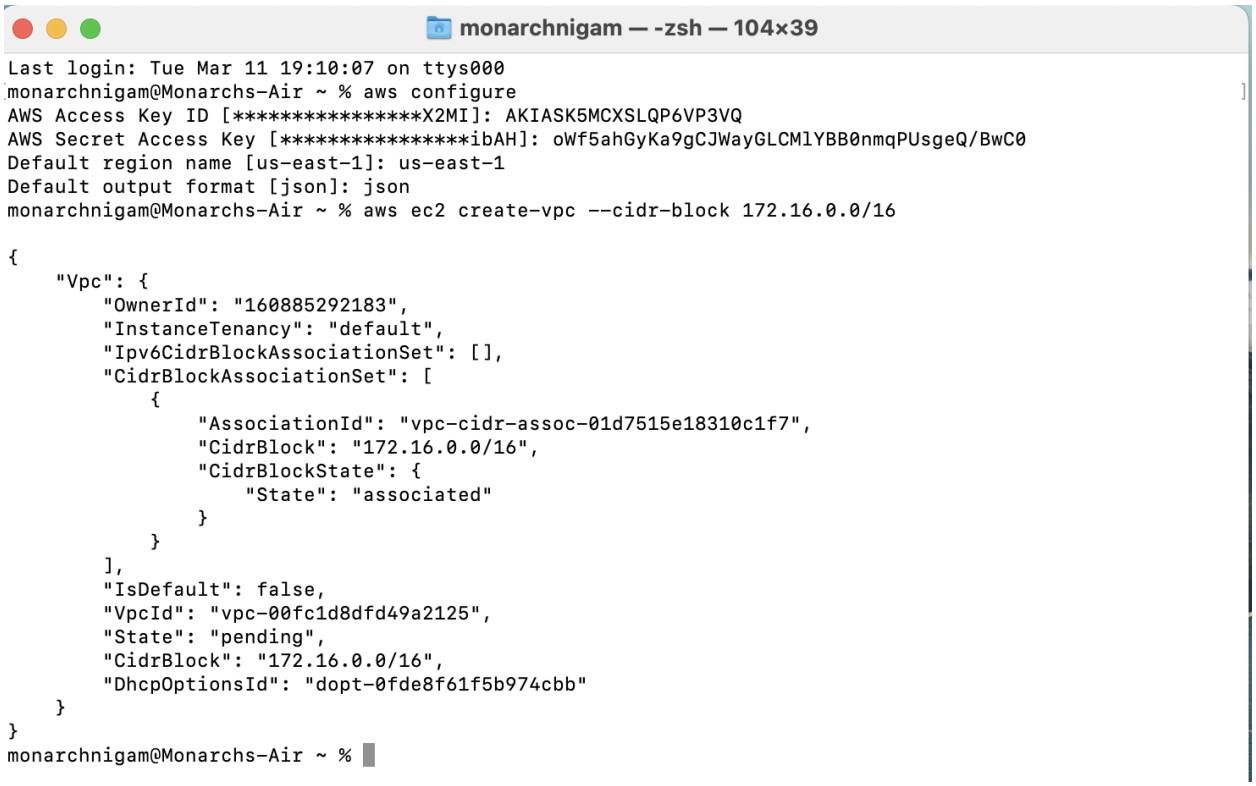
AWS LAB | Monarch Nigam

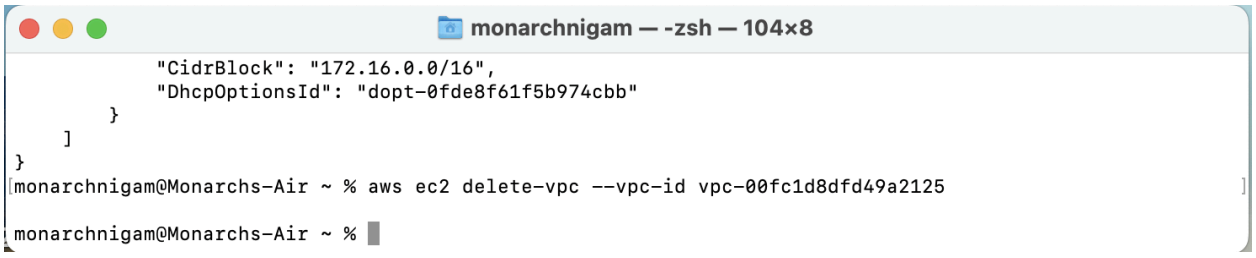
EXERCISE 4.1 To complete Exercise 4.1: Create a New VPC using AWS CLI,

1. Create a New VPC Create a VPC with the primary CIDR 172.16.0.0/16.



2. Note that the VPC is initially in the pending state as AWS creates it.

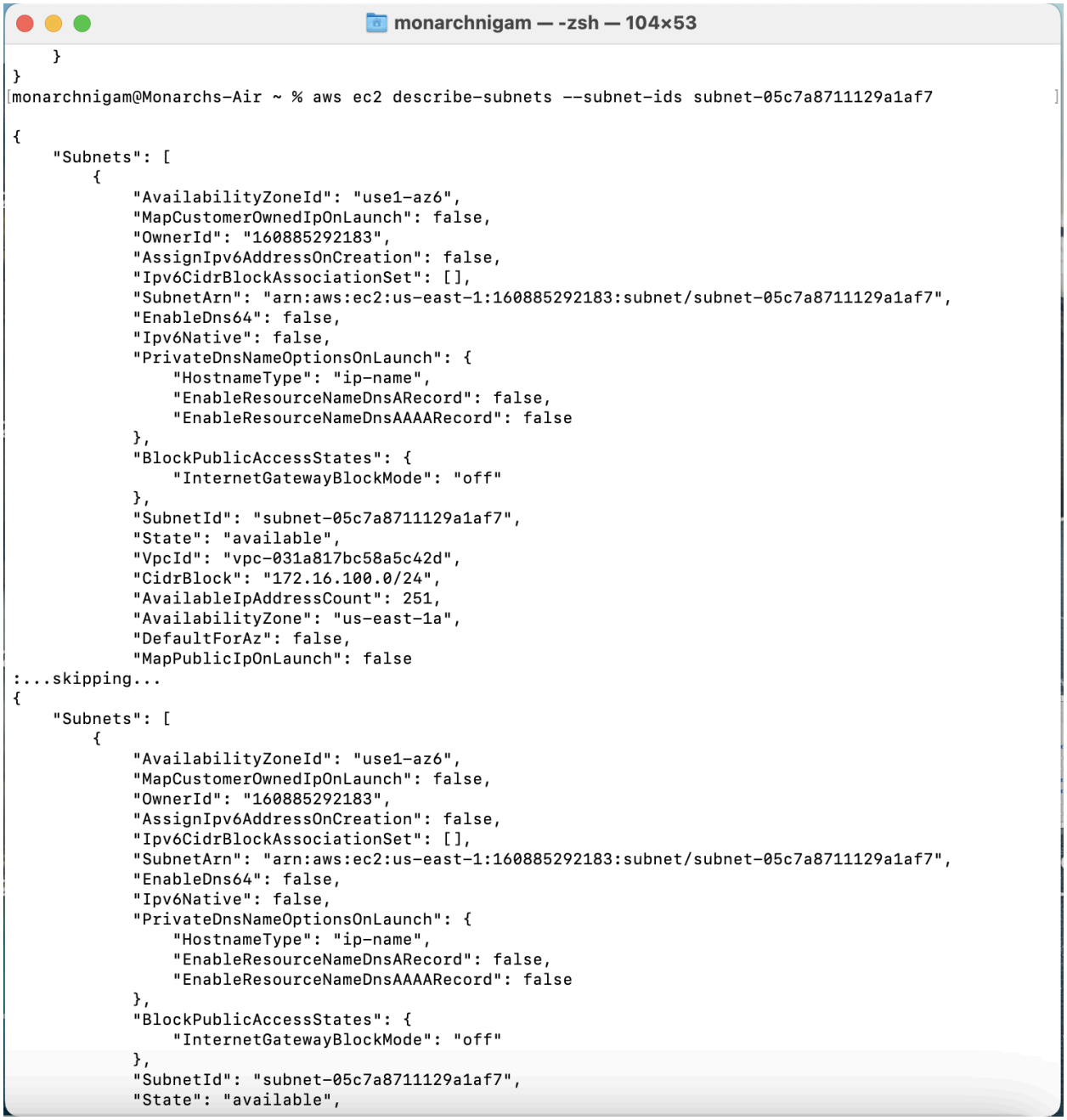
3. You can delete an unneeded VPC using code like this:



EXERCISE 4.2 Create a New Subnet

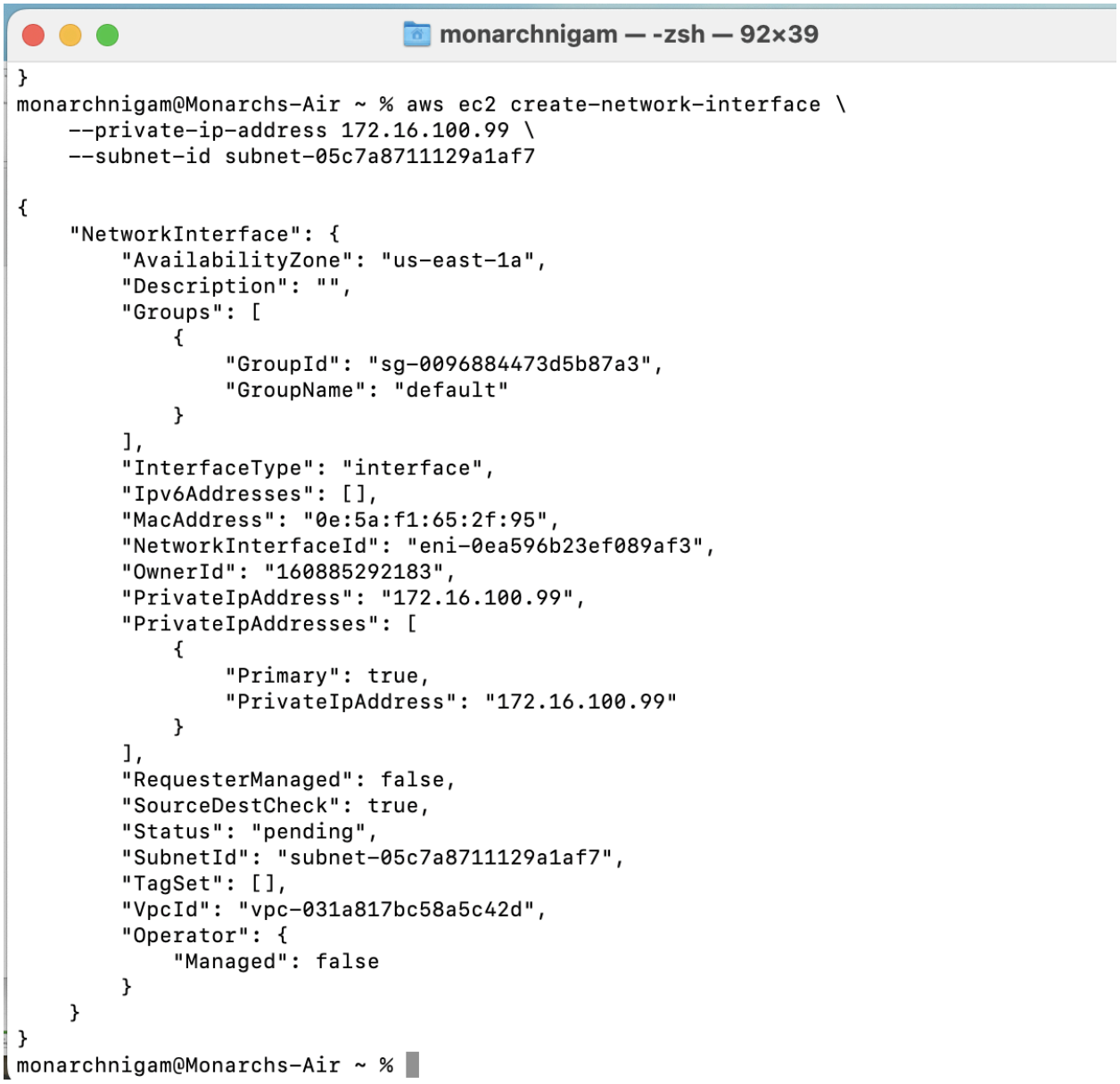
1. **Create a subnet in the VPC you created earlier. Choose an availability zone and assign the CIDR block 172.16.100.0/24.**

2. **Wait a few moments, and then check the status of the subnet as follows: aws ec2 describe-subnets --subnet-ids [subnet-id]**



