Deploy the Firepower Threat Defense Virtual Auto Scale for AWS

Auto Scale Solution for FTDv on AWS

The following sections describe how the components of the Auto Scale solution work for the FTDv on AWS.

About the Auto Scale Solution

Cisco provides CloudFormation Templates and scripts for deploying an auto-scaling group of FTDv firewalls using several AWS services, including Lambda, auto scaling groups, Elastic Load Balancing (ELB), Amazon S3 Buckets, SNS, and CloudWatch.

FTDv Auto Scale in AWS is a complete serverless implementation (i.e. no helper VMs involved in the automation of this feature) that adds horizontal auto scaling capability to FTDv instances in the AWS environment.

The FTDv Auto Scale solution is a CloudFormation template-based deployment that provides:

- Completely automated FTDv instance registration and de-registration with the FMC.
- NAT policy, Access Policy, and Routes automatically applied to scaled-out FTDv instances.
- Support for Load Balancers and multi-availability zones.
- Support for enabling and disabling the Auto Scale feature.
- Works only with FMC; the Firepower Device Manager is not supported.

Supported Software Platforms

The FTDv Auto Scale solution is applicable to the FTDv managed by the FMC, and is software version agnostic. The Cisco Firepower Compatibility Guide provides Cisco Firepower software and hardware compatibility, including operating system and hosting environment requirements.

- The Firepower Management Centers: Virtual table lists Firepower compatibility and virtual hosting environment requirements for the FMCv on AWS.
- The Firepower Threat Defense Virtual Compatibility table lists Firepower compatibility and virtual hosting environment requirements for FTDv on AWS.

Note For purposes of deploying the AWS Auto Scale solution, the minimum supported Firepower version for FTDv on AWS is Version 6.4.

Auto Scale Use Case

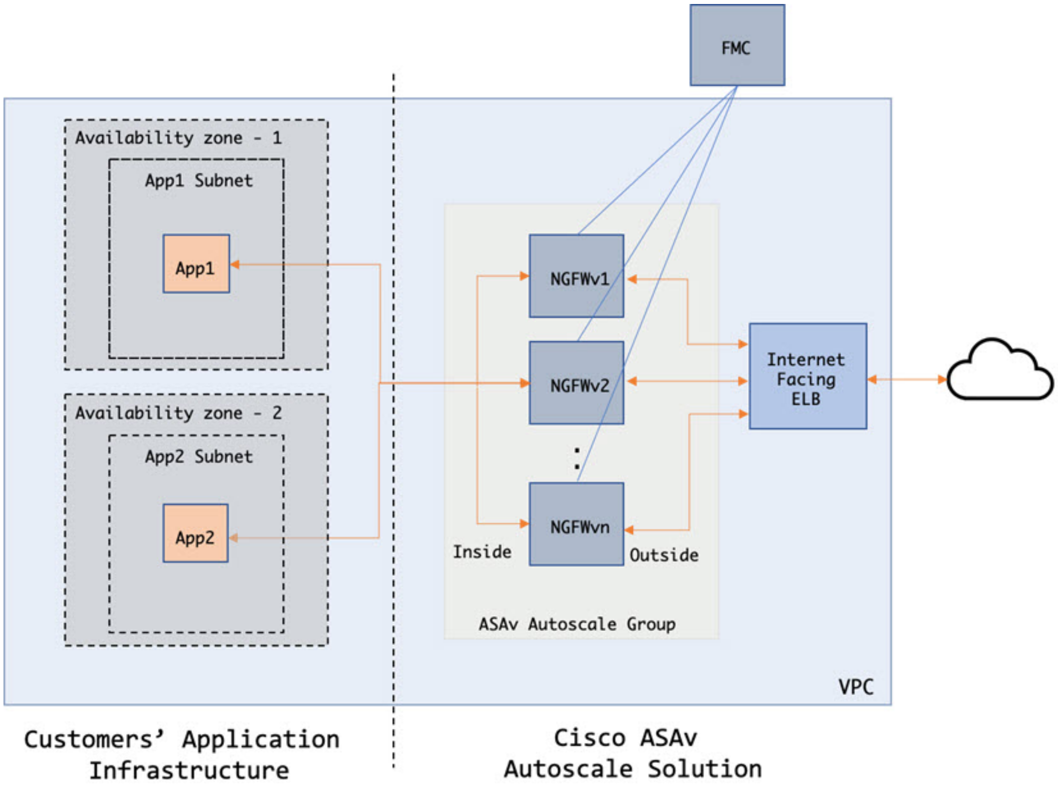
The Use Case for this FTDv AWS Auto Scale Solution is shown in Figure 1: FTDv Auto Scale Use Case Diagram, on page 3. Because the AWS Load Balancer allows only Inbound-initiated connections, only externally generated traffic is allowed to pass inside via the Cisco FTDv firewall. The Internet-facing Load Balancer will have a DNS name, and 0 to 4 ports can be kept open. Of those ports, 0 to 2 can be unsecured ports such as HTTP/80, and 0 to 2 can be secured ports such as HTTPS/443.

