**Streamlit ApplicationOverview:**

****

This application is a Streamlit-based web interface designed to facilitate the inspection of porosity defects by querying and visualizing defect-related data. The application interacts with AWS services including Amazon S3 and DynamoDB to retrieve and display images, defect masks, and reports.

**FeaturesSearch by Block Number:**

Input a block number to query the DynamoDB table and fetch relevant information.

Defect Visualization: Display localized defect images with an optional mask overlay.

Defect Report: View CSV-based defect reports if available.

Pass/Fail Status: Displays inspection status based on the presence of defect data.

**PrerequisitesEnsure you have the following before running the application:**

Python 3.8+

AWS credentials configured (with access to S3 and DynamoDB)

**Required Python packages:**

pip install streamlit pandas boto3 pillow streamlit-image-zoomAWS ConfigurationThe application uses the following AWS resources:

Amazon S3: Stores localized images, mask overlays, and defect reports in CSV format.

DynamoDB: Contains metadata and S3 file locations.



Environment VariablesEnsure your AWS credentials are properly configured through aws configure or environment variables.

Directory Structure.

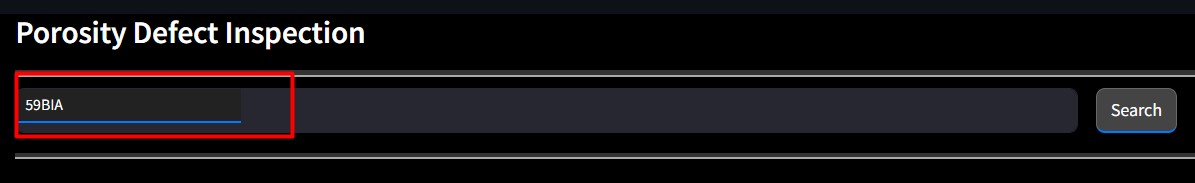
├── app.py

└── README.mdDynamoDB Table SchemaThe application queries the DynamoDB table data\_logs with the following key structure:

AttributeTypeDescriptionblock\_idStringUnique identifier for blocksimage\_url\_locationStringS3 URL for localized imagemask\_url\_locationStringS3 URL for mask imagelocalized\_url\_locationStringS3 URL for localized imagereport\_url\_locationStringS3 URL for CSV reportUsageRun the Streamlit Application

streamlit run app.pySearch for a Block Number

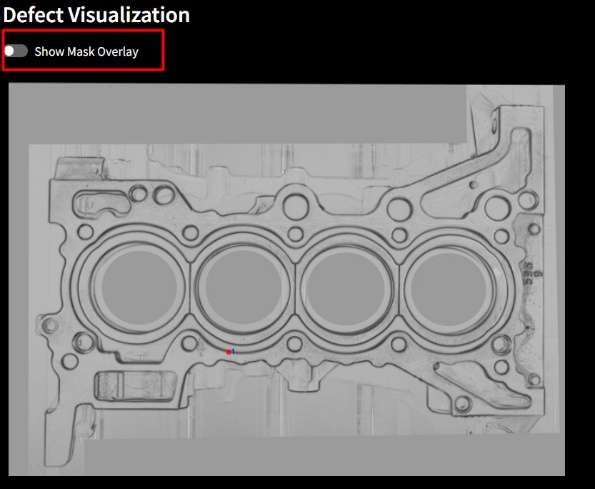
Enter a block number in the search bar.



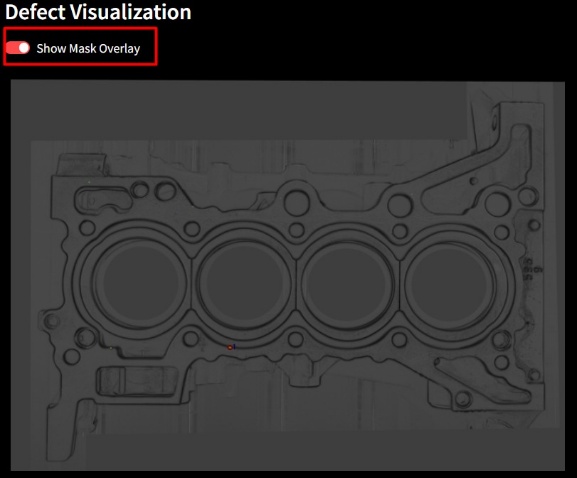
Click the "Search" button to fetch and display related data.

**Inspect Defects**

View the localized image of the defect.



Toggle the "Show Mask Overlay" to visualize the defect mask.



