मेघसङ्गणनस्य अध्ययनस्य द्वितीयदिनम्

2/13

Date
DELTA Pg No

Module - 2 Compute in the cloud

#1 compute → processing power needed to run application manage data and perform calculations.

Amazon Elastic compute cloud (AWS EC2)
→ Flexible, cost-effective, Quick
→ only pay for what you use

Multi-tenancy - Sharing underlying hardware between virtual machines.
Ex - Ek his building me alag alag kirayedar water, electricity Share karte hai.

→ we can provison windows / Linux and configure apps, databases on it.
→ we control networking
virtual scaling → increasing / decreasing resources on demand without changing hardware
Horizontal scaling - Add or renove machine / instances to handle load

Difference - Ek restaurant me crowd aagaya
virtual - single waiter ka workload badhao
Horizontal - Aur waiter lao 2-3.
Caas → compute as a service

#2

## Types of EC2 Instances

Amazon EC2 instance families
- General purpose
- Compute Optimised
- Memory optimised
- Accelerated computing
- Storage Optimized

Storage optimized
- High performance for locally stored data

| General purpose | Compute optimised | Memory optimized |
|---|---|---|
| • Balanced Resources | compute Intensive Task | Memory Intensive (Real time Analytics |
| • Diverse workload | Gaming Servers | Accelerated computing |
| • webservers | High Performance Task | • Floating point No calculation |
| • Code Repo | Scientific Modelling | • Graphics processing |
| | | • Data pattern matching |
| | | • Hardware accelerator |

#3

API - Application Programming Interface.
call through
→ AWS Management console
→ AWS Command Line Interface.
→ AWS Software Development Kit (SDK

AWS Management Console
→ set up Test environments
→ view AWS bills
→ view Monitoring
→ work with non-technical Resources.

AWS CLI → Make API calls using the terminal of your machine.

AWS SDK - Interact with AWS resources through various programming languages.

