

Experiment No 1

- Aim: To understand the benefits of cloud infrastructure & setup AWS Cloud9 IDE. Launch AWS Cloud9 IDE & perform collaboration demonstration.
- Theory:

Q1) What is AWS?

→ AWS stands for Amazon Web Services. AWS is a cloud computing platform that offers a range of services to help organisations & individuals grow. It provides flexible, scalable & cost effective IT infrastructure that can be accessed over the internet on a pay-as-you-go model.

Q2) Features of AWS

- 1) Compute power: AWS provides scalable virtual servers through EC2 (Elastic Compute Cloud) & managed serverless services like Lambda.
- 2) Storage: Offers services like S3 (Simple Storage Service) for scalable object storage & Glacier for low-cost archival storage.
- 3) Databases: Provides managed databases through RDS for SQL databases & DynamoDB for NoSQL databases.
- 4) Networking: AWS offers VPC (Virtual Private Cloud) for secure networking, Elastic Load Balancing & AWS Direct Connect for private network connections.
- 5) Security & Compliance: Security tools like AWS Identity & Access Management (IAM), Shield & compliance certifications for industry standards.
- 6) Developer Tools: Services like CodeDeploy, CodeBuild & CodePipeline for automating software development & deployment processes.

Q3) Uses of AWS.

- i) Website Hosting: AWS enables hosting websites & web applications with services like EC2, S3, & RDS.
- 2) Data Storage & Backup: Scalable storage solutions provide efficient data storage, backup & disaster recovery.
- 3) App Development: AWS supports app development with services like Lambda & AWS Amplify.
- 4) Data Analytics & Machine Learning: Services such as Redshift & SageMaker help businesses
- 5) Content Delivery: AWS CloudFront is used for delivering content globally with low latency.
- 6) Game Development: AWS offers tools for game developers to build, host & scale multiplayer games globally.

Q4) Advantages of AWS.

- i) Scalability: AWS allows businesses to scale up or down based on demand with auto-scaling capabilities.
- 2) Cost-Effectiveness: The pay-as-you-go pricing model helps business reduce upfront costs, paying only for what they use.
- 3) Global reach: AWS has data centers worldwide, providing low-latency access for customers across the globe.
- 4) Security: AWS provides a range of security features, include encryption, compliance certifications & DDoS protection.

→ Conclusion: we could understand the benefits of cloud infrastructure & setup AWS Cloud9 IDE.

Experiment No 2

- Aim: To build your application using AWS code build & deploy on S3 using AWS code pipeline.
 - Theory:
- Q1) What is IAM? How to create an IAM User (AWS management console)?
- AWS IAM (Identity & Access Management) is a service that helps manage access to AWS resources securely. It allows you to create & manage users, groups, roles & permissions, controlling who can access specific resources & services. With IAM, you can define & enforce granular policies that specify what actions users or services can perform on AWS resources.

How to create IAM user :

- 1) Sign in to AWS Management console.
- 2) Navigate to IAM.
- 3) Select 'Users' & then select the "Add User" button.
- 4) Provide User Details
- 5) Set Permissions
- 6) Add metadata tags to the user
- 7) Review the settings & then click "Create User"

Q2) Features of IAM?

- 1) Granular Permissions: IAM allows you to control permissions on a per-resource, per-action basis, offering detailed access control.
- 2) User management: You can create users with individual credentials & permissions.

- 3) Role-Based Access Control: Roles allow for temporary access to AWS resources, commonly used for cross-account access or service-level permissions
- 4) Password policies: IAM allows you to enforce password strength requirements & expiration policies

Q3) What is AWS Account Root User?

→ The AWS Account Root User is the original user created when you set up your AWS account. This user has full, unrestricted access to all AWS services & resources within the account, making it a high privilege user. Only use the root user for tasks that require it.

Q4) How do we create a Group (AWS Management console)?

- 1) Sign into the AWS Management Console
- 2) Open the IAM console
- 3) Select groups from the navigation pane
- 4) Select Create New Group
- 5) Enter a name for the group
- 6) Select Next Step.
- 7) Select the check box next to the policy you want to use
- 8) Select Next Step again
- 9) Select Create Group.