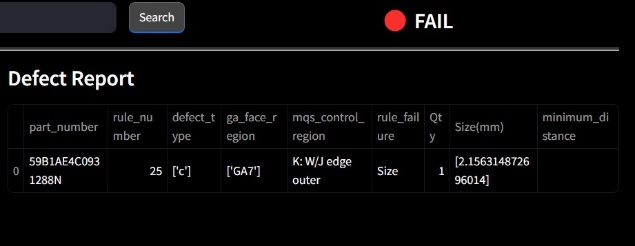
**Check Inspection Status**

"🔴 FAIL" is displayed if a defect report is found.

"🟢 PASS" if no defect report is available.

**View CSV Report**

Displays tabular data from the defect CSV if present.



Code StructureAWS Clients Initializations3 = boto3.client("s3")

*dynamodb = boto3.resource("dynamodb")*

*table = dynamodb.Table(DYNAMODB\_TABLE)Query DynamoDBdef query\_dynamodb(block\_id):*

*response = table.scan(FilterExpression=boto3.dynamodb.conditions.Attr('block\_id').eq(block\_id))*

*return response.get("Items", [])Fetch S3 Filesdef fetch\_s3\_file(s3\_url):*

*bucket, key = s3\_url.replace("s3://", "").split("/", 1)*

*response = s3.get\_object(Bucket=bucket, Key=key)*

*return response['Body'].read()Image Visualizationimage\_zoom(blended\_image, mode="scroll", size=(700, 500), keep\_aspect\_ratio=True, zoom\_factor=8.0, increment=0.8)*

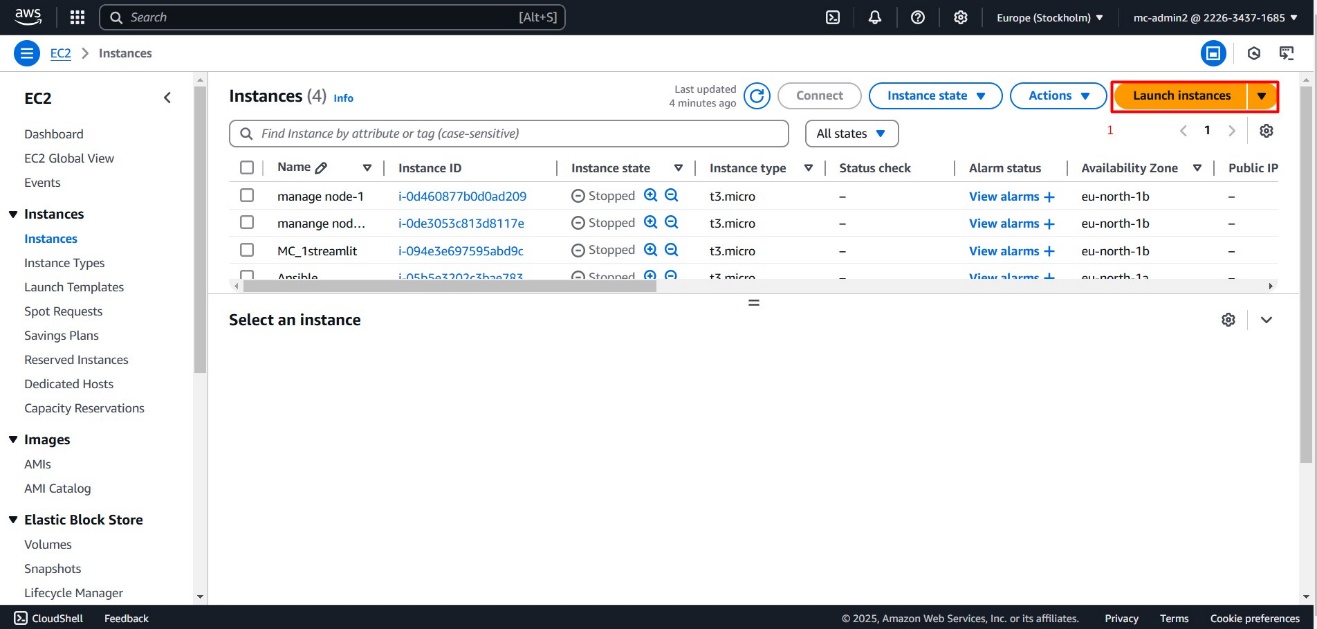
Custom StylingThe application includes custom CSS for a dark theme and improved UI elements.

