

# 173- [JAWS] - 활동 - 인스턴스 생성 문제 해결

## 활동 - AWS CLI를 사용한 EC2 인스턴스 생성 관련 문제 해결

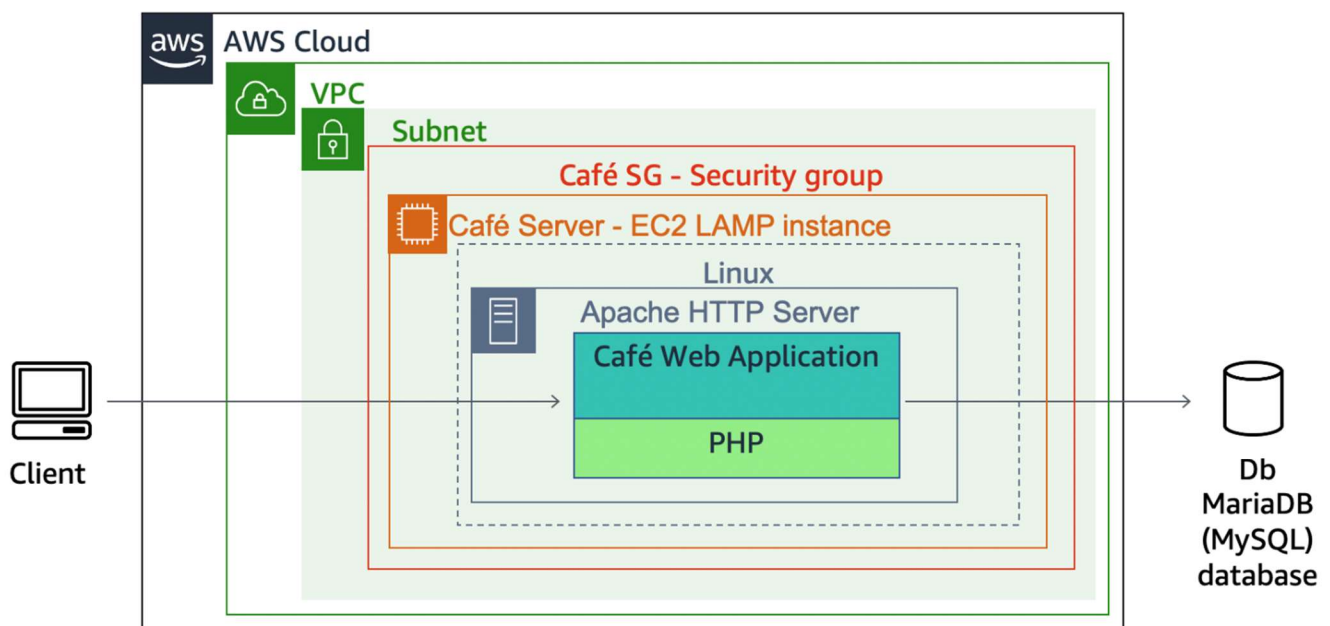
### 활동 개요

이 활동에서는 AWS Command Line Interface (AWS CLI)를 사용하여 Amazon Elastic Compute Cloud (Amazon EC2) 인스턴스를 시작합니다.

인스턴스를 생성할 때 참조하는 *user-data* 파일은 인스턴스가 Apache 웹 서버, MariaDB 관계형 데이터베이스 (MySQL 관계형 데이터베이스의 한 갈래), PHP 를 실행하도록 구성합니다. 이러한 소프트웨어 패키지가 한 컴퓨터에 설치된 것을 LAMP 스택 (**L**inux, **A**pache 웹 서버, **M**ySQL, **P**HP)이라고 합니다. LAMP 스택은 데이터베이스 백엔드 배포를 포함한 웹사이트를 하나의 컴퓨터에서 생성하는 데 흔히 사용됩니다.

동일한 *user-data* 파일이 인스턴스에서 웹사이트 파일을 배포하고 SQL 스크립트를 실행합니다. 그 결과 인스턴스가 업데이트된 버전의 카페 웹사이트를 호스트하게 됩니다. 새 버전의 웹사이트는 고객의 온라인 주문을 지원합니다.

이 다이어그램에는 이번 활동에서 완료하게 될 작업이 요약되어 있습니다.



## 활동 목표

이 활동을 완료하면 다음을 할 수 있게 됩니다.

- AWS CLI 를 사용하여 Amazon EC2 인스턴스를 **시작**합니다.
- AWS CLI 명령 및 Amazon EC2 서비스 설정에 대한 **문제를 해결**합니다.

## 비즈니스 사례 관련성

카페의 새로운 비즈니스 요구 사항 - 온라인 주문



첫 번째 버전의 카페 웹사이트가 게시되었을 때 Martha 와 Frank 는 Nikhil 과 Sofia 의 작업 결과물에 매우 만족했습니다. 고객들 역시 드디어 카페에 대한 정보를 온라인에서 찾을 수 있게 되어 기뻐했습니다. 하지만 카페를 찾는 고객 중 다수가 온라인으로 주문하고 싶다는 의견을 전했습니다.

클라우드 운영 전문가의 일은 끝이 없나봅니다. 이번 활동에서는 Nikhil 과 Sofia 의 역할을 다시 맡아 온라인 주문을 받을 수 있도록 웹사이트를 업그레이드합니다.

## 활동 단계

**소요 시간:** 이 활동을 완료하는 데는 약 **45 분**이 소요됩니다.

# AWS Management Console 액세스

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

실습시작 (Start Lab) 패널이 열리고 실습 상태가 표시됩니다.

2. **"실습 상태: 준비 (Lab status: ready)"** 메시지가 표시되면 **X**를 클릭하여 실습 시작(Start Lab) 패널을 닫습니다.
3. 지침의 맨 위에서 **AWS**를 클릭합니다.

그러면 새 브라우저 탭에서 AWS Management Console 이 열립니다. 자동으로 로그인됩니다.

**팁:** 새 브라우저 탭이 열리지 않는 경우 일반적으로 브라우저에서 팝업 창을 열 수 없음을 나타내는 배너 또는 아이콘이 브라우저 상단에 표시됩니다. 배너 또는 아이콘을 클릭하고 "팝업 허용(Allow pop ups)"를 선택합니다.

4. 이 지침과 함께 표시되도록 AWS Management Console 탭을 정렬합니다. 두 브라우저 탭이 동시에 표시되어 실습 단계를 보다 쉽게 수행할 수 있게 됩니다.

이 브라우저 탭을 열어 두십시오. 활동 뒷부분에서 이 탭으로 다시 돌아오게 됩니다.

## 과제 1: SSH 를 사용하여 Amazon Linux EC2 인스턴스에 연결

이번 과제에서는 AWS CLI 가 이미 설치된 기존 Amazon Linux EC2 인스턴스에 연결합니다.

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.](#)

### 과제 1.1: Windows SSH

#### Windows 사용자: SSH 를 사용하여 연결

이 지침은 Windows 사용자에게만 적용됩니다.

macOS 또는 Linux 를 사용하는 경우 [다음 섹션으로 건너뛰십시오.](#)

- 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.

- 필요한 소프트웨어를 다운로드합니다.

- PuTTY**를 사용하여 SSH를 통해 Amazon EC2 인스턴스에 연결합니다. 컴퓨터에 PuTTY가 설치되어 있지 않은 경우 [여기에서 다운로드](#)하십시오.

- putty.exe**를 엽니다.

