Software Installation Guide:

BioTuring Colab

Introduction

- Software to tackle biomedical challenges

Colaboratory, or “Colab” for short, is a product from Bioturing. Colab has a variety of features and pre-built notebooks their user can download and use. We are providing many tools that help users to post their data and analyze the reports. Our product can be used to write and execute arbitrary python, R code, Golang, Julia, RStudio, VS Code and many more through the browser, and is especially well suited to data analysis and education. More technically, Colab is a hosted Jupyter notebook service that requires no setup to use. Users can build their own notebook.

System Requirements

Before installing the Colab, some pre-installation steps are required:

	Basic Requirements for BioColab installation

	Basic recommendation	Optional
CPU	16 core	This is basic requirement to start BioColab and based on requirement Resources as well as machine can be added
RAM	64 Gb	As above
HDD	/ partition can be 100 GB	as above
	Data Volume : 1TB	As above
OS	Any OS. Ubuntu 20.04 and above.	BioColab is more supportive with Linux OS. For better performance linux OS is recommended.
AWS Instance	Support any type of instance type. Depend on need	AWS g5xlarge in case using GPU
Platform	Docker / Kubernetes	
Note		
	In case we want to use Notebook with BioColab. We need to install NVIDIA and below would be a requirement.	
	The system has one or multiple NVIDIA GPU(s) (at least 16 GB memory per GPU) - with Turing architecture or above.	
	BioColan supports any Linux OS. We are recommending Ubuntu 20.04 or above.	
	SSL can be configured later also.	
	Please contact support@bioturing.com to get the token for your company.	
Security		
	The BioColab platform uses HTTPS protocol to securely communicate over the network.	
	All of the users need to authenticate using a BioTuring account or the company's SSO to access the platform.	
	We highly recommend setting up a private VPC network for IP restriction.	
	The data stays behind the company firewall.	
	The BioColab platform does not track any usage logs.	
Data visibility		

	Data can be uploaded to Personal Workspace or Data Sharing group.
	In the Personal Workspace, only the owner can see and manipulate the data she/he uploaded.
	In the Data Sharing group, only people in the group can see the data.
	In the Data Sharing group, only people with sufficient permissions can manipulate the data.

- The system has one or multiple NVIDIA GPU(s) (at least 16 GB memory per GPU) - with Colab Bioturing - architecture or above.
- The system is running Ubuntu 20.04 or above.
- SSL certificate and a domain name for users to securely access the platform on the web browser. I can be installed later too.
- Please contact support@bioturing.com to get the token for your company.

Note: The ideal system that we recommend for most companies is AWS g5.8xlarge for GPU based. Instance can be chosen based on requirement. If the notebook is not based on GPU, we can select lower.
Our Product is containerized applications, Here we are illustrated GPU based instances, Kindly select based on your requirement. It can be run on Docker using Docker engine and Kubernetes.

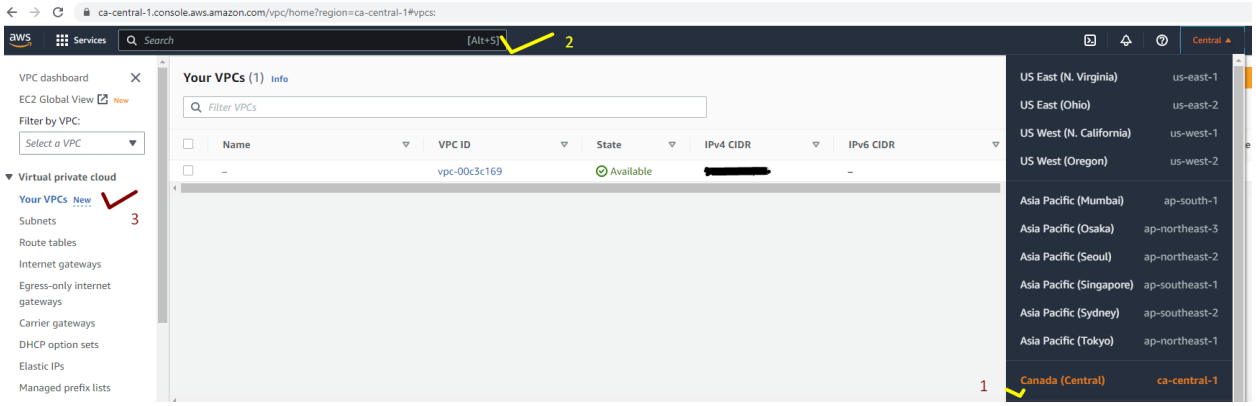
Download and Install

> **Note:** We suggest starting from scratch to avoid package/driver conflicts.
>
> For Tag naming conversion, kindly select based on your architecture.

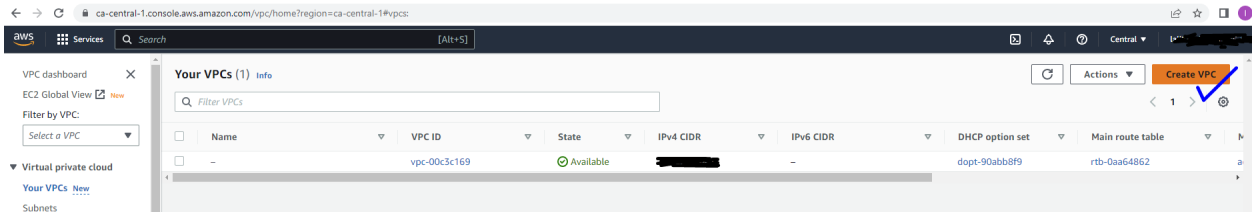
Login to AWS console with admin user account to launch an Ec2 instance.
Note: It's up to the client. How they are going to manage infrastructure, Load, Network, Access and traffic …etc.

- Create a VPC
  - *https://docs.aws.amazon.com/vpc/latest/userguide/what-is-amazon-vpc.html*
  - *https://docs.aws.amazon.com/vpc/latest/userguide/create-vpc.html*

1. Select the appropriate region for VPC.
2. Search VPC on the search box.
3. Click on Your VPCs New



4. Click on Create VPC



5. Follow the steps given in the image below.

VPC > Your VPCs > Create 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

Resources to create Info

Create only the VPC resource or the VPC and other networking resources.

(•) VPC only ( ) VPC and more

Name tag - *optional*

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

lalit_install_vpc

IPv4 CIDR block Info

(•) IPv4 CIDR manual input

( ) IPAM-allocated IPv4 CIDR block

IPv4 CIDR

192.168.0.0/20

IPv6 CIDR block Info

(•) No IPv6 CIDR block

( ) IPAM-allocated IPv6 CIDR block

( ) Amazon-provided IPv6 CIDR block

( ) IPv6 CIDR owned by me

Tenancy Info

Default

Tags

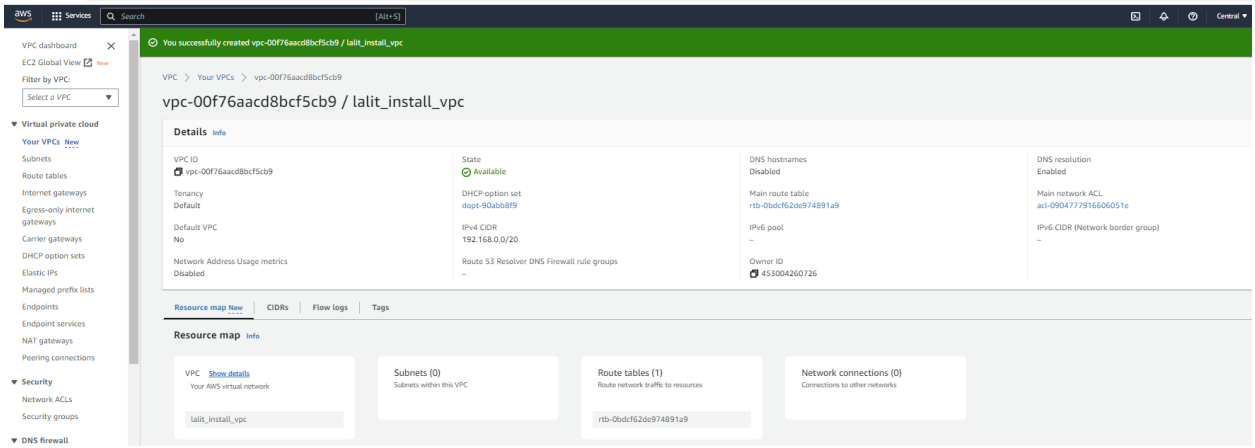
A tag is a label that you assign to an AWS resource. Each tag consists of a key and an optional value. You can use tags to search and filter your resources or track your AWS costs.

Key	Value - *optional*	
lalit_install_testing	lalit-test-Talk2data	Remove
Name	lalit_install_vpc	Remove

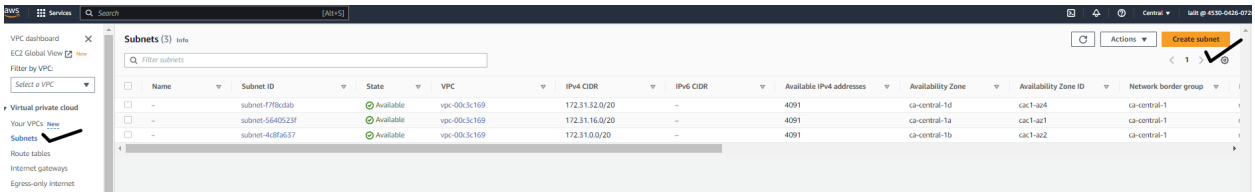
Add new tag

You can add 48 more tags.

Cancel Create VPC



- Create Subnet



VPC > Subnets > Create subnet

Create subnet Info

VPC

VPC ID
Create subnets in this VPC.

vpc-00f76aacd8bcf5cb9 (lalit_install_vpc)

Associated VPC CIDRs

IPv4 CIDRs
192.168.0.0/20

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.

lalit_install_subnet

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.

Canada (Central) / ca-central-1a

IPv4 CIDR block Info

192.168.0.0/24

▼ Tags - *optional*

Key	Value - *optional*	
Name	lalit_install_subnet	Remove
lalit_install_testing	lalit-test-Talk2data	Remove

Add new tag

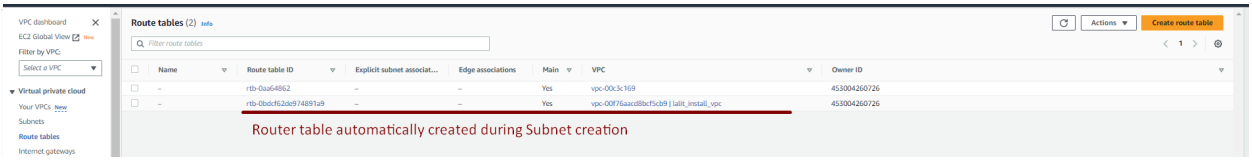
You can add 48 more tags.

