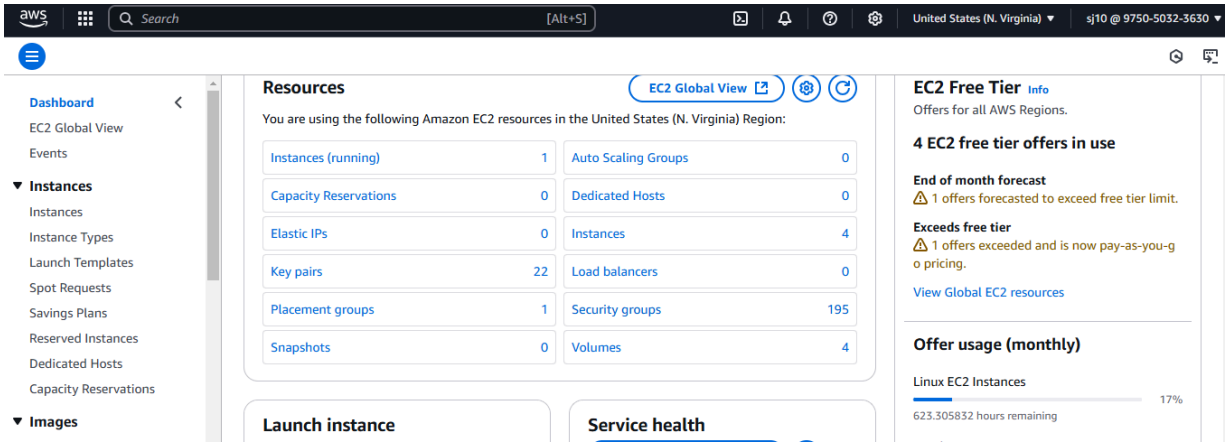
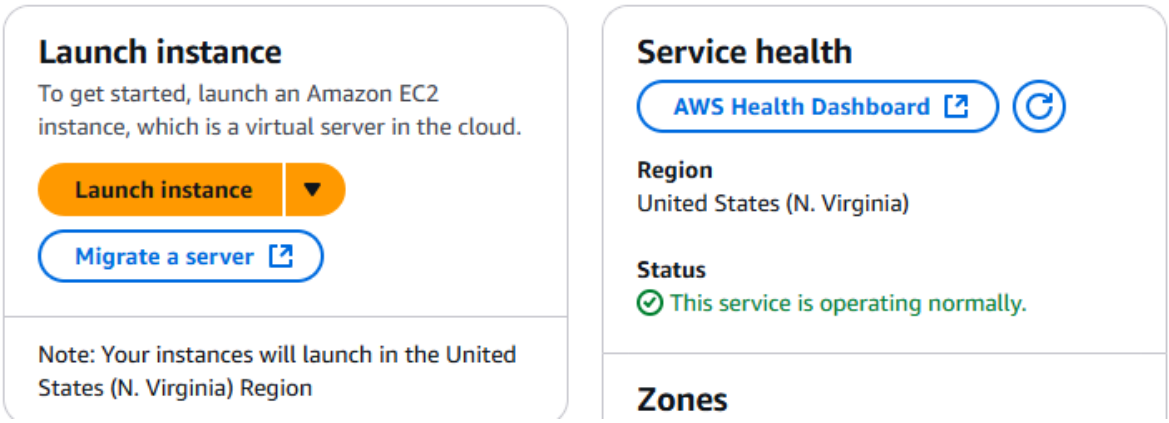
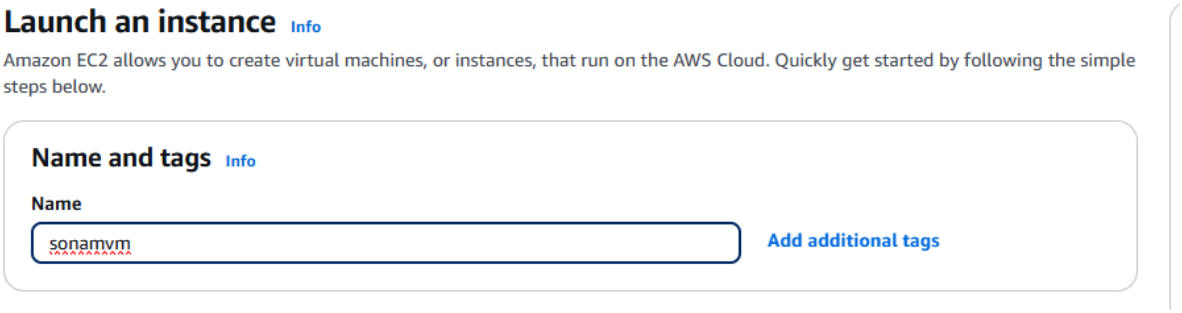
Creating AWS Instance

Login to your AWS Console Account
Search for EC2 and click on EC2 you will be redirected to AWS EC2 Dashboard