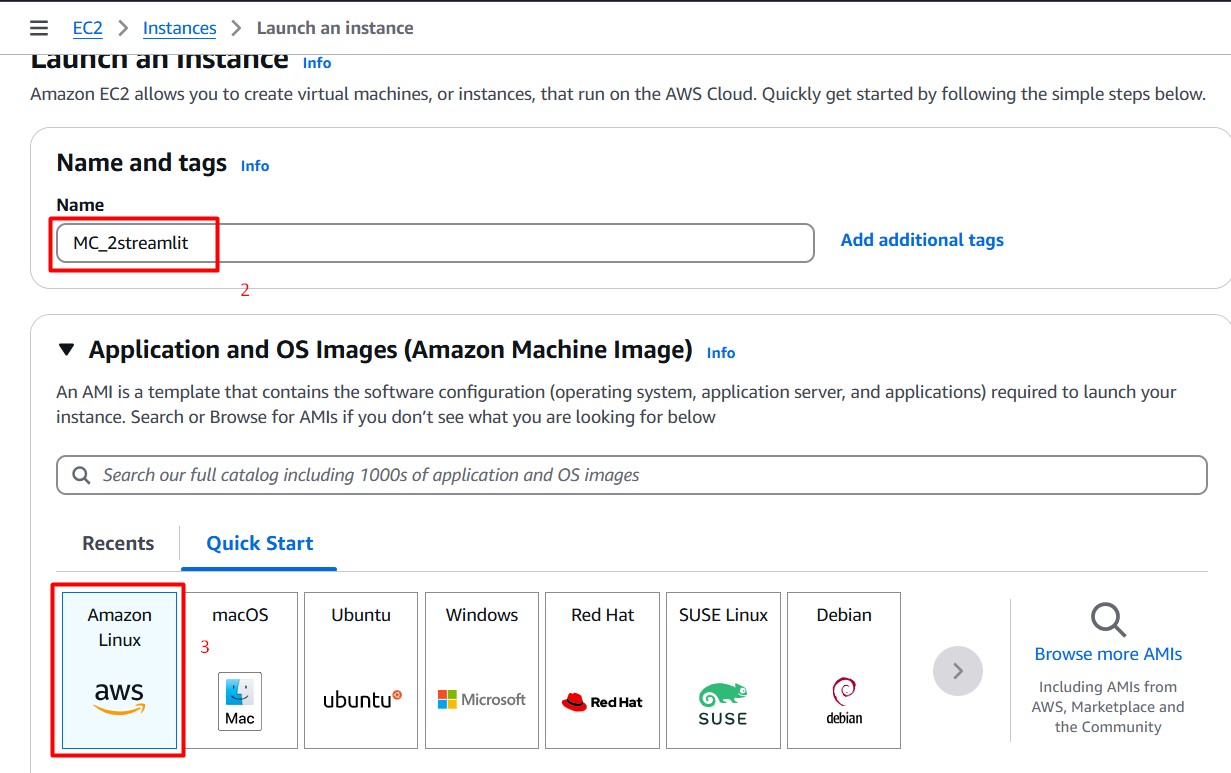
Error HandlingDisplays warnings if no data is found.

