

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.

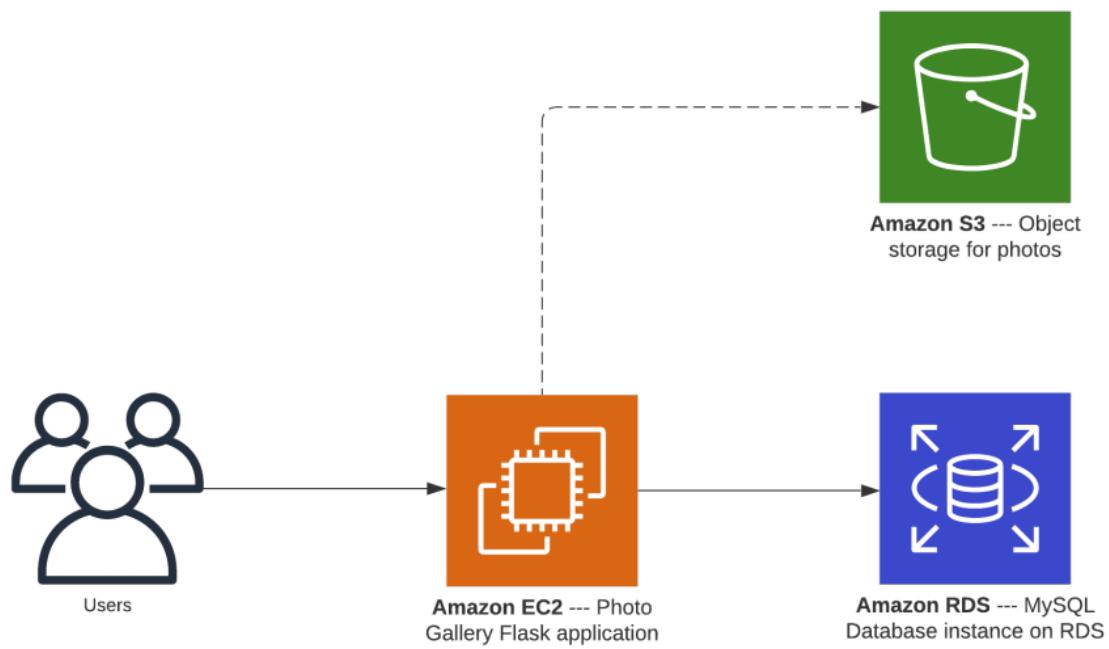


Fig.1 Architecture diagram of the Photo Gallery application - SQL Variant

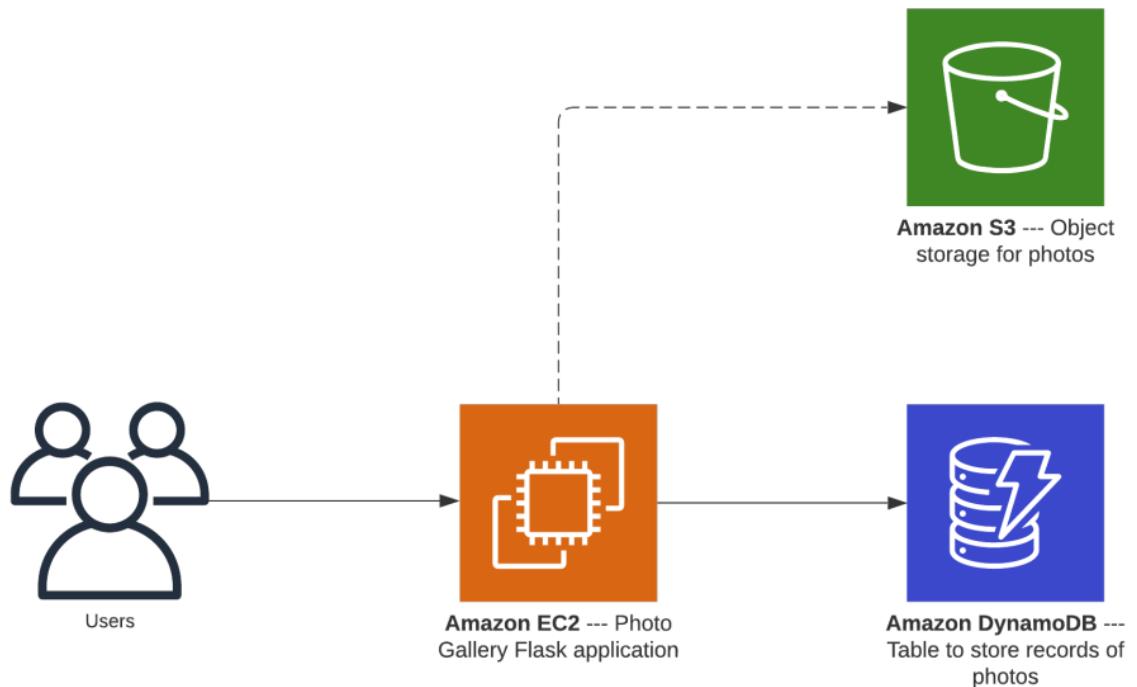
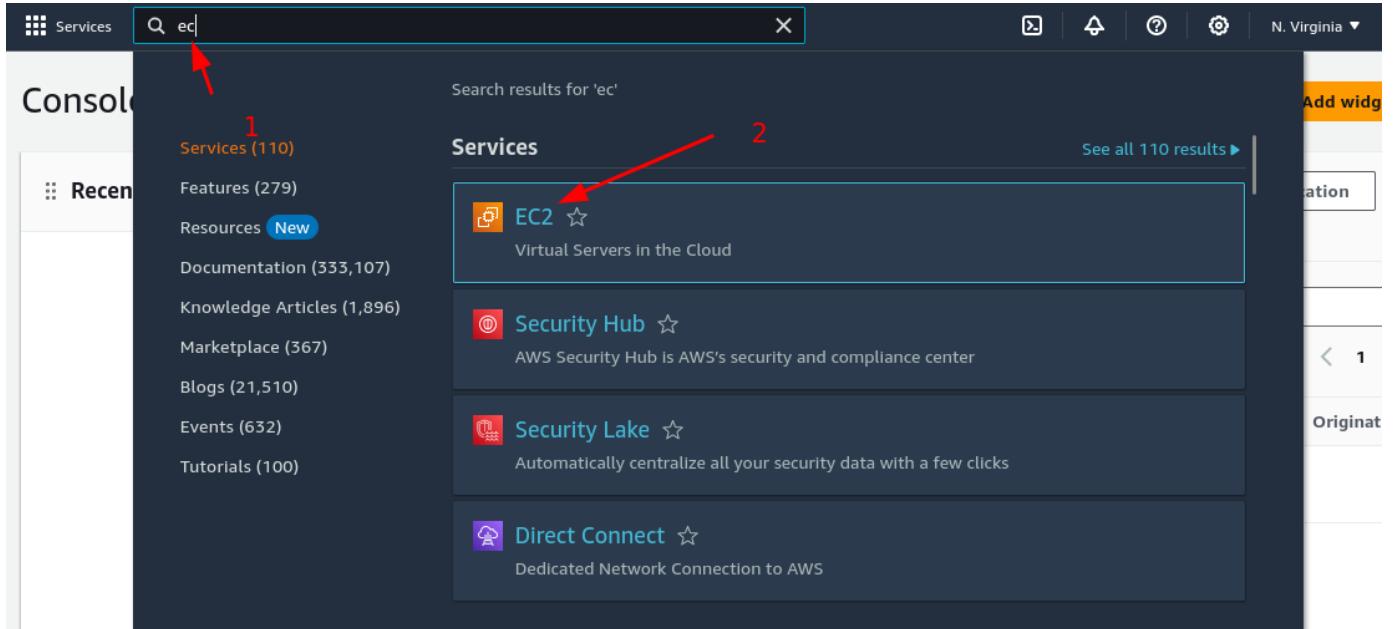


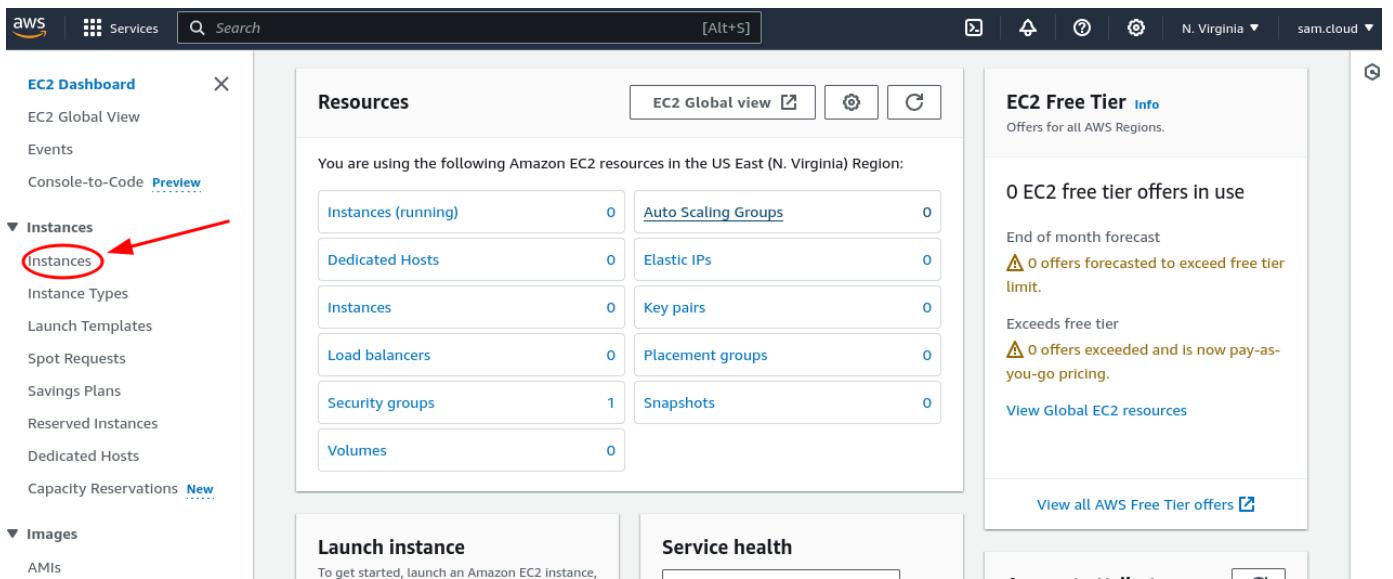
Fig. 2 Architecture diagram of the Photo Gallery application - SQL Variant

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

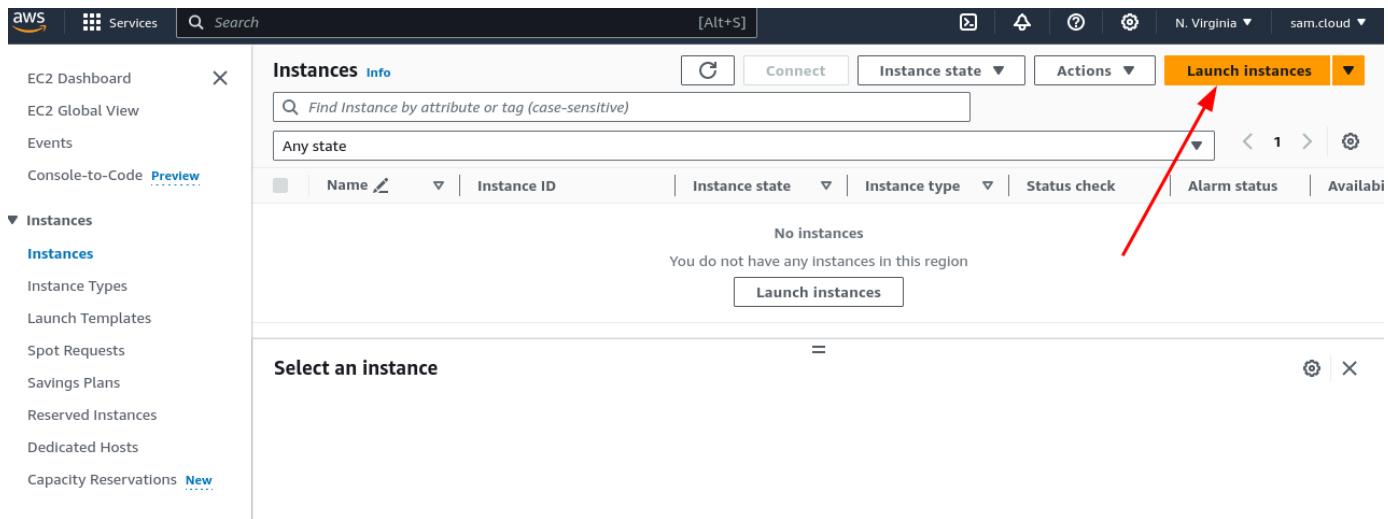
- In the AWS Management Console, under Compute, click EC2. You can also use the search input in the navigation bar by typing “EC2”



- 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 first 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

S | Services | Search | [Alt+S] | N. Virginia ▾ | sam.cloud ▾

EC2 > Instances > Launch an instance

Launch an instance Info

Amazon EC2 allows you to create virtual machines, or instances, that run on the AWS Cloud. Quickly get started by following the simple steps below.

