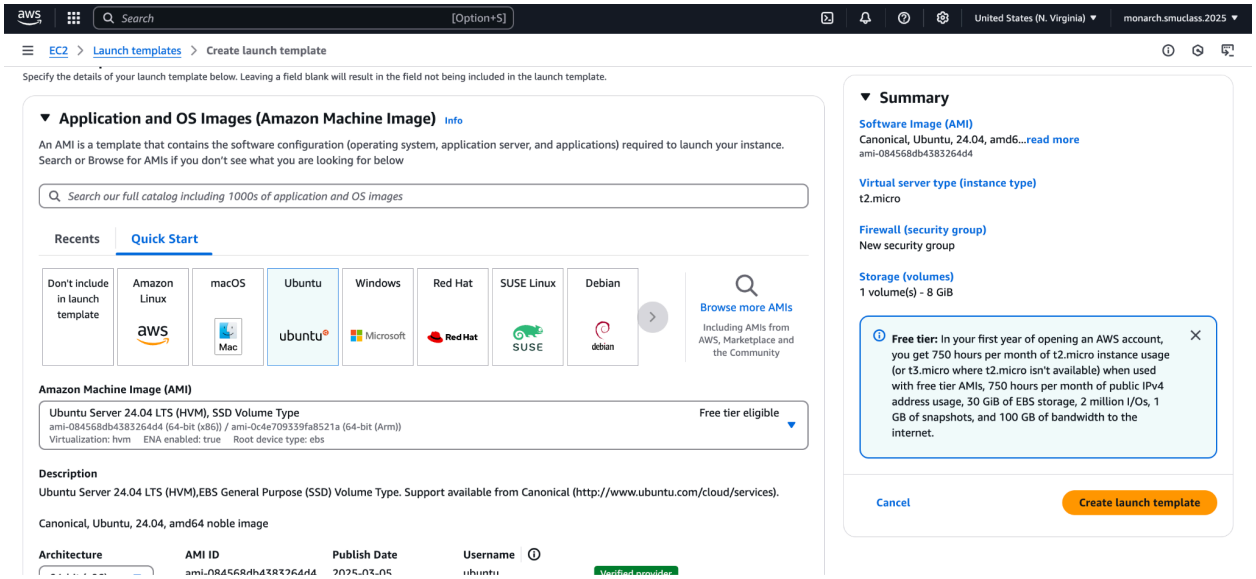
EXERCISE 11.1

Configure and Launch an Application Using Auto Scaling

1. From the EC2 Dashboard, create a launch configuration using the Ubuntu Server LTS AMI on the Quick Start tab and a t2.micro instance type.

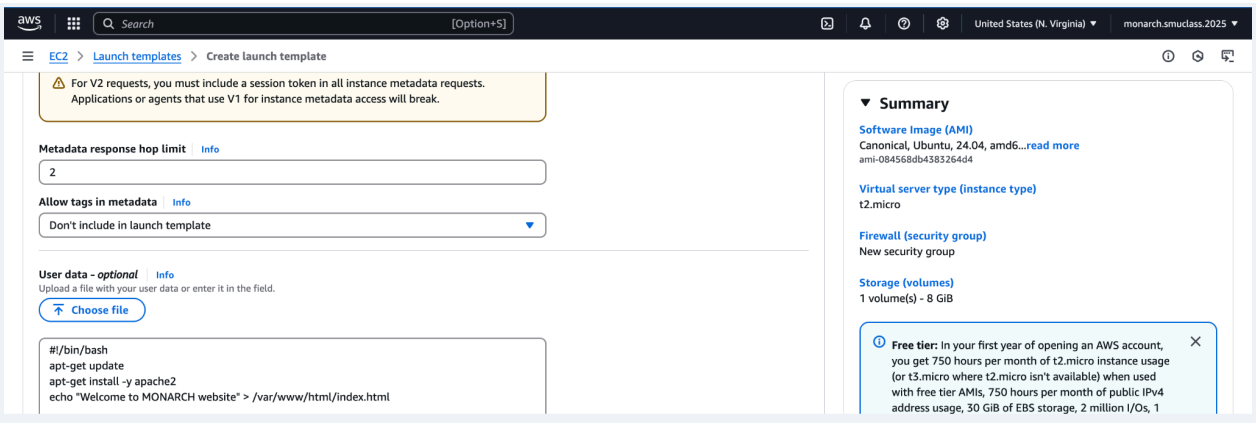


2. On the Create Launch Configuration page, name your configuration. You don't need to select an IAM role or enable monitoring for this exercise.

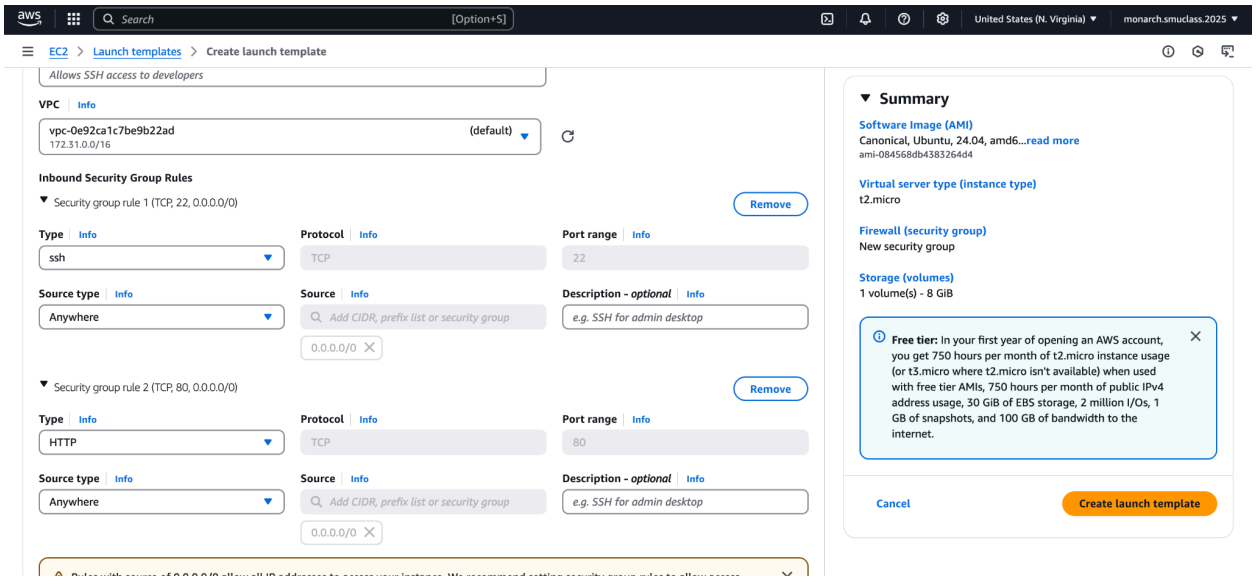
Save the following commands to a file on your local computer, name the file `start.sh`, expand the Advanced Details section, click the As File radio button, and select your `start.sh` script from your computer. This will install the Apache web server and create a simple `index.html` web page:

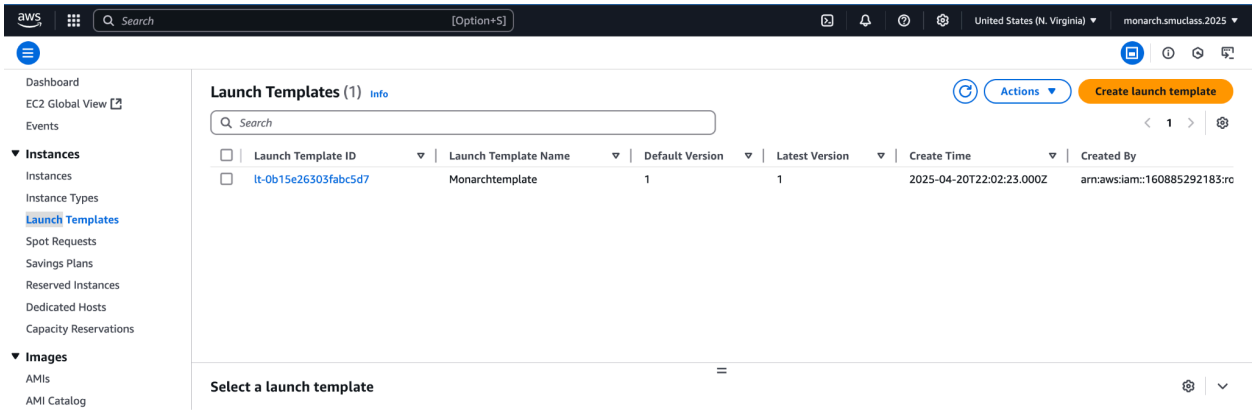
```
    #!/bin/bash
```

```
apt-get update
apt-get install -y apache2
echo "Welcome to my website"> index.html
```

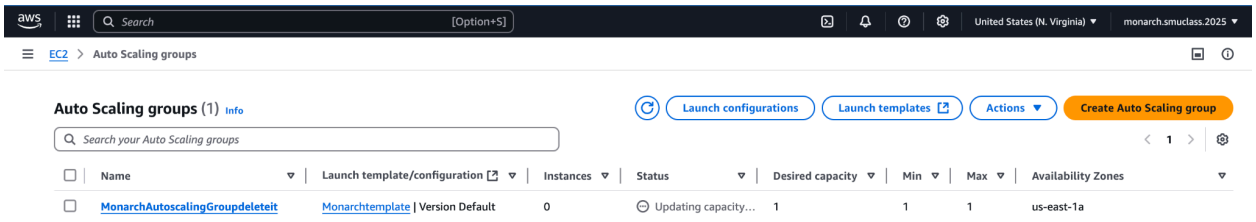


3. `cp index.html /var/www/html`
4. Choose (or create) a security group that will permit all HTTP traffic via port 80 and then create the configuration, making sure to select a valid key pair so that you'll be able to use SSH to log into the instance later if necessary.

