플러그인을 사용하여 Amazon S3에 WordPress 자산을 저장합니다.

S3 버킷에 WordPress 자산을 저장하려면

1. Plugins 또는 Dashboard 페이지에서 다음 메시지를 찾습니다.

Make sure to and upload the , , , files to the [CDN](#) to ensure proper operation.

2. wp-includes를 선택합니다. Includes files export 페이지에서, Start를 선택합니다. 권한이 제대로 설정된 경우, 파일이 Amazon S3로 업로드됩니다. 업로드가 완료되면, 창을 닫습니다.

Includes files export

This tool will upload files of the selected type to content delivery network provider.

Total files: 369
Processed: 0
Status: -
Time elapsed: -
Last response: -

3. (선택 사항) Amazon S3 콘솔을 열고 이 파일이 S3 버킷에 저장되었는지를 확인합니다.
4. theme files를 선택합니다. Theme files export 페이지에서, Start를 선택합니다.
5. custom files를 선택합니다. Custom files export 페이지에서, Start를 선택합니다.
6. (선택 사항) 모든 파일을 업로드한 후, Hide this message를 선택합니다.
7. (선택 사항) Dashboard 페이지의 Welcome to WordPress 창에서, View your site를 선택합니다. 브라우저를 사용하여 페이지 소스를 봅니다. 스타일시트의 URL 및 다른 자산이 S3 버킷을 가리키고 있음을 알 수 있습니다.
8. Appearance 페이지에서 테마를 변경하는 경우, W3 Total Cache에서 업데이트된 테마 파일을 업로드하라는 메시지를 표시합니다.

사용자 지정 도메인 이름 사용

도메인 이름을 가지고 있는 경우, 이를 Elastic Beanstalk가 제공한 도메인 이름과 연결하여 사용자가 도메인 이름을 사용하여 WordPress 사이트에 액세스하도록 할 수 있습니다. CNAME 레코드를 생성하여 쿼리를 WordPress 웹 사이트로 라우팅하려면 도메인 등록 대행자와 같은 DNS 서비스를 사용하면 됩니다. 자세한 내용은 DNS 서비스에 대한 설명서를 참조하십시오. 또는 DNS 서비스로 Amazon Route 53을 사용하고, 쿼리를 WordPress 웹 사이트로 라우팅하도록 이를 구성할 수 있습니다. WordPress 사이트에 대해 도메인 이름이 없는 경우 Amazon Route 53을 사용하는 도메인 이름을 구매할 수 있습니다.

자세한 내용은 [AWS Elastic Beanstalk 개발자 안내서](#)의 Elastic Beanstalk 환경의 도메인 이름 단원을 참조하십시오.

5단계: 애플리케이션 버전 업데이트

WordPress 구성 및 WordPress 플러그인 설치 과정에서, 애플리케이션 서버를 변경한 바 있습니다. Auto Scaling가 확장 이벤트에 대한 응답으로 새 애플리케이션 서버를 시작하거나 현재 애플리케이션 서버를 종료하고 이를 대체했어야 하는 경우, 이러한 변경 사항이 새 애플리케이션 서버에 적용되지 않습니다. 새로 시작한 애플리케이션 서버가 현재 애플리케이션 서버와 일치하도록 하려면 Elastic Beanstalk를 사용하여 새 애플리케이션 서버를 생성하십시오.

WordPress 업데이트를 설치하거나, WordPress 사이트의 구성을 변경하거나 또는 새 플러그인을 설치할 때마다 이러한 절차를 따라야 합니다.

사전 조건

WordPress 설치를 변경하기 전에 Auto Scaling 그룹에서 실행 중인 인스턴스가 단 1개인지 확인합니다.

Auto Scaling 그룹을 1개의 인스턴스로 제한하려면

1. <https://console.aws.amazon.com/elasticbeanstalk/>에서 Elastic Beanstalk 콘솔을 엽니다.
2. 환경을 선택합니다.
3. 탐색 창에서 Configuration을 선택합니다.
4. Scaling에서 기어 모양 아이콘을 선택합니다.
5. Minimum instance count 및 Maximum instance count에서, 1을 입력한 다음, Apply를 선택합니다.

Elastic Beanstalk를 사용하여 새 애플리케이션 버전 추가

각 애플리케이션 버전에는 버전 정보뿐만 아니라 고유한 ZIP 또는 WAR 파일이 있습니다.

