**ASSIGNMENT SUBMISSION**

**USE CASE TITLE : SET UP A VIRTUAL CLOUD NETWORK (VCN) WITH SUBNETS, ROUTING, AND LOAD BALANCERS FOR AN ENTERPRISE-LEVEL APPLICATION ON OCI**

**NAME : SANJITH.A.R**

**ROLL NUMBER : 24IT021**

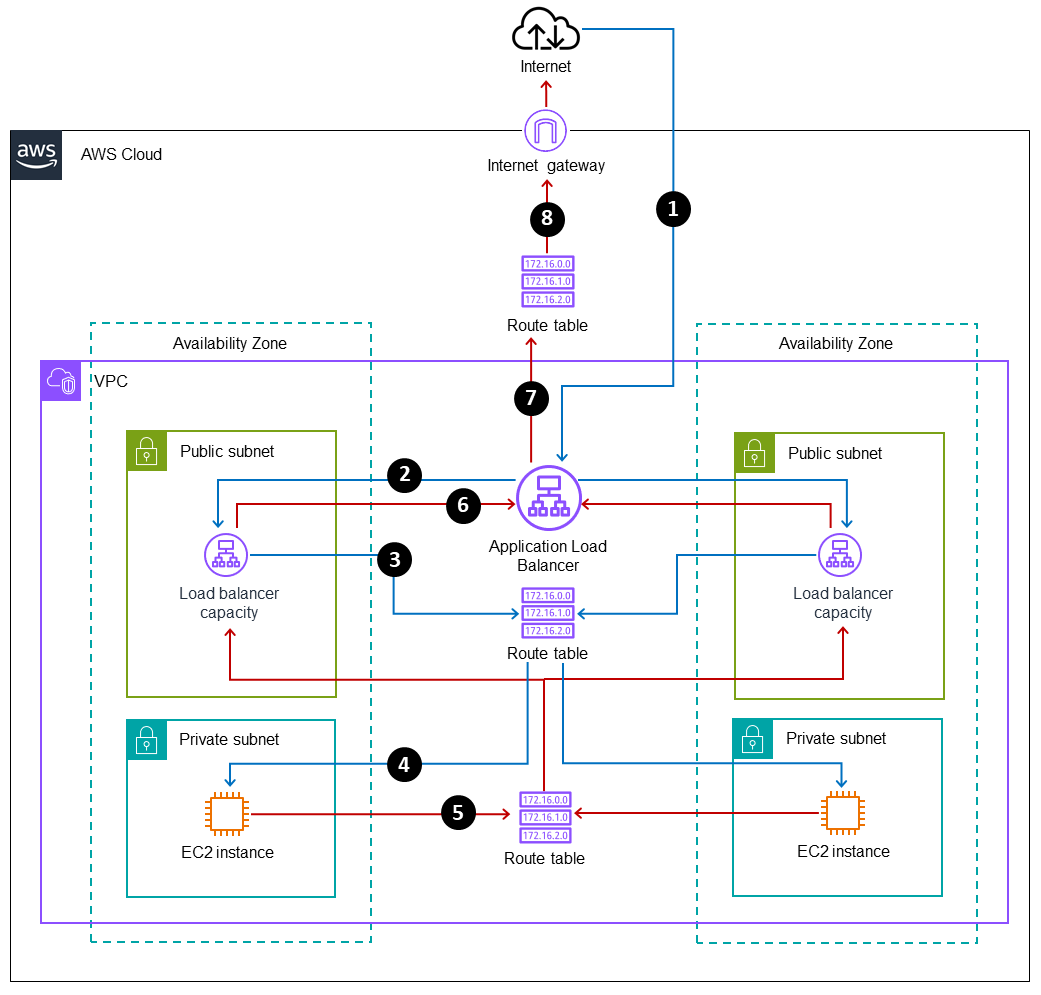
**COLLEGE NAME : MUTHYAMMAL POLYTECHNIC COLLEGE**

**COLLEGE CODE : 327**

**DEPARTMENT : DIT**

**DATE OF SUBMISSION : 27-09-2025**

**1. Architecture Diagram:**

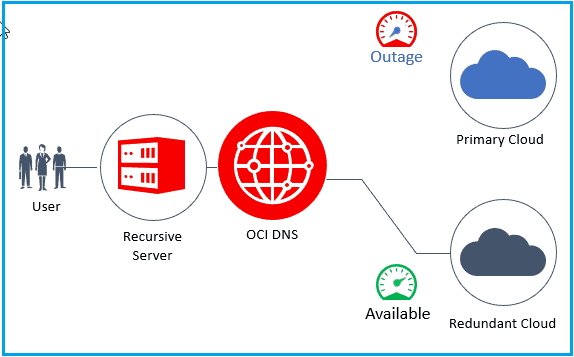
****

1. **VCN (Virtual Cloud Network):**
   * The outermost layer, representing the network boundary.
   * The VCN connects all resources within the network.
2. **Subnets:**
   * Divide the VCN into at least two subnets: one for public resources (e.g., load balancers, bastion hosts) and one for private resources (e.g., app servers, database servers).
   * Public subnet can be associated with an Internet Gateway (IGW) for internet access.
3. **Routing:**
   * Show route tables linked to subnets.
   * Internet Gateway (IGW) is linked to the public subnet route table.
   * NAT Gateway (optional) can be used in the private subnet for outbound internet traffic.
   * Dynamic Routing Gateway (DRG) if connecting to an on-premise network or other cloud networks.
4. **Load Balancers:**
   * Place Load Balancer in the public subnet for distributing traffic to application instances in the private subnet.
   * Label Load Balancer as a critical element to handle traffic distribution.

**Key Components to Label:**

* **VCN (Virtual Cloud Network):** The boundary in which your resources live.
* **Public Subnet:** Contains resources like load balancers and bastion hosts.
* **Private Subnet:** Houses application servers, databases, etc.
* **Internet Gateway** **(IGW):** Provides internet access to resources in the public subnet.
* **Route Tables:** Define how traffic is routed between subnets and the internet.
* **Load Balancer:** Distributes incoming traffic to application servers in the private subnet.
* **NAT Gateway:** Provides outbound internet access for private subnet instances

**2. OCI Services Selection:**



1. **Networking:**

* For effective networking, OCI offers several robust services designed to meet enterprise-level requirements.
  1. **Virtual Cloud Network (VCN):**
* **Justification**: The VCN is a fundamental building block for networking in OCI. It allows you to create a private, isolated network within the OCI environment, similar to a traditional on-premises network.
  1. **Subnets:**
* **Justification**: Subnets allow you to organize your VCN into smaller network segments for better security, isolation, and organization.
