EXPERIMENT 3

Aim: To study and Implement Infrastructure as a service using Amazon EC2 Compute.

Objective:

- Discuss different types of compute solutions, their features and benefits
- Discuss the basic features and concepts of Amazon EC2
- Describe Amazon EC2 instance types and how to choose an instance type
- Describe how to use Amazon EC2 to launch and configure an instance
- Describe how to manage Amazon EC2 instances
- Use Amazon EC2 to launch and manage and instance

Theory: To study and implement Infrastructure as a Service (IaaS) using Amazon EC2 Compute involves exploring the foundational principles and functionalities of Amazon EC2. Amazon EC2, or Elastic Compute Cloud, is a core component of Amazon Web Services (AWS), providing resizable compute capacity in the cloud. The theory behind this experiment revolves around understanding EC2's role in delivering virtual servers known as instances, which can be quickly provisioned and configured to meet varying workload demands. EC2 offers a wide range of instance types optimized for different use cases, such as general-purpose, compute-optimized, memory-optimized, and storage-optimized instances, allowing users to choose resources tailored to their specific requirements. Furthermore, EC2's scalability enables users to easily scale resources up or down based on demand, ensuring optimal performance and cost efficiency. Key concepts include selecting the appropriate Amazon Machine Image (AMI), configuring instance details, managing security settings, and monitoring instance performance. By studying and implementing IaaS using Amazon EC2 Compute, participants gain insights into the flexibility, scalability, and cost-effectiveness of cloud-based infrastructure provisioning and management. This experiment serves as a practical exploration of cloud computing principles and empowers participants to leverage EC2 effectively for deploying and managing virtual computing resources in the cloud.

Implementation And Output:

Task 1: Launching your EC2 instance

In this task, you launch an EC2 instance with termination protection. Termination protection prevents you from accidentally terminating an EC2 instance. You also deploy your instance with a user data script to deploy a simple web server.

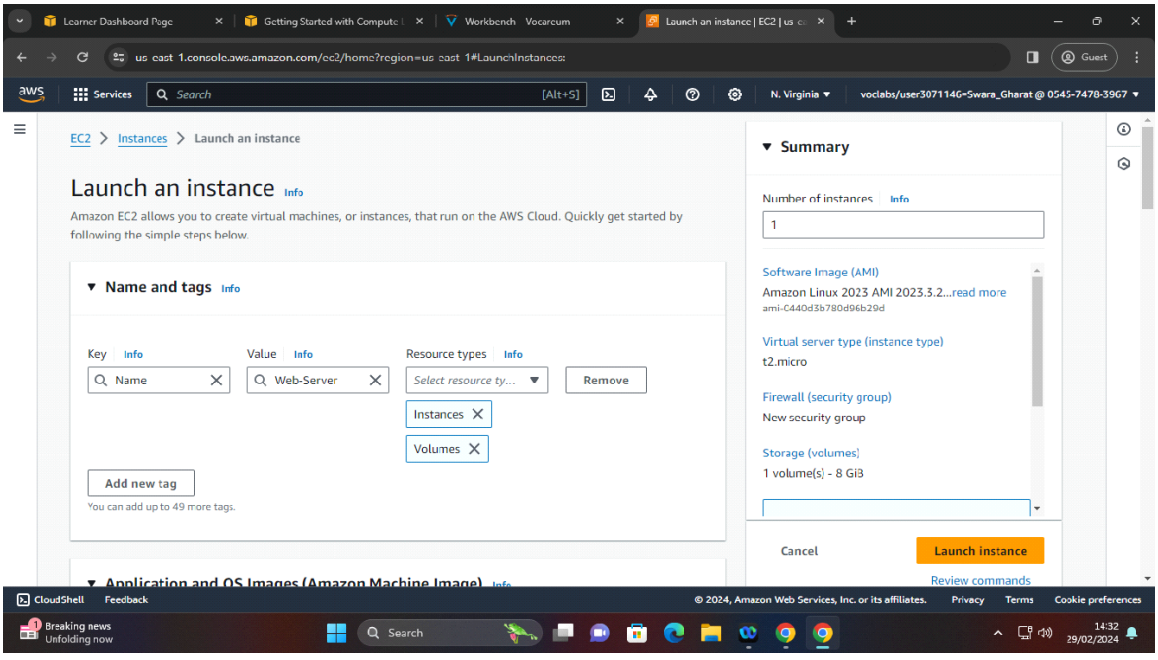
1. In the AWS Management Console on the **Services** menu, enter **EC2**. From the search results, choose **EC2**.
2. In the left navigation pane, choose **EC2 Dashboard** to ensure that you are on the dashboard page.
3. In the **Launch instance** section, choose the **Launch instance** button.

Step 1: Name your EC2 instance

Using tags, you can categorize your AWS resources in different ways (for example, by purpose, owner, or environment). This categorization is useful when you have many resources of the same type. You can quickly identify a specific resource based on the tags that you have assigned to it. Each tag consists of a key and a value, both of which you define.

When you name your instance, AWS creates a key-value pair. The key for this pair is **Name**, and the value is the name that you enter for your EC2 instance.

4. In the **Name and tags** pane, in the **Name** text box, enter Web-Server
5. Choose the **Add additional tags** link.
6. From the **Resource types** dropdown list, select **Instances** and **Volumes**.



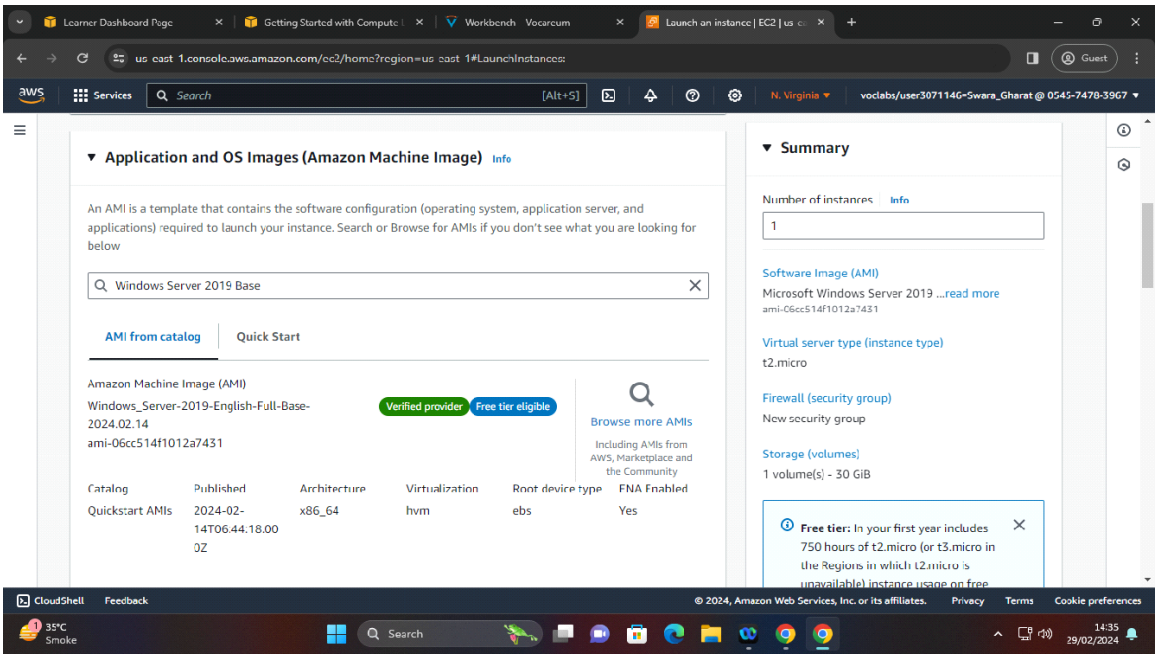
Step 2: Choose an AMI

An Amazon Machine Image (AMI) provides the information required to launch an instance, which is a virtual server in the cloud. An AMI includes the following:

- A template for the root volume for the instance (for example, an operating system or an application server with applications)
- Launch permissions that control which AWS accounts can use the AMI to launch instances
- A block device mapping that specifies the volumes to attach to the instance when it is launched

The **Quick Start** list contains the most commonly used AMIs. You can also create your own AMI or select an AMI from the AWS Marketplace, an online store where you can sell or buy software that runs on AWS.

7. Locate the **Application and OS Images (Amazon Machine Image)** section. It is just below the **Name and tags** section.
8. In the search box, enter Windows Server 2019 Base and press Enter.
9. Next to **Microsoft Windows Server 2019 Base**, choose **Select**.

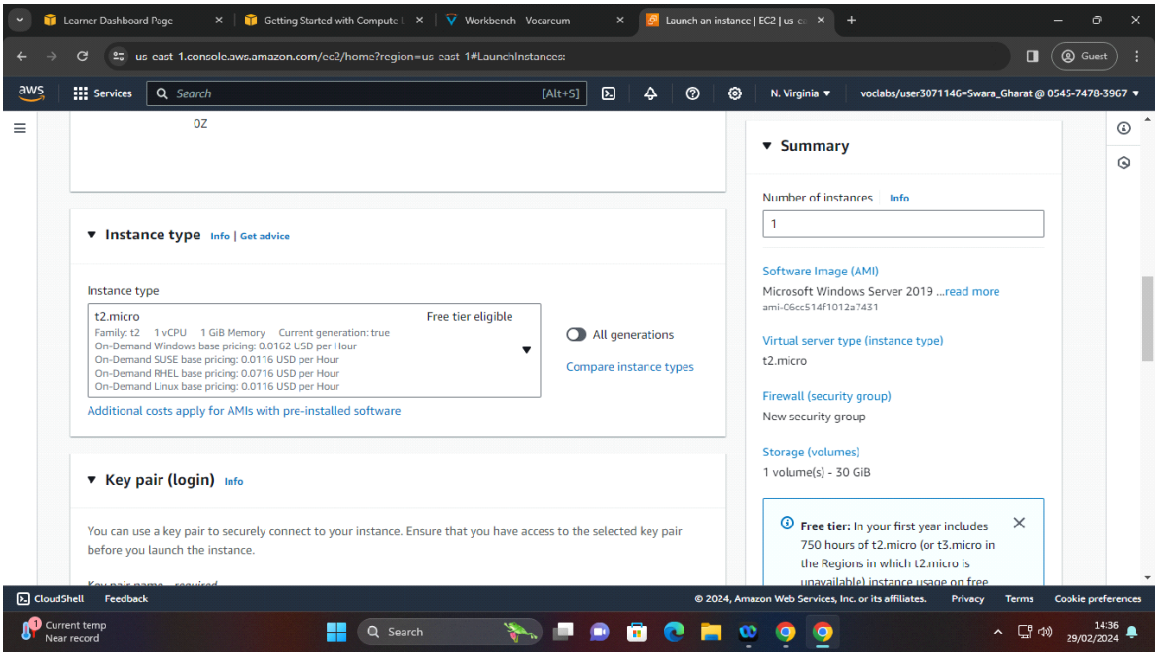


Step 3: Choose an instance type

Amazon EC2 provides a wide selection of instance types that are optimized to fit different use cases. Instance types comprise varying combinations of CPU, memory, storage, and networking capacity and give you the flexibility to choose the appropriate mix of resources for your applications. Each instance type includes one or more instance sizes so that you can scale your resources to the requirements of your target workload.

