**VPC Flow Logs**

**Overview**

We’re going to create two EC2 instances and diagnose a connectivity issue between the two, using VPC flow logs.

I will be using ap-southeast-2 (Melbourne) for this demo, but you can use any region you like. VPC Flow Logs are available in *all* regions.

When AWS states that **VPC Flow Logs are available in all regions**, it means that you can enable and use **VPC Flow Logs** for your Virtual Private Cloud (VPC) in **any AWS region globally**. VPC Flow Logs allow you to capture detailed network traffic information, such as incoming and outgoing IP traffic to and from network interfaces within your VPC. This data can be used for:

* **Monitoring network traffic**
* **Troubleshooting connectivity issues**
* **Security auditing**
* **Analyzing traffic patterns**

Since it's available in all regions, you can enable and collect Flow Logs for VPCs in any AWS region, ensuring consistent monitoring capabilities for your network infrastructure regardless of geographic location.

**Instructions**

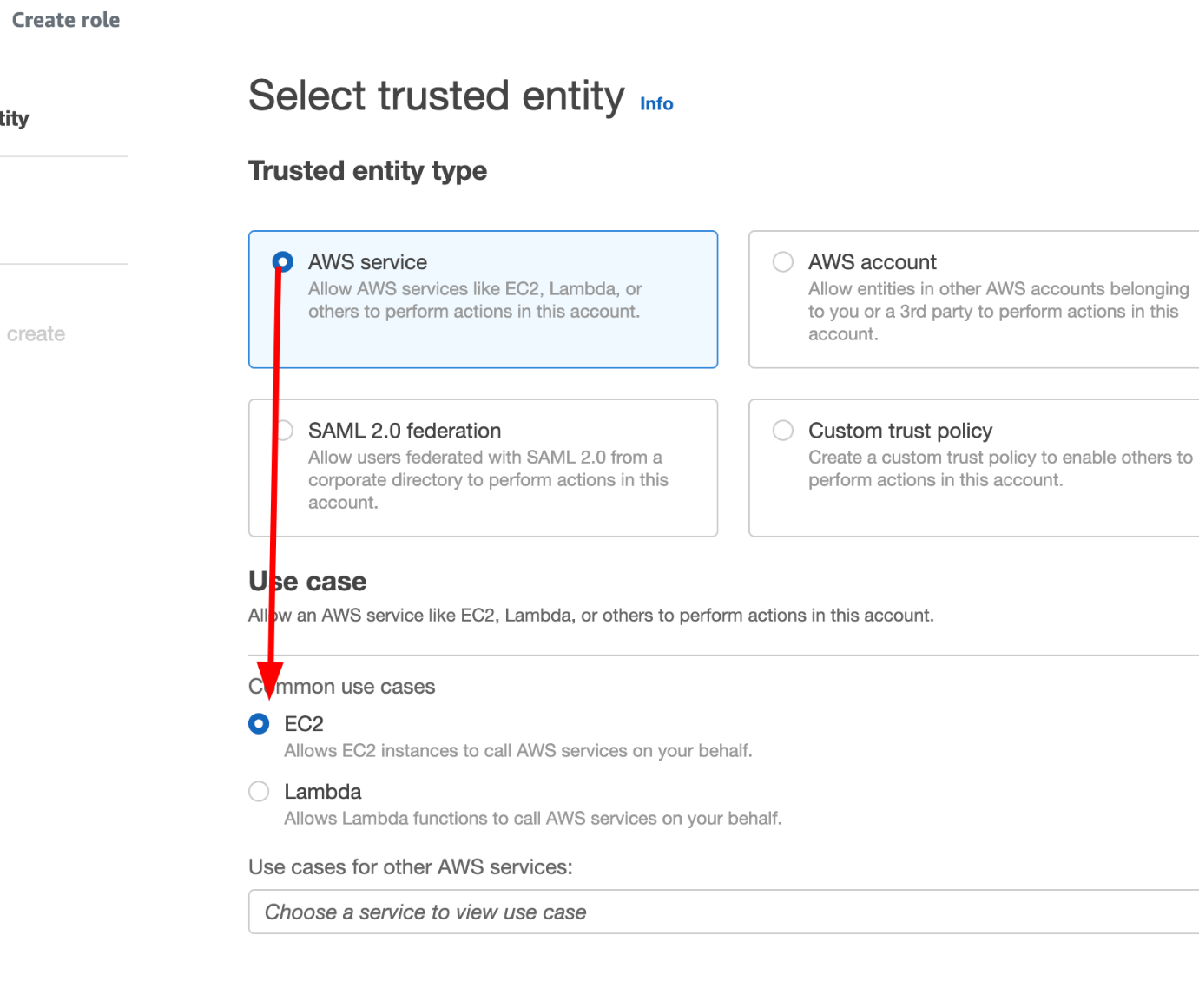
**Stage 1 - Creating IAM roles**

***EC2 SSM Session Manager role***

Head to the IAM console: <https://us-east-2.console.aws.amazon.com/iamv2/home>

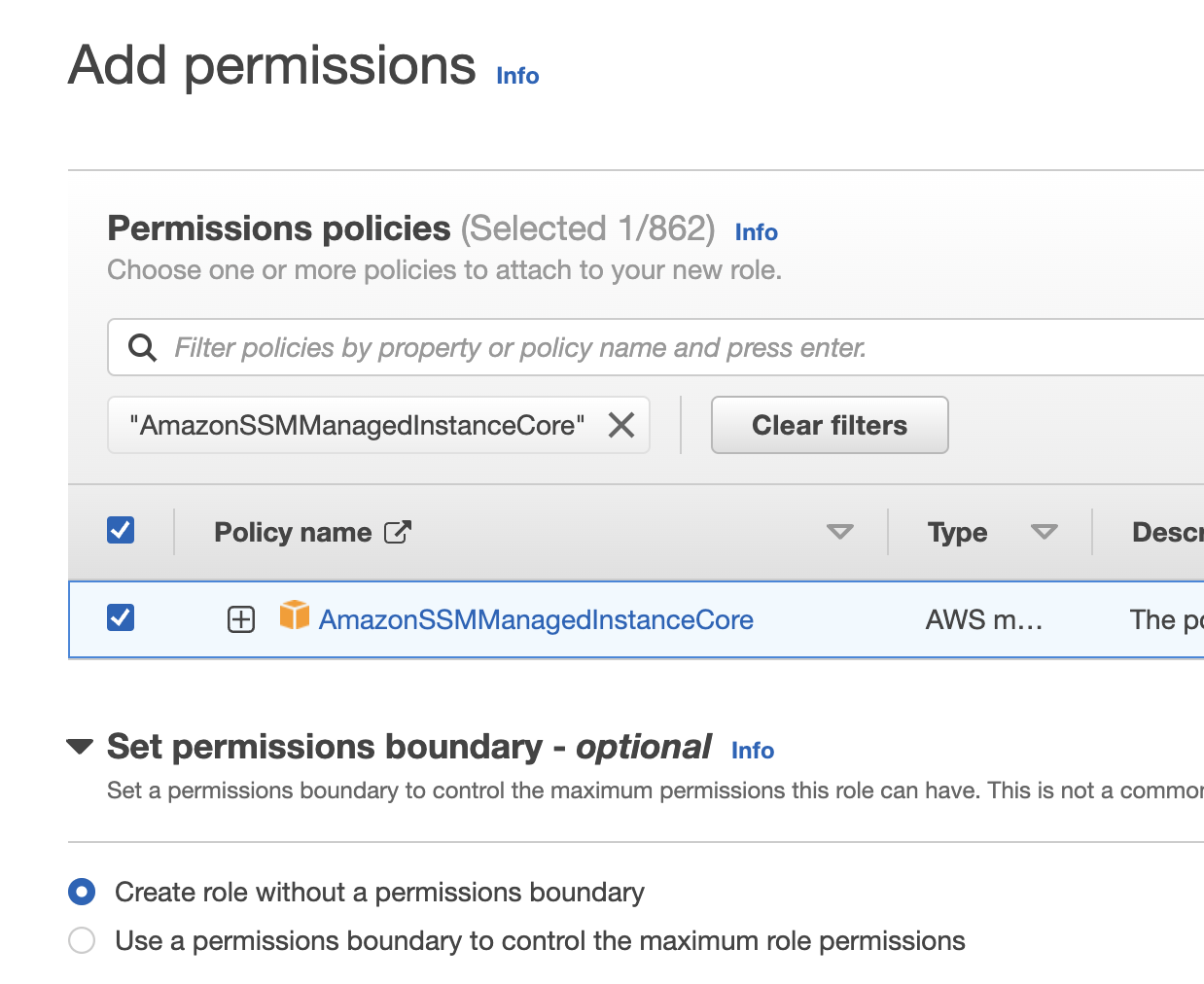
Go to Roles and click on Create role

Leave “AWS service” selected and choose “EC2”

[](https://github.com/saiguda654/vpc_flow_logs_diagnose/blob/main/images/Untitled.png)

Click Next

On the Add Permissions page, search for and select AmazonSSMManagedInstanceCore

[](https://github.com/saiguda654/vpc_flow_logs_diagnose/blob/main/images/Untitled%201.png)

Click Next

Set the “Role Name” to Demo-SSM-Role

Click Create role

***VPC Flow Logs role***

Go to Roles and click on Create role

Select “Custom trust policy” and enter the following policy

{

"Version": "2012-10-17",

"Statement": [

{

"Effect": "Allow",

"Principal": {

"Service": "vpc-flow-logs.amazonaws.com"

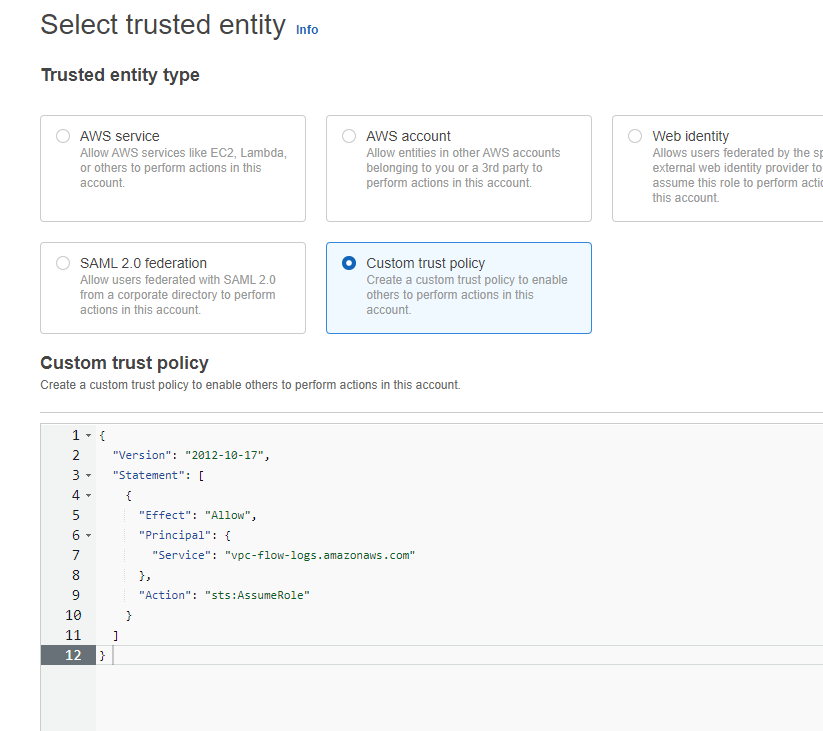
},

"Action": "sts:AssumeRole"

}

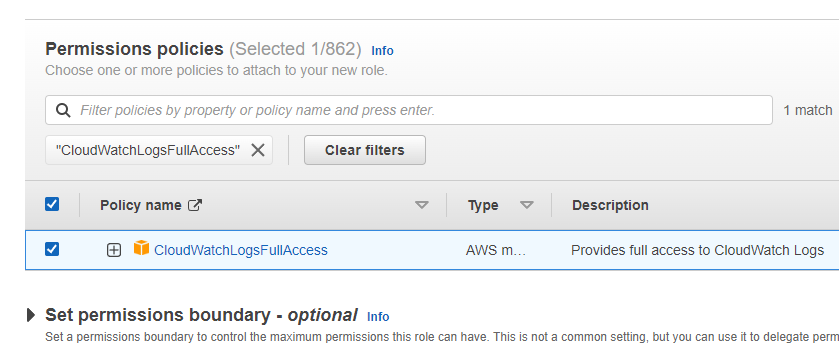
]

}

[](https://github.com/saiguda654/vpc_flow_logs_diagnose/blob/main/images/Untitled%202.png)

Click Next

Under “Add permissions”, search for and select CloudWatchLogsFullAccess

[](https://github.com/saiguda654/vpc_flow_logs_diagnose/blob/main/images/Untitled%203.png)

Click Next

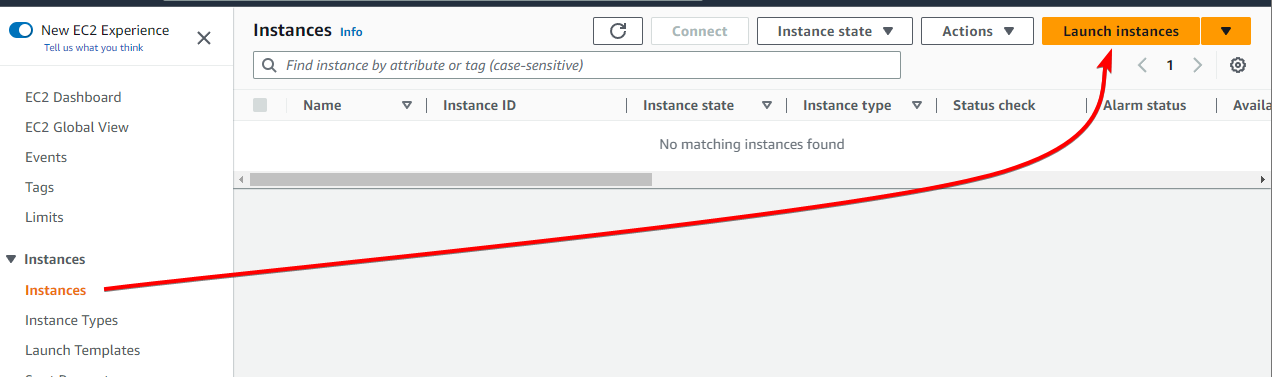
In the “Role name” enter Demo-VPC-Flow-Logs