Name and tags Info

Name
 Add additional tags

▼ Application and OS Images (Amazon Machine Image) Info

An AMI is a template that contains the software configuration (operating system, application server, and applications) required to launch your instance. Search or Browse for AMIs if you don't see what you are looking for below

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

Quick Start

Browse more AMIs
Including AMIs from AWS, Marketplace and the Community

Amazon Machine Image (AMI)

Amazon Linux 2023 AMI Free tier eligible

ami-0277155c3f0ab2930 (64-bit (x86), uefi-preferred) / ami-07ce5684ee3b5482c (64-bit (Arm), uefi)
Virtualization: hvm ENA enabled: true Root device type: ebs

Description
Amazon Linux 2023 AMI 2023.3.20240131.0 x86_64 HVM kernel-6.1

Architecture 64-bit (x86) **Boot mode** uefi-preferred **AMI ID** ami-0277155c3f0ab2930 **Verified provider**

S | Services | Search | [Alt+S] | N. Virginia ▾ | sam.cloud ▾

EC2 > Instances > Launch an instance

Launch an instance Info

Amazon EC2 allows you to create virtual machines, or instances, that run on the AWS Cloud. Quickly get started by following the simple steps below.

Name and tags Info

Let's give our EC2 instance a name (it can be changed later)

Name Add additional tags

- To find the right Ubuntu image, type “Ubuntu” in the search input and find 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

Selected AMI: (ami-0277155c3f0ab2930) (Quickstart AMIs)

X ▼

Quickstart AMIs (9)
Commonly used AMIs

My AMIs (0)
Created by me

AWS Marketplace AMIs (2323)
AWS & trusted third-party AMIs

Community AMIs (500)
Published by anyone

Refine results

Free tier only Info

OS category

All Linux/Unix
 All Windows

Architecture

64-bit (Arm)
 32-bit (x86)
 64-bit (x86)
 64-bit (Mac)
 64-bit (Mac-Arm)

ubuntu (9 filtered, 9 unfiltered)

ubuntu Ubuntu 64-bit (x86) 64-bit (Arm) Select

Ubuntu Server 22.04 LTS (HVM), SSD Volume Type
ami-0c7217cdde317cfec (64-bit (x86)) / ami-05d47d29a4c2d19e1 (64-bit (Arm))
Ubuntu Server 22.04 LTS (HVM), EBS General Purpose (SSD) Volume Type.
Support available from Canonical (<http://www.ubuntu.com/cloud/services>).
Platform: ubuntu Root device type: ebs
Virtualization: hvm ENA enabled: Yes

ubuntu Ubuntu 64-bit (x86) 64-bit (Arm) Select

Ubuntu Server 20.04 LTS (HVM), SSD Volume Type
ami-06aa3f7caf3a30282 (64-bit (x86)) / ami-0a75bd84854bc95c9 (64-bit (Arm))
Ubuntu Server 20.04 LTS (HVM), EBS General Purpose (SSD) Volume Type.
Support available from Canonical (<http://www.ubuntu.com/cloud/services>).
Platform: ubuntu Root device type: ebs
Virtualization: hvm ENA enabled: Yes

ubuntu Ubuntu 64-bit (x86) 64-bit (Arm) Select

Ubuntu Server 20.04 LTS (HVM) with SQL Server 2022 Standard
ami-032346ab877c418af (64-bit (x86))
Microsoft SQL Server 2022 Standard edition on Ubuntu Server 20.04 LTS.
Platform: ubuntu Root device type: ebs
Virtualization: hvm ENA enabled: Yes

- 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

▼ Instance type [Info](#) | [Get advice](#)

Instance type

t2.micro

Free tier eligible

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

All generations

[Compare instance types](#)

Additional costs apply for AMIs with pre-installed software

- 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.

▼ 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*

Default value ▾

 [Create new key pair](#)



Create key pair

Key pair name 1
Key pairs allow you to connect to your instance securely.

ECE4150 2

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

Key pair type 3

RSA
RSA encrypted private and public key pair

ED25519
ED25519 encrypted private and public key pair

Private key file format 3

.pem
For use with OpenSSH

.ppk
For use with PuTTY

⚠ 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](#) 4

Cancel Create key pair

▼ 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 1

ECE4150 2

Create new key pair 3

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:

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. **X**

▼ Configure storage [Info](#)

[Advanced](#)

1x GIB Root volume (Not encrypted)

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

[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](#)

► Advanced details [Info](#)

▼ Summary

Number of Instances | [Info](#)

1



[Software Image \(AMI\)](#)

Ubuntu Server 20.04 LTS (HVM),...[read more](#)
ami-06aa3f7caf3a30282

[Virtual server type \(instance type\)](#)

t2.micro

[Firewall \(security group\)](#)

New security group

Click Here

[Storage \(volumes\)](#)

1 volume(s) - 8 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 I/Os, 1 GB of snapshots, and 100 GB of bandwidth to the internet. X

[Cancel](#)

Launch instance

[Review commands](#)

- 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

S | Services | Search [Alt+S] | N. Virginia ▾ sam.cloud ▾

EC2 > Instances > Launch an instance

Success
Successfully initiated launch of instance (i-0564c2c9b5c80185f)

▶ Launch log

Next Steps

What would you like to do next with this instance, for example "create alarm" or "cr" < 1 2 3 4 5 6 7 8 >

Create billing and free tier usage alerts

To manage costs and avoid surprise bills, set up email notifications for billing and free tier usage thresholds.

[Create billing alerts](#)

Connect to your instance

Once your instance is running, log into it from your local computer.

[Connect to instance](#)

[Learn more](#)

Connect an RDS database

Configure the connection between an EC2 instance and a database to allow traffic flow between them.

[Connect an RDS database](#)

[Create a new RDS database](#)

[Learn more](#)

Create EBS snapshot policy

Create a policy that automates the creation, retention, and deletion of EBS snapshots

[Create EBS snapshot policy](#)

Manage detailed monitoring

Enable or disable detailed monitoring for the instance. If you enable detailed monitoring, the Amazon EC2 console displays monitoring graphs with a 1-minute period.

[Manage detailed monitoring](#)

Create Load Balancer

Create a application, network gateway or classic Elastic Load Balancer

[Create Load Balancer](#)



[View all instances](#)

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

The screenshot shows the AWS EC2 Instances page. On the left, a sidebar lists various EC2-related options like Dashboard, Global View, Events, and Instances. The Instances section is expanded, showing sub-options such as Instances, Instance Types, Launch Templates, and Spot Requests. The main pane displays a table of instances. A single row is selected, showing the instance ID i-0564c2c9b5c80185f, the name ECE4150-EC2, the state as Running, and the type t2.micro. A red arrow points from the bottom right towards the 'Running' status indicator.

	Name	Instance ID	Instance state	Instance type	Status check
<input type="checkbox"/>	ECE4150-EC2	i-0564c2c9b5c80185f	Running	t2.micro	Initializing

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

The screenshot shows the same EC2 Instances page after renaming the instance. The instance's name has been changed to "150-Lab2". A red arrow points from the bottom right towards the new name. The rest of the interface remains the same, showing the instance in the running state.

	Name	Instance ID	Instance state	Instance type	Status check
<input checked="" type="checkbox"/>	150-Lab2	i-0564c2c9b5c80185f	Running	t2.micro	2/2 check

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

Instances (1/1) [Info](#)

[Connect](#) [Instance state ▾](#)

[Find Instance by attribute or tag \(case-sensitive\)](#)

Any state

[Name](#) [Instance ID](#)

<input checked="" type="checkbox"/>	Name	Instance ID
<input checked="" type="checkbox"/>	ECE4150-Lab2	i-0564c2c9b5c80185f

[Actions ▾](#) [Launch instances ▾](#)

- [Connect](#)
- [View details](#)
- [Manage instance state](#)
- [Instance settings](#)
- [Networking](#)
- [Security](#)
- [Image and templates](#)
- [Monitor and troubleshoot](#)

Instance type [m2.micro](#) Status check [2/2 check](#)

Instances

- [Instances](#)
- [Instance Types](#)
- [Launch Templates](#)
- [Spot Requests](#)
- [Savings Plans](#)
- [Reserved Instances](#)
- [Dedicated Hosts](#)
- [Capacity Reservations](#) [New](#)

Images

- [AMIs](#)

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

EC2 > Instances > [i-0564c2c9b5c80185f](#) > Connect to instance

Connect to instance [Info](#)

Connect to your instance i-0564c2c9b5c80185f (ECE4150-Lab2) using any of these options

[EC2 Instance Connect](#) [Session Manager](#) [SSH client](#) [EC2 serial console](#)

No associated key pair

This instance is not associated with a key pair. Without a key pair, you can't connect to the instance through SSH.

You can connect using EC2 Instance Connect with just a valid username. You can connect using Session Manager if you have been granted the necessary permissions.

Instance ID
 [i-0564c2c9b5c80185f \(ECE4150-Lab2\)](#)

1. Open an SSH client.
2. Locate your private key file. The key used to launch this instance is id_rsa
3. Run this command, if necessary, to ensure your key is not publicly viewable.
 chmod 400 "id_rsa"
4. Connect to your instance using its Public DNS:
 ec2-44-203-73-228.compute-1.amazonaws.com

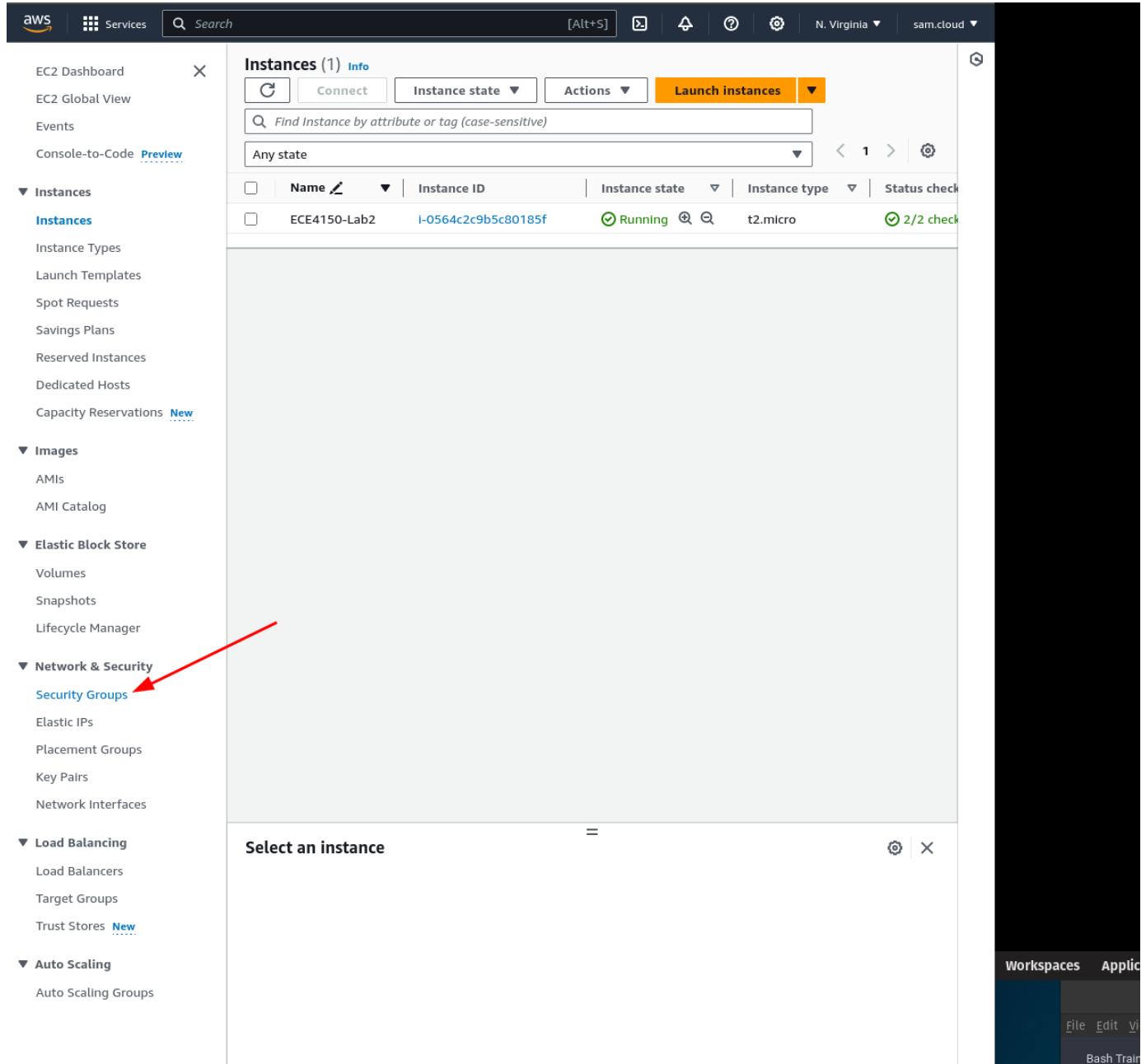
Example:
 ssh -i "id_rsa" ubuntu@ec2-44-203-73-228.compute-1.amazonaws.com

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

[Cancel](#)

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

The screenshot shows the AWS EC2 Security Groups page. On the left, there's a sidebar with 'Instances' expanded, showing options like Instances, Instance Types, Launch Templates, etc. The main area displays 'Security Groups (2)'. It has a search bar, an 'Actions' dropdown, and a 'Create security group' button highlighted with a red arrow. Below is a table with columns: Name, Security group ID, and Security group name. Two rows are listed: one named '-' with ID 'sg-05682e4eb870bf62b' and name 'default', and another with ID 'sg-0c565585202eacc31' and name 'launch-wizard-1'.