EXERCISE 4.3 Create and Attach a Primary ENI

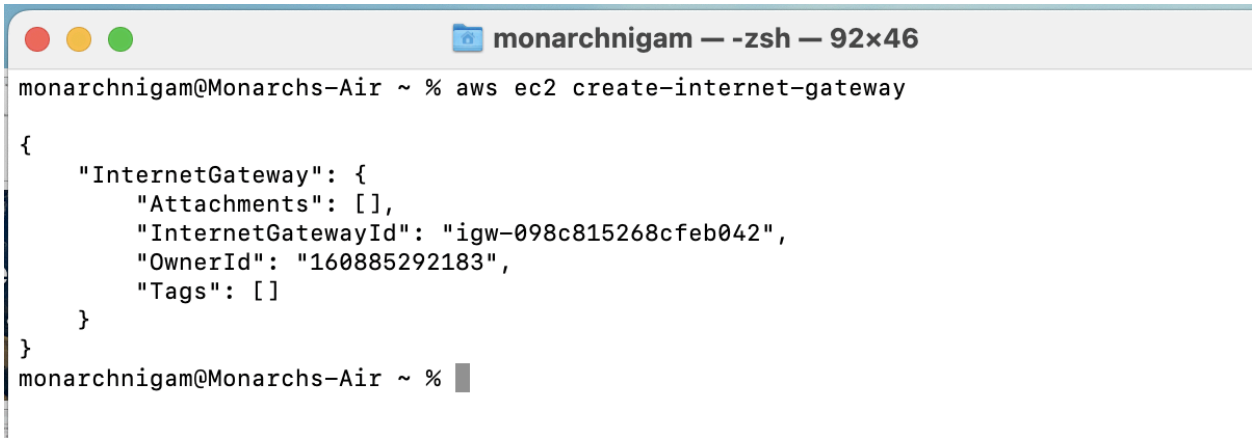
1. **Create an ENI in the subnet of your choice.**



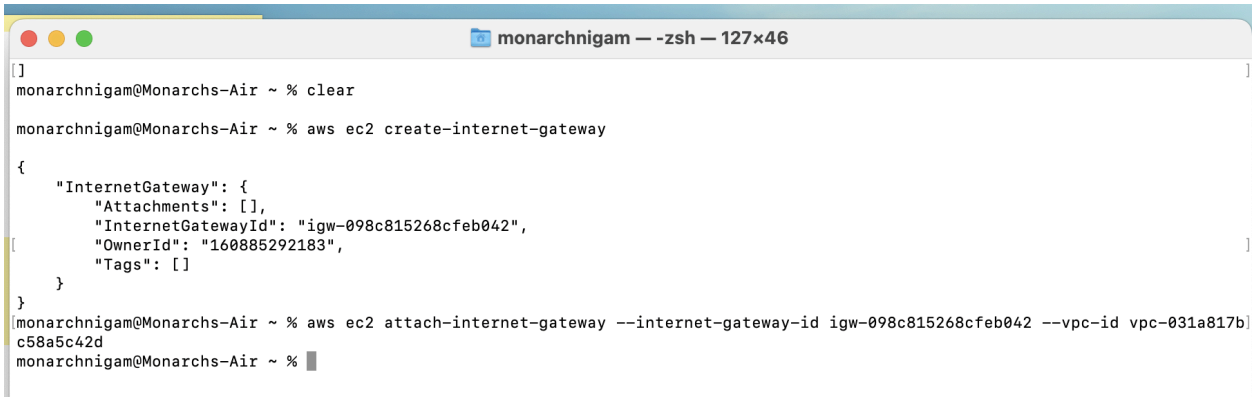
2. **Verify the Network Interface Use the NetworkInterfaceId from the output to check its status:**

EXERCISE 4.4 Create an Internet Gateway and Default Route In this exercise, you'll create an Internet gateway and attach it to the VPC you used in the previous exercises.

1. **Create an Internet gateway using the following command: aws ec2 create-internet-gateway**

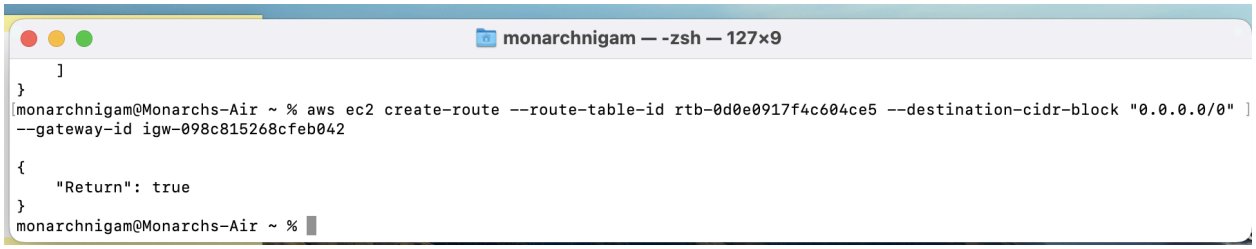


2. **Attach the Internet gateway to the VPC you used in the previous exercises using the following command (a successful execution will generate no output):**



3. **Retrieve the route table ID of the main route table for the VPC: aws ec2 describe-route-tables-filters Name=vpc-id,Values=[vpc-id]**

4. **To create a default route in the main route table, issue the following command: aws ec2 create-route --route-table-id rtb-097d8be97649e0584 --destination-cidr-block "0.0.0.0/0" --gateway-id igw-0312f81aa1ef24715**

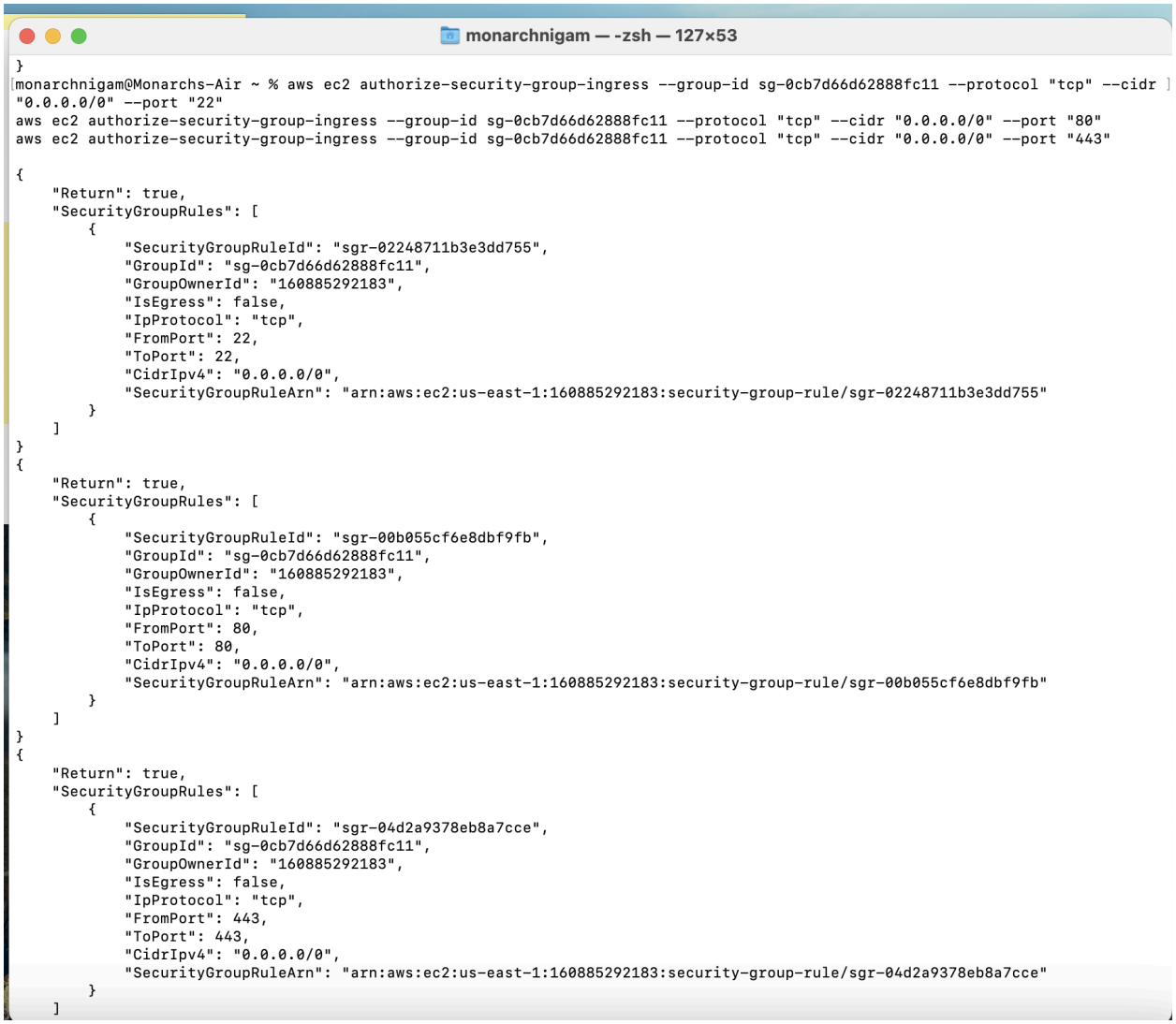


5. **Verify the Route Table**

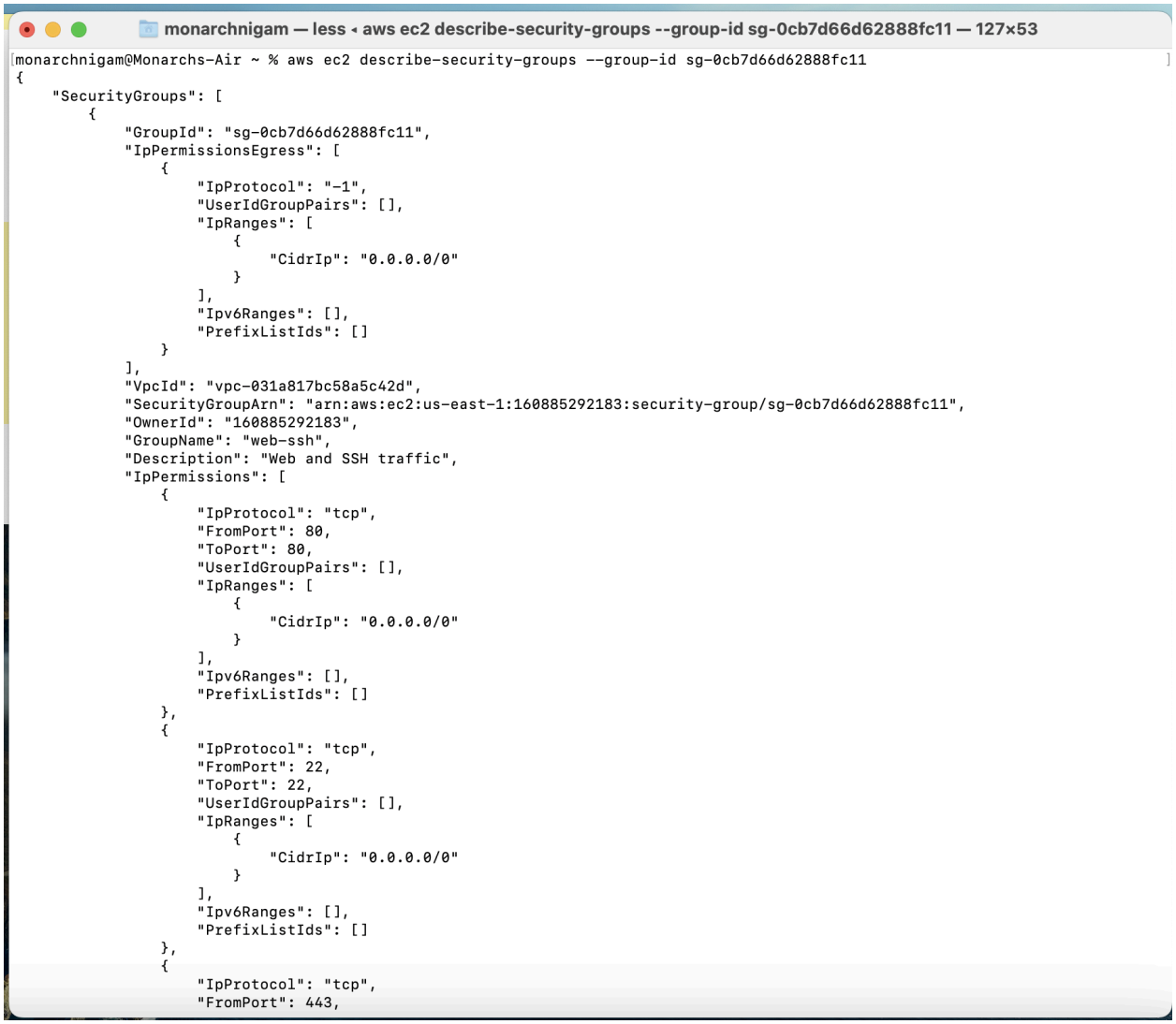
EXERCISE 4.5 Create a Custom Security Group In this exercise, you'll create a new security group that allows SSH, HTTP, and HTTPS access from any IP address.

