**Launching Amazon EC2 Instances**

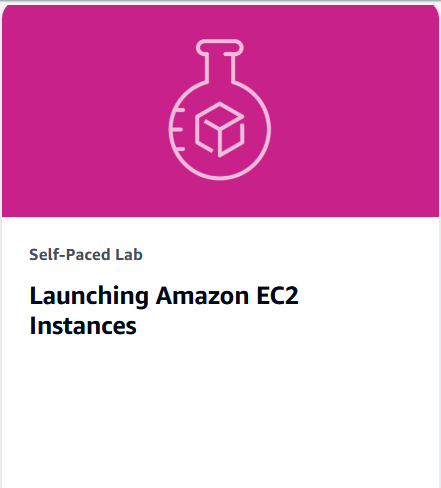
**Lab Topics**

* [Lab overview](https://labs.skillbuilder.aws/sa/lab/arn%3Aaws%3Alearningcontent%3Aus-east-1%3A470679935125%3Ablueprintversion%2FSPL-BE-200-CPBDEC-1%3A1.0.10-6501cc0b/en-US#lab-overview)
* [Start lab](https://labs.skillbuilder.aws/sa/lab/arn%3Aaws%3Alearningcontent%3Aus-east-1%3A470679935125%3Ablueprintversion%2FSPL-BE-200-CPBDEC-1%3A1.0.10-6501cc0b/en-US#start-lab)
* [Task 1: AWS Console login and Amazon EC2 console overview](https://labs.skillbuilder.aws/sa/lab/arn%3Aaws%3Alearningcontent%3Aus-east-1%3A470679935125%3Ablueprintversion%2FSPL-BE-200-CPBDEC-1%3A1.0.10-6501cc0b/en-US#task-1-aws-console-login-and-amazon-ec2-console-overview)
* [Task 2: Configure, Launch and Review the lifecycle of the an EC2 instance](https://labs.skillbuilder.aws/sa/lab/arn%3Aaws%3Alearningcontent%3Aus-east-1%3A470679935125%3Ablueprintversion%2FSPL-BE-200-CPBDEC-1%3A1.0.10-6501cc0b/en-US#task-2-configure-launch-and-review-the-lifecycle-of-the-an-ec2-instance)
* [Task 3: Launch an EC2 Instance with additional configuration and test User data implementation](https://labs.skillbuilder.aws/sa/lab/arn%3Aaws%3Alearningcontent%3Aus-east-1%3A470679935125%3Ablueprintversion%2FSPL-BE-200-CPBDEC-1%3A1.0.10-6501cc0b/en-US#task-3-launch-an-ec2-instance-with-additional-configuration-and-test-user-data-implementation)
* [Task 4: Access and Explore the EC2 Instance Virtual Machine](https://labs.skillbuilder.aws/sa/lab/arn%3Aaws%3Alearningcontent%3Aus-east-1%3A470679935125%3Ablueprintversion%2FSPL-BE-200-CPBDEC-1%3A1.0.10-6501cc0b/en-US#task-4-access-and-explore-the-ec2-instance-virtual-machine)
* [Task 5: Modify the attributes for EC2 Instance Virtual Machine](https://labs.skillbuilder.aws/sa/lab/arn%3Aaws%3Alearningcontent%3Aus-east-1%3A470679935125%3Ablueprintversion%2FSPL-BE-200-CPBDEC-1%3A1.0.10-6501cc0b/en-US#task-5-modify-the-attributes-for-ec2-instance-virtual-machine)
* [Task 6: Terminate and clean up the EC2 instance virtual machine](https://labs.skillbuilder.aws/sa/lab/arn%3Aaws%3Alearningcontent%3Aus-east-1%3A470679935125%3Ablueprintversion%2FSPL-BE-200-CPBDEC-1%3A1.0.10-6501cc0b/en-US#task-6-terminate-and-clean-up-the-ec2-instance-virtual-machine)
* [Conclusion](https://labs.skillbuilder.aws/sa/lab/arn%3Aaws%3Alearningcontent%3Aus-east-1%3A470679935125%3Ablueprintversion%2FSPL-BE-200-CPBDEC-1%3A1.0.10-6501cc0b/en-US#conclusion)
* [End lab](https://labs.skillbuilder.aws/sa/lab/arn%3Aaws%3Alearningcontent%3Aus-east-1%3A470679935125%3Ablueprintversion%2FSPL-BE-200-CPBDEC-1%3A1.0.10-6501cc0b/en-US#end-lab)

**Lab overview**

Amazon Elastic Compute Cloud (Amazon EC2) is a web service that provides secure, resizable compute capacity in the cloud. It is designed to make web-scale cloud computing easier for developers. Amazon EC2’s simple web service interface allows us to obtain and configure capacity with minimal friction. It provides us with complete control of our computing resources and lets we run on Amazon’s proven computing environment.

In this lab, we explore the selection and creation of appropriate Amazon EC2 virtual machine components, such as networking, storage, and security. We also get an opportunity to configure, access, and terminate an Amazon EC2 instance.



**Objectives**

By this lab, we will be able to do the following:

* Navigate the Amazon EC2 service console.
* Understand the lifecycle of an Amazon EC2 instance.
* Understand the main components of an Amazon EC2 instance and how to configure them.
* Launch and access an Amazon EC2 instance.
* Terminate an Amazon EC2 instance.

**Amazon EC2 Overview**

Amazon EC2 provides scalable computing capacity in the Amazon Web Services (AWS) Cloud. Amazon EC2 provides the following features:

* Virtual computing environments, known as instances.
* Preconfigured templates for our instances, known as Amazon Machine Images (AMIs), that package the bits we need for our server (including the operating system and additional software).
* Various configurations of CPU, memory, storage, and networking capacity for our instances, known as instance types.
* Secure login information for our instances using key pairs (AWS stores the public key, and we store the private key in a secure place).
* Storage volumes for temporary data that’s deleted when we stop, hibernate, or terminate our instance, known as instance store volumes.
* Persistent storage volumes for our data using Amazon Elastic Block Store (Amazon EBS), known as Amazon EBS volumes.
* Multiple physical locations for our resources, such as instances and Amazon EBS volumes, known as Regions and Availability Zones.
* A firewall that enables us to specify the protocols, ports, and source IP ranges that can reach our instances using security groups.
* Static IPv4 addresses for dynamic cloud computing, known as Elastic IP addresses.
* Metadata, known as tags that we can create and assign to our Amazon EC2 resources.
* Virtual networks we can create that are logically isolated from the rest of the AWS Cloud, and that we can optionally connect to our own network, known as virtual private clouds (VPCs).

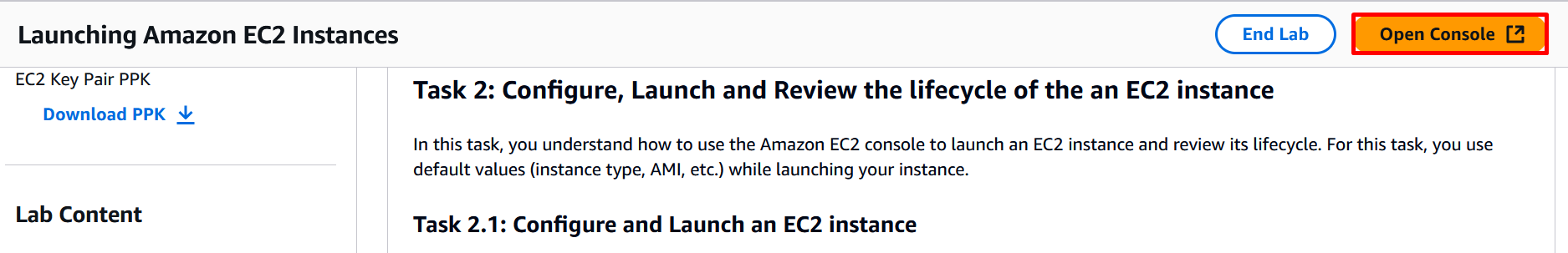
**Start lab**

1. To launch the lab, at the top of the page, click **Start Lab**.
2. To open the lab, click **Open Console**.

**Task 1: AWS Console login and Amazon EC2 console overview**

In this task, we login to the AWS Management Console and review different components of the Amazon EC2 Service Console.

1. Log in to the AWS Management Console using the instructions provided in the **Start lab** section.

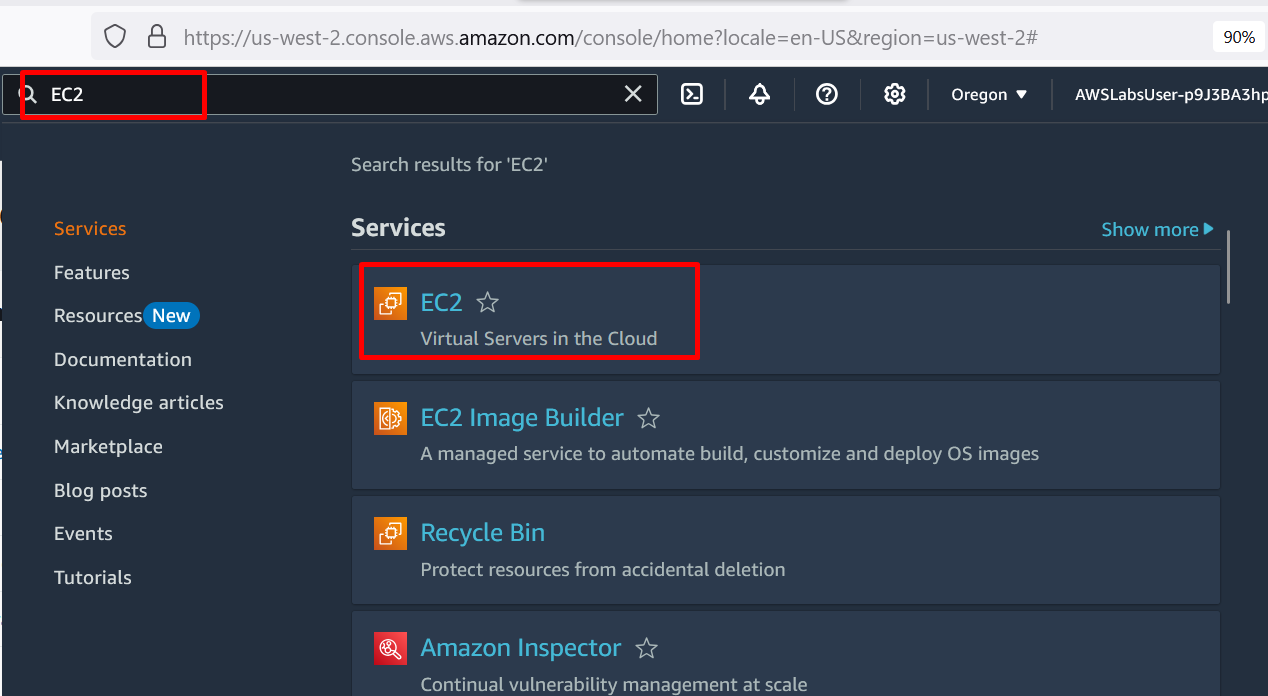


**Task 2: Configure, Launch and Review the lifecycle of an EC2 instance**

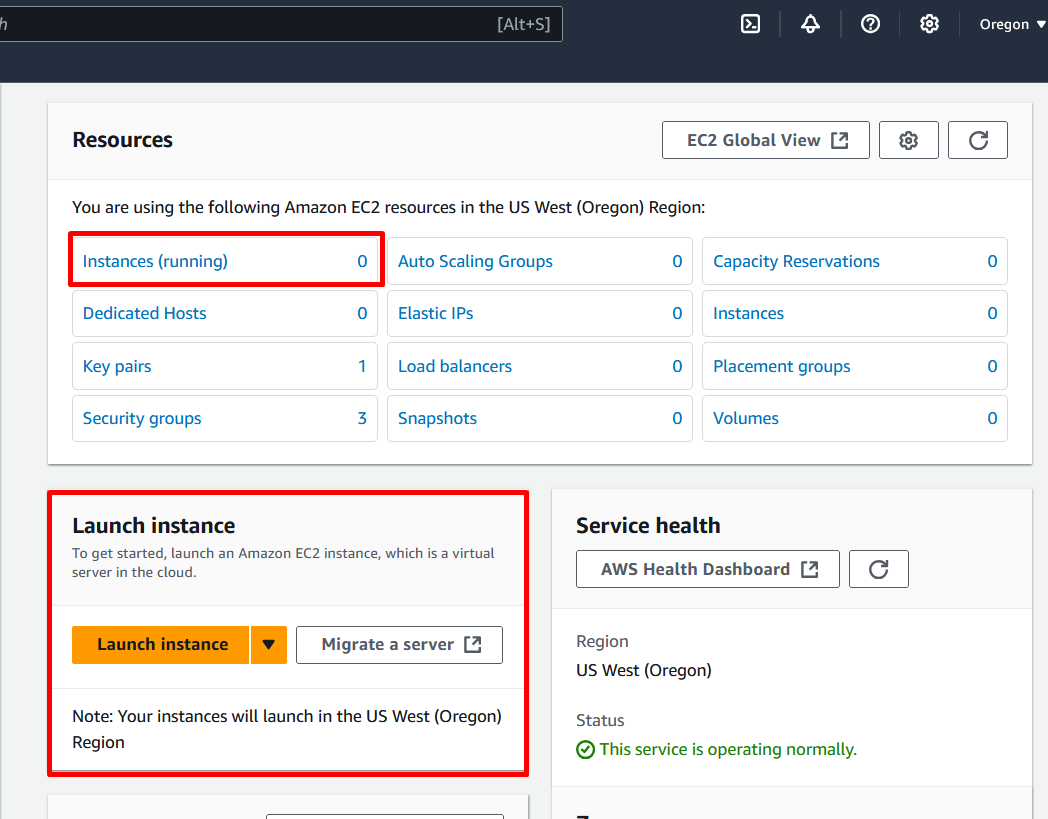
In this task, we understand how to use the Amazon EC2 console to launch an EC2 instance and review its lifecycle. For this task, we use default values (instance type, AMI, etc.) while launching our instance.

**Task 2.1: Configure and Launch an EC2 instance**

1. Navigate back to the EC2 console by choosing the **EC2 Dashboard** option from the left navigation menu.

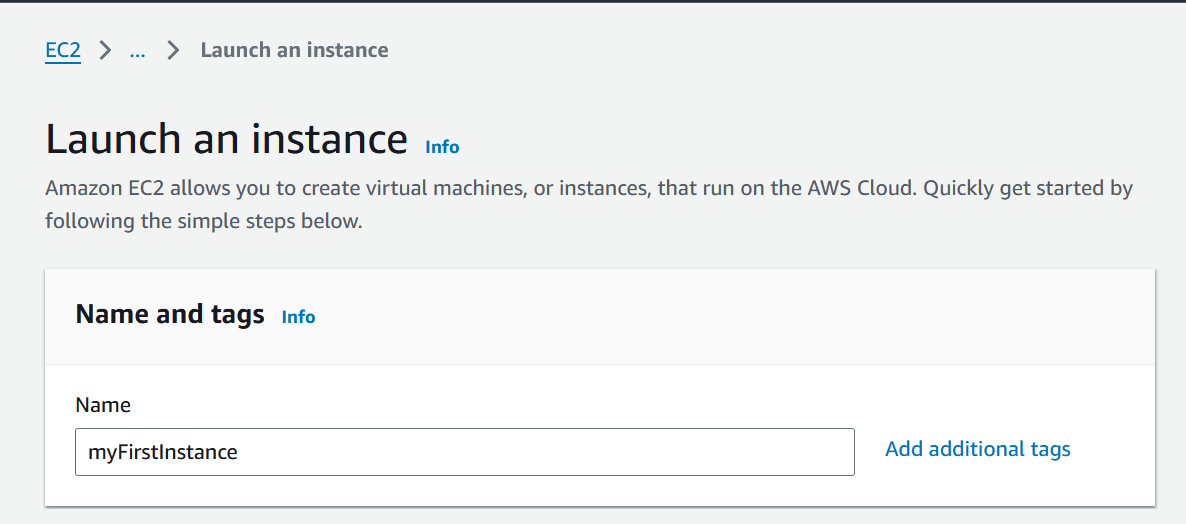


1. On the EC2 Dashboard, choose **Launch instances** and then click **Launch instances**.

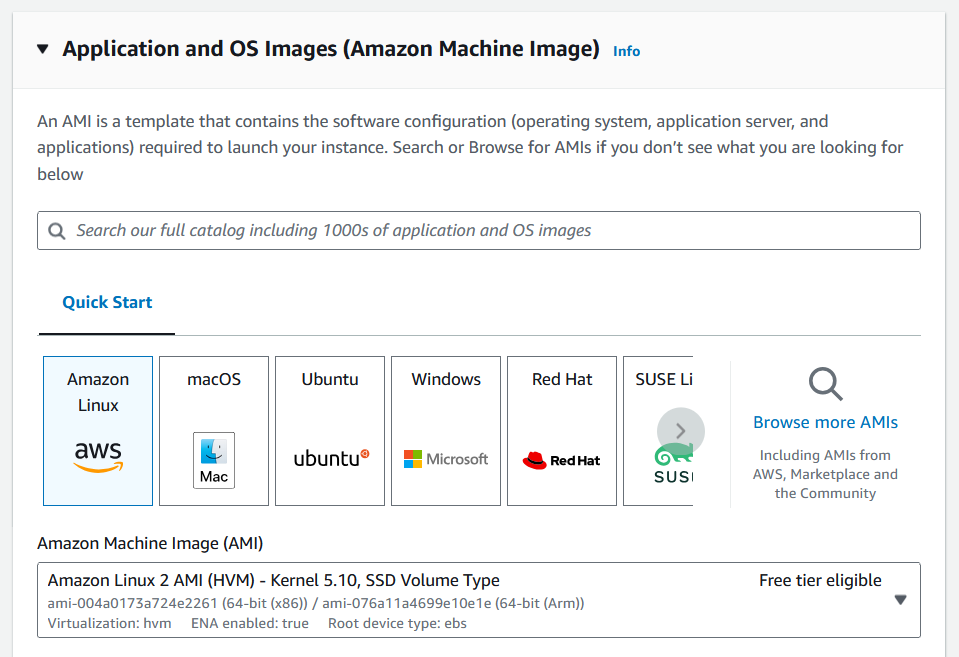


On the **Launch an instance** page, make the following selections:

