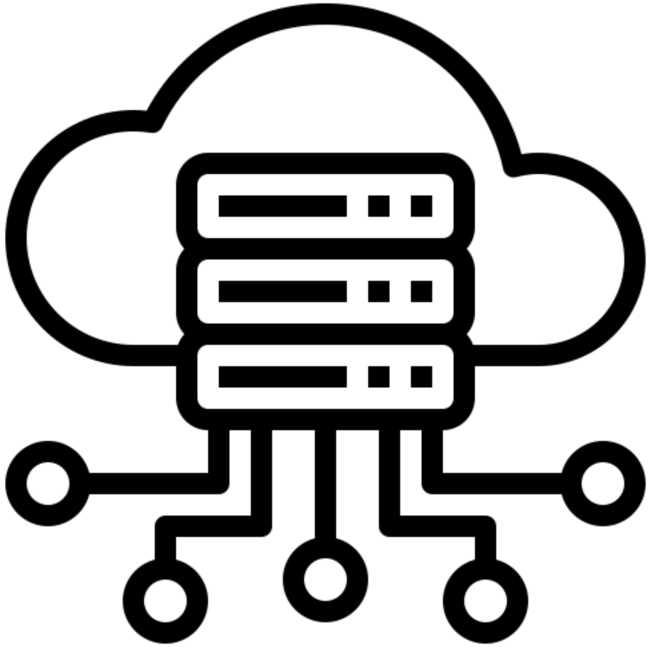
CLOUD COMPUTING

AWS/Azure

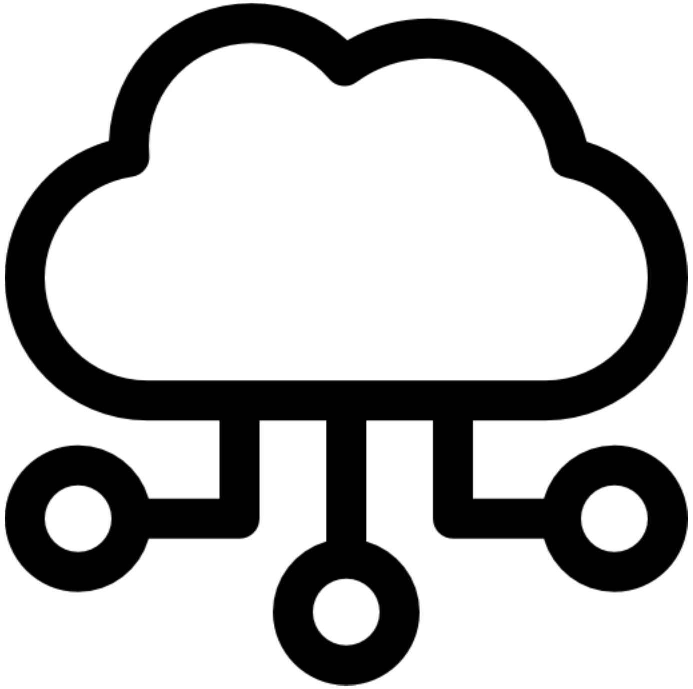
What is Cloud Computing?

- **AWS (Amazon Web Services)** and **Azure (Microsoft Azure)** fall under the domain of cloud computing.
- **Cloud computing** is the delivery of computing services—including servers, storage, databases, networking, software, and analytics—over the internet ("the cloud").
- Key Benefits of Cloud computing:
  - Offers faster innovation
  - flexible resources
  - economies of scale, etc.



Key Characteristics of Cloud Computing

- **On-Demand Self-Service:**
  - Users can provision resources as needed without human intervention.
- **Broad Network Access:**
  - Services are available over the network and accessed through standard mechanisms.
- **Resource Pooling:**
  - Provider's computing resources are pooled to serve multiple consumers using a multi-tenant model.
- **Rapid Elasticity:**
  - Resources can be elastically provisioned and released to scale rapidly.
- **Measured Service:**
  - Resource usage can be monitored, controlled, and reported for transparency.



Cloud Service Models

- **Infrastructure as a Service (IaaS):**
  - Provides virtualized computing resources (including virtual machines, storage, and networking) over the internet.
  - Examples: AWS EC2, Azure Virtual Machines.

- **Platform as a Service (PaaS):**
  - Delivers hardware and software tools over the internet which offers a platform for developers to build, run, and manage applications without the complexity of maintaining the underlying infrastructure.
  - Examples: AWS Elastic Beanstalk, Azure App Services.

- **Software as a Service (SaaS):**
  - Delivers software applications over the internet on a subscription basis eliminating the need for local installation and maintenance.
  - Examples: Google Workspace, Microsoft Office 365.

Benefits of Cloud Computing

- **Cost Efficiency:**
  - Reduces capital expenditure on hardware and software.
- **Scalability:**
  - Easily scales resources up or down based on demand.
- **Business Continuity:**
  - Ensures data backup and disaster recovery.
- **Collaboration Efficiency:**
  - Allows teams to collaborate from different locations.
- **Automatic Updates:**
  - Providers perform regular software updates.

Challenges of Cloud Computing

- **Security and Privacy Concerns**
  - Data breaches and unauthorized access risks.
  - Ensuring data encryption and protection of sensitive information.
- **Compliance Issues**
  - Difficulty in meeting industry-specific regulations (e.g., GDPR, HIPAA).
  - Complexities in managing data across different legal jurisdictions.
- **Dependency on Internet Connectivity**
  - Service accessibility is reliant on stable internet connections.
  - Risk of significant disruptions during outages or slowdowns.
- **Potential Vendor Lock-In**
  - Challenges in migrating data and applications between providers.
  - Limited flexibility due to proprietary technologies.
- **Performance Variability**
  - Latency issues impacting real-time application performance.
  - Inconsistent resource availability affecting business operations.

Key Players in Cloud Computing

- Amazon Web Services (AWS):
  - Market leader with a wide range of services.
- Microsoft Azure:
  - Strong integration with Microsoft products.
- Google Cloud Platform (GCP):
  - Known for its data analytics and machine learning capabilities.
- IBM Cloud:
  - Strong in hybrid cloud and AI.
- Oracle Cloud:
  - Specializes in enterprise applications.

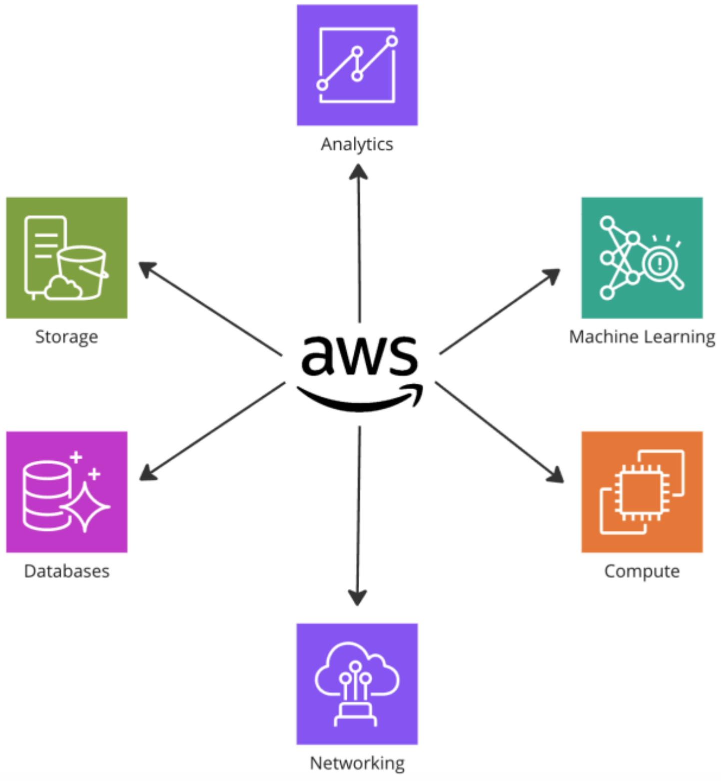
GREENBOOTCAMPS

AWS - Amazon Web Services

- Launched in 2006, AWS is a comprehensive cloud computing platform provided by Amazon.

AWS Key Services:

- **Compute:** EC2 (Elastic Compute Cloud), Lambda (Serverless Computing), etc
- **Storage:** S3 (Simple Storage Service), EBS (Elastic Block Store), etc
- **Database:** RDS (Relational Database Service), DynamoDB (NoSQL Database) , etc
- **Networking:** VPC (Virtual Private Cloud), Route 53 (DNS Service) , etc



Amazon Web Services (AWS) Overview

- **Global Infrastructure**
  - AWS boasts an extensive network of data centers across multiple geographic regions, ensuring low-latency access and high availability for users worldwide.
- **Comprehensive Service Portfolio**
  - Offering over 200 fully-featured services, AWS caters to a wide range of computing needs, from basic storage and compute to advanced machine learning and IoT solutions.
- **Pay-as-you-go Model**
  - AWS implements a flexible pricing structure, allowing customers to pay only for the resources they use, without upfront commitments or long-term contracts.
- **Security and Compliance**
  - With a shared responsibility model, AWS provides robust security measures and compliance certifications, helping organizations meet various regulatory requirements.

AWS Core Services: EC2 and S3

Service	Description	Key Features
Amazon EC2 (Elastic Compute Cloud)	Scalable virtual servers in the cloud	Multiple instance types, auto-scaling, load balancing
Amazon S3 (Simple Storage Service)	Object storage service for any amount of data	Durability, availability, scalability, security

AWS infrastructure and deployment

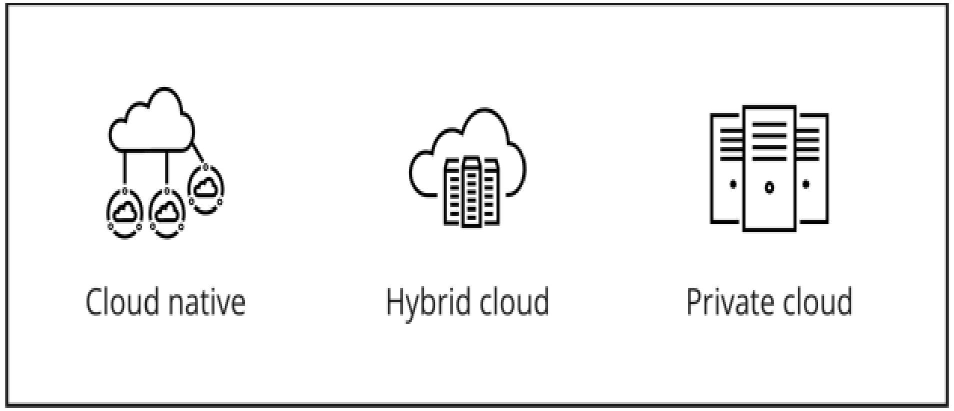


Fig: Deployment options

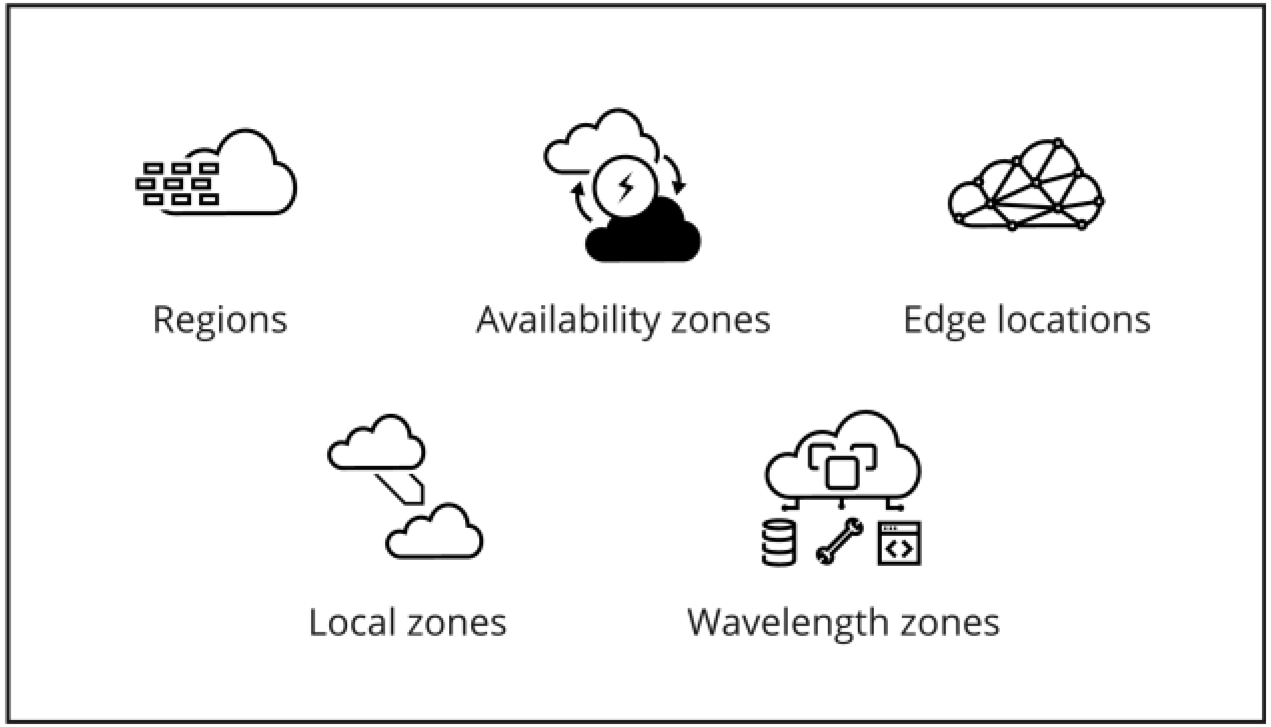
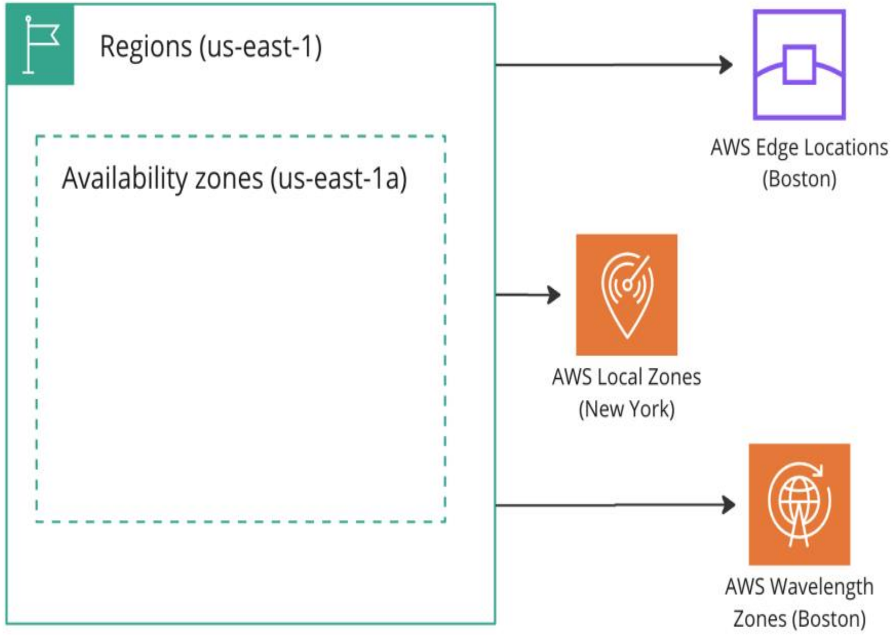


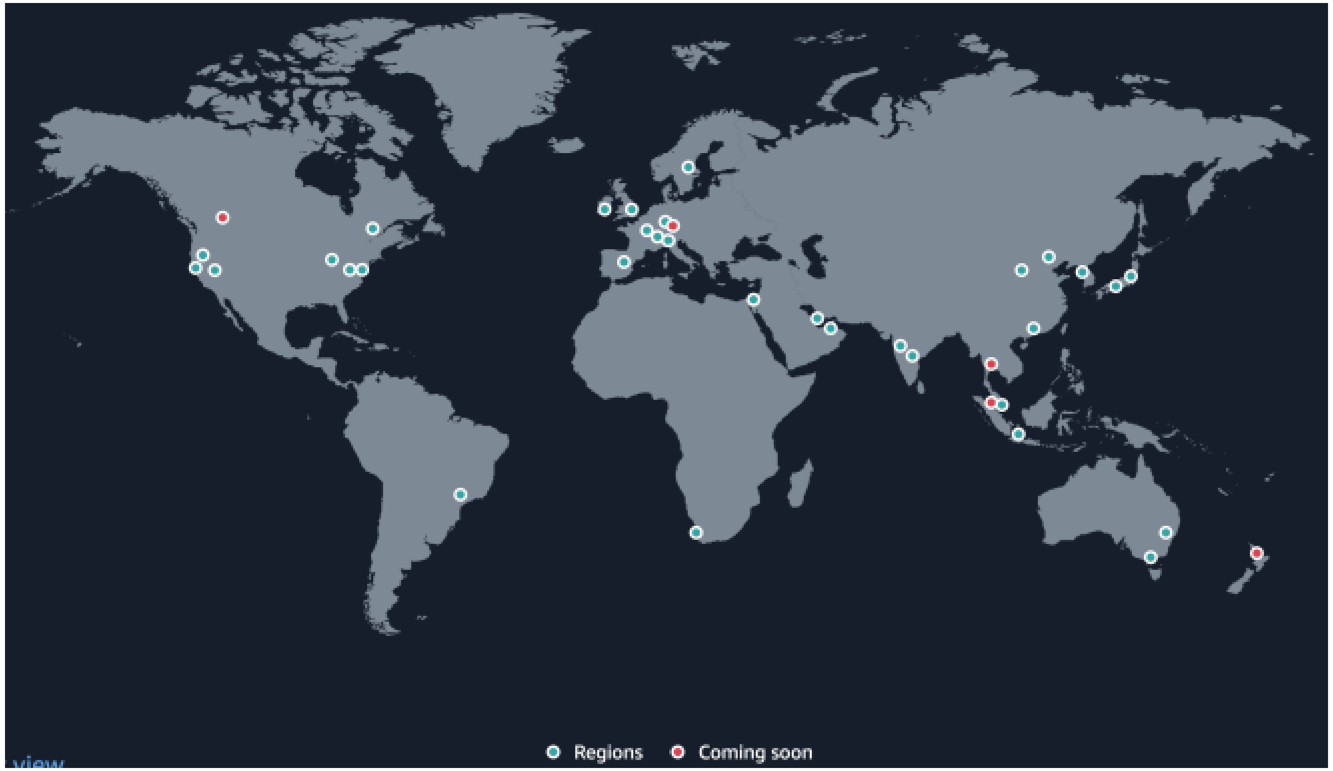
Fig: AWS global infrastructure

AWS global infrastructure

- Worldwide accessibility and low-latency connections
- Global network of regions and availability zones


Regions (us-east-1)
Availability zones (us-east-1a)
AWS Edge Locations (Boston)
AWS Local Zones (New York)
AWS Wavelength Zones (Boston)

