aws.qwiklabs.com

Architecting on AWS - Lab 1 - Explore using the AWS API to deploy an EC2 instance | Qwiklabs

Qwiklabs

30-39 minutes

Note: Do not include any personal, identifying, or confidential information into the lab environment. Information entered may be visible to others.

Corrections, feedback, or other questions? Contact us at *AWS Training and Certification*.

Lab Overview

The Amazon Web Services (AWS) environment is an integrated collection of hardware and software services designed to enable quick and inexpensive use of resources. The AWS application programming interface (API) sits atop the AWS environment. An

API represents a way to communicate with a resource. With AWS, nothing gets done without using the AWS API. There are different ways to interact with AWS resources. The AWS Management Console provides a simple web interface for Amazon Web Services. The AWS Command Line Interface (CLI) is a unified tool to manage your AWS services through the command line. AWS CloudFormation is a service that gives you an easy way to create and provision AWS resources; and, manage them in an orderly and predictable fashion. Whether you access AWS through the AWS Management Console or the command line tools, you are using tools that make calls to the AWS API.

In this lab, you will interact with the Amazon API using the AWS Management Console, the AWS Command Line Interface (CLI), and AWS CloudFormation. You will use each of these services to explore the different methods to launch an Amazon EC2 instance.

Objectives

After completing this lab, you should be able to:

- Understand the fundamentals of using the AWS Management Console to create resources.
- Understand the fundamentals of using the AWS CLI to create resources.
- Understand the fundamentals of using AWS CloudFormation to create resources.

Prerequisites

This lab requires:

- Access to a notebook computer with Wi-Fi and Microsoft Windows, macOS, or Linux (Ubuntu, SuSE, or Red Hat)
- An internet browser such as Chrome, Firefox, or Microsoft Edge
- A plaintext editor

Duration

This lab requires approximately **45** minutes to complete.

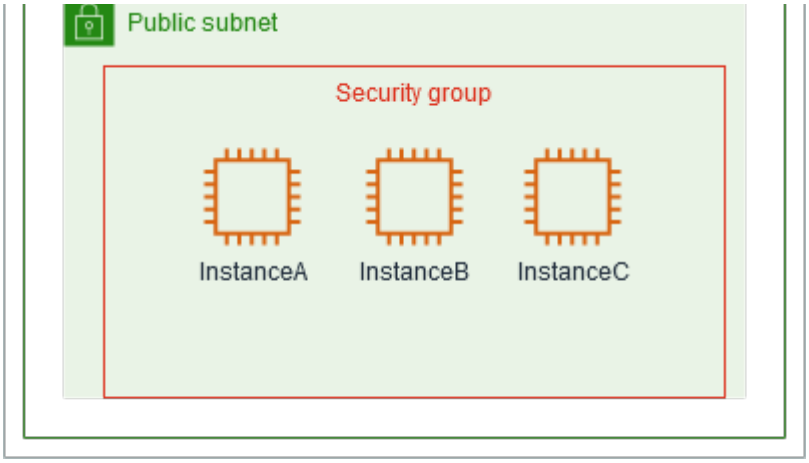
AWS Services Not Used in This Lab

AWS services not used in this lab are disabled in the lab environment. In addition, the capabilities of the services used in this lab are limited to what the lab requires. Expect errors when accessing other services or performing actions beyond those provided in this lab guide.

Lab Environment

The lab environment provides you with the following resources to get started: An Amazon Virtual Private Cloud (Amazon VPC), underlying necessary network structure, a Security Group allowing the HTTP port, an Amazon EC2 instance with the Amazon CLI installed, and an associated Amazon EC2 instance profile. The instance profile contains the permissions necessary to the allow the AWS Systems Manager Session Manager feature to access the Amazon EC2 instance.

The following diagram shows the expected architecture of the important lab resources you build and how they should be connected at end of the lab.



Start Lab

1. At the top of your screen, launch your lab by choosing Start Lab

   This starts the process of provisioning your lab resources. An estimated amount of time to provision your lab resources is displayed. You must wait for your resources to be provisioned before continuing.

