Parul® University

AWS Fundamental

, Assistant Professor
Computer Science & Engineering

Parul®
University

DIGITAL LEARNING CONTENT

CHAPTER-3

AWS Compute Services

Parul®
University

DIGITAL LEARNING CONTENT

Cloud Computing - IAAS

- **Definition: Cloud computing refers to the on demand provision of computational resources (data, software) via a computer network, rather than from a local computer.**

Parul®
University

DIGITAL LEARNING CONTENT

What is Cloud Computing?

Cloud computing is the delivery of computing services over the internet.

Computing services include common IT infrastructure such as virtual machines, storage, databases, and networking. Cloud services also expand the traditional IT offerings to include things like Internet of Things (IoT), machine learning (ML), and artificial intelligence (AI).

As cloud computing uses the internet to deliver these services, it doesn't have to be constrained by physical infrastructure the same way that a traditional datacenter is.

Parul®
University

DIGITAL LEARNING CONTENT

What is Cloud Computing?

That means if you need to increase your IT infrastructure rapidly, you don't have to wait to build a new datacenter—you can use the cloud to rapidly expand your IT footprint.

Cloud computing is the delivery of computing resources, like storage, servers, databases, and software, over the internet. It allows users to access these resources on-demand, without having to purchase and maintain their own physical infrastructure.

Cloud computing offers many benefits, including:

Cost savings: Users only pay for the services they use, which can help lower operating costs.

Parul®
University

DIGITAL LEARNING CONTENT

What is Cloud Computing?

Flexibility: Users can scale their resources up or down as their needs change.

Innovation: Users can access the resources they need to innovate faster.

Efficiency: Users can run their infrastructure more efficiently.

Parul®
University

DIGITAL LEARNING CONTENT

What is Cloud Computing?

Some common cloud computing services include:

Infrastructure as a service: Provides compute and storage services.

Platform as a service: Provides an environment for developing and deploying cloud apps.

Software as a service: Delivers apps as services.

Serverless computing: Offloads back-end infrastructure management tasks to the cloud provider.

Function-as-a-service: A subset of serverless computing that allows developers to run portions of application code in response to specific events.

Packets

- **Cloud computing providers are typically responsible for infrastructure security, physical hardware, and software updates. Customers are responsible for data encryption, identity and access management, and application-level security.**

Parul®
University

DIGITAL LEARNING CONTENT

IAAS

IaaS stands for Infrastructure as a Service, as the name suggests, and is a cloud technology service where the IaaS providers make available various infrastructural services on the Cloud.

For example, through virtualization technology, the provider hosts infrastructural components such as virtual machines, virtual LANs, networks, storage, hard drives, etc.

IaaS (infrastructure as a service) describes the provision of infrastructure and computing resources as a service.

Parul®
University

DIGITAL LEARNING CONTENT

IAAS

IaaS (includes all the physical computing resources that facilitate the delivery of apps as a service).

IaaS is also known as HaaS (hardware as a service).

Amazon Web Services (AWS) is one of the best IaaS examples. Cisco Metapod, Microsoft Azure, and Google Compute Engine (GCE) are also some example of IaaS.

IAAS

The IaaS provider provides his client with either a dashboard or an API. Then, it allows access to the end-user to all the services.

This way, the end-user has complete control over the entire infrastructure.

Along with infrastructural services, the provider will supply accompanying services.

These services, at last, will be rendered to support this infrastructural assistance.

Some examples of these services are detailed billing, load balancing, and clustering, log access.

In addition to all the above, the IaaS provider will enable data-storage-related services such as data back-up, replication, and recovery.

Parul®
University

DIGITAL LEARNING CONTENT

Example and Applications of IAAS

Examples of IaaS

- Microsoft Azure
- Amazon Web Services
- Google Compute Engine
- Cisco Metapod
- Joyent
- OpenStack