  1. **Internet Gateway:**
* **Justification**: This service allows instances within a VCN to access the internet.
  1. **Dynamic Routing Gateway (DRG):**
* **Justification**: A DRG is used for connecting your OCI VCN to your on-premises data center or to another VCN (peering).
  1. **Fast Connect:**
* **Justification**: For secure, dedicated, high-bandwidth connectivity to OCI, Fast Connect allows customers to establish a private connection between their on-premises data center and Oracle Cloud, bypassing the public internet for lower latency and higher security.

1. **Security:**

* For securing your network and infrastructure in OCI, a combination of identity, access, encryption, and monitoring services is necessary.

1. **Identity and Access Management (IAM):**

* **Justification**: IAM enables you to define and control access to resources by setting policies based on roles and compartments.

1. **Web Application Firewall (WAF):**

* **Justification**: WAF helps protect your applications from common web vulnerabilities and attacks such as SQL injection, cross-site scripting, and DDoS attacks.

1. **Network Security Groups (NSG):**

* **Justification**: NSGs are used to define firewall rules for controlling traffic to and from instances within a VCN.

1. **Cloud Guard:**

* **Justification**: Cloud Guard is an AI-driven security service that provides threat detection and remediation across your OCI environment.

1. **Key Management (KMS):**

* **Justification**: OCI's KMS enables the creation and management of encryption keys for securing data at rest.

1. **OCI Bastion:**

* **Justification**: Bastion provides secure access to private instances within your VCN, allowing administrators to log in to these resources without exposing them directly to the internet.

1. **Traffic Management:**

* Traffic management ensures that your traffic is routed efficiently, securely, and in accordance with business requirements.

1. **Load Balancer:**

* **Justification**: OCI’s Load Balancer service distributes incoming traffic across multiple instances to ensure high availability and scalability of applications.

1. **Traffic Management (DNS and Global Traffic Management):**

* **Justification**: OCI’s Traffic Management service, combined with DNS, allows for global traffic distribution and performance optimization.

1. **OCI Cloud Firewall:**

* **Justification**: The Cloud Firewall service allows you to create rules that govern the traffic flow between networks (e.g., VCNs, on-premises).