Remove

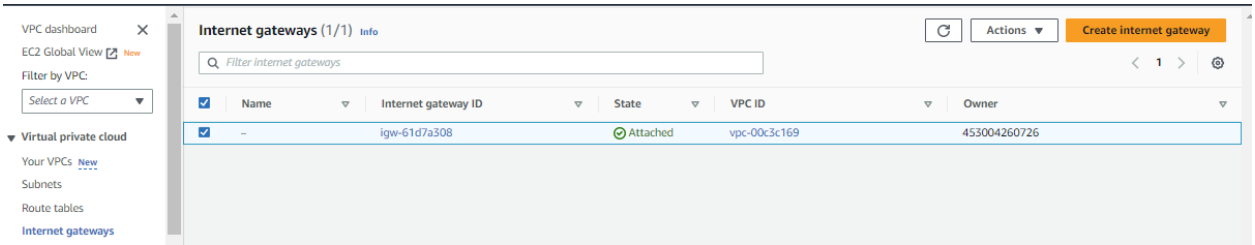
Add new subnet

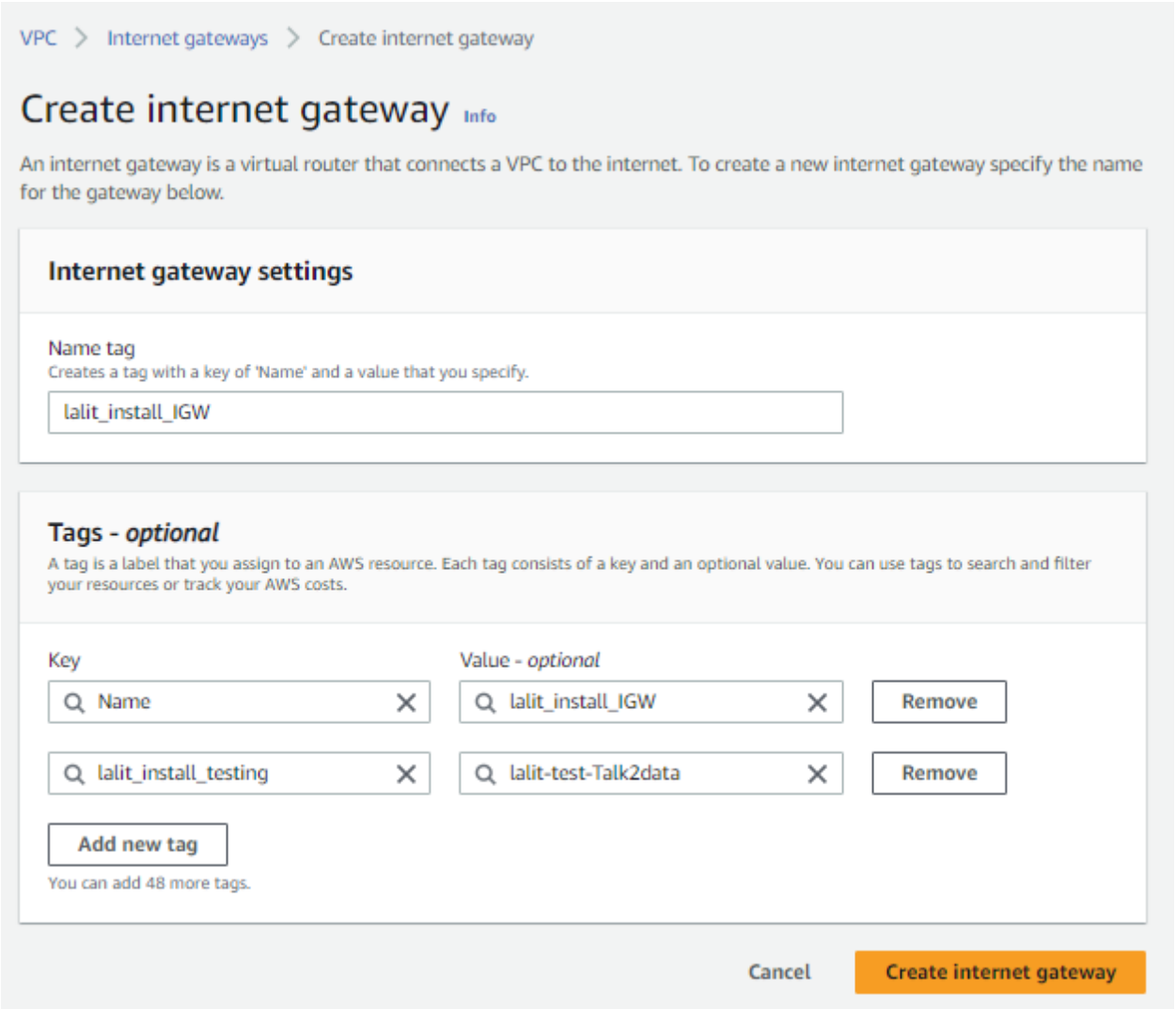
Cancel Create subnet

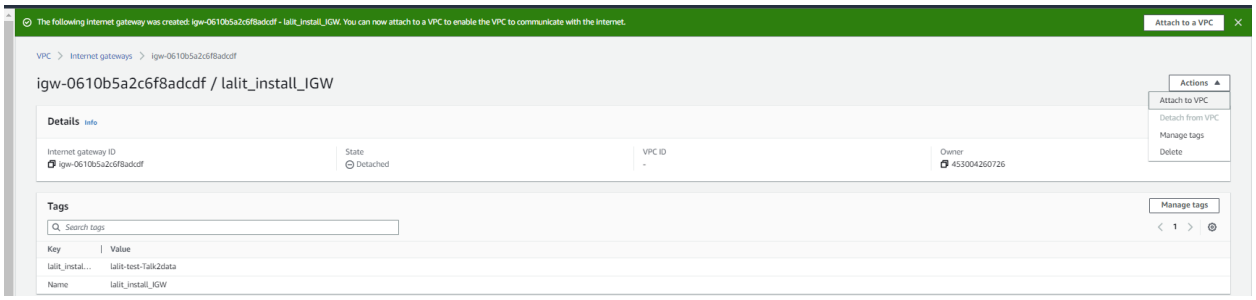
- Verify Router Table



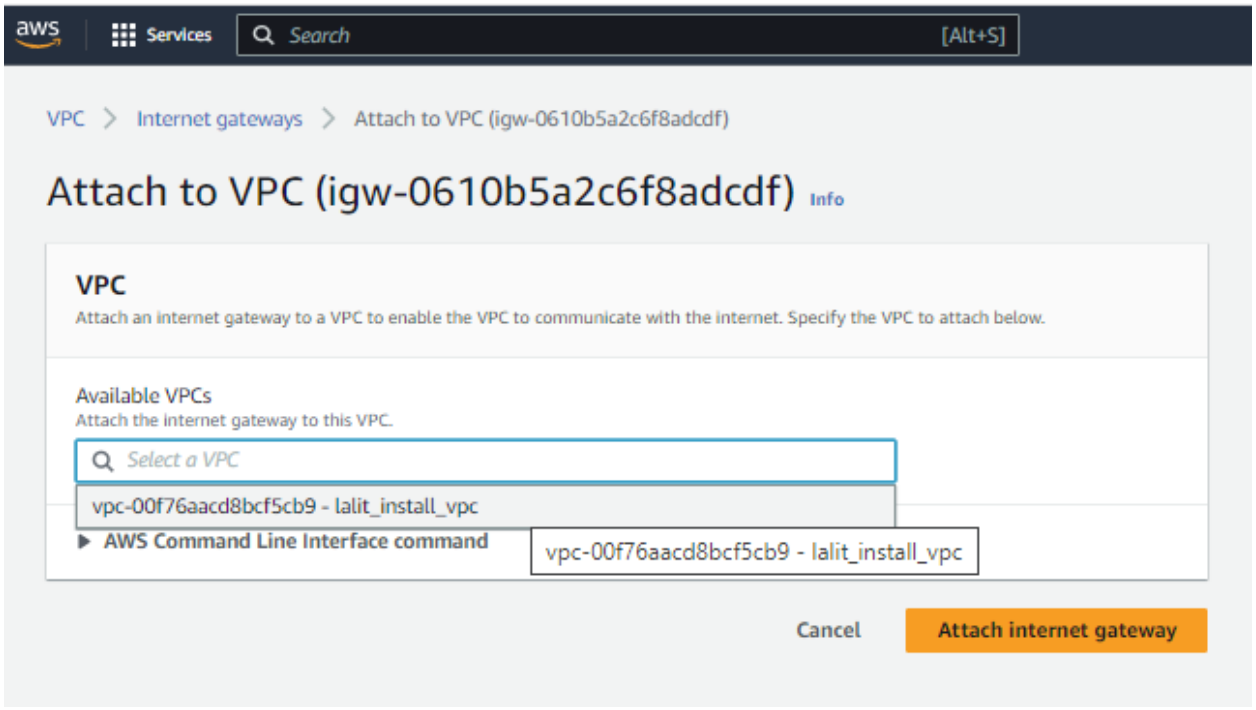
- Create Internet Gateway



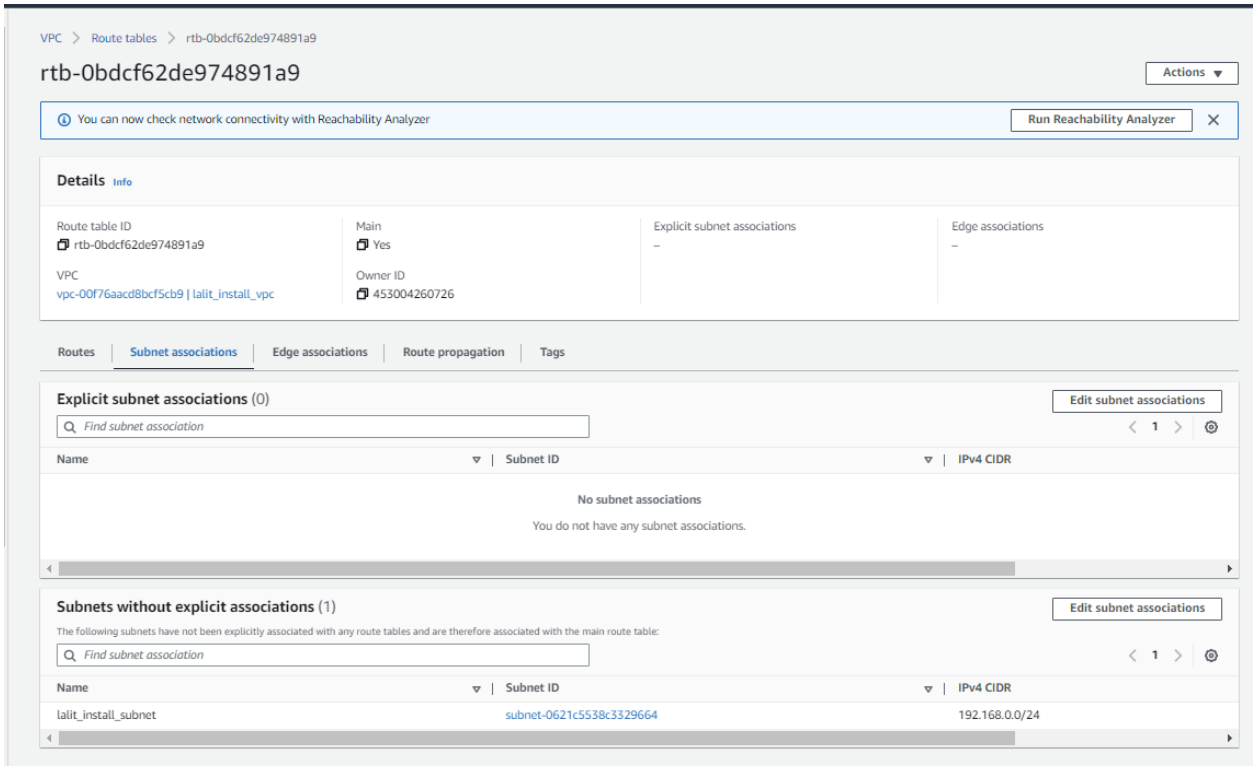


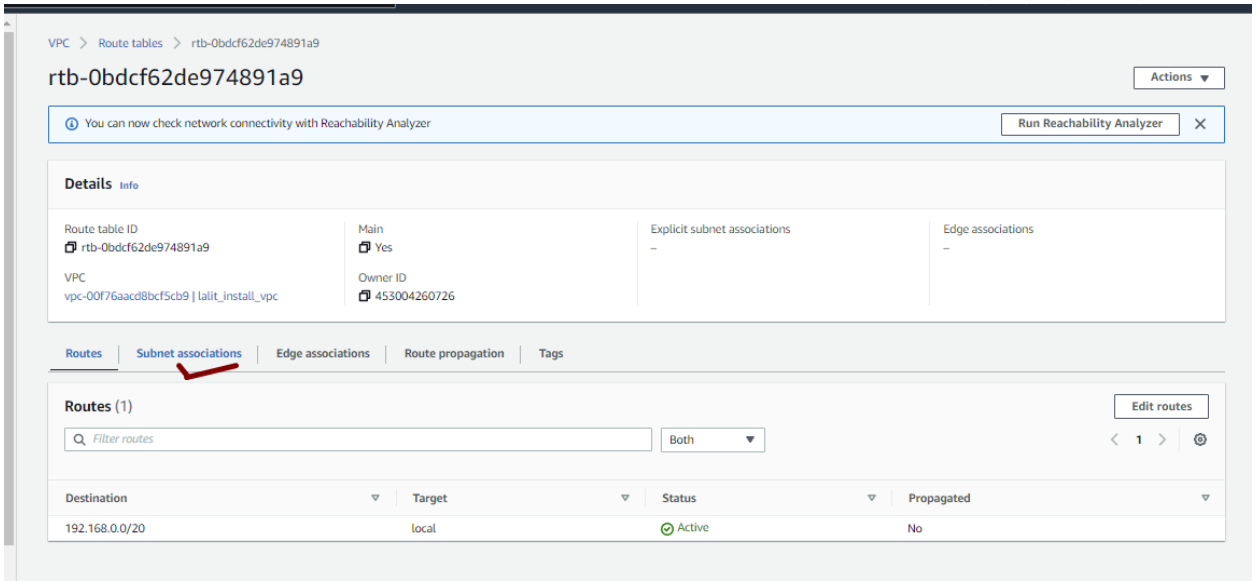


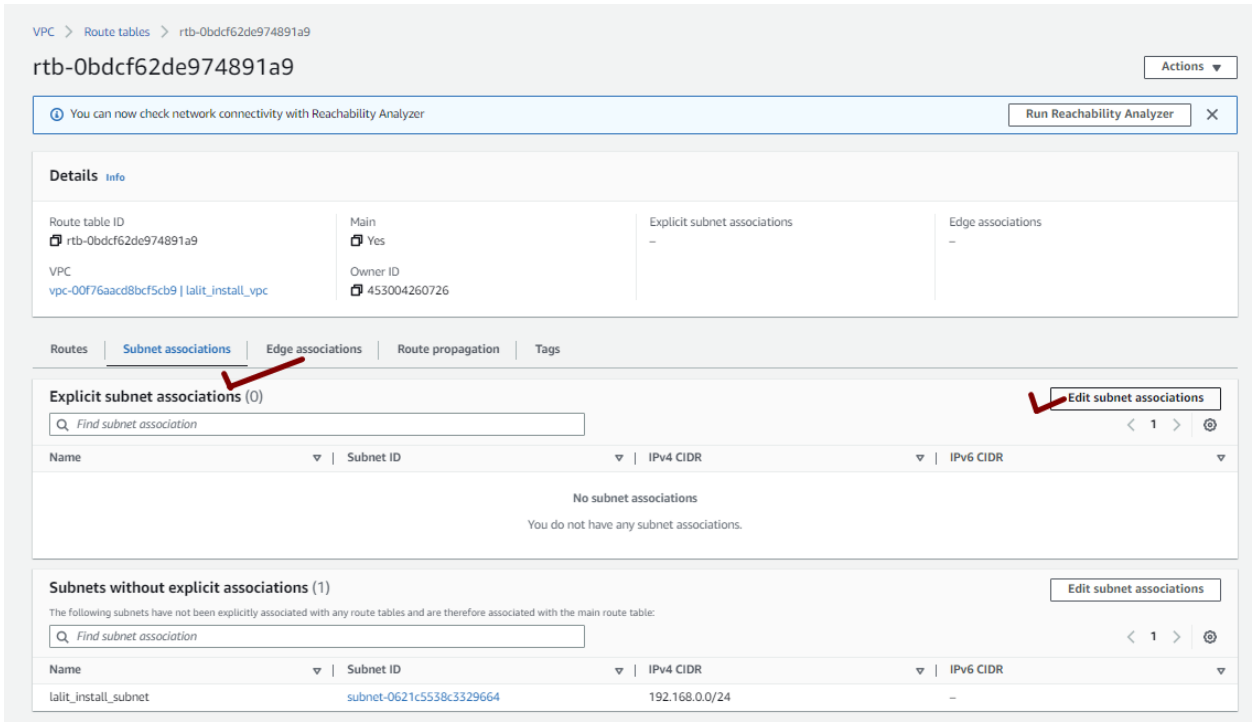
- IGW – Attach to VPC

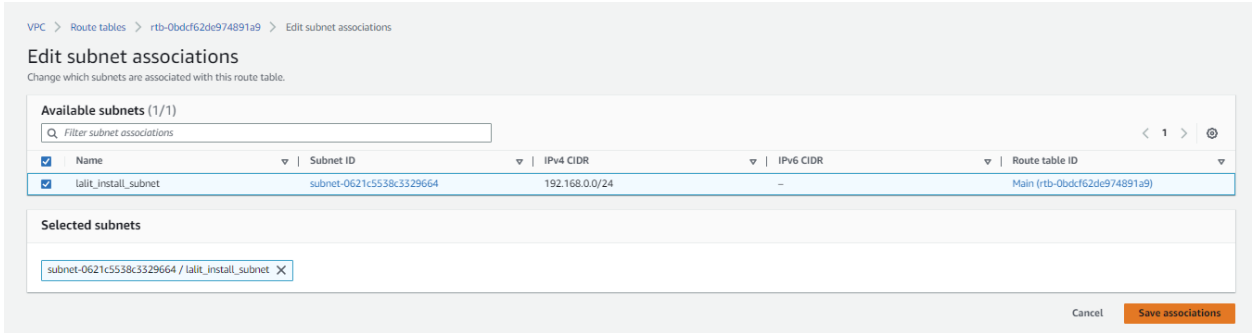


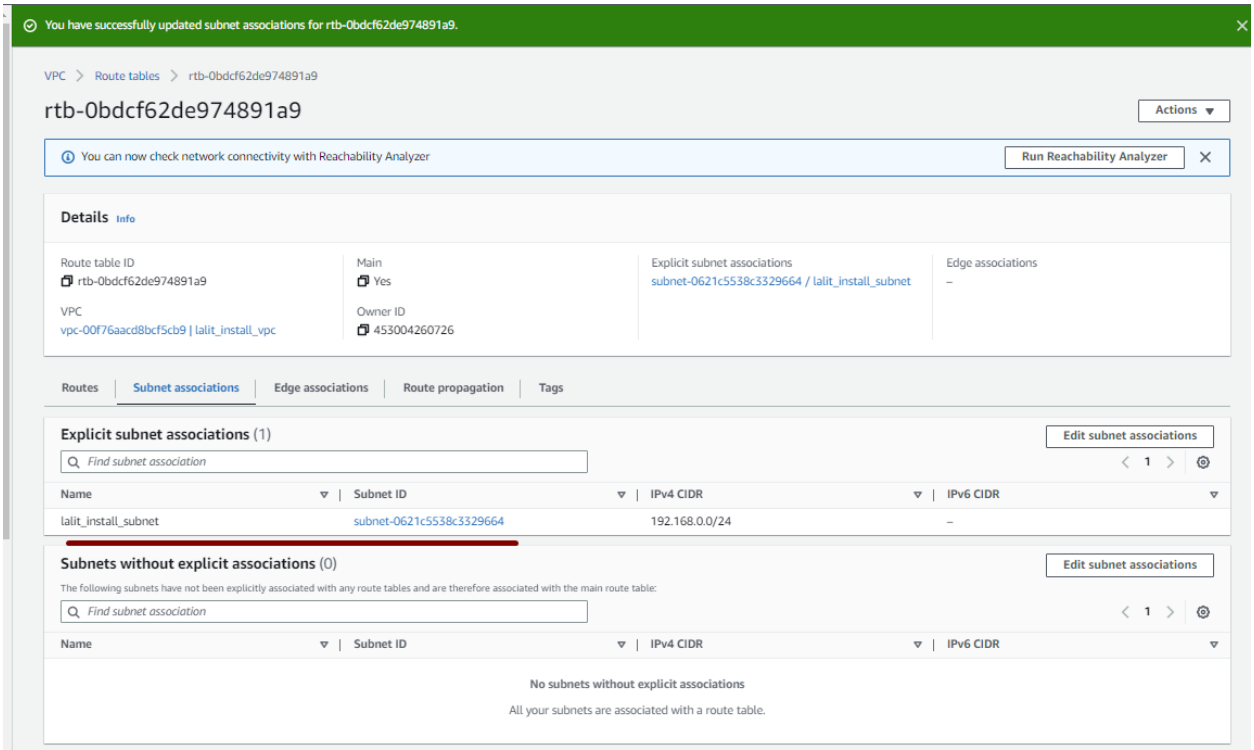
- Update Router

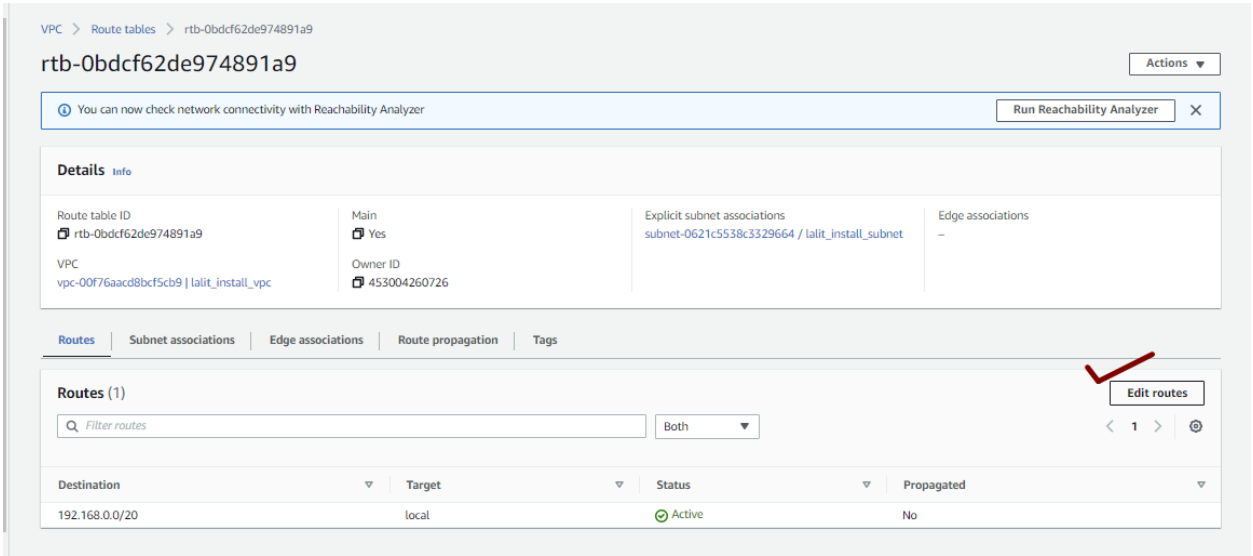

VPC > Route tables > rtb-0bdcf62de974891a9
rtb-0bdcf62de974891a9
Actions
You can now check network connectivity with Reachability Analyzer
Run Reachability Analyzer
Details Info
Route table ID
rtb-0bdcf62de974891a9
Main
Yes
Explicit subnet associations
–
Edge associations
–
VPC
vpc-00f76aacd8bcf5cb9 | lalit_install_vpc
Owner ID
453004260726
Routes
Subnet associations
Edge associations
Route propagation
Tags
Explicit subnet associations (0)
Edit subnet associations
Find subnet association
Name
Subnet ID
IPv4 CIDR
No subnet associations
You do not have any subnet associations.
Subnets without explicit associations (1)
Edit subnet associations
The following subnets have not been explicitly associated with any route tables and are therefore associated with the main route table:
Find subnet association
Name
Subnet ID
IPv4 CIDR
lalit_install_subnet
subnet-0621c5538c3329664
192.168.0.0/24

