



# Cisco Spaces: Connector AMI

- Launch Connector 3 as an EC2 Instance from AMI , on page 1

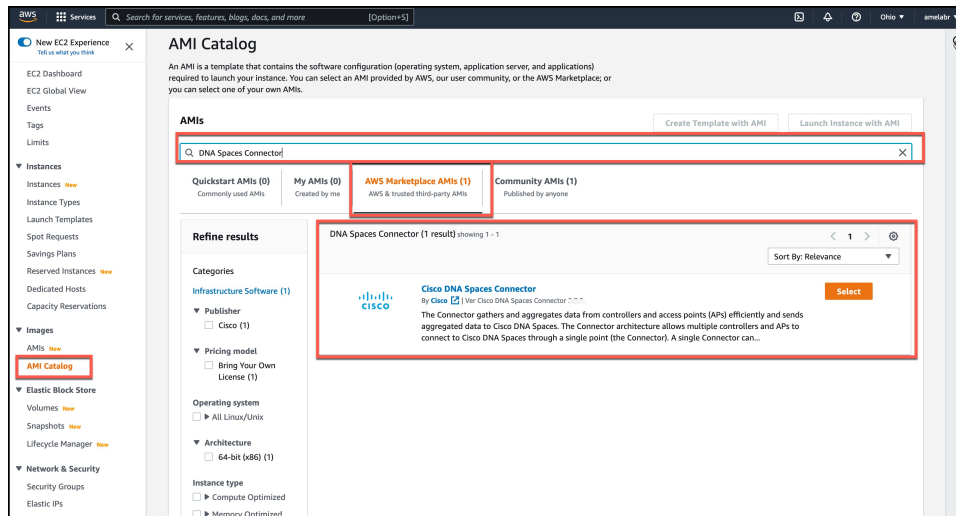
## Launch Connector 3 as an EC2 Instance from AMI

This chapter provides information about how to launch a connector 3 as an EC2 instance from Amazon Machine Images (AMI), configure the connector 3 instance, and finally obtain a URL to log in to the connector connector and CLI.

**Step 1** Log in to your [Amazon Web Services](#) account and navigate to the **EC2 Dashboard**. In the left-navigation pane, choose **Images > AMI Catalog**.

**Step 2** In the AMIs search area, click **AWS Marketplace AMIs** and enter **DNA Spaces Connector**. Press **Enter**.

*Figure 1: Configuration*



**Step 3** Click the displayed image and click **Select**.

**Step 4** In the **Cisco DNA Spaces Connector** window displayed, click **Continue**.

## Launch Connector 3 as an EC2 Instance from AMI

Figure 2: AWS Marketplace AMIs

**Cisco DNA Spaces Connector**  
Cisco Systems, Inc. [View Profile](#)  
0 AWS reviews [View Reviews](#)  
Bring Your Own License

[Overview](#) | [Product details](#) | [Pricing](#) | [Usage](#) | [Support](#)

The Cisco DNA Spaces: Connector enables Cisco DNA Spaces to communicate with multiple controllers efficiently, by allowing each controller to transmit client data without missing any client information

Typical total price  
**\$0.093/Hr**  
Total pricing per instance for services hosted on t2.large in us-east-1.  
[See additional pricing information.](#)

Latest version  
Cisco DNA Spaces Connector3 October2023

Delivery methods  
Amazon Machine Image  [ⓘ](#)

Operating systems  
Other AlmaLinux 8  
CentOS 7

Video  
[Product Video](#)  [ⓘ](#)

Categories  
Network Infrastructure

[Continue](#)

