IAM:

* STS API Calls: credentials are returned with the following components
  + Session Token
  + Access Key ID
  + Secret Access Key
  + Expiration

Web Identity Federation: Authenticate to AWS from third-party providers such as facebook or google.

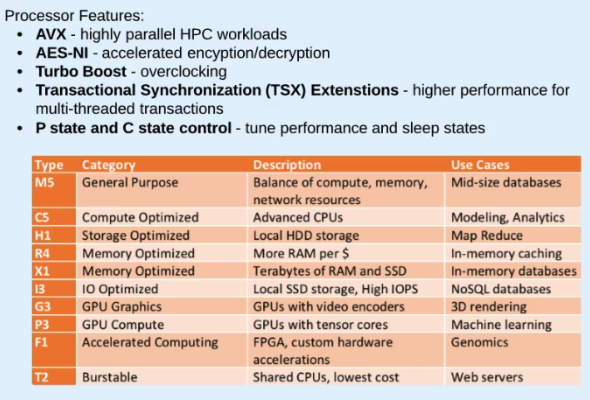
Server-Based Compute Services

# EC2:

Types of Virtualization for Linux instances

* HVM AMIs (Hardware Virtual Machines (recommended standard)
  + Runs like bare metal hardware
  + Get hardware extensions (works like normally do in datacenter)
* PV AMIs (Paravirtual)
  + Host hardware that does not have explicit support for virtualization
  + No special hardware extensions (enhanced n etworking or GPU processing)

# Instance Types:



User data command on instance:

* Curl http<nolink>://169.254.169.254/latest/user-data

Public IP

* Static public IP = Elastic IP Address
  + When you stop and start instance: keep same public IP
  + When you terminate instance: only then will the public IP get put back into the pool
  + Have to attach elastic IP address and it will replace the default public IP assigned
* Public IP Address: when you stop the instance AWS takes back public address

EBS Snapshots: stored in S3

* Can create AMI (image) for an instance from a snapshot
* Create a volume from a snapshot
* Network attached volumes (persistent volumes, they can live past the life of an EC2 instance)

EFS (Elastic File System)

* Shared file system for EC2
* Elastic (thin provisioned)
* Only compatible with Linux Servers \*\*\*\*\*\*\*\*
* Can mount to on prem servers

Placement Groups: (NO EXTRA CHARGE FOR THIS)

* Cluster Placement Group
  + Cluster of instances (AWS DRS)
  + Example: Can put all instances on one host in datacenter or physically close as possible to reduce latency and maximize network throughput
  + Use Enhanced networking to maximize benefit
  + Not supported for T2 Instances
  + Best to launce all instances needed at ONE TIME
* Spread Placement Group
  + Place instances on distinct host hardware (spread it out amongst hardware)

Billing:

* On demand –
  + most expensive for long running instances
  + Most flexible
  + Charged per second or per hour
* Reserved: set time period of 1 – 3 years
  + Pay upfront, partial upfront, no upfront
    - Standard (up to 75% discount)
      * Capacity reservation in an AZ
    - Convertible (up to 54% discount)
      * Can change/upgrade policy
    - Scheduled (1yr term, 5-10% discount)
      * Schedule exact times when the instance can run/launch

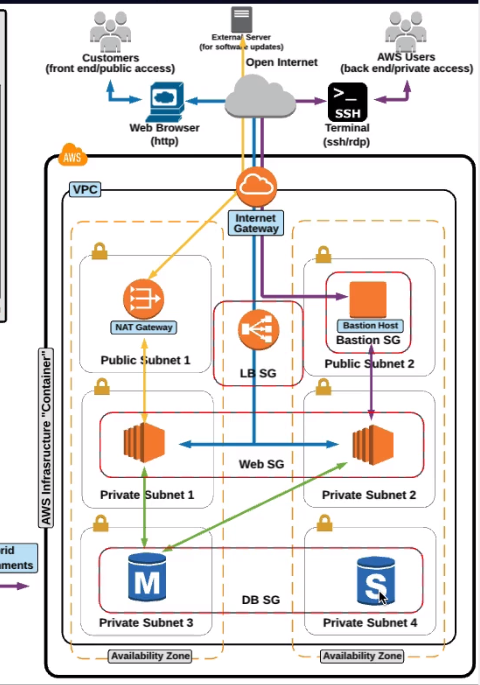
VPC (Virtual Private Cloud)

* If you do not specify a subnet when creating an instance it will be created in the default VPC and subnet