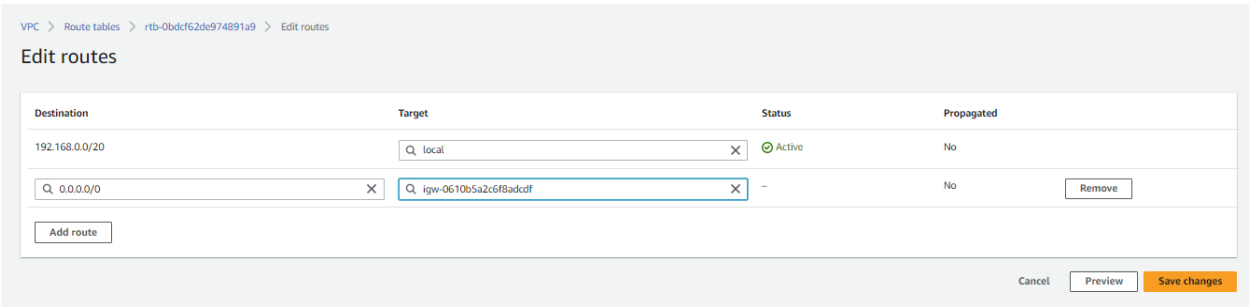

VPC > Route tables > rtb-0bdcf62de974891a9
rtb-0bdcf62de974891a9
Actions
You can now check network connectivity with Reachability Analyzer
Run Reachability Analyzer
Details Info
Route table ID
rtb-0bdcf62de974891a9
Main
Yes
Explicit subnet associations
–
Edge associations
–
VPC
vpc-00f76aacd8bcf5cb9 | lalit_install_vpc
Owner ID
453004260726
Routes
Subnet associations
Edge associations
Route propagation
Tags
Routes (1)
Edit routes
Filter routes
Both
Destination
Target
Status
Propagated
192.168.0.0/20
local
Active
No


VPC > Route tables > rtb-0bdcf62de974891a9
rtb-0bdcf62de974891a9
Actions
You can now check network connectivity with Reachability Analyzer
Run Reachability Analyzer
Details Info
Route table ID
rtb-0bdcf62de974891a9
Main
Yes
Explicit subnet associations
–
Edge associations
–
VPC
vpc-00f76aacd8bcf5cb9 | lalit_install_vpc
Owner ID
453004260726
Routes
Subnet associations
Edge associations
Route propagation
Tags
Explicit subnet associations (0)
Edit subnet associations
Find subnet association
Name
Subnet ID
IPv4 CIDR
IPv6 CIDR
No subnet associations
You do not have any subnet associations.
Subnets without explicit associations (1)
Edit subnet associations
The following subnets have not been explicitly associated with any route tables and are therefore associated with the main route table:
Find subnet association
Name
Subnet ID
IPv4 CIDR
IPv6 CIDR
lalit_install_subnet
subnet-0621c5538c3329664
192.168.0.0/24
–

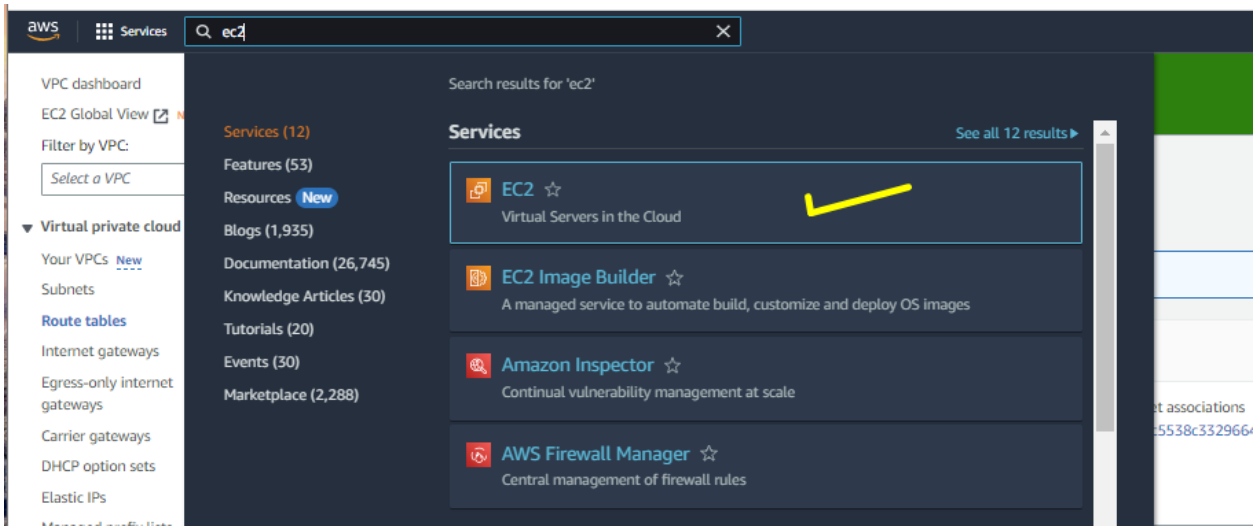

VPC > Route tables > rtb-0bdcf62de974891a9 > Edit subnet associations
Edit subnet associations
Change which subnets are associated with this route table.
Available subnets (1/1)
Filter subnet associations
Name
Subnet ID
IPv4 CIDR
IPv6 CIDR
Route table ID
lalit_install_subnet
subnet-0621c5538c3329664
192.168.0.0/24
–
Main (rtb-0bdcf62de974891a9)
Selected subnets
subnet-0621c5538c3329664 / lalit_install_subnet
Cancel
Save associations

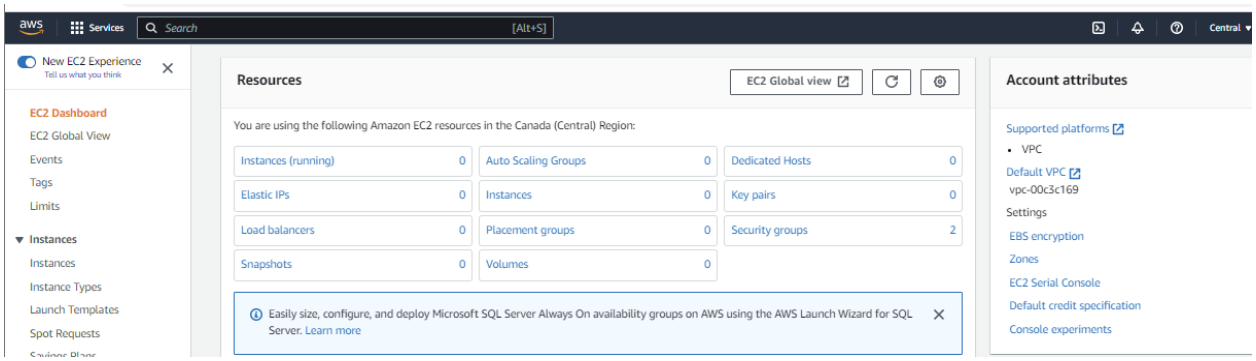


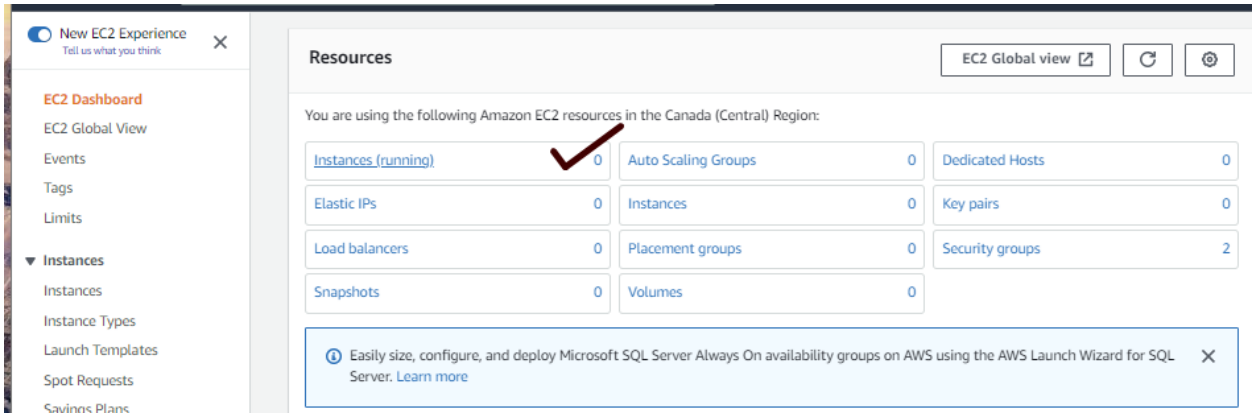


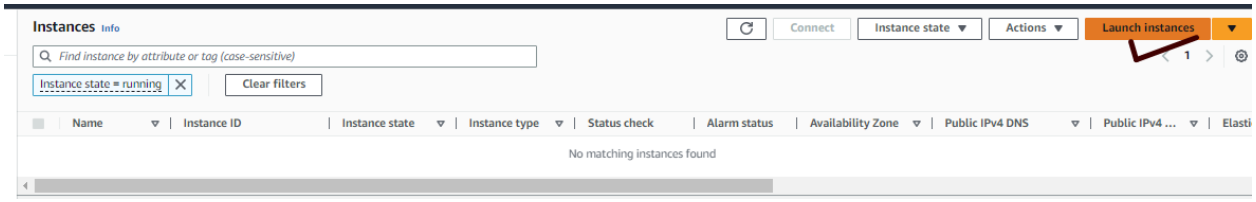


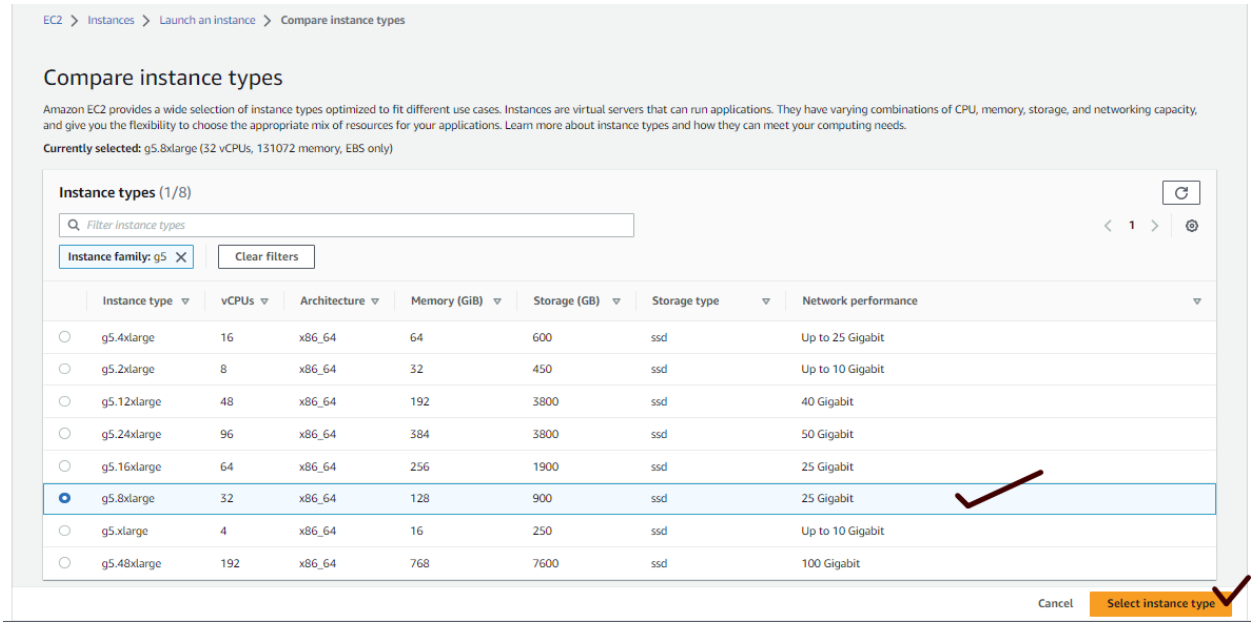
- Create instance based on your requirement


aws
Services
ec2
VPC dashboard
EC2 Global View
Filter by VPC:
Select a VPC
Virtual private cloud
Your VPCs New
Subnets
Route tables
Internet gateways
Egress-only internet gateways
Carrier gateways
DHCP option sets
Elastic IPs
Search results for 'ec2'
Services (12)
Features (53)
Resources New
Blogs (1,935)
Documentation (26,745)
Knowledge Articles (30)
Tutorials (20)
Events (30)
Marketplace (2,288)
Services
See all 12 results
EC2
Virtual Servers in the Cloud
EC2 Image Builder
A managed service to automate build, customize and deploy OS images
Amazon Inspector
Continual vulnerability management at scale
AWS Firewall Manager
Central management of firewall rules

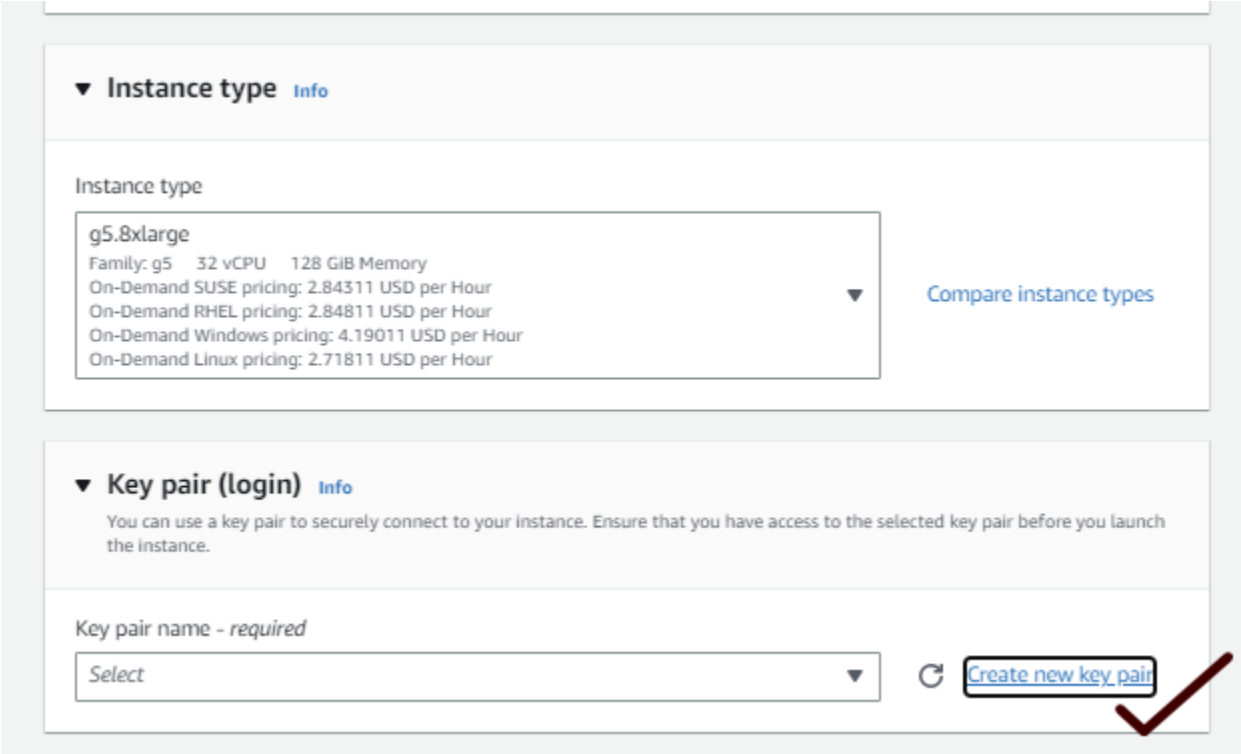

aws
Services
Search
[Alt+S]
Central
New EC2 Experience
Tell us what you think
EC2 Dashboard
EC2 Global View
Events
Tags
Limits
Instances
Instances
Instance Types
Launch Templates
Spot Requests
Resources
EC2 Global view
You are using the following Amazon EC2 resources in the Canada (Central) Region:
Instances (running) 0
Auto Scaling Groups 0
Dedicated Hosts 0
Elastic IPs 0
Instances 0
Key pairs 0
Load balancers 0
Placement groups 0
Security groups 2
Snapshots 0
Volumes 0
Easily size, configure, and deploy Microsoft SQL Server Always On availability groups on AWS using the AWS Launch Wizard for SQL Server. Learn more
Account attributes
Supported platforms
VPC
Default VPC
vpc-00c3c169
Settings
EBS encryption
Zones
EC2 Serial Console
Default credit specification
Console experiments