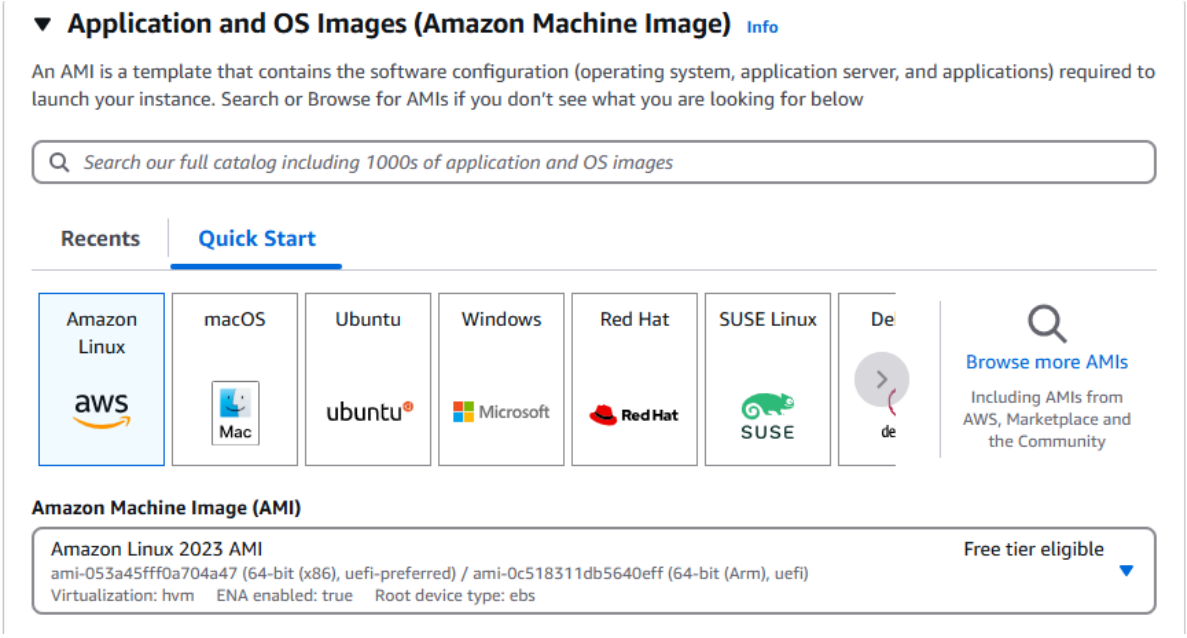


Make sure your Zone must be us-east-1 (N. Virginia).
click on Launch Instance.

