Auto Scaling the VM-Series on AWS v2.1

Deployment Guide

http://www.paloaltonetworks.com

Last Update: January 2019

Version number	Comments
2.1-CS	• Panorama is required for this deployment. • Panorama in HA is NOT supported. • Firewall template can be deployed to existing VPC. • Can launch templates with either network load balancer or application load balancer. • For different VPC deployments, traffic no longer traverses the public internet.

Contents

1. About

Auto Scaling the VM-Series on AWS uses AWS CloudFormation Templates and scripts to automate the deployment of an Auto Scaling architecture protected by the VM-Series. The templates leverage several AWS services including Lambda, auto scaling groups, Elastic Load Balancing (ELB), S3, SNS, and CloudWatch, along with VM-Series automation capabilities such as PAN-OS API and bootstrapping. Once deployed, the VM-Series will be able to scale out and scale in automatically as application workload resource demands increase.

The AWS CloudFormation Templates and scripts can be found on the Palo Alto Networks GitHub repository:

https://github.com/PaloAltoNetworks/aws-elb-autoscaling/tree/master/Version-2.1-Community-Supported

Enhancements included in this release include a more defined separation of the VM-Series firewall VPC from the application VPC's. This separation allows security teams to deliver security services to the internal business groups, application developers and DevOps teams who build, ship and maintain applications.

A separate firewall VPC enables security to deliver separate billing and management for each template (application VPC). The clearer separation also allows security and application teams to implement specific AWS policy configurations such as restrictive security groups, no IGW (Internet Gateway) etc. on the application VPC's for a stronger security posture while leaving all security of inbound traffic to the security team. [Note that the enforcement of these policy-based capabilities on the application VPC's can be facilitated through services such as RedLock by Palo Alto Networks, while the VM-Series provides the inline, network security protections and visibility. As the number of protected applications VPC's grow, security teams can use the auto scaling stack of firewalls for elastic, on-demand, security. Each application (via its related internal load balancer) is mapped to a load balancing rule in the external load balancer.

This architecture uses a load balancer sandwich for protecting Internet facing applications, for other use cases, see https://github.com/PaloAltoNetworks/aws-elb-autoscaling

Important Notes:

1. This deployment requires **Panorama** (physical or virtual) be deployed SEPERATELY, PRIOR to getting started. Panorama in HA is NOT supported.
2. Auto Scaling the VM-Series on AWS has NOT been tested in GovCloud.

3. Cross account deployments are not supported in v2.1-CS but will be supported in a future template. Please leave this field blank in the template.

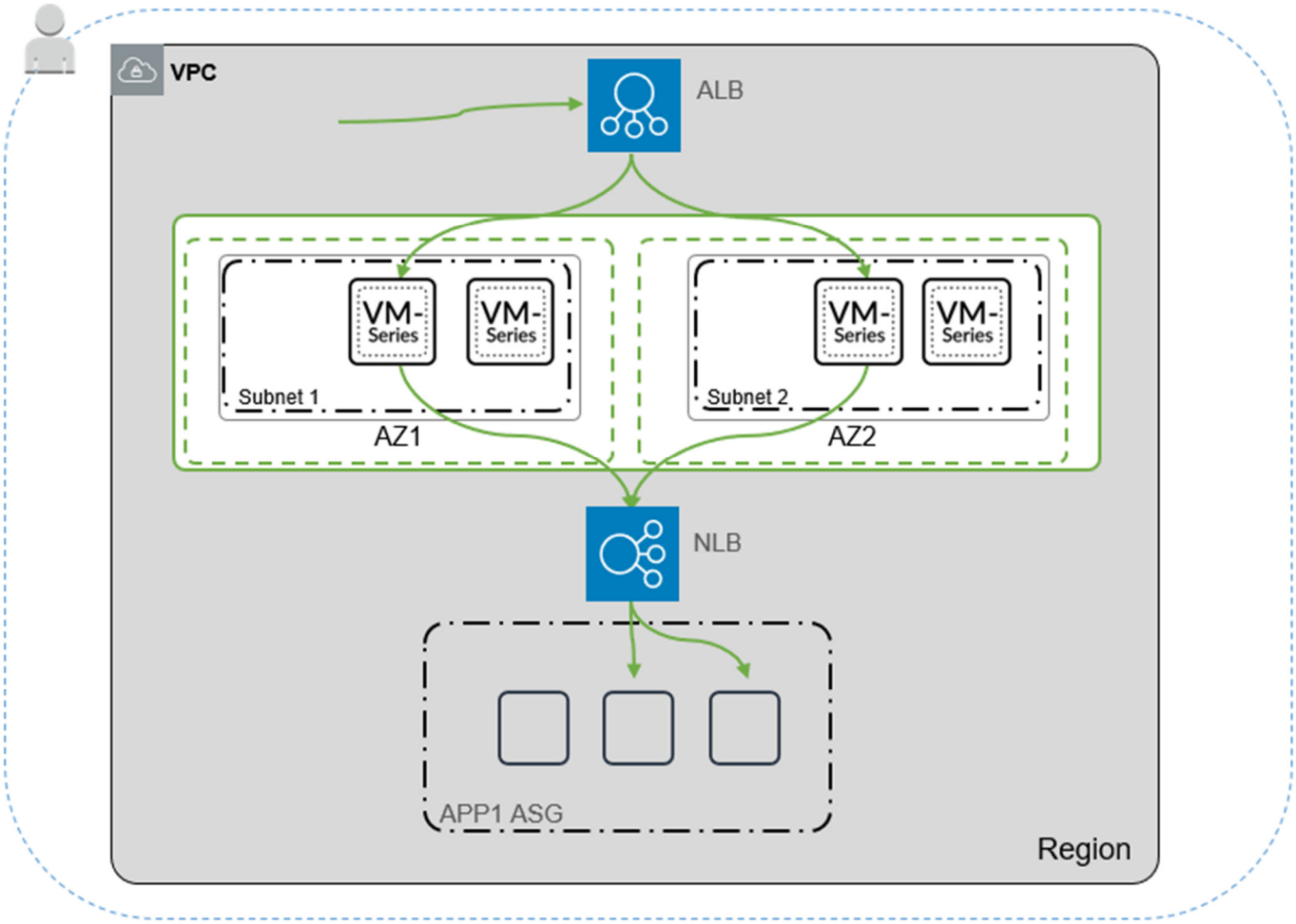
2. Support Policy

Community Supported**:** This template and deployment guide are released under an as-is, best effort, support policy. These scripts should be 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.