1. **Create a security group named web-ssh in the VPC you created previously: Make a note of the security group ID in the output:**

2. **Using the security group ID, create three rules to allow SSH, HTTP, and HTTPS access from any IP address.**



3. **Verify the rules by viewing the security group: aws ec2 describe-security-groups**

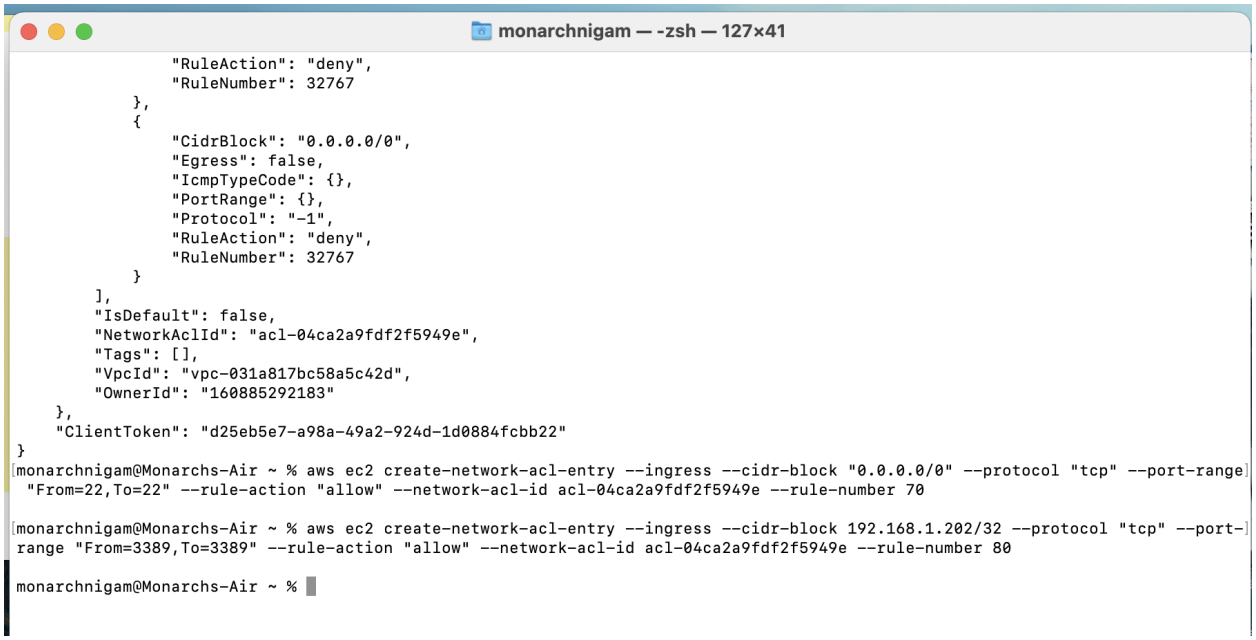


EXERCISE 4.6 Create an Inbound Rule to Allow Remote Access from Any IP Address NACL rule order matters! Create a new NACL and attach it to the subnet you used in Exercise 4.3.

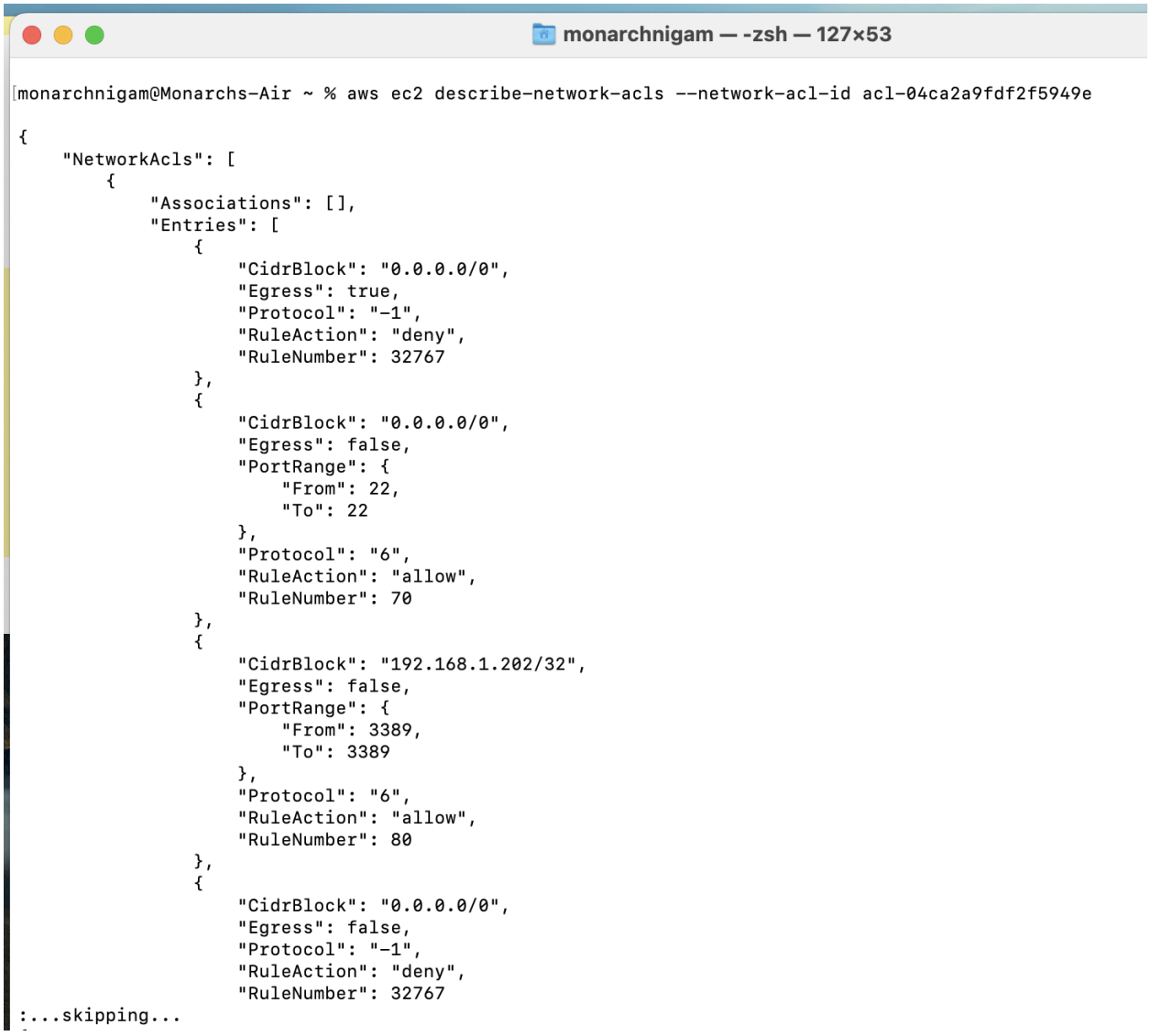
1. Create a new network ACL using the following command: aws ec2 create-network-acl --vpc-id [vpc-id]

2. Create an inbound rule entry to allow SSH (TCP port 22) access from any IP address. We'll use the rule number 70.

3. Modify the command from step 2 to create rule number 80 that allows RDP (TCP port 3389) access from your public IP address (use https://ifconfig.co to find it if you don't already know it). Use a /32 prefix length.

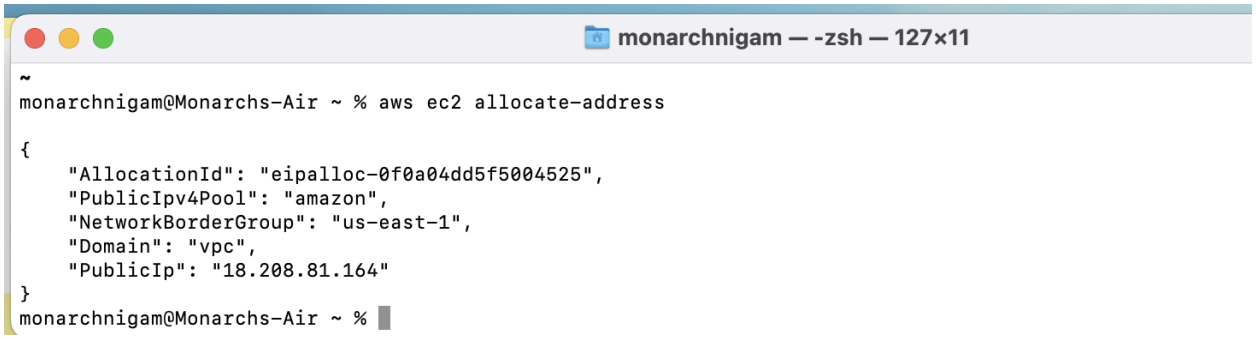


4. Use the following command to verify that the rules made it into your NACL: aws ec2 describe-network-acls --network-acl-id [network-acl-id]

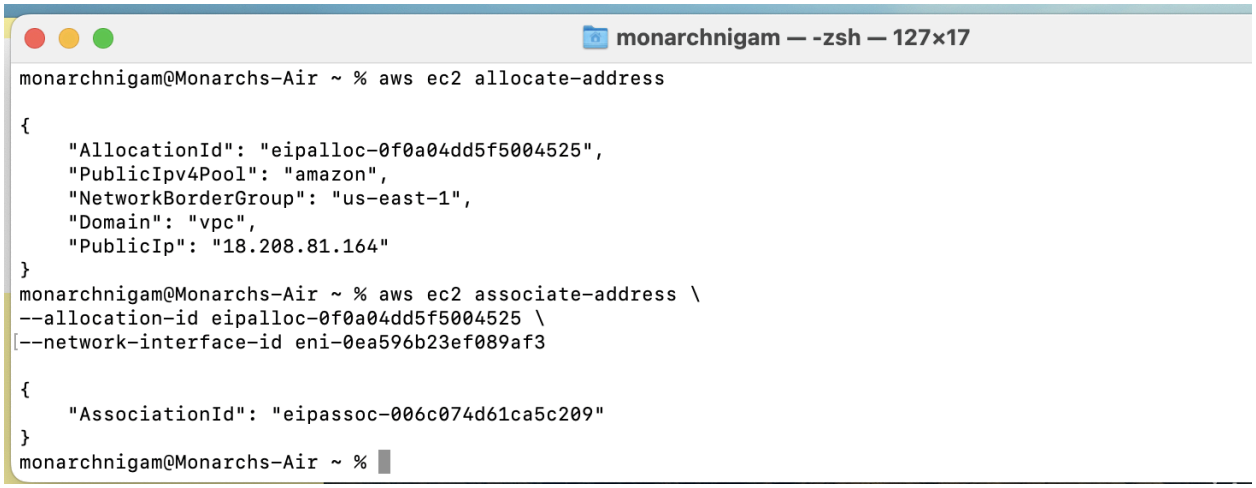


EXERCISE 4.7 Allocate and Use an Elastic IP Address Allocate an elastic IP address and associate it with the instance you created earlier.

