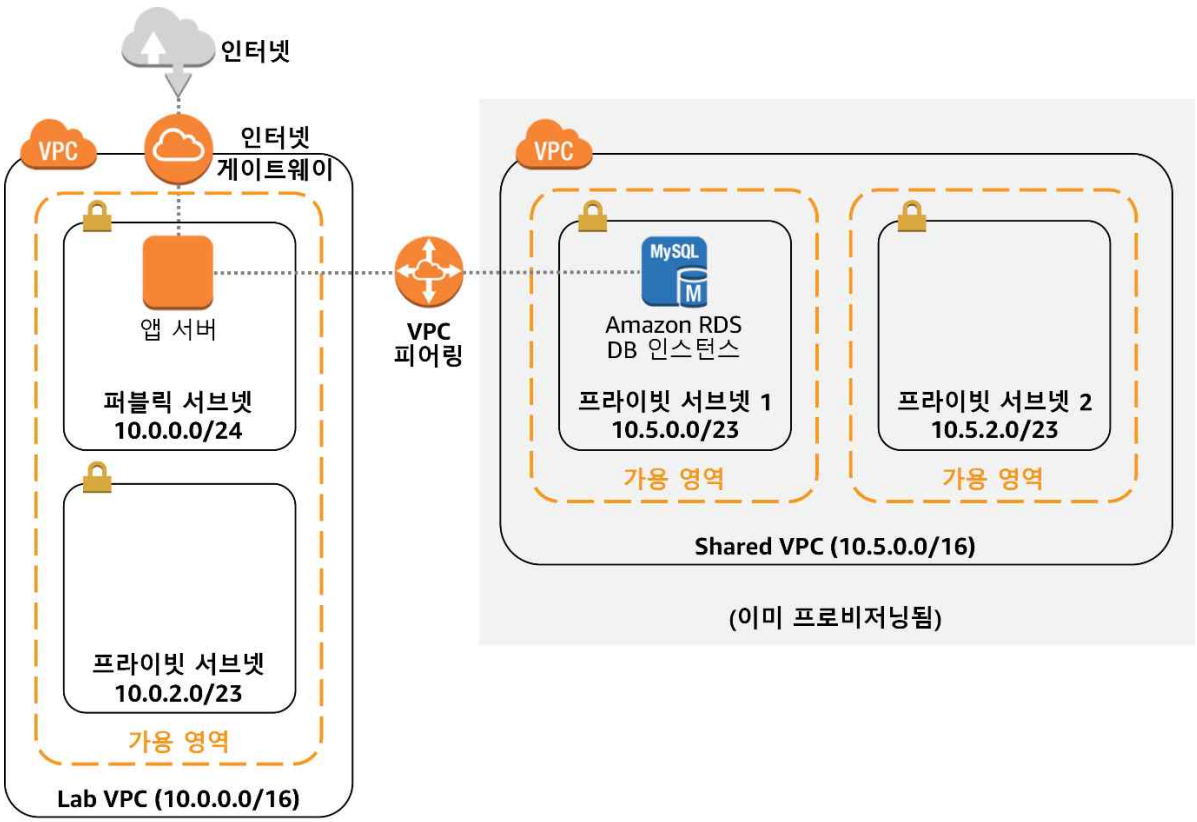
Architecting on AWS – 실습 3
– 가상 사설 클라우드 생성

실습 개요

일반적인 네트워킹은 어렵습니다. 장비, 케이블 배선, 복잡한 구성, 전문 기술이 필요하기 때문입니다. 다행히 Amazon Virtual Private Cloud(Amazon VPC)는 복잡한 요소를 거치지 않고 보안 프라이빗 네트워크를 간편하게 배포할 수 있습니다.

이 실습에서는 자체 VPC 를 구축하고, 서브넷을 생성하고, VPC 구성 요소 간에 트래픽을 보내는 방법을 보여줍니다. 다음 이미지는 최종 아키텍처를 보여줍니다.



선택 사항 **챌린지** 작업이 제공됩니다. 챌린지 작업에서는 공유 서비스 VPC 에 대한 VPC 피어링 연결을 생성합니다. 그런 다음 애플리케이션 및 데이터베이스를 사용하여 VPC 간 연결을 테스트합니다.

목표

이 실습을 완료하면 다음을 할 수 있게 됩니다.

- VPC 생성
- 퍼블릭 및 프라이빗 서브넷 생성
- 인터넷 게이트웨이 생성
- 라우팅 테이블 구성 및 서브넷에 연결

소요 시간

이 실습을 완료하는 데는 약 **40 분**이 소요됩니다.

실습 시작

1. 이 링크를 마우스 오른쪽 버튼으로 클릭한 다음 자신의 컴퓨터로 arc_lab3_template.json 을 다운로드 준비합니다.(강사배포 파일 사용)
2. AWS Management Console 의 **서비스** 메뉴에서 **Management & Governance > CloudFormation** 을 클릭합니다.
3. **Create stack** 을 클릭하고 아래 단계에 따라 스택을 생성합니다.