3. Topology

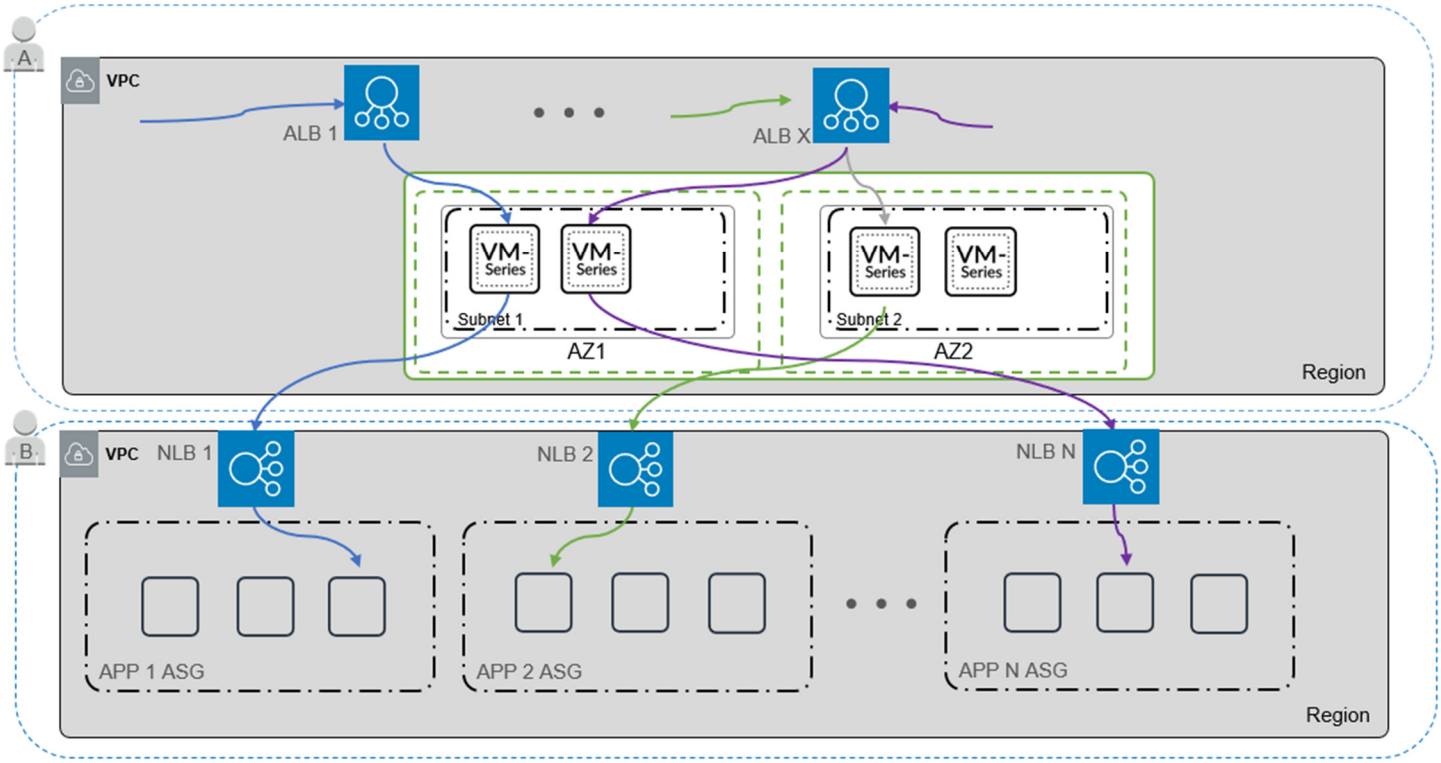
To help you manage increased application scale, version this template uses a hub and spoke architecture that simplifies deployment. The hub and spoke architecture are built using two templates that support single and multi-VPC deployment both within a single AWS account and across AWS accounts.

- **Firewall Template**—The firewall template deploys an application load balancer and VM-Series firewalls within auto scaling groups across two Availability Zones (AZs). This internet-facing application load balancer distributes traffic that enters the VPC across the pool of VM-Series firewalls. The VM-Series firewalls automatically publish custom PAN-OS metrics that enable auto scaling.



- **Application Template**—The application template deploys your preference of Network or Application Load Balancer and one auto scaling group with a web server in each AZ.

Together these templates allow you to deploy a load balancer sandwich topology with an internet-facing application load balancer and an internal network load balancer. The application load balancer is accessible from the internet and distributes traffic that enters the VPC across a pool of VM-Series firewalls. The firewalls then route traffic using NAT policy to the internal network load balancer(s), which distributes traffic to an auto scaling tier of web or application servers. The VM-Series firewalls are enabled to publish custom PAN-OS metrics to AWS CloudWatch where you can monitor the health and resource load on the VM-Series firewalls and then use that information to trigger a scale in or scale out event in the respective auto scaling group of firewalls.



Components used in this Template

Firewall Template

The firewall template deploys a new VPC with two Availability Zones (AZs), subnets, route tables, and security groups required for routing traffic across these AZs, and an AWS NAT gateway. It also deploys an external network or application load balancer, and an Auto Scaling Group (ASG) with a VM-Series firewall. You can deploy this template as a new VPC or into firewall template created for an existing VPC. Although Panorama is a prerequisite for this deployment, the template does not deploy Panorama

Application Template

The application template deploys a network or application load balancer and an ASG with a web server. Because the network load balancer has a unique IP address per AZ, and the NAT policy rule on the firewalls must reference a single IP address, there is one ASG for each of the two AZs. All the firewalls in an ASG have identical configuration.

This version of the auto scaling solution includes 5 application templates variations:

1. The application ALB template allows you to deploy the Application Load Balancer version of the application template within a previously existing VPC or a new VPC using the same AWS Account.
2. The application NLB template allows you to deploy the Network Load Balancer version of the application template within a previously existing VPC or a new VPC using the same AWS Account.
3. There is also a template that allows you to deploy the application into the same VPC as the VM-series.

Lambda Functions

AWS Lambda provides robust, event-driven automation without the need for complex orchestration software. In the firewall template, AWS Lambda monitors a Simple Queue Service (SQS) to learn about network load balancers that publish to the queue. When the Lambda function detects a new network load balancer, it creates a new NAT policy rule and applies it to the VM-Series firewalls within the ASG. The firewalls have a NAT policy rule for each application, and the firewalls use the NAT policy rule (that maps the port to network load balancer IP address) to forward traffic to the network load balancer in front of the application web servers. Upon deletion of the templates, the Lambda functions also deletes all the configuration items that Lambda added to the device group and template stack in Panorama. This includes the NAT Rule, Address Object, and Static Routes that were pushed to the VM-Series. The Lambda function will handle delicensing as well.

Panorama

The Panorama management server provides centralized monitoring and management of multiple Palo Alto Networks next-generation firewalls from a single location. Panorama allows you to oversee all applications, users, and content traversing your network, and then use this knowledge to create application enablement policies that protect and control the network.

Unlike v2.0 a previously deployed Panorama is required for Auto Scaling v2.1-CS. Due to the use of Panorama a bootstrap.xml config file is no longer needed in the S3 Bootstrap bucket. A Sample configuration is included in the GitHub repo so that you can copy the configuration from the template stack and device group when you create them in your existing panorama. **The untrust and trust zones created in Panorama all must be lower case**. In Panorama you must configure your

- Network interfaces using dhcp. Only eth1/1 should automatically create default route
- trust and untrust zones. All zones must be lower case
- Security Policy. Zones in security policy will be untrust and trust.
- Administrator account for admin account named pandemo
- Virtual router with a naming convention VR-<TemplateStackName>

If you are not familiar with Panorama please visit the TechDocs link below for more information
https://docs.paloaltonetworks.com/panorama/9-0/panorama-admin.html

Lambda will populate the following in Panorama once the application template has launched

- NAT policy
- Address object for LB in Application Template
- Static routes in the virtual router.
- Tcp81 service object

License Deactivation key

- The template requires a License Deactivation API Key and the "Verify Update Server Identity" to be enabled to deactivate the license keys from Panorama. The License Deactivation Key should be obtained from Palo Alto Customer Support Portal. Steps on how to activate this can be found below.
  https://www.paloaltonetworks.com/documentation/81/virtualization/virtualization/license-the-vm-series-firewall/deactivate-the-licenses/install-a-license-deactivation-api-key

Panorama VM-Auth-Key

- For the bootstrapped firewalls to connect to Panorama to receive their bootstrap configuration, we need a vm-auth-key. The following link will walk you through how to generate this vm-auth key.

  https://docs.paloaltonetworks.com/vm-series/8-1/vm-series-deployment/bootstrap-the-vm-series-firewall/generate-the-vm-auth-key-on-panorama

Panorama API Key

- To authenticate the API, we need a Panorama API Key. The following link will walk you through generating an API Key. Lambda needs this to auto configure template and device group options.
  https://docs.paloaltonetworks.com/pan-os/8-1/pan-os-panorama-api/get-started-with-the-pan-os-xml-api/get-your-api-key

