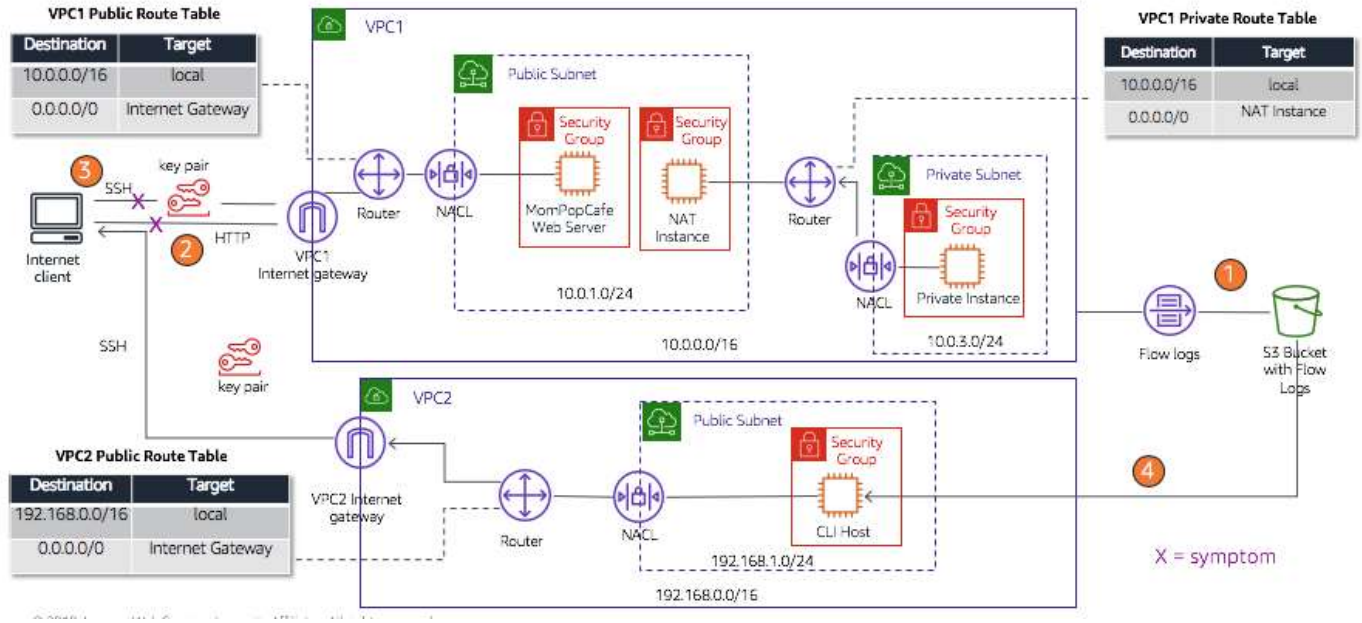
활동 - VPC 문제 해결

활동 개요

이번 활동에서는 VPC 흐름 로그를 사용하고 Amazon VPC 구성의 문제를 해결하는 연습을 합니다. 아래 다이어그램에 표시된 것과 같이 VPC 2 개가 포함된 시작 환경이 제공됩니다.

이 다이어그램에는 번호가 붙은 주황색 동그라미 4 개(1 번~4 번)로 활동 진행 순서가 표시되어 있습니다.



포함되는 작업은 다음과 같습니다.

1. 흐름 로그를 보관하기 위한 Amazon Simple Storage Service(Amazon S3) 버킷을 **생성**하고 VPC 1.2 에서 VPC 흐름 로그를 사용 설정합니다.
2. 이번 과제는 2 개의 부분으로 구성되며 각 부분은 다이어그램에 2 번과 3 번으로 레이블이 지정되어 있습니다. 이러한 부분을 거치며 여러분은 다음 작업을 하게 됩니다.
    1. VPC1 네트워크 구성의 **문제를 해결**합니다. VPC1 에는 2 개의 인스턴스가 포함된 *퍼블릭 서브넷*이 있습니다. 이중 하나는 웹사이트를 실행하는 Cafe Web Server 인스턴스입니다. 퍼블릭 서브넷에서 VPC2 인터넷 게이트웨이로 네트워크 트래픽을 매핑하는 *퍼블릭 라우팅 테이블*이 있습니다. 이러한 방식으로 네트워크가 구성되어야 합니다. 하지만 VPC1 에는 해결이 필요한 문제가 2 개 있습니다.
    2. VPC1 의 **문제를 해결**합니다. CLI 호스트 인스턴스가 제공됩니다. CLI 호스트에서 AWS Command Line Interface(AWS CLI) 명령을 실행하게 됩니다. CLI 호스트는 문제가 발생한 Virtual Private Cloud(VPC) 외부에서 실행되어야 합니다. 따라서

이름이 *VPC2* 인 두 번째 VPC 가 생성되었습니다. CLI 호스트는 VPC2 에서 실행됩니다. VPC2 에는 구성 문제가 없습니다. CLI 호스트 인스턴스는 물리적 머신이 인터넷을 통해 VPC1 에 연결할 때와 동일한 방법으로 VPC1 리소스에 연결합니다. VPC1 과 VPC2 는 피어링되지 않은 상태입니다.

3. 활동을 시작한 후 발생한 증상에 관한 **문제를 해결**합니다. 문제는 다이어그램에 빨간색 X 로 표시되어 있습니다. 하지만 *증상*과 *근본 원인*은 다르다는 점에 유의하십시오.
4. 흐름 로그를 CLI 호스트에 **다운로드**하고 로그 항목을 분석합니다. 문제 해결 중에 취한 조치가 흐름 로그에 반영되어 있습니다.

소요 시간

이 활동을 완료하는 데는 약 **75 분**이 소요됩니다.

활동 목표

이 활동을 완료하면 다음을 할 수 있게 됩니다.

- VPC 구성 문제를 **파악**합니다.
- VPC 구성 문제를 **해결**합니다.
- VPC 흐름 로그를 **사용 설정**합니다.
- grep 을 사용하여 흐름 로그를 **분석**합니다.

비즈니스 사례 관련성

카페 경영진이 새로운 요청을 했습니다.

최근 카페 웹 서버가 실행되는 VPC 의 구성에 몇 가지 변경사항이 적용되었습니다. 카페 배포의 네트워크 보안을 개선하려는 뚜렷한 의도를 가지고 실행된 변경이었지만 고객들이 웹사이트에 액세스할 수 없다며 불평하기 전까지 몇 가지 문제가 발견되지 않았습니다. 문제를 조사하기 시작한 Nikhil 과 Sofía 는 얼마 지나지 않아 Amazon Elastic Compute Cloud(Amazon EC2) 인스턴스에서 실행되는 웹 서버에 대한 보안 셸(SSH) 연결을 설정하지 못했다는 사실을 발견했습니다. 웹 서버에 연결할 수 없었기 때문에 EC2 인스턴스에 발생했을지도 모를 구성 오류를 진단하지도 못했습니다. Sofía 와 Nikhil 은 해결해야 할 근본적인 네트워킹 구성 오류가 있다고 생각합니다.

이 문제를 불만스럽게 생각하는 Martha 와 Frank 는 문제를 최대한 빨리 해결하고 싶어 합니다. 이번 활동에서는 Nikhil 과 Sofía 의 역할을 맡아 최선을 다해 문제를 해결합니다.

활동 단계

활동 환경 시작

이 실습의 결과는 역동적인 VPN 연결에 의해 영향을 받게 될지도 모릅니다.

5. 지침의 맨 위에서 실습 시작(Start Lab)을 클릭하여 실습을 시작합니다.

   실습시작(Start Lab) 패널이 열리고 실습 상태가 표시됩니다.

6. "**실습 상태: 준비(Lab status: ready)**" 메시지가 표시되면 **X** 를 클릭하여 실습 시작(Start Lab) 패널을 닫습니다.

과제 1: SSH 를 사용해 CLI 호스트 EC2 인스턴스에 연결

이번 과제에서는 AWS CLI 가 이미 설치된 기존 Amazon Linux CLI 호스트 EC2 인스턴스에 연결합니다. 인스턴스에 연결할 때 보안 셸(SSH)을 사용합니다.

Windows 사용자는 과제 1.1 을 따라야 합니다. macOS 및 Linux 사용자는 모두 과제 1.2 를 따라야 합니다.

macOS/Linux 사용자 - 로그인 지침을 보려면 여기를 클릭합니다.

Windows 사용자용 과제 1.1: SSH

이 지침은 Windows 사용자에게만 적용됩니다.

macOS 또는 Linux 를 사용하는 경우 다음 섹션으로 건너뛰십시오.