- 시간 초과가 발생하지 않도록 다음과 같이 PuTTY를 구성합니다.

- 연결 (Connection)**을 클릭합니다.

- 킵얼라이브 간 시간차(초) (Seconds between keepalives)**를 30으로 설정합니다.

이렇게 하면 PuTTY 세션을 더 오래 열어둘 수 있습니다.

- 다음과 같이 PuTTY 세션을 구성합니다.

- 세션 (Session)**을 클릭합니다.

- 호스트 이름 또는 IP 주소 (Host Name or IP address)**: 인스턴스의 **IPv4 퍼블릭 IP 주소(IPv4 Public IP address)**를 복사하여 붙여넣습니다. 이 주소를 찾으려면 EC2 Console로 돌아가서 **인스턴스 (Instances)**를 클릭합니다. 연결하려는 인스턴스 옆의 확인란을 선택하고 **설명(Description)** 탭에서 **IPv4 퍼블릭 IP (IPv4 Public IP)** 값을 복사합니다.

- PuTTY로 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.

- 호스트를 신뢰하고 호스트에 연결하려면 **예 (Yes)**를 클릭합니다.

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

그러면 EC2 인스턴스에 연결됩니다.

12. [Windows 사용자: 다음 과제로 건너뛰려면 여기를 클릭하십시오.](#)

## 과제 1.2: macOS/Linux SSH

이 지침은 Mac/Linux 사용자에게만 적용됩니다. Windows 사용자인 경우 [다음 과제로 건너뛰니다.](#)

13. 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.
- X를 클릭하여 세부 정보 (Details) 패널을 닫습니다.

14. 터미널 창을 열고 디렉토리를 labsuser.pem 파일이 다운로드된 디렉터리로 변경(cd)합니다.

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

```
cd ~/Downloads
```

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

```
chmod 400 labsuser.pem
```

16. AWS Management Console 로 돌아간 후 EC2 서비스에서 인스턴스 (Instances)를 클릭합니다. 연결하려는 인스턴스 옆의 확인란을 선택합니다.

17. 설명 (Description) 탭에서 IPv4 퍼블릭 IP (IPv4 Public IP) 값을 복사합니다.

18. 터미널 창으로 돌아가서 다음 명령을 실행합니다(<public-ip>를 복사한 실제 퍼블릭 IP 주소로 바꿈).

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

19. 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 2: AWS CLI 구성

참고: AWS 를 통해 이용 가능한 다른 일부 Linux 배포와는 달리 Amazon Linux 인스턴스에는 AWS CLI 가 사전 설치되어 있습니다.

20. 보안 인증을 사용하여 AWS CLI 소프트웨어를 업데이트합니다.

```
aws configure
```

21. 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):** 지침 상단에 있는 세부 정보 (Details) 드롭다운 메뉴를 클릭한 다음 보기 (Show)를 클릭합니다. **LabRegion** 값을 복사하여 터미널 창에 붙여넣습니다.
- **기본 출력 형식 (Default output format):** json

## 과제 3: AWS CLI 를 사용하여 EC2 인스턴스 생성

### 과제 3.1: 스크립트 세부 정보 확인

22. 편집할 스크립트 파일이 있는 디렉터리로 이동하고 다음 명령을 실행하여 파일의 백업을 생성합니다.

```
cd ~/sysops-activity-files/starters
cp create-lamp-instance-v2.sh create-lamp-instance.backup
```

팁: 파일을 수정하기 전에 백업하는 것은 좋은 습관입니다.

23. 즐겨 쓰는 명령줄 텍스트 에디터 (예: VI)에서 create-lamp-instance-v2.sh 스크립트 파일을 엽니다.

vi create-lamp-instance-v2.sh

24. 스크립트의 내용을 분석합니다.

**팁:** VI 를 사용 중이라면 `:set number` 를 입력한 후 **ENTER** 키를 눌러 행 번호를 표시할 수 있습니다.

**1 행:**

- 이 파일은 Bash 파일이므로 첫 행에 `#!/bin/bash` 가 포함되어 있습니다.

**7~11 행:**

- 인스턴스 크기가 `t3.small`로 설정되어 있으며, 이는 데이터베이스와 웹 서버를 실행하기에 충분한 크기입니다.

```
7 # Hard coded values
8 instanceType="t3.small"
9 echo "Instance Type: $instanceType"
10 profile="default"
11 echo "Profile: $profile"
```

**16~29 행:**

- 이 스크립트는 모든 AWS 리전 목록을 불러오기 위해 AWS CLI `describe-regions` 명령을 호출합니다. 또한 각 리전에서 이름이 Cafe VPC 인 기존 VPC 를 쿼리합니다. VPC 를 찾으면 카페 LAMP 인스턴스가 배포되어야 하는 **vpc** ID 와 **리전**을 캡처하고 while 루프에서 벗어납니다.

```
16 # get vpcid
17 vpc=""
18 while [[ "$vpc" == "" ]]; do
19   for i in $(aws ec2 describe-regions | grep RegionName | cut -d '"' -f4) ; do
20     region=$i;
21     vpc=$(aws ec2 describe-vpcs --region $i --filters "Name=tag:Name,Values='Cafe VPC'" --profile $profile | grep VpcId | cut -d '"' -f4 | sed -n 1p);
22     if [[ "$vpc" != "" ]]; then
23       break;
24     fi
25   done
26 done
27 echo
28 echo "VPC: $vpc"
29 echo "Region: $region"
```

**31~57 행:**

- 스크립트가 AWS CLI 명령을 호출하여 EC2 인스턴스를 생성하는 데 필요한 **서브넷 ID**, **키 페어 이름**, **AMI ID** 값을 조회합니다.

- 32 행이 백슬래시 (\)로 끝나는 것을 확인합니다. 이 문자를 사용하면 명령 1 개를 스크립트 파일에 있는 다른 행에 래핑할 수 있습니다. 이 스크립트를 읽기 쉽게 하기 위해 이 기법이 여러 번 사용되었습니다.

```
31 vpc=$(aws ec2 describe-vpcs \
32 --filters "Name=tag:Name,Values='Cafe VPC'" \
33 --region $region \
34 --profile $profile | grep VpcId | cut -d '"' -f4 | sed -n 1p)
35 echo "VPC: "$vpc
36
37 # get subnetId
38 subnetId=$(aws ec2 describe-subnets \
39 --filters "Name=tag:Name,Values='Cafe Public Subnet 1'" \
40 --region $region \
41 --profile $profile \
42 --query "Subnets[*]" | grep SubnetId | cut -d '"' -f4 | sed -n 1p)
43 echo "Subnet Id: "$subnetId
44
45 # Get keypair name
46 key=$(aws ec2 describe-key-pairs \
47 --profile $profile --region $region | grep KeyName | cut -d '"' -f4 )
48 echo "Key: "$key
49
50 # Get AMI ID
51 imageld=$(aws ssm get-parameters \
52 --names '/aws/service/ami-amazon-linux-latest/amzn2-ami-hvm-x86_64-gp2' \
53 --profile $profile \
54 --region $region | grep ami- | cut -d '"' -f4 | sed -n 2p)
55 echo "AMI ID: "$imageld
```

