OCI Architect Associate Master Cheat Sheet

Getting Started with OCI

==========

1. Global Footprint - 16 active (11 commercial, 5 Govt), 20 new regions (17 commercial, 3 Govt)
2. Interconnect with Azure - Ashburn and London, other regions planned
3. OCI Region - Multiple fault de-correlated, completely independent data centers: AD; Grouping of hardware and infra with in AD: FD
4. One AD Regions - Next 12 months, Region or AD will be added
5. Off-box Network virtualization - All virtualization put into custom silicon cards, includes all storage and network I/O
6. OCI Services - Identity, Networking, Compute, Storage, Database, Autonomous DB, Serverless, Analytics, Next Layer Services, Security, Data movement, Edge
7. Differentiation - Off-box Network virtualization, Bare metal + Local NVMe storage, All SSD storage, No Network, memory or CPU over-subscription; Battle tested; DB options(BM,VM,Exadata,RAC); Enterprise App support (EBS, JDE)
    a. Aggressive and Predictable pricing (Cheaper than AWS); SLAs on Performance, Management and Availability; BYOL and Universal Cloud Credits; Support thru one org

OCI Identity and Access Management

==========

1. IAM - enables to control what type of access a group of users have and to which specific resources
2. Each OCI resource has unique OCID
3. IAM uses traditional identity concepts - Principals, Users, Group, AuthN, AuthZ; New capability - Compartments
4. Principals - IAM entity interact with OCI resources; IAM users and Instance Principals; User has no permissions until placed in groups; Group having at least one policy with permission to tenancy or compartment
5. Group - collection of users; same user can be a member of multiple groups; Instance Principals - let instances to make API calls against other OCI services
6. Authentication - Username and Password; API siging key; Auth Tokens (Don't expire)
7. Authorization - define specific privileges in policies and associating them with principals; policies cannot be attached to user; policies written in human readable format; Default deny all;

IAM Policies

==========

1. Policy Syntax: Allow <subject> to <verb> <resource-type> in <location> where <conditions>

2. Verb: inspect(list), read(list+metadata), use(read+existing resource), manage(all permission)
3. Resource Type: Aggregate Resource Type (all-resources, instance-family etc), Individual Resource Type(instances, databases etc)
4. Verbs & Permissions - INSPECT & VOLUME INSPECT; USE & VOLUME_WRITE; MANAGE & VOLUME_CREATE -> API Operations
5. Common Policies: Network Admins, InstanceLaunchers
6. Advanced Policy Syntax: 2 types of variables added to conditions; request and target; Ex: request.operation, targets.group.name

IAM Compartments

==========

1. Organize and control access to resources
2. Compartment Quotas similar to Service Limits but set by Admins using policies; 3 types of quota policies (set, unset, zero);
3. Ex: zero compute quotas /*bm*/ in tenancy (zeroed out BM instance)
4. Main Menu -> Governance -> Compartment Explorer -> List all resources in compartment

Policy Inheritance and Attachment

==========

1. Compartment inherit policies from parent compartments; policy created must be attached to a compartment/tenancy (B:C, A:B:C);
2. Compartment move with all its contents; cannot have a same name; two compartment with same parent cannot have same name;
3. Policy implications - compartment hierarchy down to the compartment being moved, to a shared ancestor of current and target parent; policy attached directly to a compartment moved is not automatically updated and is invalid;

IAM-Tags

========

1. Tagging - Free Form Tags (Basic implementation, key/value) Ex: Env:Production, Project:Alpha; Defined Tags - more features and control;
   contained in tag Namespaces; Defined Schema, secured with policy; Ex: Namespace = Operations, Human Resources etc
2. Tag Namespace - container for a set of tag keys with tag definitions; key/value pair; Namespace.Key = Value; Tag Namespace cannot be deleted but retired; reactivate to use again; must be setup in tenancy to start using; variable can be used for volume
3. Ex: ${iam.principal.name} at ${oci.datetime}; Defined tags work with policies; Ex; use tag-namespaces

Virtual Cloud Network

==========

CIDR (Classless Inter Domain Routing)

