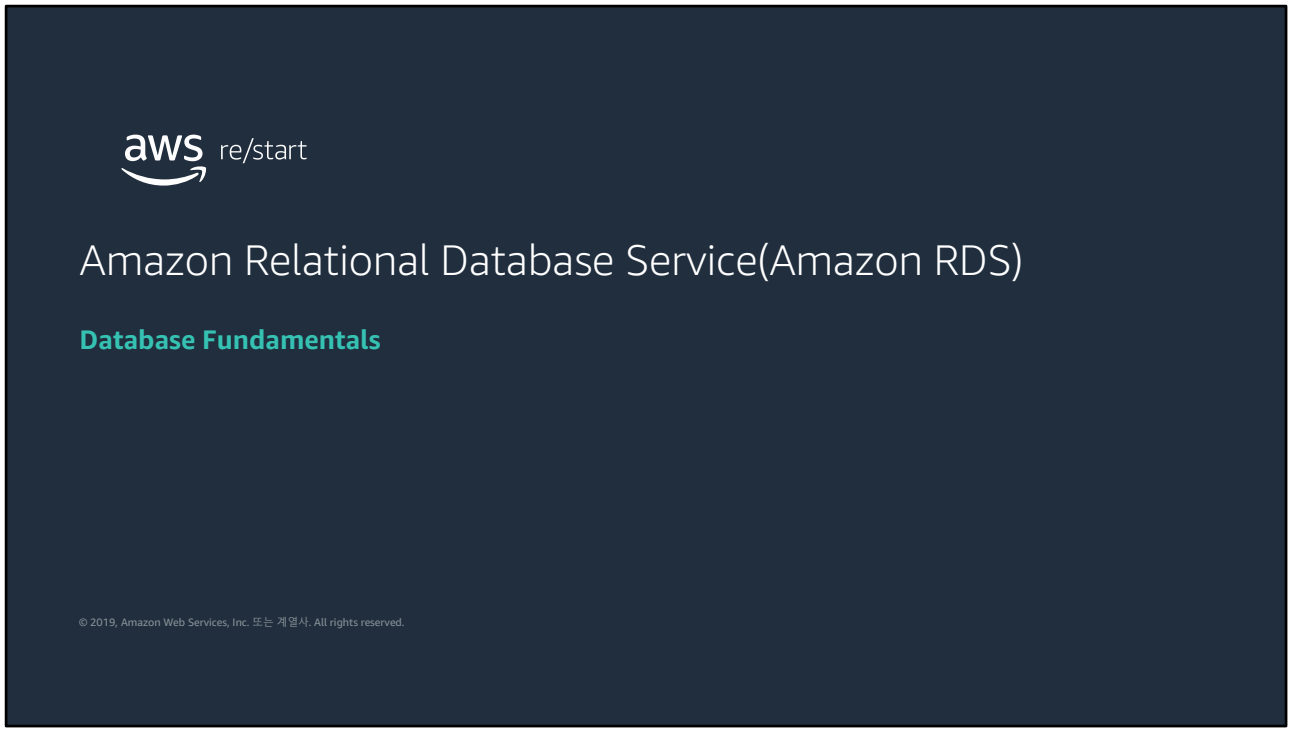


Amazon Relational Database Service(Amazon RDS)를 시작하겠습니다.

이 모듈에서는 데이터베이스 솔루션과 관련된 다음과 같은 주요 개념을 살펴볼 수 있습니다.

- 클라우드의 다양한 데이터베이스 서비스 이해
- 비관리형 데이터베이스 솔루션과 관리형 데이터베이스 솔루션의 차이점 확인

이 모듈의 목표는 솔루션을 강화하는 데 사용할 수 있는 데이터베이스 리소스를 이해하는 데 도움을 주기 위한 것입니다. 또한 다양한 선택 사항이 솔루션 가용성에 미치는 영향을 이해할 수 있도록 사용 가능한 다양한 서비스 기능을 검토합니다.

교육 내용

이 강의의 핵심

배울 내용은 다음과 같습니다.

- Amazon RDS에 대한 개요 제공
- Amazon RDS가 제공하는 일부 옵션 설명
- Amazon RDS의 백업 옵션 살펴보기
- Amazon RDS의 6개 데이터베이스 유형 설명
- Amazon RDS가 제공하는 고가용성 실현
- Amazon RDS의 몇 가지 용례 검토
- Amazon RDS용 AWS Command Line Interface(AWS CLI)의 몇 가지 명령 살펴보기

주요 용어:

- Amazon Relational Database Service(Amazon RDS)
- 데이터베이스 인스턴스

이 모듈에서 학습할 내용은 다음과 같습니다.

- Amazon RDS에 대한 개요 제공
- Amazon RDS가 제공하는 일부 옵션 설명
- Amazon RDS 백업 옵션 살펴보기
- Amazon RDS의 6개 데이터베이스 유형 설명
- Amazon RDS가 제공하는 고가용성 실현
- Amazon RDS의 몇 가지 용례 검토
- Amazon RDS용 AWS Command Line Interface(AWS CLI)의 몇 가지 명령 살펴보기

이 모듈의 목표는 솔루션을 강화하는 데 사용할 수 있는 데이터베이스 리소스를 이해하는 데 도움을 주기 위한 것입니다. 또한 다양한 선택 사항이 솔루션 가용성에 미치는 영향을 이해할 수 있도록 사용 가능한 다양한 서비스 기능을 검토합니다.