#### 59~124 행:

- 스크립트의 이 부분은 이전에 AWS 계정에서 스크립트가 이미 실행되었는데 이번에 또 다시 실행되는 경우 AWS 계정을 정리합니다. 스크립트는 이름이 *cafeserver* 인 인스턴스와 이름에 *cafeSG*가 포함된 보안 그룹이 이미 존재하는지 확인합니다. 두 리소스 중 하나가 발견되면 삭제하라는 메시지가 표시됩니다.



```

57 #check for existing cafe instance
58 existingEc2Instance=$(aws ec2 describe-instances W
59 --region $region W
60 --profile $profile W
61 --filters "Name=tag:Name,Values=cafeserver" "Name=instance-state-name,Values=running" W
62 | grep InstanceId | cut -d '"' -f4)
63 if [[ "$existingEc2Instance" != "" ]]; then
64     echo
65     echo "WARNING: Found existing running EC2 instance with instance ID "$existingEc2Instance"."
66     echo "This script will not succeed if it already exists. "
67     echo "Would you like to delete it? [Y/N]"
68     echo ">"
69
70     validResp=0
71     while [ $validResp -eq 0 ];
72     do
73         read answer
74         if [[ "$answer" == "Y" || "$answer" == "y" ]]; then
75             echo
76             echo "Deleting the existing instance..."
77             aws ec2 terminate-instances --instance-ids $existingEc2Instance --region $region --profile $profile
78             #wait for confirmation it was terminated
79             aws ec2 wait instance-terminated --instance-ids $existingEc2Instance --region $region --profile $profile
80             validResp="1"
81         elif [[ "$answer" == "N" || "$answer" == "n" ]]; then
82             echo "Ok, exiting."
83             exit 1
84         else
85             echo "Please reply with Y or N."
86         fi
87     done
88
89     sleep 10 #give it 10 seconds before trying to delete the SG this instance used.
90 fi
91

```

```

92 #check for existing cafeSG security Group
93 existingMpSg=$(aws ec2 describe-security-groups \W
94 --region $region \W
95 --query "SecurityGroups[?contains(GroupName, 'cafeSG')]" \W
96 --profile $profile | grep GroupId | cut -d '"' -f4)
97
98 if [[ "$existingMpSg" != "" ]]; then
99     echo
100     echo "WARNING: Found existing security group with name \"$existingMpSg\"."
101     echo "This script will not succeed if it already exists. "
102     echo "Would you like to delete it? [Y/N]"
103     echo ">>"
104
105     validResp=0
106     while [ $validResp -eq 0 ];
107     do
108         read answer
109         if [[ "$answer" == "Y" || "$answer" == "y" ]]; then
110             echo
111             echo "Deleting the existing security group..."
112             aws ec2 delete-security-group --group-id $existingMpSg --region $region --profile $profile
113             validResp="1"
114         elif [[ "$answer" == "N" || "$answer" == "n" ]]; then
115             echo "Ok, exiting."
116             exit 1
117         else
118             echo "Please reply with Y or N."
119         fi
120     done
121     sleep 10 #give it 10 seconds before trying to recreate the SG
122 fi
123

```

126~154 행:

- 스크립트가 포트 22 및 80 이 열린 새 보안 그룹을 생성합니다.

```

124 # CREATE a security group and capture the name of it
125 echo
126 echo "Creating a new security group..."
127 securityGroup=$(aws ec2 create-security-group --group-name "cafeSG" \
128 --description "cafeSG" \
129 --region $region \
130 --group-name "cafeSG" \
131 --vpc-id $vpc --profile $profile | grep GroupId | cut -d '"' -f4 )
132 echo "Security Group: "$securityGroup
133
134 # Open ports in the security group
135 echo
136 echo "Opening port 22 in the new security group"
137 aws ec2 authorize-security-group-ingress \
138 --group-id $securityGroup \
139 --protocol tcp \
140 --port 22 \
141 --cidr 0.0.0.0/0 \
142 --region $region \
143 --profile $profile
144
145 echo "Opening port 80 in the new security group"
146 aws ec2 authorize-security-group-ingress \
147 --group-id $securityGroup \
148 --protocol tcp \
149 --port 8080 \
150 --cidr 0.0.0.0/0 \
151 --region $region \
152 --profile $profile

```

#### 156~168 행:

- 스크립트가 새로운 EC2 인스턴스를 생성합니다.
- 8 행과 10 행에 설정된 값과 16~57 행에 수집된 값이 이 AWS CLI 호출에서 어떻게 사용되는지 확인합니다.
- 또한 user-data 스크립트에 대한 참조도 확인합니다.

**참고:** 잠시 후 user-data 파일의 세부 정보를 확인하라는 메시지가 표시됩니다.

25. 아직 Bash 스크립트를 열고 있을 때 **157~168 행**에서 다시 확인합니다.

- 인스턴스를 생성하기 위한 전체 호출이 **instanceDetails** 라는 이름의 변수에 캡처됩니다. 그런 다음 이 변수의 내용이 177 행에서 터미널에 전달되고 Python JavaScript Object Notation (JSON)을 통해 읽기 쉽도록 형식이 지정됩니다.

```

154 echo
155 echo "Creating an EC2 instance in "$region
156 instanceDetails=$(aws ec2 run-instances W
157 --image-id $imageId W
158 --count 1 W
159 --instance-type $instanceType W
160 --region us-east-1 W
161 --subnet-id $subnetId W
162 --security-group-ids $securityGroup W
163 --tag-specifications 'ResourceType=instance,Tags=[{Key=Name,Value=cafeserver}]' W
164 --associate-public-ip-address W
165 --iam-instance-profile Name=LabInstanceProfile W
166 --profile $profile W
167 --user-data file://create-lamp-instance-userdata-v2.txt W
168 --key-name $key )

```

179~188 행:

- **instancetype** 가 instanceDetails 에서 파싱된 다음 while 루프가 인스턴스에 퍼블릭 IP 주소가 할당되었는지 10 초마다 확인합니다. 확인에 성공하면 퍼블릭 IP 주소가 터미널에 기록됩니다.

```

175 echo
176 echo "Instance Details...."
177 echo $instanceDetails | python -m json.tool
178
179 # Extract instancetype
180 instancetype=$(echo $instanceDetails | python -m json.tool | grep InstanceType | sed -n 1p | cut -d '"' -f4)
181 echo "instancetype="$instancetype
182 echo
183 echo "Waiting for a public IP for the new instance..."
184 publp=""
185 while [[ "$publp" == "" ]]; do
186     sleep 10;
187     publp=$(aws ec2 describe-instances --instance-id $instancetype --region $region --profile $profile | grep PublicIp | sed -n 1p | cut -d '"' -f4)
188 done

```

26. 파일을 변경하지 않은 채로 텍스트 편집기를 종료합니다 (VI 를 사용 중이라면 :q! 명령 입력).
27. 다음 명령을 실행하여 user-data 스크립트의 내용을 표시합니다.

```
vi create-lamp-instance-userdata-v2.txt
```

User-Data 스크립트 4 번째 행의 **yum -y install httpd mariadb-server** 뒤에 **wget** 을 추가하여 다음과 같이 변경합니다. (변경 후 :wq 를 사용하여 저장, 종료 합니다.)

```
yum -y install httpd mariadb-server wget
```

user-data 스크립트가 인스턴스 시작 이후 어떻게 인스턴스에서 일련의 명령을 실행하는지 확인합니다. 이러한 명령으로 웹 서버, PHP, 데이터베이스 서버가 설치됩니다.

