**AWS CSA-Pro Notes ACloudGuru Section 7 Business Continuity**

**Exam Tips:**

**General Concepts:**

**Know the difference between Business Continuity, Disaster Recovery, and Service Levels**

* Business Continuity- Seeks to minimize business disruptions when something unexpected happens
* Disaster Recovery- In the case of a Disaster what are your plans to recover?
* Service Levels- An agreed goal or target for a given service on its performance or availability

**Know the difference between High Availability and Fault Tolerance**

* High Availability- Designing in redundancies to reduce the chance of impacting service levels
* Fault Tolerance- Designing in the ability to absorb problems without impacting service levels

**Understand the inter-relationships and how AWS uses the terms**

**Know the difference between RTO and RPO**

* RTO- Is the amount of time you are allowed to be down based on SLA’s
* RPO- Is the amount of data you are allowed to lose based on SLA’s

**Know the four general types of DR architectures and trade-offs of each**

* Backup and Restore- Cost the least takes the longest to restore
* Pilot Light- Cost second least takes the second longest to restore
* Warm Standby- Cost second most takes the second least to restore
* Multi-Site- Cost most and able to restore almost instantly

**Storage Options:**

**Understand the HA capabilities and limitations of AWS storage options**

**Know when to use each storage option to achieve the required level of recovery capability**

**Understand RAID and the potential benefits and limitations**

* Never use RAID5 and RAID6
* RAID0 is Striping- Allows for better performance
* RAID1 is Mirroring- Allows for better redundancy

**Compute Options:**

**Understand why horizontal scaling is preferred from an HA perspective**

* Balancing your load across many instances leave room for failure.
* If you have a single large instance you have a single point of failure

**Know that compute resources are finite in an AZ and know how to guarantee their availability**

* Reserved instances and Dedicated Hosts are the only way to guarantee space

**Understand how Auto Scaling and ELB can contribute to HA**

* Dynamically scale your instances horizontally based on ELB health checks
* When one instances dies the ELB will health check it and it will fail
* Auto Scaling will trigger and bring up a new instance for you

**Database Options:**

**Know the HA attributes of various Database services**

* DynamoDB- Multi-AZ and managed for you
* RDS- Aurora is multi-AZ by default. Use the Multi-AZ option to have an active standby deployment of your RDS database
* Redshift- No Multi-AZ. Go with a multi node deployment and take snapshots frequently

**Understand the different HA approaches and risks for Memcached and Redis**

* Memcached allows for no Multi-AZ deployments
* Redis does
* You can architect your infrastructure with highly available caching engines but at the end of the day they are just caches. If you really need HA then think about using an actual database

**Know which RDS options require manual failover and which are automatic**

* Automatic for Multi-AZ deployments of RDS
* Read Replicas will not auto failover unless you are using Aurora

**Network Options:**

**Know which networking components are not redundant across AZs and how to architect for them to be redundant**

**Understand the capabilities of Route 53 and Elastic IP in the context of HA**

* Route 53 health checks to auto detect system problems

**Concepts:**

**Terms:**

**Business Continuity:**

* Seeks to minimize business activity disruption when something unexpected happens

