Cloud Computing

Date: .....
Page: .....

→ Cloud computing is the on-demand delivery of compute power, database, storage, applications, & other IT resources via the Internet with pay-as-you-go pricing.

→ AWS is a secure cloud platform that offers a broad set of global cloud-based products.

② Discuss the key benifits of Adopting Cloud Computing with suitable example.

→ Benifits are:

A) Pay-as-you-Go Pricing:
Instead of Investing heavily in physical hardware & data centers, you pay only for the resources you actually use.

B) Massive Economies of Scale:
Get Access to powerful IT resources at a lower cost than building own data center.

C) On-Demand Scaling:
With help of Cloud computing, Resources can be scaled up or down quickly to match demand. For ex: An online retailer can handle a sudden surge in traffic during a sale by automatically increasing server capacity, & then scale back down afterward.

Date: .....
Page: .....

(4) Speed & Agility:
Instead of waiting weeks for hardware & setup, cloud resources can be provisioned in minutes. This allows businesses to launch new projects or features rapidly, increasing their overall agility in the market.

(5) Reduced operational costs:
By moving to the cloud, no longer need to manage & maintain physical data centers. The cloud provider takes care of infrastructure management, security & maintenance, letting you focus on your core business operations.

(6) Global Reach:
Cloud platforms allows us to deploy applications globally in minutes. This means we can easily serve costumer around the world with low latency & high performance.

Example:
Consider a growing ecommerce company that experiences seasonal traffic spikes. By adopting cloud computing the company can:
(i) Launch its online store without a large upfront investment in hardware.
(ii) Scale its resources automatically during peak shopping seasons.
(iii) Reduce costs by paying only for extra capacity when needed.
(iv) Quickly expand to new international markets using glo

Date: .....
Page: .....

(3) How do cloud service models such as SaaS, PaaS & IaaS differ from each other?

(i) Infrastructure as a Service (IaaS):
- Offers virtual computing resources like servers & storage over the cloud.
- Full control over OS, application & other software layers.
- The provider manages only the basic hardware, storage & networking.
- Manually handle everything above that level, including installing & managing software.
- Scaling is also managed manually, such as adding virtual machines or storage when needed.
- Best suited for building custom infrastructure solutions.
- Pricing is typically pay-as-you-go based on the resources used.
- Ex: AWS EC2, Google Compute Engine, Digital Ocean.

(ii) ~~Product~~ Platform as a Service (PaaS):
- Provides a complete platform for developing & deploying applications.
- Control over apps, data & settings, but not the underlying hardware.
- The provider takes care of the operating system, runtime & infrastructure.
- You are responsible for building & managing your application code & data.
- It comes with built-in tools to help your app scale.

Date: .....
Page: .....

- Ideal for creating development platforms, APIs & databases.
- Costs depend on the amount of computing power, storage & other resources you use.
- Ex :- Heroku, AWS Elastic Beanstalk, Google App Engine.

(iii) Software as a Service (SaaS) :-
- Delivered over the Internet & fully managed by the provider.
- You only adjust basic settings & manage your data.
- The provider handles everything from the application to the underlying system.
- You simply input your data, access the app & use it.
- It automatically scales to meet demand.
- often used for end-user apps like email or CRM systems.
- Pricing is usually a subscription per user or per month.
- Examples: Microsoft 365, Zoom, Slack etc.

Date: .....
Page: .....

4) Discuss various cloud computing Deployment models with a suitable example.

① Public Cloud:
- Infrastructure & resources are owned & operated by third party providers.
- Services are delivered over the Internet & shared among multiple organizations.
- offers a pay-as-you-go model, meaning organization only pay for the resources they actually use.

Ex: Netflix uses AWS to host its streaming service, using EC2 for Computing, S3 for storage & cloudfront for content delivery, enabling global content distribution.

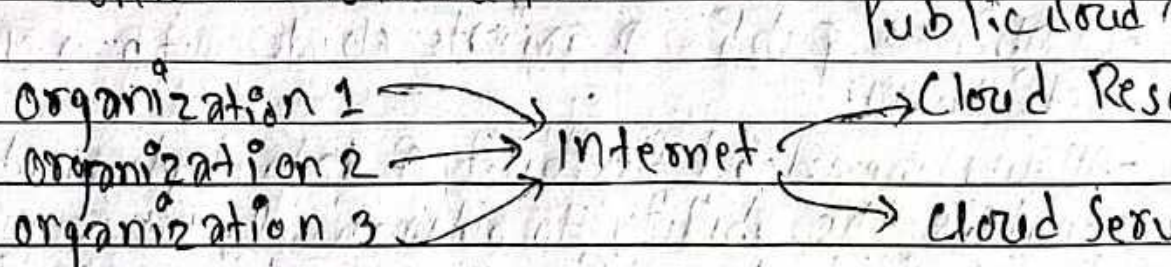


fig: P. C. P

② Private Cloud:
- Infrastructure is dedicated solely to organization.
- offers greater control, security & customization.
- Can be deployed either on-premises or hosted by a third-party provider (hosted private cloud).

Date :
Page :

Ex: A large bank maintains its own private cloud using VMware solutions to host sensitive customer data & core banking applications, ensuring security & compliance.

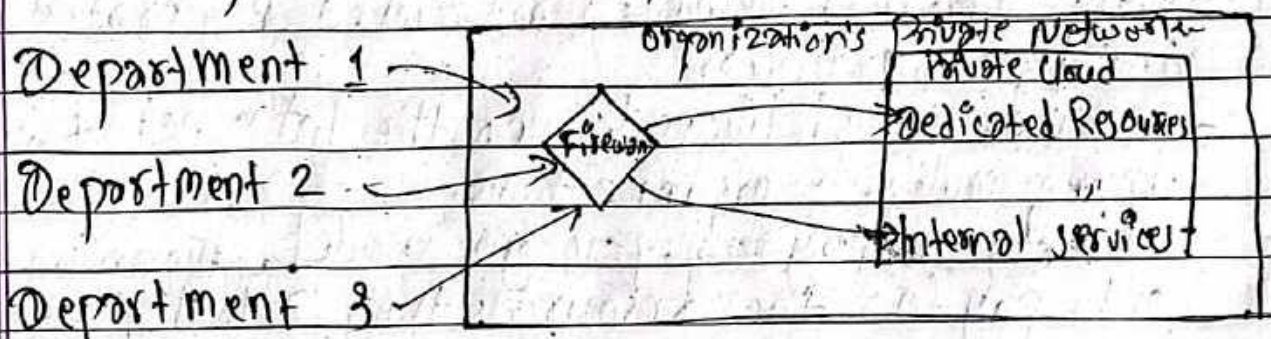
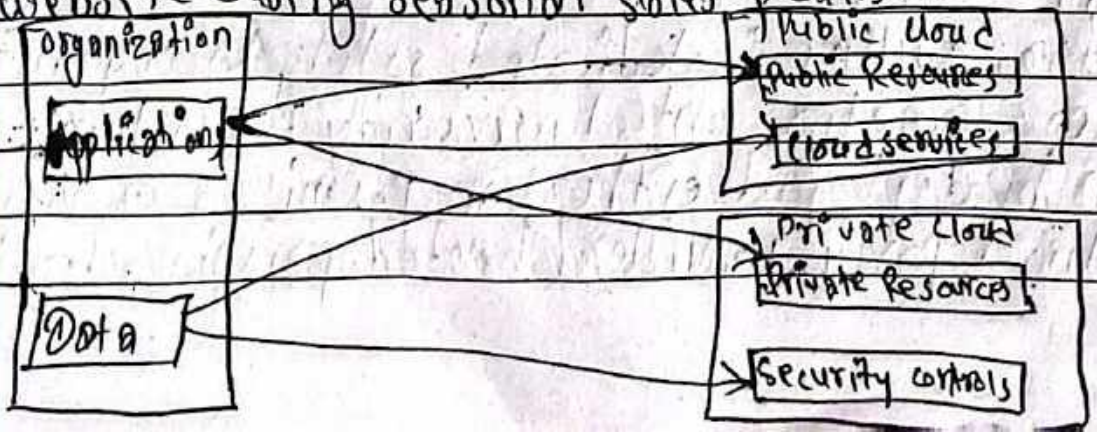


fig: Private cloud.

