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.

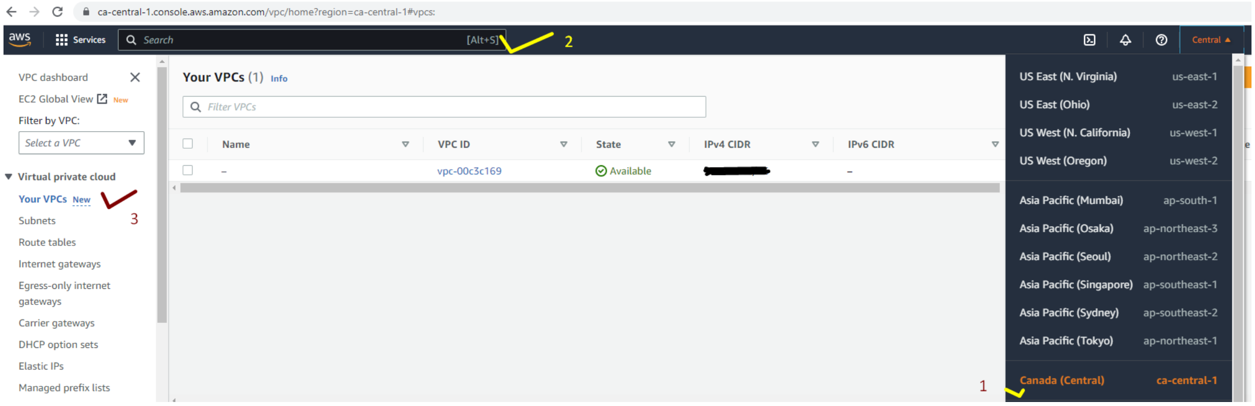
Download and Install

> **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.

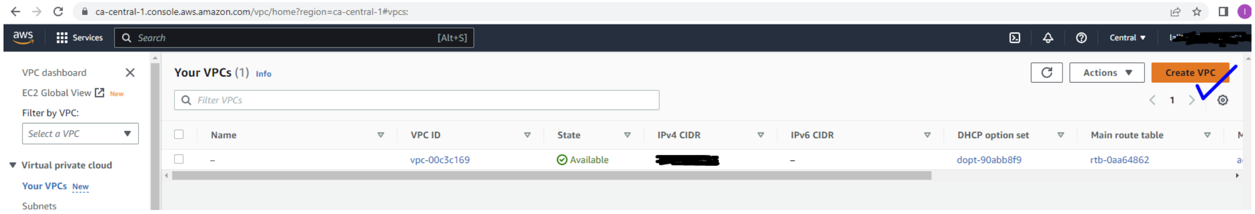
- Create a VPC
  - *https://docs.aws.amazon.com/vpc/latest/userguide/what-is-amazon-vpc.html*
  - *https://docs.aws.amazon.com/vpc/latest/userguide/create-vpc.html*

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


ca-central-1.console.aws.amazon.com/vpc/home?region=ca-central-1#vpcs:
aws
Services
Search
[Alt+S]
2
Central ▲
VPC dashboard
EC2 Global View New
Filter by VPC:
Select a VPC
▼ Virtual private cloud
Your VPCs New
3
Subnets
Route tables
Internet gateways
Egress-only internet gateways
Carrier gateways
DHCP option sets
Elastic IPs
Managed prefix lists
Your VPCs (1) Info
Filter VPCs
US East (N. Virginia) us-east-1
US East (Ohio) us-east-2
US West (N. California) us-west-1
US West (Oregon) us-west-2
Asia Pacific (Mumbai) ap-south-1
Asia Pacific (Osaka) ap-northeast-3
Asia Pacific (Seoul) ap-northeast-2
Asia Pacific (Singapore) ap-southeast-1
Asia Pacific (Sydney) ap-southeast-2
Asia Pacific (Tokyo) ap-northeast-1
1
Canada (Central) ca-central-1

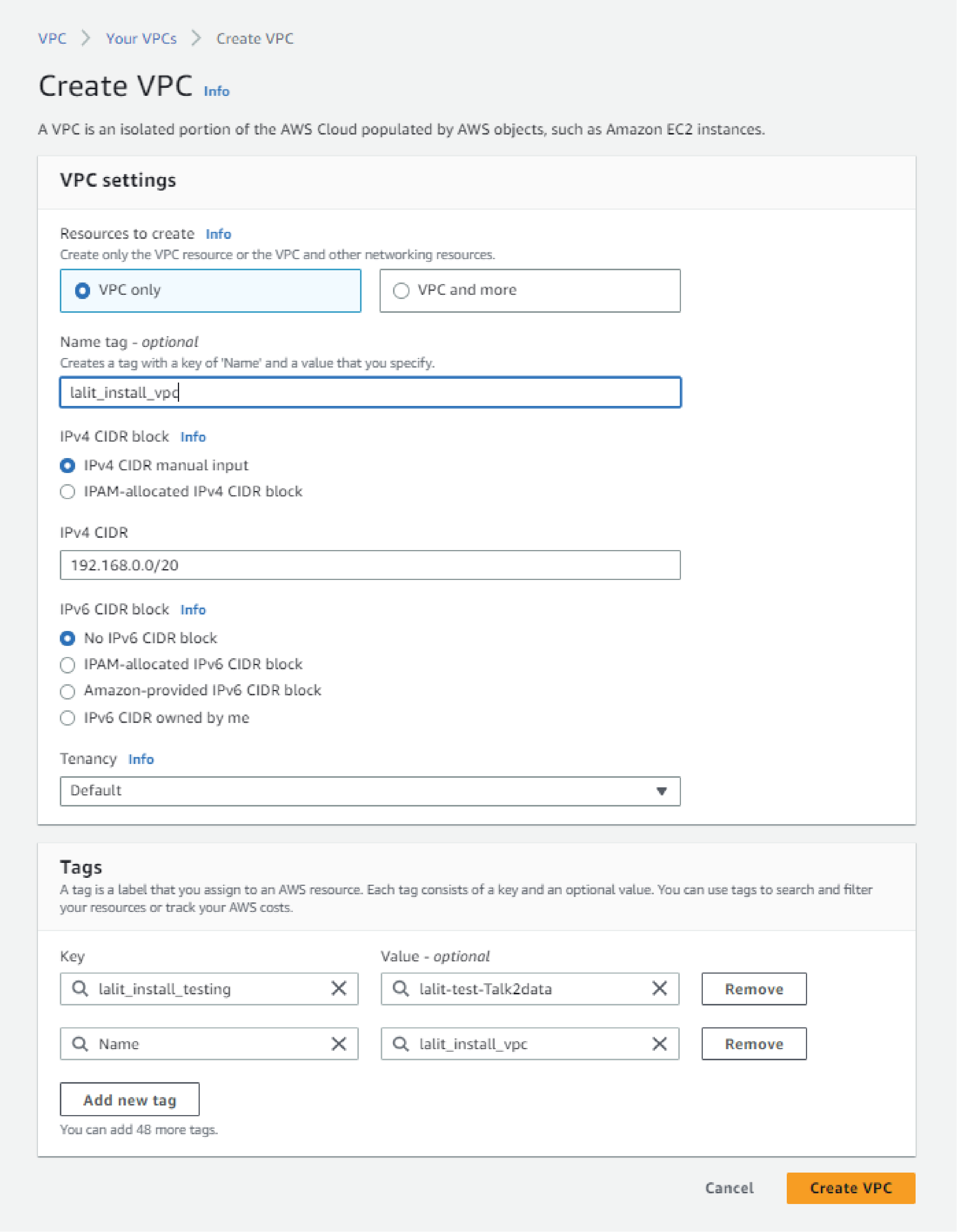
□	Name	VPC ID	State	IPv4 CIDR	IPv6 CIDR
□	–	vpc-00c3c169	✓ Available	[redacted]	–

4. Click on Create VPC



□	Name	VPC ID	State	IPv4 CIDR	IPv6 CIDR	DHCP option set	Main route table
□	–	vpc-00c3c169	✓ Available	[redacted]	–	dopt-90abb8f9	rtb-0aa64862

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

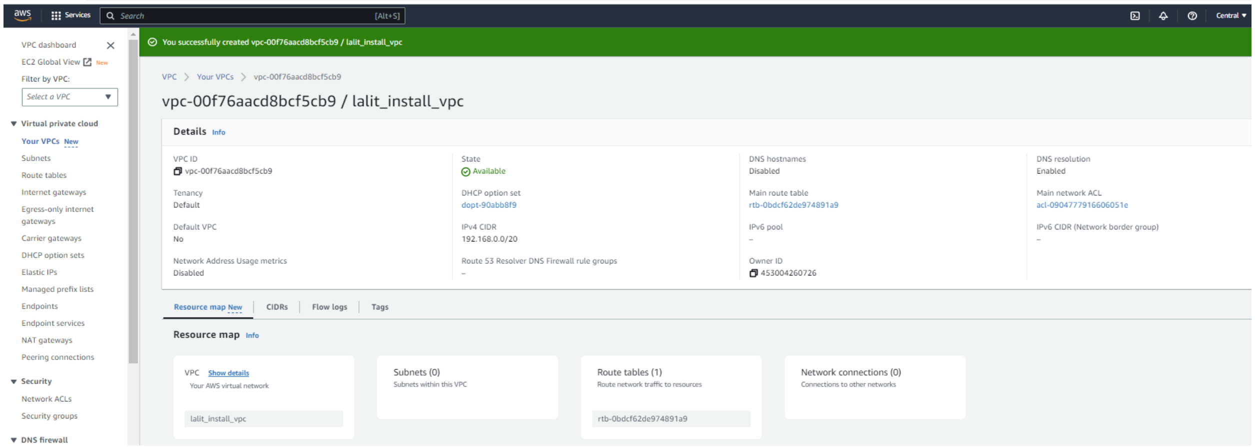
Tags

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.



You successfully created vpc-00f76aacd8bcf5cb9 / lalit_install_vpc

vpc-00f76aacd8bcf5cb9 / lalit_install_vpc

Details Info

VPC ID vpc-00f76aacd8bcf5cb9	State Available	DNS hostnames Disabled	DNS resolution Enabled
Tenancy Default	DHCP option set dopt-90abb8f9	Main route table rtb-0bdcf62de974891a9	Main network ACL acl-0904777916606051e
Default VPC No	IPv4 CIDR 192.168.0.0/20	IPv6 pool –	IPv6 CIDR (Network border group) –
Network Address Usage metrics Disabled	Route 53 Resolver DNS Firewall rule groups –	Owner ID 453004260726	

Resource map Info

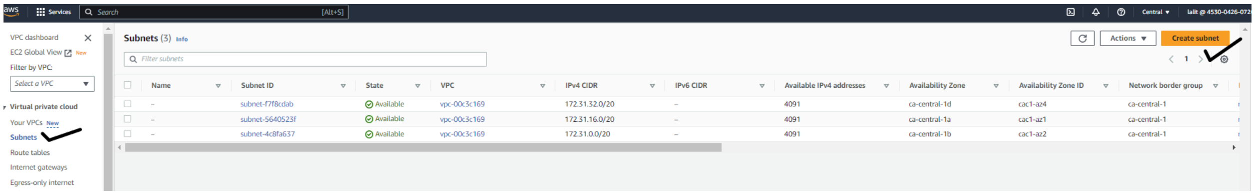
VPC Show details
Your AWS virtual network
lalit_install_vpc

Subnets (0)
Subnets within this VPC

Route tables (1)
Route network traffic to resources
rtb-0bdcf62de974891a9

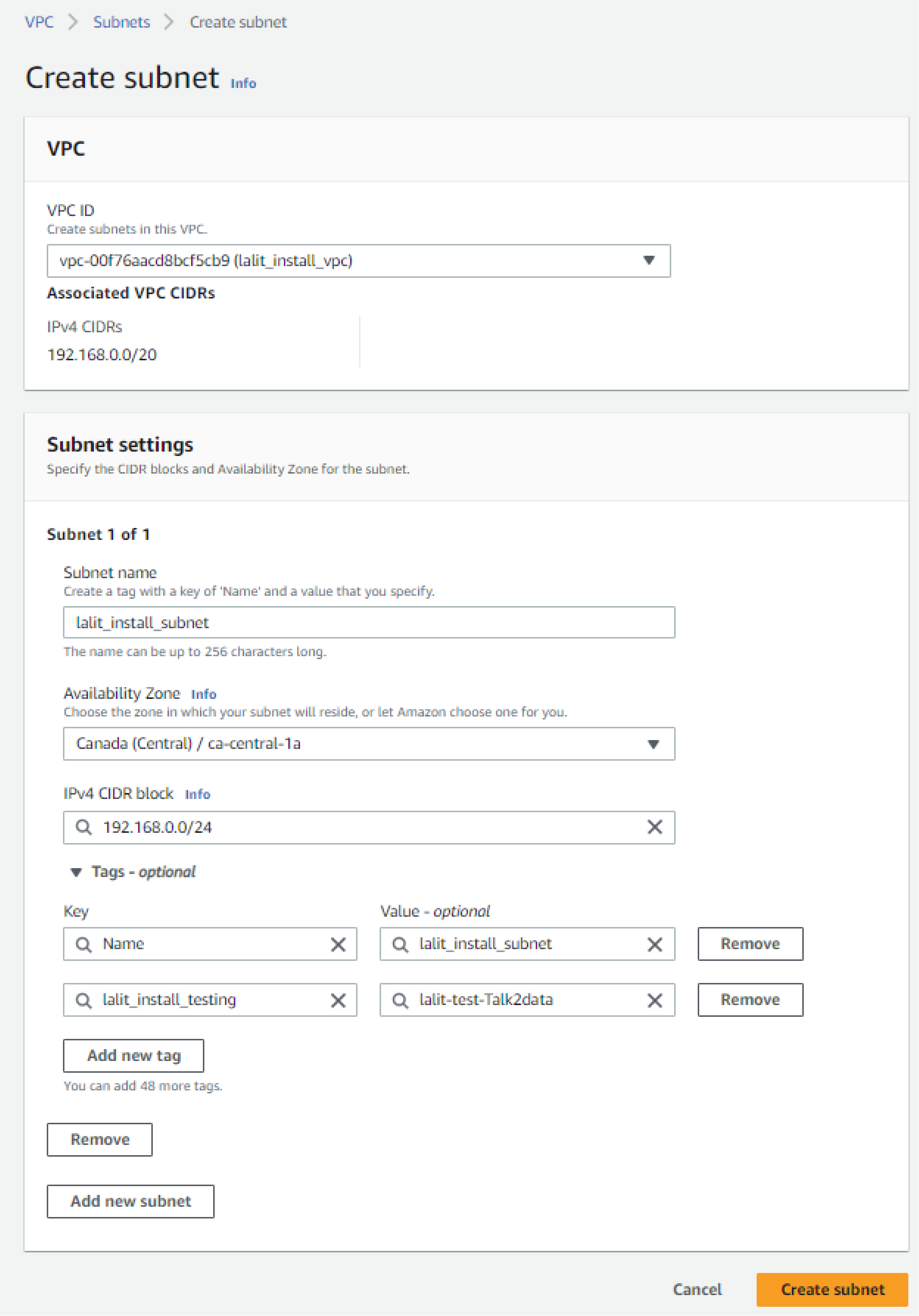
Network connections (0)
Connections to other networks

- Create Subnet



Subnets (3) Info

	Name	Subnet ID	State	VPC	IPv4 CIDR	IPv6 CIDR	Available IPv4 addresses	Availability Zone	Availability Zone ID	Network border group
□	–	subnet-f7f8cdab	Available	vpc-00c3c169	172.31.32.0/20	–	4091	ca-central-1d	cac1-az4	ca-central-1
□	–	subnet-5640523f	Available	vpc-00c3c169	172.31.16.0/20	–	4091	ca-central-1a	cac1-az1	ca-central-1
□	–	subnet-4c8fa637	Available	vpc-00c3c169	172.31.0.0/20	–	4091	ca-central-1b	cac1-az2	ca-central-1



Create subnet

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
Choose the zone in which your subnet will reside, or let Amazon choose one for you.
Canada (Central) / ca-central-1a

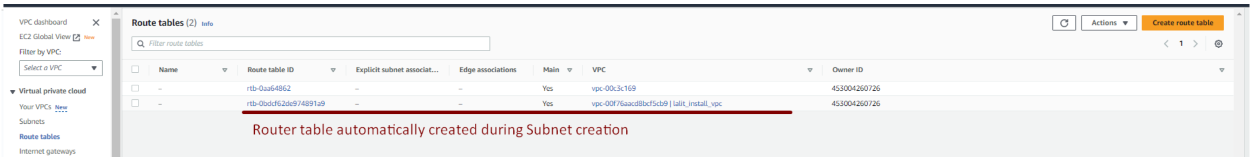
IPv4 CIDR block
192.168.0.0/24

▼ Tags - *optional*

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

You can add 48 more tags.

- Verify Router Table

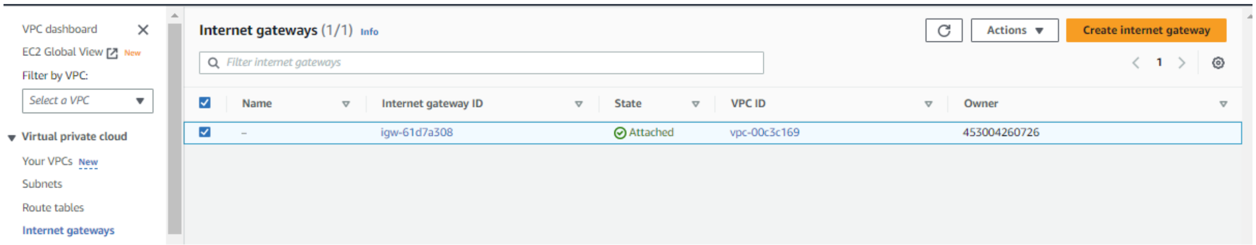


Route tables (2) Info

	Name	Route table ID	Explicit subnet associat...	Edge associations	Main	VPC	Owner ID
□	–	rtb-0aa64862	–	–	Yes	vpc-00c3c169	453004260726
□	–	rtb-0bdcf62de974891a9	–	–	Yes	vpc-00f76aacd8bcf5cb9 | lalit_install_vpc	453004260726

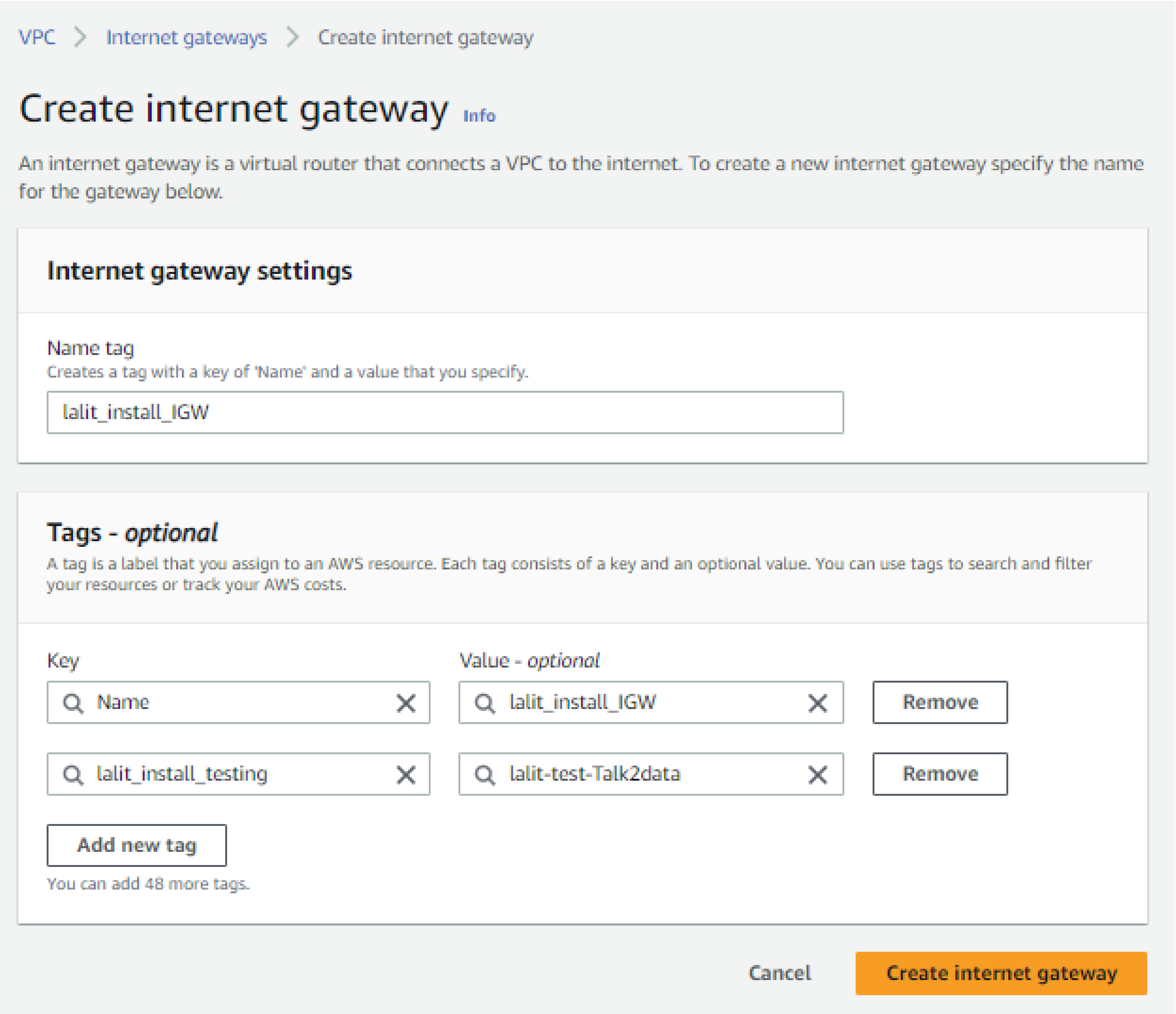
Router table automatically created during Subnet creation

- Create Internet Gateway



Internet gateways (1/1) Info

☑	Name	Internet gateway ID	State	VPC ID	Owner
☑	–	igw-61d7a308	✓ Attached	vpc-00c3c169	453004260726



VPC > Internet gateways > Create internet gateway

Create internet gateway Info

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

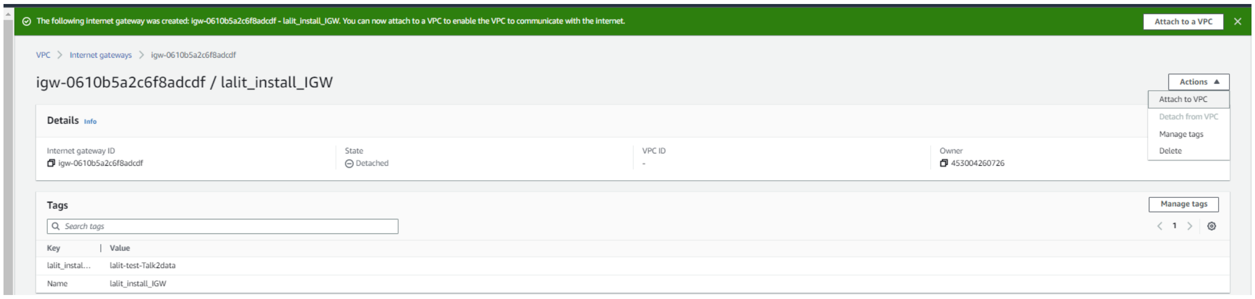
Internet gateway settings

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

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

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

You can add 48 more tags.