**Disaster Recovery:**

* Act of responding to an event that threatens business continuity

**High Availability (HA):**

* Designing in redundancies to reduce the chance of impacting service levels

**Fault Tolerance:**

* Designing in the ability to absorb problems without impacting service levels

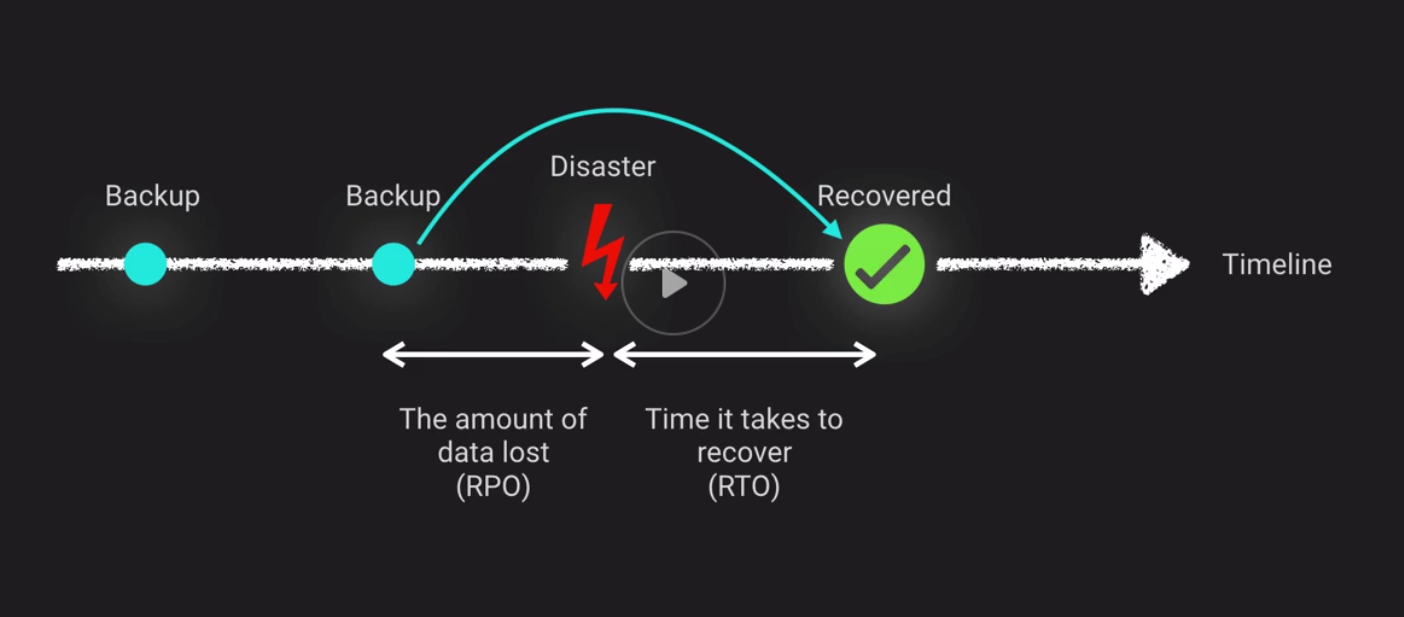
**Service Level Agreement (SLA):**

* An agreed goal or target for a given service on its performance or availability

**Recovery Time Objective (RTO):**

* Time that it takes after a disruption to restore business processes to their service levels
* The amount of time you can be done based on your SLA’s