Note Secured ports need an SSL/TLS certificate, as described SSL Server Certificate, on page 6 in the Prerequisites.

The Internet-facing load balancer can be a Network Load Balancer or an Application Load Balancer. All of the AWS requirements and conditions hold true for either case. As indicated in the Use Case diagram, the right side of the dotted line is deployed via the FTDv templates. The left side is completely user-defined.

Note Application-initiated outbound traffic will not go through the FTDv.

Figure 1: FTDv Auto Scale Use Case Diagram



Port-based bifurcation for traffic is possible. This can be achieved via NAT rules; see Configure Objects, Device Group, NAT Rules, Access Policies in FMC, on page 13. For example, traffic on Internet-facing LB DNS, Port: 80 can be routed to Application-1; Port: 88 traffic can be routed to Application-2.

How the Auto Scale Solution Works

To scale FTDv instances in and out, an external entity called the Auto Scale Manager monitors metrics, commands an auto scale group to add or delete FTDv instances, registers and deregisters the FTDv devices with the managing FMC, and configures FTDv instances.

The Auto Scale Manager is implemented using AWS Serverless architecture and communicates with AWS resources, the FTDv, and the FMC. We provide CloudFormation templates to automate the deployment of Auto Scale Manager components. The template also deploys other resources required for complete solution to work.

Note Serverless Auto Scale scripts are only invoked by CloudWatch events, hence they only run when an instance is launched.

Auto Scale Solution Components

The following components make up the Auto Scale solution.

CloudFormation Template

The CloudFormation template is used to deploy resources required by Auto Scale solution in AWS. The template consists of:

- Auto Scale Group, Load Balancer, Security Groups, and other miscellaneous components.
- The template takes user input to customize the deployment.

Note The template has limitations in validating user input, hence it is the user's responsibility to validate input during deployment.

Lambda Funtions

The Auto Scale solution is a set of Lambda functions developed in Python, which gets triggered from Lifecycle hooks, SNS, CloudWatch event/alarm events. Basic functionality of this includes:

- Trigger Scale In / Scale Out operations.
- Register a new FTDv with the FMC.
- Configure a new FTDv via FMC.
- Unregister (remove) a scaled-in FTDv from the FMC.

Lambda Functions are delivered to customer in the form of a Python package.

Scale In / Scale Out Plugin

- Scale In / Scale Out Plugin ensures the correct number of Amazon EC2 instances are available to handle the load for the application.
- Scaling plugins can be configured by built-in AWS framework for autoscaling or by using custom Lambda functions.

Lifecycle Hooks

- Lifecycle hooks are used to get lifecycle change notification about an instance.
- In case of instance launch, a Lifecycle hook is used to trigger a Lambda function which can add interfaces to an FTDv instance, and register outside interface IPs to target groups.
- In case of instance termination, a Lifecycle hook is used to trigger a Lambda function to deregister an FTDv instance from the target group.

Simple Notification Service (SNS)

- Simple Notification Service (SNS) from AWS is used to generate events.
- Due to the limitation that there is no suitable orchestrator for Serverless Lambda functions in AWS, the solution uses SNS as a kind of function chaining to orchestrate Lambda functions based on events.

Auto Scale Solution Prerequisites

Download Deployment Files

Deployment scripts and templates for the FTDv – AWS Auto Scale solution are available from the GitHub repository at:

- https://github.com/CiscoDevNet/cisco-ftdv/tree/master/autoscale/aws

VPC

You should create the VPC as required for your application requirements. It is expected that the VPC have an Internet gateway with at least one subnet attached with a route to the Internet. Refer to the appropriate sections for the requirements for Security Groups, Subnets, etc.

Subnets

Subnets can be created as needed for the requirements of the application. The FTDv VM requires 3 subnets for operation as shown in the Use Case. Please note that if multiple availability zone support is needed, then subnets are needed for each zone as subnets are zonal properties within the AWS Cloud.