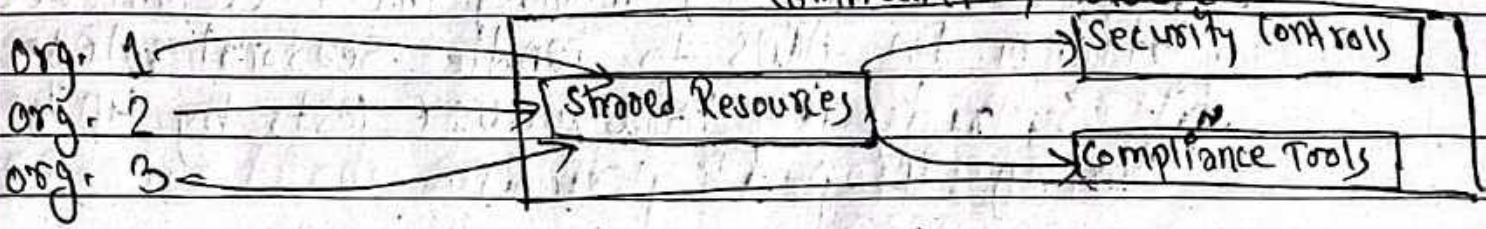
③ Hybrid cloud:
- Combines public & private clouds with orchestration b/n them.
- Allows workload flexibility & data deployment options.
- Enables the ability to automatically move workloads from private to public cloud during peak times.

Ex: A retail company keeps customer data in private cloud but uses AWS public cloud for their e-commerce website during seasonal sales peaks.

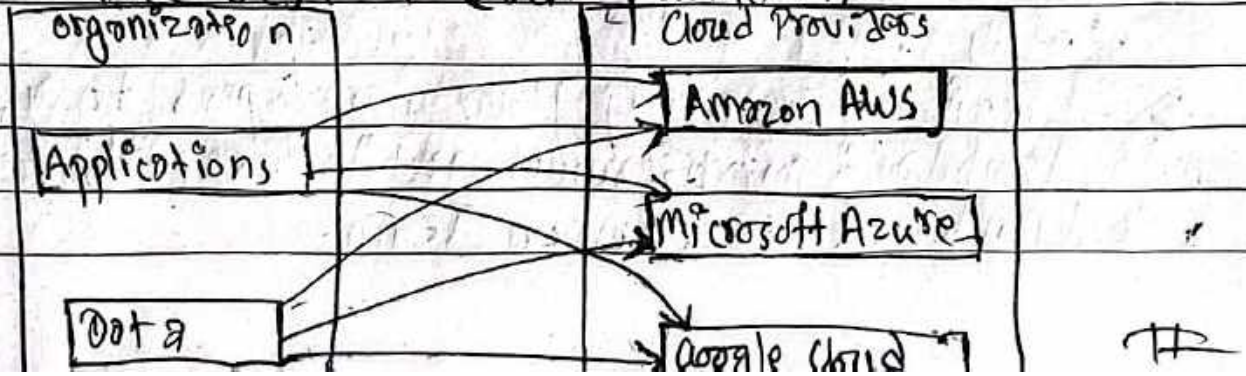


Date: .....
Page: .....

(4) Community Cloud:
- Infrastructure shared by several organizations with common concerns.
- costs are shared among community members.
- Often used by government agencies within a specific jurisdiction to share resources & maintain sector-specific compliance.
- Ex:- Healthcare providers share a HIPAA-compliant cloud infrastructure for medical records & Imaging services while splitting the costs.



(5) Multi Cloud:
- Uses multiple public cloud providers for different services.
- Reduces vendor lock-in & optimizes resources.
- Allows orgs. to pick best-in-class services from different providers for different needs.
- Ex:- A company uses GCP for machine learning, AWS for hosting, & Azure for development tools, leveraging the best of each platforms.



Date :
Page :

Q5) Discuss six AWS cloud Adoption Framework with a suitable example.

(i) Business Perspective:
- It is about justifying the Investment.
- The business perspective ensures that business & IT objectives meets the investment.
- Stakeholders include: Business owners, Business Managers, Finance Managers, Strategy stakeholders.

Ex: A retail company moving their ecommerce platform to AWS to handle seasonal sales spikes, reducing Infrastructure costs by 40% during off-peak periods.

(ii) People Perspective:
- The people perspective evaluates skills, requirements & roles in the organization.
- It is about making sure that you have right skills, competence, & processes in place to move to the cloud.
- The evaluation process helps us implement necessary changes or improvements.
- Roles: People Managers, Human resources (HR), staffing.

Ex: Training existing Java developers to use AWS lambda & microservices architecture, establishing a cloud center of excellence Team.

Date: .....
Page: .....

(3) Governance Perspective:
- It is about minimizing the Risk, compliance & security policies.
- Establishes frameworks for cloud operations.
- It helps us understand the gaps.
- Giving us an understanding of how to ensure processes & staff skills.
→ Roles Include: Chief Information Officer (CIO), Enterprise architects, Business analysts, Program managers, Portfolio managers.

Ex: Implementing AWS organizations to manage multiple accounts, setting up Aws. config rules to enforce (compliance standards like HIPAA.

(4) Platform Perspective:
- The platform perspective helps you deploy new cloud solutions.
- It also helps you migrate on-premises workload to the cloud.
- Roles in the platform perspective: CTO, solutions architects, IT mangers.

Ex: Designing a three-tier web application using Amazon EC2 for compute, Amazon RDS for database, & Amazon S3 for static content.

Date :
Page :

(5) Security Perspective:
- The security perspective ensures that the organization's security objectives are met.
- Ensure data protection & cybersecurity.
- Implements identity management & access controls.
- Roles (CISO, IT security analysis, IT security managers).
Ex:- Using AWS IAM roles, AWS WAF for web application security, & AWS shield for DDos protection.

(6) Operations Perspective:
- The operations perspective is about running the business.
- Ensuring that the business operations meet the expectations.
- It includes a year-to-year, quarter-to-quarter, & day-to-day business.
- It helps define the necessary changes needed for successful cloud adoption.

- Roles: IT operations managers, IT support managers.

Examples: Setting up Amazon CloudWatch for monitoring, implementing automated backup solutions, & using AWS systems Manager for patch management.

Date: .....
Page: .....

1) How does cloud computing impact the overall cost structure for businesses compared to traditional IT infrastructure?

Traditional IT costs:
(i) High upfront investments in hardware & infrastructures.
(ii) Ongoing maintenance & upgrades of physical equipment.
(iii) Need for dedicated IT staff for management.
(iv) Expensive to scale up infrastructure quickly.
(v) High energy consumption for data centers.
(vi) Software licenses require large upfront payments.
(vii) Difficult & expensive to implement disaster recovery.

Cloud Computing Costs:
- Pay-as-you-go pricing model reduces upfront expenses.
- No need for inhouse infrastructure management.
- Scalable resources to match business growth.
- lower energy costs as infrastructure is shared.
- Built in disaster recovery & high availability features.
- Subscription based software reduces licensing complexity.
- Remote accessibility & global reach at lower cost.

For example:
Traditional IT needs high upfront costs for hardware, while cloud computing offers a pay-as-you-go model. startups can use cloud services without heavy investment in equipment. Traditional IT requires dedicated staff,

Date :
Page:

while cloud providers manage Infrastructure. Scaling traditional IT is costly, but cloud offers instant scalability. Software licensing in traditional IT is expensive, while cloud uses flexible subscriptions.

Date: .....
Page: .....

7) Discuss the importance of Identity & Access Management in cloud Computing with a suitable example?

⇒ Importance of IAM in cloud Computing:

- Ensures only authorized users (human or machine) can access specific cloud resources, preventing unauthorized access & potential data breaches.

- Implements the principle of least privilage, granting users only the minimum necessary permission to perform their assigned tasks, thus limiting the impact of compromised accounts or malicious insiders.

- Centralizes Identity & access management across the cloud environment.