=====

1. IP Address => Network address + Host address; Subnet mask - seperates IP into network and host; 0 address assigned to network address,
2. 255 address assigned to broadcast address; Ex: xxx.xxx.xxx.xxx/n; commonly used netmasks 8 bits(Class A), 16 bits(Class B), 24 bits(Class C)
3. IPv4 addresses 32 bit long with 4 octets of 8 bits each
4. 192.168.1.0/24 => 192.168.1.0 - 192.168.1.255
5. 128 64 32 16 8 4 2 1 -> 2^7 2^6 2^5 2^4 2^3 2^2 2^1 2^0
6. 192 = 1 1 0 0 0 0 0 0 (i.e 128+64)
7. /27 hosts - 2x2x2 = 8 subnets; 2x2x2x2x2= 32 hosts;/27 hosts - 2x2x2 = 8 subnets; 2x2x2x2x2= 32 hosts;

Intro VCN

=========

private network in OC datacenter -> firewall + communication gateway; covers single IPv4 CIDR; resides in single region; avoid overlapping with on-premise or other cloud ip ranges; Recommended RFC1918 range(10.0.0.0/8, 172.16/12, 192.168/16 - these IP ranges cannot be routed on internet)

Allowable OCI VCN size range is /16 to /30; VCN reservesfirst 2 ip and last 1 ip in each subnet CIDR; VCN is regional; Regional subnet spans across ADs in multiAD region; Subnets can be private, public; VNIC enables compute engine to connect to VCN;

IP Addresses:

==========

Private IP addresses - Each instance must have one private IP; Instances > VNICs (Primary and Secondary); VNICs can have up to 31 secondary private IPs; private IP can have optional public ip; Multiple VNICs on VM - use case virtual appliance, different VCN for management;

Public IP addresses - IPv4 address reachable from internet; assigne to a private IP; multiple public IPs can be assigned to a single resource;

Pub IP Types - Ephemeral and Reserved; Ephemeral can assigned to primary Ip only, 1 per VNIC only; Reserved can be upto 32;

No charge for using public IP (both Ephemeral and Reserved); provided by oracle for OCI Public LB, NAT Gateway, DRG, OKE etc (cannot choose/edit, but view)

For IGW and Autonomous DB (cannot choose/edit/view)

Routing and Gateways

==========

1. RT - contains rules about how IP packets can travel to different ip add addresses out of VCN; consists of set of rules specifies:
2. Destination CIDR block; next hop for traffic that matches that CIDR; Each subnet can have only one RT; No route rules req within VCN;
3. RT used only if the destination IP is not within VCN CIDR block; RT needs to be updated when a gateway(any type) is added;
4. IGW - provides path for network traffic b/w VCN and Internet; one IGW per VCN; add a rule in VCN RT with 0.0.0.0/0;
5. NAT GW - gives entire private network access to internet without assigning each host a public IP; only outbound from hosts;
6. use case: update, patches; can have more than one NAT GW
7. Service GW - let resources in VCN access public OCI services such as Object Storage without using IGW or NAT GW; traffic from VCN
8. that is destined to one of supported OCI public services used instance private IP for routing, travels over OCI network
9. fabric, and never traverses to internet; Use case: Back DB to Object storage; Service CIDR labels in RT Destination CIDR;
10. DRG - virtual router provides path for private traffic b/w VCN and destination other than internet; can use IPsec VPN or Fastconnect;
11. after adding DRG add an entry in RT for traffic flow (Route Target DRG); DRG is standalone object, attach it to VCN, 1:1 relationship

SSH Proxy for connecting to instance in private subnet from Bastion host

==========

ssh -t -o ProxyCommand='ssh -i ~/.ssh/id_rsa opc@BASTION_PUBLIC_IP -W %h:%p %r' -i ~/.ssh/id_rsa opc@INSTANCE_PRIVATE_IP

Peering

=======

1. Local Peering - Connecting multiple VCN with in same region; resources can communicate using private IPs; A local peering Gateway;
2. Add an entry in RT after creating LPG; no overlapping CIDR; no transitive peering
3. Remote Peering - connecting VCN in different regions; Use case: DR; Requires RPC(Remote Peering Connection) on DRG; RPC is connection
4. point for remotely peered VCN
5. HandsOn: Add an entry in SL and RT for the second VCN CIDR and vice versa, create LPG on both VCN;
6. Establish Peering Connection in one of the VCN LPG

Security VCN

==========

1. SL - common set of firewall rules associated with subnet; rules specify type of traffic allowed; associated at launch time or later; stateful or stateless
2. NSG - provides a virtual firewall for a set of cloud resources; set of rules that apply only to a set of VNICs in a single VCN;
3. Compute instances, LB and DB instances support NSGs; can select NSG as source or destination instead of CIDR in SL;
4. Oracle recommendation to use NSGs; SL + NSG - union of rules from both
5. Stateful Security Rules - Connection Tracking: response tracked and automatically allowed regardless of any egress rules;
6. Default SL rules are stateful
7. Stateless Rules - response traffic is not automatically allowed; must create a corresponding egress rule; NO connection tracking;
8. Use case: scenarios with large number of connections (Load Balancing, Big Data); Attach NSG to VNICs on compute instance

