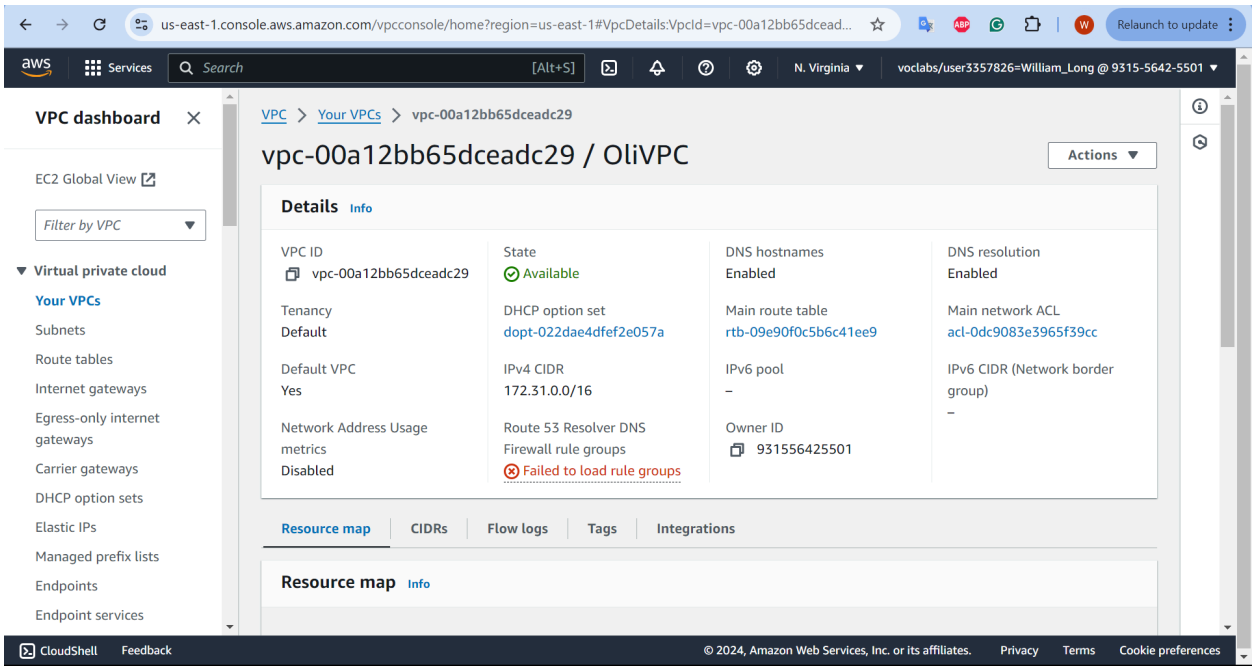
Milestone 3 Documentation

Oliver Long and Jake Jump

Deployment

Create a VPC:

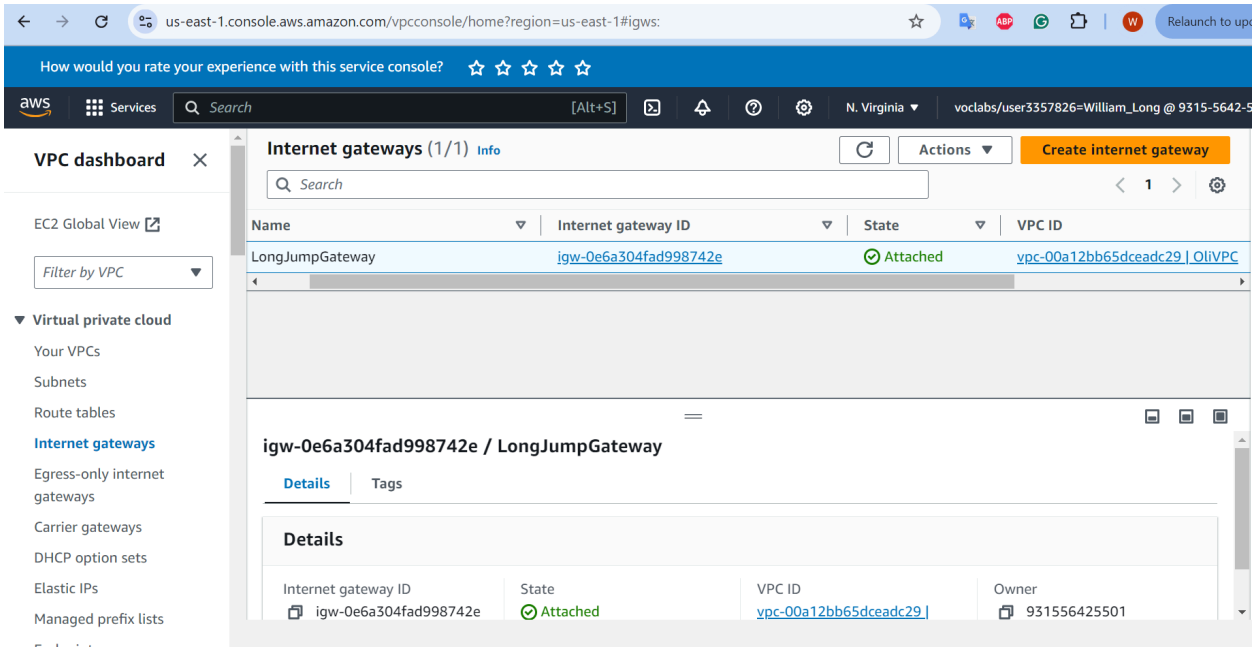
- Enter a name for VPC
- Enter a CIDR block
- Tenancy: default
- Create Subnets:
    - Enter a CIDR block



Created the VPC

Set Up Internet Gateway:

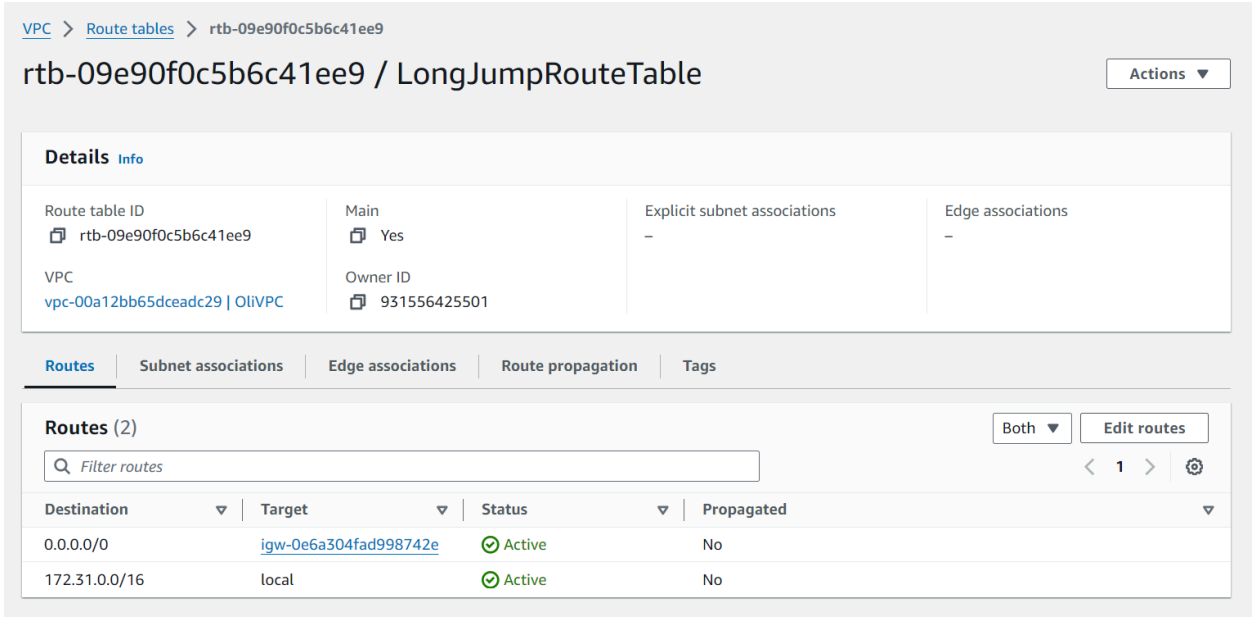
- In the VPC dashboard, select and create an Internet Gateway
- Create a name
- Click Create Internet Gateway.
- Actions -> Attach to VPC