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

The screenshot shows the 'Create security group' wizard. The 'Basic details' step is selected. A red arrow labeled '1' points to the 'Security group name' field, which contains 'ECE4150-5G'. Another red arrow labeled '2' points to the 'Description' field, which contains 'Allows SSH and HTTP access to instance'. A third red arrow points from the 'VPC Info' dropdown, which contains 'vpc-01ba9790c80584d69', to the right with the text 'Do not change (default)'.

- 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**

Inbound rules [Info](#)

Inbound rule 1

Type Info SSH	Protocol Info TCP	Port range Info 22	Delete
Source type Info Anywhere-IPv4	Source Info <input type="text" value="0.0.0.0/0"/> X	Description - optional Info <input type="text"/>	
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. [X](#)

Outbound rules [Info](#)

Do not change anything from Outbound rules

Outbound rule 1

Type Info All traffic	Protocol Info All	Port range Info All	Delete
Destination type Info Custom	Destination Info <input type="text"/> X	Description - optional Info <input type="text"/>	
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. [X](#)

Tags - optional

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

No tags associated with the resource.

[Add new tag](#)

You can add up to 50 more tags

[Cancel](#) [Create security group](#)

- 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**

Instances (1/1) [Info](#)

[Connect](#) [Instance state ▾](#)

[Find Instance by attribute or tag \(case-sensitive\)](#)

Any state

[Name](#) [Instance ID](#)

ECE4150-Lab2 i-0564c2c9b5c80185f

Actions [Launch instances](#)

- Connect
- View details
- Manage instance state
- Instance settings
- Networking
- Security** [Change security groups](#)
- Image and templates
- Monitor and troubleshoot

Instance type [2.micro](#) Status check [2/2 check](#)

Get Windows password Modify IAM role

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
- Plastic IPs

- Remove the attached security group and add your new security group, then click “Save”

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	Network interface ID
i-0564c2c9b5c80185f (ECE4150-Lab2)	eni-0f8263a72fb1d5708

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
Select security groups	Add security group

Security groups associated with the network interface (eni-0f8263a72fb1d5708)

Security group name	Security group ID
launch-wizard-1	sg-0c565585202eacc31

[Remove](#)

[Cancel](#) [Save](#)

aws 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.

1. Search Our rule (ECE4150...) Add the security group

Select security groups Add security group

default (sg-03682e4eb870bf62b)
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

aws 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**”
- Launch 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

ORACLE

Microsoft SQL Server



IBM Db2

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 

Microsoft SQL Server 

IBM Db2 

The screenshot shows the AWS RDS MySQL Community Edition configuration page. At the top, the engine is set to MySQL Community. A blue box highlights the 'Known issues/limitations' section, which links to a detailed page about compatibility issues. Below this, the 'Engine version' is set to MySQL 8.0.28, indicated by a red arrow. The 'Templates' section shows three options: Production, Dev/Test, and Free tier, with Free tier selected (indicated by a blue circle and a red arrow). The 'Availability and durability' section lists deployment options: Single DB Instance, Multi-AZ DB instance, and Multi-AZ DB Cluster, with Single DB Instance selected.

Edition

MySQL Community

Known issues/limitations

Review the [Known issues/limitations](#) to learn about potential compatibility issues with specific database versions.

Engine version [Info](#)

View the engine versions that support the following database features.

▼ Hide filters

Show versions that support the Multi-AZ DB cluster [Info](#)

Create a Multi-AZ DB cluster with one primary DB instance and two readable standby DB instances. Multi-AZ DB clusters provide up to 2x faster transaction commit latency and automatic failover in typically under 35 seconds.

Show versions that support the Amazon RDS Optimized Writes [Info](#)

Amazon RDS Optimized Writes improves write throughput by up to 2x at no additional cost.

Engine Version

MySQL 8.0.28

Templates

Choose a sample template to meet your use case.

Production

Use defaults for high availability and fast, consistent performance.

Dev/Test

This instance is intended for development use outside of a production environment.

Free tier

Use RDS Free Tier to develop new applications, test existing applications, or gain hands-on experience with Amazon RDS.

[Info](#)

Availability and durability

Deployment options [Info](#)

The deployment options below are limited to those supported by the engine you selected above.

Single DB Instance (not supported for Multi-AZ DB cluster snapshot)

Creates a single DB instance with no standby DB instances.

Multi-AZ DB instance (not supported for Multi-AZ DB cluster snapshot)

Creates a primary DB instance and a standby DB instance in a different AZ. Provides high availability and data redundancy, but the standby DB instance doesn't support connections for read workloads.

Multi-AZ DB Cluster

Creates a DB cluster with a primary DB instance and two readable standby DB instances, with each DB instance in a different Availability Zone (AZ). Provides high availability, data redundancy and increases capacity to serve read workloads.

- 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.

Settings

DB instance identifier [Info](#)

Type a name for your DB instance. The name must be unique across all DB instances owned by your AWS account in the current AWS Region.

 1

The DB instance identifier is case-insensitive, but is stored as all lowercase (as in "mydbinstance"). Constraints: 1 to 60 alphanumeric characters or hyphens. First character must be a letter. Can't contain two consecutive hyphens. Can't end with a hyphen.

▼ Credentials Settings

Master username [Info](#)

Type a login ID for the master user of your DB instance.

 2

1 to 16 alphanumeric characters. The first character must be a letter.

Manage master credentials in AWS Secrets Manager

Manage master user credentials in Secrets Manager. RDS can generate a password for you and manage it throughout its lifecycle.

 If you manage the master user credentials in Secrets Manager, some RDS features aren't supported.

[Learn more](#) 

Auto generate a password

Amazon RDS can generate a password for you, or 3 you can specify your own password.

Master password [Info](#)

 3

Constraints: At least 8 printable ASCII characters. Can't contain any of the following: / (slash), ' (single quote), " (double quote) and @ (at sign).

Confirm master password [Info](#)

 4

- Follow the default configurations as shown below

The screenshot shows the AWS RDS Instance Configuration page. At the top, there are tabs for Services, Search, and [Alt+S]. On the right, it shows N. Virginia and sam.cloud.

Instance configuration

The DB instance configuration options below are limited to those supported by the engine that you selected above.

DB Instance class [Info](#)

▼ Hide filters

Include previous generation classes

Standard classes (includes m classes)
 Memory optimized classes (includes r and x classes)
 Burstable classes (includes t classes)

db.t3.micro
2 vCPUs 1 GiB RAM Network: 2,085 Mbps

Storage

Storage type [Info](#)

General Purpose SSD (gp2)
Baseline performance determined by volume size

Allocated storage [Info](#)

20 GiB
The minimum value is 20 GiB and the maximum value is 6,144 GiB

ⓘ After you modify the storage for a DB instance, the status of the DB instance will be in storage-optimization. Your instance will remain available as the storage-optimization operation completes.
[Learn more](#)

▼ Storage autoscaling

Storage autoscaling [Info](#)

Provides dynamic scaling support for your database's storage based on your application's needs.

Enable storage autoscaling
Enabling this feature will allow the storage to increase after the specified threshold is exceeded.

Maximum storage threshold [Info](#)

Charges will apply when your database autoscales to the specified threshold

1000 GiB
The minimum value is 22 GiB and the maximum value is 6,144 GiB

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

Connectivity Info



Compute resource

Choose whether to set up a connection to a compute resource for this database. Setting up a connection will automatically change connectivity settings so that the compute resource can connect to this database.

Don't connect to an EC2 compute resource

Don't set up a connection to a compute resource for this database. You can manually set up a connection to a compute resource later.

Connect to an EC2 compute resource

Set up a connection to an EC2 compute resource for this database.

Network type Info

To use dual-stack mode, make sure that you associate an IPv6 CIDR block with a subnet in the VPC you specify.

IPv4

Your resources can communicate only over the IPv4 addressing protocol.

Dual-stack mode

Your resources can communicate over IPv4, IPv6, or both.

Virtual private cloud (VPC) Info

Choose the VPC. The VPC defines the virtual networking environment for this DB instance.

Default VPC (vpc-01ba9790c80584d69)

6 Subnets, 6 Availability Zones

Only VPCs with a corresponding DB subnet group are listed.

i After a database is created, you can't change its VPC.

DB subnet group Info

Choose the DB subnet group. The DB subnet group defines which subnets and IP ranges the DB instance can use in the VPC that you selected.

default

Public access Info

Yes

RDS assigns a public IP address to the database. Amazon EC2 instances and other resources outside of the VPC can connect to your database. Resources inside the VPC can also connect to the database. Choose one or more VPC security groups that specify which resources can connect to the database.

No

RDS doesn't assign a public IP address to the database. Only Amazon EC2 instances and other resources inside the VPC can connect to your database. Choose one or more VPC security groups that specify which resources can connect to the database.

- Attach the security group that you created previously

VPC security group (firewall) [Info](#) 1

Choose one or more VPC security groups to allow access to your database. Make sure that the security group rules allow the appropriate incoming traffic.

Choose existing
 Choose existing VPC security groups

Create new
 Create new VPC security group

Existing VPC security groups

Choose one or more options ▾

ECE4150-5G X 2

Availability Zone [Info](#)

No preference ▾

RDS Proxy

RDS Proxy is a fully managed, highly available database proxy that improves application scalability, resiliency, and security.

Create an RDS Proxy [Info](#)

RDS automatically creates an IAM role and a Secrets Manager secret for the proxy. RDS Proxy has additional costs. For more information, see [Amazon RDS Proxy pricing](#).

Certificate authority - *optional* [Info](#) 3

Using a server certificate provides an extra layer of security by validating that the connection is being made to an Amazon database. It does so by checking the server certificate that is automatically installed on all databases that you provision.

rds-ca-2019 (default)
 Expiry: Aug 22, 2024 ▾

If you don't select a certificate authority, RDS chooses one for you.

► Additional configuration

- Database Authentication

Database authentication

Database authentication options [Info](#)

Password authentication
 Authenticates using database passwords.

Password and IAM database authentication
 Authenticates using the database password and user credentials through AWS IAM users and roles.

Password and Kerberos authentication
 Choose a directory in which you want to allow authorized users to authenticate with this DB instance using Kerberos Authentication.

- Under Monitoring, “Enable Enhanced Monitoring” as shown below

Monitoring

Enable Enhanced Monitoring
Enabling Enhanced Monitoring metrics are useful when you want to see how different processes or threads use the CPU.

Granularity
60 seconds ▾

Monitoring Role
default ▾
Clicking "Create database" will authorize RDS to create the IAM role rds-monitoring-role

The screenshot shows the AWS RDS "Create database" wizard. At the top, there's a navigation bar with the AWS logo, services menu, search bar, and account information (N. Virginia, sam.cloud). Below the navigation, the "Monitoring" section is visible, which includes "Enable Enhanced Monitoring" (checked), "Granularity" set to "60 seconds", and "Monitoring Role" set to "default". A note says "Clicking 'Create database' will authorize RDS to create the IAM role rds-monitoring-role". A red arrow points from the "Additional configuration" section to the "Initial database name" input field. The "Additional configuration" section contains a note about database options like encryption and backup. The "Database options" section is also shown, with the "Initial database name" field containing "photogallerydb" highlighted by a red box. A note below says "If you do not specify a database name, Amazon RDS does not create a database."

