

Up and Running with Amazon EC2

December 7th, 2021 PDM Code Clinic
Clayton Cook

Overview

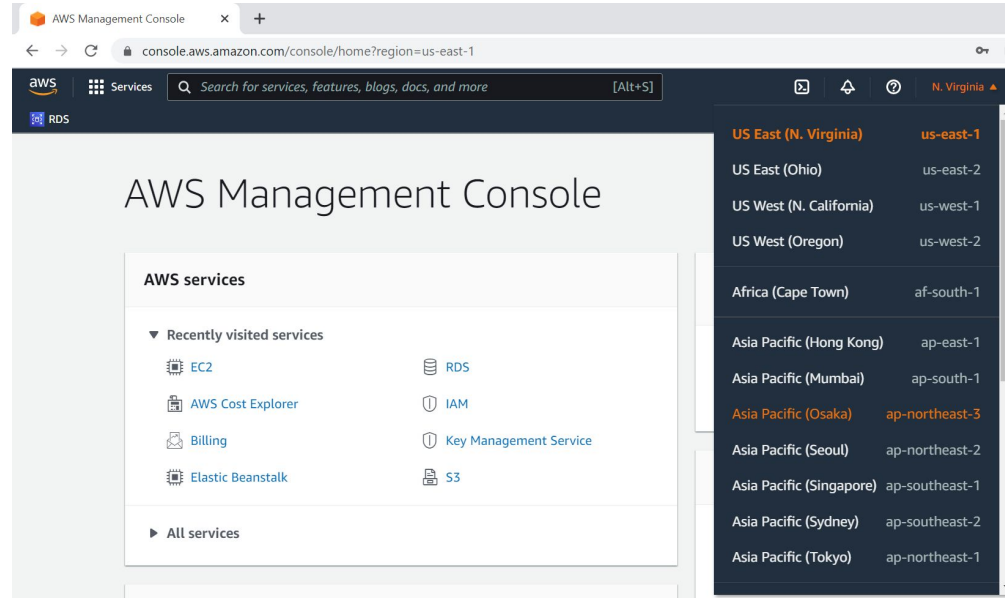
- Getting an EC2 instance up and running through the Amazon management console
 - Ssh, http and database connections
 - Installing Docker During Creation of EC2 Instance
- Connecting using SSH (Visual Studio Code)
- Cost of Cloud Resources
- Example: Running a Dockerized Django app on an EC2 instance

What is Amazon EC2?

- Amazon Elastic Compute Cloud (Amazon EC2) is a web service that provides secure, resizable compute capacity in the cloud.
- Can get a server up and running to use in minutes

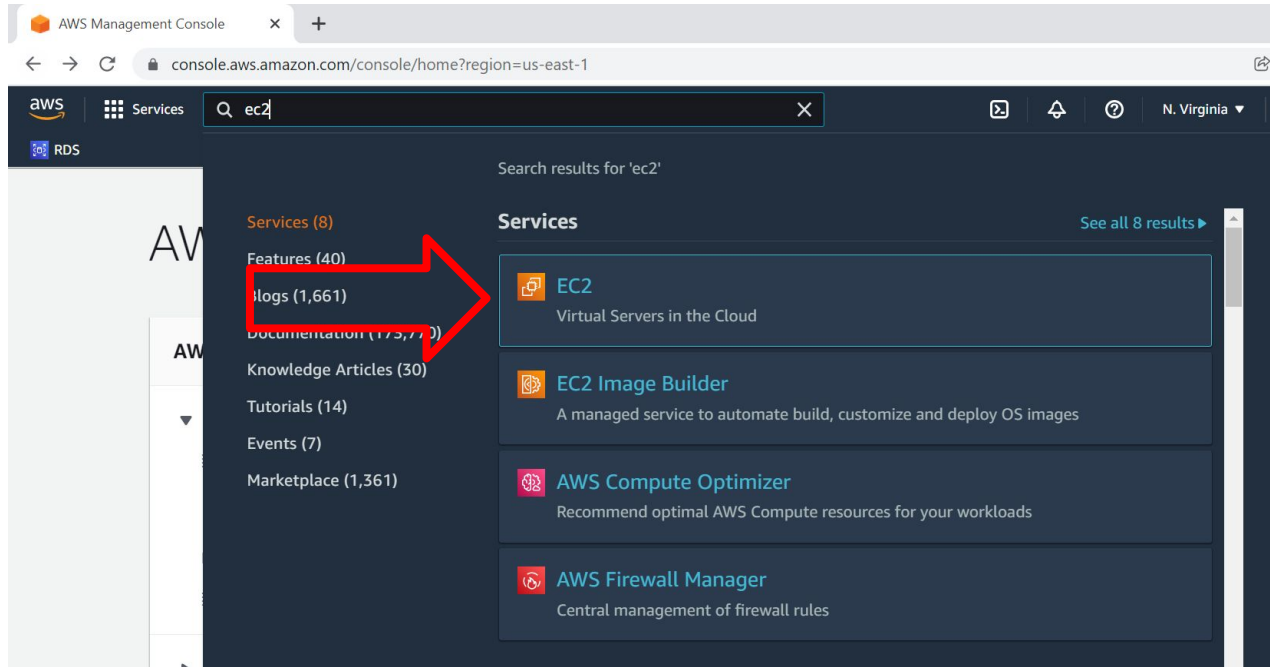
EC2 Instance Up and Running on Amazon

- Will be using the AWS Management Console
- First Pick Your Region where the EC2 instance will be running



