VM-Series for AWS

AWS Cloud Formation Template

Deployment Guide

How to deploy a two-tiered application environment secured by the VM-Series firewall

http://www.paloaltonetworks.com

© 2016 Palo Alto Networks. Proprietary and Confidential

Last Update: May 2016

Table of Contents

Version History

Version number	Comments
1.0	Initial GitHub check-in
1.1	Update links in doc to point to GitHub

1. About CFTs

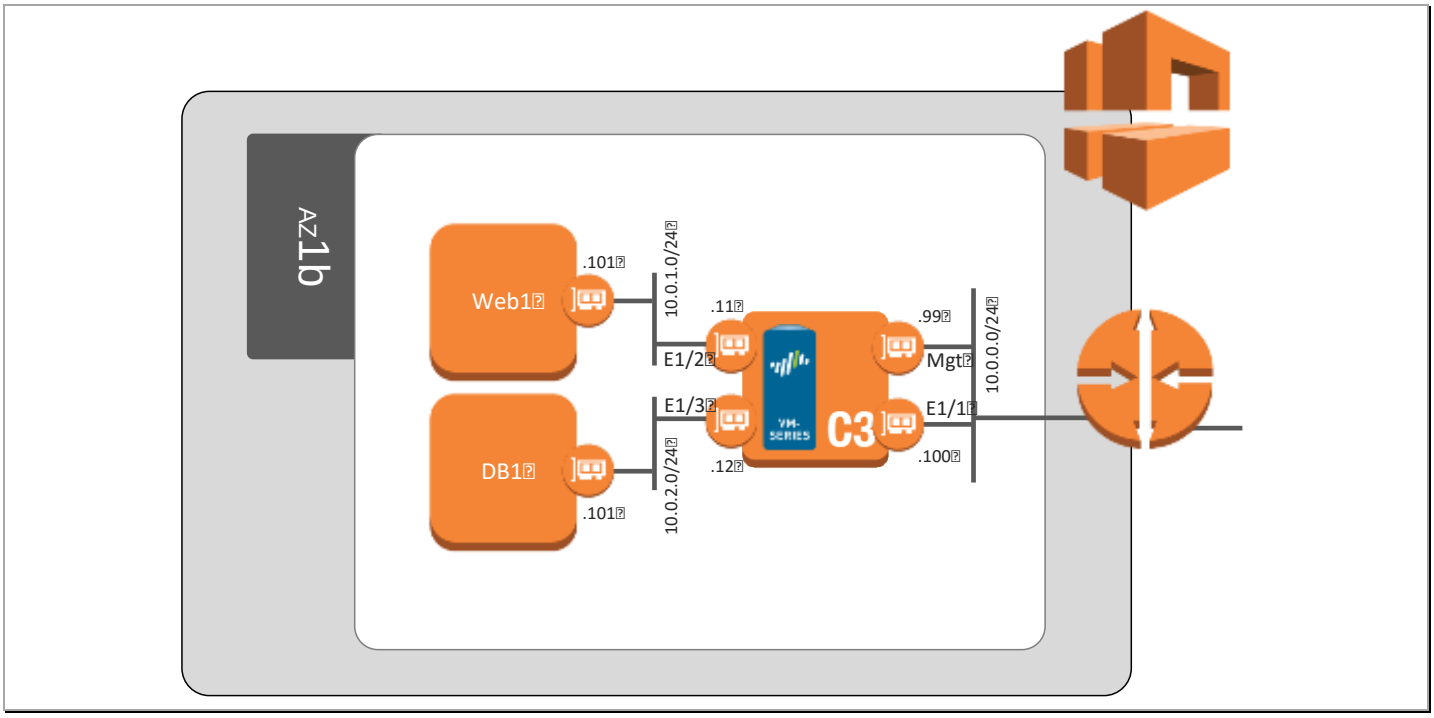
AWS CloudFormation Templates (CFTs), are JSON files that can launch nearly all AWS resources including VPCs, subnets, security groups, route tables, plus many more. AWS CFTs are used for ease of deployment and are key to any auto-scaling environment.

For more information on CFTs and sample CFTs refer to Amazon's documentation

https://aws.amazon.com/cloudformation/aws-cloudformation-templates/

There are also many sample templates available here https://docs.aws.amazon.com/AWSCloudFormation/latest/UserGuide/sample-templates-services-us-west-2.html

This document will explain how to deploy a sample CFT that launches everything that is shown below. This includes, a WordPress server, a MySQL server, a VM-Series firewall and the subnets. In addition, the firewall uses a native bootstrapping feature that allows for additional configuration of the firewall (such as routes, security policies, etc.) Once the sample template has been deployed, the network topology should align with the following:



2. Support Policy

This CFT is released under an as-is, best effort, support policy. These scripts should be seen as community supported and Palo Alto Networks will contribute our expertise as and when possible.

We do not provide technical support or help in using or troubleshooting the components of the project through our normal support options such as Palo Alto Networks support teams, or ASC (Authorized Support Centers) partners and backline support options. The underlying product used (the VM-Series firewall) by the scripts or templates are still supported, but the support is only for the product functionality and not for help in deploying or using the template or script itself.

Unless explicitly tagged, all projects or work posted in our GitHub repository (at https://github.com/PaloAltoNetworks/aws) or sites other than our official Downloads page on https://support.paloaltonetworks.com are provided under the best effort policy.

3. Instances used

When using this sample CFT the following instance types are used:

Instance name	Instance type
WordPress Web Server	t1.micro
WordPress DB Server	t1.micro
VM Series Firewall Bundle 2	c3.xlarge
Security controller	t2.micro

Note: There are costs associated with each instance type launched, please refer to the Amazon EC2 pricing page https://aws.amazon.com/ec2/pricing/

4. Prerequisites

Here are the prerequisites required to successfully launch this template.

4.1 Create an AWS account

If you do not have an AWS account already, go to https://aws.amazon.com/console/ and create an account.

4.2 Add a credit card to your AWS account

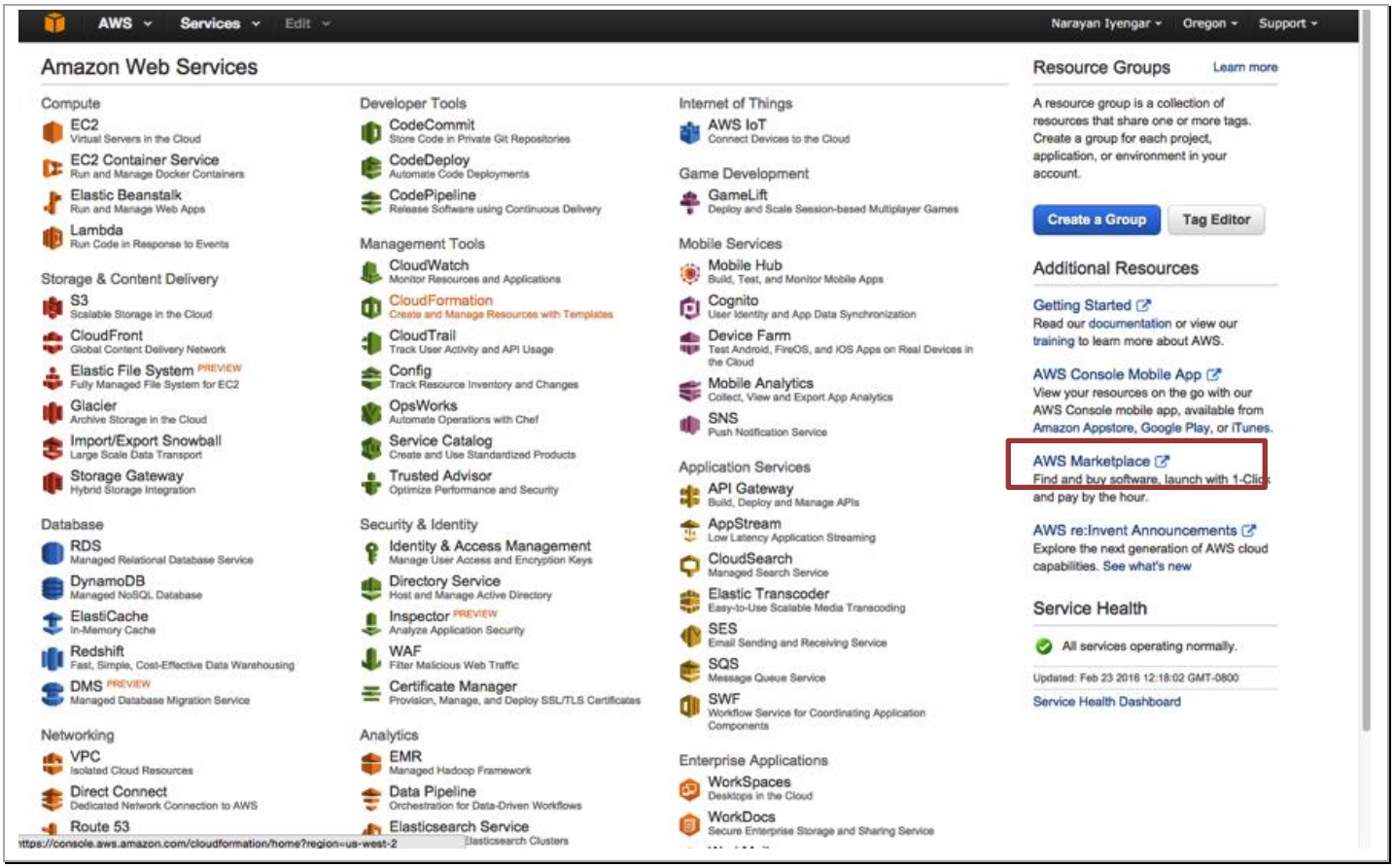
In order to continue you will need to add a method of payment to your AWS account. Use the following https://console.aws.amazon.com/billing/home#/paymentmethods

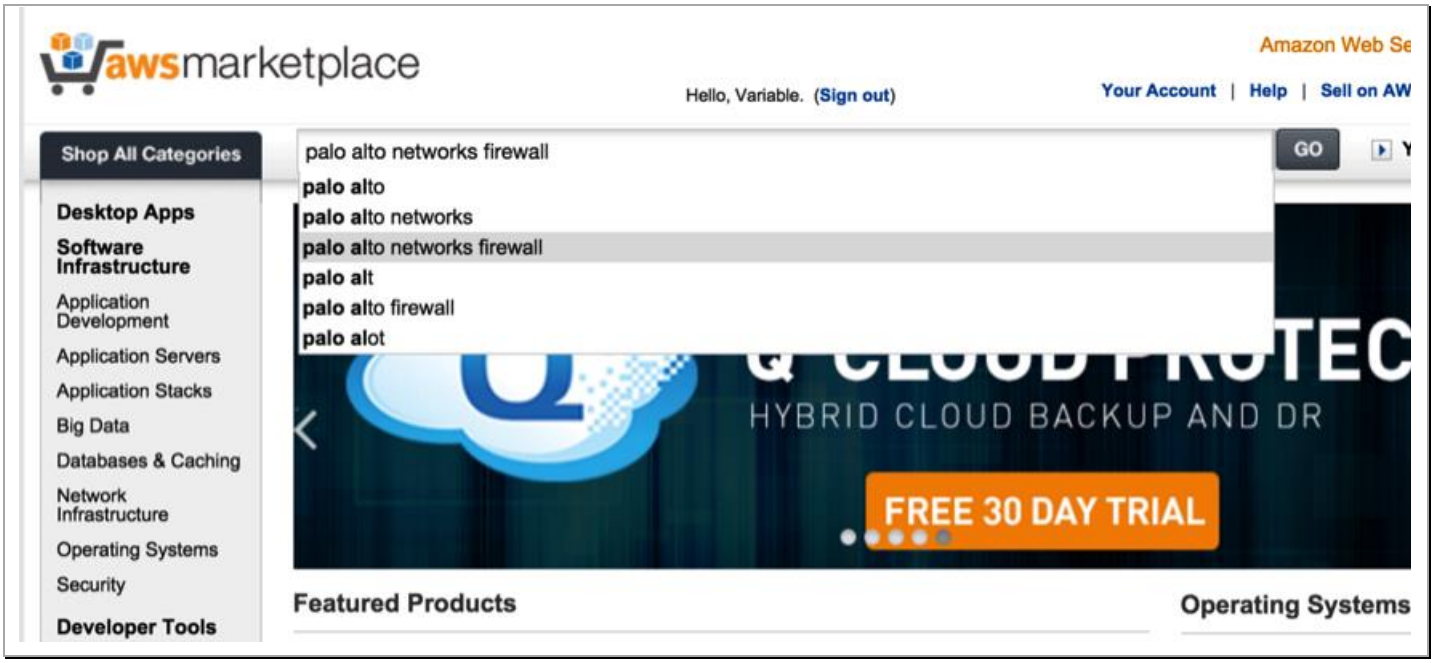
If creating a new account, you may receive a phone call from AWS for verification purposes.

4.3 Review and accept the EULA

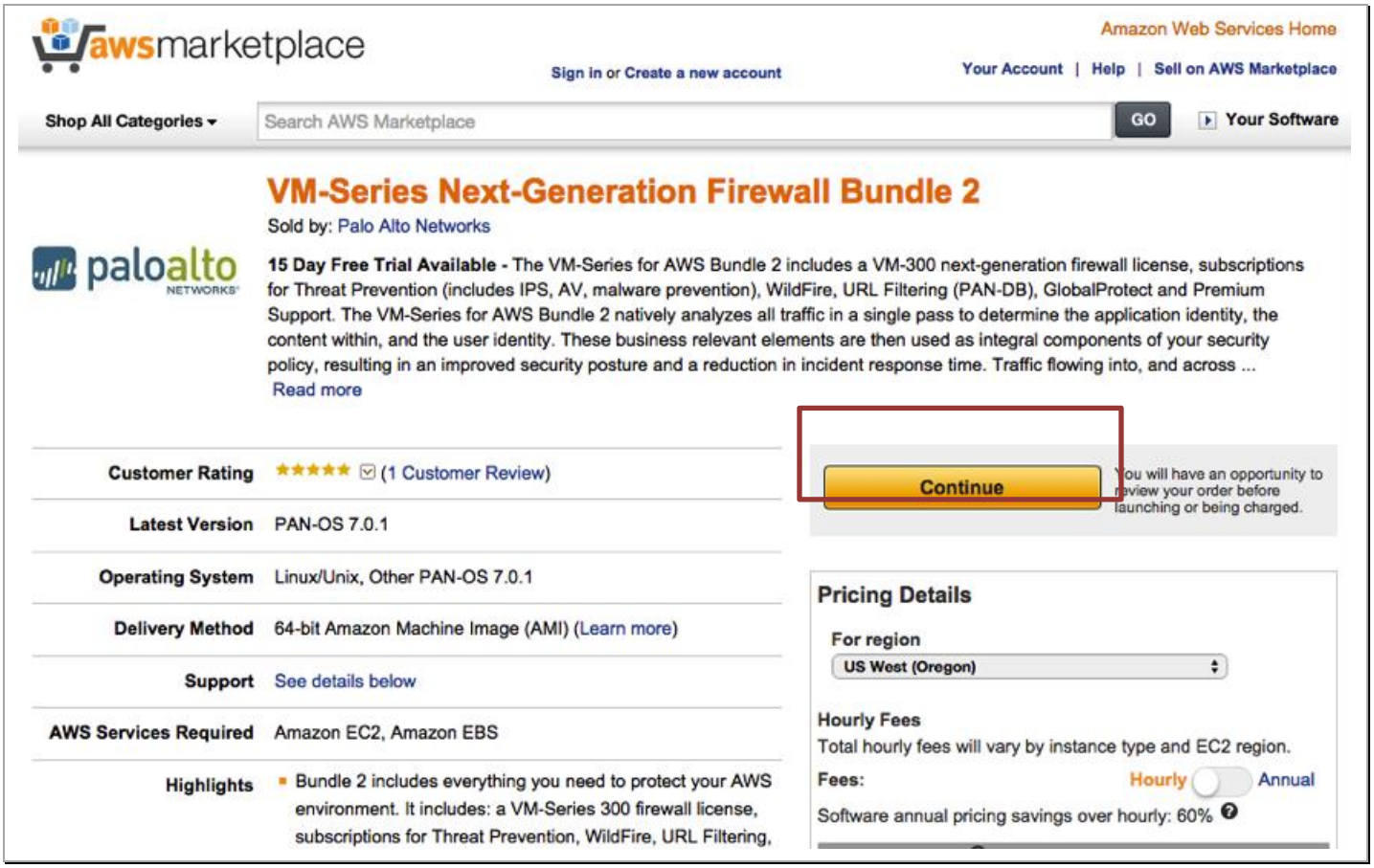
If this is your first time using AWS to launch a VM-Series firewall bundle, you will need to review and accept the software license agreement for the VM-Series.

Click on **AWS Marketplace** and search for **Palo Alto Networks firewall**:

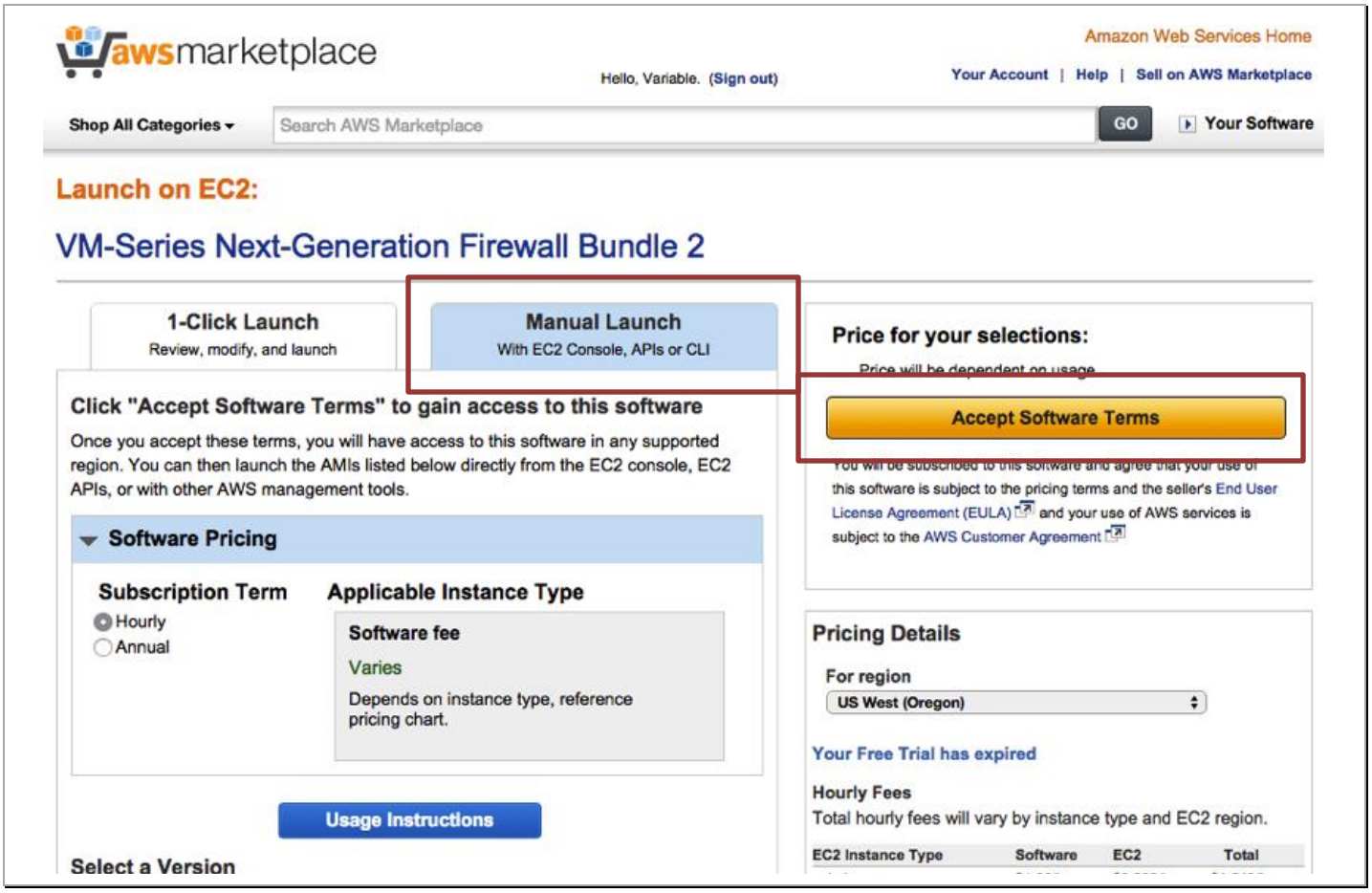




Select **VM-Series Next Generation Firewall Bundle 2**

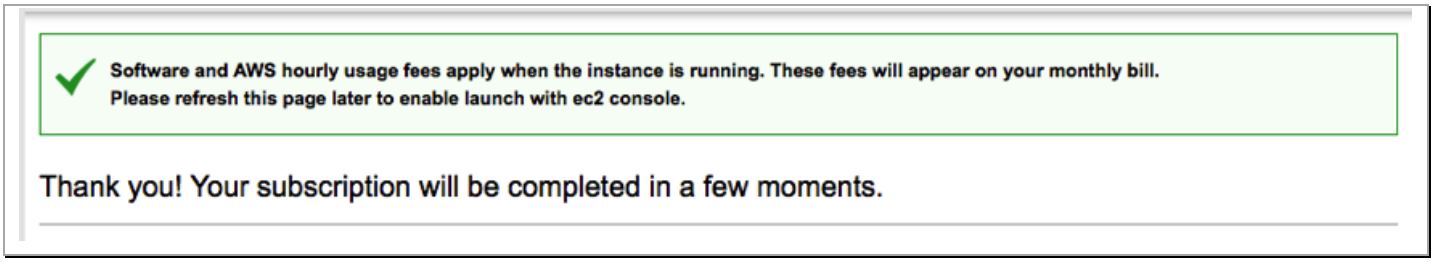


Click **Continue**.