DNS

====

1. Default VCN Components - RT, SL, set of DHCP options; can't delete but change;
2. Internal DNS - enable instances to use hostnames instead of IP to talk to each other; Options - Internet and VCN resolver (default),
3. Custom resolver (resolve on-premise hostnames using IPsec VPN or FastConnect)
4. DNS label (optional); VCN - <VCN DNS label>.oraclevcn.com; Subnet - <subnet DNS label>.<VCN DNS label>.oraclevcn.com;
5. Instance FQDN - <hostname>.<subnet DNS label>.<VCN DNS label>.oraclevcn.com; Instance FQDN resolves instance private IP;

VCN Review

==========

- Subnets - 1 RT, 5* SL (can be increased)
- All hosts communicate within VCN
- SL manage connectivity north-south(incoming/outgoing VCN traffic) and east-west(internal VCN traffic b/w multiple subnets)
- OCI follows whitelist model
- NSG is recommended for use over SL;

Connectivity - VPN Connect and FastConnect

==========

Connectivity to On-premise Networks

==========

1. Public Internet - Internet Gateway/NAT Gateway; Reserved and Ephemeral IPs; Internet data out pricing (first 10TB free)
2. VPN Connect - IPSec authentication and encryption; 2 options - 1. OCI managed VPN service 2. Software VPN (running on OCI compute)
3. FastConnect - Private connection; separate from internet; consistent network experience; ports speeds in 1 Gbps and 10 Gbps increments;
4. VPC basics: VPN - using a public network to make end to end connection between two private networks in a secure fashion using a standard protocol (IPSec).
5. Tunnel - a way to deliver packets through the internet to private RFC1918 addresses.
6. Authentication - provide mechanism to authenticate who you are.
7. Encryption - packets needs to be encrypted, so they cannot be sniffed over the public internet.
8. Static routing - configure a router to send traffic for particular destinations in preconfigured directions.
9. Dynamic routing - using a routing protocol such as BGP to figure out what paths traffic would take.
10. IPSec Two modes - transport mode (IPSec encrypts and authenticates only the actual payload of the packet and header info stays intact)
11. Tunnel mode (IPSec encrypts and authenticates entire packet. After encryption, the packet is then encapsulated to form a new IP packet that had different header info); OCI supports only tunnel mode.
12. DRG can be used to establish a connection with on-premise via IPSec VPN or FastConnect; After attaching DRG add entry to RT for traffic flow;
13. VPN Connect(IPSec) - managed VPN service to securely connect on-premise to OCI VCN using IPSec; ensures secure remote connectivity via industry standard IPSec encryption; suitable for running POCs; offered for free; OCI provisions redundant VPN tunnels;
14. VPN Connect workflow - CPE Object (virtual representation of actual network device which terminates the IPSec tunnel; could be router, firewall or VA)

CPE = Customer Premises Equipment

==========

1. Create a VCN

2. Create a DRG

3. Attach DRG to VCN

4. Update VCN router to route traffic to DRG

5. Create CPE object and add on-premise router Public IP address

6. From DRG create an IPSec connection b/w CPE and DRG and provide a static or BGP routing

7. Configure on-premise CPE router

On-prem setup - LibreSWAN VM in AWS - https://docs.oracle.com/en-us/iaas/Content/Network/Reference/libreswanCPE.htm

SL - Ingress rules for TCP 4500, 500 and UDP 4500, 500

On LibreSWAN VM, update /etc/ipsec.config (update VPN tunnel public IP) and /etc/ipsec.secrets (update IP and shared secret);

run ipsec verify and sudo service ipsec restart; add entry in VCN RT and run sudo ipsec auto --status | grep "==" in Libreswan VM;

FastConnect

==========

1. Provides a dedicated and private connection with higher bandwidth options, and a more reliable and consistent networking experience when compared to internet based connections.
2. Connect to OCI directly or via pre-integrated network partners;
3. Ports speeds of 1 Gbps and 10 Gbps increments;
4. Extend remote DC into Oracle ("Private Peering") or connect to Public resources ("Public Peering" - doesn't use DRG);
5. No charges for inbound/outbound data transfer;
6. Uses BGP protocol;
7. Connection scenarios: 1. colocation in data centre; 2. through oracle provider (Microport, AT&T/Layer 2 or 3)
8. Virtual Circuit - isolated network path that runs over one or more physical network connections to provide a single, logical connection b/w customer's edge router and their DRG.