   If you are prompted for a token, use the one distributed to you (or credits you have purchased).

2. Open your lab by choosing Open Console

   This opens an AWS Management Console sign-in page.

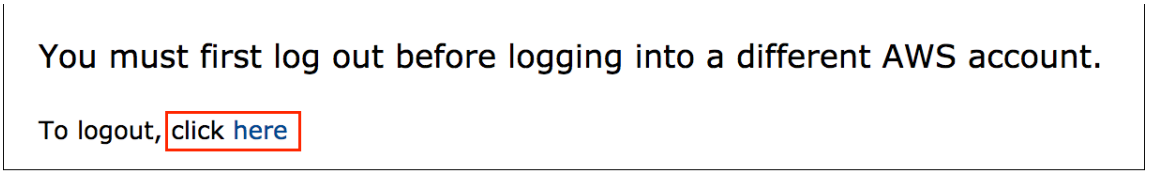
3. On the sign-in page, configure:

- **IAM user name:**
- **Password:** Paste the value of **Password** from the left side of the lab page
- Choose Sign In

  Do not change the Region unless instructed.

  Common Login Errors

  Error: You must first log out



If you see the message, **You must first log out before logging into a different AWS account:**

- Choose **here**
- Close your browser tab to return to your initial lab window
- Choose Open Console again

Amazon Elastic Compute Cloud - (Amazon EC2) provides scalable computing capacity in the Amazon Web Services (AWS) Cloud. You can use Amazon EC2 to launch virtual servers as you need, configure security and networking, and manage storage. Amazon EC2 enables you to scale up or down to handle changes in requirements.

Task 1: Create an Amazon EC2 Instance using the AWS Management Console

There is more than one method to provision the resources you want in your AWS environment. The web based AWS Management Console provides a graphical user interface to interact with services offered by Amazon Web Services. In this task, you launch an Amazon EC2 instance using the AWS Management Console.

Task 1.1: Navigate to the EC2 console

4. If you have not already opened the AWS Management Console, follow the instructions in the **Start Lab** section to log in to the AWS Management Console.
5. On the Services menu, choose **EC2**.

Note: You can also search for in the unified search bar at the top of the console.

Caution This lab is designed to use the new EC2 Console. If **New EC2 Experience** is available at the top-left of your screen, ensure **New EC2 Experience** is selected.

The **Amazon EC2 Management Console** is displayed.

Task 1.2: Begin the Launch instances wizard

6. From the console dashboard, choose the Launch instance button.
7. Choose the Launch instance button.

   The **Step 1: Choose an Amazon Machine Image (AMI)** page is displayed.

Task 1.3: Choose an Amazon Machine Image

In this task, you choose an Amazon Machine Image (AMI). The AMI contains a copy of the disk volume used to launch the instance.

Examine the list of AMIs that are displayed. There are many versions of Microsoft Windows and Linux. These quick launch images are regularly updated to incorporate security patches and software that helps you use AWS services. You can also create your own AMI that includes your own data and applications, or you can select pre-built commercial applications from the **AWS Marketplace**.

8. Ensure you are on the **Quick Start** tab, located on the left side of the navigation pane.

   For this lab, use the *Amazon Linux 2* operating system from the **Quick Start** library, and not from other libraries.

9. Beside the **Amazon Linux 2 AMI** (at the top), choose the Select button. Leave the default selection of 64-bit (x86).

   The **Step 2: Choose an Instance Type** page is displayed.

   Task 1.4: Choose the Amazon EC2 instance type

   This page enables you to choose an **Instance Type**. Each Instance Type allocates a specific combination of virtual CPUs, memory, disk storage and network performance.

   You can hover over the icons to review a description of each field.

   For this lab, use a **t3.micro** Instance Type.

10. Select the checkbox next to the **t3.micro** instance type.
11. Choose the Next: Configure Instance Details button.

    The **Step 3: Configure Instance Details** page is displayed.

