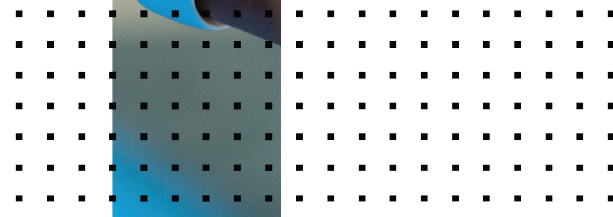
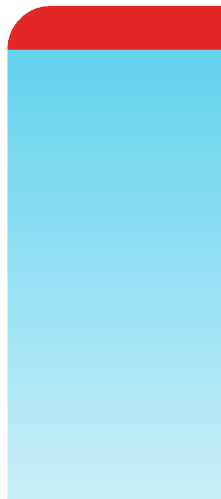


AWS Deployment Guide

FortiDeceptor 6.2.0



FORTINET DOCUMENT LIBRARY

<https://docs.fortinet.com>

FORTINET VIDEO LIBRARY

<https://video.fortinet.com>

FORTINET BLOG

<https://blog.fortinet.com>

CUSTOMER SERVICE & SUPPORT

<https://support.fortinet.com>

FORTINET TRAINING & CERTIFICATION PROGRAM

<https://www.fortinet.com/training-certification>

FORTINET TRAINING INSTITUTE

<https://training.fortinet.com>

FORTIGUARD LABS

<https://www.fortiguard.com>

END USER LICENSE AGREEMENT

<https://www.fortinet.com/doc/legal/EULA.pdf>

FEEDBACK

Email: techdoc@fortinet.com



April 10, 2026

FortiDeceptor 6.2.0 AWS Deployment Guide

00-620-809392-20260410

TABLE OF CONTENTS

Change Log	4
About FortiDeceptor VM on AWS	5
Licensing	5
FortiDeceptor Cloud topology	6
Minimum system requirements	6
AWS Requirements	7
Virtual Machine Model Specifications	7
Preparing FortiDeceptor for deployment	8
Prepare the FortiDeceptor image for AWS	8
Preparing the network in AWS	9
Creating a Virtual Private Cloud (VPC)	10
Creating subnets in the VPC	11
Creating an internet Gateway	13
Creating a route table	13
Associating subnets with a route table	15
Allocating an elastic IP address	16
Create a bucket	18
Import the FortiDeceptor image to AWS AMI	19
Set up the VM Import Service Role	19
Importing the FortiDeceptor image with python script	21
Importing the FortiDeceptor image with AWS EC2 toolkit	22
Importing the uploaded VHD file as snapshot	23
Creating AMI from a snapshot	24
Python script for importing the FortiDeceptor image	26
Check the imported image in AMIs	30
Create an instance with the imported AMI image	30
Connect the Instance with the Serial Console	33
Associate Public IP to instance port1	35
Configure secondary IPs	37
Configuring the FortiDeceptor Manager and AWS Client	40
Configure the client	40
Configuring FortiDeceptor Manager	42
Manage Cloud Clients	43
Configure the deployment network	44
Deploy the decoys	45
Checking for multiple IPs	45
Record the MAC address	45
Configuring decoys on FortiDeceptor manager	46

Change Log

Date	Change Description
2025-10-06	Initial release.
2025-10-23	Updated Configure the client on page 40 , Configure the client on page 40 and Deploy the decoys on page 45 .
2026-02-13	Updated Deploy the decoys on page 45 .
2026-03-18	Updated Create a bucket on page 18 and Import the FortiDeceptor image to AWS AMI on page 19 .
2026-04-10	Updated Minimum system requirements on page 6 .

About FortiDeceptor VM on AWS

FortiDeceptor VM is a 64-bit virtual appliance version of FortiDeceptor. It is deployed in a virtual machine environment. Once the virtual appliance is deployed and set up, you can manage FortiDeceptor VM via its GUI in a web browser on your management computer.

This document provides information about deploying a FortiDeceptor VM in the Amazon Web Services (AWS) environment. This includes how to configure the virtual hardware settings of the virtual appliance. This guide presumes that the reader has a thorough understanding of virtualization servers.

This document does not cover configuration and operation of the virtual appliance after it has been successfully installed and started. For that information, see the [FortiDeceptor Administration Guide](#) in the [Fortinet Document Library](#).

Licensing

Fortinet offers the FortiDeceptor in a stackable license model. This model allows you to expand your VM solution as your environment expands. For information on purchasing a FortiDeceptor license, contact your Fortinet Authorized Reseller, or visit https://www.fortinet.com/how_to_buy/.

When configuring your FortiDeceptor, ensure that you configure hardware settings as outlined in the following table and consider future expansion. Contact your Fortinet Authorized Reseller for more information.

Technical Specification	Details
AWS support	<ul style="list-style-type: none">t3.medium for 2 NICsc5.4xlarge for 6 NICs The available EC2-instance type is determined by the zone.
Virtual CPUs (min / max)	4 / Unlimited*
Virtual Network Interfaces	2-6 NICs
Virtual Memory (min / max)	8GB / Unlimited**
Virtual Storage (min / max)	HDD 50G/ 16TB***

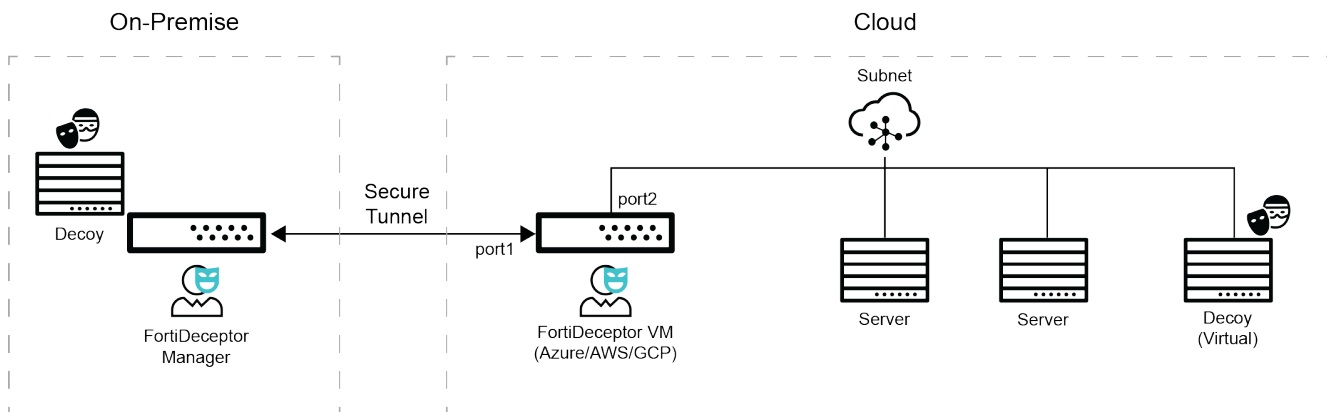
For more information, see the FortiDeceptor product data sheet available on the Fortinet web site, <https://www.fortinet.com/content/dam/fortinet/assets/data-sheets/FortiDeceptor.pdf>.

After placing an order for FortiDeceptor, a license registration code is sent to the email address used in the order form. Use the license registration code provided to register the FortiDeceptor with Customer Service & Support at <https://support.fortinet.com>.

Upon registration, you can download the license file. You will need this file to activate your FortiDeceptor. You can configure basic network settings from the CLI to complete the deployment. Once the license file is uploaded and validated, the CLI and GUI will be fully functional.

FortiDeceptor Cloud topology

The cloud appliance is deployed over the public infrastructure but uses a different method for decoy deployment. This new method requires less HW requirements for the cloud appliance itself.



The cloud decoy deployment method is as follows:

- The cloud appliance will be deployed over the cloud infrastructure.
- An on-premise FortiDeceptor Manager will manage the cloud appliance over a propriety network tunnel.
- The propriety network tunnel allows managing the cloud appliance and decoy deployment provisioning over layer2 tunnel communication over layer3.
- The cloud appliance network interfaces will hold IP addresses in the cloud segment. Each IP address represents a network decoy.
- The network decoy will run on the on-premise FortiDeceptor Manager and use the same IP address as the cloud appliance network interfaces.
- The cloud IP address will tunnel over Layer2 to the IP address on the on-premise FortiDeceptor Manager.
- The idea is to run a light appliance in the cloud while running the actual network decoys inside the on-premise FortiDeceptor Manager in a sandbox mode. The cloud network is isolated from the rest of the decoys, the on-premise networks.

While the cloud appliance uses different hardware requirements, the on-premise FortiDeceptor Manager HW requirements that should serve the cloud appliance decoys is the same concept as today.

Minimum system requirements

The following are the minimum system requirements to deploy decoys with FortiDeceptor for AWS.

AWS Requirements

Technical Specification	Details
AWS support	<ul style="list-style-type: none">t3.medium for 2 NICsc5.4xlarge for 6 NICs The available EC2-instance type is determined by the zone.

Virtual Machine Model Specifications

Model	CPU	Memory	Storage	NIC
VME	4	8 GB	128 GB	2-6
VMS	12	16 GB	500 GB	2-6

Preparing FortiDeceptor for deployment

To prepare FortiDeceptor for deployment, download the FortiDeceptor image from FortiCloud. Prepare the AWS network by creating a Virtual Public Cloud, subnets, an Internet gateway, and route table. After the network is prepared you will need to import an AMI image to create a VM instance, then associate the instance with public IP addresses to deploy the decoys.

To prepare for deployment:

1. Prepare the FortiDeceptor image.
2. Prepare the network in AWS.
3. Create a bucket.
4. Import the FortiDeceptor image to AWS AMI.
5. Check the imported image.
6. Create an instance from the AMI image.
7. Verify the instance.
8. Associate a public IP to port1
9. Configure multiple IPs for deployment.

Prepare the FortiDeceptor image for AWS

Download the image archive file for the AWS platform and unzip it to get image file *fdc.aws.vhd*.

To download the FortiDeceptor image:

1. Log in to [FortiCloud](#).
2. In the banner, click *Support > Downloads > Firmware Download*. The *Download/Firmware Images* page opens.
3. From the *Select Product* dropdown, select *FortiDeceptor*.
4. Click the *Download* tab.
5. In the *Image File Path* section, click the image folder until you reach the image page.

6. Select *FDC_VM-vx.x.x-buildxxxx-FORTINET.out.aws.zip*

Welcome to the Firmware Images download center for Fortinet's extensive line of security solutions.