- 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.

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

DB identifier	Status	Role	Engine	Region & AZ
photogallerydb	Creating	Instance	MySQL Community	us-east-1d

The screenshot shows the AWS RDS (Amazon Relational Database Service) console. On the left, there's a sidebar with various navigation links like Dashboard, Databases, Query Editor, etc. The main area displays a success message: "Successfully created database photogallerydb". It also features an introductory message about Aurora I/O-Optimized. Below these messages, the "Databases" section is shown, containing a table with one row for the newly created database.

DB identifier	Status	Role	Engine
photogallerydb	Configuring-enhanced-monitoring	Instance	MySQL Co

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

The screenshot shows the AWS RDS console for the 'photogallerydb' database. The left sidebar lists various RDS management options like Dashboard, Databases, Query Editor, etc. The main area displays the 'Summary' tab for the database, showing metrics such as DB Identifier (photogaller ydb), Status (Backing up), Role (Current activity), Engine (MySQL), and Recommendations (Region & AZ: us-east-1d). Below the summary is a navigation bar with tabs: Connectivity & security (selected), Monitoring, Logs & events, Configuration, and a search bar. The 'Connectivity & security' tab shows the endpoint and port details, with the endpoint 'photogallerydb.cbkym040ofou.us-east-1.rds.amazonaws.com' highlighted by a red box. To the right, networking details like Availability Zone (us-east-1d), VPC (vpc-01ba9790c80584d69), and Subnet group (default-vpc-01ba9790c80584d69) are listed.

- 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 [New](#)
- Elastic IPs
- Placement Groups
- Key Pairs
- Network Interfaces

Load Balancing

- Load Balancers
- Target Groups
- Trust Stores [New](#)

Auto Scaling

- Auto Scaling Groups

Resources

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.

Service health

AWS Health Dashboard

Regions

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](#)

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

Edit inbound rules

Action	Security group ID	Security group name
Edit inbound rules	582e4eb870bf62b	default
Edit outbound rules	988a9fe8116a289	ECE4150-5G
Manage tags	65585202eacc31	launch-wizard-1
Manage stale rules		
Copy to new security group		
Delete security groups		

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 <small>Info</small> SSH	Protocol <small>Info</small> TCP
Port range <small>Info</small> 22	Source type <small>Info</small> Custom	Source <small>Info</small> <input type="text"/> 0.0.0.0/0 X
Description - optional <small>Info</small> <input type="text"/>		

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. X

Cancel Preview changes Save rules

Edit inbound rules [Info](#)

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

Inbound rules [Info](#)

Inbound rule 1

Security group rule ID
sgr-0d55d80c938ef7532

Type [Info](#)
SSH

Port range [Info](#)
22

Protocol [Info](#)
TCP

Source type [Info](#)
Custom

Source [Info](#)
 

Description - optional [Info](#)

Inbound rule 2

Security group rule ID
sgr-0866aeee28143cf69

Type [Info](#)
MYSQL/Aurora

Port range [Info](#)
3306

Protocol [Info](#)
TCP

Source type [Info](#)
Custom

Source [Info](#)
 

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).

DynamoDB

Dashboard

Tables

Update settings

Explore items

PartiQL editor

Backups

Exports to S3

Imports from S3

Integrations New

Reserved capacity

Settings

Tables (0) Info

Actions ▾ Delete Create table

Find tables by table name Any tag key Any tag value

Na... Status Partition key Sort key Indexes Deletion protection

You have no tables in this account in this AWS Region.

Create table

aws Services Search [Alt+S]

DynamoDB > Tables > Create table

Create table

Table details Info

DynamoDB is a schemaless database that requires only a table name and a primary key when you create the table.

Table name
This will be used to identify your table.

Between 3 and 255 characters, containing only letters, numbers, underscores (_), hyphens (-), and periods (.)

Partition key
The partition key is part of the table's primary key. It is a hash value that is used to retrieve items from your table and allocate data across hosts for scalability and availability.
 String

1 to 255 characters and case sensitive.

Sort key - optional
You can use a sort key as the second part of a table's primary key. The sort key allows you to sort or search among all items sharing the same partition key.
 String

1 to 255 characters and case sensitive.

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

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](#)

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.

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](#)

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 the [Storage](#) tab of the [Amazon S3 pricing page](#).

Bucket Key

Using an S3 Bucket Key for SSE-KMS reduces encryption costs by lowering calls to AWS KMS. S3 Bucket Keys aren't supported for DSSE-KMS. [Learn more](#)

- Disable
- Enable

▼ Advanced settings

Object Lock

Store objects using a write-once-read-many (WORM) model to help you prevent objects from being deleted or overwritten for a fixed amount of time or indefinitely. Object Lock works only in versioned buckets. [Learn more](#)

- Disable
- Enable
Permanently allows objects in this bucket to be locked. Additional Object Lock configuration is required in bucket details after bucket creation to protect objects in this bucket from being deleted or overwritten.

 Object Lock works only in versioned buckets. Enabling Object Lock automatically enables Versioning.

 After creating the bucket, you can upload files and folders to the bucket, and configure additional bucket settings.

Cancel

Create bucket

- 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](#)

[C](#) [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

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

Amazon S3 > Buckets > photobucket-choephel-2024-4150

photobucket-choephel-2024-4150 [Info](#)

Objects Properties Permissions Metrics Management Access Points

Objects (2) [Info](#)

[C](#) [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

Name	Type	Last modified	Size	Storage class
photos/	Folder	-	-	-
thumbnails/	Folder	-	-	-

The screenshot shows the AWS S3 console interface. At the top, there's a navigation bar with the AWS logo, 'Services' dropdown, a search bar, and various global settings like 'Global' and 'sam.cloud'. Below the navigation bar, the path 'Amazon S3 > Buckets > photobucket-choephel-2024-4150' is displayed. The main title 'photobucket-choephel-2024-4150' has an 'Info' link next to it. Below the title, there are tabs for 'Objects' (which is selected), 'Properties', 'Metrics', 'Management', and 'Access Points'. A red arrow points from the text above to the 'Permissions' tab. Under the 'Objects' tab, there's a summary section for 'Objects (2)'. It includes buttons for 'Create folder' (disabled), 'Upload' (highlighted in orange), 'Copy S3 URI', 'Copy URL', 'Download', 'Open', 'Delete', and 'Actions'. Below this is a search bar with 'Find objects by prefix'. The main content area shows a table with two entries: 'photos/' and 'thumbnails/'. The table columns are 'Name', 'Type', 'Last modified', 'Size', and 'Storage class'. The 'photos/' entry is a folder, and the 'thumbnails/' entry is also a folder. There are checkboxes to the left of each row.

- 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"  
        ],  
        "MaxAge": 3600  
    },  
    {  
        "AllowedHeaders": [  
            "Authorization"  
        ],  
        "AllowedMethods": [  
            "PUT"  
        ],  
        "MaxAge": 3600  
    }  
]
```

```

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

```

Double Check your S3 Bucket config

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.

Object Ownership

Bucket owner preferred

ACLs are enabled and can be used to grant access to this bucket and its objects. If new objects written to this bucket specify the bucket-owner-full-control canned ACL, they are owned by the bucket owner. Otherwise, they are owned by the object writer.

[Edit](#)



Access control list (ACL)

Grant basic read/write permissions to other AWS accounts. [Learn more](#)

The console displays combined access grants for duplicate grantees
To see the full list of ACLs, use the Amazon S3 REST API, AWS CLI, or AWS SDKs.

Grantee	Objects	Bucket ACL
Bucket owner (your AWS account) Canonical ID: c8f04a5174295e6f31fa1d5aec5632d3ec82f9ef9df78056911c721d84d989d0	List, Write	Read, Write
Everyone (public access) Group: http://acs.amazonaws.com/groups/global/AllUsers	-	-
Authenticated users group (anyone with an AWS account) Group: http://acs.amazonaws.com/groups/global/AuthenticatedUsers	-	-
S3 log delivery group Group: http://acs.amazonaws.com/groups/s3/LogDelivery	-	-

[Edit](#)



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	Users	Roles	Policies	Identity providers
0	0	6	0	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

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

This screenshot shows the 'Specify user details' step of the AWS IAM 'Create user' wizard. The 'User name' field contains 'ec2_instance_access'. A note below the field states: 'The user name can have up to 64 characters. Valid characters: A-Z, a-z, 0-9, and + = , . @ _ - (hyphen)'. There is an optional checkbox for 'Provide user access to the AWS Management Console'. A callout box provides instructions for generating programmatic access keys for AWS CodeCommit or Amazon Keyspaces. At the bottom, there are 'Cancel' and 'Next' buttons.

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.

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.

Copy permissions
Copy all group memberships, attached managed policies, and inline policies from an existing user.

Permissions policies (2/1174)

Choose one or more policies to attach to your new user.

Filter by Type

Policy name	Type
<input checked="" type="checkbox"/> AmazonDynamoDBFullAccess	AWS managed
<input type="checkbox"/> AmazonDynamoDBFullAccesswithDataPip...	AWS managed
<input type="checkbox"/> AmazonDynamoDBReadOnlyAccess	AWS managed
<input type="checkbox"/> AWSApplicationAutoscalingDynamoDBTab...	AWS managed
<input type="checkbox"/> AWSLambdaDynamoDBExecutionRole	AWS managed
<input type="checkbox"/> AWSLambdaInvocation-DynamoDB	AWS managed
<input type="checkbox"/> DynamoDBCloudWatchContributorInsight...	AWS managed
<input type="checkbox"/> DynamoDBKinesisReplicationServiceRoleP...	AWS managed
<input type="checkbox"/> DynamoDBReplicationServiceRolePolicy	AWS managed

▶ Set permissions boundary - optional

Cancel Previous Next



Screenshot of the AWS IAM 'Create user' review step.

The 'User details' section shows:

- User name: ec2_instance_access
- Console password type: None (highlighted with a red arrow)
- Require password reset: No

A note at the bottom right says: "In the next step, we need to change this".

The 'Permissions summary' section shows:

Name	Type	Used as
AmazonDynamoDBFullAccess	AWS managed	Permissions policy
AmazonS3FullAccess	AWS managed	Permissions policy

The 'Tags - optional' section shows:

- No tags associated with the resource.
- Add new tag button.
- You can add up to 50 more tags.

At the bottom right are 'Cancel', 'Previous', and a large orange 'Create user' button (highlighted with a red arrow).

Screenshot of the AWS IAM 'Users' list page.

The left sidebar shows:

- Identity and Access Management (IAM)
- Dashboard
- Access management
 - User groups
 - Users** (highlighted)
 - Roles
 - Policies
 - Identity providers

The main area shows:

Users (1) info

An IAM user is an identity with long-term credentials that is used to interact with AWS in an account.

User name	Path	Group	Last activity
ec2_instance_access	/	0	

A red arrow points to the 'ec2_instance_access' link in the User name column, with the text "Click on this" below it.

S | Services | Search | [Alt+S] | X | ⚡ | ⓘ | ⓘ | 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)

Permissions are defined by policies attached to the user directly or through groups.

Filter by Type All types

Policy name	Type	Attached via
AmazonDynamoDB...	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.

Screenshot of the AWS IAM 'Create access key' wizard Step 1: Access key best practices & alternatives.

The 'Use case' section shows several options:

- 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.

A red arrow points to the 'Application running on an AWS compute service' option. A red box highlights the 'Alternative recommended' note: "Assign an IAM role to compute resources like EC2 instances or Lambda functions to automatically supply temporary credentials to enable access." A red box also highlights the 'Confirmation' checkbox: "I understand the above recommendation and want to proceed to create an access key."

Click Next

Screenshot of the AWS IAM 'Create access key' wizard Step 2: Set description tag - optional.

The 'Set description tag' section shows:

- Step 1: Access key best practices & alternatives
- Step 2 - optional: Set description tag
- Step 3: Retrieve access keys

The 'Description tag value' section contains:

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

The screenshot shows the AWS IAM 'Create access key' page. At the top, there's a green banner with the message: 'Access key created. This is the only time that the secret access key can be viewed or downloaded. You cannot recover it later. However, you can create a new access key any time.' Below this, the breadcrumb navigation shows: IAM > Users > ec2_instance_access > Create access key. The main section is titled 'Retrieve access keys' with an 'Info' link. It contains a table with two columns: 'Access key' and 'Secret access key'. The 'Access key' column shows 'AKIATCKATVG576GTFPPQ' with a copy icon. The 'Secret access key' column shows '*****' followed by a redacted string and a 'Show' link. A red box highlights this table. To the left, there are three steps: Step 1 (Access key best practices & alternatives), Step 2 (optional Set description tag), and Step 3 (Retrieve access keys). Step 3 has a red box around the 'Save the key somewhere safe' link. On the right, there's a 'Access key best practices' section with a bulleted list of recommendations and a link to 'best practices for managing AWS access keys'. At the bottom right are 'Download .csv file' and 'Done' buttons.

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

- In the files 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.