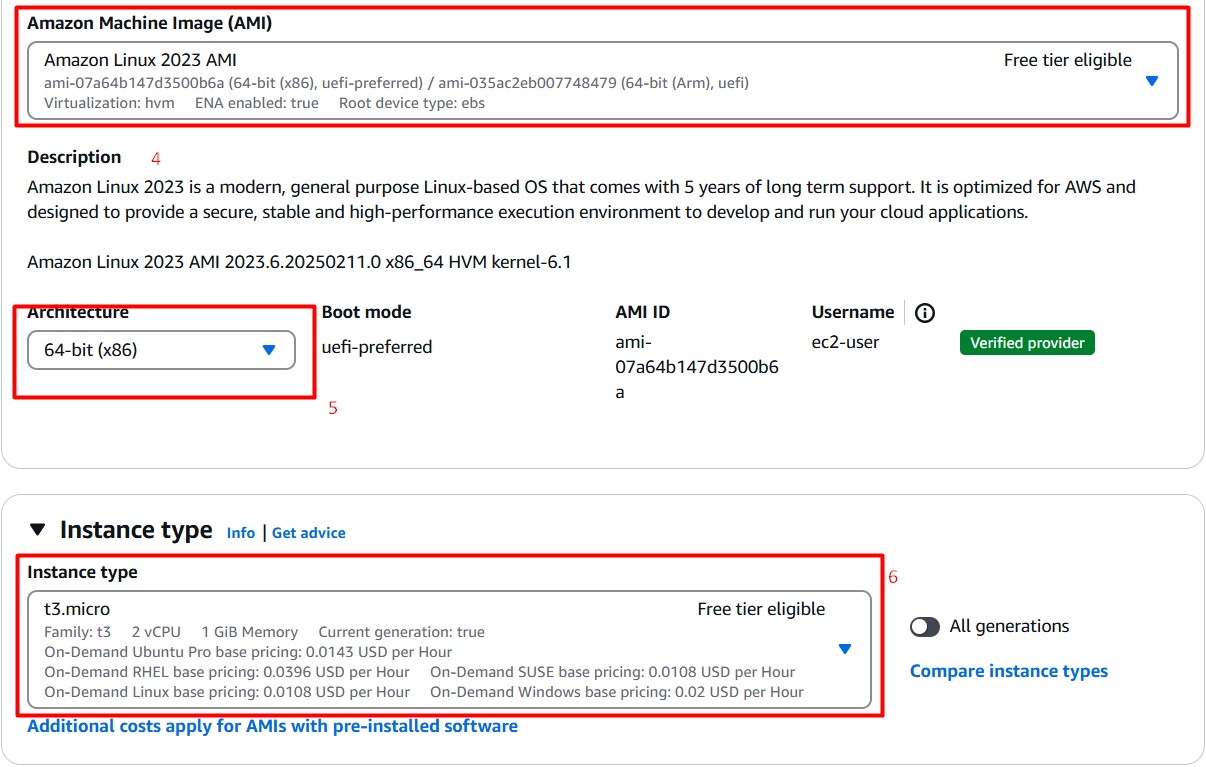
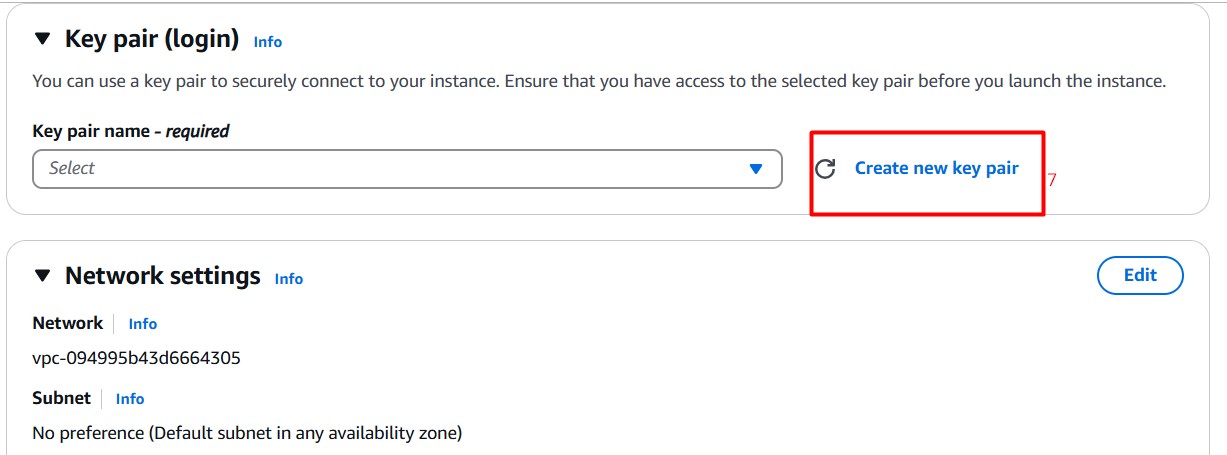
Error messages are shown for failed image loads.

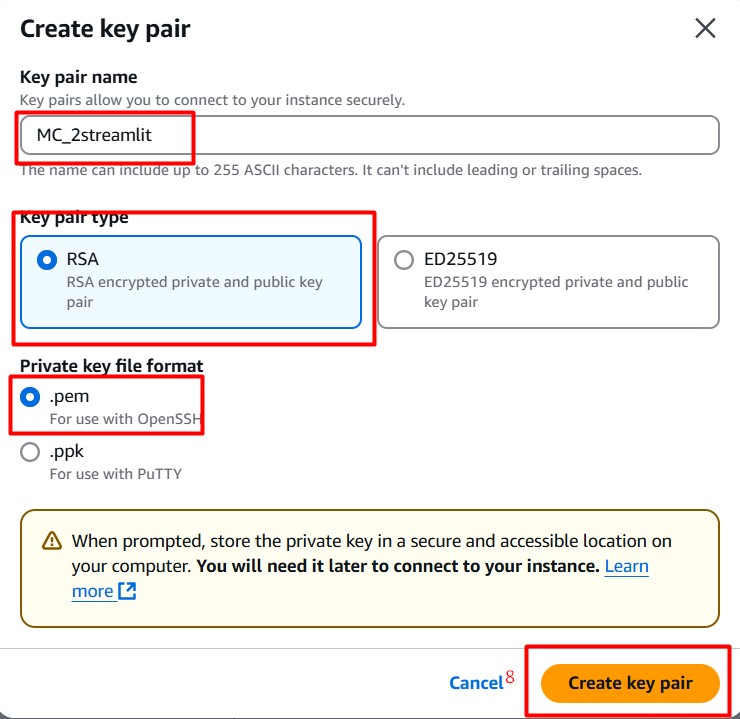
CustomizationModify AWS resource names by changing BUCKET\_NAME and DYNAMODB\_TABLE