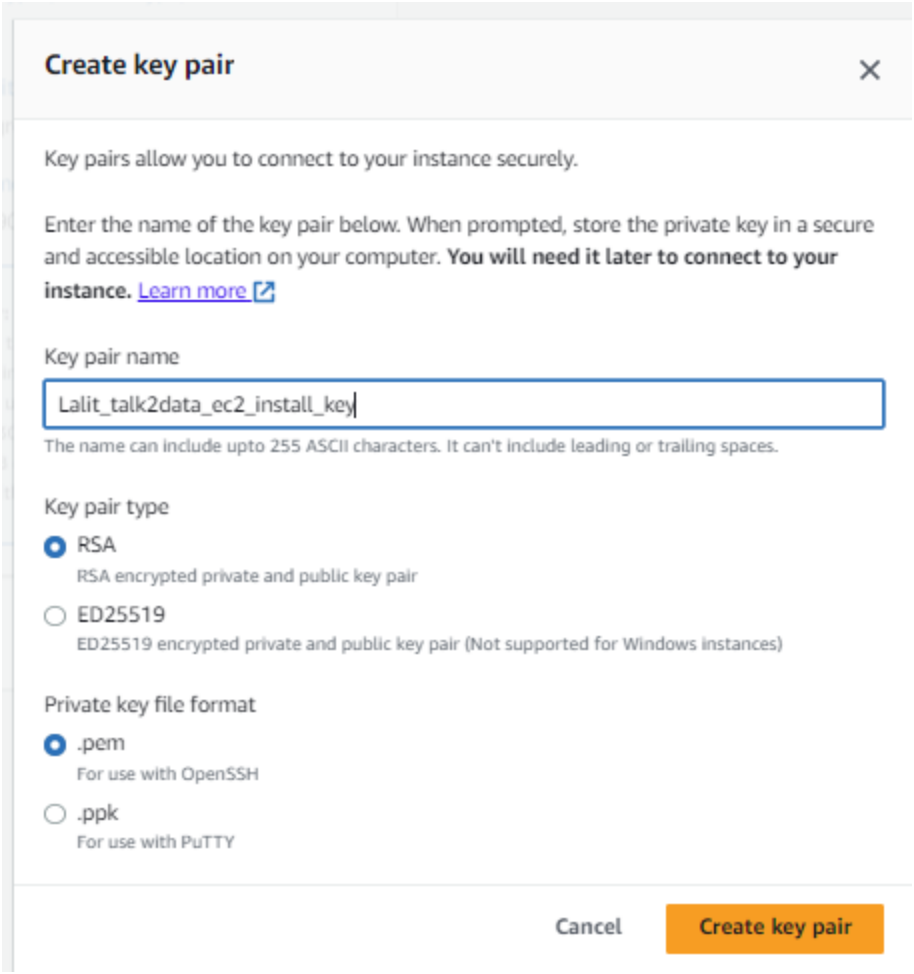

New EC2 Experience
Tell us what you think
EC2 Dashboard
EC2 Global View
Events
Tags
Limits
Instances
Instances
Instance Types
Launch Templates
Spot Requests
Savings Plans
Resources
EC2 Global view
You are using the following Amazon EC2 resources in the Canada (Central) Region:
Instances (running) 0
Auto Scaling Groups 0
Dedicated Hosts 0
Elastic IPs 0
Instances 0
Key pairs 0
Load balancers 0
Placement groups 0
Security groups 2
Snapshots 0
Volumes 0
Easily size, configure, and deploy Microsoft SQL Server Always On availability groups on AWS using the AWS Launch Wizard for SQL Server. Learn more


Instances 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
Public IPv4 ...
Elasti
No matching instances found

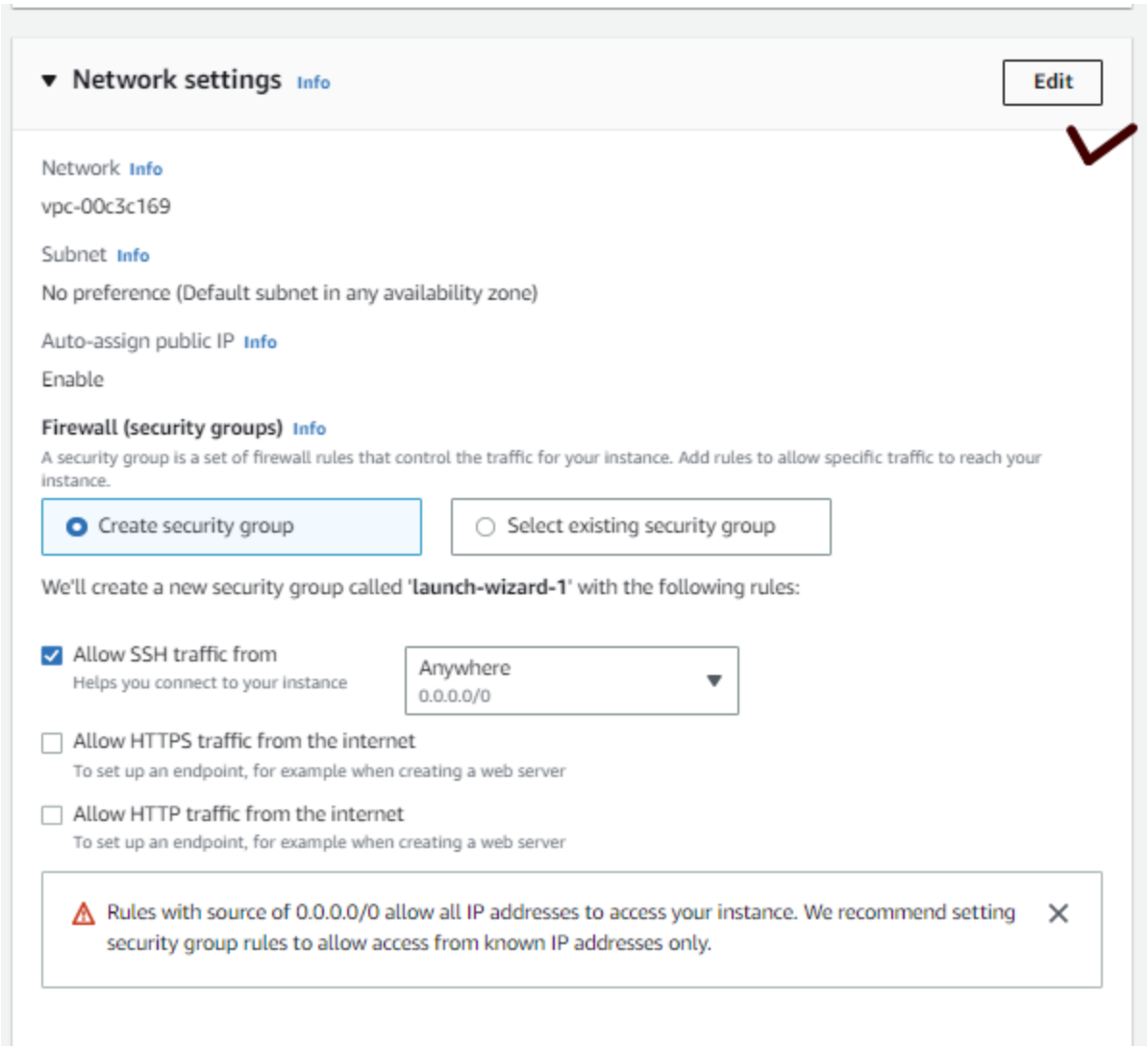


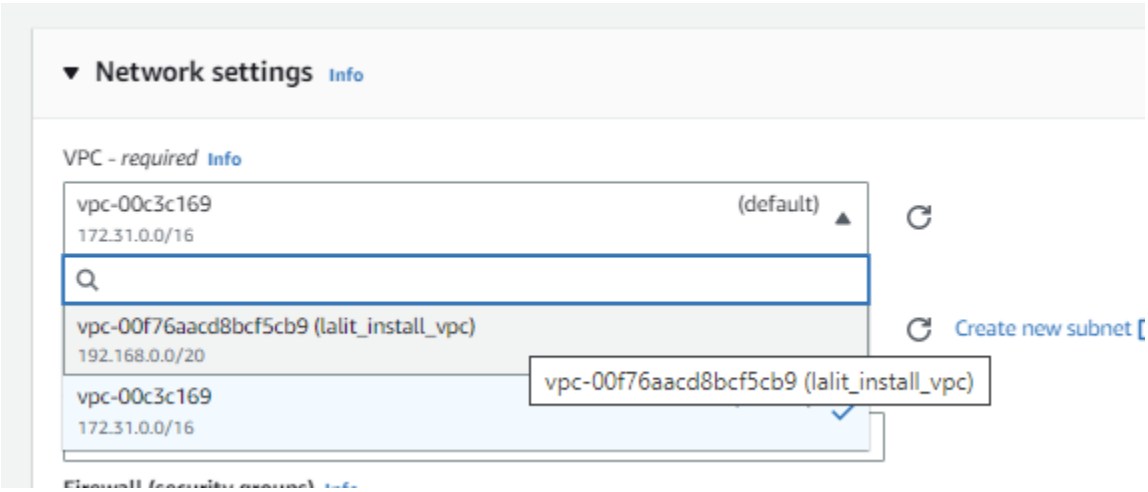
- Create a new key for SSH login to server



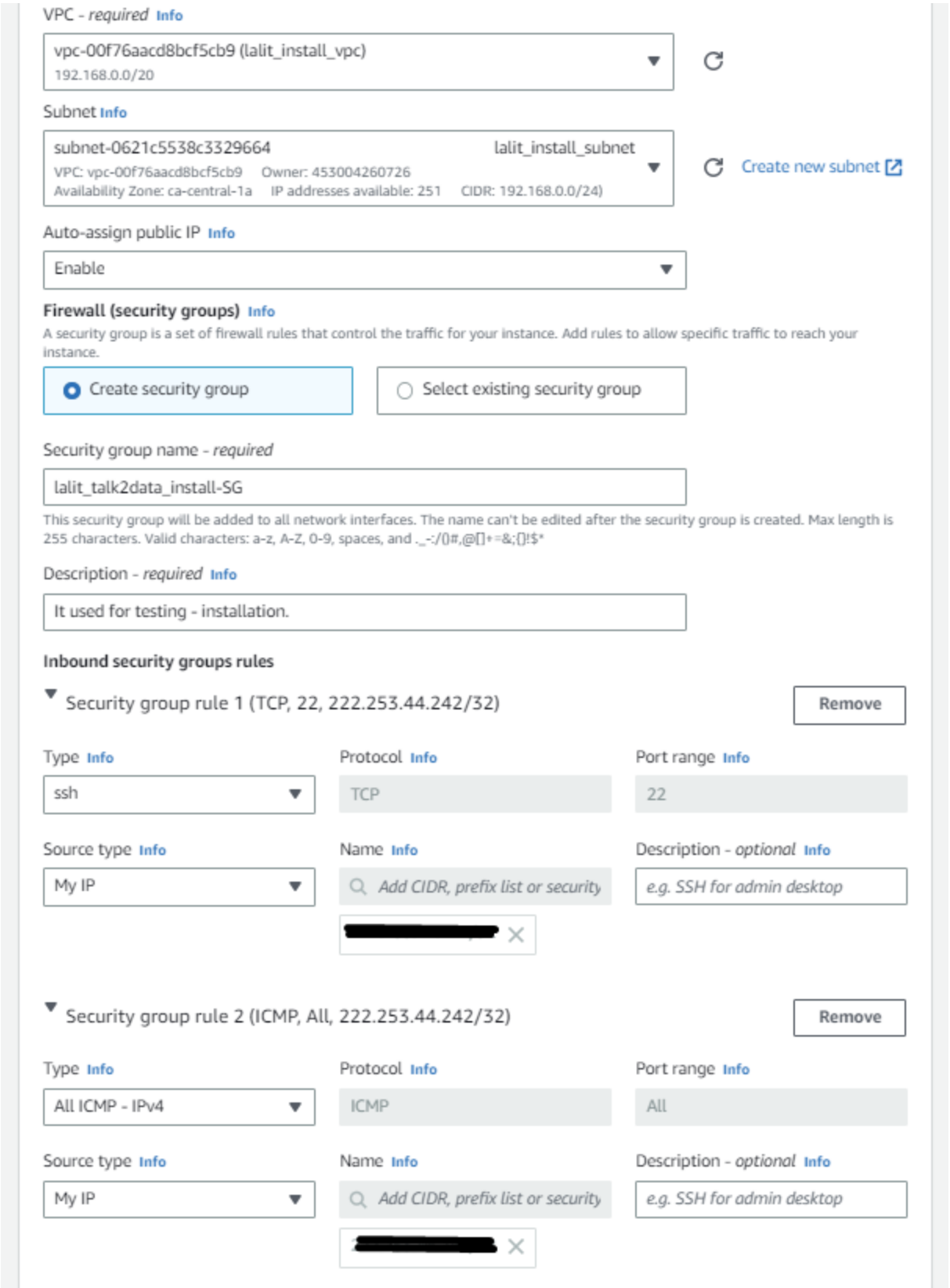


- Select the VPC, which we created earlier.

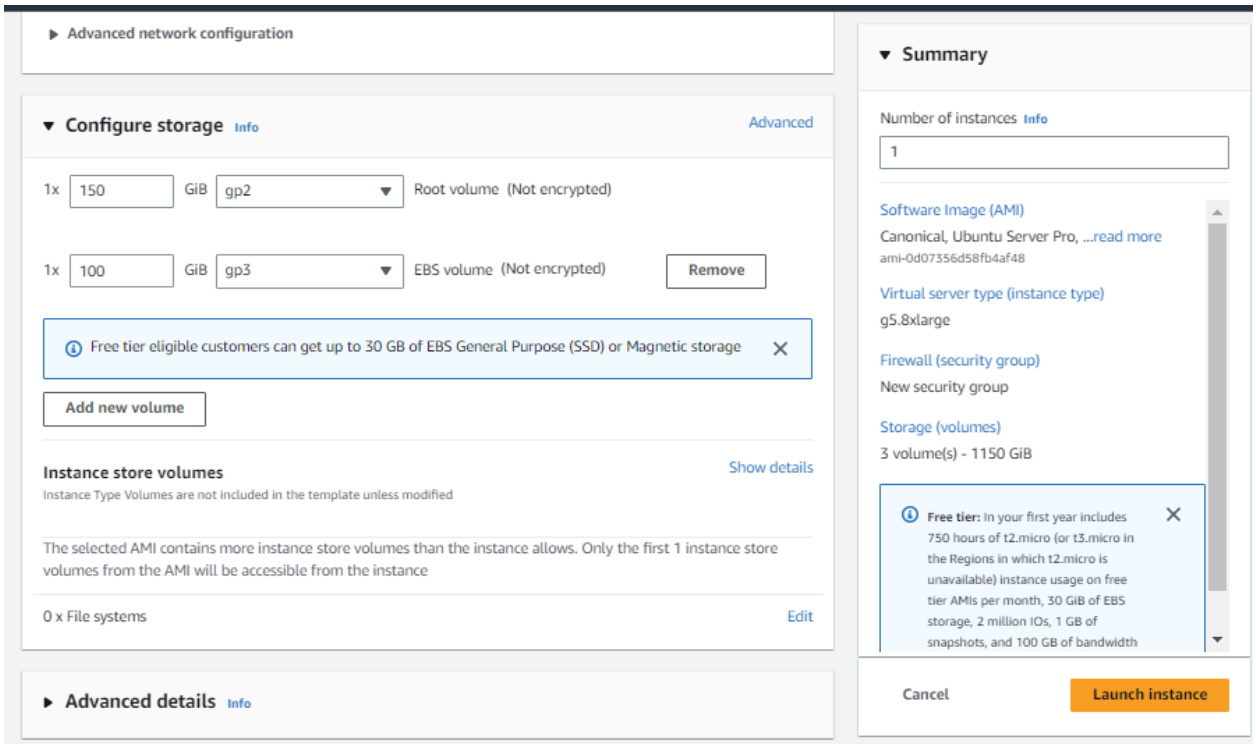


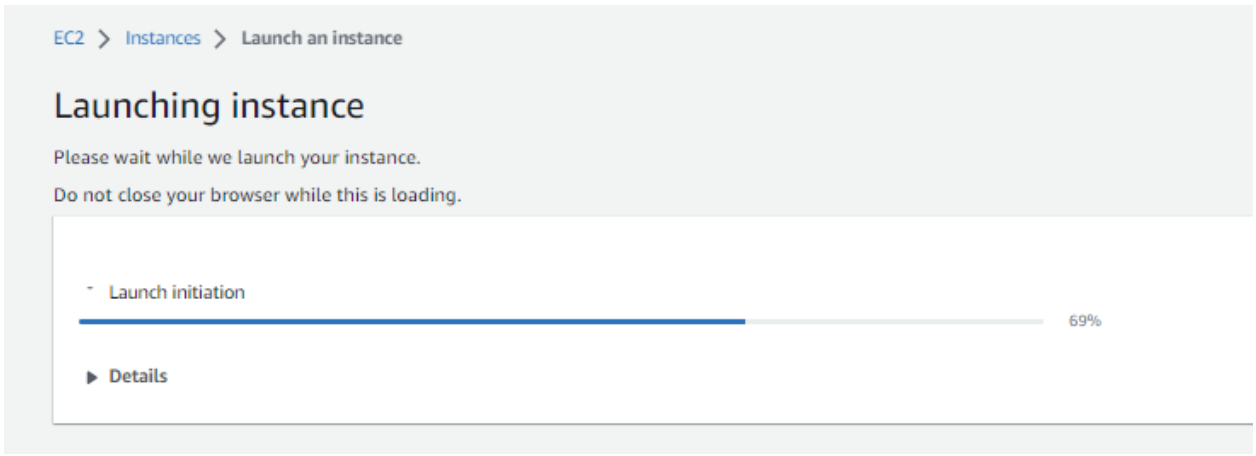


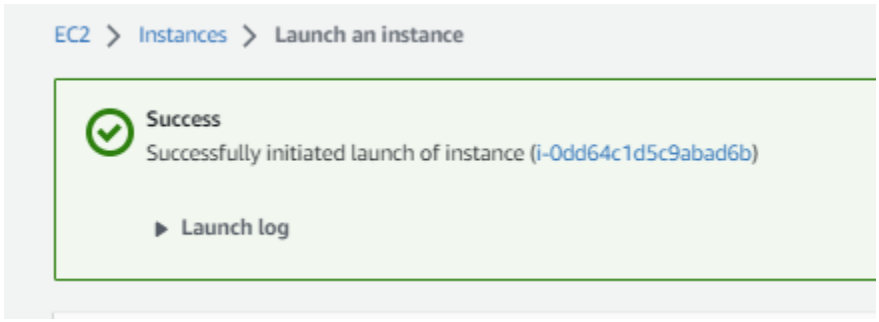
- Create Security Group based on requirement



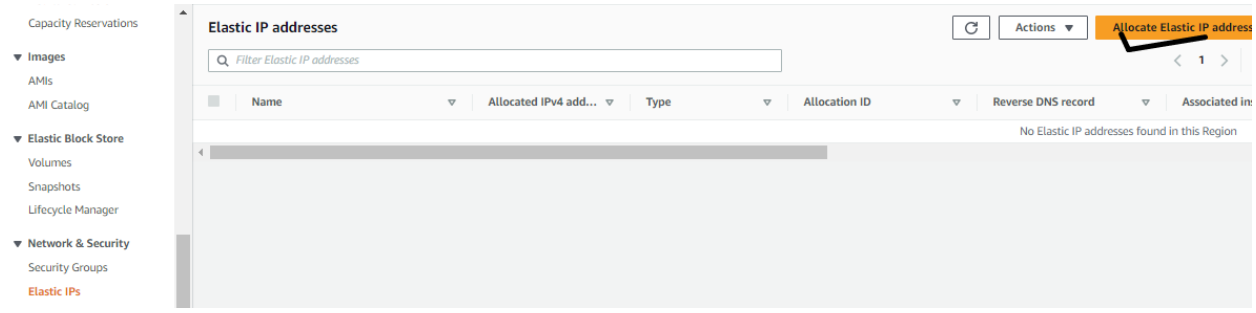
- Attach Disk for data volume







- Assign Elastic IP to this instance



Allocate Elastic IP address Info

Elastic IP address settings Info

Network Border Group Info

ca-central-1

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*

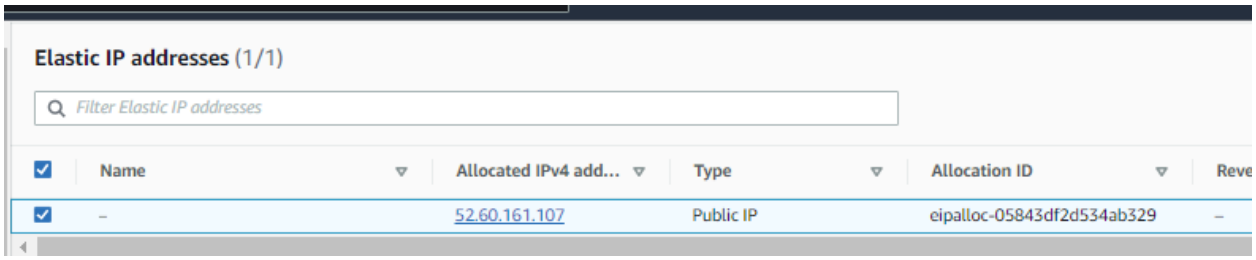
A tag is a label that you assign to an AWS resource. Each tag consists of a key and an optional value. You can use tags to search and filter your resources or track your AWS costs.

