**Oracle Cloud Infrastructure (OCI) Overview:**

- Many large enterprises trust and run their critical workloads on OCI.

- OCI provides a next-generation cloud platform with a global footprint of regions.

- It offers multicloud support with Microsoft Azure and a hybrid offering called Dedicated Region Cloud@Customer.

**OCI can be categorized into seven major categories:**

**1. Core infrastructure:** Provides compute, storage, and networking services for running applications and managing data.

**2. Database services:** Offers various flavors of Oracle and open-source databases, including autonomous databases and support for virtual machines, bare metal servers, and Exadata.

**3. Data and AI services:** Includes managed Apache Spark, data catalog, data integration, data science platform, and Apache Kafka for data processing, tracking, ingestion, ETL, and machine learning.

**4. Governance and administration:** Provides security, identity, observability, and management services, with features like compartments, network protection, encryption, logging, and application performance management.

**5. Developer services:** Offers managed services like APEX, Terraform, and resource manager for developers.

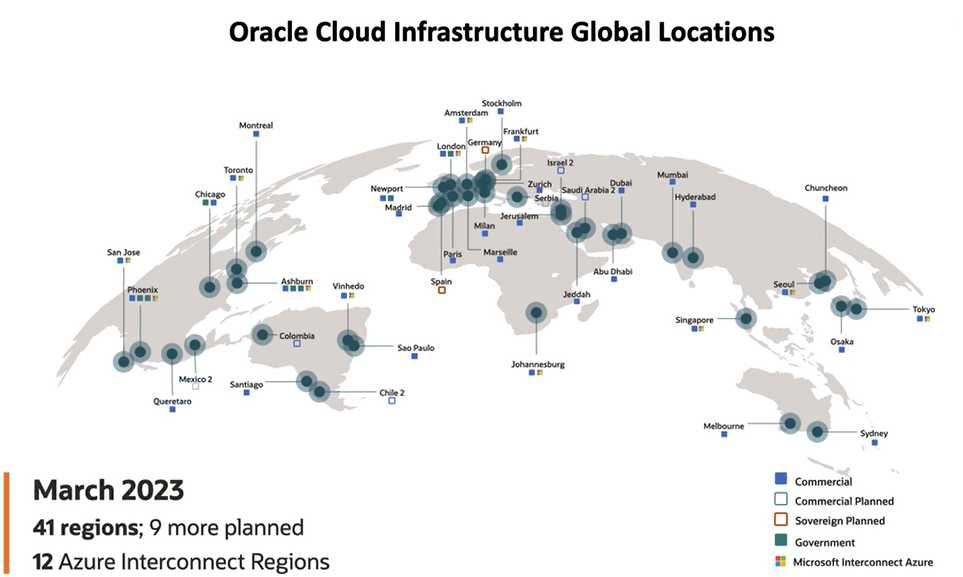
**6. Analytics:** Includes Oracle Analytics Cloud, a managed analytics service integrating with third-party solutions.

**7. Application services:** Provides managed serverless functions, API Gateway, event service, and a suite of connected SaaS applications for various business functions.

- OCI has over 80 services available, and users can try them using the free tier account.

- OCI supports users in their journey to the cloud, whether starting fresh or migrating their existing data centers.

- OCI is a comprehensive and extensive cloud platform designed to meet diverse enterprise needs.

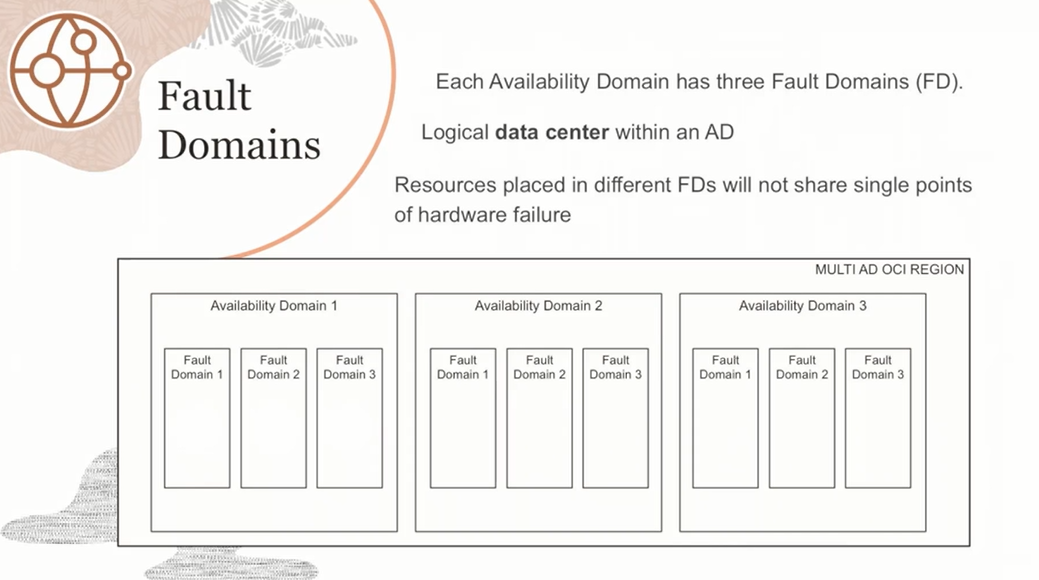


**OCI Architecture - Core Constructs and Choosing a Region**

**- \*\*Regions\*\*:** Regions are localized geographic areas that consist of one or more availability domains. These regions are essential components of OCI's physical architecture.

