

**Amazon AWS 4-6 Documentation**

Ryan Chen

1/3/2024

**Purpose:**

As a continuation of labs 1-3, this series of labs aims to practice previously learned fundamentals such as EC2 usage as well as incorporating new technologies such as Amazon EBS, RDS, and ELB. At the same time, it provides a thorough introduction to configuring real world services such as web servers connected to databases and load balanced services.

**Background Information:**

Three new services are introduced in these series of labs: Amazon EBS, RDS, and ELB.

Amazon EBS (Elastic Block Storage) is a service that provides persistent storage for EC2 cloud instances. EBS has numerous advantages over traditional storage, such as being more scalable, fault-tolerant, and easy to configure. EBS volumes are network attached and persistent across system restarts, allowing your data to be securely stored when the EC2 cloud instance is powered off. Furthermore, EBS volumes are automatically replicated in single availability zones, providing greater redundancy for accessing data. The resiliency of these volumes makes them suitable for use in the cloud.

On the other hand, Amazon RDS (Relational Database Service) is a storage service that allows creation of easily scalable and manageable relational databases in the cloud. First off, databases primarily take two forms: relational and nonrelational databases. Relational databases work exactly like their name, where data is stored in relation to other data. Data is stored in rows and columns, which allows it to be easily queried for later. Primarily, columns contain data attributes and rows contain data values. This differs from the nonrelational structure, in which data is stored using various methods. Popular methods include key-value pairs, document-based storage, and graph representations. Relational databases are more suitable for long term structured data which rarely changes form, whereas nonrelational databases are more flexible and allow for changing data storage forms. Amazon RDS allows you to configure six different popular relational databases easily: Amazon Aurora, Oracle, Microsoft SQL Server, PostgreSQL, MySQL and MariaDB. Data is stored in the cloud and is easily resizable to meet individual customer needs.

Finally, Amazon ELB (Elastic Load Balancing) provides high performance load balancing to applications that require constantly changing computing resource amounts. Using ELB, computing load is distributed evenly across currently allocated/running EC2 instances, guaranteeing that resources are fully utilized. ELB works in conjunction with Amazon’s auto resource scaling to increase or decrease allocated resources based on current computing load. Auto scaling automatically manages EC2 instances during traffic spikes to provide greater fault tolerance. For instance, an E-commerce website using AWS might utilize ELB and auto scaling to provide fault tolerance during a product launch where traffic reaches an all-time high. Without this service, traffic load might exceed computing resources, causing the web server to drop requests or even crash entirely.

**Lab 4:**

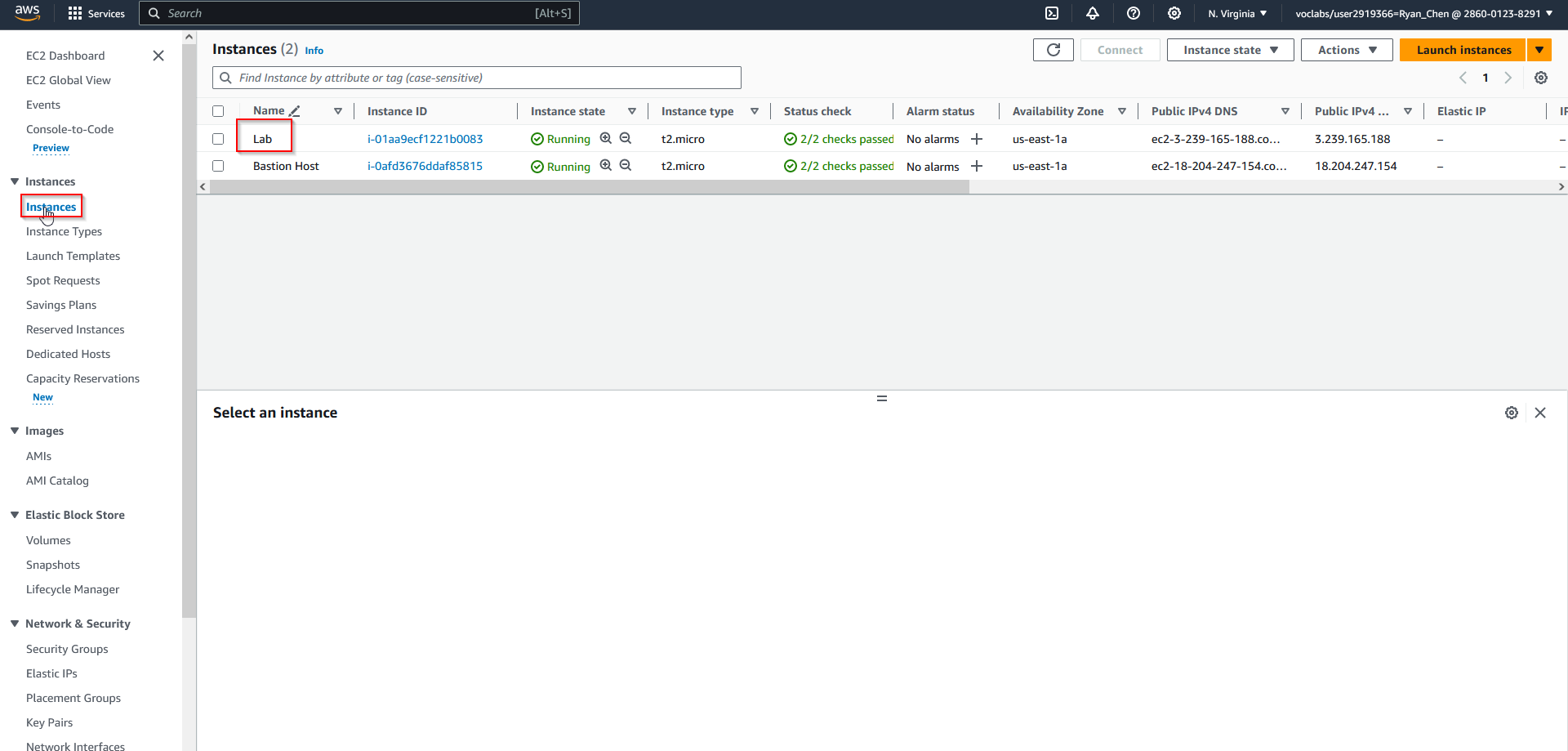
**Network Diagram:**



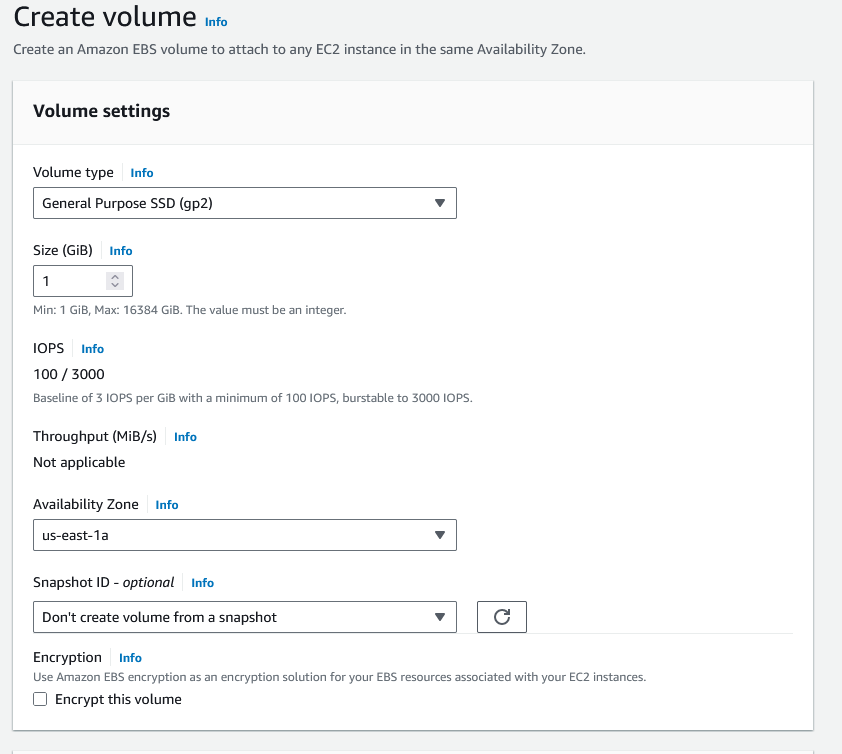
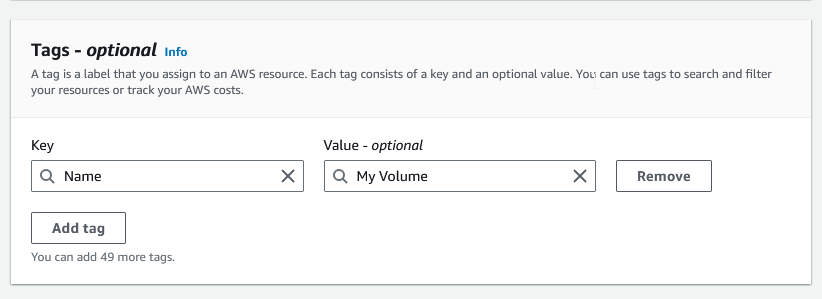
**Lab Instructions:**

This lab provides an introduction to Amazon EBS, which is a service that provides high performance and availability storage. Within this lab, we configure a new EBS volume and attach it to an EC2 instance. Furthermore, we utilize the snapshot feature to restore files.

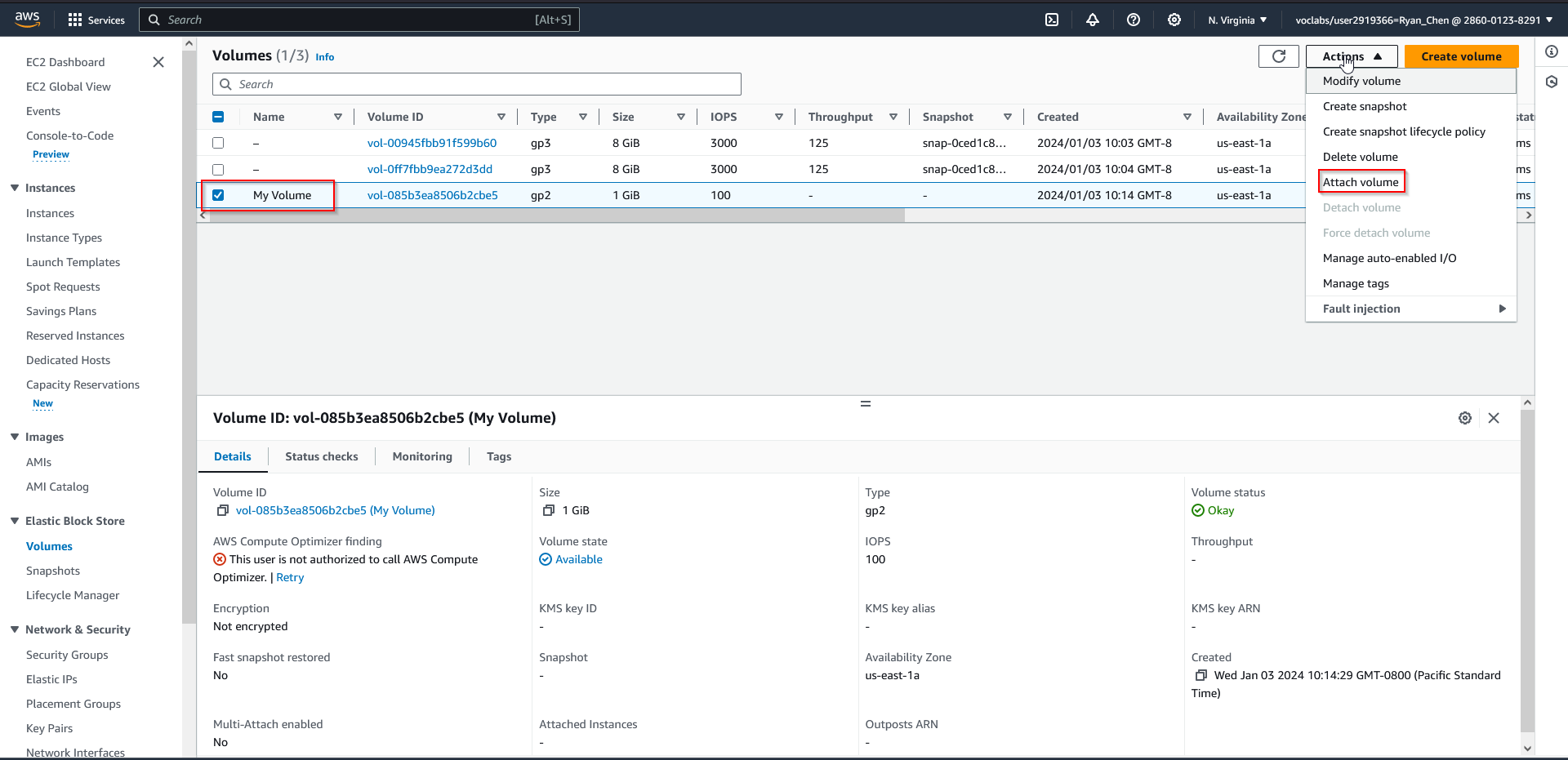
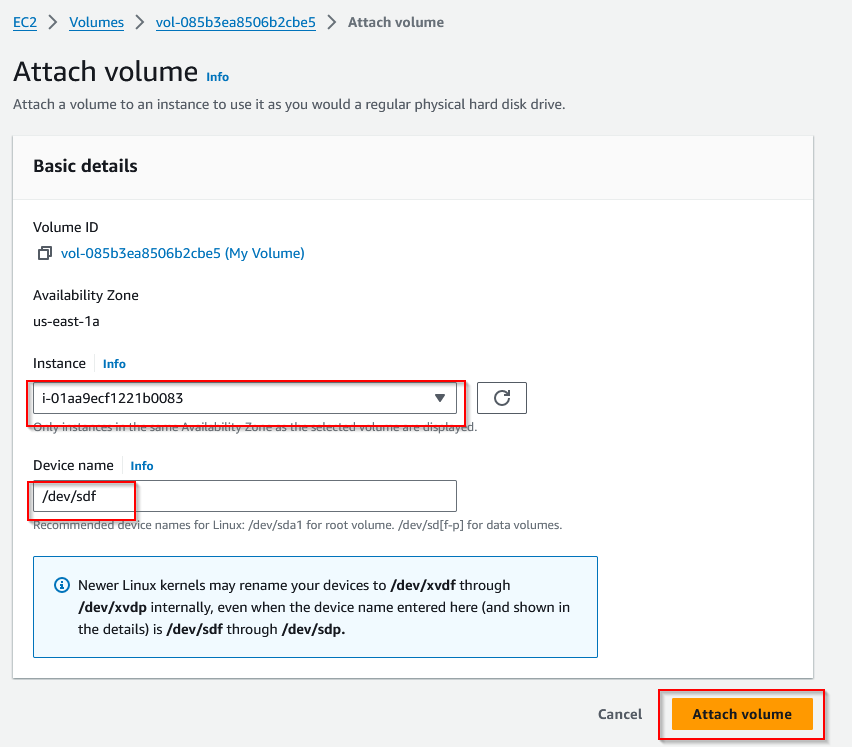
1. Navigate to EC2, and then click on the Instances tab. Observe the pre-created instance.



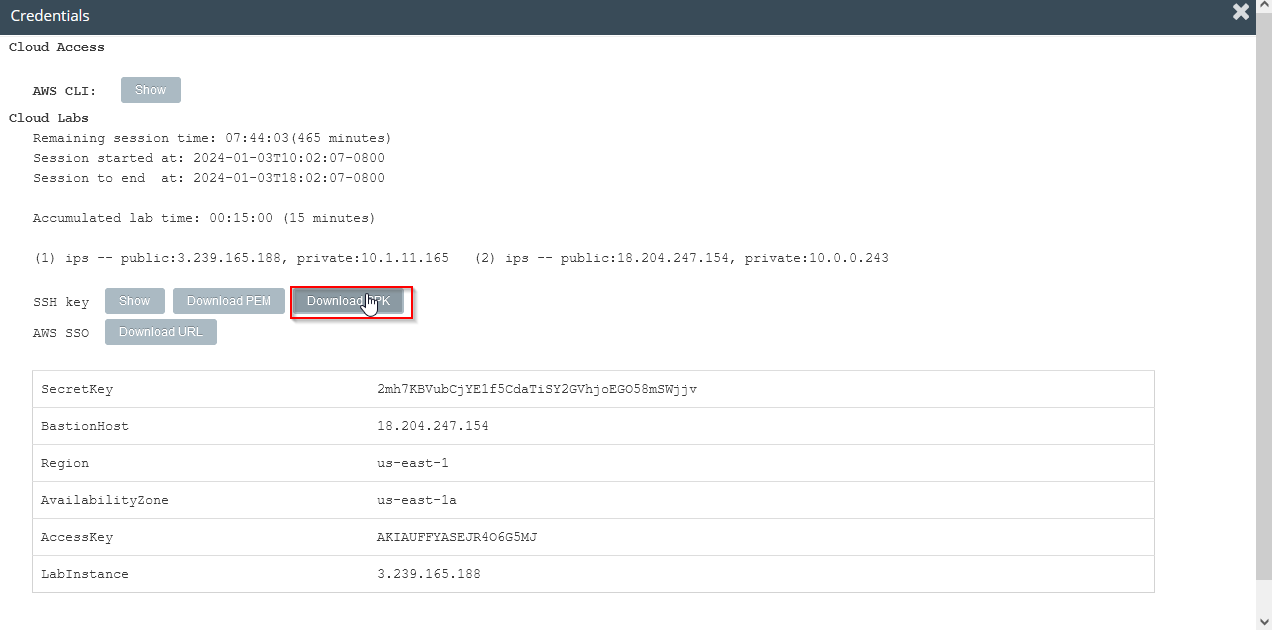
1. Choose Volumes and create a new volume with given settings.

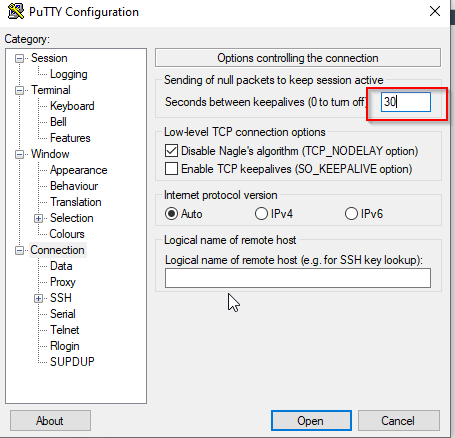
1. After creating the new volume, attach it and choose the Lab instance.

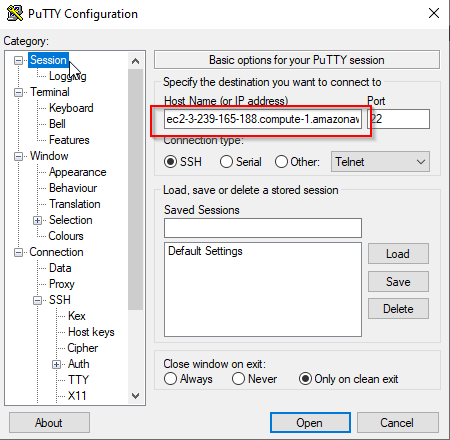
1. Now, connect to the Lab instance using PuTTY on Windows or SSH on Linux. Download the certificate and connect.



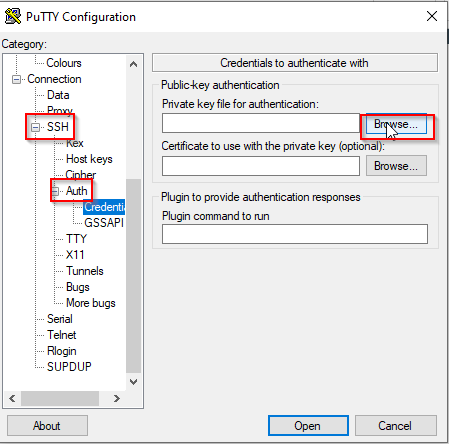
1. Set the PuTTY timeout to 30 seconds.



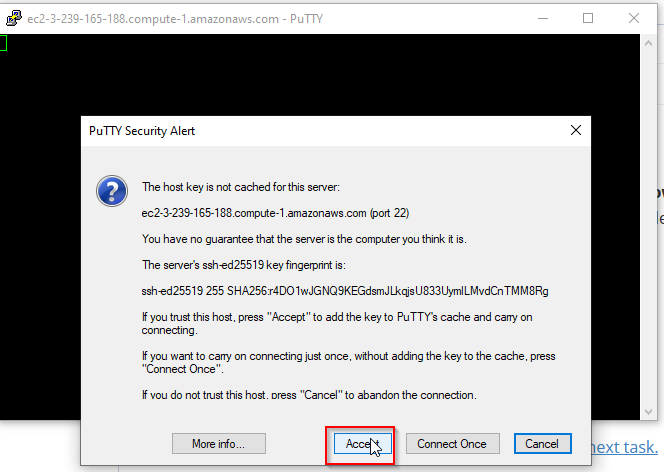
1. Paste the public DNS of the Lab instance.



1. Select the key that was downloaded earlier.



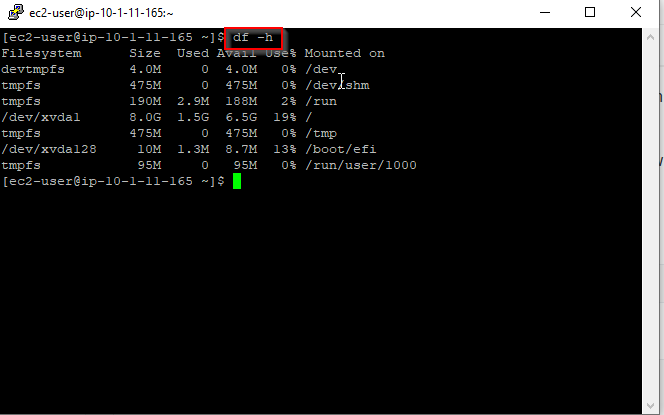
1. When prompted, accept and connect to the machine.



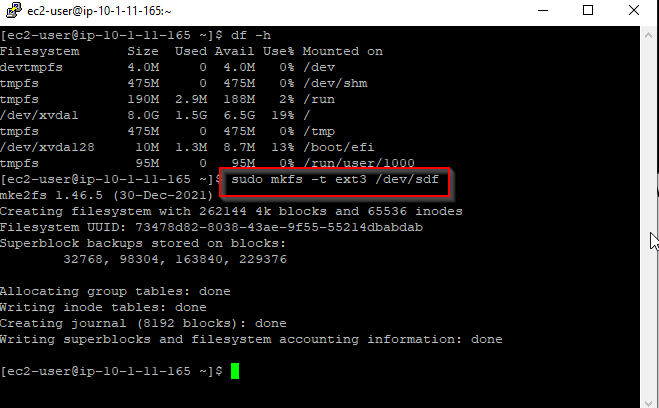
1. Login as “ec2-user”



1. To check available storage on the instance, type “df -h”. Observe that the created EBS volume is not yet attached.



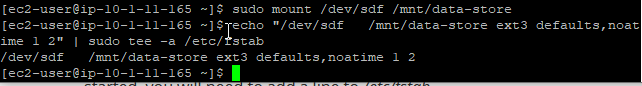
1. Create an ext3 file system on the new volume.



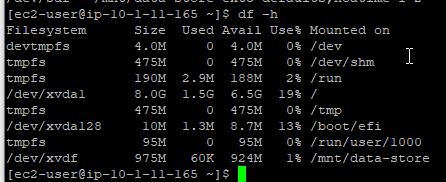
1. Create a new directory for mounting the new storage volume.



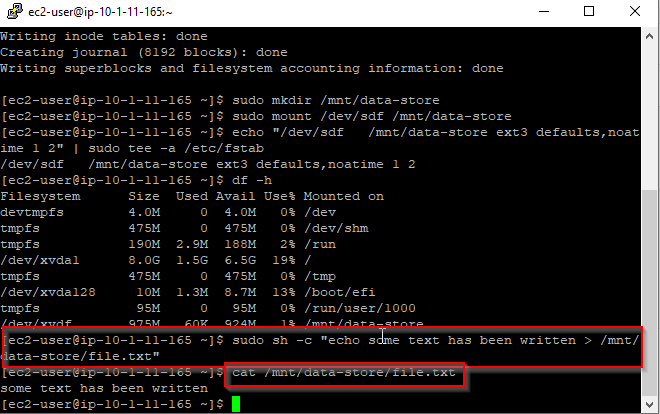
1. Mount the new volume and set it to mount automatically when booting.