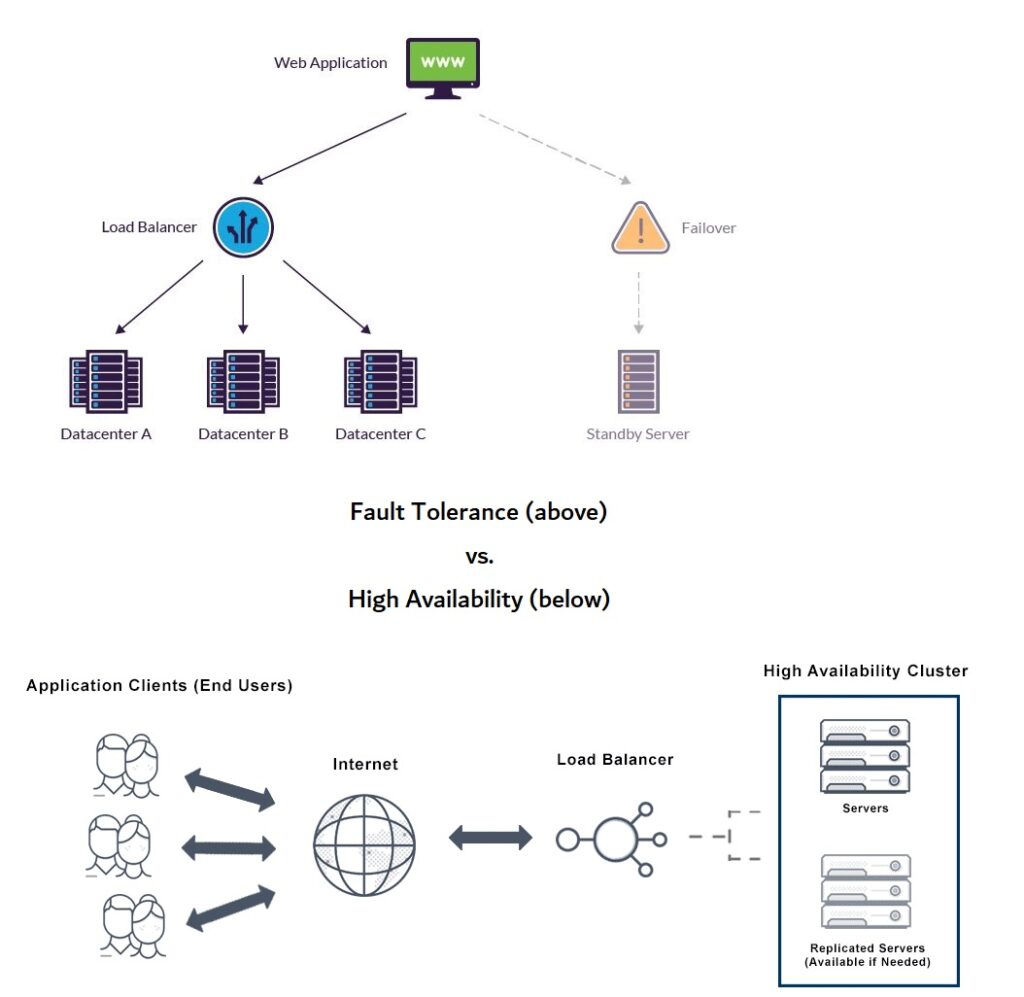
1. **Autonomous Database (with Application Traffic Routing):**

* **Justification**: If your application relies on databases that require high availability and performance, Autonomous Database can integrate with traffic management to ensure that user requests are routed to the appropriate database instance.

1. **OCI VPN (Site-to-Site or Remote Access VPN):**

* **Justification**: OCI VPN is ideal for securely connecting remote offices or users to the Oracle Cloud.

**3. Fault-Tolerant Networking Solution:**



**1. High Availability (HA) Architecture:**

* High availability ensures that the system remains operational and accessible, even if some components fail.

1. **Multi-Availability Domain (AD) Deployment:**

* **Design**: Deploy resources across multiple Availability Domains (ADs) within a region.

1. **Load Balancers with Auto-Scaling:**

* **Design**: Use **OCI Load Balancer** to distribute incoming traffic evenly across multiple instances, located in different ADs or availability zones.

1. **Redundant Networking Paths:**

* **Design**: Set up **multiple internet gateways** or **Fast Connect** links to ensure that if one internet connection goes down, another one can take over.

1. **Cross-Region Redundancy:**

* **Design**: For global applications, consider deploying resources in multiple OCI regions and configure cross-region **VPC peering** or **Fast Connect** links.

