GEORGIA INSTITUTE OF TECHNOLOGY SCHOOL OF ELECTRICAL AND COMPUTER ENGINEERING

ECE 4150-A Fall 2024

Lab: Serverless Photo Gallery Application using AWS Lambda, API Gateway, S3, DynamoDB and Cognito

---

References:

- A. Bahga, V. Madisetti, “Cloud Computing Solutions Architect: A Hands-On Approach”, ISBN: 978-0996025591
- https://aws.amazon.com/documentation/

Due Date:

The lab report will be **due on 11:59PM, Feb 16, 2024 .**

In this lab, we will create a Photo Gallery application composed of albums and photos using two variations to store records of photos: SQL and NoSQL. We are going to integrate the same method we used in the previous lab.

In the SQL variant of the application, the records of albums and photos are maintained in a MySQL database instance on Amazon RDS. Whereas in the NoSQL variant of the application, the records of photos are held in a DynamoDB table. This application is implemented in Python and uses the Flask web framework. We are going to deploy the application on an Amazon EC2 instance.

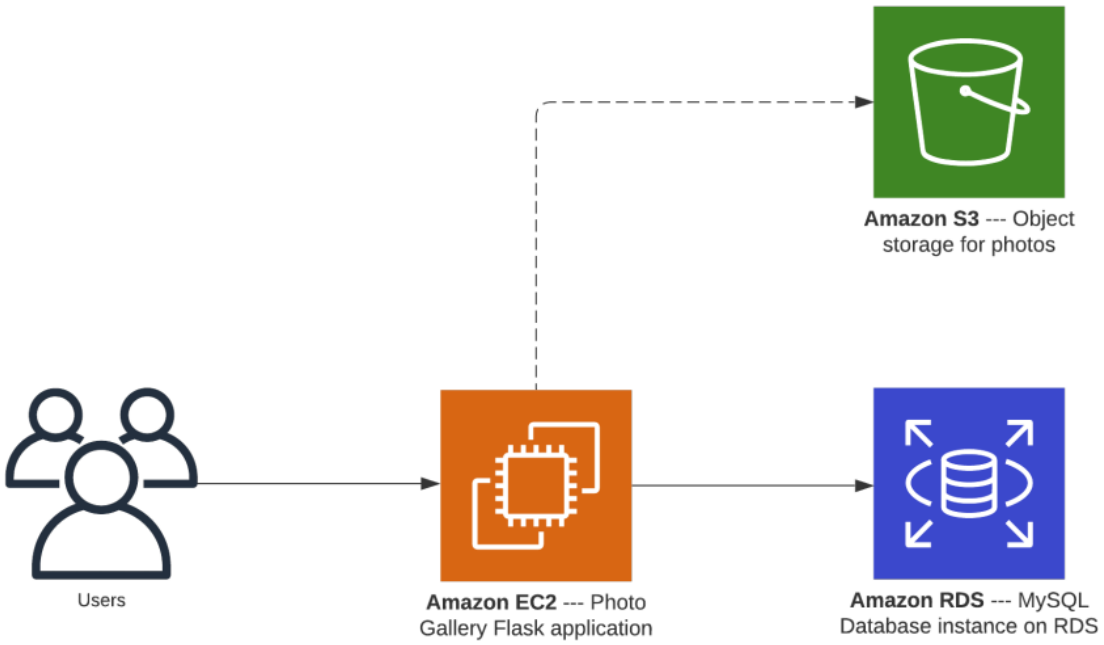


Fig.1 Architecture diagram of the Photo Gallery application - SQL Variant

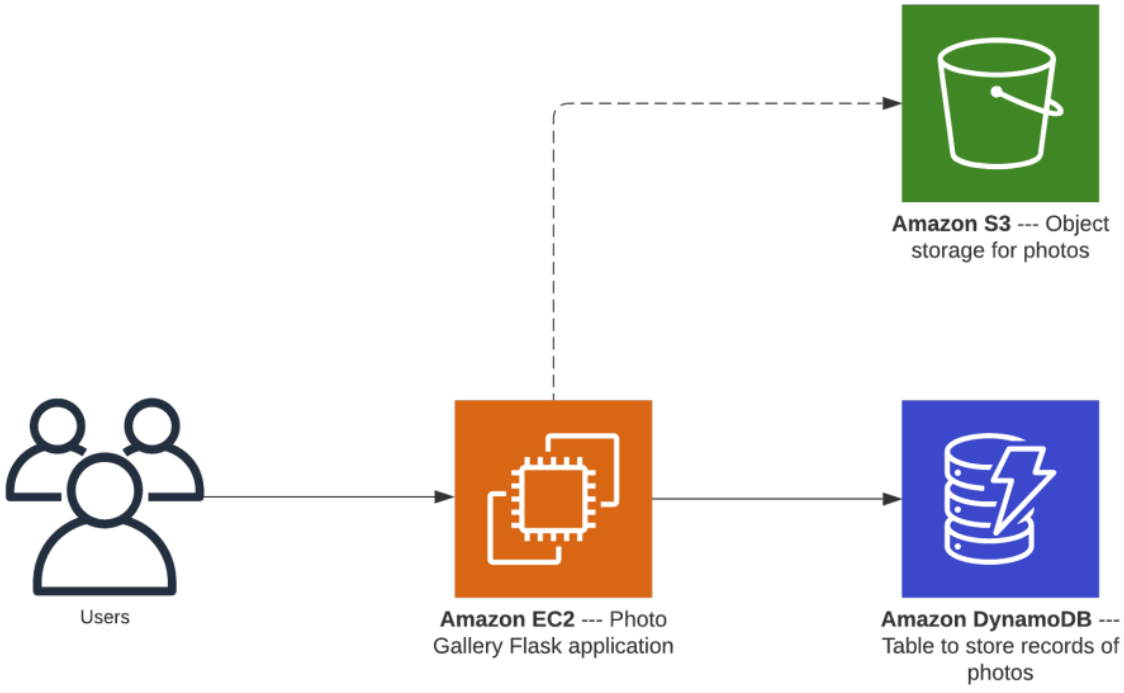
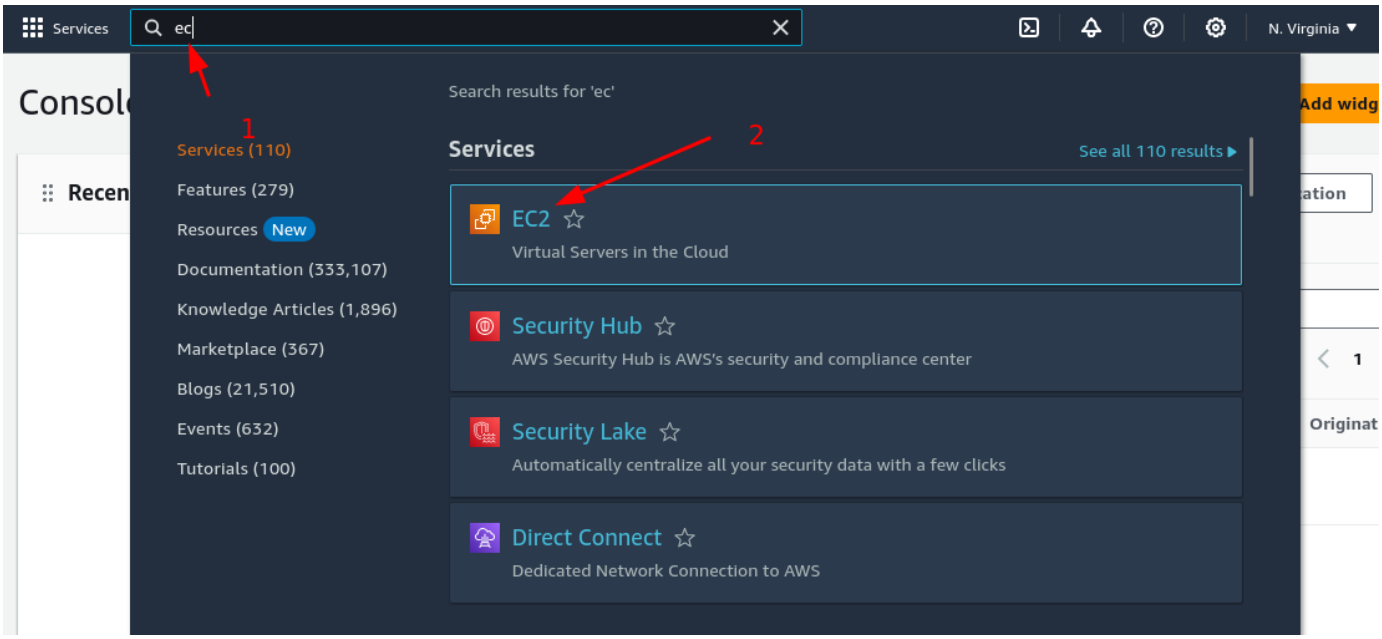


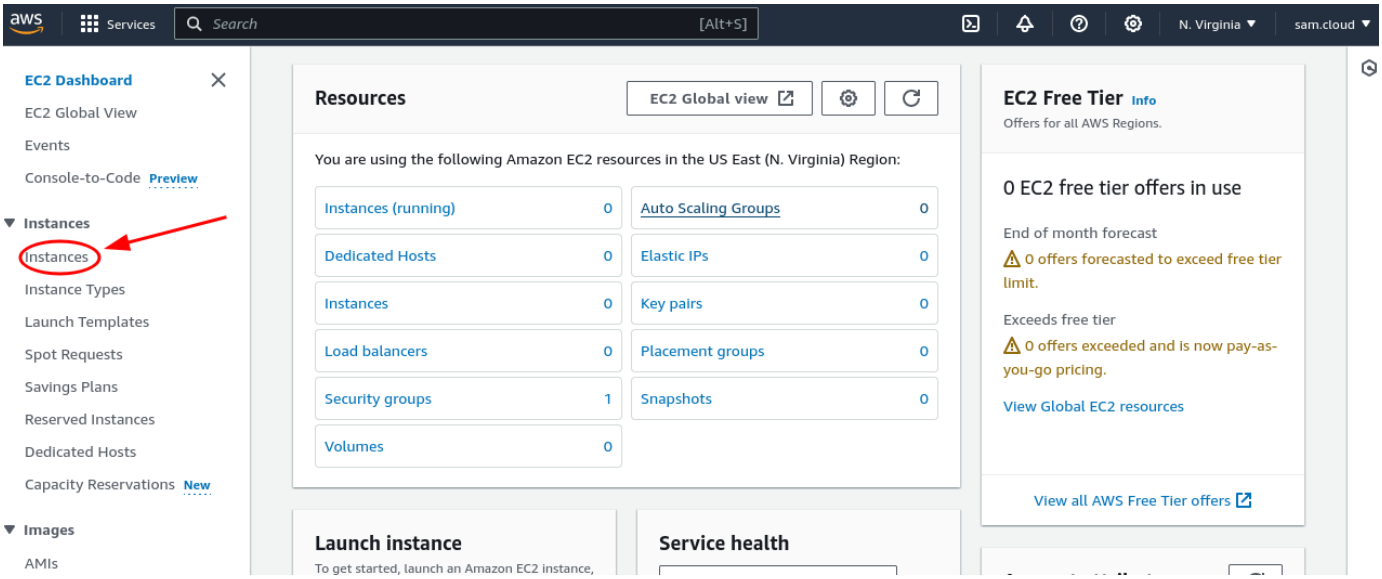
Fig. 2 Architecture diagram of the Photo Gallery application - SQL Variant

1. Create EC2 instance for hosting the static website and resources (10 points)**

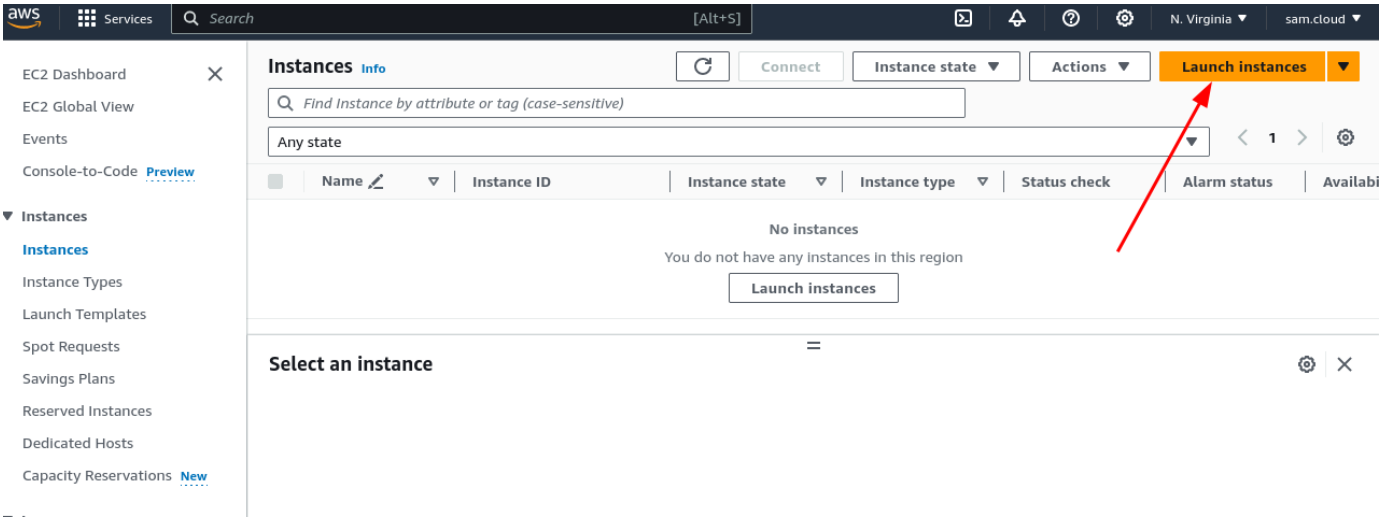
- In the AWS Management Console, under Compute, click EC2. You can also use the search input in the navigation bar by typing “EC2”



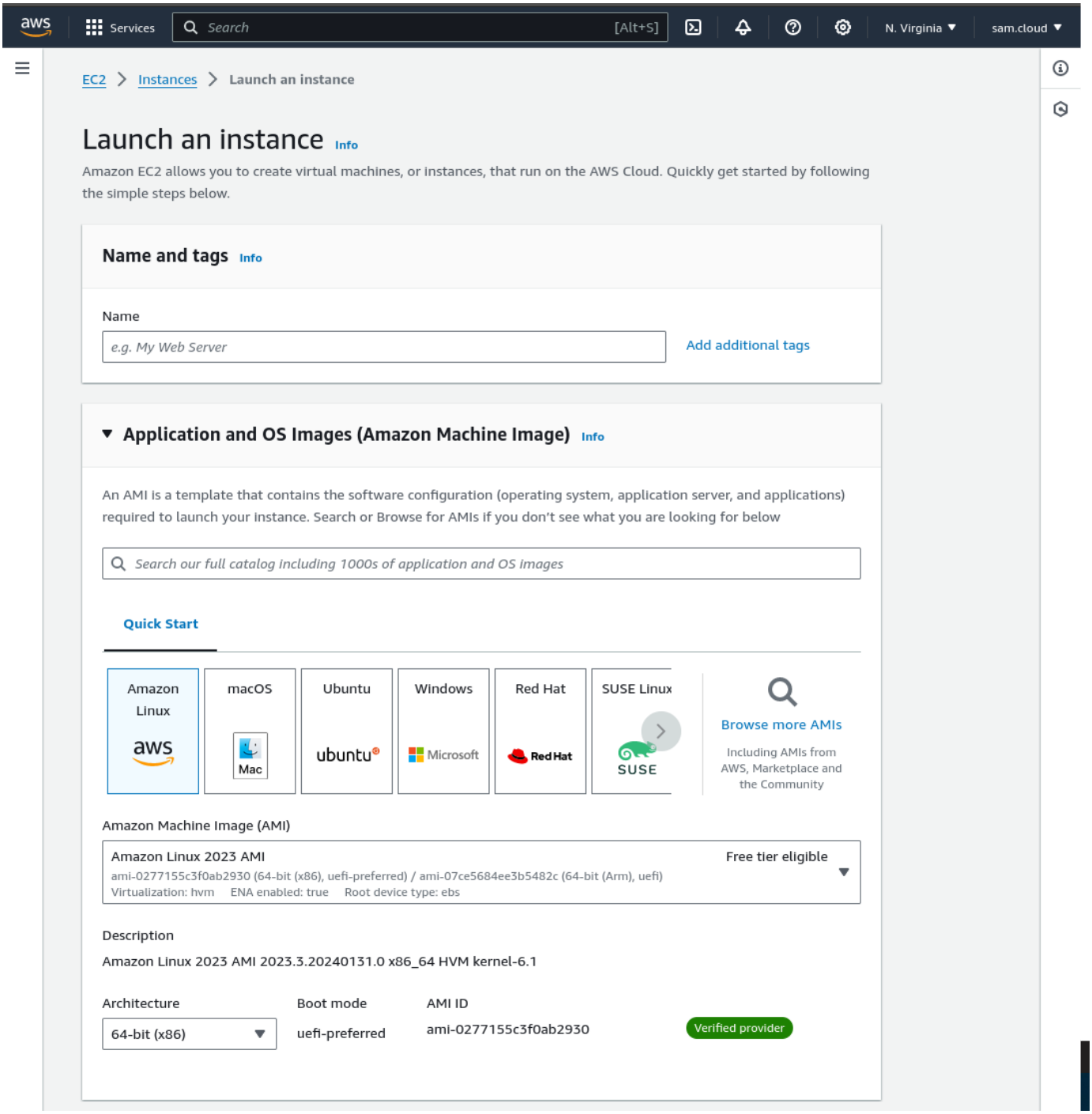
- Once in the EC2 console, click Instances under the instances dropdown menu in the navigation bar on the left

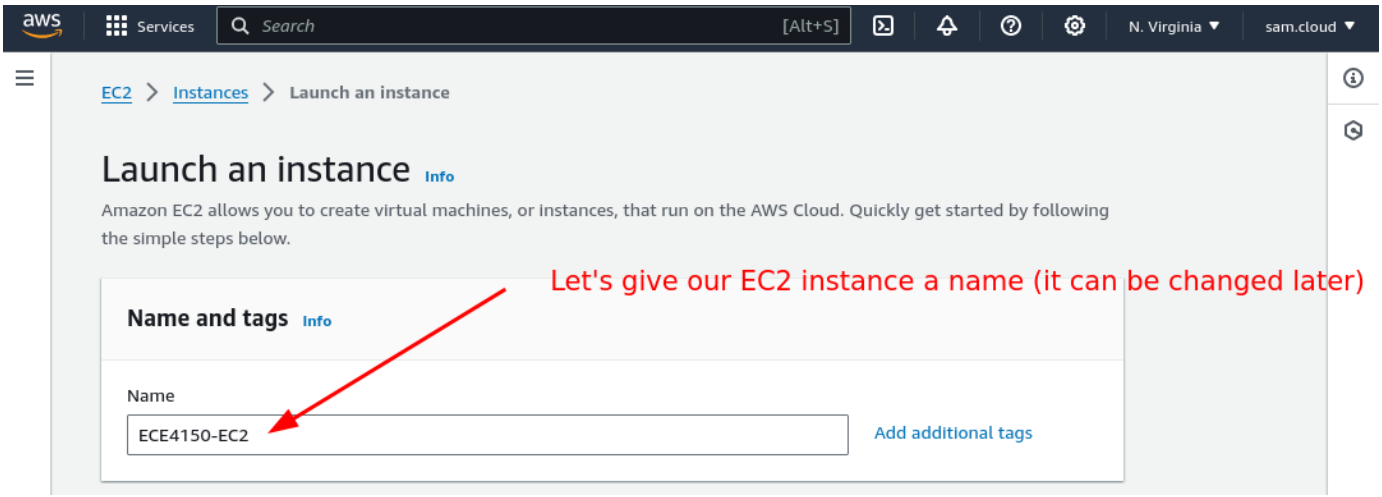


- Lunch an instance by clicking the orange button “Launch instances”



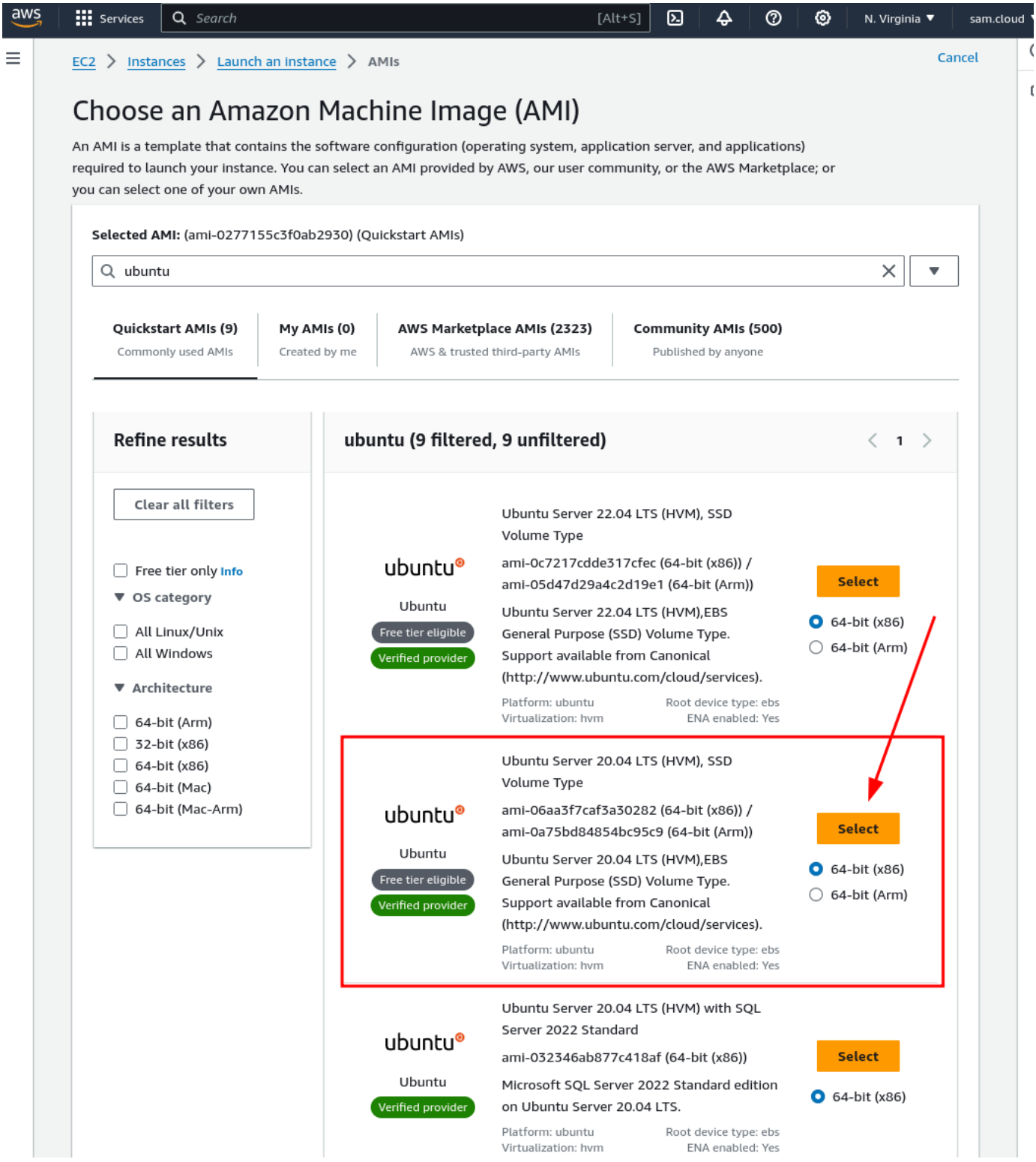
- In the rst step to launching an instance, you have the option to choose the **Amazon Machine Image** (AMI) that you would like to use for your server instance. Go through the list to get familiar with the images that AWS offers. For our lab, we are going to use an Ubuntu image to run our ask server



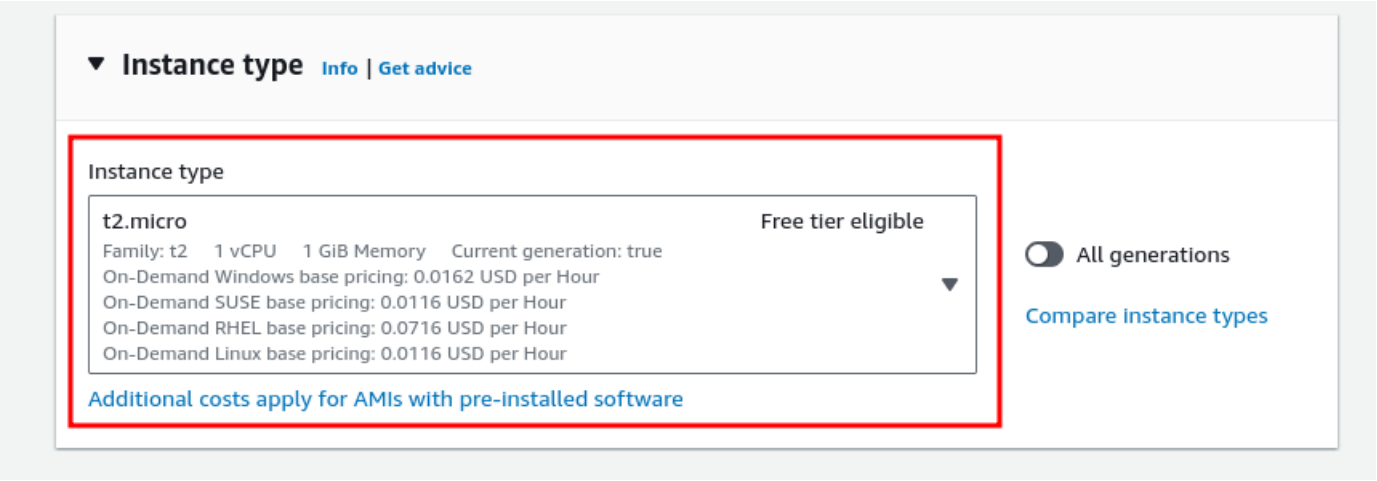


- To fnd the right Ubuntu image, type “**Ubuntu**” in the search input and nd the most recent Ubuntu version. In this given time, the most recent snapshot is **20.04**

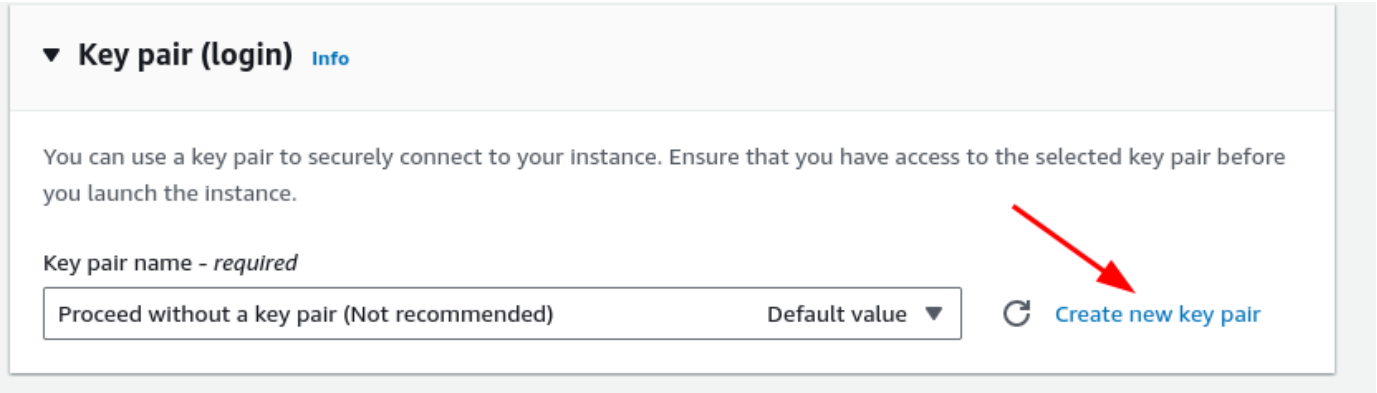
- Select the **Ubuntu Server 20.04 LTS (HVM), SSD Volume Type** image

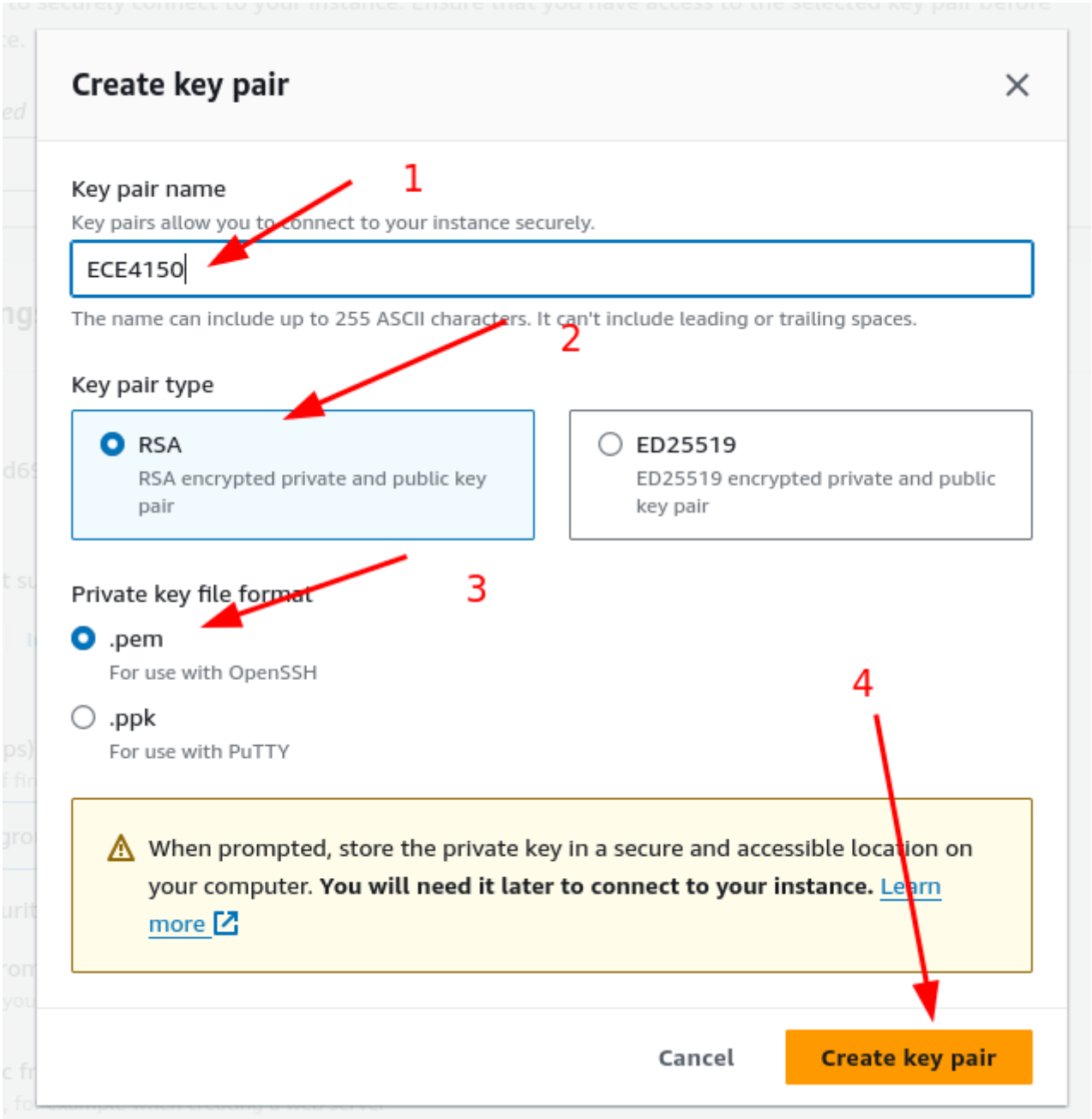


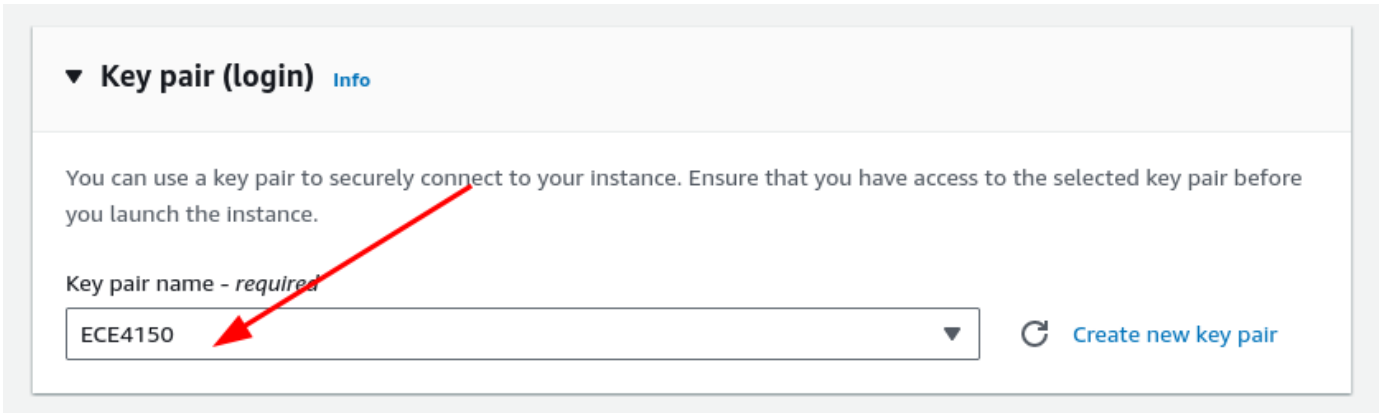
- In the second step, we can select the type of instance that we would like to use. For our lab, we will choose the **t2.micro** since it is the free tier eligible version



- To access an instance, AWS gives you a private key that you can use to access the instance remotely. You will have the option to create a new key pair or attach it to an existing key pair To access your instance, you will use the SSH protocol to connect via the terminal or PowerShell
- To create a new one, select "**Create a new key pair**" and name the key pair (**ECE4150**). Once you placed the key's name, download the key by clicking the "**Download Key Pair**" button or **it will automatically download a copy for you**.







Ensure you store that key in a secure place because you won't have access to the instance again if you lose it

Network settings Info

Edit

Network Info

vpc-01ba9790c80584d69

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:

[x] Allow SSH traffic from
Helps you connect to your instance

Anywhere
0.0.0.0/0

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

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

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

Configure storage Info

Advanced

1x 8 GiB gp2 Root volume (Not encrypted)

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

Add new volume

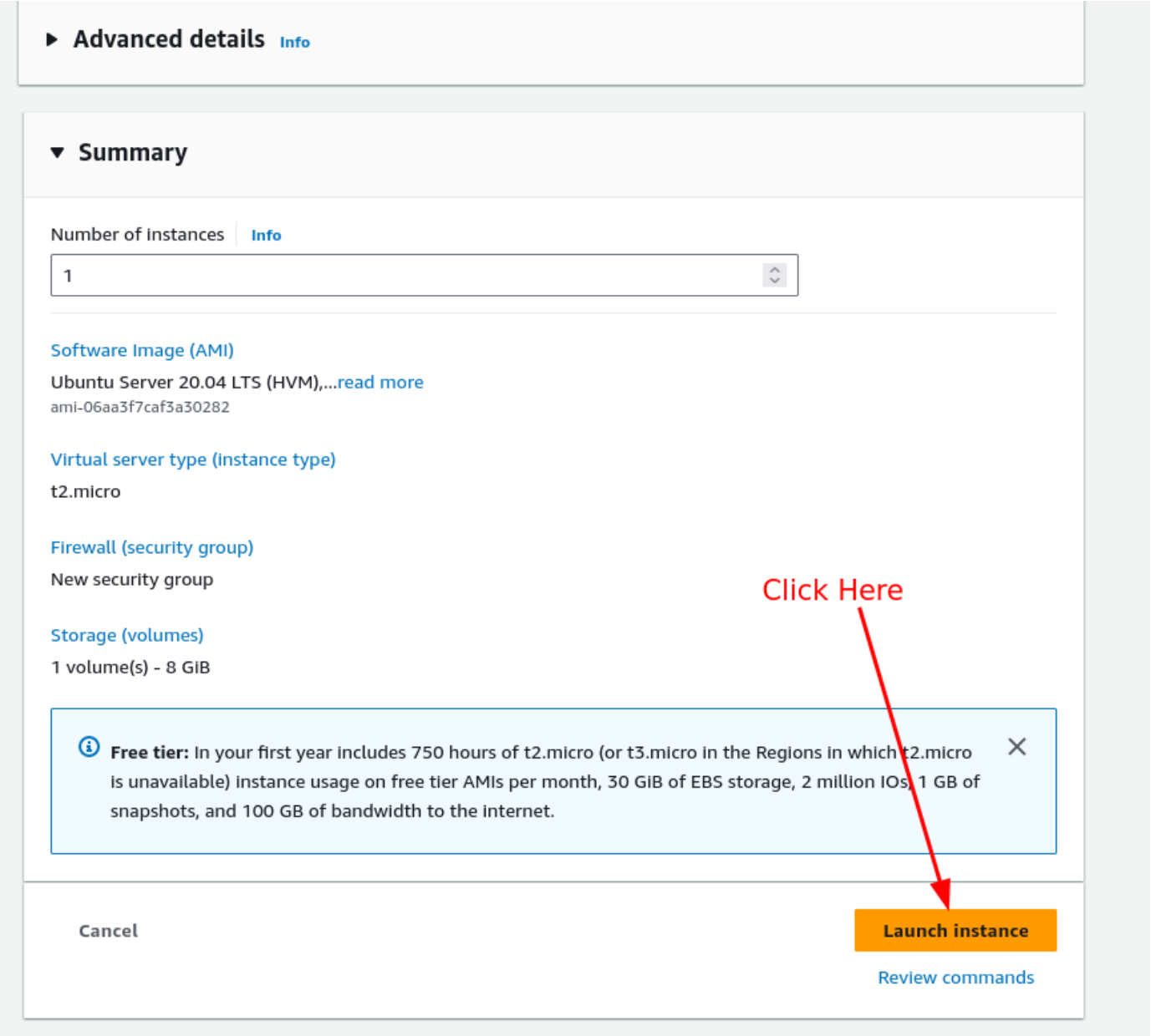
The selected AMI contains more instance store volumes than the instance allows. Only the first 0 instance store volumes from the AMI will be accessible from the instance

Click refresh to view backup information

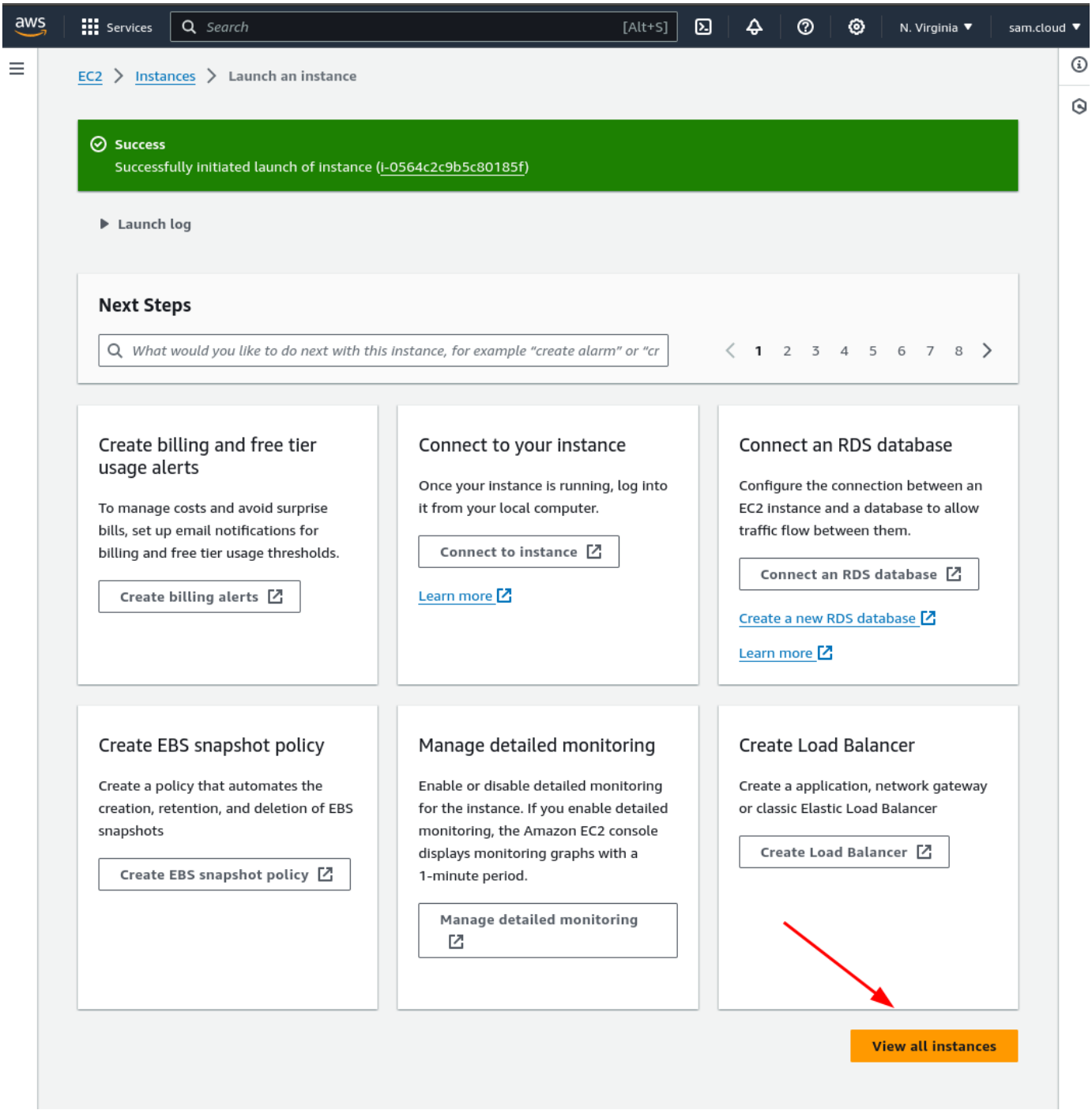
The tags that you assign determine whether the instance will be backed up by any Data Lifecycle Manager policies.