Note If multiple availability zone support is needed, then subnets are needed for each zone as subnets are zonal properties within the AWS Cloud

Outside Subnet

The Outside subnet should have route with '0.0.0.0/0' to the Internet gateway. This will contain the Outside interface of the FTDv, and also the Internet-facing NLB will be in this subnet.

Inside Subnet

This can be similar to the Application subnets, with or without NAT/Internet gateway. Please note that for FTDv health probes, it should be possible to reach the AWS Metadata Server (169.254.169.254) via port 80.

Management Subnet

This subnet is the FTDv Management interface, assigned an elastic IP address (EIP) and required to have the default route to Internet.

Note To avoid an EIP on the Management interface, see Appendix - Lambda Function to Access VPC Private IPs, on page 19.

Application Subnets

There is no restriction imposed on this subnet from the Auto Scale solution, but in case the application needs Outbound connections outside the VPC, there should be respective routes configured on the subnet. This is because outbound-initiated traffic does not pass through Load Balancers. See the AWS Elastic Load Balancing User Guide.

Security Groups

All connections are allowed in the provided Auto Scale Group template. You need only the following connections for the Auto Scale Solution to work.

Table 1: Required Ports

Port	Usage	Subnet
8305	FMC to FTDv Secured tunnel connection	Management subnets
Health Probe port (default: 8080)	Internet-facing Load Balancer health probes	Outside, Inside Subnets
Application ports	Application data traffic	Outside, Inside Subnets

Security Groups or ACls for the FMC Instance

To allow HTTPS connections between Lambda Functions and the FMC, a range of IP addresses should be white-listed. If a range of allowable IP addresses is not possible, then you should manually place the Lambda functions in the VPC. See Appendix - Lambda Function to Access VPC Private IPs, on page 19.

Afterwards, the FTDv and the FMC Security Groups or ACLs can be updated with only the NAT Gateway IP address.

Amazon S3 Bucket

Amazon Simple Storage Service (Amazon S3) is an object storage service that offers industry-leading scalability, data availability, security, and performance. You can place all the required files for both the firewall template and the application template in the S3 bucket.

When templates are deployed, Lambda functions get created referencing Zip files in the S3 bucket. Hence the S3 bucket should be accessible to the user account.

SSL Server Certificate

If the Internet-facing Load Balancer has to support TLS/SSL, a Certificate ARN is required. Refer to the following links for more information:

- Working with Server Certificates
- Create a Private Key and Self-Signed Certificate for Testing
- Create AWS ELB with Self-Signed SSL Cert (Third-party link)

Example of ARN: arn:aws:iam::[AWS Account]:server-certificate/[Certificate Name]

Lambda Layer

A Lambda Layer is required to be created to provide few Python libraries for Lambda functions.

A file named ***autoscale_layer.zip*** needs to be created in this directory to provide some essential Python libraries to Lambda functions. The following libraries needs to be available to the lambda function:

```
pycrypto==2.6.1
paramiko==2.7.1
requests==2.23.0
scp==0.13.2
jsonschema==3.2.0
```

The ***autoscale_layer.zip*** can be created in a Linux environment, such as Ubuntu 18.04 with Python 3.6 installed.

```
#!/bin/bash
mkdir -p layer
virtualenv -p /usr/bin/python3.6 ./layer/
source ./layer/bin/activate
pip3 install pycrypto==2.6.1
pip3 install paramiko==2.7.1
pip3 install requests==2.23.0
pip3 install scp==0.13.2
pip3 install jsonschema==3.2.0
echo "Copy from ./layer directory to ./python\n"
mkdir -p ./python/.libs_cffi_backend/
cp -r ./layer/lib/python3.6/site-packages/* ./python/
cp -r ./layer/lib/python3.6/site-packages/.libs_cffi_backend/* ./python/.libs_cffi_backend/
zip -r autoscale_layer.zip ./python
```

Use the ***autoscale_layer.zip*** file in S3 bucket to create a Lambda Layer. Record the ARN for further usage.

Example of ARN:

```
arn:aws:lambda:us-east-1:[AWS Account]:layer:[Layer Name]:[version]
```

See AWS Lambda layers for more information.

KMS Master Key

This is required if the FMC and FTDv passwords are in encrypted format. Otherwise this component is not required. Passwords should be encrypted using only the KMS provided here. If KMS ARN is entered on CFT, then passwords have to be encrypted. Otherwise passwords should be plain text.

For more information about master keys and encryption, see the AWS document Creating keys and the AWS CLI Command Reference about password encryption and KMS.