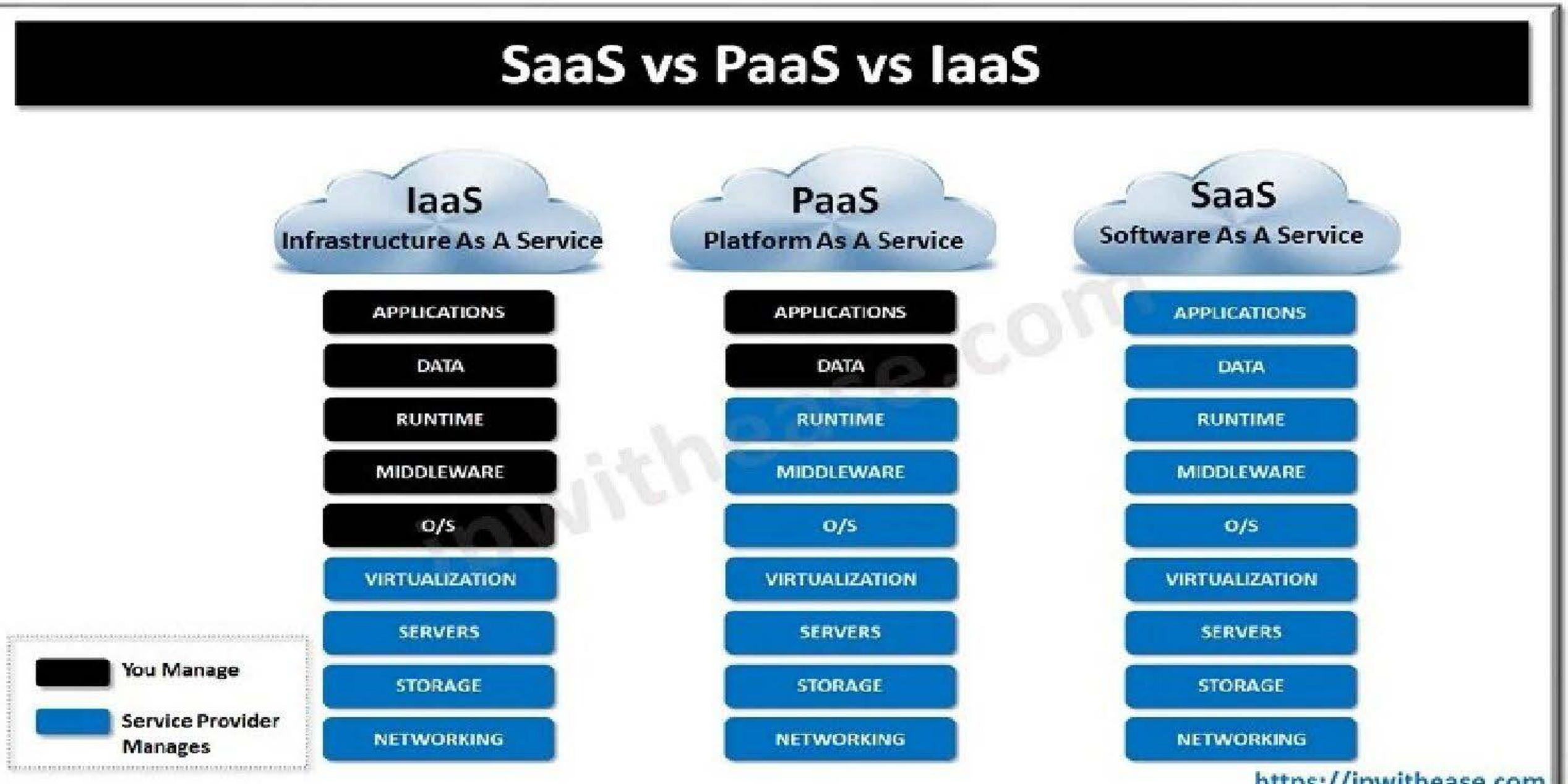
Applications of IaaS:

- IaaS is used in the following scenarios in the current business environment:
  - Development environment
  - Testing environment
  - Data storage
  - Data Analytics

Parul®
University

DIGITAL LEARNING CONTENT

IAAS Vs PAAS Vs SAAS



Parul®
University

DIGITAL LEARNING CONTENT

Amazon EC2 Overview

Amazon EC2 is a scalable virtual computing service that offers a variety of instance types for different workloads. It supports multiple instance types, including general-purpose, compute-optimized, and GPU instances, allowing users to choose the best-suited options for their needs.

EC2 also provides features like high-precision time synchronization and a wide range of operating system and software choices. It integrates with other AWS services and offers secure environments with VPCs.

Key Features:
1) Scalable Virtual Servers:
EC2 provides on-demand virtual servers (instances) that can be easily scaled up or down to meet changing workload demands.

Amazon EC2 Overview

2) Multiple Instance Types:

EC2 offers various instance types, including:

General Purpose: Balanced mix of CPU, memory, and network resources, suitable for general applications.

Compute Optimized: Designed for compute-intensive tasks, such as batch processing and scientific modeling.

GPU: Powered by NVIDIA GPUs, optimized for applications like machine learning, graphics rendering, and deep learning.

3) Operating Systems and Software:

EC2 supports a wide variety of operating systems (Linux, Windows, etc.) and software through Amazon Machine Images (AMIs) and the AWS Marketplace.

Amazon EC2 Overview

4) Integration with Other AWS Services:

EC2 can integrate with other AWS services like RDS, SimpleDB, and SQS.

5) Security and Networking:

EC2 leverages Amazon Virtual Private Cloud (VPC) for security and offers features like Enhanced Networking for low latency and high throughput.

6) Global Infrastructure:

EC2 instances can be deployed in multiple locations (Regions and Availability Zones) for redundancy and low latency.

Parul®
University

DIGITAL LEARNING CONTENT

Amazon EC2 Overview

7) High-Precision Time:

The Amazon Time Sync Service provides a reliable time source for EC2 instances.

8) Visual:

A screenshot of the EC2 dashboard can be used to visually represent the features and functionalities of the service.

Parul®
University

DIGITAL LEARNING CONTENT

EC2 Dashboard Screenshot

aws Services Resource Groups [redacted] N. Virginia Support

New EC2 Experience
Tell us what you think

- EC2 Dashboard New
- Events New
- Tags
- Limits
- Instances
  - Instances
  - Instance Types
  - Launch Templates
  - Spot Requests
  - Savings Plans
  - Reserved Instances
  - Dedicated Hosts New
  - Scheduled Instances
  - Capacity Reservations
- Images
  - AMIs
- Elastic Block Store
  - Volumes
  - Snapshots
  - Lifecycle Manager

Resources

You are using the following Amazon EC2 resources in the US East (N. Virginia) Region:

Running instances	1	Elastic IPs	1
Dedicated Hosts	0	Snapshots	15
Volumes	3	Load balancers	0
Key pairs	9	Security groups	8
Placement groups	0		

Easily size, configure, and deploy Microsoft SQL Server Always On availability groups on AWS using the AWS Launch Wizard for SQL Server. Learn more

Launch instance

To get started, launch an Amazon EC2 instance, which is a virtual server in the cloud.