Amazon Relational Database Service

Amazon Relational Database Service(Amazon RDS)부터 시작하겠습니다.

Amazon RDS

Amazon RDS란?

- 클라우드에서 관계형 데이터베이스를 설정하고 운영하는 관리형 데이터베이스 서비스

aws re/start

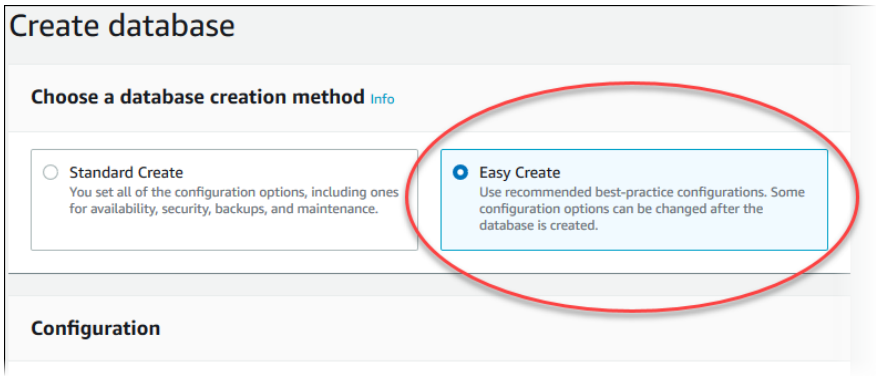
Amazon RDS는 클라우드에서 관계형 데이터베이스를 설정하고 운영하는 관리형 데이터베이스 서비스입니다.

비관리형, 독립 실행형 관계형 데이터베이스를 실행하는 것은 시간이 많이 소요되고 범위가 제한될 수 있습니다. 이러한 문제점을 해결하기 위해 AWS는 지속적인 관리가 필요 없이 관계형 데이터베이스를 설정, 운영 및 크기 조정하는 서비스를 제공합니다. Amazon RDS는 비용 효율적이고 크기 조정 가능한 용량을 제공하는 한편, 시간이 많이 소모되는 관리 태스크를 자동화합니다.

Amazon RDS를 사용하면 애플리케이션에 집중할 필요가 없어지므로 애플리케이션에 필요한 성능, 고가용성, 보안 및 호환성을 제공할 수 있습니다. Amazon RDS를 사용할 때 주된 초점은 데이터 및 애플리케이션 최적화입니다.

Amazon RDS에서 데이터베이스(DB) 인스턴스 생성

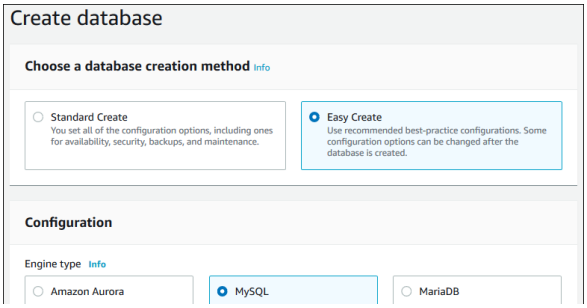
- DB 인스턴스를 생성하는 가장 쉬운 방법은 AWS Management Console을 사용하는 것입니다.
- DB 인스턴스를 만든 후에는 MySQL Workbench 같은 표준 MySQL 유틸리티를 사용하여 DB 인스턴스의 데이터베이스에 연결할 수 있습니다.



DB 인스턴스를 생성하는 방법

MySQL 예시

- **Configuration** 섹션에서 **MySQL**을 선택합니다.
- 다음으로, 요구 사항 질문에 답변하는 데 필요한 나머지 단계를 완료한 다음 **Create**를 선택합니다.
- 새 인스턴스를 사용하려면 DB 인스턴스 클래스와 스토리지의 양에 따라 최대 **20분**까지 걸릴 수 있습니다.
- AWS CLI를 사용하여 DB 인스턴스에 연결합니다.



aws re/start

Amazon RDS 백업

Amazon RDS의 백업 옵션은 다음과 같습니다.