In this step, you choose a **t2.micro** instance. This instance type has 1 virtual CPU and 1 GiB of memory.

10. In the **Instance type** section, keep the default instance type, **t2.micro**.

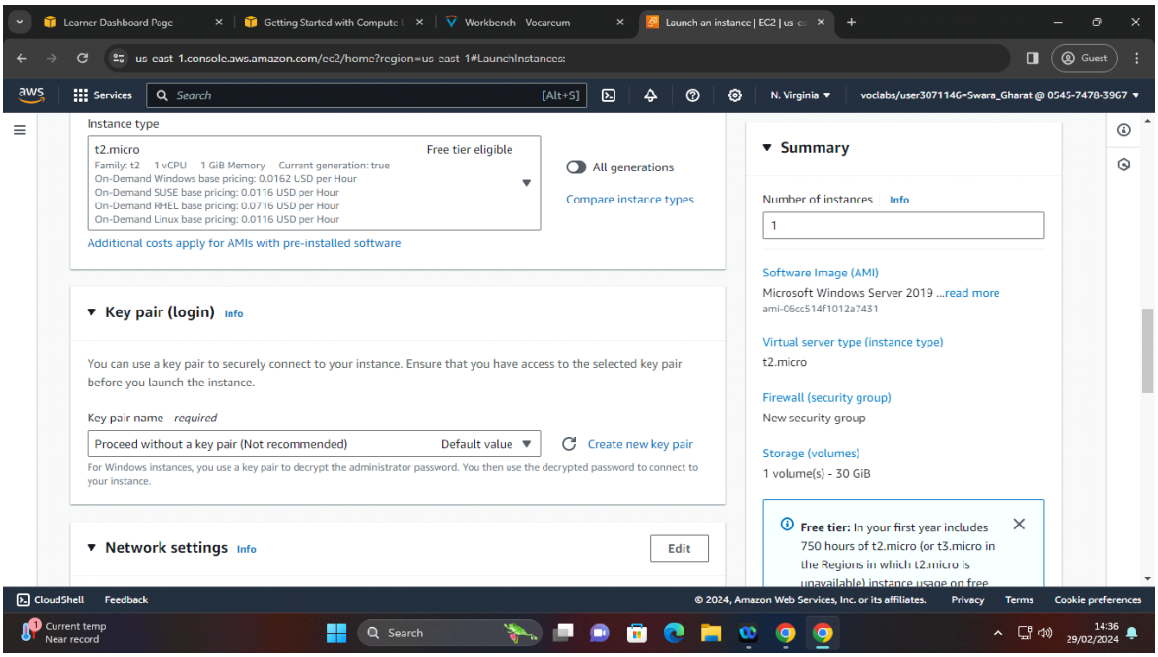


Step 4: Configure a key pair

Amazon EC2 uses public key cryptography to encrypt and decrypt login information. To log in to your instance, you must create a key pair, specify the name of the key pair when you launch the instance, and provide the private key when you connect to the instance.

In this lab, you do not connect to your instance using an SSH key, so you do not need to configure a key pair.

11. In the **Key pair (login)** section, from the **Key pair name - *required*** dropdown list, choose **Proceed without a key pair (not recommended)**.

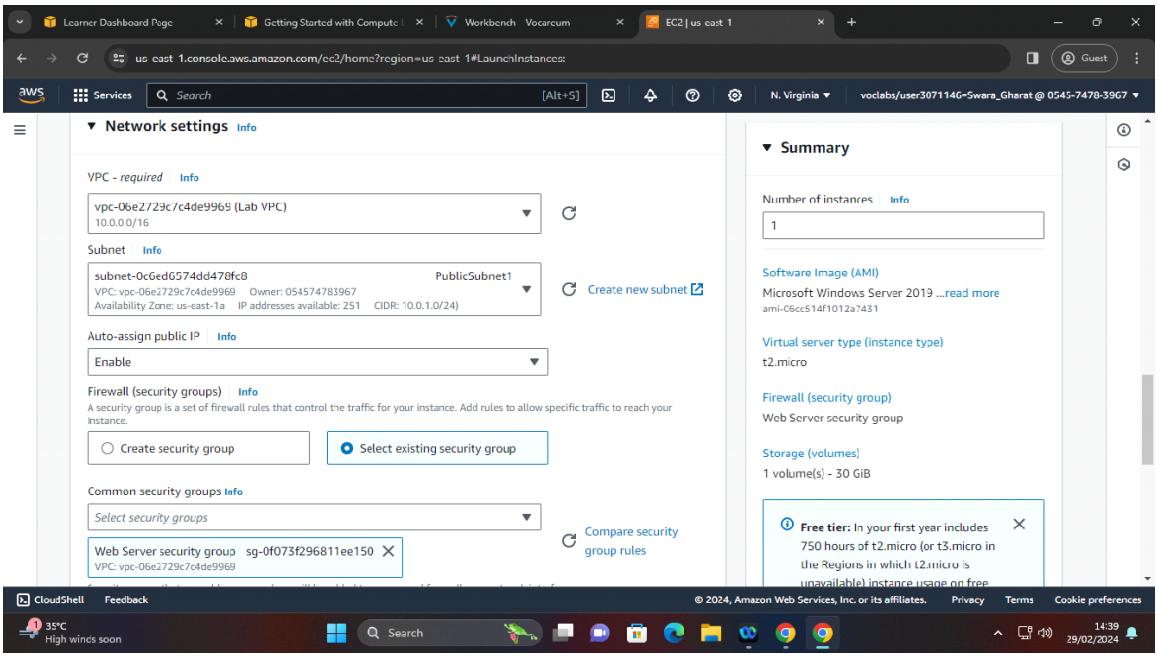


Step 5: Configure the network settings

You use this pane to configure networking settings.

The virtual private cloud (VPC) indicates which VPC you want to launch the instance into. You can have multiple VPCs, including different ones for development, testing, and production.

12. In the **Network settings** section, choose **Edit**.
13. From the **VPC - *required*** dropdown list, choose **Lab VPC**.
    The Lab VPC was created using an AWS CloudFormation template during the setup process of your lab. This VPC includes two public subnets in two different Availability Zones.
14. For **Security group name - *required***, choose **Select existing security group**.
15. From **Common security groups**, select Web Server security group.
    A security group acts as a virtual firewall that controls the traffic for one or more instances. When you launch an instance, you associate one or more security groups with the instance. You add rules to each security group that allow traffic to or from its associated instances. You can modify the rules for a security group at any time; the new rules are automatically applied to all instances that are associated with the security group.



Step 6: Add storage

Amazon EC2 stores data on a network-attached virtual disk called Amazon Elastic Block Store (Amazon EBS).

You launch the EC2 instance using a default 30 GiB disk volume. This is your root volume (also known as a boot volume).

16. In the **Configure storage** section, keep the default storage configuration.

Step 7: Configure advanced details

17. Expand the **Advanced details** section.
18. For **IAM instance profile**, choose the role that has **LabInstanceProfile** in the name.

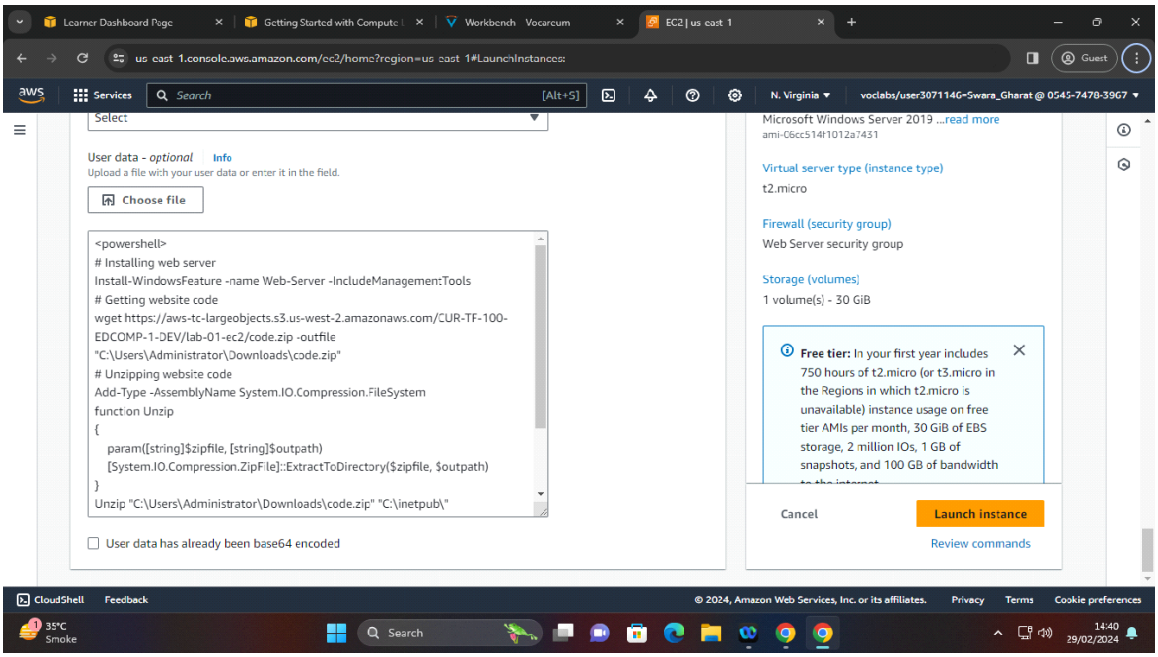
    When you no longer require an EC2 instance, you can terminate it, which means that the instance stops, and Amazon EC2 releases the instance's resources. You cannot restart a terminated instance. If you want to prevent your users from accidentally terminating the instance, you can turn on (enable) termination protection for the instance, which prevents users from terminating instances.
19. From the **Termination protection** dropdown list, choose **Enable**.

    When you launch an instance in Amazon EC2, you have the option of passing user data to the instance. These commands can be used to perform common automated configuration tasks and even run scripts after the instance starts.