- Helps organizations meet regulatory compliance requirements (e.g., GDPR, HIPAA).

- Streamlines user access with single Sign-On (SSO), boosting productivity.

- Mitigates risks like data breaches, insider threats, & compliance violations.

- Protects sensitive data by controlling access & modification permission.

Date :
Page :

- Automate user provisioning & deprovisioning, reducing administrative costs.

- Enables secure collaboration with controlled access to shared resources.

- Scales with the cloud environment & adapts to changing business needs.

Ex:-
Imagine a company storing sensitive customer data in a cloud-based database. Without IAM, any employee with access to the cloud environment could potentially access & manipulate this data, leading to data breach. With IAM, the company can define specific roles & permissions, ensuring that only authorized personnel (e.g. database administrator) can access the customer data, & even then, their access can be limited to specific actions. (e.g. read-only access for customer support).

Date: .....
Page: .....

8) Discuss the shared Security Model in Cloud Computing with a suitable Example.

⇒ The shared security model in c.c refers to the division of security responsibilities bet^n the cloud service provider & customer.

Both parties share the responsibility for securing data, applications, & services within the cloud environment but each has specific rules & obligations.

⇒ Cloud provider responsibilities:-
- Secures the cloud Infrastructure (physical servers, data centers, networking).
- Manages network security, firewalls, & physical security of data centers.
- Ensures availability & disaster recovery of the cloud Infrastructure.

⇒ Customer's responsibilities:
- Secures data (encryption at rest & in transit).
- Manages access controls (user permissions, identity management).
- Configures & updates OS & applications.
- Monitors & handles security patches & vulnerabilities

⟶

Date :
Page :

Example:
For a business using AWS:
- AWS (Provider) secures the physical infrastructure network & hypervisors.
- The customer secures the application, manages data encryption, & configures access controls.

Responsibilities vary depending on the service model (IaaS, PaaS, SaaS). In IaaS, the customer has more control over security than in SaaS, where the provider handles more security tasks.

Date: .....
Page: .....

10) Discuss the Amazon EC2 instance lifecycle with a suitable diagram.

Amazon EC2 Instance:
An Amazon EC2 instance transitions through different states from the moment you launch it through to its termination.

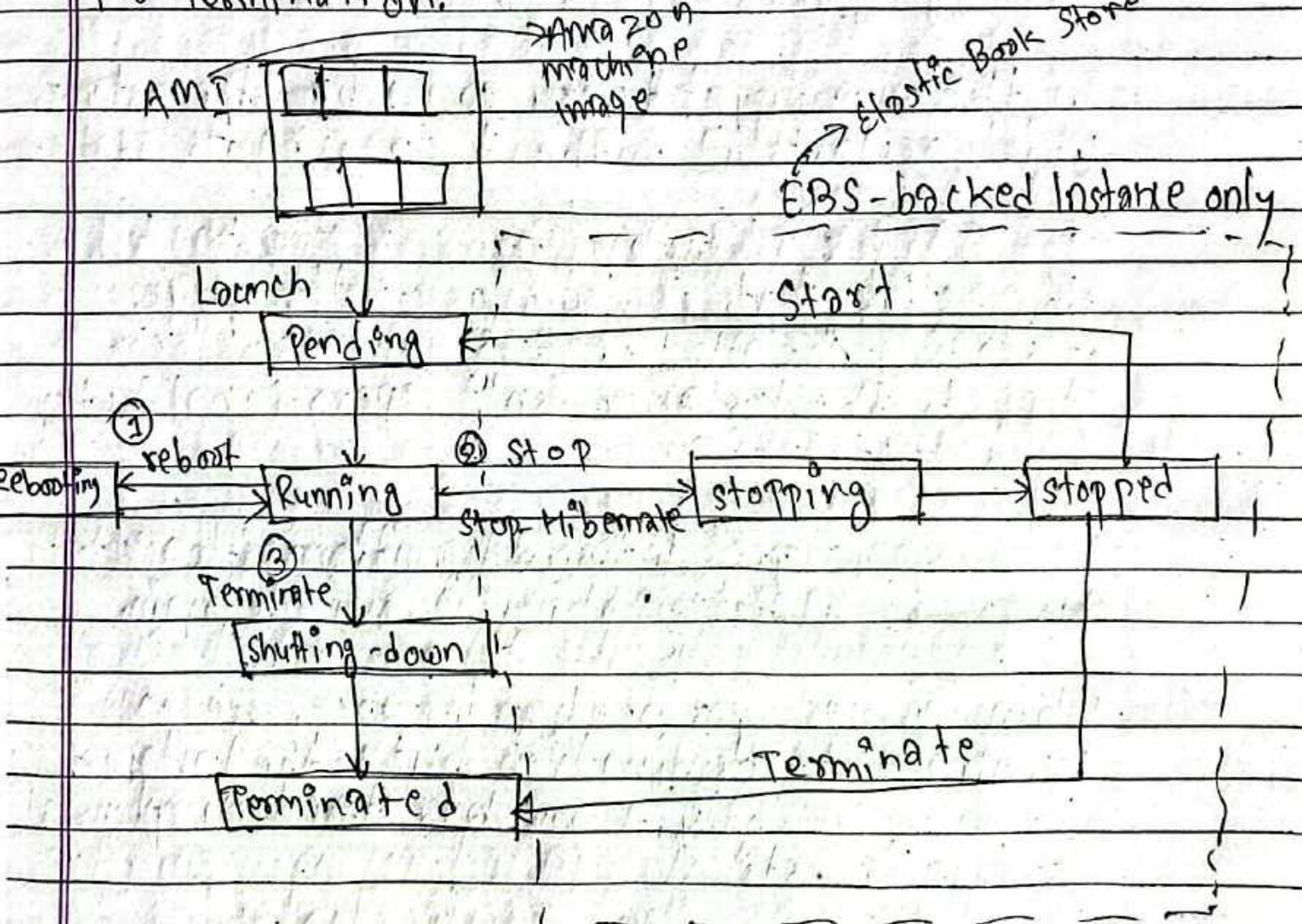


fig: Amazon EC2 Instance

Date: .....
Page: .....

- When you launch an EC2 Instance, it first enters the pending state as AWS allocates the required resources & initializes the Instance.

- Once initialization is complete, the Instance transitions to the running state where it is fully operational & accessible.

- In the running state, you can reboot the Instance, which restarts it without changing its state.

- For EBS backed Instance, you can stop the Instance; this action shuts it down gracefully while preserving the attached EBS volumes. When stopped, the Instance isn't operational but can be restarted later.

- Restarting a stopped Instance transitions it back to the running state, making it available again.

- When we no longer need an Instance, we can terminate it. Termination shuts the Instance down permanently, & any data stored on ephemeral storage is lost.

- There is a brief shutting-down phase during termination, making the transition from running to terminated states.

Date: .....
Page: .....

Q) 6) Buddha Air is planning to move its entire Infrastructure from On-premises to Cloud based Infrastructure. Identify & Compare the total cost of Ownership (TCO) for on-premises & cloud based ~~information~~ Infrastructure. The total cost of ownership should include Server costs, Storage Cost, Network Cost & IT labor cost.

⇒ TCO Comparison: On-Premises Vs. Cloud-Based Infrastructure

↳ TCO is a financial estimate that helps organizations evaluate the direct & Indirect costs of a product or system.

Comparison on basis of:

(1) Server Costs:

(A) On-Premises:
- High upfront for physical servers.
- value decreases over 3-5 years.
- ongoing costs for repairs & support.

(B) Cloud Based:
- Regular fees based on usage (pay-as-you-go).
- Flexible scaling, but can increase with resource demand

(2) Storage Costs:

(A) On-premises:
- Purchase of physical storage devices (e.g. SAN, NAS).
- Additional costs for backup systems.
- Ongoing storage system upkeep.

Date: .....
Page: .....

(B) Cloud-Based:
- Charges based on the amount of stored data.
- Backup & Redundancy often included, but may have extra charges.

(3) Network Costs:
(A) On-Premises:
- Expense for routers, switches & bandwidth
- Ongoing network system maintenance.

(B) Cloud-Based:
- charges for data ingress & egress.
- Maybe included or billed seprately.

