**Securing your environment**

**In-Class Activity  
Creating Security Groups**

**Overview:**   
In this in-class activity you will get experience securing your first Virtual Private Cloud (VPC) in the cloud using Amazon Web Services (AWS). You will build upon what you learned earlier in the AWS Cloud Foundations course and what has been discussed in class. You will be doing this on a real AWS network.

There are three main tools that you can use to secure your resources in AWS:

1. **Routes** – Routes are the pathways between different parts of networks. Without a path, resources cannot be accessed. In the previous in-class activity we tailored the routes so that:
   1. Instances within the VPC could communicate with each other. This includes instances in the public subnets communicating with instances in the private subnets.
   2. Instances within the public subnets could also be accessed from nodes on the Internet through an Internet Gateway.
   3. While instances on the private subnets cannot be accessed directly from the Internet, using a NAT Instance, instances on the private subnets can reach out to the Internet to get things like patches.

Routes were covered in the previous in-class activity and will not be covered again here.

1. **Network ACLs (NACLs)** – NACLs are virtualized network firewalls implemented in the cloud. You can use NACLs to control the flow of data from one or more subnets to/from an AWS subnet. A NACL protects a subnet and every instance or service that exists on that subnet. You can protect resources on ASW subnets using NACLs, security groups or both. Security groups can provide finer control than NACLs so many organizations only use security groups. Other organizations, frequently in response to regulation, use both. For this in-class activity, we will only use security groups, not NACLs but most of what we do with security groups can also be done with NACLs at the subnet level.
2. **Security Groups** – Security groups are basically software based firewalls that can be use at the instance level. The same security group can also be used for all instances in a particular subnet, much like a NACL. For this in-class activity, we will focus on security groups.

We will set up 2 different security groups. Using these security groups we will control the flow of data based on source, destination and port numbers. In future in-class activities we may open additional ports in these security groups. The security groups are:

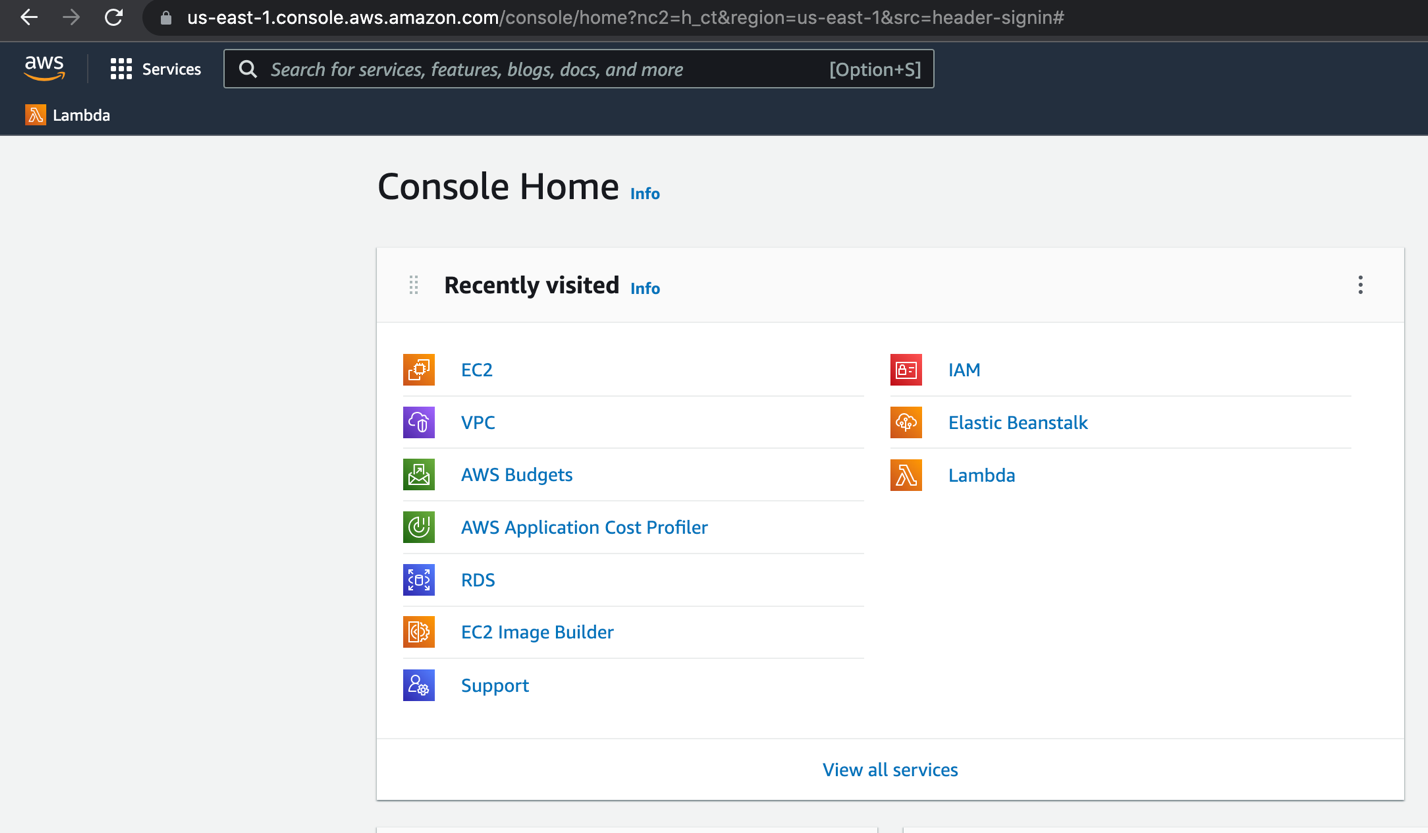
1. **WebApp Security Group** – This security group will protect our web and application servers on our public subnets. These servers will respond to HTTP (port 80) and HTTPS (port 443) requests to serve up web pages. We will also open ports 8080-8085 for future custom applications. We will also want to be able to PING instances on this network so we will want to open up ICMP ports. Finally, we will want to open up RDP and SSH so we can remote desktop into our instances on these subnets.
2. **Database Security Group** – The database servers that our Web and Application Servers will need to access will reside in our private subnets. These will only be accessed from other parts of our VPC. There will be no direct access to the database servers from the Internet. Web and Application servers will communicate over ports 3306 (MySQL), 5432 (Postgres) and 1521 (Oracle) with the Database servers. We will need to open these ports within the security group.