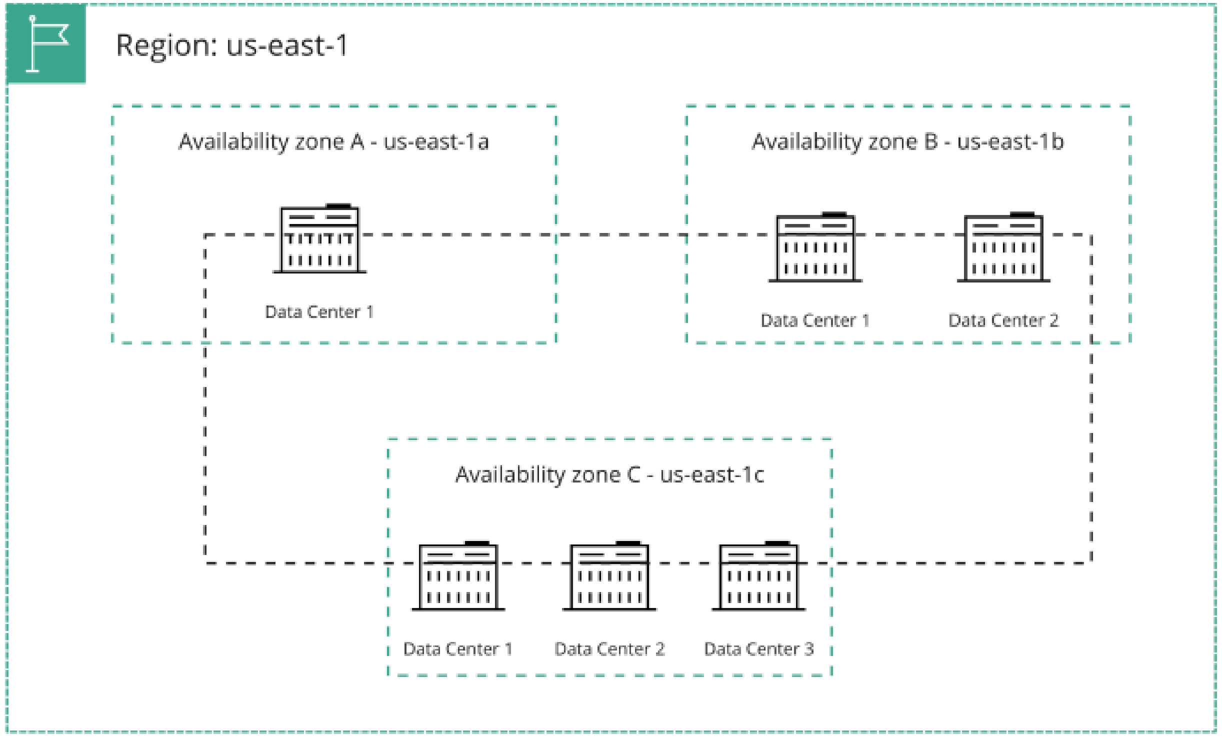
Introduction to AWS regions



AWS availability zones

Availability zones live within AWS regions

- Physically separated with independent power, cooling, and networking
- Designed for fault tolerance and high availability
- Provide redundancy and isolation, even during localized failures

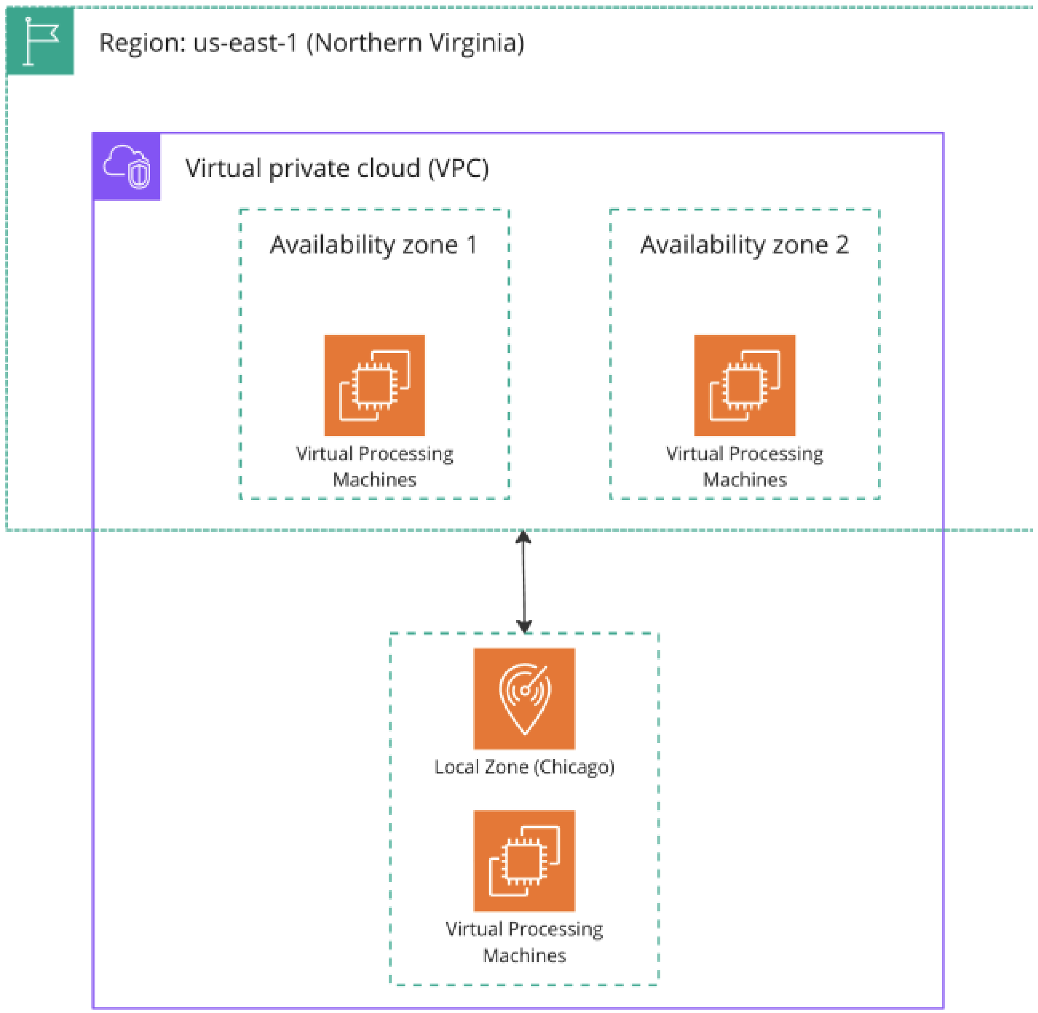


Region: us-east-1
Availability zone A - us-east-1a
Data Center 1
Availability zone B - us-east-1b
Data Center 1
Data Center 2
Availability zone C - us-east-1c
Data Center 1
Data Center 2
Data Center 3

AWS local zones

Extension of AWS region

- Brings compute, storage, and databases closer to users
- Low-latency access in specific geographic areas beyond standard regions

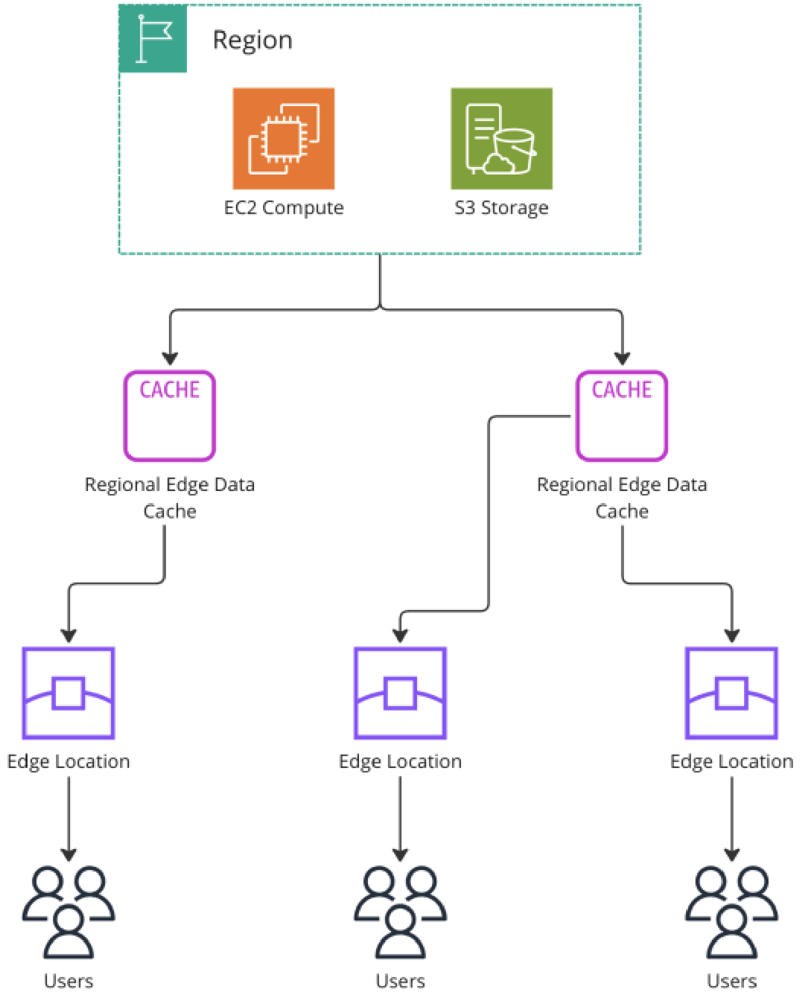

Region: us-east-1 (Northern Virginia)
Virtual private cloud (VPC)
Availability zone 1
Virtual Processing Machines
Availability zone 2
Virtual Processing Machines
Local Zone (Chicago)
Virtual Processing Machines

Edge Locations

Global data centers that utilize local zones for faster access to data

- Enhance content delivery by storing cached data in your local zone
- Reduce latency for end-users
- Faster access to data

AWS wavelength zones: Use edge locations to power telecommunication networks

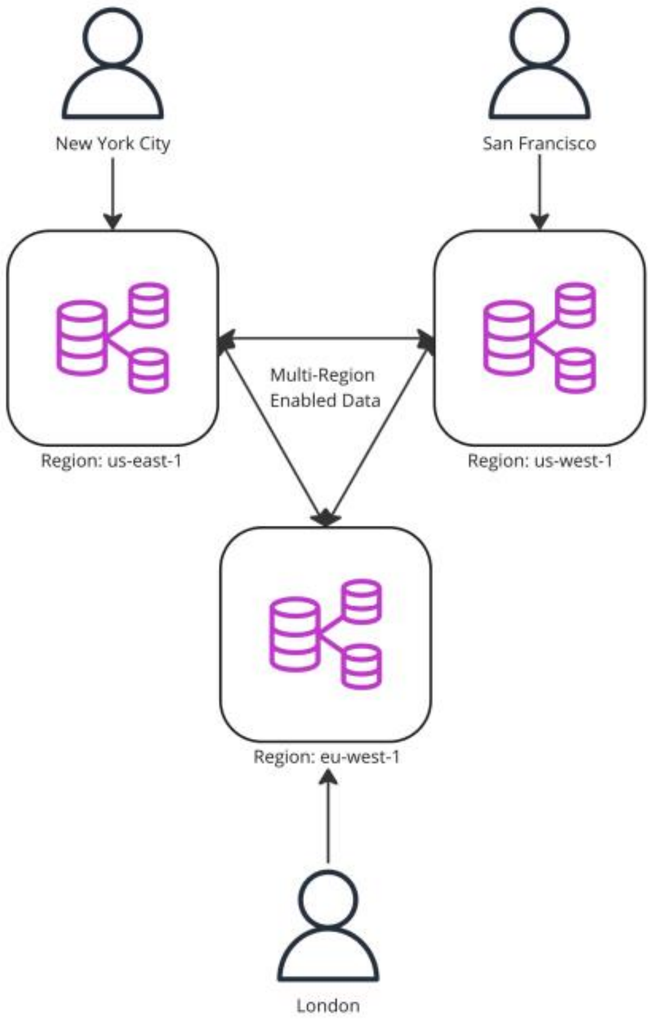


Multi-region deployment in AWS

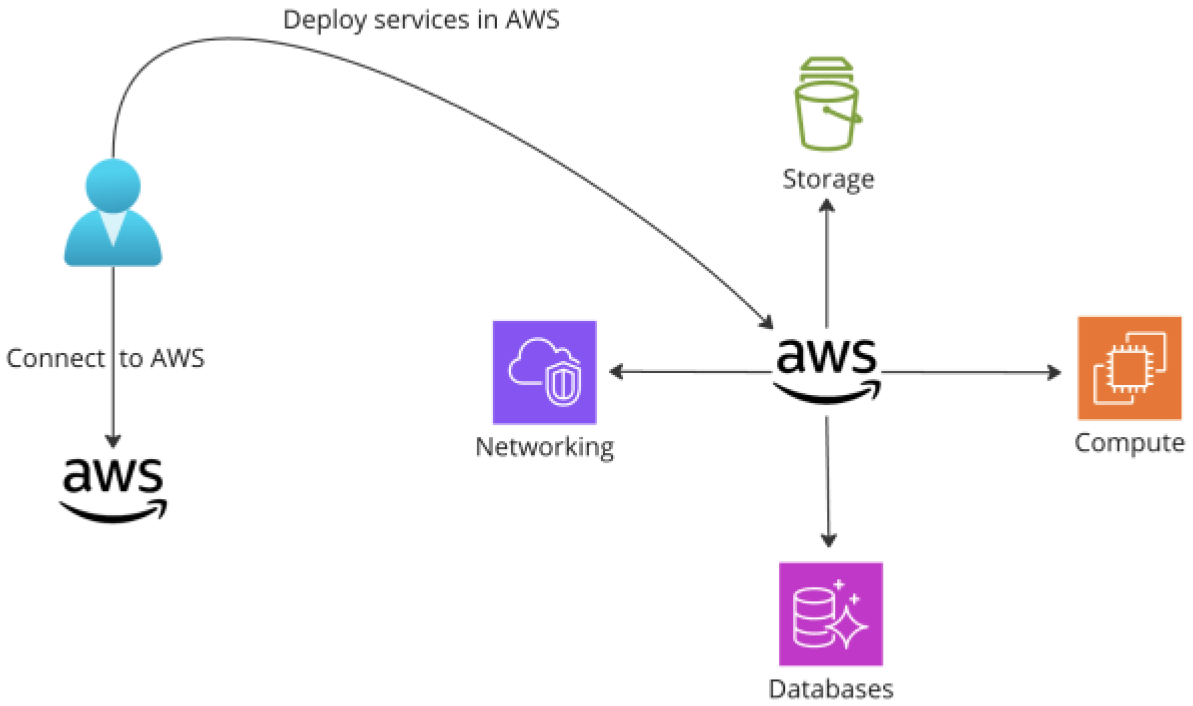
Deploying applications to multiple AWS regions worldwide

Advantages

- Mitigate regional failures
- Optimize latency for global users



Connect and deploy to AWS



Connectivity options in AWS

Three distinct connectivity choices based on usage

Public Internet

AWS Direct Connect

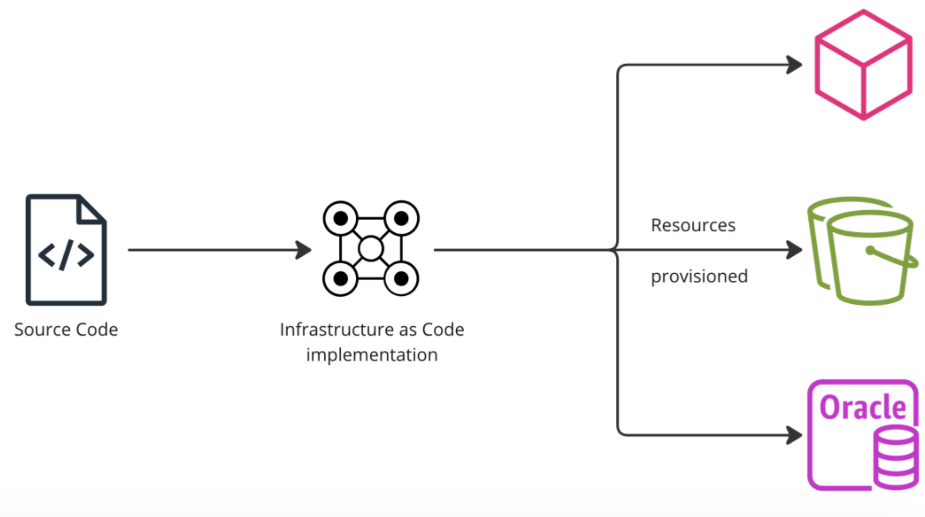
AWS VPN

Infrastructure as Code (IaC)

Infrastructure as Code (IaC) is a method to provision and manage infrastructure using code and templates

Why we should use IaC?

- Version Controlling
- Code based configurations
- Consistency and reproducibility



AWS CloudFormation

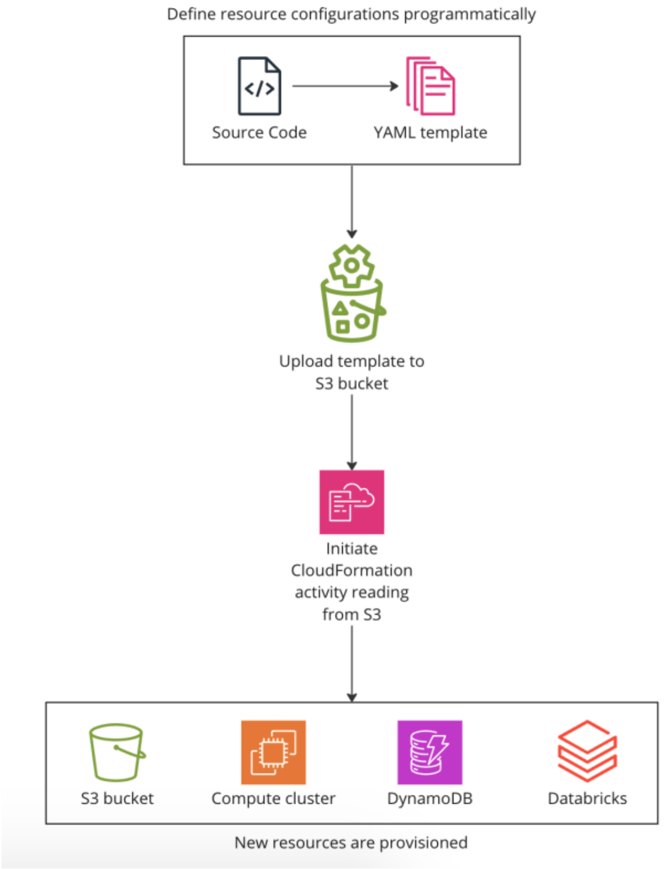
Enables you to define and provision AWS infrastructure as code

Key Features

- Declarative templates using JSON or YAML files
- Version control
- Continuous Integration/Continuous Deployment (CI/CD)

Method for Enabling IaC:

- Author a YAML template defining a stack of resources needed with their configurations
- Upload the template to an S3 bucket
- Initiate CloudFormation activity
- CloudFormation provisions the resources in the defined configurations


Define resource configurations programmatically
Source Code
YAML template
Upload template to S3 bucket
Initiate CloudFormation activity reading from S3
S3 bucket
Compute cluster
DynamoDB
Databricks
New resources are provisioned

Storage types in AWS

Object Storage:

Storage architecture that manages and organizes data as discrete units called "objects"

Key characteristics:

- Horizontal scaling
- Metadata management
- Storing unstructured data

Amazon S3:

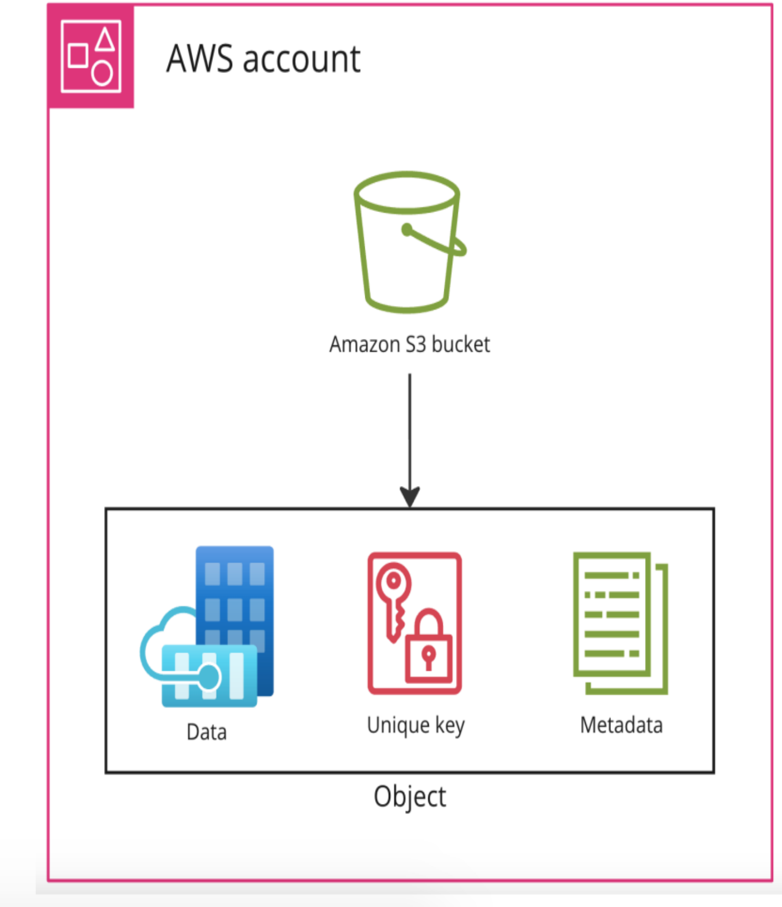
A highly scalable and durable object storage

service offered by AWS

Key characteristics:

- Available in all AWS regions
- Classes: S3 standard,

  S3 intelligent tiering, S3 One zone IA, S3 glacier, S3 glacier deep archive, S3 on Outposts



Storage Classes

Amazon S3 Standard	Amazon S3 Intelligent tiering	Amazon S3 One zone IA	Amazon S3 Glacier	Amazon S3 Glacier Deep Archive	Amazon S3 on Outposts
Durable, Scalable and available	Automatic cost optimization	Cost-effective, single availability zone	Low cost, archival storage	Lowest cost, longest retrieval time	Combine private and public cloud data
Frequent access	Moves objects between tiers based on changing access patterns	Infrequent access	Long-term archival with retrieval times ranging from minutes to hours		

Block storage

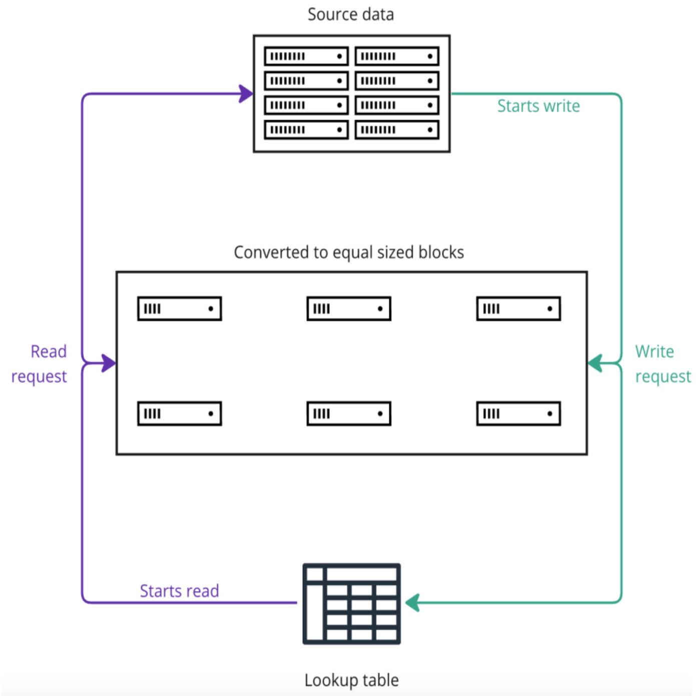
Block Storage:

Divides data into fixed-sized blocks, each with

its unique address

Key characteristics:

- Running I/O intensive transactional web applications
- Right-size big data analytics engines



Amazon EBS:

A scalable, high-performance block storage service designed for use with Amazon compute services

Key characteristics:

- Run applications with 99.99% availability

File storage

File Storage:

Organizes and stores data in a hierarchical structure

Key characteristics:

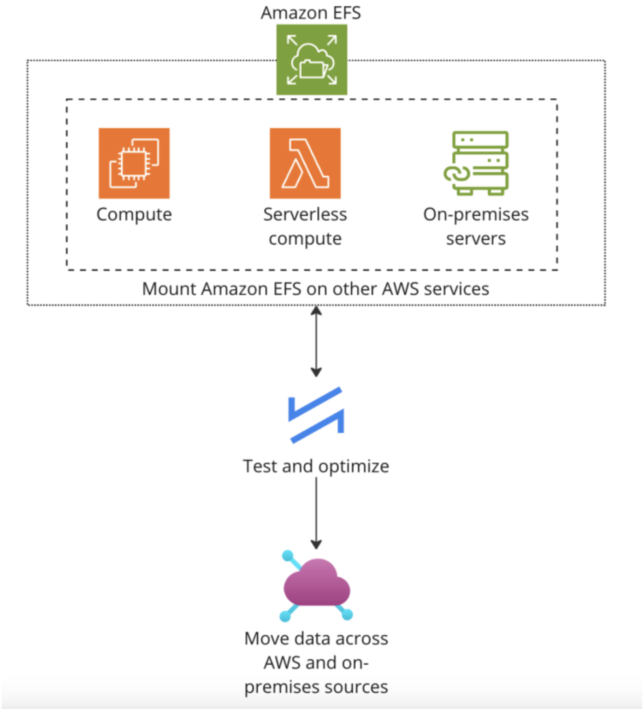
- Allows multiple concurrent reads and writes across users and services
- Stores metadata about files

Amazon EFS:

File storage service designed for use with AWS cloud services and on-premises resources

Use cases

- Simplify DevOps
- Enhance content management systems
- Accelerate data science



Amazon EFS
Compute
Serverless compute
On-premises servers
Mount Amazon EFS on other AWS services
Test and optimize
Move data across AWS and on-premises sources

Cache storage

Cache Storage:

Storing frequently accessed data in a quickly retrievable location

Key characteristics:

- Accelerates application response times by reducing data retrieval latency
- Minimizes the load on backend servers

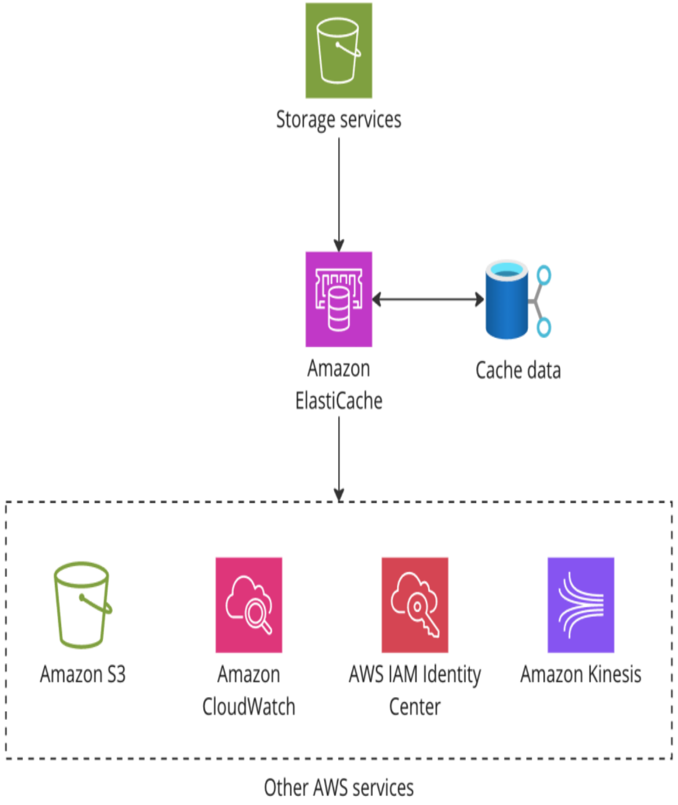
Amazon ElastiCache:

- Caching service that enables seamless, high speed access to frequently used data

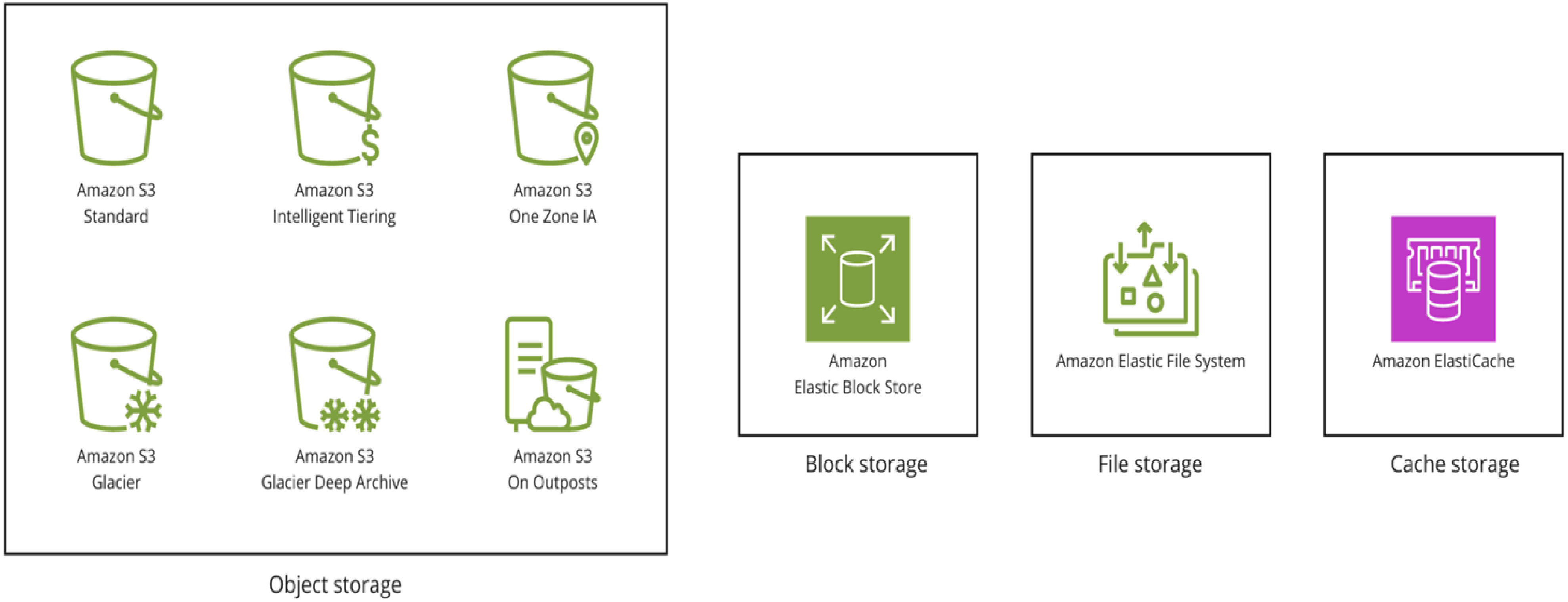
Use cases

- Store web application session data in memory
- Accelerates access to real-time analytics

data



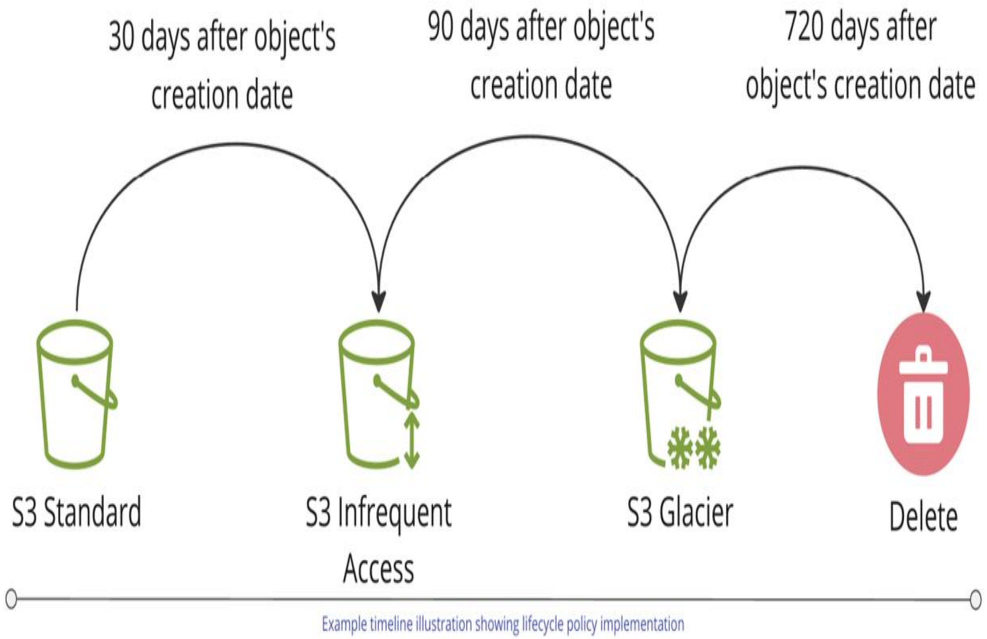
Revisiting the storage types



Storage Lifecycle Policy

Defines the transition of objects between storage classes in S3, based on predefined rules

- Cost and performance optimization
- Improves data management and compliance

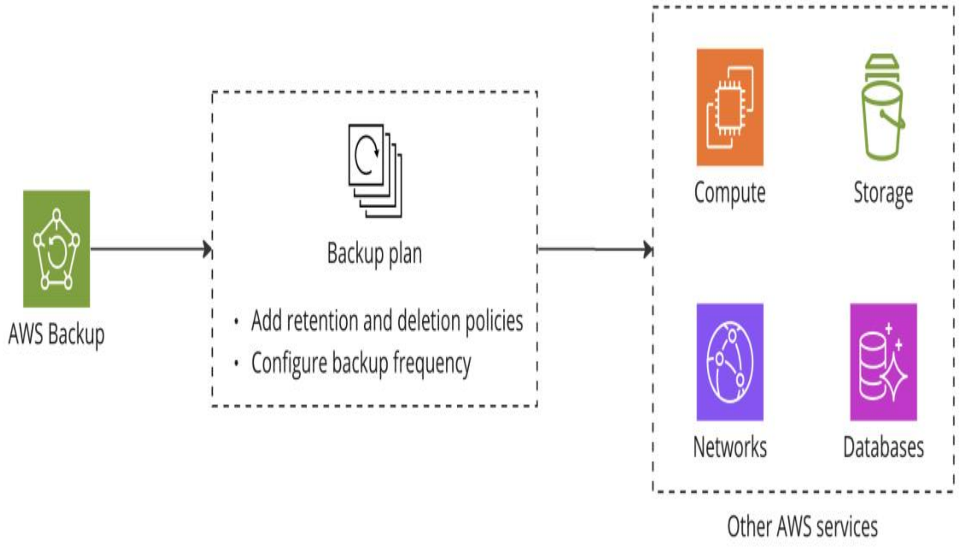

30 days after object's creation date
90 days after object's creation date
720 days after object's creation date
S3 Standard
S3 Infrequent Access
S3 Glacier
Delete

Example timeline illustration showing lifecycle policy implementation

AWS Backup

Cost-effective, fully managed service that centralizes and automates backup across AWS services

- Cross-region backups
- Set retention and deletion policies



AWS Compute Services

Why do we need compute services?

- Imagine it's Black Friday
- Your website crashes due to high traffic
- What do you do?

Importance: scalability, flexibility, and cost-efficiency

backbone of our digital solutions

Solution : Providing computing power on demand

Meeting the challenge with AWS

Server Based

- Continuous availability
- Dedicated resources
- More control
- Customization (like owning a car)

Serverless

- On-demand execution
- No server management
- Event-driven
- Cost-effective It's like using a taxi service; it's there when you need it and gone when you don't

EC2

- Virtual servers in the cloud
- Customizable configurations (OS, storage, location)
- Focus on customization

Lambda

- Serverless computing platform
- Name comes from Lambda calculus
- Event-driven architecture (file uploads, database changes)
- Focus on convenience

In real life

EC2

- Hosting websites
- Scalability and customization
- Big data analytics

Lambda

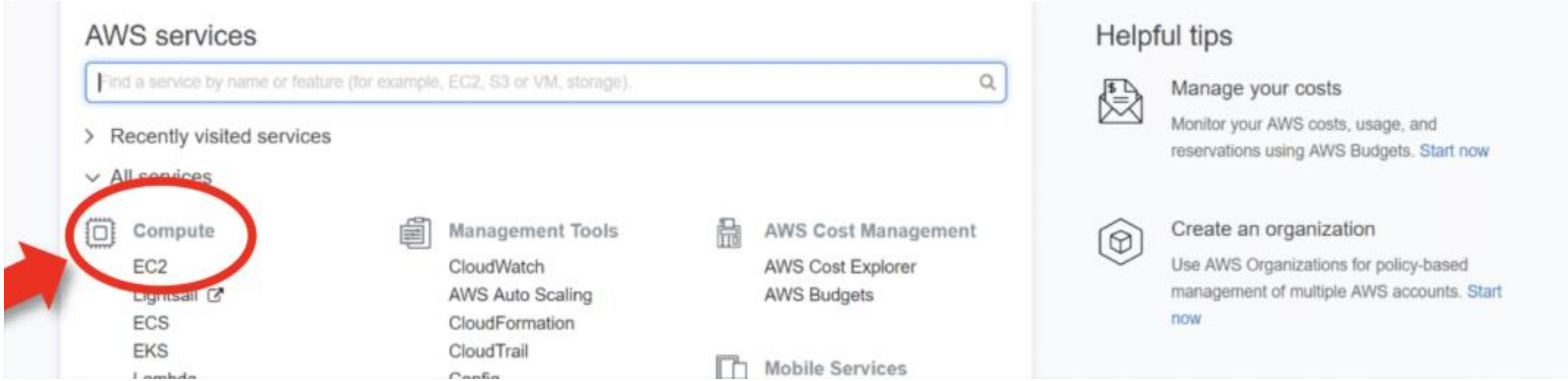
- Real-time image processing
- Event-driven tasks
- Automated backups

How to create an EC2 instance

Step 1: First, log into your AWS account and click on “services” present on the left of the AWS management console, i.e. the primary screen.

From the drop-down menu of options, tap on “EC2”. To create an AWS free tier account refer to Amazon Web Services (AWS) – Free Tier Account Set up.

Under Resources >> Click on “Instances running” — It will show if any EC2 instances are running or not.



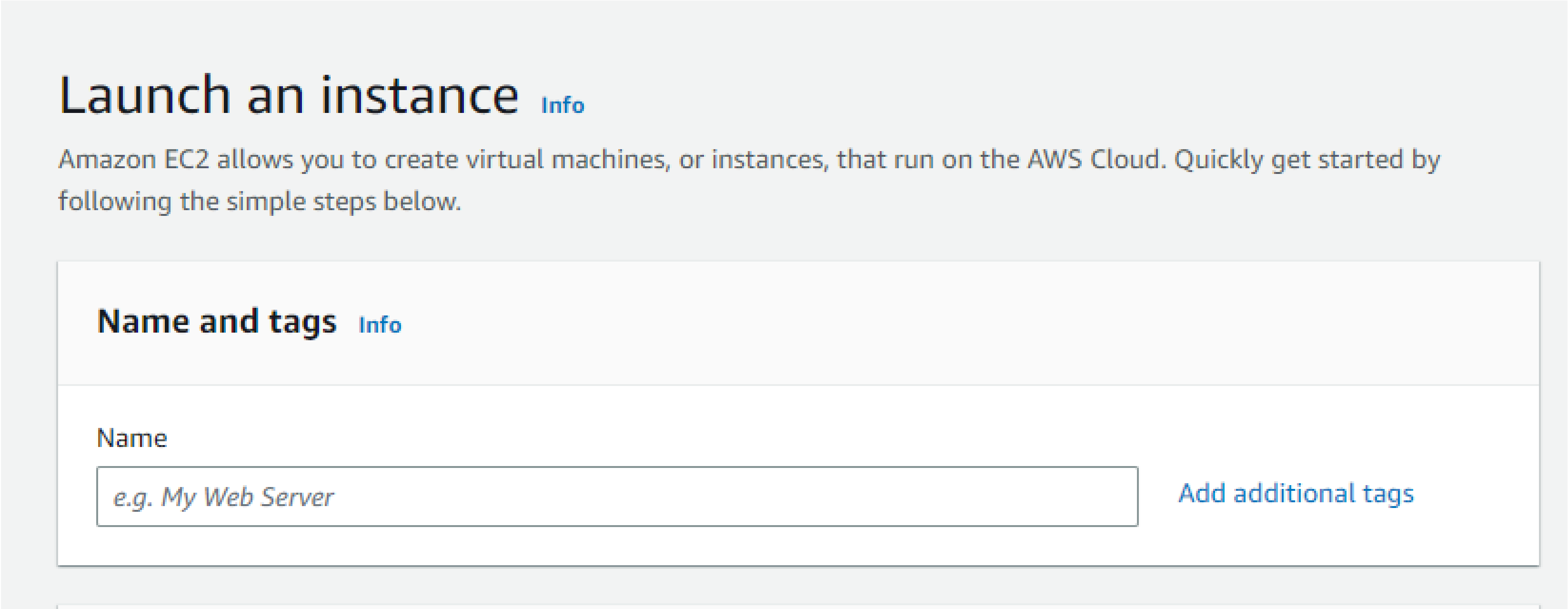
Monitor your AWS costs, usage, and reservations using AWS Budgets. Start now

Use AWS Organizations for policy-based management of multiple AWS accounts. Start now

How to create an EC2 instance

Step 2: Click on the launch instance click on the launch instance, after clicking on it you will be redirected to a launch page where we can create an instance.