Created and connected the internet gateway to the VPC

Configure Route Tables:

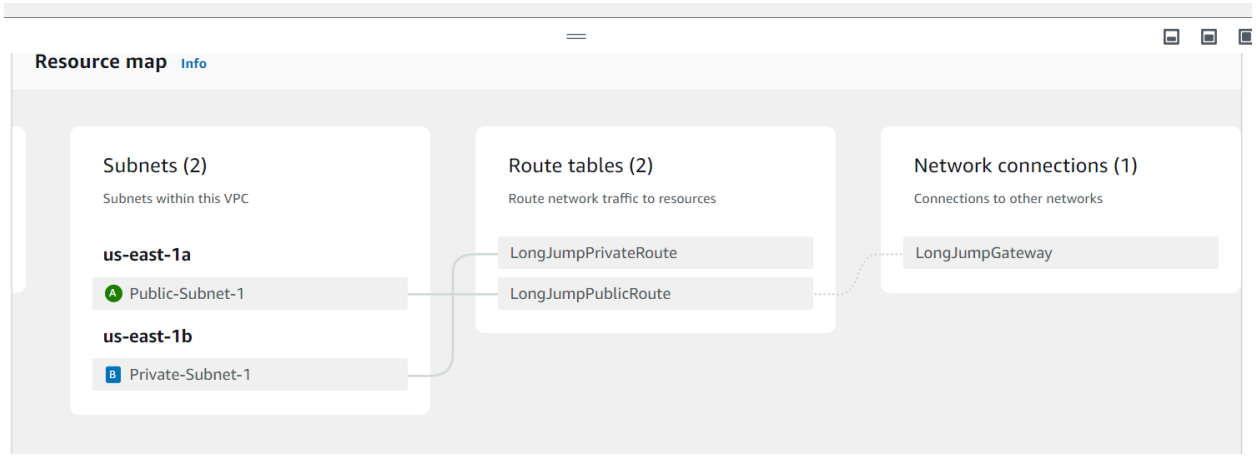
- Create route table
- Enter a name
- Select VPC
- After creating, select the route table and click Actions then Edit routes
- Add route with destination for internet 0.0.0.0/0 traffic
- Select the Internet Gateway from the dropdown as a target
- Associate this route table with the public subnet



Created the routing table, pointing to the Internet gateway

Create Subnets:

- Select Subnets
- Create Subnet
- VPC ID: VPC created
- Subnet name: cool name
- Availability Zone: select an availability zone
- IPv4 CIDR block: enter a CIDR block
- Do this again for the private subnet



Resource map of subnets and routing tables to the internet gateway inside the VPC

Create Security Groups:

Web Server Security Group:

- Enter Name
- Select your VPC
- Inbound rules:
    - HTTPS
    - Source: 0.0.0.0/0

Application Server Security Group:

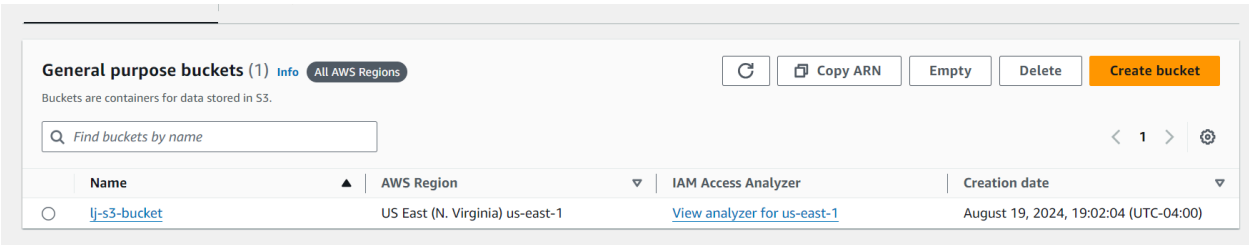
- Repeat above steps
- Inbound rules:
    - Allow traffic only from the Web Server's private IP range

Name	Security group ID	Security group name	VPC ID
–	sg-0d0ad197362c07a3d	default	vpc-00a12bb65dceadc29
–	sg-0de5e9d41c4a5284e	LongJumpWebServer	vpc-00a12bb65dceadc29
–	sg-0c513ac6089ea3689	LongJumpApplicationServer	vpc-00a12bb65dceadc29

Security Groups

Create an S3 Bucket:

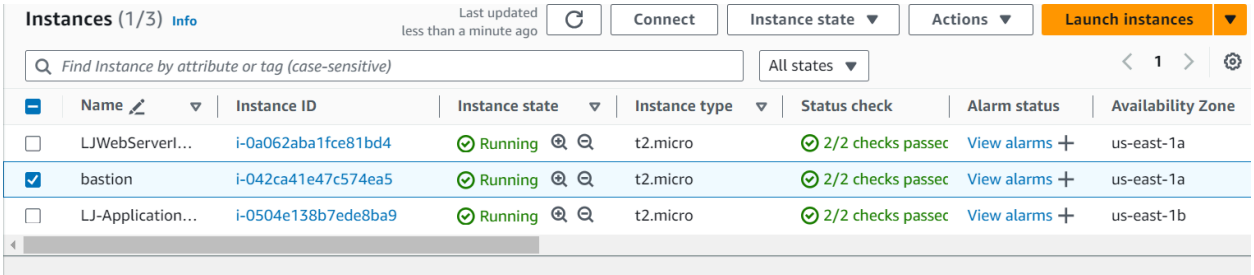
- Select s3 in the console
- Create bucket
- Make a bucket name
- Region: Same region as the VPC
- Object Ownership: Enable bucket owner enforced
- Bucket settings: Enable default encryption using AWS KMS



S3 Bucket

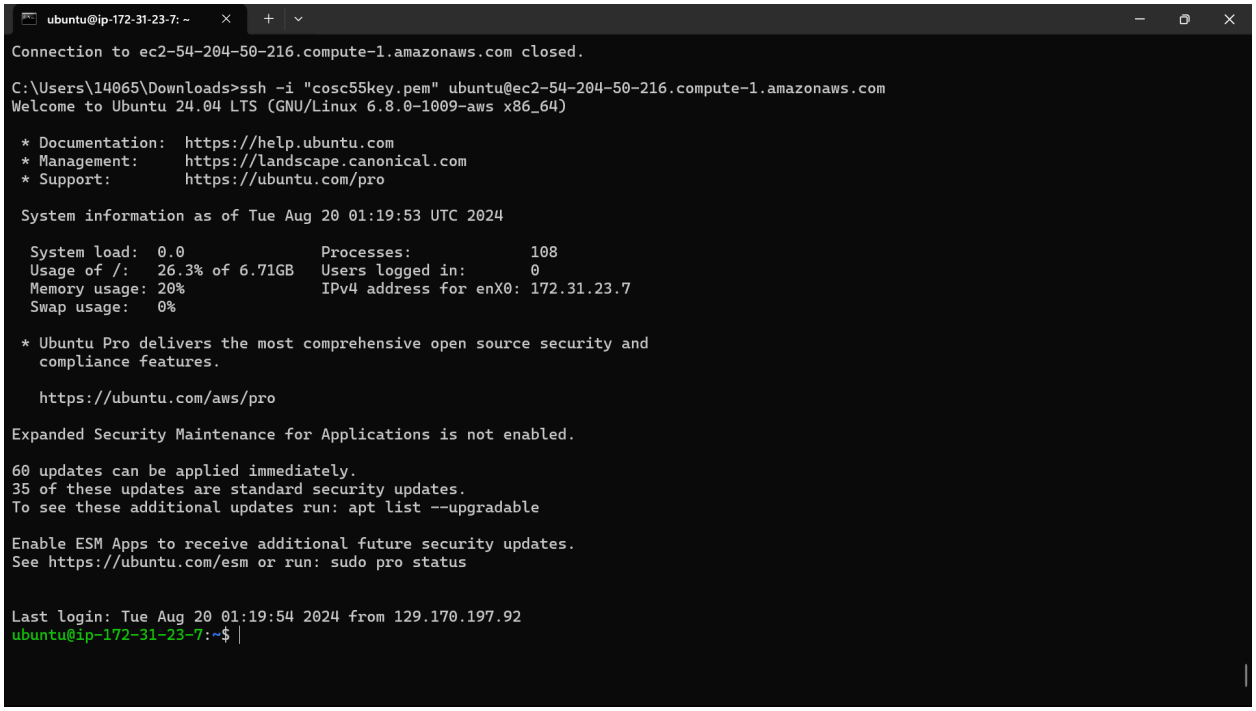
Instance Set Up:

- Create 3 EC2 instances:
  - One in the public subnet for front-end development
  - One in the private subnet for backend
  - One “Bastion” to access the EC2 in the private subnet



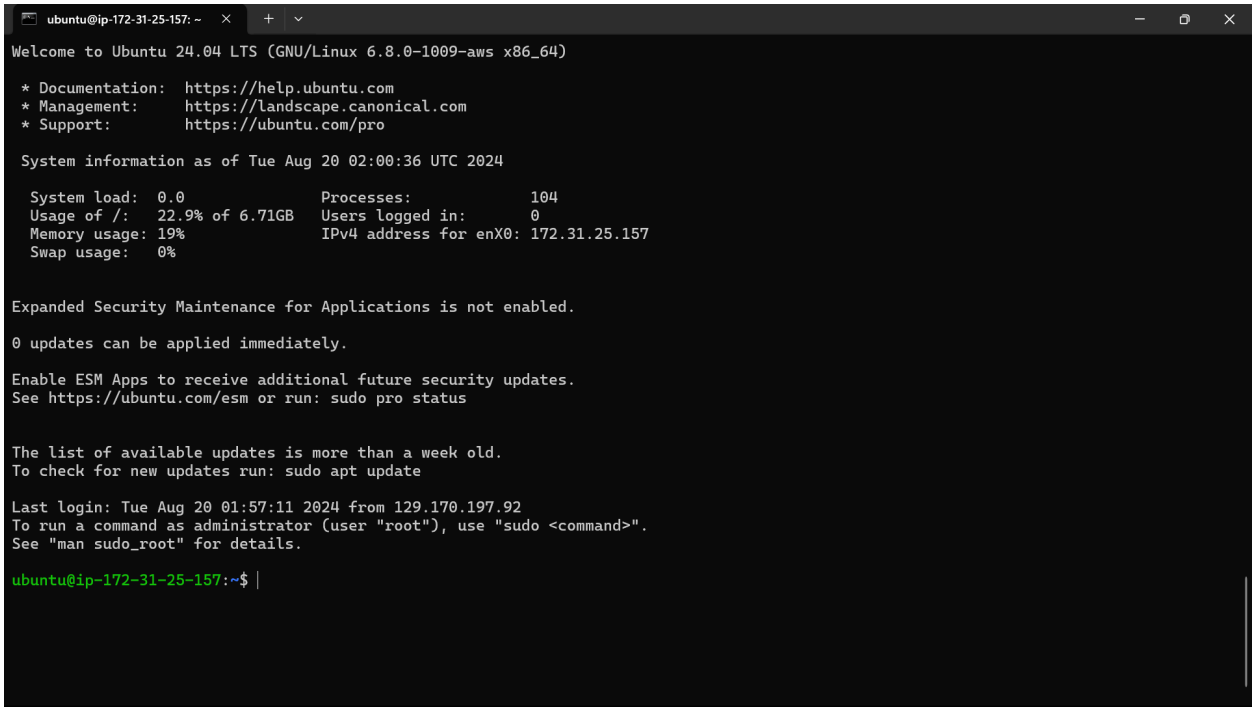
Instances set up

Testing