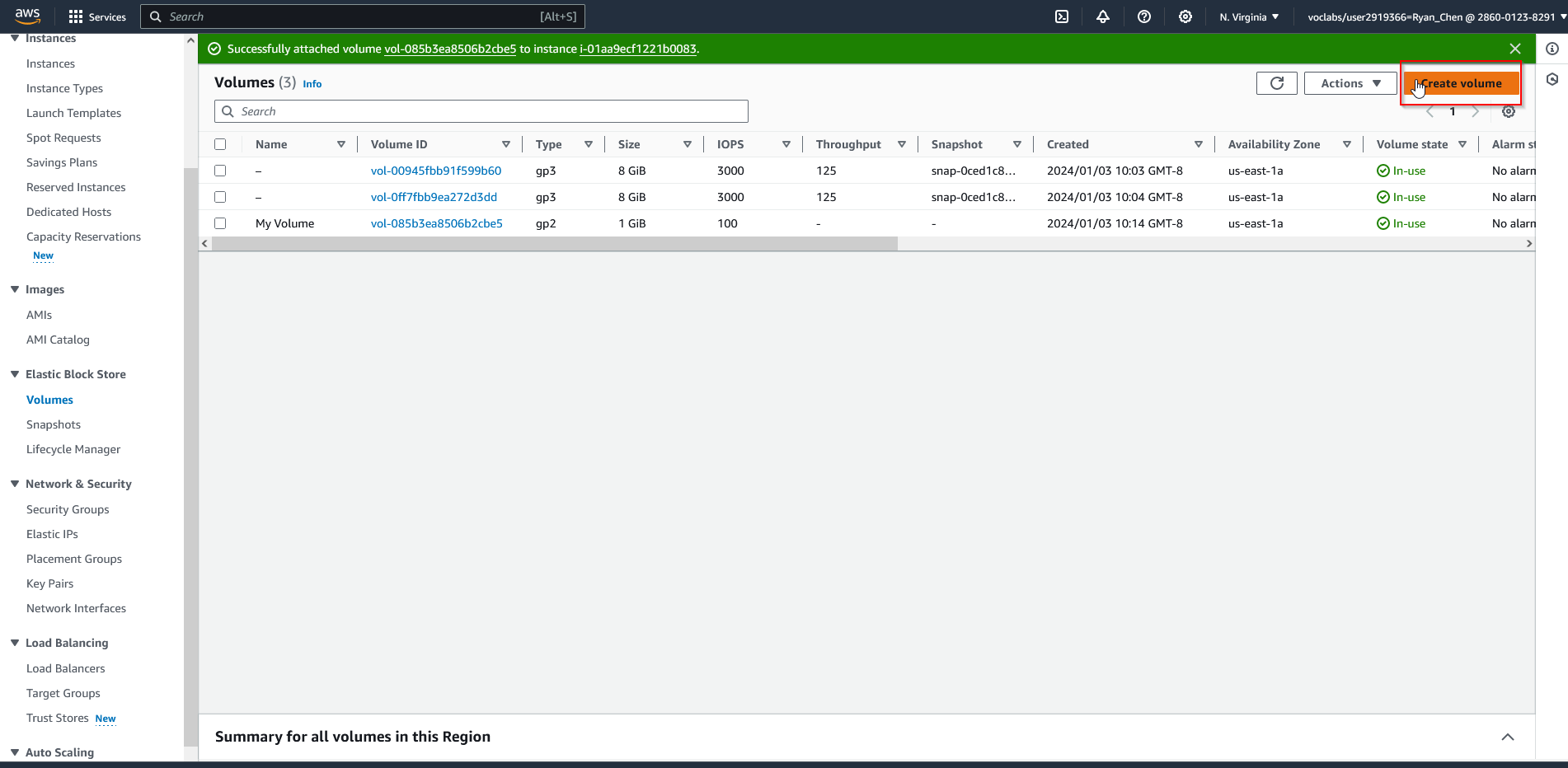
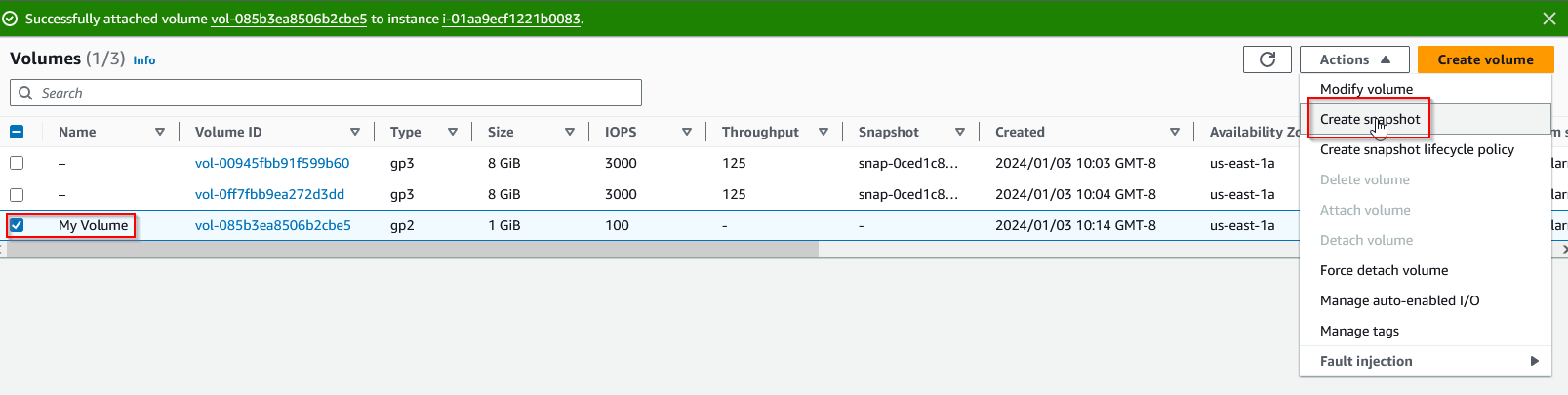
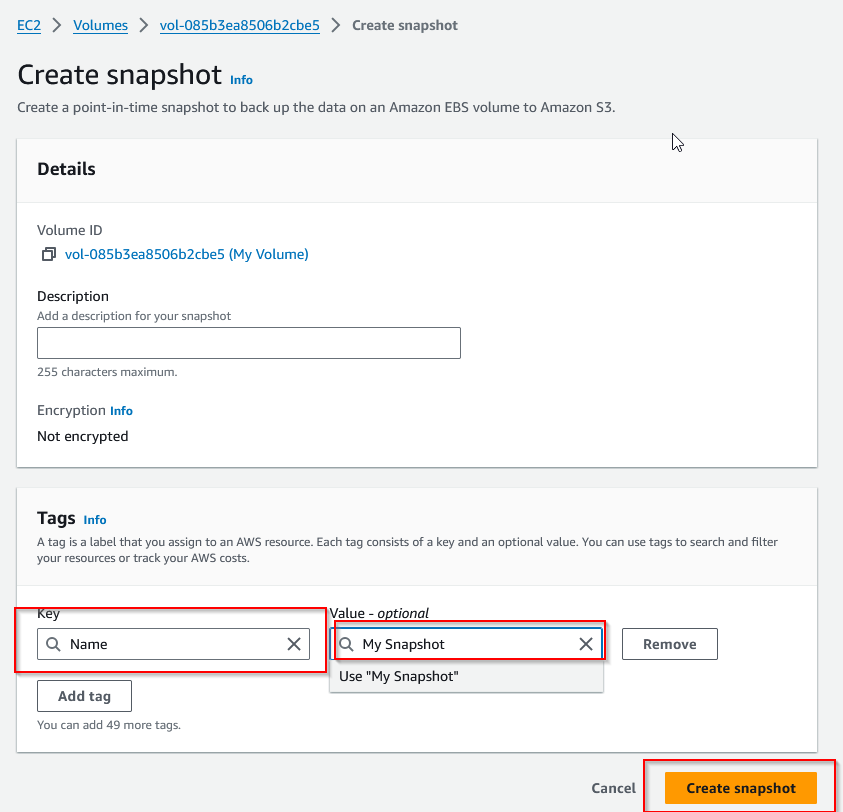
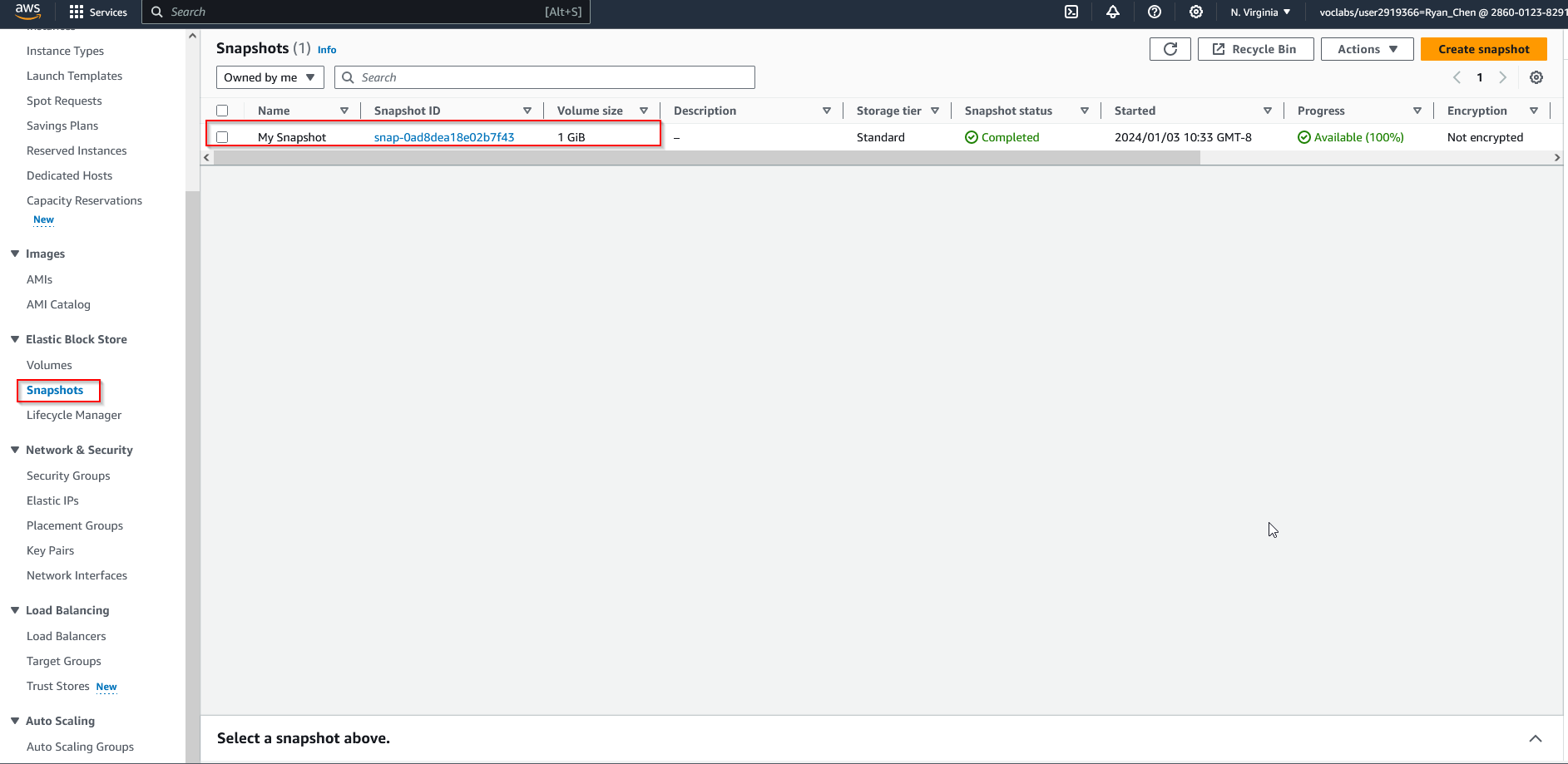
1. Observe that the volume has now been added under “/dev/xvdf”.



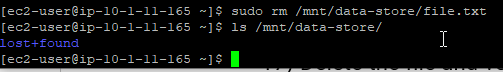
1. Write a file to the new volume and check that it is functioning properly.



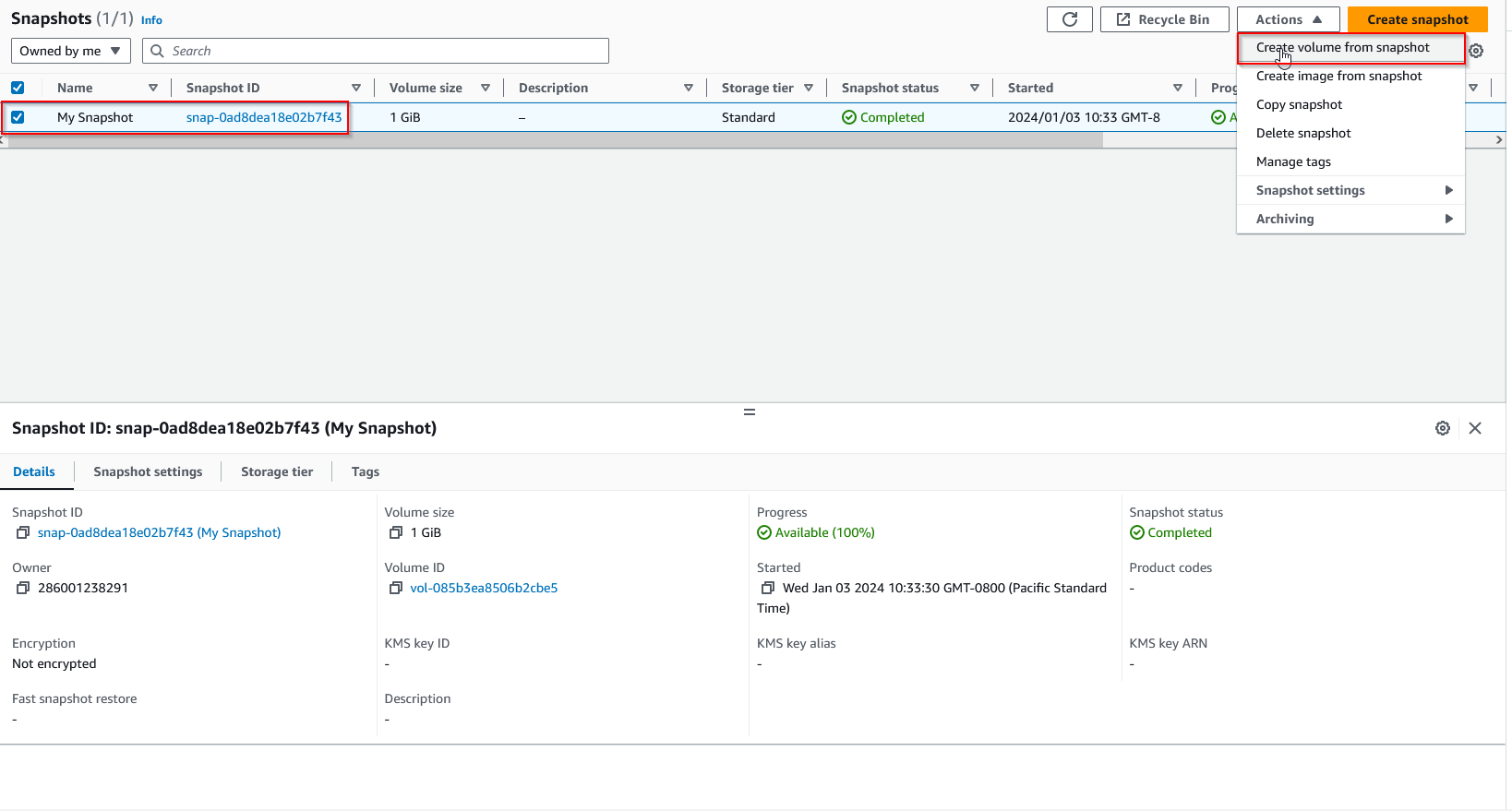
1. Now, create a new backup volume by navigating to the Volumes tab.

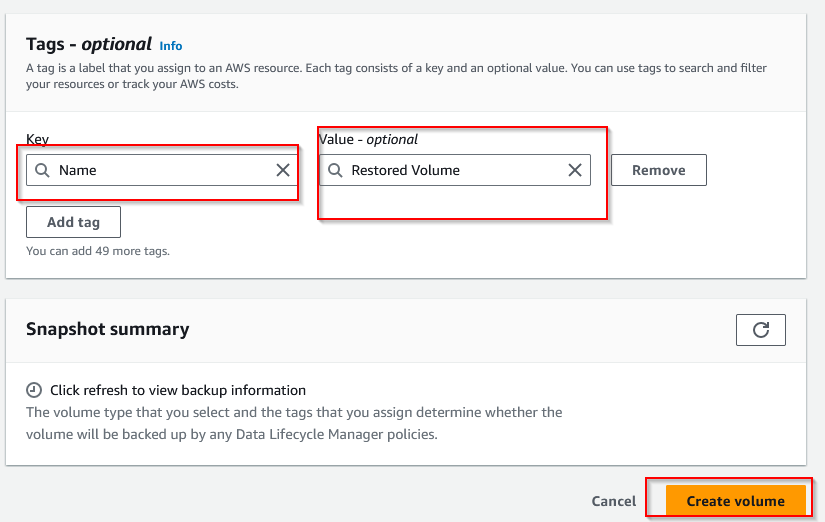
   

1. Delete the file and verify it has been deleted. Now, attempt to restore it using the backup storage snapshot.

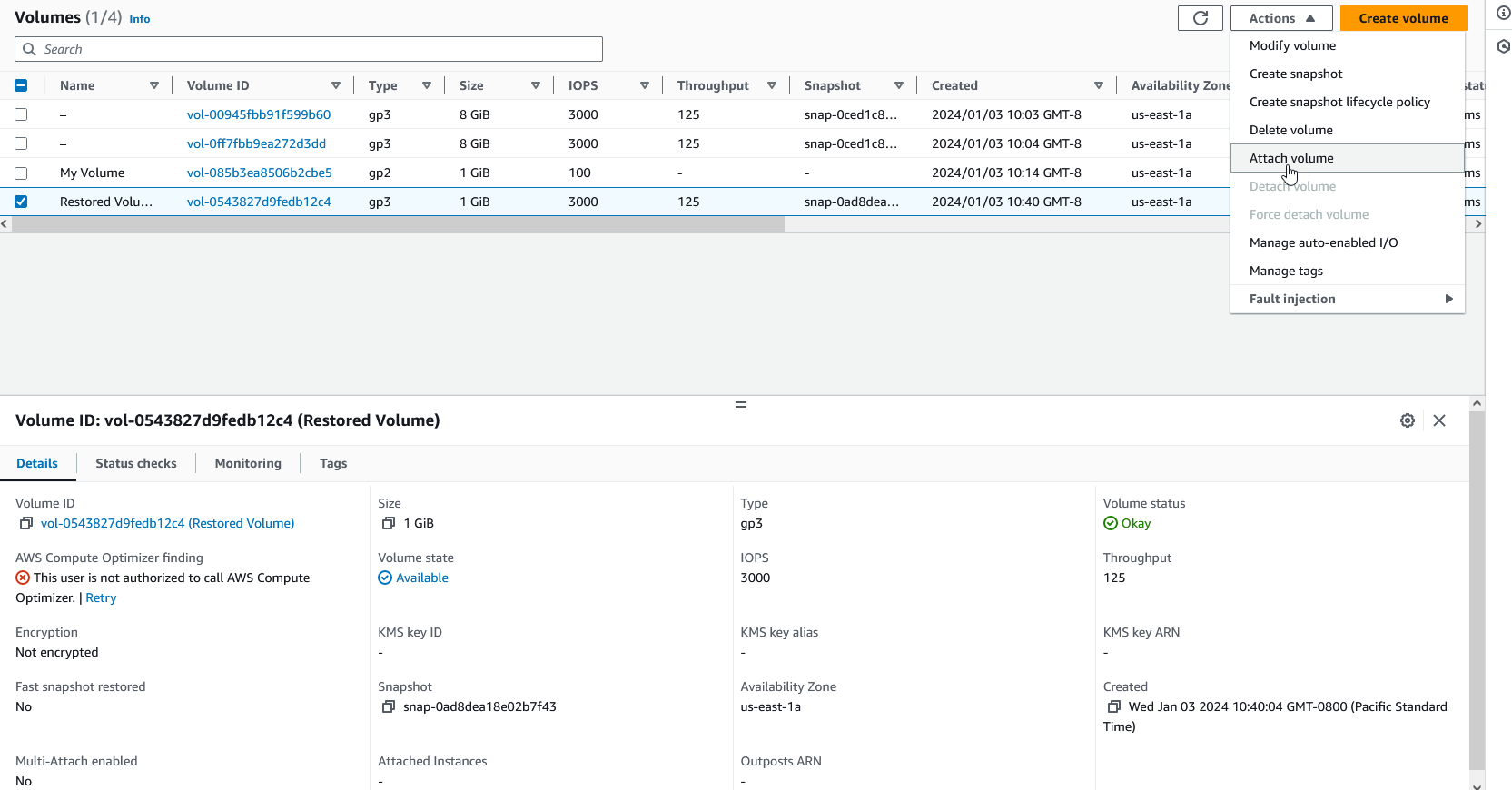


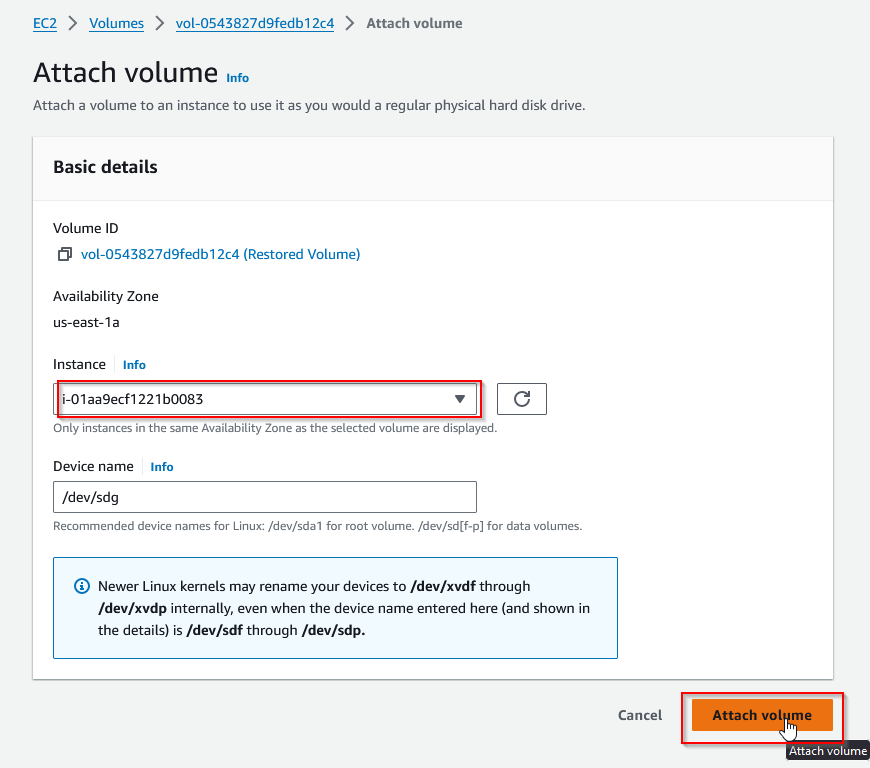
1. Create a new volume from the backup snapshot.