20. Copy the following commands, and paste them into the **User data** text box.

```
<powershell>
# Installing web server
Install-WindowsFeature -name Web-Server -IncludeManagementTools
# Getting website code
wget https://aws-tc-largeobjects.s3.us-west-2.amazonaws.com/CUR-TF-100-EDCOMP-1-DEV/lab-01-ec2/code.zip -outfile "C:\Users\Administrator\Downloads\code.zip"
# Unzipping website code
Add-Type -AssemblyName System.IO.Compression.FileSystem
function Unzip
{
    param([string]$zipfile, [string]$outpath)
    [System.IO.Compression.ZipFile]::ExtractToDirectory($zipfile, $outpath)
}
Unzip "C:\Users\Administrator\Downloads\code.zip" "C:\inetpub\"
# Setting Administrator password
$Secure_String_Pwd = ConvertTo-SecureString "P@ssW0rD!" -AsPlainText -Force
$UserAccount = Get-LocalUser -Name "Administrator"
$UserAccount | Set-LocalUser -Password $Secure_String_Pwd
</powershell>
```

21. The script does the following:
    a. Installs a Microsoft Internet Information Services (IIS) web server
    b. Creates a simple web site
    c. Sets the password for the Administrator user



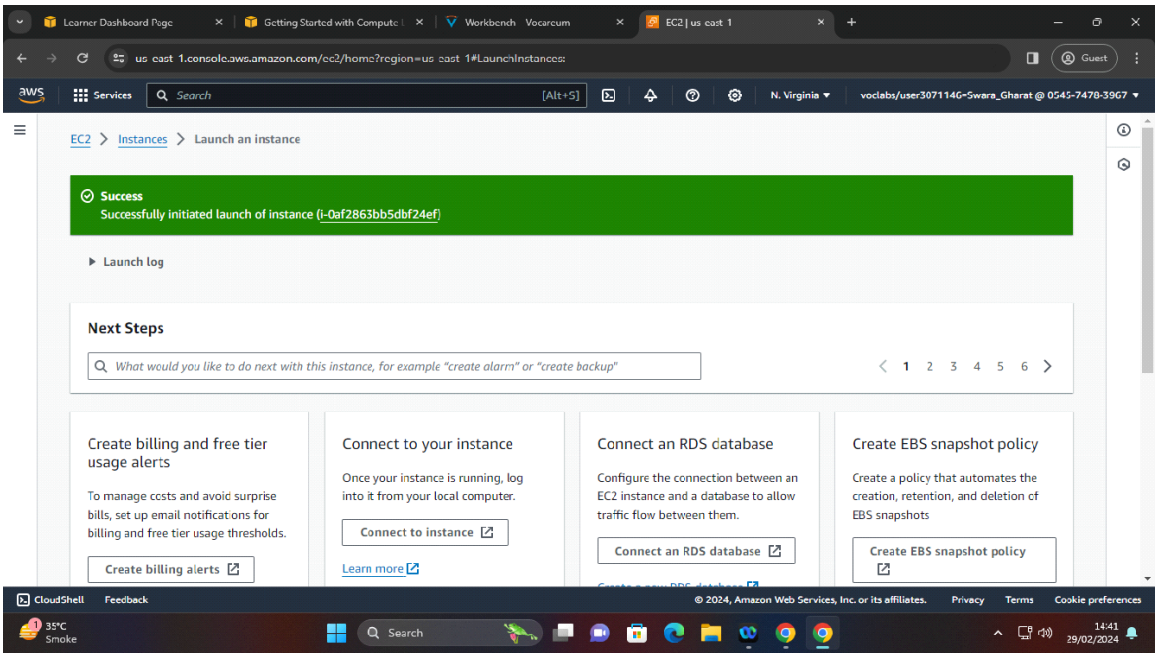
Step 8: Launch an EC2 instance

Now that you have configured your EC2 instance settings, it is time to launch your instance.

22. In the **Summary** section, choose **Launch instance**.
    A message indicates that you have successfully initiated the launch of your instance.

23. Choose **View all instances**
    The instance appears in a **Pending** state, which means that it is being launched. It then changes to **Running**, which indicates that the instance has started booting. There will be a short time before you can access the instance.
    The instance receives a public Domain Name System (DNS) name that you can use to contact the instance from the Internet.
24. Next to your **Web-Server**, select the check box. The **Details** tab displays detailed information about your instance.
    To view more information in the **Details** tab, drag the window divider upward.
    Review the information displayed in the **Details, Security** and **Networking** tabs.

25. Wait for your instance to display the following:
    Note: Refresh if needed.
    a. **Instance State:** Running
    b. **Status Checks:** 2/2 checks passed



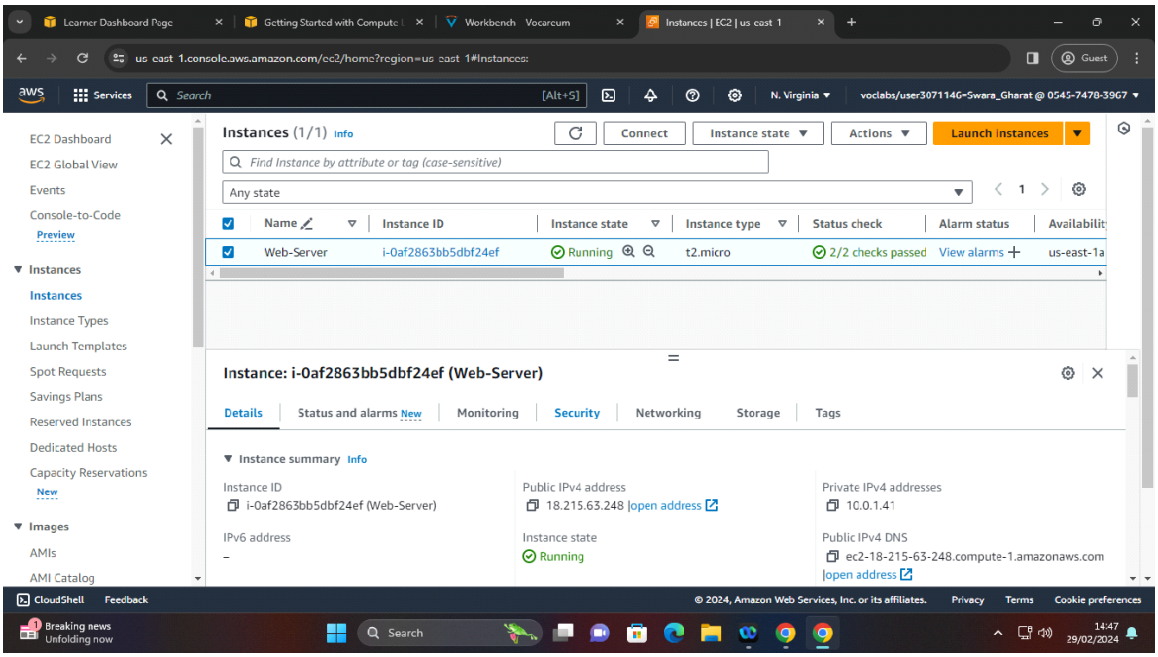
Task 2: Monitor your instance

Monitoring is an important part of maintaining the reliability, availability, and performance of your EC2 instances and your AWS solutions.

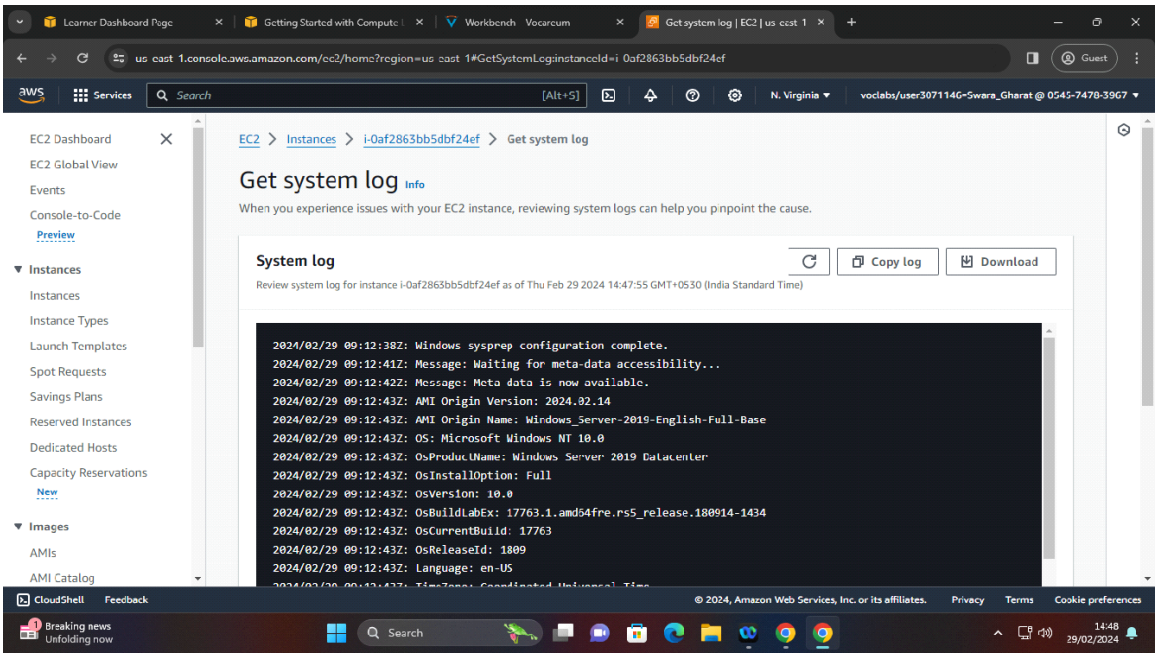
26. Choose the **Status checks** tab.

    With instance status monitoring, you can quickly determine whether Amazon EC2 has detected any problems that might prevent your instances from running applications. Amazon EC2 performs automated checks on every running EC2 instance to identify hardware and software issues.

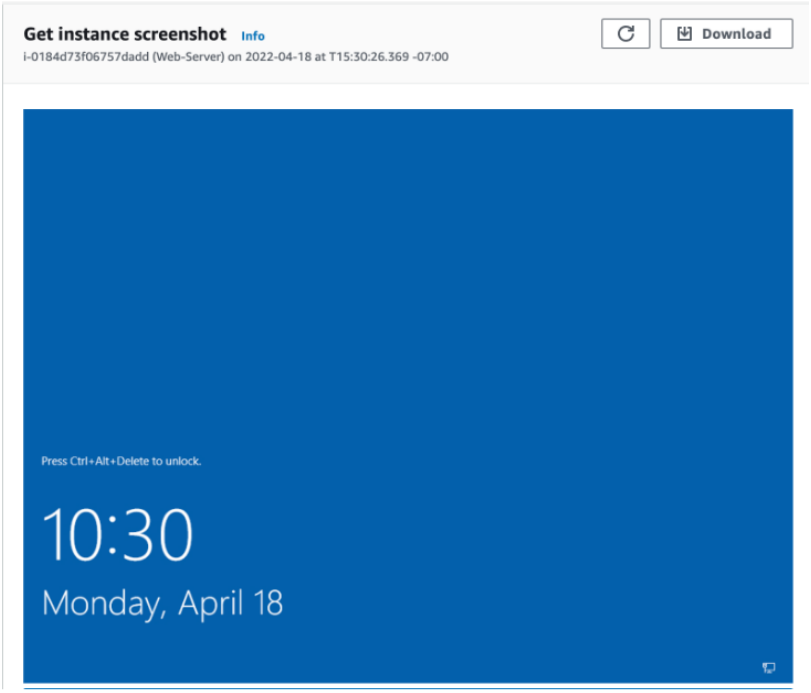
    Notice that both the **System reachability** and **Instance reachability** checks have passed.