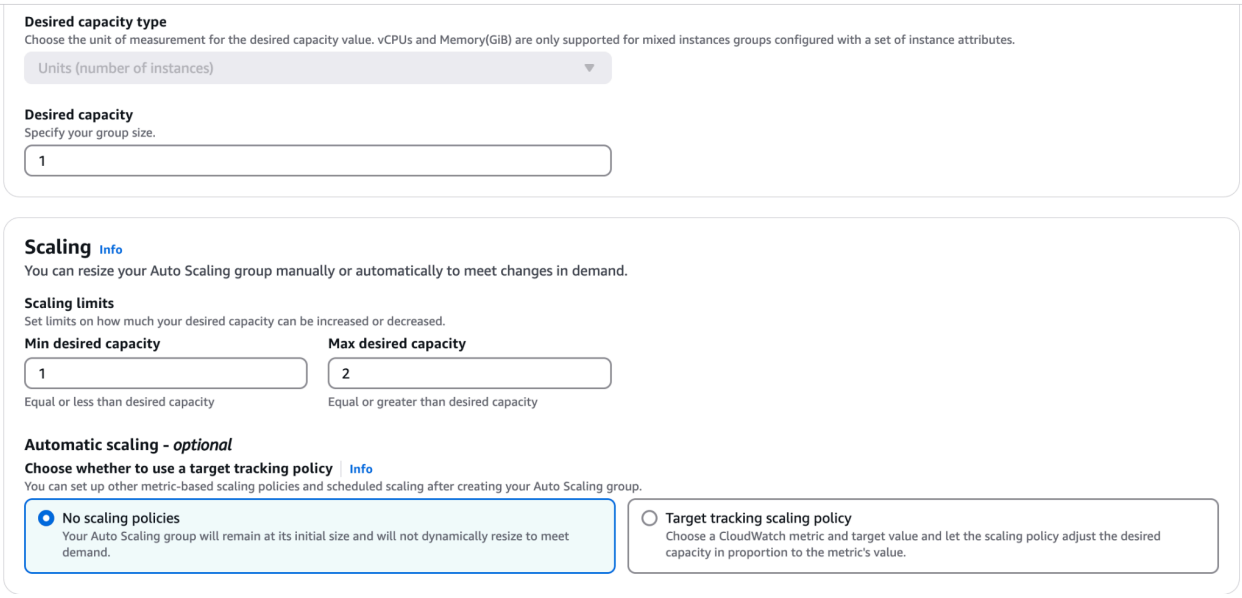




5. Create an Auto Scaling group that will use your new launch configuration. Give your group a name and leave the Group Size value at 1 instance.

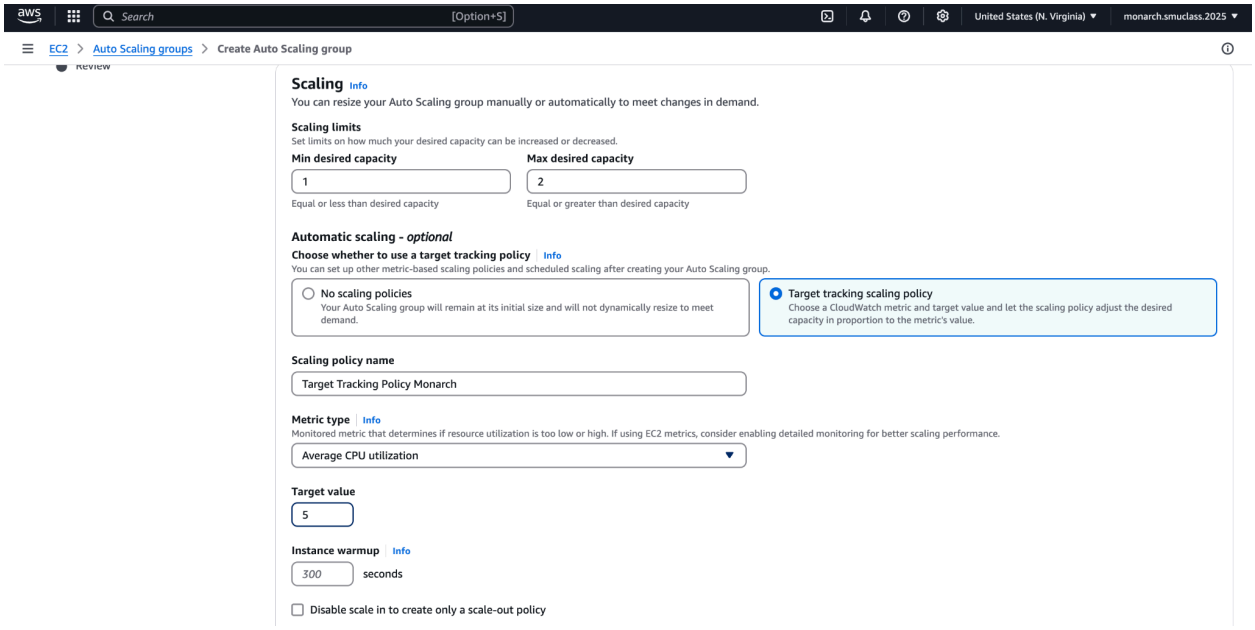


6. Select a virtual private cloud (VPC) into which your instances will launch, click once inside the Subnet field, and select the subnet where you'd like your instances to live. Assuming you're working with your default VPC, any one of the public subnets you'll be shown should be fine.
7. On the Configure Scaling Policies page, set Minimum Capacity to 1 and Maximum Capacity to 2.

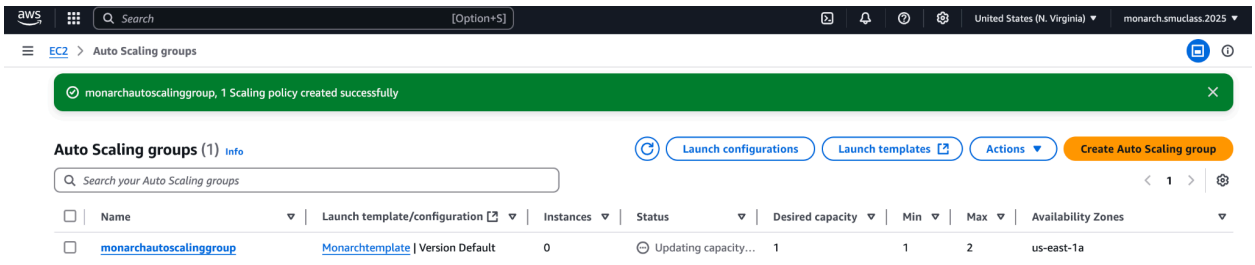


8. Select Target Tracking Scaling Policy and then leave the Metric Type value as Average CPU Utilization and edit its Target Value to 5 percent.

   This unusually low value will make it easier to confirm that the autoscaling is working—normally a value of 70 percent to 80 percent would make sense.

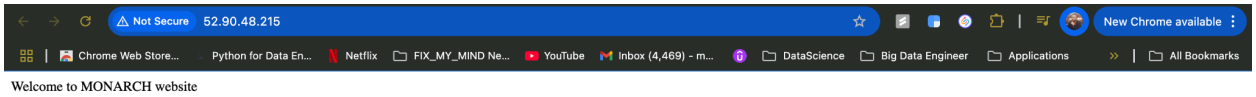


9. For this exercise, you don't need to configure notifications or tags. Click Create to finish creating the group.



10. Your group will immediately try to start the first instance. It may take a few minutes for everything to load. When it does, point your browser to the IP address associated with the instance (you can retrieve that from the EC2 Instances Dashboard).