**Step 5** In the **Image Summary** window displayed, click **Launch Instance from AMI**

Figure 3: Launch Instance from AMI

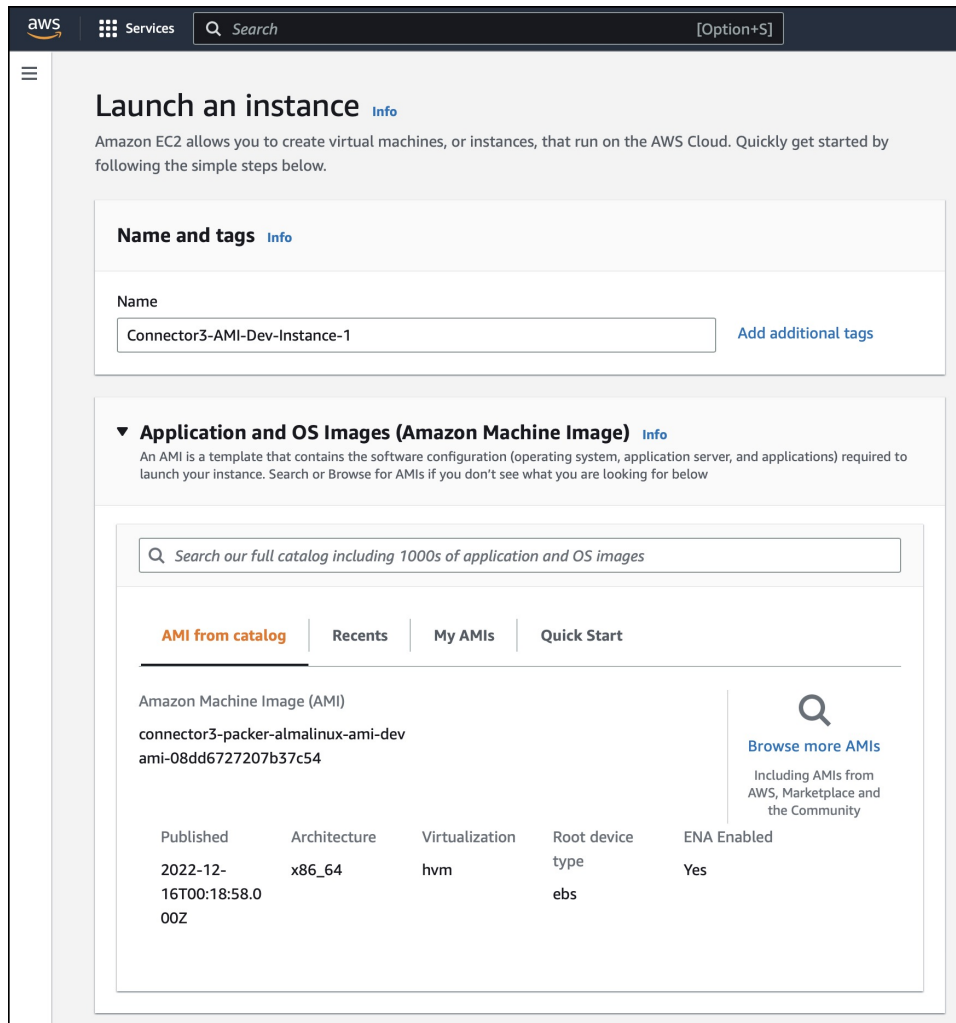
EC2 > AMIs > ami-0fd326aca1b04cf96

**Image summary for ami-0fd326aca1b04cf96 (Connector3-b84-Jan-QA-Img)** [EC2 Image Builder](#) [Actions](#) [Launch Instance from AMI](#)

AMI ID <a href="#">ami-0fd326aca1b04cf96 (Connector3-b84-Jan-QA-Img)</a>	Image type machine	Platform details Linux/UNIX	Root device type EBS
AMI name <a href="#">cisco-dna-spaces-connector3-b84-jan2023-8.4.0-22-DEV</a>	Owner account ID <a href="#">038249548279</a>	Architecture x86_64	Usage operation RunInstances
Root device name <a href="#">/dev/sda1</a>	Status Available	Source <a href="#">038249548279/cisco-dna-spaces-connector3-b84-jan2023-8.4.0-22-DEV</a>	Virtualization type hvm
Boot mode -	State reason -	Creation date <a href="#">Fri Jan 27 2023 12:11:41 GMT-0800 (Pacific Standard Time)</a>	Kernel ID -
Block devices <a href="#">/dev/sda1+snap-00412ac8bc1448df9:15truegg2</a>	Description -	Product codes -	RAM disk ID -
Deprecation time -	Last launched time -		

**Step 6** In the **Launch an Instance** window displayed, enter an instance name, and add any additional labels for your instance by clicking the **Add Additional tags** button.

Figure 4: Launch Instance from AMI



**Step 7** Choose an instance with the corresponding **Type** as **t2.medium** that has **vCPU** value as **2** and **Memory (GB)** as **4**. Click **Next: Configure Instance Details**.