Click on **Manual** Launch, Review the agreement and then click **Accept Software Terms**

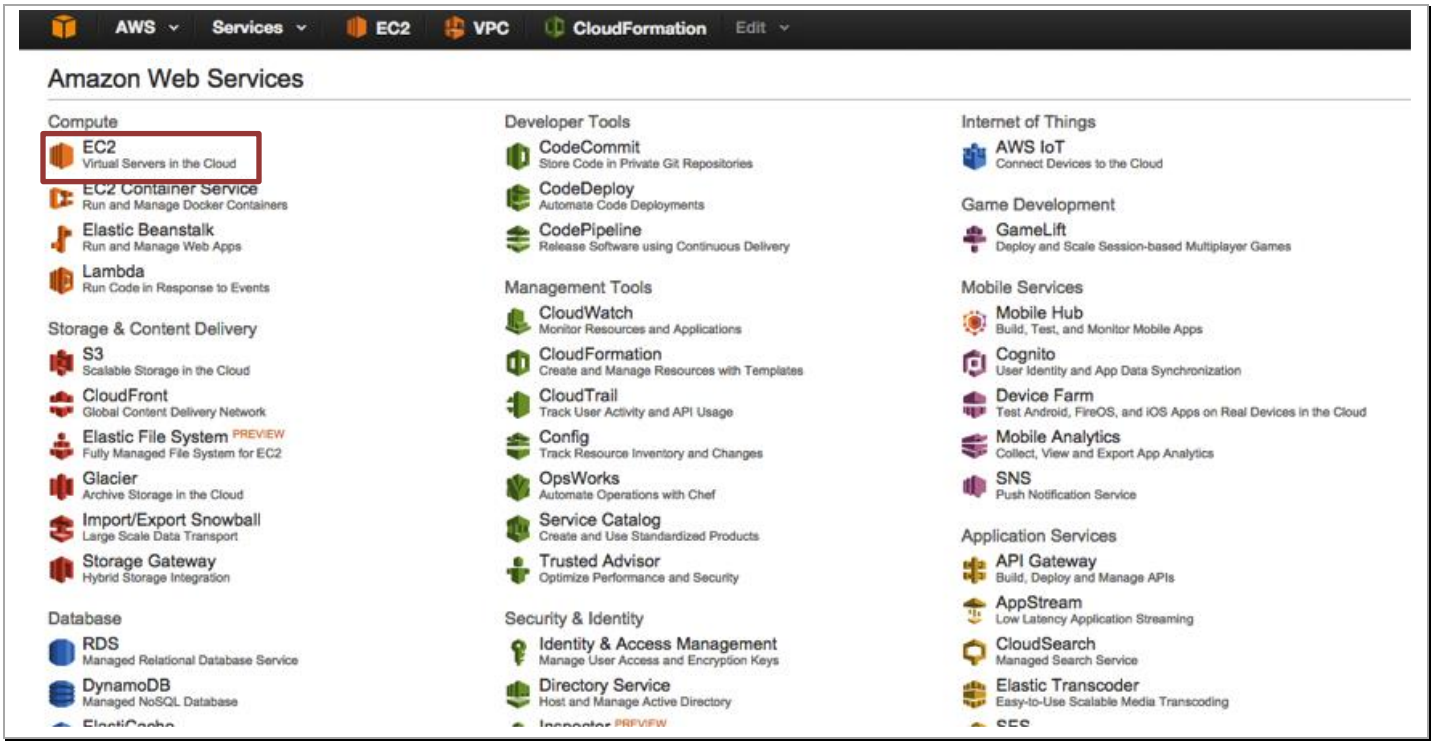
You should see this screen:



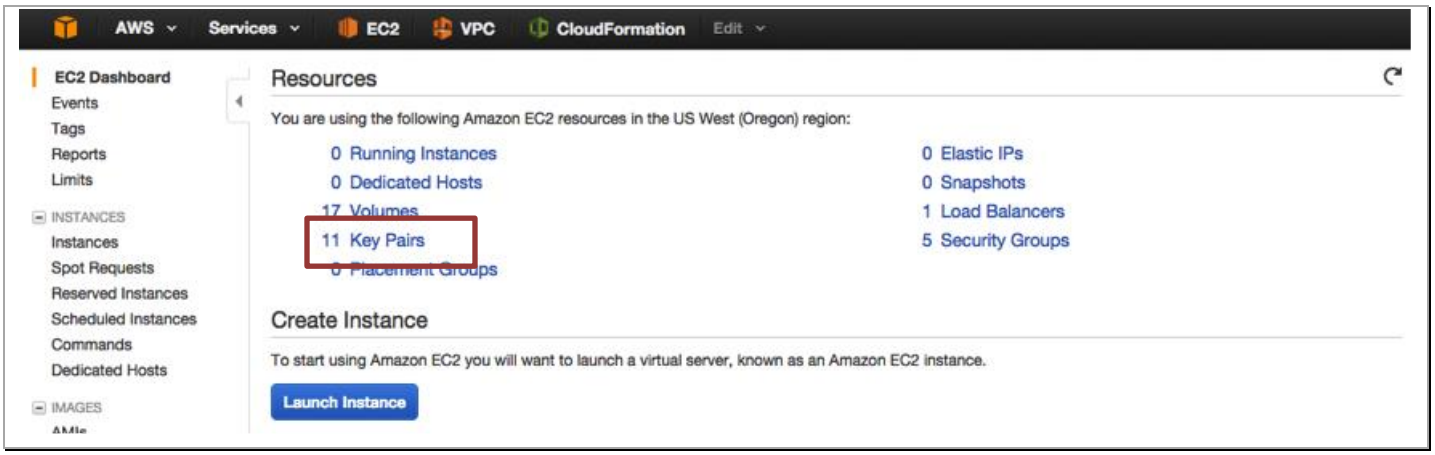
You can now close the browser tab or window and continue with the next step.

4.4 Create and download an SSH keypair

Sign into the AWS console https://www.amazon.com and click on **EC2**

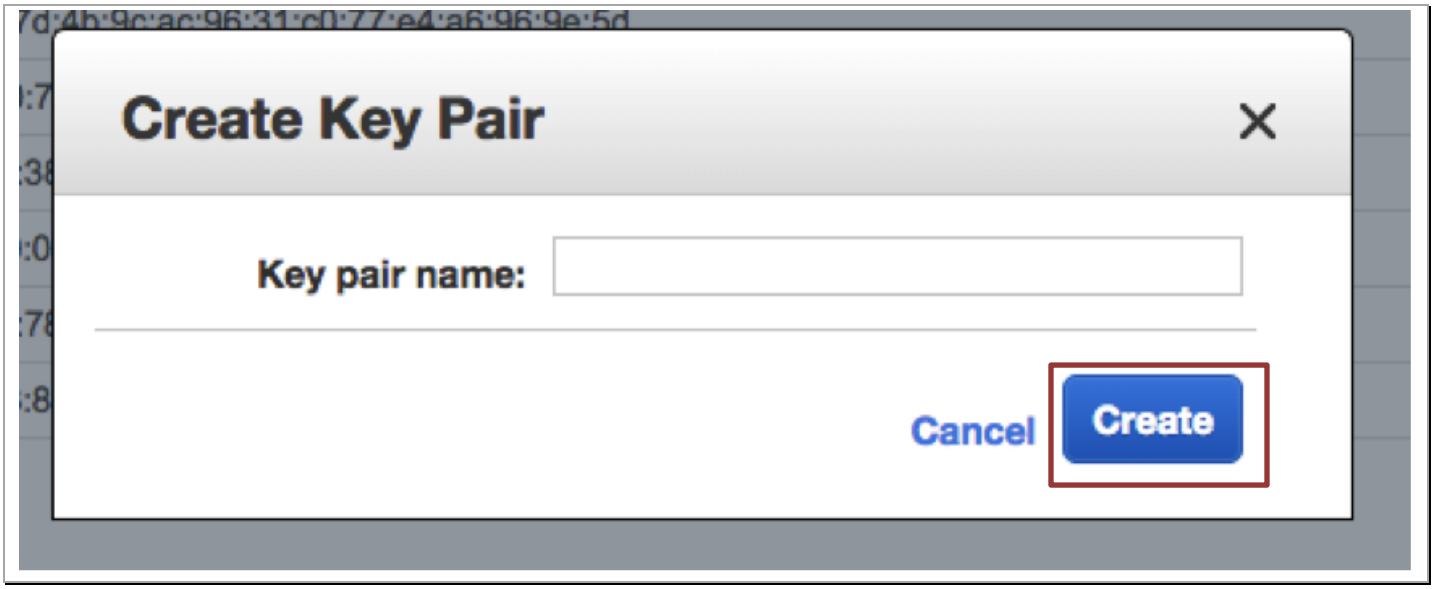


Click **KeyPairs**



Click **Create Key Pair**

Give it a name

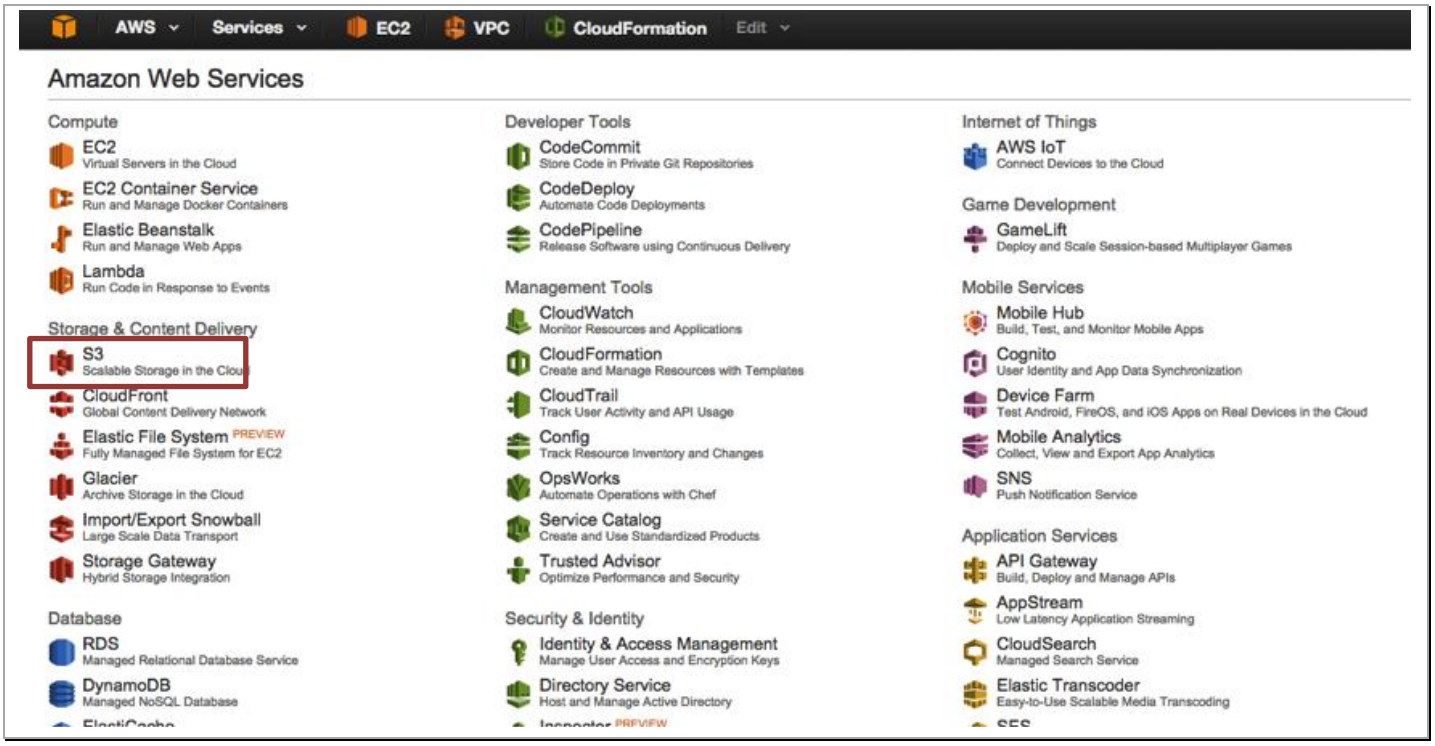


And click **Create**. This should now prompt you to save the just generated private key. Save the key.

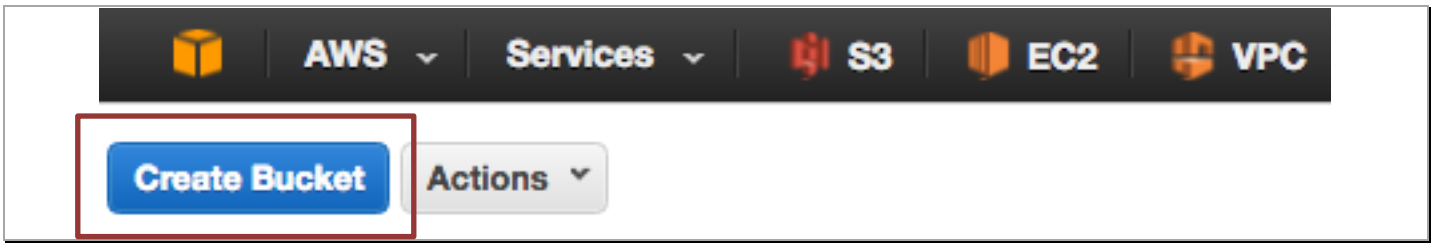
4.5 Create a Bootstrap Bucket

Bootstrapping is a feature of the VM-Series firewall that allows you to load a pre-defined configuration into the firewall during boot-up. This ensures that the firewall is configured and ready at initial boot-up, thereby removing the need for manual configuration. The bootstrapping feature also enables automating deployment of the VM-Series.

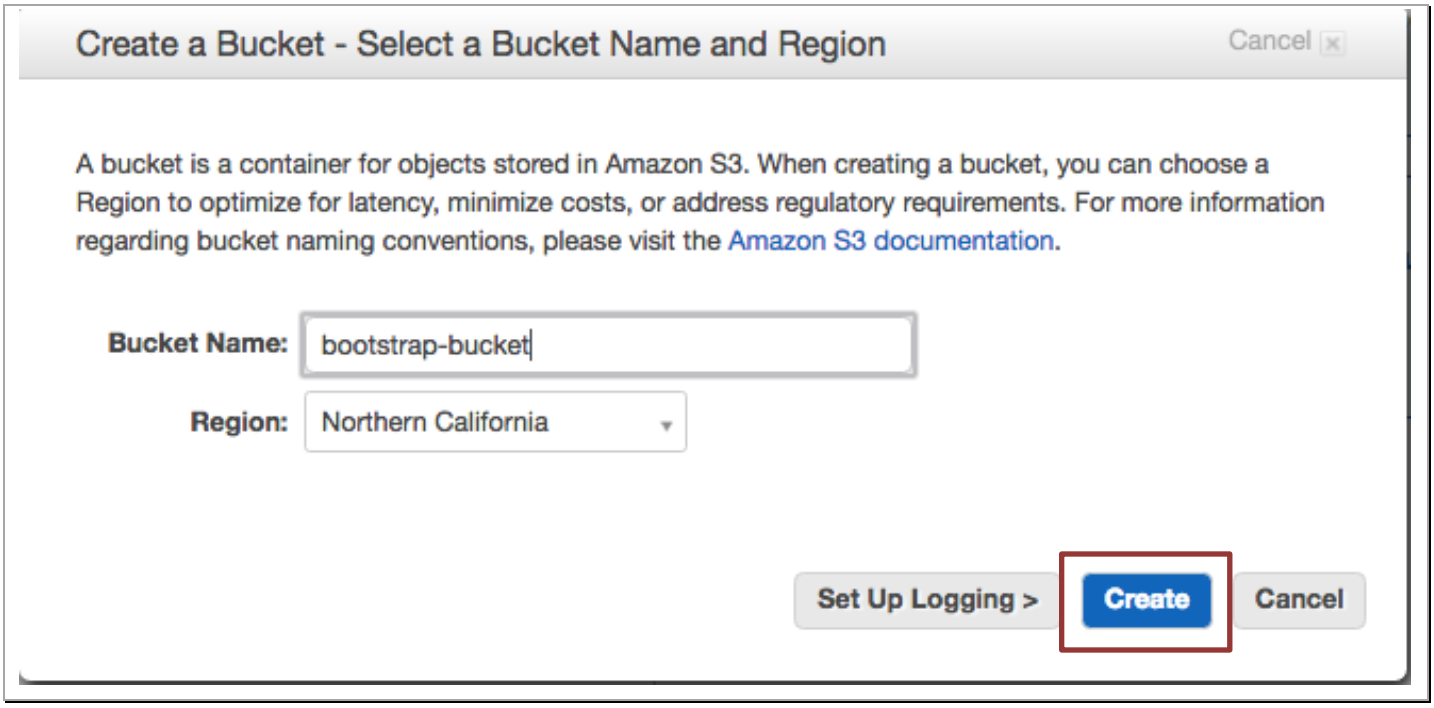
In order to create a Bootstrap bucket, Sign into the AWS console https://www.amazon.com and click on **S3**



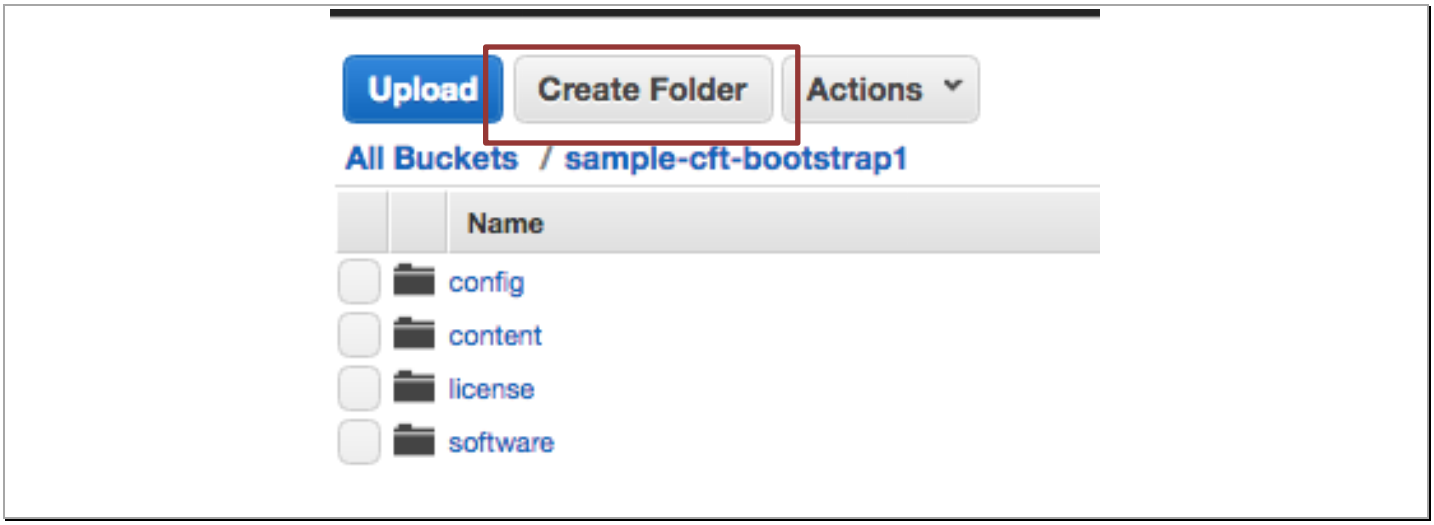
Click **Create Bucket**:



Enter a bucket name and select a region and click **Create**:



You will need to enter a globally unique bucket name. AWS will warn you if the name is not unique. Once the bucket is created, click on the newly created bucket and add four folders called **config, license, software** and **content** by clicking on **Create Folder**:



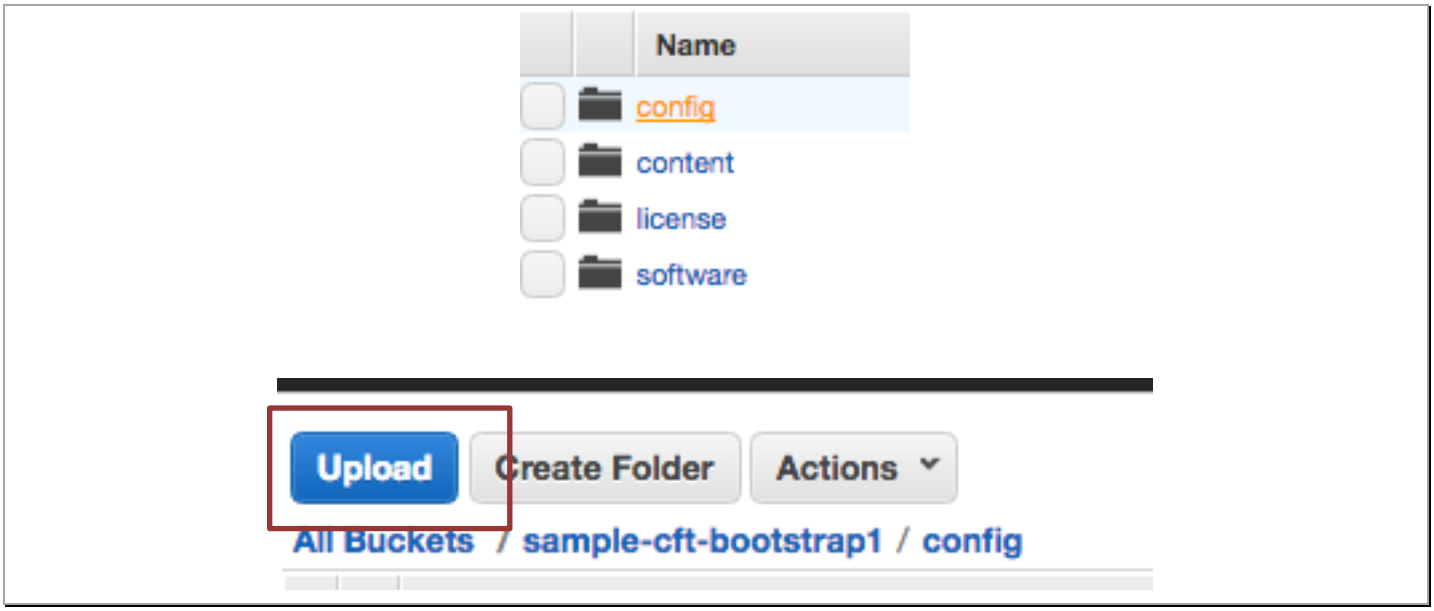
Download the following files and save them in a known location:

https://raw.githubusercontent.com/PaloAltoNetworks/aws/master/two-tier sample/bootstrap/bootstrap.xml

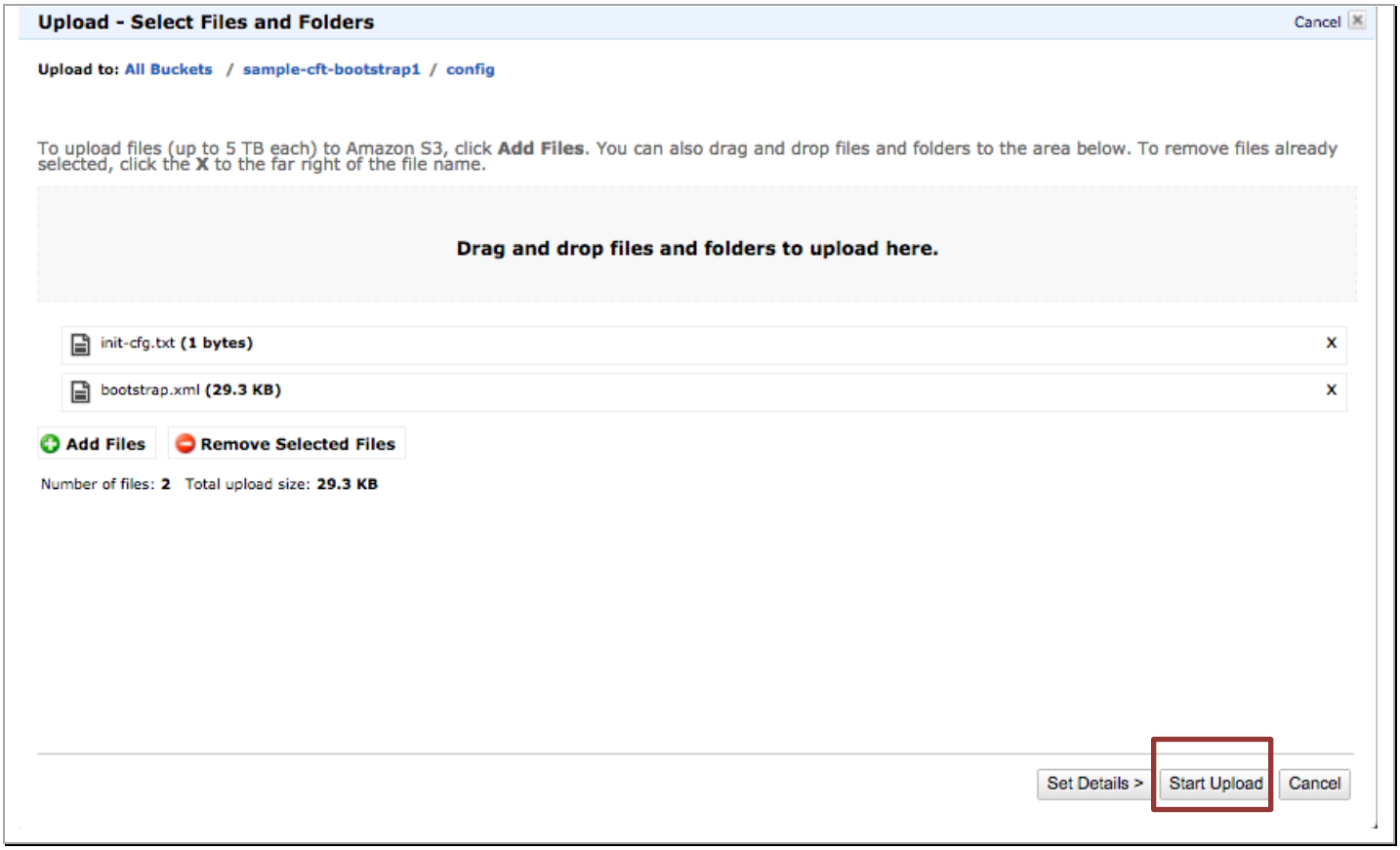
https://raw.githubusercontent.com/PaloAltoNetworks/aws/master/two-tier sample/bootstrap/init-cfg.txt

https://github.com/PaloAltoNetworks/aws/raw/master/two-tier_sample/bootstrap/panupv2-all-contents-600-3449

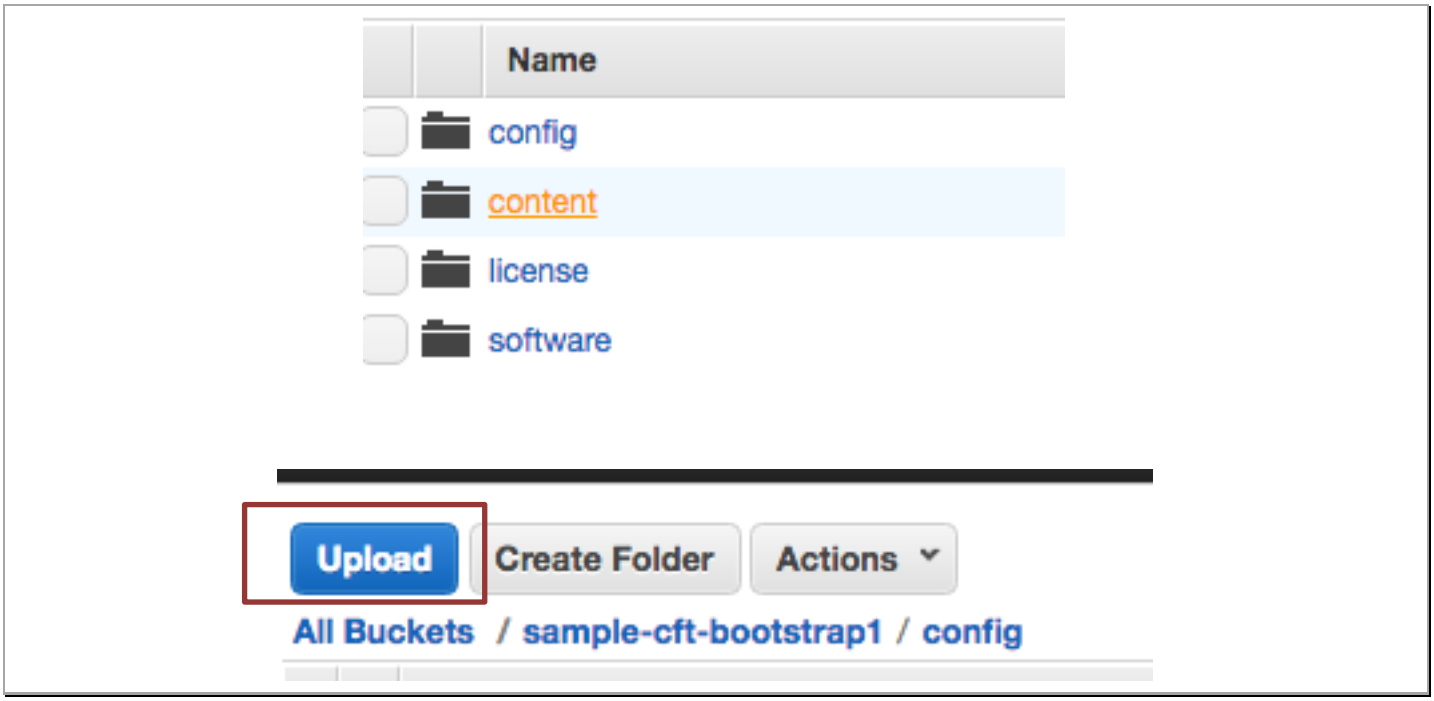
Now click on the **config** folder in the **S3** console and click **Upload:**



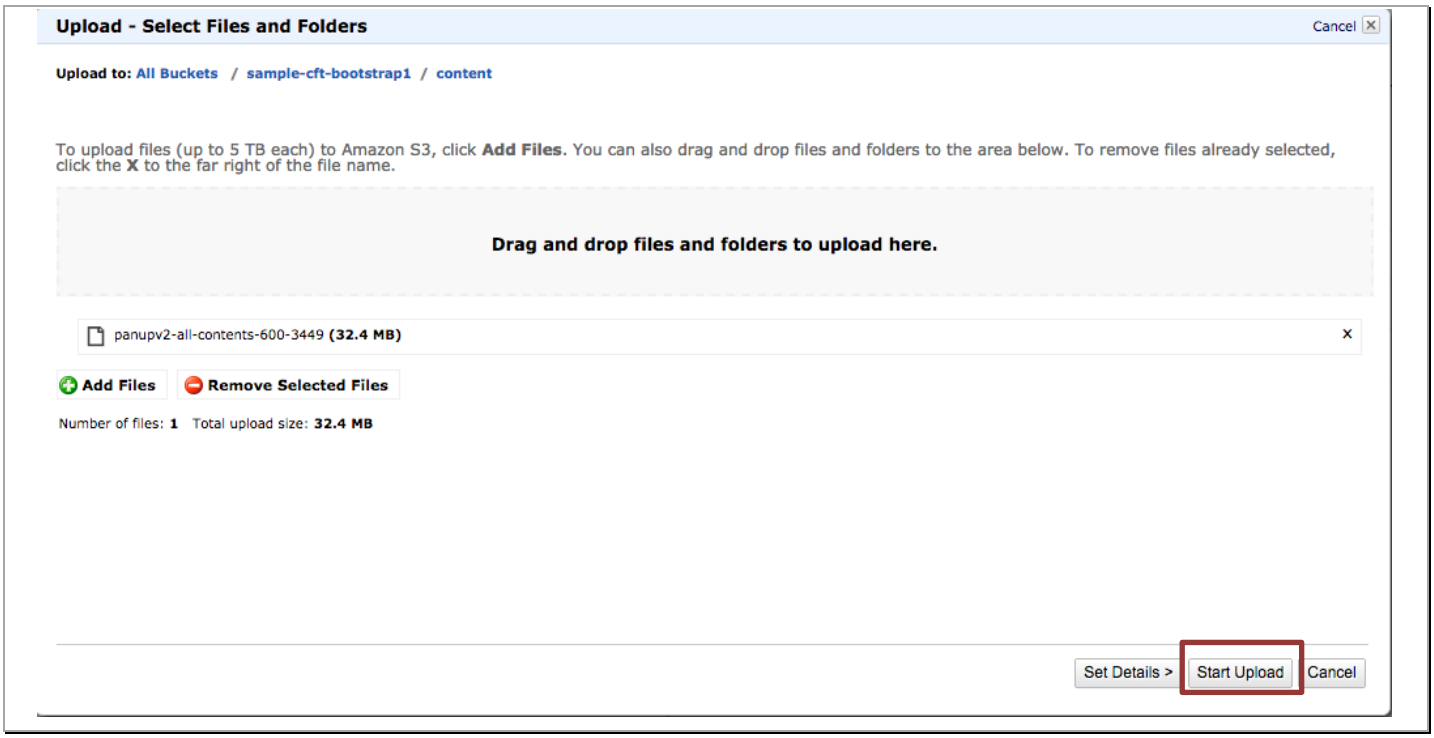
Select **Add Files** and select the two files (bootstrap.xml and init-cft.txt) downloaded previously and click **Start Upload**:



Now click on the **content** folder ins the **S3** console and click **Upload**:



Select **Add Files** and select the file (panupv2-all-contents-600-3449) downloaded previously and click **Start Upload**:



NOTE: Please create the folders using the console. Creating folders locally on your machine and uploading them may not work as AWS doesn't upload empty folders.

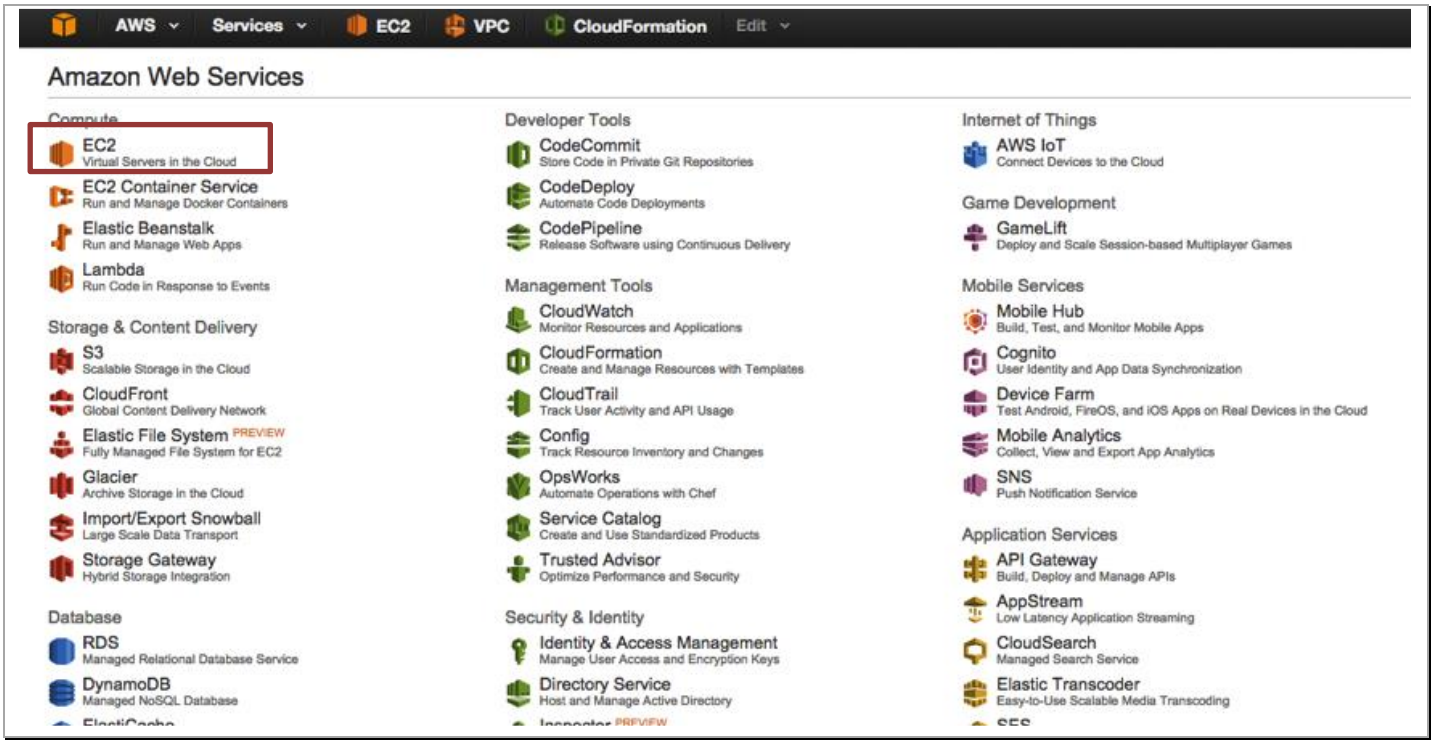
4.6 Download the Template

Download and save the CloudFormation template and save in a known location:

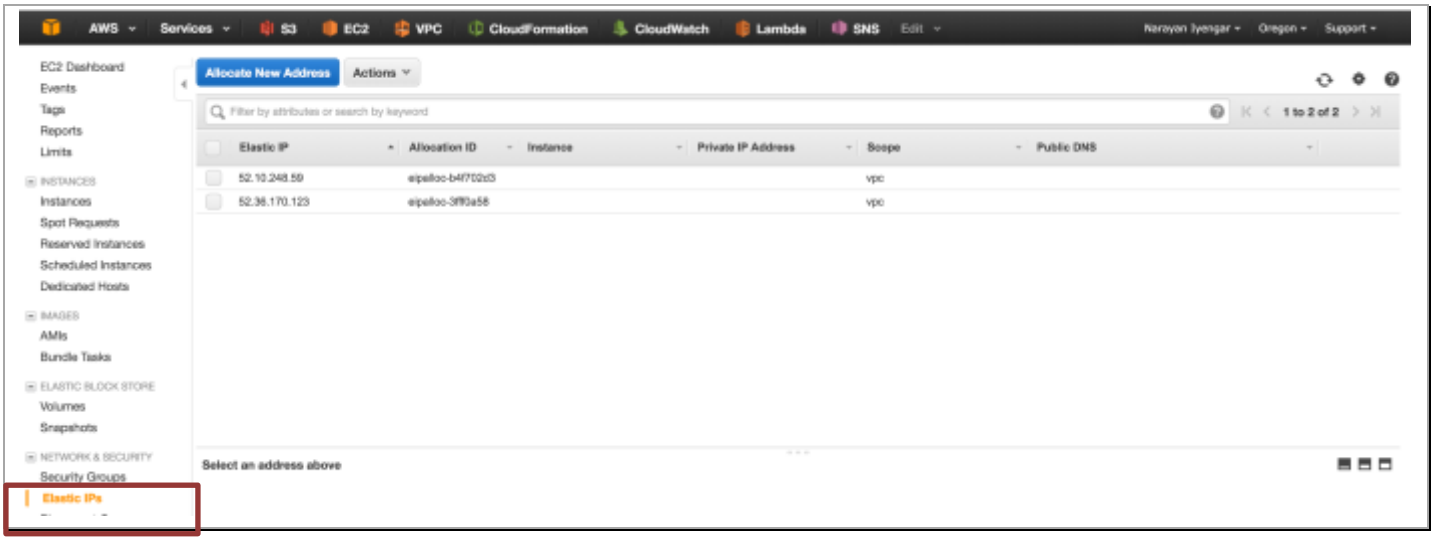
https://raw.githubusercontent.com/PaloAltoNetworks/aws/master/two-tier sample/pan-sample-cft.json