## 과제 3.2: 스크립트 실행해보기

28. 셸 스크립트가 어떻게 작동하는지 알았으니 다음을 실행해봅니다.

```
./create-lamp-instance-v2.sh
```

스크립트 실행이 실패하고 완료되지 않은 상태로 종료됩니다. 이는 정상적인 동작입니다.

스크립트에 의도적으로 문제를 남겨 두었기 때문입니다. 여러분의 도전 과제는 문제를 찾아 해결하는 것입니다. 변경 사항을 적용한 후 언제든지 스크립트를 다시 실행하여 문제가 해결되었는지 확인해도 됩니다.

## 과제 3.3 문제 해결

### 문제 1

터미널 출력 결과에 따르면 `run-instances`에 대한 호출이 실패하고 *"RunInstances 작업을 불러오던 중 예러가 발생했습니다 (InvalidAMIID.NotFound): 이미지 ID '[ami-xxxxxxxxxx]'가 존재하지 않습니다."*와 같은 메시지가 표시됩니다.

```
Creating an EC2 instance in us-west-2
An error occurred (InvalidAMIID.NotFound) when calling the RunInstances operation: The image id '[ami-0044a0897b53acfb6]' does not exist
```

문제 1을 해결하기 위한 팁:

- Bash 스크립트에서 이 오류의 원인이 있는 행을 찾습니다. 스크립트에서 잘못된 것으로 보이는 부분이 있습니까?
- `run-instances` 명령에 사용된 리전 값이 정확합니까?
- 문제를 찾아 스크립트를 올바르게 수정했다고 생각된다면 스크립트를 다시 실행합니다. 마지막으로 스크립트를 실행했을 때 생성된 기존 보안 그룹 또는 인스턴스를 삭제할 것인지 묻는 메시지가 표시되면 항상 `y`로 응답합니다. 오류가 해결되었습니까?

```

154 echo
155 echo "Creating an EC2 instance in "$region
156 instanceDetails=$(aws ec2 run-instances W
157 --image-id $imageId W
158 --count 1 W
159 --instance-type $instanceType W
160 --region $region W
161 --subnet-id $subnetId W
162 --security-group-ids $securityGroup W
163 --tag-specifications 'ResourceType=instance,Tags=[{Key=Name,Value=cafeserver}]' W
164 --associate-public-ip-address W
165 --iam-instance-profile Name=LabInstanceProfile W
166 --profile $profile W
167 --user-data file://create-lamp-instance-userdata-v2.txt W
168 --key-name $key )

```

## 문제 2

`run-instances`에 대한 호출이 성공하면 퍼블릭 IP 주소가 새 인스턴스에 할당됩니다. 하지만 SSH를 통해 키 페어 파일로 인스턴스에 연결하려고 하면 연결에 실패합니다(*Permission denied* 오류).

```

[ec2-user@cli-host starters]$ ssh -i path-to/$key ec2-user@34.218.237.141
Warning: Identity file path-to/ not accessible: No such file or directory.
The authenticity of host '34.218.237.141 (34.218.237.141)' can't be established.
ECDSA key fingerprint is SHA256:C8hbJyGPUW63NsdydLmBwzvqa9LQoQsVnnk8+xJ3X1U.
ECDSA key fingerprint is MD5:83:b9:04:aa:68:c9:e5:f2:f1:ee:3b:df:c2:81:cb:92.
Are you sure you want to continue connecting (yes/no)? yes
Warning: Permanently added '34.218.237.141' (ECDSA) to the list of known hosts.
Permission denied (publickey,gssapi-keyex,gssapi-with-mic).

```

### 문제 2를 해결하기 위한 팁:

- 이 인스턴스는 Amazon Linux 2 인스턴스입니다. 기본 사용자 이름은 **ec2-user**입니다.
- 키 페어 파일에 대한 권한이 정확하게 설정되어 있습니까 (예: `chmod 400`)? 어떤 키 페어 파일을 사용 중입니까? 인스턴스가 연결에 사용할 것을 요구하는 키 페어 파일을 사용 중입니까?
- 중요한 정보 소스인 [AWS CLI 명령 참조](#)를 사용하는 것을 잊지 마십시오.
- 문제를 찾아 스크립트를 올바르게 수정했다고 생각된다면 스크립트를 다시 실행합니다.
  - SSH를 통해 인스턴스에 연결할 수 있습니까? 키 페어를 이미 다운로드한 랩톱에서 SSH 연결을 시도해 봅니다. 인스턴스가 완전히 부팅되는 데는 최대 5분이 걸립니다.
  - SSH를 통해 연결할 수 있어야 합니다. 이 SSH 연결을 열어 둡니다.





인스턴스가 생성되었으며 SSH 를 통해 정상적으로 연결할 수 있으나 테스트 웹페이지를 로드할 수 없습니다.

### 문제 3 을 해결하기 위한 팁:

- 웹 서버는 TCP 포트 80 을 통해 실행됩니다.
- SSH 를 통해 인스턴스에 연결한다면 웹 서버 서비스가 실행 중인지 확인할 수 있습니까? 웹 서버 서비스 이름은 **httpd** 입니다.

29. SSH 를 통해 인스턴스에 연결되어 있는 동안 이 명령을 실행하여 포트 스캐닝 도구인 nmap 을 설치합니다.

```
sudo yum install -y nmap
```

30. 그런 다음 이 명령을 실행합니다. 이때 *<public-ip>*는 LAMP 인스턴스의 실제 퍼블릭 IP 주소로 대체합니다.

```
nmap -Pn <public-ip>
```

- 이 명령으로 반환된 결과에서 액세스 가능한 포트를 알 수 있습니다. 기대했던 결과가 나왔습니까?

```
[ec2-user@web-server ~]$ nmap -Pn 35.87.49.27

Starting Nmap 6.40 ( http://nmap.org ) at 2023-10-14 04:47 UTC
Nmap scan report for ec2-35-87-49-27.us-west-2.compute.amazonaws.com (35.87.49.27)
Host is up (0.00028s latency).
Not shown: 998 filtered ports
PORT      STATE SERVICE
22/tcp    open  ssh
8080/tcp   closed http-proxy

Nmap done: 1 IP address (1 host up) scanned in 4.86 seconds
[ec2-user@web-server ~]$
```

```
[ec2-user@cli-host starters]$ aws ec2 describe-instances --query "Reservations[0].Instances[0].SecurityGroups[0].GroupId"
"sg-0f72d75dd68a73866"
[ec2-user@cli-host starters]$
```

```
[ec2-user@cli-host starters]$ aws ec2 authorize-security-group-ingress --group-id sg-0f72d75dd68a73866 --protocol tcp --port 80 --cidr 0.0.0.0/0
```



```
[ec2-user@cli-host starters]$ aws ec2 describe-security-groups --group-ids sg-0f72d75dd68a73866
{
  "SecurityGroups": [
    {
      "IpPermissionsEgress": [
        {
          "IpProtocol": "-1",
          "PrefixListIds": [],
          "IpRanges": [
            {
              "CidrIp": "0.0.0.0/0"
            }
          ],
          "UserIdGroupPairs": [],
          "Ipv6Ranges": []
        }
      ],
      "Description": "cafeSG",
      "IpPermissions": [
        {
          "PrefixListIds": [],
          "FromPort": 80,
          "IpRanges": [
            {
              "CidrIp": "0.0.0.0/0"
            }
          ],
          "ToPort": 80,
          "IpProtocol": "tcp",
          "UserIdGroupPairs": [],
          "Ipv6Ranges": []
        },
        {
          "PrefixListIds": [],
          "FromPort": 8080,
          "IpRanges": [
            {
              "CidrIp": "0.0.0.0/0"
            }
          ],
          "ToPort": 8080,
          "IpProtocol": "tcp",
          "UserIdGroupPairs": [],
          "Ipv6Ranges": []
        }
      ]
    }
  ]
}
```

**웹페이지를 다시 테스트하고 user-data 스크립트가 실행되었는지 확인합니다.**

31.3 번째 문제를 해결했다고 생각되면 브라우저에서 `http://<public-ip>`를 로드합니다. 이때 `<public-ip>`는 인스턴스의 실제 IPv4 퍼블릭 IP 주소로 대체합니다.