27. Choose the **Monitoring** tab.
    This tab displays Amazon CloudWatch metrics for your instance. Currently, there are not many metrics to display because the instance was recently launched.
    You can choose a graph to see an expanded view.
    Amazon EC2 sends metrics to Amazon CloudWatch for your EC2 instances. Basic (5 minute) monitoring is turned on by default and is free. You can turn on detailed (1 minute) monitoring. With detailed monitoring, you will be charged per metric that you send to CloudWatch.
28. At the top of the page, choose the **Actions** dropdown menu. Select **Monitor and troubleshoot Get system log**.
    The system log displays the console output of the instance, which is a valuable tool for problem diagnosis. It is especially useful for troubleshooting service configuration issues that could cause an instance to terminate or become unreachable. If you do not see a system log, wait a few minutes and then try again.



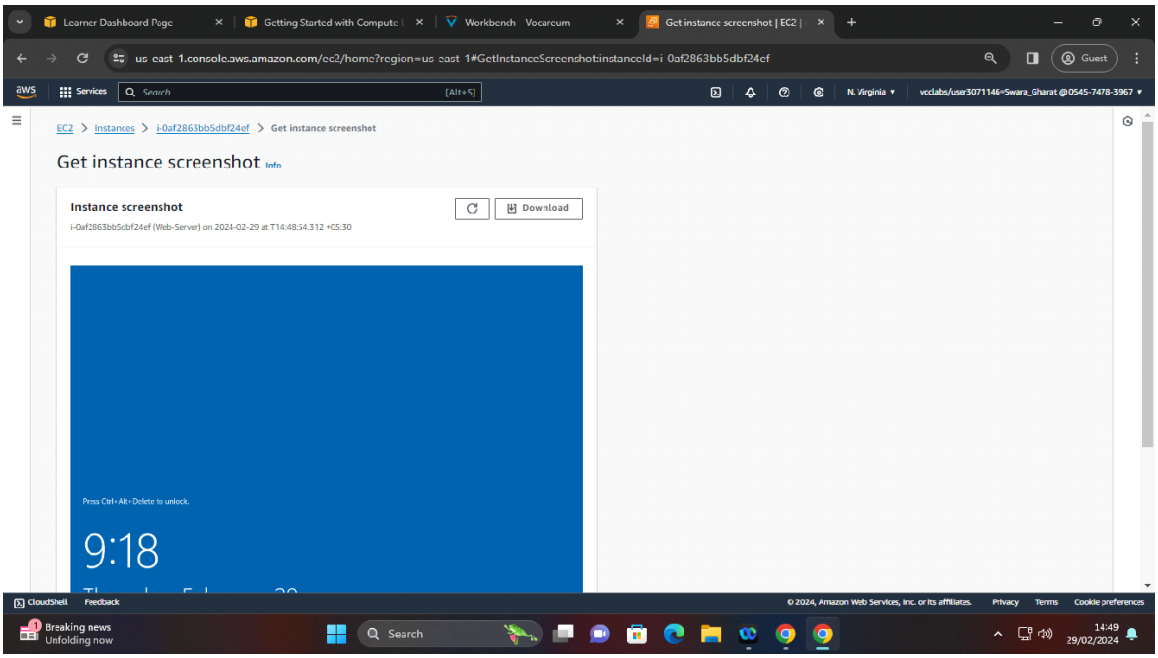
29. Scroll through the log and review the messages in the output.
30. To return to the Amazon EC2 dashboard, choose **Cancel**.
31. With your **Web-Server** selected, choose the **Actions** dropdown menu, and select **Monitor and troubleshoot Get instance screenshot**.
    This option shows you what your EC2 instance console would look like if a screen were attached to it. Because this is a Windows instance, the screenshot shows a locked log-in screen.



If you are unable to reach your instance via SSH or RDP, you can capture a screenshot

of your instance and view it as an image. This option provides visibility about the status of the instance for quicker troubleshooting.

32. At the bottom of the page, choose **Cancel**.



Task 3: Updating your security group and accessing the web server

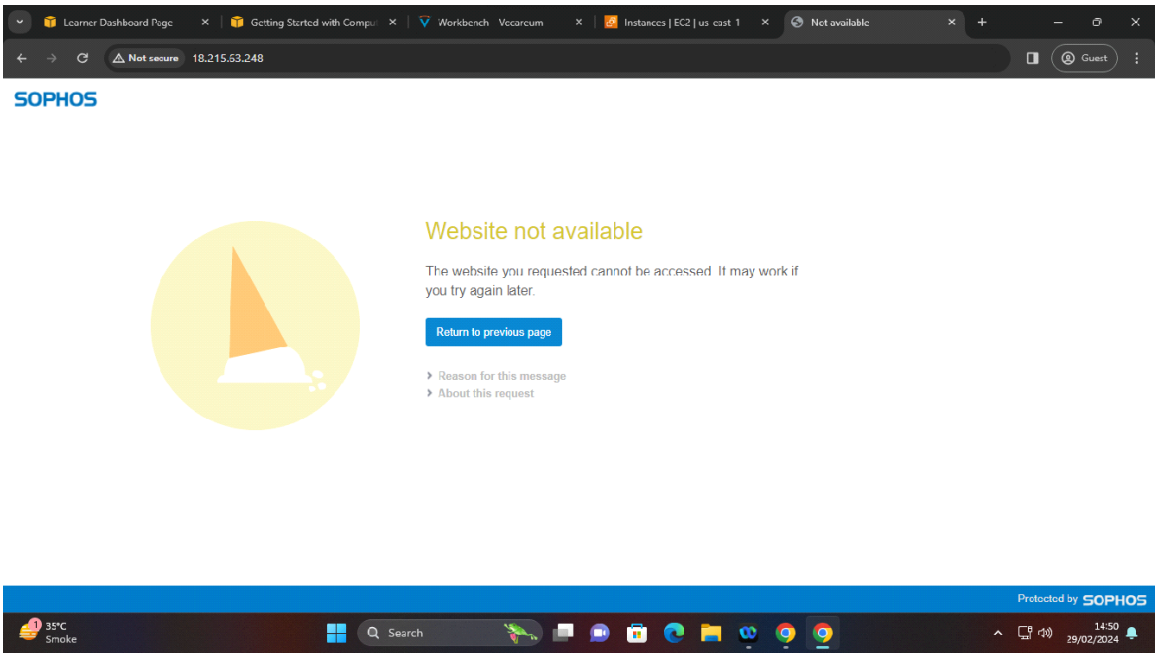
When you launched the EC2 instance, you provided a script that installed a web server and created a simple web page. In this task, you access content from the web server.

33. Select the check box next to the Amazon EC2 **Web-Server** that you created, and then choose the **Details** tab.
34. Copy the **Public IPv4 address** of your instance to your clipboard.
35. In your web browser, open a new tab, paste the IP address you just copied, and then press Enter.

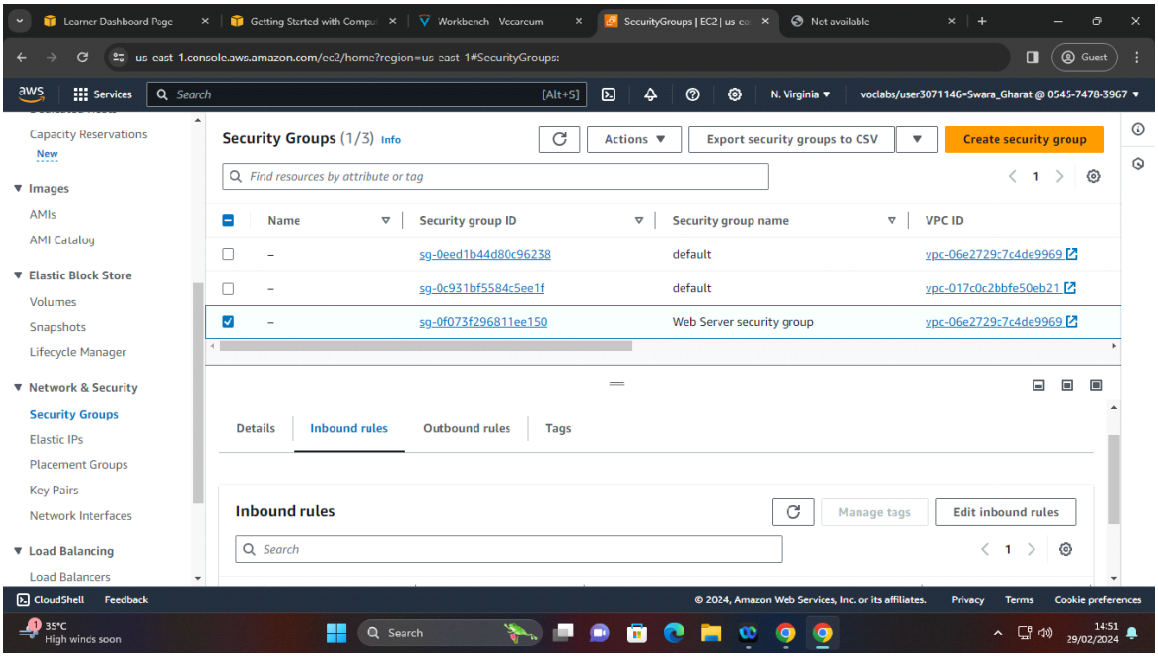
    Question: Are you able to access your web server? Why not?

    You are not currently able to access your web server because the security group is not permitting inbound traffic on port 80, which is used for HTTP web requests. This step is a demonstration of how to use a security group as a firewall to restrict the network traffic that is allowed in and out of an instance.

    To correct this issue, you now update the security group to permit web traffic on port 80.

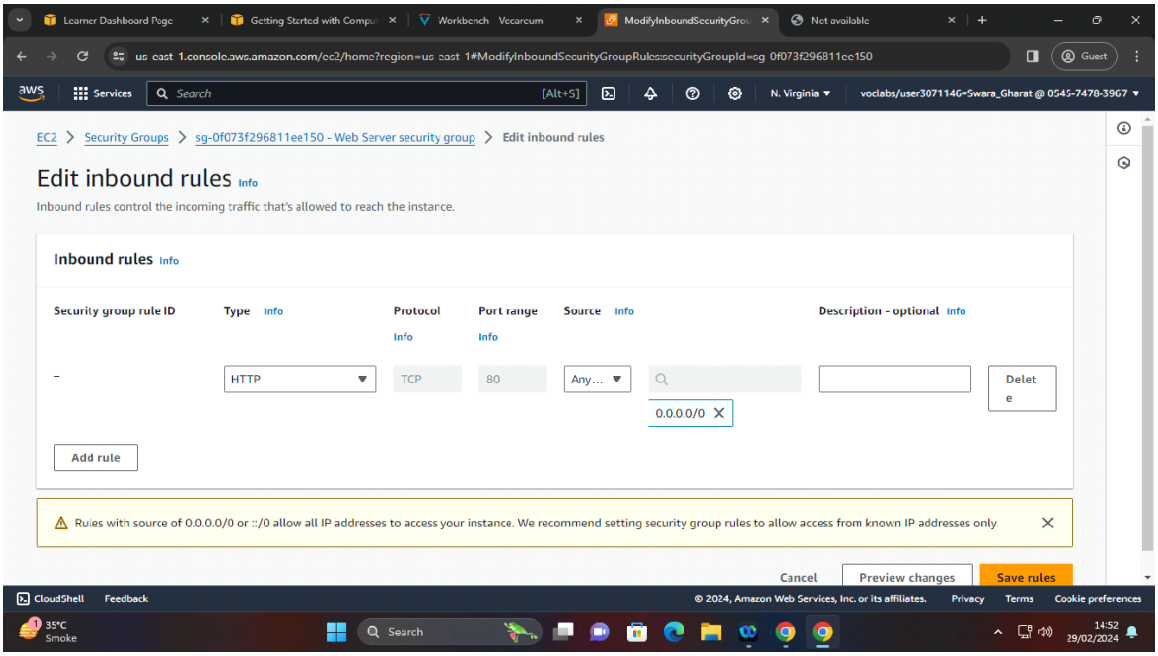


36. Keep the browser tab open, but return to the **EC2 Management Console** tab.
37. In the left navigation pane, choose **Security Groups**.
38. Next to **Web Server security group**, select the check box.