4.7 Check Elastic IPs

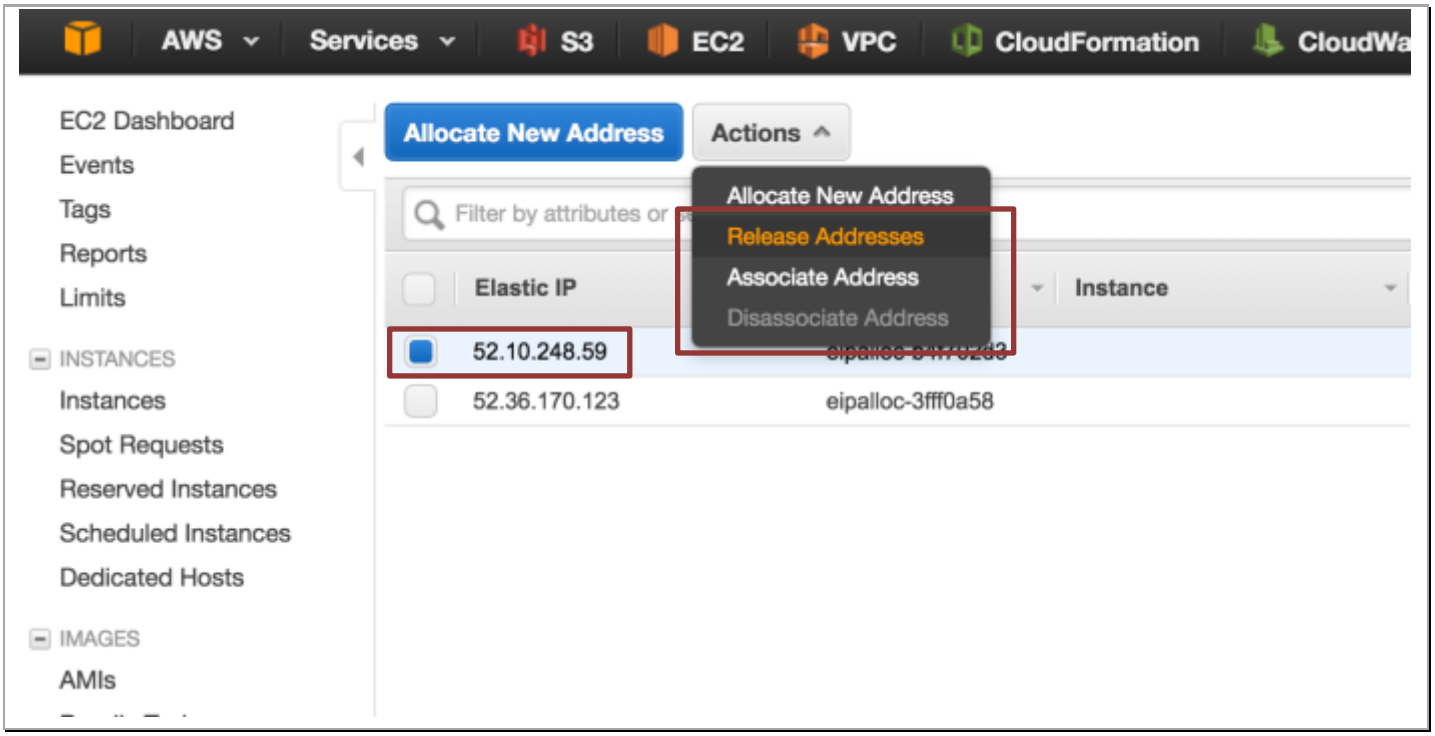
By default, each AWS account has a 5 elastic IP (EIP) limit per region unless a limit increase has been requested (via an AWS support ticket). In order to launch this template, you will need two EIPs. To check any allocated or associated EIPs, on the AWS console click on **EC2:**



And click on Elastic IPs:



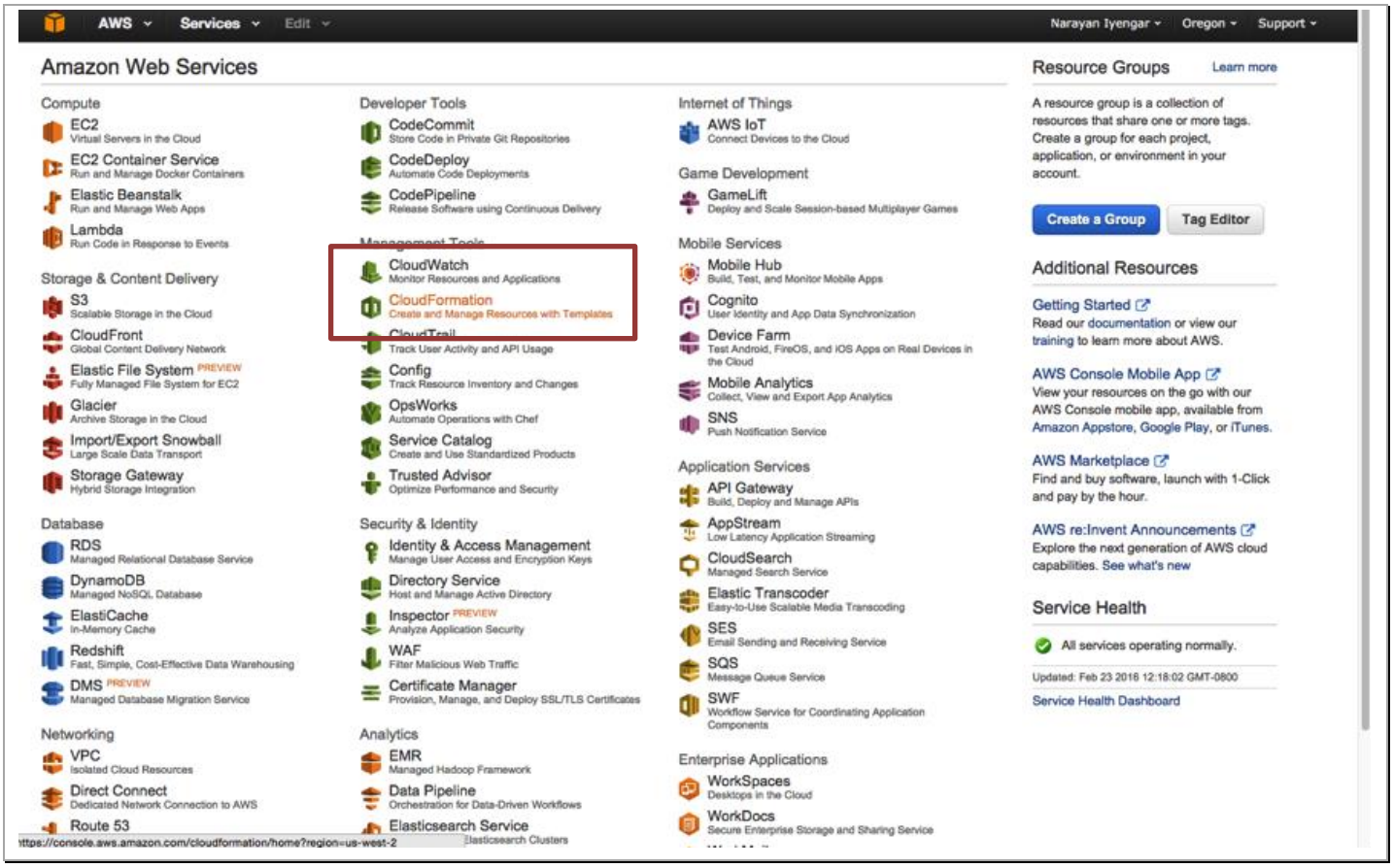
If there are no EIPs allocated, proceed to Section 4. If there are more than 3 EIPs allocated and you have not requested an EIP limit increase, the template launch will fail. You can either release an EIP or request a limit increase via an AWS support ticket. In order to release an allocated EIP, simply click on the EIP and click **Actions**, **Release Addresses**



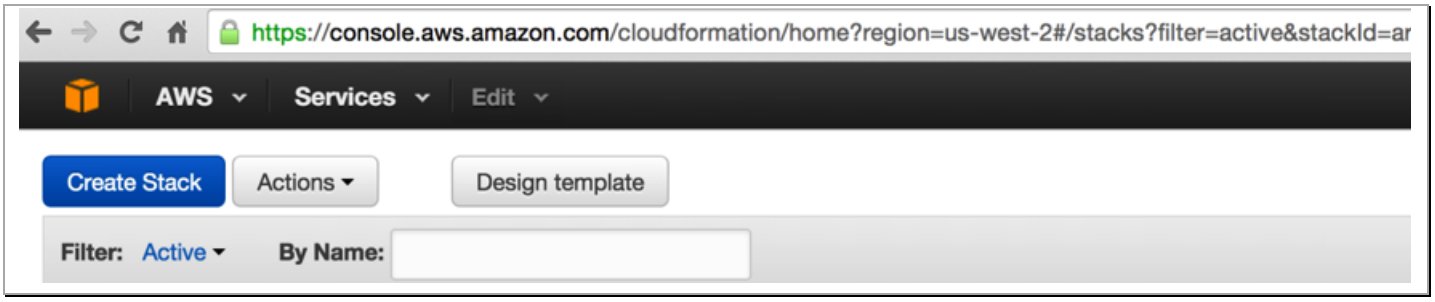
If the EIP is associated with an instance, you will need to disassociate the address first and then release the address. If you are relying on the address for other work, please be aware that disassociating the address and releasing the address could cause work disruption.

5. Launch The CFT

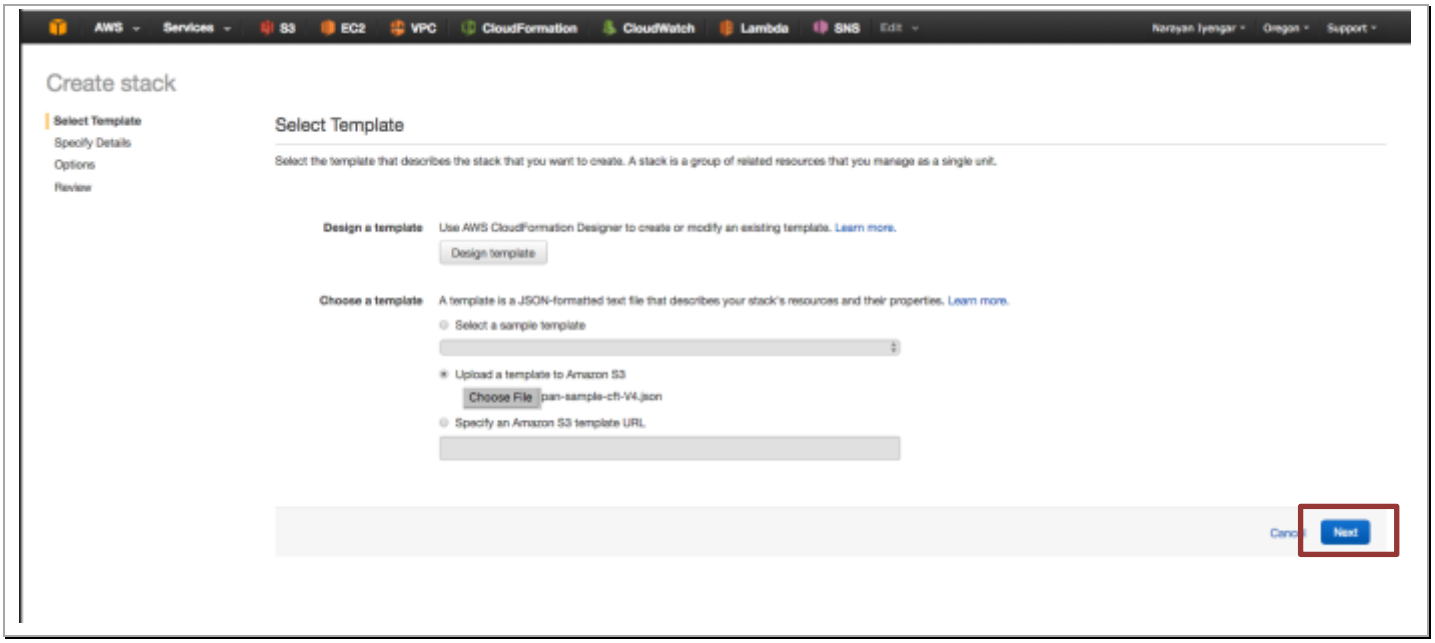
Login in to the AWS console https://console.aws.amazon.com and click on **CloudFormation**



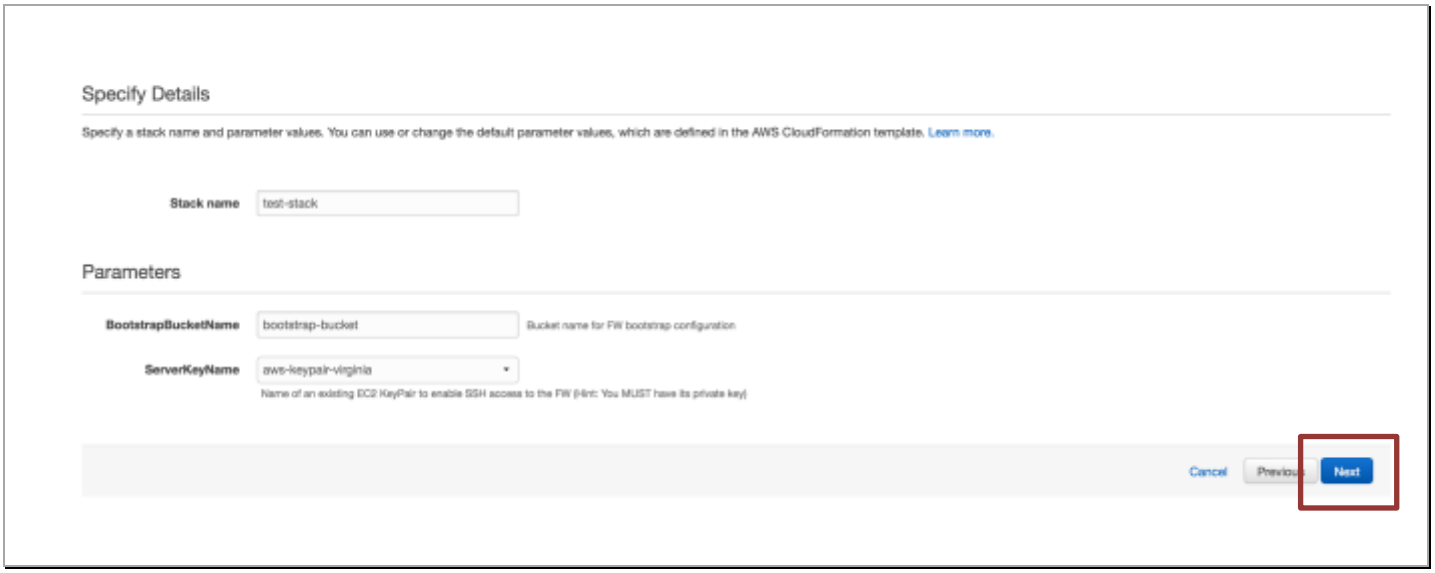
Click **Create Stack**:



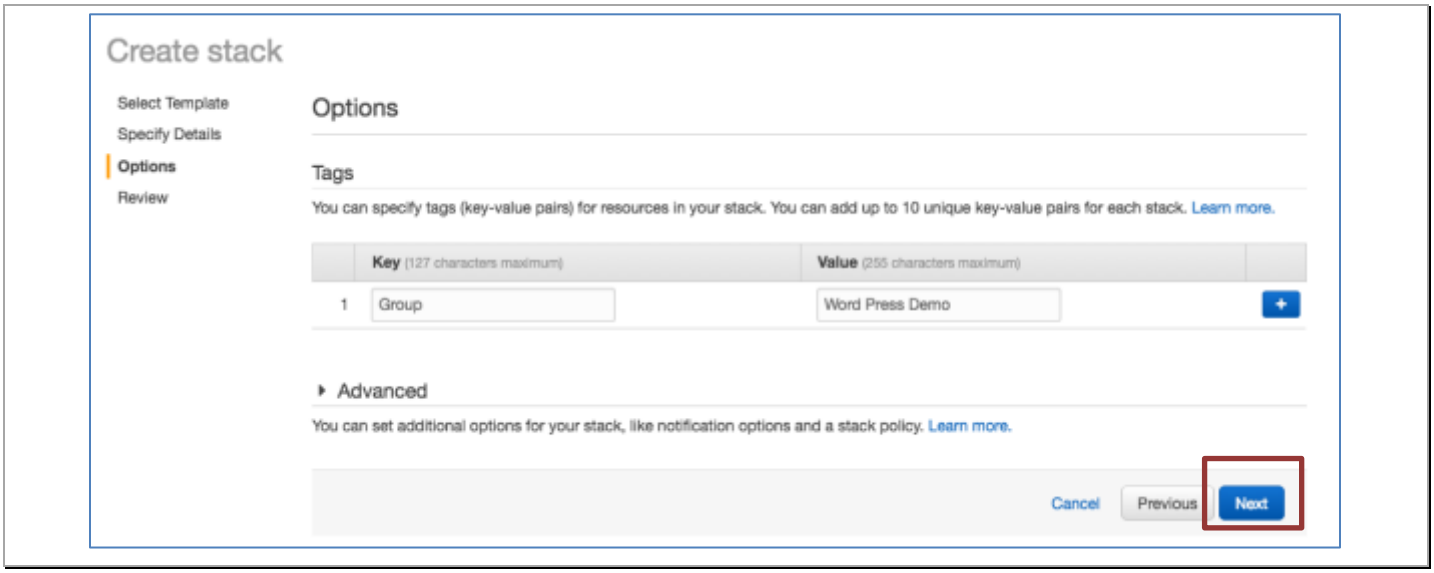
Select "**Choose File**" and select the template downloaded in Section 4.6 into the box and click **Next**:



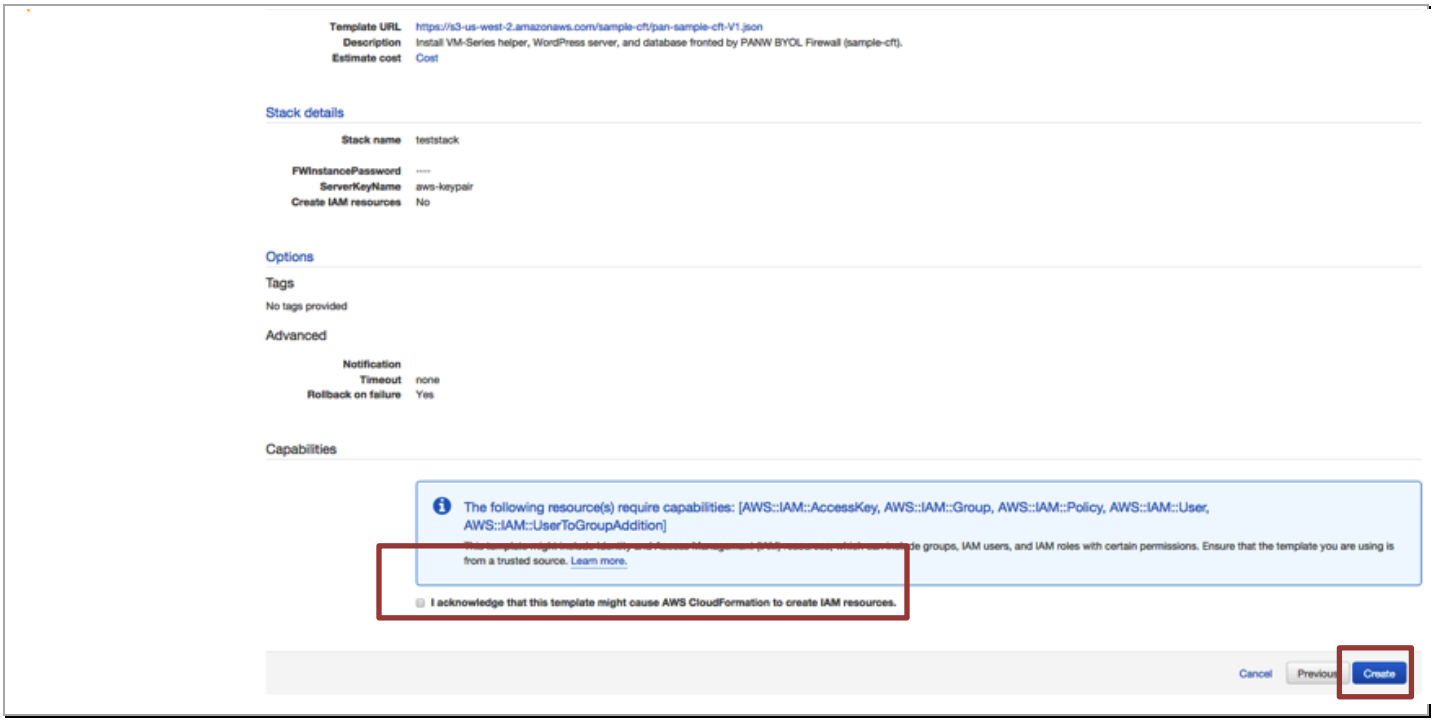
In the next screen specify a "**Stack Name**". This can be anything. In the **Parameters** section, specify the bucket name of the bootstrapping bucket that was created in section 3.5 and select a **Serverkey** for which you have the private key. Refer to section 2.4 on how to generate a keypair. Once satisfied, click **Next**.



