



# ELB 로드 밸런서와 리스너

# 학습 내용

## 강의의 핵심

배울 내용은 다음과 같습니다.

- Elastic Load Balancing 기능 설명하기
- 로드 밸런서를 사용하여 아키텍처의 구성 요소 식별하기
- 리스너 및 대상 그룹의 목적 이해하기
- 로드 밸런서를 생성 및 구성하는 데 사용되는 AWS Command Line Interface(AWS CLI) 명령 검토하기

주요 용어:

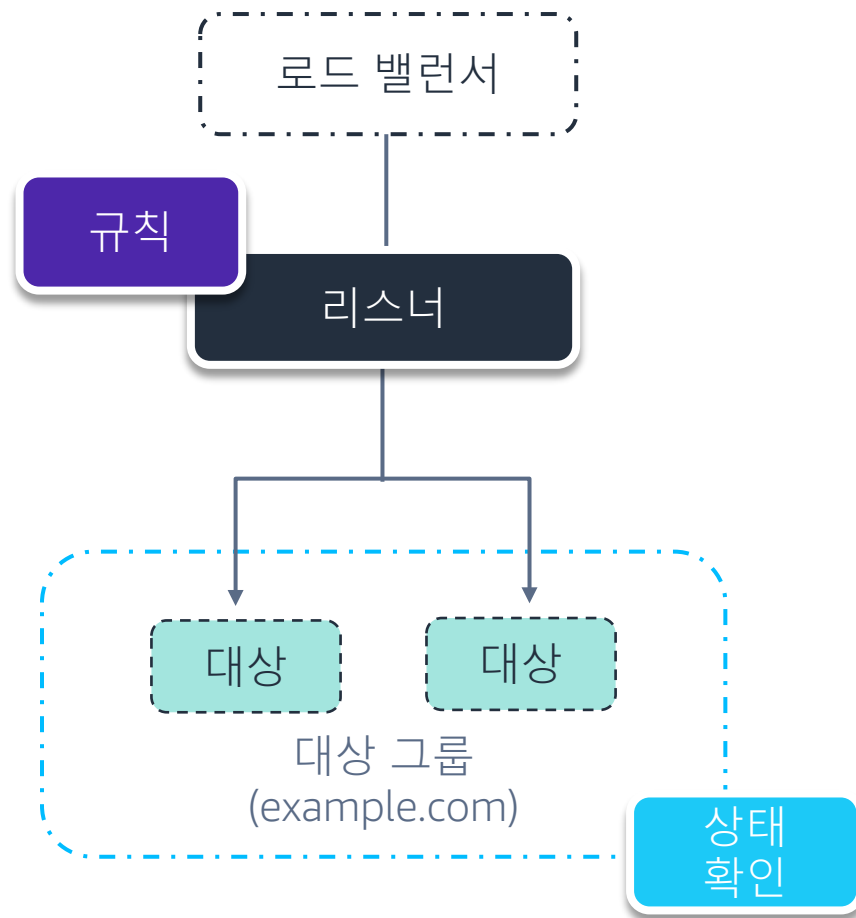