Select Product

FortiDeceptor

Release Notes Download

Image File Path

/ FortiDeceptor/ v4.00/ 4.1/ 4.1.0/

Image Folders/Files

[Up to higher level directory](#)

Name	Size (KB)	Date Created	Date Modified	
FDC_1000F-v400-build0128-FORTINET.out	200,705	2021-12-16 16:12:30	2021-12-16 16:12:59	HTTPS Checksum
FDC_1000G-v400-build0128-FORTINET.out	200,705	2021-12-16 16:12:37	2021-12-16 16:12:26	HTTPS Checksum
FDC_VM-v400-build0128-FORTINET.out	200,705	2021-12-16 16:12:48	2021-12-16 16:12:29	HTTPS Checksum
FDC_VM-v400-build0128-FORTINET.out.aws.zip	128,782	2021-12-16 16:12:16	2021-12-16 16:12:37	HTTPS Checksum
FDC_VM-v400-build0128-FORTINET.out.azure.zip	128,580	2021-12-16 16:12:23	2021-12-16 16:12:03	HTTPS Checksum
FDC_VM-v400-build0128-FORTINET.out.gcp.tar.gz	128,587	2021-12-16 16:12:29	2021-12-16 16:12:58	HTTPS Checksum
FDC_VM-v400-build0128-FORTINET.out.kvm.zip	127,648	2021-12-16 16:12:59	2021-12-16 16:12:15	HTTPS Checksum
FDC_VM-v400-build0128-FORTINET.out.ovf.esx.zip	127,500	2021-12-16 16:12:17	2021-12-16 16:12:48	HTTPS Checksum
FDC_VM-v400-build0128-FORTINET.out.vmware.zip	127,661	2021-12-16 16:12:51	2021-12-16 16:12:17	HTTPS Checksum

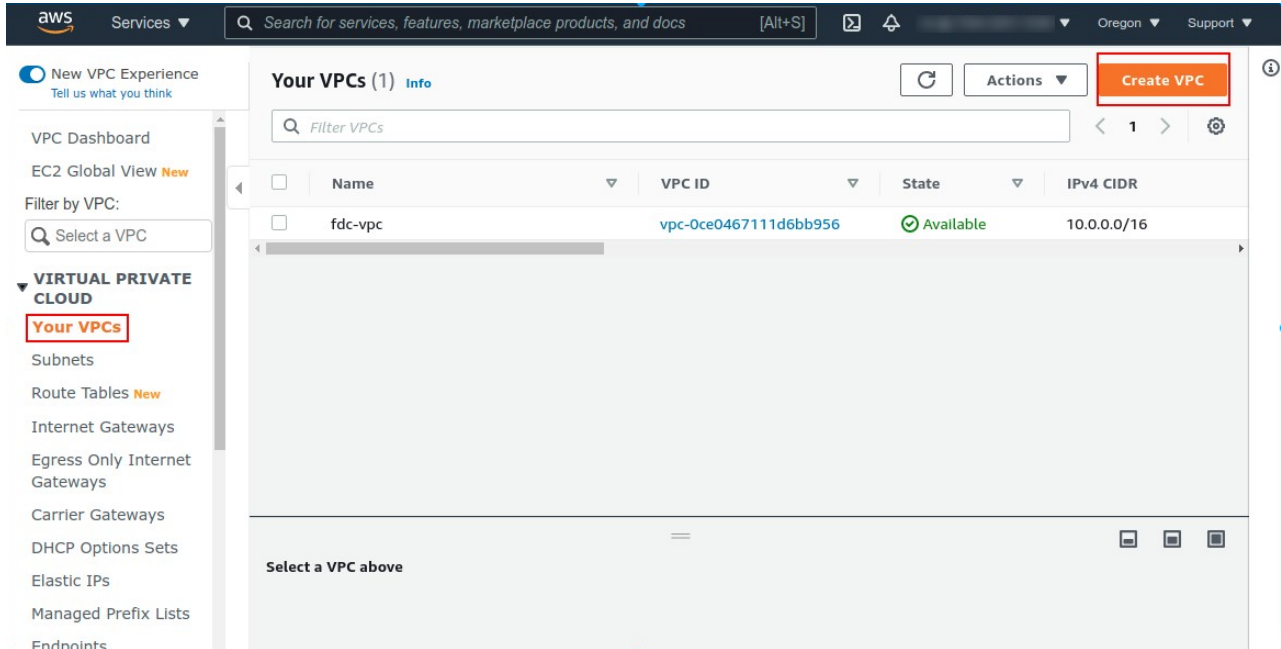
Preparing the network in AWS

To prepare the network, create a Virtual Private Cloud (VPC) and create several subnets. Next you will create an Internet Gateway and route table. Associate the subnets with the route table and then allocate an elastic IP address.

Creating a Virtual Private Cloud (VPC)

To create a VPC in AWS:

1. In the Services menu, go to *Virtual Private Cloud > Your VPCs*.
2. Click *Create VPC*. The *Create VPC* page opens.



3. Configure the following settings:

Name Tag	Enter a name for the VPC such as <i>fdc-vpc</i> .
IPv4 CIDR block	Enter the IP address for the VPC

Create VPC Info

A VPC is an isolated portion of the AWS cloud populated by AWS objects, such as Amazon EC2 instances.

VPC settings

Name tag - optional
Creates a tag with a key of 'Name' and a value that you specify.

IPv4 CIDR block Info

IPv6 CIDR block Info

No IPv6 CIDR block
 Amazon-provided IPv6 CIDR block
 IPv6 CIDR owned by me

Tenancy Info

Tags

A tag is a label that you assign to an AWS resource. Each tag consists of a key and an optional value. You can use tags to search and filter your resources or track your AWS costs.

Key	Value - optional	
<input type="text" value="Name"/>	<input type="text" value="fdc-vpc"/>	<input type="button" value="Remove"/>

You can add 49 more tags.

Creating subnets in the VPC

Create several subnets in VPC for FortiDeceptor management and deployment.

To create subnets in the VPC:

1. In the *Services* menu, go to *Virtual Private Cloud > Subnets*.
2. Click *Create subnet*. The *Create subnet* page opens.

3. Configure the following settings:

VPC ID	Select an ID from the dropdown.
Subnet name	Enter a name for the subnet such as <i>fdc-mgmnet</i> .
IPv4 CIDR block	Enter the IP address for the network.

Create subnet [Info](#)

VPC

VPC ID
Create subnets in this VPC.

vpc-0ce0467111d6bb956 (fdc-vpc) ▼

Associated VPC CIDRs

IPv4 CIDRs
10.0.0.0/16

Subnet settings

Specify the CIDR blocks and Availability Zone for the subnet.

Subnet 1 of 1

Subnet name
Create a tag with a key of 'Name' and a value that you specify.

fdc-mgmnet

The name can be up to 256 characters long.

Availability Zone [Info](#)
Choose the zone in which your subnet will reside, or let Amazon choose one for you.

No preference ▼

IPv4 CIDR block [Info](#)

10.0.1.0/24 ✕

▼ **Tags - optional**

Creating an internet Gateway

To create an internet Gateway:

1. In the *Services* menu, go to *Virtual Private Cloud > Internet Gateways*.
2. Click *Create Internet Gateway*. The *Create Internet Gateway* page opens.
3. In the *Name tag* field, enter a name for the tag such as *fdc-publicaccess-gw*.

VPC > Internet gateways > Create internet gateway

Create internet gateway [Info](#)

An internet gateway is a virtual router that connects a VPC to the internet. To create a new internet gateway specify the name for the gateway below.

Internet gateway settings

Name tag
Creates a tag with a key of 'Name' and a value that you specify.

Tags - optional

A tag is a label that you assign to an AWS resource. Each tag consists of a key and an optional value. You can use tags to search and filter your resources or track your AWS costs.

Key	Value - optional	
<input type="text" value="Name"/>	<input type="text" value="fdc-publicaccess-gw"/>	<input type="button" value="Remove"/>

You can add 49 more tags.

4. Click *Create Internet Gateway*.

Creating a route table

To create a route table:

1. In the *Services* menu, go to *Virtual Private Cloud > Route Tables*.
2. Click *Create route table*. The *Create route table* page opens.

3. In the *Name* field, enter a name for the table such as *fdcvpc-default-route*.

VPC > Route tables > Create route table

Create route table [Info](#)

A route table specifies how packets are forwarded between the subnets within your VPC, the internet, and your VPN connection.

Route table settings

Name - optional
Create a tag with a key of 'Name' and a value that you specify.

VPC
The VPC to use for this route table.

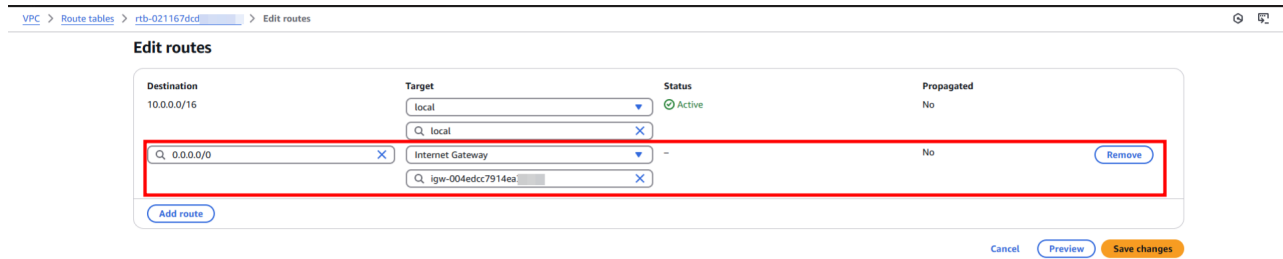
Tags

A tag is a label that you assign to an AWS resource. Each tag consists of a key and an optional value. You can use tags to search and filter your resources or track your AWS costs.

Key	Value - optional	
Q Name X	Q fdcvpc-default-route X	Remove
<input type="button" value="Add new tag"/>		
You can add 49 more tags.		

4. Click *Create route table*.
5. Open the route table you created to edit it.
6. Under *Edit routes*, configure the following settings:

Destination	Enter 0.0.0.0/0.
Target	Enter the Internet gateway you created.



7. Click *Save changes*.

Associating subnets with a route table

Associate a subnet with the route table to apply route rules to that specific subnet.

To associate subnets with Route Table:

1. In the *Services* menu, go to *Virtual Private Cloud > Subnets*.
2. Click the subnet you created. The *Edit route table association* page opens.