Click Create role

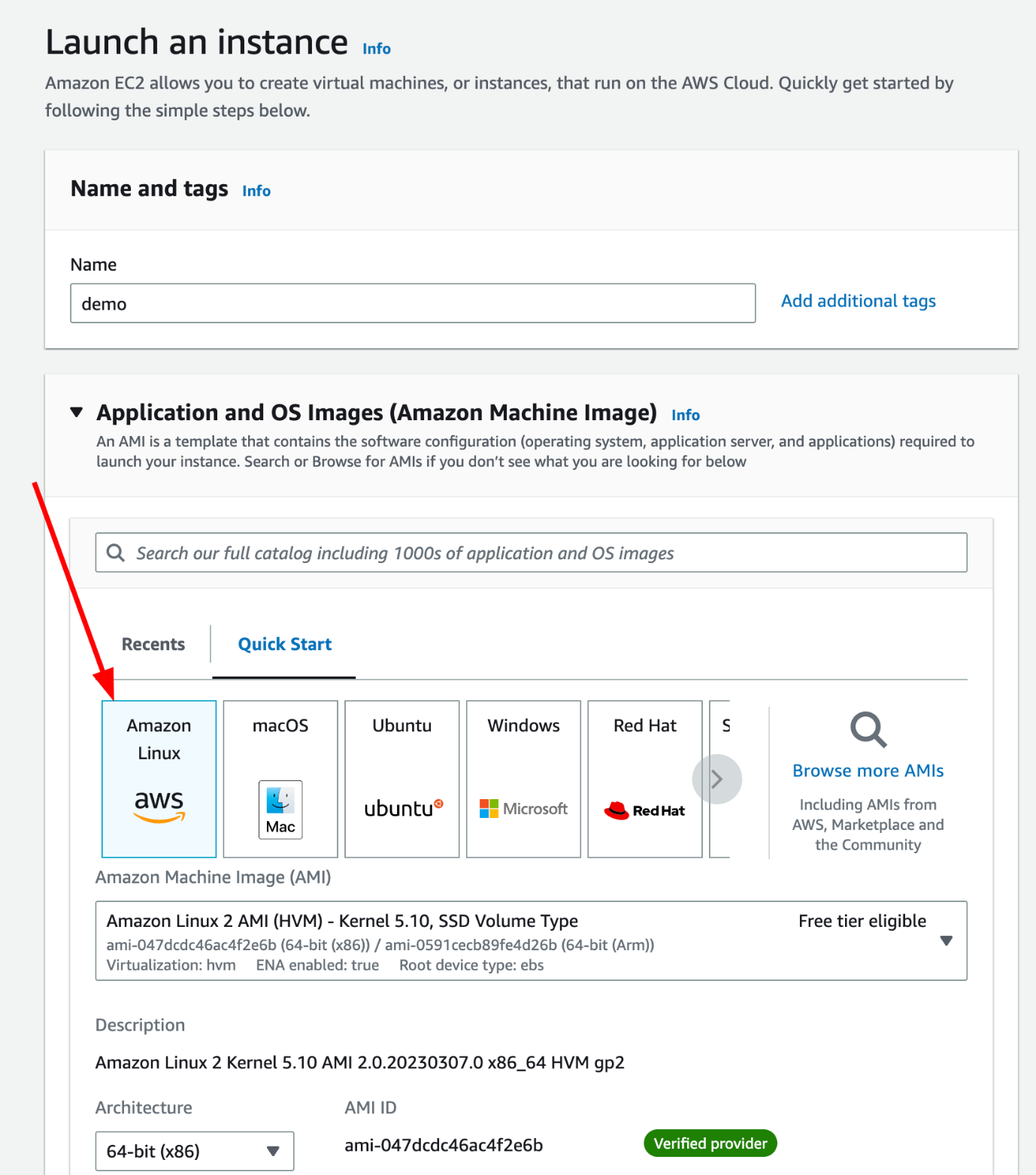
**Stage 2 - Creating EC2 instances**

Head to the EC2 dashboard: <https://ap-southeast-2.console.aws.amazon.com/ec2/home>

Click on Launch instances

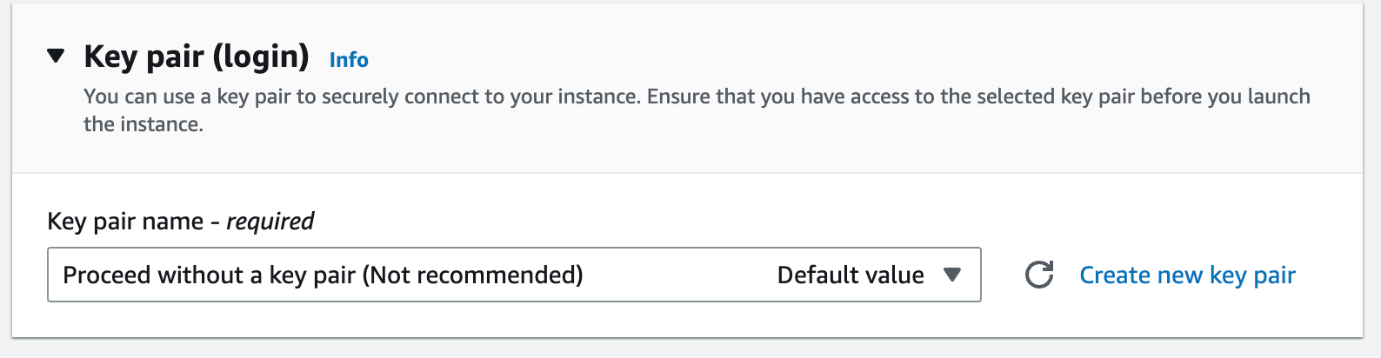
[](https://github.com/saiguda654/vpc_flow_logs_diagnose/blob/main/images/Untitled%204.png)

Set the instance name to anything you like and make sure Amazon Linux is selected

[](https://github.com/saiguda654/vpc_flow_logs_diagnose/blob/main/images/Untitled%205.png)

Leave the “Instance Type” on the default (probably t2.micro/t3.micro)

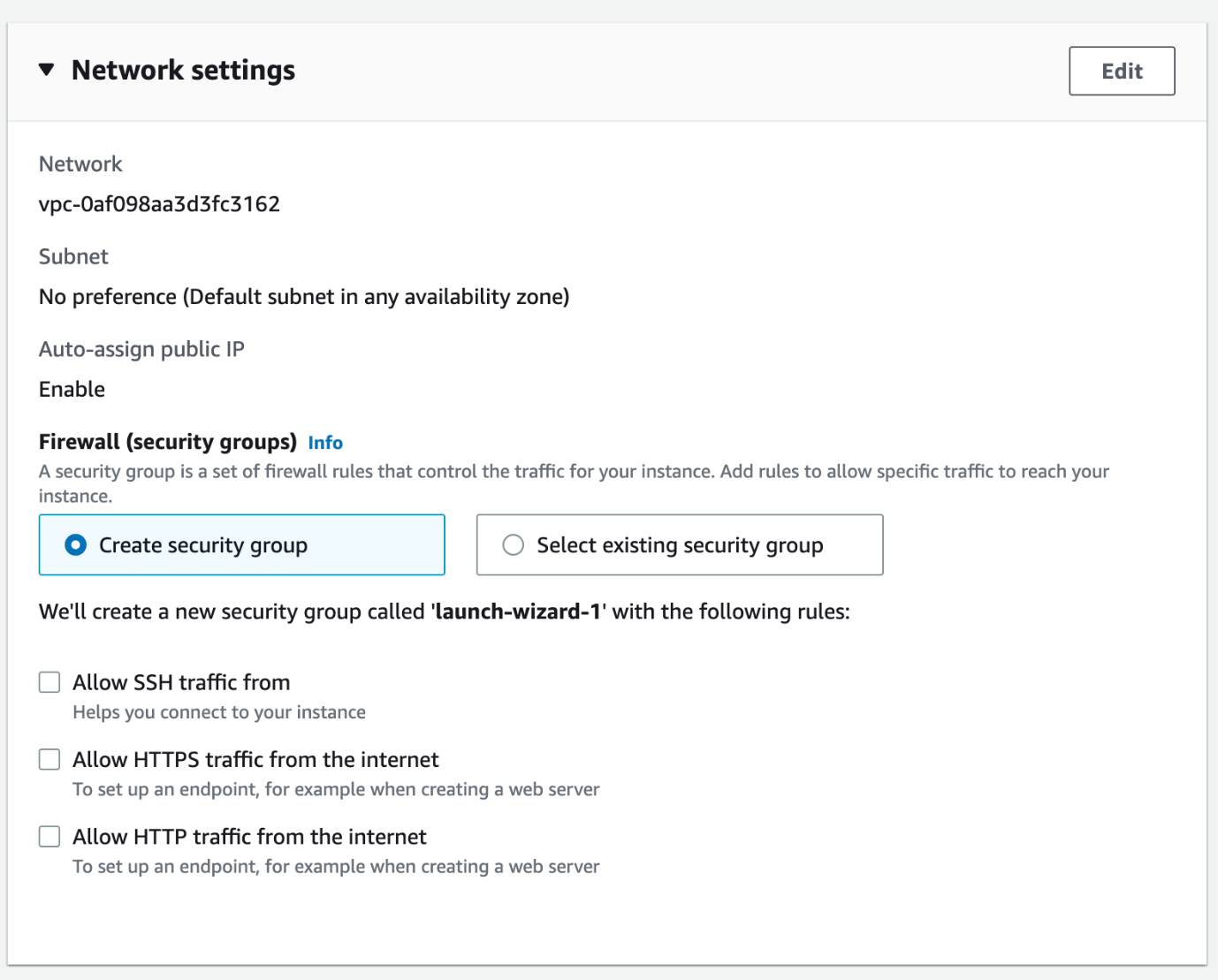
Under “Key Pair” set to “Proceed without a key pair”

[](https://github.com/saiguda654/vpc_flow_logs_diagnose/blob/main/images/Untitled%206.png)

Under “Network Settings”, leave the default VPC selected.

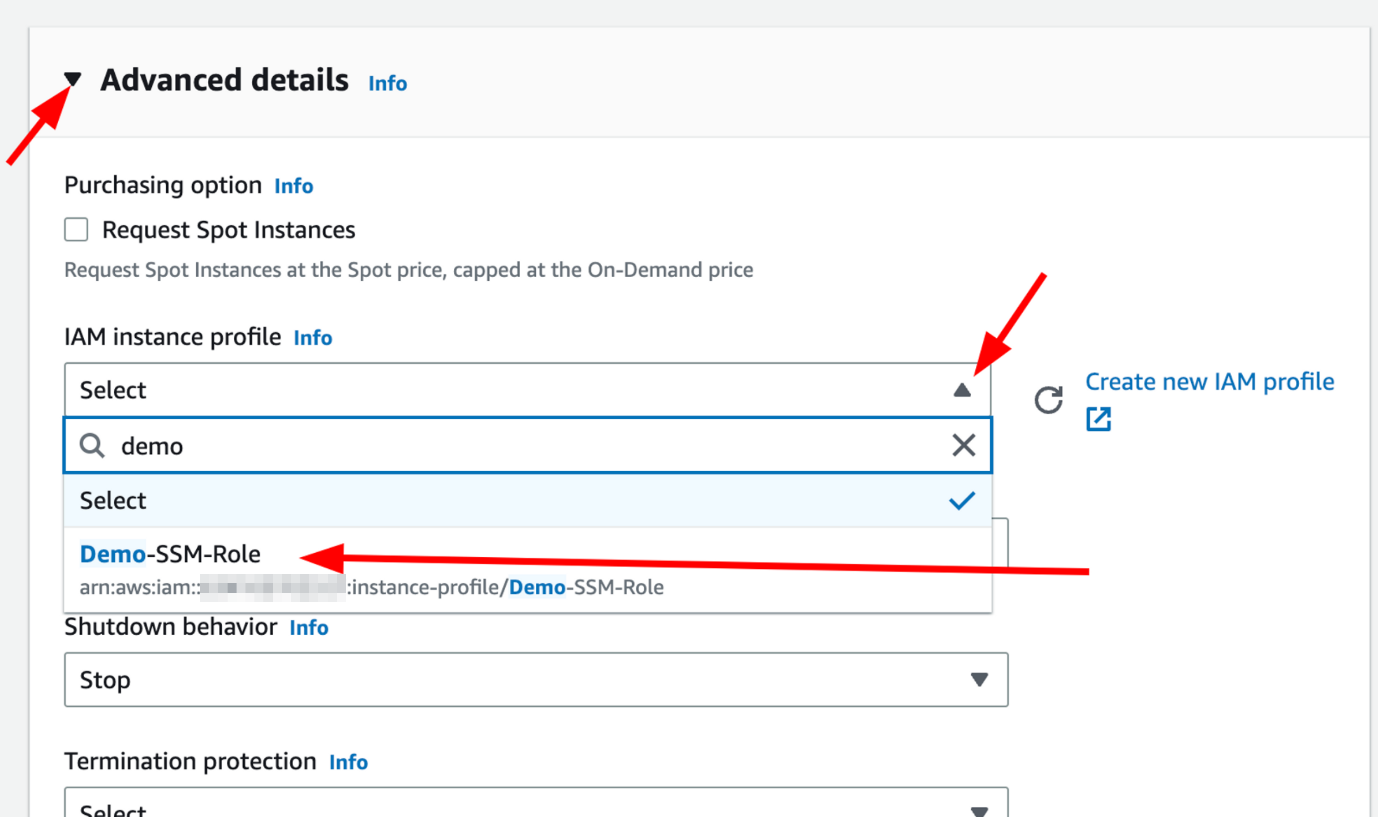
Leave “Create security group” selected

Uncheck “Allow SSH traffic from”

[](https://github.com/saiguda654/vpc_flow_logs_diagnose/blob/main/images/Untitled%207.png)