**Recovery Point Objective (RPO):**

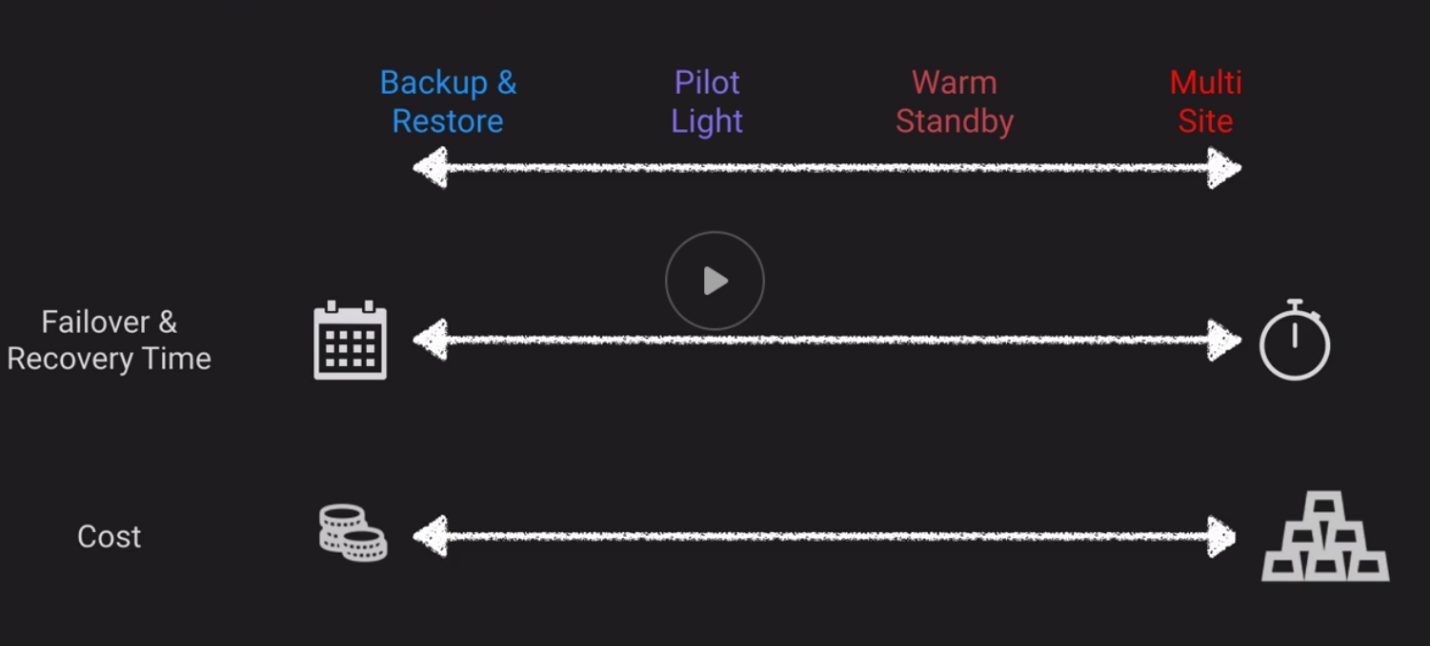
* Acceptable amount of data loss measured in time
* The amount of data your company can lose based on SLA’s when down

**Disaster Types:**

* Hardware Failure
* Deployment Failure
* Load Induced
* Data Induced
* Credential Expiration
* Dependency
* Infrastructure Disaster
* Identifier Exhaustion
* Human Error