1. Allocate an EIP using the following command: aws ec2 allocate-address



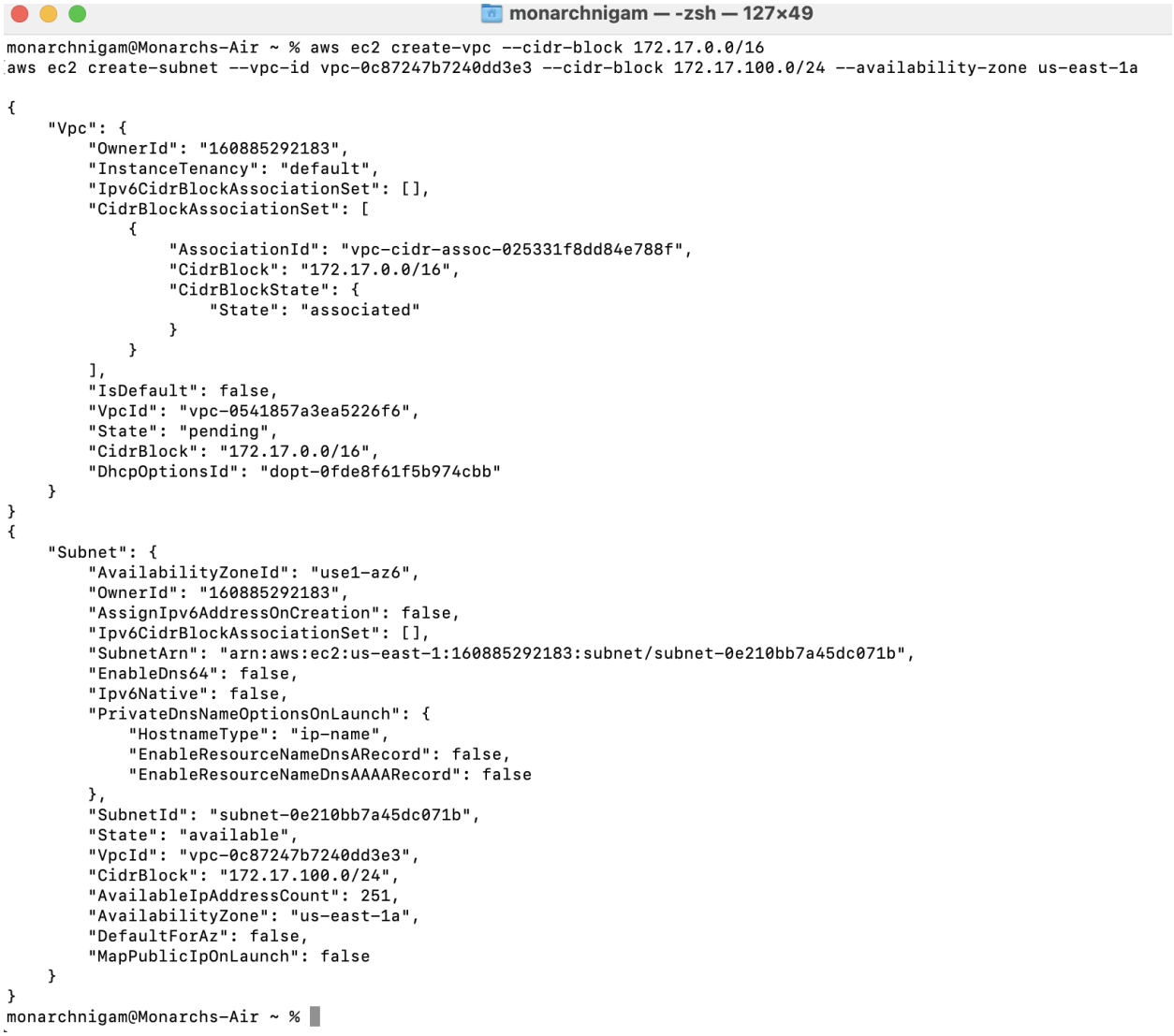
2. Associate the EIP to the elastic network interface you created earlier:



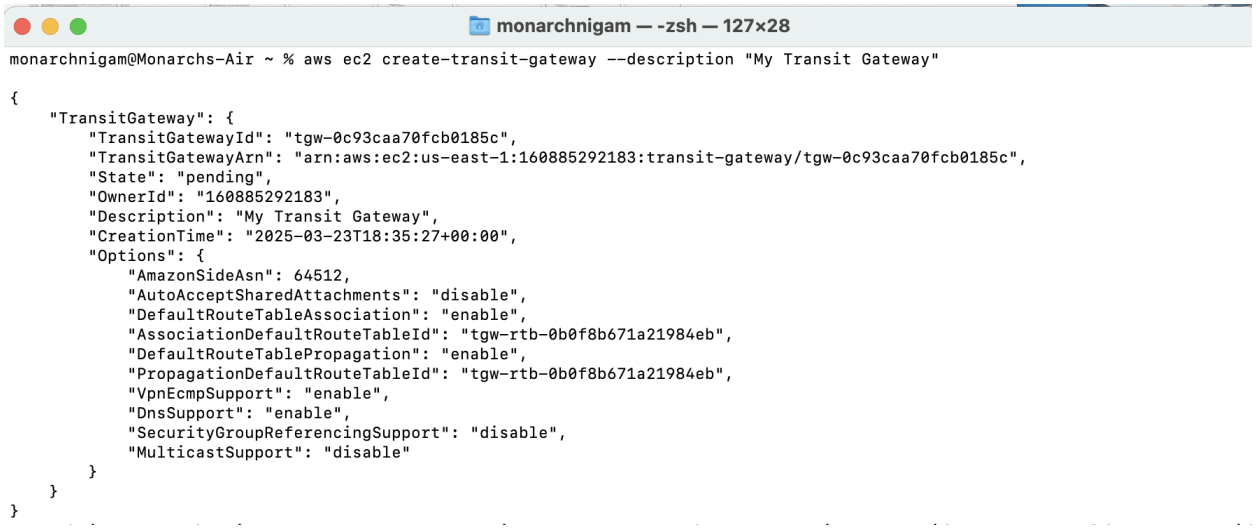
3. Verify that the EIP is associated with the elastic network interface: aws ec2 describe-network-interfaces --network-interface-ids eni-0863f88f670e8ea06

EXERCISE 4.8 Create a Transit Gateway In this exercise, you'll create another VPC and subnet. You'll then create a transit gateway and attach it to both VPCs. Finally, you'll configure the transit gateway to route between the VPC subnets.

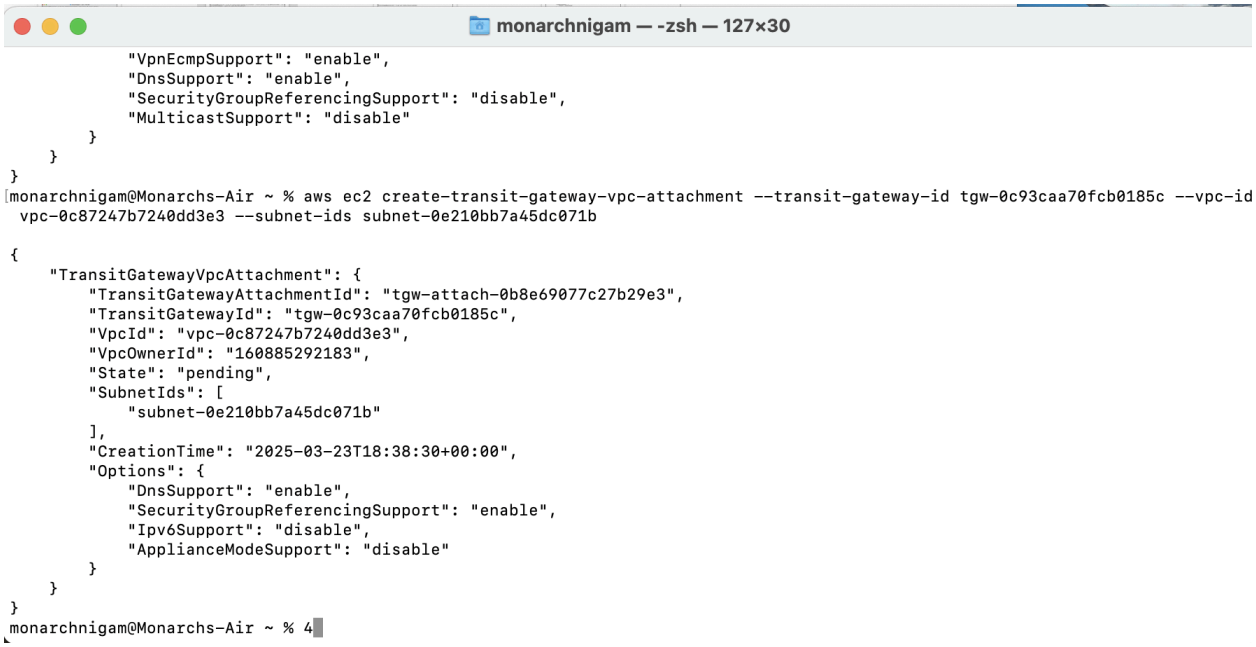
1. Create a new VPC and subnet:



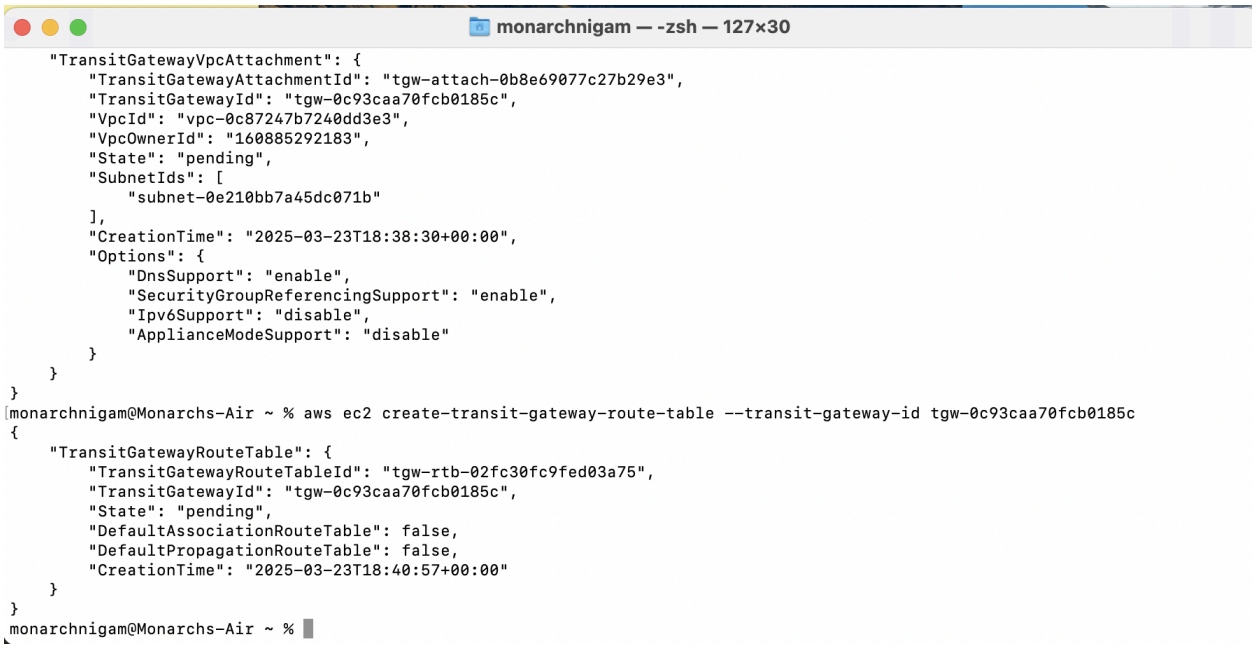
2. Create a transit gateway:



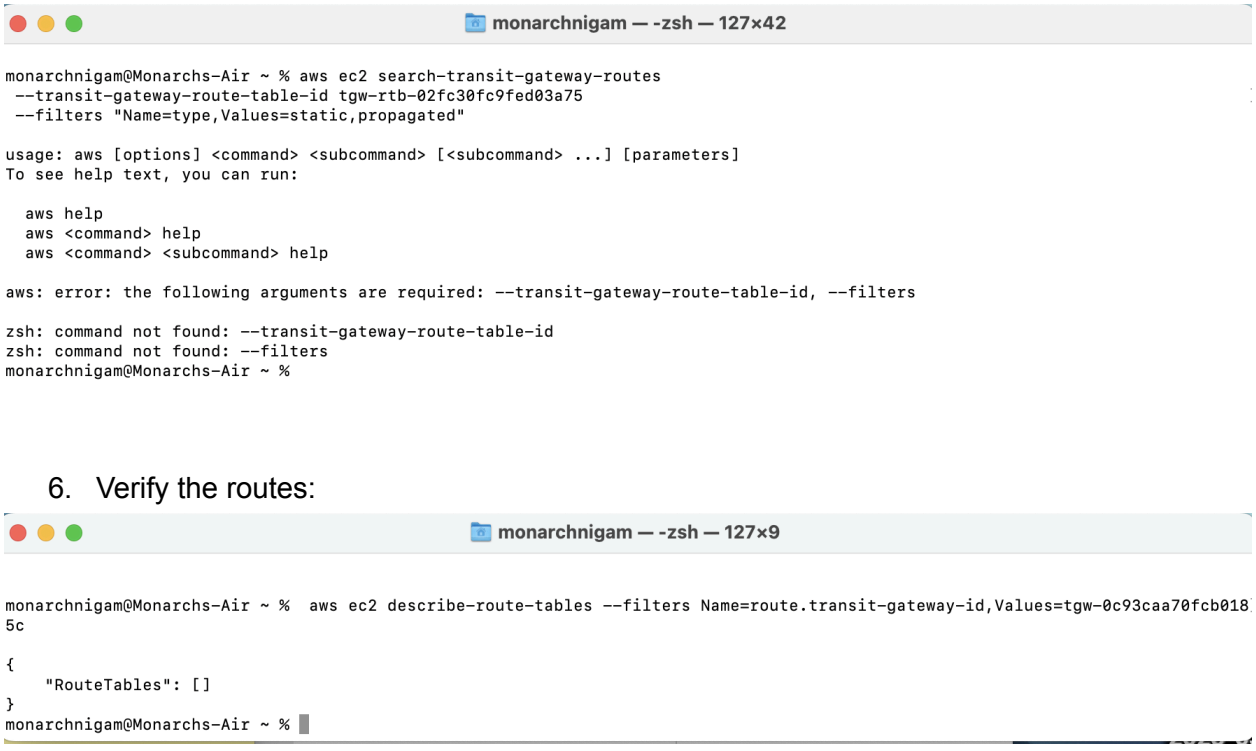
3. Attach the transit gateway to both VPCs:



4. View the transit gateway routes:



5. Create routes in each subnet's route table:

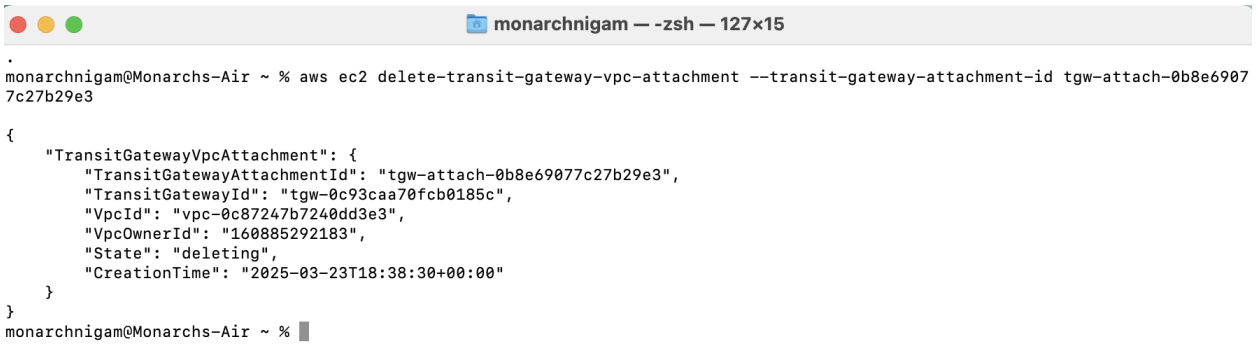


6. Verify the routes:

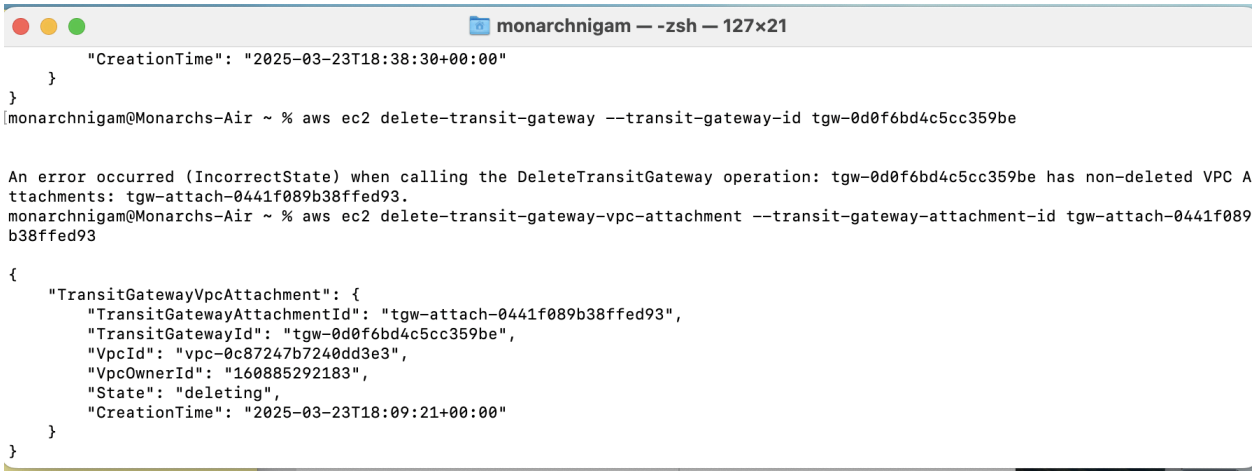
EXERCISE 4.9 Create a Blackhole Route In this exercise, you'll create a blackhole route for the range of addresses 172.16.100.64–172.16.100.71. You'll then delete your transit gateway.

1. **Create a Blackhole Route:** Run this command to create a blackhole route for the 172.16.100.64/29 subnet:

2. **List Transit Gateway Routes:** Get the transit gateway attachment IDs using:



3. **Delete Transit Gateway Attachments:** Delete the attachments using the following commands:

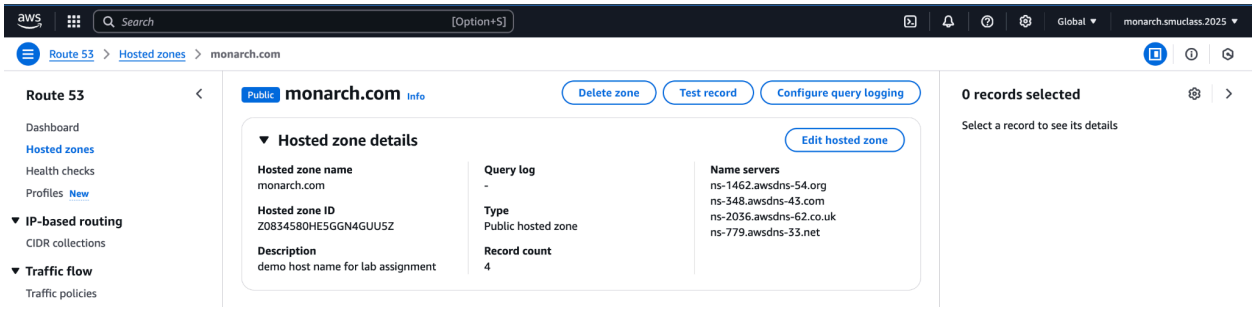


EXERCISE 8.1

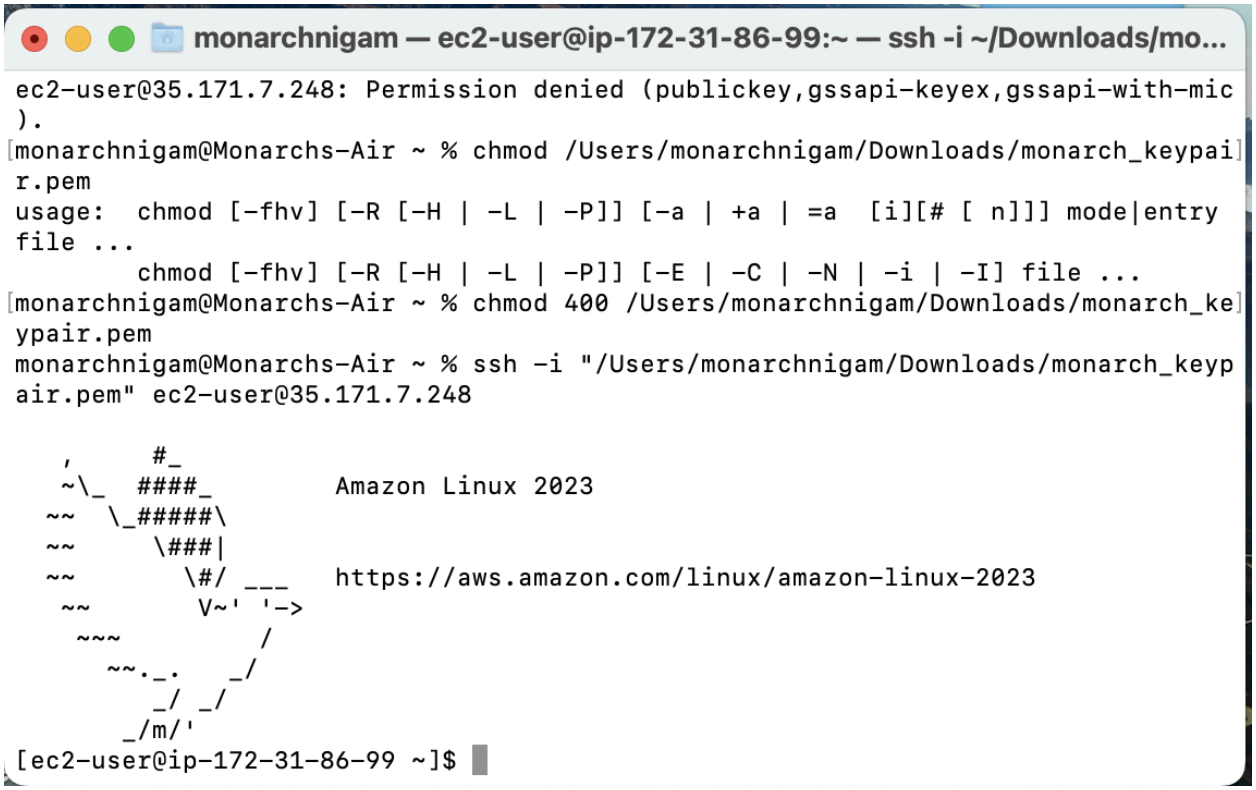
Create a Hosted Zone on Route 53 for an EC2 Web Server

1. Click the Hosted Zones link and then the Create Hosted Zone button in the Route 53 dashboard.

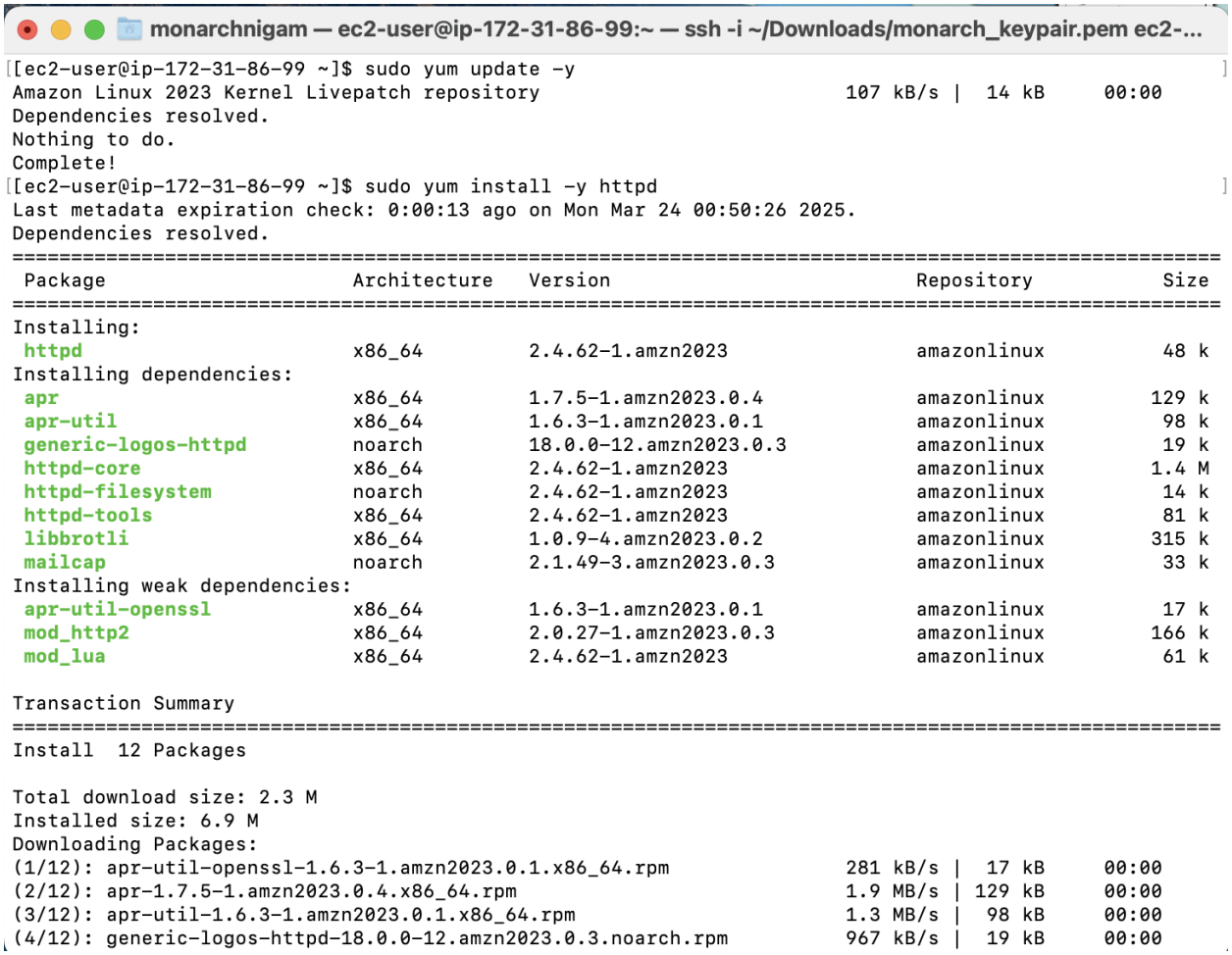
2. Enter a valid domain name that's currently not managed and select the Public Hosted Zone type. If you don't already have an existing domain on Route 53 and don't want to create a new one just for this exercise, use a fake domain—even something like example.com.

