

LAMBTON COLLEGE



A Report on [Lab 7 & 8 on AWS Academy Data Analytics]

121 Brunel Rd, Mississauga ON L4Z 3E9

A Group assignment with screenshots of Lab 7 and 8

On Aws academy

Big Data Analytics DSMM

Under the supervision

Of

Professor Teresa Zhu

Submitted by:

Onyinye Mbanefo (C0831578)
Aadarsha Chapagain (C0825975)
Roshan Acharya (C0831342)
Anjana Kuriakose (C0829580)

Submitted to:

Lambton College
Prof. Teresa Zhu

Submission Date:

December 1, 2022

Lab 7 Analyze Streaming Data with Amazon Kinesis Data Firehose, Amazon Elasticsearch Service, and Kibana

Objectives:

- Access Amazon Kinesis Data Firehose and Amazon Elasticsearch Service (Amazon ES) in the AWS Management Console
- Create a Kinesis Data Firehose delivery stream
- Integrate a Kinesis Data Firehose delivery stream with Amazon ES
- Build visualizations with Kibana

Task 1: Review the infrastructure

Task 1.1: Review the Amazon EC2 instance

The screenshot shows the AWS EC2 Instances page. On the left, there's a sidebar with navigation links like EC2 Dashboard, EC2 Global View, Events, Tags, Limits, Instances (with sub-links for Instance Types, Launch Templates, Spot Requests, Savings Plans, Reserved Instances), Images (AMIs, AMI Catalog), and Elastic Block Store. The main content area has a heading 'Instances (1/1) Info'. It shows a table with one row for 'AES Kibana Demo'. The table columns include Name (AES Kibana Demo), Instance ID (i-0efa23a55decceda8), Instance state (Running), Instance type (t2.micro), Status check (2/2 checks passed), Alarm status (User: arn:aws:...), Availability Zone (us-east-1a), and Public IP (ec2-107-20-99-169). Below the table, there's a detailed view for the selected instance, showing its summary, security, networking, storage, status checks, monitoring, and tags. The public IP address 107.20.99.169 is highlighted in the summary section.

Public IPv4 address copied to be used in later task

AesDemoWebserverIAMRole

Summary

Creation date: December 01, 2022, 04:17 (UTC-05:00)

ARN: arn:aws:iam::912602678905:role/AesDemoWebserverIAMRole

Last activity: 15 minutes ago

Maximum session duration: 1 hour

Permissions | Trust relationships | Tags (1) | Access Advisor | Revoke sessions

Permissions policies (3)

You can attach up to 10 managed policies.

Policy name	Type	Description
AesDemoWebserverAMPolicy1	Customer inline	
AesDemoWebserverAMPolicy2	Customer inline	
AesDemoWebserverAMPolicy3	Customer inline	

AesDemoWebserverIAMRole showing three separate AWS Identity and Access Management (IAM) policies were created to control access to the resources that are used in the lab.

Task 1.2: Review the Kinesis Data Firehose delivery stream

Amazon Kinesis

Transform source records with AWS Lambda

Data transformation: Enabled

Lambda function: aes-demo-lambda-function

Runtime: python3.9

Description: -

Lambda function version: \$LATEST

Timeout: 5 minutes

Buffer size: 1 MiB

Destination settings

Specify the destination settings for your delivery stream.

OpenSearch Service destination

Domain: aes-kibana-demo

Index: apache_logs

Index rotation: No rotation

Retry duration: 120 seconds

Buffer hints

Buffer size: 5 MiB

aes-kibana-demo-firehose-stream configurations

Task 1.3: Review the Amazon ES cluster

The screenshot shows the AWS OpenSearch Service console with the 'Domains' section selected. A message at the top indicates a successful update to service software version R20220928-P1. The 'aes-kibana-demo' domain is selected. The 'General information' card displays the domain name (aes-kibana-demo), status (Active), ARN (arn:aws:sesus-east-1:912602678905:domain/aes-kibana-demo), version (Elasticsearch 6.3 Upgrade available), and service software version (R20220928-P1). The 'Kibana URL' field is highlighted with a red box and contains the URL: https://search-aes-kibana-demo-iw7k3cp15rkekvsnfas2gdfgi.us-east-1.es.amazonaws.com/_plugin/kibana.

The aes-kibana-demo Amazon ES cluster Kibana URL

Task 2: Configure Kibana

The screenshot shows the Kibana Dev Tools console. The left sidebar has 'Dev Tools' selected. In the main area, a 'Console' tab is active. A 'DELETE /apache_logs' command is entered in the text input. The response pane shows the JSON object: { "acknowledged": true }.

Deleting Apache Logs

The screenshot shows the Kibana Dev Tools Console. On the left, there's a sidebar with links: Discover, Visualize, Dashboard, Timeline, Alerting, Dev Tools (which is selected), and Management. The main area has tabs for Dev Tools and Console. In the Dev Tools tab, there's a code editor with a syntax highlighter. The code is a JSON object for a PUT request:

```

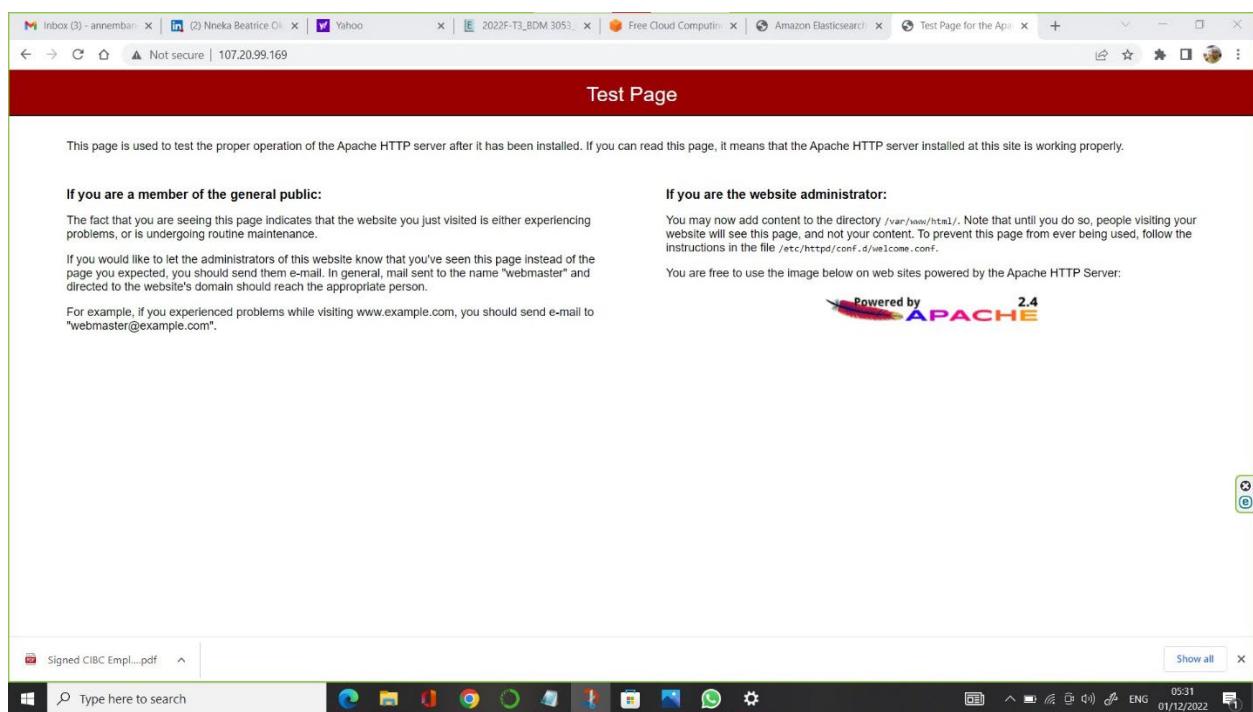
1 PUT apache_logs
2 {
3     "settings": {
4         "index": {
5             "number_of_shards": 10,
6             "number_of_replicas": 0
7         }
8     },
9
10    "mappings": {
11        "access_logs": {
12            "properties": {
13                "agent": { "type": "text" },
14                "browser": { "type": "keyword" },
15                "bytes": { "type": "text" },
16                "city": { "type": "keyword" },
17                "country": { "type": "keyword" },
18                "datetime": { "type": "date", "format": "dd/MMM
/yyy:HH:mm:ss Z" },
19                "host": { "type": "text" },
20                "location": { "type": "geo_point" },
21                "referrer": { "type": "text" },
22                "os": { "type": "keyword" },
23                "request": { "type": "text" },
24                "response": { "type": "text" },
25                "webpage": { "type": "keyword" },
26                "referring_page": { "type": "keyword" }
27            }
28        }
29    }
30 }
31
32

```

At the bottom of the code editor, there are buttons for Run, Stop, and Refresh. Below the code editor is a search bar with placeholder text "Type here to search". The status bar at the bottom right shows the date and time: 01/12/2022 05:07.

