Elastic Compute Cloud

Amazon Elastic Compute Cloud (EC2) is a web service that provides resizable compute capacity in the cloud. Amazon EC2 reduces the time requires to obtain and boot new server instances to minutes, allowing you to quickly scale capacity, both up and down, as your computing requirements change.

EC2 Pricing Models

1. On Demand
    - Allows you to pay a fixed rate by the hour (or by the second) with no commitment
2. Reserved
    - Provides you with a capacity reservation, and offer a significant discount on the hourly charge for an instance. Contract Terms are 1 year or 3 year terms.
3. Spot
    - Enables you to bid whatever price you want for instance capacity, providing for even greater savings if your applications have flexible start and end times
4. Dedicated Hosts
    - Physical EC2 server dedicated for your use. Dedicated hosts can help you reduce costs by allowing you to use your existing server-bound software licenses.

On Demand

On Demand Pricing is useful for:

- Users that want the low cost and flexibility of Amazon EC2 without any up-front payment or long-term commitment
- Applications with short term, spiky, or unpredictable workloads that cannot be interrupted
- Applications being developed or tested on Amazon EC2 for the first time

Reserved Pricing

Reserved pricing is useful for:

- Applications with steady state or predictable usage
- Applications that require reserved capacity
- Users able to make upfront payments to reduce their total computing costs even further

Reserved pricing types

1. Standard Reserved instances
    - These offer up to 75% off on demand instances. The more you pay up front and the longer the contract, the greater the discount.
2. Convertible reserved instances
    - THese offer up to 54% off on demand capability to change the attributes of the RI as long as the exchange results in the creation of Reserved Instances of equal or greater value
3. Scheduled Reserved Instances
    - These are available to launch within the time windows you reserve. This option allows you to match your capacity reservation to a predictable recurring schedule that only requires a fraction of a day, a week, or a month

Spot Pricing

Spot pricing is useful for

- Applications that have flexible start and end times
- Applications that are only feasible at very low compute prices
- Users with urgent computing needs for large amounts of additional capacity

If the Spot Instance is terminated by Amazon EC2, you will not be charged for a partial hour of usage. However, if you terminate the instance yourself, you will be charged for any hour in which the instance ran

Dedicated Hosts Pricing

Dedicated Hosts pricing is useful for

- Useful for regulatory requirements that may not support multi-tenant virtualization
- Great for licensing which does not support multi-tenancy or cloud deployments
- Can be purchased on-demand (hourly)
- Can be purchased as a reservation for up to 70% off the on-demand price

EC2 Instance Types

Family	Speciality	Use case
F1	Field Programmable Gate Array	Genomics research, financial analytics, real-time video processing, big data etc
I3	High Speed Storage	NoSQL DBs, Data Warehousing etc
G3	Graphics Intensive	Video Encoding/ 3D Application Streaming
H1	High Disk Throughput	MapReduce-based workloads, distributed file systems such as HDFS and MapR-FS
T3	Lowest Cost, General Purpose	Web Servers/Small DBs
D2	Dense Storage	Fileservers/Data Warehousing/Hadoop
R5	Memory Optimized	Memory Intensive Apps/DBs
M5	General Purpose	Application Servers
C5	Compute Optimized	CPU Intensive Apps/DBs
P3	Graphics/General Purpose GPU	Machine Learning, Bit Coin Mining etc
X1	Memory Optimized	SAP HANA/Apache Spark etc
Z1D	High compute capacity and a high memory footprint.	Ideal for electronic design automation (EDA) and certain relational database workloads with high per-core licensing costs.
A1	Arm-based workloads	Scale-out workloads such as web servers
U-6tb1	Bare Metal	Bare metal capabilities that eliminate virtualization overhead

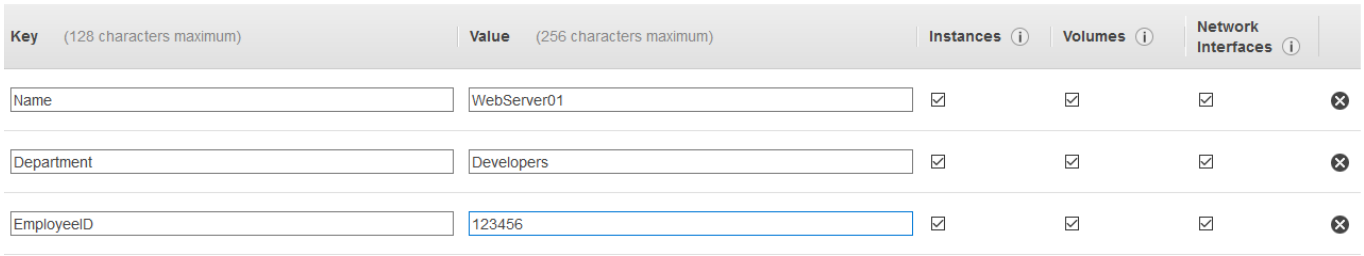
EC2 Instance Types - Mnemonic

- F → For FPGA
- I → For IOPS
- G → Graphics
- G → High Disk Throughput
- T → Cheap general purpose (think T2 micro)
- D → Density
- R → For ram
- M → Main choice for general purpose apps
- C → for compute
- P → graphics (think pics)
- X → Extreme memory
- Z → extreme memory and CPU
- A → arm-based workloads
- U → Bare Metal

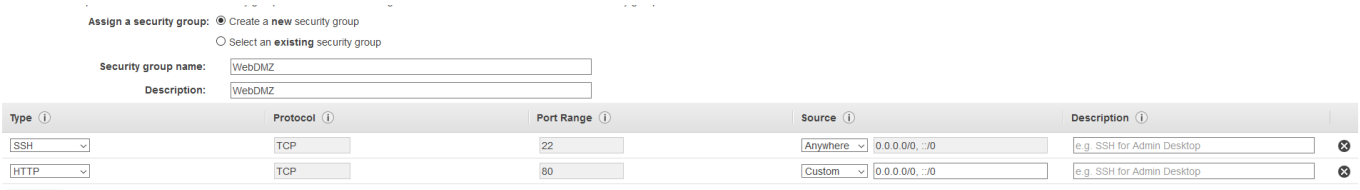
Launch EC Demo

Provision an EC2 and create a web server

AWS Console → EC2 → Launch Instance → Choose Virtual Machine (go with Amazon Linux 2 AMI) → Choose Free Tier (t2.micro) → Configure Instance → Only setting we want to change for now is "Protect against accidental termination" → Add Storage → For now we don't need to add additional volumes, keep it as default → Add Tags → We can add a few tags here, its optional



→ Configure Security Group → Change the group name, and description; can also specify more protocols to use with different ip ranges, but we can just make these changes →



→ Review → Launch → Create New Key Pair → give it a name → download key pair (will need this to connect to our ec2 instance) → launch instance

SSH to our EC2 instance

- One way is to AWS Console → EC2 → Select the instance → At the top tabs hit "Connect" → Double check the user name → connect → this will open a session directly in the browser with your ec2 instance
- Can also use terminal on Mac or Linux
  - you will need the key pair downloaded earlier for this method
  - create a new folder `mkdir SSH` move the `mv {keypairfilename}.pem SSH` to the SSH folder
  - `cd SSH` → `chmod 400 [keypairfilename].pem` → `ssh ec2-user@[public ipaddress of the instance] -i [keypairfilename].pem` → make sure port 22 isn't blocked
- Another method is to use plugins that are made for chrome
  - in google extensions store → secure shell app → download / install → open the secure shell app

- `copy the username from ec2`, `hostname is the public ip address`, in windows terminal, navigate to where the keypairfile was download and enter `ssh-keygen -y -f keypairfilename..pem > somename.pub`, then just rename the keypairfile `ren keypairfile.pem to keypair` remove the `.pem`, going back to chrome extension, for identity import the two files (keypair, and somename.pub), now we can connect

Using the EC2 Instance
first check for any updates → `yum update -y`

- now install apache → `yum install httpd -y`
- now we can change directory to the apache folder where we store our folders that are all accessible over port 80: `cd /var/www/html`
- now we need to create a new file `sudo nano index.html` → create a simple web page → `<html><body><h1>Hello From AWS</h1></body></html>` → Ctrl + X, accept the file name, hit enter
- Turn on the apache server → `service httpd start`
- to enable apache to start automatically if the ec2 restarts make sure to enable `chkconfig on`
- Now we should be able to view our web page in the browser

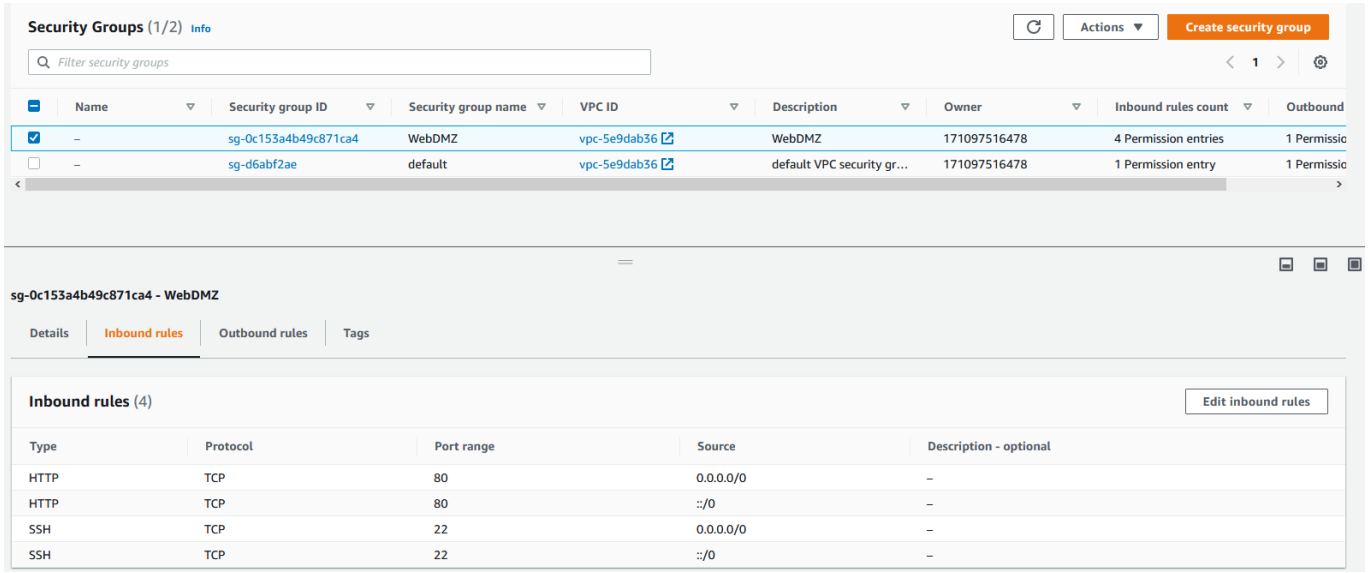
Exam Tips

- Termination protection is turned off by default, you must turn it on
- On an EBS-backed instance, the default action is for the root EBS volume to be deleted when the instance is terminated
- EBS Root volumes of the DEFAULT AMI's CAN be encrypted. You can also use a third party tool (such as bit locker etc) to encrypt the root volume, or this can be done when creating AMI's (lab to follow) in the AWS console or using the API
- Additional volumes can be encrypted

Security Groups - Demo

AWS Console → Ec2 → Instance → Select the instance → focus on security in the bottom pane → In the left hand panel → Under Network & Security → Security Groups → Here we can create/edit/delete security groups → select the security group

attached to the instance of ec2 → in the bottom panel highlight inbound rules → here we can edit our rules