The following internet gateway was created: igw-0610b5a2c6f8adcdf - lalit_install_IGW. You can now attach to a VPC to enable the VPC to communicate with the internet.

igw-0610b5a2c6f8adcdf / lalit_install_IGW

Details

Internet gateway ID: igw-0610b5a2c6f8adcdf

State: Detached

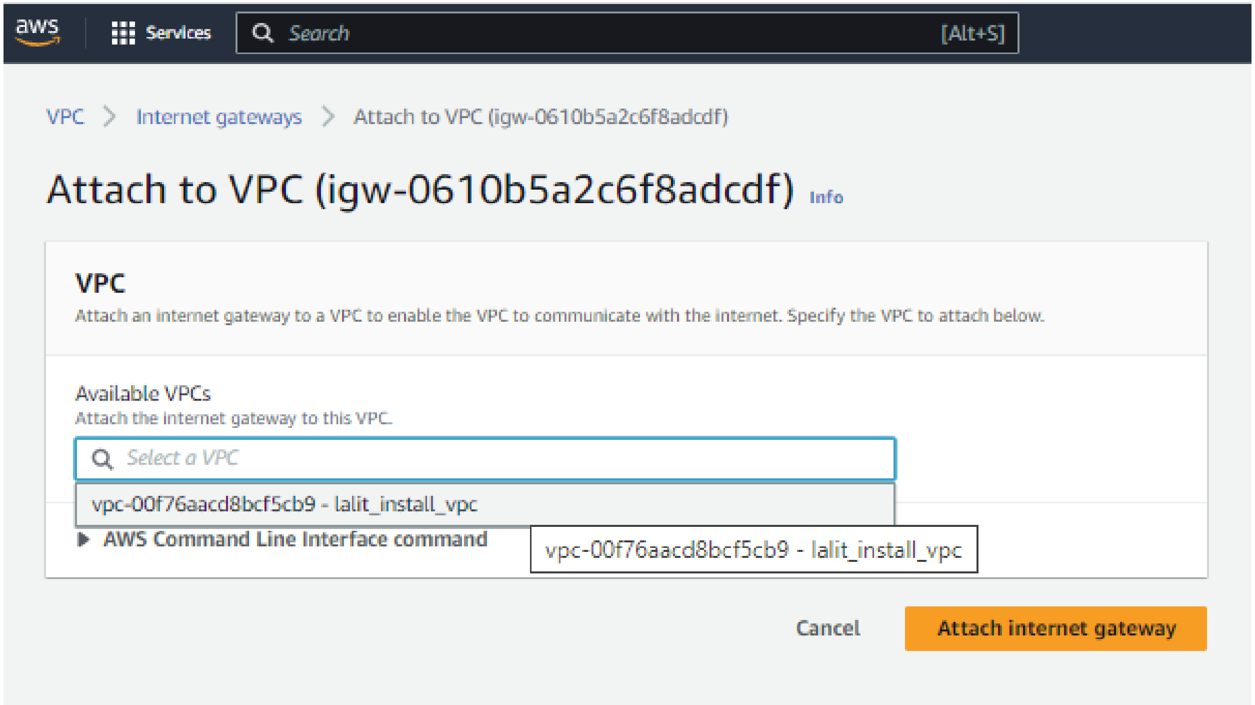
VPC ID: -

Owner: 453004260726

Tags

Key	Value
lalit_instal...	lalit-test-Talk2data
Name	lalit_install_IGW

- IGW – Attach to VPC


aws
Services
Search
[Alt+S]
VPC > Internet gateways > Attach to VPC (igw-0610b5a2c6f8adcdf)
Info
Select a VPC
AWS Command Line Interface command
Cancel
Attach internet gateway

Attach to VPC (igw-0610b5a2c6f8adcdf)

VPC

Attach an internet gateway to a VPC to enable the VPC to communicate with the internet. Specify the VPC to attach below.

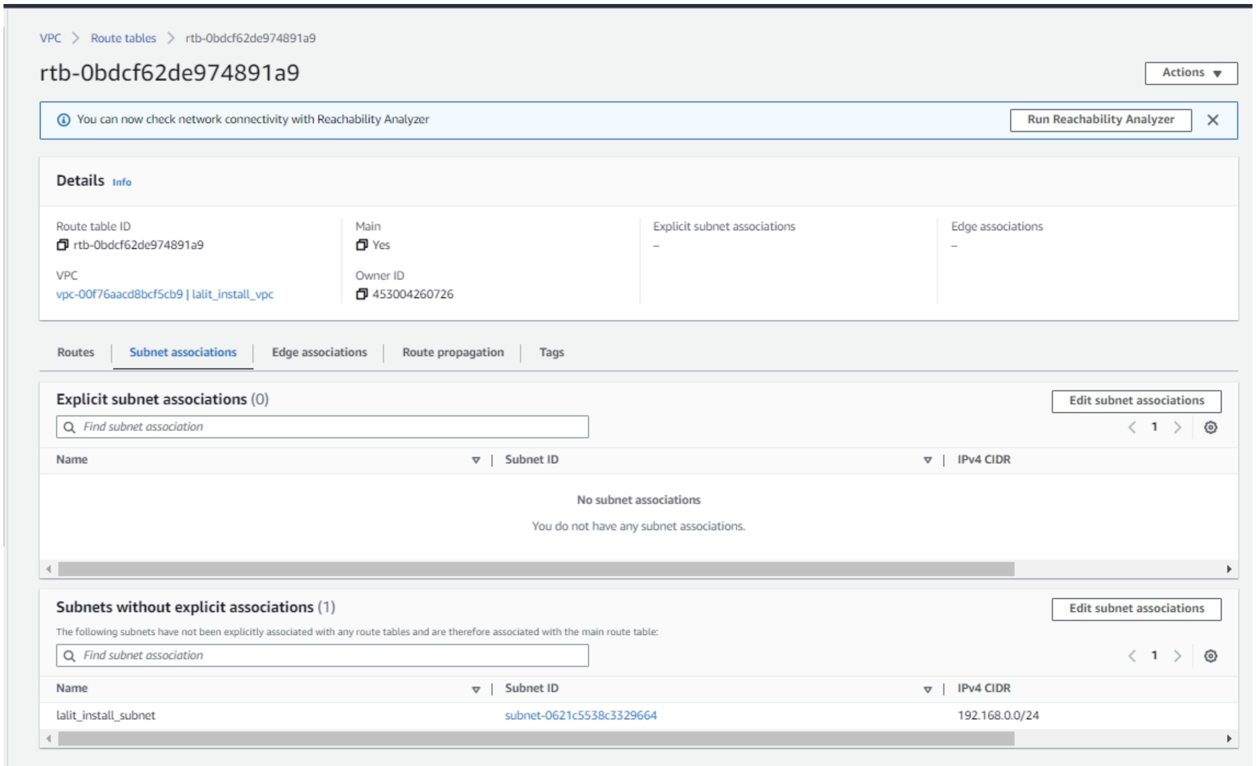
Available VPCs

Attach the internet gateway to this VPC.

vpc-00f76aacd8bcf5cb9 - lalit_install_vpc

vpc-00f76aacd8bcf5cb9 - lalit_install_vpc

- Update Router



VPC > Route tables > rtb-0bdcf62de974891a9

rtb-0bdcf62de974891a9

ⓘ You can now check network connectivity with Reachability Analyzer

Run Reachability Analyzer
✕

Details Info

Route table ID	Main	Explicit subnet associations	Edge associations
rtb-0bdcf62de974891a9	Yes	–	–
VPC	Owner ID		
vpc-00f76aacd8bcf5cb9 | lalit_install_vpc	453004260726		

Routes | Subnet associations | Edge associations | Route propagation | Tags

Explicit subnet associations (0)

Edit subnet associations
Find subnet association
< 1 >

Name	Subnet ID	IPv4 CIDR

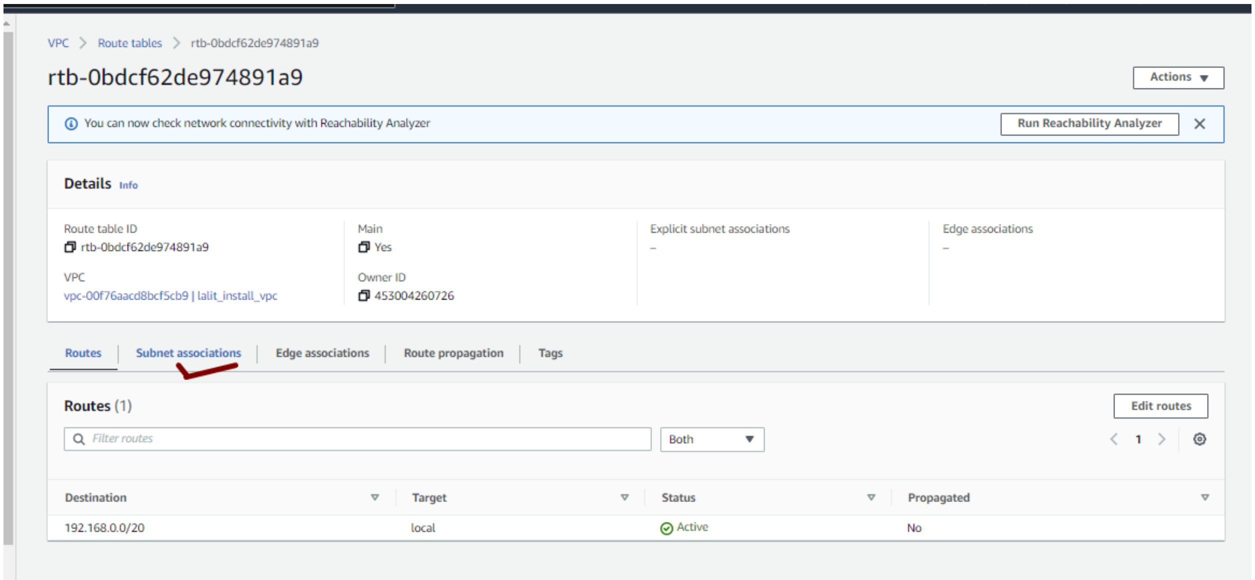
No subnet associations

You do not have any subnet associations.

Subnets without explicit associations (1)

The following subnets have not been explicitly associated with any route tables and are therefore associated with the main route table:

Name	Subnet ID	IPv4 CIDR
lalit_install_subnet	subnet-0621c5538c3329664	192.168.0.0/24



VPC > Route tables > rtb-0bdcf62de974891a9

rtb-0bdcf62de974891a9

ⓘ You can now check network connectivity with Reachability Analyzer

Run Reachability Analyzer
✕

Details Info

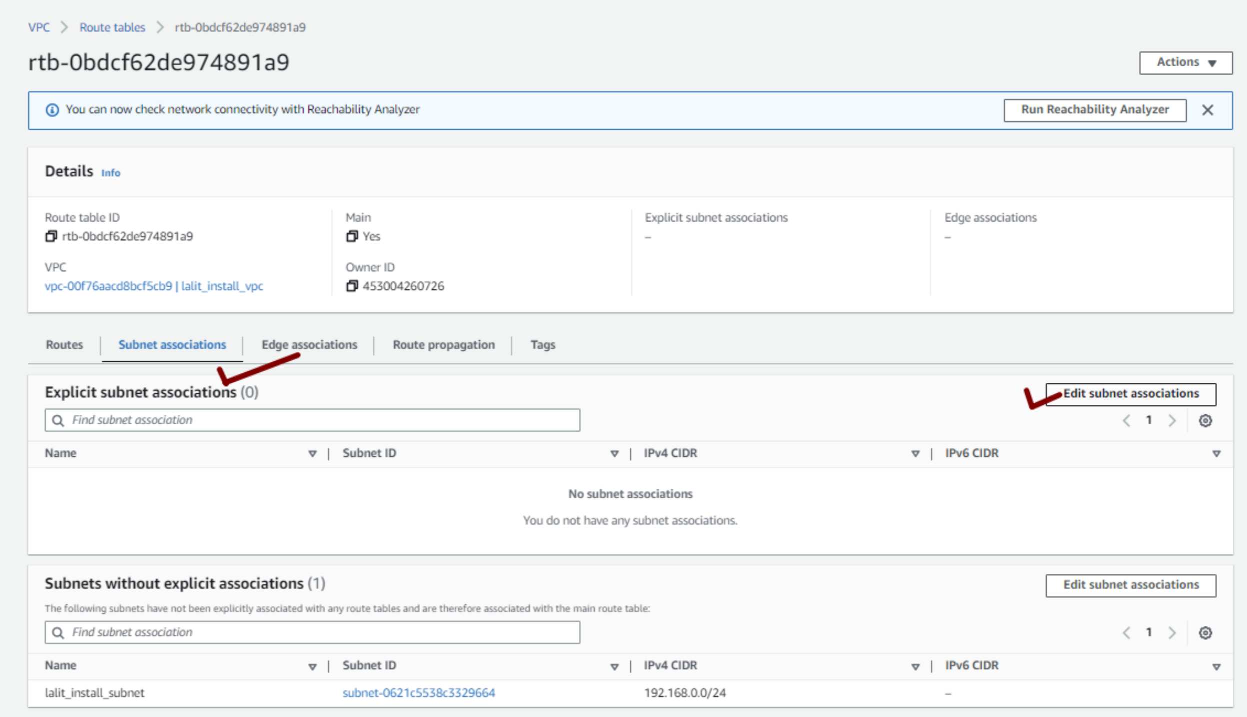
Route table ID	Main	Explicit subnet associations	Edge associations
rtb-0bdcf62de974891a9	Yes	–	–
VPC	Owner ID		
vpc-00f76aacd8bcf5cb9 | lalit_install_vpc	453004260726		

Routes | Subnet associations | Edge associations | Route propagation | Tags

Routes (1)

Edit routes
Filter routes
Both ▼
< 1 >

Destination	Target	Status	Propagated
192.168.0.0/20	local	✓ Active	No



VPC > Route tables > rtb-0bdcf62de974891a9

rtb-0bdcf62de974891a9

You can now check network connectivity with 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		

Explicit subnet associations (0)

Name	Subnet ID	IPv4 CIDR	IPv6 CIDR

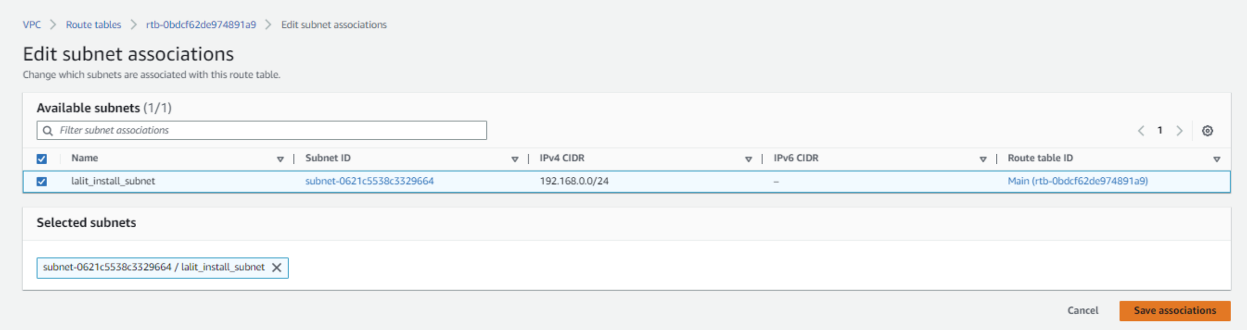
No subnet associations

You do not have any subnet associations.

Subnets without explicit associations (1)

The following subnets have not been explicitly associated with any route tables and are therefore associated with the main route table:

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

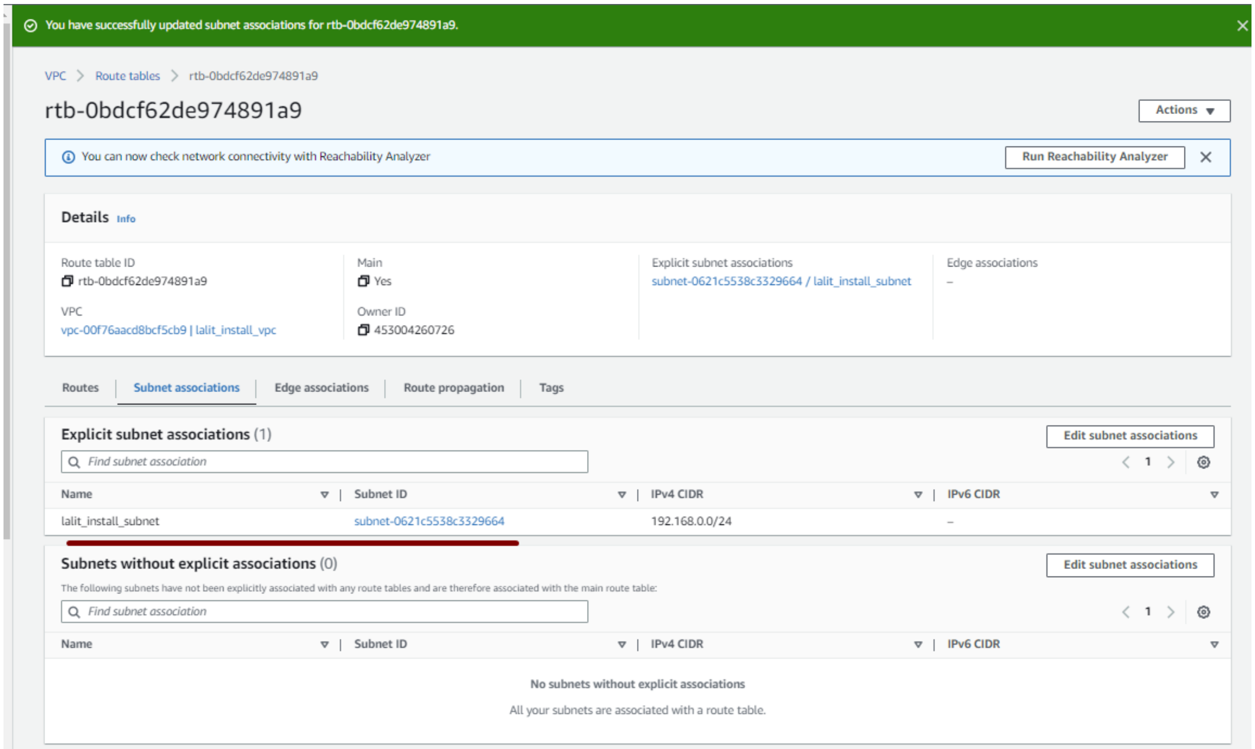
Change which subnets are associated with this route table.

Available subnets (1/1)

☑	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



You have successfully updated subnet associations for rtb-0bdcf62de974891a9.

VPC > Route tables > rtb-0bdcf62de974891a9

rtb-0bdcf62de974891a9

You can now check network connectivity with Reachability Analyzer

Details Info

Route table ID	Main	Explicit subnet associations	Edge associations
rtb-0bdcf62de974891a9	Yes	subnet-0621c5538c3329664 / lalit_install_subnet	–
VPC	Owner ID		
vpc-00f76aacd8bcf5cb9 | lalit_install_vpc	453004260726		

Explicit subnet associations (1)

Name	Subnet ID	IPv4 CIDR	IPv6 CIDR
lalit_install_subnet	subnet-0621c5538c3329664	192.168.0.0/24	–

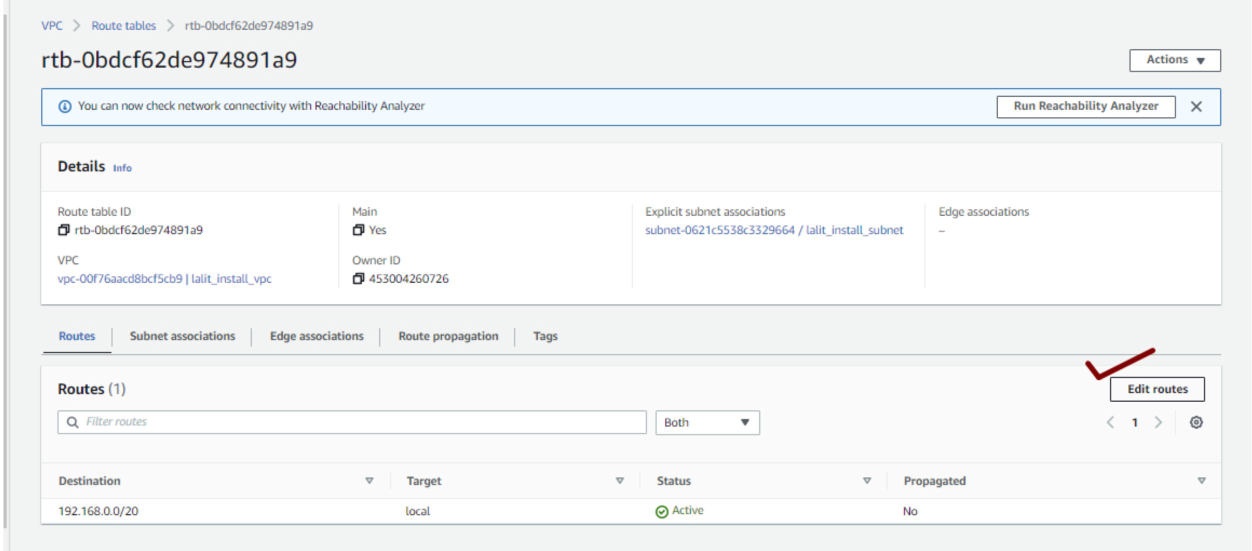
Subnets without explicit associations (0)

The following subnets have not been explicitly associated with any route tables and are therefore associated with the main route table:

Name	Subnet ID	IPv4 CIDR	IPv6 CIDR

No subnets without explicit associations

All your subnets are associated with a route table.



