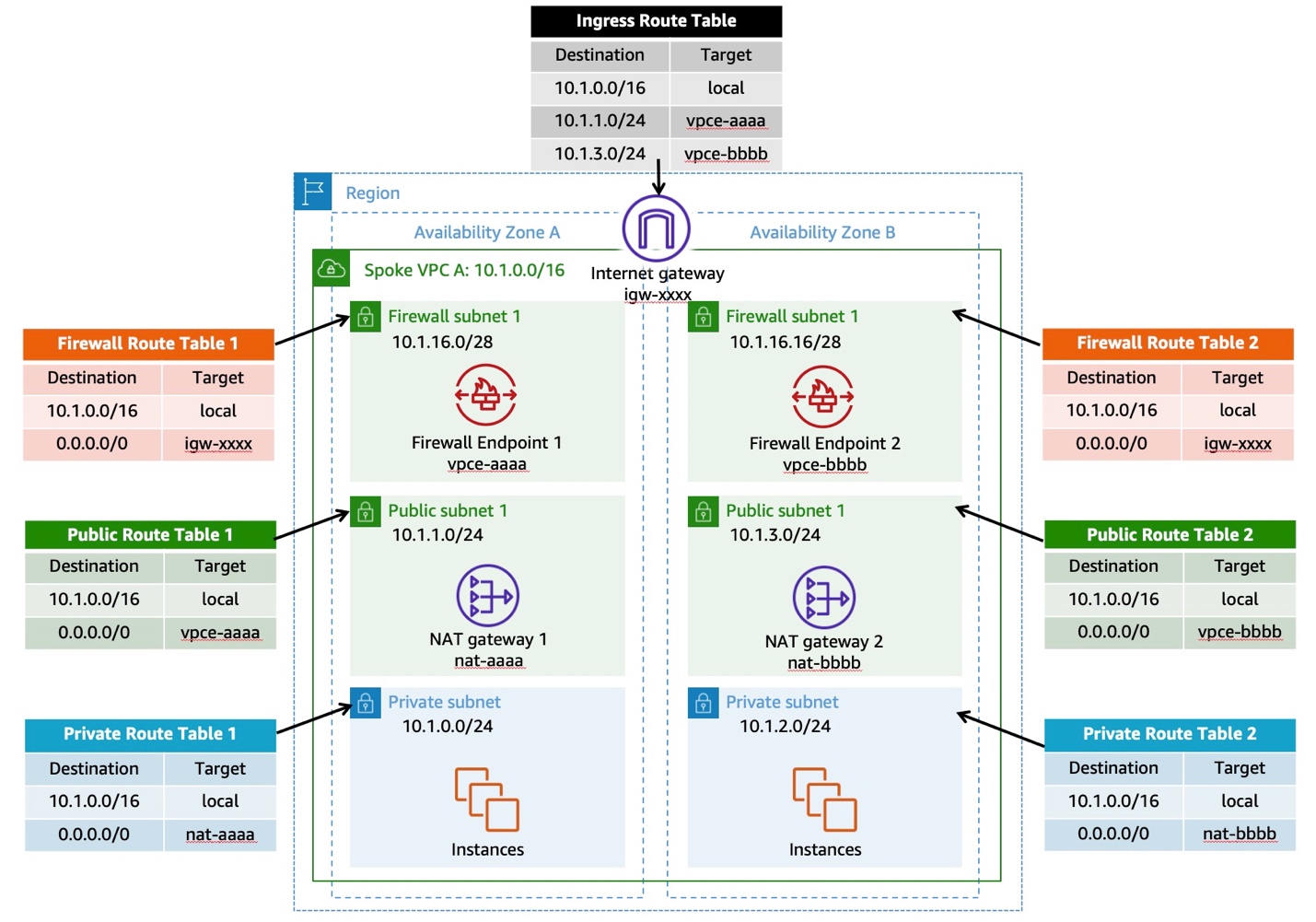
DISTRIBUTED DEPLOYMENT MODEL

For the distributed deployment model, we deploy AWS Network Firewall into each VPC which requires protection. Each VPC is protected individually and blast radius is reduced through VPC isolation. Each VPC does not require connectivity to any other VPC or AWS Transit Gateway. Each AWS Network Firewall can have its own firewall policy or share a policy through common rule groups (reusable collections of rules) across multiple firewalls. This allows each AWS Network Firewall to be managed independently, which reduces the possibility of misconfiguration and limits the scope of impact. Following use cases are covered:

* Protect traffic between a workload in public subnet (e.g. EC2 Instance) and the Internet
* Protect/Filter traffic between an AWS service (e.g. ALB, NLB) in a public subnet and the Internet

As shown in the *Figure 1: Distributed Architecture*, AWS Network Firewall is deployed in a dedicated Firewall subnet which has access to Internet Gateway (IGW). Returning traffic from IGW is returned back to Firewall Elastic Network Interface (ENI) using Ingress Routing table attached to the IGW. This ensures the traffic is symmetric for full inspection. NAT Gateway deployed in dedicated public subnet allows instances in private subnet to communicated with resources on the Internet.

[](https://networkfirewall.workshop.aws/images/setup/distributed_model/anfw_distributed_2az.jpg)

*Figure 1: Distributed Architecture*

# DEPLOY RESOURCES

To deploy resources , follow below steps:

##### **Step 1 : Download CloudFormation template:**

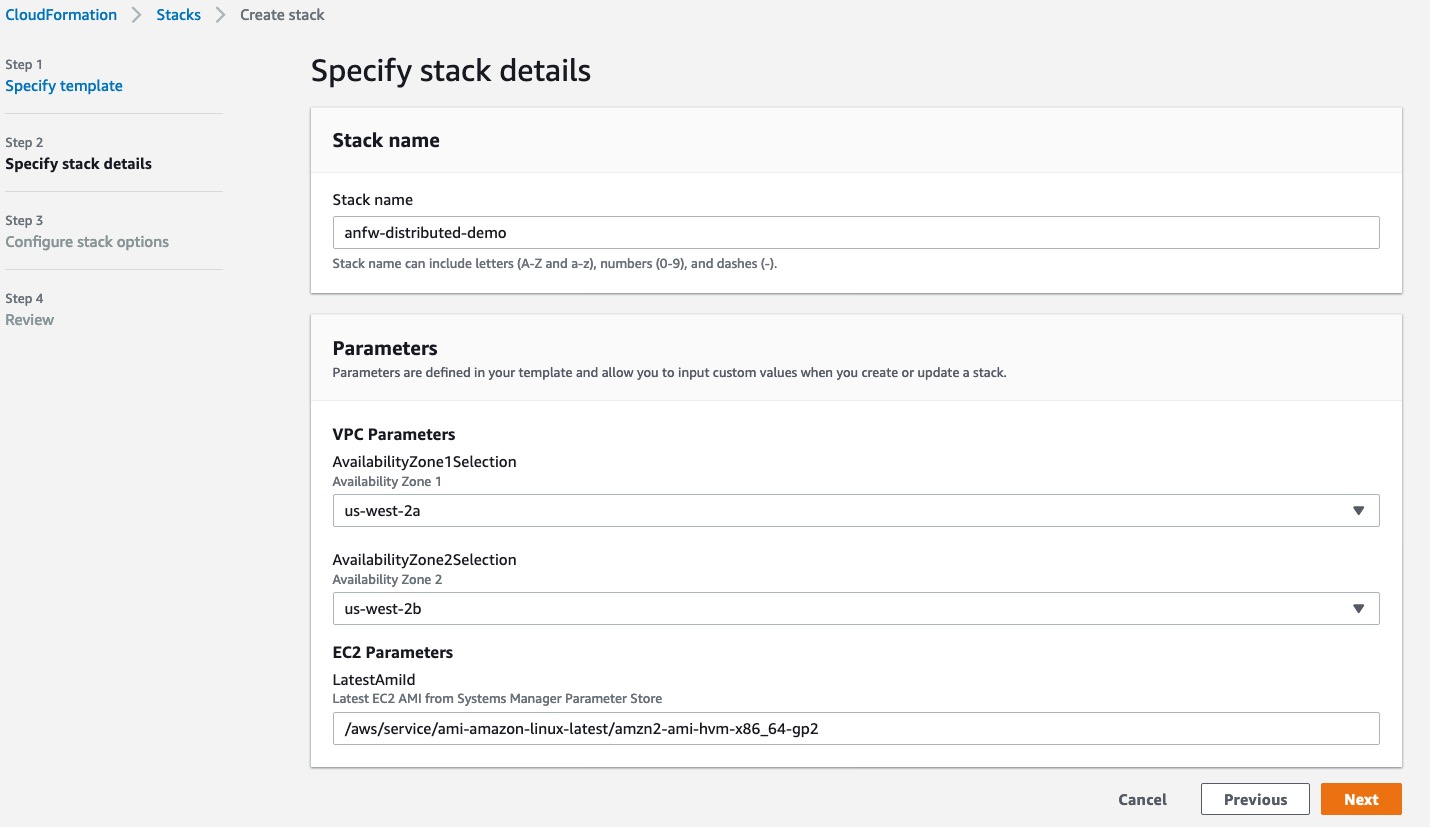
| File Name | Purpose | Template Download |
| --- | --- | --- |
| anfw-distributed-template-2az.yaml | This templaye deploys AWS Network Firewall and related resources in distributed architecture across 2 AZs. | [Click to download](https://networkfirewall.workshop.aws/anfw-distributed-template-2az.yaml) |

##### **Step 2 : Deploy using AWS CloudFormation**

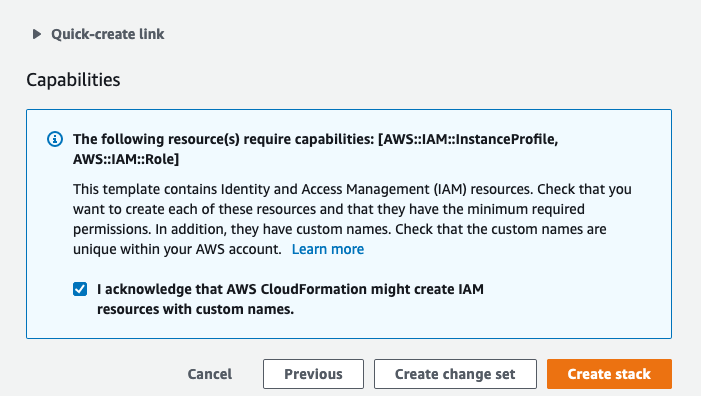
* Click [here](https://us-east-1.console.aws.amazon.com/cloudformation/home?region=us-east-1#/stacks/create/template) to open AWS CloudFormation Console in us-east-1 region.
* Select Upload template file, click on Choose file to provide the template downloaded in Step-1. Click Next once done.

[](https://networkfirewall.workshop.aws/images/setup/distributed_model/setup_distributed_model_create_stack.jpg)

* On Specify stack details , provide a meaningful Stack name as well as select a single Availability Zone to be used for this workshop. Leave rest as default and press Next.

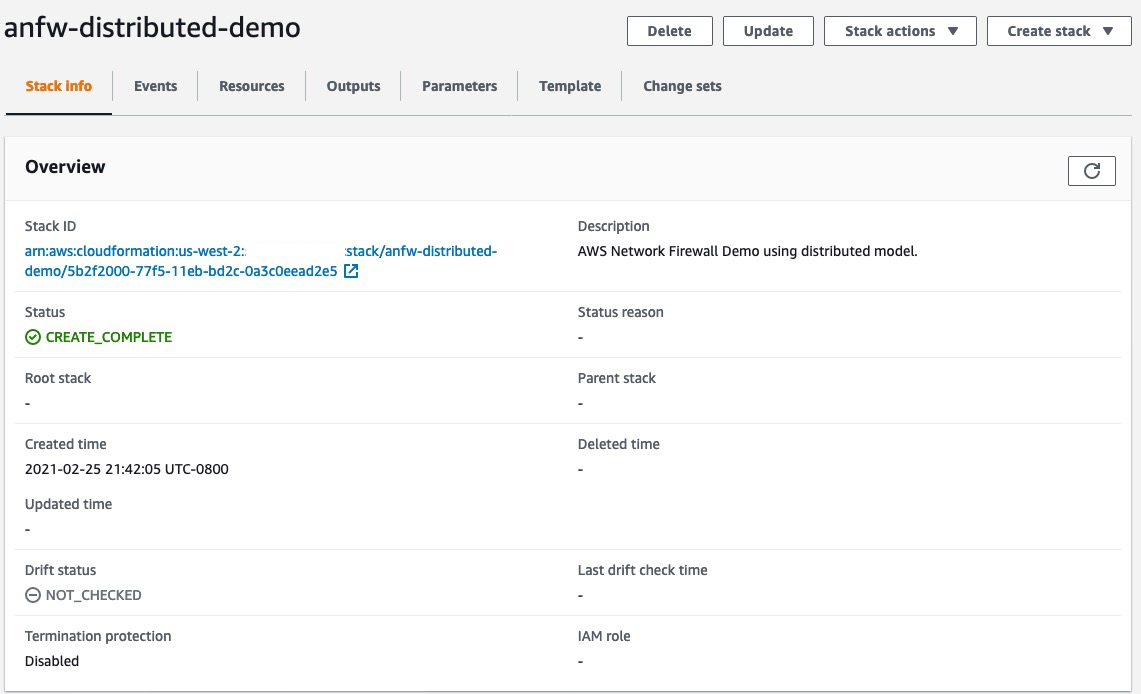
[](https://networkfirewall.workshop.aws/images/setup/distributed_model/setup_distributed_model_stack_details.jpg)

* Review and accept the defaults on Configure stack options.
* On Review page, at the bottom Acknowledge the capacities for IAM Instance Profile and IAM Roles being created as part of this stack as shown in below picture. Press “Create stack” once ready.