Launch instance

Note: Your instances will launch in the US East (N. Virginia) Region

Account attributes

Supported platforms
- VPC

Default VPC
vpc-d83db0a1

Settings
- EBS encryption
- Zones
- Default credit specification
- Console experiments

Explore AWS

Save up to 90% on EC2 with Spot Instances

Optimize price-performance by combining EC2 purchase options in a single EC2 ASG. Learn more

Enable Best Price-Performance with AWS Graviton2

AWS Graviton2 powered EC2 instances enable up to 40% better

Parul®
University

DIGITAL LEARNING CONTENT

Amazon EC2 Instance Types

Amazon EC2 instance types: the t2.micro instance, which is eligible for the AWS Free Tier, and the m5.large instance, which is a general-purpose instance

t2.micro:

- **This is a small, burstable performance instance, meaning it provides a baseline level of CPU performance but can burst to higher levels when needed.**
- **It is eligible for the AWS Free Tier, which offers free usage for a certain number of hours per month.**
- **These instances are well-suited for web servers, code repositories, and other workloads that don't require constant high CPU usage.**

Amazon EC2 Instance Types

m5.large:

- **This is a general-purpose instance, offering a balance of compute, memory, and networking resources.**
- **It is suitable for a variety of workloads, including web applications, microservices, and database servers.**
- **The m5 family is known for its strong performance, making it a good choice for applications with moderate CPU and memory requirements.**

Parul®
University

DIGITAL LEARNING CONTENT

Amazon EC2 General Purpose Instances

- **General-purpose instances, like the m5.large, provide a good balance of compute, memory, and networking resources.**
- **They are well-suited for diverse workloads, including web servers, code repositories, and applications that don't have a specific need for high CPU or memory.**
- **Examples of general-purpose instance families include m5, m6, T3, and T4g.**

Parul®
University

DIGITAL LEARNING CONTENT

Amazon EC2 Burstable Performance Instances

- **Burstable performance instances, like the t2.micro, offer a baseline level of CPU performance with the ability to burst above that baseline.**
- **They are a cost-effective option for workloads that don't require constant high CPU usage, but may need to burst to handle spikes in demand.**
- **Examples of burstable performance instance families include T2 and T3.**

Parul®
University

DIGITAL LEARNING CONTENT

AWS instance type table.

Instance Size	vCPU	Memory (GiB)	Instance Storage (GiB)	Network Bandwidth (Gbps)***	EBS Bandwidth (Mbps)
m5n.large	2	8	EBS-Only	Up to 25	Up to 4,750
m5n.xlarge	4	16	EBS-Only	Up to 25	Up to 4,750
m5n.2xlarge	8	32	EBS-Only	Up to 25	Up to 4,750
m5n.4xlarge	16	64	EBS-Only	Up to 25	4,750
m5n.8xlarge	32	128	EBS Only	25	6,800
m5n.12xlarge	48	192	EBS-Only	50	9,500
m5n.16xlarge	64	256	EBS Only	75	13,600
m5n.24xlarge	96	384	EBS-Only	100	19,000
m5n.metal	96*	384	EBS-Only	100	19,000
m5dn.large	2	8	1 x 75 NVMe SSD	Up to 25	Up to 4,750
m5dn.xlarge	4	16	1 x 150 NVMe SSD	Up to 25	Up to 4,750
m5dn.2xlarge	8	32	1 x 300 NVMe SSD	Up to 25	Up to 4,750
m5dn.4xlarge	16	64	2 x 300 NVMe SSD	Up to 25	4,750
m5dn.8xlarge	32	128	2 x 600 NVMe SSD	25	6,800
m5dn.12xlarge	48	192	2 x 900 NVMe SSD	50	9,500
m5dn.16xlarge	64	256	4 x 600 NVMe SSD	75	13,600
m5dn.24xlarge	96	384	4 x 900 NVMe SSD	100	19,000
m5dn.metal	96*	384	4 x 900 NVMe SSD	100	19,000

Elastic Load Balancing (ELB)

- **Elastic Load Balancing (ELB) on AWS offers several types, with Application Load Balancers (ALBs) handling HTTP/HTTPS traffic and Network Load Balancers (NLBs) managing TCP connections efficiently.**
- **ALBs can perform content-based routing, while NLBs focus on low latency and high throughput for TCP/UDP traffic**

Application Load Balancers (ALBs):

Function:

- **Operates at the application layer (Layer 7) of the OSI model, routing traffic based on request content like HTTP headers or URL paths.**

Uses:

- **Ideal for web applications, APIs, and scenarios requiring advanced routing logic.**

Protocols:

- **Primarily supports HTTP and HTTPS, offering advanced routing and content-based routing capabilities.**

Elastic Load Balancing (ELB)

Network Load Balancers (NLBs):

Function:

- **Operates at the network layer (Layer 4) of the OSI model, distributing traffic based on IP protocol data (TCP/UDP).**

Uses:

- **Suitable for high-performance, low-latency applications and scenarios needing maximum throughput, like game servers or streaming services.**