0 x File systems Edit

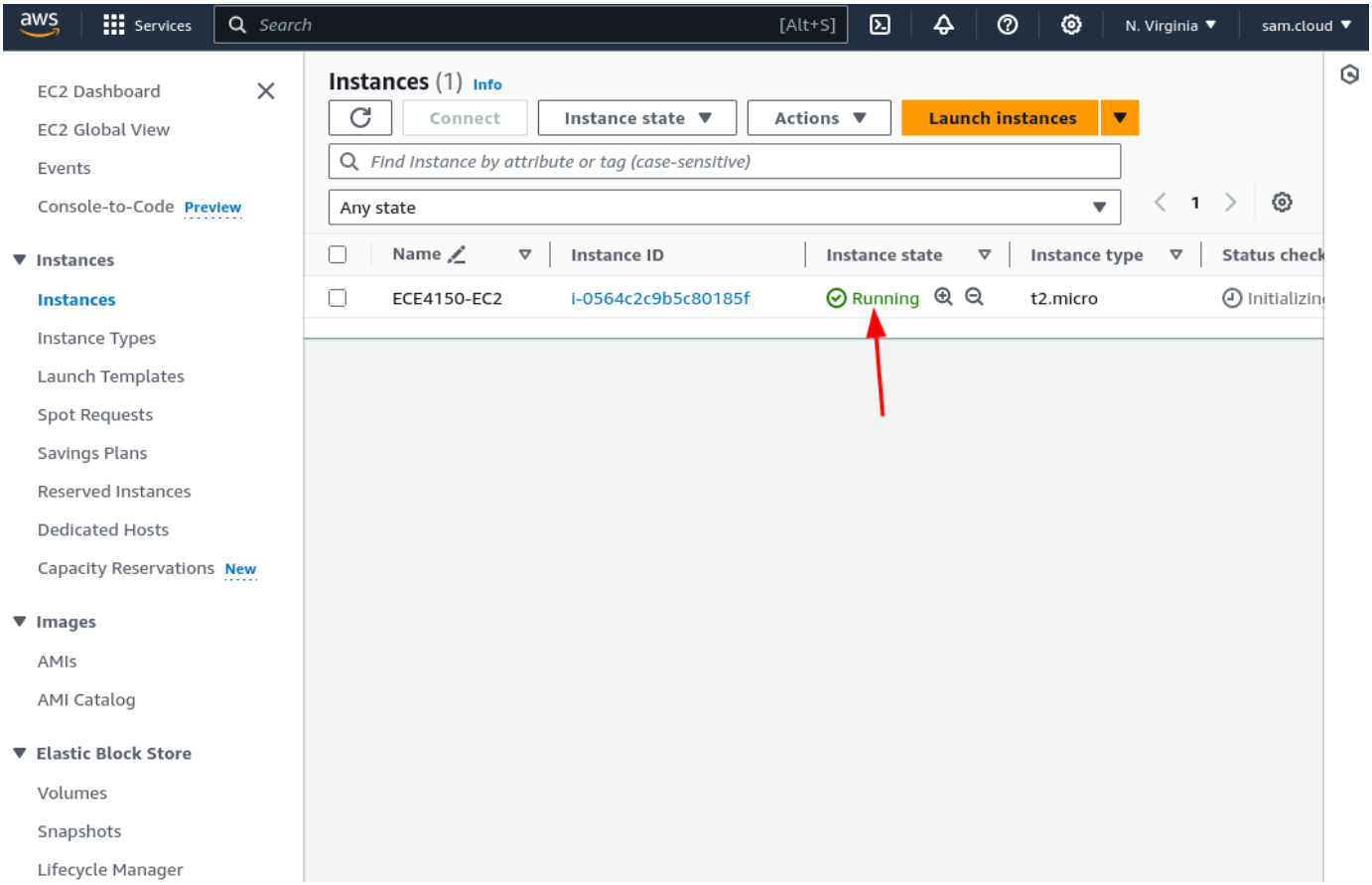
Advanced details Info



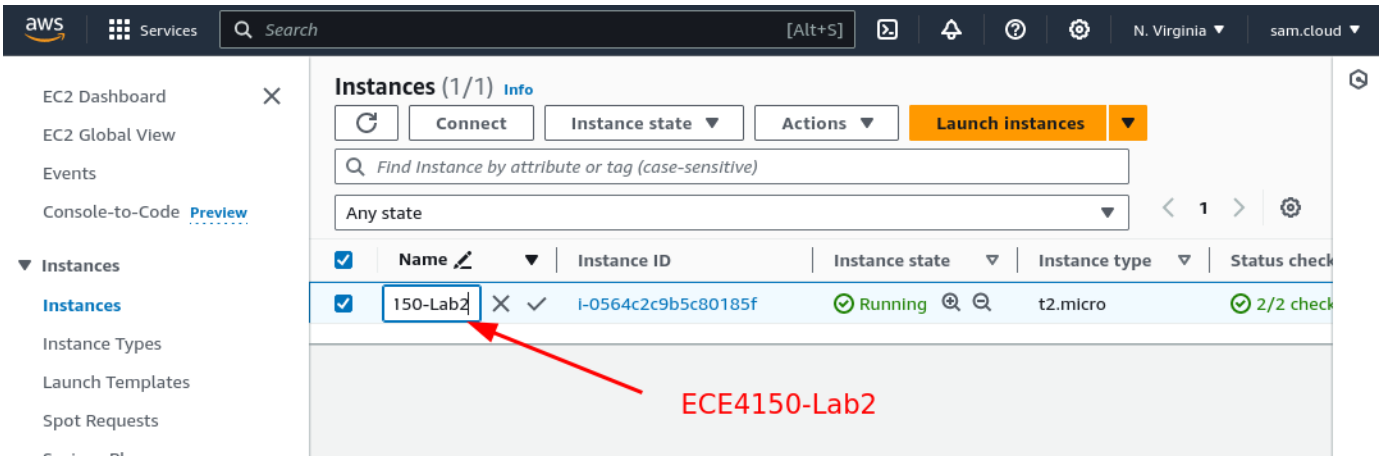
- Click “**Launch instance**”
- It might take a few seconds or minutes for the server instance to be available. However, you can view the instance status by clicking the “**View Instances**” button



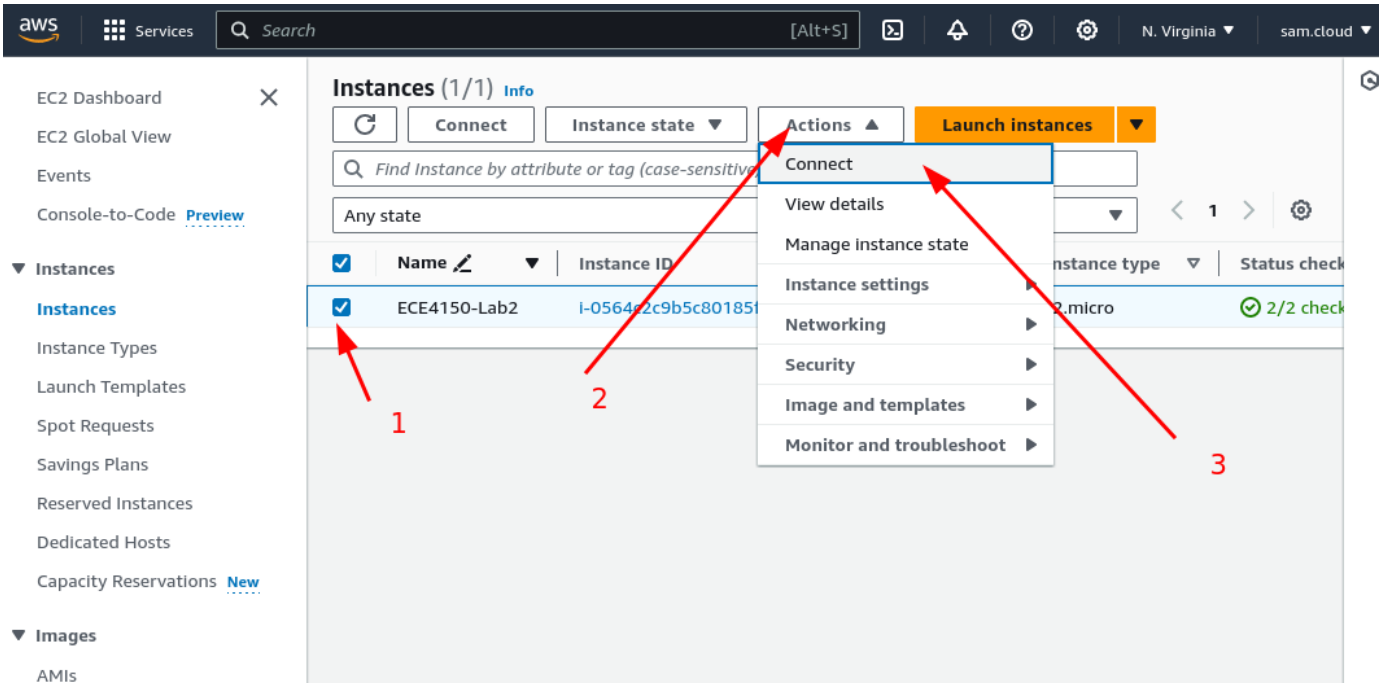
- You should see your instance running, as shown below (refresh your page if don't see anything after 3-5 minutes later)



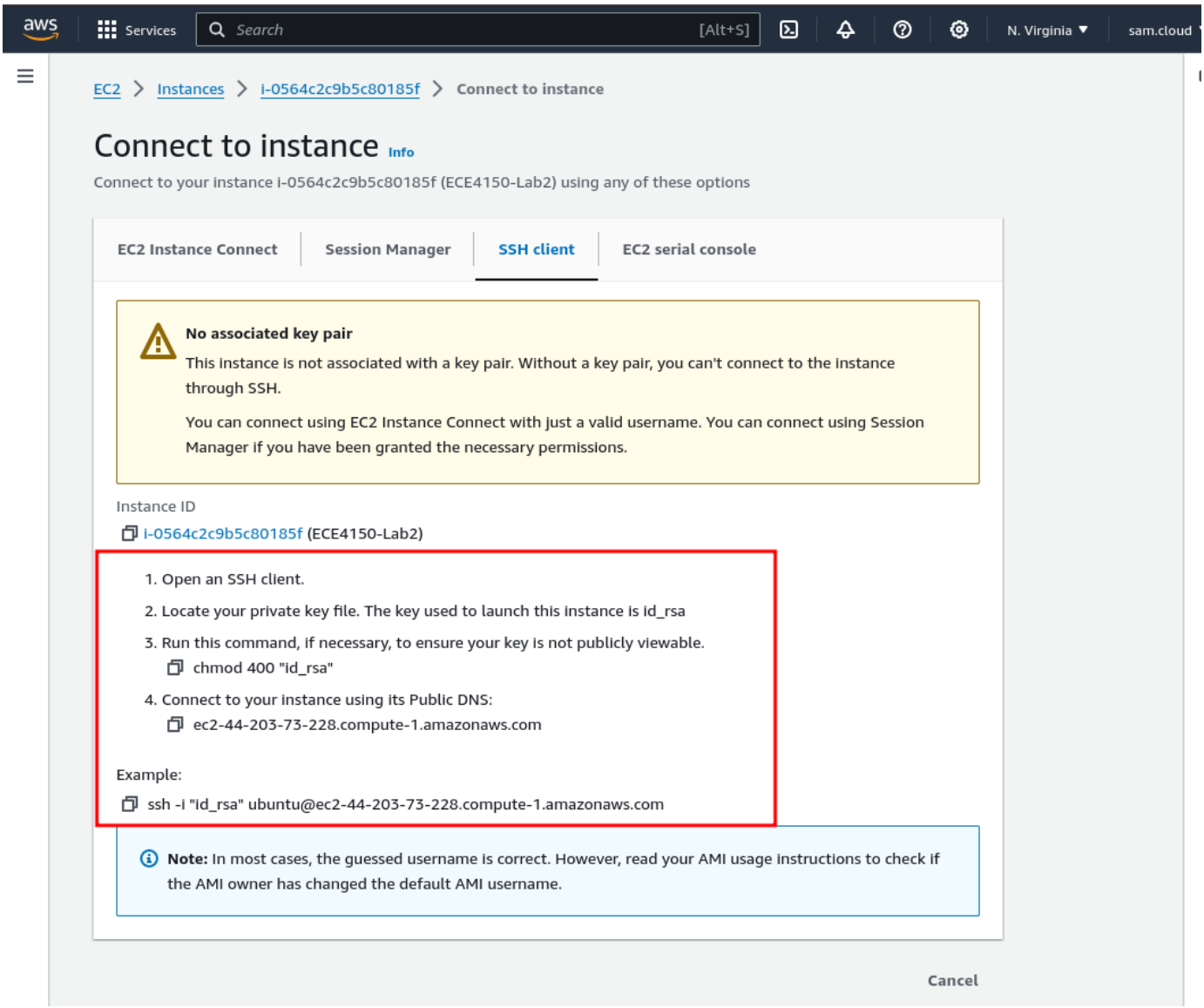
- If you want to change the name of the instance, click in the section below the **Name** column (renamed it to ECE4150-Lab2)



- To connect to your new server instance, select your instance, click the “**Actions**” dropdown, and click connect

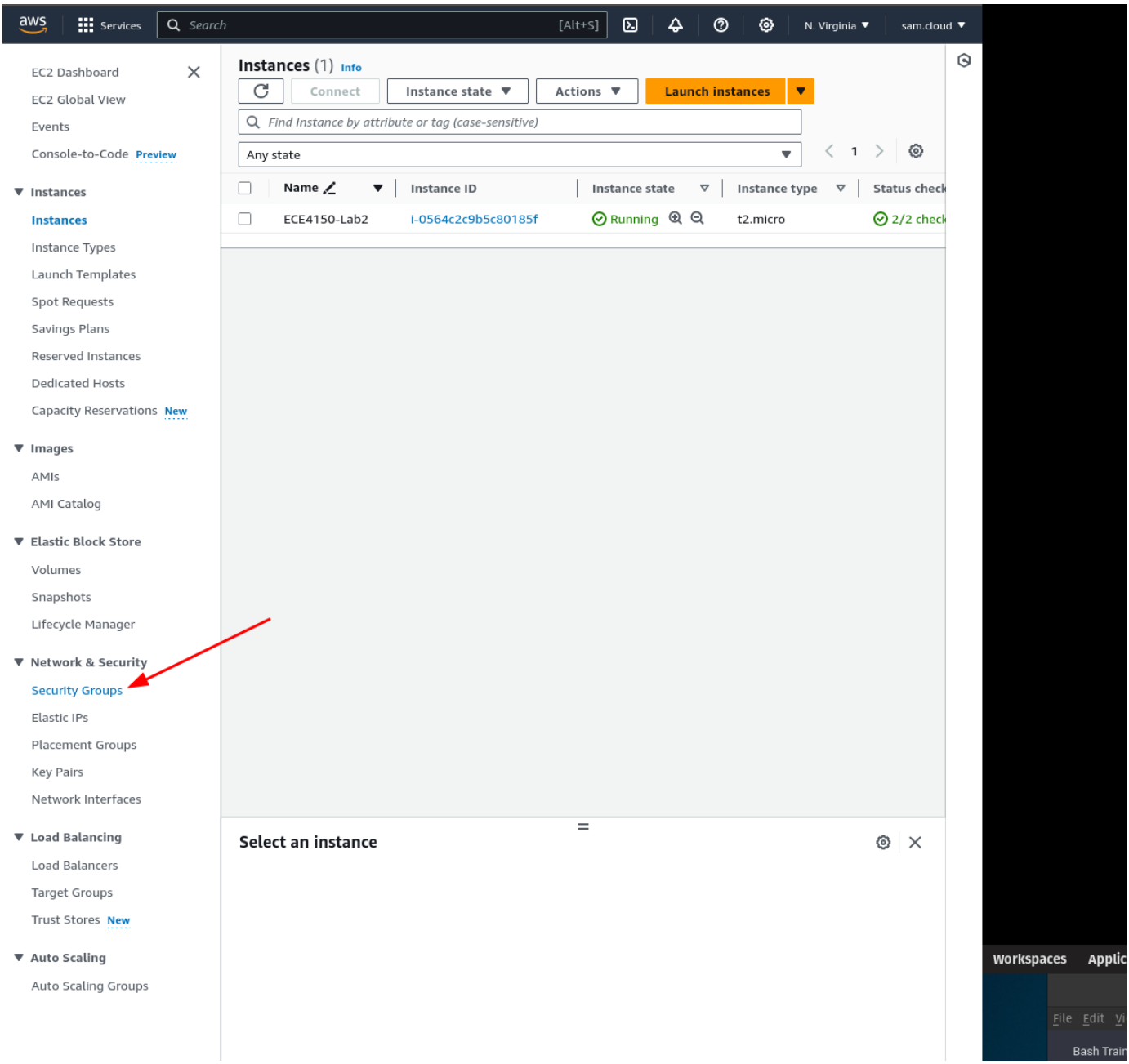


- Under the **SSH client** tab, you will nd instructions to understand how to connect to your instance remotely

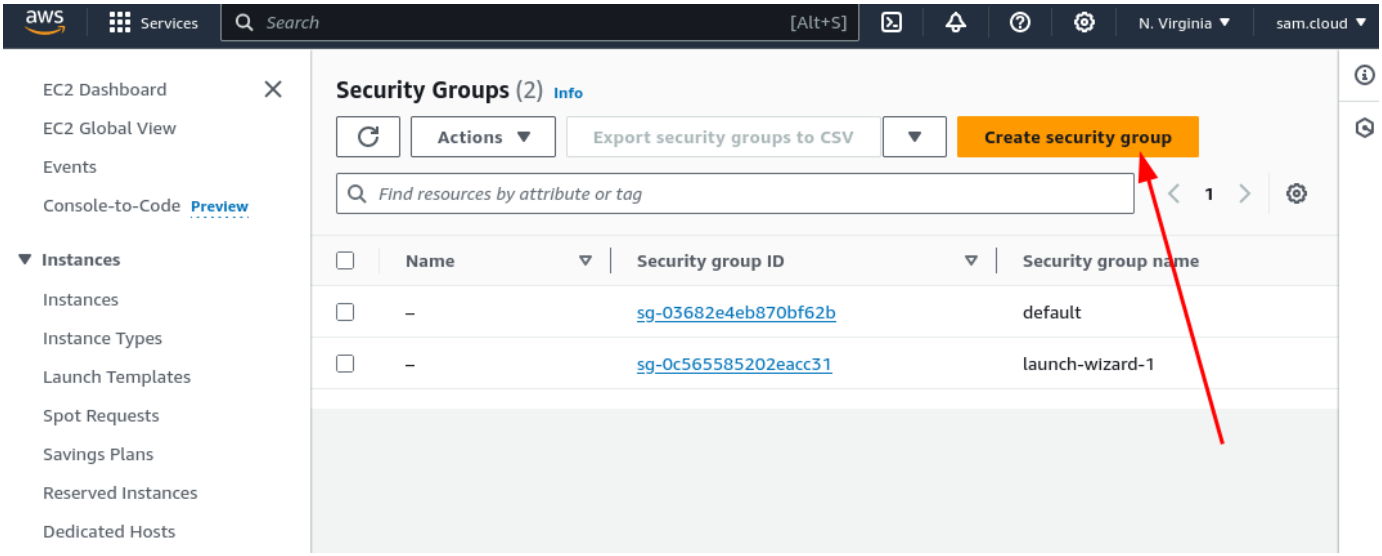


2. Create a Security Group (SG) to allow access from your EC2 from anywhere or a specific location (4 points)

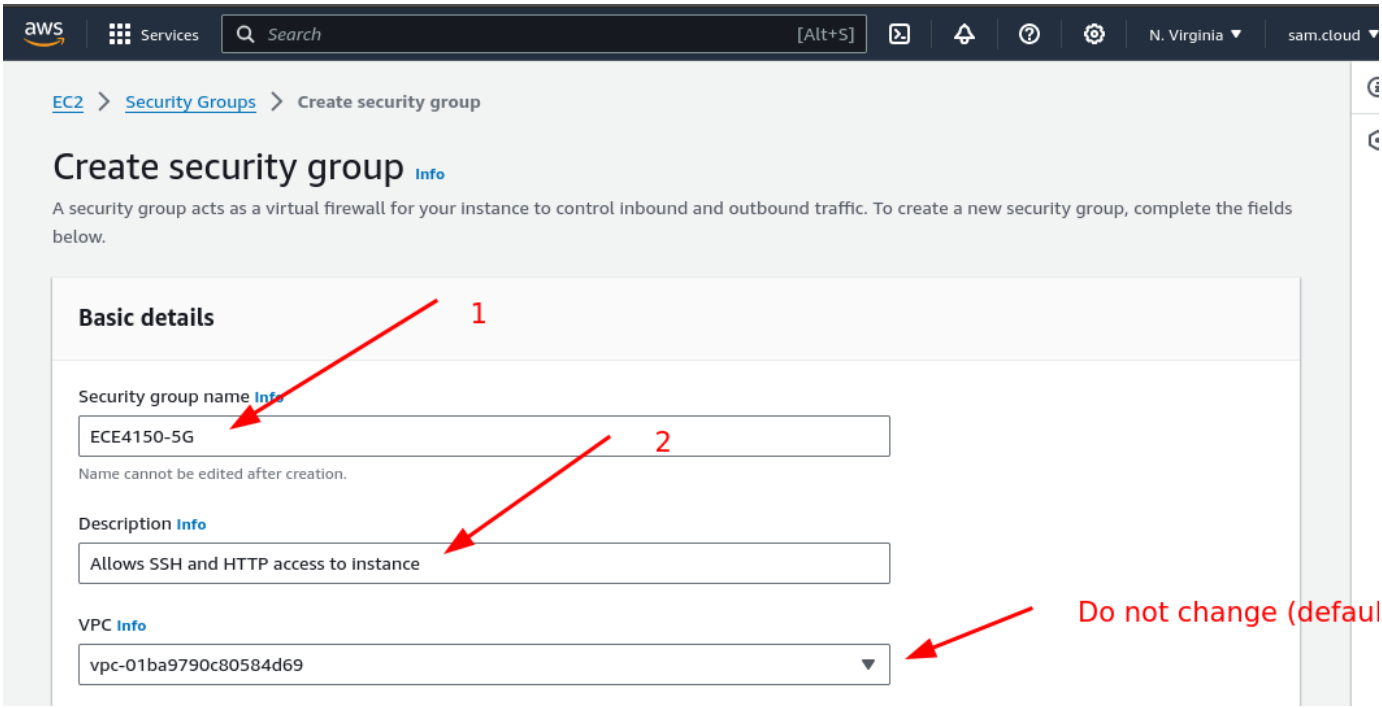
- In your EC2 Console, under **Network & Security**, click **Security Groups**. You can also use the search input in the navigation bar by typing "**Security Groups**" Under "**Features**"



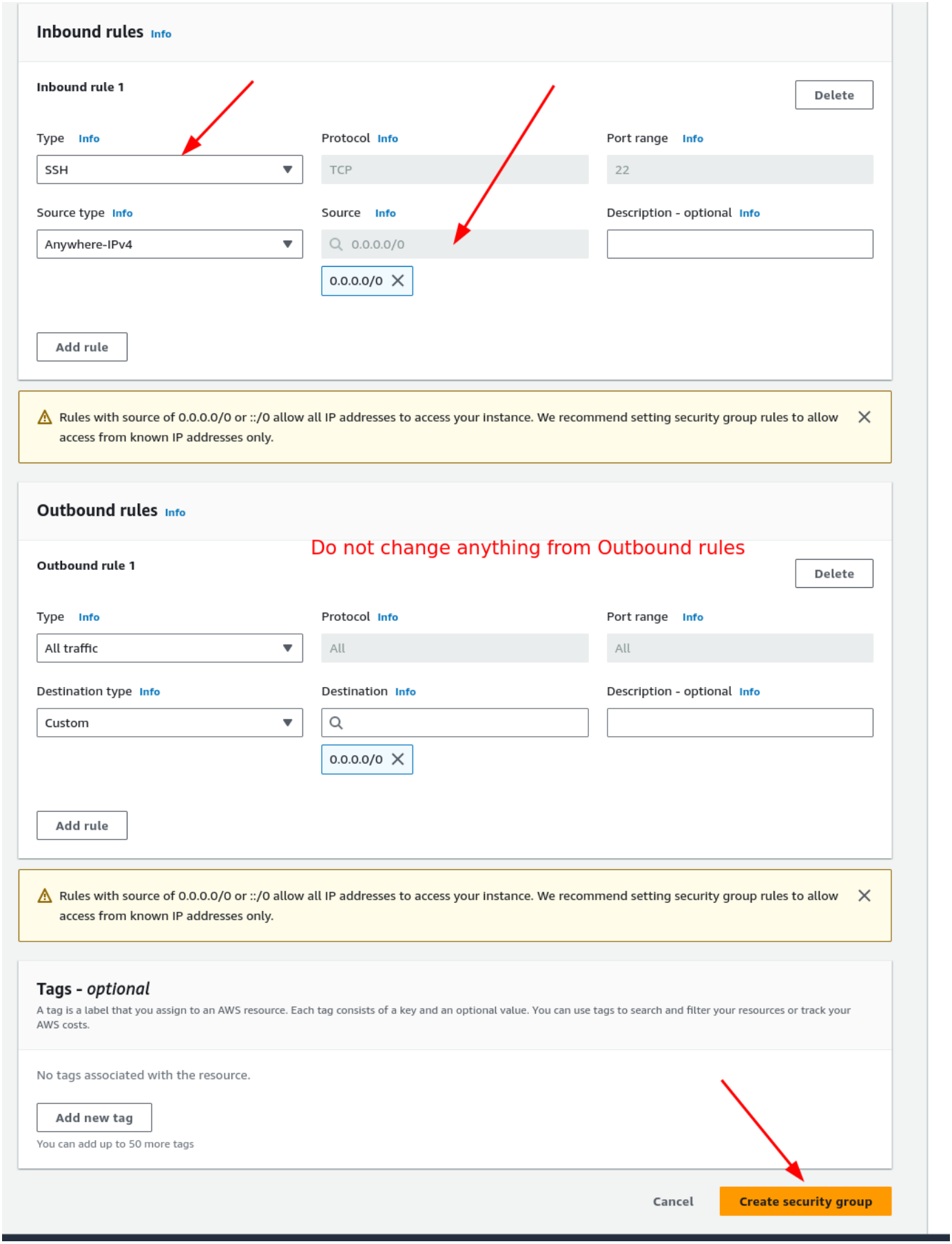
- Create a new Security Group by clicking the orange button called "**Create security group**" in the top right corner



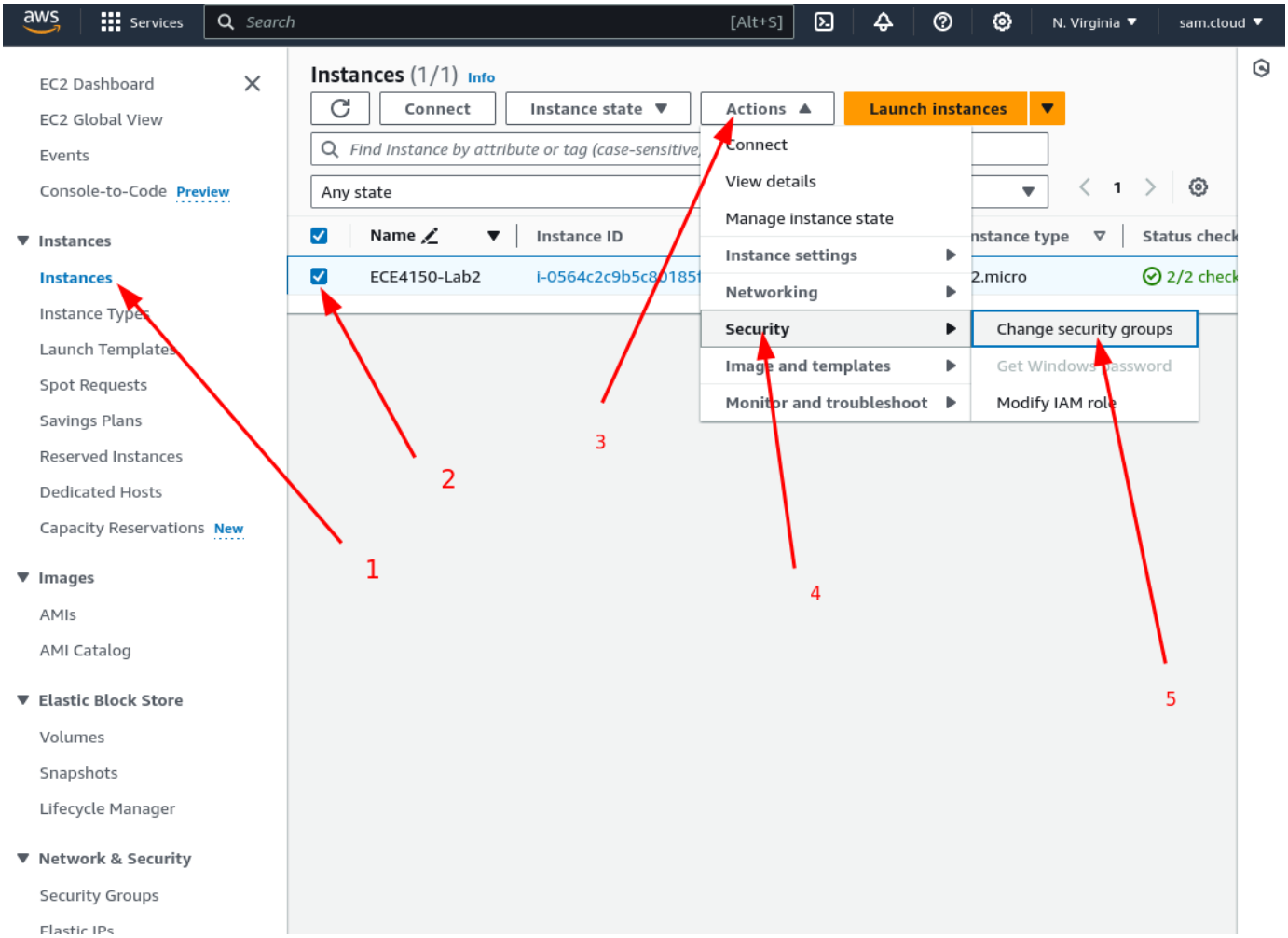
- Add a name and description to the security group as shown below



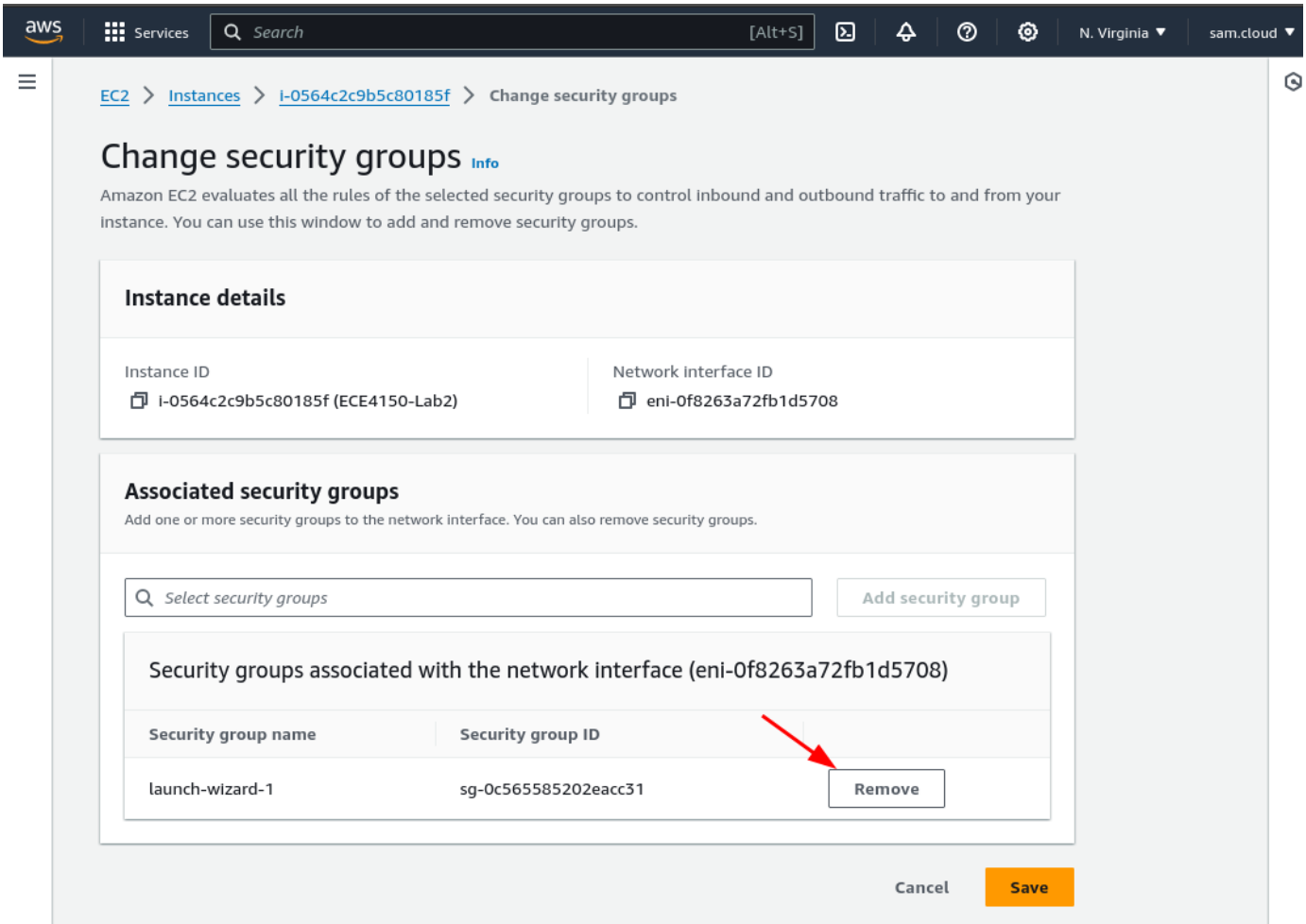
- At this time, we are going to **add one single inbound rule**. We will **allow access through SSH from anywhere**, as shown below. **Do not change the configuration for the outbound rule**

