FRANKFURT UNIVERSITY OF APPLIED SCIENCES

Project Report

Project: Digitalization SS 2023

IoT - From the Microcontroller to the Cloud

by

Alexander Atanassov
and
Binyam Tekeste

Supervisor
Prof. Dr. Oliver Hahm

Submission Date
July 28th, 2023

Table of Contents

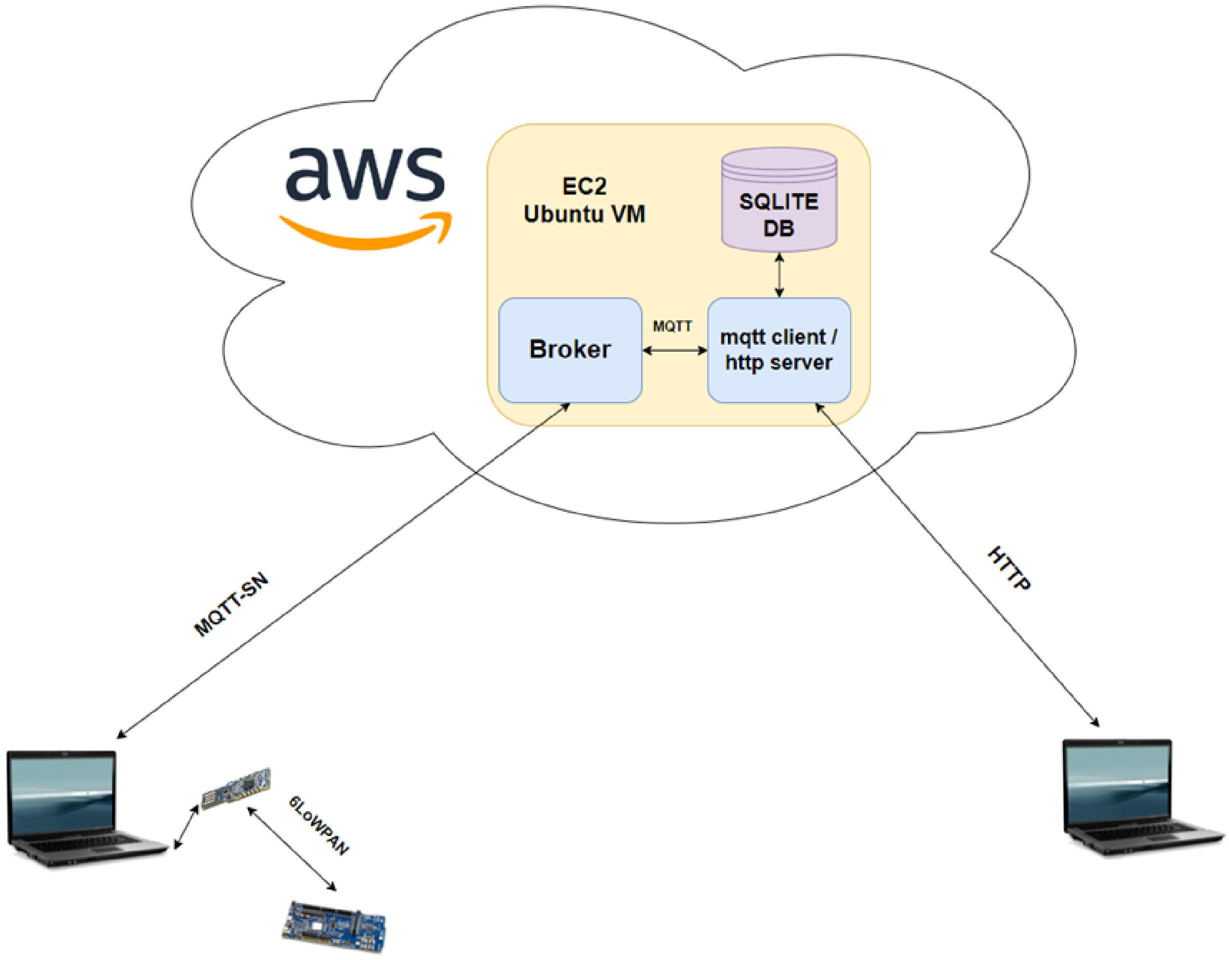
- Overview
- Set Up
  - Virtual Machine
  - Border Router
  - Sensor Node
  - Cloud
  - User Interface
- Start The System
- Sources

Overview

Project description:

The goal of the project is to create a system where sensor data can be send from a microcontroller to a cloud. The data should be send every *x* (e.g. 60) seconds and must be stored at the cloud. Also a user interface should be implemented where the data can be visualized. The UI must run on any local PC. The user should also be able to send commands back to the sensor node (e.g. request the sensor data immediately).

System architecture:



The system consists of three main parts:

- The first part is responsible for sending the sensor data to the cloud. It consists of a *nrf52840dk* board which captures the data and send it to the cloud using MQTT-SN protocol, a *nrf52840dongle* as a border router to route the packets between a 6Lo network and a normal IPv6 network and a Linux system with IPv6 connection to forwards the packets.
- The second part is the cloud. It consists of a MQTT broker as a middleware between the sensor node and the backend script. The backend script is responsible for storing the data into a SQLITE database and sending it to the end user.
- The last part is the user interface which is a desktop program and can run on any PC with IPv6 connection.

Set Up

In this chapter all of the system components will be explained. For each component there will be a guide how to set it up.

Virtual Machine

Overview:

For this project, a Linux operating system is required. It can be either be a host system or a guest system in a virtual machine. The machine on which Linux is running is used to forward IPv6 packets from the the sensor node to the cloud and back. It is also used for implementation tasks, connecting with the boards, monitoring the board terminal and debugging tasks. If a Linux system is already installed, the VirtualBox installation can be skipped.

Set up:

The following steps will explain how to set up an Ubuntu virtual machine. After that, it will be explained what configurations have to be done and what software have to be installed.

Virtual box and Ubuntu installation:

1. Install VirtualBox and the extension pack.
2. Download an Ubuntu Desktop image.
3. Follow the tutorial to create a Ubuntu virtual machine.
4. Install the Guest Additions.

Configurations and installations:

1. Allow IPv6 forwarding.
   - Open the file `/etc/sysctl.conf`
   - Make sure that the forwarding is enabled

   ```
   net.ipv6.conf.all.forwarding=1
   ```

2. Set up a wireguard IPv6 connection
   - Install wireguard

   ```
   sudo apt install wireguard
   ```

   - Set up a new wireguard connection using e.g. Advanced Network Configuration
   - Add the connection details:

   ```
   [Interface]
   PrivateKey = ****************************************
   Address = 2001:470:7347:c200:1000::/64
   ```

```
[Peer]
PublicKey = OzafSSqhtZDkHLgSIXY6a3n6Yi4EK9W3npfbWKA0VFc=
AllowedIPs = ::/0
Endpoint = 176.9.132.254:51820
PersistentKeepAlive = 60
```

3. To be able to flash programs on the boards, the following software has to be installed:
   - J-Link
   - gcc-multilib
   - arm-none-eabi-gcc
   - openocd
   - nrfjprog
   - nrf-udev
   - nrfutil
4. Clone the RIOT repository,

```
git clone git@github.com:RIOT-OS/RIOT.git
```

5. Clone this project.

```
git clone https://github.com/alllexander1/IoTProject.git
```

Border Router

Overview:

The border router is used to route packets between the sensor node and the internet. This component is necessary because the sensor node cannot directly send packets to the internet, since it is in a 6Lo network. The border router routes the packets between a 6Lo network and a 'normal' IPv6 network. It has two interfaces: A downstream to run 6LoWPAN and an IPv6 uplink. The program is based on the "gnrc_border_router" example program with small modifications so it can run on a *nrf52840dongle*.

Set up:

1. Open the Makefile of the *Border Router* program.
2. Set the path to the RIOT directory corectly.

```
RIOTBASE ?= $<Path to RIOT>
```

3. Set the IPv6 prefix to the prefix of your network. Make sure to change the last two digits.

```
IPV6_PREFIX ?= 2001:470:7347:c211::/64
```

4. Open a terminal and navigate to the *Border Router* folder.
5. Check the path to the serial port of the dongle.

```
make list-ttys
```

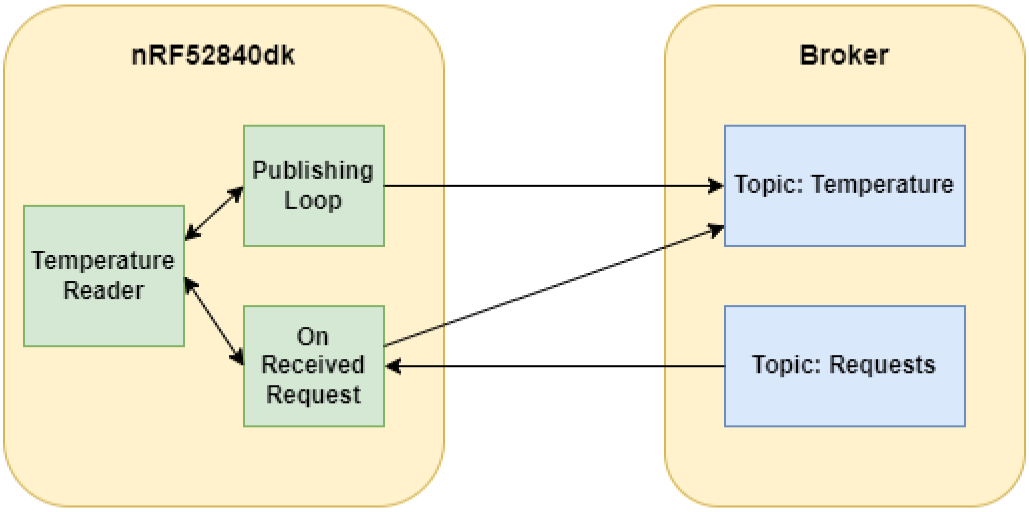
6. Flash the program on a *nrf52840dongle*. Eg for port: "/dev/ttyACM0".

```
PORT=/dev/ttyACM0 BOARD="nrf52840dongle" make all flash
```

Sensor Node

Overview:

The sensor node component is responsible for sending temperature values to the cloud. To do that, the *SAUL* module is used to read the current temperature from the build-in sensor. To send the data to the cloud, MQTT-SN (MQTT for Sensor Networks) protocol is used. The temperature data is sent every *x* seconds by publishing the value on a topic called "temperature". The sensor node is also listening for commands from the user. For that, it subscribes for a topic "requests". When a request is received, a callback function checks the command and publish the temperature immediately. For now there is only one valid command "get_temp_now". To make sure that the temperature reading function is not called by the callback function and the publishing loop at the same time, it is synchronized using *mutex*. For the MQTT-SN client, *EMCUTE* module is used. The diagram bellow shows the communication process between the sensor node and the MQTT broker.



Set up:

1. Open a terminal and unlock the *nrf52840dk* board.

```
nrfjprog --recover
```

2. In the *Sensor Node* folder open the main.c file.
3. Change the destination IPv6 address to the address of your EC2 instance.

    ```
    char * dest_addr = "<EC2 IPv6 address>";
    ```

4. The publishing interval can be changed here:

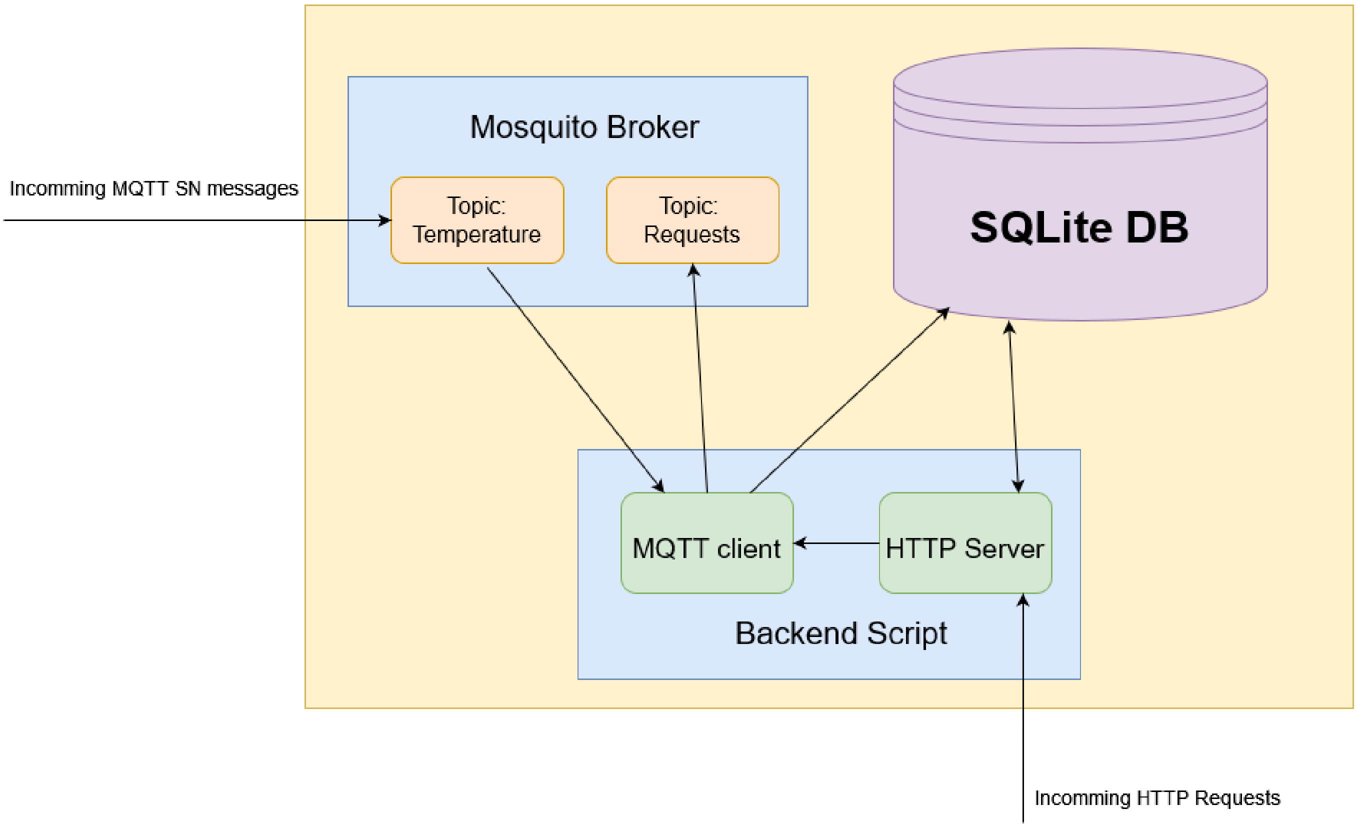
    ```
    xtimer_sleep(60);
    ```

- Note: The program should be flashed when the cloud part is running, since a connection loop or reconnecting mechanism is not implemented.

Cloud

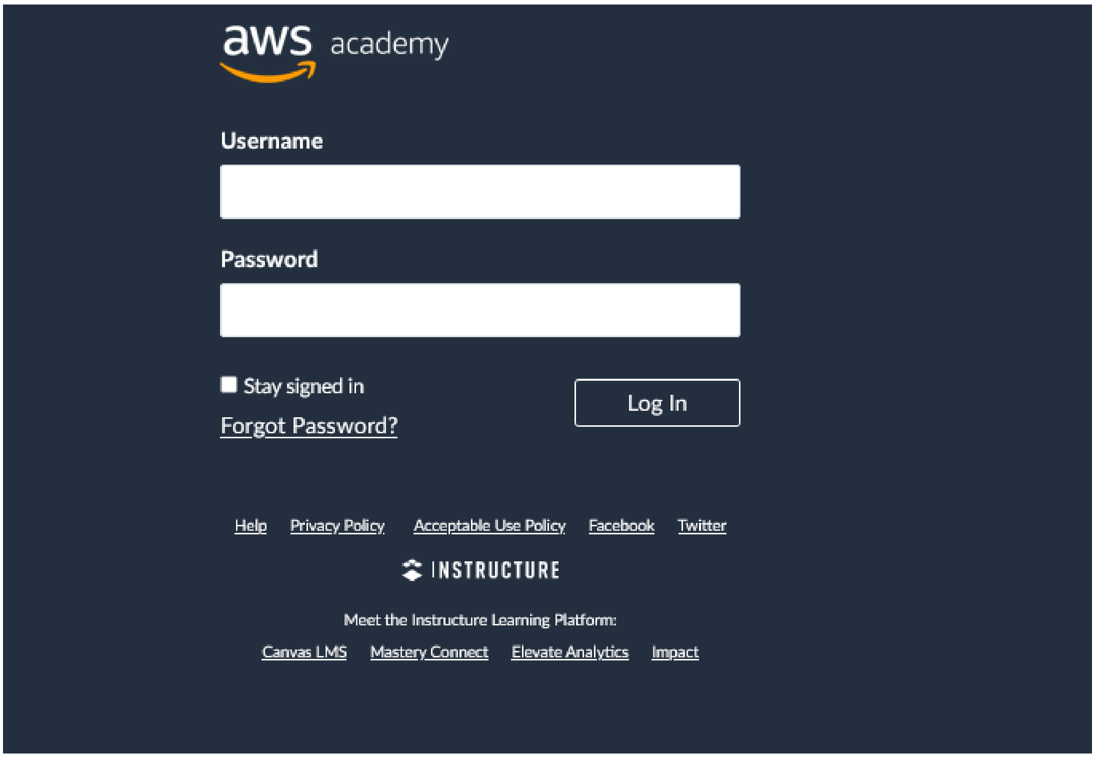
Overview:

For the cloud part AWS (Amazon Web Services) is used. The cloud system consists of an Ubuntu EC2 instance with an IPv6 address. The instance itself consists of three components. The first one is a *Mosquito RSMB broker* which is a middleware between MQTT clients. The second component is a *SQLITE database* where the temperature values and the corresponding date and time are stored. The last part is a *Backend Script* which is responsible for the communication with the sensor node and the end user. It consists of two parts: a MQTT client (Paho) and a HTTP server (Flask). The MQTT client is listening for messages from the topic "temperature". When a message is received, it is processed and stored at the database. The HTTP server has two APIs. The first one receives the latest ID available at the user interface and returns newer data entries if available. The second one is for sending a request to the sensor node. For that the MQTT client publishes a message "get_temp_now" on the topic "requests".