Load Balancer

==========

Load Balancing Intro

==========

1. Primer - LB sits b/w clients and backends; perform tasks such as: Service Discovery, Health Check, Algorithm; LB benefits: FT and HA; Scale, Naming abstraction
2. OCI Load Balancing Service - provides HA and scale; Public and Private LB options; Supported Protocols: TCP, HTTP/1.0, HTTP/1.1, HTTP/2, WebSocket supports SSL termination, end-to-end SSL, SSL tunneling; adv features such as session persistence and content based routing;
3. Key differentiator - Private or Public LB(with Public IP); Provisioned bandwidth(100 Mbps, 400 Mbps, 8 Gbps); Single LB for TCP(L4) and HTTP(L7);
4. Public LB - Accepts traffic from internet; regional service; in multi-ad regions, regional set or 2 AD specific subnets required;
5. LB service creates primary and standby LB, each in a different AD; supports AD failover; Floating public IP is attached to primary and in the event of AD outage Floating IP is attached to standby LB;
6. Concepts: Listener - entity that checks for incoming traffic on LB's IP address; Backend server - App servers responsible for generating content in reply to the incoming TCP or HTTP traffic;

Load Balancing Policy(round robin, IP hash, least connection) - tell the LB how to distribute traffic to backend servers; Backend set - logical entity defined by a list of backend servers, a load balancing policy and a health check policy; Health check - test to confirm availability of backend servers, support TCP/HTTP level health checks;

7. LB Policies: Round Robin - default, distribute traffic sequentially to each backend server; IP hash - request from a particular client are always directed to same backend server; least connection - routes incoming non sticky request traffic to backend server with the fewest active connections;
8. TCP LB considers policy and weight criteria; HTTP LB w/cookie based session persistence forwards requests using cookie's session info;
9. For non-sticky HTTP requests, LB applies policy and weight criteria
10. Health check - activated for backends, backend sets, overall LB; A LB IP can have 16 listeners(Port numbers); Each listener = backend set with 1 or more backend servers; Health API provides 4-state health status - ok, warning, critical, unknown; Health status updated every 3 mins, no fine granularity available; Backend Actions - Drain state(maintenance), Offline state, Backup state(DR);

Private Load Balancer

==========

1. Assigned a private IP address from the subnet hosting the LB; can be regional or AD specific; In AD specific subnet both primary and failover LB in same AD;

Compute

=======

Compute Intro

==========

2. Form Factors - Bare Metal(BM): Direct hardware access/single tenant model; Virtual Machines(VM): A hypervisor to virtualize the underlying
3. BM into smaller VMs/multi-tenant model; Dedicated VM hosts (DVH): run VM on dedicated servers/single tenant
4. BM use cases - workloads that are performance intensive; workloads not virtualized; workload req specific hypervisor; BYO licensing
5. BM shapes - BM.Standard, BM.DenseIO, BM.HPC, BM.GPU etc; 1 OCPU equivalent to one physical core of a processor with hyperthreading enabled;
6. AMD EPYC Use cases - cheaper, ideal for maximizing price performance; supports oracle apps ebs, JD edwards, peoplespft etc; certified to run Cloudera, Hortonworks, MapR, Transwarp; HPC workloads

Images

======

1. Oracle provided images, Custom, BYOL
2. Oracle provided images - template of virtual hard drive that determines the OS and other s/w for an instance;
3. Linux image username = opc(oracle/centos), ubuntu; default firewall rule ssh only; provide startup script using cloud init windows image username = opc with one time pw; includes windows update utility
4. Custom image - create from instance boot disk and use it to launch other instances; during creation instance shutdown and remains unavailable; only includes boot volumes; image size cannot exceed 300 GB; windows custom image cannot be exported or downloaded out of tenancy;
5. Image Import/Export - share custom images across tenancies and regions; uses OCI object storage; supports both windows and linux;
6. Supported modes - Emulation mode(emulated hardware/IO), Paravirtualized(includes driver to enable virtualization), Native mode(HVM)
7. BYOI - enables lift-and-shift cloud migration projects, support old/new OS, encourage experimentation, increase infra flexibility;
8. on-prem -> qcow2 image format -> Object storage -> Custom image -> instance