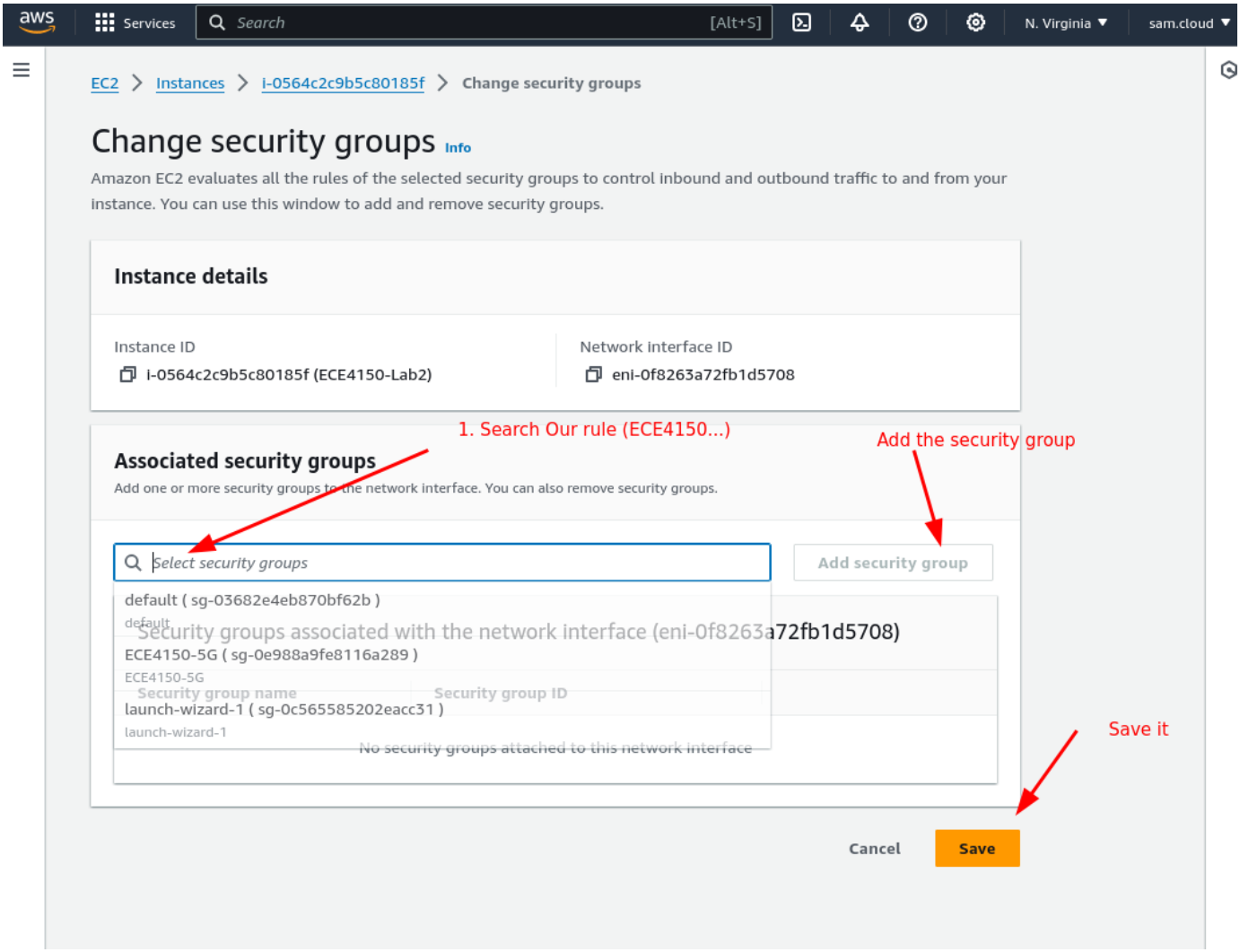


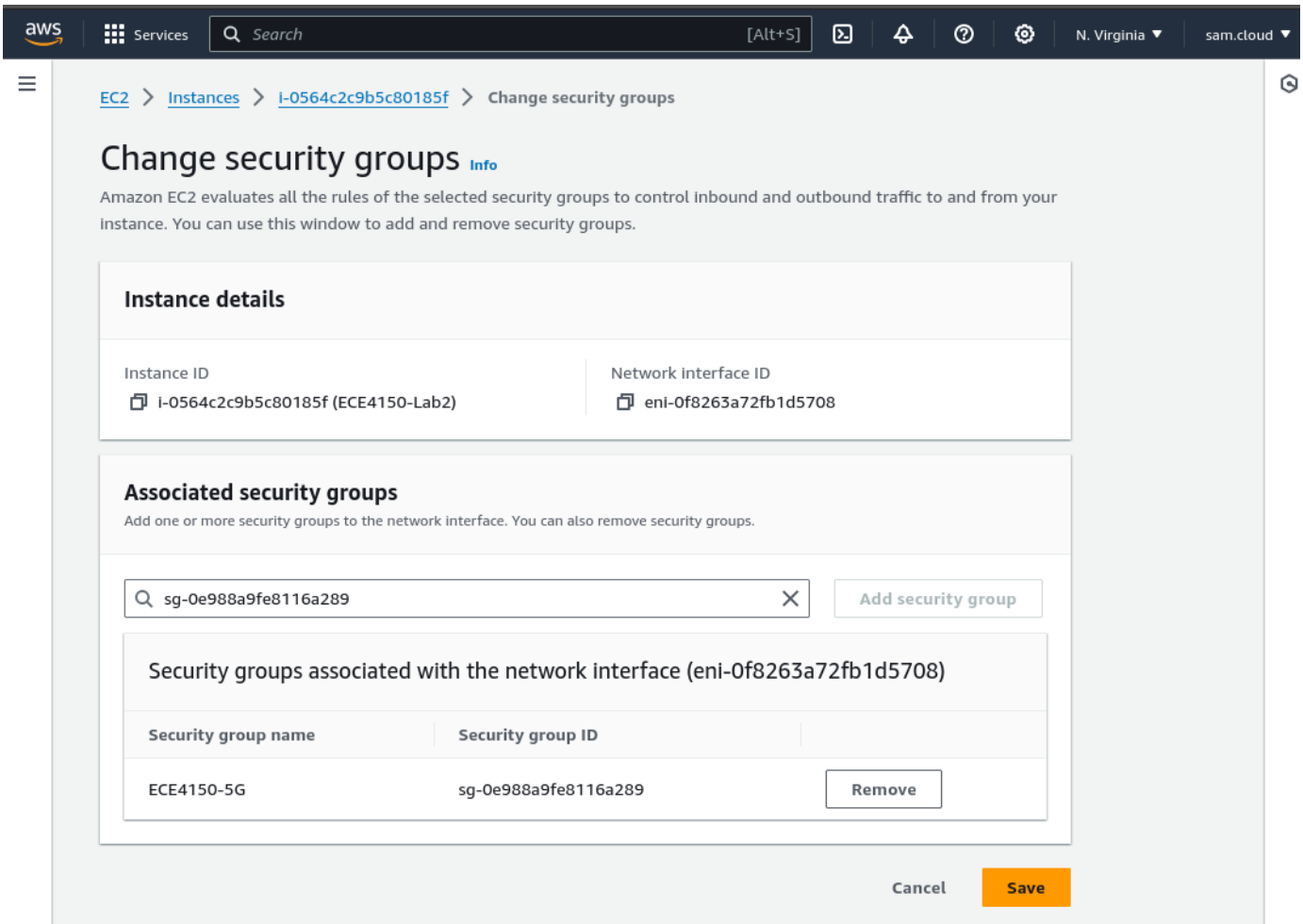
- Attached the new security group to the EC2 instance. Click “**instances**” in the left navigation menu. Select your server instance, and under **Actions**, click **Security** and **Change security groups**



- Remove the attached security group and add your new security group, then click "**Save**"




Services
Search
[Alt+S]
N. Virginia
sam.cloud
EC2 > Instances > i-0564c2c9b5c80185f > Change security groups
Change security groups Info
Amazon EC2 evaluates all the rules of the selected security groups to control inbound and outbound traffic to and from your instance. You can use this window to add and remove security groups.
Instance details
Instance ID
i-0564c2c9b5c80185f (ECE4150-Lab2)
Network interface ID
eni-0f8263a72fb1d5708
1. Search Our rule (ECE4150...)
Add the security group
Associated security groups
Add one or more security groups to the network interface. You can also remove security groups.
Select security groups
Add security group
default (sg-03682e4eb870bf62b)
default
Security groups associated with the network interface (eni-0f8263a72fb1d5708)
ECE4150-5G (sg-0e988a9fe8116a289)
ECE4150-5G
Security group name
Security group ID
launch-wizard-1 (sg-0c565585202eacc31)
launch-wizard-1
No security groups attached to this network interface
Save it
Cancel
Save


Services
Search
[Alt+S]
N. Virginia
sam.cloud
EC2 > Instances > i-0564c2c9b5c80185f > Change security groups
Change security groups Info
Amazon EC2 evaluates all the rules of the selected security groups to control inbound and outbound traffic to and from your instance. You can use this window to add and remove security groups.
Instance details
Instance ID
i-0564c2c9b5c80185f (ECE4150-Lab2)
Network interface ID
eni-0f8263a72fb1d5708
Associated security groups
Add one or more security groups to the network interface. You can also remove security groups.
sg-0e988a9fe8116a289
Add security group
Security groups associated with the network interface (eni-0f8263a72fb1d5708)
Security group name
Security group ID
ECE4150-5G
sg-0e988a9fe8116a289
Remove
Cancel
Save

3. Create an RDS database instance for storing records of photos (10 points)

- In the AWS Management Console, under Database, click **RDS**. You can also use the search input in the navigation bar by typing “**RDS**”
- Lunch a database instance by clicking the orange button “**Create database**”
- Follow the configurations as shown below

Create database

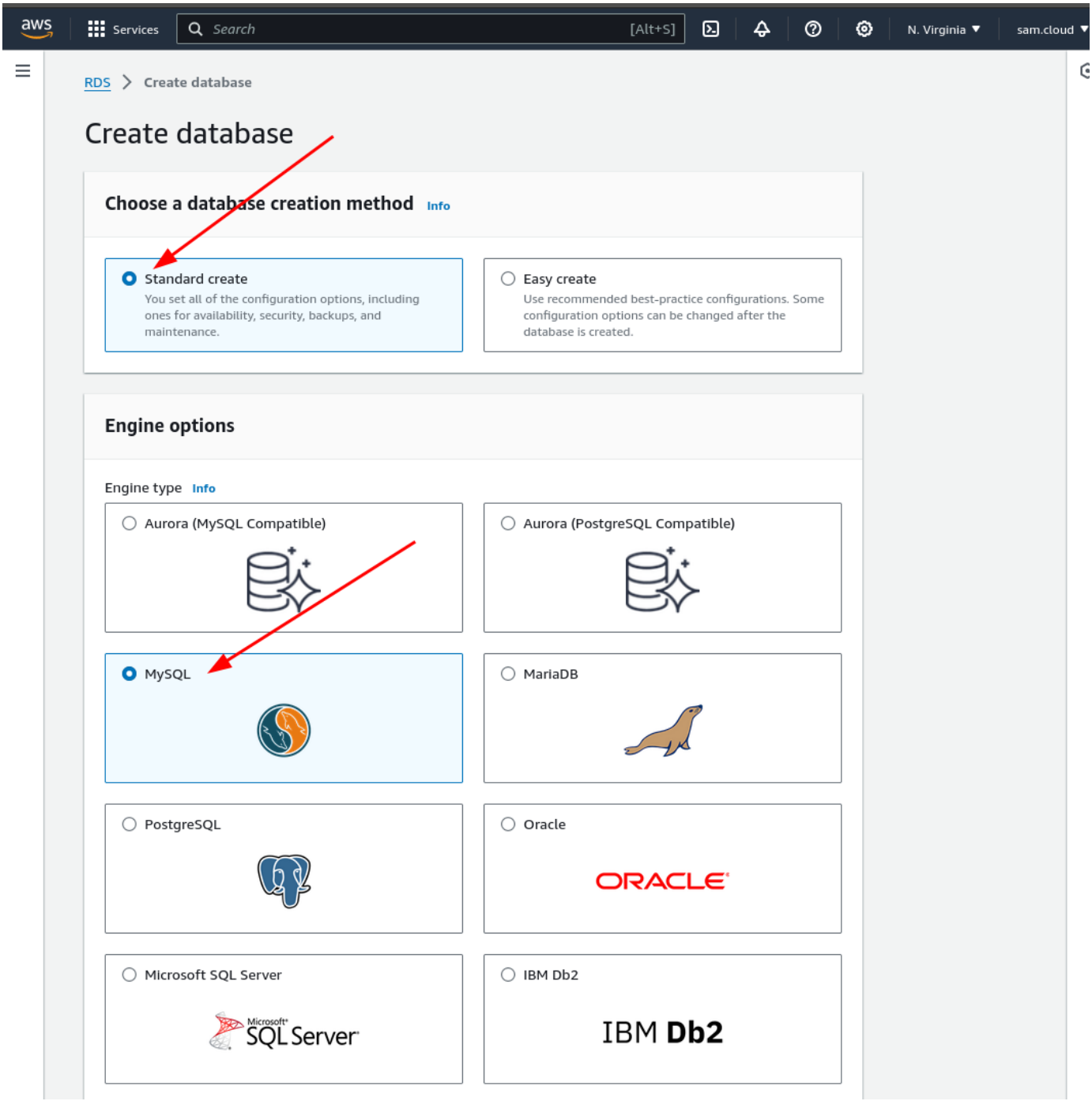
Choose a database creation method Info

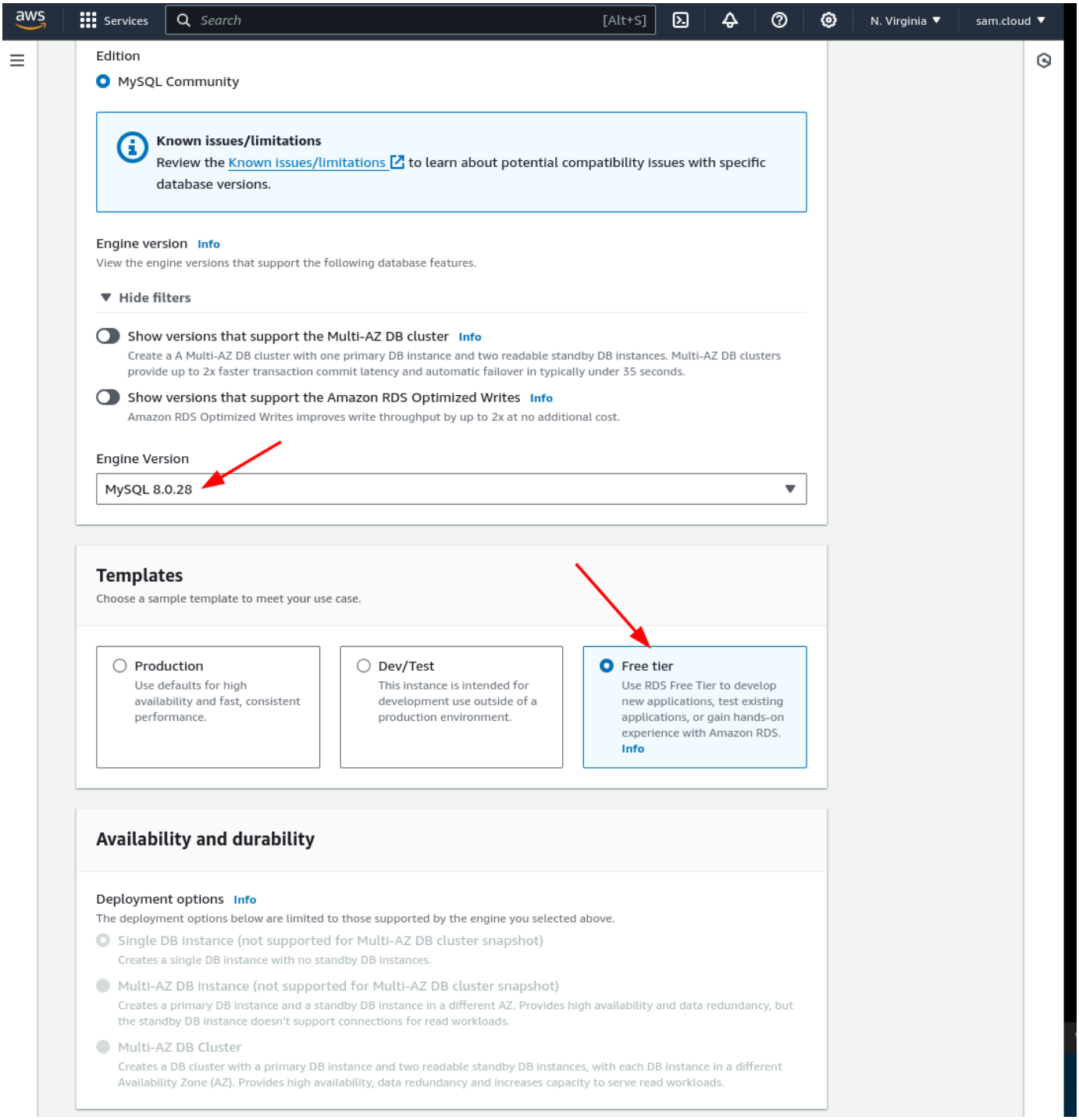
- (•) **Standard create**
  You set all of the configuration options, including ones for availability, security, backups, and maintenance.
- ( ) **Easy create**
  Use recommended best-practice configurations. Some configuration options can be changed after the database is created.

Engine options

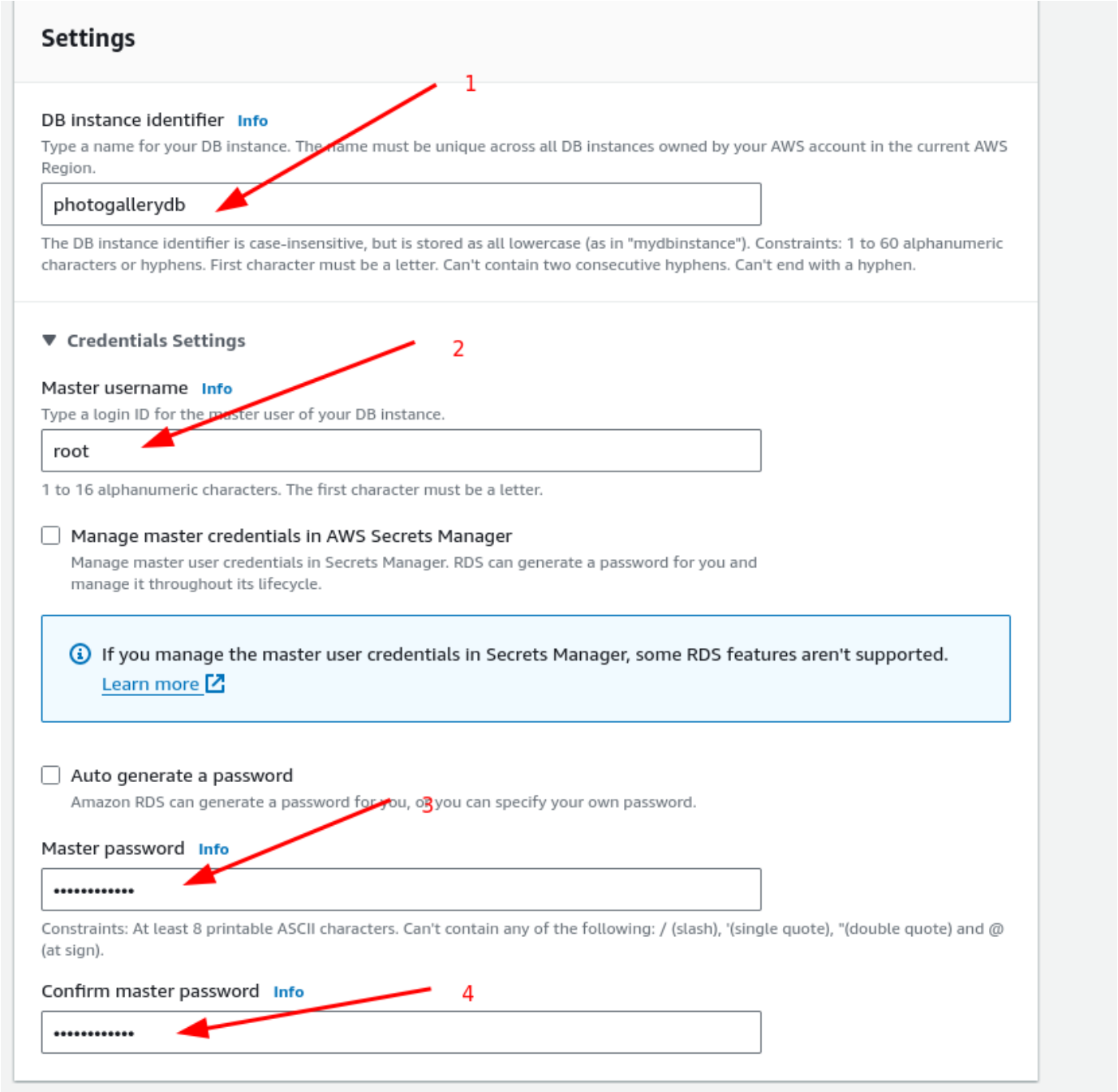
Engine type Info

- ( ) Aurora (MySQL Compatible)
- ( ) Aurora (PostgreSQL Compatible)
- (•) MySQL
- ( ) MariaDB
- ( ) PostgreSQL
- ( ) Oracle
- ( ) Microsoft SQL Server
- ( ) IBM Db2

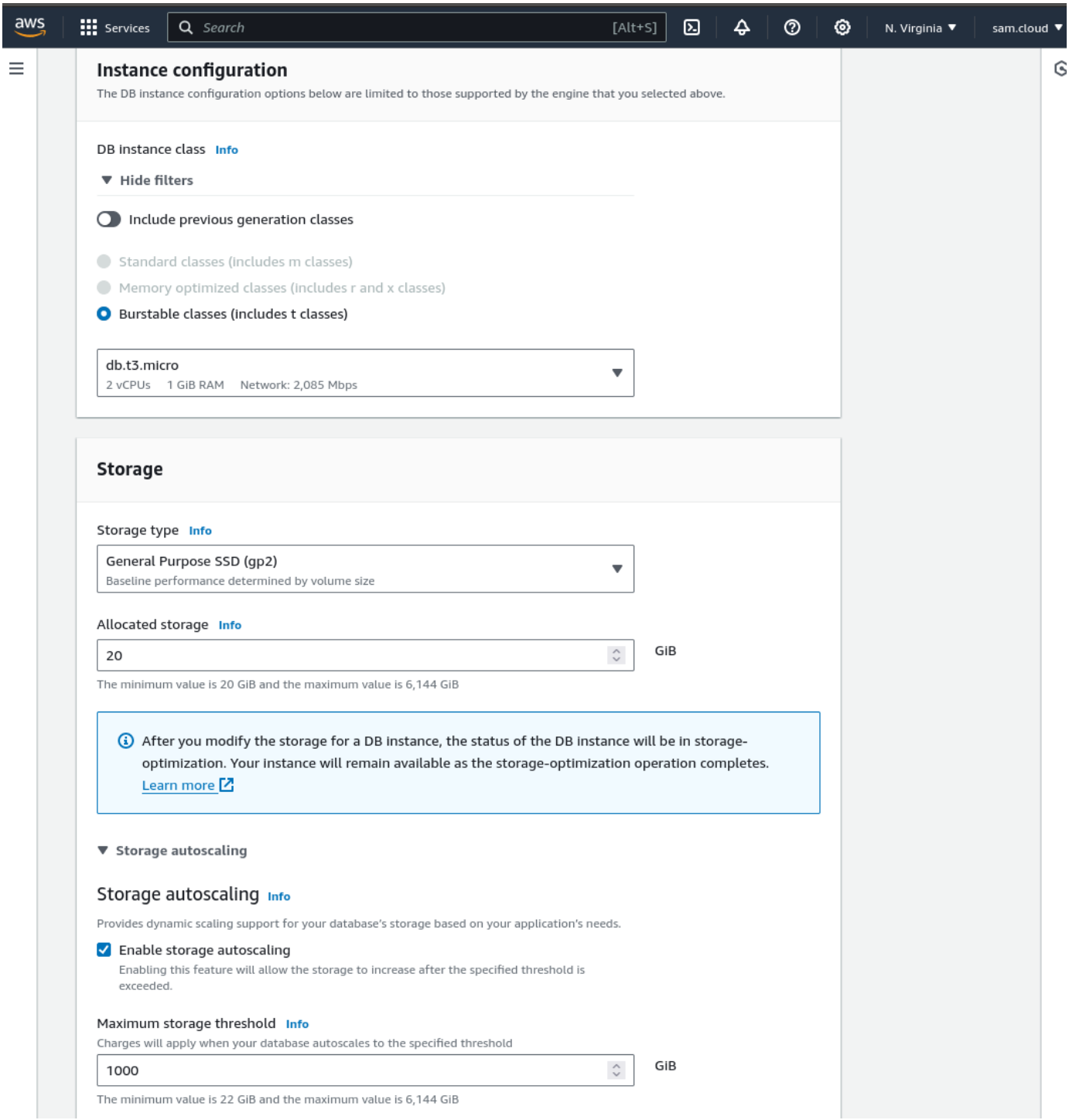

aws
Services
Search
[Alt+S]
N. Virginia
sam.cloud
RDS
Create database
Create database
Choose a database creation method
Info
Standard create
You set all of the configuration options, including ones for availability, security, backups, and maintenance.
Easy create
Use recommended best-practice configurations. Some configuration options can be changed after the database is created.
Engine options
Engine type
Info
Aurora (MySQL Compatible)
Aurora (PostgreSQL Compatible)
MySQL
MariaDB
PostgreSQL
Oracle
ORACLE
Microsoft SQL Server
Microsoft SQL Server
IBM Db2
IBM Db2



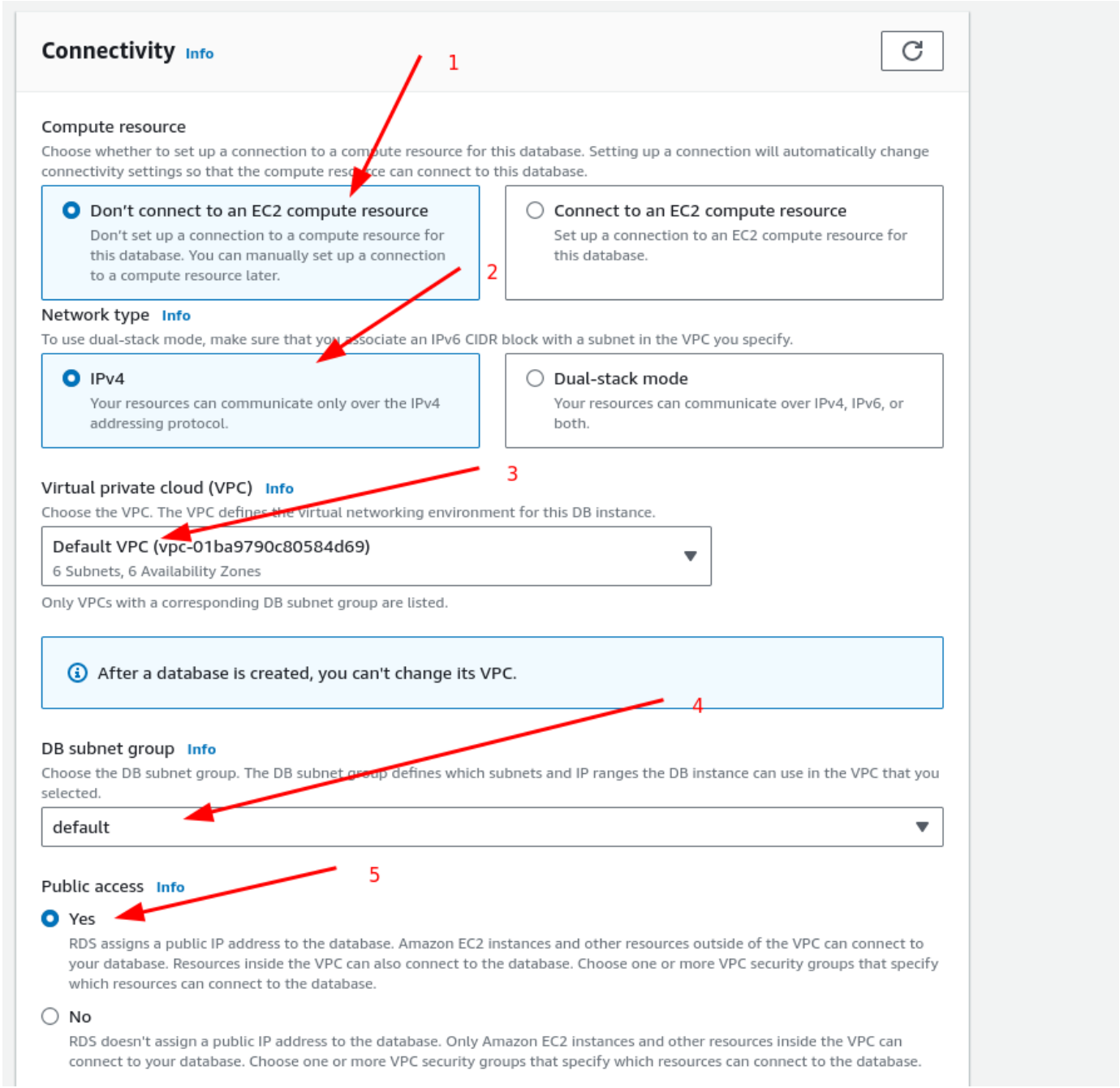
- Under settings, for the “**DB instance identifier**”, type the name of your DB instance. For this lab, it is going to be **photogallerydb**. For the Master username, type the the login ID use to connect to the database instance; we are going to use **root** for the login ID. Create a password for authentication.



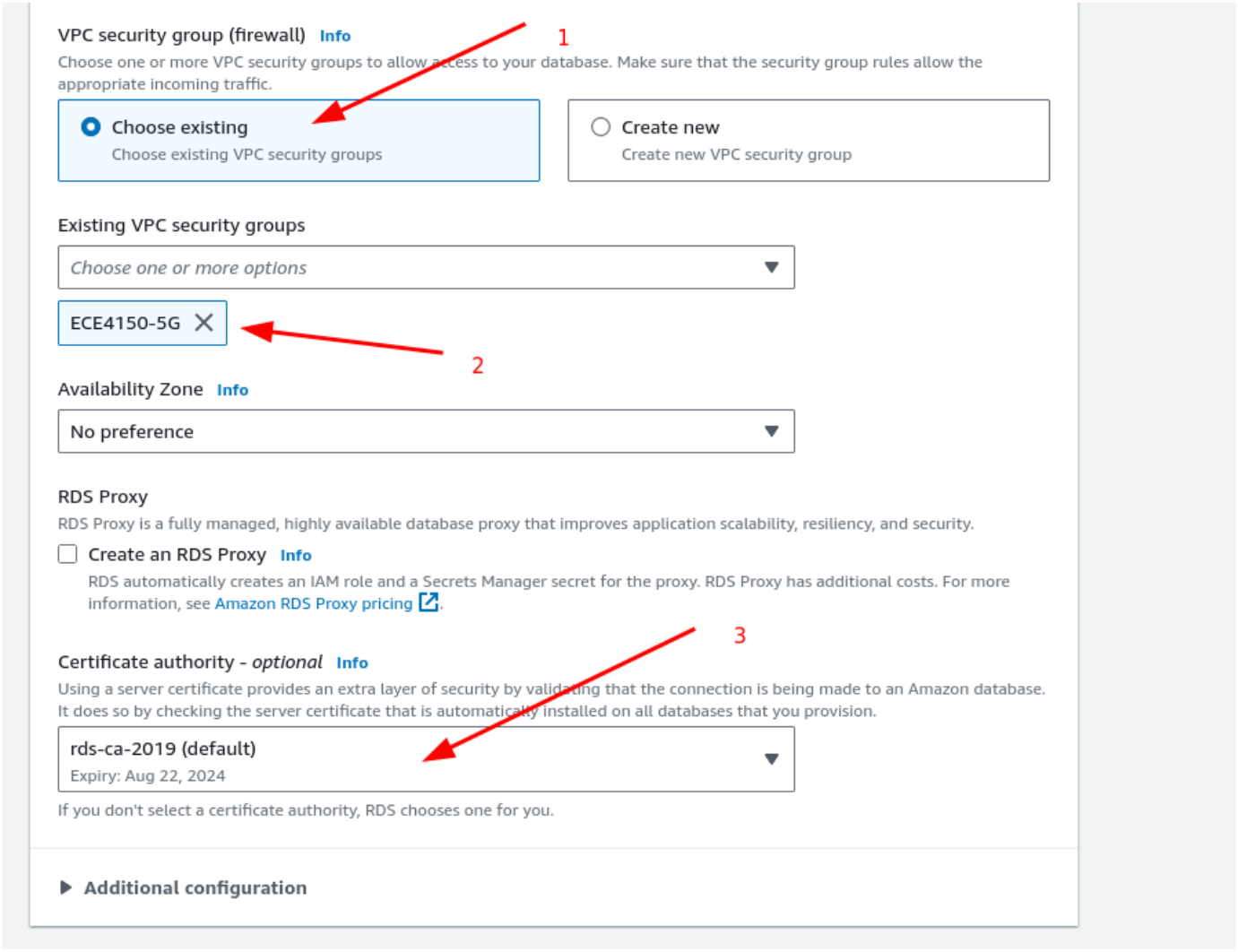
- Follow the default con gurations as shown below



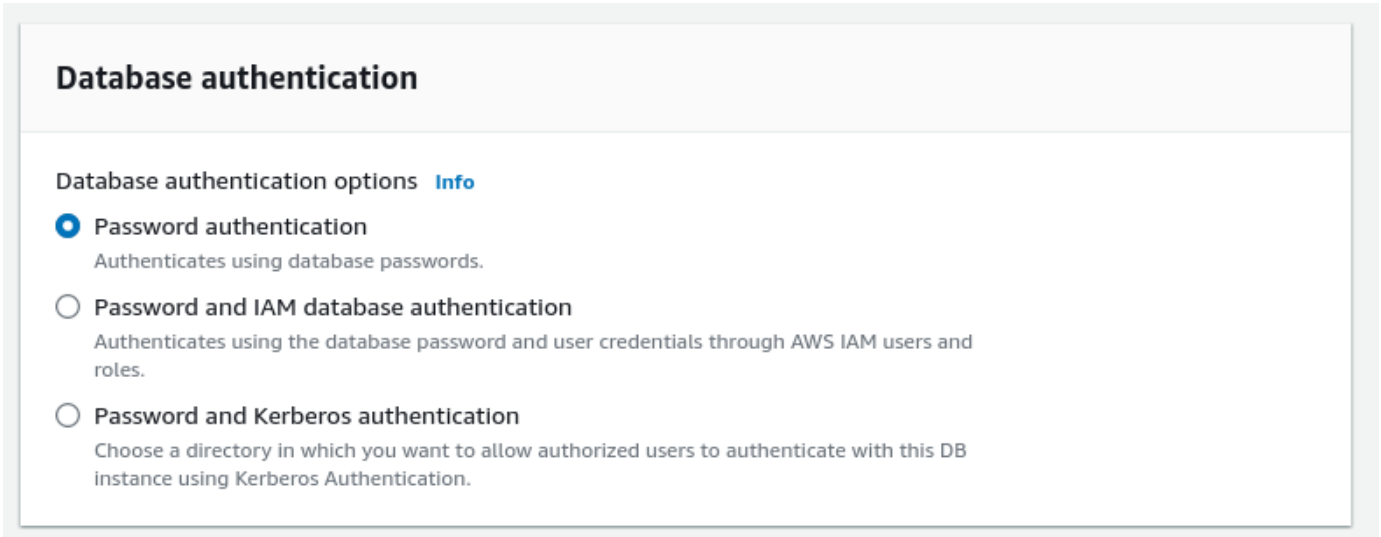
- Make sure you select the make your database publicly available over the internet, as shown below



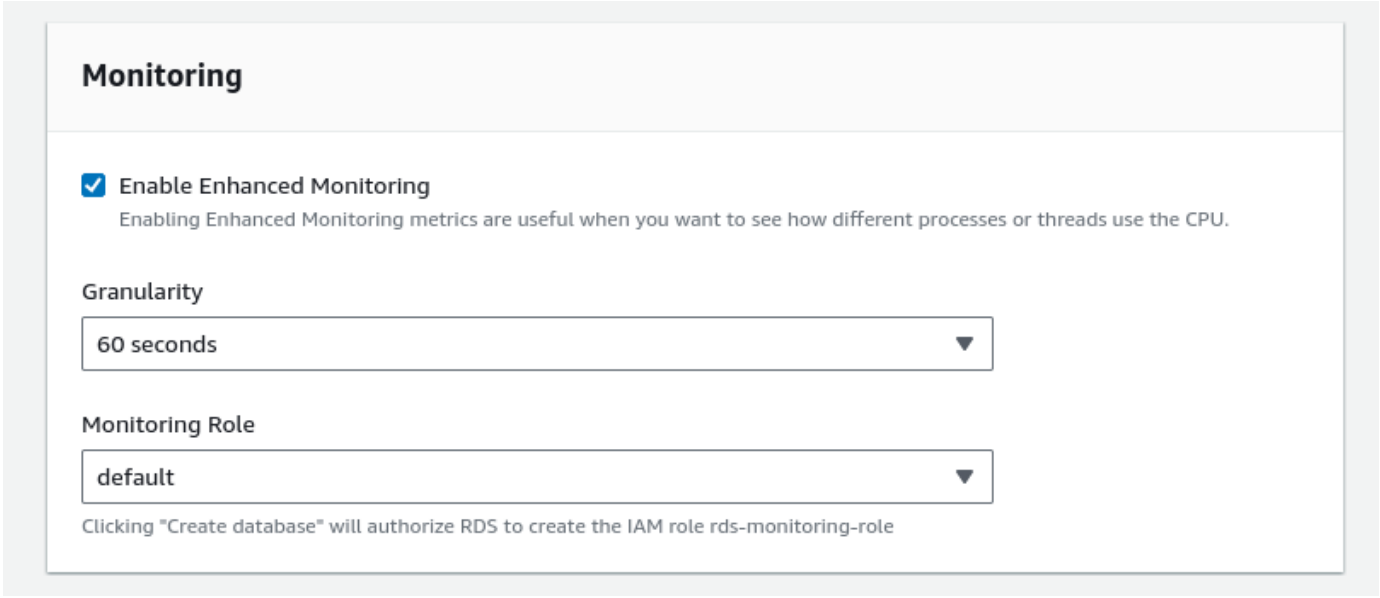
- Attach the security group that you created previously