- 로드 밸런서
- 리스너
- 대상 그룹



# ELB 로드 밸런서와 리스너

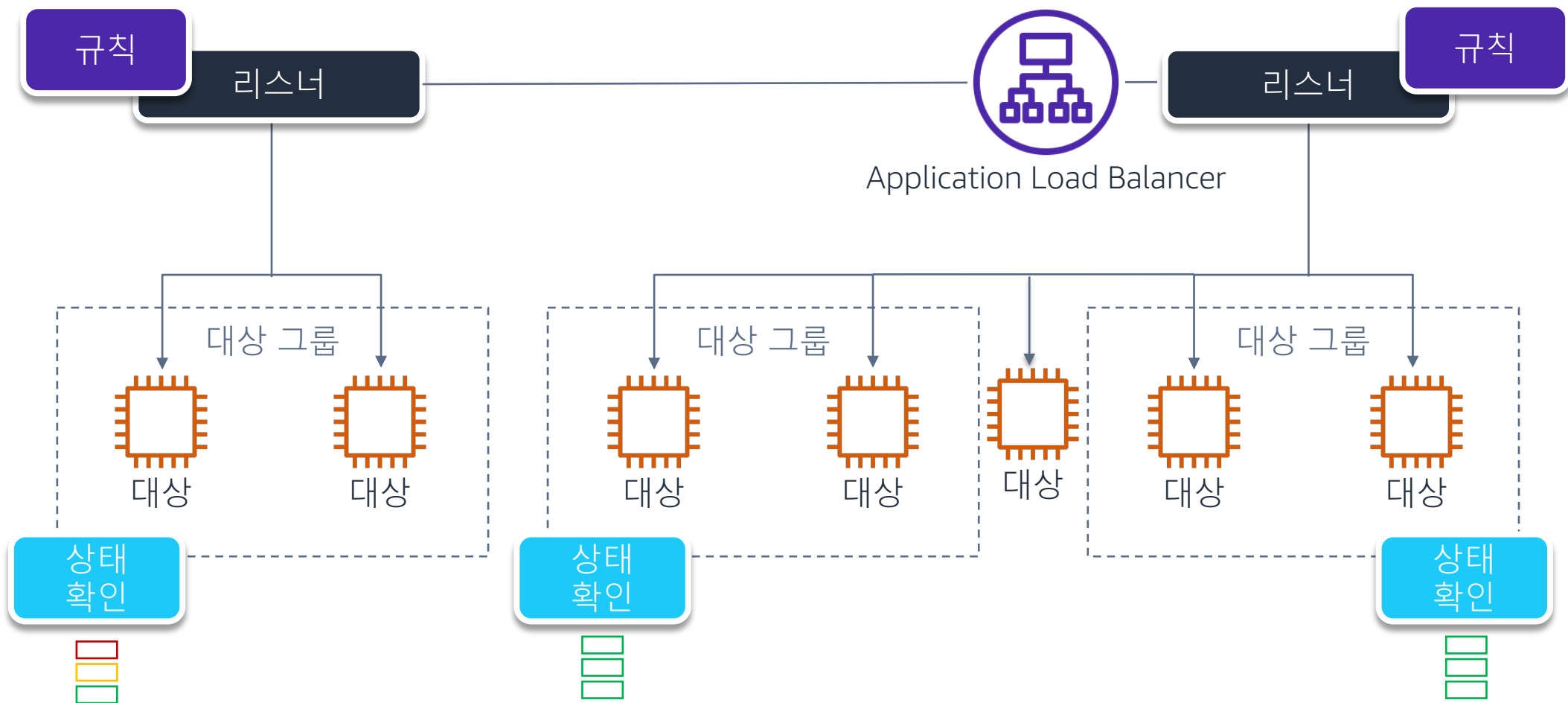
# 로드 밸런서: 리스너

- 로드 밸런서가 수신 대기하는 포트 및 프로토콜을 정의하는 프로세스입니다.
- 각 로드 밸런서는 트래픽을 수락하기 위해 하나 이상의 리스너가 필요합니다.
- 최대 50개의 리스너가 로드 밸런서에 있을 수 있습니다.
- 라우팅 규칙은 리스너에 정의됩니다.