Configure all the requirements to Create a new instance like the name of the instance as shown in the figure below.



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

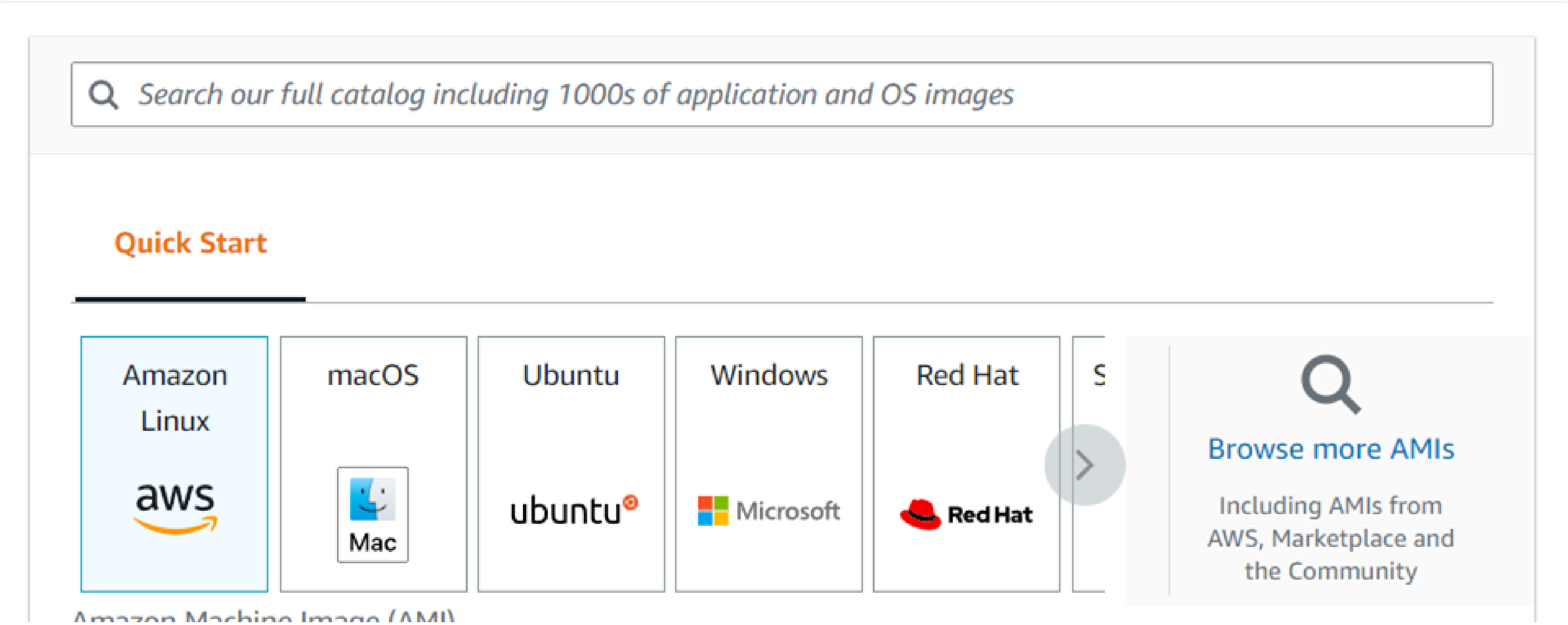
Name

e.g. My Web Server

GREENBOOTCAMPS

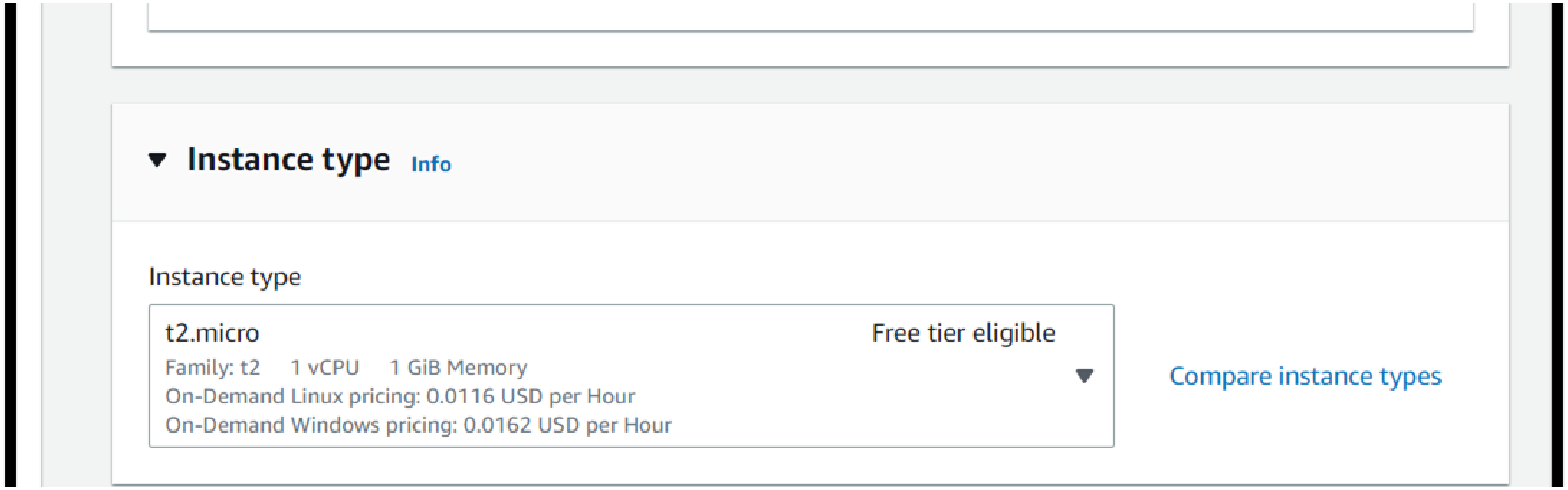
How to create an EC2 instance

Step 3: Select AMI – Required operating system from the available. There are different types of OS available select the OS as per your requirement.


Search our full catalog including 1000s of application and OS images
Quick Start
Amazon Linux
aws
macOS
Mac
Ubuntu
ubuntu
Windows
Microsoft
Red Hat
Red Hat
S
Browse more AMIs
Including AMIs from AWS, Marketplace and the Community
Amazon Machine Image (AMI)

How to create an EC2 instance

Step 4: By default, it selects a free tier of storage. (IF YOU ARE ELIGIBLE FOR THE FREE TIER). From the available storage specifications, select a free tier-eligible storage service. The instance type includes the no.of CPUs required and the Memory required for your application. By default, the instance type is “t2.micro” which is a free tier-eligible service. Do not select any other which leads to the billing amount. To know more about instance types refer to Amazon EC2 – Instance Types.



▼ Instance type Info

Instance type

t2.micro Free tier eligible ▼
Family: t2 1 vCPU 1 GiB Memory
On-Demand Linux pricing: 0.0116 USD per Hour
On-Demand Windows pricing: 0.0162 USD per Hour

Compare instance types

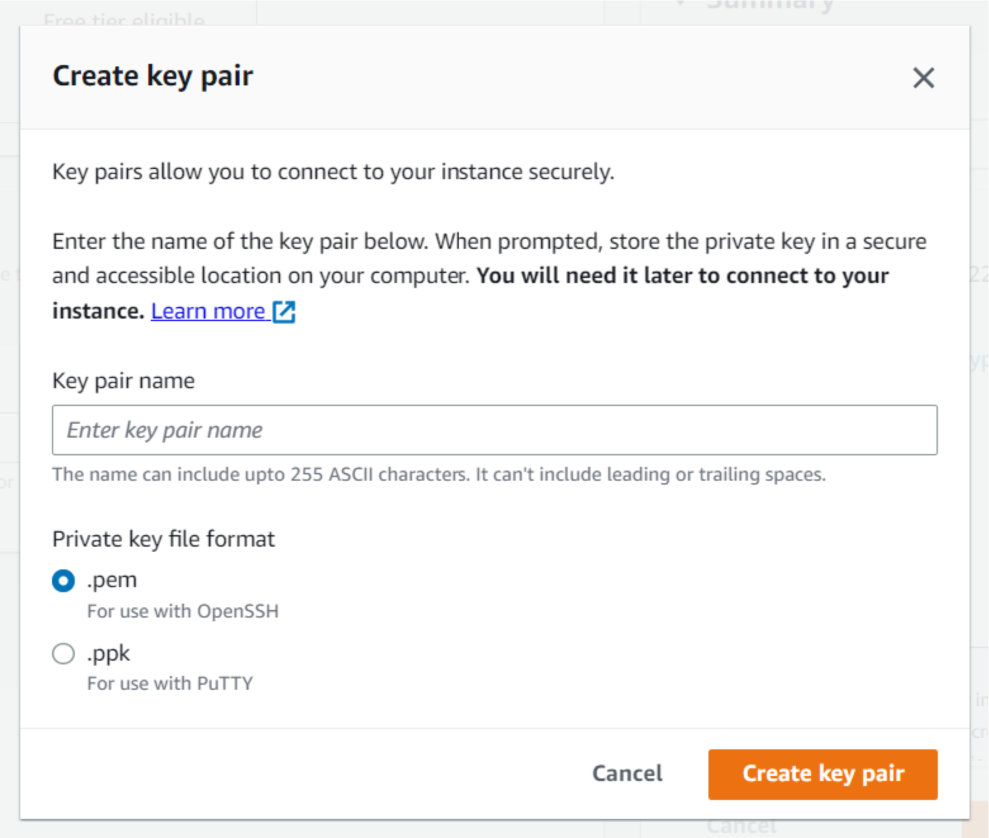
How to create an EC2 instance

Step 4: Now, create a key-value pair, by clicking on “Create new key pair”.

A window will pop up for creating key pair as shown below.

The key value pair plays a major role while connecting to the EC2-Instance it will act as an SSH-Key to connect to the instance.

Create Key-PairEnter name>>Select “.pem” and create. Automatically key pair which was created will be downloaded. Select the created key pair.



Create key pair

Key pairs allow you to connect to your instance securely.

Enter the name of the key pair below. 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

Key pair name

Enter key pair name

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

Private key file format

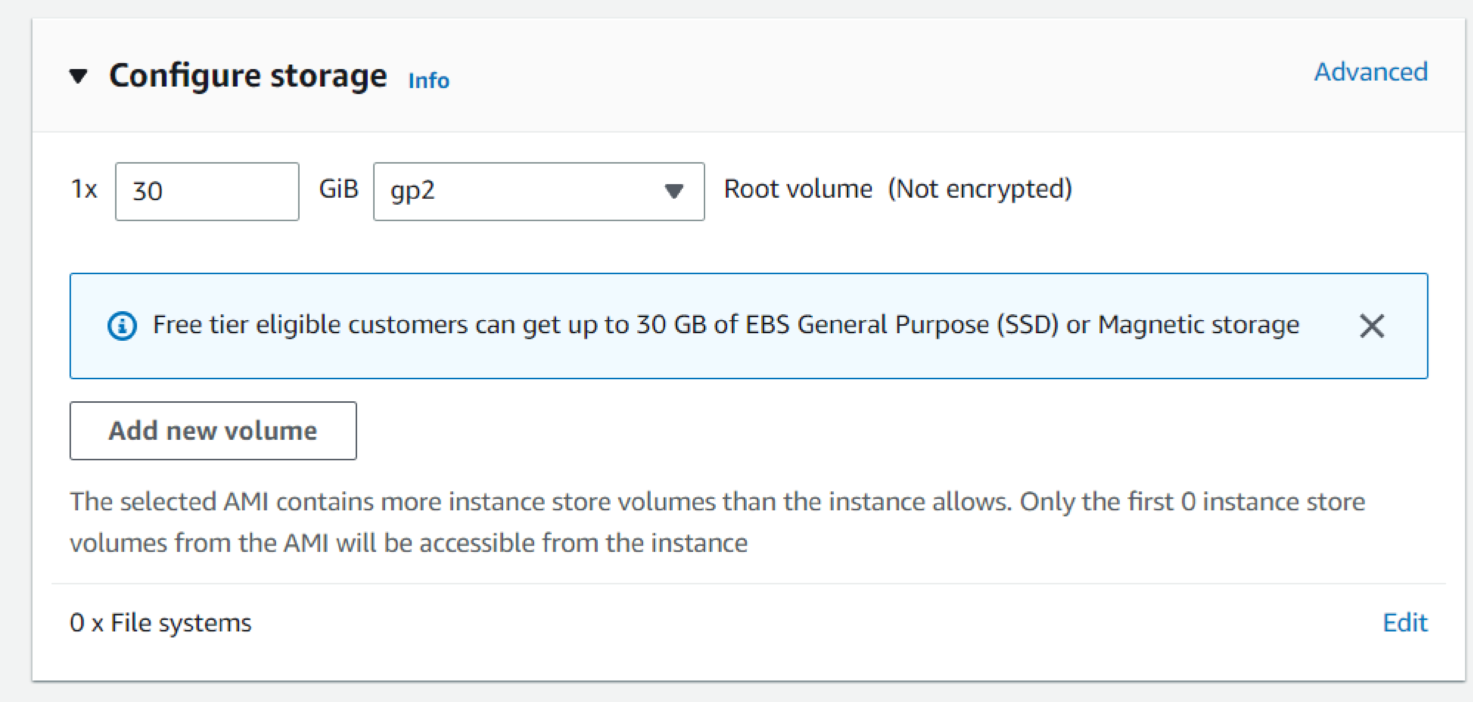
● .pem
For use with OpenSSH

○ .ppk
For use with PuTTY

How to create an EC2 instance

Step 5: Keep the network settings as default settings and make changes if required.

Storage As mentioned in the picture, Free tier eligible can get up to 30 GB of EBS Storage. Keep it as default.



▼ Configure storage Info Advanced

1x 30 GiB gp2 ▼ Root volume (Not encrypted)

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

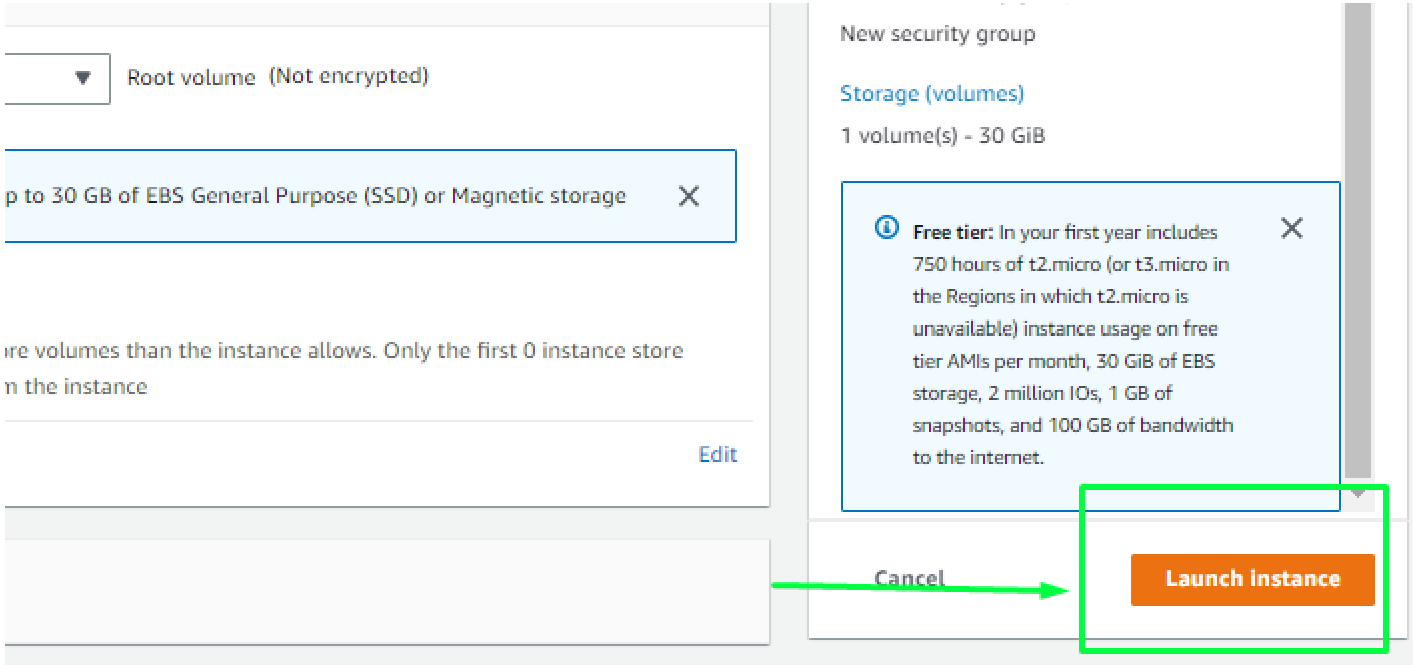
Add new volume

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

0 x File systems Edit

How to create an EC2 instance

Step 6: Launching Instance At last, Check if all the selected are eligible for a free tier or not and click on “Launch instance”.That’s it, an instance will be created.



Root volume (Not encrypted)

p to 30 GB of EBS General Purpose (SSD) or Magnetic storage

ıre volumes than the instance allows. Only the first 0 instance store
n the instance

New security group

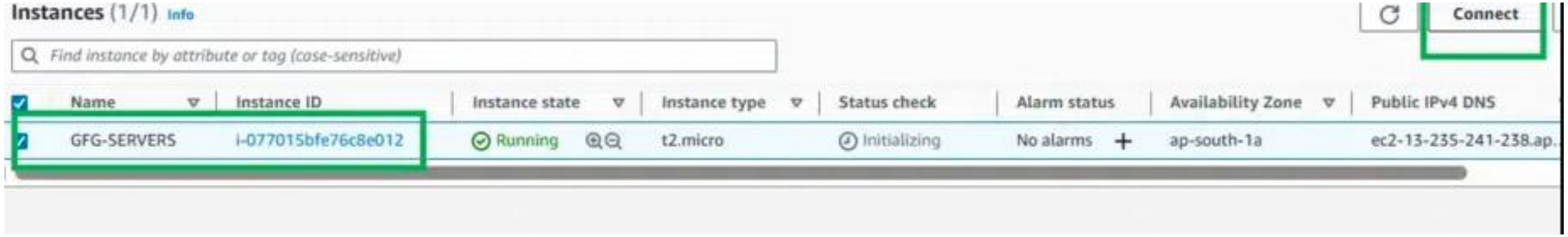
Storage (volumes)

1 volume(s) - 30 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 IOs, 1 GB of snapshots, and 100 GB of bandwidth to the internet.

Steps To Connect Terminal Using SSH-Key

Step 1: Select the server to which you want to connect and click on the connect button at the top of that instance as shown in the image below.