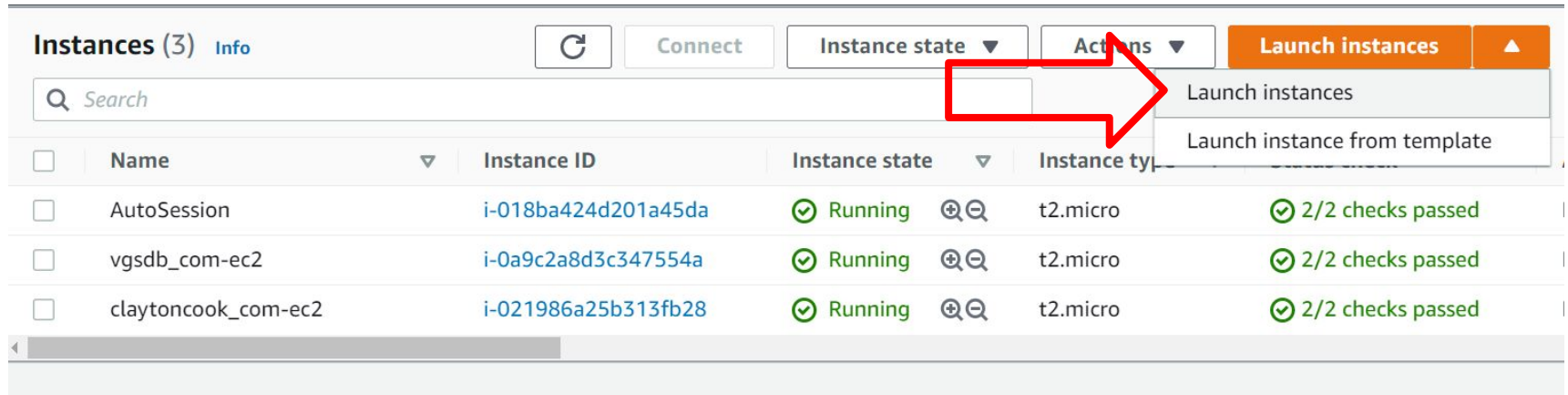
EC2 Instance Up and Running on Amazon

- Go to EC2 Console and Type “EC2”



EC2 Instance Up and Running on Amazon

- Launch Instance and Follow build instructions



The screenshot displays the AWS Management Console interface for EC2 instances. At the top, there's a header bar with 'Instances (3)' and an 'Info' link. Below this is a search bar and a table of instances. A red arrow points to the 'Actions' dropdown menu, which is open, showing options like 'Launch instances' and 'Launch instance from template'.

	Name	Instance ID	Instance state	Instance type	
<input type="checkbox"/>	AutoSession	i-018ba424d201a45da	Running	t2.micro	2/2 checks passed
<input type="checkbox"/>	vgsdb_com-ec2	i-0a9c2a8d3c347554a	Running	t2.micro	2/2 checks passed
<input type="checkbox"/>	claytoncook_com-ec2	i-021986a25b313fb28	Running	t2.micro	2/2 checks passed


EC2 Instance Up and Running on Amazon


- Step 1: Select Amazon Machine Image (AMI)
 - **Ubuntu Server 20.04 LTS (HVM), SSD Volume Type**


1. Choose AMI 2. Choose Instance Type 3. Configure Instance 4. Add Storage 5. Add Tags 6. Configure Security Group 7. Review

Step 1: Choose an Amazon Machine Image (AMI)

[Cancel and Exit](#)



	Ubuntu Server 20.04 LTS (HVM), SSD Volume Type - ami-083654bd07b5da81d (64-bit x86) / ami-04fe9398b2a27a600 (64-bit Arm)	Select
Free tier eligible	Ubuntu Server 20.04 LTS (HVM),EBS General Purpose (SSD) Volume Type. Support available from Canonical (http://www.ubuntu.com/cloud/services).	<input checked="" type="radio"/> 64-bit (x86) <input type="radio"/> 64-bit (Arm)
Root device type: ebs Virtualization type: hvm ENA Enabled: Yes		

	Ubuntu Server 18.04 LTS (HVM), SSD Volume Type - ami-0279c3b3186e54acd (64-bit x86) / ami-0528007a60177dd84 (64-bit Arm)	Select
Free tier eligible	Ubuntu Server 18.04 LTS (HVM),EBS General Purpose (SSD) Volume Type. Support available from Canonical (http://www.ubuntu.com/cloud/services).	<input checked="" type="radio"/> 64-bit (x86) <input type="radio"/> 64-bit (Arm)
Root device type: ebs Virtualization type: hvm ENA Enabled: Yes		

EC2 Instance Up and Running on Amazon

- Step 2: Choose an Instance Type
- t2.micro (Free Tier Eligible) and is smallest compute size

[1. Choose AMI](#) [2. Choose Instance Type](#) [3. Configure Instance](#) [4. Add Storage](#) [5. Add Tags](#) [6. Configure Security Group](#) [7. Review](#)

Step 2: Choose an Instance Type

Amazon EC2 provides a wide selection of instance types optimized to fit different use cases. Instances are virtual servers that can run applications. They have varying combinations of CPU, memory, storage, and networking capacity, and give you the flexibility to choose the appropriate mix of resources for your applications. [Learn more](#) about instance types and how they can meet your computing needs.

Filter by: All instance families Current generation [Show/Hide Columns](#)

Currently selected: t2.micro (- ECU, 1 vCPUs, 2.5 GHz, -, 1 GiB memory, EBS only)

	Family	Type	vCPUs	Memory (GiB)	Instance Storage (GB)	EBS-Optimized Available	Network Performance	IPv6 Support
<input type="checkbox"/>	t2	t2.nano	1	0.5	EBS only	-	Low to Moderate	Yes
<input checked="" type="checkbox"/>	t2	t2.micro <small>Free tier eligible</small>	1	1	EBS only	-	Low to Moderate	Yes
<input type="checkbox"/>	t2	t2.small	1	2	EBS only	-	Low to Moderate	Yes
<input type="checkbox"/>	t2	t2.medium	2	4	EBS only	-	Low to Moderate	Yes

EC2 Instance Up and Running on Amazon

- Step 3: Configure Instance Details
- Can use defaults

1. Choose AMI

2. Choose Instance Type

3. Configure Instance

4. Add Storage

5. Add Tags

6. Configure Security Group

7. Review

Step 3: Configure Instance Details

Configure the instance to suit your requirements. You can launch multiple instances from the same AMI, request Spot instances to take advantage of the lower pricing, assign an access management role to the instance, and more.