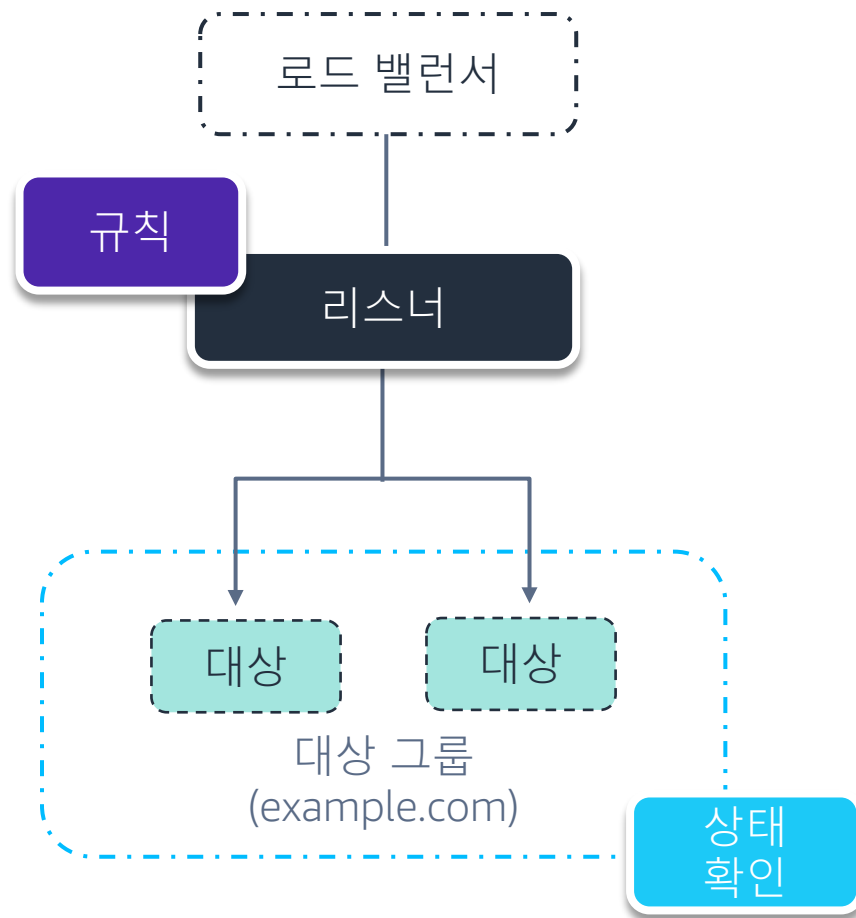
# Application Load Balancer

## 구성 요소

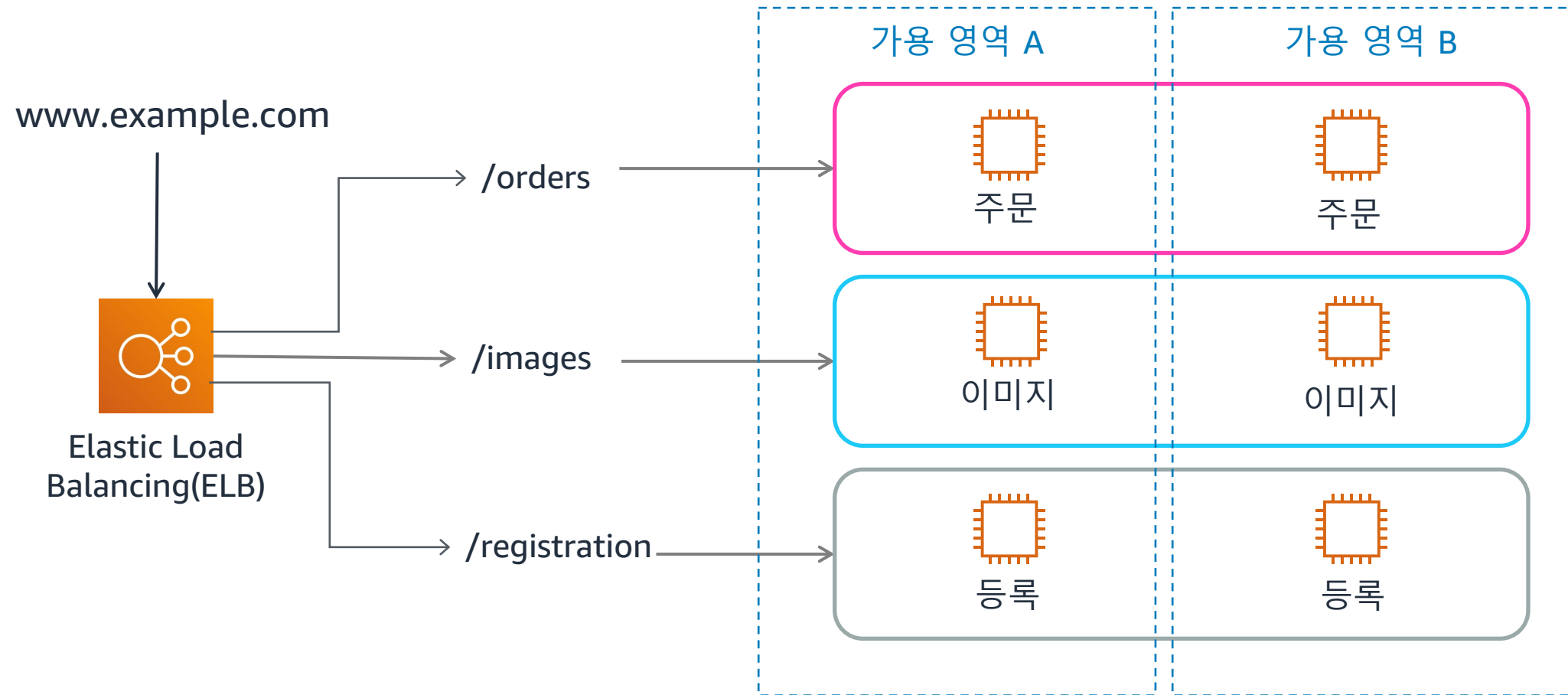


# 로드 밸런서: 대상 그룹

- 지원을 제공하는 등록된 대상 그룹
  - Amazon Elastic Compute Cloud(Amazon EC2) 인스턴스
  - Amazon Elastic Container Service(Amazon ECS) 컨테이너 인스턴스
- 단일 대상에는 여러 대상 그룹 등록이 있을 수 있습니다.



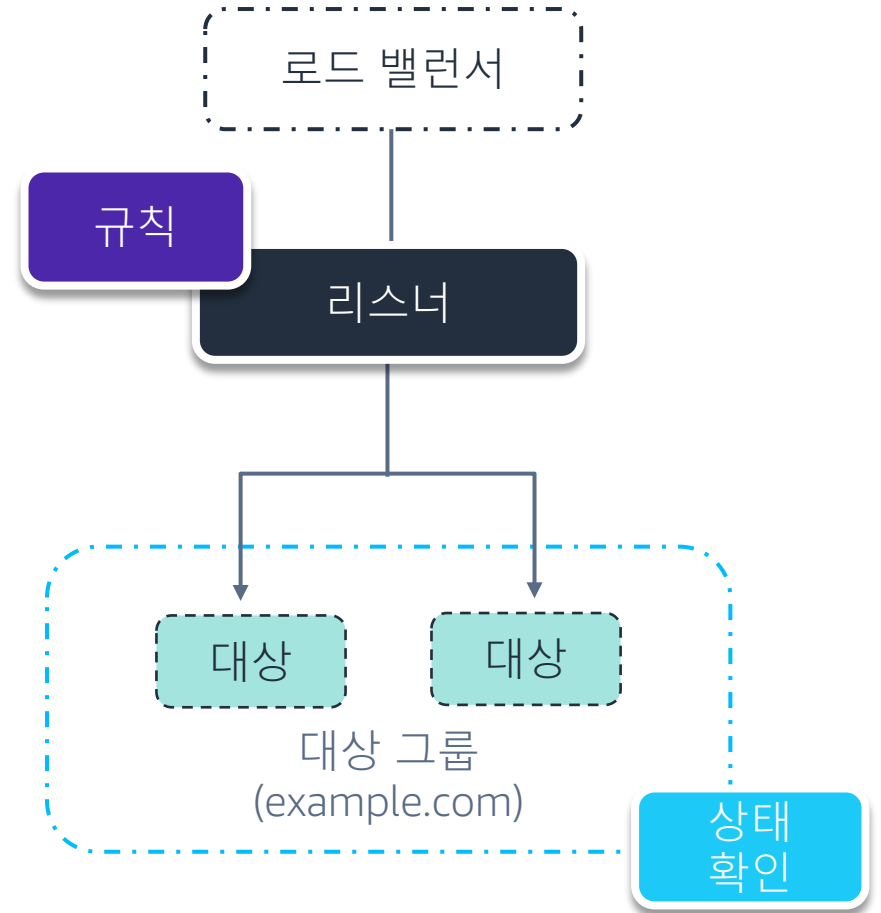
# Application Load Balancer 예제



# Application Load Balancer 생성: AWS Command Line Interface(AWS CLI)

1. **create-load-balancer** 명령을 사용하여 로드 밸런서를 생성합니다. 동일한 가용 영역의 서브넷이 아닌 2개의 서브넷을 지정해야 합니다.

```
aws elbv2 create-load-balancer \  
--name my-load-balancer \  
--subnets subnet-12345678 subnet-23456789 \  
--security-groups sg-12345678
```





