**Aws/Devops**

**Syllabus:-**

All the Basics:-

1. SDLC

2. Team Structure.

3. Basics of cloud computing

4. Basics of AWS/DEVOPS

5. Basic Linux OS

6. Shell scripting

AWS:- (We are going to learn below AWS services Such as)

1. EC2

2. VPC

3. S3

4. EFS

5. IAM

6. SNS

7. Cloud Watch

8. CLoud Trai-l

9. RDS

( With Theory and company use based practicals)

DEVOPS Tools:-

1. GIT/GITHUB

2. MAVEN

3. JENKINS

4. DOCKER

5. ANSIBLE

6. JIRA (Basic)

Some part of Advanced tools:-

1. Terraform

2. Kubernetes

**21/02/2024 Tues**

**AWS Definition-**

It is a public cloud platform by amazon which provides SAAS, PAAS, IAAS.

**Devops Defination-**

Devops is the culture to improve the delivery of application with automation ,Quality, Continuous Monitoring & Continuous Testing

**Software Definition:**

Set or group of multiple programs running together. Software Program/Program: Set of logical instructions**.**

So, a Software company focuses on creating, managing, delivering, Maintaining software.

**Types of IT companies in Market:**

**1. Product Based IT companies: -**

These companies create their own software products.

e.g.: Oracle, IBM, CISCO, MICROSOFT....

**2. Service Based IT companies: -**

These companies provide these services to other companies.

E.g.: Infosys, Accenture, Techm, Capgemini......

**SDLC (Software Development Lifecycle)**

This is nothing but a process which is followed in a company/project to develop

and deliver a software app/software product asap.

**Stages/Phases of SDLC:**

**1. Requirement Phase: (Planning Phase)**

**BA** collect the requirements from the customer/client as per there needs and create a proper requirement documentation then this requirement is shared within the internal project team**.**

**BA: Business Analyst:**

A person who interacts with the client.

**2. Analysis Phase:-(Planning Phase)**

Once the requirement is gathered, the next step is to analyse the customer requirement within the project team, and create an internal documentation.

**3. Design Phase:-(Planning Phase)**

**a. High Level Design:**

High level software Architecture

**b. Low Level Design:**

In detail software design, means we will design minute things such as which technology should we implement, which features should we implement etc etc

**4. Development Phase:**

Actual development/coding is started in this phase. In this phase the code which is developed is called as **Source Code**.

**5. Testing Phase:**

After development is done, a through testing is done. multiple testing teams work together and test our created software. there are multiple types of testing which is done.

**6. Deployment Phase:**

Here the created software is given/delivered to the customer. software is deployed in the customer env.

**7. Maintenance Phase:**

In case there is any issue related to software delivered to our client, and our client

raises any issue, in this phase we solve the client’s problem

**22/02/2024 wed**

**SDLC Models:**

**There are multiple types of SDLC models:**

**1. Waterfall Model**

**2. Spiral Model**

**3. V model**

**4. Agile**

**5. Hybrid**

**1. Waterfall Model**

**In case of Waterfall Model, the output of one phase is the input to the other phase**

**e.g.** the output of R and A phase is the input to the system design phase Without completion of a previous phase the next phase will not start once the previous/prior phase is completed then the next phase is started**.**

**Advantages of Waterfall Model:**

1. Simple and Easy to understand and use

2. Easy to Manage

3. Phases are completed and worked upon one at a time.

4. requirement and documentation are very well understood and noted down

5. Clearly Defined Stages

**Disadvantages of Waterfall model:**

**1. Time to Market:**

the software application or the product released to the market will take time

compared to other models. Time to market is very high

**2. High Amount of Risk:**

In case there is any issue found during a phase, then all the things would restart right from the beginning.

**for e.g**. If there is an issue found in the testing phase, then the issue are checked and analysed, redeveloped and then re-tested when would take time Hence for complex projects, with high business logic and analytics waterfall model is not recommended**.**

**3. Poor model for large or ongoing projects**

**4. it is not a feedback-based model.**

**Req+ Analyasis**

**Design phase**

**Devlopement phase**

**Testing phase**

**Deployment phase**

**Maintaining phase**

**23/02/2024 Thurse**

**2.Agile Model:**

**Meaning of Agile ---- Move Quickly**

A large Project is Broken down into smaller chunks/parts and then developed and released to the customer one by one in iteration. this model has a customer feedback-based approach

Once any Version is released to the customer, the customer will give feedback and then based on the feedback enhancements are done accordingly and our software is re-released.

**release ---> feedback ---> Enhancements ---> Re-release**

**Advantages of Agile Model:**

1. Time to Market is less

2. Customer Feedback Based Model

3. its suitable for complex projects

4. its suitable for ongoing projects

5. less risk

6. Frequent delivery of new changes.

7. if there are changes in requirements, then we can adapt those

changes in our project without any issue, or its smoothly done.

**Disadvantages:**

1. Less Documentation

2. complex structure compared to waterfall

3. Maintenance

**When Agile is implemented then we call that term as scrum. in Scrum there are multiple terms.**

**1. Scrum Team:**

This is nothing but a team which consist of multiple dev, testers, devops engg, BA and etc

**2. Scrum Master:**

The person leading the Scrum team is called Scrum Master and its job is to **for e.g**. take update, monitoring tasks of the member etc

**3. Scrum Ceremonies:**

**a. Planning Meeting:**

we plan all the roles and responsibilities and features in this meeting

**b. Daily Scrum Meeting:**

here daily we are connecting/collaboration with each other and discussing the project daily updates

**c. Review Meeting:**

All the new features developed are reviewed

**d. Retrospective Meeting:**

In case there was any hurdle or problem during the sprint cycle then we can discuss that and discuss on what steps we can implement to avoid issues in future**.**

**Virtual Machine:**

Virtual Machine is a machine which consists of OS, Storage, RAM, CPU etc and is accessed over the network. Hardware is not physically present in front of user.

**28/02/2024 Wed**

**Roles and Responsibilities of Devops Engg:**

1. Creating the virtual infra in cloud.

2. Managing/maintaining the virtual infra in cloud.

3. Monitoring the virtual infra in cloud.

4. Deploying the latest Software application in Non-production and Production envs

5. Implementing Latest tools and technologies to automate daily deployments and tasks.

**Environment:**

This consists of multiple virtual machines hosting a software application and server. here there are multiple types of envs**.**

**1. production env (PE):**

this is an env where client accesses the software application. software is deployed on this env in the deployment/delivery stage of SDLC. PE is a highly critical and IMP env and must have zero downtime**.**

**2. Tests Env.**

here in test envs, multiple different testing teams access these Virtual machines to test the software running/deployed on these machines if the application is running fine on tests envs and the testing done by the testers is successful then we can move the application to the production and release to the client

**Types of testing Envs:**

**1. Devoplement Env:**

here the development team will access this env to test dev changes.

**2. QA Env: (Quality Assurance/Analysis)**

this is a testing team env, here QA team (testing team) will test all the application features thoroughly. here multiple testers having diff skill set, some are manual tests, automation testers, ATT tests, API testers will test the application for them this env is created.

**3. UAT Env: (User Acceptance Testing)**

here UAT team will access/use this env for UAT testing, here in UAT testing client specific scenarios are tested. means how the application is used by client is checked.

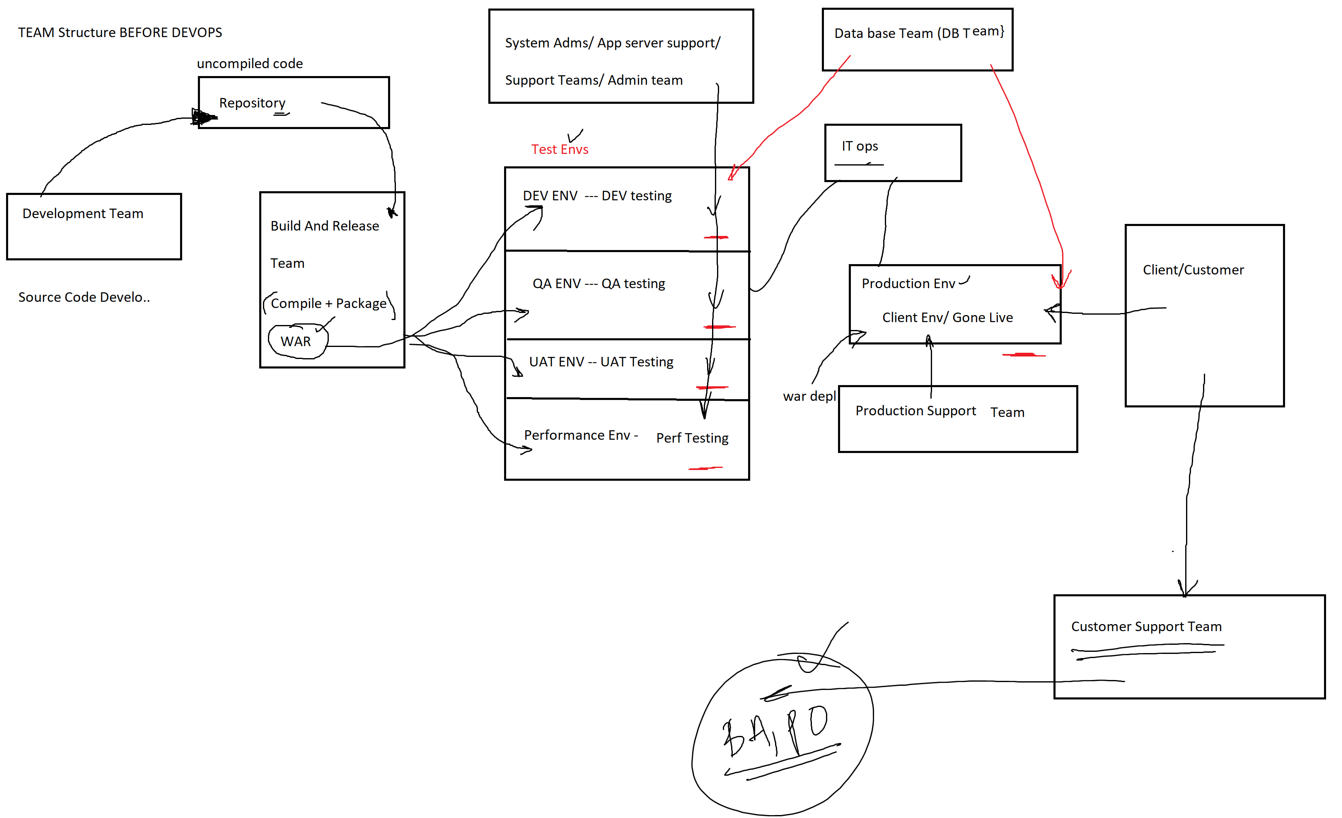
**4. Performance Env:**

here the application performance is tested by performance team.

- CPU utilization

- Load Testing and more

**29/02/2024 Thurse**

**Team Sructure Before Devops**

**Development Team:**

Source Code Development

**This Source code is then pushed to the repository**

**Repository:**

it is a centralized location for our source code, where multiple developers can collaborate

**Build and Release Team:**

**Compile-** Convert human readable code to machine readable code.

**the work of build and release team is to compile and package the source code to prescribed format**,

**for e.g**. here we are considering WAR package. This **War is then deployed on Multiple Tests Envs for testing purpose. Once testing is successful by Testing Env, then this war is pushed to Production Enc,** where the client will use/access the application.

As Production Env is a highly critical env, and needs zero downtime, hence to manage this env there is a **Production support team**. Production support team works on management of prod env.

**for e.g**. managing VMs, servers, software installations OS upgrade, OS installation, software updates and etc.

**System Admins/Support Team:**

This team basically work on management of test envs, like creating maintaining VMs, servers, OS, software’s, updates etc....

**By DB support Team (Database):**

DB is a software which has the application data. DB team is specifically working on management, creation, updating of DB in multiple diff envs.

**IT-Ops Team (IT operations):**

Basically, this team works on documentation, monitoring of all envs.

**Customer support team:**

Handle client queries

**Roles & Responcibilities of Devops Eng**(after devops)

*1. Source Code Management*

*2. Build and release*

*3. Deployment of software application on multiple envs*

*4. Database management and monitoring*

*5. Report generation*

**AFTER DEVOPS Team:**

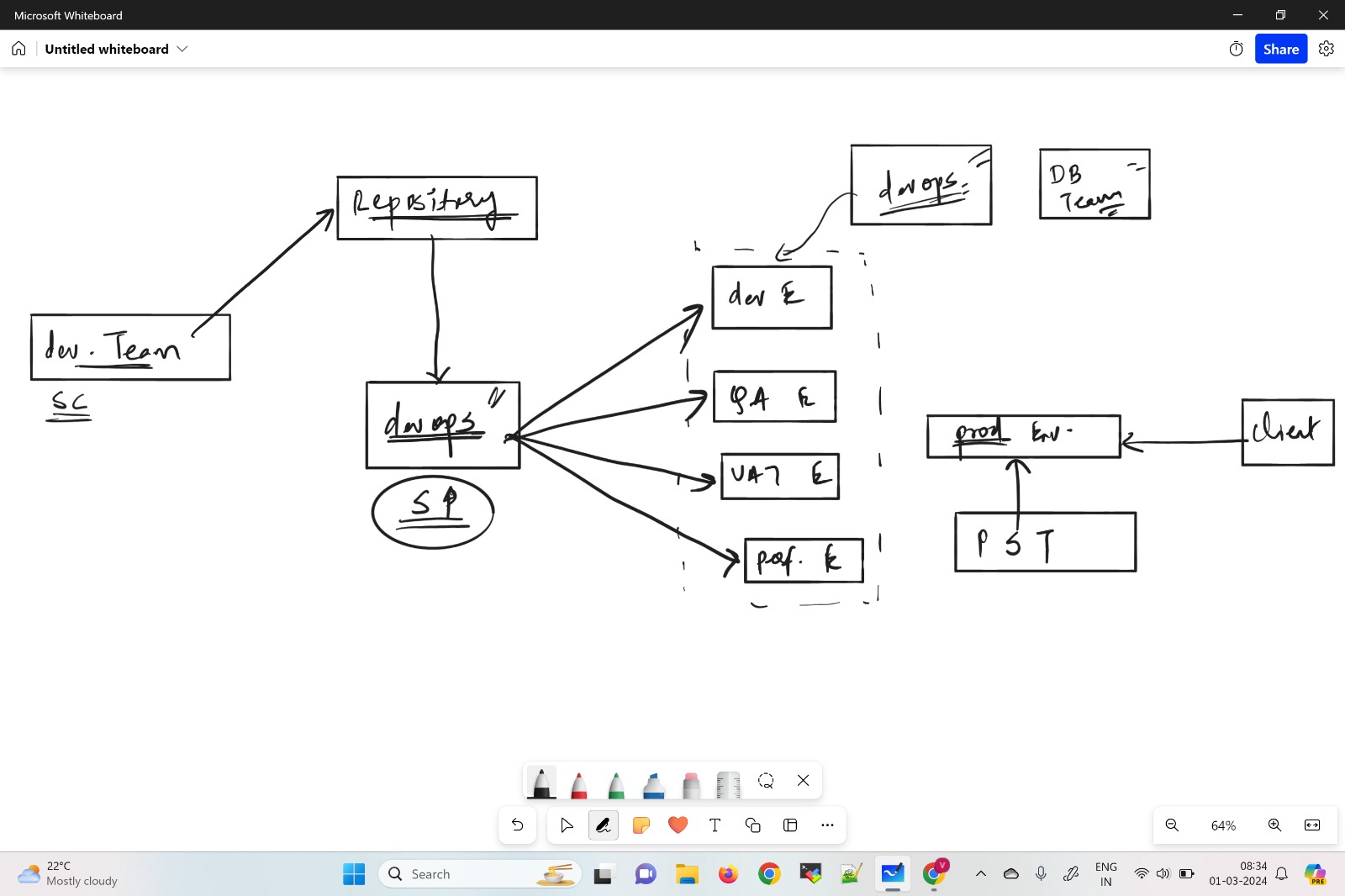
1. includes some part of dev team, i.e. source code management, and build.

2. includes full part of build and release team.

3. devops team includes full part of system admin team.

4. some part of DB support team.

5. some part of it-ops team.