Number of instances ⓘ

1

Launch into Auto Scaling Group ⓘ

Purchasing option ⓘ

☐ Request Spot instances

Network ⓘ

vpc-ce8625b3 (default) ⌵

Create new VPC

Subnet ⓘ

No preference (default subnet in any Availability Zone) ⌵

Create new subnet

Auto-assign Public IP ⓘ

Use subnet setting (Enable) ⌵

Hostname type ⓘ

Use subnet setting (IP name) ⌵

DNS Hostname ⓘ

☒ Enable IP name IPv4 (A record) DNS requests

☒ Enable resource-based IPv4 (A record) DNS requests

☐ Enable resource-based IPv6 (AAAA record) DNS requests

Cancel

Previous

Review and Launch

Next: Add Storage

EC2 Instance Up and Running on Amazon

- Step 3: Configure Instance Details (Docker Install)
- If you want Docker Installed on setup can use advanced rules and run bash script below

```
#!/bin/bash
sudo snap install docker
sudo addgroup --system docker
sudo adduser ubuntu docker
newgrp docker
sudo snap disable docker
sudo snap enable docker
```

EC2 Instance Up and Running on Amazon

- Step 3: Configure Instance Details (Docker Install)

1. Choose AMI

2. Choose Instance Type

3. Configure Instance

4. Add Storage

5. Add Tags

6. Configure Security Group

7. Review

Step 3: Configure Instance Details

File systems 

Add file system

 [Create new file system](#)

▼ Advanced Details

Enclave 

☐ Enable

Metadata accessible 

Enabled

Metadata version 

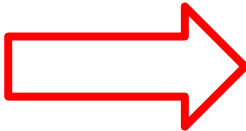
V1 and V2 (token optional)

Metadata token response hop limit 

1

User data 

☒ As text ☐ As file ☐ Input is already base64 encoded



```
#!/bin/bash
sudo snap install docker
sudo addgroup --system docker
sudo adduser ubuntu docker
newgrp docker
sudo snap disable docker
sudo snap enable docker
```

EC2 Instance Up and Running on Amazon

- Step 4: Add Storage

- Default of 8 GiB

1. Choose AMI

2. Choose Instance Type

3. Configure Instance

4. Add Storage

5. Add Tags

6. Configure Security Group

7. Review

Step 4: Add Storage

Your instance will be launched with the following storage device settings. You can attach additional EBS volumes and instance store volumes to your instance, or edit the settings of the root volume. You can also attach additional EBS volumes after launching an instance, but not instance store volumes. [Learn more](#) about storage options in Amazon EC2.

Volume Type ⓘ	Device ⓘ	Snapshot ⓘ	Size (GiB) ⓘ	Volume Type ⓘ	IOPS ⓘ	Throughput (MB/s) ⓘ	Delete on Termination ⓘ	Encryption ⓘ
Root	/dev/sda1	snap-0c97f1c43c6bb2043	<input type="text" value="8"/>	<div>General Purpose SSD (gp2) ▾</div>	100 / 3000	N/A	<input checked="" type="checkbox"/>	<div>Not Encrypt ▾</div>

Add New Volume

EC2 Instance Up and Running on Amazon

- Step 5: Add Tags

- Tags are metadata attached to an AWS resource. Each tag is a label consisting of a user-defined key and value. Tags can help you manage, identify, organize, search for, and filter resources.

1. Choose AMI

2. Choose Instance Type

3. Configure Instance

4. Add Storage

5. Add Tags

6. Configure Security Group

7. Review

Step 5: Add Tags

A tag consists of a case-sensitive key-value pair. For example, you could define a tag with key = Name and value = Webserver.

A copy of a tag can be applied to volumes, instances or both.

Tags will be applied to all instances and volumes. [Learn more](#) about tagging your Amazon EC2 resources.

Key (128 characters maximum)

Value (256 characters maximum)

Instances ⓘ

Volumes ⓘ

Network
Interfaces ⓘ

This resource currently has no tags

Choose the Add tag button or [click to add a Name tag](#).
Make sure your [IAM policy](#) includes permissions to create tags.

Add Tag

(Up to 50 tags maximum)

EC2 Instance Up and Running on Amazon

- Step 6: Configure Security Group
 - A set of firewall rules that control the traffic for your instance

Step 6: Configure Security Group

Description: launch-wizard-9 created 2021-12-06T09:09:43.409-05:00

Type ⓘ	Protocol ⓘ	Port Range ⓘ	Source ⓘ	Description ⓘ	
SSH ▾	TCP	22	Custom ▾ 0.0.0.0/0	e.g. SSH for Admin Desktop	✕
Custom TCP F ▾	TCP	8000	Custom ▾ 0.0.0.0/0, ::/0	e.g. SSH for Admin Desktop	✕
HTTP ▾	TCP	80	Custom ▾ 0.0.0.0/0, ::/0	e.g. SSH for Admin Desktop	✕
HTTPS ▾	TCP	443	Custom ▾ 0.0.0.0/0, ::/0	e.g. SSH for Admin Desktop	✕
Custom TCP F ▾	TCP	6543	My IP ▾ 104.52.8.79/32	e.g. SSH for Admin Desktop	✕

Add Rule



Warning

Rules with source of 0.0.0.0/0 allow all IP addresses to access your instance. We recommend setting security group rules to allow access from known IP addresses only.

EC2 Instance Up and Running on Amazon

- Step 6: Configure Security Group
- SSH: Port 22 - so you can remote into server
- TCP Port 8000 - I use this port so I can run development server to test connections
- HTTP Port 80 - Default port for connection to server over the internet
- HTTPS Port 443 - Default port for connection to server that is using a TLS certificate for encrypted web traffic data
- TCP Port 6543 - I use this port for database connection, note that I set the source to my IP address so that connection is limited

EC2 Instance Up and Running on Amazon

- Step 7: Review

1. Choose AMI2. Choose Instance Type3. Configure Instance4. Add Storage5. Add Tags6. Configure Security Group7. Review

Step 7: Review Instance Launch

Please review your instance launch details. You can go back to edit changes for each section. Click **Launch** to assign a key pair to your instance and complete the launch process.

Improve your instances' security. Your security group, launch-wizard-9, is open to the world.

Your instances may be accessible from any IP address. We recommend that you update your security group rules to allow access from known IP addresses only.

You can also open additional ports in your security group to facilitate access to the application or service you're running, e.g., HTTP (80) for web servers. [Edit security groups](#)

AMI Details

Free tier eligible

Ubuntu Server 20.04 LTS (HVM), SSD Volume Type - ami-083654bd07b5da81d

Ubuntu Server 20.04 LTS (HVM),EBS General Purpose (SSD) Volume Type. Support available from Canonical (<http://www.ubuntu.com/cloud/services>).