[](https://networkfirewall.workshop.aws/images/setup/distributed_model/distributed_stack_capabilities.png)

##### **Step 3 : Success**

Once all the resources in the template are successfully provisioned, status of CloudFormation Stack would change to CREATE\_COMPLETE as shown in picture below. Once completed, move to Lab 1 to review & verify the resources provisioned.

[](https://networkfirewall.workshop.aws/images/setup/distributed_model/distributed_stack_successful.png)

This template provision resources across 2 Availability Zones. In production, it is highly advised to use multiple AZs to ensure high availability.

LAB 1 - VERIFY FIREWALL RESOURCES

* This lab assumes you have deployed AWS Network Firewall using Distributed Deployment Model Setup. Same steps are applicable for Centralized Deployment Model.
* Resource names may vary depending on the CloudFormation stack name you provided)

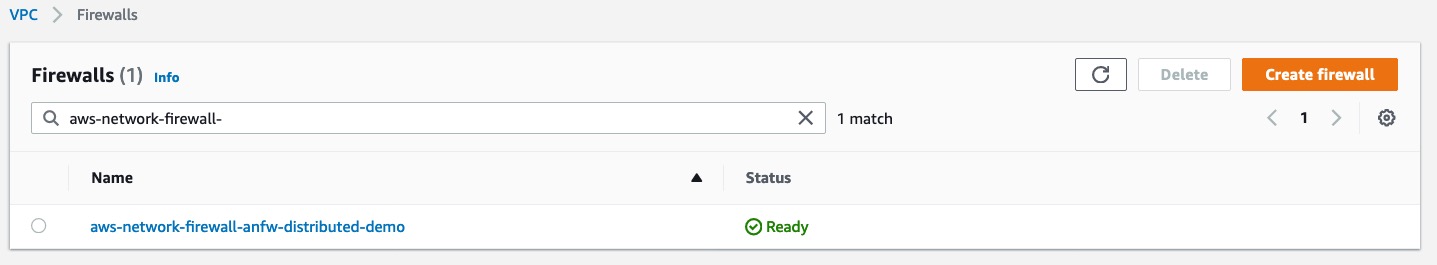
#### **Step 1 - List Firewalls**

Since we have already provisioned an AWS Network Firewall as part of our Setup instructions, let’s verify the policy and rule groups created by CloudFormation template.

AWS Network Firewall is listed under VPC in the AWS Web console:

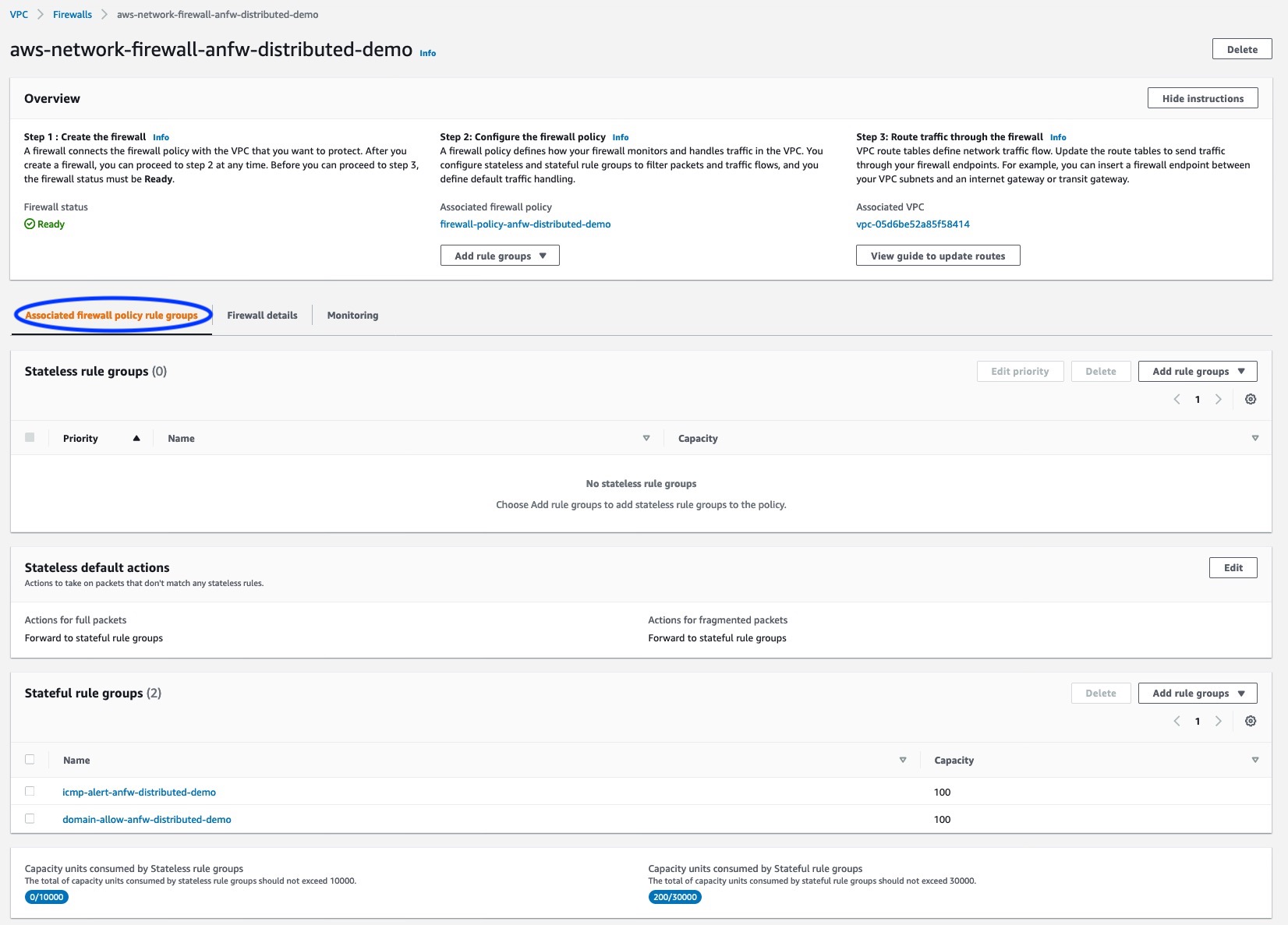
[](https://networkfirewall.workshop.aws/images/lab1/lab1_step1_figure1.png#floatleft)

In the AWS Web Console, click on VPC -> Firewalls to list the currently provisioned Firewalls.

[](https://networkfirewall.workshop.aws/images/lab1/lab1_step1_figure2.jpg)

#### **Step 2 - Firewall Details**

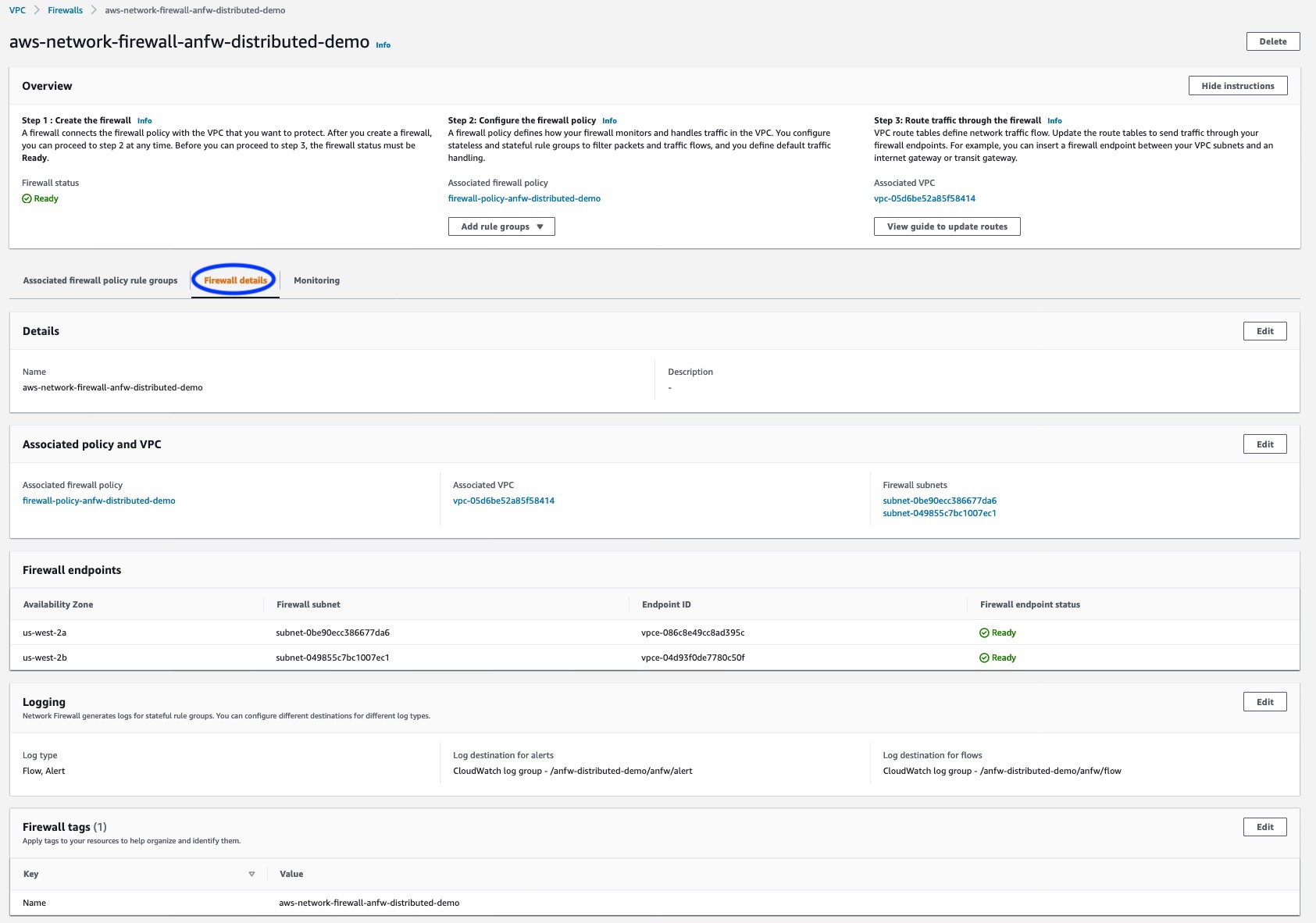
Click on aws-network-firewall-anfw-distributed-demo Firewall to see more details.

[](https://networkfirewall.workshop.aws/images/lab1/lab1_step2_figure1.jpg)

At this step, review following information :

1. What is the current status of the Firewall?
2. What policy has been associated with the Firewall?
3. What is the default Stateless action?
4. What Stateful rules are configured in the policy mentioned above?