**1/03/2024 Fri**

**Devops Team: (After Devops)**

**Role and resp.**

1. Creating and Maintaining cloud infrastructure. cloud infra consists of compute (RAM, CPU), storage, networking, DB and etc. EC2, VPC, EFS, S3, IAM, RDS

2. Monitoring of cloud infrastructure. (CW, CT, SNS)

3. Source Code Management (git, GitHub)

4. Build and release (Maven)

5. CI/CD (Continuous integration and continuous deployment/delivery) --- (Jenkins)

6. Containerization (Docker)

7. Configuration management (Ansible)

**Advanced Tools:**

8. Infatuations Provisioning (terraform)

9. orchestration (Kubernetes)

10. ticketing tool (Jira)

**DEVOPS Team:**

1. includes some part of dev team, i.e. source code management, and build.

2. includes full part of build and release team.

3. devops team includes full part of system admin team.

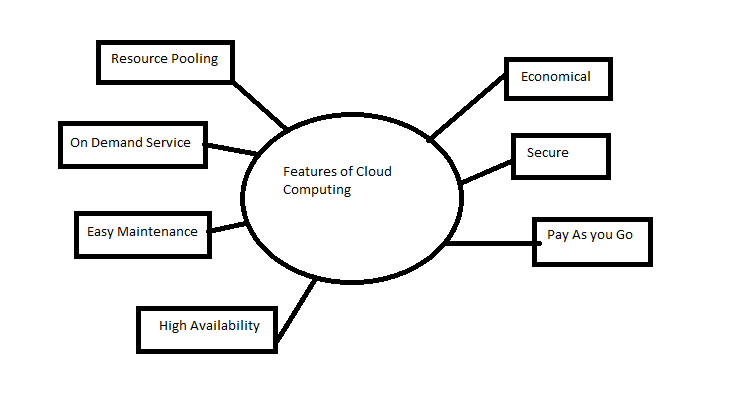
4. some part of DB support team.

5. some part of it ops team**.**

**What is cloud computing?**

* Simply put, cloud computing is the delivery of computing services—including servers, storage, databases, networking, software, analytics, and intelligence—over the Internet (“the cloud”) to offer faster innovation, flexible resources, and economies of scale.
* You typically pay only for cloud services you use, helping lower your operating costs, run your infrastructure more efficiently and scale as your business needs change.

**Features**

****

**Types of cloud**

1. Private Cloud
2. Public Cloud
3. Hybrid Cloud
4. Multi-Cloud

**Private Cloud**

* Private cloud is a cloud computing enjoinment in which all hardware and software resources are dedicated exclusively to, or accessible by a single customer.
* It’s a on-premises infrastructure.
* Companies choose private cloud because their workload deals with confidential documents, financial data etc.
* Private cloud is a single-tenant environment, meaning all resources are accessible to one customer only—this is referred to as isolated access. Private clouds are typically hosted on-premises in the customer's data center.

**ADVANTAGES/Disadvantages**

* Full control of hardware and software choices.
* Freedom to customize hardware and software in any way.
* Greater visibility into security and access control as workload is behind customer’s firewall.
* Higher cost of installing hardware/software
* Infrastructure management cost is high
* Limited flexibility.

**Public Cloud**

* Public cloud is a multi-tenant cloud environment, where the same computing resources are shared among multiple customers—sometimes hundreds or thousands of them.
* In public cloud, an independent cloud services provider owns and maintains the infrastructure, and access to resources is offered on a subscription basis or via pay-per-use pricing.
* Customer doesn’t have to maintain any hardware and no need for DC management.

**Advantages/Disadvantages**

* **Greater elasticity and scalability:**

With public cloud, a customer can add capacity in response to unexpended surges in traffic, without purchasing and installing new hardware.

* **Lower cost of entry**:

Most customers can begin using public cloud services without adding physical compute resources of their own.

* **Faster access to the latest technologies**:

In many cases, economies of scale enable cloud providers to offer the latest hardware and software faster than customers could if they had to purchase and install them themselves.

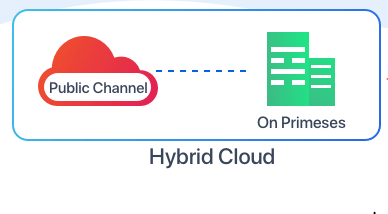
* **Only Disadvantage**:

could be that overall control over the hardware is not there, may be in some cases security can be an issue.

**04/01/2024 Mon**

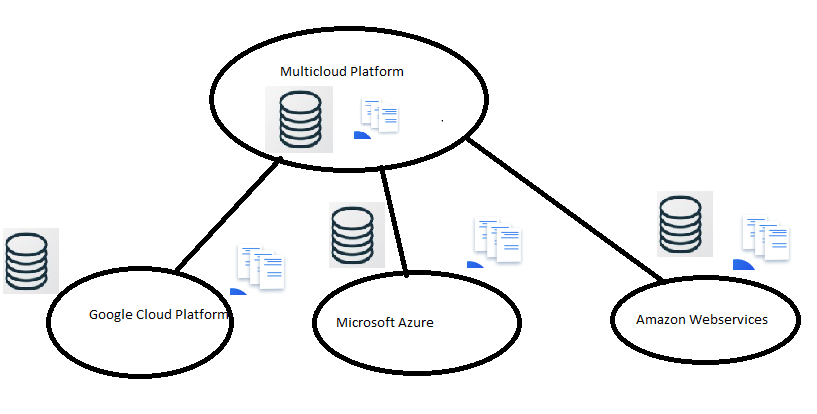
**Hybrid cloud**

* A hybrid cloud integrates public and private cloud infrastructures. In this model, the two types of cloud are joined together into a single, flexible infrastructure, and the enterprise can choose the optimal cloud environment for each individual application or workload.
* In simple words infra is shared in between your local data center and public cloud.

****

**Multi-cloud**

* Here if you could see in the multi cloud architecture, some applications are deployed in AWS others are deployed in Azure etc.
* This is done in large organizations as per client requirements

****

**05/03/24 Tues**

**Categories of Cloud/ Services provided by Cloud**

**1.SAAS (Software as a service):**

* Here a particular software program is hosted as a service over the internet, and the user doesn’t need to have any software technical knowledge to use that service

**E.g.** Google Drive, Drop box etc**.**

**2. PASS (Platform as a service):**

* PASS or a Pass (application platform as a service) is category of cloud computing service that enables customer to run an application.
* The only major difference between SaaS and PaaS is that - SaaS provides access to the software over the internet while PaaS provides a platform for software development
* **E.g**

|  |  |
| --- | --- |
| **PaaS Service** | **AWS** |
| App Deployment | ✓ Amazon Elastic Beanstalk |
| Big Data Processing | ✓ Amazon EMR |
| Data Warehousing | ✓ Amazon Redshift |
| DbaaS | ✓ Amazon Aurora, Amazon RDS |

**3.IAAS (infrastructure as a service)**

* This as the name suggest, this is a service where the software infra such as physical servers, RAM, CPU etc. is provided as a service to the customer as per their requirement.
* Mostly used by companies to run their day-to-day operations.
* Hence, unlike SaaS and PaaS, IaaS is a whole virtualization technology where clients have access to the whole infrastructure through an API or dashboard.
* E.g Ec2 services

**AWS – Infrastructure as a service**

* AWS is a cloud service where we are creating our infrastructure.
* Here instead of local DC infra we are using AWS
* Main Features

a. Infra Maintenance is not on our side.

b. H/W related upgrade or downgrade Is not on our side.

c. Pay as you go(use). Service provided on rental basis.

d. Only pay for the time you use. (Per Hour billing format)

e. On Demand, Flexible, Scalable etc.

**Features**

* **Elastic Capacity**

Scaling Up and Down in mins

No need to provision

Optimization of resources such as CPU/RAM etc. is possible.

Can manage unexpected peaks in traffic

* **Quick and Easy Deployments.**

IT infra is no longer a barrier

Easier to test software on different solutions

No need to wait for provisioning

**No initial investment needed**

Automation and reusable components-

Creation of application templates, instances storages etc. is possible.

**Availability regions and zones**

**Global AWS Infra**



-

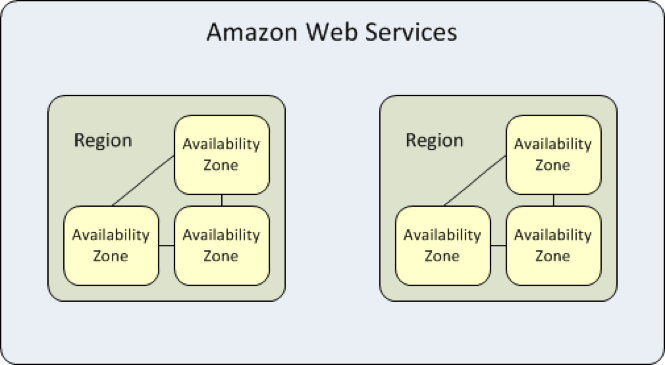
* Every DC location is called as a region
* Every region has multiple zones
* AWS breaks down a region into multiple zones
* Every zone is independent of each other.
* Each zone is represented alphabetically like a, b, c …. etc.
* The AWS Cloud spans 105 Availability Zones within **33 geographic regions** around the world, with announced plans for 24 more Availability Zones and 8 more AWS Regions in Australia, Canada, India, Israel, New Zealand, Spain, Switzerland, and United Arab Emirates (UAE).

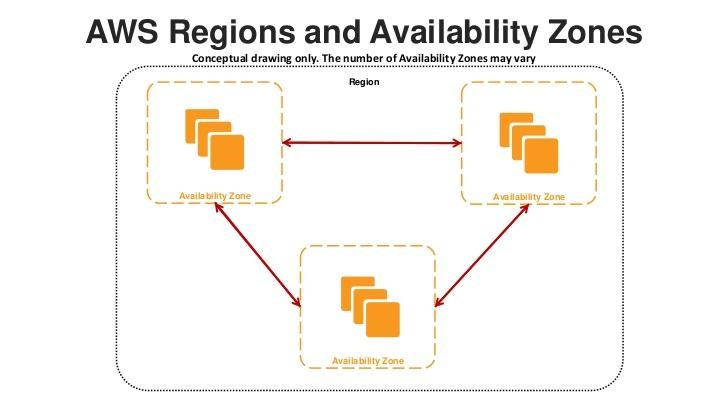
**Aws Regions**

* AWS has the concept of a Region, which is a physical location around the world where we cluster data centers.
* We call each group of logical data centers an Availability Zone **(AZ).**
* Each AWS Region consists of multiple, isolated, and physically separate AZs within a geographic area.
* Unlike other cloud providers, who often define a region as a single data center, the multiple AZ design of every AWS Region offers advantages for customers. Each AZ has independent power, cooling, and physical security and is connected via redundant, ultra-low-latency networks.
* AWS customers focused on high availability can design their applications to run in multiple AZs to achieve even greater fault-tolerance.
* AWS infrastructure Regions meet the highest levels of security, compliance, and data protection.

**Availability zones**

* An Availability Zone (AZ) is one or more discrete data centers with redundant power, networking, and connectivity in an AWS Region.
* AZs give customers the ability to operate production applications and databases that are more highly available, fault tolerant, and scalable than would be possible from a single data center.
* All AZs in an AWS Region are interconnected with high-bandwidth, low-latency networking, over fully redundant, dedicated metro fiber providing high-throughput, low-latency networking between AZs.
* All traffic between AZs is encrypted.
* The network performance is sufficient to accomplish synchronous replication between AZs. AZs make partitioning applications for high availability easy.
* If an application is partitioned across AZs, companies are better isolated and protected from issues such as power outages, lightning strikes, tornadoes, earthquakes, and more.
* AZs are physically separated by a meaningful distance, many kilometers, from any other AZ, although all are within 100 km (60 miles) of each other.





**The AWS universe**

1. Management + Interface
2. Cross Service Features
3. Platform Building Blocks (PAAS)
4. Infrastructure Building Blocks (IAAS)

**06/03/2024 Wed**

**Creation of AWS Account and Group Making**

**07/03/2024 Thurse**

**Creation of budgets**

After loging to AWS Console,

1. Go to my acc. And click on billing and cost management .
2. Then choose Budget in Budget and Planning section
3. Click on ‘Create Budget’
4. Use Simplified template and Zero spend budget which is bydefault
5. Simply put email in the box and click oncreate budget

This is how the zero spend budget is been created , if budget exceeds you will get the Mail

**Whatsapp group created**

**08/03/2024 fri**

**Elastic Compute cloud**

**Introduction**

* Amazon EC2 is a service that provides reusable compute capacity in the cloud.
* Free tier details:-

a. 12 MONTHS FREE  
b. 750 hours per month of Linux, RHEL, or SLES t2.micro or t3.micro instance dependent on region.  
c. 750 hours per month of Windows t2.micro or t3.micro instance dependent on region.

d. 30 GB volume per month free.

<https://aws.amazon.com/ec2/pricing/>

**Benefits of amazon ec2**

* Elastic web-scale computing, i.e. its scalable.
* Completely controlled by the user.
* Can be easily used for cloud hosting.
* Can be configured with other AWS services.
* Other benefits such as its Reliable, Secure, Inexpensive, easy to start.

**Features of EC2**

* Virtual computing service.
* Preconfigured templates are available or else we are create our own templates.
* Various config of CPU, Memory, storage etc. can be done.
* Secure login using keypairs
* Persistent volumes/storages are available for your data.
* Multiple physical locations for your resources.
* Firewall that enables you to specify the ports, protocols, IP etc.
* Static IP addresses available.
* Isolated Virtual **network**s possible**.**

**Types/categories of instances**

1. General Purpose Instances.
2. Compute Optimized Instances.
3. GPU Graphic instances/ Accelerated computing
4. Memory Optimized
5. Storage Optimized instances

[https://aws.amazon.com/ec2/instance-types](https://aws.amazon.com/ec2/instance-types/)

**1. General Purpose Instances.**

General purpose instances provide a balance of compute, memory and networking resources, and can be used for a variety of diverse workloads. These instances are ideal for applications that use these resources in equal proportions such as web servers and code repositories.

**2. Compute Optimized Instances.**

Compute Optimized instances are ideal for compute bound applications that benefit from high performance processors. Instances belonging to this category are well suited for batch processing workloads, media transcoding, high performance web servers, high performance computing (HPC), scientific modeling, dedicated gaming servers and ad server engines, machine learning inference and other compute intensive applications.

3**. GPU Graphic instances/ Accelerated computing**

Accelerated computing instances use hardware accelerators, or co-processors, to perform functions, such as floating point number calculations, graphics processing, or data pattern matching, more efficiently than is possible in software running on CPUs.

**4. Memory Optimized**

Memory optimized instances are designed to deliver fast performance for workloads that process large data sets in memory.

**5. Storage Optimized instances**

Storage optimized instances are designed for workloads that require high, sequential read and write access to very large data sets on local storage. They are optimized to deliver tens of thousands of low-latency, random I/O operations per second (IOPS) to applications

**Ways to launch instances**

1. **On Demand instance:-**

* Lets you pay for compute capacity by the hour with no long term commitments.
* When we directly launch the instances when we need is called on demand.

1. **Spot Instances:-**

* Amazon EC2 Spot Instances let you take advantage of unused EC2 capacity in the AWS cloud.
* Earlier biding on spare ec2 capacity was done to request spot instances, now in the new pricing model, biding is not required.
* When you use a spot instance you are taking a risk that if demand increases you will lose access to the spot instance (you are given a 2 minute warning before termination). For his reason it is common to use a mixture of on-demand/reserved instances and spot instances so that you can withstand instance terminations.

**3.Reserved instances**

* Lets say we need a instance for 1 year, then we will go for reserved instances, where amazon gives you upto 75% discount.
* Here the discount model works on how the payment is done.
* There are 3 types of payments models
* Full Upfront
* Partial Upfront
* No upfront..

**4. Dedicated Host**

Dedicated host is like a pvt cloud system where Particular resources are dedicated to Particular user only by cloud service provider , No any other user can use This resouces .

Due to this resources is dedicated to particular user only they are highly expensive as no resource pulling is done here ..