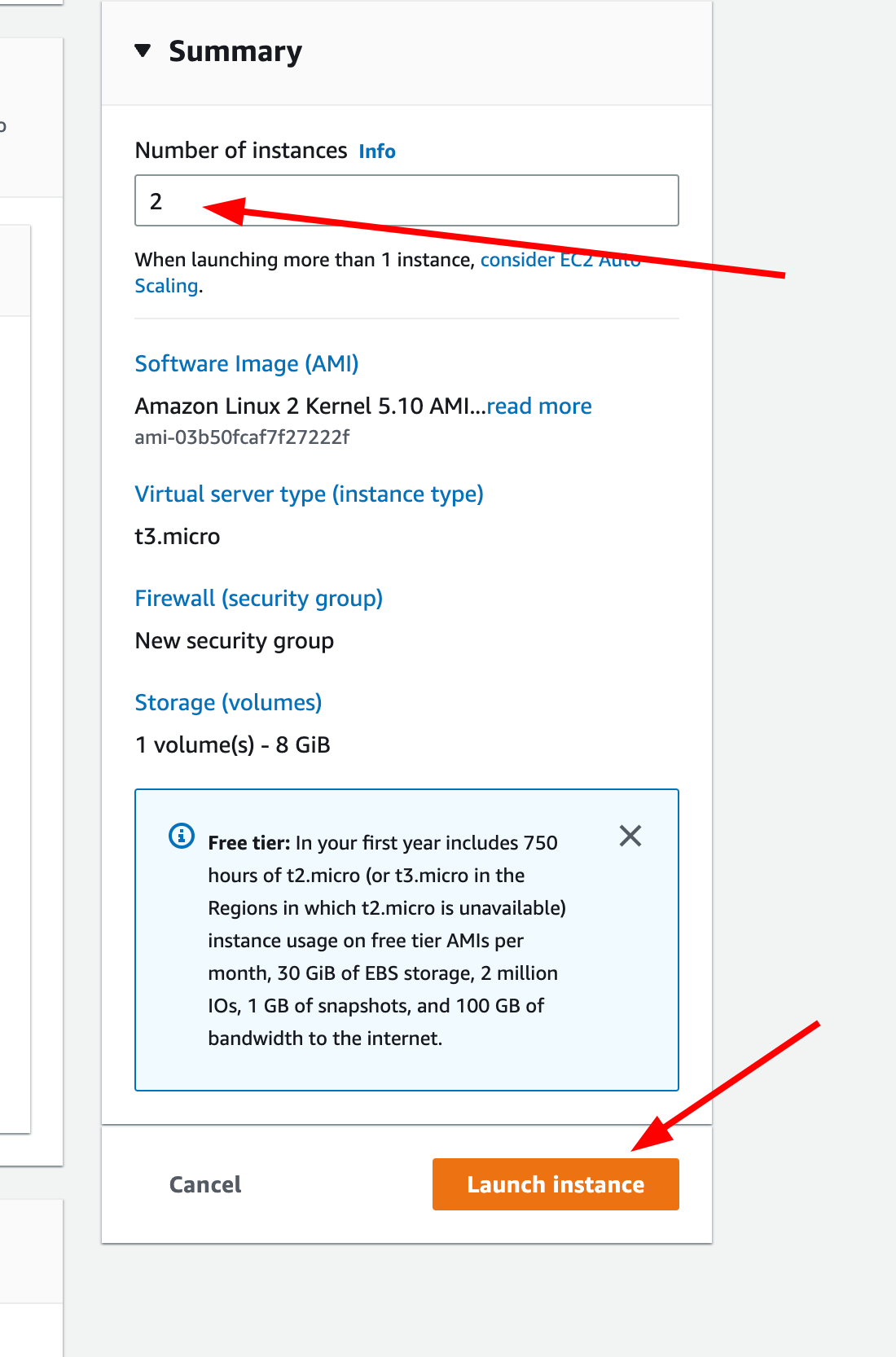
Expand “Advanced details”

Under “IAM instance profile” select the role you created in the previous step

[](https://github.com/saiguda654/vpc_flow_logs_diagnose/blob/main/images/Untitled%208.png)

Leave all other settings as they are.

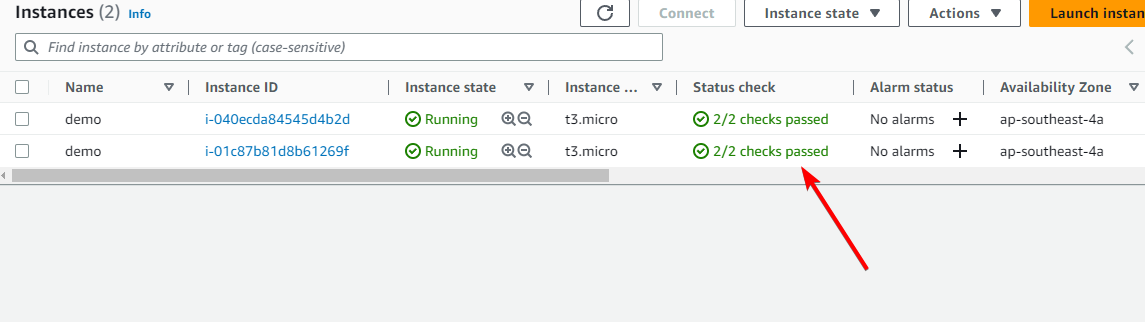
Change the number of instances to launch to 2 and click Launch instance

[](https://github.com/saiguda654/vpc_flow_logs_diagnose/blob/main/images/Untitled%209.png)

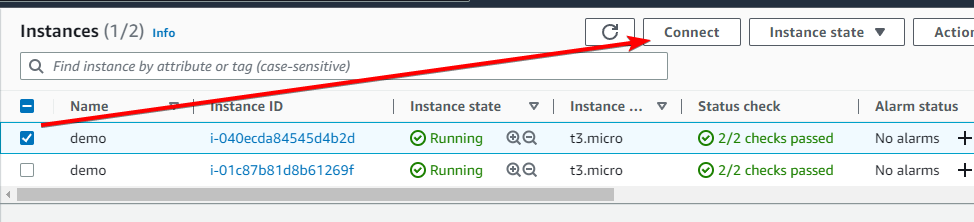
**Stage 3 - Login to both instances**

Head to the EC2 console: <https://ap-southeast-2.console.aws.amazon.com/ec2/home>

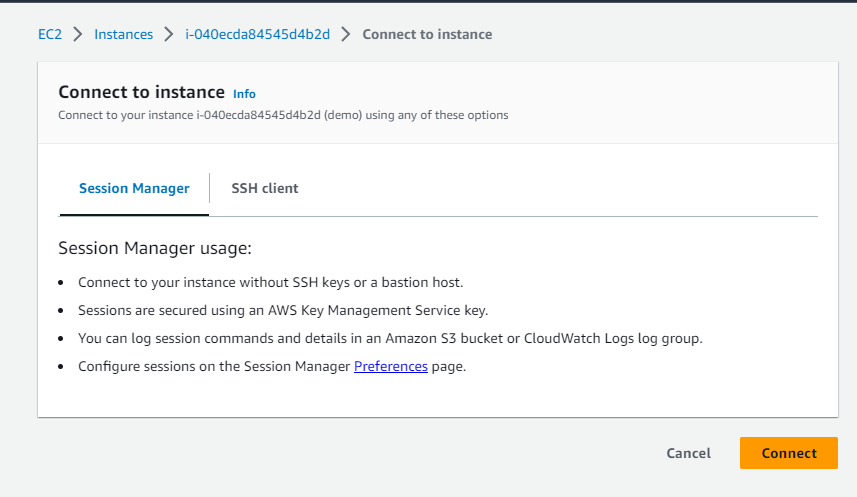
Go to Instances, and check both instances have passed both health checks

[](https://github.com/saiguda654/vpc_flow_logs_diagnose/blob/main/images/Untitled%2010.png)

If they have, select the first instance and click Connect

[](https://github.com/saiguda654/vpc_flow_logs_diagnose/blob/main/images/Untitled%2011.png)

On the next page, the Session Manager tab should already be selected, and the Connect button should be orange (not disabled). Click on Connect

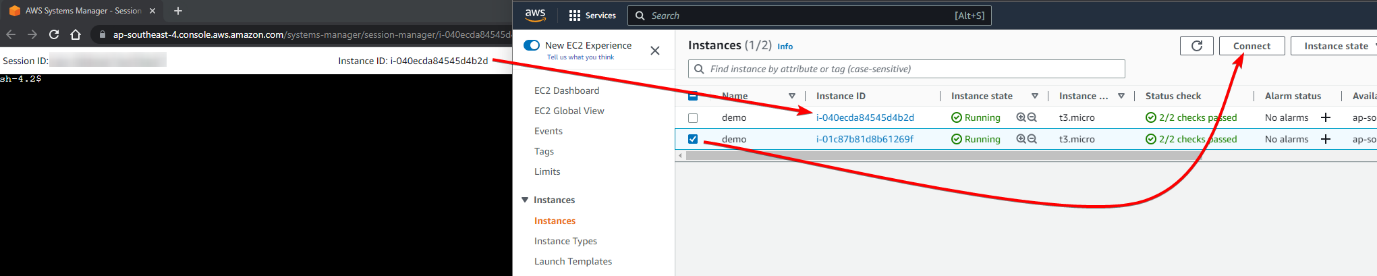
[](https://github.com/saiguda654/vpc_flow_logs_diagnose/blob/main/images/Untitled%2012.png)

This is going to open a new tab, and provide you with a shell to the instance, just like you would get using SSH.

In a new tab, head back to the EC2 console: <https://ap-southeast-2.console.aws.amazon.com/ec2/home>

Go to Instances, and select the *other* instance. You can compare the instance ID from the other tab / window to confirm you’re not connecting to the same instance twice.

Click on Connect

[](https://github.com/saiguda654/vpc_flow_logs_diagnose/blob/main/images/Untitled%2013.png)

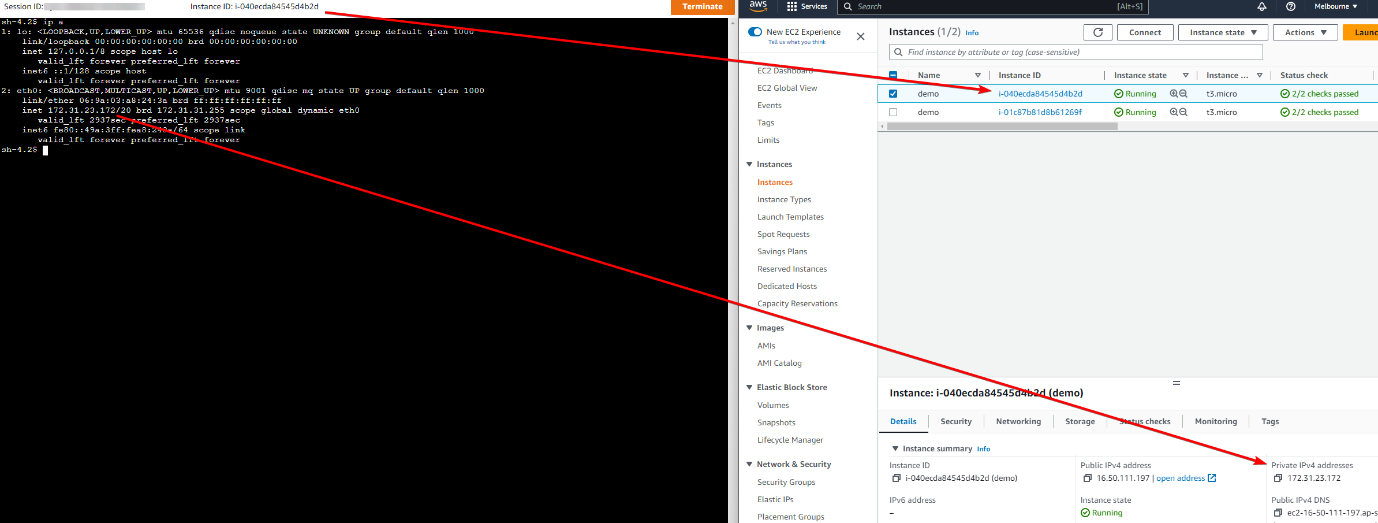
Again, on the next page, click on Connect

Now you should have two tabs, or two windows, with a shell to both of your newly created instances.

**Stage 4 - Test connectivity between the instances**

On both of the instance shells, run ip a. This is short for ip address and shows all of the IP addresses assigned to *all* interfaces.

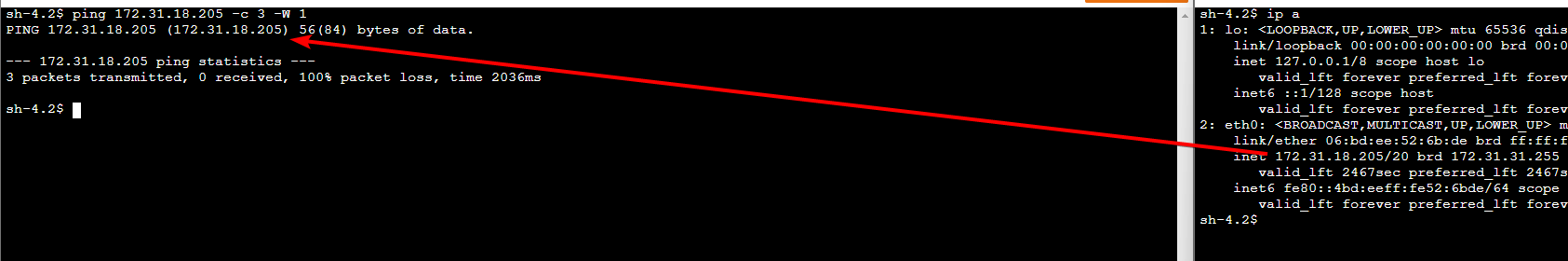
The IP on interface eth0 is the same IP that is shown in the EC2 console

[](https://github.com/saiguda654/vpc_flow_logs_diagnose/blob/main/images/Untitled%2014.png)

*Tip*: ctrl + L or entering the command clear will clear the screen to make things easier to read. So, if you notice my terminal output disappearing, that is why.

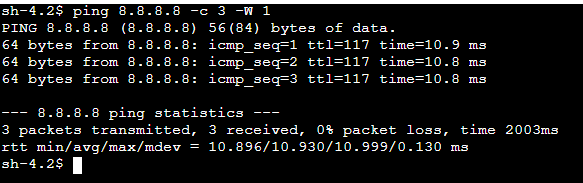
Now let’s try pinging the other instance. Grab the IP of the opposite instance and enter into the shell ping <ip address> -c 3 -W 1 followed by the IP. In my case this is ping 172.31.18.205 -c 3 -W 1

The -c 3 tells ping to send 3 ping packets, then exit. By default, ping will ping continuously forever (unless you exit, using ctrl + c). The -W 1 tells ping to wait 1 second, and if it doesn’t hear a response, consider that packet timed out. Ping packets should rarely take over 1 second to return, even pinging a server on the other side of the world.

[](https://github.com/saiguda654/vpc_flow_logs_diagnose/blob/main/images/Untitled%2015.png)

As you can see, the ping output is telling us there were 3 packets sent, 0 packets received, and 100% packet loss. Meaning the other instance didn’t respond at all.

Let’s confirm it’s not an issue on our side, we’ll try pinging a known working host, Google’s DNS servers, which have the super easy to remember IP of 8.8.8.8

[](https://github.com/saiguda654/vpc_flow_logs_diagnose/blob/main/images/Untitled%2016.png)

Much healthier.