**- \*\*Availability Domains\*\*:** Availability domains are fault-tolerant data centers located within a region. They are interconnected by a high-speed, low-latency network. Think of them as logical data centers that provide reliability and redundancy.

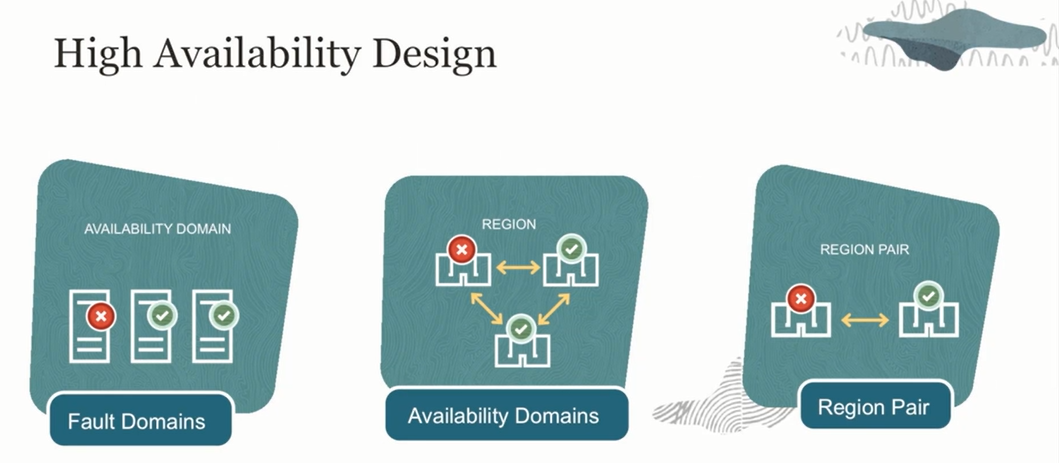
**- \*\*Fault Domains\*\*:** Fault domains are subsets of hardware and infrastructure within an availability domain. They are designed to ensure anti-affinity, meaning that critical components are distributed across different fault domains to enhance resilience.



- **\*\*OCI's Geographic Footprint\*\*:** OCI has a global presence with numerous regions worldwide. This extensive coverage allows users to select the region closest to their target audience, ensuring minimal latency and optimal performance.

- **\*\*Multicloud Partnership\*\*:** OCI has established a strategic partnership with Microsoft Azure, enabling seamless integration and interoperability between the two platforms.

**- \*\*Dedicated Region Cloud@Customer\*\*:** OCI offers a unique hybrid cloud solution called Dedicated Region Cloud@Customer. It combines the benefits of OCI's cloud services with on-premises infrastructure, providing a secure and flexible hybrid cloud environment.



* Availability domains are isolated from each other and designed to be fault-tolerant, minimizing the impact of failures.
* Each availability domain contains multiple fault domains, acting as separate logical data centers to avoid single points of hardware failure.
* Fault domains allow resources to be distributed across different domains, ensuring high availability and reducing the risk of outages.
* Replication of applications and databases across fault domains and availability domains increases resilience and ensures continuous availability.
* Technologies like Oracle Data Guard can be used for data synchronization between copies of databases in different domains.
* Even in regions with a single availability domain, the fault domain construct can be utilized to achieve higher availability and prevent single points of failure.
* OCI offers region pairs with two data centers in most countries, enabling disaster recovery, backup, and compliance with data residency requirements.
* OCI provides SLAs (Service Level Agreements) for availability, management, and performance.



In the example provided in the lesson, let's assume we have a region with three availability domains (AD1, AD2, and AD3), and each availability domain consists of three fault domains (FD1, FD2, and FD3).

To design an application architecture with high availability and avoidance of single points of failure, the following steps are suggested:

1. Create a software-defined virtual network for your application.

2. Deploy multiple copies of your application across all fault domains within an availability domain. This means replicating the application tier across FD1, FD2, and FD3.

3. Similarly, replicate the database tier across the fault domains to ensure redundancy and fault tolerance.

4. To further enhance availability, deploy another set of the application and database in a separate availability domain. This creates a dual availability domain setup with two copies of the application and two copies of the database.

5. To keep the data synchronized between the copies, you can use technologies like Oracle Data Guard, which maintains primary and standby data instances in sync.

6. By leveraging fault domains and availability domains, you achieve higher availability and resilience. If one fault domain or availability domain experiences an issue or failure, the application and database instances in the other fault domains and availability domains continue to run smoothly, ensuring continuous availability of your application.