자동	백업 기간 동안 DB 인스턴스의 자동화된 백업 생성(일일 전체 스냅샷 및 트랜잭션 로그)
수동	DB 인스턴스의 스토리지 볼륨 스냅샷 생성

aws re/start

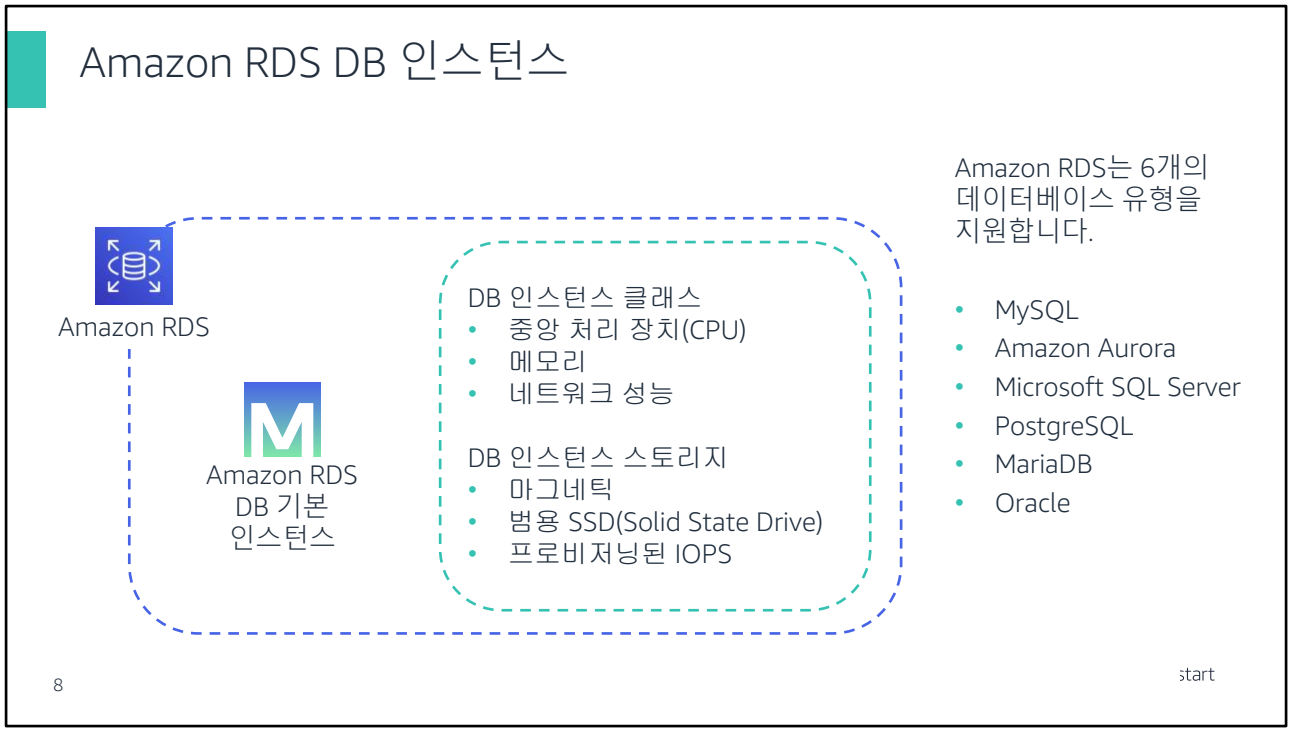
Amazon RDS는 완전관리형 서비스이기 때문에 Amazon RDS에서 자동으로 수행하는 하나의 태스크는 주기적인 DB 인스턴스 백업입니다. 전체 인스턴스는 특정 백업 기간 동안 스토리지 볼륨 스냅샷에 백업되며 지정된 백업 보존 기간에 따라 유지됩니다. DB 인스턴스의 첫 번째 스냅샷에는 전체 데이터가 포함됩니다. 후속 스냅샷은 증분형이며, 가장 최근 스냅샷 이후로 변경된 데이터만 포함됩니다.

선택 사항으로, 스냅샷을 생성하여 데이터베이스 인스턴스를 수동으로 백업할 수 있습니다.

백업 작업에 대한 자세한 내용은 Amazon RDS DB 인스턴스 백업 및 복원을 참조하십시오.

Amazon RDS 백업 및 스냅샷 관리를 위한 지원 문서 및 정보는 Amazon RDS DB 인스턴스 백업 및 복원을 참조하십시오.

또한 Amazon RDS FAQ를 참조하십시오.

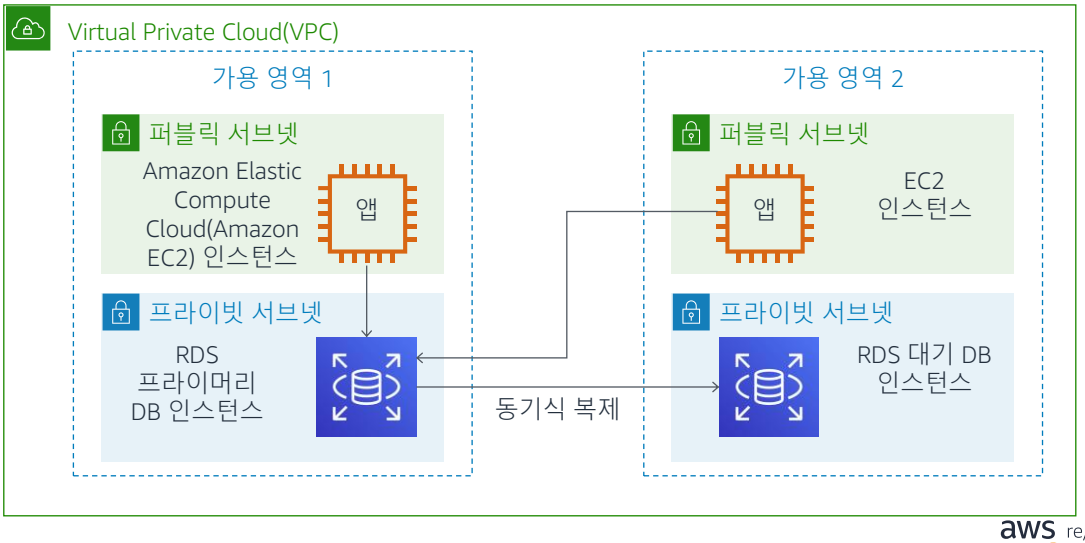


Amazon RDS의 기본 빌딩 블록은 DB 인스턴스입니다. **DB 인스턴스는 격리된 데이터베이스 환경입니다.** 사용자가 생성한 여러 데이터베이스가 포함될 수 있으며, 독립 실행형 데이터베이스 인스턴스에 사용하는 것과 동일한 도구 및 애플리케이션을 사용하여 액세스할 수 있습니다. 데이터베이스 인스턴스의 리소스는 데이터베이스 인스턴스 클래스에 따라 결정되며, 스토리지 유형은 디스크 유형에 따라 결정됩니다.