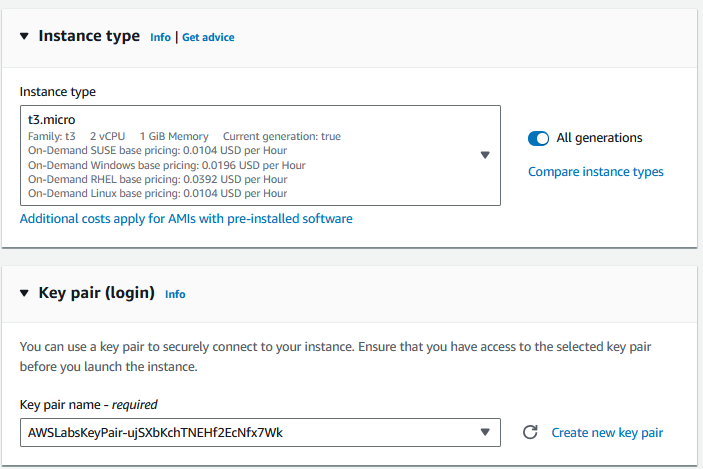
1. In the **Names and tags** section, locate the **Name** textbox and enter **myFirstInstance**.



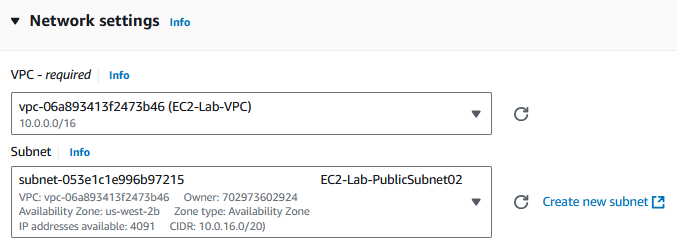
1. In the **Application and OS Images** section, locate the **AMI Machine Image (AMI)** dropdown menu and select the image name that starts with **Amazon Linux 2 AMI (HVM)**.

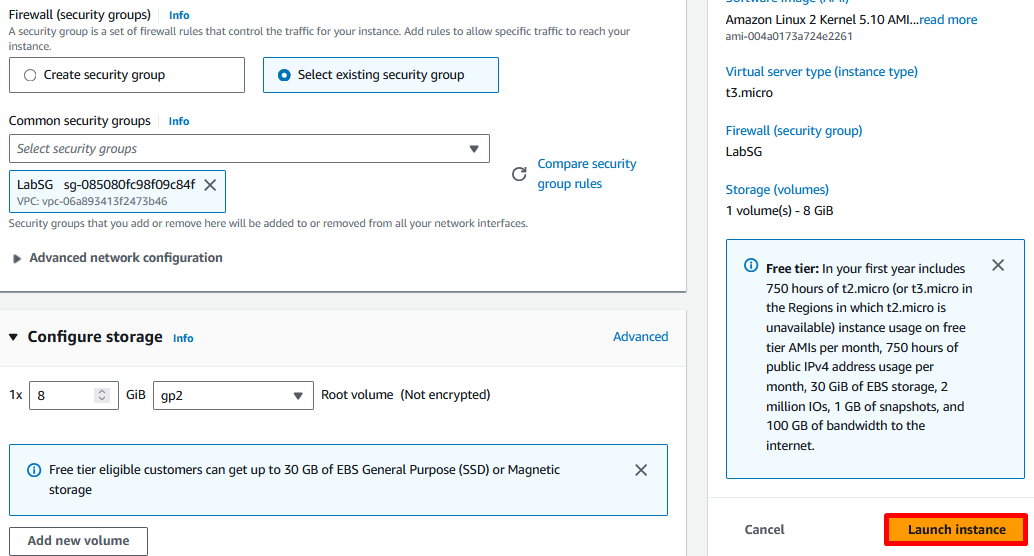


1. In the **Instance type** section, locate the **Instance type** dropdown menu and select **t3.micro** if not already selected.



1. In the **Key pair (login)** section, locate the **Key pair name** dropdown menu and choose an existing key pair (e.g. AWSLabsKeyPair-xxx) that was already created for this lab.
2. In the **Network settings** section, we see that the default VPC is already selected.

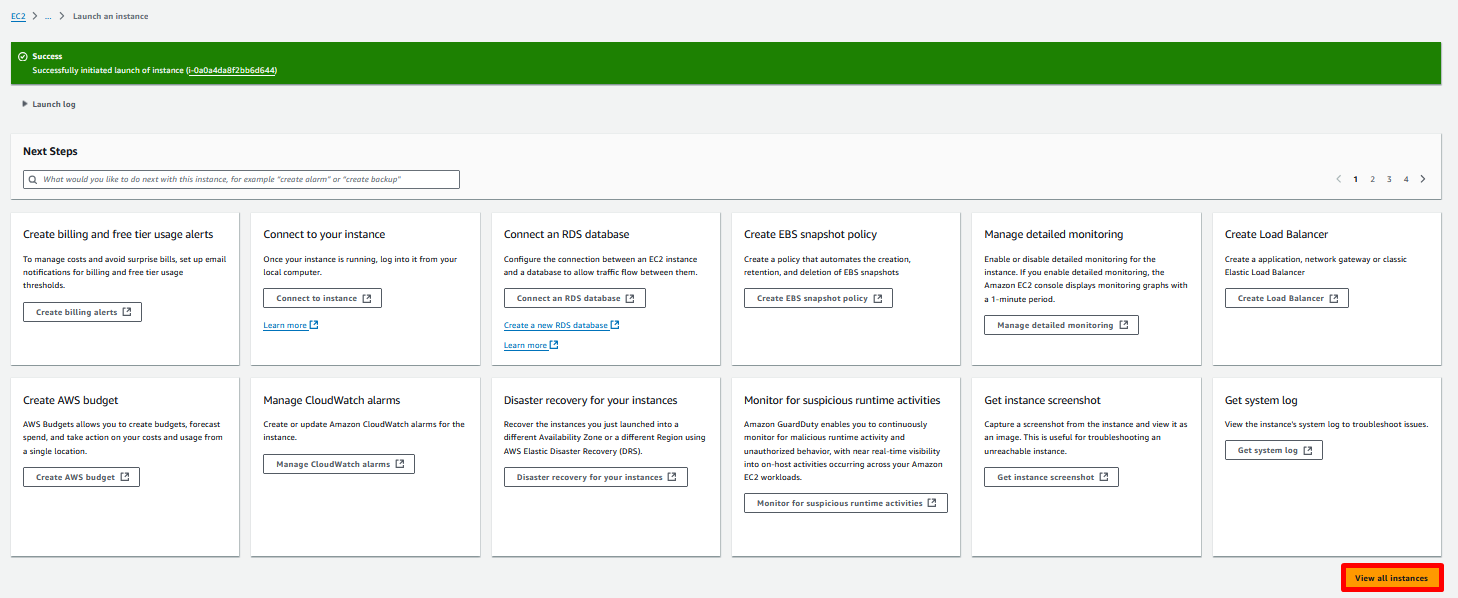




*Rules with source of 0.0.0.0/0 allow all IP addresses to access our instance. We recommend setting security group rules to allow access from known IP addresses only.*

1. Keep default values for remaining properties and click **Launch instance**.

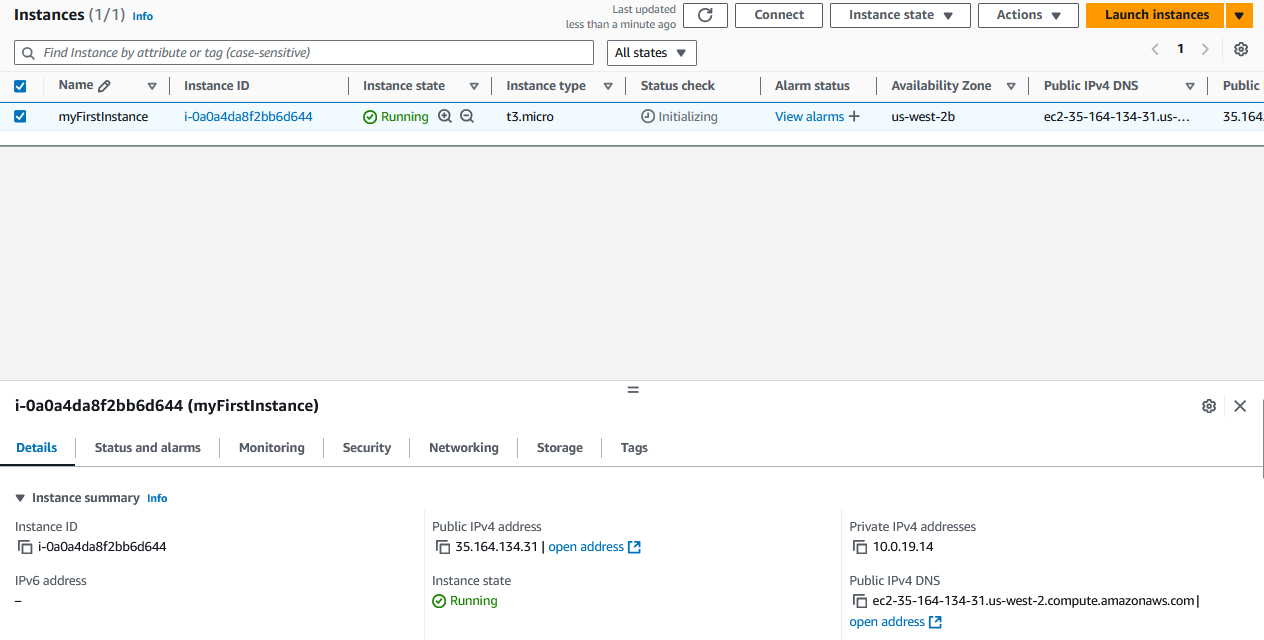
After successful launch, we see the following message:



**Task 2.2: Review EC2 Instance States**

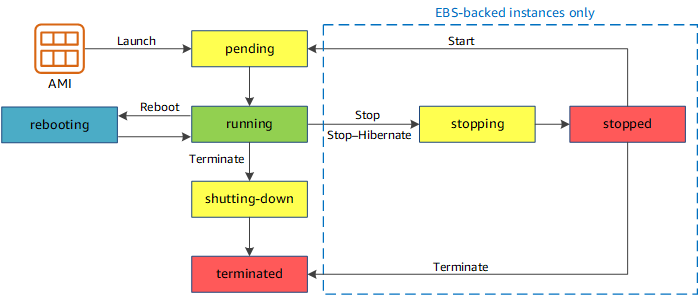
1. Choose and click **View all instances**.

At this point, we are re-directed to the **Instances** page where we see additional information regarding the EC2 instance including the Name, Instance ID, Instance state and Instance type.

  
  
 Looking at the **Instance state** column, we see that the EC2 instance state will be in **running** state.

**Noted:** When we launch an instance, it enters the *Pending* state. The instance type that we specified at launch determines the hardware of the host computer for our instance. An Amazon Machine Image (AMI) is used at launch to boot the instance. After the instance is ready, it enters the *Running* state and we can connect to our running instance and use it the way that we use a computer sitting in front of us.

The following illustration represents the transitions between instance states:

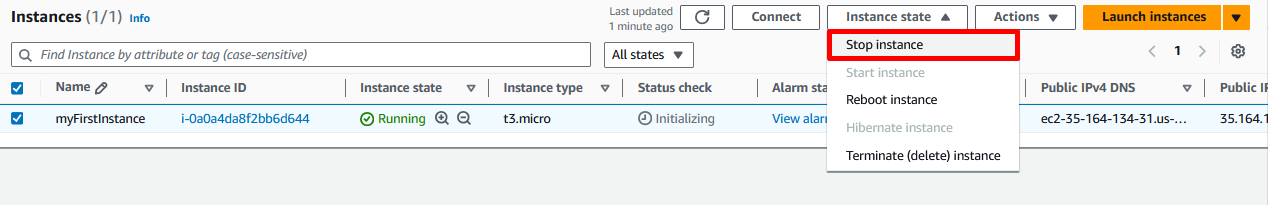


1. On the **Instances** page, select **myFirstInstance** and choose **Instance state**.

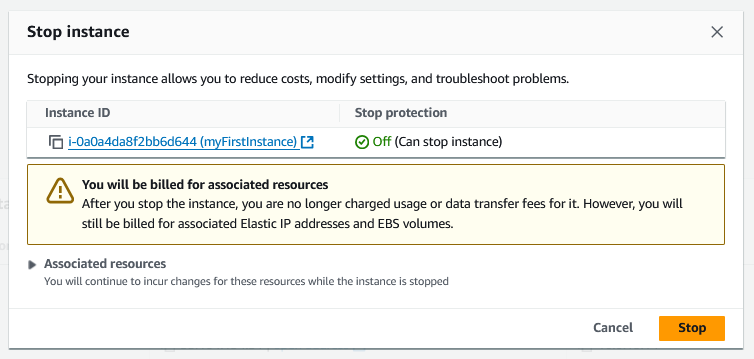
Here, we see different options like:

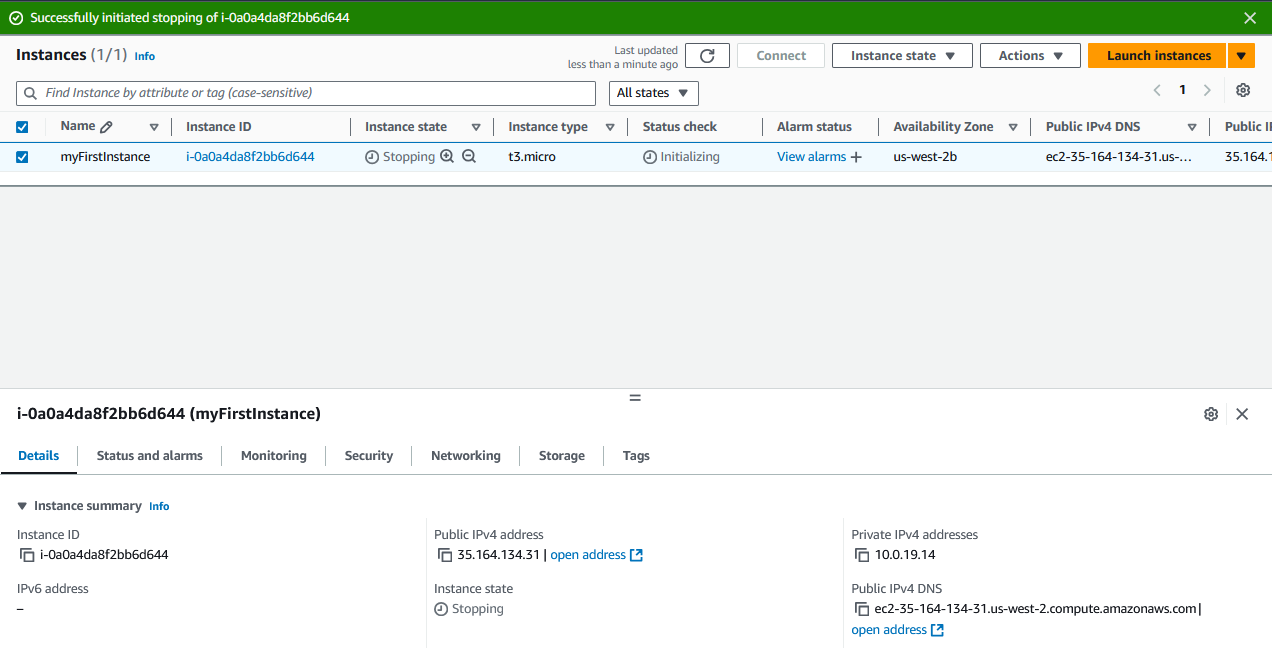
* Stop instance
* Start instance
* Reboot instance
* Hibernate instance
* Terminate instance
* **Instance Stop/Start:** When we stop our instance, it enters the *Stopping* state, and then the *Stopped* state. When we stop and start our instance, we lose any data on the instance store volumes on the previous host computer. However, the instance retains its private IPv4 address, which means that an Elastic IP address associated with the private IPv4 address or network interface is still associated with our instance. If our instance has an IPv6 address, it retains its IPv6 address as well.
* **Instance Reboot:** We can also reboot our instance using the Amazon EC2 console. Rebooting an instance is equivalent to rebooting an operating system. The instance remains on the same host computer and maintains its public DNS name, private IP address, and any data on its instance store volumes. It typically takes a few minutes for the reboot to complete, but the time it takes to reboot depends on the instance configuration.
* **Instance Hibernate:** When we hibernate an instance, we signal the operating system to perform hibernation (suspend-to-disk), which saves the contents from the instance memory (RAM) to our Amazon EBS root volume. We persist the instance’s Amazon EBS root volume and any attached Amazon EBS data volumes. When we start our instance, the Amazon EBS root volume is restored to its previous state and the RAM contents are reloaded. Previously attached data volumes are reattached and the instance retains its instance ID. Our instance also retains its private IPv4 address, which means that an Elastic IP address associated with the private IPv4 address or network interface is still associated with our instance. If our instance has an IPv6 address, it retains its IPv6 address.
* **Instance termination:** When we have decided that we no longer need an instance, we can terminate it. As soon as the status of an instance changes to shutting-down or terminated, we stop incurring charges for that instance. After we terminate an instance, it remains visible in the console for a short while, and then the entry is automatically deleted.