   1 단계: 템플릿 지정

   - **Template source:** **Upload a template file** 을 선택합니다.
   - **Upload a template file:** **Choose file** 을 클릭하고 다운로드한 **arc_lab3_template.json** 파일을 선택합니다.
   - **Next** 를 클릭합니다.

2 단계: 스택 세부 정보 지정

- **Stack name:** arclab3
- **Next** 를 클릭합니다.

3 단계: 스택 옵션 구성

- **Next** 를 클릭합니다.

4 단계: 검토

- **I acknowledge that...** 의 체크박스에 체크합니다.
- **Create stack** 을 클릭합니다.

AWS CloudFormation 에서는 이제 템플릿을 사용하여 리소스의 *스택* 을 생성합니다.

Stack info 탭을 클릭합니다.

- **Status** 가 **CREATE_COMPLETE** 로 변경될 때까지(약 10 분) 대기합니다.

  참고 필요한 경우 새로 고침 아이콘을 15 초마다 클릭하면 화면이 업데이트됩니다.

4. **Outputs** 탭을 클릭합니다.

AWS CloudFormation 스택에서 지정된 리소스 ID 및 리소스 링크와 같은 *출력 정보* 를 제공할 수 있습니다.

- **Endpoint:** 생성된 RDS 의 Database Endpoint 입니다(예: *inventory-db.c7x7ui272727.us-west-2.rds.amazonaws.com*).
- **Region:** 생성된 리소스들의 리전 코드입니다.

작업 1: VPC 생성

이 작업에서는 AWS 클라우드에서 새 VPC 를 생성합니다.

VPC 는 AWS 계정 전용 가상 네트워크입니다. VPC 는 AWS 클라우드에서 다른 가상 네트워크와 논리적으로 분리되어 있습니다. Amazon Elastic Compute Cloud(Amazon EC2) 인스턴스와 같은 AWS 리소스를 VPC 에서 시작할 수 있습니다. VPC 의 IP 주소 범위를 수정하고, 서브넷을 생성하고, 라우팅 테이블, 네트워크 게이트웨이 및 보안 설정을 구성할 수 있습니다.

5. AWS Management Console 의 **Services** 메뉴에서 **Networking & Content Delivery > VPC** 를 클릭합니다.

6. 화면 왼쪽 상단에 **New VPC Experience** 가 표시되면, **New VPC Experience** 가 선택되었는지 확인하십시오. 이 실습은 새로운 EC2 콘솔을 사용하도록 설계되었습니다.

VPC 관리 콘솔에는 몇 가지 VPC 아키텍처를 자동으로 생성할 수 있는 VPC 마법사가 있습니다. 그러나 이 실습에서는 VPC 구성 요소를 수동으로 생성합니다.

7. 왼쪽 탐색 창에서 **Your VPCs** 를 클릭합니다.

VPC 목록이 표시됩니다. 기본 VPC 가 제공되어 AWS 사용을 시작하면 리소스를 시작할 수 있습니다. 이후 실습에서 사용할 공유 VPC 도 있습니다. 하지만 지금은 자체 VPC 를 생성합니다.

VPC 의 CIDR 범위는 **10.0.0.0/16** 로 **10.0.x.x** 로 시작하는 모든 IP 주소가 포함됩니다. 이 범위에는 65,000 개 이상의 주소가 포함됩니다. 나중에 이 주소를 별도의 서브넷으로 분할할 것입니다.

8. **Create VPC** 를 클릭하고 다음을 구성합니다.

- **Name tag - optional:** Lab VPC
- **IPv4 CIDR:** 10.0.0.0/16

9. **Create VPC** 를 클릭한 다음 **Your VPCs** 를 클릭합니다.

10. **Lab VPC** 를 선택하고 유일하게 선택한 VPC 인지 확인합니다.

11. 페이지 하단에서 **Tags** 탭을 클릭합니다.

태그 는 리소스를 식별하는 데 유용합니다. 예를 들어, 태그를 사용하여 개발/테스트/프로덕션 환경 또는 비용 센터를 식별할 수 있습니다.

12. 위 VPC 목록에서 Actions 를 클릭하고 *"Edit VPC settings"를* 클릭합니다.

이 옵션은 다음과 같이 VPC 에 있는 Amazon EC2 인스턴스에 *친숙한* DNS 이름을 할당합니다.

ec2-52-42-133-255.us-west-2.compute.amazonaws.com

13. **"DNS setting"**으로 내려가서 **"Enable DNS hostnames"**를 체크해줍니다**.**

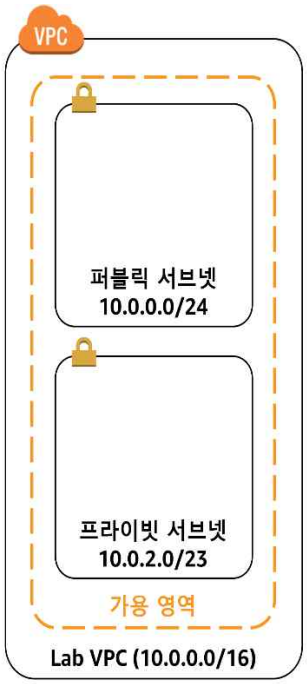
14. Save 를 클릭합니다.