Creating a new Index

Task 3: Populate the web server log with data



Task 4: Create the Kibana index

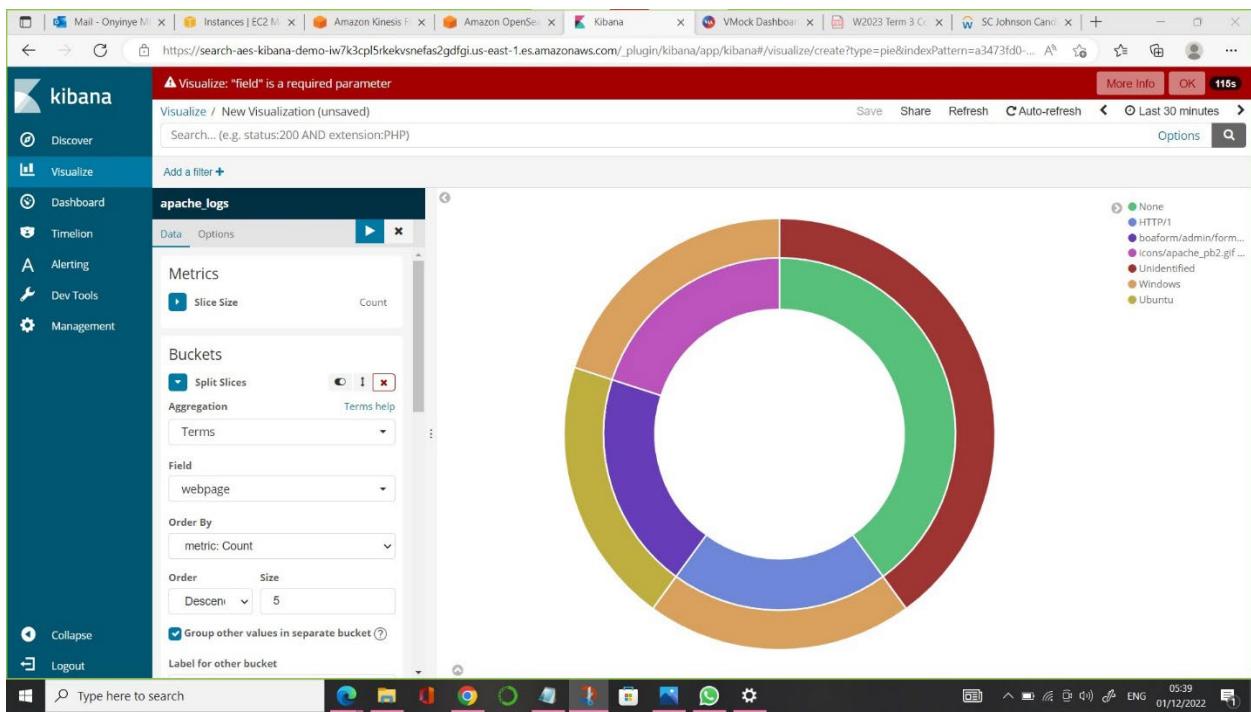
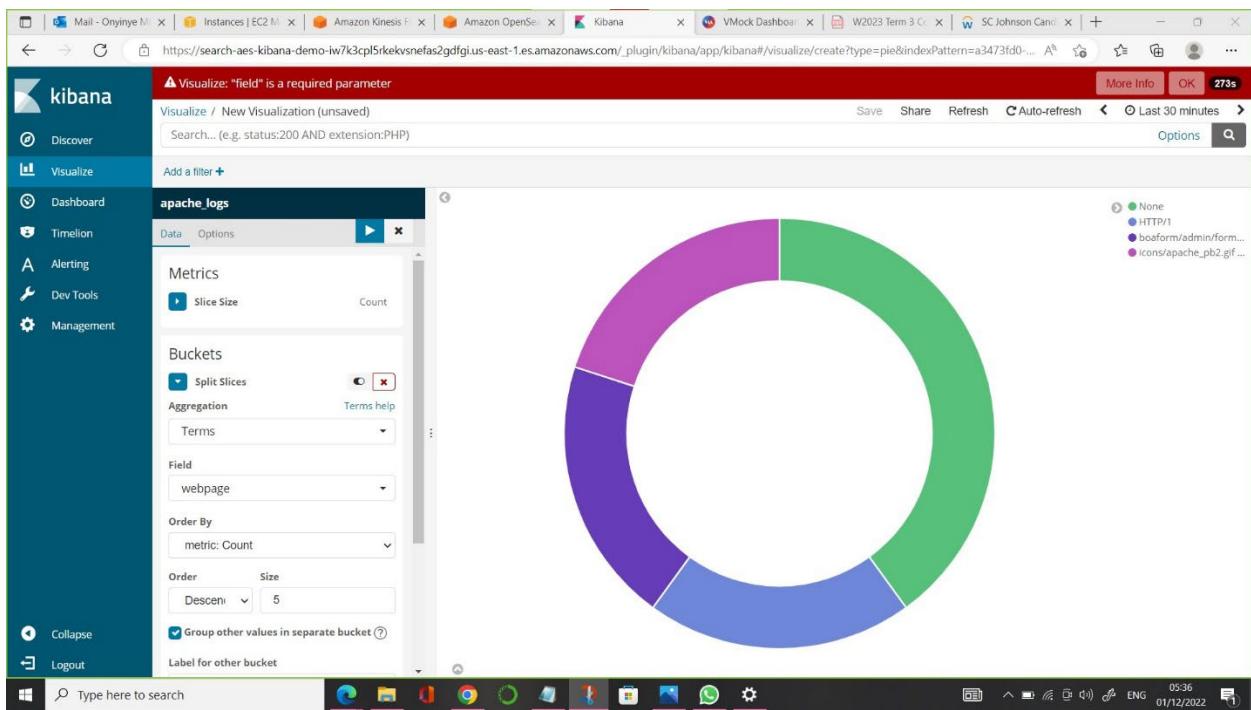
The screenshot shows the Kibana Management interface with the 'Index Patterns' tab selected. A single index pattern named 'apache_logs' is listed. Below it, a table displays various fields with their types and searchability status. The table includes columns for Name, Type, Format, Searchable, Aggregatable, and Excluded.

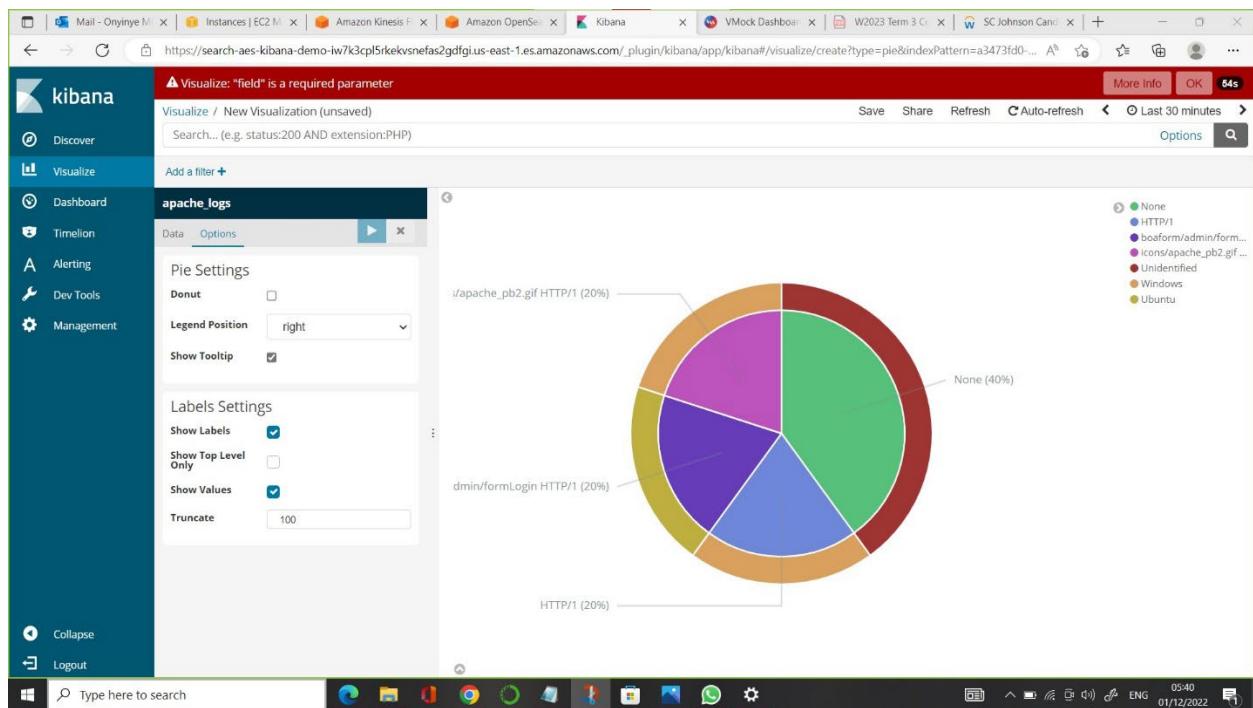
Name	Type	Format	Searchable	Aggregatable	Excluded
_id	string		•	•	
_index	string		•	•	
_score	number				
_source	_source				
_type	string		•	•	
agent	string		•		
browser	string		•	•	
bytes	string		•		
city	string		•	•	
country	string		•	•	