**Difference**

* On demand - kind of "default" mode. You request an instance, if there is free capacity, you will get the instance. No long term commitment, but once you get an instance, it's yours.
* Reserved - AWS reserves the capacity for you. You have guarantee that you will get the instance type in the selected region or AZ.
* Spot instance - it's kind of auction / bidding of unused capacity. You ask for an instance, you provide your maximum price and if there is free capacity and your price is at the current price or higher, you will get an instance. The difference is - if the free capacity is exhausted, or the current price is higher than your maximum bid price, your spot instance is terminated . You can get a termination warning event upfront.

**Differant OS can we launch-**

* Windows
* RHEL
* Fedora
* CentOs
* Ubuntu
* Suse Linux and many many more.

**Some Important concepts**

* Protocols
* Public/Private IP
* Keypairs
* DNS
* Volumes
* Default AMI
* Security Groups

**11/03/2024 Mon**

# **Basic terminologis**

**Protocol**:-

these are rules and regulations for communication between two electronic devices

there are multiple types of protocols used for communication associated with different ports such as

https/http ---- for website access

rdp --for windows access

ssh --- for linux access

ftp -- file transfer protocol

**IP**

:- internet protocol, its address of our machine

there are 2 types of IPs

1. public ip

2. Private ip

Private Ip is used for internal LAN communication for an internal network.

Private IP does not recognize the internet, and vice versa.

If we want to go to the internet we will need a public ip.

Private IP is provided by LAN admin and public ip is provided by ISP (internet service provider)

Private ip is free and public ip is paid if we are not using it.

**DNS**:- its a domain name system which is a human readable name we can use it instead of IPs E.g velocity.com amazon.com

**Volumes**:- its a storage such as SSD

**KeyPair**: Kepairs are used for login purpose for our ec2 instances.

there are 2 keys here, 1 is a public key and second is a private key.

private key is with the user and public key is with aws.

so with a private key we can login to the machine, where it will be authenticated with respect to its public key

**Pvt Public key Cryptography** - Public key cryptography is a method of encrypting or signing data with two different keys and making one of the keys, the public key, available for anyone to use. The other key is known as the private key. Data encrypted with the public key can only be decrypted with the private key.

E.g whatsapp encrypt data with user , i.e third party or person canot access it or can not read data..

**Security groups**:- its a 1st layer of firewall of our EC2 instance.

Security group has 2 rules

1. inbound rules--- for incoming traffic towards ec2 instance

2. outbound rule --- for outgoing traffic from the ec2 instance.

here with the help of SG we can manage what traffic from which protocol or port can access our ec2 instance and vice-versa

**AMI:**

AMI is Amazon Machine Image which is a template which consists of Operating system and software configuration.

We can launch a EC2 instance with the help of AMI.

**12/03/2024 Tues**

**[PRACTICAL]**

**Creating a instances (virtual machines ) Using EC2 Instances**

1. Select region first
2. Name the instance
3. Select AMI
4. Select zone from subnet
5. Create key pair (it is region specific use on for region in which it is created)
6. Create security walls or use bydefault
7. Create volume or select volume( bydefault for free tire use 30 gb volume)
8. Select how many machines need to create
9. Launch instance

KEY Pair –

pem. (privacy enhance mail) we can use this extenssion for windows and linux

ppk. (putty private key) this extension use for linux only for connect

instance from putty app

**13/03/2024 wed**

**[PRACTICAL]**

**Coneect to the EC2 instance that we created in EC2 service**

**There are two ways to connect to EC2 Instance i.e by manualy and By using remote desktop file**

**Manual way to connect EC2 instance**

1. Select instance which have to connect and copy public IP/ public DNS
2. Open RDP on PC and put Publlic IP which is copy from instance
3. Connect the machine by ‘CONNNECT’ tab
4. Then click on ‘RDP client’ tab
5. Copy username from there and click on Get password
6. Upload Private key and decrypt password copy password
7. Enter the copied Username and Password in RDP

Instance will be launch ..

**By using remote desktop file**

1. Select the machine which have to connect and connect the machine by ‘CONEECT’ tab
2. Then click on ‘RDP client’ tab And then click on ‘**Download remote desktop file’**
3. Then click on get password and decrypt password by uploading Private key and copy it
4. Go to download and open the file which is downloaded automatically when we click on download remote desktop file
5. Simply paste password i.e copied

Instance will be launched ..

**\*Mock group feb 17 is been created with all student by Mahesh sir**

**14/03/2024 Thurse**

**No class as due to shantanu sir health issue …**

**15/03/2024 fri**

**Launching and connecting Linux EC2 instance by using Putty and Mobaxterm**

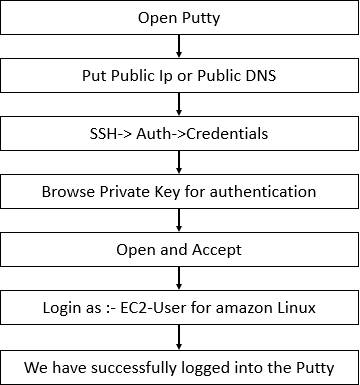
**Different software to login or Different Ways to Launch Linux:**

All the Steps are same for Launching Linux EC2 Instance. Only the difference is that we have to choose AWSLinux.

* + - **Putty:**

Putty is nothing but our **SSH (Secure shell)** client and

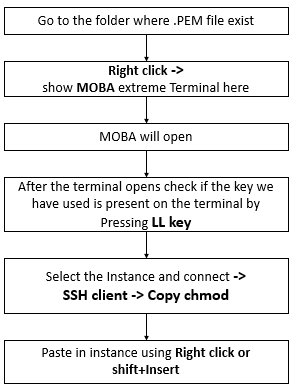
While Creating the Putty Always choose **(. PPk (Putty Private Key))** file.



**Black screen which opened is known as Shell Terminal or SSH Terminal**

* **MOBA:**

MOBA only support .PEM file.

**16/03/2024 Satur**

**Motivational lecture by Mahesh sir**

**17/03/2024 Sunday & 18/03/2024 Monday off**

**19/03/2024 Tues**

**Launching web page on http web server in windows operating system**

**20/03/2024 Wed**

**Basic commonds in linux operating system**

In windows C drive is the primary partition which consists of all the major or important files like program files, Users etc

**C  
(Primary Partition)**

Program   
files

Windows

User

Etc….

Different Folders inside **C**

**[Username@ ip-address Directory] #**

Here in Linux everything starts with (/) (**root**) because it is top level root directory and it consists of all the files in it same as C from windows.

**/  
(Top level root directory)**

Boot

Root

User

Etc….

Different Folders inside ***/***

* Here **@** is used to separate the username from IP address.
* **#** indicates Superuser

**Some Basic Commands of Linux:**

* **/ =**

**top level root directory**

1. **/bin**:
   * (binary) Contains all the commands which can be run with the help of all users.
2. **/boot**:
   * Container for bootable files for the OS.
3. **/etc**:
   * Contains all the configuration files, including machine-related and hardware-related configuration files.
4. **/home/username**:
   * Home directory for other users.
5. **/opt**:
   * Optional application software packages.
6. **/root**:
   * Home directory for the root user.
7. **/sbin**:
   * (system binary) Contains commands which only can be run by the root user.
8. **whoami**:
   * Tells us the current user on the terminal.
9. **hostname**:
   * Gives the machine name.
10. **ifconfig**:
    * Shows the private IP.
11. **date**:
    * Gives the machine date.
12. **uname**:
    * Shows OS name.
13. **uname -a**:
    * Shows all OS details.
14. **clear**:
    * Clears the terminal.
15. **history**:
    * Gives all the command history.
16. **pwd**:
    * Prints the **present working directory.**
17. **ls**:
    * Lists directory content.
18. **ls -ltr**:
    * Lists directory content in long format with time sorting in reverse.
19. **man ls**:
    * Opens manual for the ls command.
20. **cd directory\_path**:
    * Changes the directory.
21. **sudo su - username**:
    * Superuser do switch user.

**21/03/2024 Thurse**

**Commonds in Linux to creating file and directories**

1. **mkdir .velocity**:
   * Creates hidden directories.
2. **\*: e.g.: (ls -ltr \*html) (ls -ltr test1\*)…..etc**
   * Represents all.
3. **rmdir filename**:
   * Deletes the file but will ask for permission (press y for yes and n for no).
4. **rmdir -f filename**:
   * Forcefully deletes the file.
5. **mkdir dirname**:
   * Creates a directory.
6. **mkdir dir1 dir2 dir2**:
   * Creates multiple directories.
7. **mkdir -p dir1/dir2/dir3**:
   * Creates parent directories.
8. **rmdir dirname**:
   * Removes a directory.
9. **rm -rf dirname**:
   * Removes a directory in recursive mode.
10. **rm -rf. filename**:
    * Deletes hidden files and directories.
11. **rm -rf \*html**:
    * Deletes all files ending with ".html".

**VI MODE (Most popular editor):**

VI is nothing but same like notepad and there are 2 types of operations in it **Edit mode and Command Mode** (using commands).

Further we have 2 modes of operation;

**Command mode**

**Insert mode(i):** Type and insert.

• **:q:** Quit Vi editor.

* Exits Vi editor.

• **:q!:** Quit Vi editor without saving changes.

* Exits Vi editor without saving changes.

• **:wq:** Save and quit Vi editor.

* Saves changes and exits Vi editor.

• **:wq!:** Save and quit Vi editor forcefully.

* Saves changes and exits Vi editor, even if the file is read-only.

• **I:** Switch to insert mode.

* Enters insert mode for typing text.

• **Vi index.html:** Open a file in Vi editor.

* Opens a file named "index.html" in Vi editor.

• **/Keyword:** Search for a keyword.

* Searches for a specified keyword in the file.

• **Shift + G:** Go to the bottom of the file.

* Moves the cursor to the bottom of the file.

• **Shift + H:** Go to the head of the file.

* Moves the cursor to the beginning of the file.

• **dd:** Delete line.

* Deletes the current line.

• **Ctrl + D:** Save and exit.

* Saves changes and exits Vi editor.

• **Ctrl + U:** Move half a page up.

* Scrolls the screen up by half a page.

• **Ctrl + F:** Move a full page forward.

* Scrolls the screen down by one full page

• **Ctrl + B:** Move a full page backward.

* Scrolls the screen up by one full page.

**22/03/2024 Fri**

**Commonds in linux for Copy ,Paste and Move file or folder**

put method

cp filename destpath --- copy command

cp -r foldername destpath ---- to copy folder -r=recursive

get method

cp sourcepath/filename . ---- here . represents current localtion

cp -r sourcepath/foldername .

mv filename destpath --- cut and paste

mv filename newname ---- rename

grep pattern filename ---- to search a pattern in a file or use \* for all files

grep -i pattern filename ---- for case insensative

grep -ic pattern filename --- for count

wget url ---- to download from internet

tar -cvzf tarname.tar.gz foldername ----- it will compress and archive the folder in tar.gz formant

-c=create

-v=verbose

-z=gzip

-f=forcefully

tar -xvzf tarname.tar.gz

-x=extract

zip -r zipname.zip folder

unzip zipname.zip

**23/03/2024 Satur**

**Symbolic links in linus i.e harlink and softlink**

ln -- harlink --- ln file harlink -- does not work for directory -- backup

ln -s -- softlink -- shortcut ---- ln -s file/dir softlink-name

unlink softlink-name -- removes link

ln -s /mnt/file1 softlink-file1 --- create file in /mnt

with root chalao:-

yum list

yum list installed

yum list available

yum install package-name -y

yum remove package-name -y

yum update -y --- for patching -- to patch particular package -- yum update $package-name

yum install httpd -y

service httpd start

service httpd stop

service httpd status

service httpd restart

by using systemctl -- simlar to service command

**24/03/2024 Sunday off**

**25/03/2024 Monday Holi hollyday**

**26/03/2024 Tuesday**

**Linux commonds to add user and to give sudo permission to user**

useradd\*

create passwd\*

to see user cat /etc/passwd\*

switch user sudo su -\* or sudo -su\*

give admin permission

to perform imp commonds or operation we have to use sudo user i.e sudo

e.g sudo yum install package\*

to perform need to give permission commond vi /etc/sudoers

edit root ALL=(ALL) ALL

username ALL=(ALL) ALL

use tab button to give space in file

and save by :x!\*

then logout\* from current user and then again login

to add group commond is groupadd groupname\* and check group by cat /etc/group\*

\*in linux the user we add itself form a group bydefault

hence when we run the commond cat /etc/group\* we can see user as at a group location as a group

to add users in the group commond is

gpasswd -a username groupname\* to add single user

and to add multiple user commond is

gpasswd -M user,user2,user3 gruopname\*

to view user in group run commond cat /etc/group\*

to give sudo permission to group

run commond vi /etc/sudoers\*

and edit wheel permissio as

%wheel ALL=(ALL) ALL

%groupname ALL=(ALL) ALL

and save it by x!\*

and logout\* and then login again

to delete the user from group commond is

gpasswd -d username groupname\*

to delete user from group commond is

userdel username\*

man gpasswd to know extra commonds

to give permission to user to

read(4) write(2) and excute(1)

i.e rwx

d = directory

- = file

l = softlink

lets take a directory

drwxr\_xr\_ \_ 2 root root 6 mar 26 03:27 velocity

here first root stands for owner/user who create file

where second root stands for group .

here in above directory

owner,group and user permission resp.

d rwx r\_x r\_ \_

i.e

owner have access of file to read write and excute

group have permisson to read and excute while

user have the permission to read only

\*

no other user can access root directory but mnt directory can

access by anyone

we can check permisson by creating any file at mnt

and check

to channge the permission commond is

chmod -R 777 filename/directory name

777 is the full permisson give to file

i.e 4+2+1 4+2+1 4+2+1 i.e rwx rwx rwx

we can give any permisson such as 400 621 741

and any

to change owner of the file commond is

chown username filename\* and check by ls -ltr

to change group commond is

chgrp groupname filename\*

**27/03/2024 wed**

**Shell Scripting**

-shell script is nothing but the bunch of commonds

which can run together , with the help of shell script

we cant automate the daily repetative tasks as well as

any repetative ask ..

-shell script has a extenssion .sh e.g velocity.sh

- to write shell script we have to write shebang statement

first i.e #!/bin/bash then we give give discription

with #

e.g of shell script

#!/bin/bash

# this is my first shell script

echo "hello all, welcome to pune"

:x!

e.g

#!/bin/bash

# this is my shell script

# describe variable

name=Akash

echo "hello all my name is $name"

- to run or excute the script commond is ./filename

- before excute commond we have to give permission to

user to excut file . to give permission commond is

chmod -r 777 filename

**Types of varible**

**1)local variable** -the variable which is use only for

particular script and not applicable for other script

**2)environment variable** -the variable which is use for

any script and available on machine for any script

**3)special variable** - the varibles can define from outside

of the script is called special variable

-no need to describe variable in script

- Special variables

1) $n where n =0,1,2,3,4....