이제 VPC 에서 실행된 Amazon EC2 인스턴스에서 DNS 호스트 이름을 자동으로 수신합니다. Amazon Route 53 을 사용하여 나중에 더욱 유의미한 DNS 이름(예: *app.company.com*)을 추가할 수도 있습니다.

작업 2: 서브넷 생성

서브넷 은 VPC 에 속하는 하위 범위의 IP 주소입니다. 지정된 서브넷으로 AWS 리소스를 시작할 수 있습니다. 인터넷에 연결해야 하는 리소스에는 *퍼블릭 서브넷* 을 사용하고, 인터넷과 격리된 상태를 유지해야 하는 리소스에는 *프라이빗 서브넷* 을 사용합니다.

이 작업에서는 다음 이미지와 같이 Lab VPC 에서 퍼블릭 서브넷과 프라이빗 서브넷을 생성합니다.



퍼블릭 서브넷 생성

퍼블릭 서브넷은 인터넷과 연결되는 리소스에 사용됩니다.

15. 왼쪽 탐색 창에서 **Subnets** 를 클릭합니다.
16. **Create subnet** 을 클릭하고 다음을 구성합니다.

- **VPC 에서 VPC ID :** "*Lab VPC*"를 선택합니다
- **Subnet name:** Public Subnet
- **Availability Zone:** 목록에서 **첫 번째** 가용 영역을 선택합니다(*No Preference* 를 선택하지 **마십시오**).
- **IPv4 CIDR block:** 10.0.0.0/24

17. **Create subnet** 를 클릭합니다.

참고 VPC 의 CIDR 범위는 **10.0.0.0/16** 이며, 여기에는 모든 **10.0.x.x** IP 주소가 포함됩니다. 방금 생성한 서브넷의 CIDR 범위는 **10.0.0.0/24** 이며, 모든 **10.0.0.x** IP 주소가 포함되어 있습니다. 이러한 범위는 서로 비슷해 보이지만 서브넷은 CIDR 범위가 **/24** 이기 때문에 VPC 보다 작습니다.

이제 서브넷 안에서 시작되는 모든 인스턴스에 퍼블릭 IP 주소가 자동으로 할당되도록 서브넷을 구성할 것입니다.

18. **Public Subnet** 을 선택(체크)합니다.
19. Actions 를 클릭하고 *Edit subnet settings* 를 선택합니다.
20. **Enable auto-assign public IPv4 address** 를 선택합니다.
21. Save 를 클릭합니다.

참고 이 서브넷의 이름이 **Public Subnet** 이지만 퍼블릭 상태는 아닙니다. 퍼블릭 서브넷에는 인터넷 게이트웨이가 있어야 합니다. 이후 실습에서 인터넷 게이트웨이를 생성하고 연결합니다.

프라이빗 서브넷 생성

프라이빗 서브넷은 인터넷과 격리된 상태를 유지해야 하는 리소스에 사용됩니다.

22. 방금 학습한 내용을 사용하여 다음 설정을 포함하여 다른 서브넷을 생성합니다.

- **VPC ID:** "*Lab VPC*"
- **Subnet name:** Private Subnet
- **Availability Zone:** 목록에서 **첫 번째** 가용 영역을 선택합니다(*No Preference* 를 선택하지 **마십시오**).
- **IPv4 CIDR block:** 10.0.2.0/23

23. Create subnet 를 클릭합니다.

참고 CIDR 블록 **10.0.2.0/23** 에는 **10.0.2.x** 및 **10.0.3.x** 로 시작하는 모든 IP 주소가 포함되어 있습니다. 인터넷에서 액세스할 수 있어야 하는 특별한 경우를 제외하고 프라이빗 서브넷은 대부분의 리소스를 프라이빗으로 유지해야 하기 때문에 크기가 퍼블릭 서브넷의 두 배입니다.

이제 VPC 에 서브넷이 2 개 있습니다. 그러나 완전히 격리되어 있기 때문에 VPC 밖의 리소스와 통신할 수 없습니다. 이제 인터넷 게이트웨이를 통해 인터넷에 연결되도록 퍼블릿 서브넷을 구성합니다.

작업 3: 인터넷 게이트웨이 생성

인터넷 게이트웨이 는 수평적 확장으로 이중화를 지원하는 고가용성 VPC 구성 요소로서 VPC 의 인스턴스와 인터넷 간 통신이 가능합니다. 인터넷 게이트웨이로 인해 네트워크 트래픽에 가용성 위험이나 대역폭 제약이 발생하지 않습니다.

인터넷 게이트웨이를 사용하는 목적은 다음 두 가지입니다.