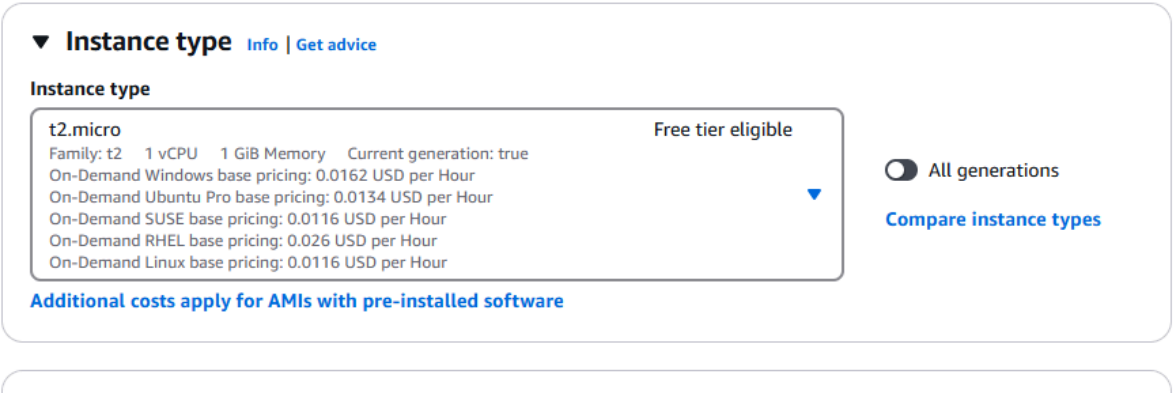




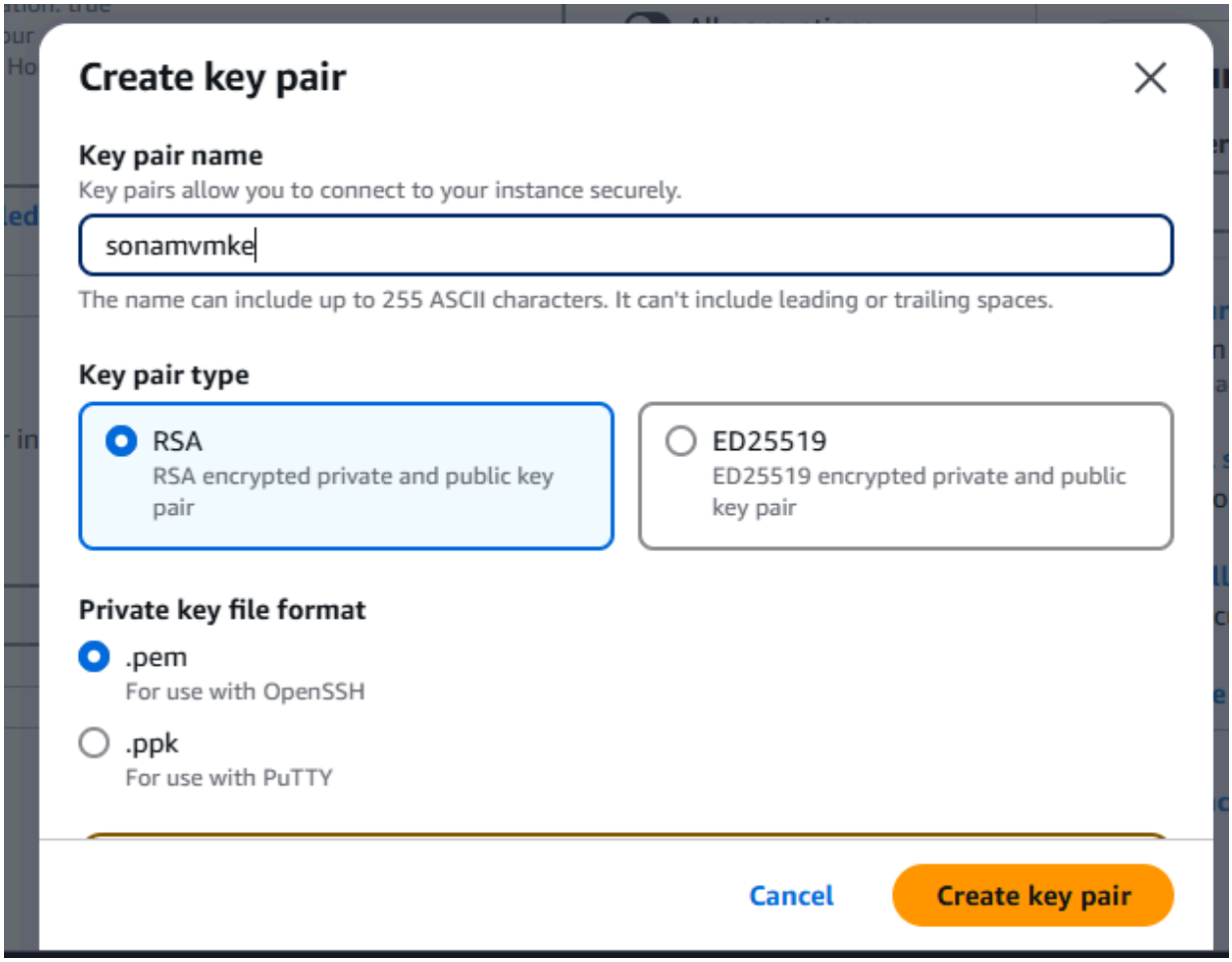
Select Image (AMI) Amazon Linux

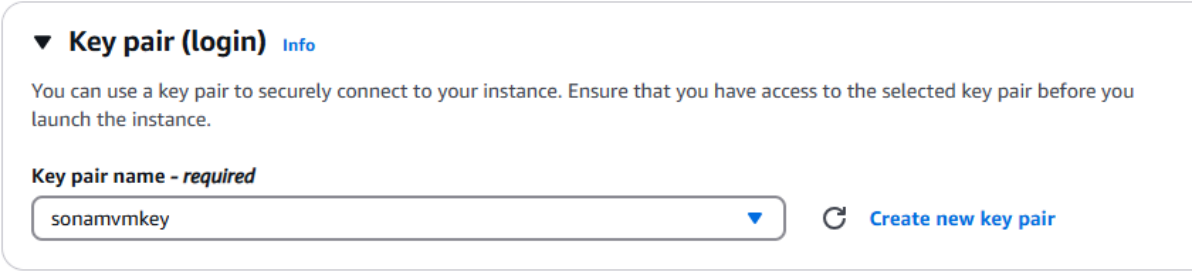


Instance Type: t2 micro

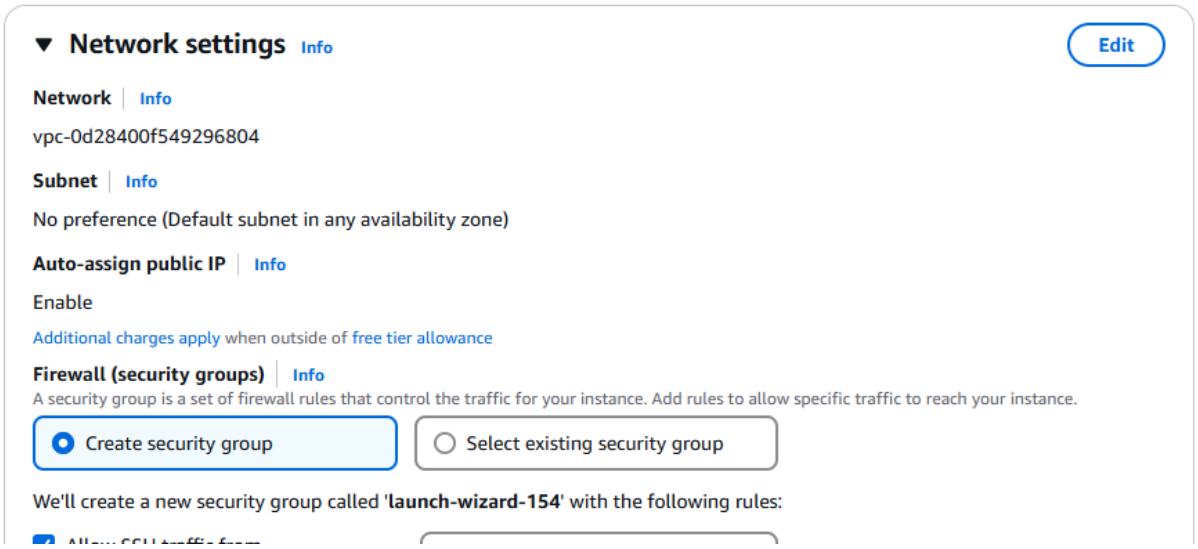


Create key pair:



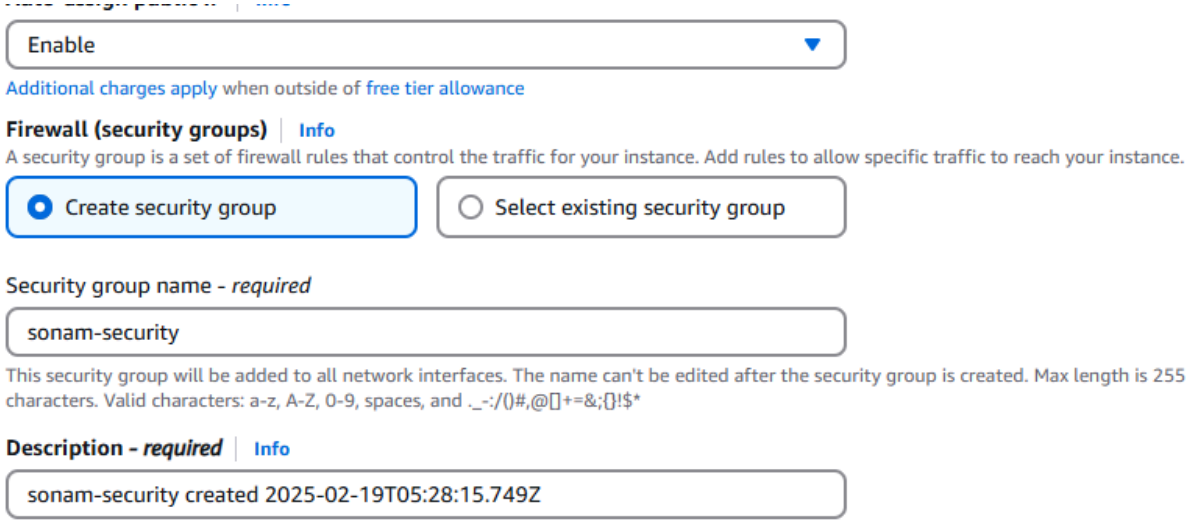


Security Groups to manage the firewall

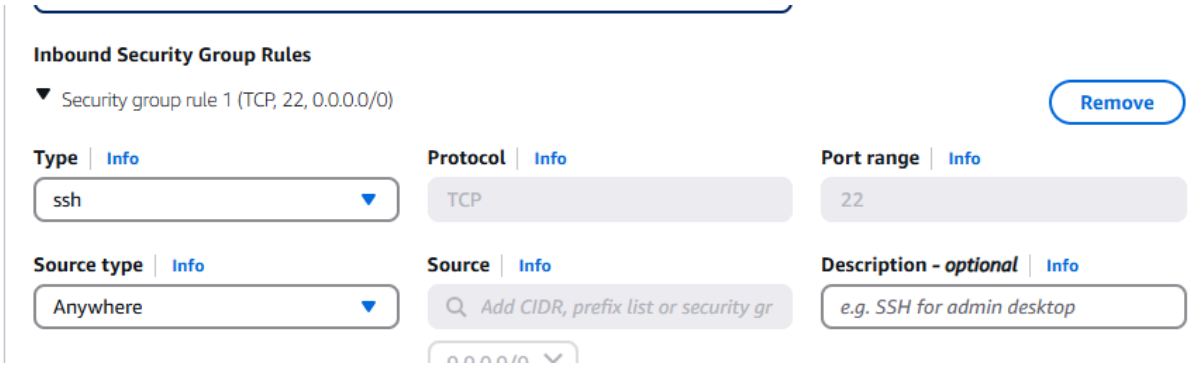


click on Edit.

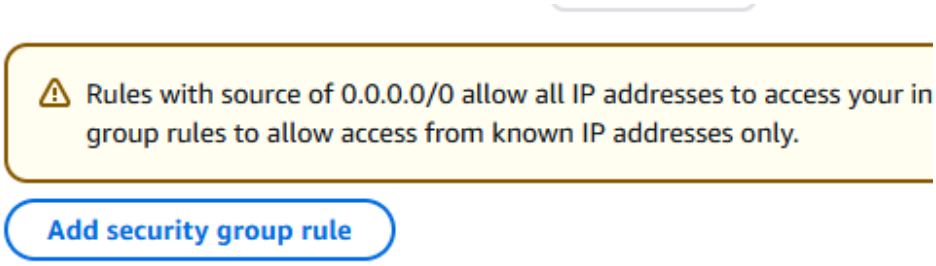
Give name and description of your security Group:



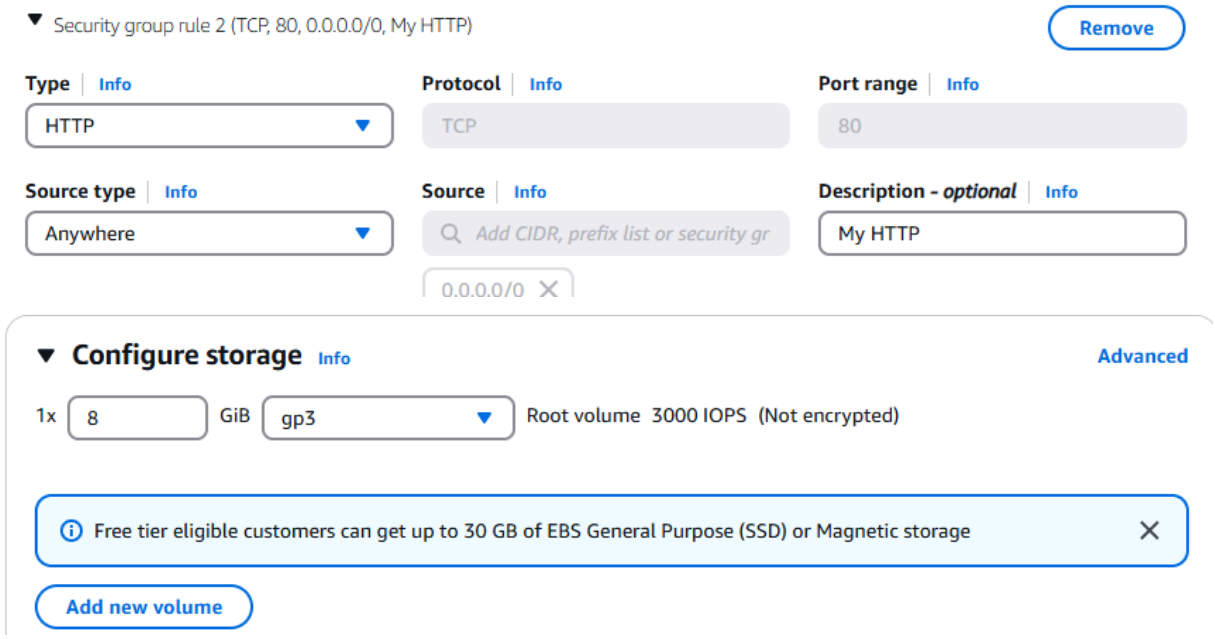
By default security added to use openssh