2) $@ and $\* = is use to print parameter we use for scripting

3) $# = is use to show count of special variables

4) $?=to show pass or fail status of commond

if commond is pass exit status will be zero if it is fail exit status will be non-zero

**28/08/2024 Thursday**

**Shell scripting Using Special Variable and using User Input method**

**Special variables**

**1) $n where n =0,1,2,3,4....**

**2) $@ and $\* = is use to print parameter we use for scripting**

**3) $# = is use to show count of special variables**

**4) $?=to show pass or fail status of commond**

**if commond is pass exit status will be zero if it is fail exit status will be non-zero**

to excute after using special variable commond is

**./filename.sh param1 param2 param3...\***

**there are two methods to give input in sript i.e**

**1) special variable to excute commond is**

**./filename.sh param1 param2 param3...\***

**2) by user input method to excute commond**

**. /user\_filename.sh**

**Basic Operators in shell**

**1)Arithmatic operators ( +, /, \*, -, % )**

**2)Relational operators**

**3)Boolean operators logical operators (! ,O )**

**4)String operatos**

**5)file test operators**

**1) Arithmatic Operators**

lets a=10 b =20

expr $a + $b then 10+20=30

for multipilcation expr $a /\* $b

e.g shell script is

a=20

b=10

value=expr $a + $b

echo "addition is $value"

value=expr $a - $b

echo "substraction is $value"

:x!

and excute by ./filename.sh

**2) Relational operators**

**3) Bullean Operator**

"logical !" =not which alters the output

"logical or" =if a=10 and b=20

[$a -lt $b] -O [$a -gt $b] if any one true then output is true

"logical and " if any one is false then output value is false

**4) String Operator**

zero string i,e empty string

= , !=

-n non zero and -z zero string

**5) file test operator**

**Decision making statement**

**using IF-ELSE**

if [$a -lt $b]

then

commonds

fi

if true then excute if false then not excute

**if-else statement**

a=10 b=20

if [$a -gt $b ]

then commonds

else commond

fi

if true true then excute and if false else excute

**for multiple conditions if -elif**

a=10 b=20

if [$a -gt $b ]

then commands

elif [$a -lt $b ]

then commonds

elif [$a -eq $b ]

then commonds

else commond

fi

**29/02/2024 Friday**

**Different types of Loops**

there are multiple loops they are

**1) for loop**

**2) while loop**

**3) Until loop**

**4)nested loop**

**5)infinite loop**

**1) for loop**

for i in 1 2 3 4 5 6 ...

do

echo "loop number $i"

done

another syntex of for loop

#!/bin/bash

# Using a for loop to print numbers from 1 to 10

for (( i = 1; i <= 10; i++ ));

do

echo "$i"

done

**2)while loop**

a=0

while [ $a -lt 10 ]

do

echo"$a"

a=expr $a + 1

done

if value is true then it will be excute

**3) Until loop**

a=0

until [ ! $a -lt 10 ]

do

echo"$a"

a=expr $a + 1

done

if valur is false then only it will be excute

**4) Nested loop**

e.g

for (( i=1;i=5;i++ ))

do

for (( j=1;j≤5;j++ ))

do

echo "$i"

done

echo""

done

**5) infinite loop**

**infinite**

**loop controll statement**

**i.e 1) break statement**

**2) continue statement**

**30/03/2024 Saturday & 31/03/2024 Sunday off**

**01/04/2024 Monday**

**Concept of Elastic IP**

**1. Allocation of EIP:-**

taking an EIP from amazon's pool of IP addresses to our aws account.

**2. release of EIP:-**

giving back the EIP from our account to AWS

**3. assocaition of eip:-**

attaching an eip to our resource. resource can be a ec2 isntance or a network interface.

**4. disassociation of eip:-**

detaching eip from our resource.

**5. re-association of eip:-**

reattaching eip to someother resource without manually detaching it.

**02/04/2024 Tuesday**

**Creating EBS volume and attaching to the windows EC2 instance ..**

**1. creating volume**

Creating a volume is to taking the volume from cloud to our SPECIFIC ZONE

Volume is zone specific

**2. Delete Volume**

To deleting volume is the process to return to volume from zone to cloud

**3. Attach Volume**

It is the process to attached volume to ec2 instance in the zone

**4. Detached Volume –––––**

To detached the volume from instance

**03/04/2024 wed**

**Creating ebs volume and attached to linux ec2 instance**

**Steps**

1) create linux ec2 instance and connect it

2) create a volume of 5 gb or any size in the same zone

in which instance is created (volume is zone specific)

3)attached the volume to instance that created previous

4} to check volume attached to instance , can use commond df -h

5) before mount the volume to any directory of instance we have

to format it first and then we need empty folder to mount as to

avoid overright of data to each other

6) to format volume commond is \*mkfs -t ext4 /dev/xvdb

7)after formating the volume to mount over empty folder commond

is \*mount /dev/xvdc /directory path

8)to unmount the volume commond is \*umount /dev/xvdb

**Activity :**

1)create one ec2 instance

2)create one volume of 5 gb

3)attached it to instance

4)format the volume and make file system

5)mount it over any empty directory of instance

6)create some file in the directory where the volume is mount

7)unmount the volume and detached from instance

8)create another instance and attched the 5 gb volume to the instance

9) check the data that we create in first instance in empty folder in which

the 5 gb volume is mount is seen in second instance

diff between du and df = du gives the size of the folder where the df gives

the size of the disk

**df -h**

**fdisk -l**

**lsblk**

**04/04/2024 thurseday**

**Theory related to AMI & Snapshot**

**An Amazon Machine Image (AMI) is a template that contains a software configuration (for example, an operating system, an application server, and applications). From an AMI, you launch an instance, which is a copy of the AMI running as a virtual server in the cloud.**

AMI is the backup of machine where Snapshot is the backup of volume .

both work on the principle of point in time backup .

-In EBS Snapshot written block is only store

-with the help of ebs snapshot we can transfer ebs volume from one zone to

another zone as well as one region to another region

- both ebs snapshot and AMI store at the backend in s3

A snapshot and an Amazon Machine Image (AMI) are related concepts in the context of AWS (Amazon Web Services) infrastructure, particularly with Amazon Elastic Compute Cloud (EC2). While they are both used for creating backups or templates of EC2 instances, they serve slightly different purposes and have different characteristics. Here's a detailed comparison between a snapshot and an AMI:

**Snapshot:**

1. **Purpose:** A snapshot is a point-in-time copy of an Amazon Elastic Block Store (EBS) volume. It captures the data stored on the volume at the time the snapshot is initiated.
2. **Content:** Snapshots only contain the data that has been written to the EBS volume, not the operating system or any other configuration details of the EC2 instance.
3. **Usage:** Snapshots are primarily used for data backup and disaster recovery. They can be used to restore data to an EBS volume or to create new volumes.
4. **Independence:** Snapshots are independent of EC2 instances. They can be created and managed separately from EC2 instances.
5. **Lifecycle:** Snapshots can be kept as backups for long-term storage. They are incremental, meaning that only changes since the last snapshot are stored, making them efficient for storing backups.

**Amazon Machine Image (AMI):**

1. **Purpose:** An AMI is a template that contains the root volume (the operating system, application server, and applications) required to launch an EC2 instance.
2. **Content:** An AMI includes all the necessary information to launch an EC2 instance, such as the operating system, application server, applications, and related configurations.
3. **Usage:** AMIs are used to launch new EC2 instances. They serve as blueprints for creating EC2 instances with specific configurations and software pre-installed.
4. **Dependence:** AMIs are tied to EC2 instances. When you launch an EC2 instance from an AMI, you essentially create a copy of the AMI.
5. **Lifecycle:** AMIs are typically used for creating and launching new EC2 instances. They can be versioned and shared across AWS accounts.

**Key Differences:**

1. **Scope of Content:** A snapshot captures only the data on an EBS volume, while an AMI captures the entire state (operating system, applications, and configurations) needed to launch an EC2 instance.
2. **Usage and Dependency:** Snapshots are used for data backup and restoration, independent of EC2 instances. AMIs are used for launching EC2 instances, and each instance is tied to an AMI.
3. **Lifecycle and Versioning:** Snapshots are incremental backups that can be retained for long-term storage. AMIs are versioned templates used for launching EC2 instances, and newer versions can be created and managed.

In summary, while both snapshots and AMIs are important for managing and deploying EC2 instances in AWS, they serve different purposes: snapshots for data backup and restoration, and AMIs for launching EC2 instances with specific configurations. Understanding these differences helps in effectively utilizing AWS resources for backup, recovery, and deployment needs.

We have EBS PPT .. FOR MORE Theory ..

**05/04/2024 Friday**

**Creating AMI, Creating instance with AMI, go to cross zone , cross region and share to another account with AMI**

1) Create a ec2 instance

2) Install tree httpd and maven on it..

3) Create html file in var folder

4) Create ami image from instance

5) Run command chkconfig httpd on

6) Delete instance and create new instance from that ami image

7) Access web page created in first instance through the second instance that created from ami image

8) Add git on it and create one 5gb extra storage and mount it over empty folder e.g mnt

9) Create some files in mnt folder..

10) Create AMI and snapshot of this instance ..

11) Create another instance from Ami and also create storage from snapshot

12) Take this instance through cross xone and cross region

13) Share the instance with another's account

**06/04/2024 & 07/08/2024 Saturday Sunday off**

**08/04/2024 Monday**

**Creating snapshot & creating volume from it**

**09/04/2024 Tuesday off Gudipadava..**

**10/04/2024 wed.**

**Elastic load balancer**

**\*appication and network load balancer difference .**

**There are four types of load balancer i.e**

1) Application load balancer

2) Network

3) GATEWAY load balancer – new addition to AWS

4) Classic load balancer – will be decommissioned soon by AWS

**Osi model ..**

1) Physical layer

2) Data link layer

3) Network layer

4) Transport lyer tcp tls udp... network load balancer

5) Session layer

6) Presentation layer

7) Appliacation layer http https ..appication load balancer

**Components of load balancer**

**1) listner** a)front end lister incoming request

b)back end listner it distrubute traffic among the ec2 instance

**2) target group -** group of ec2 instance connected to load balancer i.e target group

**3) Target –target** are nothing but the EC2 instance that are attached to the target group

**4) Health check –** load balancer also check health of the machine

**\*Diff between application load balancer and network load balancer**

Application Load Balancers (ALB) and Network Load Balancers (NLB) are both important tools used in distributed systems to ensure high availability, scalability, and reliability of applications. While they both perform load balancing functions, there are key differences in how they operate and the scenarios in which they are best suited.

**1. Layer of Operation:**

* **Application Load Balancer (ALB):** Operates at Layer 7 (Application Layer) of the OSI model. It understands HTTP/HTTPS traffic and can make routing decisions based on content such as URL paths or headers.
* **Network Load Balancer (NLB):** Operates at Layer 4 (Transport Layer) of the OSI model. It routes traffic based on IP protocol data (IP address and ports), making it suitable for both TCP and UDP traffic.

**2. Target Group Routing:**

* **ALB:** Routes traffic to different target groups based on rules that can inspect HTTP/HTTPS content. For example, it can route requests based on hostnames or URL paths.
* **NLB:** Routes traffic based on IP address and port, which is typically used for applications that rely on high-throughput and low-latency connections. NLB forwards the connection to a target group of instances based on configured rules.

**3. Protocol Support:**

* **ALB:** Primarily designed for HTTP and HTTPS traffic but can also support WebSocket and HTTP/2 traffic. It's ideal for web applications.
* **NLB:** Supports TCP, UDP, and TLS traffic, making it suitable for a broader range of use cases, including applications that require long-lived connections or protocols beyond HTTP.

**4. Load Balancing Algorithms:**

* **ALB:** Uses advanced routing algorithms for HTTP/HTTPS traffic, including round-robin, least outstanding requests, or IP-based routing.
* **NLB:** Employs flow-based and static IP-based routing algorithms. NLB preserves the source IP of the incoming connection and routes packets accordingly, which is useful for applications that rely on the visibility of the original source IP.

**5. Performance and Scalability:**

* **ALB:** Designed to handle high volumes of HTTP/HTTPS traffic efficiently. It automatically scales to accommodate traffic spikes and provides built-in health checks.
* **NLB:** Optimized for handling millions of requests per second with ultra-low latency. NLB is suitable for extreme performance demands and is often used for high-performance computing or IoT applications.

**6. Use Cases:**

* **ALB:** Ideal for microservices-based architectures, web applications, and APIs where content-based routing and advanced request handling are required.
* **NLB:** Suited for scenarios such as TCP/UDP-based applications, high-throughput applications, and situations where source IP preservation or low-latency routing is essential.

In summary, the choice between using an Application Load Balancer (ALB) or a Network Load Balancer (NLB) depends on your specific use case, application architecture, and performance requirements. ALB is more tailored towards HTTP/HTTPS traffic and provides advanced routing capabilities at the application layer, while NLB operates at lower layers of the network stack and is designed for ultra-high performance and flexibility in handling various types of network traffic.

11/04/2024 Thursday

**Practical of Creating a Elastic Load Balancer**

Creating a load balancer typically involves setting up a system that distributes incoming network traffic across multiple servers to ensure optimal resource utilization, maximize throughput, minimize response time, and avoid overload on any single server. Here’s a general guide on how to create a basic load balancer:

**To create load balancer we have to follow some ateps**

1)to create load balancer we need target group and to create we need target . so first we create target i.e EC2 instance

2) after creating instance creates target group & will add this instance in target group as a target

3) after that we can create load balancer, to create load balancer first we have to choose type of LB .. We will create application load balancer so we choose ALB..

4)while creating ALB we have to choose mapping zones .. i.e we have to choose zones in which we have to distrubute the incoming trauffic

5) we have to attach security group to LB

6)We will add listner and port and forward targeting group to LB and thus we can create load balancer & distrubute trauffic among multiple zones

**12/04/2024 Friday**

**Therory of load balancer & creating Network load balancer**

* **Elastic load balancer**
* A load balancer is a device that distributes network or application traffic across a number of servers.
* ELB distributes incoming traffic amongst various EC2 instances or available servers.
* ELB can distribute the load in between single AZ or multiple AZ
* ELB helps increase fault tolerance and makes sure there is high availability of the application 24\*7.
* **How does it work?**
* Load balancer checks on which server the traffic is less and accordingly forwards the request to the corresponding server.
* Incase any of the server is down it will move the request to any of the heathy server which is configured.
* Here we have to note that the load balancer will also do health checks on a regular time intervals and forward requests only to the healthy server.
* Usually load balancing is done together with auto-scaling, if any server is faulty then it gets replaced by a healthy server.
* **Types of load balancers**
* Application load balancer
* Network load balancer
* Classic load balancer – will be decommissioned soon by AWS
* GATEWAY load balancer – new addition to AWS
* **OSI (open system interconnection) model**
* The Open Systems Interconnection model is a conceptual model that characterizes and standardizes the communication functions of a telecommunication or computing system without regard to its underlying internal structure and technology

**Application load balancer**

* Application load balancer is used for mobile applications or web applications.
* ALB operates at the application layer of the OSI model, i.e. layer 7
* ALB is able to inspect application level content and route traffic based on HTTP and HTTPS protocol.
* Also ALB can route based on HTTP headers. For eg we want to access application which has a header /foo or /bar its possible

**Network load balancer**

* Network load balancer works on Layer 4 i.e is transport layer of OSI model.
* Basically network load balancer provides low latency (response time) and can manage heavy loads at a time.
* Network LB works on TCP/UDP/TLS protocols.

**Key difference**

* NLB just forward requests whereas ALB examines the contents of the HTTP request header to determine where to route the request.
* When you need to seamlessly support spiky or high-volume inbound TCP requests we use the network load balancer.
* ALBs are typically used for web applications. If you have a microservices architecture, etc

**HTTP vs TCP**

* Hypertext Transfer Protocol (HTTP) and Transmission Control Protocol (TCP) are both computer protocols involved in the transfer of data, but while they individually serve their own purpose, they have a close relationship.
* HTTP is a request-response protocol that allows users to communicate data on the World Wide Web (WWW) and transfer hypertext. The protocol remains one of the primary means of using the Internet and provides users a way to interact with web resources such as HTML files by transmitting hypertext messages between clients (such as a web browser like Chrome) and a server. Essentially, it’s used to load web pages using hypertext links.
* TCP, short for Transmission Control Protocol, is a communication standard that enables application programs and computing devices to exchange data and/or messages over networks.

**The Main Differences Between HTTP and TCP**

* HTTP typically uses port 80 – this is the port that the server “listens to” or expects to receive from a Web client. TCP doesn’t require a port to do its job.
* HTTP is faster in comparison to TCP as it operates at a higher speed and performs the process immediately. TCP is relatively slower.
* TCP tells the destination computer which application should receive data and ensures the proper delivery of said data, whereas HTTP is used to search and find the desired documents on the Internet.
* TCP contains information about what data has or has not been received yet, while HTTP contains specific instructions on how to read and process the data once it’s received.
* TCP manages the data stream, whereas HTTP describes what the data in the stream contains.
* TCP operates as a three-way communication protocol, while HTTP is a single-way protocol.

**Components of elastic load balancer**

* **Listener:-**

A listener is a process that checks for connection requests. It is configured with a protocol and a port for front-end (client to load balancer) connections, and a protocol and a port for back-end (load balancer to back-end instance) connections.

It listens to the incoming requests and fwds requests accordingly.

* **Target Group:-**

This is nothing but a cluster of EC2 instances. The ELB will only fwd the traffic to EC2 instances which are part of target group**.**

* **Target:-**

This is nothing but our individual EC2 instance, there is where we are going to target our traffic.