Example:

```
$ aws kms encrypt --key-id <KMS-ARN> --plaintext 'MyC0mplIc@tedProtect1oN'
{
    "KeyId": "KMS-ARN",
    "CiphertextBlob":
"AQICAHgcQFAGtz/hvaxMtJvY/x/rfHnKI3clFPpSXUU7HQRnCAFwfXhXHJAHL8tcVmDqurALAAAAajBoBgkqhki
G9w0BBwagWzBZAgEAMFQGCSqGSIb3DQEHATAeBglghkgBZQMEAS4wEQQM45AIkTqjSekX2mniAgEQgCcOav6Hhol
+wxpWKtXY4y1Z1d0z1P4fx0jTdosfCbPnUExmNJ4zdx8="
}
$
```

The value of *CiphertextBlob* key should be used as a password.

Python 3 Environment with AWS CLI

A *utility.py* file can be found in the cloned repository top directory. It should be used to Zip files after Configuration.json modification and upload to required S3 bucket. In order to run the *utility.py* file there should be a Python 3 environment with AWS CLI setup.

https://docs.aws.amazon.com/cli/latest/userguide/cli-chap-configure.html

Auto Scale Deployment

Preparation

It is expected that the Application is either deployed or its deployment plan is available.

Input Parameters

The following input parameters should be collected prior to deployment.

Table 2: Auto Scale Input Parameters

Parameter	Allowed Values/Type	Description
Autoscale Manager Stack Template URL	String	This is S3 URL from which the *deploy.yaml* Nested Stack imports the Auto Scale Manager CloudFormation yaml file.
Autoscale Group Stack Template URL	String	This is S3 URL from which the *deploy.yaml* Nested Stack imports the Auto Scale Group CloudFormation yaml file.
VPC Id	String	This should be configured as per AWS requirements. Example: vpc-81f042fb
Number of Availability Zones	Integer	The number of availability zones that FTDv should span across, between 1 and 3. In the case of an ALB deployment, the minimum value is 2, as required by AWS. Example: 2
List of Availability Zones	Comma separated string	A comma-separated list of zones in order. **Note** The order in which these are listed matters. Subnet lists should be given in the same order. Example: us-east-1a, us-east-1b, us-east-1c

Parameter	Allowed Values/Type	Description
S3 Bucket Name	String	The bucket name, please note this should be accessible from your AWS account. Example: mygroup-autoscale-lambda
Group Name Prefix	String	This is the Auto Scale Group Name Prefix. Example: Cisco-FTDv
Pod Number	Integer	The Pod number. Example: 1
Minimum Number of Instances	Integer	The minimum number of instances that the Auto Scale Group can spin up based on demand. Example: 5
Maximum Number of Instances	Integer	The maximum number of instances that the Auto Scale Group will spin up based on demand. This many instances will be created as part of a stack deployment. **Note** Recommended to keep it 1 initially. Example: 1
Scaling Action	String	"Custom Scaling via Lambda Functions" or "AWS provided Dynamic Scaling". "AWS provided Dynamic Scaling" is Scaling Policies from AWS Auto Scale Group itself. It is Simple Scaling Type and in case of Scale In, the newest instance gets deleted. "Custom Scaling via Lambda Functions" is using the Lambda function, a custom logic is written. Example: Custom Scaling via Lambda Functions
Scale Out Threshold (CPU)	Integer	The Scale Out Threshold CPU value. The minimum value is 1 and the maximum value is 99; please note this is Group Average. Example: 70
Scale Out Data Points	Integer	If '3' is the value then 3 times threshold breach should happen in 3x60 seconds. Example: 3
Scale In Threshold (CPU)	Integer	The Scale In Threshold CPU value. The minimum value is 0 and maximum value is 99; please note that the Scale In threshold should be lower than the Scale Out threshold. Example: 15