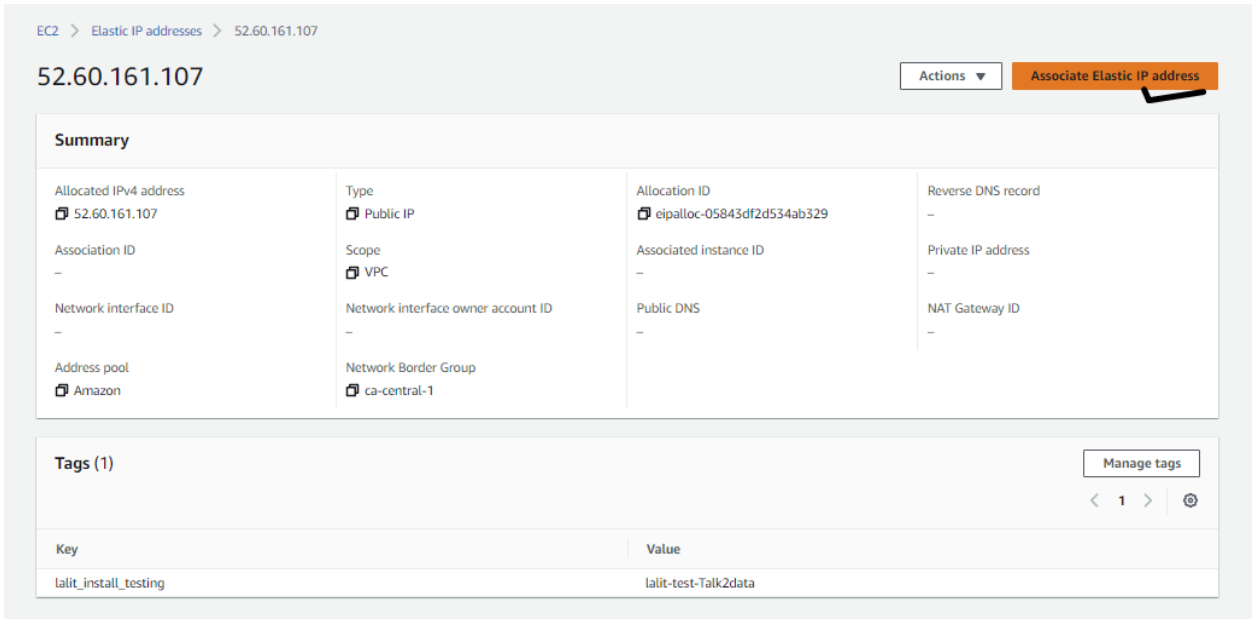
No tags associated with the resource.

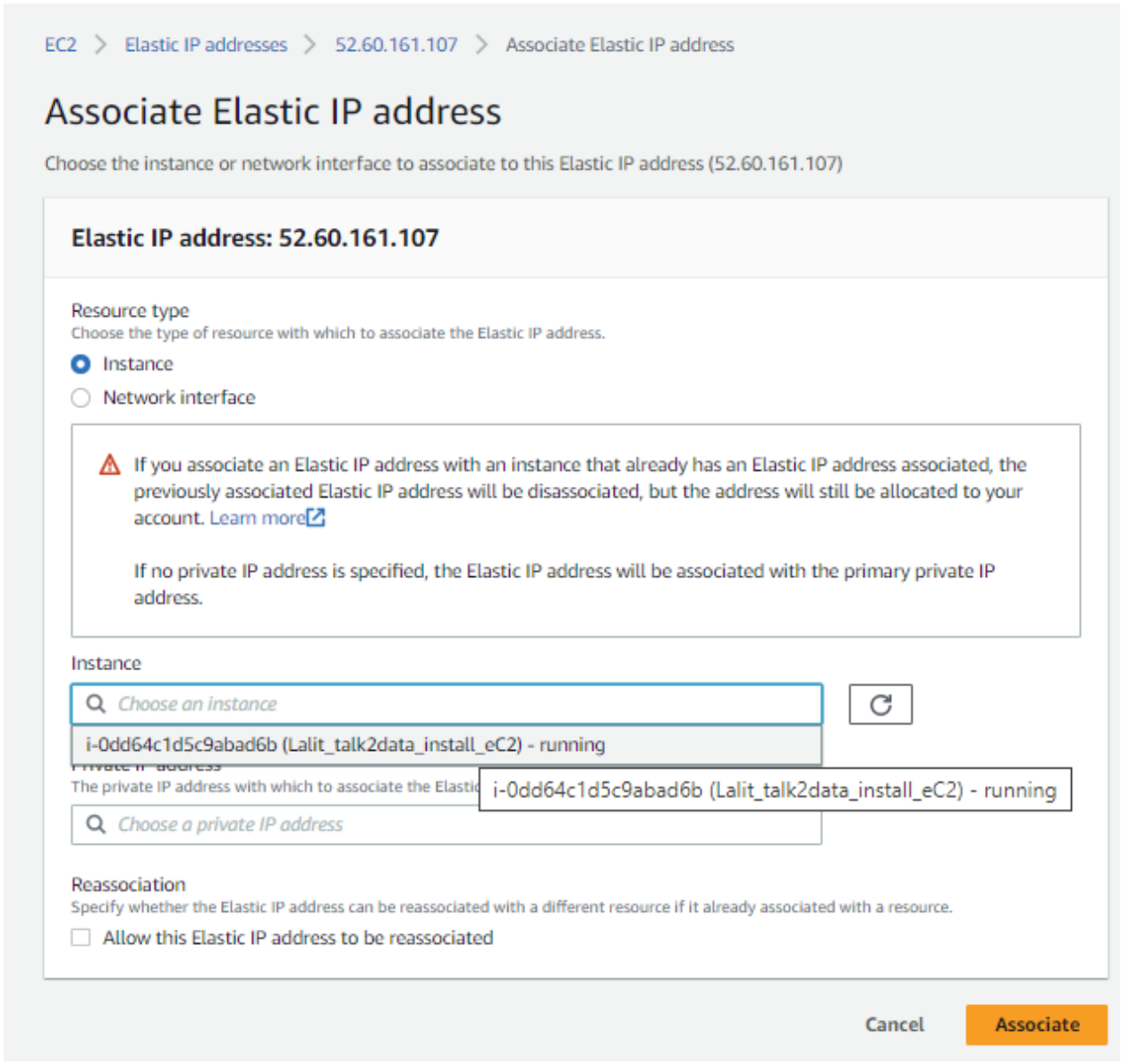
Add new tag

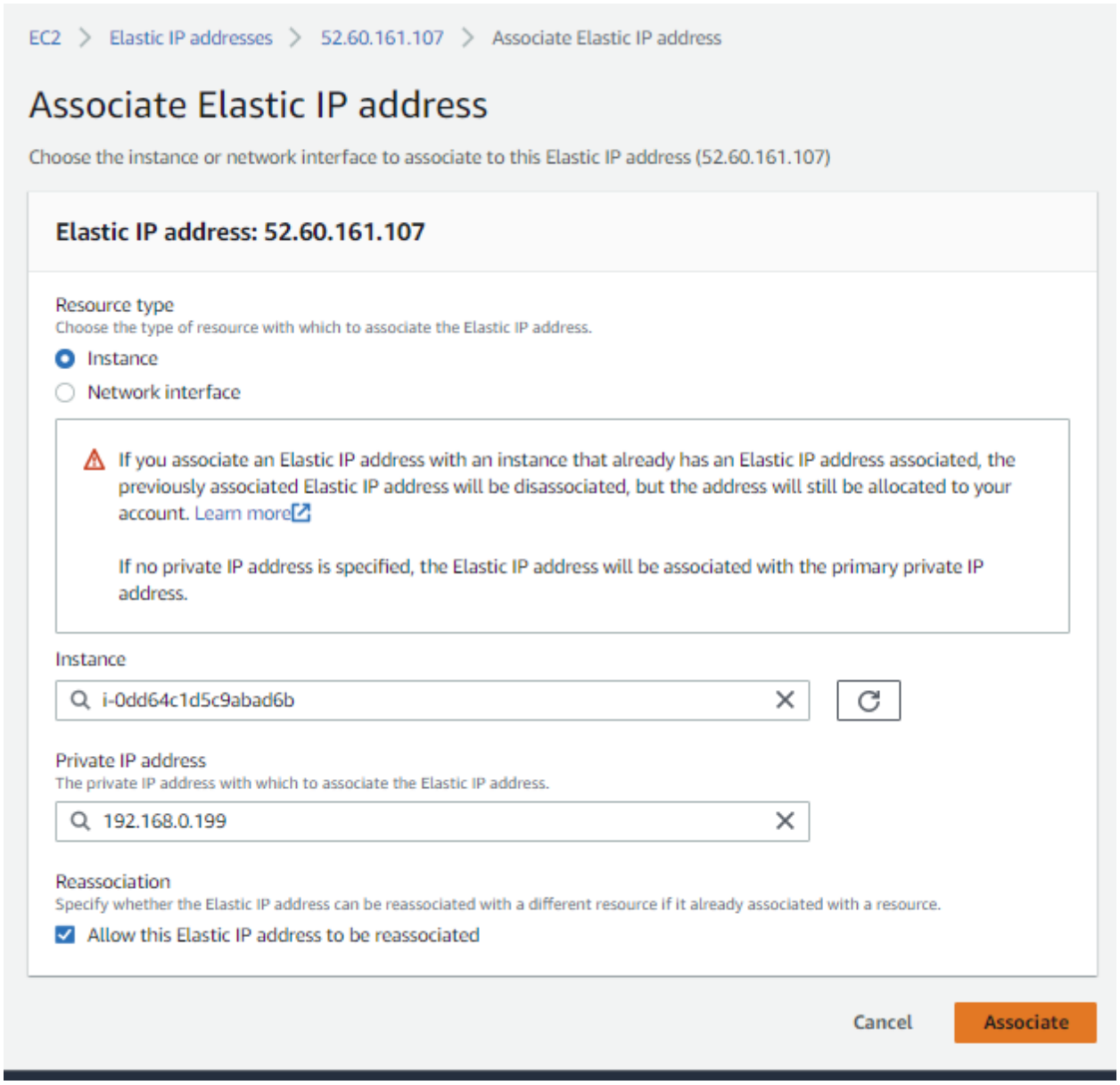
You can add up to 50 more tag

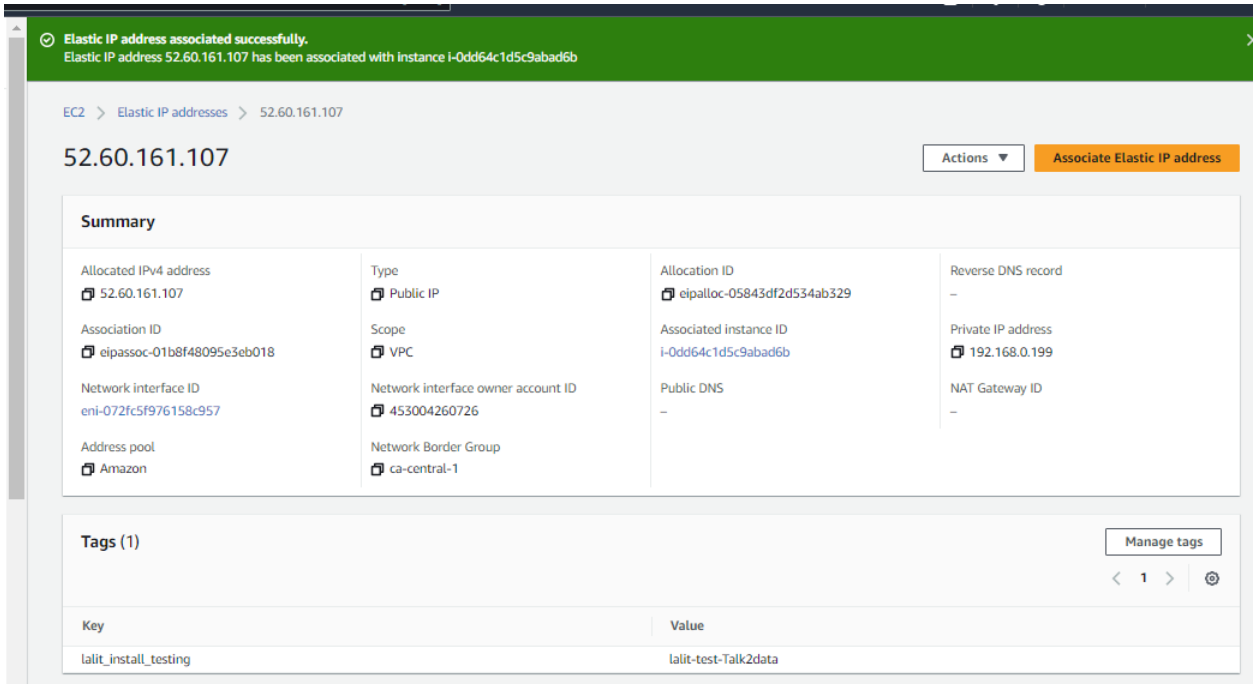
Cancel Allocate

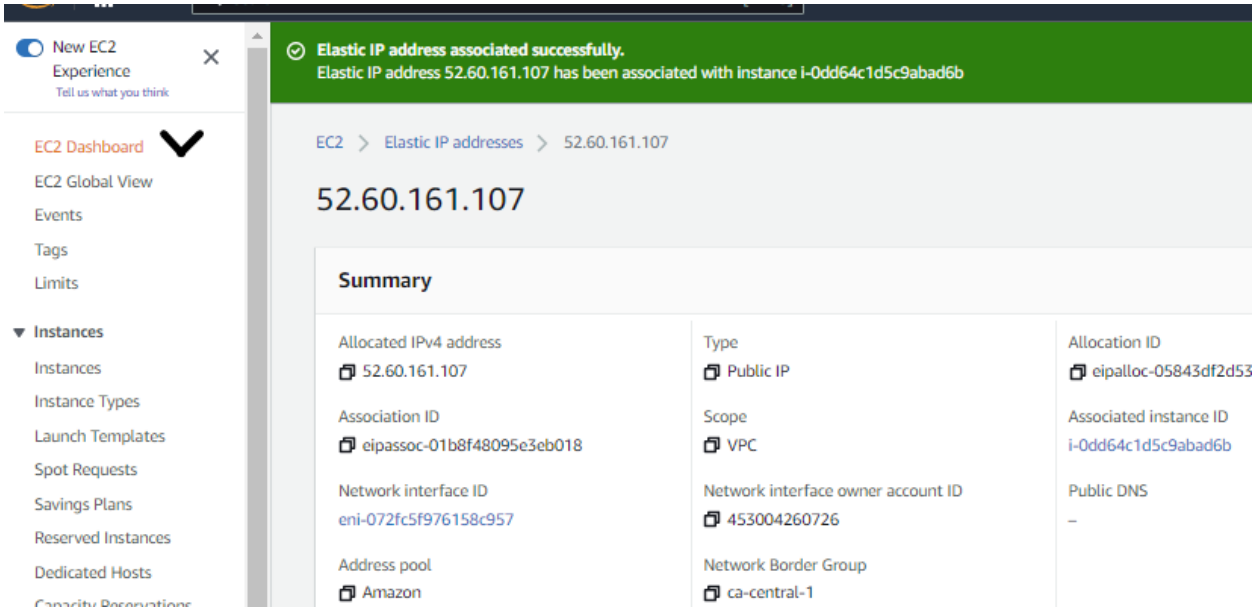

Elastic IP addresses (1/1)
Filter Elastic IP addresses
Name
Allocated IPv4 add...
Type
Allocation ID
Reve
–
52.60.161.107
Public IP
eipalloc-05843df2d534ab329
–