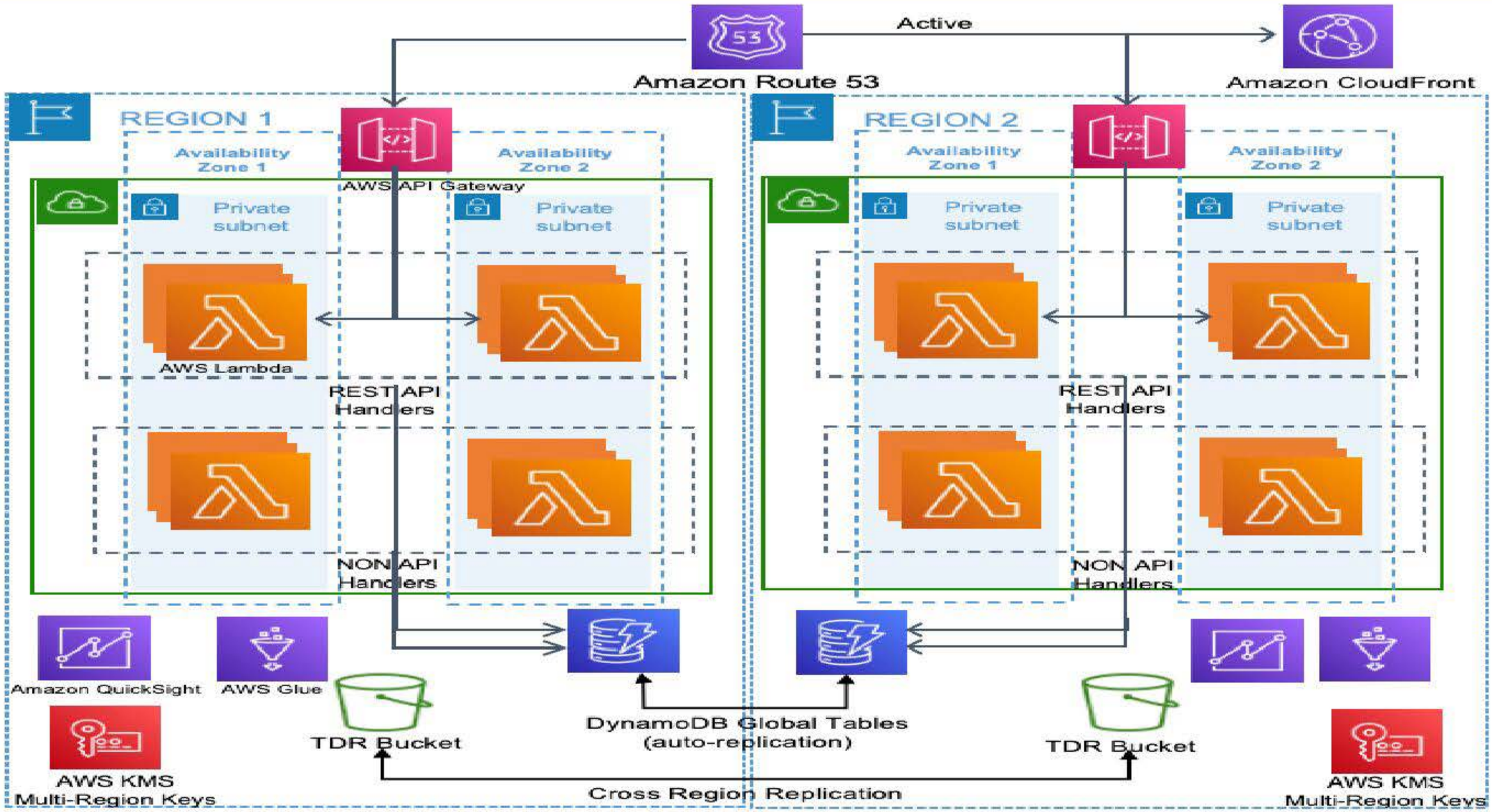
Protocols:

- **Supports TCP and UDP, providing faster and more efficient traffic distribution.**

Parul®
University

DIGITAL LEARNING CONTENT

ELB architecture diagram.



Parul®
University

DIGITAL LEARNING CONTENT

Amazon EC2 Auto Scaling

Auto scaling automatically adjusts EC2 capacity in response to changes in demand, like traffic spikes or lulls. This allows you to handle increased traffic without manually provisioning more resources and reduce costs when demand drops

Definition:
Auto scaling is a cloud computing feature that automatically adjusts the number of virtual machines (EC2 instances) or other resources to match the current workload.

Use Case:
One of the primary use cases is to handle traffic spikes. When an application experiences a sudden increase in traffic, auto scaling can automatically launch more EC2 instances to handle the load, preventing performance issues or downtime.

Amazon EC2 Auto Scaling

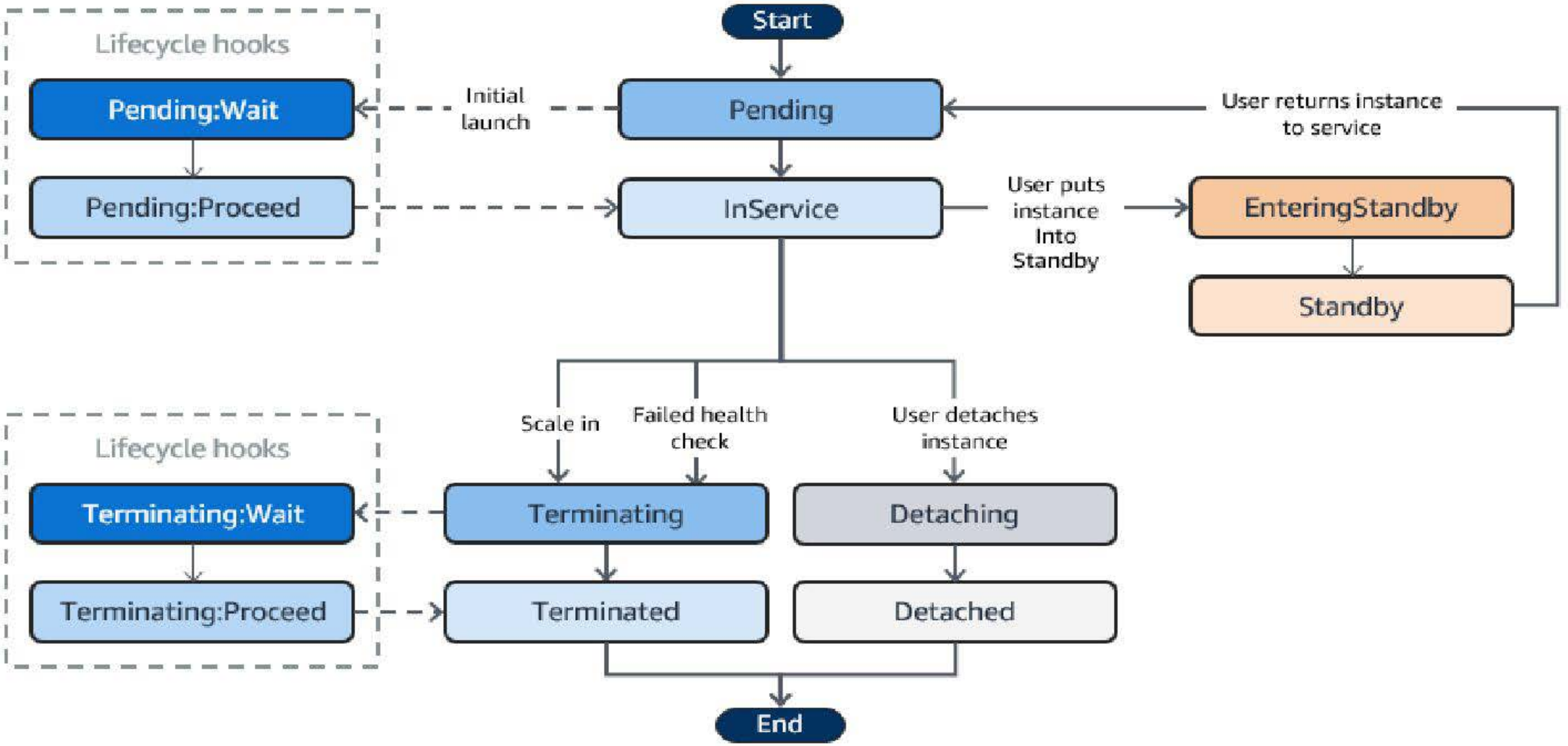
Benefits:

- **Cost Optimization: It allows you to pay only for the resources you need, reducing costs when demand is low.**
- **Improved Performance: By automatically scaling up during high demand, auto scaling ensures consistent application performance.**
- **Reduced Manual Work: It eliminates the need to manually manage resources, freeing up time and resources for other tasks.**

Parul®
University

DIGITAL LEARNING CONTENT

Amazon EC2 Auto Scaling Life Cycle Diagram



Parul®
University

DIGITAL LEARNING CONTENT

Launch EC2

To launch an EC2 instance, start by selecting the Amazon Machine Image (AMI) with Amazon Linux 2. Then, choose the instance type "t2.micro" and configure the security group to allow SSH (port 22) and HTTP (port 80) traffic

Detailed Steps:

1. Choose AMI (Amazon Linux 2):
In the AWS Management Console, navigate to EC2 and click "Launch Instance."
Select the AMI with Amazon Linux 2 from the "Choose an AMI" section.

2. Select Instance Type (t2.micro):
Choose "t2.micro" from the available instance types.

Amazon EC2 Auto Scaling

3. Configure Security Group (Allow SSH/HTTP):

- **On the "Configure Security Group" step, create a new security group or select an existing one.**
- **Add a new rule to allow SSH traffic on port 22.**
- **Add another rule to allow HTTP traffic on port 80.**

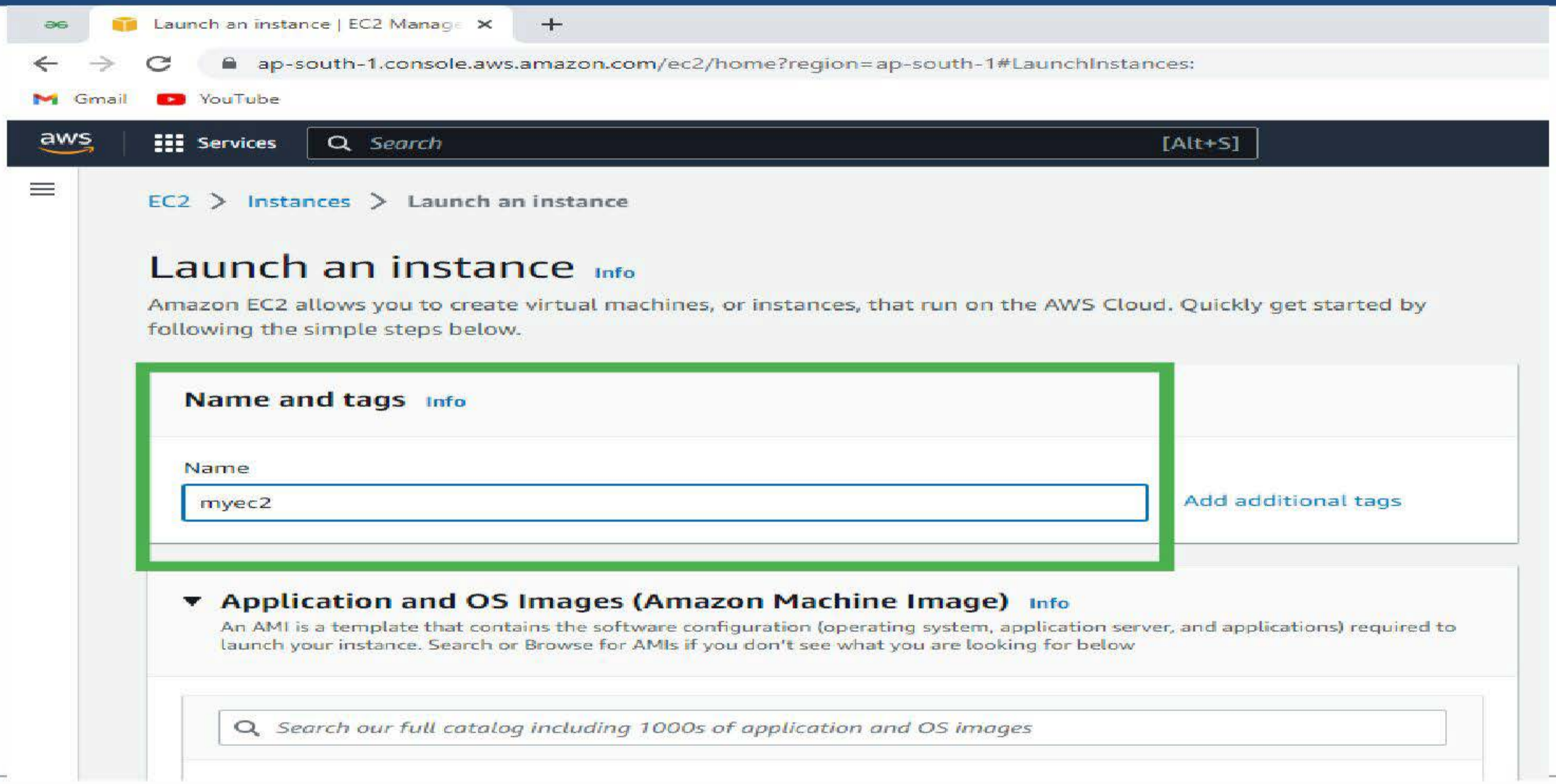
4. Launch Instance:

Review your settings and click "Launch Instance" to launch your EC2 instance

Parul®
University

DIGITAL LEARNING CONTENT

EC2 Launch Screenshot



Parul®
University

DIGITAL LEARNING CONTENT

Serverless Computing (AWS Lambda)

Serverless computing is a cloud computing execution model where developers build and run applications without managing the underlying servers. It's a pay-as-you-go service where a cloud provider handles infrastructure provisioning, management, and scaling, and you only pay for the resources consumed during function execution.

While it's called "serverless," it doesn't mean there are no servers; instead, the server infrastructure is abstracted away from the developer, making it seem like there are none.

Key aspects of Serverless:

No Server Management:
Developers focus on writing code and deploying it, while the cloud provider handles all server-related tasks.

Pay-as-you-go Pricing:
Users pay only for the resources they use during function execution, making it

cost-effective.
Automatic Scaling:
Serverless applications automatically scale up or down based on demand, ensuring optimal performance.

Function-as-a-Service:
Applications are broken down into small, independent functions that are executed in response to events.

Key aspects of Serverless:

Stateless Functions:

Serverless functions are typically stateless, meaning they don't retain data between invocations, requiring external storage or databases for state management.

Pros and challenges of Serverless:

Benefits of Serverless:

Reduced infrastructure management overhead: Developers can focus on coding without managing servers.

Cost savings: Pay-as-you-go model reduces infrastructure costs.

Scalability: Automatic scaling ensures performance under varying load.

Increased agility: Easier to deploy and update applications.

Challenges of Serverless:

Vendor lock-in: Can be challenging to migrate to another cloud provider.

Statelessness: Requires careful consideration of state management.

Complexity: Some aspects, like debugging and integration, can be challenging.

Cold starts: Initial function execution can be slower than traditional applications.

Parul®
University

DIGITAL LEARNING CONTENT

AWS Lambda

AWS Lambda is a powerful serverless computing service that automatically runs code in response to events, without requiring you to manage the underlying infrastructure.

It supports event-driven applications triggered by events such as HTTP requests, DynamoDB table updates, or state transitions. You simply upload your code (as a .zip file or container image), and Lambda handles everything from provisioning to scaling and maintenance.

It automatically scales applications based on traffic, handling server management, auto-scaling, security patching, and monitoring.

AWS Lambda is ideal for developers who want to focus on writing code without worrying about infrastructure management.

Parul®
University

DIGITAL LEARNING CONTENT

AWS Lambda

Parul®
University

DIGITAL LEARNING CONTENT

AWS Lambda

AWS lambda are server-less compute functions are fully managed by the AWS where developers can run their code without worrying about servers. AWS lambda functions will allow you to run the code with out provisioning or managing servers.

Example:
Process data from Amazon S3 buckets.
Respond to HTTP requests.
Build serverless applications.

Parul®
University

DIGITAL LEARNING CONTENT

Use Cases of AWS Lambda Functions

File Processing: AWS lambda can be triggered by using simple storage services (S3). Whenever files are added to the S3 service Lambda data processing will be triggered.

Web Applications: You can combine both web applications and AWS lambda which will scale up and down automatically based on the incoming traffic.