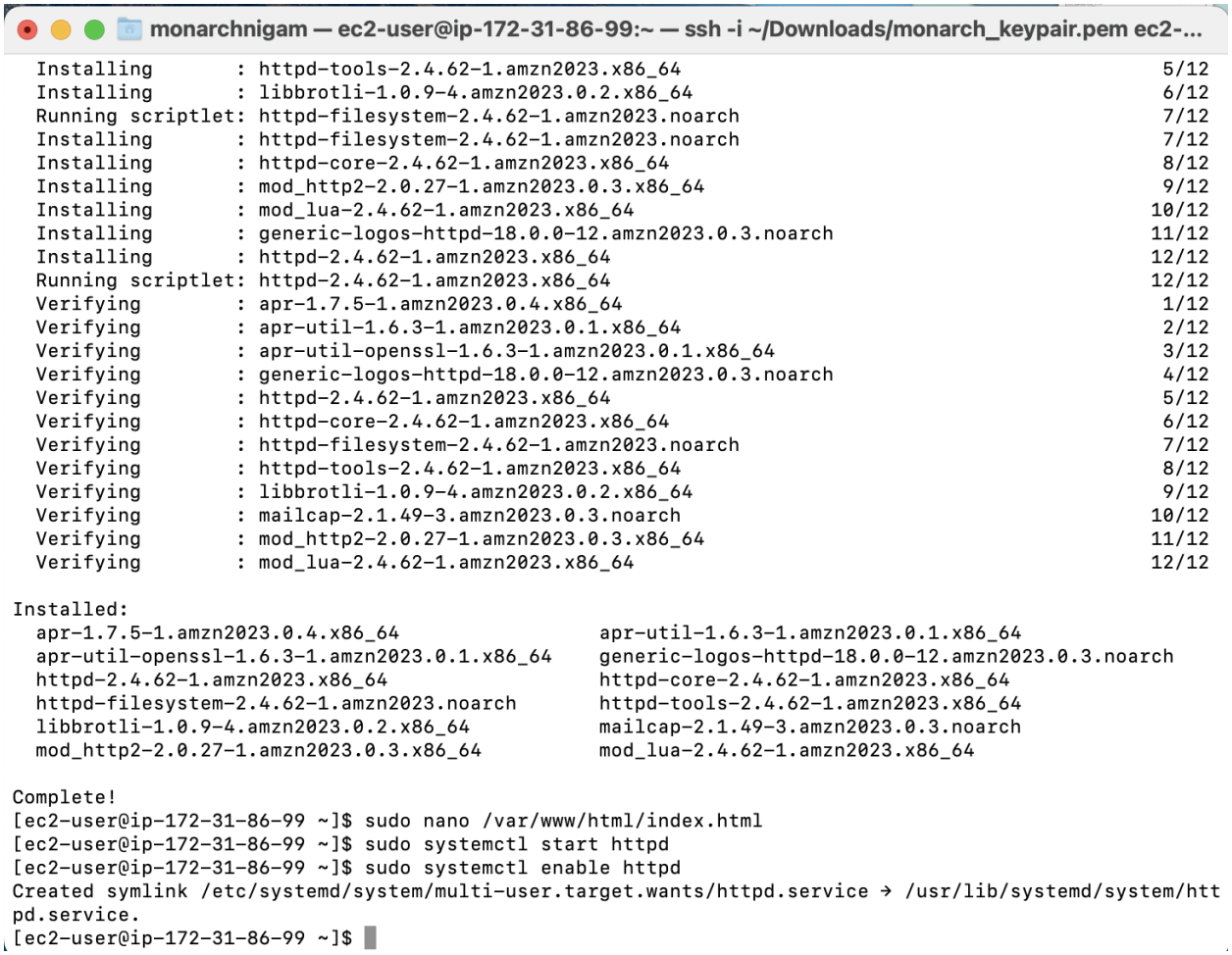


3. You'll be given name server and SOA record sets. Select those records one at a time and spend a few minutes exploring.
4. From the Amazon EC2 Dashboard, create an AWS Linux AMI-based instance the way you did in Exercise 2.1 in Chapter 2, “Compute Services.” You'll configure this instance as a simple web server so that you can test the domain. When configuring your security group, confirm that port 22 (SSH) is open and open HTTP port 80 (and HTTPS port 443 if necessary) so that web traffic can get in.



Log into your instance (as you did in Exercise 2.1) and run the following commands to update the software repository, install the Apache web server, create a default index.html web page

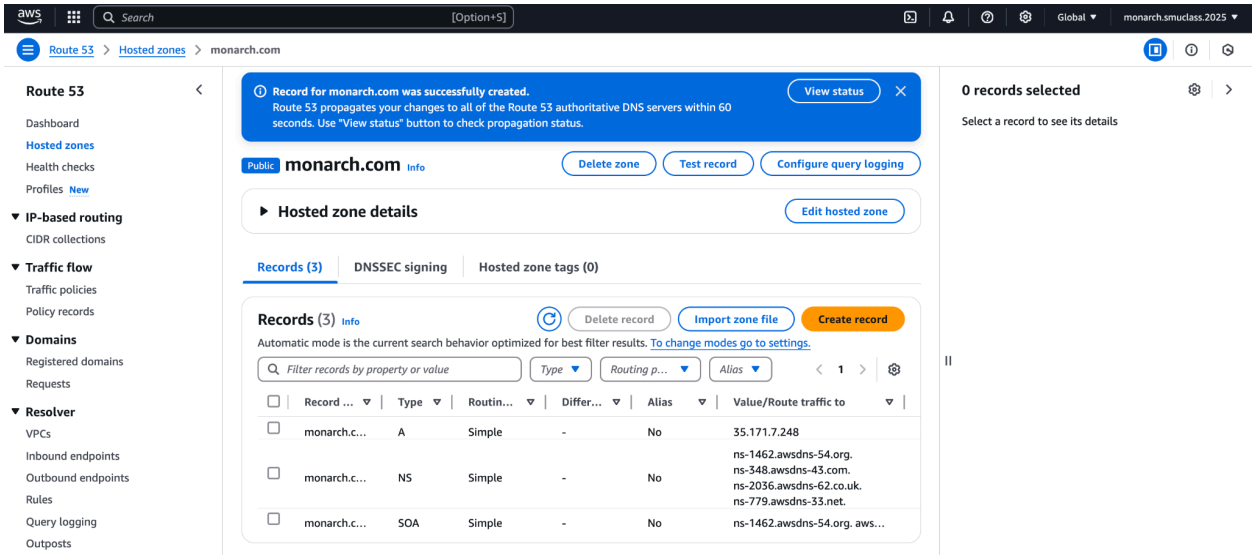

monarchnigam — ec2-user@ip-172-31-86-99:~ — ssh -i ~/Downloads/monarch_keypair.pem ec2-...
[ec2-user@ip-172-31-86-99 ~]$ sudo yum update -y
Amazon Linux 2023 Kernel Livepatch repository 107 kB/s | 14 kB 00:00
Dependencies resolved.
Nothing to do.
Complete!
[ec2-user@ip-172-31-86-99 ~]$ sudo yum install -y httpd
Last metadata expiration check: 0:00:13 ago on Mon Mar 24 00:50:26 2025.
Dependencies resolved.
Package Architecture Version Repository Size
Installing:
httpd x86_64 2.4.62-1.amzn2023 amazonlinux 48 k
Installing dependencies:
apr x86_64 1.7.5-1.amzn2023.0.4 amazonlinux 129 k
apr-util x86_64 1.6.3-1.amzn2023.0.1 amazonlinux 98 k
generic-logos-httpd noarch 18.0.0-12.amzn2023.0.3 amazonlinux 19 k
httpd-core x86_64 2.4.62-1.amzn2023 amazonlinux 1.4 M
httpd-filesystem noarch 2.4.62-1.amzn2023 amazonlinux 14 k
httpd-tools x86_64 2.4.62-1.amzn2023 amazonlinux 81 k
libbrotli x86_64 1.0.9-4.amzn2023.0.2 amazonlinux 315 k
mailcap noarch 2.1.49-3.amzn2023.0.3 amazonlinux 33 k
Installing weak dependencies:
apr-util-openssl x86_64 1.6.3-1.amzn2023.0.1 amazonlinux 17 k
mod_http2 x86_64 2.0.27-1.amzn2023.0.3 amazonlinux 166 k
mod_lua x86_64 2.4.62-1.amzn2023 amazonlinux 61 k
Transaction Summary
Install 12 Packages
Total download size: 2.3 M
Installed size: 6.9 M
Downloading Packages:
(1/12): apr-util-openssl-1.6.3-1.amzn2023.0.1.x86_64.rpm 281 kB/s | 17 kB 00:00
(2/12): apr-1.7.5-1.amzn2023.0.4.x86_64.rpm 1.9 MB/s | 129 kB 00:00
(3/12): apr-util-1.6.3-1.amzn2023.0.1.x86_64.rpm 1.3 MB/s | 98 kB 00:00
(4/12): generic-logos-httpd-18.0.0-12.amzn2023.0.3.noarch.rpm 967 kB/s | 19 kB 00:00



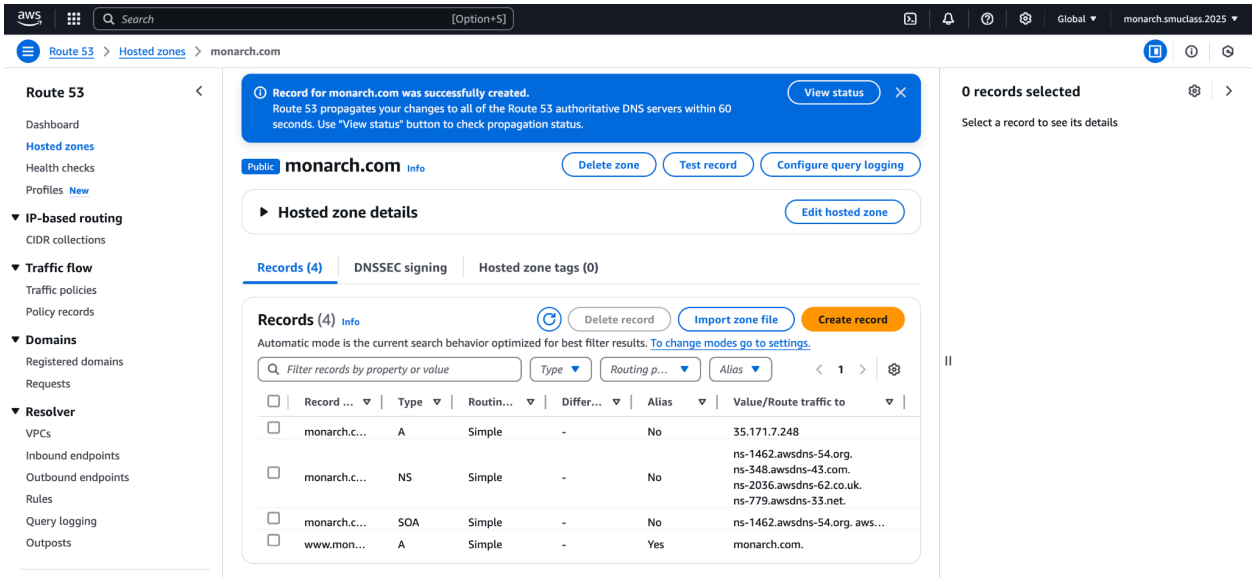
5. You can confirm your web server is working by pointing your browser to the instance's public IP address. You may want to keep this instance running, as it will also be helpful for the next exercise.

Welcome to Monarch AWS web server!

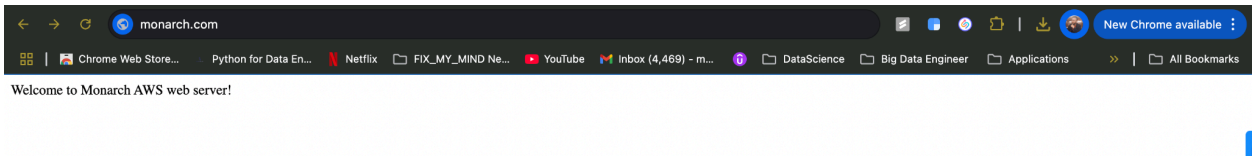
6. Back in the Route 53 Dashboard, click the Create Record button to create an A record to map your domain name to the IP address of a web server instance running on EC2. Leave the Name field to the left of the example.com value blank and enter your server's IP address (or any realistic fake address if your domain isn't real) in the Value field. Click Create.



7. Click Create Record Set once again to create a second A record. This time, you'll enter **www** in the Name field.
8. Click the Yes radio button next to Alias, click once inside the Alias Target field, and select the example.com value that should be among those that appear.



9. Domain propagation might take a few hours, but assuming you're working with an actual server, you should eventually be able to point your browser to example.com and your server's root web page should load.