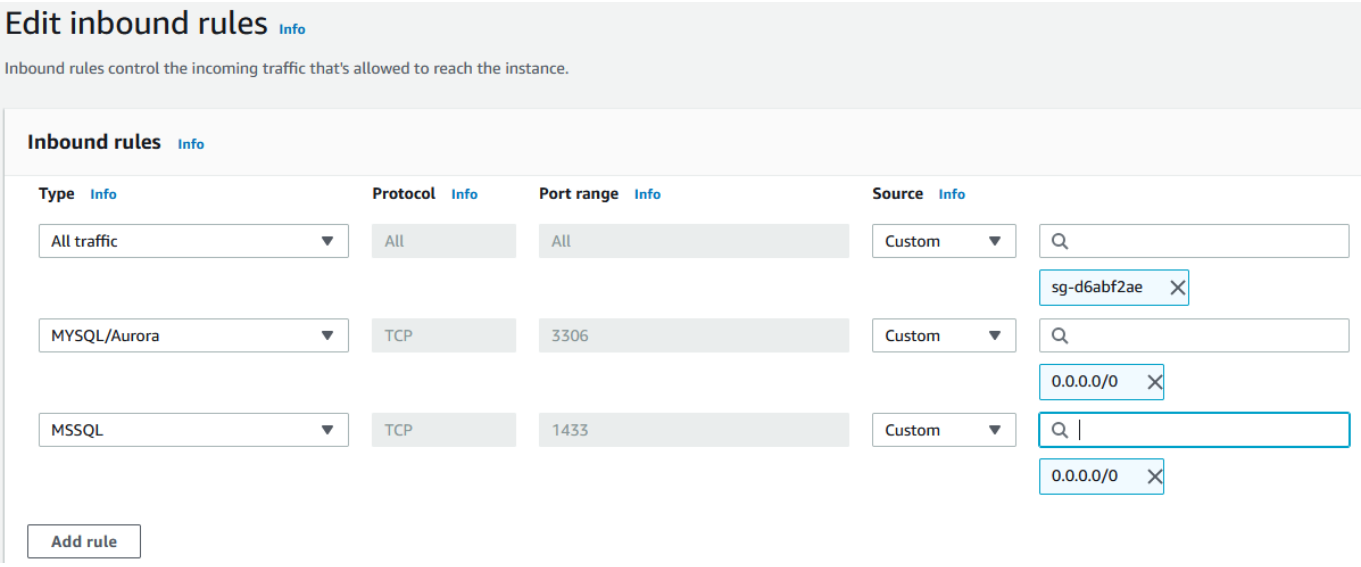
Rule Chanes takes effect immediately

Important note about outbound rules
security groups are stateful.
So essentially when you create an Inbound rule,
an Outbound rule is created automatically.
So if you allow HTTP in,
it is automatically allowed out as well.
If you allow RDP in or if you allow SSH
or you allow MySQL in,
it's automatically going to be allowed back out.

I can allow traffic over Oracle RDS,
or Redshift or SMTPS, whatever.
Whatever port it is, I can allow but
I can't actually go in and block an individual port.
I can't go in and say,
"Hey, don't allow any traffic across this."
There's no way of blacklisting a particular port
and likewise there's no way of blacklisting
a particular IP address.
You can't do that with security groups.
Where you can do that is with network access controllers
and again we going to look at that
in the VPC section.

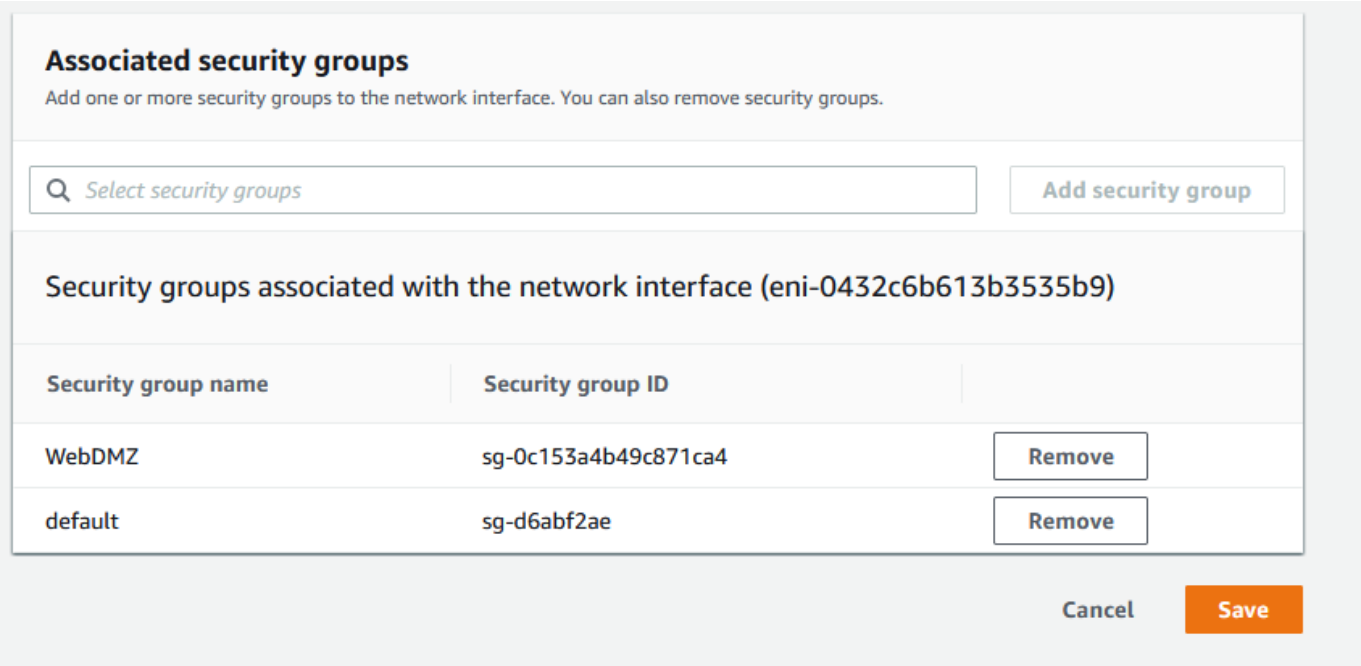
when you create a security group,
everything is blocked by default.
You have to go in and allow something.
So everything is blocked by default
but when you go in and allow HTTP or MySQL
then the traffic is allowed through.

We can edit our inobound rules



We can add additional security groups

EC2 → select the instance → actions → security → change security groups → search for the group → add security group → save



Exam Tips

- All inbound traffic is blocked by default
- All outbound traffic is allowed
- changes to security groups take effect immediately
- you can have any number of EC2 instances within a security group
- you can have multiple security groups attached to EC2 isntances
- Security groups are stateful
- if you create an inbound rule allowing traffic in, that traffic is automatically allowed back out again
- You cannot block specific ip addresses using security groups, instead use network access control lists
- you can specify allow rules, but not deny rules

EBS 101

What is EBS

Amazon Elastic Block Store (EBS) provides persistent block storage volumes for use with Amazon EC2 instances in the AWS Cloud. Each Amazon EBS volume is automatically replicated within its availability zone to protect you from component failure, offering high availablility and durability.

5 Different Types of EBS Storage

1. General Purpose (SSD)
2. Provisioned IOPS (SSD)
3. Throughput Optimised Hard Disk Drive
4. Cold Hard Disk Drive
5. Magnetic

Comparison

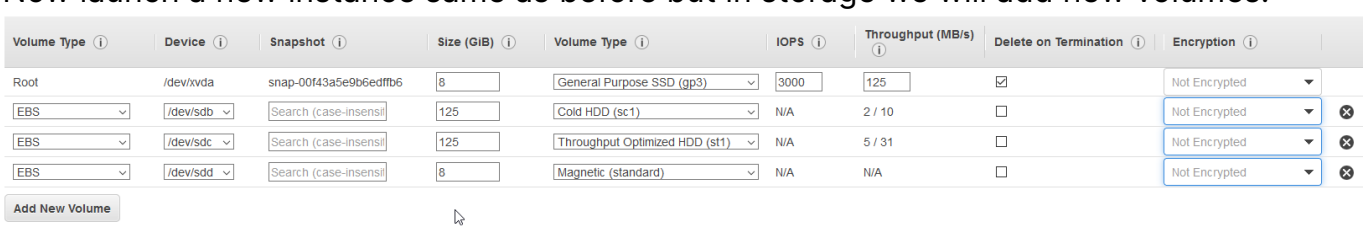
Solid-State Drives (SSD)			Hard disk Drives (HDD)		
Volume Type	**General Purpose SSD**	**Provisioned IOPS SSD**	**Throughput Optimized HDD**	**Cold HDD**	**EBS Magnetic**
Description	General purpose SSD volume that balances price and performance for a wide variety of transactional workloads	Highest-performance SSD volume designed for mission-critical applications	Low cost HDD volume designed for frequently accessed, throughput-intensive workloads	Lowest cost HDD volume designed for less frequently accessed workloads	Previous generation HDD
Use Cases	**Most Work Loads**	**Databases**	**Big Data & Data Warehouses**	**File Servers**	**Workloads where data is infrequently accessed**
API Name	gp2	io1	st1	sc1	Standard
Volume Size	1 GiB - 16 TiB	4 GiB - 16 TiB	500 GiB - 16 TiB	500 GiB - 16 TiB	1 GiB-1 TiB
Max. IOPS**/ Volume	16,000	64,000	500	250	40-200

EBS Volumes & and Snapshots - Demo

AWS Console ⟶ EC2 ⟶ Select running instance ⟶

- Where ever the EC2 instance is located (availability zone) the volumes (Elastic block store (EBS) -- located in the left hand panel) will also be located in the same availability zone
- When we terminate the ec2 instance we terminate the ebs instance as well

Now launch a new instance same as before but in storage we will add new volumes.



These will be our new EBS instances.
We don't need to add any tags, and for security groups we will use the same one as before "WebDMZ" security group

Now we have multiple EBS volumes.
We can easily increase storage on any of the volumes by going to Volumes ⟶ select the volume ⟶ action ⟶ modify volume ⟶ and increase storage

We can also modify other things on the volumes, we can change storage medium, storage size, etc.

If we terminate the EC2 instances, it won't terminate the new EBS volumes. They persist, when you terminate an EC2 instance by default the root storage will be terminated as well. But any additional storage attached to that EC2 instance will not be terminated.

- What happens when you want to move your ec2 instance or your ebs volumes
  - select your ebs volumes → actions → create snapshot → give it a name → create snapshop (it takes time to take effect) → select the snapshot → action → With snapshot we can create volumes, images, etc → select create image → provide it a name → can leave everything else as default, but you should take a note of the virtualization type → create || this created a new image which can be used to provision new ec2 instances in different availability zones
    - Select the image we created → launch → choose instance type → configure instance details → Configure INstance → Can switch the Availability Zone, its under "Subnet" → Storage → Tags → use existing security group → launch
    - Now we have successfully migrated data from one ebs in one ec2 availability zone to another ec2 instance in another availability zone
      - Take a snapshot → turn it into an AMI → Use the AMI to launch an EC2 in other availability zone
      - We can also copy the AMI into different regions
        - Select the AMI image → actions → copy AMI → select the destination → Copy AMI

Exam Tips

- Volumes exist on EBS. think of EBS as virtual hard disk
- Snapshots exists on S3. Think of snapshots as photograph of the disk
- Snapshots are point in time copies of Volumes
- Snapshots are incremental -- This means that only the blocks that have changed since your last snapshot are moved to S3
- If this is your first snap,shot it may take some time to create
- To create a snapshot for Amazon EBS volumes that serve as root devices, you should stop the instance before taking the snapshot
- However you can take a snap while the instance is running
- You can create AMI's from both Snapshots and Volumes
- You can change EBS volume sizes on the fly, including changing the size and storage type
- Volumes will always be in the same availability zone as the EC2 instance
- To move an EC2 volume from one AZ to another, take a snapshot of it, create an AMI from the snapshot and then use the AMI to launch the EC2 instance in a new AZ

- To move an EC2 volume from one region to another, take a snapshot of it, create an AMI from the snapshot and then copy the AMI from one region to the other. Then use the copied AMI to launch the new EC2 instance in the new region.