- 라우팅 테이블에서 인터넷에 연결할 대상 제공
- 퍼블릭 IPv4 주소가 할당된 인스턴스에 네트워크 주소 변환(NAT) 실행

이 작업에서는 인터넷 트래픽이 퍼블릿 서브넷에 액세스할 수 있도록 인터넷 게이트웨이를 생성합니다.

24. 왼쪽 탐색 창에서 **Internet Gateways** 를 클릭합니다.
25. **Create internet gateway** 를 클릭하고 다음을 구성합니다.

- **Name tag:** Lab IGW

26. **Create internet gateway** 를 클릭합니다.

이제 사용자의 Lab VPC 에 인터넷 게이트웨이를 연결할 수 있습니다.

27. **Actions** 를 클릭하고 *Attach to VPC* 를 선택합니다.
28. **VPC** 에서 *Lab VPC* 를 선택합니다.
29. **Attach internet gateway** 를 선택합니다.

이제 인터넷 게이트웨이가 Lab VPC 에 연결됩니다. 인터넷 게이트웨이를 생성하여 VPC 에 연결했어도 퍼블릿 서브넷 라우팅 테이블도 인터넷 게이트웨이를 사용하도록 구성해야 합니다.

작업 4: 라우팅 테이블 구성하기

라우팅 테이블 은 네트워크 트래픽이 향하는 방향을 결정하는 데 사용되는 *경로* 라고 부르는 규칙 세트를 포함합니다. VPC 에 있는 각 서브넷은 라우팅 테이블에 연결되어 있어야 합니다. 테이블이 서브넷에 대한 라우팅을 제어합니다. 서브넷은 한 번에 하나의 라우팅 테이블에만 연결할 수 있지만, 여러 서브넷을 같은 라우팅 테이블에 연결할 수 있습니다.

인터넷 게이트웨이를 사용하려면 서브넷의 라우팅 테이블에 인터넷에 바인딩된 트래픽을 인터넷 게이트웨이로 향하도록 지시하는 경로가 포함되어야 합니다. 인터넷 게이트웨이로 가는 경로가 있는 라우팅 테이블에 서브넷이 연결된 경우 이를 *퍼블릭 서브넷* 이라고 합니다.

이 작업에서는 다음을 수행합니다.

- 인터넷 바운드 트래픽용 퍼블릭 라우팅 테이블 생성
- 인터넷 게이트웨이로 인터넷 바운드 트래픽을 보내는 라우팅 테이블에 경로 추가
- 퍼블릭 서브넷을 새 라우팅 테이블에 연결

30. 왼쪽 탐색 창에서 **Route Tables** 를 클릭합니다.

여러 라우팅 테이블이 표시되지만 Lab VPC 와 연결된 라우팅 테이블은 하나입니다. 이 라우팅 테이블은 트래픽을 로컬로 라우팅하기 때문에 *프라이빗 라우팅 테이블* 이라고 부릅니다.

31. **VPC** 열에 **Lab VPC** 를 표시하는 라우팅 테이블을 선택합니다. (열을 확장하여 이름을 볼 수 있습니다.)

32. **Name** 열에 커서를 놓고 연필 아이콘을 클릭합니다.
33. 이름을 `Private Route Table`로 입력한 다음 **Save**를 클릭합니다.
34. 생성된 라우팅 테이블의 라우팅 테이블 ID 를 클릭하여 들어가서 페이지 하단에서 **Routes** 탭에 정보가 보입니다.

경로는 단 하나입니다. **10.0.0.0/16**(Lab VPC 의 범위)로 향하는 모든 트래픽이 *로컬*로 라우팅 된다는 것을 알 수 있습니다. 따라서 VPC 내 모든 서브넷이 서로 통신할 수 있습니다.

이제 퍼블릭 트래픽을 인터넷 게이트웨이로 전송할 새 퍼블릭 라우팅 테이블을 생성합니다.

35. **Create route table**을 클릭하고 다음을 구성합니다.

- **Name - optional:** `Public Route Table`
- **VPC:** *Lab VPC*

36. **Create route table**를 클릭합니다.
37. **Routes** 탭에서 **Edit routes**를 클릭합니다.

이제 인터넷 바운드 트래픽(**0.0.0.0/0**)을 인터넷 게이트웨이로 보내는 경로를 추가합니다.

38. **Add route**를 클릭하고 다음을 구성합니다.

- **Destination:** `0.0.0.0/0`
- **Target:** *Internet Gateway* 및 *Lab IGW*를 선택합니다.

39. **Save changes**를 클릭합니다.

마지막 단계는 이 새 라우팅 테이블을 퍼블릭 서브넷과 연결하는 것입니다.

40. **Subnet associations** 탭을 클릭합니다.

41. Edit subnet associations 를 클릭합니다. (첫번째의 명시적 서브넷 연결에서)

42. **Public Subnet** 이 있는 행을 선택합니다.