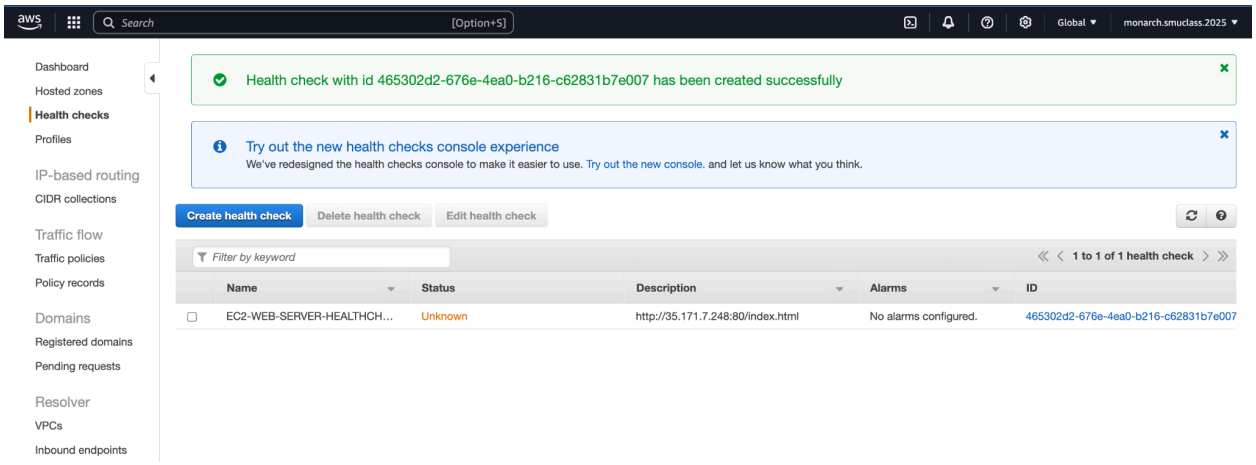


EXERCISE 8.2

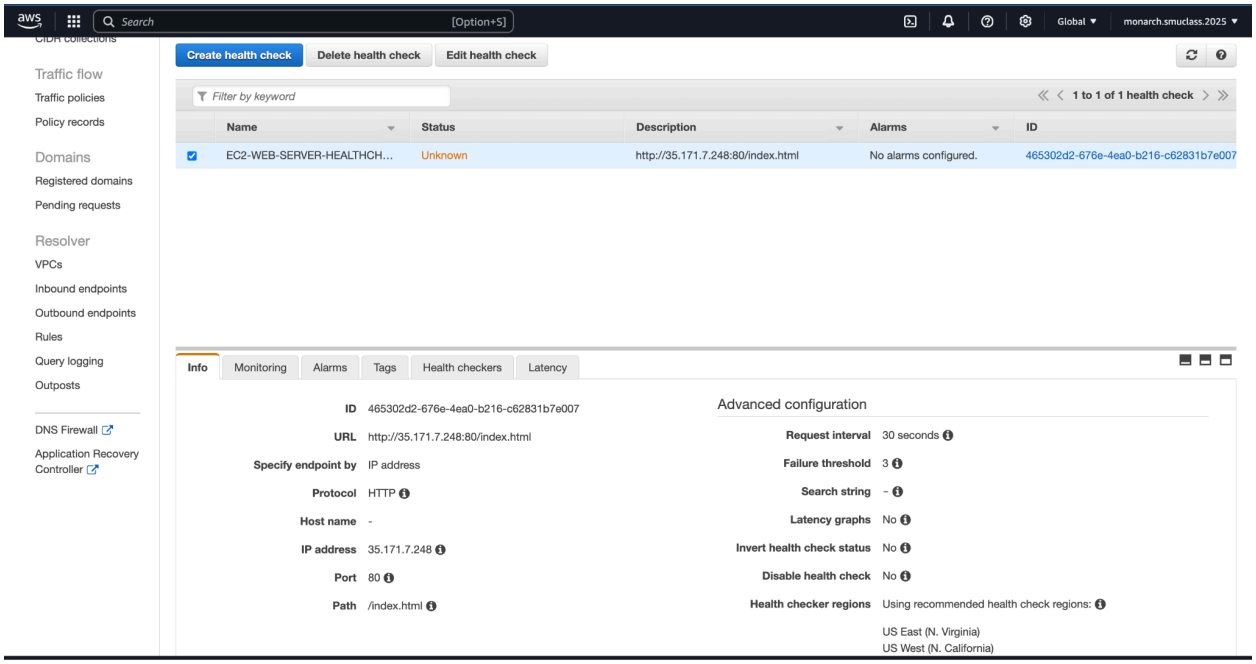
Set Up a Health Check

1. In the Route 53 GUI, click Create Health Check.
2. Give your check a name and select the Endpoint radio button for What To Monitor. You can use the web server you created in Exercise 8.1 as the health check endpoint.

3. With the IP address radio button selected for Specify Endpoint By, choose either HTTP or HTTPS for the protocol (this will depend on how you configured Transport Layer Security [TLS] encryption for your website—HTTP is a good bet if you're not sure).
4. Enter your server's IP address, and enter either **80** or **443** for the Port value (again, this will depend on your site's TLS settings—80 is a good bet if you're not sure).
5. Enter the name of a file in the web root directory of your website as the Path value. This is the resource the health check will try to load to test your site's health. You could use `index.html` (or `index.php`, if you happen to have a PHP application running), but, over the long term, that will probably cause a lot of unnecessary overhead. Instead, consider creating a small file called something like `test.html` and entering that for Path.



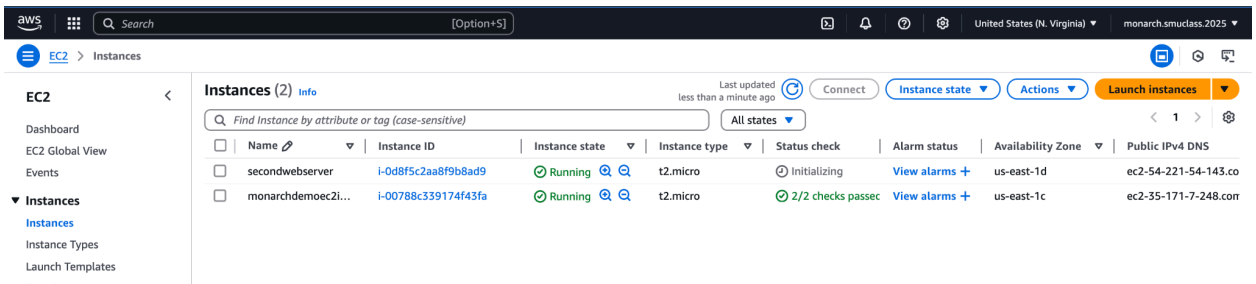
6. You can create an alarm to alert you to failed tests or simply click Create Health Check.

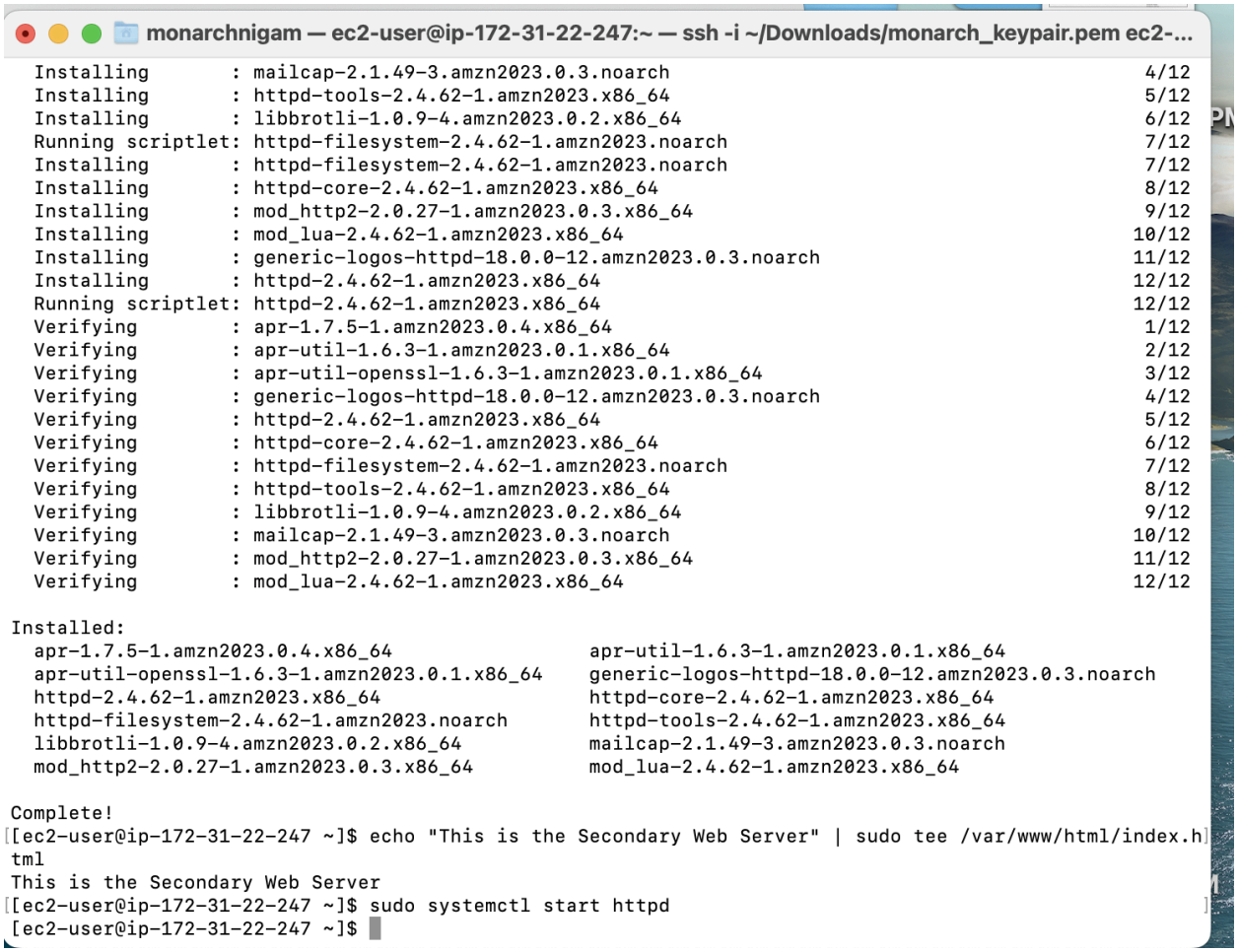


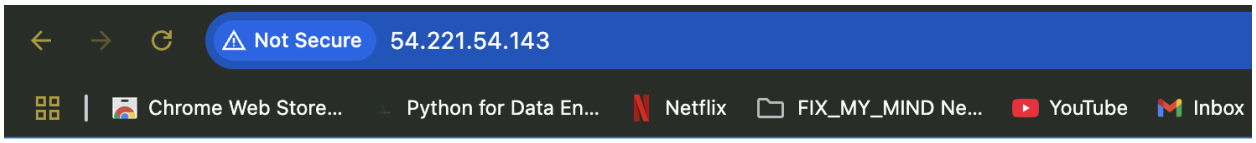
EXERCISE 8.3

Configure a Route 53 Routing Policy

1. Make sure you have two separate web-facing resources running. You could create a second Apache web server instance in addition to the one you launched in Exercise 8.1, or perhaps create a simple static website in S3 the way you did in Exercise 3.4 in Chapter 3, “AWS Storage.” In this case, though, your bucket name will have to exactly match the domain name.