데이터베이스 인스턴스와 스토리지는 성능 특성과 가격이 다르므로 데이터베이스 요구에 따라 성능과 비용을 맞춤 설정할 수 있습니다. DB 인스턴스를 생성하려면 먼저 실행할 데이터베이스 엔진을 지정해야 합니다. Amazon RDS는 현재 MySQL, Amazon Aurora, Microsoft SQL Server, PostgreSQL, MariaDB, Oracle의 6개 데이터베이스를 지원합니다.

Amazon RDS의 고가용성

다중 AZ 배포를 통한 고가용성: 복제

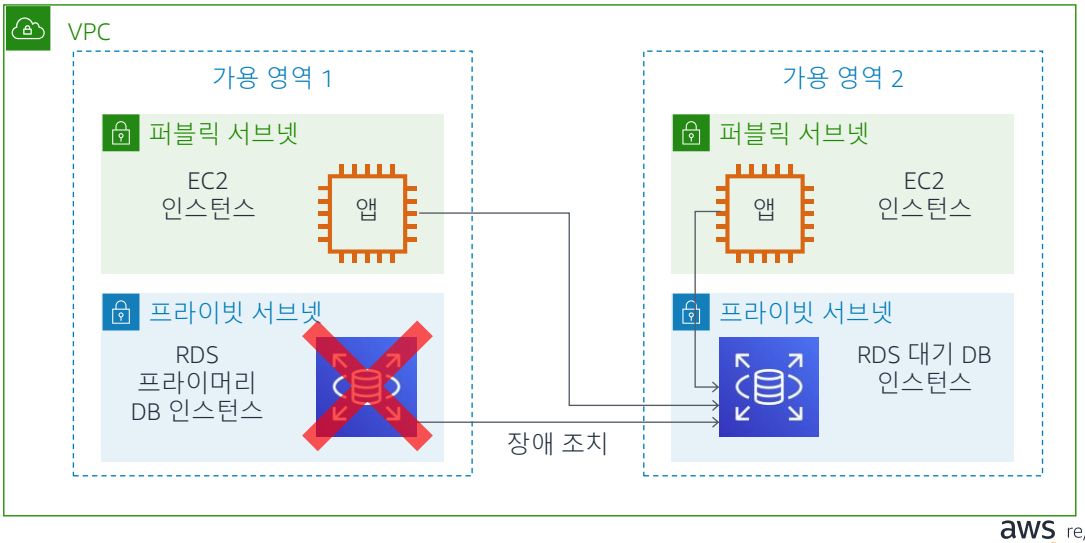


aws re/start

Amazon RDS의 주요 기능 중 하나는 다중 AZ 배포를 사용하여 고가용성 데이터베이스 인스턴스를 구성할 수 있다는 것입니다. 이 구성은 자동으로 동일한 Virtual Private Cloud(VPC)의 다른 가용 영역에 데이터베이스 인스턴스의 대기 복사본을 생성합니다. 초기 전체 복사본을 생성하면 트랜잭션은 대기 복사본에 동기식으로 복제됩니다. 가용 영역에서 여러 데이터베이스를 실행하면 계획된 시스템 유지 관리 중의 가용성을 향상할 수 있습니다. 따라서 데이터베이스를 가용 영역에서의 실패 및 중단으로부터 보호할 수 있습니다.

다이어그램에서 RDS DB 인스턴스는 고가용성으로 구성됩니다. 기본 인스턴스는 가용 영역 1에서 실행되며 가용 영역 1과 2에서 실행되는 애플리케이션의 요청을 수행합니다. 대기 복제본은 가용 영역 2에서 실행되며 데이터 이중화를 제공합니다. 또한 기본 인스턴스에서 동기적으로 복제됩니다.

다중 AZ 배포를 통한 고가용성: 장애 조치



aws re/start

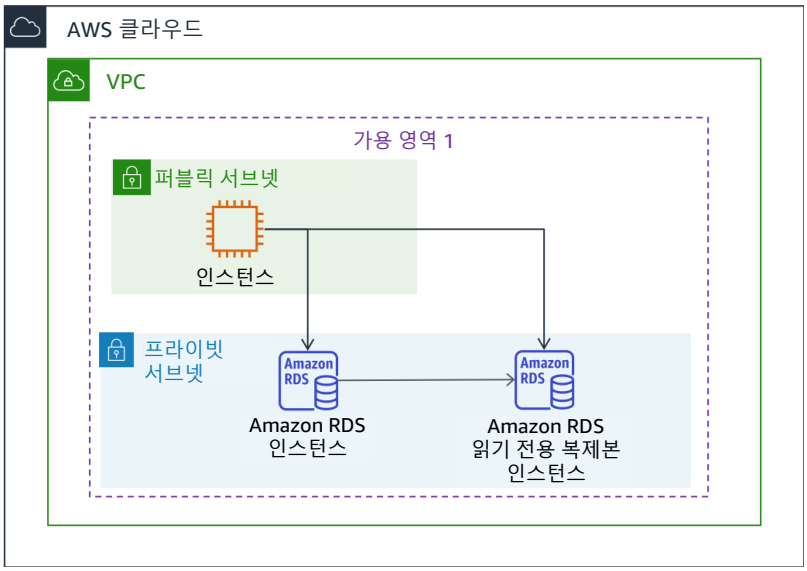
프라이머리 데이터베이스 인스턴스가 실패하면 Amazon RDS는 자동으로 대기 데이터베이스 인스턴스를 새로운 기본 인스턴스로 온라인에 가져옵니다. 그러면 두 애플리케이션의 요청이 새로운 기본 인스턴스로 리디렉트됩니다. 요청하는 애플리케이션은 Amazon RDS 도메인 이름 시스템(DNS) 엔드포인트를 사용하여 이름으로 데이터베이스를 참조합니다. 그 결과로 애플리케이션 코드를 변경할 필요 없이 장애 조치가 수행됩니다. 또한 동기식 복제로 인해 데이터 손실이 발생하지 않습니다.