7. 작업을 완료하기 전에 이 단계에 포함된 3 개의 주요 항목을 읽어보십시오. 세부 정보(Details) 패널을 연 후에는 이러한 지침을 볼 수 없습니다.
    - 현재 읽고 있는 지침 위에 있는 세부 정보(Details) 드롭다운 메뉴를 클릭한 다음 보기(Show)를 클릭합니다. 보안 인증(Credentials) 창이 열립니다.
    - **PPK 다운로드(Download PPK)** 버튼을 클릭하고 **labsuser.ppk** 파일을 저장합니다. 브라우저에서 이 파일은 일반적으로 다운로드(Downloads) 디렉터리에 저장됩니다.
    - **X** 를 클릭하여 세부 정보(Details) 패널을 닫습니다.
8. 필요한 소프트웨어를 다운로드합니다.
    - **PuTTY** 를 사용하여 SSH 를 통해 Amazon EC2 인스턴스에 연결합니다. 컴퓨터에 PuTTY 가 설치되어 있지 않은 경우 여기에서 다운로드하십시오.
9. **putty.exe** 를 엽니다.
10. 시간 초과가 발생하지 않도록 다음과 같이 PuTTY 를 구성합니다.
    - **연결(Connection)**을 클릭합니다.
    - **킵얼라이브 간 시간차(초)(Seconds between keepalives)**를 **30** 으로 설정합니다.

    이렇게 하면 PuTTY 세션을 더 오래 열어둘 수 있습니다.

11. 다음과 같이 PuTTY 세션을 구성합니다.
    - **세션(Session)**을 클릭합니다.
    - **Host Name (or IP address)**(호스트 이름(또는 IP 주소)): CLI 호스트 인스턴스의 **IPv4 퍼블릭 IP 주소**를 복사하여 붙여넣습니다. 주소를 찾으려면 현재 읽고 있는 지침 위에 있는 세부 정보(Details) 드롭다운 메뉴를 클릭한 다음 보기(Show)를 클릭합니다. *CliHostIP* 값을 복사합니다.
    - PuTTy 로 돌아간 후 **연결(Connection)** 목록에서 **SSH** 를 확장합니다.
    - **Auth** 를 클릭합니다(확장하지 말 것).
    - **탐색(Browse)**을 클릭합니다.
    - 다운로드한 lab#.ppk 파일을 찾아 선택합니다.
    - **열기(Open)**를 클릭하여 선택합니다.
    - **열기(Open)**를 클릭합니다.
12. 호스트를 신뢰하고 호스트에 연결하려면 **예(Yes)**를 클릭합니다.

13. **로그인(login as)** 메시지가 나타나면 **`ec2-user`**를 입력합니다.

    그러면 EC2 인스턴스에 연결됩니다.

14. Windows 사용자: 다음 과제로 건너뛰려면 여기를 클릭하십시오.

macOS/Linux 사용자용 과제 1.1: SSH

이 지침은 Mac/Linux 사용자에게만 적용됩니다. Windows 사용자인 경우 다음 과제로 건너뜁니다.

15. 작업을 완료하기 전에 이 단계에 포함된 3 개의 주요 항목을 읽어보십시오. 세부 정보(Details) 패널을 연 후에는 이러한 지침을 볼 수 없습니다.
    - 현재 읽고 있는 지침 위에 있는 세부 정보(Details) 드롭다운 메뉴를 클릭한 다음 보기(Show)를 클릭합니다. 보안 인증(Credentials) 창이 열립니다.
    - **PEM 다운로드(Download PEM)** 버튼을 클릭하고 **labsuser.pem** 파일을 저장합니다.
    - **X**를 클릭하여 세부 정보(Details) 패널을 닫습니다.
16. 터미널 창을 열고 디렉터리를 labsuser.pem 파일이 다운로드된 디렉터리로 변경(cd)합니다.

    예를 들어 다운로드(Downloads) 디렉터리에 저장된 경우 다음 명령을 실행합니다.

    ```
    cd ~/Downloads
    ```

17. 다음 명령을 실행하여 키에 대한 권한을 읽기 전용으로 변경합니다.

    ```
    chmod 400 labsuser.pem
    ```

18. CLI 호스트 인스턴스의 **IPv4 Public IP 주소**를 복사하여 붙여넣습니다. 주소를 찾으려면 현재 읽고 있는 지침 위에 있는 세부 정보(Details) 드롭다운 메뉴를 클릭한 다음 보기(Show)를 클릭합니다.
19. *CliHostIP* 값을 복사합니다.
20. 터미널 창으로 돌아가서 다음 명령을 실행합니다(**<public-ip>**를 복사한 실제 퍼블릭 IP 주소로 바꿈).

    ```
    ssh -i labsuser.pem ec2-user@<public-ip>
    ```

21. 이 원격 SSH 서버에 대한 첫 번째 연결을 허용할 것인지 묻는 메시지가 나타나면 `yes` 를 입력합니다.

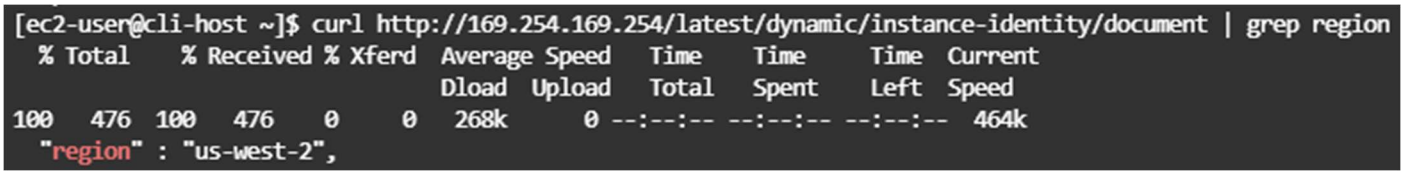
    인증에 키 페어를 사용 중이므로 암호를 묻는 메시지는 나타나지 않습니다.

과제 1.2: CLI 호스트 EC2 인스턴스의 AWS CLI 구성

22. 다음 방법으로 CLI 호스트 인스턴스가 실행 중인 리전을 찾습니다.

    ```
    curl http://169.254.169.254/latest/dynamic/instance-identity/document | grep region
    ```

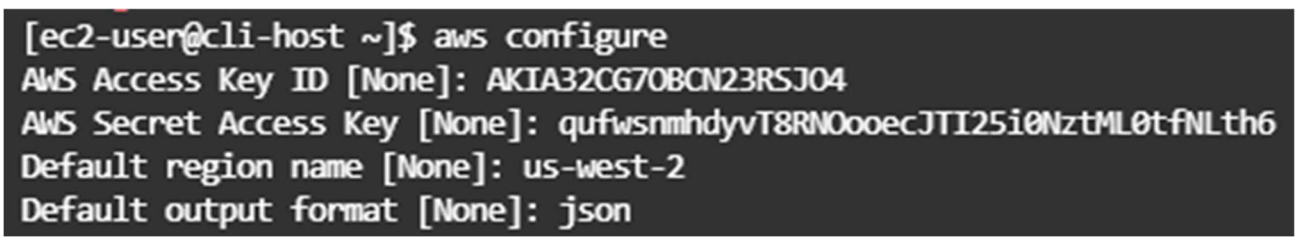
    잠시 후 이 리전 정보가 필요합니다.



23. 보안 인증을 사용하여 AWS CLI 소프트웨어를 업데이트합니다.

    ```
    aws configure
    ```

24. 메시지가 나타나면 다음 정보를 입력합니다.
    - **AWS 액세스 키 ID**: 지침 상단에 있는 세부 정보(Details) 드롭다운 메뉴를 클릭한 다음 보기(Show)를 클릭합니다. **AccessKey** 값을 복사하여 터미널 창에 붙여넣습니다.
    - **AWS 보안 액세스 키**: 동일한 보안 인증(Credentials) 화면에서 **SecretKey** 값을 복사해 붙여넣습니다.
    - **기본 리전 이름(Default region name)**: EC2 인스턴스가 실행 중인 리전 이름을 입력합니다. 이 정보는 방금 전에 찾은 것입니다. 예를 들어 `us-east-1` 또는 `eu-west-2` 를 입력합니다.
    - **기본 출력 형식(Default output format)**: `json`



과제 2: VPC 흐름 로그 사용 설정

첫 번째 과제에서는 VPC 흐름 로그에서 흐름 로그를 보관하기 위한 S3 버킷을 생성합니다. 그러고 나서 VPC1 의 네트워크 인터페이스 사이의 IP 트래픽에 대한 정보를 흐름 로그에 캡처하기 위해 VPC1 에서 VPC 흐름 로그를 사용 설정합니다. 이렇게 하면 흐름 로그가 S3 버킷에 저장됩니다.