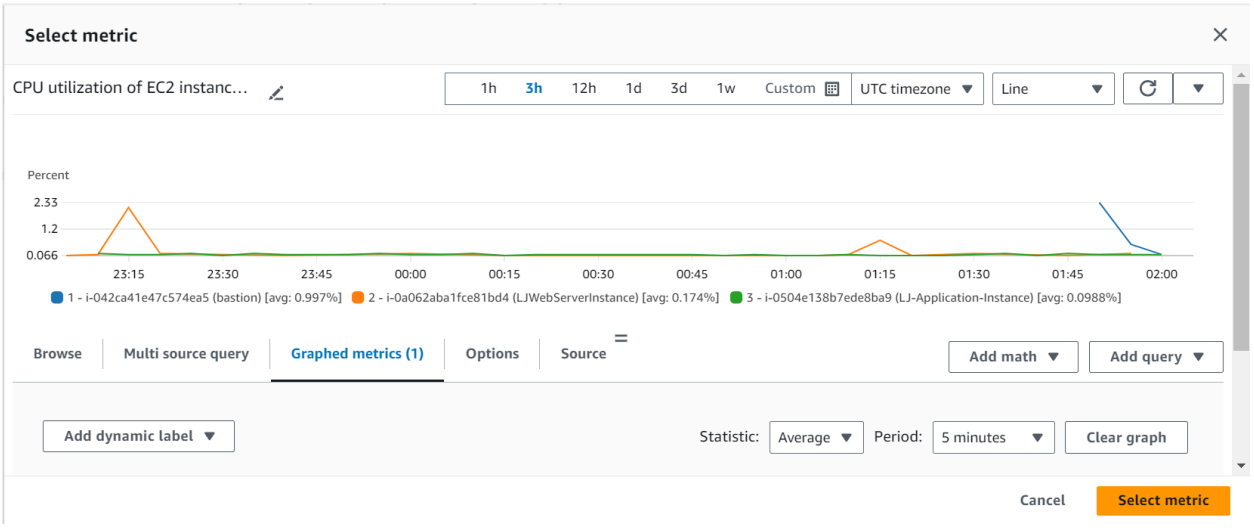
Successful SSH into the Web Server instance

- Use SSH key found in AWS gui



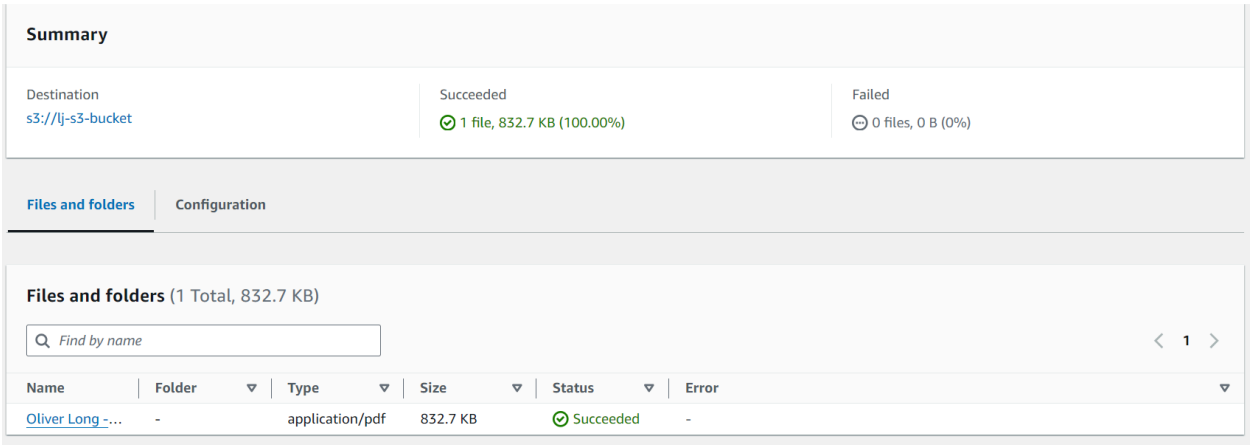
Successful SSH into the bastion instance to connect to the private instance for the application server