Parameter	Allowed Values/Type	Description
Scale In Data Points	Integer	If '8' is the value then 8 times threshold breach should happen in 8x60 seconds. Example: 8
Email-Id for Notifications	String	Auto Scale events will be sent to this email address. Example: admin@company.com
Cool-Down Period	Integer	The time, in seconds, taken for a Scale Out action to bring up new FTDv and configure it to receive traffic. Example: 1320
Disable/Enable Debug Logs	String	Enable/disable debug logs. The default is debug enabled. Example: disable
Load Balancer Type	String	The type of Internet-facing load balancer, either “application” or “network”. Example: application
Number of Unsecured Ports	Integer	The total number of unsecured ports to be opened. Example: 2
List of Unsecured Ports	Comma separated integers	A comma-separated list of port numbers to be opened on a Load Balancer, with either HTTP or TCP as protocol based on chosen Load Balancer type. Min: 0 and Max: 2. Make sure ports are valid TCP ports. Example: 80, 8000
ARN of SSL/TLS Certificate	String	The ARN of an existing TLS/SSL Certificate in AWS ACM. If specified, TLS with port 443 gets opened on Load Balancer. If any secured port has to be opened, the Certificate ARN has to be entered. Otherwise this is optional. Example: arn:aws:iam::[AWS Account]:server-certificate/[Certificate Name]
Number of Secured Ports	Integer	The total number of secured ports to be opened. Example: 2
List of Secured Ports	Comma separated integers	A comma-separated list of port numbers to be opened on a Load Balancer, with either HTTPS or TLS as protocol based on chosen Load Balancer type. Min: 0 and Max: 2. Make sure ports are valid TCP ports. Example: 443, 8443

Parameter	Allowed Values/Type	Description
Health Check Port	Integer	This port is used by the Target group for health probes. The default is 8080. Health probes arriving at this port on FTDv will be routed to the AWS Metadata server. It should be a valid TCP port. Example: 8080
Connection Draining Period	Integer	The time (in seconds), for the connection drain to occur for graceful shutdown. The default is 3 minutes (180 seconds). Example: 180
FTDv Instance Type	String	The type of AWS EC2 instances type. Only instances that support FTDv should be used. See Firepower Release Notes. Example: C4.xlarge
FTDv AMI Id	String	FTDv AMI Id (Valid Cisco FTDv AMI ID). Example: ami-0de5d3956a718f517 Note: Please choose correct AMI ID as per the region and desired version of the image. The Auto Scale feature supports Firepower version 6.4+, BYOL/PAYG images. In either case you should have accepted a License in the AWS marketplace. In the case of BYOL, please update 'licenseCaps' key in Configuration JSON with features such as 'BASE', 'MALWARE', 'THREAT', 'URLFilter' etc. During registration, the FMC allocates required the licenses. If the FMC does not have licenses then the FTDv becomes non-compliant.
List of Management Subnets	Comma separated list	A comma-separated list of management subnet-ids. The list should be in the same order as the corresponding availability zones. Example: subnet-0778e74f6e603b13b, subnet-02d1d7842f5f111c8, subnet-01c1d55b157335002
List of Inside Subnets	Comma separated list	A comma-separated list of inside subnet-ids. The list should be in the same order as the corresponding availability zones. Example: subnet-0379e9e745772e8f3, subnet-0a199b943939b9b6f, subnet-0ac38cf812f69d23c

Parameter	Allowed Values/Type	Description
List of Outside Subnets	Comma separated list	A comma-separated list of outside subnet-ids. The list should be in the same order as the corresponding availability zones. Example: subnet-086096d4a09745b70, subnet-08566e6c2f99a4e6a, subnet-0bd03219da105a27c
Auto Scale Manager Name Prefix	String	User-defined prefix. Example: Cisco-FTDv
Lambda Layer ARN	String	This is a regional resource. Example: arn:aws:lambda:us-east-1:[AWS Account]:layer:[Layer Name]:[version]
Configuration JSON File Name	String	User-defined file name. Example: Configuration.json
KMS ARN	String	The ARN of an existing KMS (AWS KMS key to encrypt at rest). If specified, the FMC and FTDv passwords should be encrypted. The password encryption should be done using only the specified ARN. Generating Encrypted Password Example: " aws kms encrypt --key-id <KMS ARN> --plaintext <password> ". Please used such generated passwords as shown. Example: arn:aws:kms:us-east-1:[AWS Account]:key/7d586a25-5875-43b1-bb68-a452e2f6468e
FMC IP	Numeric string	The IP address of the managing FMC to which the newly deployed FTDv should be registered. Example: 10.10.17.21
FMC User Name	String	The Network-Admin or higher privileged user created in the managing FMC. Example: apiuser-1
FMC Password	String	Please use a plain text password if KMS ARN is not mentioned. If mentioned, then an encrypted password should be used. Example: Cisco123@ or AQICAHgcQAtz/hvaxMtJvY/x/rnKI3clFPpSXUHQRnCAajB
FTDv Password	String	Please use a plain text password if KMS ARN is not mentioned. If mentioned, then an encrypted password should be used. Example: Cisco123@ or AQIAgcQFAGtz/hvaxMtJvY/x/rfHnKI3clFPpSXUU7HQRnCB

Configure Objects, Device Group, NAT Rules, Access Policies in FMC

You can manage the FTDv using the Firepower Management Center (FMC), a full-featured, multidevice manager on a separate server. The FTDv registers and communicates with the FMC on the Management interface that you allocated to the FTDv virtual machine. See About Firepower Threat Defense Virtual with Firepower Management Center for more information.