```
env.py - ECE4813-Lab2 - Visual Studio Code
File Edit Selection View Go Run Terminal Help
env.py x photobucket-choephel-2024-4150 Untitled-1 ...
utils > env.py > ...
1 # IAM User access configuration
2 AWS_ACCESS_KEY="AKIATCKATV[REDACTED]"
3 AWS_SECRET_ACCESS_KEY="RhVuXYVg9emAxwv02pdt0oc[REDACTED]"
4 AWS_REGION="us-east-1"
5
6 # S3 Bucket for photo objects
7 PHOTOGALLERY_S3_BUCKET_NAME="photobucket-[REDACTED]-2024-4150"
8
9 # MySQL Configuration
10 RDS_DB_HOSTNAME='<>'
11 RDS_DB_USERNAME='root'
12 RDS_DB_PASSWORD='[REDACTED]'
13 RDS_DB_NAME='photogallerydb'
14
15 # DynamoDB Table
16 DYNAMODB_TABLE='PhotoGallery'
17
18 ##### INSERT NEW ENVIRONMENT VARIEABLES HERE #####
19
20 #####
21 #####
```

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 file 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 file 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 file transfer protocols. Go to <https://filezilla-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
```

```
tmux 116x27

System load: 0.0          Processes:         97
Usage of /: 21.4% of 7.57GB  Users logged in:      0
Memory usage: 22%          IPv4 address for eth0: 172.31.92.61
Swap usage:  0%          

Expanded Security Maintenance for Applications is not enabled.

0 updates can be applied immediately.

Enable ESM Apps to receive additional future security updates.
See https://ubuntu.com/esm or run: sudo pro status

The list of available updates is more than a week old.
To check for new updates run: sudo apt update
New release '22.04.3 LTS' available.
Run 'do-release-upgrade' to upgrade to it.

Last login: Fri Feb  2 00:35:50 2024 from 128.61.121.25
To run a command as administrator (user "root"), use "sudo <command>".
See "man sudo_root" for details.

ubuntu@ip-172-31-92-61:~$
```

Create a folder **data**, in ec2

```
tmux 116x27

ubuntu@ip-172-31-92-61:~$ mkdir data
ubuntu@ip-172-31-92-61:~$ ls
data
ubuntu@ip-172-31-92-61:~$ cd data/
ubuntu@ip-172-31-92-61:~/data$ pwd
/home/ubuntu/data
ubuntu@ip-172-31-92-61:~/data$ _
```

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
inputoho-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/