Click on the Firewall details tab :

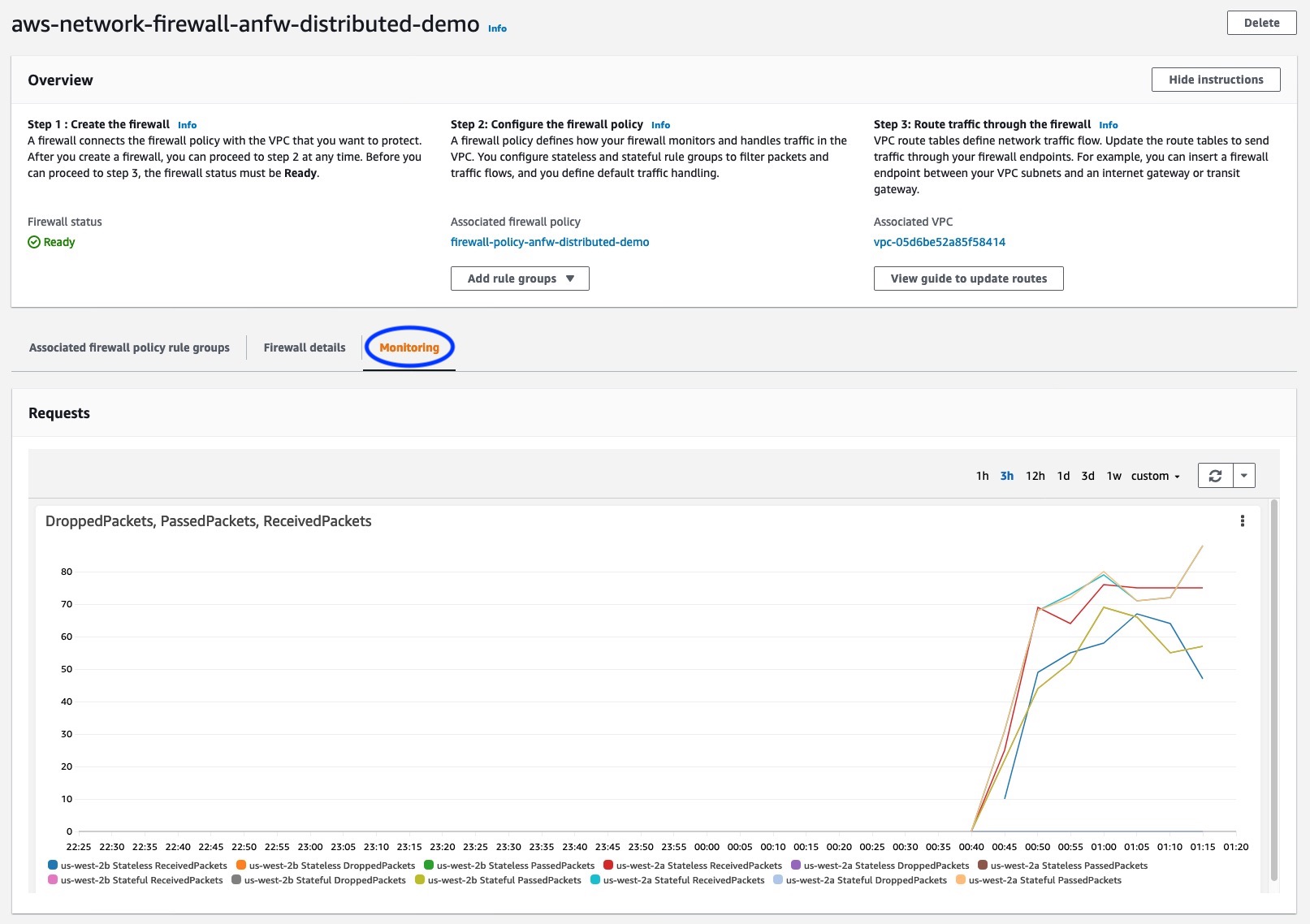
[](https://networkfirewall.workshop.aws/images/lab1/lab1_step2_figure2.jpg)

The Firewall details tab provides the following details :

* Details of Firewall endpoints & status
* Logging configuration displaying Flow/Alert type and CloudWatch Log Group configurations
* Tags

#### **Step 3 - Firewall Monitoring details**

Click on the Monitoring tab:

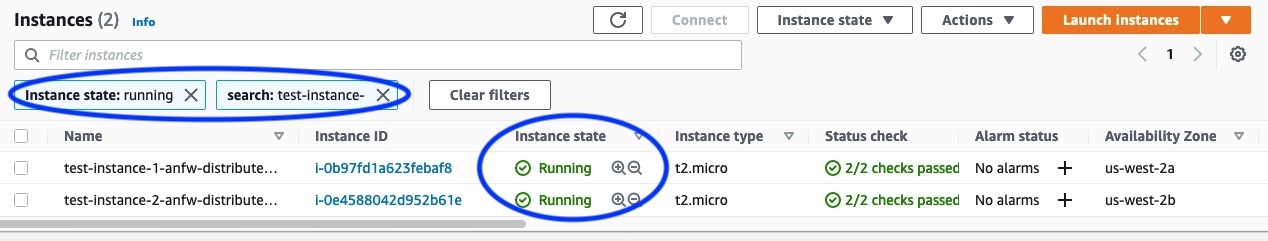
[](https://networkfirewall.workshop.aws/images/lab1/lab1_step3_figure1.jpg)

This tab provides details on Firewall metrics e.g.:

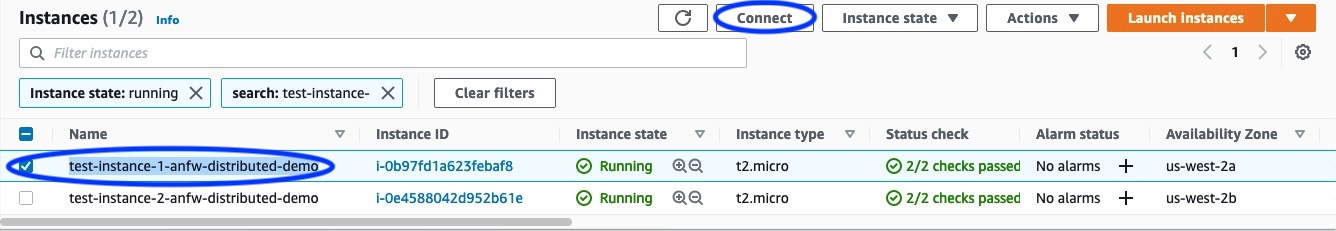
* How many packets are received/passed/dropped by Firewall Stateless Inspection
* How many packets are received/passed/dropped by Firewall Stateful Inspection

#### **Step 4 - List and verify EC2 instance**

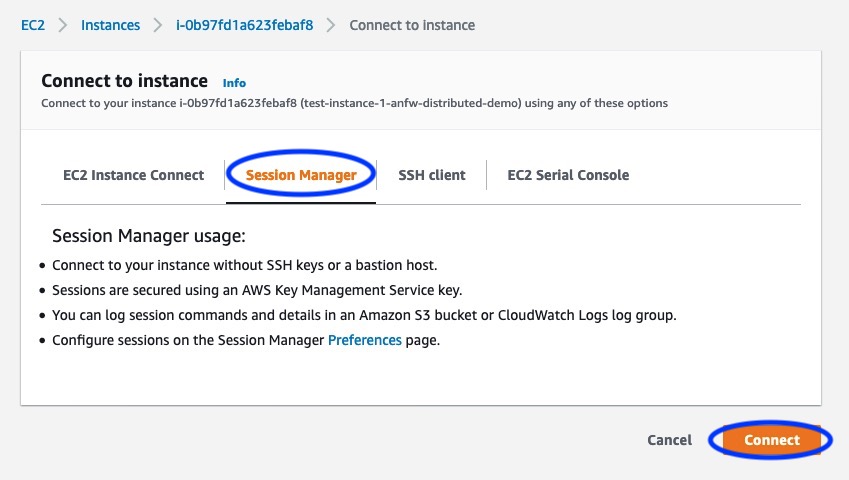
* CloudFormation stack for Distributed Deployment Model Setup creates EC2 instances test-instance-\* in Private Subnet in each AZ. In this step, we will EC2 instance details.
* EC2 instances are listed under EC2 in the AWS Web console. Launch Amazon EC2 console in the region where you have created your Distributed Deployment Model Setup and verify EC2 instance is listed and running.

[](https://networkfirewall.workshop.aws/images/lab1/lab1_step4_figure1.jpg)

* Verify you can connect to the EC2 instance:
  + Select one of the instance: test-instance-1-anfw-distributed-demo or test-instance-2-anfw-distributed-demo (instance name may vary depending on the CloudFormation stack name you provided) and cick on connect:

[](https://networkfirewall.workshop.aws/images/lab1/lab1_step4_figure2.jpg)

* + Select Session Manager and click on Connect:

[](https://networkfirewall.workshop.aws/images/lab1/lab1_step4_figure3.jpg)

* + You should see a new browser tab opened like one below:

[](https://networkfirewall.workshop.aws/images/lab1/lab1_step4_figure4.png)

* + You are now connected to EC2 instance. You can terminate the session by clicking on Terminate:

[](https://networkfirewall.workshop.aws/images/lab1/lab1_step4_figure5.png)

* + Repeat the steps for second instance as required.

# LAB 2 - EGRESS WEB FILTERING

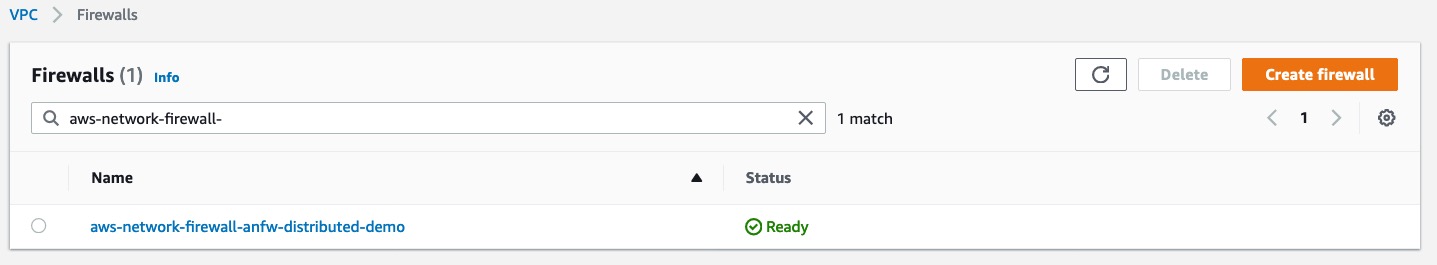
#### **Step 1 - List Firewalls & Rule Groups**

* In this step, we will verify the rule group: domain-allow-anfw-distributed-demo.

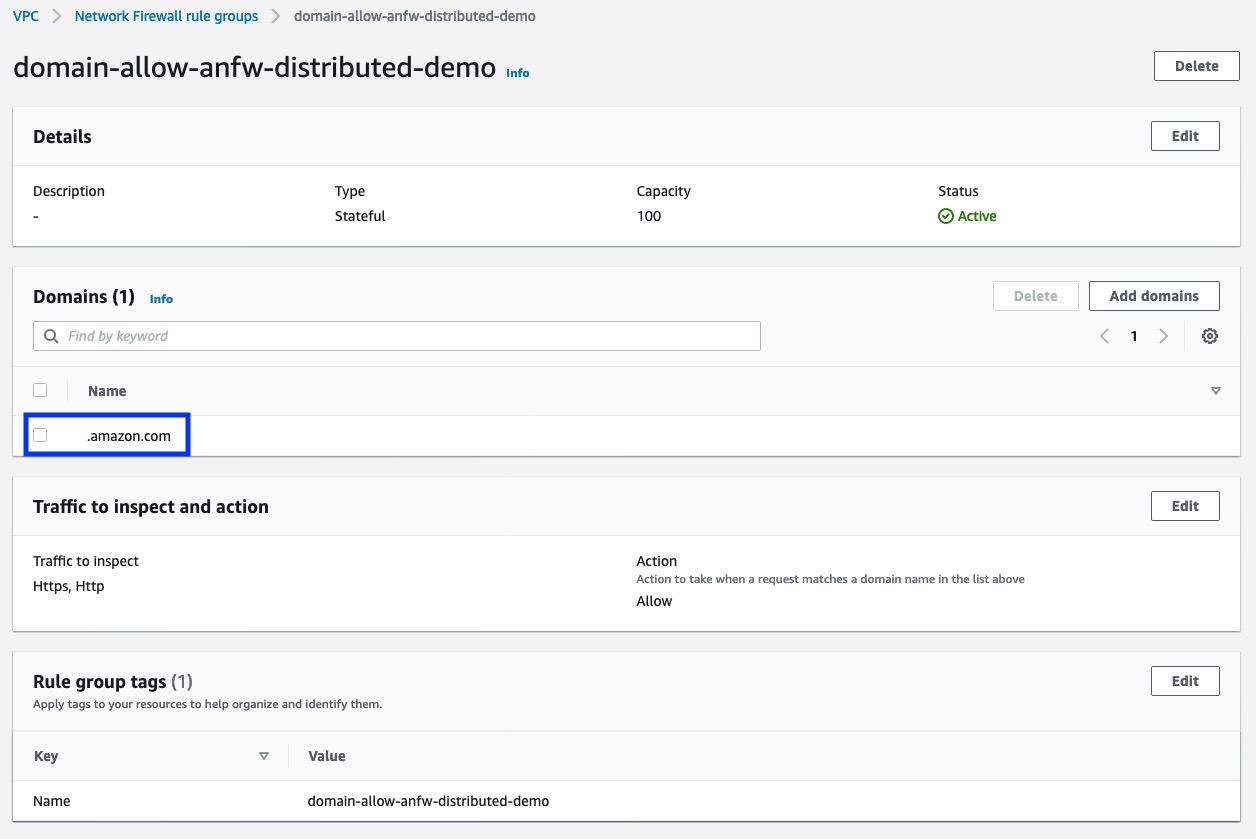
AWS Network Firewall is listed under VPC in the AWS Console:

[](https://networkfirewall.workshop.aws/images/lab2/lab2_step1_figure1.png#floatleft)

* Click on aws-network-firewall-anfw-distributed-demo firewall to see more details.