VPC > Route tables > rtb-0bdcf62de974891a9

rtb-0bdcf62de974891a9

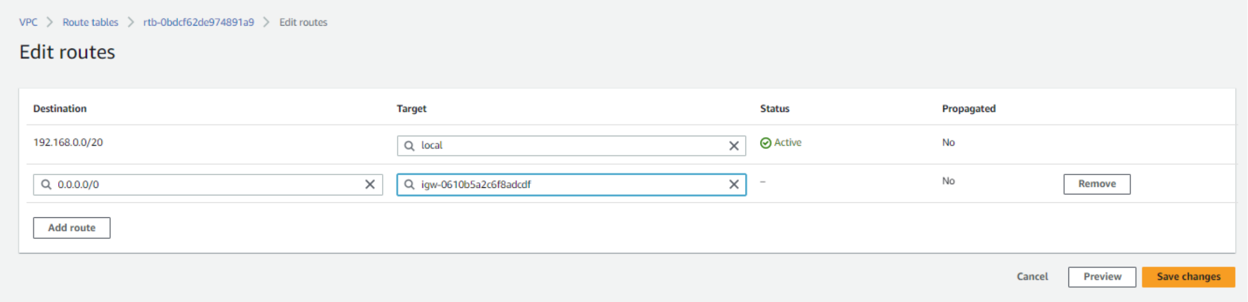
You can now check network connectivity with Reachability Analyzer

Details Info

Route table ID	Main	Explicit subnet associations	Edge associations
rtb-0bdcf62de974891a9	Yes	subnet-0621c5538c3329664 / lalit_install_subnet	–
VPC	Owner ID		
vpc-00f76aacd8bcf5cb9 | lalit_install_vpc	453004260726		

Routes (1)

Destination	Target	Status	Propagated
192.168.0.0/20	local	Active	No

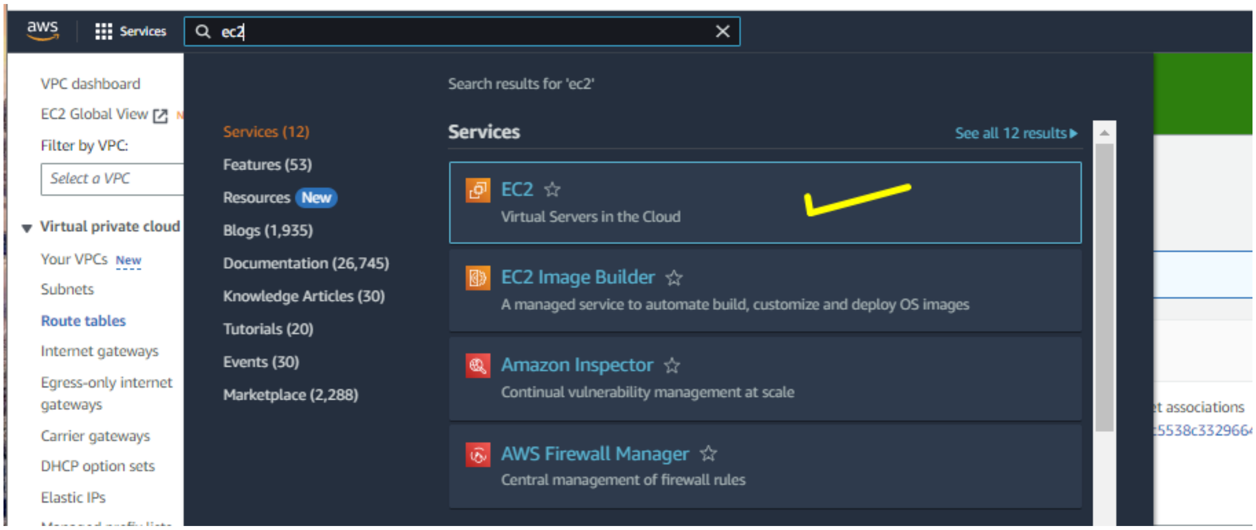


VPC > Route tables > rtb-0bdcf62de974891a9 > Edit routes

Edit routes

Destination	Target	Status	Propagated	
192.168.0.0/20	local	Active	No	
0.0.0.0/0	igw-0610b5a2c6f8adcdf	-	No	Remove

- Create instance based on your requirement



Search results for 'ec2'

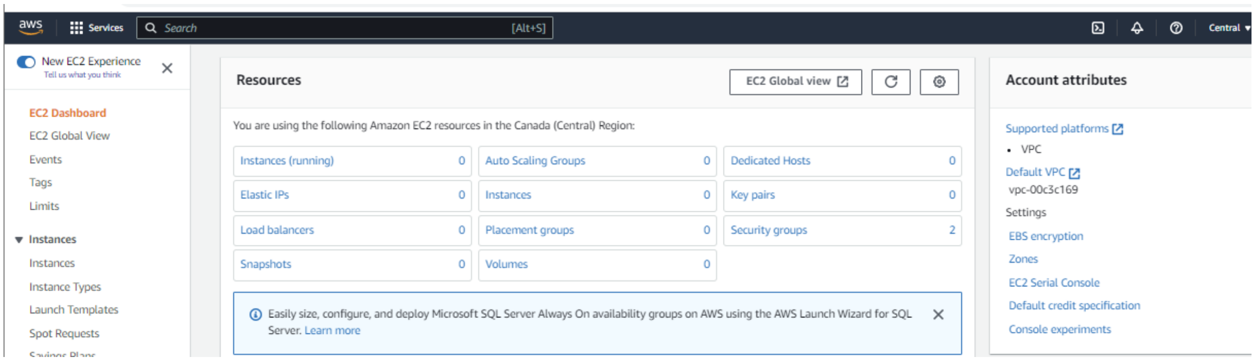
Services

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

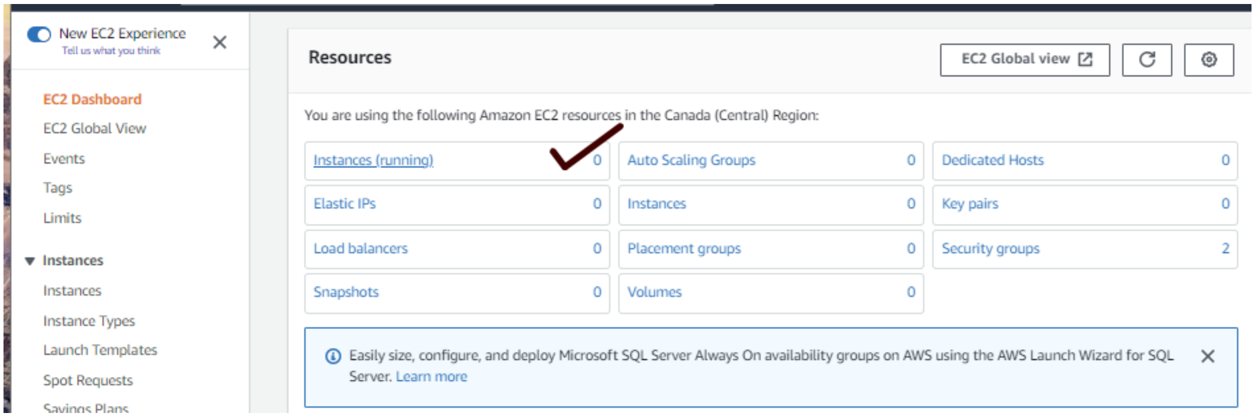


Resources

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

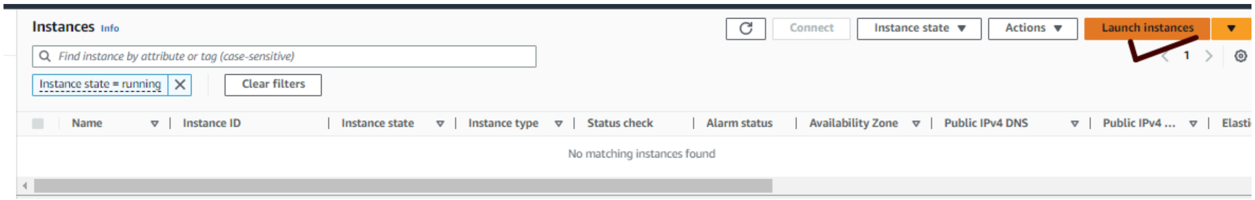


Resources

You are using the following Amazon EC2 resources in the Canada (Central) Region:

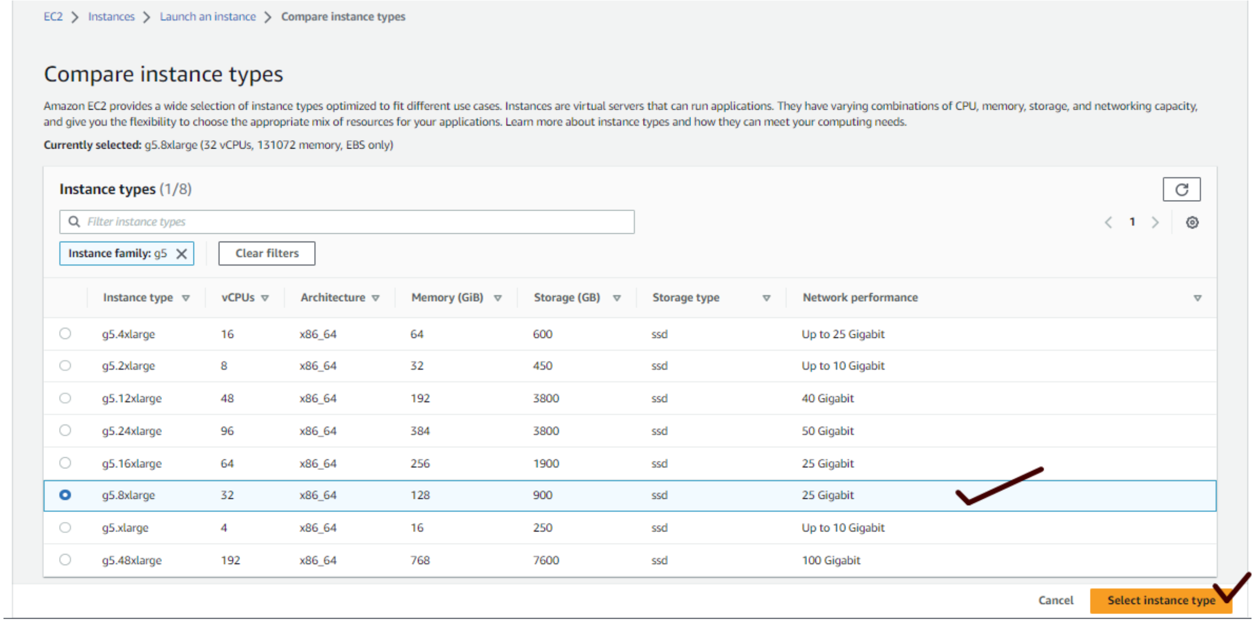
Instances (running)	0	Auto Scaling Groups	0	Dedicated Hosts	0
Elastic IPs	0	Instances	0	Key pairs	0
Load balancers	0	Placement groups	0	Security groups	2
Snapshots	0	Volumes	0		

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



Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone	Public IPv4 DNS	Public IPv4 ...	Elasti

No matching instances found



EC2 > Instances > Launch an instance > Compare instance types

Compare instance types

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

Currently selected: g5.8xlarge (32 vCPUs, 131072 memory, EBS only)

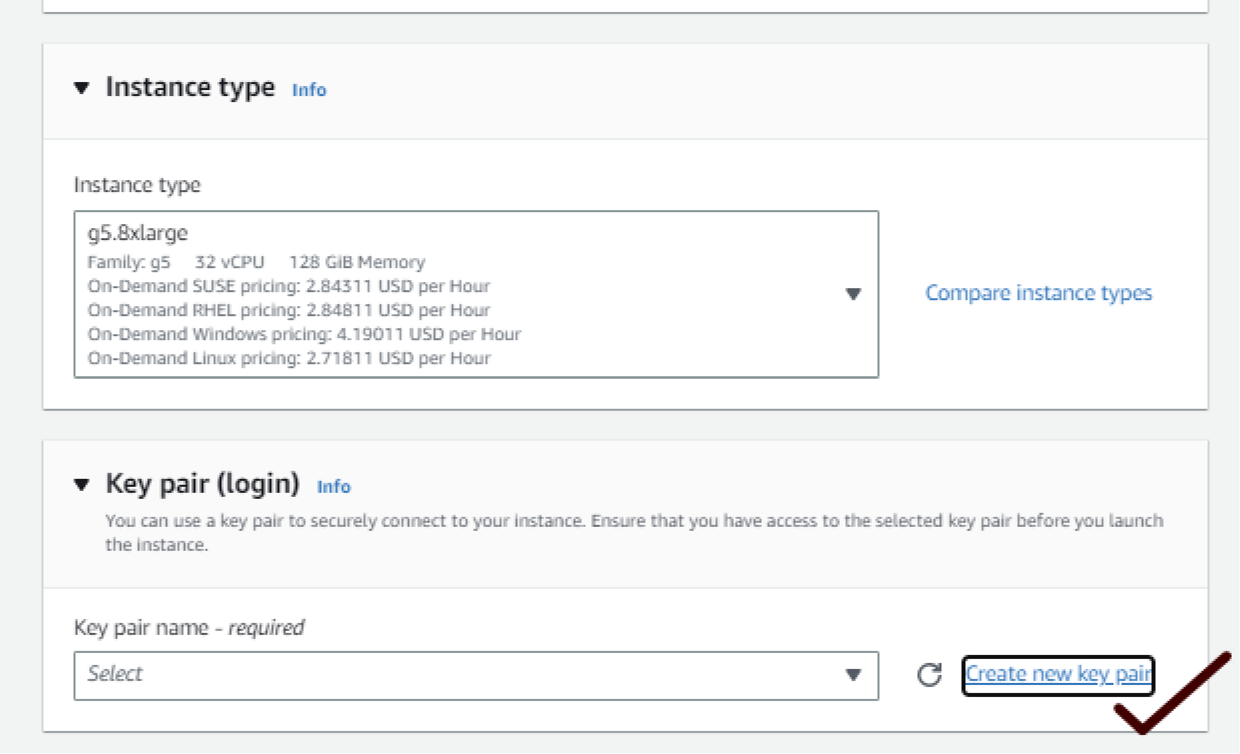
Instance types (1/8)

Filter instance types

Instance family: g5 ✕ | Clear filters

	Instance type	vCPUs	Architecture	Memory (GiB)	Storage (GB)	Storage type	Network performance
○	g5.4xlarge	16	x86_64	64	600	ssd	Up to 25 Gigabit
○	g5.2xlarge	8	x86_64	32	450	ssd	Up to 10 Gigabit
○	g5.12xlarge	48	x86_64	192	3800	ssd	40 Gigabit
○	g5.24xlarge	96	x86_64	384	3800	ssd	50 Gigabit
○	g5.16xlarge	64	x86_64	256	1900	ssd	25 Gigabit
●	g5.8xlarge	32	x86_64	128	900	ssd	25 Gigabit
○	g5.xlarge	4	x86_64	16	250	ssd	Up to 10 Gigabit
○	g5.48xlarge	192	x86_64	768	7600	ssd	100 Gigabit

- Create a new key for SSH login to server



Instance type Info

Instance type

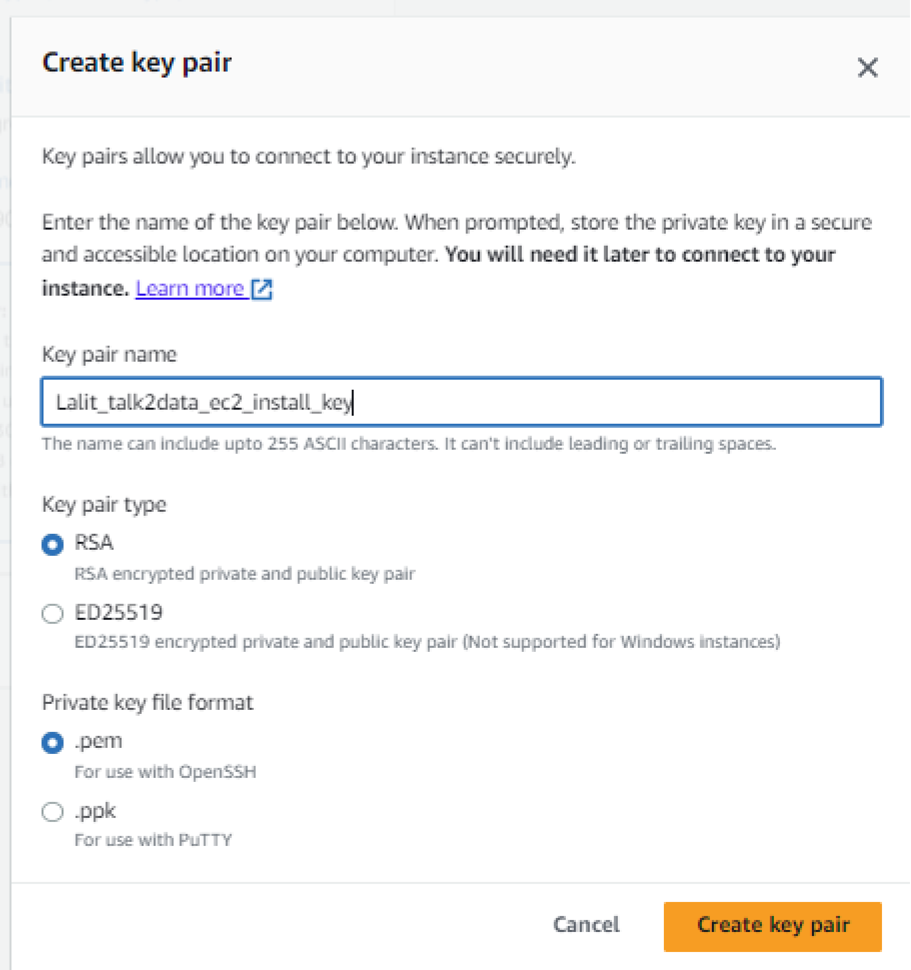
g5.8xlarge
Family: g5 32 vCPU 128 GiB Memory
On-Demand SUSE pricing: 2.84311 USD per Hour
On-Demand RHEL pricing: 2.84811 USD per Hour
On-Demand Windows pricing: 4.19011 USD per Hour
On-Demand Linux pricing: 2.71811 USD per Hour

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



Key pairs allow you to connect to your instance securely.

Enter the name of the key pair below. When prompted, store the private key in a secure and accessible location on your computer. **You will need it later to connect to your instance.** Learn more

Key pair name

Lalit_talk2data_ec2_install_key

The name can include upto 255 ASCII characters. It can't include leading or trailing spaces.

Key pair type

● RSA
RSA encrypted private and public key pair

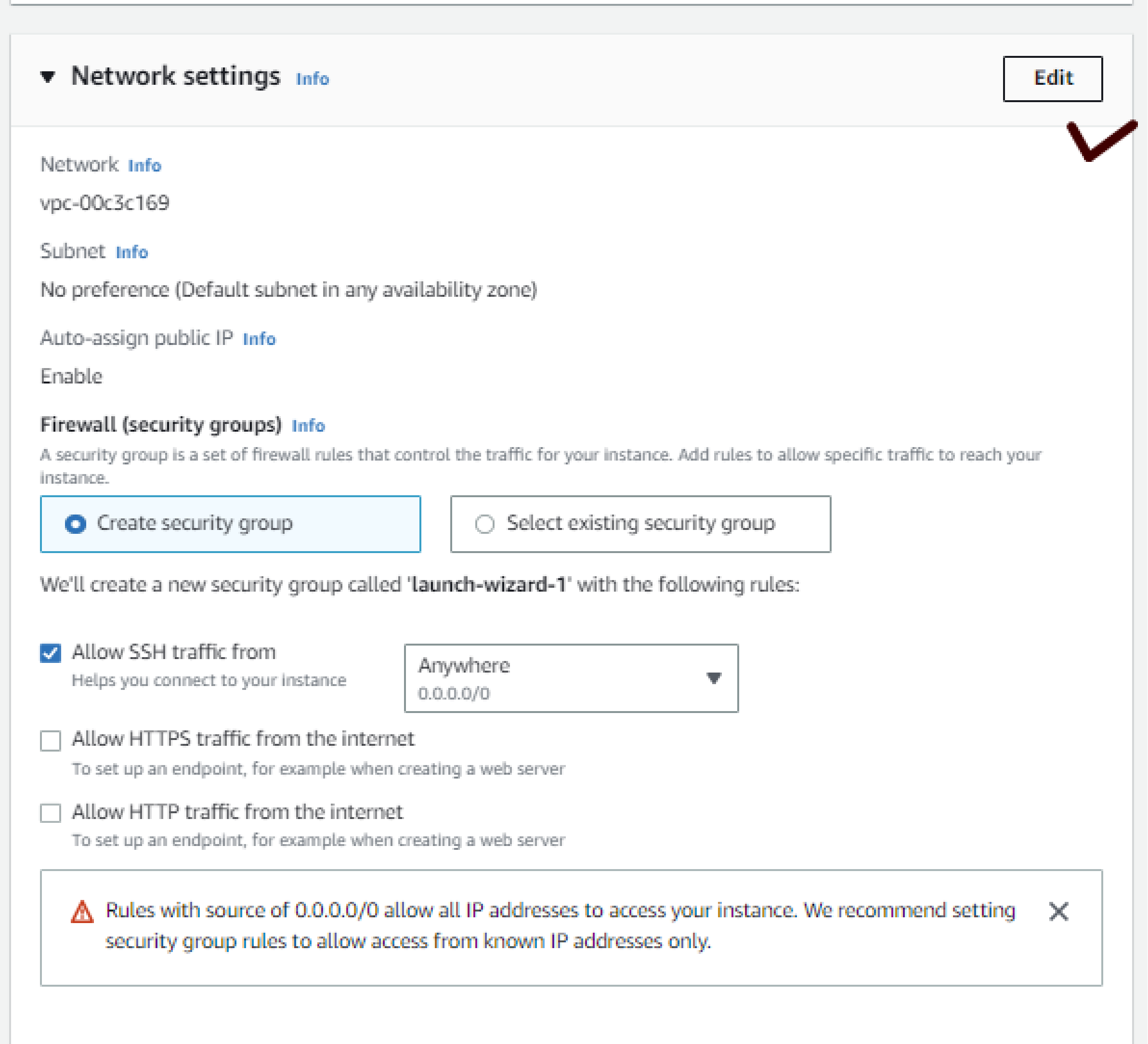
○ ED25519
ED25519 encrypted private and public key pair (Not supported for Windows instances)

Private key file format

● .pem
For use with OpenSSH

○ .ppk
For use with PuTTY

- Select the VPC, which we created earlier.



▼ Network settings Info

Network Info

vpc-00c3c169

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-1**' with the following rules:

☑ Allow SSH traffic from — Anywhere 0.0.0.0/0

Helps you connect to your instance

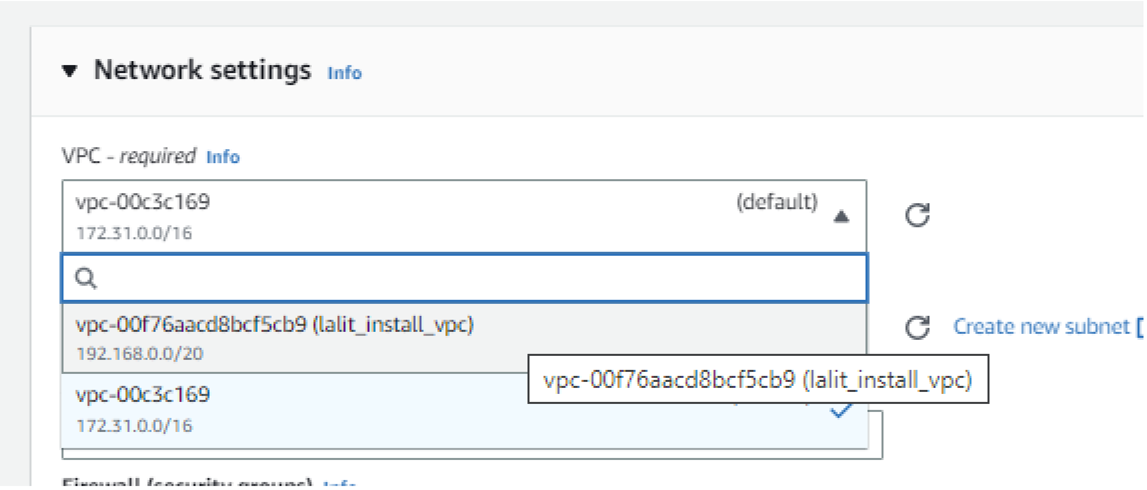
□ Allow HTTPS traffic from the internet

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.