Index patterns created

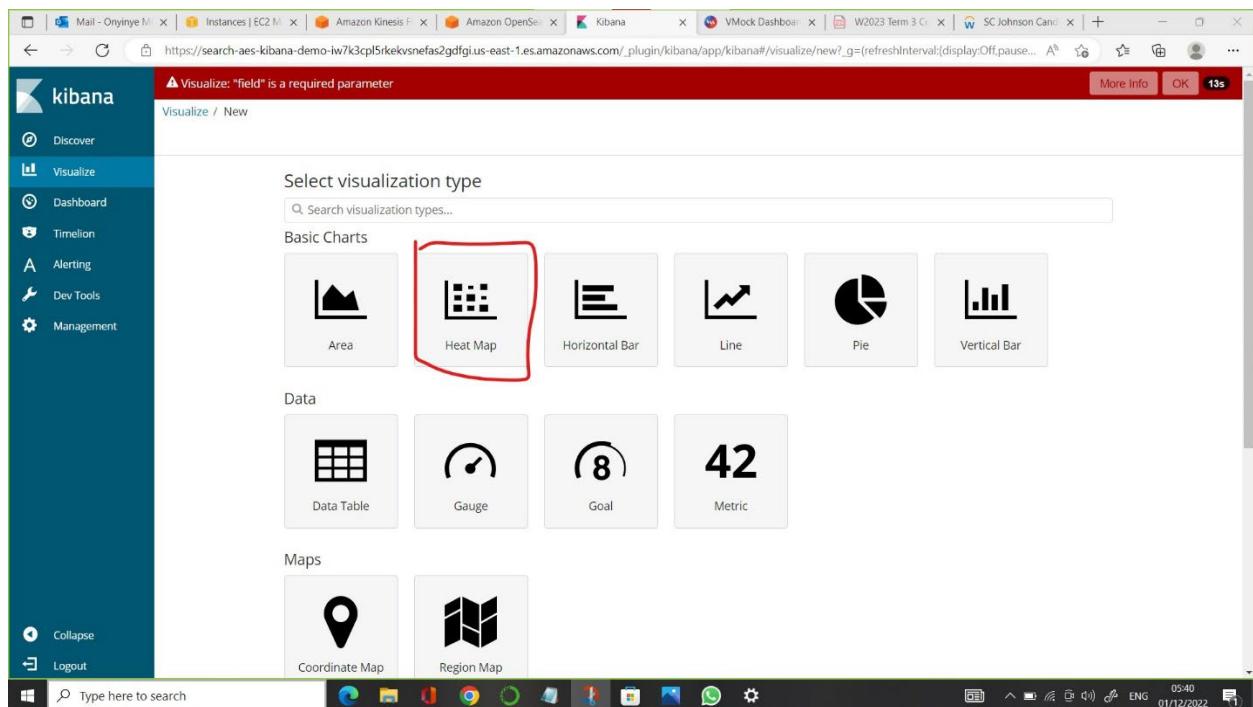
Task 5: Create the pie chart for your PoC

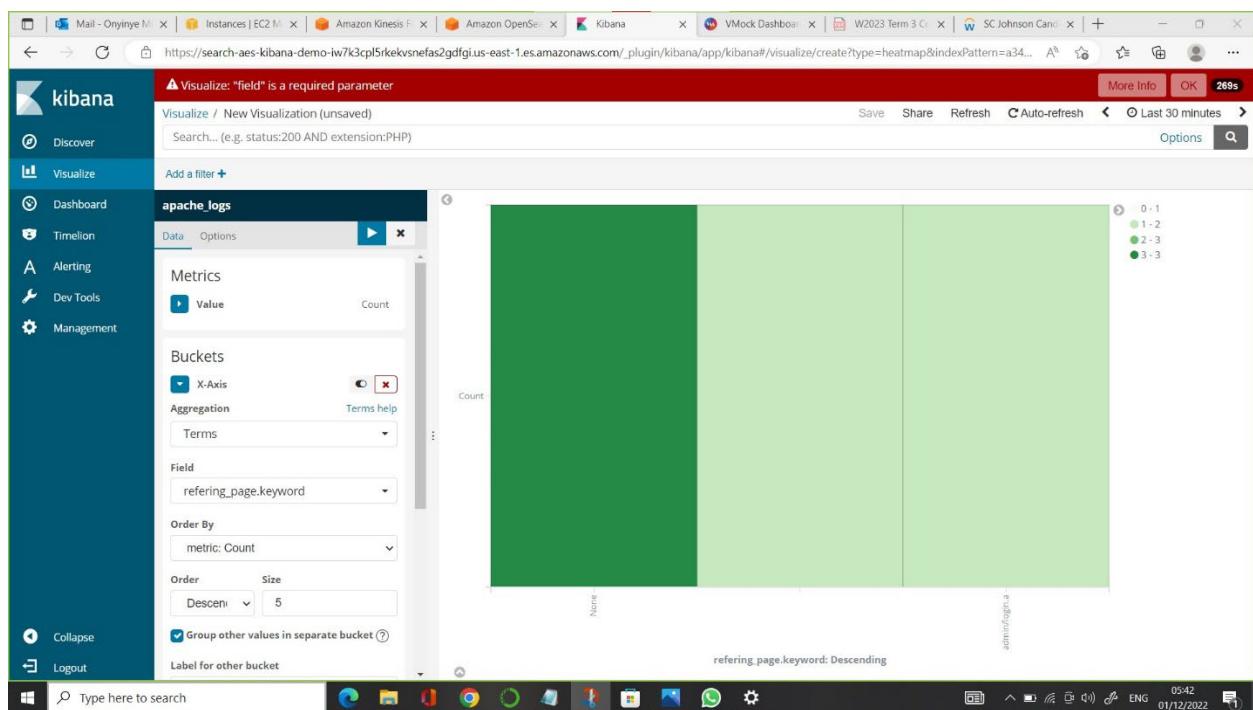
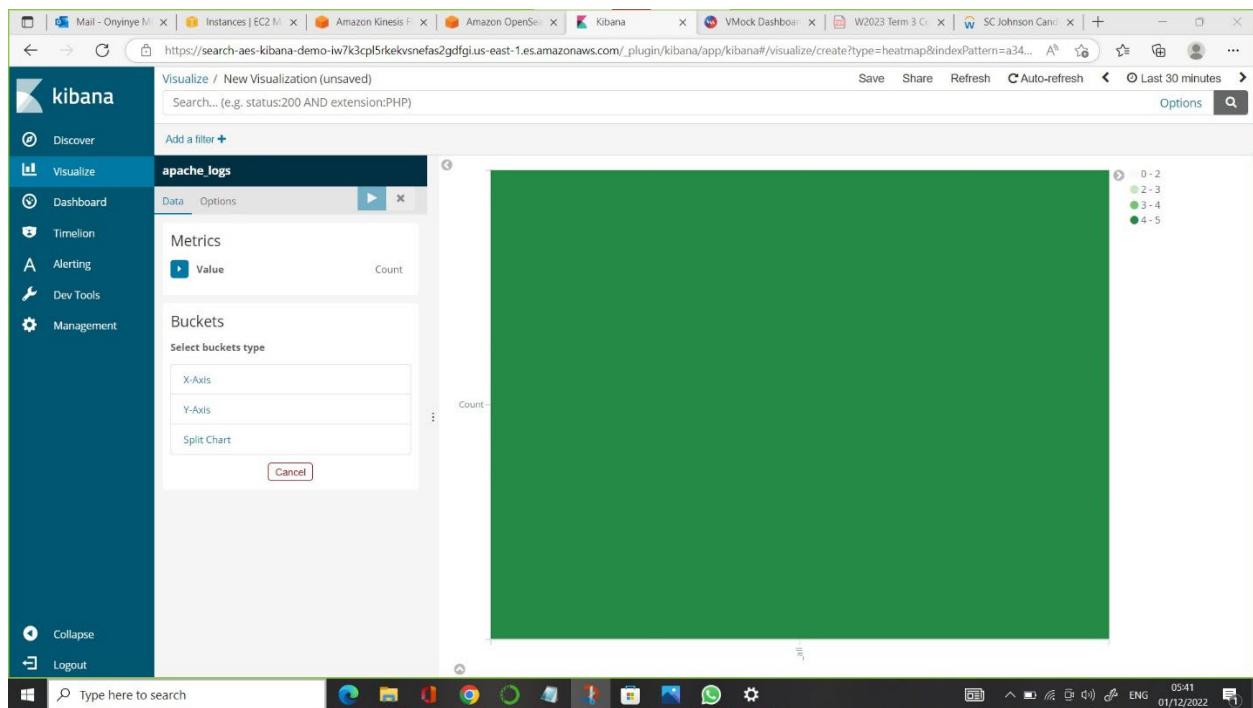
The screenshot shows the Kibana Visualize interface with a new visualization being created. The visualization type is set to 'pie'. A modal dialog is open, prompting the user to select a bucket type for the slices. The options shown are 'Split Slices' and 'Split Chart'. The main area of the screen is dominated by a large, solid teal circle, which is the visual representation of the pie chart.