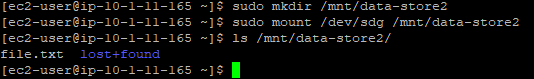


1. Attach the restored volume to the Lab instance.



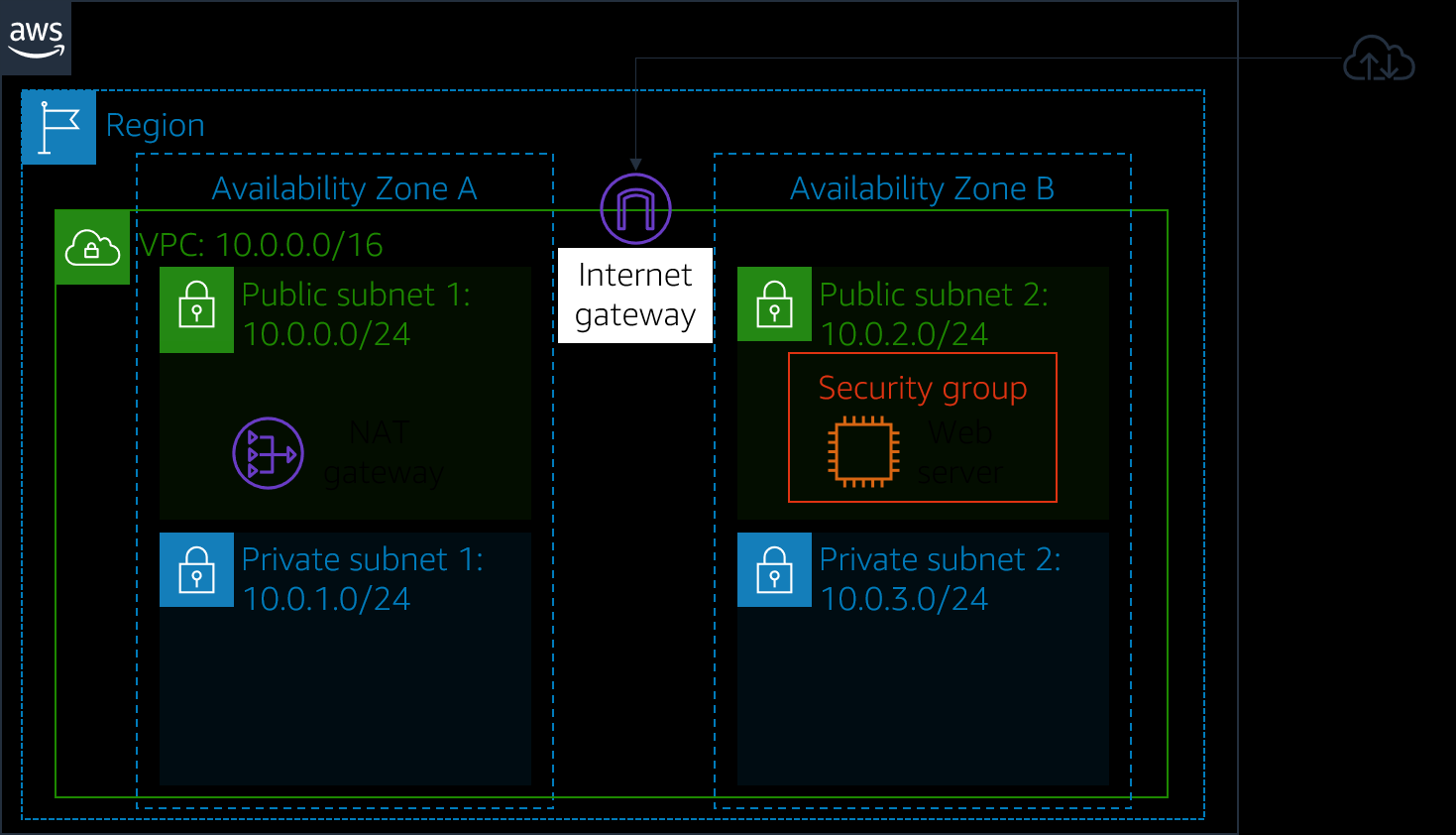
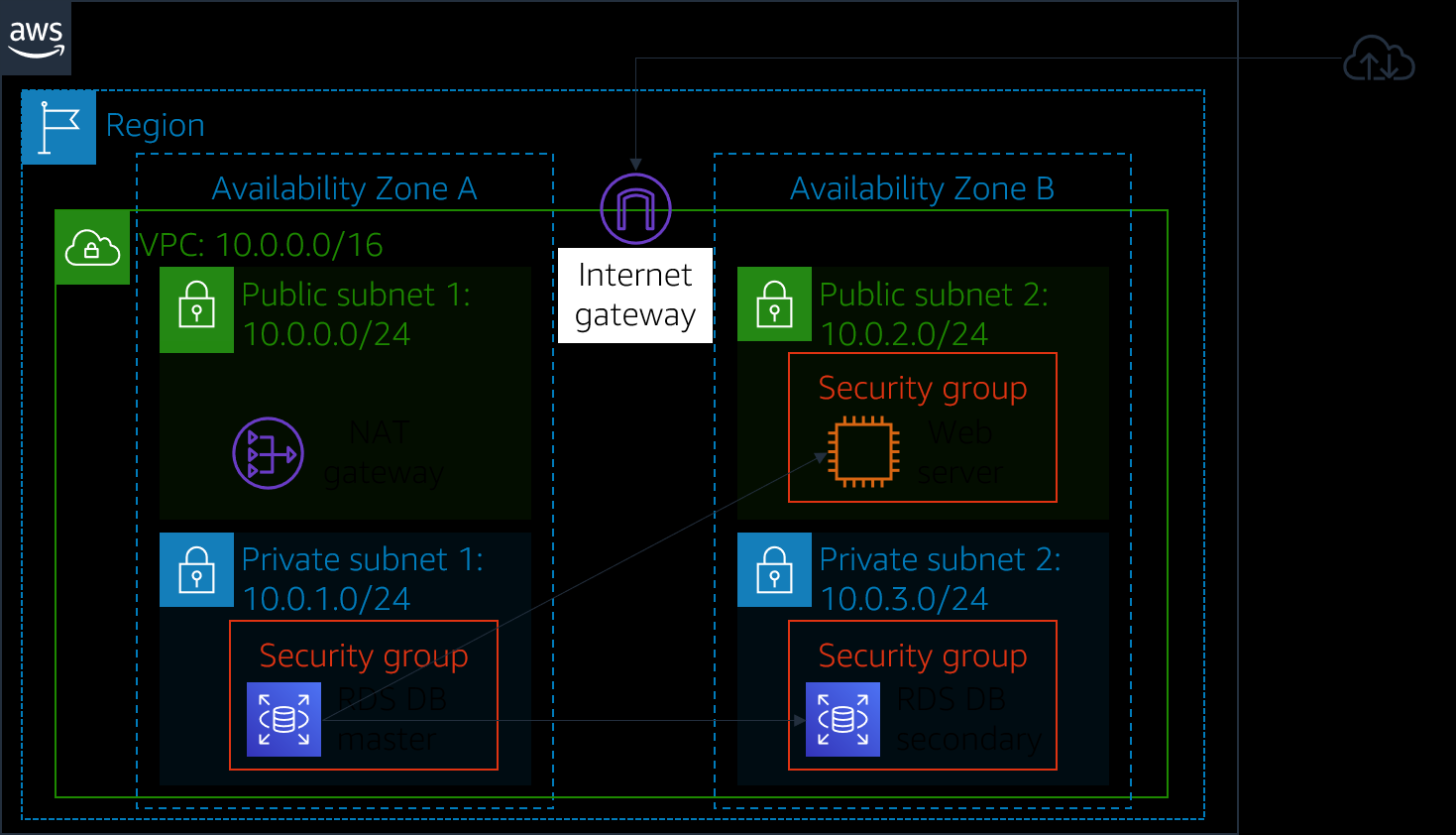


1. Follow previous steps to mount the EBS and view the restored file intact.



**Lab 5:**

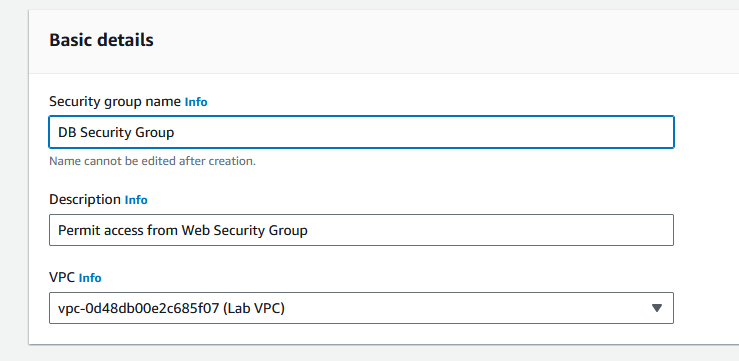
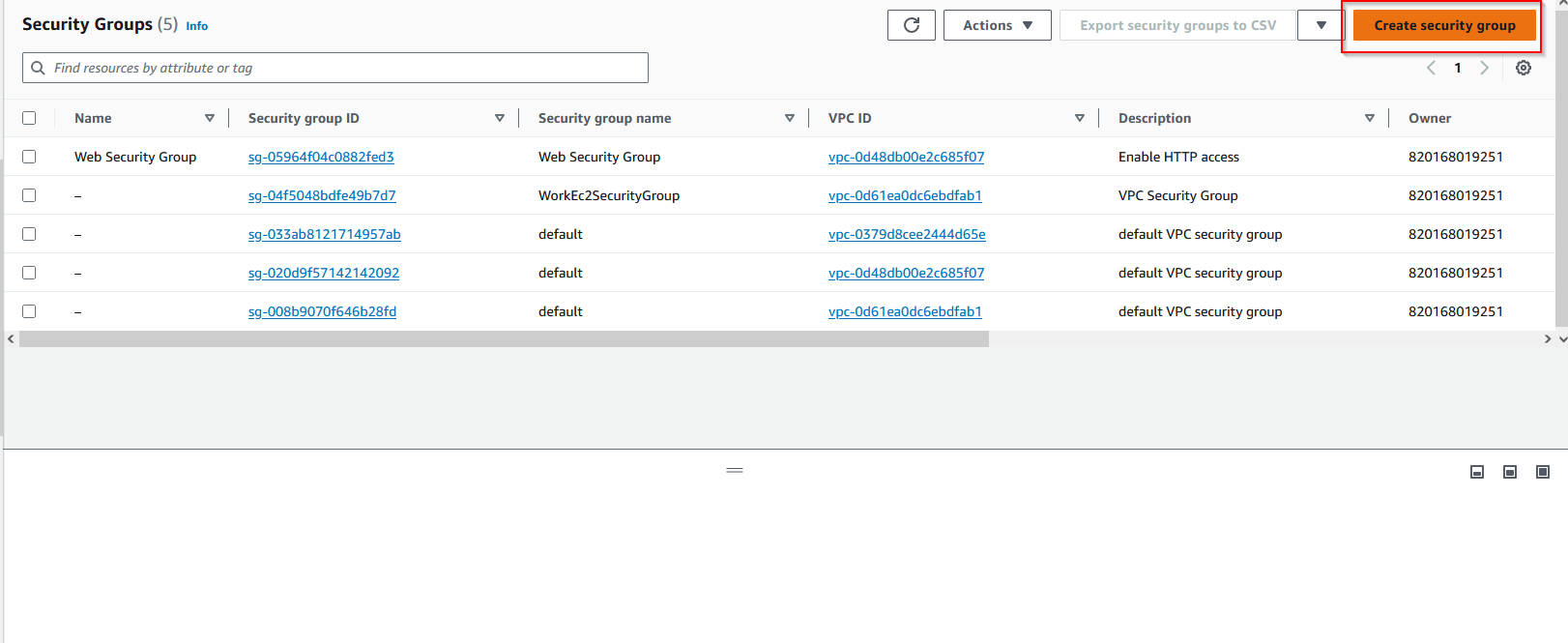
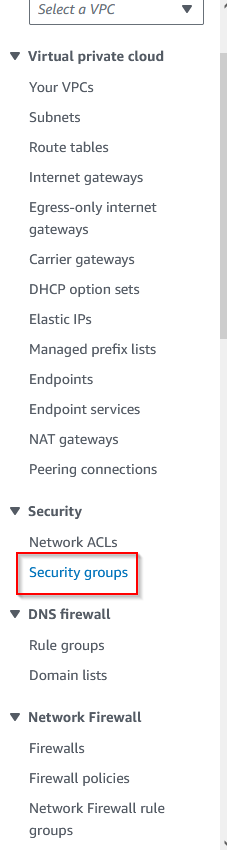
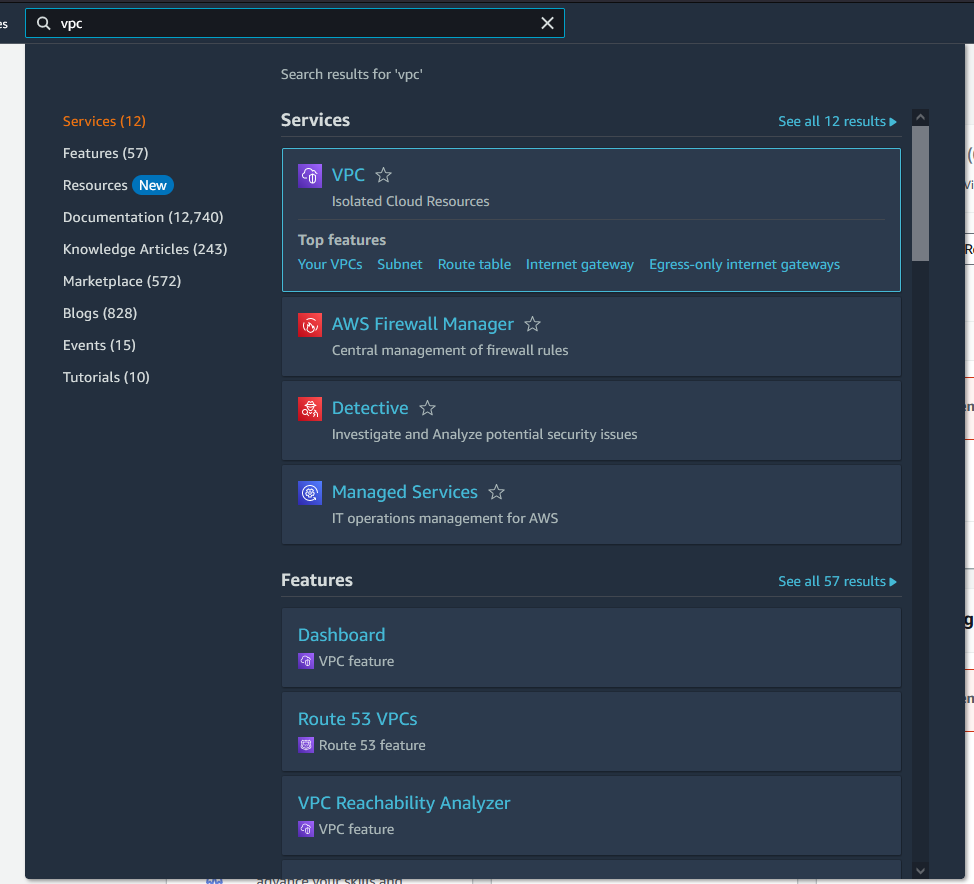
**Network Diagram:**

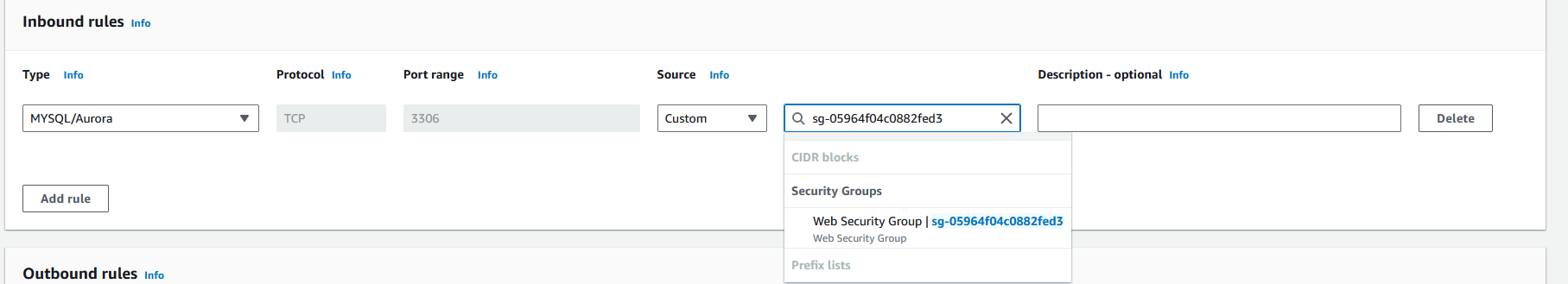
**Lab Instructions:**

This lab provides hands on learning for Amazon’s RDS service, which is an easily scalable relational cloud database. Within it, we learn how to launch an RDS database, as well as configure the database to permit connections from a web server. Finally, we also connect the web application and interact with the database. These skills are easily transferable to configuring future databases, as RDS allows you to choose from numerous different database types such as Microsoft SQL, PostgreSQL, MySQL, and other.

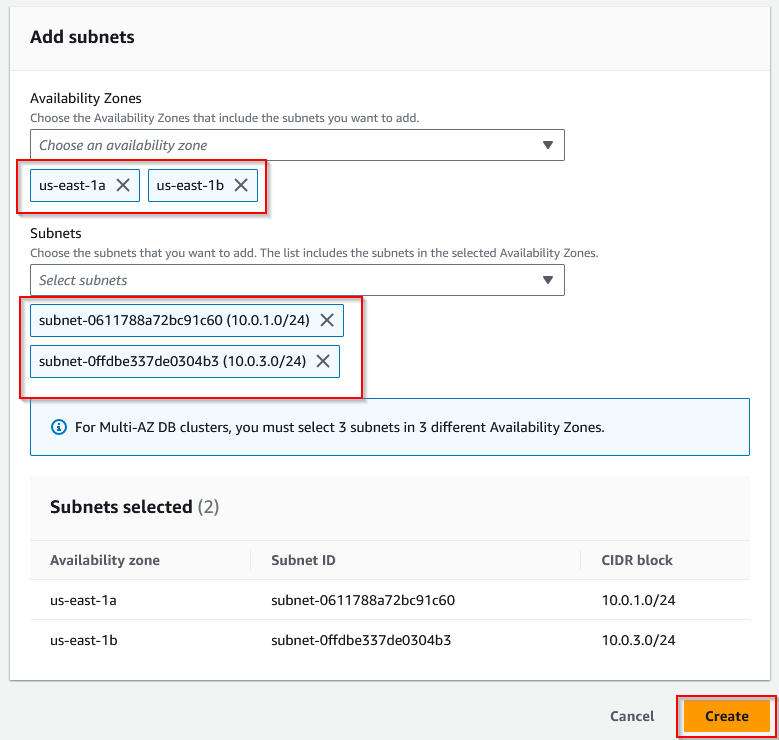
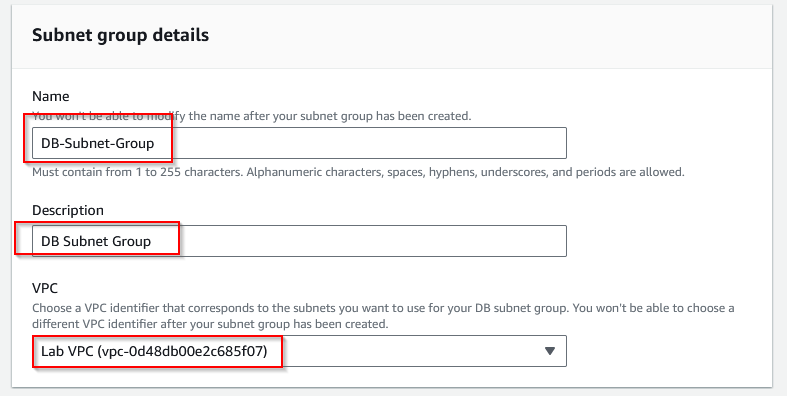
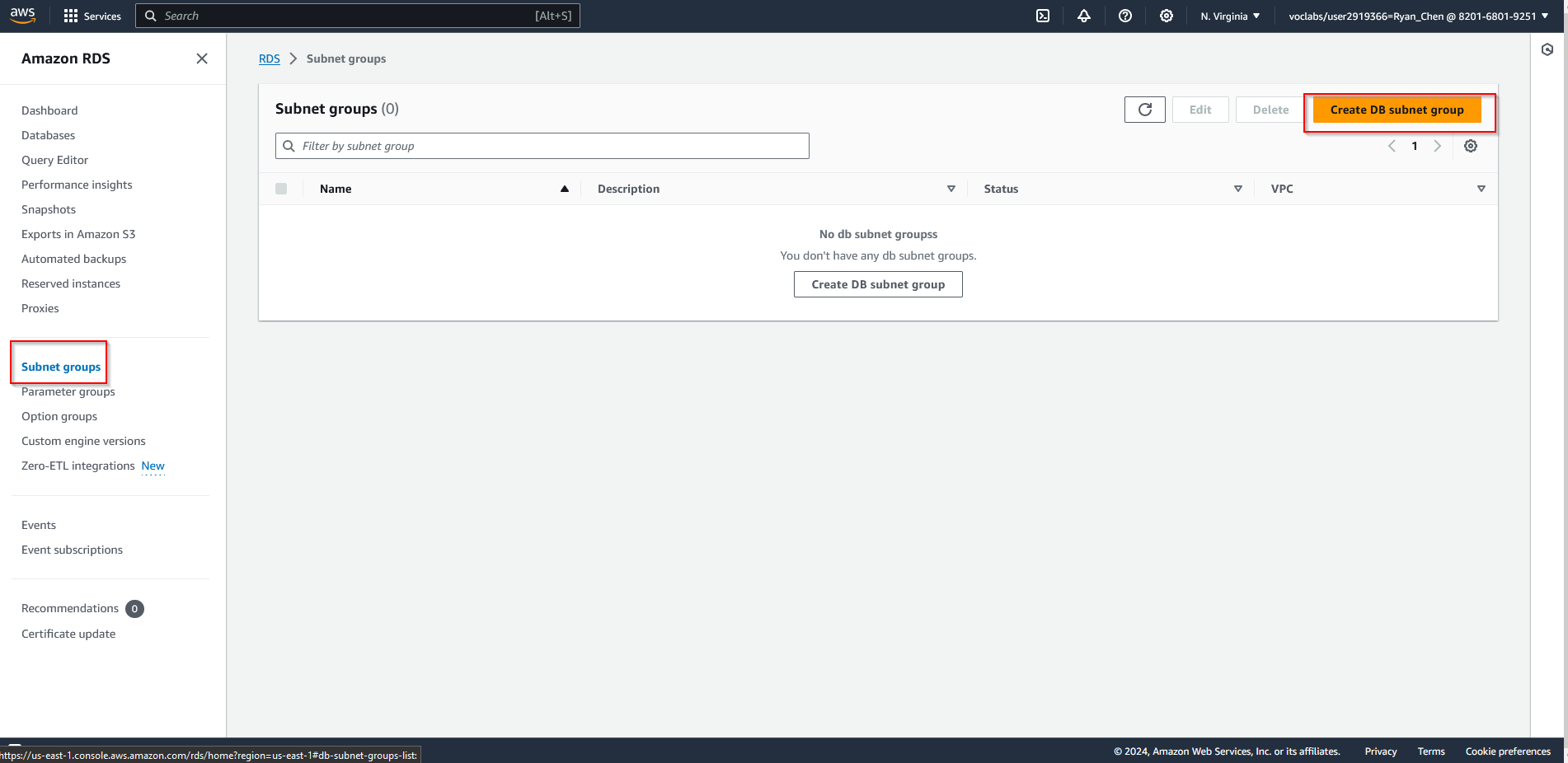
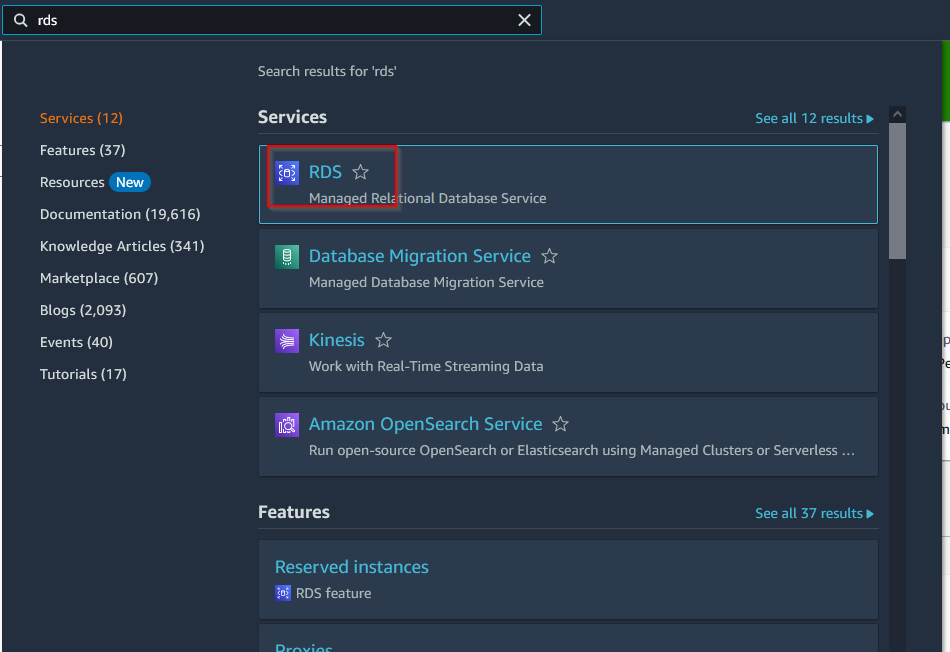
1. Navigate to VPC, then select Security Groups and create a new security group following the given specifications.