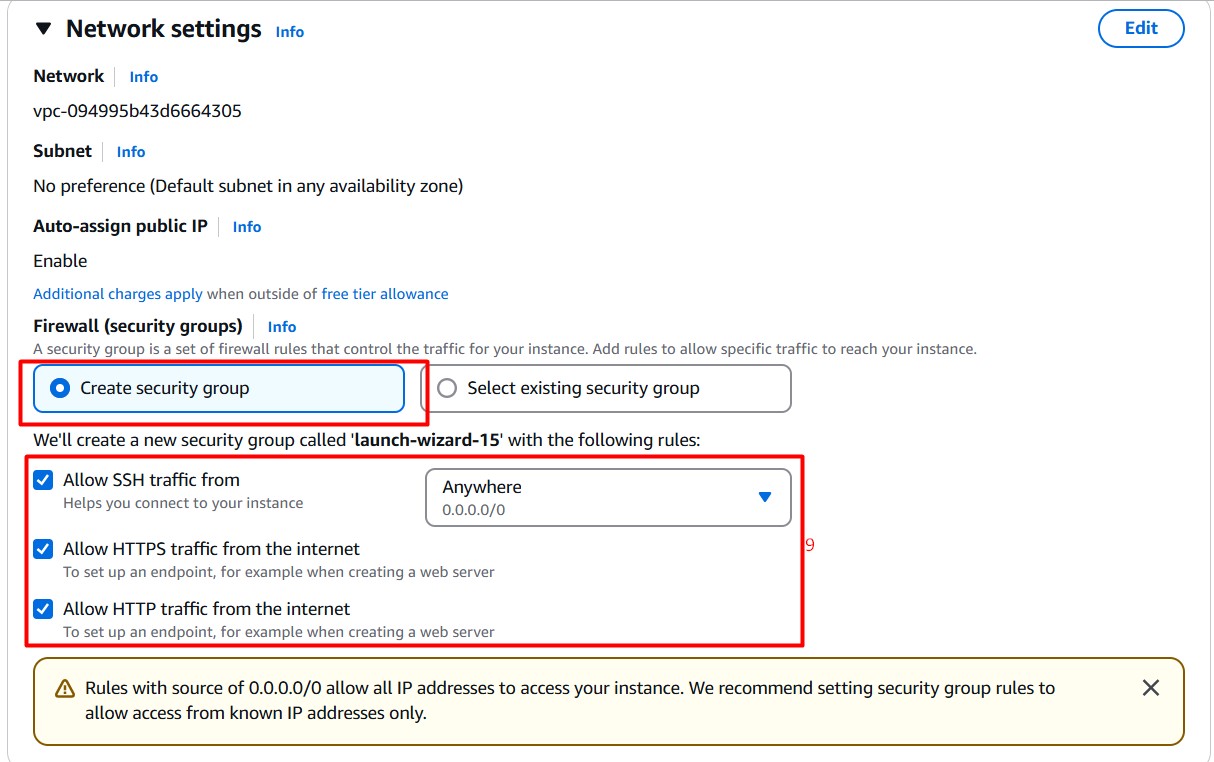
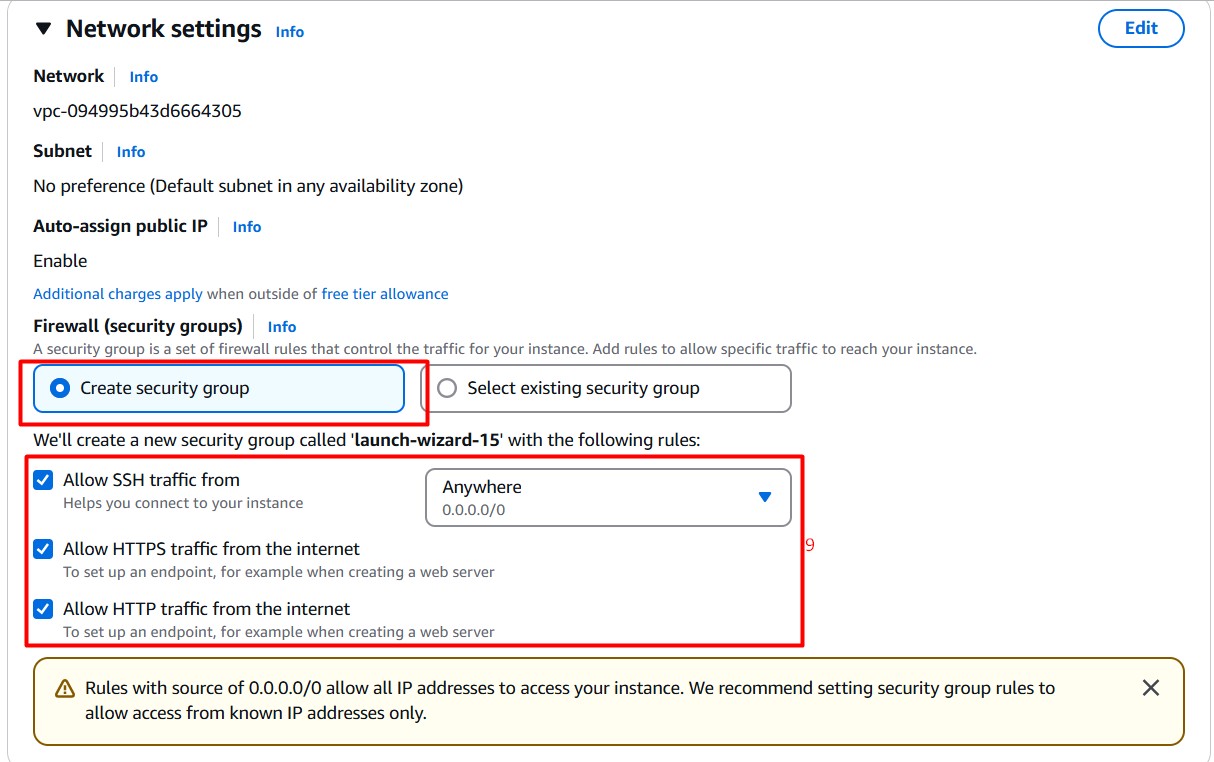
**Deploying an streamlit application on AWS EC2**