문제 3을 해결했다면 *웹 서버 로드됨 (Hello From Your Web Server!)*이라는 메시지가 표시됩니다.



팁: 전체 로그 파일을 보고 싶다면 `sudo cat /var/log/cloud-init-output.log` 를 사용합니다.

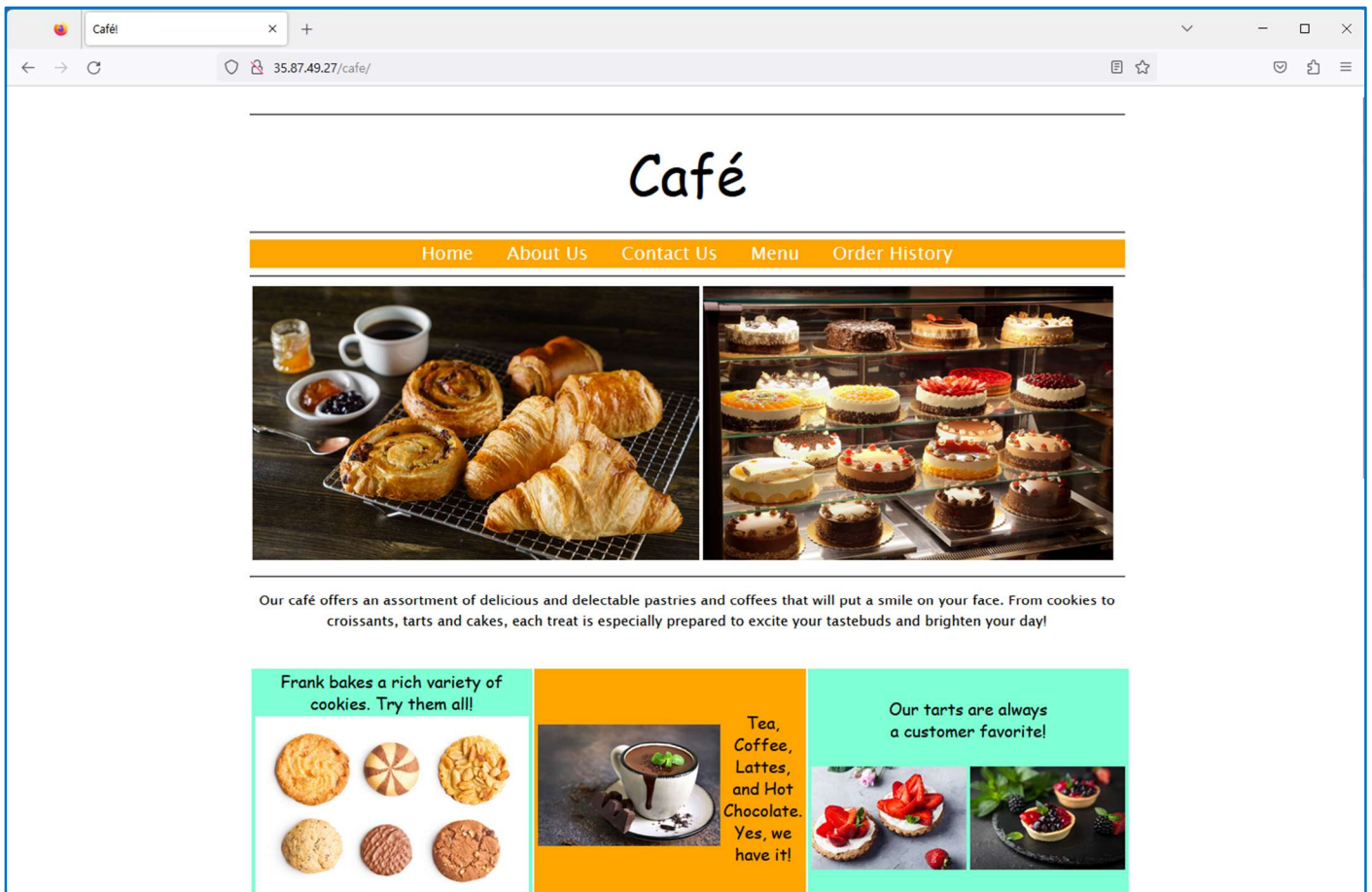
```
Cloud-init v. 19.3-46.amzn2.0.1 running 'init-local' at Sat, 14 Oct 2023 04:34:50 +0000. Up 9.92 seconds.
Cloud-init v. 19.3-46.amzn2.0.1 running 'init' at Sat, 14 Oct 2023 04:34:52 +0000. Up 11.39 seconds.
ci-info: +-----+-----+-----+-----+-----+-----+
ci-info: | Device | Up | Address | Mask | Scope | Hw-Address |
ci-info: +-----+-----+-----+-----+-----+-----+
ci-info: | eth0 | True | 10.200.0.159 | 255.255.255.0 | global | 02:b8:a4:fa:d7:83 |
ci-info: | eth0 | True | fe80::b8:a4ff:fe80:d783/64 | . | link | 02:b8:a4:fa:d7:83 |
ci-info: | lo | True | 127.0.0.1 | 255.0.0.0 | host | . |
ci-info: | lo | True | ::1/128 | . | host | . |
ci-info: +-----+-----+-----+-----+-----+-----+
ci-info: +-----+-----+-----+-----+-----+-----+
ci-info: | Route | Destination | Gateway | Genmask | Interface | Flags |
ci-info: +-----+-----+-----+-----+-----+-----+
ci-info: | 0 | 0.0.0.0 | 10.200.0.1 | 0.0.0.0 | eth0 | UG |
ci-info: | 1 | 10.200.0.0 | 0.0.0.0 | 255.255.255.0 | eth0 | U |
ci-info: | 2 | 169.254.169.254 | 0.0.0.0 | 255.255.255.255 | eth0 | UH |
ci-info: +-----+-----+-----+-----+-----+-----+
ci-info: +-----+-----+-----+-----+-----+-----+
ci-info: | Route | Destination | Gateway | Interface | Flags |
ci-info: +-----+-----+-----+-----+-----+-----+
ci-info: | 9 | fe80::/64 | :: | eth0 | U |
ci-info: | 11 | local | :: | eth0 | U |
ci-info: | 12 | ff00::/8 | :: | eth0 | U |
ci-info: +-----+-----+-----+-----+-----+-----+
Cloud-init v. 19.3-46.amzn2.0.1 running 'modules:config' at Sat, 14 Oct 2023 04:34:54 +0000. Up 13.41 seconds.
Loaded plugins: extras suggestions, langpacks, priorities, update-motd
--> amazon-ssm-agent-3.2.1377.0-1.amzn2.x86_64 from installed removed (updateinfo)
--> ec2-hibinit-agent-1.0.2-5.amzn2.noarch from installed removed (updateinfo)
--> 1:system-release-2-16.amzn2.x86_64 from amzn2-core removed (updateinfo)
--> 1:system-release-2-15.amzn2.x86_64 from installed removed (updateinfo)
--> expat-2.1.0-15.amzn2.0.2.x86_64 from installed removed (updateinfo)
--> mdadm-4.0-5.amzn2.0.3.x86_64 from amzn2-core removed (updateinfo)
--> amazon-linux-extras-2.0.1-1.amzn2.noarch from installed removed (updateinfo)
--> amazon-linux-extras-yum-plugin-2.0.1-1.amzn2.noarch from installed removed (updateinfo)
--> amazon-linux-extras-2.0.3-1.amzn2.noarch from amzn2-core removed (updateinfo)
--> ec2-hibinit-agent-1.0.2-6.amzn2.noarch from amzn2-core removed (updateinfo)
--> amazon-linux-extras-yum-plugin-2.0.3-1.amzn2.noarch from amzn2-core removed (updateinfo)
--> mdadm-4.0-5.amzn2.0.2.x86_64 from installed removed (updateinfo)
--> libtiff-4.0.3-35.amzn2.0.14.x86_64 from installed removed (updateinfo)
--> expat-2.1.0-15.amzn2.0.3.x86_64 from amzn2-core removed (updateinfo)
```