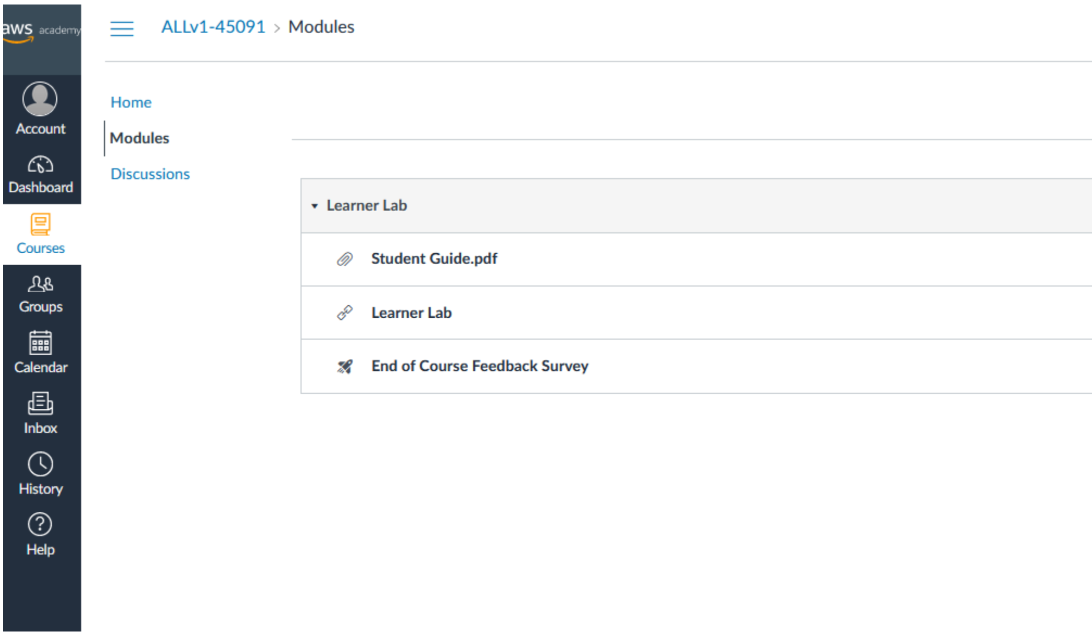


Set up

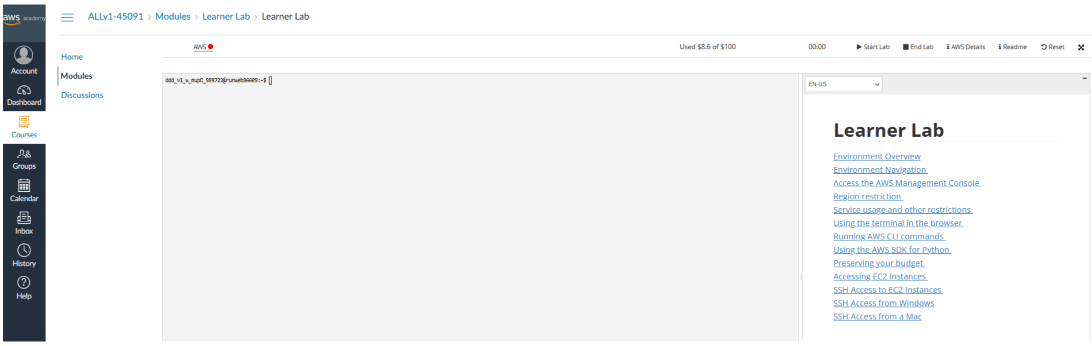
Create EC2 instance:



After signing up, you have to put in your username and password and hit the log in button.

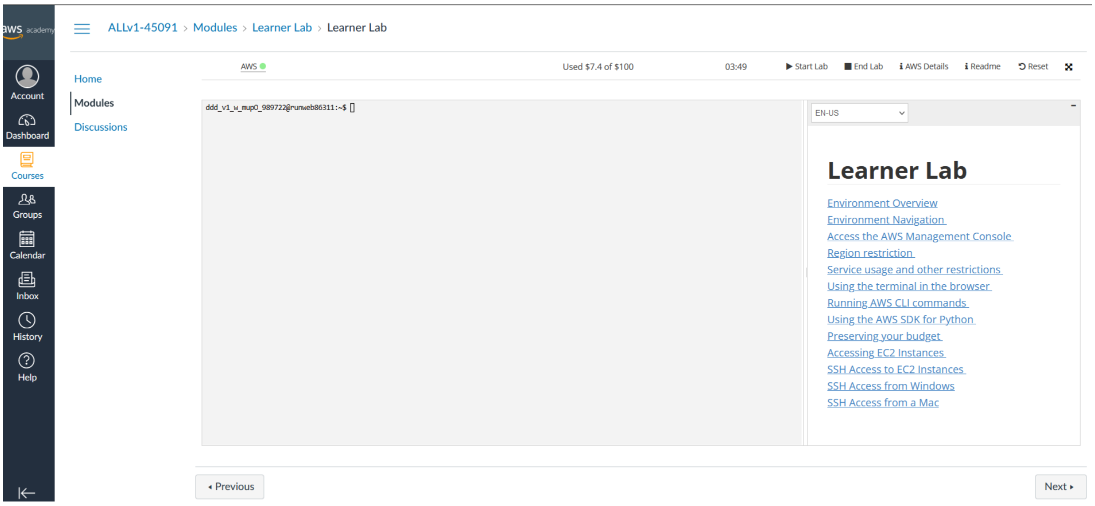


In the aws academy navigation you'll click on Modules and select Learner Lab.



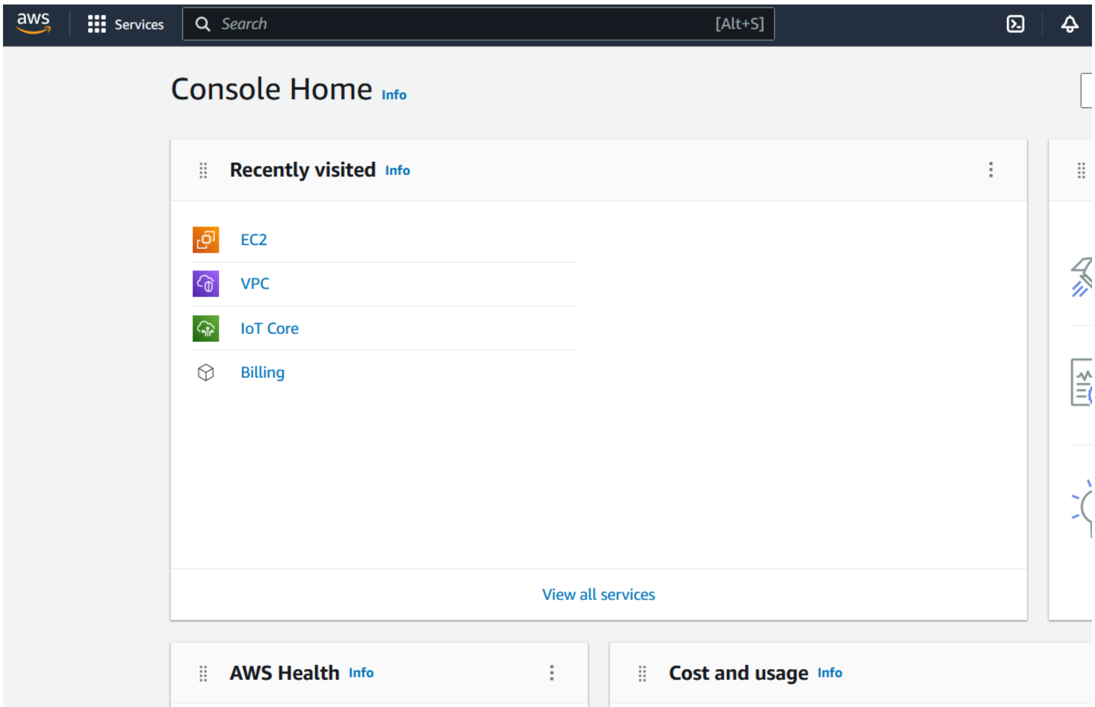
```
ddd_v1_w_mup0_989722@runweb86609:~$ 
```

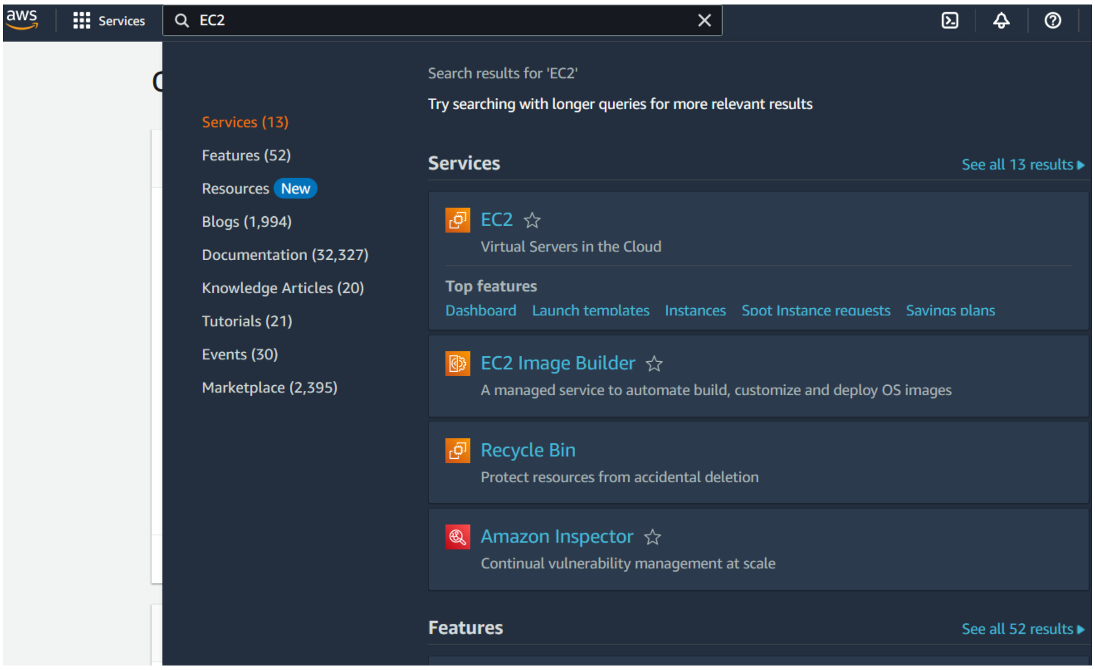
In the Learner Lab environment click on the right side "Start Lab" and wait till the red button next to "AWS" on the left side turns green.



```
ddd_v1_w_mup0_989722@runweb86311:~$ 
```

Click on "AWS" next to the green button and wait. If nothing happens, change your browser or check your ad blockers and settings.





Search results for 'EC2'

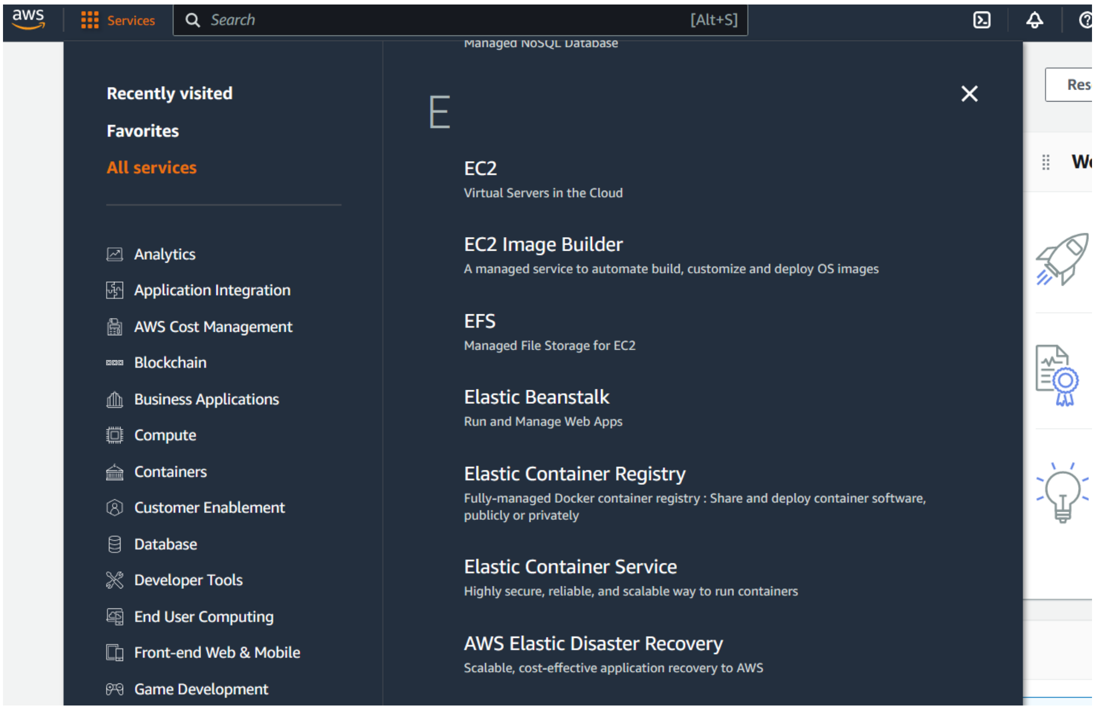
Try searching with longer queries for more relevant results

EC2 ☆
Virtual Servers in the Cloud

EC2 Image Builder ☆
A managed service to automate build, customize and deploy OS images

Recycle Bin
Protect resources from accidental deletion

Amazon Inspector ☆
Continual vulnerability management at scale



Services Search [Alt+S]

Recently visited

Favorites

All services

- Analytics
- Application Integration
- AWS Cost Management
- Blockchain
- Business Applications
- Compute
- Containers
- Customer Enablement
- Database
- Developer Tools
- End User Computing
- Front-end Web & Mobile
- Game Development

Managed NoSQL Database

E

EC2
Virtual Servers in the Cloud

EC2 Image Builder
A managed service to automate build, customize and deploy OS images

EFS
Managed File Storage for EC2

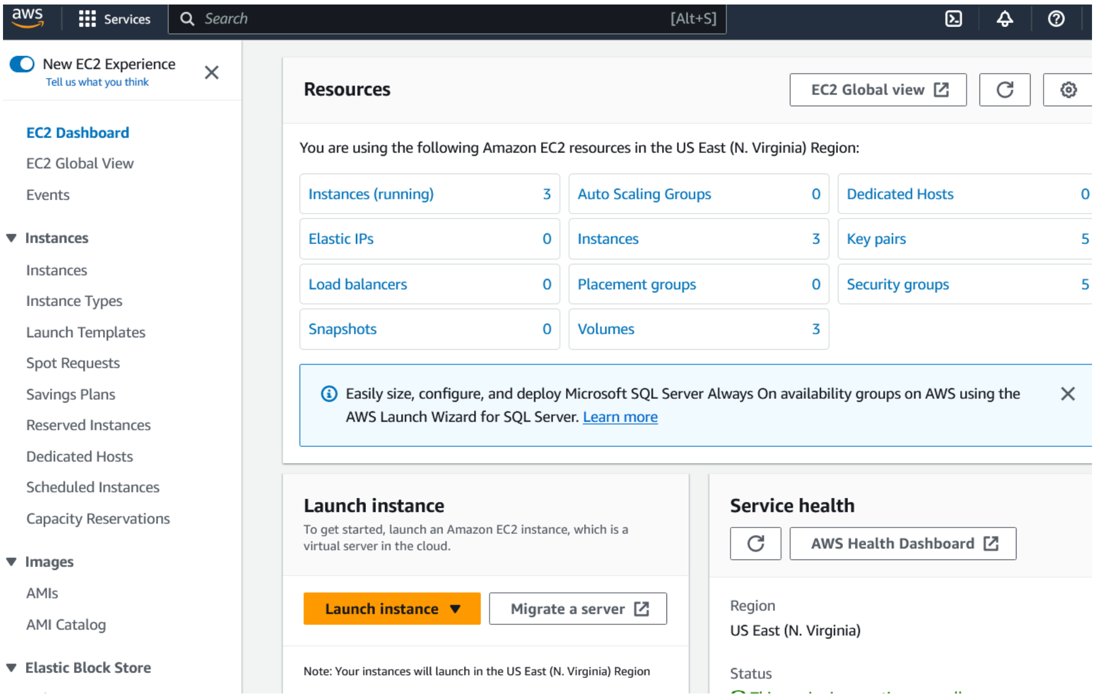
Elastic Beanstalk
Run and Manage Web Apps

Elastic Container Registry
Fully-managed Docker container registry : Share and deploy container software, publicly or privately

Elastic Container Service
Highly secure, reliable, and scalable way to run containers

AWS Elastic Disaster Recovery
Scalable, cost-effective application recovery to AWS

On your first visit, click on Services and scroll down to E. Choose EC2. A other option would be to write EC2 in the search bar and choose it from there. After visiting the EC2 (Instances (running)) Page you can also click on EC2 under Recently visited.



Services Search [Alt+S]

New EC2 Experience
Tell us what you think

- EC2 Dashboard
- EC2 Global View
- Events
- ▼ Instances
  - Instances
  - Instance Types
  - Launch Templates
  - Spot Requests
  - Savings Plans
  - Reserved Instances
  - Dedicated Hosts
  - Scheduled Instances
  - Capacity Reservations
- ▼ Images
  - AMIs
  - AMI Catalog
- ▼ Elastic Block Store

Resources   EC2 Global view

You are using the following Amazon EC2 resources in the US East (N. Virginia) Region:

Instances (running)	3	Auto Scaling Groups	0	Dedicated Hosts	0
Elastic IPs	0	Instances	3	Key pairs	5
Load balancers	0	Placement groups	0	Security groups	5
Snapshots	0	Volumes	3		

Easily size, configure, and deploy Microsoft SQL Server Always On availability groups on AWS using the AWS Launch Wizard for SQL Server. Learn more

Launch instance
To get started, launch an Amazon EC2 instance, which is a virtual server in the cloud.