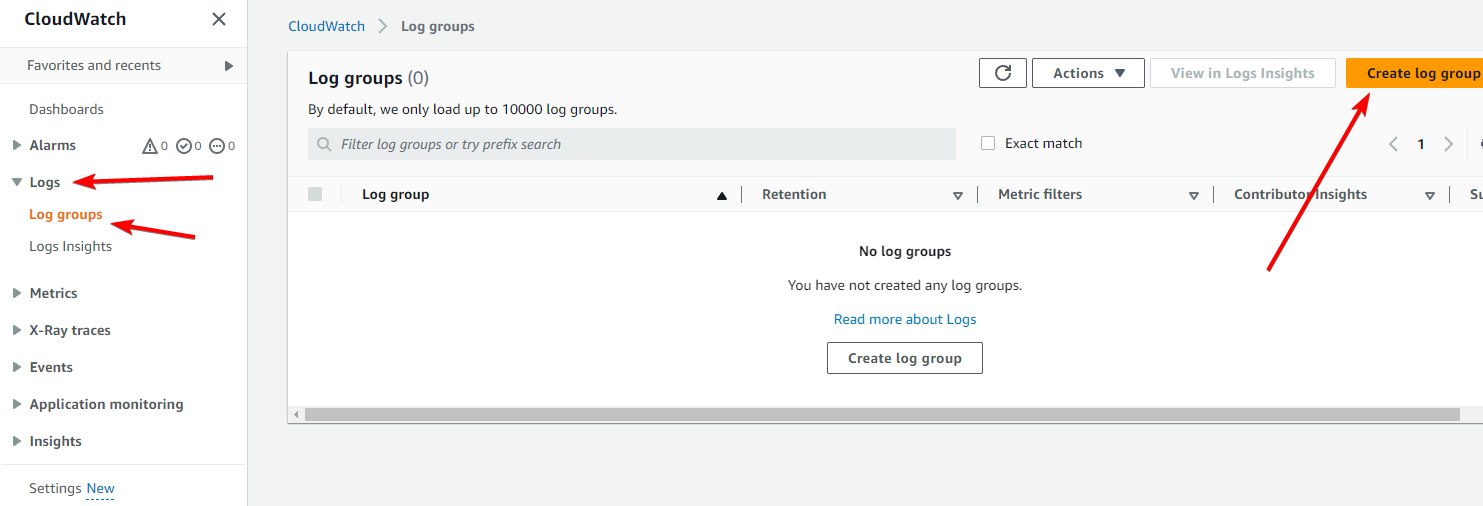
So now we know that outbound ping is working, but we still don’t know where along the path to the other instance the ping packet is being lost or blocked.

Both instances should have been created in the same subnet (availability zone), which will make diagnosing the issue a bit simpler. If the servers were in different networks, in different data centers, on different sides of the world, this could be fairly complicated to troubleshoot because there would a lot more hops (routers), firewalls, and networks, that we would need to take into account.

**Stage 5 - Creating a CloudWatch log group**

Head to the CloudWatch console: [https://ap-southeast-4.console.aws.amazon.com/cloudwatch](https://ap-southeast-4.console.aws.amazon.com/cloudwatch/)

Go to Logs, then Log Groups, then click Create log group

[](https://github.com/saiguda654/vpc_flow_logs_diagnose/blob/main/images/Untitled%2017.png)

Set the Log group name to VPC-Flow-Logs-Demo

Click Create

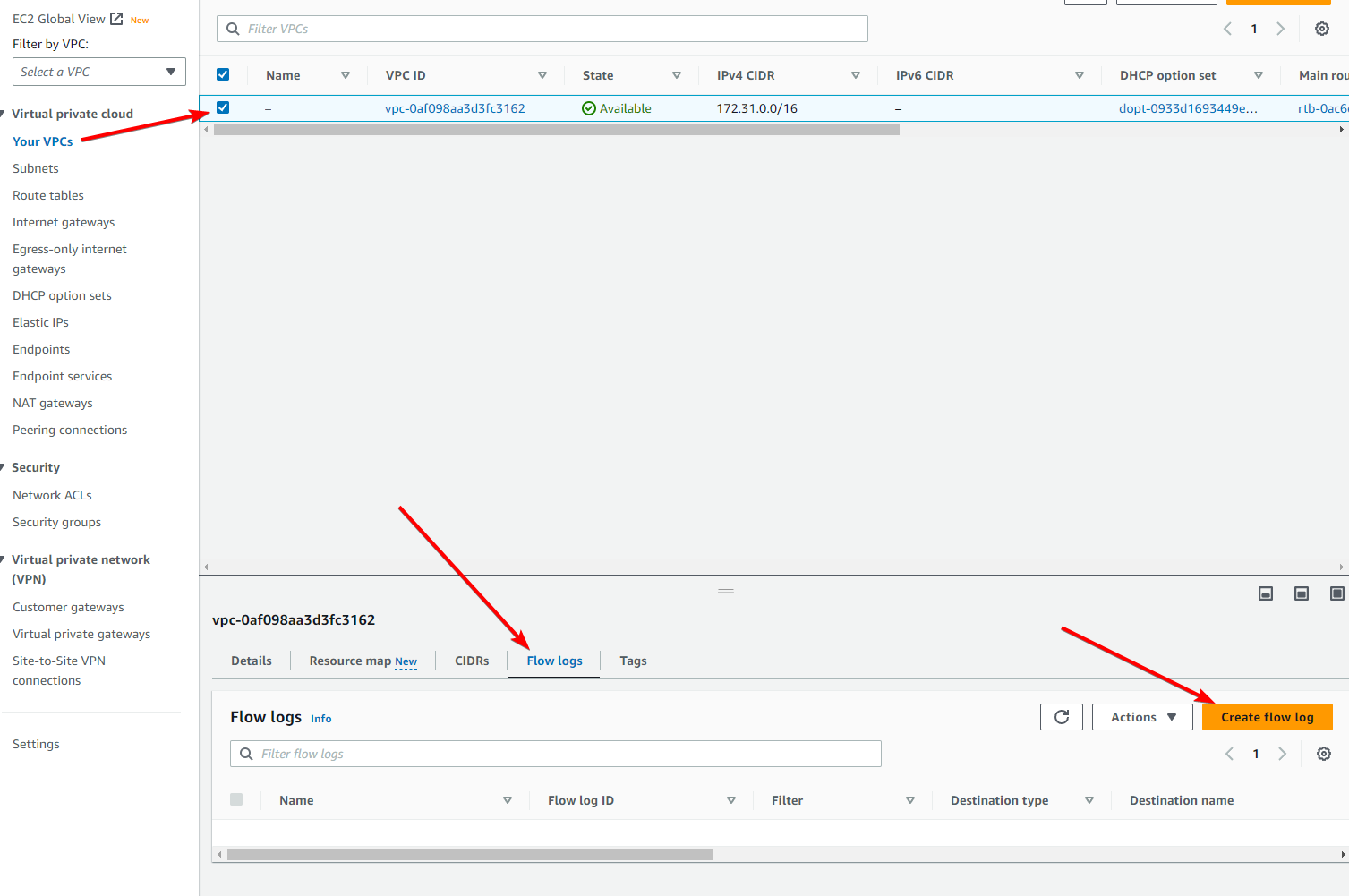
**Stage 6 - Creating a VPC Flow Log**

There are a few places we can create the VPC flow log; the VPC itself, the subnet, or the ENI of an instance. For this demo, because we don’t have any other instances in the VPC, we’re going to put the VPC flow log on the VPC itself.

Head to the VPC console: <https://ap-southeast-4.console.aws.amazon.com/vpc>

Go to “Your VPCs”, select the VPC your instances are deployed in (for me, that’s the default VPC, yours *should* be the same unless you selected a different VPC in step 2)

Go to the “Flow Logs” tab and click Create flow log

[](https://github.com/saiguda654/vpc_flow_logs_diagnose/blob/main/images/Untitled%2018.png)

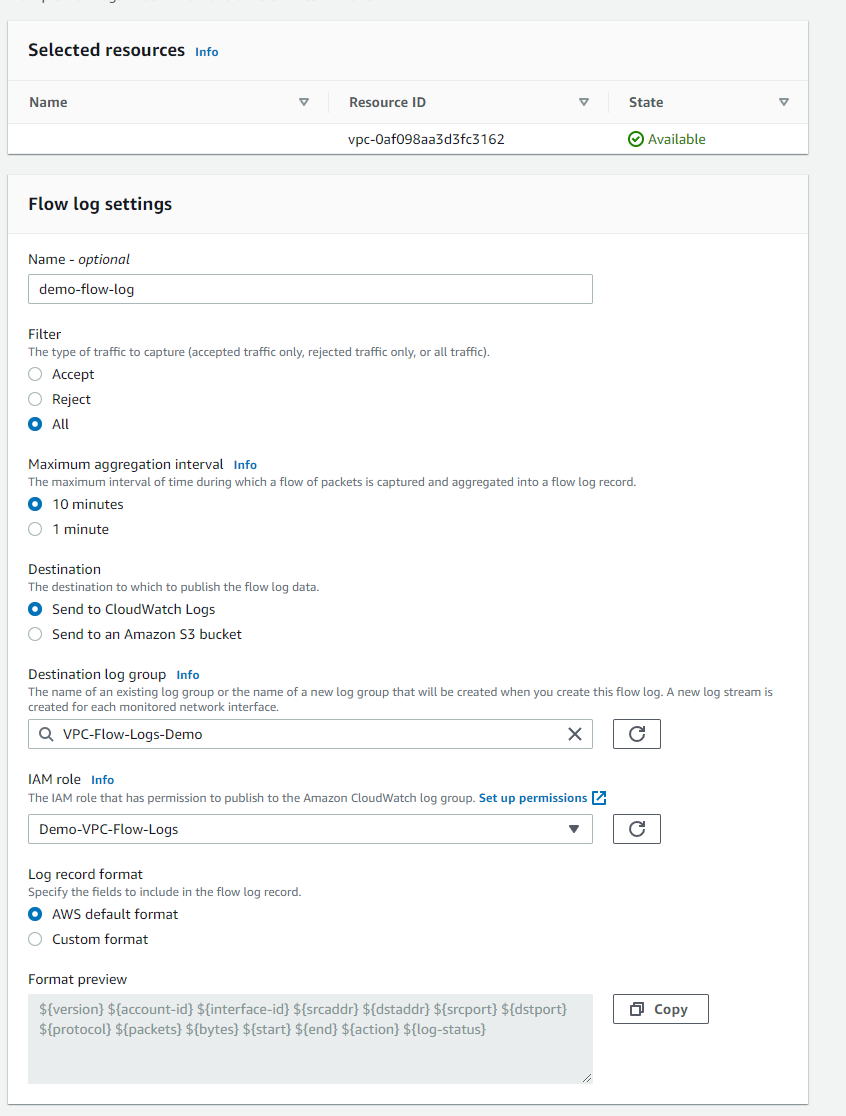
On the next page, set the name of the VPC Flow Log to demo-flow-log

Change “Maximum aggregation interval” to “1 minute”

Under “Destination log group” select the log group you created in stage 5

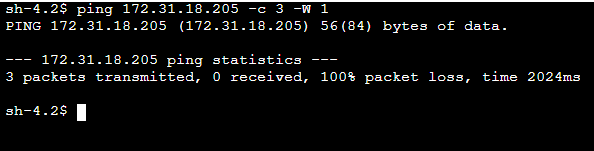
Under “IAM role” select the VPC flow log role you created in stage 1 (Demo-VPC-Flow-Logs)

Your settings should look like this:

[](https://github.com/saiguda654/vpc_flow_logs_diagnose/blob/main/images/Untitled%2019.png)

Click Create flow log

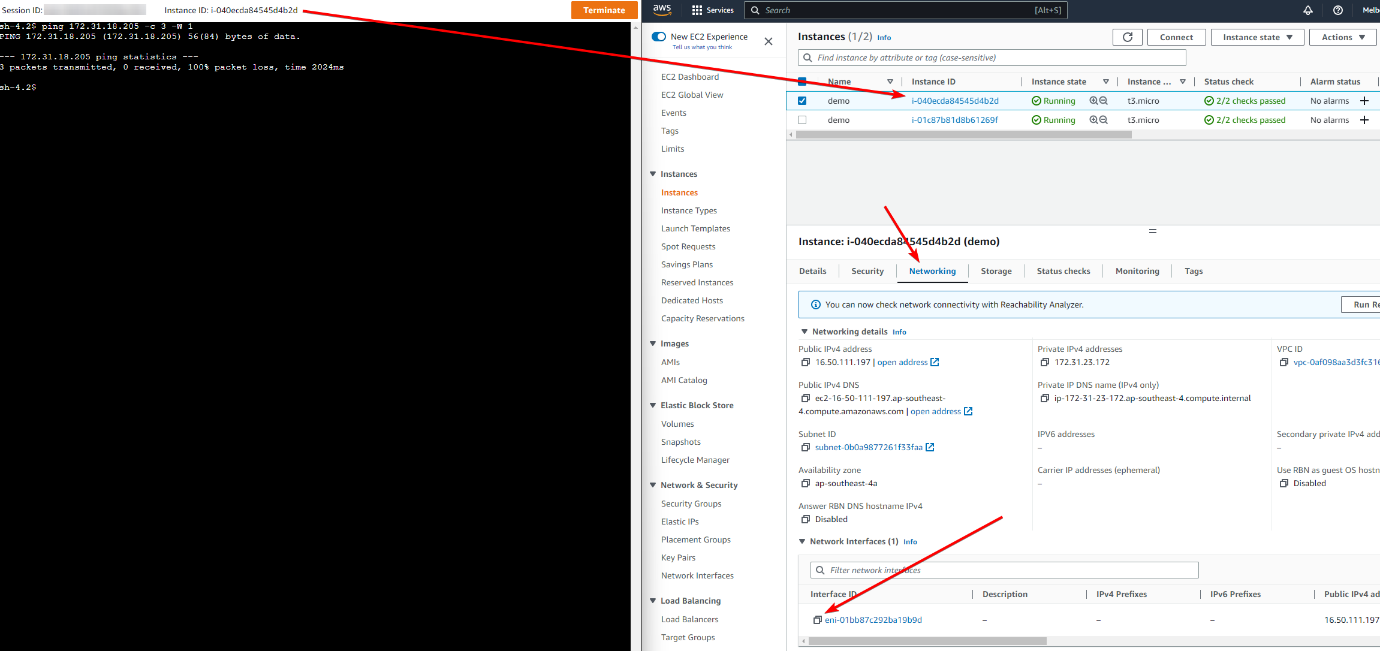
Go back to your SSM Session Manager console (where you were running the ping commands earlier), and try pinging the other instance again

[](https://github.com/saiguda654/vpc_flow_logs_diagnose/blob/main/images/Untitled%2020.png)

For this next step we will need the ENI ID of the instance we are pinging *from*.