## 과제 4: 새 웹사이트의 기능 확인

33. 새 웹사이트가 배포되었는지 확인합니다.

웹 브라우저에서 `http://<public-ip>/cafe/`를 로드합니다. 이때 `<public-ip>`는 실제 IPv4 퍼블릭 IP 주소로 대체합니다.

모든 작업이 올바르게 완료되었다면 개선된 버전의 카페 웹사이트가 표시됩니다. 축하합니다.

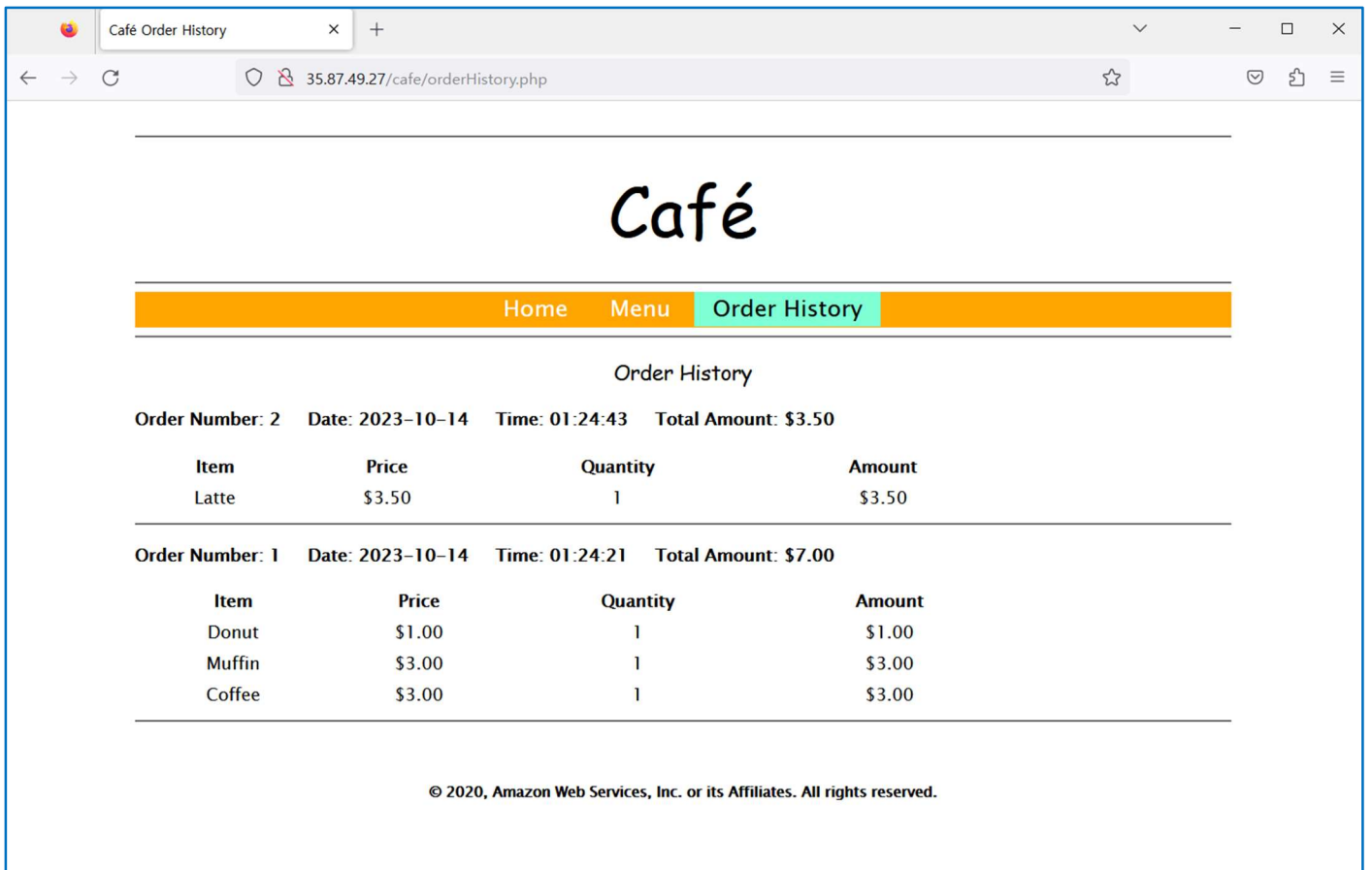


34. 테스트삼아 주문을 해보겠습니다.

- **메뉴 (Menu)** 링크를 클릭합니다. [http:// <public-ip>/cafe/menu.php](http://<public-ip>/cafe/menu.php) 라는 새 페이지가 로드됩니다.
- 가장 좋아하는 디저트 몇 개를 골라 수량을 설정하고 하단으로 스크롤하여 **주문 제출 (Submit Order)**을 클릭합니다.
- **주문 확인 (Order Confirmation)** 페이지와 함께 항목 세부 정보가 표시됩니다.
- 다른 항목으로 주문을 다시 한 다음 **주문 내역 (Order History)** 페이지를 클릭합니다. 주문 2 건의 세부 정보가 캡처된 것을 확인할 수 있습니다.

**참고:** 주문 세부 정보는 여러분이 시작한 인스턴스에서 실행 중인 데이터베이스에 캡처 및 저장됩니다.





## 카페 소식



Nikhil 과 Sofia 는 카페에서 큰 명성을 얻고 있습니다. 고객들은 이제 온라인으로 주문할 수 있다는 사실을 알아차리기 시작했습니다. Nikhil 과 Sofia 는 배포 스크립트의 문제를 해결하도록 도와준 Mateo 에게 고마움을 느낍니다.

한편 Martha 와 Frank 는 새 웹사이트의 또 다른 이점도 발견했습니다. 매일 저녁 온라인으로 이루어진 익일 수령 주문을 확인할 수 있다는 점이었습니다. Martha 와 Frank 가 주문 내역을 미리 볼 수 있기 때문에 다음 날 아침에 더 만들어야 할 디저트를 파악하여 품질을 예방할 수 있습니다. Martha 와 Frank 는 온라인 웹사이트를 구축하는 것이 비즈니스에 얼마나 도움이 되는지 깨닫기 시작했습니다.