**AWS Continuum of HA:**

**Disaster Recovery Architectures:**

****

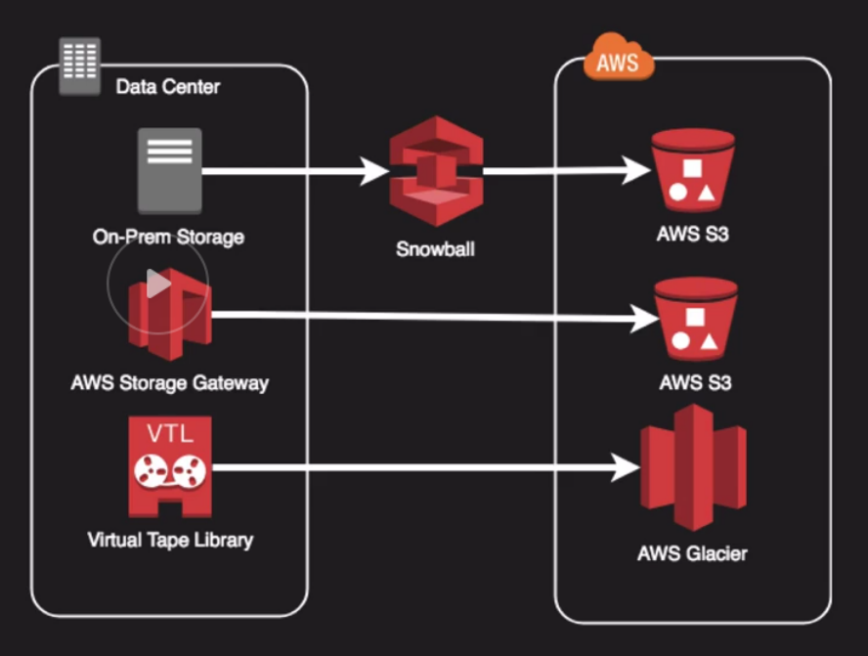
**Backup and Restore:**

**Pros:**

* Very common entry point into AWS
* Minimal effort to configure

**Cons:**

* Least flexibility
* Analogous to off-site backup storage

****

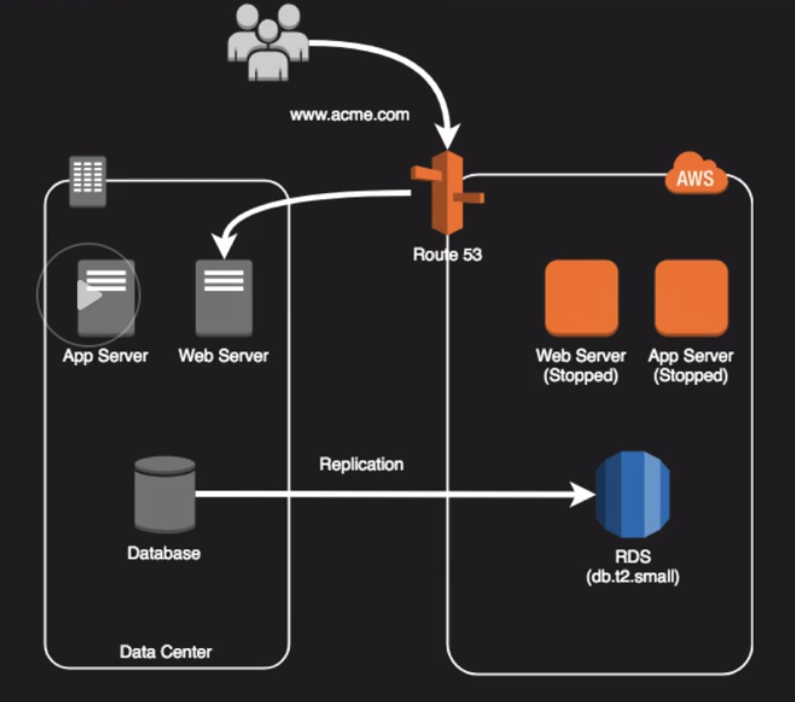
**Pilot Light:**

**Pros:**

* Cost effective way to maintain a “hot site” concept
* Suitable for a variety of landscapes and applications

**Cons:**

* Usually requires manual intervention for failover
* Spinning up cloud environments will take minutes or hours
* Must keep AMIs up-to-date with on-prem counterparts

****

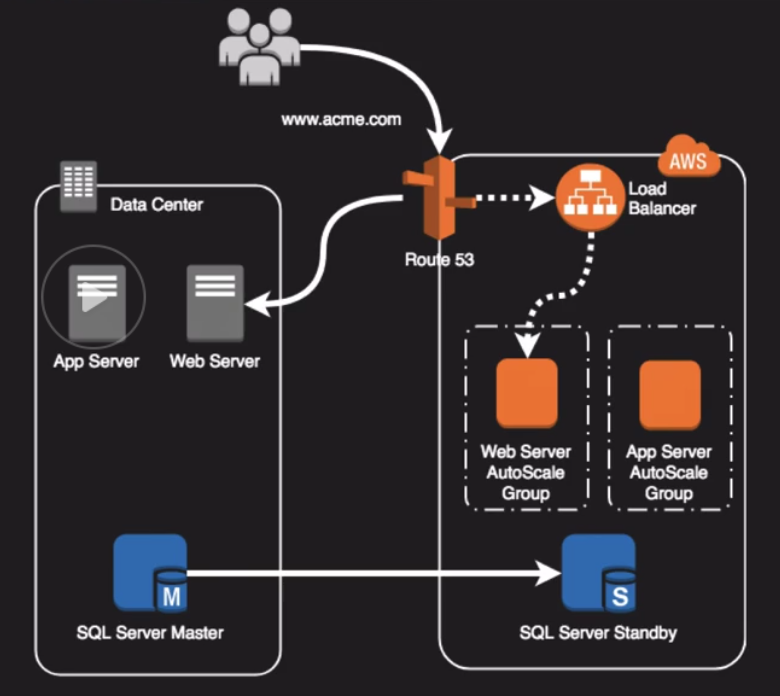
**Warm Standby:**

**Pros:**

* All services are up and ready to accept a failover faster within minutes or seconds
* Can be used as a “shadow environment” for testing or production staging