**Objective:**

1. Learn about the different methods of securing a VPC.
2. Gain hands-on experience creating and configuring Security Groups in AWS.

**Process:**

1. Login to AWS.
2. Go to the AWS Console.
3. Select “VPC” (for Virtual Private Cloud) under the “AWS services” menu. You may have to select “All services” if “VPC” is not listed under “Recently visited services”.

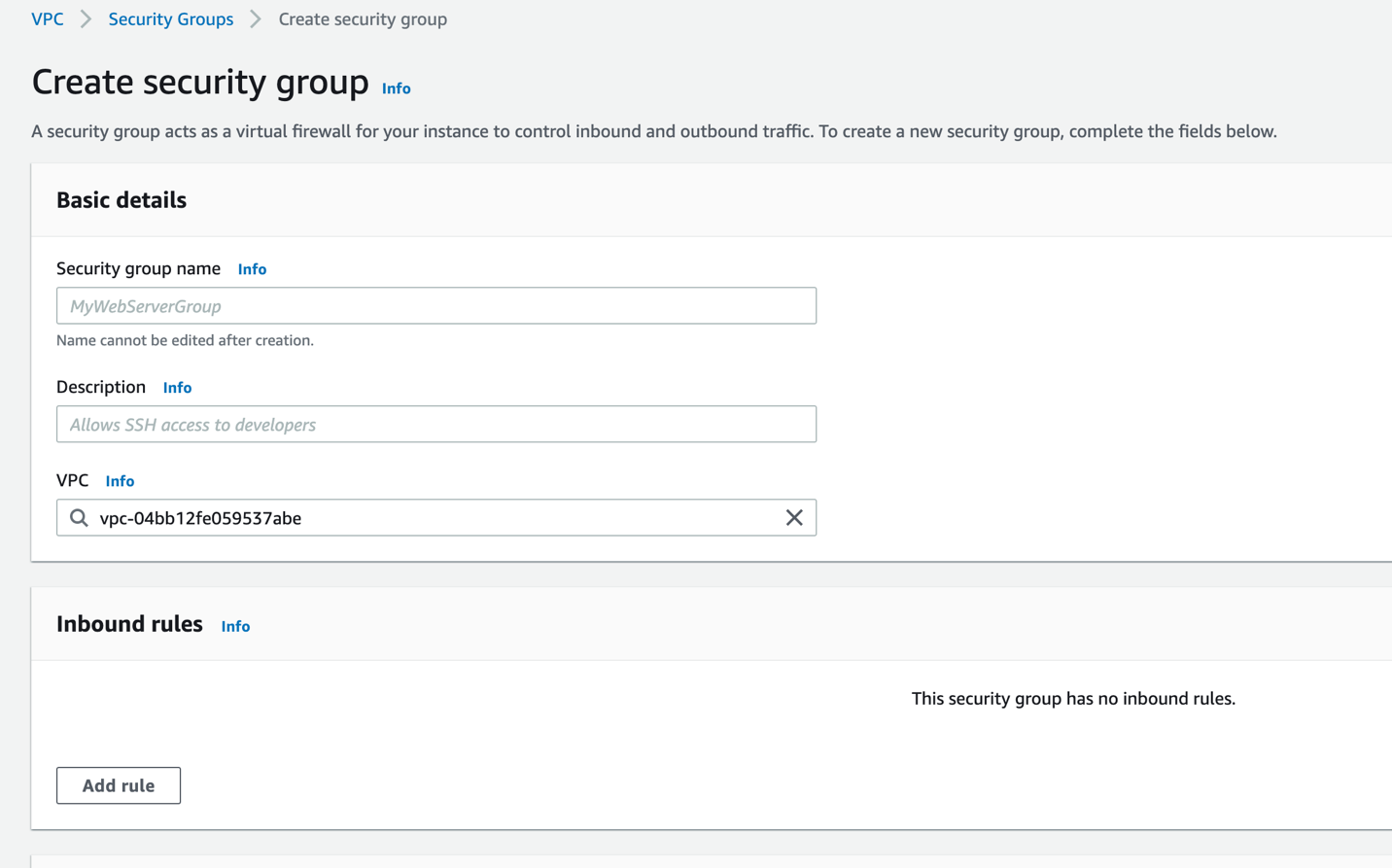


1. In this step we will create two security groups. We will give them a name, a description and associate them with a VPC. However, we will not specify any