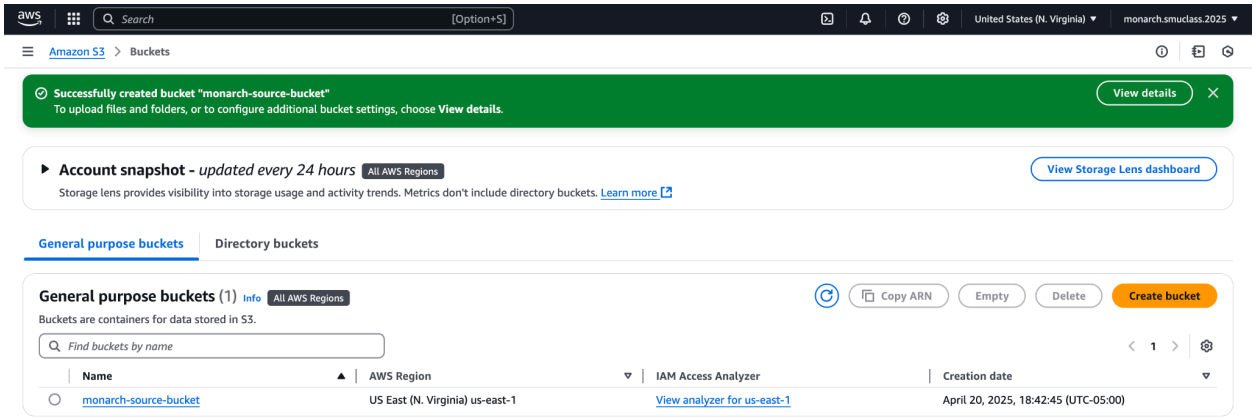
Considering how small the `index.html` page is, it might be hard to get the Auto Scaling tool to launch a second instance. You'll probably need to "cheat" by running a busywork command such as the following on the server command line to raise your CPU level high enough:

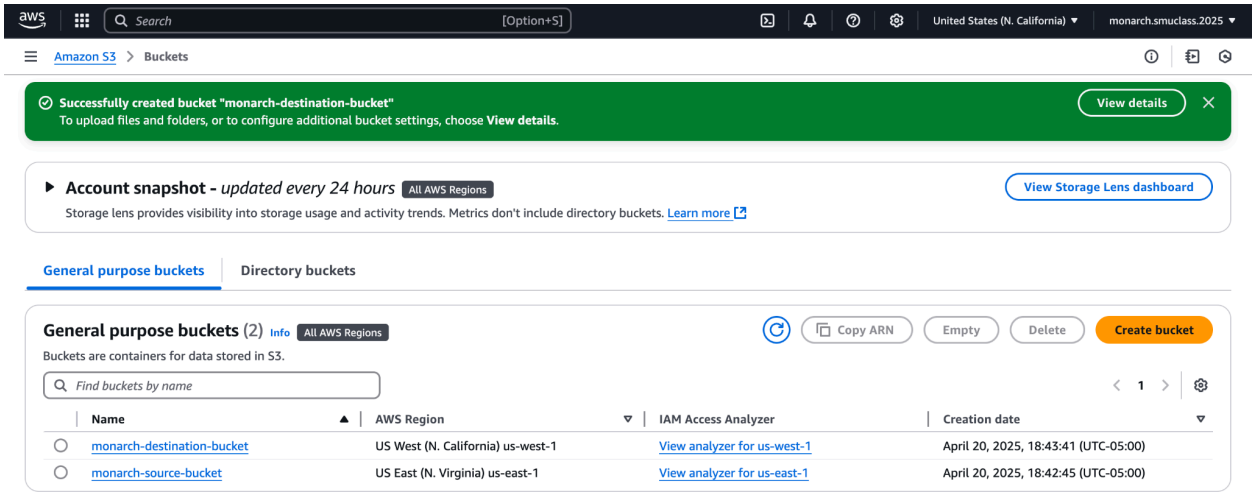


EXERCISE 11.2

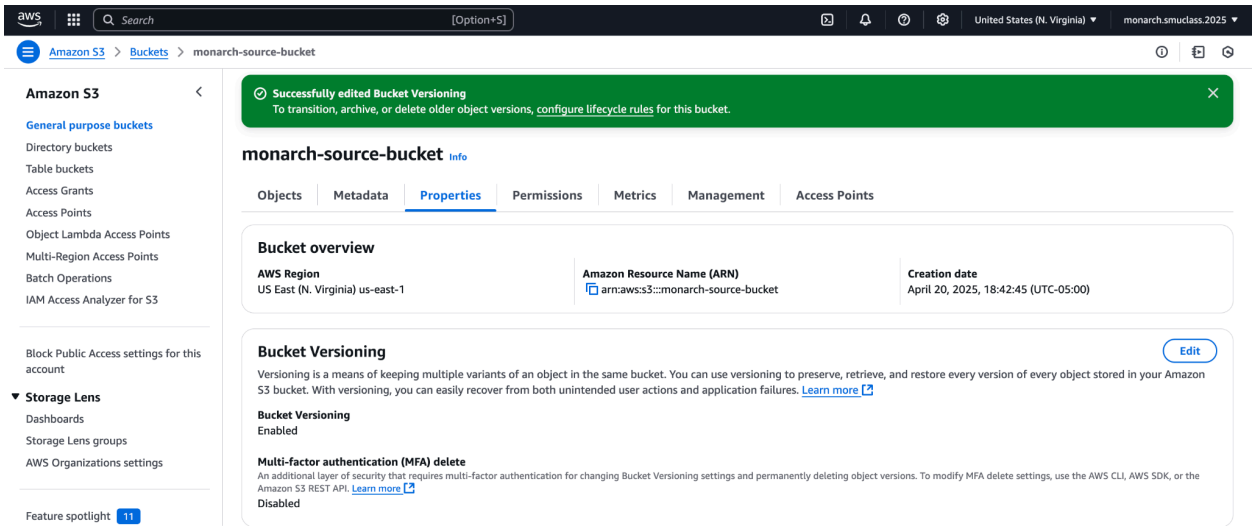
Sync Two S3 Buckets as Cross-Region Replicas

1. Use the S3 Console to create two buckets in two separate regions. One will be your source and the other your destination.



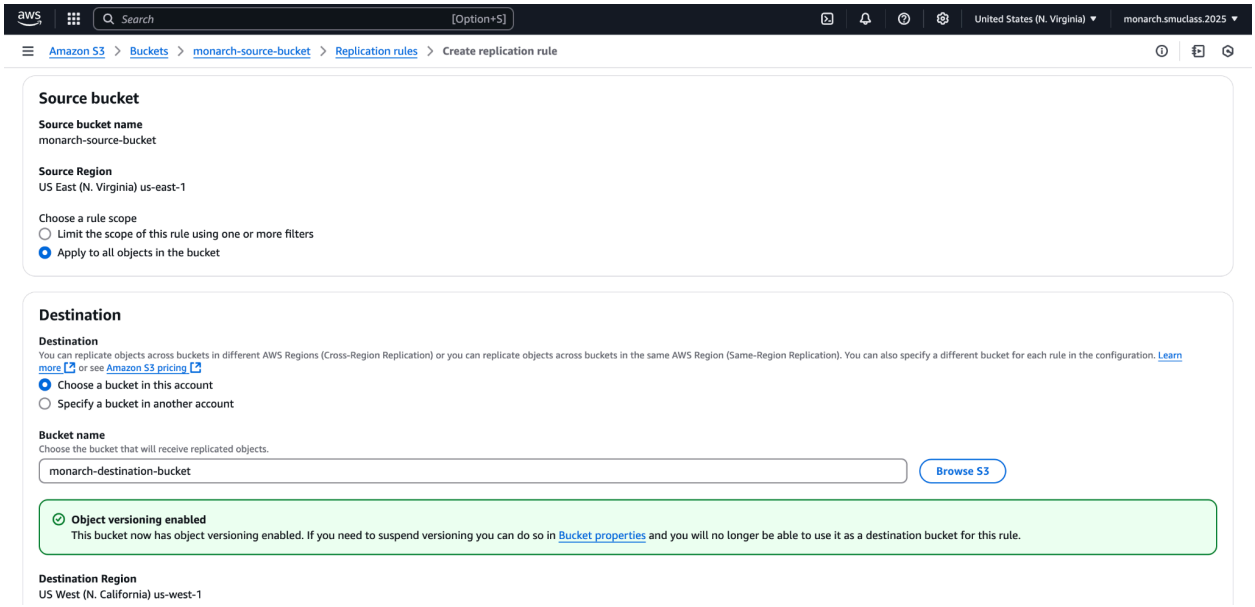


2. Click the name of your source bucket in the S3 Console, then click the Management tab, then the Create Replication Rule button. If versioning wasn't already enabled for the bucket, you can click the Enable Versioning button to set it.