The example demonstrates the concept of distributing resources across fault domains and availability domains to avoid single points of failure. By implementing such a design, you can maximize the uptime and reliability of your application, even in the event of failures or disruptions in specific domains.

**Several hybrid and multicloud offerings provided by Oracle Cloud Infrastructure (OCI) are discussed. Here's an explanation of the examples mentioned:**

**1. Dedicated Region Cloud@Customer**: This offering enables customers to have an air-gapped private isolated region within their own data center. It provides the full range of OCI's cloud services, allowing customers to meet data residency, compliance, and latency-sensitive application requirements. The Dedicated Region is installed, operated, supported, and upgraded by Oracle, similar to public cloud regions.

**2. Oracle Cloud VMware Solution**: This service allows customers to migrate their on-premises VMware-based virtual machines (VMs) seamlessly to the OCI cloud. It can be used for disaster recovery or complete data center migration, providing a VMware-based environment in OCI.

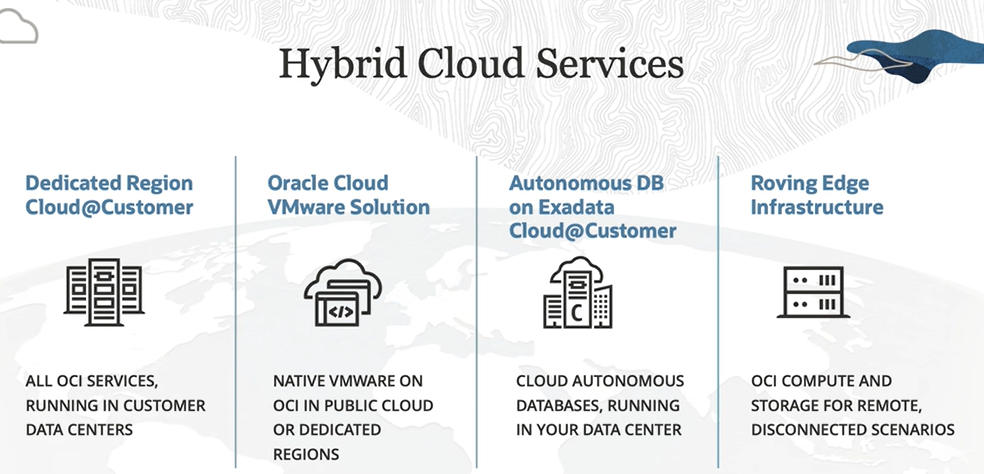
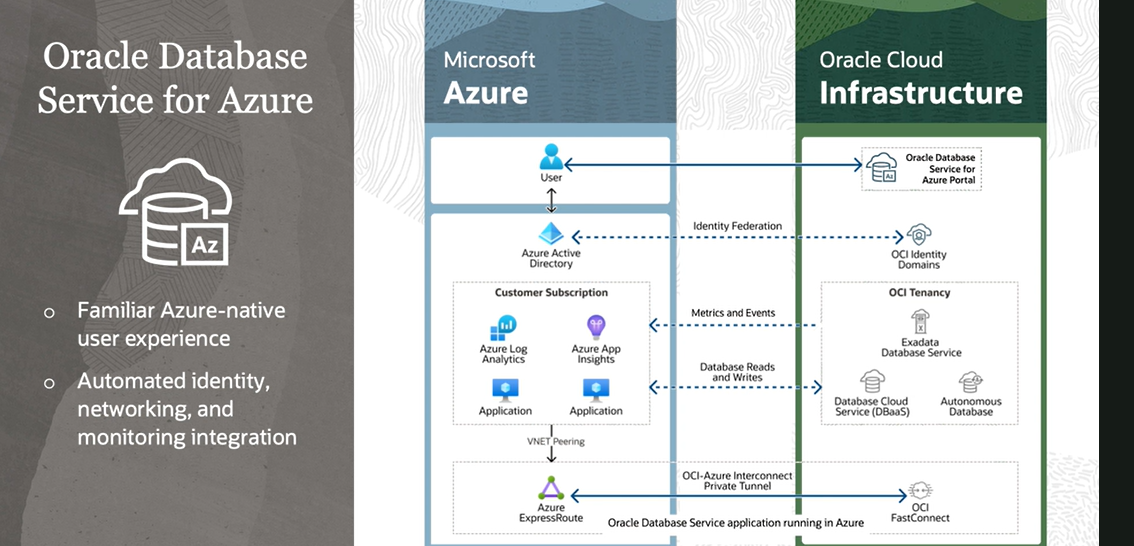
**3. Autonomous DB on Exadata Cloud@Customer**: This offering brings Oracle's Autonomous Database capabilities to customers' own on-premises environments. It enables organizations to leverage the advanced features of Autonomous Database while maintaining data within their data centers for governance, compliance, and security reasons.

**4. Roving Edge Infrastructure**: This offering provides a portable, ruggedized server optimized for edge computing in remote and disconnected environments. It allows processing and computing at the edge, where connectivity may be limited or intermittent.