Head to the EC2 console: <https://ap-southeast-4.console.aws.amazon.com/ec2/home>

Go to Instances, and select the instance you were running the ping commands from. Go to the “Networking” tab, scroll down to “Network Interfaces” and click the copy icon next to the ENI ID

[](https://github.com/saiguda654/vpc_flow_logs_diagnose/blob/main/images/Untitled%2021.png)

Keep this ID handy for the next step

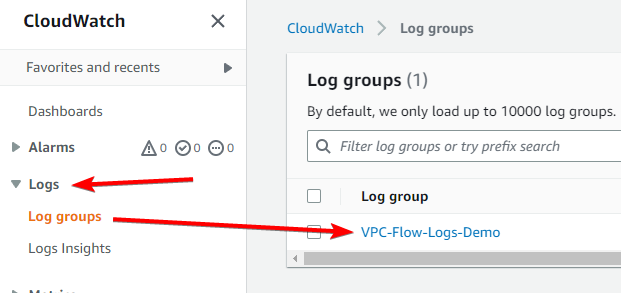
*Tip:* You can also get the ENI ID from the instance itself, using the Instance Metadata Service (IMDS), by running this command:

echo $(curl -s http://169.254.169.254/latest/meta-data/network/interfaces/macs/"$(curl -s http://169.254.169.254/latest/meta-data/network/interfaces/macs)"interface-id)

**Stage 7 - Viewing VPC Flow Logs**

Head to the CloudWatch console: <https://ap-southeast-4.console.aws.amazon.com/cloudwatch/home>

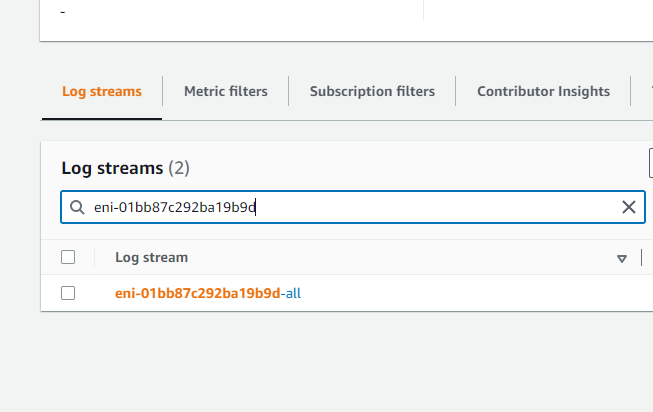
Go to Logs, then Log Groups, then click on the [VPC-Flow-Logs-Demo](https://ap-southeast-4.console.aws.amazon.com/cloudwatch/home?region=ap-southeast-4#logsV2:log-groups/log-group/VPC-Flow-Logs-Demo) log group

[](https://github.com/saiguda654/vpc_flow_logs_diagnose/blob/main/images/Untitled%2022.png)

If you don’t see any log streams, try refreshing in 1-2 minutes. Remember we set the aggregation level to 1 minute in the VPC Flow Logs settings, meaning VPC will send flow logs to CloudWatch every minute.

You will see there’s log streams for each ENI that has sent traffic on the VPC, in my case, that’s one per instance, so two. In a production environment, enabling VPC Flow Logs on the VPC itself might yield hundreds or thousands of ENI streams.

Paste the ENI ID you copied in the previous stage into the Log Stream search

[](https://github.com/saiguda654/vpc_flow_logs_diagnose/blob/main/images/Untitled%2023.png)

Click on the log stream, and you will (most likely) see a huge output of confusing looking log entries. These are all of the connection flows that have gone through this ENI, so that will include the ping attempts we made, but also any other traffic such as DNS lookups, bots / IP scanners from the internet attempting to connect to our external IP, session manager traffic, etc.

The format of the log entries can also be a bit daunting, you can read what each column means here: <https://docs.aws.amazon.com/vpc/latest/userguide/flow-logs.html#flow-logs-default>

But as a quick summary, for an example log entry like this:

2 1234567890123 eni-01ed204d4ab726d6d 70.232.69.76 172.31.26.160 443 57588 6 6 614 1678845342 1678845368 ACCEPT OK

2 is the VPC Flow Log version

1234567890123 is your AWS account ID

eni-01ed204d4ab726d6d is the ENI ID

70.232.69.76 is the source IP

172.31.26.160 is the destination IP (note it doesn’t show your public IP, because that has already been translated to your internal IP)

443 is the source port

57588 is the destination port

6 is the protocol (6 is TCP, 1 is ICMP, 17 is UDP. You can view the protocols here: <https://www.iana.org/assignments/protocol-numbers/protocol-numbers.xhtml>)

6 is the number of packets transferred in this flow

614 is the number of bytes transferred in this flow

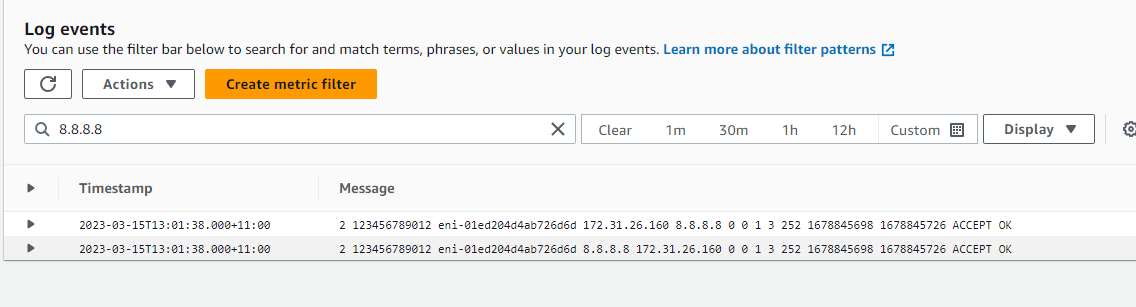
1678845342 is the start time in Unix EPOCH time

1678845368 is the end time in Unix EPOCH time

ACCEPT shows that the traffic was accepted (not blocked by an ACL or security group)

OK is the log status, showing that the flow was logged successfully.

Let’s see if we can find our (working) ping packets to 8.8.8.8 from earlier. In the search bar up the top of the page, enter 8.8.8.8 and hit enter.

[](https://github.com/saiguda654/vpc_flow_logs_diagnose/blob/main/images/Untitled%2024.png)

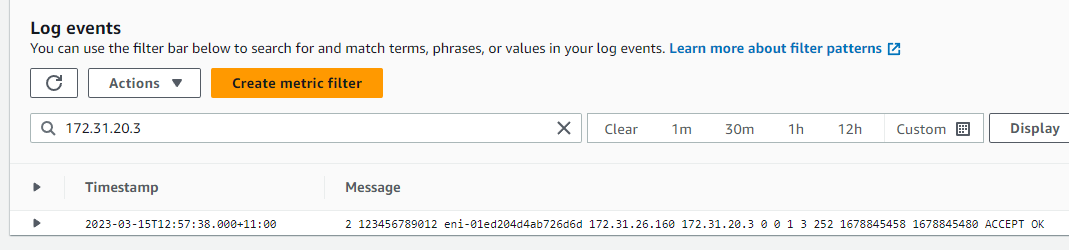
So, you can see the sent packet, of protocol 1 which is ICMP, and the action was ACCEPT

2 123456789012 eni-01ed204d4ab726d6d 172.31.26.160 8.8.8.8 0 0 1 3 252 1678845698 1678845726 ACCEPT OK

Then you can see the return packet which was also accepted

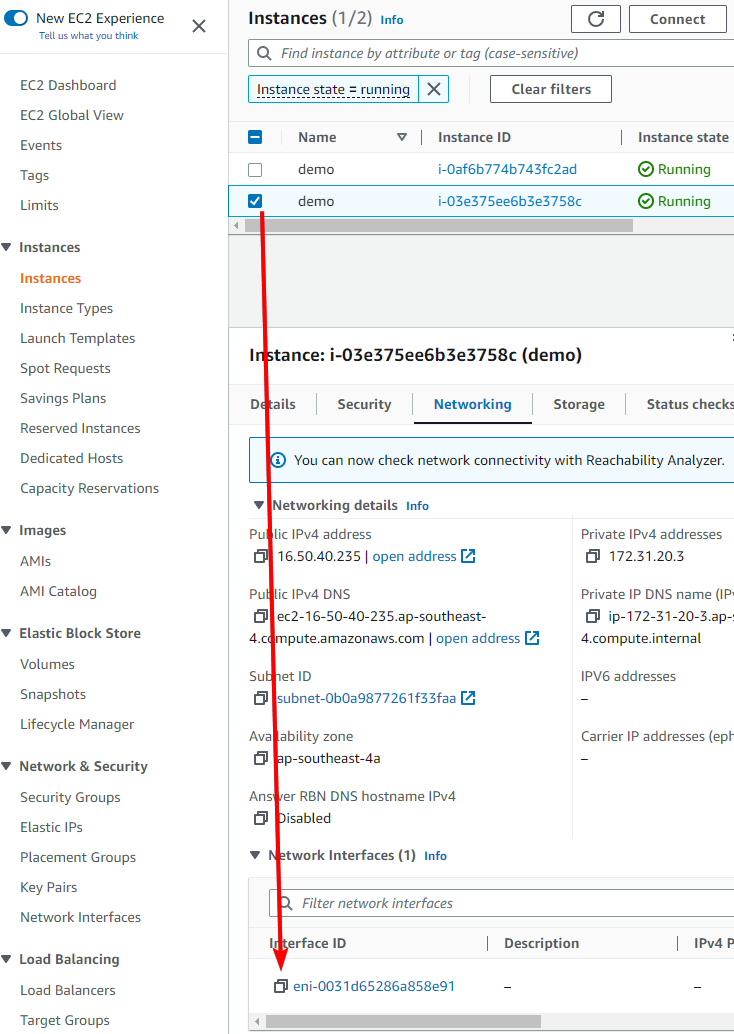
2 123456789012 eni-01ed204d4ab726d6d 8.8.8.8 172.31.26.160 0 0 1 3 252 1678845698 1678845726 ACCEPT OK

Let’s look for our broken pings now, search for the IP address you were pinging (the destination instance)

[](https://github.com/saiguda654/vpc_flow_logs_diagnose/blob/main/images/Untitled%2025.png)

We can see it was accepted, but there were no response packets, indicating the packet was blocked (or lost) elsewhere, but this tells us the security group on *this* ENI isn’t the problem.

Let’s check the ENI of the destination instance. Again, grab the ENI ID of the *destination* instance from the EC2 console

[](https://github.com/saiguda654/vpc_flow_logs_diagnose/blob/main/images/Untitled%2026.png)

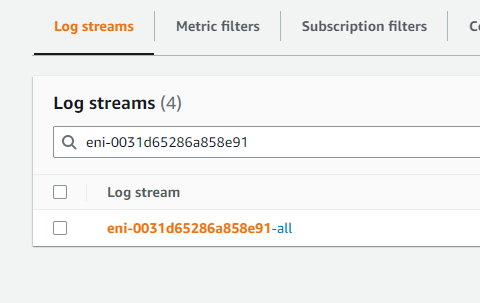
or by running the following command on the *destination* instance

echo $(curl -s http://169.254.169.254/latest/meta-data/network/interfaces/macs/"$(curl -s http://169.254.169.254/latest/meta-data/network/interfaces/macs)"interface-id)

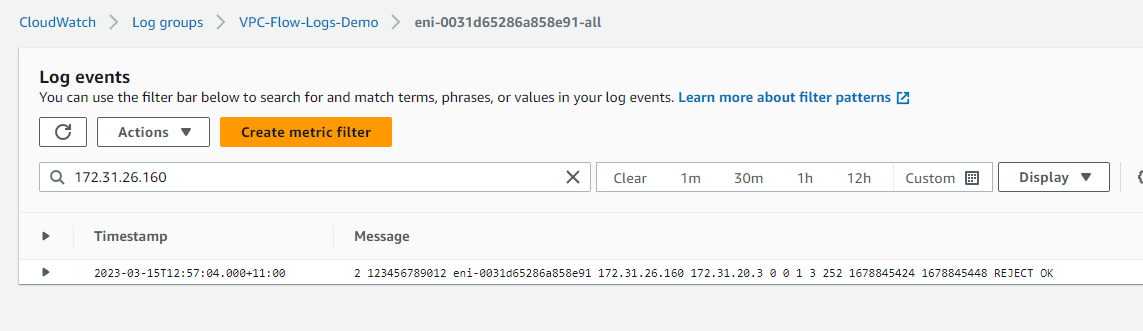
Head back to the CloudWatch console: <https://ap-southeast-4.console.aws.amazon.com/cloudwatch>

Go to Logs, then Log Groups, then click on the [VPC-Flow-Logs-Demo](https://ap-southeast-4.console.aws.amazon.com/cloudwatch/home?region=ap-southeast-4#logsV2:log-groups/log-group/VPC-Flow-Logs-Demo) log group

Then search for the ENI ID of the destination instance

[](https://github.com/saiguda654/vpc_flow_logs_diagnose/blob/main/images/Untitled%2027.png)