3. In the *Route table ID* field, select the route table you just created.

VPC > Subnets > subnet-006d45750a48dba2f > Edit route table association

Edit route table association Info

Subnet route table settings

Subnet ID
📄 subnet-006d45750a48dba2f

Route table ID
rtb-0c0026d0a29fba43f (fdc-default-route) ▼ ↻

📘 You can now check network connectivity with Reachability Analyzer Run Reachability Analyzer ✕

Routes (2)

🔍 < 1 > ⚙️

Destination	Target
10.0.0.0/16	local
0.0.0.0/0	igw-0a81c150d772962e4

Cancel Save

4. Click Save.

Allocating an elastic IP address

Allocate a public IP for public access to FortiDeceptor management port later. This step is not required for deployment.

To allocate an elastic IP address:

1. In the *Services* menu, go to *Virtual Private Cloud > Elastic IPs*.
2. Select an elastic IP. The *Elastic IP address settings* window opens.

3. Click *Allocate*.

Allocate Elastic IP address [Info](#)

Elastic IP address settings [Info](#)

Network Border Group [Info](#)

Public IPv4 address pool

- Amazon's pool of IPv4 addresses
- Public IPv4 address that you bring to your AWS account (option disabled because no pools found) [Learn more](#)
- Customer owned pool of IPv4 addresses (option disabled because no customer owned pools found) [Learn more](#)

Global static IP addresses

AWS Global Accelerator can provide global static IP addresses that are announced worldwide using anycast from AWS edge locations. This can help improve the availability and latency for your user traffic by using the Amazon global network. [Learn more](#)

[Create accelerator](#)

Tags - optional

A tag is a label that you assign to an AWS resource. Each tag consists of a key and an optional value. You can use tags to search and filter your resources or track your AWS costs.

No tags associated with the resource.

[Add new tag](#)

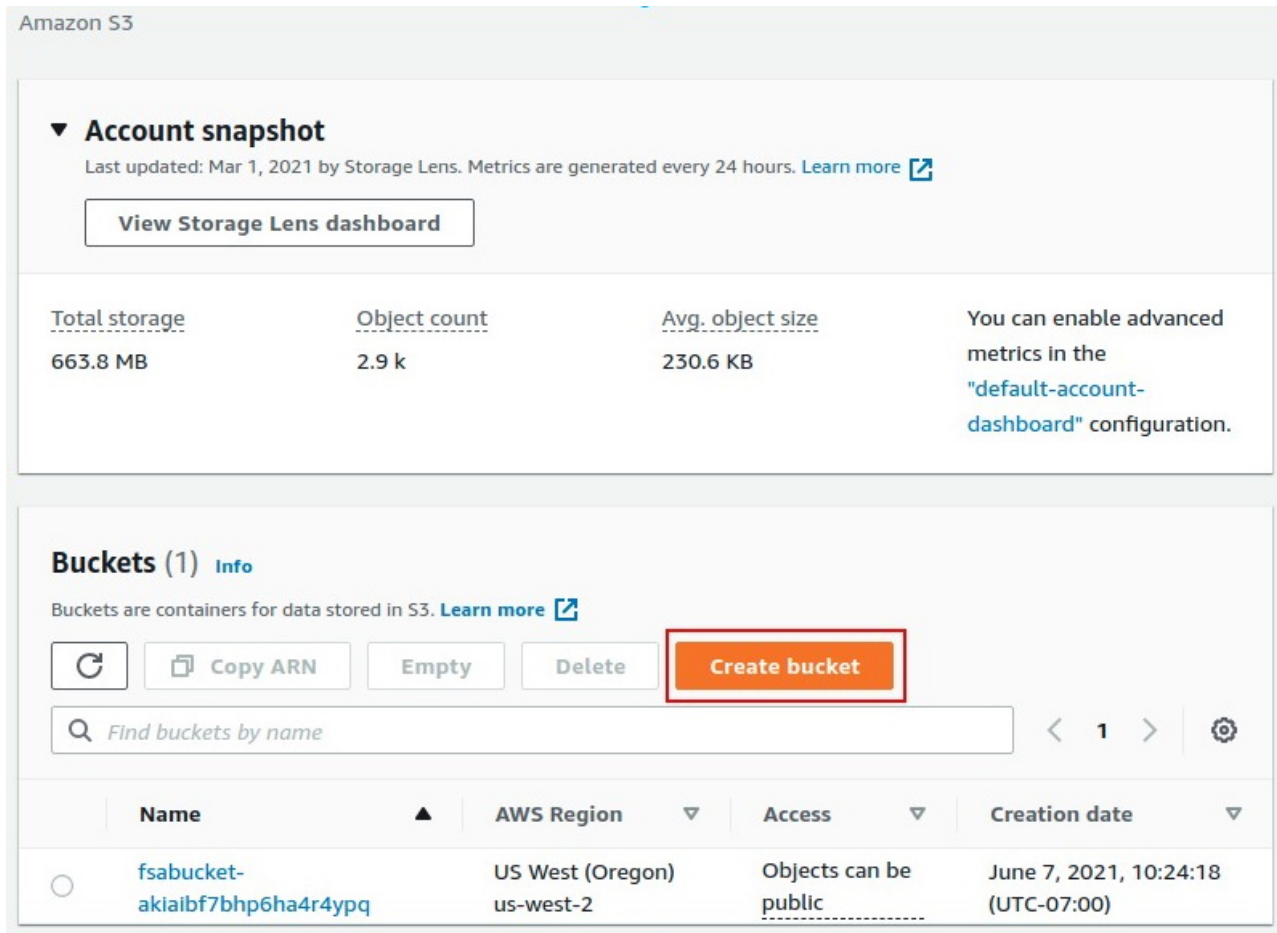
You can add up to 50 more tag

Cancel [Allocate](#)

Create a bucket

To create an AWS bucket:

1. In the AWS Management Console, click *Create Bucket*. The *Create bucket* wizard opens.
2. Configure the bucket settings and click *Create bucket*.



3. Configure the bucket policy.
 - Add a bucket policy to the S3 bucket so the AWS VM Import service can access the FortiDeceptor image file.
 - a. In the S3 console, select your bucket and go to the *Permissions* tab.
 - b. Under *Bucket policy*, click *Edit* and paste the following JSON (replace *YOUR_ACCOUNT_ID* and *BUCKET-NAME* with your actual values):

```
{
  "Version": "2012-10-17",
  "Statement": [
    {

```

```

        "Effect": "Allow",
        "Principal": {
"AWS": "arn:aws:iam::YOUR_ACCOUNT_ID:role/vmimport"
        },
        "Action": "s3:GetObject",
        "Resource": "arn:aws:s3:::BUCKET-NAME/*"
    }
]
}

```

Import the FortiDeceptor image to AWS AMI

Go to IAM Service and create users and roles with proper permissions. Then get the *Access Key ID/Secret Key* from the *My Security Credentials* menu. You can only get the Secret Key at the time you create the Access Key.

Set up the VM Import Service Role

Before importing the image, create an IAM role named *vmimport*. AWS uses this role to convert the disk image to an AMI.

Step 1: Create the Trust Relationship

This step allows the AWS VM Import service to *assume* the role to perform tasks in your account.

Option	Description
CLI	Save the following JSON as <code>trust-policy.json</code> and run: <pre>aws iam create-role --role-name vmimport --assume-role-policy-document file://trust-policy.json</pre>
GUI	In the IAM Console, go to <i>Roles > Create Role</i> . Select <i>Custom trust policy</i> and paste the JSON below: <pre> { "Version": "2012-10-17", "Statement": [{ "Effect": "Allow", "Principal": {"Service": "vmie.amazonaws.com"}, "Action": "sts:AssumeRole", "Condition": { "StringEquals": { "sts:Externalid": "vmimport" } } }] } </pre>

Option	Description
	<pre> }] } </pre>

Step 2: Attach the Permissions Policy

Grant the *vmimport* role permission to access your S3 bucket and create the EC2 AMI.

Option	Description
CLI	<p>Save the following JSON as <i>role-policy.json</i> and run:</p> <pre>aws iam put-role-policy --role-name vmimport --policy-name vmimport --policy-document file://role-policy.json</pre>
GUI	<ol style="list-style-type: none"> 1. Select the <i>vmimport</i> role you just created. 2. Under the <i>Permissions</i> tab, click <i>Add permissions > Create inline policy</i>. 3. Click the <i>JSON</i> tab and paste the following (replace <i>BUCKET-NAME</i> with your actual values): <pre> { "Version": "2012-10-17", "Statement": [{ "Action": ["s3:GetBucketLocation", "s3:GetObject", "s3:ListBucket"], "Effect": "Allow", "Resource": ["arn:aws:s3:::BUCKET-NAME",] }, { "Action": ["ec2:ModifySnapshotAttribute", "ec2:CopySnapshot", "ec2:RegisterImage", "ec2:Describe*"], "Effect": "Allow", "Resource": "*" }] } </pre>