Launch instance ▼   Migrate a server

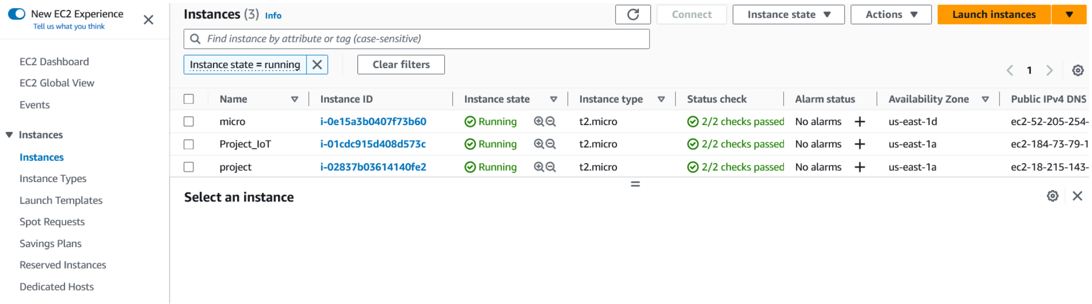
Note: Your instances will launch in the US East (N. Virginia) Region

Service health
AWS Health Dashboard

Region
US East (N. Virginia)

Status

Click on Instances (running)



New EC2 Experience
Tell us what you think

- EC2 Dashboard
- EC2 Global View
- Events
- ▼ Instances
  - Instances
  - Instance Types
  - Launch Templates
  - Spot Requests
  - Savings Plans
  - Reserved Instances
  - Dedicated Hosts

Instances (3) Info   Connect   Instance state ▼   Actions ▼   Launch instances ▼

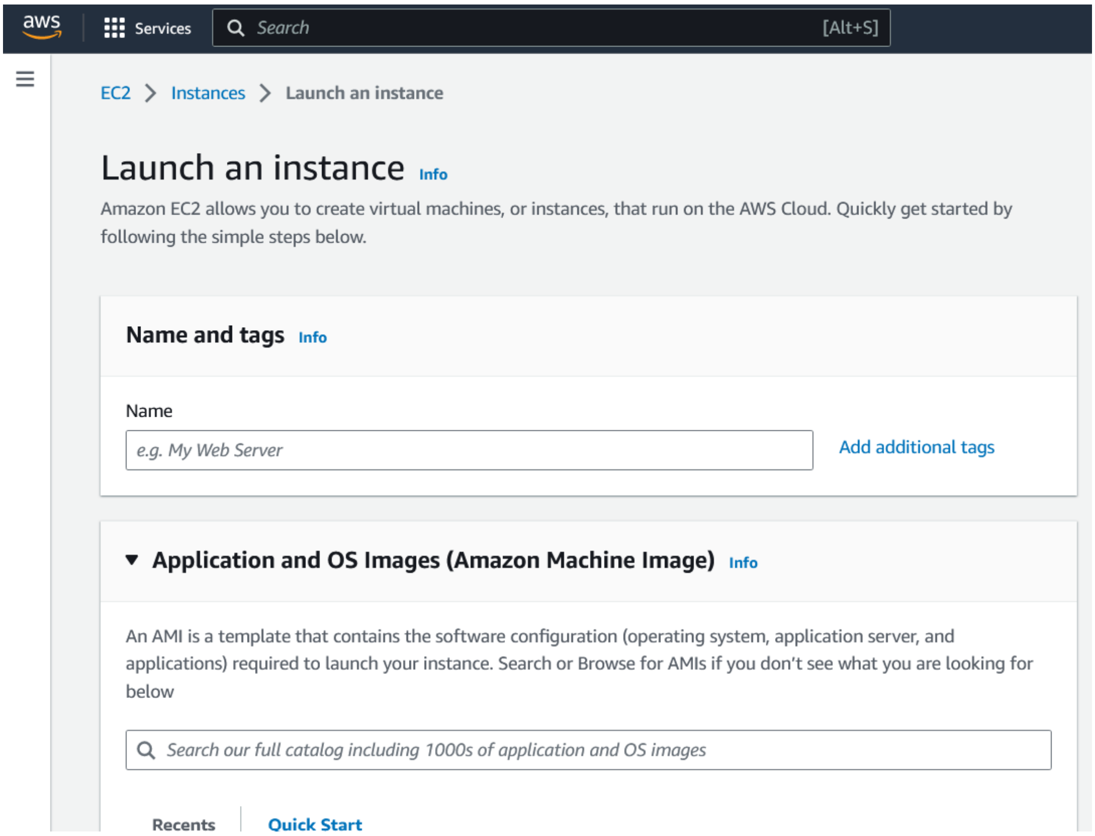
Find instance by attribute or tag (case-sensitive)

Instance state = running   Clear filters

□	Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone	Public IPv4 DNS
□	micro	i-0e15a3b0407f73b60	✓ Running	t2.micro	✓ 2/2 checks passed	No alarms +	us-east-1d	ec2-52-205-254-
□	Project_IoT	i-01cdc915d408d573c	✓ Running	t2.micro	✓ 2/2 checks passed	No alarms +	us-east-1a	ec2-184-73-79-1
□	project	i-02837b03614140fe2	✓ Running	t2.micro	✓ 2/2 checks passed	No alarms +	us-east-1a	ec2-18-215-143-

Select an instance

Click on the yellow "Launch instances" button on the right top corner.



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 Info

Name

e.g. My Web Server

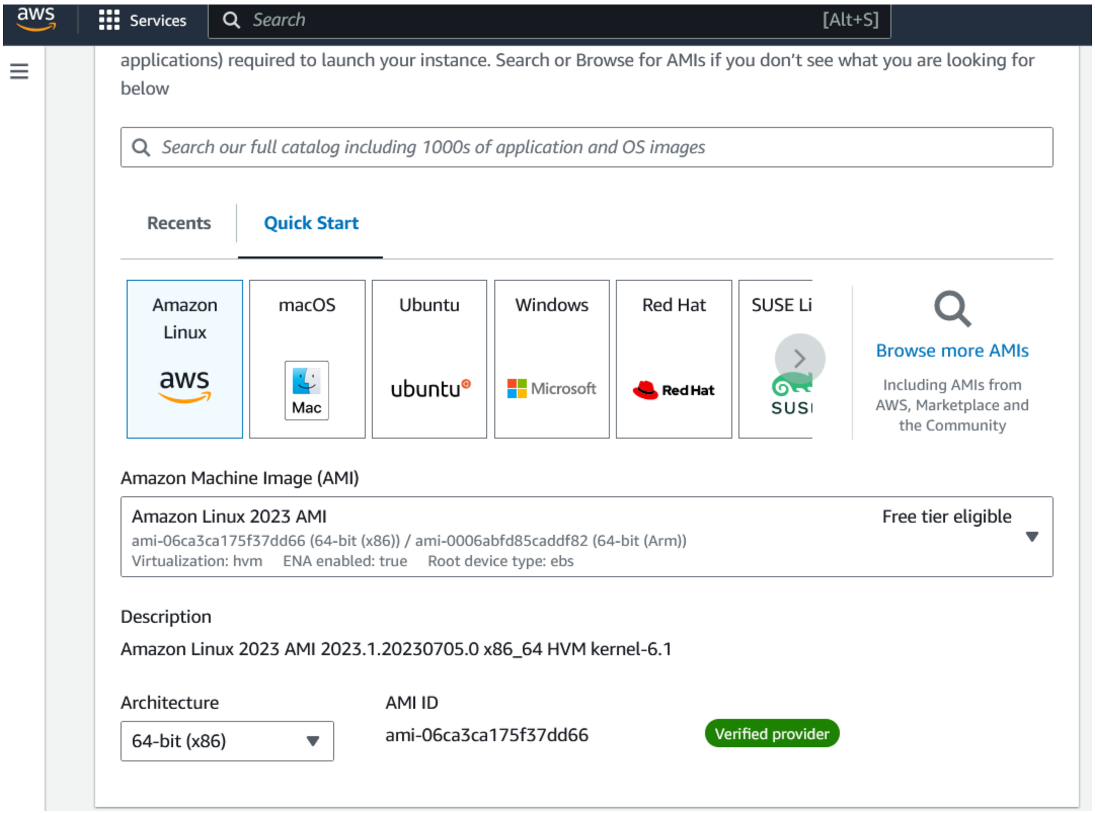
Add additional tags

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

Search our full catalog including 1000s of application and OS images

Give the instance a name



applications) required to launch your instance. Search or Browse for AMIs if you don't see what you are looking for below

Search our full catalog including 1000s of application and OS images

Amazon Machine Image (AMI)

Amazon Linux 2023 AMI — Free tier eligible
ami-06ca3ca175f37dd66 (64-bit (x86)) / ami-0006abfd85caddf82 (64-bit (Arm))
Virtualization: hvm ENA enabled: true Root device type: ebs

Description

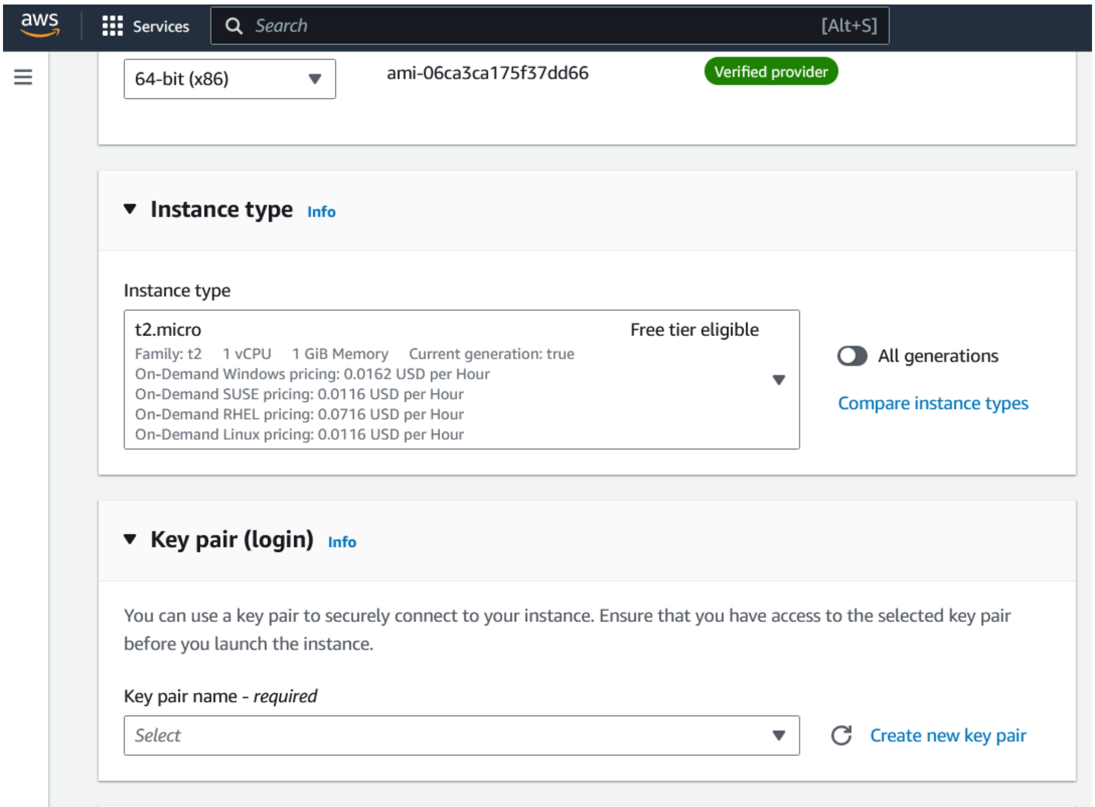
Amazon Linux 2023 AMI 2023.1.20230705.0 x86_64 HVM kernel-6.1

Architecture: 64-bit (x86)

AMI ID: ami-06ca3ca175f37dd66

Verified provider

Select a Amazon Machine Image (AMI). For our project we used ubuntu. Click the ubuntu option. Leave the Server and the Architecture as it is.



64-bit (x86) ami-06ca3ca175f37dd66 Verified provider

▼ Instance type Info

Instance type

t2.micro Free tier eligible
Family: t2 1 vCPU 1 GiB Memory Current generation: true
On-Demand Windows pricing: 0.0162 USD per Hour
On-Demand SUSE pricing: 0.0116 USD per Hour
On-Demand RHEL pricing: 0.0716 USD per Hour
On-Demand Linux pricing: 0.0116 USD per Hour

All generations

Compare instance types

▼ Key pair (login) Info

You can use a key pair to securely connect to your instance. Ensure that you have access to the selected key pair before you launch the instance.

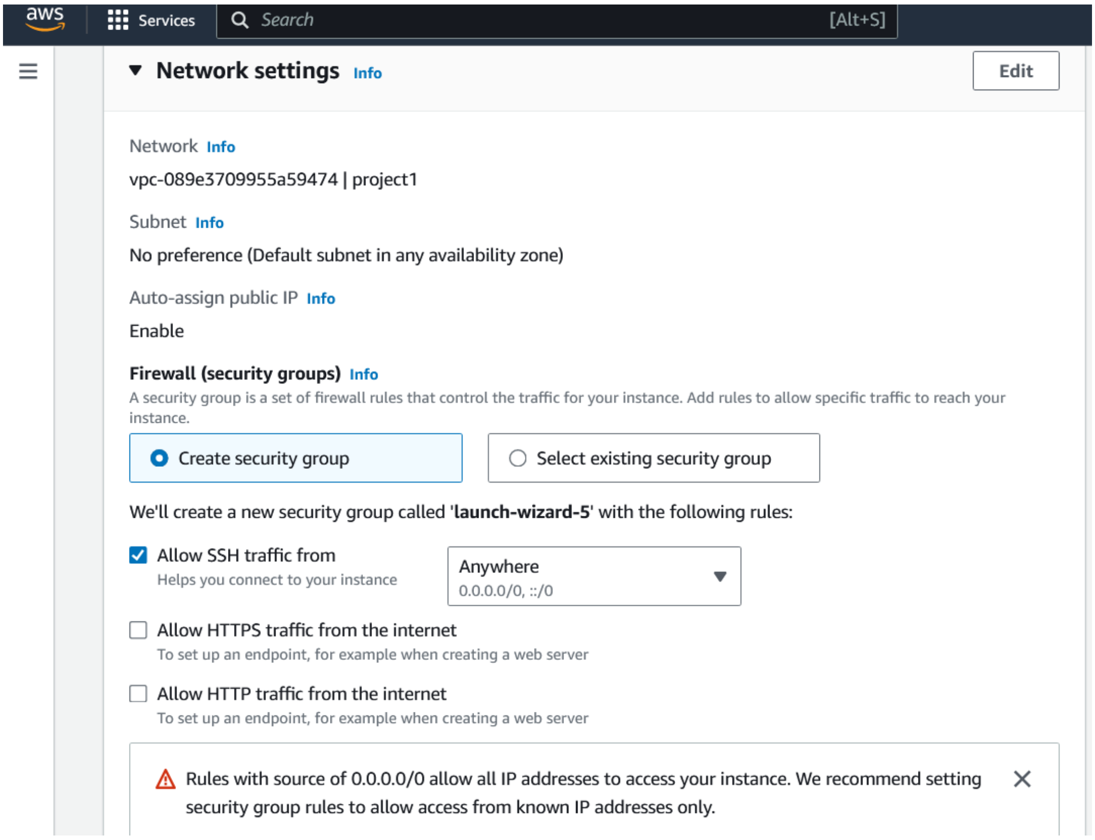
Key pair name - *required*

Select

Create new key pair

For the instance type, select the Free tier eligible version. Which is the t2.micro.

If you don't have a Key pair, click on "Create a new key pair" on the right side. In the Window that will open, give the key pair a name. Select RSA for the key pair type. For the private file formate we used .pem so we could use OpenSSH. If PuTTY is the preferred to use, choose .ppk Click on the yellow button with the text "Create key pair" on it. Choose a place to safe your key. Choose the Key pair you created.



▼ Network settings Info

Network Info

vpc-089e3709955a59474 | project1

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

We'll create a new security group called '**launch-wizard-5**' with the following rules:

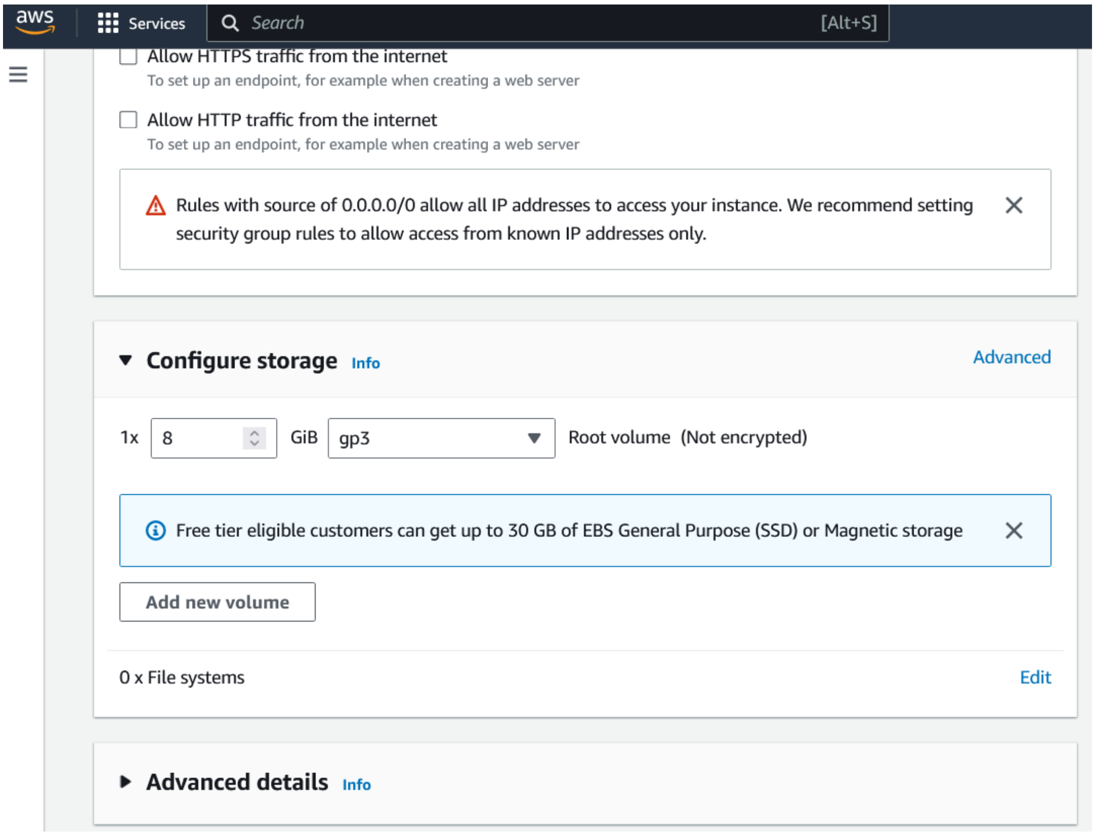
☑ Allow SSH traffic from — Anywhere 0.0.0.0/0, ::/0
Helps you connect to your instance

□ Allow HTTPS traffic from the internet
To set up an endpoint, for example when creating a web server

□ Allow HTTP traffic from the internet
To set up an endpoint, for example when creating a web server

⚠ 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.