Root Device Type: ebsVirtualization type: hvm

Edit AMI

Instance Type

Instance Type	ECUs	vCPUs	Memory (GiB)	Instance Storage (GB)	EBS-Optimized Available	Network Performance
t2.micro	-	1	1	EBS only	-	Low to Moderate

Cancel

Previous

Launch

EC2 Instance Up and Running on Amazon

- Step 8: Create Key Pair and Download

Select an existing key pair or create a new key pair

A key pair consists of a **public key** that AWS stores, and a **private key file** that you store. Together, they allow you to connect to your instance securely. For Windows AMIs, the private key file is required to obtain the password used to log into your instance. For Linux AMIs, the private key file allows you to securely SSH into your instance. Amazon EC2 supports ED25519 and RSA key pair types.

Note: The selected key pair will be added to the set of keys authorized for this instance. Learn more about [removing existing key pairs from a public AMI](#).

Create a new key pair

Key pair type

☒ RSA ☐ ED25519

Key pair name


ec2_instance_example01

Download Key Pair

You have to download the **private key file** (*.pem file) before you can continue. **Store it in a secure and accessible location.** You will not be able to download the file again after it's created.

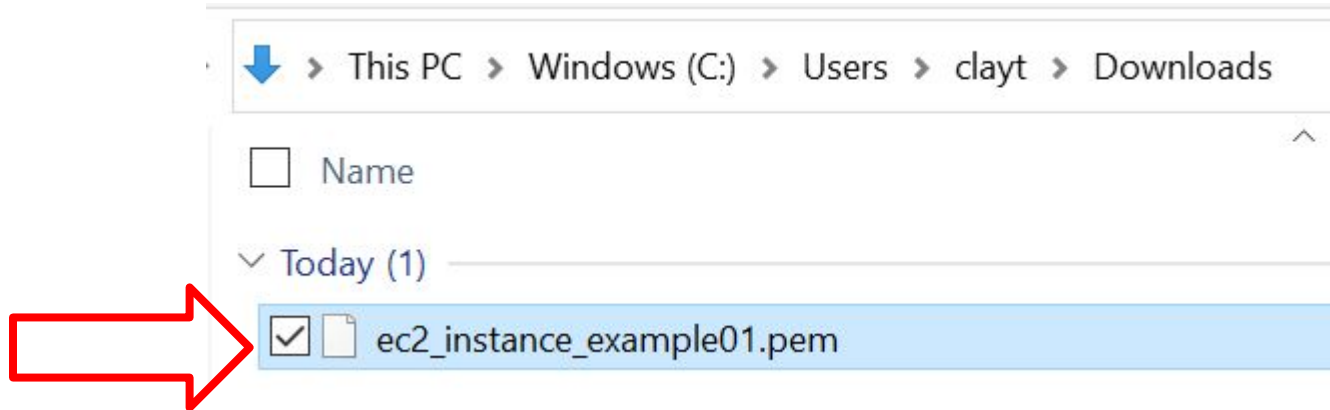
Cancel

Launch Instances



EC2 Instance Up and Running on Amazon

- Step 8: Create Key Pair and Download



EC2 Instance Up and Running on Amazon

- Step 9: Launch Instance

Select an existing key pair or create a new key pair

A key pair consists of a **public key** that AWS stores, and a **private key file** that you store. Together, they allow you to connect to your instance securely. For Windows AMIs, the private key file is required to obtain the password used to log into your instance. For Linux AMIs, the private key file allows you to securely SSH into your instance. Amazon EC2 supports ED25519 and RSA key pair types.

Note: The selected key pair will be added to the set of keys authorized for this instance. Learn more about [removing existing key pairs from a public AMI](#).

Create a new key pair

Key pair type

☒ RSA ☐ ED25519

Key pair name

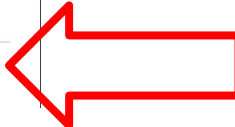
ec2_instance_example01

Download Key Pair

You have to download the **private key file** (*.pem file) before you can continue. **Store it in a secure and accessible location.** You will not be able to download the file again after it's created.

Cancel

Launch Instances



EC2 Instance Up and Running on Amazon

- Instance on EC2 Console after Status Checks

The screenshot displays the Amazon EC2 console interface. At the top, there's a header for 'Instances (1/4)' with a search bar, a refresh button, a 'Connect' button, and a dropdown for 'Instance state'. To the right are 'Actions' and a 'Launch instances' button. Below this is a table listing four EC2 instances. The first three are 'AutoSession', 'vgsdb_com-ec2', and 'claytoncook_com-ec2'. The fourth instance, 'PDM_Instance', is selected with a blue checkmark. Below the table, a detailed view for 'Instance: i-00818ea4739370146 (PDM_Instance)' is shown. This view includes tabs for 'Details', 'Security', 'Networking', 'Storage', 'Status checks', 'Monitoring', and 'Tags'. The 'Details' tab is active, showing a summary of the instance's configuration and status.

Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone
AutoSession	i-018ba424d201a45da	Running	t2.micro	2/2 checks passed	No alarms	us-east-1
vgsdb_com-ec2	i-0a9c2a8d3c347554a	Running	t2.micro	2/2 checks passed	No alarms	us-east-1
claytoncook_com-ec2	i-021986a25b313fb28	Running	t2.micro	2/2 checks passed	No alarms	us-east-1
PDM_Instance	i-00818ea4739370146	Running	t2.micro	2/2 checks passed	No alarms	us-east-1

Instance: i-00818ea4739370146 (PDM_Instance)

Details | Security | Networking | Storage | Status checks | Monitoring | Tags


Instance summary Info

Instance ID i-00818ea4739370146 (PDM_Instance)	Public IPv4 address 52.87.203.223 open address	Private IPv4 addresses 172.31.91.183
IPv6 address -	Instance state Running	Public IPv4 DNS ec2-52-87-203-223.compute-1.amazonaws.com open address