High Availability

Elastic Load Balancing

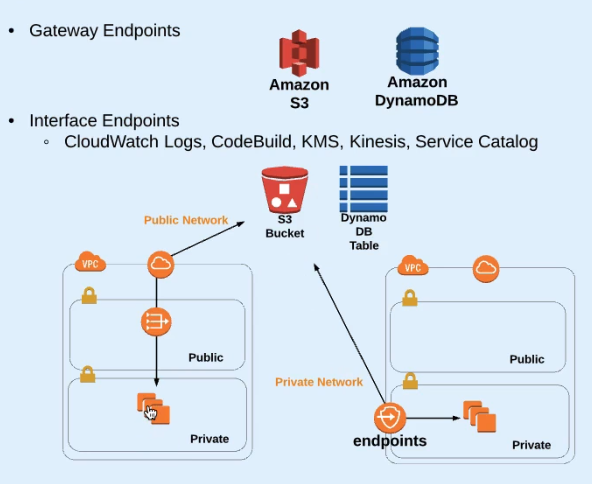
* EC2 instances being launched behind the scenes to load balance traffic coming in from the interweb
* Three types of ELB
  + Classic load balancer
    - Simple balancing of traffic
    - Support TCP, SSL, HTTP, HTTPS
    - No special routing requests
  + Application ELB
    - Content-based “rules”
    - Route based on HTTP headers
    - Target groups (groups of EC2 instances)
    - Supports ECS (Elastic container service), EKS (kubernetes), HTTPS, HTTP/2, WebSockets, Sticky Sessions, WAF
  + Network LB
    - Layer 4 TCP/UDP load balancing
    - Designed for extreme performance
      * Does not need to scale, already designed to handle huge traffic spikes
    - No SSL offloading
    - No instances just a software designed load balancer
      * Just passes traffic
    - Multiple IP addresses, must point to DNS that has all IP addresses associated with DNS name
* Bastion Host: EC2 instance that lives in a public subnet and is used as a gateway for traffic that is destined for instances in private subnets. Admins use for SSH/RDP access from web
  + All traffic must pass through it first from the web
  + Used for Administrators/users to access the servers when a Load balancer is present (all traffic is traveling through this load balancer and offloading ssl so admins can’t get to the instances behind the LB



* NAT Gateway
  + For traffic to get a route from private subnet to the internet
  + NAT Instance
    - Identical to NAT Gateway except it is executed differently by configuring an actual EC2 instance
    - More of a legacy feature in AWS

VPC Endpoints:

* Used when you don’t want certain data to travel from private subnets over the internet (needs to stay off the web)
  + Essentially adds a route for the vpc to the service you want so that it bypasses the internet gateway and you use internal IP addresses
* Gateway Endpoints
  + S3
  + DynamoDB
* Interface Endpoints
  + CloudWatch Logs
  + CodeBuild
  + KMS
  + Kinesis
  + Service Catalog



Stateless Applications: (Store state information off-instance)

* Stores data from a session to a shared common database table (no SQL database)
  + Instead of saving data locally to the instances that the load balancer sends the traffic too
* Shared Filesystem
* All instances point to a common database so no matter what instance it hits, it can see all the data

HA vs Fault Tolerance

* HA – architectures that continue to remain available to end users in the event of a systems or component failure
* Fault Tolerance – architectures that not only remain available during an outage but also suffer no degradation in performance during the outage
* Scalability – ability to easily increase or decrease in size and capacity
  + Vertical scaling – increase capacity of single instance
  + Horizontal scaling – add or remove number of instances
* Elasticity – ease of a systems ability to change or adapt
  + Scaling in or out
  + Updating firewall rules
  + Remapping IP addresses

AutoScaling:

* Automatically increase or decrease instances available for your application
* Based on metrics from Cloudwatch

DNS – Route 53

Hosted Zones – records for your domain

Failover routing policy –

* Using S3 for DNS failover
  + Used as a primary endpoint as a static webpage
* Setup routing Policy to “Failover” and set Primary as Load balancer and secondary as S3 bucket containing website

Cloud Front: Content Delivery Network

* Web Distribution:
  + Used for distribution of static and dynamic content
    - .html
    - .css
    - .php and graphic files
    - Media files HTTP or HTTPS
    - Web forms
    - Live streaming
* RTMP: streaming media files using Adobe flash media servers
  + Must be stored in an S3 bucket
* Accelerate delivery of content to users by caching content at edge locations and will pull the content based off of geo-location from the closest cached edge location
* Origin: the source of content
  + S3 Bucket
  + ELB
  + Public HTTP endpoint
  + Video streaming
* Large objects at the origin, you want a longer TTL so it isn’t always trying to move the large files as much
* Query strings and TTL need to be set based off of usage, size, and variables changing