25. 다음 명령을 사용하여 흐름 로그를 보관할 S3 버킷을 생성합니다.

```
aws s3api create-bucket --bucket flowlog#### --region <region> --
create-bucket-configuration LocationConstraint=<region>
```

명령에서 **####** 기호를 임의의 숫자 4 개로 바꾸고 **<region>** 2 개를 모두 EC2 인스턴스가 생성된 리전(예: **eu-west-2**)으로 바꿉니다.

리전이 *us-east-1* 이라면 명령을 실행하기 전에 명령에서 **--create-bucket-configuration LocationConstraint=<region>** 부분을 삭제합니다.

26. 다음 명령을 실행하여 VPC1 의 VPC ID 를 확인합니다. 이 ID 는 VPC 흐름 로그를 사용 설정하는 데 필요합니다.

```
aws ec2 describe-vpcs --query
'Vpcs[*].[VpcId,Tags[?Key==`Name`].Value,CidrBlock]' --filters
"Name=tag:Name,Values='VPC1'"
```

27. 다음 명령을 실행하여 VPC1 의 VPC 흐름 로그를 사용 설정합니다.

```
aws ec2 create-flow-logs --resource-type VPC --resource-ids <vpc-
id> --traffic-type ALL --log-destination-type s3 --log-destination
arn:aws:s3:::<flowlog####>
```

위 명령에서 **<flowlog####>** 부분을 실제 버킷 이름으로 바꿉니다. 또한 **<vpc-id>**를 VPC1 의 실제 VPC ID 로 바꿉니다. VPC ID 는 여러분이 실행한 **describe-vpcs** 명령에서 반환된 값입니다. 지침 위에 있는 세부 정보(Details) 드롭다운 메뉴에서 보기(Show)를 클릭해도 찾을 수 있습니다.


```
{
    "Unsuccessful": [],
    "FlowLogIds": [
        "fl-0d6e1ee1d8a04ea3a"
    ],
    "ClientToken": "P4lh9YKYG2VmR0oDRQlKIhpTTfmDtBq1TpTblmiUyGI="
}
```


28. 명령이 정상적으로 실행되었다면 **FlowLogId** 와 **ClientToken** 이 반환됩니다.
29. 다음 명령을 실행하여 흐름 로그가 생성되었는지 확인합니다.

```
aws ec2 describe-flow-logs
```


```
{
    "FlowLogs": [
        {
            "LogDestinationType": "s3",
            "Tags": [],
            "ResourceId": "vpc-0cb4973697fdec13d",
            "CreationTime": "2023-08-03T13:01:22.292Z",
            "TrafficType": "ALL",
            "FlowLogStatus": "ACTIVE",
            "LogFormat": "${version} ${account-id} ${interface-id} ${srcaddr} ${dstaddr} ${srcport} ${dstport} ${protocol} ${packets} ${bytes} ${start} ${end} ${
action} ${log-status}",
            "FlowLogId": "fl-0d6e1ee1d8a04ea3a",
            "MaxAggregationInterval": 600,
            "LogDestination": "arn:aws:s3:::flowlog1234",
            "DeliverLogsStatus": "SUCCESS"
        }
    ]
}
```


명령 출력 결과에는 **FlowLogStatus** 가 *ACTIVE* 인 흐름 로그 1 개가 생성된 것과 S3 버킷에 연결된 로그 대상을 확인할 수 있습니다.

이제 흐름 로그가 생성되었으니 활동의 다음 과제로 이동합니다. 여기에서는 문제 해결을 다룹니다. 활동 끝부분에서는 과제 3 에서 실행한 작업에 의해 생성된 흐름 로그를 분석합니다.

과제 3: 리소스에 대한 액세스 분석 및 문제 해결

이번 과제에서는 웹 서버 인스턴스에 대한 액세스를 분석하고 몇 가지 네트워킹 문제를 해결합니다. 카페 웹 서버 인스턴스가 VPC1 의 *퍼블릭 서브넷*에서 실행된다는 점을 기억합니다. 이 활동 시작 부분의 다이어그램을 다시 참고하여 네트워크 구성 방법을 자세히 알아봅니다.

30. **WebServerIP** 퍼블릭 IP 주소를 복사합니다. 지침 위에 있는 세부 정보(Details) 드롭다운 메뉴에서 보기(Show)를 클릭하면 주소를 찾을 수 있습니다.
31. 새 브라우저 탭을 열고 URL 표시줄에 IP 주소를 붙여넣은 후 웹페이지 로드를 시도해봅니다.

    잠시 후 페이지 로드가 실패하고 *ERR_CONNECTION_TIMED_OUT* 메시지가 표시됩니다. 이는 정상적인 동작입니다.

    나중에 돌아올 수 있도록 브라우저 탭을 열어둡니다.

32. CLI 호스트에 연결된 터미널에서 다음 명령을 실행하여 웹 서버 인스턴스에 대한 세부 정보를 찾습니다.

    ```
    aws ec2 describe-instances --filter "Name=ip-
    address,Values='<WebServerIP>'"
    ```

    위 명령에서 **<WebServerIP>**를 실제 WebServerIP 주소로 바꿉니다. 주소는 지침 위에 있는 세부 정보(Details) 드롭다운 메뉴에서 찾을 수 있습니다.

```
{
    "Reservations": [
        {
            "Instances": [
                {
                    "Monitoring": {
                        "State": "disabled"
                    },
                    "PublicDnsName": "",
                    "State": {
                        "Code": 16,
                        "Name": "running"
                    },
                    "EbsOptimized": false,
                    "LaunchTime": "2023-08-03T12:49:08.000Z",
                    "PublicIpAddress": "34.214.108.241",
                    "PrivateIpAddress": "10.0.1.200",
                    "ProductCodes": [],
                    "VpcId": "vpc-0cb4973697fdec13d",
                    "CpuOptions": {
                        "CoreCount": 1,
                        "ThreadsPerCore": 2
                    },
                    "StateTransitionReason": "",
                    "InstanceId": "i-0f6620c74c679fc8d",
                    "EnaSupport": true,
                    "ImageId": "ami-01f8103a2082c0718",
                    "PrivateDnsName": "ip-10-0-1-200.us-west-2.compute.internal",
                    "KeyName": "vockey",
                    "SecurityGroups": [
                        {
                            "GroupName": "c88654a1914577l4528830t1w811897679940-WebSecurityGroup-E08WHL4CRMZ5",
                            "GroupId": "sg-08a7f7ba1fdb5aa23"
                        }
                    ],
                    "ClientToken": "c8865-WebIn-Y3JQH48IOLXZ",
                    "SubnetId": "subnet-087d22cddbfb02d9f",
                    "InstanceType": "t3.micro",
                    "CapacityReservationSpecification": {
                        "CapacityReservationPreference": "open"
                    },
                    "NetworkInterfaces": [
                        {
                            "Status": "in-use",
                            "MacAddress": "06:fb:8d:b5:3a:cb",
                            "SourceDestCheck": false,
```

```
                "VpcId": "vpc-0cb4973697fdec13d",
                "Description": "",
                "NetworkInterfaceId": "eni-019205face8e6f5e9",
                "PrivateIpAddresses": [
                    {
                        "PrivateIpAddress": "10.0.1.200",
                        "Primary": true,
                        "Association": {
                            "PublicIp": "34.214.108.241",
                            "PublicDnsName": "",
                            "IpOwnerId": "amazon"
                        }
                    }
                ],
                "SubnetId": "subnet-087d22cddbfb02d9f",
                "InterfaceType": "interface",
                "Attachment": {
                    "Status": "attached",
                    "DeviceIndex": 0,
                    "DeleteOnTermination": true,
                    "AttachmentId": "eni-attach-082ea96625277863e",
                    "AttachTime": "2023-08-03T12:49:08.000Z"
                },
                "Groups": [
                    {
                        "GroupName": "c88654a1914577l4528830t1w811897679940-WebSecurityGroup-E08WHL4CRMZ5",
                        "GroupId": "sg-08a7f7ba1fdb5aa23"
                    }
                ],
                "Ipv6Addresses": [],
                "OwnerId": "811897679940",
                "PrivateIpAddress": "10.0.1.200",
                "Association": {
                    "PublicIp": "34.214.108.241",
                    "PublicDnsName": "",
                    "IpOwnerId": "amazon"
                }
            }
        ],
        "SourceDestCheck": false,
        "Placement": {
            "Tenancy": "default",
            "GroupName": "",
            "AvailabilityZone": "us-west-2a"
        },
```

```
                "Hypervisor": "xen",
                "BlockDeviceMappings": [
                    {
                        "DeviceName": "/dev/xvda",
                        "Ebs": {
                            "Status": "attached",
                            "DeleteOnTermination": true,
                            "VolumeId": "vol-065535a4a132acc21",
                            "AttachTime": "2023-08-03T12:49:09.000Z"
                        }
                    }
                ],
                "Architecture": "x86_64",
                "RootDeviceType": "ebs",
                "IamInstanceProfile": {
                    "Id": "AIPA32CG7OBCIDPF7VKJB",
                    "Arn": "arn:aws:iam::811897679940:instance-profile/c88654a191457714528830t1w811897679940-CafeInstanceProfile-bdZFcr5Z9s2w"
                },
                "RootDeviceName": "/dev/xvda",
                "VirtualizationType": "hvm",
                "Tags": [
                    {
                        "Value": "Cafe Web Server",
                        "Key": "Name"
                    },
                    {
                        "Value": "WebInstance",
                        "Key": "aws:cloudformation:logical-id"
                    },
                    {
                        "Value": "c88654a191457714528830t1w811897679940",
                        "Key": "aws:cloudformation:stack-name"
                    },
                    {
                        "Value": "arn:aws:cloudformation:us-west-2:811897679940:stack/c88654a191457714528830t1w811897679940/c80ab000-31fb-11ee-809d-0a0b4305a
0ab",
                        "Key": "aws:cloudformation:stack-id"
                    },
                    {
                        "Value": "c88654a191457714528830t1w811897679940",
                        "Key": "cloudlab"
                    }
                ],
                "HibernationOptions": {
                    "Configured": false
                },
                "MetadataOptions": {
                    "State": "applied",
                    "HttpEndpoint": "enabled",
                    "HttpTokens": "optional",
                    "HttpPutResponseHopLimit": 1
                },
                "AmiLaunchIndex": 0
            }
        ],
        "ReservationId": "r-01491fec995153970",
        "RequesterId": "658754138699",
        "Groups": [],
        "OwnerId": "811897679940"
    }
]
}
```