[](https://networkfirewall.workshop.aws/images/lab2/lab2_step1_figure2.jpg)

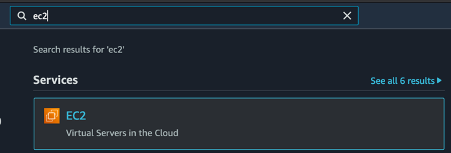
* Click on domain-allow-anfw-distributed-demo firewall rule group to see more details.

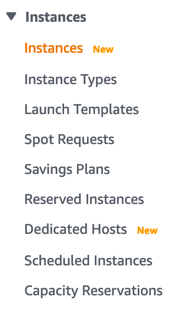
[](https://networkfirewall.workshop.aws/images/lab2/lab2_step1_figure3_with_annotation.jpg)

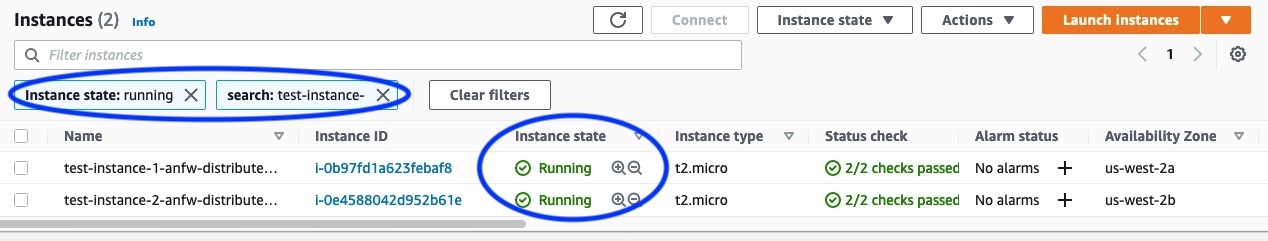
* You can observe that the Stateful rule group domain-allow-anfw-demo will allow traffic matching domain .amazon.com. You can add further domains here for testing purposes or leave it as default.

#### **Step 2 - List & Verify EC2 Instances**

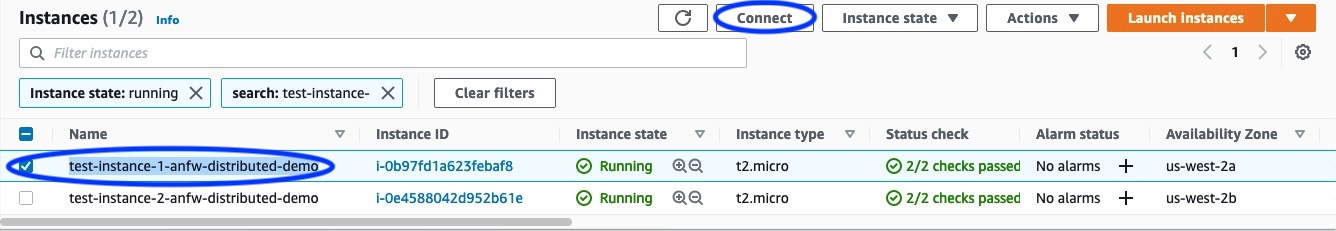
* CloudFormation stack for Distributed Deployment Model Setup creates EC2 instances test-instance-\* in Private Subnet in each AZ. In this step, we will verify EC2 instance details.
* EC2 instances are listed under EC2 in the AWS Web console. Launch Amazon EC2 console in the region where you have created your Distributed Deployment Model Setup and verify EC2 instance is listed and running.

[](https://networkfirewall.workshop.aws/images/lab2/lab2_step2_figure1.png)

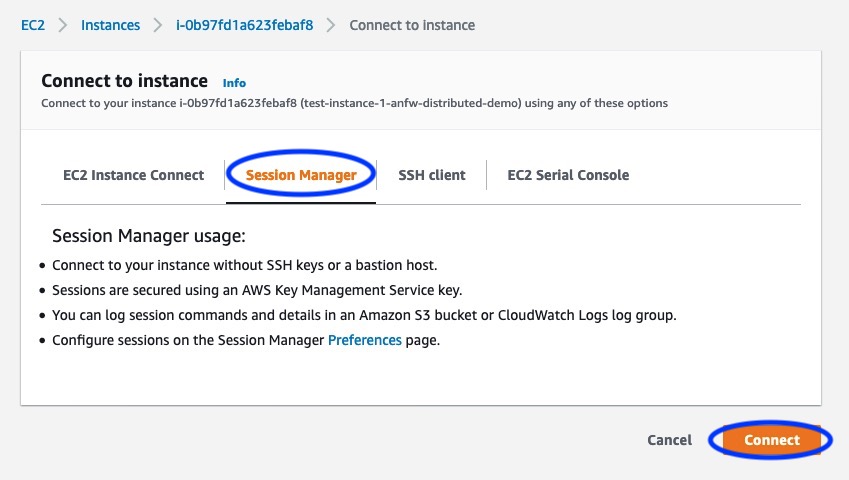
[](https://networkfirewall.workshop.aws/images/lab2/lab2_step2_figure2.png)

[](https://networkfirewall.workshop.aws/images/lab2/lab2_step2_figure3.jpg)

* Connect to EC2 instance:
  + Select one of the instance: test-instance-1-anfw-distributed-demo or test-instance-2-anfw-distributed-demo (instance name may vary depending on the CloudFormation stack name you provided) and cick on connect:

[](https://networkfirewall.workshop.aws/images/lab2/lab2_step2_figure4.jpg)

* Select Session Manager and click on Connect:

[](https://networkfirewall.workshop.aws/images/lab2/lab2_step2_figure5.jpg)

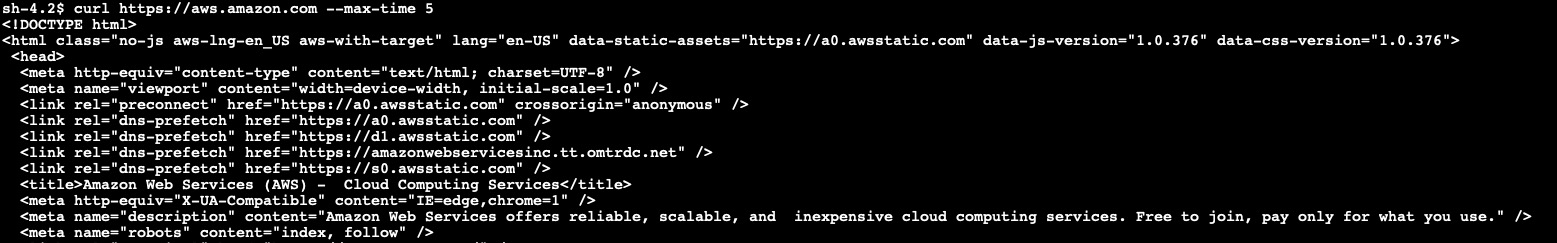
* Now we are ready to test the firewall policy.

#### **Step 3 - Verify Domain Filtering**

* On the EC2 instance, run the command below:

curl https://aws.amazon.com --max-time 5

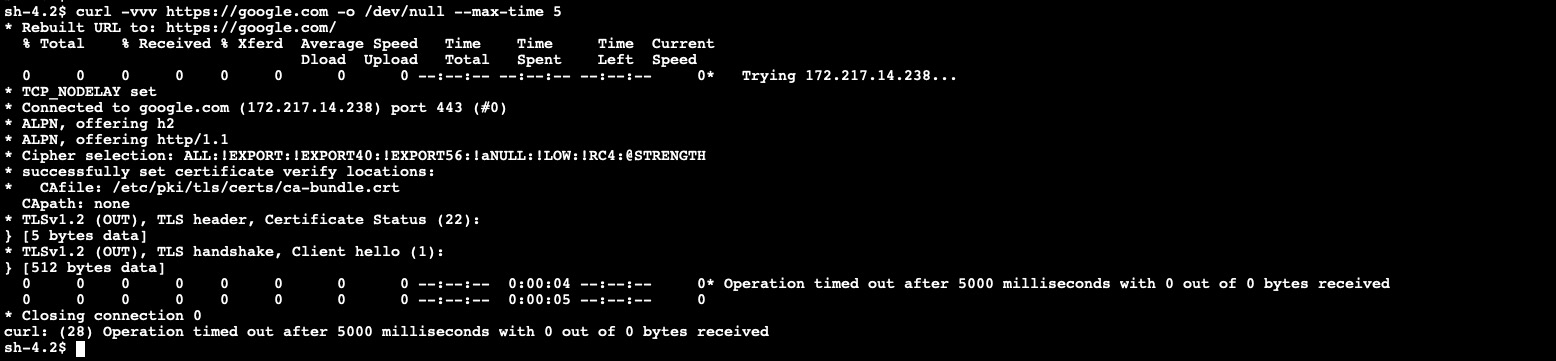
* You will observe the output for the curl command. Notice that the curl command is completed successfuly.

[](https://networkfirewall.workshop.aws/images/lab2/lab2_step3_figure1.jpg)

* Let’s test again with another domain which isn’t permitted by default. On the EC2 instance, Run the below command:

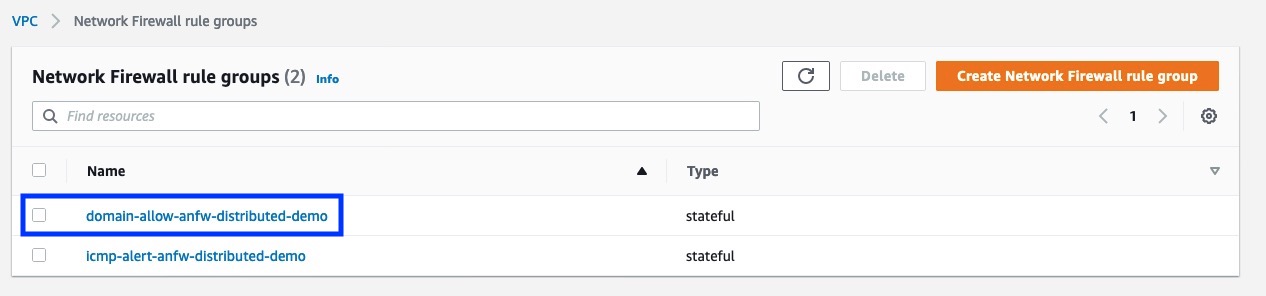
curl -vvv https://google.com -o /dev/null --max-time 5

* Notice that the curl command times out eventually without any data returning.