Bootstrap files

This template requires a init-cfg.txt file so that the VM-Series firewall has the basic configurations needed to

- Perform interface swap so the VM-Series untrust traffic uses AWS ENI for eth0
- Communicate to Panorama for device group and template configuration

A sample init-cfg.txt file is provided in the Auto Scaling GitHub repo and has the basic configuration to get started. This auto-scaling solution requires the swapping of the dataplane and management interfaces to enable the load balancer to forward web traffic to the auto-scaling tier of VM-Series firewalls. For details on management interface mapping for use with amazon ELB see the following link

https://docs.paloaltonetworks.com/vm-series/8-1/vm-series-deployment/set-up-the-vm-series-firewall-on-aws/about-the-vm-series-firewall-on-aws/management-interface-mapping-for-use-with-amazon-elb.html#

Launch the Firewall Template

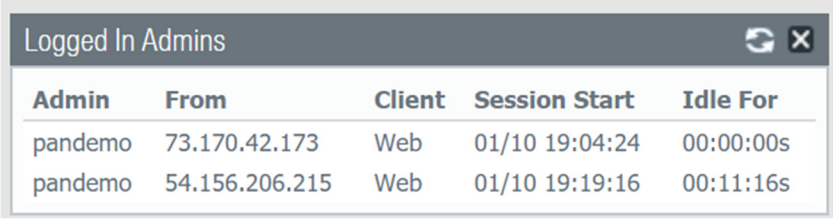
You can choose to deploy the firewall and application templates the same VPC or in a separate VPCs.

This version the template does not support cross account deployments. It does however support a hub and spoke architecture whereby you can deploy the firewall template in one AWS account and use it as a hub to secure your applications (spokes) that belong to the same or to different AWS accounts.

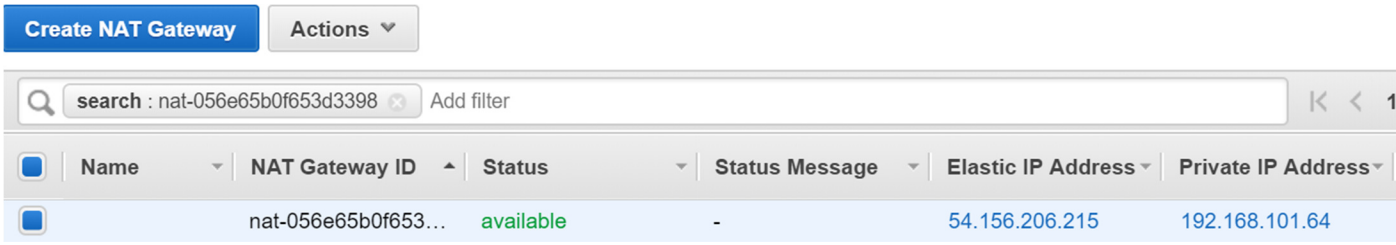
This workflow tells you how to deploy the external load balancer and the VM-Series firewalls using the firewall template. The vm-auth-key must be configured on Panorama prior to launching this template.

This firewall template includes an AWS NAT gateway that the firewalls use to initiate outbound requests for retrieving updates, connecting to Panorama, and publishing metrics to AWS CloudWatch. The NAT Gateways also have Elastic IP addresses attached to them for each zone. Below are the recommend management interface security configurations in this use case.

- SSH SECURITY: Explicitly whitelist the IP address range you will connect to from your management interface. Upon initial deployment of the firewall Template, you will NOT be able to lock down the security group in AWS For HTTPS. You must leave HTTPS open or Lambda will not be able to connect to Panorama. Once you receive the following conformation of connection in Panorama it is now safe to lock down port 443.



- When you lock down port 443 you will lock down the IP range you connect from, as well as the EIP's Assigned to the NAT Gateways. You can find both NAT Gateway EIP's in AWS by navigating to VPC>NAT Gateways. Here you will see two NAT gateways and the EIP's associated with them. Note that EIP information for the security group for HTTPS.



- **PANORAMA PORT 3978 SECURITY:** This port must be able to receive traffic from any IP address.

STEP 1 | Review the checklist for deploying the Auto Scaling the VM-Series Template.

Make sure that you have completed the following tasks:

- (For PAYG only) Reviewed and accepted the EULA for the PAYG bundle you plan to use.
- (For BYOL only) Obtained the auth code. You need to enter this auth code in the /license folder of the bootstrap package.
- Downloaded the files required to launch the VM-Series Auto Scaling template from the GitHub repository

https://github.com/PaloAltoNetworks/aws-elb-autoscaling/tree/master/Version-2.1-Community-Supported

STEP 2 | Modify the init-cfg.txt file and upload to the /config folder.

- Because Panorama is used to bootstrap the VM-Series, your init-cfg.txt file should be modified as follows. No bootstrap.xml file is needed.

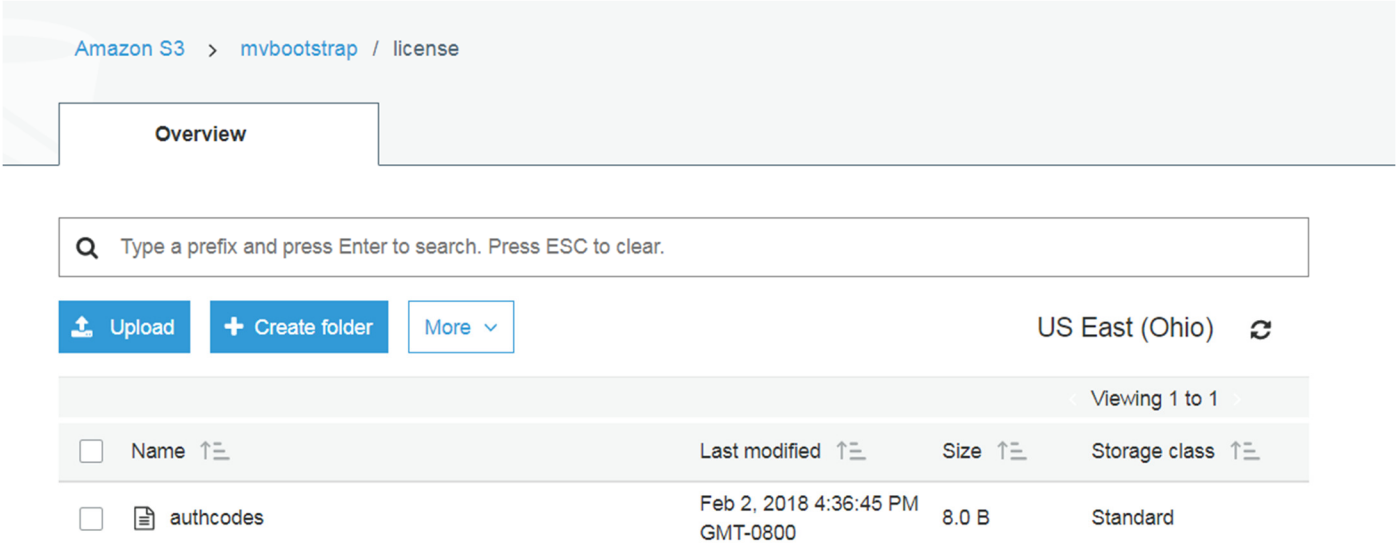
```
type=dhcp-client
ip-address=
default-gateway=
netmask=
ipv6-address=
ipv6-default-gateway=
hostname=
vm-auth-key=
panorama-server=
panorama-server-2=
tplname=AWS-tmplspoke1
dgname=AWS-dgspoke1
dns-primary=169.254.169.253
dns-secondary=8.8.8.8
op-command-modes=mgmt-interface-swap
dhcp-send-hostname=yes
dhcp-send-client-id=yes
dhcp-accept-server-hostname=yes
dhcp-accept-server-domain=yes
```

- Verify that the command for swapping the management interface (mgmt) and the dataplane interface (ethernet 1/1) on the VM-Series firewall on AWS in present. The command is listed above as op-command-modes=mgmt-interface-swap. Use the AWS DNS server of 169.254.169.253 for faster load balancer DNS name resolution.