    Task 1.5: Configure instance details

    The VPC and other necessary networking infrastructure are provided for you in this lab. You do not need to create them. In this task, you further configure the Amazon EC2 instance.

    Verify that the Region displayed at the top-right of the console is the same as the **LabRegion** value on the left side of this lab page.

12. Configure the following settings for the instance:

    - **Network** *LabVPC*
    - **Subnet** *LabPublicSubnet*
    - **Auto-assign Public IP** *Enable*

13. Expand the Advanced Details section, if necessary.

14. Paste the following in the **User Data** section:

```
#!/bin/bash -ex
sudo yum update -y
sudo yum install -y httpd php
sudo service httpd start
sudo systemctl enable httpd.service
cd /var/www/html
wget https://us-west-2-tcprod.s3.amazonaws.com
/courses/ILT-TF-200-ARCHIT/v7.1.1/lab-1-
EC2/scripts/instanceAdata.zip
unzip instanceAdata.zip
```

    This user data script performs the necessary functions to enable this Amazon EC2 instance as a simple web server.

    The remaining settings on the page can be left at their default values.

15. Choose the Next: Add Storage button.

    The **Step 4: Add Storage** page is displayed.

    Task 1.6: Add storage

    This page can be used to modify ephemeral instance storage and add additional Amazon Elastic Block Store (EBS) disk volumes attached to the instance. The EBS volumes can be configured in both their size and performance.

    You can hover over the icons to review a description of each field.

    In this lab the default storage settings are all that is need. No changes are required.

16. Choose the Next: Add Tags button.

The **Step 5: Add Tags** page is displayed.

Task 1.7: Add Tags

Tags allow you to categorize your AWS resources in different ways, such as by purpose, owner, or environment. Tags can be applied to most AWS Cloud resources. Each tag consists of a *Key* and a *Value*, both of which you define. One use of tags is when you must manage many resources of the same type. You can quickly search for and identify a specific resource by the tag you have applied to it.

In this task, you add a tag to the Amazon EC2 instance.

17. Choose Add Tag then configure:

- **Key:**
- **Value:**

This name will appear on the instance in the Amazon EC2 Management Console.

18. Choose the Next: Configure Security Group button.

The **Step 6: Configure Security Group** page is displayed.

Task 1.8: Select a security group

Security groups allow you to define both the allowed/denied and the inbound/outbound traffic for the Elastic Network Interface (ENI). The ENI is attached to an Amazon EC2 instance. A pre-configured security group which allows all traffic over port 80 has been provided to you for use in this lab. Port 80 is the default port for HTTP traffic, and is necessary for the web server you launch in this lab to work correctly.

19. Choose the *Select an existing security group* option near the

Assign a security group: text.

20. Fill in the checkbox next to the security group which has a name like **xxxx-LabInstanceSecurityGroup-xxxx**.

    Note: The Security Group ID should match the **LabInstanceSecurityGroupID** value found to the left of these instructions.

21. Choose the Review and Launch button.

22. There is a warning box displayed regarding adding an open port 22 to the security group. This message box can safely be closed. Choose the **Continue** button. Port 22 is not needed in this lab.

    The **Step 7: Review Instance Launch** page is displayed.

    Task 1.9: Review the instance launch

    This page displays a summary of the configuration for the Amazon EC2 instance you are about to launch. Take a moment to review that the configuration is correct.

23. Choose the Launch button.

    A **Select an existing key pair or create a new key pair** window is displayed.

24. Select **Proceed without a key pair** from the drop-down menu.

25. Fill the checkbox next to the phrase that starts with **I acknowledge that without a key pair ...**.

26. Choose the Launch Instances button.

    The **Launch Status** page is displayed.

    Your Amazon EC2 instance is now launched and configured as you specified.

27. Choose the View Instances button.

    The **Amazon EC2 console** is displayed.

    The Amazon EC2 instance name *InstanceA* is initially in a *pending* state. The state then changes to *running* indicating that the instance has finished booting.

28. Occasionally choose the console refresh button and wait for the **Instance State** to change to **Running**.

    Congratulations, you have launched an Amazon EC2 instance using the Amazon Management Console.