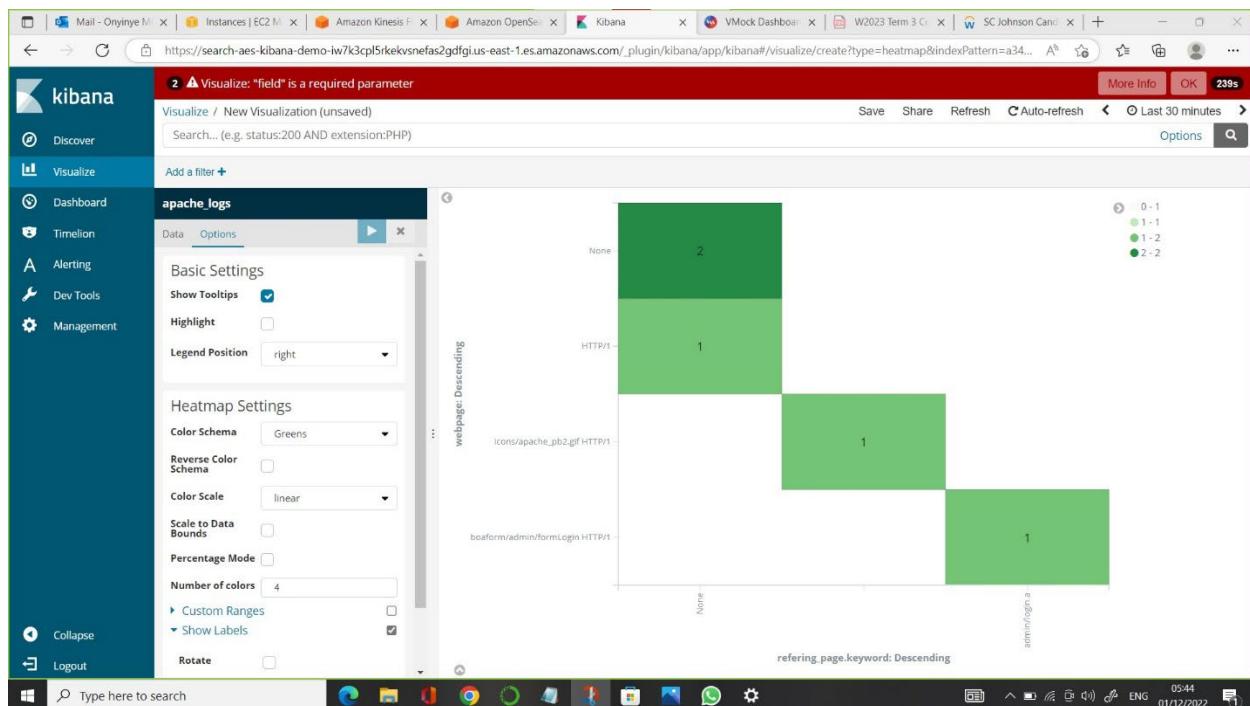
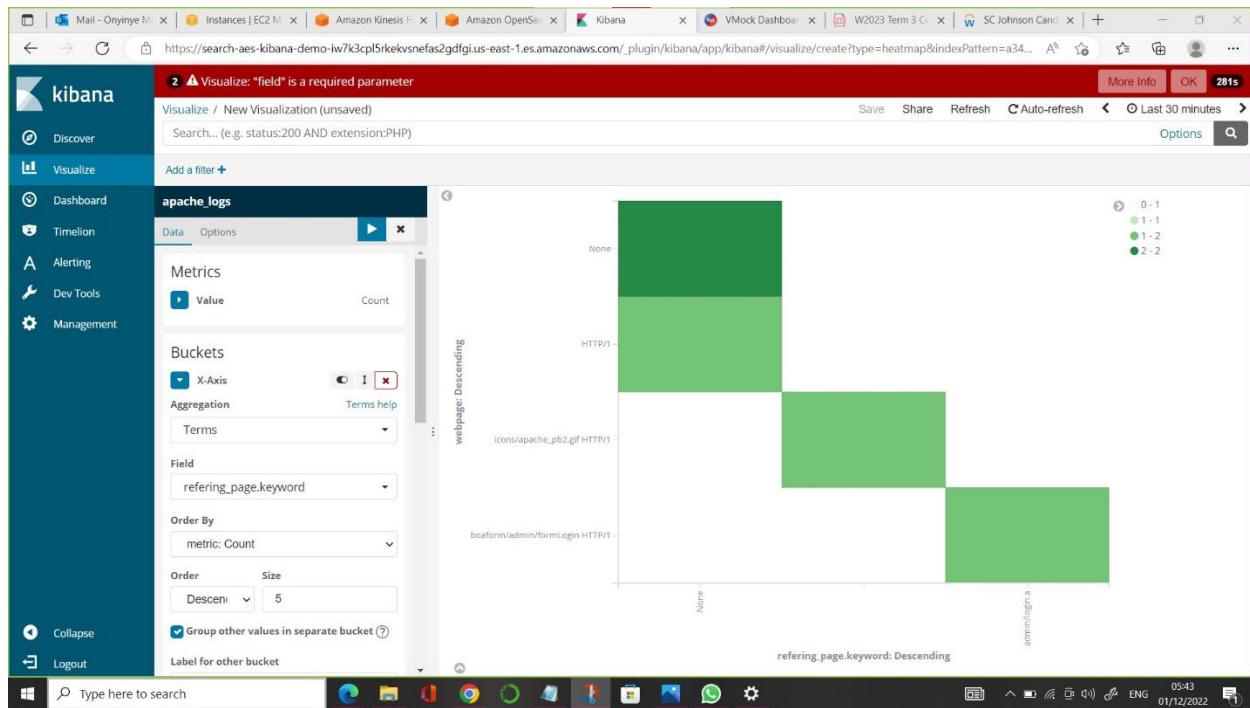




Task 6: Create a heat map for your PoC







Lab Complete.

Lab 8: Analyze IoT Data with AWS IoT Analytics

Objectives:

- Access the AWS IoT Analytics Service in the AWS Management Console
- Create an AWS IoT Analytics channel
- Create an AWS IoT Analytics data store
- Create an AWS IoT Analytics pipeline
- Create an AWS IoT Core rule
- Query an AWS IoT Analytics data store

Task 1: Create a channel

The screenshot shows the AWS IoT Analytics console interface. On the left, there's a sidebar with 'AWS IoT Analytics' at the top, followed by 'Channels' (which is selected), 'Pipelines', 'Data stores', 'Datasets', and 'Notebooks'. Below that are 'Settings', 'Documentation', 'Forums', and 'Contact us'. At the bottom of the sidebar, there's a link to 'New console experience' and a 'Tell us what you think' button. The main content area has a blue header bar with the message 'Introducing the new IoT Analytics console experience' and 'We're updating the console experience for you. Try the new experiences and let us know what you think. You can turn off the new experience from the navigation menu.' A green success message box says 'You successfully created a channel, mychannel.' with a 'View channel' button. Below this is a table titled 'Channels (1)'. The table has columns: Name, Storage type, Status, Last message arrival time, Created, and Last updated. It shows one row for 'mychannel' with 'Service managed' as the storage type, 'Active' status, and both 'Created' and 'Last updated' times as 'Dec 1, 2022 6:00:30 AM -0500'. At the bottom of the page, there's a feedback section, a search bar, and a footer with copyright information and links to 'Privacy', 'Terms', and 'Cookie preferences'.

Channel successfully created.

Task 2: Create a data store

The screenshot shows the AWS IoT Analytics console interface. On the left, a sidebar menu includes 'Data stores' under the 'Pipelines' section. The main content area displays a success message: 'You successfully created a channel, mychannel.' and 'You successfully created a data store, my_datastore.' Below this, a table lists the data store 'my_datastore' with details: Name (my_datastore), Status (Active), Last message arrival time (No message received or updated), Storage information (Service managed), File format (JSON), Created (Dec 1, 2022 6:01:45 AM -0500), and Last updated (Dec 1, 2022 6:01:45 AM -0500). A 'Create data store' button is visible at the top right of the table.

Data store created successfully.

Task 3: Create a pipeline

The screenshot shows the AWS IoT Analytics console interface. On the left, a sidebar menu includes 'Activities' under the 'Pipelines' section. The main content area displays a success message: 'You successfully created a pipeline, my_pipeline.' Below this, a table lists the pipeline 'my_pipeline' with details: Name (my_pipeline), Created (Dec 1, 2022 6:05:44 AM -0500), and Last updated (Dec 1, 2022 6:05:44 AM -0500). A 'Create pipeline' button is visible at the top right of the table.

Task 4: Create an AWS IoT Core rule

Task 4.1 Create the AWS IoT Core rule

The screenshot shows the 'Specify rule properties' step of the AWS IoT Core rule creation wizard. On the left, a sidebar lists navigation options like 'Connect', 'Test', and 'Manage'. The main area shows a 'Rule properties' form with fields for 'Rule name' (set to 'Send_IOT'), 'Rule description - optional' (containing 'A description of your new rule'), and 'Tags - optional' (empty). Buttons for 'Cancel' and 'Next' are at the bottom right.