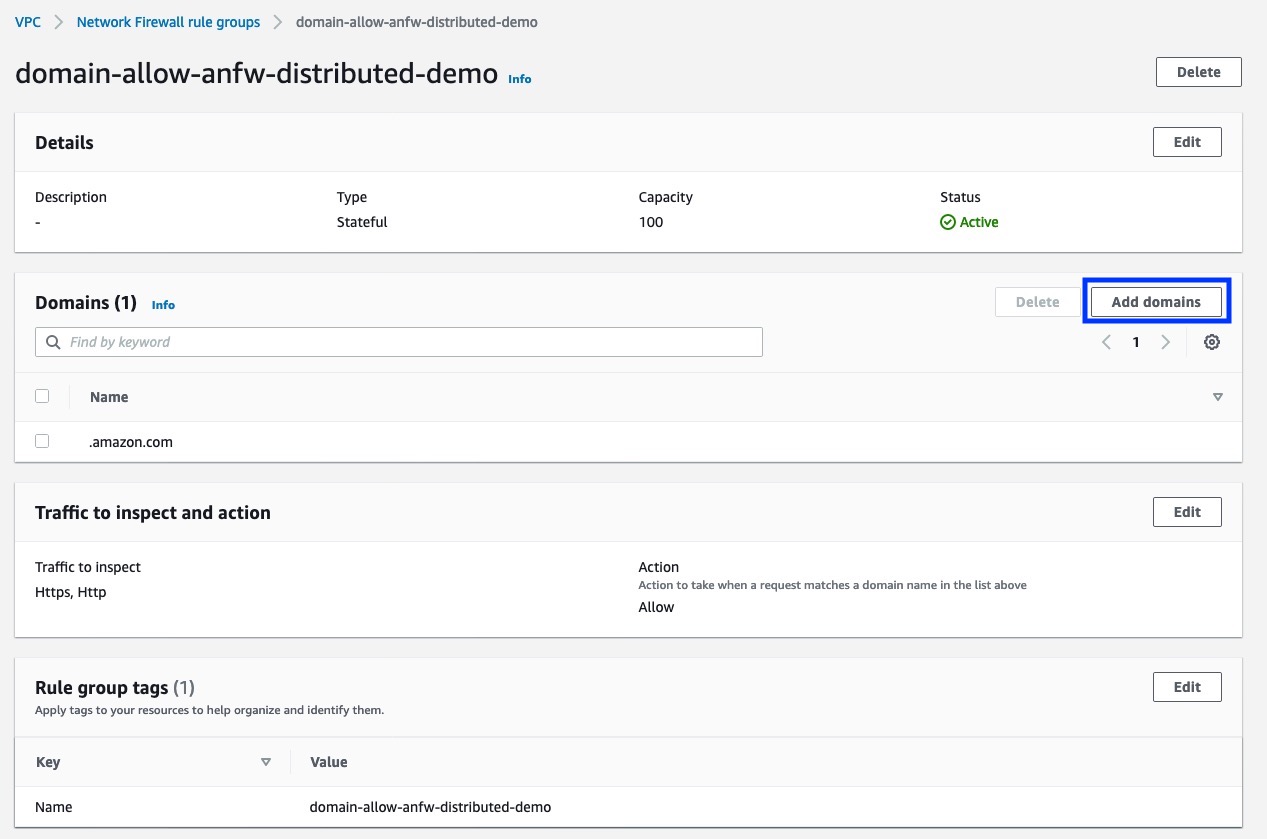
[](https://networkfirewall.workshop.aws/images/lab2/lab2_step3_figure2.jpg)

#### **Step 4 - Update Domain Rule Group**

* On AWS Network Firewall console, under Network Firewall rule groups, Select domain-allow-anfw-distributed-demo rule group.

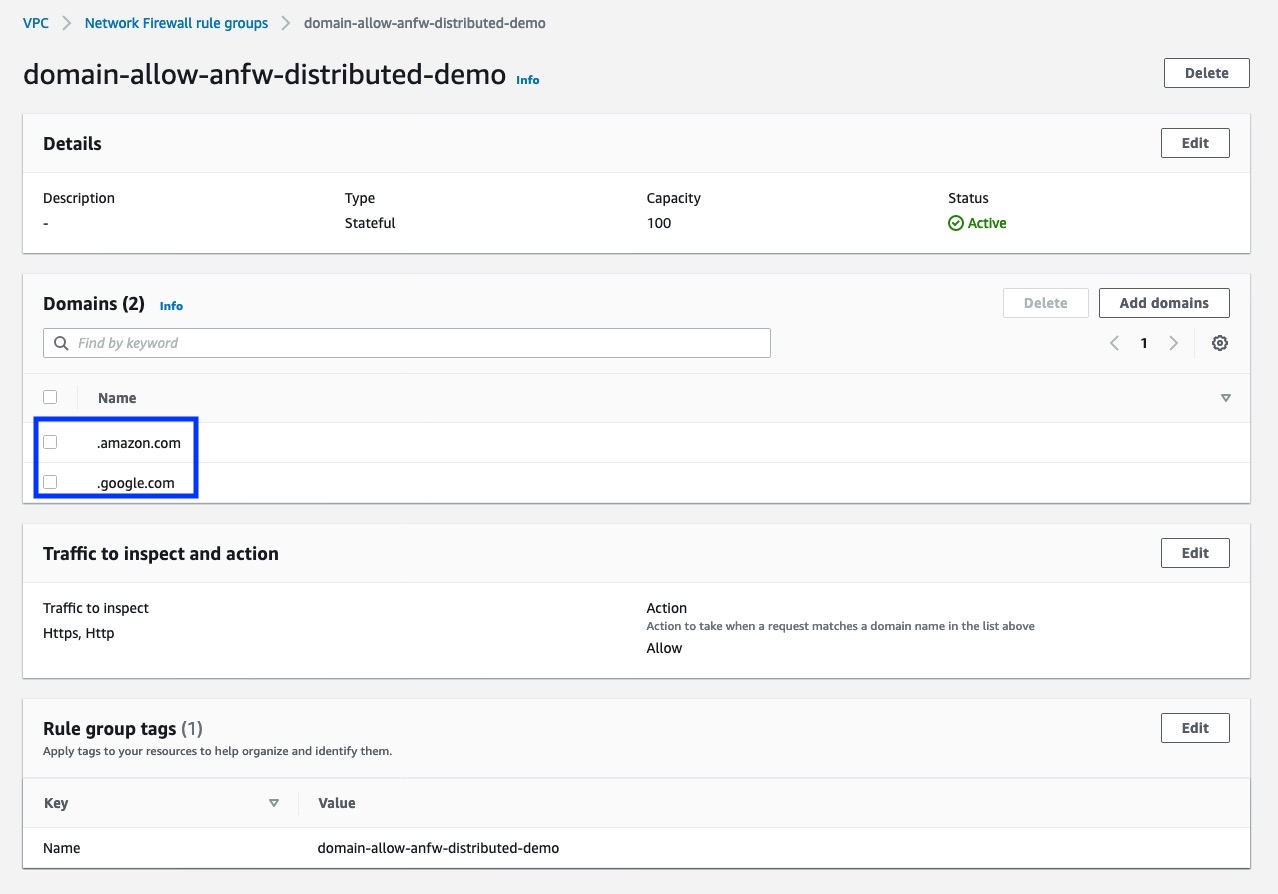
[](https://networkfirewall.workshop.aws/images/lab2/lab2_step4_figure1_with_annotation.jpg)

* Click on Add domains and add .google.com.

[](https://networkfirewall.workshop.aws/images/lab2/lab2_step4_figure2_with_annotation.jpg)

[](https://networkfirewall.workshop.aws/images/lab2/lab2_step4_figure3.jpg)

* Now we have 2 domains that are allowed in this Rule Group.

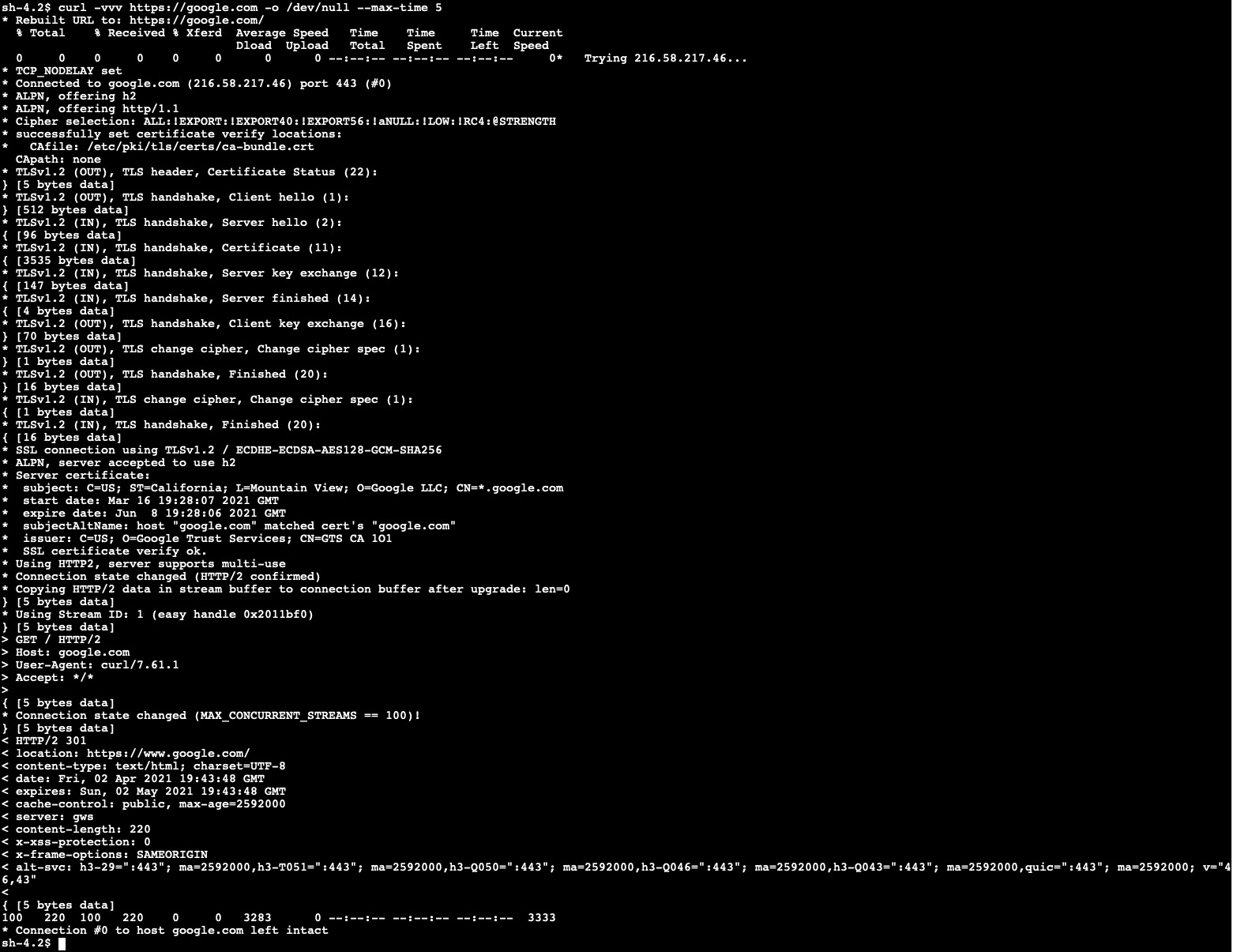
[](https://networkfirewall.workshop.aws/images/lab2/lab2_step4_figure4_with_annotation.jpg)

#### **Step 5 – Verify updated domain success**

* Run the below command again on the EC2 instance:

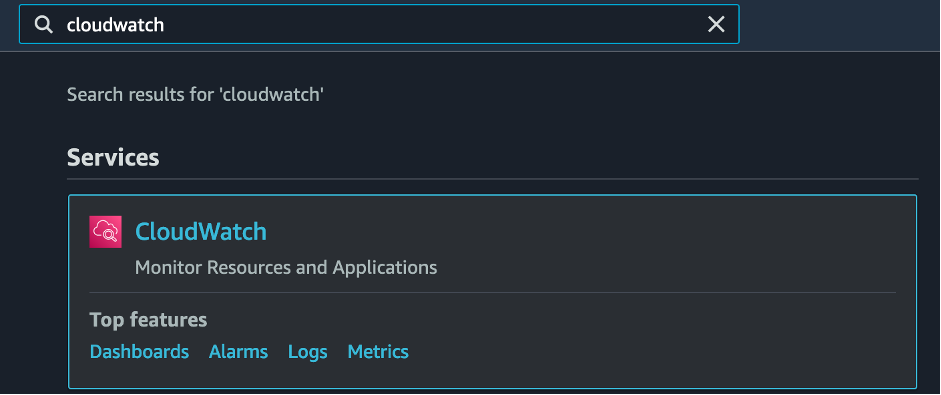
curl -vvv https://google.com -o /dev/null --max-time 5

* Notice that the curl command completes successfuly now that we have permitted the domain in Rule Group.