43. Save associations 를 클릭합니다.

이 Public Subnet 은 인터넷 게이트웨이를 통해 인터넷으로 트래픽을 전송하는 라우팅 테이블 항목이 있기 때문에 이제 *퍼블릭* 서브넷입니다.

요약하면 다음과 같이 퍼블릭 서브넷을 생성할 수 있습니다.

- 인터넷 게이트웨이 생성
- 라우팅 테이블 생성
- **0.0.0.0/0** 트래픽을 인터넷 게이트웨이로 보내는 라우팅 테이블에 경로 추가
- 라우팅 테이블을 서브넷과 연결(이에 따라 *퍼블릿 서브넷* 이 됨)

작업 5: 앱 서버용 보안 그룹 생성

보안 그룹 은 인스턴스에 대한 인바운드 및 아웃바운드 트래픽을 제어하는 가상 방화벽 역할을 합니다. 보안 그룹은 *서브넷* 수준이 아니라 *인스턴스 네트워크 인터페이스* 수준에서 작동합니다. 따라서 각 인스턴스마다 트래픽을 제어하는 자체 방화벽이 있을 수 있습니다. 시작할 때 특정 그룹을 지정하지 않으면 인스턴스가 자동으로 VPC 의 *기본 보안 그룹* 에 할당됩니다.

이 작업에서는 사용자에게 HTTP 를 통한 앱 서버 액세스를 허용하는 보안 그룹을 생성합니다.

44. 왼쪽 탐색 창에서 **Security Groups** 를 클릭합니다.

45. Create security group 을 클릭하고 다음을 구성합니다.

- **Security group name:** App-SG
- **Description:** Allow web access
- **VPC:** *Lab VPC* 를 선택

참고 기존에 등록되어 있는 VPC 는 **X** 버튼을 눌러서 삭제 후 *Lab VPC* 를 선택하십시오.

46. **Inbound rules** 에서 Add Rule 을 클릭하고 다음을 구성합니다.

- **Type:** *HTTP*
- **Source:** *Anywhere-ipv4*

47. 페이지 하단에서 Create security group 을 클릭합니다.

인바운드 규칙은 인스턴스에 도달하는 것이 허용되는 트래픽을 결정합니다. 인터넷 어디서나(0.0.0.0/0) 나오는 HTTP(포트 80) 트래픽을 허용하도록 구성했습니다.

다음 작업에서는 이 애플리케이션 보안 그룹을 사용합니다.

작업 6: 퍼블릭 서브넷에서 앱 서버 시작

VPC 가 제대로 구성되었는지 테스트하기 위해 이제 Amazon EC2 인스턴스를 퍼블릭 서브넷에서 시작하여 인터넷에서 앱 서버에 액세스할 수 있는지 확인해보겠습니다.

48. Services 메뉴에서 **Compute > EC2** 를 클릭합니다.

49. 화면 왼쪽 상단에 **New EC2 Experience** 가 표시되면, **New EC2 Experience** 가 선택되었는지 확인하십시오. 이 실습은 새로운 EC2 콘솔을 사용하도록 설계되었습니다.

50. 페이지를 아래로 스크롤하고 Launch instance 를 클릭한 다음 *Launch instance* 를 선택합니다.

51. 다음을 구성합니다.

1 단계(태그 추가)

- **Name:** App Server

2 단계(AMI 선택)

- **AMI:** Amazon Linux 2 AMI 64-bit (x86)

3 단계(인스턴스 유형 선택)

- **Instance Type:** t2.micro (t2.micro 를 시작할 수 없는 경우 t3.micro 를 사용해 보십시오. 이는 리전에 따라 일부 인스턴스 유형은 사용할 수 없기 때문입니다.)

4 단계 (키페어)

- *Proceed without a key pair* 를 선택합니다.

5 단계(인스턴스 구성)

- 아래로 내려와서 **네트워크 설정**에서 **[편집]** 버튼을 누른다
- **VPC :** *Lab VPC* 를 선택
- **Subnet:** *Public Subnet*
- 퍼블릭 IP 자동 할당은 활성화로 설정하고
- 방화벽(보안 그룹)을 '기존 보안 그룹 선택' 으로 선택하고
- 일반 보안 그룹을 App-SG 로 선택한다
- 맨 마지막으로 스크롤하고 **Advanced Details** 섹션을 확장합니다
- **IAM 인스턴스 프로파일 :** *Inventory-App-Role* 을 선택해준다
- 다음을 복사하고 **User data** 에 붙여 넣습니다. (배포된 파일사용)

```
#!/bin/bash
# Install Apache Web Server and PHP
yum install -y httpd mysql
amazon-linux-extras install -y php7.2
# Download Lab files
wget https://s3.us-west-2.amazonaws.com/arclab.applaycrew.com/Lab3Files/inventory-app.zip
unzip inventory-app.zip -d /var/www/html/
# Download and install the AWS SDK for PHP
wget https://s3.us-west-2.amazonaws.com/arclab.applaycrew.com/Lab3Files/aws.zip
unzip aws -d /var/www/html
# Turn on web server
chkconfig httpd on
service httpd start
```

7 단계(인스턴스 구동)

- **Launch Instances** 를 클릭합니다.