On the next screen you can specify tags (optional) otherwise click **Next**. You can create Key Value pairs that allow you to filter instances based on those tags. Tags provide a convenient, filtered view of just the instances launched by the template.

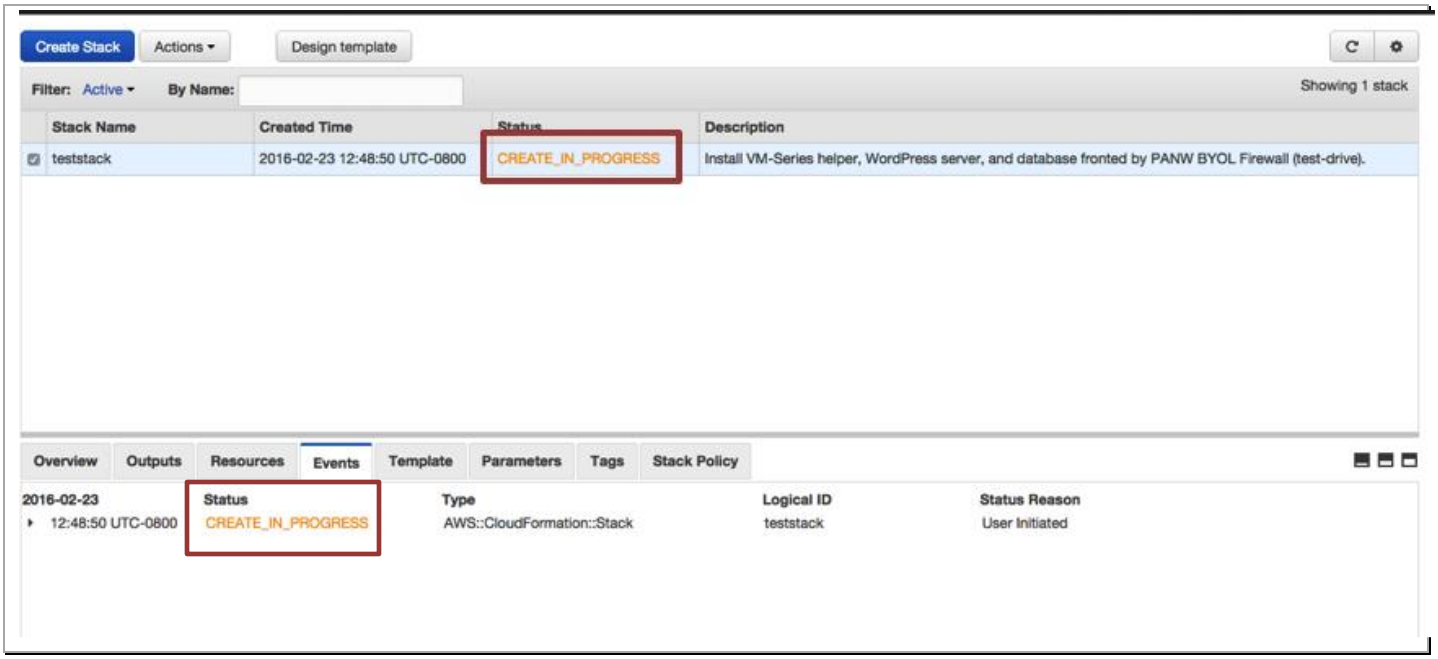


Next, review and check acknowledge at the bottom and click **Create**.

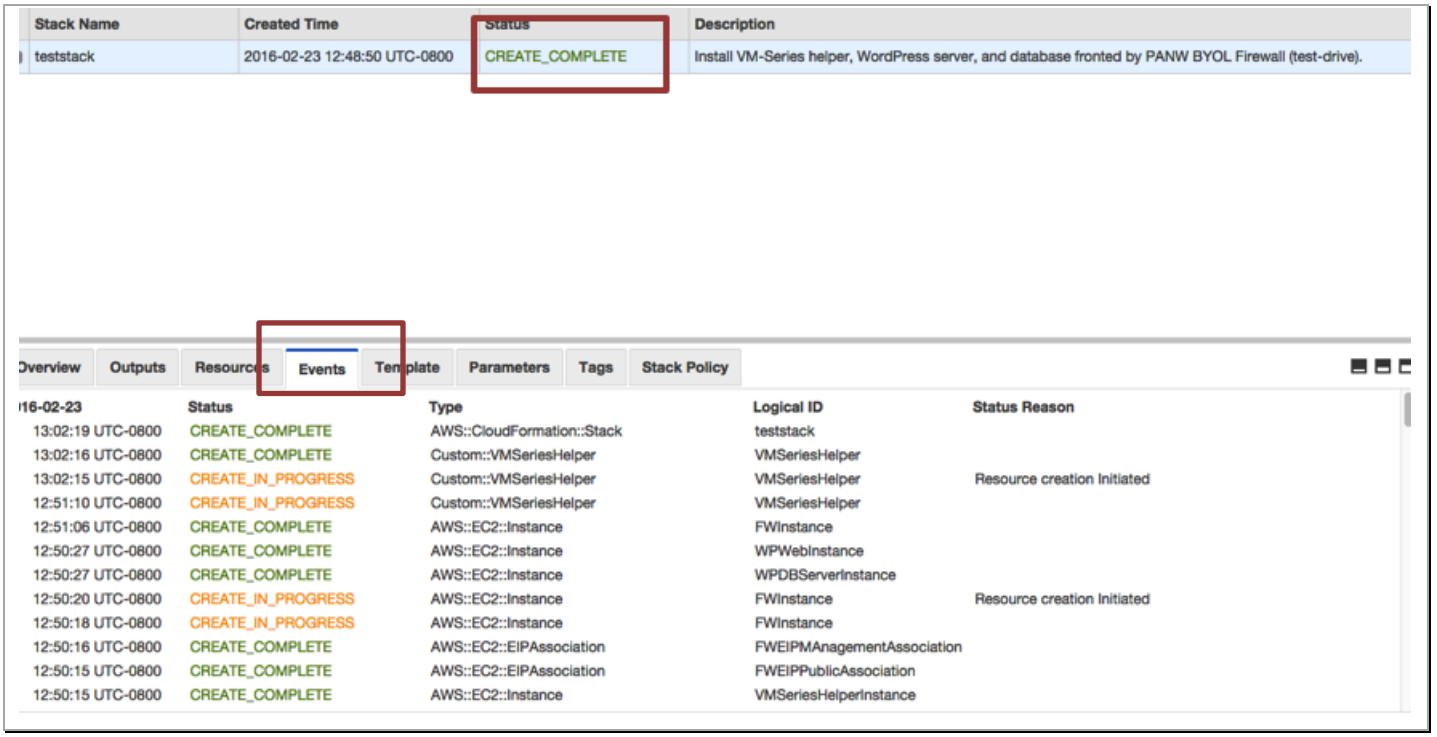


Once launched you should be able to monitor the stack creation progress in the next screen by clicking on the **Events** tab.

Note: The template takes about 10-15 minutes to fully deploy and be operational.

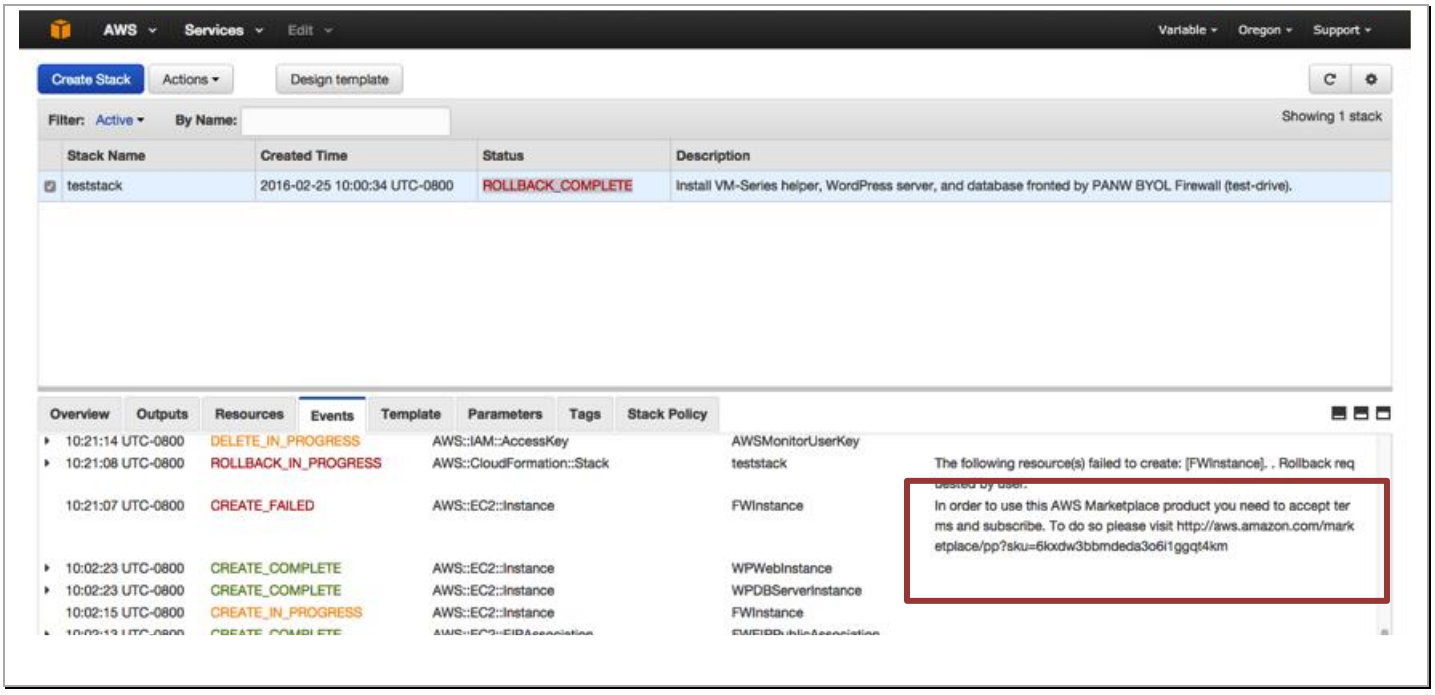


If the CFT was successfully launched, you should see an event as below:



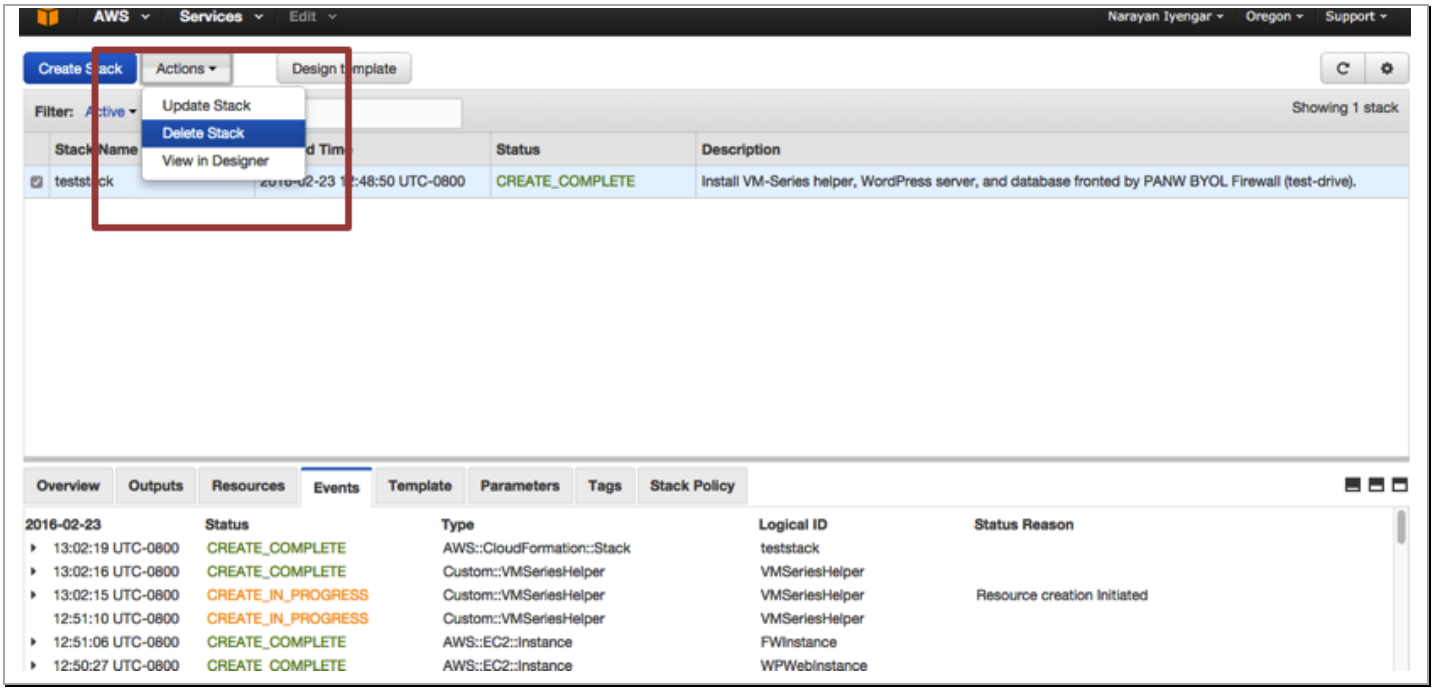
If there were any errors during the creation of the stack, you will need to drill down to the specific event in the **Events** tab and **Outputs** tab to debug and then create a new stack after fixing any errors.

For instance, if you did not accept the VM-Series EULA, then you will get an error as seen below



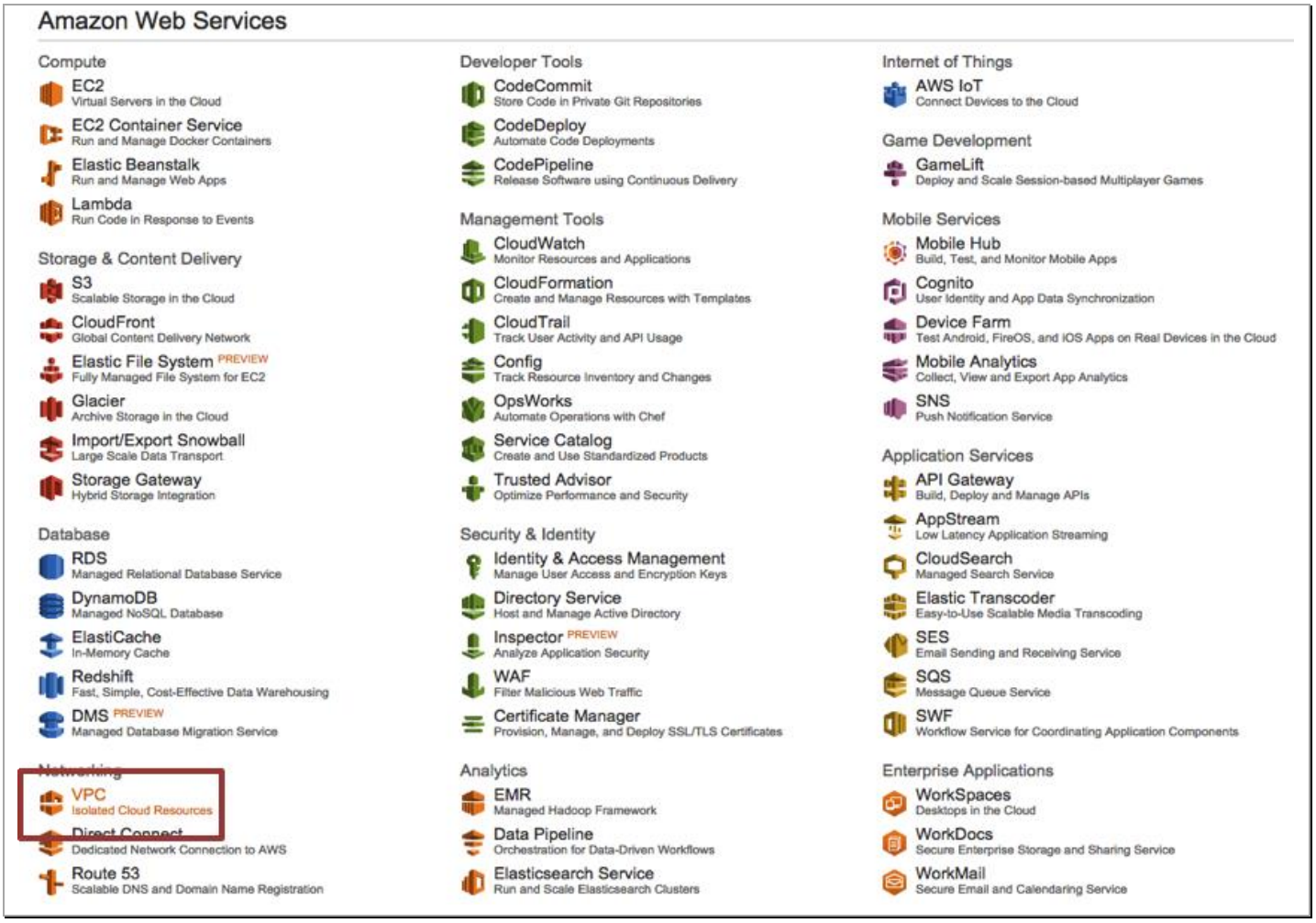
Refer to section 2.3 to review and accept the EULA for the VM-Series NGFW

Note: If you need to relaunch the CFT, first delete the current stack under Actions, Delete Stack.