AMI Types (EBS vs Instance Store)

AMI's

You can select your AMI based on

- Region (see Regions and Availability Zones)
- Operating system
- Architecutre (32-bit or 64-bit)
- Launch Permissions
- Storage for the Root Device (Root Device Volume)
    - Instance Store (EPHEMERAL STORAGE)
    - EBS Backed Volumes

EBS vs Instance Store Volumes

- All AMI's are categorized as either backed by Amazon EBS or backed by instance store
- For EBS Volumes: The root device for an instance launched from the AMI is an Amazon EBS volume created from an Amazon EBS snapshot
- For Instance Store Volumes: The root device for an instance launched from the AMI is an instance store volume created from a template stored in Amazon S3

AWS Console → EC2 → Launch → Keeping everything as default → switch the security group to use our WebDMZ group

And that launches our Amazon Linux AMI

Now we can also launch another EC2 instance but this time:
AWS Console → EC2 → Launch → On the left hand pane, select Community AMIs instead of Quick start → We can sort by OS, architecture or root device type, for now select instance store → choose the default from the top → You will notice a lot of instance types are blocked,

⊘	General purpose	m4.4xlarge	16	64	EBS only	Yes	High	Yes
⊘	General purpose	m4.10xlarge	40	160	EBS only	Yes	10 Gigabit	Yes
⊘	General purpose	m4.16xlarge	64	256	EBS only	Yes	25 Gigabit	Yes
☐	General purpose	m3.medium	1	3.75	1 x 4 (SSD)	-	Moderate	-
☐	General purpose	m3.large	2	7.5	1 x 32 (SSD)	-	Moderate	-
☐	General purpose	m3.xlarge	4	15	2 x 40 (SSD)	Yes	High	-
☐	General purpose	m3.2xlarge	8	30	2 x 80 (SSD)	Yes	High	-

Choose one that is available and works best → Leave the configuration as default →

Step 4: Add Storage

Your instance will be launched with the following storage device settings. You can attach additional instance store volumes to your instance. You can also attach additional EBS volumes after launching an instance, but not instance store volumes. Learn more about storage options in Amazon EC2.

Volume Type	Device	Snapshot	Size (GiB)	Volume Type	IOPS	Throughput (MB/s)	Delete on Termination
Instance Store 0	/dev/sdb	N/A	4	SSD	N/A	N/A	N/A

Notice the volume type is instance store: we can attach additional instance store volumes to the instance, also attach additional EBS volumes after launching an instance, but not instance store volumes.

→ Add tags → Configure security group, WebDMZ → launch

Now we have two EC2 instances: one with an instance store and one with a normal ebs backed.

- if we got o volumes, we don't see the instance store volume, only the ebs backed volume.
- EBS backed volumes are persistent storage: if the EC2 instance is terminated the volumes will persist
- The Instance store EC2 can only be stopped or terminated; this is because it is instance stored backed: meaning you can't start the instance store ec2 instance in another hyper visor; if the hypervisor has any issues you might lose everything. where as the ebs backed ec2 can be stopped and started in another hypervisor, simply by turning it off and restarting it again.

Exam Tips

- Instance store volumes are sometimes called Ephemeral Storage
- Instance store volumes cannot be stopped. If the underlying host fails, you will lose your data.
- EBS backed instances can be stopped. You will not lose the data on this instance if it is stopped
- You can reboot both, you will not lose your data
- By default, both ROOT volumes will be deleted on termination. However, with EBS volumes, you can tell AWS to keep the root device volume

ENI vs ENA vs EFA

ENI

Elastic Network Interface -- essentially a virtual network cark

- A primary private IPv4 address from the IPv4 address range of your VPC
- One or more secondary private IPv4 addresses from the IPv4 address range of your VPC
- One Elastic IP address (IPv4) per private IPv4 address
- One public IPv4 address
- One or more IPv6 addresses
- One or more security groups
- A MAC address
- A source/destination check flag
- A description

Scenarios for Network Interfaces

- Create a management network
- Use network and security appliances in your VPC
- Create dual-homed instances with workloads/roles on distinct subnets
- Create a low-budget, high-availability solution

ENA

Enhanced Networking. Uses single root I/O virtualization (SR-IOV) to provide high-performance networking capabilities on supported instance types

- It uses single root I/O virtualization (SR-IOV) to provide high-performance networking capabilities on supported instance types. SR-IOV is a method of device virtualization that provides higher I/O performance and lower CPU utilization when compared to traditional virtualizaed network interfaces
- Enhanced networking provides higher bandwidth, higher packet per second (PPS) performance, and consistently lower inter-instance latencies. There is no additional charge for using enhanced networking
- Use where you want good network performance

Depending on your instance type, enhanced networking can be enabled using:

- Elastic Network Adapter (ENA), which supports network speeds of up to 100 Gbpsd for supported instance types
- OR
- Intel 82599 Virtual Function (VF) interface, which supports network speeds of up to 10Gbps for supported instance types. This is typically used on older isntances

In any scenario question, you probably want to choose ENA over VF if given the option

EFA

A network device that you can attach to your Amazon EC2 instance to accelerate High Performance Computing (HPC) and machine learning applications

What is an Elastic Fabric Adapter

- An Elastic Fabric Adapter (EFA) is a network device that you can attach to your Amazon Ec2 instance to accelerate High Performance Computing (HPC) and machine learning applications
- EFA provides lower and more consistent latency and higher throughput than the TCP transport traditionally used in cloud-based HPC systems
- EFA can use OS-bypass enables HPC and machine learning applications to bypass the operating system kernel and to communicate directly with the EFA device. IT makes it a lot faster with a lot lower latency. Not supported with Windows currently, only Linux.

Exam Tips

In the exam you will be given different scenarios and you will be asked to choose whether you should use an ENI, EN or EFA

- ENI
    - For basic networking. Perhaps you need a separate management network to your production network or a separate logging network and you need to do this at a low cost. In this scenario use multiple ENIs for each network
- Enhanced Network
    - For when you need speeds netweek 10Gbps and 100Gbps. Anywhere you need reliable, high throughput
- Elastic Fabric Adaptor
    - For when you need to accelerate high performance computing (HPC) and machine learning applications or if you need to do an OS by-pass. If you see a scenario question mentioning HPC or ML and asking what network adaptor you want, choose EFA

Encrypted Root Device Volumes and Snapshots - Demo

Creating an Encrypted EC2 instance
AWS Console → EC2 → Launch an instance → Default is fine → storage, we can encrypt storage here

Editing an EC2 instance to be encrypted
AWS Console → EC2 → Volumes → Select volume → actions → create snapshot --

> name it → create snapshot → Now go to Snapshots → select the snapshot → actions → copy → select "Encrypt this snapshot" → choose a standard aws/ebs key, provide it a description → copy → Select the Encrypted Copy → Actions → Create Image → Name, description, etc → create (will create an encrypted AMI) → We can use this AMI to launch encrypted EC2 instance → go to AMI → selected the AMI image → launch → default ec2 instance , and you will notice in storage section the device is already encrypted

Exam Tips

- Snapshots of encrypted volumes are encrypted automatically
- Volumes restored from encrypted snapshots are encrypted automatically
- You can share snapshots, but only if they are unencrypted
- These snapshots can be shared with other AWS accounts or made public
- You can now encrypt root device volumes upon creation of the EC2 instance
- For uncrypted devices
    - Create a snapshot of the unencrypted root device volume
    - Create a copy of the snapshot and select the encrypt option
    - Create an AMI from the encrypted snapshot
    - Use that AMI to launch a new encrypted instances

Spot Instances & Spot Fleets

What is an EC2 Spot Instance?

Amazon EC2 Spot Instances let you take advantage of unused EC2 capacity in the AWS Cloud. Spot Instances are available at up to a 90% discount compared to On-Demand prices. You can use Spot Instances for various stateless, fault-tolerant, or flexible applications, such as big data, containerized workloads, CI/CD, web servers, high-performance computing (HPC), and other test and development workloads

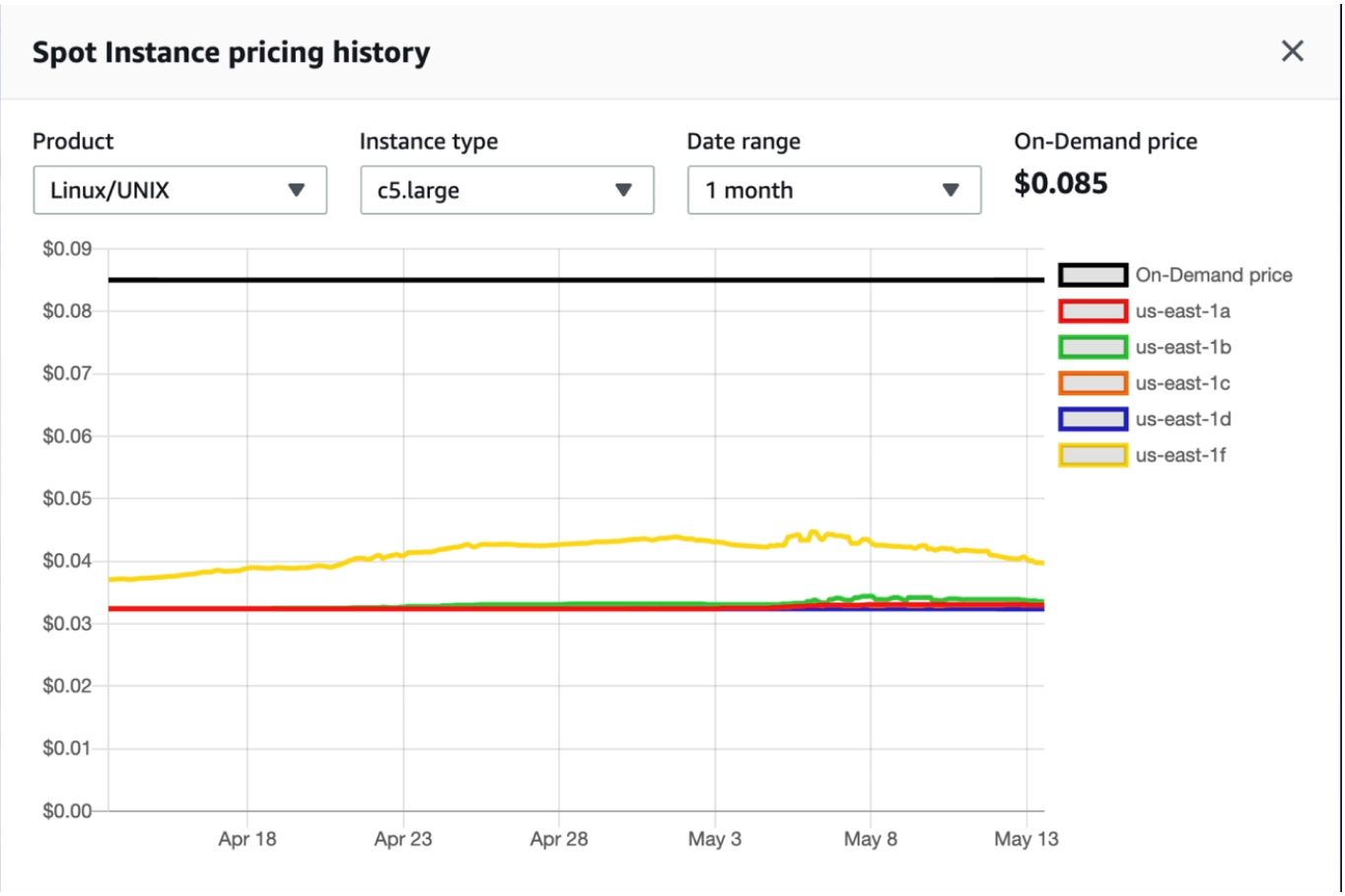
Best for: flexible applications

Spot Prices
To use Spot Instances, you must first decide on your maximum Spot price. The instance will be provisioned so long as the Spot price is BELOW your maximum Spot price.

- The hourly Spot price varies depending on capacity and region
- If the Spot price goes above your maximum, you have two minutes to choose whether to stop or terminate your instance.

Spot Blocks

- You may also use a Spot Block to stop your Spot Instances from being terminated even if the Spot price goes over your max Spot price. You can set Spot blocks for between one to six hours currently.