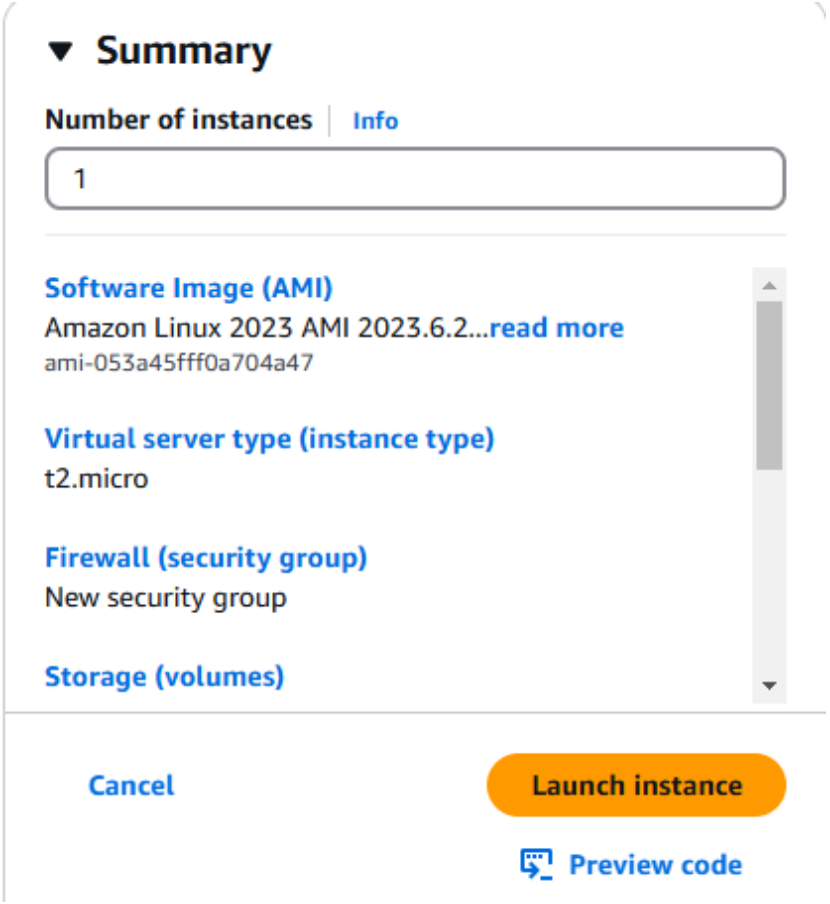
Open one more port http



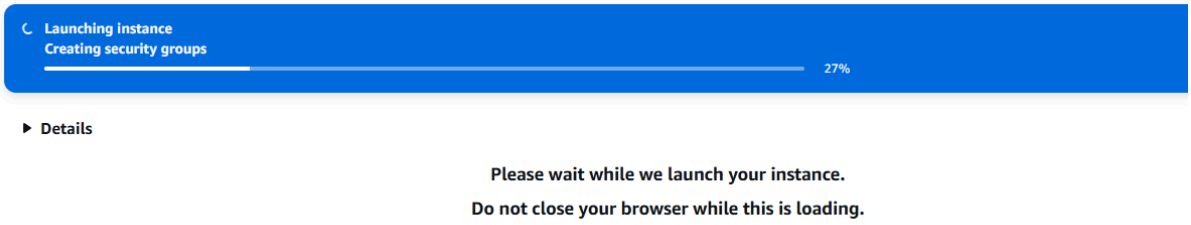
click on add security group rule

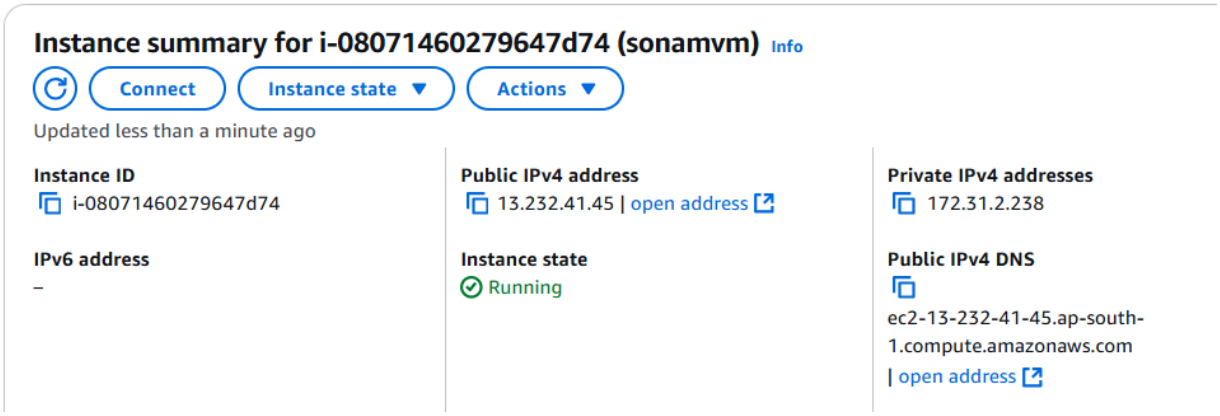


By default it is taking 8 GB for your storage.

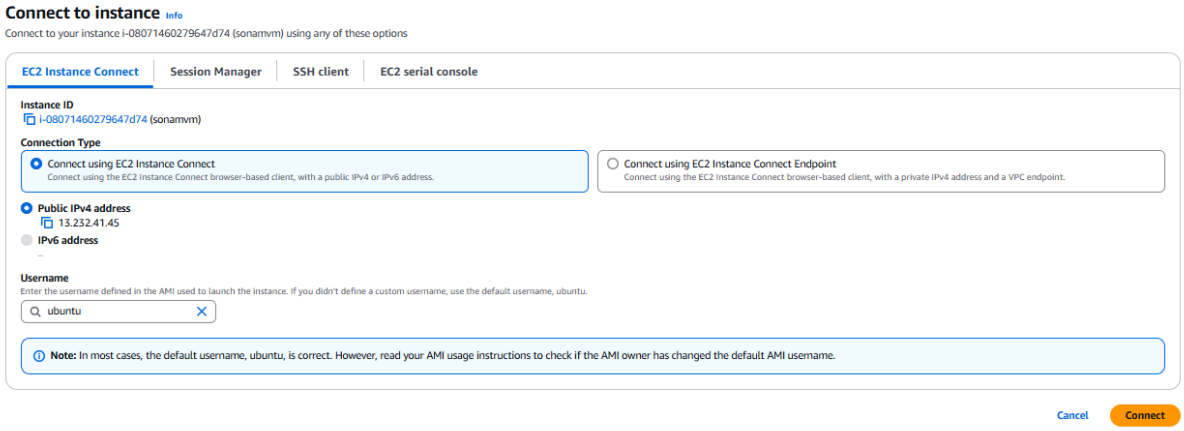


Verify the Summary and Launch Instance.