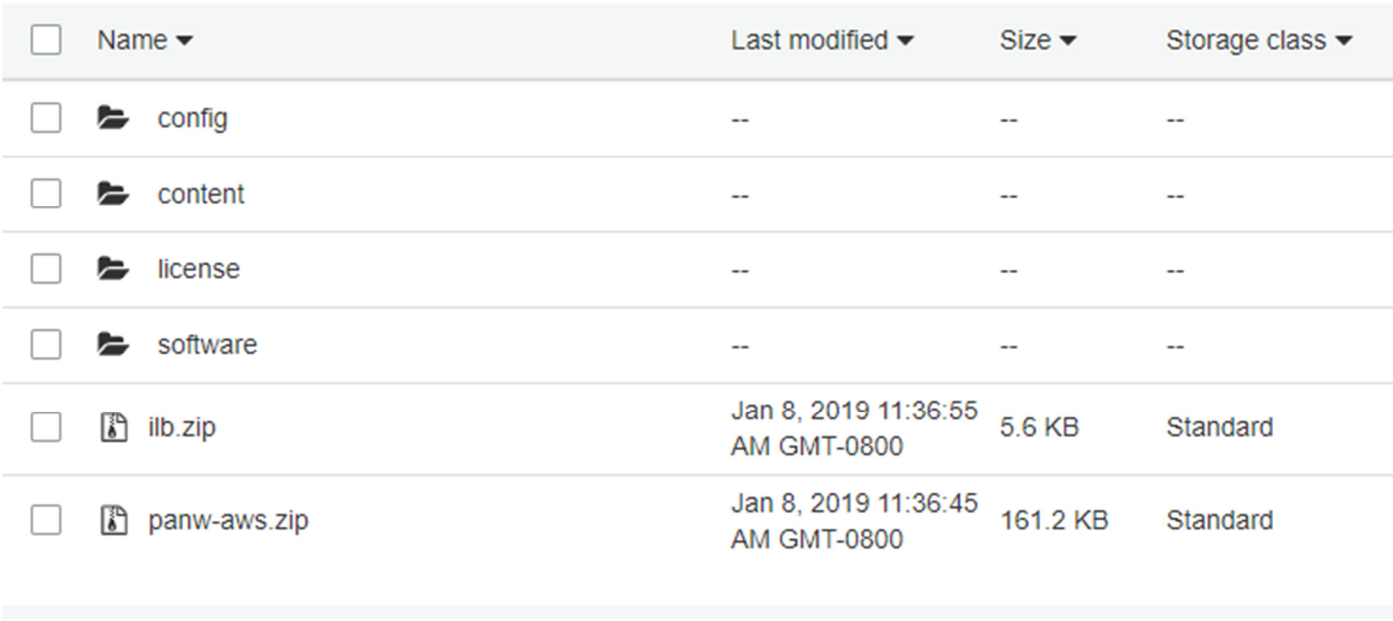
STEP 3 | (For BYOL only) Add the license auth code in the /license folder of the bootstrap package.

- Create a new .txt file with a text editor, such as Notepad and name it authcodes with NO extension.

- Add the auth code for your BYOL licenses to this file, then save. The auth code must support the number of firewalls that may be required for your deployment, so you must use an auth code bundle. If you use individual auth codes instead of a bundle, the firewall retrieves only the license key for the first auth code included in the file.



STEP 4 | Upload Lambda code for Firewall template (panw-aws-zip) and Application template (ilb.zip) to an S3 bucket. You can use the same S3 bucket that you use for bootstrapping.



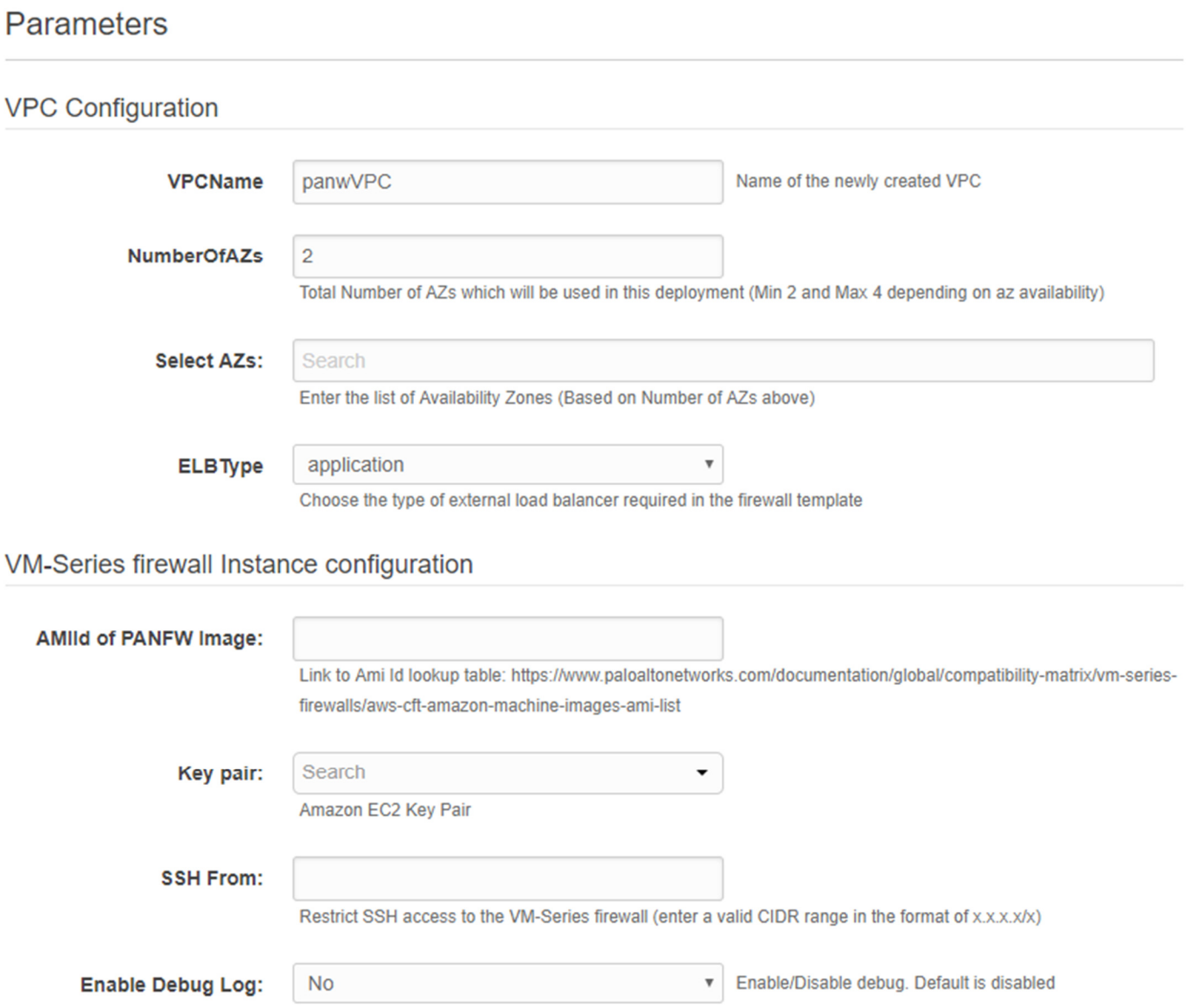
STEP 5 | Select the firewall template.