**Cons:**

* Resources would need to be scaled to accept production load
* Still requires some environment adjustments but could be scripted

****

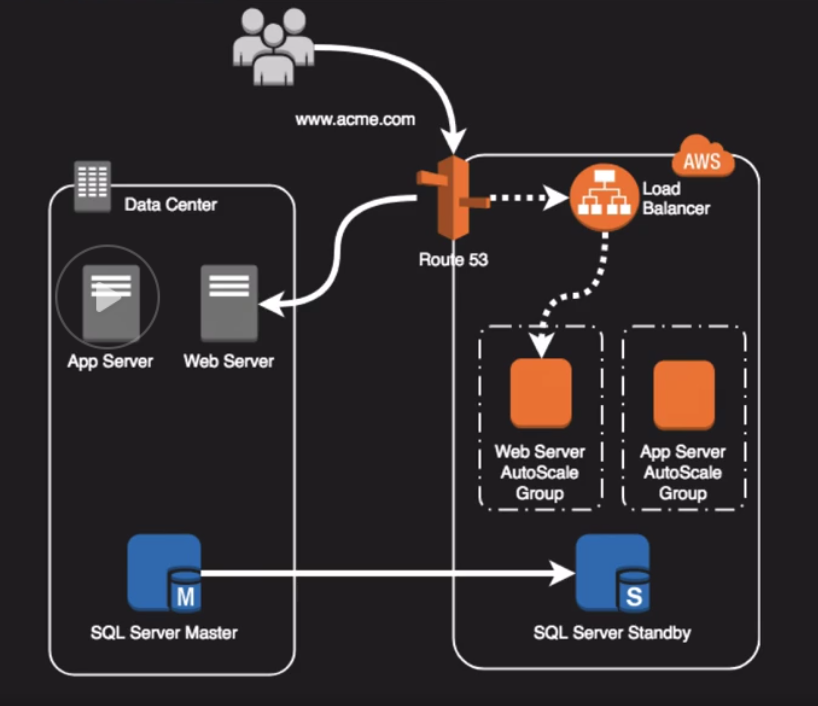
**Multi-Site:**

**Pros:**

* Ready all the time to take full production load. Effectively a mirrored data center
* Fails over in seconds or less
* No or little intervention required to fail over

**Cons:**

* Most expensive DR solution
* Can be perceived as wasteful as you have resources just standing around waiting for the primary to fail

****

**Storage HA Options:**

**EBS Volumes:**

* Annual failure rate less than 0.2% compared to commodity hard drive at 4%
* Availability target of 99.999%
* Replicated automatically within a single AZ
* Vulnerable to AZ failure
* Easy to snapshot, which is stored on S3 and multi-AZ durable
* You can copy snapshots to other regions as well
* Supports RAID configurations

**RAID Configs:**

**RAID0:**

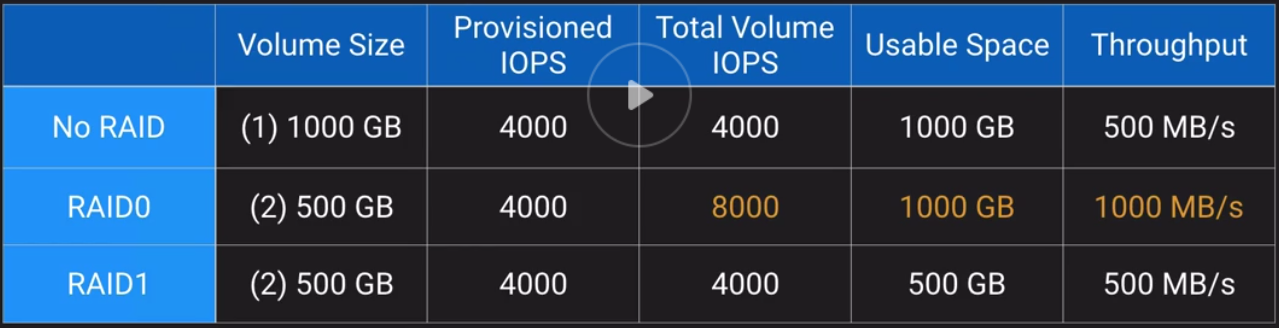
* Striping option of RAID needing 2 disks to work
* Offers no redundancy, but increases Reads and Writes to your disks