(4) IT labor costs:
(A) On premises:
- Salaries for staffs for managing Infrastructure
- Ongoing Training for staff.
- Potentially higher labor costs during maintenance & upgrades.

(B) Cloud Based:
- often fewer staff required for management
- Focused on cloud specific tools.

Date: .....
Page: .....

On-Premises Infrastructure	Cloud based Infrastructure
(i) High initial purchase, maintenance & depreciation.	(i) Subscription fees, flexible scaling.
(ii) High hardware & maintenance costs.	(ii) Pay-per-use, potentially lower.
(iii) High Infrastructure & bandwidth costs.	(iii) Variable data transfer fees.
(iv) High labor cost due to staffing & training.	(iv) low labor cost, but skilled staff still needed.

Date: .....
Page: .....

Q) 11) Discuss the four pillars of cost optimization for Compute services with a suitable example.

⇒ 1) Right-sizing:

- Focus:
  A) Selecting the correct Instance types, sizes, & configurations.
  B) Avoiding over-provisioning or under-utilization of resources.

- Best Practices:
  i) Use AWS Compute Optimizer to get recommendation on instance types.
  ii) Monitor CPU, memory, & network usage with Amazon CloudWatch
  iii) Scale workloads based on demand with AWS Auto Scaling.

Example:
A startup initially chooses m5.large EC2 instances for its web application but later finds that t3.medium instances provide the same performance at a low cost. By right-sizing, they save 30% on compute expenses.

Date: .....
Page: .....

(2) Increase Elasticity:

Focus:
- Dynamically adjusting compute resources based on actual workload demand.
- Avoiding the need to keep instances running 24/7 when not required.

Best Practices:
- Use AWS Auto Scaling to add or remove instances based on traffic.
- Leverage Amazon EC2 spot instances for fault-tolerant workloads.
- Implement AWS Lambda for event-driven apps to avoid paying for idle resources.

Ex:-
A video streaming company experiences peak demand in the evening & lower traffic during the day. By using AWS Auto Scaling, it automatically scales up EC2 instances in the evening & scales down at night, reducing costs by 40%.

(3) Choose the Right Pricing Model:

Focus:
- Selecting the best pricing options for different workload types.
- Balancing cost & flexibility.

Date: .....
Page: .....

Best Practices:
- Use On-Demand Instances for unpredictable workload types.
- Reserve capacity with AWS Reserved Instances for long-term savings.
- Use EC2 Spot Instances for batch processing & non-critical workloads.

Ex:
A data analytics firm has a steady workload that requires EC2 Instances 24/7. Instead of paying On-Demand pricing, they purchase Reserved Instances (RI.) for a 1 year term, saving up to 72% on Compute costs.

(4) Optimize Storage & compute together
Focus:
- Reducing costs by selecting the right combination of compute & storage solutions
- Using efficient data processing & caching mechanisms.

Best Pratices:
- Store Infrequently accessed data in Amazon S3 Glacier instead of high-cost EC2 Storage.
- Use Amazon RDS or Aurora Instead of

Date: .....
Page: .....

running databases on EC2.
- Optimize database performance with Amazon Elasticache

Example:
An AI company running deep learning workloads reduces compute costs by using Amazon S3 for storing training datasets instead of keeping them on expensive EBS volumes attached to EC2 instance. This saves 60% on storage costs.

VPC - Virtual Private cloud
NACLs - Network Access & control List
EC2 - Elastic Compute cloud
NAT - Network Address Translation

Date:
Page:

Q) 9) You have a small business with a website that is hosted on an Amazon EC2 instance. You have customer data that is stored on a backend database that you want to keep private. You want to use Amazon VPC to set up a VPC that meets the following Requirements:

a) Your web server & database server must be in separate subnets.
b) The first address of your network must be 10.10.1. Each subnet must have 254 total IPV4 addresses.
c) Your customers must always be able to access your web server.
d) Your database server must be able to access the internet to make patch updates.
e) Your architecture must be highly available & use at least one custom firewall layer.
f) Design the architecture with brief explanation of each services used.

→ Diagram का basis पर explanation है।

1) VPC & subnet Configuration:
↳ Created VPC with CIDR block 10.10.1.0/24
↳ Divided into 4 subnets across 2 Availability Zones (AZs):
i) Public Subnets: (254 Addresses each)
10.10.1.0/25 & 10.10.2.0/25
ii) Private subnets:
10.10.1.128/25 & 10.10.2.128/25
↳ Each Subnet placed in different AZ for high availability.

Date: .....
Page: .....

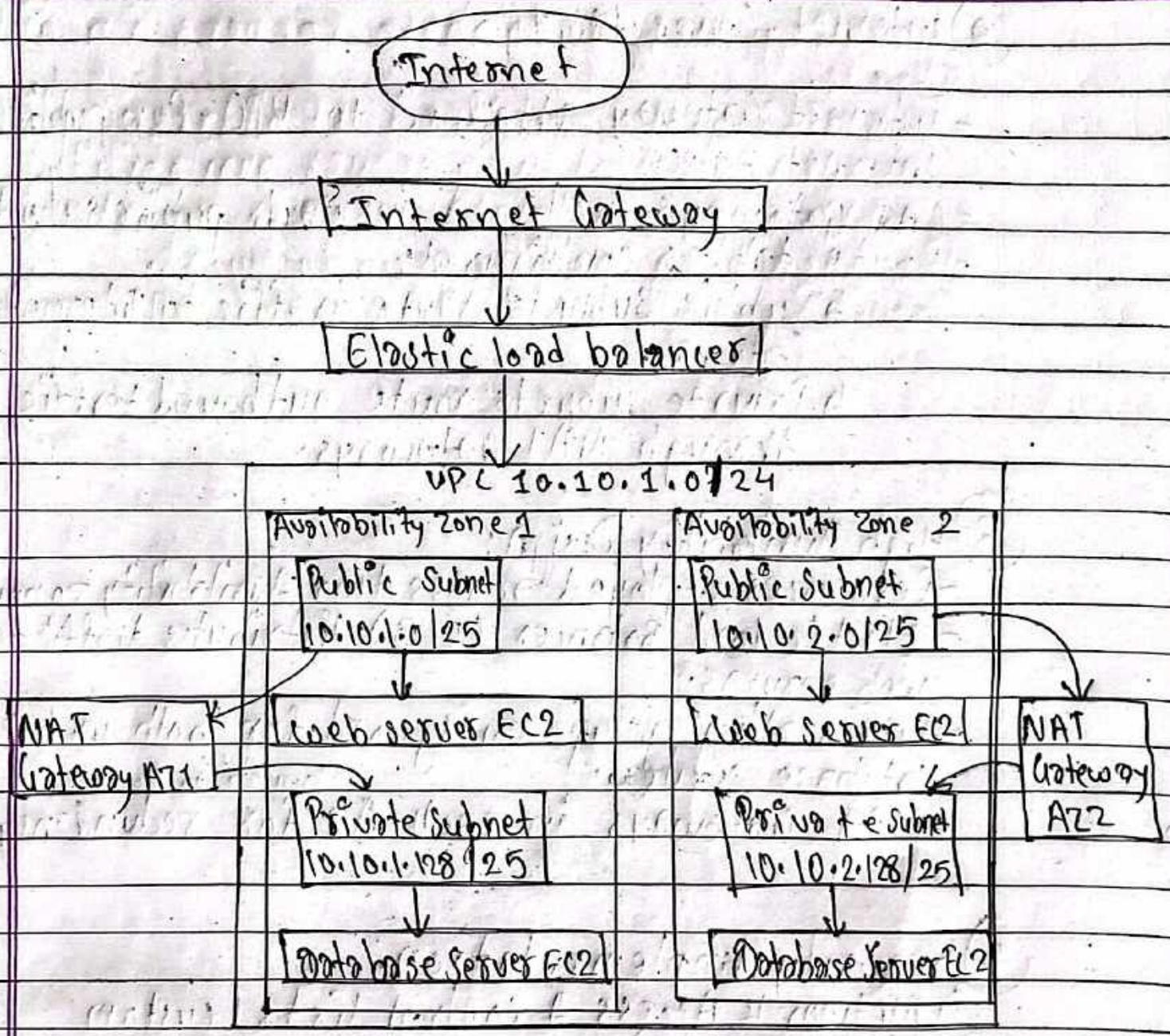


fig: VPC Network architecture Diagram

P.T.O →

Date: .....
Page: .....