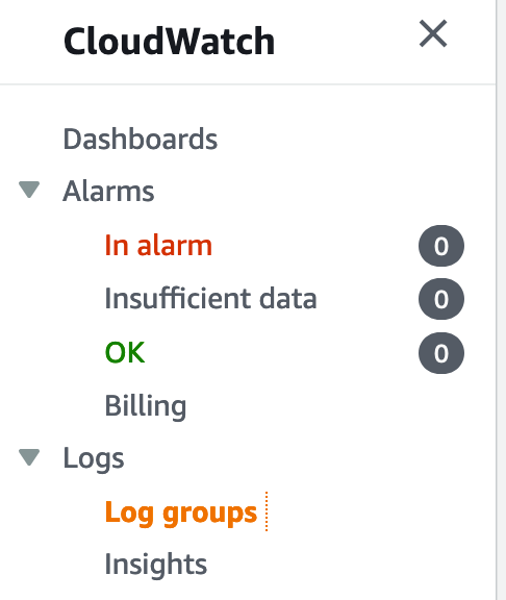
[](https://networkfirewall.workshop.aws/images/lab2/lab2_step5_figure1.jpg)

#### **Step 6 - Verify Alert Logs captured in CloudWatch**

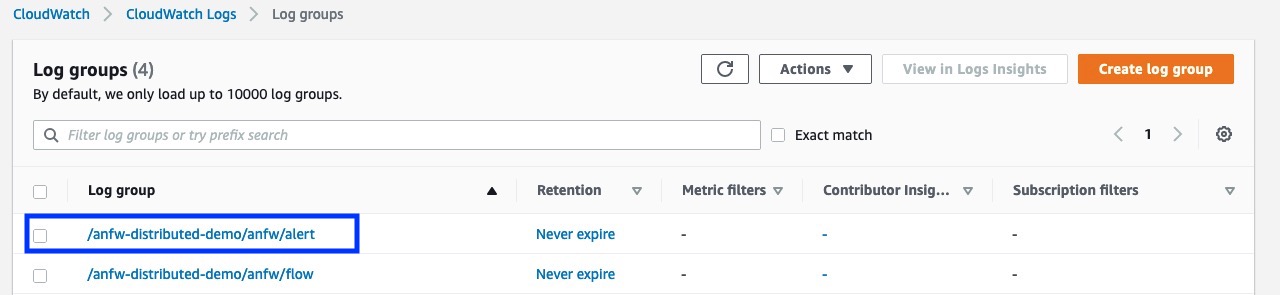
* AWS Log groups are listed under CloudWatch in the AWS Web console:

[](https://networkfirewall.workshop.aws/images/lab2/lab2_step6_figure1.png)

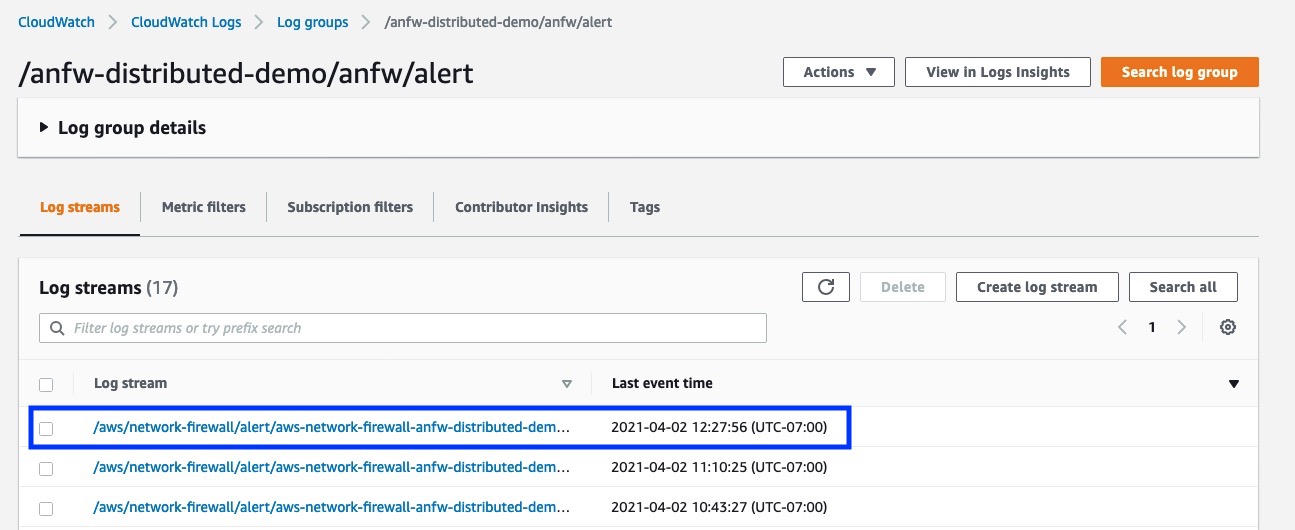
* Click on Log groups under CloudWatch

[](https://networkfirewall.workshop.aws/images/lab2/lab2_step6_figure2.png)

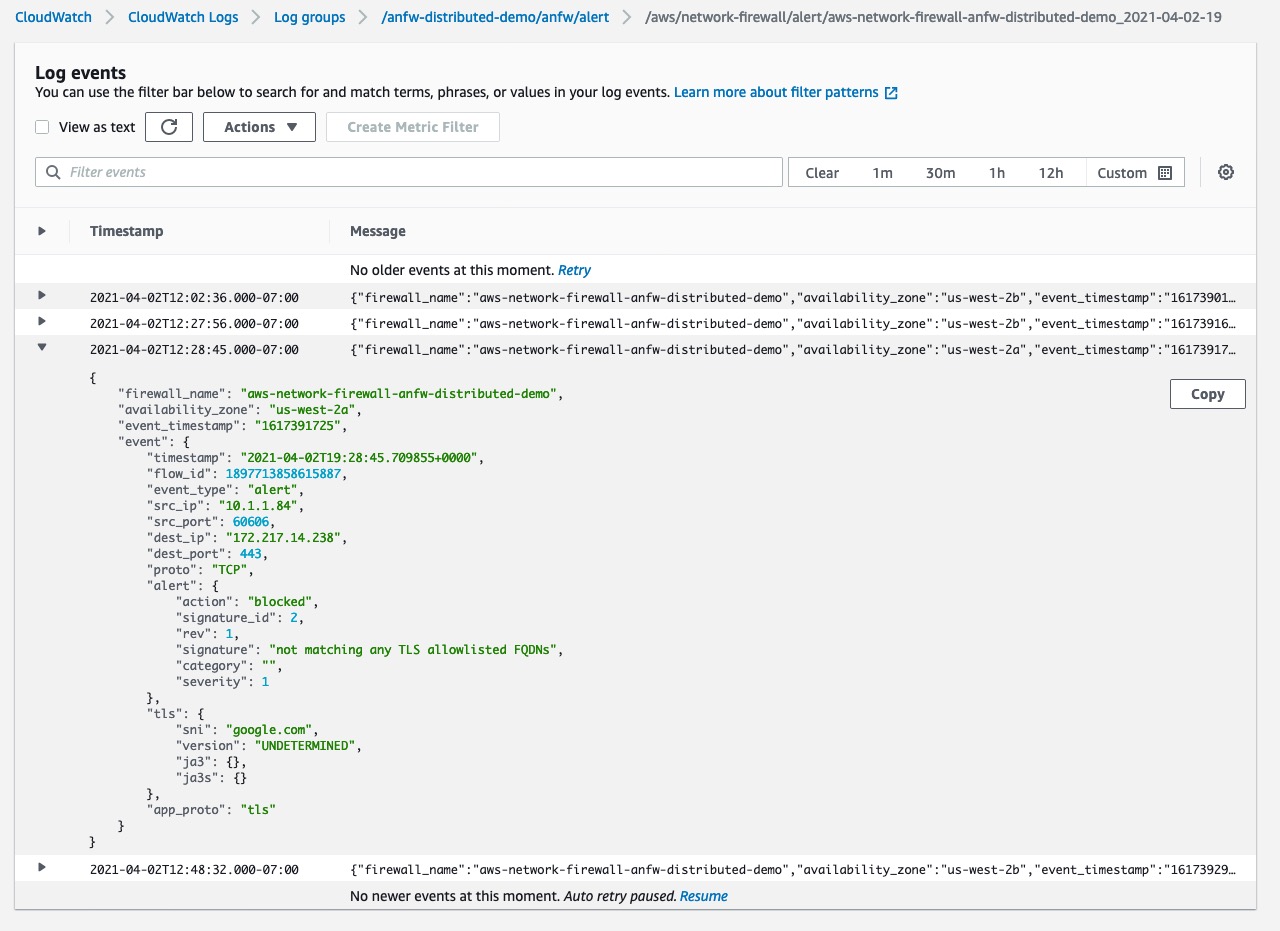
* Click on Log group /anfw-distributed-demo/anfw/alert

[](https://networkfirewall.workshop.aws/images/lab2/lab2_step6_figure3_with_annotation.jpg)

* Select the latest Log Streams /aws/network-firewall/alert/aws-network-firewall-anfw-distributed-demo\_\*

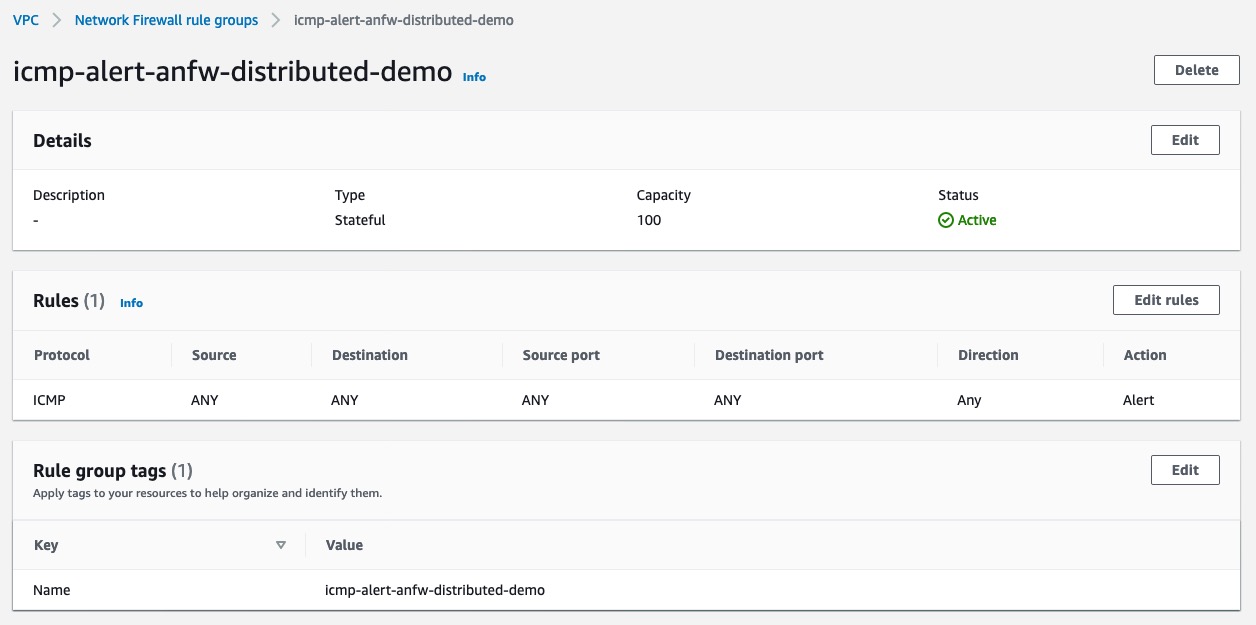
[](https://networkfirewall.workshop.aws/images/lab2/lab2_step6_figure4_with_annotation.jpg)

* You will observe alerts being captured with all details under the log group. A sample screenshot of alert is below:

[](https://networkfirewall.workshop.aws/images/lab2/lab2_step6_figure5.jpg)

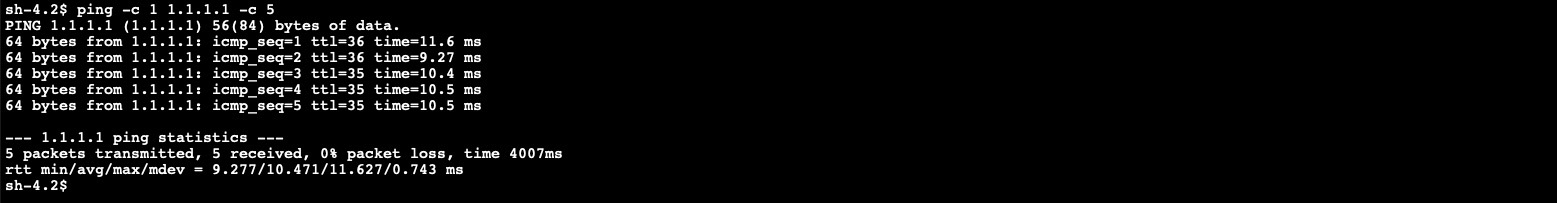
#### **Optional - ICMP Alerts**

* As you may have noticed, we have also provisioned another Rule Group: icmp-alert-anfw-\*. This allows all ICMP traffic to pass through but logs the traffic in logs.