→ Conclusion: Hence we have successfully deployed code using S3 bucket.

Experiment No 3

→ Aim : Configure a bucket to host a static website & securely share a bucket object by using a resigned URL

→ Theory:

Q1) what is S3? Advantages of Amazon S3

→ Amazon S3 (Simple Storage Service) is an object storage service that offers industry-leading scalability, data availability, security & performance. It offers scalable, secure & durable cloud storage, allowing you to store & retrieve any amount of data at any time from anywhere.

• Advantages of S3

- 1) Scalability: S3 can handle any volume of data, w/o any limit on the size of storage.
- 2) Durability: S3 is designed to provide 99.999999999% (11 nines) of durability, ensuring data is protected against failure.
- 3) Availability: It offers high availability for access to your data.
- 4) Security: S3 provides built-in encryption & integrates with AWS IAM for secure data access control.

Explain

Q2) Amazon S3 concepts - Buckets, Objects, Keys, Regions & Data consistency Model.

→ 1) Buckets: A bucket is a container for objects stored in Amazon S3. You can store any no. objects in a bucket & can have upto 100 buckets in your account.

2) Objects: Objects are fundamental entities stored in Amazon S3. Objects consists of object data & metadata. An object is uniquely identified within a bucket by a key (name) & a Version ID.

3) Keys: An object key (or key name) is the unique identifier for an object within a bucket. Every object has exactly one key. The combination of a bucket, object key & optionally, version ID uniquely identify each object.

4) Regions: Amazon S3 buckets are created in specific AWS regions (geographical locations). Data is stored in regions where the bucket is created, ensuring low latency & compliance with local regulation.

• Data Consistency Model:

1) Strong Read-After-Write consistency: When you upload a new object, S3 immediately reflects this change across all regions, so subsequent read requests will retrieve the latest data.

2) Eventual consistency: For operations like overwrites (PUT or DELETE), there may be a slight delay for consistency across all regions.

Q3) How to create an S3 bucket?

→ 1) Login to the AWS management console

2) Navigate to Amazon S3

3) Click on 'Create Bucket' button - Provide a unique bucket name

4) Configure Bucket options - Versioning, Logging, Encryption, Tags

5) Set Permissions -

6) Review & Create - After configuring the options, review your settings & click "Create Bucket"

→ Conclusion: Hence we have successfully hosted a static website using AWS

Experiment No 4

→ Aim: launch & connect to your EC2 instance by using AWS System Manager.

→ Theory:

Q1) What is EC2?

→ Amazon EC2 (Elastic Compute Cloud) is a web service provided by AWS that offers scalable computing capacity in the cloud. EC2 allows users to run virtual servers, called instances w/o having to invest in physical hardware.

Key features

- 1) Scalability: Automatically scales up or down based on traffic & resource needs.
- 2) Elasticity: Quickly handles capacity to handle changing demands.
- 3) Security: EC2 instances are located within secure, isolated environments using security groups, network ACL's & VPC's.

Q2) Types of Reserved Instances.

→ Reserved Instances in EC2 allows you to make a commitment to AWS for a 1 or 3-year term to receive a discount compared to On-demand pricing.

Types

1) Standard Reserved Instance:

2) Convertible Reserved Instance.

A standard reserved instance provides a more significant discount than a convertible Reserved Instance (RI).

You can exchange convertible RI but can't exchange a standard RI. Standard RI can be sold in the Reserved Instance marketplace. Convertible RI can't be sold.

Standard RI are best for predictable workloads with

a steady state of usage. Convertible RI's are suitable if your needs might change over time but you still want to save money.

Q3) How to create an EC2 instance?

- 1) Login to AWS management console.
- 2) In the AWS services menu, search & select "EC2".
- 3) Click on 'Launch Instance' to start creating new instance.
- 4) Choose an Amazon Machine Image (AMI) such as Amazon Linux 2, Ubuntu, Windows Server etc.
- 5) Choose an instance type based on your performance & resource needs.
- 6) Configure Instance Details.
- 7) Add storage.
- 8) Add tags.
- 9) Configure Security Groups.
- 10) Review & launch.
- 11) Access your instance.

→ Conclusion: Hence we successfully launched & connected an EC2 instance.