Boot volumes

==========

1. Created automatically and associated with an instance until you terminate the instance; encrypted, have fast performance, lower
2. Launch time, higher durability for BM and VM instances; can be scaled to a larger shape using boot volumes; can be preserved during
3. Termination; cannot be detached from running instance; can manually backup, assign backup policy, create clone of boot volumes;
4. Custom boot volumes - can specify custom boot volume size; Linux default 46.6 GB max 100 GB; windows default 256 GB max 500 GB;
5. Custom image - can be shared across regions/tenancies; no cost to store; instance not available during creation; limit 25/compartment
6. Boot volume backup - no downtime; preserve entire state of running OS as backup; cost to store; creates crash-consistent backup;
7. Cannot do boot volume backup/boot volume clone at the same time

Instance configuration, pools, and Autoscaling

==========

1. Running instance -> Config (OS image, metadata, shape, VNICs, Storage, subnets)
2. Config -> Multiple instances (different ADs, Manage all together, Attach to LB)
3. Instance config and pool - Use cases:
4. IC - clone an instance and save to config file; create standardized baseline instance template; easily deploy instances from CLI with a single config; automate the provisioning of many instances, its resource and handle the attachments.

5. IP - Centrally manage a group of instance workloads; update large no of instances at a time; maintain HA and distribute across AD; scale out instances on-demand by increasing size of the pool
6. Autoscaling Configurations - enables automatically adjust no of instances in an instance pool based on perf metrics such as CPU or memory;

Instance metadata and Lifecycle

==========

1. Instance metadata - includes OCID, name, compartment, shape, region, AD, creation date, state, image, custom metadata such as SSH key etc;
2. Service runs on every instance and is an HTTP endpoint listening on 169.254.169.254; can get instance metadata by logging/using metadata service; oracle provided linux instance - curl http://169.254.169.254/opc/v1/instance, /metadata, /metadata/<key-name>/; can add/update metadata
3. Instance Lifecycle - Start, Stop, Reboot, Terminate (can preserve boot volume and attach to a different instance as data volume or launch a new instance); Resource billing - Standard shape, billing pauses in STOP state; Dense I/O, billing continues in STOP state;
4. GPU shape, billing continues in STOP state; HPC shape, billing continues in STOP state

Block Volume

==========

Local NVMe

==========

1. some instance shapes include locally attached NVMe devices; used for workload requiring high storage performance requirements;
2. not protected, no RAID, no snapshots, no backup; customer responsible for durability of data; Ex: BM.DenseIO2.52(8 drives / 51.2TB),
3. VM.DenseIO2.8/16/24 (2/4/8 drives / 6.4/12.8/25.6 TB); Protecting NVMe SSD devices - RAID1; RAID10, RAID6; SLA - Min suppoted IOPS;

Block Volume

==========

1. Let you store data on block volumes independently beyond lifespan of compute instances; uses protocol such as iSCSI; create, attach, Connect, move volumes, as needed; Typical scenarios - Persistent and durable storage, Expand an instance storage, Instance scaling;
2. Capacity 50GB to 32 TB(1GB increments); Disk type NVMe SSD; IOPS 2-75 IOPS/GB upto 35K IOPS*; Throughput/vol upto 480MB/s;

3. Latency Sub ms; Per instance limits 32 attachment/instance upto 1 PB up to 620K or more IOPS; Durability Multiple replica across AD;
4. Security Encrypted at Rest and in-transit
5. Block Volume Elastic Performance - Performance Level (Lower cost[NO VPU charge, 2 IOPS/GB, NA for Boot Vol], Balanced[purchase 10 VPU/GB,
6. 60 - 25K IOPS], Higher Performance[purchase 20 VPU/GB, 75 - 35K IOPS])
7. Attach Block Volume - iSCSI or Paravirtualized; By Default all Block volumes, Read/Write; can also be read only to prevent modification;
8. Detach and Delete Block volume - cannot undo delete operation;
9. Block volume offline Resize - expand size of block/boot volumes; Expand existing vol in place with offline resizing(cannot resize attached vol),
10. Restore from vol backup to a larger volume, Clone an existing vol to a new, large volume
11. Balanced Performance - Default for Block/Boot volumes; Change performance is dynamic; Backup/Clone;

Backup and Restoration

==========

1. complete point-in-time snapshot copy of block vol; encrypted and stored in Object storage and restored as new vol to any AD within same region;
2. can copy block vol from one region to another; 2TB vol takes 30 min first time, 50GB boot vol takes few mins; on-demand, one-off block
3. vol backups provide a choice of incremental vs full backup options; Automated policy based backups - Bronze (monthly incremental/12 months), silver(weekly incremental/4 weeks), Gold(daily incremental/7 days); Customized backup policy NA today; Yearly backup full retained 5 years