# Application Load Balancer 생성: 명령 결과

## 명령

```
aws elbv2 create-load-balancer \  
--name my-load-balancer \  
--subnets subnet-12345678 subnet-23456789 \  
--security-groups sg-12345678
```

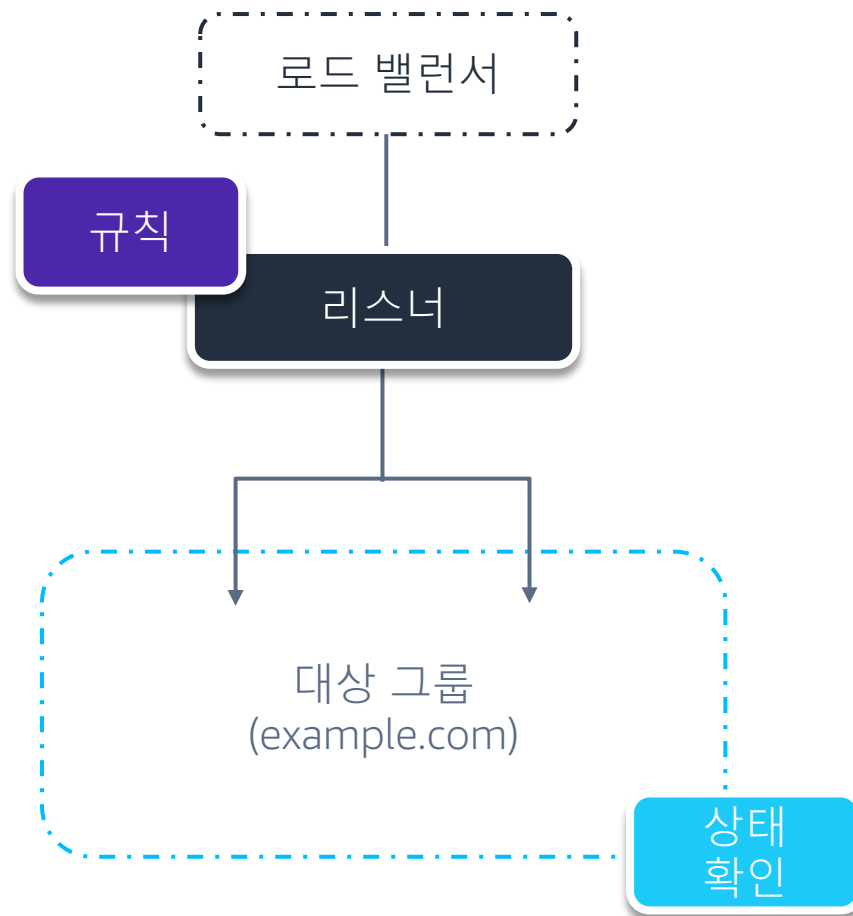
## 결과

```
{  
  "LoadBalancers": [{  
    "LoadBalancerArn": "arn:aws:elasticloadbalancing:us-east-  
1:123456789012:loadbalancer/app/my-load-balancer/1234567890123456",  
    "DNSName": "my-load-balancer-1234567890123456.us-east-1.elb.amazonaws.com",  
    "CanonicalHostedZoneId": "Z35SXDOTRQ7X7K",  
    "CreatedTime": "2019-03-28T16:33:59.670Z",  
    "LoadBalancerName": "my-load-balancer", ...  
  }]  
}
```

# 로드 밸런서의 대상 그룹 생성

2. **create-target-group** 명령을 사용하여 대상 그룹을 생성합니다. EC2 인스턴스가 실행 중인 VPC를 지정해야 합니다.

```
aws elbv2 create-target-group \  
--name my-targets \  
--protocol HTTP \  
--port 80 \  
--vpc-id vpc-12345678
```



# 로드 밸런서의 대상 그룹 생성: 명령 결과

## 명령

```
aws elbv2 create-target-group \  
--name my-targets \  
--protocol HTTP \  
--port 80 \  
--vpc-id vpc-12345678
```