All the objects used for FTDv configuration should be created by user.

Important A device group should be created and rules should be applied on it. All the configurations applied on device group will be pushed to FTDv instances.

Objects

Create the following objects:

Table 3: FMC Configuration Objects for FTDv Management

Object Type	Name	Value
Host	aws-metadata-server	169.254.169.254
Port	health-check-port	8080/any other port as required
Zone	Inside/ any other name	—
Zone	Outside/ any other name	—

NAT Policy

A typical NAT rule converts internal addresses to a port on the outside interface IP address. This type of NAT rule is called interface Port Address Translation (PAT). See Configure NAT in Managing the Firepower Threat Defense Virtual with the Firepower Management Center for information about the NAT policy.

One mandatory rule is required in your NAT policy:

- Original-sources: any-ipv4
- Original-destination-port: 8080/or any health port that user configures
- Translated-destination: aws-metadata-server
- Translated-destination-port: 80

Similarly, any data-traffic NAT rules can be added, so that this configuration will be pushed to FTDv devices.

Access Policy

Configure access control to allow traffic from inside to outside. An Access Policy with all required policies can be created, health port object should be allowed such that traffic on this port is allowed to reach. See Configure Access Control in Managing the Firepower Threat Defense Virtual with the Firepower Management Center for information about the Access Policy.

Update the Configuration JSON file

The ***Configuration.json*** file can be found in the *autoscale_manager* folder, which is part of the archive ZIP obtained from the GitHub repository. Please note the JSON key should not be changed. Any static routes for the FTDv VM should be configured in the JSON file.

See the following for an example of a static route configuration.

```
{
  "interface": "inside",
  "network": "any-ipv4",
  "gateway": "",
  "metric": "1"
}
```

All the values in the JSON file are modifiable according to your requirements, except the default FTDv password.

Upload Files to Amazon Simple Storage Service (S3)

Once the Configuration.json file is modified and all of the required parameters are collected, the files should be uploaded to the Amazon S3 bucket. Please note that *autoscale_manager*, *autoscale_grp*, and *scale_functions* should be Zipped and uploaded.

The *utility.py* file in the root directory of the GitHub clone will do that for you. Once the Configuration.json file is modified, please run the following command (Python 3.6 environment with AWS CLI configured):

```
$ python utility.py --create-zip-file true --upload-file true --s3-bucket
mygroup-autoscale-lambda
```

This will Zip the required files and upload to the S3 bucket.

Note You can manually Zip the files, however this will create some issue with the directory structure for the Lambda functions. Therefore we recommend that you create the Zip files through the *utility.py* function.

In case only the Zip file creation is needed, without the need to upload to the S3 bucket, then please run the following command and upload to the S3 bucket manually (this is helpful if AWS CLI is not setup).

```
$ python utility.py --create-zip-file true
```

In case of manual upload, please upload Zip files, YAML files.

Deploy a Nested Stack

After all of the prerequisites are completed for deployment, you can create the AWS CloudFormation stack.

Use the *deploy.yaml* file in top directory of the cloned repo.

Provide the parameters as collected in Input Parameters, on page 8.

Validate Deployments

Once the template deployment is successful, you should validate that the Lambda functions and the CloudWatch events are created according to the *asm.yaml* and *asg.yaml* CloudFormation template. Subscription confirmation emails are sent for email notification.

Auto Scale Maintenance Tasks

Scaling Processes

This topic explains how to suspend and then resume one or more of the scaling processes for your Auto Scale group.

Start and Stop Scale Out Actions

To start and stop scale out actions, follow these steps.

- For AWS Dynamic Scaling—Refer to the following link for information to enable or disable scale out actions:

  Suspending and Resuming Scaling Processes

- For AWS Custom Lambda—Navigate to CloudWatch CPU upper threshold alarm and edit the alarms to add or remove the Scale Out SNS topic.

Start and Stop Scale In Actions

To start and stop scale in actions, follow these steps.