Specifying rule properties

The screenshot shows the 'Configure SQL statement' step of the rule creation wizard. It displays a large text input field containing the SQL query 'SELECT * FROM 'iot/aus_weather''. Below the input field, a status bar indicates 'SQL Line 1, Column 32'. Navigation buttons for 'Cancel', 'Previous', and 'Next' are visible at the bottom right.

SQL Statement

The screenshot shows the AWS IoT Rule Actions configuration interface. On the left, a sidebar lists various AWS services and device management options. The main area is titled "Attach rule actions" and "Step 4: Review and create". A SQL query "SELECT * FROM 'iot/aus_weather'" is displayed in a code editor-like box. Below it, the "Rule actions" section is titled "Rule actions" and describes how to define actions for matched messages. An "Action 1" section is shown, where "IoT Analytics" is selected as the action type. The "Channel name" is set to "mychannel". There are options for "Batch mode" and "Use batch mode". An "IAM role" dropdown contains "IoTLabAccessRole", with a note that AWS IoT will automatically create a policy. A "Create new role" button is also present. At the bottom, a "Add rule action" button is available.

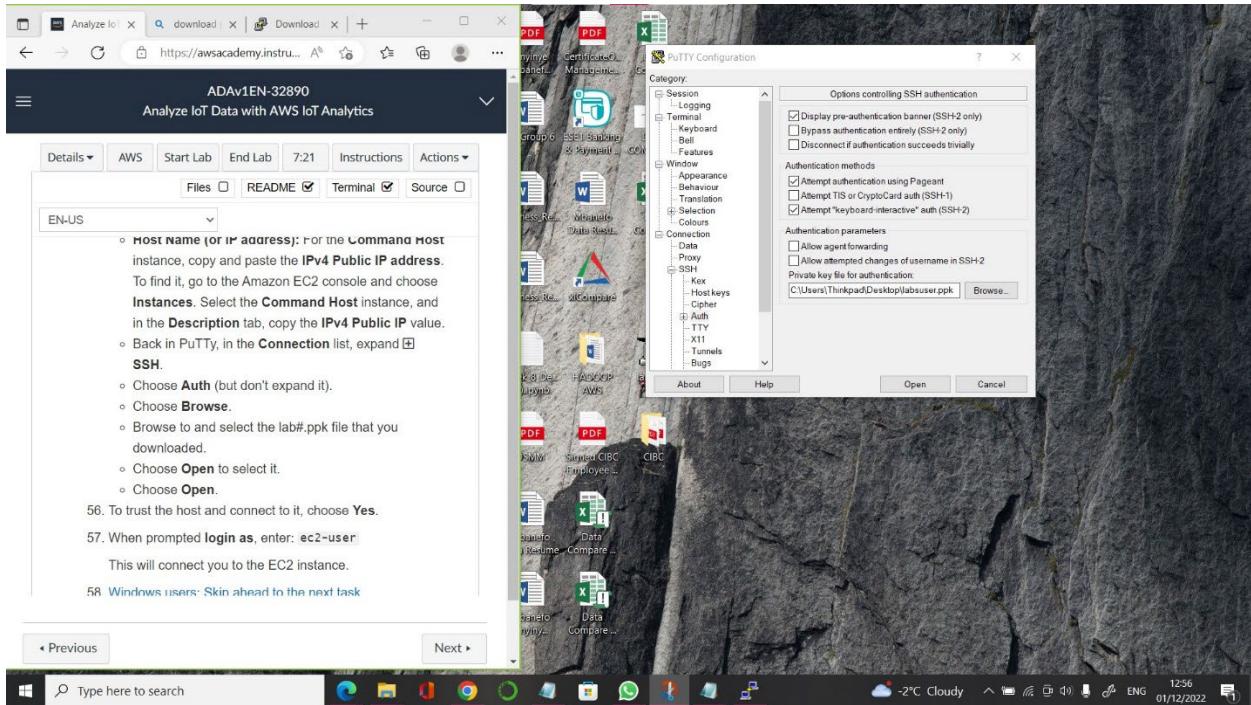
Rule Actions

The screenshot shows the AWS IoT Rules list page. A green success message at the top says "Successfully created rule Send_IOT." The main area displays the "Rules (1) Info" section, which includes a table of rules. The table has columns for Name, Status, Rule topic, and Created date. One rule is listed: "Send_IOT" (Status: Active, Rule topic: iot/aus_weather, Created date: December 01, 2022, 12:27:10 (UTC-0500)). Action buttons for Activate, Deactivate, Edit, and Delete are shown above the table, along with a "Create rule" button. A search bar and pagination controls are also present.

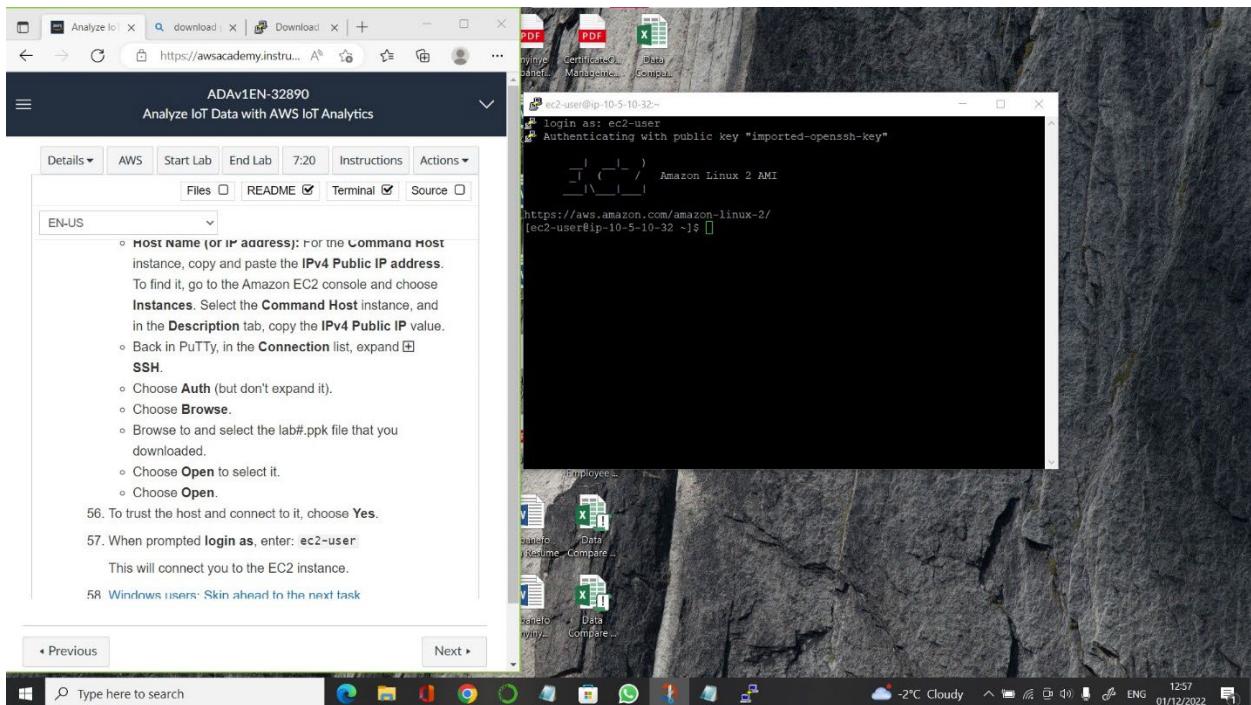
Send_IOT rule created.