1. From the **Instance state** dropdown menu, click **Stop instance**.



1. On the **Stop instance**? Pop-up window, click **Stop**.



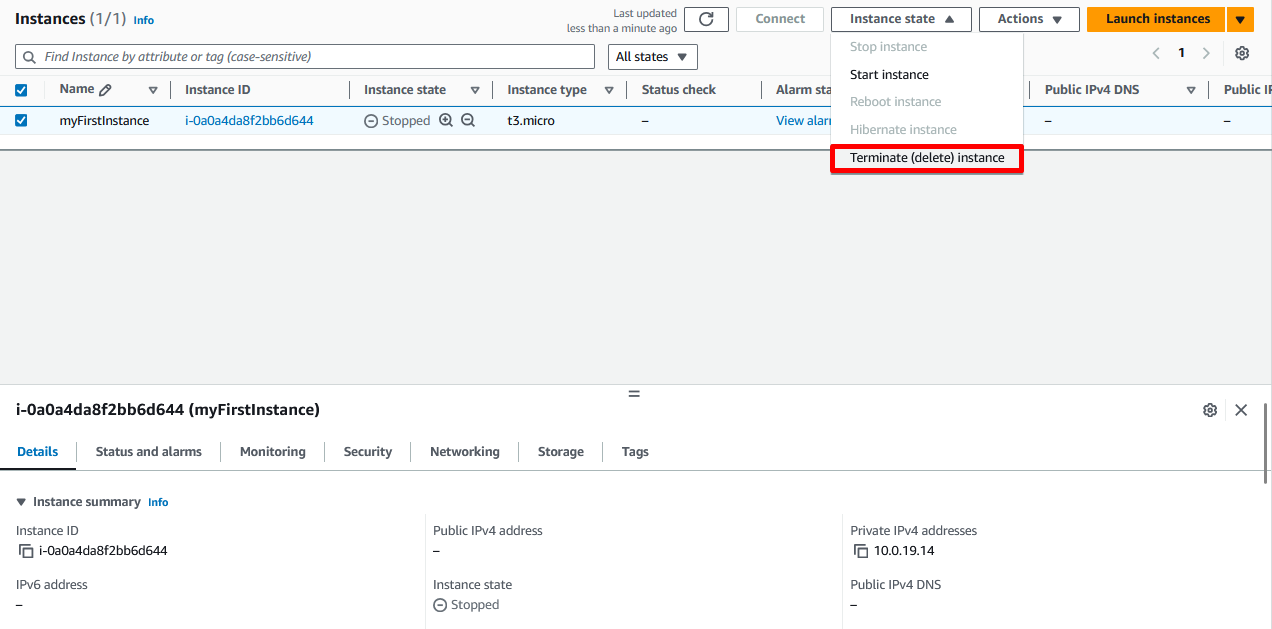


A Successfully initiated stopping message is displayed on top of the screen.

**Task 2.3: Review EC2 Instance Summary**

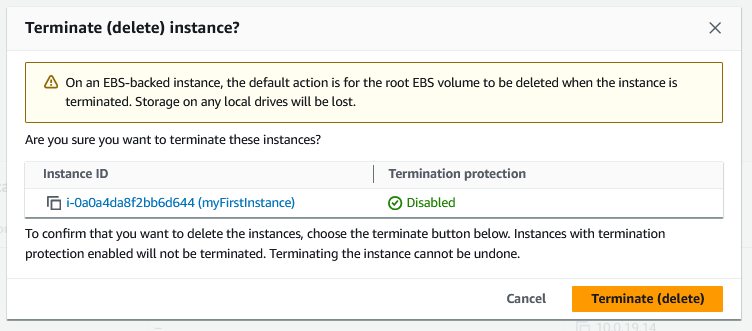
* + 1. On the **Instances** page, select **myFirstInstance**, choose **Actions** and then click **View details**.

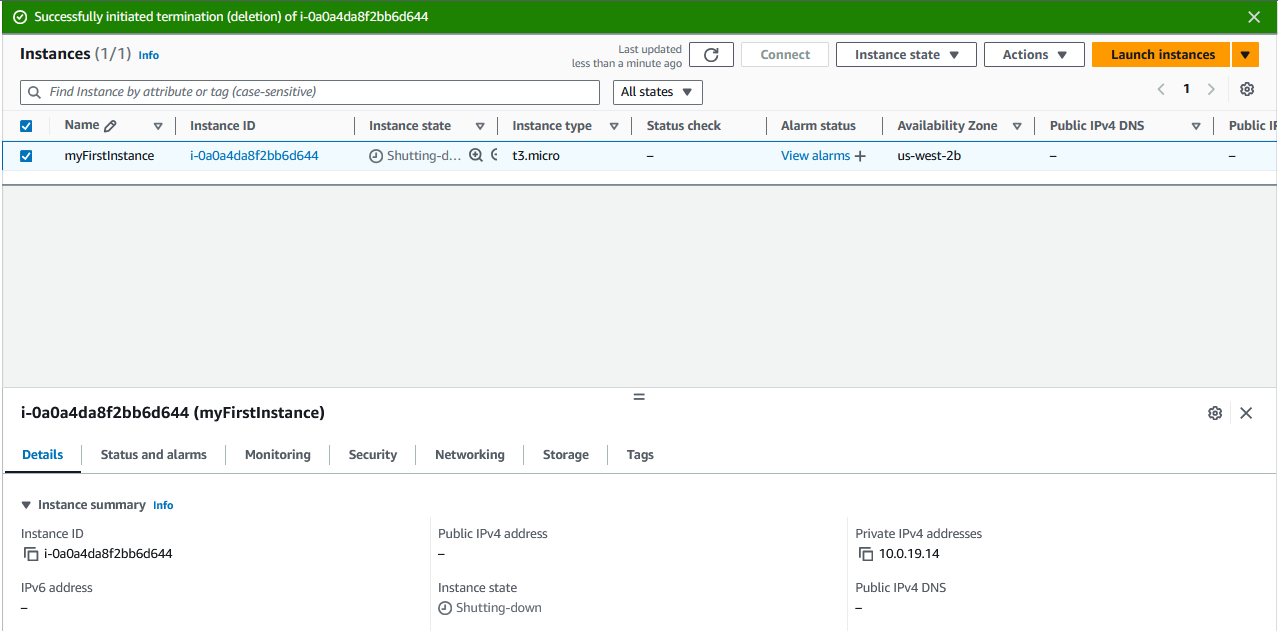
Here we see the detailed summary of our EC2 instance like the Instance ID, Public and Private IPv4 addresses, Instance state, Instance type and many more. We also review details based on different categories by choosing individual tabs for Security, Networking, Storage, Status checks, Monitoring and Tags.



**Task 2.4: Terminate the EC2 Instance**

1. On the **Instance summary for myFirstInstance** page, choose **Instance state** and then click **Terminate instance**.
2. On the **Terminate instance?** Pop-up window, click **Terminate (delete)**.



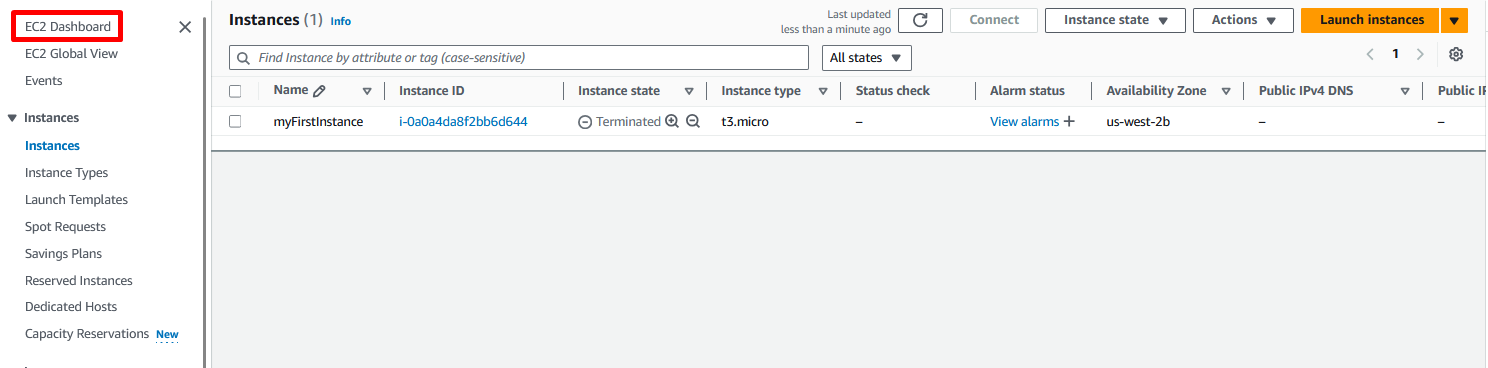


A Successfully initiated termination (deletion) message is displayed on top of the screen.

**Congratulations!** We have successfully demonstrated the launch and termination of our first EC2 instance.

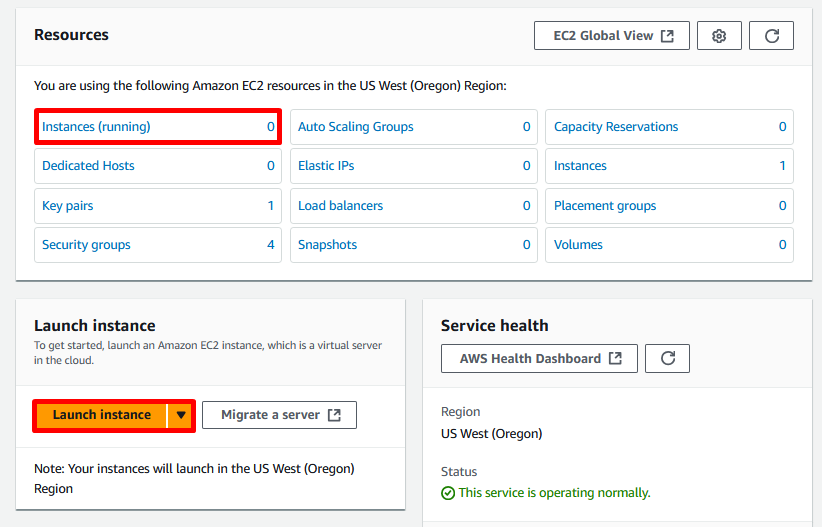
**Task 3: Launch an EC2 Instance with additional configuration and test User data implementation**

In this task, we launch another Amazon EC2 instance with additional configuration options and deploy a sample PHP application using the user data section.



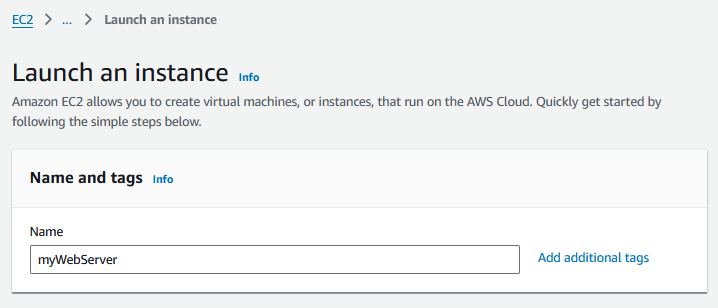
**Task 3.1: Launch an EC2 Instance and deploy sample PHP application using the User data section**

1. Navigate back to the EC2 console by choosing **EC2 Dashboard** from the left navigation menu.
2. On the EC2 Dashboard, choose **Launch instances** and then click **Launch instances**.

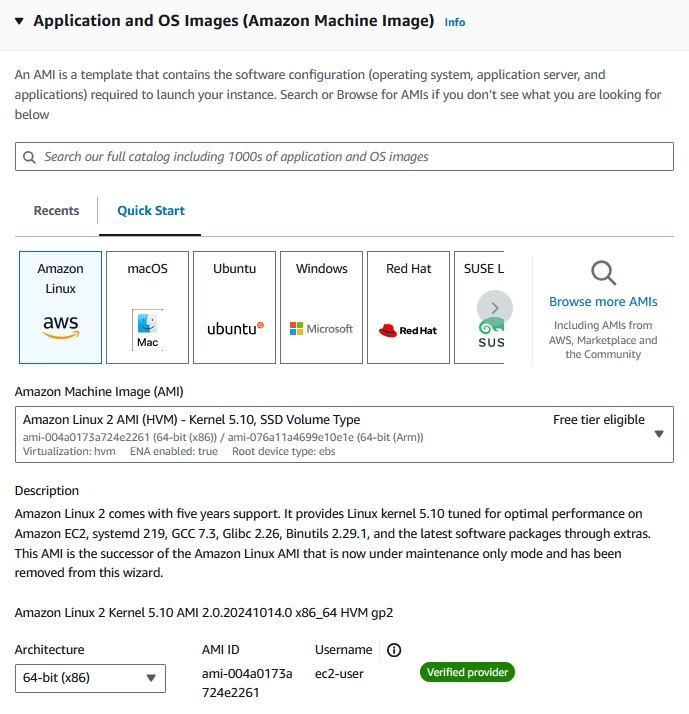


On the **Launch an instance** page, make the following selections:

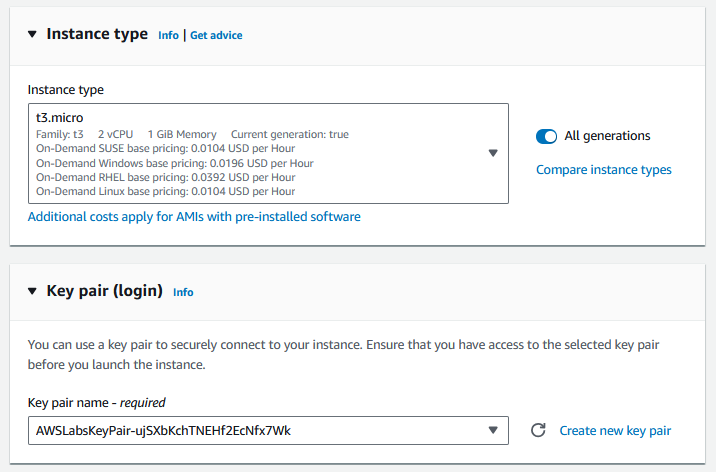
1. In the **Names and tags** section, locate the **Name** textbox and enter **myWebServer**.



1. Locate the **Resource type’s** dropdown menu, and select **Volumes** and **Network interfaces**. Select **Instances** if not already selected.
2. In the **Application and OS Images** section, locate the **AMI Machine Image (AMI)** dropdown menu and select the image name that starts with **Amazon Linux 2 AMI (HVM)**.