    Task 1.10: Connect to the web page hosted on the new instance

    In this task, you connect to the newly launched Amazon EC2 instance to verify that the web server is working as intended.

29. Select **InstanceA** from the list.

30. Select the **Networking** tab in the lower pane.

    Note: The horizontal edges of the containers displayed on the console can be resized if you need to make any section of the console larger.

31. Locate the **Public IPv4 DNS** value.

32. Copy the public DNS value. Do not choose the open address option, as HTTPS is not setup for this lab environment.

33. Open a new browser tab and paste the public DNS value for *InstanceA* in the URL address bar.

    Caution: You may need to wait for at least 3 minutes for the Amazon EC2 instance to finish health checks and for the web server to initially start. Also, SSL/TLS was not setup for your Amazon EC2 instance in this lab. Attempting to navigate to the

DNS address for the web server prefixed with HTTPS:// will fail.

The web server test page is displayed. This web page is hosted by *InstanceA*.

Congratulations, you have verified the web server on *InstanceA* is working and reachable.

Task 2: Create an Amazon EC2 instance using the AWS Command Line Interface

In this task you use the AWS Command Line Interface (CLI) to create and launch a new Amazon EC2 instance. The AWS CLI makes it easy to automate the provisioning and configuration of AWS resources.
The new instance will also be configured as a web server.

The complete documentation of the AWS CLI command 'run-instances' can be found in the official documentation https://docs.aws.amazon.com/cli/latest/reference/ec2/run-instances.html.

An Amazon EC2 instance pre-configured with the AWS CLI has been provided to you for use in this lab. It has the name *Command Host*.

Task 2.1: Create a connection to the Command Host using AWS Systems Manager Session Manager

34. Return to the Amazon EC2 Management Console.
35. In the left navigation pane, choose **Instances**.
36. Clear the checkbox next to the instance with the name **InstanceA**.
37. Fill the checkbox next to the instance with the name **Command Host**.

38. Choose the **Connect** button.

    The **Connect to instance** page is displayed.

39. Select the **Session Manager** tab.

    With Session Manager, you can connect to Amazon EC2 instances without requiring exposing the SSH port on your firewall or Amazon Virtual Private Cloud (Amazon VPC) security group. Refer to AWS Systems Manager Session Manager for more complete information on this feature.

40. Choose the **Connect** button.

    A new browser tab or window opens with a connection to the instance.

Task 2.2: Obtain the AMI to Use

The *Command Host* uses the Amazon Linux 2 Operating System (OS). This OS is pre-installed with the AWS CLI, so installation of the AWS CLI is not necessary as it is with other OS choices. Similar to using the AWS Management Console, one of the parameters required when launching an Amazon EC2 instance is specifying the Amazon Machine Image (AMI) to be used. The AMI is used to populate the boot disk of the instance. The quick start selection of AMIs are continually patched and updated by AWS, so it is recommended to always use the latest AMI when launching instances.

You will use the **AWS Systems Manager Parameter Store** to obtain the ID of the most recent *Amazon Linux 2* AMI. AWS maintains a list of standard AMIs in the Parameter Store, making this task easier to automate.

41. Run this command in your *Command Host* session:

```
# Set the Region
```

```
AZ=`curl -s http://169.254.169.254/latest/meta-data/placement/availability-zone`
export AWS_DEFAULT_REGION=${AZ::-1}

# Obtain latest Linux AMI
AMI=$(aws ssm get-parameters --names /aws/service/ami-amazon-linux-latest/amzn2-ami-hvm-x86_64-gp2 --query 'Parameters[0].[Value]' --output text)

echo $AMI
```

This command did the following:

- Obtained the Region where the instance is running.
- Called the AWS Systems Manager (*ssm*) and used the **get-parameters** command to retrieve a value from the Parameter Store.
- The AMI requested was for Amazon Linux 2 (*amzn2-ami*).
- The AMI ID has been stored in an Environment Variable called *AMI*.