Task 4.2 Configure your environment to run the Python script

Task 4.2.1: Windows users – Use SSH to connect



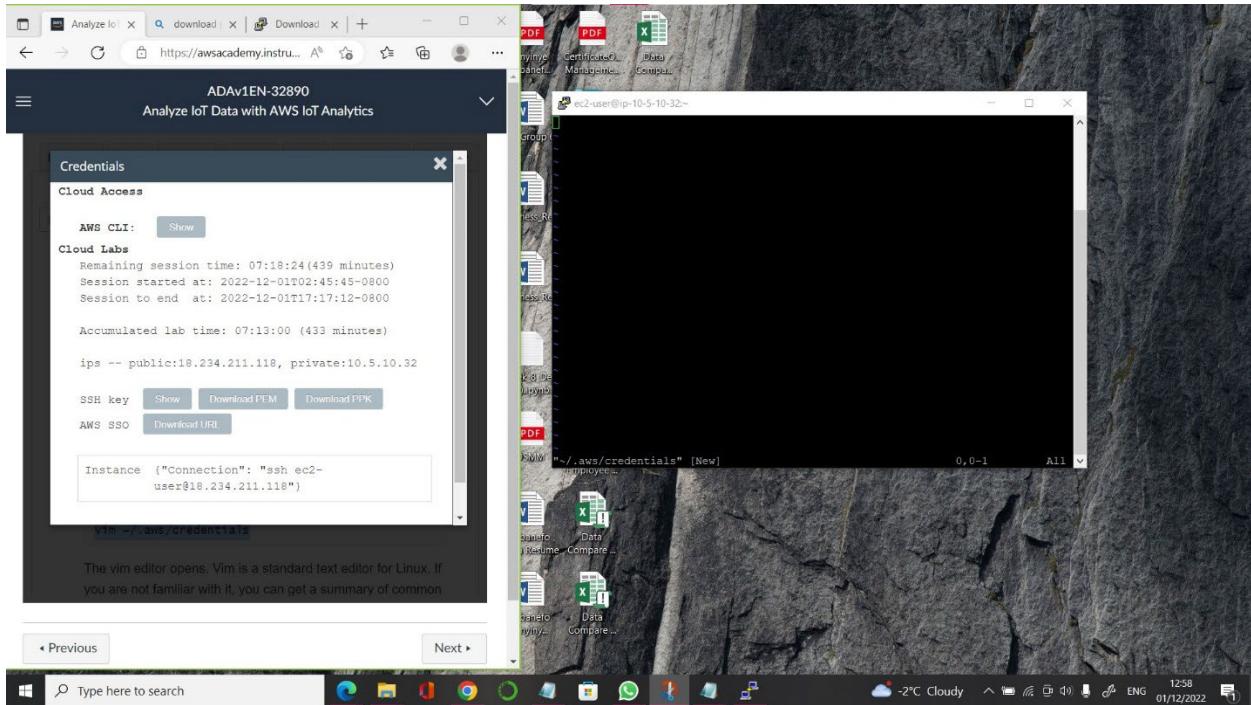
Putty Configuration



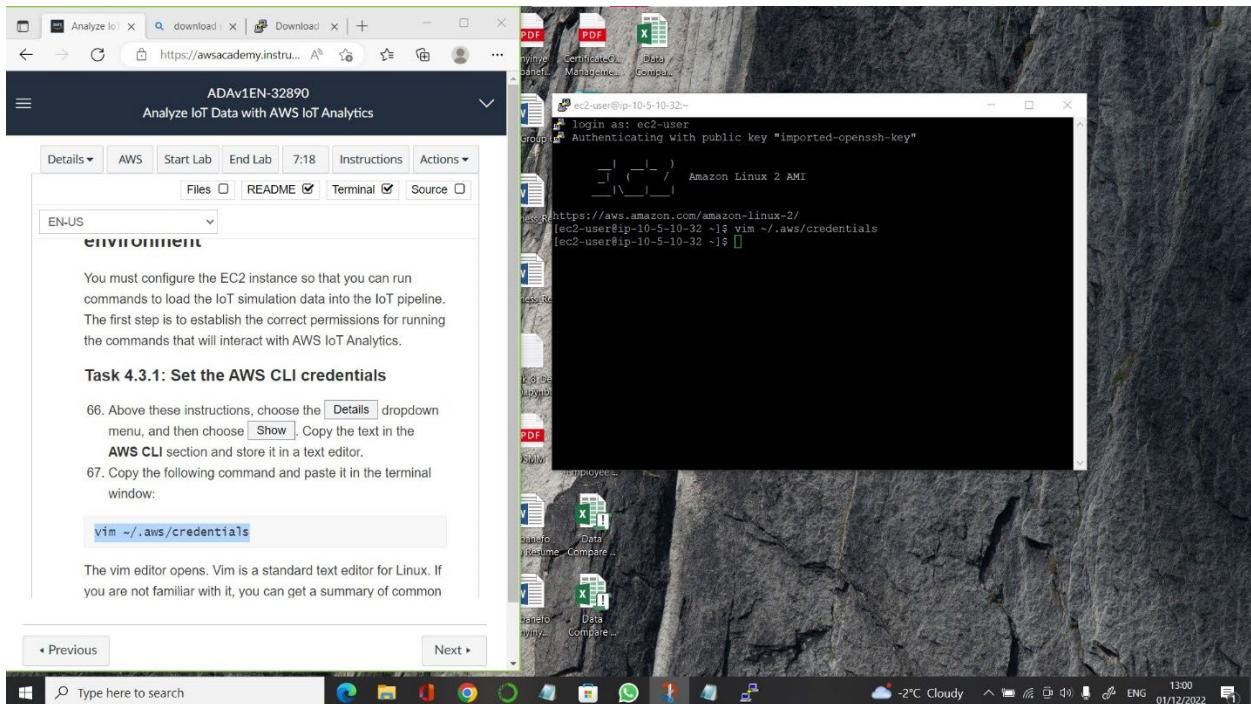
Connecting to ec2 instance

Task 4.3: Configure the Amazon EC2 environment

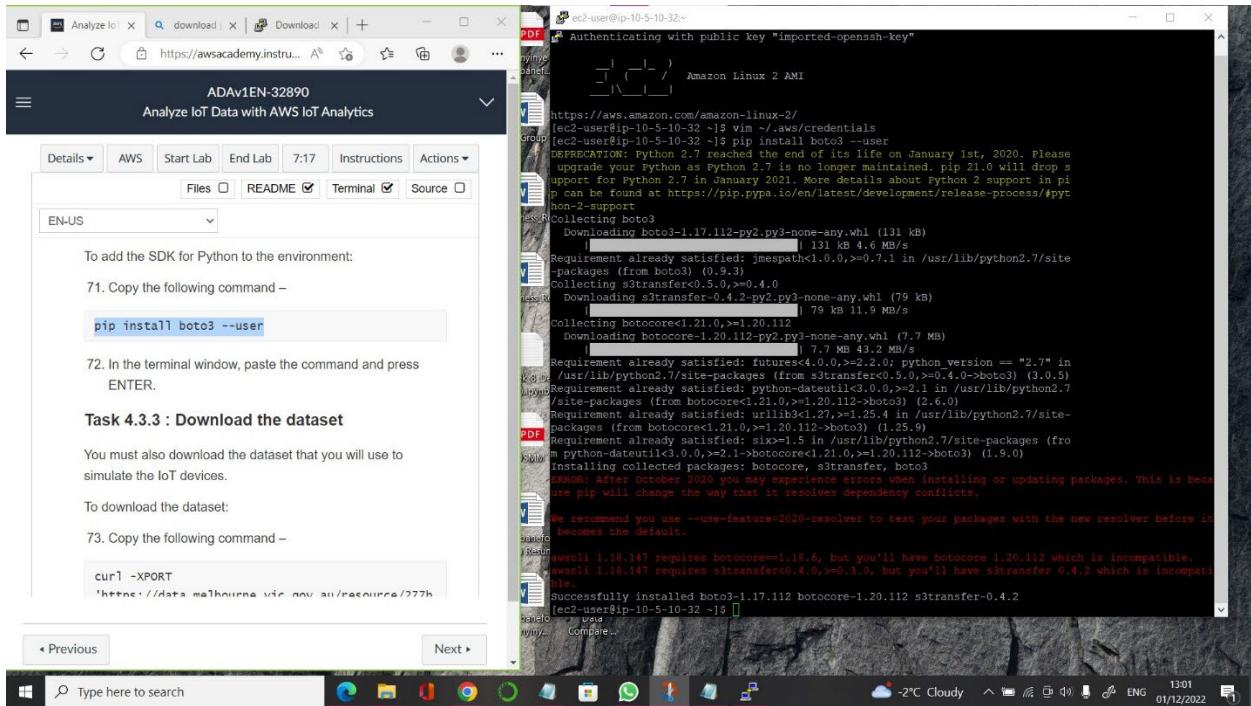
Task 4.3.1: Set the AWS CLI credentials



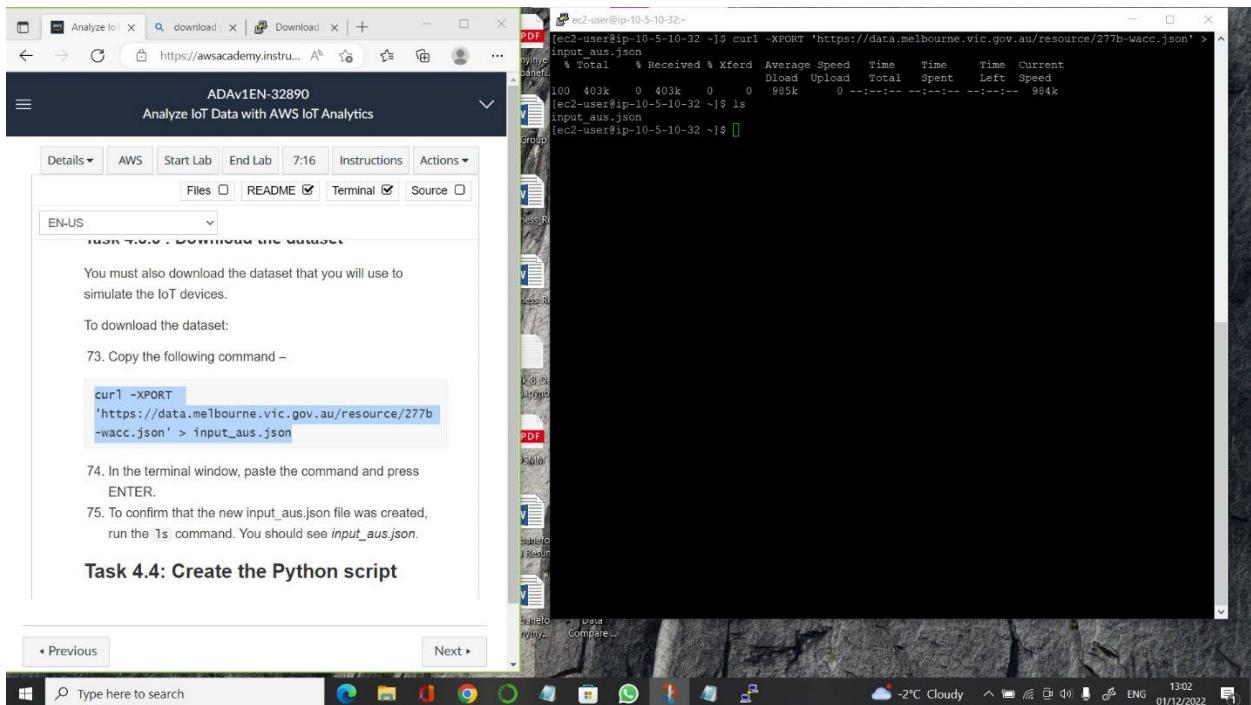
Downloading AWS CLI details



Task 4.3.2: Install the AWS SDK for Python (boto)



Task 4.3.3: Download the dataset



Task 4.4: Create the Python script

```
ec2-user@ip-10-5-10-32:~$ python upload_raw_data_iot.py

# An array of boto3 IoT clients
iotBoto3Client = [botoc3.client('iot-data') for i in range(processes)]

def publish_wrapper(lineID, line):
    # Publish raw data to AWS IoT
    client = iotBoto3Client[lineID % 4]

    line_read = line.strip()
    print "Publish: ", os.getpid(), lineID, line_read[:70], "..."

    payload = json.loads(line_read)

    client.publish(
        topic="iot/us/weather",
        qos=1,
        payload=json.dumps(payload))

if __name__ == '__main__':
    pool = mp.Pool(processes)
    jobs = []
    print "Begin Data Ingestion"
    for ID, line in enumerate(fileinput.input()):
        # Create job for each raw object
        res = jobs.append(pool.apply_async(publish_wrapper, (ID, line)))

    for job in jobs:
        job.get()

    print "Data Ingested Successfully"

upload raw data iot.py 37L, 950B
```

Task 5: Create a dataset

Name	Type	Triggers	Status	Created	Last updated
my_dataset	Query	No trigger has been set yet.	Active	Dec 1, 2022 1:08:34 PM -0500	Dec 1, 2022 1:08:34 PM -0500

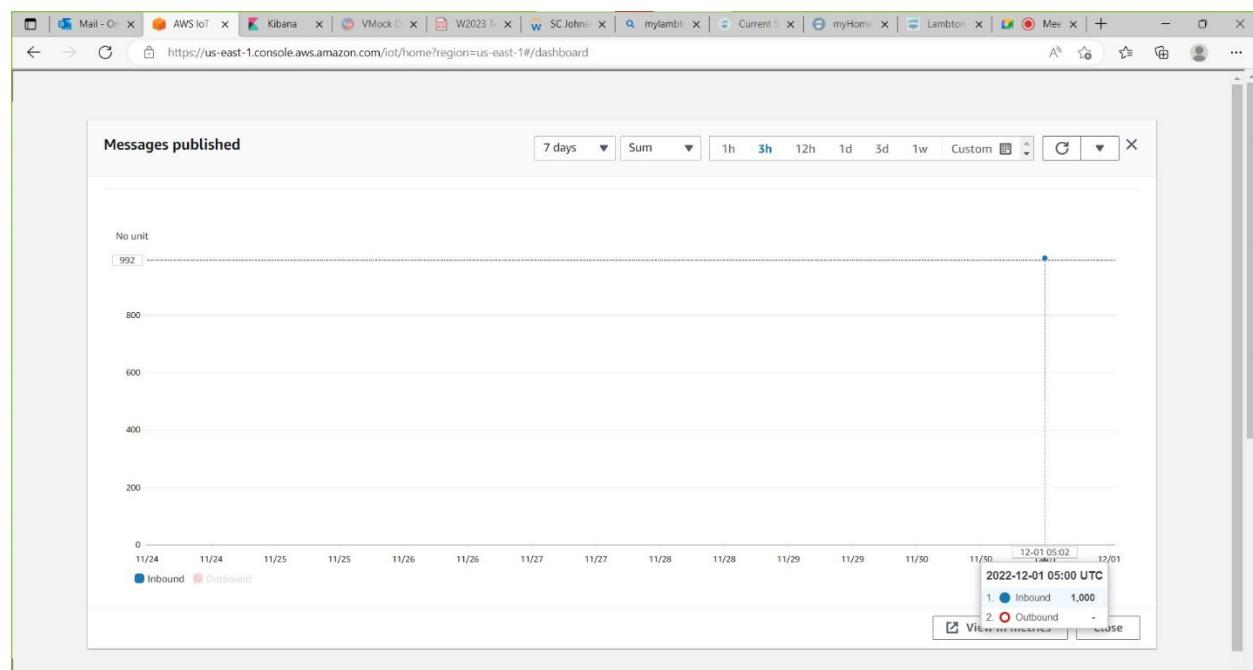
Task 5.1: Load the data to the MQTT topic

```
ec2-user@ip-10-5-10-32:~$ Publish: 1562 952 {"timestamp": "2015-05-31T16:00:00.000", "mac": "0013a20040b31563", "board": ..}
Publish: 1560 953 {"timestamp": "2015-05-31T15:50:00.000", "mac": "0013a20040b31571", "board": ..}
Publish: 1560 954 {"timestamp": "2015-05-31T15:40:00.000", "mac": "0013a20040b31571", "board": ..}
Publish: 1560 955 {"timestamp": "2015-05-31T15:35:00.000", "mac": "0013a20040b31583", "board": ..}
Publish: 1560 956 {"timestamp": "2015-05-31T15:30:00.000", "mac": "0013a20040b31583", "board": ..}
Publish: 1561 957 {"timestamp": "2015-05-31T15:25:00.000", "mac": "0013a20040b31583", "board": ..}
Publish: 1559 958 {"timestamp": "2015-05-31T15:10:00.000", "mac": "0013a20040b31571", "board": ..}
Publish: 1562 959 {"timestamp": "2015-05-31T15:05:00.000", "mac": "0013a20040b31583", "board": ..}
Publish: 1560 960 {"timestamp": "2015-05-31T14:55:00.000", "mac": "0013a20040b31583", "board": ..}
Publish: 1561 961 {"timestamp": "2015-05-31T14:45:00.000", "mac": "0013a20040b31583", "board": ..}
Publish: 1562 962 {"timestamp": "2015-05-31T14:40:00.000", "mac": "0013a20040b31571", "board": ..}
Publish: 1559 963 {"timestamp": "2015-05-31T14:35:00.000", "mac": "0013a20040b31583", "board": ..}
Publish: 1561 964 {"timestamp": "2015-05-31T14:30:00.000", "mac": "0013a20040b31571", "board": ..}
Publish: 1560 965 {"timestamp": "2015-05-31T14:25:00.000", "mac": "0013a20040b31583", "board": ..}
Publish: 1562 966 {"timestamp": "2015-05-31T14:20:00.000", "mac": "0013a20040b31571", "board": ..}