(2) Internet Connectivity:

- Internet Gateway attached to VPC for public Internet Access
- NAT Gateways deployed in each public subnet
- Route tables configured:
  - ↳ Public subnets route traffic to Internet Gateway.
  - ↳ Private subnets route outbound traffic through NAT Gateways.

(3) High Availability Design:

- Resources deployed across two Availability zones.
- Elastic Load Balancer (ELB) distributes traffic to web servers.
- Auto scaling groups configured for both web & database servers.
- NAT Gateways in each AZ for redundancy. → Availability zones.

(4) Security Implementation:

(1) Network Access & control lists (custom firewall layer):

- Public subnet: Allow Inbound HTTP/HTTPS from anywhere
- Private subnet: Allow Inbound only from public subnet.

Date:
Page:

(iii) Security Groups:
- Web servers allow Inbound HTTP/HTTPS.
- Database servers allow Inbound only from web servers.
- Outbound rules for database patches through NAT gateway.

(5) Service Placement & Access:

(i) Web servers placed in public subnets:
- Directly accessible through ELB.
- Auto scaling Group maintains availability.

(ii) Database servers in private subnets:
- No direct Internet Access.
- Can access Internet for patches via NAT Gateway.
- Protected from direct external access.

(6) Elastic load Balancer (ELB):
- Distributes Incoming traffic
- Health checks on web servers
- SSL termination point.

Hence, This architecture successfully meets all requirements by:

Date: .....
Page: .....

- seperating web & database servers in different subnets.
- Maintaining the specified IP addressing scheme.
- Ensuring constant web server availability.
- Enabling secure database server updates.
- Implementing multiple security layers.
- Providing high availability across all components.

Date: .....
Page: .....

12) Differentiate containers with virtual machines with a suitable architecture.

⇒ Containers & virtual Machines are both technologies that allow multiple isolated environments to run on a single physical host. But they differ in their architecture, performance & use cases.

Containers	Virtual Machines (VMs)
(i) Isolate at the application level.	(i) Isolate at the hardware level.
(ii) Share the host OS kernel.	(ii) Run a full OS per VM.
(iii) Light weight, smaller in size.	(iii) Larger due to the full OS & resources
(iv) Faster startup, lower overhead.	(iv) Slower startup, higher resource usage
(v) Lower overhead, as it shares the host OS.	(v) Higher overhead, as each VM includes its own OS.
(vi) Highly portable, easy to move between environments.	(vi) Less portable, requires matching hypervisor.
(vii) Microservices, DevOps, testing & CI/CD.	(viii) Running multiple different OSes, legacy applications.

→ Architecture:

(1) Containers:
- Host OS: Runs directly on the host system's kernel.
- Container Engine (e.g. Docker) Manages & runs containers.
- Containers: Each container runs a specific application with only the necessary libraries & dependencies, sharing the OS kernel.

Date :
Page :

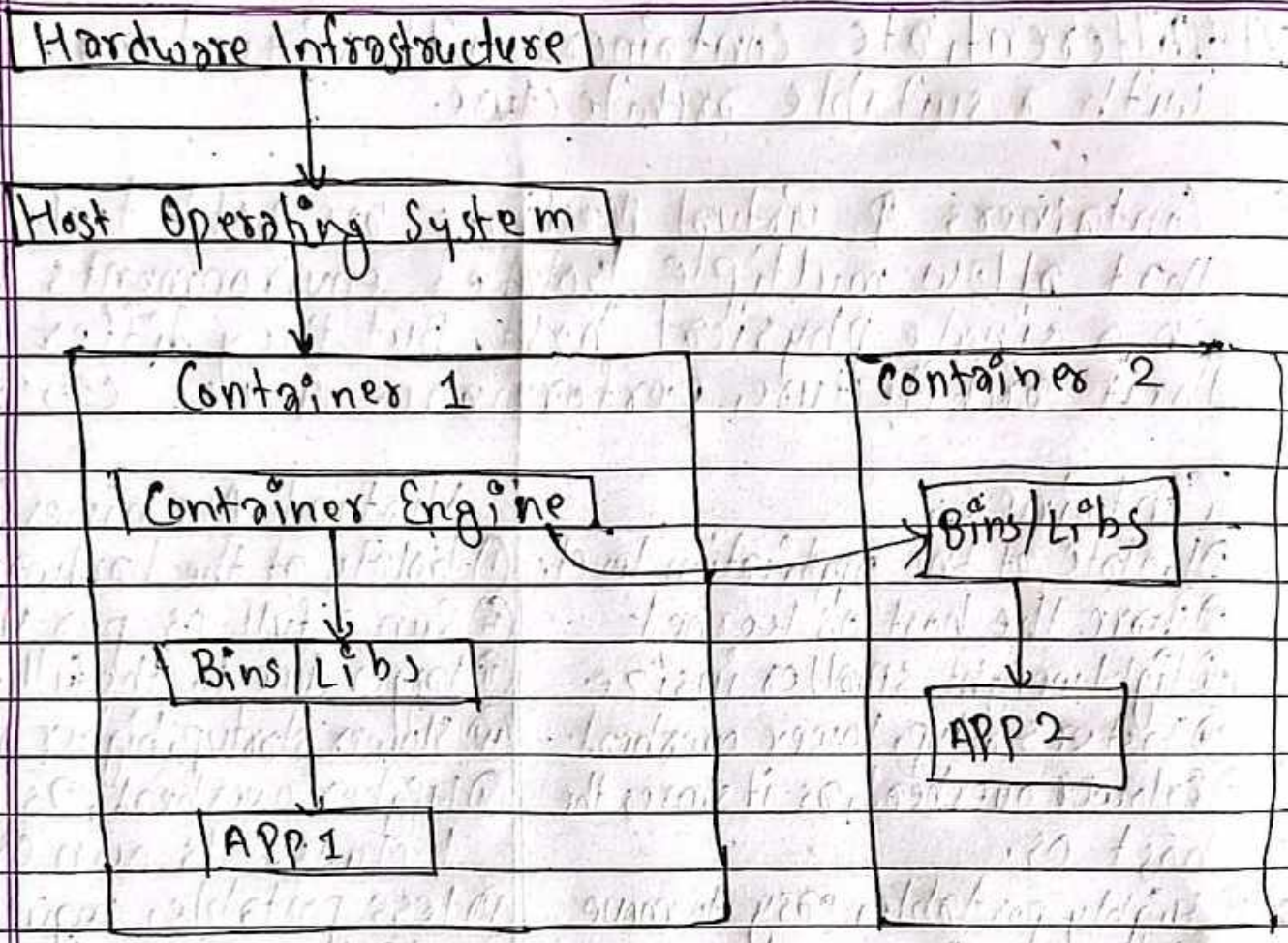


fig: Container Architecture.

- Containers are faster, lighter & more efficient for microservices & application isolation.

(2) VMs:

- Host OS: Runs the hypervisor (ex: VMware, Hyper-V).
- Hypervisor: Virtualizes the hardware, allowing multiple guest OSes to run independently.
- Guest OS: Each VM runs its own full OS, along with the applications.

fig: Virtual machine Architecture.

- VMs provide full isolation with more resources, ideal for running different OSes or legacy apps.

Date: .....