Search for the IP address of the *source* instance, and we can see the problem

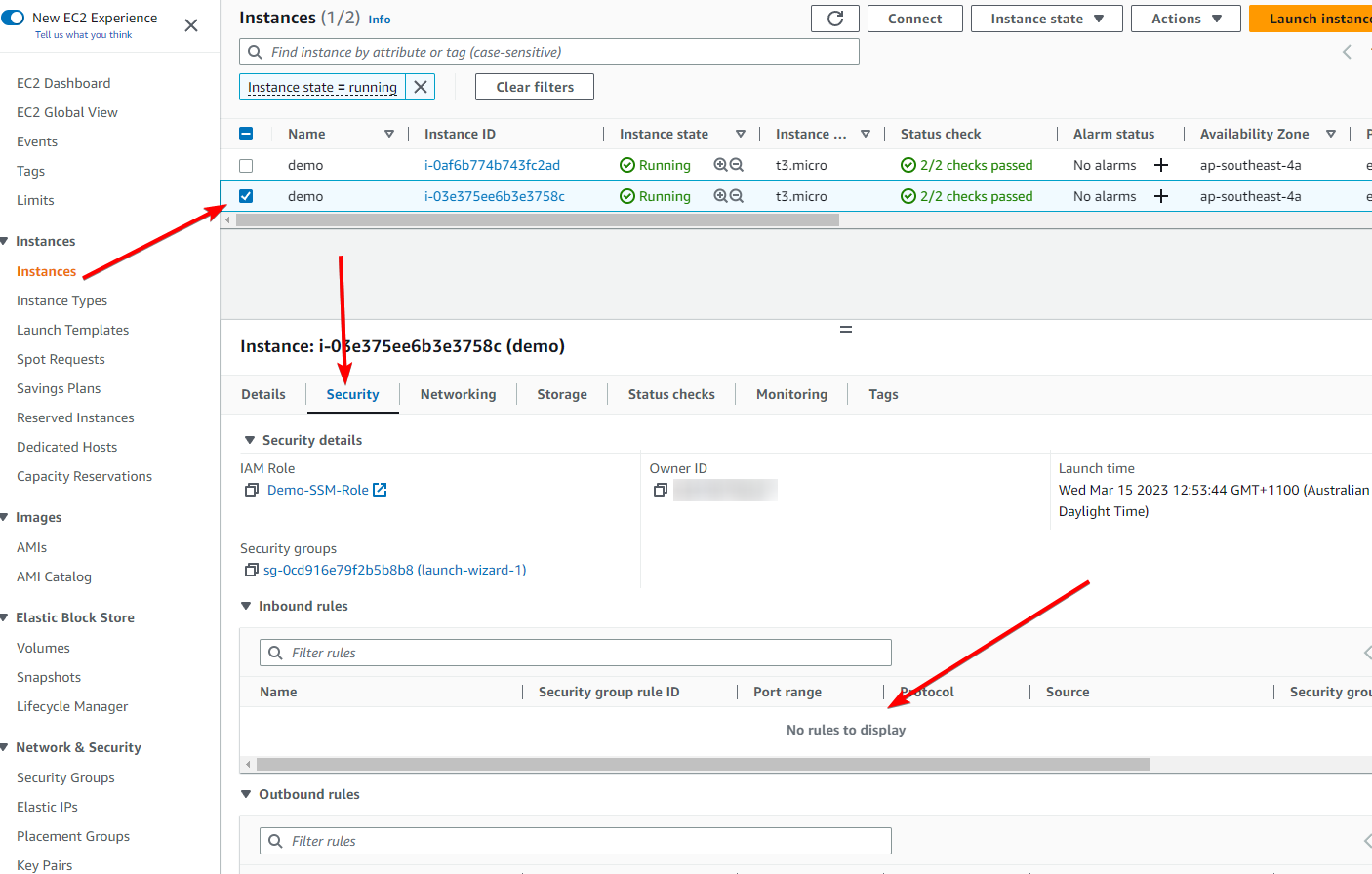
[](https://github.com/saiguda654/vpc_flow_logs_diagnose/blob/main/images/Untitled%2028.png)

It’s being rejected at the destination ENI. Because we know both of these instances are in the same subnet, that rules out the issue being a Network ACL, and is very likely going to be a security group rule.

**Stage 8 - Resolving the network issue**

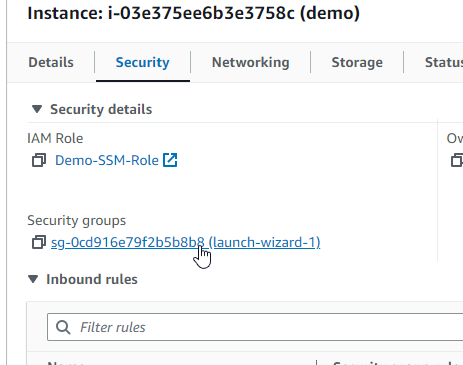
Head to the EC2 console: <https://ap-southeast-4.console.aws.amazon.com/ec2/home>

Go to Instances and select the *destination* instance, then go to the “Security” tab

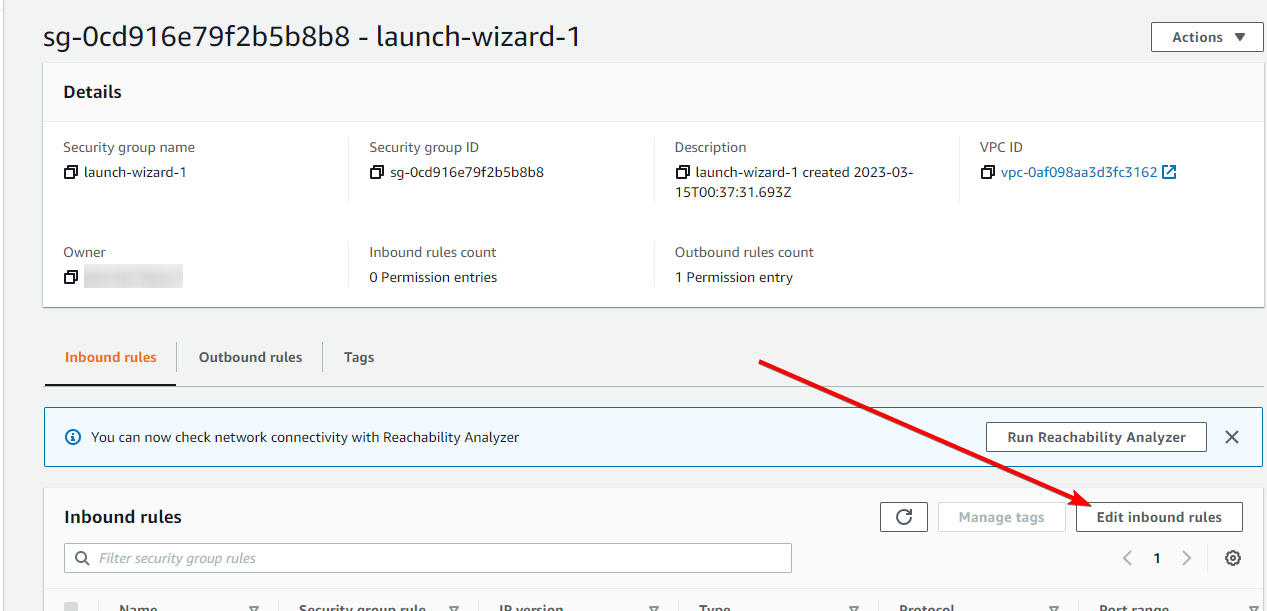
[](https://github.com/saiguda654/vpc_flow_logs_diagnose/blob/main/images/Untitled%2029.png)

You can see the security group attached to this instance (and therefore the primary ENI) doesn’t have *any* inbound rules, meaning *all* new connections will be blocked. I say “new connections” because, as mentioned in Adrian’s courses, security groups are stateful, meaning if you make an outbound connection, any related return connections will be *allowed*.

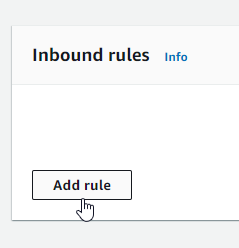
Let’s fix this up, click on the security group

[](https://github.com/saiguda654/vpc_flow_logs_diagnose/blob/main/images/Untitled%2030.png)

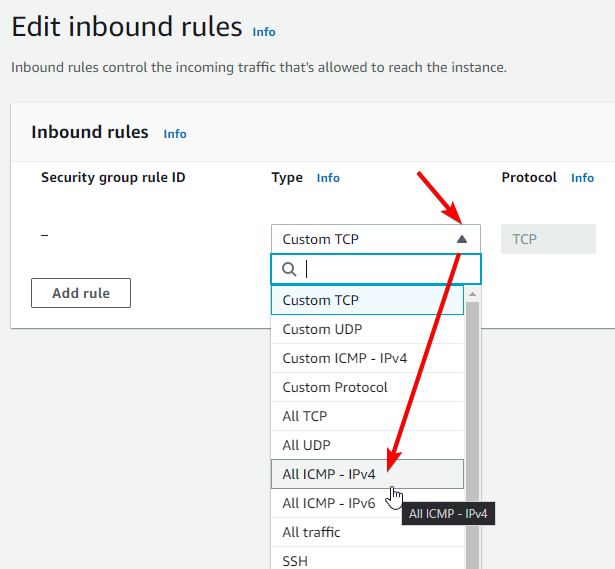
Then click on Edit inbound rules

[](https://github.com/saiguda654/vpc_flow_logs_diagnose/blob/main/images/Untitled%2031.png)

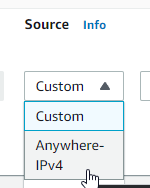
Click Add rule

[](https://github.com/saiguda654/vpc_flow_logs_diagnose/blob/main/images/Untitled%2032.png)

Under “Type” select “All ICMP - IPv4”

[](https://github.com/saiguda654/vpc_flow_logs_diagnose/blob/main/images/Untitled%2033.png)

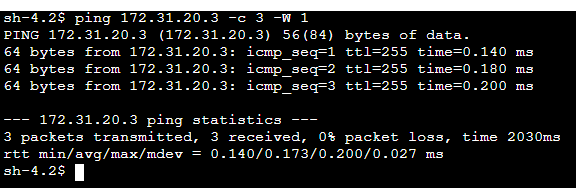
Change “Source” to “Anywhere-IPv4”

[](https://github.com/saiguda654/vpc_flow_logs_diagnose/blob/main/images/Untitled%2034.png)

Then click Save rules

We’ve just allowed all ICMP traffic, which is what ping uses (it doesn’t by default use TCP or UDP), to any ENI that has this security group attached.

If we go back to our session manager console on our *source* instance, we can see ping is now working

[](https://github.com/saiguda654/vpc_flow_logs_diagnose/blob/main/images/Untitled%2035.png)

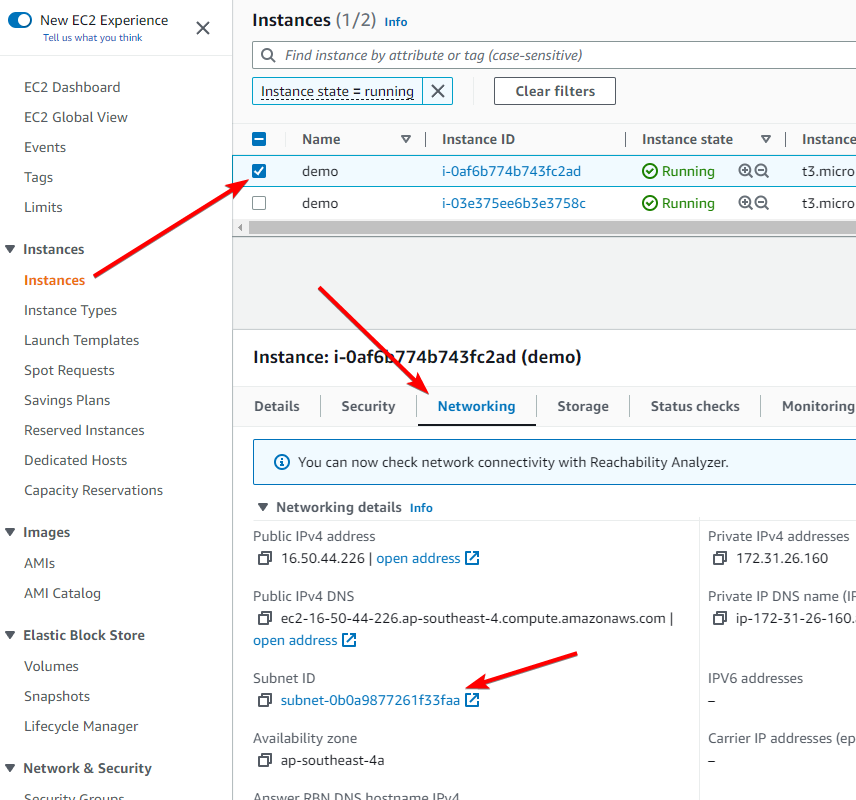
Thanks to VPC Flow Logs we found the location of the issue and resolved it 🙌

**Stage 9 - Bonus: Network ACLs and VPC Flow Logs**

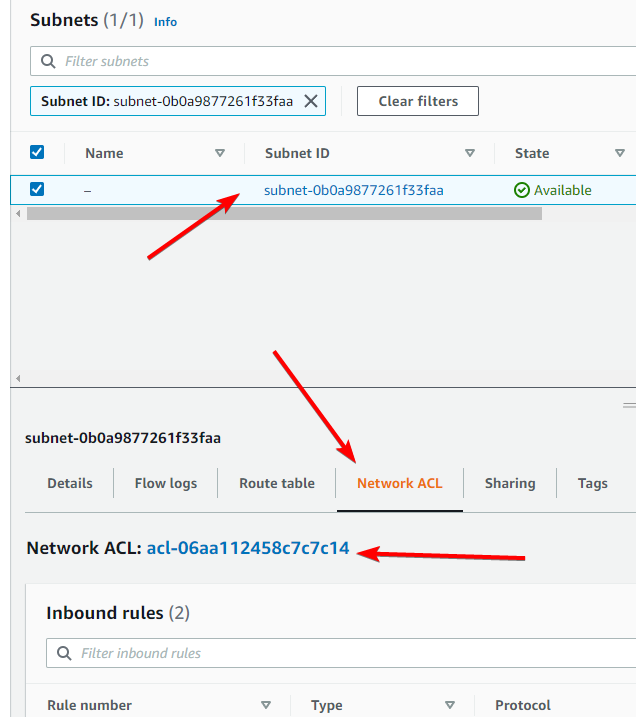
Let’s see what happens if block ICMP at the Network ACL level

Head to the EC2 console: <https://ap-southeast-4.console.aws.amazon.com/ec2/home>