**t2.medium** corresponds to a standard window with 2vCPUs and 4-GB memory and is the recommended setting.

Figure 5: Configure Instance Details

The screenshot shows the AWS console interface for configuring an EC2 instance. The 'Instance type' section is expanded, showing a dropdown menu with 't2.medium' selected. Below the dropdown, it lists 'Family: t2', '2 vCPU', and '4 GiB Memory'. Pricing information is also visible: 'On-Demand Linux pricing: 0.0464 USD per Hour' and 'On-Demand Windows pricing: 0.0644 USD per Hour'. A 'Compare instance types' link is present. The 'Key pair (login)' section is also expanded, showing a dropdown menu with 'connector-ami-test-key' selected. A 'Create new key pair' link is visible next to the dropdown.

**Note** You can have a more advanced configuration by choosing an option with higher vCPU and memory, by choosing an instance type with one of the following configurations. If an exact match is unavailable, you can choose a configuration with the next-available vCPU or memory:

- 4 vCPUs and 8-GB memory (referred to in this document as **Advanced1**)
- 8 vCPUs and 16-GB memory (referred to in this document as **Advanced2**)

**Step 8** Choose a **Network** and a **Subnet**. Click **Next: Add Storage**.

Figure 6: Add Storage

The screenshot shows the AWS console interface for configuring network settings. The 'Network settings' section is expanded, showing a dropdown menu for 'VPC - required' with 'vpc-' selected. Below it, a 'Subnet' dropdown menu is expanded, showing 'subnet-' selected. The 'Subnet info' section displays details: 'VPC: vpc-0...', 'Owner: 199547563901', 'Availability Zone: us-east-1d', 'IP addresses available: 247', and 'CIDR: 1...'. A 'Create new subnet' link is visible next to the dropdown.

**Step 9** Enter the value of **Size(GB)** as 120. Click **Next: Configure Security Group**.

Figure 7: Configure Storage

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

1x  GiB  Root volume (Encrypted)

---

0 x File systems [Edit](#)

**Step 10**

Configure a security group by following these steps:

- a) Create a new security group or modify an existing one by clicking the respective radio button.

Figure 8: Configure Security Group

▼ **Network settings** [Info](#)

Network [Info](#)

vpc:

Subnet [Info](#)

No preference (Default subnet in any availability zone)

Auto-assign public IP [Info](#)

Enable

**Firewall (security groups)** [Info](#)

A security group is a set of firewall rules that control the traffic for your instance. Add rules to allow specific traffic to reach your instance.

Create security group  Select existing security group

Security groups [Info](#)

- b) Configure rules permitting inbound traffic to specific ports, as shown in the following image. You can allow inbound traffic to these ports for all IP addresses or choose to restrict them for specific IP addresses.

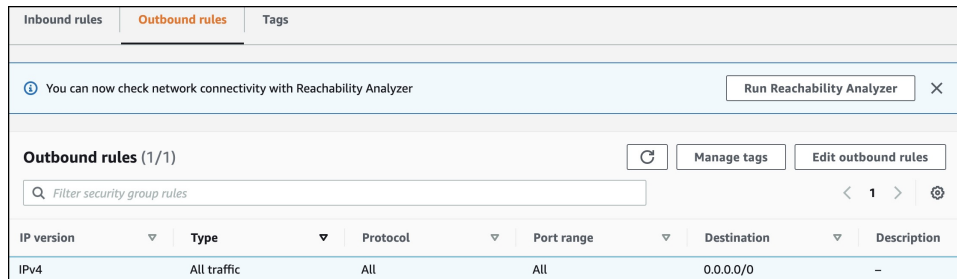
Figure 9: Configure These Inbound Rules Permitting Traffic to Specific Ports

	Name	Security group rule...	IP version	Type	Protocol	Port range
<input type="checkbox"/>	--	sg-r-0497e0b5ee57ae7...	IPv4	HTTPS	TCP	443
<input type="checkbox"/>	--	sg-r-0b120f3989c477140	IPv4	Custom UDP	UDP	2003
<input type="checkbox"/>	--	sg-r-084f5c1391adb52fa	IPv4	Custom TCP	TCP	8000
<input type="checkbox"/>	--	sg-r-02070569e30bbd...	IPv4	Custom UDP	UDP	161
<input type="checkbox"/>	--	sg-r-0bb0c8051cee0daf8	IPv4	SSH	TCP	22
<input type="checkbox"/>	--	sg-r-0c502fa77173670d8	IPv4	Custom TCP	TCP	8004

**Note** Using an inbound rule, you can also specify the network subnet range that can access this instance (For example, through SSH).

c) Configure the outbound rule shown in the following image.

**Figure 10: Configure This Outbound Rule**



**Note** For various connector services to work, you must open specific ports. See the respective **Information About Open Ports** section of the connector service for more information.