▼ Network settings Info

VPC - *required* Info

vpc-00c3c169 (default)
172.31.0.0/16

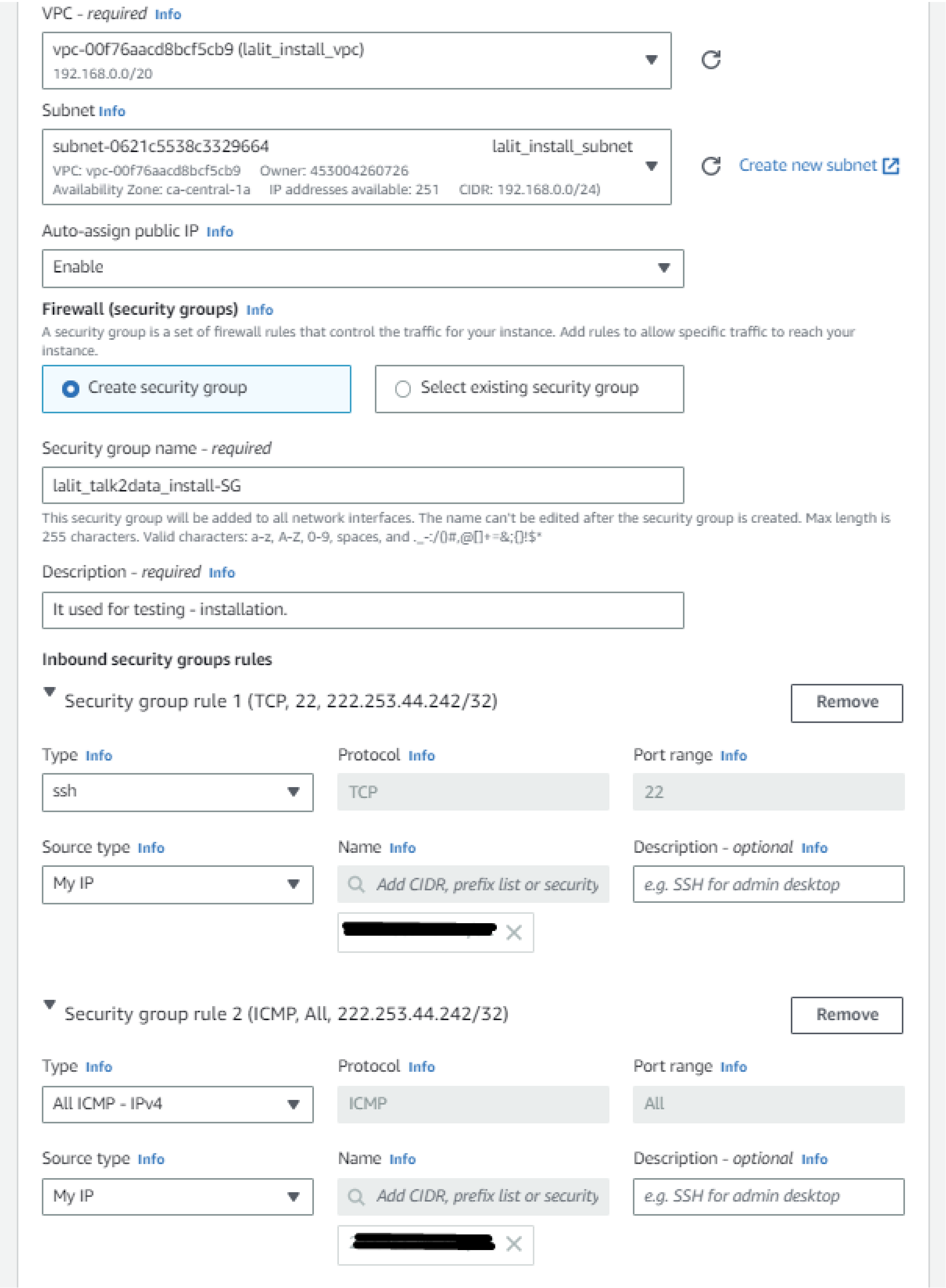
vpc-00f76aacd8bcf5cb9 (lalit_install_vpc)
192.168.0.0/20

vpc-00c3c169 ✓
172.31.0.0/16

vpc-00f76aacd8bcf5cb9 (lalit_install_vpc)

Firewall (security groups) Info

- Create Security Group based on requirement



VPC - *required* Info

vpc-00f76aacd8bcf5cb9 (lalit_install_vpc)
192.168.0.0/20

Subnet Info

subnet-0621c5538c3329664 lalit_install_subnet
VPC: vpc-00f76aacd8bcf5cb9 Owner: 453004260726
Availability Zone: ca-central-1a IP addresses available: 251 CIDR: 192.168.0.0/24)

Create new subnet

Auto-assign public IP Info

Enable

Firewall (security groups) Info

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

● Create security group ○ Select existing security group

Security group name - *required*

lalit_talk2data_install-SG

This security group will be added to all network interfaces. The name can't be edited after the security group is created. Max length is 255 characters. Valid characters: a-z, A-Z, 0-9, spaces, and ._-:/()#,@[]+=&;{}!$*

Description - *required* Info

It used for testing - installation.

Inbound security groups rules

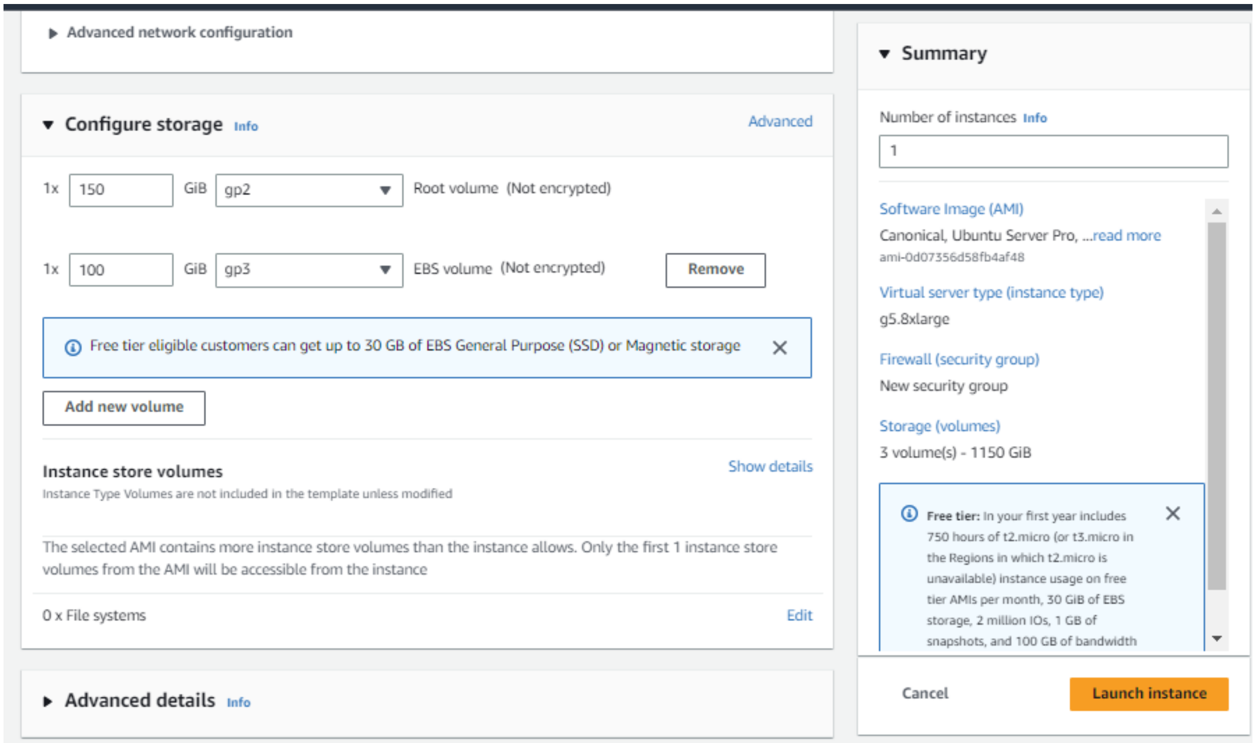
Security group rule 1 (TCP, 22, 222.253.44.242/32) Remove

Type Info: ssh
Protocol Info: TCP
Port range Info: 22
Source type Info: My IP
Name Info: *Add CIDR, prefix list or security*
[redacted]
Description - *optional* Info: *e.g. SSH for admin desktop*

Security group rule 2 (ICMP, All, 222.253.44.242/32) Remove

Type Info: All ICMP - IPv4
Protocol Info: ICMP
Port range Info: All
Source type Info: My IP
Name Info: *Add CIDR, prefix list or security*
[redacted]
Description - *optional* Info: *e.g. SSH for admin desktop*

- Attach Disk for data volume



▸ Advanced network configuration

▾ Configure storage Info — Advanced

1x 150 GiB gp2 ▾ Root volume (Not encrypted)

1x 100 GiB gp3 ▾ EBS volume (Not encrypted) — Remove

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

Add new volume

Instance store volumes — Show details

Instance Type Volumes are not included in the template unless modified

The selected AMI contains more instance store volumes than the instance allows. Only the first 1 instance store volumes from the AMI will be accessible from the instance

0 x File systems — Edit

▸ Advanced details Info

▾ Summary

Number of instances Info: 1

Software Image (AMI): Canonical, Ubuntu Server Pro, ...read more
ami-0d07356d58fb4af48

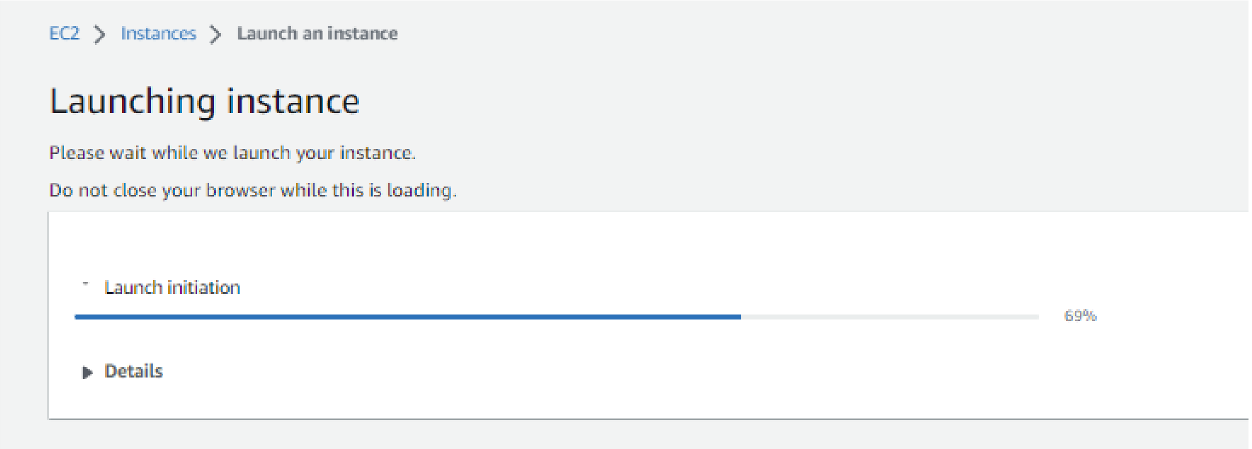
Virtual server type (instance type): g5.8xlarge

Firewall (security group): New security group

Storage (volumes): 3 volume(s) - 1150 GiB

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

Cancel — Launch instance



EC2 > Instances > Launch an instance

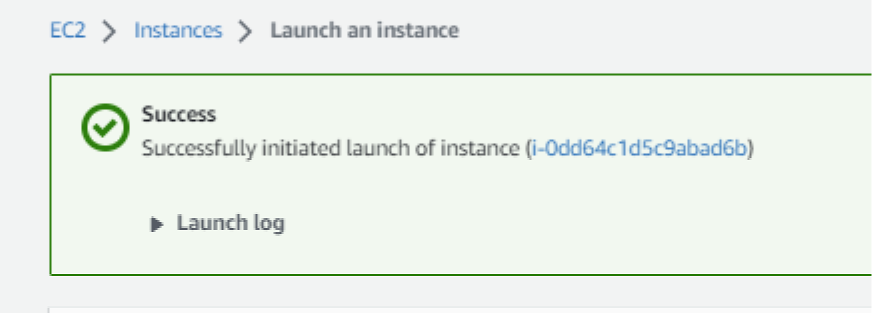
Launching instance

Please wait while we launch your instance.

Do not close your browser while this is loading.

Launch initiation 69%

▸ Details

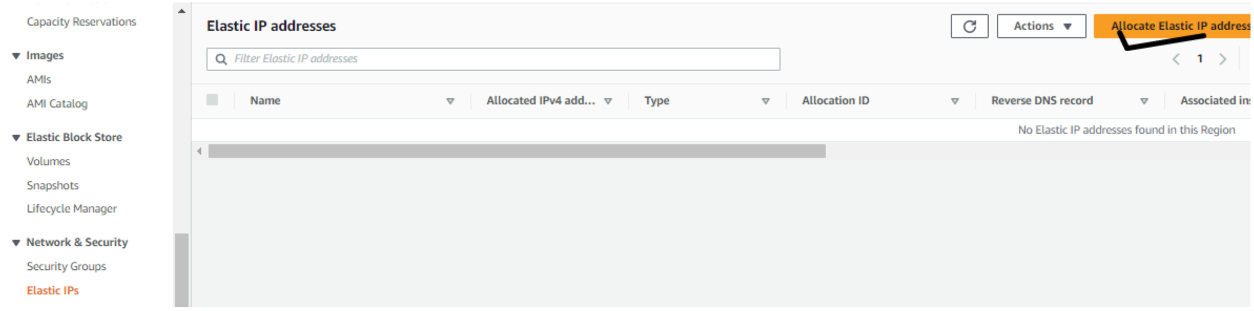


EC2 > Instances > Launch an instance

Success
Successfully initiated launch of instance (i-0dd64c1d5c9abad6b)

▸ Launch log

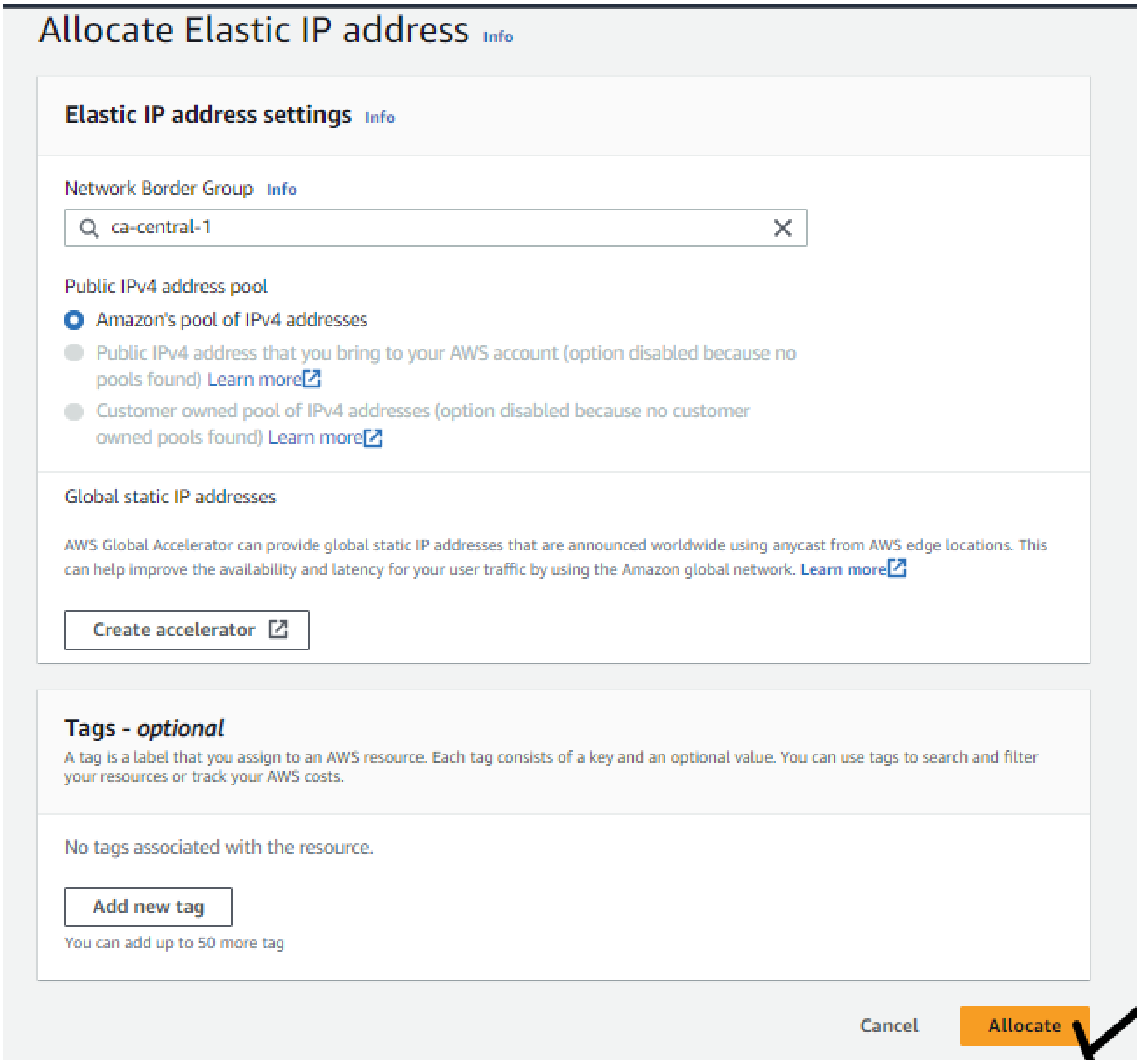
- Assign Elastic IP to this instance



Elastic IP addresses

□	Name	Allocated IPv4 add...	Type	Allocation ID	Reverse DNS record	Associated in

No Elastic IP addresses found in this Region



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

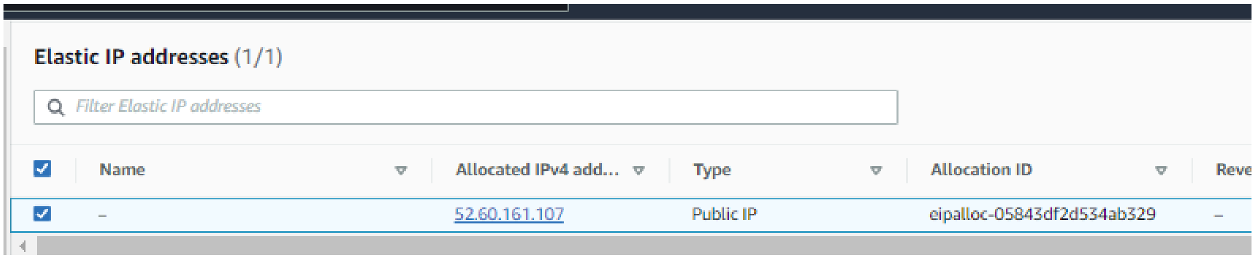
AWS Global Accelerator can provide global static IP addresses that are announced worldwide using anycast from AWS edge locations. This can help improve the availability and latency for your user traffic by using the Amazon global network. Learn more

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.

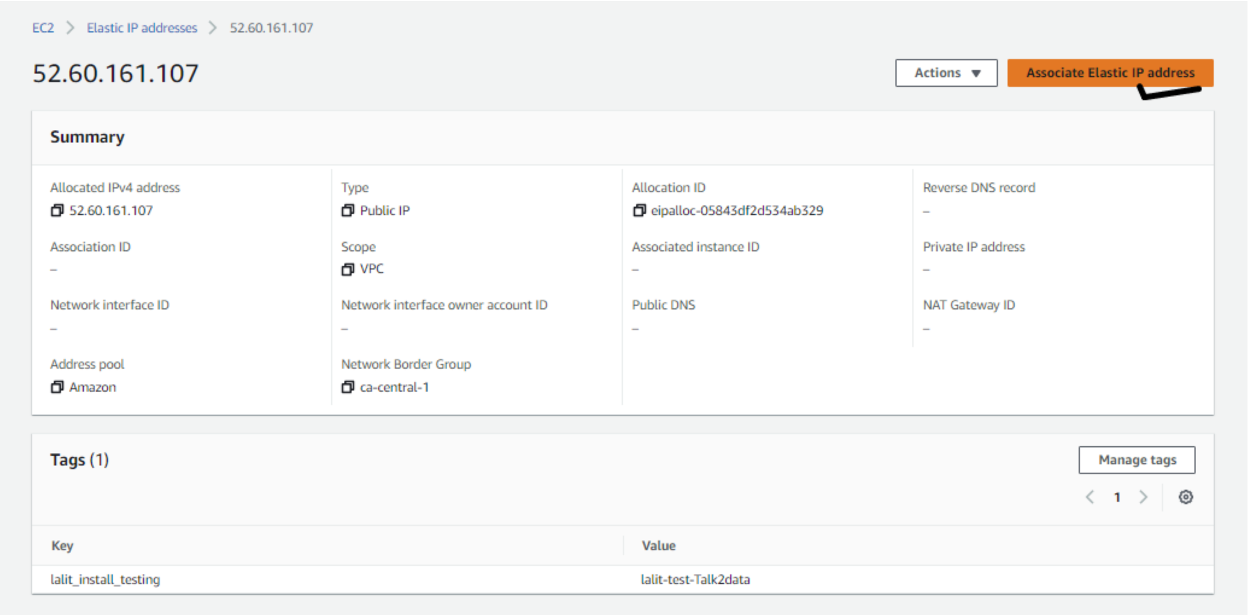
You can add up to 50 more tag



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

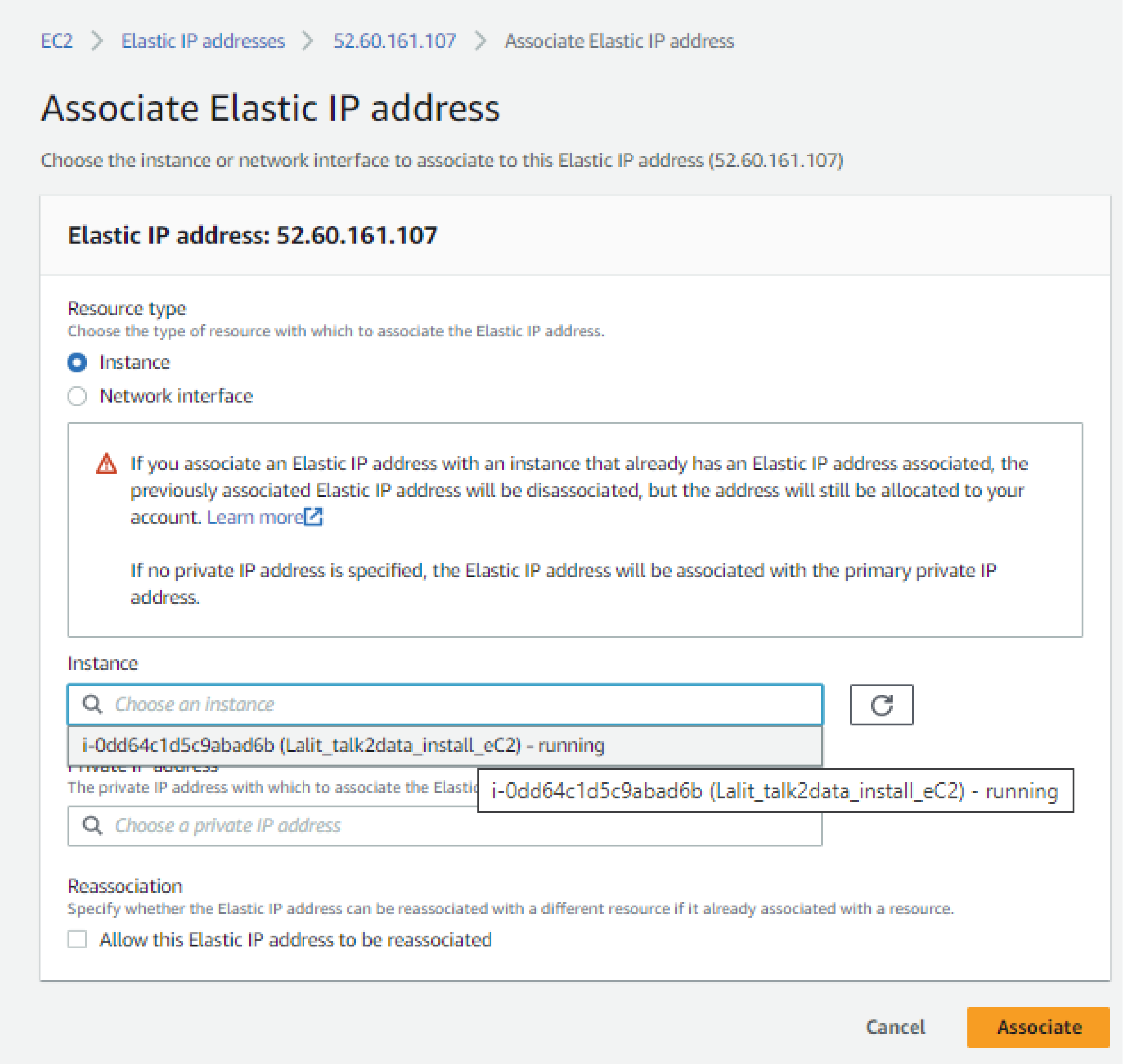
52.60.161.107

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)