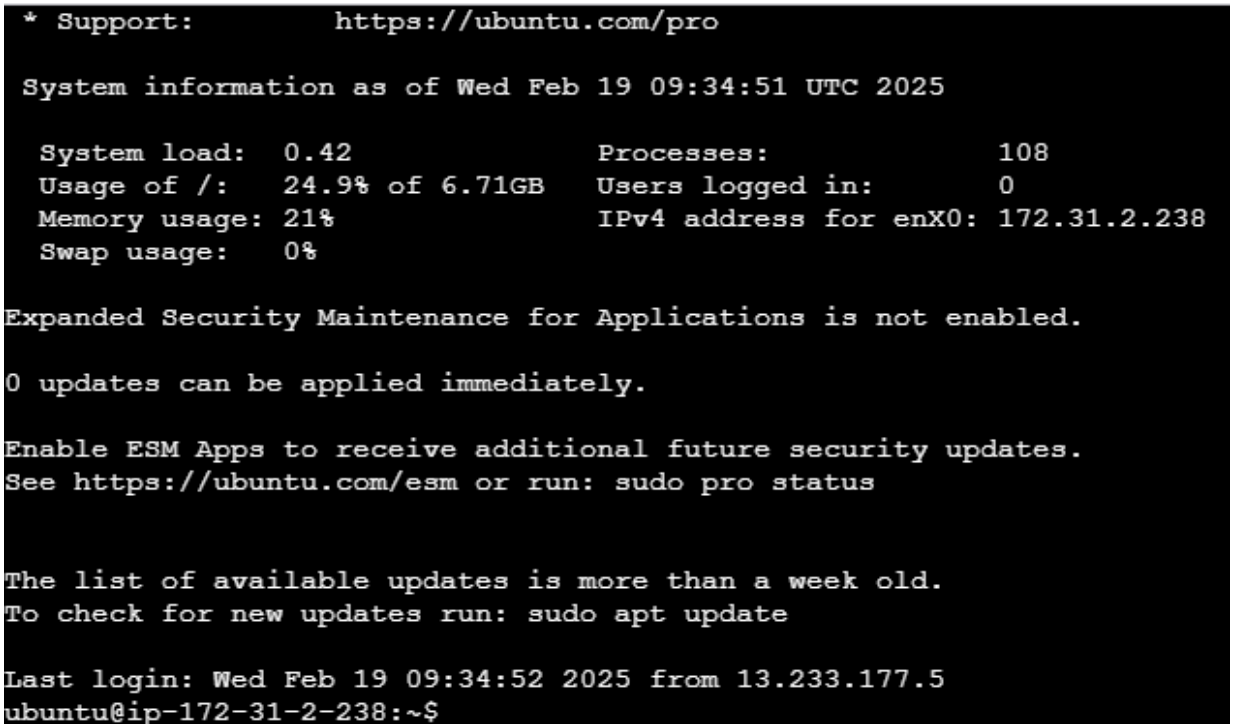




Click on instance Link and click on connect,



Click on Connect and you can access Your instance.



you can run your commands here to work with your instance.

Bucket Policy Generation Steps:

Create Bucket using Some unique name.

General configuration

AWS Region

Asia Pacific (Mumbai) ap-south-1

Bucket type Info

General purpose
Recommended for most use cases and access patterns. General purpose buckets are the original S3 bucket type. They allow a mix of storage classes that redundantly store objects across multiple Availability Zones.

Bucket name Info

sonamsoni123

Bucket name must be unique within the global namespace and follow the bucket naming rules. See rules for bucket

Copy settings from existing bucket - *optional*

Only the bucket settings in the following configuration are copied.

Choose bucket

Format: s3://bucket/prefix

Object Ownership Info

Control ownership of objects written to this bucket from other AWS accounts and the use of access c

ACLs disabled (recommended)
All objects in this bucket are owned by this account. Access to this bucket and its objects is specified using only policies.

Object Ownership

Bucket owner enforced

Block Public Access settings for this bucket

Public access is granted to buckets and objects through access control lists (ACLs), bucket policies, access point policies, c
blocked, turn on Block all public access. These settings apply only to this bucket and its access points. AWS recommends
settings, ensure that your applications will work correctly without public access. If you require some level of public acces
below to suit your specific storage use cases. Learn more

- [x] **Block *all* public access**
  Turning this setting on is the same as turning on all four settings below. Each of the following settings are independent of one anothe
  - [x] **Block public access to buckets and objects granted through *new* access control lists (ACLs)**
    S3 will block public access permissions applied to newly added buckets or objects, and prevent the creation of new public access A
    that allow public access to S3 resources using ACLs.
  - [x] **Block public access to buckets and objects granted through *any* access control lists (ACLs)**
    S3 will ignore all ACLs that grant public access to buckets and objects.
  - [x] **Block public access to buckets and objects granted through *new* public bucket or access point policies**

Bucket Versioning

Versioning is a means of keeping multiple variants of an object in the san
bucket. With versioning, you can easily recover from both unintended use

Bucket Versioning

- ( ) Disable
- (•) Enable

No tags required to add

Default encryption Info

Server-side encryption is automatically applied to new objects stored in this bucket.