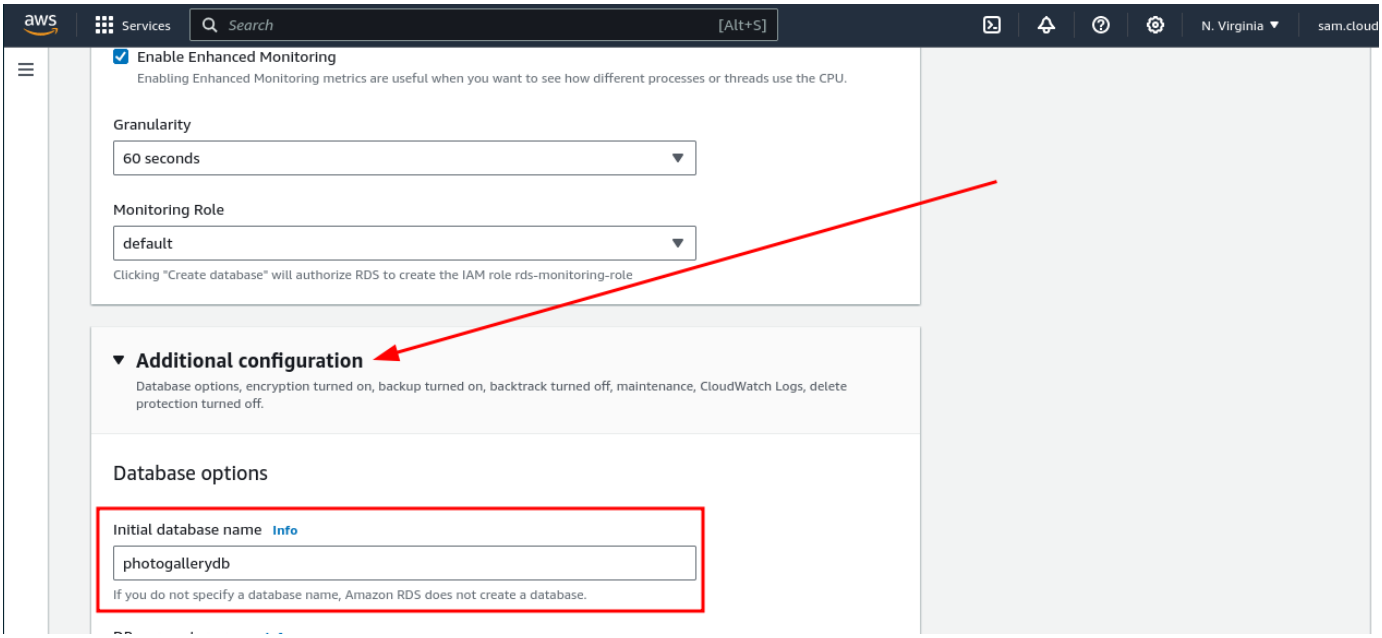


- **Database Authentication**

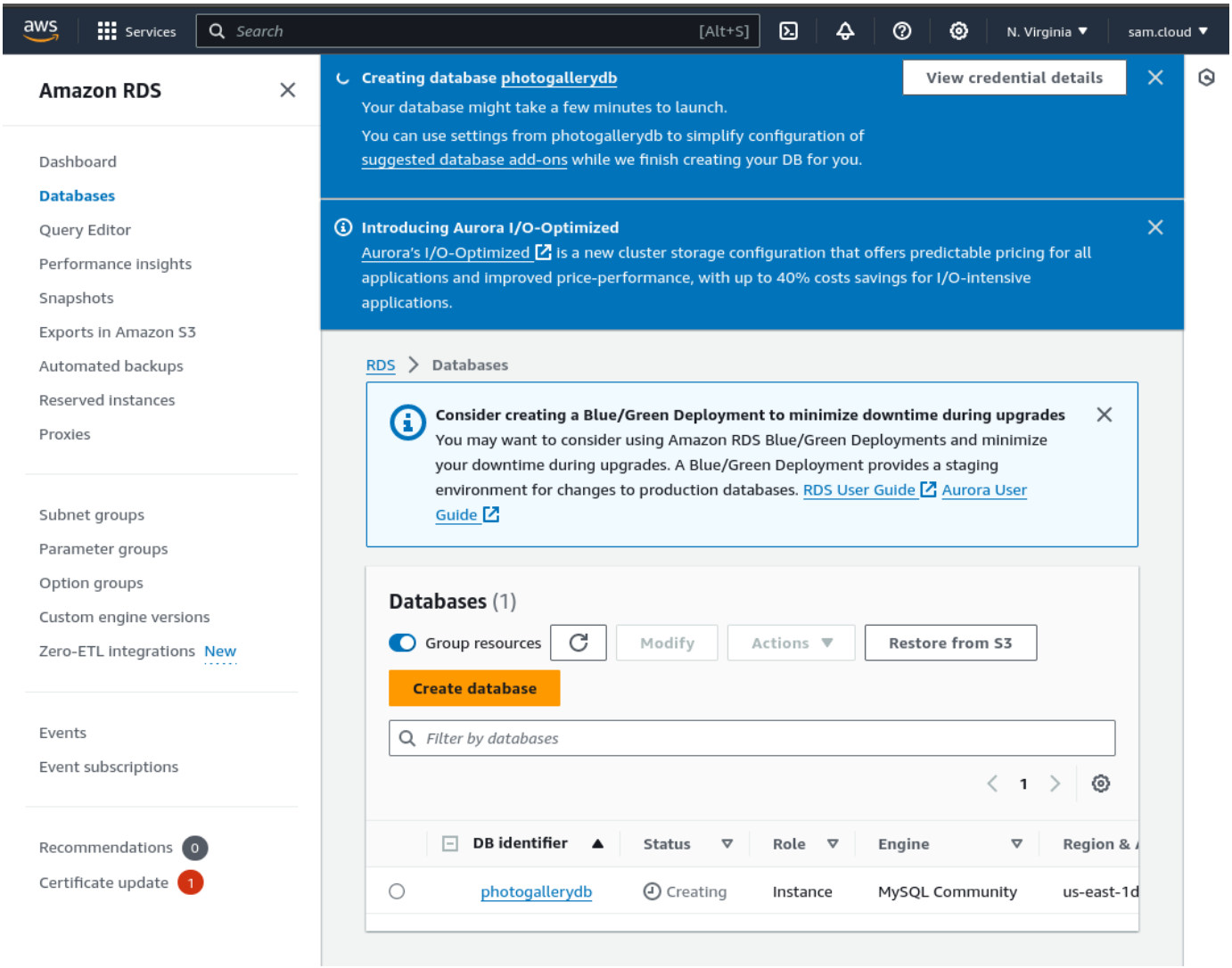


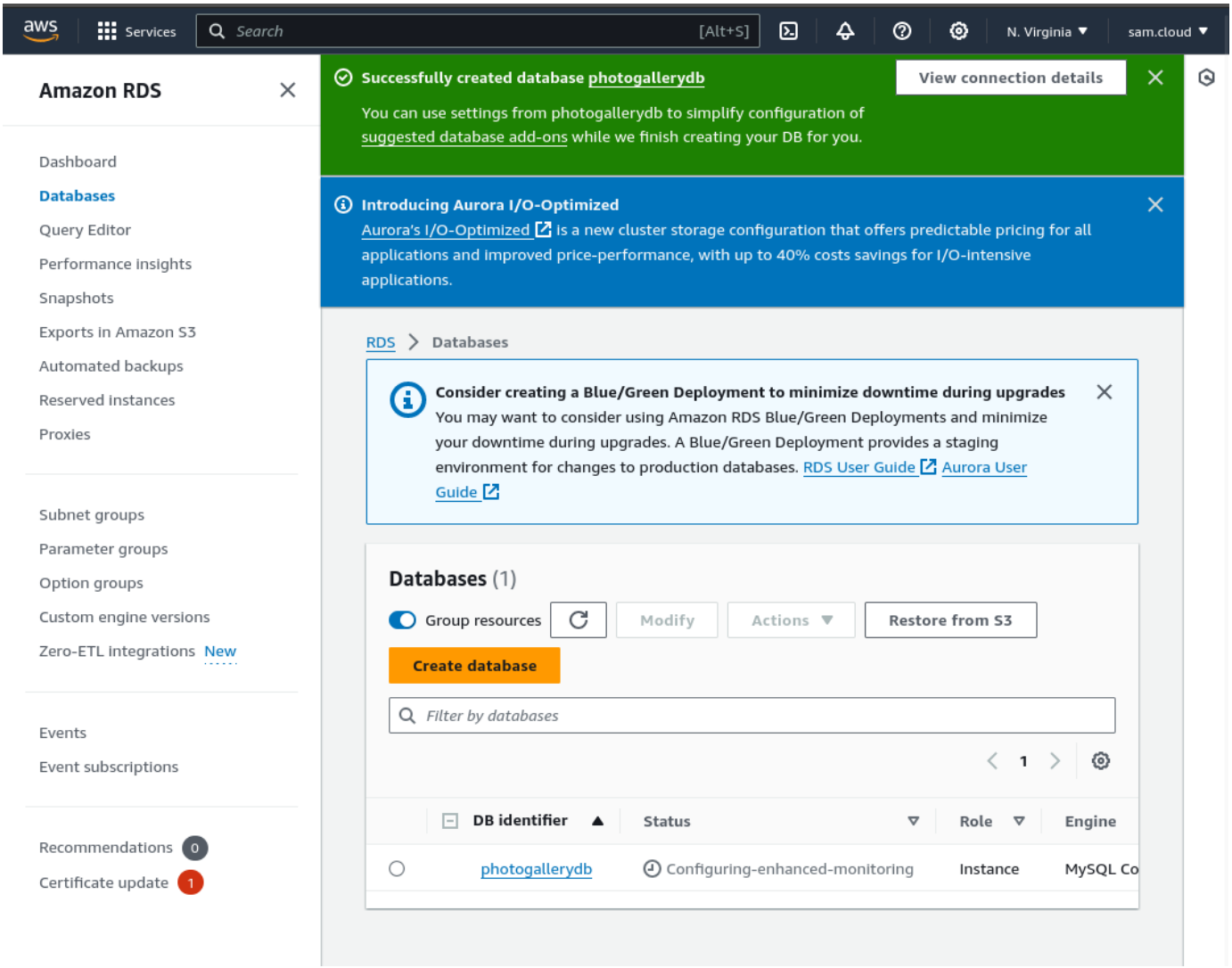
- Under **Monitoring**, "**Enable Enhanced Monitoring**" as shown below



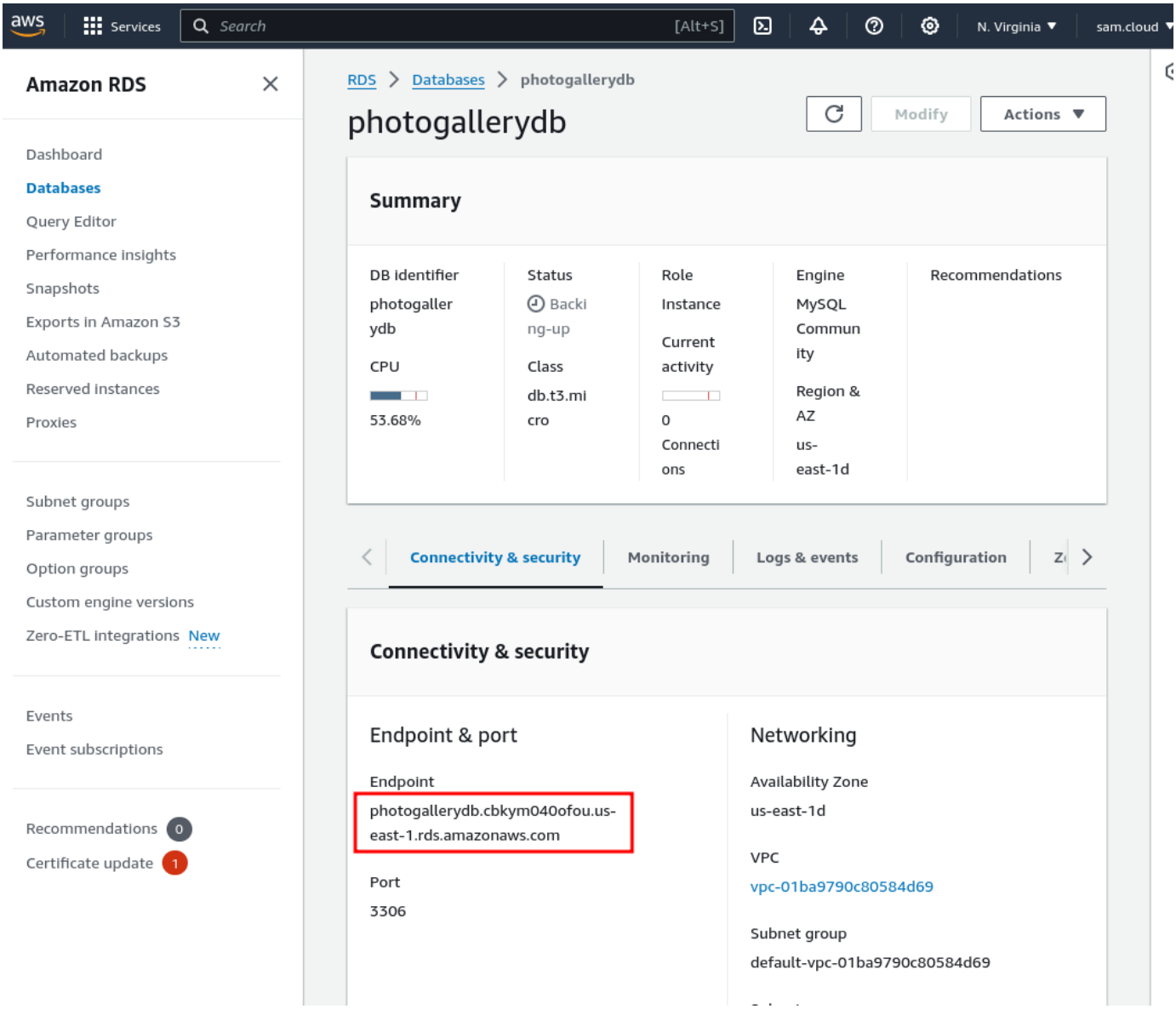


- Once you completed the configuration, create the database by clicking **Create database**
- It might take around 10-20 minutes to create, depending on the allocation. It will change from Creating to Available under Status

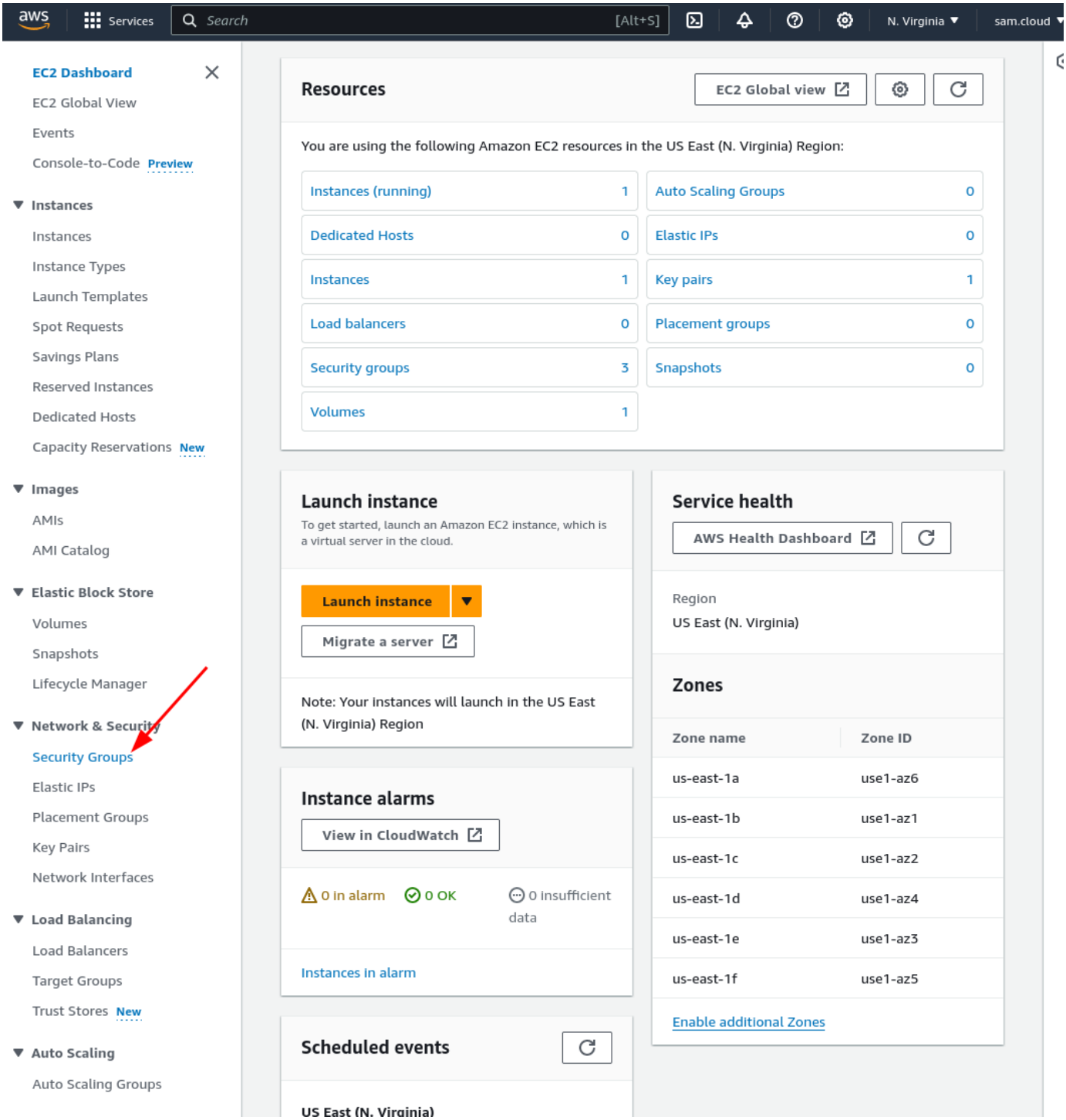

aws
Services
Search
[Alt+S]
N. Virginia
sam.cloud
Amazon RDS
Dashboard
Databases
Query Editor
Performance insights
Snapshots
Exports in Amazon S3
Automated backups
Reserved instances
Proxies
Subnet groups
Parameter groups
Option groups
Custom engine versions
Zero-ETL integrations New
Events
Event subscriptions
Recommendations 0
Certificate update 1
Creating database photogallerydb
Your database might take a few minutes to launch.
You can use settings from photogallerydb to simplify configuration of suggested database add-ons while we finish creating your DB for you.
View credential details
Introducing Aurora I/O-Optimized
Aurora's I/O-Optimized is a new cluster storage configuration that offers predictable pricing for all applications and improved price-performance, with up to 40% costs savings for I/O-intensive applications.
RDS > Databases
Consider creating a Blue/Green Deployment to minimize downtime during upgrades
You may want to consider using Amazon RDS Blue/Green Deployments and minimize your downtime during upgrades. A Blue/Green Deployment provides a staging environment for changes to production databases. RDS User Guide Aurora User Guide
Databases (1)
Group resources
Modify
Actions
Restore from S3
Create database
Filter by databases
1
DB identifier
Status
Role
Engine
Region &
photogallerydb
Creating
Instance
MySQL Community
us-east-1d

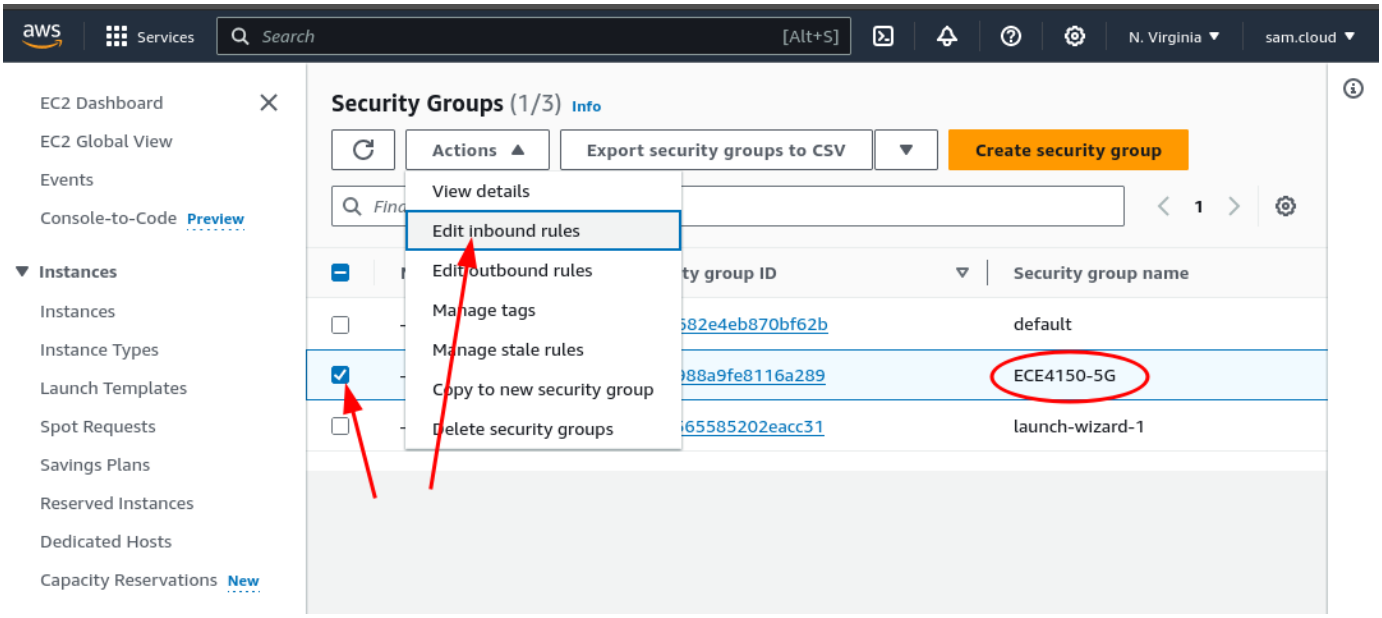


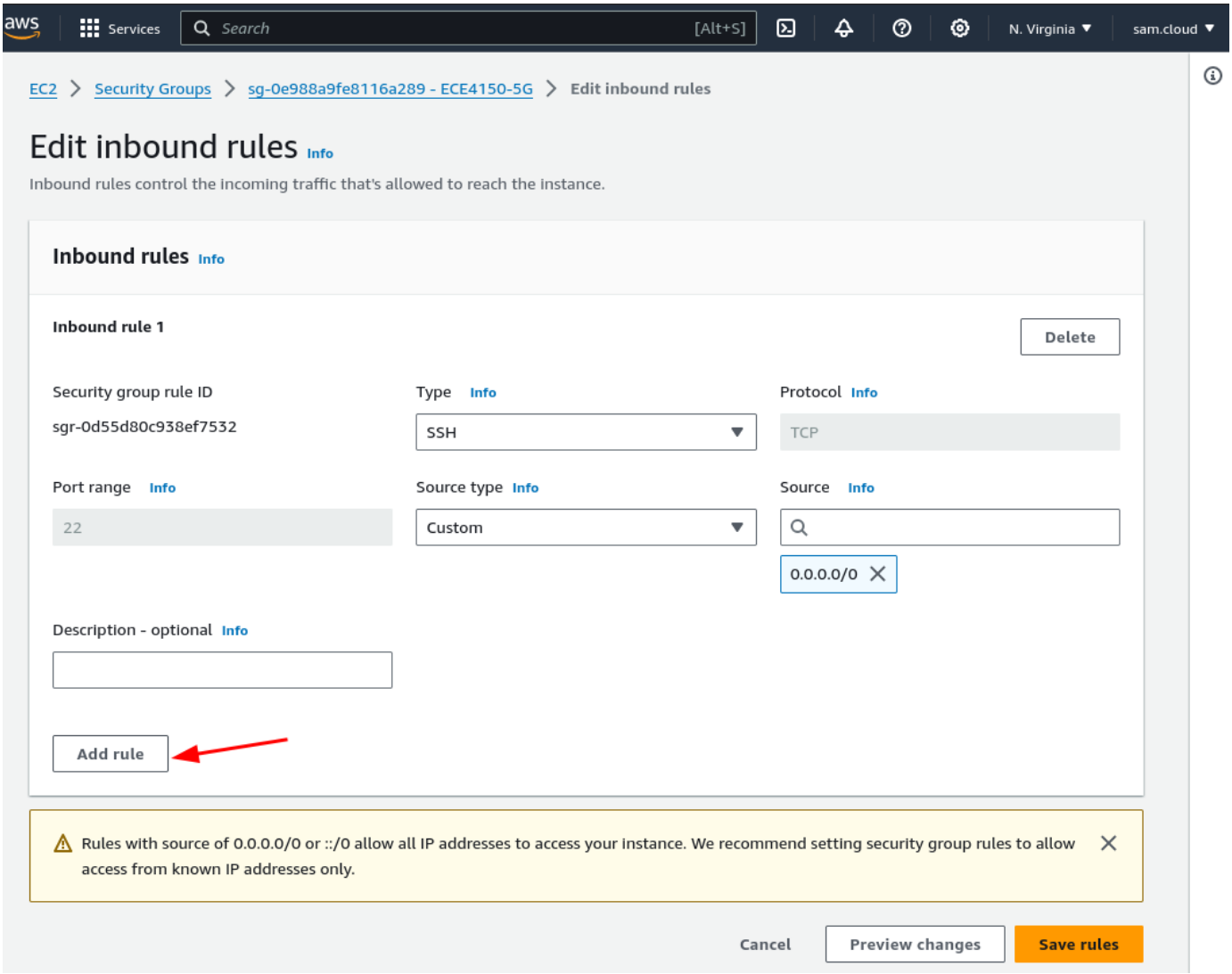
- Click on the newly created database, copy and save the endpoint to the database instance located under **Endpoint & port**, as shown below

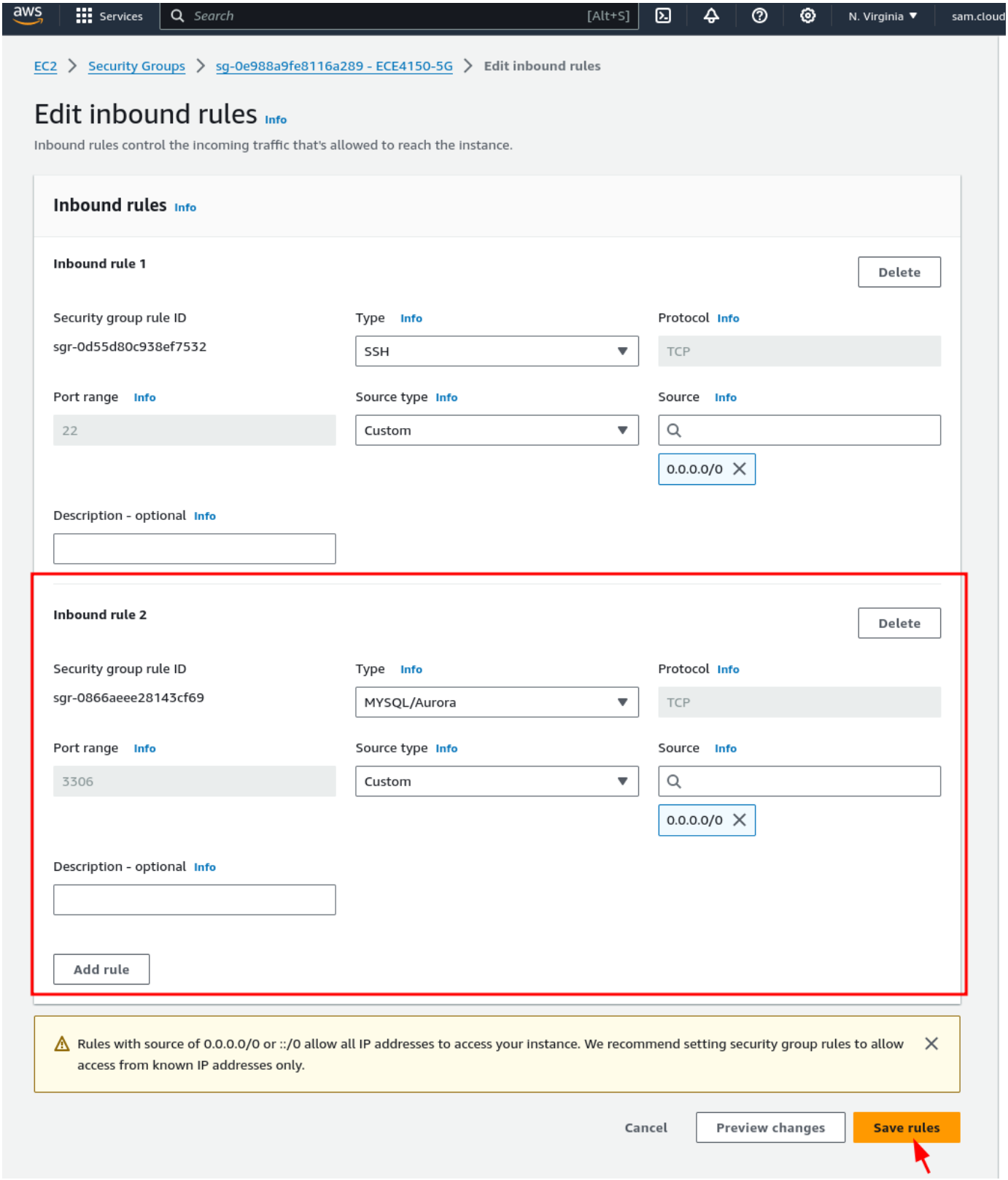


- Go back to the Security Group section under the EC2 Console (You can go there through search) and add another inbound rule for a **Custom TCP** type from port **3306** (or add a **MySQL/Aurora** rule type) and configure the source coming from anywhere (**0.0.0.0/0**)


aws
Services
Search
[Alt+S]
N. Virginia
sam.cloud
EC2 Dashboard
EC2 Global View
Events
Console-to-Code Preview
Instances
Instances
Instance Types
Launch Templates
Spot Requests
Savings Plans
Reserved Instances
Dedicated Hosts
Capacity Reservations New
Images
AMIs
AMI Catalog
Elastic Block Store
Volumes
Snapshots
Lifecycle Manager
Network & Security
Security Groups
Elastic IPs
Placement Groups
Key Pairs
Network Interfaces
Load Balancing
Load Balancers
Target Groups
Trust Stores New
Auto Scaling
Auto Scaling Groups
Resources
EC2 Global view
You are using the following Amazon EC2 resources in the US East (N. Virginia) Region:
Instances (running) 1
Auto Scaling Groups 0
Dedicated Hosts 0
Elastic IPs 0
Instances 1
Key pairs 1
Load balancers 0
Placement groups 0
Security groups 3
Snapshots 0
Volumes 1
Launch instance
To get started, launch an Amazon EC2 instance, which is a virtual server in the cloud.
Launch instance
Migrate a server
Note: Your instances will launch in the US East (N. Virginia) Region
Service health
AWS Health Dashboard
Region
US East (N. Virginia)
Zones
Zone name
Zone ID
us-east-1a use1-az6
us-east-1b use1-az1
us-east-1c use1-az2
us-east-1d use1-az4
us-east-1e use1-az3
us-east-1f use1-az5
Enable additional Zones
Instance alarms
View in CloudWatch
0 in alarm
0 OK
0 insufficient data
Instances in alarm
Scheduled events
US East (N. Virginia)

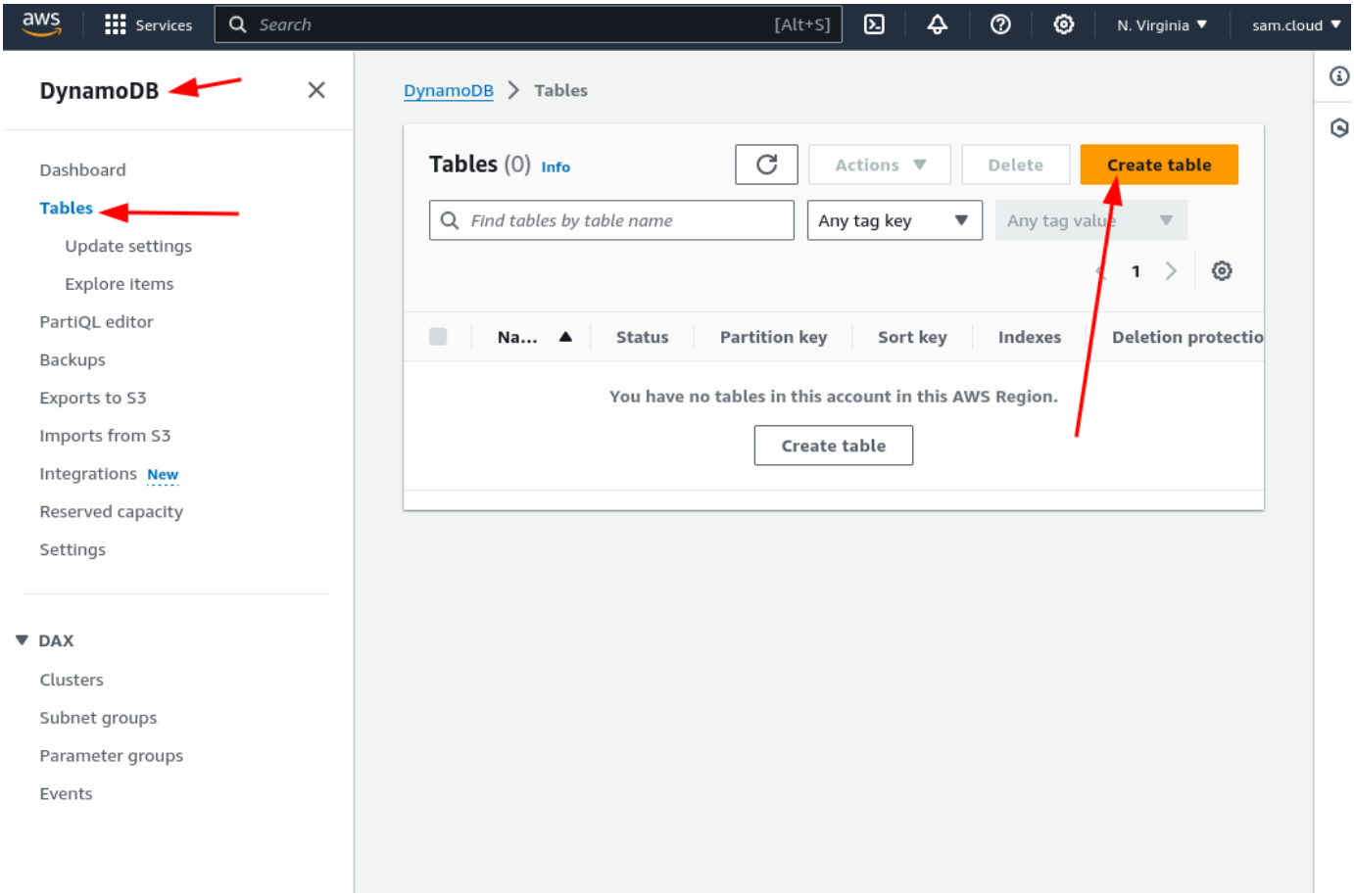

aws
Services
Search
[Alt+S]
N. Virginia
sam.cloud
EC2 Dashboard
EC2 Global View
Events
Console-to-Code Preview
Instances
Instances
Instance Types
Launch Templates
Spot Requests
Savings Plans
Reserved Instances
Dedicated Hosts
Capacity Reservations New
Security Groups (1/3) Info
Actions
Export security groups to CSV
Create security group
View details
Edit inbound rules
Edit outbound rules
Manage tags
Manage stale rules
Copy to new security group
Delete security groups
Security group name
default
ECE4150-5G
launch-wizard-1

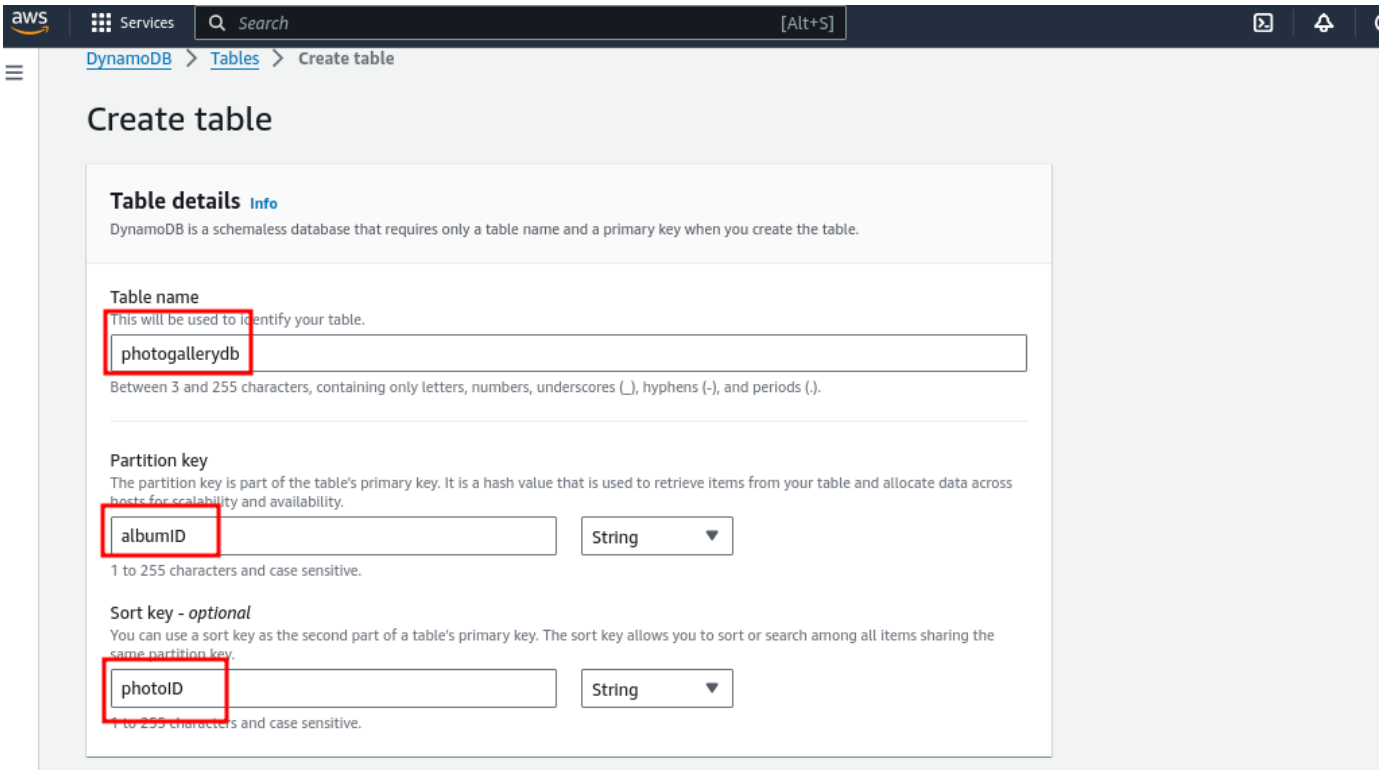

aws
Services
Search
[Alt+S]
N. Virginia
sam.cloud
EC2
Security Groups
sg-0e988a9fe8116a289 - ECE4150-5G
Edit inbound rules
Edit inbound rules Info
Inbound rules control the incoming traffic that's allowed to reach the instance.
Inbound rules Info
Inbound rule 1
Delete
Security group rule ID
sgr-0d55d80c938ef7532
Type Info
SSH
Protocol Info
TCP
Port range Info
22
Source type Info
Custom
Source Info
0.0.0.0/0
Description - optional Info
Add rule
Rules with source of 0.0.0.0/0 or ::/0 allow all IP addresses to access your instance. We recommend setting security group rules to allow access from known IP addresses only.
Cancel
Preview changes
Save rules



4. Create a DynamoDB table for storing records of photos (2 points)

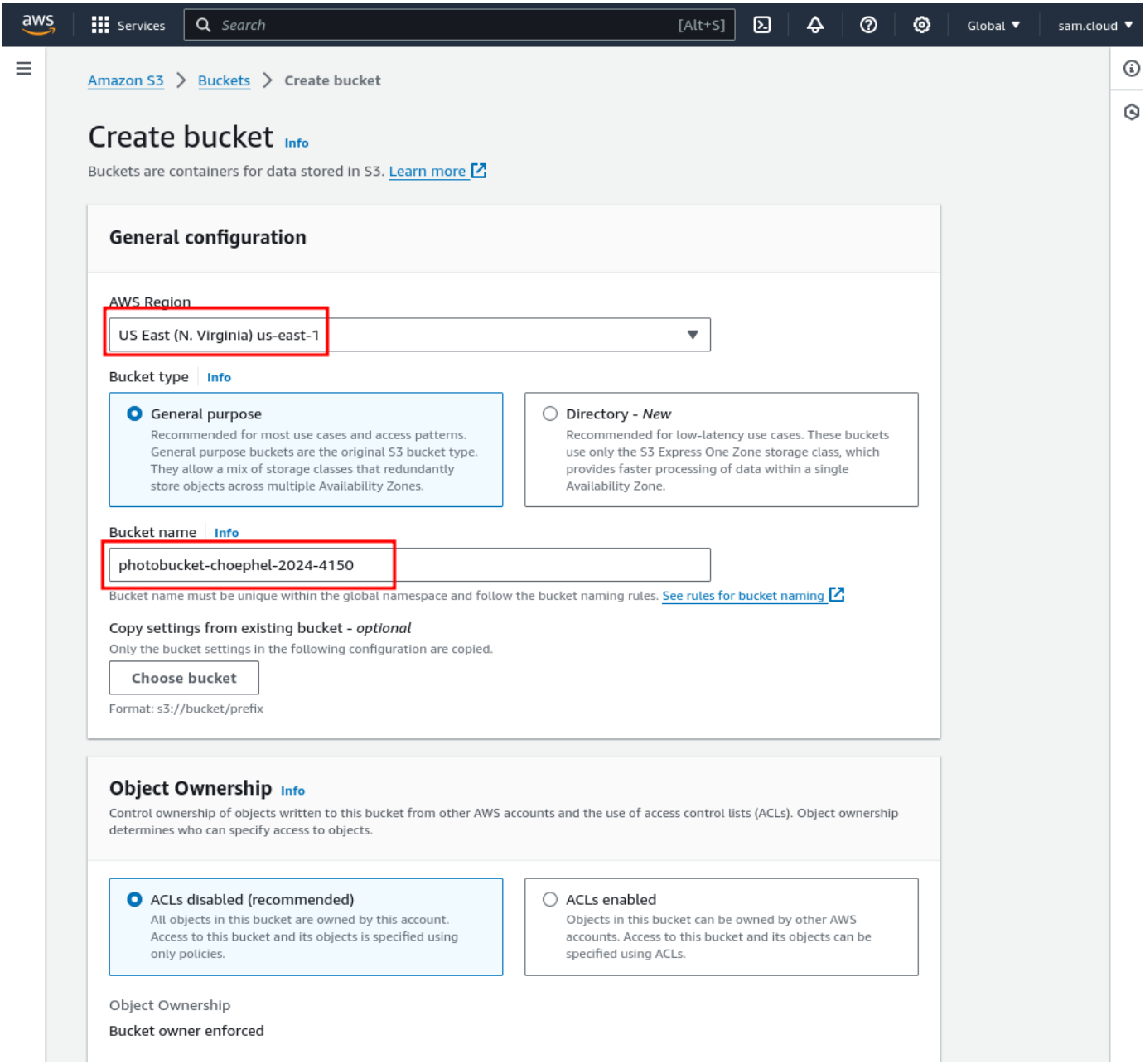
- Create a **DynamoDB** table named **PhotoGallery** with a partition key and a sort key named **albumID** and **photoID**, respectively, as shown below. Click **Create** (keep the other configuration to default as it is).