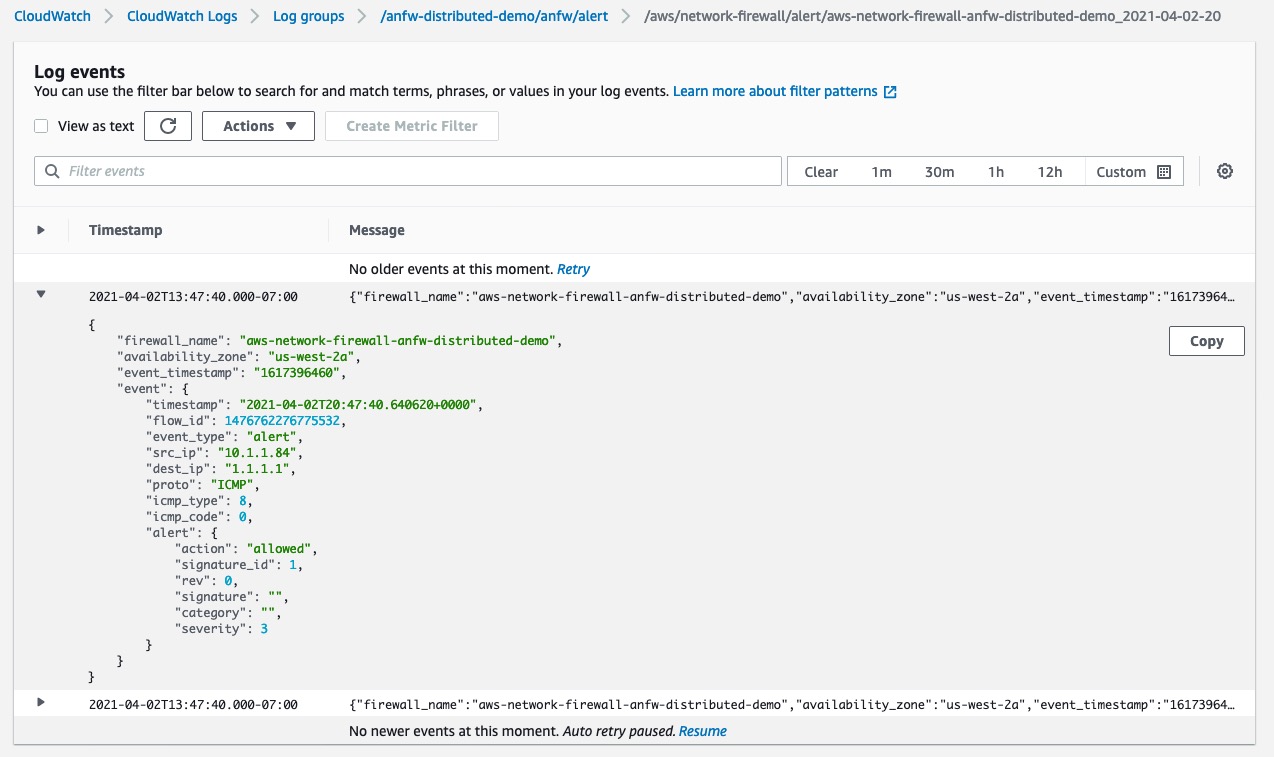
[](https://networkfirewall.workshop.aws/images/lab2/lab2_optional_figure1.jpg)

* To test this Rule Group, on your EC2 instance, execute following command:

ping -c 1 1.1.1.1 -c 5

[](https://networkfirewall.workshop.aws/images/lab2/lab2_optional_figure2.jpg)

* Follow the instructions provided under Step 6 to view the logs. A sample log screen shot is as follows:

[](https://networkfirewall.workshop.aws/images/lab2/lab2_optional_figure3.jpg)

# LAB 3 - USING OPEN SOURCE RULES WITH AWS NETWORK FIREWALL

Let’s take a look how we can utilize Open Source/Suricata compatible rules in AWS Network Firewall. For this example, we’ll choose Suricata specific rules from the community such as Proofpoint’s OPEN ruleset found [here](https://rules.emergingthreats.net/open/). More details on open-source and commercial rules from Proofpoint are available [here](https://www.proofpoint.com/au/products/et-pro-ruleset). AWS Network Firewall can be setup in various deployment models depending on the requirements. To get more details on the deployment models and how to setup firewalls, go through AWS Network Firewall [documentation](https://docs.aws.amazon.com/network-firewall/latest/developerguide/what-is-aws-network-firewall.html) and [Deployment Model for AWS Network Firewall (blog)](https://aws.amazon.com/blogs/networking-and-content-delivery/deployment-models-for-aws-network-firewall/).

To use open-source rules with AWS Network Firewall, you must follow below steps :

#### **1. Download/Clone Rules**

For this lab, we’ll choose “User-Agents” category from Proofpoint’s OPEN rules which can be used to detect suspicious user-agents. Rules for this category are available at <https://rules.emergingthreats.net/open/suricata-5.0/rules/emerging-user_agents.rules>

Download the rules using ‘wget’ to your local machine using following commands :

wget https://rules.emergingthreats.net/open/suricata-5.0/rules/emerging-user\_agents.rules -O emerging-user-agents.rules

Once downloaded, note down the location where the rules are saved so we can use it in next steps.

#### **2. Create Rule Group with Suricata-compatible rules**

To create a Rule Group with above rule set - Navigate to the AWS Console → VPC → Network Firewall rule group and click on “Create Network Firewall rule group”. Select “stateful rule group” from the page as displayed in Figure 2.

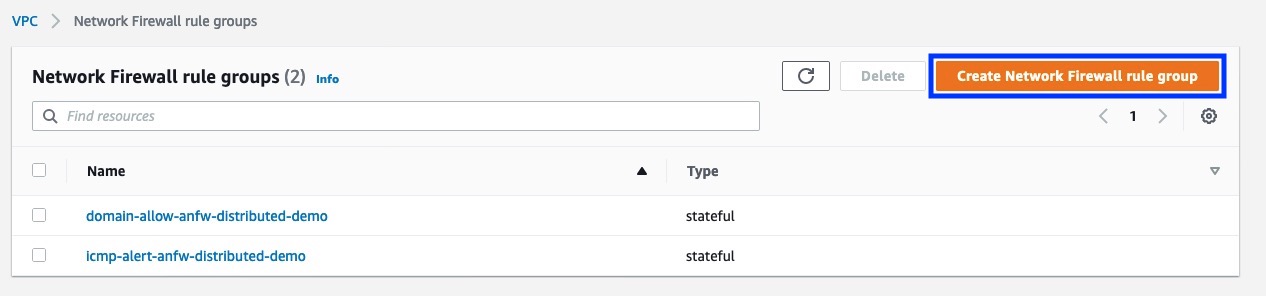
[](https://networkfirewall.workshop.aws/images/lab3/lab3_step2_figure1_with_annotation.jpg)

Figure 1 : Create Network Firewall rule group

Select “Stateful rule group” from the page as displayed below.

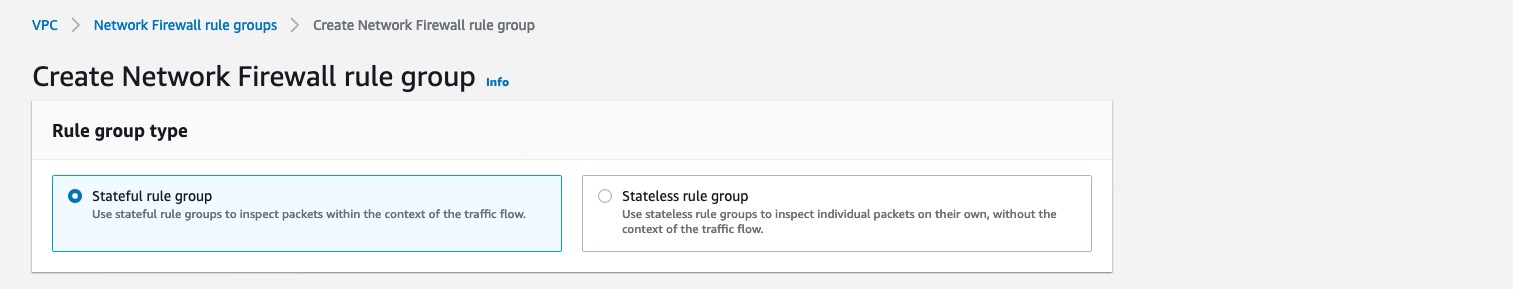
[](https://networkfirewall.workshop.aws/images/lab3/lab3_step2_figure2.jpg)

Figure 2 : Create Network Firewall rule group - Stateful rule group

Once selected, define a meaningful name as well as capacity for the rule group ([see more on Capacity here](https://docs.aws.amazon.com/network-firewall/latest/developerguide/rule-group-capacity.html)) and select the “Suricata compatible IPS rules” option. Once selected, a text input field would be available to input the ruleset.

Copy and paste the rules from Step 1 which contains the Emerging Threats User-Agents rules and press “Create stateful rule group” as displayed in Figure 3.

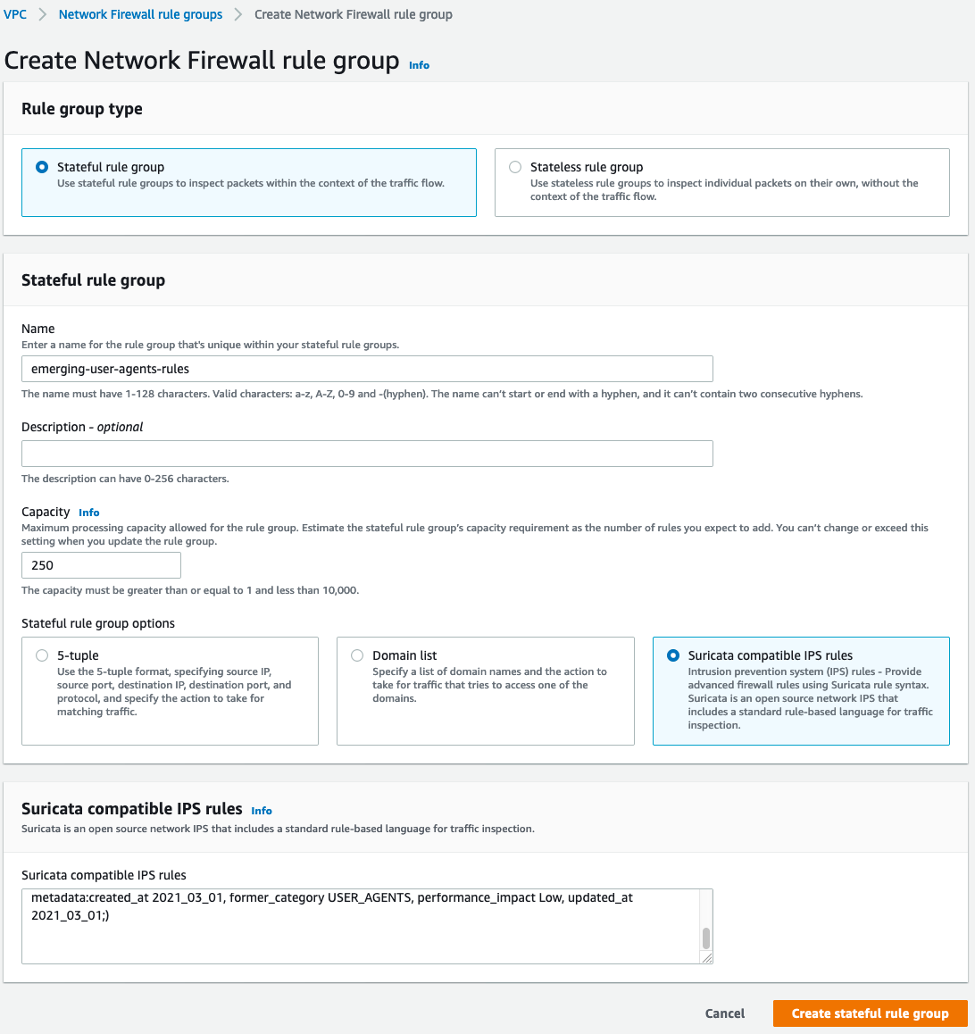
[](https://networkfirewall.workshop.aws/images/lab3/lab3_step2_figure3.png)

Figure 3 : Create Network Firewall rule group - Suricata compatible IPS rules

Optional: You can also use following AWS CLI commands to create the Rule Group:

aws network-firewall create-rule-group --rule-group-name emerging-user-agents-rules --type STATEFUL --capacity 250 —-rules file:*//emerging-user-agents.rules*

The HOME\_NET rule group variable is used to define source IP range eligible for processing in the Stateful Domain list and optionally Suricata compatible IPS Rule Groups. By default, it is set to the VPC CIDR where firewall endpoints are deployed. With centralized deployment model, this variable must be expanded on each rule group to include all CIDR ranges of your VPCs and on-premises networks to make them eligible for processing. See [documentation](https://docs.aws.amazon.com/network-firewall/latest/developerguide/stateful-rule-groups-domain-names.html#stateful-rule-groups-domain-names-home-net) for more details.