Connecting using SSH

- Connecting With Management Console

EC2 > Instances > i-00818ea4739370146

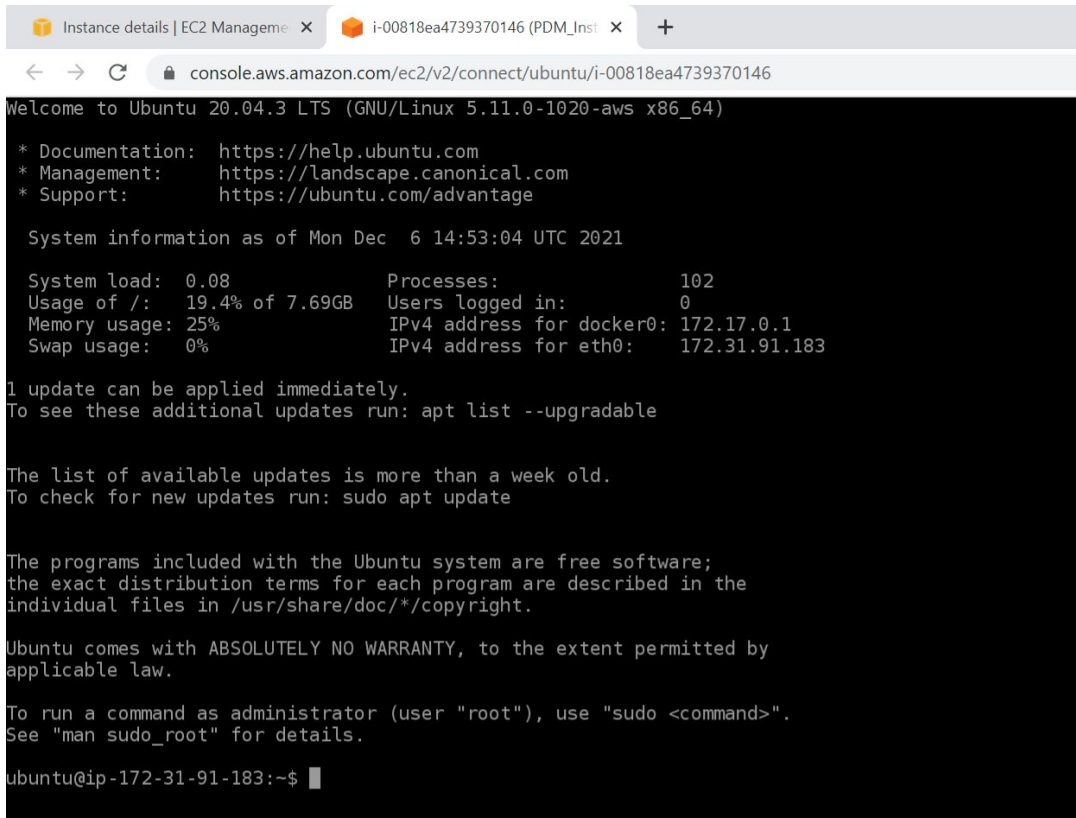
Instance summary for i-00818ea4739370146 (PDM_Instance) [Info](#)  [Connect](#) [Instance state ▼](#) [Actions ▼](#)

Updated less than a minute ago

Instance ID  i-00818ea4739370146 (PDM_Instance)	Public IPv4 address  52.87.203.223 open address 	Private IPv4 addresses  172.31.91.183
IPv6 address —	Instance state  Running	Public IPv4 DNS  ec2-52-87-203-223.compute-1.amazonaws.com open address 
Hostname type IP name: ip-172-31-91-183.ec2.internal	Private IP DNS name (IPv4 only)  ip-172-31-91-183.ec2.internal	Answer private resource DNS name IPv4 (A)
Instance type t2.micro	Elastic IP addresses —	VPC ID  vpc-ce8625b3 

Connecting using SSH

- Connecting With Management Console



The screenshot shows a web browser window with the AWS Management Console. The address bar displays the URL `console.aws.amazon.com/ec2/v2/connect/ubuntu/i-00818ea4739370146`. The terminal window shows the following output:

```
Welcome to Ubuntu 20.04.3 LTS (GNU/Linux 5.11.0-1020-aws x86_64)

* Documentation:  https://help.ubuntu.com
* Management:    https://landscape.canonical.com
* Support:       https://ubuntu.com/advantage

System information as of Mon Dec  6 14:53:04 UTC 2021

System load:  0.08               Processes:            102
Usage of /:   19.4% of 7.69GB    Users logged in:     0
Memory usage: 25%               IPv4 address for docker0: 172.17.0.1
Swap usage:   0%                IPv4 address for eth0:  172.31.91.183

1 update can be applied immediately.
To see these additional updates run: apt list --upgradable

The list of available updates is more than a week old.
To check for new updates run: sudo apt update

The programs included with the Ubuntu system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Ubuntu comes with ABSOLUTELY NO WARRANTY, to the extent permitted by
applicable law.

To run a command as administrator (user "root"), use "sudo <command>".
See "man sudo_root" for details.

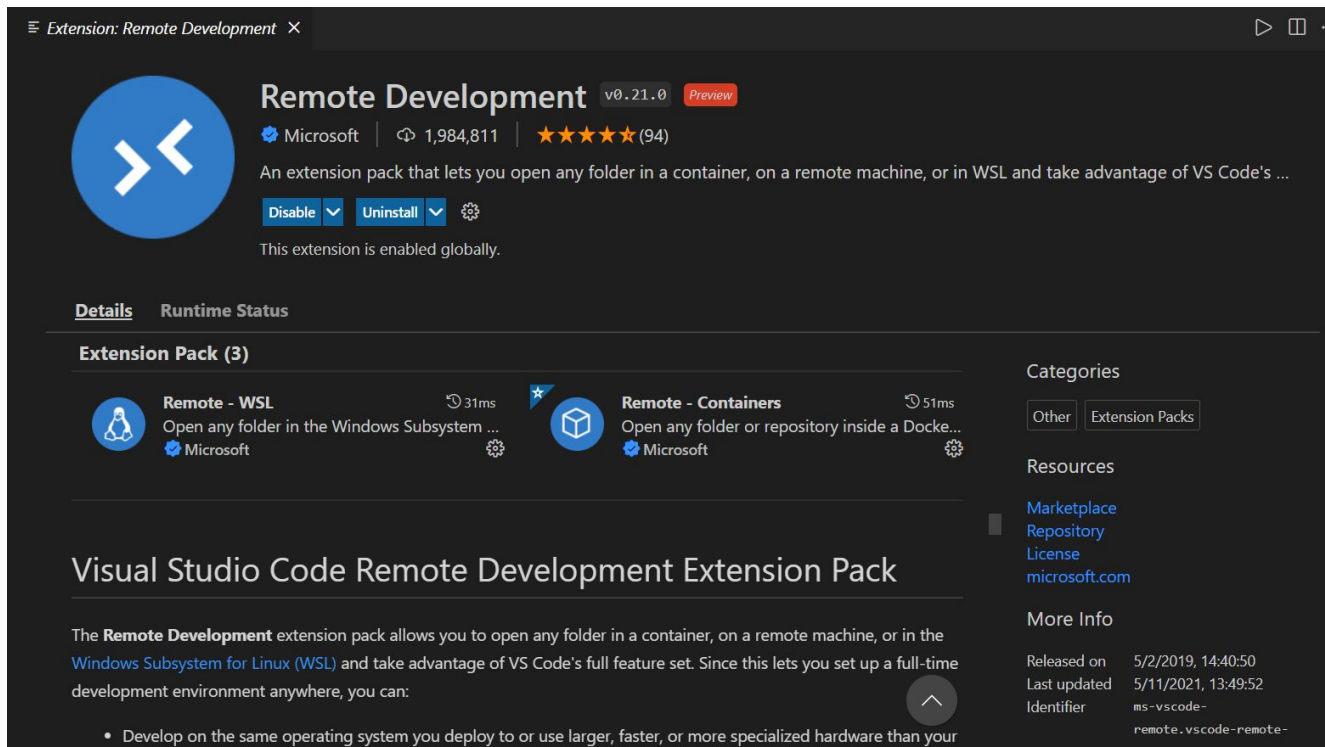
ubuntu@ip-172-31-91-183:~$
```