Step 3: Operational User Permissions

The IAM user responsible for running the import commands and launching the instance must have the required permissions. Make sure the appropriate policy is attached to that user or group.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "ec2:RunInstances",
        "ec2:StartInstances",
        "ec2:StopInstances",
        "ec2:Describe*",
        "ec2:ImportSnapshot",
        "ec2:RegisterImage",
        "ec2:CreateNetworkInterface",
        "ec2:AttachNetworkInterface",
        "ec2:AssignPrivateIpAddresses",
        "ec2:AssociateAddress",
        "ec2:CreateTags"
      ],
      "Resource": "*"
    }
  ]
}
```

For detailed instructions on creating the vmimport role required for importing and exporting VM images, see: https://docs.aws.amazon.com/vm-import/latest/userguide/vmie_prereqs.html#vmimport-role

You can import the FortiDeceptor one of two ways:

- [With a python script](#)
- [With the AWS EC2 toolkit](#) (Recommended)

Importing the FortiDeceptor image with python script

Install Python3, boto3 in Linux, and copy the import script to any work folder. Execute the script to import the FortiDeceptor image into AWS as AMI private image.

To get a copy of the Python script, see [Python script for importing the FortiDeceptor image on page 26](#).



Before you begin, make sure you have copied the *fdc.aws.vhd* file to the current directory. To get a copy of the file, see [Python script for importing the FortiDeceptor image on page 26](#).

Example command:

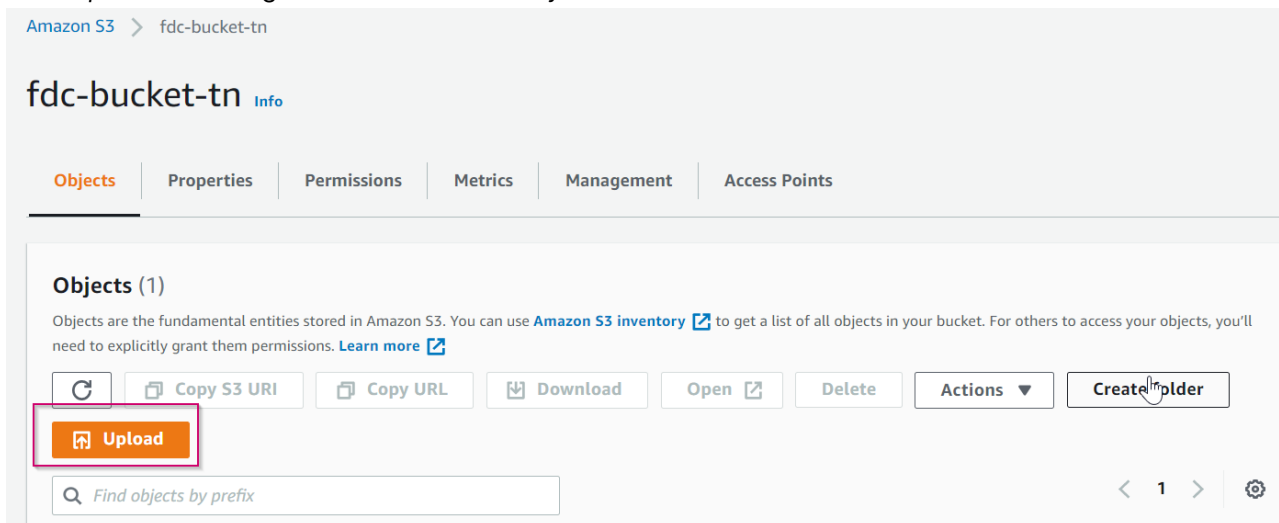
```
python3 FDC_import_as_AWS_AMI.py \
-f /fdc.aws.vhd \
```

```
-n fdcv4.1.0b0090 \  
-a x86_64 \  
-s 1 \  
-r us-west-2 \  
-i AKIA2UEJLWR3DIUPLLF8 \  
-k Uj8Q08TKpgHX5krbR88GkWwnQm2Ko4k14cpUhk99 \  
-b fdcbucket-akiaibf7bhp6ha4r4ypq
```

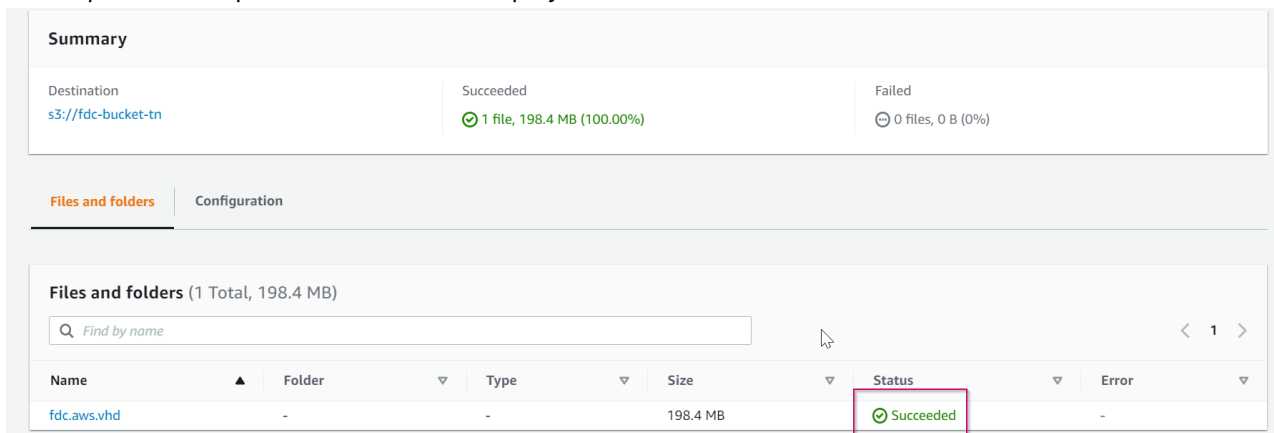
Importing the FortiDeceptor image with AWS EC2 toolkit

To upload the image to a storage bucket:

1. Install the AWS CLI.
2. In the *Buckets* list, open the bucket you created in the previous step.
3. Click *Upload* and navigate to `fdc.aws.vhd` on your device.



4. Click *Upload*. The upload *Status* should display *Succeeded*.



Importing the uploaded VHD file as snapshot

Use the `import-snapshot` command to import a disk.

To import a disk:

1. Run `import-snapshot --description "My FDC VM" --disk-container`.

```
aws ec2 import-snapshot --description "My FDC VM" --disk-container
file://C:\private\aws\containers.json
```

Specify the URL of the S3 bucket, or provide the S3 bucket name and key.

```
{
  "Description": "My FDC VHD",
  "Format": "VHD",
  "UserBucket": {
    "S3Bucket": "fdc-bucket-tn",
    "S3Key": "fdc.aws.vhd"
  }
}
```

The following image shows the response of above command. The status shown is `active`, which means that the import is in progress.

```
C:\Users\nhou>aws ec2 import-snapshot --description "My FDC VM" --disk-container "file://C:\private\aws\containers.json"
{
  "Description": "My FDC VM",
  "ImportTaskId": "import-snap-0aba8b9978bedc8d9",
  "SnapshotTaskDetail": {
    "Description": "My FDC VM",
    "DiskImageSize": 0.0,
    "Progress": "0",
    "Status": "active",
    "StatusMessage": "pending",
    "UserBucket": {
      "S3Bucket": "fdc-bucket-tn",
      "S3Key": "fdc.aws.vhd"
    }
  },
  "Tags": []
}
```

2. Use the `describe-import-snapshot-tasks` command to check the status of an import snapshot task.

```
aws ec2 describe-import-snapshot-tasks --import-task-ids import-snap-
0aba8b9978bedc8d9
```

The snapshot is ready to use when the status is `complete`.

```
C:\Users\nhou>aws ec2 describe-import-snapshot-tasks --import-task-ids import-snap-0aba8b9978bedc8d9
{
  "ImportSnapshotTasks": [
    {
      "Description": "My FDC VM",
      "ImportTaskId": "import-snap-0aba8b9978bedc8d9",
      "SnapshotTaskDetail": {
        "Description": "My FDC VM",
        "DiskImageSize": 208028160.0,
        "Format": "VHD",
        "SnapshotId": "snap-083a9220a5876cf77",
        "Status": "completed",
        "UserBucket": {
          "S3Bucket": "fdc-bucket-tn",
          "S3Key": "fdc.aws.vhd"
        }
      },
      "Tags": []
    }
  ]
}
```

Creating AMI from a snapshot

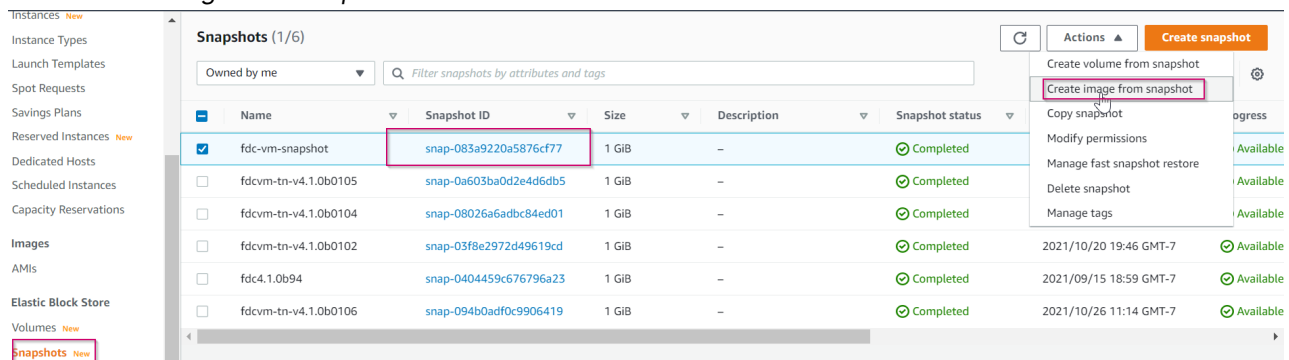
You can create an AMI with either the CLI or the AWS Management Console.

To create the AMI with the CLI:

```
aws ec2 register-image --name fdc-img-cm --architecture x86_64 --root-device-name
/dev/sda1 --virtualization-type hvm --ena-support --block-device-mappings
DeviceName=/dev/sda1,Ebs={SnapshotId=snap-
083a9220a5876cf77,VolumeSize=1,VolumeType=gp2,DeleteOnTermination=true}
DeviceName=/dev/sdb,Ebs={VolumeType=gp2,VolumeSize=80,DeleteOnTermination=true}
```

To create the AMI with the AWS web console:

1. Choose *Snapshots* in the navigation pane of EC2.
2. Select the snapshot you imported.
3. Click *Create image from snapshot* in the *Actions* menu.



4. Enter the *Image Name* and *Description*.

EC2 > Snapshots > snap-083a9220a5876cf77 > Create image from snapshot

Create image from snapshot [Info](#)

Create a new image from a snapshot taken from the root device volume of an instance.

Image settings

Snapshot ID
[snap-083a9220a5876cf77 \(fdc-vm-snapshot\)](#)

Image name
A descriptive name for the image.

3 - 128 characters. Valid characters are a-z, A-Z, 0-9, spaces, and - _ . / () [] ' @.

Description
A description for the image.

255 characters maximum

- Configure the *Block device mappings information*, and click *Create Image*.

Block device mappings - optional [Info](#)

i Provisioned IOPS SSD (io2) volumes with a size greater than 16 TiB, IOPS greater than 64,000, or IOPS:GiB ratio greater than 500:1 are supported with R5b instances only.

▼ Volume 1

Device type	Device name	Snapshot
Root	/dev/sda1	snap-083a9220a5876cf77
Size (GiB)	Volume type	IOPS
<input type="text" value="1"/>	General Purpose SSD (gp2) ▼	100 / 3000
Throughput (MB/s)	Termination behavior	Encryption
-	<input checked="" type="checkbox"/> Delete on termination	<input type="checkbox"/> Encrypt volume

▼ Volume 2 Remove volume

Device type	Device name	Snapshot
EBS ▼	/dev/sdb ▼	Use default ▼
Size (GiB)	Volume type	IOPS
<input type="text" value="50"/>	General Purpose SSD (gp2) ▼	150 / 3000
Throughput (MB/s)	Termination behavior	Encryption
-	<input checked="" type="checkbox"/> Delete on termination	<input type="checkbox"/> Encrypt volume

Cancel
Create image

Python script for importing the FortiDeceptor image

To view the help message for the for this script use the command -h.