#### **3. Modify Firewall Policy to add and forward traffic to Stateful Rule Groups**

To add the newly created Stateful Rule in previous step to Firewall Policy - Navigate to AWS Console → VPC → Firewall Policies and click on the policy used by your firewall as shown in Figure 4.

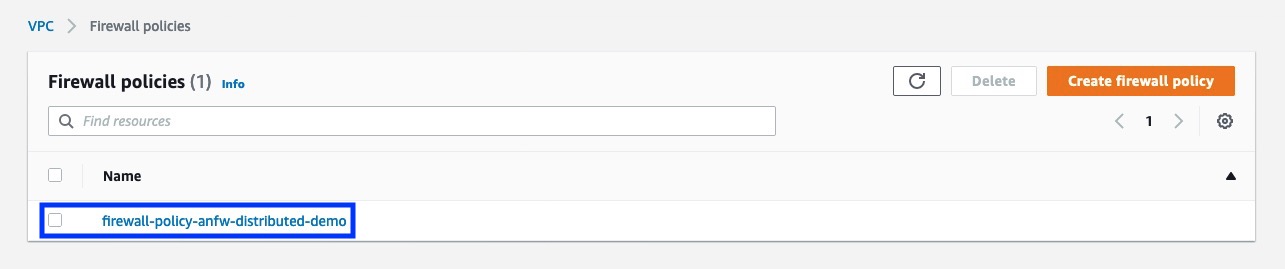
[](https://networkfirewall.workshop.aws/images/lab3/lab3_step3_figure1_with_annotation.jpg)

Figure 4 : Click on the in-use Firewall policy

Before we add the new rule group, lets remove rule groups added in previous labs. To remove the rules, click on the rules and select Delete. Now under “Stateful rule groups” section, click “Add rule groups” and select “Add stateful rule group to firewall policy” as shown in Figure 5.

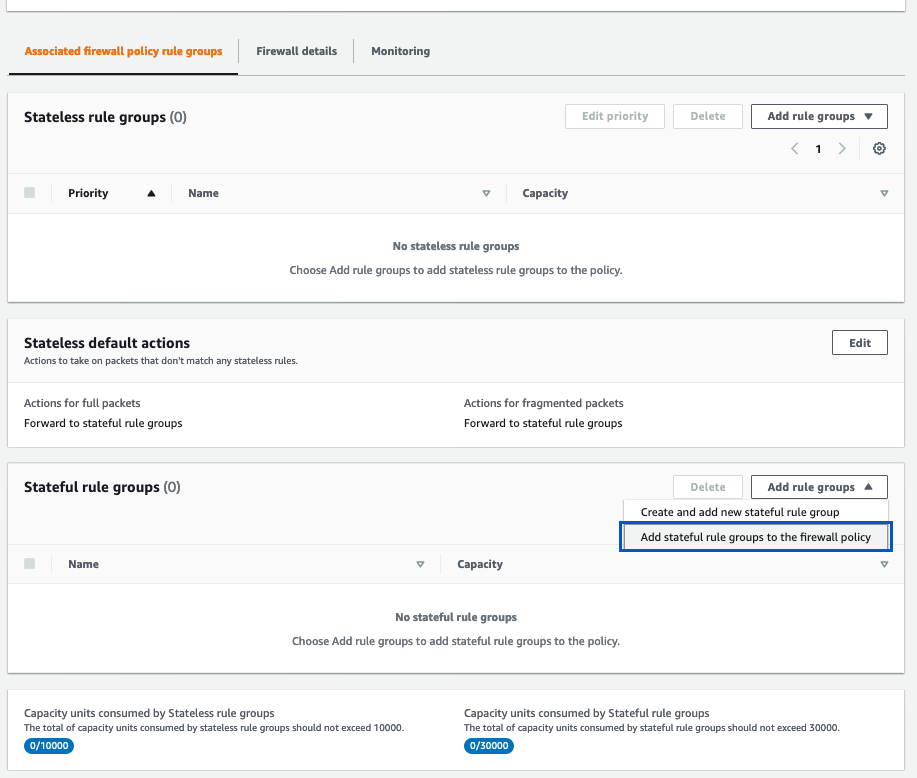
[](https://networkfirewall.workshop.aws/images/lab3/lab3_step3_figure2_with_annotation.jpg)

Figure 5 : Add Statefule Rule Group

In next step - select the stateful rule group you’ve created in Step 2 and click “Add stateful rule group” to update the policy.

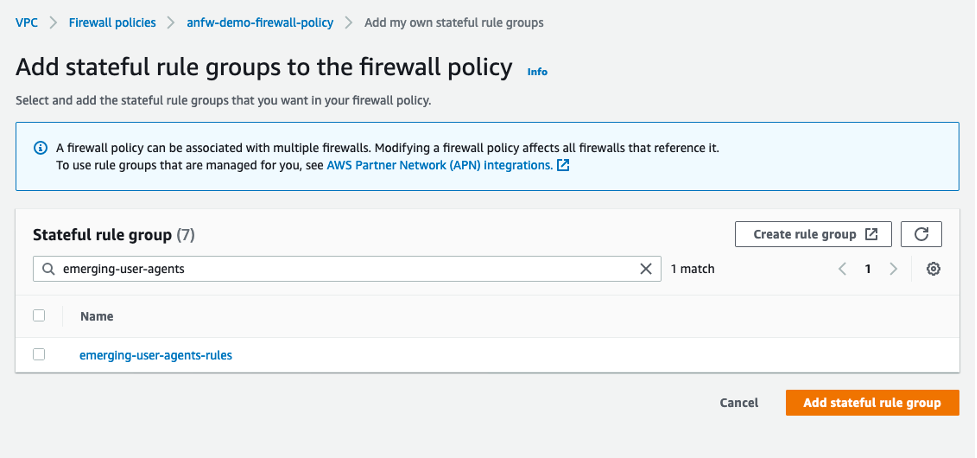
[](https://networkfirewall.workshop.aws/images/lab3/lab3_step3_figure3_with_annotation.png)

Figure 6 : Update the policy

To ensure traffic is forwarded to stateful inspection engine, you also must add a custom defined stateless rule group which cover the interesting traffic or you can set a default action for all stateless traffic to be forwarded to stateful rule groups in the firewall policy. In the example below, we have used the stateless default actions to forward traffic towards the stateful rule groups.

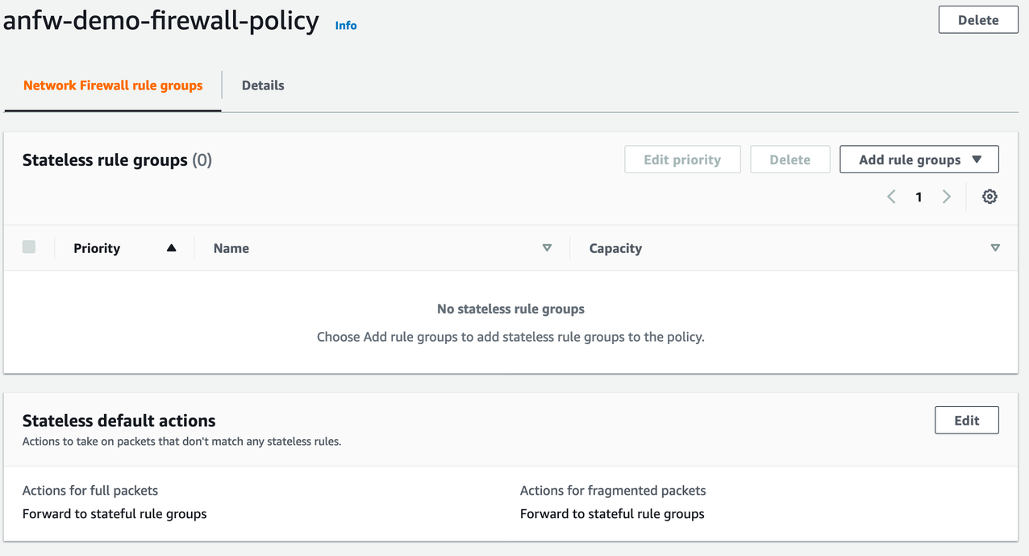
[](https://networkfirewall.workshop.aws/images/lab3/lab3_step3_figure5_with_annotation.png)

Figure 7 : AWS Network Firewall Policy with Stateful default actions.

Optional: You can also complete these steps using the AWS CLI:

Create a file policy.json with ARN of Stateful Rule created in Step 2:

{

"StatelessDefaultActions": [

"aws:forward\_to\_sfe"

],

"StatelessFragmentDefaultActions": [

"aws:forward\_to\_sfe"

],

"StatefulRuleGroupReferences": [

{

"ResourceArn": "arn:aws:network-firewall:us-west-2:XXXXXXXXXX:stateful-rulegroup/emerging-user-agents-rules"

}

]

}

Fetch the latest Update Token and update the policy using policy document you created above (policy.json):

UPDATETOKEN=(`aws network-firewall describe-firewall-policy --firewall-policy-name anfw-demo-firewall-policy --output text --query UpdateToken`)

aws network-firewall update-firewall-policy --firewall-policy-name anfw-demo-firewall-policy --firewall-policy file://policy.json --update-token $UPDATETOKEN

Verify the policy :

aws network-firewall describe-firewall-policy --firewall-policy-arn arn:aws:network-firewall:us-west-2:XXXXXXXXXX:firewall-policy/anfw-demo-firewall-policy --region us-west-2

#### **4. Test & monitor**

To generate interesting traffic, you must have compute resources (e.g. an EC2 instance) which are protected by AWS Network Firewall.

In this lab, we’ll use an EC2 instance (mentioned as secure host) to generate request against signature # 2029569 (also displayed below) from the Proofpoint’s OPEN rule set imported earlier in Step 3 which detects the suspicious user-agents .

alert http $HOME\_NET any -> $EXTERNAL\_NET any (msg:"ET USER\_AGENTS Observed Suspicious UA (easyhttp client)"; flow:established,to\_server; http.user\_agent; content:"easyhttp client"; bsize:15; classtype:bad-unknown; sid:2029569; rev:1; metadata:attack\_target Client\_Endpoint, created\_at 2020\_03\_04, deployment Perimeter, former\_category USER\_AGENTS, signature\_severity Informational, updated\_at 2020\_03\_04;)

To test , use following command on secure host :

wget -U "easyhttp client" http:*//www.amazon.com -o /dev/null*

This command will generate a HTTP GET request with a user agent as “easyhttp client”. To check the logs, navigate to AWS Console → CloudWatch → Log groups & select the configured log group for your firewall. You should have received an alert from AWS Network Firewall on above traffic as displayed in Figure 8 below.

[](https://networkfirewall.workshop.aws/images/lab3/lab3_step3_figure1.png)

Figure 8 : CloudWatch logs for Alert from AWS Network Firewall

And that concludes how to import Suricata rule references into AWS Network Firewall.