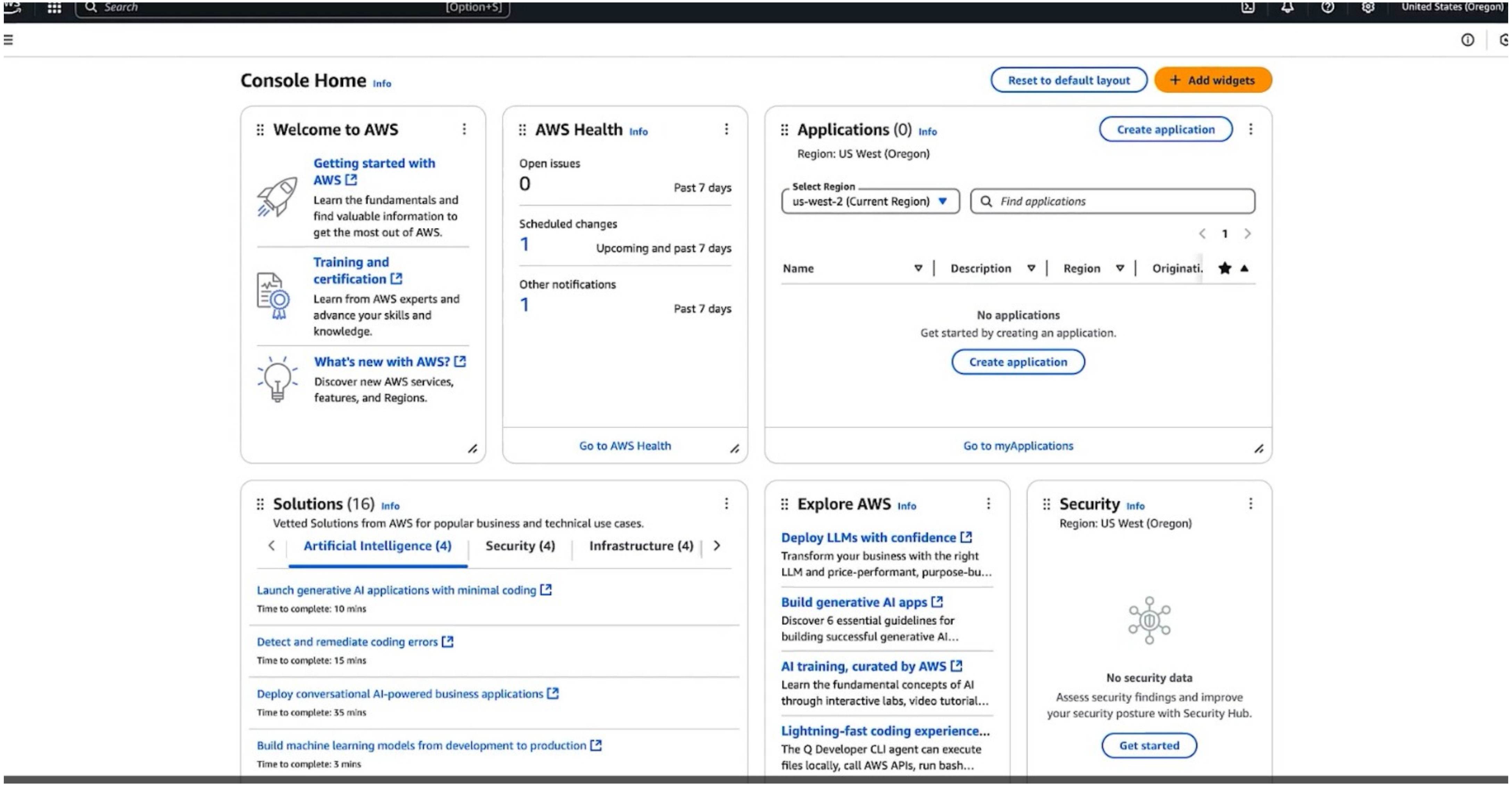
#4 Demo of Amazon $EC_2$ Launch

#5 Amazon $EC_2$ Pricing
→ ondemand
→ Savings Plan (upto 72% savings)
→ Reserved Instances (upto 75% off with 1year/3year term)

| | | |
|---|---|---|
| Allupfront | Partial upfront | No upfront |

~~Sport~~
→ Spot Instances - [upto 90% off
2minute warning
AWS can ~~take~~ Reclaim it anytime

→ Dedicated Host → Isolated, security sensitive
exclusive



Search [Option+S]

United States (Oregon)

Console Home Info

Reset to default layout

+ Add widgets

Welcome to AWS

Getting started with AWS

Learn the fundamentals and find valuable information to get the most out of AWS.

Training and certification

Learn from AWS experts and advance your skills and knowledge.

What's new with AWS?

Discover new AWS services, features, and Regions.

AWS Health Info

Open issues

0 Past 7 days

Scheduled changes

1 Upcoming and past 7 days

Other notifications

1 Past 7 days

Go to AWS Health

Applications (0) Info

Create application

Region: US West (Oregon)

Select Region

us-west-2 (Current Region)

Find applications

| Name | Description | Region | Originati. |
|---|---|---|---|

No applications

Get started by creating an application.

Create application

Go to myApplications

Solutions (16) Info

Vetted Solutions from AWS for popular business and technical use cases.

Artificial Intelligence (4) | Security (4) | Infrastructure (4)

Launch generative AI applications with minimal coding

Time to complete: 10 mins

Detect and remediate coding errors

Time to complete: 15 mins

Deploy conversational AI-powered business applications

Time to complete: 35 mins

Build machine learning models from development to production

Time to complete: 3 mins

Explore AWS Info

Deploy LLMs with confidence

Transform your business with the right LLM and price-performant, purpose-bu...

Build generative AI apps

Discover 6 essential guidelines for building successful generative AI...