1. In the AWS Management Console, select **CloudFormation>Create Stack**.

2. Select **Upload a template to Amazon S3**, choose the firewall template and click **Open** and **Next**.
3. Specify the **Stack name**. The stack name allows you to uniquely identify all the resources that this template deploys.

STEP 6 | Configure the parameters for the VPC.

1. Be sure to select at least two availability zones

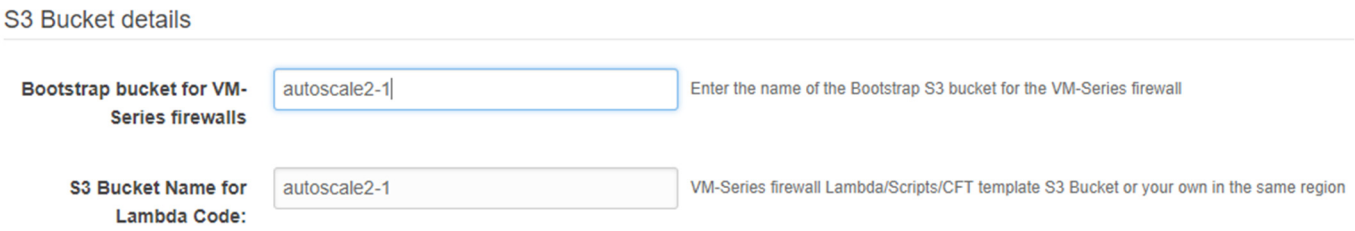


2. Look up the AMI ID for the VM-Series firewall and enter it. Make sure that the AMI ID matches the AWS region, PAN-OS version and the BYOL or PAYG licensing option you opted to use.

3. Select the EC2 **Key pair** (from the drop-down) for launching the firewall. To log in to the firewalls, you must provide the name of this key pair and the private key associated with it.
4. For the **SSH from field**, the firewalls will be managed by Panorama and do NOT have an EIP for the management interface. But just in case you decide to assign an EIP configure the IP range you would connect from.
5. Select **Yes** if you want to **Enable Debug Log**. Enabling the debug log generates more verbose logs that help with troubleshooting issues with the deployment. These logs are generated using the stack name and are saved in AWS CloudWatch.

By default, the template uses CPU utilization as the scaling parameter for the VM-Series firewalls. Custom PAN-OS metrics are automatically published to the CloudWatch namespace that matches the stack name you specified earlier.

STEP 7 | Specify the name of the Amazon S3 bucket(s).

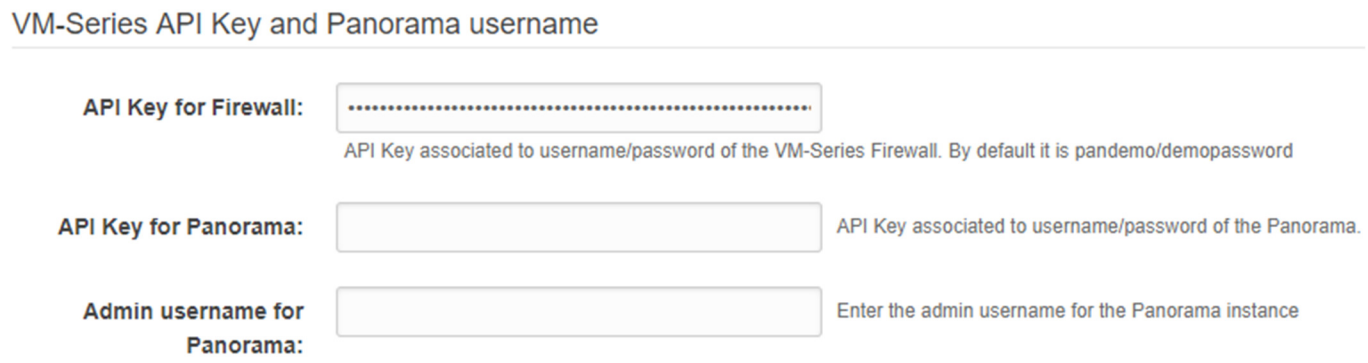


1. Enter the name of the S3 bucket that contains the bootstrap package.

2. If the bootstrap bucket is not set up properly or if you enter the bucket name incorrectly, the bootstrap process fails, and you cannot be able to log in to the firewall. Health checks for the load balancers also fail.

3. Enter the name of the S3 bucket that contains the panw-aws.zip file. As mentioned earlier you can use one S3 bucket for the Bootstrap and Lambda code.

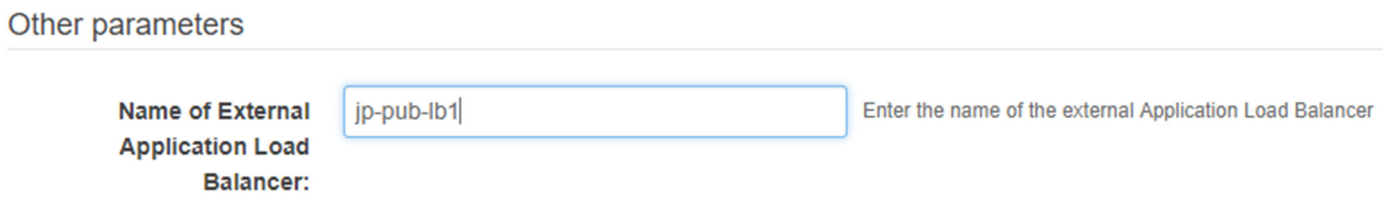
STEP 8 | Specify the keys for enabling API access to the firewall and Panorama.



1. Enter the key that the firewall must use to authenticate API calls. The default key is based on the sample bootstrap.xml file and you should only use it for testing and evaluation. For a production deployment, you must create a separate PAN-OS login just for the API call and generate an associated key.

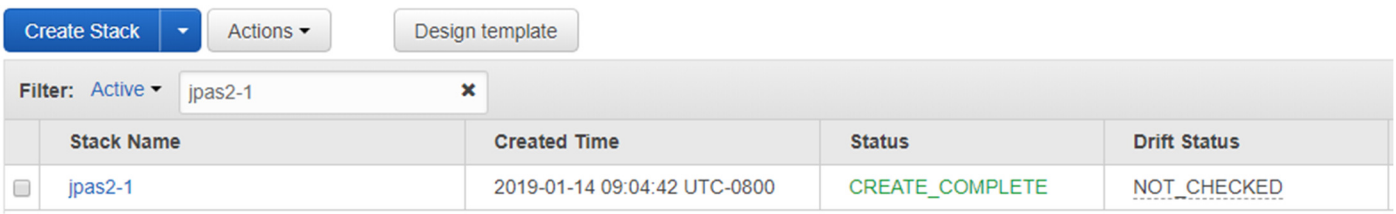
2. Enter the API Key to allow AWS Lambda to make API calls to Panorama. For a production deployment, you should create a separate login just for the API call and generate an associated key.

STEP 9 | Enter the name for the application load balancer.