If your session disconnects, it will lose the information stored in environment variables. To launch the new Amazon EC2 instance once you reconnect *Command Host*, you need to re-run all the steps in task 2, starting with the above commands to obtain the AMI ID.

Task 2.3: Obtain the Subnet to Use

When launching an instance, setting the **SubnetId** parameter determines where in the network of your Amazon Virtual Private

Cloud the instance is located. For this task, you are launching the new Amazon EC2 instance in a public subnet.

The following command will retrieve the *SubnetId* for a Public Subnet that has been provided for this lab.

42. Run this command in your *Command Host* session:

```
SUBNET=$(aws ec2 describe-subnets --filters
'Name=tag:Name,Values=LabPublicSubnet' --query
Subnets[].SubnetId --output text)

echo $SUBNET
```

This command uses the AWS CLI to retrieve the Subnet ID of the existing subnet which has a tag named **LabPublicSubnet**.

Note: The **Subnet ID** should match the *LabPublicSubnet* found to the left of these instructions.

Task 2.4: Obtain the Security Group to Use

In this task, you locate a provided security group for use with the new instance's Elastic Network Interface (ENI). A **LabInstanceSecurityGroup** has been provided as part of this lab, which allows inbound HTTP requests. This is the same security you used for **InstanceA**.

43. Run this command in your *Command Host* session:

```
SG=$(aws ec2 describe-security-groups --filters
Name=tag:Name,Values=LabInstanceSecurityGroup
--query SecurityGroups[].GroupId --output text)

echo $SG
```

The command retrieves the **Security Group ID** of the *LabInstanceSecurityGroup*.

Note: The **Security Group ID** should match the *LabInstanceSecurityGroup* value found to the left of these instructions.

Task 2.5: Download a User Data script

To install and configure the new instance as web server, you provide a **User Data script** that will be automatically ran when the instance launches. This script is similar to the user data that was used when you launched **InstanceA**.

44. Run this command in your *Command Host* session to download the User Data script:

```
cd ~
wget https://us-west-2-tcprod.s3.amazonaws.com
/courses/ILT-TF-200-ARCHIT/v7.1.1/lab-1-
EC2/scripts/UserDataInstanceB.txt
```

45. Run this command in your *Command Host* session to view the contents of the script:

```
cat UserDataInstanceB.txt
```

The script is used to install the web server application to the Amazon EC2 instance.

Task 2.6: Launch the Instance

You now have all the necessary information required to launch the new web server instance! To differentiate from the web

server already launched you tag this instance as **InstanceB**.

46. Run this command in your *Command Host* session to start a new Amazon EC2 instance, *InstanceB*:

```
INSTANCE=$(\
aws ec2 run-instances \
--image-id $AMI \
--subnet-id $SUBNET \
--security-group-ids $SG \
--user-data file://./UserDataInstanceB.txt \
--instance-type t3.micro \
--tag-specifications
'ResourceType=instance,Tags=
[{Key=Name,Value=InstanceB}]' \
--query 'Instances[*].InstanceId' \
--output text \
)
```

The *run-instances* command launches a new instance using these parameters:

- **Image:** Uses the AMI value obtained earlier from the Parameter Store.
- **Subnet:** Specifies the *LabPublicSubnet* obtained earlier, and by association, where in the network to launch the instance.
- **Security Group:** Uses the *LabInstanceSecurityGroup*, which permits inbound HTTP traffic on port 80.
- **User Data:** References the User Data script you downloaded. This script installs the web server application.
- **Instance Type:** Specifies the type of instance to launch.
- **Tags:** Assigns a *Name* tag with the value of *InstanceB* to the

new Amazon EC2 instance.

The **query** parameter specifies that the command should return the *Instance ID* once the instance is launched.

The **output** parameter specifies that the output of the command should be in *text* format.

The ID of the new instance has been stored in the *INSTANCE* environment variable.

47. Run this command in your *Command Host* session to display the instance id launched by the previous command.

```
echo $INSTANCE
```

Task 2.7: Wait for the Instance to be Ready

You can monitor the status of the instance via the EC2 Management Console, but you can also query the status via the AWS CLI.