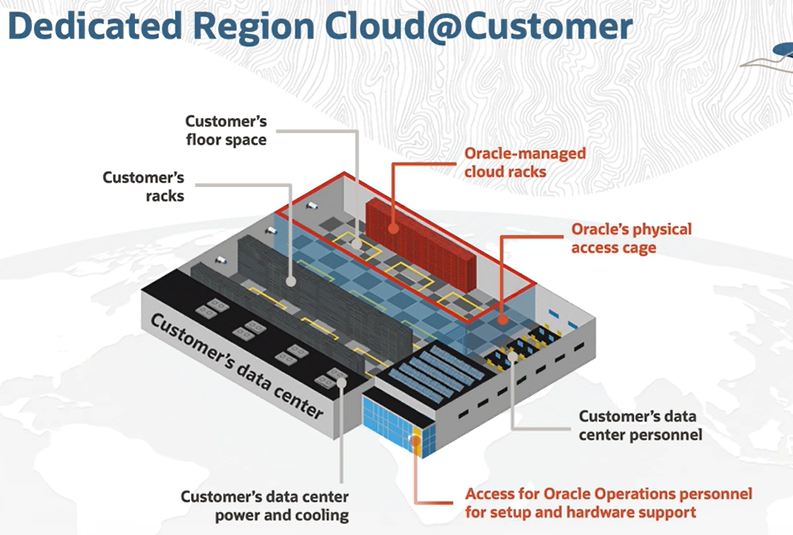
**5. OCI Azure Interconnect**: Oracle and Microsoft have established a private interconnect between OCI and Azure. This low-latency interconnect enables running applications on Azure while leveraging Oracle databases on OCI seamlessly. It offers private connectivity with latency of less than 2 milliseconds and is available in 12 regions worldwide.

**6. Oracle Database Service for Azure**: This service allows Azure customers to provision, access, and operate Oracle Database services within OCI. It provides a simplified experience for Azure users to connect their Azure subscription to the OCI tenancy. The service automates the configuration of the interconnect, streamlines the deployment process, and provides unified telemetry and monitoring across both environments.

These hybrid and multicloud offerings from OCI provide customers with flexibility, choice, and the ability to leverage OCI services in various deployment models based on their specific needs and requirements.

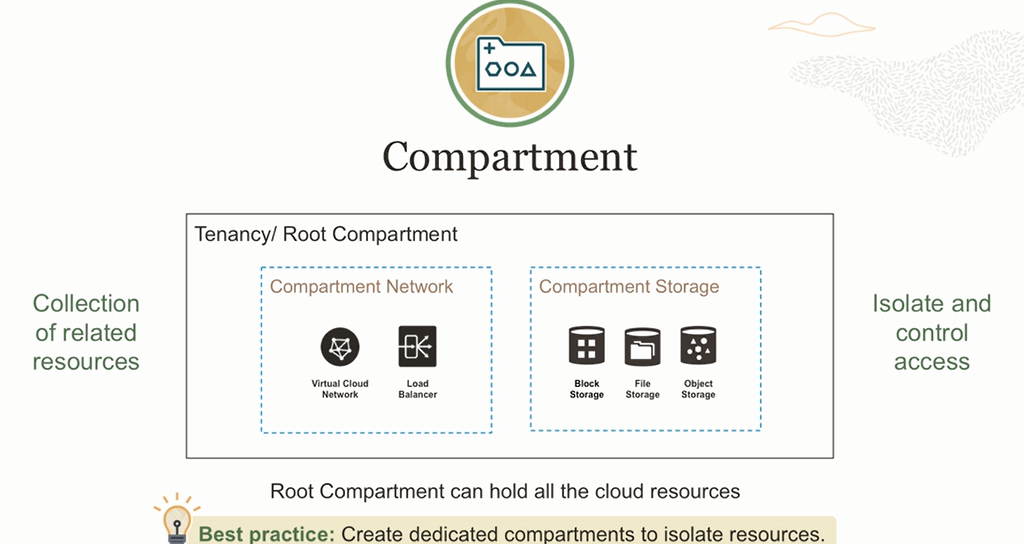
  


* The Oracle Database Service for Azure is a special service provided by Oracle. It allows customers who use Microsoft Azure (a different cloud provider) to easily use Oracle's database services within their Azure environment.
* Imagine you have been using Azure for your cloud needs, but you also want to use Oracle's powerful database services. With this service, you can connect your Azure account to Oracle's cloud infrastructure with just a few clicks. The service takes care of setting up the connection for you, so you don't have to worry about the technical details.
* Once the connection is established, you can start using Oracle's database services as if they were part of Azure. It provides you with a unified experience, so you can manage your Azure and Oracle resources together. You can monitor and track metrics, logs, and events from your Oracle databases within Azure's monitoring tools. This makes it easier for you to keep an eye on your databases and ensures you have all the information you need in one place.
* The best part is that there is no extra cost to use this service. It's included in your existing Azure subscription, so you don't have to worry about additional charges.
* In summary, the Oracle Database Service for Azure allows Azure users to seamlessly access and operate Oracle's database services within their Azure environment without any extra cost. It simplifies the process and provides a unified experience for managing both Azure and Oracle resources.