## Step 11

In the displayed **Select an existing key pair or create a new key pair** dialog box, do either of the following:

- Choose **Create a new key pair** from the drop-down list. Provide a **Key pair name** and click **Download Key Pair** to download it. Then click **Launch Instance** to launch the instance.
- Choose **Choose an existing key pair** from the drop-down list. Select the previously downloaded key pair from the **Select a key Pair** drop-down list. Then click **Launch Instance** to launch the instance.

**Figure 11: Create a New Key Pair**

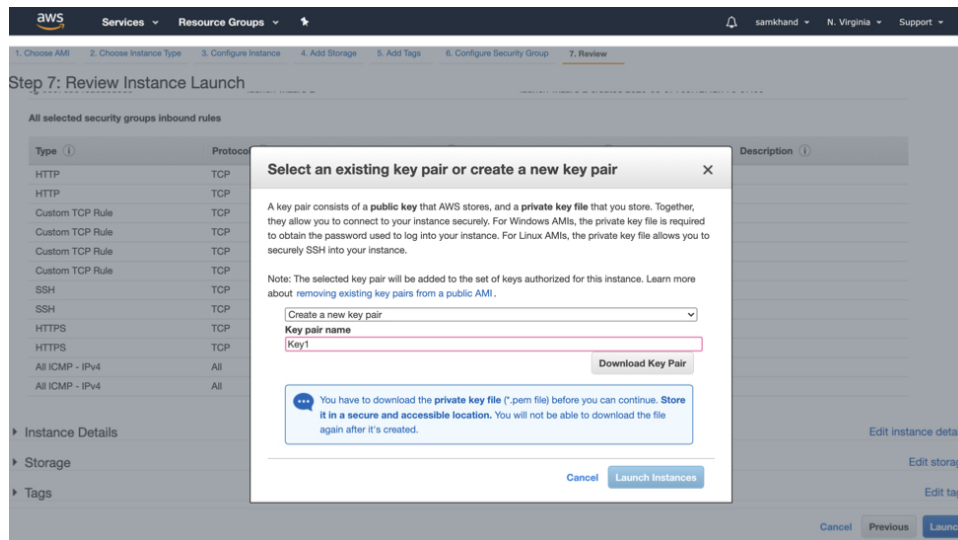
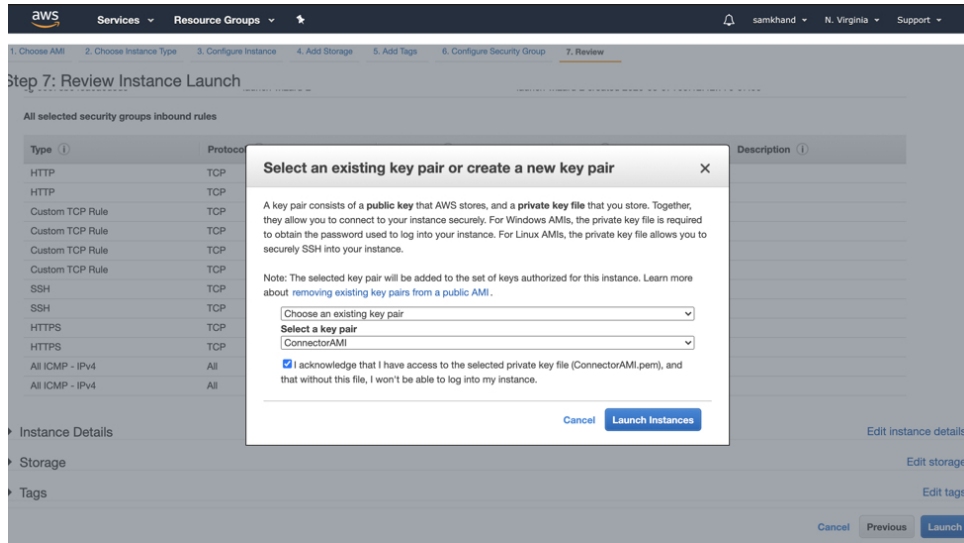


Figure 12: Choose an Existing Key Pair



**Step 12** After you have downloaded the key pair (.pem) file to your system, navigate to the file location. Configure appropriate permissions for the .PEM file using the **chmod** command.

```
chmod 400 /path/to/MyAccessKey1.pem
```

**Step 13** Review the instance and click **Launch**.

Figure 13: Review Instance and Launch

▼ **Summary**

Number of instances [Info](#)