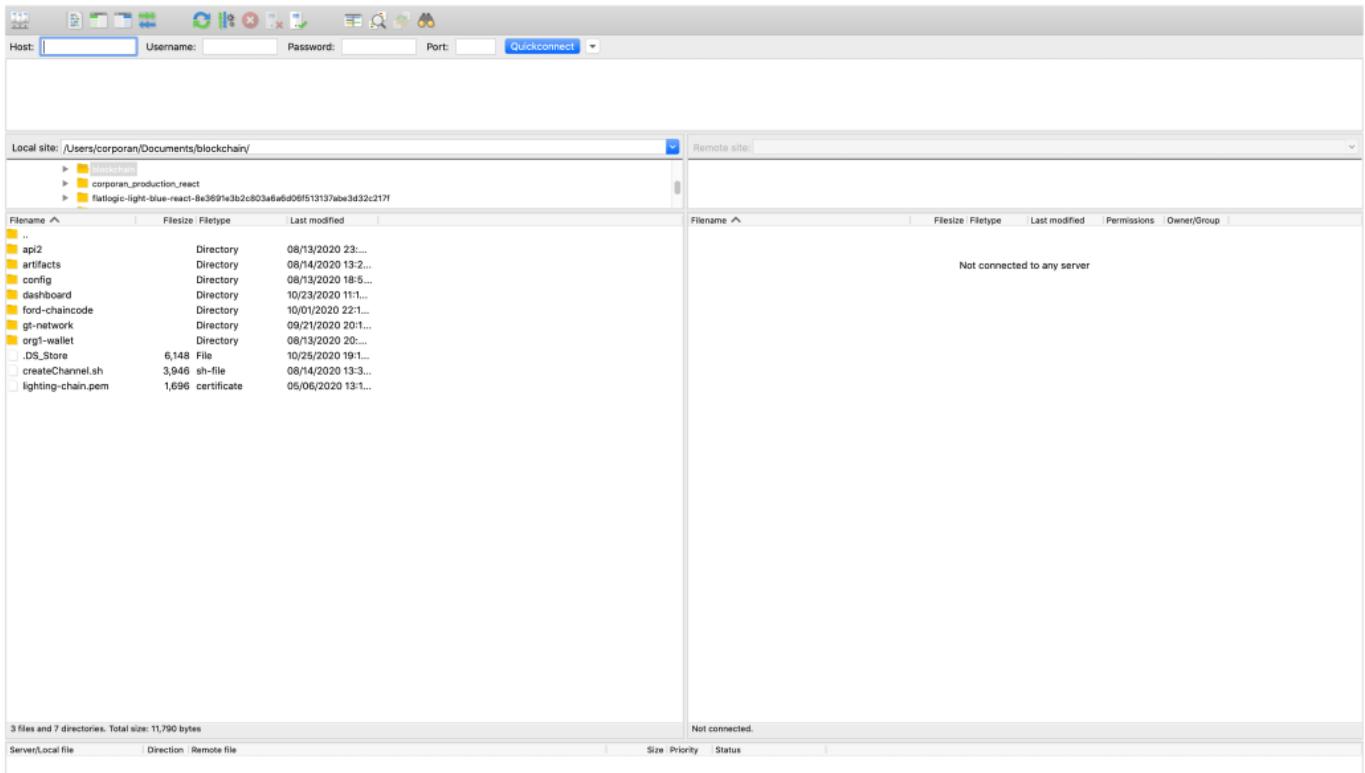
```

```

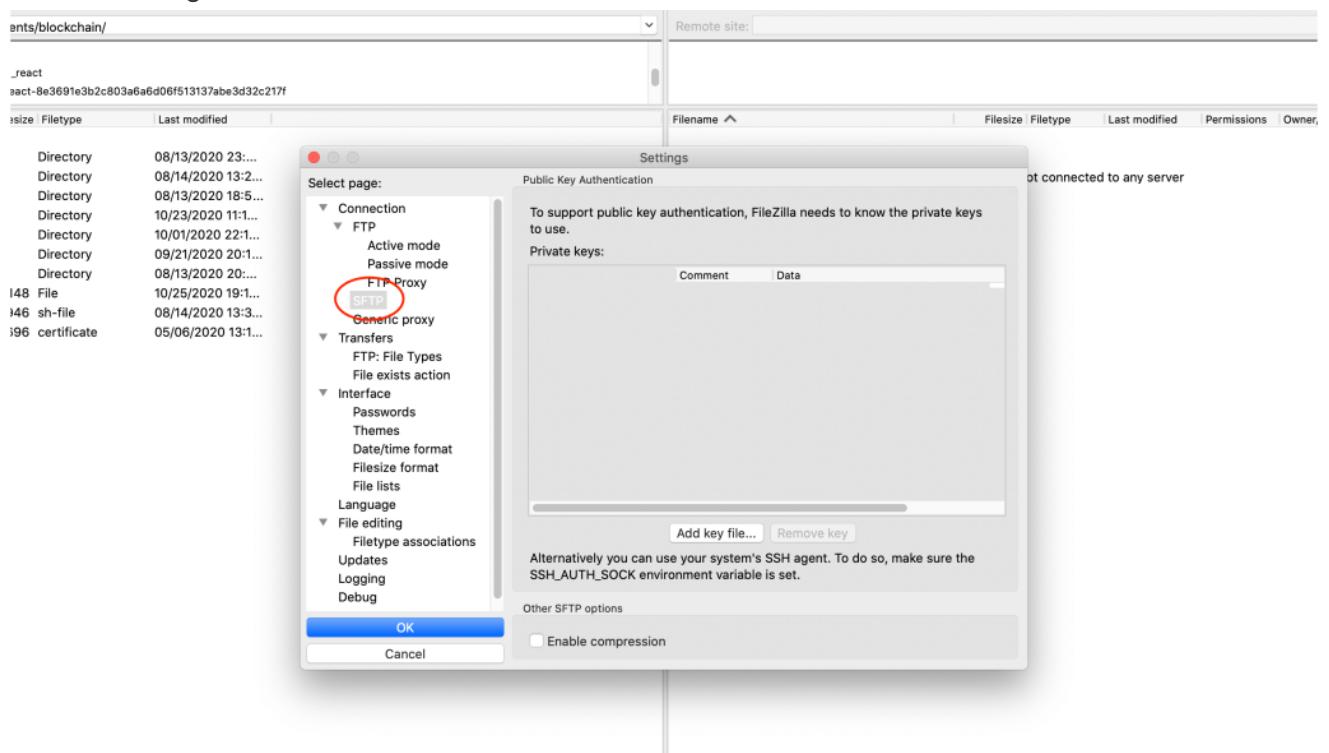
● → ECE4150-Lab2 scp -i ~/Downloads/ece4150.pem -r utils ubuntu@44.204.54.182:/home/ubuntu/data/
env.py                                           100% 587    20.4KB/s  00:00
sqlCommands.sql                                 100% 992    30.3KB/s  00:00
env.cpython-39.pyc                            100% 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

```

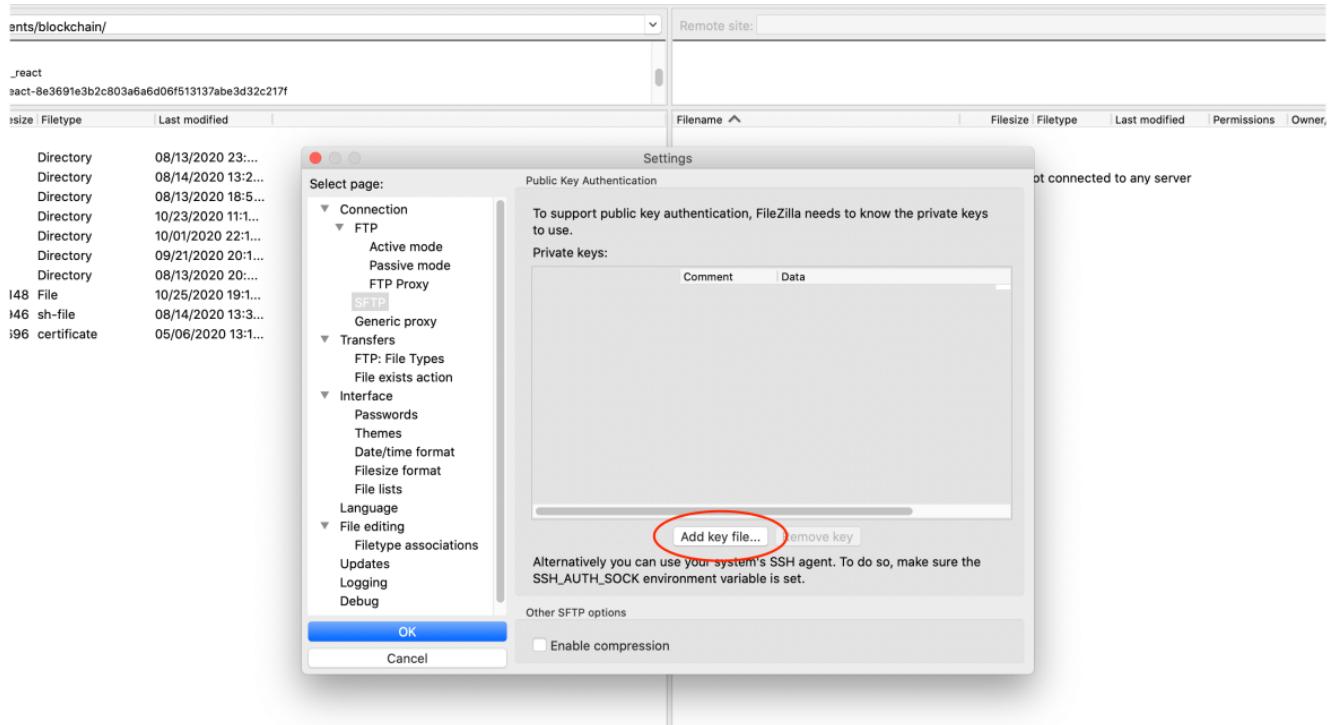
- If you want a more user-friendly tool to transfer data, FileZilla is the right choice. It is free, supports Windows/Linux/Mac systems, and has a friendly user interface. It supports FTP, SFTP, and other file transfer protocols. Go to <https://filezilla-project.org/> to download it.



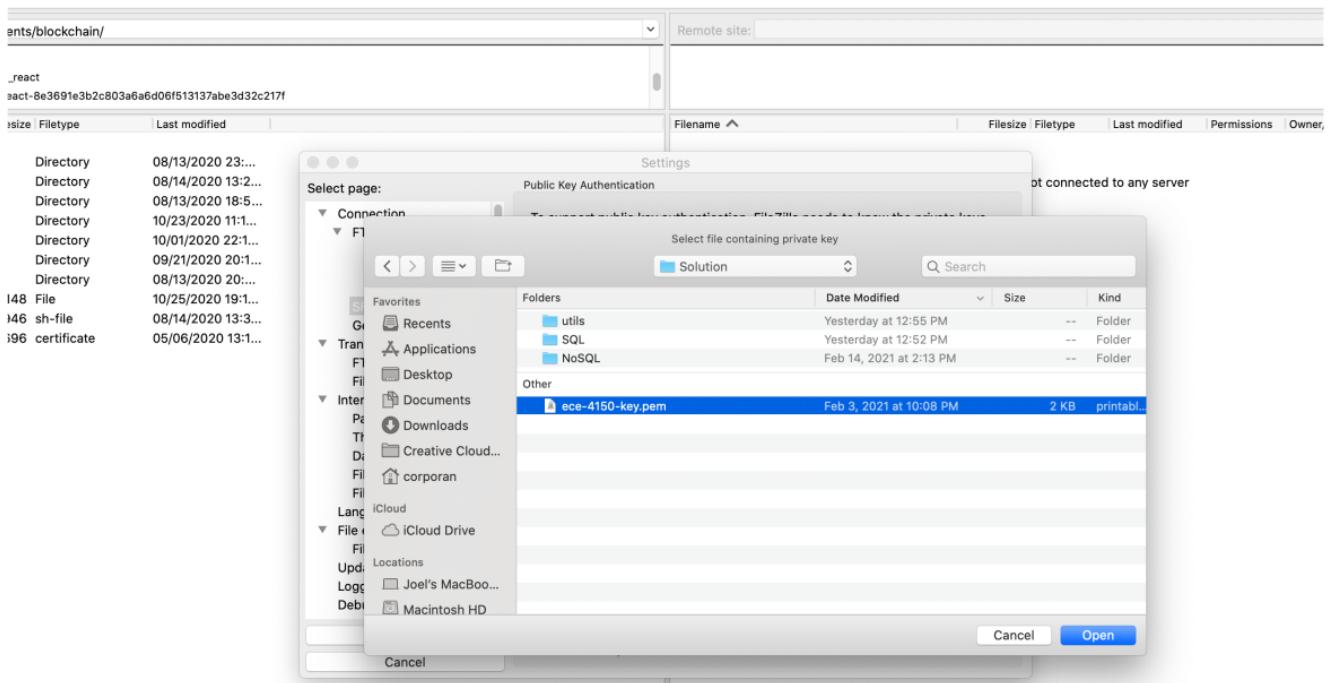
- If you want to use FileZilla to upload to or download data from a normal FTP server if you have the user and password, just put the information in the “Host”, “Username”, “Password” box and connect. However for Amazon instance, we use key-pair to log in instead of password for better safety. So it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



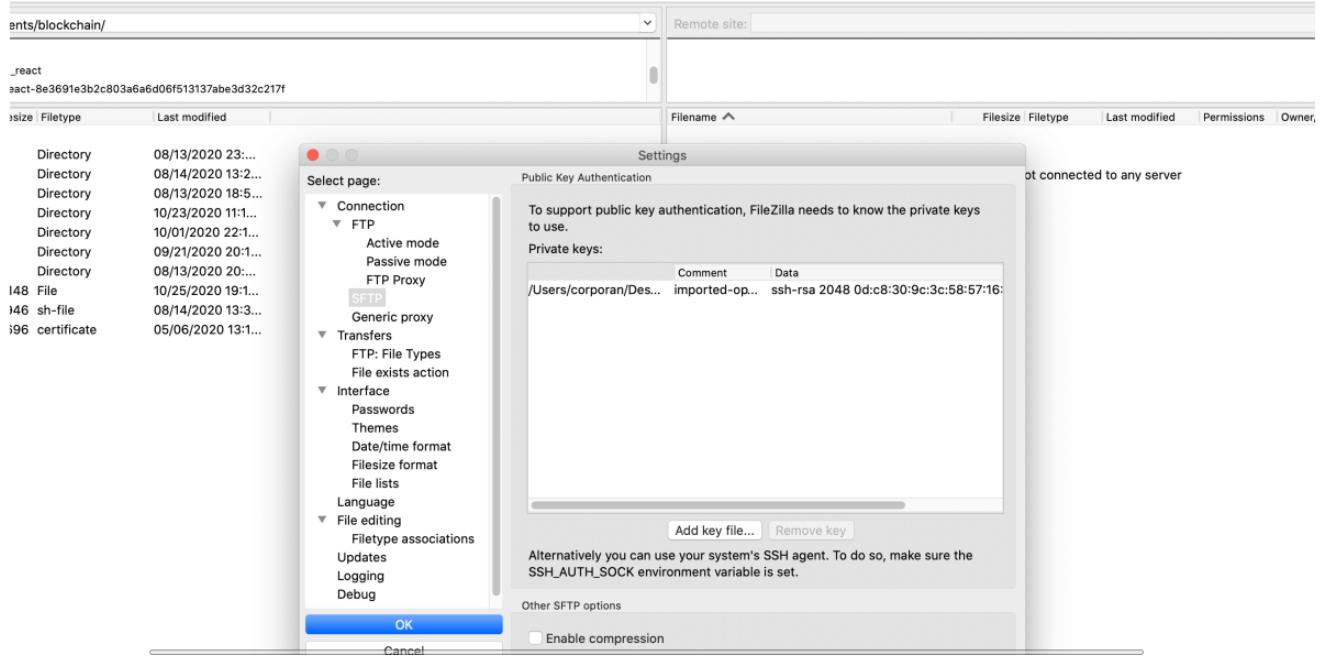
- Click “Add key file...”:



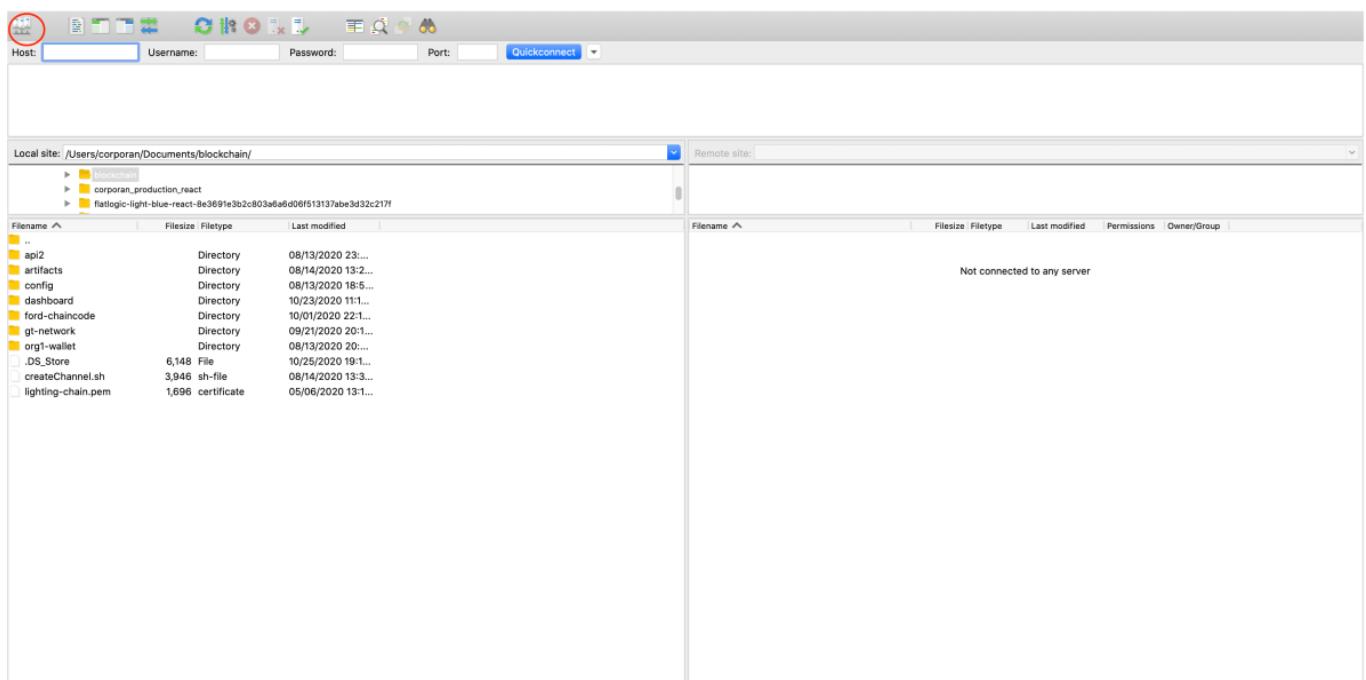
- Then select the ".pem" file you used to connect to Amazon instance using SSH.



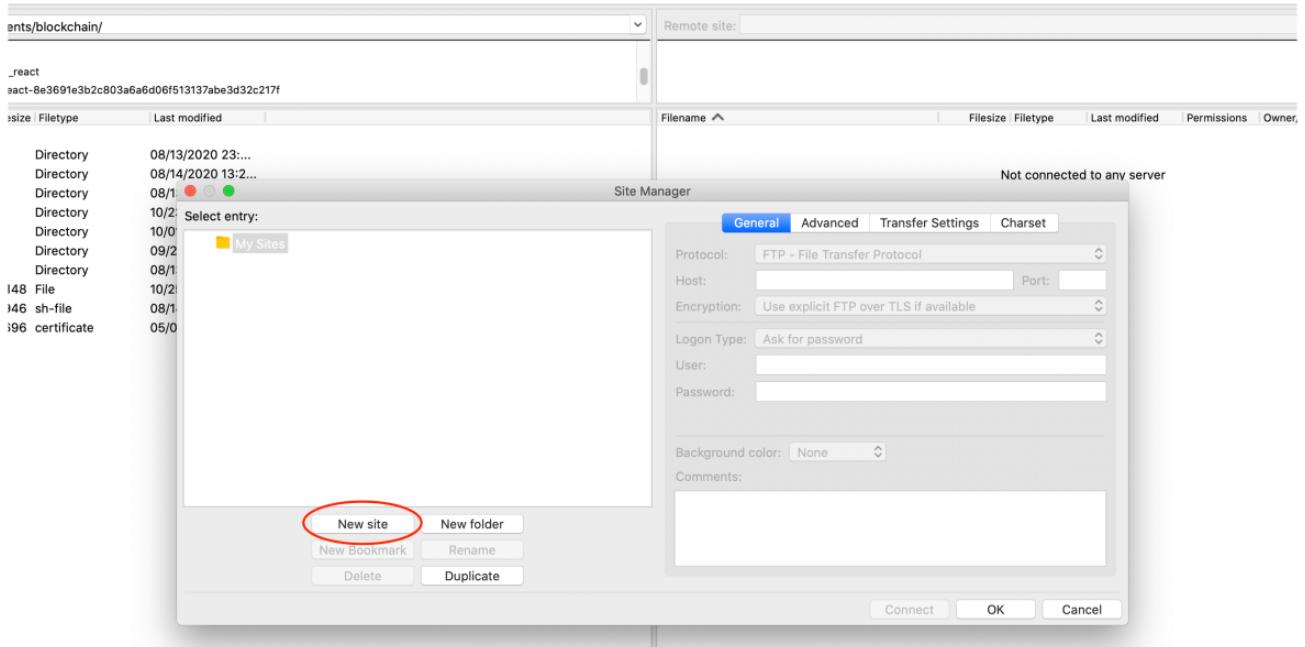
- You will see the a private key has been added.



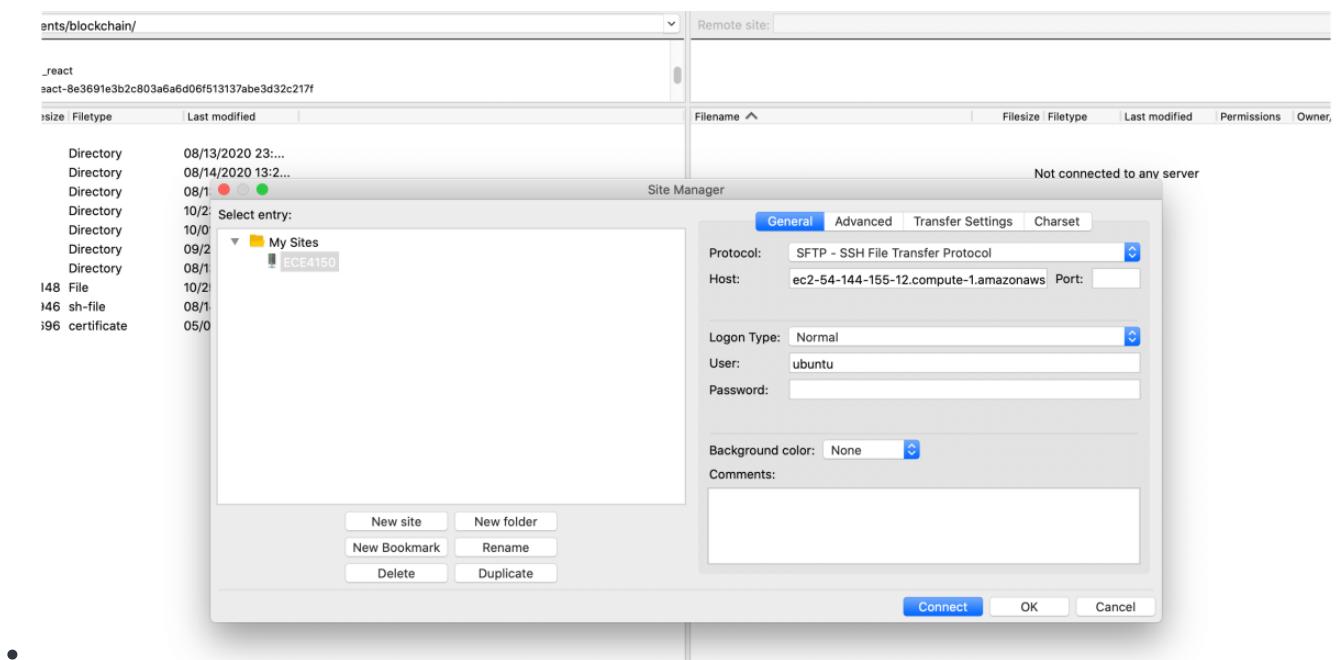
- Close “Settings” and go back to the main interface and click the button to open the “Site manager”, as shown below



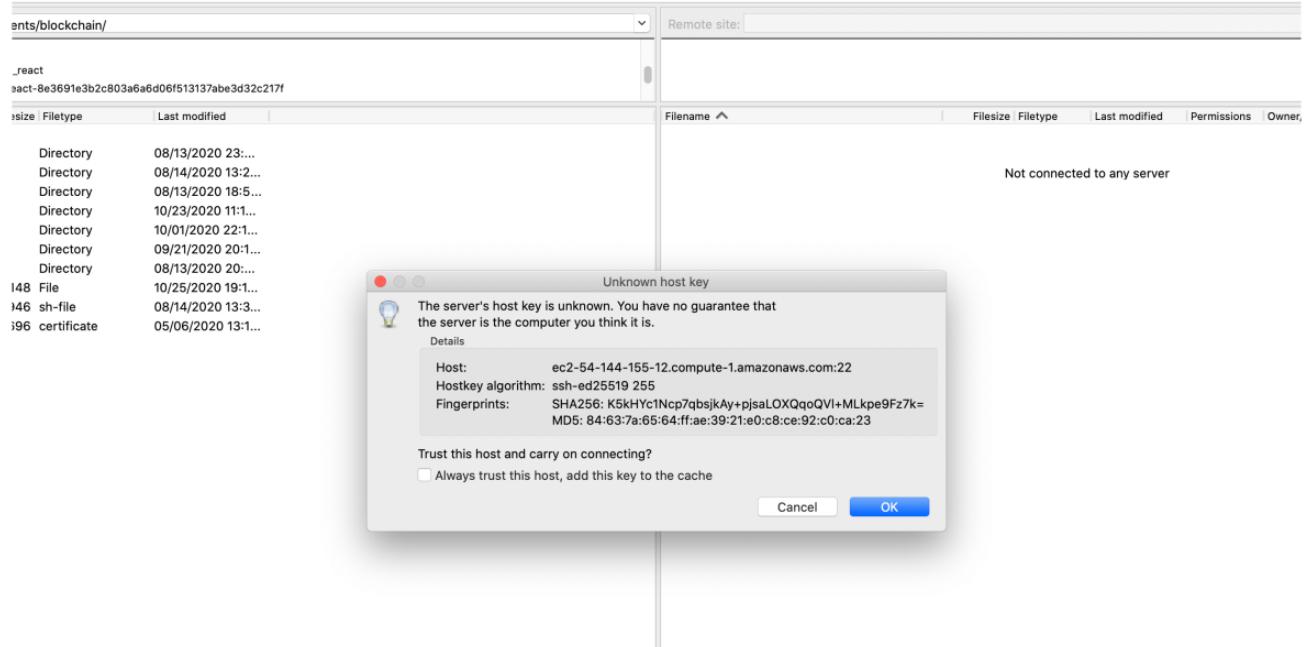
- Click “New Site” and give a name.



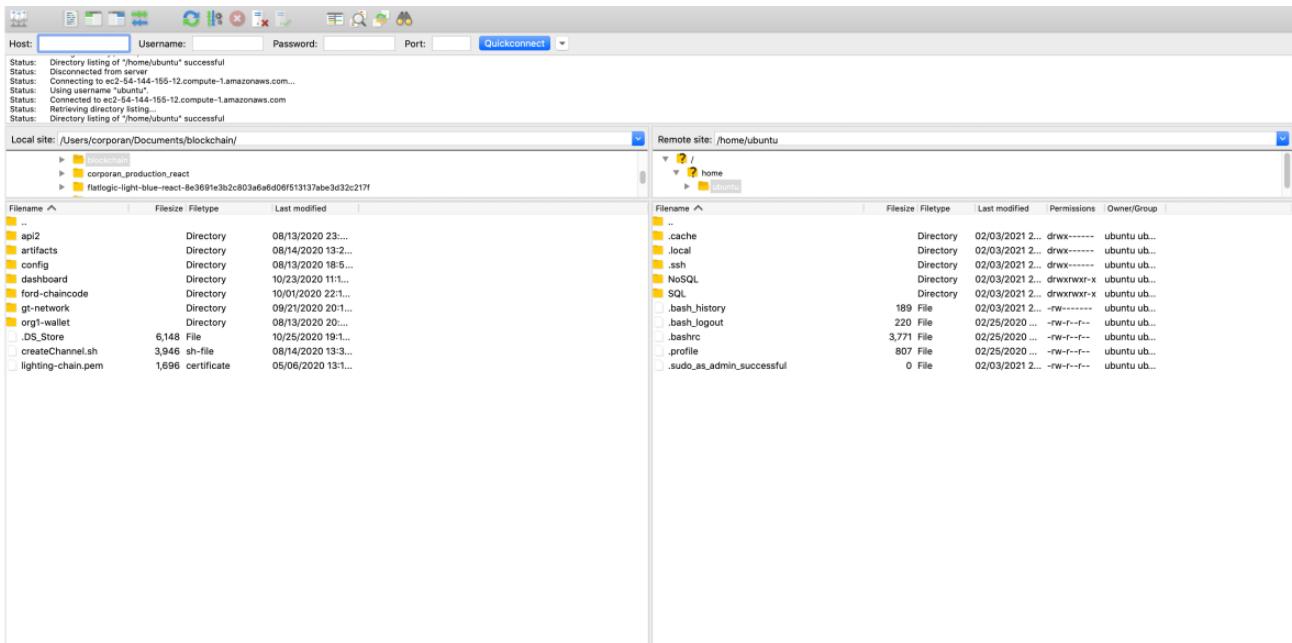
- Put the host URL of the EC2 instance in the “Host” box. Set “Protocol” as “SFTP”, “Logon Type” as “Normal”, “User” as “ubuntu” and leave “Password” as blank. Then click “Connect”.



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



- All right. Now you have logged in the EC2 instance. You can drag and drop to transfer the files between your 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 install 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 configure the source coming from anywhere (**0.0.0.0/0**). This configuration will allow you to access your application inside the EC2 through the 5000 port.

The screenshot shows the 'Inbound rule 4' configuration for a security group. It includes fields for Security group rule ID (empty), Type (Custom TCP), Protocol (TCP), Port range (5000), Source type (Anywhere-IPv4), and Source (0.0.0.0/0). A note at the bottom cautions against using 0.0.0.0/0 for security reasons. Buttons for 'Add rule', 'Cancel', 'Preview changes', and 'Save rules' are visible.

Inbound rule 4

Security group rule ID: -

Type: [Info](#) Custom TCP

Protocol: [Info](#) TCP

Port range: [Info](#) 5000

Source type: [Info](#) Anywhere-IPv4

Source: [Info](#) 0.0.0.0/0 [X](#)

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. [X](#)

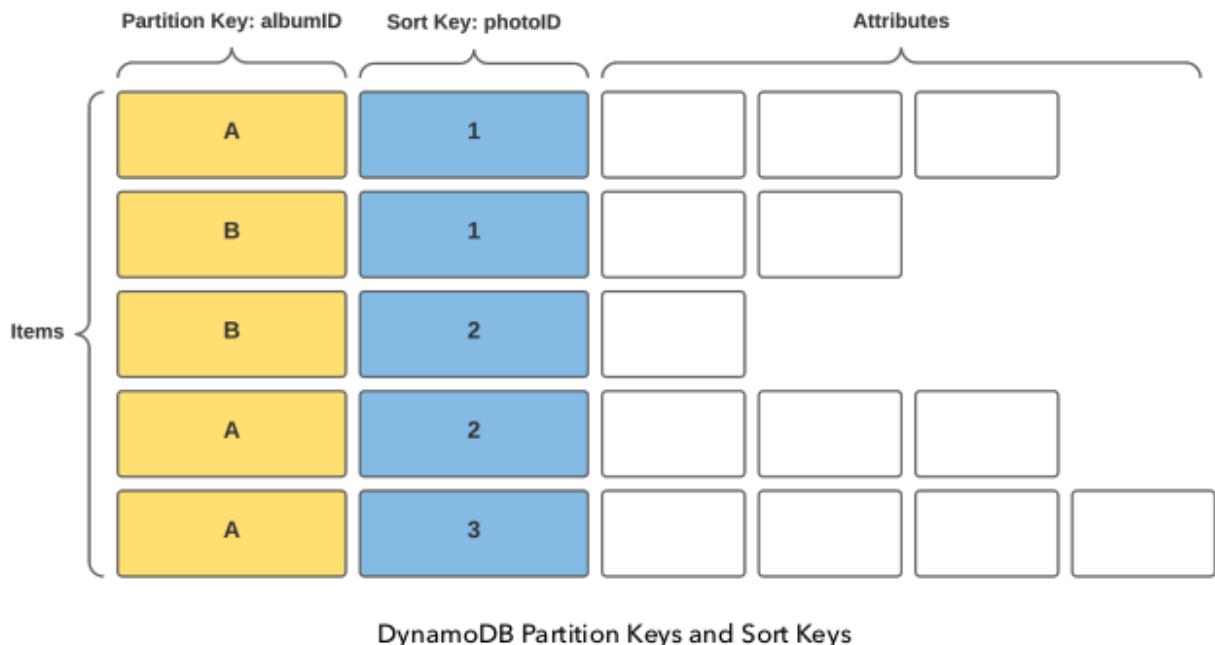
Cancel [Preview changes](#) [Save rules](#)

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.



- 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.

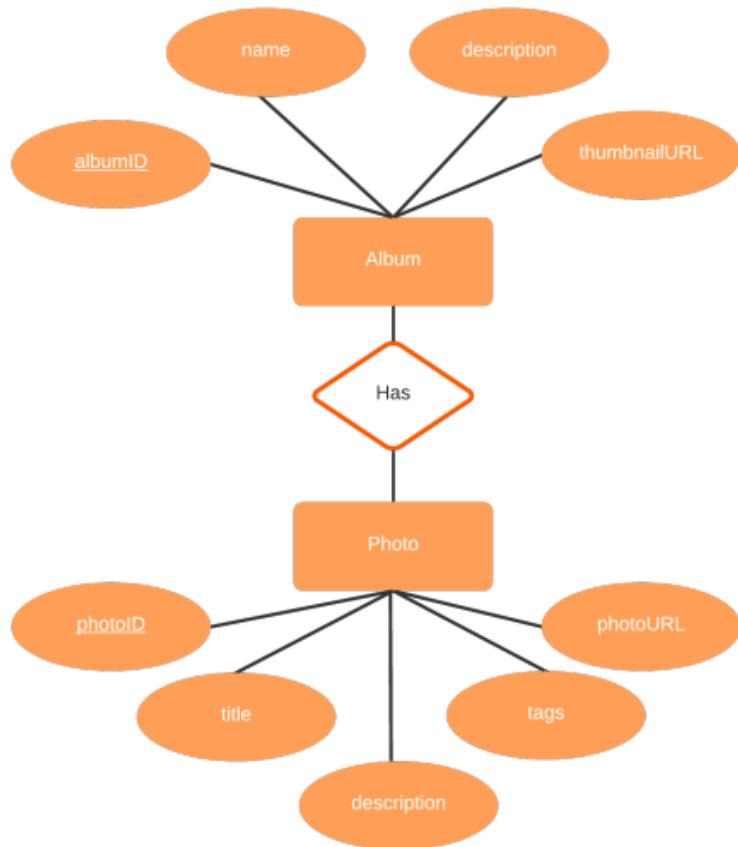
`yourEC2InstanceHostName.com:5000`

The screenshot shows a "Photo Gallery" application interface. At the top, there's a navigation bar with "Photo Gallery", "Home | Create Album", and a search bar labeled "Search albums". Below the navigation, a large image of a snowy mountain peak with colorful prayer flags hanging across it is displayed. Underneath the image, the word "test" appears twice, followed by the text "Created on February 02, 2024".

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 find 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$ _
```