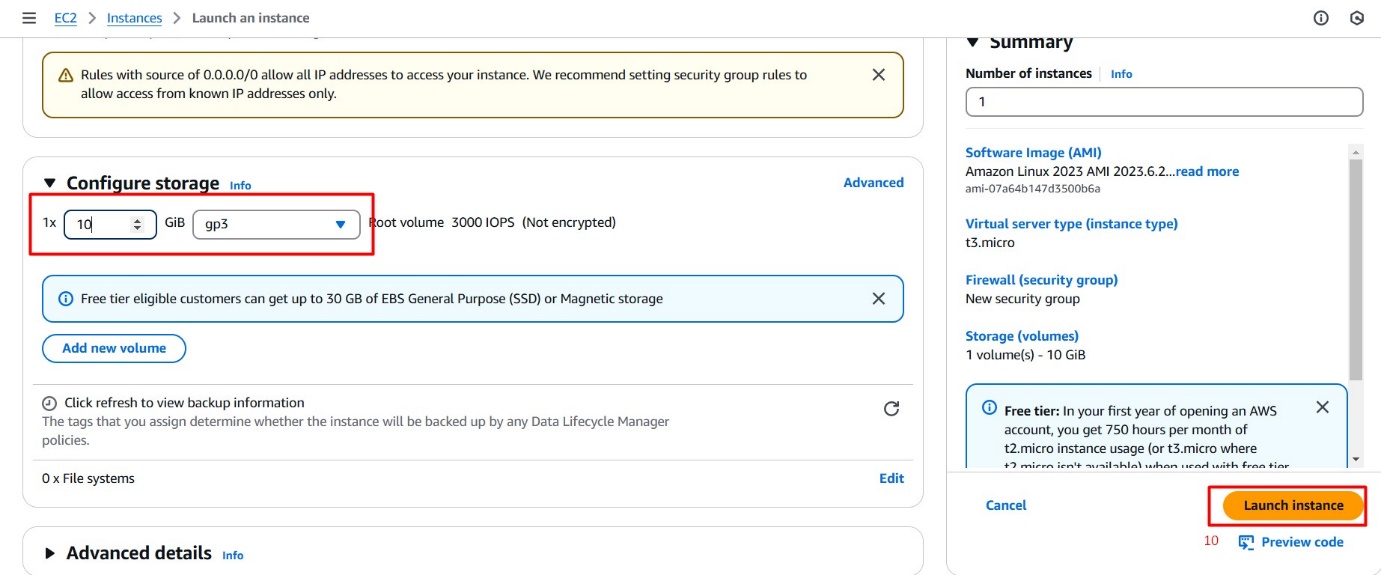
****launching an instances

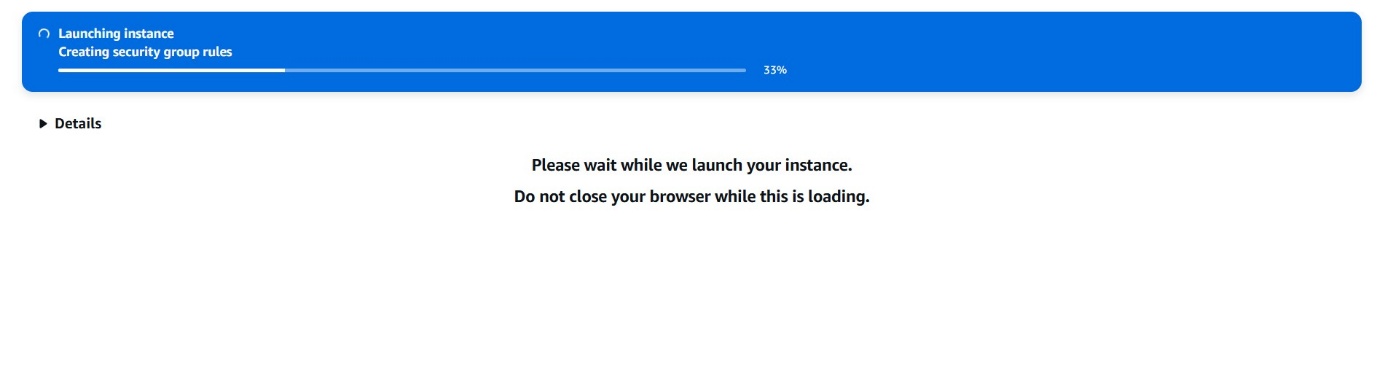
****

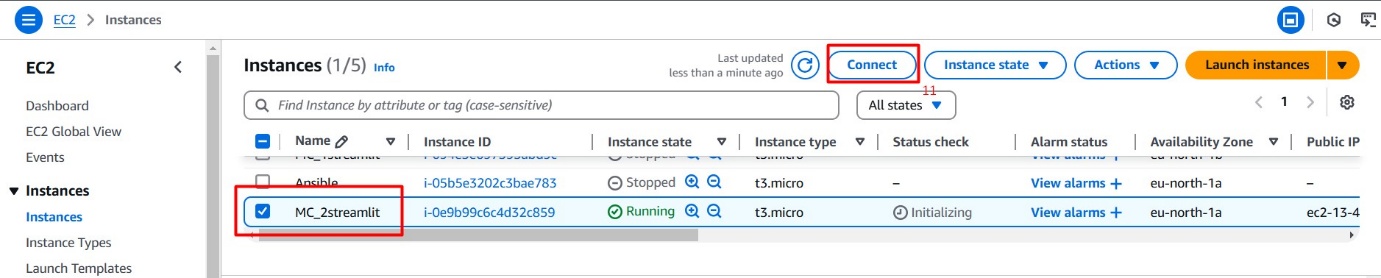
****

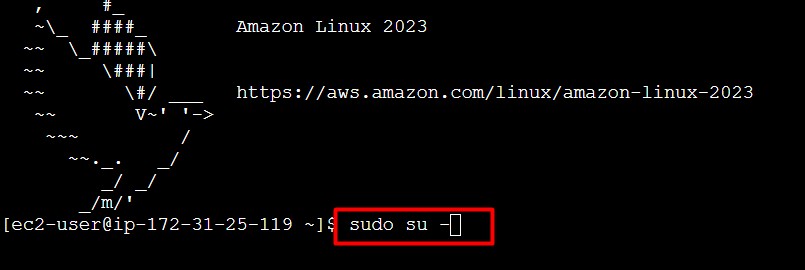
****

****

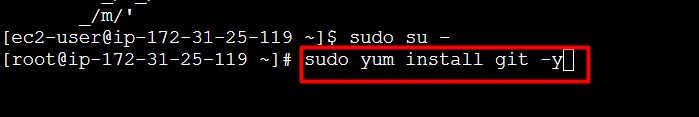