큰 JavaScript Object Notation(JSON) 문서가 반환되며, 이 문서에서 문제 해결에 필요한 것 이상의 세부 정보를 확인할 수 있습니다.

33. 관련 있는 세부 정보만 반환되도록 쿼리 파라미터를 사용하여 클라이언트 측에서 결과를 필터링합니다.

    다음 명령을 사용하면 인스턴스 상태, 프라이빗 IP 주소, 인스턴스 ID, 인스턴스에 적용되는 보안 그룹, 인스턴스가 실행되는 서브넷, 인스턴스와 연결된 키 페어 이름만 반환됩니다. *<WebServerIP>*를 실제 WebServerIP 주소로 바꾸는 것을 잊지 마십시오.

```
aws ec2 describe-instances --filter "Name=ip-
address,Values='<WebServerIP>'" --query
```

```
'Reservations[*].Instances[*].[State,PrivateIpAddress,InstanceId,SecurityGroups,SubnetId,KeyName]'
```

명령 결과에서 인스턴스가 실행되고 있다는 좋은 소식을 알 수 있습니다. 또한 활동의 뒷부분에서 사용할 수 있는 유용한 추가 정보도 반환됩니다.

34. *웹 서버* 인스턴스와의 SSH 연결 설정을 시도합니다. 방법은 다음과 같습니다.

`exit` 을(를) 입력하여 현재 CLI 호스트에 대한 SSH 세션의 연결을 해제합니다.

- 사용 중인 운영 체제에 맞는 단계를 완료합니다.
  - **Windows 사용자:**
    - 앞서 다운로드한 putty.exe 파일을 실행하여 **PuTTY** 를 시작합니다.
    - **호스트 이름 (또는 IP 주소)[Host Name (or IP address)]**에는 **WebServerIP** 주소를 입력합니다. 주소는 지침 위에 있는 세부 정보(Details) 드롭다운 메뉴에서 보기(Show)를 클릭하여 찾을 수 있습니다.
    - **SSH** 를 확장하고 **Auth** 를 클릭합니다.
    - **탐색(Browse)**을 클릭하고 앞서 다운로드한 .ppk 파일을 엽니다.
    - **연결(Connection)**을 클릭하고 **킵얼라이브 간 시간(초, 0 부터 끌 때까지)[Seconds between keepalives (0 to turn off)]:**를 `10` 으로 설정합니다.
    - **세션(Session)**[카테고리(Category) 패널 상단]을 클릭합니다.

저장된 세션(Saved Sessions) 텍스트 영역에서 `webserver` 을(를) **입력합니다**.

- **저장(Save)**을 클릭합니다.
- 마지막으로 **열기(Open)**를 클릭합니다.

- **macOS 및 Linux 사용자**
  - 동일한 터미널 창에서 키보드의 위쪽 화살표 키를 사용하여 앞서 사용한 SSH 연결 세부 정보를 로드합니다. 하지만 이번에는 연결 세부 정보 끝부분의 퍼블릭 IP 주소를 **WebServerIP** 주소로 변경합니다. 주소는 지침 위에 있는 세부 정보(Details) 드롭다운 메뉴에서 보기(Show)를 클릭하여 찾을 수 있습니다. 또한 명령에 `-o ConnectTimeout=10`(을)를 추가합니다.
  - 명령은 다음 예시와 비슷해야 합니다(*<WebServerIP>*는 실제 웹 서버 퍼블릭 IP 주소). **`ssh -i labsuser.pem ec2-user@<WebServerIP> -o ConnectTimeout=10`**
- 10 초 후 SSH 를 통한 연결 시도가 실패합니다(*작업 시간 초과* 또는 *연결 시간 초과* 오류 발생). 이는 정상적인 동작입니다.

35. 이번에는 시간 제한을 15 초로 설정하여 다시 연결해봅니다.

    Windows 사용자를 위한 팁: PuTTY(inactive) 창에서 **PuTTY** 단어를 마우스 오른쪽 버튼으로 클릭하고 **설정 변경(Change Settings)**을 선택합니다. **연결(Connection)**(연결)을 클릭한 다음 *킵얼라이브 간 시간차(초)(Seconds between keepalives)*를 15 로 설정합니다. **Apply** 를 클릭하고 PuTTY 단어를 마우스 오른쪽 버튼으로 다시 클릭한 다음 **저장된 세션 > webserver(Saved Sessions > webserver)**를 선택합니다.

    연결이 다시 실패합니다. 이는 정상적인 동작입니다.

    조금 전에 실행한 **`describe-instances`** 명령의 출력 결과에 vockey 라는 이름의 인스턴스에 연결하기 위한 키 페어가 표시된 것을 확인합니다. 다운로드한 labsuser 키와 같은 키 페어(이름만 다름)로, 이를 통해 잘못된 키 페어로 인해 발생한 문제가 아님을 알 수 있습니다.

36. CLI 호스트 인스턴스에 대한 SSH 연결을 다시 설정합니다.
    - **macOS 및 Linux 사용자:** 키보드의 위쪽 화살표 키를 사용하여 명령을 찾은 후 다시 실행합니다. 명령은 다음과 비슷합니다. **`ssh -i labsuser.pem ec2-user@CliHostIP`**
    - **Windows 사용자:** PuTTY(inactive) 창에서 **PuTTY** 단어를 마우스 오른쪽 버튼으로 클릭하고 **저장된 세션 > cliHost(Saved Sessions > cliHost)**를 선택합니다. **로그인(login as)** 메시지가 표시되면 **`ec2-user`** 을(를) 입력합니다.

    CLI 호스트 인스턴스에 다시 정상적으로 연결될 것입니다.

문제 해결 도전 과제 1

웹 서버 인스턴스가 실행 중이라는 것을 확인했습니다. 하지만 웹 서버 인스턴스가 제공해야 하는 웹페이지를 로드할 수 없습니다. 또한 정확한 SSH 키 페어를 사용해도 SSH 를 사용하여 웹 서버 인스턴스에 연결할 수 없습니다. 어떤 문제로 인한 것일까요?

AWS Management Console 을 사용하지 않고 AWS CLI 프로그래밍 방식 액세스만 사용하여 조사해봅니다.