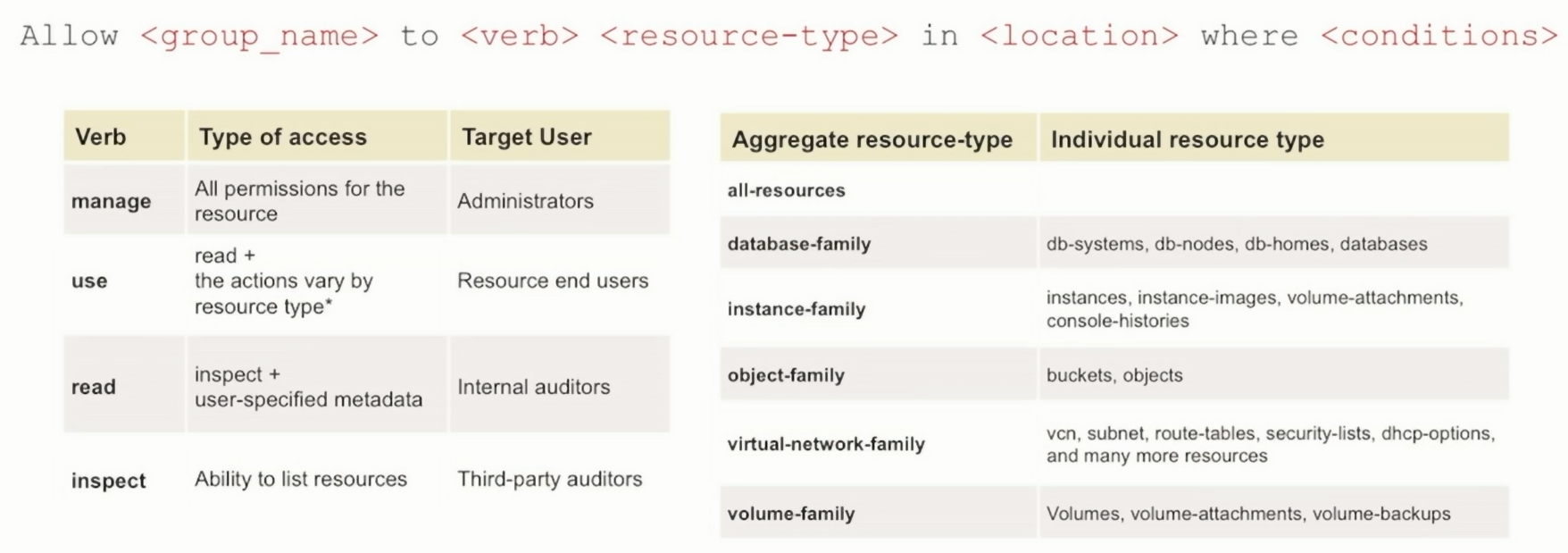


**OCI Identity and Access Management (IAM)**

* IAM, which stands for Identity and Access Management, is a service that focuses on two main aspects: authentication (AuthN) and authorization (AuthZ).
* Authentication is about verifying someone's identity. It ensures that a person is who they claim to be. On the other hand, authorization deals with permissions and what someone is allowed to do. It assigns roles to users, and each role has a set of permissions associated with it.
* One important feature of IAM is identity domains. An identity domain is like a container that holds users and groups. It represents a user population in OCI (Oracle Cloud Infrastructure) along with associated configurations and security settings. Users and groups are created within an identity domain, and policies are written to manage access control.
* Policies are scoped to a tenancy, an account, or a compartment. Compartments provide logical isolation for resources. Resources in the cloud, such as compute instances, block storage, databases, and others, are referred to as resources. Each resource is assigned a unique identifier called an Oracle Cloud ID (OCID), automatically generated by OCI. The OCID consists of different components, including the resource type, realm, region, and a unique ID.
* When working with the management console, you won't directly interact with OCIDs. However, if you use the CLI (Command Line Interface) or SDK (Software Development Kit), you will need to use OCIDs to reference resources. Oracle generates these unique identifiers, so you don't have to worry about creating them yourself.
* In summary, OCI IAM provides authentication and authorization services. It ensures the identity of users and controls their permissions through roles and policies. Identity domains serve as containers for users and groups, while OCIDs are unique identifiers for cloud resources.

**Compartments:**   






**Authentication in OCI:**

* Username/password
* API signing Keys
* Authentication Tokens.

**Authorization in OCI:** Policies

**VCN:**  
- A Virtual Cloud Network (VCN) is like a private network that you create in Oracle Cloud for secure communication.

- It exists in a specific region and offers high availability, scalability, and security without you having to worry about it.

- A VCN has an address space, which is like a range of IP addresses that can be divided into smaller networks called subnets.

- Your compute instances, like web servers or databases, are placed in these subnets and get assigned IP addresses.