****

****

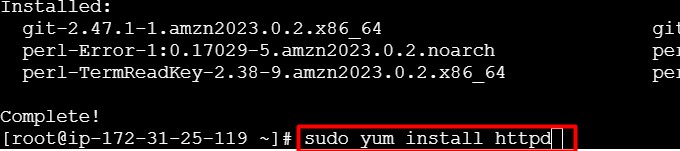
****

****

Installing git

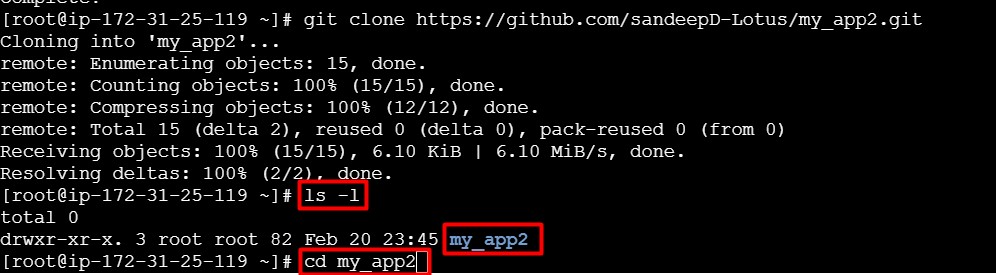
****

Installing httpd

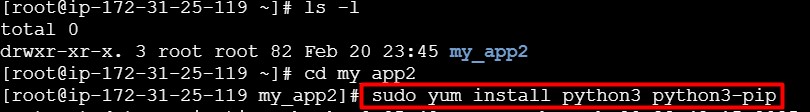
****

Clone the git repo where the application was