상태 페이지에서 인스턴스 실행을 알립니다.

52. **View Instances** 를 클릭합니다.

계속하기 전에 인스턴스 상태가 **running** 으로 표시될 때까지 대기합니다.

팁 디스플레이를 업데이트하려면 주기적으로 새로 고침 아이콘을 클릭하십시오.

53. 인스턴스가 실행 중이면 **App Server** 를 선택하여 ID 를 클릭합니다.
54. APP Server 인스턴스 요약 내용에서 **IPv4 Public IP** 주소를 복사합니다.
55. 새 웹 브라우저 탭을 열고, IP 주소를 주소 표시줄에 붙여 넣은 다음 Enter 를 누릅니다.

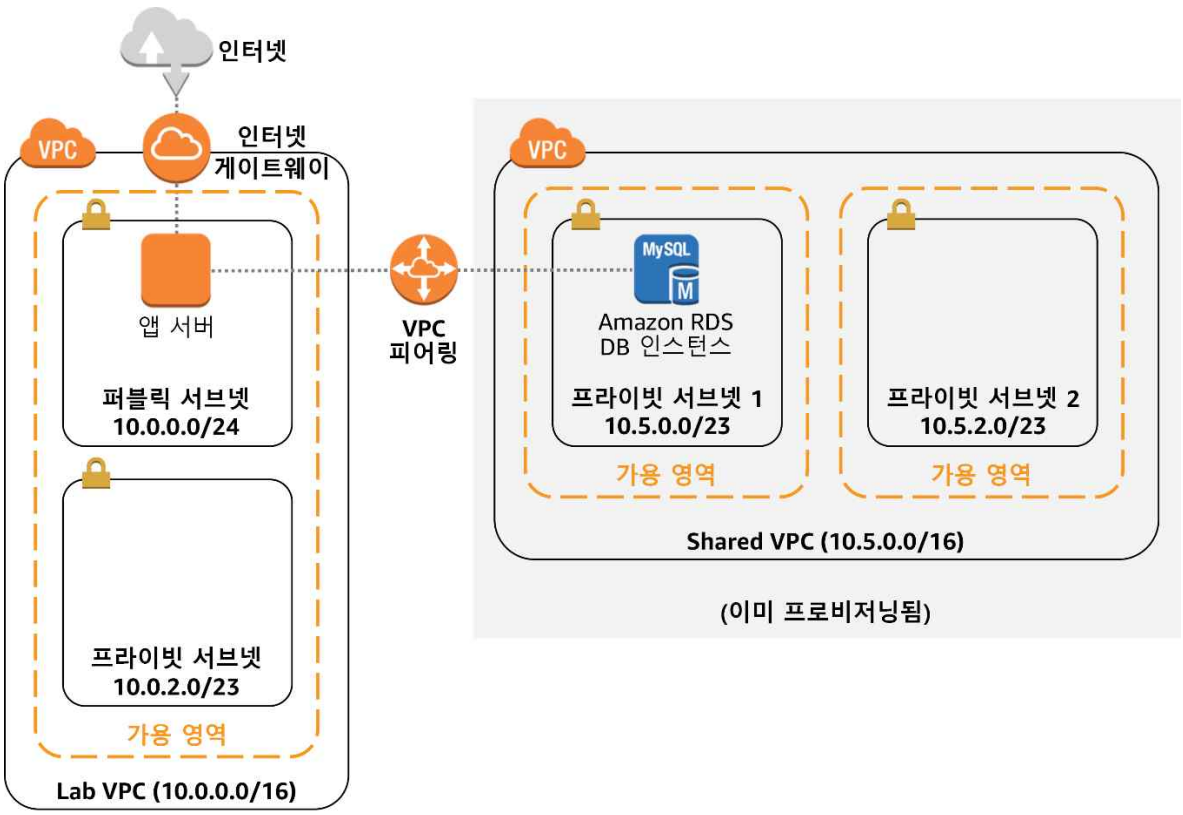
VPC 가 제대로 구성되었다면 Inventory 애플리케이션과 **Please configure settings to connect to database** 메시지가 표시될 것입니다. 데이터베이스 설정은 아직 구성되지 않았지만 Inventory 애플리케이션 모양을 보면 퍼블릭 서브넷이 제대로 구성되었는지 알 수 있습니다.

참고 Inventory 애플리케이션이 표시되지 않은 경우 60 초 동안 기다린 다음 브라우저 탭을 새로 고쳐 다시 시도하십시오. EC2 인스턴스가 부팅되어 소프트웨어를 설치하는 스크립트가 실행되기까지 몇 분이 소요될 수 있습니다.

챌린지: VPC 피어링 구성

참고 이 챌린지 작업은 **선택 사항**이며 실습 시간이 남는 경우에 제공됩니다. 실습 끝으로 건너뛰려면 이부분은 skip 하십시오.