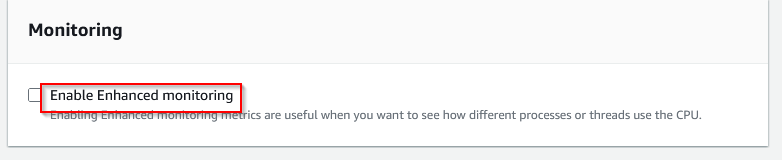
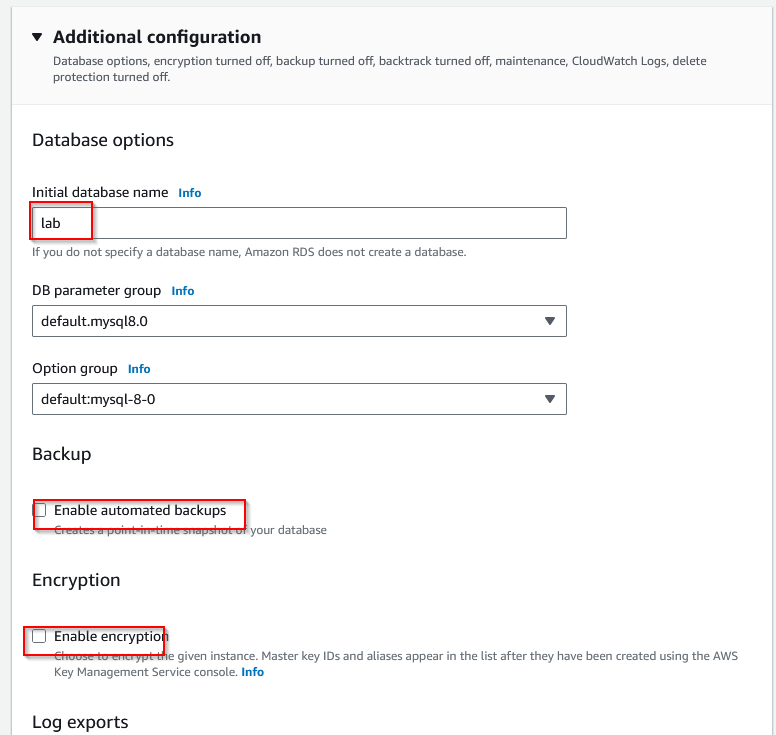
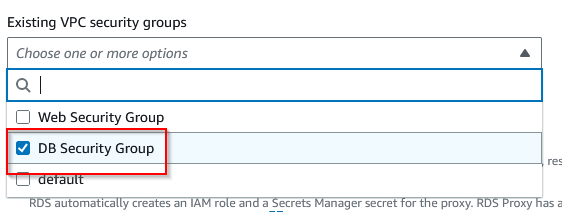
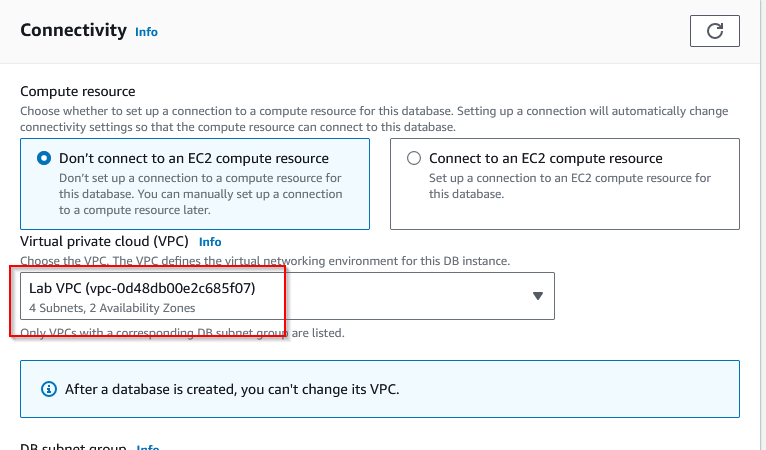
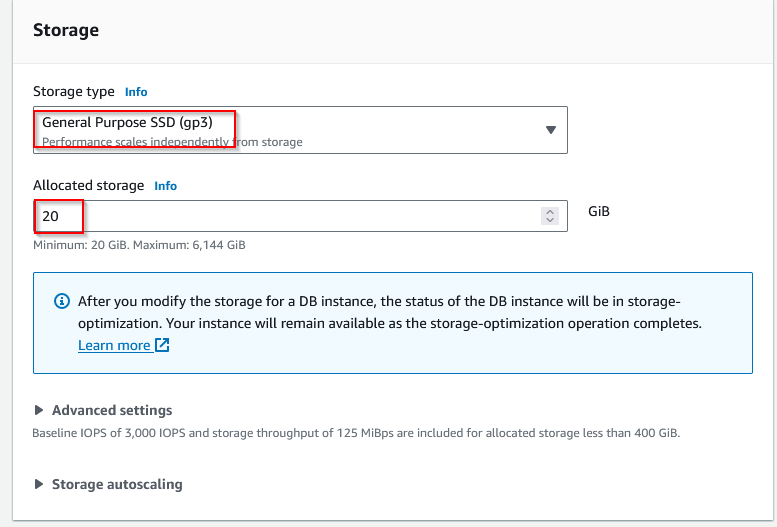
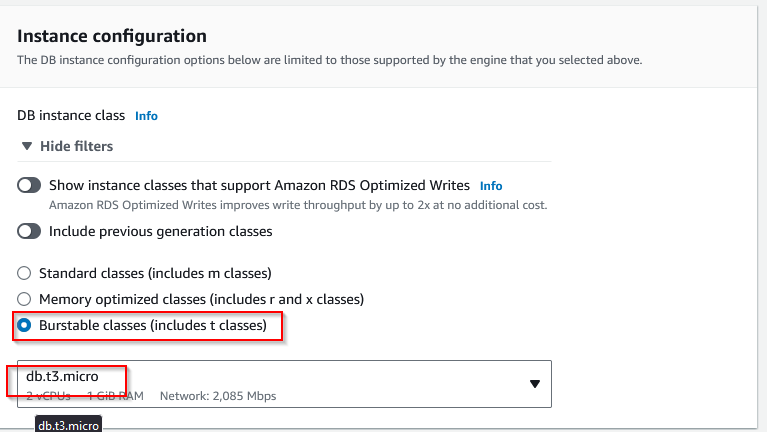
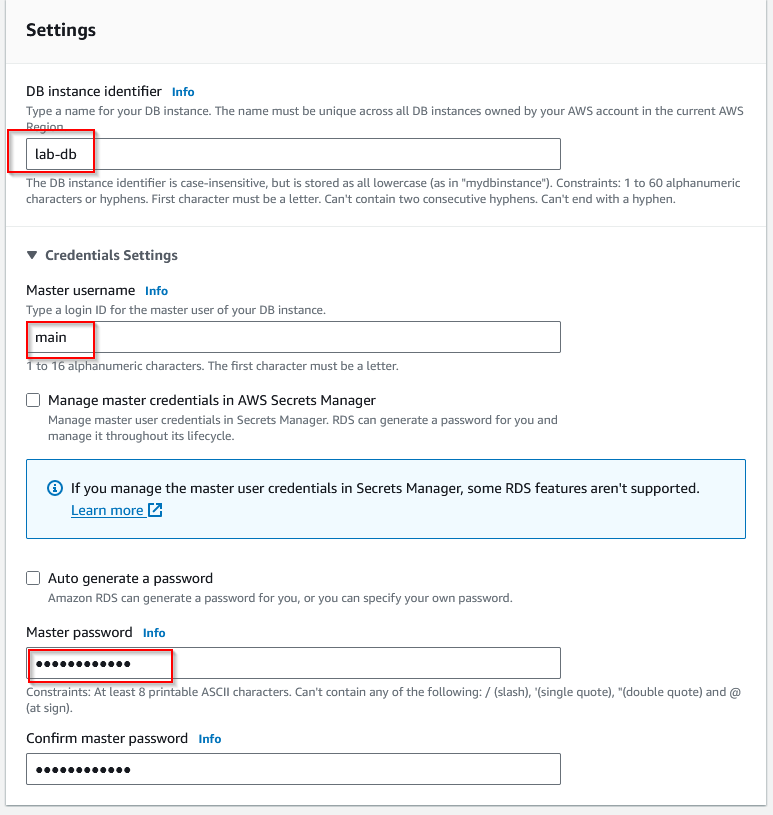
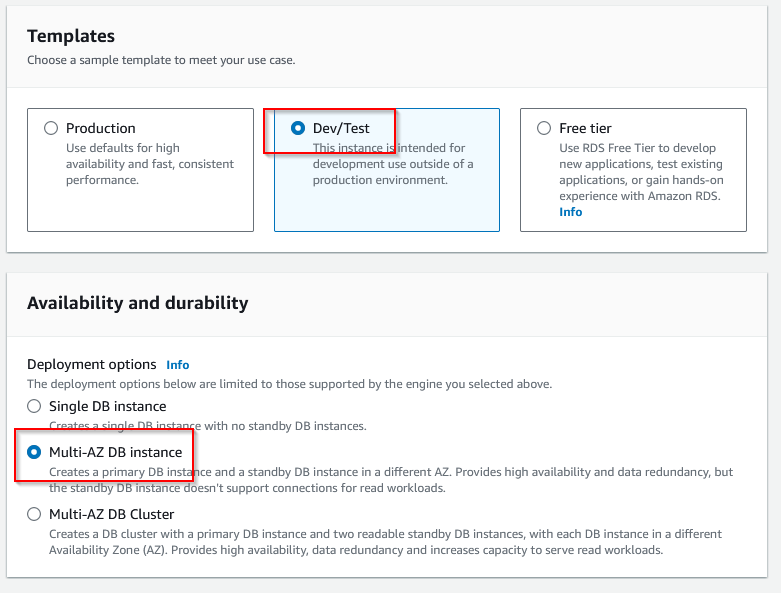
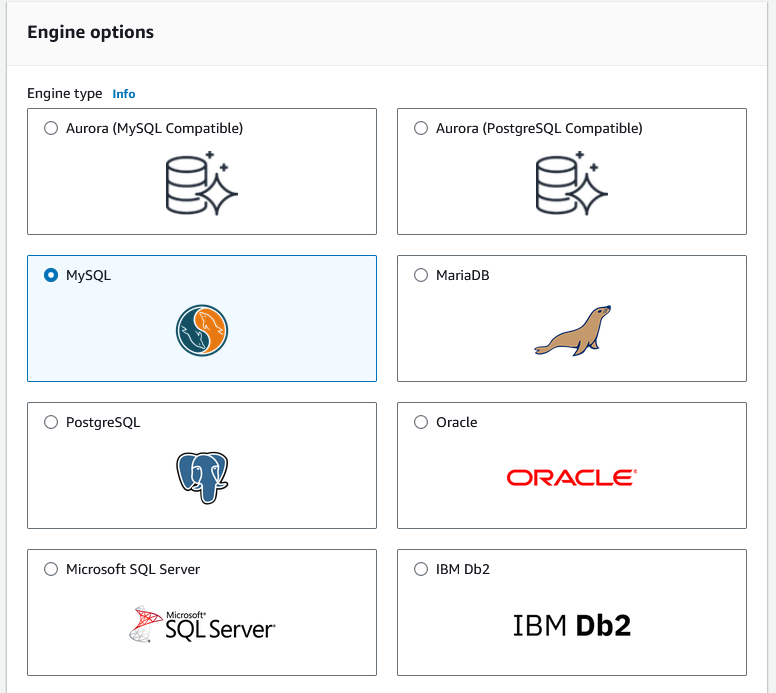
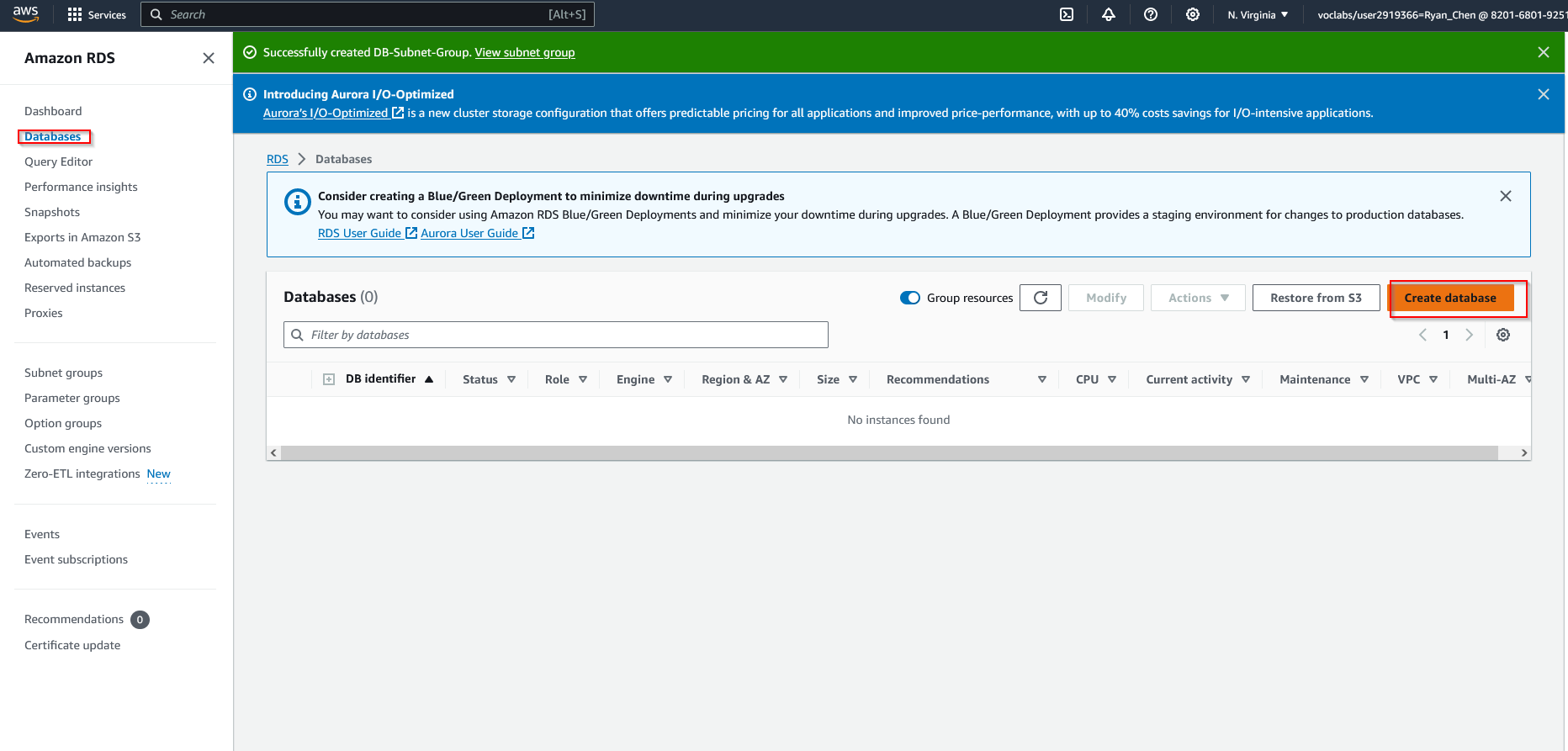
1. Add an inbound rule that utilizes MySQL, and select “Web Security Group”.



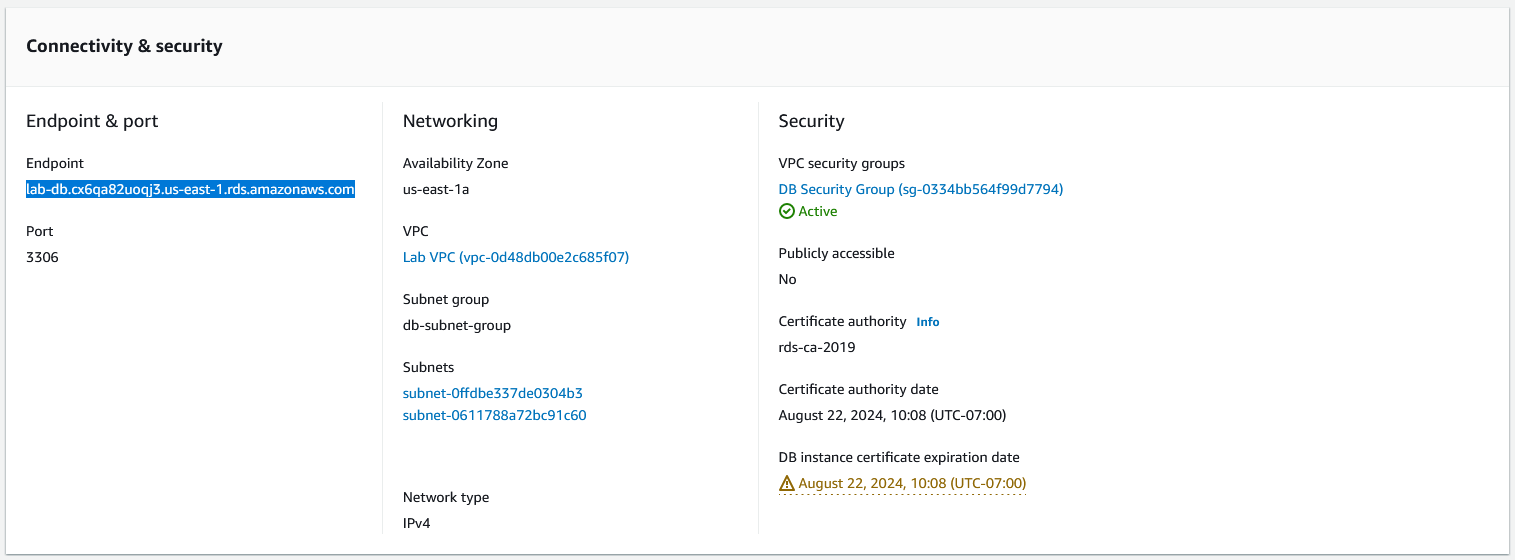
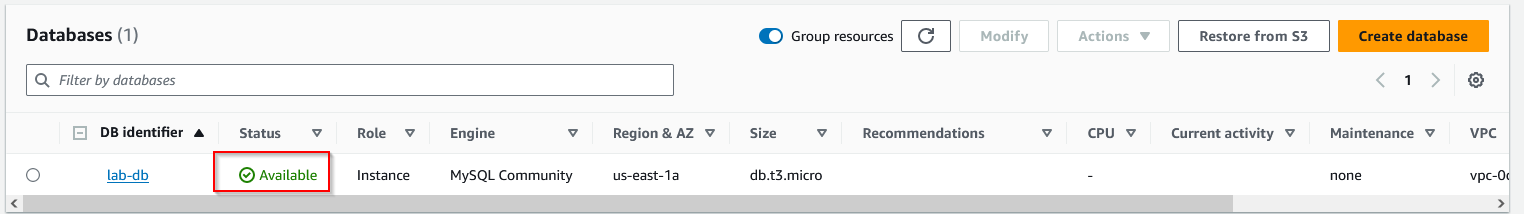
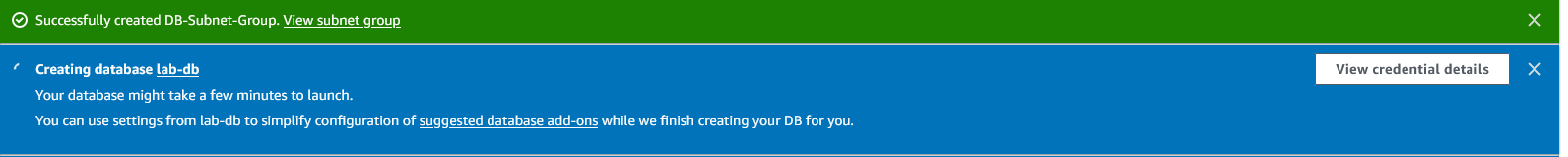
1. Navigate to RDS in the Services tab and create new subnet group with given configurations.



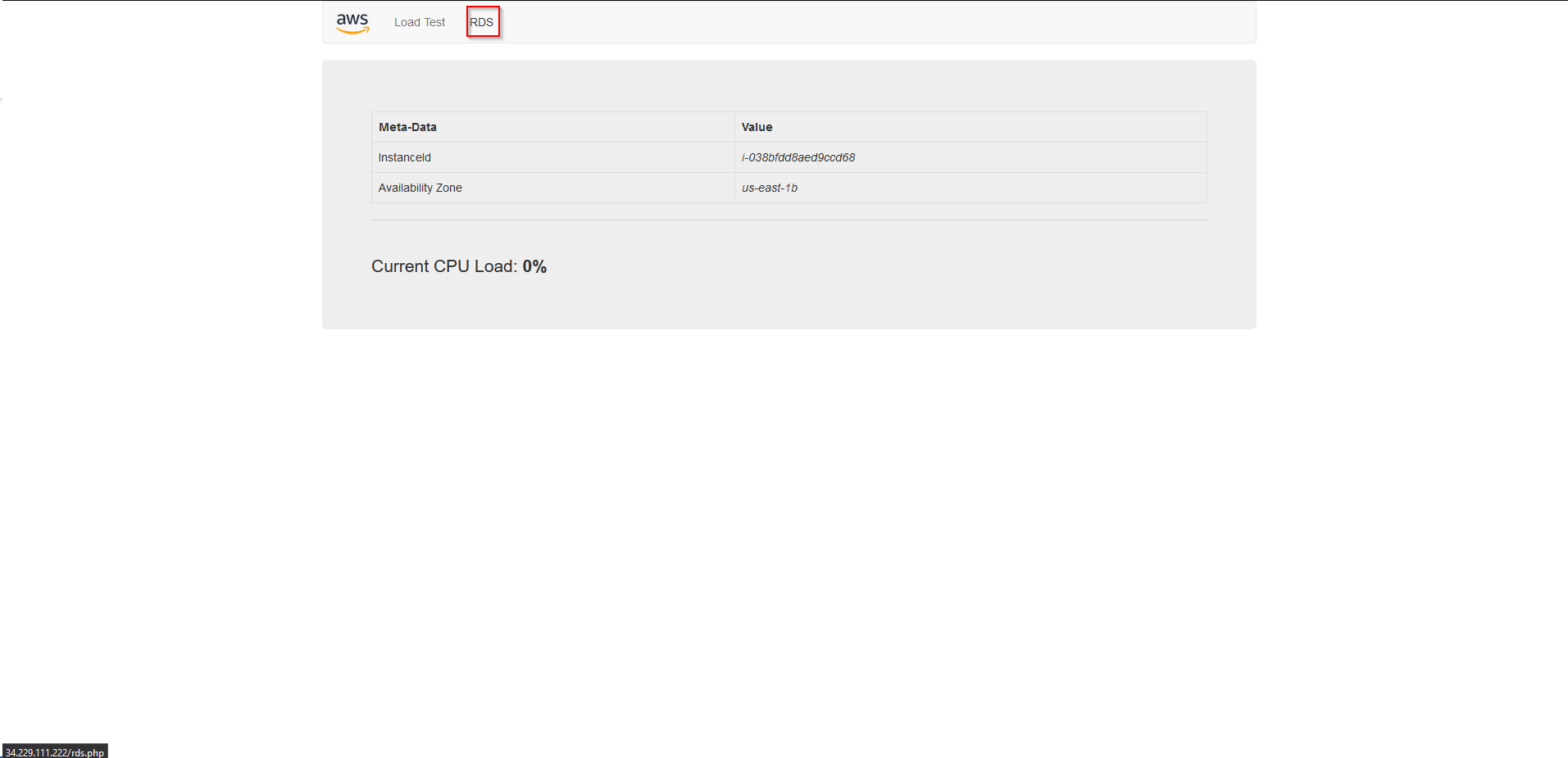
1. Now, create anew Amazon RDS DB instance. To do so, navigate to the databases option on the left hand side, then create a new database based on given configurations.



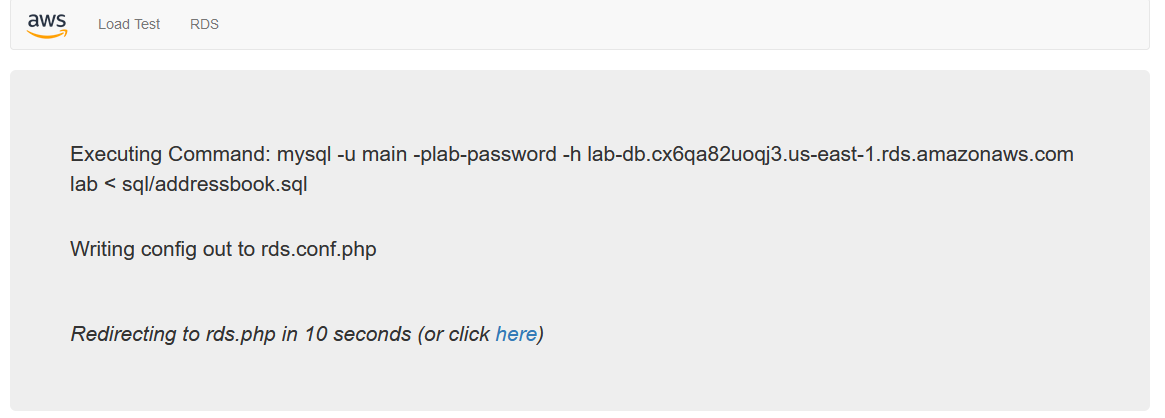
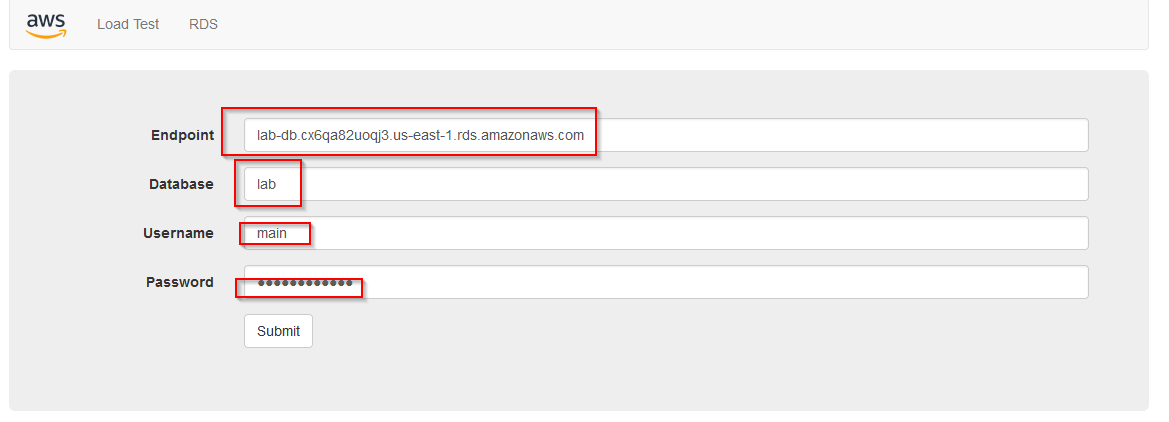
1. Wait for the database to deploy and copy the endpoint link when it finishes deploying.