Encryption type Info

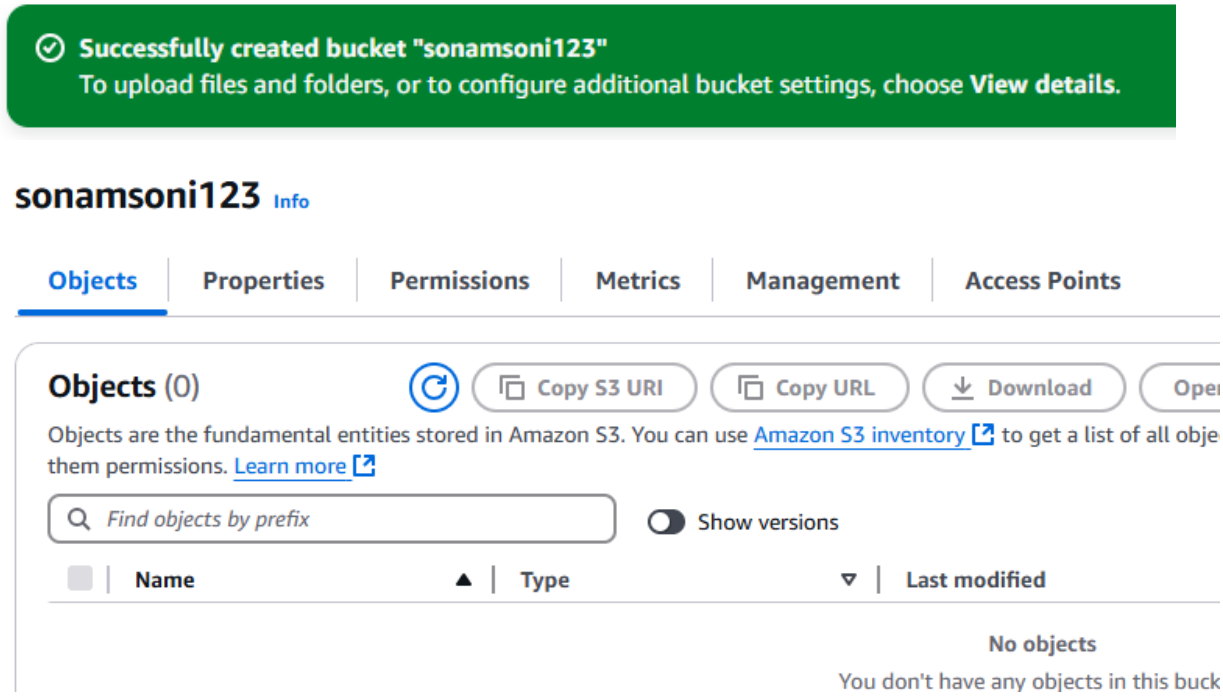
- (•) Server-side encryption with Amazon S3 managed keys (SSE-S3)
- ( ) Server-side encryption with AWS Key Management Service keys (SSE-KMS)
- ( ) Dual-layer server-side encryption with AWS Key Management Service keys (DSSE-KMS)
  Secure your objects with two separate layers of encryption. For details on pricing, see **DSSE-KMS pricing** on

Bucket Key

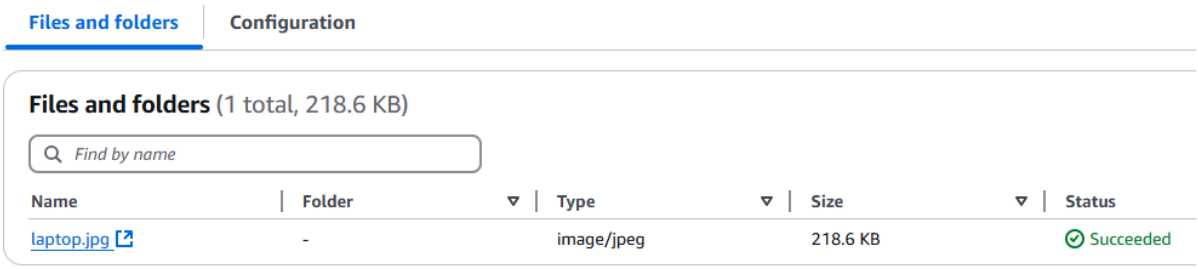
Using an S3 Bucket Key for SSE-KMS reduces encryption costs by lowering calls to AWS KMS. S3 Bucket Keys arer

- ( ) Disable
- (•) Enable

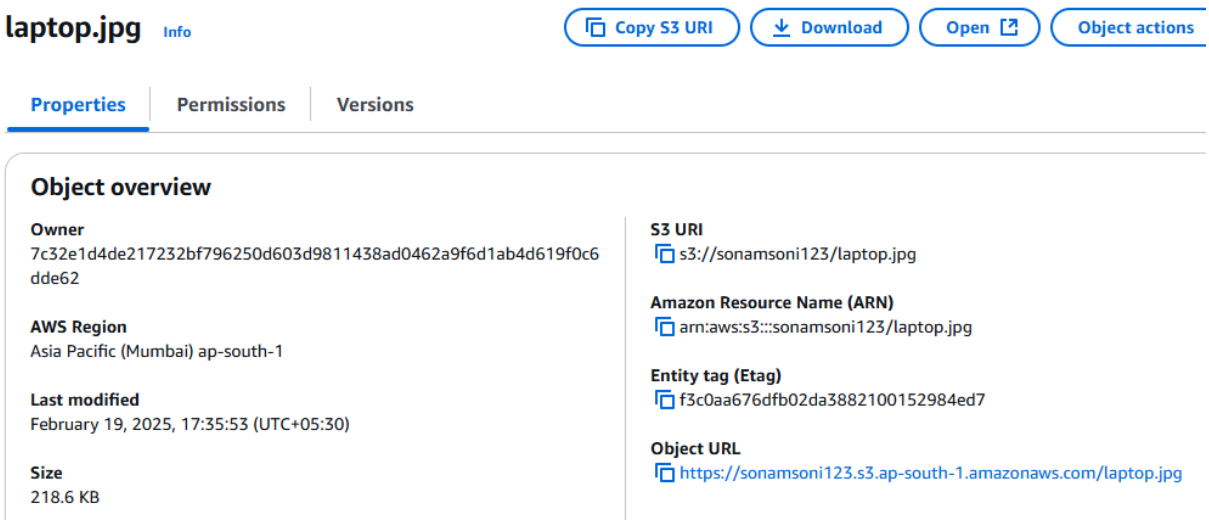
Click on Create Bucket



See the details of your bucket here.
you can add objects inside the bucket.
Click on upload --> select files --> upload