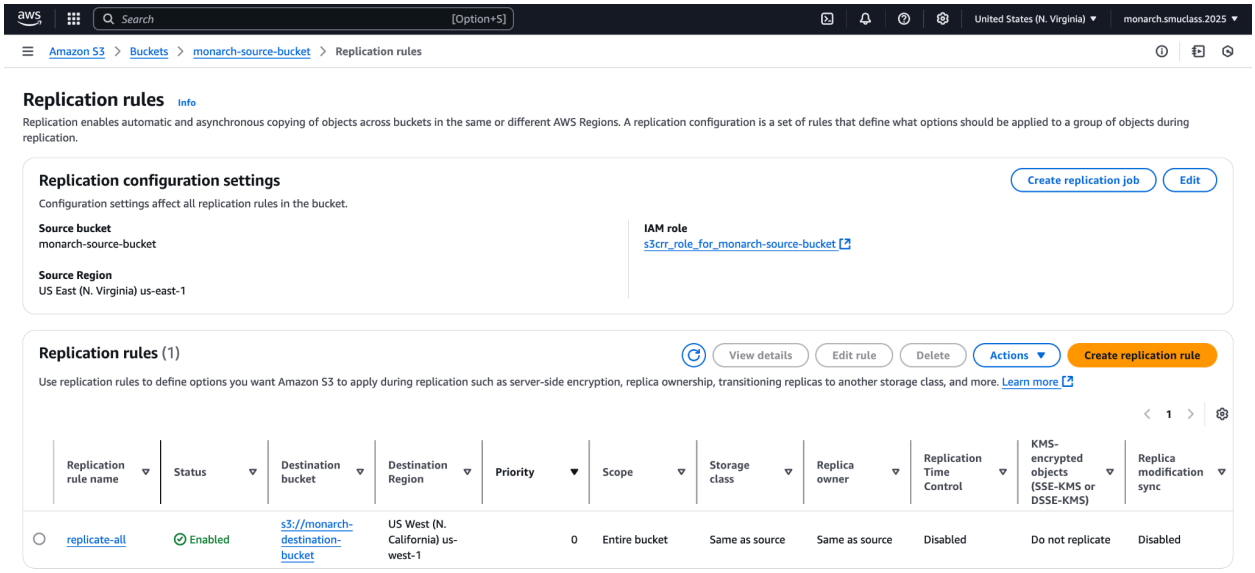


3. Define the objects you want replicated from your source bucket. Choose Apply To All Objects for this exercise.

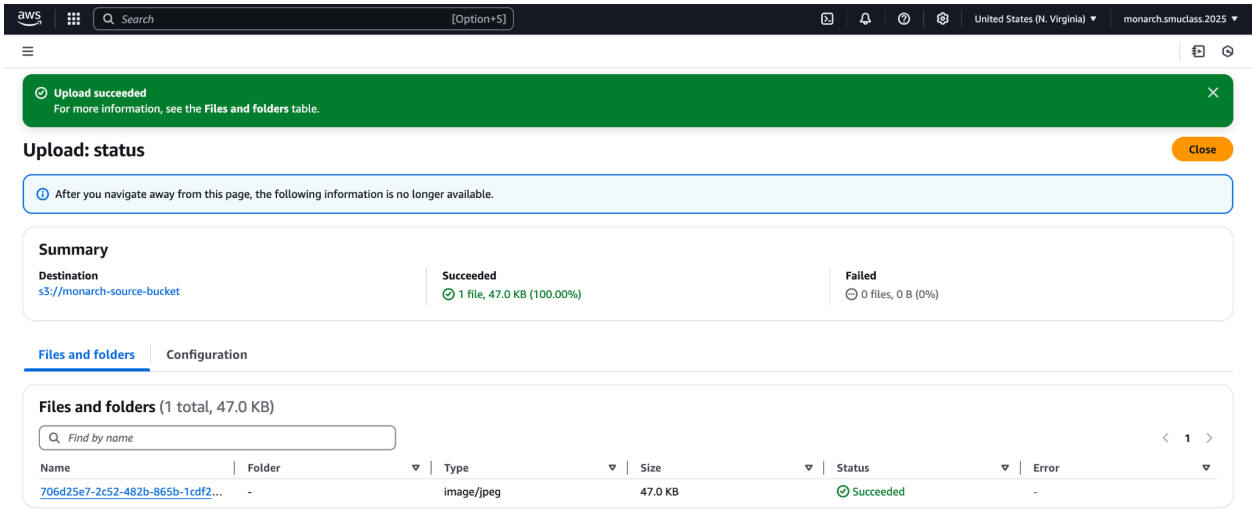
4. In the Destination Definition section, select the name of the second bucket you created as the destination.

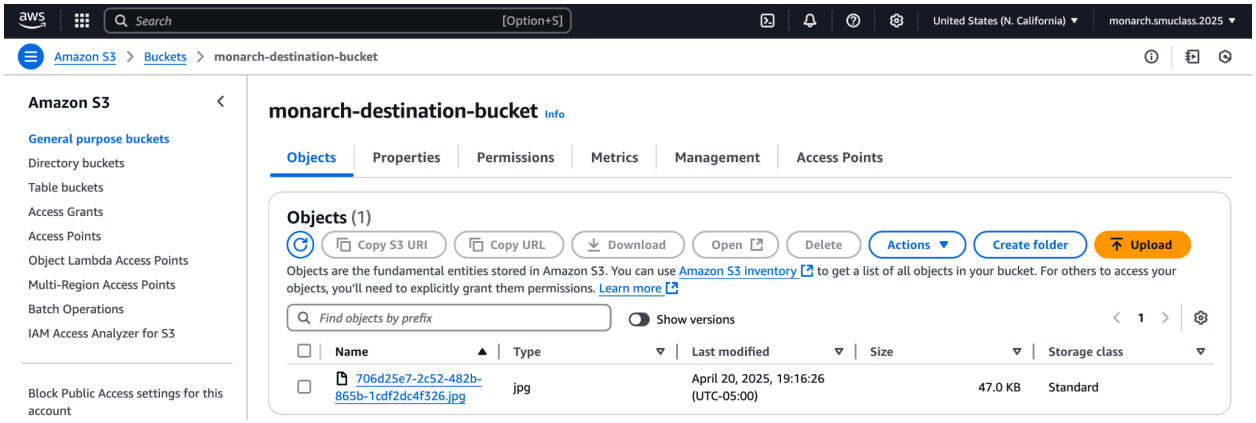


5. Choose From Existing IAM Roles and select Create New Role from the drop-down menu. This will create the permissions necessary to move assets between buckets. You can then review your settings and save the rule.



6. Save your rule and confirm the replication is working by uploading a file to the source bucket and then checking to see that a copy has been created in the destination bucket.



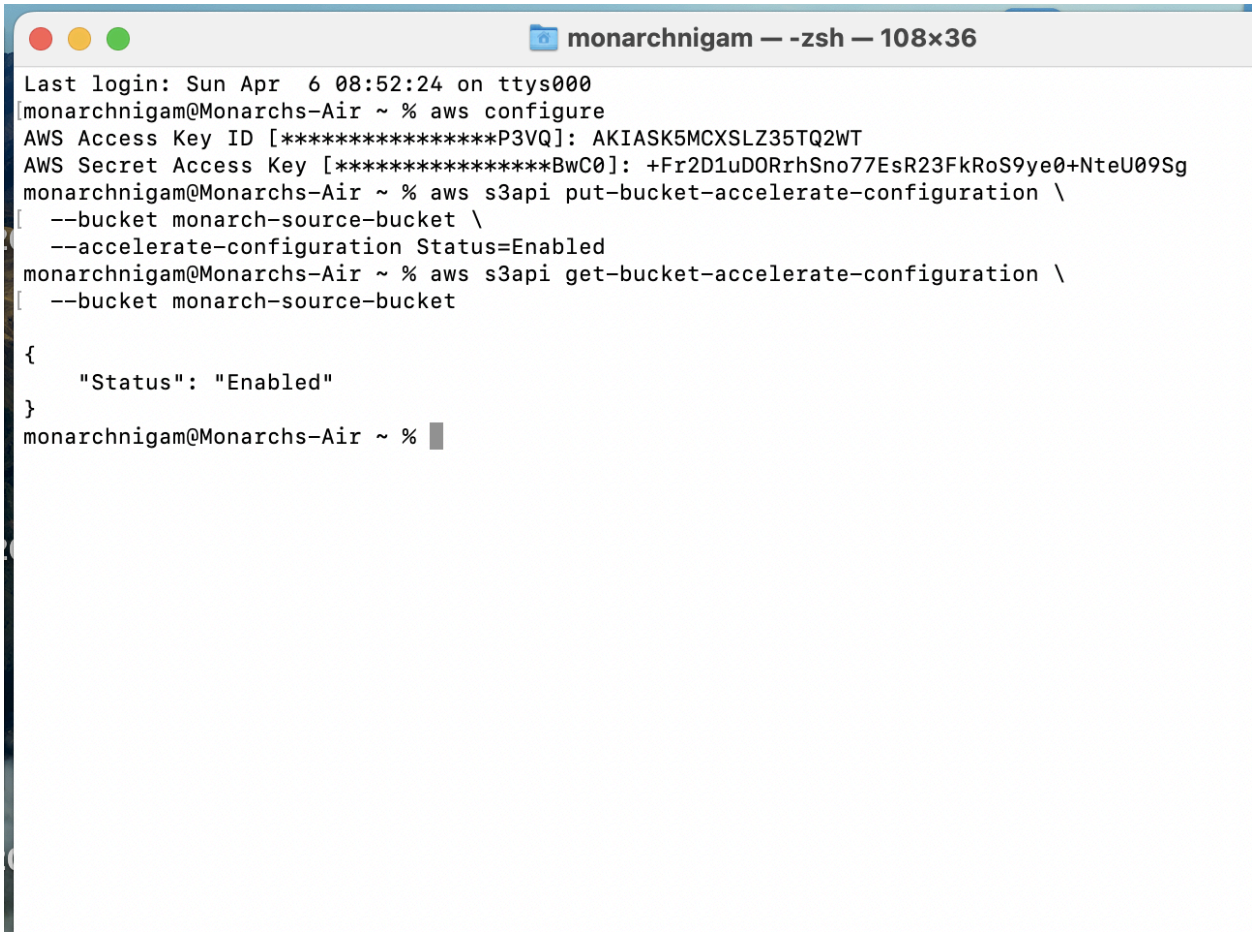


As we can see the uploaded file(image), has automatically replicated in the destination bucket