Use Cases

Spot Instances are useful for the following tasks:

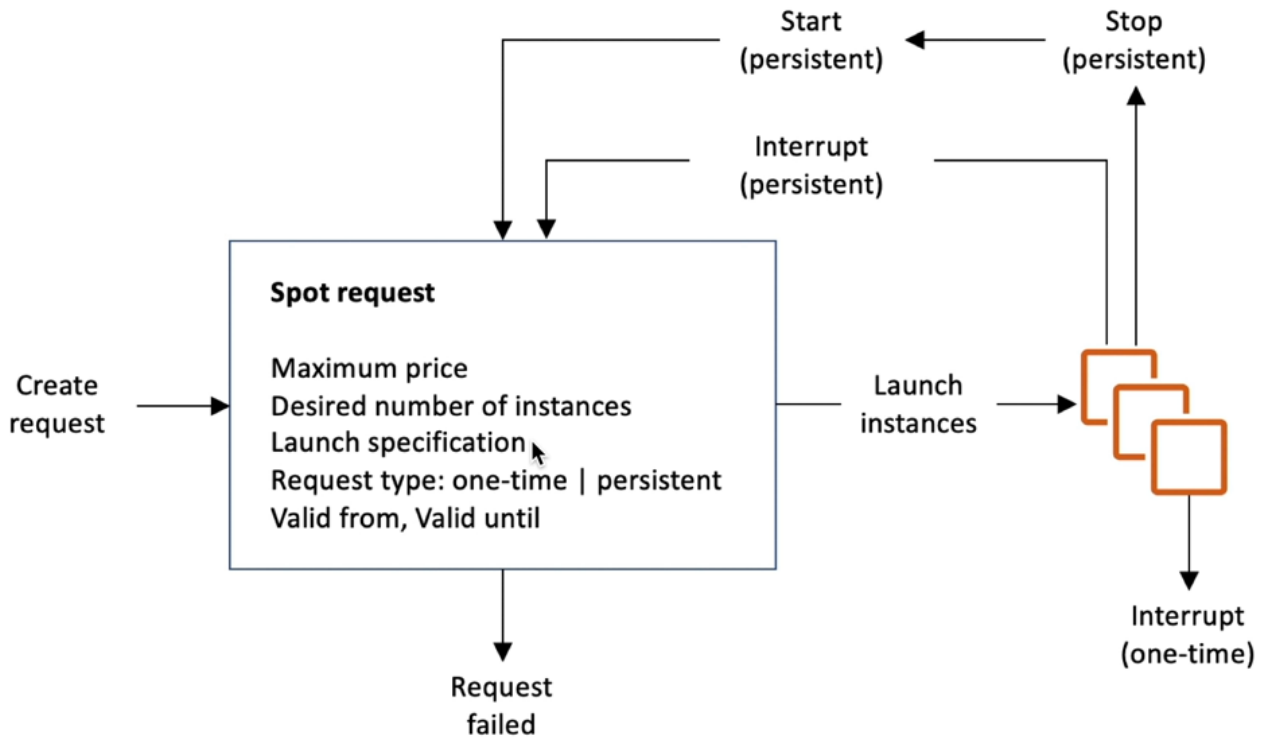
- Big Data and analytics
- Containerized workloads
- CI/CD and testing
- Web services
- Image and media rendering
- High-performance computing

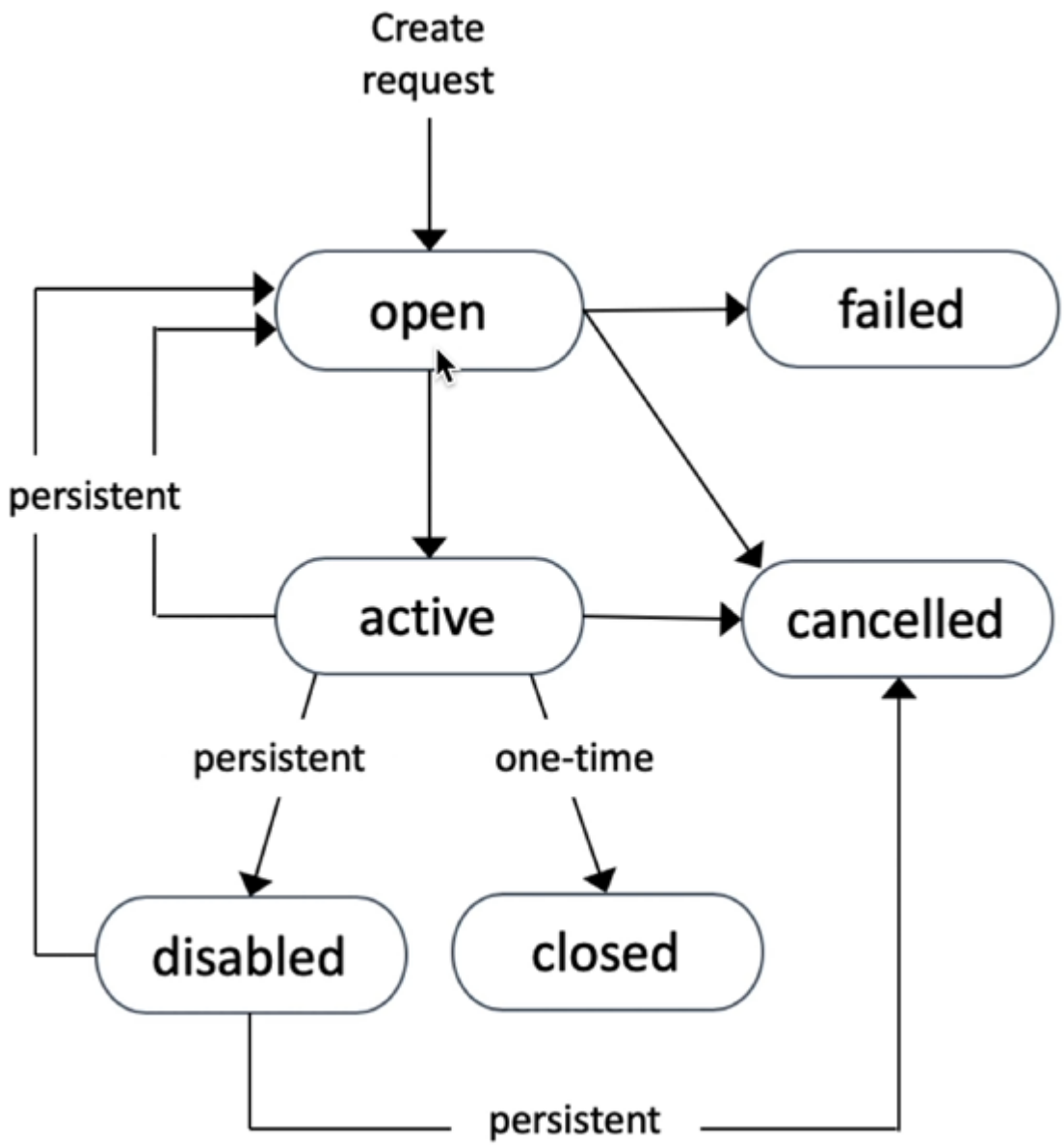
NOT GOOD FOR

- Persistent workloads
- Critical jobs
- Databases

Terminate Spot Instances

How to terminate Spot Instances





Spot Fleets

A Spot Fleet is a collection of Spot Instances and, optionally, On-Demand Instances

The Spot Fleet attempts to launch the number of Spot Instances and On-Demand Isntances to meet the target capacity you specified in the Spot Fleet request. The request for Spot Instances is fulfilled if there is available capacity and the maximum price you spcified in the request exceeds the current Spot price. The Spot Fleet also attempts to maintain it target capacity fleet if your Spot Instances are interrupted

Launch Pools

Spot Fleets will try and match the target capacity with your price restraints

1. Set up different launch pools. Define things like EC2 instance type, operating system, and Availability Zone.

2. You can have multiple pools, and the fleet will choose the best way to implement depending on the strategy you define
3. Spot fleets will stop launching instances once you reach your price threshold or capacity desire

Strategies

You can have the following strategies with Spot Fleets

- capacityOptimized
  - The spot instances come from the pool with optimal capacity for the number of instances launching
- diversified
  - The spot instances are distributed across all pools
- lowestPrice
  - The spot instances come from the pool with the lowest price. This is the default strategy
- InstancePoolsToUseCount
  - The spot instances are distributed across the number of spot instance pools you specify. This parameter is valid only when used in combination with lowestPrice

Exam Tips

- Spot Instances save up to 90% of the cost of on-demand instances
- useful for any type of computing where you dno't need persistent storage
- you can block spot instances from terminating by using spot block
- a spot fleet is a collectiong of spot instance and, optionally, on-demand instances

EC2 Hibernate

We know we can stop and terminate EC2 instances. If we stop the instance, the data is kept on the disk (with EBS) and will remain on the disk until the EC2 instance is started. If the instance is terminated, then by default the root device volume will also be terminated

When we start our EC2 instance, the following happens

- Operating system boots up
- use data script is run (bootstrap scripts)
- Applications Start (can take some time)

EC2 Hibernate

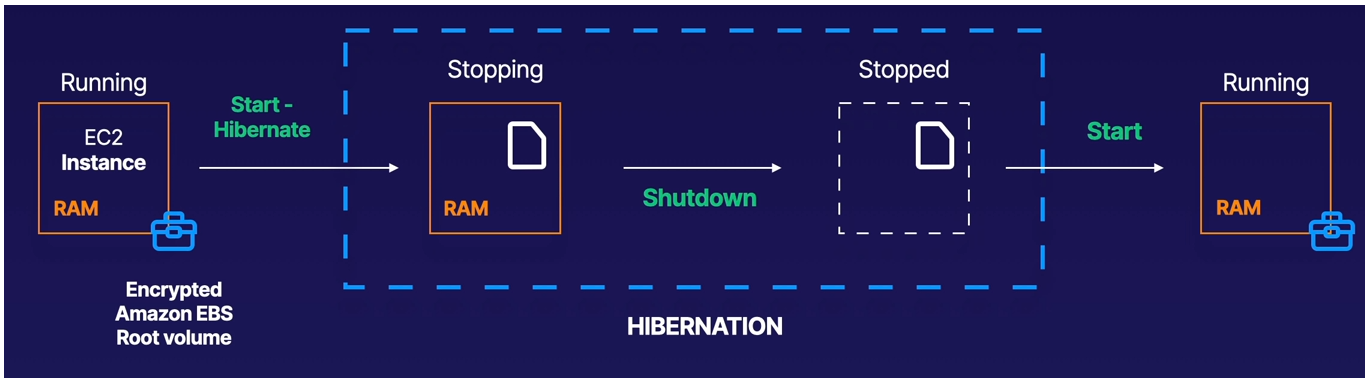
When you hibernate an EC2 instance, the operating system is told to perform hibernation

(suspend-to-disk). Hibernation saves the contents from the instance memory (RAM) to your Amazon EBS root volume. We persist the instance's Amazon EBS root volume and any attached Amazon EBS data volumes

- saves the ram to the EBS root volume

When you start your instance out of hibernation:

- The Amazon EBS root volume is restored to its previous state
- The RAM contents are reloaded
- The processes that were previously running on the instance are resumed\
- Previously attached data volumes are reattached and the instance retains its intsnace ID



Starting your EC2 Instance with EC2 Hibernate

With EC2 hibernate, the instance boots much faster. THe operating system does not need to reboot because the in-memory state (RAM) is preserved. This is useful for:

1. Long-running processes
2. Services that take time to initialize

AWS Console → EC2 → Launch an instance → Amazon linux 2 AMI → "Enable hibernation as an additional stop behavior" → Next → Root device must be encrypted if we use hibernation, increase storage to atleast 25GB → Next → Use WebDMZ → Launch

Now we have the option of hibernate when we want to stop our EC2 instance.

Exam Tips

- EC2 hibernate preserves the in-memory RAM on persistent Storage (EBS)
- Much faster to boot up because you do not need to reload the operating system
- instance RAM must be less than 150GB
- Instance families include C3, C4, C5, M3, M4, M5, R3, R4, and R5

- Available for windows, Amazon Linux 2 AMI, and Ubuntu
- Instances can't be hibernated for more than 60 days
- Available for on-demand instances and reserved instances

CloudWatch 101

What is CloudWatch

Amazon Cloud is a monitoring service to monitor your AWS resources, as well as the applications that you run on AWS.