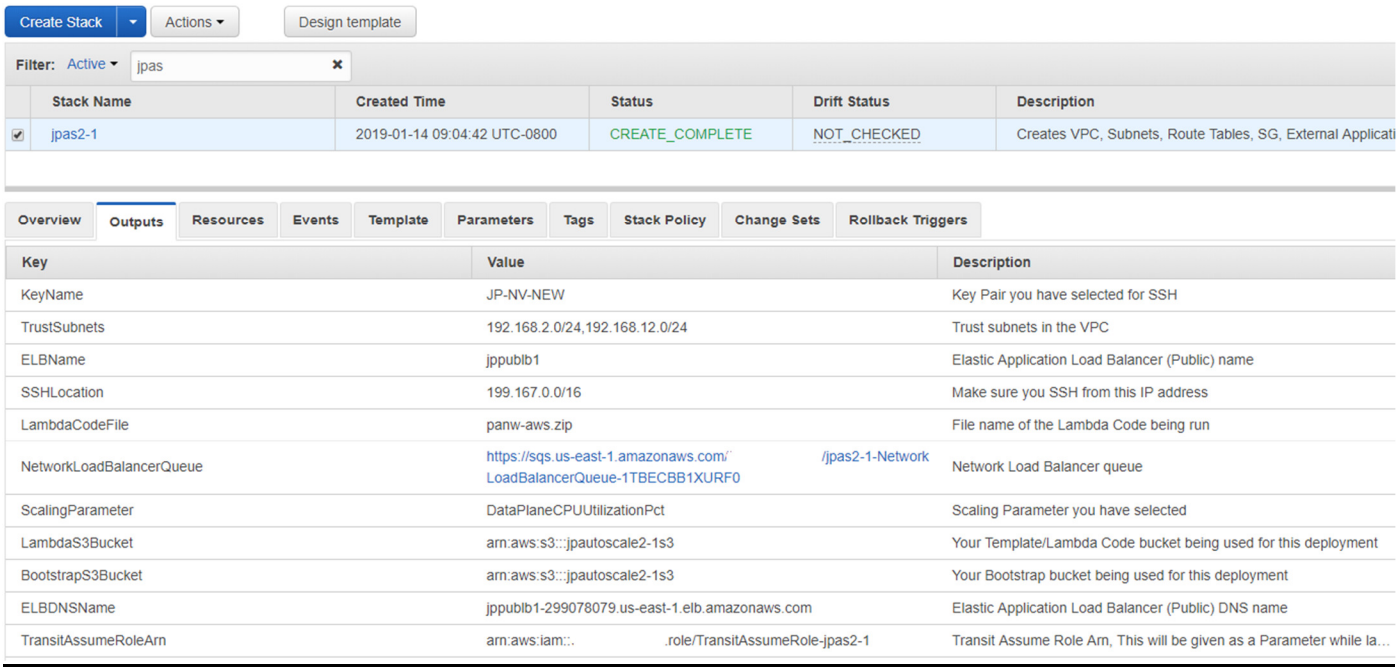
STEP 10 | Review the template settings and launch the template.

- Select I acknowledge that this template might cause AWS CloudFormation to create IAM resources.
- Click Create to launch the template. The CREATE_IN_PROGRESS event displays.
- On Successful deployment the status updates to CREATE_COMPLETE



STEP 11 | Verify that the template has launched all required resources.

- On the EC2 Dashboard, select **Auto Scaling Groups**. Verify that in each AZ, you have one ASG for the VM-Series firewalls. The ASG name prefix includes the stack name.
- On the AWS Management Console, select the stack name to view the **Output** for the list of resources.
- Your output should look like the output in the following image.
    - Take note of the Network Load Balancer Queue name.
    - Take note of the Elastic Load Balancer public DNS name.



You must use Panorama to access the user interface on the firewall.

- It may take up to 20 minutes for the firewalls to boot up and be available to handle traffic.
- When you finish testing or a production deployment, the only way to ensure charges stop occurring is to completely delete the stack. Shutting down instances, or changing the ASG maximum to 0 is not sufficient.

STEP 12 | Save the following information. You need to provide these values as inputs when deploying the application template.

- IP addresses of the NAT Gateway in each AZ. You need this IP address to restrict HTTPS access to your Panorama so that Lambda using the EIP's for the NAT Gateway can communicate with Panorama when needed. Save the ELBDNSName for the browser test at after launching the application template.

- Network Load Balancer SQS URL. A lambda function in the firewall stack monitors this queue so that it can learn about any network load balancers that you deploy and create NAT policy rules (one per application) in the Panorama that enable the firewalls to send traffic to the network load balancer IP address.

Launch the Application Template

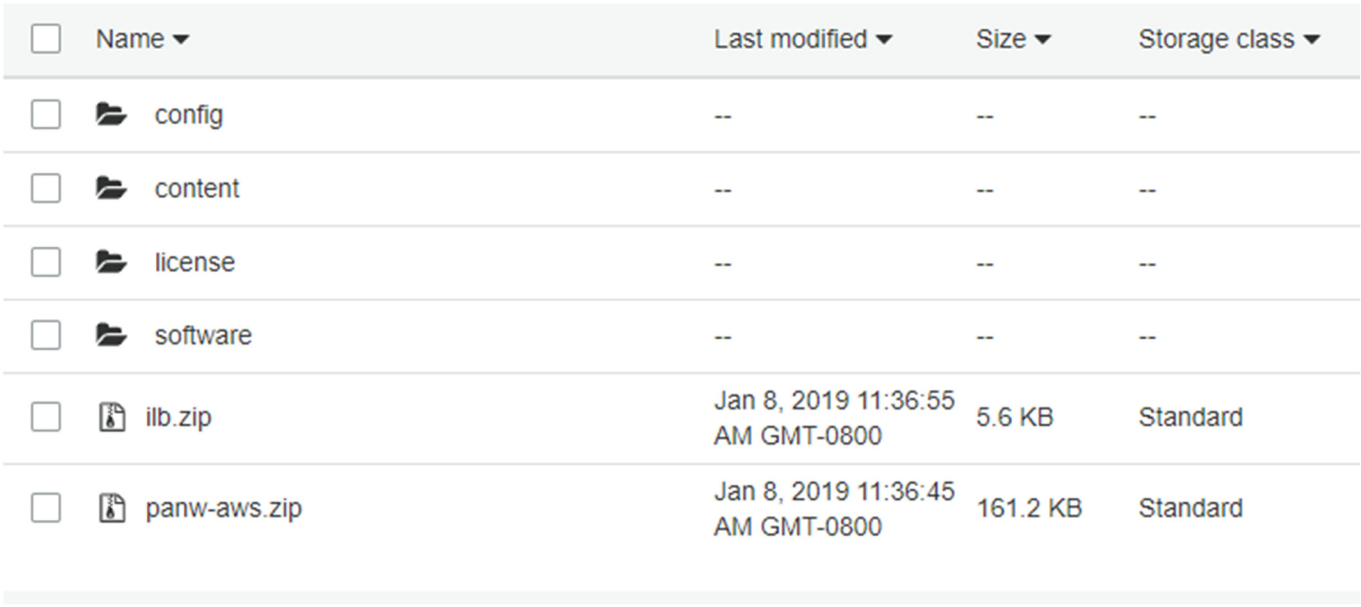
The application template allows you to complete the sandwich topology and is provided so that you can evaluate the auto scaling solution. This application template deploys either an application or network load balancer and a pair of web servers behind the auto scaling group of VM-Series firewalls, which you deployed above using the firewall template. Use this template to evaluate the solution but build your own template to deploy to production. For a custom template, make sure to enable SQS Messaging Between the application template and firewall template.

When launching the application template, you must select the template based on whether you want to deploy the application template within the same VPC in which you deployed the firewall

template or in a separate VPC. When deploying into a separate VPC cross account deployments are not supported in AWS Auto Scaling (v2.1-CS).

STEP 1 | Upload the ilb.zip file into an S3 bucket to use when launching the application template.

1. Upload the ilb.zip file to the S3 bucket you created earlier or to a new bucket. As mentioned earlier you can use the same bucket for everything.