Cloning and Volume Group

==========

1. Clone - copying an entire existing block volume to a new volume without needing to go thru a backup and restore process;
2. copying takes 15 mins for 1 TB volume; can be created in same AD, no need to detach the volume; cannot be copied to another region;
3. source volume attached - 1 clone at a time, detached 10 clone at a time
4. Volume Groups - Group together block and boot volume across multiple compartment across multiple compute instances; create volume group
5. backups; manually trigger full or incremental backup; ideal for protection and management of enterprise applications; no addl cost

Boot Volumes

==========

1. Attach a Boot volume to an instance as a Block volume for troubleshooting - stop instance and click on 'boot volume' filter, detach
2. boot volume, attach as block volume in a running instance

File Storage

==========

File Storage Intro

==========

1. Use cases - Oracle Apps Lift and Shift, General Purpose File Systems, Big Data and Analytics, HPC Scale Out Apps, Test/Dev DB, Microservice containers
2. FS service features: AD-local; supports NFS v.3; NLM for file locking; Full POSIX semantics; Data protection(10000 snapshot/FS);
3. security(128 bit, data-at-rest enc for all FS & metadata); Console Management, APIs, CLI, data-path commands, and Terraform;
4. can create 100 FS and 1 mount targets per AD per account
5. Mount target - NFS endpoint that lives in subnet; AD-specific; has an IP and DNS name to use in mount command; requires 3 private IPs in the subnet (dont use /30); 2 IPs for mount target creation, 1 IP for HA; Best practice to place FSS mount target in its own subnet
6. File System - primary resource for storing files in FSS; to access FS, create a new mount target; 100 FS/mount target; accessible from OCI VM/BM instances; accessible from on-premises thru FastConnect/VPN
7. Export Path - make a file system available through a mount target; unique path; FS cannot have overlapping export paths;
8. sudo mount 10.0.0.6:/example1/path /mnt/mountpointA
9. Mounting an OCI File System –
    a. Launch OCI instance from console;
    b. Use NFSv3 protocol to mount FSS volume
    c. Install nfs-utils or nfs-common
    d. Create a directory
    e. on FSS console, click on mount targets
    f. Use private IP to mount volume using nfs command
        i. open ingress firewall for both tcp and udp ports 111, 2048-2050;open egress firewall for source port both tcp and udp ports 111, 2048-2050

File Storage Security

==========

1. 4 distinct & seperate layers of security; 1. IAM Service(OCI useres, policies), 2. Security Lists(CIDR blocks),
2. Export Options(Export Options, CIDR blocks), 4. NFS v3. Unix Security(Unix users)
3. Security List - all or nothing
4. Export options - limit clients ability to connect to FS and view or write data; created automatically and allow full access for all

5. NFS clients; give specific access to NFS clients (READ ONLY, READ-WRITE)

File Storage Snapshots

==========

1. read-only, space efficient, point-in-time backup of FS; created under root folder, hidden directory(.snapshot); soft limit 10000;
2. restore snapshot - cp -r .snapshot/sn_name/* destination_dir_name;

Object Storage

==========

Object storage Intro

==========

1. internet scale, high performance storage platform; ideal for storing unlimited unstructured data(image, media files, logs, backup);
2. data managed as objects using API; Regional service; 2 distinct storage classes: hot and cold; support private access thru service gateway;
3. supports adv features such as cross region copy, pre-auth requests, lifecycle rules, multipart upload;
4. OS scenarios - Content Repository, Archive/Backup, Log data, Large datasets, Big Data/Hadoop support, HDFS connector
5. OS service features - Strong consistency, Durability (data stored redundantly in AD/FD), Performance, Custom Metadata, Encryption
6. OS resources - Object (object+metadata), Bucket(logical container, must be unique within tenancy), Namespace(logical entity)
7. Object naming - /n/os_namespace/b/bucket/o/obj_name; Flat Hierarchy; use prefixes and hierarchies;
8. Standard Storage Tier(HOT) - fast, immediate, frequent access; strong consistency, retrieval is instantaneous; cant downgraded to archive
9. Archive Storage Tier(COLD) - rarely accessed, compliance, audit logs etc; min retention 90 days; restore before download; TTFB 4 hrs
10. can't upgrade to Standard Tier