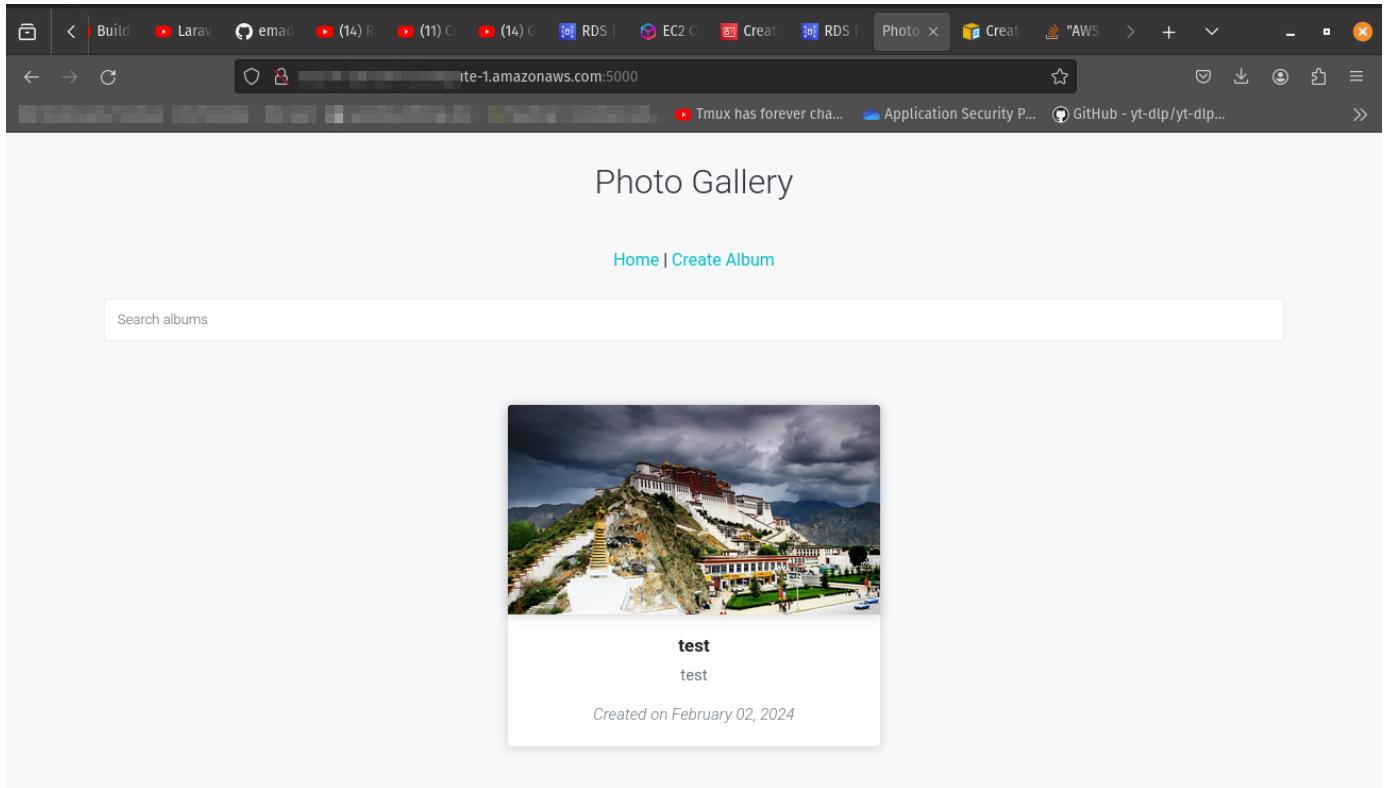
- Run the following command within the SQL root directory to run the Flask web framework using the SQL variant:

```
python3 app.py
```

```
ubuntu@ip-172-31-92-61:~/data/SQL$ python3 app.py
 * Serving Flask app 'app'
 * Debug mode: on
WARNING: This is a development server. Do not use it in a production deployment. Use a production WSGI server instead.
 * Running on all addresses (0.0.0.0)
 * Running on http://127.0.0.1:5000
 * Running on http://172.31.92.61:5000
Press CTRL+C to quit
 * Restarting with stat
 * Debugger is active!
 * Debugger PIN: 356-693-310
```