i-00818ea4739370146 (PDM_Instance)

Public IPs: 52.87.203.223 Private IPs: 172.31.91.183

Connecting using SSH

- I use Visual Studio Code extension **Remote Development**



The screenshot shows the Visual Studio Code interface with the 'Remote Development' extension page. The extension is by Microsoft, version v0.21.0, and is marked as a 'Preview'. It has 1,984,811 downloads and a 4.5-star rating from 94 reviews. The extension is currently enabled globally. Below the main description, there are two sub-extensions: 'Remote - WSL' and 'Remote - Containers'. The 'Remote - WSL' sub-extension is highlighted, showing it opens any folder in the Windows Subsystem for Linux (WSL) and is also by Microsoft. The 'Remote - Containers' sub-extension opens any folder or repository inside a Docker container and is also by Microsoft. The page also includes a 'Details' tab, a 'Runtime Status' section, and a 'Visual Studio Code Remote Development Extension Pack' section. The 'Visual Studio Code Remote Development Extension Pack' section explains that the 'Remote Development' extension pack allows you to open any folder in a container, on a remote machine, or in the Windows Subsystem for Linux (WSL) and take advantage of VS Code's full feature set. It also provides a link to the 'Marketplace Repository License' on 'microsoft.com'. The 'More Info' section shows the release date (5/2/2019, 14:40:50), the last updated date (5/11/2021, 13:49:52), and the identifier (ms-vscode-remote.remote-vscode-remote).

Extension: Remote Development X

Remote Development v0.21.0 Preview

Microsoft | 1,984,811 | ★★★★★ (94)

An extension pack that lets you open any folder in a container, on a remote machine, or in WSL and take advantage of VS Code's ...

[Disable](#) [Uninstall](#) ⚙️

This extension is enabled globally.

Details **Runtime Status**

Extension Pack (3)

Remote - WSL 31ms
Open any folder in the Windows Subsystem ...
Microsoft

Remote - Containers 51ms
Open any folder or repository inside a Docker...
Microsoft

Visual Studio Code Remote Development Extension Pack

The **Remote Development** extension pack allows you to open any folder in a container, on a remote machine, or in the [Windows Subsystem for Linux \(WSL\)](#) and take advantage of VS Code's full feature set. Since this lets you set up a full-time development environment anywhere, you can:

- Develop on the same operating system you deploy to or use larger, faster, or more specialized hardware than your

Categories
[Other](#) [Extension Packs](#)

Resources
[Marketplace](#)
[Repository](#)
[License](#)
[microsoft.com](#)

More Info
Released on 5/2/2019, 14:40:50
Last updated 5/11/2021, 13:49:52
Identifier ms-vscode-remote.remote-vscode-remote

Connecting using SSH


- HostName can be found on EC2 management console as the Public IPv4 DNS

EC2 > Instances > i-00818ea4739370146

Instance summary for i-00818ea4739370146 (PDM_Instance) [Info](#) [Refresh](#) [Connect](#) [Instance state ▼](#) [Actions ▼](#)

Updated less than a minute ago

Instance ID i-00818ea4739370146 (PDM_Instance)	Public IPv4 address 52.87.203.223 open address	Private IPv4 addresses 172.31.91.183
IPv6 address -	Instance state Running	Public IPv4 DNS ec2-52-87-203-223.compute-1.amazonaws.com open address
Hostname type IP name: ip-172-31-91-183.ec2.internal	Private IP DNS name (IPv4 only) ip-172-31-91-183.ec2.internal	Answer private resource DNS name IPv4 (A)
Instance type t2.micro	Elastic IP addresses -	VPC ID vpc-ce8625b3