Shared VPC 라고 하는 다른 VPC 가 이 실습의 일부로 제공되었습니다. 작업은 다음 아키텍처 다이어그램과 같이 Lab VPC 와 Shared VPC 사이의 피어링 연결을 생성하는 것입니다.



VPC 피어링 연결 은 두 VPC 사이에서 비공개 방식으로 트래픽을 라우팅할 수 있게 하는 두 VPC 사이의 네트워킹 연결입니다. 마치 같은 네트워크에 있는 것처럼 양쪽 VPC 의 인스턴스에서 서로 통신할 수 있습니다. 귀하의 VPC 사이에, 혹은 다른 AWS 계정에 속한 VPC 또는 다른 AWS 리전에 있는 VPC 와 VPC 피어링 연결을 생성할 수 있습니다.

데이터베이스에 이미 공유 VPC 가 프로비저닝되어 있습니다. VPC 피어링을 구성한 후에는 데이터베이스를 Inventory 애플리케이션에 연결하여 피어링이 제대로 구성되어 있는지 확인합니다.

피어링 연결 생성

56. AWS Management Console 의 **Services** 메뉴에서 **Networking & Content Delivery > VPC** 를 클릭합니다.

57. 왼쪽 탐색 창에서 **Peering Connections** 를 클릭합니다.

먼저 두 VPC 를 링크로 연결하는 VPC 구성 요소인 피어링 연결을 생성합니다.

58. **Create Peering Connection** 을 클릭하고 다음을 구성합니다.

- **Name - optional:** Lab-Peer
- **VPC ID (Requester):** *Lab VPC*
- **VPC ID (Accepter):** *Shared VPC*

59. **Create Peering Connection** 을 클릭합니다.

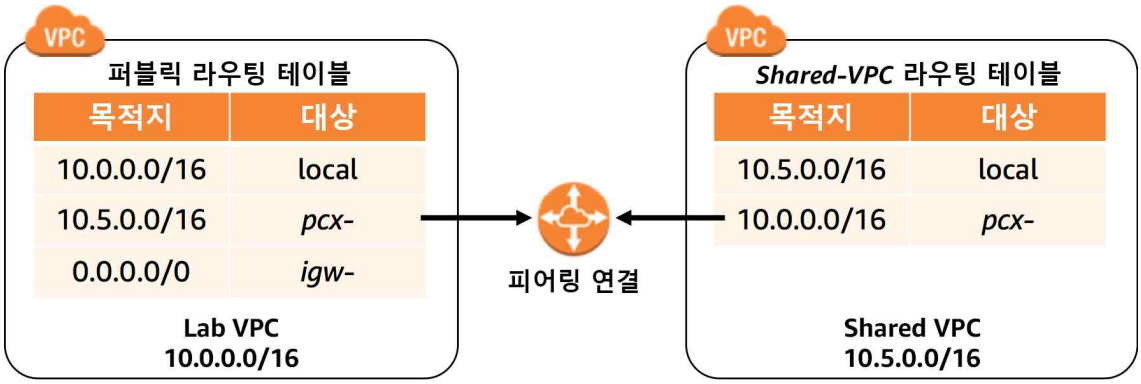
피어링 연결이 생성되면 대상 VPC 가 이를 *수락* 해야 합니다. 대상 VPC 를 다른 계정에서 소유하고 있거나 피어링 연결을 *생성* 하는 사용자가 대상 VPC 의 연결을 *수락* 할 권한이 없을 수 있기 때문입니다. 하지만 이 실습에서는 직접 연결을 수락합니다.

60. **Actions** 를 클릭하고 *Accept Request* 를 선택합니다.

61. **Accept request** 를 클릭합니다.

라우팅 테이블 구성

이제 다음 이미지와 같이 두 VPC 의 라우팅 테이블을 업데이트하여 Lab VPC 에서 피어링 연결로 트래픽을 전송합니다.



62. 왼쪽 탐색 창에서 **Route Tables** 를 클릭합니다.

63. **Public Route Table** 을 선택합니다.

대상 IP 주소가 Shared VPC 범위에 있는 경우 피어링 연결로 트래픽을 전송하도록 Lab VPC 와 연결된 퍼블릭 라우팅 테이블을 구성합니다.

64. **Routes** 탭에서 Edit routes 를 클릭합니다.

65. Add route 를 클릭하고 다음을 구성합니다.

- **Destination:** 10.5.0.0/16 (Shared VPC 의 CIDR 범위.)
- **Target:** *Peering Connection* 을 선택하고 *Lab-Peer* 를 선택합니다.

66. Save changes 를 클릭합니다.

이제 Shared VPC 에서 Lab VPC 로 향하는 트래픽의 역방향 흐름을 구성합니다.

67. 왼쪽 탐색 창에서 **Route Tables** 를 클릭합니다.

68. **Shared-VPC Route Table** 을 선택하고 유일하게 선택한 라우팅 테이블인지 확인합니다.