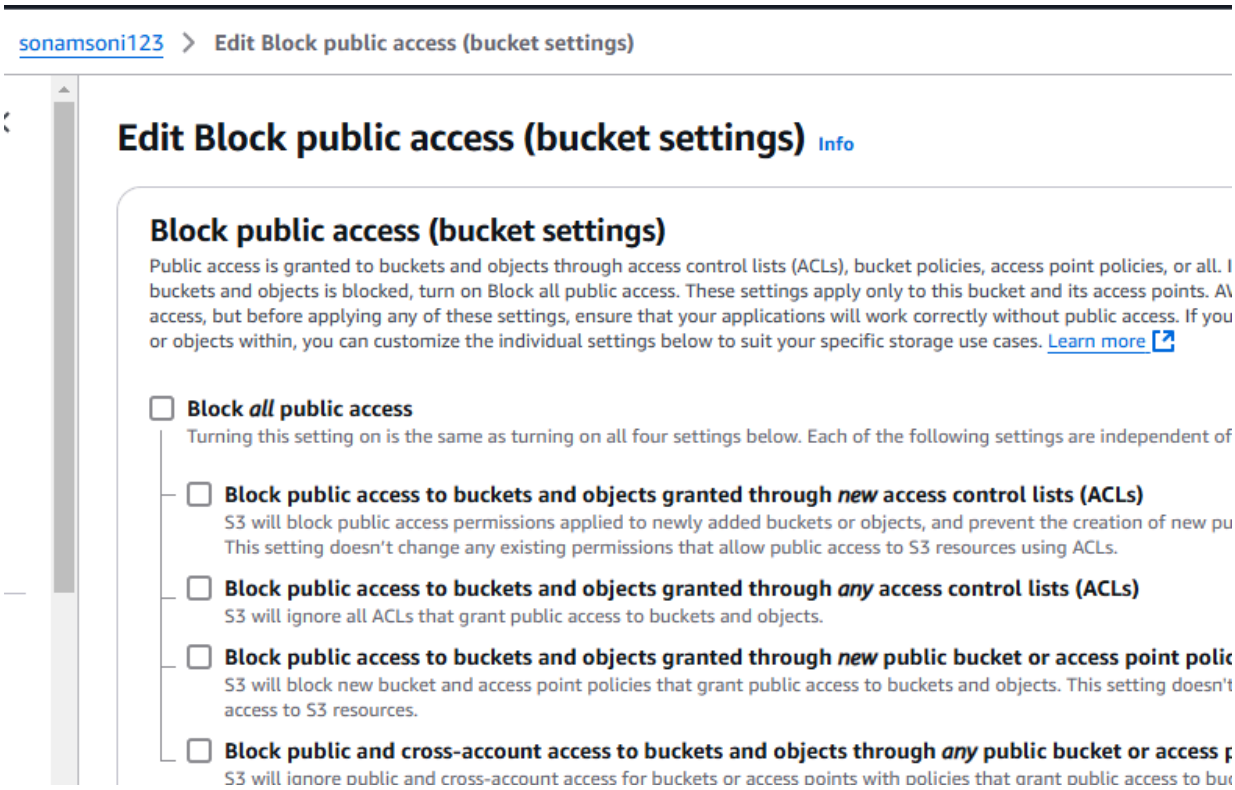


Individual files become a bucket Object.
See the object details by clicking on It



This Object URL is not accessible directly.

To give the access permission. Go to your Bucket permission.



Save the changes.

Now to give access generate Policy by clicking on Policy Generate.

AWS Policy Generator

The AWS Policy Generator is a tool that enables you to create policies that control access to Amazon Web S
For more information about creating policies, see key concepts in Using AWS Identity and Access Managem

Step 1: Select Policy Type

A Policy is a container for permissions. The different types of policies you can create are an IAM Policy, an S
VPC Endpoint Policy, and an SQS Queue Policy.

Select Type of Policy S3 Bucket Policy

Step 2: Add Statement(s)

A statement is the formal description of a single permission. See a description of elements that you can us

Effect Allow Deny

Principal *

Use a comma to separate multiple values.

AWS Service Amazon S3

Use multiple statements to add permissions for more than one service.

Use multiple statements to add permissions for more than one servi

Actions 1 Action(s) Selected All A

ource Name (ARN)

- [ ] GetMultiRegionAccessPointPolicyStatus
- [ ] GetMultiRegionAccessPointRoutes
- [x] GetObject
- [ ] GetObjectAcl

{BucketNam

Click on Add Statement:

You added the following statements. Click the button below to Generate a policy.

Principal(s)	Effect	Action	Resource	Conditions
• *	Allow	• s3:GetObject	arn:aws:s3:::sonamsoni123/*	*None*

Generate Policy:

Changes made below will **not be reflected in the policy generator tool.**

```
{
  "Id": "Policy1739967011716",
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "Stmt1739966293071",
      "Action": [
        "s3:GetObject"
      ],
      "Effect": "Allow",
      "Resource": "arn:aws:s3:::sonamsoni123/*",
      "Principal": "*"
    }
  ]
}
```

Copy and paste in your Policy:

Bucket ARN

arn:aws:s3:::sonamsoni123

Policy

```
1  {
2    "Id": "Policy1739967011716",
3    "Version": "2012-10-17",
4    "Statement": [
5      {
6        "Sid": "Stmt1739966293071",
7        "Action": [
8          "s3:GetObject"
9        ],
10       "Effect": "Allow",
11       "Resource": "arn:aws:s3:::sonamsoni123/*",
12       "Principal": "*"
13     }
14   ]
15 }
```

Save Changes.

Then you can Try to access Objects of your buckets.