EC2 > Elastic IP addresses > 52.60.161.107
52.60.161.107
Actions
Associate Elastic IP address
Summary
Allocated IPv4 address
52.60.161.107
Type
Public IP
Allocation ID
eipalloc-05843df2d534ab329
Reverse DNS record
–
Association ID
–
Scope
VPC
Associated instance ID
–
Private IP address
–
Network interface ID
–
Network interface owner account ID
–
Public DNS
–
NAT Gateway ID
–
Address pool
Amazon
Network Border Group
ca-central-1
Tags (1)
Manage tags
1
Key
Value
lalit_install_testing
lalit-test-Talk2data

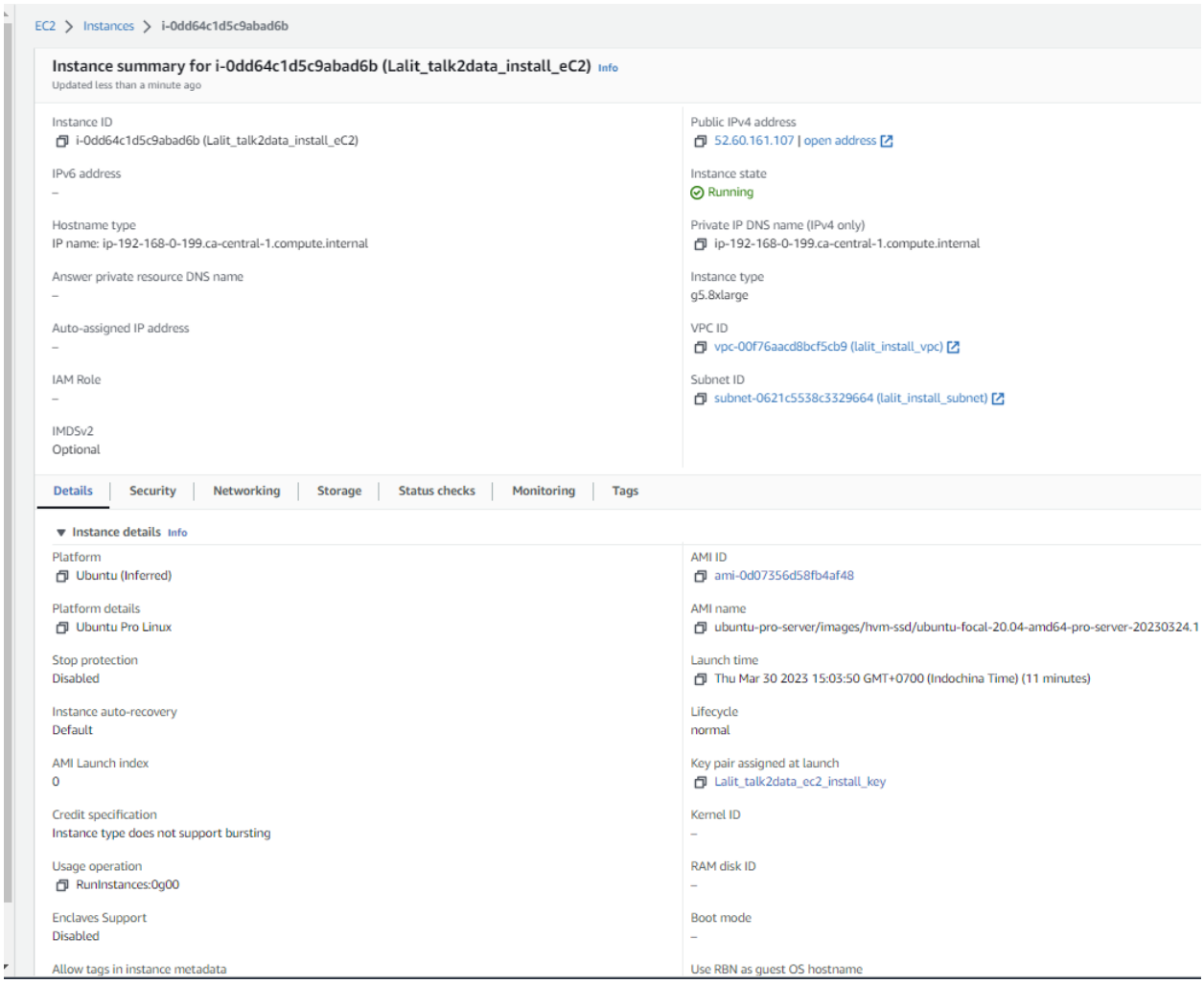

EC2 > Elastic IP addresses > 52.60.161.107 > Associate Elastic IP address
Associate Elastic IP address
Choose the instance or network interface to associate to this Elastic IP address (52.60.161.107)
Elastic IP address: 52.60.161.107
Resource type
Choose the type of resource with which to associate the Elastic IP address.
Instance
Network interface
If you associate an Elastic IP address with an instance that already has an Elastic IP address associated, the previously associated Elastic IP address will be disassociated, but the address will still be allocated to your account. Learn more
If no private IP address is specified, the Elastic IP address will be associated with the primary private IP address.
Instance
Choose an instance
i-0dd64c1d5c9abad6b (Lalit_talk2data_install_eC2) - running
The private IP address with which to associate the Elastic
i-0dd64c1d5c9abad6b (Lalit_talk2data_install_eC2) - running
Choose a private IP address
Reassociation
Specify whether the Elastic IP address can be reassociated with a different resource if it already associated with a resource.
Allow this Elastic IP address to be reassociated
Cancel
Associate


EC2 > Elastic IP addresses > 52.60.161.107 > Associate Elastic IP address
Associate Elastic IP address
Choose the instance or network interface to associate to this Elastic IP address (52.60.161.107)
Elastic IP address: 52.60.161.107
Resource type
Choose the type of resource with which to associate the Elastic IP address.
Instance
Network interface
If you associate an Elastic IP address with an instance that already has an Elastic IP address associated, the previously associated Elastic IP address will be disassociated, but the address will still be allocated to your account. Learn more
If no private IP address is specified, the Elastic IP address will be associated with the primary private IP address.
Instance
i-0dd64c1d5c9abad6b
Private IP address
The private IP address with which to associate the Elastic IP address.
192.168.0.199
Reassociation
Specify whether the Elastic IP address can be reassociated with a different resource if it already associated with a resource.
Allow this Elastic IP address to be reassociated
Cancel
Associate

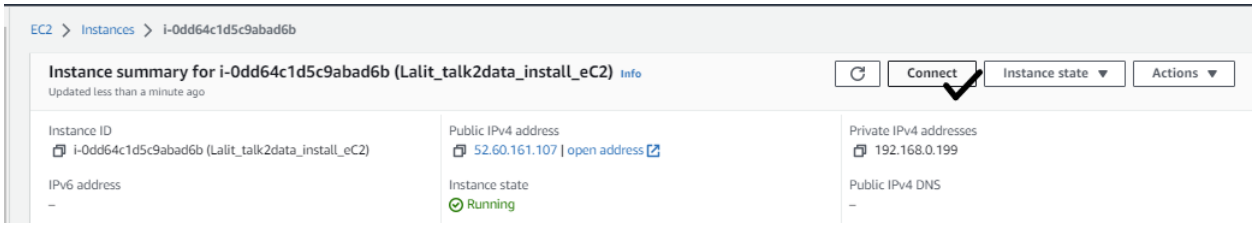




- Verify Public IP of instance



- Login to instance



- **Update the system:**
    - sudo apt update && sudo apt upgrade -y
    - sudo apt install build-essential wget curl gnupg lsb-release ca-certificates xfsprogs -y
- **Install NVIDIA CUDA Toolkit 11.7.**

- wget https://developer.download.nvidia.com/compute/cuda/11.7.1/local_installers/cuda_11.7.1_515.65.01_linux.run
- sudo sh cuda_11.7.1_515.65.01_linux.run

```
ubuntu@ip-192-168-0-199:~$ wget https://developer.download.nvidia.com/compute/cuda/11.7.1/local_installers/cuda_11.7.1_515.65.01_linux.run
nux.ru--2023-03-30 08:20:01--  https://developer.download.nvidia.com/compute/cuda/11.7.1/local_installers/cuda_11.7.1_515.65.01_linux.run
Resolving developer.download.nvidia.com (developer.download.nvidia.com)... 152.195.19.142
Connecting to developer.download.nvidia.com (developer.download.nvidia.com)|152.195.19.142|:443... nconnected.
HTTP request sent, awaiting response... 200 OK
Length: 3524358811 (3.3G) [application/octet-stream]
Saving to: ‘cuda_11.7.1_515.65.01_linux.run’

cuda_11.7.1_515.65.01_linux.run          100%[=====================================================================================>]   3.28G   110MB/s    in 28s

2023-03-30 08:20:29 (119 MB/s) - ‘cuda_11.7.1_515.65.01_linux.run’ saved [3524358811/3524358811]

ubuntu@ip-192-168-0-199:~$ sudo sh cuda_11.7.1_515.65.01_linux.run
```

```
ubuntu@ip-192-168-0-199:~$ sudo sh cuda_11.7.1_515.65.01_linux.run
 Failed to verify gcc version. See log at /var/log/cuda-installer.log for details.
ubuntu@ip-192-168-0-199:~$ cat /var/log/cuda-installer.log
[INFO]: Driver not installed.
[INFO]: Checking compiler version...
[INFO]: gcc location:
[ERROR]: Missing gcc. gcc is required to continue.
ubuntu@ip-192-168-0-199:~$ sudo apt install build-essential wget curl gnupg lsb-release ca-certificates xfsprogs -y
Reading package lists... Done
Building dependency tree
Reading state information... Done
lsb-release is already the newest version (11.1.0ubuntu2).
```

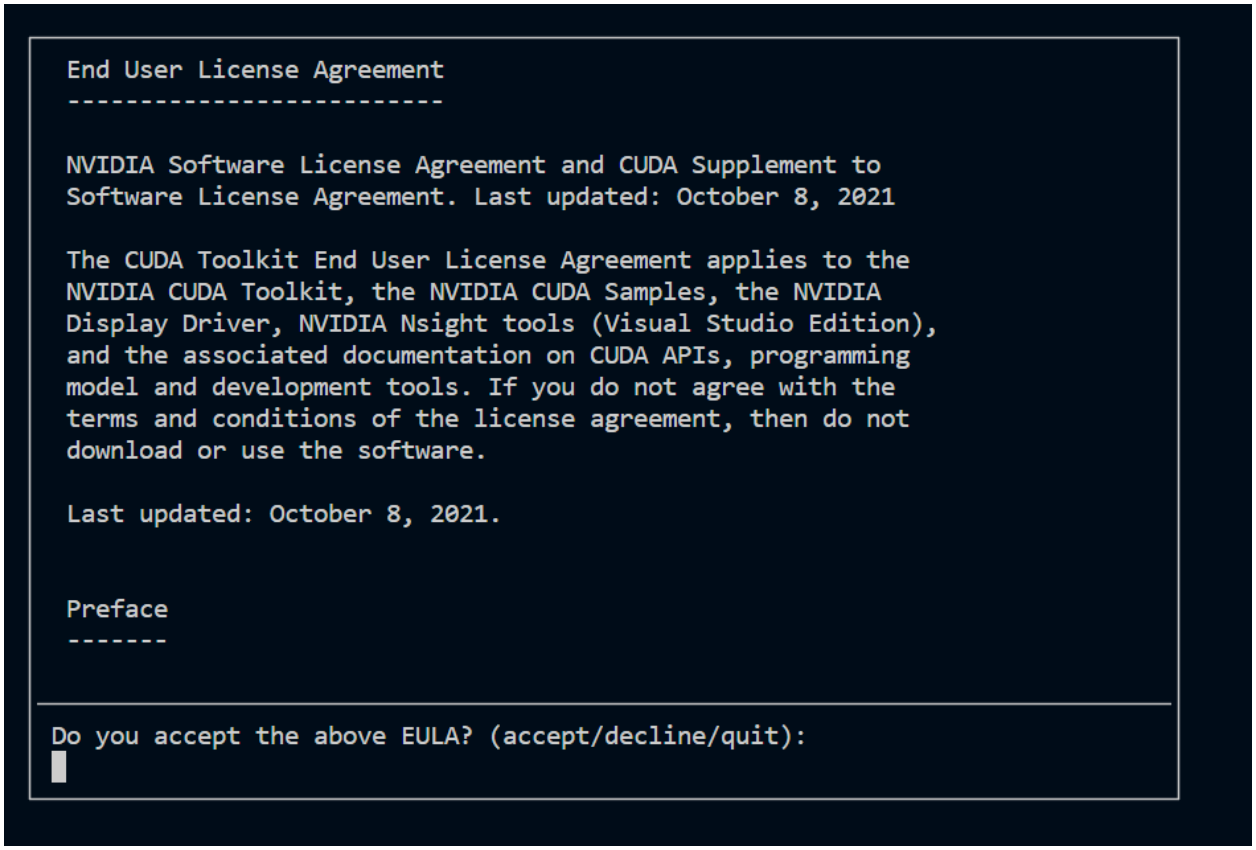
- **Reboot the server and try again.**

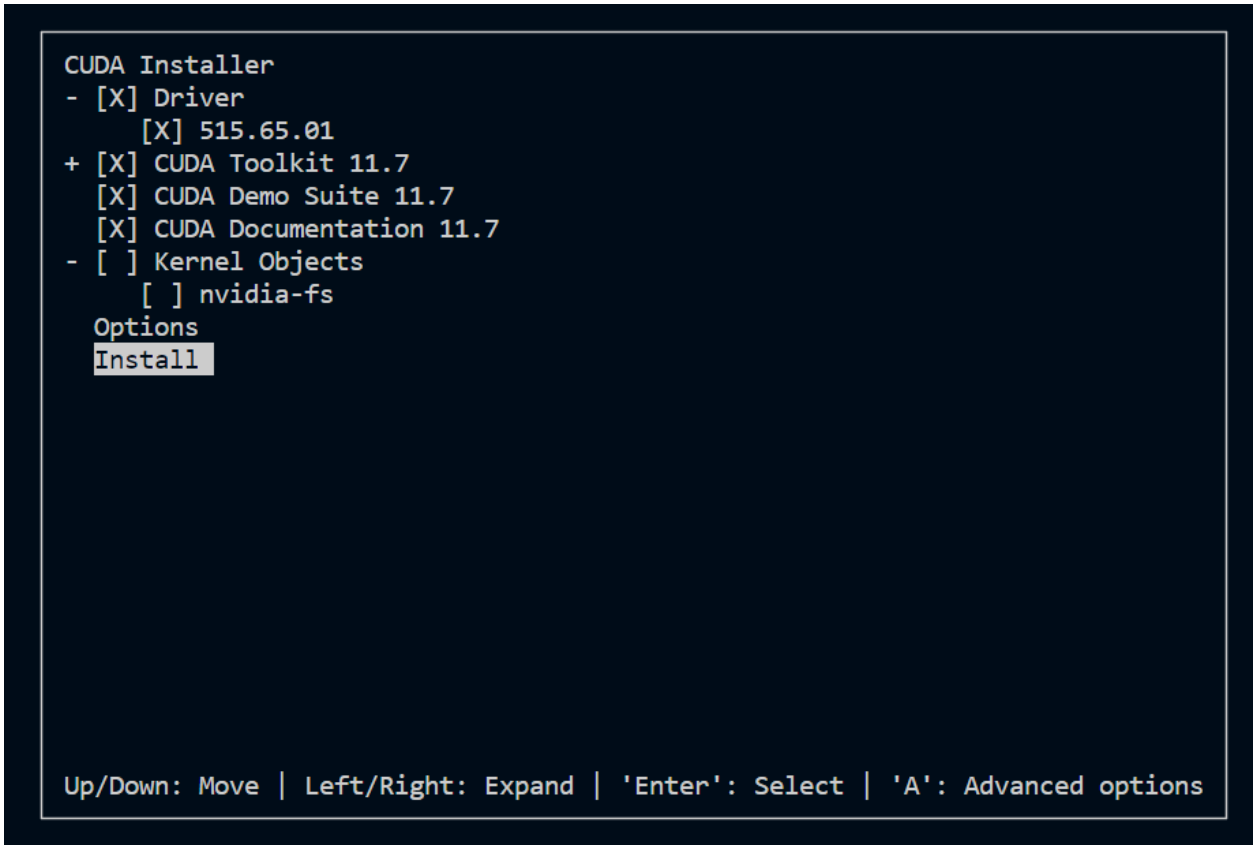
- lspci | grep -i nvidia
- uname -m && cat /etc/*release

```
Last login: Thu Mar 30 08:17:02 2023 from 222.253.44.242
ubuntu@ip-192-168-0-199:~$  gcc --version
gcc (Ubuntu 9.4.0-1ubuntu1~20.04.1) 9.4.0
Copyright (C) 2019 Free Software Foundation, Inc.
This is free software; see the source for copying conditions.  There is NO
warranty; not even for MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.

ubuntu@ip-192-168-0-199:~$ sudo sh cuda_11.7.1_515.65.01_linux.run
```

- **Accept license**


End User License Agreement
NVIDIA Software License Agreement and CUDA Supplement to
Software License Agreement. Last updated: October 8, 2021
The CUDA Toolkit End User License Agreement applies to the
NVIDIA CUDA Toolkit, the NVIDIA CUDA Samples, the NVIDIA
Display Driver, NVIDIA Nsight tools (Visual Studio Edition),
and the associated documentation on CUDA APIs, programming
model and development tools. If you do not agree with the
terms and conditions of the license agreement, then do not
download or use the software.
Last updated: October 8, 2021.
Preface
Do you accept the above EULA? (accept/decline/quit):