이는 Shared VPC 용 라우팅 테이블입니다. 대상 IP 주소가 Lab VPC 범위 내에 있는 경우 피어링 연결로 트래픽을 전송하도록 구성합니다.

69. 선택된 라우팅 테이블 ID 를 클릭하여 하단 **Routes** 탭에서 Edit routes 를 클릭합니다.

70. Add route 를 클릭하고 다음을 구성합니다.

- **Destination:** 10.0.0.0/16 (Lab VPC 의 CIDR 범위.)
- **Target:** *Peering Connection* 을 선택하고 *Lab-Peer* 를 선택합니다.
- Save changes 를 클릭합니다.

이제 트래픽이 다른 VPC 용으로 정해질 때 피어링 연결을 통해 트래픽을 전송하도록 라우팅 테이블이 구성되었습니다.

피어링 연결 테스트

데이터베이스에 이미 공유 VPC 가 프로비저닝되어 있습니다. 이제 피어링 연결을 통해 데이터베이스에 액세스하도록 Inventory 애플리케이션을 구성하여 피어링 연결을 테스트합니다.

71. Inventory 애플리케이션을 사용하여 웹 브라우저 탭으로 돌아갑니다.

72. **Settings** 를 클릭하고 다음을 구성합니다.

    AWS 콘솔 브라우저에서 CloudFormation 의 스택에서 arclab3 을 클릭하면 스택 정보가 나오는데 output(출력)탭을 선택하여 아래와 같은 형태의 Endpoint 값을 클립보드로 복사한다

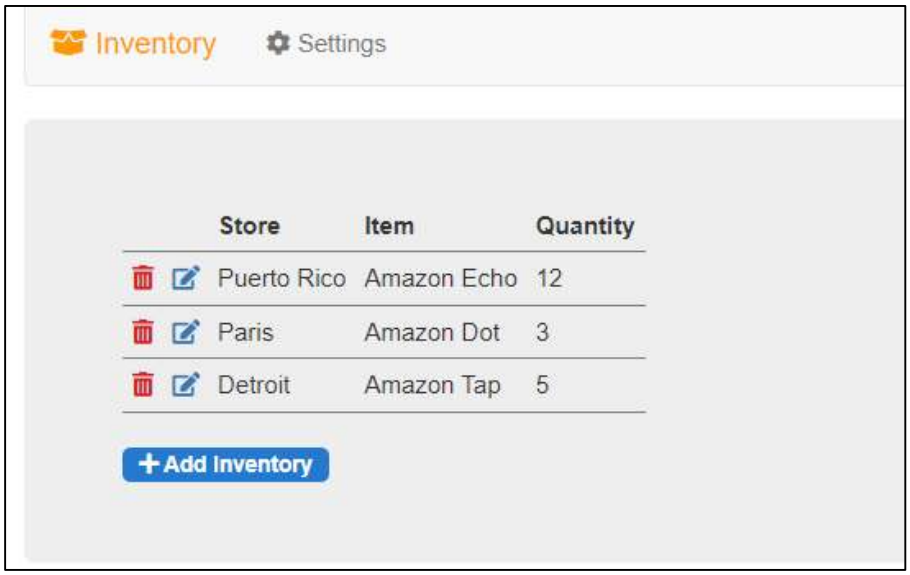
    "inventory-db.cvqmolmcdbyj.us-east-1.rds.amazonaws.com"

- **Endpoint:** 생성된 CloudFormation Stack 의 Outputs 중 **Endpoint** 값을 붙여 넣습니다.
- **Database:** inventory

- **Username:** dbadmin
- **Password:** lab-password

73. **Save** 를 클릭합니다.

이제 애플리케이션에 데이터베이스의 데이터가 표시되어야 합니다.



이를 통해 Shared VPC 에 인터넷 게이트웨이가 없기 때문에 피어링 연결이 작동한다는 것을 알 수 있습니다. 피어링 연결을 통해서만 데이터베이스에 액세스할 수 있습니다.

결론

축하합니다! 다음 작업이 성공적으로 완료되었습니다.

- VPC 생성
- 퍼블릭 및 프라이빗 서브넷 생성
- 인터넷 게이트웨이 생성
- 라우팅 테이블 구성 및 서브넷에 연결

실습 종료

다음 순서 따라 실습 과정에서 생성된 리소스를 정리하십시오.

1. **EC2:** App Server Instance 삭제
2. **VPC:** Peering Connections ---> Lab-Peer 삭제

   Delete related route table entries (관련 라우팅 테이블 항목 삭제) **선택**
3. **VPC:** Your VPCs ---> Lab VPC 삭제
4. **CloudFormation:** Stack 삭제, 장시간 소요, Shared VPC/R/보안 그룹이 모두 삭제됨
5. **S3:** Bucket(cf-templates 으로 시작하는 이름)의 파일과 Bucket 삭제
6. **Amazone RDS →스냅샷**도 반드시 직접 삭제한다
7. **끝.**