- For AWS Dynamic Scaling—Refer to the following link to enable or disable scale in actions:

  Suspending and Resuming Scaling Processes

- For AWS Custom Lambda—Navigate to CloudWatch CPU upper threshold alarm and edit the alarms to add or remove the Scale In SNS topic.

Health Monitor

A health monitor is configured in such a way that, if unhealthy IP targets increase more than or equal to 1, remains 60 minutes then an SNS event is published.

- If there are unhealthy IPs which belong to a valid FTDv VM, that instance gets deleted.
- If those IPs are not from a valid FTDv VM, then only IPs are removed from the Target Group.

Disable Health Monitor

To disable a health monitor, navigate to the SNS subscription. From the AWS Lambda subscription for ASG group topic, delete the subscription. Optionally, you can also delete email subscription.

Enable Health Monitor

To enable a health monitor, navigate to the SNS subscription. Create a subscription and choose the correct Topic ARN (Auto Scale Group Topic ARN), AWS Lambda as protocol. Further, choose Auto Scale Lifecycle hook lambda as the Lambda ARN.

Disable Lifecycle Hooks

In the unlikely event that Lifecycle hook needs to be disabled, if disabled it won't add additional interfaces to Instances. It can also cause a series of failed deployment of FTDv instances.

Disable Auto Scale Manager

To disable Auto Scale Manager, respective CloudWatch Events "notify-instance-launch" and "notify-instance-terminate" should be disabled. Disabling this won't trigger Lambda for any new events. But already executing Lambda actions will continue. There is no abrupt stop of Auto Scale Manager. Trying abrupt stopping by stack deletion or deleting resources can cause an indefinite state.

Load Balancer Targets

Because the AWS Load Balancer does not allow instance-type targets for instances having more than one network interface, the Gigabit0/1 interface IP is configured as a target on Target Groups. As of now however, the AWS Auto Scale health checks work only for instance-type targets, not IPs. Also, these IPs are not automatically added or removed from target groups. Hence our Auto Scale solution programmatically handles both of these tasks. But in the case of maintenance or troubleshooting, there could be a situation demanding manual effort to do so.

Register a Target to a Target Group

To register an FTDv instance to the Load Balancer, its Gigabit0/1 instance IP (outside subnet) should be added as a target in Target Group(s). See Register or Deregister Targets by IP Address.

Deregister a Target to a Target Group

To deregister an FTDv instance to the Load Balancer, its Gigabit0/1 instance IP (outside subnet) should be deleted as a target in Target Group(s). See Register or Deregister Targets by IP Address.

Instance Stand-by

AWS does not allow instance reboot in the Auto Scale group, but it does allow a user to put an instance in Stand-by and perform such actions. However, this works best when the Load Balancer targets are instance-type. But FTDv VMs cannot be configured as instance-type targets, because of multiple network interfaces.

Put an Instance in Stand-by

If an instance is put into stand-by, its IP in Target Groups will still continue to be in the same state until the health probes fail. Because of this, it is recommended to deregister respective IPs from the Target Group before putting the instance into stand-by state; see Deregister a Target to a Target Group, on page 16 for more information.

Once the IPs are removed, see Temporarily Removing Instances from Your Auto Scaling Group.

Remove an Instance from Stand-by

Similarly you can move an instance from stand-by to running state. After removal from stand-by state, the instance's IP should be registered to Target Group targets. See Register a Target to a Target Group, on page 16.

For more information about how to put instances into stand-by state for troubleshooting or maintenance, see the AWS News Blog.

Remove/Detach Instance from Auto Scale Group

To remove an instance from the Auto Scale group, first it should be moved to stand-by state. See "Put Instances on Stand-by". Once the instance is in the stand-by state it can be removed or detached. See Detach EC2 Instances from Your Auto Scaling Group.

There won't be any changes on the FMC side. Any changes required should be manually performed.

Terminate an FTDv Instance

To terminate an instance it should be put into stand-by state; see Instance Stand-by, on page 16. Once the instance is in stand-by, you can proceed to terminate.

Instance Scale-In Protection

To avoid an accidental removal of any particular instance from the Auto Scale group, it can be made as Scale-In protected. If an instance is Scale-In protected, it won't be terminated due to a Scale-In event.

Please refer to the following link to put an instance into Scale-In protected state.

https://docs.aws.amazon.com/autoscaling/ec2/userguide/as-instance-termination.html

Important It is recommended to make the minimum number of instances which are healthy (the target IP should be healthy, not just the EC2 instance) as Scale-In protected.