Monitors performance

- Compute
    - EC2 instances
    - Autoscaling Groups
    - Elastic Load Balancers
    - Route53 Health Checks
- Storage & Content Delivery
    - EBS Volumes
    - Sotrage Gateways
    - CloudFront

Host level metrics consist of:

- CPU
- Network
- Disk
- Status Check

What is AWS Cloud Trail

AWS CloudTrail increases visibility into your user and resource activity by recording AWS Management Console actions and API calls. You can identify which users and accounts called AWS, the source IP address from which the calls were made, and when the calls occurred.

- Basically acts like a security camera and records what is happening

- CloudWatch ⟶ Monitors performance

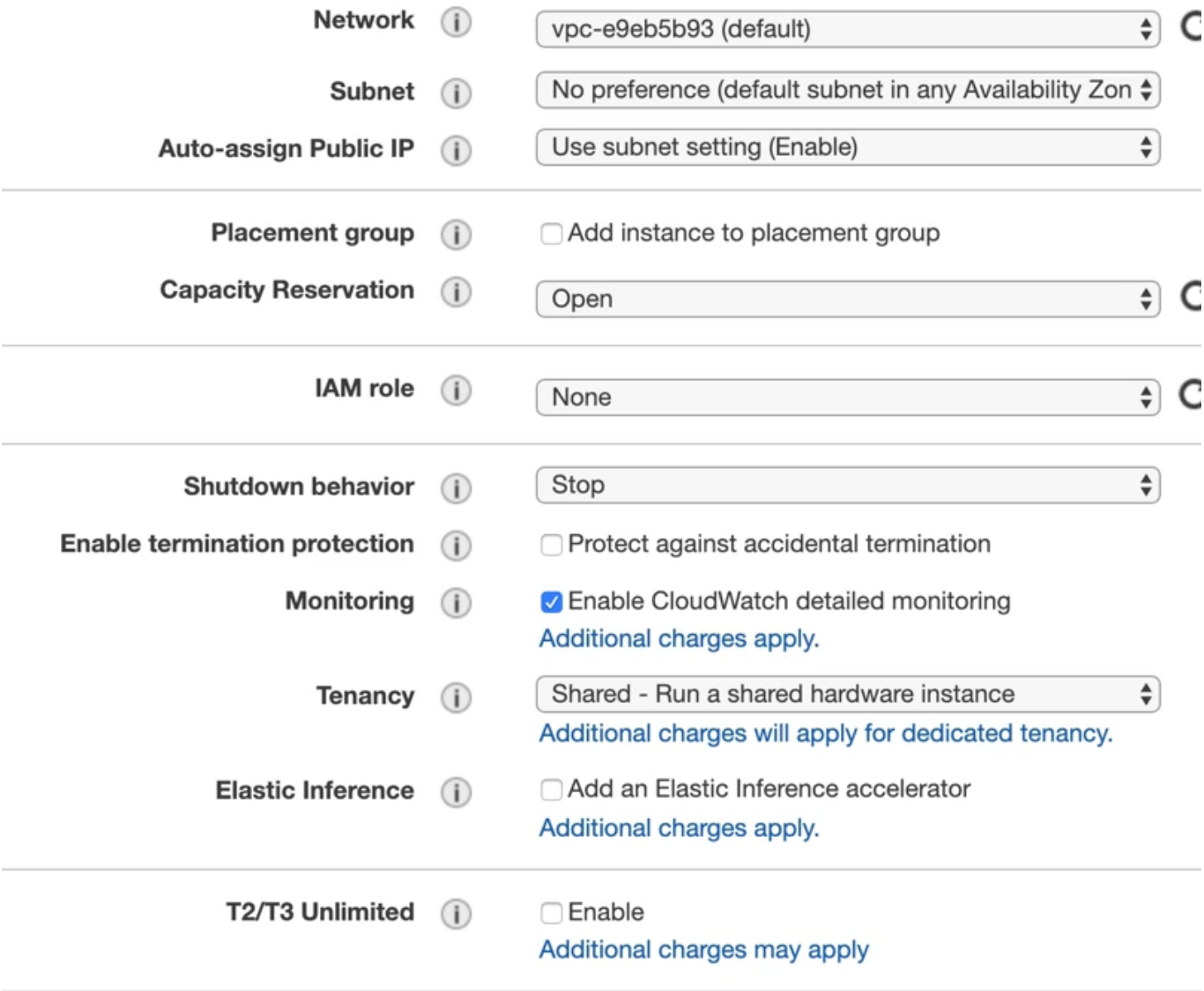
- CloudTrail ⟶ monitors API calls in the AWS platform

Exam Tips

- Remember
  - CloudWatch is used for monitoring performance
  - CloudWatch can monitor most of AWS as well as your applications that run on AWS
  - CloudWatch with EC2 will monitor events every 5 minutes by default
  - you can have 1 minute intervals by turning on detailed monitoring
  - You can create CloudWatch alarms which trigger notifications
  - CloudWatch is all about performance. CloudTrails is all about auditing

CloudWatch - Demo

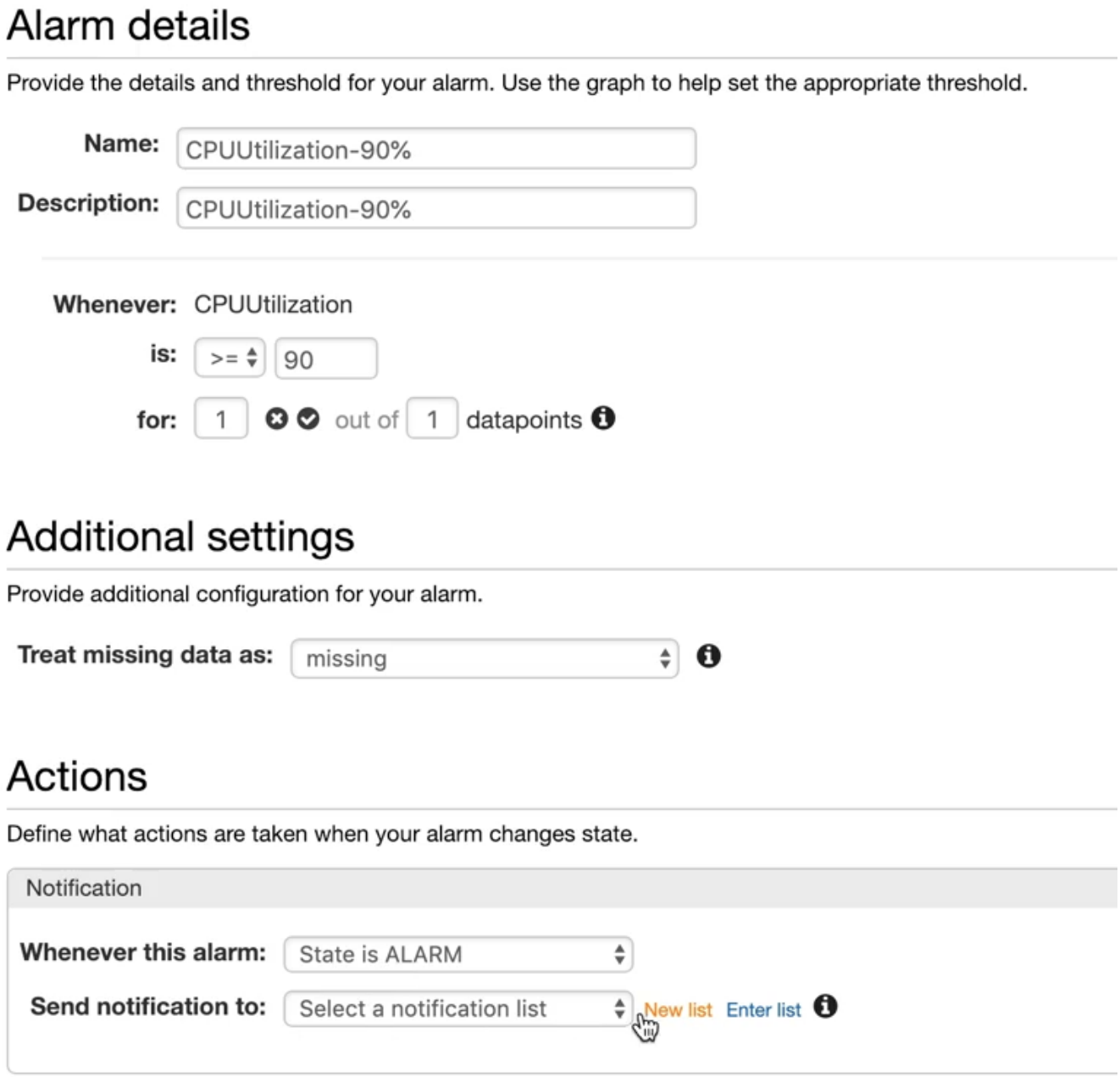
AWS Console → EC2 → Launch Instance → Amazon Linux 2 AMI → Enable Monitoring "Enable CloudWatch detailed monitoring" →



→ can leave everything else as default → launch

Set up alert for when our ec2 instance cpu utilization is above a certain percentage

AWS Console → CloudWatch → Alarms → Create Alarm → Select Metric → EC2 → Perinstance metric → choose the right instance (remember the ec2 instance id) metric -> select "CPUUtilization" → provide a name, and description, define the percentage for the cpu to set the alarm



Define what to do when alarm is set, we can send an email when alarm is triggered. → Create Alarm → confirm email address

Exam Tips

- Standard Monitoring = 5 mintes
- Detailed Monitoring = 1 minute

- What can you do with CloudWatch
  - Dashboards - Creates awesome dashboards to see what is happening with your AWS environment
  - Alarms - allows you to set Alarms that notify you when particular thresholds are hit
  - Events - CloudWatch Events helps you to respond to state changes in your AWS resources
  - Logs - CloudWatch Logs helps you to aggregate, monitor, and store logs
  - CloudTrail → monitors API calls in the AWS platform
  - CloudWatch → performance

AWS Command Line (CLI) - Demo

AWS Console → IAM → Users → name it, allow programmatic access → AdministratorAccess → Create User → Download the User CSV, contains the access key and secret access key

EC2 → Laucnh an instance, with everything as default → create a security group → create a new key pair → download the keypair → launch the instance

SSH into the EC2 instance through command line / terminal on local desktop

to use AWS Command Line all we need to do is type `aws` and then the service we want. `aws s3 ls` show what s3 buckets we have, for example.

- `aws configure` → configure our AWS CLI to use the user we created earlier
  - `Access Key ID` → the user access key id we created
  - `Access Key` → user access key we created
  - `Default region name` → us-east-1
  - `Deafult output format` → none, text, json
- `aws s3 mb s3://testbucket` → mb = make bucket, and we are calling it "testbucket", it will create the bucket as long as it's a unique name globally
- `cd ~` → `cd .aws` → `ls` → this contains your config and credentials files
  - the secret access key id and secret access key are stored in the config file

Exam Tips

- You can interact with AWS from anywhere in the world just by using the command line

- You will need to set up access in IAM
- Basic commands will be useful to know

IAM Roles - Demo

- Roles enables us to interact with AWS platform without having to pass our EC2 instances, user access key ids and access key

AWS Console → IAM → Create Role → you can choose EC2 → AdministractorAccess for this role → give it a name → Create role

now go to EC2 → select the running instance → Actions → Security → Modify IAM Role → Attach the Admin access role created earlier → now SSH into EC2 → we can view `aws s3 ls` for example without configuring our AWS for the EC2 instance