## 결과

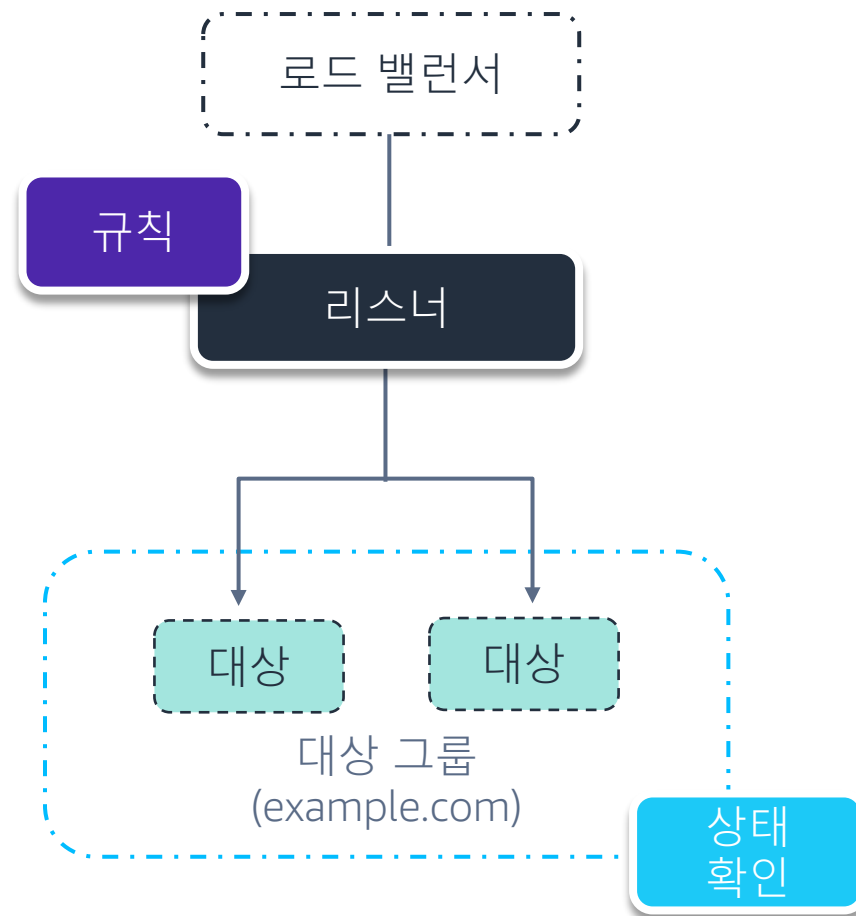
```
{  
  "TargetGroups": [{  
    "TargetGroupArn":  
    "arn:aws:elasticloadbalancing:us-east-  
1:123456789012:targetgroup/my-targets/9546a5cf349b582c",  
    "TargetGroupName": "my-targets",  
    "Protocol": "HTTP",  
    "Port": 80,  
    "VpcId": "vpc-1234567890123",  
    "HealthCheckProtocol": "HTTP",  
    "HealthCheckPort": "traffic-port",  
    "HealthCheckIntervalSeconds": 30,  
    "HealthCheckTimeoutSeconds": 5,  
    "HealthyThresholdCount": 5,  
    "UnhealthyThresholdCount": 2,  
    "HealthCheckPath": "/",  
    "Matcher": {  
      "HttpCode": "200"....
```

# 대상 그룹에 EC2 인스턴스 등록

3. **register-targets** 명령을 사용하여 인스턴스를 대상 그룹에 등록합니다.

```
aws elbv2 register-targets \  
--target-group-arn targetgroup-arn \  
--targets Id=i-12345678 Id=i-23456789
```