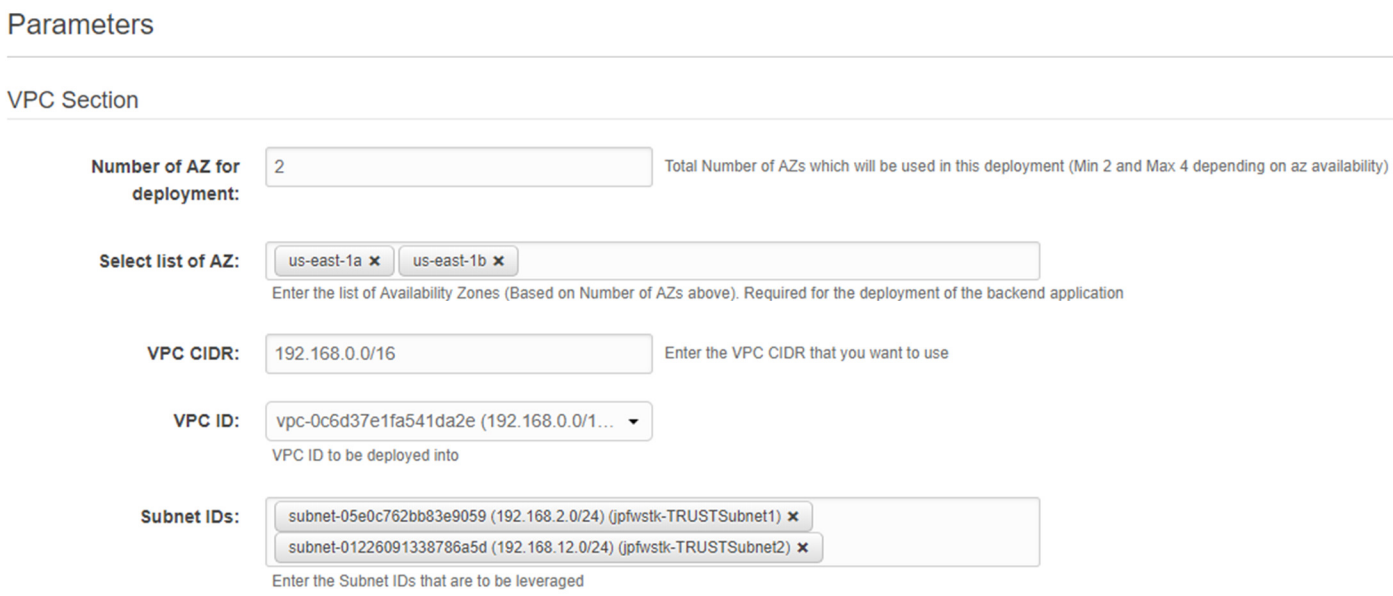
STEP 2 | Select the application launch template you wish you launch.

1. In the AWS Management Console, select **CloudFormation>Create Stack**.
2. Select **Upload a template to Amazon S3**, to choose the application template to deploy the resources that the template launches within the same VPC as the firewalls, or to a different VPC. Click **Open** and **Next**.
3. Specify the **Stack name**. The stack name allows you to uniquely identify all the resources that are deployed using this template.

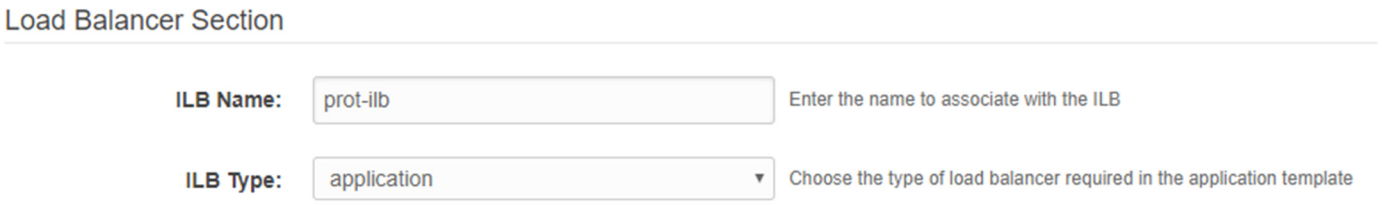
STEP 3 | Configure the parameters for the VPC and network load balancer.

1. Select the two Availability Zones that your setup will span in **Select list of AZ**. If you are deploying within the same VPC make sure to select the same Availability Zones that you selected for the firewall template.

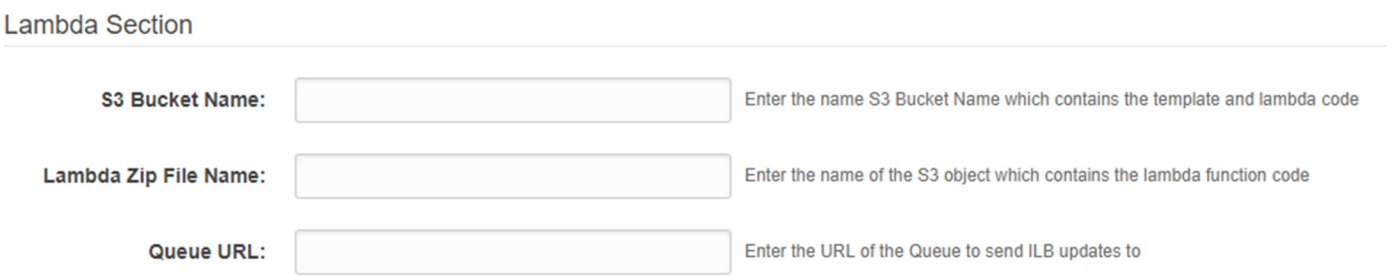
2. If deploying to a new VPC enter a **CIDR Block for the VPC**. The default CIDR is 192.168.0.0/16.
3. If deploying to the same VPC you will **select the previous VPC** and use the Trust subnets.



STEP 4 | Select the load balancer type.



STEP 5 | Configure the parameters for Lambda.



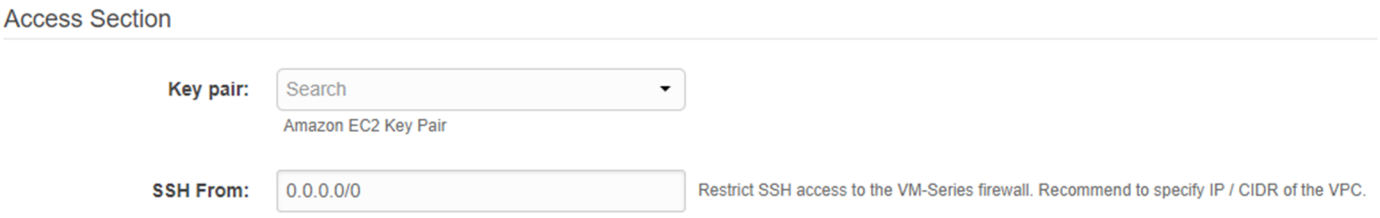
1. Enter the S3 bucket name where ilb.zip is stored.
2. Enter the name of the zip file name.
3. Paste the SQS URL that you copied earlier.

STEP 6 | Modify the web server EC2 instance type to meet your needs.

1. For evaluation and testing purposes the default value should work.

STEP 7 | Select the EC2 **Key pair** (from the drop-down) for launching the web servers. To log in to the web servers, you must provide the key pair name and the private key associated with it.

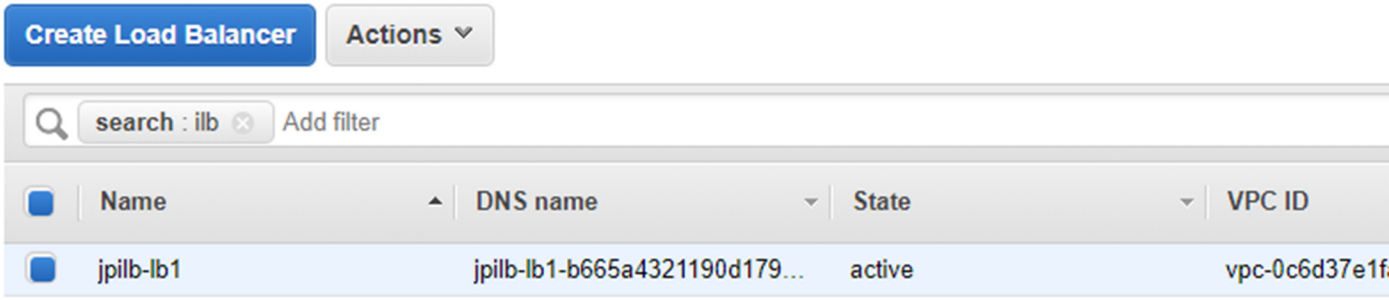
STEP 8 | Select the IP address of the network you will be accessing the servers from for management access only. Web traffic comes through the ELBDNS name you copied when you launched the firewall template.