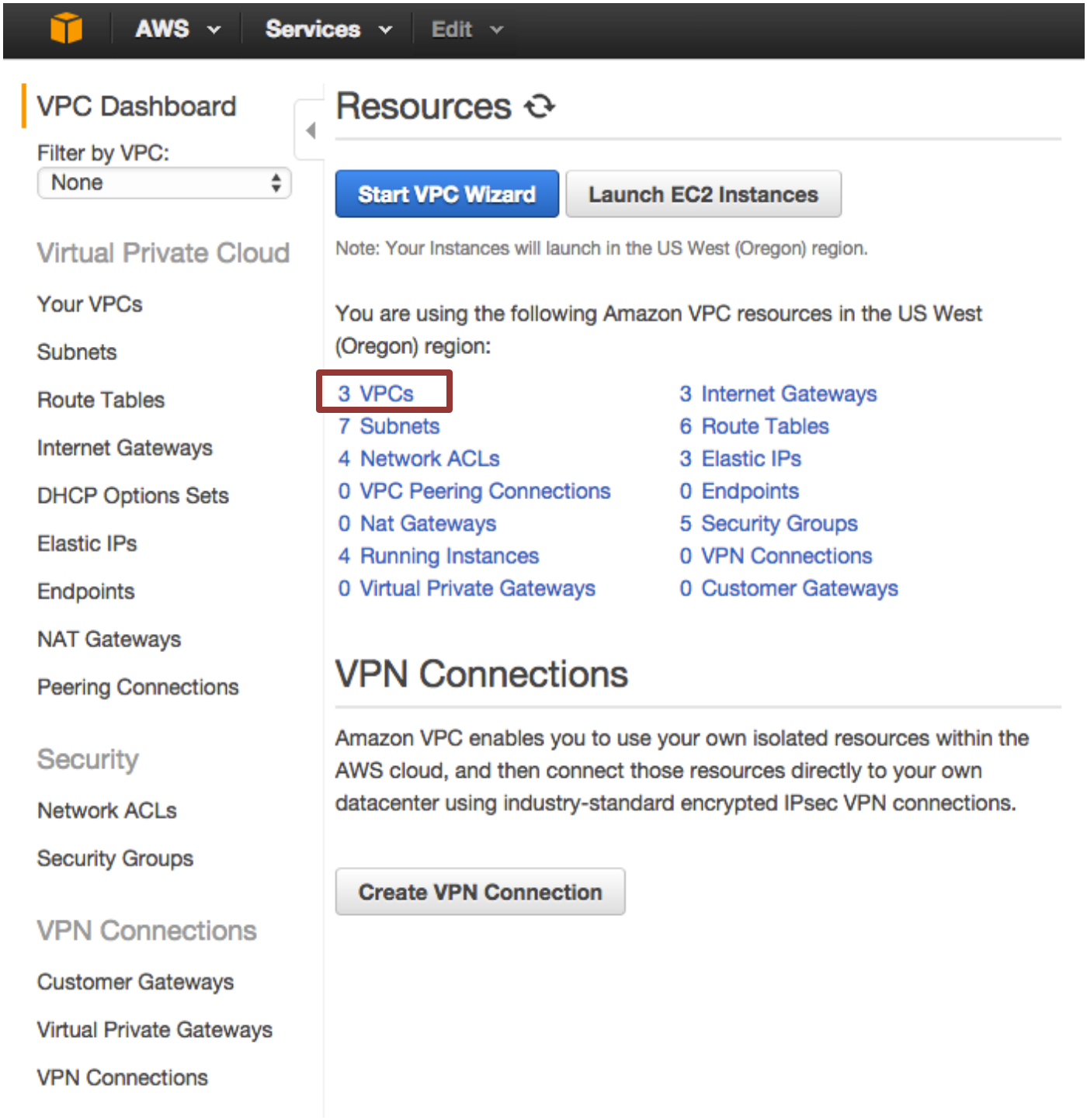


6. Review what was created

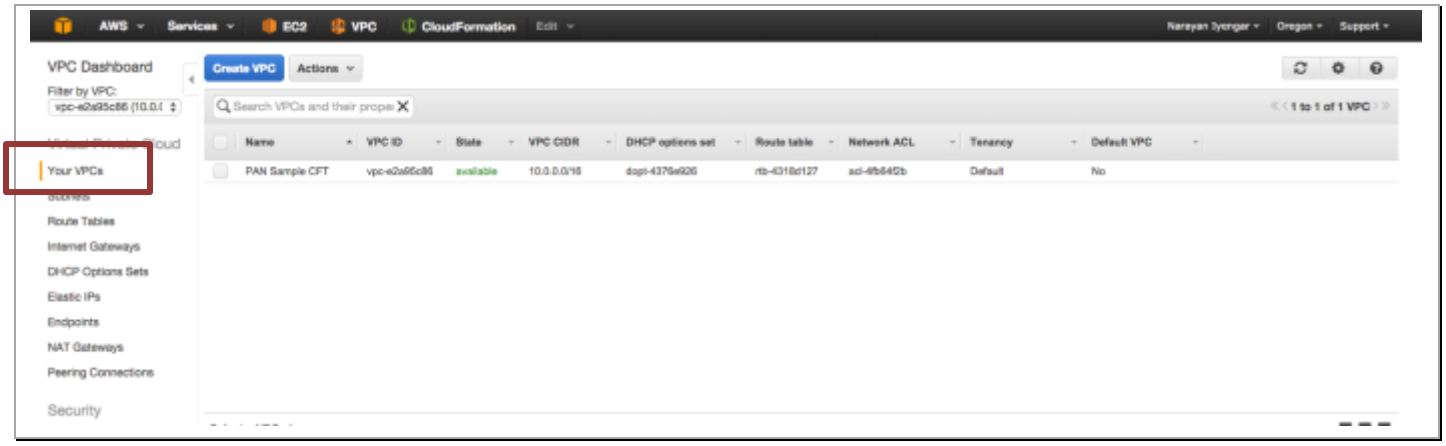
Let's review what the CFT has launched. The newly created VPC can be accessed via:



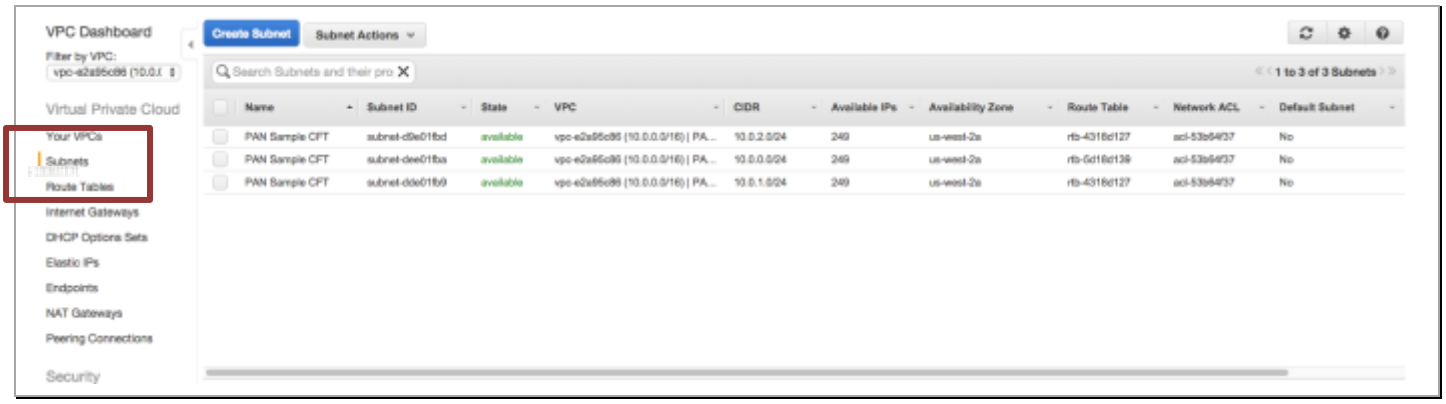
Here you should see all VPCs created in your account:



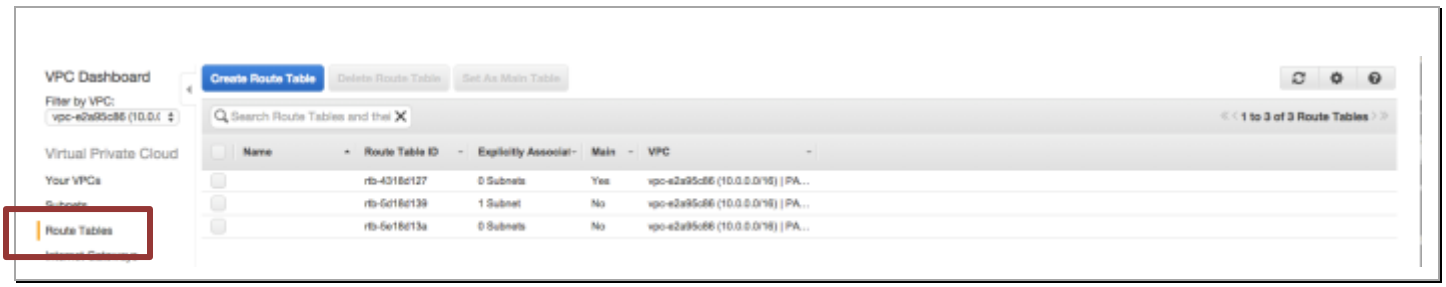
Here is the sample **VPC**:



On the left you can review **subnets**:

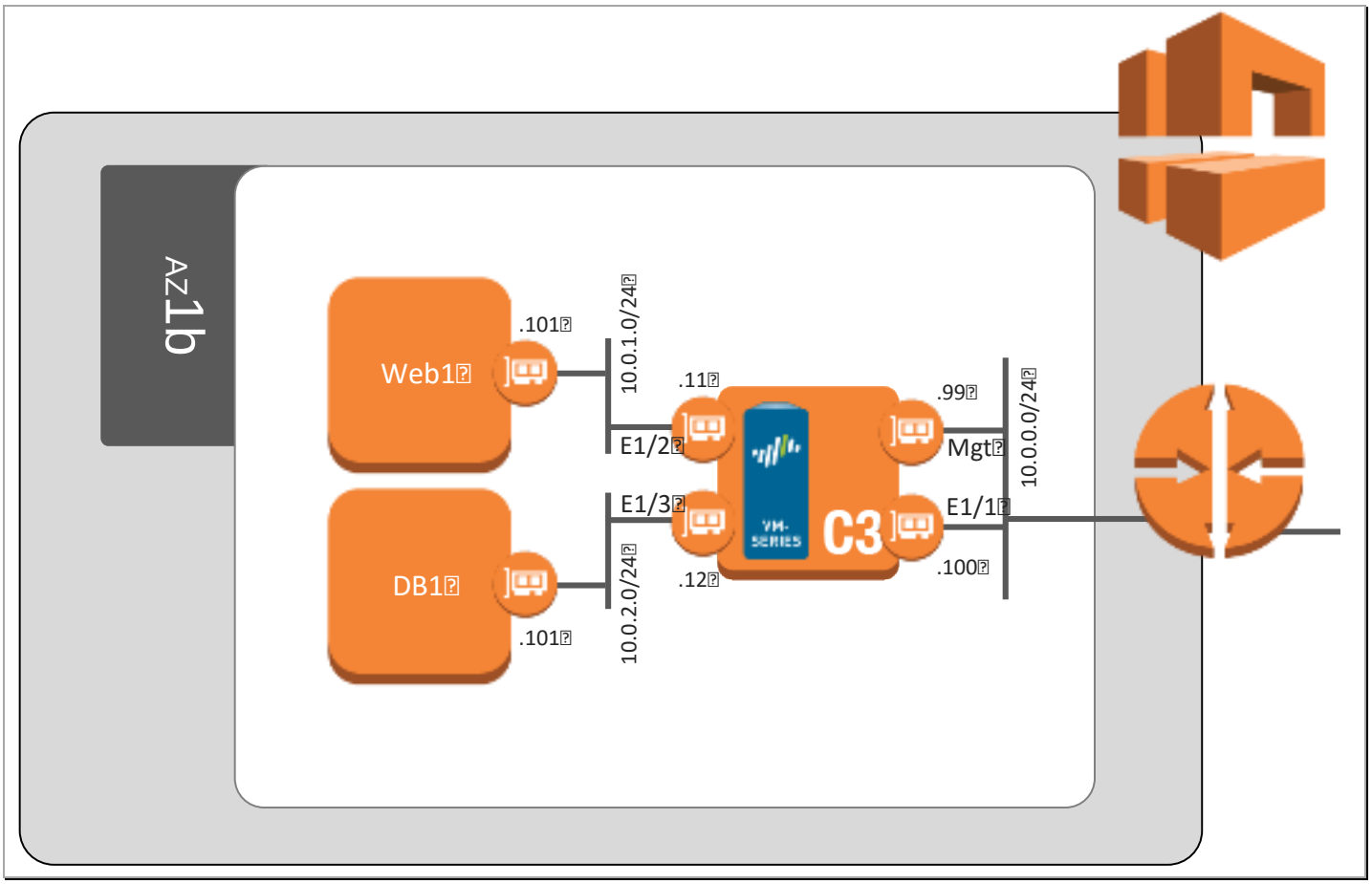


Route tables:



And **Elastic IPs (EIPs)**:

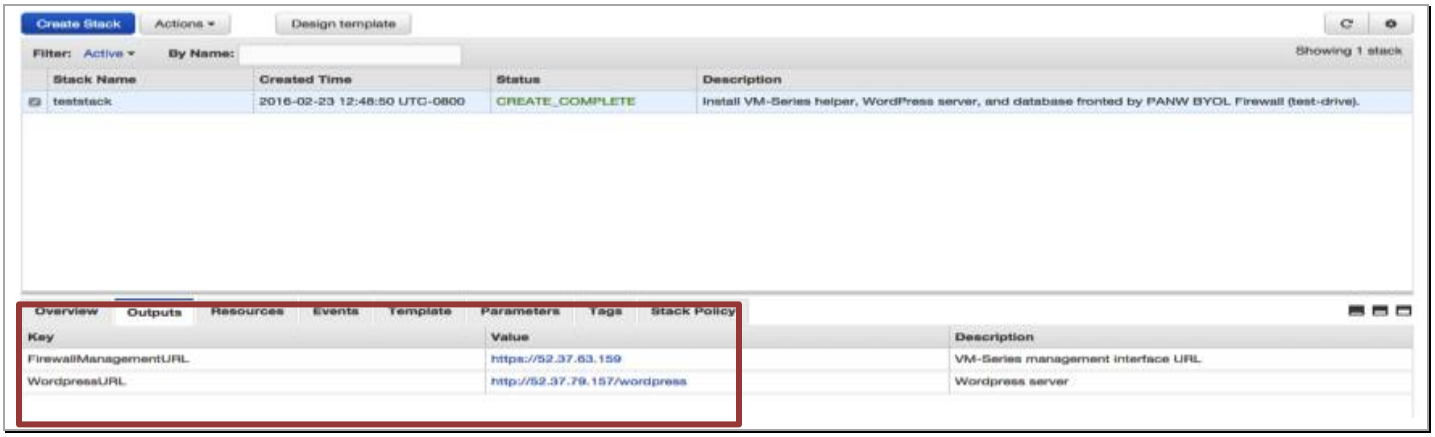
All of this matches the topology shown previously:



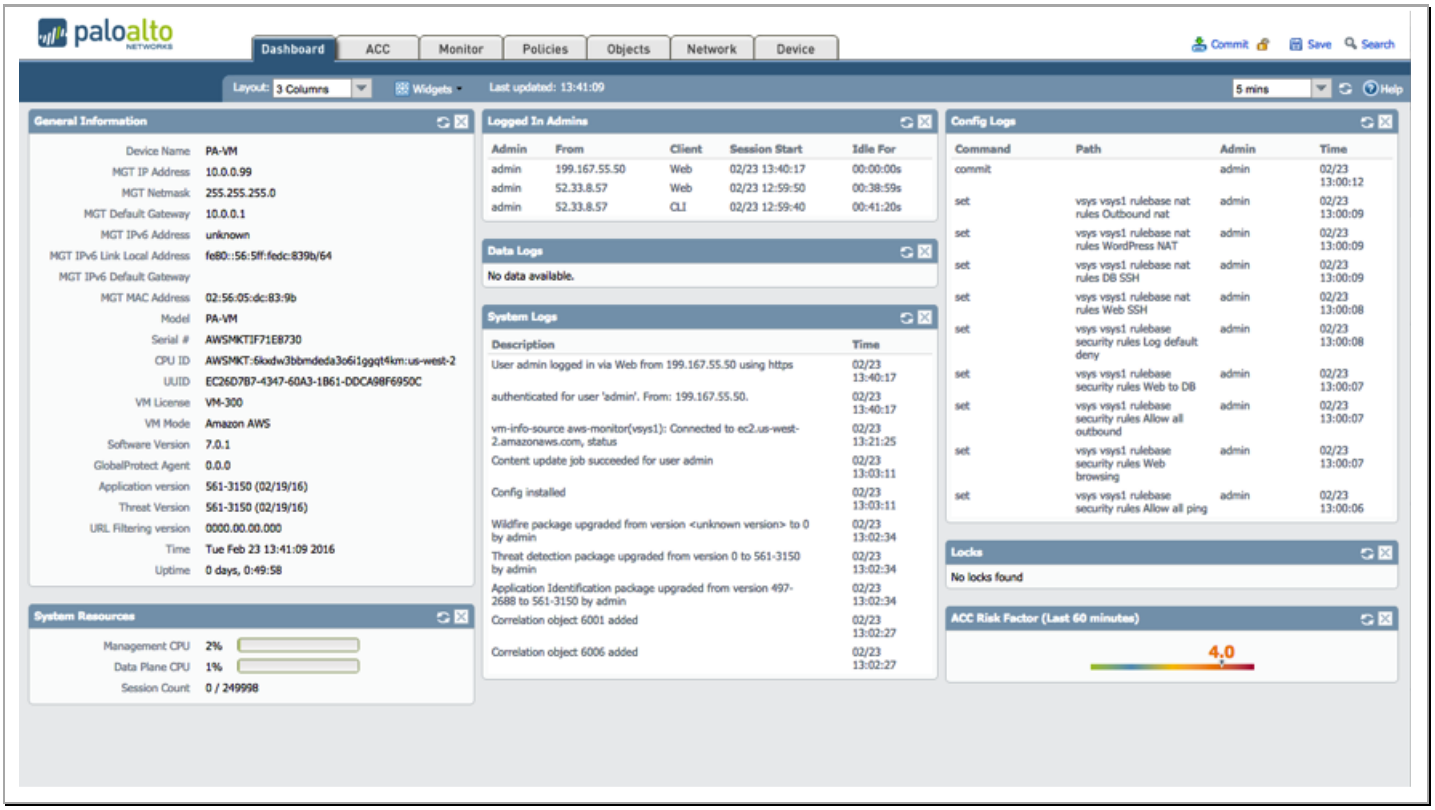
7. Access the firewall

NOTE: Bootstrapping a VM-Series firewall takes approximately 9 minutes. So once the stack has been created successfully, it may be a while before the firewall is up and you are able to log into the firewall.

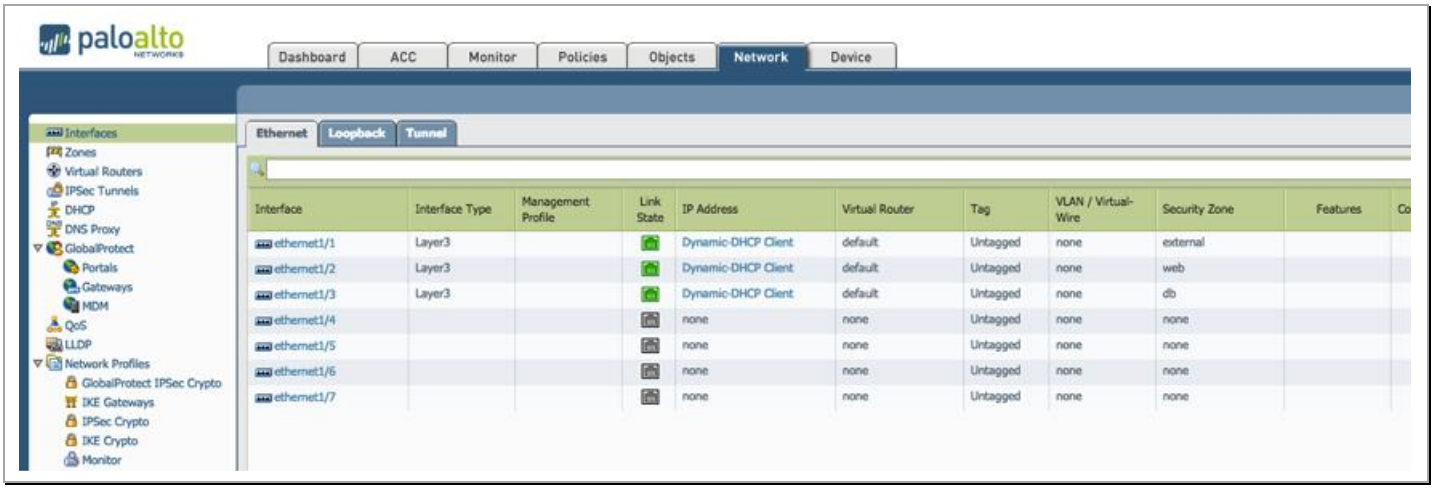
Once stack creation is complete, you should see two lines under the **Outputs** tab:



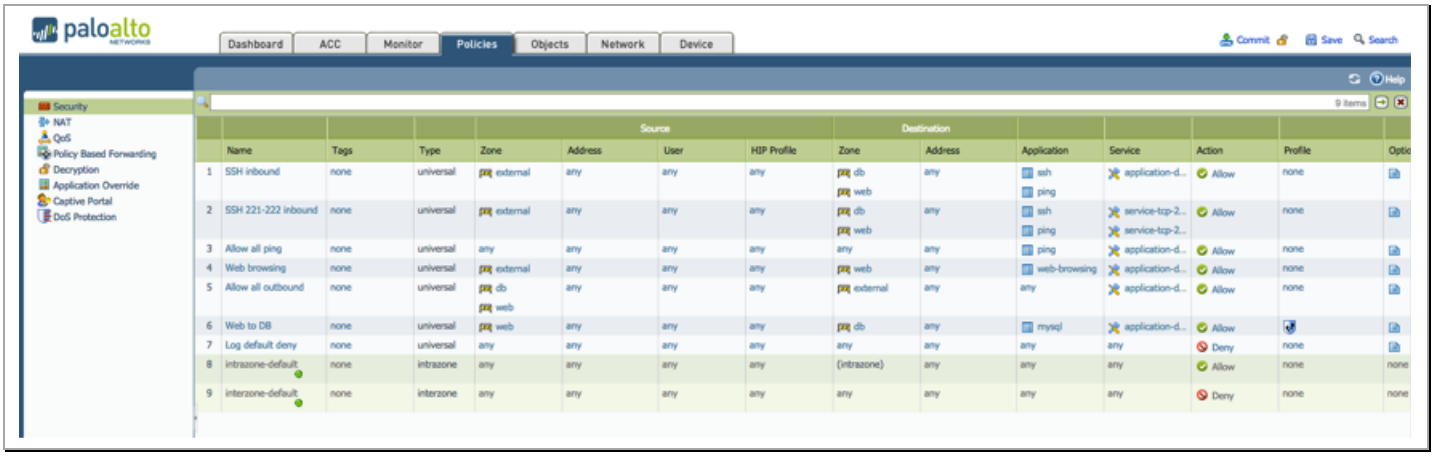
You should now be able to login to the firewall using the **username: admin** and password: **paloalto**