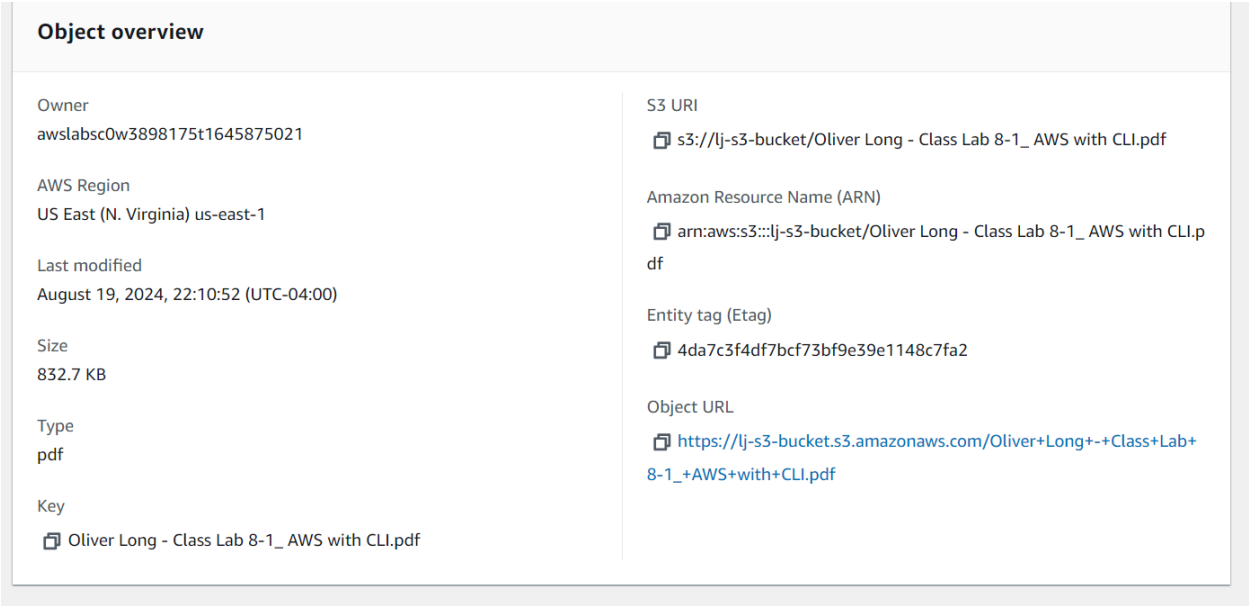
- Used public ip on bastion and private ip on the application instance

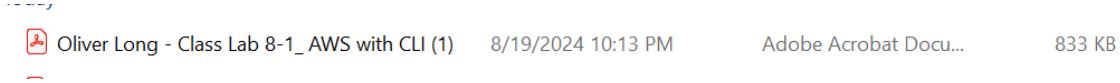


Monitors CPU Utilization of EC2 instances in CloudWatch

- Navigate to CloudWatch and use “Add query” to select what metrics you would like visually represented







Upload and Download to s3 bucket with a testing file

- Went into the s3 bucket interface and uploaded and downloaded a testing file to make sure the bucket works.