AI training, curated by AWS

Learn the fundamental concepts of AI through interactive labs, video tutorial...

Lightning-fast coding experience...

The Q Developer CLI agent can execute files locally, call AWS APIs, run bash...

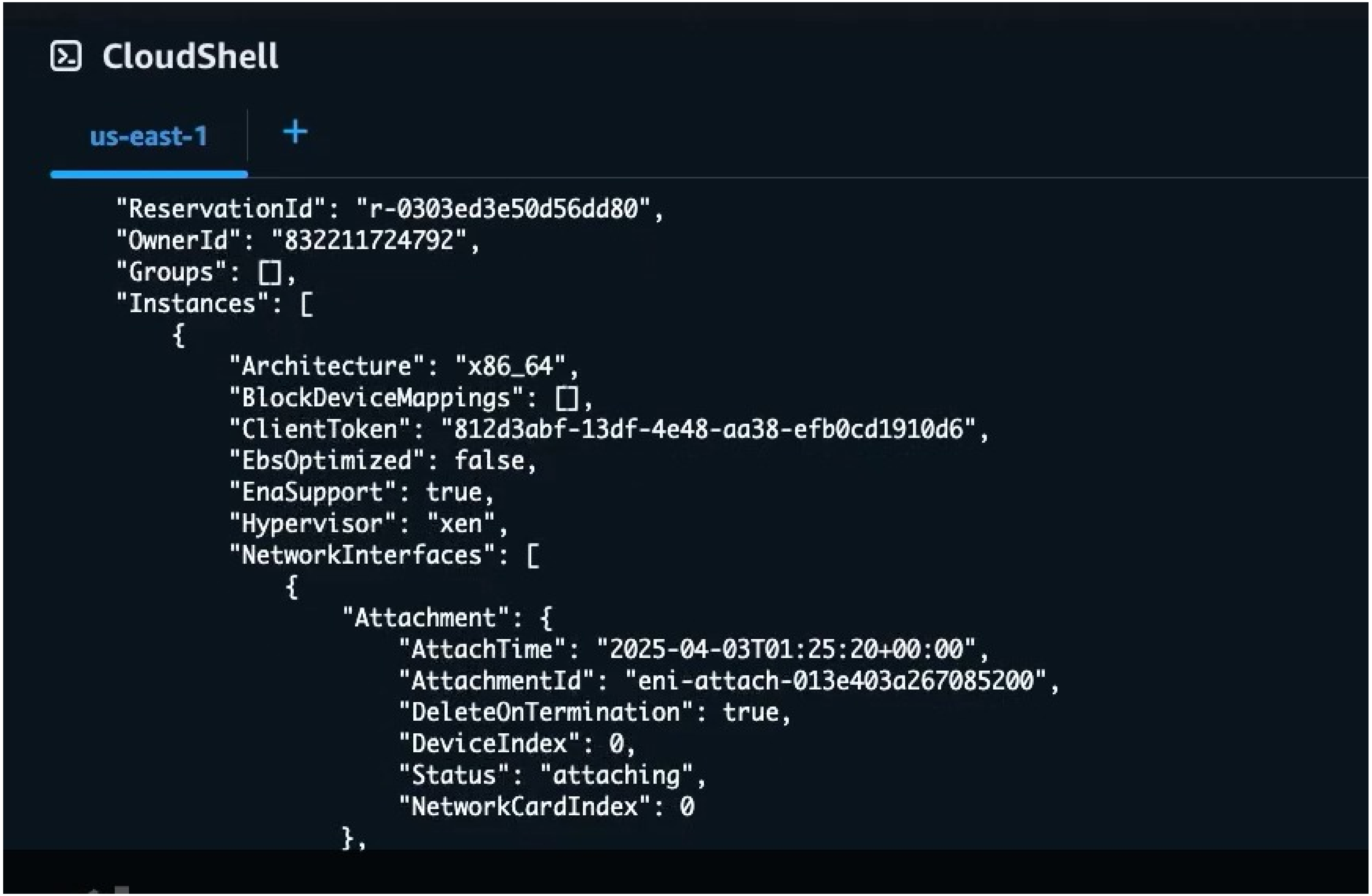
Security Info

Region: US West (Oregon)