CUDA Installer
- [X] Driver
[X] 515.65.01
+ [X] CUDA Toolkit 11.7
[X] CUDA Demo Suite 11.7
[X] CUDA Documentation 11.7
- [ ] Kernel Objects
[ ] nvidia-fs
Options
Install
Up/Down: Move | Left/Right: Expand | 'Enter': Select | 'A': Advanced options

```
ubuntu@ip-192-168-0-199:~$ sudo sh cuda_11.7.1_515.65.01_linux.run
===========
= Summary =
===========

Driver:   Installed
Toolkit:  Installed in /usr/local/cuda-11.7/

Please make sure that
 -   PATH includes /usr/local/cuda-11.7/bin
 -   LD_LIBRARY_PATH includes /usr/local/cuda-11.7/lib64, or, add /usr/local/cuda-11.7/lib64 to /etc/ld.so.conf and run ldconfig as root

To uninstall the CUDA Toolkit, run cuda-uninstaller in /usr/local/cuda-11.7/bin
To uninstall the NVIDIA Driver, run nvidia-uninstall
Logfile is /var/log/cuda-installer.log
ubuntu@ip-192-168-0-199:~$
```

```
ubuntu@ip-192-168-0-199:~$ lspci | grep -i nvidia
00:1e.0 3D controller: NVIDIA Corporation Device 2237 (rev a1)
ubuntu@ip-192-168-0-199:~$ uname -m && cat /etc/*release
x86_64
DISTRIB_ID=Ubuntu
DISTRIB_RELEASE=20.04
DISTRIB_CODENAME=focal
DISTRIB_DESCRIPTION="Ubuntu 20.04.6 LTS"
NAME="Ubuntu"
VERSION="20.04.6 LTS (Focal Fossa)"
ID=ubuntu
ID_LIKE=debian
PRETTY_NAME="Ubuntu 20.04.6 LTS"
VERSION_ID="20.04"
HOME_URL="https://www.ubuntu.com/"
SUPPORT_URL="https://help.ubuntu.com/"
BUG_REPORT_URL="https://bugs.launchpad.net/ubuntu/"
PRIVACY_POLICY_URL="https://www.ubuntu.com/legal/terms-and-policies/privacy-policy"
VERSION_CODENAME=focal
UBUNTU_CODENAME=focal
ubuntu@ip-192-168-0-199:~$
0
```

- **nvidia-smi**

```
ubuntu@ip-192-168-0-199:~$ nvidia-smi
Thu Mar 30 08:36:04 2023
+-----------------------------------------------------------------------------+
| NVIDIA-SMI 515.65.01    Driver Version: 515.65.01    CUDA Version: 11.7     |
|-------------------------------+----------------------+----------------------+
| GPU  Name        Persistence-M| Bus-Id        Disp.A | Volatile Uncorr. ECC |
| Fan  Temp  Perf  Pwr:Usage/Cap|         Memory-Usage | GPU-Util  Compute M. |
|                               |                      |               MIG M. |
|===============================+======================+======================|
|   0  NVIDIA A10G         Off  | 00000000:00:1E.0 Off |                    0 |
|  0%   22C    P0    57W / 300W |      0MiB / 23028MiB |     10%      Default |
|                               |                      |                  N/A |
+-------------------------------+----------------------+----------------------+

+-----------------------------------------------------------------------------+
| Processes:                                                                  |
|  GPU   GI   CI        PID   Type   Process name                  GPU Memory |
|        ID   ID                                                   Usage      |
|=============================================================================|
|  No running processes found                                                 |
+-----------------------------------------------------------------------------+
ubuntu@ip-192-168-0-199:~$
```

```
ubuntu@ip-192-168-0-199:/$  nvidia-smi
Thu Mar 30 10:11:57 2023
+-----------------------------------------------------------------------------+
| NVIDIA-SMI 515.65.01    Driver Version: 515.65.01    CUDA Version: 11.7     |
|-------------------------------+----------------------+----------------------+
| GPU  Name        Persistence-M| Bus-Id        Disp.A | Volatile Uncorr. ECC |
| Fan  Temp  Perf  Pwr:Usage/Cap|         Memory-Usage | GPU-Util  Compute M. |
|                               |                      |               MIG M. |
|===============================+======================+======================|
|   0  NVIDIA A10G         Off  | 00000000:00:1E.0 Off |                    0 |
|  0%   25C    P0    56W / 300W |   2992MiB / 23028MiB |      0%      Default |
|                               |                      |                  N/A |
+-------------------------------+----------------------+----------------------+

+-----------------------------------------------------------------------------+
| Processes:                                                                  |
|  GPU   GI   CI        PID   Type   Process name                  GPU Memory |
|        ID   ID                                                   Usage      |
|=============================================================================|
|    0   N/A  N/A      7829      C   ...t2d_temp/tmpofbrlq3l/exec      249MiB |
|    0   N/A  N/A      7830      C   ...t2d_temp/tmpofbrlq3l/exec      249MiB |
|    0   N/A  N/A      7831      C   ...t2d_temp/tmpofbrlq3l/exec      249MiB |
|    0   N/A  N/A      7832      C   ...t2d_temp/tmpofbrlq3l/exec      249MiB |
|    0   N/A  N/A      8272      C   ...t2d_temp/tmp754zzxnp/exec      249MiB |
|    0   N/A  N/A      8273      C   ...t2d_temp/tmp754zzxnp/exec      249MiB |
|    0   N/A  N/A      8274      C   ...t2d_temp/tmp754zzxnp/exec      249MiB |
|    0   N/A  N/A      8275      C   ...t2d_temp/tmp754zzxnp/exec      249MiB |
|    0   N/A  N/A      8276      C   ...t2d_temp/tmp754zzxnp/exec      249MiB |
|    0   N/A  N/A      8278      C   ...t2d_temp/tmp754zzxnp/exec      249MiB |
|    0   N/A  N/A      8280      C   ...t2d_temp/tmp754zzxnp/exec      249MiB |
|    0   N/A  N/A      8281      C   ...t2d_temp/tmp754zzxnp/exec      249MiB |
+-----------------------------------------------------------------------------+
ubuntu@ip-192-168-0-199:/$
```

- **Install docker.**
    - curl https://get.docker.com | sh
    - sudo systemctl --now enable docker

Method 1: Using Docker

- **Configure DNS entry. We are using Cloudflare.**

- **Install the NVIDIA container toolkit.**

    - distribution=$(. /etc/os-release;echo IDVERSION_ID) \
      && curl -fsSL https://nvidia.github.io/libnvidia-container/gpgkey | sudo gpg --dearmor -o /usr/share/keyrings/nvidia-container-toolkit-keyring.gpg \
      && curl -s -L https://nvidia.github.io/libnvidia-container/$distribution/libnvidia-container.list | \
      sed 's#deb https://#deb [signed-by=/usr/share/keyrings/nvidia-container-toolkit-keyring.gpg] https://#g' | \
    - sudo tee /etc/apt/sources.list.d/nvidia-container-toolkit.list
    - sudo apt update
    - sudo apt install nvidia-docker2
    - sudo systemctl restart docker

    - nvidia-docker version

```
ubuntu@ip-192-168-0-199:~$  nvidia-docker version
NVIDIA Docker: 2.12.0
Client: Docker Engine - Community
 Version:           23.0.2
 API version:       1.42
 Go version:        go1.19.7
 Git commit:        569dd73
 Built:             Mon Mar 27 16:16:18 2023
 OS/Arch:           linux/amd64
 Context:           default
permission denied while trying to connect to the Docker daemon socket at unix:///var/run/docker.sock: Get "http://%2Fvar%2Frun%2Fdocker.sock/v1.24/version": dial
unix /var/run/docker.sock: connect: permission denied
ubuntu@ip-192-168-0-199:~$
```

- **Issue:**

permission denied while trying to connect to the Docker daemon socket at unix:///var/run/docker.sock: Get "http://%2Fvar%2Frun%2Fdocker.sock/v1.24/version": dial unix /var/run/docker.sock: connect: permission denied
ubuntu@ip-192-168-0-199:~$ ls -lhart /var/run/docker.sock
srw-rw---- 1 root docker 0 Mar 30 08:59 /var/run/docker.sock

ubuntu@ip-192-168-0-199:~$ sudo chmod 666 /var/run/docker.sock

```
permission denied while trying to connect to the Docker daemon socket at unix:///var/run/docker.sock: Get "http://%2Fvar%2Frun%2Fdocker.sock/v1.24/version": dial
unix /var/run/docker.sock: connect: permission denied
ubuntu@ip-192-168-0-199:~$ ls -lhart /var/run/docker.sock
srw-rw---- 1 root docker 0 Mar 30 08:59 /var/run/docker.sock
ubuntu@ip-192-168-0-199:~$ sudo chmod 666 /var/run/docker.sock
ubuntu@ip-192-168-0-199:~$ nvidia-docker version
NVIDIA Docker: 2.12.0
Client: Docker Engine - Community
 Version:           23.0.2
 API version:       1.42
 Go version:        go1.19.7
 Git commit:        569dd73
 Built:             Mon Mar 27 16:16:18 2023
 OS/Arch:           linux/amd64
 Context:           default

Server: Docker Engine - Community
 Engine:
  Version:          23.0.2
  API version:      1.42 (minimum version 1.12)
  Go version:       go1.19.7
  Git commit:       219f21b
  Built:            Mon Mar 27 16:16:18 2023
  OS/Arch:          linux/amd64
  Experimental:     false
 containerd:
  Version:          1.6.19
  GitCommit:        1e1ea6e986c6c86565bc33d52e34b81b3e2bc71f
 runc:
  Version:          1.1.4
  GitCommit:        v1.1.4-0-g5fd4c4d
 docker-init:
  Version:          0.19.0
  GitCommit:        de40ad0
ubuntu@ip-192-168-0-199:~$
```

- **Make sure that /dev/shm size is at least half of physical memory.**
  To change the configuration for /dev/shm, add one line to /etc/fstab. For example, if the system has 128 GB of physical memory:
    - tmpfs /dev/shm tmpfs defaults,size=64g 0 0
- **Run the command below to make the change immediately:**
    - sudo mount -o remount /dev/shm

```
ubuntu@ip-192-168-0-199:~$ sudo vi /etc/fstab
ubuntu@ip-192-168-0-199:~$ sudo mount -o remount /dev/shm
```

```
ubuntu@ip-192-168-0-199:~$ df -h
Filesystem       Size  Used Avail Use% Mounted on
/dev/root        146G   13G  133G   9% /
devtmpfs          63G     0   63G   0% /dev
tmpfs             64G     0   64G   0% /dev/shm
tmpfs             13G  1.1M   13G   1% /run
tmpfs            5.0M     0  5.0M   0% /run/lock
tmpfs             63G     0   63G   0% /sys/fs/cgroup
/dev/loop0        25M   25M     0 100% /snap/amazon-ssm-agent/6312
/dev/loop1       9.0M  9.0M     0 100% /snap/canonical-livepatch/164
/dev/loop2        64M   64M     0 100% /snap/core20/1852
/dev/nvme0n1p15  105M  6.1M   99M   6% /boot/efi
/dev/loop3       117M  117M     0 100% /snap/core/14946
/dev/loop4        92M   92M     0 100% /snap/lxd/24061
/dev/loop5        50M   50M     0 100% /snap/snapd/18596
/dev/loop6        56M   56M     0 100% /snap/core18/2714
tmpfs             13G     0   13G   0% /run/user/1000
/dev/loop7        44M   44M     0 100% /snap/certbot/2836
ubuntu@ip-192-168-0-199:~$
```

    - sudo mkdir -p /config/ssl
    - sudo mv tls.crt /config/ssl
    - sudo mv tls.key /config/ssl

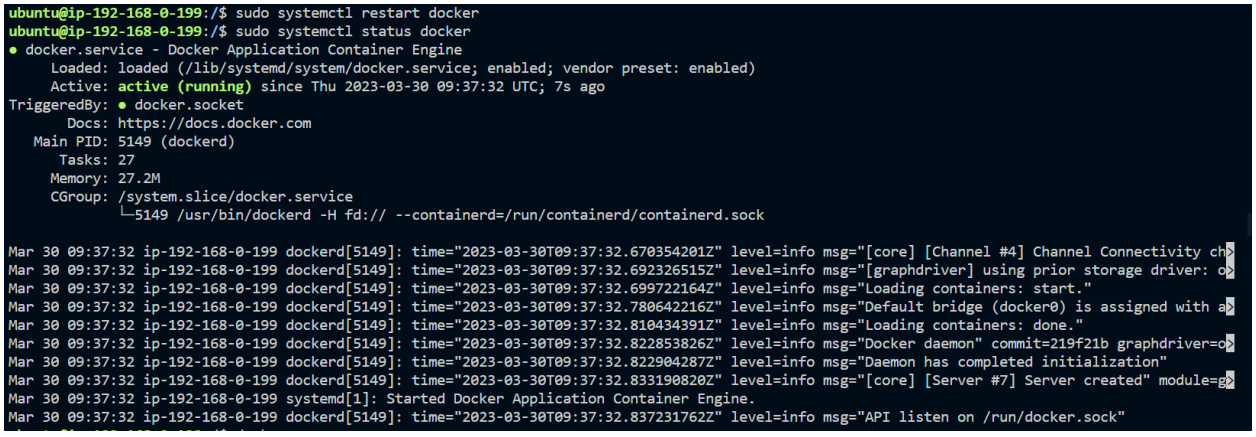
- **Create default directories to store user data. We highly recommend using persistent storage for these directories. In the commands below, we use an empty EBS volume.**
- **Pull the BBrowserX image.**

    - sudo mkfs -t ext4 /dev/nvme2n1
    - sudo mkdir /data
    - sudo mount /dev/nvme2n1 /data
    - sudo mkdir /data/app_data
    - sudo mkdir /data/user_data

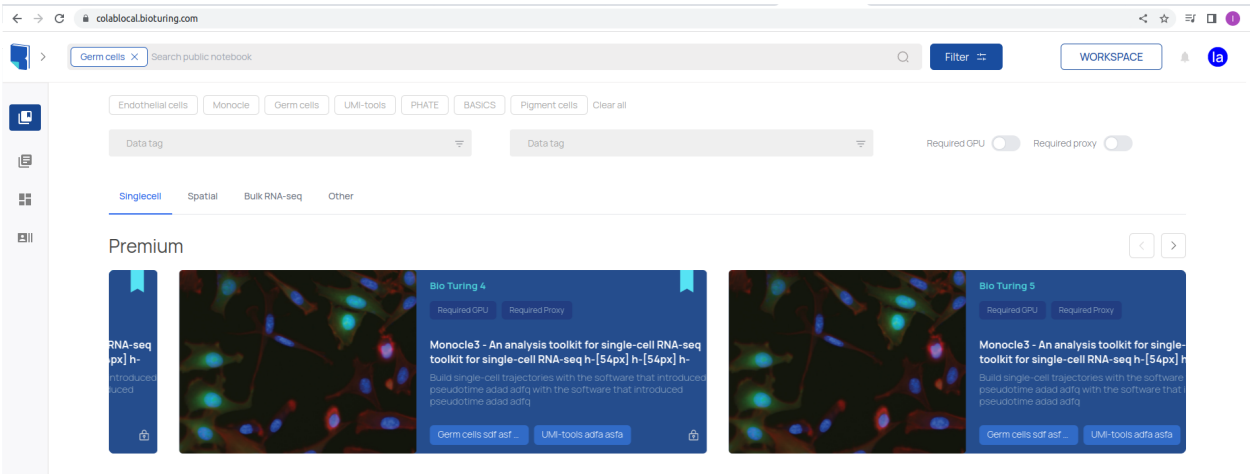
- sudo docker pull bioturing/bioturing-colab:1.0.1

- **Run the docker image.**

```
docker run -it -d \
   -e WEB_DOMAIN='<yourcompany.com>' \
   -e BIOTURING_TOKEN='<your token from BioTuring>' \
   -e SSO_DOMAINS='<your company email address, example: @bioturing.com>' \
   -e ADMIN_USERNAME='<your admin user name>' \
   -e ADMIN_PASSWORD='<your admin password>' \
   -v /data/user_data:/data/user_data \
   -v /data/app_data:/data/app_data \
   --name bioturing-colab \
   --gpus all \
   --shm-size=64gb \
   -p 443:443 \
   -p 80:80 \
   bioturing/bioturing-colab:1.0.1
```



Wait for a few minutes for the platform to download all of the required services. After that, the BioTuring Colab is up and running.



Method 2: Using Kubernetes

========== Reference

https://www.howtoforge.com/how-to-install-containerd-container-runtime-on-ubuntu-22-04/#comments

https://blog.antosubash.com/posts/setup-micro-k8s-with-ubuntu

- Patch container engines (Docker, Containerd)

Install NVidia container toolkit on each node following the guide:
https://docs.nvidia.com/datacenter/cloud-native/container-toolkit/install-guide.html

Check container engines (Docker, Containerd)

https://github.com/bioturing/installation

- **Check container engines (Docker, Containerd)**
  For microk8s :
  - microk8s kubectl describe no | grep Runtime

  For vanilla :
  - kubectl describe no | grep Runtime
- **If container engine is Containerd, add these lines to : /etc/containerd/config.toml**

```
privileged_without_host_devices = false
base_runtime_spec = ""
[plugins."io.containerd.grpc.v1.cri".containerd.runtimes.runc.options]
  SystemdCgroup = true
[plugins."io.containerd.grpc.v1.cri".containerd.runtimes.nvidia]
  privileged_without_host_devices = false
  runtime_engine = ""
  runtime_root = ""
  runtime_type = "io.containerd.runc.v1"
  [plugins."io.containerd.grpc.v1.cri".containerd.runtimes.nvidia.options]
  BinaryName = "/usr/bin/nvidia-container-runtime"
  SystemdCgroup = true
[plugins."io.containerd.grpc.v1.cri".cni]
bin_dir = "/opt/cni/bin"
conf_dir = "/etc/cni/net.d"
```

- **After that, restart containerd**
  - sudo systemctl restart containerd

    - sudo nvidia-container-cli --load-kmods info

- **If container engine is Docker, add these lines to : /etc/docker/daemon.json**

```
{
   "default-runtime": "nvidia",
   "runtimes": {
      "nvidia": {
         "path": "nvidia-container-runtime",
         "runtimeArgs": []
      }
   }
}
```

- **After that, restart docker**
    - sudo systemctl restart docker
    - sudo nvidia-container-cli --load-kmods info

```
ubuntu@ip-192-168-0-186:~$ microk8s kubectl describe no | grep Runtime
  Container Runtime Version:  containerd://1.6.8
ubuntu@ip-192-168-0-186:~$ cat /etc/docker/daemon.json
{
    "runtimes": {
        "nvidia": {
            "path": "nvidia-container-runtime",
            "runtimeArgs": []
        }
    }
}
ubuntu@ip-192-168-0-186:~$ docker -v
Docker version 23.0.2, build 569dd73
ubuntu@ip-192-168-0-186:~$ sudo systemctl restart docker
do nvidia-container-cli --load-kmods infoubuntu@ip-192-168-0-186:~$ sudo nvidia-container-cli --load-kmods info
NVRM version:   515.65.01
CUDA version:   11.7

Device Index:   0
Device Minor:   0
Model:          NVIDIA A10G
Brand:          Nvidia
GPU UUID:       GPU-fd9ffd32-13f4-61dd-ce58-197c9f1b9fb5
Bus Location:   00000000:00:1e.0
Architecture:   8.6
```

- **Install BioTuring Colab on K8S**

  We support all k8s engines: GKE (Google Kubernetes Engine), EKS (Amazon Elastic Kubernetes Service), AKS (Azure Kubernetes Service), MicroK8s, and vanilla K8S.