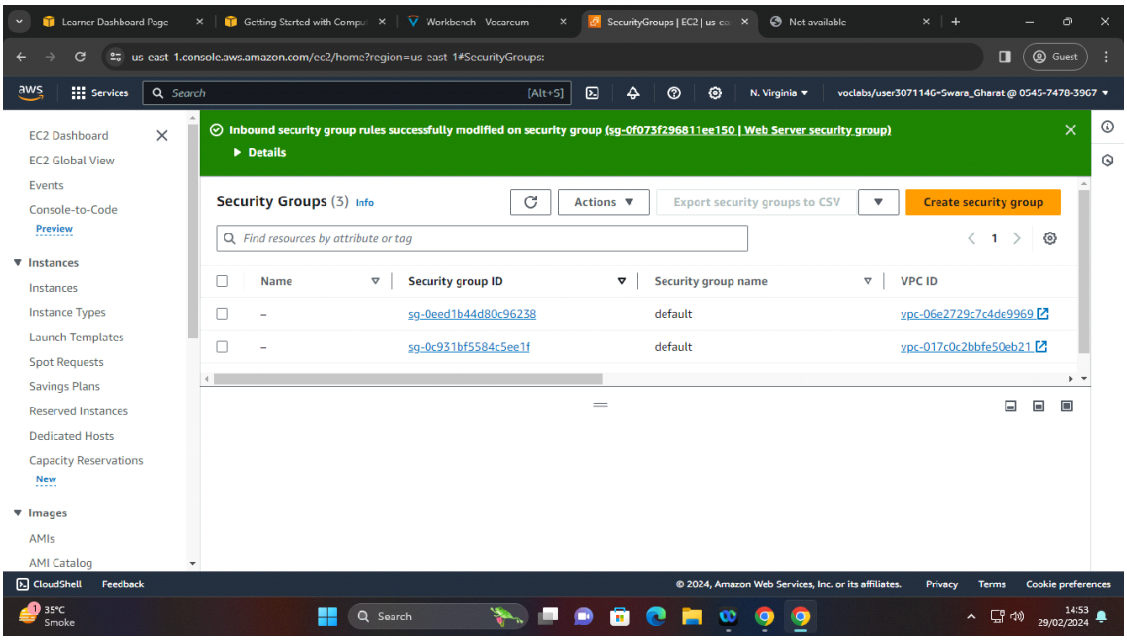


39. Choose the **Inbound rules** tab.
    The security group currently has no rules.
40. Choose **Edit inbound rules**, and then choose **Add rule**, and configure the following options:
    a. **Type:** Choose **HTTP**.

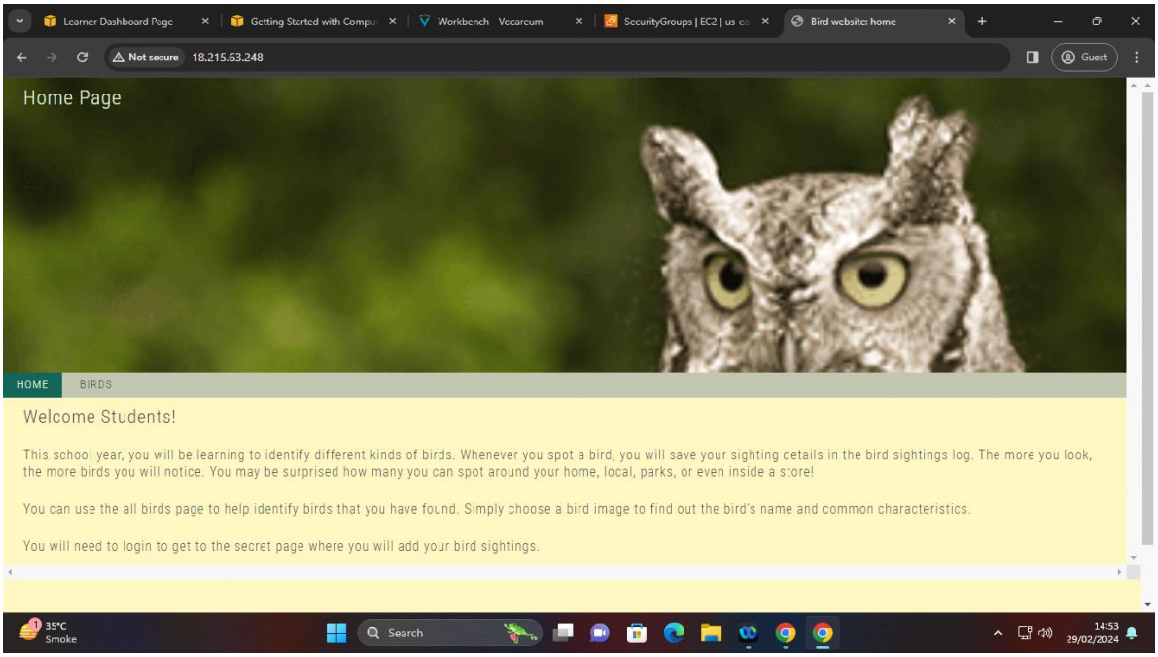
b. **Source:** Choose **Anywhere-IPv4**.



41. **Note:** Notice the *"Rules with source of 0.0.0.0/0 allow all IP addresses to access your instance. We recommend setting security group rules to allow access from known IP addresses only."* While this is true and common best practice, this lab allows access from any IP address (Anywhere) to simplify both the security group configuration and testing of the website running on your EC2 instance.
    In this lab, you can only add a new Ingress rule. You cannot change a rule once it has been created. Double check the configuration before choosing **Save rules**.
42. Choose **Save rules**



43. Return to the web server browser tab with the public IPv4 address that you previously opened, and choose to refresh the page.
    You should now find a web website with the message **Welcome Students!**
    Note: If the web site is not loading, verify that the URL in the address bar begins with `http://` and not `https://`.



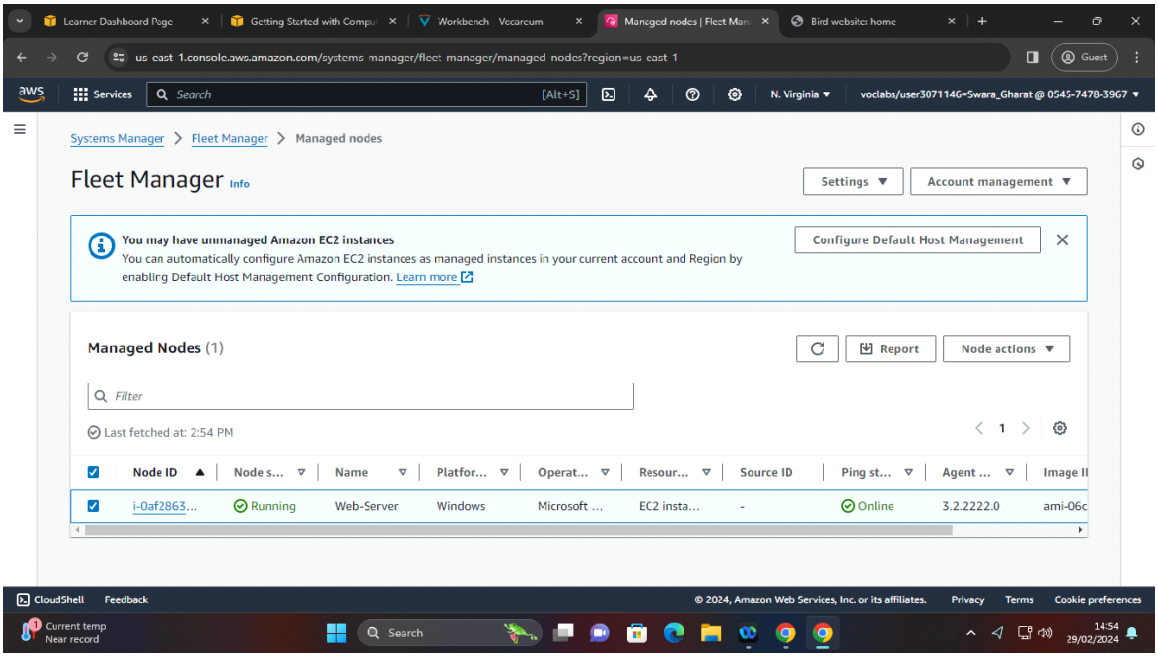
Task 4: Connecting to your instance using AWS Systems Manager Fleet Manager

With the Fleet Manager capability of AWS Systems Manager, you can remotely manage and configure your managed nodes. A managed node is any machine configured for Systems Manager.

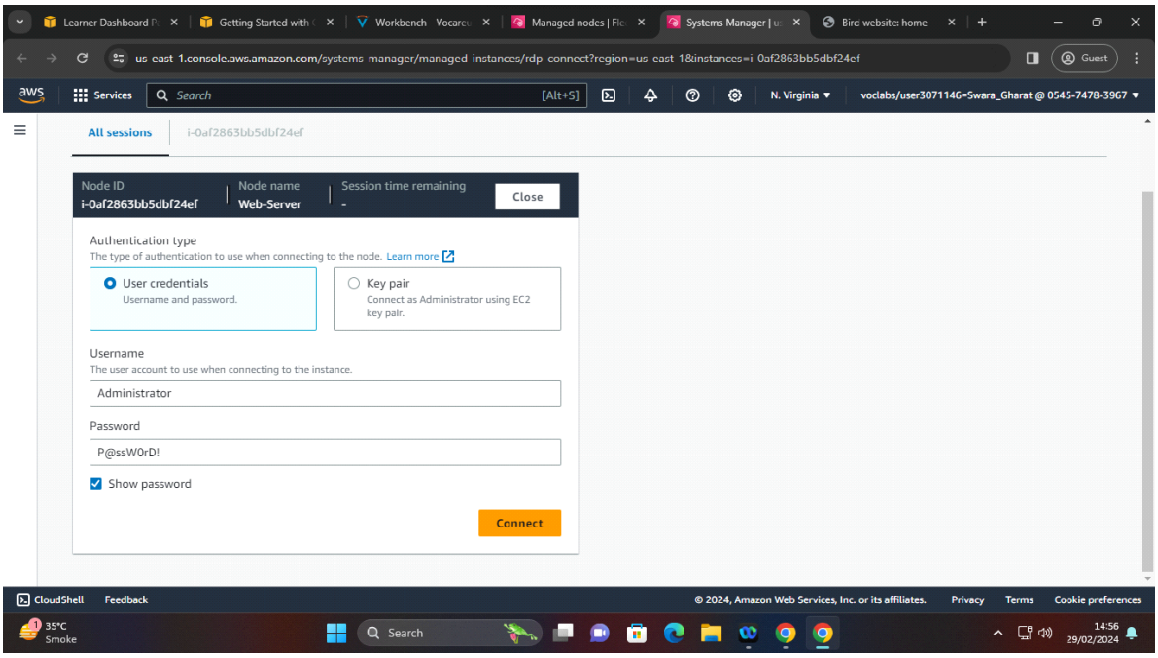
When you started this lab, your AWS user was automatically given permissions to use Systems Manager. In addition, the AWS Identity and Access Management (IAM) policy that you selected when configuring your EC2 instance turned on Systems Manager for your Web-Server instance.