5. Create an S3 Bucket for storing photos (2 points)

- Create a new S3 bucket for storing photos. **Uncheck Block all public access under Bucket setting for Block Public Access section**
- For the name of the bucket, use **photobucket-lastname-year-courseNumber** (e.g., **photobucket-choephel-2024-4150**)


aws
Services
Search
[Alt+S]
Global
sam.cloud
Amazon S3 > Buckets > Create bucket
Create bucket Info
Buckets are containers for data stored in S3. Learn more
General configuration
AWS Region
US East (N. Virginia) us-east-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.
Directory - New
Recommended for low-latency use cases. These buckets use only the S3 Express One Zone storage class, which provides faster processing of data within a single Availability Zone.
Bucket name Info
photobucket-choephel-2024-4150
Bucket name must be unique within the global namespace and follow the bucket naming rules. See rules for bucket naming
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 control lists (ACLs). Object ownership determines who can specify access to objects.
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.
ACLs enabled
Objects in this bucket can be owned by other AWS accounts. Access to this bucket and its objects can be specified using ACLs.
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, or all. In order to ensure that public access to this bucket and its objects is blocked, turn on Block all public access. These settings apply only to this bucket and its access points. AWS recommends that you turn on Block all public access, but before applying any of these settings, ensure that your applications will work correctly without public access. If you require some level of public access to this bucket or objects within, you can customize the individual settings below to suit your specific storage use cases. Learn more

- [ ] **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 another.
  - [ ] **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 ACLs for existing buckets and objects. This setting doesn't change any existing permissions that allow public access to S3 resources using ACLs.
  - [ ] **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.
  - [ ] **Block public access to buckets and objects granted through *new* public bucket or access point policies**
    S3 will block new bucket and access point policies that grant public access to buckets and objects. This setting doesn't change any existing policies that allow public access to S3 resources.
  - [ ] **Block public and cross-account access to buckets and objects through *any* public bucket or access point policies**
    S3 will ignore public and cross-account access for buckets or access points with policies that grant public access to buckets and objects.

⚠ **Turning off block all public access might result in this bucket and the objects within becoming public**
AWS recommends that you turn on block all public access, unless public access is required for specific and verified use cases such as static website hosting.

- [x] I acknowledge that the current settings might result in this bucket and the objects within becoming public.

Bucket Versioning

Versioning is a means of keeping multiple variants of an object in the same bucket. You can use versioning to preserve, retrieve, and restore every version of every object stored in your Amazon S3 bucket. With versioning, you can easily recover from both unintended user actions and application failures. Learn more

Bucket Versioning

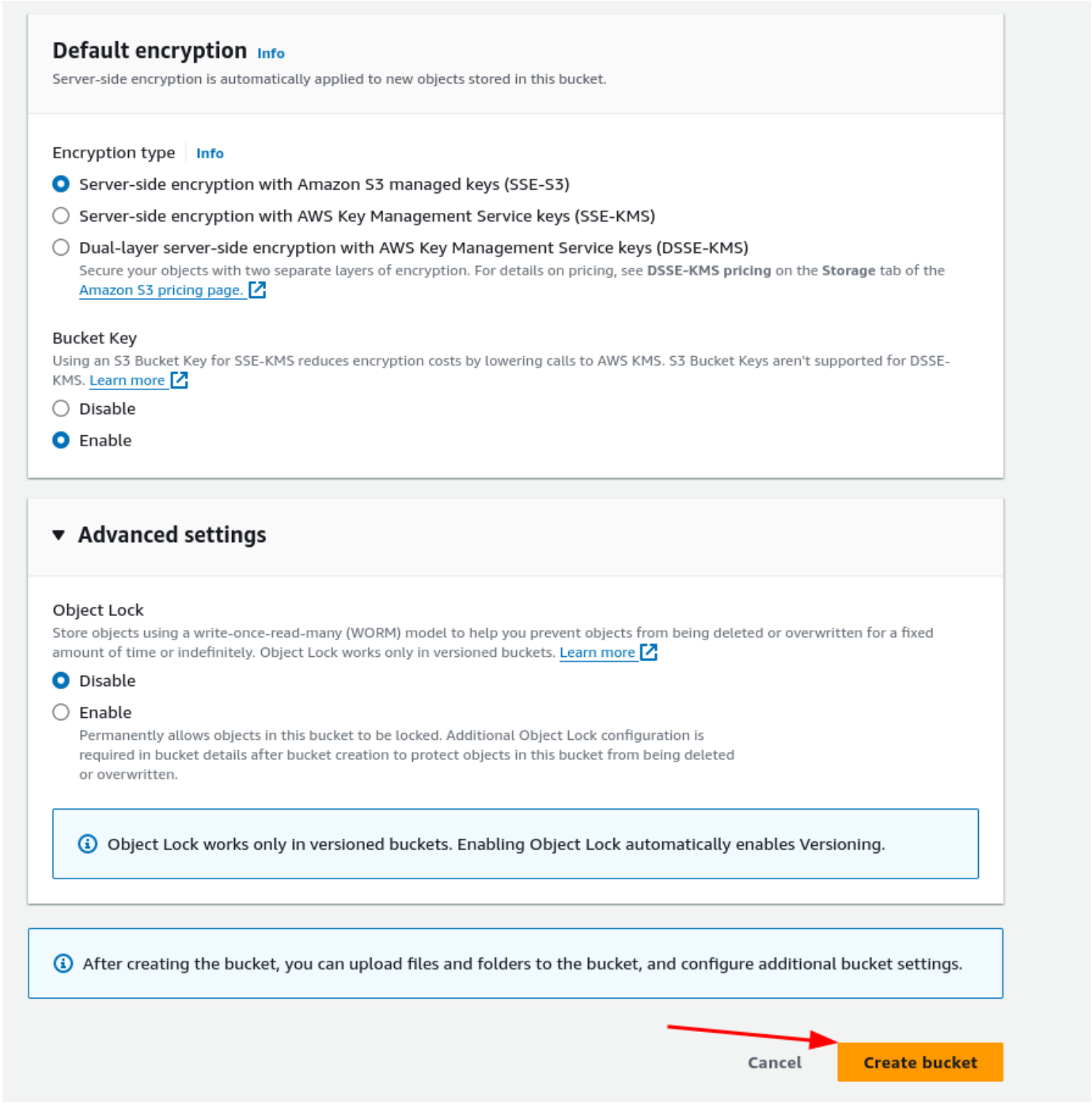
- (•) Disable
- ( ) Enable

Tags - *optional* (0)

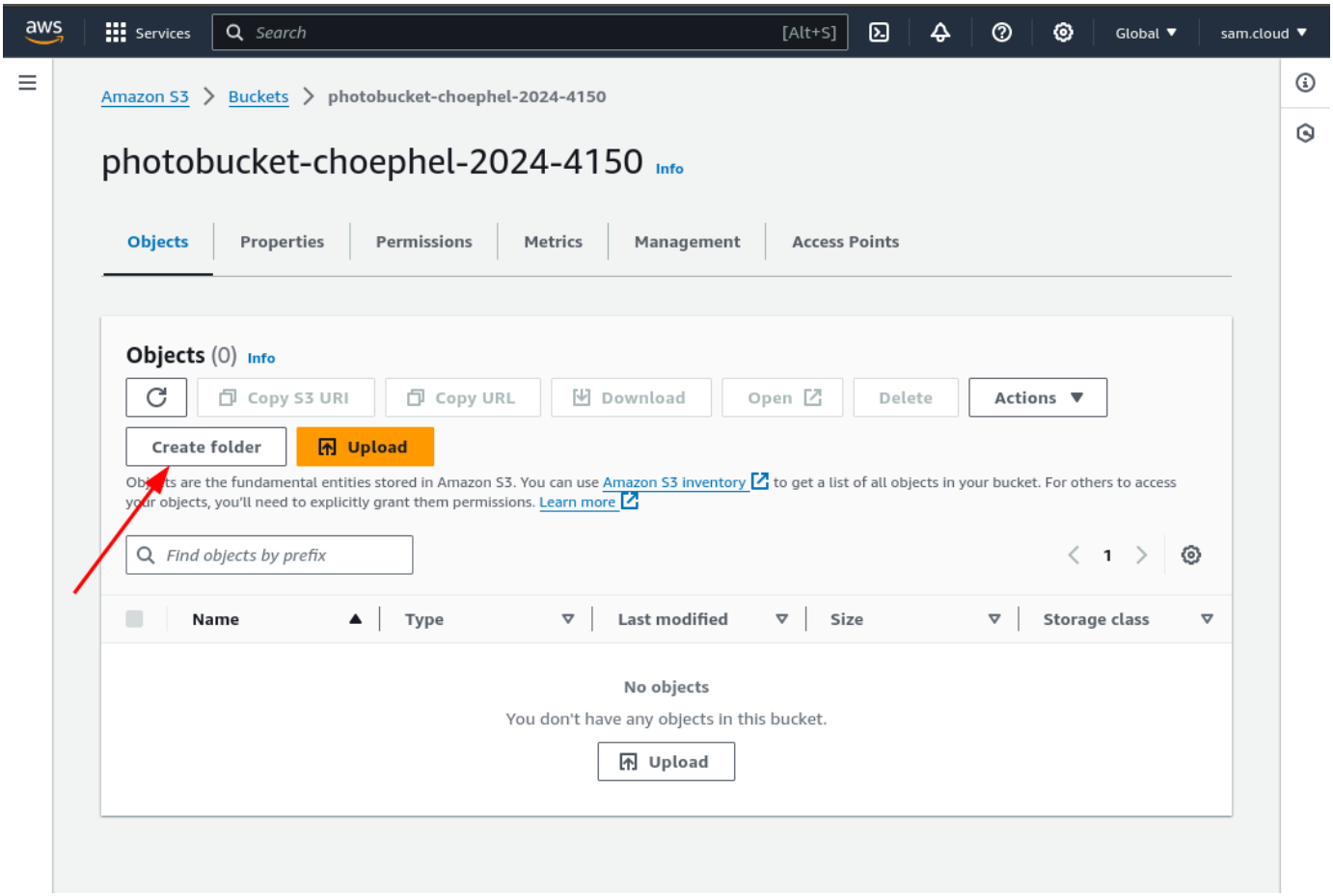
You can use bucket tags to track storage costs and organize buckets. Learn more

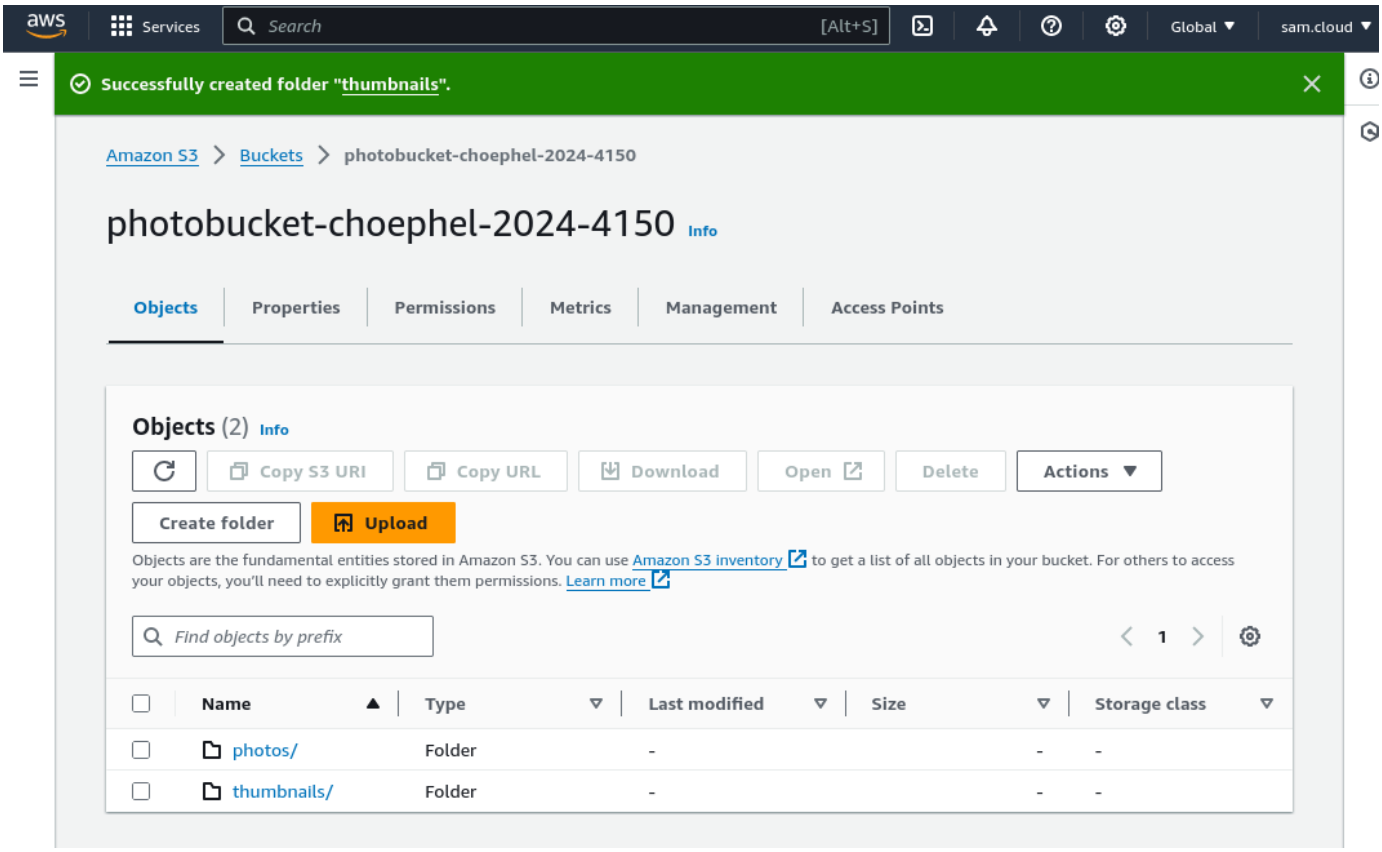
No tags associated with this bucket.

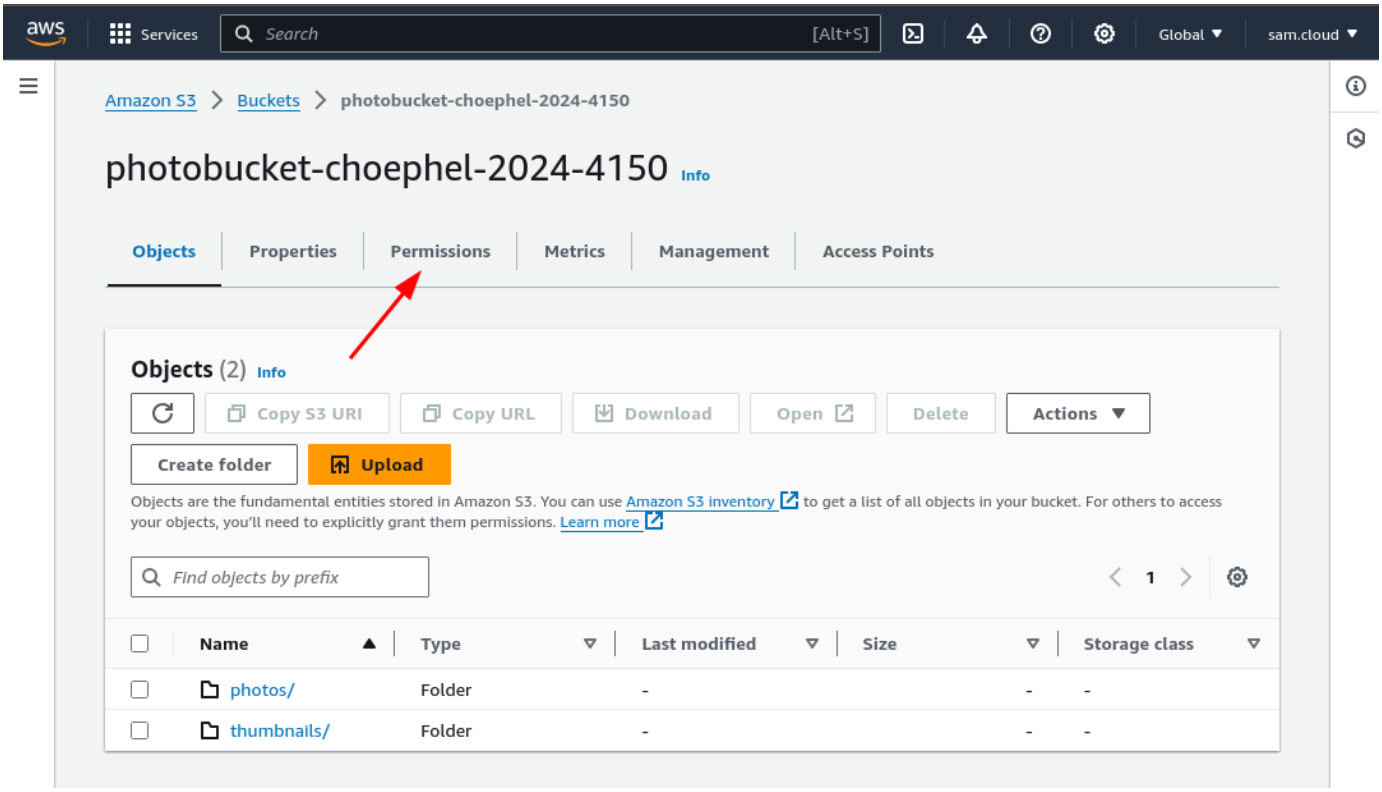
Add tag



- Create two folders in this bucket. One called '**photos**' and another one called '**thumbnails**'


aws
Services
Search
[Alt+S]
Global
sam.cloud
Amazon S3 > Buckets > photobucket-choephel-2024-4150
photobucket-choephel-2024-4150 Info
Objects
Properties
Permissions
Metrics
Management
Access Points
Objects (0) Info
Copy S3 URI
Copy URL
Download
Open
Delete
Actions
Create folder
Upload
Objects are the fundamental entities stored in Amazon S3. You can use Amazon S3 inventory to get a list of all objects in your bucket. For others to access your objects, you'll need to explicitly grant them permissions. Learn more
Find objects by prefix
1
Name
Type
Last modified
Size
Storage class
No objects
You don't have any objects in this bucket.
Upload


aws
Services
Search
[Alt+S]
Global
sam.cloud
Successfully created folder "thumbnails".
Amazon S3 > Buckets > photobucket-choephel-2024-4150
photobucket-choephel-2024-4150 Info
Objects
Properties
Permissions
Metrics
Management
Access Points
Objects (2) Info
Copy S3 URI
Copy URL
Download
Open
Delete
Actions
Create folder
Upload
Objects are the fundamental entities stored in Amazon S3. You can use Amazon S3 inventory to get a list of all objects in your bucket. For others to access your objects, you'll need to explicitly grant them permissions. Learn more
Find objects by prefix
1
Name
Type
Last modified
Size
Storage class
photos/
Folder
-
-
-
thumbnails/
Folder
-
-
-



- Add a bucket policy below to enable public access to the photos uploaded. Replace '**lastname**' with your lastname of the S3 bucket created

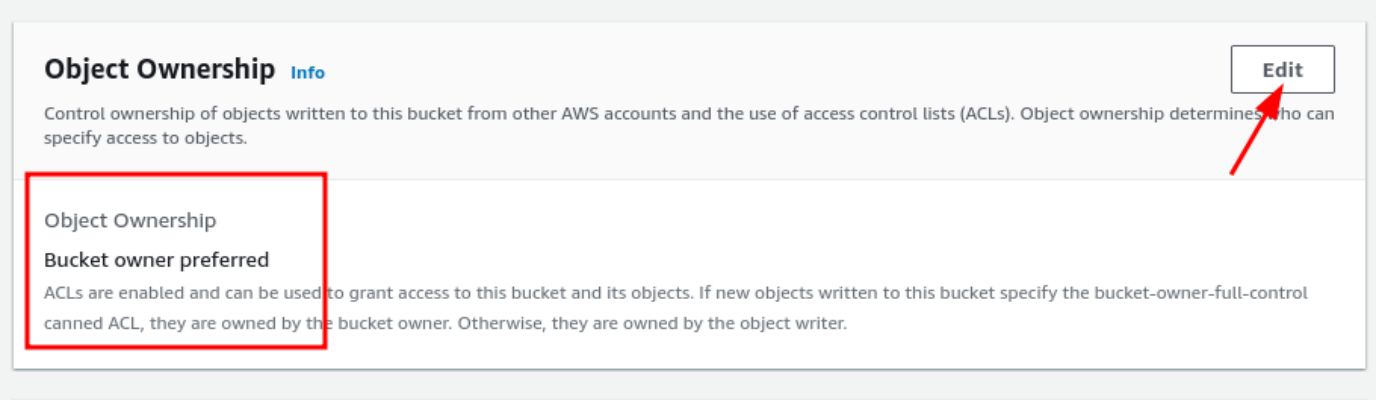
```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "PublicReadGetObject",
            "Effect": "Allow",
            "Principal": "*",
            "Action": "s3:GetObject",
            "Resource": "arn:aws:s3:::photobucket-lastname-2024-
4150/photos/*"
        }
    ]
}
```

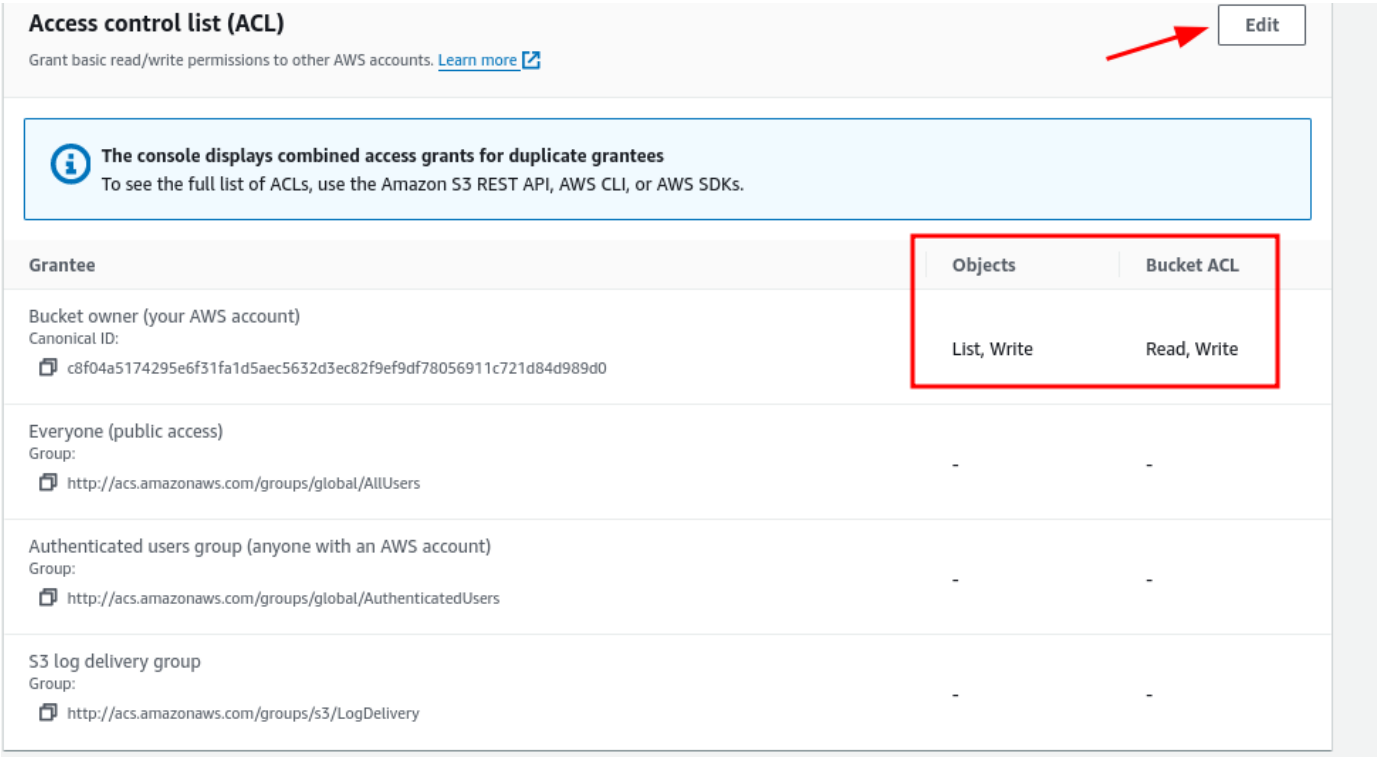
- **Cross-origin resource sharing (CORS)**

```
[
    {
        "AllowedHeaders": [
            "Authorization"
        ],
        "AllowedMethods": [
            "GET"
        ],
```

```
        "AllowedOrigins": [
            "*"
        ],
        "ExposeHeaders": []
    }
]
```

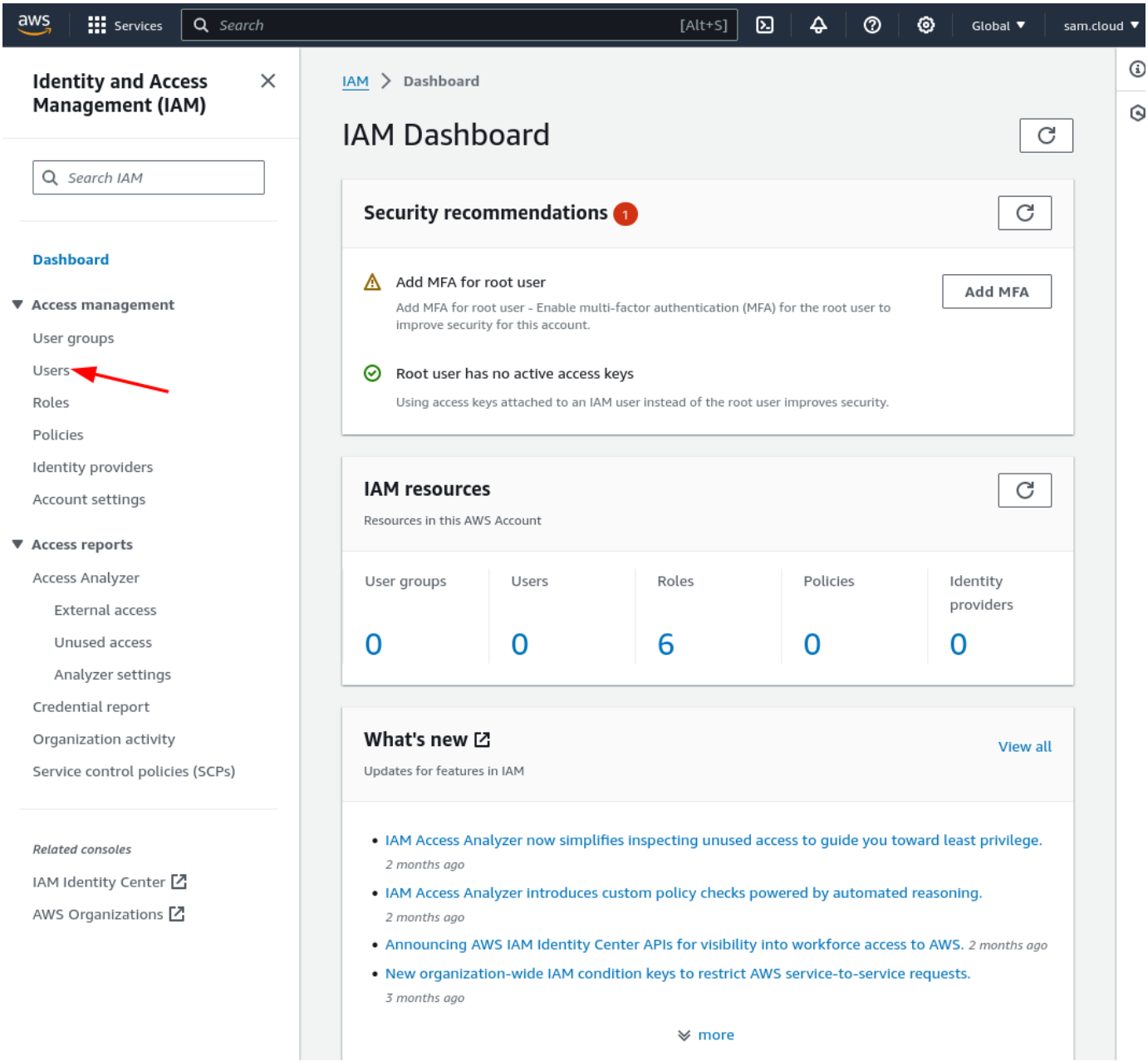
Double Check your S3 Bucket config

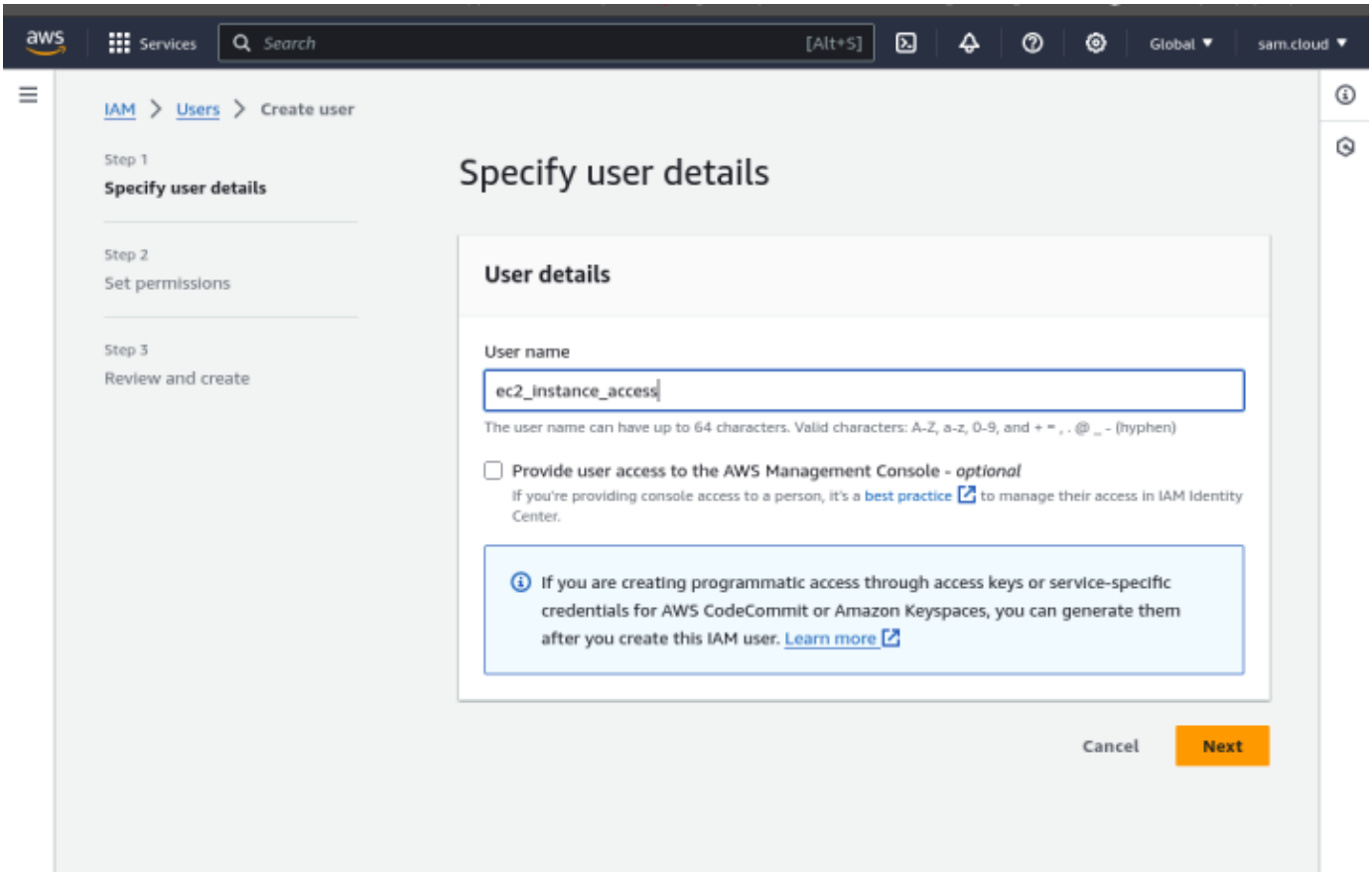


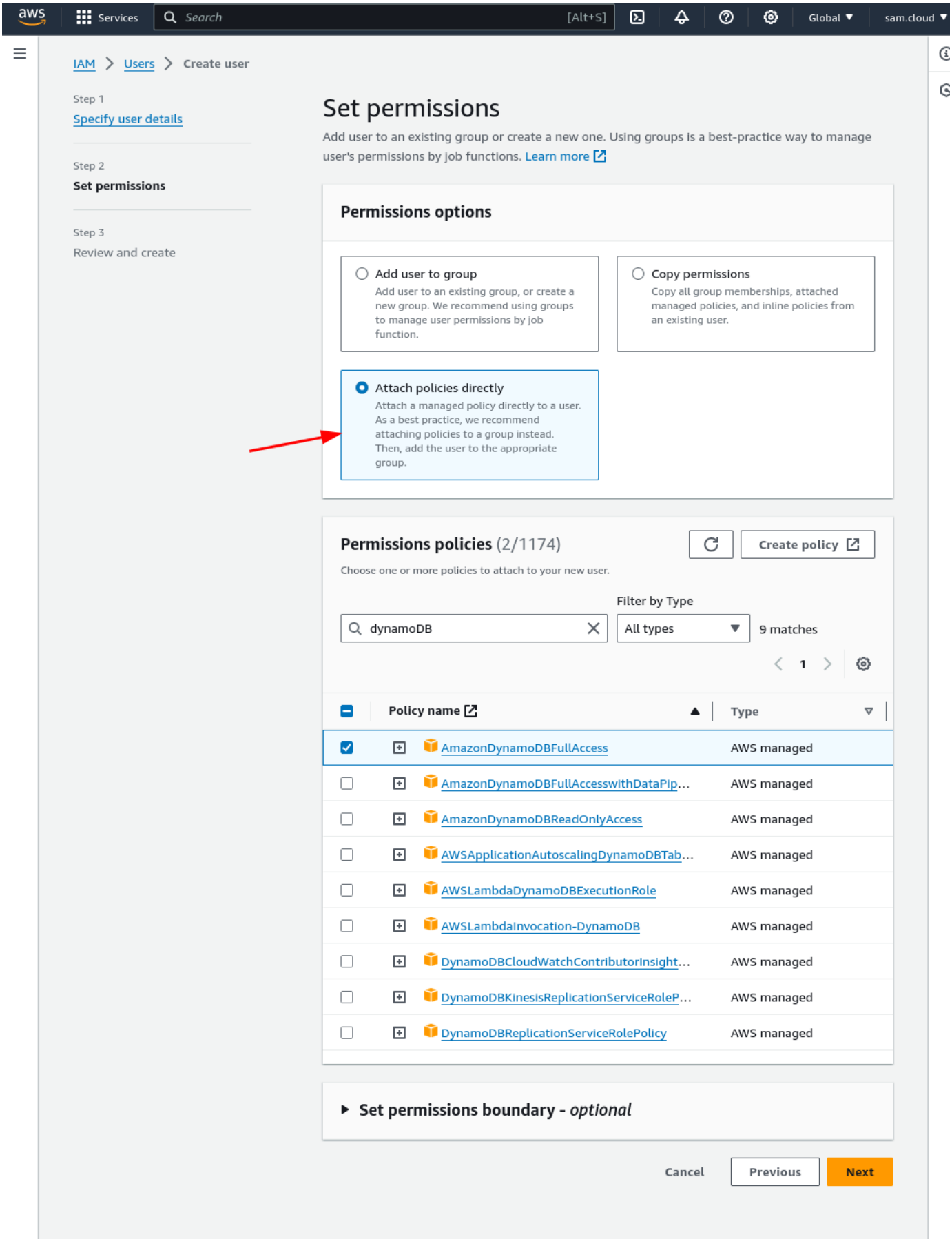


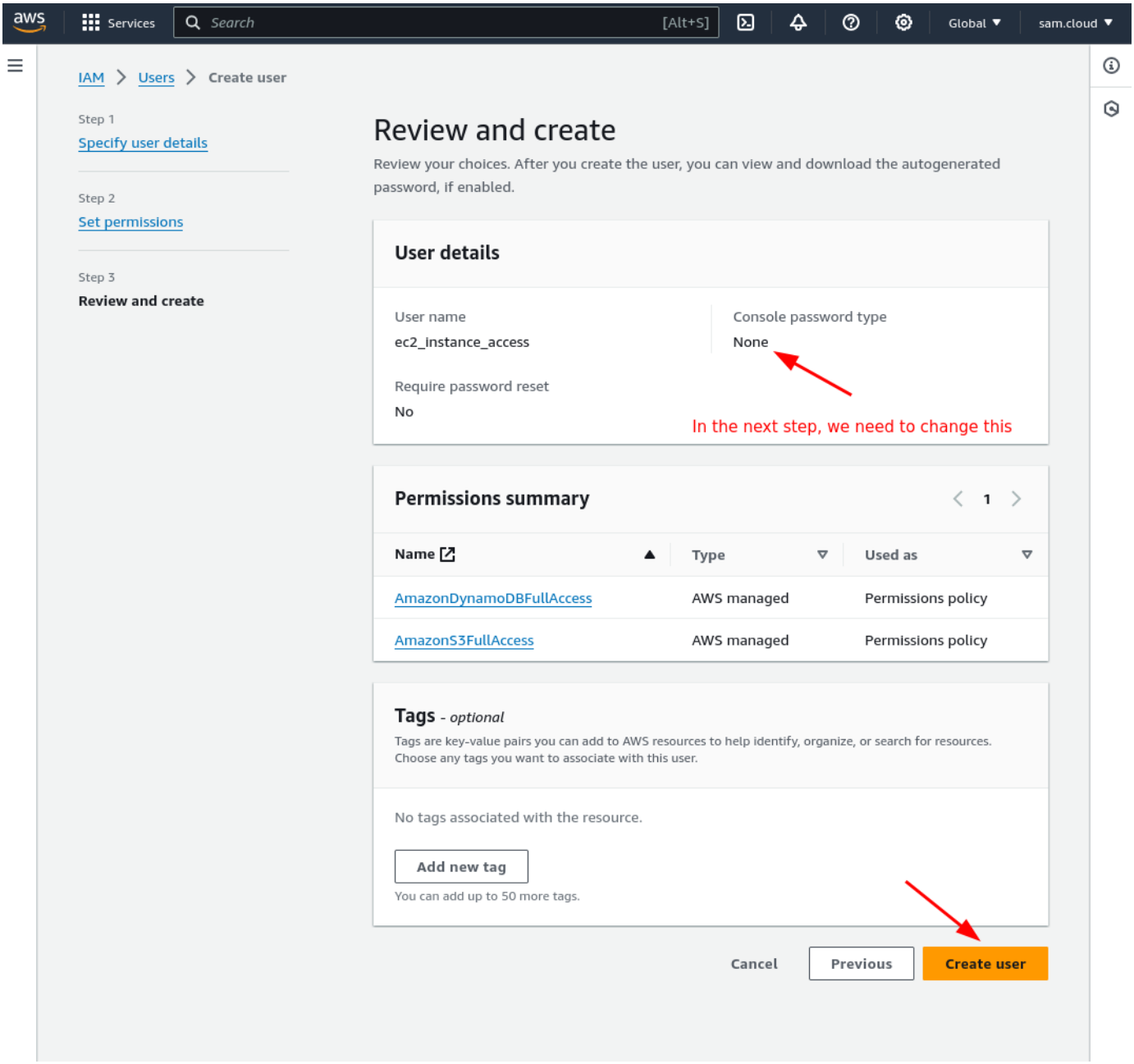
6. Create IAM User (2 points)