Leave the Network Settings as they are.



□ Allow HTTPS traffic from the internet
To set up an endpoint, for example when creating a web server

□ Allow HTTP traffic from the internet
To set up an endpoint, for example when creating a web server

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.

▼ Configure storage

1x 8 GiB gp3 Root volume (Not encrypted)

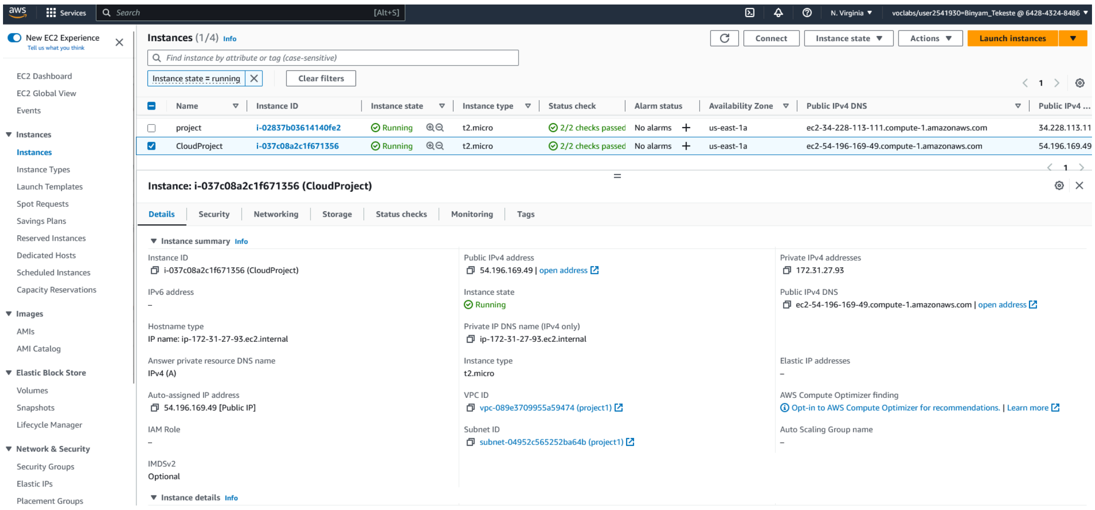
Free tier eligible customers can get up to 30 GB of EBS General Purpose (SSD) or Magnetic storage

0 x File systems

▶ Advanced details

Set the configure storage to 30 GiB Scroll down. Check the summary and click on the yellow button ("Launch instance") on the lower right corner.

The Instance will be shown under Instances after a little while. Under Status checked it might be shown as Initializing. After a while it will turn to 2/2 checks passed.



Instances (1/4)

Find instance by attribute or tag (case-sensitive)

Instance state = running

	Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone	Public IPv4 DNS	Public IPv4 ...
□	project	i-02837b03614140fe2	Running	t2.micro	2/2 checks passed	No alarms	us-east-1a	ec2-34-228-113-111.compute-1.amazonaws.com	34.228.113.11
☑	CloudProject	i-037c08a2c1f671356	Running	t2.micro	2/2 checks passed	No alarms	us-east-1a	ec2-54-196-169-49.compute-1.amazonaws.com	54.196.169.49

Instance: i-037c08a2c1f671356 (CloudProject)

▼ Instance summary

Instance ID
i-037c08a2c1f671356 (CloudProject)

IPv6 address
–

Hostname type
IP name: ip-172-31-27-93.ec2.internal

Answer private resource DNS name
IPv4 (A)

Auto-assigned IP address
54.196.169.49 [Public IP]

IAM Role
–

IMDSv2
Optional

Public IPv4 address
54.196.169.49 | open address

Instance state
Running

Private IP DNS name (IPv4 only)
ip-172-31-27-93.ec2.internal

Instance type
t2.micro

VPC ID
vpc-089e3709955a59474 (project1)

Subnet ID
subnet-04952c565252ba64b (project1)

Private IPv4 addresses
172.31.27.93

Public IPv4 DNS
ec2-54-196-169-49.compute-1.amazonaws.com | open address

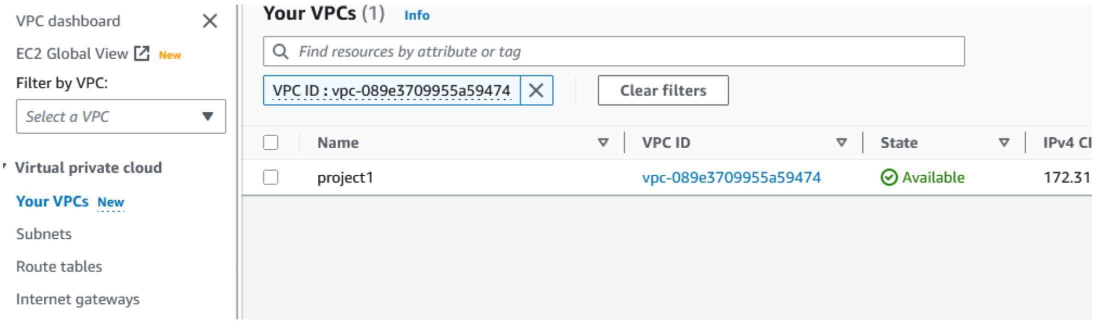
Elastic IP addresses
–

AWS Compute Optimizer finding
Opt-in to AWS Compute Optimizer for recommendations. | Learn more

Auto Scaling Group name
–

▼ Instance details

Click the checkbox next to the instance name. In the lower part of the screen you will find the instance summary. Search for VPC ID and click on the ID.

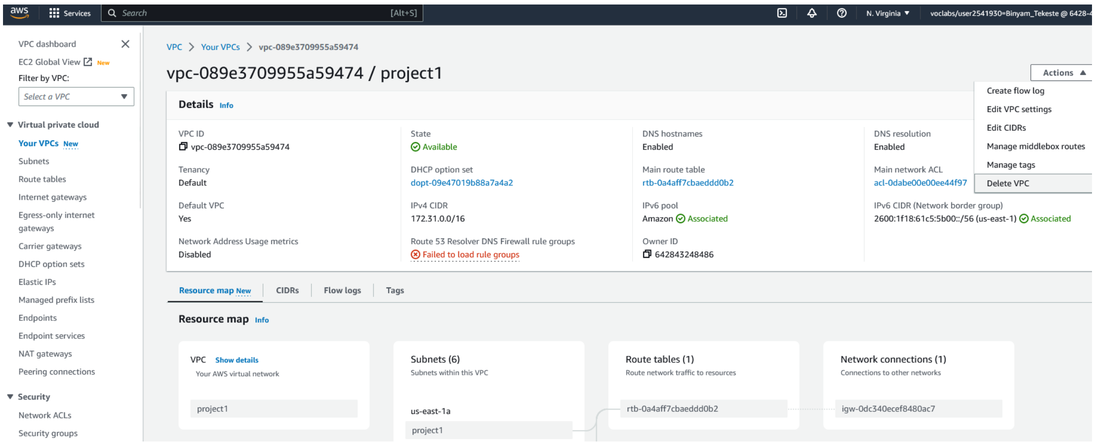


Your VPCs (1)

VPC ID : vpc-089e3709955a59474

	Name	VPC ID	State	IPv4 CI
□	project1	vpc-089e3709955a59474	✓ Available	172.31

On this page, click again on the VPC ID.



vpc-089e3709955a59474 / project1

Details

VPC ID: vpc-089e3709955a59474
State: Available
DNS hostnames: Enabled
DNS resolution: Enabled
Tenancy: Default
DHCP option set: dopt-09e47019b88a7a4a2
Main route table: rtb-0a4aff7cbaeddd0b2
Main network ACL: acl-0dabe00e00ee44f97
Default VPC: Yes
IPv4 CIDR: 172.31.0.0/16
IPv6 pool: Amazon Associated
IPv6 CIDR (Network border group): 2600:1f18:61c5:5b00::/56 (us-east-1) Associated
Network Address Usage metrics: Disabled
Route 53 Resolver DNS Firewall rule groups: Failed to load rule groups
Owner ID: 642843248486

Resource map

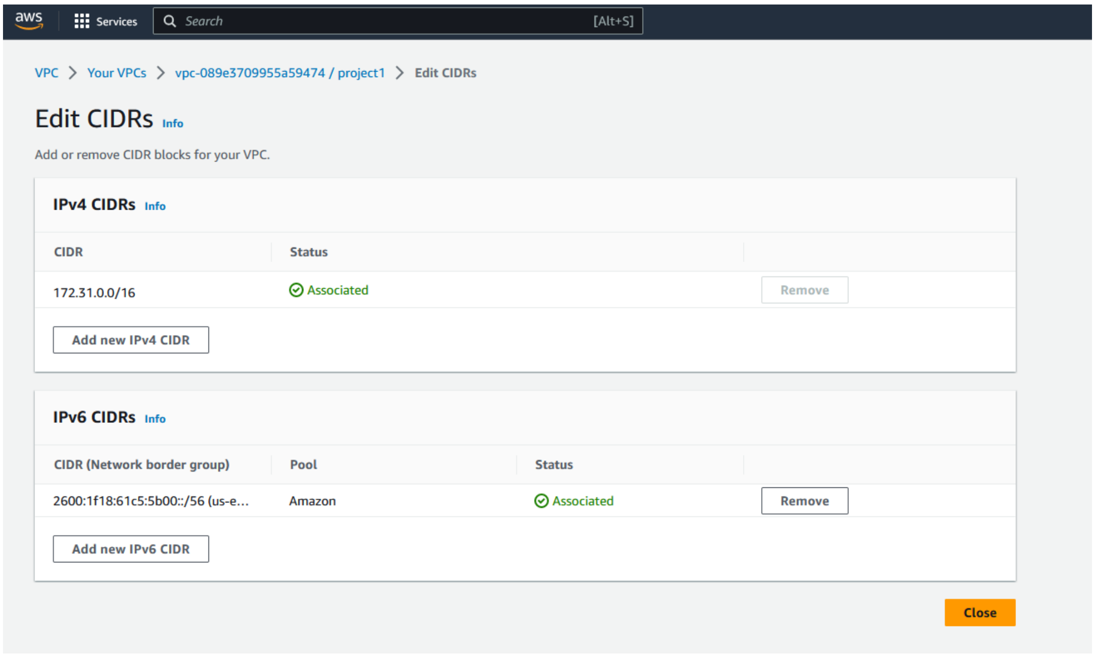
VPC Show details
Your AWS virtual network
project1

Subnets (6)
Subnets within this VPC
us-east-1a
project1

Route tables (1)
Route network traffic to resources
rtb-0a4aff7cbaeddd0b2

Network connections (1)
Connections to other networks
igw-0dc340ecef8480ac7

Click on the "Actions" button on the upper right corner and choose "Edit CIDRs".



Edit CIDRs

Add or remove CIDR blocks for your VPC.

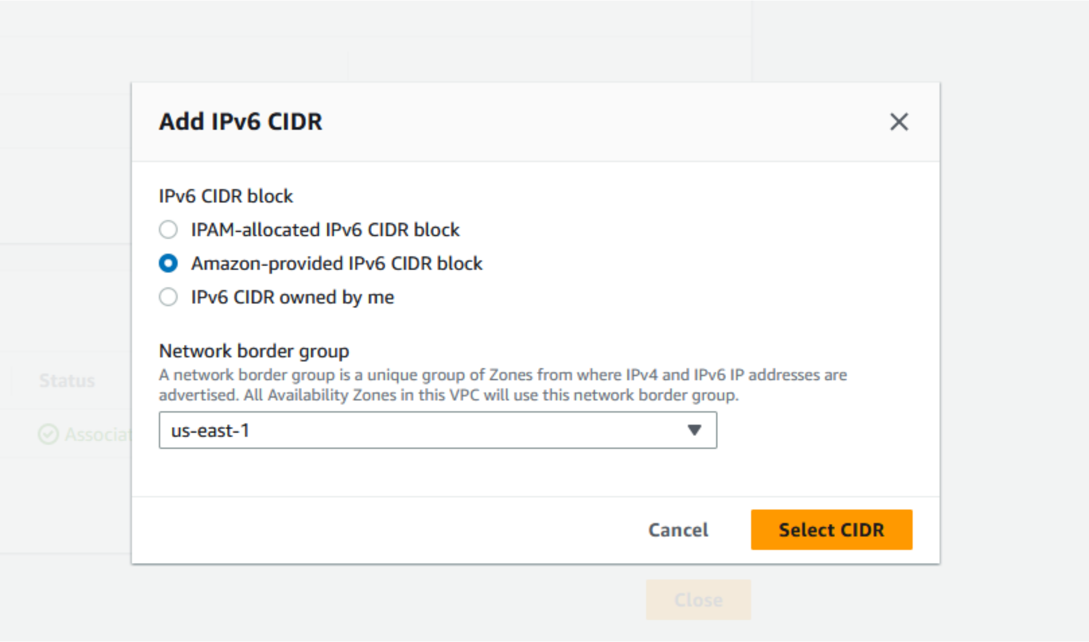
IPv4 CIDRs

CIDR	Status	
172.31.0.0/16	✓ Associated	Remove

IPv6 CIDRs

CIDR (Network border group)	Pool	Status	
2600:1f18:61c5:5b00::/56 (us-e...	Amazon	✓ Associated	Remove

Click on "Add new IPv6 CIDR



Add IPv6 CIDR

IPv6 CIDR block

○ IPAM-allocated IPv6 CIDR block

● Amazon-provided IPv6 CIDR block

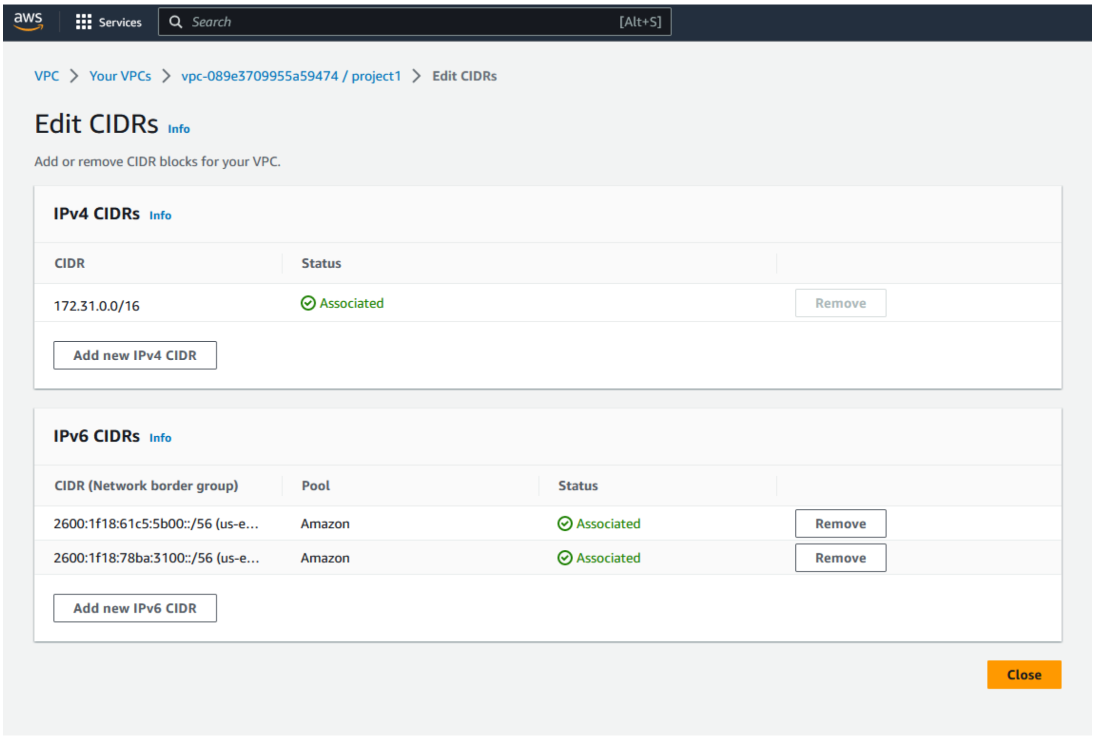
○ IPv6 CIDR owned by me

Network border group

A network border group is a unique group of Zones from where IPv4 and IPv6 IP addresses are advertised. All Availability Zones in this VPC will use this network border group.

us-east-1

Select "Amazon-provided IPv6 CIDR block and click on "Select CIDR".



VPC > Your VPCs > vpc-089e3709955a59474 / project1 > Edit CIDRs

Edit CIDRs Info

Add or remove CIDR blocks for your VPC.