Publish: 1559 967 {"timestamp": "2015-05-31T14:15:00.000", "mac": "0013a20040b31583", "board": ..}
Publish: 1561 968 {"timestamp": "2015-05-31T14:10:00.000", "mac": "0013a20040b31583", "board": ..}
Publish: 1560 969 {"timestamp": "2015-05-31T14:05:00.000", "mac": "0013a20040b31583", "board": ..}
Publish: 1559 970 {"timestamp": "2015-05-31T13:55:00.000", "mac": "0013a20040b31583", "board": ..}
Publish: 1562 971 {"timestamp": "2015-05-31T13:50:00.000", "mac": "0013a20040b31583", "board": ..}
Publish: 1560 972 {"timestamp": "2015-05-31T13:45:00.000", "mac": "0013a20040b31583", "board": ..}
Publish: 1561 973 {"timestamp": "2015-05-31T12:55:00.000", "mac": "0013a20040b31556", "board": ..}
Publish: 1562 974 {"timestamp": "2015-05-31T12:50:00.000", "mac": "0013a20040b31583", "board": ..}
Publish: 1559 975 {"timestamp": "2015-05-31T12:20:00.000", "mac": "0013a20040b31583", "board": ..}
Publish: 1560 976 {"timestamp": "2015-05-31T12:15:00.000", "mac": "0013a20040b31571", "board": ..}
Publish: 1561 977 {"timestamp": "2015-05-31T12:10:00.000", "mac": "0013a20040b31583", "board": ..}
Publish: 1562 978 {"timestamp": "2015-05-31T12:05:00.000", "mac": "0013a20040b31571", "board": ..}
Publish: 1560 979 {"timestamp": "2015-05-31T12:00:00.000", "mac": "0013a20040b31556", "board": ..}
Publish: 1559 980 {"timestamp": "2015-05-31T12:00:00.000", "mac": "0013a20040b31583", "board": ..}
Publish: 1562 981 {"timestamp": "2015-05-31T11:55:00.000", "mac": "0013a20040b31571", "board": ..}
Publish: 1560 982 {"timestamp": "2015-05-31T11:50:00.000", "mac": "0013a20040b31583", "board": ..}
Publish: 1561 983 {"timestamp": "2015-05-31T11:45:00.000", "mac": "0013a20040b31583", "board": ..}
Publish: 1561 984 {"timestamp": "2015-05-31T11:35:00.000", "mac": "0013a20040b31571", "board": ..}
Publish: 1562 985 {"timestamp": "2015-05-31T11:30:00.000", "mac": "0013a20040b31583", "board": ..}
Publish: 1561 986 {"timestamp": "2015-05-31T11:25:00.000", "mac": "0013a20040b31571", "board": ..}
Publish: 1562 987 {"timestamp": "2015-05-31T11:20:00.000", "mac": "0013a20040b31583", "board": ..}
Publish: 1561 988 {"timestamp": "2015-05-31T11:15:00.000", "mac": "0013a20040b31571", "board": ..}
Publish: 1562 989 {"timestamp": "2015-05-31T11:10:00.000", "mac": "0013a20040b31583", "board": ..}
Publish: 1561 990 {"timestamp": "2015-05-31T11:05:00.000", "mac": "0013a20040b31571", "board": ..}
Publish: 1560 991 {"timestamp": "2015-05-31T11:00:00.000", "mac": "0013a20040b31583", "board": ..}
Publish: 1562 992 {"timestamp": "2015-05-31T10:50:00.000", "mac": "0013a20040b31583", "board": ..}
Publish: 1561 993 {"timestamp": "2015-05-31T10:45:00.000", "mac": "0013a20040b31571", "board": ..}
Publish: 1562 994 {"timestamp": "2015-05-31T10:40:00.000", "mac": "0013a20040b31583", "board": ..}
Publish: 1560 995 {"timestamp": "2015-05-31T10:35:00.000", "mac": "0013a20040b31583", "board": ..}
Publish: 1561 996 {"timestamp": "2015-05-31T10:20:00.000", "mac": "0013a20040b31583", "board": ..}
Publish: 1562 997 {"timestamp": "2015-05-31T10:15:00.000", "mac": "0013a20040b31556", "board": ..}
Publish: 1559 998 {"timestamp": "2015-05-31T10:10:00.000", "mac": "0013a20040b31583", "board": ..}
Publish: 1560 999 {"timestamp": "2015-05-31T10:05:00.000", "mac": "0013a20040b31556", "board": ..}
Data Ingested Successfully
[ec2-user@ip-10-5-10-32 ~]$
```