48. Run this command in your *Command Host* session:

```
aws ec2 describe-instances --instance-ids $INSTANCE
```

All information related to the instance will be displayed in JSON format. Amongst this information is the instance status.

49. Press the 'q' button to return to the console.

Specific information can be filtered from the *describe-instances* command by using the **query** parameter.

50. Run this command in your *Command Host* session:

```
aws ec2 describe-instances --instance-ids $INSTANCE --query
```

```
'Reservations[].Instances[].State.Name'
--output text
```

This is the same command, but rather than displaying all available information about the instance, only the instance *State* details are displayed.

This status displays as either **pending** or **running**.

Repeat the above command until the status of **running** is returned to the terminal.

Task 2.8: Test connectivity to the new web server

In this step, you test that the new web server is working. You can retrieve a URL to the instance via the AWS CLI.

51. Run this command in your *Command Host* session:

```
aws ec2 describe-instances --instance-ids
$INSTANCE --query
Reservations[].Instances[].PublicDnsName
--output text
```

This returns the public DNS value for *InstanceB*.

52. Copy the public DNS value that is returned in the terminal.

The value should be similar to: *ec2-35-11-22-33.us-west-2.compute.amazonaws.com*

53. Paste the DNS name into a new web browser tab, then press the Enter key.

Caution: You may need to wait for at least 3 minutes for the Amazon EC2 instance to finish health checks and for the web server to initially start. Also, SSL/TLS was not setup for your

Amazon EC2 instance in this lab. Attempting to navigate to the DNS address for the web server prefixed with HTTPS:// will fail.

The web server page is displayed. This page is hosted by *InstanceB*.

You can also find this same DNS information about *InstanceB* in the Amazon EC2 Management Console.

54. Return to the web browser tab containing the Amazon EC2 Management Console.
55. Expand the navigation menu if needed, by choosing the menu icon in the top left corner.
56. Choose the **Instances** option from the navigation menu.
57. Choose the console Refresh button.

    The list should now include *InstanceB*. This is the instance which was launched from the *Command Host* via the AWS CLI.

    As demonstrated in this task, the AWS CLI makes it possible to programmatically access and control AWS services. These commands can be placed in a script and run as a standard process to deploy consistent, reliable infrastructure.

    Congratulations you have used the AWS CLI to create and launch an Amazon EC2 instance. You then confirmed that the Public DNS returned a web page hosted on *InstanceB*.

Task 3: Create an Amazon EC2 instance using the CloudFormation service

In this task you use the CloudFormation service to launch a new Amazon EC2 instance into the lab provided Amazon VPC.

58. Download the CloudFormation template by right-clicking this link and choosing to save: Task3.yaml to your local storage.

59. Open this file in a Text Editor (not a Word Processor).

    Examine the file. You will notice several sections:

- The Parameters section is used to prompt for inputs that can be used elsewhere in the template. This template defines where an AMI can be retrieved from parameter store.
- The Resources section is used to define which AWS cloud resources to be deployed. The resources section in this template is defining an Amazon EC2 instance.

    The template is written in a format called YAML, which is commonly used for configuration files. The syntax of the file is important, including the indents and hyphens. CloudFormation templates can also be written in JSON.

    You now use this **template** to launch a **CloudFormation stack**.

    Task 3.1: Navigate to the CloudFormation console

60. In the **AWS Management Console**, on the Services menu, choose **CloudFormation**.

    Note: You can also search for in the unified search bar at the top of the console.

    Task 3.2: Run the template in CloudFormation

61. Choose the Create stack button.

    The **Create stack** page is displayed.

    Note: If the console starts you on the **Stacks** page instead of the Amazon CloudFormation landing page, then you can get to the **Create stack** page in two steps.

- Choose the Create stack drop-down menu.
- Choose the **With new resources (standard)** button.