```
import boto3
import time, sys, os, traceback
import json
import pprint
from datetime import datetime
```

```

from types import SimpleNamespace

global_region_name="us-west-2"
global_aws_access_key_id=""
global_aws_secret_access_key=""
global_bucket=""

class DatetimeEncoder(json.JSONEncoder):
    def default(self, obj):
        if isinstance(obj, datetime):
            return obj.strftime('%Y-%m-%dT%H:%M:%SZ')
        elif isinstance(obj, date):
            return obj.strftime('%Y-%m-%d')
        # Let the base class default method raise the TypeError
        return json.JSONEncoder.default(self, obj)

def check_return(resp):
    if resp != None:
        if resp['ResponseMetadata']['HTTPStatusCode'] == 200:
            return 0
    return -1

def list_bucket():
    bna = []
    for bucket in s3.buckets.all():
        bna.append(bucket.name)
    return bna

def resp2obj(resp):
    s = json.dumps(resp, cls=DatetimeEncoder)
    return json.loads(s, object_hook=lambda d: SimpleNamespace(**d))

def bucket_exists(s3s, fk):
    for b in s3s.buckets.all():
        if b.name == fk:
            return True
    return False

def import_as_AMI(filename, imagename, arch, size):
    if filename is None:
        print("Incorrect parameter")
        return

    fn = filename #sys.argv[1]
    fk = imagename #sys.argv[2]
    arch = arch #sys.argv[3]
    size = size #sys.argv[4]
    s3s = boto3.resource('s3', region_name=global_region_name, aws_access_key_id=global_aws_
access_key_id, aws_secret_access_key=global_aws_secret_access_key)
    s3c = boto3.client('s3', region_name=global_region_name, aws_access_key_id=global_aws_access_
key_id, aws_secret_access_key=global_aws_secret_access_key)
    buck=global_bucket

```

```

if not bucket_exists(s3s, buck):
    bucket = s3s.create_bucket(ACL='private', Bucket=buck, CreateBucketConfiguration=
{'LocationConstraint':global_region_name})
    if bucket != None:
        bucket.wait_until_exists()
    else:
        print("Failed to create bucket %s" % (buck))
        return
else:
    bucket = s3s.Bucket(buck)
    bucket = s3s.Bucket(buck)
    s3c.delete_object(Bucket=buck, Key=fk)
    bucket.upload_file(fn, fk)
    ec2 = boto3.client('ec2', region_name=global_region_name, aws_access_key_id=global_aws_access_
key_id, aws_secret_access_key=global_aws_secret_access_key)
    try:
        resp = ec2.import_snapshot(
            Description='import FDC image snapshot',
            DiskContainer={
                'Format': 'VHD',
                'UserBucket': {
                    'S3Bucket': buck,
                    'S3Key': fk
                }
            })
        r = resp2obj(resp)
    except Exception as e:
        print('''Please make sure you have the service role 'vmimport' with below permissions:
-- Resource to s3:your-bucket
*) s3:ListBucket
*) s3:GetBucketLocation
*) s3:GetObject
-- Resource to ec2:*
*) ec2:ModifySnapshotAttribute
*) ec2:CopySnapshot
*) ec2:RegisterImage
*) ec2:Describe*

        For more information, please refer to https://docs.aws.amazon.com/vmimport/latest/userguide/vmie\_prereqs.html , section 'Required service role'
        ''')
        print(traceback.format_exc())
        sys.exit(-1)

print("Importing image: taskid={}".format(r.ImportTaskId))
while True:
    time.sleep(10)
    resp = ec2.describe_import_snapshot_tasks(ImportTaskIds=[r.ImportTaskId])
    #print(resp)
    if check_return(resp) == 0:
        taskdetail = resp['ImportSnapshotTasks'][0]
        st = taskdetail['SnapshotTaskDetail']['Status']

```

```

        print("Importing image: {}".format(st))
        if st == 'completed':
            break
        elif st == "deleted":
            print(taskdetail)
            return

    print("Imported image successfully")
    r = resp2obj(resp)
    ec2s = boto3.resource('ec2', region_name=global_region_name, aws_access_key_id=global_aws_
access_key_id, aws_secret_access_key=global_aws_secret_access_key)
    snapshot = ec2s.Snapshot(r.ImportSnapshotTasks[0].SnapshotTaskDetail.SnapshotId)
    snapshot.create_tags(Tags=[{'Key': 'Name', 'Value': fk}])
    resp = ec2.register_image(Name=fk, Architecture=arch, RootDeviceName='/dev/sda1',
        BlockDeviceMappings=[{'DeviceName': '/dev/sda1',
            'Ebs':
{'SnapshotId': snapshot.id, 'VolumeType': 'gp2', 'VolumeSize': int(size), 'DeleteOnTermination': True}},
            {'DeviceName': '/dev/sdb',
            'Ebs':
{'VolumeType': 'gp2', 'VolumeSize': 50, 'DeleteOnTermination': True}},],
        VirtualizationType='hvm', EnaSupport=True)
    if check_return(resp) == 0:
        print("Registered image successfully")
    else:
        print("Failed to register image")
        print(resp)
    r = resp2obj(resp)
    image = ec2s.Image(r.ImageId)
    image.create_tags(Tags=[{'Key': 'Name', 'Value': fk}])
    s3c.delete_object(Bucket=buck, Key=fk)
    print("Deleted the image file from bucket {}".format(buck))

if __name__ == "__main__":
    import argparse
    parser = argparse.ArgumentParser()

    parser.add_argument("-r", "--region_name", help="region_name")
    parser.add_argument("-i", "--aws_access_key_id", help="aws_access_key_id")
    parser.add_argument("-k", "--aws_secret_access_key", help="aws_secret_access_key")
    parser.add_argument("-b", "--bucket", help="The bucket name")
    parser.add_argument("-f", "--filename", help="The FDC AWS vhd full file name")
    parser.add_argument("-n", "--imagename", help="The AMI image name on AWS")
    parser.add_argument("-a", "--arch", help="Optional: default is 86_64")
    parser.add_argument("-s", "--size", help="Optional: The size of the image file, default is
1GB. ")
    args = parser.parse_args()

    global_region_name=args.region_name
    global_aws_access_key_id=args.aws_access_key_id
    global_aws_secret_access_key=args.aws_secret_access_key

```

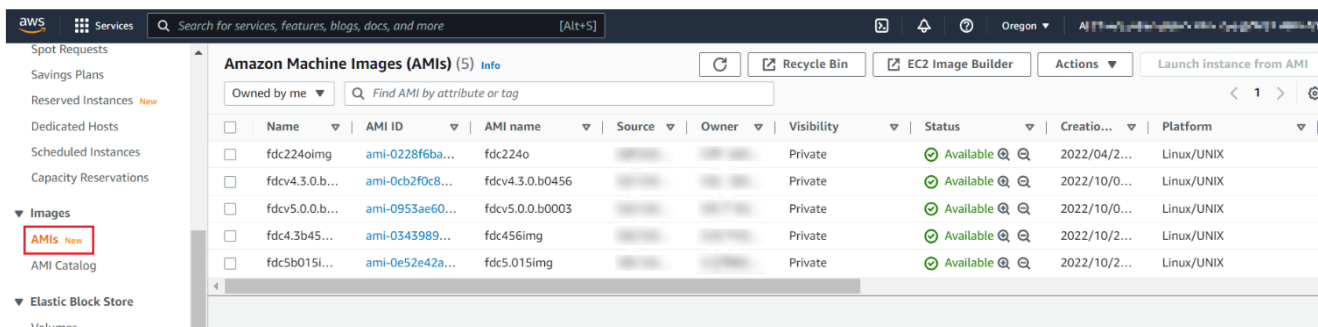
```
global_bucket="fdcbucket".lower()
if args.bucket:
    global_bucket = args.bucket

filename=args.filename
imagename=args.imagename
arch="x86_64"
if args.arch:
    arch = args.arch
size=1
if args.size:
    size=args.size