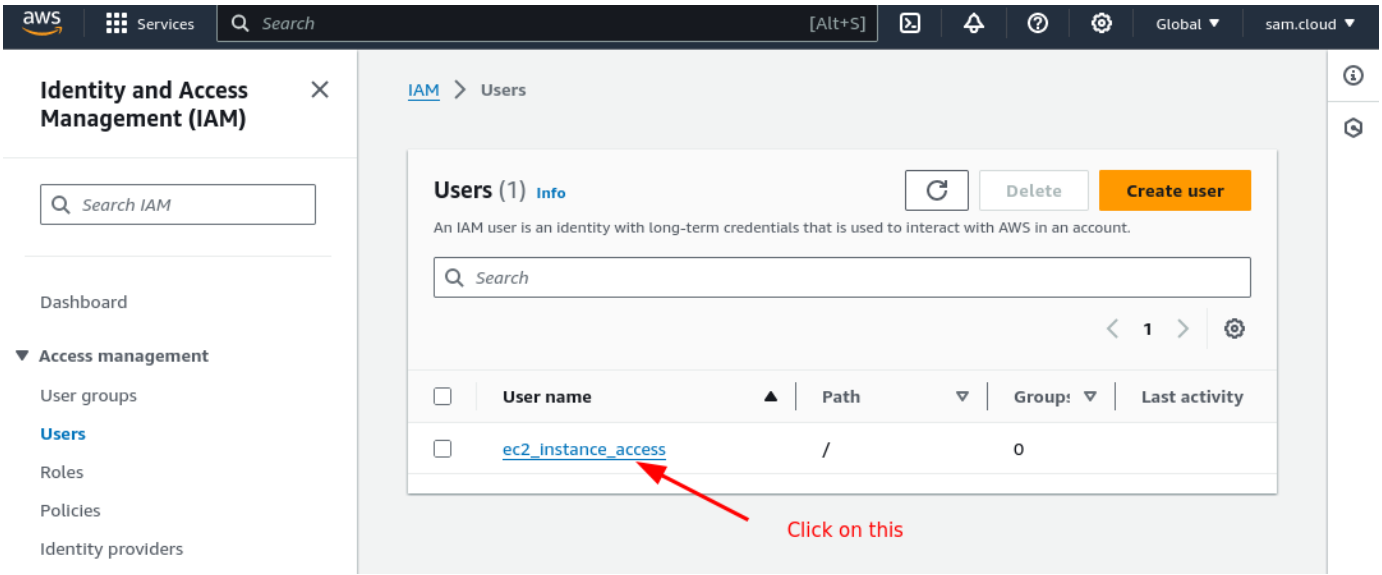
In the AWS IAM console, create a new IAM user called **ec2_instance_access** for the EC2 instance to S3 service, as shown below. Make sure that the Access Type is only **programmatic**. Attach policy called **AmazonS3FullAccess** and **AmazonDynamoDBFullAccess** to this user. Save the **Access key ID** and **Secret access key**

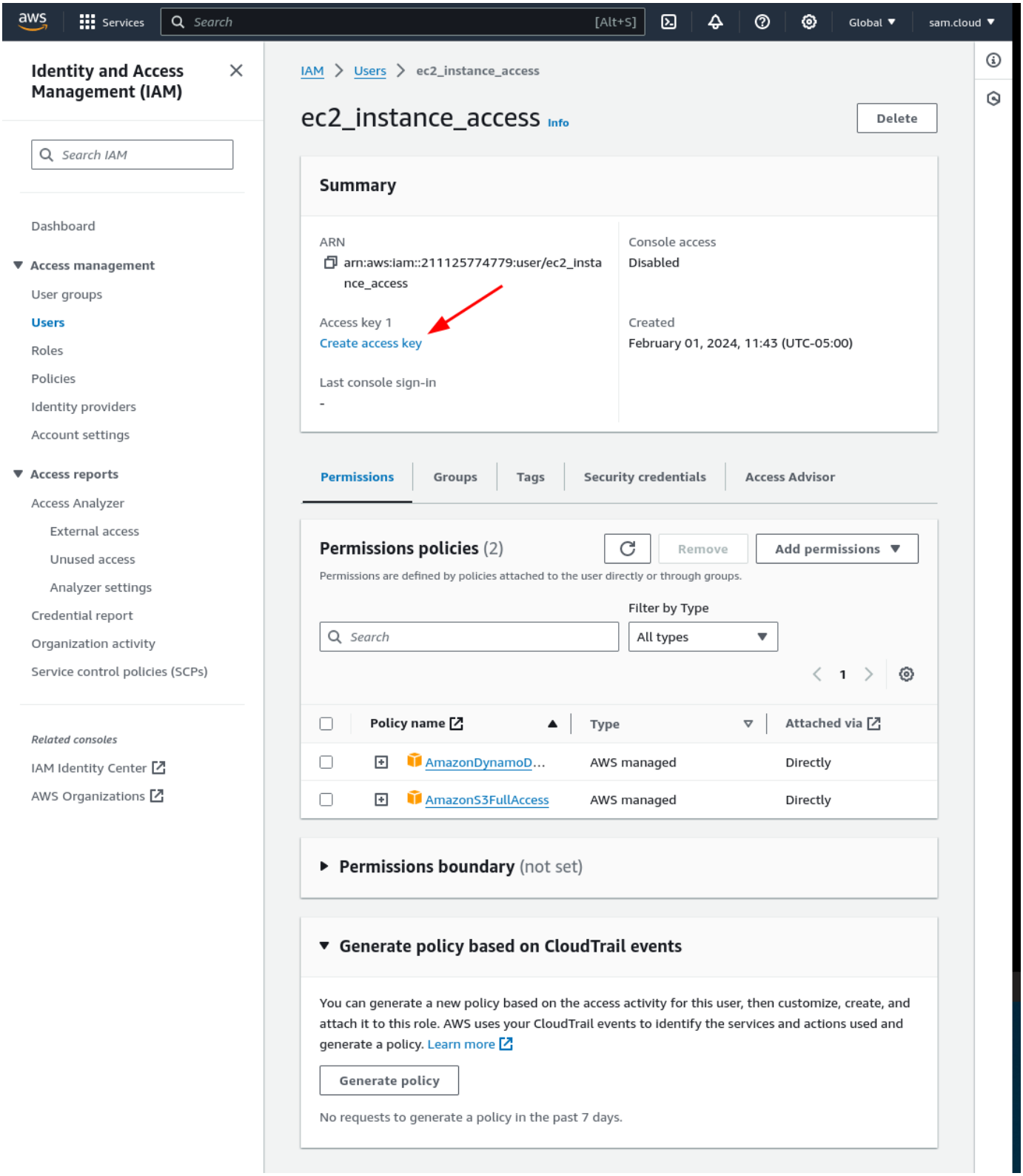

aws
Services
Search
[Alt+S]
Global
sam.cloud
Identity and Access Management (IAM)
Search IAM
Dashboard
Access management
User groups
Users
Roles
Policies
Identity providers
Account settings
Access reports
Access Analyzer
External access
Unused access
Analyzer settings
Credential report
Organization activity
Service control policies (SCPs)
Related consoles
IAM Identity Center
AWS Organizations
IAM > Dashboard
IAM Dashboard
Security recommendations 1
Add MFA for root user
Add MFA for root user - Enable multi-factor authentication (MFA) for the root user to improve security for this account.
Add MFA
Root user has no active access keys
Using access keys attached to an IAM user instead of the root user improves security.
IAM resources
Resources in this AWS Account
User groups
0
Users
0
Roles
6
Policies
0
Identity providers
0
What's new
Updates for features in IAM
View all
IAM Access Analyzer now simplifies inspecting unused access to guide you toward least privilege.
2 months ago
IAM Access Analyzer introduces custom policy checks powered by automated reasoning.
2 months ago
Announcing AWS IAM Identity Center APIs for visibility into workforce access to AWS. 2 months ago
New organization-wide IAM condition keys to restrict AWS service-to-service requests.
3 months ago
more

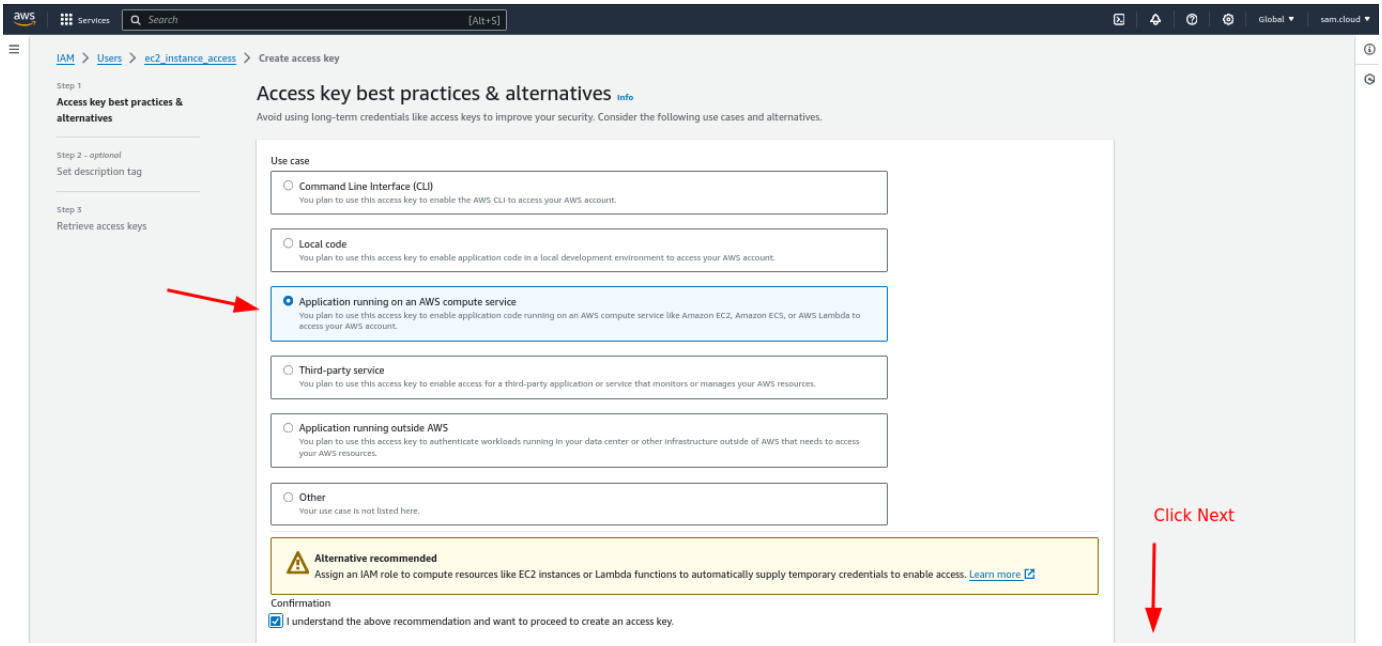

aws
Services
Search
[Alt+S]
Global
sam.cloud
IAM > Users > Create user
Step 1
Specify user details
Step 2
Set permissions
Step 3
Review and create
Specify user details
User details
User name
ec2_instance_access
The user name can have up to 64 characters. Valid characters: A-Z, a-z, 0-9, and + = , . @ _ - (hyphen)
Provide user access to the AWS Management Console - optional
If you're providing console access to a person, it's a best practice to manage their access in IAM Identity Center.
If you are creating programmatic access through access keys or service-specific credentials for AWS CodeCommit or Amazon Keyspaces, you can generate them after you create this IAM user. Learn more
Cancel
Next

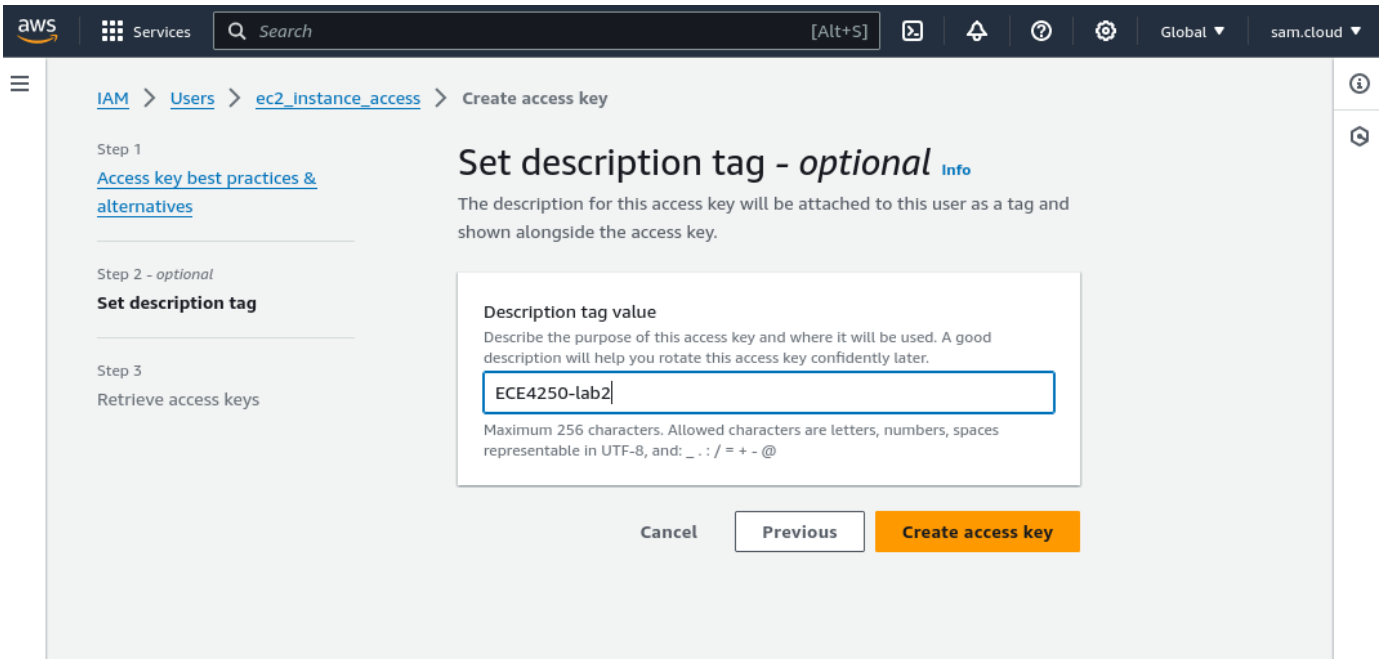

aws
Services
Search
[Alt+S]
Global
sam.cloud
IAM > Users > Create user
Step 1
Specify user details
Step 2
Set permissions
Step 3
Review and create
Set permissions
Add user to an existing group or create a new one. Using groups is a best-practice way to manage user's permissions by job functions. Learn more
Permissions options
Add user to group
Add user to an existing group, or create a new group. We recommend using groups to manage user permissions by job function.
Copy permissions
Copy all group memberships, attached managed policies, and inline policies from an existing user.
Attach policies directly
Attach a managed policy directly to a user. As a best practice, we recommend attaching policies to a group instead. Then, add the user to the appropriate group.
Permissions policies (2/1174)
Create policy
Choose one or more policies to attach to your new user.
Filter by Type
dynamoDB
All types
9 matches
1
Policy name
Type
AmazonDynamoDBFullAccess
AWS managed
AmazonDynamoDBFullAccesswithDataPip...
AWS managed
AmazonDynamoDBReadOnlyAccess
AWS managed
AWSApplicationAutoscalingDynamoDBTab...
AWS managed
AWSLambdaDynamoDBExecutionRole
AWS managed
AWSLambdaInvocation-DynamoDB
AWS managed
DynamoDBCloudWatchContributorInsight...
AWS managed
DynamoDBKinesisReplicationServiceRoleP...
AWS managed
DynamoDBReplicationServiceRolePolicy
AWS managed
Set permissions boundary - optional
Cancel
Previous
Next

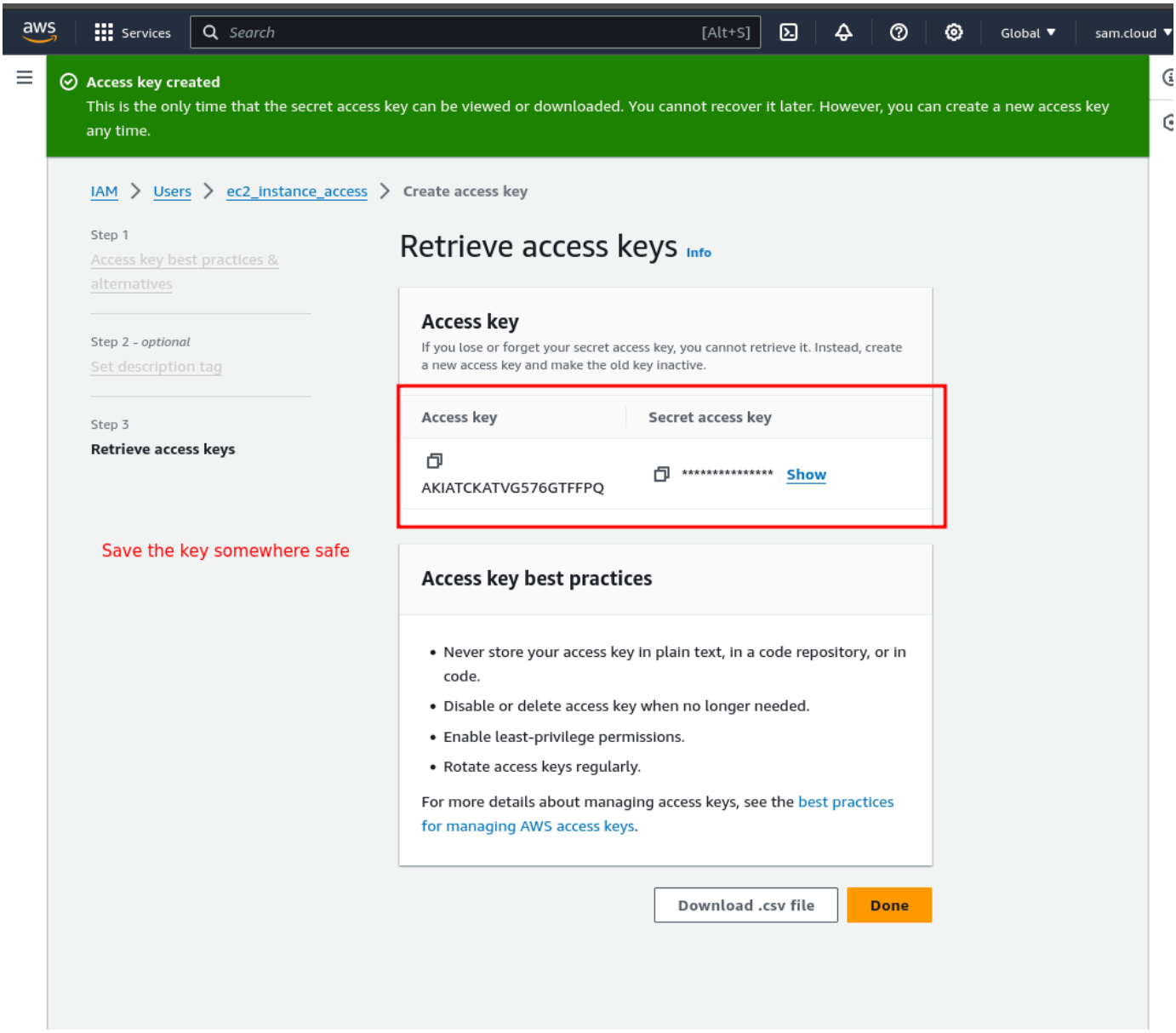

aws
Services
Search
[Alt+S]
Global
sam.cloud
IAM > Users > Create user
Step 1
Specify user details
Step 2
Set permissions
Step 3
Review and create
Review and create
Review your choices. After you create the user, you can view and download the autogenerated password, if enabled.
User details
User name
ec2_instance_access
Console password type
None
Require password reset
No
In the next step, we need to change this
Permissions summary
1
Name
Type
Used as
AmazonDynamoDBFullAccess
AWS managed
Permissions policy
AmazonS3FullAccess
AWS managed
Permissions policy
Tags - optional
Tags are key-value pairs you can add to AWS resources to help identify, organize, or search for resources. Choose any tags you want to associate with this user.
No tags associated with the resource.
Add new tag
You can add up to 50 more tags.
Cancel
Previous
Create user


aws
Services
Search
[Alt+S]
Global
sam.cloud
Identity and Access Management (IAM)
Search IAM
Dashboard
Access management
User groups
Users
Roles
Policies
Identity providers
IAM > Users
Users (1) Info
Delete
Create user
An IAM user is an identity with long-term credentials that is used to interact with AWS in an account.
Search
1
User name
Path
Group:
Last activity
ec2_instance_access
/
0
Click on this


aws
Services
Search
[Alt+S]
Global
sam.cloud
Identity and Access Management (IAM)
Search IAM
Dashboard
Access management
User groups
Users
Roles
Policies
Identity providers
Account settings
Access reports
Access Analyzer
External access
Unused access
Analyzer settings
Credential report
Organization activity
Service control policies (SCPs)
Related consoles
IAM Identity Center
AWS Organizations
IAM > Users > ec2_instance_access
ec2_instance_access Info
Delete
Summary
ARN
arn:aws:iam::211125774779:user/ec2_instance_access
Console access
Disabled
Access key 1
Create access key
Created
February 01, 2024, 11:43 (UTC-05:00)
Last console sign-in
-
Permissions
Groups
Tags
Security credentials
Access Advisor
Permissions policies (2)
Remove
Add permissions
Permissions are defined by policies attached to the user directly or through groups.
Filter by Type
Search
All types
1
Policy name
Type
Attached via
AmazonDynamoD...
AWS managed
Directly
AmazonS3FullAccess
AWS managed
Directly
Permissions boundary (not set)
Generate policy based on CloudTrail events
You can generate a new policy based on the access activity for this user, then customize, create, and attach it to this role. AWS uses your CloudTrail events to identify the services and actions used and generate a policy. Learn more
Generate policy
No requests to generate a policy in the past 7 days.

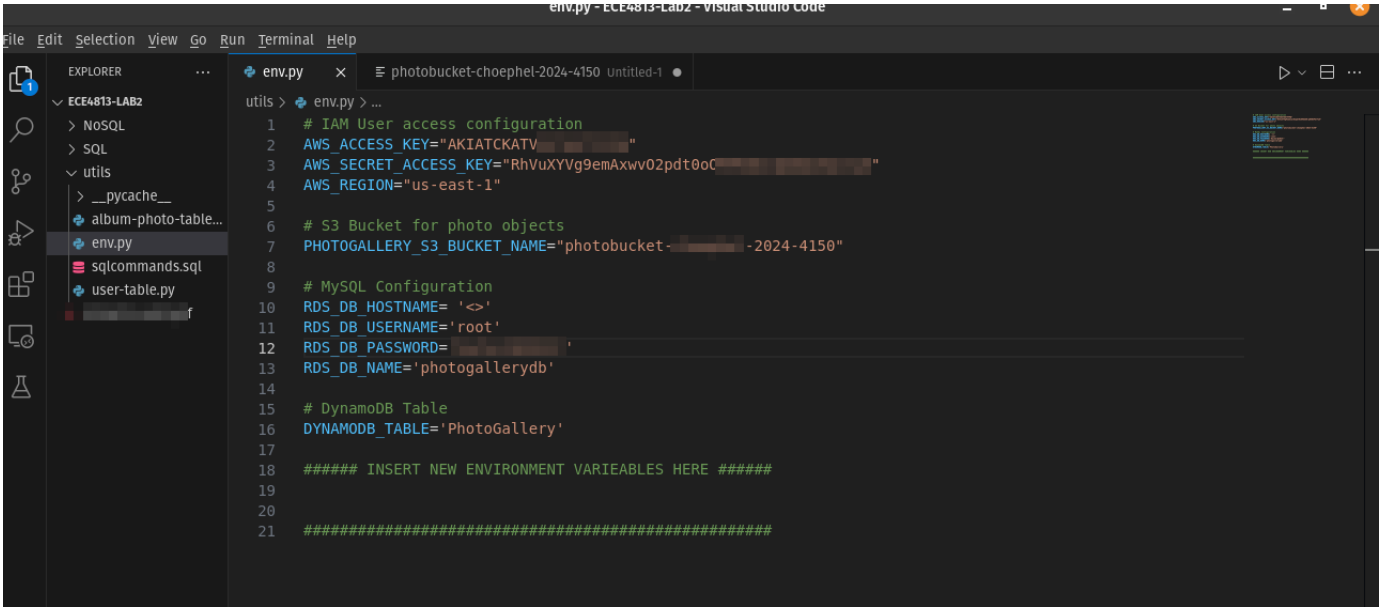

IAM > Users > ec2_instance_access > Create access key
Step 1
Access key best practices & alternatives
Step 2 - optional
Set description tag
Step 3
Retrieve access keys
Access key best practices & alternatives Info
Avoid using long-term credentials like access keys to improve your security. Consider the following use cases and alternatives.
Use case
Command Line Interface (CLI)
You plan to use this access key to enable the AWS CLI to access your AWS account.
Local code
You plan to use this access key to enable application code in a local development environment to access your AWS account.
Application running on an AWS compute service
You plan to use this access key to enable application code running on an AWS compute service like Amazon EC2, Amazon ECS, or AWS Lambda to access your AWS account.
Third-party service
You plan to use this access key to enable access for a third-party application or service that monitors or manages your AWS resources.
Application running outside AWS
You plan to use this access key to authenticate workloads running in your data center or other infrastructure outside of AWS that needs to access your AWS resources.
Other
Your use case is not listed here.
Alternative recommended
Assign an IAM role to compute resources like EC2 instances or Lambda functions to automatically supply temporary credentials to enable access. Learn more
Confirmation
I understand the above recommendation and want to proceed to create an access key.
Click Next


aws
Services
Search
[Alt+S]
Global
sam.cloud
IAM > Users > ec2_instance_access > Create access key
Step 1
Access key best practices & alternatives
Step 2 - optional
Set description tag
Step 3
Retrieve access keys
Set description tag - optional Info
The description for this access key will be attached to this user as a tag and shown alongside the access key.
Description tag value
Describe the purpose of this access key and where it will be used. A good description will help you rotate this access key confidently later.
ECE4250-lab2
Maximum 256 characters. Allowed characters are letters, numbers, spaces representable in UTF-8, and: _ . : / = + - @
Cancel
Previous
Create access key



7. Update env.py script and move the entire code directory to the EC2 instance (2 points)

- In the les provided, locate the **env.py** script under the **utils** directory and update all the variables required to run the Photo Gallery Application using the two variants.
- **DO NOT PLACE THE NAME INSIDE THE ANGLE BRACKETS.** For example, if the name of the S3 bucket to store photos is photobucket-choephel-2024-4150 replace **<PHOTOGALLERY_S3_BUCKET_NAME>** for the name of the bucket.



8. Transferring Files between your computer to and Amazon EC2 instance (2 points)

- For linux/Unix/Mac system, we can use a command-line tool “scp” (secure copy) to transfer files between your laptop and Amazon instance. With scp you can copy files between computers on a network.
- To upload a le from your computer to the EC2 instance:
  `$scp -i amazonkey.pem lab2/code.py ubuntu@ec2-host.amazonaws.com:~/data/`
- To upload a directory from your computer to the EC2 instance:

```
$scp -i amazonkey.pem -r lab2 ubuntu@ec2-host.amazonaws.com:~/data/
```

- To download a le from the EC2 instance to your computer to:

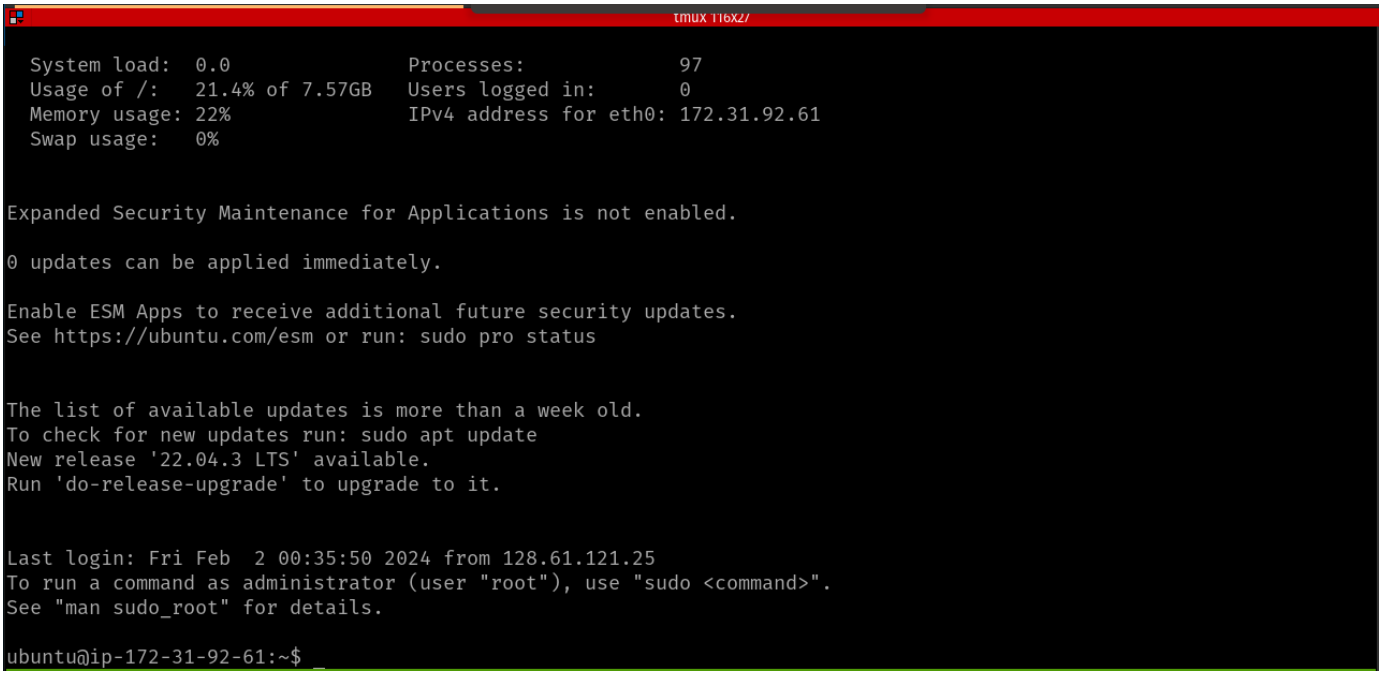
```
$scp -i amazonkey.pem ubuntu@ec2-host.amazonaws.com:/data/code.py ~/Download
```

- To download a directory from your computer to the EC2 instance:

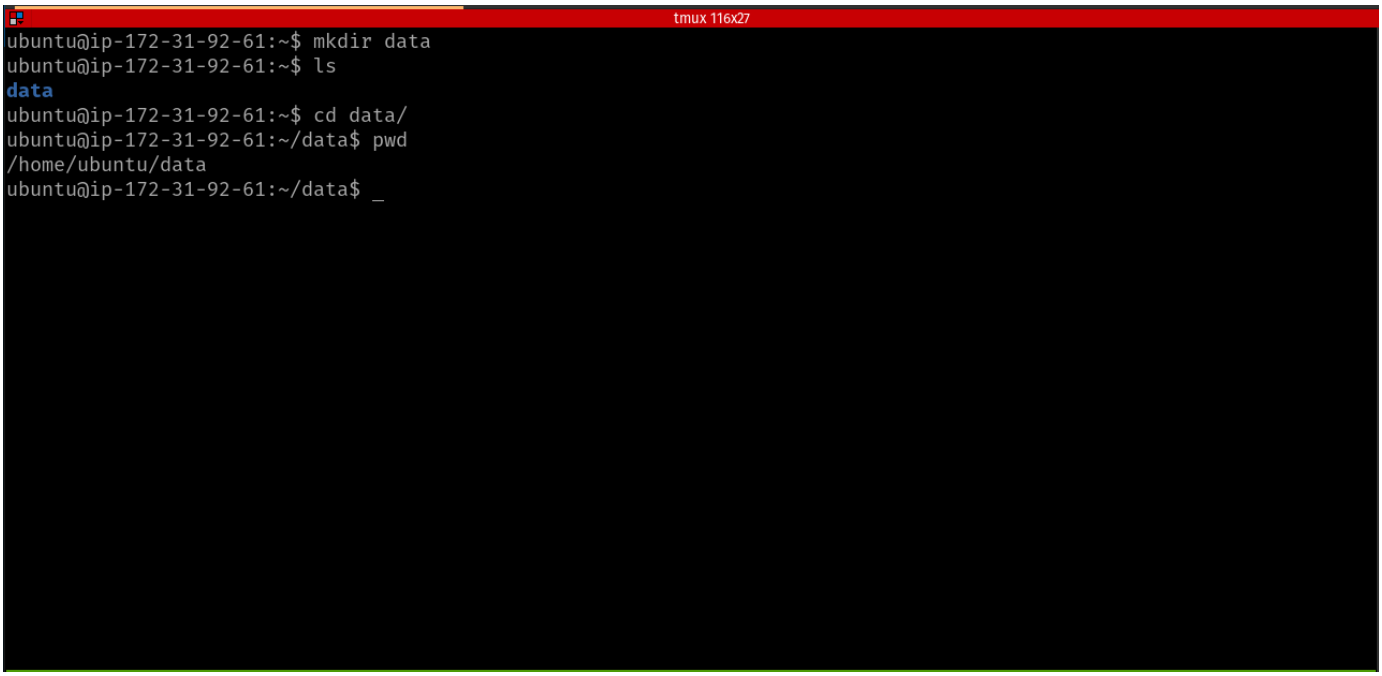
```
$scp -i amazonkey.pem -r ubuntu@ec2-host.amazonaws.com:/data/lab2 ~/Download
```