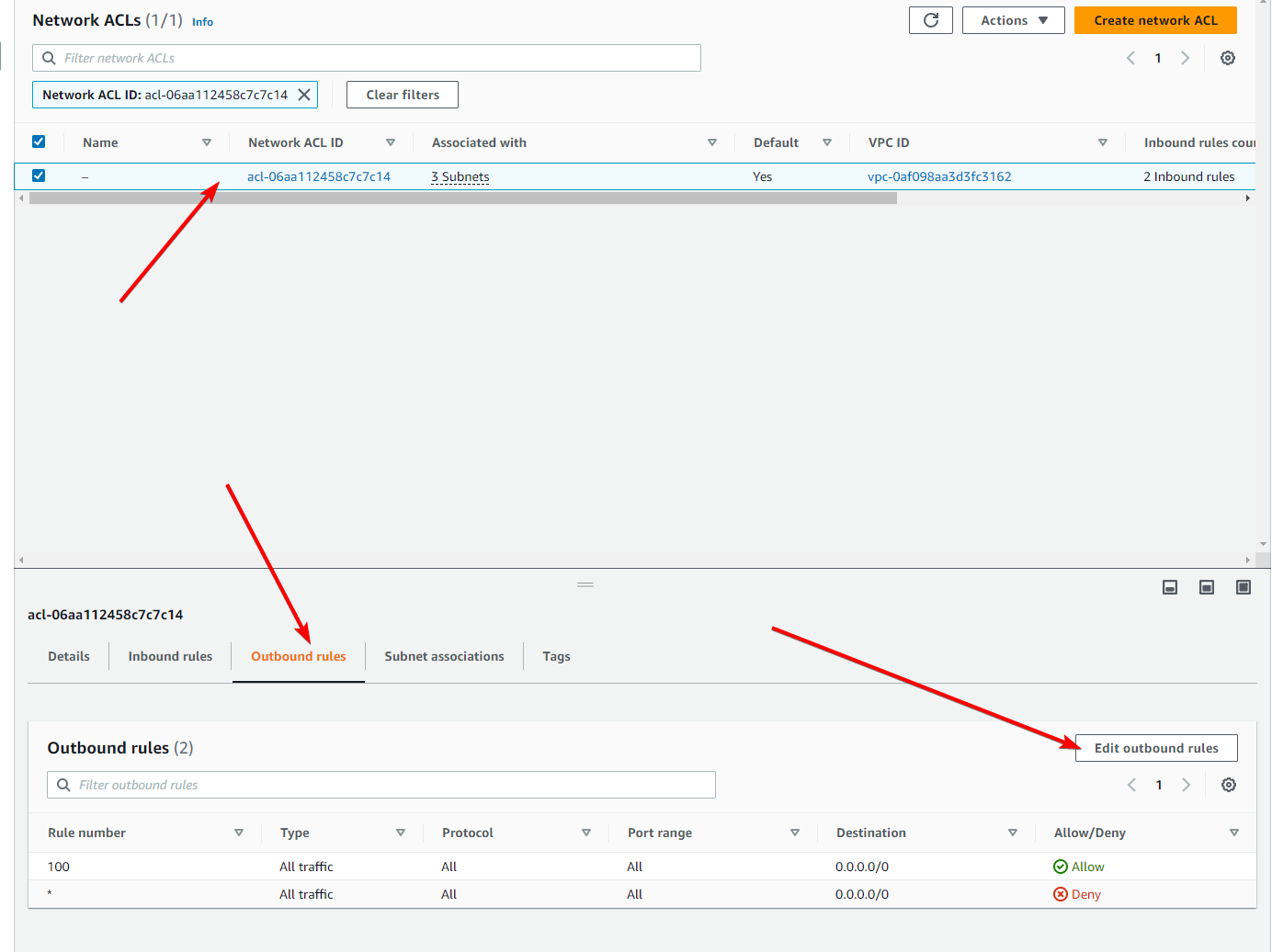
Go to Instances and select the *source* instance, then go to the “Networking” tab and click on the subnet

[](https://github.com/saiguda654/vpc_flow_logs_diagnose/blob/main/images/Untitled%2036.png)

Click on the subnet, go to the “Network ACL” tab, and click on the ACL ID

[](https://github.com/saiguda654/vpc_flow_logs_diagnose/blob/main/images/Untitled%2037.png)

Click on the network ACL, go to the “Outbound rules” tab, and click Edit outbound rules

[](https://github.com/saiguda654/vpc_flow_logs_diagnose/blob/main/images/Untitled%2038.png)

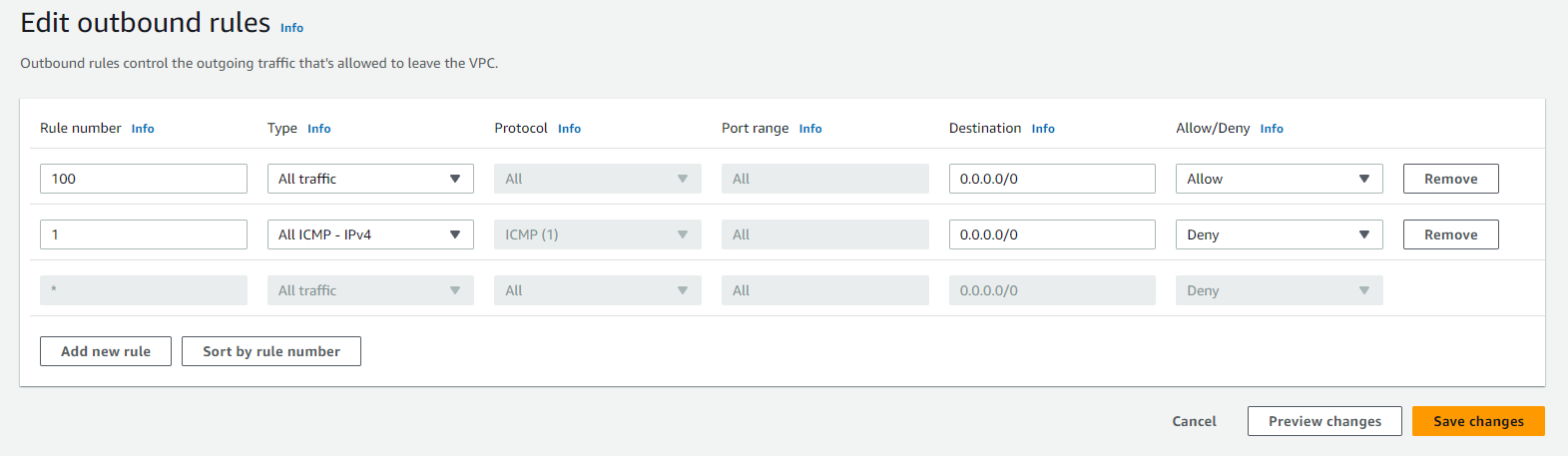
You will see that by default, network ACLs allow everything in and out.

Click on Add new rule

Enter the rule number as 1, Network ACL rules are executed in order of rule number, lowest to highest, so if you were to set the rule number to 101 it would be *behind* the allow all rule, and would be ignored.

Set the “Type” to “All ICMP - IPv4”

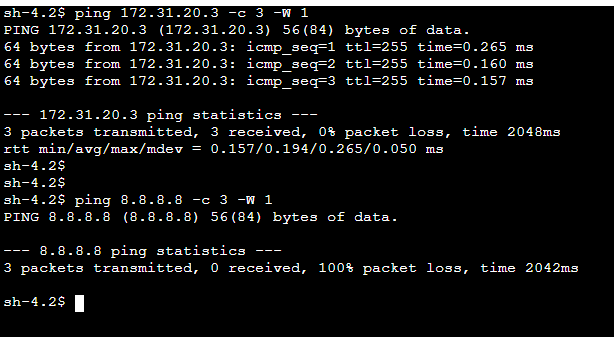
Change the “Allow/Deny” to “Deny”

[](https://github.com/saiguda654/vpc_flow_logs_diagnose/blob/main/images/Untitled%2039.png)

Click Save changes

Head back to the session manager window of the source instance, and try pinging the destination instance. You should see it still work, because it’s in the same subnet and therefore isn’t affected by the network ACL.

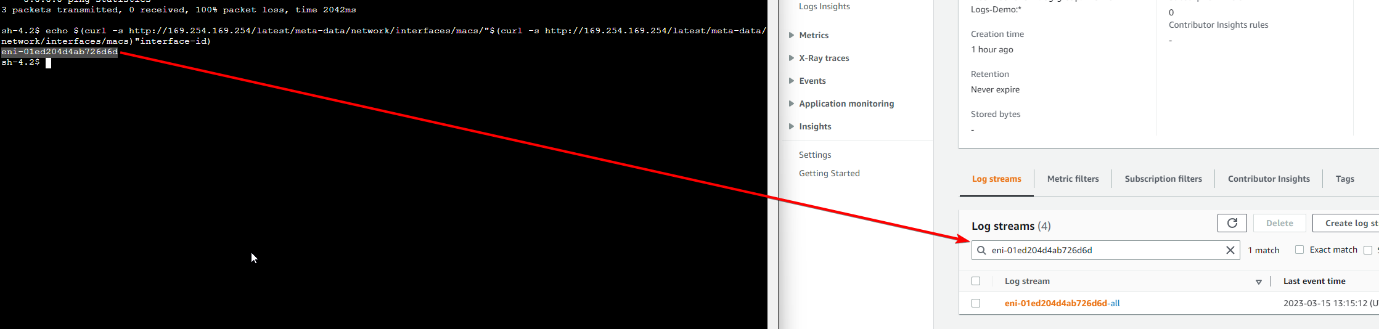
If you try pinging an external IP like 8.8.8.8 however, you should see it fail

[](https://github.com/saiguda654/vpc_flow_logs_diagnose/blob/main/images/Untitled%2040.png)

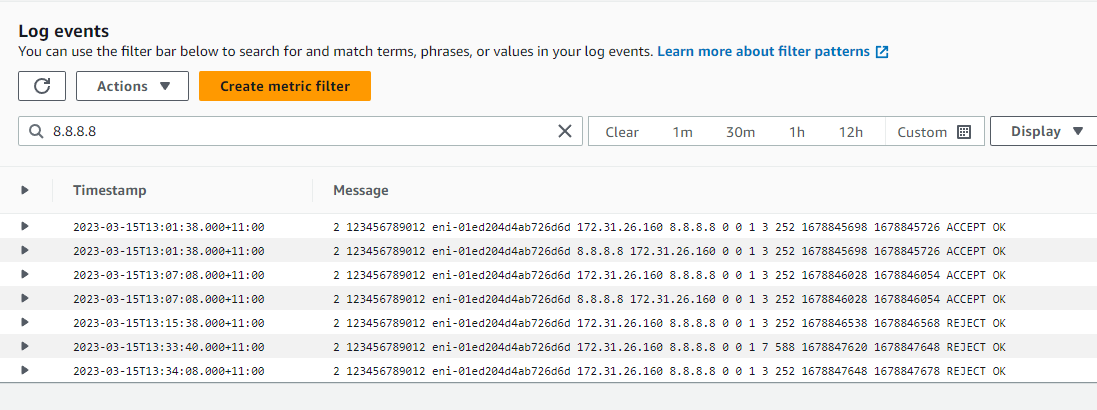
Head to the CloudWatch console: <https://ap-southeast-4.console.aws.amazon.com/cloudwatch/home>

Go to Logs, then Log Groups, then click on the [VPC-Flow-Logs-Demo](https://ap-southeast-4.console.aws.amazon.com/cloudwatch/home?region=ap-southeast-4#logsV2:log-groups/log-group/VPC-Flow-Logs-Demo) log group

Enter the ENI ID of the instance you were just using to ping outbound

[](https://github.com/saiguda654/vpc_flow_logs_diagnose/blob/main/images/Untitled%2041.png)

And we can see the ICMP packets were rejected

[](https://github.com/saiguda654/vpc_flow_logs_diagnose/blob/main/images/Untitled%2042.png)

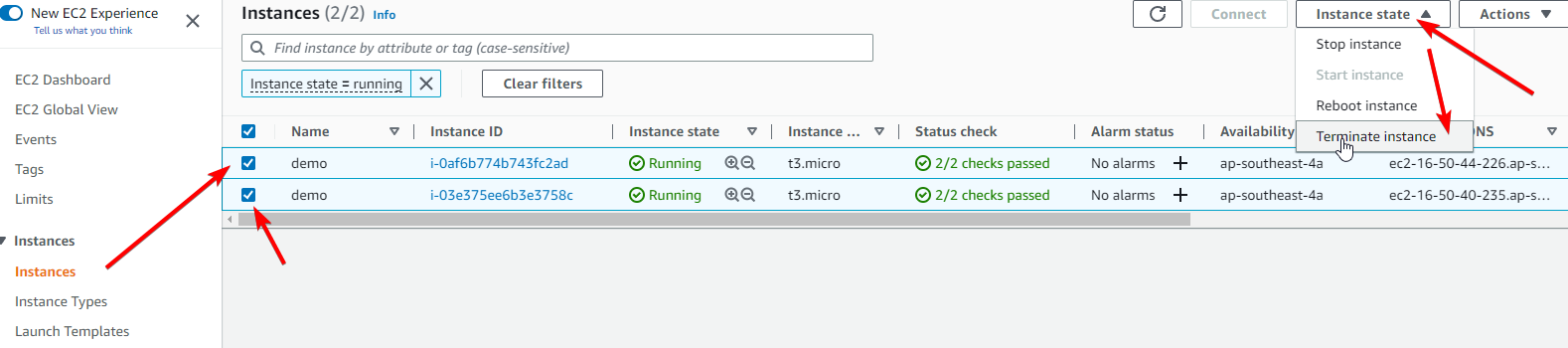
Unfortunately, there’s no way to tell based on these logs whether it’s the security group, or the network ACL that is rejecting the traffic, but this at least gives you an idea on where to look.

**Stage 10 - Clean up**

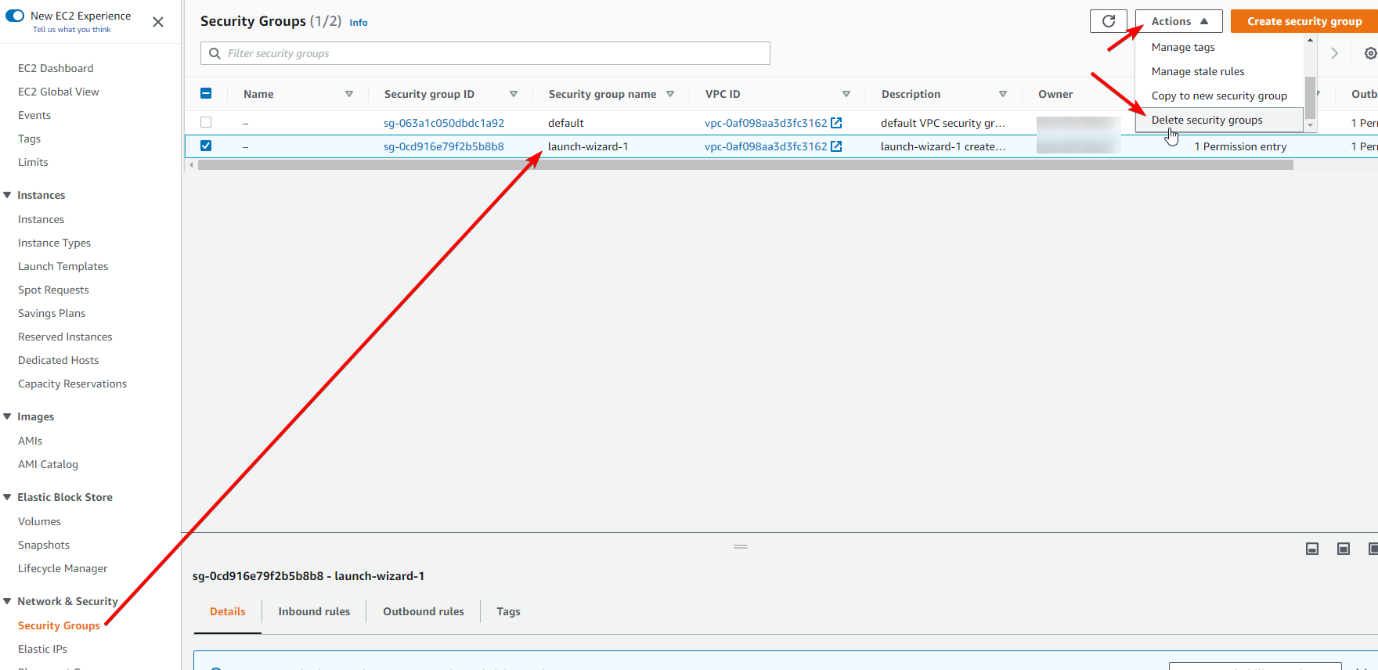
Head to the EC2 console: <https://ap-northeast-1.console.aws.amazon.com/ec2/v2/home>