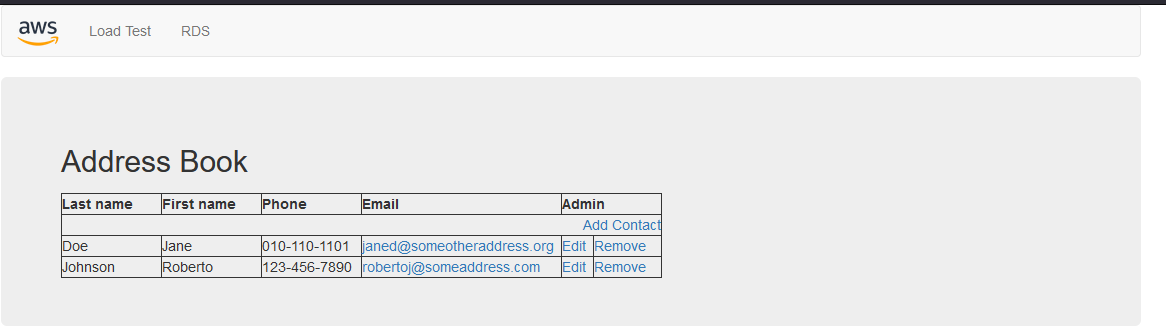
1. Copy and paste the link for deploying RDS and select RDS.



1. Fill in information based on previously configured data.

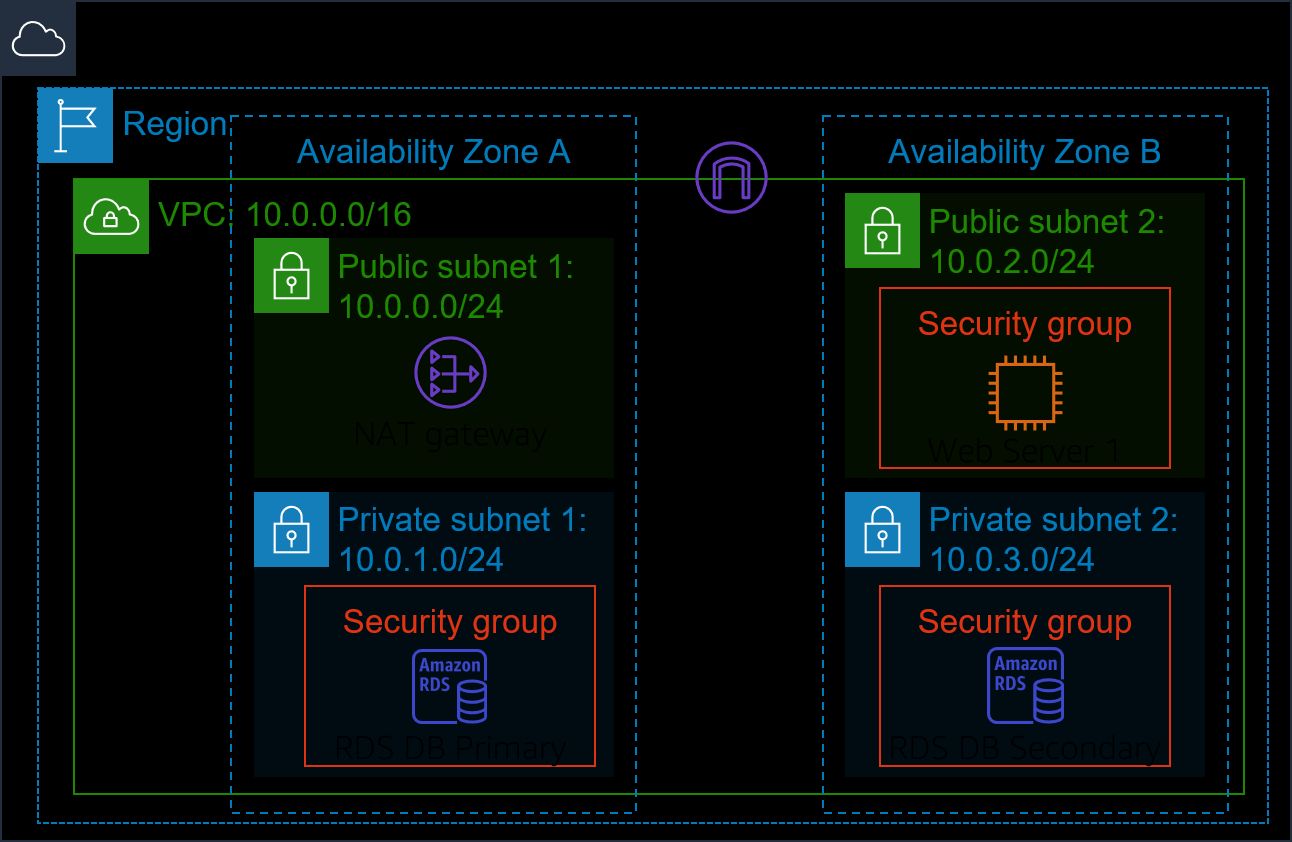
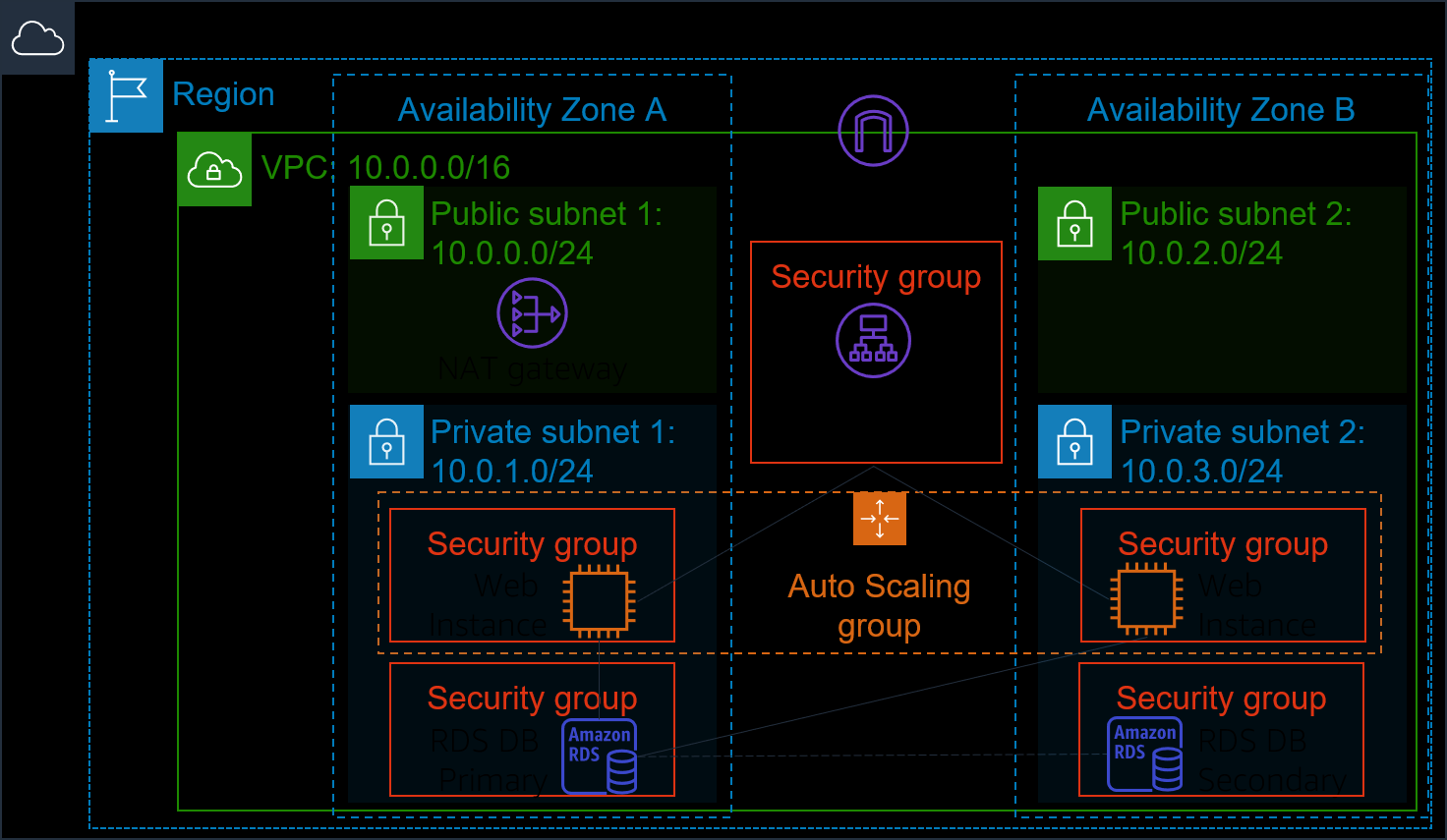


1. The application and database are now running and able to be edited, deleted, or added to.



**Lab 6:**

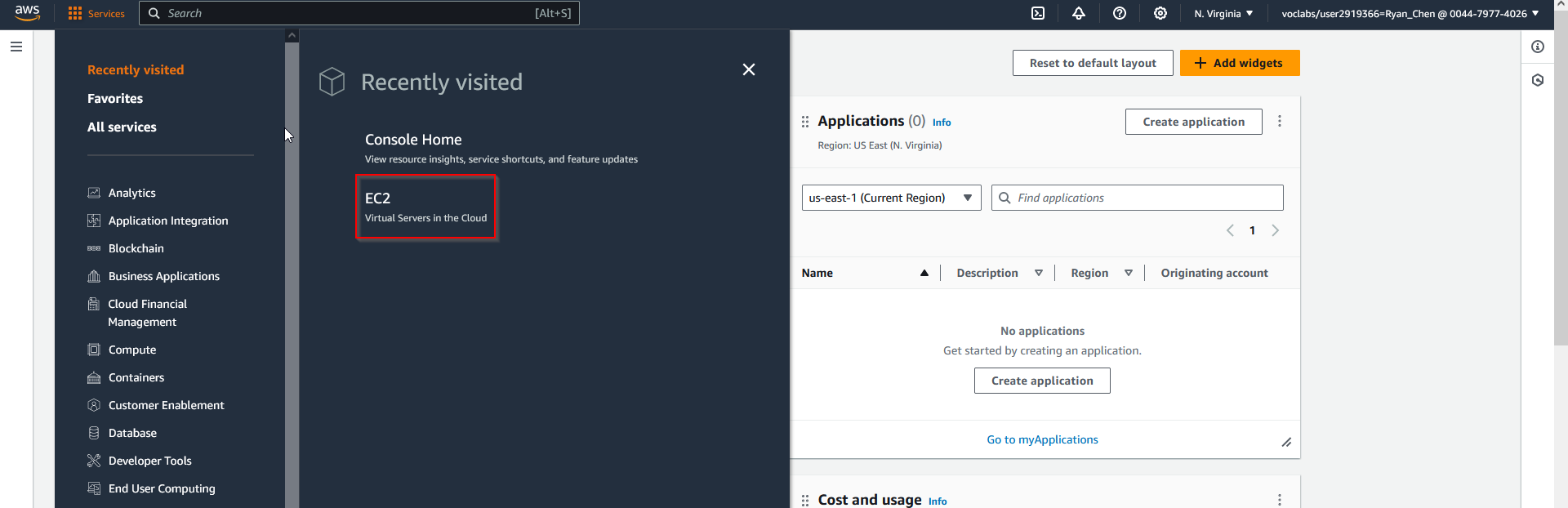
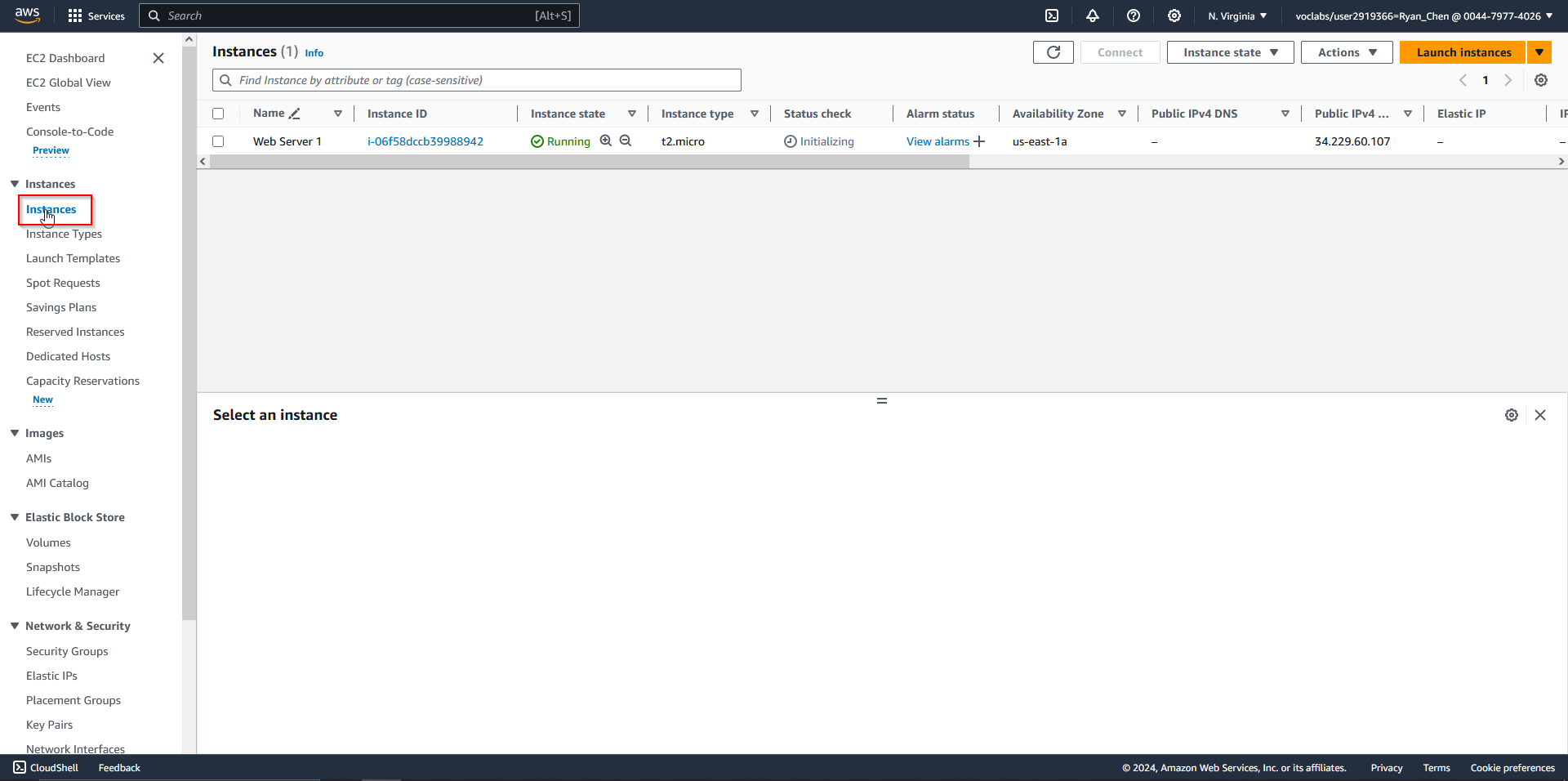
**Network Diagram:**

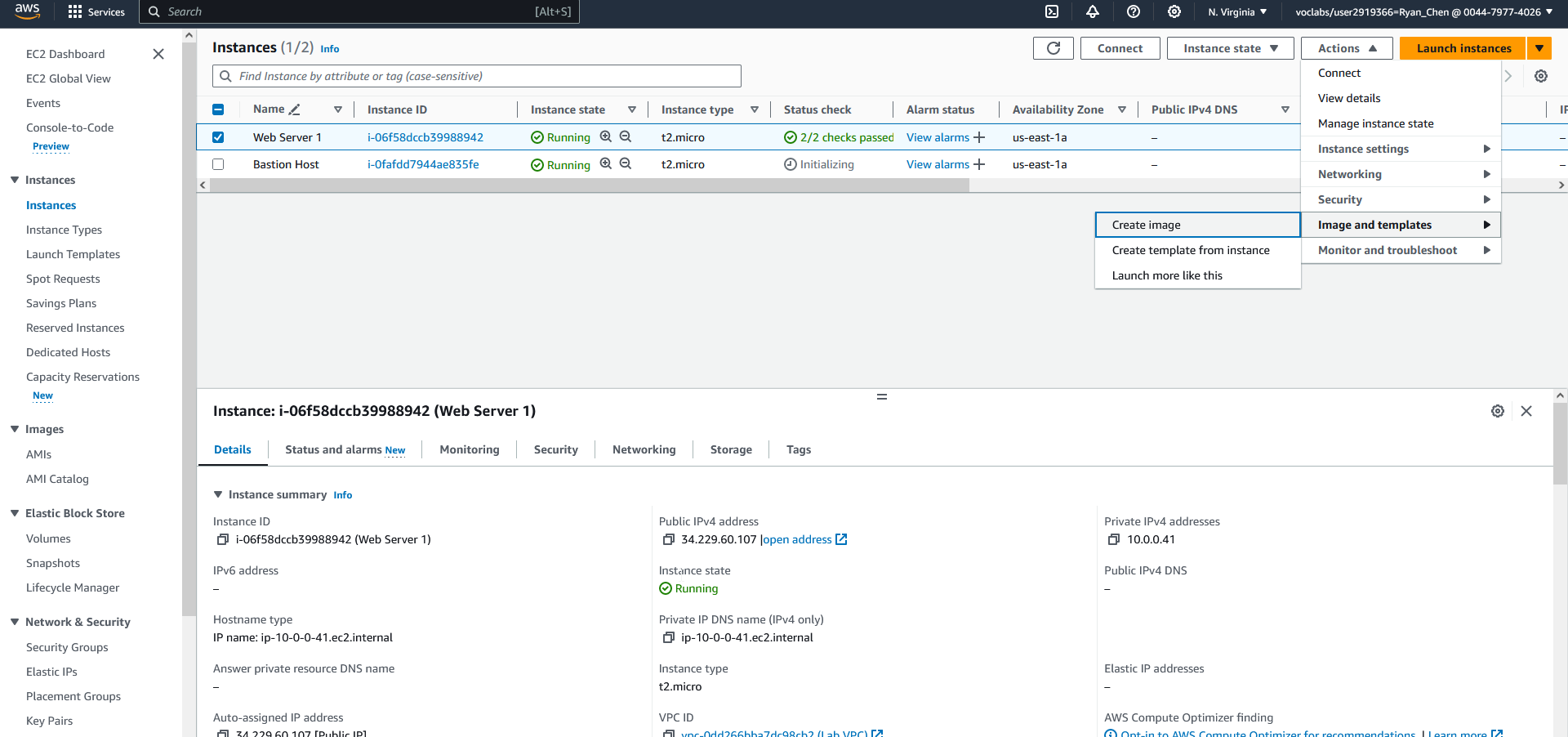
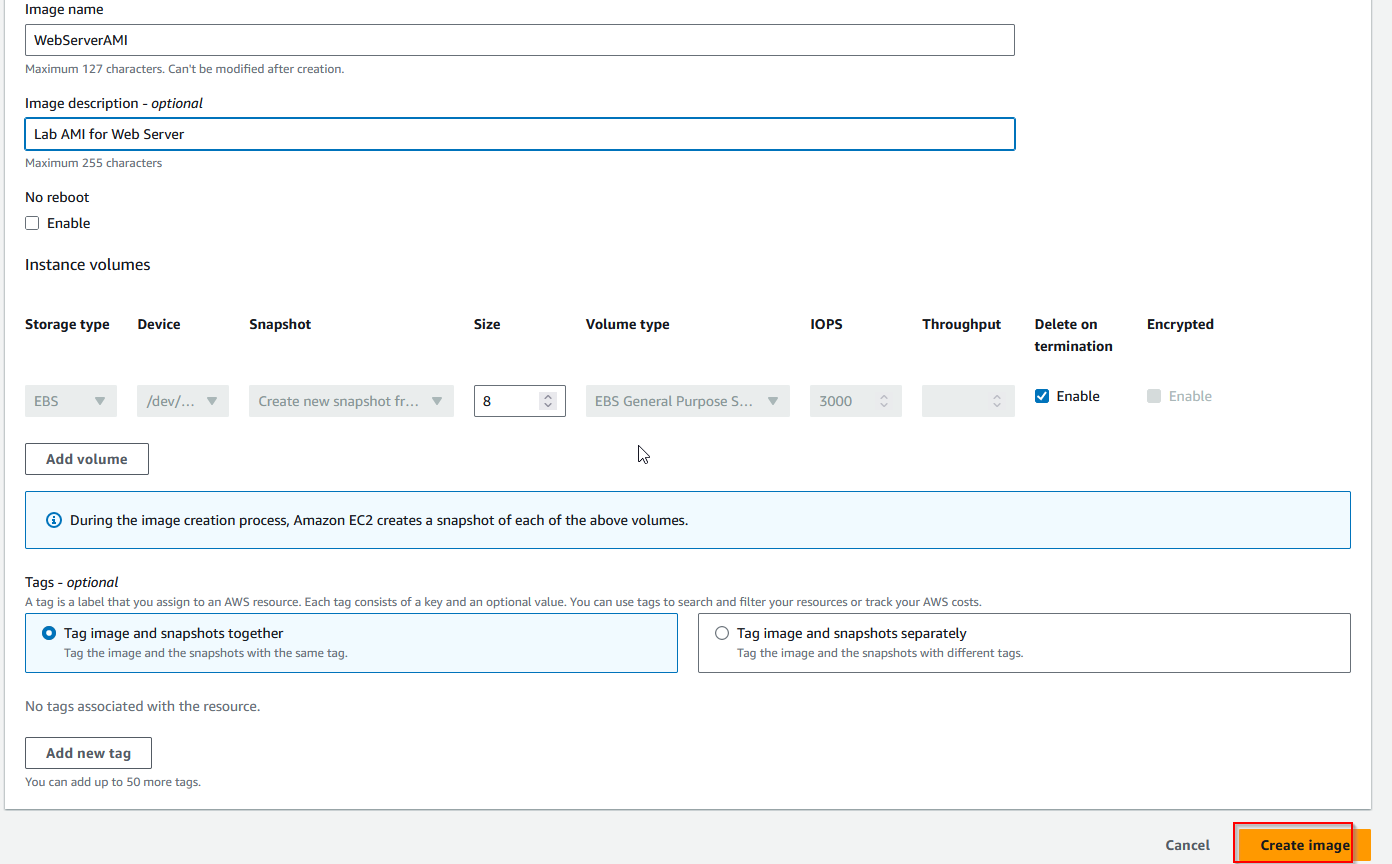
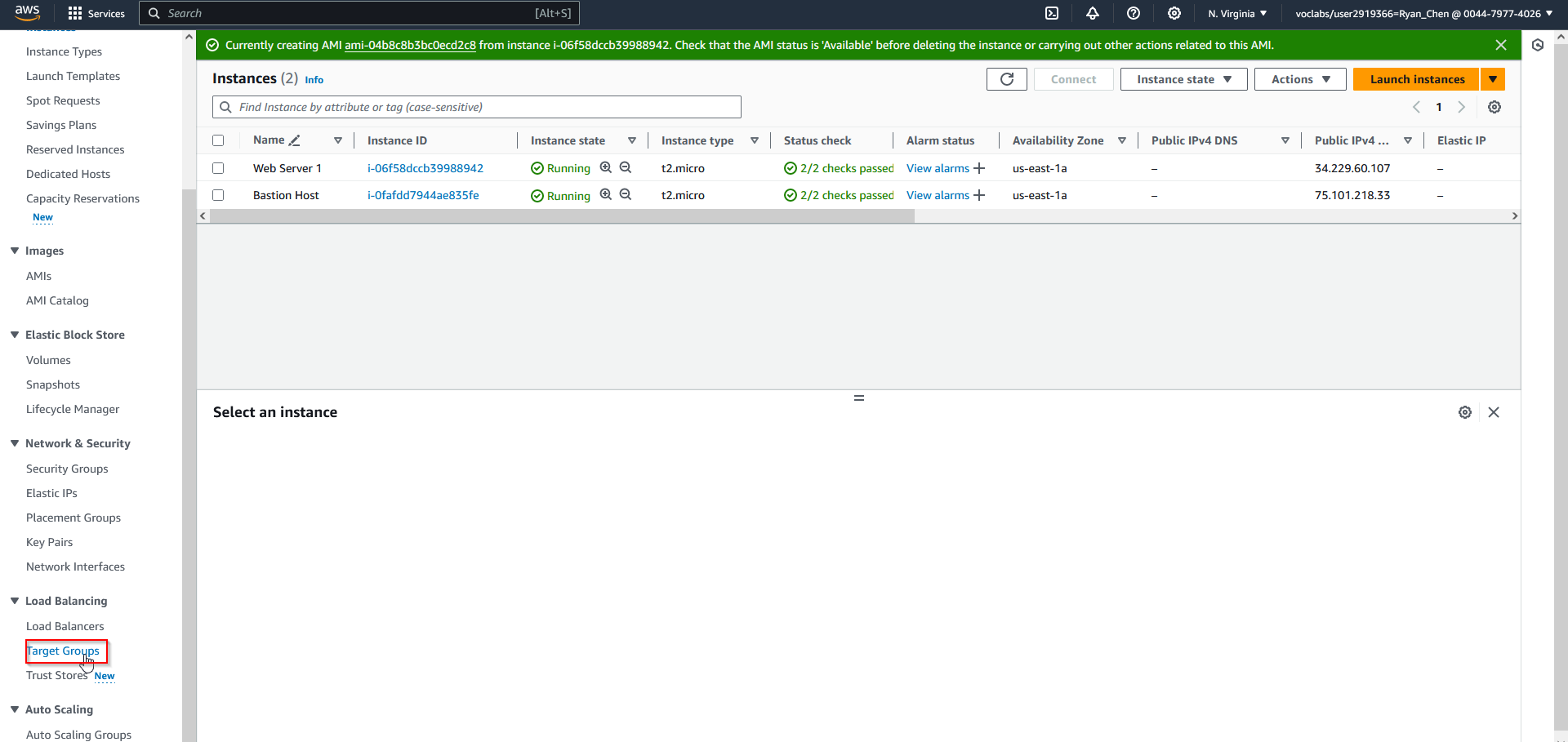
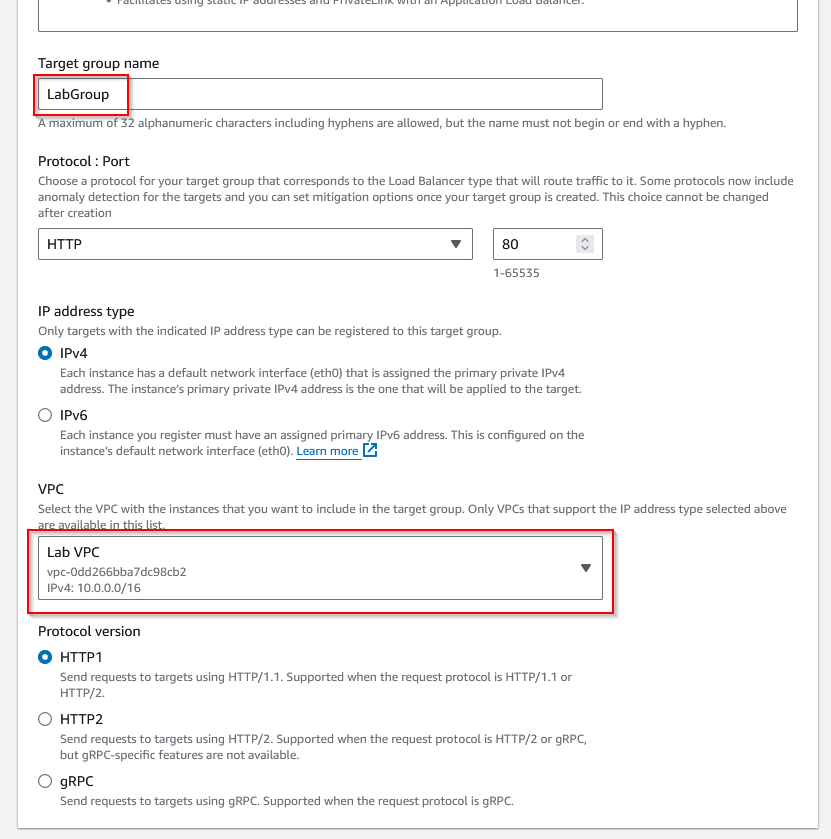
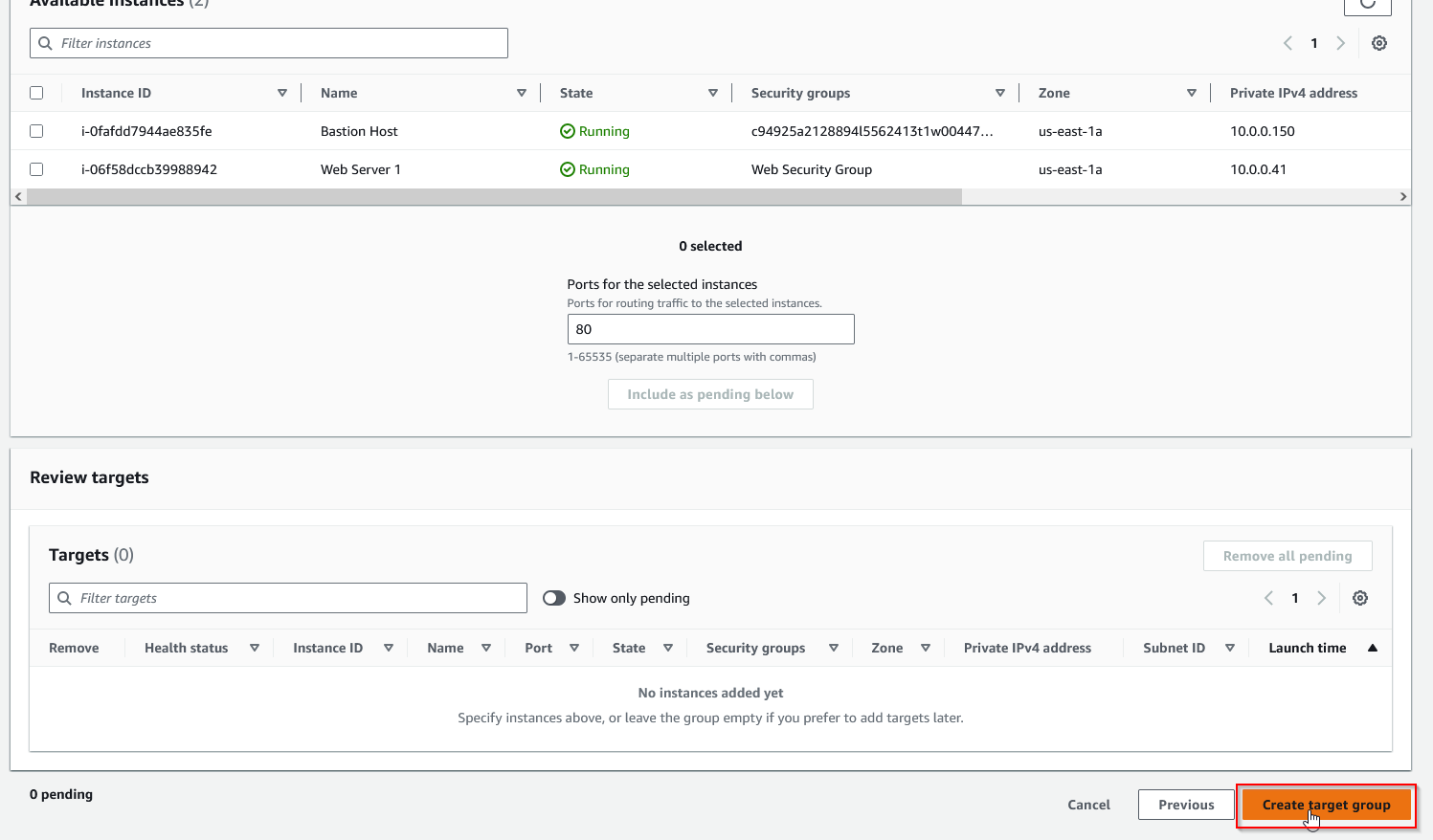
**Lab Instructions:**