힌트:

- **힌트 1**: nmap 유틸리티를 사용하여 웹 서버 EC2 인스턴스에서 어떤 포트가 열려있는지 확인합니다. 이를 위해서는 **`sudo yum install -y nmap`** 을(를) 실행하여 CLI 호스트 인스턴스에 유틸리티를 설치해야 합니다. 그러고 나서 **`nmap <WebServerIP>`**을(를) 실행합니다. 이때 *<WebServerIP>*는 웹 서버 인스턴스의 실제 퍼블릭 IP 주소로, 지침 위에 있는 세부 정보(Details) 드롭다운 메뉴에서 보기(Show)를 클릭하여 찾을 수 있습니다. nmap 이 열려 있는 포트를 찾을 수 없다면 다른 것이 인스턴스에 대한 액세스를 막고 있을 수 있을까요?

```
[ec2-user@cli-host ~]$ nmap 54.218.96.251

Starting Nmap 6.40 ( http://nmap.org ) at 2023-08-03 13:42 UTC
Note: Host seems down. If it is really up, but blocking our ping probes, try -Pn
Nmap done: 1 IP address (0 hosts up) scanned in 3.03 seconds
```

- **힌트 2**: **`aws ec2 describe-security-groups`** 명령을 사용하여 보안 그룹 세부 정보를 확인합니다. 필요한 경우 AWS CLI 명령 참조 설명서를 참조합니다.
  - *group-ids* 파라미터를 사용하면 명령 결과를 더욱 쉽게 분석할 수 있습니다.
  - 지침 위에 있는 세부 정보(Details) 드롭다운 메뉴에서 보기(Show)를 클릭하면 웹 서버 인스턴스와 연결된 보안 그룹의 보안 그룹 ID 를 찾을 수 있습니다(WebServerSgId 로 표시됨). 또한 여러분이 실행한 **`describe-instances`** 명령은 보안 그룹 ID 도 반환합니다.
  - **`describe-security-groups`** 명령을 실행한 후 출력 결과를 분석합니다. 웹 서버 EC2 인스턴스에 적용된 보안 그룹 설정이 연결을 허용하는 것으로 보입니까?
- **힌트 3**: 웹 서버가 실행되는 서브넷과 연결된 라우팅 테이블의 설정을 확인합니다.
  - **`aws ec2 describe-route-tables`** 명령을 사용합니다.
  - 명령을 실행할 때 다음 예와 같은 필터를 적용하면 도움이 될 수 있습니다. **`--filter "Name=association.subnet-id,Values='<VPC1PubSubnetID>'"`**

*<VPC1PubSubnetID>*를 실제 서브넷 ID 값으로 바꿉니다. 이 ID 는 지침 위에 있는 세부 정보(Details) 드롭다운 메뉴에서 보기(Show)를 클릭하여 찾을 수 있습니다. **`describe-instances`** 명령을 실행하면 서브넷 ID 값도 반환됩니다.

- **힌트 4**: **`describe-route-tables`** 명령의 출력 결과를 분석할 때 서브넷에는 *public* 이라는 레이블이 지정되어 있음을 기억합니다.
    - 경로에서 문제를 확인할 수 있습니까? 새로운 경로를 정의해야 한다면 **`aws ec2 create-route`** 명령을 사용합니다. 자세한 내용은 AWS CLI 명령 참조 설명서를 참조합니다.
    - 경로 생성을 완료하려면 *route-table-id* 와 *gateway-id* 를 알아야 합니다. 두 값은 모두 지침 위에 있는 세부 정보(Details) 드롭다운 메뉴에서 보기(Show)를 클릭하여 찾을 수 있습니다. 앞서 **`describe-route-tables`** 명령을 실행할 때도 route-table-id 를 확인할 수 있었을 것입니다.
    - 원한다면 **`aws ec2 describe-internet-gateways`** 을(를) 사용하여 gateway-id 를 찾아도 됩니다. 명령을 정상적으로 실행하려면 다른 파라미터를 지정해야 할 수 있습니다. 어떤 파라미터가 필요한지는 참조 설명서에 나와 있습니다.

문제를 해결했다고 생각하면 웹 서버 페이지 로드를 시도했던 브라우저 탭으로 돌아가 페이지를 새로 고칩니다. http://WebServerIP 페이지에 *웹 서버 로드됨(Hello from your web server!)*라고 표시되어야 합니다. URL 끝에 **`/cafe/`**을(를) 붙이면 카페 웹사이트가 표시됩니다.

축하합니다. 웹사이트에 액세스할 수 없는 문제를 해결했습니다. 하지만 다른 문제가 아직 남아 있으며 다음 섹션에서 이 문제를 다루게 됩니다.

문제 해결 도전 과제 2

37. 이제 웹 액세스 문제를 해결했으니 웹 서버를 호스트하는 EC2 인스턴스에 SSH 를 사용하여 다시 연결해봅니다. 웹 서버 인스턴스에 연결하는 방법은 다음과 같습니다.

`exit` 을(를) 입력하여 현재 CLI 호스트에 대한 SSH 세션의 연결을 해제합니다.

- **사용** 중인 운영 체제에 맞는 단계를 완료합니다.
    - **Windows 사용자:** PuTTY(inactive) 창에서 **PuTTY** 단어를 마우스 오른쪽 버튼으로 클릭하고 **설정 변경(Change Settings)**을 선택합니다. **연결(Connection)**을 클릭한 다음 *킵얼라이브 간 시간차(초)(Seconds between keepalives)*를 10 으로 설정합니다. **Apply** 를 클릭하고 PuTTY 단어를 마우스 오른쪽 버튼으로 다시 클릭한 다음 **저장된 세션 > webserver(Saved Sessions > webserver)**를 선택합니다.

    - **macOS 및 Linux 사용자:** 키보드의 위쪽 화살표 키를 사용하여 이전에 사용했던 것과 동일한 SSH 연결 세부 정보를 로드합니다. 하지만 연결 세부 정보 끝부분의 퍼블릭 IP 주소를 *<WebServerIP>* 주소로 변경합니다. 주소는 지침 위에 있는 세부 정보(Details) 드롭다운 메뉴에서 보기(Show)를 클릭하여 찾을 수 있습니다. 또한 명령에 **-o ConnectTimeout=10**(을)를 추가합니다. 명령은 다음 예시와 비슷해야 합니다(*xxxxx-xxxxxx* 는 번호, *<WebServerIP>*는 실제 퍼블릭 WebServerIP 주소). **ssh -i labsuser.pem ec2-user@<WebServerIP> -o ConnectTimeout=10**
- 10 초 후 SSH 를 통한 연결 시도가 실패합니다(*작업 시간 초과* 오류 발생). 이 또한 정상적인 동작입니다.

어떤 문제가 남아 있을 수 있을까요?

웹 서버가 실행 중이라는 것은 이미 확인했습니다. 정확한 SSH 키 페어를 사용하여 연결했다는 사실도 알고 있습니다. 웹 서버 인스턴스가 실행되는 서브넷을 인터넷에 연결하기 위한 라우팅 테이블 항목을 정상적으로 생성했습니다. 또한 보안 그룹이 기본 SSH 포트인 포트 22 에서 연결을 허용한다는 것도 확인했습니다.

힌트:

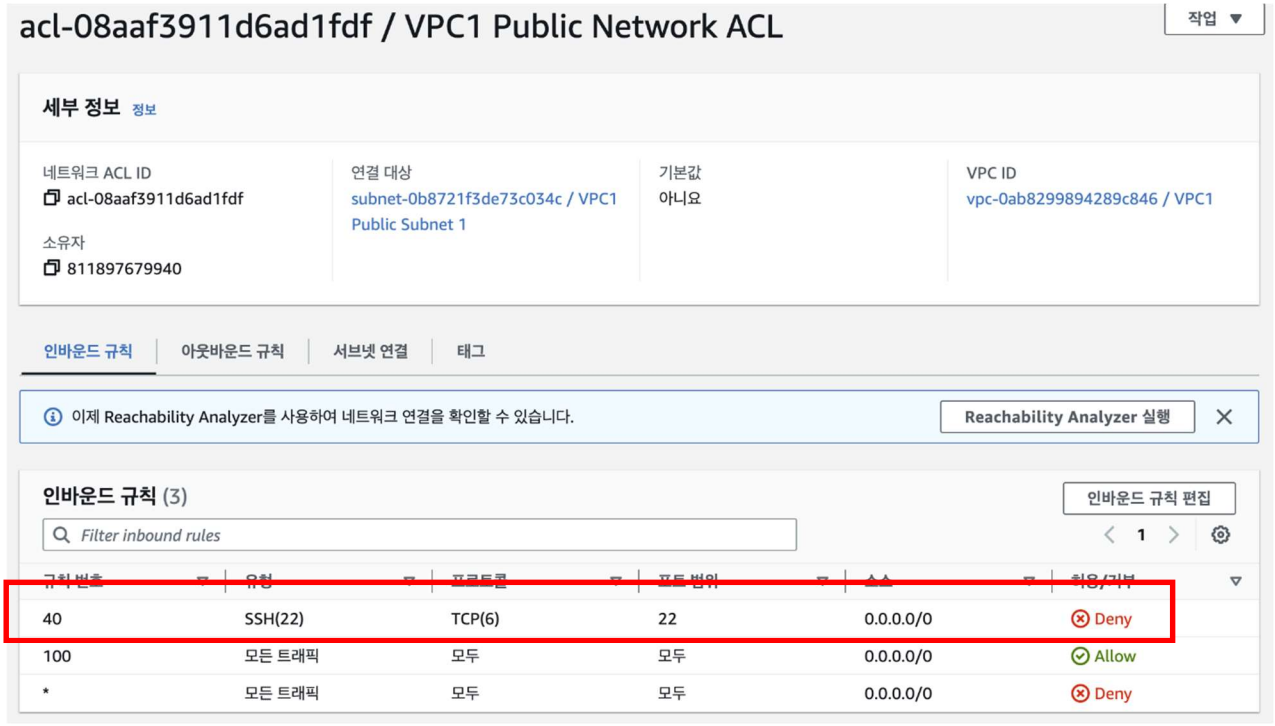
- CLI 호스트 인스턴스에 대한 SSH 연결을 다시 설정합니다.
- 인스턴스가 실행 중인 서브넷과 연결된 네트워크 ACL 의 액세스 제어 목록(네트워크 ACL) 설정을 확인합니다. 확인하려면 다음 명령을 실행합니다. 이때 *<VPC1PublicSubnetID>*를 지침 위에 있는 세부 정보(Details) 드롭다운 메뉴에서 보기(Show)를 클릭하여 찾을 수 있는 실제 서브넷 ID 로 바꿉니다.

```
aws ec2 describe-network-acls --filter "Name=association.subnet-id,Values='VPC1PublicSubnetID'" --query 'NetworkAcls[*].[NetworkAclId,Entries]'
```

```
[
    [
        "acl-0077af396b2556bd0",
        [
            {
                "RuleNumber": 100,
                "Protocol": "-1",
                "Egress": true,
                "CidrBlock": "0.0.0.0/0",
                "RuleAction": "allow"
            },
            {
                "RuleNumber": 32767,
                "Protocol": "-1",
                "Egress": true,
                "CidrBlock": "0.0.0.0/0",
                "RuleAction": "deny"
            },
            {
                "RuleNumber": 40,
                "Protocol": "6",
                "PortRange": {
                    "To": 22,
                    "From": 22
                },
                "Egress": false,
                "RuleAction": "deny",
                "CidrBlock": "0.0.0.0/0"
            },
            {
                "RuleNumber": 100,
                "Protocol": "-1",
                "Egress": false,
                "CidrBlock": "0.0.0.0/0",
                "RuleAction": "allow"
            },
            {
                "RuleNumber": 32767,
                "Protocol": "-1",
                "Egress": false,
                "CidrBlock": "0.0.0.0/0",
                "RuleAction": "deny"
            }
        ]
    ]
```


- 명령 실행 결과를 분석합니다. 문제를 일으킬 만한 항목이 있습니까?
- **`delete-network-acl-entry`** 명령을 사용하여 문제를 일으킬 수 있는 네트워크 ACL 항목을 모두 삭제합니다. 필요한 경우 AWS CLI 명령 참조 설명서를 참조합니다.

문제를 해결했다고 생각되면 컴퓨터에서 웹 서버 인스턴스에 대한 SSH 세션을 다시 설정하여 연결할 수 있는지 확인합니다. 연결할 수 있다면 문제가 해결된 것입니다. 정확한 EC2 인스턴스에 연결했음을 확인했다면 연결한 후 **`hostname`** 명령을 실행합니다. 호스트 이름이 *web-server* 로 표시되어야 합니다.



축하합니다. 웹 서버에 연결할 수 없게 하는 SSH 액세스 문제를 해결했습니다.

과제 4: 흐름 로그 분석

네트워크 문제를 해결했습니다. 이 과정에서 활동 초반에 VPC 흐름 로그를 사용 설정할 때 생성했던 몇 가지 흥미로운 항목이 흐름 로그에 생성되었습니다. 마지막 과제에서는 흐름 로그를 쿼리하여 흐름 로그에 캡처된 활동을 확인하겠습니다.

과제 4.1: 흐름 로그 다운로드 및 추출

38. CLI 호스트 EC2 인스턴스에 다시 연결합니다.
39. CLI 호스트에 흐름 로그 파일을 다운로드하기 위한 로컬 디렉터리를 만듭니다.

```
mkdir flowlogs
```

40. 디렉터리를 새 디렉터리로 변경합니다.

```
cd flowlogs
```

41. 버킷 이름을 다시 찾기 위해 S3 버킷을 나열합니다.

```
aws s3 ls
```

```
[ec2-user@cli-host ~]$ mkdir flowlogs
[ec2-user@cli-host ~]$ cd flowlogs/
[ec2-user@cli-host flowlogs]$ aws s3 ls
2023-08-04 01:03:36 flowlog1234
[ec2-user@cli-host flowlogs]$
```

42. 다음 명령을 실행하여 흐름 로그를 다운로드합니다. 이때 *<flowlog####>* 부분을 실제 버킷 이름으로 바꿉니다.

```
aws s3 cp s3://<flowlog####>/ . --recursive
```

명령이 정상적으로 실행되면 여러 파일이 다운로드된 것을 확인할 수 있습니다.

```
[ec2-user@cli-host flowlogs]$ ls
AWSLogs
[ec2-user@cli-host flowlogs]$
```

43. 필요한 경우 **cd** 명령과 **ls** 명령을 반복적으로 사용합니다. 또는 **cd** 명령을 사용한 후 TAB 키를 여러 번 누릅니다. 디렉터리를 변경하고 다운로드한 폴더 구조 안으로 이동해야 합니다. **ls** 을(를) 실행하면 다운로드된 로그 파일을 모두 볼 수 있습니다. 로그는 *AWSLogs/<account-num>/vpcflowlogs/<region>/yyyy/mm/dd* 하위 디렉터리에 있습니다.

파일 이름은 모두 log.gz 로 끝나는데, 이는 GNU zip 파일로 압축되었다는 의미입니다.

44. 이 명령을 실행하여 로그를 추출합니다.

```
gunzip *.gz
```

45. **ls** 을(를) 다시 실행합니다. 파일이 추출된 것을 확인할 수 있습니다.

```
[ec2-user@cli-host 811897679940]$ ls
vpcflowlogs
[ec2-user@cli-host 811897679940]$ cd vpcflowlogs/
[ec2-user@cli-host vpcflowlogs]$ ls
us-west-2
[ec2-user@cli-host vpcflowlogs]$ cd us-west-2/
[ec2-user@cli-host us-west-2]$ ls
2023
[ec2-user@cli-host us-west-2]$ cd 2023
[ec2-user@cli-host 2023]$ ls
08
[ec2-user@cli-host 2023]$ cd 08
[ec2-user@cli-host 08]$ ls
04
[ec2-user@cli-host 08]$ cd 04
[ec2-user@cli-host 04]$ ls
811897679940_vpcflowlogs_us-west-2_fl-062915a6e08bb16a7_20230804T0100Z_6d1128b2.log.gz
811897679940_vpcflowlogs_us-west-2_fl-062915a6e08bb16a7_20230804T0105Z_d202dfc4.log.gz
811897679940_vpcflowlogs_us-west-2_fl-062915a6e08bb16a7_20230804T0105Z_df958770.log.gz
811897679940_vpcflowlogs_us-west-2_fl-062915a6e08bb16a7_20230804T0110Z_7846fbec.log.gz
811897679940_vpcflowlogs_us-west-2_fl-062915a6e08bb16a7_20230804T0110Z_d612b26b.log.gz
811897679940_vpcflowlogs_us-west-2_fl-062915a6e08bb16a7_20230804T0115Z_6f1d79c9.log.gz
811897679940_vpcflowlogs_us-west-2_fl-062915a6e08bb16a7_20230804T0115Z_7dbda65f.log.gz
811897679940_vpcflowlogs_us-west-2_fl-062915a6e08bb16a7_20230804T0120Z_3c817e28.log.gz
811897679940_vpcflowlogs_us-west-2_fl-062915a6e08bb16a7_20230804T0120Z_7963f9a1.log.gz
811897679940_vpcflowlogs_us-west-2_fl-062915a6e08bb16a7_20230804T0125Z_9a28ceb8.log.gz
[ec2-user@cli-host 04]$
```

```
[ec2-user@cli-host 04]$ gunzip *.gz
[ec2-user@cli-host 04]$ ls
811897679940_vpcflowlogs_us-west-2_fl-062915a6e08bb16a7_20230804T0100Z_6d1128b2.log
811897679940_vpcflowlogs_us-west-2_fl-062915a6e08bb16a7_20230804T0105Z_d202dfc4.log
811897679940_vpcflowlogs_us-west-2_fl-062915a6e08bb16a7_20230804T0105Z_df958770.log
811897679940_vpcflowlogs_us-west-2_fl-062915a6e08bb16a7_20230804T0110Z_7846fbec.log
811897679940_vpcflowlogs_us-west-2_fl-062915a6e08bb16a7_20230804T0110Z_d612b26b.log
811897679940_vpcflowlogs_us-west-2_fl-062915a6e08bb16a7_20230804T0115Z_6f1d79c9.log
811897679940_vpcflowlogs_us-west-2_fl-062915a6e08bb16a7_20230804T0115Z_7dbda65f.log
811897679940_vpcflowlogs_us-west-2_fl-062915a6e08bb16a7_20230804T0120Z_3c817e28.log
811897679940_vpcflowlogs_us-west-2_fl-062915a6e08bb16a7_20230804T0120Z_7963f9a1.log
811897679940_vpcflowlogs_us-west-2_fl-062915a6e08bb16a7_20230804T0125Z_9a28ceb8.log
[ec2-user@cli-host 04]$
```

과제 4.2: 로그 분석

이 활동 섹션에서는 흐름 로그를 분석합니다. 구체적으로는 오류가 발생한 SSH 연결 시도가 로그에 캡처되었는지 확인합니다.