import_as_AMI(filename, imagename, arch, size)
```

Check the imported image in AMIs

In the AWS console go to *Images > AMIs*. Verify the AMI you uploaded is displayed.

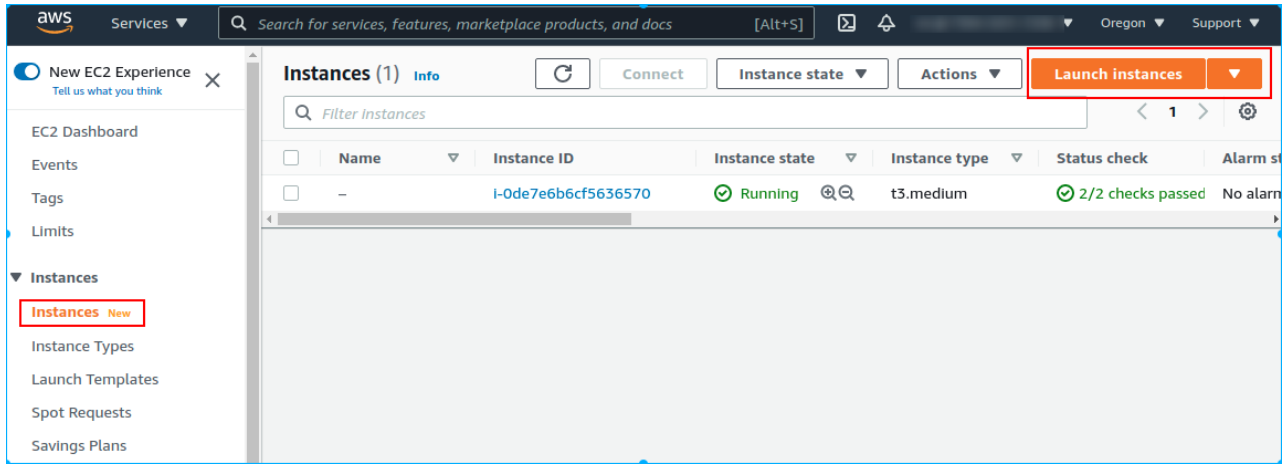


Create an instance with the imported AMI image

The *Instance Wizard* specifies all the launch parameters required for launching an instance. Where the launch instance wizard provides a default value, you can accept the default or specify your own value, like choosing the AMI you created in the last step, configuring your own network interfaces and specifying the security group.

To create an instance with an imported image:

1. In the AWS console go to *Instances > Instances*.
2. Click *Launch Instances*. Instance wizard opens.



3. In Step 1: Choose an Amazon Machine Image (AMI), click *My AMIs* and then select the image you just created, then click *Next*.

EC2 > Instances > Launch an instance

Launch an instance [Info](#)

Amazon EC2 allows you to create virtual machines, or instances, that run on the AWS Cloud. Quickly get started by following the simple steps below.

Name and tags

Name

 [Add additional tags](#)

▼ **Application and OS Images (Amazon Machine Image) [Info](#)**

An AMI is a template that contains the software configuration (operating system, application server, and applications) required to launch your instance. Search or Browse for AMIs if you don't see what you are looking for below

Recents
My AMIs
Quick Start

Owned by me

Shared with me

[Browse more AMIs](#)

Including AMIs from AWS, Marketplace and the Community

Amazon Machine Image (AMI)

fdcimg-
▼

ami-04c6e6ca2ab670e23
2022-04-20T06:44:08.000Z Virtualization: hvm ENA enabled: true Root device type: ebs

4. In *Step 2: Chose an Instance Type*, select the instance type. For more information, see [Elastic network interfaces](#).

▼ **Instance type [Info](#)**

Instance type

c5.4xlarge

Family: c5 16 vCPU 32 GiB Memory

On-Demand Linux pricing: 0.68 USD per Hour

On-Demand Windows pricing: 1.416 USD per Hour

[Compare instance types](#)

5. In *Step 3: Configure Network settings and Security Group*, click *Edit*.
 - a. Select 2 to 6 NICs. You must configure at least two NICs.



Six NICs can be attached to one cloud appliance. Five of the six NICs can be used for decoys.

- b. Click *Add network interface* to add more network interfaces



Make sure ports 22, 443, 8443 are open in FortiDeceptor port1. This allows the FortiDeceptor Manager to communicate with the cloud clients.

6. In *Configure storage*, configure the storage settings and click *Launch Instance*.

▼ **Configure storage** [Info](#) Advanced

1x	1	GiB	gp2 ▼	Root volume	
1x	50	GiB	gp2 ▼	EBS volume	Remove

Connect the Instance with the Serial Console

To connect the instance with the Serial Console:

1. In the AWS Management Console, go to *EC2 > Instances*.
2. Click the instance you created to open it. The *Instance summary for <instance_id>* page opens.
3. Click *Actions > Monitor and troubleshoot > EC2 Serial Console*.

Stay on the Instance Summary page.

The screenshot displays the AWS Management Console interface for an EC2 instance. The breadcrumb navigation at the top reads 'EC2 > Instances > i-0de7e6b6cf5636570'. The main heading is 'Instance summary for i-0de7e6b6cf5636570' with an 'Info' link. Below the heading, it says 'Updated less than a minute ago'. There are four buttons: 'Refresh', 'Connect', 'Instance state' (with a dropdown arrow), and 'Actions' (with an upward arrow). The 'Actions' menu is open, showing options: 'Connect', 'Manage instance state', 'Instance settings', 'Networking', 'Security', 'Image and templates', and 'Monitor and troubleshoot'. The 'Monitor and troubleshoot' sub-menu is also open, listing: 'Get system log', 'Get instance screenshot', 'Manage detailed monitoring', 'Manage CloudWatch alarms', 'EC2 Serial Console' (highlighted with a red box), and 'Replace root volume'. The instance details are as follows:

Instance ID	Public IP	Private IPv4 addresses
i-0de7e6b6cf5636570	35	10.0.2.218 10.0.1.38
IPv6 address	Instance state	Public IPv4 DNS
-	Run	-
Private IPv4 DNS	Instance type	
ip-10-0-1-38.us-west-2.compute.internal	t3.medium	
VPC ID	AWS Compute Optimizer finding	
vpc-0ce0467111d6bb956 (fdc-vpc)	Opt-in to AWS Compute Optimizer for recommendations. Learn more	
Subnet ID		
subnet-006d45750a48dba2f (fdc-mgmnet)		

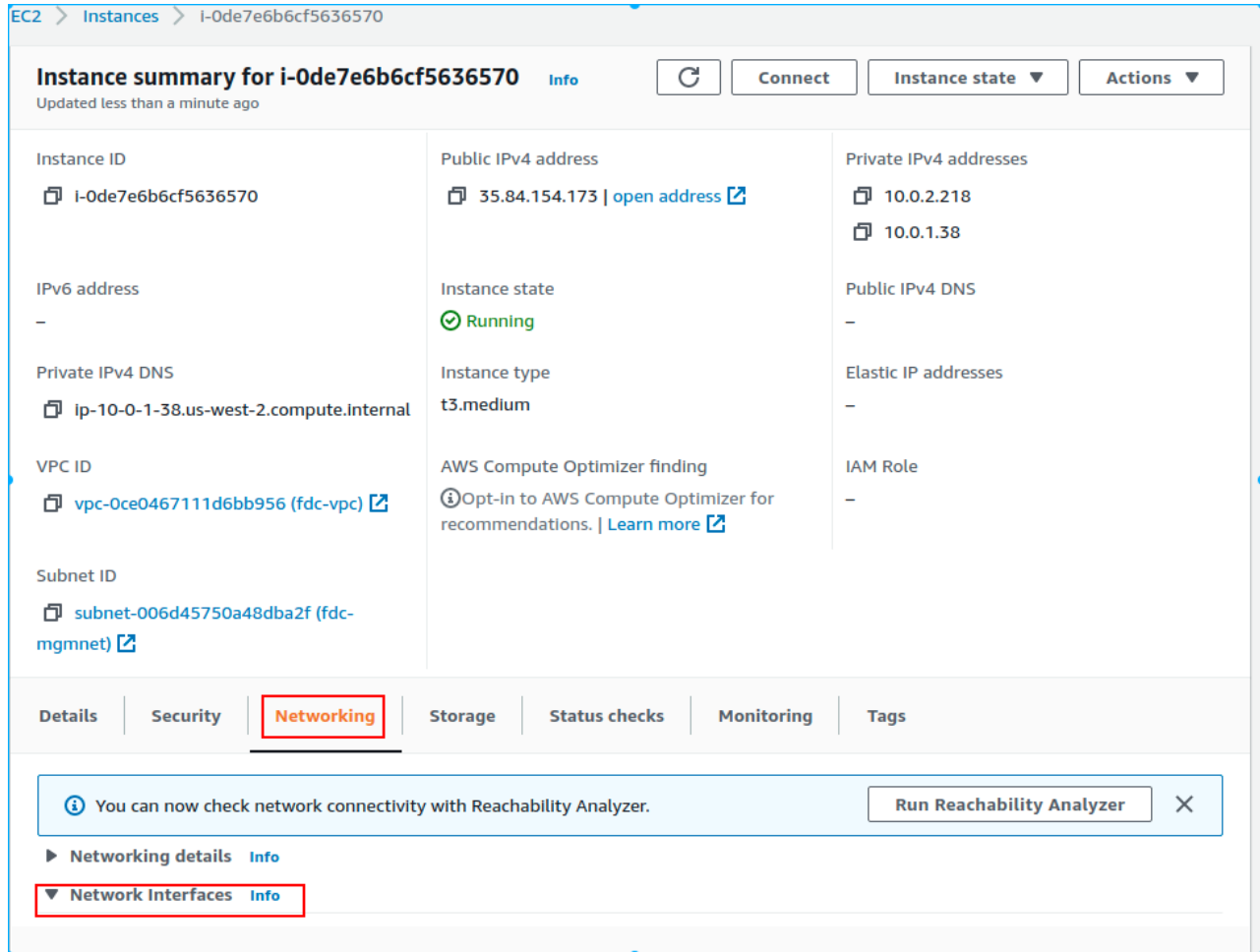
Below the instance details, there are tabs for 'Details', 'Security', 'Networking', 'Storage', 'Status checks', 'Monitoring', and 'Tags'. The 'Details' tab is selected. Under 'Instance details', the following information is shown:

Platform	AMI ID	Monitoring
Linux/UNIX (Inferred)	ami-04d319f5610ee6f66 (fdc4.1.0)	disabled

Associate Public IP to instance port1

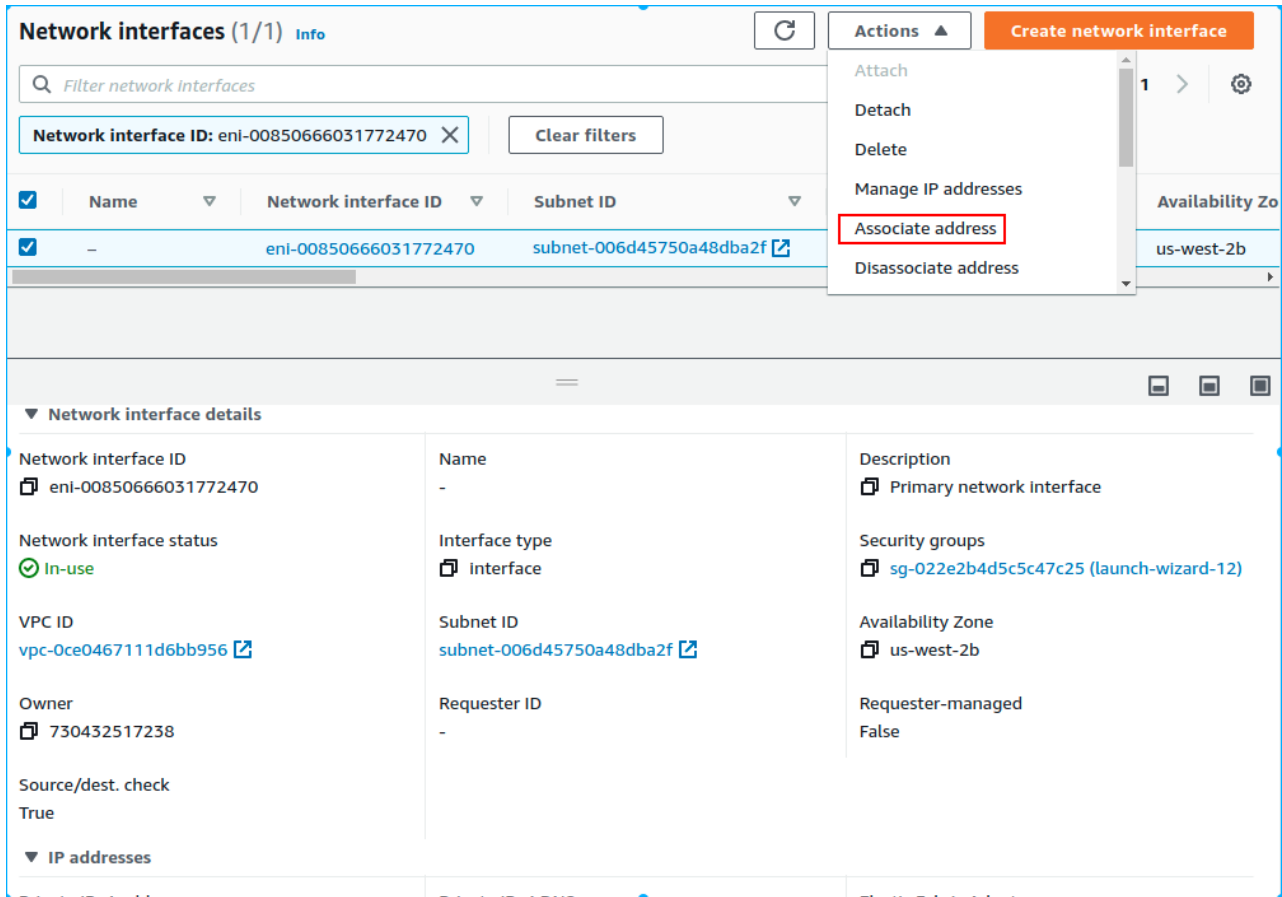
To associate a public IP to an instance:

1. In the *Instance Summary* page, click the *Networking* tab.



2. Expand the *Network interfaces* section.

3. Click *Actions > Associate address*. The *Associate Elastic IP address* page opens.



4. From the *Elastic IP address* dropdown, select the elastic IP you created.

EC2 > Network Interfaces > Associate Elastic IP address

Associate Elastic IP address [Info](#)

Associate an Elastic IP address with one of the private IPv4 addresses for the network interface.

Association details

Network interface
eni-00850666031772470

Elastic IP address
35.84.154.173

Private IPv4 address
Choose a private IPv4 address

Allow reassociation
 Allow the Elastic IP address to be reassociated with this network interface

Cancel Associate

5. Click *Associate*.

Keep the *Network Interfaces* page open.

Configure secondary IPs