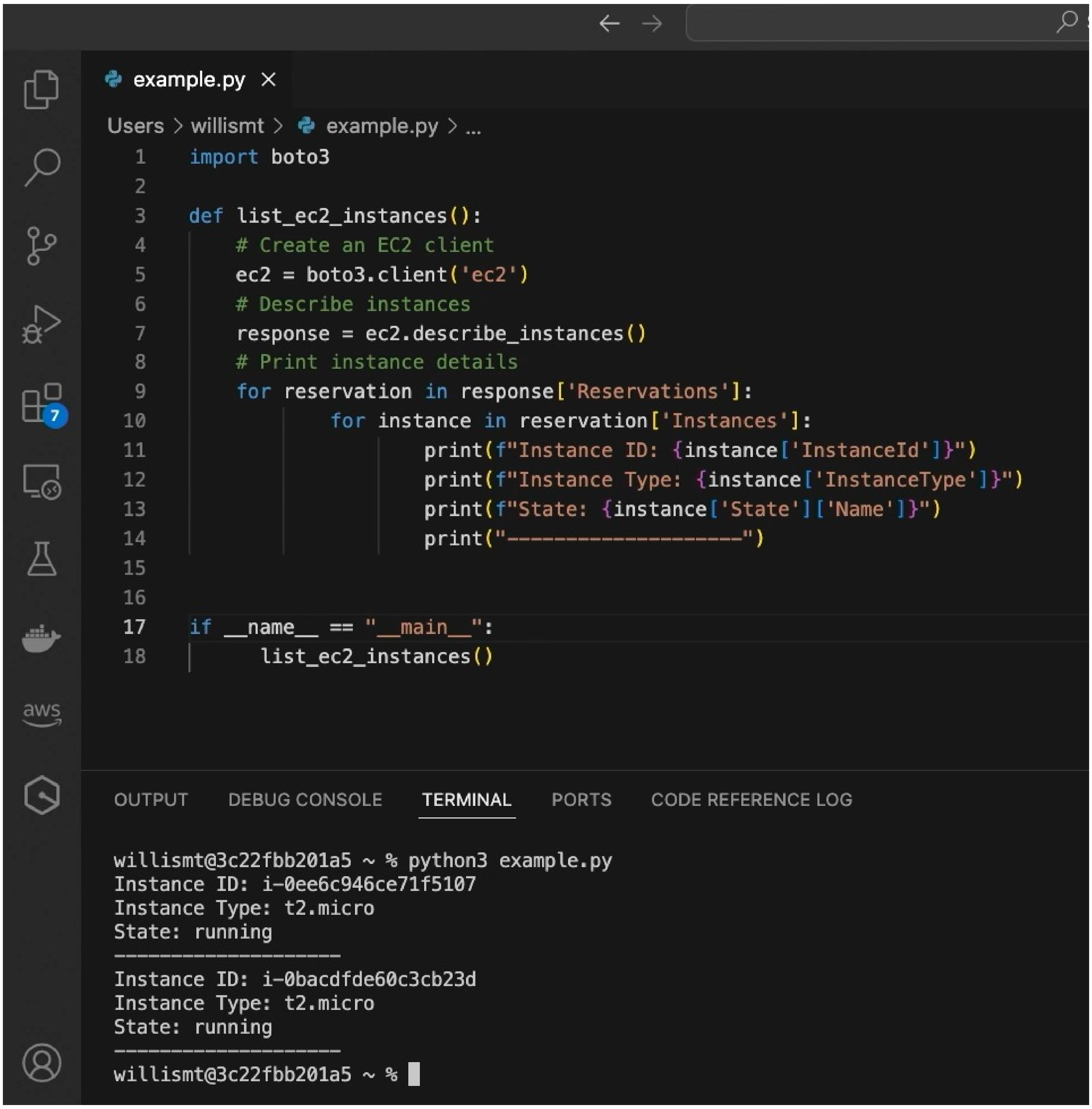
No security data

Assess security findings and improve your security posture with Security Hub.