Data Ingested successfully

Task 6: Access the query results in your dataset

Task 6.1: Check that the messages have finished publishing



Task 6.2: Access the query results

The screenshot shows the AWS IoT Analytics console interface. On the left, a sidebar lists 'Channels', 'Pipelines', 'Data stores', 'Datasets' (which is selected), and 'Notebooks'. The main content area has a blue header bar with the message: 'Introducing the new IoT Analytics console experience. We're updating the console experience for you. Try the new experiences and let us know what you think. You can turn off the new experience from the navigation menu.' Below this, a green bar says: 'You've successfully started the query for your dataset. Dataset content version: a19fa66d-d5dc-43d5-9cc6-a3c568fca470.' A table provides dataset details: ARN info (arn:aws:iotanalytics:us-east-1:486241939139:dataset/my_dataset), Type (Query), Status (Active), Created (Dec 1, 2022 1:08:34 PM -0500), and Last updated (Dec 1, 2022 1:08:34 PM -0500). Below the table are tabs for 'Details', 'Content' (which is selected), 'Schedule', 'Dataset content retention settings', 'Dataset content delivery rules', and 'Tags'. Under 'Content', a section titled 'Dataset contents (1)' shows a table with one row. The columns are 'Date' (Dec 1, 2022 1:15:00 PM -0500), 'Name' (a19fa66d-d5dc-43d5-9cc6-a3c568fca470), 'Status' (Succeeded, circled in red), and 'Duration' (2533 ms). At the bottom of the page, there's a feedback link 'Tell us what you think'.

The screenshot shows the 'Result preview' section of the AWS IoT Analytics console. It displays a table titled '134d16f9-a382-41f5-a283-659067e6b4c4.csv'. The table has 13 columns: boardtype, elevation, temp_max, boardid, timestamp, humidity_avg, temp_avg, humidity_min, light_max, light_min, mac, and longitude. There are 10 rows of data. The first few rows are: 1, 2.74, 7.4, 510, 2015-06-05T04:05:00.000, 54.4, 7.4, 54.4, 3.9, 3.9, 0013a20040b31571, 144.9413253; 1, 0.03, 8.1, 509, 2015-06-05T04:15:00.000, 63.7, 8.1, 63.7, 2.6, 2.6, 0013a20040b31583, 144.9404851; 1, 0.03, 8.1, 509, 2015-06-05T04:05:00.000, 63.9, 8.1, 63.9, 2.6, 2.6, 0013a20040b31583, 144.9404851; 1, 0.03, 8.1, 509, 2015-06-05T03:55:00.000, 63.4, 8.1, 63.4, 2.6, 2.6, 0013a20040b31583, 144.9404851; 1, 0.71, 8.1, 506, 2015-06-05T04:15:00.000, 55.4, 8.1, 55.4, 8.4, 8.4, 0013a20040b3155c, 144.9404816; 1, 2.74, 7.4, 510, 2015-06-05T03:45:00.000, 53.8, 7.4, 53.8, 3.8, 3.8, 0013a20040b31571, 144.9413253; 1, 0.03, 8.1, 509, 2015-06-05T03:45:00.000, 64.2, 8.1, 64.2, 2.6, 2.6, 0013a20040b31583, 144.9404851; 1, 2.74, 6.8, 510, 2015-06-05T02:50:00.000, 54.8, 6.8, 54.8, 2.3, 2.3, 0013a20040b31571, 144.9413253; 1, 0.03, 7.7, 509, 2015-06-05T03:05:00.000, 63.9, 7.7, 63.9, 2.5, 2.5, 0013a20040b31583, 144.9404851. At the bottom of the page, there's a feedback link 'Tell us what you think'.

Task 7: Query the dataset

Task 7.1 Create a dataset to find the average maximum temperature

The screenshot shows the AWS IoT Analytics console interface. On the left, a sidebar menu includes 'Datasets' (which is selected and highlighted in orange). The main content area displays a success message: 'You've successfully started the query for your dataset.' Below this, another message says 'You successfully create a dataset, findaverage.' A 'View dataset' button is present. The central part of the screen shows a table titled 'Datasets (2)' with two entries:

Name	Type	Triggers	Status	Created	Last updated
findaverage	Query	No trigger has been set yet.	Active	Dec 1, 2022 1:18:17 PM -0500	Dec 1, 2022 1:18:17 PM -0500
my_dataset	Query	No trigger has been set yet.	Active	Dec 1, 2022 1:08:34 PM -0500	Dec 1, 2022 1:08:34 PM -0500

This screenshot shows the 'Content' tab for the 'findaverage' dataset. It displays a success message: 'You've successfully started the query for your dataset.' Below this, another message says 'You successfully create a dataset, findaverage.' A 'View dataset' button is present. The central part of the screen shows a table titled 'Dataset contents (1)' with one entry:

Date	Name	Status	Duration
Dec 1, 2022 1:19:06 PM -0500	11fdbbe6e-8e18-4f3b-8e51-3ddd9257f4ab	Succeeded	2045 ms

A red circle highlights the 'Succeeded' status column for the first row.

Result Preview

The screenshot shows the AWS IoT Analytics Result Preview interface. At the top, there are three success messages:

- You've successfully started the query for your dataset. Dataset content version: a1f6ased-05dc-44d5-9ccb-a3c368fcfa470.
- You successfully created a dataset, findaverage.
- You've successfully started the query for your dataset. Dataset content version: 11fb0e1e-0e18-4f5b-8e51-5dddb257f4ab.

A green bar indicates "Last updated". Below this, the "Result preview" section shows a CSV file named "fcf0b6a9-c744-4a88-832c-879ae52a6fca.csv". A red circle highlights the value "8.429699999999999" in the first row under the column labeled "_colo". To the right of the CSV file is a "Download" button. Below the CSV preview is a table with columns: Date, Name, Status, and Duration. One row is shown:

Date	Name	Status	Duration
Dec 1, 2022 1:19:06 PM -0500	11fb0e1e-0e18-4f5b-8e51-5dddb257f4ab	Succeeded	2045 ms

The task bar at the bottom shows various icons and the date/time: 01/12/2022, -1°C Cloudy, 13:19, ENG, 01/12/2022.

Lab Complete.