Experiment No 5

→ Aim: Create & Deploy windows EC2 instance & make that instance as a web server.

→ Theory:

Q1) What is EBS?

→ Amazon EBS (Elastic Block Store) provides scalable, high-performance block storage resources that can be used with Amazon EC2 instances.

Features of EBS

1) Multiple volume types :

2) Scalability

3) Backup & recovery

4) Data protection

Q2) What are the EBS volume types?

→ 1) Solid state drive (SSD) volumes : SSD-backed volumes are optimized for transactional workloads involving frequent read/write operations with small I/O size, where the dominant performance attribute is IOPS.

SSD-backed volume types include General Purpose SSD & Provisioned IOPS SSD.

2) Hard disk drive (HDD) volumes : HDD-backed volumes are optimized for large streaming workloads where the dominant performance attribute is throughput. HDD volume types include Throughput Optimized HDD & Cold HDD.

→ Q3) Explain HDD?

→ HDD (Hard Disk Drive) volumes in EBS are designed for throughput-intensive, large, sequential workloads where data is processed in large chunks. These volumes are cost-effective for workloads that do not require high IOPS but need high throughput, making them ideal for tasks like big data analytics etc.

HDD-based EBS volumes

- 1) Throughput Optimized HDD
- 2) Cold HDD.

→ Conclusion: Hence we have successfully deployed Windows EC2 instance.

Experiment No 6

→ Aim: To understand AWS Lambda, its workflow, various function & create your first lambda function using python

→ Theory:

Q1) What is Lambda?

→ Lambda runs your code on high-availability compute infrastructure & performs all of the administration of the compute resources, including server & OS maintenance, capacity provisioning & automatic scaling, & logging. You organize your code into Lambda functions. The Lambda service runs your f" only when needed & scales automatically.

Q2) How does Lambda work?

→ AWS Lambda f" are triggered by events.

The basic process is:

- Event source: A change in state or update in AWS triggers the Lambda function
- Lambda function: AWS Lambda runs the code you provide in response to the event.
- Execution Environment: Lambda runs your code in a managed, isolated environment.
- Scaling: Lambda automatically scales to handle requests by running as many instances of the f" as needed in parallel based on the incoming traffic.

Q3) How to create a Lambda function?

- Open the AWS Lambda console
- Click 'Create Function'
- Choose Author from scratch
- Enter a function name

- 5) Select a runtime
- 6) choose an execution
- 7) Write & Upload your code
- 8) Configure triggers
- 9) Test your function
- 10) Deploy & Monitor

Q4) How is lambda priced?

→ In lambda you are charged based on the number of requests for your functions & the duration it takes for your code to execute. Lambda counts a request each time it starts executing in response to an event notification trigger such as from Amazon SNS. Duration is calculated from the time your code begins executing until it returns or otherwise terminates, rounded up to the nearest 1ms*. The price depends on the amt of memory you allocate to your fⁿ.

→ Conclusion: Hence we have successfully understood AWS lambda functions & created one using python.

Experiment No 7

→ Aim: To create a lambda function which will log "An object has been added" once you add an object to a specific bucket in S3 by using email service.

→ Theory:

Q) Why is Lambda cool?

→ AWS Lambda is considered cool because it enables serverless computing, which allows developers to focus on writing code w/o worrying about managing infrastructure. Key reasons why Lambda is appreciated include:

- No Server Management: You don't need to provision, manage or scale servers, AWS automatically takes care of that.
- Event driven execution: Lambda runs your code in response to events, such as HTTP request or a file being uploaded to Amazon S3.
- Scalability: It automatically scales up or down based on the demand, so you don't have to worry about scale infrastructure.
- Integration with other AWS services: Lambda integrates with a wide range of AWS services like API Gateway, DynamoDB making it highly flexible for building cloud-native apps.

Q) What is a Data Centre?

→ A Data Centre is a facility that houses computing resources such as servers, storage systems, networking equipment, & infrastructure required to run IT operations & store, process or distribute data. They are critical for supporting online services, applications & data storage for business & organizations.

Key aspects of data centres include:

- Power supply: Redundant power sources to ensure uptime
- Cooling systems: To prevent hardware from overheating
- Networking equipment: Routers, switches & firewalls for connectivity & security
- Security: Physical & cybersecurity measures to protect data & equipment.