**2. Redundancy:**

* Redundancy ensures that multiple backup resources are in place to maintain continuous service during failures.

1. **Redundant Compute Instances:**

* **Design**: Use multiple compute instances across different ADs, spread across availability zones, to ensure redundancy in case an instance or availability zone becomes unavailable.

1. **Redundant Data Storage:**

* **Design**: Implement **Oracle Cloud Block Volumes** for persistent data storage, with **data replication** across multiple ADs.

1. **Redundant Networking and VPN Connectivity:**

* **Design**: Set up **redundant VPN connections** or **site-to-site VPNs** for secure and reliable hybrid cloud or remote office connectivity.

1. **Database Redundancy:**

* **Design**: Use **Oracle Autonomous Database** with **Data Guard** for database redundancy. Set up **Active Data Guard** in a different AD or region for disaster recovery.

**3. Seamless Transactions:**

* Seamless transactions ensure that your applications and users can continue interacting without interruptions, even if there are failures in part of the infrastructure.
  + 1. **Session Persistence and Sticky Sessions:**
* **Design**: Use **OCI Load Balancer** with **session persistence** (also known as sticky sessions) for applications that require the user to maintain an ongoing session with the same instance.
  + 1. **Graceful Failover:**
* **Design**: Ensure that your application is designed with graceful failover in mind.
  + 1. **Distributed Transactions with Transactional Integrity:**
* **Design**: For distributed applications, use **Oracle RAC (Real Application Clusters)** or **Micro services with Event-Driven Architecture** for highly available transactional systems.
  + 1. **Monitoring and Automated Remediation:**
* **Design**: Implement **OCI Monitoring** and **Event Services** to automatically detect service degradation or failures, and trigger remediation actions (such as starting a backup instance or re-routing traffic).
  + 1. **Multi-Region Load Balancing:**
* **Design**: Use **OCI Global Traffic Management** to direct user traffic to the closest region based on their geographic location or to failover to a secondary region during regional outages.

**4. Disaster Recovery (DR):**

* Disaster recovery is a critical part of fault tolerance, as it ensures the system can recover from catastrophic failures.

1. **Cross-Region Disaster Recovery:**

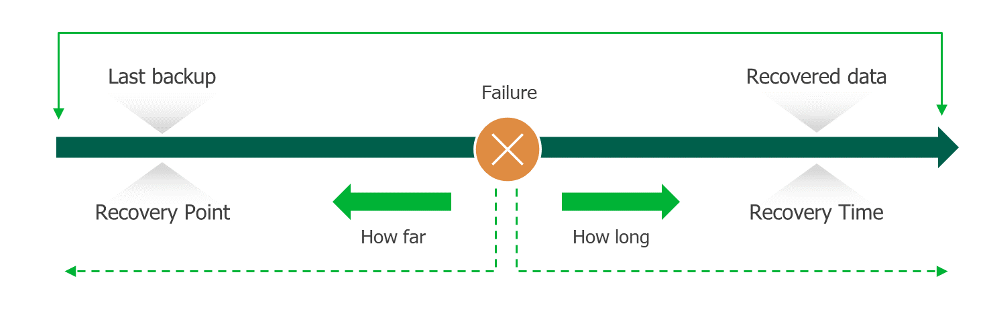
* **Design**: Set up cross-region replication for critical components like compute instances, storage, and databases.

1. **Backup Automation:**

* **Design**: Use **OCI Backup Service** to automate regular backups of your databases, compute instances, and storage volumes.

.

**4. Disaster Recovery Plan:**



1. **RTO (Recovery Time Objective):**

* Maximum acceptable downtime.
* **Target:**
* Critical systems: 1 hour.
* Others: up to 24 hours.

1. **RPO (Recovery Point Objective):**

* Maximum acceptable data loss.
* **Target:**
* Critical systems: 15 minutes.
* Others: up to 24 hours.

1. **Failover Strategy:**

* **Critical systems:** Automatic failover to standby systems.
* **Others:** Manual or semi-automated failover.
* Failback procedures restore original systems after resolution.