Here are the interfaces to zone mappings:

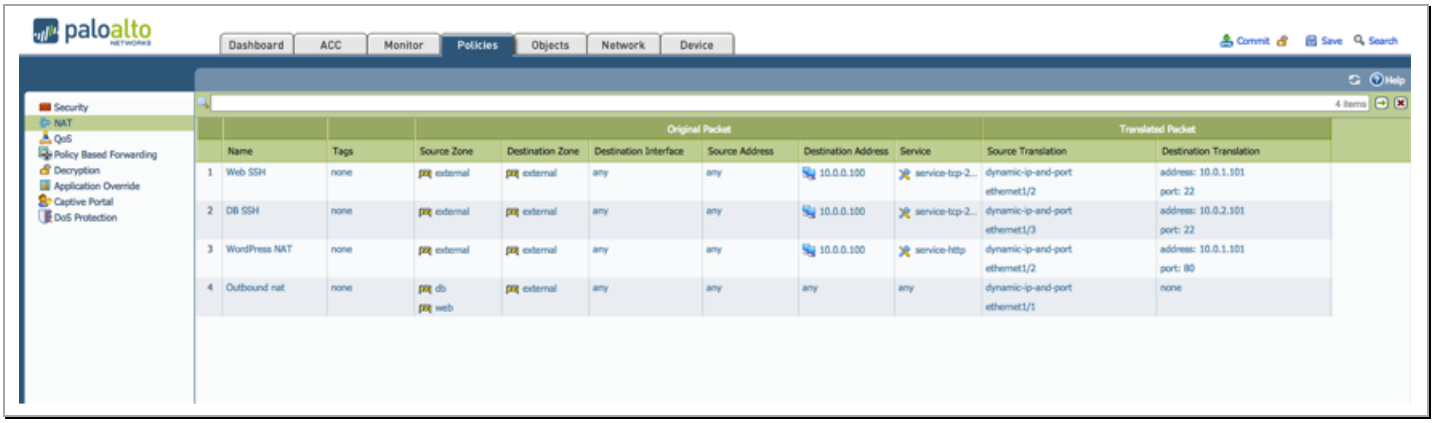


In the policies tab you can review the security policies:



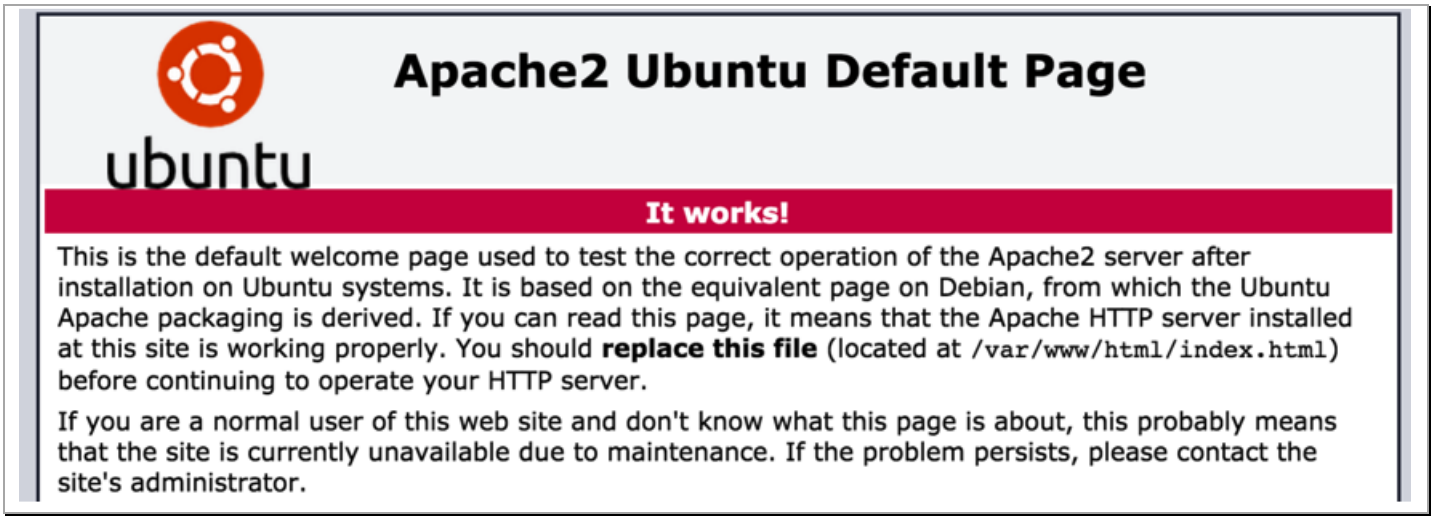
These policies are defined to allow ssh access on ports 221 and 222 to the web and db server respectively (for troubleshooting purposes), secures N/S traffic and E/W traffic between zones.

And the NAT policies allow for ssh access to the web and db servers as well as directing web traffic to the web server only. There is also a rule for source NAT from web and db servers to the outside world.

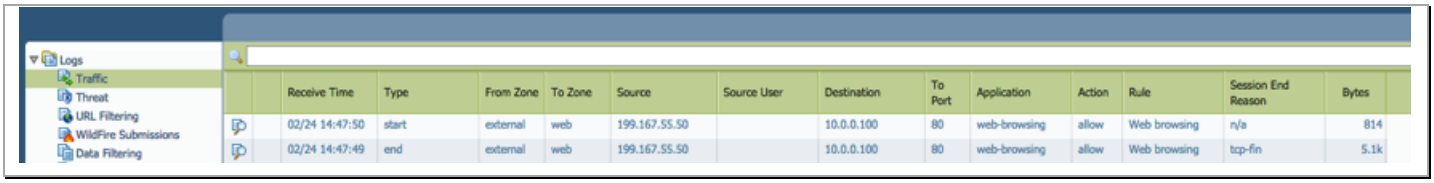


8. Access the Webserver

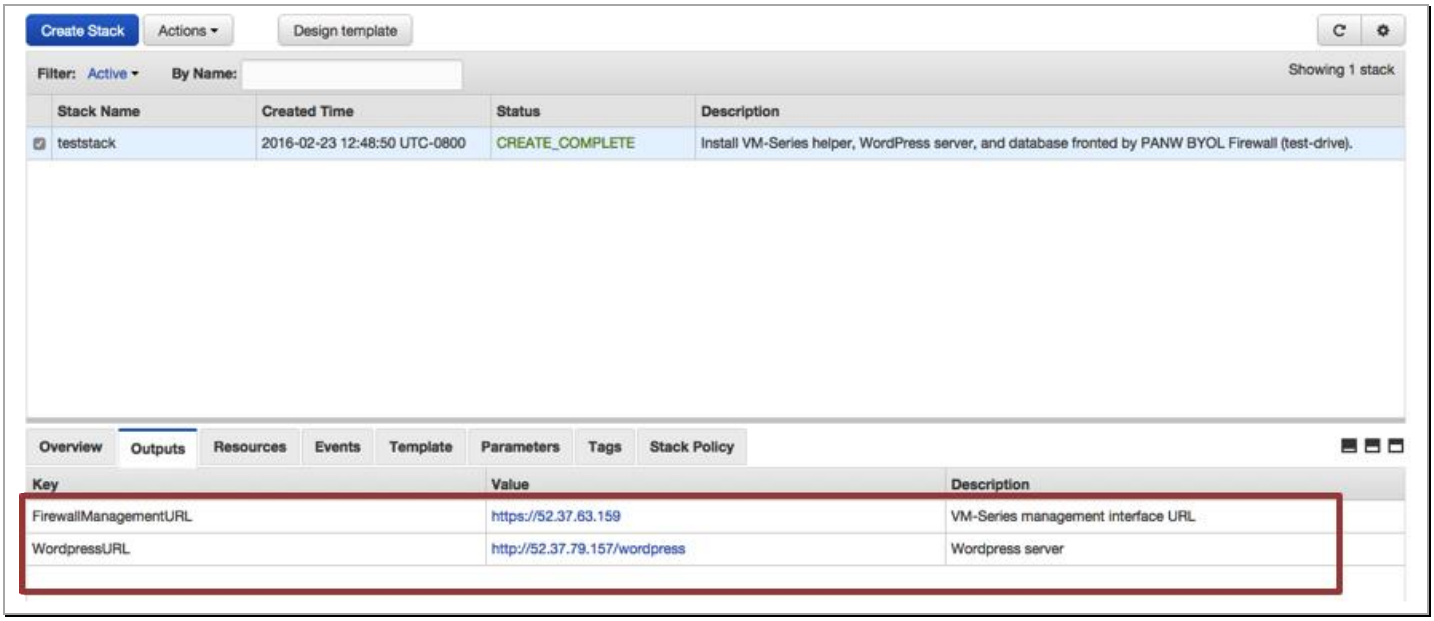
Using the second URL in the output tab access the static content of the webserver so in a web browser just type http://<webserver-IP>/ and you should see:



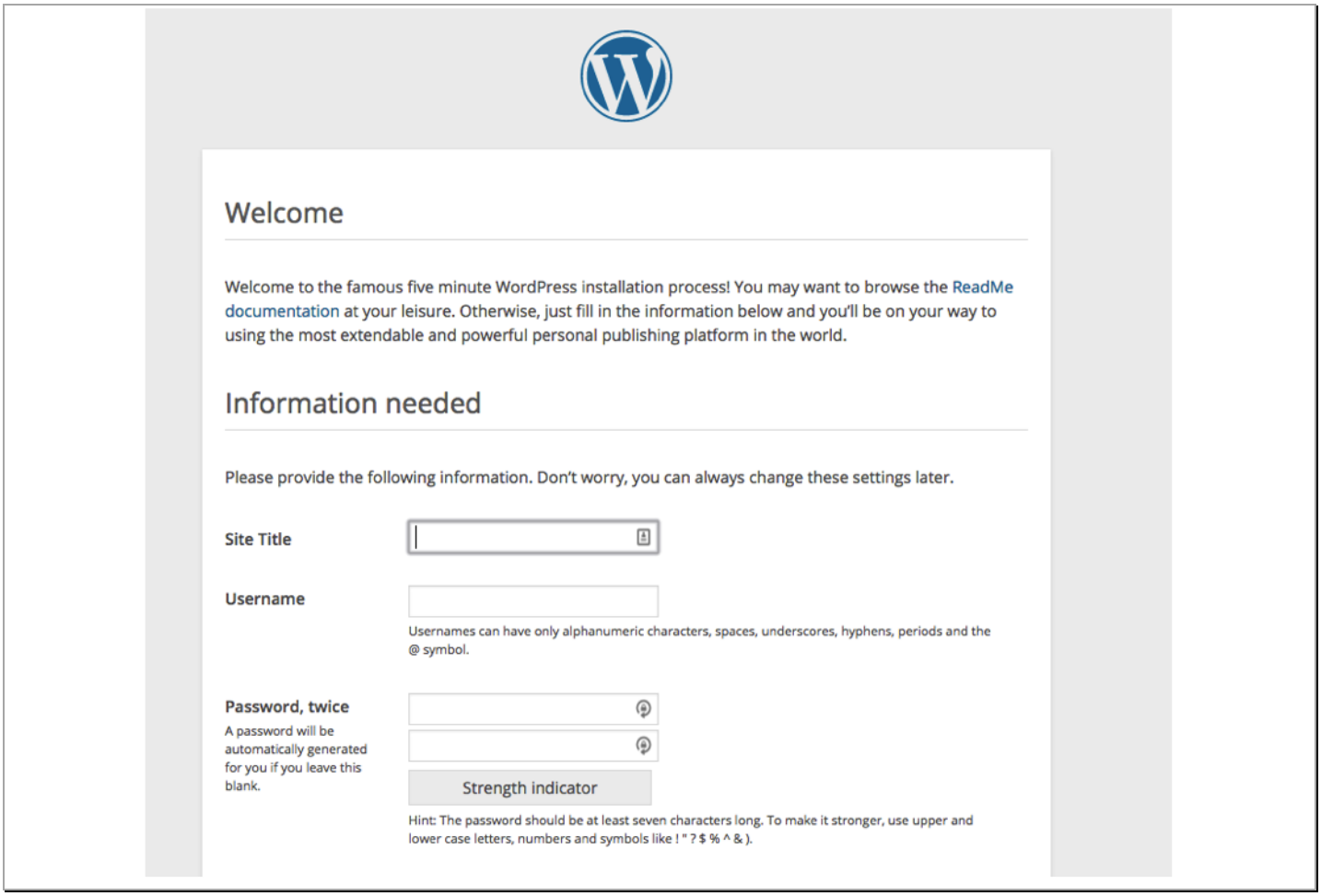
Check firewall logs to verify that the traffic is passing through the firewall:



Now let us verify we pass east-West traffic through the firewall. In the browser, head to the wordpress server (http://<webserver-IP/wordpress>), this should be the second link in the AWS console **Outputs** tab:

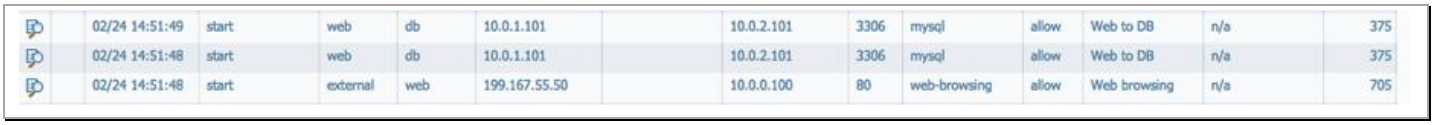


And you should see the WordPress welcome screen:



Note: You don't need to actually configure the new WordPress server for the purpose of the test drive. In its initial, un-configured state, it will generate the traffic we need to test the VM-Series firewall.

Now, head back to the firewall and verify that the traffic did indeed go through the firewall from web to db :



02/24 14:51:49	start	web	db	10.0.1.101		10.0.2.101	3306	mysql	allow	Web to DB	n/a	375
02/24 14:51:48	start	web	db	10.0.1.101		10.0.2.101	3306	mysql	allow	Web to DB	n/a	375
02/24 14:51:48	start	external	web	199.167.55.50		10.0.0.100	80	web-browsing	allow	Web browsing	n/a	705

You have now successfully deployed a cloud formation template with a VM-Series firewall in AWS.

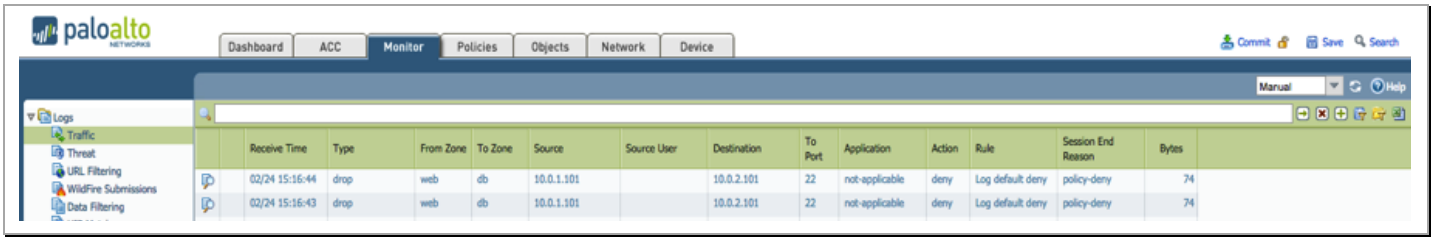
9. Launch some attacks

9.1 SSH from Web Server to DB Server

Let's simulate a compromised web server that is being used to attack the database. This is a common attack strategy of getting a foothold on the web front-end server and then expanding to the other application tiers with the ultimate goal of accessing all data in the database.

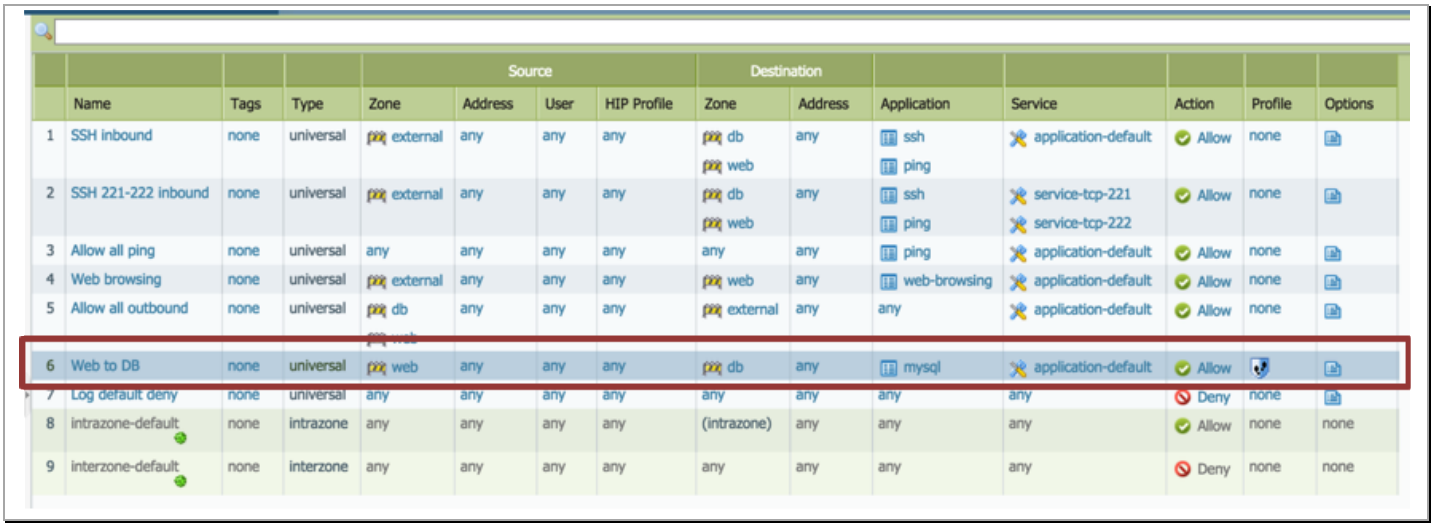
Go to http://<webserver-IP>/sql-attack.html and simulate a web to db ssh attempt by clicking on the **LAUNCH WEB TO DB SSH ATTEMPT.**

This launches a CGI script that attempts to ssh as root to the db server from the web server. Now return to the firewall's monitor tab to note the failed traffic:



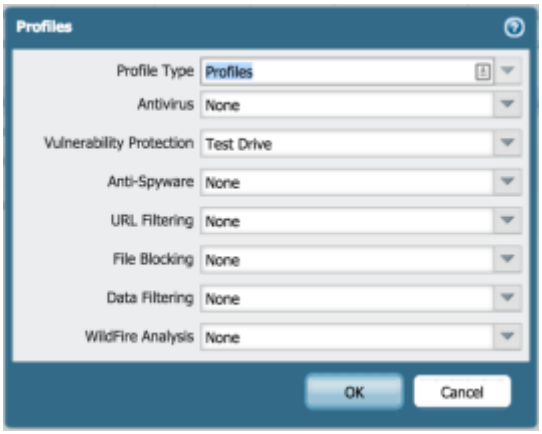
9.2 SQL Brute force attack

On the firewall's security policies tab, under Security, Rule 6, you will notice that the web to db traffic is protected further by a vulnerability profile:



				Source				Destination						
	Name	Tags	Type	Zone	Address	User	HIP Profile	Zone	Address	Application	Service	Action	Profile	Options
1	SSH inbound	none	universal	external	any	any	any	db web	any	ssh ping	application-default	Allow	none	
2	SSH 221-222 inbound	none	universal	external	any	any	any	db web	any	ssh ping	service-tcp-221 service-tcp-222	Allow	none	
3	Allow all ping	none	universal	any	any	any	any	any	any	ping	application-default	Allow	none	
4	Web browsing	none	universal	external	any	any	any	web	any	web-browsing	application-default	Allow	none	
5	Allow all outbound	none	universal	db web	any	any	any	external	any	any	application-default	Allow	none	
6	Web to DB	none	universal	web	any	any	any	db	any	mysql	application-default	Allow		
7	Log default deny	none	universal	any	any	any	any	any	any	any	any	Deny	none	
8	intrazone-default	none	intrazone	any	any	any	any	(intrazone)	any	any	any	Allow	none	none
9	interzone-default	none	interzone	any	any	any	any	any	any	any	any	Deny	none	none

Now click on the icon in the Profile column and you will see all the threat protection profiles



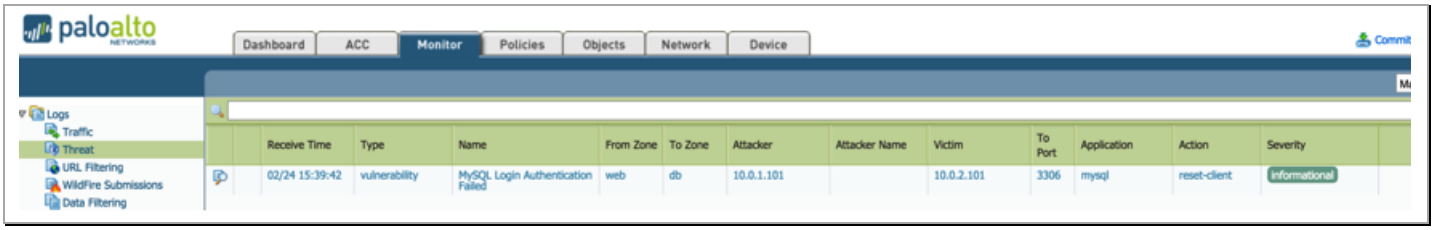
Note the Vulnerability Protection profile. This is a custom profile created just for this lab. It is part of the default vulnerability protection profile but is called out separately for the purpose of this demo environment.

