5 EC2 Basics

EC2 Architecture and Resilience

EC2 Instances run on EC2 hosts using a hypervisor, that manages the instances and is responsible :

- virtualization of the hardware (cpu, memory, network card etc.).
- the isolation of the instances from each other and from the host.

EC2 hosts can be **shared hosts or dedicated hosts.** If you pay for dedicated host, you pay for the entire host and not the individual instances running on it.

Hosts are in 1 AZ. If the host fails, all instances on it will fail. **If the AZ fails, all hosts in it will fail. So EC2 is AZ resilient.** (important for exam)

If an instance is stopped and then started again it will be relocated to another host. This host will also be in the same AZ. If an instance is just restarted, it will stay on the same host.

You can't connect EBS volumes from one AZ to an instance in another AZ. You can't connect an instance in one AZ to an EBS volume in another AZ. Both services are AZ bound.

Use of EC2:

- traditional server: OS + applications
- Long running compute tasks
- burst or steady state workloads
- monolithic applications
- migrated application workloads
- disaster recovery

EC2 Instance Types

Instance type: R5dn.4xlarge

- R: instance family
- 5: generation
- d: instance has local storage (additional capabilties - optional)
- n: instance has network optimization (additional capabilties - optional)
- 4xlarge: size of the instance - number of vCPUs and memory

Instance types:

- **General purpose:** balance of compute, memory and networking resources
  - ***A1, M6g:*** Arm based processors (Graviton)
  - ***T3, T3a:*** burstable performance - cheaper assuming nominal low levels with occasional spikes
  - ***M5, M5a:*** Steady state workloads, alternative to T3/T3a - Intel/AMD architecture
- **Compute optimized** high performance processors
  - ***C5, C5n:*** compute intensive applications: media encoding, gaming, Scientific modeling
- **Memory optimized:** high memory to CPU ratio

    - ***R5, R5a:*** memory intensive applications: in-memory caches, real-time analytics, certain DB applications (in-memory operations)
    - ***X1, X1e:*** high performance databases, in-memory databases, real-time processing of big data. Lowest cost per GiB of RAM
    - High memory u-Xtb1: Highest memory of all AWS instances
    - z1d: high frequency processors, high memory, high compute
- **Accelerated computing:** hardware accelerators or co-processors
    - ***P3, P3dn:*** GPU Instances (Tesla v100 GPUs) - machine learning, deep learning, high performance computing
    - ***G4dn:*** GPU Instances (NVIDIA T4 Tensor) - graphics intensive applications, machine learning, video encoding
    - ***F1:*** hardware acceleration for custom hardware
- **Storage optimized:** high, sequential read and write access to large data sets on local storage
    - ***I3, I3en:*** Local high performance SSD (NVMe) NoSQL DBs, data warehousing, Elasticsearch, real-time analytics
    - ***D2:*** Massively parallel processing data warehousing, MapReduce and Hadoop distributed computing
    - ***H1:*** Big data analytics, Apache Hadoop, Apache Spark, log processing

Storage refresher

- Direct attached storage (local): storage on the EC2 host
- Network attached storage - Volumes deliverd over the network (EBS)

Storage can be ephemeral or persistent:

- Ephemeral Storage - Temporary storage - lost when the instance is stopped eg. instance store volumes (local storage)
- Persistent Storage - Permanent storage - lives on past the lifetime on the instance eg. EBS volumes Important for exam: know what storage is ephemeral and what is persistent.

Types of storage:

- Block storage:
    - Volumes presented to the OS as a collection of blocks
    - no structure provided, it's a collection of blocks. The OS has to manage the structure
    - mountable and bootable (most EC2 instances boot from block storage)
    - EBS volumes are block storage
- File storage:
    - presented to the OS as a file share
    - has provided structure (folders, files etc)
    - mountable
    - NOT bootable
    - EFS is file storage
- Object storage:

    - collection of objects with metadata
    - flat structure (no folders etc)
    - NOT mountable
    - NOT bootable
    - sclable, durable, secure, low cost
    - S3 is object storage

Storage performance:

- IO (block) size: size of the data block being read or written (eg. 4KB)
- IOPS: Input/Output Operations Per Second
- Throughput = IO size * IOPS in MB/s --> Maximize throughput by increasing maximizing IOPS for the right block size

Elastic Block Store (EBS) Service Architecture

Block storage service for EC2 instances

- volumes are raw disks connected to EC2 instances through the network
- volumes can be encrypted using KMS
- EC2 instances sees block device and can format it with a file system (eg. ext4 (linux), NTFS (windows))
- EBS volumes are provisioned in 1 AZ => are AZ bound (AZ resilient), you can't connect an EBS volume from one AZ to an instance in another AZ. Because EBS in 1 AZ is different from EBS in another AZ (different hardware, different network etc.)
- EBS volumes are attached to 1 instance at a time. They can be attached to multiple instances at the time using multi-attach (but this is not the default and has to be managed to prevent data corruption(e.g. wrtiting to the same block at the same time))
- EBS volumes can be detached from an instance and attached to another instance
- EBS volumes are not lifecyle bound to an instance, they can live on past the lifetime of an instance.
- EBs volumes can be backed up using snapshots (incremental backups) - snapshots are stored in S3

EBS Volume Types

General Purpose SSD (gp2 and gp3)

Provisioned IOPS SSD (io1 and io2)

Provisioned IOPS SSD — Provides high performance for mission-critical, low-latency, or high-throughput workloads. Used for smaller volumes but need high performance (High IOPS) - eg. databases

3 types of IOPS:

- io1
- io2
- io2 Block Express

Hard Disk Drives (HDD) (st1 and sc1)

- st1: Throughput optimized HDD

    - Big data, data warehousing, log processing
    - fast throughput, low cost
    - 125 GB - 16 TB
    - performance:
        - 1 MB IOPS
        - Max 500 IOPS -> max 500 MB/s
        - Base performance 40 MB/s per TB
        - Burst performance 250 MB/s per TB (hardcap at max 500 MB/s)
- sc1: Cold HDD
    - Lowest cost storage for infrequently accessed workloads
    - 125 GB - 16 TB
    - performance:
        - 1 MB IOPS
        - Max 250 IOPS -> max 250 MB/s
        - Base performance 12 MB/s per TB
        - Burst performance 80 MB/s per TB (hardcap at max 250 MB/s)

EC2 Instance Store Volumes

- Ephemeral storage block storage - temporary storage (changing host, stopping instance, changing instance type or hardware failure causes data loss)
- Directly attached to one EC2 host - instances on that host can access it (not via network)
- 1 or more volumes per instance can be attached
- Lost when the instance is stopped (ephemeral)
- High performance, low latency (faster than EBS (network attached storage))
- included in the price of the instance (you pay for it anyway)
- Volumes are attached at launch time and can't be attached later (exam tip)

Choosing between EBS and Instance Store Volumes

Network interfaces, Instance IPs and DNS

- EC2 instances have at least 1 (primary) Elastic Network Interface (ENI) but you can add more ENI's that can be in another subnet but in the same AZ.
- Networking attributes of an instance are tied to the (primary) ENI. An ENI has:
    - MAC address
    - 1 primary private IPv4 IP address
    - 1 or more secondary private IPv4 IP addresses
    - 0 or 1 public IPv4 address (if instance is in a public subnet) - inside VPC Internet Gateway translates public IP to private IP and vice versa. So inside VPC you use private IP's.
    - 1 Elastic IP address (public) per private IP address (if assigned removes public IPv4 address)
    - 0 or more IPv6 address (always public) (if enabled)
    - Security Groups
    - Source/Destination Check flag (enabled by default, must be disabled for NAT instances): checks if the source or destination is the IP address of the interface. If not, the packet is dropped because the attached instance is not concerned by package. This is useful for security but not for NAT instances.