  Ensure that helm (version 3) is installed.

  First, check the Helm version

  - microk8s enable helm3
  - microk8s helm3 version

- **Add BioTuring Helm charts**
  - https://bioturing.github.io/charts/ : for kubernetes

Example:

For Vanilla K8s:

- helm repo add bioturing https://bioturing.github.io/charts for Microk8s:
- microk8s helm3 repo add bioturing https://bioturing.github.io/charts

```
ubuntu@ip-192-168-0-186:~$ microk8s enable helm3
Infer repository core for addon helm3
Addon core/helm3 is already enabled
ubuntu@ip-192-168-0-186:~$ microk8s helm3 version
version.BuildInfo{Version:"v3.9.1+unreleased", GitCommit:"7112315b8d78eab23e9542c4fd824375429ca965", GitTreeState:"clean", GoVersion:"go1.19.5"}
ubuntu@ip-192-168-0-186:~$ microk8s helm3 repo add bioturing https://bioturing.github.io/charts/apps/
"bioturing" already exists with the same configuration, skipping
ubuntu@ip-192-168-0-186:~$
```

Helm chart Values
Kubernetes: >=

Key	Type	Default	Description
image.tag	string	"1.0.1"	image tag
secret.data.domain	string	"colab.com"	your domain
secret.data.ssodomains	string	""	allow domains
secret.data.bbtoken	string	""	bioturing access token
secret.data.allowips	string	""	allow ips
secret.admin.username	string	admin	username
secret.admin.password	string	turing2022	password
secret.server.useletsencrypt	string	"false"	
secret.server.lcall	string	"C.UTF-8"	
secret.server.lclang	string	"C.UTF-8"	
secret.server.certificate	string	""	CRT base64 string
secret.server.key	string	""	KEY base64 string
service.type	string	ClusterIP	
service.ports.http.port	int	80	
service.ports.https.port	int	443	
persistence.dirs.app.size	string	5Gi	APP size
persistence.dirs.app.storageClass	string	""	
persistence.dirs.user.size	string	5Gi	USER size
persistence.dirs.shm.size	string	1Gi	SHM size
persistence.dirs.user.storageClass	string	""	
persistence.dirs.user.existingClaim	bool	FALSE	
ingress.enabled	bool	TRUE	
ingress.className	string	""	
ingress.annotations	object	{}	
ingress.tls.enabled	bool	TRUE	
resources	object	{}	
autoscaling	object	{}	

nodeSelector	object	{}	
tolerations	object	{}	
affinity	object	{}	
podAnnotations	object	{}	
podSecurityContext	object	{}	
securityContext	object	{}	
serviceAccount.name	string	""	
gpu.enabled	bool	TRUE	
gpu.runtimeClassName	string	"nvidia"	

- **For Containerd runtime :**

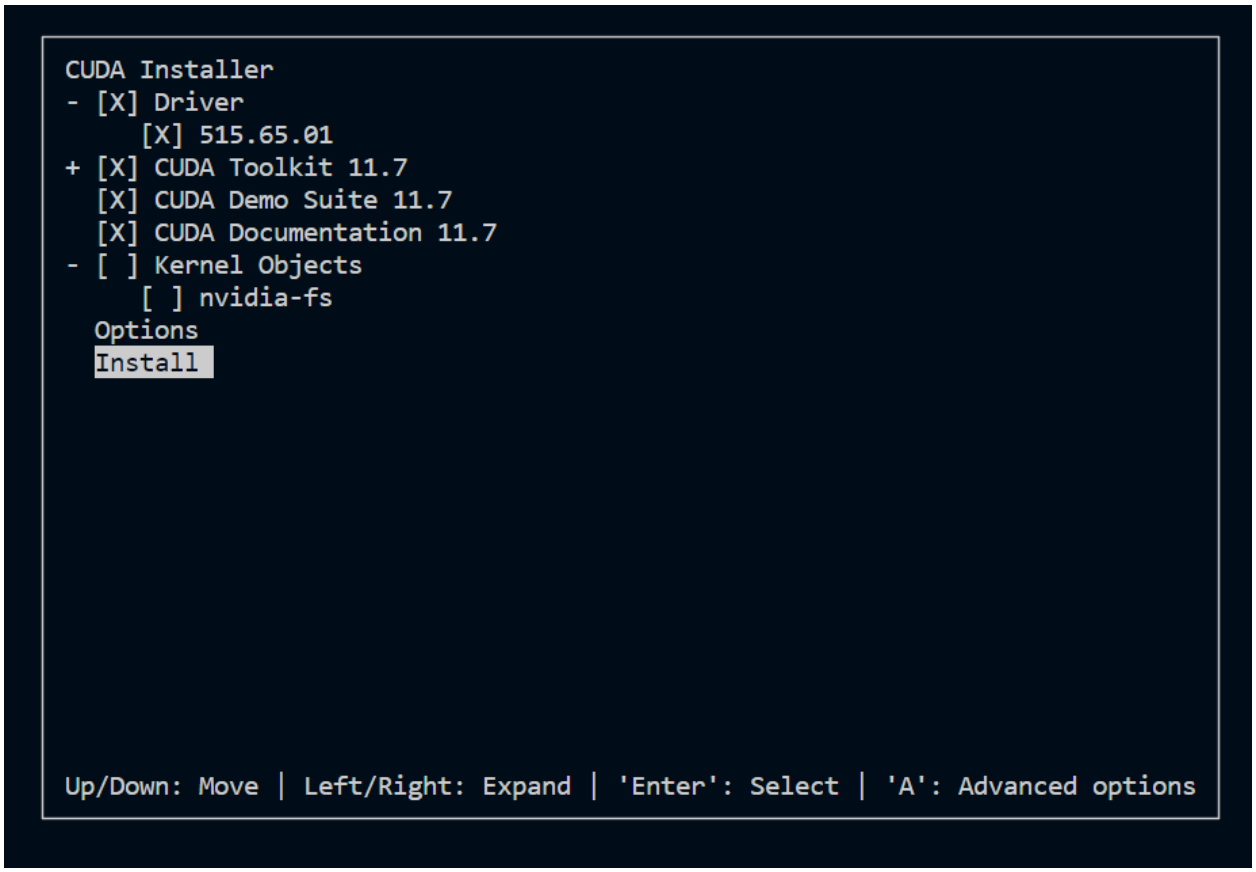
  gpu.runtimeClassName="nvidia"

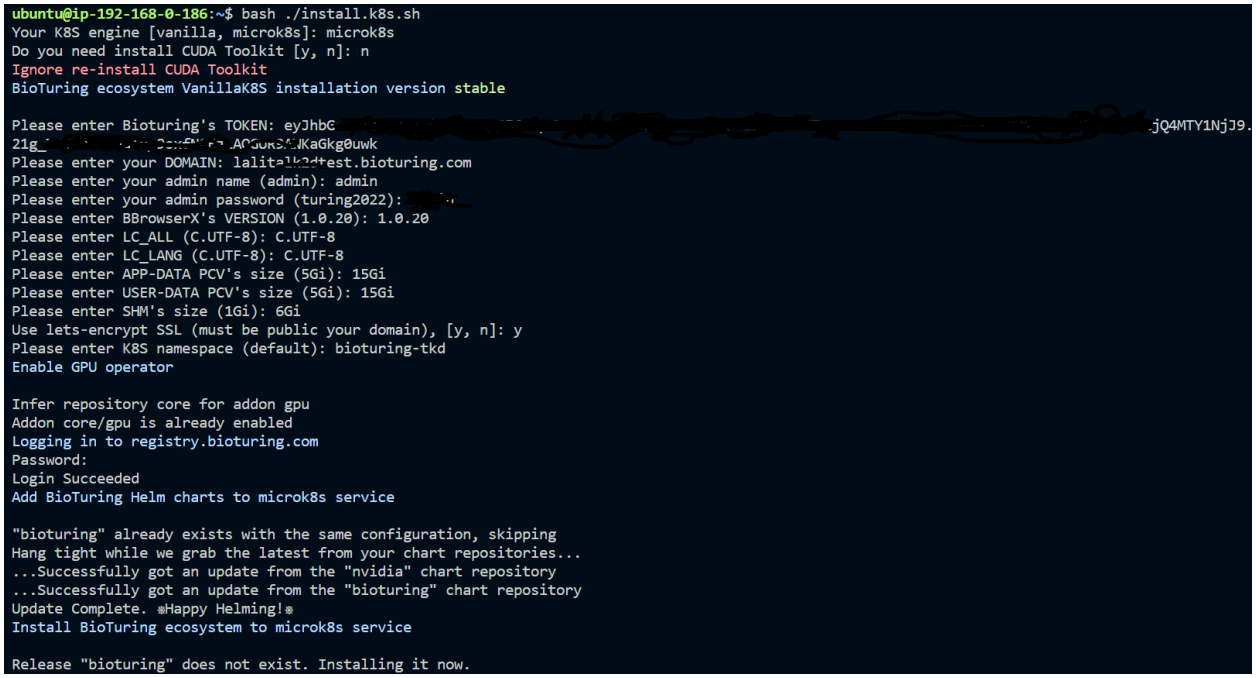
- **For Docker runtime :**

  gpu.runtimeClassName=""

- **Simple Installation (Recommended):**


End User License Agreement
NVIDIA Software License Agreement and CUDA Supplement to
Software License Agreement. Last updated: October 8, 2021
The CUDA Toolkit End User License Agreement applies to the
NVIDIA CUDA Toolkit, the NVIDIA CUDA Samples, the NVIDIA
Display Driver, NVIDIA Nsight tools (Visual Studio Edition),
and the associated documentation on CUDA APIs, programming
model and development tools. If you do not agree with the
terms and conditions of the license agreement, then do not
download or use the software.
Last updated: October 8, 2021.
Preface
Do you accept the above EULA? (accept/decline/quit):
accept


CUDA Installer
- [X] Driver
[X] 515.65.01
+ [X] CUDA Toolkit 11.7
[X] CUDA Demo Suite 11.7
[X] CUDA Documentation 11.7
- [ ] Kernel Objects
[ ] nvidia-fs
Options
Install
Up/Down: Move | Left/Right: Expand | 'Enter': Select | 'A': Advanced options



Going through this interactive installation to finish the installation. After this step, just access the BioTuring System via the specified domain in the installation process. If it's not in the DNS, please add the ip/domain to the local machine DNS host file.

Check pods information

- microk8s kubectl get all
- microk8s kubectl get pods
- microk8s kubectl get services --all-namespaces
- microk8s kubectl get services
- microk8s kubectl get pvc
- microk8s kubectl logs bioturing-colab
- microk8s.kubectl -n ingress get pods
- microk8s.kubectl -n ingress logs <your pod name here> | grep reload

Check secrets

- bioturing-colab-tls
- bioturing-colab
- bioturingregred

microk8s kubectl edit secrets mysecret

Example:

microk8s kubectl edit secrets bioturing-colab-tls

Manual Installation

Please replace paths to your certificate, key, admin password, and other helm chart values of your choice.

```
BBTOKEN="USE TOKEN OBTAINED FROM BIOTURING"
SSLCRT="base64 -w 0 ./bioturing.com.crt" # <- (REPLACE THIS WITH A PATH TO YOUR CRT CERTFICATE)
SSLKEY="base64 -w 0 ./bioturing.com.key" # <- (REPLACE THIS WITH A PATH TO YOUR KEY)
ADMIN_USERNAME="admin"
ADMIN_PASSWORD="admin" # <- (CHANGE YOUR PASSWORD IF NECESSARY)
USELETSENCRYPT="false"
SVHOST="k8stest.bioturing.com" # <- (CHANGE THIS TO YOUR K8S INGRESS DOMAIN)
APP_DATA_SIZE="50Gi" # <- (CHANGE THIS TO YOUR APP-PVC SIZE)
USER_DATA_SIZE="100Gi" # <- (CHANGE THIS TO YOUR USER-PVC SIZE)
SHM_SIZE="64Gi" # <- (CHANGE THIS TO YOUR SHM SIZE)
CHART_VERSION="1.0.23" # <- (CHANGE IT IF NECESSARY)
LC_ALL="C.UTF-8" # <- (CHANGE IT IF NECESSARY)
LC_LANG="C.UTF-8" # <- (CHANGE IT IF NECESSARY)
```

For Microk8s:

```
microk8s helm3 repo update
microk8s helm3 registry login -u admin registry.bioturing.com
microk8s helm3 upgrade --install --set secret.data.bbtoken="${BBTOKEN}" \
 --set secret.data.domain="${SVHOST}" \
 --set secret.server.certificate="${SSLCRT}" \
 --set secret.server.key="${SSLKEY}" \
 --set secret.server.useletsencrypt="${USELETSENCRYPT}" \
 --set secret.server.lcall="${LC_ALL}" \
 --set secret.server.lclang="${LC_LANG}" \
 --set secret.admin.username="${ADMIN_USERNAME}" \
```

```
 --set secret.admin.password="${ADMIN_PASSWORD}" \
 --set persistence.dirs.app.size="${APP_DATA_SIZE}" \
 --set persistence.dirs.user.size="${USER_DATA_SIZE}" \
 --set persistence.dirs.shm.size="${SHM_SIZE}" \
bioturing bioturing/colab --version ${CHART_VERSION}
```

For Vanilla k8s:

```
helm repo update
helm registry login -u admin registry.bioturing.com
helm upgrade --install --set secret.data.bbtoken="${BBTOKEN}" \
 --set secret.data.domain="${SVHOST}" \
 --set secret.server.certificate="${SSLCRT}" \
 --set secret.server.key="${SSLKEY}" \
 --set secret.server.useletsencrypt="${USELETSENCRYPT}" \
 --set secret.server.lcall="${LC_ALL}" \
 --set secret.server.lclang="${LC_LANG}" \
 --set secret.admin.username="${ADMIN_USERNAME}" \
 --set secret.admin.password="${ADMIN_PASSWORD}" \
 --set persistence.dirs.app.size="${APP_DATA_SIZE}" \
 --set persistence.dirs.user.size="${USER_DATA_SIZE}" \
 --set persistence.dirs.shm.size="${SHM_SIZE}" \
bioturing bioturing/colab --version ${CHART_VERSION}
```

SSO setup: There are various service providers, who work as IDP. Here we use Jumpcloud and okta.

– please select yourself.

https://colab.bioturing.com/dashboard/sso

SSO configuration: ***kindly contact support@bioturing.com for all product related questions, issues, including training.***

Configuration

All set with SSO

Troubleshooting

[Provide tips for troubleshooting common issues that may arise during or after installation.]

- Issue:

  permission denied while trying to connect to the Docker daemon socket at unix:///var/run/docker.sock: Get "http://%2Fvar%2Frun%2Fdocker.sock/v1.24/version": dial unix /var/run/docker.sock: connect: permission denied

- Solution:

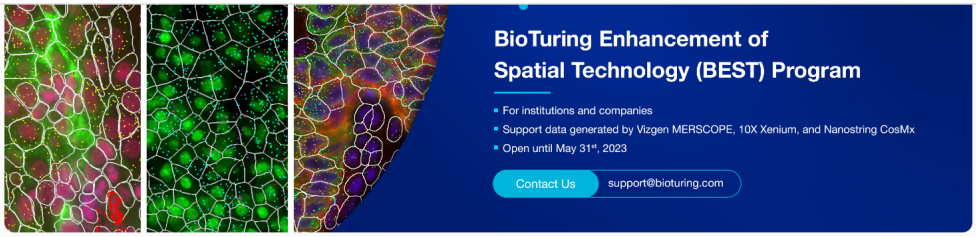
  docker ps -a
  groups
  sudo groupadd docker
  sudo groups
  sudo usermod -aG docker $USER
  sudo systemctl start docker
  newgrp docker
  docker ps -a

  ubuntu@ip-192-168-0-199:~$ ls -lhart /var/run/docker.sock
  srw-rw---- 1 root docker 0 Mar 30 08:59 /var/run/docker.sock
  ubuntu@ip-192-168-0-199:~$ sudo chmod 666 /var/run/docker.sock

Conclusion

BioTuring

- All platforms
- BioTuring database
- Cell type prediction
- Tutorial videos
- Publication
- Document



browserX

A GPU-accelerated single-cell platform

EXPLORE

TALK2DATA

Analyze single-cell data in the context of hundreds of million cells

EXPLORE

BioTuring Lens

Analyze spatial omics data with advanced AI algorithms

EXPLORE

Bio Colab

Data science platform and API for Bioinformaticians

EXPLORE

VINCI

Create and publish elegant visualization

EXPLORE