* **Health checks:-**

This is done to check whether instance is healthy or not. ELB does some prior health checks before registering targets and forwarding traffic to it.

* **Some important points**
* Load balancer is region specific.
* The load balancer will have a public IP, but the EC2 instances may or may not have public IP, they can communicate with each others based on private IP.
* A load balancer will only fwd. the request to the ec2 instance, it doesn’t mean that it will fwd. all the requests. In case of application load balancer it will only fwd http or https requests as it supports that protocol only.
* ELB has nothing to do with the other outbound traffic which is generated from EC2 instance to the internet or any other ec2 machine.
* If ELB is deleted then EC2 instances attached wont be deleted
* <https://aws.amazon.com/elasticloadbalancing/pricing/>
* Its always recommended to have equal number of ec2 instances in each zone. (cross zone load balancing)

**13/04/2024 & 14/04/2024 Saturday and Sunday off**

**14/04/2024 External mock Interview**

**Questions ask in interview**

**1) cat/etc/passwd|head What will the output of this commond**

Here there are two commond i.e cat and head and one operater i.e | ..

Cat commond will show the content of the file passwd and operator | will take this output as input for commond head which show first few lines of the file content ..

Hence this commond will display the upper content of first few lines of the file passwd

**2) What are the types of load balancer & which type of load balancer you use in current company ?**

There are two types of load balancer currently use in company . they are application load balancer and network load balancer ..

ALB is use in companied for high volumes .. it will read and understand the http and https traffics and distribute among the zones as per request .. it works on round robin algorithm. It is Ideal for microservices-based architectures, web applications, and APIs where content-based routing and advanced request handling are required.It is Designed to handle high volumes of HTTP/HTTPS traffic efficiently.

Where NLB is use in companies for scenarios such as TCP/UDP-based applications . Optimized for handling millions of requests per second with ultra-low latency. NLB is suitable for extreme performance demands and is often used for high-performance computing .

**3) How to create multiple files in linux ?**

To create multiple file in linux we use comond “touch” and giving file name by adding space in each file name thus ..

**4) How to create multiple files with itertrative approach without writing the files name using loop**

**5) why you are using genaral purposer ec2 instance even**

**though have multiple types of instance**

**6)write a script use loop for 1 to 10 with print**

**odd number and even number seprative and name even**

**and odd**

**7) how to fire multiple commond in single line**

**8) types of load balancer**

**9)diff between network load balancer and application**

**load balancer**

**10) cat/etc/passwd|head**

**10)mistakes in sheel script written by sahil**

**11)why we are using application load balaner**

**12)can configure my ec2 instance only from**

**application load balancer**

**13) protocols supported in nlb**

**14) why and where nlb use**

**15)cat/etc/passwd|head**

**16) types of instance in aws**

**17) why we are using devops as a statergy**

**18) diff ways to launched instance in aws**

**19) use of -v**

**20)many ways to create files**

**21) how to create file from cat**

**22) history >>life**

**23) history >life**

**24) mistake in sahil script**

**25)diff betn alb and nlb**

**26)configuration off insatnce which instance**

**you are using**

**27) why you are uing t2 micro not large instance**

**28) how to loggin to insatnce if you loose key**

**explore man cat man mkdir**

**du and df commond corn job**

**smdt server on linux**

**white paper of aws**

**link amazon.com/faqs**

**15/04/2024 Monday**

**Practical of disturbuting trauffic on particular taget group with the help of Application load balancer**

**Activity**

**1)** Create ec2 instance for payment , order & billing in each zones ..(i.e create 9 instance , 3 in each zones )

**2)** Install httpd server in every machine and create home web page in var folder also create parent directory in html folder for payment ,order and billing purspose ( single directory in single folder )

**3)** Create four target grops i,e for overall trauffic containg all the instance , for payment traffics contains payments instance and resp. for orders and billing

**4)** Create load balancer and attach target groups to load balancer

**5)** Create listener and rules with giving path for trauffic ..

**16/04/2024 Tuesday**

**Theory of Autoscaling Group and its components**

**What is Autoscaling ?**

Autoscaling is the ability to automatically adjust the number of EC2 instances in a fleet based on real-time demand. This concept is fundamental for ensuring that your application or service maintains optimal performance and availability while also efficiently utilizing resources.

Autoscaling is region specific

Autoscaling is done in multiple zones and it is recomanded to do in multiple zone

Autoscaling is totaly free

**Types of autoscaling :**

There are two types of autoscaling i.e one is **virtual autoscaling** and one **Horizontal autoscaling**

**a)** **virtual autoscaling :**

to increase the capacity of virtual machines automatically as per need is done in the virtual autoscaling.

In virtual autoscaling capacity of ec2 instances is increases as per real time demand means increasing the internal storage , changing the configuration and CPU utilisation in the same instance that are already present in environment.(or load balancer)

**b) Horizontal Autoscaling :**

to scale up or scale down number of instances as per need is done in Horizontal autoscaling .

**Horizontal Autoscaling** is mainly done in AWS rather than virtual autoscaling . as per need the number of instances are increase or decrease in environment.(or load balancer)

( Scale up= scale out )

( Scale down = scale in )

**What is scaling?**

* In cloud computing, scaling is the process of adding or removing compute, storage, and network services to meet the demands a workload makes for resources in order to maintain availability and performance as utilization increases.
* In simple words, as per our requirement increasing the capacity of anything is called scalability.
* **Auto-scaling** is nothing but automatic scaling of our capacity as per the requirement.

**Auto-scaling in aws**

* Here basically the concept is whenever the load on the server increases/decreases auto-scaling functionality helps to launch as well as terminate the EC2 instances.
* So here if you see scaling is done in both ways, we are increasing as well as decreasing the horizontal capacity.
* Scale OUT/UP means increasing the number of ec2 instances
* Scale IN/DOWN means decreasing the number of ec2 instances
* Auto-scaling is region specific, we cannot do auto-scaling between 2 regions.
* AS happens in Availability zones and we can distribute the instances evenly in Azs
* Here we create group of EC2 instances which can scale up and down as per the conditions we set.
* Auto scaling ensures that we have right number of EC2 instances to suffice our needs all the time, also autoscaling helps us reduce the cost by cutting down the number of instances when not needed.
* IMP Point:- No extra cost is needed for Auto-Scaling, only cost of EC2 instances

**Some imp points**

* Auto scaling makes sure that all the launches instances are balanced in between the set availability zones.
* Just incase these are not balanced or we later add the instances, auto scaling tries to balance it out automatically
* Here the criteria is we can only add a additional EC2 instance to the ASG group only if ec2 instance is in running state, its in the same AZ which is configured with ASG, and if the ASG has reached its max limit then we cannot add as the request fails.

**ELB with autoscaling**

* We can attach one or more ELB to the ASG.
* The ELB must be in the same region.
* Once we configure this, any EC2 instance existing or added by the ASG will be automatically registered with the ASG defined ELB.
* While enabling ELB with ASG we always need to enable ELB health check as well, by default ASG uses EC2 health check option.

**Health check**

* Auto scaling by default classifies its EC2 instances status healthy or unhealthy with the help of EC2 status check.
* By default the health check grace period is 300 sec, this means once the instance is launched after that till 300 seconds there will be no health check, its not recommended to keep this value as 0 as, if this is kept zero once the instance is launched immediately it will start health check and we all know that it takes some time for the instance to get into “initialized state i.e 2/2 status”
* Until the grace period timer expires any unhealthy status reported wont be acted upon by the ASG.

**Some imp points**

* Unlike rebalancing, here the termination of instances will happen 1st incase if its unhealthy, then the ASG attempts to launch new instances to replace the terminated one.
* The Elastic IP and EBS volume (additional) gets detached from the terminated instances, then we may need to manually attach them to new instances.
* Basic monitoring is enabled by default and is free of cost (300 sec).
* We can also make instances in stand by state manually incase we want to do any patch activity, the ASG will not do any health check on these machines, and there wont be any load forwarded on this machine for that time period.

**Auto-scaling components**

1. **Launch Configuration:-**

Here we have to configure which type of instance we want, we can select the AMI, key pair, security group, instance type etc. Here once the Launch config is created we cannot edit the same. We can only delete or copy it.

**2. Auto-Scaling Group:-**

here we can select the Group Name, Group Size (Min instances, Max instances, desired instances), also the health check period which is 300 sec by default.

**3. Scaling policies**

* **Some more points**
* Warm up time of an instance:-

Instance warm-up defines the number of seconds it takes for a newly launched instance to warm up.

This prevents the ASG from adding more instances than needed. Before this launch, the instance warm-up time was set to a default value of 300 seconds.

Warm-up value for Instances allows you to control the time until a newly launched instance can contribute to the CloudWatch metrics, so when warm-up time has expired, an instance is considered to be a part Auto Scaling group and will receive traffic.

**Some more points**

* Cool down period:-

A cooldown period is **a period of time after each scaling action is complete**. During the cooldown period, scaling actions triggered by alarms will be denied.

The cooldown period is a configurable setting for your Auto Scaling group that helps to ensure that it doesn't launch or terminate additional instances before the previous scaling activity takes effect

* The amount of time to wait for a previous scaling activity to take effect is called the cooldown period.

**17/04/2024 wed off for class Ramnavami**

**-**

**18/04/2024 Thursday**

**Theory of Autoscaling – Tracking policies**

**Auto-scaling policies**

* Manual :- keep scaling policies same (as it is)

Here the values of min, max and desired will be the same and If I have to change this value then I need to do it manually.

* Dynamic:-

1. Target tracking policy
2. Simple scaling policy
3. Step scaling policy
4. Scheduled scaling policy
5. Predictive scaling policy

* **Target tracking policy**
* It helps to auto scale based on the metrics like Average CPU Utilization, Load balancer request per target, and so on. Simply stated it scales up and down the resources to keep the metric at a fixed value.
* For example, if the configured metric is Average CPU Utilization and the value is 60%, the Target Tracking Policy will launch more instances if the Average CPU Utilization goes beyond 60%. It will automatically scale down when the usage decreases. Target Tracking Policy works using a set of CloudWatch alarms which are automatically set when the policy is configured.
* One metric value set i.e 60 %
* <https://docs.aws.amazon.com/autoscaling/ec2/userguide/as-scaling-simple-step.html>
* AWS recommends to use Target Tracking policy for a metric like Average CPU utilization hence mostly this is used.

**Simple Scaling**

* Simple scaling relies on a metric as a basis for scaling.
* For example, you can set a CloudWatch alarm to have a CPU Utilization threshold of 80%, and then set the scaling policy to add 20% more capacity to your Auto Scaling group by launching new instances.
* Accordingly, you can also set a CloudWatch alarm to have a CPU utilization threshold of 30%. When the threshold is met, the Auto Scaling group will remove 20% of its capacity by terminating EC2 instances.

**Step Scaling**

* Step Scaling further improves the features of simple scaling.
* Step scaling applies “step adjustments” which means you can set multiple actions to vary the scaling depending on the size of the alarm breach.

**More policies**

* Scheduled scaling – Set a schedule for eg weekend load is high scale up etc.
* Predictive scaling – based on history and uses AI

**Diff between Rebalancing and Helth check\***

**19/04/2024 Friday**

**Creating autoscaling group practicle –Manual scaling**

Steps to creating autobalncing group

1) first create ec2 instance and install httpd server in it .. make html file in var folder

2) run chkconfig commond to run server even though machine start after creating AMI

3) create AMI image of ec2 instance after succefully creating AMI we can delete the instance

4) create a launched template from AMI ..

5) create target group and not need to add target in it it will add by autoscaling group automatically as per need

6) create load balancer and attached target group to it

7) create autoscaling group with the help of launched template

8) add load balancer to autoscaling group

9) add health check grace period

10) whiile creating add dessired capacity min. capacity and max capacity in gropu size

\*Interview important questions

**1)health check grace period and warm up period**

**2)health check and rebalancing**

**20/04/2024 21/04/2024 Saturday Sunday off**

**22/04/2024 Monday**

**Creating autoscaling group practicle –Target tracking Policies**

**To give extra load to check auto scaling group woking or not**

**This three most important commands use**

yum install stress –y

amazon-linux-extras insatll epel -y

stress -c 20

**.23/04/2024 Tuesday**

**VPC virtual private cloud –theory**

VPC virtual private cloud

- it is completely isolated

- we can decide who can acces the cloud and cloud acces to whom

- we can create multiple vpc in acc/region

**types of VPCs** –

**1)default VPC (create by aws)**

**(It is public vpc)**

**-2) non default vpc (create by aws user) [can be pvt or public]**

-VPC is a region specific service

-Vpc is full network and divided into parts known as Subnet which is divide on ip base

**- Types of subnet**

**a)Public subnet**

(we can access the internet and internet can acces the subnet)

**b)Private subnet**

(we can not acces internet by pvt subnet and vice versa)

-EC2 instance is created in subnet

-if subnet is public we can give public ip to instance & connect to internet but

-if subnet is private then we cannot acces internet through ec2 even though we have given public ip to instance

-Bydefault all the vpc and subnets are private , to connect with internet you need to make it public

-To make vpc public from pvt we can use component **"Internet Gateway"**

-Public subnet have the access of internet gateway where pvt subnet do not have to the access of internet gateway

**-"Route Table"** 2nd component of vpc

-In the route table we mention the rules and regulation of network trauffic flow within vpc and from outside vpc to vpc and vice versa.

- multiple subnet or single subnet can attached to route table

- **types of RT**-**a)private root table** -(no internet gateway entry)

**b)public root table** -(internet gateway entry)

-in the internal network of vpc ,internal communication

is enable.i.e we can communicate between subnet a,b,c which are associate with the same vpc internally , bydfault it is enable by aws .

\***diff between internet gateway and route table**

-Internet gateway is the component wihcih provides

itternet access where route table the rules and regulation

are define by which we can access the internet .

-Internet gateway is the 'gate' where route table is provide

'route'

**24/04/2024 Wednesday**

**VPC virtual private cloud –theory**

**interview question**

**\*Diff between NAT gatewaty and internet gatewaty**

**\*Diff between security group and NACL**

1)security group it attached to ec2 instance ,

(first layer of firwallfrom ec2 side and second number

of firewall from subnet side)

2)NACL (network acces control list)

(second layer of firewall from ec2 side and first

layerof firewall from subnet side),nacl attached to subnet

NACL firewall can allows and also denied inbound trauffic

vpc pairing = to pair two vpc

pricing..

* VPC
* VPC is nothing but a virtual private cloud.
* It helps you to create a virtual isolated env in the same cloud.
* Multiple virtual isolated Environments are also possible.
* Amazon Virtual Private Cloud is a commercial cloud computing service that provides users a virtual private cloud, by "provision[ing] a logically isolated section of Amazon Web Services Cloud".
* Amazon Virtual Private Cloud (VPC) gives you complete control over your virtual networking environment, including resource placement, connectivity, and security.
* Why do we need vpc ?
* You need a VPC: a virtual private network that keeps your servers safe from the ravages of the public internet.
* Multiple Connectivity Options:-

Your AWS VPC can be connected to a variety of resources, such as the internet, your on-premise data center, other VPCs in your AWS account, or VPCs in other AWS accounts; once connected, you can make your resources accessible or inaccessible in your VPC from outside of your VPC based on your requirement.

* Secure
* Simple
* All the scalability and reliability of AWS is available.
* Use cases
* Host a public facing website.
* Host a multi-tier application, such as it will have web-layer, LB layer, DB layer etc.
* Hosting of scalable web applications in your cloud.
* Manage multiple projects, create isolated networks for those projects.
* Some IMP terms
* VPC:- this is nothing but a full isolated network.
* Default VPC:- in every region there is a small default VPC created by amazon, this default VPC has all the standard routing rules set by amazon.
* Non Default VPC
* Subnets
* Internet gateway
* Network ACLs (Network Access Control List)
* Security Groups
* NAT Gateway/Instance.
* Routing Tables.
* IPs, DNS etc
* VPC peering

**Default limits**

* We can have only up to 5 non-default amazon VPC’s per region
* You can create up to 200 subnets per VPC
* We can create up to 200 network ACL per amazon VPC
* We can have up to 5 Elastic IP addresses per AWS account per region.
* Count of IPv4 CIDR blocks / VPC – 5
* Count of IPv6 CIDR blocks / VPC – 1
* Count of Internet gateways / Region – 5
* Count of NAT gateways / Availability Zone - 5

**subnets**

* A subnet, or subnetwork, is a network inside a network.
* When we break down a network into smaller networks is called as subnet or subnetwork.
* This breaking down of network is based on IP. (CIDR)
* Basically there are 2 types of subnets

1. Public Subnet
2. Private Subnet

**Route table**

* A route table contains a set of rules, called routes, that are used to determine where network traffic from your subnet or gateway is directed.
* To put it simply, a route table tells network packets which way they need to go to get to their destination.
* There are 2 types of routing Public routing and Private routing
* Public routing is where we can route to the internet with the help of internet gateway.
* Private routing means we can routing between private subnets or internal network only.