Let's finally trigger the attack. Head back to the sql-attack.html page at https://<webserver-IP>/sql-attack.html

Click on Launch Brute Force Attack to start a script that will generate multiple failed MySQL authentication attempts.

LAUNCH BRUTE FORCE SQL ROOT PASSWORD GUESSING

This will launch some scripted attacks on the SQL server and use the pre-configured threat protection to show and block those attacks on the VM-Series firewall. Now return to the firewall and click the Monitor tab and then click on Threats in the left hand pane under Logs and notice the new vulnerability log message regarding the failed MySQL events:

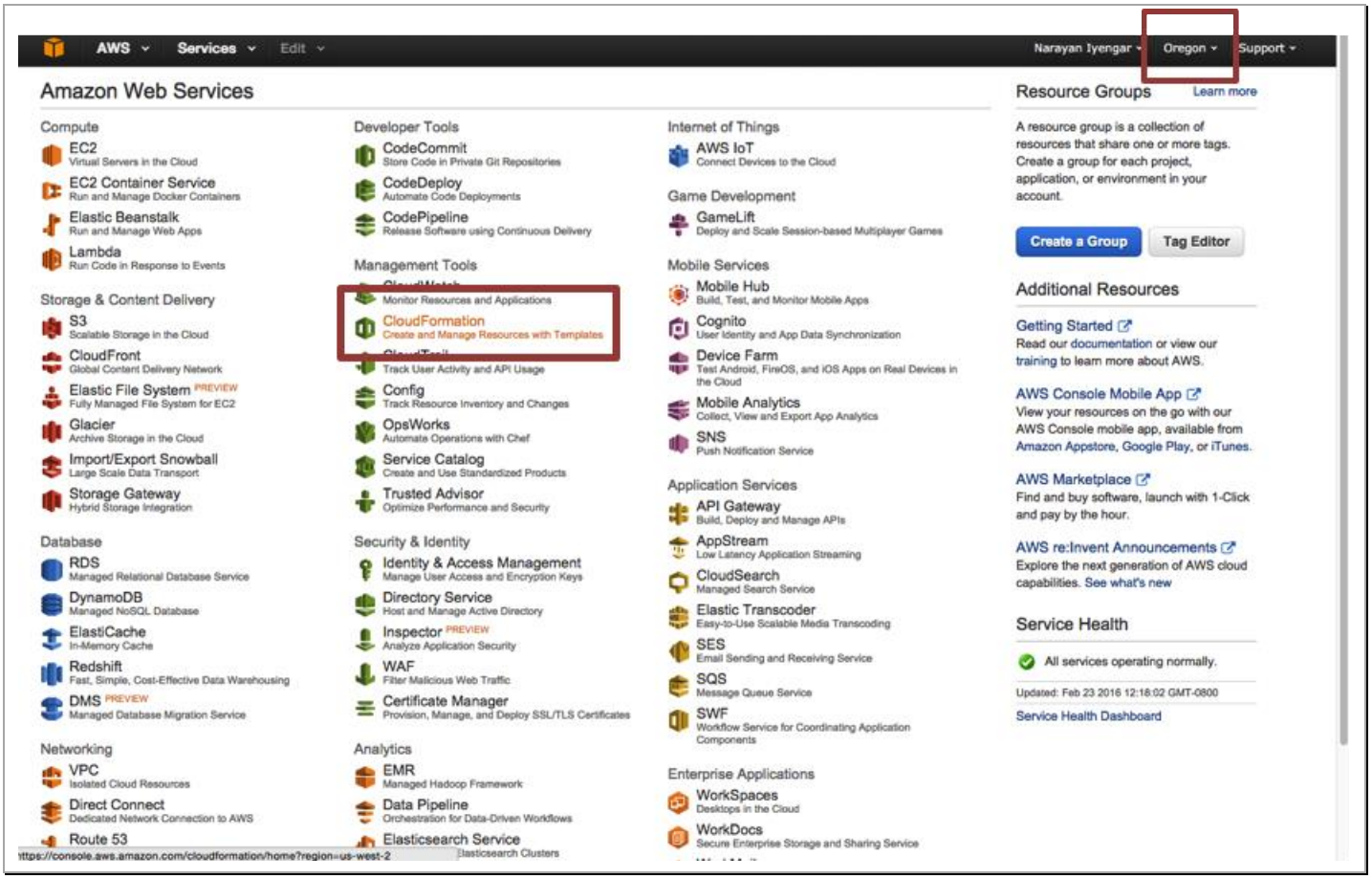


The CGI script you launched above attempted to login to the MySQL database multiple times with an incorrect password. The VM-Series firewall saw this activity and using the vulnerability profile, reset the connection and logged the activity.

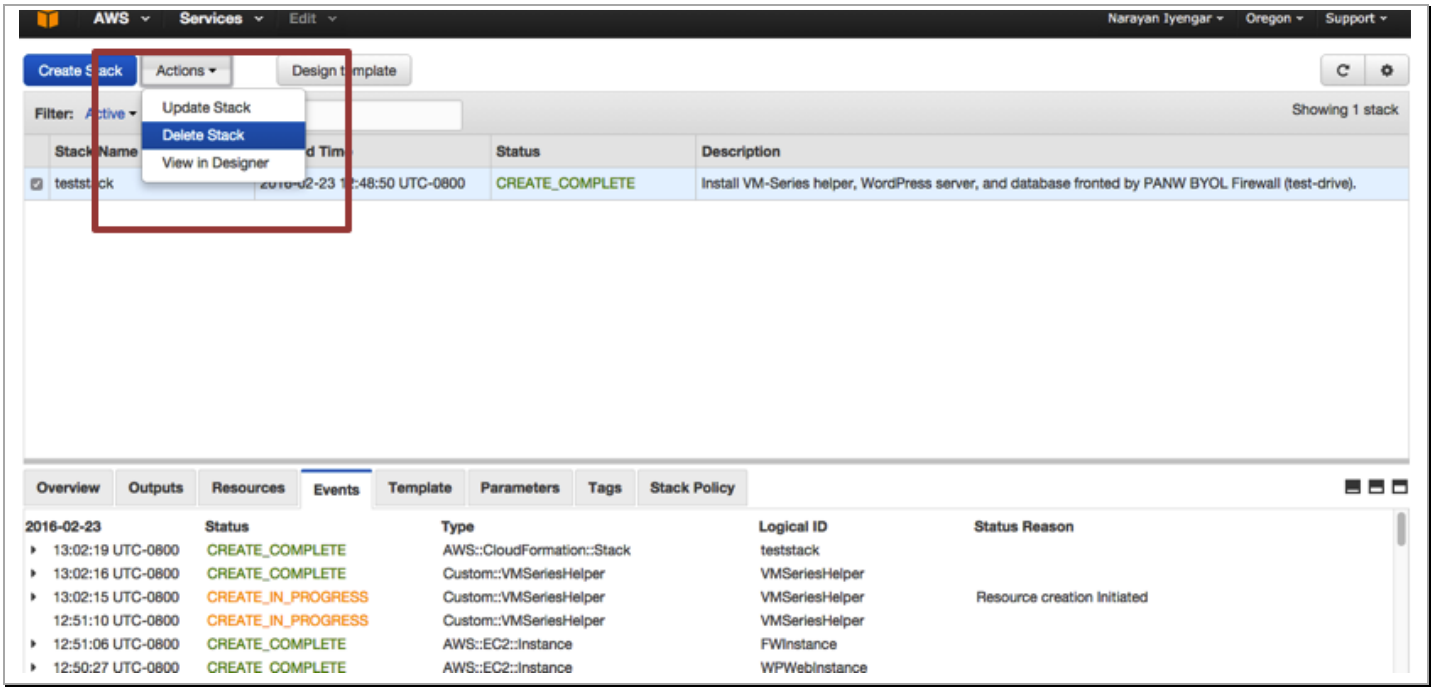
10. Cleanup

10.1 Delete the Stack

Once done with the template, feel free to play around with various thins. If done, cleanup as follows. In the AWS management console, click on **CloudFormation**:



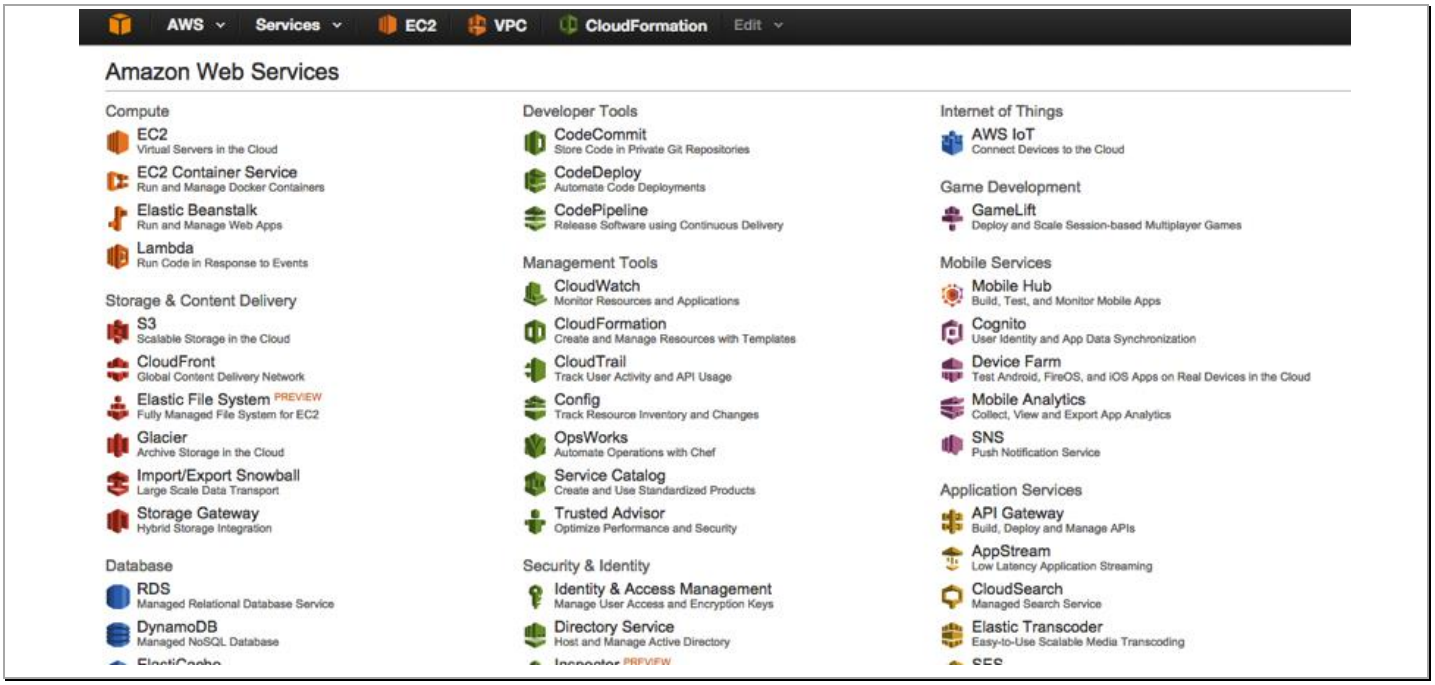
Under **Actions**, click **Delete Stack:**



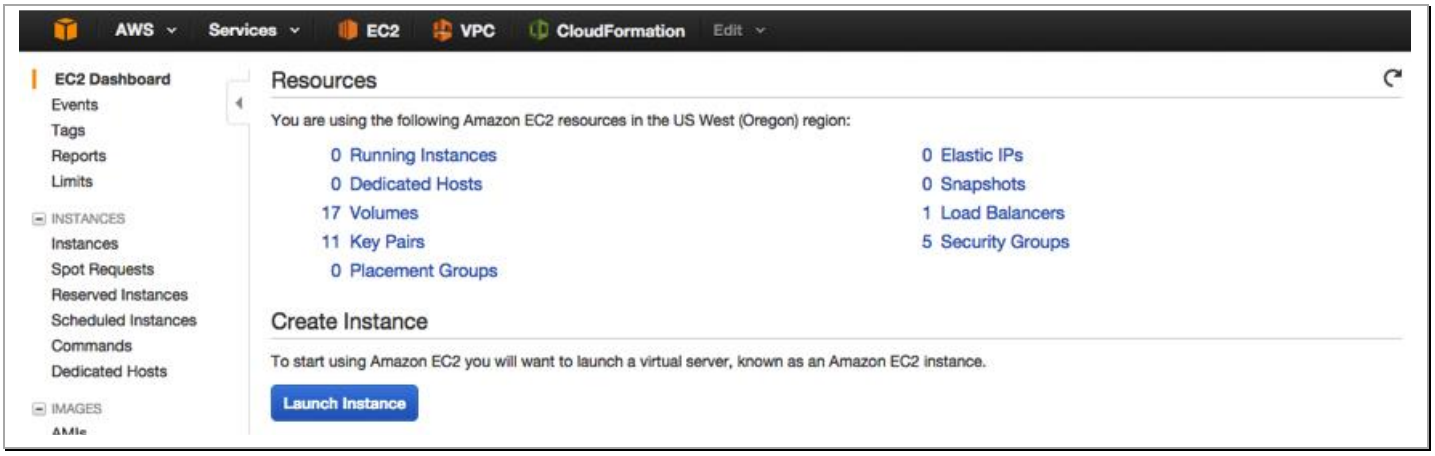
This should delete all the resources created via the template and release any Elastic IPs associated with the firewall.

10.2 Delete keys

As part of the template certain keys are created to access the VM-Series firewall. These keys need to be manually deleted. To do that, go to the **EC2** console:

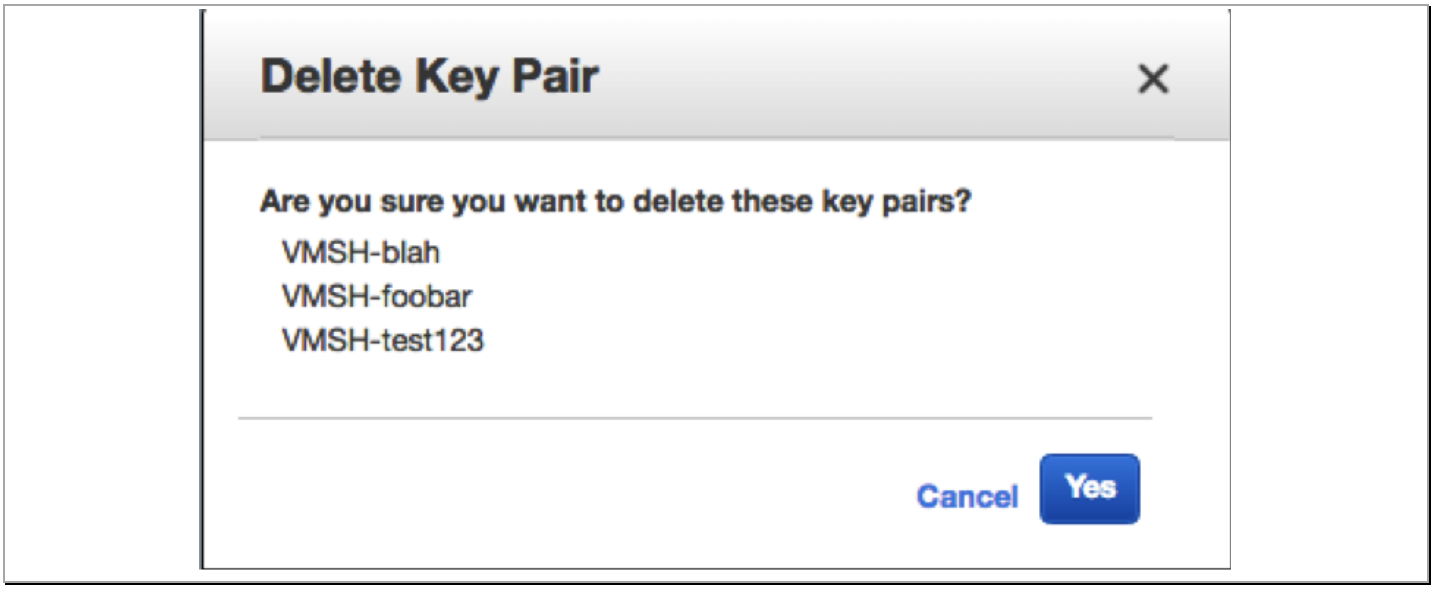


Click on **Key Pairs**:



Select all keys that start with **VMSH** and click **Delete:**

And confirm **Yes** on the next screen:



11. Conclusion

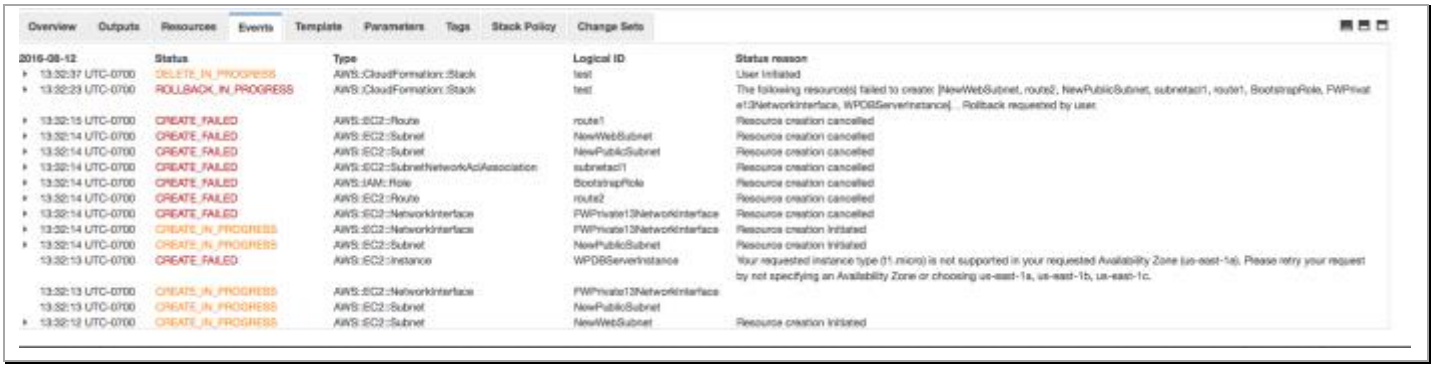
You have successfully deployed a sample CFT in AWS and demonstrated how the next generation VM-Series firewall can not only secure traffic inbound into your VPC, but within the VPC itself.

Appendix A

Troubleshooting tips

1. Stack creation fails

Occasionally stack creation fails due to various unknown reasons. Maybe AWS is updating their software, maybe that particular region is having a service outage. These errors are usually transient in nature and generally will go away when the stack is deleted and re-launched (OR launched in a different region) If the errors are consistent, then please read on for other troubleshooting tips. For instance, one of the errors encountered maybe as follows:



The error indicates that no t1.micro instances are available in the selected availability zone. This is a transient error and the fix is to redeploy the template.

2. EIP Exhaustion

If the account does not have a minimum two unallocated and unassociated elastic IPs, stack creation will fail.

If you encounter this error, please refer to Section 3.6 for more details.

3. Bootstrapping not working

If the VM-Series firewall is up and you are able to access the login page, but unable to login using the username/password: admin/paloalto, then chances are boostrapping has failed. There could be several reasons:

a. Corrupt configuration files

Please ensure that the bootstrap.xml and init-cft.txt files mentioned in Section 3.5 are not corrupted.

b. Incorrect bootstrap bucket-name

Another reason for bootstrapping to fail is that the bootstrap bucket name (Parameter: BootstrapBucketName) was mentioned incorrectly during stack creation (template launch). Please make sure the bucket name created in Section 3.5 is mentioned when launching the template.