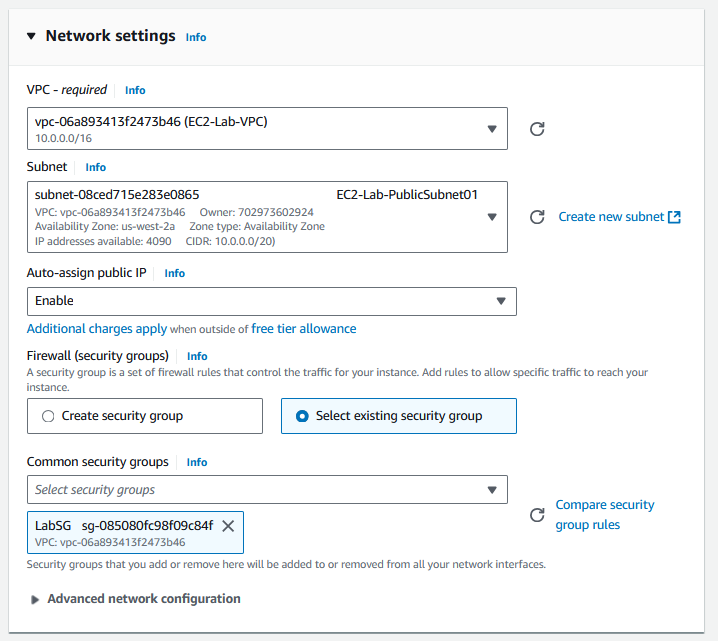


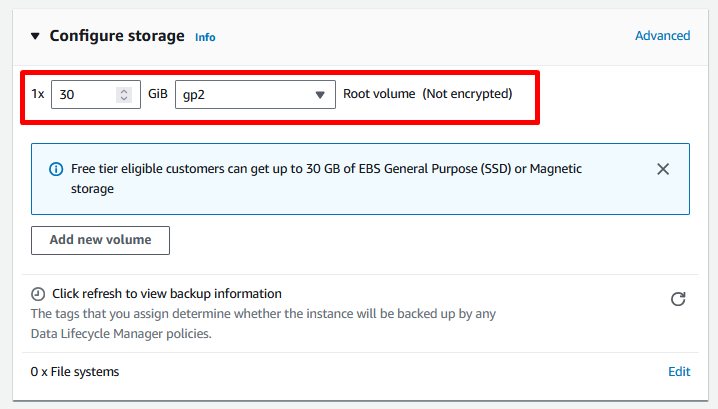
1. In the **Instance type** section, locate the **Instance type** dropdown menu and select **t3.micro**.
2. In the **Key pair (login)** section, locate the **Key pair name** dropdown menu and choose an existing keypair (e.g. AWSLabsKeyPair-xxx) that was already created for this lab.



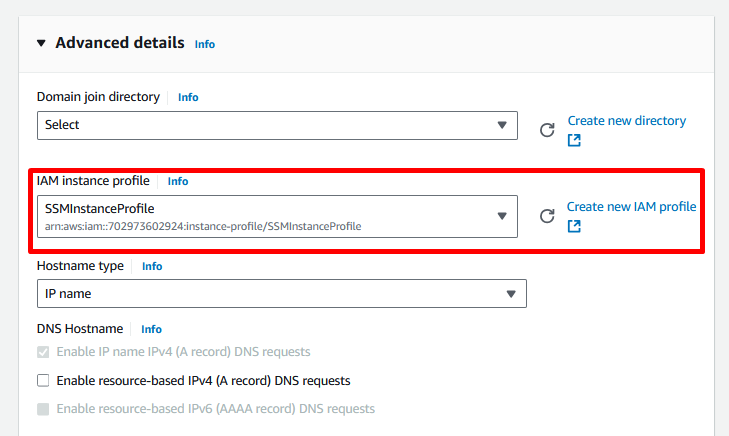
1. In the **Network settings** section, click **Edit** and make the following selections:

* Locate the **VPC** dropdown menu and select **EC2-Lab-VPC**.
* Locate the **Subnet** dropdown menu and select **EC2-Lab-PublicSubnet01**.
* For **Firewall**, select **Select existing security group** and then select **LabSG** from the **Common security group’s** dropdown menu.



1. In the **Configure storage** section, increase the size of root volume to **30** GiB.   
     
    
2. Expand the **Advanced details** section and configure the following:

* Locate the **IAM instance profile** dropdown menu and choose **SSMInstanceProfile**.



* Scroll down to the **User data** textbox and enter the following code:

#!/bin/bash

yum update -y

amazon-linux-extras install -y lamp-mariadb10.2-php7.2 php7.2

yum install -y httpd mariadb-server

systemctl start httpd

systemctl enable httpd

usermod -a -G apache ec2-user

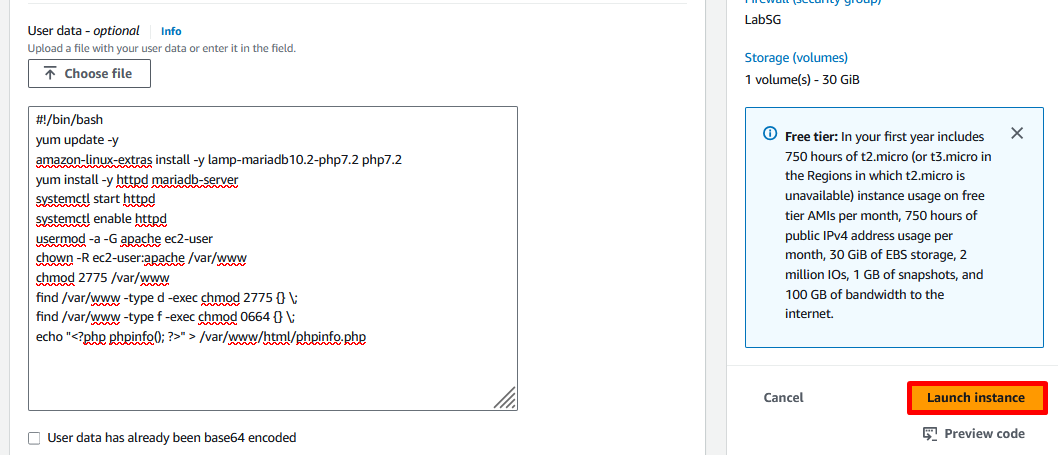
chown -R ec2-user:apache /var/www

chmod 2775 /var/www

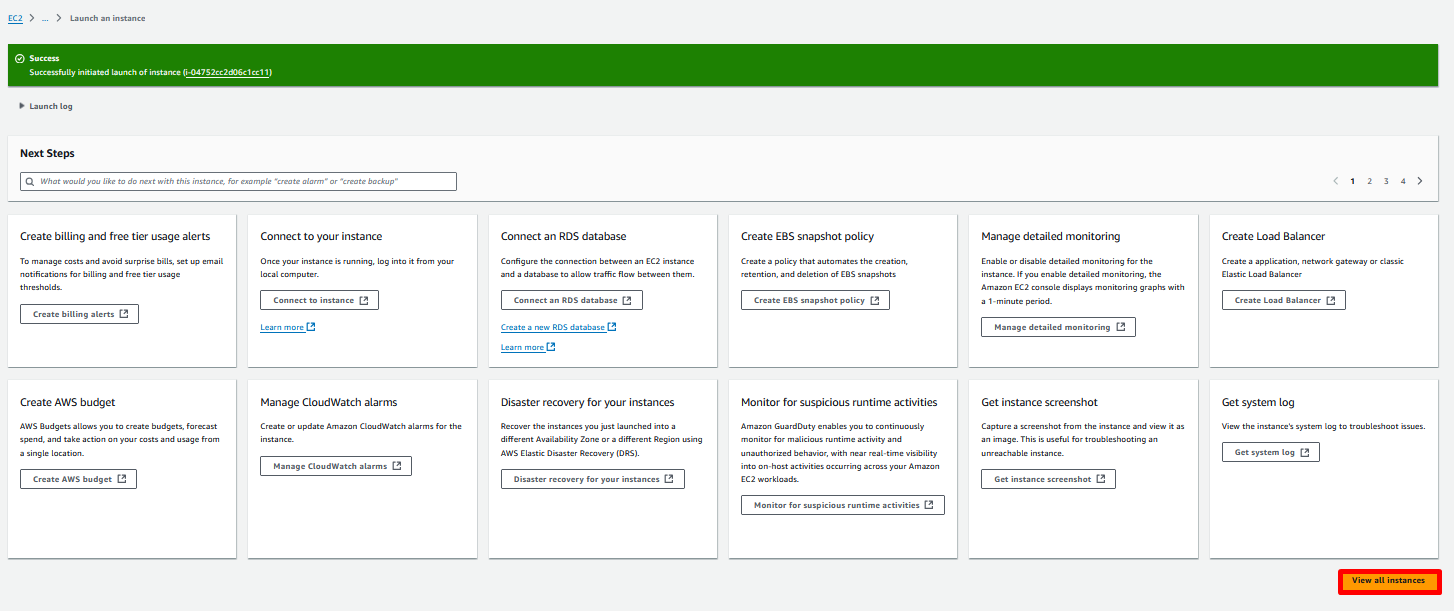
find /var/www -type d -exec chmod 2775 {} \;

find /var/www -type f -exec chmod 0664 {} \;

echo "<?php phpinfo(); ?>" > /var/www/html/phpinfo.php

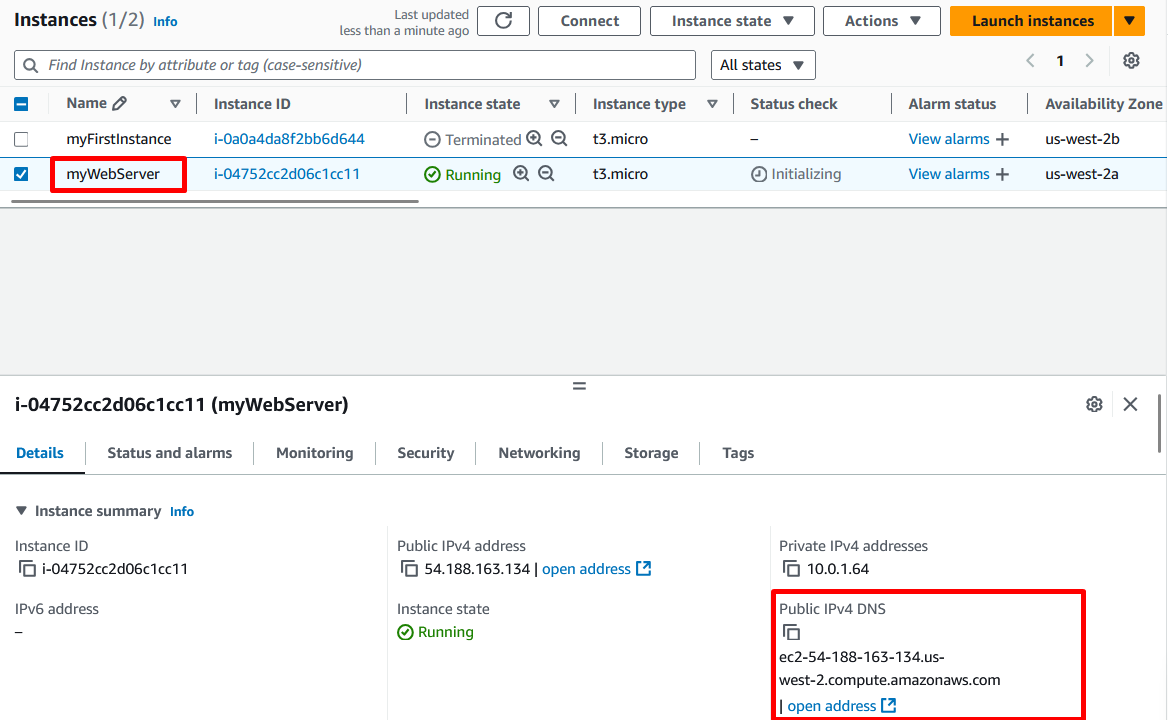


1. Keep default values for remaining properties and click **Launch instance**.

A Successfully initiated launch of instance message is displayed.  
  


1. Choose and click **View all instances**.

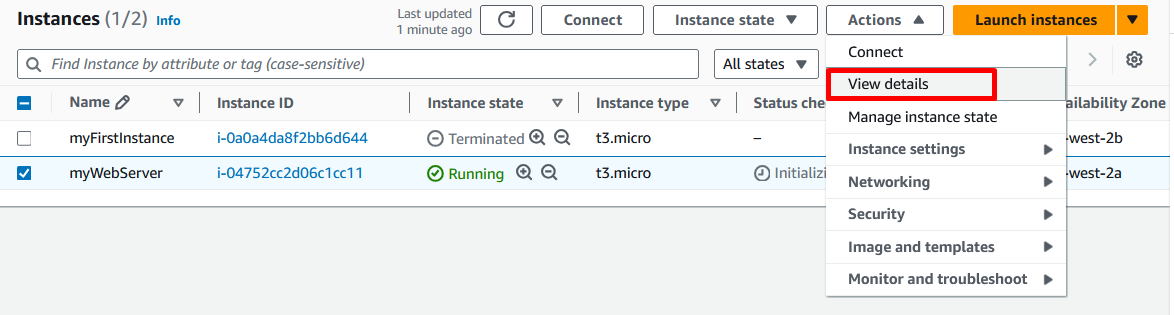
It might take few minutes for the **myWebServer** instance to finish its status checks. We can confirm that by checking the **Status check** column where it shows Initializing and eventually moves to 2/2 checks passed. Use the refresh option as required.



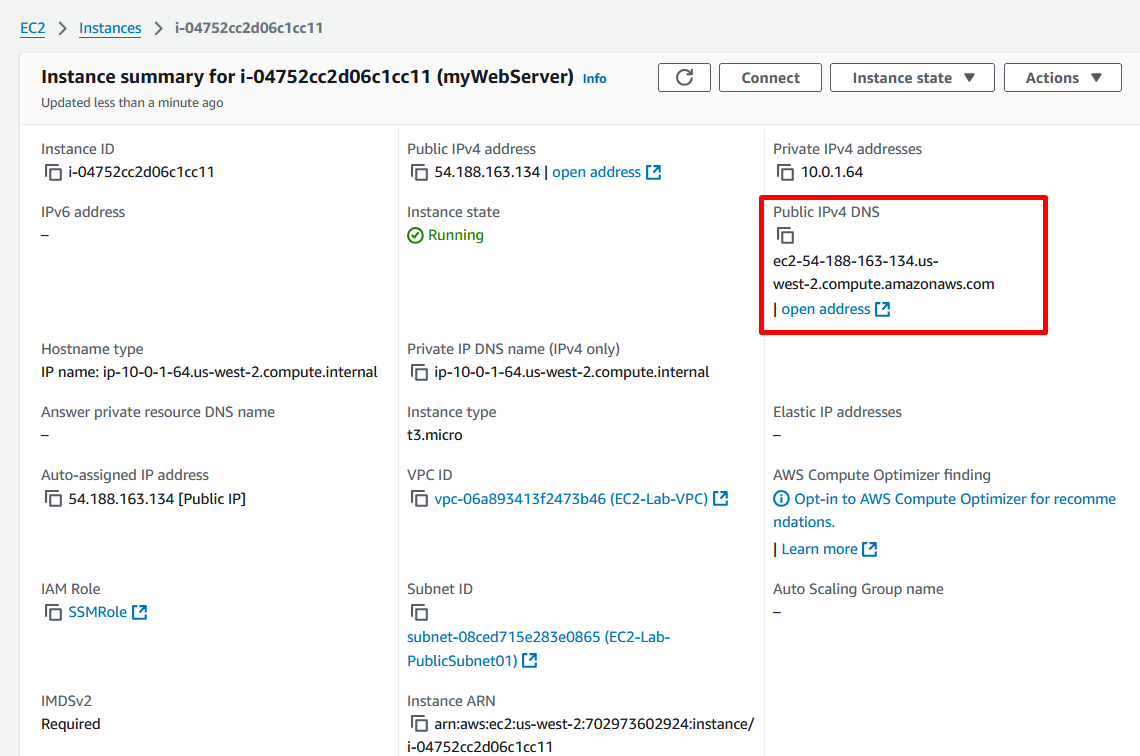
**Task 3.2: Test User data implementation and verify our PHP application using EC2 instance URL**

In this task, we test the user data implementation and verify the PHP application installation.

