Disaster Recovery and Migration

Disaster recovery

Disaster recovery is about preparing for those disaster that can happen to the data center.

RPO: Recovery point objective. How often do you run backups. How far can you go back just before the data loss. "How much data did you lose just before the disaster". It will be measure in terms of time, the lower the better, that means that you basically lose little to no data.

RTO: Recovery time object. How long will it take you recover from the disaster. How much downtime after the disaster. Again also in time the lower the better.

Disaster recovery strategies

All these strategies will have different RPO and RTO and different cost.

Backup and restore

High RPO!

You make regular backups basically to the cloud, then when disaster strikes then you can just restore from the snapshot for RDS, or EBS.

This is cheap and the only cost is storing the backups, and high RPO and high RTO. Depending on how often do you make the backups.

Pilot light

A small version of the app is always running in the cloud. The useful and critical application.

Faster backup and restore since the critical system are already up. Then you just have to spin up the other non-core systems on the fly.

Lower RPO and RTO, but a little bit more expensive.

Warm standby

A full replication of the system but is scaled down. When disaster strikes then you scale it up.

Way faster RTO, and lower RPO but you are spending more now.

Multi Site / Hot Site Approach

You have basically a full replication not scaled down running concurrently on the cloud environment ready to go.

Very fast RTO and low RPO, his is the most expensive.

All AWS Multi Region

Deploy everything to the cloud to multiple regions, when one region is down just failover to the other.

Recovery tips

Backup do EBS snapshots, RDS automated backups. Push those snapshots and backup to S3.

High availability, use Route 53 to automatic migrate DNS from region to region.

RDS multi-AZ

DMS

Database migration service. You can quickly and securely migrate on-premise database to AWS. During migration the source database still remain available.

Supports lots of engine. Oracle to Oracle. Posgres to Postgres. Even Microsoft SQL server to Aurora.

Continuous data replication. You have to create an EC2 instance to do the migration and continuous replication if you need. The EC2 instances will basically pull data from the database and migrate it to the target.

Source and targets

Sources: On-premises and EC2 instances databases, even S3

Target: On-premise and EC2 instance databases. RDS, Redshift, DynamoDB, S3, Kinesis Data Stream

You can transform source to target they all work.

Schema conversion tools (SCT)

Convert database schema from one engine to another, if the engine aren't the same you will need to use this tool. If the database engine are the same then you don't need it.

Continuous replication

To set up continuous replication and not just a one time migration. You would still need SCT if the data engine aren't the same. Then you need the EC2 instance to do full migration and chance data capture (CDC) for continuous replication if there are new changes to the database.

RDS and Aurora Migration

RDS MySQL to Aurora

1. DB snapshot from RDS then restore it as Aurora. But there are downtime if you are taking the snapshot
2. Create an Aurora read replica from your RDS. Then do replication. Then promote the Aurora as own DB cluster. No downtime but there are cost

External MySQL to Aurora

1. Percona XtraBackup to backup your on-premise MySQL to S3, then restore Aurora from S3
2. Mysqldump and pipe it to Aurora, but this is slower since it doesn't use S3

If both database are up and running

Use DMS to do continuous replication.

RDS PostgresSQL to Aurora

1. DB snapshot and then restore it as PostgresSQL Aurora, since Aurora support Postgres and MySQL
2. The second option create read replica, then do replication and promote the Aurora as own DB cluster, same with MySQL.

External Postgres to Aurora

1. Create backup and put it in S3, then import it using Aurora extension

Other on-premise services

VM Import / export: You can move existing VM to AWS or export them back to on-premise

AWS Application Discovery Service: Discover your on-premise server layout and plan migration

AWS Database Migration Service: Help migrate on-premise database to the cloud.

AWS Server migration service: Help migrate on-premise server to cloud incrementally.

AWS Application Migration Service: Migrate existing websites, applications, servers, and VM, and data to the cloud. Life and shift solution that simplify migration to the cloud. Convert your existing data center and replicate it on the cloud basically.

AWS Backup

Centrally manage and automate backup across AWS service. No need to create custom script and manual processes.

Supports lots of services for making backups. Support cross-region and cross-account backups for disaster recovery.

On-demand and scheduled backups. Define frequently of how often the backup is done.

Data is backed up in S3.

You can apply write once read many. The backup cannot be deleted.

Transferring large amount of data to AWS

200 TB of data with 100 Mbps internet.

1. Over the internet / Site-to-site VPN: Immediate setup but very long time of transferring, so this isn't going to work
2. Direct connect (1 Gbps): A month to set up DX, then still lots of time to do it.
3. Snowball: Order snowballs, then one week to load data and sent it back.
4. On-going replication: Site-to-site. DataSync. DMS, or DX.

VMware Cloud on AWS

Have VMware virtual machines on the cloud basically if you use VMware on-premise.

Sub-services under VMware are available and you have access to AWS cloud services like S3, EC2, Direct Connect.

Revision #3
Created 2023-02-28 01:49:09 UTC by Tamarine
Updated 2023-03-01 00:42:30 UTC by Tamarine