62. Configure the following:
    - Choose the Template is ready option.
    - Choose the Amazon S3 URL option.
    - Copy the **TaskTemplateUrl** value from the left side of lab instructions and paste in the **Amazon S3 URL** text box.
63. Choose the Next button.

    The **Specify stack details** page is displayed.
64. Set **Stack name:**
65. Choose the Next button.

    The **Configure stack options** page is displayed. This page can be used to specify additional parameters. You can browse the page, but leave settings at their default values.
66. Choose the Next button.

    The **Review** page is displayed. This page is a summary of all settings.
67. Choose the Create stack button.

    The stack will now enter the CREATE_IN_PROGRESS status.
68. Choose the **Stack info** tab.
69. Occasionally choose the console Refresh button.
70. Wait for the stack status to be CREATE_COMPLETE before continuing.
71. Choose the **Events** tab and scroll through the listing.

    The listing shows (in reverse order) the activities performed by CloudFormation, such as starting to create a resource and then completing the resource creation. Any errors encountered during the creation of the stack will be listed in this tab.

72. Choose the **Resources** tab.

    The listing shows the resources that are being created. CloudFormation determines the optimal order for resources to be created, such as creating the VPC before the subnet.

    Which method should you use to create Amazon EC2 instances in your AWS environment?

- **Launch from the AWS Management Console** when you quickly need to launch a one-off or temporary instance.
- **Launch via a script** when you need to automate the creation of an instance in a repeatable, reliable manner.
- **Launch via CloudFormation** when you wish to launch related resources together, and want to use the concept of infrastructure as code.

Task 3.3: Test connectivity to instance C public IP address

73. Select the **Outputs** tab.
74. Select the URL link for the *InstanceCPublicDNS*.
75. A new browser tab opens to the webpage hosted on *InstanceC*.

    Alternatively, you can copy and paste the Public DNS value into a new browser tab.

    Caution: You may need to wait for at least 3 minutes for the Amazon EC2 instance to finish health checks and for the web server to initially start. Also, SSL/TLS was not setup for your Amazon EC2 instance in this lab. Attempting to navigate to the DNS address for the web server prefixed with HTTPS:// will fail.

    Congratulations you have used a CloudFormation template to create and launch an Amazon EC2 instance. You then confirmed that the public DNS returned a web page hosted on

InstanceC.

Optional Task 4: Examine and run a second CloudFormation template

This task is provided to you if you have extra lab time or want to learn something a little more advanced. This task is optional and is not necessary to complete. You can end the lab now if you choose by following the steps to end the lab, otherwise keep reading.

In this task you download and review a second CloudFormation template. You determine what resources the AWS CloudFormation template defines and then create a stack using the template in the AWS CloudFormation Console.

Task 4.1: Navigate to the CloudFormation console

76. In the **AWS Management Console**, on the Services menu, choose **CloudFormation**.

    Note: You can also search for in the unified search bar at the top of the console.

Task 4.2: Obtain and review the CloudFormation template

77. Download the CloudFormation template by right-clicking this link and choosing to save: optional.yaml
78. Open the downloaded file in a Text Editor (not a Word Processor).
79. Review the CloudFormation template.
80. Predict what resources are created by this template.

Task 4.3: Create the CloudFormation stack

81. Choose the Create stack button.

    The **Create stack** page is displayed.

    Note: If the console starts you on the **Stacks** page instead of the Amazon CloudFormation landing page, then you can get to the **Create stack** page in two steps.

- Choose the Create stack drop-down menu.
- Choose the **With new resources (standard)** button.

82. Configure the following:

- Choose the Template is ready option.
- Choose the Amazon S3 URL option.
- Copy the **OptionalTemplateUrl** value from the left side of lab instructions and paste in the **Amazon S3 URL** text box.
- Choose the Next button.

    The **Specify stack details** page is displayed.

83. Set **Stack name:** as .

84. Choose the Next button.

    The **Configure stack options** page is displayed. This page can be used to specify additional parameters. You can browse the page, but leave settings at their default values.

85. Choose Next

    The **Review** page is displayed. This page is a summary of all settings.