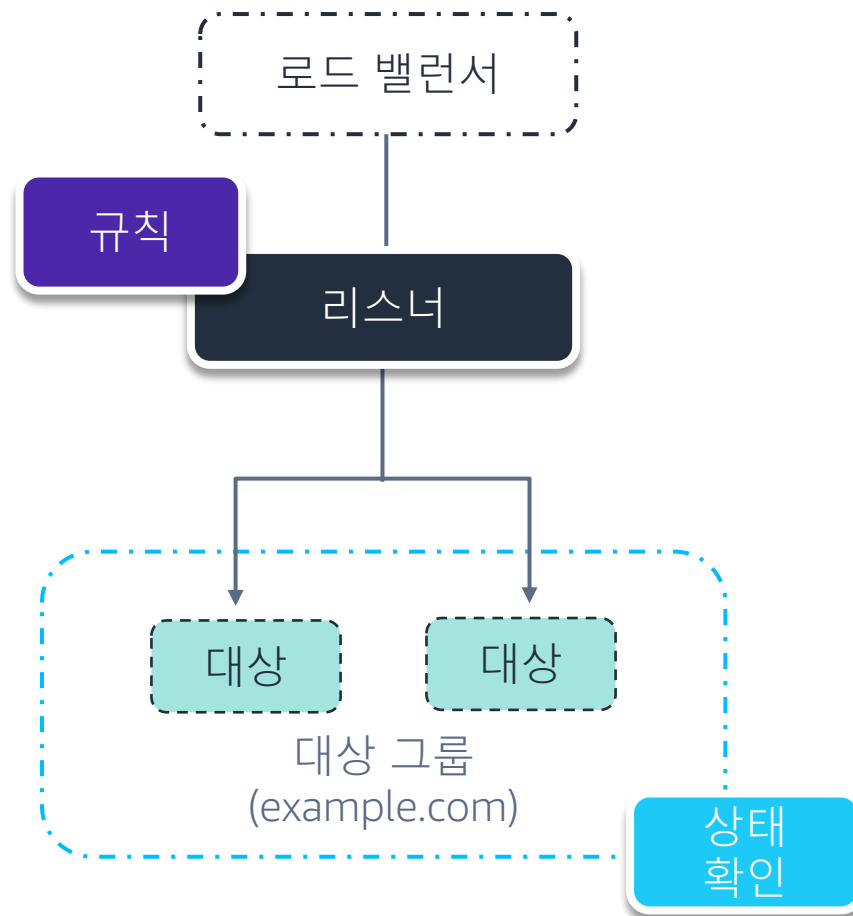
이 명령은 결과가 없습니다.



# 로드 밸런서에 대한 리스너 생성

4. **create-listener** 명령을 사용하여 로드 밸런서에 대한 리스너를 생성합니다.

```
aws elbv2 create-listener \  
--load-balancer-arn loadbalancer-arn \  
--protocol HTTP \  
--port 80 \  
--default-actions  
Type=forward,TargetGroupArn=targetgroup-arn
```



# 로드 밸런서에 대한 리스너 생성: 명령 결과

## 명령

```
aws elbv2 create-listener \  
--load-balancer-arn loadbalancer-arn \  
--protocol HTTP \  
--port 80 \  
--default-actions  
Type=forward,TargetGroupArn=targetgroup-arn
```

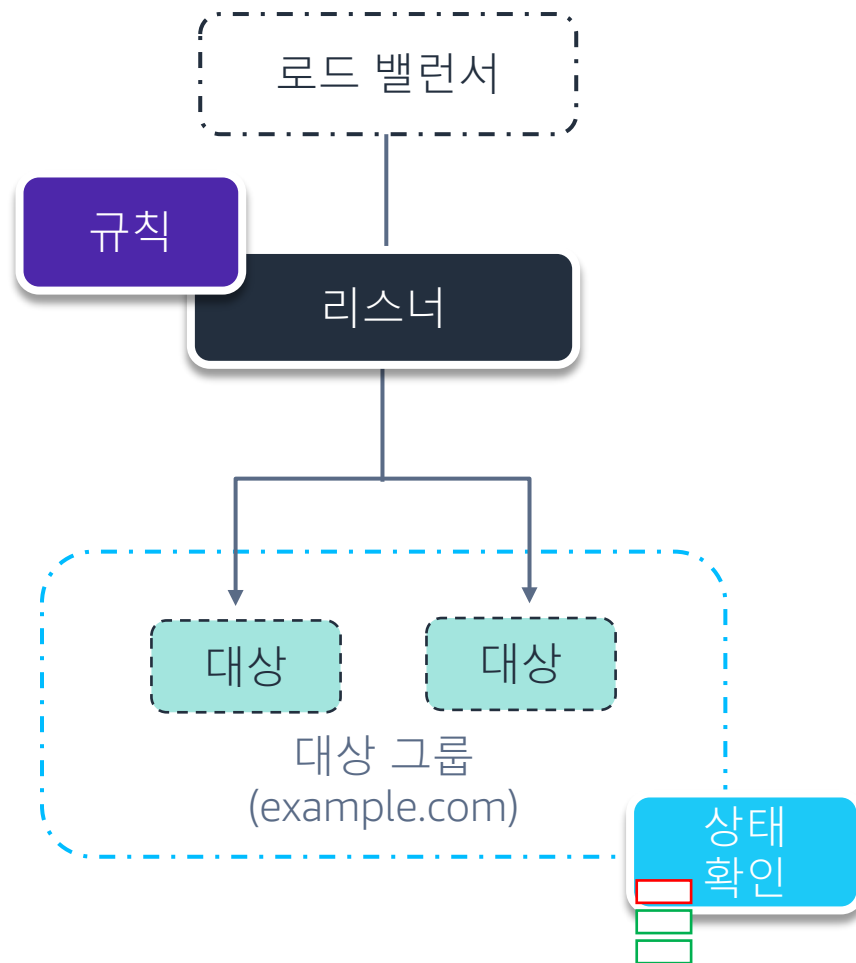
## 결과

```
{  
  "Listeners": [  
    {  
      "ListenerArn":  
        "arn:aws:elasticloadbalancing:us-east-1:123456789012:listener/app/my-load-balancer/1234567890123456/14f6f490e5f8bdbf",  
      "LoadBalancerArn":  
        "arn:aws:elasticloadbalancing:us-east-1:123456789012:loadbalancer/app/my-load-balancer/1234567890123456",  
      "Port": 80,  
      "Protocol": "HTTP",  
      "DefaultActions": [  
        {  
          "Type": "forward",  
          "TargetGroupArn":  
            "arn:aws:elasticloadbalancing:us-east-1:123456789012:targetgroup/my-targets/9546a5cf349b582c"  
        }  
      ]  
    }  
  ]  
}
```