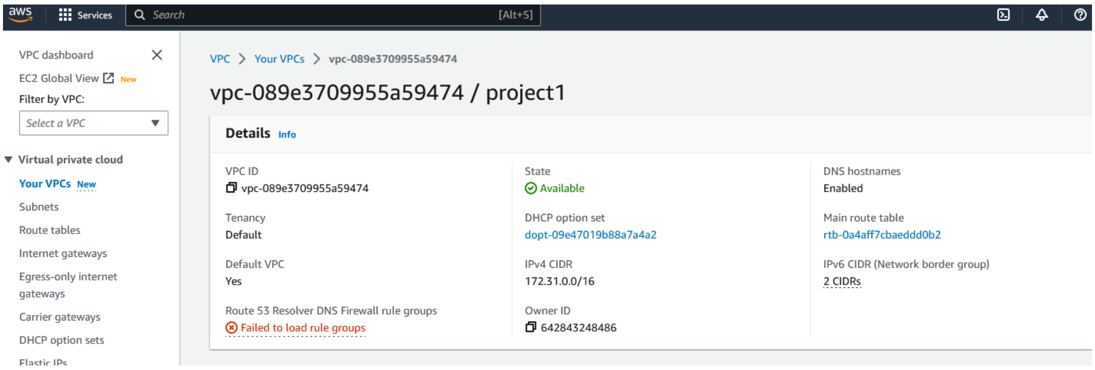
IPv4 CIDRs Info

CIDR	Status	
172.31.0.0/16	Associated	Remove

IPv6 CIDRs Info

CIDR (Network border group)	Pool	Status	
2600:1f18:61c5:5b00::/56 (us-e...	Amazon	Associated	Remove
2600:1f18:78ba:3100::/56 (us-e...	Amazon	Associated	Remove

You should now see the status "Associated". Click on the VPC ID on the top to see details to your VPC ID.



VPC > Your VPCs > vpc-089e3709955a59474

vpc-089e3709955a59474 / project1

Details Info

VPC ID: vpc-089e3709955a59474

State: Available

DNS hostnames: Enabled

Tenancy: Default

DHCP option set: dopt-09e47019b88a7a4a2

Main route table: rtb-0a4aff7cbaeddd0b2

Default VPC: Yes

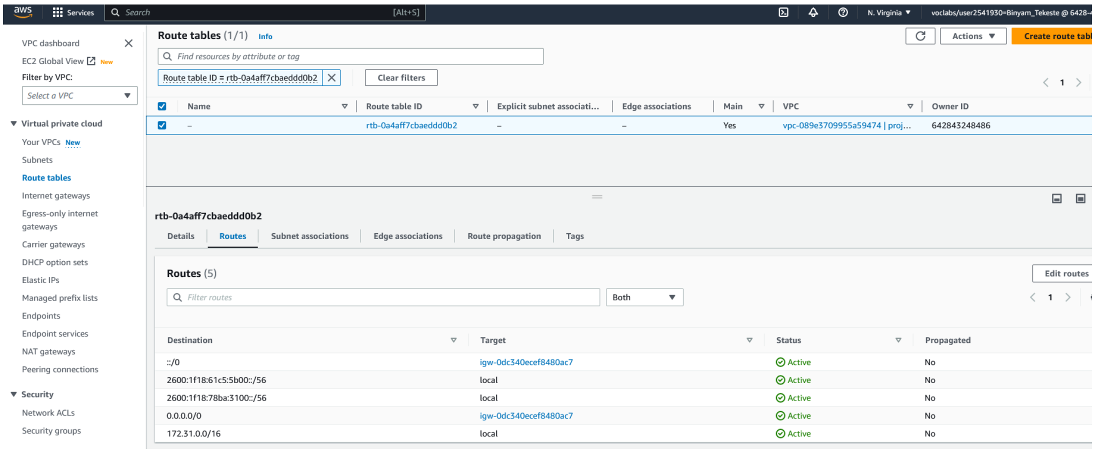
IPv4 CIDR: 172.31.0.0/16

IPv6 CIDR (Network border group): 2 CIDRs

Route 53 Resolver DNS Firewall rule groups: Failed to load rule groups

Owner ID: 642843248486

Here you should see your IPv6. Click on Route tables on the left side navigation bar.



Route tables (1/1) Info

Find resources by attribute or tag

Route table ID = rtb-0a4aff7cbaeddd0b2

☑	Name	Route table ID	Explicit subnet associati...	Edge associations	Main	VPC	Owner ID
☑	–	rtb-0a4aff7cbaeddd0b2	–	–	Yes	vpc-089e3709955a59474 | proj...	642843248486

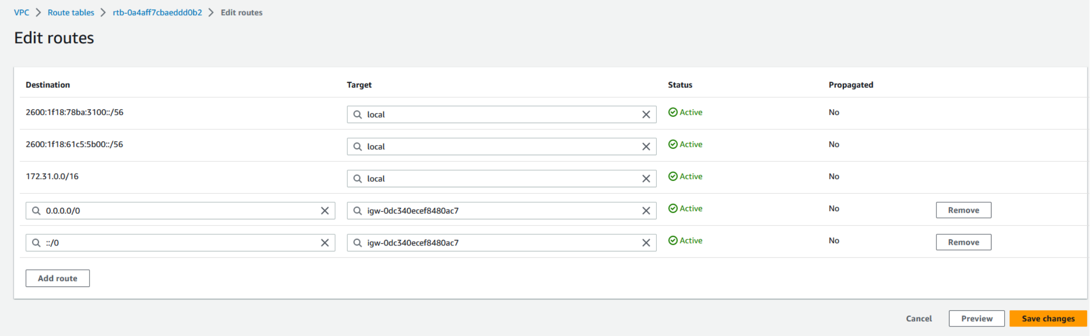
rtb-0a4aff7cbaeddd0b2

Routes (5)

Filter routes

Destination	Target	Status	Propagated
::/0	igw-0dc340ecef8480ac7	Active	No
2600:1f18:61c5:5b00::/56	local	Active	No
2600:1f18:78ba:3100::/56	local	Active	No
0.0.0.0/0	igw-0dc340ecef8480ac7	Active	No
172.31.0.0/16	local	Active	No

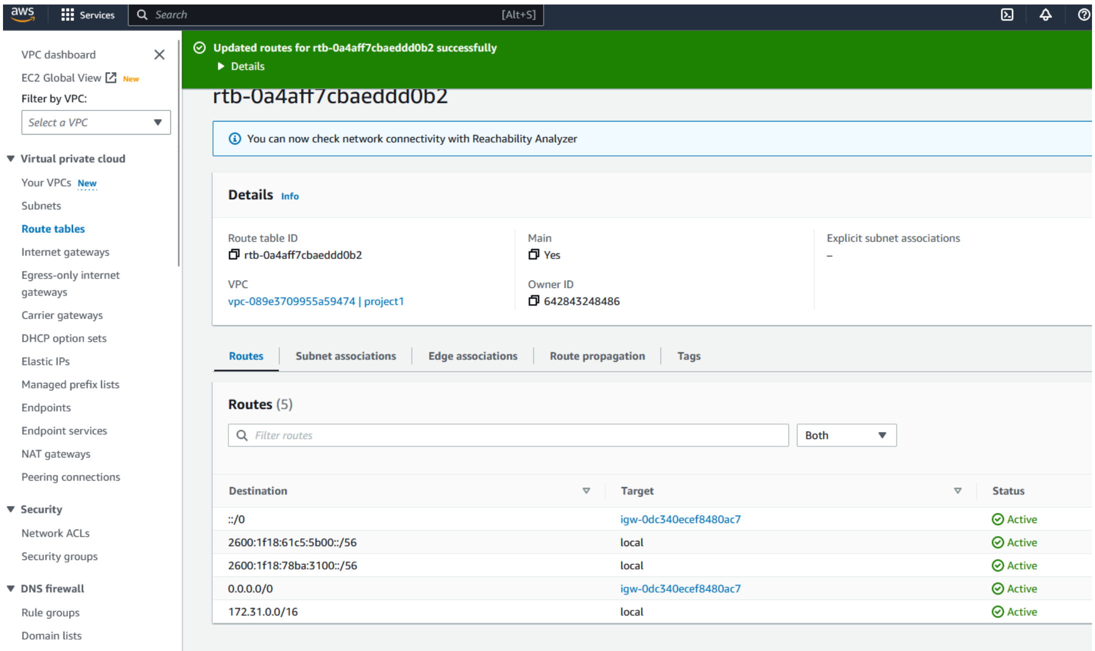
In the lower part select the tier with the name "Routes" and click the button "Edit routes" on the right side.



Edit routes

Destination	Target	Status	Propagated
2600:1f18:78ba:3100::/56	local	Active	No
2600:1f18:61c5:5b00::/56	local	Active	No
172.31.0.0/16	local	Active	No
0.0.0.0/0	igw-0dc340ecef8480ac7	Active	No
::/0	igw-0dc340ecef8480ac7	Active	No

Click "Add route" for the destination use ::/0 and for the target use your amazon default value. Click "Add route again and use the destination 0.0.0.0./0 if it is not already set. use the same default value as a target. Save the changes with the click on the yellow button. Click on "Route tables" at the top.



Updated routes for rtb-0a4aff7cbaeddd0b2 successfully

Details

rtb-0a4aff7cbaeddd0b2

You can now check network connectivity with Reachability Analyzer

Details Info

Route table ID: rtb-0a4aff7cbaeddd0b2

Main: Yes

Explicit subnet associations: –

VPC: vpc-089e3709955a59474 | project1

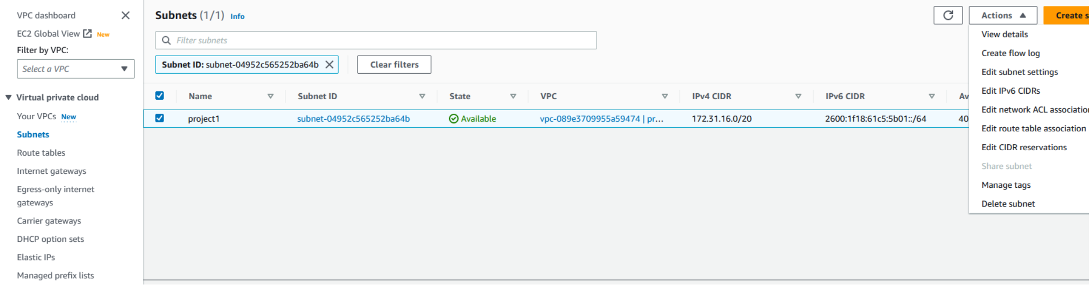
Owner ID: 642843248486

Routes (5)

Filter routes

Destination	Target	Status
::/0	igw-0dc340ecef8480ac7	Active
2600:1f18:61c5:5b00::/56	local	Active
2600:1f18:78ba:3100::/56	local	Active
0.0.0.0/0	igw-0dc340ecef8480ac7	Active
172.31.0.0/16	local	Active

Our new routes are now shown in the lower part. Click on Subnets

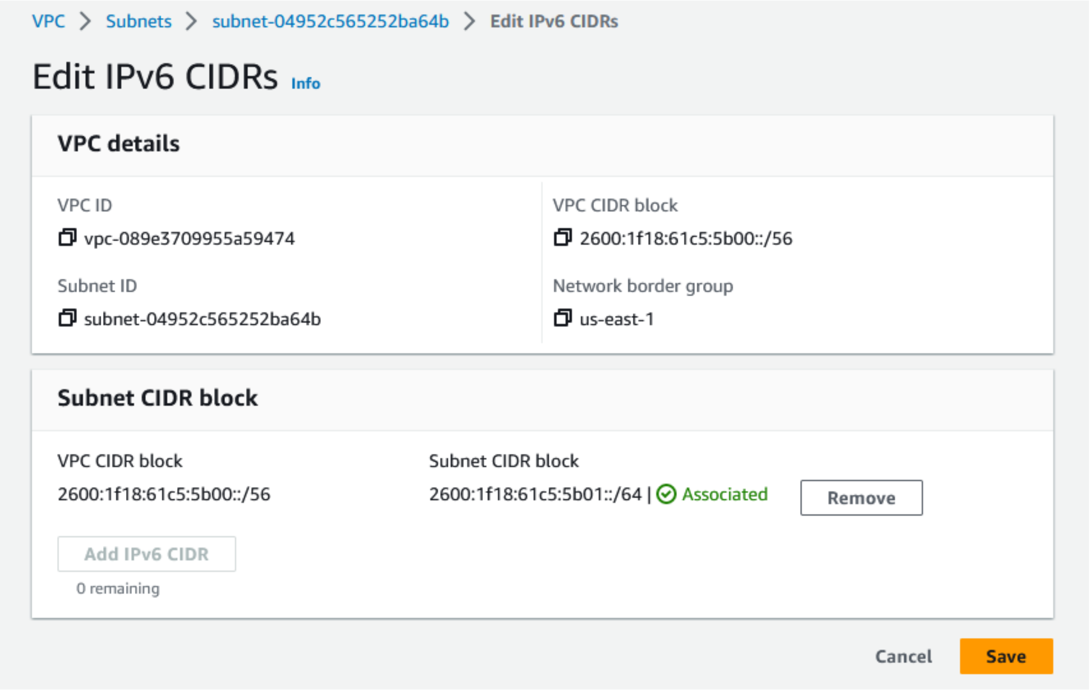


Subnets (1/1) Info

Subnet ID: subnet-04952c565252ba64b

☑	Name	Subnet ID	State	VPC	IPv4 CIDR	IPv6 CIDR	Av
☑	project1	subnet-04952c565252ba64b	✓ Available	vpc-089e3709955a59474 | pr...	172.31.16.0/20	2600:1f18:61c5:5b01::/64	40

Click the checkbox next to the name. Click on the "Action" button on the upper right corner and select "Edit IPv6 CIDRs.



Edit IPv6 CIDRs

VPC details

VPC ID
vpc-089e3709955a59474

VPC CIDR block
2600:1f18:61c5:5b00::/56

Subnet ID
subnet-04952c565252ba64b

Network border group
us-east-1

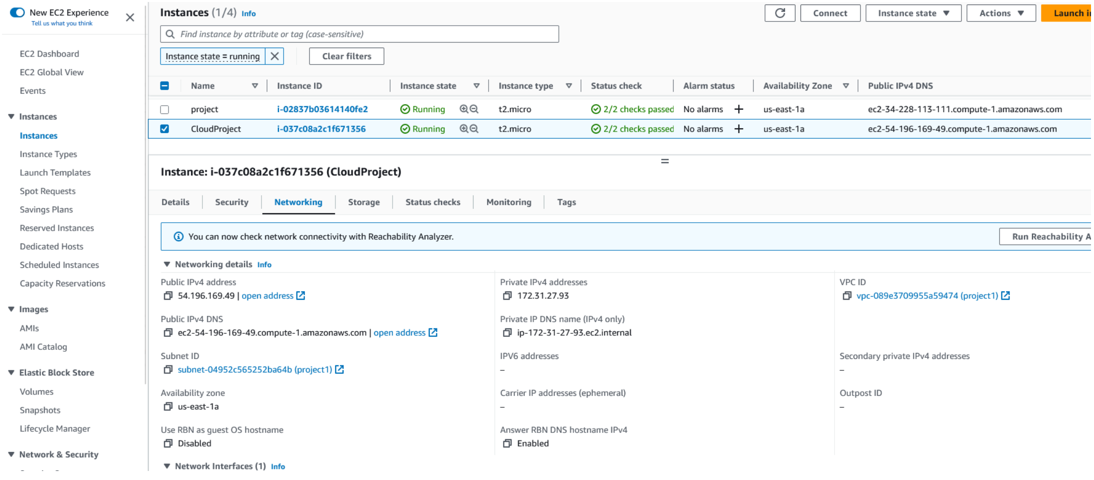
Subnet CIDR block

VPC CIDR block
2600:1f18:61c5:5b00::/56

Subnet CIDR block
2600:1f18:61c5:5b01::/64 | ✓ Associated

0 remaining

If the Subnet CIDR block is not already set, Click on "Add IPv6 CIDR" and save the changes by clicking the yellow button on the right lower corner. Then go back to your instances screen.



Instances (1/4) Info

Instance state = running

☐	Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone	Public IPv4 DNS
☐	project	i-02837b03614140fe2	Running	t2.micro	2/2 checks passed	No alarms	us-east-1a	ec2-34-228-113-111.compute-1.amazonaws.com
☑	CloudProject	i-037c08a2c1f671356	Running	t2.micro	2/2 checks passed	No alarms	us-east-1a	ec2-54-196-169-49.compute-1.amazonaws.com

Instance: i-037c08a2c1f671356 (CloudProject)

You can now check network connectivity with Reachability Analyzer.