Exam Tips

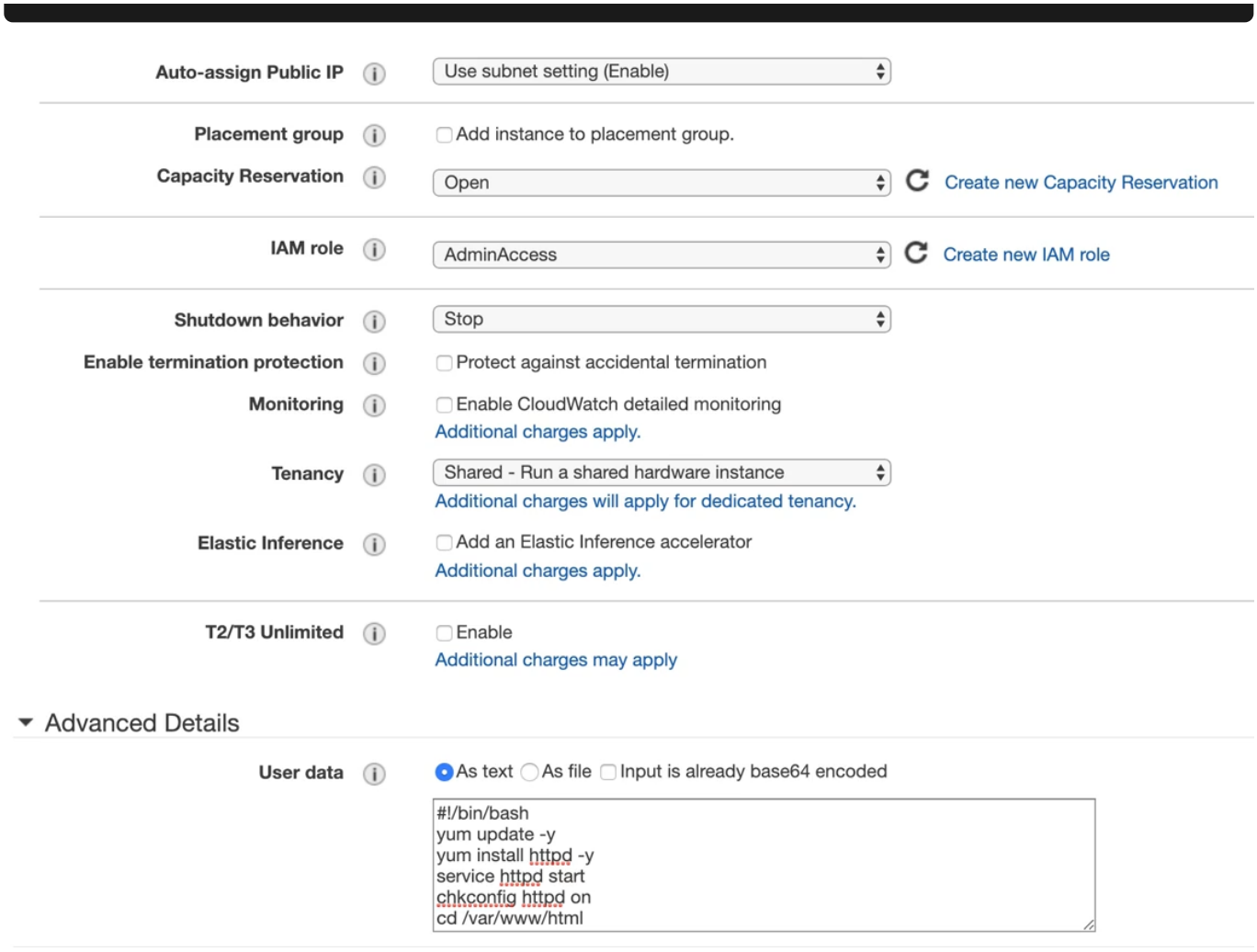
- Roles are more secure than storing your access key and secret access key on individual EC2 instances
- Roles are easier to manage
- Roles can be assigned to an EC2 instance after it is created using both the console and command line
- Roles are universal -- can use them in any region

Bootstrap Scripts (Bash Scripting) - Demo

AWS Console → EC2 → launch a new Amazon linux 2 AMI → attach the admin access IAM role → and we can define our script at the bottom, under advanced details

Install an Apache web server

```
#!/bin/bash
yum update -y
yum install httpd -y
service httpd start
chkconfig httpd on
cd /var/www/html
echo "<html><h1>Hello From AWS</h1></html>" > index.html
aws s3 mb s3://[uniquebucketname]
aws s3 cp index.html s3://[uniquebucketname]
```



Instance Metadata - Demo

SSH into EC2 instance → elevate privilege to root `sudo su` → `curl http://169.254.169.254/latest/user-data` here we can view the user-data passed to the EC2 instance, for example the script we created for the EC2 instance earlier → We can output this result to a file for example `curl http://169.254.169.254/latest/user-data > bootstrap.txt`, then we can view the file `cat bootstrap.txt` → We can also get the data on the individual EC2 instance `curl http://169.254.169.254/latest/meta-data`, additional arguments to the end of the url for any other output we wish to view

For example getting public ip address `curl http://169.254.169.254/latest/user-data/public-ipv4` will show the public ip address, and we can output this to a file as well `curl http://169.254.169.254/latest/user-data/public-ipv4 > myip.txt` and view the file as well `cat myip.txt`

Exam Tips

- Used to get information about an instance (such as public ip)
- `curl http://169.254.169.254/latest/meta-data`
- `curl http://169.254.169.254/latest/user-data`

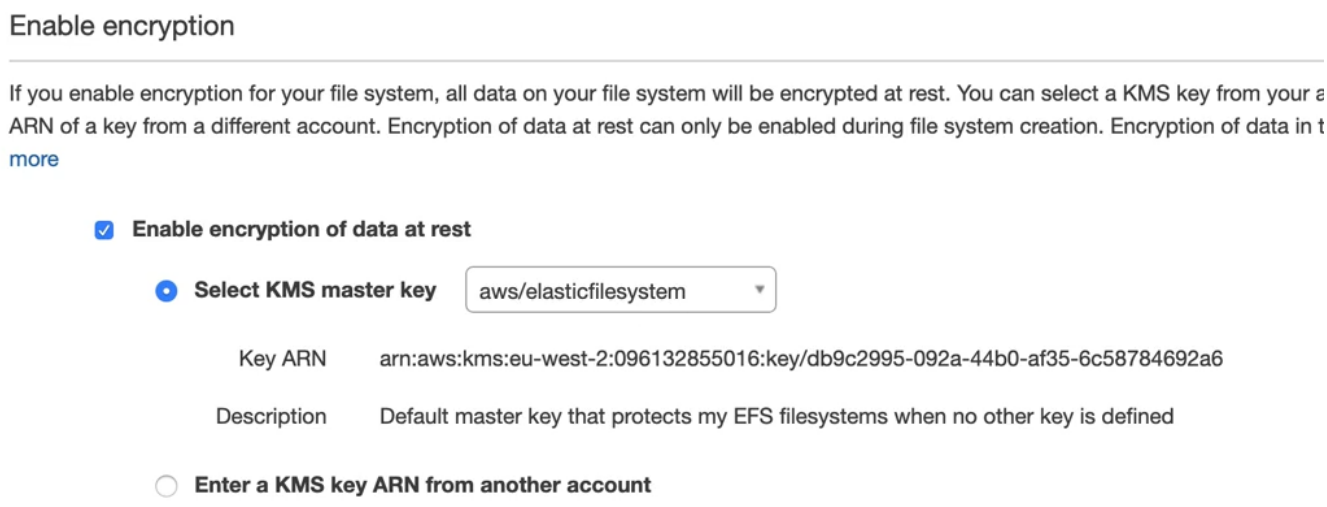
EFS - Demo

What is EFS?

Amazon Elastic File System (Amazon EFS) is a file storage service for Amazon Elastic Compute Cloud (Amazon EC2) instances. Amazon EFS is easy to use and provides a simple interface that allows you to create and configure file systems quickly and easily. With Amazon EFS, storage capcity is elastic, growing and shrinking automatically as you add and remove files, so your applications have the storage they need, when they need it.

With EBS volumes you can only mount it to one EC2 instance; you can't have two EC2 instances sharing one EBS volume. But you can have two EC2 instances sharing an EFS volume.

AWS Console → EFS (under storage) → Create File System → select AZ's you want to spread it across, and by default it uses the "default security group" → Next → Turn on encryption → next → and it'll create our EFS volume



Now create two EC2 instance with the following bootstrap script

```
#!/bin/bash
yum update -y
yum install httpd -y
service httpd start
```

```
chkconfig httpd on
yum install amazon-efs-utils -y
```

- updating the ec2 instance
- installing apache web server
- starting the apache server
- making sure the apache server is back up, if the ec2 instance restarts
- installing amazon efs utility tools, that will allow us to mount it later

Everything else can be default, with WebDMZ security group, and download the access key pair before launch

EC2 → Security Groups → select default → inobound rules → add rule → NFS → and for cidr, we can use our security group `sg-...` for WebDMZ

EC2 → select a running instance → ssh into it, `ssh ec2-user@[ec2 public ipaddress] -i [ec2 key pair].kem` → switch to root user `sudo su` → make sure the html folder exists, lets us know apache was installed `cd /var/www/html` → switch to `www` folder

Second EC2 → SSh into it in the same way in a different terminal → elevate user → make sure apache is installed → switch to `www` folder

EFS → select the EFS volume → scroll down and select "Amazon EC2 mount intructions from local VPC"

Amazon EC2 mount instructions (from local VPC)

To set up your EC2 instance:

- Using the Amazon EC2 console, associate your EC2 instance with a VPC security group that enables access to your mount target. For example, if you assigned the "default" security group to your mount target, you should assign the "default" security group to your EC2 instance. Learn more
- Open an SSH client and connect to your EC2 instance. (Find out how to connect.)
- If you're using an Amazon Linux EC2 instance, install the EFS mount helper with the following command:

  `sudo yum install -y amazon-efs-utils`

  You can still use the EFS mount helper if you're not using an Amazon Linux instance. Learn more

  If you're not using the EFS mount helper, install the NFS client on your EC2 instance:
  - On a Red Hat Enterprise Linux or SUSE Linux instance, use this command:

    `sudo yum install -y nfs-utils`
  - On an Ubuntu instance, use this command:

    `sudo apt-get install nfs-common`

Mounting your file system

1. Open an SSH client and connect to your EC2 instance. (Find out how to connect).

→ We've already installed the `amazon-efs-utils` through our bootstrap script so no need to run the command; `sudo/yum install -y amazon-efs-utils`

→ follow the other instructions →

`sudo mount it efs fs-9816b629:/ efs` means no encryption in transit, we don't want this. Because we want encryption in transit and in rest
`sudo mount -t efs -o tls fs-9816b269:/ efs` this will enable encryption in transit
only difference for us is we dont want to set this to the efs directory, we want to set it to the apache html directory
so change it to: `sudo mount -t efs -o tls fs-9816b269:/ /var/www/html` → mounts the efs to the hmtl directory
Do this for both EC2 instances through the CLI

now switch to html directory and create a simple webpage file
`cd html` → `echo "<html><h1>Hello FROM AWS EFS</h1></html>" > index.html` → view the file → `cat index.html`

Switch to the other EC2 insance and change directory to html
`cd html` → view all files → `ls` → you can see there's a html file there → view the file → `cat index.html` → this should be the same file created from the first EC2 instance; but since the volume is being shared between the two instances they both can access it.

Exam Tips

- Supports the network file system version 4 (NFSv4) protocol
- you only pay for the storage you use (no pre-provisioning required)
- can scale up to the petabytes
- can support thousands of concurrent nfs connections
- data is stored across multiple AZ's witihn a region
- Read After Write Consistency

Amazon FSx for Windows and Lustre

For Windows

Amazon FSx for Windows File Server provides a fully managed natice microsoft windows file system so you can easily move your windows-based applications that require file storage to AWS. Amazon FSx is built on Windows Servers.

How is Windows FSx different to EFS

Windows FSx

- A managed Windows Server that runs Windows Server Message Block (SMB)-based file services
- Designed for Windows and Windows application
- Supports AD users, access control lists, groups and security policies, along with Distributed File System (DFS) namespaces and replication

EFS

- A managed NAS filer for EC2 instances based on Network File System (NFS) version 4
- One of the first network file sharing protocols native to Unix and Linux

Lustre

Amazon FSx for Lustre is fully managed file system that is optimized for compute-intensive workloads, such as high-performance computing, machine learning, media data processing workflows, and electronic design automation (EDA)

With Amazon FSx, you can launch and run a Lustre file system that can process massive data sets at up to hundreds of gigabytes per second of throughput, millions of IOPS, and sub-millisecond latencies

How is Lustre FSx different to EFS

Lustre FSx

- Designed specifically for fast processing of workloads such as machine learning, high performance computing (HPC), video processing, financial modeling, and electronic design automation (EDA)
- Lets you launch and run a file system that provides sub-millisecond access to your data and allows you to read and write data at speeds of up to hundreds of gigabytes per second of throughput and millions of IOPS

EFS

- A managed NAS filer for EC2 instances based on Network File System (NFS) version 4
- One of the first network file sharing protocols native to Unix and Linux

Exam Tips

In the exam there will be different scenarios and asked to choose whether you should use an EFS, FSx for Windows or FSx for Lustre

- EFS → When you need distributed, highly resilient storage for Linux instances and Linux based applications
- Amazon FSx for Windows → When you need centralised storage for Windows-based applications such as Sharepoint, Microsoft SQL Server, Workspaces, IIS Web Server or any other native Microsoft Application
- Amazon FSx for Lustre → When you need high-speed, high-capacity distributed storage. This will be for applications that do high performance compute (HPC), financial modelling etc. Remember that FSx for Lustre can sotre data directly on S3

EC2 Placement Groups

Three Types of Placement Groups:

1. Clustered Placement Group
2. Spread Placement Group
3. Partitioned

Clustered Placement Group

A cluster placement group is a grouping of instances within a single AZ. Placement groups are recommended for applications that need low network latency, high network throughput, or both.