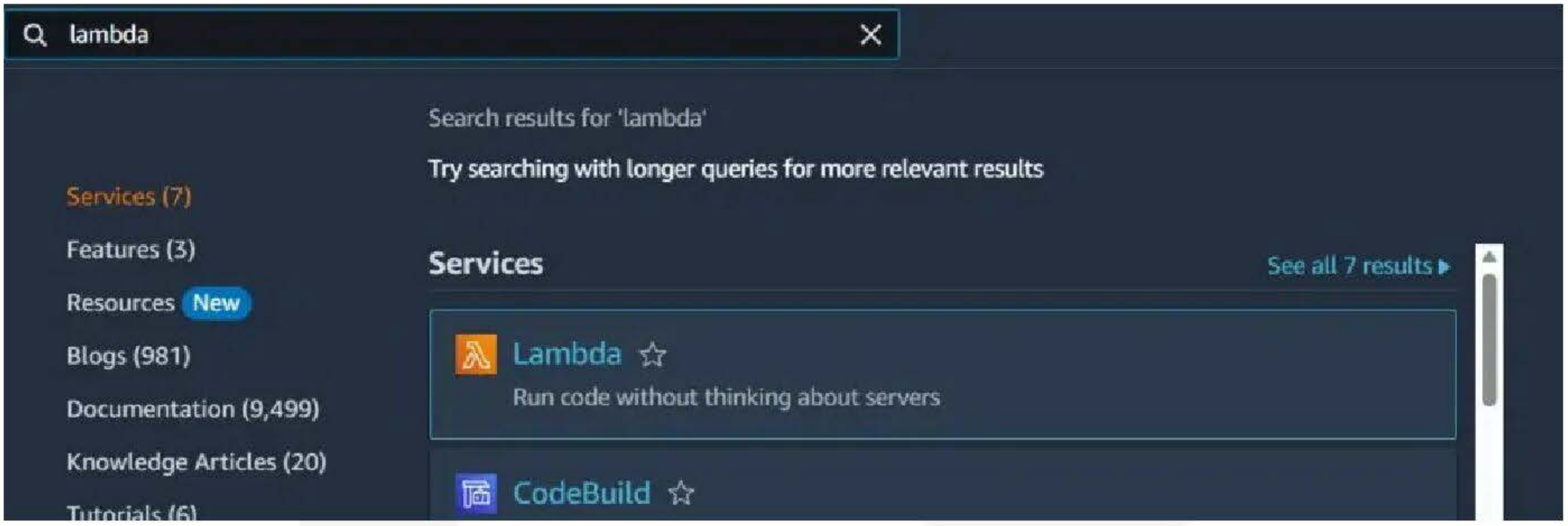
IoT (Internet of Things) applications: You can trigger the AWS lambda based on certain conditions while processing the data from the device which are connected to the IOT applications. It will analyze the data which are received from the IOT application.

Stream Processing: Lambda functions can be integrated with the Amazon kinesis to process real-time streaming data for application tracking, log filtering, and so on.

Parul®
University

DIGITAL LEARNING CONTENT

AWS Lambda Console



Parul®
University

DIGITAL LEARNING CONTENT

Characteristics of Lambda

AWS Lambda is a serverless compute service that lets you run code without managing servers. Key characteristics include:

No servers to manage:
AWS Lambda abstracts the underlying infrastructure. Users only need to upload their code, and AWS handles the rest, including server provisioning, scaling, and maintenance.

Automatic scaling:
Lambda automatically scales the execution of code in response to incoming requests. It can handle a few requests per day or thousands per second without requiring any manual configuration.

Parul®
University

DIGITAL LEARNING CONTENT

Characteristics of Lambda

Event-driven execution:
Lambda functions are triggered by events from various AWS services or external sources, such as changes to data in S3 buckets, messages in Kinesis streams, or HTTP requests from API Gateway.

Pay-per-use pricing:
Users are charged only for the compute time consumed by their code, with no charges when the code is not running. This makes it a cost-effective solution for applications with variable traffic patterns.

Built-in fault tolerance:
AWS Lambda is designed for high availability and fault tolerance. It automatically runs code in multiple availability zones to protect against infrastructure failures.

Parul®
University

DIGITAL LEARNING CONTENT

Step-by-Step Guide to Create Lambda + API Gateway

1) **Create the Lambda Function**
2) **Login to AWS Console**
3) **Navigate to AWS LambdaClick “Create function”**
   - **Name: helloWorldFunction**
   - **Runtime: Python 3.x**
   - **Choose “Author from scratch”**
   - **Role: Choose “Create a new role with basic Lambda permissions”**
4) **Click “Create function”**
5) **Click Deploy after entering the code.**

Parul®
University

DIGITAL LEARNING CONTENT

Step-by-Step Guide to Create Lambda + API Gateway

1) **Add an API Gateway Trigger**
2) **In the Lambda function console, click "+ Add trigger"**
3) **Select API Gateway**
4) **Choose:**
   - **API type: HTTP API or REST API**
   - **Security: Open (for testing) or choose AWS IAM/Auth**
5) **Click Add**
6) **Test the API**
   - **After adding the trigger, AWS will generate an API endpoint URL.**
   - **You can now test it:Copy the API endpoint URL**
   - **Paste it into your browser or test with curl:**