inbound or outbound rules (a.k.a. will not open any ports). We will create a new instance and associate the instance with a security group and see that we cannot access the instance or PING the instance until we have opened a few ports by specifying inbound rules.

We will start by creating the following security groups with the following descriptions within your VPC.

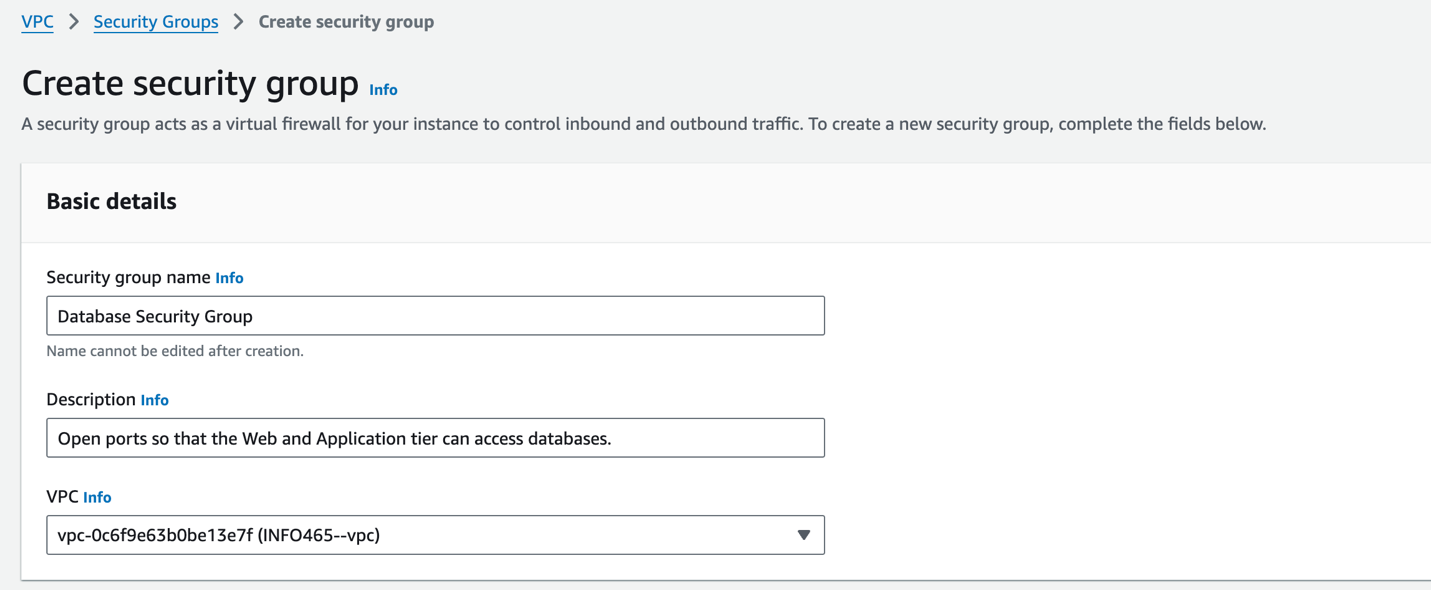
* 1. Web Security Group
  2. Database Security Group

Select “Security Groups” under “SECURITY” under the “VPC Dashboard” and click the “Create security group” button.