Amazon RDS의 확장성

Amazon RDS 읽기 전용 복제본 및 크기 조정

특징
- 비동기식 복제
- 필요한 경우 기본 인스턴스를 승격할 수 있음

기능
- 읽기 중심의 데이터베이스 워크로드
- 읽기 쿼리 오프로드



aws re/start

고가용성 구성 외에 Amazon RDS는 확장성을 제공하는 또 다른 방법을 제공합니다. 예를 들어 MySQL, MariaDB, PostgreSQL, Amazon Aurora에 대해 **읽기 전용 복제본**을 생성할 수 있습니다. 소스 DB 인스턴스의 업데이트는 읽기 전용 복제본 인스턴스에도 비동기적으로 복사됩니다. 애플리케이션에서 읽기 전용 복제본으로 읽기 쿼리를 라우팅하여 소스 DB 인스턴스의 로드를 줄일 수 있습니다. 또한 읽기 중심의 데이터베이스 워크로드를 처리하기 위해 읽기 전용 복제본을 사용하여 단일 DB 인스턴스의 용량 제한을 넘어 스케일 아웃할 수도 있습니다. 읽기 전용 복제본은 프라이머리 DB 인스턴스로 승격할 수도 있습니다. 그러나 이 옵션은 비동기 복제를 사용하기 때문에 수동 액션이 필요합니다.

프라이머리 DB 인스턴스에서 다른 리전에 읽기 전용 복제본을 생성할 수 있습니다. 이 기능은 재해 복구(DR) 요구 사항을 충족하거나 사용자에게 더 가까운 지리적 영역에 있는 읽기 전용 복제본으로 읽기를 보내 대기 시간을 줄이는 데 도움이 될 수 있습니다.

참고: 인스턴스 클래스를 변경하려면 가동 중단이 필요합니다.

크기 조정에 대한 자세한 내용은 수직 및 수평으로 Amazon RDS 인스턴스 크기 조정을 참조하십시오.

스토리지 유형에 대한 자세한 내용은 Amazon RDS DB 인스턴스의 스토리지

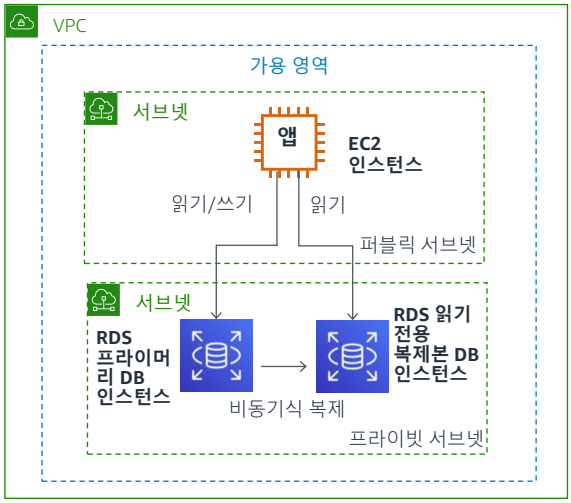
작업을 참조하십시오.

읽기 전용 복제본에 대한 자세한 내용은 Amazon RDS 읽기 전용 복제본을 참조하십시오.

Amazon RDS 크기 조정

Amazon RDS 크기 조정:

- 인스턴스 클래스
  - 인스턴스 클래스를 변경하여 컴퓨팅 및 메모리 용량 증대
- 스토리지 용량
  - 기존 인스턴스를 위한 스토리지 크기 조정
- 읽기 워크로드가 읽기 전용 복제본을 사용할 수 있음



aws re/start

인스턴스 클래스 또는 **스토리지 용량**을 변경하여 데이터베이스 서버의 용량을 늘릴 수 있습니다. 인스턴스 클래스를 변경하면 인스턴스에서 사용 가능한 CPU와 메모리를 늘릴 수 있습니다. 할당된 스토리지를 수정하면 가동 중단 없이 스토리지 용량을 늘릴 수 있습니다.

다이어그램은 읽기 전용 복제본을 생성하여 읽기 중심의 데이터베이스 워크로드의 성능을 높일 수 있음을 보여 줍니다. 또한 단일 데이터베이스 인스턴스의 용량 제한을 넘어 스케일 아웃할 수 있습니다.

참고: 인스턴스 클래스를 변경하려면 가동 중단이 필요합니다.

Amazon RDS 심층 분석

Amazon RDS 사용이 적합한 경우와 적합하지 않은 경우

애플리케이션에 다음이 필요한 경우 Amazon RDS를 사용합니다.

- 복잡한 트랜잭션 또는 복잡한 쿼리
- 높은 내구성

애플리케이션에 다음이 필요한 경우 Amazon RDS를 사용해서는 안 됩니다.