**RAID1:**

* Mirroring option of RAID needing 2 disks
* 1 drive can fail as it is basically writing the same data to 2 disk drives
* Reads and Writes are impacted and RAID1 has 50% capacity as RAID0

**RAID5 and RAID6 are not recommended for use in AWS:**

* EBS volumes are accessed over a network
* Writing parity bits sucks up a lot of I/O on your volumes

****

**S3 Storage:**

* Standard Storage Class
* Standard Infrequent Access
* One-Zone Infrequent Access

**Amazon EFS:**

* Implementation of the NFS file systems
* True file system as opposed to block storage (EBS) or object storage (S3)
* File locking, strong consistency, concurrently accessible
* Each file object and metadata is stored across multiple AZs
* Can be accessed from all AZs concurrently
* Mount targets are highly available

**Amazon Storage Gateway:**

* Good way to migrate on-prem data to AWS for offsite backup
* Best for continuous sync needs

**Snowball:**

* Various options for migrating data to AWS based on volume
* Only for batch transfers of data

**Glacier:**

* Safe offsite archive storage
* Long-term storage with rare retrieval needs

**Compute HA Options:**

* Up-to-Date AMIs are critical for rapid fail-over
* AMIs can be copied to other regions for safety or DR staging
* Horizontally scalable architectures are preferred because risk can be spread across multiple smaller machines versus one larger machine
* Reserved Instances is the only way to guarantee that resources will be available when needed
* Auto Scaling and Elastic Load Balancing work together to provide automated recovery by maintaining minimum instances
* Route 53 Health Checks also provide “self-healing” redirection of traffic
  + You can do these health checks on anything with an HTTP endpoint

**Database HA Options:**

* If possible, choose DynamoDB over RDS because of inherent fault tolerance
* If DynamoDB can’t be used, choose Aurora because of redundancy and automatic recovery features
* If Aurora can’t be used, choose Multi-AZ RDS
* Frequent RDS snapshots can protect against data corruption or failure and they won’t impact performance of multi-AZ deployments
* Regional replications is also an option, but will not be strongly consistent
* If Database on EC2, you will have to design your own HA

Multi-AZ has synchronous replication

Read Replicas have asynchronous replication

If we lose a master DB RDS will auto promote a standby to be the new master and replication continuous.

Read Replica promotion to a master is not automatic unless you are using Amazon Aurora.

**HA for Redshift:**

* Currently, Redshift does not support multi-AZ deployments
* Best HA option is to use a multi-node cluster which supports data replication and node recovery
* A single node Redshift cluster does not support data replication and you will have to restore from a snapshot on S3 if a drive fails

**HA for ElasticCache:**

**Memcached:**

* Because Memcached does not support replication, a node failure will result in data loss
* Use multiple nodes in each shard to minimize data loss on node failure
* Launch multiple nodes across available AZs to minimize data loss on AZ failure

**Redis:**

* Use multiple nodes in each shard and distribute the nodes across multiple AZs
* Enable multi-AZ on the replication group to permit automatic failover if the primary node fails
* Schedule regular backups of your Redis cluster

**Network HA Options:**

* By creating subnets in the available AZs, you create multi-AZ presence for your VPC
* Best practice is to create at least 2 VPN tunnels into your Virtual Private Gateway
* Direct Connect is not HA by default, so you need to establish a secondary connection via another Direct Connect
* Route 53 Health Checks provide basic level of redirection DNS resolution
* Elastic IPs allow you flexibility to change out backing assets without impacting name resolution
* For multi-AZ redundancy of NAT Gateways, create gateways in each AZ with route for private subnets to use the local Gateway

**Extra:**

What is cross-zone load balancing?

Sending load equally across all active AZ’s attached to a specific load balancer.

What does Aurora Postgre not support in terms of Read Replicas?

Postgre Aurora does not support Read Replicas in separate regions.