One convenient feature of Fleet Manager is the ability to connect to your EC2 instance using a browser. In this task, you connect to your Windows desktop using Fleet Manager.

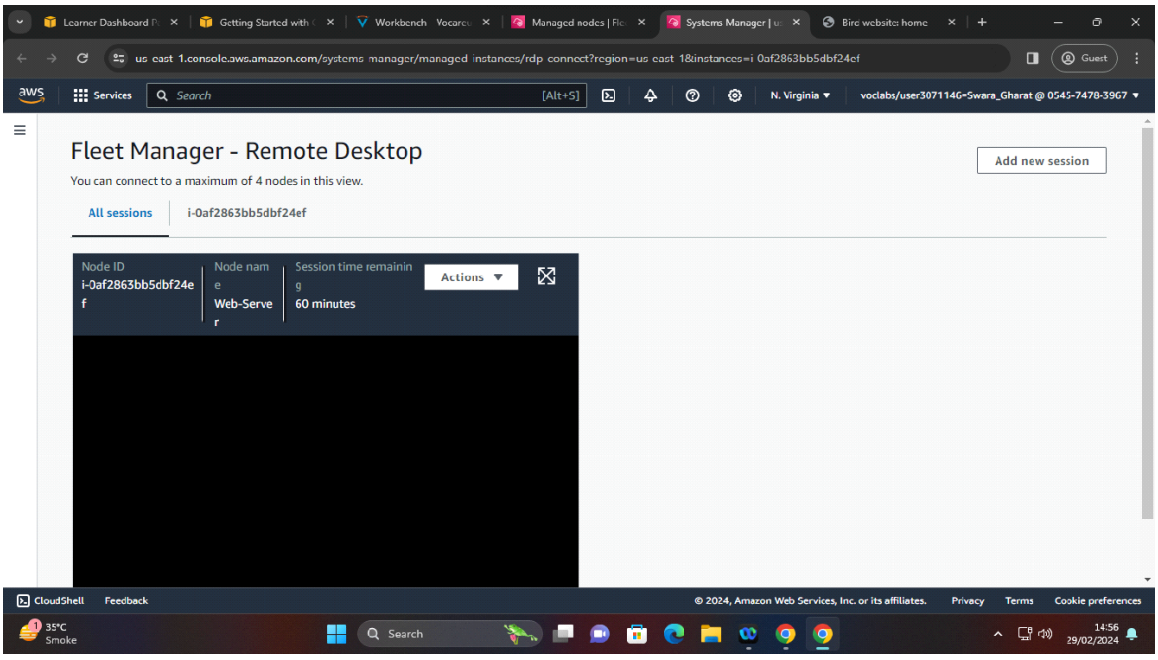
44. In the AWS Management Console on the **Services** menu, search for and select **Systems Manager**.
45. In the left navigation pane, select **Fleet Manager.**



46. Under **Managed nodes**, select your **Web-Server** EC2 instance.
47. From the **Node actions** dropdown list, choose **Connect with Remote Desktop**.
    A new tab opens.
48. Enter the following values:
    a. **Username:** Administrator
    b. **Password:** P@ssW0rD!



49. Choose **Connect**.
    After several seconds, the pane displays the Windows desktop. You can navigate this desktop just like you would on a local computer. As you learned earlier, with Amazon EC2, you can quickly access compute resources. Instead of buying physical hardware and configuring an operating system, all you have to do is launch an EC2 instance, and all of that work is done for you automatically in minutes.



50. To disconnect from your **Web-Server** instance, choose **Actions** and then choose **End session**.

51. In the pop-up window, choose **End session** again .

Task 5: Resizing your instance

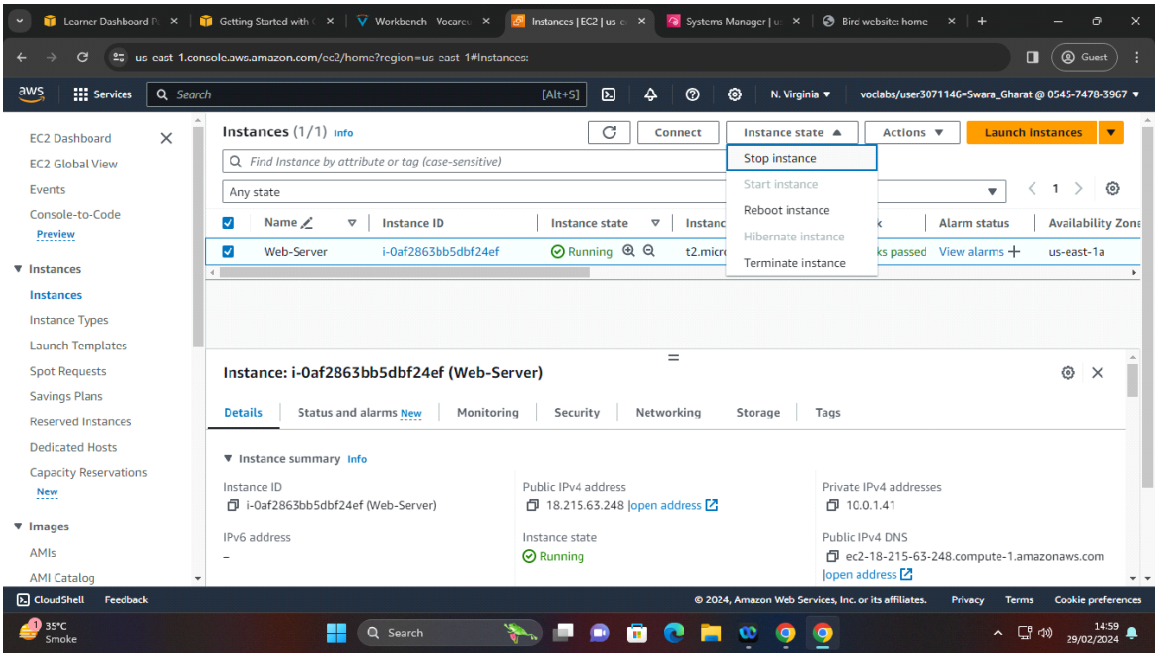
As your needs change, you might find that your instance is overutilized (too small) or underutilized (too large). If so, you can change the instance type. For example, if a t2.micro instance is too small for its workload, you can change it to an m5.medium instance.

Stop your instance

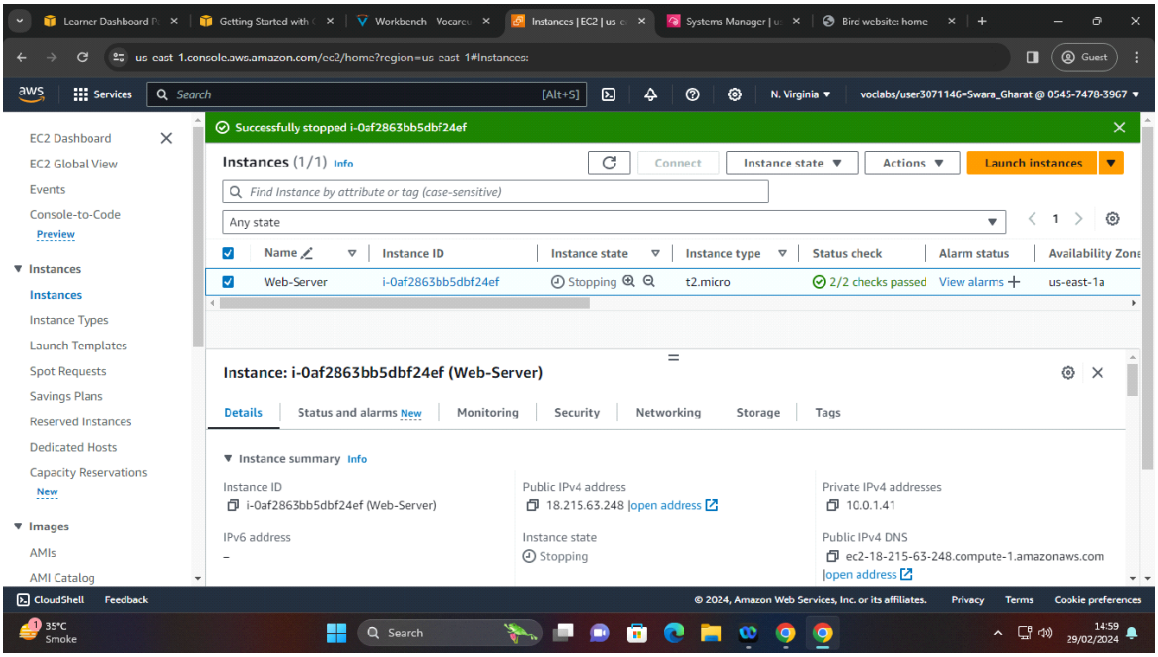
Before you can resize an instance, you must stop it.

When you stop an instance, it is shut down. There is no charge for a stopped EC2 instance, but the storage charge for attached EBS volumes remains.

52. From the AWS Management Console on the **Services** menu, choose **EC2**
53. On the **EC2 Management Console**, in the left navigation pane, choose **Instances**.
54. Select the check box next to your **Web-Server** instance. At the top of the page, choose the **Instance state** dropdown menu, and choose **Stop instance**.



55. In the **Stop instance?** pop-up window, choose **Stop**.
    Your instance performs a normal shutdown and then stops running.
56. Wait for the **Instance state** to display **Stopped**.