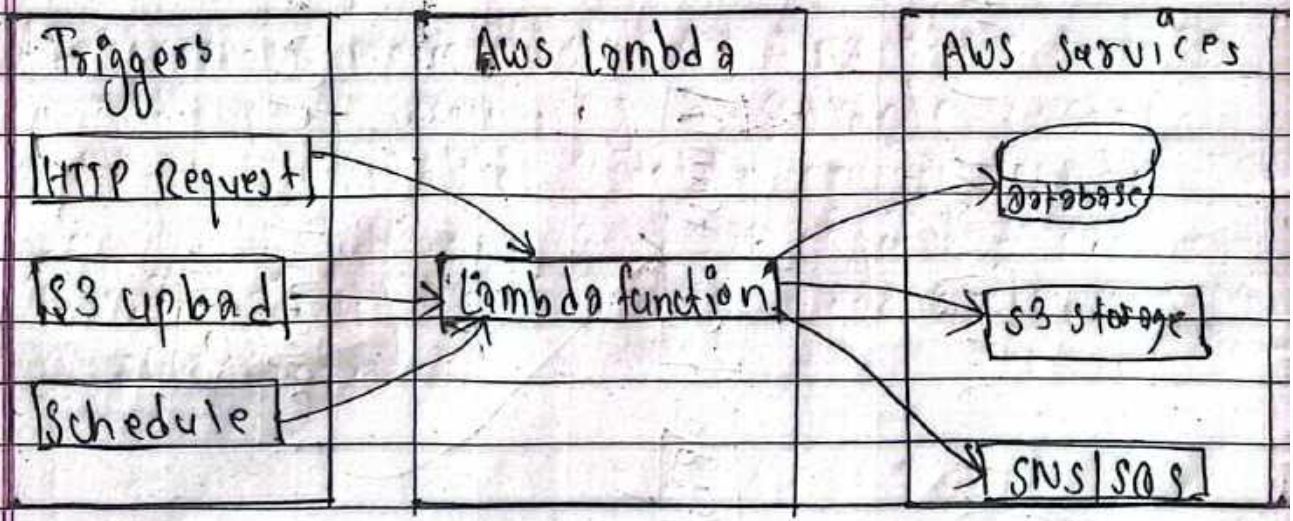
Page: .....

Date : .....
Page: .....

Q)13) Serverless compute plays a vital role in computing environment. Justify with reference to AWS Lamda Function.

⇒ Serverless computing is a critical component of modern cloud computing, enabling developers to focus on writing code without worrying about infrastructure management.

AWS lambda is a prime example of a serverless compute service that allows users to run code without provisioning or managing servers.



Justification:

① No Server Management:
AWS Lambda automatically provisions & scales infrastructure based on demand. Users do not need to manage servers, operating systems, or software updates.

Date: .....
Page: .....

(2) Event driven Execution:
lambda functions are triggered by various AWS services (eg. S3, API gateway, DynamoDB, Cloudwatch). This event driven Approach optimizes resource usage & efficiency

(3) Auto scaling:
lambda scales automatically to handle workloads. If multiple function Invocations occur simultaneously, AWS provisions additional compute resources dynamically.

(4) Cost-Efficiency (Pay-as-you-go):
Users are billed for the actual execution time of the function, measured in miliseconds. There is no charge when the function is idle, making it cost-effective compared to traditional server-based solutions.

(5) High availability & fault Tolerance:
AWS Lambda runs functions across multiple availability Zones, ensuring resilience & fault tolerance without requiring manual configuration.

) Seamless Integration with AWS Ecosystem:
lambda Integrates with various AWS services such as API Gateway, S3, DynamoDB, SNS & more making it an essential part of cloud native architectures.

Date: .....
Page: .....

(7) Security & Compliance:
AWS manages the underlying security, including patching & updates. Lambda also allows users to define permissions using IAM roles, ensuring secure access to resources.

(8) Rapid Development & Deployment:
Developers can focus on writing business logic rather than managing infrastructure. Code updates can be deployed quickly using CI/CD pipelines.

AWS Lambda exemplifies the power of serverless computing by providing a scalable, cost-efficient & highly available compute environment. It eliminates infrastructure management overhead, making cloud applications more agile, responsive & efficient.

Date :
Page :

14) Discuss the Importance of Block Storage & object Storage in cloud computing Architecture. How elastic Block storage can be attached to specific EC2 Instance running windows/Linux operating system?

⇒ Cloud computing provides various storage solutions, primarily Block Storage & Object Storage each serving different use cases.

Block Storage Importance:
- Provides fast data access for databases & critical apps like MySQL & oracle.
- Delivers consistent performance needed for business applications.
- Enable quick file modifications ideal for operating system.
- Supports multiple users working on same files simultaneously.
- Ensures accurate data recording for financial transactio
- Creates bootable drives for quick system recovery.
- Allows easy volume resizing as storage needs grow

Object Storage Importance:
- Stores large amounts of media files cost-effectivel
- Maintains data & copies across locations for reliability.
- Uses tags to organize & find files easily.

Date: .....
Page: .....

- Integrates well with web applications & browsers.
- Keeps track of file versions automatically.
- Moves older data to cheaper storage tiers.
- Delivers content globally through content networks.

→ Attaching Elastic Block Storage (EBS) to an EC2 Instance (Linux / Windows)

① Method 1:
- Navigate to the AWS Management Console → EC2 Dashboard
- Click volumes under EBS & create new EBS volume.
- Attach to EC2 Instance
- Run below commands in EC2 Instance.

```
#df -h (check available disk space)
#lsblk (list all available block devices)
#mkfs -t xfs /dev/xvdf (format new volume)
#file -s /dev/xvdf (verify the file system)
#mkdir -p /apps/volume/newvolume (create a mount directo
#mount /dev/xvdf /apps/volume/newvolume
#df -h
      ↳ Check if the volume is mounted.
→ mount the volume
```

Date : .....
Page: .....

→ Method 2:

lsblk (verify available storage)
↳ Output:

NAME	MAJ:MIN	RM	SIZE	RO	TYPE	MOUNTPOINTS
xvda	202:0	0	8G	0	disk	
├─xvda1	202:1	0	8G	0	part	/
├─xvda127	259:0	0	1M	0	part	
└─xvda128	259:1	0	10M	0	part	/boot/efi
xvdb	202:16	0	10G	0	disk	

sudo mkfs -t ext4 /dev/xvdb
↳ format the volume as EXT4.

sudo mkdir /mnt/mydata
↳ create a mount directory.

sudo mount /dev/xvdb /mnt/mydata
↳ mount the volume.

Hence, the method above demonstrate how to attach, format, & mount EBS volumes in EC2 Instance.

Date: .....
Page: .....

Q17) Discuss the role of Security Group in Cloud Computing Environment with a suitable example.

⇒ A security Group acts as a virtual firewall for cloud resources like (virtual machines) to control inbound & outgoing traffic.

It defines rules to allow/block access based on IP addresses, protocols (e.g. HTTP, SSH), & ports.

Key Roles:
- Restricts unauthorized access to Instances (e.g. blocking external traffic except on specific port).
- Automatically allows return traffic if the outgoing/incoming request is permitted (no need for reverse rules).
- Applied directly to cloud resources (e.g. EC2 Instances in AWS) for granular control.
- Works with Network ACLs for multi-layered security.
- The same security group can be attached to multiple VMs, making it simple to manage & update the rules.

Date: .....
Page: .....

Example: Securing a Web Server:
Imagine a company hosts a website on a cloud server (e.g., AWS EC2). The security Group rules could be:

- Allow HTTP (port 80) & HTTPS (port 443):
  ↳ lets users access the website from any IP.
- Allow SSH (port 22):
  ↳ Only from the admin's office IP (e.g. 203.0.113.1) for server management.
- Deny all other traffic:
  ↳ Blocks unnecessary access (default behavior).

Working:
- The website is accessible to users, admins can securely manage the server, & hackers are blocked from unused ports.
- Rules are easy to update (e.g: adding a new IP for SSH access).

Security Groups are essential for securing cloud resource by filtering traffic, reducing attack risks, & ensuring compliance. Properly configured rules balance accessibility & security.

Date: .....
Page: .....

Q)15) Your company have two VPC running in different AWS Regions. Each VPC has Private & public subnets. You are required to exchange information between Instances running in private subnets. Design the VPC peering with appropriate services.

⇒ When we have two VPCs in different AWS regions, each with public & private subnets, we can enable private communication between instances in the private subnets using an Inter-region VPC peering connection.