Get started



```
    "ReservationId": "r-0303ed3e50d56dd80",
    "OwnerId": "832211724792",
    "Groups": [],
    "Instances": [
        {
            "Architecture": "x86_64",
            "BlockDeviceMappings": [],
            "ClientToken": "812d3abf-13df-4e48-aa38-efb0cd1910d6",
            "EbsOptimized": false,
            "EnaSupport": true,
            "Hypervisor": "xen",
            "NetworkInterfaces": [
                {
                    "Attachment": {
                        "AttachTime": "2025-04-03T01:25:20+00:00",
                        "AttachmentId": "eni-attach-013e403a267085200",
                        "DeleteOnTermination": true,
                        "DeviceIndex": 0,
                        "Status": "attaching",
                        "NetworkCardIndex": 0
                    },
```


example.py

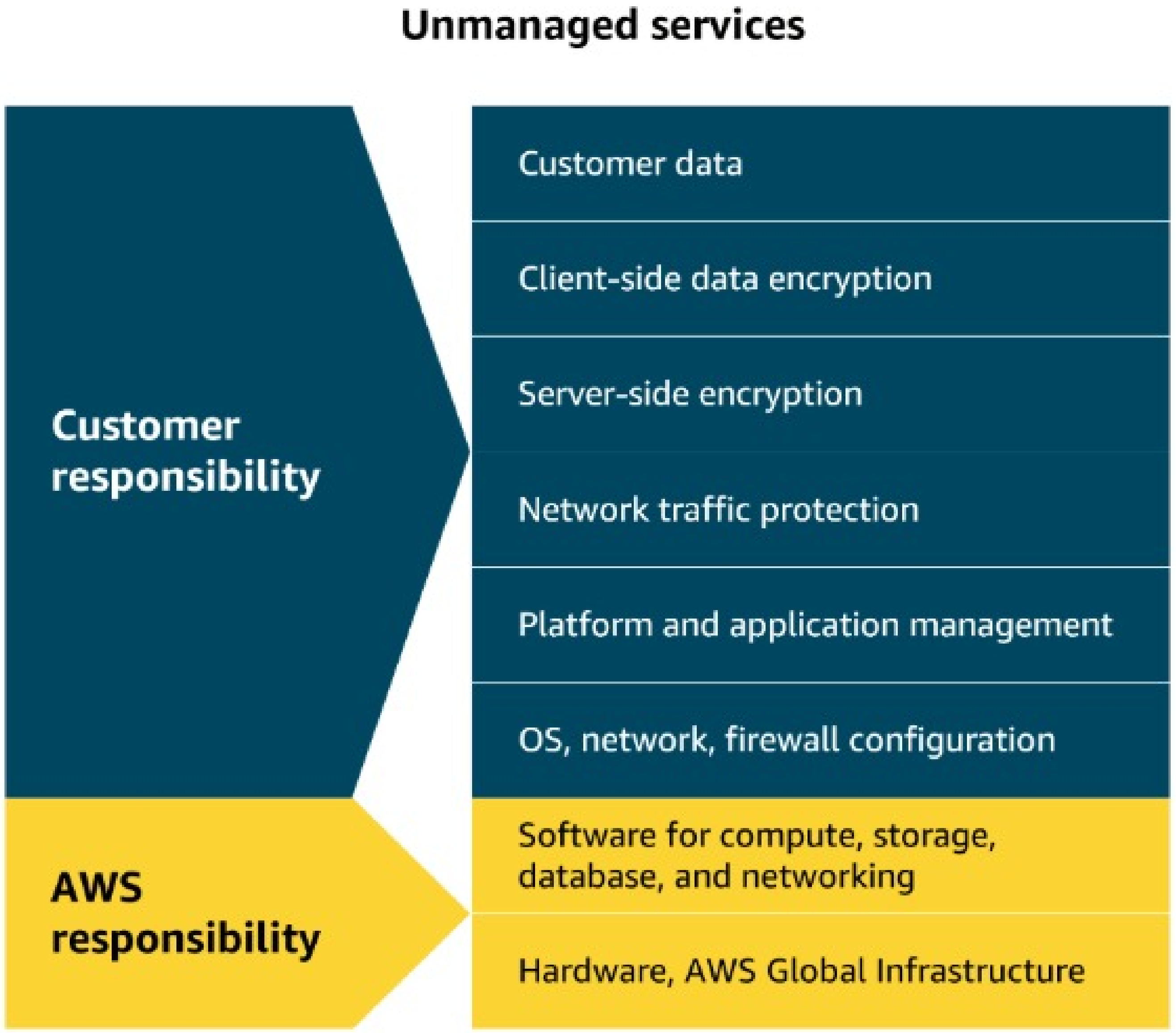
Users > willismt > example.py > ...

```python
import boto3

def list_ec2_instances():
    # Create an EC2 client
    ec2 = boto3.client('ec2')
    # Describe instances
    response = ec2.describe_instances()
    # Print instance details
    for reservation in response['Reservations']:
            for instance in reservation['Instances']:
                    print(f"Instance ID: {instance['InstanceId']}")
                    print(f"Instance Type: {instance['InstanceType']}")
                    print(f"State: {instance['State']['Name']}")
                    print("--------------------")


if __name__ == "__main__":
        list_ec2_instances()
```

```
willismt@3c22fbb201a5 ~ % python3 example.py
Instance ID: i-0ee6c946ce71f5107
Instance Type: t2.micro
State: running
--------------------
Instance ID: i-0bacdfde60c3cb23d
Instance Type: t2.micro
State: running
--------------------
willismt@3c22fbb201a5 ~ %
```