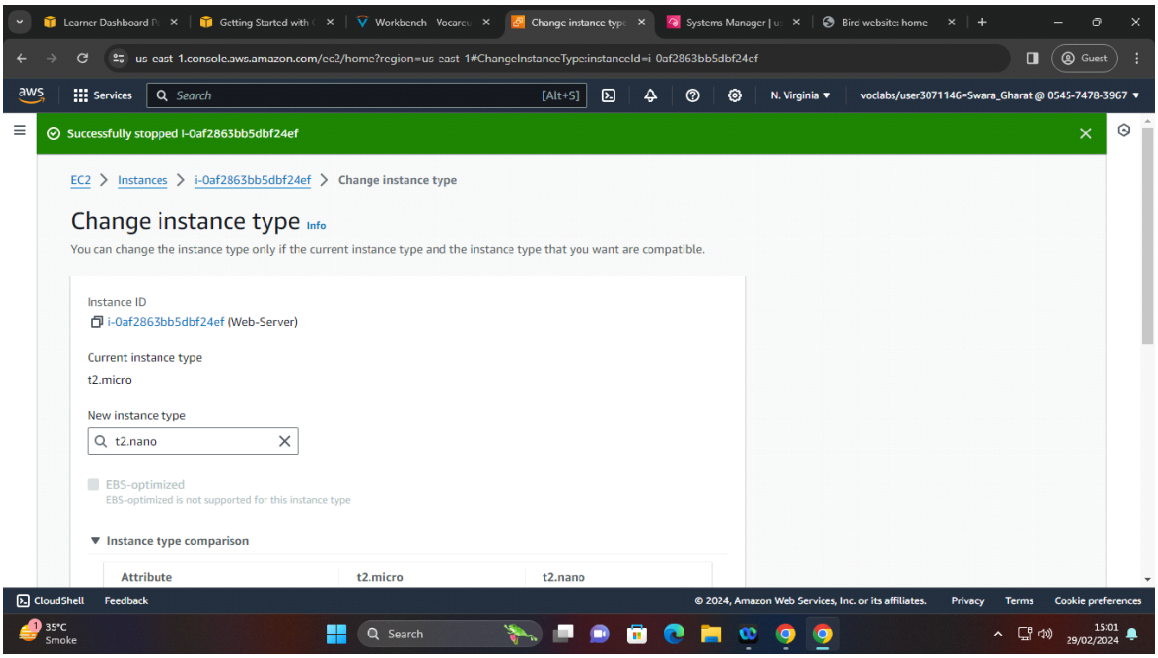


Change the instance type

57. Select the check box next to your **Web-Server**. From the **Actions** dropdown menu, select **Instance settings Change instance type**, and then configure the following option:
    a. **Instance type:** Select **t2.nano**.

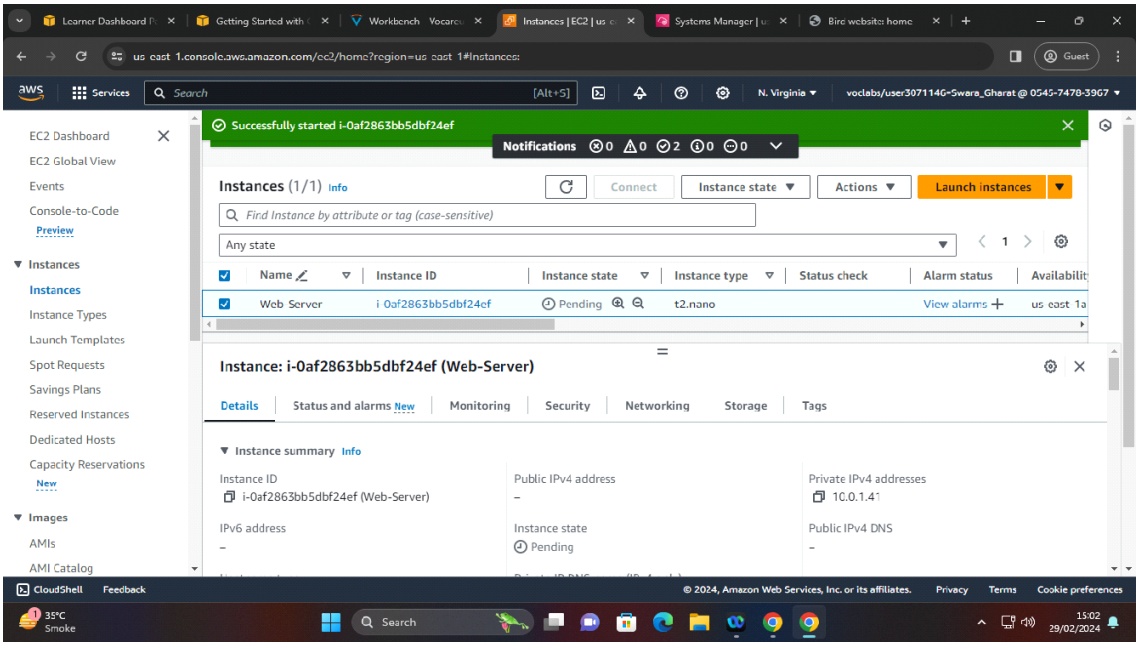
58. Choose **Apply**
    Note: You are restricted from using other instance types in this lab.

Start the resized instance



When the instance is started again, it is a t2.nano instance. You now start the instance again, which has less memory but more disk space.

59. In left navigation pane, choose **Instances**. Next to your **Web-Server**, select the check box.
60. From the **Instance state** dropdown menu, choose **Start instance**.
    Once the instance is restarted, the **Instance state** displays **Running**.

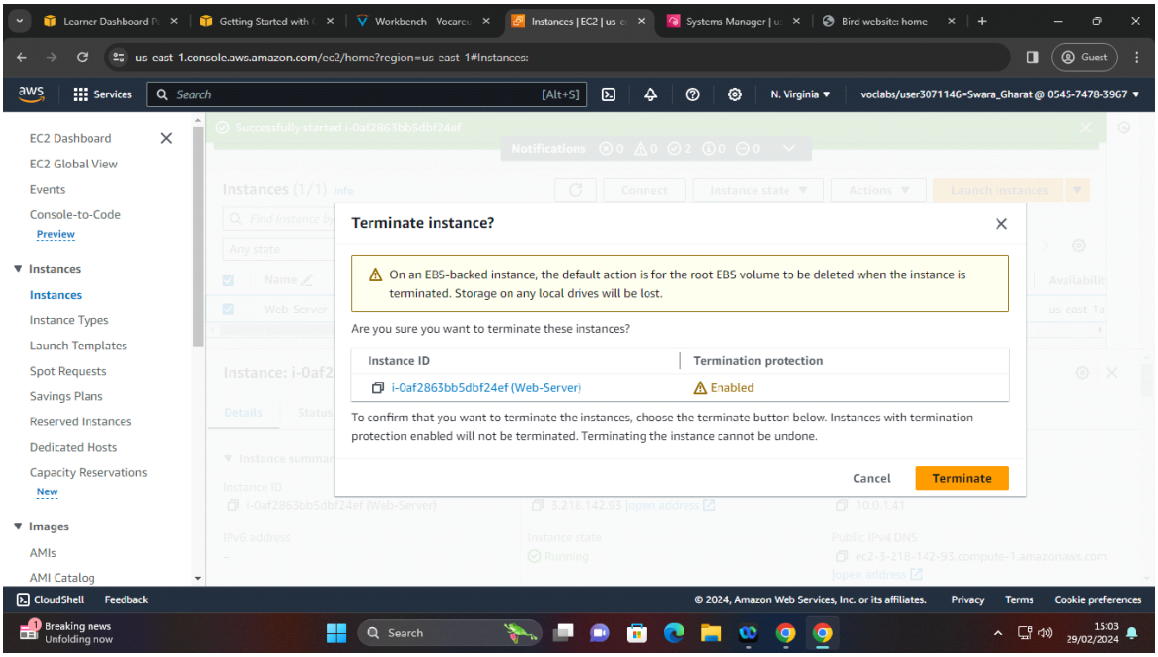


Task 6: Testing termination protection

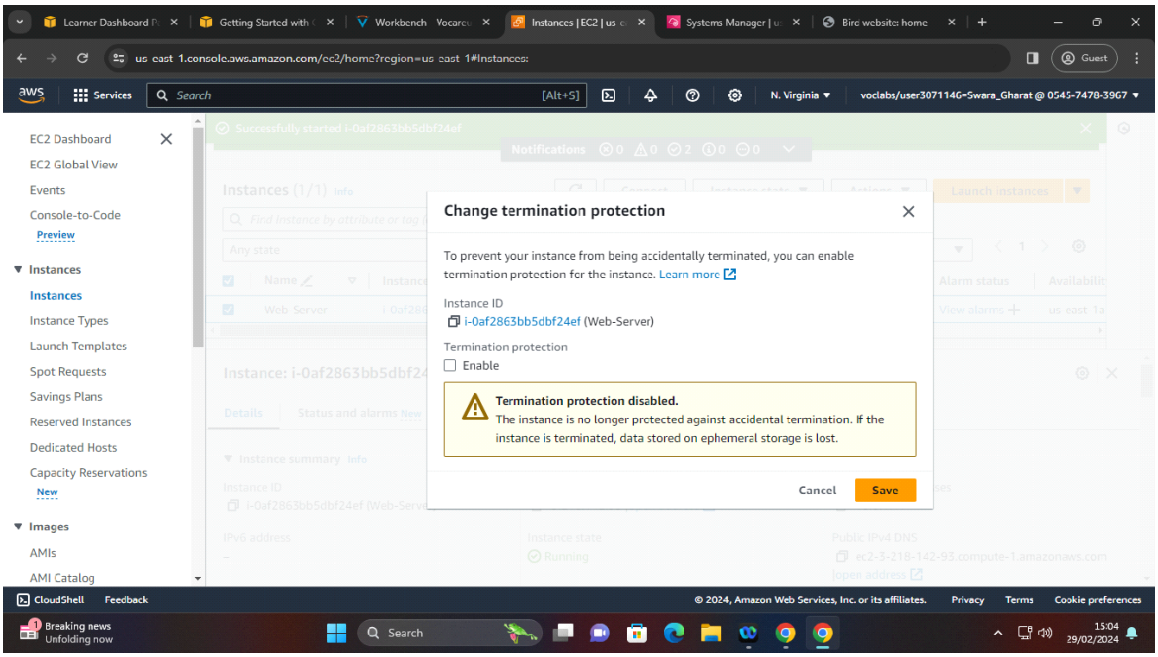
You can delete your instance when you no longer need it. This is referred to as terminating your instance. You cannot connect to or restart an instance after it has been terminated.

In this task, you learn how to use termination protection.