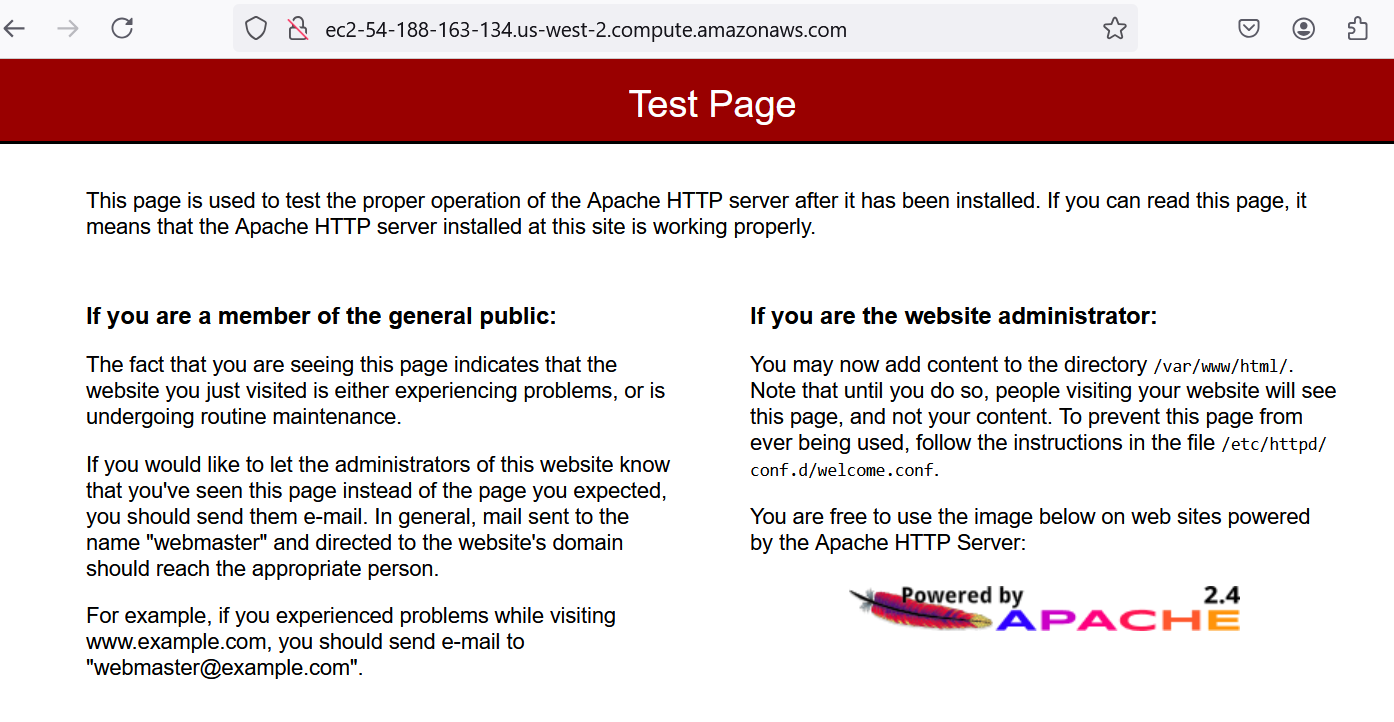
* 1. On the **Instances** page, select **myWebServer** instance, choose **Actions** and then click **View details**.



* 1. On the **Instance summary for myWebServer** page, copy the value of **Public IPv4 DNS** and paste it into a new browser tab. We can also use **Public IPv4 address** instead of **Public IPv4 DNS**.



At this point, we see the following Test Page for our sample PHP application.



This confirms that the user data implementation was successful.

**Congratulations!** We have successfully launched an Amazon EC2 instance with additional configuration options and deployed a sample PHP application using the user data section.

**Task 4: Access and Explore the EC2 Instance Virtual Machine**

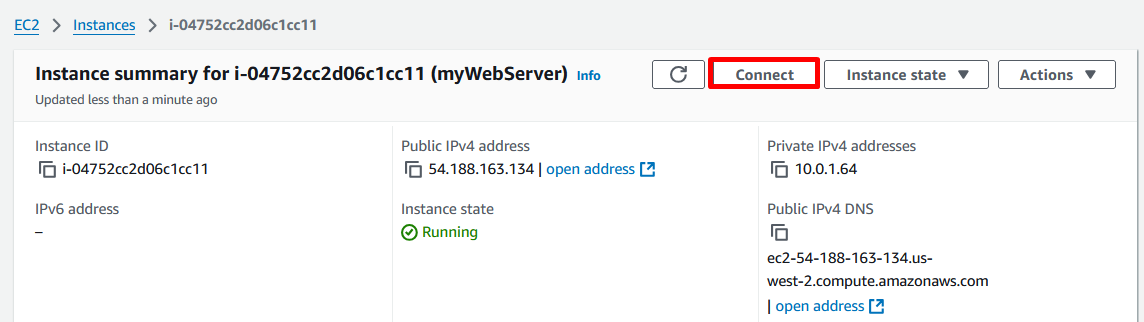
In this task, we understand how to access an Amazon EC2 instance. For this lab, we use Session Manager to connect to the **myWebServer** instance.

Session Manager is a fully managed AWS Systems Manager capability that lets we manage our Amazon EC2 instances through an interactive one-click browser-based shell or through the AWS CLI. We can use Session Manager to start a session with an instance in our account. After the session is started, we can run bash commands as we would through any other connection type.

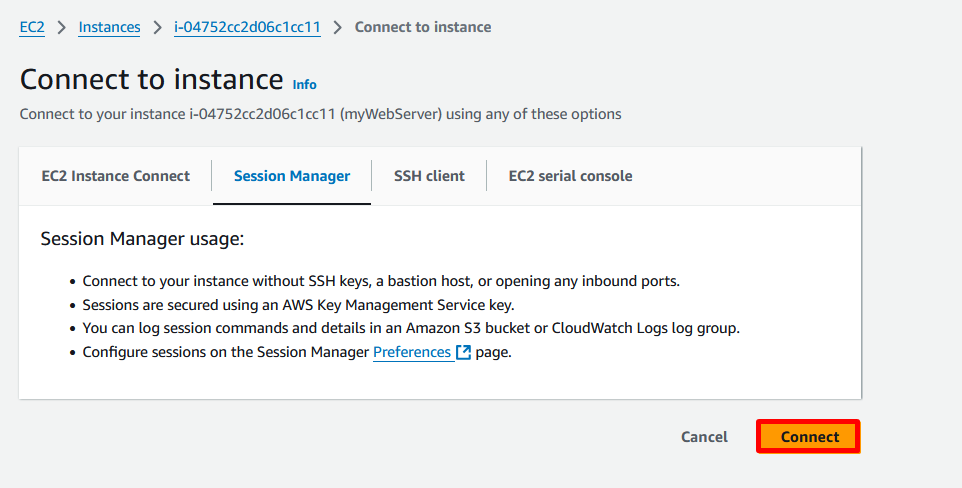
**Task 4.1: Connect to the EC2 Instance**

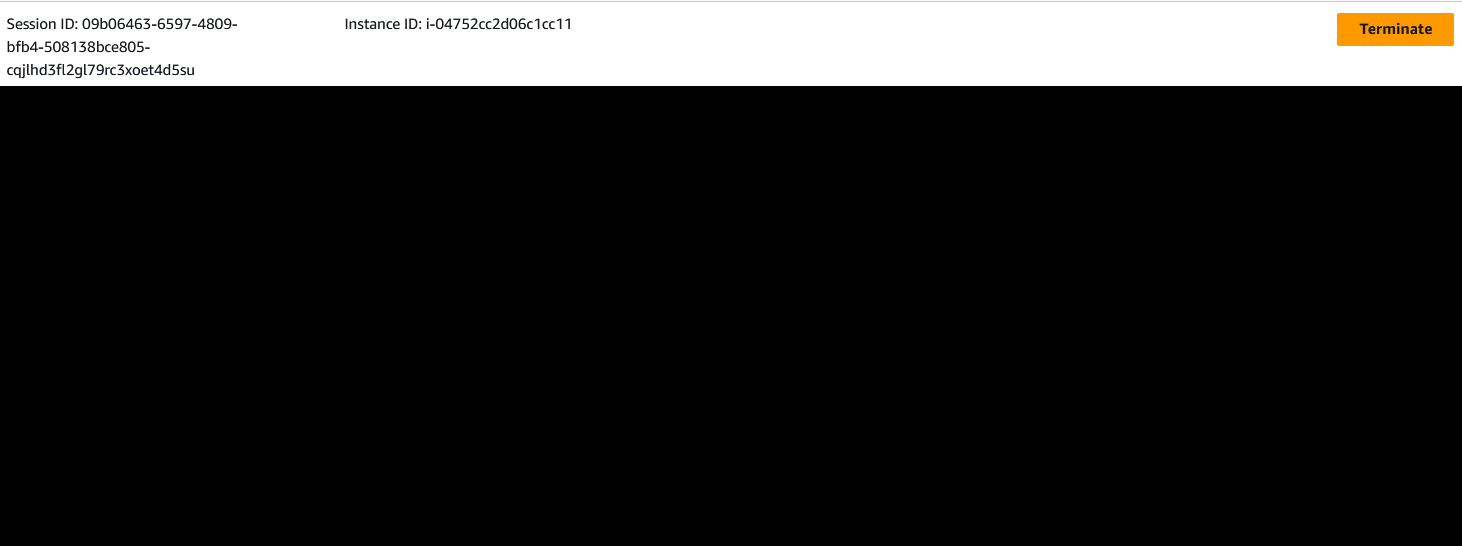
In this task, we connect to the **myWebServer** instance using Session Manager.

1. Return to the EC2 Management Console browser tab.
2. On the **Instance summary for myWebServer** page, click **Connect**.



1. Select the **Session Manager** tab if not already selected, and click **Connect**.



After successful connection, a new browser window opens up to the following terminal session:  
  


**Task 4.2: Run yum update on the instance**

In this task, we explore the **myWebServer** instance and perform basic tasks using the terminal session.

1. To update all the presently installed packages to their latest versions, enter the following command:

sudo yum update -y

**Expected output:**

Loaded plugins: extras\_suggestions, langpacks, priorities, update-motd

No packages marked for update

**Task 4.3: Explore the public keypair that was used while launching the EC2 instance**

When we launch an instance, we are prompted for a key pair. If we plan to connect to the instance using SSH, we must specify a key pair. We can choose an existing key pair or create a new one.

When our instance boots for the first time, the public key that we specified at launch is placed on our Linux instance in an entry within the *~/.ssh/authorized\_keys* file. When we connect to our Linux instance using SSH, to log in we must specify the private key that corresponds to the public key.

In this task, we explore the keypair that was already created as part of the initial lab setup.

1. To switch to the root user, enter the following command:

sudo su

**Expected output:**

*None, unless there is an error.*

1. To navigate to the *.ssh* directory and list its contents, enter the following command:

cd /home/ec2-user/.ssh/ && ls

**Expected output:**

authorized\_keys

1. To explore the created public keypair for our EC2 instance in the *authorized\_keys* file, enter the following command:

cat authorized\_keys

**Expected output:**

ssh-rsa AAAAB3NzaC1yc2EAAAADAQABAAABAQCmr3XiE7bf8lQQOE8ntvotkhq2aDIAhwvkHrq11+XR9SzaAhj9CK15JAmH/4FwsbbUB03d9Wc04gKl7uN8rzuKC/oQAsbkCrsEZKaYsNL+NcDM6yjpvZgM7pXIvPrJDUPXjgZJ1OZlWfmriuKQesJcXiES4ZUwUh7Kj9RMKDAxIrzfZLbquDPNgOGYdXglq4s/nMDd6

hYQQ/HKQ55eZJmTY3tsRwEdtIc5jUhg8nPO4VNPfp86rUA25+o74PbbujmwE4p/XnpQez8TP0ftKvZSwR0HDl0ELAVPoPB4P8sg3gdfyt3o2EHhJXXWiqcZ2gNZjiHjTE6SC+yESVRPLVQj AWSLabsKeyPair-jHF5HuP4curuTBRWD5Zxzp

**Task 4.4: Navigate to the /var/www/html directory and review the phpinfo.php file**

In this task, we navigate to the */var/www/html* directory and explore the PHP application file that was created as part of the user data installation process.

1. To navigate to the */var/www/html* directory and list it contents, enter the following command:

cd /var/www/html/ && ls

**Expected output:**

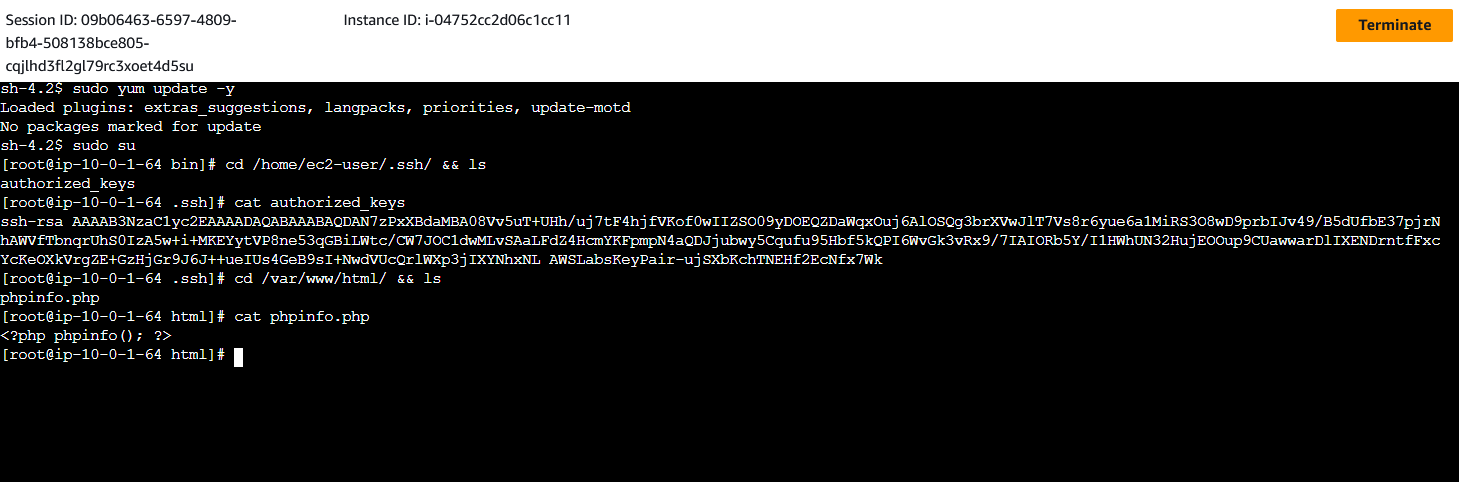
phpinfo.php

1. To view the contents of the *phpinfo.php* file, enter the following command:

cat phpinfo.php

**Expected output:**

<?php phpinfo(); ?>

The **phpinfo.php** file was referenced by our application running on the **myWebServer** instance when we tested the application URL in the previous task.  
  
All the above commands are given in a terminal.  
  


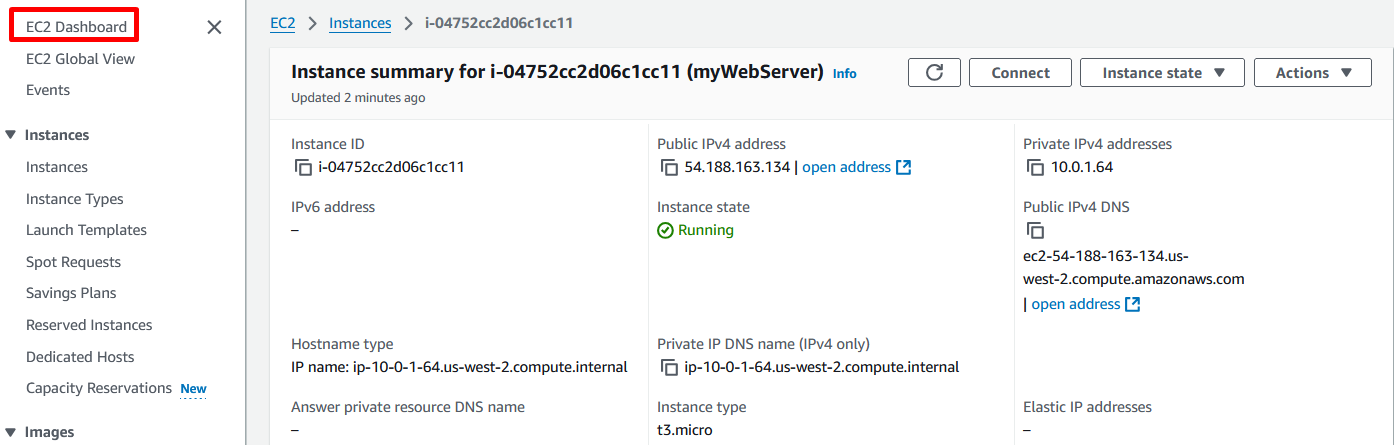
**Congratulations!** We have successfully accessed an Amazon EC2 instance using Session Manager and explored the instance by performing basic tasks using the terminal session.