Unmanaged services
Customer responsibility
Customer data
Client-side data encryption
Server-side encryption
Network traffic protection
Platform and application management
OS, network, firewall configuration
AWS responsibility
Software for compute, storage, database, and networking
Hardware, AWS Global Infrastructure

Customer and AWS responsibilities in the AWS Shared Responsibility Model.

## #6 Scaling Amazon EC2

Scalability - System ko permanently ~~badha~~ bada karna

Elasticity - Load badhe to resource ~~bad~~ badhao, kam to hata do.

scale out → Horizontal Scaling → Adding more resources to pool

scaling up → vertical → Making existing instances more powerful.

Amazon cloudwatch ⇒ collecting and monitoring data about instances

## #7 Elastic Load Balancer (ELB)

ELB automatically distributes incoming applicati-on traffic across multiple resources, such as EC2 instances to optimize performance and reliability.

Routing Methods

① Round Robin → Evenly distribute to servers in cyclic manner

(ii) Least connections - Routes traffic to server with the fewest active connections.

(iii) IP Hash → Uses client's IP address to consistently route traffic to same server

(iv) Least Response Time - Directs to the server which have fastest response time to minimize latency.

#8 Messaging and ~~Queing~~ Queuing
Monolithic
Tightly coupled Architecture - Direct dependency
Loosely coupled → Independent components.
↳ Microservices.

Amazon Simple Queue Service (SQS
→ Send, store and Receive Messages
Payload → Data within a message
Queue → Messages are placed till process

Amazon Simple Notification service. (SNS)
↳ A channel for Messages to be delivered

Amazon Eventbridge - Serverless service that connects different parts of application using events.