This lab introduces Amazon ELB, a load balancing service that provides automatic scaling of infrastructure and greater fault-tolerance. Furthermore, we also learn how to leverage Amazon’s auto scaling in order to provide more flexible resource allocation to match with our application demand at any given point.

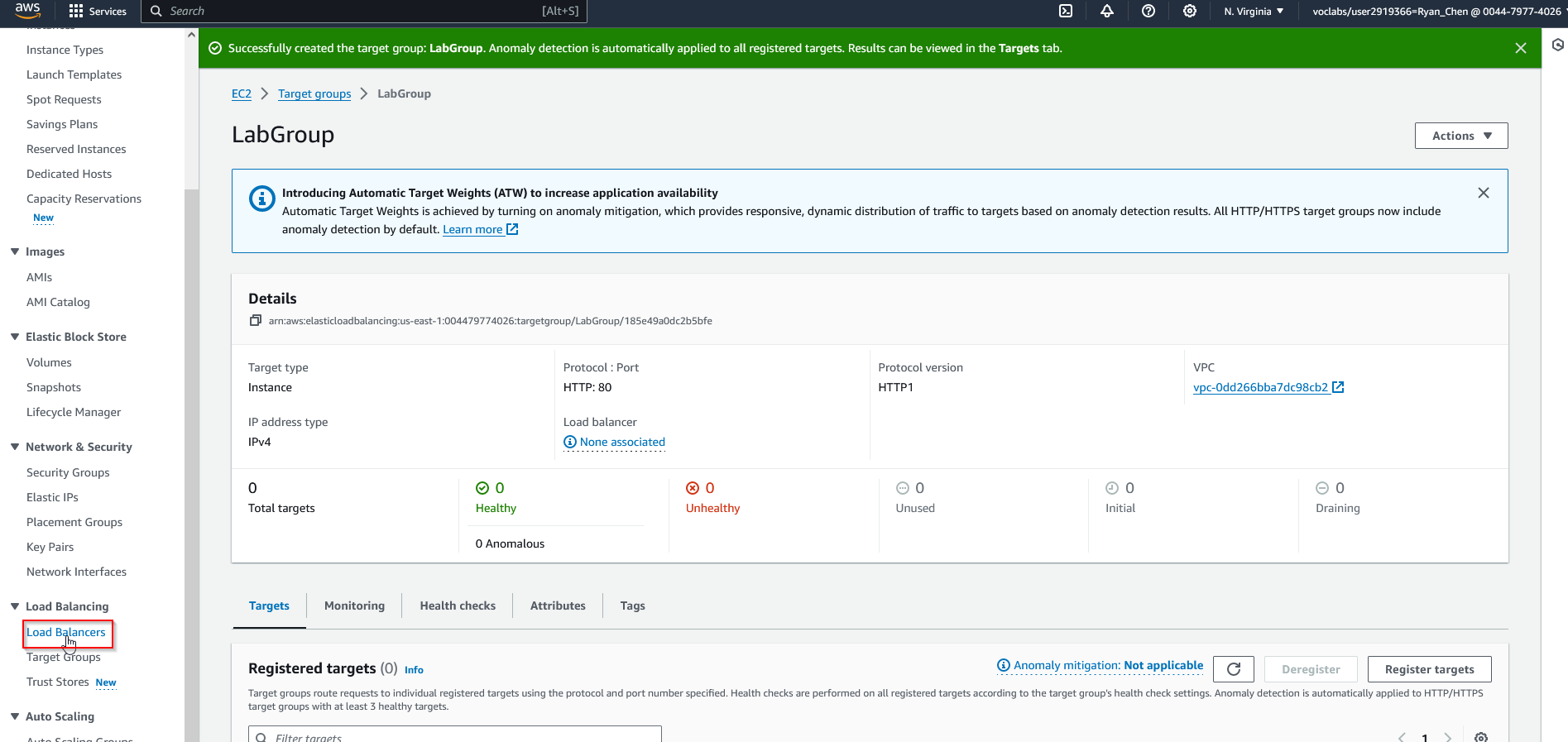
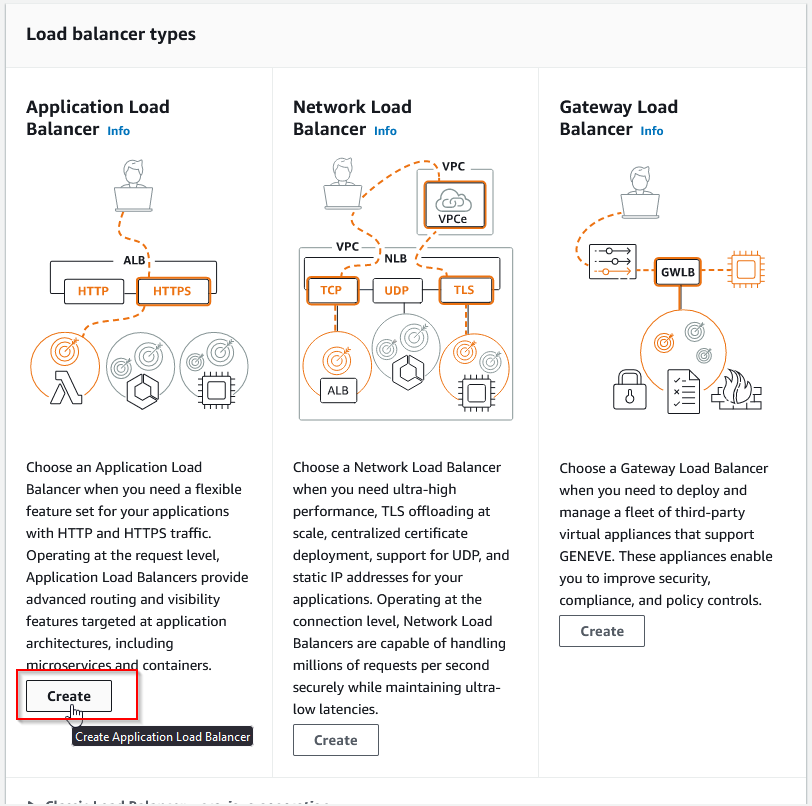
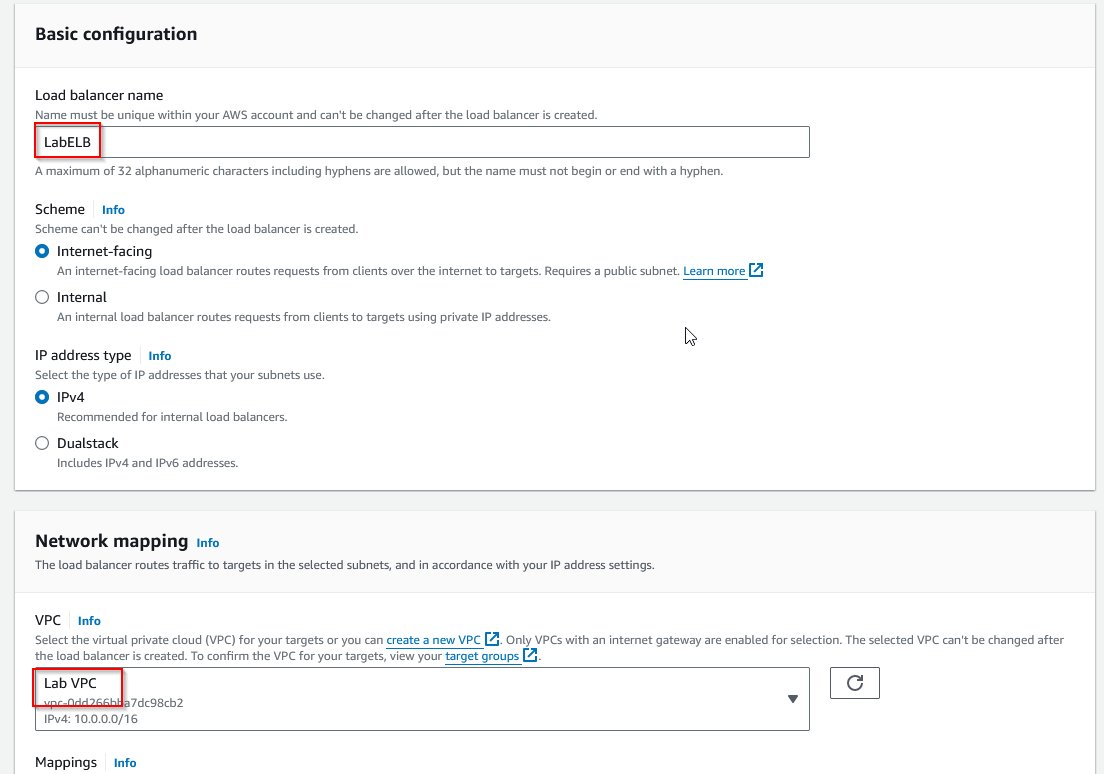
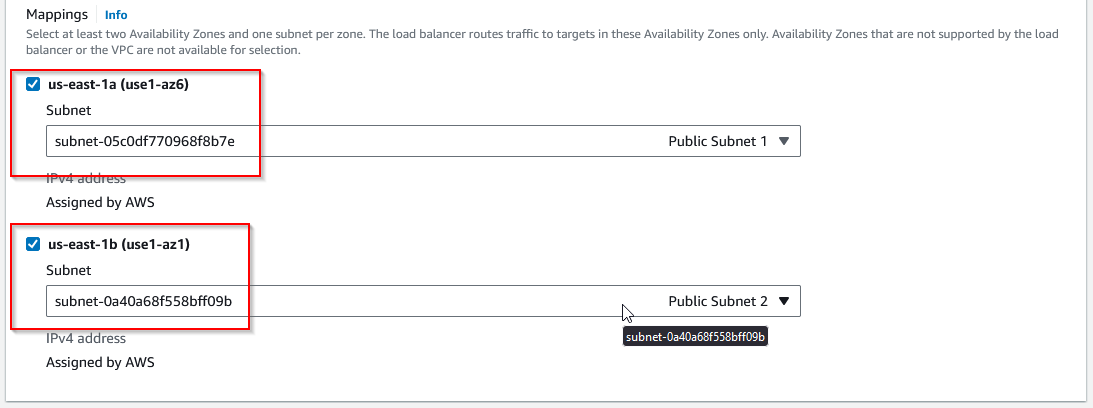
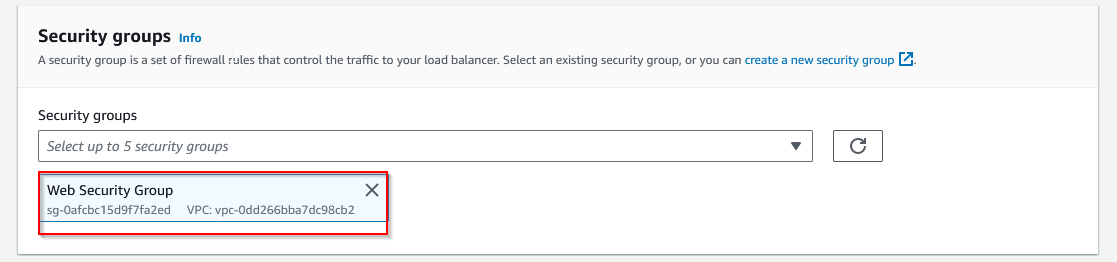
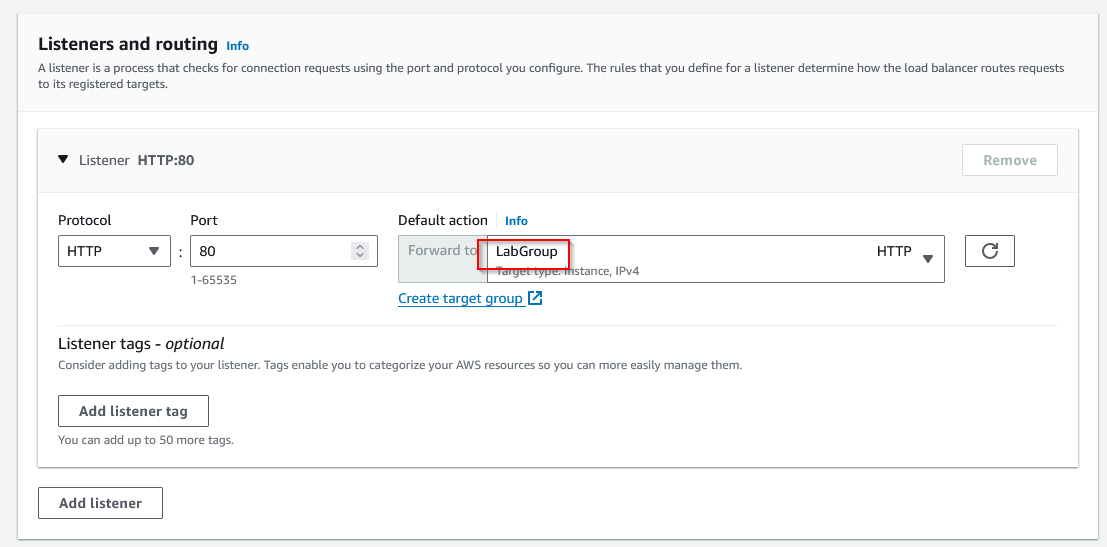
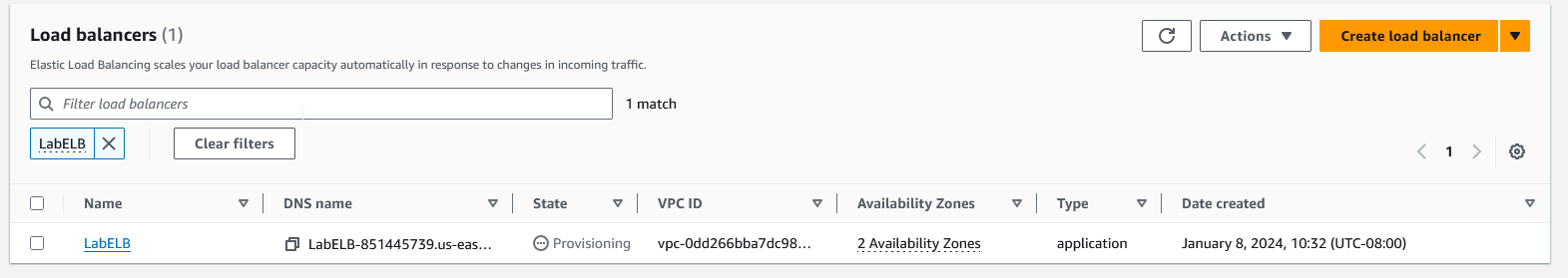
1. Select EC2 in the management console and navigate to the Instances section. Wait until Web Server 1 has passed its checks, then continue.