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

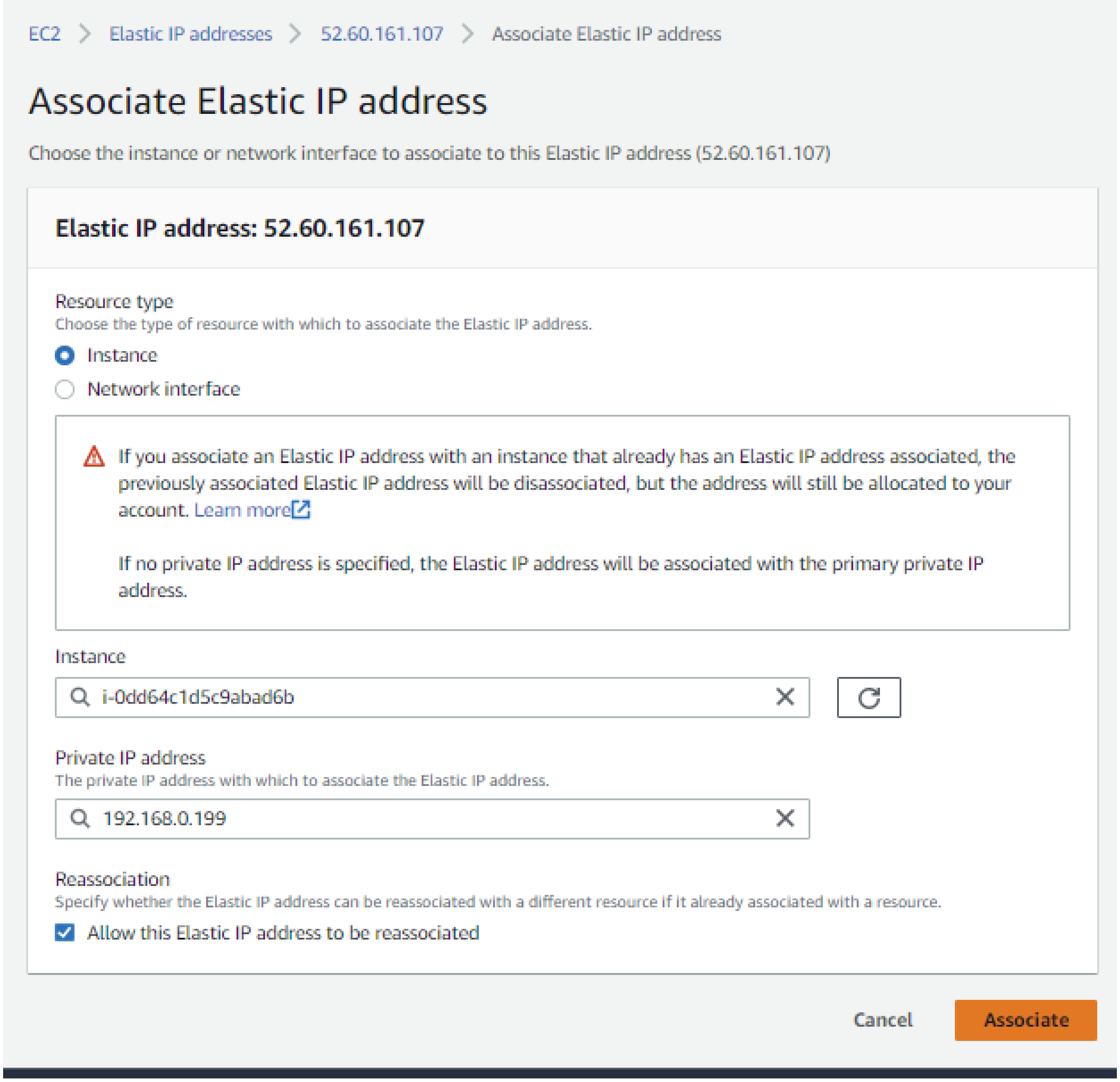
⚠ If you associate an Elastic IP address with an instance that already has an Elastic IP address associated, the previously associated Elastic IP address will be disassociated, but the address will still be allocated to your account. Learn more

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

Private IP address
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



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

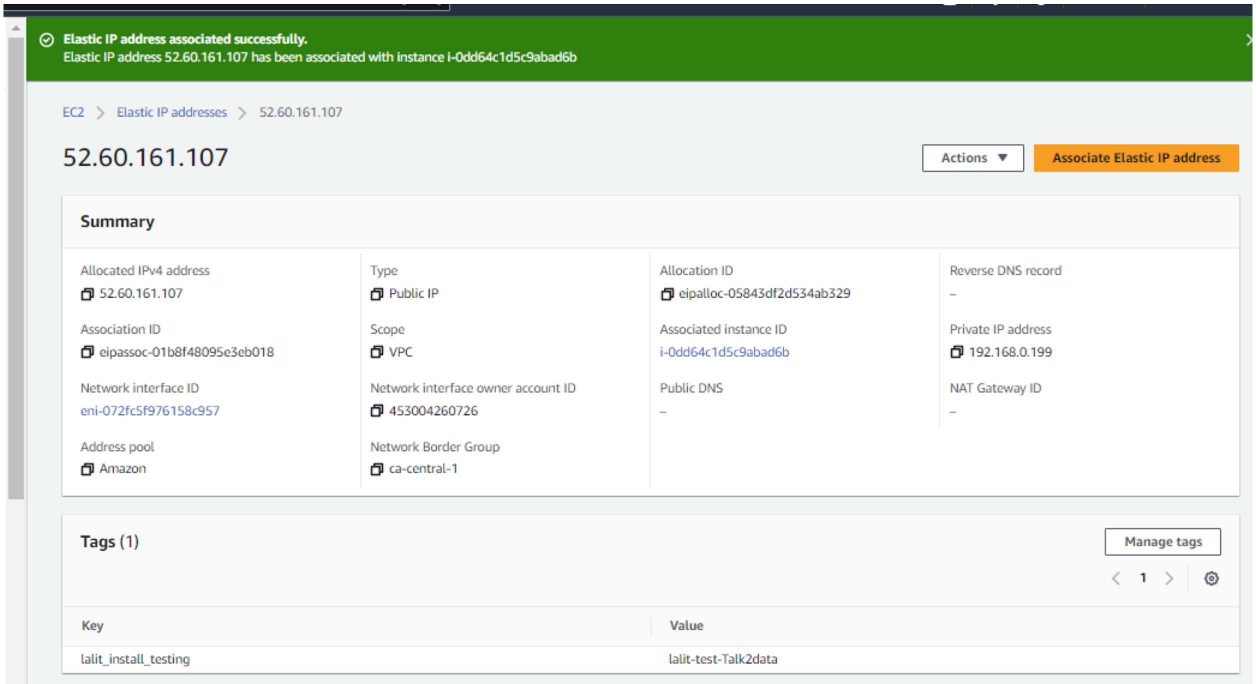
The private IP address with which to associate the Elastic IP address.

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


EC2 > Elastic IP addresses > 52.60.161.107
Actions ▼
Associate Elastic IP address
Manage tags
1

Elastic IP address associated successfully.
Elastic IP address 52.60.161.107 has been associated with instance i-0dd64c1d5c9abad6b

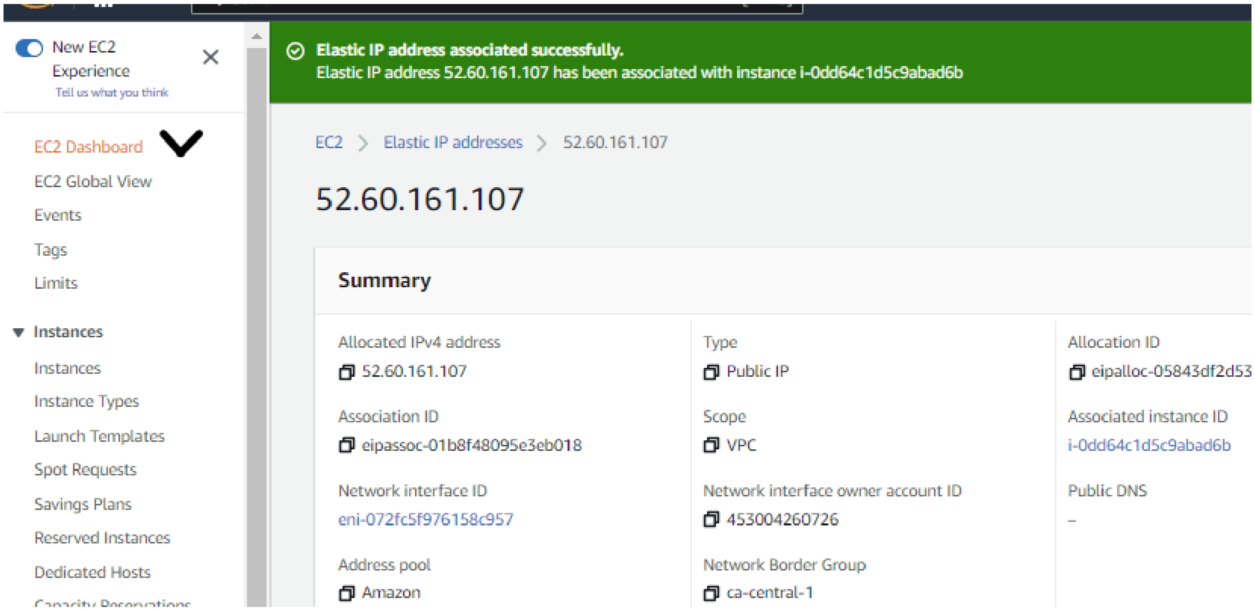
52.60.161.107

Summary

Allocated IPv4 address 52.60.161.107	Type Public IP	Allocation ID eipalloc-05843df2d534ab329	Reverse DNS record –
Association ID eipassoc-01b8f48095e3eb018	Scope VPC	Associated instance ID i-0dd64c1d5c9abad6b	Private IP address 192.168.0.199
Network interface ID eni-072fc5f976158c957	Network interface owner account ID 453004260726	Public DNS –	NAT Gateway ID –
Address pool Amazon	Network Border Group ca-central-1		

Tags (1)

Key	Value
lalit_install_testing	lalit-test-Talk2data



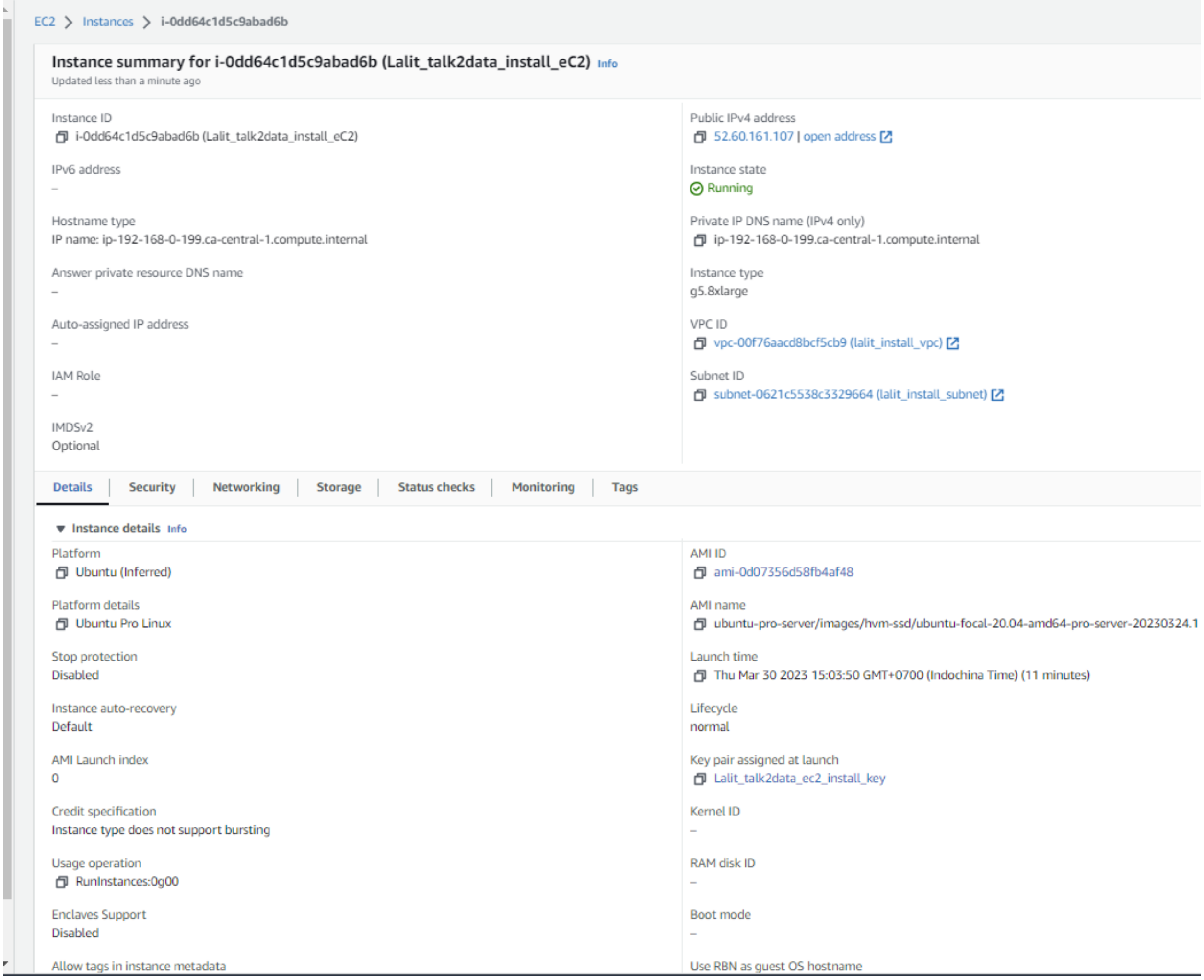
Elastic IP address associated successfully.
Elastic IP address 52.60.161.107 has been associated with instance i-0dd64c1d5c9abad6b

52.60.161.107

Summary

Allocated IPv4 address 52.60.161.107	Type Public IP	Allocation ID eipalloc-05843df2d53
Association ID eipassoc-01b8f48095e3eb018	Scope VPC	Associated instance ID i-0dd64c1d5c9abad6b
Network interface ID eni-072fc5f976158c957	Network interface owner account ID 453004260726	Public DNS –
Address pool Amazon	Network Border Group ca-central-1	

- Verify Public IP of instance



Instance summary for i-0dd64c1d5c9abad6b (Lalit_talk2data_install_eC2)
Updated less than a minute ago

Instance ID
i-0dd64c1d5c9abad6b (Lalit_talk2data_install_eC2)

IPv6 address
–

Hostname type
IP name: ip-192-168-0-199.ca-central-1.compute.internal

Answer private resource DNS name
–

Auto-assigned IP address
–

IAM Role
–

IMDSv2
Optional

Public IPv4 address
52.60.161.107 | open address

Instance state
Running

Private IP DNS name (IPv4 only)
ip-192-168-0-199.ca-central-1.compute.internal

Instance type
g5.8xlarge

VPC ID
vpc-00f76aacd8bcf5cb9 (lalit_install_vpc)

Subnet ID
subnet-0621c5538c3329664 (lalit_install_subnet)

Instance details

Platform
Ubuntu (Inferred)

Platform details
Ubuntu Pro Linux

Stop protection
Disabled

Instance auto-recovery
Default

AMI Launch index
0

Credit specification
Instance type does not support bursting

Usage operation
RunInstances:0g00

Enclaves Support
Disabled

Allow tags in instance metadata

AMI ID
ami-0d07356d58fb4af48

AMI name
ubuntu-pro-server/images/hvm-ssd/ubuntu-focal-20.04-amd64-pro-server-20230324.1

Launch time
Thu Mar 30 2023 15:03:50 GMT+0700 (Indochina Time) (11 minutes)

Lifecycle
normal

Key pair assigned at launch
Lalit_talk2data_ec2_install_key

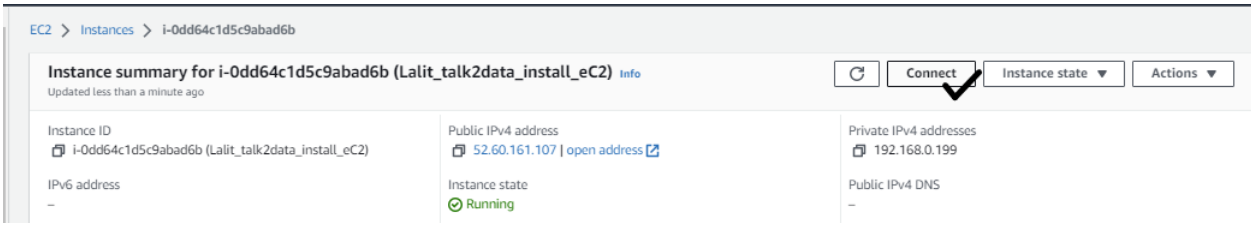
Kernel ID
–

RAM disk ID
–

Boot mode
–

Use RBN as guest OS hostname

- Login to instance



Instance summary for i-0dd64c1d5c9abad6b (Lalit_talk2data_install_eC2)
Updated less than a minute ago

Instance ID
i-0dd64c1d5c9abad6b (Lalit_talk2data_install_eC2)

IPv6 address
–

Public IPv4 address
52.60.161.107 | open address

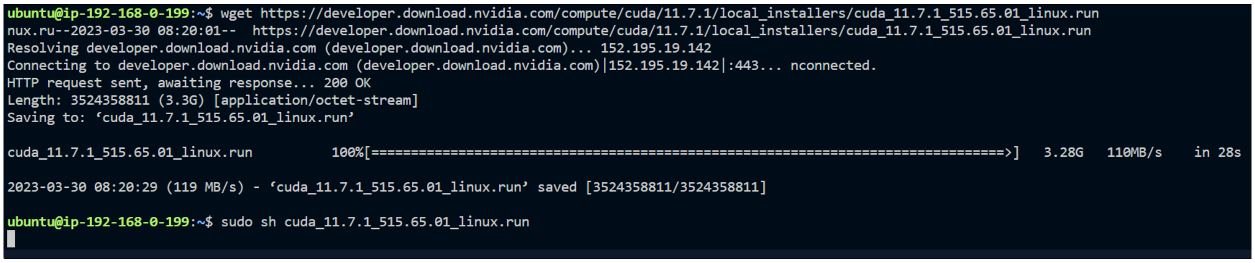
Instance state
Running

Private IPv4 addresses
192.168.0.199

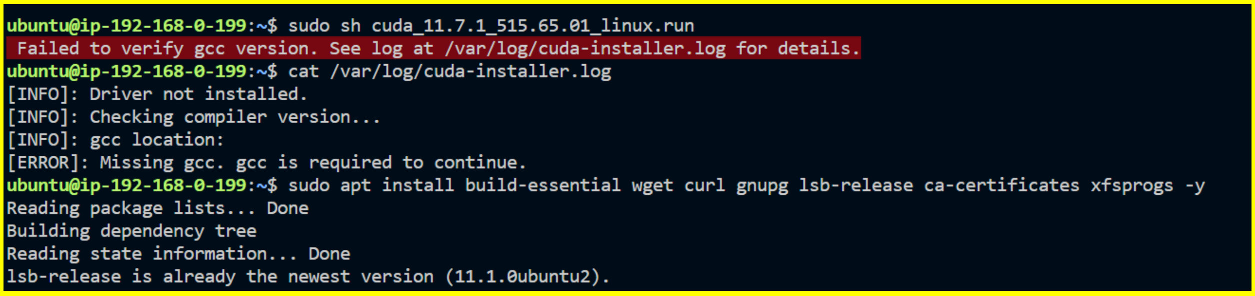
Public IPv4 DNS
–

- **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



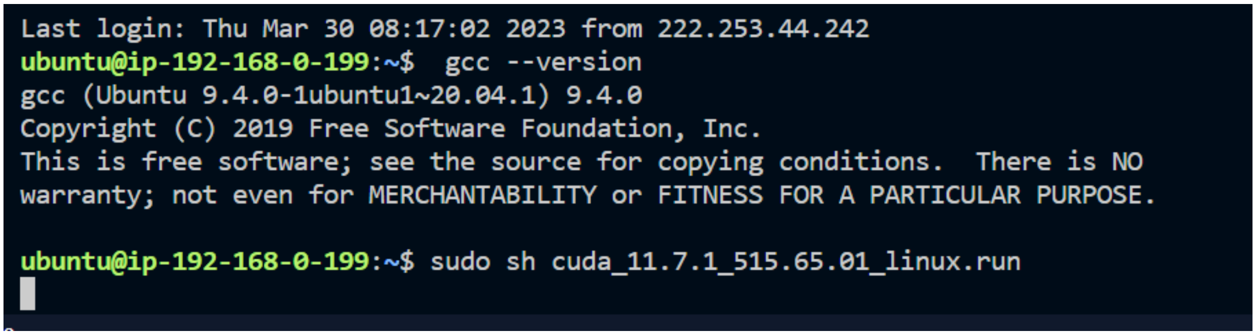
```
ubuntu@ip-192-168-0-199:~$ wget https://developer.download.nvidia.com/compute/cuda/11.7.1/local_installers/cuda_11.7.1_515.65.01_linux.run
nux.ru--2023-03-30 08:20:01--  https://developer.download.nvidia.com/compute/cuda/11.7.1/local_installers/cuda_11.7.1_515.65.01_linux.run
Resolving developer.download.nvidia.com (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).
```