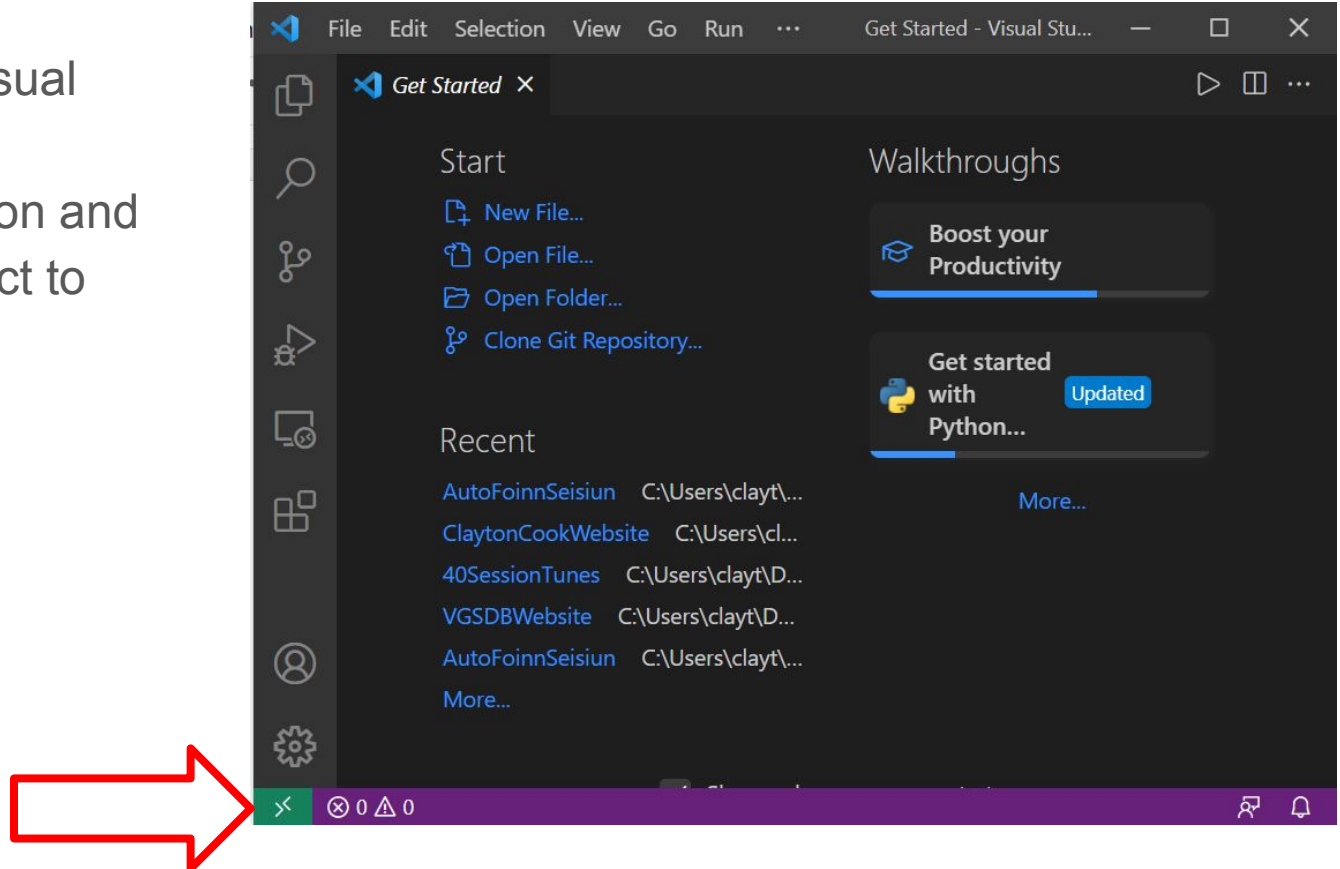
Connecting using SSH

- Setup you config file for connection settings

```
≡ config  X
C: > Users > clayt > .ssh > ≡ config
27  Host PDM_Instance
28      HostName ec2-52-87-203-223.compute-1.amazonaws.com
29      User ubuntu
30      IdentityFile "C:\Users\clayt\.ssh\ec2_instance_example01.pem"
31
```

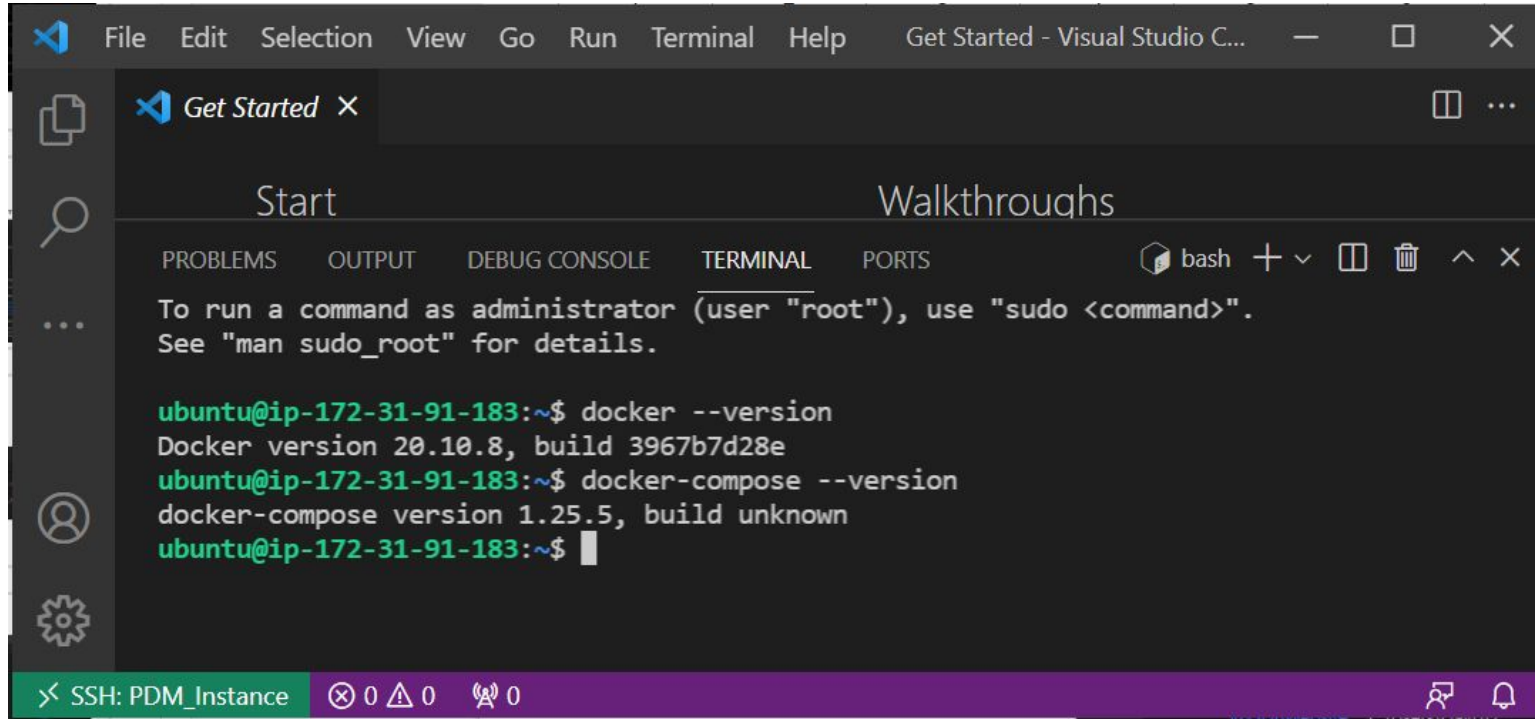
Connecting using SSH

- Connecting Visual Studio
- Click Green Icon and Select “Connect to Host...”