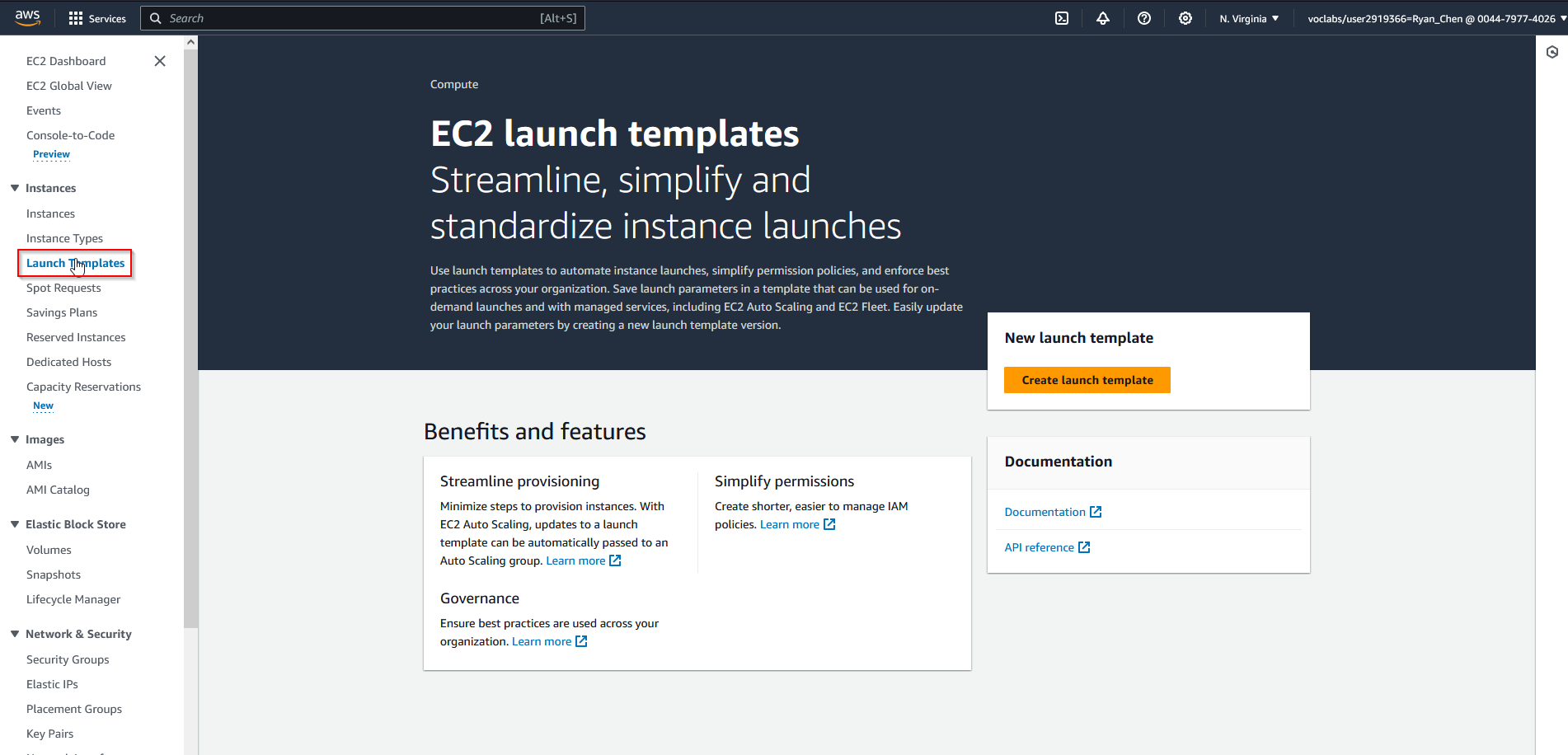
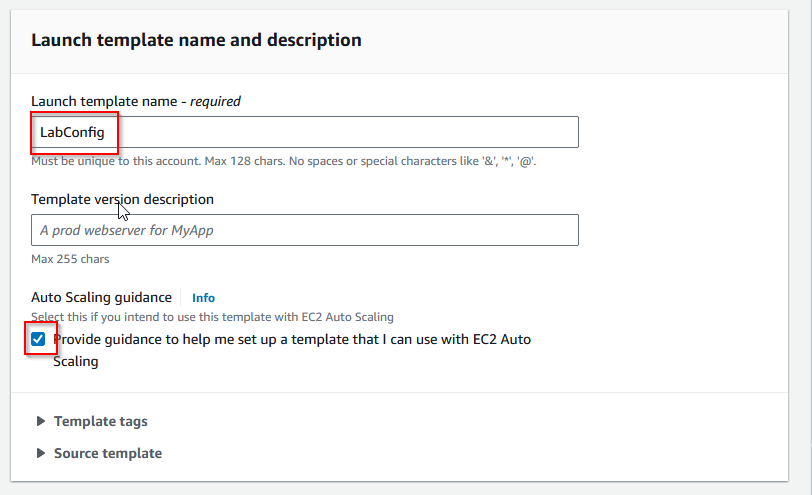
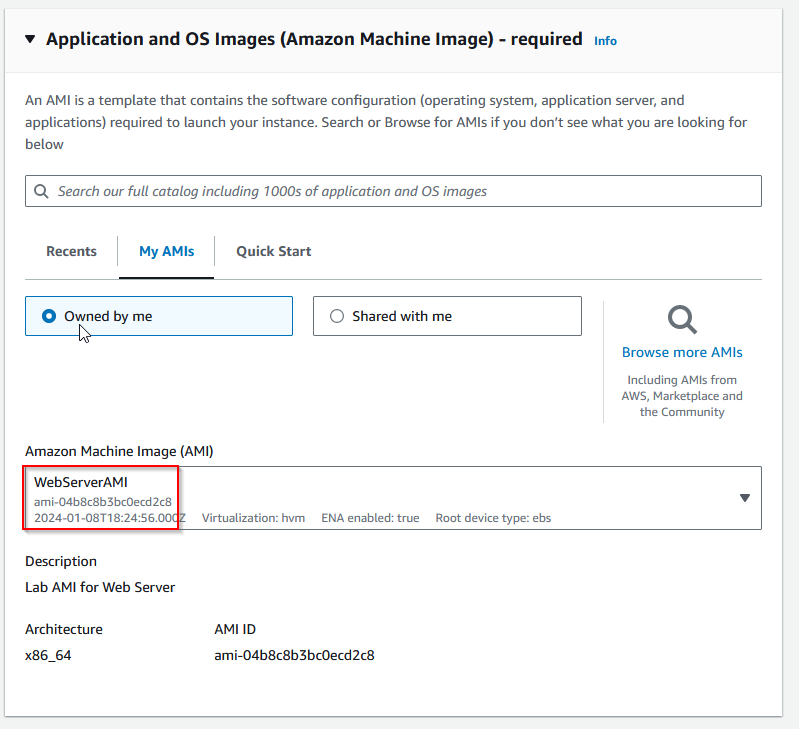
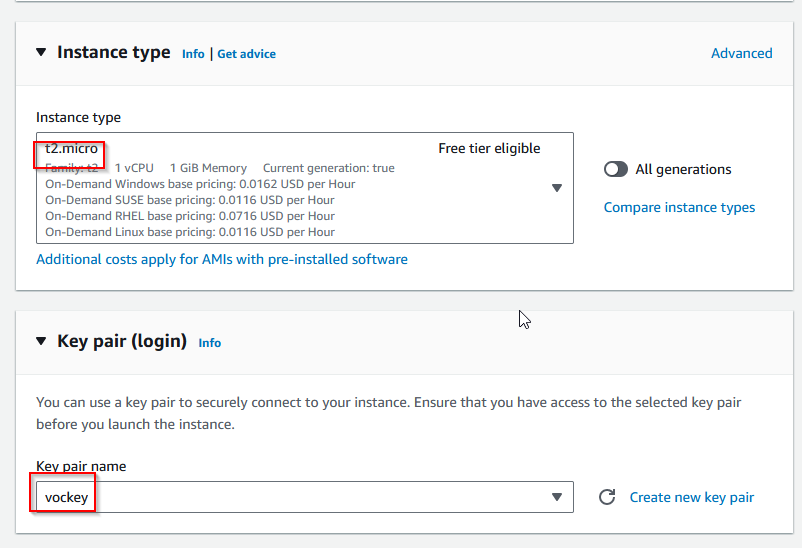
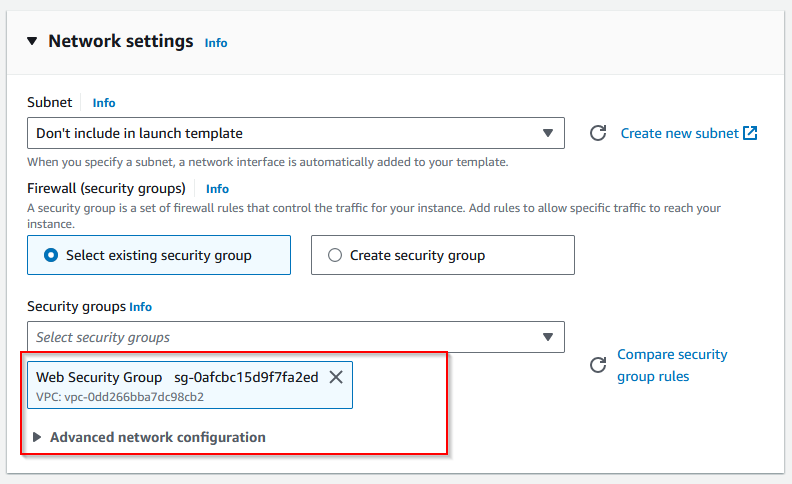
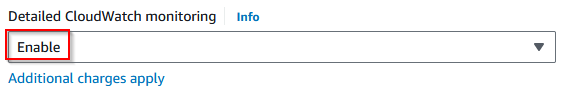
 

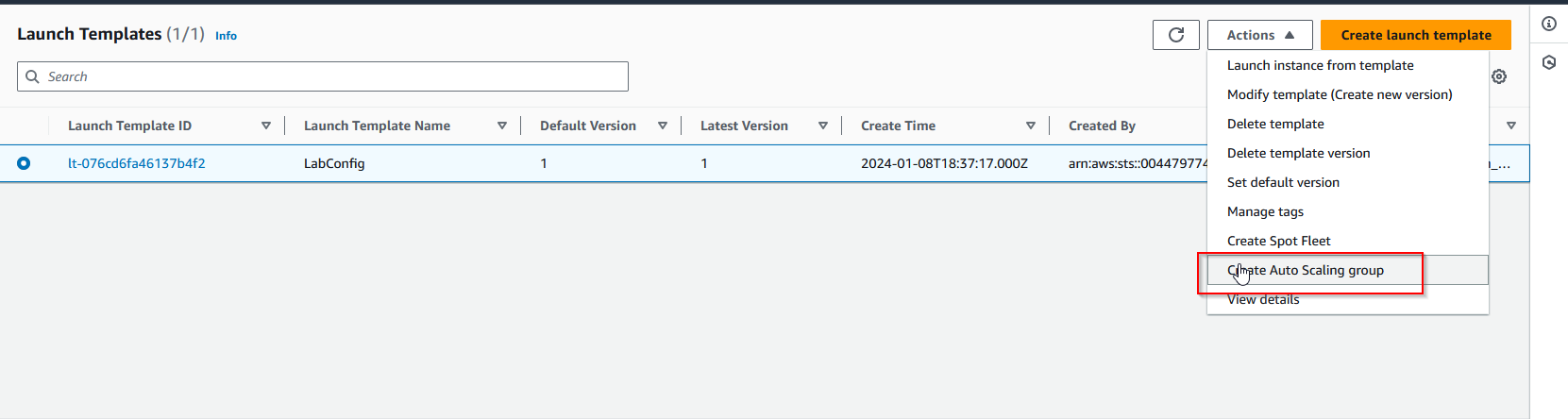
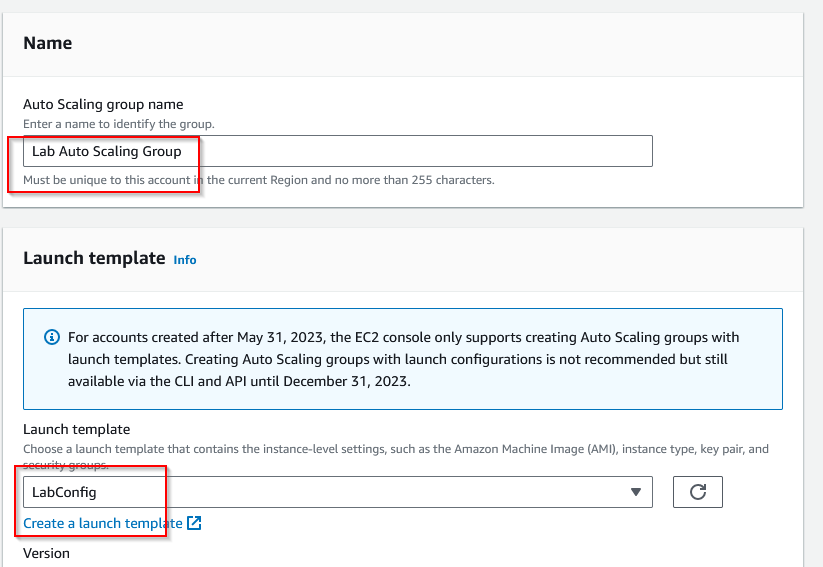
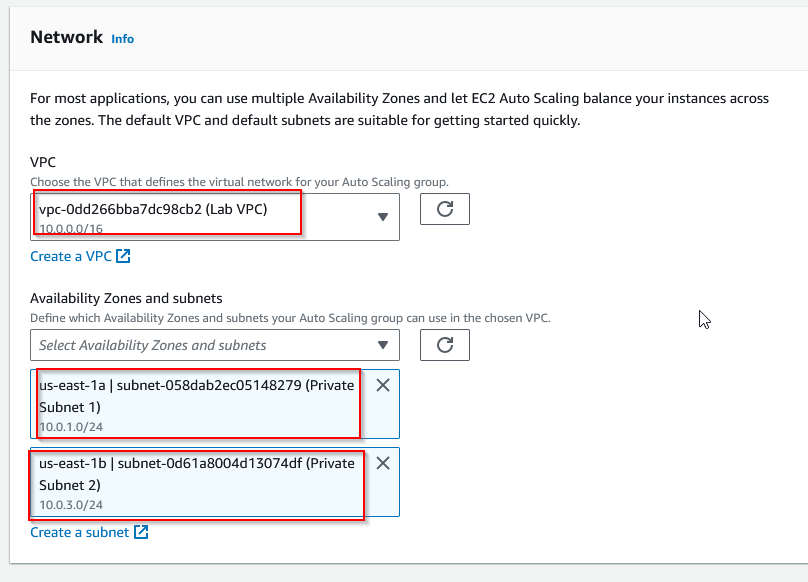
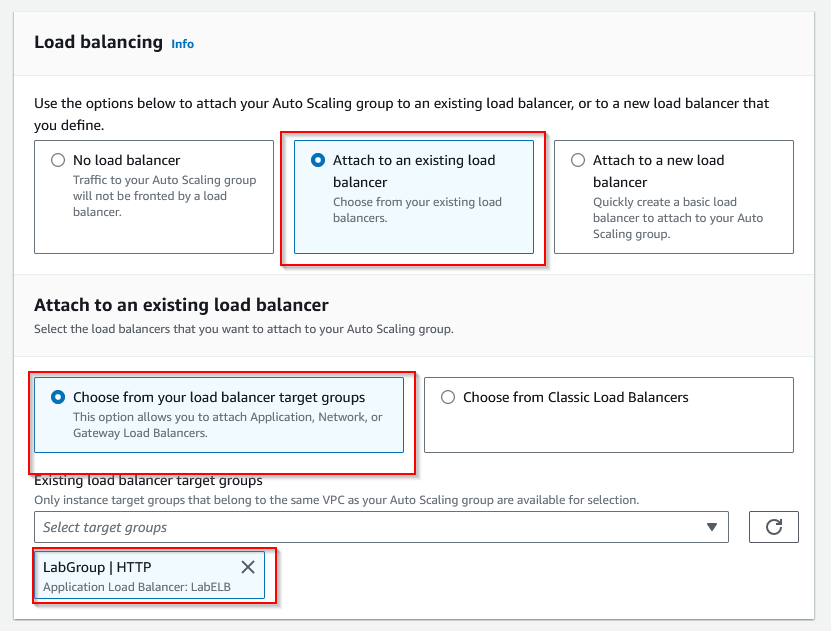
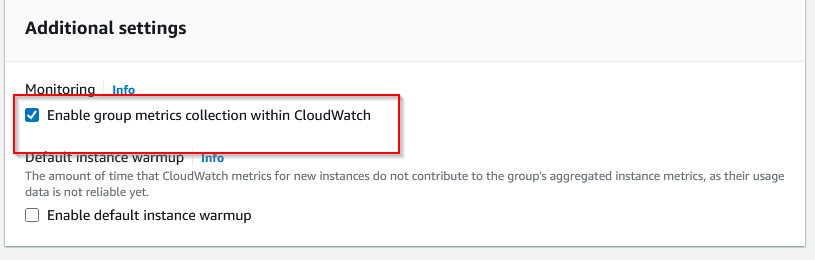
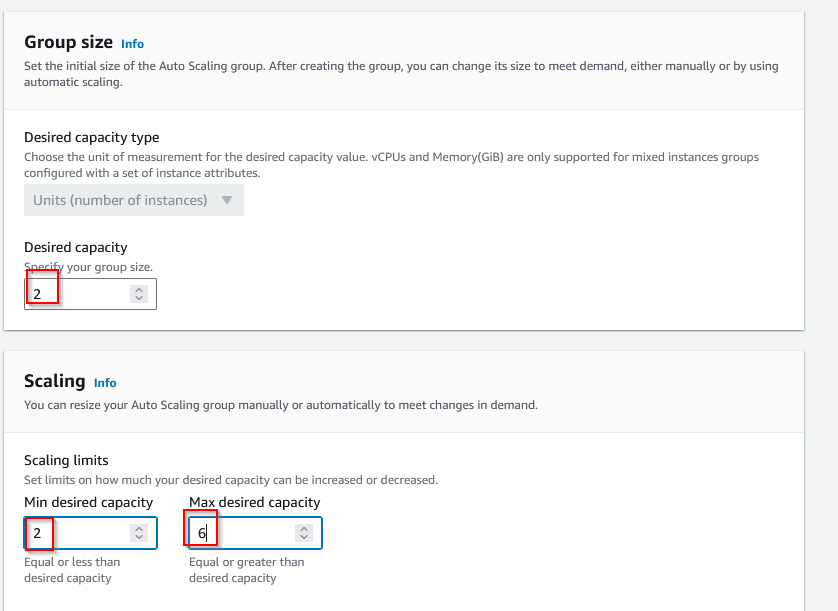
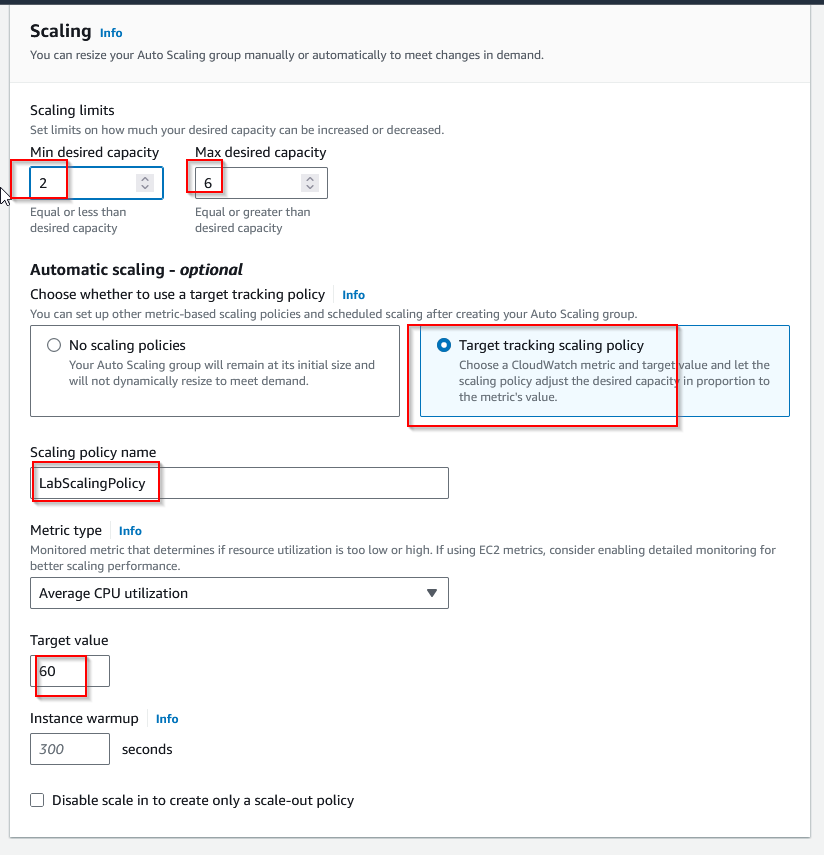
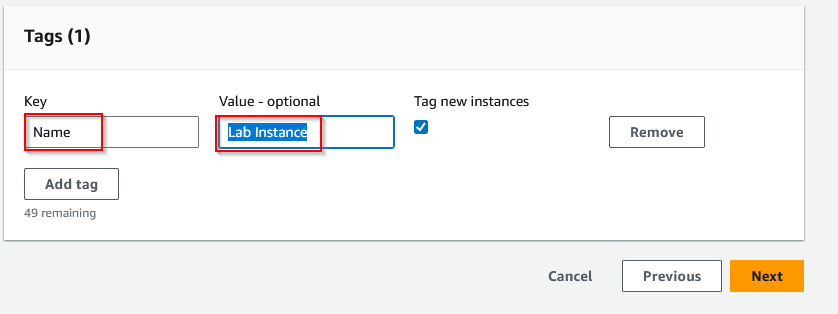
1. When the Web Server has finished loading, create a new image based on the Web Server. Then, create a new target group titled “LabGroup” with the given settings.

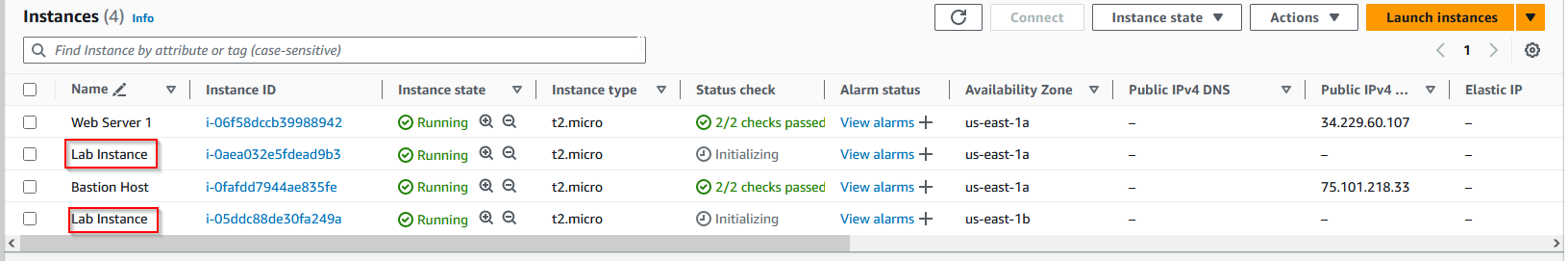
1. Now, go to the Load Balancers section and create a new Load Balancer with given settings.

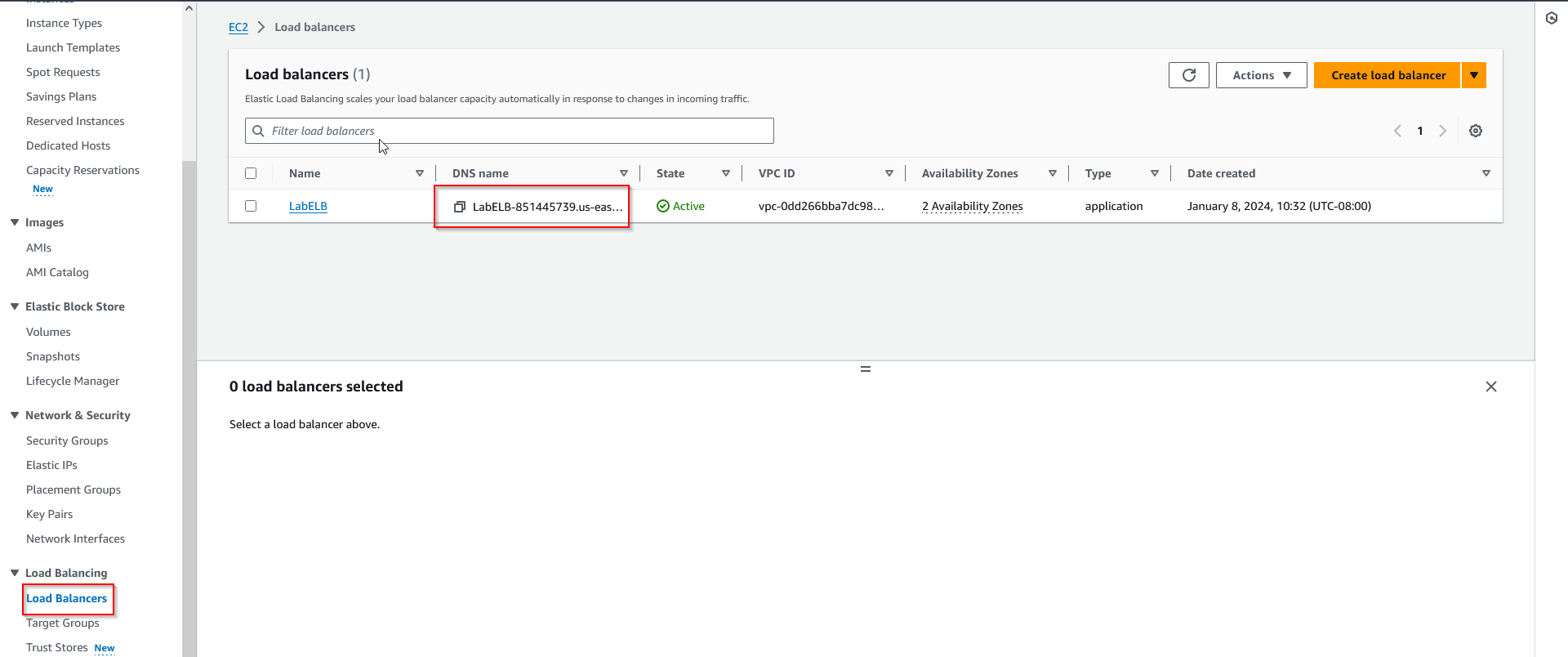
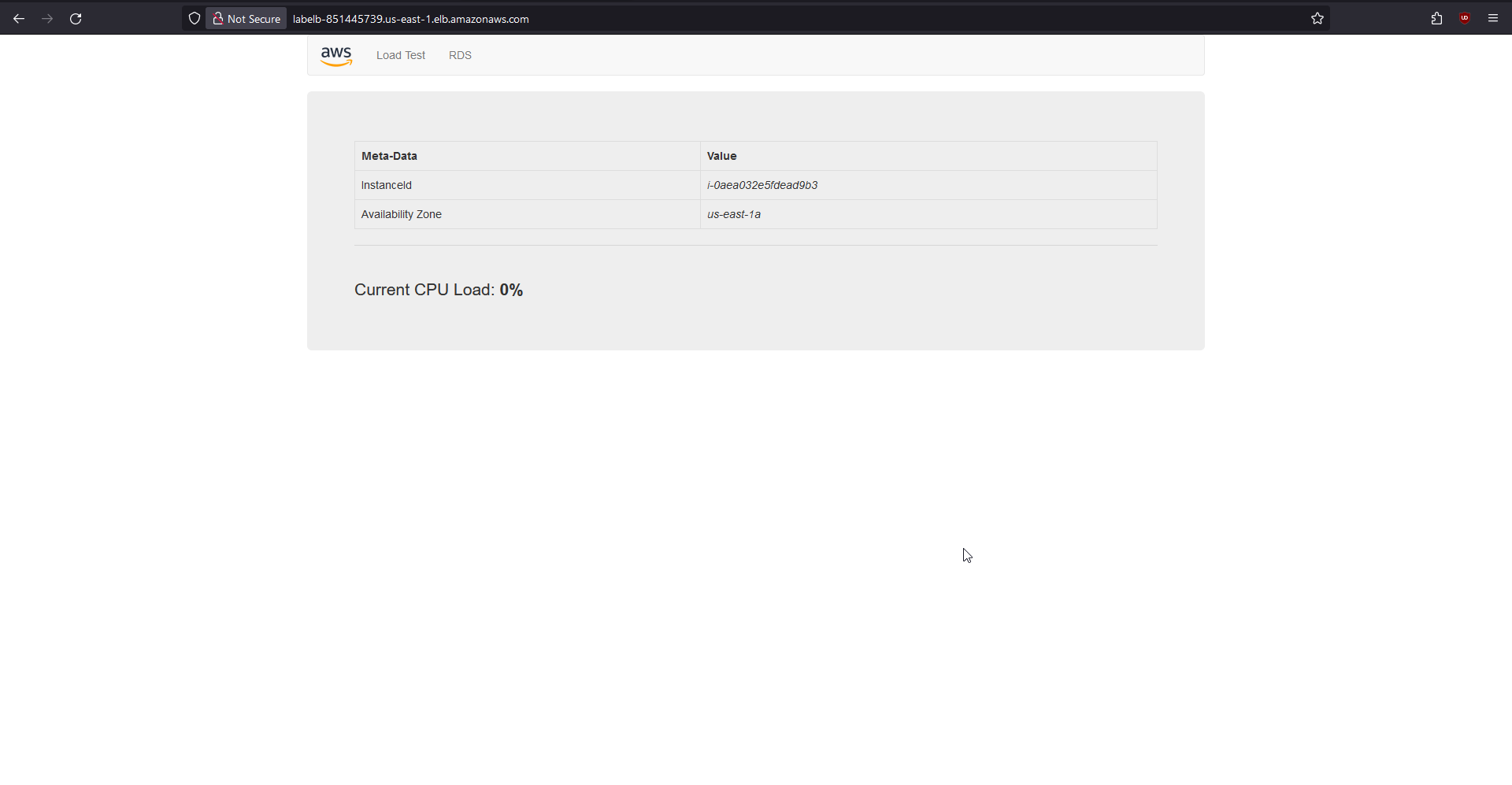
      