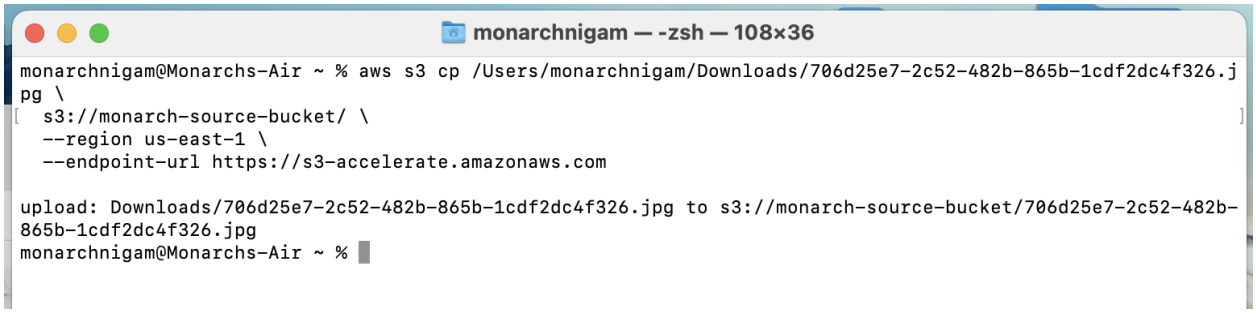
EXERCISE 11.3

1. Upload to an S3 Bucket Using Transfer Acceleration Enable S3 Transfer Acceleration on an existing bucket (substitute the name of your bucket for my-bucket-name). $ aws s3api put-bucket-accelerate-configuration \ --bucket my-bucket-name \ --accelerate-configuration Status=Enabled

In below screenshot we can see commands successful run



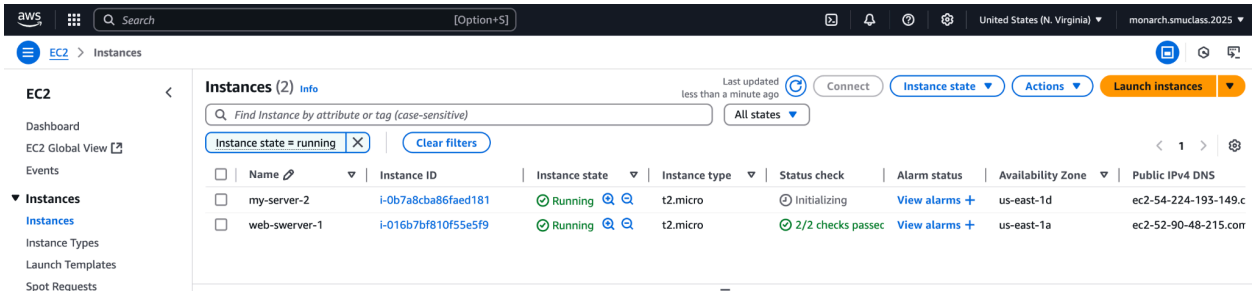
2. Transfer a file by specifying the s3-accelerate endpoint. Make sure you're using the correct file and bucket names and region. $ aws s3 cp filename.mp4 s3://my-bucket-name \ --region us-east-1 \ --endpoint-url http://s3-accelerate.amazonaws.c



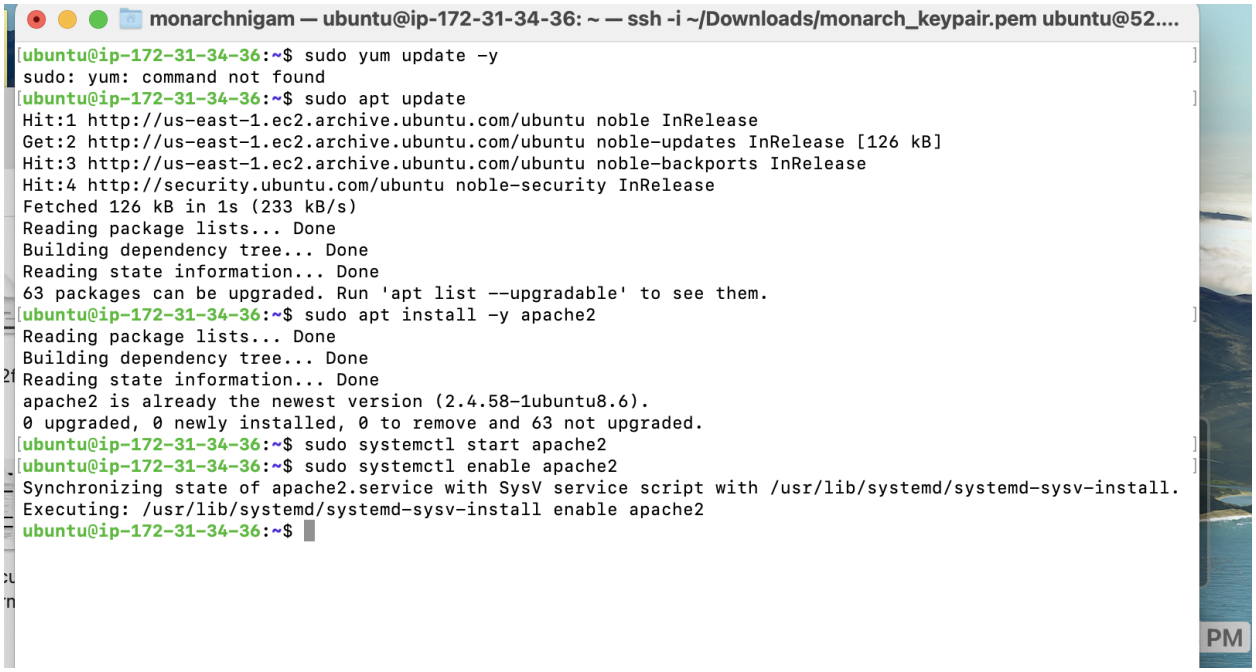
EXERCISE 11.4

Create and Deploy an EC2 Load Balancer

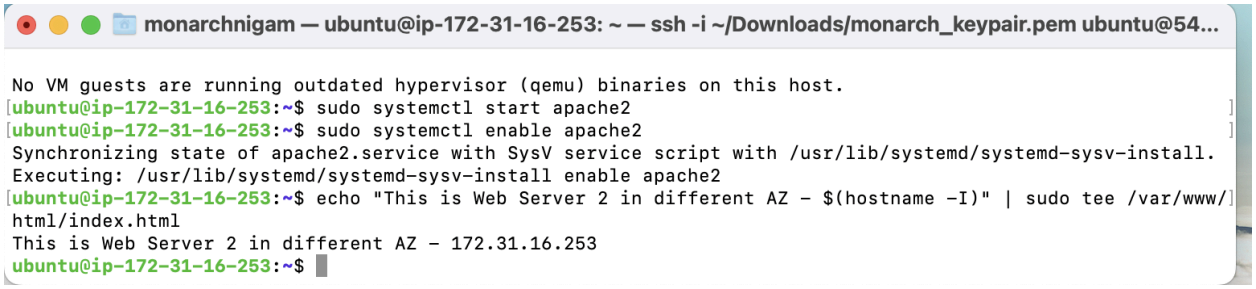
1. Create one instance in each of two subnets. You select the subnets on the Configure Instance Details page. Make a note of the subnet designations you choose

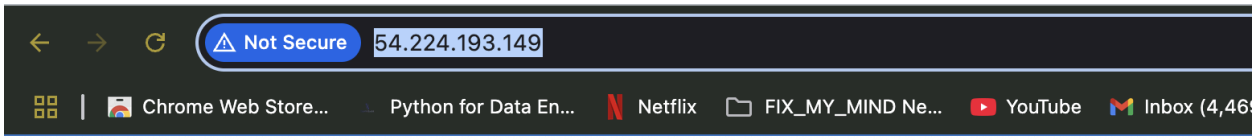


2. Use SSH to access your instances, install the Apache web server, and create unique `index.html` files containing a short note describing the server—perhaps including the server's IP address. This will make it easier to know which instance your load balancer has loaded when you test everything out later.