**Internet gateway**

* An internet gateway is a horizontally scaled, redundant, and highly available VPC component that allows communication between your VPC and the internet.
* If a VPC does not have an Internet Gateway, then the resources in the VPC cannot be accessed from the Internet.
* A public subnet is a subnet that's associated with a route table that has a route to an internet gateway.
* In simple terms this means, when a subnet is connected to internet its called as public subnet
* An internet gateway has a Public IP attached to it.
* There's no additional charge for having an internet gateway in your account.

**NAT Gateway**

* NAT is nothing but network address translation.
* This will translate your private IP to public IP so you can access internet from the private IP,
* NAT gateway is usually created in public subnet with a public IP of its own.
* Ideally the NAT gateway will sit in between your IGW and private subnet.
* The NAT gateway will accept all the packets from private subnet and attach its own IP address to it and forward the packets to the IGW.
* https://aws.amazon.com/vpc/pricing/

**Nat gateway vs internet gateway**

* Internet Gateway (IGW) allows instances with public IPs to access the internet.
* IGW allows resources within your public subnet to access the internet, and the internet to access said resources.
* Two way communication
* NAT Gateway (NGW) allows instances with no public IPs to access the internet.
* It only works one way. Through NAT we can go from Private to Public but a public entity cannot access the private n/w.
* One way communication
* If we want 2 way NAT then we have to setup reverse NATing
* In cloud computing we do not go for reverse NATing which is a standard

**Security groups**

* A security group acts as a virtual firewall for your EC2 instances to control incoming and outgoing traffic.
* Network ACL (access control list)
* This acts as a firewall for associated subnets.
* It controls both incoming and outgoing traffic at the subnet level.
* It’s the 1st firewall at the network level.
* A network access control list (ACL) is an layer of security for your VPC that acts as a firewall for controlling traffic in and out of one or more subnets.

**NACL**

* A network access control list (ACL) is an optional layer of security for your VPC that acts as a firewall for controlling traffic in and out of one or more subnets.
* The following are the parts of a network ACL rule:
* **Rule number**. Rules are evaluated starting with the lowest numbered rule. As soon as a rule matches traffic, it's applied regardless of any higher-numbered rule that might contradict it.
* **Type**. The type of traffic; for example, SSH. You can also specify all traffic or a custom range.
* **Protocol**. You can specify any protocol that has a standard protocol number.
* **Port range**. The listening port or port range for the traffic. For example, 80 for HTTP traffic.
* **Source**. [Inbound rules only] The source of the traffic (CIDR range).
* **Destination**. [Outbound rules only] The destination for the traffic (CIDR range).
* **Allow/Deny**. Whether to *allow* or *deny* the specified traffic.
* Default network ACL
* The default network ACL is configured to allow all traffic to flow in and out of the subnets with which it is associated. Each network ACL also includes a rule whose rule number is an asterisk. This rule ensures that if a packet doesn't match any of the other numbered rules, it's denied. You can't modify or remove this rule.

**SG VS Network ACL’s**

* Security Group
* Operates at instance level.
* Supports only allow traffic.
* Its stateful- return traffic is automatically allowed.
* Evaluates all the rules before deciding whether to allow traffic or not.
* Network acl’s
* Operates at subnet level.
* Supports both allow and deny traffic.
* Its stateless- return traffic must be explicitly allowed.
* Its processes rules in numeric order when deciding whether to allow traffic or not.

**Vpc peering**

* A VPC peering connection is a networking connection between two VPCs that enables you to route traffic between them using private IPv4 addresses or IPv6 addresses.
* Instances in either VPC can communicate with each other as if they are within the same network. You can create a VPC peering connection between your own VPCs, or with a VPC in another AWS account.
* The VPCs can be in different regions (also known as an inter-region VPC peering connection).
* VPC Peering
* AWS uses the existing infrastructure of a VPC to create a VPC peering connection; it is neither a gateway nor a VPN connection, and does not rely on a separate piece of physical hardware. There is no single point of failure for communication or a bandwidth bottleneck.
* Multiple VPC peering connections
* A VPC peering connection is a one to one relationship between two VPCs.
* You can create multiple VPC peering connections for each VPC that you own, but transitive peering relationships are not supported.
* You do not have any peering relationship with VPCs that your VPC is not directly peered with.
* Some basics on the VPC connection
* The owner of the requester VPC sends a request to the owner of the accepter VPC to create the VPC peering connection.
* The accepter VPC can be owned by you, or another AWS account, and cannot have a CIDR block that overlaps with the requester VPC's CIDR block’s.
* The owner of the accepter VPC accepts the VPC peering connection request to activate the VPC peering connection.
* To enable the flow of traffic between the VPCs using private IP addresses, the owner of each VPC in the VPC peering connection must manually add a route to one or more of their VPC route tables that points to the IP address range of the other VPC (the peer VPC).
* Pricing
* If the VPCs in the VPC peering connection are within the same region, the charges for transferring data within the VPC peering connection are the same as the charges for transferring data across Availability Zones.
* If the VPCs are in different regions, inter-region data transfer costs apply.
* https://aws.amazon.com/ec2/pricing/on-demand/#Data\_Transfer

**25/04/2024 thurseday off due to shantanu sir health issue**

**26/04/2024 friday**

**Calculations in VPC**

ip add types

1)Ipv4 [32 bit]

2)Ipv6 [128 bit]

-currently ipv4 ip address is use widely ..eg 172.31.41.9

it contains 3 dots and four number ..

-the number are known as octate , each ocate has size 8 bit and hence the of ipv4 is 32 bit..

- each octate has range between 0 to 255

- so as per rule the 1 st ip add will be 0.0.0.0 &

last ip add will be 255.255.255.255

-256^4 ip add available

- Previously ip was divided on class base method i.e class A , class B , class c etc..

-ip dividation base on network to host ratio according to class

- in case of class this dividation be like as 8:24, in case of clss b 16:16 , in case of class c 32:8

- first network bit remains constant where host bit can be vary from 0 to 255

-this was old method of ip addresing in which network to host ratio is predecided

-currently for ip addresing the method use known as **Modern CIDR Notation**

- WE give prefix after ip to assign bit to host and network.

-bits reserve for network are constant and bit reserve for host can be varry

**CIDR Notation-**

**classless interdomain routing**

in this we give prefix after ip and that bit will be reserve for network

ip ranges allowd in vpc -

* 10.0.0.0-10.255.255.255 (10.0.0.0/8)
* 172.16.0.0-172.31.255.255
* 192.168.0.0-192.168.255.255
* Whenever we use any IP range the below 5 ip addresses are reserved with AWS as below. Lets consider 1st example.
* 10.0.0.0 – N/W address
* 10.0.0.1 – Reserved for AWS VPC
* 10.0.0.2 – Reserved for DNS server
* 10.0.0.3 – Reserved for Future use
* 10.0.0.255 – Reserved for Broadcast address
* Apart from these 5 we can use any. (.0, .1, .2, .3, .255)

**-:** **Practical creating VPC**

1) We have choose 'VPCs' service in services

2) Click to your VPCs and then on 'create vpc'

3) Give name to our VPC , Give 'IPV4 CIDR block' range with

prefixes to give pvt IPs for VPCs, then simply click on

'create vpc'

4) After creating vpc we have to create subnet , to create

subnet we choose 'subnet' tab which is just below the your VPCs

tab

5) Click on 'create subnet' ,choose VPCs and add subnet with

same zone or multiple zone as per need

6) While adding subnet to VPCs we have to choose 'availibility

zone' and give particular 'IPV4 VPC CIDR Block' to provide pvt

IPs to subnet

7) After creating subnet we will create 3 ec2 instance in

three subnets

8) While creating instance we have to choose the 'VPCs'

that we have created and 'enable auto assign public ip'

9) After creating instance we have make them public by giving

them public ip with the help of 'Internet Gateway' & 'Route table'

10) To create internet gateway we click on 'internet gateway' tab

>>create internet gateway >>naming the internet gateway>>

thus it will it will be create and then attached it to VPCs

by using action >> attached vpc

11)after that to create route table

tab Route table>> create route table >> naming the route table

>>select vpc and thus it will be create

12)after creating Route table we have to associate with vpc to associate

click on subnet associations>>add subnet association >> add subent

which we have to make public to the it

13) to give acces the internet gateway we have to add route in route

table to add route

click on add route>>choose0.0.0.0/0 as it represents internet access

in aws & then choose target as internet gateway

Thus VPCs & Subnets are been created and they are being

public as attached to Internet Gateway by Route Table,

We have also created instance in each subnets also ..

as we make subnet public instance have get there public ip

add and we can connect the instances now..

**NAT gateway** :- -A NAT gateway is a Network Address Translation (NAT) service.

- You can use a NAT gateway so that instances in a private subnet can connect to services outside your VPC but external services cannot initiate a connection with those instances. -To create NAT gateway click on nat gateway tab and name it thus it will be creates . - NAT gateway is creates in public subnet so it can access to internet .it have its personal ip which gives to instance so to access to internet by keeping instance private .

in short

1) create vpc ,give IPV4 CIDR block range to provide pvt IPs to vpc

2) create subnet, give VPCs IPV4 CIDR block range to provide pvt IPs

to subnet

3) Crete instances in subnet that we have created

4) to make this instance publice provide them internet gateway and

route table

5) create internet gateway and attached it to VPCs

6) create Route Table , associated subnets to it and give internet

getway access by adding route 0.0.0.0/0

thus we can create vpc and subnet and also create instance in

particular subnet ..

to Delete vpc we have to delete instance first then we can delete VPCs .

Subnet ,internet gateway and route table attachéd to vpc will get automatically deleted after deleting vpc

**27/04/2024 Saturday& 28/042024 Sunday weakly off**

**28/04/2024 mock interview by pradnya borse**

**Important questions perspective of interview**

**1) Diff between ELB & AlB**

1)alb work on 7th layer i.e application layer where nlb work on 4th layer i.e transmission layer

2)alb understand and examine https and http protocol and tranf request to particular server where nlb, It cant understand the http,https protocol

3)alb is designed for http,https trauffic where nlb supports TCP & UDP protocol

4)alb route trauffic base on url paths and hostname where nlb route trauffic based on ip add &port

5)ALB is ideal for microservices based archetectur,web application where NLB isSuited for scenarios such as TCP/UDP-based applications, high-throughput applications & high perfomnce machine learning

**2) Diff between Internet gateway & NAT gateway**

**3) Diff between health check grace period and warm up period**

**4)Steps to create load balancer**

1)to create load balancer we need target group and to create we need target . so first we create target i.e EC2 instance

2) after creating instance creates target group & will add this instance in target group as a target

3) after that we can create load balancer, to create load balancer first we have to choose type of LB .. We will create application load balancer so we choose ALB..

4)while creating ALB we have to choose mapping zones .. i.e we have to choose zones in which we have to distrubute the incoming trauffic

5) we have to attach security group to LB

6)We will add listner and port and forward targeting group to LB and thus we can create load balancer & distrubute trauffic among multiple zones

**5)Theory of vpc and creting vpc**

1) create vpc ,give IPV4 CIDR block range to provide pvt IPs to vpc

2) create subnet, give VPCs IPV4 CIDR block range to provide pvt IPs to subnet

3) Crete instances in subnet that we have created

4) to make this instance publice provide them internet gateway and route table

5) create internet gateway and attached it to VPCs

6) create Route Table , associated subnets to it and give internet getway access by adding route 0.0.0.0/0

thus we can create vpc and subnet and also create instance in

particular subnet ..

**5) Diff between security group and NACL**

**6) Diff between Rebalancing and Helth check**

**7) Diff between EBS volume & EFS file system & S3**

**8)Diff between du & df commond**

**29/04/2024 Monday**

**Practicle of creating VPCs ,Creating pvt route table**

**Creating NAT gateway**

**Jump from one ec2 to another ec2 in same VPC ..**

**Ping one ecc2 with another ec2 in same VPC**

**-we create vpc**

-we creates subnets in vpc

-we creates instnces in that subnets

-we create internet gateway and attached it to vpc

-we creates two route table i.e one is pvt and one is public

- we will associte public subnet to public route table and private subnet to pvt route table

-ping google.com to check server internet connection

- for check internal connection between subnets commond is - ping pvt ip

-to check connection first we have to allow inbound rule in security group i.e allow icmp protocol because ping commond work on icmp protocol.

-to login in 2nd insatnce from 1st instance from same vpc which have connection of internet gateway we need keypair .

-we can copy data of 2nd instance keypair and make a vi file in 1st instance and give permission

chmod 400 to the file

-then we run commond =

ssh -i "filename.pem" ec2-user@pvt ip 2nd instance

this process if know as **jump process** and 1st instance is known as **Jump Server..**

we also call as **bastian host**

-To give internet to 2 nd instance which is in pvt subnet we have to connect it NAT Gateway

-NAT gateway create in Public Subnet

-We will create NAT gateway & allow allocate elastic ip

-to connect pvt instance , pvt subnet,we need to give route in pvt route table ..

-NAT is a one way communication i.e you connect to internet from machine but you cannot connect that machine from the inernet

-after doing work we need to delete nat gateway, then to delete instance, then to release elastic ip & delete vpc..

**30/04/2024 Tuesday**

**Nacl firewall**

**NACL -Network access controll layer**

-it is first layer of firewall in case of VPCs

-it configure both inbound and outbond trauffic in ec2

-in case of security group which is 2nd layer of firewall in case

of VPCs , it only configured inbound trauffic and outbound trauffic

is manage by AWS itself

-security group is **statefull** where NACL is **stateless** so it configur

both inbound & outbond trauffic

-it works on rule number method method in which every inbound or

outbond rule have given specific number . in which lowest rule number

have 1st priority to excute

- if rule number x < rule number y then rule x is excute first and

then rule y will be excuted

to add NACL FIREWALL to VPCs following are the steps ..

1)

**VPC peering**

To do connection between two VPC from same region or cross region

we need component **"VPC peering"**

**Steps**

1) lets create VPC-a and VPC-b in same region

2) Create two subnet in VPC-a in which one we will keep it pvt and one will be public

3) Create one more subnet in VPC-b & we will keep it as pvt

4) Create internet gateway and attach to VPC-a for to create public subnet

5) Create two route table for VPC-a in which one is pvt(do not have entry of internet gateway) & one is public(with entry of internet gateway) & also create pvt route table for VPC-b

6) Associates two subnets that created in VPC-a to pvt and public route table of that VPCs, also have to add route for internet gateway in public route table

7) Also associte subnet created in VPC-b to the pvt route table of VPC-b

8) Now, to do connection between two vpc we need vpc peering component . so in peering tab we will choose peering connection

9) Peering connection>>create peering>>name the peering>>select vpc acceptor and vpc reciver>>create peerig coonection (acceptor is the one who have at least one public subnet

for internet connection from where can jump to another VPC which act as a reciver )

10) After that need to accept request from accepator , to accecpt request select peering connection>>actions - accept request .note\* if accepator & requester is in same region we can accecpt request from same zone but if acceptor is in cross region we need to go in that region to accept the request

11)now , we need to add route in route table

We will add CIDR block of pvt subnet of vpc-b in pvt route table of vpc-a &

CIDR block of pvt subnet of vpc-a in pvt route table of vpc-b

For establish peering connection between two vpc (which are in same region )

12)now all the connections are done , route tables are updated

So we can jump from public ec2 instance vpc-a to pvt ec2 instance of vpc-a

13) to login to pvt ec2 instance from public ec2 instance we need a key ...