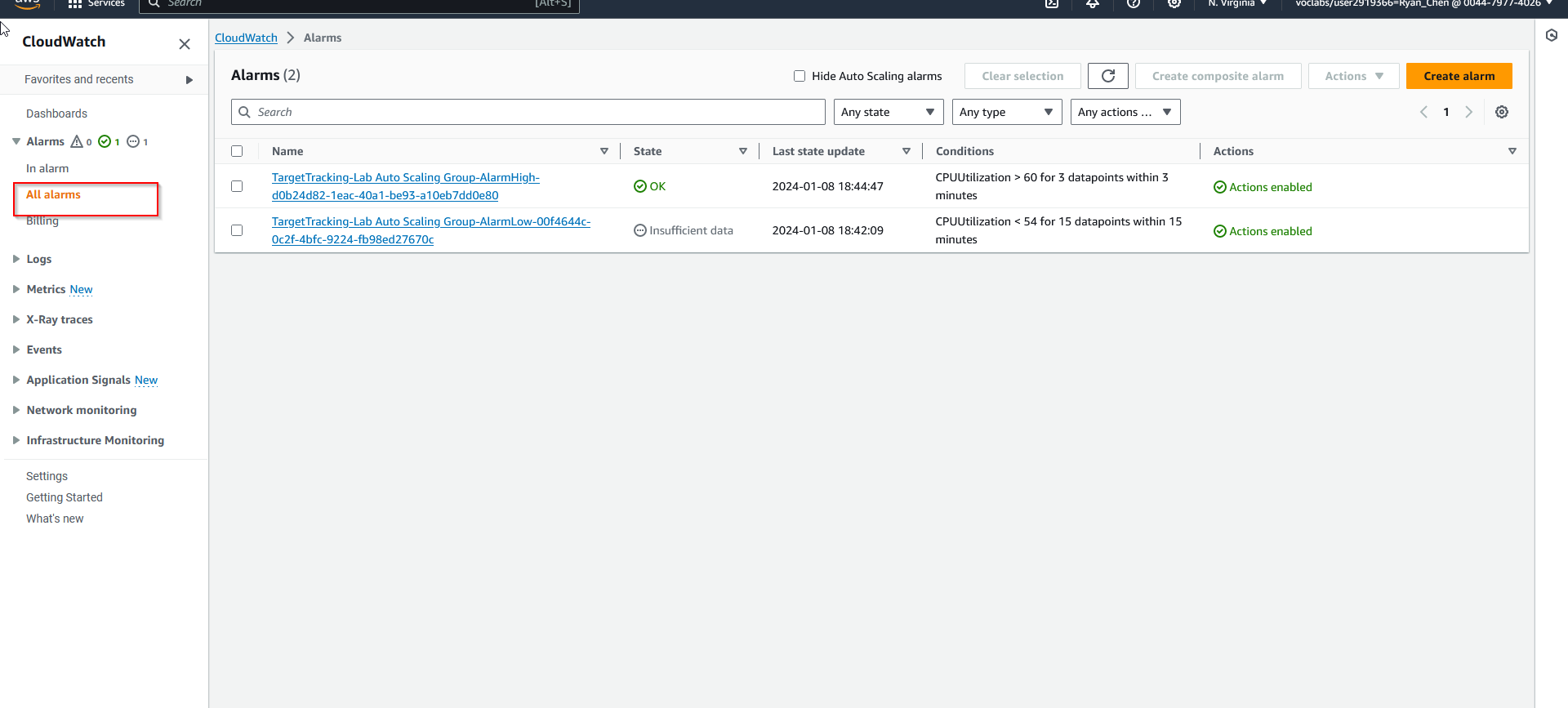
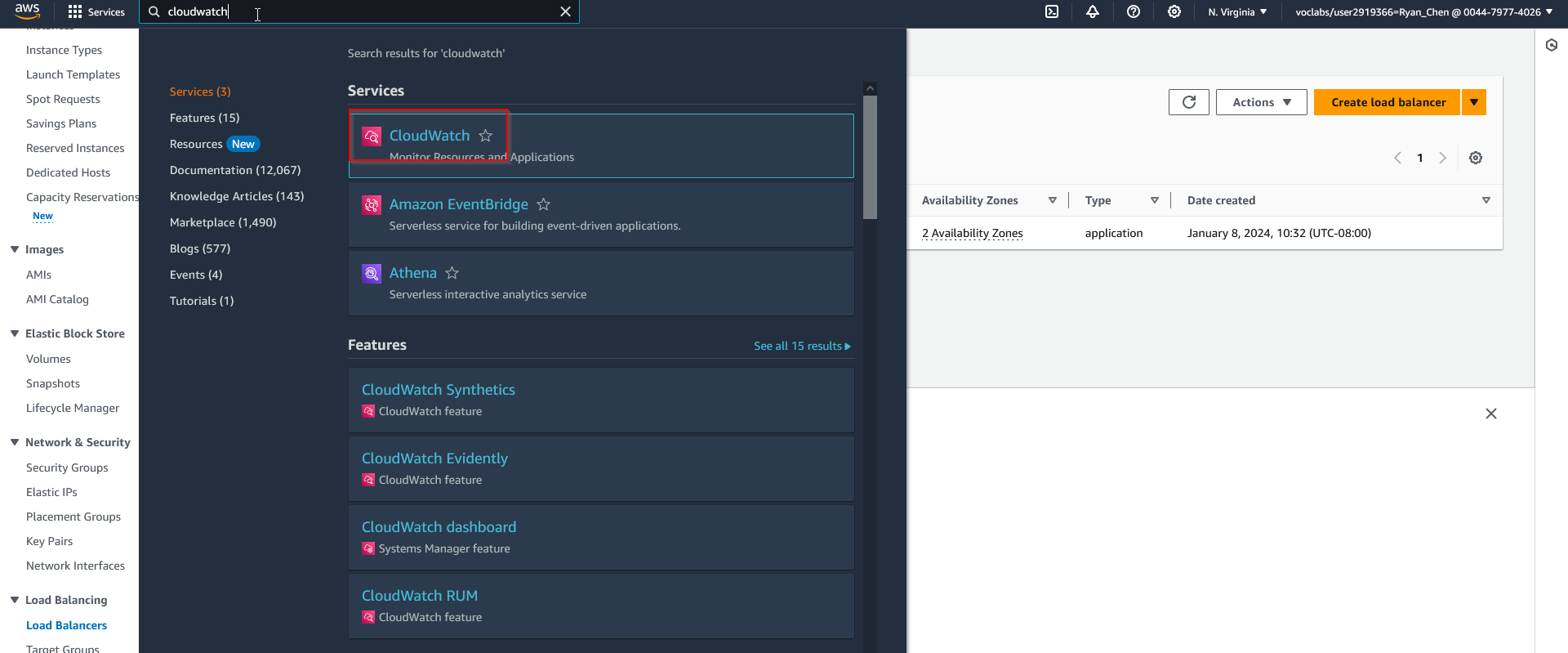
1. Now, create a launch template for the autoscaling group. Navigate to Launch Templates and create a new template using given settings.      
2. Now, create an autoscaling group that utilizes this launch template. Navigate to the Actions menu and create a new auto scaling group with given settings.

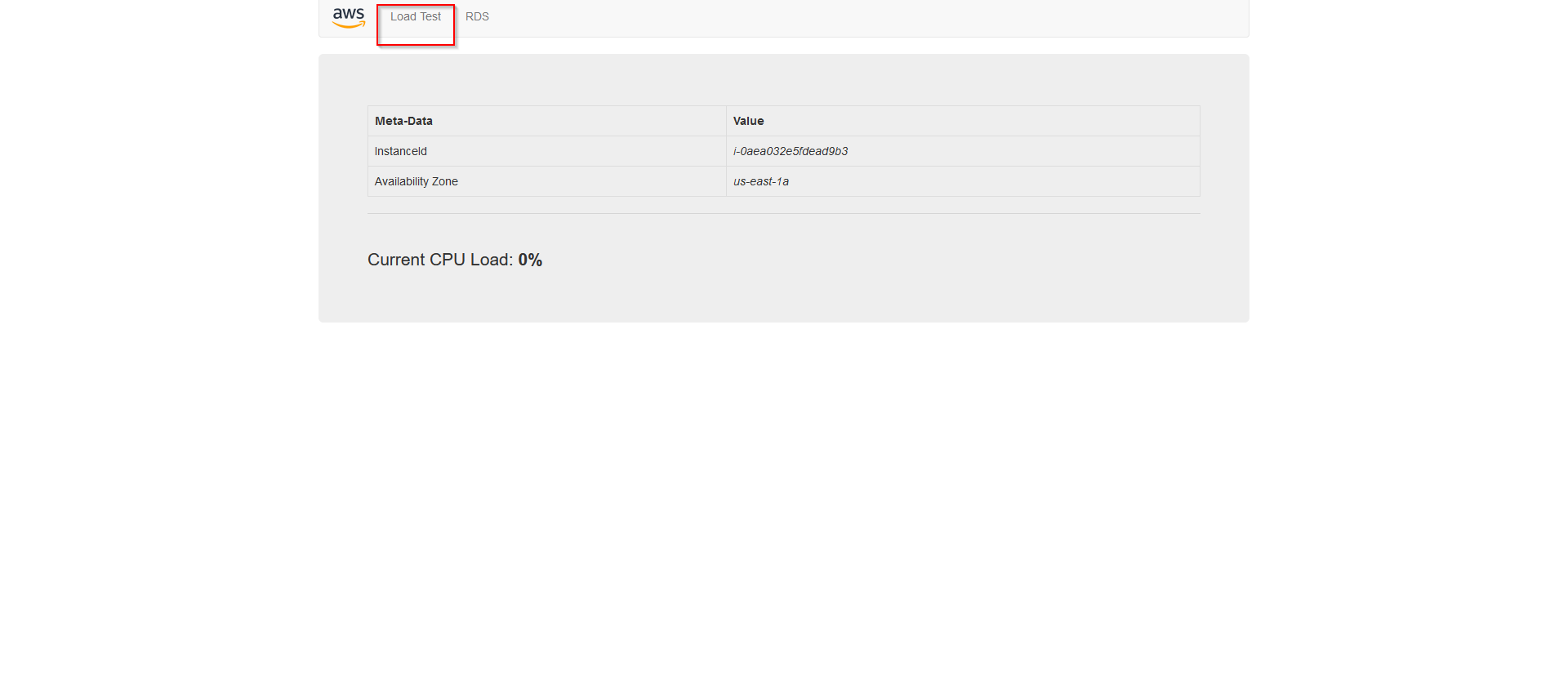
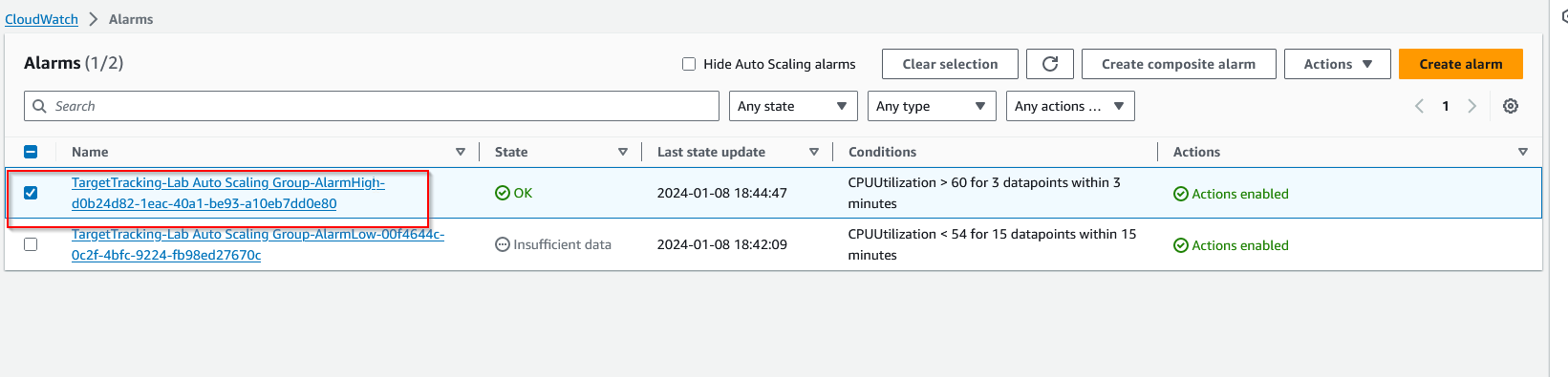
1. Load balancing and autoscaling have now been successfully deployed. Now, we have to verify their condition. In the Instances panel, we can see two new instances created by our configured load balancing services.

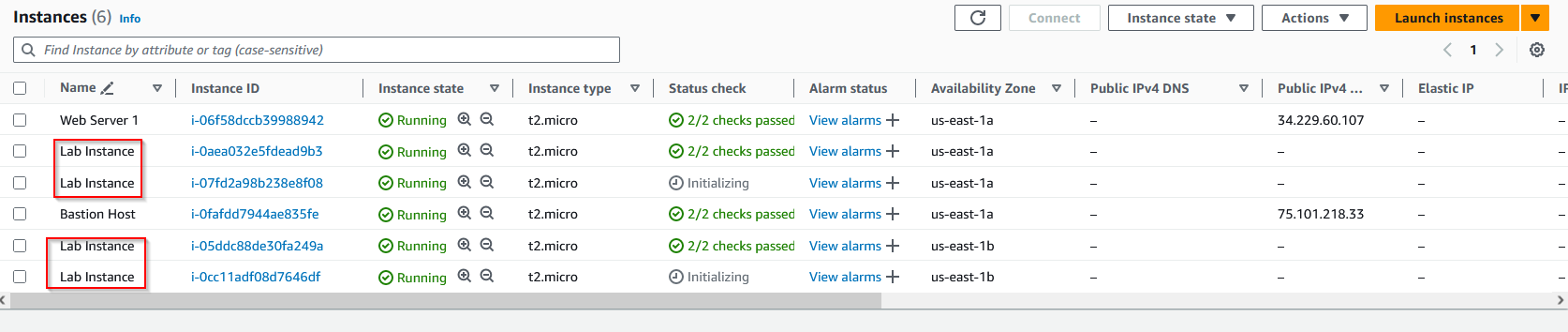
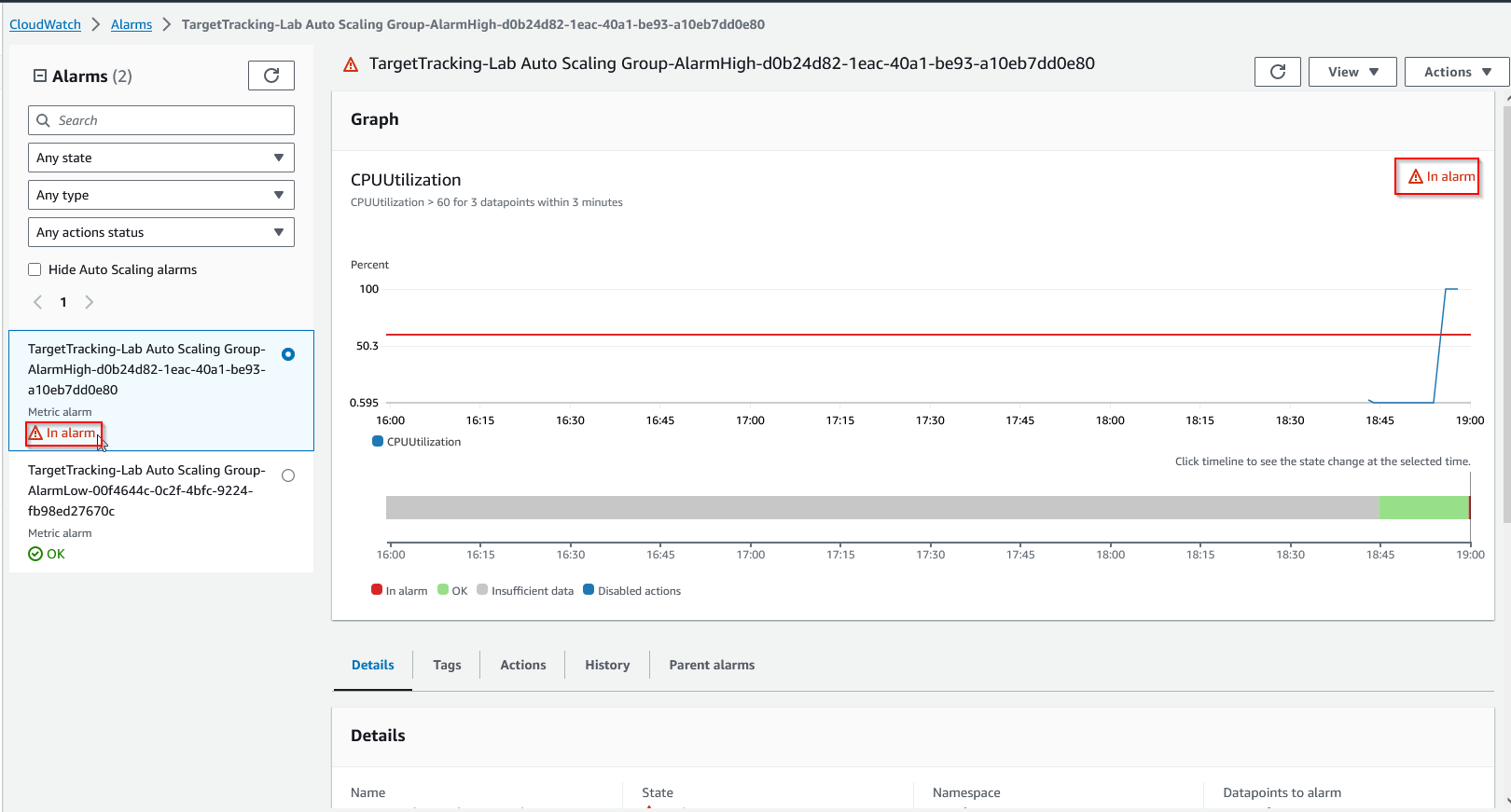
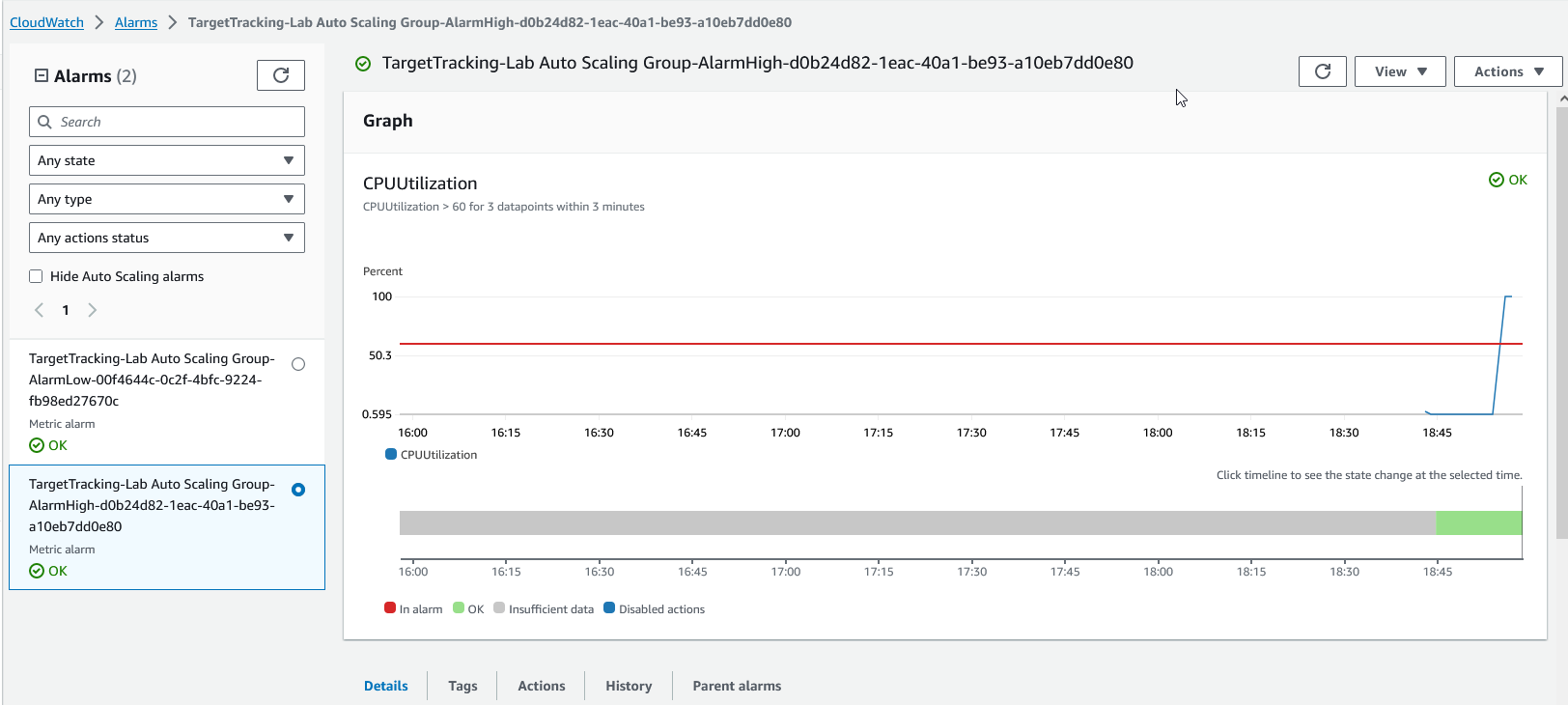


1. Check that the load balancing by pasting the DNS link of the service into your browser and verifying that the request is loaded. 
2. Now, verify that auto scaling is working by increasing the load to cause auto scaling to create more instances. To do so, we will use Amazon Cloudwatch, which allows us to stress test our infrastructure.



1. Choose the High alarm and perform a stress test using built in options from the deployed service.



1. Observe CPU usage increasing. Once CPU usage has been over 60% for 3 minutes, the alarm will trigger more instances to deploy via auto scaling. 

**Conclusion:**

Labs 4-6 serve as a more practical demonstration of AWS services in the real world. Setting up recovery snapshots demonstrates real world redundancy, database creation and usage mimics real world web applications, and auto scaling provides a realistic demonstration of sudden usage spikes. All of these situations build upon previously learned services such as EC2, and allow us to practice our skills to be prepared for configuring real world systems.