This is Web Server 1 - 172.31.34.36

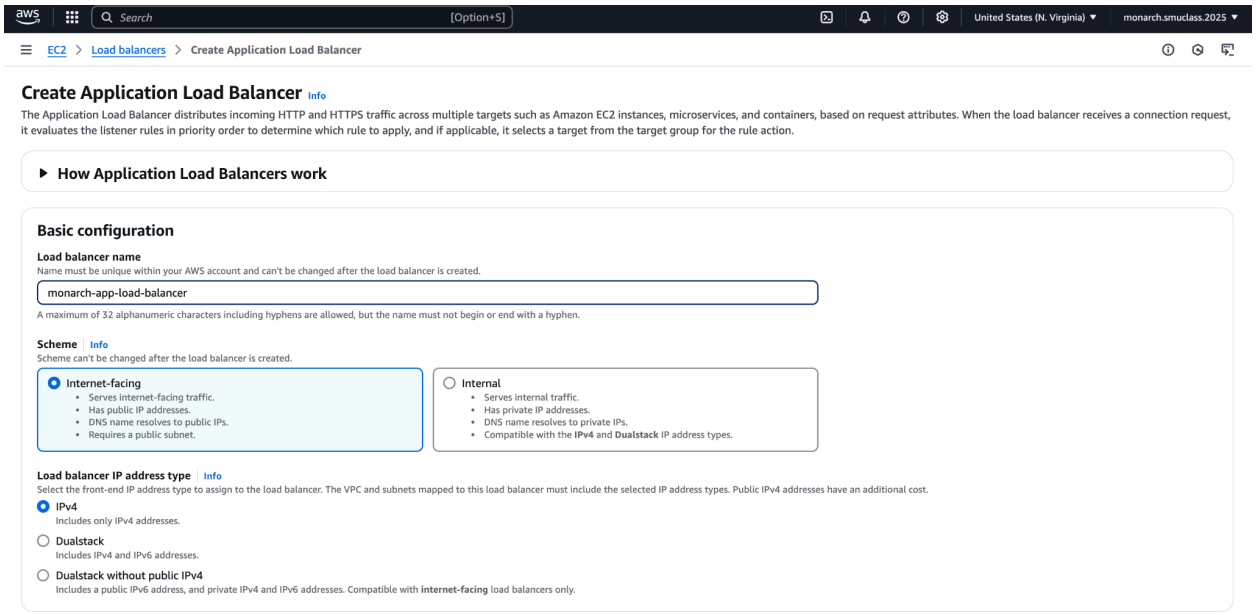




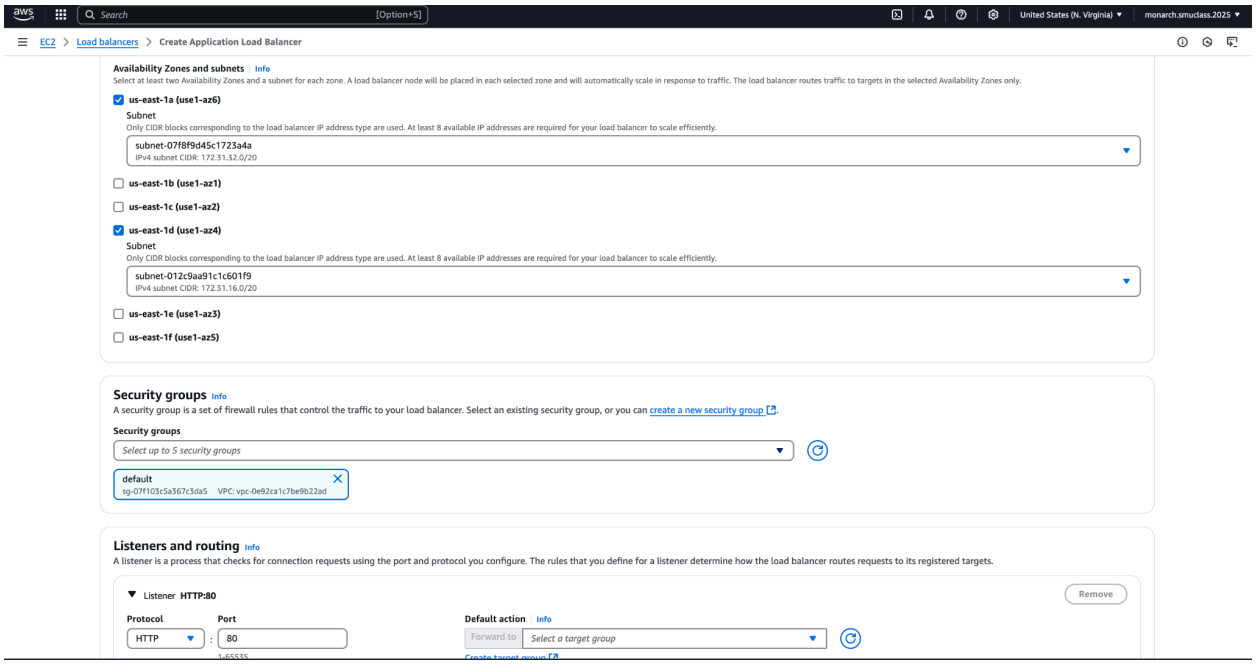
This is Web Server 2 in different AZ - 172.31.16.253

In the above screenshots I have shown that two EC2 instances are running in two Different AZ

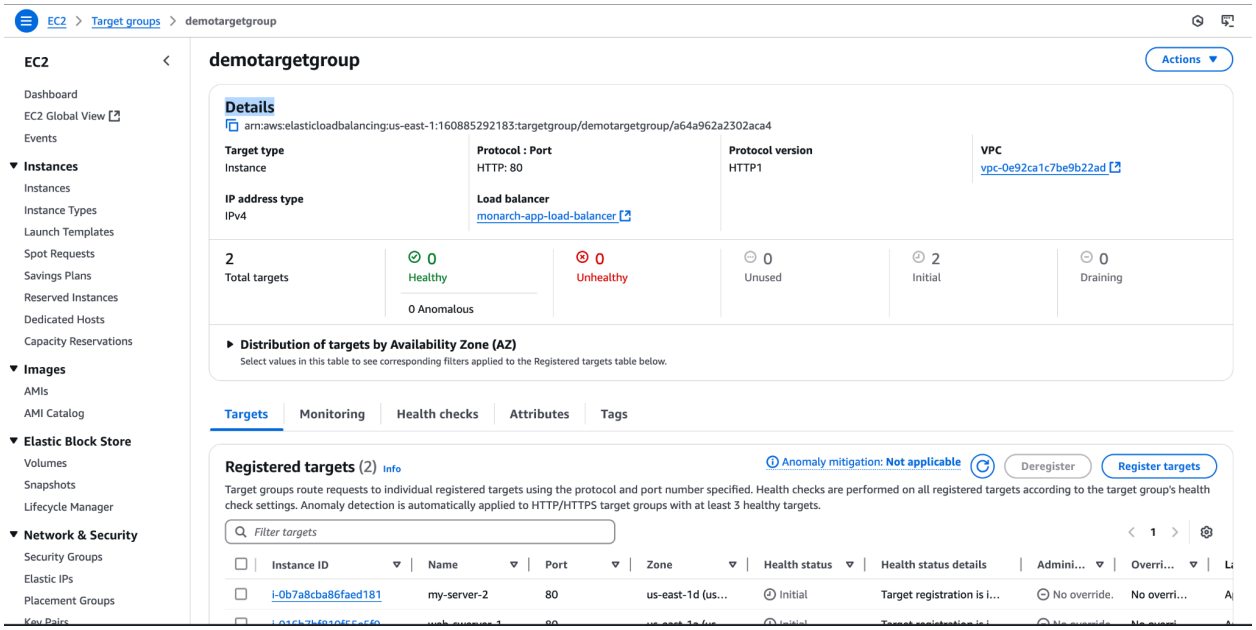
3. From the EC2 Dashboard, create a load balancer. Select an application load balancer, give it a name, and specify the “Internet-facing” scheme, IPv4 for the IP address type, and the default HTTP and port 80 values for the listener. Make sure you're using the right VPC (the default should be fine) and select (and note) two availability zones. For this exercise, you can ignore the warning about using an unencrypted listener.



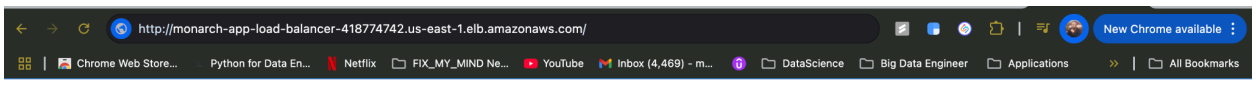
4. Select (or create) a security group that allows access to port 80.



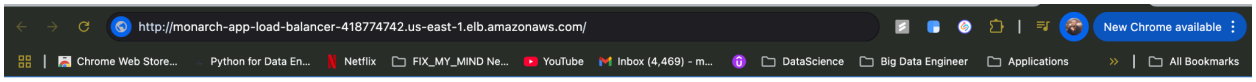
5. Click the Create Target Group link and change the Target Type value to IP and add a path to a file the health check can use. For this exercise, enter `/index.html`.



6. On the Register Targets page, register your two EC2 instances as targets to the group using their private IPs. You can retrieve those private IP addresses from the EC2 Instances console or from the command prompt of the instance itself using the `ip addr` command. - Depicted in the above screenshot
7. Once your balancer is active (which can take a couple of minutes), get the DNS name of your balancer from the Load Balancers console and paste it into your browser. The `index.html` page of one of your instances should load. Refresh the browser a few times so that you can confirm that both instances are being accessed.