## 실습 완료

축하합니다. 활동을 마쳤습니다.

35. 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.

36. 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 양식](#)에서 세부 정보를 제공해 주십시오.

*© 2022 Amazon Web Services, Inc. and its affiliates. All rights reserved. 본 내용은 Amazon Web Services, Inc.의 사전 서면 허가 없이 전체 또는 일부를 복제하거나 재배포할 수 없습니다. 상업적인 복제, 대여 또는 판매는 금지됩니다.*

# Troubleshooting the Creation of an EC2 Instance

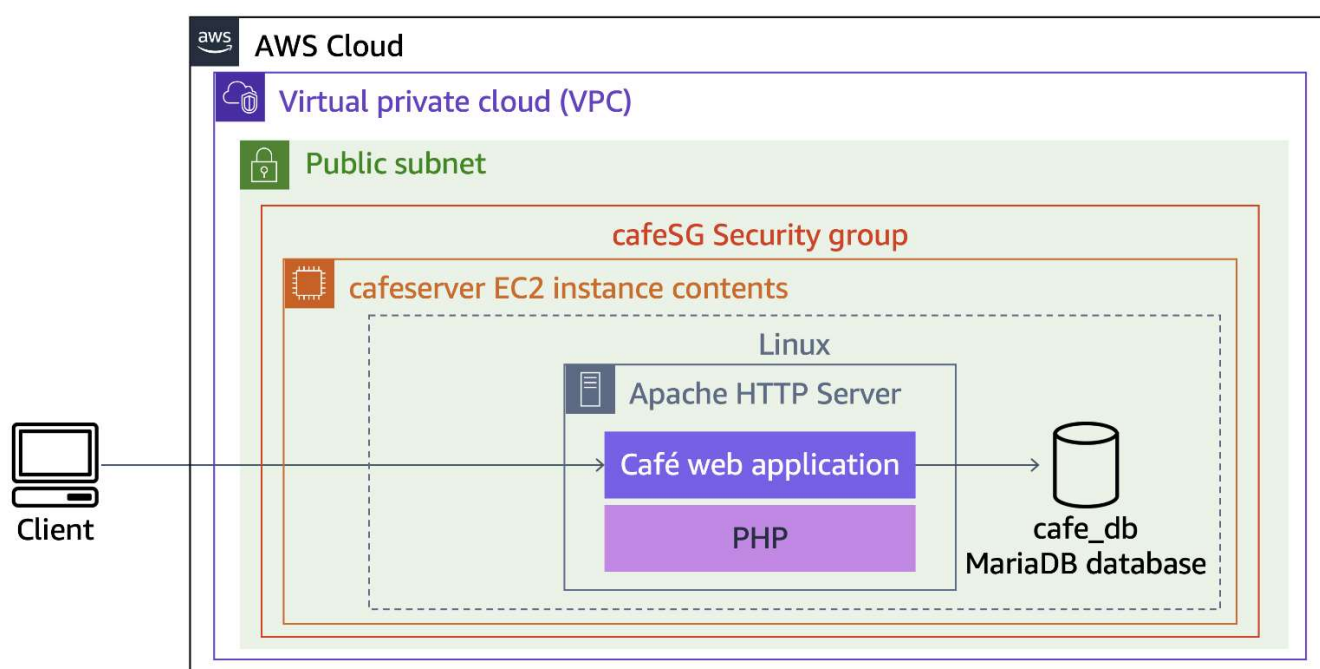
## Activity overview

In this activity, you use the AWS Command Line Interface (AWS CLI) to launch Amazon Elastic Compute Cloud (Amazon EC2) instances.

When you create the instance, you will reference a user data script to configure the instance to have an Apache web server, a MariaDB relational database (which is a fork of the MySQL relational database), and PHP running on the instance. Together, these software packages installed on a single machine are often referred to as a LAMP stack (**L**inux, **A**pache web server, **M**ySQL, and **P**HP). Using a LAMP stack is a common way to create a website with a database backend on a single machine.

The same user data file will deploy website files and run database configuration scripts on the instance. The result will be an instance that hosts the Café Web Application.

The following diagram shows the architecture that you will create in this activity.



## Objectives

After completing this activity, you should be able to do the following:

- Launch an EC2 instance by using the AWS CLI.
- Troubleshoot AWS CLI commands and Amazon EC2 service settings by using basic troubleshooting tips and the open-source nmap utility.

## Duration

This activity will require approximately **45 minutes** to complete.

## Accessing the AWS Management Console

---

1. At the top of these instructions, choose **Start Lab** to launch your lab.

A **Start Lab** panel opens displaying the lab status.

2. Wait until the message "Lab status: ready" appears, and then choose **X** to close the **Start Lab** panel.
3. Next to **Start Lab**, choose **AWS** to open the AWS Management Console on a new browser tab. The system automatically signs you in.

**Tip:** If a new browser tab does not open, a banner or icon at the top of your browser will indicate that your browser is preventing the site from opening pop-up windows. Choose the banner or icon, and choose **Allow pop ups**.

4. Arrange the AWS Management Console so that it appears alongside these instructions.

**Important:** Do not change the lab Region unless specifically instructed to do so.

## Task 1: Connecting to the CLI Host instance

---

In this task, you use EC2 Instance Connect to connect to the *CLI Host* EC2 instance that was created when the lab was provisioned. You will use this instance to run CLI commands.

5. On the **AWS Management Console**, in the **Search** bar, enter and choose **EC2** to open the **Amazon EC2 Management Console**.
6. In the navigation pane, choose **Instances**.
7. From the list of instances, select the **CLI Host** instance.
8. Above the list of instances, choose **Connect**.
9. On the **EC2 Instance Connect** tab, choose **Connect**.

**Note:** If you prefer to use an SSH client to connect to the EC2 instance, see the guidance to [Connect to Your Linux Instance](#).

Now that you are connected to the CLI Host instance, you can configure and use the AWS CLI to call AWS services.

## Task 2: Configuring the AWS CLI

---

In this task, you configure the AWS CLI by providing the configuration parameters that were made available to you when the lab was provisioned. After configuration, you run CLI commands to interact with AWS services.

**Note:** Amazon Linux Amazon Machine Images (AMIs) already have the AWS CLI preinstalled.

10. To set up the AWS CLI profile with credentials, run the following command:



**Tip:** In EC2 Instance Connect, you might need to use the context (right-click) menu to paste values in the terminal.

11. When prompted, enter the following information:

- **AWS Access Key ID:** At the top of these instructions, choose `Details`, and then choose `Show`. Copy the **AccessKey** value into the terminal window.
- **AWS Secret Access Key:** Copy the **SecretKey** value from the lab instructions into the terminal window.
- **Default region name:** Copy the **LabRegion** value from the lab instructions into the terminal window.
- **Default output format:** Enter `json`

## Task 3: Creating an EC2 instance by using the AWS CLI

In this task, you observe and run a *shell script*, which was provided to you, to create an EC2 LAMP instance by using AWS CLI commands. The script intentionally contains issues. Your challenge is to find the issues and resolve them. As you resolve each issue, you can run the script again to check that the issue was resolved.

### Task 3.1: Observe the script details

First, create a backup of the script that you will edit in a later step.