- There are different mechanisms in a VCN for communication. An internet gateway allows instances to communicate with the internet and access websites.

- A NAT gateway provides outbound communication to the internet but blocks incoming connections from the internet, ensuring security.

- A Service Gateway allows resources in the VCN to securely access Oracle Cloud services like object storage without going through the internet.

- A Dynamic Routing Gateway enables private communication between your VCN and destinations outside the internet, such as your on-premises environment.

Routing:  
- In OCI routing, we have a concept called route tables that determine how traffic flows in a Virtual Cloud Network (VCN).

- Route tables contain route rules that specify where the traffic should go based on its destination.

- Traffic within the VCN subnet is automatically handled by the VCN itself, so no additional routing is needed.

- For traffic going outside the VCN, we can use a NAT gateway to access the internet and a Dynamic Routing Gateway for communication with on-premises environments.

- The route table entries determine the paths for the traffic. The more specific route takes priority.

- If you have multiple VCNs, they can communicate with each other through local peering if they are in the same region or remote peering if they are in different regions.

- To handle communication in complex environments with many VCNs, a newer version of the Dynamic Routing Gateway (DRG v2) allows for simplified and scalable connectivity, supporting up to 300 VCNs on a single DRG. Additional DRGs can be connected via remote peering if needed.

**Security list and Network security groups:**

* Within a Virtual Cloud Network (VCN), there is a concept called a security list, which acts like a firewall for subnets and controls the traffic allowed in and out.
* Security lists consist of rules that define the type of traffic allowed or blocked. These rules can be stateful (allowing traffic in both directions) or stateless (allowing traffic in only one direction).
* Each rule specifies the source, destination, protocol, and port for the allowed traffic. For example, allowing web traffic (TCP on port 80) from anywhere on the internet.
* Subnets within the VCN can have their own security lists, with different rules based on the specific requirements.
* Network Security Groups (NSGs) are similar to security lists but apply to specific virtual network interface cards (VNICs) within a single VCN. NSGs can be used as the source or destination in the rules, providing more flexibility.
* With NSGs, instances within the same subnet can have different security configurations, allowing for more granular control over traffic.
* By using security lists and network security groups, you can define and manage firewall rules to secure the traffic within your VCN.

**Example:**

Imagine you have a virtual network in the cloud called MyCloudNetwork. Within this network, you have two subnets: PublicSubnet and PrivateSubnet. You also have two instances running in these subnets: WebServer and DatabaseServer.

Now, to protect your instances and control the incoming and outgoing traffic, you can use security lists. Think of security lists as sets of rules that act like firewalls for your subnets. These rules define what types of traffic are allowed or blocked.

For example, let's say you want to allow web traffic to reach your WebServer from anywhere on the internet. In the security list associated with the PublicSubnet, you would create a rule that says, "Allow TCP traffic on port 80 (web traffic) from any source IP (anywhere on the web)." This rule ensures that incoming web traffic can reach your WebServer.

Similarly, you want to protect your DatabaseServer in the PrivateSubnet and only allow database-related traffic from the WebServer. In the security list associated with the PrivateSubnet, you would create a rule that says, "Allow TCP traffic on port 1521 (database port) from the IP address of the WebServer." This rule ensures that the DatabaseServer only accepts database connections from the trusted WebServer.

Now, let's talk about network security groups (NSGs). NSGs are similar to security lists but provide more fine-grained control. Imagine you want to have different security configurations for your instances within the same subnet.

For instance, you might have two instances in the PublicSubnet: WebServer1 and WebServer2. WebServer1 needs to allow SSH (secure shell) access for remote administration, but WebServer2 doesn't require that access. With network security groups, you can assign different rules to these instances based on their specific needs.

You can create an NSG called WebServerNSG and associate it with WebServer1. In this NSG, you would create a rule that allows SSH traffic from your trusted IP address for remote administration. Meanwhile, WebServer2 can have a different NSG called WebServer2NSG with different rules tailored to its requirements.

So, using security lists and network security groups, you can define and manage the firewall rules to protect your instances within the virtual network. These rules ensure that the right traffic is allowed, while unauthorized or potentially harmful traffic is blocked, providing a secure environment for your cloud resources.

**Tenancy Setup**

1. Don't use the tenancy administrator account for day-to-day operations: It's not a good practice to use the main administrative account for regular tasks. Instead, designate specific administrators for your account who will handle the daily operations.