- 간단한 GET 또는 PUT 요청 및 NoSQL 데이터베이스가 처리할 수 있는 쿼리
- 관계형 데이터베이스 관리 시스템 (RDBMS) 사용자 지정

애플리케이션에 다음이 필요한 경우 Amazon RDS를 사용합니다.
- 복잡한 트랜잭션 또는 복잡한 쿼리
- 높은 내구성

애플리케이션에 다음이 필요한 경우 Amazon RDS를 사용해서는 안 됩니다.
- 간단한 GET 또는 PUT 요청 및 NoSQL 데이터베이스가 처리할 수 있는 쿼리
- 관계형 데이터베이스 관리 시스템(RDBMS) 사용자 지정

Amazon RDS를 사용하지 않아야 하는 경우 DynamoDB와 같은 NoSQL 데이터베이스 솔루션을 사용할 수 있습니다. Amazon RDS의 또 다른 대안은 Amazon EC2 인스턴스에서 관계형 데이터베이스 엔진을 실행하는 것입니다. 그러면 데이터베이스를 사용자 지정할 수 있는 옵션이 더 많이 제공됩니다.

또한 RDS에 대한 권장 용례는 다음과 같습니다.
- 자동 백업을 활성화하고, 데이터베이스에 최소한의 쓰기 작업이 있을 때 백업이 수행되도록 백업 기간을 설정합니다.
- 다중 AZ 배포를 사용하는 경우 DB 인스턴스의 장애 조치를 테스트하여 전환을 완료하는 데 시간이 얼마나 걸리는지 파악할 수 있습니다. 애플리케이션이 성공적으로 새 프라이머리 DB 인스턴스에 액세스할 수 있는지 확인합니다.

DB 인스턴스 클래스에 대한 자세한 내용은 DB 인스턴스 클래스를 참조하십시

오.

DB 스토리지 유형에 대한 자세한 내용은 Amazon RDS DB 인스턴스 스토리지를 참조하십시오.

Amazon RDS의 모범 실무에 관한 자세한 내용은 Amazon RDS의 모범 실무를 참조하십시오.

용례

웹 및 모바일 애플리케이션	전자 상거래 애플리케이션	모바일 및 온라인 게임
• 높은 처리량 • 대규모 스토리지 확장성 • 고가용성	• 저렴한 데이터베이스 • 데이터 보안 • 완전관리형 솔루션	• 신속한 용량 확장 • 오토스케일링 • 데이터베이스 모니터링

Amazon RDS는 높은 처리량, 광범위한 스토리지 확장성 및 고가용성을 갖춘 데이터베이스가 필요한 웹 및 모바일 애플리케이션에 적합합니다. Amazon RDS에는 라이선스 제한이 없으므로 이러한 애플리케이션의 가변적인 사용 패턴에 적합합니다. 크고 작은 규모의 전자 상거래 비즈니스의 경우 Amazon RDS는 온라인 판매 및 소매를 위한 유연하고 안전하며 비용이 저렴한 데이터베이스 솔루션을 제공합니다. 모바일 및 온라인 게임에는 높은 처리량과 가용성을 갖춘 데이터베이스 플랫폼이 필요합니다. Amazon RDS에서 데이터베이스 인프라를 관리하므로 게임 개발자는 데이터베이스 서버의 프로비저닝, 크기 조정 또는 모니터링에 대해 걱정할 필요가 없습니다.

AWS CLI의 Amazon RDS 명령

이 섹션에서는 다양한 Amazon RDS 명령을 살펴봅니다.

스냅샷 생성: 명령

다음 명령 사용:

```
$ aws rds
create-db-snapshot
--db-instance-identifier
mytestdb
--db-snapshot-identifier
mydbsnapshot
```

예상되는 결과:

```
{
 "DBSnapshot": {
   "DBSnapshotIdentifier": "mydbsnapshot",
   "DBInstanceIdentifier": "mytestdb",
   "Engine": "mysql",
   "AllocatedStorage": 20,
   "Status": "creating",
   "Port": 3306,
   "AvailabilityZone": "us-east-1f",
   "VpcId": "vpc-123456789012",
   "InstanceCreateTime": "2019-04-03T17:18:34.920Z",
   "MasterUsername": "admin",
   "EngineVersion": "5.6.40",
   "LicenseModel": "general-public-license",
   "SnapshotType": "manual",
   "OptionGroupName": "default:mysql-5-6",…
```


aws re/start

19

화면에 스냅샷을 생성하는 AWS CLI 명령이 나와 있습니다. 예상되는 결과:

```
{
   "DBSnapshot": {
      "DBSnapshotIdentifier": "mydbsnapshot",
      "DBInstanceIdentifier": "mytestdb",
      "Engine": "mysql",
      "AllocatedStorage": 20,
      "Status": "creating",
      "Port": 3306,
      "AvailabilityZone": "us-east-1f",
      "VpcId": "vpc-123456789012",
      "InstanceCreateTime": "2019-04-03T17:18:34.920Z",
      "MasterUsername": "admin",
      "EngineVersion": "5.6.40",
      "LicenseModel": "general-public-license",
      "SnapshotType": "manual",
      "OptionGroupName": "default:mysql-5-6",
      "PercentProgress": 0,
      "StorageType": "gp2",
      "Encrypted": false,
      "DBSnapshotArn": "arn:aws:rds:us-east-
1:123456789012:snapshot:mydbsnapshot",
      "IAMDatabaseAuthenticationEnabled": false
```

```
        }
    }
```

스냅샷에서 DB 복원: 명령

다음 명령 사용:

```
$ aws rds restore-db-instance-from-db-snapshot
--db-instance-identifier mytestdb-new
--db-snapshot-identifier mydbsnapshot
```

예상되는 결과는 다음과 유사한 값을 포함하는 JSON 객체입니다.

```
DBINSTANCE  mytestdb-new  db.t2.micro  MySQL     50       sa
creating  3  n  5.6.40  general-public-license
```


화면은 스냅샷에서 데이터베이스를 복원하는 명령을 보여 줍니다. 스냅샷은 새 데이터베이스 인스턴스로만 복원할 수 있습니다.