12. To change to the directory where the script file exists and create a backup of it, run the following commands:

```
cd ~/sysops-activity-files/starters
cp create-lamp-instance-v2.sh create-lamp-instance.backup
```

**Tip:** It is a good practice to back up files before you modify them.

13. Open the *create-lamp-instance-v2.sh* script file in a command line text editor, such as VI.

**Tip:** To open the file in read-only mode in the VI editor, use the following command:

```
view create-lamp-instance-v2.sh
```

14. Analyze the contents of the script:

**Tip:** If you are using VI, you can display the line numbers by typing `:set number` and then pressing Enter.

- **Line 1:** This is a bash file, so the first line contains `#!/bin/bash`.
- **Lines 7–11:** The instance size is set to `t3.small`, which should be large enough to run the database and web server.
- **Lines 16–29:** The script invokes the AWS CLI `describe-regions` command to get a list of all AWS Regions. In each Region, the script queries for an existing VPC that is named *Cafe VPC*. After

finding the VPC, the script captures the *VPC ID* and *Region*, and breaks out of the while loop. This is the VPC where the LAMP instance for the Café Web Application should be deployed.

- **Lines 31–55:**
  - The script invokes AWS CLI commands to look up the *subnet ID*, *key pair name*, and *AMI ID* values that are needed to create the EC2 instance.
  - Notice that line 32 ends with a backslash ( \ ) character. This character can be used to wrap a single command on another line in the script file. This technique is used many times in the script to make it easier to read.
- **Lines 57–122:** The script cleans up the AWS account for situations where this script already ran in the account and is being run again. The script checks whether an instance that is named *cafeserver* already exists and whether a security group that includes *cafeSG* in its name already exists. If either resource is found, the script prompts you to delete them.
- **Lines 124–152:** The script creates a new security group with ports 22 and 80 open.
- **Lines 154–168:**
  - The script creates a new EC2 instance. Notice how this part of the script uses the values that were set in lines 8 and 10, and the values that were collected in lines 16–57.
  - Notice the reference to the user data file. You will review the details of this file in another step.
  - The entire call to create the instance is captured in a variable that is named *instanceDetails*. The contents of this variable are then echoed out to the terminal on line 177, and they are formatted for easier viewing by using a Python JSON tool.
- **Lines 179–188:** The *instanceId* value is parsed out of the *instanceDetails* variable. Then, a while loop checks every 10 seconds to see if a public IP address has been assigned to the instance. After the check succeeds, the public IP address is written to the terminal.

15. Exit the text editor.

**Tip:** If you are using VI, enter `:q!`

16. To display the contents of the user data script, run the following command:

```
cat create-lamp-instance-userdata-v2.txt
```

Notice how the user data script runs a series of commands on the instance after it is launched. These commands will install a web server, PHP, and a database server.

## Task 3.2: Try to run the script

17. Now that you have an idea what the shell script is designed to do, try to run it:

```
./create-lamp-instance-v2.sh
```

The script fails and exits without successfully completing. *This behavior is expected.*

## Task 3.3: Troubleshoot issues

### Issue #1

The terminal output displays the following message: "An error occurred (InvalidAMIID.NotFound) when calling the RunInstances operation: The image id '[ami-xxxxxxxxxx]' does not exist".

#### Tips to help you resolve issue #1:

- Locate the line in the script that led to this error. Does anything look incorrect in the script?
- Was the correct Region value used for the run-instances command?

- After you identify the issue, update the script to fix the issue, and run the script again. If you are prompted to delete any existing security groups or instances that were created when the script ran previously, always reply with `Y`. Is the error resolved?

After you fix the issue, the `run-instances` command succeeds, and a public IPv4 address is assigned to the new instance.

### Try to connect to the webpage

In a browser, navigate to the following address. Replace `<public-ip>` with the Public IPv4 address of the new instance that you created: `http://<public-ip>`

The attempt fails. You need to resolve issue #2.

## Issue #2

The `run-instances` command succeeded, and a public IP address was assigned to the new instance. However, you cannot load the test webpage.

18. Connect to the new *LAMP instance* by using EC2 Instance Connect, which is the method that you used to connect to the CLI Host instance.

### Tips to help you resolve issue #2:

- The web server runs on TCP port 80. Is it open?
  - Can you verify that the web server service is running? The web server service name is *httpd*.
19. In the terminal window for the *CLI Host* instance, run the following command to install `nmap`, which is a port scanning tool:

```
sudo yum install -y nmap
```

20. Next, run the following command. Replace `<public-ip>` with the actual public IPv4 address of your *LAMP instance*:

```
nmap -Pn <public-ip>
```

The output from this command shows which ports are accessible. Do the results match what you expected? Do you know what needs to be changed based on the output?

## Test whether the user data script ran

21. After you identify and resolve the issue, in a browser, navigate to the following address. Replace `<public-ip>` with the Public IPv4 address of the new instance that you created: `http://<public-ip>`

If you resolved issue #2 successfully, you should see the following message: "Hello From Your Web Server!"

22. Check the log file that shows whether the user data script ran as expected.

In the terminal window for the *LAMP instance*, run the following command to see the log file entries as they are written:

```
sudo tail -f /var/log/cloud-init-output.log
```

On an Amazon Linux instance, the *cloud-init* service runs the commands in the user data file.

Observe the log file entries. Notice the message that is related to the installation of MariaDB and PHP. There should be no error messages.

You should also see messages related to files for the Café Web Application that were downloaded and extracted to this instance, such as "Create Database script completed". When you are finished with your observations, to exit the tail utility, enter **Ctrl-C** **Tip:** To view the entire log file, run `sudo cat /var/log/cloud-init-output.log`

## Task 4: Verifying the functionality of the website

---

23. Verify that the website is deployed.

In a browser, navigate to the following address. Replace *<public-ip>* with the Public IPv4 address of the new instance that you created: `http://<public-ip>`

If you are successful, you should see the home page for the café website. Congratulations!

24. Test whether you can order items through the website.

- Choose the **Menu** link. A new page loads at `http://<public-ip>/cafe/menu.php`.
- Choose a few desserts to order, and then choose **Submit Order**. The Order Confirmation page displays with line-item details.
- Place another order for different items. Then, choose the **Order History** page. The details of both orders were captured.

**Note:** The order details are being captured and stored in the database that is running on the LAMP instance that you launched.

## Conclusion

---

Congratulations! You now have successfully done the following:

- Launched an EC2 instance by using the AWS CLI.
- Troubleshoot AWS CLI commands and Amazon EC2 service settings by using basic troubleshooting tips and the open-source nmap utility.

## Lab complete

---

Congratulations! You have completed the activity.

25. At the top of this page, choose **End Lab** and then choose **Yes** to confirm that you want to end the activity.

A panel appears indicating that "You may close this message box now. Lab resources are terminating."

26. To close the **End Lab** panel, choose the **X** in the upper-right corner.

## Additional resources

---

- [Launching, Listing, and Terminating Amazon EC2 Instances](#)
- [Connect to Your Linux Instance Using EC2 Instance Connect](#)
- [User Data and Shell Scripts](#)

For more information about AWS Training and Certification, see [AWS Training and Certification](#).

*Your feedback is welcome and appreciated.*

If you would like to share any suggestions or corrections, please provide the details in our [AWS Training and Certification Contact Form](#).

© 2023, Amazon Web Services, Inc. or its affiliates. All rights reserved. This work may not be reproduced or redistributed, in whole or in part, without prior written permission from Amazon Web Services, Inc. Commercial copying, lending, or selling is prohibited.