**Task 5: Modify the attributes for EC2 Instance Virtual Machine**

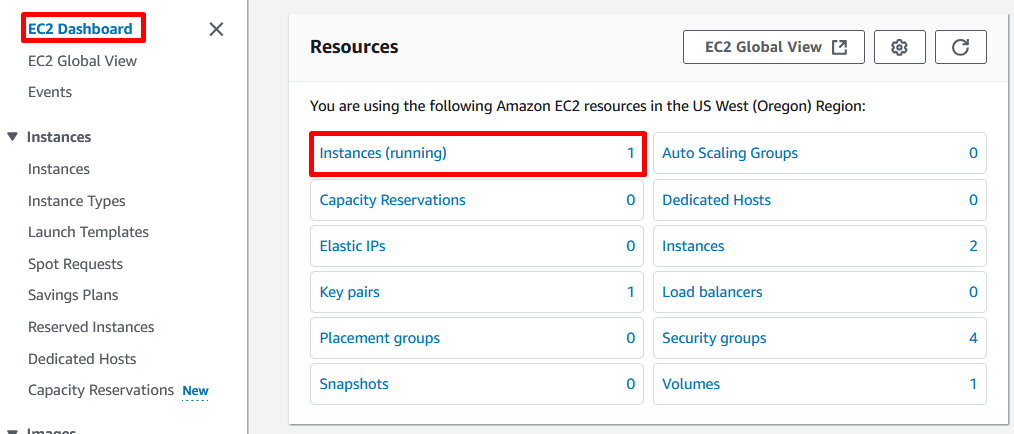
In this task, we modify attributes for the **myWebServer** instance and re-test our application.

**Task 5.1: Stop the EC2 Instance**

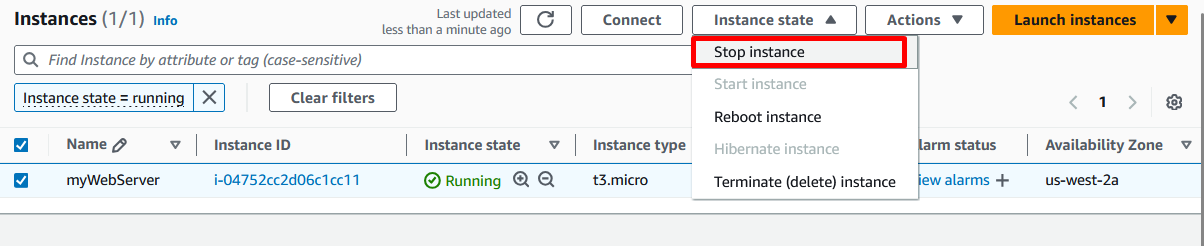
1. Return to the EC2 Management Console browser tab.
2. Choose **EC2 Dashboard** from the left navigation menu. If **EC2 Dashboard** is not visible, choose the navigation icon and select **EC2 Dashboard** from the list.



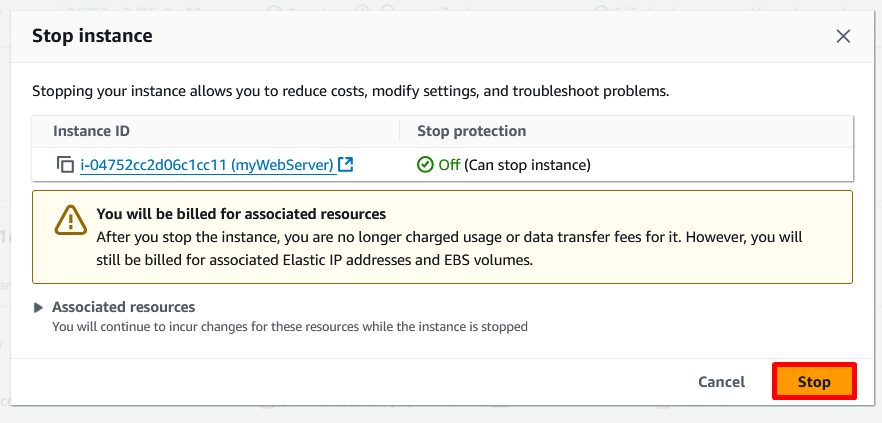
1. On the EC2 Management Console, choose and click **Instances (running)**.



1. On the **Instances** page, select **myWebServer** instance, choose **Instance state** and then click **Stop instance**.

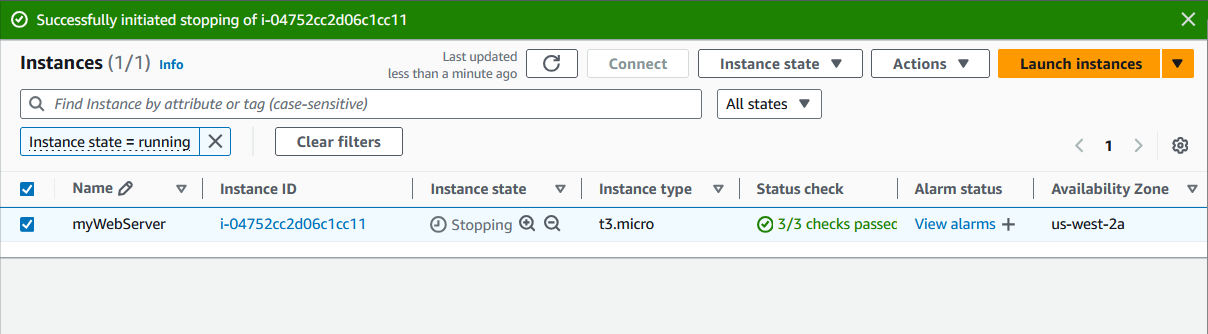


1. On the **Stop instance?** Pop-up window, click **Stop**.



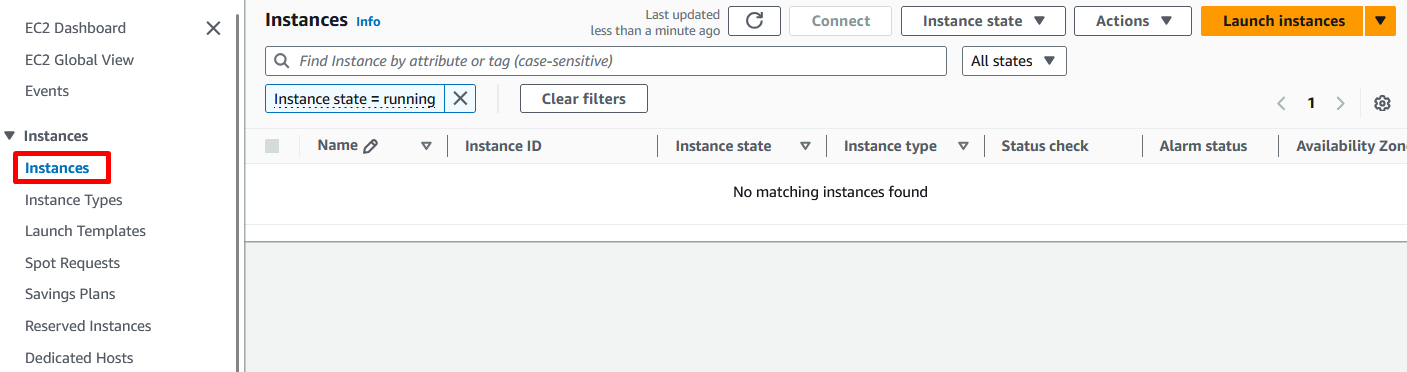
A Successfully initiated stopping message is displayed on top of the screen.

1. Choose **Clear filters** and use the refresh option as required to verify that the instance state has moved to **Stopped** state.

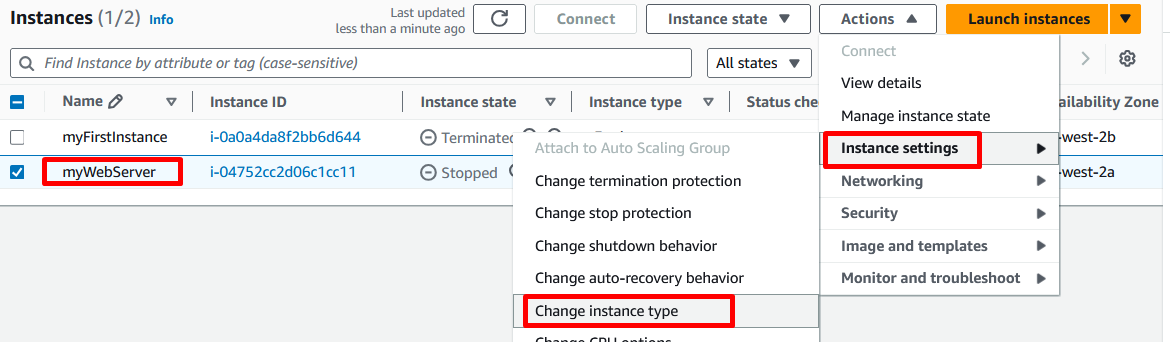


**Task 5.2: Change instance type for the Web Server Instance**

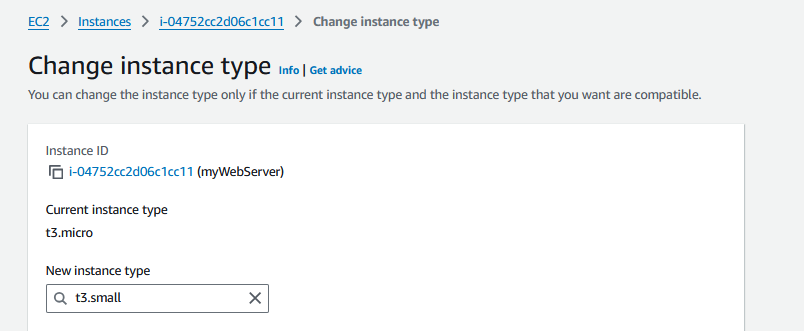
1. On the **Instances** page, select **myWebServer** instance and:



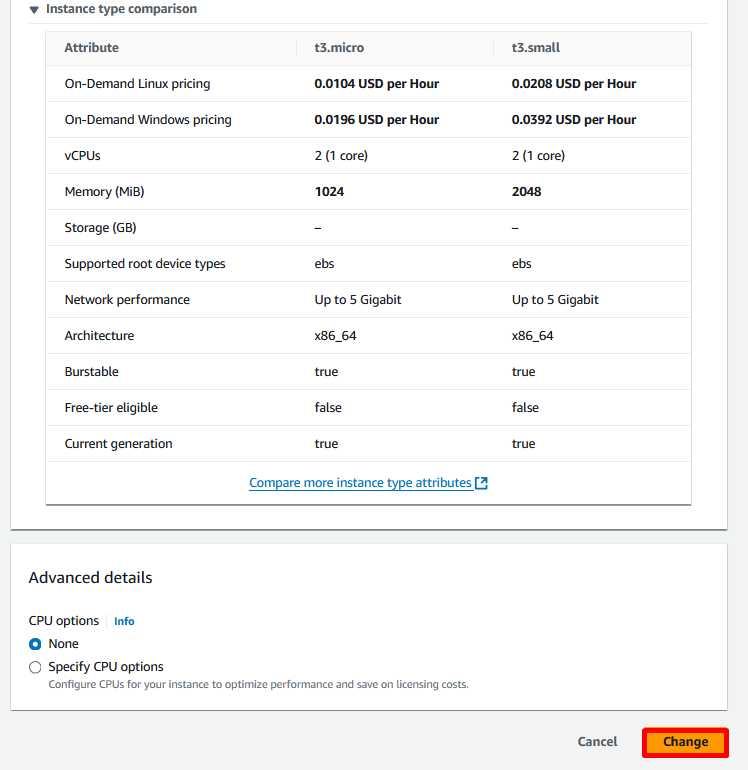
* Choose **Actions**.
* Choose and click **Instance settings**.
* Choose and click **Change instance type**.



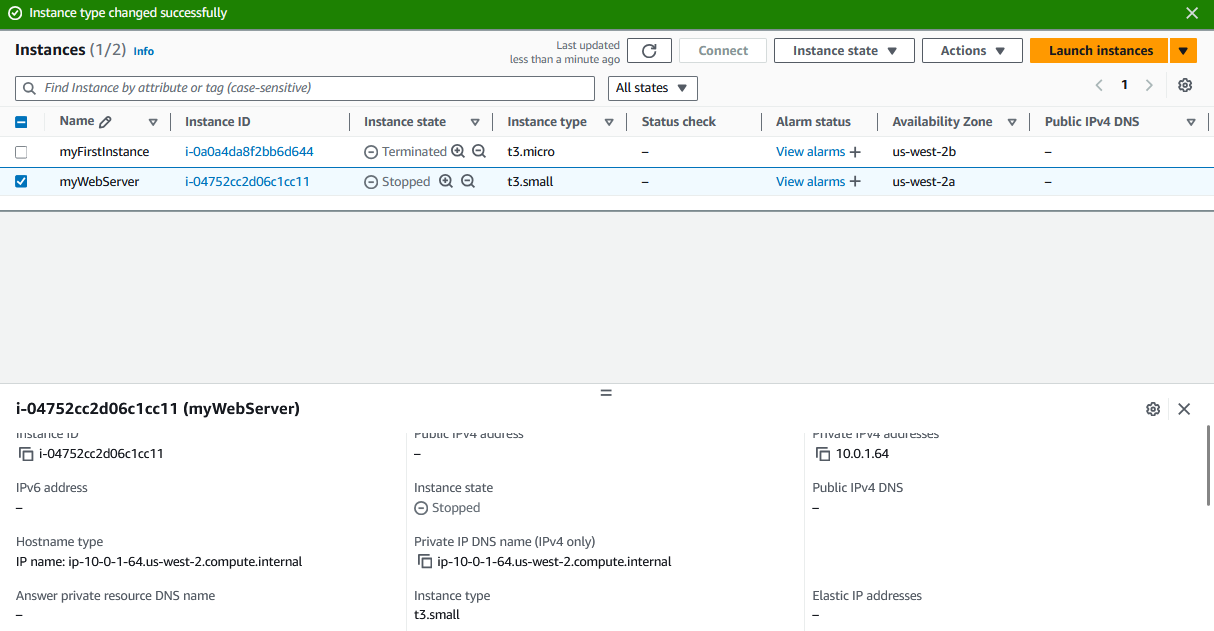
1. On the **Change instance type** page, locate the **Instance type** dropdown menu and change the instance type from **t3.micro** to **t3.small**.



1. Choose and click **change**.

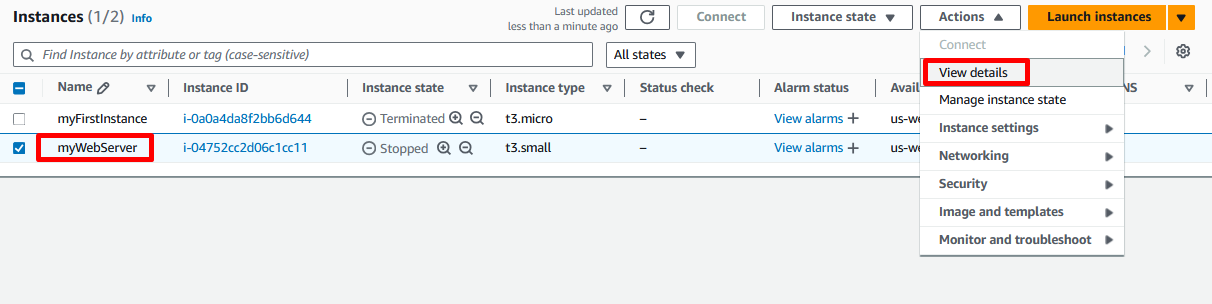


An Instance type changed successfully message is displayed on top of the screen.