Object Storage Capabilities

==========

1. Managing Access and Authentications - Pre-authenticated request (access without credentials); unique url; can revoke access any time;
2. All buckets created as private by default; making public gives anonymous access; pre-auth default 1 week;

3. Cross-region Copy - Use case: DR, compute instance from custom image in diff region; write an IAM policy for each region;
4. bulk copying not supported; cannot be copied from Archive storage
5. Object Lifecycle Management - rules to automatically archive/delete after a specified no of days; requires IAM policy;
6. bucket or prefix level; delete rules take priority than archive; rule can be enabled/disabled
7. Multi-part upload - upload in parallel to reduce amount of time;
    a. Create object parts
    b. Initiate an upload(CreateMultipartUpload API)
    c. Upload object parts (restart from failed part)
    d. Commit the upload

Oracle Database on OCI

==========

1. OCI Database service - Mission critical, enterprise grade cloud DB service(Exadata, RAC, BM, VM); Complete lifecycle automation;
2. HA and scalability (RAC and Data Guard, Dynamic CPU and Storage scaling); Security(TDE, Encrypted RMAN backup/Block vol encryption);
3. OCI platform integration; BYOL
4. VM DB Systems - 2 Types: 1-node(1VM), 2-node(2VM clustered with RAC enabled); can have single DB home(single DB); memory allocation
5. depends on VM shape; storage can be scaled but CPU core cannot be changed; can select old db versions; Data guard across AD requires
6. Enterprise Edition;
7. VMDB System Storage Architecture - Block Storage -> ASM Disk Groups + Data, +RECO -> Data | RECO -> ASM
8. VMDB Fast Provisioning - Block storage -> Physical Vol on VM -> Vol Group on VM -> Logical Vol -> ext4 FS System mount(u01-BITS, u02-DATA, u03-RECO)
9. VM RAC DB cannot be deployed in Fast Provisioning; supports only 18c and 19c; storage scaling depends on initial storage specified;
10. BM DB Systems - rely on BM servers running Oracle Linux; 1-node DB system(single BM server/Locally attached 51 TB NVMe storage, start
11. with 2 core and scale up/down OCPU based on requirement (52CPU core/768GB RAM), Dataguard within/across ADs(Enterprise Edition), if
12. node fails launch another system and restore db from current backup)
13. BM DB Storage Architecture - NVMe -> ASM Disk Groups + Data, +RECO -> Data | RECO -> ASM (auto repair/notify failure via ticket)
14. Exadata DB Systems - Full Oracle Db with all advanced options; On fastest and most available db cloud platform (scale out CPU/storage,
15. infiniband, PCIe flash); All Public cloud benefits; specify zero cores when launch Exadata -> provision and stops Exadata;
16. Billed for First month, then by the hour; Each OCPU added billed by the hour; Scaling from 1/4 to 1/2 rack required DB deployment is backed up
17. Exadata DB System Storage Architecture - Local Storage -> ASM Disk Groups + Data, +RECO -> Data | RECO -> ASM

18. Backup provisioned on Exadata storage (40% DATA 60% RECO) Backup not provisioned on Exadata storage(80% DATA 20% RECO)
19. Database Editions and Options - Standard, Enterprise, EE High Performance, EE Extreme Performance (TDE in all editions)

Managing DB Systems

==========

1. Console - Launch DB system; Start/stop/reboot DB systems (Billing continue in STOP except VM DB); Scale CPU cores (BM DB only);
2. Scale up storage (VM DB system only); Terminate DB system (permanently deletes db running on it; take manual backup/data pump to OCI OS)
3. Patching DB Systems - Automated Applicable Patch Delivery; N-1 patches; Availability during patching (rolling for Exadata and RAC,
4. Data Guard required for 1-node otherwise downtime); 2 step process (DB system first, then DB is patched); Identity and Access controls;
5. Backup/Restore - managed backup for VM/BM DB systems; Exadata requires creating Backup config file; Backup stored in OS or Local Storage;
6. DB in private subnet leverage Service Gateway; Backup options - Automatic incremental (once a day/retained 30 days),
7. On-demand/Standalone/full backups; Restore from latest/timestamp/SCN;
8. Automatic Backups - Oracle owned OS; cannot be viewed; enabled to run b/w midnight and 6 AM; can specify 2 hr scheduling window; retention periods - 7, 15, 30, 45, and 60 days; backup jobs retry automatically; Oracle notified if backup job stuck; All backup encrypted
9. HA and Scalability - Robust infra(2-way or 3-way mirrored storage, redundant inifiniband fabric for cluster networking); DB options to enable HA(RAC for VM and Exadata, Automated Dataguard across ADs); Dynamic CPU and Storage scaling
10. Oracle Data Guard - supported on VM and BM; Limited to one standby db per Primary db on OCI; Switchover(upgrade)/Failover(DR) - can be manually invoked by Enterprise Manager, DGMGRL or SQL*PLUS
11. Security Features for DB - Instance security isolation (BM DB); Network Security and access list; Secure and HA connectivity;
12. Use Authentication and Authorization; Data encryption; End-to-end TLS, Auditing