So first we log in to public ec2 instance and create vi file of pvt key of the pvt instance in which we have to login ( **vi key.pem** ) in it and give permission to user to read the file by commond – **“chmod 400 key.pem”**

To login in pvt ec2 insatnce from public ec2 insatnce in same region we have commond

**ssh –i “key.pem” ec2-user@pvt ip** ...........(pvt ip of pvt insatnce)

14)thus by using this commond we can login to pvt instance in from same vpc

15)for login to another pvt instance which is in another vpc , we will follow the same steps

i.e we will create vi file of pvt key of that instance and give permission by chmod commond to read file by and run commond ‘ **ssh –i “key.pem” ec2-user@pvt ip ‘**

16) now we will establish connection between two VPCs which are in cross regions.

So create vpc-c in cross region

we need another peering connection . For this connection so we will create one more peering connection ..

16) create subnet , ec2 instance in that subnet and route table ..

Update route table route entries as per new vpc CIDR block ..

17) now to jump from one vpc to to another vpc in cross region we need pvt key of that instance of cross region . so create key file >>give permission by chmod commond >>login using ‘ **ssh –i “key.pem” ec2-user@pvt ip ‘**

**Thus we can jump from one vpc to another vpc and this process is known as jump process & the server from where we have jump known as Jump Server or Bastian Host.**

**1/05/2024 Wednesday**

**EFS elastic file system**

-Data store as a file system

-It is a file sysytem whcih is use for shairing data for multiple ec2 instance

-Two types **1)one zone**

**2)all zone**

-EFS work on **NFS**(network file system protocol ) protocol where EBS work on **San technology.**

-EFS have there IP and also security group , Where EBS do no not have ip

-Work on pay as you go model.

-EFS is non butable storage where EBS is botable storage.

**Differantiative points between EFS file systen & EBS storage**

**EBS elastic block storage**.

EBS Volume:

1. This is the Block Storage solution for EC2 instances where data is stored as a block.

2. EBS volume is attached only for single ec2 instance.

3. EBS volume works on SAN technology.

4. EBS volume is Zone Specific, i.e if an ec2 instance is in zone a the corresponding ebs should be in zone a.

5. EBS volume is a bootable volume.

6. EBS volume can be attached to both windows as well as linux machines.

7. if you create an EBS volume of 10 GB and add only 2 GB data you pay for all 10 GB volume, i.e we pay for unused 8GB storage as well.

8. No volume autoscaling and no auto performance optimization.

9. EBS is directly attached to the EC2 instance and then is mounted on a folder, i.e it becomes a part of ec2 instance.

10. No SG of its own, it uses ec2 SG.

11. EBS is used as a bootable storage for ec2 instance OS as well as running softwares.

**EFS elastic file system.**

EFS Volume:

1. EFS is a elastic file system volume which stores data as a file system object.

2. EFS is attached simultaneously to multiple ec2 instances.

3. EFS works on NFS protocol with 2049 port number.

4. EFS can work in one zone or all zones.

5. EFS is a non bootable volume.

6. EFS works only with unix or linux like os.

7. You only pay for the data you store in EFS, i.e GB/month.

8. Storage scales up in EFS as we add data, and auto perfomance optimization is also available.

9. EFS is a network file system which we can attach to ec2 instance with the help of Private IP or DNS, so the mounting is done with the help of network.

10. EFS is created inside a VPC and has its own SG.

11. EFS is used as a common data source/store for our application data which is needed by our applicaiton running on multiple ec2 instances.

**02/05/2024 Thursday.**

**Detail theory of Elastic file storage & creating EFS storage mount over all the zones**

Mount target , Storage classes & Detail Archetecture of EFS

**Archetecure of EFS –**

**For one zone**

**For all zone**

**Mount Target**

- it is nothing but the point where EFS is attached to subnet.

- Mount target is nothing but pvt ip which is use to connect to pvt subnet.

- EFS have there public DNS which is use to connect to public subnet.

-PVT ip is use for connection to one zone instance in the subnet in which pvt ip taken radomalu from the CIDR block of subnet.

-Public DNS is use for connection of all zone

**Storage classess**

1) frequently access data -daily acces data

2) infrequently access data

diff storage classes

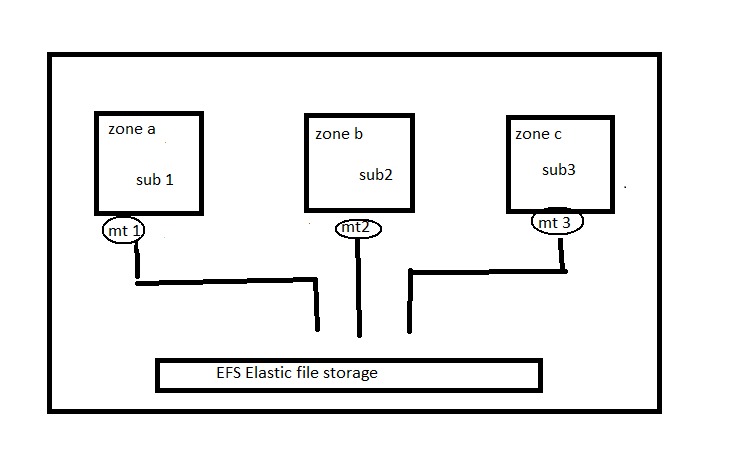
depend upon all zone and single zone price is varry

1) standard storage class default class use for frequently access data

2)standard infrequent access data use for infrequently access data

3)Archiver- use for very infreqently access data..

**Archetecture of EFS**



**Practicle :- Creating EFS storage mount over all the zones**

create efs - to create we need security group

->>create security grp>>naming it >>not add any inbond rule

to create efs >>go to efs service >>create efs service

>>naming efs>>select vpc>>customize

>> file system time for all zone is regional and

for single zone its one zone

>>if need automatic backup we can check to clickbox if not not need selct id

also it is chargable)

>>life cycle manager-it is use to transsion data from one class to another automatically

when >>throghput mode enhances >>elastic >>next

>>mount target >>choose security grp for subnet>>next

>>policies not needed >> review >>next

thus EFS CREATED

>>launched instances a,b, c in three subnets

>> connect all the three instance via moba

>> go to efs >> attached efs>>if need to mount with public ip choose publice dns if need to connect

with pvt ip

>> to mount efs to insatnce run commond

-- sudo mount -t commond

>> first we need to update security grp of efs ,,we will add NFS protocol rule with cidr block of vpc/cidr block of subnet /secuity group

firewall to firewall access - to give access of security group to security group access

to unmount the EFS commond is umount /folder name

**assignment**

>>cereate own vpc

>>create 4 subnet a1, a2 ,a3 & a4

>>two in a zone, one in zone b ,one in zone b

>> a1 is public , a2 a3 a4 is pvt subnet

>>create efs mount over a2 a3 a4 subnet

**03/05/2024 Friday**

**Simple Storage Service S3**

-s3 is nothing but simple storgae wervice provide by aws as a storage interface where can simply store and retrive data from anywhere anytime ..

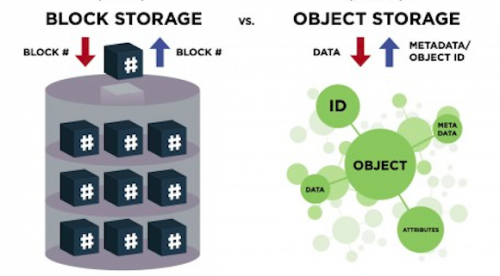
- anyone who have access can retrive this data .

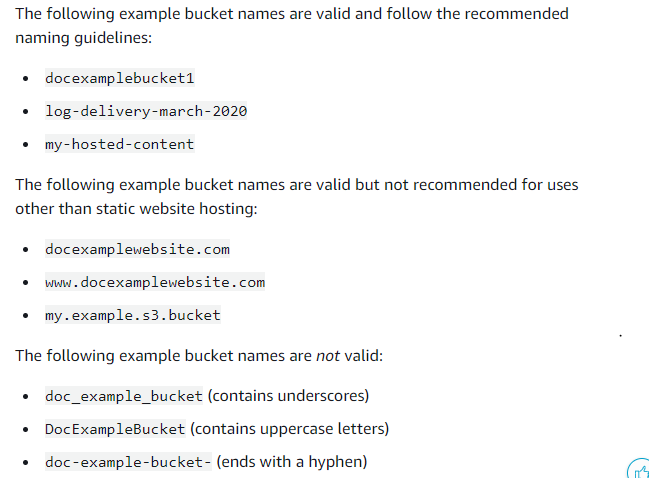
- in s3 data store in the form of object and not in form of blocks

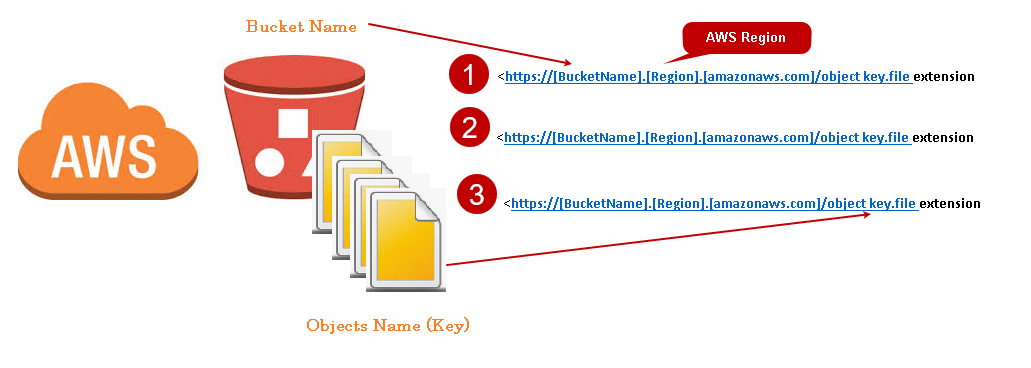
- Amazon Simple Storage Service (S3) is a scalable and highly durable object storage service

- S3 is widely used for a variety of use cases including data storage, backup and recovery, static website hosting, content distribution, and data archiving.

**Simple storage service (s3)**

* **What is a object storage?**
* Object storage is a computer data storage architecture that manages data as objects, as opposed to other storage architectures like file systems which manages data as a file hierarchy, and block storage which manages data as blocks.
* 
* **what is s3 in aws?**
* Amazon S3 or Amazon Simple Storage Service is a service offered by Amazon Web Services that provides object storage through a web service interface.
* This is similar to google drive but more advanced for technical people.
* This is not like block storage we cannot mount anywhere.
* It’s a only drive we do not need to create any file system.
* It is designed to make web-scale computing easier for IT people.
* Basically, s3 is storage for the internet which has a simple webservices interface for simple storing and retrieving data anytime or from anywhere on the internet.
* S3 has a distributed data-size architecture where objects are redundantly stored in multiple locations (min 3 zones).
* **Advantages**
* Reliable Security
* All-time Availability
* Very Low cost
* Ease of Migration
* The Simplicity of Management
* Distributed architecture
* An AWS Availability Zone is a physically isolated location within an AWS Region. Within each AWS Region, S3 operates in a minimum of three AZs, each separated by miles to protect against local events like fires, power down, etc.
* **S3 concepts**
* Buckets
* Keys
* Objects
* S3 bucket resources
* S3 storage classes
* **Buckets/Keys/Objects**
* An Amazon S3 bucket is a public cloud storage resource available in Amazon Web Services' (AWS) Simple Storage Service (S3), an object storage offering.
* Amazon S3 buckets, which are similar to file folders, store objects, which consist of data.
* Data is stored inside a bucket.
* Bucket is nothing but a flat container of objects
* The total volume of data and number of objects you can store are unlimited. Individual Amazon S3 objects can range in size from a minimum of 0 bytes to a maximum of 5 TB.
* You can create or upload multiple folders in one bucket but you cannot create a bucket inside a bucket (Nested bucket not possible)
* S3 bucket is region specific.
* You can have 100 buckets per account, this number can be increased.
* By default buckets and its objects are private, thus by default only the owner can access the buckets.
* There are some naming rules while creating a s3 bucket.
* A S3 bucket name should be globally unique across all the regions and accounts.
* Bucket names must be between 3 and 63 characters long.
* Bucket names can consist only of lowercase letters, numbers, dots (.), and hyphens (-).
* Bucket names must begin and end with a letter or number.
* Bucket names must not be formatted as an IP address (for example, 192.168.5.4).
* <https://docs.aws.amazon.com/AmazonS3/latest/userguide/bucketnamingrules.html>



* An object is a file which is kept in a bucket. File can be a .txt, .html, .mp3. a photo etc.
* The object key (or key name) uniquely identifies the object in an Amazon.
* An object size stored in an s3 bucket can be upto 5TB.
* Each object is stored and retrieved by an unique key (ID or Name)
* An object in aws s3 is uniquely identified and address through the service endpoint (region), bucket name, object name (key), and version (optional)
* Here the owner of the bucket can give permissions to objects, make them public or give access to particular account.
* 
* **S3 sub resources**
* Lifecycle
* Website
* Versioning
* Access control lists
* **S3 object lifecycle**
* A lifecycle configuration is a set of rules that are applied to the objects in specific S3 buckets.
* Each rule specifies which objects are affected and when those objects will expire (on a specific date or after some number of days).
* **Website**
* You can use Amazon S3 to host a static website. On a static website, individual webpages include static content.
* When you configure a bucket as a static website, you must enable static website hosting, configure an index document, and set permissions.
* You can enable static website hosting using the Amazon S3 console, REST API, the AWS SDKs, the AWS CLI, or AWS CloudFormation.
* Configuring a static website using a custom domain registered with Route 53
* **Versioning**
* Versioning in Amazon S3 is a means of keeping multiple variants of an object in the same bucket.
* You can use the S3 Versioning feature to preserve, retrieve, and restore every version of every object stored in your buckets.
* With versioning you can recover more easily from both unintended user actions and application failures.
* Versioning-enabled buckets can help you recover objects from accidental deletion or overwrite
* For example, if you delete an object, Amazon S3 inserts a delete marker instead of removing the object permanently.
* The delete marker becomes the current object version.
* You can still recover the object by deleting the delete marker.
* This versioning is incremental versioning.
* You can use versioning with S3 lifecycle policy to delete older versions to move them to cheaper s3 versions.
* Versioning can be applied to all objects in a bucket, not partially applied
* **Access Control Lists**
* Amazon S3 access control lists (ACLs) enable you to manage access to buckets and objects.
* Each bucket and object has an ACL attached to it as a subresource.
* It defines which AWS accounts or groups are granted access and the type of access.
* When a request is received against a resource, Amazon S3 checks the corresponding ACL to verify that the requester has the necessary access permissions.
* <https://docs.aws.amazon.com/AmazonS3/latest/userguide/acl-overview.html>
* **Storage classes in amazon s3**
* Amazon s3 standard
* Amazon glacier deep archive
* Amazon glacier
* Amazon s3 infrequent access (standard IA)
* Amazon s3 one-zone IA
* Amazon S3 intelligent tiering

Theory Link:- <https://aws.amazon.com/s3/storage-classes/>

Pricing link:- <https://aws.amazon.com/s3/pricing/>

**04/05/2024 Saturday off.**

**05/05/2024 Sunday**

**External mock by Abhishek chavan**

**Questions**

1. **Basic AWS Services:**
   * What is AWS? Explain its significance in the cloud computing industry.
   * Describe the core components of AWS (e.g., EC2, S3, RDS).
   * What is the difference between Amazon S3 and Amazon EBS?
   * How do you ensure security in AWS?
2. **Compute Services:**
   * What is Amazon EC2? How does it differ from other compute services?
   * Explain the concept of auto-scaling in AWS.
   * How do you choose between EC2 instances based on your application's requirements?
   * What is AWS Lambda, and what are its use cases?
3. **Storage Services:**
   * What is Amazon S3? Describe its features and use cases.
   * Explain the difference between Amazon EBS and Amazon S3.
   * How do you ensure data durability in Amazon S3?
   * What is Amazon Glacier, and when would you use it?
4. **Database Services:**
   * Describe Amazon RDS and its benefits.
   * What is Amazon Aurora, and how does it differ from other RDS databases?
   * Explain the concept of Multi-AZ deployments in Amazon RDS.
   * What is Amazon DynamoDB, and what are its key features?
5. **Networking Services:**
   * What is Amazon VPC, and why is it important?
   * How do you establish connectivity between on-premises data centers and AWS VPC?
   * Explain the purpose of AWS Direct Connect.
   * What is Amazon Route 53, and what are its key features?
6. **Security and Identity Services:**
   * What is AWS IAM, and how do you use it to manage access to AWS resources?
   * Explain the difference between IAM roles, users, and groups.
   * How do you encrypt data in AWS?
   * Describe AWS Shield and AWS WAF.
7. **Monitoring and Management Services:**
   * What is Amazon CloudWatch, and how does it help in monitoring AWS resources?
   * Explain the difference between CloudWatch Logs and CloudTrail.
   * What is AWS CloudFormation, and how does it work?
   * How do you use AWS CLI and SDKs to interact with AWS services programmatically?
8. **cicd**
9. **jar and war files**
10. **significance of cloud computing**
11. **diff between on premices and cloud adv disadv**
12. **diff types of cloud plateomrm**
13. **eg of pvt cloud**
14. **complte harrarchy of aws**
15. **om s.. linux commond shell scri[ting**
16. **diff between agile and waterfall**
17. **rt factory**
18. **what is repository**
19. **services provided by aws**
20. **how many regions and Zones present**
21. **diifert types of insatnce and there use case**
22. **ways to launched insatnace**
23. **pvt pub key cryptography**
24. **types of AMI**
25. **HOW To shair ami Cross account**
26. **what is kernal , DNS**
27. **Why linux is prefer over windows**
28. **what is root directory sub-directories presemt under top most roo directory i.e / and**
29. **use cases of subdirectories**
30. **how to create web page steps**

**06/05/2025 Monday**

**Storage clsses in S3 & motivation**

**Creationg s3 bucket ,**

**S3 storage Classes:-**

**1. Standard Storage Class:-**

Default storage class.