- If you want a more user-friendly tool to transfer data, FileZilla is the right choice. It is free, supports Windows/Linux/Mac systems, and has a friendly user interface. It supports FTP, SFTP, and other le transfer protocols. Go to https:// lezilla-project.org/ to download it (You can either visit their official website for the manual or a few minutes of Youtube video could help you how to use it).

```
ssh -i ece4150.pem ubuntu@44.204.54.182
```



Create a folder **data**, in ec2



Upload NoSQL

```
scp -i ~/Downloads/ece4150.pem -r NoSQL
ubuntu@44.204.54.182:/home/ubuntu/data/
```

```
➜  ECE4150-Lab2 ls
ECE4813-Lab2.pdf  NoSQL  SQL  test  utils
➜  ECE4150-Lab2 scp -i ~/Downloads/ece4150.pem -r NoSQL ubuntu@44.204.54.182:/home/ubuntu/data/
index.html                                    100% 3948   127.6KB/s   00:00
viewphotos.html                               100% 4532   148.5KB/s   00:00
photoForm.html                                100% 3370   111.1KB/s   00:00
searchPhoto.html                              100% 4570   146.5KB/s   00:00
albumForm.html                                100% 3150   102.9KB/s   00:00
searchAlbum.html                              100% 3866   126.3KB/s   00:00
photodetail.html                              100% 6028   194.9KB/s   00:00
.DS_Store                                     100% 6148   199.5KB/s   00:00
app.py                                        100%   11KB 370.3KB/s   00:00
.DS_Store                                     100% 6148   197.7KB/s   00:00
.DS_Store                                     100% 6148   196.6KB/s   00:00
Class 12-J Pic.JPG                            100%  176KB   1.3MB/s   00:00
Canon_PowerShot_S40.jpg                       100%   32KB 987.1KB/s   00:00
image9.jpg                                    100%  122KB   3.0MB/s   00:00
kjyvq6i.jpg                                   100%  111KB   2.8MB/s   00:00
43900614915_60e9e32b2c_o.jpg                  100% 6054KB   7.9MB/s   00:00
andreas-gucklhorn-HmbepSBAHUU-unsplash.jpg    100% 1344KB   6.5MB/s   00:00
inputho-out.jpg                               100%  228KB   3.5MB/s   00:00
image5.jpg                                    100%  128KB   3.0MB/s   00:00
Farewell Pic.jpg                              100%  149KB   3.3MB/s   00:00
```

Upload SQL

```
scp -i ~/Downloads/ece4150.pem -r SQL
ubuntu@44.204.54.182:/home/ubuntu/data/
```

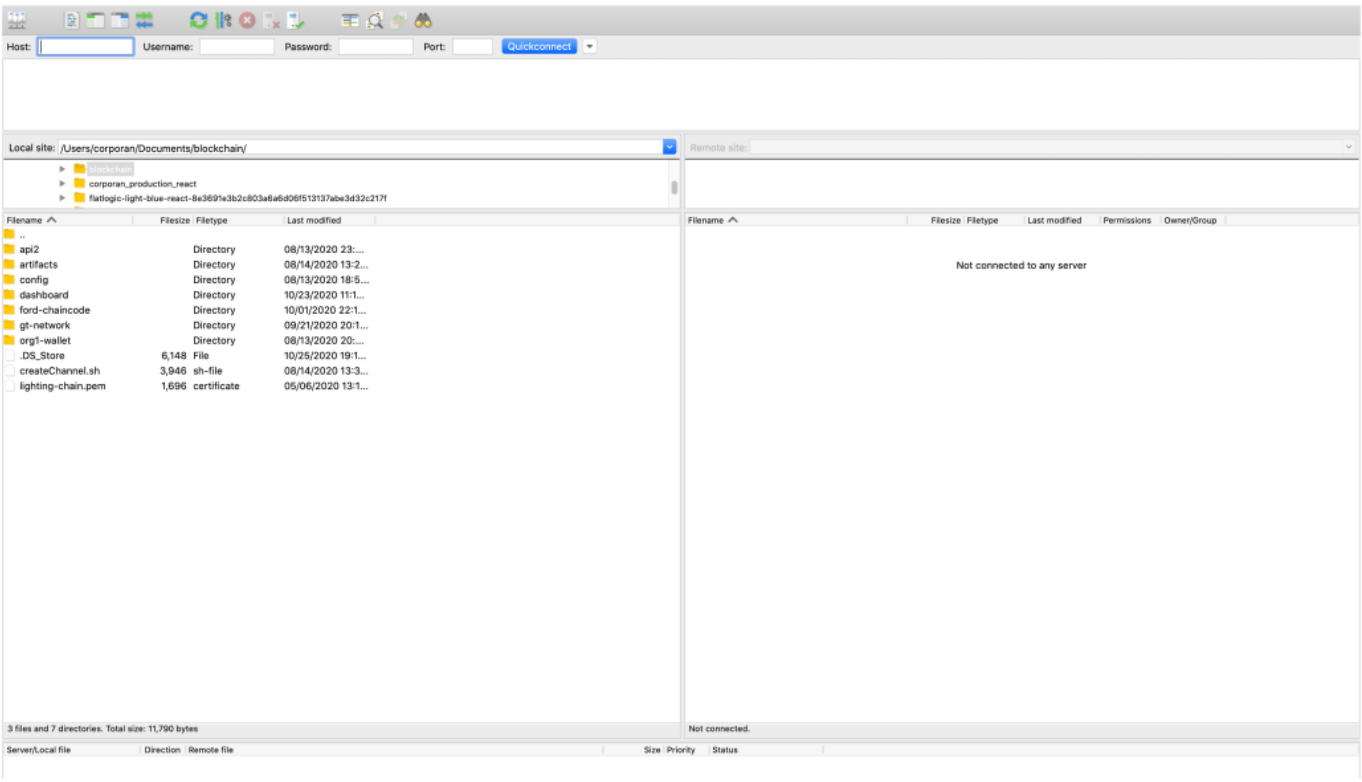
```
➜  ECE4150-Lab2 scp -i ~/Downloads/ece4150.pem -r SQL ubuntu@44.204.54.182:/home/ubuntu/data/
index.html                                    100% 3948   110.0KB/s   00:00
viewphotos.html                               100% 4532   137.3KB/s   00:00
photoForm.html                                100% 3370    98.6KB/s   00:00
searchPhoto.html                              100% 4570   130.5KB/s   00:00
albumForm.html                                100% 3150    94.7KB/s   00:00
searchAlbum.html                              100% 3866   111.7KB/s   00:00
photodetail.html                              100% 6028   149.8KB/s   00:00
.DS_Store                                     100% 6148   162.0KB/s   00:00
app.py                                        100%   13KB 376.8KB/s   00:00
.DS_Store                                     100% 6148   173.3KB/s   00:00
.DS_Store                                     100% 6148   160.5KB/s   00:00
84FC94D8-3604-4A7B-B0EC-D81252881836.jpeg     100%  756KB   3.0MB/s   00:00
Canon_PowerShot_S40.jpg                       100%   32KB 741.4KB/s   00:00
```

upload utils

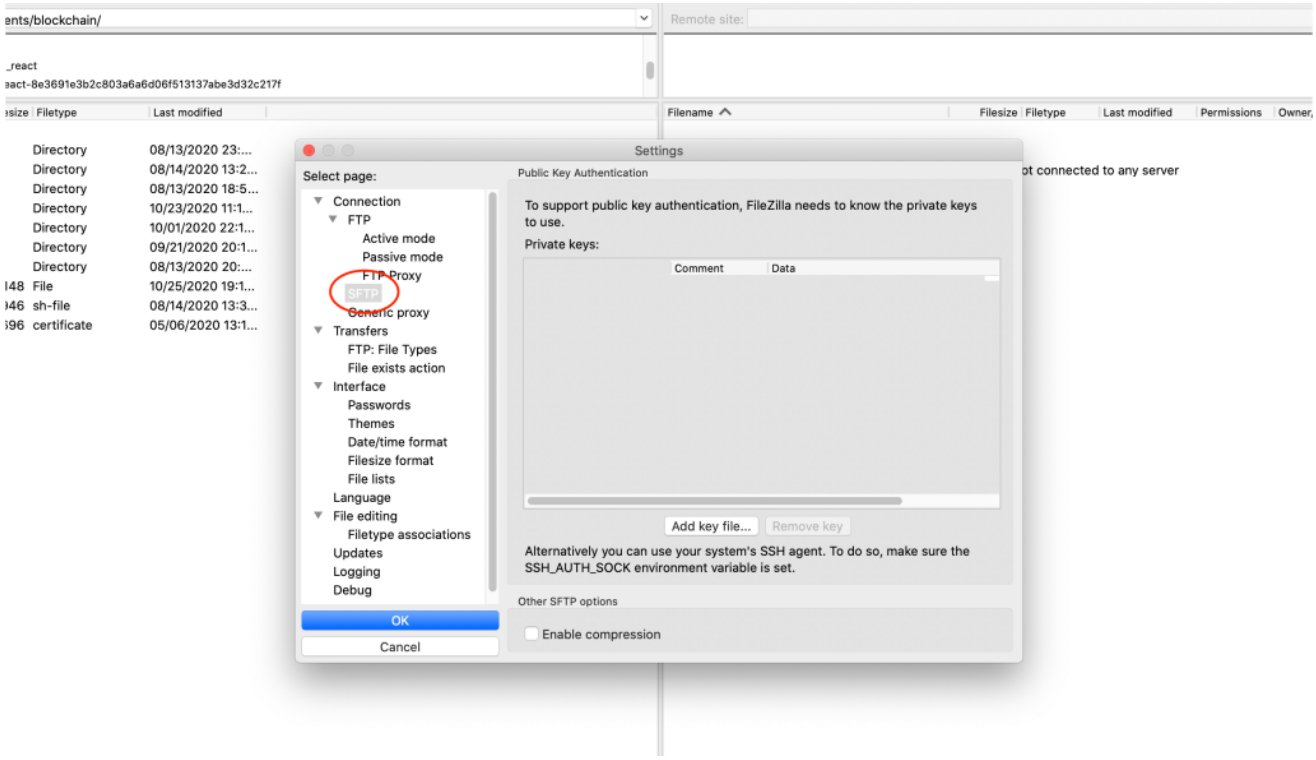
```
scp -i ~/Downloads/ece4150.pem -r utils
ubuntu@44.204.54.182:/home/ubuntu/data/
```

```
➜  ECE4150-Lab2 scp -i ~/Downloads/ece4150.pem -r utils ubuntu@44.204.54.182:/home/ubuntu/data/
env.py                                        100%  587    20.4KB/s   00:00
sqlcommands.sql                               100%  992    30.3KB/s   00:00
env.cpython-39.pyc                            100%  470    14.1KB/s   00:00
user-table.py                                 100%  981    30.2KB/s   00:00
album-photo-tables.py                         100% 1402    36.2KB/s   00:00
➜  ECE4150-Lab2
```

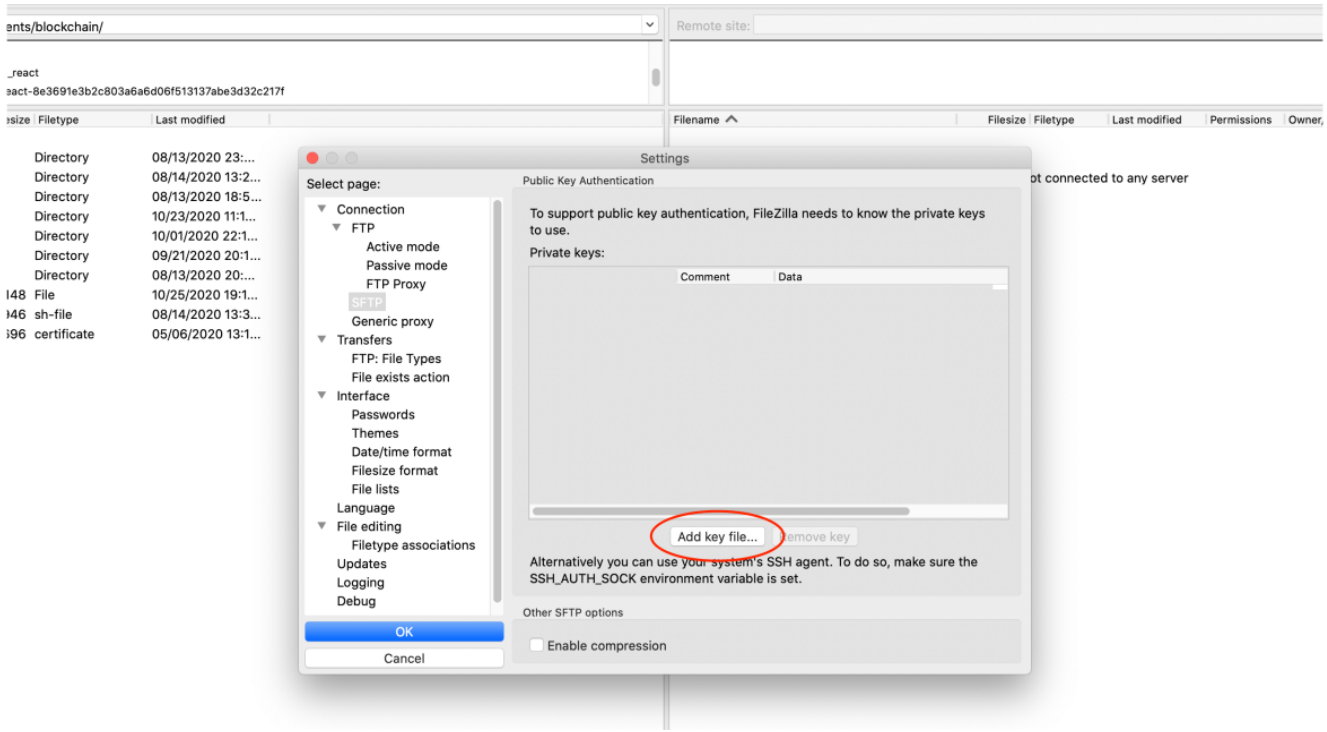
- If you want a more user-friendly tool to transfer data, FileZilla is the right choice. It is free, supports Windows/Linux/Mac systems, and has a friendly user interface. It supports FTP, SFTP, and other le transfer protocols. Go to https:// lezilla-project.org/ to download it.



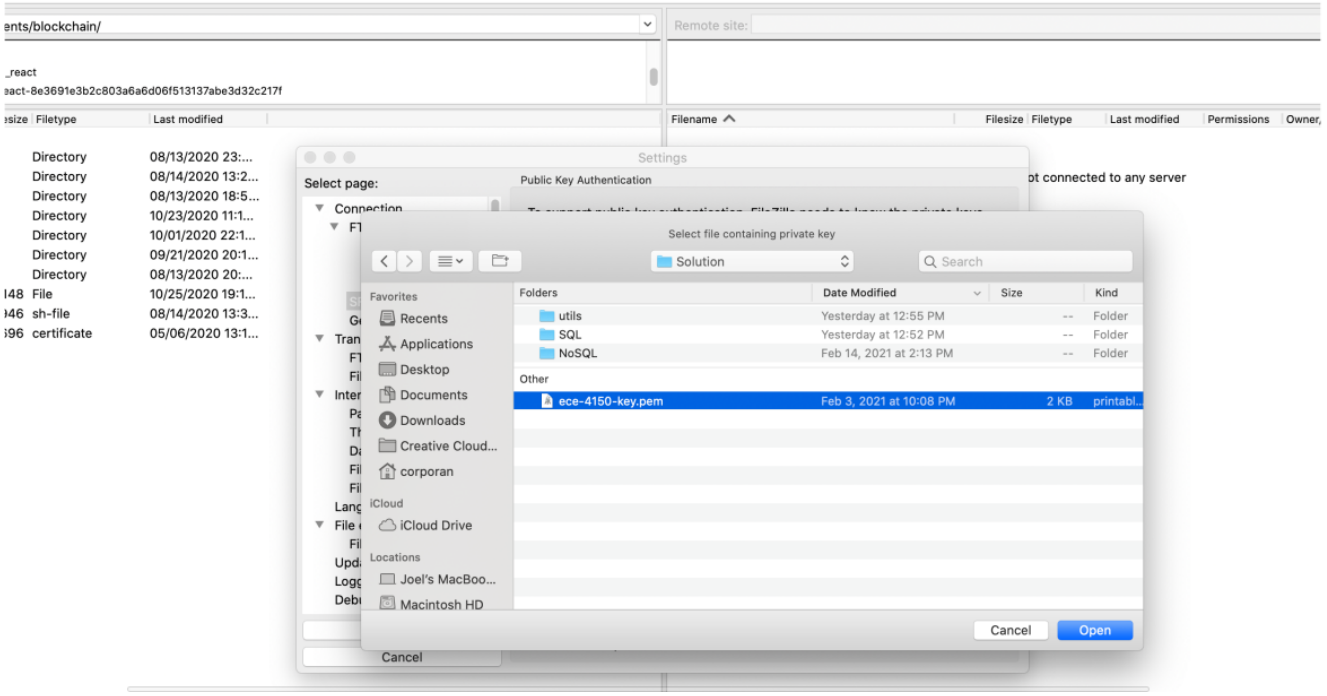
- If you want to use FileZila to upload to or download data from a normal FTP server if you have the user and password, just put the information in the “Host”, “Username”, “Password” box and connect. However for Amazon instance, we use key-pair to log in instead of password for better safety. So it is a little bit more complicated to configure.
- Under “Settings” and click “SFTP”:



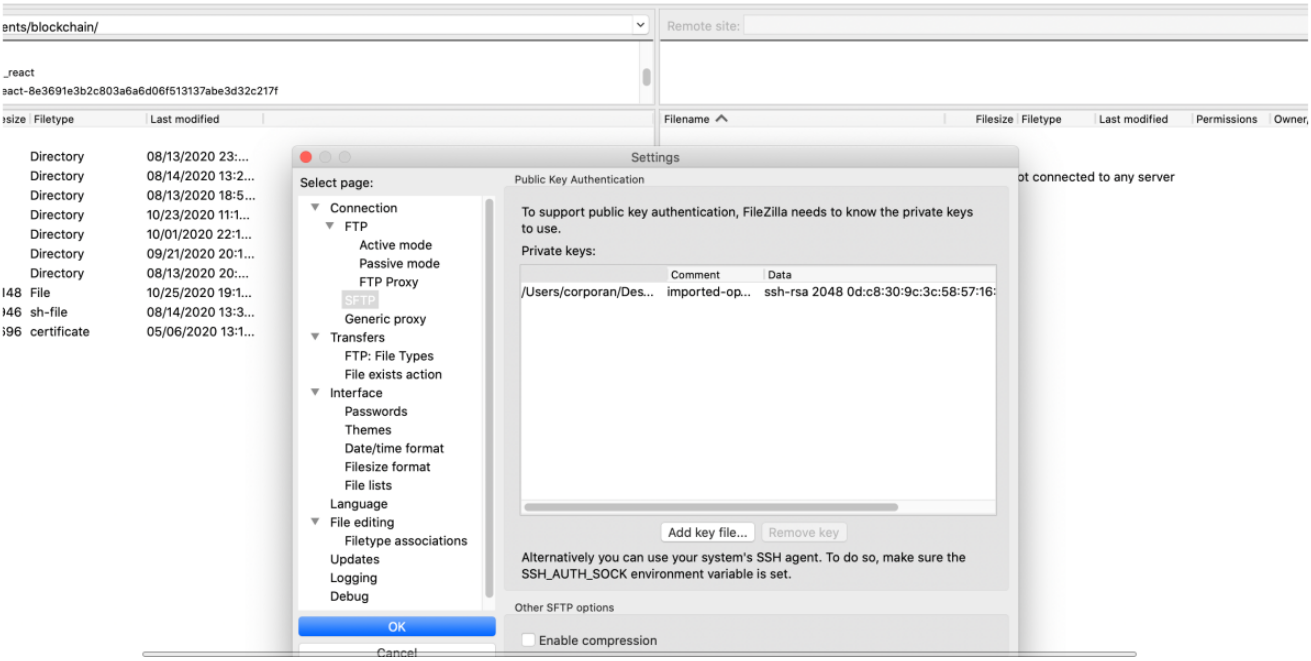
- Click “Add key file…”:



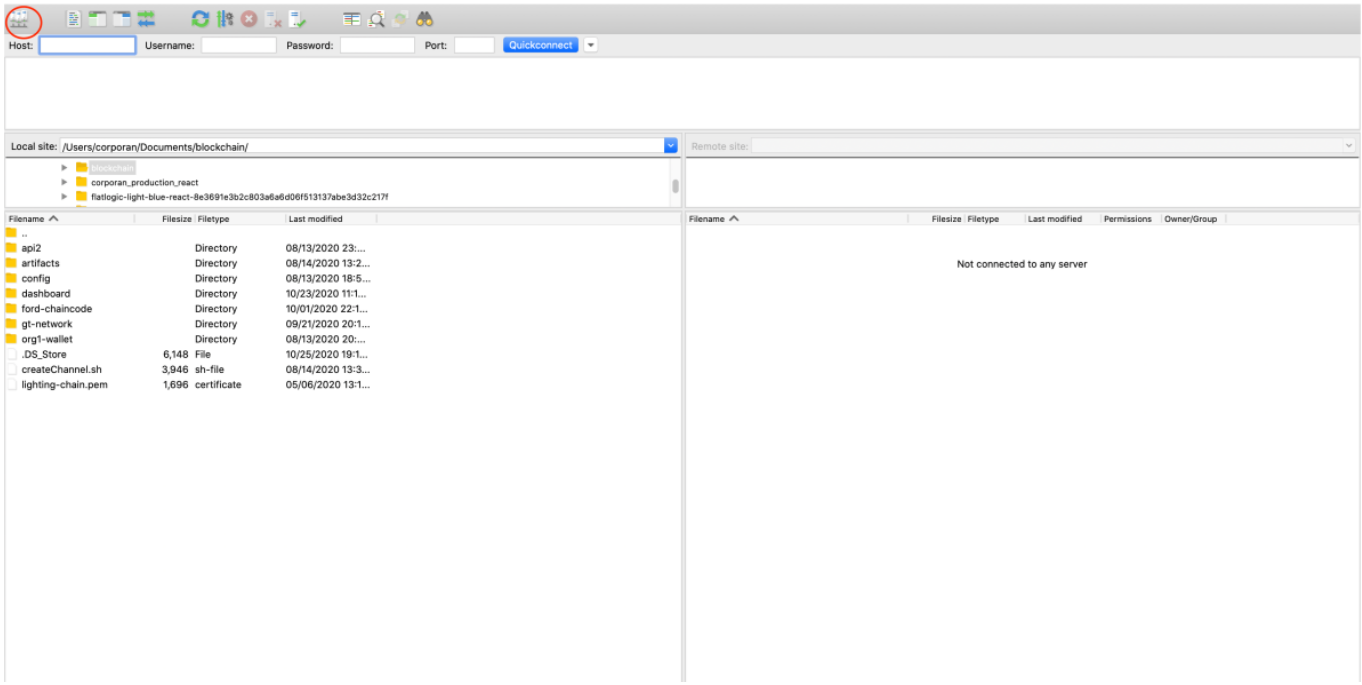
- Then select the ”.pem” file you used to connect to Amazon instance using SSH.



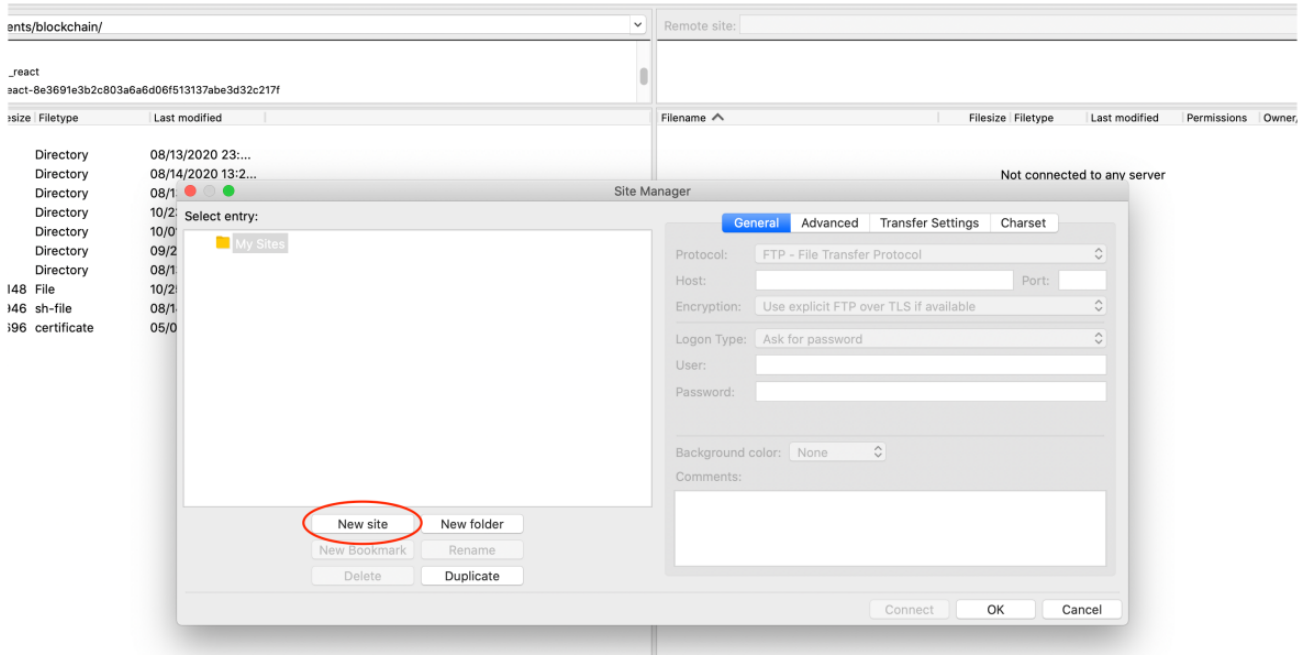
- You will see the a private key has been added.



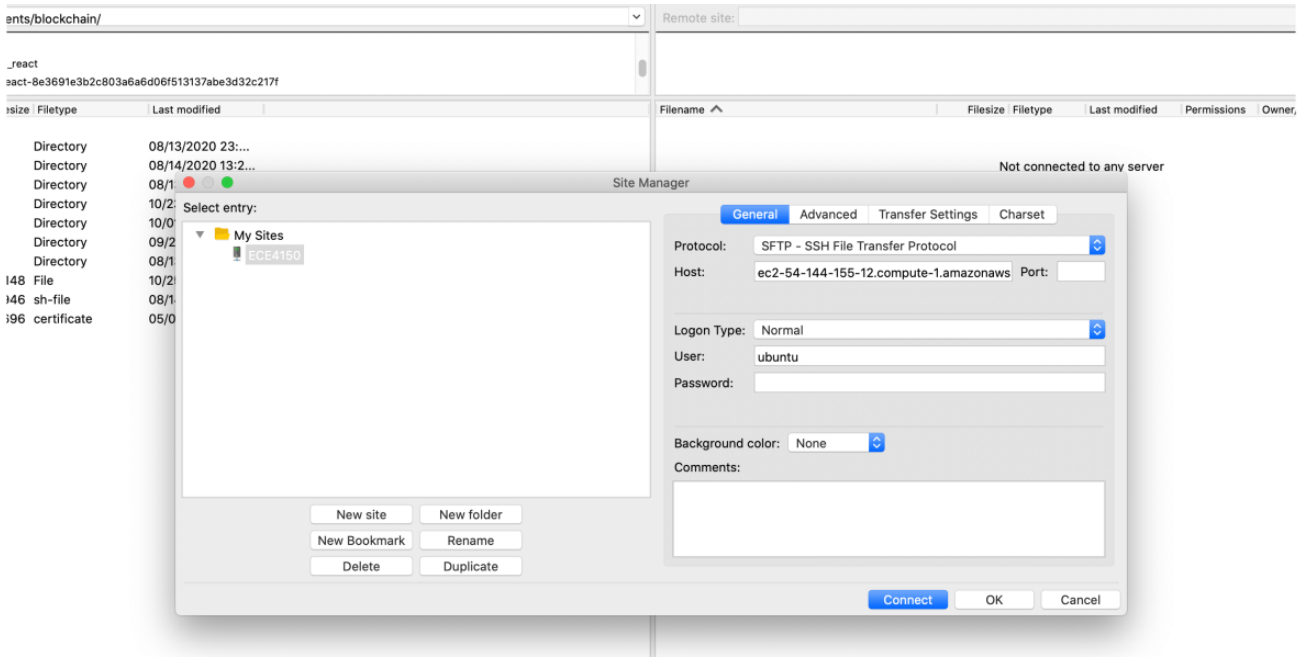
- Close “Settings” and go back to the main interface and click the button to open the “Site manager”, as shown below



- Click “New Site” and give a name.

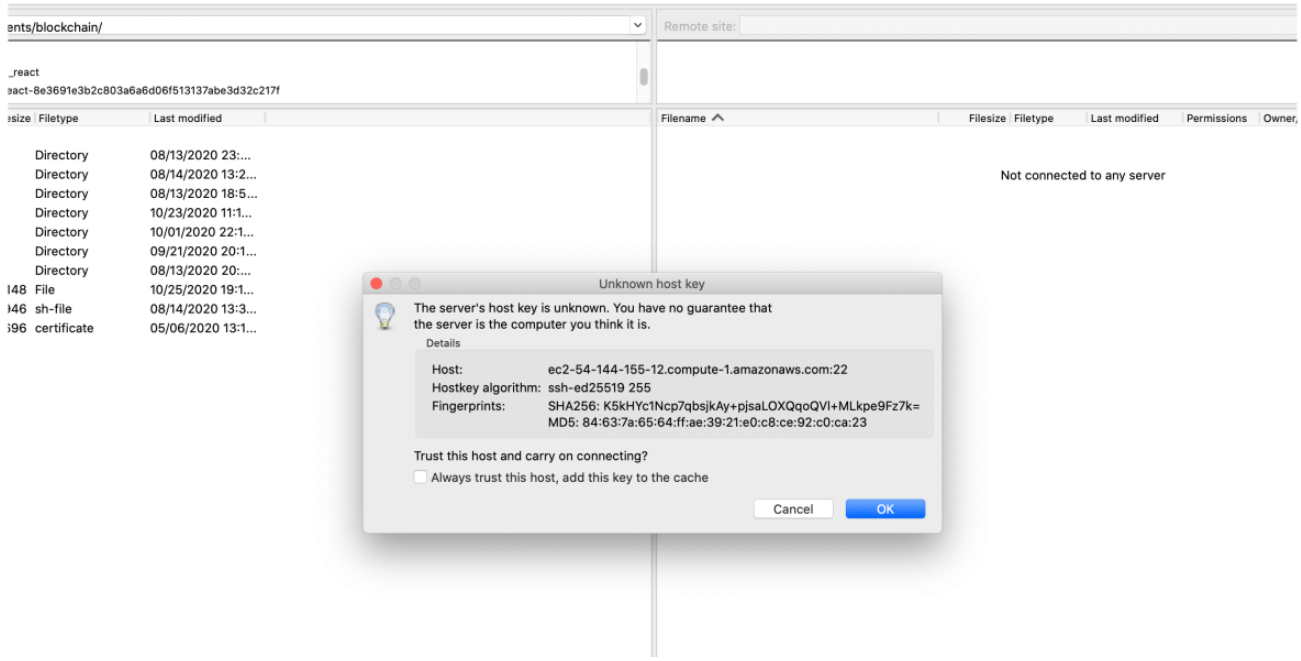


- Put the host URL of the EC2 instance in the “Host” box. Set “Protocol” as “SFTP”, “Logon Type” as “Normal”, “User” as “ubuntu” and leave “Password” as blank. Then click “Connect”.

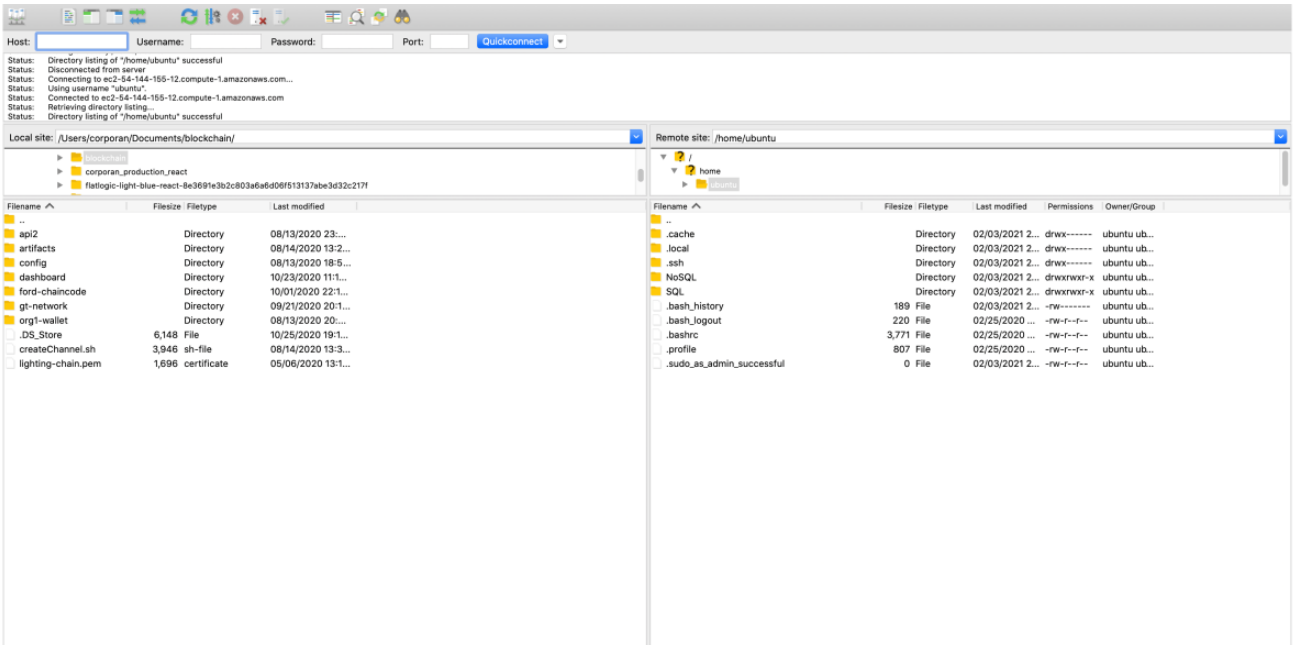


-

- There will be a dialogue box to ask you about “Unknown host key”, just click “Ok”.



- All right. Now you have logged in the EC22 instance. You can drag and drop to transfer the les between you computer and the remote machine.



9. Getting familiar with the code and installing the necessary libraries inside the EC2 instance (2 points)

- Update your EC2 instance by running this command once you are inside the instance:

```
sudo apt update
```

- Take a look at the NoSQL and SQL directory and get familiar with the code provided to run the Flask web framework (app.py). To run the framework, we need a couple of libraries using the terminal. First, make sure you have the package installer pip3 and Python3 installed on the instance. You can use this command to instance these dependencies:

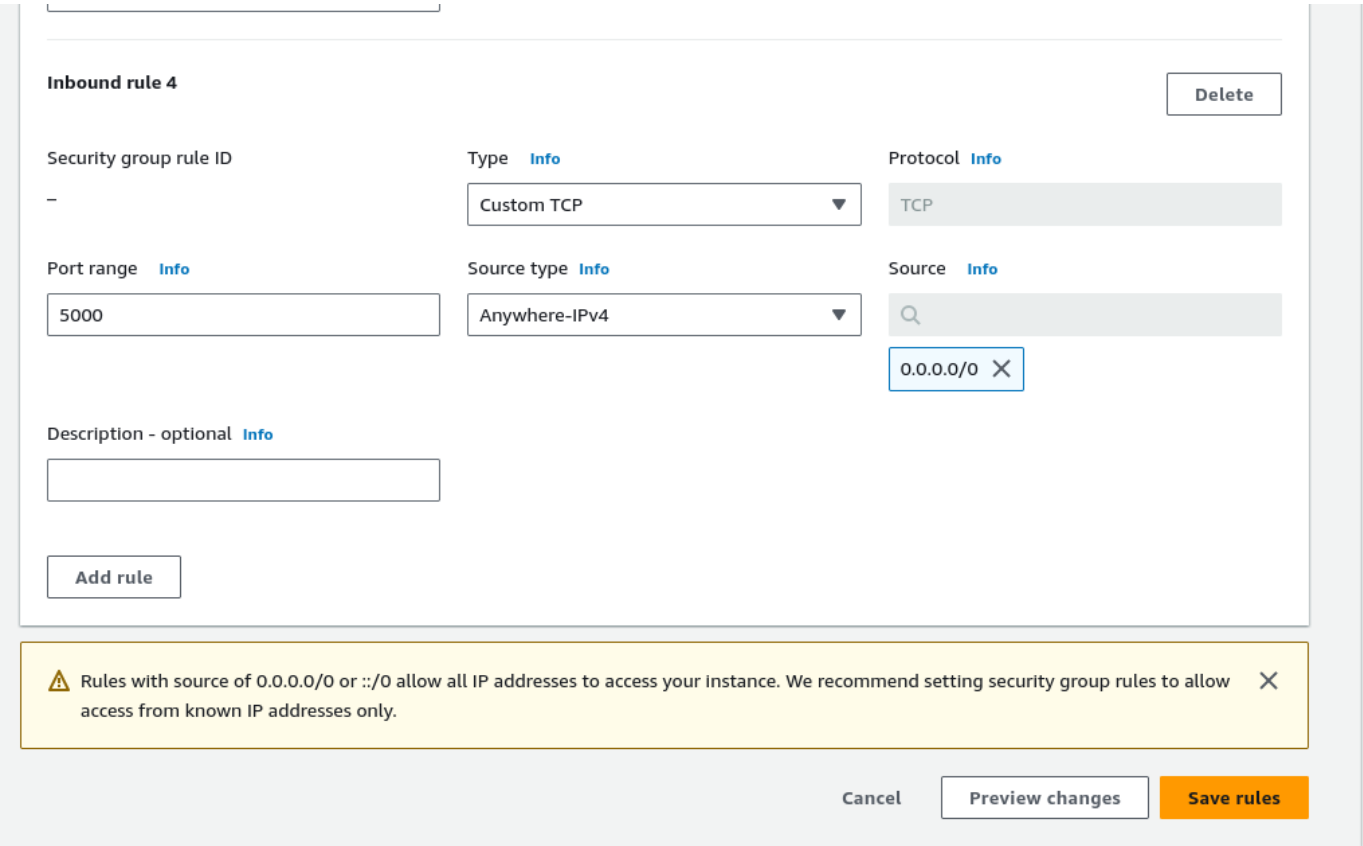
```
sudo apt install python3-pip
```

The python version used for this lab was 3.9.0 and pip3 20.3

- Now, we will install a couple of libraries that we will use for both SQL and NoSQL variants. After installing python3 and pip3, install the following libraries using the package installer pip

```
pip3 install exifread flask PyMySQL boto3 pytz
```

- If you get an error about missing libraries, use pip3 to install those libraries.
- Go back to the Security Group section under the EC2 Console and add another inbound rule for a **Custom TCP** type from port **5000** and con gure the source coming from anywhere (**0.0.0.0/0**). This con guration will allow you to access your application inside the EC2 through the 5000 port.