Connecting using SSH

- Connecting Visual Studio

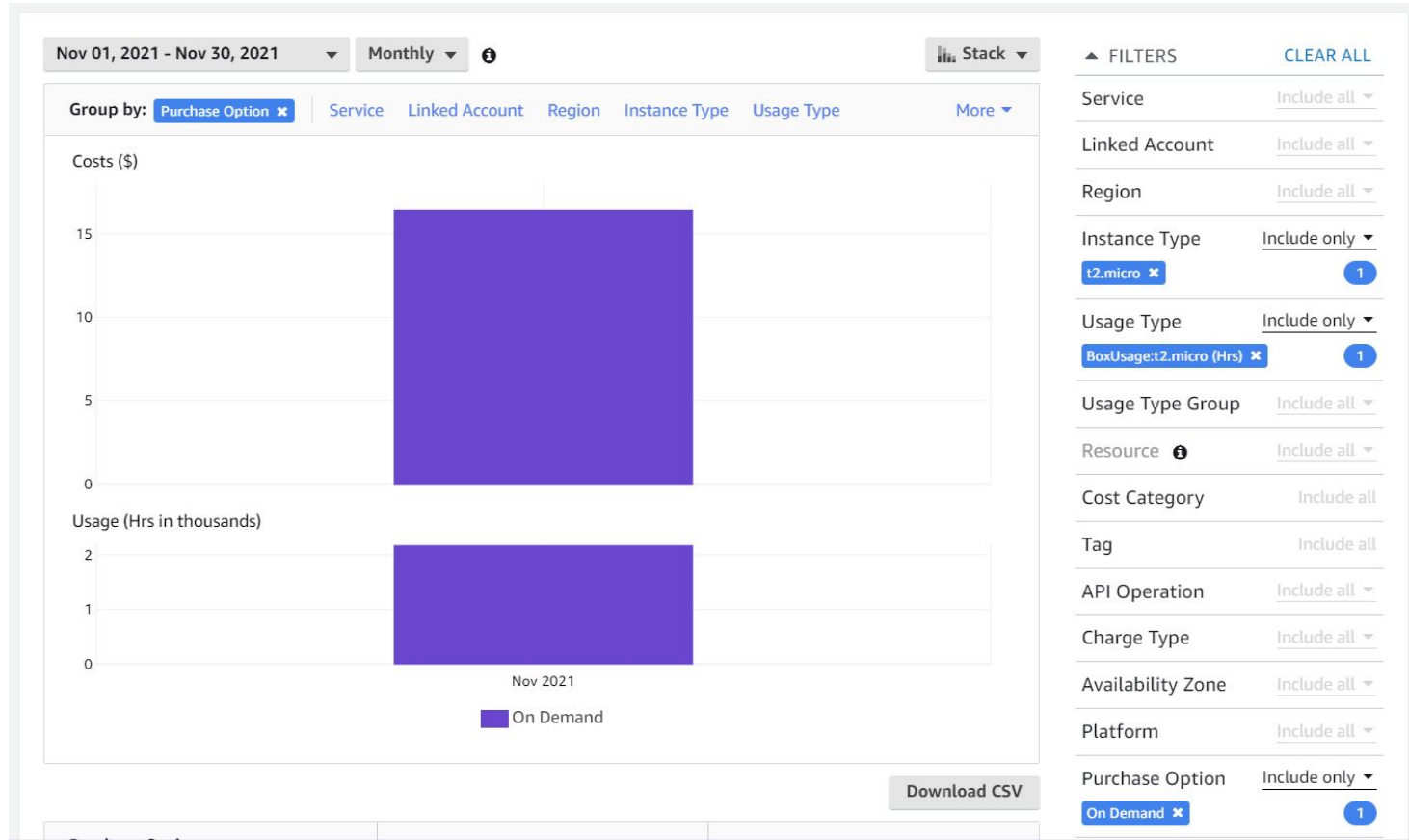


The screenshot shows the Visual Studio Code interface with a terminal window open. The terminal is connected via SSH to a PDM Instance. The terminal output shows the results of running 'docker --version' and 'docker-compose --version' commands.

```
ubuntu@ip-172-31-91-183:~$ docker --version
Docker version 20.10.8, build 3967b7d28e
ubuntu@ip-172-31-91-183:~$ docker-compose --version
docker-compose version 1.25.5, build unknown
ubuntu@ip-172-31-91-183:~$
```

Cost of Cloud Resources

- AWS Cost Explorer



Cost of Cloud Resources

- Will be dependent on location of server
- 00.00758 \$/Hr ~ \$6/month
- Can lower cost with purchase of Savings Plans within Cost Management console
- For first year of AWS can use a month worth of t2.micro for free

Running a Dockerized Django app on an EC2 instance

Run A Few Commands to Install Example Applications

- `git clone https://github.com/claytoncook12/PDMPresentation20211207.git`
- `cd PDMPresentation20211207/`

Run Docker Image

- `docker-compose -f docker-compose.yml build`
- `docker-compose -f docker-compose.yml up -d`

Connecting to Running Instance

- 52.87.203.223 (Will Change if Instance is Stopped or Rebooted)

EC2 > Instances > i-00818ea4739370146

Instance summary for i-00818ea4739370146 (PDM_Instance) [Info](#)

Updated less than a minute ago

[Refresh](#) [Connect](#) [Instance state ▼](#) [Actions ▼](#)

Instance ID i-00818ea4739370146 (PDM_Instance)	Public IPv4 address 52.87.203.223 open address	Private IPv4 addresses 172.31.91.183
IPv6 address -	Instance state Running	Public IPv4 DNS ec2-52-87-203-223.compute-1.amazonaws.com open address
Hostname type IP name: ip-172-31-91-183.ec2.internal	Private IP DNS name (IPv4 only) ip-172-31-91-183.ec2.internal	Answer private resource DNS name IPv4 (A)
Instance type t2.micro	Elastic IP addresses -	VPC ID vpc-ce8625b3
AWS Compute Optimizer finding Opt-in to AWS Compute Optimizer for recommendations. Learn more	IAM Role -	Subnet ID subnet-dfc15cfe