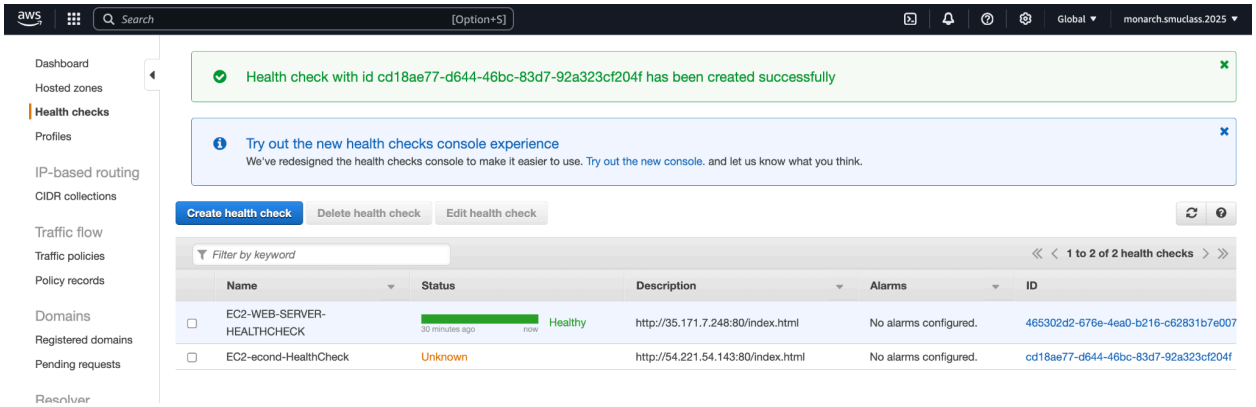




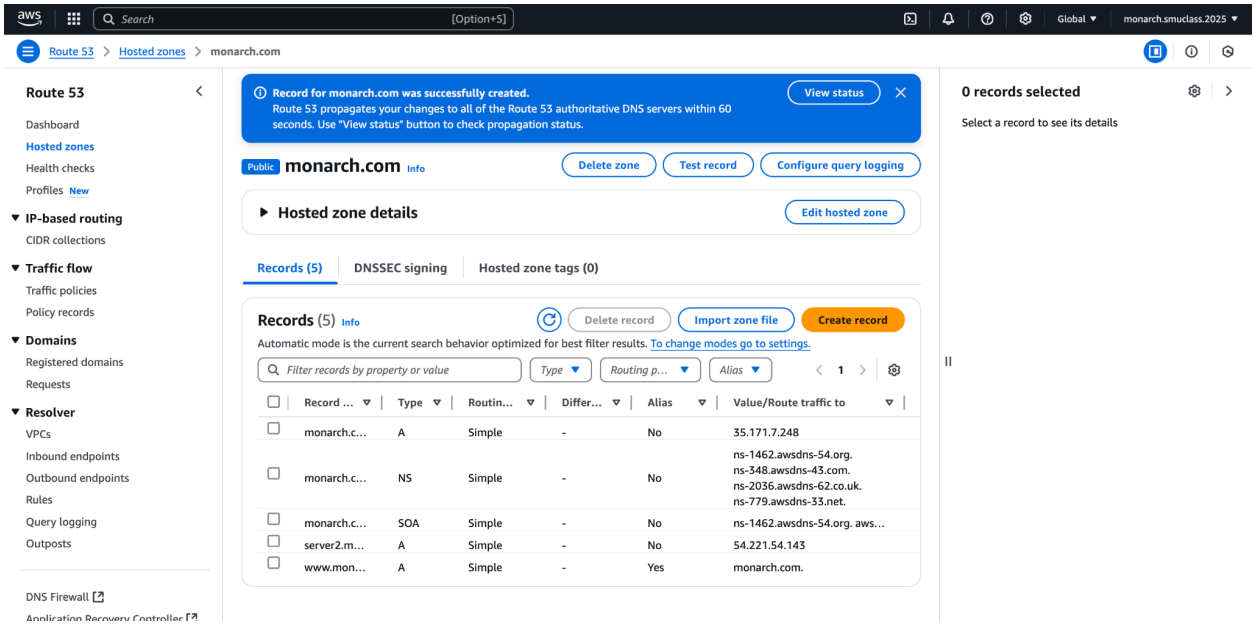
This is the Secondary Web Server

2. Configure a health check for each of your resources. You create a health check for an S3 static website by selecting Domain Name for Specify Endpoint By. Note that

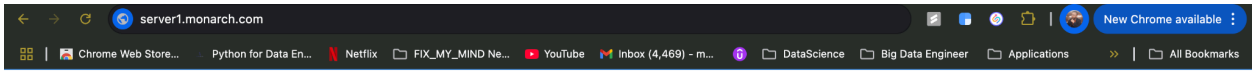
you'll need to strip off the http:// and trailing / characters from the endpoint before the health check configuration will accept it.



3. Assuming you have an active hosting zone (perhaps the one from Exercise 8.1), use the Route 53 GUI to create a record set for each of your instances. You might enter domain names like server1.example.com and server2.example.com—it's up to you.



4. Test your configuration by first loading the primary website by its normal URL. Now, disable your primary website. You could do that by deleting or renaming its index.html file, by blocking HTTP access in the instance security group, or in the case of an S3, by disabling the static website setting. Point your browser to the same primary URL. If failover is working, you should see the index.html web page from your secondary resource.

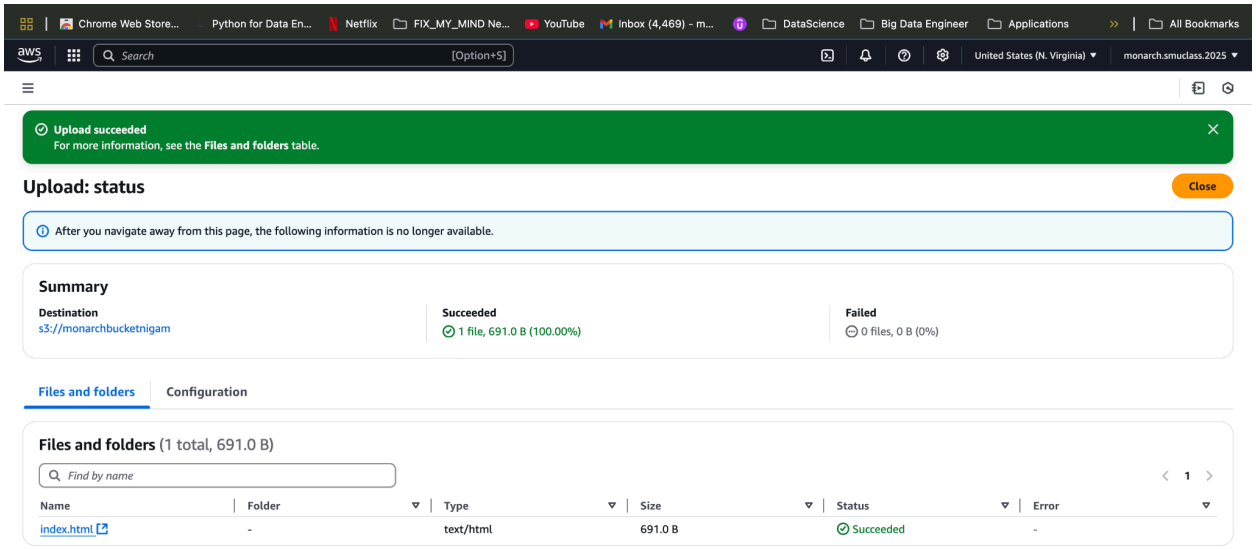


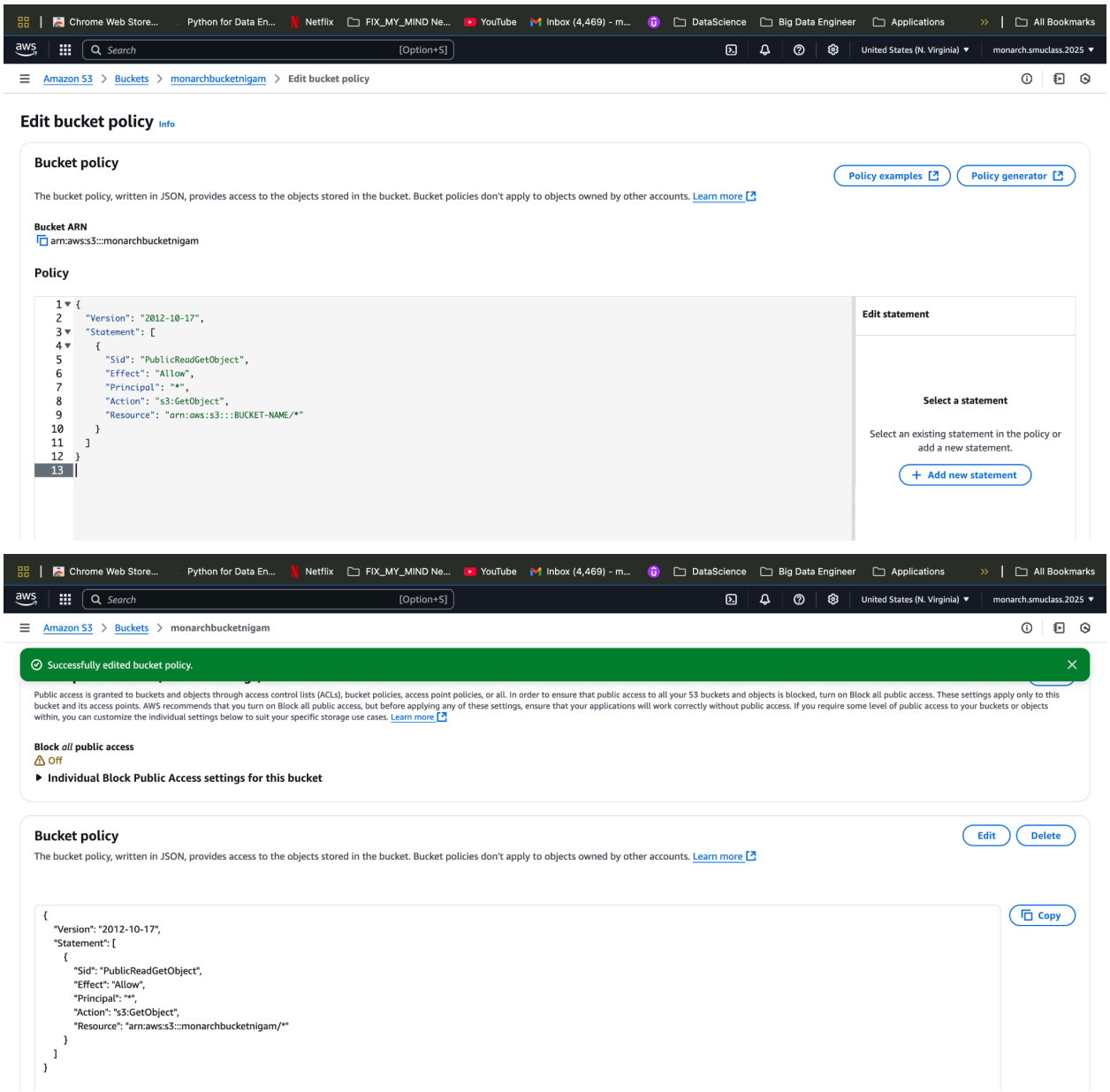
This is the Secondary Web Server

EXERCISE 8.4

Create a CloudFront Distribution for Your S3-Based Static Website

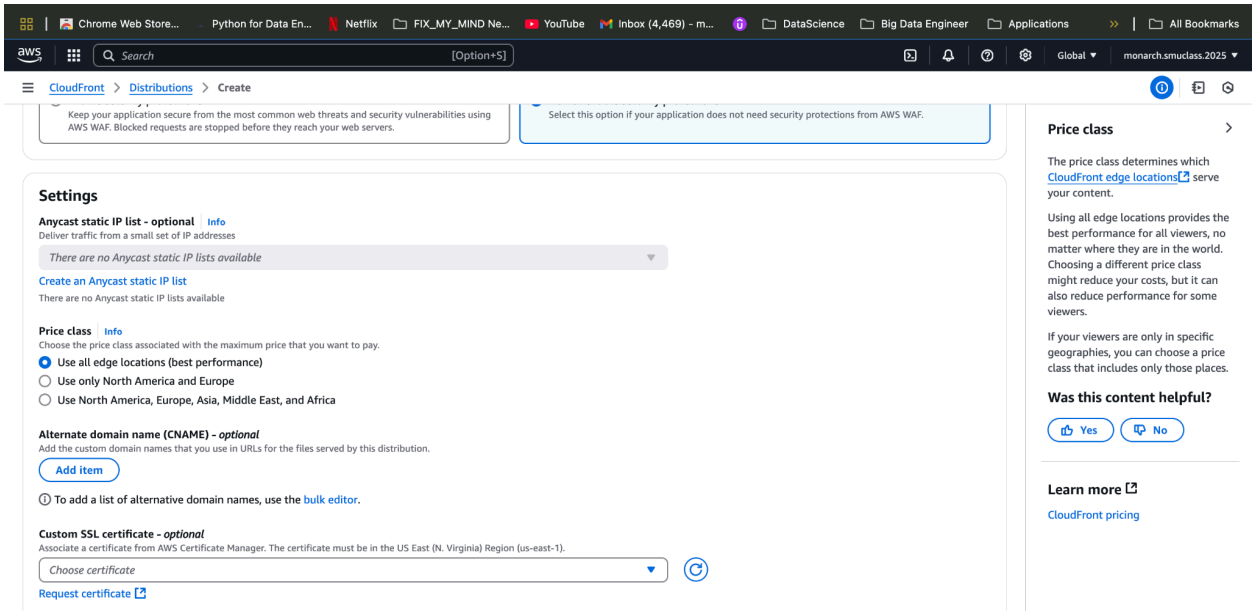
1. Create a static website in S3 the way you did in Exercise 3.4 in Chapter 3.





2. From the CloudFront dashboard, start a new distribution.
3. Click once inside the Origin domain box, and a list of all the available resources within your account will be displayed. The S3 bucket containing your website files should be included. Select it.
4. Scroll down and select a price class in the Distribution Settings section. Using the All Edge Locations setting will provide the best performance for the greatest number of customers, but it will also cost more. By the way, if you're only creating this as an

experiment and you're unlikely to get more than a few hundred requests, then don't worry about the cost; it'll be in the low pennies.



The next three steps will apply only if you have a valid DNS domain to work with.

I dont have any valid DNS and it will cost me to get one. Professor asked us to skip the steps if it is costing money. I have already been charged with the below cost.