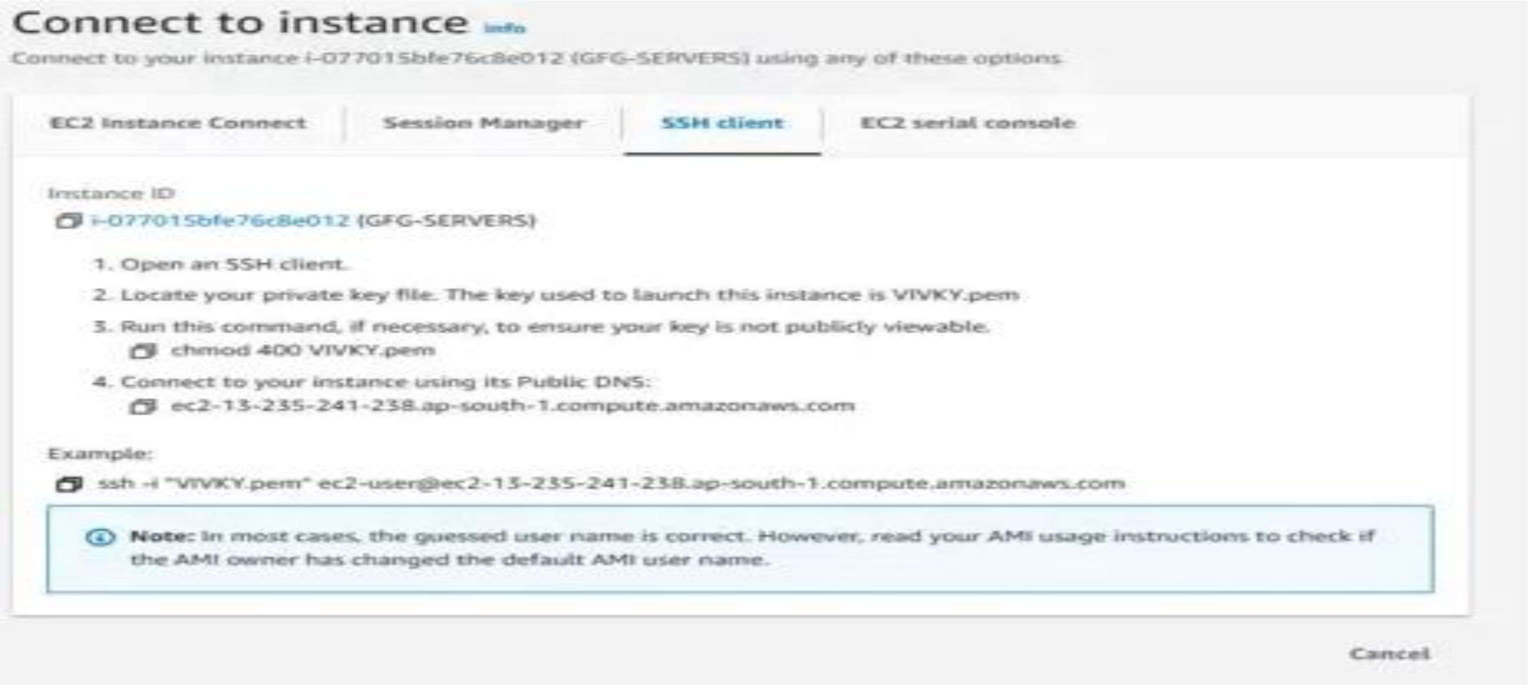


Instances (1/1) Info

☑	Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone	Public IPv4 DNS
☑	GFG-SERVERS	i-077015bfe76c8e012	Running	t2.micro	Initializing	No alarms	ap-south-1a	ec2-13-235-241-238.ap.

Steps To Connect Terminal Using SSH-Key

Step 2: Copy the SSH key which is right following the example it will acct as a key-pair to connect to EC2-Instance.



Connect to instance

Connect to your instance i-077015bfe76c8e012 (GFG-SERVERS) using any of these options

Instance ID

i-077015bfe76c8e012 (GFG-SERVERS)

1. Open an SSH client.
2. Locate your private key file. The key used to launch this instance is VIVKY.pem
3. Run this command, if necessary, to ensure your key is not publicly viewable.
   ```
   chmod 400 VIVKY.pem
   ```
4. Connect to your instance using its Public DNS:
   ```
   ec2-13-235-241-238.ap-south-1.compute.amazonaws.com
   ```

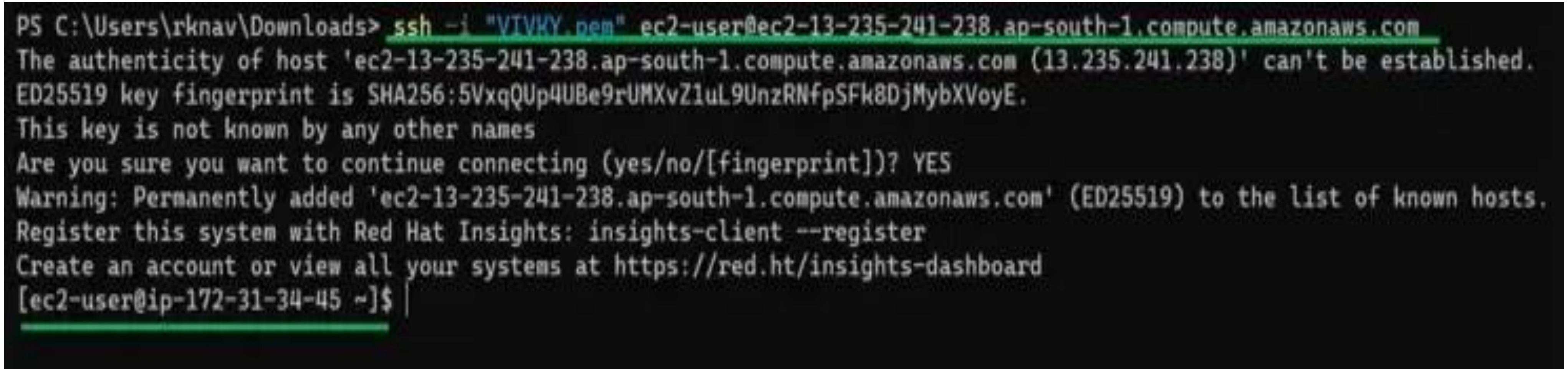
Example:

```
ssh -i "VIVKY.pem" ec2-user@ec2-13-235-241-238.ap-south-1.compute.amazonaws.com
```

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

Steps To Connect Terminal Using SSH-Key

Step 3: Open the terminal and go to the folder where your .pem file is located and paste the key that you have copied in AWS and paste it in the terminal.



```
PS C:\Users\rknav\Downloads> ssh -i "VIVKY.pem" ec2-user@ec2-13-235-241-238.ap-south-1.compute.amazonaws.com
The authenticity of host 'ec2-13-235-241-238.ap-south-1.compute.amazonaws.com (13.235.241.238)' can't be established.
ED25519 key fingerprint is SHA256:5VxqQUp4UBe9rUMXvZ1uL9UnzRNfpSFk8DjMybXVoyE.
This key is not known by any other names
Are you sure you want to continue connecting (yes/no/[fingerprint])? YES
Warning: Permanently added 'ec2-13-235-241-238.ap-south-1.compute.amazonaws.com' (ED25519) to the list of known hosts.
Register this system with Red Hat Insights: insights-client --register
Create an account or view all your systems at https://red.ht/insights-dashboard
[ec2-user@ip-172-31-34-45 ~]$ |
```

References: https://www.geeksforgeeks.org/amazon-ec2-creating-an-elastic-cloud-compute-instance/

More on storage types

Active Storage (Direct Storage)

- Like your recent emails, readily accessible
- Ideal for day-to-day operations
- AWS S3: designed for ease of access and management

Archival Storage

- Like old emails, accessed infrequently
- Ideal for long-term data retention
- AWS Glacier: cost-effective for long-term
- storage

Diving into S3

- S3 stands for Simple Storage Service
- Highly scalable, durable, and secure
- Wide variety of use cases like website hosting, data backup, and content distribution

Glacier

- Designed for long-term storage
- Cost-effective solution for data archiving and backup

In practice

S3

- Hosting static websites
- Real-time big data analytics

Glacier

- Historical data archiving
- Long-term backups

Understanding database types

Relational Databases (RDS)

- Like a well-organized bookshelf
- Structured data with clear relations
- Ideal for traditional applications
- AWS RDS: the sturdy bookshelf of the digital world

NoSQL Databases (DynamoDB)

- Like a dynamic magazine rack
- Flexible schema for unstructured data
- Ideal for mobile apps, IoT, gaming
- AWS DynamoDB: adaptable and ready for ever-changing content

DynamoDB

- Designed for web-scale applications
- Provides single-digit millisecond latency
- ideal for mobile, web , gaming and IoT applications

Understanding DynamoDB's key-value pairs

- DynamoDB uses a key-value model
- A key maps to a value
- This key represents the "key" and the safety deposit box represents a "value"

In practice

RDS

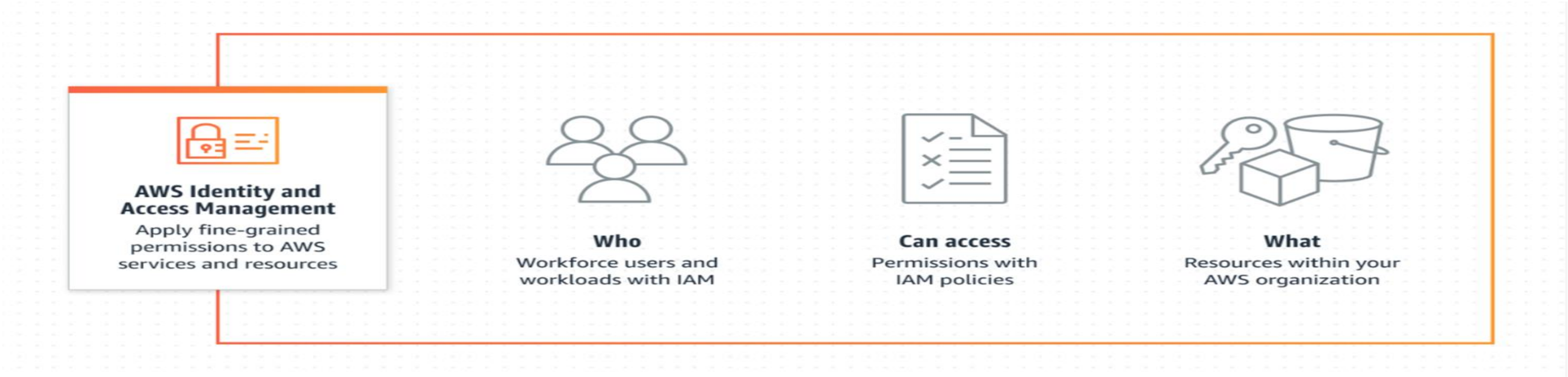
- Financial systems
- E-commerce platforms

DynamoDB

- Real-time bidding systems
- Leaderboards for gaming

Introduction to IAM

- IAM = Identity and Access Management
- Acts as a gatekeeper
- Only authenticated users are allowed in
- Ensures authenticated users are authorized



AWS Identity and Access Management
Apply fine-grained permissions to AWS services and resources

Who
Workforce users and workloads with IAM

Can access
Permissions with IAM policies

What
Resources within your AWS organization

Introducing KMS

- KMS = Key Management Service
- High-security vault
- Create and manage cryptographic keys
- Safeguarding information

- Very secure safe
  - You control who accesses it
- Sensitive customer data that needs encryption
- Master key encrypts data
- People with permissions access this key

AWS Shield and AWS compliance

AWS Shield

- Manages who's accessing your applications
- Keeps malicious traffic away
- AWS Shield Standard
  - Common DDoS attacks
- AWS Shield Advanced
  - Mitigation capabilities
  - 24/7 Response Team

AWS compliance

- Follows laws of digital land Helps with regulatory requirements
  - HIPAA for healthcare
  - GDPR for data protection in Europe
- Provides resources/documentation for data compliance

AWS pricing overview

Pay-as-you-go

- Only pay for services you consume
- No long-term contracts or complex licensing
- Automatically scale services and your costs with your workload

Savings plans

- Like a gym membership
- Save money on our usage
- Reduce cost by modernizing workloads
- Centralize cost management

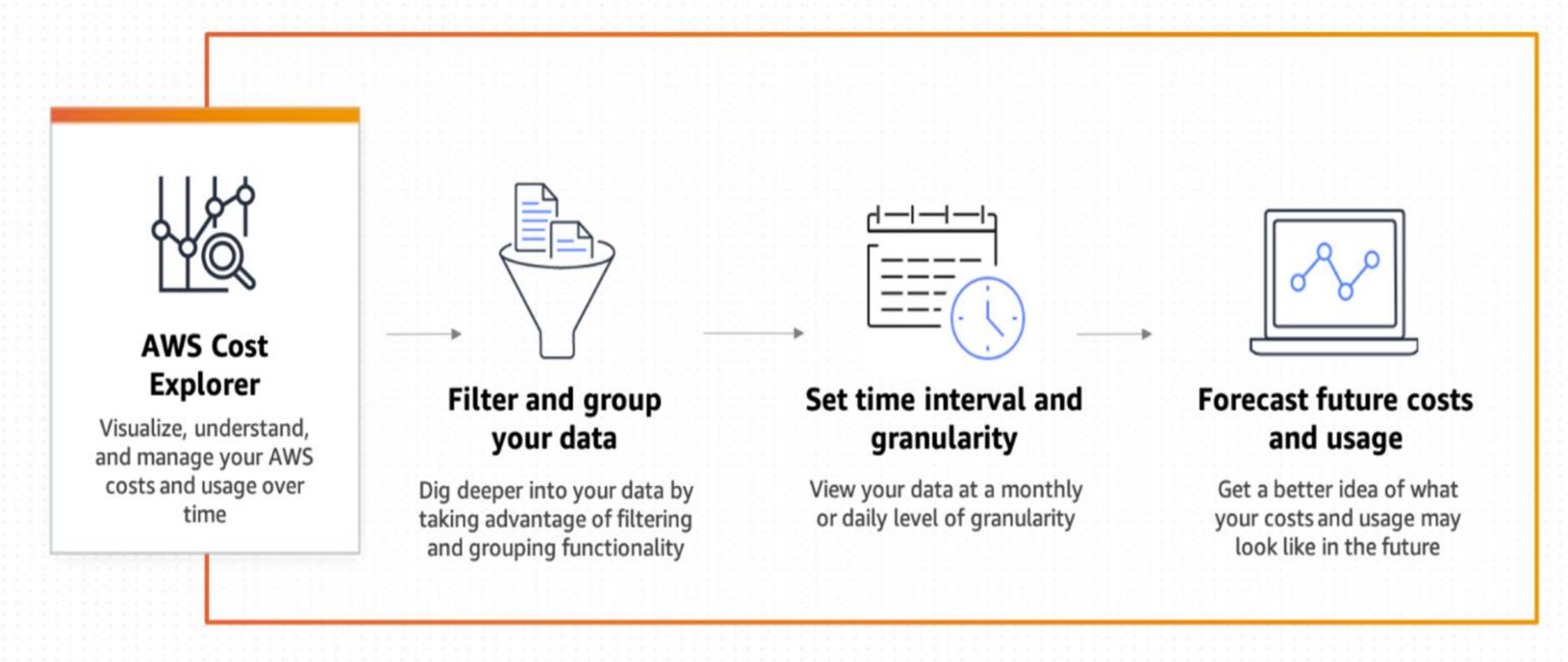
Reserved Instances

- Best for steady-state workloads
- Save money and retain flexibility with reserving capacity

Spot instances

- Similar to flying stand-by
- Ideal for flexible and interruptible workloads
- Up to 90% savings compared to on demand prices
- Best practices for fault-tolerant applications

AWS cost explorer



AWS Cost Explorer

Visualize, understand, and manage your AWS costs and usage over time

Filter and group your data

Dig deeper into your data by taking advantage of filtering and grouping functionality

Set time interval and granularity

View your data at a monthly or daily level of granularity

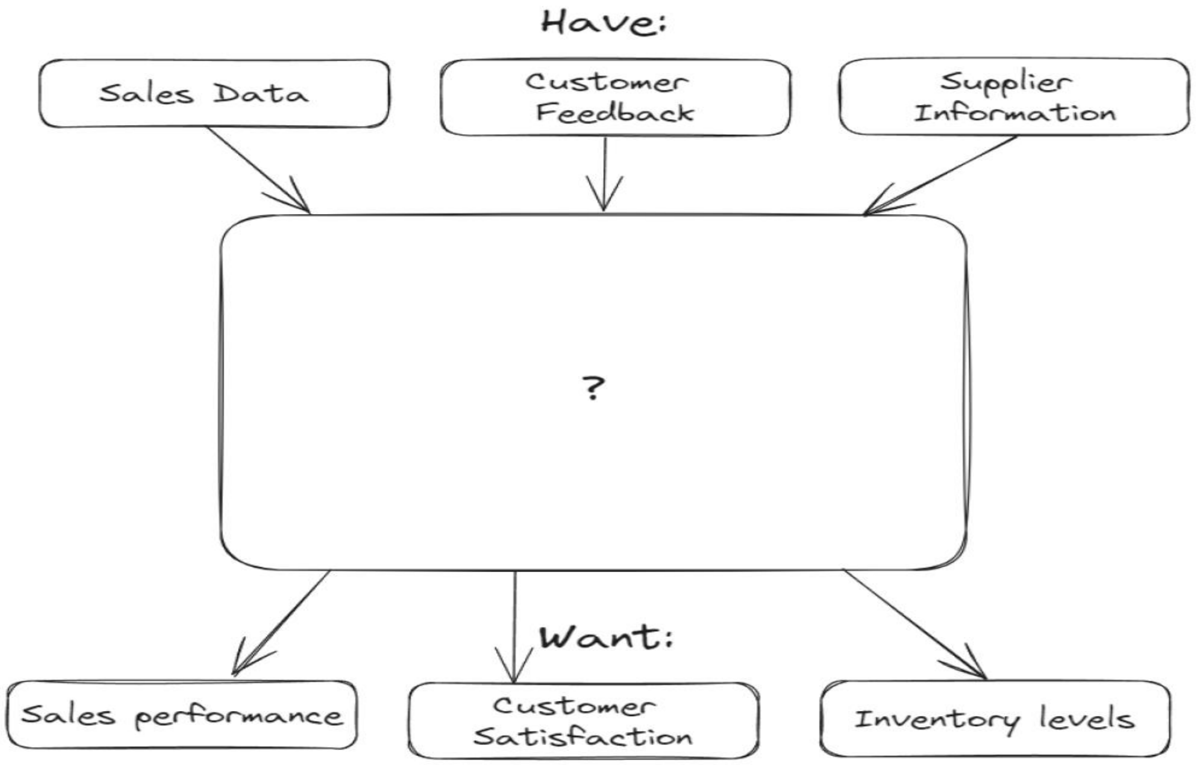
Forecast future costs and usage

Get a better idea of what your costs and usage may look like in the future

AWS budgets and cost alarms

- Set custom budget thresholds for your services
- Receive alerts before costs exceed your budget
- Integrate with AWS Cost Explorer for detailed budget tracking and forecasts

Gathering information from data



Introduction to Redshift

- Like a library
- Redshift as a data warehousing solution
- Extremely scalable
- Fast query performance

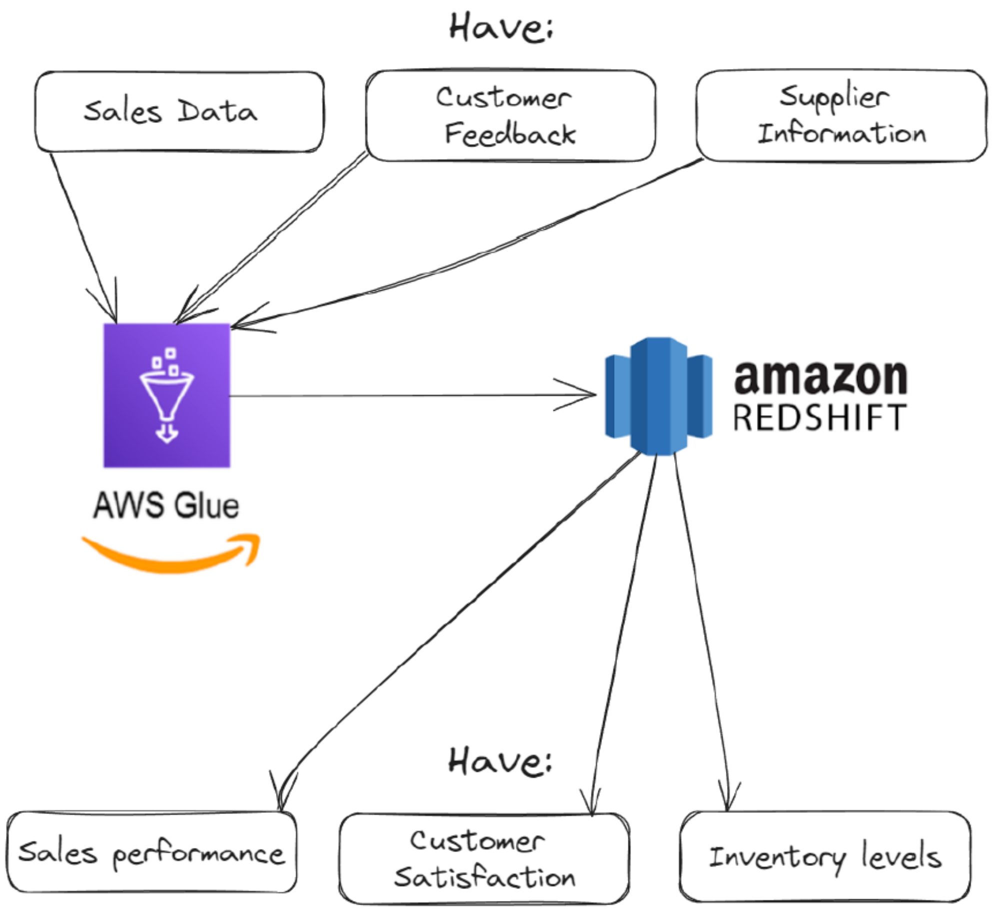
How does Redshift work?