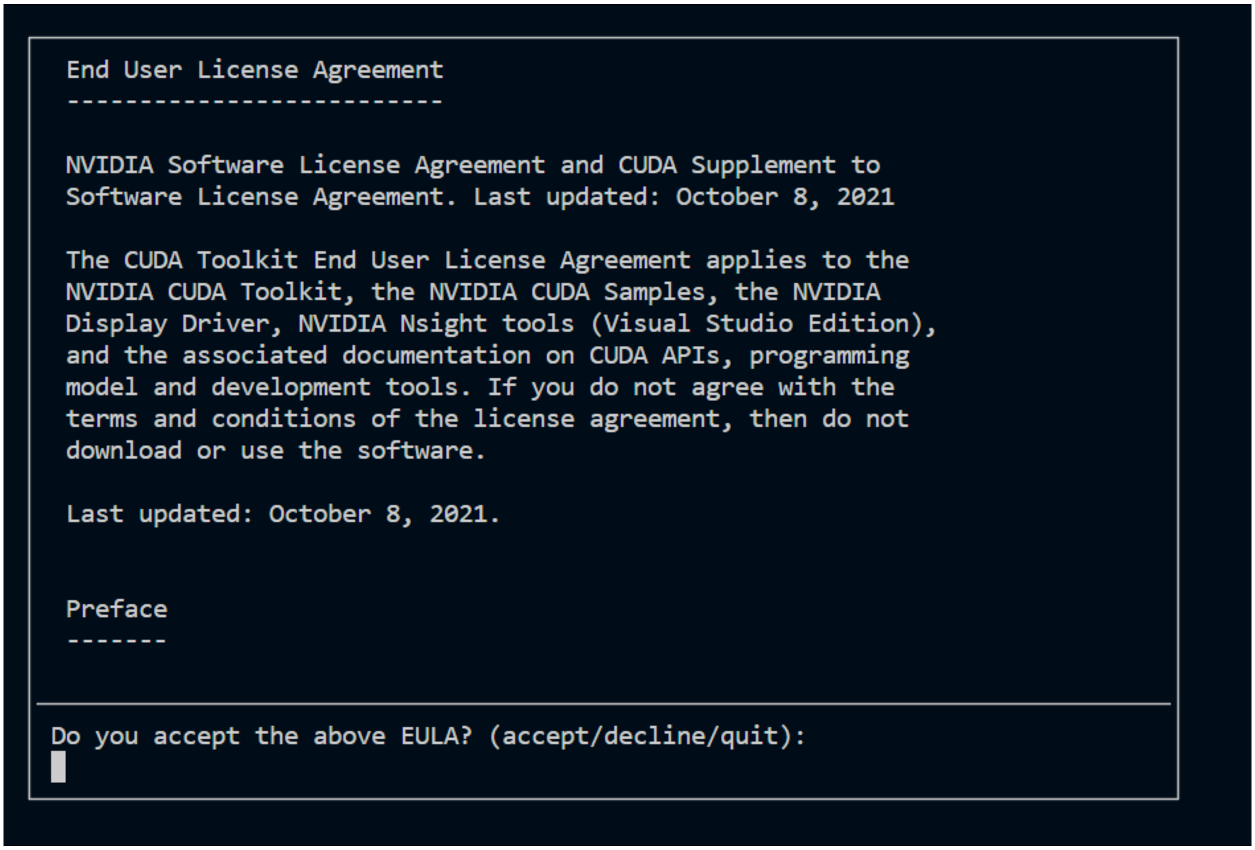
- **Reboot the server and try again.**

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

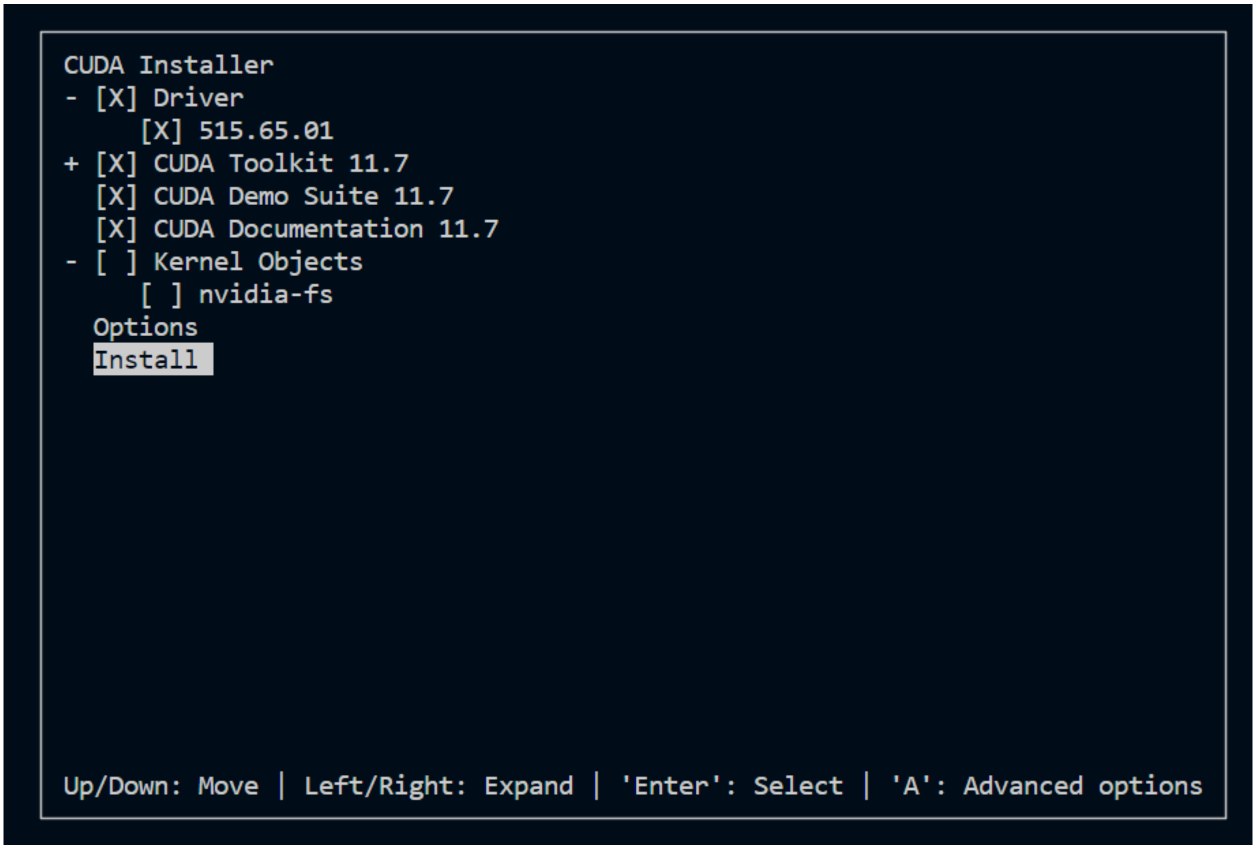


```
Last login: Thu Mar 30 08:17:02 2023 from 222.253.44.242
ubuntu@ip-192-168-0-199:~$  gcc --version
gcc (Ubuntu 9.4.0-1ubuntu1~20.04.1) 9.4.0
Copyright (C) 2019 Free Software Foundation, Inc.
This is free software; see the source for copying conditions.  There is NO
warranty; not even for MERCHANTABILITY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

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

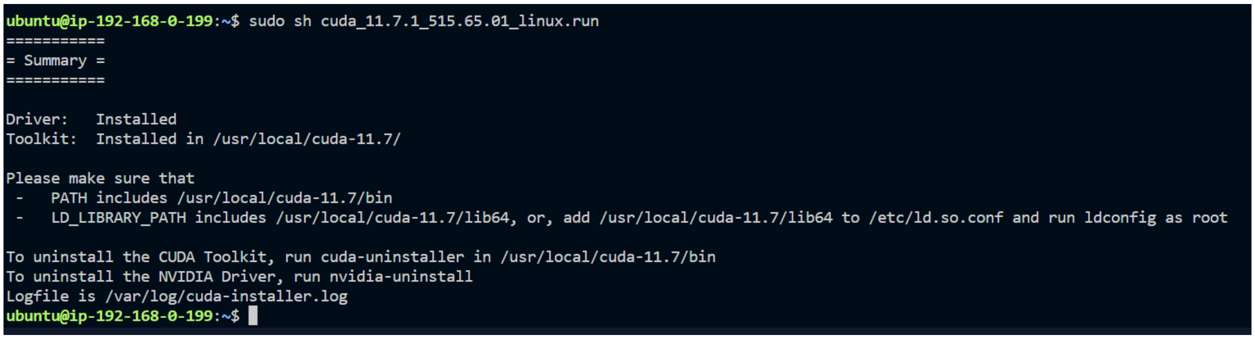
- **Accept license**



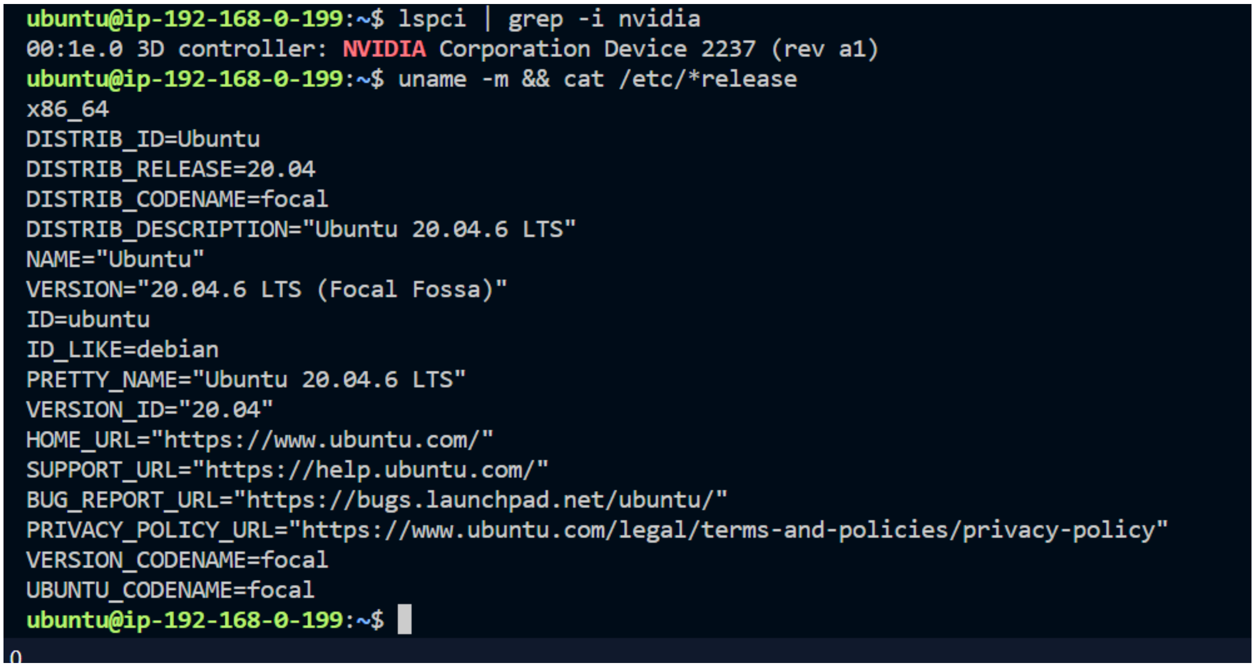
```
End User License Agreement
--------------------------

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
```

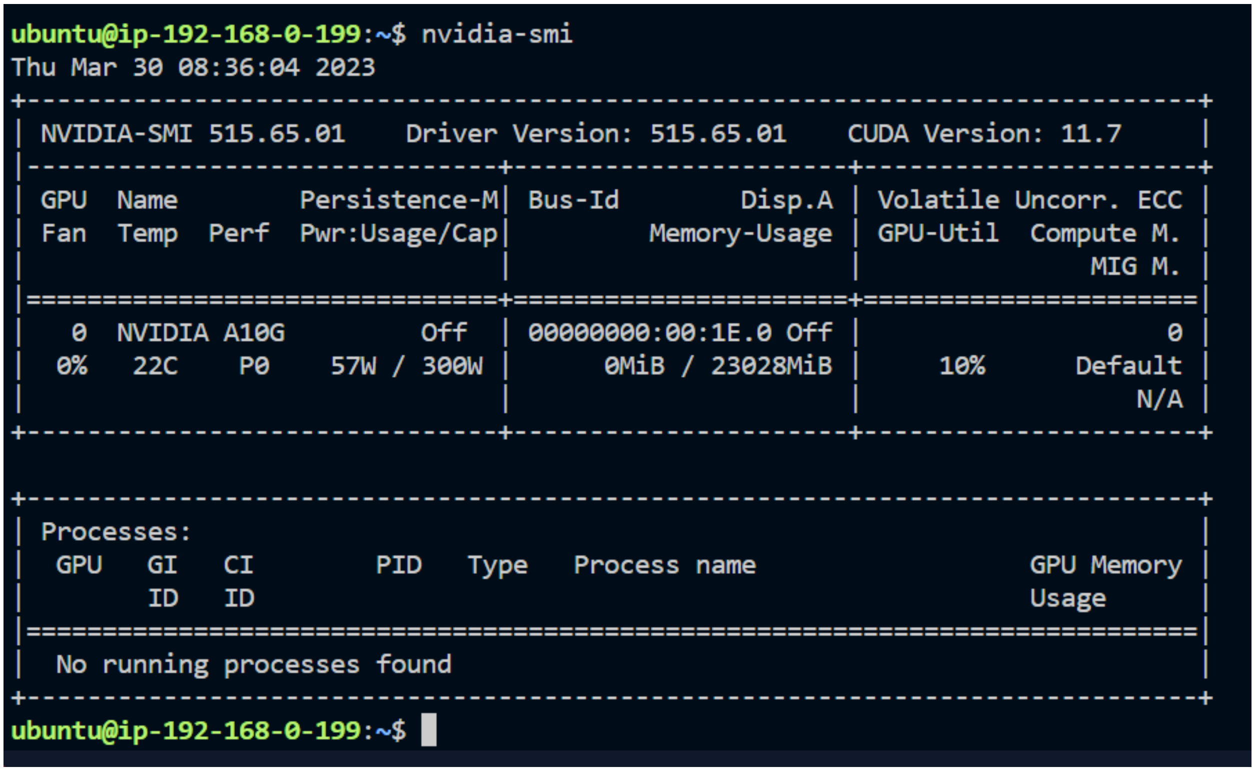


```
ubuntu@ip-192-168-0-199:~$ sudo sh cuda_11.7.1_515.65.01_linux.run
===========
= Summary =
===========

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:~$
```

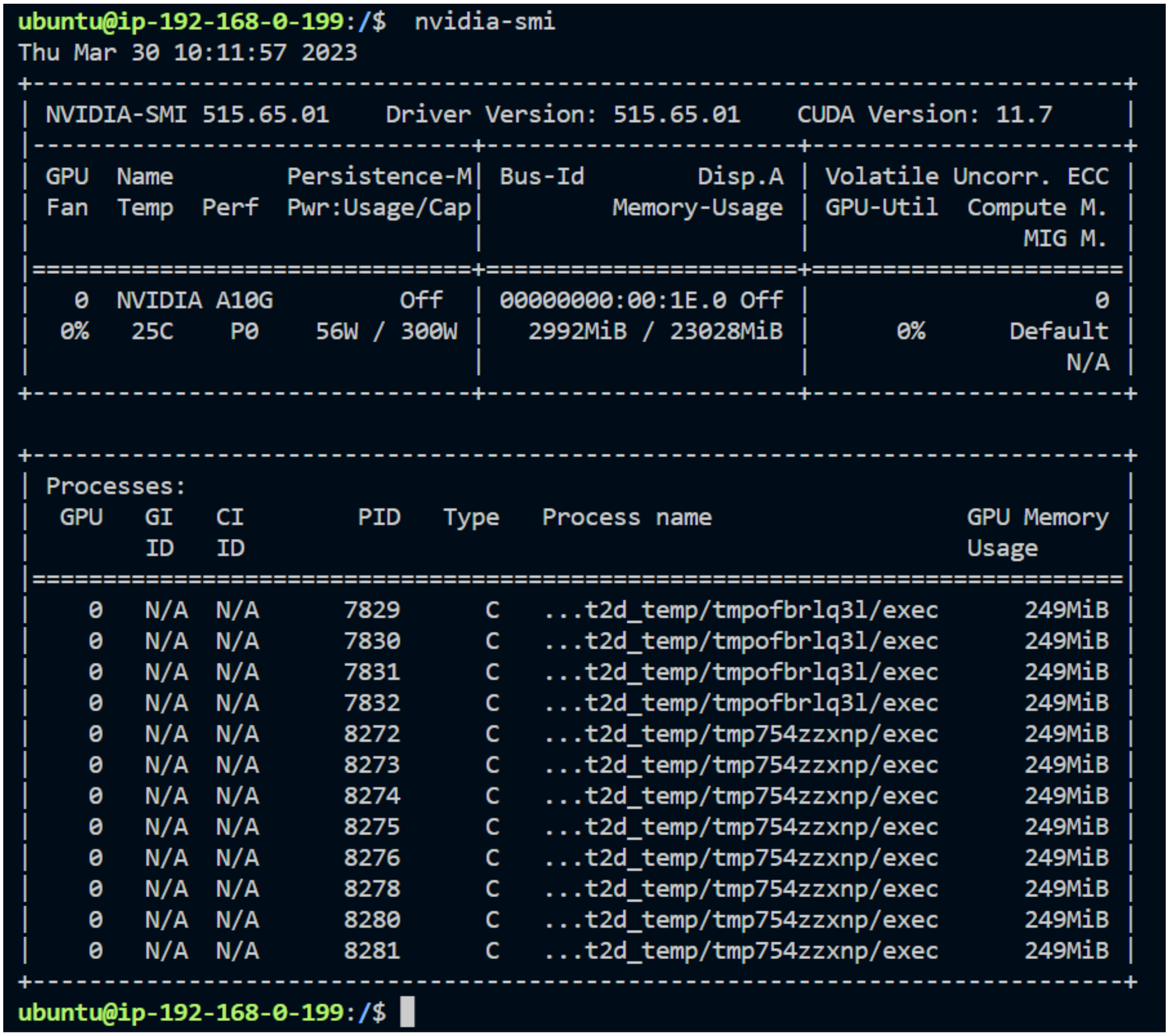


```
ubuntu@ip-192-168-0-199:~$ lspci | grep -i nvidia
00:1e.0 3D controller: NVIDIA Corporation Device 2237 (rev a1)
ubuntu@ip-192-168-0-199:~$ uname -m && cat /etc/*release
x86_64
DISTRIB_ID=Ubuntu
DISTRIB_RELEASE=20.04
DISTRIB_CODENAME=focal
DISTRIB_DESCRIPTION="Ubuntu 20.04.6 LTS"
NAME="Ubuntu"
VERSION="20.04.6 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:~$
```

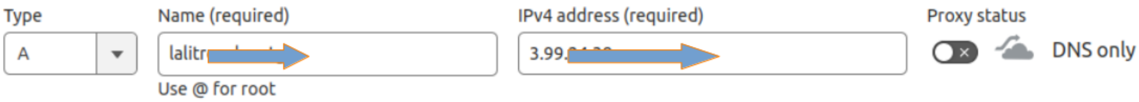


```
ubuntu@ip-192-168-0-199:/$  nvidia-smi
Thu Mar 30 10:11:57 2023
+-----------------------------------------------------------------------------+
| NVIDIA-SMI 515.65.01    Driver Version: 515.65.01    CUDA Version: 11.7     |
|-------------------------------+----------------------+----------------------+
| 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.**



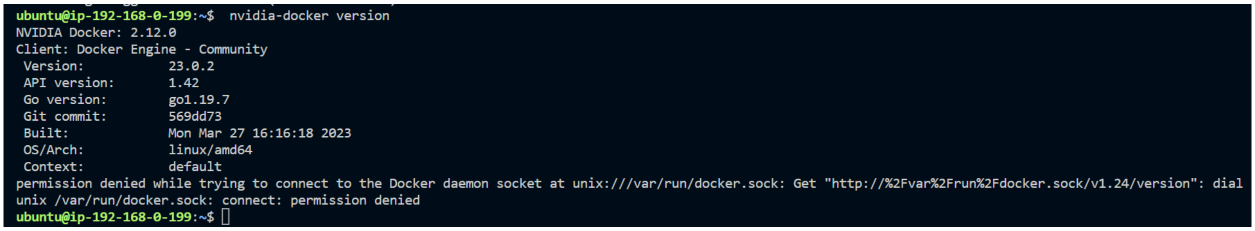
Type	Name (required)	IPv4 address (required)	Proxy status
A	lalitr[redacted]	3.99.[redacted]	DNS only

Use @ for root

- **Install the NVIDIA container toolkit.**

    ○ distribution=$(. /etc/os-release;echo $ID$VERSION_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