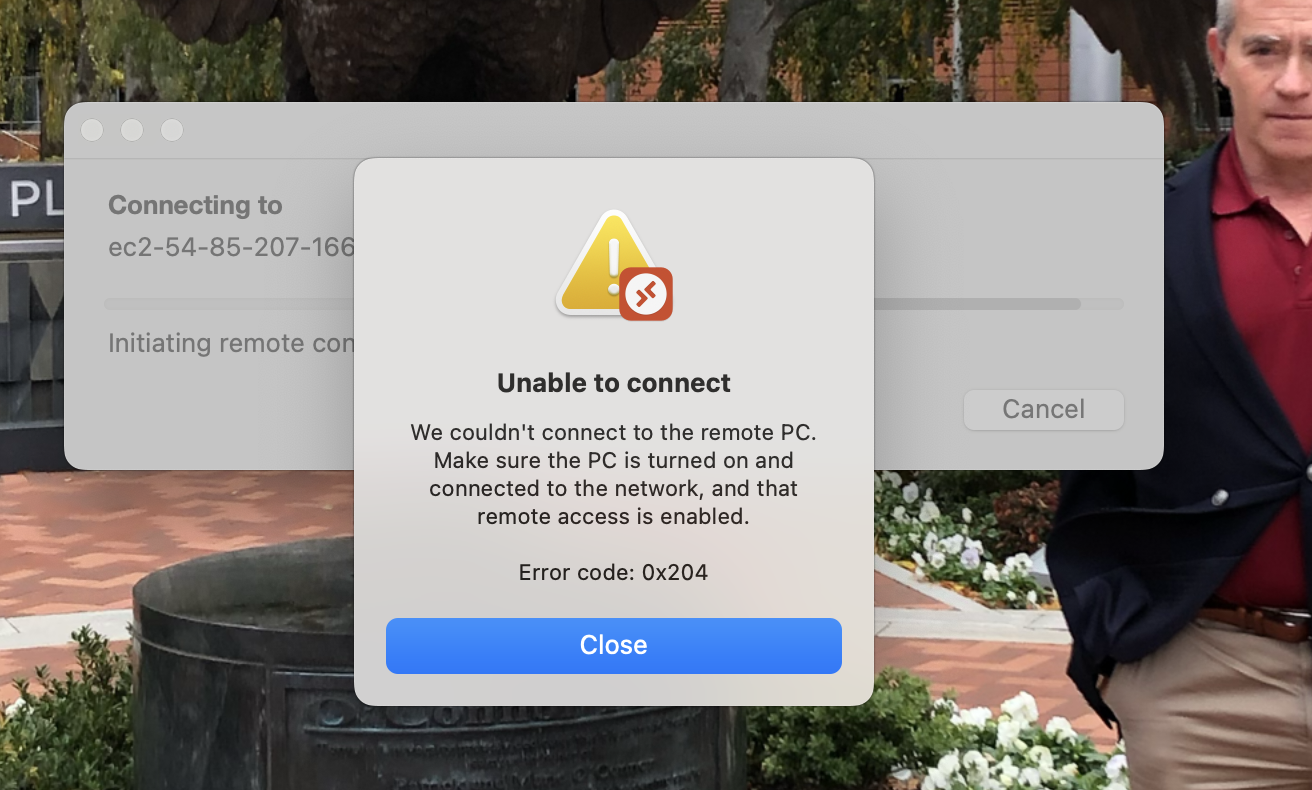


Enter “Web Security Group” for Security Group name, “Web and Application Tier Security Group” for the Description. Be sure to select your VPC.

Follow the same process to create a Database Security Group. The description should be “Open ports so that the Web and Application tier can access databases”.