46. 로그의 구조를 분석합니다. 방법은 다음과 같습니다.
    - 실행한 **`ls`** 명령으로 반환된 파일 이름 중 하나를 복사합니다.
    - 터미널 창에 **`head`** 을(를) 입력하고 한 칸 띈 다음 복사한 파일 이름을 붙여넣습니다. 명령을 실행합니다.

- 헤더 행은 각 로그 항목에 포함된 데이터 종류를 나타냅니다. 각 항목에는 이벤트 소스의 IP 주소(4 번째 열), 대상 포트(7 번째 열), 시작 및 종료 타임스탬프(Unix 타임스탬프 형식), 결과로 나타난 작업(ACCEPT 또는 REJECT) 등의 정보가 포함되어 있습니다.

자세한 내용은 VPC 흐름 로그 레코드 설명서를 참조하십시오.

47. 현재 디렉터리에 있는 각 로그 파일을 확인하여 *REJECT*라는 단어가 포함된 행을 반환하는 **grep** 명령을 실행합니다.

```
grep -rn REJECT .
```

이 명령은 VPC 설정이 요청을 거부한 모든 이벤트를 포함하는 큰 데이터 집합을 반환합니다.

```
[ec2-user@cli-host 04]$ head 811897679940_vpcflowlogs_us-west-2_fl-062915a6e08bb16a7_20230804T0100Z_6d1
128b2.log
version account-id interface-id srcaddr dstaddr srcport dstport protocol packets bytes start end action
 log-status
2 811897679940 eni-0b5316cba9c003e46 35.203.210.129 10.0.1.130 50383 46662 6 1 44 1691111022 1691111043
 REJECT OK
2 811897679940 eni-0b5316cba9c003e46 162.216.149.245 10.0.1.130 54090 47342 6 1 44 1691111022 169111104
3 REJECT OK
2 811897679940 eni-0b5316cba9c003e46 167.94.146.30 10.0.1.130 38671 10051 6 1 44 1691111022 1691111043
REJECT OK
2 811897679940 eni-0b5316cba9c003e46 205.210.31.142 10.0.1.130 53925 8991 6 1 44 1691111022 1691111043
REJECT OK
2 811897679940 eni-0b5316cba9c003e46 205.210.31.153 10.0.1.130 54213 2080 6 1 44 1691111022 1691111043
REJECT OK
2 811897679940 eni-0b5316cba9c003e46 185.94.111.1 10.0.1.130 52628 1900 17 1 122 1691111022 1691111043
REJECT OK
2 811897679940 eni-0b5316cba9c003e46 35.203.211.207 10.0.1.130 52156 9541 6 1 44 1691111022 1691111043
REJECT OK
2 811897679940 eni-0b5316cba9c003e46 143.42.164.34 10.0.1.130 49114 2083 6 1 44 1691111022 1691111043 R
EJECT OK
2 811897679940 eni-0b5316cba9c003e46 114.239.112.178 10.0.1.130 62249 23 6 2 80 1691111050 1691111076 R
EJECT OK
```

48. 얼마나 많은 레코드가 반환되었는지 확인합니다.

```
grep -rn REJECT . | wc -l
```

결과에는 결과 집합에 행이 몇 개 있는지 표시됩니다.

```
[ec2-user@cli-host 04]$ grep -rn REJECT . | wc -l
584
[ec2-user@cli-host 04]$
```

49. 다음 방법으로 검색 범위를 좁혀 22 가 포함된 행만 찾습니다. 22 는 액세스가 차단되었을 때 웹 서버에 연결을 시도한 포트 번호입니다.

```
grep -rn ' 22 ' . | grep REJECT
```

이 명령을 사용하면 더 적은 수의 결과가 반환됩니다.

```
[ec2-user@cli-host 04]$ grep -rn ' 22 ' . | grep REJECT
./811897679940_vpcflowlogs_us-west-2_fl-062915a6e08bb16a7_20230804T0115Z_7dbda65f.log:39:2 811897679940
 eni-0b5316cba9c003e46 198.199.119.73 10.0.1.130 45109 22 6 1 40 1691111737 1691111763 REJECT OK
./811897679940_vpcflowlogs_us-west-2_fl-062915a6e08bb16a7_20230804T0120Z_3c817e28.log:46:2 811897679940
 eni-005a470ddc44cbb41 45.119.212.196 10.0.1.186 37504 22 6 1 60 1691112157 1691112180 REJECT OK
./811897679940_vpcflowlogs_us-west-2_fl-062915a6e08bb16a7_20230804T0120Z_3c817e28.log:80:2 811897679940
 eni-005a470ddc44cbb41 45.119.212.196 10.0.1.186 37504 22 6 1 60 1691112129 1691112157 REJECT OK
./811897679940_vpcflowlogs_us-west-2_fl-062915a6e08bb16a7_20230804T0120Z_7963f9a1.log:47:2 811897679940
 eni-0b5316cba9c003e46 186.236.228.182 10.0.1.130 10101 22 6 2 120 1691112100 1691112126 REJECT OK
./811897679940_vpcflowlogs_us-west-2_fl-062915a6e08bb16a7_20230804T0125Z_9a28ceb8.log:19:2 811897679940
 eni-0b5316cba9c003e46 85.241.50.58 10.0.1.130 41798 22 6 4 240 1691112311 1691112335 REJECT OK
./811897679940_vpcflowlogs_us-west-2_fl-062915a6e08bb16a7_20230804T0105Z_d202dfc4.log:65:2 811897679940
 eni-005a470ddc44cbb41 182.208.131.42 10.0.1.186 58688 22 6 9 576 1691111230 1691111256 REJECT OK
./811897679940_vpcflowlogs_us-west-2_fl-062915a6e08bb16a7_20230804T0105Z_df958770.log:21:2 811897679940
 eni-005a470ddc44cbb41 182.208.131.42 10.0.1.186 58692 22 6 9 576 1691111257 1691111283 REJECT OK
./811897679940_vpcflowlogs_us-west-2_fl-062915a6e08bb16a7_20230804T0105Z_df958770.log:24:2 811897679940
 eni-0b5316cba9c003e46 65.49.1.45 10.0.1.130 42488 22 6 1 40 1691111299 1691111317 REJECT OK
./811897679940_vpcflowlogs_us-west-2_fl-062915a6e08bb16a7_20230804T0110Z_d612b26b.log:63:2 811897679940
 eni-0b5316cba9c003e46 176.113.115.210 10.0.1.130 64001 22 6 1 40 1691111650 1691111676 REJECT OK
./811897679940_vpcflowlogs_us-west-2_fl-062915a6e08bb16a7_20230804T0115Z_6f1d79c9.log:2:2 811897679940
eni-005a470ddc44cbb41 192.241.235.20 10.0.1.186 48290 22 6 1 40 1691111775 1691111794 REJECT OK
[ec2-user@cli-host 04]$
```

오류가 발생한 SSH 연결에 해당하는 로그 항목만 표시되도록 결과 집합을 가려내려면 결과를 더욱 세부적으로 필터링해야 합니다.

웹 서버에 SSH 를 사용하여 연결하려는 시도가 실패한 것은 로컬 머신에서 이루어졌음을 기억합니다. 다음 단계에서는 인터넷을 통해 로컬 머신에 접속할 수 있는 IP 주소를 확인합니다.

50. 인터넷을 통해 로컬 머신에 접속할 수 있는 IP 주소를 확인합니다.

    방법은 다음과 같습니다.