- Columns and rows
- Optimized for analysis

GREENBOOTCAMPS

Redshift and AWS Glue in action

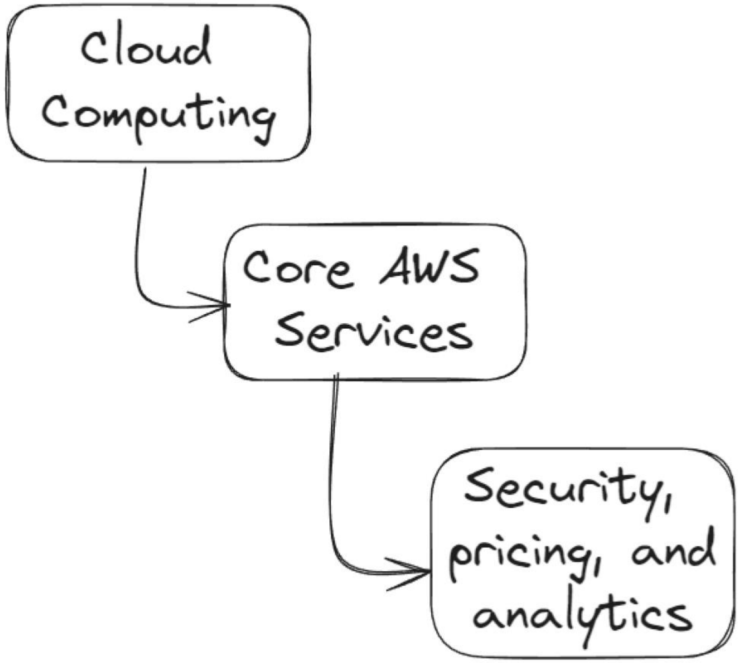
- 1. Data Preparation with AWS Glue
  - Discovers input data
  - Cleans and prepares data
- 2. Loading Data into Redshift
  - AWS Glue to load the data into Redshift
- 3. Analysis with Redshift
  - Run queries to analyze data



Analytics in AWS

- Can redshift handle unstructured data?
  - Not quite
  - Designed for structured data
- This lesson:
  - Understanding how AWS systems work together for analytics
  - Unravel complexity

AWS ecosystem: A Recap



Analytics and core AWS services

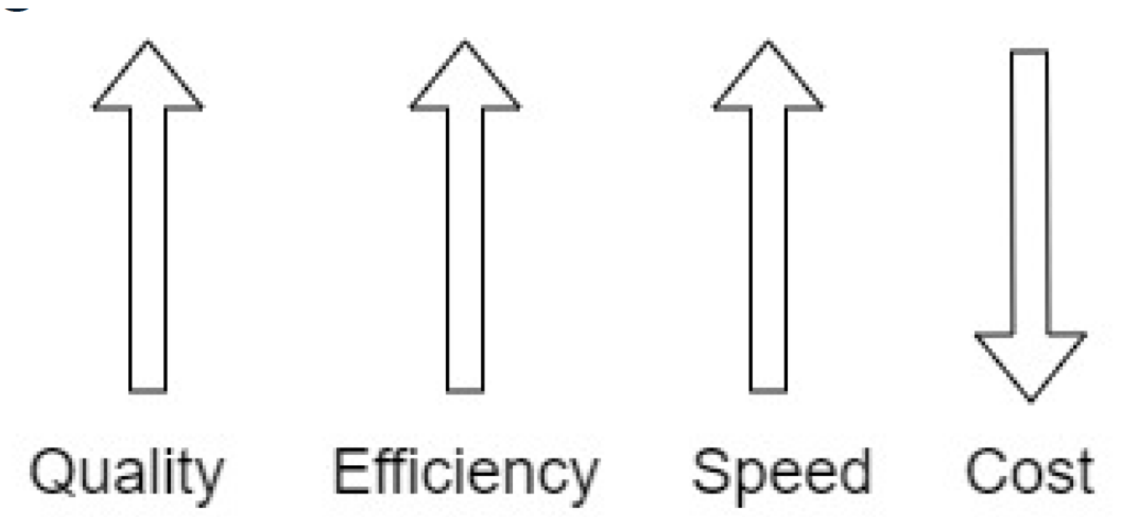
- EC2 and Lambda provide compute resources
- S3 and Glacier store data being analyzed

Global architecture and analytics

- Regions
- Availability zones
- Edge locations

Cost efficiency through analytics

- Analysis helps scrutinize patterns
- Creates avenues for cost optimization
- Helps maximize utility of AWS services



Azure - Microsoft Azure

- Launched in 2010, Azure is a cloud computing platform and service created by Microsoft.

Azure Key Services:

- **Compute:** Virtual Machines, Azure Functions (Serverless Computing), etc
- **Storage:** Blob Storage, Azure Files, etc
- **Database:** Azure SQL Database, Cosmos DB (NoSQL Database), etc
- **Networking:** Azure Virtual Network, Azure DNS, etc

Resources

Resource represents purchased service

- Web hosting
- Virtual machine
- Database

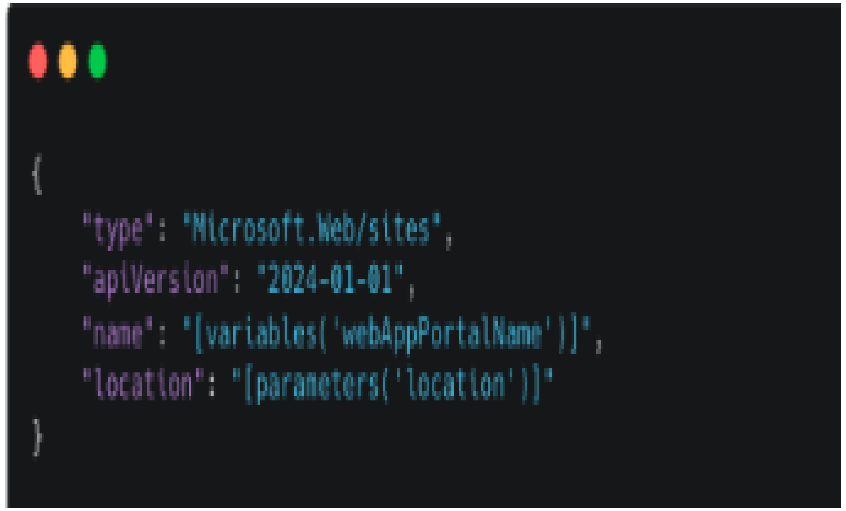
Generic JSON template

- Text-based file presenting structured data in JavaScript

Resource requires a Resource Group

Logical groupings to hold related resources

- Life-cycle
- Permissions
- Policies



```
{
    "type": "Microsoft.Web/sites",
    "apiVersion": "2024-01-01",
    "name": "[variables('webAppPortalName')]",
    "location": "[parameters('location')]"
}
```

Resources

Resource represents purchased service

- Web hosting
- Virtual machine
- Database

Generic JSON template

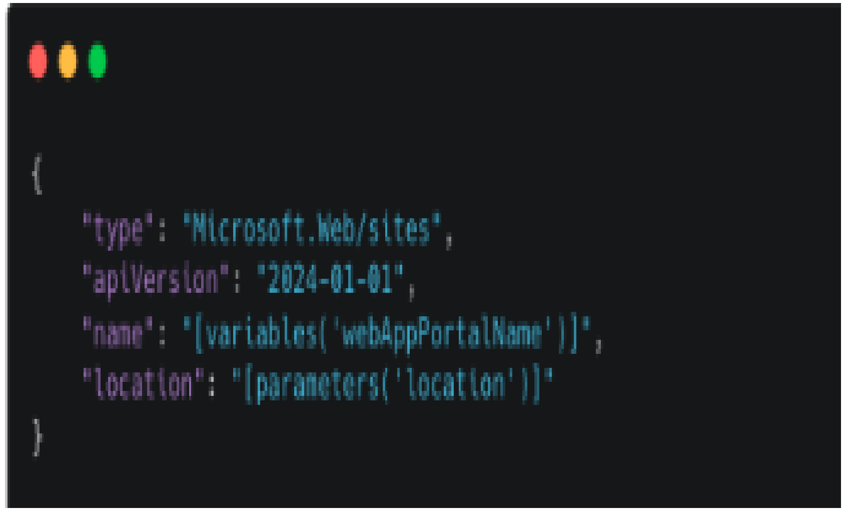
- Text-based file presenting structured data in JavaScript

Resource requires a Resource Group

Logical groupings to hold related resources

- Life-cycle
- Permissions
- Policies

Manage, monitor and maintain resources within group



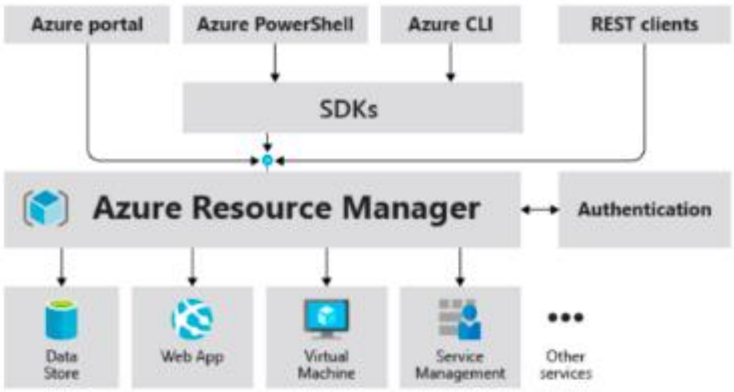
```
{
    "type": "Microsoft.Web/sites",
    "apiVersion": "2024-01-01",
    "name": "[variables('webAppPortalName')]",
    "location": "[parameters('location')]"
}
```

Azure Resource Manager (ARM)

Centralized management layer for resources and resource groups

Checks privileges against Active Directory for resource:

- Creation
- Management
- Deletion



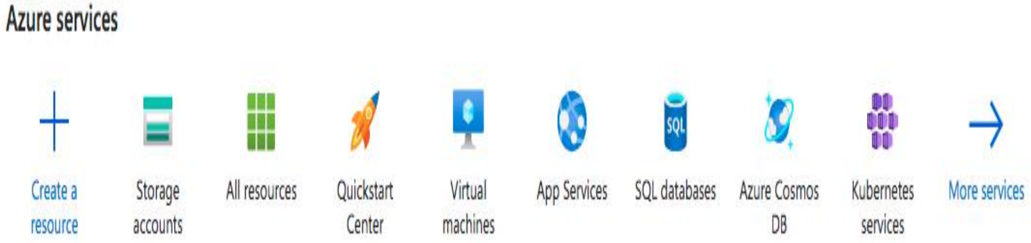
Core Offering

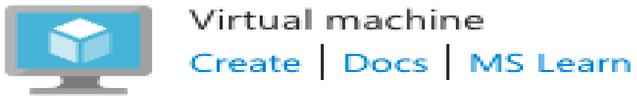
- Database
- Computation
- Storage
- Networking

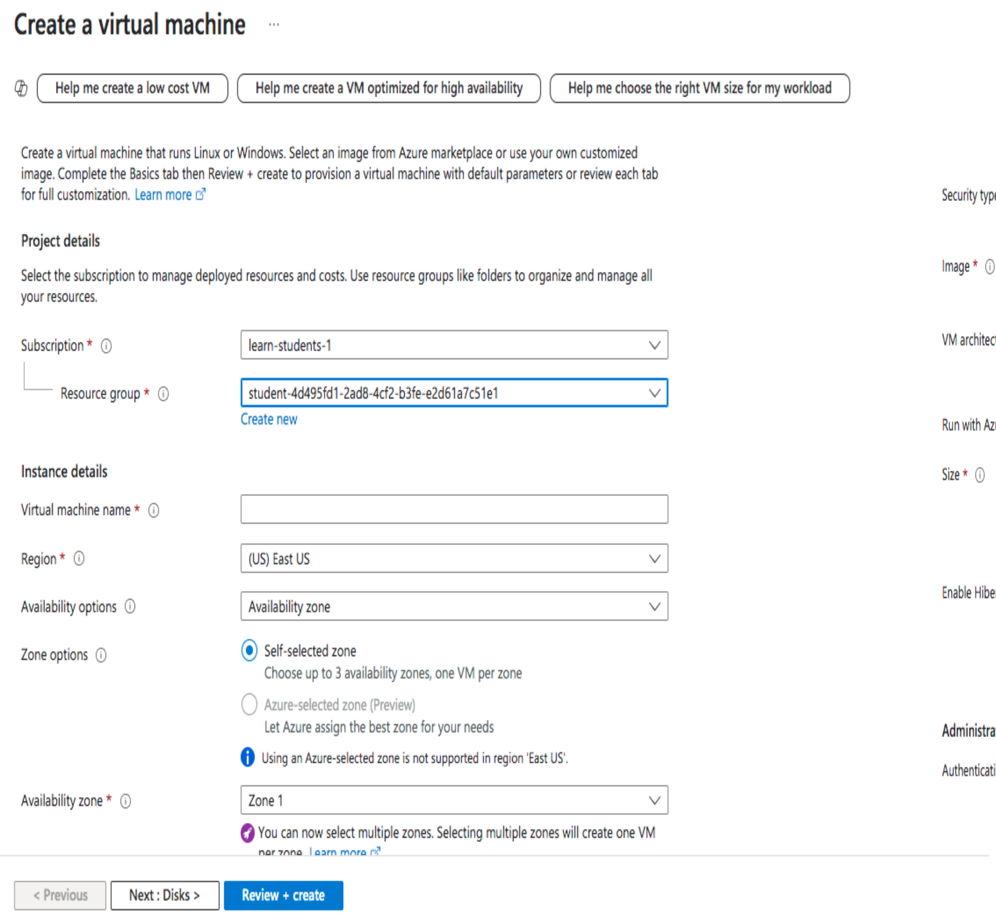
Create a VM in Azure

To create a new virtual machine, select Create and choose "Azure virtual machine" from the dropdown

Set a custom name for your virtual machine under the subscription pre-fixed with learn-students- and the resource group student-







Create a virtual machine ...

Help me create a low cost VM
Help me create a VM optimized for high availability
Help me choose the right VM size for my workload

Create a virtual machine that runs Linux or Windows. Select an image from Azure marketplace or use your own customized image. Complete the Basics tab then Review + create to provision a virtual machine with default parameters or review each tab for full customization. Learn more

Project details

Select the subscription to manage deployed resources and costs. Use resource groups like folders to organize and manage all your resources.

Subscription * : learn-students-1

Resource group * : student-4d495fd1-2ad8-4cf2-b3fe-e2d61a7c51e1

Create new

Instance details

Virtual machine name * :

Region * : (US) East US

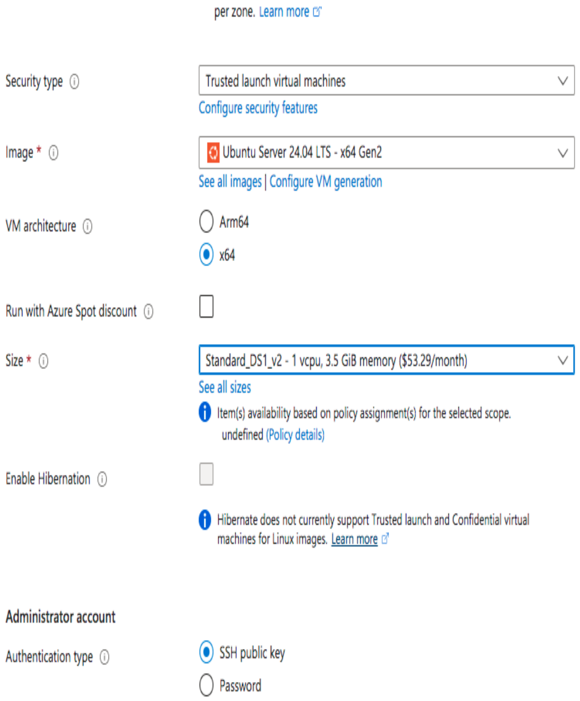
Availability options : Availability zone

Zone options :
● Self-selected zone
Choose up to 3 availability zones, one VM per zone
○ Azure-selected zone (Preview)
Let Azure assign the best zone for your needs

Using an Azure-selected zone is not supported in region 'East US'.

Availability zone * : Zone 1

You can now select multiple zones. Selecting multiple zones will create one VM per zone. Learn more



per zone. Learn more

Security type : Trusted launch virtual machines

Configure security features

Image * : Ubuntu Server 24.04 LTS - x64 Gen2

See all images | Configure VM generation

VM architecture :
○ Arm64
● x64

Run with Azure Spot discount : □

Size * : Standard_DS1_v2 - 1 vcpu, 3.5 GiB memory ($53.29/month)

See all sizes

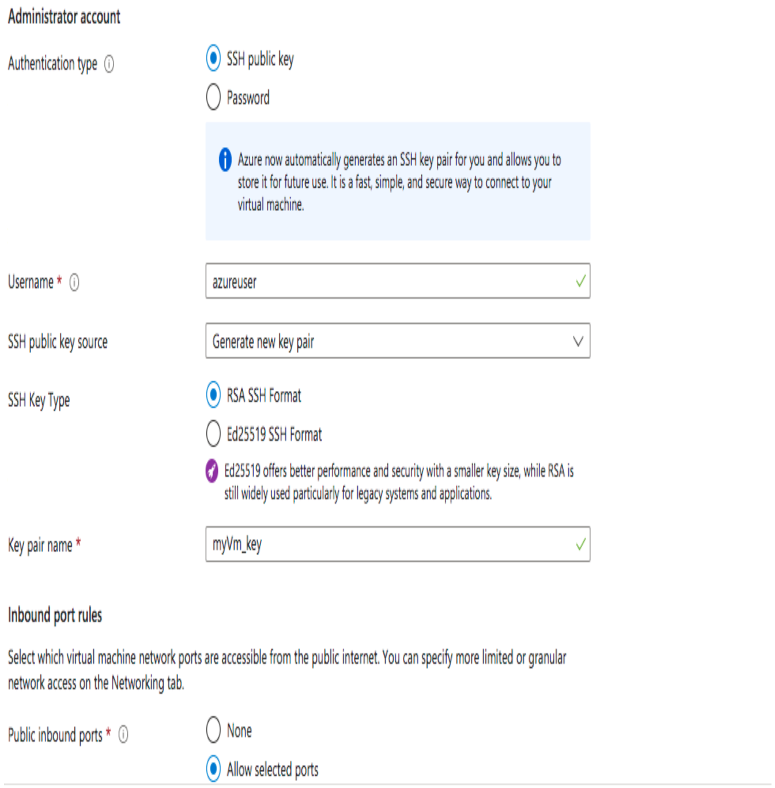
Item(s) availability based on policy assignment(s) for the selected scope. undefined (Policy details)

Enable Hibernation : □

Hibernate does not currently support Trusted launch and Confidential virtual machines for Linux images. Learn more

Administrator account

Authentication type :
● SSH public key
○ Password



Administrator account

Authentication type :
● SSH public key
○ Password

Azure now automatically generates an SSH key pair for you and allows you to store it for future use. It is a fast, simple, and secure way to connect to your virtual machine.