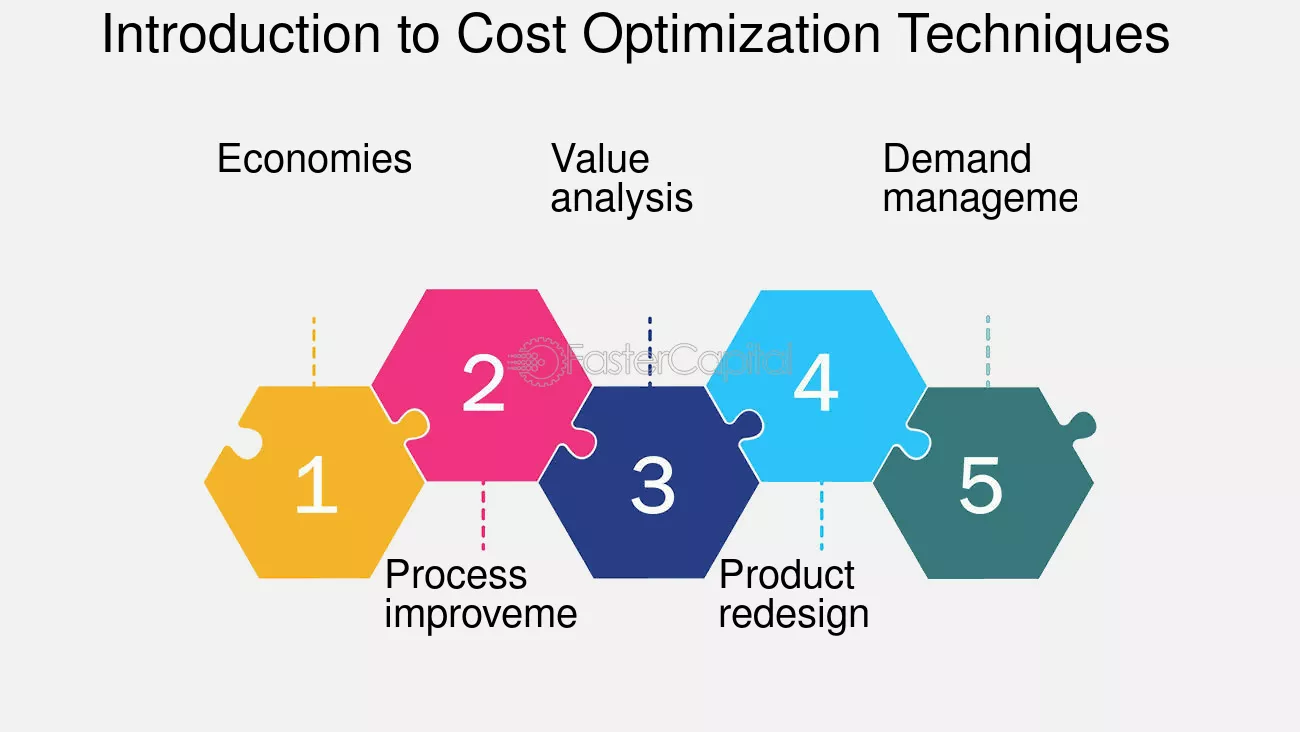
1. **Replication:**

* Real-time replication for critical systems.
* Scheduled backups for non-critical systems.
* Data stored in multiple geographic locations.

1. **Testing:**

* DR tests conducted quarterly to ensure readiness.

### 5. Cost Optimization Strategies:



**1. Right-Sizing Resources:**

* Regularly monitor usage and adjust compute, storage, and database resources to match actual demand.

**2. Auto-Scaling:**

* Implement auto-scaling for workloads to handle traffic spikes while minimizing idle capacity.

**3. Reserved & Spot Instances:**

* Use reserved instances for predictable workloads and spot/preemptible instances for flexible, non-critical tasks.

**4. Storage Tiering:**

* Move infrequently accessed data to lower-cost storage tiers (e.g., archival storage).

**5. Cost Monitoring & Alerts:**

* Use tools like AWS Cost Explorer, Azure Cost Management, or GCP Billing to track usage and set alerts.

**6. Security Without Overspend:**

* Use built-in security features (e.g., firewalls, identity management) before investing in third-party tools.
* Apply the principle of least privilege to reduce overhead and risk.

**7. License Optimization:**

* Reevaluate software and service licenses regularly to avoid over-provisioning or underutilization.

**8. Automation:**

* Automate startup/shutdown of non-production resources during off-hours.

**9. Right-Sizing and Auto-Scaling:**

* Continuously monitor system usage to ensure resources match actual demand. Use auto-scaling to handle peak loads efficiently while reducing costs during low usage periods.

**10. Tiered Storage and Resource Scheduling:**

* Move infrequently accessed data to low-cost storage tiers. Schedule non-critical environments (e.g., dev/test) to shut down during off-hours to avoid unnecessary charges.