    - AWS Management Console 에 로그인합니다.
    - EC2 인스턴스가 실행 중인 리전의 **EC2** 서비스로 이동합니다.
    - **Security Groups** 를 클릭합니다.
    - **WebSecurityGroup** 을 클릭한 다음 **Inbound** 탭을 클릭합니다.
    - **Edit** 을 클릭한 다음 **Add Rule** 을 클릭합니다.
    - 방금 만들어진 3 번째 행에서 **Source** 에 **My IP** 를 선택합니다.
    - 자동으로 입력된 클래스 없는 도메인 간 라우팅(CIDR) 블록의 IP 주소(*/32* 로 끝남)를 복사합니다.

      /32 접미사를 제외한 IP 주소를 복사합니다.

    - 그런 다음 **Cancel** 을 클릭합니다. 이 계정의 보안 그룹을 수정하지 않아도 됩니다. 이 단계의 목적은 IP 주소를 확인하는 것입니다.

51. CLI Host SSH 터미널 세션으로 돌아가 흐름 로그에 대한 쿼리를 더욱 좁은 범위로 실행합니다. 이때 *<ip-address>*를 복사한 IP 주소로 바꿉니다.

```
grep -rn ' 22 ' . | grep REJECT | grep <ip-address>
```

이제 결과 집합의 행 수가 SSH 를 사용하여 웹 서버 인스턴스에 연결하려고 시도했다 실패한 횟수와 일치합니다.

쿼리를 통해 반환된 각 로그 항목에 탄력적 네트워크 인터페이스 ID 가 있음을 확인합니다.

```
[ec2-user@cli-host 04]$ grep -rn ' 22 ' . | grep REJECT | grep '182.208.131.42'
./811897679940_vpcflowlogs_us-west-2_fl-062915a6e08bb16a7_20230804T0105Z_d202dfc4.log:65:2 811897679940
 eni-005a470ddc44cbb41 182.208.131.42 10.0.1.186 58688 22 6 9 576 1691111230 1691111256 REJECT OK
./811897679940_vpcflowlogs_us-west-2_fl-062915a6e08bb16a7_20230804T0105Z_df958770.log:21:2 811897679940
 eni-005a470ddc44cbb41 182.208.131.42 10.0.1.186 58692 22 6 9 576 1691111257 1691111283 REJECT OK
[ec2-user@cli-host 04]$
```

52. 다음 명령을 실행하고 *<WebServerIP>*를 실제 IP 주소로 바꿉니다.

```
aws ec2 describe-network-interfaces --filters
"Name=association.public-ip,Values='<WebServerIP>'" --query
'NetworkInterfaces[*].[NetworkInterfaceId,Association.PublicIp]'
```

흐름 로그에 기록된 네트워크 인터페이스 ID 가 네트워크 인터페이스의 일부로 웹 서버 인스턴스에 할당된 네트워크 인터페이스와 일치한다는 것을 결과 집합을 통해 알 수 있습니다.

```
[ec2-user@cli-host 04]$ aws ec2 describe-network-interfaces --filters "Name=association.public-ip,Value
s='35.164.169.237'" --query 'NetworkInterfaces[*].[NetworkInterfaceId,Association.PublicIp]'
[
    [
        "eni-005a470ddc44cbb41",
        "35.164.169.237"
    ]
]
[ec2-user@cli-host 04]$
```

53. 타임스탬프를 사람이 읽을 수 있는 형식으로 변환합니다.

각 로그 항목 끝부분에 REJECT 라는 용어 앞에 있는 긴 번호 2 개를 확인합니다.

이 번호는 Unix 형식의 타임스탬프입니다. 1 번째 타임스탬프는 캡처된 각 이벤트의 시작 시각을 나타냅니다. 2 번째 타임스탬프는 종료 시각을 나타냅니다. Linux **date** 명령줄 유틸리티를 사용하여 이러한 타임스탬프를 사람이 읽을 수 있는 형식으로 변환할 수 있습니다. 예를 들어 타임스탬프가 *1554496931* 이라면 다음 명령을 실행합니다.

```
date -d @1554496931
```

필터링된 REJECT 결과에 캡처된 타임스탬프중 하나로 **date -d @** 명령을 실행합니다. 여러분이 이 활동을 진행한 시간에 상응하는 오늘의 시간이 표시됩니다. **date** 명령을 실행하여 현재 시각과 결과를 확인합니다.

```
[ec2-user@cli-host 04]$ grep -rn ' 22 ' . | grep REJECT
./811897679940_vpcflowlogs_us-west-2_fl-062915a6e08bb16a7_20230804T0115Z_7dbda65f.log:39:2 811897679940
 eni-0b5316cba9c003e46 198.199.119.73 10.0.1.130 45109 22 6 1 40 1691111737 1691111763 REJECT OK
./811897679940_vpcflowlogs_us-west-2_fl-062915a6e08bb16a7_20230804T0120Z_3c817e28.log:46:2 811897679940
 eni-005a470ddc44cbb41 45.119.212.196 10.0.1.186 37504 22 6 1 60 1691112157 1691112180 REJECT OK
./811897679940_vpcflowlogs_us-west-2_fl-062915a6e08bb16a7_20230804T0120Z_3c817e28.log:80:2 811897679940
 eni-005a470ddc44cbb41 45.119.212.196 10.0.1.186 37504 22 6 1 60 1691112129 1691112157 REJECT OK
./811897679940_vpcflowlogs_us-west-2_fl-062915a6e08bb16a7_20230804T0120Z_7963f9a1.log:47:2 811897679940
 eni-0b5316cba9c003e46 186.236.228.182 10.0.1.130 10101 22 6 2 120 1691112100 1691112126 REJECT OK
./811897679940_vpcflowlogs_us-west-2_fl-062915a6e08bb16a7_20230804T0125Z_9a28ceb8.log:19:2 811897679940
 eni-0b5316cba9c003e46 85.241.50.58 10.0.1.130 41798 22 6 4 240 1691112311 1691112335 REJECT OK
./811897679940_vpcflowlogs_us-west-2_fl-062915a6e08bb16a7_20230804T0105Z_d202dfc4.log:65:2 811897679940
 eni-005a470ddc44cbb41 182.208.131.42 10.0.1.186 58688 22 6 9 576 1691111230 1691111256 REJECT OK
./811897679940_vpcflowlogs_us-west-2_fl-062915a6e08bb16a7_20230804T0105Z_df958770.log:21:2 811897679940
 eni-005a470ddc44cbb41 182.208.131.42 10.0.1.186 58692 22 6 9 576 1691111257 1691111283 REJECT OK
./811897679940_vpcflowlogs_us-west-2_fl-062915a6e08bb16a7_20230804T0105Z_df958770.log:24:2 811897679940
 eni-0b5316cba9c003e46 65.49.1.45 10.0.1.130 42488 22 6 1 40 1691111299 1691111317 REJECT OK
./811897679940_vpcflowlogs_us-west-2_fl-062915a6e08bb16a7_20230804T0110Z_d612b26b.log:63:2 811897679940
 eni-0b5316cba9c003e46 176.113.115.210 10.0.1.130 64001 22 6 1 40 1691111650 1691111676 REJECT OK
./811897679940_vpcflowlogs_us-west-2_fl-062915a6e08bb16a7_20230804T0115Z_6f1d79c9.log:2:2 811897679940
eni-005a470ddc44cbb41 192.241.235.20 10.0.1.186 48290 22 6 1 40 1691111775 1691111794 REJECT OK
[ec2-user@cli-host 04]$
```

grep 은 VPC 흐름 로그 파일에서 유의미한 데이터를 추출하는 강력하고도 간단한 방법입니다. 로그를 통해 보고서를 실행하거나 분석 대시보드를 생성하는 도구는 시중에 많이 나와 있습니다. 그중 한 솔루션은 Amazon Athena 서비스입니다. Amazon Athena 를 사용하면 로그를 수집하여 데이터베이스의 데이터로 만듭니다. 그런 다음 정형 쿼리 언어(SQL) 쿼리를 실행하여 로그에서 유의미한 정보를 추출할 수 있습니다. Amazon Athena 에 대한 자세한 내용은 여기에서 확인합니다.

활동 완료

축하합니다. 활동을 마쳤습니다.

54. 이 페이지의 상단에서 실습 종료(End Lab)를 클릭하고 예(Yes)를 클릭하여 활동 종료를 확인합니다.

    "삭제가 시작되었습니다.("DELETE has been initiated...) 이제 이 메시지 상자를 닫아도 됩니다.'(You may close this message box now.')라는 내용의 패널이 표시됩니다.

55. 오른쪽 상단 모서리에 있는 **X** 를 클릭하여 패널을 닫습니다.

추가 리소스

AWS Training and Certification 에 대한 자세한 내용은 https://aws.amazon.com/training/을 참조하십시오.

여러분의 피드백을 환영합니다. 제안이나 수정 사항을 공유하려면 AWS Training and Certification 연락처 양식에서 세부 정보를 제공해 주십시오.