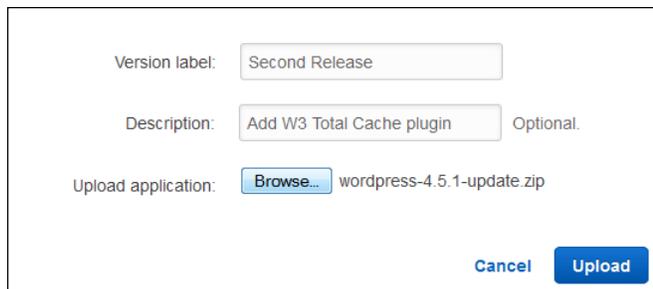
새 애플리케이션 버전을 추가하려면

1. WordPress 설치 업데이트를 마치면, SCP 또는 WinSCP를 사용하여 애플리케이션 서버의 웹 폴더(/var/app/current)를 로컬 컴퓨터에 다운로드합니다. 애플리케이션 서버의 퍼블릭 DNS 이름 및 키 페어의 .pem 파일의 경로를 알아야 합니다.
2. 다운로드한 파일에서 새 .zip 파일을 생성합니다.

3. <https://console.aws.amazon.com/elasticbeanstalk/>에서 Elastic Beanstalk 콘솔을 엽니다.
4. Actions, View Application Versions를 선택합니다.



5. Upload를 선택합니다.
6. Upload 페이지에서 다음을 수행합니다.
 - a. 버전 레이블을 입력합니다.
 - b. (선택 사항) 설명을 입력합니다.
 - c. Browse를 선택하고 새 애플리케이션 버전의 소스로 새 .zip 파일을 선택합니다.
 - d. Upload를 선택합니다.

A screenshot of the 'Upload application' form in the AWS Elastic Beanstalk console. It has three input fields: 'Version label' with the text 'Second Release', 'Description' with the text 'Add W3 Total Cache plugin' and 'Optional.' to its right, and 'Upload application' with a 'Browse...' button and the filename 'wordpress-4.5.1-update.zip'. At the bottom right, there are 'Cancel' and 'Upload' buttons.

7. Auto Scaling 그룹의 크기를 줄인 경우 이를 복원합니다.
 - a. 탐색 창에서 Configuration을 선택합니다.
 - b. Scaling에서 기어 모양 아이콘을 선택합니다.
 - c. Minimum instance count 및 Maximum instance count의 이전 값을 복원한 다음, Apply를 선택합니다.

6단계: 정리

이제 5단계를 완료했으므로 WordPress 웹 사이트가 배포되었고 프로덕션 작업에 사용할 준비가 되었습니다.

WordPress 웹 사이트 사용을 마쳤으면, 계정에 추가 요금이 발생하는 것을 막기 위해 이 자습서에서 생성한 AWS 리소스는 정리해야 합니다.

Elastic Beanstalk 리소스 삭제

환경을 종료할 때, Elastic Beanstalk는 자신이 생성한 리소스를 정리합니다. 예를 들어, 로드 밸런서 및 Auto Scaling 그룹을 삭제하고 EC2 인스턴스를 종료합니다.

Elastic Beanstalk 리소스를 삭제하려면

1. <https://console.aws.amazon.com/elasticbeanstalk/>에서 Elastic Beanstalk 콘솔을 엽니다.
2. 다음과 같이 애플리케이션 버전을 삭제합니다.
 - a. 애플리케이션 이름을 선택한 다음, Application Versions를 선택합니다.
 - b. 애플리케이션 버전을 선택한 다음, Delete를 선택합니다.
 - c. 확인 메시지가 나타나면 Delete를 선택합니다.
 - d. Done을 선택합니다.
3. 다음과 같이 환경을 종료합니다.
 - a. Environments를 선택한 다음, 환경을 선택합니다.
 - b. Actions, Terminate Environment를 선택합니다.
 - c. 확인 메시지가 나타나면 Terminate를 선택합니다.
4. 종료 작업을 마치면 Actions, Delete Application을 선택합니다.

Amazon RDS 데이터베이스 삭제

데이터베이스를 삭제하려면

1. <https://console.aws.amazon.com/rds/>에서 Amazon RDS 콘솔을 엽니다.
2. 탐색 창에서 Instances를 선택합니다.

3. DB 인스턴스를 선택합니다.
4. Instance Actions에 대해 Delete를 선택합니다.
5. 메시지가 표시되면, Create final Snapshot에 대해 No를 선택하고, 확인 확인란을 선택하고, Delete를 선택합니다.

Amazon S3 버킷 삭제

버킷을 삭제하려면

1. <https://console.aws.amazon.com/s3/>에서 Amazon S3 콘솔을 엽니다.
2. 버킷에 대한 컨텍스트(오른쪽 버튼 클릭) 메뉴를 열고 Delete Bucket을 선택합니다.
3. 메시지가 나타나면 버킷 이름을 입력하고 Delete를 선택합니다.