- secondary ENI's have the same attributes as the primary ENI but they can be detached and attached to another instance. They can be in another subnet but in the same AZ.

exam questions:

- if you remove elastic IP can you recover previous public IP? No, you can't, a different public IP will be assigned.
- if you stop an instance in a public subnet and start it again, will it have the same public IP? No, it will have a different public IP, because the public IP is assigned at launch time. They are dynamic.
- if you restart an instance in a public subnet, will it have the same public IP? Yes, because the public IP is assigned at launch time.

exam tips:

- secondary ENI's are usefull for legacy software licensing that is tied to the MAC address of the instance. You can detach the ENI and attach it to another instance.
- different security groups can be assigned to different ENI's of the same instance. So you can have different security rules for different ENI's of the same instance: eg management ENI and data ENI.
- OS never sees the public IPv4 address, it only sees the private IPv4 address. The public IPv4 address is translated by the VPC Internet Gateway to the private IPv4 address and vice versa.

Amzon Machine Images (AMI)

- Amazon Machine Images (AMI) 's are the images which can create EC2 instances of a certain configuration.

- AMI's are stored in S3 - so they are regional objects

- lifecyle:

    - launch: create an instance with existing AMI
    - configure: customize the instance - install software, attach certain type of EBS volumes etc.
    - create image: create a new AMI from the instance (baking)
    - launch: create new instances from the newly created AMI

- AMI's can be:

    - private: only your account (default)
    - shared: shared with specific accounts
    - public: shared with everyone

- AMI is a container for the following:

    - root volume: boot volume of the instance
    - block device mapping: maps the block device id (eg. /dev/sda1) to the snapshot id of the EBS volume
    - launch permissions: who can launch the AMI
    - metadata: name, description, version, architecture, kernel ID, RAM disk ID, block device mapping

  Exam power ups:

    - AMI is a regional object stored in S3, you can use it in all AZ's in the region
    - AMI's can't be edited, you have to create a new AMI from an instance with the updated configuration
    - AMI's can be copied between regions and this includes the snapshots of the EBS volumes
    - AMI's can be shared with other accounts, by default only your account but you can share it with specific accounts or make it public
    - billing: you pay for the cost of the EBS snapshots the AMI references. Same as normal EBS snapshots: you pay for the size of the snapshot not the size of the volume.

EC2 Purchasing Options (launch types)

On Demand

- default purchasing option
- instances are isolated but they run on shared hardware
- instances of different sizes run on the same hardware consuming the allocated resources
- `per second billing while instance is running (minimum 1 minute)`
- no upfront payment, pay as you go
- predictable pricing
- `No reserved capacity (reserved instances get priority over on demand instances in case of capacity issues)`
- good for:
    - short term, unpredictable workloads
    - development and testing (unknown workloads)
    - applications with short term spikes
    - applications that can't be interrupted (but short term)

Spot Instances

- buy unused EC2 instances at a discount (up to 90%) compared to on demand
- set spot price (max price you are willing to pay):
    - if current spot price is below your max price, you get the instance and pay the current spot price
    - if current spot price goes above your max price, your instance is terminated
- `never use spot instances for workloads that can't be interrupted`
- use for burst workloads that can be interrupted (split into multiple loads): eg. big data, analytics, containerized workloads, CI/CD, rendering, machine learning
- use for statelss workloads (no data stored on the instance)

Reserved Instances

- long term commitment (1 or 3 years) to EC2 instances
- `up to 75% discount compared to on demand`
- if reserved instance is not used, you still pay for it
- 3 types of reserved instances:
    - Standard: fixed term, fixed payment
    - Convertible: can change the instance type, OS, tenancy
    - Scheduled: not supported by all instance types or regions

        - reserved instances for specific time windows (eg. every day from 9-5)
        - or reserved instance for specific recurring events (eg. every friday for 24h)
        - or reserved instance number of hours per month (eg. 100 hours per month)
- reduce costs by paying upfront:
    - no upfront (per second fee)
    - partial upfront (reduced per second fee)
    - all upfront (no per second fee)

Dedicated Hosts

- physical server dedicated for you
- server is designed for specific family of instances (eg A, T, R)
- `pay for the entire host, not the individual instances`
- use if licensing is tied to the physical server (exam question)
- host affinity: instances are placed on the same host (if possible)
- all instances on the host are your instances -> no noisy neighbors
- capacity management is required

Dedicated Instances

- instances run on dedicated hardware but not the entire host
- only your instances run on the hardware
- use if compliance requirements require no shared hardware with other AWS customers
- you don't want to manage the host yourself