Designing the setup:

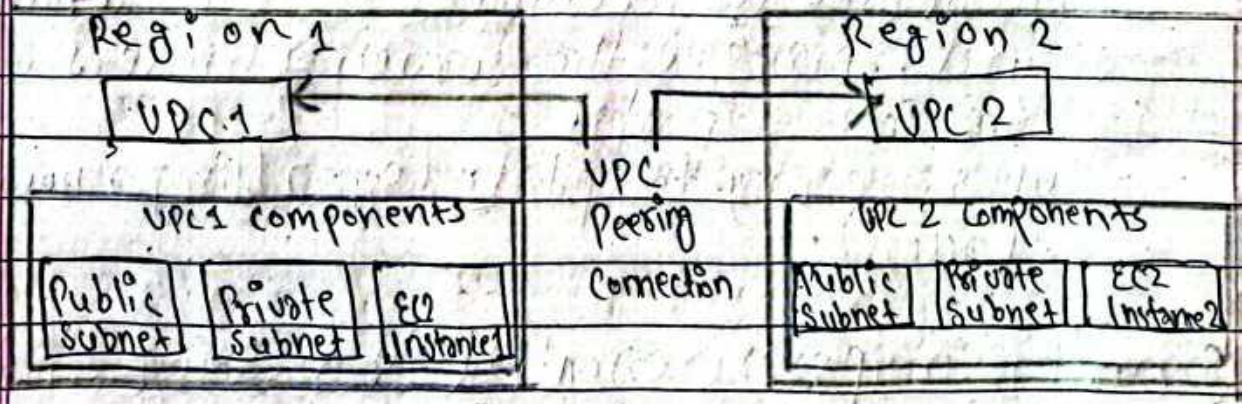


fig: VPC peering Connection.

→ Steps (contd from next page):

(v). (Optional) Enable DNS Resolution:
- If the app rely on DNS, enable DNS resolution for the peering connection so that private DNS names resolve to the correct IP addresses.

Date: .....
Page: .....

Steps:

(i) Ensure Unique CIDR Blocks:
Verify that the IP addresses ranges (CIDR blocks) of both VPCs do not overlap. This is essential for the peering connection to work correctly.

(ii) Create the VPC peering Connection:
- In the region of the requester VPC, initiate a VPC peering connection & specify the VPC ID of the peer in the other region.
- Switch to the accepter VPC's region & accept the peering request. This connection is managed through the AWS VPC service.

(iii) Update Route Tables in Private Subnets:
- In each VPC's private subnet route Tables, add a route that directs traffic destined for the other VPC's CIDR block to the peering connection.
- This ensures that the Instances in one VPC can reach the Instances in the other VPC using private IP addresses.

(iv) Configure Security groups:
- Modify the security groups associated with the Instances in the private subnets to allow the necessary Inboud & outbound traffic (based on IP & port) from the remote VPC's CIDR range.

Date: .....
Page: .....

Q)16) Your company has on-premises Data Center & Cloud Data Recovery Center. The private network of on-premises Data Center need to communicate with private subnet of cloud Data Recovery Center. Design the architecture of site to site VPN in AWS cloud & on-premises Data Center to enable secured private communication between two private subnets.

⇒ Designing a site-to-site VPN between an On-Premises Data Center & a cloud Data Recovery Center:

① AWS Cloud Setup (Data Recovery Center):
- Create a VPC in the AWS region where the Cloud Data Recovery Center is hosted. Within this VPC, setup one or more private subnets where recovery resources reside.
- Attach a virtual private gateway to the VPC. This gateway serves as the AWS side endpoint for the VPN connection.

② On-Premises Setup (Data Center):
- Use a hardware or software VPN device at your on-premises data center. Configure this device as Customer Gateway. This device must

Date : .....
Page: .....

support IPSec to handle encrypted traffic.

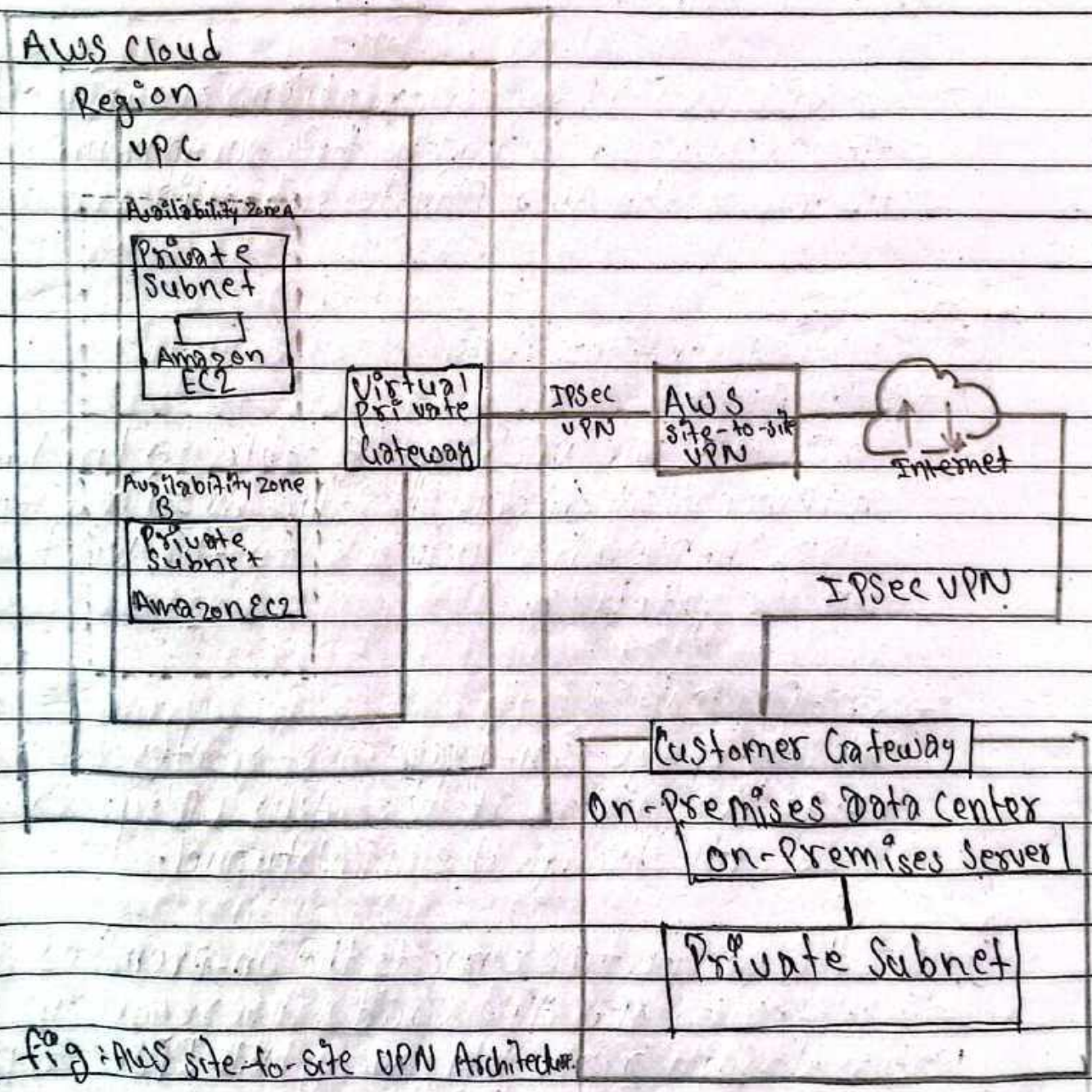


fig: AWS site-to-site VPN Architecture.

(3) VPN Connection:

In the AWS console, create a site-to-site VPN connection that links the virtual private Gateway (VGW) to the on-premises Customer Gateway (CGW). AWS automatically

Date: .....
Page: .....

provides two VPN tunnels for redundancy & high availability.

- The VPN uses IPSec encryption, ensuring that data exchanged between the on-premises network & the AWS private subnet is secure.

(4) Routing Configuration:

→ In AWS:

↳ Update the VPC's route table to include a route that directs traffic destined for the on-premises network to the virtual private gateway.

→ On-premises:

↳ Configure on-premises router to send traffic aimed at the AWS VPC's private subnet through the VPN tunnel.