Used for frequently accessed data.

Storage cost is expensive as compared to other classes.

Retrival time is instant

retrival cost is Not applicable.

works in >= 3 zones

**2. Standard Infrequent access:-**

Used for data which you want to access once in 15 days or twice or thrice in a month.

Storage cost is less than that of standard.

It has instant retrival.

Retrival cost exist

works in >= 3 zones.

**3. One Zone IA:-**

it is exactly same as that of standard IA but works only in One Zone.

Storage cost is less than that of Standard IA.

**4. Glacier Instant Retrival:-**

Storage cost is cheaper than Standard IA.

used for data which we access at a frequency of once or twice in 2-3 months.

Instant retrival of data

Retrival cost exist and is more.

works in >=3 zones

**5. Glacier Archive:-**

Storage cost is cheaper than Standard IA.

used for data which we access at a frequency of once or twice in 2-3 months.

Data retrival time is upto 12-24 hrs.... exact time we will see in s3 console.

Retrival cost exist and is comparatively less than instant retrival

works in >=3 zones

**6. Glacier deep archive:-**

cheapest storage cost.

used for data which we access at a freq of once in 6 months, or twice a year.

Data retrival time is approx 24-48 hrs... exact time we will see in s3 console.

Retrival cost is more compared to Glacier archive.

works in >=3 zones.

**07/05/2024 Tuesday**

**Retrieving data ,Versioning and life cycle manager , life cycle rule, Static website hosting& Cross region replication**

[

{

"Condition": {

"HttpErrorCodeReturnedEquals": "404"

},

"Redirect": {

"HostName": "vel-buck-123.s3-website.ap-south-1.amazonaws.com",

"ReplaceKeyPrefixWith": "#!/"

}

}

]

**Static website hosting**

S3 host static websites & there is no need to be provision as no need to install server ..

**We have to create s3 bucket and and host static website**

**8/05/2024 wed**

**S3 cross region replication**

* What is cross region replication
* Amazon S3 now supports cross-region replication, a new feature that automatically replicates data across AWS regions.
* With cross-region replication, every object uploaded to an S3 bucket is automatically replicated to a destination bucket in a different AWS region that you choose.
* For example, you can use cross-region replication to provide lower-latency data access in different geographic regions.
* Cross-region replication can also help if you have a compliance requirement to store copies of data hundreds of miles apart.
* There is no additional charge for using cross-region replication. You pay Amazon S3’s usual charges for storage, requests, and inter-region data transfer for the replicated copy of data.
* Cross-region replication is available in the US Standard, US West (Oregon), US West (N. California), EU (Ireland), EU (Frankfurt), Asia Pacific (Tokyo), Asia Pacific (Singapore), Asia Pacific (Sydney), and South America (Sao Paulo) regions.
* Imp points
* For cross region replication to work, we need to make sure versioning is enabled
* Here whatever is updated to the source will be updated to the destination bucket.
* But vice-versa is not possible, means if we make changes in the destination bucket the source bucket is not updated.
* One more thing, if we delete object in source bucket, the object in the destination bucket is not deleted, and vice-versa

purpose

1) backup cross region

2) reduce latency

**s3 multi part upload**

s3 uplod data by breaking it into sevral package and upload it

to destination .. data should be equal or more than 250 gb

it reduces into upload time

**cli commond of s3**

aws cli

why s3 use in company

The ways to connect to aws account

**three ways**

1 GUI

2 CLI

3 API

aws --version

aws cli ..

we can connect to aws console using aws cli via laptop & via EC2 instance

1) using ec2 linux

log in with aws cli need acess key& secreate acces key

Access key = AKIA2UC27VNLP72MPCSF

Secreate access key= wjp7L2uU42vd/2/RtdpUOm4OVRMQfzCCUDHF9fpI

Access key =AKIA2UC27VNLHEPSTTHY

Secreate access key = InQZzhQXhpY3097PY540L1b8KZv8i4OgNWPF6WMs

aws configure

access key :

secrete access key :

default region : ap-south-1(to change the region)

default output format:table,json,text

aws ec2 help

aws ec2 describe-instances

ls-ltr

.aws >> credential

[xyz]

aws\_access\_key\_id=

aws\_secrete\_access\_key=

>>config

[profile xyz]

output=table,json,text

region=ap-south-1

aws ec2 describe-instances

aws ec2 describe-instances --profile xyz

aws ec2 stop-instances --instance-ids

aws ec2 stop-instances --instance-ids

assignment:-

**launched ec2 instance with the help of aws cli**

aws ec2 run-instances --image-id ami-019c34e6666f4275e --instance-type t2.micro --key-name test-a.pem –subnet-id subnet-0a1f3bd80d1b343b4-- security-group-ids sg-0a6177dafc174f616

aws ec2 run-instances --image-id ami-019c34e6666f4275e --count 1 --instance-type t2.micro --key-name test-a --security-groups launch-wizard-1

aws ec2 run-instances --image-id ami-12345678 --instance-type t2.micro --key-name my-keypair --subnet-id subnet-12345678 --security-group-ids sg-12345678 --tag-specifications 'ResourceType=instance,Tags=[{Key=Name,Value=MyInstance}]'

aws ec2 run-instances --image-id ami-019c34e6666f4275e --instance-type t2.micro --key-name test-a.pem –subnet-id subnet-0a1f3bd80d1b343b4-- security-group-ids sg-0a6177dafc174f616

**09/05/2024 Thursday**

**Commonds for s3 in aws cli**

aws s3 help

aws s3 ls ..(**to list bucket** )

uri(unique resource identifire) & url ()

aws s3 ls s3://bucket-name ..(**to list content of the bucket** )

**to make bucket**

aws s3 mb s3://bucket-name ...(in default region )

aws s3 ls s3://bucket-name --region us-east-1 ..(in particular bucket)

**to delete bucket**

aws s3 rb s3://bucket-name

aws s3 rb s3://bucket-name --force ..(to forcefully delete bucket)

aws s3 rm s3://bucket-name/object name...(to delete particular content of the bucket)

**to upload file**

aws s3 cp filename s3://bucket name

**to upload folder**

aws s3 cp folder name s3://bucket name --recursive ..(folder will not copy content of the folder will copy in recurcive mode)

**to download file**

aws s3 cp s3://bucket name/filename destination path

**to download folder**

aws s3 cp s3://bucket name/folder name destination path --recursive

**to move or rename object**

aws s3 mv filename s3://bucket name

aws s3 mv s3://bucket name/filename s3://bucket name/rename

**to change class of object**

aws s3 cp filename s3://bucket name --storage-class STANDARD\_IA

**for synchronise ec2 to s3**

aws s3 sync folder name s3://bucket name

**for synchronise s3 to ec2**

aws s3 sync s3://bucket name folder name

**10/05/2024 Friday**

**Theory of IAM (Identity & Access management)**

**11/05/2024 Saturday off**

**12/05/2024 Sunday**

**External mock by rohan joshi**

**13/05/2024 Monday**

**Off due to election**

**14/05/2024 Tuesday**

**Practicle of creating User and adding Policies to access aws services**

**15/05/2024 Wednesday**

**Programmatic Access, IAM Role , SNS**

To give programmatic access to user we need to configure it to the platform from where we run the commond i.e may from aws cli using laptop or using ec2 instance ..

To configure it we need account credentials i.e access key & Secreate access key

**Simple notification service (SNS)**

* Amazon Simple Notification Service (Amazon SNS) is a fully managed messaging service for both application-to-application (A2A) and application-to-person (A2P) communication.
* SNS is a fast, flexible, fully managed notification service, provided by amazon.
* It delivers the messaged to the subscripting endpoints such as emails, SMS, devices, other AWS services.
* We can send individual messages or fan out messages to large number of recipients.
* When a topic is created, Amazon SNS will assign a unique ARN (Amazon Resource Name) to the topic, which will include the service name (SNS), region, AWS ID of the user and the topic name.
* This is a region specific service

**benefits**

* Inexpensive
* Pay as you go.
* Reliable
* Instantaneous, push-based delivery.
* Web-based AWS Management Console offers the simplicity of a point-and-click interface.

**SNS topic**

* An Amazon SNS topic is a logical access point that acts as a communication channel.
* When publishers have information or updates to notify their subscribers about, they can publish a message to the topic – which immediately triggers Amazon SNS to deliver the message to all applicable subscribers.

**SNS Supported Transport Protocols**

* HTTP, HTTPS – Subscribers specify a URL as part of the subscription registration; notifications will be delivered through an HTTP POST to the specified URL.
* Email, Email-JSON – Messages are sent to registered addresses as email. Email-JSON sends notifications as a JSON object, while Email sends text-based email.
* SQS – Users can specify an SQS queue as the endpoint; SNS will enqueue a notification message to the specified queue (which subscribers can then process using SQS APIs such as ReceiveMessage, DeleteMessage, etc.)
* SMS – Messages are sent to registered phone numbers as SMS text messages

**SNS Supported Endpoints**

* Email ..Notifications
  + SNS provides the ability to send Email notifications
* Mobile Push Notifications
  + SNS provides an ability to send push notification messages directly to apps on mobile devices. Push notification messages sent to a mobile endpoint can appear in the mobile app as message alerts, badge updates, or even sound alerts
  + Supported push notification services
    - Amazon Device Messaging (ADM)
    - Apple Push Notification Service (APNS)
    - Google Cloud Messaging (GCM)
    - Windows Push Notification Service (WNS) for Windows 8+ and Windows Phone 8.1+
    - Microsoft Push Notification Service (MPNS) for Windows Phone 7+
    - Baidu Cloud Push for Android devices in China
* SMS Notifications
* HTTP/HTTPS Endpoints

SNS provides the ability to send notification messages to one or more HTTP or HTTPS endpoints, with the help of POST method.

* Etc ….

**Pricing**

* https://aws.amazon.com/sns/pricing/

**16/05/2024 Thursday**

**Cloud monitoring services Cloud trail and Cloudwatch**

**hello good morning**

**myself xyz . basically i belongs from sangamner ... currentlly**

**i am living in new sanghvi pune i have did my graduation in**

**bsc xhem from univercity of pune in 2021.. after that i was fascinated about**

**devops i switch my carrier as an devops eng.. i have did certificatio**

**in devops enginear .i have started my professional carrier**

**with bvg ind ltd on the project MEMS..**

**i was working as a**

**during this projest i have a**

**oppertunities to work on various aws services such as ec2 ebs**

**vpc . we also use cloud trail cloud watch**

**in our projest we particualry user amazon linus os**

**so i have also hands on shel scripting**

**apart from shell scriping i have aslo knowlege of cidd for this**

**particular purpose we jenkins (uses of tools need to mention)**

**so this is the short introduction of myself**

**name**

**work exp**

**comapny cient name**

**domain of project**

**platform use**

**servces use for-infra ,storage, monitoring, data base**

**operating system, shell script ,linux commond**

"Hello, my name is Akash Surya. I'm currently working as a DevOps Engineer at XYZ Technologies, where I've been involved in architecting and maintaining robust CI/CD pipelines and ensuring efficient cloud infrastructure management for the past 5 years.

In my role, I've successfully implemented infrastructure as code using tools like Terraform and Ansible, enabling seamless provisioning and configuration management across multiple environments. One of my key achievements was optimizing our deployment processes, reducing deployment times by 30% through automation and performance tuning.

I have a strong background in containerization technologies, including Docker and Kubernetes, and have led initiatives to containerize our applications, improving scalability and resource utilization. Additionally, I'm experienced in monitoring and logging solutions such as Prometheus and ELK stack, ensuring high availability and performance of our applications.

What excites me about DevOps is its transformative impact on software delivery, enabling faster innovation and improved collaboration between development and operations teams. Outside of work, I contribute to open-source projects related to DevOps tools and enjoy keeping up with emerging trends in cloud-native technologies.

I'm thrilled about the opportunity to discuss how my skills and experiences align with the goals of your team at [Company Name]. I'm looking forward to sharing more details about my background and how I can contribute to optimizing your DevOps processes."

This introduction provides a concise overview of the candidate's experience, key skills, achievements, passion for DevOps, and readiness for the interview. Feel free to adapt this example based on your specific experiences and the requirements of the role you're applying for.

Top of Form

**INTRODUCE YOURSELF AS A DEVOPS ENG. WITH ROLLES AND RESP.**

Good morning everyone, I'm Akash Suryavanshi, Originally from Sangamner but currently residing in New Sanghvi, Pune.

I graduated with a BSc in Chemistry from the University of Pune in 2021 and then delved into DevOps,

earning certification in the field.

As a DevOps engineer, my resp. include creating and managing virtual infrastructure

using AWS services like EC2, VPC, EFS, S3, IAM, and RDS,

Monitoring infrastructure with CloudTrail and CloudWatch,

Deploying new tools and technologies, Automating deployment processes with tools like Git, Maven,

Ansible, Jenkins, Docker, and Jira,

Managing source code with Git or GitHub and Maven,

Implementing Jenkins for continuous integration and continuous delivery,

Utilizing Docker for containerization,

Employing Ansible for configuration management,

Mastering advanced DevOps tools like Terraform and Kubernetes,

& leveraging Linux commands and shell scripting for projects using Amazon Linux.

That's a brief overview of my background and expertise.

Thank you!

Hello good morning all,

Myself akash suryavanshi. Basically i belongs from sangamner,currentlly

i am living in new sanghvi,Pune. I have did my graduation in

Bsc Chem from Univercity of pune in 2021.After that i was fascinated about

devops I started learning about devops.I have did certification

in Devops.Talking about rolles and resp. , as a devops engg.

my roles & resp. are

1)To create virtual infrastructure. To manage & maintining the virtual infra.

we use EC2, VPC ,EFS,S3,IIM & RDS this AWS services Particulary for creating virtual infra

2)To monitoring virtual infra.we particulary use cloudtrail and cloudwatch services for monitoring

3)Deploying new tool & technologies in various production and non-prod. env.

4)Automating this deployment process by implementing various devops tools such as git,maven,ansible,jenkins,docker & jira

5)For managing source we use Git or Github as a repository & for build and release source code we use Maven tool

6)For continuous integration & continuous delivery of software i.e CICD which we is the heart of devops we use Jenkins

6)For contenerization purpose we mainly use Docker,

7)For configuration managment we uses ANSIBLE

8)I also hands on the Advanced devops tools such as teraform use for infaution provisioning & kubernetes for orchestration

9)In our project we particulary use Amazon linux as a OS, so i have also hands on linux commond

and shell scripting.

so this is the short introduction of myself.

THANK YOU !