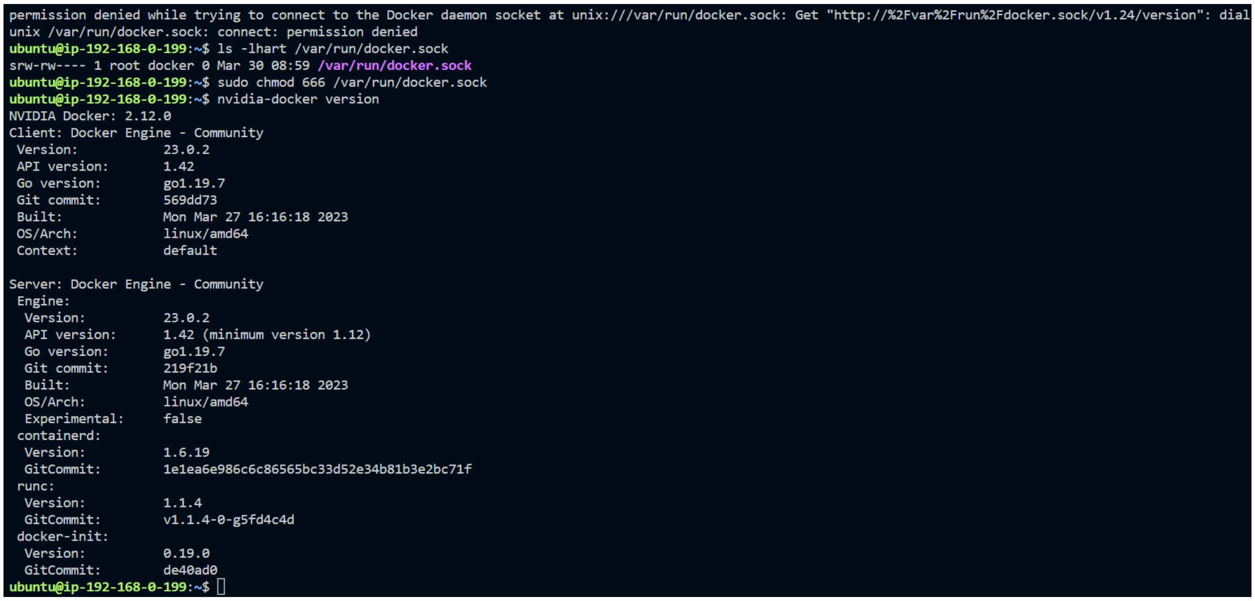


```
ubuntu@ip-192-168-0-199:~$  nvidia-docker version
NVIDIA Docker: 2.12.0
Client: Docker Engine - Community
 Version:           23.0.2
 API version:       1.42
 Go version:        go1.19.7
 Git commit:        569dd73
 Built:             Mon Mar 27 16:16:18 2023
 OS/Arch:           linux/amd64
 Context:           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:**

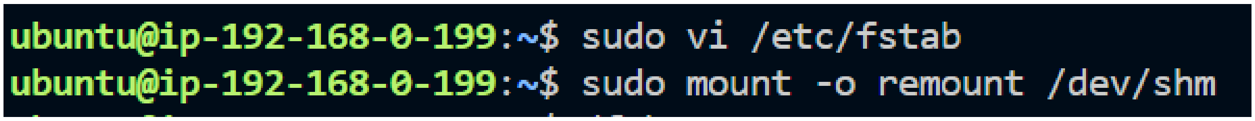
permission denied while trying to connect to the Docker daemon socket at unix:///var/run/docker.sock: Get "http://%2Fvar%2Frun%2Fdocker.sock/v1.24/version": dial unix /var/run/docker.sock: connect: permission denied
ubuntu@ip-192-168-0-199:~$ ls -lhart /var/run/docker.sock
srw-rw---- 1 root docker 0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

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

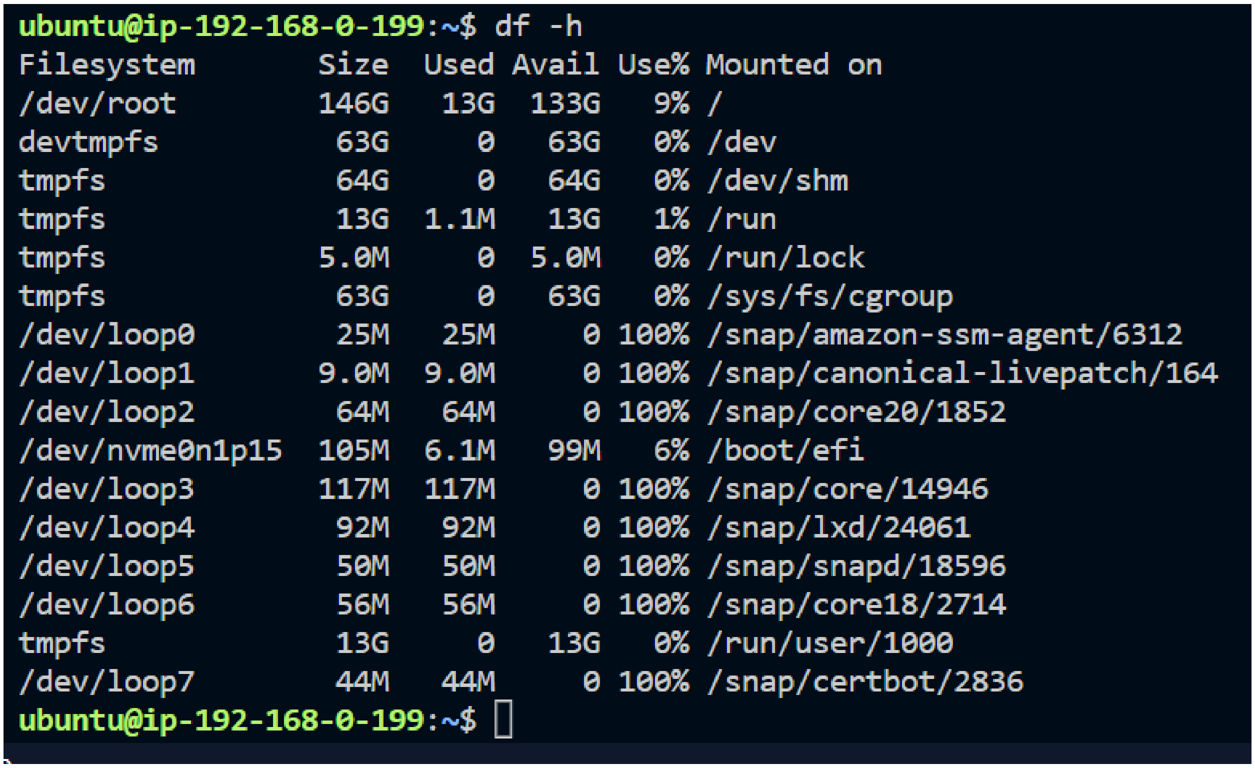


```
permission denied while trying to connect to the Docker daemon socket at unix:///var/run/docker.sock: Get "http://%2Fvar%2Frun%2Fdocker.sock/v1.24/version": dial
unix /var/run/docker.sock: connect: permission denied
ubuntu@ip-192-168-0-199:~$ ls -lhart /var/run/docker.sock
srw-rw---- 1 root docker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

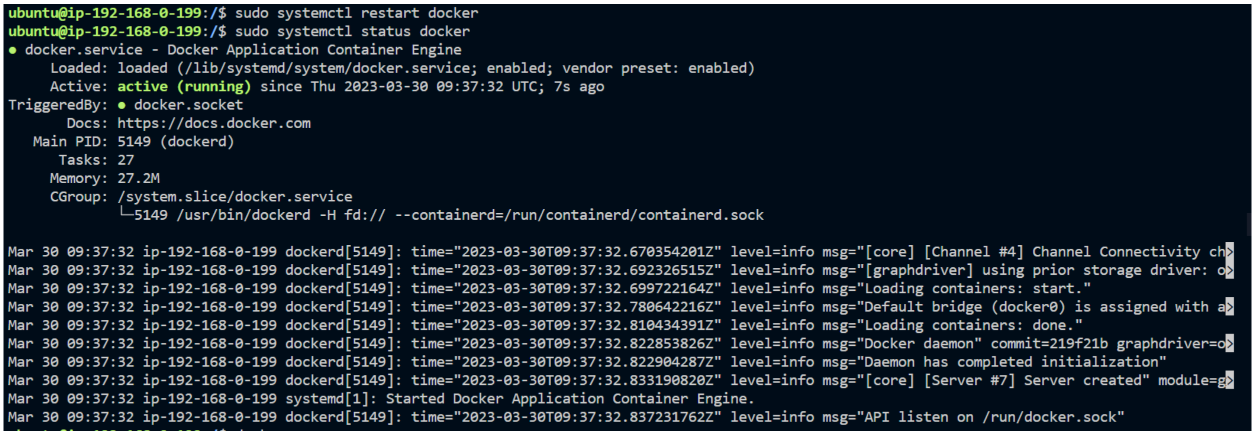
- **We can install SSL certificates later too. It's not mandatory during installation.**
  - sudo mkdir -p /config/ssl
  - sudo mv tls.crt /config/ssl
  - sudo mv tls.key /config/ssl

- **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

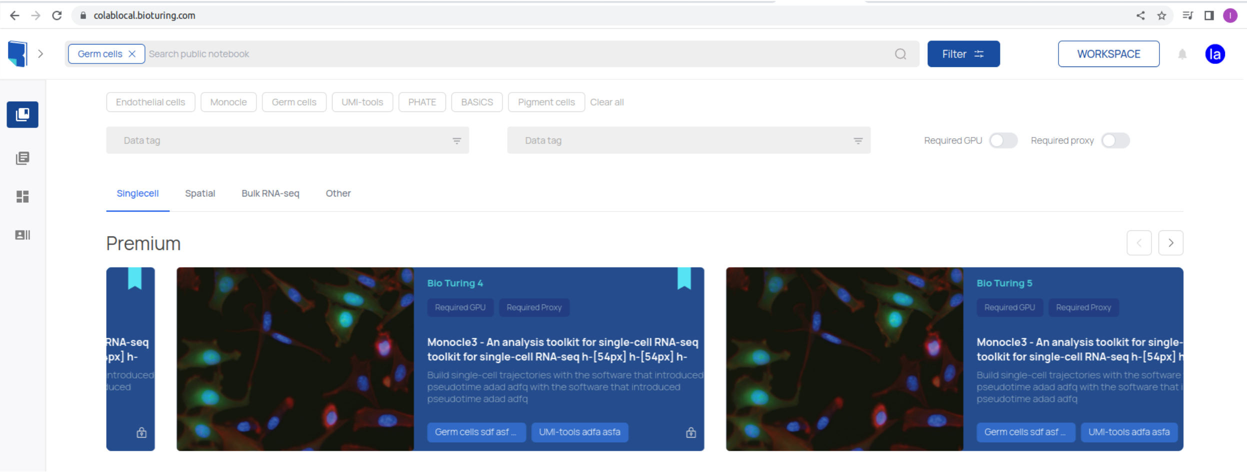
- **Run the docker image.**

```
docker run -it -d \
   -e APP_DOMAIN_URL='<yourcompany.com>' \
   -e COLAB_TOKEN='<your token from BioTuring>' \
   -e SSO_DOMAINS='<your company email address, example: @bioturing.com>' \
   -e ADMIN_USER_NAME='<your company username>' \
   -e ADMIN_PASSWORD='<your company password>' \
   -e AMZ_S3_FUSE_MODE='<S3 fuse mode [true / false]>' \
   -e API_KEY='<API key value>' \
   -e COLLABORATIVE_MODE='<colloaborative mode [true / false]>' \
   -e CONTENT_PATH='<content path: /s3/colab/content>' \
   -e DATA_PATH='<data path: /s3/colab/data>' \
   -e DEBUG_MODE='<debug mode [true / false]>' \
   -e DEV_MODE='<dev mode [true / false]>' \
   -e HOST='<host IP address>' \
   -e HUB_API_KEY='<hub api key>' \
   -e HUB_LIST_IPS='<hub list>' \
   -e HUB_SECRET_COOKIE='<hub secret cookie value>' \
   -e JAEGER_ENDPOINT='<jaegerapp-agent end potint :
aegerapp-agent.cattle-system.svc.cluster.local>' \
   -e JWT_SECRET_KEY='<jwt secret key>' \
   -e LOG_PATH='<log path>' \
   -e MANUAL_REGISTER_KERNEL='<manual register kernel>' \
   -e MEMCACHED_LIST='< memcached list :
memcached.database.svc.cluster.local=11211>' \
   -e MQTT_LIST_IPS='<mqtt list IP :
biocolab-0.biocolab.bioturing.svc.cluster.local,biocolab-1.biocolab.bioturing.svc.cluster.l
ocal>' \
   -e PG_DATABASE=<'database name: biocolab>' \
   -e PG_HOST=<'postgresql host name: postgresql-bhub.database.svc.cluster.local>' \
   -e PG_HUB_DATABASE=<'postgresql hub database name: bioturing>' \
   -e PG_USERNAME=<'postgresql user name: pgpostgres>' \
   -e PG_PASSWORD=<'postgresql password name: pgbioturing>' \
   -e REDIS_LIST=<' redis list: colabredis.bioturing.svc.cluster.local=6379>' \
   -e REDIS_PASSWORD=<'redispassword: rbioturing>' \
   -e ROOT_PATH=<'root path: /srv/apps>' \
   -e SECRET_KEY=<'secret key>' \
   -e SERVICE_ENDPOINT=<'service endpoint: biocolab.bioturing.svc.cluster.local>' \
   -e TMP_PATH=<'temp path: /srv/apps/tmp>' \
   -e TRACING_MODE=<'tracing mode [true / false] >' \
   -e UPLOAD_PATH=<'upload /s3/colab/upload>' \
   -e USE_REDIS_CACHE=<'use redis cache [true / false]>' \
   -e USER_PATH=<'user path: /srv/apps/user>' \
   -p 443:443 \
   -p 80:80 \
   bioturing/bioturing-colab:1.0.1
```



```
ubuntu@ip-192-168-0-199:/$ sudo systemctl restart docker
ubuntu@ip-192-168-0-199:/$ sudo systemctl status docker
● docker.service - Docker Application Container Engine
     Loaded: loaded (/lib/systemd/system/docker.service; enabled; vendor preset: enabled)
     Active: active (running) since Thu 2023-03-30 09:37:32 UTC; 7s ago
TriggeredBy: ● docker.socket
       Docs: https://docs.docker.com
   Main PID: 5149 (dockerd)
      Tasks: 27
     Memory: 27.2M
     CGroup: /system.slice/docker.service
             └─5149 /usr/bin/dockerd -H fd:// --containerd=/run/containerd/containerd.sock

Mar 30 09:37:32 ip-192-168-0-199 dockerd[5149]: time="2023-03-30T09:37:32.670354201Z" level=info msg="[core] [Channel #4] Channel Connectivity ch>
Mar 30 09:37:32 ip-192-168-0-199 dockerd[5149]: time="2023-03-30T09:37:32.692326515Z" level=info msg="[graphdriver] using prior storage driver: o>
Mar 30 09:37:32 ip-192-168-0-199 dockerd[5149]: time="2023-03-30T09:37:32.699722164Z" level=info msg="Loading containers: start."
Mar 30 09:37:32 ip-192-168-0-199 dockerd[5149]: time="2023-03-30T09:37:32.780642216Z" level=info msg="Default bridge (docker0) is assigned with a>
Mar 30 09:37:32 ip-192-168-0-199 dockerd[5149]: time="2023-03-30T09:37:32.810434391Z" level=info msg="Loading containers: done."
Mar 30 09:37:32 ip-192-168-0-199 dockerd[5149]: time="2023-03-30T09:37:32.822853826Z" level=info msg="Docker daemon" commit=219f21b graphdriver=o>
Mar 30 09:37:32 ip-192-168-0-199 dockerd[5149]: time="2023-03-30T09:37:32.822904287Z" level=info msg="Daemon has completed initialization"
Mar 30 09:37:32 ip-192-168-0-199 dockerd[5149]: time="2023-03-30T09:37:32.833190820Z" level=info msg="[core] [Server #7] Server created" module=g>
Mar 30 09:37:32 ip-192-168-0-199 systemd[1]: Started Docker Application Container Engine.
Mar 30 09:37:32 ip-192-168-0-199 dockerd[5149]: time="2023-03-30T09:37:32.837231762Z" level=info msg="API listen on /run/docker.sock"
```

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



RNA-seq
px] h-
ntroduced
luced

Monocle3 - An analysis toolkit for single-cell RNA-seq toolkit for single-cell RNA-seq h-[54px] h-[54px] h-

Build single-cell trajectories with the software that introduced pseudotime adad adfq with the software that introduced pseudotime adad adfq

Monocle3 - An analysis toolkit for single- toolkit for single-cell RNA-seq h-[54px] h

Build single-cell trajectories with the software pseudotime adad adfq with the software that i pseudotime adad adfq

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]
```

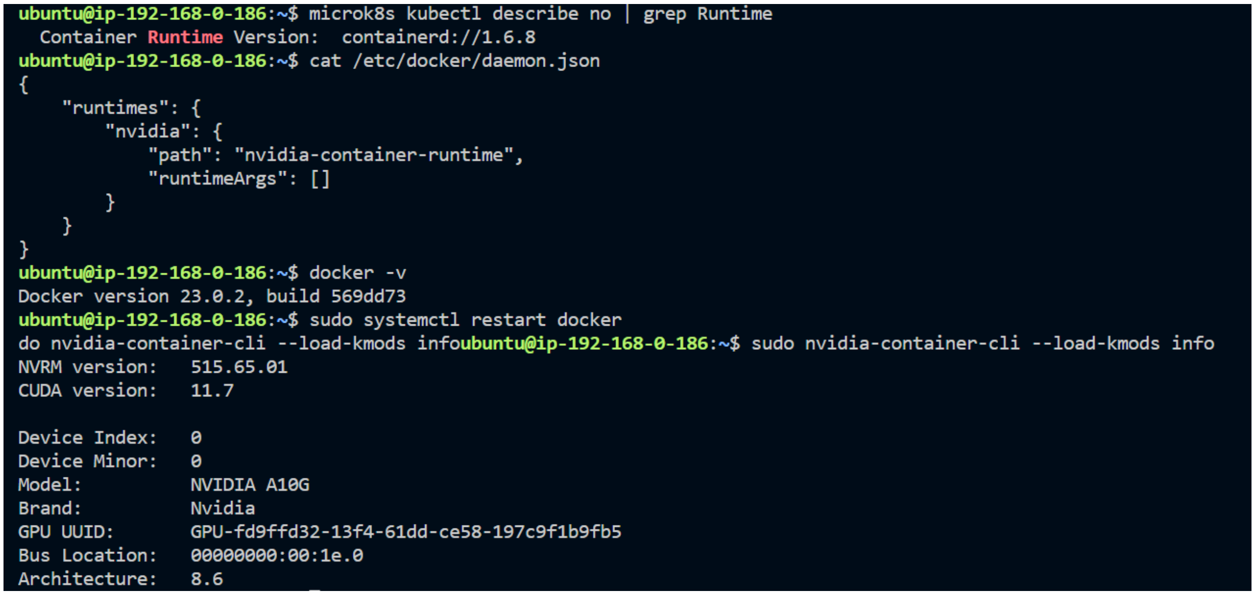
```
  privileged_without_host_devices = false

  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.

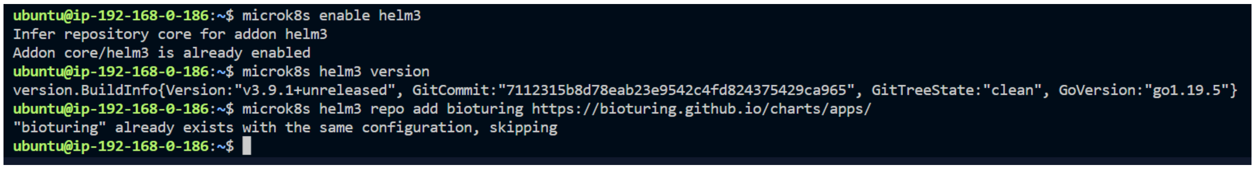
  First, check the Helm version
  - microk8s enable helm3
  - microk8s helm3 version

- **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: >=

	Type	Default	Description
image.tag	string	"1.0.1"	image tag
secret.data.domain	string	"colab.com"	your domain
secret.data.colab_token	string	""	Biocolab token
secret.data.aria2c_list_ips	string	""	aria2c list
secret.data.host	int		
secret.data.hub_api_key	string	""	
secret.data.hub_list_ips	string	""	
secret.data.hub_secret_cookie	string	"false"	
secret.data.jaeger_endpoint	string	"jaegerapp-agent.cattle-system.svc.cluster.local"	

secret.data.jwt_secret_key	string	"C.UTF-8"	
secret.data.log_path	string	"/logs"	
secret.data.manual_register_kernel	bool	"true"	
secret.data.memcached_list	string	"memcached.database.svc.cluster.local=11211"	
secret.data.mqtt_list_ips	string	""	
secret.admin.username	string	admin	
secret.admin.password	string	turing2022	
secret.colabvariable.amz_s3_fuse_mode	bool	"true"	
secret.colabvariable.api_key	string	""	
secret.colabvariable.collaborative_mode	bool	"false"	
secret.colabvariable.content_path	string	"/s3/colab/content"	
secret.colabvariable.data_path	bool	"/s3/colab/data"	
secret.colabvariable.debug_mode	bool	"true"	
secret.colabvariable.dev_mode	bool	"false"	
secret.colabvariable.upload_path	string	"/s3/colab/upload"	
secret.colabvariable.secret_key	bool	"true"	
secret.colabvariable.service_endpoint	string	"biocolab.bioturing.svc.cluster.local"	
secret.colabvariable.tmp_path	string	"/srv/apps/tmp"	
secret.colabvariable.tracing_mode	bool	"true"	

secret.pgvariable.pg_database	**string**	**"biocolab"**	
secret.pgvariable.pg_host	**string**	**"postgresql-bhub.database.svc.cluster.local"**	
secret.pgvariable.pg_hub_database	**string**	**"bioturing"**	
secret.pgvariable.pg_username	**string**	**"pgpostgres"**	
secret.pgvariable.pg_password	**string**	**"pgpostgres"**	
secret.redisvariable.redis_list	**string**	**"colabredis.bioturing.svc.cluster.local=6379"**	
secret.redisvariable.redis_password	**bool**	**"rbioturing"**	
secret.redisvariable.use_redis_cache	**bool**	**"false"**	
secret.redisvariable.user_path	**string**	**"/srv/apps/user"**	
secret.redisvariable.root_path	**string**	**"/srv/apps"**	
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"**	

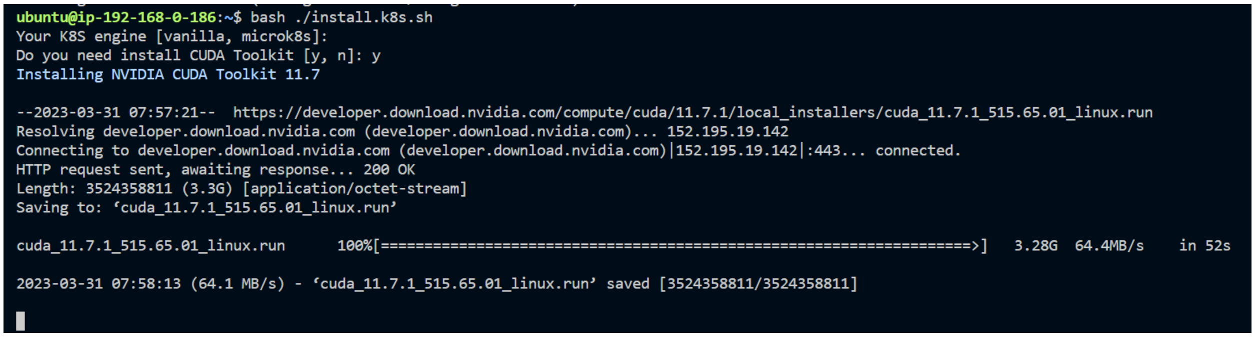
- **For Containerd runtime :**

```
gpu.runtimeClassName="nvidia"
```