You can also test each of the variants outside your EC2. If you have Python3 and pip3 installed on your computer, install all the dependencies mentioned above and run the following command inside the directory the variant that you would like to test:

```
python3 app.py
```

10.Photo Gallery using NoSQL (2 points)

- For the NoSQL variant, we will store the album and photo records within the same table using the partition key and sort key for the albumID and photoID, respectively.

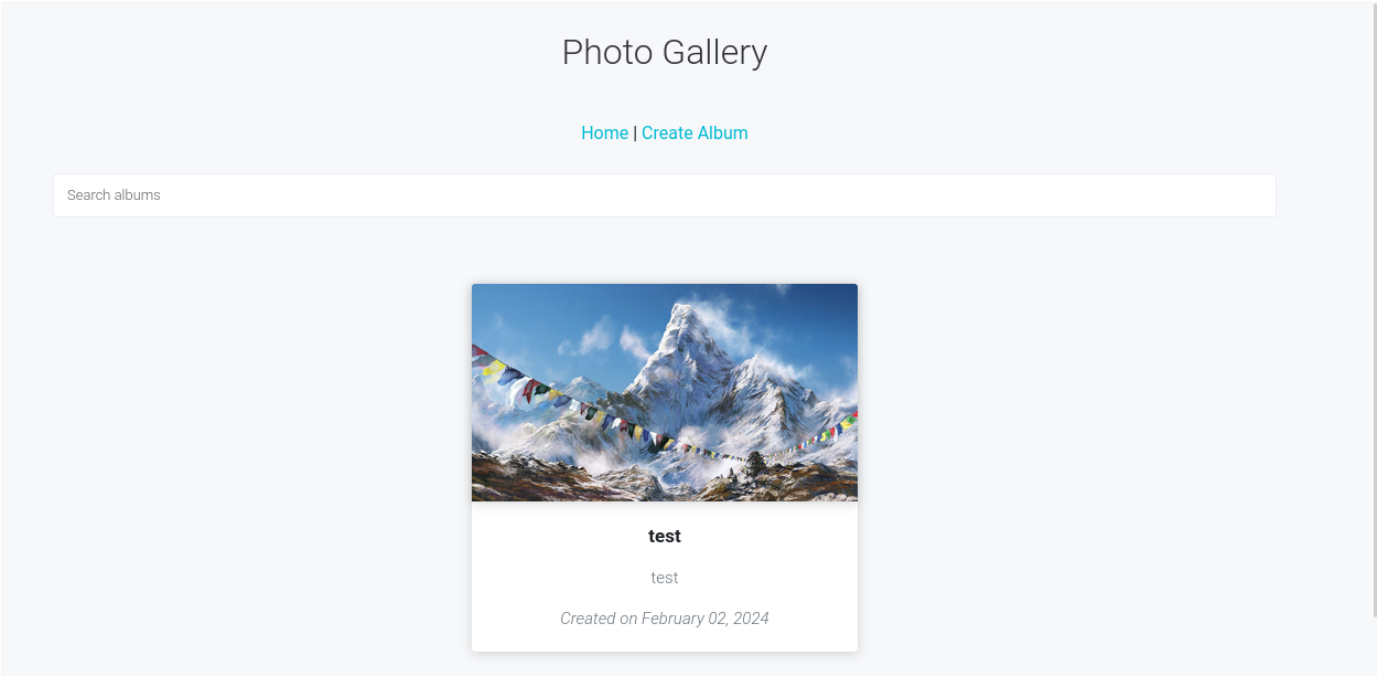
DynamoDB Partition Keys and Sort Keys

- Inside the NoSQL directory, run the following command to run the Flask web framework using the NoSQL variant:

```
python3 app.py
```

- Copy the hostname (e.g., URL or endpoint) from your EC2 instead of the link provided in the terminal when you run the Flask web framework. Add the port name (5000) and add it to the browser. It is advised to use the following browsers: Chrome and Firefox.

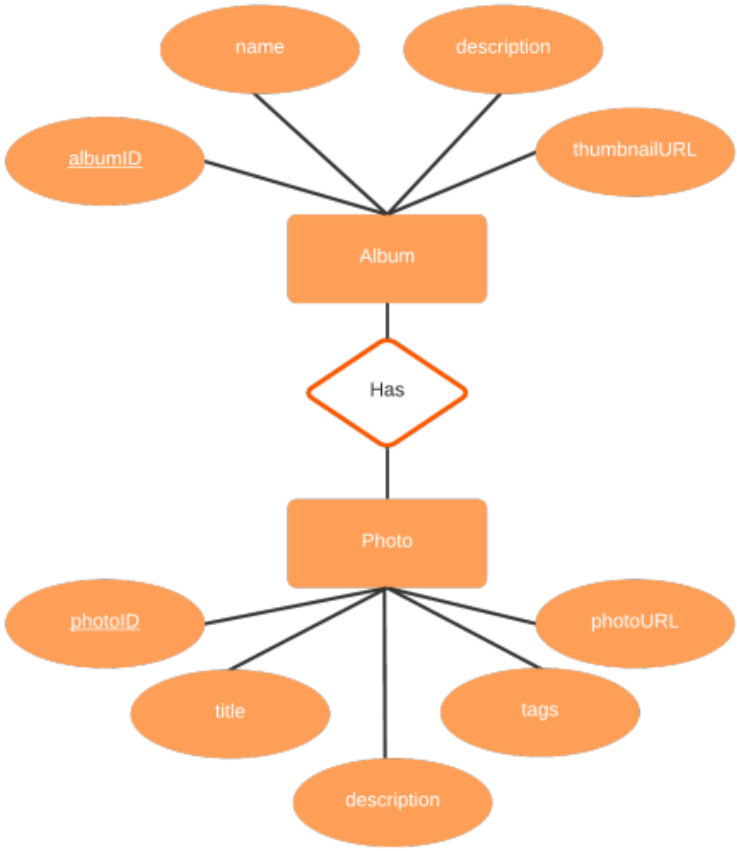
```
yourEC2InstanceHostName.com:5000
```



Remember to turn off your application if you want to use the other variant

11.Photo Gallery using SQL (2 points)

- For the SQL variant, we will store the album and photo records in different tables.



Entity Relationship Diagram that show the relation between albums and photos

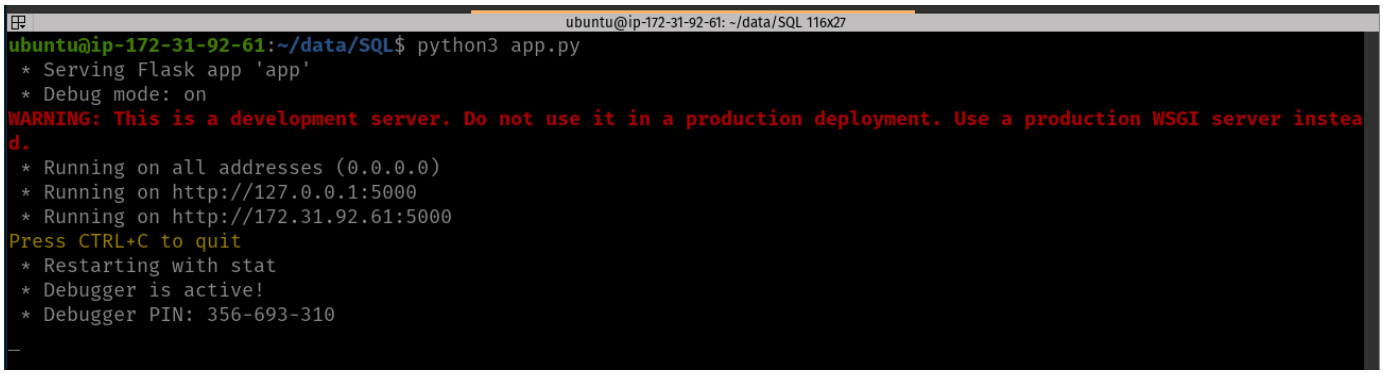
- Inside the utils directory, you will nd a **sqlcommands.sql** that includes the **Album** and **Photo** tables (the **User** table will be used in the last exercise).
- Within the same directory, you will find a python script to create the SQL tables for **Album** and **Photo**. Get familiar with the script's code structure because the method to call the RDS instance is in some way similar to the one used in the web framework. Run the following command to create the **Album** and **Photo** table:

```
python3 album-photo-tables.py
```

```
ubuntu@ip-172-31-92-61:~/data/utils$ python3 album-photo-tables.py
Connecting to RDS instance
Connected to RDS instance

Server version: 8.0.28

Creating table for albums.

Table for albums created.

Creating table for photos.

Table for photos created.
ubuntu@ip-172-31-92-61:~/data/utils$ _
```

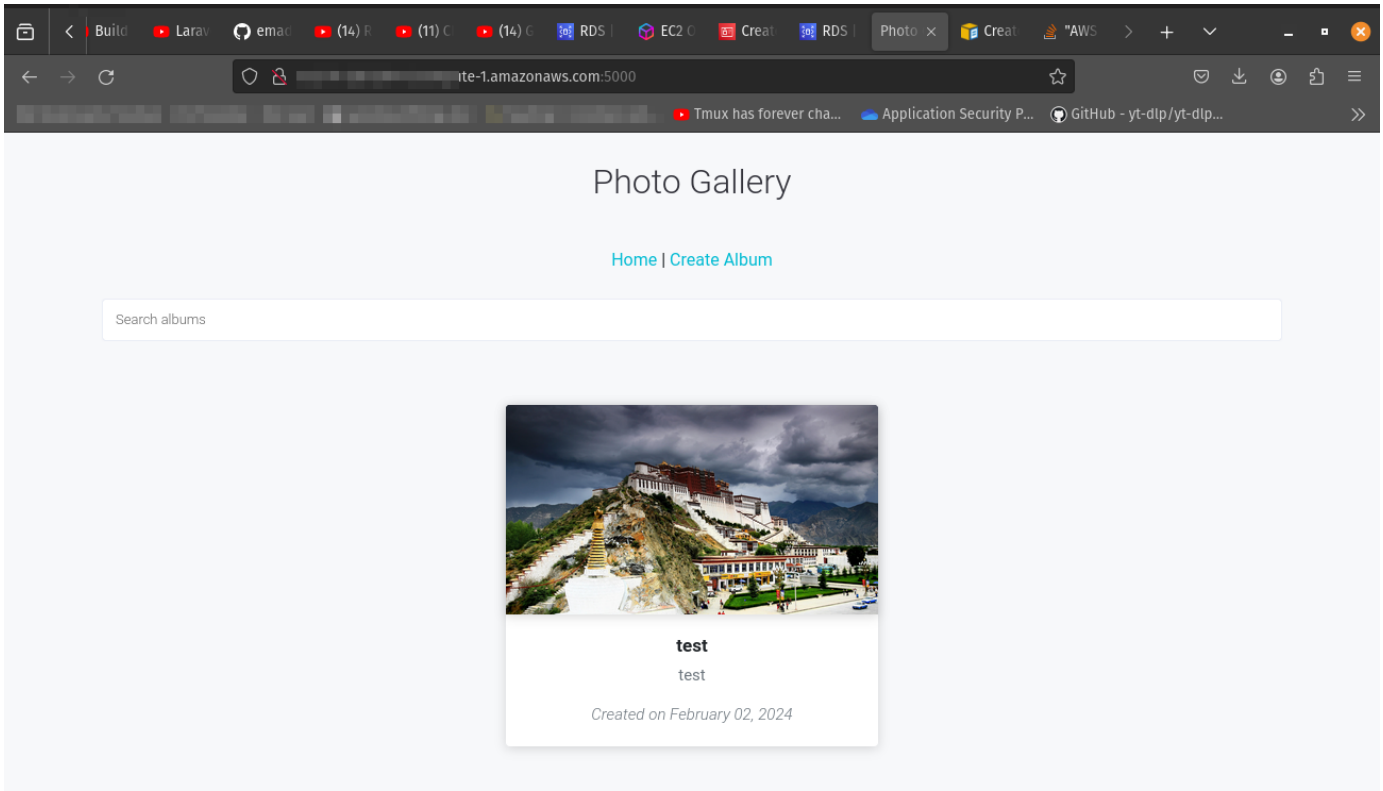
- Run the following command within the SQL root directory to run the Flask web framework using the SQL variant:

```
python3 app.py
```



- Copy the hostname (URL) from your EC2 instead of the link provided in the terminal when you run the Flask web framework. Add the port name (5000) and add it to the browser. It is advised to use the following browsers: Chrome and Firefox.

```
yourEC2InstanceHostName.com:5000
```



Remember to turn off your application if you want to use the other variant

12.Exercises: User Management and authentication, besides other CRUD operations, to support the Photo Gallery (60 points)

- Modify both NoSQL and SQL in the following exercises:

1. Design and implement a new method for user management and authentication without using external services, such as AWS Cognito. **(30 points)**
    1. Allow users to create an account (sign up) and sign in (log in) through the website.
    2. Store the information of the user in a new table called User (For SQL variant, we provided a sample schema that you can use. However, you might need include other attributes within that schema. For NoSQL variant, create a new table called **PhotoGalleryUser**). A few requirements for the use case:
       i. Use password hashing when signing up (**Appendix C**).
       ii. The **userID** is a universally unique identi er (UUID).
       iii. UserID and email **must be** unique. A user cannot sign up multiple times with the same email.
       iv. To login the user uses the **email** and **password** to login.
    3. To validate a new account, send an email to the user (See **Appendix A**) with a confirmation token with a URL to the server (See **Appendix B**) (the con rmation token should include the **userID**). When the user clicks the email, the URL should route to a resource in your Flask web framework. The resource will take the incoming request, decode the token string to get the **userID**, and con rm the user in the database. The resource's response should redirect to the login page. A few requirements for the use case:
       i. The user cannot use the website until the user is verified
       ii. The token should last 10 minutes. If the user does not click the link within that time, the token will be invalid.
    4. When the user logs in, the web server should store a session token in the user's browser. This token will be used to validate the navigation of that particular user in the website. The token should last only 5 minutes. After the token is expired, the user should be redirected to the login page.
2. Add a new method to allow the user to cancel its account. **(15 points**)
    1. Remove all the albums created by a user when this one cancels its account. A few requirements for the use case:
       i. For NoSQL, attached a new attribute when the album is created to identify the creator.
       ii. For SQL, update the album schema to add foreign key from the user schema. This strategy will allow you to remove the all albums created by the user.
3. Add a new method to delete a photo in the PhotoGallery. **(5 points)**
4. Add a new method to delete a whole album in the PhotoGallery. This should delete all the photos inside that particular album. (**5 points**)
5. Add a method to update the following information of a photo: title, description, and tags. For NoSQL, whenever you make an update, update the updatedAt attribute as well. In SQL, it is done automatically. (**5 points**)

For this exercise, there are designated sections (with comments) where you need to add your functions, routes, and environment variables needed. Make sure you use these sections to

insert your code. If you need to modify any other function, route, or environment variable outside this section, put a comment that identi es your modi cation.

Deliverables:

1. A video showing all the full functionalities of your Photo Gallery application, including the ones in the final exercise. Please show your AWS username clearly to indicate that you are using your account. You may briefly explain the functions that you created for the last activity, **but please make sure to limit the total length to 12 minutes**.
2. The complete code with the modifications needed to complete each exercise, including the modified SQL schema (include the modified schema in the **sqlcommands.sql** file).

Appendex

A. How to generate confirmation token

The email con rmation should contain a unique URL that a user needs to click to con rm his/her account. Typically, a con rmation URL has this form:

```
http://yourEC2InstanceHostName.com/confirm/{id}
```

Where the id is an encoded string containing the email of the particular user, a timestamp, and salt (a is random string used in hashed data to safeguard passwords in storage).

Python offered a lot of libraries to create this type of tokens. However, a recommended library for this lab is **itsdangerous** (https://itsdangerous.palletsprojects.com/en/1.1.x/). This library has a function called **URLSafeTimedSerializer** that allows to create a serialize (hash) token for URLs with a record of the time of the signing. It is a good strategy if you want to delegate the validation and expiration of signatures. Here is a simple example:

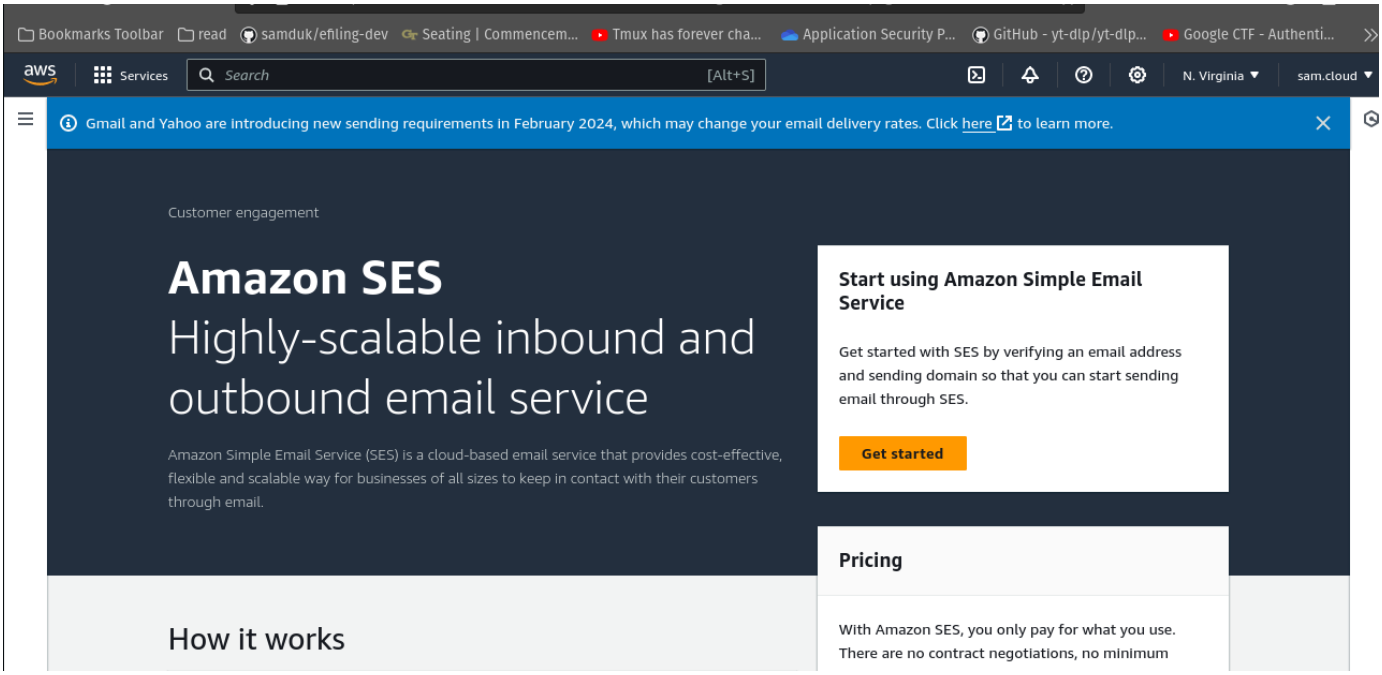
```python
from itsdangerous import URLSafeTimedSerializer

email = 'myemailaddress@gatech.edu'

serializer = URLSafeTimedSerializer('some_secret_key')
token = serializer.dumps(email, salt='some-secret-salt-for-confirmation')

print(token)
# eyJpZCI6NSwibmFtZSI6Iml0c2Rhbmdlcm91cyJ9.6YP6T0BaO67XP--9UzTrmurXSmg

try:
    email = serializer.loads(
        token,
        salt='some-secret-salt-for-confirmation',
        max_age=3600
    )

    print(token)

    #myemailaddress@gatech.edu

except Exception as e:
    print('expired token')
```

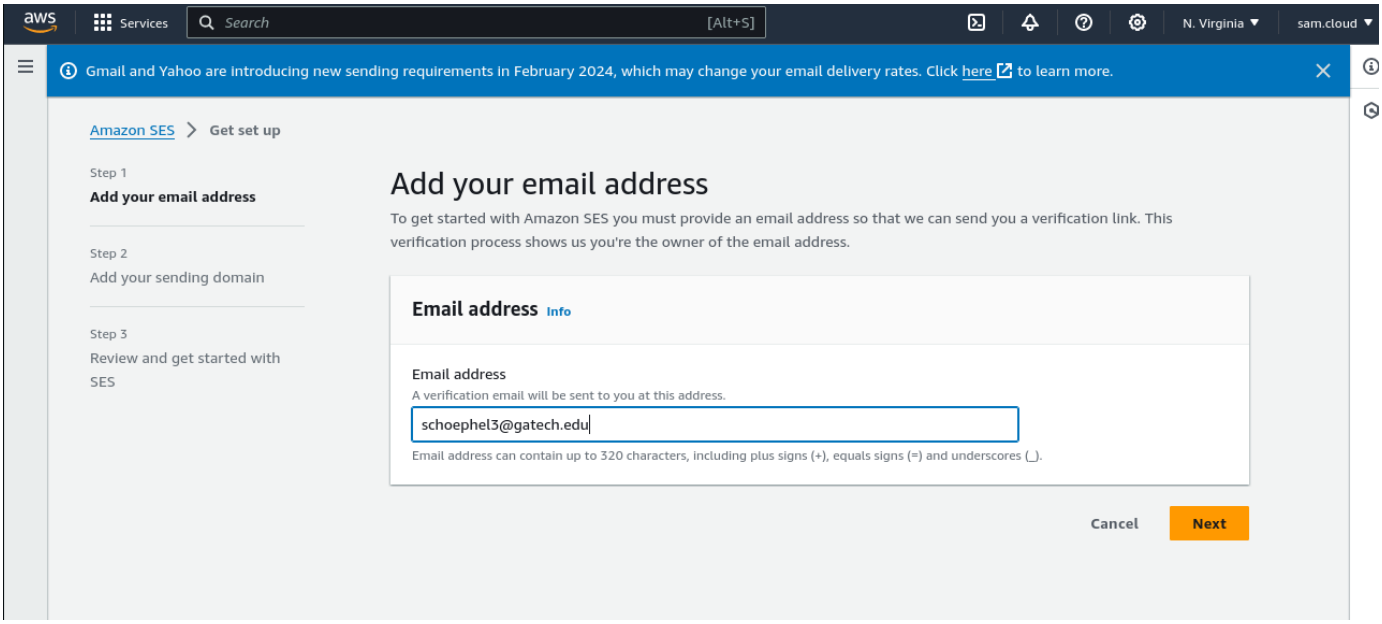
B. How to send an email (AWS SES)

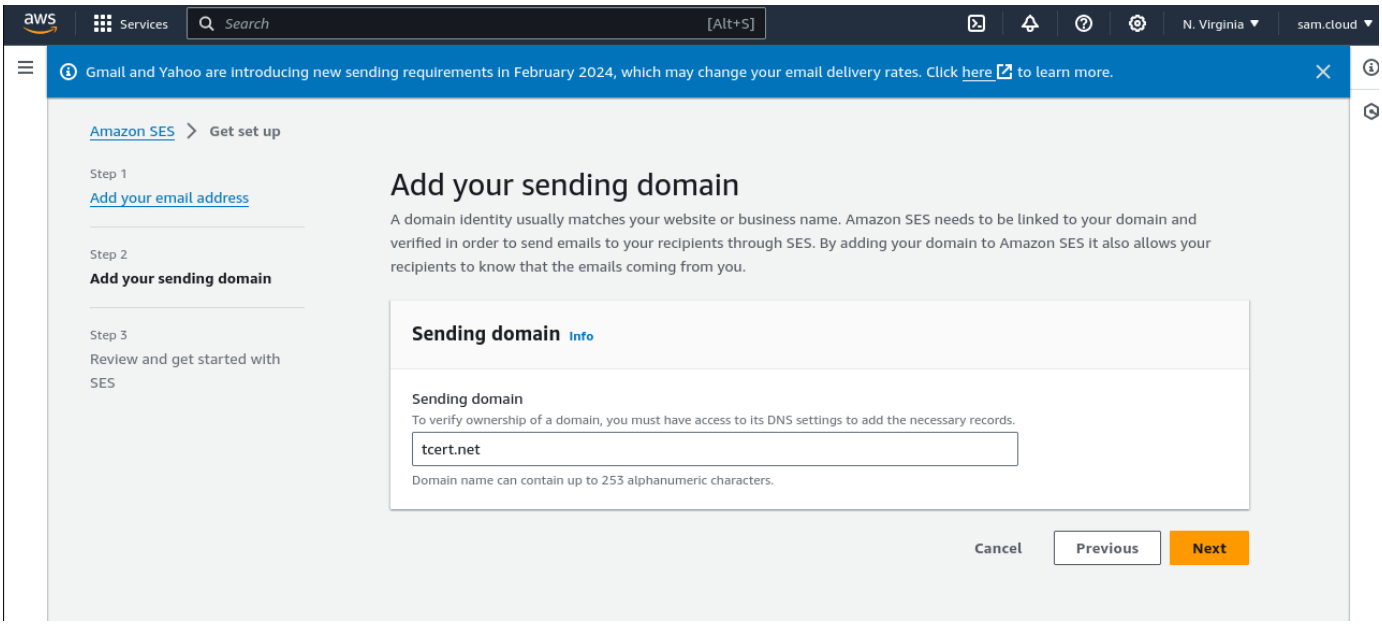
If you are trying to send a user an email to con rm their account, you need a transactional email service. AWS SES is the right solution for your application. Amazon Simple Email Service is a platform within AWS that allows you to send and receive emails using your email address or registered domain.

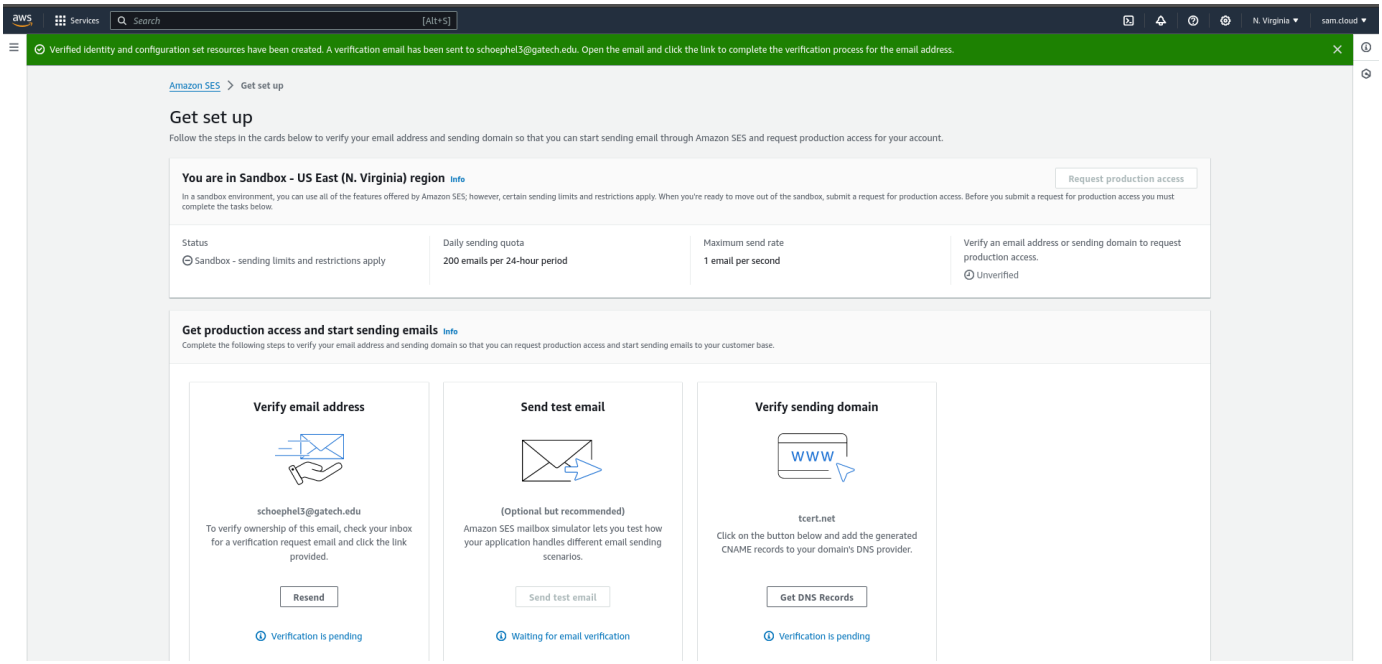
Amazon SES has proven to be a cost-effective email service. It is designed to send both bulk and transactional emails. AWS SES allows you to send emails directly from the SES console, via the Simple Mail Transfer Protocol (SMTP) interface, or through the API (You can use Boto3 to make calls the API).

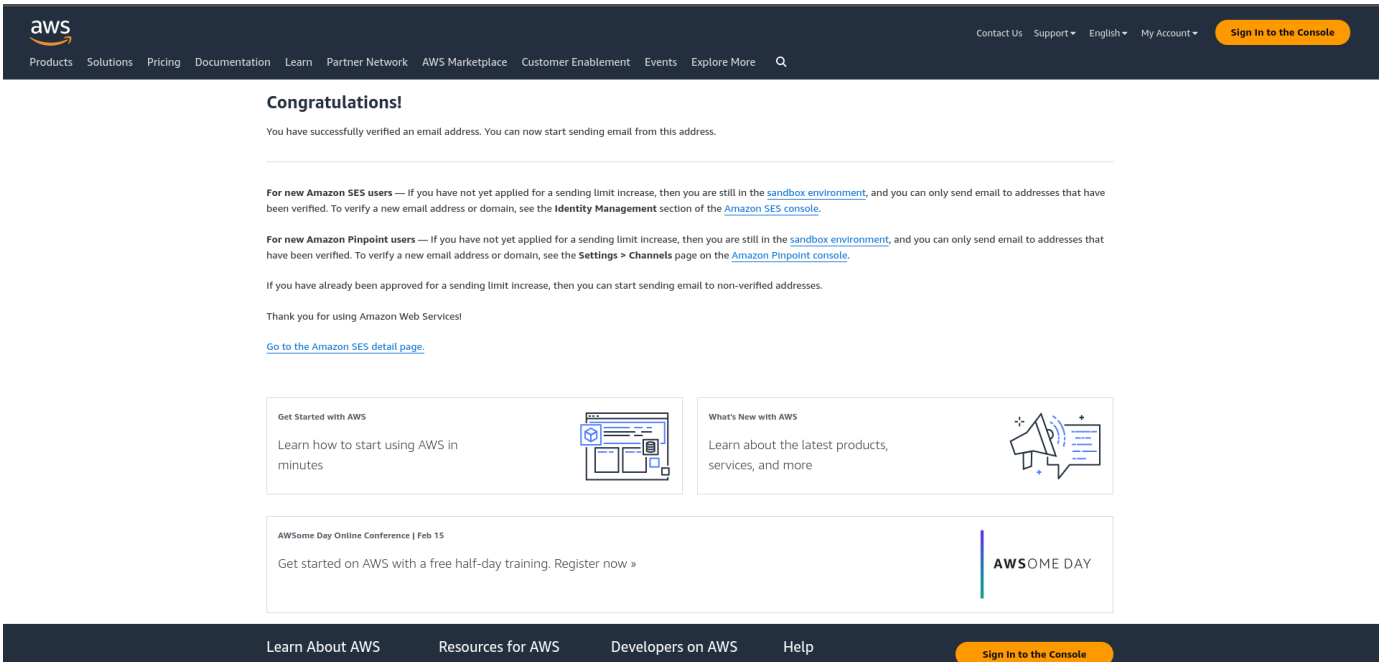
Before you can send emails with SES, you must rst verify that you own the email address you wish to use as a sender. Navigate to Amazon SES, click "Email Addresses" in the sidebar, and then click the "Verify a New Email Address" button.


Bookmarks Toolbar
read
samduk/efiling-dev
Seating | Commencem...
Tmux has forever cha...
Application Security P...
GitHub - yt-dlp/yt-dlp...
Google CTF - Authenti...
aws
Services
Search
[Alt+S]
N. Virginia
sam.cloud
Gmail and Yahoo are introducing new sending requirements in February 2024, which may change your email delivery rates. Click here to learn more.
Customer engagement
Amazon SES
Highly-scalable inbound and outbound email service
Amazon Simple Email Service (SES) is a cloud-based email service that provides cost-effective, flexible and scalable way for businesses of all sizes to keep in contact with their customers through email.
Start using Amazon Simple Email Service
Get started with SES by verifying an email address and sending domain so that you can start sending email through SES.
Get started
Pricing
With Amazon SES, you only pay for what you use. There are no contract negotiations, no minimum
How it works

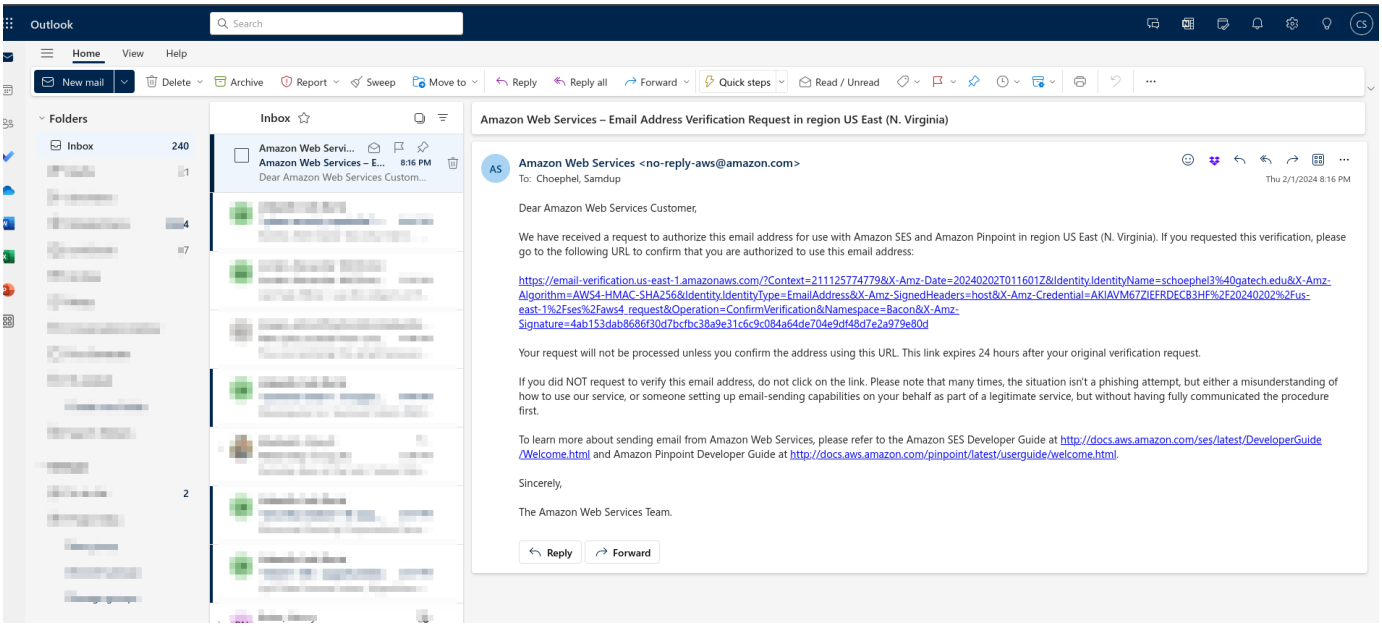

aws
Services
Search
[Alt+S]
N. Virginia
sam.cloud
Gmail and Yahoo are introducing new sending requirements in February 2024, which may change your email delivery rates. Click here to learn more.
Amazon SES
Get set up
Step 1
Add your email address
Step 2
Add your sending domain
Step 3
Review and get started with SES
Add your email address
To get started with Amazon SES you must provide an email address so that we can send you a verification link. This verification process shows us you're the owner of the email address.
Email address Info
Email address
A verification email will be sent to you at this address.
schoephel3@gatech.edu
Email address can contain up to 320 characters, including plus signs (+), equals signs (=) and underscores (_).
Cancel
Next


aws
Services
Search
[Alt+S]
N. Virginia
sam.cloud
Gmail and Yahoo are introducing new sending requirements in February 2024, which may change your email delivery rates. Click here to learn more.
Amazon SES
Get set up
Step 1
Add your email address
Step 2
Add your sending domain
Step 3
Review and get started with SES
Add your sending domain
A domain identity usually matches your website or business name. Amazon SES needs to be linked to your domain and verified in order to send emails to your recipients through SES. By adding your domain to Amazon SES it also allows your recipients to know that the emails coming from you.
Sending domain Info
Sending domain
To verify ownership of a domain, you must have access to its DNS settings to add the necessary records.
tcert.net
Domain name can contain up to 253 alphanumeric characters.
Cancel
Previous
Next





New accounts are automatically placed in a sandbox mode to help prevent fraud. You can only send emails to addresses you have personally veri ed with Amazon. If you want to remove this restriction, you must request Amazon to move out of the sandbox mode. Fortunately, this is enough since you are not deploying this application to a production environment.



Below is an example of how to send an email using boto3:

```
import boto3
from botocore.exceptions import ClientError

# Create a new SES resource and specify a region.
ses = boto3.client('ses',
                   region_name='AWS_REGION',
                   aws_access_key_id='AWS_ACCESS_KEY_ID',
                   aws_secret_access_key='AWS_SECRET_ACCESS_KEY')

SENDER = 'MyOtherVerifiedEmailAddress@gatech.edu'
RECEIVER = 'myVerifiedEmailAddress@gatech.edu'

# Try to send the email.
try:
    #Provide the contents of the email.
    response = ses.send_email(
        Destination={
            'ToAddresses': [RECEIVER],
        },
        Message={
            'Body': {
                'Text': {
                   'Data': 'This is an email from AWS SES',
                },
            },
            'Subject': {
               'Data': 'Hi, I'm sending this email from AWS SES'
            },
        },
        Source=SENDER
    )

# Display an error if something goes wrong.
except ClientError as e:
    print(e.response['Error']['Message'])

else:
    print("Email sent! Message ID:"),
    print(response['MessageId'])
```

C. Password Hashing

As we know now, it is a terrible idea to stored a password as text without any encryption. For that reason, you do not want to include passwords in your database as plain text, as this would make your users' passwords exposed if your server got hacked and your database vulnerable. That is why you need to protect sensitive data by using a hash method before storing them in your database. This is a simple step to provide a lot of security. The article below from Dustin Boswell will help you to understand and implement hashing algorithms, such as bcrypt (recommended by the author). Also, you will nd a ask version of bcrypt below.

- http://cs.wellesley.edu/~cs304/lectures/bcrypt/dustwell.html
- https://pypi.org/project/bcrypt/