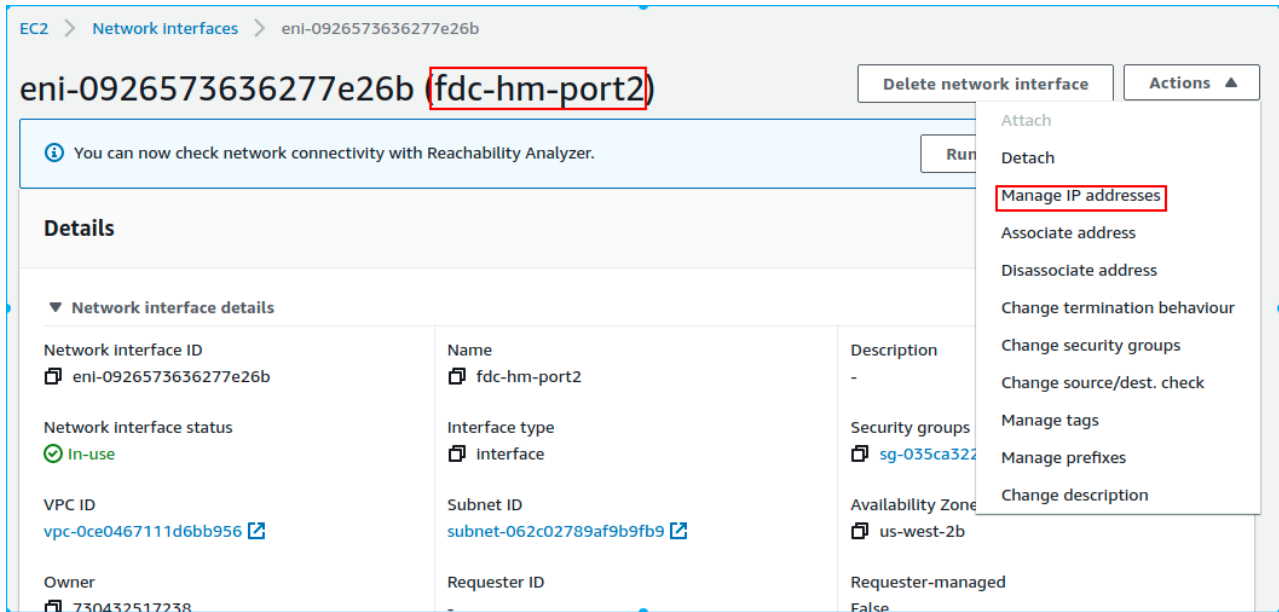
Add more Interfaces and then configure multiple secondary IPs for FDC port2 for decoy deployment. This required when adding more decoy IPs in the future.

To add more interfaces:

1. Go to *EC2 > Instance* and click the *Instance ID*.
2. Go to *Action > Networking > Attach network interface*.

To configure secondary IPs:

1. Click *Actions* > *Manage IP address*. The *Manage IP addresses* page opens.



2. In the *IPv4 addresses* section, click *Assign new IP address*.

EC2 > Network Interfaces > eni-0926573636277e26b > Manage IP addresses

Manage IP addresses [Info](#)

Assign or unassign IPv4 and IPv6 addresses to or from a network interface.

i To assign additional public IPv4 addresses to this network interface, you must **allocate** Elastic IP addresses and associate them with this network interfaces.

▼ eth1: eni-0926573636277e26b - 10.0.2.0/24

IPv4 addresses

Private IP address	Public IP address	
10.0.2.252		<button>Unassign</button>
10.0.2.242		<button>Unassign</button>
10.0.2.51		<button>Unassign</button>
10.0.2.240		<button>Unassign</button>

Assign new IP address

Configuring the FortiDeceptor Manager and AWS Client

After deploying the FortiDeceptor Client, configure both the FortiDeceptor client and the Manager. When configuration is complete, you can deploy decoys and endpoints.



We recommend setting up a security policy and trusted host to ensure the FortiDeceptor is running in a safe environment.

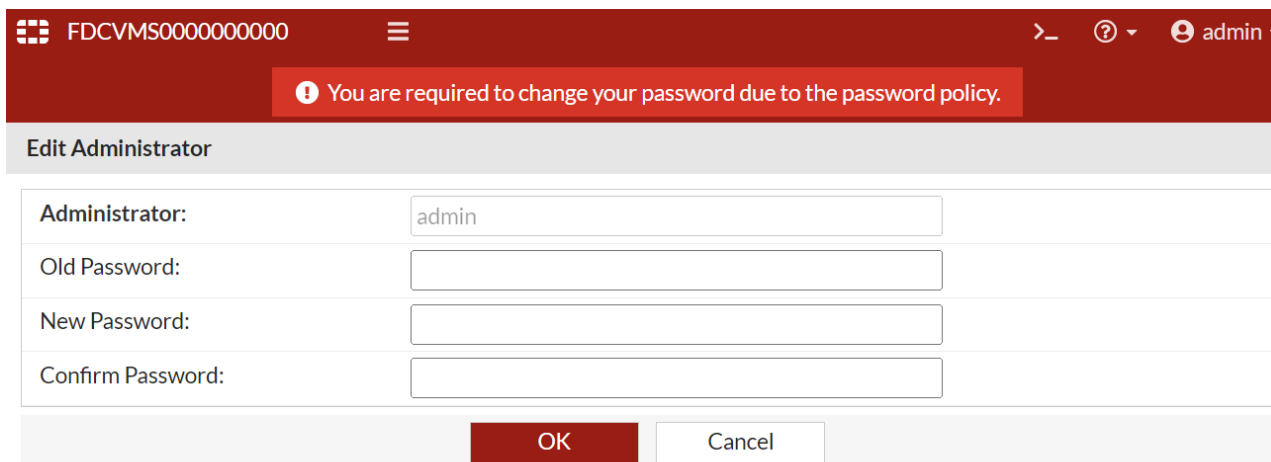
To configure FortiDeceptor and AWS Client:

1. [Configure the client on page 40.](#)
2. [Configure FortiDeceptor manager.](#)
3. [Configure the deployment network.](#)
4. [Deploy the decoys.](#)

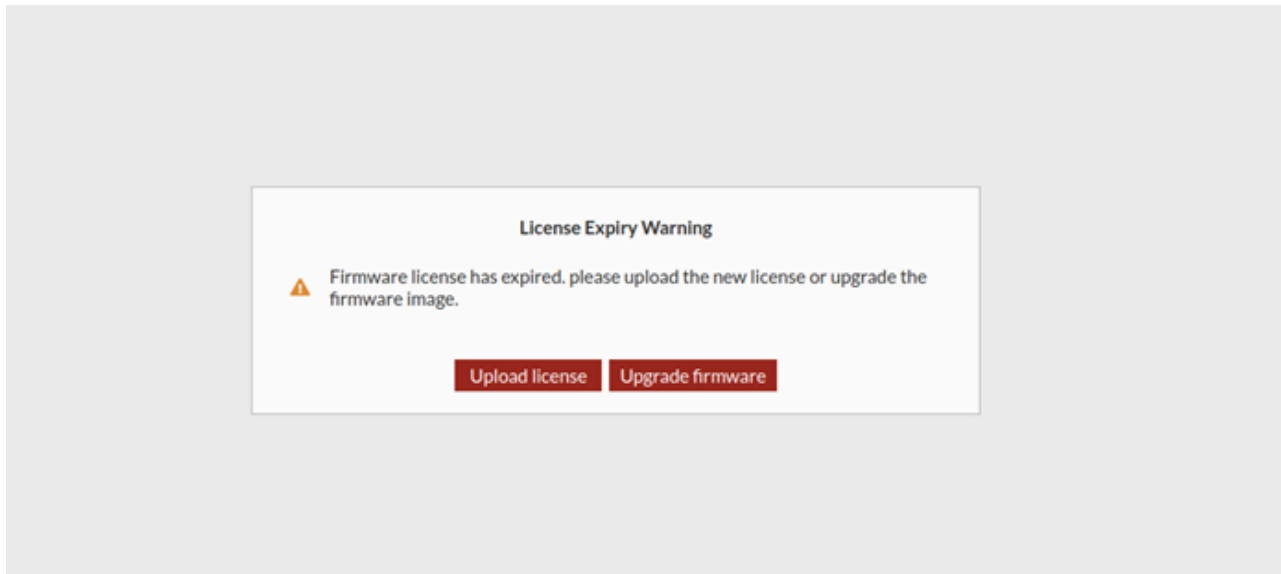
Configure the client

To configure the AWS client:

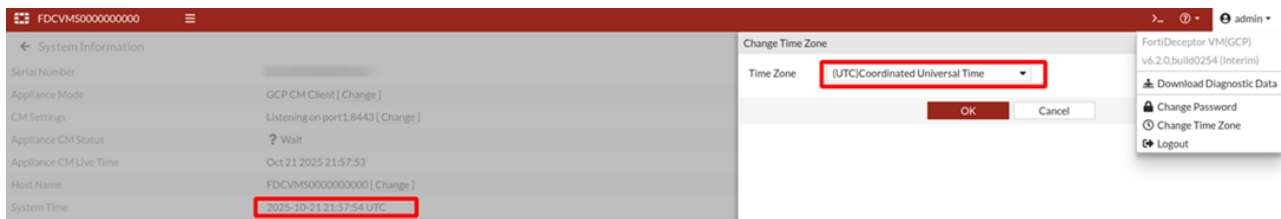
1. Log in to the AWS client with the public IP address. By default, the *admin* user account has no password.
2. After the instance reboots, you are prompted to change the password and log in again.



3. After logging in, the FortiDeceptor instance prompts you to upload the license file. Click *Upload license* to navigate to the file and click *Submit*. After the file submitted, FortiDeceptor will reboot.



4. After logging in, the dashboard shows the system time based on the time zone settings in the Administrator menu.



5. Change the Host Name.
 - a. Go to *Dashboard > System information > Host Name* and click *Change*. The *Edit Host Name* field opens.
 - b. In the *New Name* field, enter a the new Host Name.
6. Configure the client in the CM settings:
 - a. Go to *Dashboard > System Information*. Locate *CM Settings* and click *Change*
 - b. In the CM Settings, select *Wait for connections from manager*.
 - c. Configure the *Listening Interface*, *Port*, and *Encryption Method*. The *Encryption Method* must be the same as the FortiDeceptor Manager
 - d. Click *OK*.

CM Settings

Connection Type * Connect to manager Wait connections from manager

Listening Interface * port1

Port * 8443

Encryption Method * Plaintext
AES128CBC
AES192CBC
AES256CBC

Configuring FortiDeceptor Manager

Connect to the remote CM Client and apply the encryption method chosen earlier to add AWS FortiDeceptor as a cloud appliance.

To configure FortiDeceptor manager:

1. Go to *Dashboard > System Information*. Locate *CM Settings* and click *Change*.
2. In the *CM Settings*, select *Connect to remote CM client*.
3. Configure the *Client IP* (AWS appliance IP), *Port*, and *Proxy Server* if one has been configured on the FortiDeceptor.

Create connection to client ✕

Client IP *

Port * 8443

Proxy Server N/A

Ok Cancel

4. Click *Ok*. If successful the AWS appliance IP will appear in *CM Setting > CM Communication Setting* in the *Client IP* table

- Configure one or more *Supported Encryption Methods*. Make sure the method selected on the client is among those enabled on the manager.

+ Connect to remote CM client
Edit
Delete

	Client IP ↕	Port ↕	Proxy Server ↕	Update Time [PST8PDT] ↕
<input type="checkbox"/>	██████████	8443	N/A	2025/10/13 19:42:50
<input type="checkbox"/>	██████████	8443	N/A	2025/10/13 19:42:50
<input type="checkbox"/>	██████████	8443	N/A	2025/10/13 19:42:50
<input type="checkbox"/>	██████████	8443	N/A	2025/10/13 19:42:50
<input type="checkbox"/>	██████████	8443	N/A	2025/10/13 19:42:50
<input type="checkbox"/>	██████████	8443	N/A	2025/10/13 19:42:50

Supported Encryption Methods Plaintext AES128CBC AES192CBC AES256CBC

Save
Close

- Click *Save*.
- Go to *Central Management > Appliances*, select the AWS appliance and click *Approve*



Delete the previous client and add the client with new public IP once the public IP is changed.

Manage Cloud Clients

To delete a cloud appliances:

- Go to *Dashboard > System Information*. Locate *CM Settings* and click *Change*.
- Select the cloud appliance IP, and click *Delete*. Click *OK* in the confirmation dialog that appears
- Click *Save*.
- Go to *Central Management > Appliances*, then locate the relevant AWS cloud client and click *Delete*.

Configure the deployment network

To configure the deployment network:

1. Go to *Deception > Deployment Network*.
2. Click *Add New Vlan/Subnet*. The *Add New Vlan/Subnet* dialog opens.

The screenshot shows a dialog box titled "Add new VLAN/Subnet". It contains the following configuration fields:

- Name ***: Text input field containing "AWS-Network".
- Appliance ***: Dropdown menu showing "AWS".
- Interface ***: Dropdown menu showing "port2".
- Tagged Interface**: Toggle switch, currently turned off.
- Deploy Monitor ***: Text input field containing "0.0.0.0/24".
- Gateway ***: Text input field containing "0.0.0.0".