Networking details Info

Public IPv4 address
54.196.169.49 | open address

Private IPv4 addresses
172.31.27.93

VPC ID
vpc-089e3709955a59474 (project1)

Public IPv4 DNS
ec2-54-196-169-49.compute-1.amazonaws.com | open address

Private IP DNS name (IPv4 only)
ip-172-31-27-93.ec2.internal

Subnet ID
subnet-04952c565252ba64b (project1)

IPV6 addresses
–

Secondary private IPv4 addresses
–

Availability zone
us-east-1a

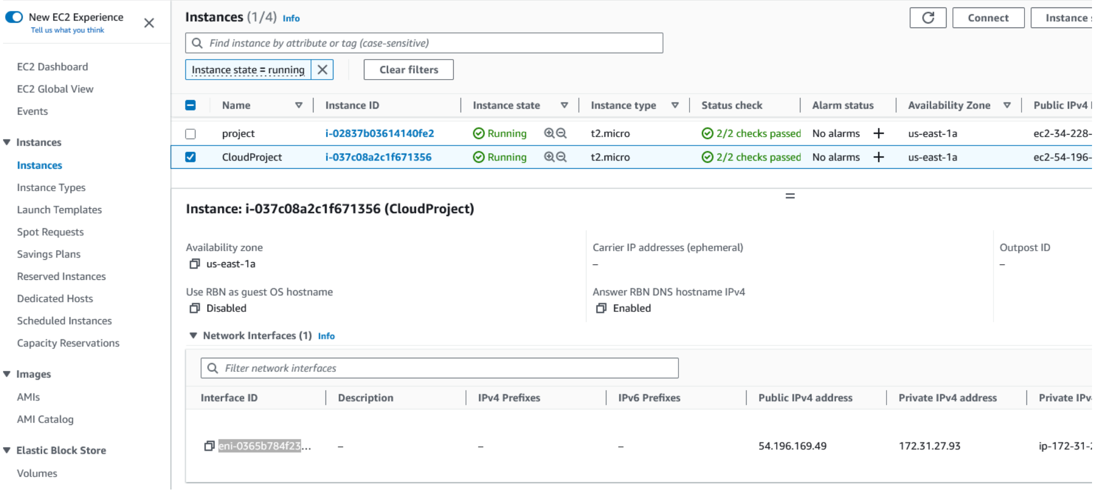
Carrier IP addresses (ephemeral)
–

Outpost ID
–

Use RBN as guest OS hostname
Disabled

Answer RBN DNS hostname IPv4
Enabled

Network Interfaces (1) Info



Instances (1/4) Info

Instance state = running

☐	Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone	Public IPv4
☐	project	i-02837b03614140fe2	Running	t2.micro	2/2 checks passed	No alarms	us-east-1a	ec2-34-228-
☑	CloudProject	i-037c08a2c1f671356	Running	t2.micro	2/2 checks passed	No alarms	us-east-1a	ec2-54-196-

Instance: i-037c08a2c1f671356 (CloudProject)

Availability zone
us-east-1a

Carrier IP addresses (ephemeral)
–

Outpost ID
–

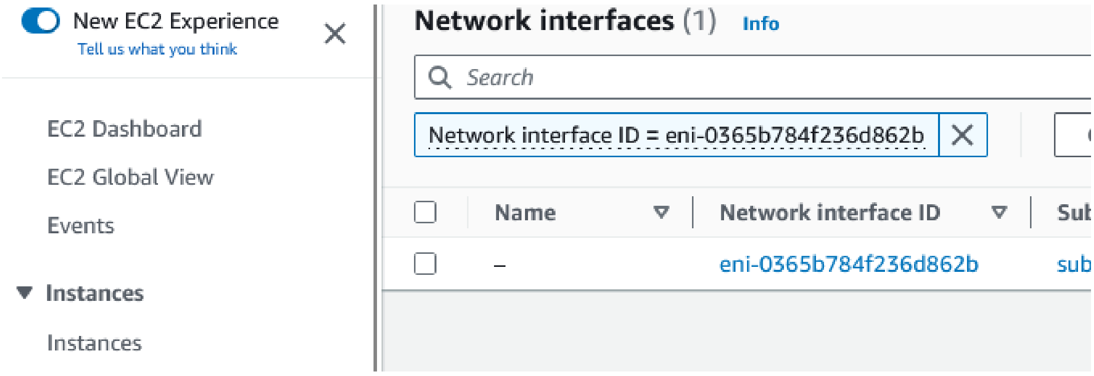
Use RBN as guest OS hostname
Disabled

Answer RBN DNS hostname IPv4
Enabled

Network Interfaces (1) Info

Interface ID	Description	IPv4 Prefixes	IPv6 Prefixes	Public IPv4 address	Private IPv4 address	Private IPv
eni-0365b784f23...	–	–	–	54.196.169.49	172.31.27.93	ip-172-31-

Click the checkbox next to your instance and scroll further down on the lower part, till you see your Network interface click on the Interface ID.

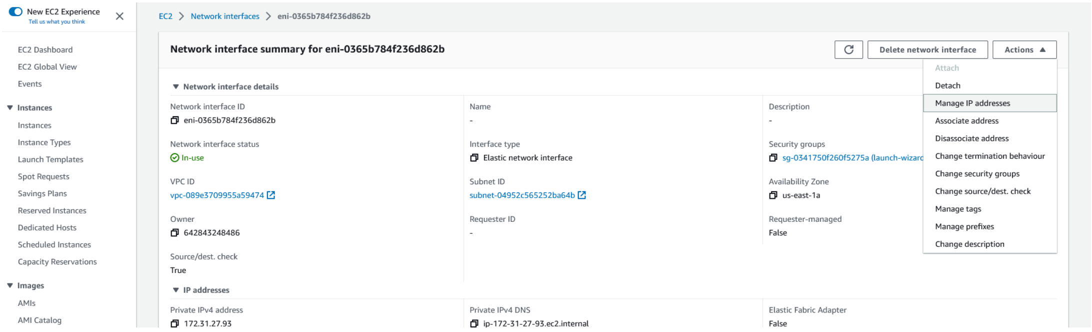


Network interfaces (1) Info

Network interface ID = eni-0365b784f236d862b

☐	Name	Network interface ID	Sub
☐	–	eni-0365b784f236d862b	sub

Click the Network Interface ID again.



EC2 > Network interfaces > eni-0365b784f236d862b

Network interface summary for eni-0365b784f236d862b

Network interface details

Network interface ID
eni-0365b784f236d862b

Name
-

Description
-

Network interface status
In-use

Interface type
Elastic network interface

Security groups
sg-0341750f260f5275a (launch-wizard

VPC ID
vpc-089e3709955a59474

Subnet ID
subnet-04952c565252ba64b

Availability Zone
us-east-1a

Owner
642843248486

Requester ID
-

Requester-managed
False

Source/dest. check
True

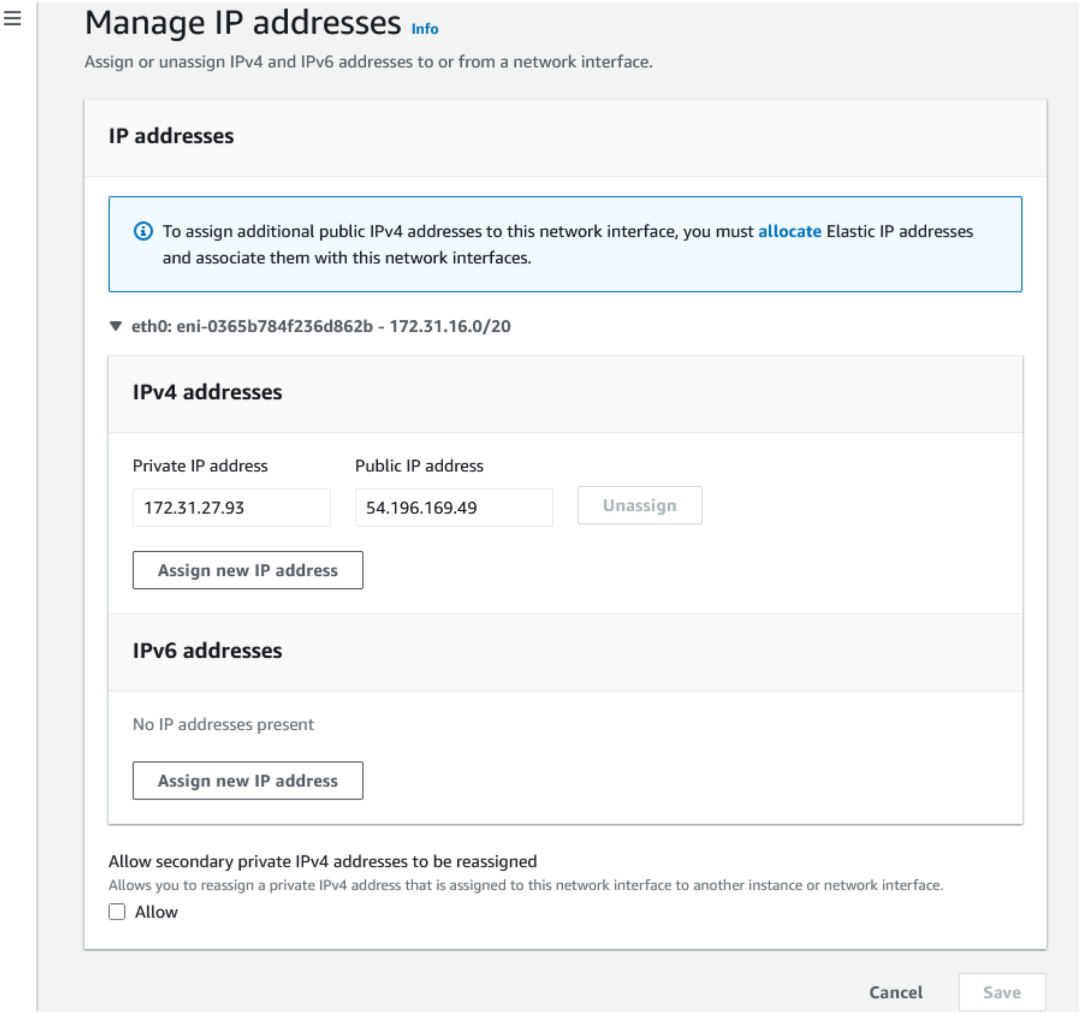
IP addresses

Private IPv4 address
172.31.27.93

Private IPv4 DNS
ip-172-31-27-93.ec2.internal

Elastic Fabric Adapter
False

Click on the "Actions" button on the upper right corner and select "Manage IP addresses".



Manage IP addresses Info

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

IP addresses

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

▼ eth0: eni-0365b784f236d862b - 172.31.16.0/20

IPv4 addresses

Private IP address: 172.31.27.93

Public IP address: 54.196.169.49

IPv6 addresses

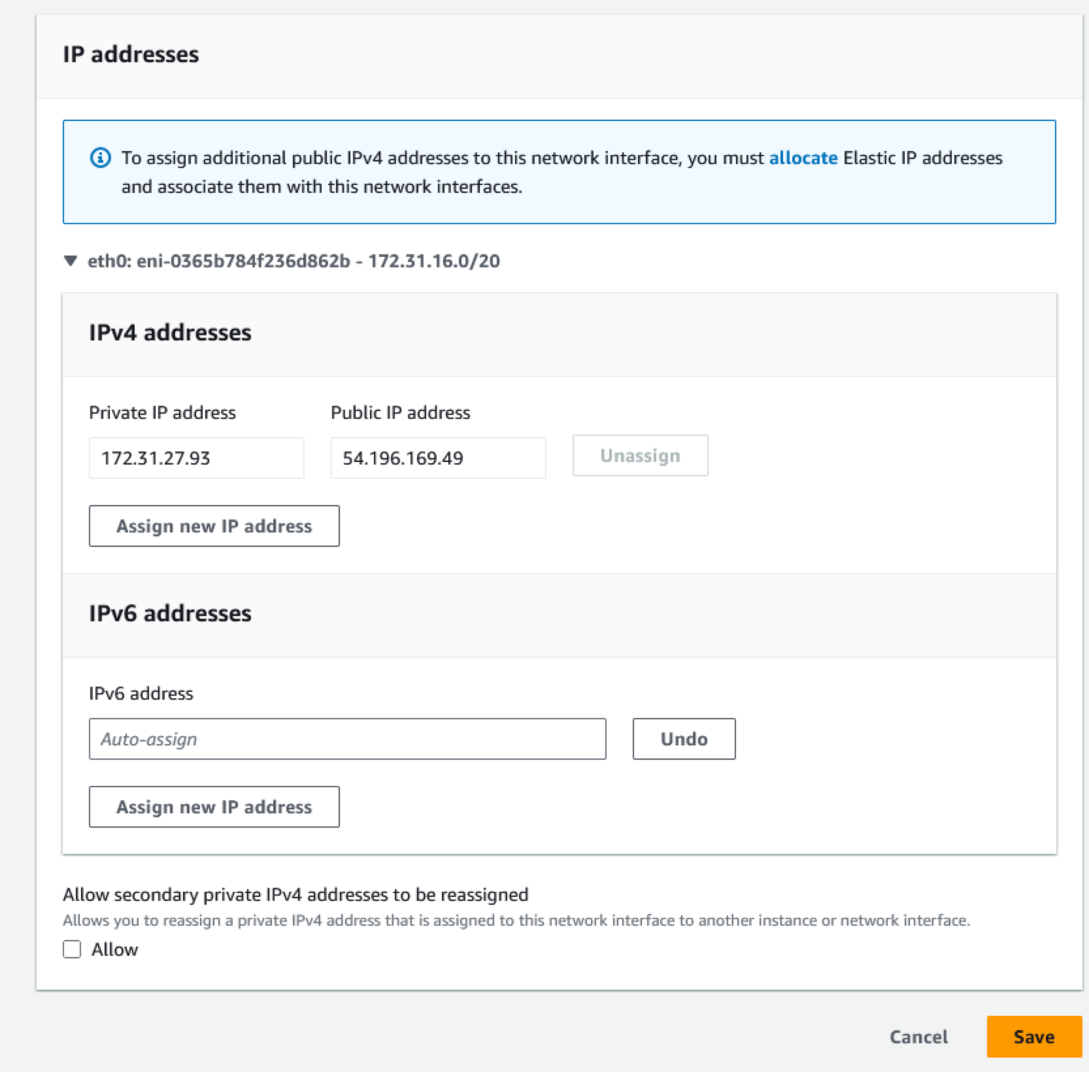
No IP addresses present

Allow secondary private IPv4 addresses to be reassigned

Allows you to reassign a private IPv4 address that is assigned to this network interface to another instance or network interface.

□ Allow

Click the button "Assign new IP address" in the IPv6 section.



IP addresses

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

▼ eth0: eni-0365b784f236d862b - 172.31.16.0/20

IPv4 addresses

Private IP address: 172.31.27.93

Public IP address: 54.196.169.49

IPv6 addresses

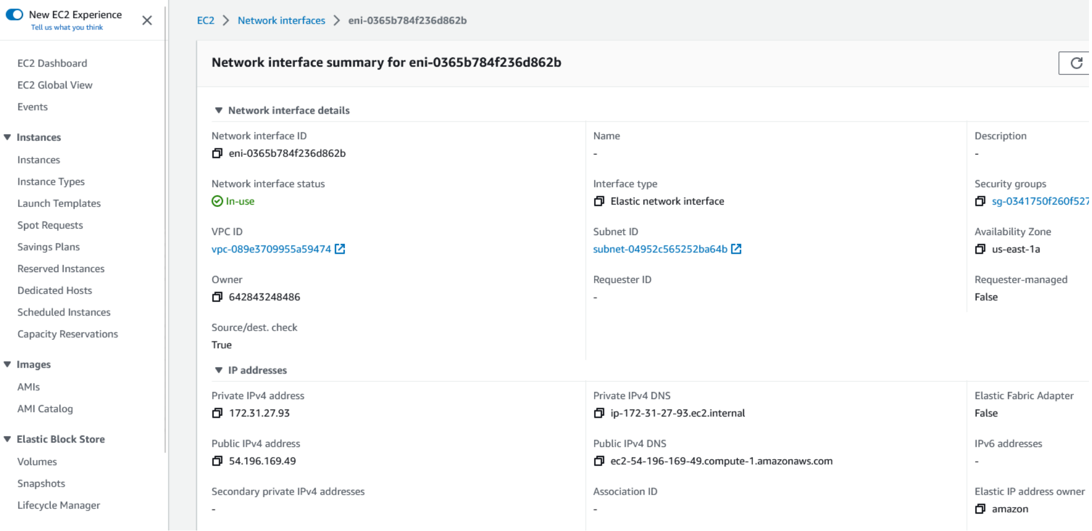
IPv6 address: *Auto-assign*

Allow secondary private IPv4 addresses to be reassigned

Allows you to reassign a private IPv4 address that is assigned to this network interface to another instance or network interface.

☐ Allow

After clicking the Assign button, leave it to Auto-assign and click the yellow "Save" button.



Network interface summary for eni-0365b784f236d862b

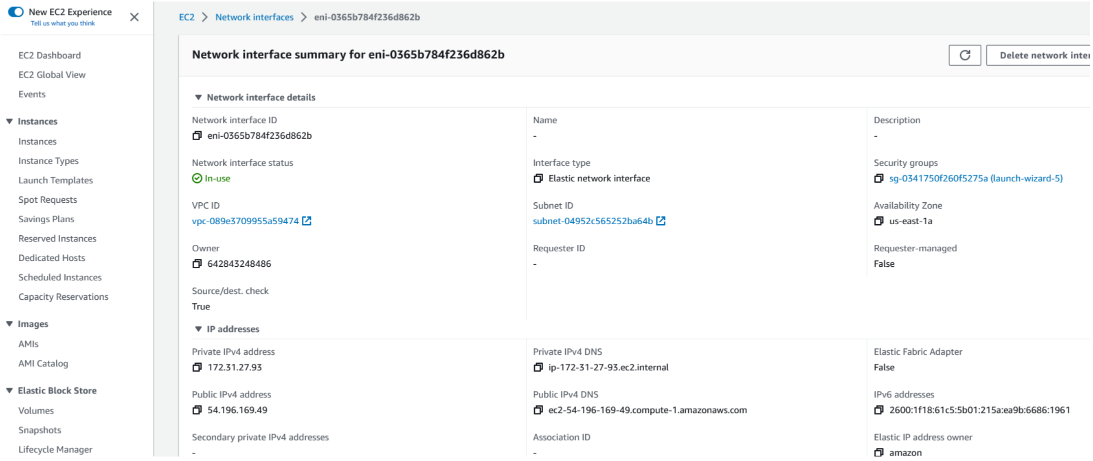
▼ Network interface details

Network interface ID eni-0365b784f236d862b	Name -	Description -
Network interface status In-use	Interface type Elastic network interface	Security groups sg-0341750f260f527
VPC ID vpc-089e3709955a59474	Subnet ID subnet-04952c565252ba64b	Availability Zone us-east-1a
Owner 642843248486	Requester ID -	Requester-managed False
Source/dest. check True		

▼ IP addresses

Private IPv4 address 172.31.27.93	Private IPv4 DNS ip-172-31-27-93.ec2.internal	Elastic Fabric Adapter False
Public IPv4 address 54.196.169.49	Public IPv4 DNS ec2-54-196-169-49.compute-1.amazonaws.com	IPv6 addresses -
Secondary private IPv4 addresses -	Association ID -	Elastic IP address owner amazon

You wil end up at the Network interface summary page. Since your IPv6 addresses is not shown, click the page refreshing button.