Username * : azureuser ✓

SSH public key source : Generate new key pair

SSH Key Type :
● RSA SSH Format
○ Ed25519 SSH Format

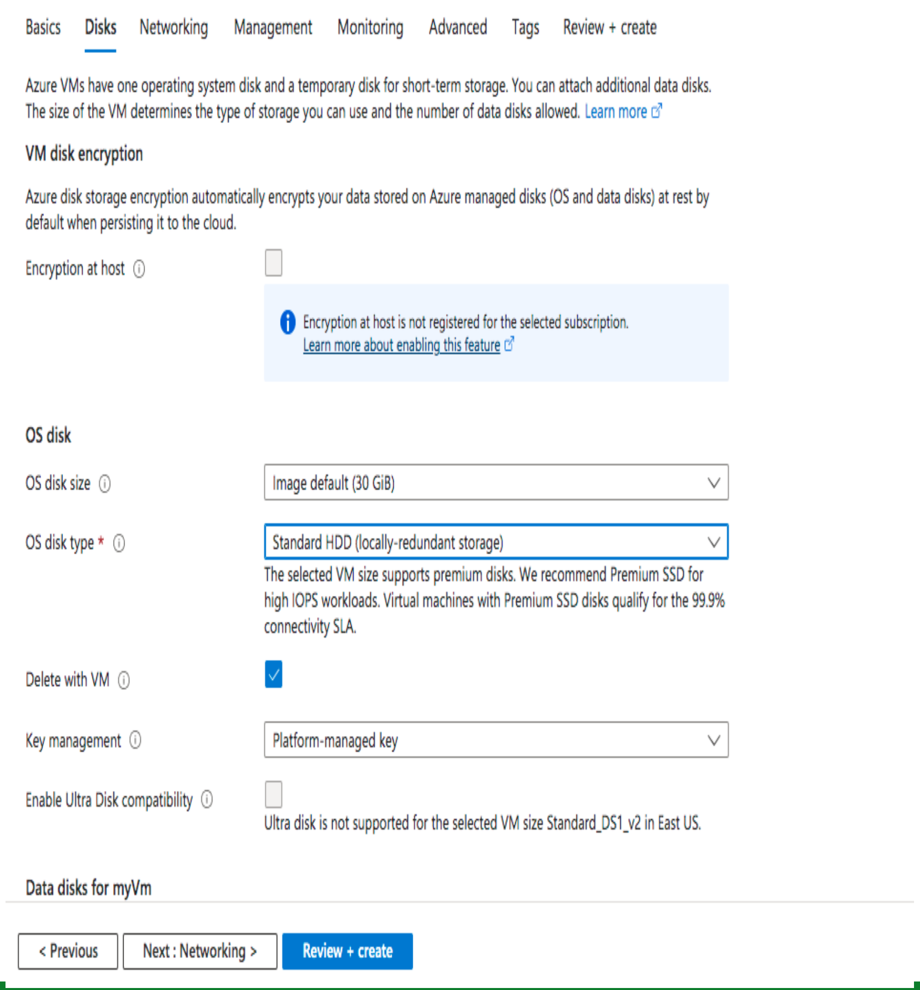
Ed25519 offers better performance and security with a smaller key size, while RSA is still widely used particularly for legacy systems and applications.

Key pair name * : myVm_key ✓

Inbound port rules

Select which virtual machine network ports are accessible from the public internet. You can specify more limited or granular network access on the Networking tab.

Public inbound ports * :
○ None
● Allow selected ports



Azure VMs have one operating system disk and a temporary disk for short-term storage. You can attach additional data disks. The size of the VM determines the type of storage you can use and the number of data disks allowed. Learn more

VM disk encryption

Azure disk storage encryption automatically encrypts your data stored on Azure managed disks (OS and data disks) at rest by default when persisting it to the cloud.

Encryption at host ⓘ □

Encryption at host is not registered for the selected subscription.
Learn more about enabling this feature

OS disk

OS disk size ⓘ: Image default (30 GiB)

OS disk type * ⓘ: Standard HDD (locally-redundant storage)

The selected VM size supports premium disks. We recommend Premium SSD for high IOPS workloads. Virtual machines with Premium SSD disks qualify for the 99.9% connectivity SLA.

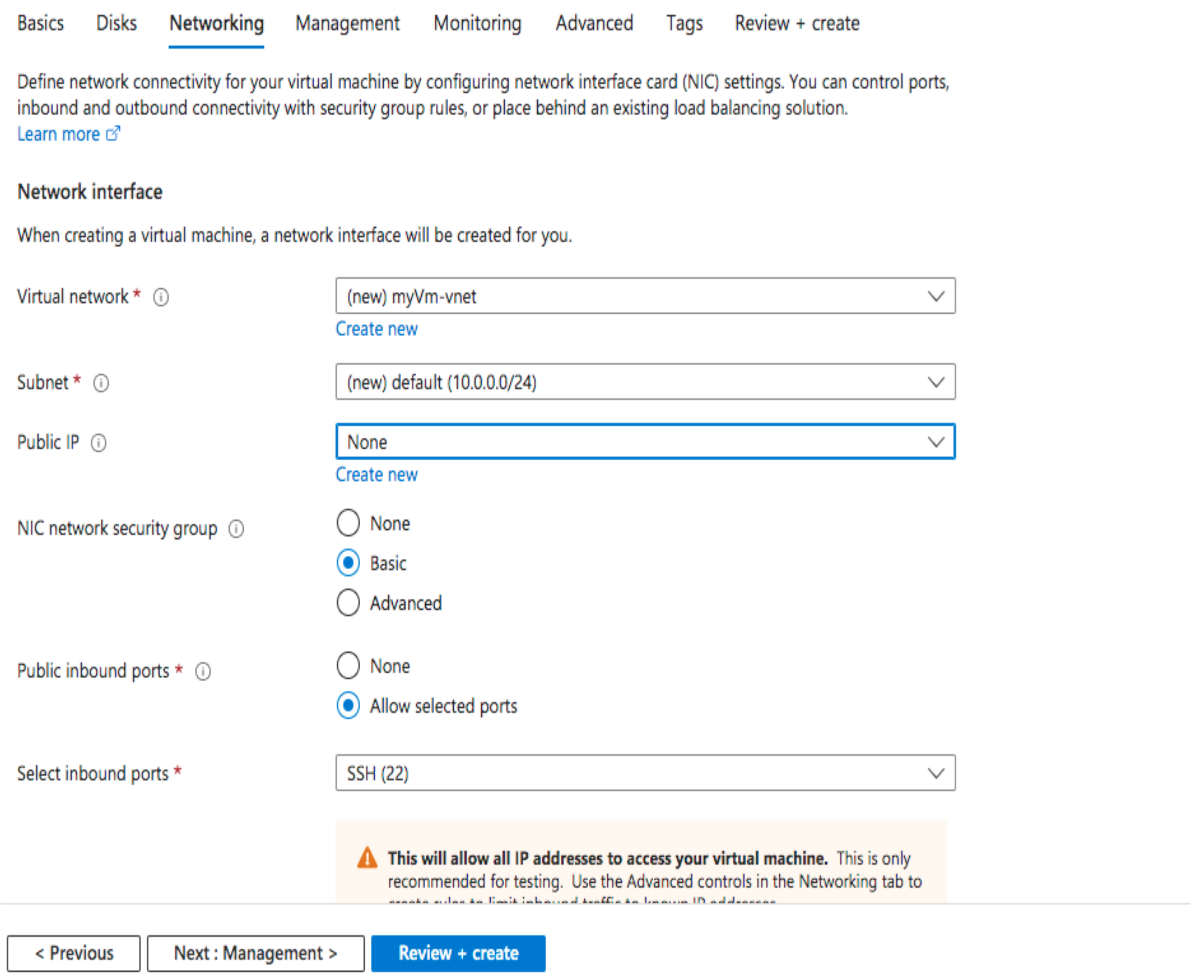
Delete with VM ⓘ ☑

Key management ⓘ: Platform-managed key

Enable Ultra Disk compatibility ⓘ □

Ultra disk is not supported for the selected VM size Standard_DS1_v2 in East US.

Data disks for myVm



Define network connectivity for your virtual machine by configuring network interface card (NIC) settings. You can control ports, inbound and outbound connectivity with security group rules, or place behind an existing load balancing solution.
Learn more

Network interface

When creating a virtual machine, a network interface will be created for you.

Virtual network * ⓘ: (new) myVm-vnet
Create new

Subnet * ⓘ: (new) default (10.0.0.0/24)

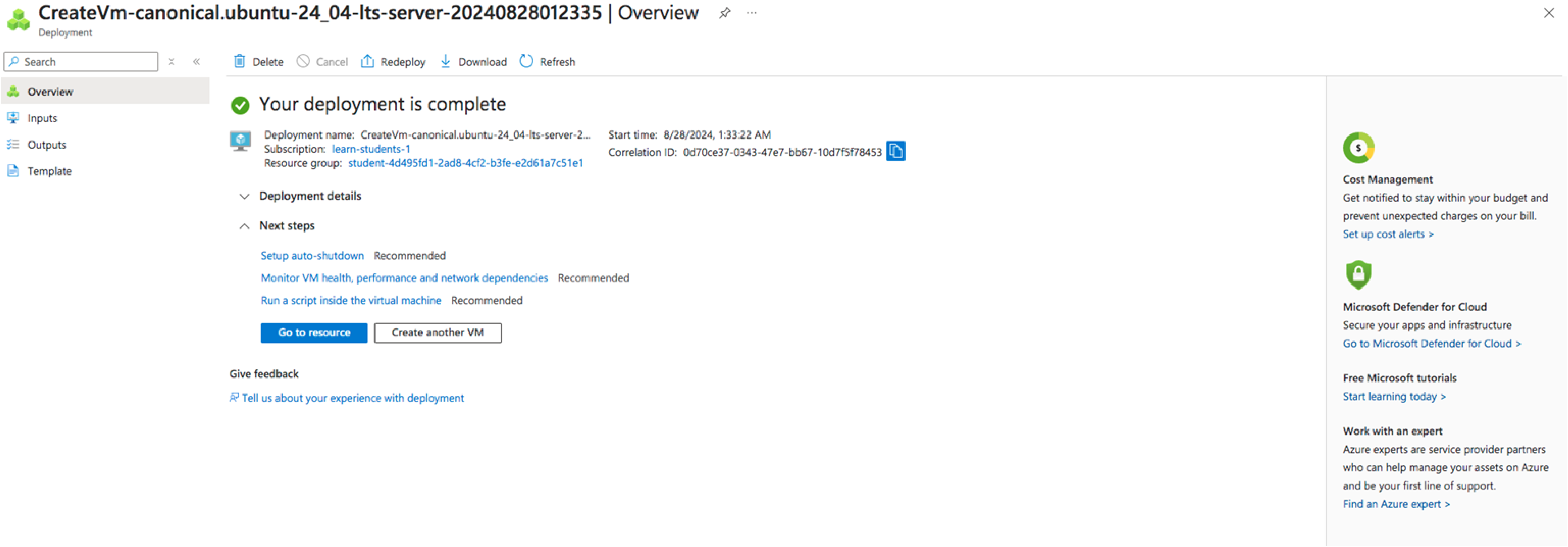
Public IP ⓘ: None
Create new

NIC network security group ⓘ: ○ None ● Basic ○ Advanced

Public inbound ports * ⓘ: ○ None ● Allow selected ports

Select inbound ports *: SSH (22)

This will allow all IP addresses to access your virtual machine. This is only recommended for testing. Use the Advanced controls in the Networking tab to create rules to limit inbound traffic to known IP addresses.



CreateVm-canonical.ubuntu-24_04-lts-server-20240828012335 | Overview

Deployment

Your deployment is complete

Deployment name: CreateVm-canonical.ubuntu-24_04-lts-server-2...
Subscription: learn-students-1
Resource group: student-4d495fd1-2ad8-4cf2-b3fe-e2d61a7c51e1

Start time: 8/28/2024, 1:33:22 AM
Correlation ID: 0d70ce37-0343-47e7-bb67-10d7f5f78453

Deployment details

Next steps

Setup auto-shutdown Recommended

Monitor VM health, performance and network dependencies Recommended

Run a script inside the virtual machine Recommended

Give feedback

Tell us about your experience with deployment

Cost Management

Get notified to stay within your budget and prevent unexpected charges on your bill.

Set up cost alerts >

Microsoft Defender for Cloud

Secure your apps and infrastructure

Go to Microsoft Defender for Cloud >

Free Microsoft tutorials

Start learning today >

Work with an expert

Azure experts are service provider partners who can help manage your assets on Azure and be your first line of support.

Find an Azure expert >

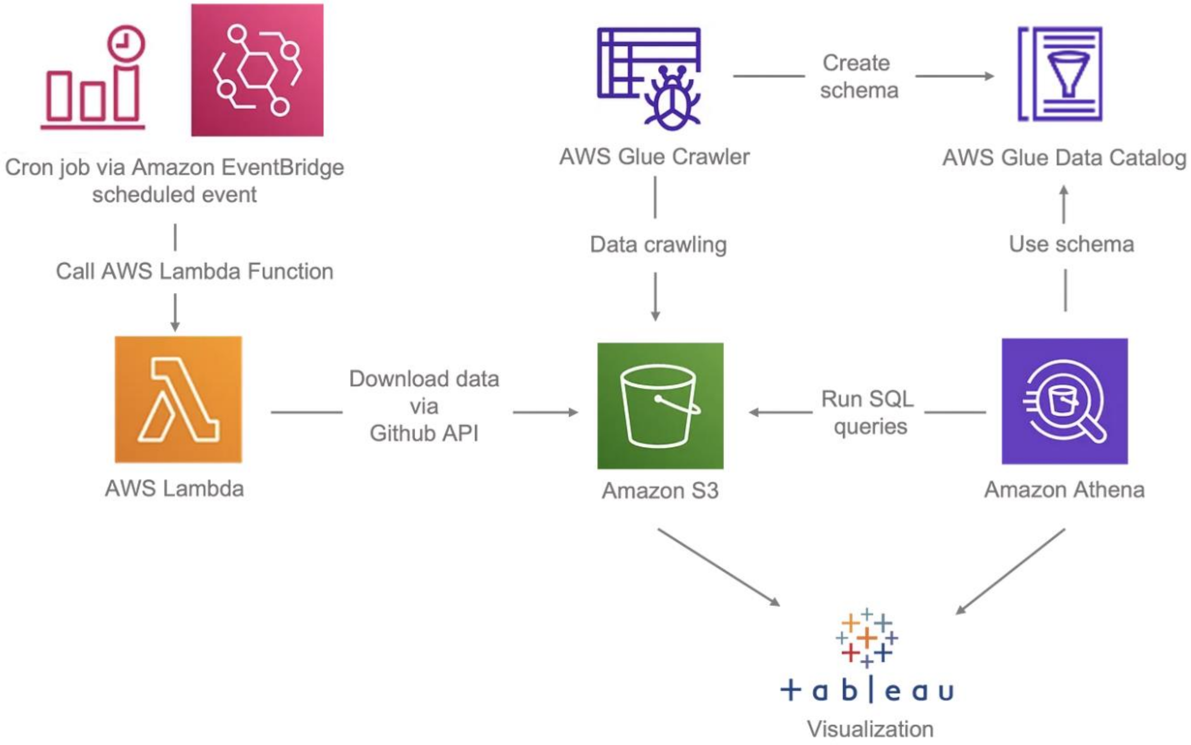
Now we look at some of the AWS Use-Cases

AWS Data Science Use-Cases

Let's say we want to build a data pipeline for ingesting the 100 latest repositories through the GitHub API and visualizing the data using Tableau.

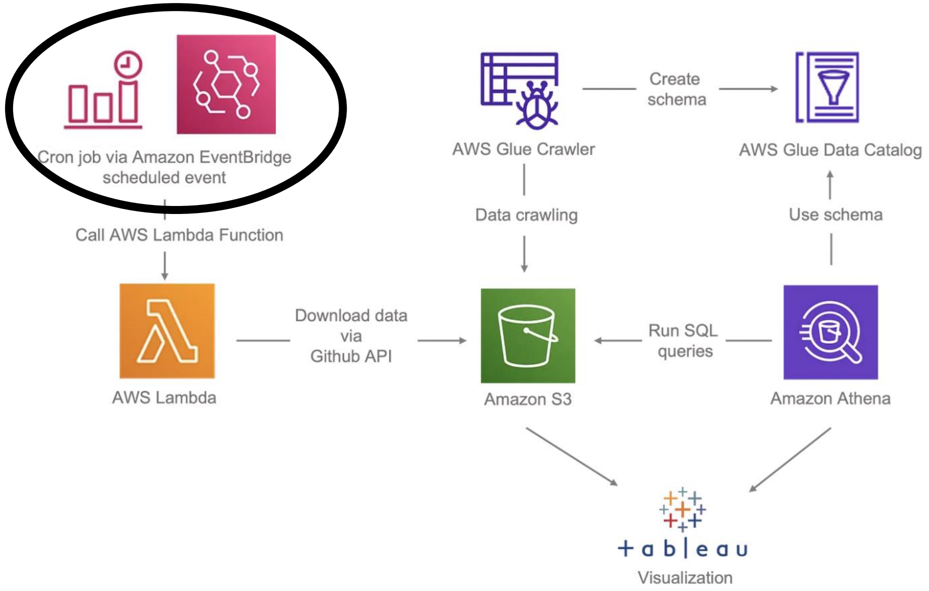
AWS Data Science Use-Cases

- **This project will follow the following structure or roadmap**



AWS Data Science Use-Cases

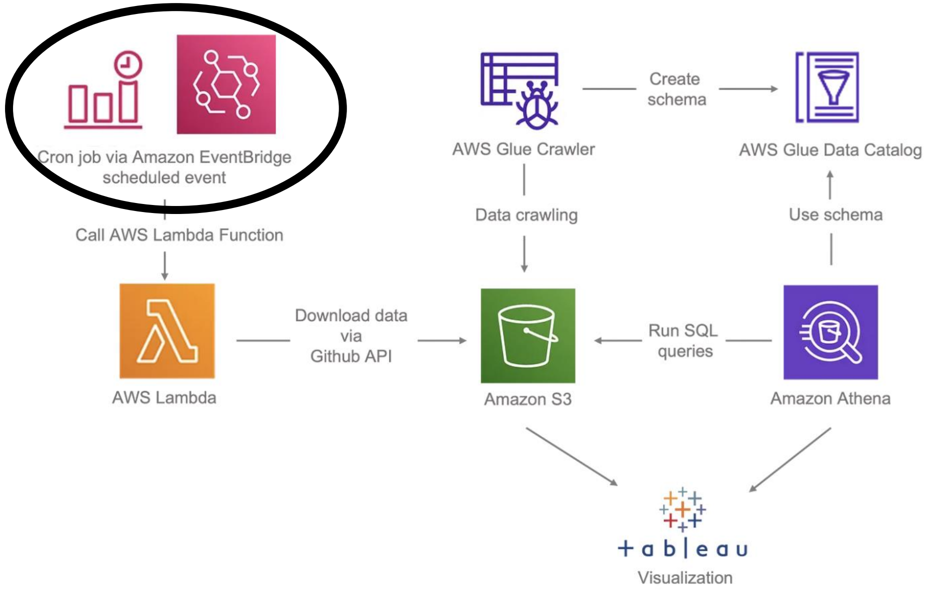
- The purpose of this step is to automate the scheduling of the data ingestion process.
- EventBridge (formerly known as CloudWatch Events) allows you to create rules that automatically trigger AWS services based on time-based schedules or events.



AWS Data Science Use-Cases

Creating an Event Bridge rule:

- Definition: A rule that matches incoming events and routes them to one or more target functions or services. In this case, the event is a time-based schedule (cron job).
- Configuration: Define the schedule using a cron expression or rate expression.
  - Cron Expression: Specifies exact times and dates for execution. For example, cron(0 12 * * ? *) triggers at 12:00 PM (UTC) every day.
  - Rate Expression: Specifies intervals for execution. For example, rate(1 hour) triggers every hour.