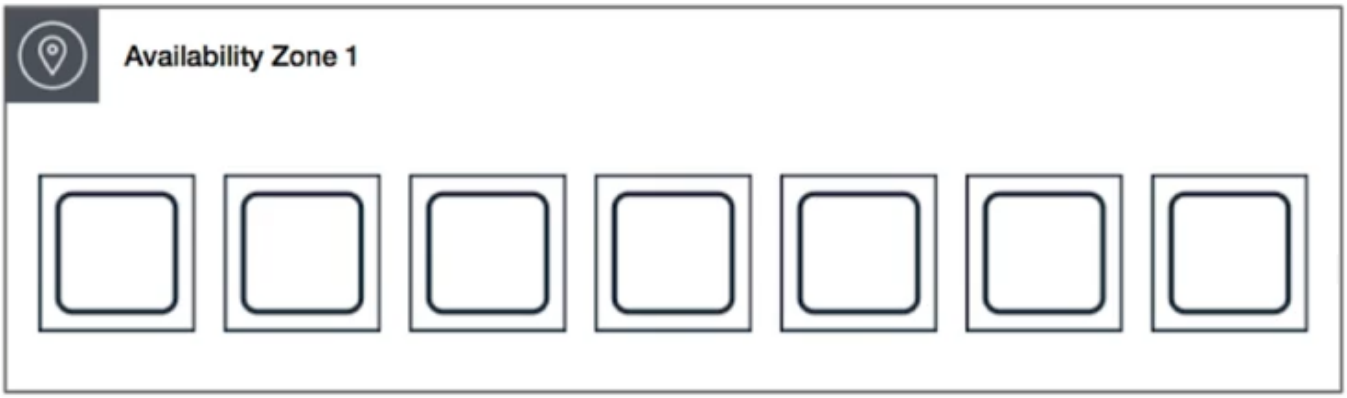
Only certain instances can be launched in to a Clustered Placement Groups.

Spread Placement Group

A spread placement group is a group of instances that are each placed on distinct underlying hardware.

Spread placement groups are recommended for applications that have a small number of critical instances that should be kept separate from each other.

THINK OF INIDIVIDUAL INSTANCES

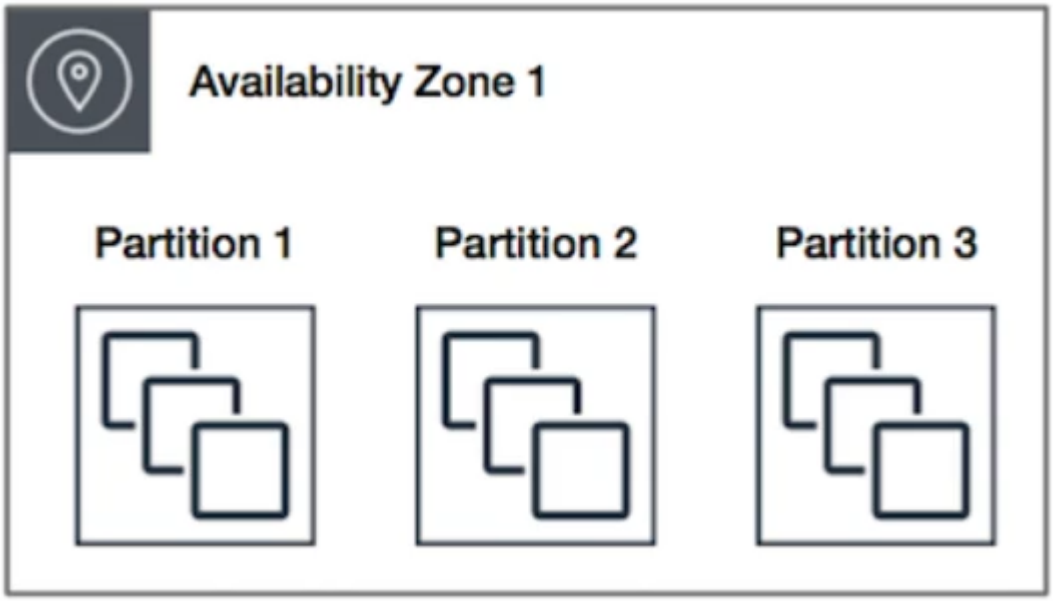


If any rack fails it will only effect that one instance, and the others will persist

Partitioned

When using partition placement groups, Amazon EC2 divides each group into logical segments called partitions. Amazon EC2 ensures that each partition within a placement group has its own set of racks. Each rack has its own network and power source. No two partitions within a placement group share the same racks, allowing you to isolate the impact of hardware failure within your application.

THINK MULTIPLE INSTANCES



Exam Tips

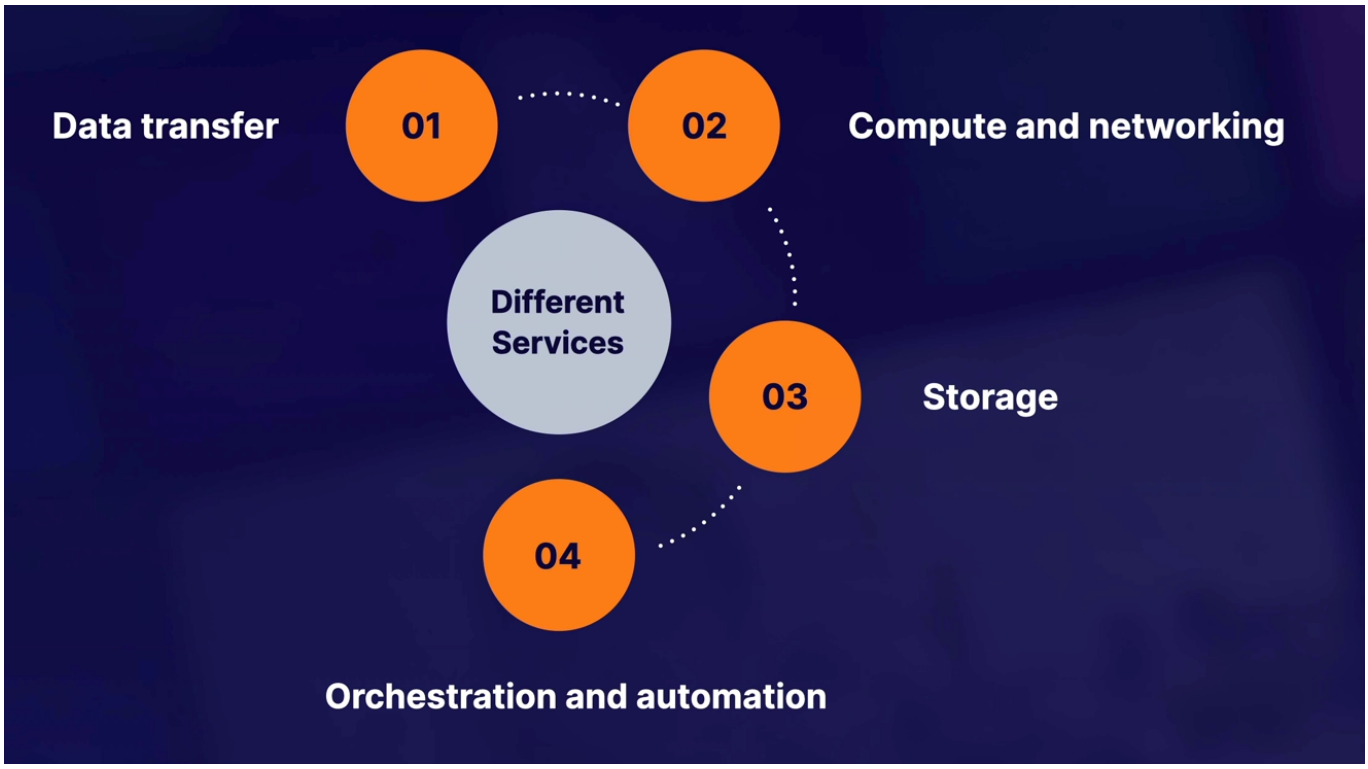
- Clustered Placement Group
    - Low network latency / high network throughput
- Spread Placement Group
    - individual critical ec2 instances
- Partitioned
    - multiple EC2 instances HDFS, HBase, and Cassandra
- A clustered placement group can't span multiple AZ
- A spread placement and partitioned group can
- The name you specify for a placement group must be unique within your AWS account
- Only certain types of instances can be launched in a placement group (Compute Optimized, GPU, Memory Optimized, Storage Optimized)
- AWS recommend homogenous instances within clustered placement groups
- You can't merge placement groups
- You can move an existing instance into a placement group. Before you move the instance, the instance must be in the stopped state. You can move or remove an instance using the AWS CLI or an AWS SDK, you can't do it via the console yet.

HPC on AWS

You can create a large number of resources in almost on time. You only pay for the resources you use - and, once finished, you can destroy the resources.

HPC is used for industries such as genomics, finance and financial risk modeling, machine learning, weather prediction, and even autonomous driving

What are the different services we can use to achieve HPC on AWS



Some ways we can get our data into AWS

- Snowball, Snowmobile (terabytes / petabytes worth of data)
- AWS DataSync to store on S3, EFS, FSx for Windows, etc
- Direct Connect

Direct Connect

AWS Direct Connect is a cloud service solution that makes it easy to establish a dedicated network connection from your premises to AWS. Using AWS Direct Connect, you can establish private connectivity between AWS and your data center, office, or colocation environment -- which, in many cases can reduce your network costs, increase bandwidth throughput, and provide a more consistent network experience than internet-based connections

HPC on AWS: Compute & Networking

What are the compute and networking services that allow us to achieve HPC on AWS

- EC2 instances that are GPU or CPU optimized
- Enhanced networking
- EC2 fleets (Spot Instances or Spot Fleets)
- Elastic Network Adapters
- Placement groups (cluster placement groups)
- Elastic Fabric Adapters

What is Enhanced Networking?

- It uses single root I/O virtualization (SR-IOV) to provide high-performance networking capabilities on supported instance types. SR-IOV is a method of device virtualization that provides higher I/O performance and lower CPU utilization when compared to traditional virtualized network interfaces
- Enhanced networking provides higher bandwidth, higher packet per second (PPS) performance, and consistently lower inter-instance latencies. There is no additional charge for using enhanced networking
- Use where you want good network performance

Depending on your instance type, enhanced networking can be enabled using an:

- Elastic Network Adapter (ENA), which supports speeds of up to 100Gbps for supported instance types

OR

- Intel 82599 Virtual Function (VF) interface, which supports network speeds of up to 10Gbps for supported instance types. This is typically used on older instances (LEGACY)
    - NOTE: in any scenario question: ENA over VF is the better option

What is an Elastic Fabric Adapter

- An elastic Fabric Adapter (EFA) is a network device you can attach to your Amazon EC2 instance to accelerate HPC and machine learning applications
- EFA provides lower, more consistent latency and higher throughput than the TCP transport traditionally used in cloud-based HPC systems
- EFA can use OS-bypass, which enables HPC and machine learning applications to bypass the operating system kernel and communicate directly with the EFA device. It makes it a lot faster with much lower latency. It is not supported with Windows currently -- only linux

HPC on AWS: Storage

Storage services that allow us to achieve HPC on AWS

Instance attached storage:

- EBS: Scale up to 64,000 IOPS with Provisioned IOPS (PIOPS)
- Instance Store: Scale to millions of IOPS; low latency

Network Storage

- Amazon S3: Distributed object-based storage; not a file system
- Amazon EFS: Scale IOPS based on total size, or use Provisioned IOPS
- Amazon FSx for Lustre: HPC-optimized distributed file system; millions of IOPs, which is also backed by S3

HPC on AWS: Orchestration & Automation

What are the orchestration and automation services that allow us to achieve HPC on AWS?