Network interface summary for eni-0365b784f236d862b

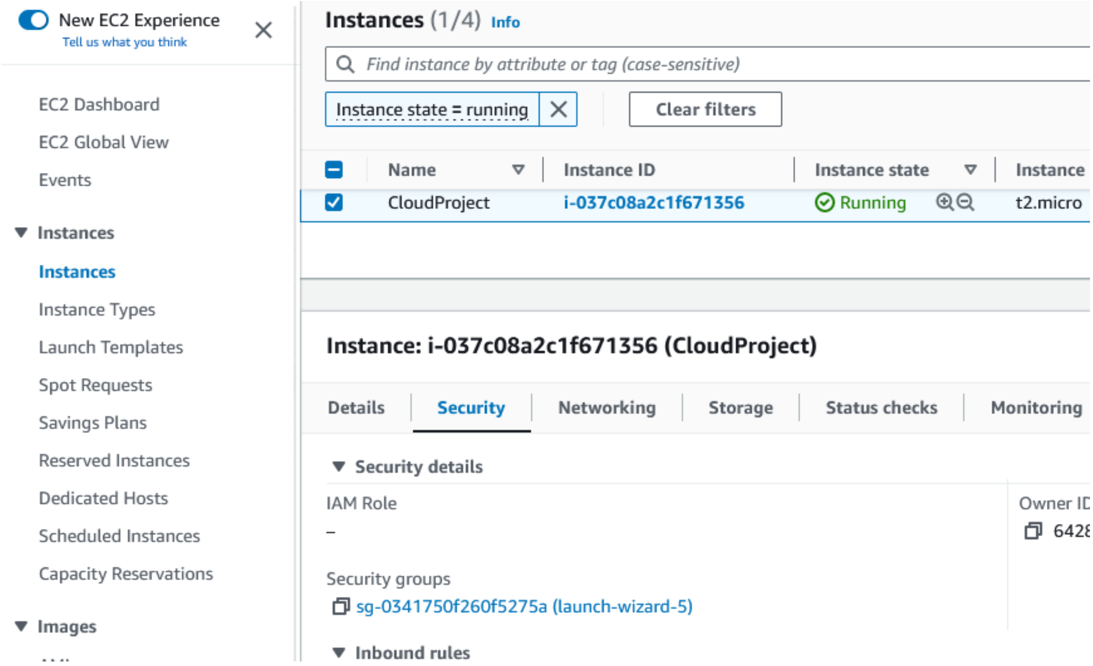
Network interface details

Network interface ID: eni-0365b784f236d862b
Name: -
Description: -
Network interface status: In-use
Interface type: Elastic network interface
Security groups: sg-0341750f260f5275a (launch-wizard-5)
VPC ID: vpc-089e3709955a59474
Subnet ID: subnet-04952c565252ba64b
Availability Zone: us-east-1a
Owner: 642843248486
Requester ID: -
Requester-managed: False
Source/dest. check: True

IP addresses

Private IPv4 address: 172.31.27.93
Private IPv4 DNS: ip-172-31-27-93.ec2.internal
Elastic Fabric Adapter: False
Public IPv4 address: 54.196.169.49
Public IPv4 DNS: ec2-54-196-169-49.compute-1.amazonaws.com
IPv6 addresses: 2600:1f18:61c5:5b01:215a:ea9b:6686:1961
Secondary private IPv4 addresses: -
Association ID: -
Elastic IP address owner: amazon

The IPv6 addresses should now be visible. Go back to your instances screen.



Instances (1/4)

Instance state = running

☐	Name	Instance ID	Instance state	Instance
☑	CloudProject	i-037c08a2c1f671356	Running	t2.micro

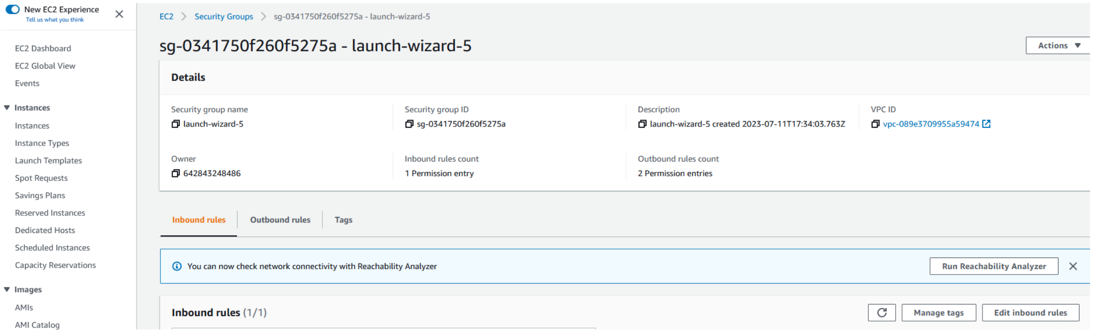
Instance: i-037c08a2c1f671356 (CloudProject)

Security details

IAM Role: –
Owner ID: 6428
Security groups: sg-0341750f260f5275a (launch-wizard-5)

Inbound rules

Click the checkbox next to your instance. On the lower part, click on the tier "Security". Click the link under "Security groups".



sg-0341750f260f5275a - launch-wizard-5

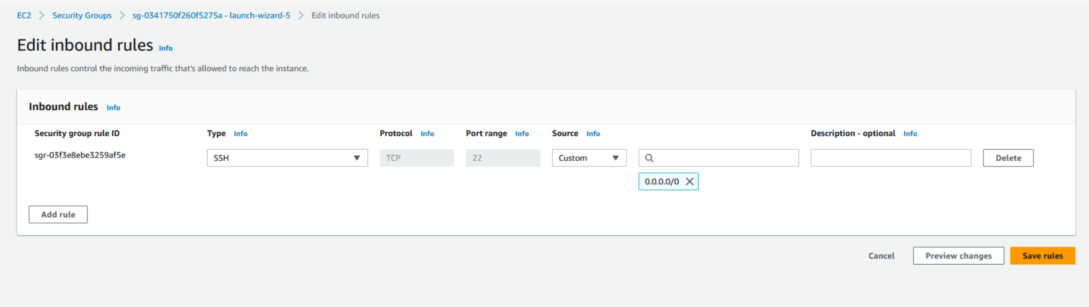
Details

Security group name: launch-wizard-5
Security group ID: sg-0341750f260f5275a
Description: launch-wizard-5 created 2023-07-11T17:34:03.763Z
VPC ID: vpc-089e3709955a59474
Owner: 642843248486
Inbound rules count: 1 Permission entry
Outbound rules count: 2 Permission entries

You can now check network connectivity with Reachability Analyzer

Inbound rules (1/1)

On the lower part, stay on the tier "Inbound rules" and click the button on the right "Edit inbound rules.



EC2 > Security Groups > sg-0341750f260f5275a - launch-wizard-5 > Edit inbound rules

Edit inbound rules

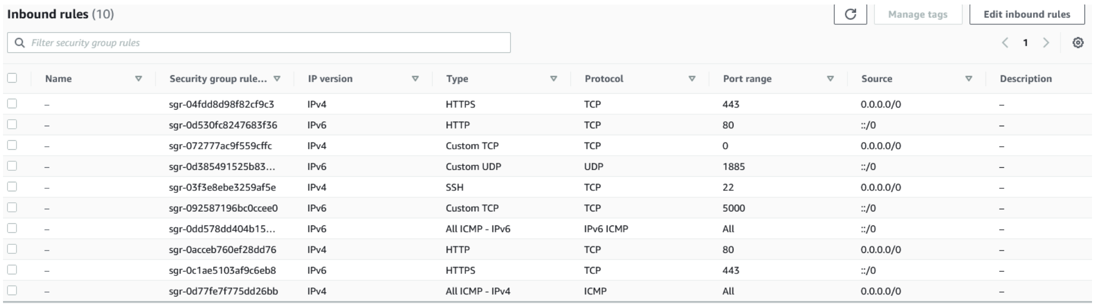
Inbound rules control the incoming traffic that's allowed to reach the instance.

Inbound rules

Security group rule ID	Type	Protocol	Port range	Source		Description - optional
sgr-03f3e8ebe3259af5e	SSH	TCP	22	Custom	0.0.0.0/0	

Info
Delete
Add rule
Cancel
Preview changes
Save rules

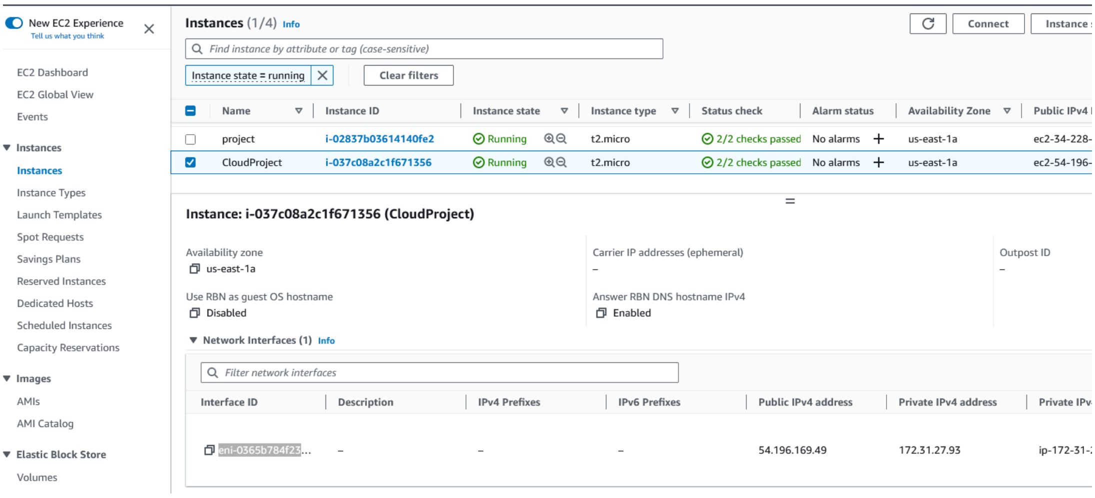
click on "Add rule".



Inbound rules (10)

Name	Security group rule...	IP version	Type	Protocol	Port range	Source	Description
–	sgr-04fdd8d98f82cf9c3	IPv4	HTTPS	TCP	443	0.0.0.0/0	–
–	sgr-0d530fc8247683f36	IPv6	HTTP	TCP	80	::/0	–
–	sgr-072777ac9f559cffc	IPv4	Custom TCP	TCP	0	0.0.0.0/0	–
–	sgr-0d385491525b83...	IPv6	Custom UDP	UDP	1885	::/0	–
–	sgr-03f3e8ebe3259af5e	IPv4	SSH	TCP	22	0.0.0.0/0	–
–	sgr-092587196bc0ccee0	IPv6	Custom TCP	TCP	5000	::/0	–
–	sgr-0dd578dd404b15...	IPv6	All ICMP - IPv6	IPv6 ICMP	All	::/0	–
–	sgr-0acceb760ef28dd76	IPv4	HTTP	TCP	80	0.0.0.0/0	–
–	sgr-0c1ae5103af9c6eb8	IPv6	HTTPS	TCP	443	::/0	–
–	sgr-0d77fe7f775dd26bb	IPv4	All ICMP - IPv4	ICMP	All	0.0.0.0/0	–

Fill in the list above. (Some ports could be different for you). After filling in all of them, click the yellow button "Save rules". Go back to your instances screen.



Instances (1/4)

	Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone	Public IPv4
□	project	i-02837b03614140fe2	Running	t2.micro	2/2 checks passed	No alarms	us-east-1a	ec2-34-228-
☑	CloudProject	i-037c08a2c1f671356	Running	t2.micro	2/2 checks passed	No alarms	us-east-1a	ec2-54-196-

Instance: i-037c08a2c1f671356 (CloudProject)

Availability zone: us-east-1a

Carrier IP addresses (ephemeral): –

Outpost ID: –

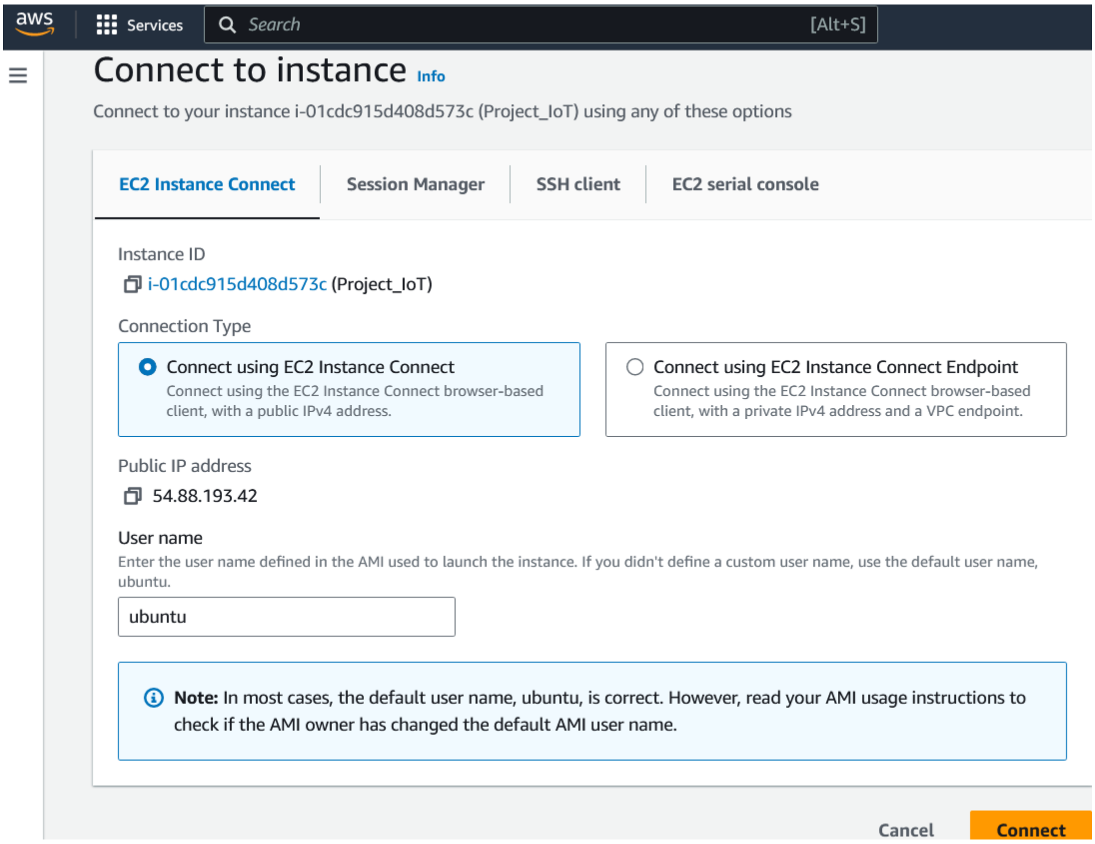
Use RBN as guest OS hostname: Disabled

Answer RBN DNS hostname IPv4: Enabled

Network Interfaces (1)

Interface ID	Description	IPv4 Prefixes	IPv6 Prefixes	Public IPv4 address	Private IPv4 address	Private IPv
eni-0365b784f23...	–	–	–	54.196.169.49	172.31.27.93	ip-172-31-

Click on the checkbox next to your instance and click the "Connect button".



Connect to instance Info

Connect to your instance i-01cdc915d408d573c (Project_IoT) using any of these options

Instance ID

i-01cdc915d408d573c (Project_IoT)

Connection Type

● Connect using EC2 Instance Connect
Connect using the EC2 Instance Connect browser-based client, with a public IPv4 address.

○ Connect using EC2 Instance Connect Endpoint
Connect using the EC2 Instance Connect browser-based client, with a private IPv4 address and a VPC endpoint.

Public IP address

54.88.193.42

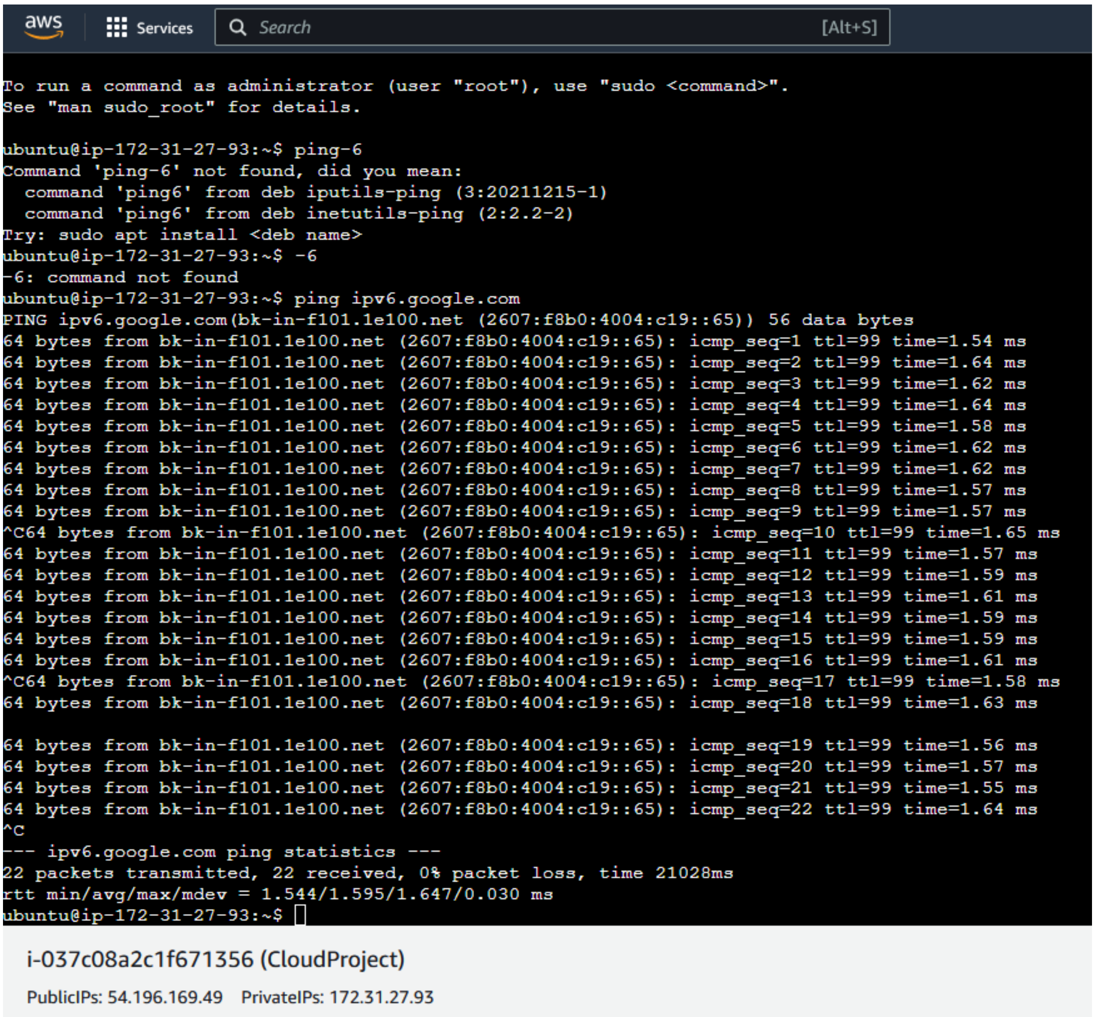
User name

Enter the user name defined in the AMI used to launch the instance. If you didn't define a custom user name, use the default user name, ubuntu.

ubuntu

Note: In most cases, the default user name, ubuntu, is correct. However, read your AMI usage instructions to check if the AMI owner has changed the default AMI user name.