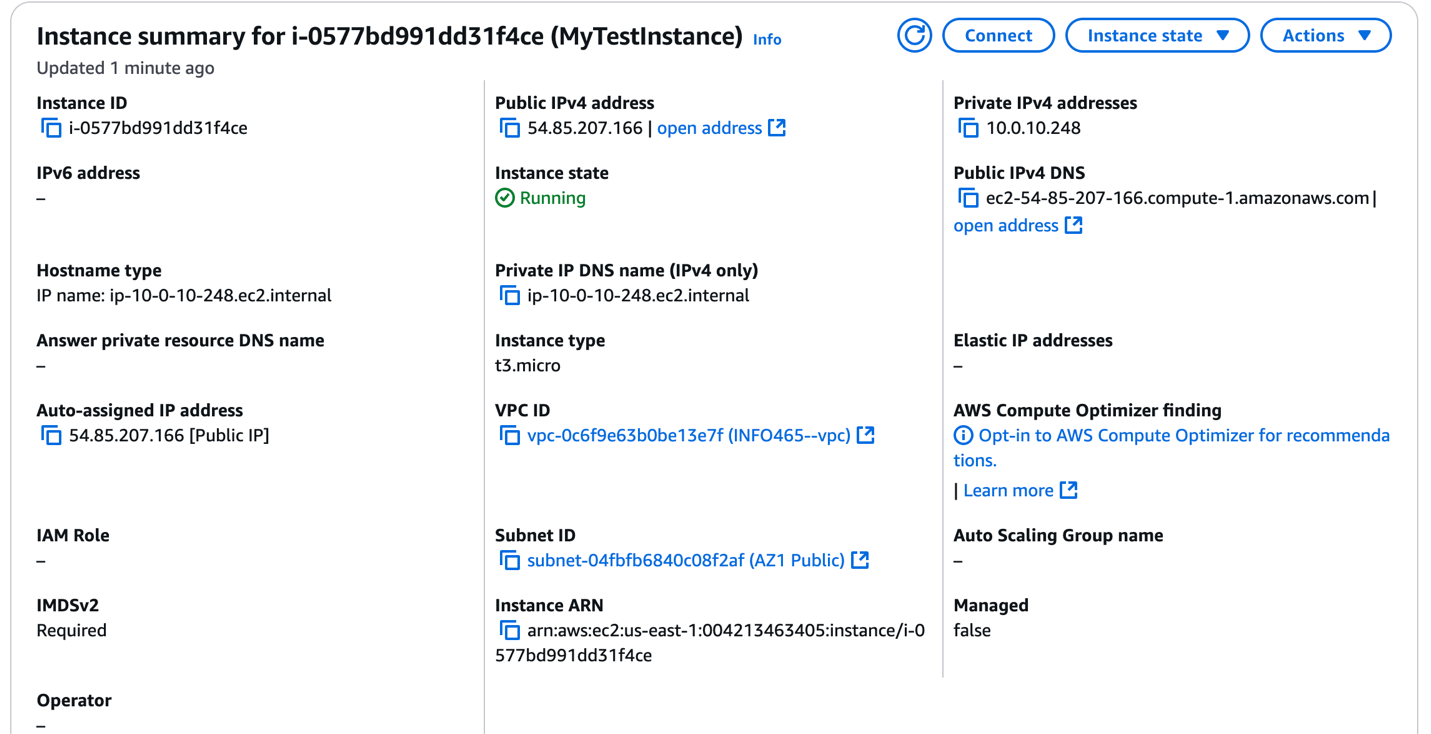


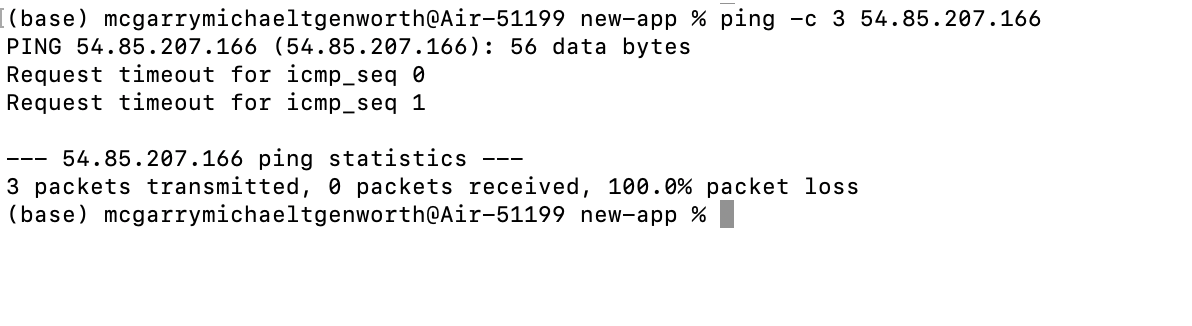
1. Using the basic directions as the previous activity, create a new instance.
   1. Name should be “Test Connectivity Instance”
   2. Pick the same “Microsoft Windows Server 2025 Base” AMI that we used last time.
   3. Pick the same “t3.micro” instance type that we used the last time.
   4. Use your .pem file we created the previous activity
   5. Edit the Network setting. Then:
      1. Select your VPC
      2. Select your AZ1 Public subnet
      3. For Firewall: Select existing security group
      4. Select Web Security Group from the dropdown list box for Common Security Groups.
   6. Accept the defaults for Configure Storage
   7. Click “Launch Instance”.
2. After the instance is running, try to connect to the instance. You will receive an error that you cannot connect.



This is because the security group does not have the Remote Desktop port 3389 open. This means communication on this port is not allowed to any instance in that security group.

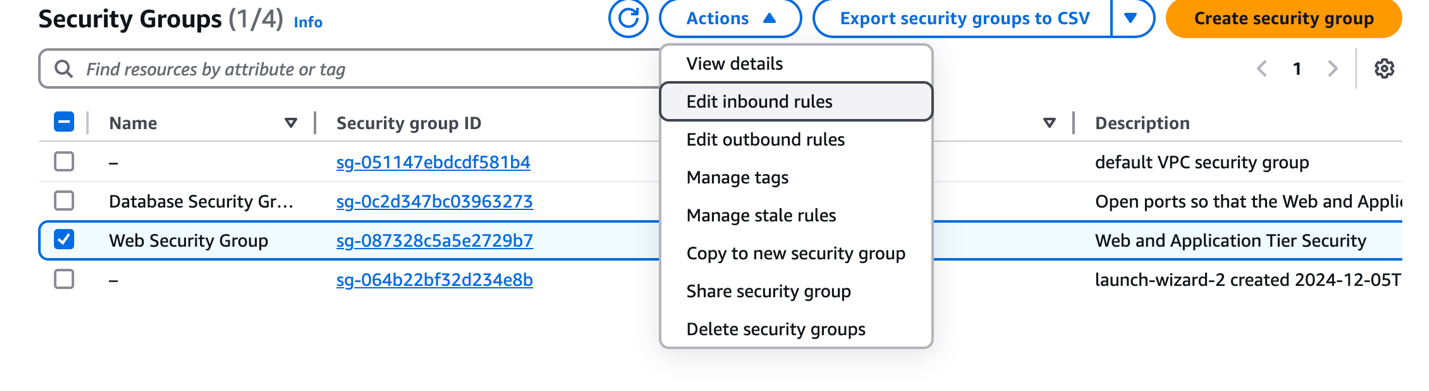
Look up the “Public IPv4 IP Address” for this instance. Open up a command prompt (Windows) or terminal session (Mac) on your laptop and try to PING the public IP address.