AWS Batch

- AWS Batch enables developers, scientists, and engineers to easily and efficiently run hundreds of thousands of batch computing jobs on AWS.
- AWS Batch supports multi-node parallel jobs, which allows you to run a single job that spans multiple EC2 instances.
- You can easily schedule jobs and luanch EC2 instances according to your needs

Orchestration & Automation: AWS ParallelCluster

AWS ParallelCluster

1. Open-source cluster management tool that makes it easy for you to deploy and manage HPC clusters on AWS
2. ParallelCluster uses a simple text file to model and provision all the resources needed for your HPC applications in an automated and secure manner
3. Automated creation of VPC, subnet, cluster, and instance types

Exam Tips

We can achieve HPC on AWS through

- Data Transfer
    - Snowball, Snowmobile
    - AWS DataSynce to store on S3, EFS, FSx for Windows, etc
    - Direct Connect
- Compute and Networking
    - EC2 instances that are GPU or CPU optimized
    - EC2 fleets (spot instances or spot fleets)
    - Placemene groups (cluster placement groups)
    - Enhanced networking single root I/O virtualization (SR-IOV)
    - Elastic network adapters or intel 82599 virtual function (VF) interface
    - Elastic Fabric Adapters
- Storage

    - Instance-attached storage
        - EBS: scale up to 64,000 IOPS with provisioned IOPS (PIOPS)
        - Instance store: scale to millions of IOPS; low latency
    - Network storage:
        - Amazon S3: Distributed object-based storage; not a file system
        - Amazon EFS: scale IOPS based on total size; or use Provisioned IOPS
        - Amazon FSx for Lustre: HPC-optimized distributed file system; millions of IOPs, which is also backed by S3
- Orchestration and automation
    - AWS Batch
    - AWS ParallelCluster

AWS WAF

AWS WAF is a web application firewall that lets you monitor the HTTP and HTTPS requests that are forwarded to Amazon CloudFront, and Application Load Balancer or API Gateway

AWS WAF also lets you control access to your content -- application layer (layer 7 in the OSI Model)

Where as a physical firewall can only go upto a layer 4

Sample query string parameter : `http://acloud.some?id=1001&name=johndoe`

You can configure conditions such as what IP addresses are allowed to make this request or what query string parameters need to be passed for the request to be allowed

Then the application load balancer or CloudFront or API Gateway wll wither allow this content to be received or to give a HTTP 403 status Code

At its most basic level, AWS WAF allows 3 different behavior

1. Allow all requets except the ones specify
2. Block all requests except the ones you specify
3. Count the requests that match the properties you specify

Extra Protection against web attacks using condiitons you specify. You can define conditions by using characteristics of web requests such as

- ip addresses that requests originate from
- country that requests originate from
- values in request headers

- strings that appear in requests, either specific strings or string that match regular expression (regex) pattersn
- length of requests
- presence of SQL code that is likely to be malicious (known as SQL injection)
- presence of a script that is likely to be malicious (known as cross-site scripting)

Exam Tips

Will be given different scenarios and will be asked how to block malicious ip addresses

- use AWS WAF
- use Network ACLs - covered more in VPC section

EC2 Summary

Amazon Elastic Compute Cloud (EC2) is a web service that provides resizable compute capacity in the cloud. Amazon EC2 reduces the time requires to obtain and boot new server instances to minutes, allowing you to quickly scale capacity, both up and down, as your computing requirements change.

EC2 Pricing Models

1. On Demand
    - Allows you to pay a fixed rate by the hour (or by the second) with no commitment
2. Reserved
    - Provides you with a capacity reservation, and offer a significant discount on the hourly charge for an instance. Contract Terms are 1 year or 3 year terms.
3. Spot
    - Enables you to bid whatever price you want for instance capacity, providing for even greater savings if your applications have flexible start and end times
4. Dedicated Hosts
    - Physical EC2 server dedicated for your use. Dedicated hosts can help you reduce costs by allowing you to use your existing server-bound software licenses.

If the Spot Instance is terminated by Amazon EC2, you will not be charged for a partial hour of usage. However, if you terminate the instance yourself, you will be charged for any hour in which the instance ran

EC2 Instance Types - Mnemonic

- F → For FPGA
- I → For IOPS
- G → Graphics
- G → High Disk Throughput
- T → Cheap general purpose (think T2 micro)
- D → Density
- R → For ram
- M → Main choice for general purpose apps
- C → for compute
- P → graphics (think pics)
- X → Extreme memory
- Z → extreme memory and CPU
- A → arm-based workloads
- U → Bare Metal

EBS

- Termination Protection is turned off by default, you must turn it on
- On an EBS-backed instance, the default action is for the root RBS volume to be deleted when the instance is terminated
- EBS Root volumes of your DEFAULT AMI's can be ecnrytped. You can also use a third part tool (such as bit locker etc) to encrypt the root volume, or this can be done when creating AMI's in the AWS Console or using the API
- Additional volumes can be encrypted
- All inbound traffic is blocked by default
- All outbound traffic is allowed
- changes to security groups take effect immediately
- you can have any number of EC2 instances within a security group
- you can have multiple security groups attached to EC2 isntances
- Security groups are stateful
- if you create an inbound rule allowing traffic in, that traffic is automatically allowed back out again
- You cannot block specific ip addresses using security groups, instead use network access control lists

- you can specify allow rules, but not deny rules

5 Different Types of EBS Storage

1. General Purpose (SSD)
2. Provisioned IOPS (SSD)
3. Throughput Optimised Hard Disk Drive
4. Cold Hard Disk Drive
5. Magnetic

Comparison

Solid-State Drives (SSD)			Hard disk Drives (HDD)		
Volume Type	**General Purpose SSD**	**Provisioned IOPS SSD**	**Throughput Optimized HDD**	**Cold HDD**	**EBS Magnetic**
Description	General purpose SSD volume that balances price and performance for a wide variety of transactional workloads	Highest-performance SSD volume designed for mission-critical applications	Low cost HDD volume designed for frequently accessed, throughput-intensive workloads	Lowest cost HDD volume designed for less frequently accessed workloads	Previous generation HDD
Use Cases	**Most Work Loads**	**Databases**	**Big Data & Data Warehouses**	**File Servers**	**Workloads where data is infrequently accessed**
API Name	gp2	io1	st1	sc1	Standard
Volume Size	1 GiB - 16 TiB	4 GiB - 16 TiB	500 GiB - 16 TiB	500 GiB - 16 TiB	1 GiB-1 TiB
Max. IOPS**/ Volume	16,000	64,000	500	250	40-200

- Volumes exist on EBS. think of EBS as virtual hard disk
- Snapshots exists on S3. Think of snapshots as photograph of the disk
- Snapshots are point in time copies of Volumes
- Snapshots are incremental -- This means that only the blocks that have changed since your last snapshot are moved to S3
- If this is your first snap,shot it may take some time to create
- To create a snapshot for Amazon EBS volumes that serve as root devices, you should stop the instance before taking the snapshot
- However you can take a snap while the instance is running
- You can create AMI's from both Snapshots and Volumes

- You can change EBS volume sizes on the fly, including changing the size and storage type
- Volumes will always be in the same availability zone as the EC2 instance
- To move an EC2 volume from one AZ to another, take a snapshot of it, create an AMI from the snapshot and then use the AMI to launch the EC2 instance in a new AZ
- To move an EC2 volume from one region to another, take a snapshot of it, create an AMI from the snapshot and then copy the AMI from one region to the other. Then use the copied AMI to launch the new EC2 instance in the new region.
- Snapshots of encrypted volumes are encrypted automatically
- Volumes restored from encrypted snapshots are encrypted automatically
- You can share snapshots, but only if they are unencrypted
- These snapshots can be shared with other AWS accounts or made public
- For uncrypted devices
    - Create a snapshot of the unencrypted root device volume
    - Create a copy of the snapshot and select the encrypt option
    - Create an AMI from the encrypted snapshot
    - Use that AMI to launch a new encrypted instances
- Instance store volumes are sometimes called Ephemeral Storage
- Instance store volumes cannot be stopped. If the underlying host fails, you will lose your data.
- EBS backed instances can be stopped. You will not lose the data on this instance if it is stopped
- You can reboot both, you will not lose your data
- By default, both ROOT volumes will be deleted on termination. However, with EBS volumes, you can tell AWS to keep the root device volume
- ENI
    - For basic networking. Perhaps you need a separate management network to your production network or a separate logging network and you need to do this at a low cost. In this scenario use multiple ENIs for each network

- Enhanced Network
    - For when you need speeds netweek 10Gbps and 100Gbps. Anywhere you need reliable, high throughput
- Elastic Fabric Adaptor
    - For when you need to accelerate high performance computing (HPC) and machine learning applications or if you need to do an OS by-pass. If you see a scenario question mentioning HPC or ML and asking what network adaptor you want, choose EFA
    - CloudWatch is used for monitoring performance
    - CloudWatch can monitor most of AWS as well as your applications that run on AWS
    - CloudWatch with EC2 will monitor events every 5 minutes by default
    - you can have 1 minute intervals by turning on detailed monitoring
    - You can create CloudWatch alarms which trigger notifications
    - CloudWatch is all about performance. CloudTrails is all about auditing

What can you do with CloudWatch
- Dashboards - Creates awesome dashboards to see what is happening with your AWS environment
- Alarms - allows you to set Alarms that notify you when particular thresholds are hit
- Events - CloudWatch Events helps you to respond to state changes in your AWS resources
- Logs - CloudWatch Logs helps you to aggregate, monitor, and store logs
- CloudTrail → monitors API calls in the AWS platform
- CloudWatch → performance

- You can interact with AWS from anywhere in the world just by using the command line
- You will need to set up access in IAM
- Basic commands will be useful to know
- Roles are more secure than storing your access key and secret access key on individual EC2 instances

- Roles are easier to manage
- Roles can be assigned to an EC2 instance after it is created using both the console and command line
- Roles are universal -- can use them in any region
- Bootstrap scripts run when when an ec2 instance first boots
- can be a powerful way of autoamting software isntalls and updates
- Used to get information about an instance (such as public ip)
- `curl http://169.254.169.254/latest/meta-data`
- `curl http://169.254.169.254/latest/user-data`
- Supports the network file system version 4 (NFSv4) protocol
- you only pay for the storage you use (no pre-provisioning required)
- can scale up to the petabytes
- can support thousands of concurrent nfs connections
- data is stored across multiple AZ's witihn a region
- Read After Write Consistency
- EFS → When you need distributed, highly resilient storage for Linux instances and Linux based applications
- Amazon FSx for Windows → When you need centralised storage for Windows-based applications such as Sharepoint, Microsoft SQL Server, Workspaces, IIS Web Server or any other native Microsoft Application
- Amazon FSx for Lustre → When you need high-speed, high-capacity distributed storage. This will be for applications that do high performance compute (HPC), financial modelling etc. Remember that FSx for Lustre can sotre data directly on S3
- Clustered Placement Group
    - Low network latency / high network throughput
- Spread Placement Group
    - individual critical ec2 instances

- Partitioned
    - multiple EC2 instances HDFS, HBase, and Cassandra
- A clustered placement group can't span multiple AZ
- A spread placement and partitioned group can
- The name you specify for a placement group must be unique within your AWS account
- Only certain types of instances can be launched in a placement group (Compute Optimized, GPU, Memory Optimized, Storage Optimized)
- AWS recommend homogenous instances within clustered placement groups
- You can't merge placement groups
- You can move an existing instance into a placement group. Before you move the instance, the instance must be in the stopped state. You can move or remove an instance using the AWS CLI or an AWS SDK, you can't do it via the console yet.

EC2 - Lab1 > EC2 Quiz