---

[Software Image \(AMI\)](#)

cisco-dna-spaces-connector3-b8...[read more](#)  
ami-0ff155022ef237286

[Virtual server type \(instance type\)](#)

t2.medium

[Firewall \(security group\)](#)

eWLC

[Storage \(volumes\)](#)

1 volume(s) - 120 GiB

**Free tier:** In your first year includes 750 hours of t2.micro (or t3.micro in the Regions in which t2.micro is unavailable) instance usage on free tier AMIs per month, 30 GiB of EBS storage, 2 million IOs, 1 GB of snapshots, and 100 GB of bandwidth to the internet. ×

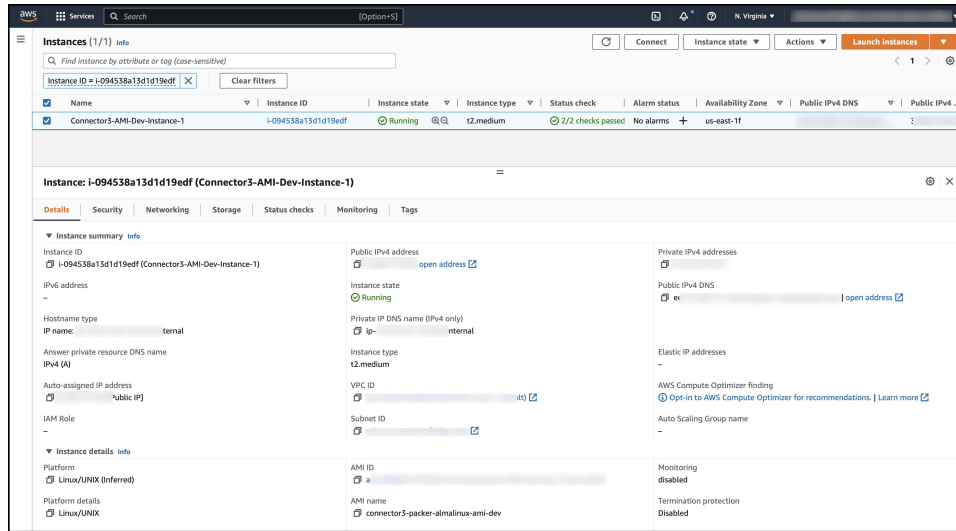
[Cancel](#) [Launch instance](#)

**Step 14**

On the EC2 dashboard, wait for the instance to finish launching and the status to change to **Running**. Alternatively, you can see the running instances on the **Instances** page. Click the instance to obtain the IPv4 address of the instance.



Figure 14: Obtain IPv4 Address of Instance

**Step 15**

Perform initial setup to configure a hostname, and change passwords for **spacesadmin** and **root** users.

a) Log in to the connector using the **ssh -i** command and the following parameters:

- The .PEM key pair downloaded in [Step 11](#)
- ec2-user
- The IPv4 address obtained in [Step 14](#)

```
ssh -i /path/to/key/MyAccessKey1.pem ec2-user@IPv4-address
```

b) Change passwords for **spacesadmin** and **root** users. Avoid a BAD PASSWORD prompt by complying with the following password requirements:

- Length is more than 14 characters.
- Includes at least one uppercase letter.
- Includes at least one lowercase letter.
- Includes at least one special character.

The following is a sample output of the command:

```
Welcome to Cisco Spaces Connector Setup
Changing password for user spacesadmin.
New password:
Retype new password:
passwd: all authentication tokens updated successfully.
Password changed successfully
Setting rbash...
Restarting docker...
Changing shell for root.
Shell changed.
Changing shell for spaces.
```

```
Remove default users...
```

```
Relabeled /etc/sudoers from unconfined_u:object_r:user_tmp_t:s0 to unconfined_u:object_r:etc_t:s0
```

```
Cisco Spaces Connector UI:  
https://XX.XXX.XX.XXX  
Username log in: spacesadmin  
The install is complete, a reboot will occur in 10 seconds...
```

Once the installation is complete, a reboot occurs within 10 seconds. Note down the public IP address before reboot.

**Step 16**

Log in to the connector and configure the connector further. Do one of the following using the public IPv4 address from the previous step (Step 15):

- Log in to the connector GUI using the browser window and the address `https://public-ipv4-address`
  - Log in to the connector CLI using the SSH command and the username **spacesadmin**. Use the command `ssh spacesadmin@public-ipv4-address`. When prompted, use the password configured for the **spacesadmin** user.
-