Oracle Autonomous Databases

==========

1. Autonomous Database - Fully automated DB operations; User runs SQL no access to OS or CDB; Exadata performance and Availability;
2. Customizable for DW or TP workload; 2 Types - 1. Serverless(Ultra-simple, Elastic) 2. Dedicated(Customizable Private Cloud)
3. Use cases for ADB - Cloud Elasticity, ML, Self driving Instance provisioning, Always online operation, All workload, JSON docs, Graphs etc
4. Oracle DBCS - Small to Large DB deployments, Single Instance or RAC, Automated Backup, Patching, Customer Controls

5. Exadata - Private/Public on-premise, Consolidation, High Performance, Scalability for Mission critical workload
6. Oracle DB - Small to Big DB transactional/DWH needs, Customer DC, DIY Model
7. Autonomous Optimizations - Specialized by Workload:

ADW	ATP
Columnar format	Row format
Create Data summaries	Creates Indexes
Memory speeds joins, aggs	Memory for caching to avoid IO

8. Statistics updated in real time while preventing plan regressions
9. Autonomous DB Deployment Options - Dedicated and Serverless
10. Dedicated - enable to provision in own dedicated Exadata cloud infra instead of shared infra with other tenants
11. Serverless - simplest config, share resources of Exadata cloud infra; quick and no min commitment;
12. Both options available for ADW and ATP
13. ADB Serverless - Oracle automates end to end mgmt of ADB; provisioning new DB, Growing/shrinking storage or compute/patching and upgrade, backup and recovery; Full lifecycle managed using service console (or via CLI/REST API);
14. Automated Tuning in ADB - Define Tables, load data, run queries; Fast performance out of box with zero tuning; Simple web-based monitoring console; Built-in resource mgmt. plans
15. ADB supports Third party BI tools, DI tools, Oracle Cloud Services; Connectivity via SQL*Net, JDBC, ODBC

Getting Started with ADB

==========

Provisioning an ADB requires

1. Database Name

2. Which DC?

3. How many CPU cores?

4. How much storage capacity?

5. Admin Password

6. License Type

7. Enable Autoscaling

1. Autoscaling ADB - automatically increase CPU core upto 3 times; can be enabled/disabled any time; Billing based on avg CPU used per hr;
2. Securing ADB - all data encrypted; DB client uses SSL/TLS1.2; IP restriction using Access control list
3. Troubleshooting ADB - firewall must allow 1522; Service Gateway/NAT GW for accessing from private subnet

4. Scaling ADB - Independently scale compute or storage; resize instantly, fully online; Memory, IO bandwidth, concurrency scales linear with CPU;
5. Close DB to save money when not used; Restart instantly;
6. Monitoring ADB - Service console based(web based, historical and real time); Performance Hub based monitoring(natively integrated, ASH)
7. ADB Backup and Recovery - auto backup, retention 60 days, weekly full, daily incremental, initiate recovery from console, NACLs stored in DB
8. ADB Cloning - full DB or only DB metadata; Full clone min storage rounded to next TB, can clone in same tenancy/regional;
9. Predefined Services for ADW - High (low concurrency), Medium(higher concurrency), Low(highest concurrency)
10. Predefined Services for ATP - High(For Reporting or Batch, run parallel and queueing), Medium(For Reporting or Batchrun parallel and queueing), Low(For Reporting or Batch), TPURGENT (For TP), TP(For TP)
11. ADB - Dedicated: private DB cloud running on Exadata infra in public cloud; multiple levels of isolation; customizable operational policies give more control; 1 cluster per quarter rack; HA SLA 10 DBs, Extreme Availability 25 DBs
12. High level deployment flow: Create VCN -> Provision ADB Exadata infra -> Create AContainerDB -> Create ADB
13. Security - DBs are always encrypted; Reduced attack surface; Database vault, Security vulnerabilities