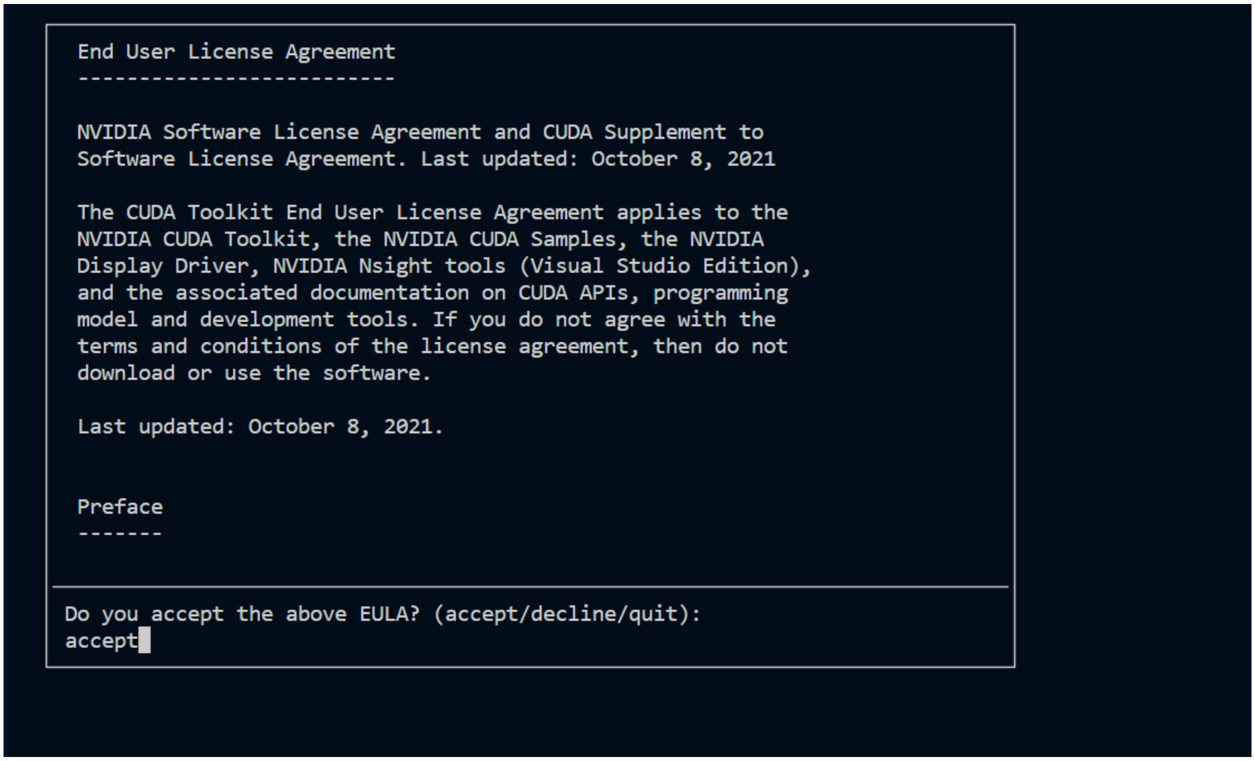
- **For Docker runtime :**

gpu.runtimeClassName=""

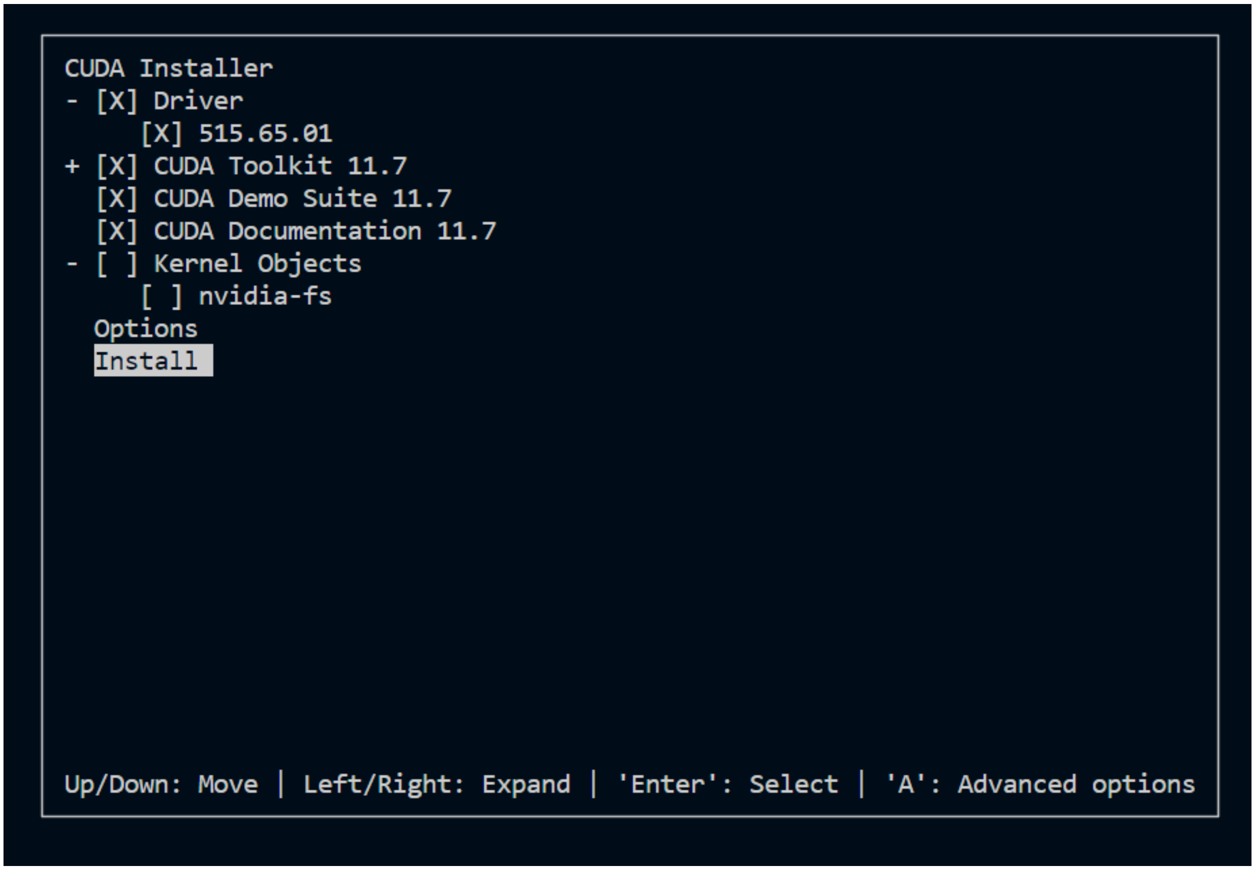
- **Simple Installation (Recommended):**



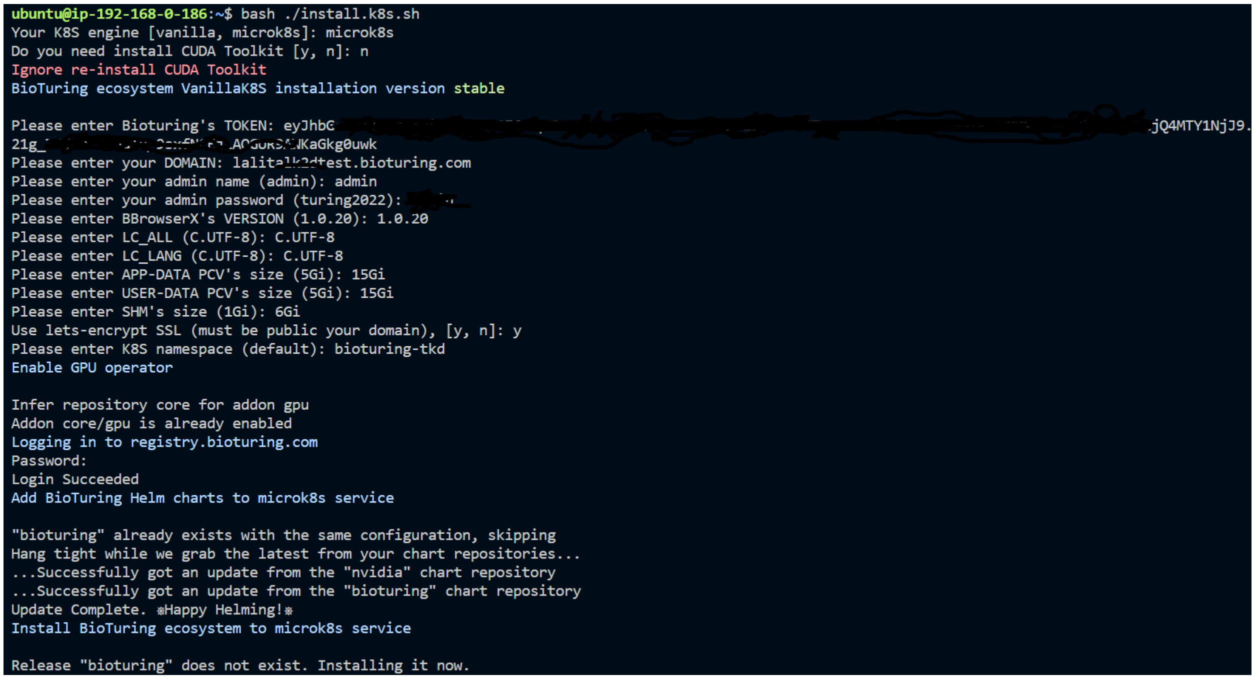
```
ubuntu@ip-192-168-0-186:~$ bash ./install.k8s.sh
Your K8S engine [vanilla, microk8s]:
Do you need install CUDA Toolkit [y, n]: y
Installing NVIDIA CUDA Toolkit 11.7

--2023-03-31 07:57:21--  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... connected.
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  64.4MB/s    in 52s

2023-03-31 07:58:13 (64.1 MB/s) - ‘cuda_11.7.1_515.65.01_linux.run’ saved [3524358811/3524358811]

```



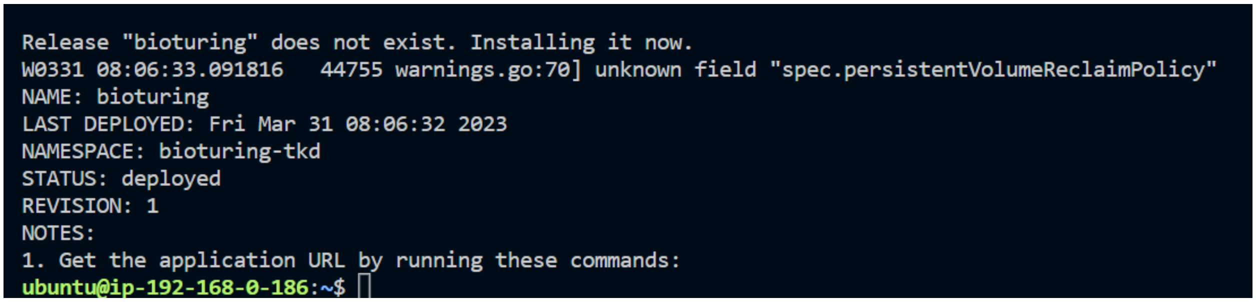
```
  End User License Agreement
  --------------------------

  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
```



```
ubuntu@ip-192-168-0-186:~$ bash ./install.k8s.sh
Your K8S engine [vanilla, microk8s]: microk8s
Do you need install CUDA Toolkit [y, n]: n
Ignore re-install CUDA Toolkit
BioTuring ecosystem VanillaK8S installation version stable

Please enter Bioturing's TOKEN: eyJhbG[redacted]jQ4MTY1NjJ9.
21g_[redacted]KaGkg0uwk
Please enter your DOMAIN: lalit[redacted]test.bioturing.com
Please enter your admin name (admin): admin
Please enter your admin password (turing2022): [redacted]
Please enter BBrowserX's VERSION (1.0.20): 1.0.20
Please enter LC_ALL (C.UTF-8): C.UTF-8
Please enter LC_LANG (C.UTF-8): C.UTF-8
Please enter APP-DATA PCV's size (5Gi): 15Gi
Please enter USER-DATA PCV's size (5Gi): 15Gi
Please enter SHM's size (1Gi): 6Gi
Use lets-encrypt SSL (must be public your domain), [y, n]: y
Please enter K8S namespace (default): bioturing-tkd
Enable GPU operator

Infer repository core for addon gpu
Addon core/gpu is already enabled
Logging in to registry.bioturing.com
Password:
Login Succeeded
Add BioTuring Helm charts to microk8s service

"bioturing" already exists with the same configuration, skipping
Hang tight while we grab the latest from your chart repositories...
...Successfully got an update from the "nvidia" chart repository
...Successfully got an update from the "bioturing" chart repository
Update Complete. ⎈Happy Helming!⎈
Install BioTuring ecosystem to microk8s service

Release "bioturing" does not exist. Installing it now.
```



```
Release "bioturing" does not exist. Installing it now.
W0331 08:06:33.091816   44755 warnings.go:70] unknown field "spec.persistentVolumeReclaimPolicy"
NAME: bioturing
LAST DEPLOYED: Fri Mar 31 08:06:32 2023
NAMESPACE: bioturing-tkd
STATUS: deployed
REVISION: 1
NOTES:
1. Get the application URL by running these commands:
ubuntu@ip-192-168-0-186:~$ 
```

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

For Microk8s:

```
microk8s helm3 repo update
microk8s helm3 registry login -u admin registry.bioturing.com
microk8s helm3 upgrade --install --set secret.data.domain="${APP_DOMAIN_URL}" \
   --set secret.data.colab_token="${COLAB_TOKEN}" \
   --set secret.data.aria2c_list_ips="${ARIA2C_LIST_IPS}" \
   --set secret.data.host="${HOST}" \
   --set secret.data.hub_api_key="${HUB_API_KEY}" \
   --set secret.data.hub_list_ips="${HUB_LIST_IPS}" \
   --set secret.data.hub_secret_cookie="${HUB_SECRET_COOKIE}" \
   --set secret.data.jaeger_endpoint="${JAEGER_ENDPOINT}" \
   --set secret.data.jwt_secret_key="${JWT_SECRET_KEY}" \
   --set secret.data.log_path="${LOG_PATH}" \
   --set secret.data.manual_register_kernel="${MANUAL_REGISTER_KERNEL}" \
   --set secret.data.memcached_list="${MEMCACHED_LIST}" \
   --set secret.data.mqtt_list_ips="${MQTT_LIST_IPS}" \
   --set secret.admin.username="${ADMIN_USER_NAME}" \
   --set secret.admin.password="${ADMIN_PASSWORD}" \
   --set secret.colabvariable.amz_s3_fuse_mode="${AMZ_S3_FUSE_MODE}" \
   --set secret.colabvariable.api_key="${API_KEY}" \
   --set secret.colabvariable.collaborative_mode="${COLLABORATIVE_MODE}" \
   --set secret.colabvariable.content_path="${CONTENT_PATH}" \
   --set secret.colabvariable.data_path="${DATA_PATH}" \
   --set secret.colabvariable.debug_mode="${DEBUG_MODE}" \
   --set secret.colabvariable.dev_mode="${DEV_MODE}" \
   --set secret.colabvariable.upload_path="${UPLOAD_PATH}" \
   --set secret.colabvariable.secret_key="${SECRET_KEY}" \
   --set secret.colabvariable.service_endpoint="${SERVICE_ENDPOINT}" \
   --set secret.colabvariable.tmp_path="${TMP_PATH}" \
   --set secret.colabvariable.tracing_mode="${TRACING_MODE}" \
   --set secret.pgvariable.pg_database="${PG_DATABASE}" \
   --set secret.pgvariable.pg_host="${PG_HOST}" \
   --set secret.pgvariable.pg_hub_database="${PG_HUB_DATABASE}" \
   --set secret.pgvariable.pg_username="${PG_USERNAME}" \
   --set secret.pgvariable.pg_password="${PG_PASSWORD}" \
   --set secret.redisvariable.redis_list="${REDIS_LIST}" \
   --set secret.redisvariable.redis_password="${REDIS_PASSWORD}" \
   --set secret.redisvariable.use_redis_cache="${USE_REDIS_CACHE}" \
   --set secret.redisvariable.user_path="${USER_PATH}" \
   --set secret.redisvariable.root_path="${ROOT_PATH}" \
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.domain="${APP_DOMAIN_URL}" \
    --set secret.data.colab_token="${COLAB_TOKEN}" \
    --set secret.data.aria2c_list_ips="${ARIA2C_LIST_IPS}" \
    --set secret.data.host="${HOST}" \
    --set secret.data.hub_api_key="${HUB_API_KEY}" \
    --set secret.data.hub_list_ips="${HUB_LIST_IPS}" \
    --set secret.data.hub_secret_cookie="${HUB_SECRET_COOKIE}" \
    --set secret.data.jaeger_endpoint="${JAEGER_ENDPOINT}" \
    --set secret.data.jwt_secret_key="${JWT_SECRET_KEY}" \
    --set secret.data.log_path="${LOG_PATH}" \
    --set secret.data.manual_register_kernel="${MANUAL_REGISTER_KERNEL}" \
    --set secret.data.memcached_list="${MEMCACHED_LIST}" \
    --set secret.data.mqtt_list_ips="${MQTT_LIST_IPS}" \
    --set secret.admin.username="${ADMIN_USER_NAME}" \
    --set secret.admin.password="${ADMIN_PASSWORD}" \
    --set secret.colabvariable.amz_s3_fuse_mode="${AMZ_S3_FUSE_MODE}" \
    --set secret.colabvariable.api_key="${API_KEY}" \
    --set secret.colabvariable.collaborative_mode="${COLLABORATIVE_MODE}" \
    --set secret.colabvariable.content_path="${CONTENT_PATH}" \
    --set secret.colabvariable.data_path="${DATA_PATH}" \
    --set secret.colabvariable.debug_mode="${DEBUG_MODE}" \
    --set secret.colabvariable.dev_mode="${DEV_MODE}" \
    --set secret.colabvariable.upload_path="${UPLOAD_PATH}" \
    --set secret.colabvariable.secret_key="${SECRET_KEY}" \
    --set secret.colabvariable.service_endpoint="${SERVICE_ENDPOINT}" \
    --set secret.colabvariable.tmp_path="${TMP_PATH}" \
    --set secret.colabvariable.tracing_mode="${TRACING_MODE}" \
    --set secret.pgvariable.pg_database="${PG_DATABASE}" \
    --set secret.pgvariable.pg_host="${PG_HOST}" \
    --set secret.pgvariable.pg_hub_database="${PG_HUB_DATABASE}" \
    --set secret.pgvariable.pg_username="${PG_USERNAME}" \
    --set secret.pgvariable.pg_password="${PG_PASSWORD}" \
    --set secret.redisvariable.redis_list="${REDIS_LIST}" \
    --set secret.redisvariable.redis_password="${REDIS_PASSWORD}" \
    --set secret.redisvariable.use_redis_cache="${USE_REDIS_CACHE}" \
    --set secret.redisvariable.user_path="${USER_PATH}" \
    --set secret.redisvariable.root_path="${ROOT_PATH}" \
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.]

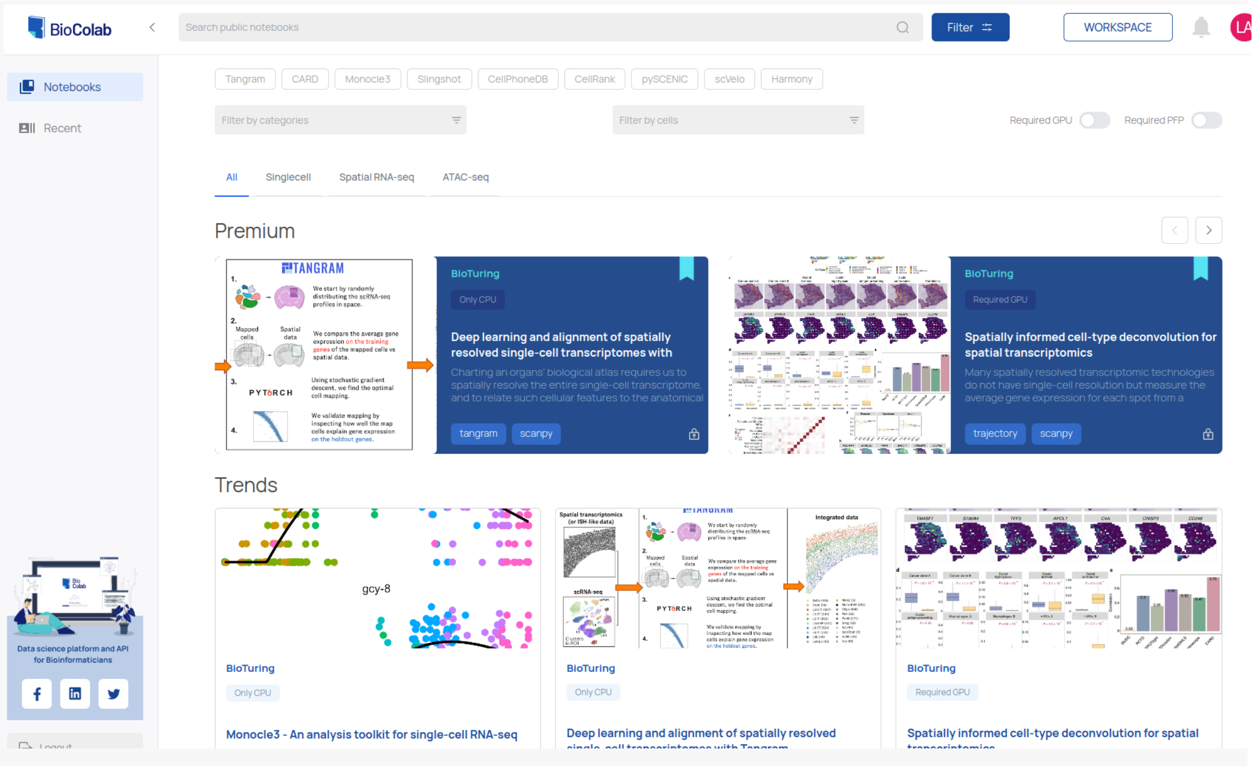
- Issue:

  ```
  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



BioColab

Search public notebooks

Filter

WORKSPACE

LA

Notebooks

Recent

Tangram CARD Monocle3 Slingshot CellPhoneDB CellRank pySCENIC scVelo Harmony

Filter by categories

Filter by cells

Required GPU

Required PFP

All Singlecell Spatial RNA-seq ATAC-seq

Premium

BioTuring

Only CPU

Deep learning and alignment of spatially resolved single-cell transcriptomes with

Charting an organs' biological atlas requires us to spatially resolve the entire single-cell transcriptome, and to relate such cellular features to the anatomical

tangram scanpy

BioTuring

Required GPU

Spatially informed cell-type deconvolution for spatial transcriptomics

Many spatially resolved transcriptomic technologies do not have single-cell resolution but measure the average gene expression for each spot from a

trajectory scanpy

Trends

BioTuring

Only CPU

Monocle3 - An analysis toolkit for single-cell RNA-seq

BioTuring

Only CPU

Deep learning and alignment of spatially resolved single-cell transcriptomes with Tangram

BioTuring

Required GPU

Spatially informed cell-type deconvolution for spatial transcriptomics

Data science platform and API for Bioinformaticians

Logout

TANGRAM
1.
We start by randomly distributing the scRNA-seq profiles in space.
2.
Mapped cells
Spatial data
We compare the average gene expression on the training genes of the mapped cells vs spatial data.
3.
PYTORCH
Using stochastic gradient descent, we find the optimal cell mapping.
4.
We validate mapping by inspecting how well the map cells explain gene expression on the holdout genes.
gcy-8
Spatial transcriptomics (or ISH-like data)
scRNA-seq
Clusters in ROI
Integrated data