예상되는 결과는 다음과 유사한 값을 포함하는 JSON 객체입니다.

```
DBINSTANCE  mytestdb-new  db.t2.micro  MySQL     50       sa
creating  3  n  5.6.40  general-public-license
```

참고: DB 인스턴스를 복원한 후에 원래 DB 인스턴스에서 사용하는 보안 그룹에 추가해야 합니다. 이전 DB 인스턴스와 동일한 기능을 원한다면 이 단계는 필수입니다.

RDS 스냅샷 복사: 명령

다음 명령 사용:

```
$ aws rds copy-db-snapshot
--source-db-snapshot-identifier
mydbsnapshot
--target-db-snapshot-identifier
mydbsnapshot-copy --copy-tags
```

예상되는 결과:

```
{
 "DBSnapshot": {
   "VpcId": "vpc-0123456789012",
   "Status": "creating",
   "Encrypted": false,
   "SourceDBSnapshotIdentifier": "mydbsnapshot",
   "MasterUsername": "admin",
   "Iops": 1000,
   "Port": 3306,
   "LicenseModel": "general-public-license",
   "DBSnapshotArn": "arn:…mydbsnapshot-copy",
   "EngineVersion": "5.6.40",
   "OptionGroupName": "default:mysql-5-6", …
   "StorageType": "gp2",
   "IAMDatabaseAuthenticationEnabled": false,
   "SourceRegion": "us-east-1",
   "DBInstanceIdentifier": "mytestdb", …
```


21

화면은 스냅샷을 복사하는 명령을 보여 줍니다. 예상되는 결과:


```
{
    "DBSnapshot": {
       "VpcId": "vpc-0123456789012",
      "Status": "creating",
      "Encrypted": false,
      "SourceDBSnapshotIdentifier": "arn:aws:rds:us-east-
1:123456789012:snapshot:rds:mydbsnapshot",
      "MasterUsername": "admin",
      "Iops": 1000,
      "Port": 3306,
      "LicenseModel": "general-public-license",
      "DBSnapshotArn": "arn:aws:rds:us-east-
1:123456789012:snapshot:mydbsnapshot-copy",
      "EngineVersion": "5.6.40",
      "OptionGroupName": "default:mysql-5-6",
      "ProcessorFeatures": [],
      "Engine": "mysql",
      "StorageType": "gp2",
      "DbiResourceId": "db-ZI7UJ5BLKMBYFGX7FDENCKADC4",
      "KmsKeyId": "arn:aws:kms:us-east-
1:123456789012:key/AKIAIOSFODNN7EXAMPLE",
      "SnapshotType": "manual",
```

```
      "IAMDatabaseAuthenticationEnabled": false,
      "SourceRegion": "us-east-1",
      "DBInstanceIdentifier": ”mytestdb",
      "InstanceCreateTime": "2019-04-30T15:45:53.663Z",
      "AvailabilityZone": "us-east-1f",
      "PercentProgress": 0,
      "AllocatedStorage": 100,
      "DBSnapshotIdentifier": "mydbsnapshot-copy″
    }
}
```

스냅샷 삭제: 명령

다음 명령 사용:

```
$ aws rds delete-db-snapshot
--db-snapshot-identifier
mydbsnapshot
```

예상되는 결과:


```
{
 "DBSnapshot": {
    "DBSnapshotIdentifier": "mydbsnapshot",
    "DBInstanceIdentifier": "mytestdb",
    "SnapshotCreateTime": "2019-04-03T21:29:21.338Z",
    "Engine": "mysql",
    "AllocatedStorage": 20,
    "Status": "deleted",
    "Port": 3306,
    "AvailabilityZone": "us-east-1f",
    "VpcId": "vpc-012345678912",
    "InstanceCreateTime": "2019-04-03T17:18:34.920Z",
    "MasterUsername": "admin",
    "EngineVersion": "5.6.40",
    "LicenseModel": "general-public-license",
    "SnapshotType": "manual", ….
```


aws re/start

22

화면은 스냅샷을 생성하는 명령을 보여 줍니다. 예상되는 결과:

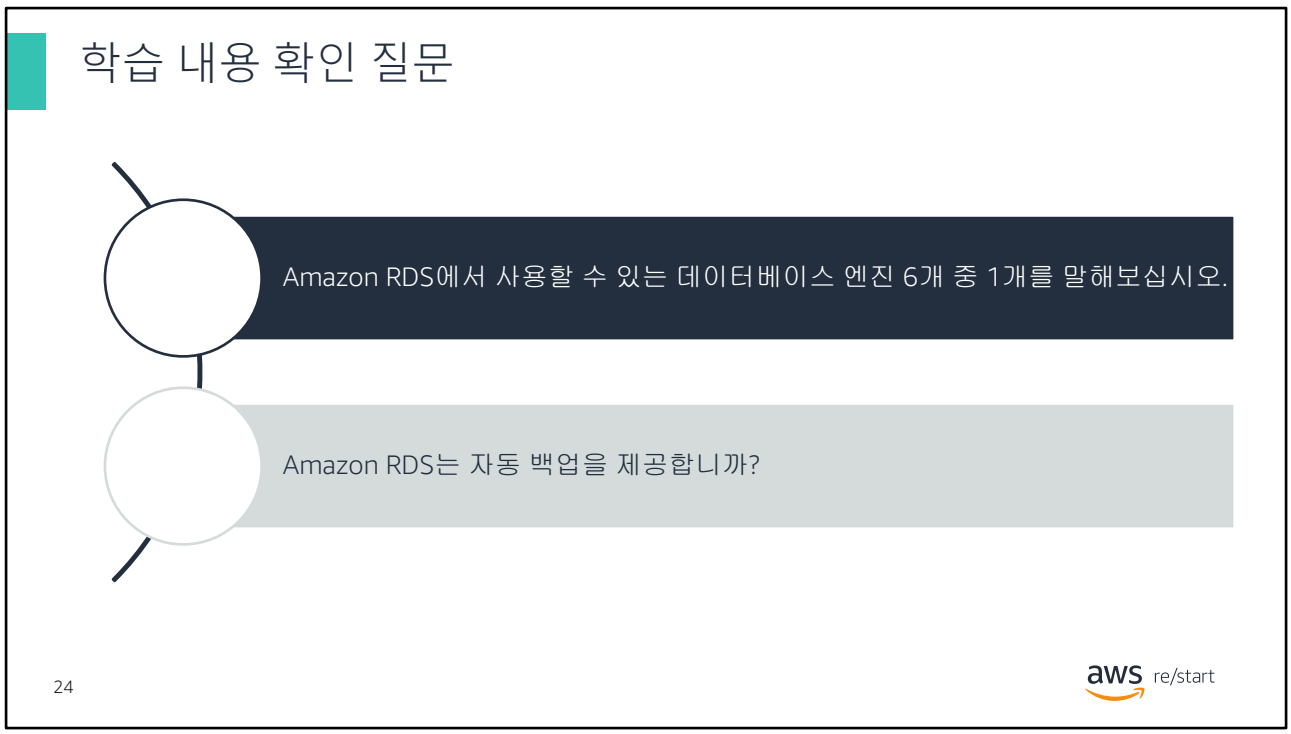

```
{
    "DBSnapshot": {
        "DBSnapshotIdentifier": "mydbsnapshot",
        "DBInstanceIdentifier": "mytestdb",
        "SnapshotCreateTime": "2019-04-03T21:29:21.338Z",
        "Engine": "mysql",
        "AllocatedStorage": 20,
        "Status": "deleted",
        "Port": 3306,
        "AvailabilityZone": "us-east-1f",
        "VpcId": "vpc-012345678912",
        "InstanceCreateTime": "2019-04-03T17:18:34.920Z",
        "MasterUsername": "admin",
        "EngineVersion": "5.6.40",
        "LicenseModel": "general-public-license",
        "SnapshotType": "manual",
        "OptionGroupName": "default:mysql-5-6",
        "PercentProgress": 100,
        "StorageType": "gp2",
        "Encrypted": false,
        "DBSnapshotArn": "arn:aws:rds:us-east-
1:012345678912:snapshot:mydbsnapshot",
```

```
        "IAMDatabaseAuthenticationEnabled": false
    }
}
```

Amazon RDS 데모

Amazon RDS 데모: Amazon Relational Database Service(RDS) 콘솔 데모를 검토하십시오.

이 동영상 데모는 학습 관리 시스템에서 찾을 수 있습니다.



1. Amazon RDS는 6개의 데이터베이스 유형을 지원합니다.
    - MySQL
    - Amazon Aurora
    - Microsoft Sequel Server
    - PostgreSQL
    - MariaDB
    - Oracle Database

2. 예, Amazon RDS는 자동 백업을 제공합니다. Amazon RDS는 자동으로 DB 인스턴스를 스냅샷으로 주기적으로 백업하는 완전관리형 서비스입니다.

핵심 사항

- Amazon RDS는 클라우드에서 관계형 데이터베이스를 손쉽게 설치, 운영 및 크기 조정할 수 있도록 하는 데이터베이스 서비스입니다.
- Amazon RDS는 자동화된 이중화 및 백업을 위한 기능을 제공합니다.
- Amazon RDS는 Amazon Aurora, PostgreSQL, MySQL, MariaDB, Oracle Database, Microsoft SQL Server를 비롯한 일련의 데이터베이스 엔진을 지원합니다.

aws re/start

이 모듈의 핵심 사항:

- Amazon RDS는 클라우드에서 관계형 데이터베이스를 손쉽게 설치, 운영 및 크기 조정할 수 있도록 하는 데이터베이스 서비스입니다.
- Amazon RDS는 관리형 서비스로, 콘솔, AWS CLI 또는 애플리케이션 프로그램 인터페이스(API) 호출을 사용하여 액세스할 수 있습니다.
- Amazon RDS는 자동화된 이중화 및 백업을 위한 기능을 제공합니다.
- Amazon RDS는 Amazon Aurora, PostgreSQL, MySQL, MariaDB, Oracle Database, Microsoft SQL Server를 비롯한 일련의 데이터베이스 엔진을 지원합니다.