86. Choose the Create stack button.

    The stack will now enter the CREATE_IN_PROGRESS status.

87. Choose the **Stack info** tab.

88. Occasionally choose the console Refresh button.

89. Wait for the stack status to be CREATE_COMPLETE.
90. Choose the **Events** tab and scroll through the listing.

    The listing shows (in reverse order) the activities performed by CloudFormation, such as starting to create a resource and then completing the resource creation. Any errors encountered during the creation of the stack will be listed in this tab.
91. Choose the **Resources** tab.

    The listing shows the resources that are being created. CloudFormation determines the optimal order for resources to be created, such as creating the VPC before the subnet.

    Task 4.4: View created resources from the console
92. Choose the **Resources** tab.

    The listing shows the resources that are being created. CloudFormation determines the optimal order for resources to be created, such as creating the VPC before the subnet.

    Was your prediction about which resources were created by the CloudFormation template correct? The new AWS CloudFormation stack should have launched a new Amazon VPC, and Amazon EC2 instance, *InstanceD*, along with the necessary supporting network pieces.
93. Select the **Outputs** tab.
94. Select the URL link for the *InstanceDPublicDNS*.
95. A new browser tab opens to the webpage hosted on *InstanceD*.

    Alternatively, you can copy and paste the Public DNS value into a new browser tab.

    Caution: You may need to wait for at least 3 minutes for the Amazon EC2 instance to finish health checks and for the web

server to initially start. Also, SSL/TLS was not setup for your Amazon EC2 instance in this lab. Attempting to navigate to the DNS address for the web server prefixed with HTTPS:// will fail. In your personal AWS environment, you can use CloudFormation signals in your templates to delay the CREATE_COMPLETE status of the stack until the instance passes all health checks.

Congratulations, you have launched a new CloudFormation stack, reviewed the resources it created, and verified connectivity to the new web server.

Conclusion

In this lab you did the following:

- Created Amazon EC2 instance using the AWS Management Console.
- Created Amazon EC2 instance using the AWS Command Line Interface.
- Created Amazon EC2 instance using the AWS CloudFormation Service.
- Deployed a basic web server to the Amazon EC2 instances and tested connectivity to each.

Lab Complete

Congratulations! You have completed the lab.

End Lab

Follow these steps to close the console, end your lab, and evaluate the experience.

96. Return to the AWS Management Console.
97. On the navigation bar, choose **awsstudent@<AccountNumber>**, and then choose **Sign Out**.
98. Choose End Lab
99. Choose OK
100. (Optional):

- Select the applicable number of stars
- Type a comment
- Choose **Submit**
- 1 star = Very dissatisfied
- 2 stars = Dissatisfied
- 3 stars = Neutral
- 4 stars = Satisfied
- 5 stars = Very satisfied

You may close the window if you don't want to provide feedback. For more information about AWS Training and Certification, see *https://aws.amazon.com/training/*.

Your feedback is welcome and appreciated.

If you would like to share any feedback, suggestions, or corrections, please provide the details in our *AWS Training and Certification Contact Form*.

Appendix

References

- AWS Systems Manager Session Manager documentation.

- The AWS CLI 'ec2 run-instances' command official documentation https://docs.aws.amazon.com/cli/latest/reference/ec2/run-instances.html.
- The Parameters section of CloudFormation templates is used to prompt for inputs that can be used elsewhere in the template.
- The Resources section of CloudFormation templates is used to define which AWS cloud resources to be deployed.

Viewing the lab templates in CloudFormation designer

Due to restraints unique to this lab environment, if you attempt to view any CloudFormation template using the CloudFormation designer tool, you may encounter object permissions errors. The following procedure should allow you a workaround to view the CloudFormation templates using the CloudFormation designer tool should you choose to do so.

To the left of these lab instructions you will find the region in which your lab was launched.

101. Copy the value from **LabRegion**
102. In the CloudFormation steps where you paste S3 URL's such as the following example:
103. Replace instances of *us-west-2* in the URL with the **LabRegion** value you copied from instructions instead.