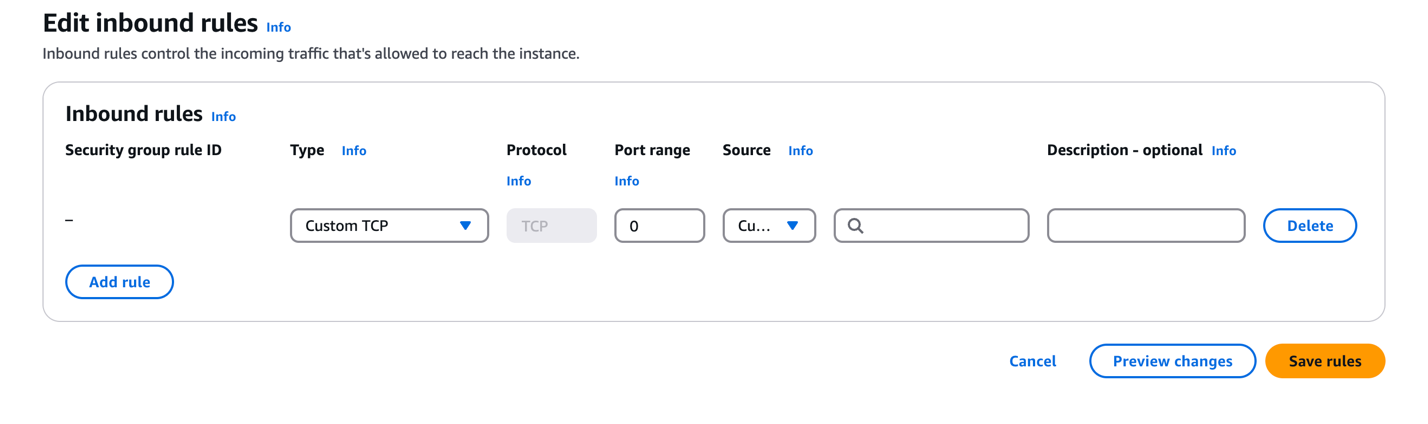


The PING will fail because the Web Security Group does not have ICMP open.

1. We will now open up the required ports for the Web Security Group. Select “Security Groups” under “Network & Security” under the “EC2 Dashboard”. Select the “Web Security Group” security group. Select “Actions…Edit inbound rules”.



You’ll see the screen below with no rules defined. Click the “Add rule” button.



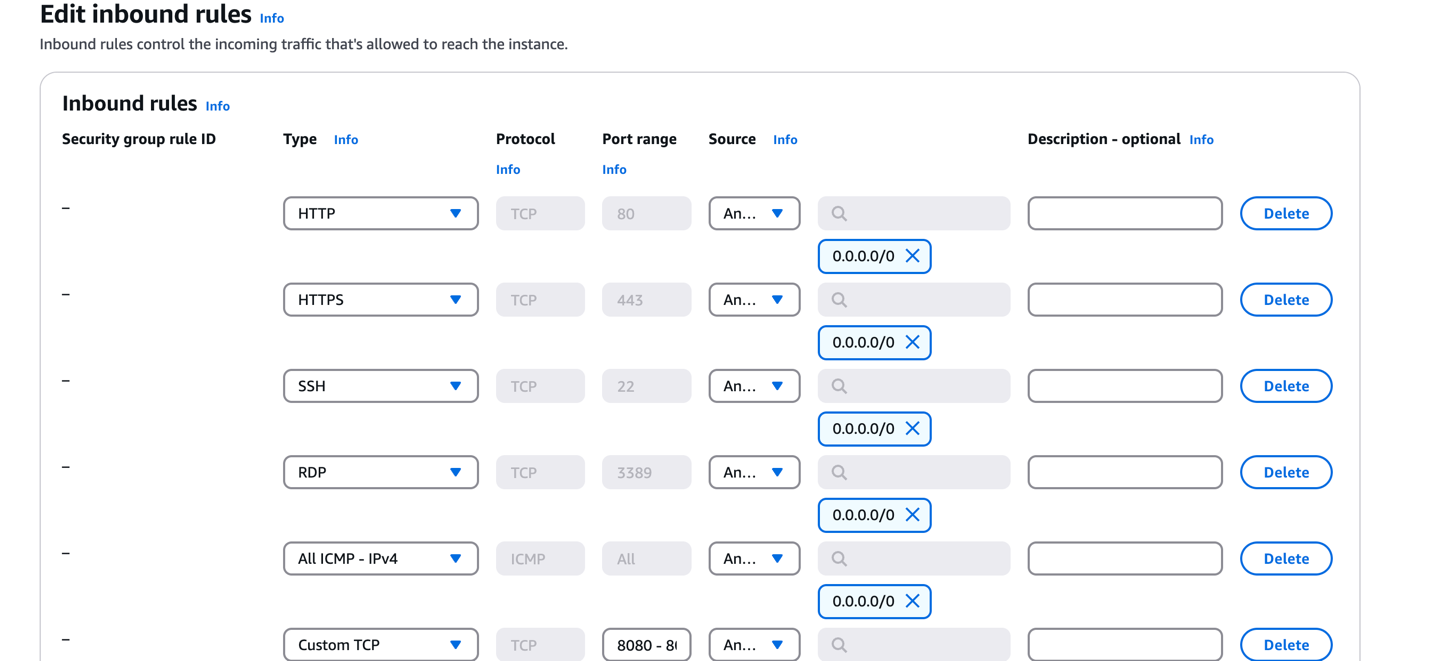
To add rules:

* Select the port to be opened from Type’s drop-down list box.
* For source, always select Anywhere IPv4.

Open the following ports:

* HTTP
* HTTPS
* RDP
* SSH
* All ICMP-IPv4
* Custom TCP with a port range of 8080-8085

Your screen should look like this:



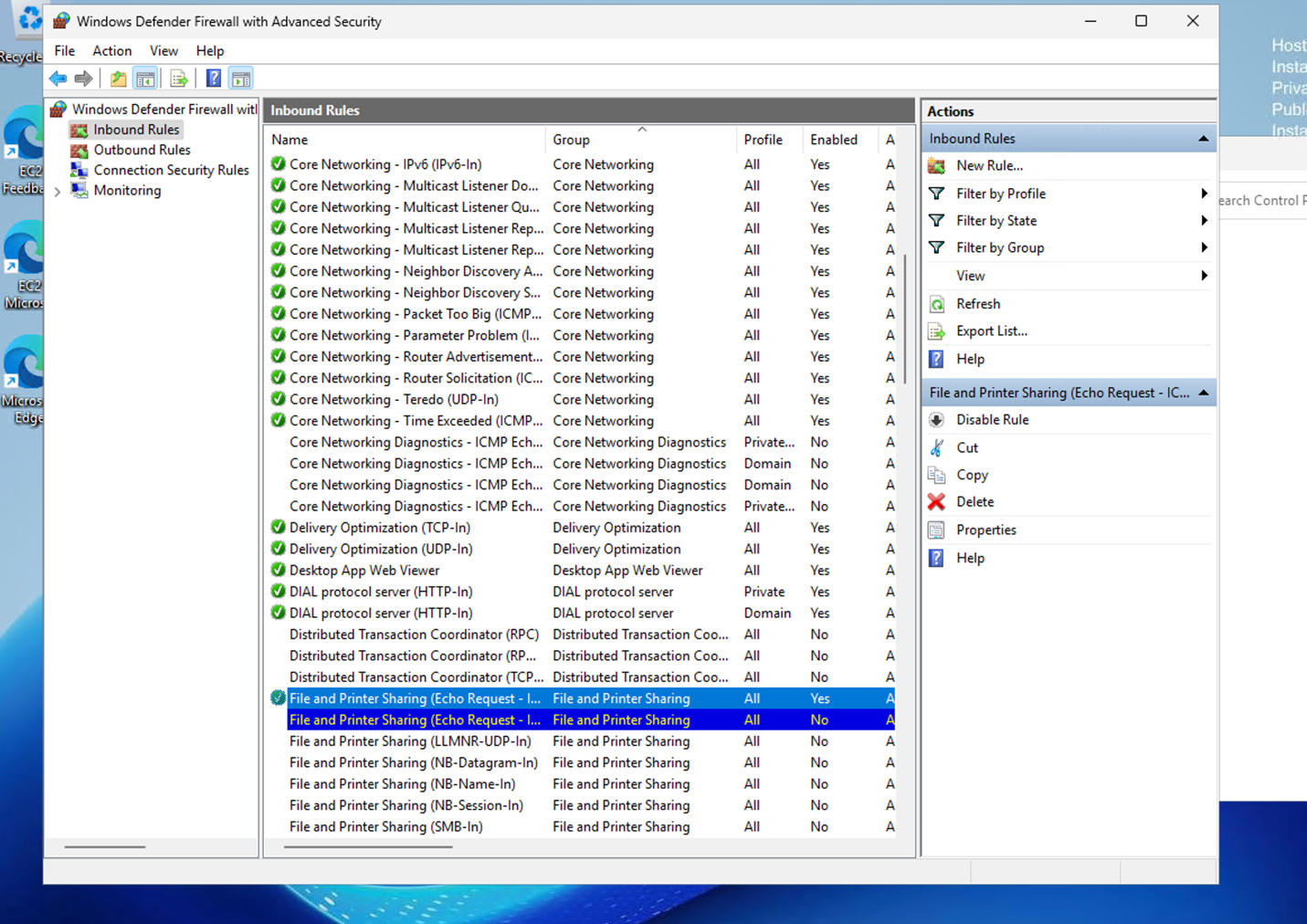
Then save the rules.