Hence, this architecture connects the on-premises data center to the AWS cloud data recovery center through a secure, encrypted IPSec VPN. By setting up a virtual private gateway in AWS & a customer gateway on-premises, & by configuring the appropriate routing, we can enable secured communication between the two private networks.

Date: .....
Page: .....

Q) 22) Discuss Six Pillars of AWS well-Architected framework with a suitable example.

=> The AWS well-Architected framework provides guidelines to build secure, high-performing, resilient, & efficient infrastructure in the cloud.

The six Pillars are:

(1) Operational Excellence:
It focuses on Managing & Monitoring the systems to support business & operational goals.

- Use tools like AWS cloudformation to deploy & Update Infrastructure automatically.
- Implement monitoring (e.g. with AWS Cloudwatch) to detect issues early.
- Regularly review & refine procedures to enhance operational efficiency.

Example:
An online retail application deploys its entire stack via cloud formation & uses cloudwatch alarms to trigger automated remediation steps.

(2) Security:
- Enforce strict IAM Policies to manage who can access resources.
- Encrypt data at rest & in transit using services like AWS KMS.

Date : .....
Page: .....

- Continuously monitor & audit security configuration & compliance.
- Ex :- A financial service encrypts sensitive customer data & restricts access with IAM roles, ensuring compliance with security standards.

(3) Reliability:
- Distribute application across multiple AZ's for redundancy.
- Use Auto Scaling & health checks to automatically replace failed instances.
- Implement regular backups & disaster recovery strategies.
- Ex :- A mission-critical service runs across two AZs so that if one zone fails, traffic is automatically routed to healthy instances in another.

(4) Performance Efficiency:
- Choose the appropriate resource types & sizes based on workload needs.
- Use services like Amazon CloudFront to deliver content quickly.
- Continuously track performance & optimize configurations for changing demands.
- Ex :- A media streaming sevice uses AWS lambda

Date: .....
Page: .....

for dynamic processing & cloudfront to cache content globally, ensuring smooth performance during traffic spikes.

(5) Cost Optimization:
- Scale resources according to demand to avoid over provisioning.
- Utilize Reserved Instances or Savings Plans for predictable workloads.
- Regularly review cost reports & adjust usage to control expenses.
- An ecommerce website uses Auto Scaling to match server capacity with user demand, minimizing unnecessary expenditure.

(6) Sustainability:
- Optimize workloads to reduce energy consumpt & waste.
- Utilize AWS services that run in environmentally efficient facilities.
- Design systems that scale only when necessary to avoid idle resources.
- Ex: A data analytics firm schedules non-critica batch jobs during off-peak hours, reducing energy usage & environmental impact.

Date: .....
Page: .....

Q.24) Discuss the Importance of availability tiers with a suitable example.

⇒ Importance of availability tiers:

① High Availability:
- Deploying across multiple tiers ensures that if one tier fails, others can take over without affecting the end user.
- This redundancy minimizes downtime & maintains continuous service availability.

② Fault tolerance:
- Systems designed with availability tiers can automatically route traffic away from failed components.
- It mitigates the risk of sigle points of failure.

③ Load Distribution:
- Spreading workloads across different tiers help s balance traffic & prevents overload on any single component.
- This improves overall performance & responsiveness, especially during peak usage.

④ Maintenance Flexibility:
- Tiers allow for scheduled maintenance or updates on one layer while the others

Date: .....
Page: .....

continue to serve traffic.
- This approach minimizes service disruption & allows for seamless updates & scalability.

(5) Cost-Effective:
While adding extra tiers may incur additional costs, the investment is justified by the significant reduction in potential revenue loss due to downtime.

Example:
- Consider a web app hosted on AWS that deploys its servers in two seperate Availability zones.
- An Elastic Load Balancer distributes incoming traffic evenly between the two AZs.
- If AZ-A experiences a failure, the ELB automatically redirects traffic to AZ-B, ensuring that users continue to access the application without interruption.
- This multi-tier setup guarantees that even if one component fails, the system remains operational & reliable.

Date :
Page :

Step-by-step design:

① VPC setup:
- Create a VPC with public subnets in two AZs.

② EC2 Instances & Auto Scaling:
- Launch an ASG with a template for EC2 Instance running a web server (e.g. Apache/Nginx).
- Configure scaling policies to add/remove instances based on CPU thresholds.

③ Application load Balancer (ALB):
① Create ALB:
- Specify subnets in both AZs for redundanc
- Attach a security group allowing HTTP/HTTPS traffic (ports 80/443).

④ Listeners:
- HTTP (Port 80): Redirect to HTTPS for securit
- " (Port 443): Use an ACM-managed SSL certificate for encryption.

⑤ Target Groups:
- Define a target group for EC2 Instance specifying health checks.
- Set thresholds for making instances as unhealthy.

Date: .....
Page: .....

) Integration:
- Link the ALB to the ASG to auto-register new instances.
- Enable cross-zone load balancing to evenly distribute traffic across AZs.

DNS configuration:
- Use Route 53 to alias the domain "example.com" to the ALB's DNS Name.

Date :
Page :

Q.20) Discuss the importance of combining load balancing with Auto Scaling in cloud computing Environment.

⇒ load balancer:
Distributes Incoming traffic evenly among multiple servers to prevent any one server from becoming overloaded.

Auto Scaling:
Automatically adjusts the number of active servers based on demand, adding servers when traffic increases & reducing them during low demand.

→ Importance of combining:
① High Availability
↳ load balancers reroute traffic if one server fails, ensuring continuous service.
↳ Auto scaling replaces failed servers or scales capacity during high traffic.

② Cost Efficiency:
↳ Auto Scaling ensures you only pay for the resources you need, while load balancing optimizes server usage.

Date: .....
Page: .....

③ Improved User Experience:
↳ Even distribution of traffic prevents overload, reducing delays & downtime.

④ Scalability:
↳ The system can handle sudden increases in traffic (like during a sale or event) without manual intervention.

⑤ Enhanced Performance:
↳ By combining both, applications run faster since traffic is shared & resources are adjusted automatically.

→ Combining Procedure:
- Step 1: Setup an Auto Scaling Group that monitors key metrics (like CPU usage).
- Step 2: Configure a Load Balancer to distribute traffic among the instances.
- Step 3: Integrate the Auto Scaling Group with the Load Balancer so new servers register automatically.
- Step 4: Continuously monitor performance & adjust scaling threshold as needed.

→ Example:
In an ecommerce website during a flash sale, Auto Scaling adds more servers when traffic spikes, and the Load Balancer evenly distributes customer requests among these servers, ensuring a smooth shopping experience.

Date :
Page :

Q19) When you run your application on cloud, you want to ensure that your architecture can scale to handle changes in demand, Discuss how to automatically scale your EC2 instances with Amazon EC2 Auto Scaling.

⇒ Amazon EC2 Auto Scaling automatically adjusts the number of EC2 instances in the application based on real-time demand.

Steps to Automatically Scale EC2 Instances:

↳ Define the settings for EC2 instances that will be used for new instances.
↳ Specify the minimum, maximum & desired number of instances.
↳ The ASG ensures your instances are distributed across multiple AZs for redundancy.
↳ Configure policies to trigger scaling actions based on performance metrics such as CPU usage or network traffic.

↓

Ex: If CPU usage exceeds 70% for a specified period, automatically add more instances (scale out). Conversely, if it drops below a certain threshold, reduce the instance count.

Date: .....
Page: .....

↳ Connect your ASG with an ELB so that new instances automatically start receiving traffic.
↳ This integration distributes user requests evenly, ensuring no single instance becomes a bottleneck.
↳ Simulate traffic loads to ensure scaling policies work as expected.

Benifits:
→ High Availability
→ Cost Efficiency
→ Improved Performance
→ Scalability
→ Improved User Experience

Example:
Consider an online retail website experiencing heavy traffic during a flash sale. Amazon EC2 Auto Scaling detects a spike in CPU usage via cloudwatch automatically launches additional instances, & the load balancer distributes incoming requests among these instances. Once the sale ends & traffic decreases, the system scales back down, saving costs while ensuring the website remains responsive.