On the first tier click on the yellow button "Connect".

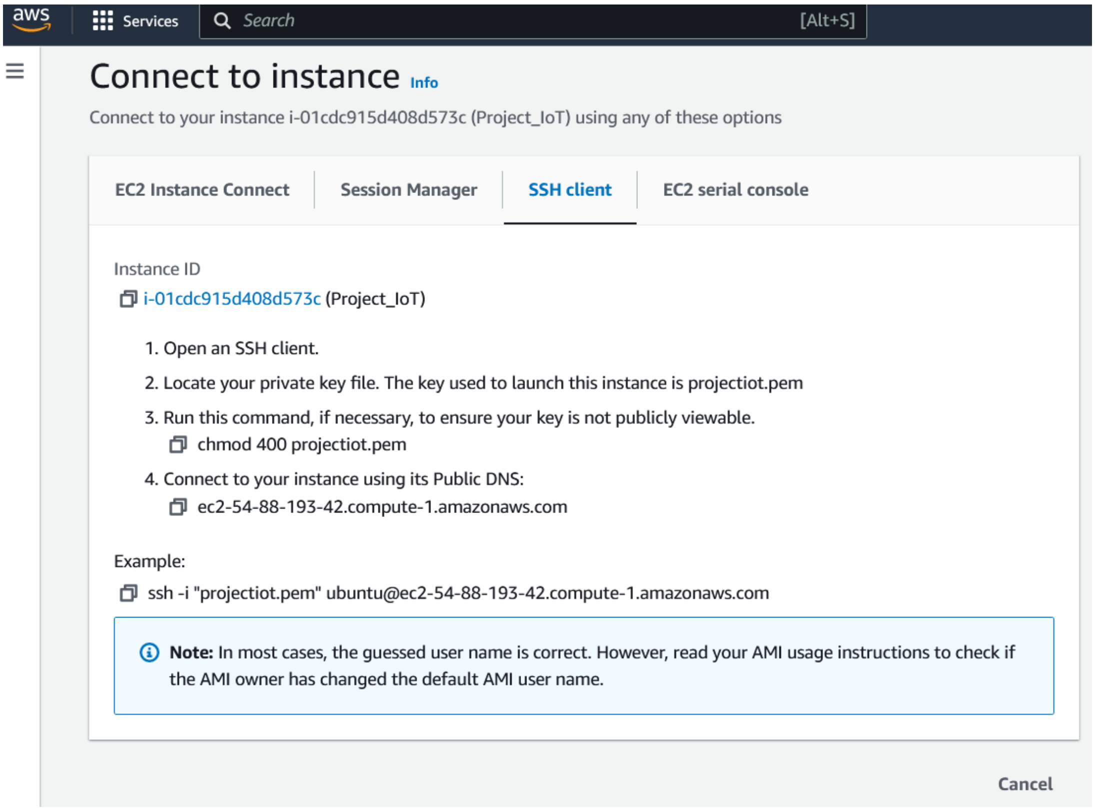


```
To run a command as administrator (user "root"), use "sudo <command>".
See "man sudo_root" for details.

ubuntu@ip-172-31-27-93:~$ ping-6
Command 'ping-6' not found, did you mean:
  command 'ping6' from deb iputils-ping (3:20211215-1)
  command 'ping6' from deb inetutils-ping (2:2.2-2)
Try: sudo apt install <deb name>
ubuntu@ip-172-31-27-93:~$ -6
-6: command not found
ubuntu@ip-172-31-27-93:~$ ping ipv6.google.com
PING ipv6.google.com(bk-in-f101.1e100.net (2607:f8b0:4004:c19::65)) 56 data bytes
64 bytes from bk-in-f101.1e100.net (2607:f8b0:4004:c19::65): icmp_seq=1 ttl=99 time=1.54 ms
64 bytes from bk-in-f101.1e100.net (2607:f8b0:4004:c19::65): icmp_seq=2 ttl=99 time=1.64 ms
64 bytes from bk-in-f101.1e100.net (2607:f8b0:4004:c19::65): icmp_seq=3 ttl=99 time=1.62 ms
64 bytes from bk-in-f101.1e100.net (2607:f8b0:4004:c19::65): icmp_seq=4 ttl=99 time=1.64 ms
64 bytes from bk-in-f101.1e100.net (2607:f8b0:4004:c19::65): icmp_seq=5 ttl=99 time=1.58 ms
64 bytes from bk-in-f101.1e100.net (2607:f8b0:4004:c19::65): icmp_seq=6 ttl=99 time=1.62 ms
64 bytes from bk-in-f101.1e100.net (2607:f8b0:4004:c19::65): icmp_seq=7 ttl=99 time=1.62 ms
64 bytes from bk-in-f101.1e100.net (2607:f8b0:4004:c19::65): icmp_seq=8 ttl=99 time=1.57 ms
64 bytes from bk-in-f101.1e100.net (2607:f8b0:4004:c19::65): icmp_seq=9 ttl=99 time=1.57 ms
^C64 bytes from bk-in-f101.1e100.net (2607:f8b0:4004:c19::65): icmp_seq=10 ttl=99 time=1.65 ms
64 bytes from bk-in-f101.1e100.net (2607:f8b0:4004:c19::65): icmp_seq=11 ttl=99 time=1.57 ms
64 bytes from bk-in-f101.1e100.net (2607:f8b0:4004:c19::65): icmp_seq=12 ttl=99 time=1.59 ms
64 bytes from bk-in-f101.1e100.net (2607:f8b0:4004:c19::65): icmp_seq=13 ttl=99 time=1.61 ms
64 bytes from bk-in-f101.1e100.net (2607:f8b0:4004:c19::65): icmp_seq=14 ttl=99 time=1.59 ms
64 bytes from bk-in-f101.1e100.net (2607:f8b0:4004:c19::65): icmp_seq=15 ttl=99 time=1.59 ms
64 bytes from bk-in-f101.1e100.net (2607:f8b0:4004:c19::65): icmp_seq=16 ttl=99 time=1.61 ms
^C64 bytes from bk-in-f101.1e100.net (2607:f8b0:4004:c19::65): icmp_seq=17 ttl=99 time=1.58 ms
64 bytes from bk-in-f101.1e100.net (2607:f8b0:4004:c19::65): icmp_seq=18 ttl=99 time=1.63 ms

64 bytes from bk-in-f101.1e100.net (2607:f8b0:4004:c19::65): icmp_seq=19 ttl=99 time=1.56 ms
64 bytes from bk-in-f101.1e100.net (2607:f8b0:4004:c19::65): icmp_seq=20 ttl=99 time=1.57 ms
64 bytes from bk-in-f101.1e100.net (2607:f8b0:4004:c19::65): icmp_seq=21 ttl=99 time=1.55 ms
64 bytes from bk-in-f101.1e100.net (2607:f8b0:4004:c19::65): icmp_seq=22 ttl=99 time=1.64 ms
^C
--- ipv6.google.com ping statistics ---
22 packets transmitted, 22 received, 0% packet loss, time 21028ms
rtt min/avg/max/mdev = 1.544/1.595/1.647/0.030 ms
ubuntu@ip-172-31-27-93:~$ 
```

i-037c08a2c1f671356 (CloudProject)

PublicIPs: 54.196.169.49 PrivateIPs: 172.31.27.93

Now your instance is connected. Go back to the Connect to instance screen.



Connect to instance Info

Connect to your instance i-01cdc915d408d573c (Project_IoT) using any of these options

Instance ID

i-01cdc915d408d573c (Project_IoT)

1. Open an SSH client.
2. Locate your private key file. The key used to launch this instance is projectiot.pem
3. Run this command, if necessary, to ensure your key is not publicly viewable.
   chmod 400 projectiot.pem
4. Connect to your instance using its Public DNS:
   ec2-54-88-193-42.compute-1.amazonaws.com

Example:

```
ssh -i "projectiot.pem" ubuntu@ec2-54-88-193-42.compute-1.amazonaws.com
```

Note: In most cases, the guessed user name is correct. However, read your AMI usage instructions to check if the AMI owner has changed the default AMI user name.

To be able to use multiple terminals of your instance, select the tier "SSH client". Copy the Example and use it in your Linux Ubuntu environment. To connect to your instance from a different terminal.

Your EC2 is established.

Installations:

1. Open the EC2 instance terminal or connect via ssh.
2. Create a directory e.g. Project and navigate into it.

```
mkdir Project
cd Project
```

3. Install the Mosquito.RSMB broker. Follow the Setting up a broker (Steps 1-3) tutorial.
4. Install SQLITE.

```
sudo apt install sqlite3
```

5. In the Project directory, create a database.

```
sqlite3 mybase.db
```

6. Create a temperature table.

```
CREATE TABLE temperature (
id INTEGER PRIMARY KEY AUTOINCREMENT,
date TEXT,
time TEXT,
value INTEGER,
scale INTEGER
);
```

Type `.exit` to close sqlite3.

7. For the backend script, install the following libraries:
   - PAHO MQTT

     ```
     pip install paho-mqtt
     ```

   - FLASK

     ```
     pip install Flask
     ```

   - FLASK CORS

     ```
     pip install flask-cors
     ```

8. Copy the *server.py* file from the *Cloud* directory into the *Project* directory.
9. Change the address of the broker.

   ```
   broker_address = "<IPv6 of the EC2 instance>"
   ```

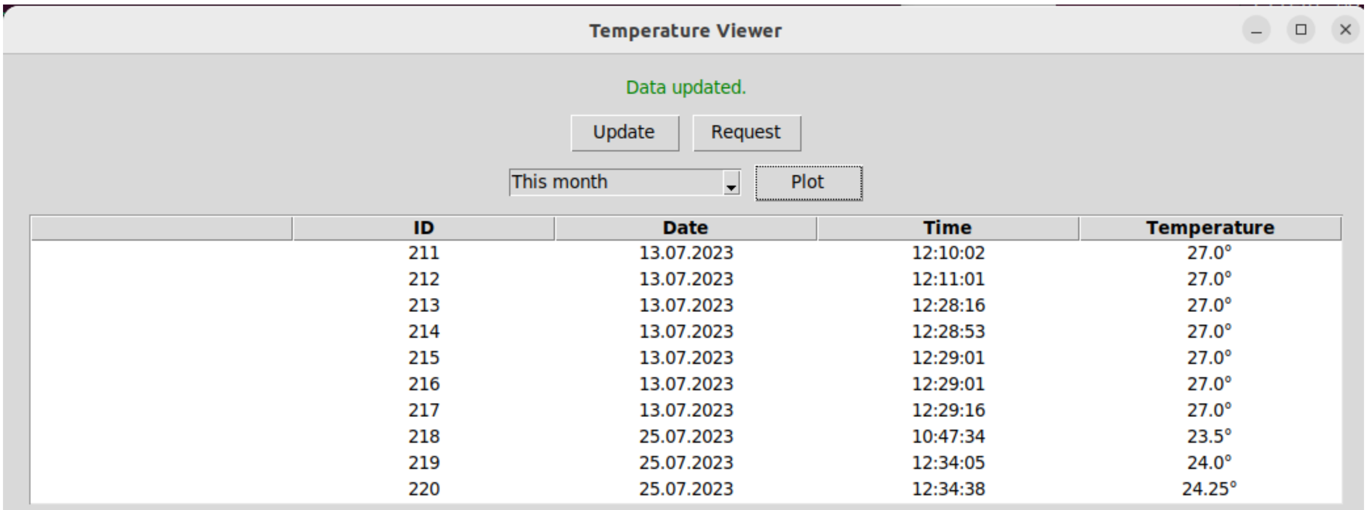
10. Enable port 1885 for incoming UDP IPv6 packets and set up a security rule.
11. Enable port 5000 for incoming TCP IPv6 packets and set up a security rule.

User Interface

Overview:

The user interface is a Desktop program which is used to visualize the stored data. The data is read from a database located at the cloud and is presented into a table. Also a temperature diagram can be created for the current day or the current month. The user can update the data or send a request to the sensor node via the cloud. Note, that the presented data is only updated by clicking on the "Update" button. For the communication between the application and the cloud, HTTP protocol is used. Since for the communication IPv6 is used, the machine of the user has to have a IPv6 network connection. The Program itself is written in

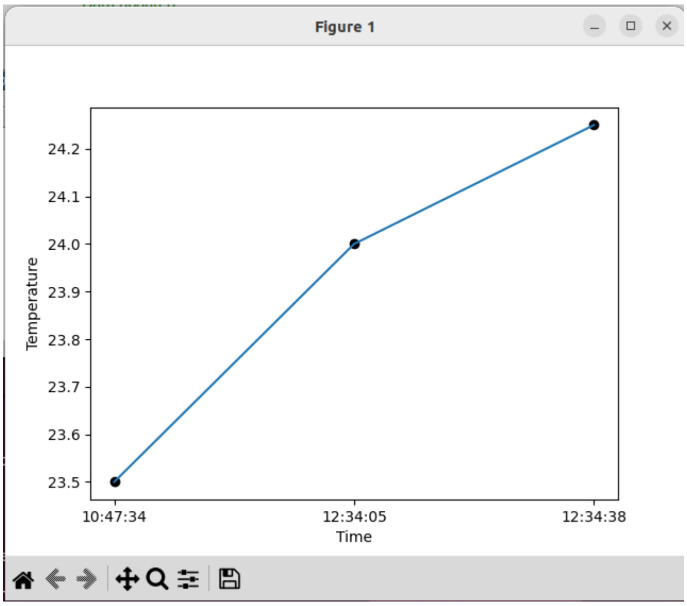
Python using the 'Tkinter' library for the graphical user interface, 'Matplotlib' for the diagrams and 'Requests' for the HTTP requests.

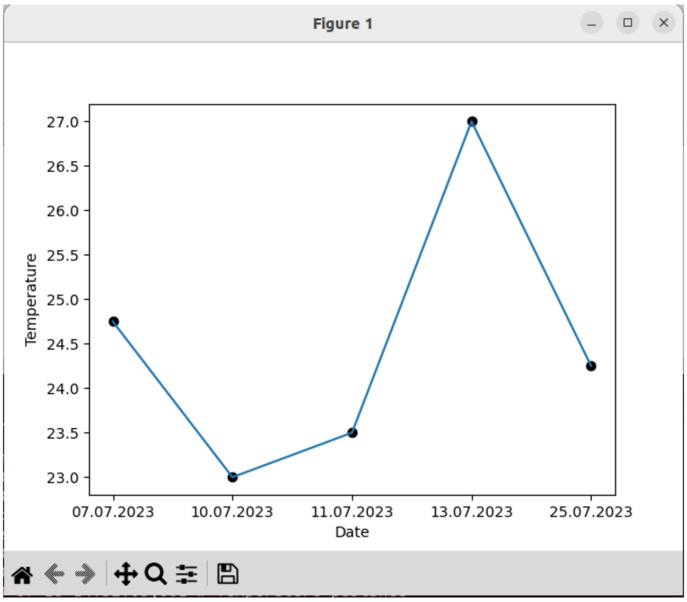


Data updated.

This month

	ID	Date	Time	Temperature
	211	13.07.2023	12:10:02	27.0°
	212	13.07.2023	12:11:01	27.0°
	213	13.07.2023	12:28:16	27.0°
	214	13.07.2023	12:28:53	27.0°
	215	13.07.2023	12:29:01	27.0°
	216	13.07.2023	12:29:01	27.0°
	217	13.07.2023	12:29:16	27.0°
	218	25.07.2023	10:47:34	23.5°
	219	25.07.2023	12:34:05	24.0°
	220	25.07.2023	12:34:38	24.25°





Set up:

1. Make sure that python is installed otherwise install it.
2. Install the following python libraries:
    - Tkinter

      ```
      sudo apt-get install python3-tk
      ```

    - Matplotlib

      ```
      pip install matplotlib
      ```

    - Resquests

```
pip install requests
```

3. Open the *UI* folder.
4. Change the address of the EC2 instance in the Client.py file.

```
self.base_url = "http://[<IPv6_Address>]:5000"
```

Start The System

1. Open two terminals and connect with the EC2 instance via ssh.
2. In both terminals navigate to the Project directory.

```
cd Project
```

3. In the first terminal start the Mosquitto RSMB broker.

```
cd mosquitto.rsmb/rsmb/src
./broker_mqtts config.conf
```

4. In the second terminal start the backend script.

```
python3 server.py
```

5. Start the border router.

```
PORT=/dev/ttyACM0 BOARD="nrf52840dongle" make term
```

6. Start the sensor node.

```
PORT=/dev/ttyACM2 BOARD="nrf52840dk" make all flash term
```

If it is already running or does not print the status restart it or reflash it.

```
PORT=/dev/ttyACM2 BOARD="nrf52840dk" make reset
```

7. Start the user interface.

   ```
   python3 Application.py
   ```

- Note: If some of the components throw errors, it might help to restart them.

Sources

- RIOT documentation: https://doc.riot-os.org/
- RIOT Github Repository: https://github.com/RIOT-OS/RIOT
- Tkinter: https://www.pythontutorial.net/tkinter/tkinter-mvc/
- Flask: https://flask.palletsprojects.com/en/2.3.x/
- Paho: http://www.steves-internet-guide.com/into-mqtt-python-client/
- Assign IPv6 address to EC2 instance: https://4sysops.com/archives/assign-an-ipv6-address-to-an-ec2-instance-dual-stack/