Change Credentials and FTDv Registration IDs

Any changes in configuration won't be automatically reflected on already running instances. Changes will be reflected on upcoming devices only. Any such changes should be manually pushed to already existing devices.

Change the FMC User Name and Password

In the case of changes to the FMC IP, username, or password—the respective changes should be performed on Auto Scale Manager Lambda function environment variables. See Using AWS Lambda Environment Variables.

When Lambda runs next time, it will reference the changed environment variables.

Note Environment variables are directly fed to Lambda Functions. There is no password complexity check here.

Change the FTDv Admin Password

A change to the FTDv password requires the user to change it on each device manually for running instances. For new FTDv devices to be onboarded, the FTDv password will be taken from the Lambda environment variables. See Using AWS Lambda Environment Variables.

Change Registration and NAT IDs

For new FTDv devices to be onboarded with different registration and NAT IDs, for FMC registration this information should be changed in Configuration.json file. The Configuration.json file can be located in Lambda resource page.

Changes to Access Policy and NAT Policy

Any changes to Access policies or NAT policies are automatically applied to upcoming instances with the help of the Device Group assignment. However, to update existing FTDv instances you need to manually push configuration changes and deploy them from the FMC.

Changes to AWS Resources

You can change many things in AWS post deployment, such as the Auto Scale Group, Launch Configuration, CloudWatch events, Scaling Policies etc. You can import your resources into a CloudFormation stack or create a new stack from your existing resources.

See Bringing Existing Resources Into CloudFormation Management for more information about how to manage changes performed on AWS resources.

Collect and Analyze CloudWatch Logs

In order to export CloudWatch logs please refer to Export Log Data to Amazon S3 Using the AWS CLI.

Auto Scale Troubleshooting

Enable/Disable Debug Logging

While deploying stacks, there is an option to set True or False for debug logs. Please note that debug logs are very descriptive and will include a lot of unnecessary logging detail from the AWS side. You can change logging after deployment through Lambda environment variables.

- Debug Logging for Auto Scale Manager—To disable logging from the Auto Scale Manager Lambda function, navigate to the Lambda function manager and change the DEBUG_DISABLED variable to “false”.
- Debug Logging for Auto Scale Group—To disable logging from Auto Scale Manager Lambda function, navigate to the Lambda lifecycle function and change DEBUG_DISABLED variable to “false”.

Note When debug is enabled, error messages getting logged as debug logs are not problematic and are error-handled.

Check Input Parameters Post Deployment

You can verify the input parameters to a CloudFormation stack in the AWS CloudFormation Console. Navigate to the required stack and check the parameter tab. You can also check inputs to Lambda Functions on the Lambda Functions environment variables tab. The configuration.json file can also be viewed on the Auto Scale Manager Lambda function itself.

Appendix - Lambda Function to Access VPC Private IPs

To enforce the Lambda Function to access VPC private IP addresses (by default the Lambda Function uses an IP address provided by AWS from its EIP pool), you need to make the following changes from the AWS Console.

AWS Lambda Functions are global in nature having AWS-provided public IPs for connections to various AWS Services, FTDv SSH connection, and FMC HTTPS connection. It is possible to have the Lambda Function access VPC elements (i.e. the FTDv instance) with a private IP address itself by putting the Lambda Function in the same VPC, with a subnet having a NAT gateway as the default route.

This configuration can be done post-deployment of the Auto Scale solution. For more information refer this link:

https://aws.amazon.com/premiumsupport/knowledge-center/internet-access-lambda-function/

After you make these changes, it will be possible to restrict FMC and FTDv Security Groups (Inbound rules) with the AWS NAT Gateway IP address. Further, if you want to avoid having an EIP on the FTDv Management interface (assuming the FMC has connectivity to the FTDv VPC), you need to make the Lambda Function private to the VPC, so that Lambda Function can access VPC elements (the FTDv instance Management private IP address).

The *asg.yaml* template and the *aws.py* python file in the *autoscale_manager* folder should be modified accordingly to connect the FTDv instances using the private IP itself.

- For *asg.yaml*— In resource *AWS::AutoScaling::LaunchConfiguration*, the parameter *AssociatePublicIpAddress* is set as “true”. This needs to be set as “false”, or the parameter should be removed. Once this is performed, the FTDv instances come up with only private IP addresses.

- For *aws.py*—Line number 62 can be copied to line number 55, by doing so the public IP is also updated with the private IP of the Management interface.

  As of now Python modules are written in such a way that only a public IP address is used. This modification can be done to make it use private IP addresses.