# 등록된 대상의 상태 확인

5. 필요에 따라 **describe-target-health 명령**을 사용하여 대상 그룹에 대해 등록된 대상의 상태를 확인합니다.

```
aws elbv2 describe-target-health --target-group-arn targetgroup-arn
```



# 등록된 대상의 상태 확인: 명령 결과

## 명령

```
aws elbv2 describe-target-health --target-group-arn targetgroup-arn
```

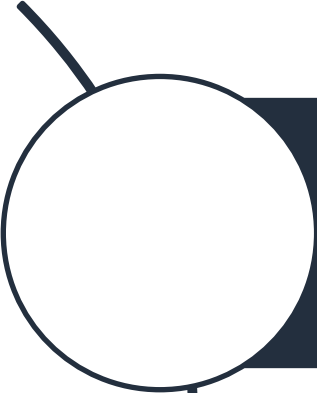
## 결과

```
{
  "TargetHealthDescriptions": [
    {
      "HealthCheckPort": "80",
      "Target": {
        "Id": "i-12345678",
        "Port": 80
      },
      "TargetHealth": {
        "State": "healthy"
      }
    },
    {
      "HealthCheckPort": "80",
      "Target": {
        "Id": "i-23456789",
        "Port": 80
      },
      "TargetHealth": {
        "State": "healthy"
      }
    }
  ]
}
```



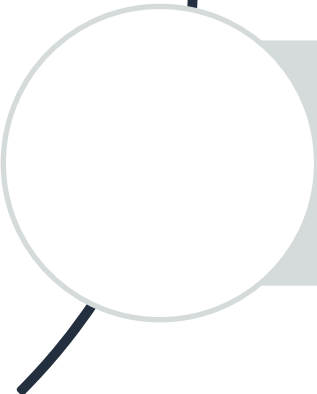


# 학습 내용 확인



웹사이트가 매우 느리게 실행되고 있습니다. 여기에는 example.com/login, example.com/products, example.com/orders 등 많은 하위 디렉터리가 있습니다. (참고: 웹사이트는 현재 EC2 인스턴스에서 호스팅됩니다.)

로드 밸런서를 사용하여 웹 사이트의 성능을 향상시키려면 어떻게 해야 합니까?



상태 확인이 로드 밸런서 또는 대상 서버에 적용됩니까?

# 핵심 사항



- 각 리스너는 클라이언트의 연결 요청을 확인한 다음, 정의된 규칙에 따라 요청을 하나 이상의 대상 그룹으로 전달합니다.
- 각 규칙은 대상 그룹, 조건 및 우선순위를 지정합니다.