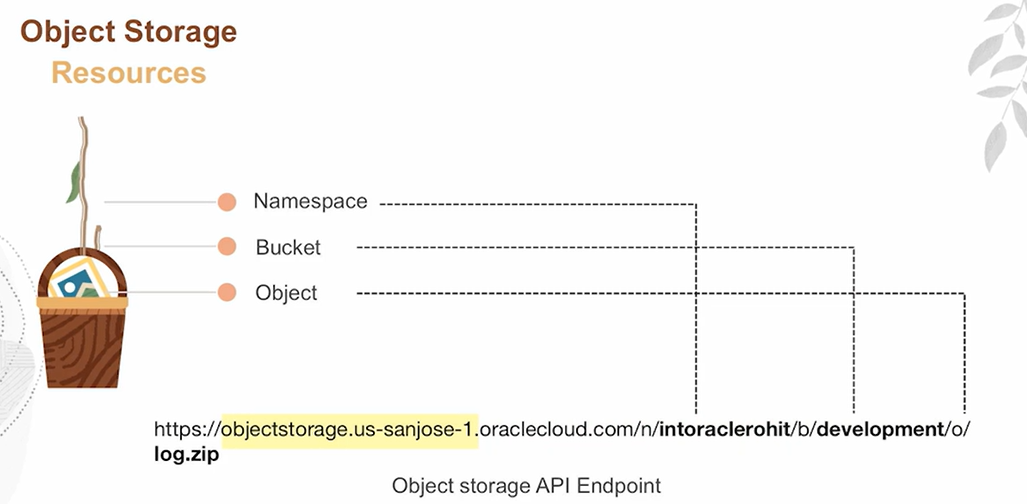
2. Create dedicated compartments to isolate resources: Compartments are like folders that help you organize and isolate your resources. It's recommended to create separate compartments for different purposes, such as sandbox (for testing), production, development, or based on regions or business units. Avoid putting everything directly under the root compartment.

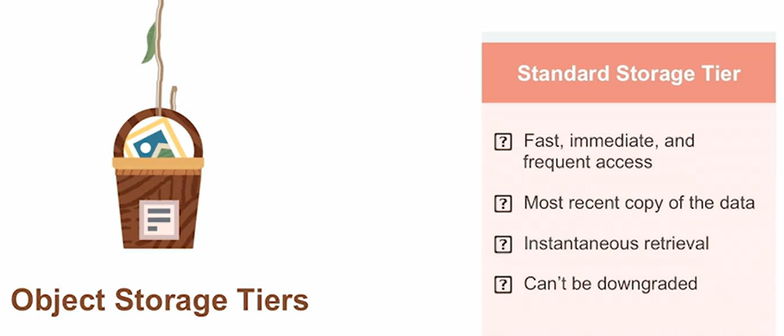
3. Enforce multi-factor authentication (MFA): Multi-factor authentication is a security measure that requires users to provide multiple pieces of evidence to verify their identity. It typically involves a combination of something you know (like a password) and something you have (like a device or security token). Enforcing MFA adds an extra layer of protection to your account.

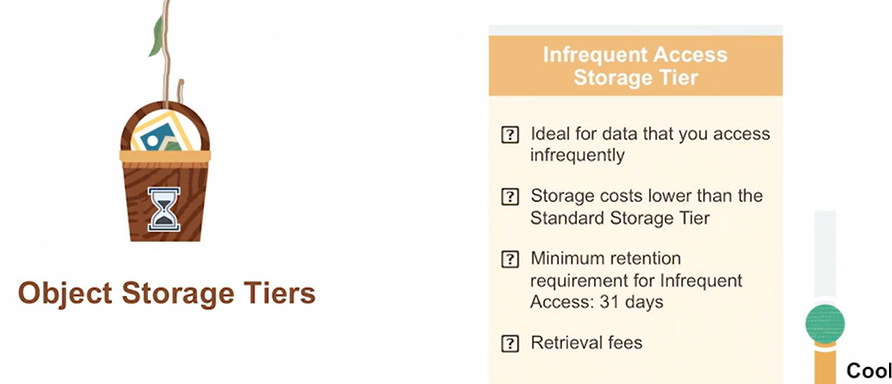
In terms of policies for OCI administrators:

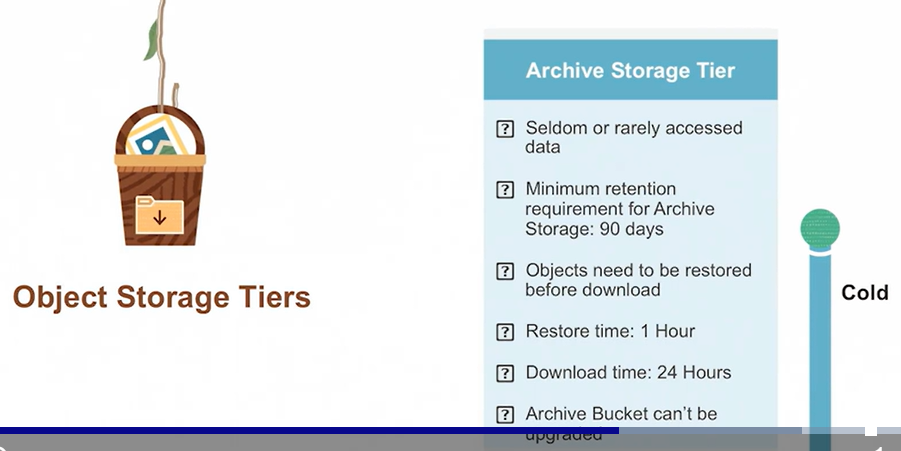
- Grant access to manage all resources: Give the OCI administrators the necessary permissions and privileges to manage all resources within the tenancy or specific compartments. This allows them to perform tasks similar to the tenancy administrator.