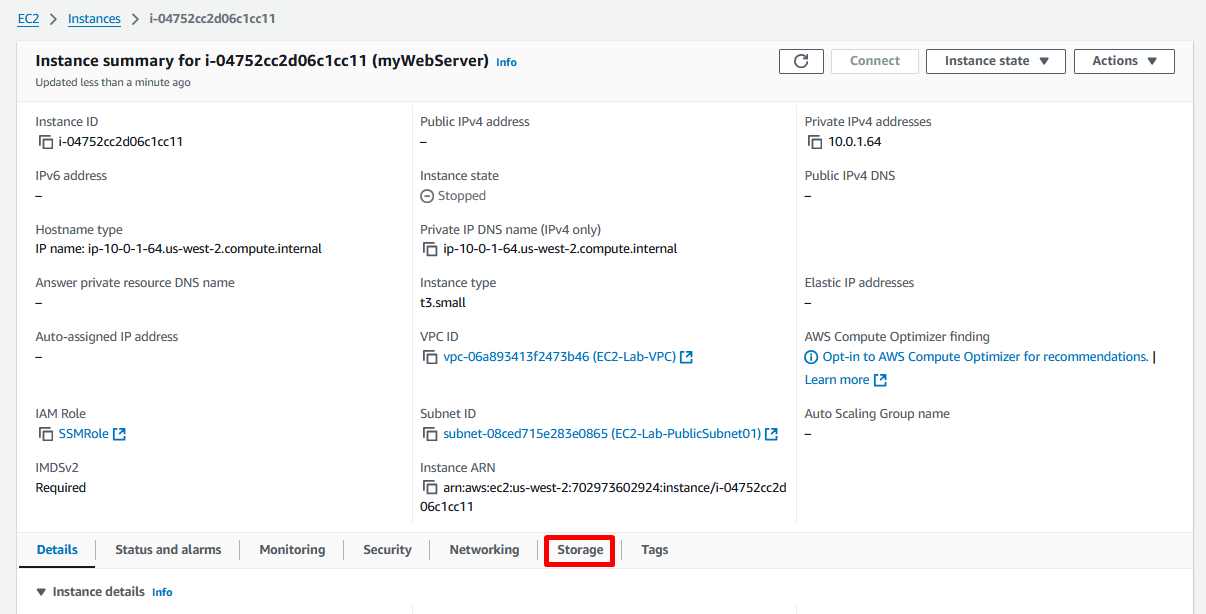


**Task 5.3: Change volume size for the Web Server Instance**

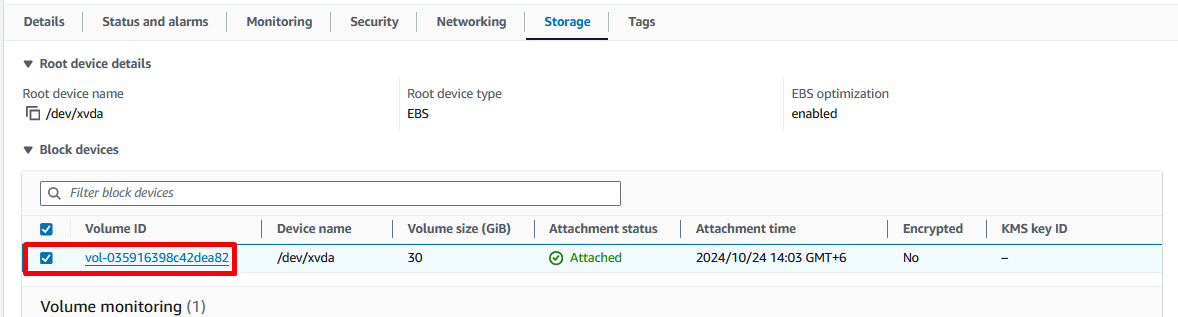
* + 1. On the **Instances** screen, select **myWebServer** instance and:
* Choose **Actions**.
* Choose and click **View details**.



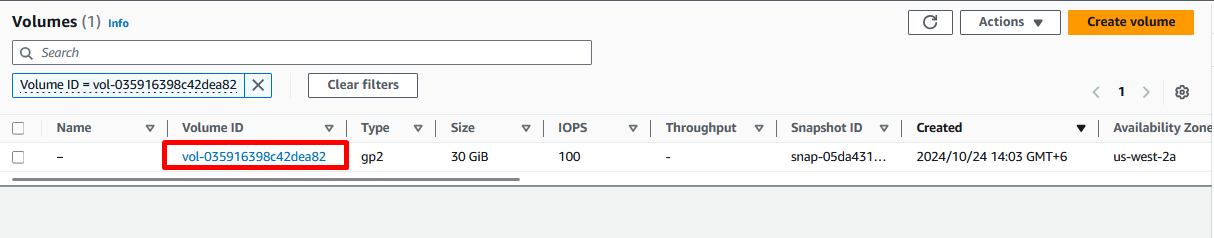
* + 1. Scroll down and choose the **Storage** tab.



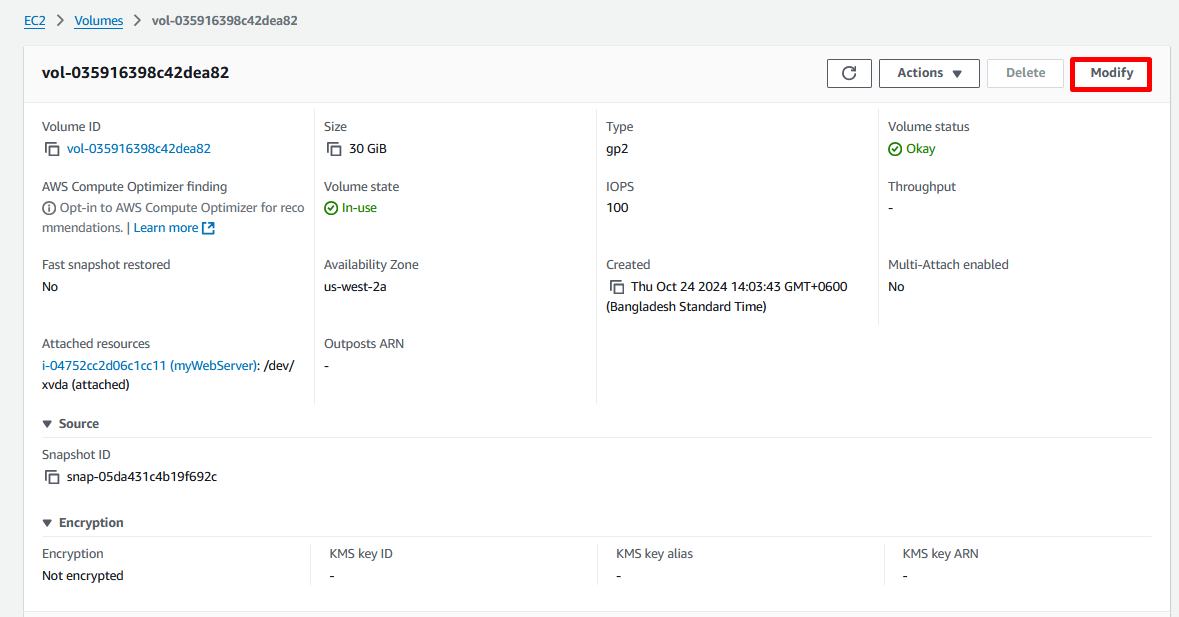
* + 1. In the **Block devices** section, select the link to the Volume ID for the root device /dev/xvda.



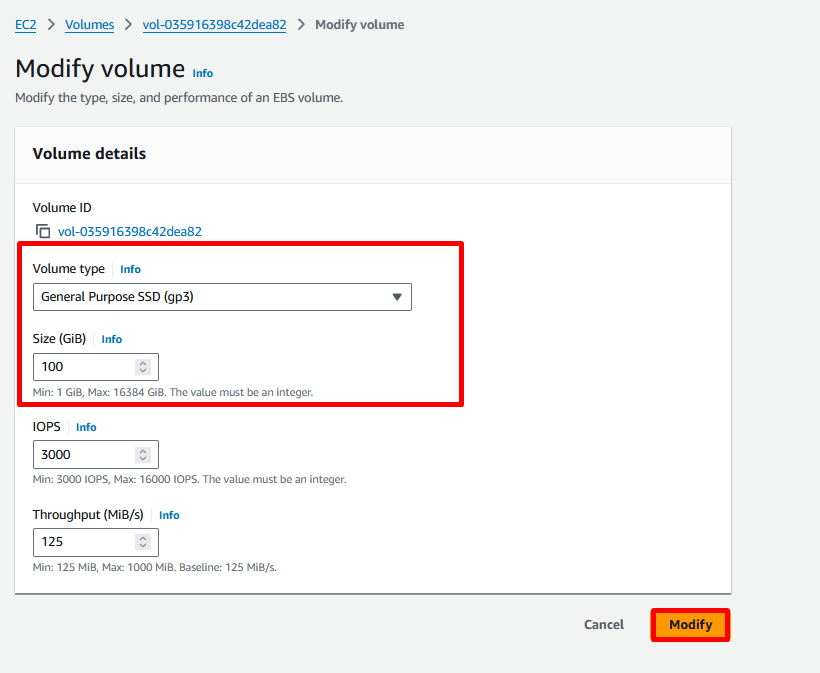
At this point, we are re-directed to the following page:



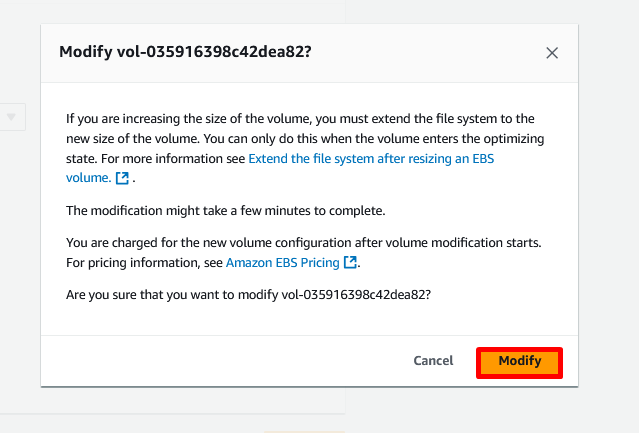
* + 1. On the **Volumes** page, select **myWebServer** Volume ID and Click Volume ID.  
       In **Volume** Details Page,
* Choose and click **Modify**.

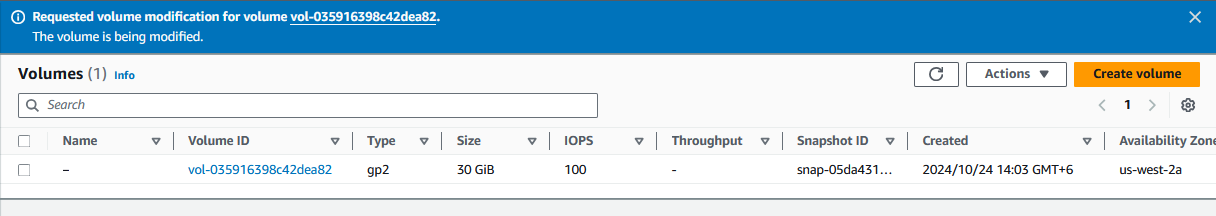
  
  
On the **Modify volume** page:

* Locate the **Volume type** dropdown menu and change the volume type from **General Purpose SSD (gp2)** to **General Purpose SSD (gp3)**.
* Locate the **Size (GiB)** textbox and change the size from **30** GiB to **100** GiB.
* Choose and click **Modify**.

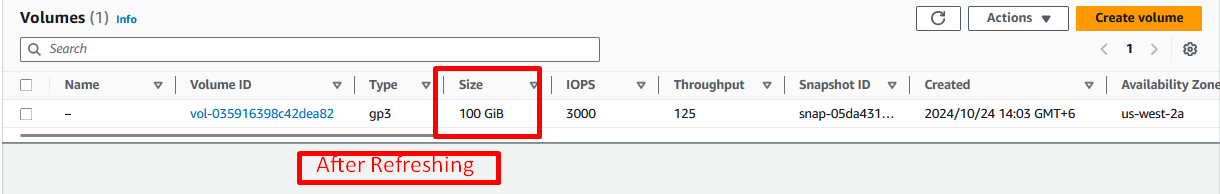


* + 1. On the **Modify vol-0587423e0105f65e8?** Pop-up window, click **Modify**.



A Requested volume modification form volume message is displayed on top of the screen.  
  
  
  
When we modify an EBS volume, it goes through a sequence of states. The volume enters the modifying state, the optimizing state, and finally the completed state. At this point, the volume is ready to be further modified.

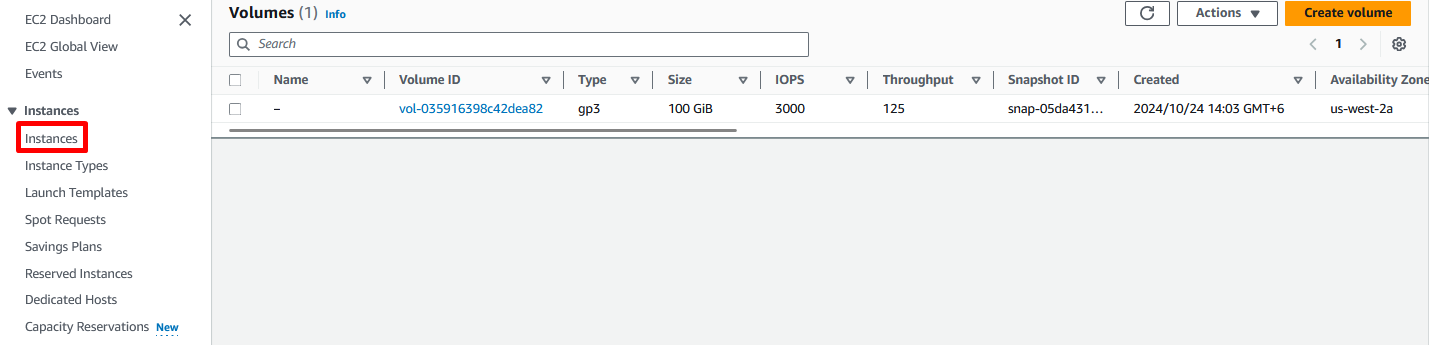
While the volume is in the optimizing state, our volume performance is in between the source and target configuration specifications. Transitional volume performance will be no less than the source volume performance. If we are downgrading IOPS, transitional volume performance is no less than the target volume performance.

Volume modification might take some time, and hence we need to wait for few minutes and use the refresh option to confirm the results of the volume modification.   
  


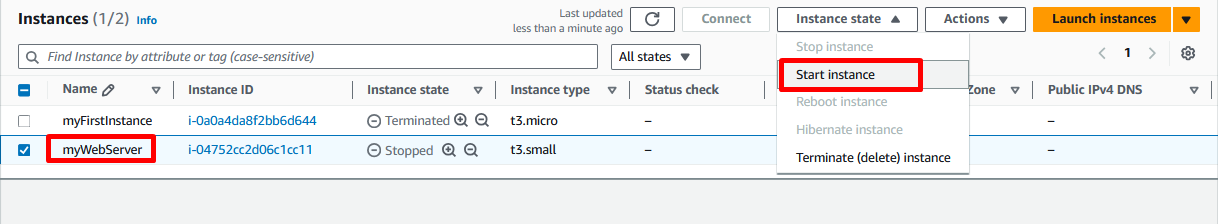
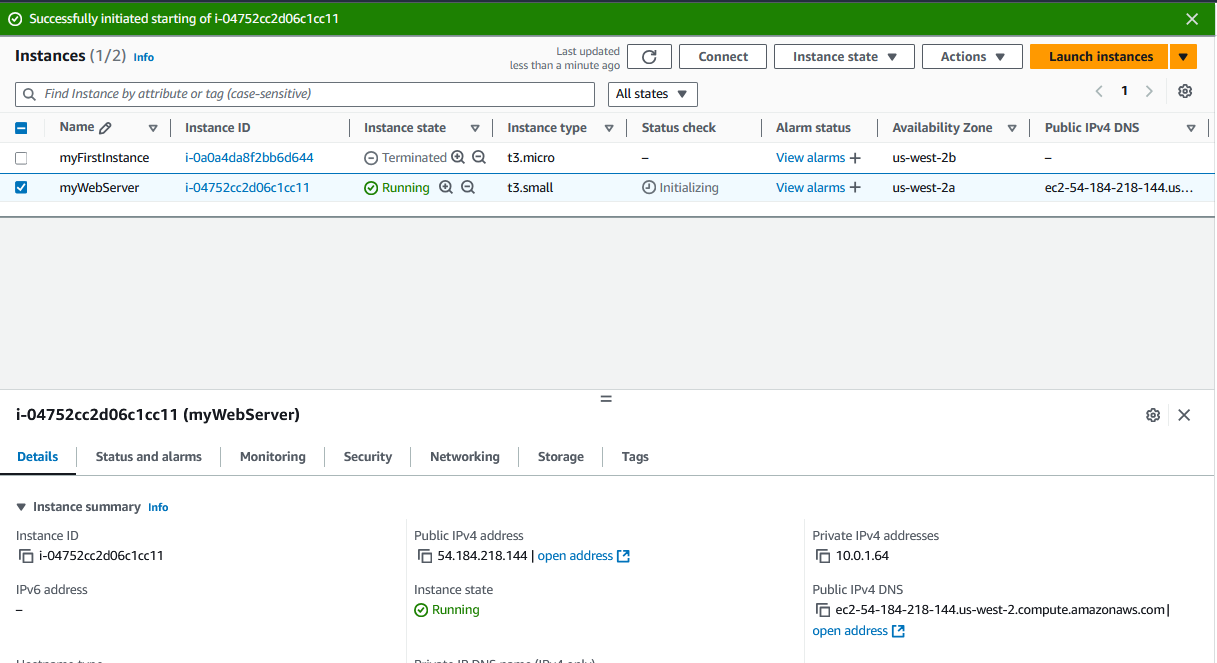
We can verify if the volume modification was successful or not by reviewing the **Volume state** column. After successful modification, the volume state moves to In-use state.