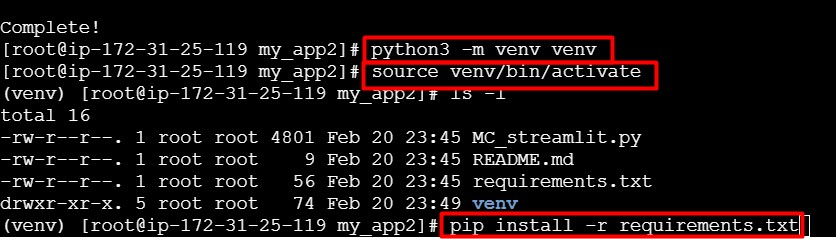
****

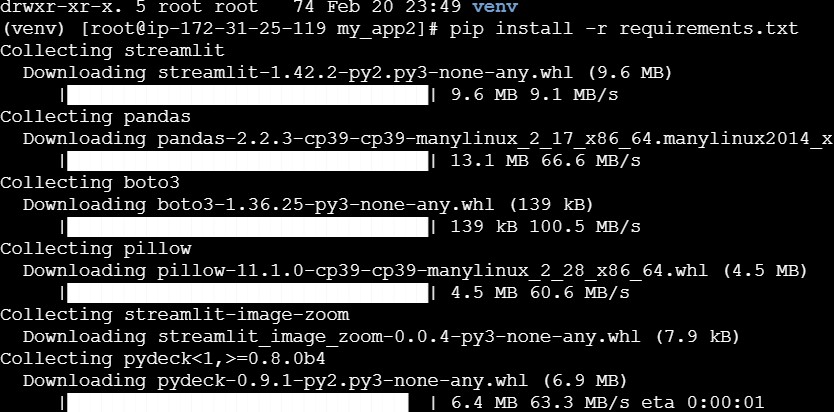
****

Installing the python3

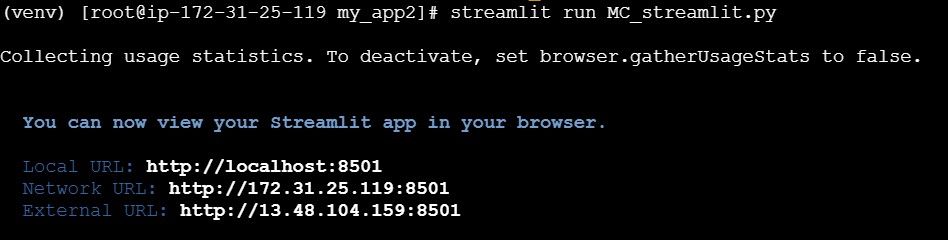
****

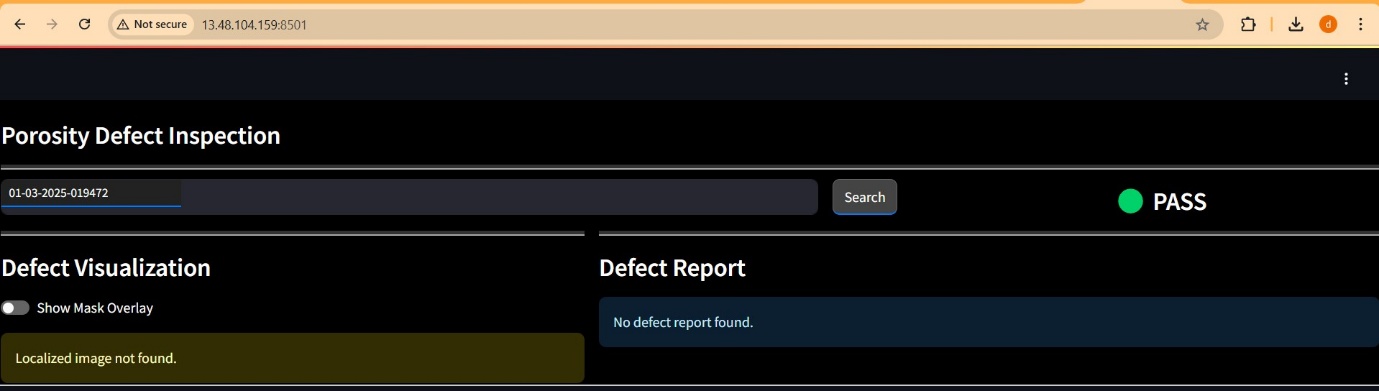
Creating the virtual environment

****

****

****

****

****