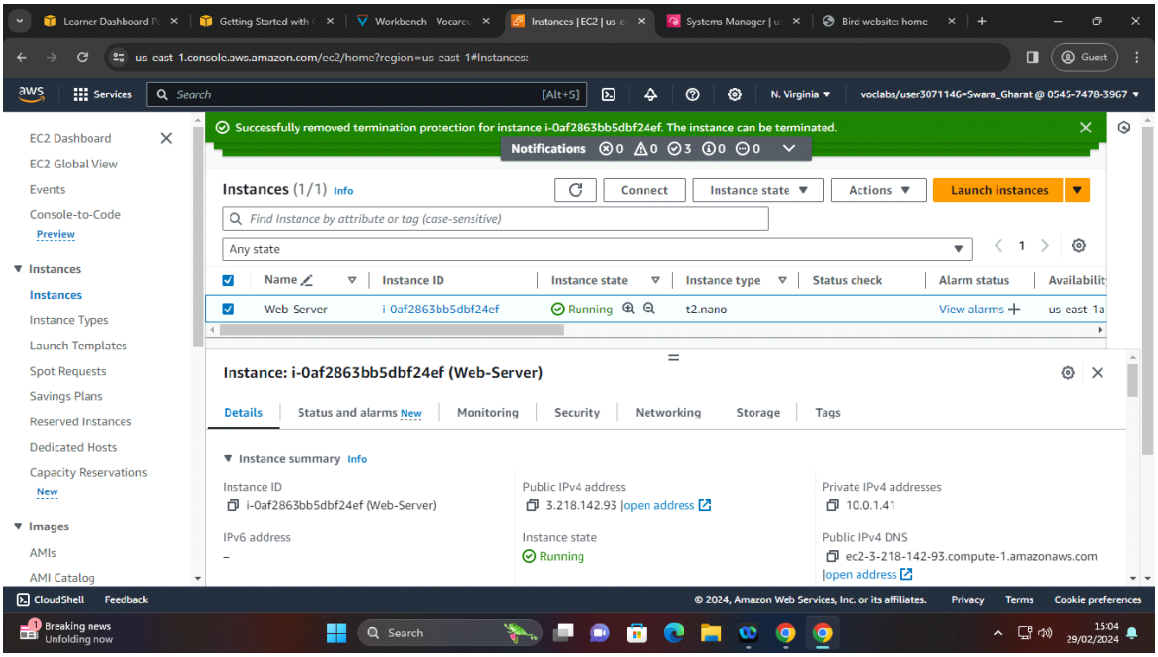
61. Select the check box next to your **Web-Server** instance. From the **Instance state** dropdown menu, choose **Terminate instance**.
62. Notice that **Termination protection** is enabled for this instance.
    This is a safeguard to prevent the accidental termination of an instance. If you really want to terminate the instance, you need to turn off termination protection.
    You can easily enable and disable termination protection from the **Actions** dropdown menu.



63. From the **Actions** dropdown menu, choose **Instance settings**, and then choose **Change termination protection**.
64. Clear the check box for **Enable**.
65. Choose **Save**.



66. Now, try to terminate the instance again.
    The instance state will now successfully be terminated.



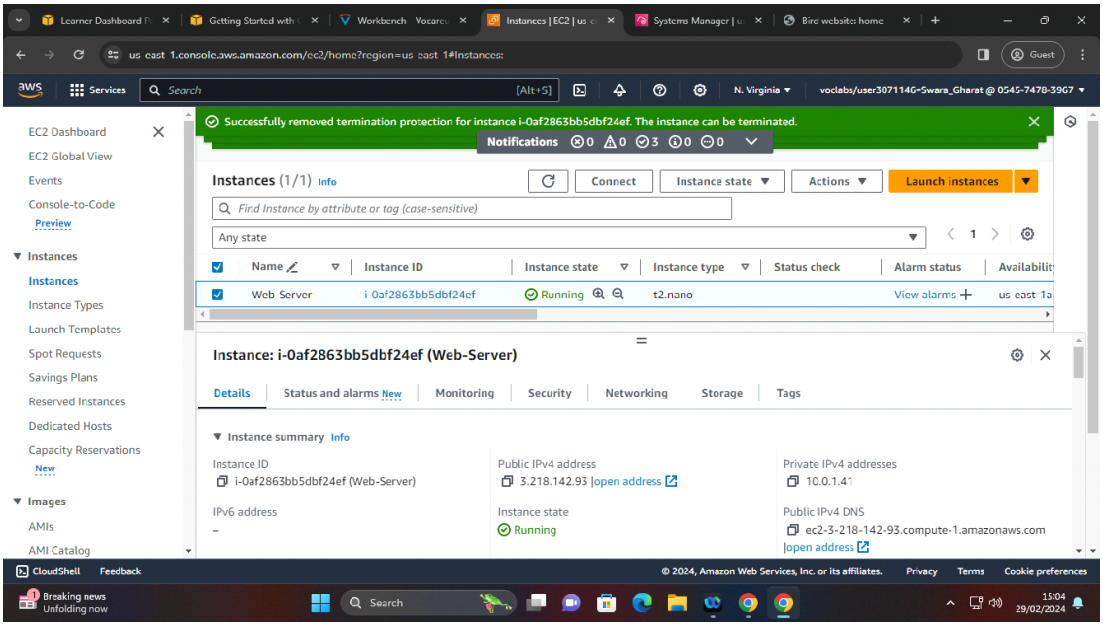
Task 7: Exploring EC2 limits

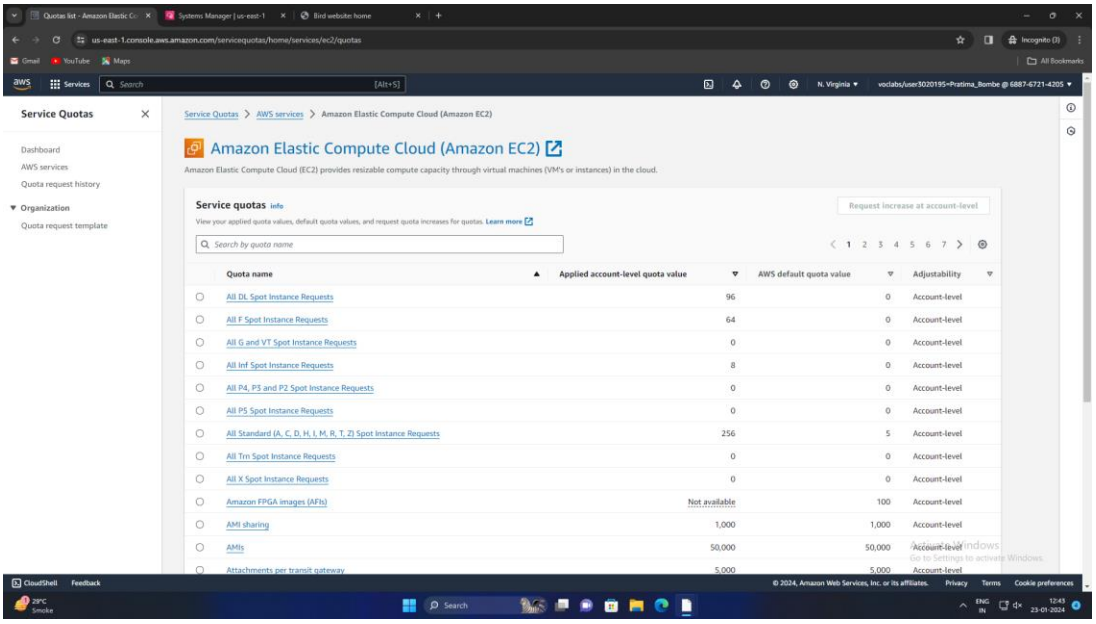
Amazon EC2 provides different resources that you can use. These resources include images, instances, volumes, and snapshots. When you create an AWS account, there are default limits on these resources on a per-Region basis.

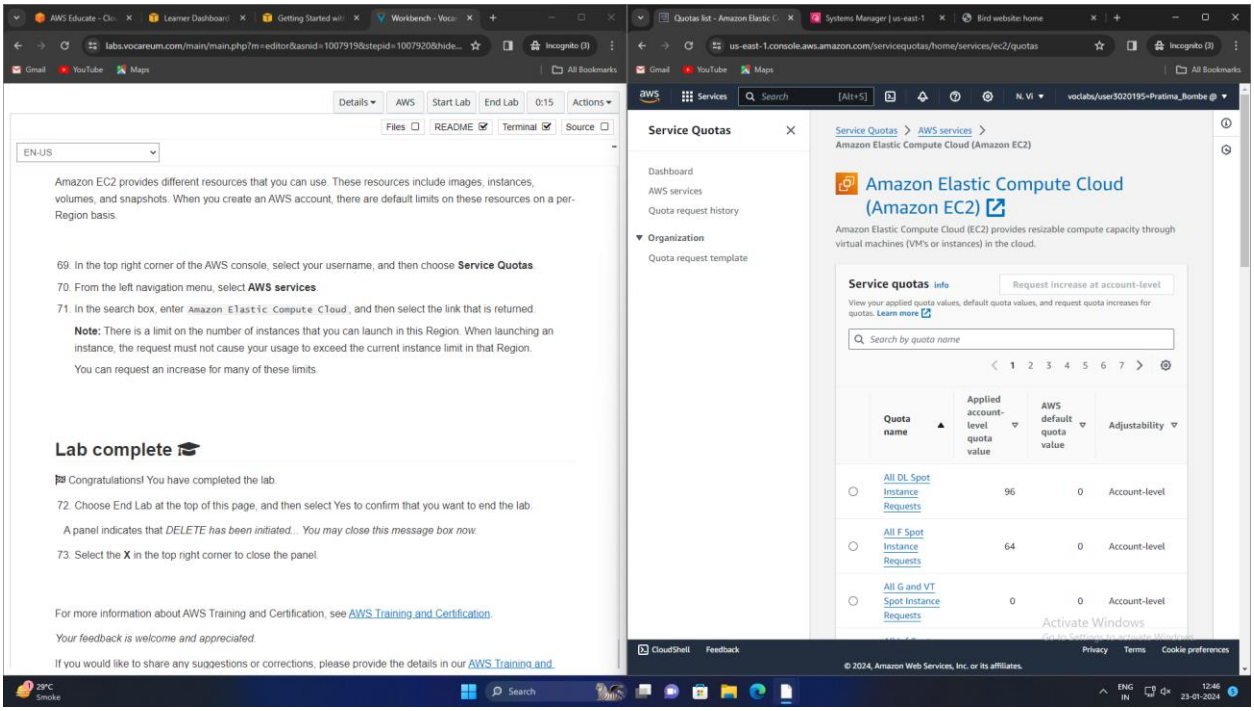
67. In the top right corner of the AWS console, select your username, and then choose **Service Quotas**.
68. From the left navigation menu, select **AWS services**.
69. In the search box, enter Amazon Elastic Compute Cloud, and then select the link that is returned.

    Note: There is a limit on the number of instances that you can launch in this Region. When launching an instance, the request must not cause your usage to exceed the current instance limit in that Region.

    You can request an increase for many of these limits.







Conclusion: In conclusion, the experiment on studying and implementing Infrastructure as a Service (IaaS) with Amazon EC2 Compute has illuminated the immense potential of cloud computing. Through this exploration, we have gained a profound understanding of the agility, scalability, and cost-efficiency that cloud platforms offer. Amazon EC2 Compute, in particular, has proven to be a robust solution for provisioning and managing virtual servers, enabling rapid deployment and dynamic resource allocation. This experiment underscores the transformative impact of cloud technologies on traditional IT infrastructure, emphasizing their ability to streamline operations, enhance flexibility, and optimize resource utilization. As organizations continue to embrace the cloud, the insights gleaned from this experiment will undoubtedly shape future strategies for infrastructure management and resource allocation.