Review the outbound rules. While we are controlling inbound data, we let anything go outbound.

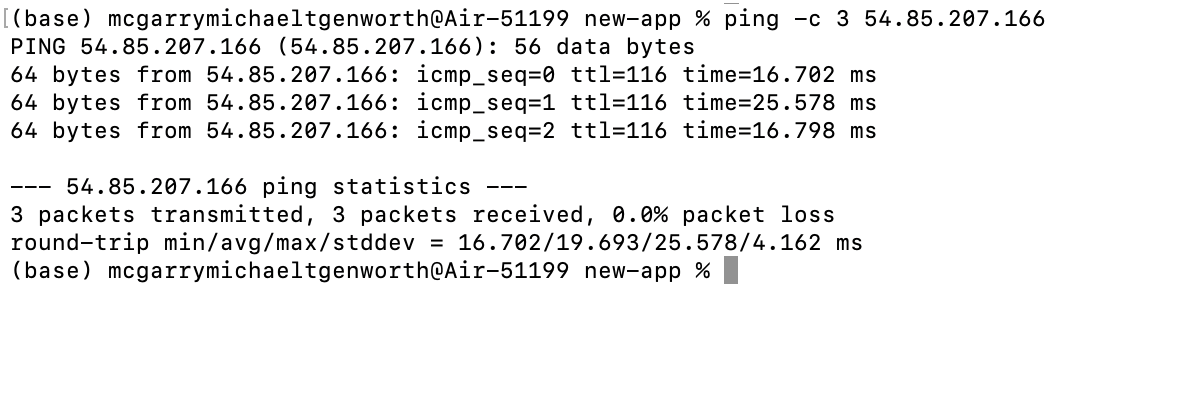
1. Now try to connect to this instance. Now that the RDP port is open, you should be able to connect. Using the IPv4 Public IP address for this instance, try to PING the instance from your computer. The PING will fail again. The issue is that by default the Windows Defender Firewall, which is on all Windows servers, blocks ICMP requests.

Connect to the “Test Connectivity Instance” instance through Remote Desktop and open up these ports on the Windows firewall by doing the following.

* Go to “Control Panel…System and Security…Windows Defender Firewall” and follow the “Advanced Settings” link on the left.
* Click on “Inbound Rules”.
* Select “File and Printer Sharing (Echo Request – ICMPv4-In)” and click “Enable Rule”



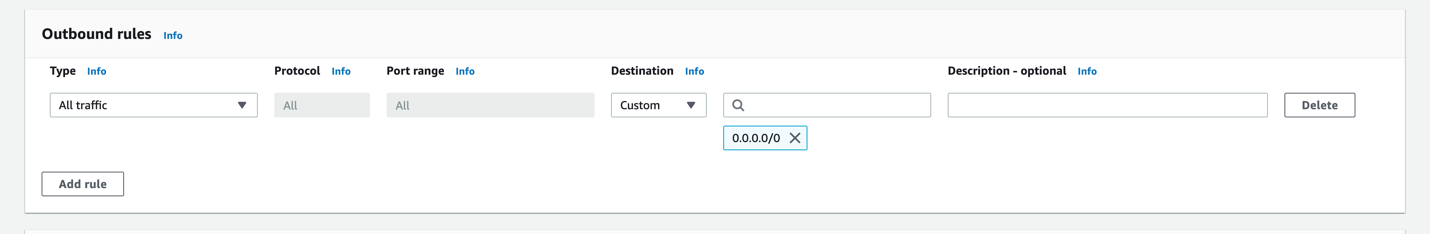
Try the PING command again – make sure you’re using the address of your instance, not the IP address in the screenshot. Now that ICMP has been opened up on both the Security Group and the Windows Firewall, the PING should succeed.



After you’ve successfully reached your server with PING, disconnect from the instance (don’t shut it down).

1. Follow the same process to open the following ports within the database security group: MySQL/Aurora, PostgreSQL, and Oracle-RDS. Your screen should look this:

Review the outbound rules. As in the previous security group, while we are controlling inbound data, we let anything go outbound.



**Post In-Class Activity Clean-up**

* 1. Please terminate the test-instance.

This concludes this in-class activity.