- ARP Spoofing Protection**: Toggle switch, currently turned off.
- Tag ***: Text input field containing "any".

At the bottom of the dialog, there are two buttons: "Save" (highlighted in red) and "Cancel".

3. Configure the network settings and click *Save*.

Deploy the decoys

Checking for multiple IPs

To check the multiple IPs on AWS platform:

1. In the AWS Management Console, go to *Network Interfaces*.

Details		
<p>▼ Network interface details</p>		
<p>Network Interface ID eni-0926573636277e26b</p>	<p>Name fdc-hm-port2</p>	<p>Description -</p>
<p>Network Interface status In-use</p>	<p>Interface type interface</p>	<p>Security groups sg-035ca322401654a75 (launch-wizard-2)</p>
<p>VPC ID vpc-0ce0467111d6bb956</p>	<p>Subnet ID subnet-062c02789af9b9fb9</p>	<p>Availability Zone us-west-2b</p>
<p>Owner 730432517238</p>	<p>Requester ID -</p>	<p>Requester-managed False</p>
<p>Source/dest. check True</p>		
<p>▼ IP addresses</p>		
<p>Private IPv4 address 10.0.2.252</p>	<p>Private IPv4 DNS -</p>	<p>Elastic Fabric Adapter False</p>
<p>Public IPv4 address -</p>	<p>Public IPv4 DNS -</p>	<p>IPv6 addresses -</p>
<p>Secondary private IPv4 addresses 10.0.2.242 10.0.2.51 10.0.2.240</p>	<p>Association ID -</p>	<p>Elastic IP address owner -</p>

2. Open the Interface you created and verify the values in *Private IPv4 address* and *Secondary private IPv4 Addresses*.
3. Record the *MAC address*. You will need this address for the network settings when you deploy the decoys.

Record the MAC address

To record the MAC Address:

- In the AWS Management Console, go to *Network Interfaces > MAC Address*.



MAC address behavior differs between deployment methods:

- FortiDeceptor Manager (Public Cloud Appliance Deployment Network):
The MAC address is automatically populated when deploying a decoy.
- FortiDeceptor DaaS Cloud (Public Cloud Deployment Network):
The MAC address is not automatically populated when deploying a decoy.

(Recommended) To prevent potential connectivity issues:

1. Record the MAC address from AWS before deployment.
2. Verify that the MAC address is correctly auto-populated (if applicable), or
3. Manually enter the correct MAC address prior to completing the deployment.

Configuring decoys on FortiDeceptor manager

To choose a cloud appliance in deployment wizard.

1. In FortiDeceptor, go to *Deception Deployment Wizard* and create a new template.
2. In the *Configuration* tab, in *Appliance Name* choose *Local*.

Deployment Wizard

Template → **2 Configuration** → Set Network

Name * AWS ✓

Appliance Name Local x v

Available Deception Oses * posv1 x v

Selected Services * POS-WEB

Automate Lures any x v

⚠ The generated lures contain random user name(s).

Generate lures Clear

3. In the *Set Network* tab, click *Add network for Deployment* and configure the following settings:

Deploy Network	Select the deployment network.
Addressing Mode	Select <i>Static</i> or <i>DHCP</i> (dynamically assigned).

Network Mask	Enter the network mask.
Gateway	Enter the gateway IP address.
MAC Address	Enter the MAC address .
IP Count	Select the number of IP addresses allocated.
Min/Max	Define the range of IP addresses available in the subnet.
IP Ranges	Enter specific IP addresses or ranges reserved for use.



In version 6.2, cloud appliances do not appear under *Appliance Name* in the *Deployment Wizard*. To deploy decoys to cloud appliances, select a *Deploy Network* that is designated as a cloud appliance the deployment network.

Add Network for Deployment
✕

Deploy Network *

✕ ▼ ✓

Addressing Mode *

Static

DHCP

Network Mask *

✓

Gateway *

✓

MAC Address

✓

IP Count *

✓

ℹ Please check our best practice deployment guide.

Min

Max

IP Ranges * (1)

10.10.8.17|

✓

✕ Cancel

✓ Done

4. Click *Done* to deploy the decoy.
5. (Optional) Deploy more decoys.
 - To deploy decoys for different interfaces, repeat [Checking for multiple IPs on page 45](#)
 - To deploy more decoys for the same interface, repeat [Configuring decoys on FortiDeceptor manager on page 46](#).
6. Attack this decoy IP via the endpoint in the cloud and check the incidents as regular deployment.



www.fortinet.com

Copyright © 2026 Fortinet, Inc. All rights reserved. Fortinet®, FortiGate®, FortiCare® and FortiGuard®, and certain other marks are registered trademarks of Fortinet, Inc., and other Fortinet names herein may also be registered and/or common law trademarks of Fortinet. All other product or company names may be trademarks of their respective owners. Performance and other metrics contained herein were attained in internal lab tests under ideal conditions, and actual performance and other results may vary. Network variables, different network environments and other conditions may affect performance results. Nothing herein represents any binding commitment by Fortinet, and Fortinet disclaims all warranties, whether express or implied, except to the extent Fortinet enters a binding written contract, signed by Fortinet's Chief Legal Officer, with a purchaser that expressly warrants that the identified product will perform according to certain expressly-identified performance metrics and, in such event, only the specific performance metrics expressly identified in such binding written contract shall be binding on Fortinet. For absolute clarity, any such warranty will be limited to performance in the same ideal conditions as in Fortinet's internal lab tests. Fortinet disclaims in full any covenants, representations, and guarantees pursuant hereto, whether express or implied. Fortinet reserves the right to change, modify, transfer, or otherwise revise this publication without notice, and the most current version of the publication shall be applicable.