- Use specific resource types for Identity and Access Management (IM): IM is the service that handles user accounts, groups, policies, etc. You need to write policies using individual resource types, such as domains, users, groups, policies, compartments, etc. These policies enable OCI administrators to create and manage users, groups, and policies effectively.

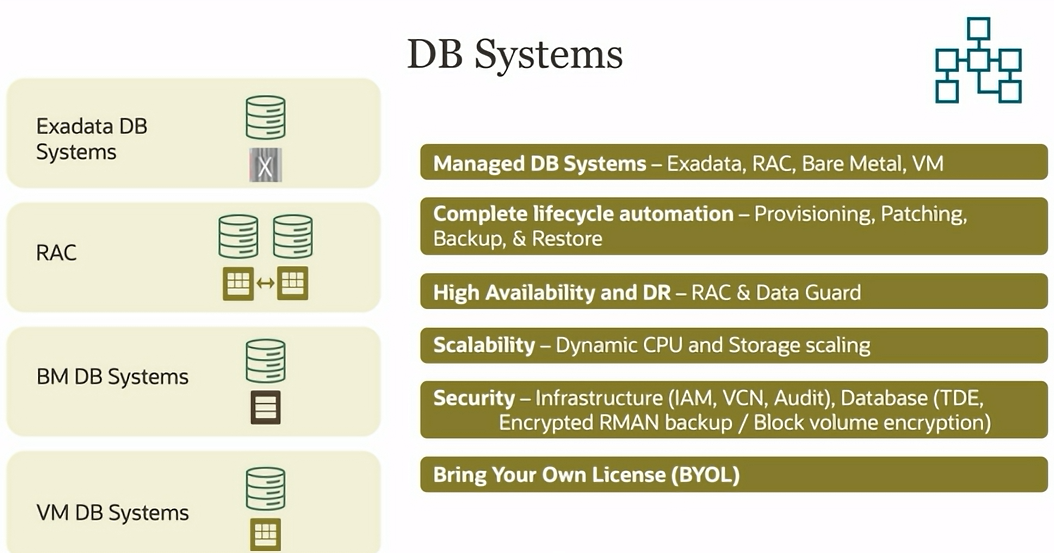
**Object storage:  
**

****

****

****

****

****

**Mysql**MySQL Cloud Service, also known as MySQL Database Cloud Service, is a service that allows you to host your MySQL databases in the cloud. It offers several features like easy provisioning, patching, and scalability, which means you can adjust your computing resources and storage as needed. It also provides security features and reduces the overall cost of managing your own MySQL database.

Two important features of this service are High Availability and HeatWave. High Availability ensures that your database remains accessible even in the event of failures. It provides automatic failover, increased uptime, and zero data loss. For production environments, it's recommended to use the High Availability option, which sets up multiple instances of the database across different domains, replicating the data between them. If one instance fails, the service automatically switches to a secondary instance without any data loss.

HeatWave is another feature that enhances the performance of MySQL Database Service. It acts as an in-memory query accelerator specifically designed for analytics and transaction queries. By utilizing HeatWave, you can significantly improve the speed of your MySQL database for these types of queries. It can scale out to support large workloads and is optimized for Oracle Cloud Infrastructure.

In simpler terms, MySQL Cloud Service allows you to store your databases in the cloud, and it offers two important features: High Availability ensures your database remains accessible even during failures, and HeatWave boosts the performance of your database for analytical and transactional queries.

**NO SQL**

Oracle NoSQL Database Cloud Service is a fully managed and elastic database service provided by Oracle. It offers high performance and predictable low latencies, which means your applications can access and process data quickly.

One of the key features of this service is its flexibility in handling different types of data models. It supports document, fixed schema, and key-value models, allowing you to store and retrieve data in different formats within the same database. This makes it easier for developers to work with diverse data structures.

Security is also a priority, as the service provides enterprise-grade security measures. You can manage access to your data and ensure it remains protected.

Since it runs in the cloud, the operating costs are lower compared to running your own database infrastructure. It's developer-friendly, with REST APIs that simplify integration with applications. The service is always available, utilizing the high availability features of Oracle Cloud Infrastructure.

Another notable aspect is that the NoSQL Database Cloud Service can be used both in the cloud and on-premises. This means you have the flexibility to choose where you want to store your data and seamlessly transition between different environments.

The top use cases for this service include modern applications such as mobile apps, IoT (Internet of Things) systems, and online advertising platforms. These applications often deal with large volumes of data, making the scalability and performance of a NoSQL database highly beneficial.

In simple terms, Oracle NoSQL Database Cloud Service is a powerful and flexible database solution that provides fast access to data, supports various data models, ensures data security, and can be used in the cloud or on-premises. It is well-suited for applications that handle large volumes of data, such as mobile apps, IoT systems, and online advertising platforms.