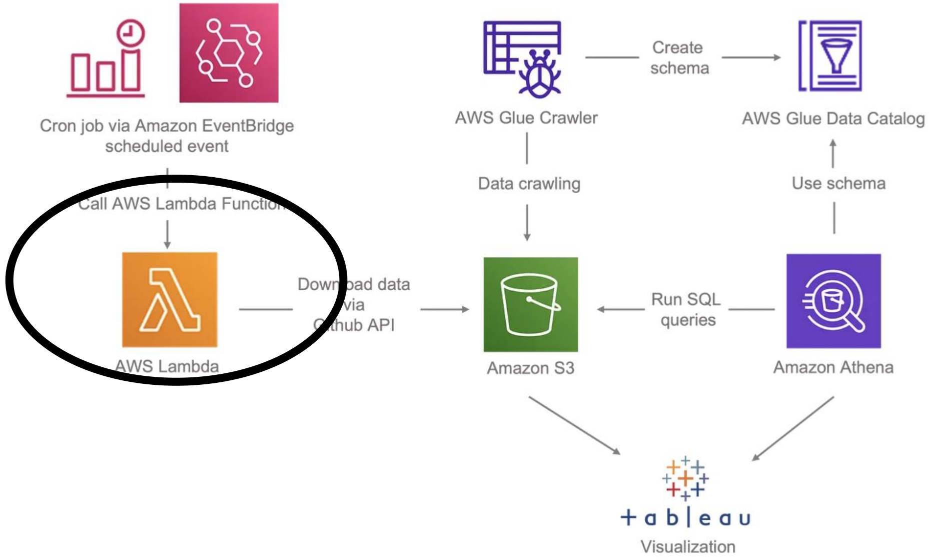


AWS Data Science Use-Cases

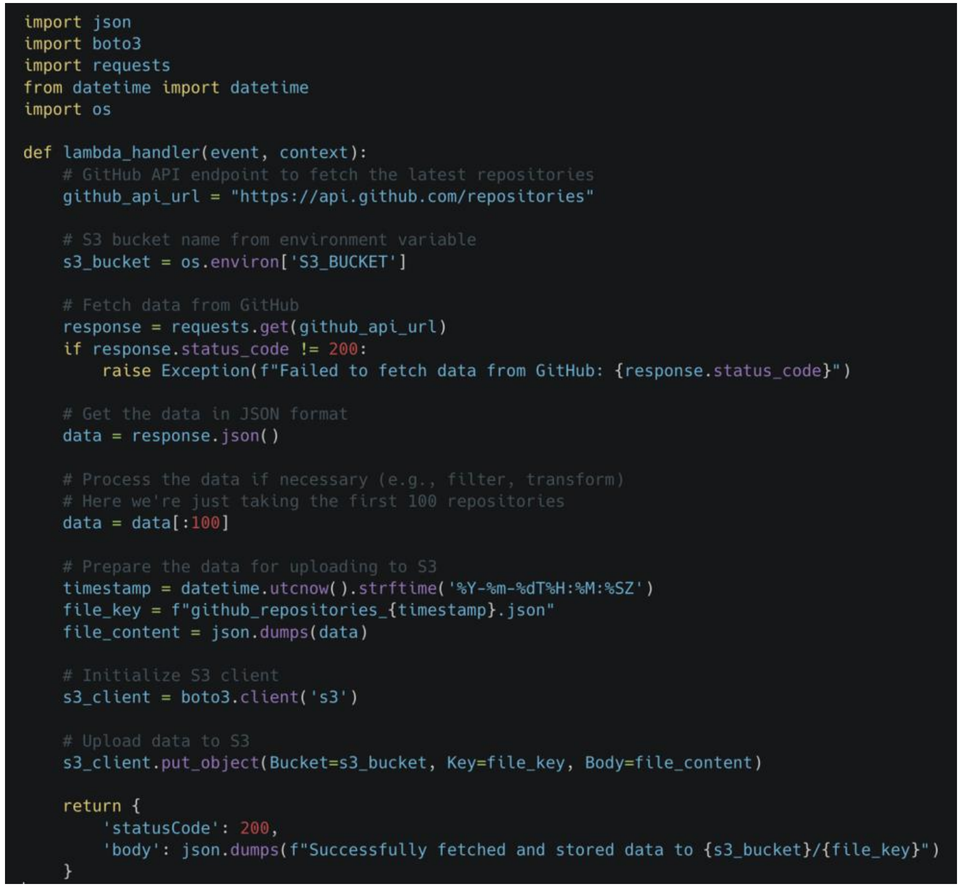
What is Lambda Function:

- AWS Lambda function is a compute service that runs code in response to events and automatically manages the underlying compute resources.

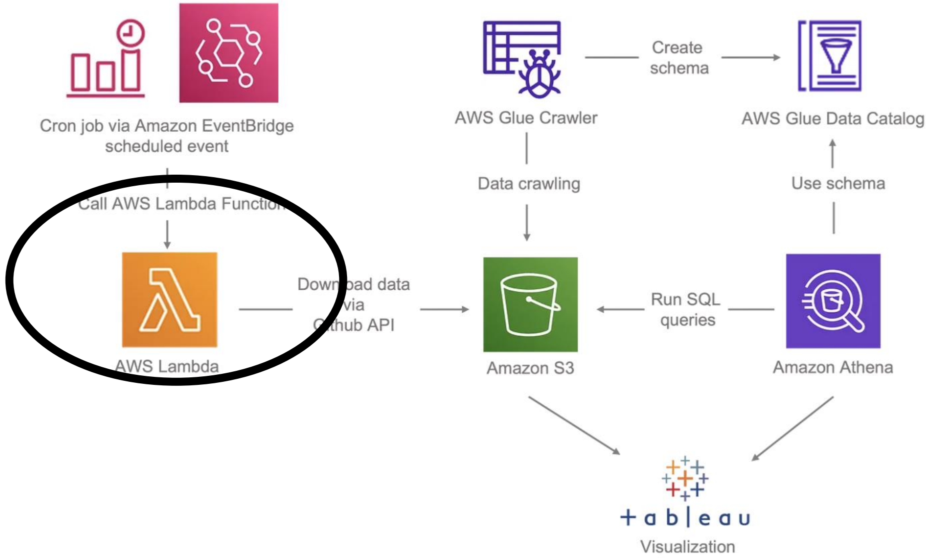
- In this project, the Lambda function is responsible for fetching the latest 100 GitHub repositories and storing the data in Amazon S3.



AWS Data Science Use-Cases



```python
import json
import boto3
import requests
from datetime import datetime
import os

def lambda_handler(event, context):
    # GitHub API endpoint to fetch the latest repositories
    github_api_url = "https://api.github.com/repositories"

    # S3 bucket name from environment variable
    s3_bucket = os.environ['S3_BUCKET']

    # Fetch data from GitHub
    response = requests.get(github_api_url)
    if response.status_code != 200:
        raise Exception(f"Failed to fetch data from GitHub: {response.status_code}")

    # Get the data in JSON format
    data = response.json()

    # Process the data if necessary (e.g., filter, transform)
    # Here we're just taking the first 100 repositories
    data = data[:100]

    # Prepare the data for uploading to S3
    timestamp = datetime.utcnow().strftime('%Y-%m-%dT%H:%M:%SZ')
    file_key = f"github_repositories_{timestamp}.json"
    file_content = json.dumps(data)

    # Initialize S3 client
    s3_client = boto3.client('s3')

    # Upload data to S3
    s3_client.put_object(Bucket=s3_bucket, Key=file_key, Body=file_content)

    return {
        'statusCode': 200,
        'body': json.dumps(f"Successfully fetched and stored data to {s3_bucket}/{file_key}")
    }
```



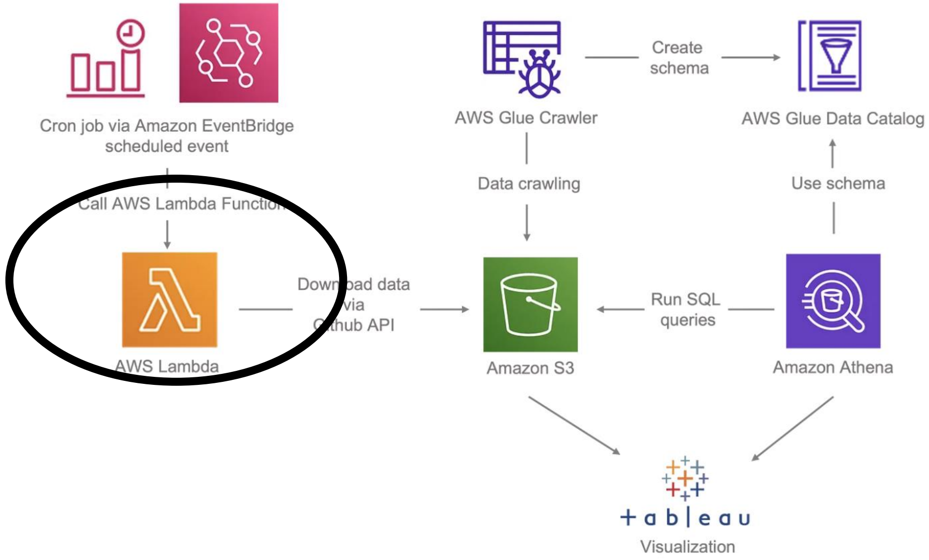
Implementation in Python for the Lambda function

AWS Data Science Use-Cases

Set the Target:

- <u>AWS Lambda Function:</u> The target in this scenario is an AWS Lambda function. When the scheduled event triggers, it invokes this Lambda function to perform the data ingestion.

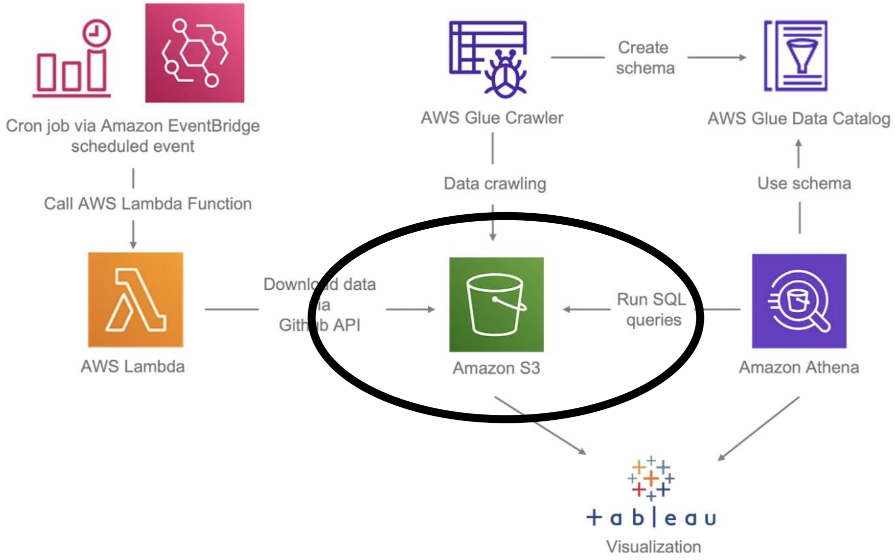
- <u>Target Configuration:</u> Specify the ARN (Amazon Resource Name) of the Lambda function to be invoked and any necessary input parameters.



AWS Data Science Use-Cases

Amazon S3 Bucket:

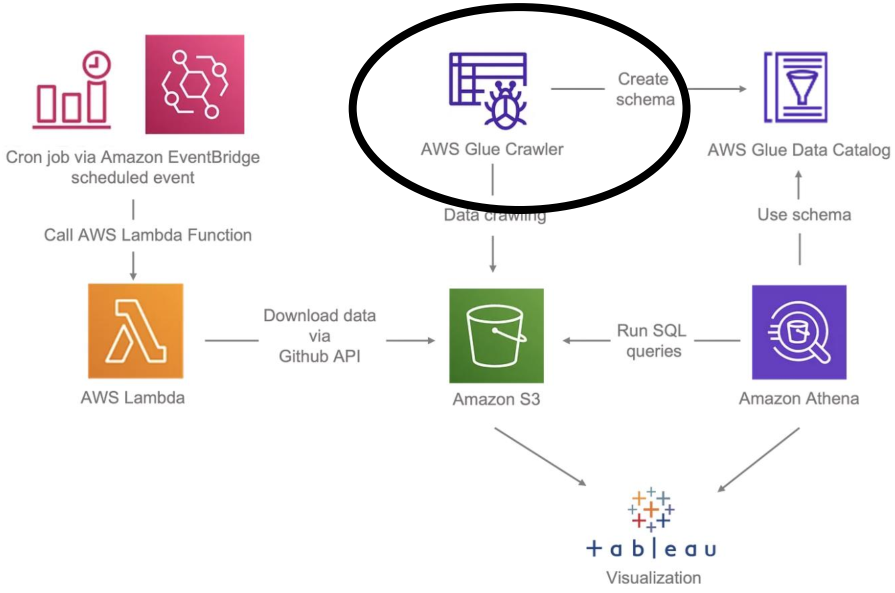
- Amazon S3 serves as the storage layer in this data pipeline.
- It provides scalable, durable, and secure storage for the data fetched from the GitHub API by the AWS Lambda function.
- The data is stored in a structured manner, enabling subsequent processing and analysis.



AWS Data Science Use-Cases

AWS Glue Crawler:

- AWS Glue Crawler is a serverless tool provided by AWS Glue, which automatically discovers and catalogs metadata about data stored in various data stores.
- The crawler can scan data in different formats and structures, inferring schemas and populating the AWS Glue Data Catalog with metadata tables.
- This makes the data easily queryable and usable by other AWS services, such as Amazon Athena, Amazon Redshift Spectrum, and AWS Glue ETL jobs.



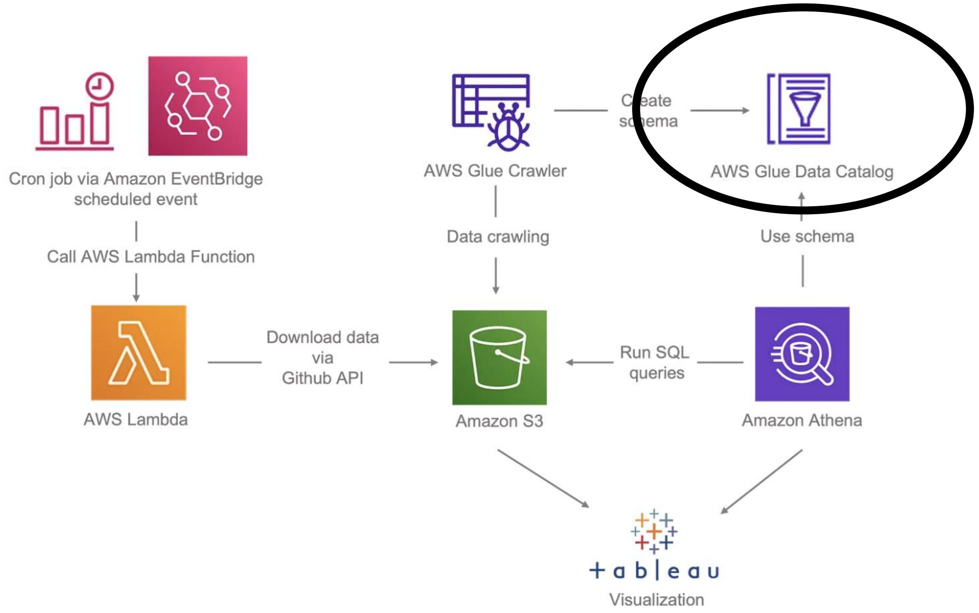
AWS Data Science Use-Cases

AWS Glue Catalog:

- The AWS Glue Data Catalog is a centralized metadata repository that stores information about data stored across various data stores, including Amazon S3, Amazon RDS, Amazon Redshift, and more.

- In simple words, the Data Catalog allows users to search for datasets, view their schema, and understand their structure without directly accessing the raw data.

- It is a core component of AWS Glue and serves as the foundation for managing, discovering, and querying data.

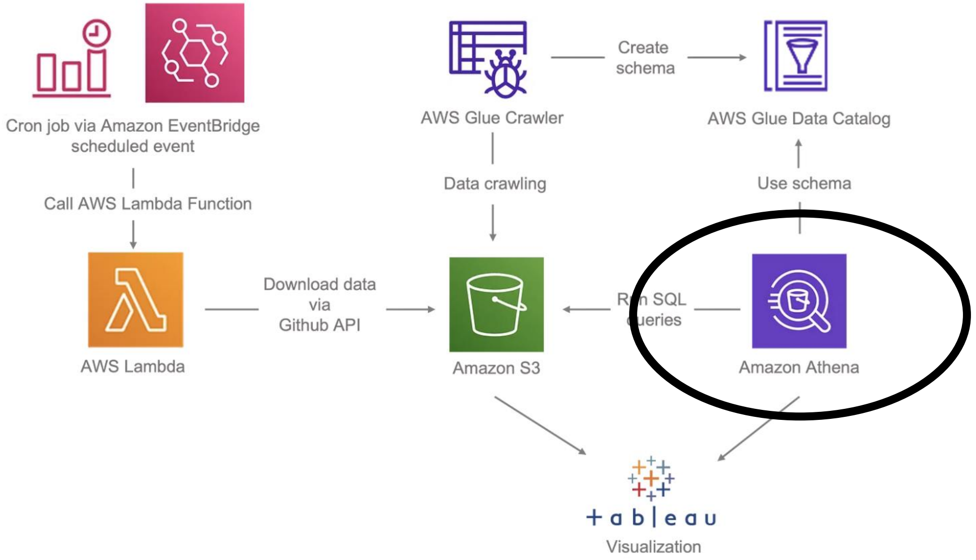


AWS Data Science Use-Cases

Amazon Athena:

- Amazon Athena is an interactive query service that makes it easy to analyse data in Amazon S3 using standard SQL.

- Athena is serverless, so there is no infrastructure to manage, and you pay only for the queries that you run.

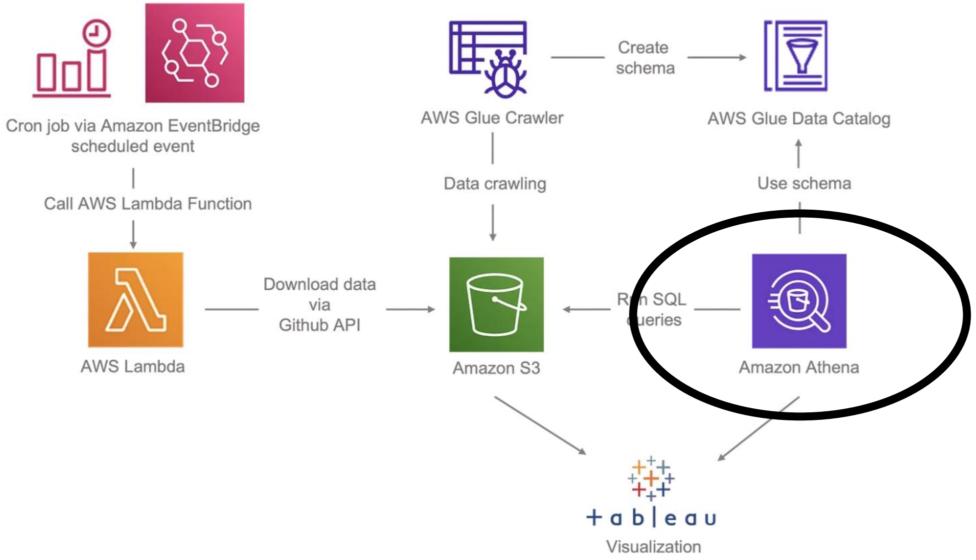


AWS Data Science Use-Cases

Amazon Athena:

Integration with Visualization Tools:

- Purpose: To enable seamless integration with data visualization and BI tools like Tableau, QuickSight, and others.
- Functionality: Athena can act as a data source for these tools, allowing them to execute SQL queries and fetch results for visualization.
- Benefit: Provides an easy way to create dashboards, reports, and visualizations based on the latest data.



AWS Data Science Use-Cases

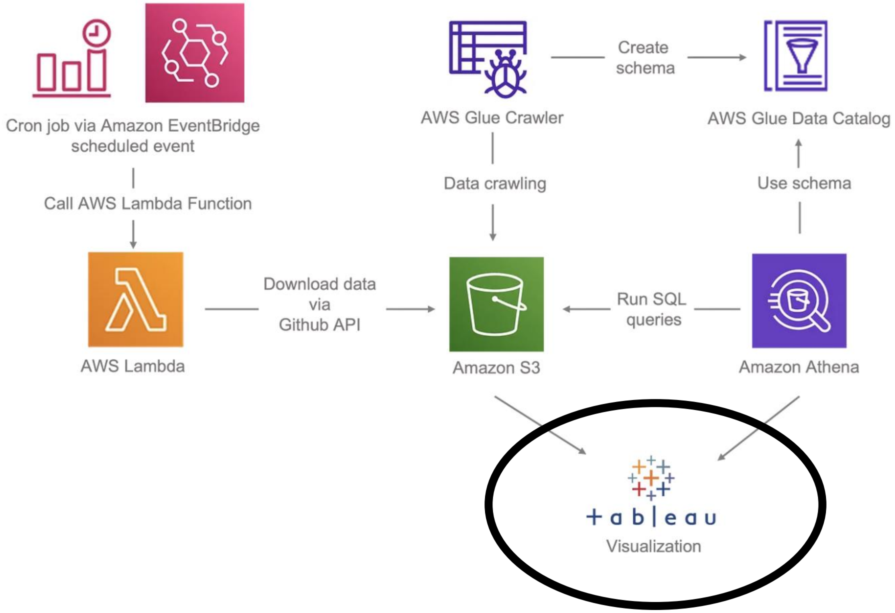
And finally, one can use Tableau to create a variety of visualizations to represent and analyse the GitHub repository data.

You can create the following visualizations:

Time Series Analysis:
Line Chart: To show trends over time, such as the number of repositories created per day/week/month.

Distribution Analysis:
Histogram: To show the distribution of a particular metric, such as the number of stars or forks.

..and many more.



Thank you!