Q3) What is IaaS?

→ Infrastructure as a Service (IaaS) is a form of cloud computing that provides virtualized computing resource over the internet. IaaS allows business to rent infrastructure - such as servers, storage, & networking - on demand from cloud service providers rather than buying & maintaining their own.

Key benefits:

- 1) Cost savings
- 2) Scalability
- 3) Flexibility
- 4) Managed infrastructure.

→ Conclusion: Hence we have successfully created an Lambda function which logs "An object has been added" by using email service.

Experiment No 8

→ Aim : Create a group in AWS with administrative permissions & add & remove users from that group.

→ Theory:

Q1) Steps to create Group in AWS with Admin?

- 1) Go to the AWS management console
- 2) Sign in with your credentials
- 3) In the AWS management console, search for 'IAM' in the search bar & click on IAM
- 4) On the left sidebar of the IAM dashboard, click on 'User groups'
- 5) Click on the 'Create group' button.
- 6) Enter a name for your group.
- 7) Assign Permissions & attach policies to the group.
- 8) Review & Create Group

Q2) Steps to add & remove users from that group.

→ Steps to add users to the group.

- 1) Go to the IAM Dashboard & click on 'User groups' from the left sidebar
- 2) Click on the group you created.
- 3) Click on the 'Add Users' button.
- 4) A list of users will appear. Select the users you want to add.
- 5) Click 'Add Users'

To remove users from the group

- 1) Go to the IAM Dashboard & click on 'User Groups' from the left sidebar.
- 2) Click on the group name.
- 3) In the Users tab of the group, you'll see the list of users in that group.
- 4) Select the users you want to delete.
- 5) Click the 'Remove Users' button.

→ Conclusion: Hence we have successfully created a group in AWS & gave it some permissions & then removed users from the group.

Experiment No 9

→ Aim : To understand terraform lifecycle, core concepts / terminologies & install it on Linux Machine

→ Theory:

Q1) How does Terraform Work?

→ Terraform operates with 2 major components :

- 1) **Terraform core**: It takes the user-provided configuration files & the Terraform-managed state file. The core processes these inputs to create a resource graph, determining how resources relate to each other & what needs be created, updated or destroyed. It handles the entire lifecycle.
- 2) **Providers**: Terraform supports various providers, including cloud providers like AWS, GCP & Azure, as well as on-premise infrastructures like VMWare & OpenStack.

Q2) What are the core concepts of terraform?

- 1) **Variables** - it is a key-value pair used by Terraform modules to allow customization.
- 2) **Provider** - it is a plugin to interact with APIs of service.
- 3) **Module** - it's a folder with terraform templates where all the configurations are defined.
- 4) **State** - it consists of cached information about the infrastructure managed by Terraform & its related configurations.
- 5) **Data Source**: It is implemented by providers to return information on external objects to terraform.
- 6) **Output values**: These are return values of a terraform module that can be used by other configurations.

Q3) Write down the steps to install Terraform on Ubuntu 18.04.

- 1) To download Terraform go to the website '<https://www.terraform.io/downloads.html>' Select appropriate package for your OS & architecture.
- 2) unzip the archive file
- 3) Change the directory to an unzipped folder.
- 4) To check whether terraform is installed run : `terraform -v`

→ Conclusion: Therefore we understood terraform lifecycle, core concepts/technologies & install it on linux Machine.

Experiment No 10

→ Aim: Create S3 bucket using Terraform.

→ Theory:

Q1) Write steps to create S3 bucket using terraform:

- 1) Write a Terraform script in Atom for creating S3 bucket on AWS
- 2) Create a new provider .tf file
Save both the files in the same directory
- 3) Open cmd & go to the directory where the .tf files are stored
- 4) Execute terraform init command to initialize the resources
- 5) Execute terraform plan to see the available resources
- 6) Execute terraform apply to apply the configuration, which will automatically create an S3 bucket based on our configurations
- 7) Execute terraform destroy to delete the configuration, which will automatically delete an EC2 instance

→ Conclusion: Therefore we have successfully created S3 bucket using Terraform.