Go to *Instances*, and select both “demo” instances, then click Instance state and then Terminate Instance

Be careful to only delete instances created in this demo

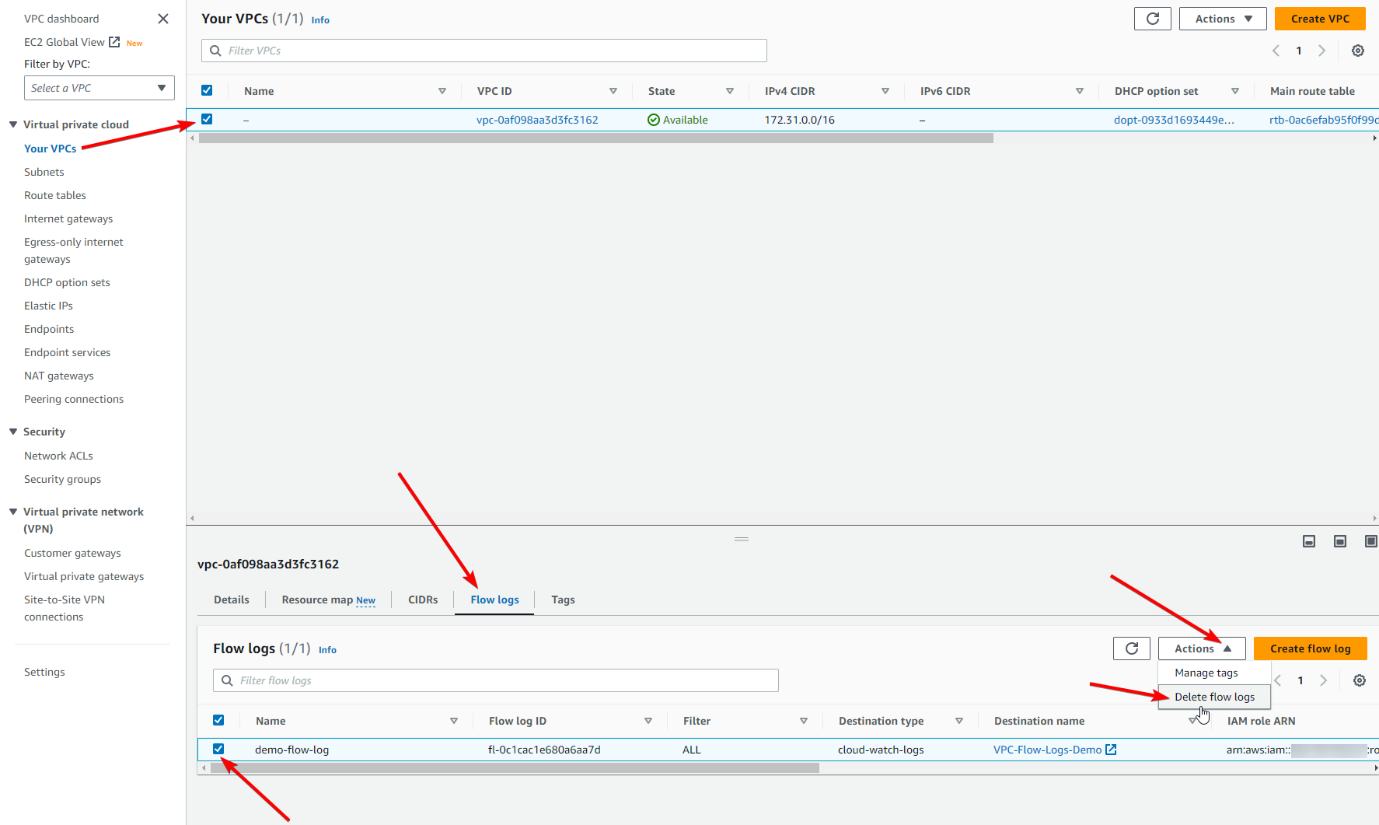
[](https://github.com/saiguda654/vpc_flow_logs_diagnose/blob/main/images/Untitled%2043.png)

Go to *Security Groups* and select “launch-wizard-1” which was created back in stage 2, then click Actions then Delete security groups

[](https://github.com/saiguda654/vpc_flow_logs_diagnose/blob/main/images/Untitled%2044.png)

Head to the VPC console: <https://ap-southeast-4.console.aws.amazon.com/vpc/home>

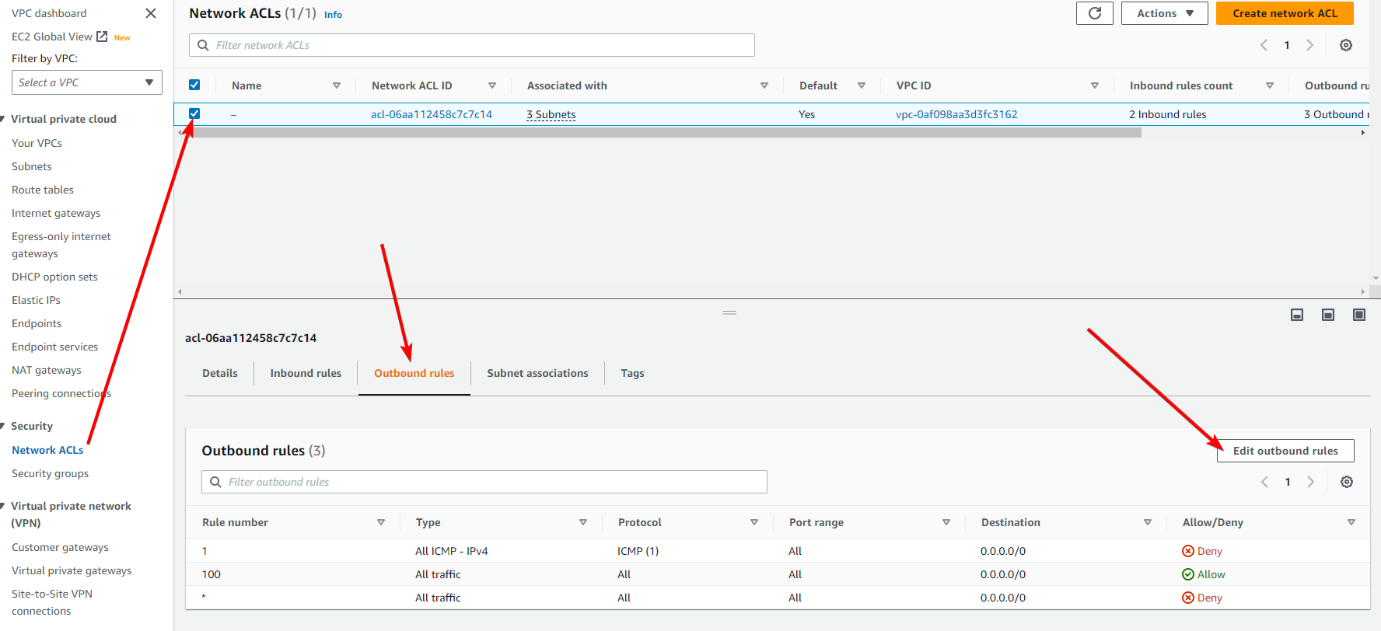
Go to *Your VPCs*, select the default VPC we used in stage 2, go to the *Flow Logs* tab, select the flow log we created, then click Actions then Delete flow logs

[](https://github.com/saiguda654/vpc_flow_logs_diagnose/blob/main/images/Untitled%2045.png)

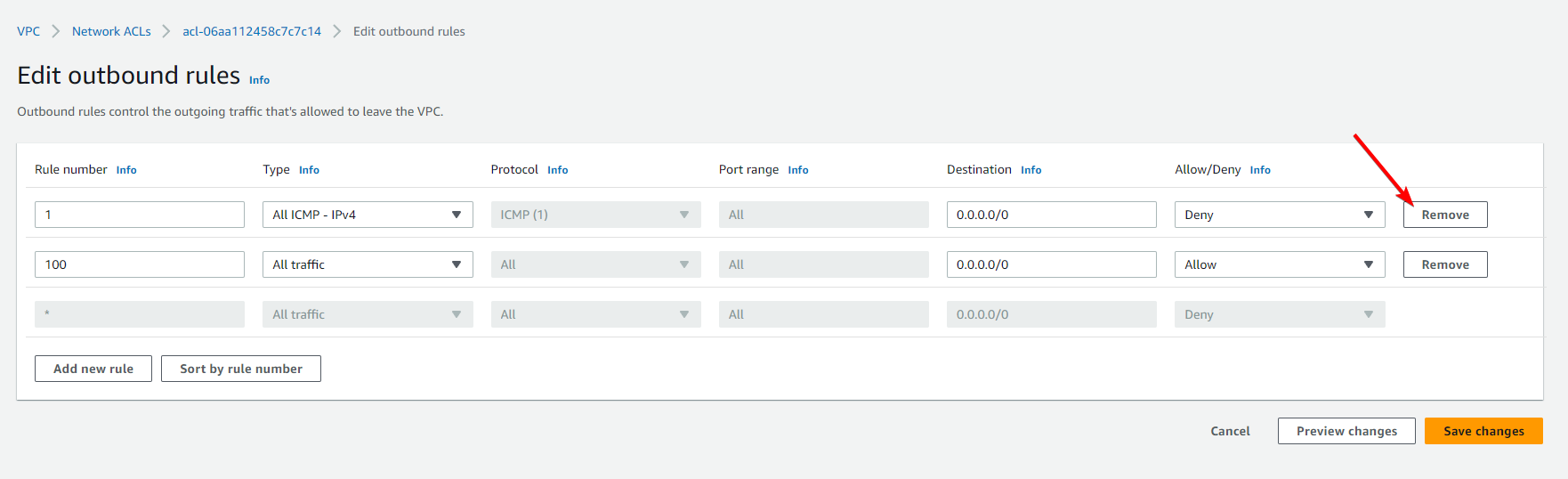
Type “delete” in the confirmation box and click Delete

*If* you did the stage 8 steps:

Go to *Network ACLs*, select the network ACL you modified (there is likely only one), go to *Outbound rules* and click Edit outbound rules

[](https://github.com/saiguda654/vpc_flow_logs_diagnose/blob/main/images/Untitled%2046.png)

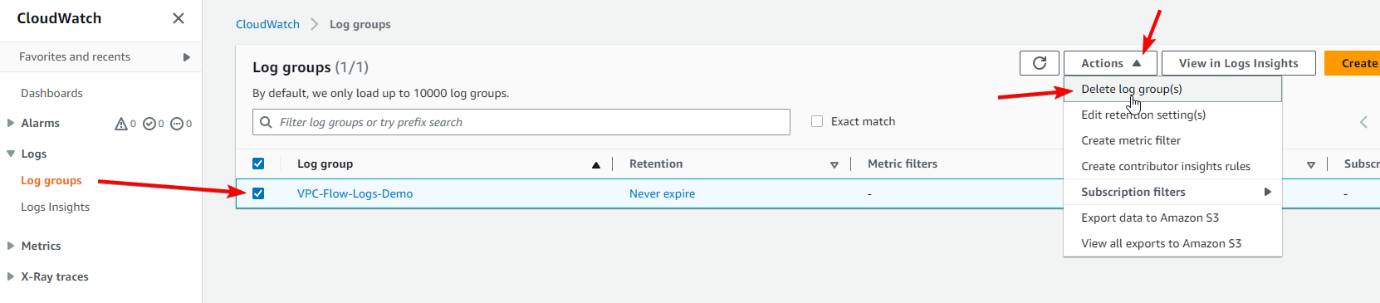
Click Remove next to the rule denying ICMP traffic

[](https://github.com/saiguda654/vpc_flow_logs_diagnose/blob/main/images/Untitled%2047.png)

Be careful *not* to delete rule 100 which allows all traffic. Click Save changes

Head to the CloudWatch console: <https://ap-southeast-4.console.aws.amazon.com/cloudwatch/home>

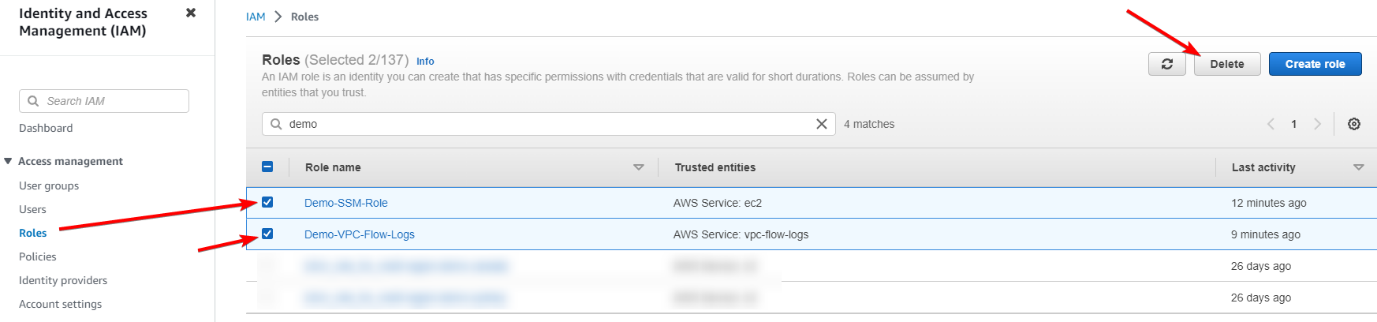
Go to *Logs*, then *Log Groups*, then select the VPC-Flow-Logs-Demo log group, click Actionsthen Delete log group(s)

[](https://github.com/saiguda654/vpc_flow_logs_diagnose/blob/main/images/Untitled%2048.png)

Click Delete in the confirmation box

Head to the IAM console: <https://us-east-1.console.aws.amazon.com/iamv2/home>

Go to *Roles*, and search for “demo”. Select the two roles we created in stage 1, and click Delete

[](https://github.com/saiguda654/vpc_flow_logs_diagnose/blob/main/images/Untitled%2049.png)

Enter “delete” in the confirmation box, and click Delete