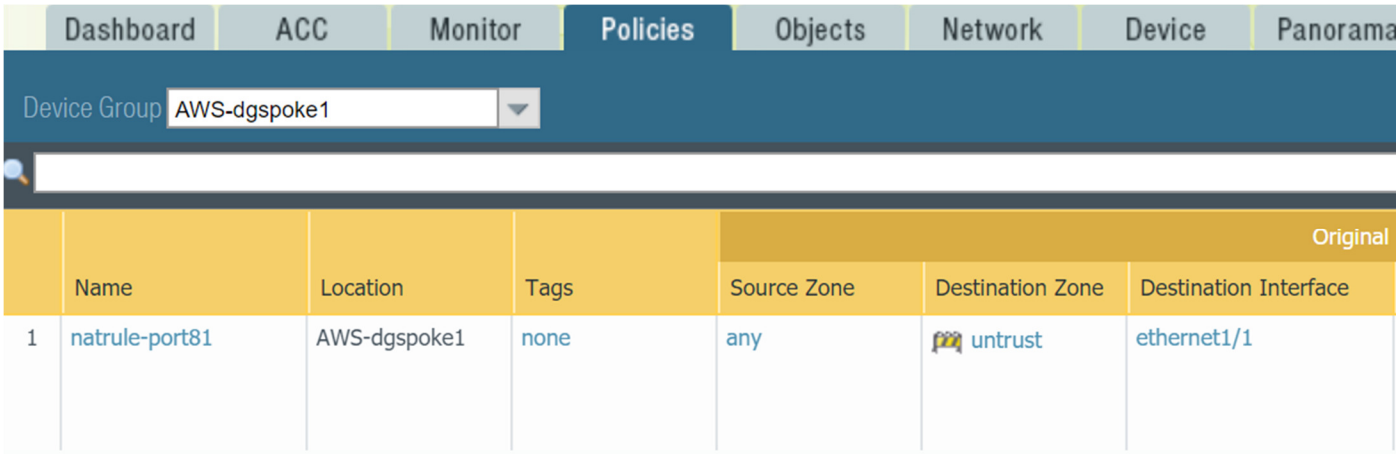
STEP 9 | Review the template settings and launch the template.

STEP 10 | After completion of the application template it can take up to 20 minutes for the web pages to become active. You will want to check the following.

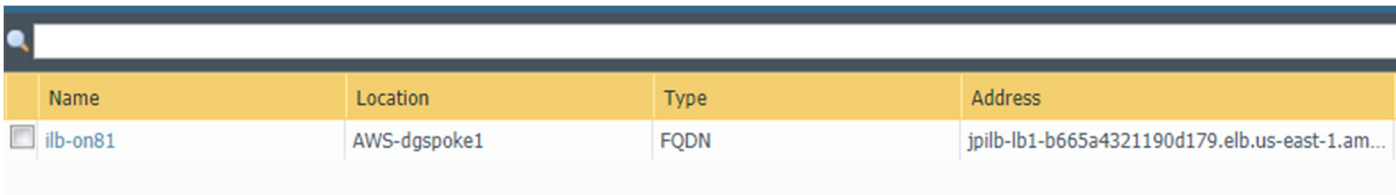
1. Verify that the application template load balancer is marked active.



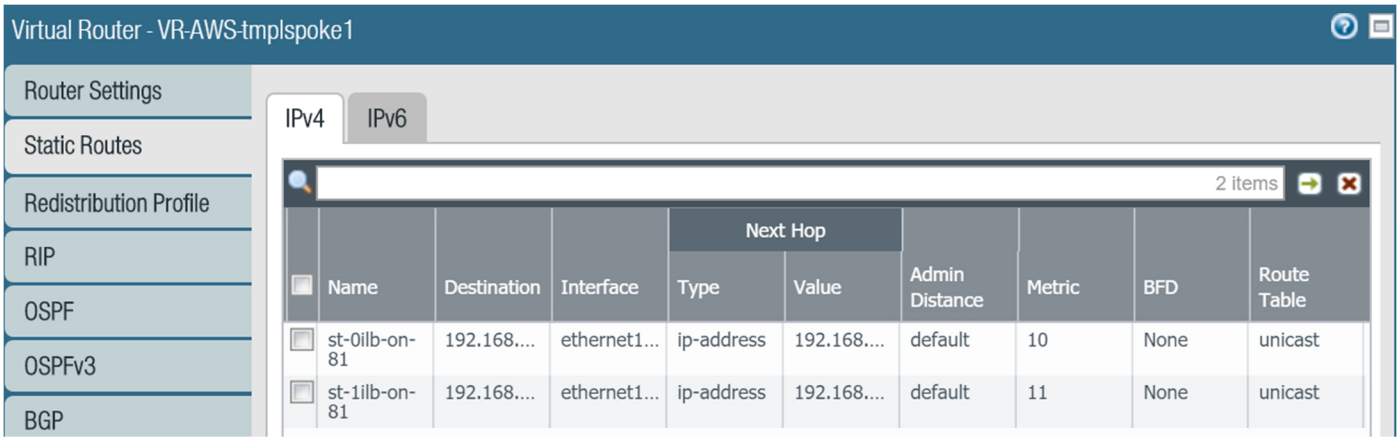
2. Verify that Panorama has a NAT object in the device group.



3. Verify that Panorama has an address object in the device group.



4. Verify that Panorama has static routes in the template stack.



STEP 11 | Get the ELBDNSName name you saved earlier for the application load balancer in Step 12 of the "Launch the Firewall" section. Enter the ELBDNSName into a web browser.

STEP 12| Upon successful launch your browser should look like this output.

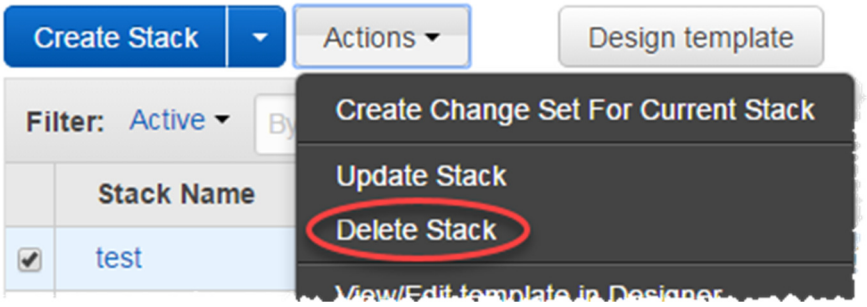
Congratulations, you have successfully launched VM-Series ASG CloudFormation. This file is coming from Webserver Region: us-east-1

StackID: arn:aws:cloudformation:us-east-1: :stack/jpappstk1/1c197cb0-18f4-11e9-b92e-0ab87eb901cc

StackName: jpappstk1

Template Cleanup

1. In the AWS Management Console, select **CloudFormation>Create Stack**.
2. Locate the firewall template and application template you launched previously and delete both templates.



For more information on deleting template stacks see the link below
https://docs.aws.amazon.com/AWSCloudFormation/latest/UserGuide/cfn-console-delete-stack.html

Note: Failure to delete your template stack will result in incurring charges from AWS.