This is Web Server 2 in different AZ - 172.31.16.253



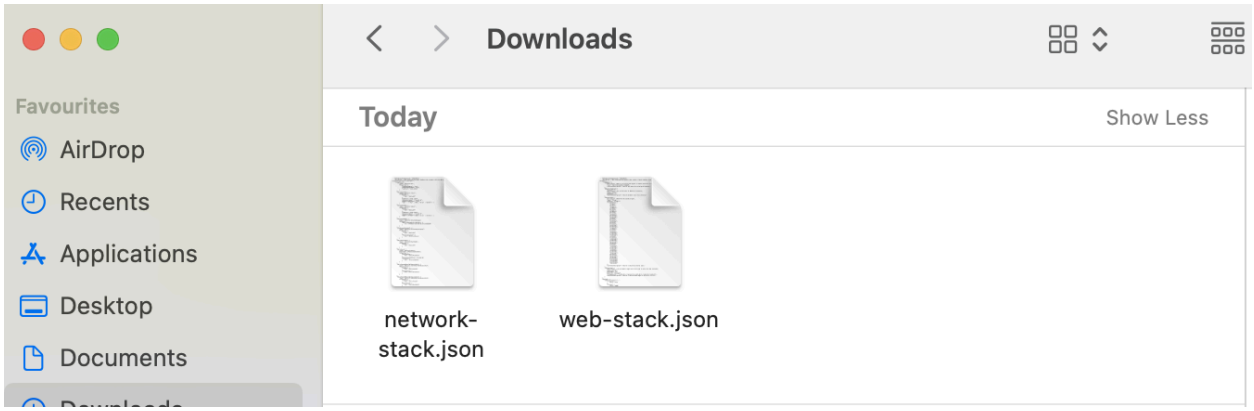
This is Web Server 1 - 172.31.34.36

EXERCISE 11.5

Create a Nested Stack

In this exercise, you'll create a nested stack that creates an EC2 Auto Scaling group and the supporting network infrastructure. To complete this exercise, you'll need to know the name of a Secure Shell (SSH) key pair in the region you're using.

1. Visit http://awscsa.github.io and click Chapter 14. Download the `web-stack.json` and `network-stack.json` CloudFormation templates.



2. Create an S3 bucket and upload the `network-stack.json` template to the bucket.
3. Edit the `web-stack.json` template and locate the `Network-Stack` resource. Change the `TemplateURL` value to the URL of the `network-stack.json` template in S3.

```
        }
    },
    "Resources": {
        "NetworkStack" : {
            "Type" : "AWS::CloudFormation::Stack",
            "Properties" : {
                "TemplateURL" : "https://monarch-source-bucket.s3.us-east-1.amazonaws.com/network-stack.json"
            }
        },
```

4. Upload the `web-stack.json` template to the same S3 bucket you used in step 2.
5. CloudFormation will begin to create the `Web` stack and the nested network stack.

   Enter the following command to view the stacks:

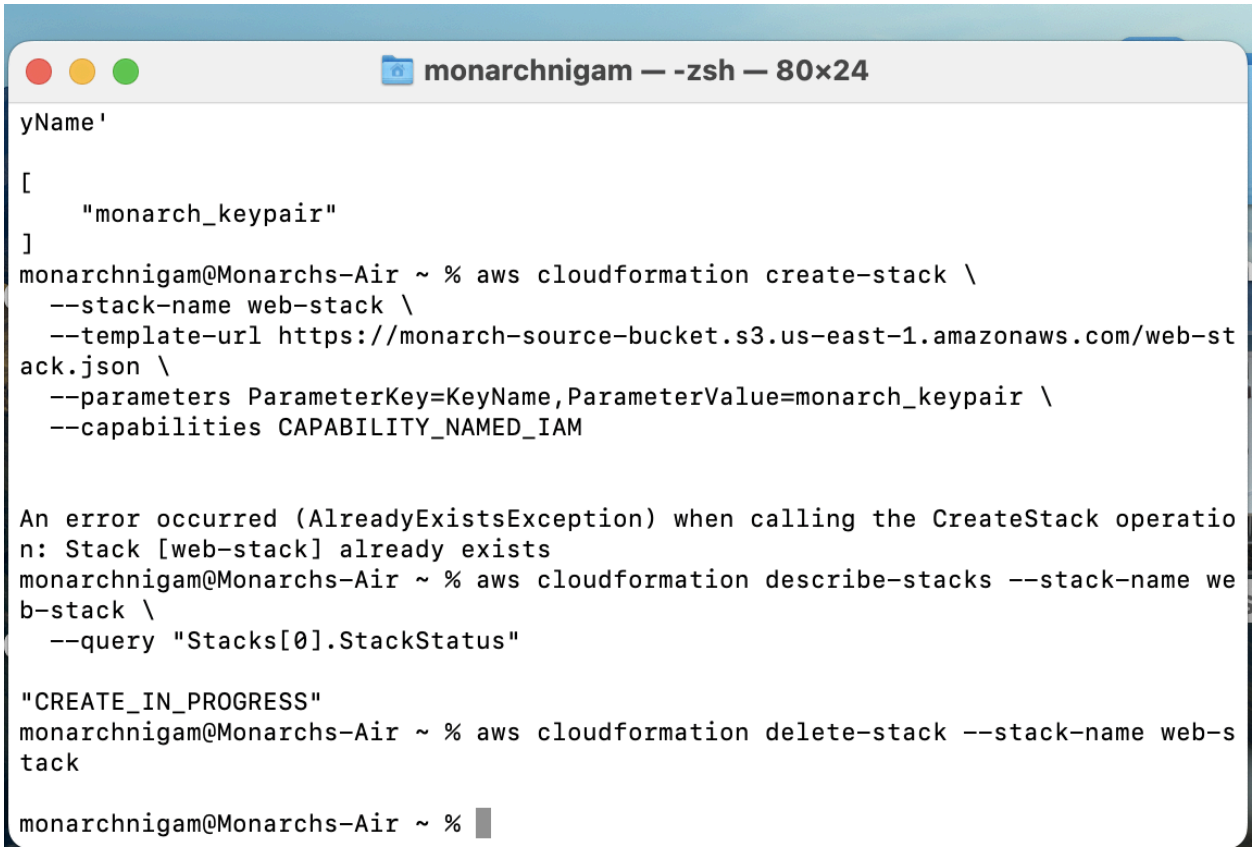
```
monarchnigam — -zsh — 80×24
monarchnigam@Monarchs-Air ~ % aws ec2 describe-key-pairs --query 'KeyPairs[*].Ke
yName'

[
    "monarch_keypair"
]
monarchnigam@Monarchs-Air ~ % aws cloudformation create-stack \
  --stack-name web-stack \
  --template-url https://monarch-source-bucket.s3.us-east-1.amazonaws.com/web-st
ack.json \
  --parameters ParameterKey=KeyName,ParameterValue=monarch_keypair \
  --capabilities CAPABILITY_NAMED_IAM


An error occurred (AlreadyExistsException) when calling the CreateStack operatio
n: Stack [web-stack] already exists
monarchnigam@Monarchs-Air ~ % aws cloudformation describe-stacks --stack-name we
b-stack \
  --query "Stacks[0].StackStatus"

"CREATE_IN_PROGRESS"
monarchnigam@Monarchs-Air ~ %
```

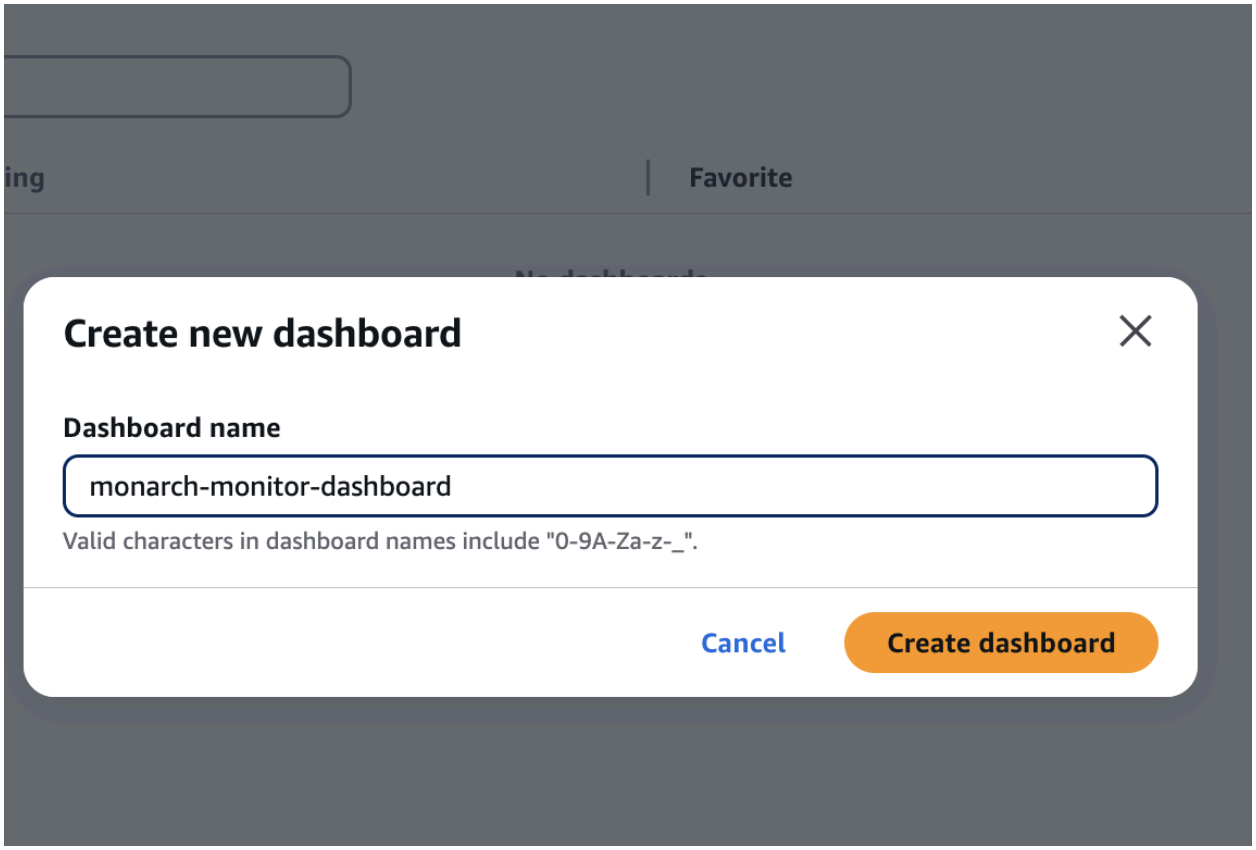
6. CloudFormation will begin to create the `Web` stack and the nested network stack. Enter the following command to view the stacks:



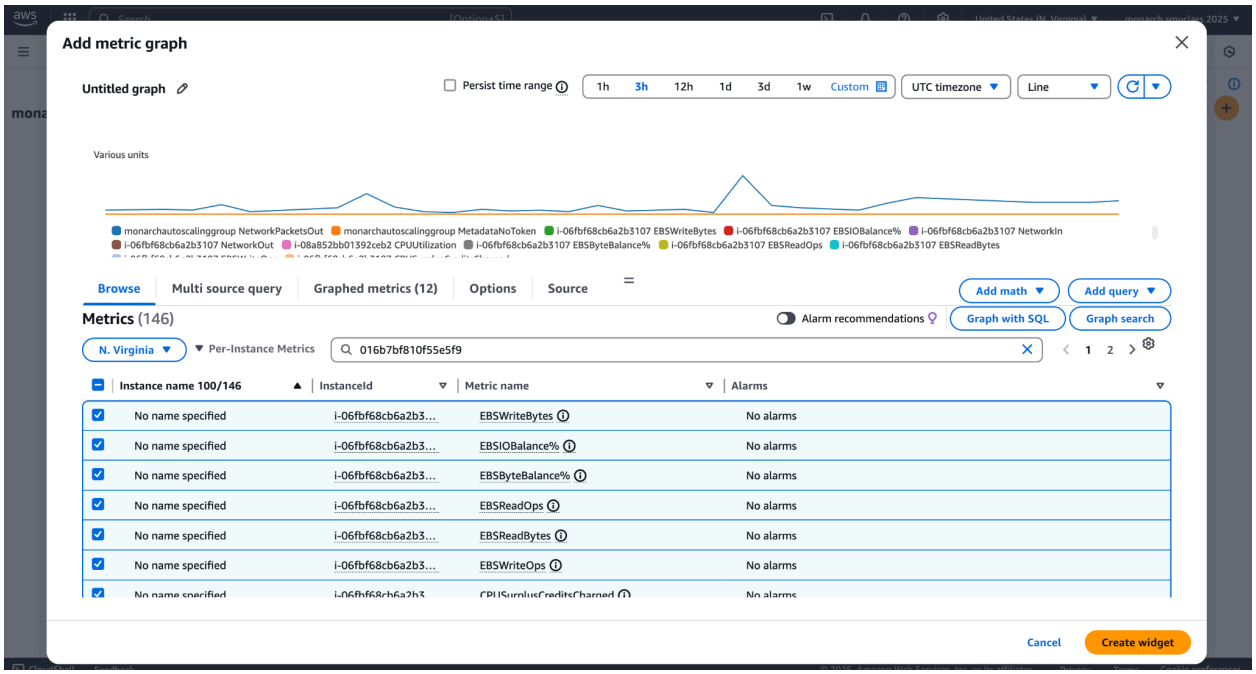
EXERCISE 11.6

Create a CloudWatch Dashboard

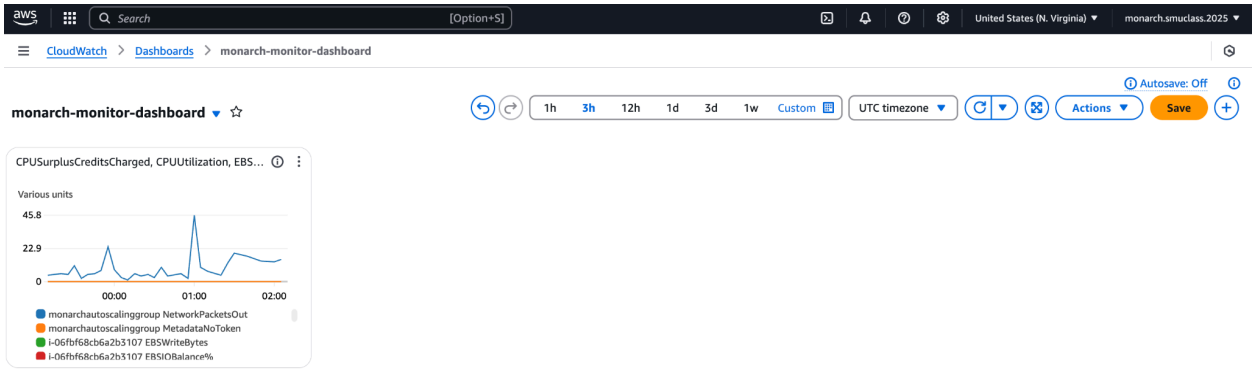
1. So that you'll get a better idea of how things will look, make sure you have at least one EC2 instance running and at least one S3 bucket with some files in it.
2. From the CloudWatch console, click Dashboards, click Create Dashboard, and give your dashboard a name. On the Add Widget page, select the Line metric. Then select the Metrics option.



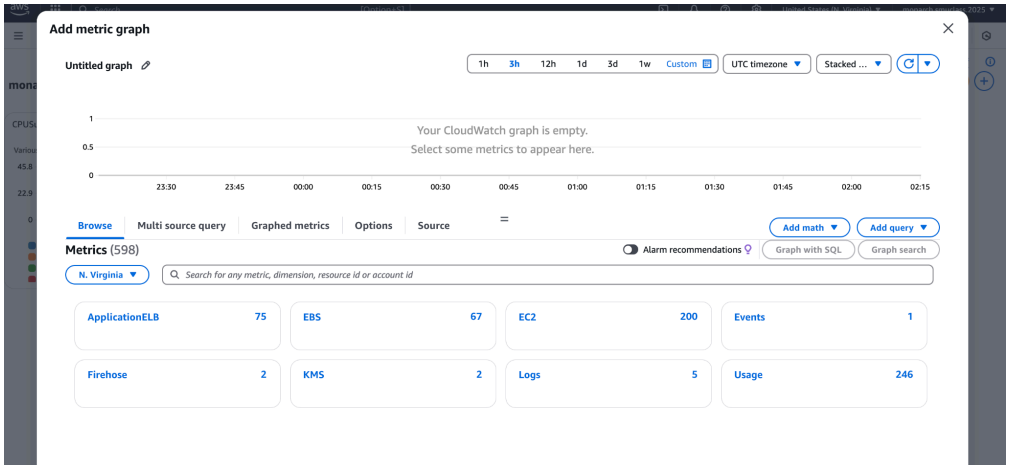
3. From the Metrics displayed in the bottom half of the screen, select the appropriate AWS region and click EC2 and then click Per-Instance Metrics. Check the boxes for a few metrics associated with your running instance. StatusCheckFailed, NetworkOut, NetworkIn, and CPUUtilization would make for a nice sample. Note how those metrics are now represented in the graph at the top of the screen. When you're satisfied with your selections, click Create Widget.



4. Your first widget will now appear in your dashboard. You can click the widget to display it full-screen. Selecting a particular metric from the legend at the bottom of the graph will show you only that metric, making it easier to understand what the values represent.



5. If your widget is still full-screen, click Close to return to the dashboard.
6. Click Add Widget to create a second widget. This time, select Stacked Area and then Storage Metrics. Select S3 metrics and then the NumberOfObjects and BucketSizeBytes metrics for your bucket. Create the widget.



In the above screenshot we cant find S3 although the S3 bucket is created with the stack files uploaded in the last exercise. Hence we will create another dashboard with services other than S3, which are visible in the above screenshot

7. Once you've added all the widgets and metrics you'll need to monitor, click Save Dashboard.