- 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 to 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 identifier (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 confirmation 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 confirm 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, attach 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 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 identifies your modification.

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).

Appendix

A. How to generate confirmation token

The email confirmation should contain a unique URL that a user needs to click to confirm his/her account. Typically, a confirmation 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 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:

```
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)
# eyJpZCI6NSwibmFtZSI6Iml0c2Rhbmldlc91cyJ9.6YP6T0Ba067XP--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 confirm 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 to the API).

Before you can send emails with SES, you must first 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

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.

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.

Domain name can contain up to 253 alphanumeric characters.

Cancel **Previous** **Next**

aws Services Search [Alt+S] N. Virginia sam.cloud

Verified identity and configuration set resources have been created. A verification email has been sent to schoephe3@gatech.edu. Open the email and click the link to complete the verification process for the email address.

Amazon SES > Get set up

Get set up

Follow the steps in the cards below to verify your email address and sending domain so that you can start sending email through Amazon SES and request production access for your account.

You are in Sandbox - US East (N. Virginia) region [Info](#) Request production access

In a sandbox environment, you can use all of the features offered by Amazon SES; however, certain sending limits and restrictions apply. When you're ready to move out of the sandbox, submit a request for production access. Before you submit a request for production access you must complete the tests below.

Status Sandbox - sending limits and restrictions apply	Daily sending quota 200 emails per 24-hour period	Maximum send rate 1 email per second	Verify an email address or sending domain to request production access. Unverified
---	--	---	---

Get production access and start sending emails [Info](#)

Complete the following steps to verify your email address and sending domain so that you can request production access and start sending emails to your customer base.

Verify email address



schoephe3@gatech.edu

To verify ownership of this email, check your inbox for a verification request email and click the link provided.

[Resend](#)

Verification is pending

Send test email



(Optional but recommended)

Amazon SES mailbox simulator lets you test how your application handles different email sending scenarios.

[Send test email](#)

Waiting for email verification

Verify sending domain



tcert.net

Click on the button below and add the generated CNAME records to your domain's DNS provider.

[Get DNS Records](#)

Verification is pending

aws Products Solutions Pricing Documentation Learn Partner Network AWS Marketplace Customer Enablement Events Explore More Contact Us Support English My Account Sign In to the Console

Congratulations!

You have successfully verified an email address. You can now start sending email from this address.

For new Amazon SES users — If you have not yet applied for a sending limit increase, then you are still in the [sandbox environment](#), and you can only send email to addresses that have been verified. To verify a new email address or domain, see the [Identity Management](#) section of the [Amazon SES console](#).

For new Amazon Pinpoint users — If you have not yet applied for a sending limit increase, then you are still in the [sandbox environment](#), and you can only send email to addresses that have been verified. To verify a new email address or domain, see the [Settings > Channels](#) page on the [Amazon Pinpoint console](#).

If you have already been approved for a sending limit increase, then you can start sending email to non-verified addresses.

Thank you for using Amazon Web Services!

[Go to the Amazon SES detail page.](#)

Get Started with AWS

Learn how to start using AWS in minutes



What's New with AWS

Learn about the latest products, services, and more



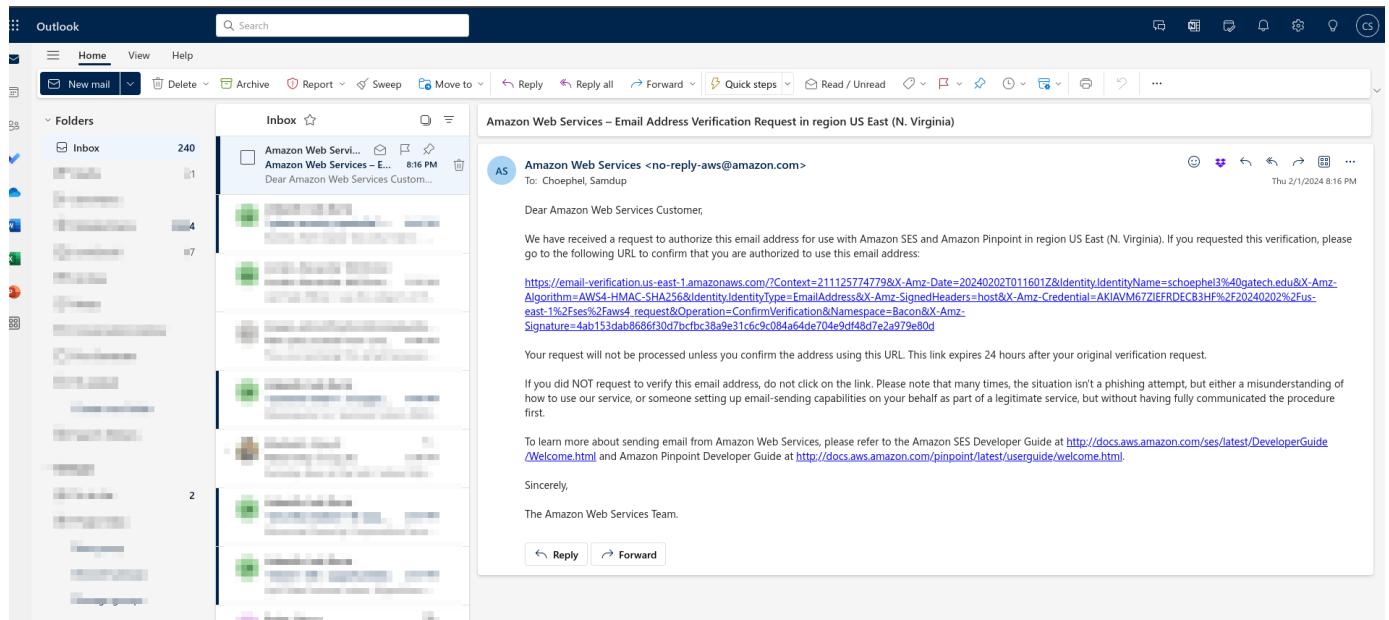
AWSome Day Online Conference | Feb 15

Get started on AWS with a free half-day training. Register now »

AWSOME DAY

Learn About AWS Resources for AWS Developers on AWS Help Sign In to the Console

New accounts are automatically placed in a sandbox mode to help prevent fraud. You can only send emails to addresses you have personally verified 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 find a link to a version of bcrypt below.

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