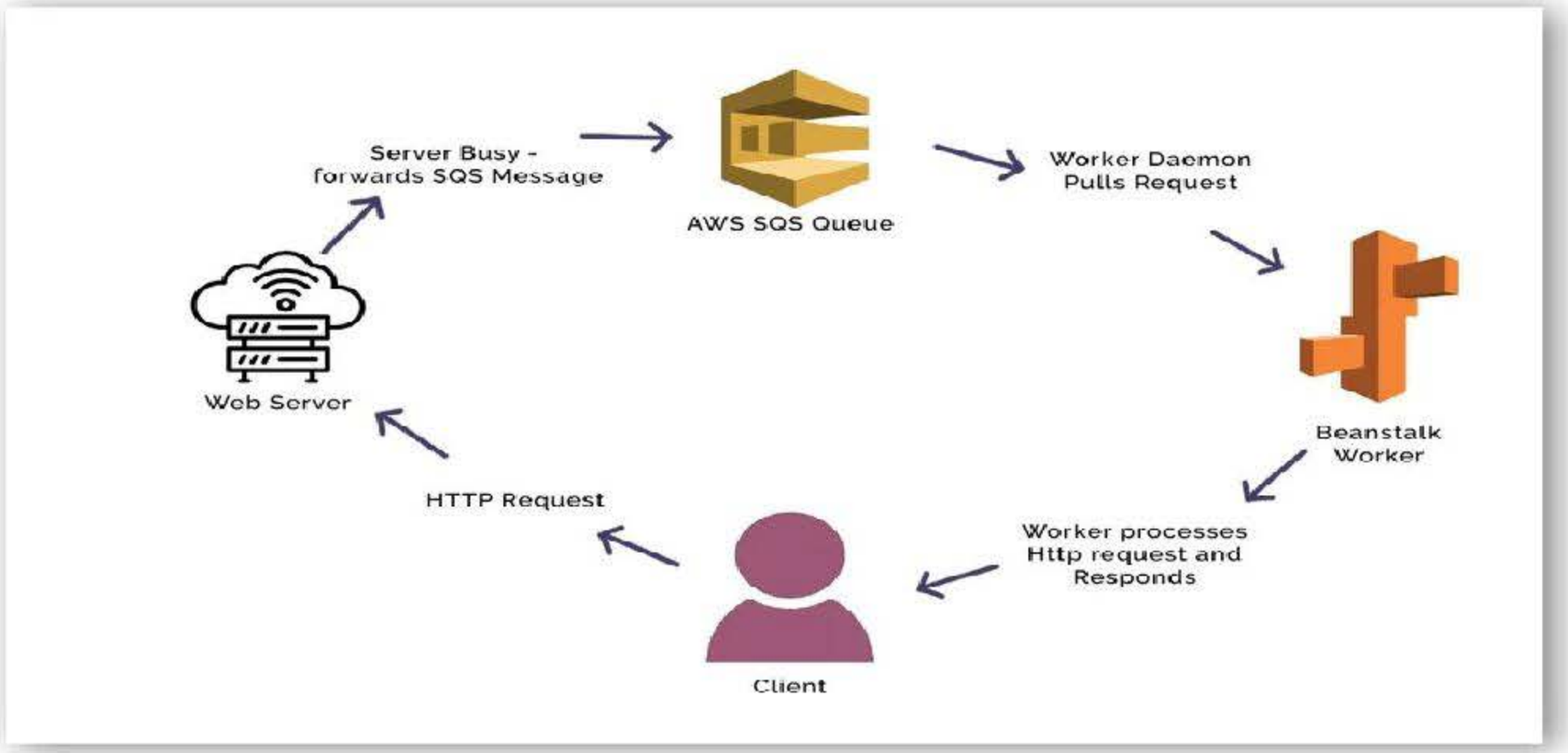
curl
https://your-api-id.execute-api.region.amazonaws.com/default/helloWorld
Function

Parul®
University

DIGITAL LEARNING CONTENT

PaaS-AWS Elastic Beanstalk

AWS Elastic Beanstalk is a Platform-as-a-Service (PaaS) that simplifies deploying and managing web applications and services on AWS.
It automates tasks like capacity provisioning, load balancing, auto-scaling, and application health monitoring, allowing developers to focus on building and deploying their applications without managing infrastructure.



Parul®
University

DIGITAL LEARNING CONTENT

PaaS-AWS Elastic Beanstalk – Key Features

Simplified Deployment:
Elastic Beanstalk automates the deployment process, handling tasks like provisioning instances, setting up load balancers, and configuring auto-scaling.

Focus on Application Development:
Developers can focus on building and deploying applications without worrying about underlying infrastructure management.

Scalability:
Elastic Beanstalk can automatically scale your application to handle traffic spikes, ensuring high availability and performance.
Support for Multiple Languages and Frameworks:
It supports popular web languages like Java, .NET, PHP, Node.js, Python, Ruby, Go, and Docker, as well as frameworks like Apache, Nginx, and IIS.

Parul®
University

DIGITAL LEARNING CONTENT

PaaS-AWS Elastic Beanstalk – Key Features

Automated Infrastructure Management:
Elastic Beanstalk manages the underlying infrastructure, including EC2 instances, load balancers, and auto-scaling groups.

Health Monitoring:
It continuously monitors the health of your application and instances, alerting you to any issues.

Parul®
University

DIGITAL LEARNING CONTENT

PaaS-AWS Elastic Beanstalk – Key Features

How it Works:

1. Upload your application:

You upload your application code to Elastic Beanstalk.

2. Select a Platform:

Choose a platform from a list of supported platforms, including Java, .NET, PHP, Node.js, Python, Ruby, Go, and Docker.

3. Configure Environment (Optional):

You can customize the environment settings, such as instance types, storage, and network configuration.

4. Elastic Beanstalk Handles the Rest:

The service automatically provisions resources, sets up load balancing, and manages auto-scaling based on your application's needs.

Parul®
University

DIGITAL LEARNING CONTENT

Deploy a sample Elastic Beanstalk

To deploy a sample application to Elastic Beanstalk, you'll create an environment, upload your application, and then launch the environment. Here's a step-by-step guide:

1. Create an Elastic Beanstalk Application and Environment:

- **Navigate to the AWS Elastic Beanstalk console.**
- **Click "Create new application".**
- **Provide an application name and optionally a description.**
- **Click "Create".**
- **Choose "Web Server" or "Worker" environment type.**
- **Provide an environment name, application source, and other desired settings like platform, instance type, etc.**
- **Click "Create environment"**

Deploy a sample Elastic Beanstalk

2. Prepare Your Application:

- **Ensure your application is properly packaged for deployment. This might involve creating a zip file, creating a Dockerfile, or other platform-specific packaging methods.**

3. Upload Your Application:

- **In the Elastic Beanstalk console, navigate to your application and environment.**
- **Click "Upload and deploy".**
- **Choose your application source (e.g., zip file).**
- **Upload the source code or application package.**

Deploy a sample Elastic Beanstalk

4. Deploy the Application:

- **Click "Deploy" to start the deployment process.**
- **Elastic Beanstalk will automatically provision and configure the necessary resources on AWS, like EC2 instances, load balancers, and security groups, based on your environment settings.**

Deploy a sample Elastic Beanstalk

5. Access Your Application:

- **Once the environment is healthy (status shows "Environment is healthy"), click the environment URL in the console.**
- **Your application should be accessible and running.**

DIGITAL LEARNING CONTENT

Parul® University

www.paruluniversity.ac.in