**Task 5.4: Restart the Web Server Instance**

* 1. To navigate back to the EC2 console, choose **EC2 Dashboard** from the left navigation menu.
  2. On the EC2 Management Console, choose **Instances** from the left navigation menu.



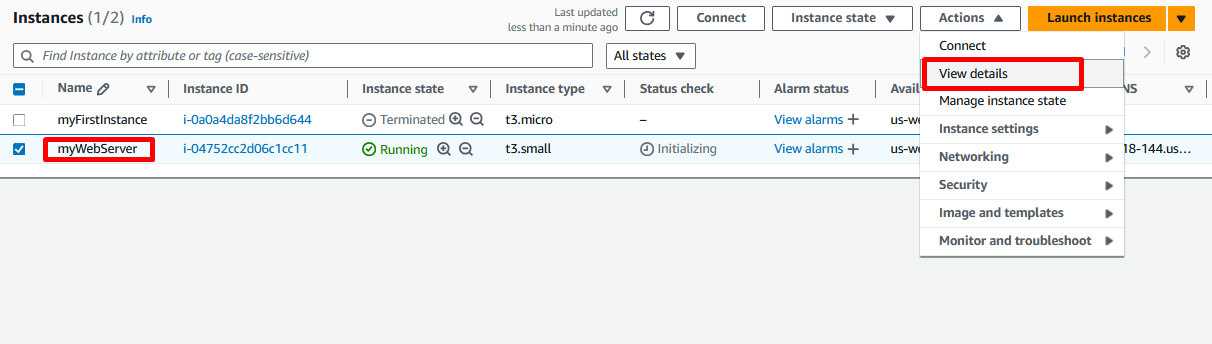
* 1. On the Instances page, select **myWebServer** instance and:
* Choose **Instance state**.
* Choose and click **Start instance**.

  
  
A Successfully initiated starting message is displayed on top of the screen.   
  


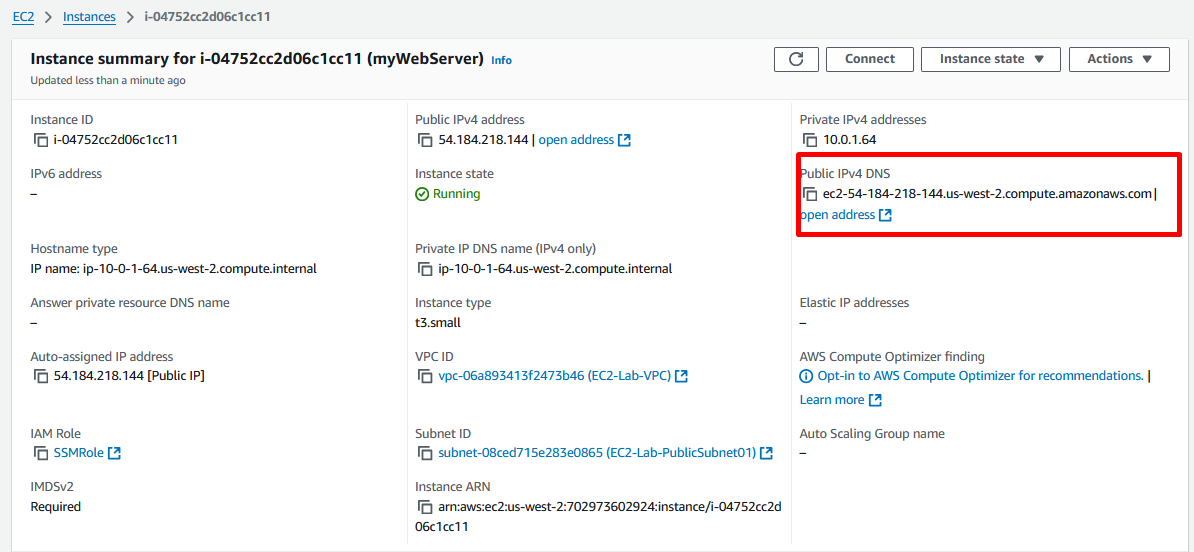
It might take few minutes for the **myWebServer** instance to finish its status checks. We can confirm that by checking the **Status check** column where it shows Initializing and eventually moves to 2/2 checks passed. Use the refresh option as required.

**Task 5.5: Test the PHP application after modifying EC2 instance attributes**

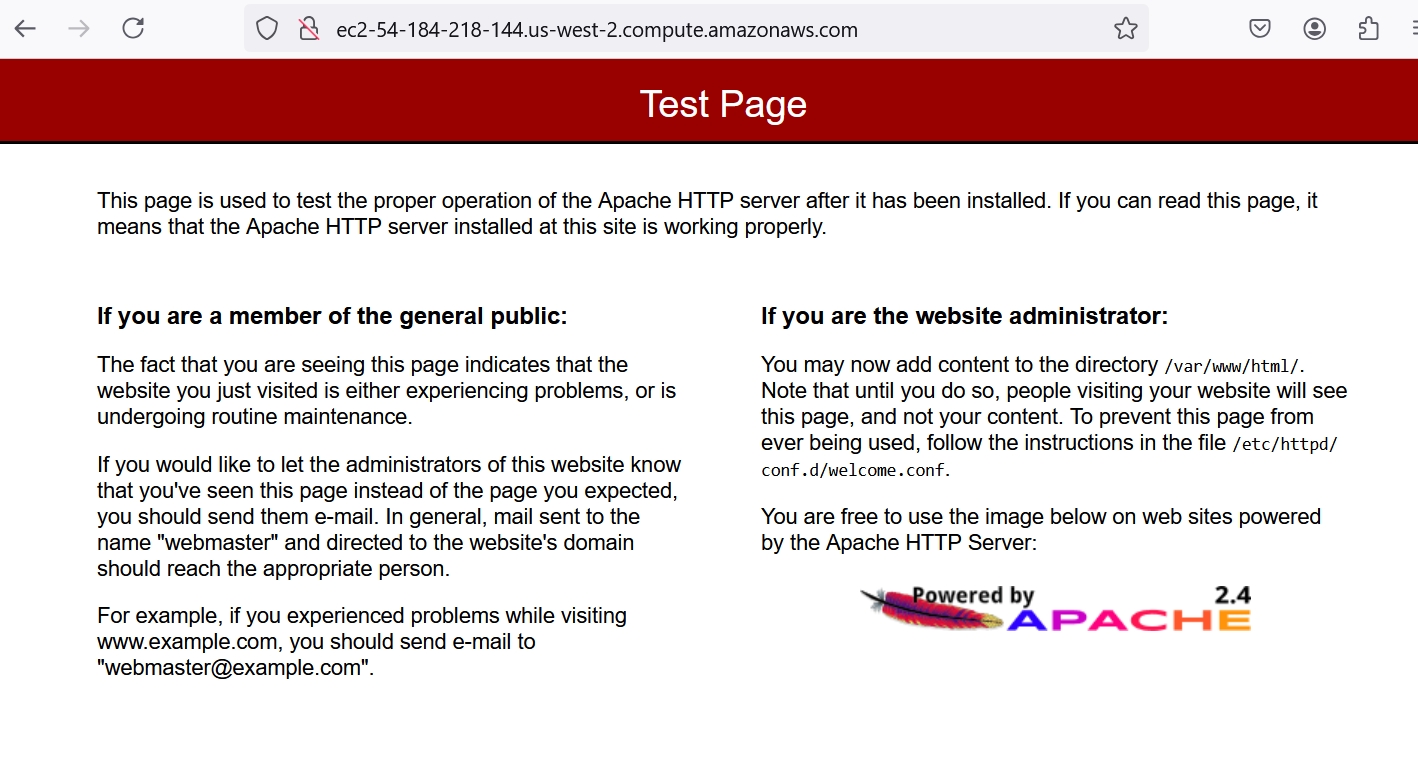
* 1. On the **Instances** page, select **myWebServer** instance and:
* Choose **Actions**.
* Choose and click **View details**.



* 1. On the **Instance summary for myWebServer** page, copy the value of **Public IPv4 DNS** and paste it into a new browser tab. We can also use **Public IPv4 address** instead of **Public IPv4 DNS.**



At this point, we see the following Test Page for our sample PHP application.



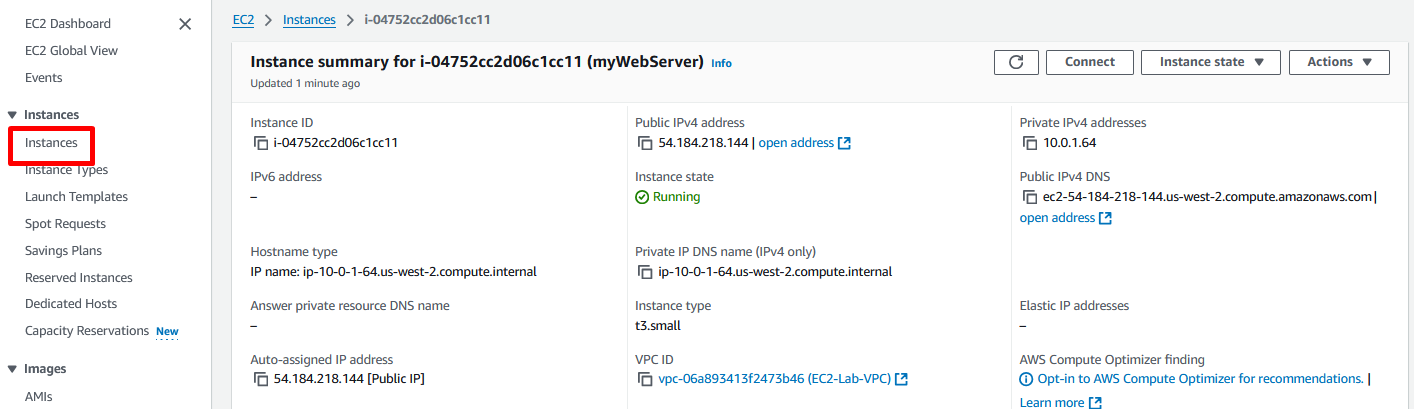
This confirms that the PHP application is successfully running again after EC2 instance modification.

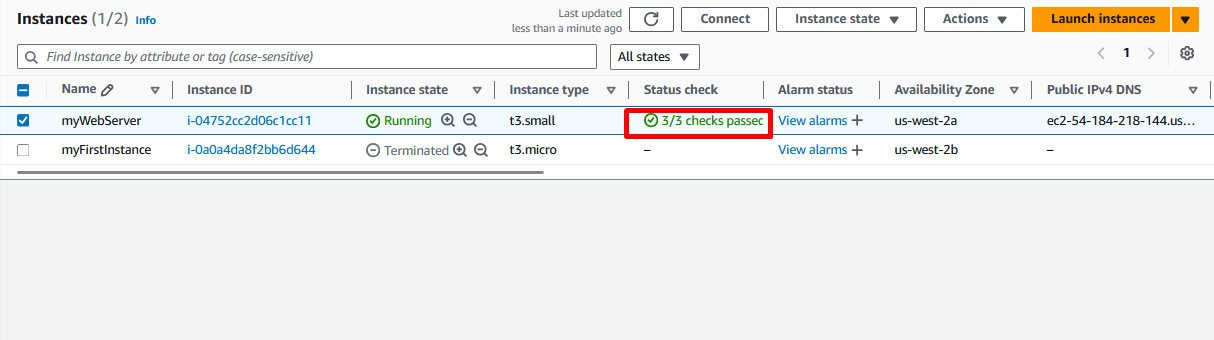
**Congratulations!** We have successfully modified attributes for an Amazon EC2 instance a re-tested our application.

**Task 6: Terminate and clean up the EC2 instance virtual machine**

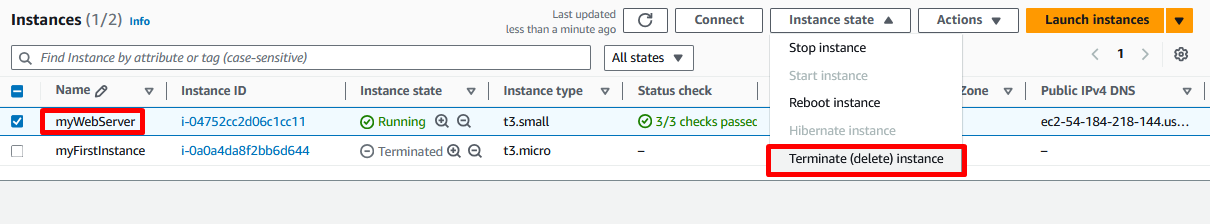
In this final task, we understand how to terminate an Amazon EC2 instance.

* 1. Return to the EC2 Management Console browser tab and click instances.

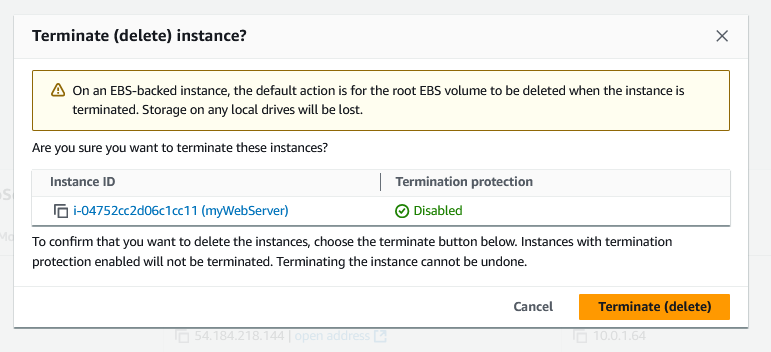


Now, we can see that the **Status check** column shows 3/3 checks passed.  
  


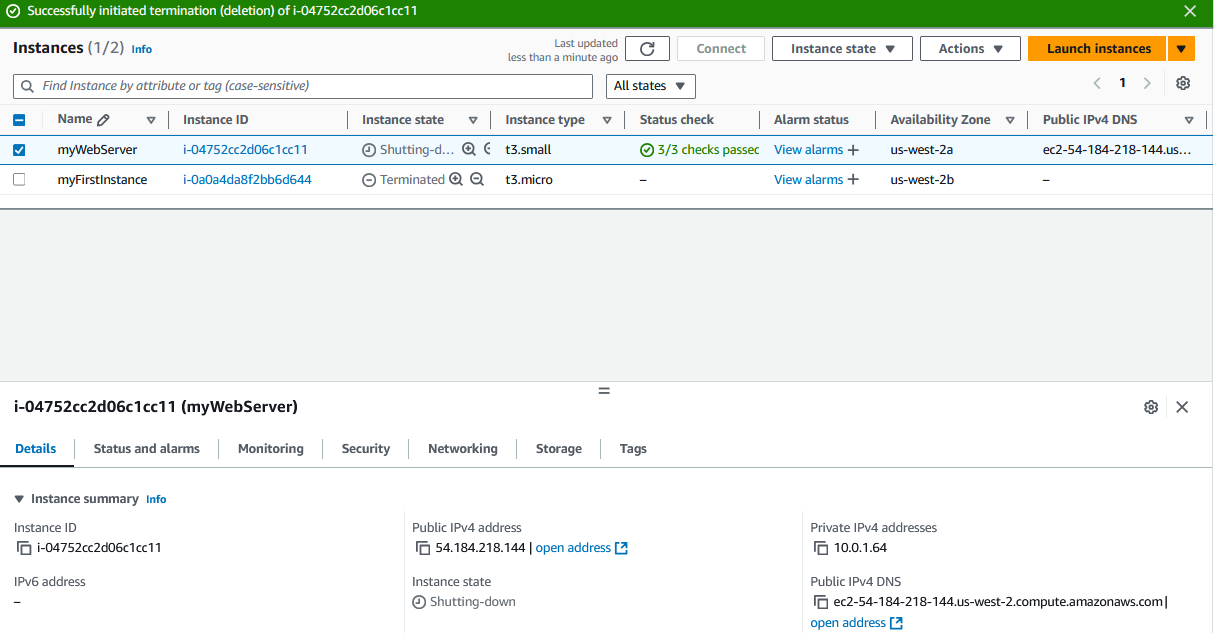
* 1. From top right corner of the instance summary page, choose **Instance state** and then click **Terminate instance**.



* 1. On the **Terminate instance?** Pop-up window, click **Terminate (delete)**



A Successfully initiated termination (deletion) message is displayed on top of the screen.



**Congratulations!** We have successfully terminated the Web Server EC2 instance.

**End lab**

Follow these steps to close the console and end our lab.

* 1. Return to the **AWS Management Console**.
  2. At the upper-right corner of the page, choose **AWSLabsUser**, and then click Sign out.
  3. Choose and click **End Lab** and then confirm that we want to end our lab.