Cantril.io

**RDS Proxy –**

* Why?
  + Opening and Closing Connections consumes resources
    - It takes time – which creates latency
  + With serverless – every Lambda opens and closes?
    - Inefficient
  + Handling failure of Database Instances is hard
    - Doing it within your application adds risks
  + DB proxies help – managing them is not trivial
    - Scaling/resilience
  + Applications => Proxy (connection pooling) => Database
* Architecture
  + Managed service
  + Runs within a VPC
  + Maintains a long term connection pool
  + Clients and Lambda functions connect to RDS proxy instead of RDS instance
  + Connections can be reused – avoiding the lag of establishment, usage, and termination for each invocation
  + Multiplexing is used so a smaller number of DB connections can be used for a larger number of client connections
  + Helps with DB failover events – abstracts away from DB failure or failover events
  + In the event there is a failover, RDS proxy establishes a new connection to the RDS instance - Client waits in unresponsive state
* When?
  + Too many connection errors – reduce # of connections to RDS
  + DB instances using T2/T3
    - Smaller / burst instances
  + AWS Lambda – time saved/connection reuse & IAM Auth
    - Lambda has access to via execution role
  + Long running connections (SaaS apps) – low latency
  + Where resilience to DB failure is a priority
  + RDS can reduce the time for failover
    - Make it transparent to the application
  + No need to wait for CNAME to move from Primary to Standby, proxy handles connection
* Key Facts
  + Fully Managed DB Proxy for RDS/Aurora
  + Autoscaling, highly available by default
  + Provides connection pooling – reduces DB Load
  + ONLY accessible from a VPC – no public internet
  + Accessed via Proxy Endpoint – no app changes
  + Can enforce SSL/TLS
  + Can reduce failover time by over 60% (66-67)
  + Abstracts failure of DB away from applications

**AWS Database Migration Service (DMS) -**

* Managed DB migration service
* Runs using a replication instance
* Source and Destination enpoints
  + Source and Target Databases
* One endpoint MUST be on AWS
* Replication instance runs multiple replication tasks
* Tasks move information moved from Source to Destination
* Replication instance performs the migration between Source and Destination Endpoints which store Connection information for source and target databases
* Jobs –
  + Full load – one off migration of all data
  + Full Load + CDC (Change Data capture) – ongoing replication which captures changes
  + CDC Only – if you want to use an alternative method to transfer the Bulk DB data – such as native tooling
    - Some DB engines – Oracle have their own import/export tools
* Doesn’t natively support Schema conversion
  + Schema Conversion Tool (SCT) can assist with schema conversion
    - Modifies schema between different DB versions or DB engines
* Allows for 0 data loss, low or 0 downtime migrations between 2 DB endpoints
* Capable of moving DB’s INTO or OUT of AWS

Schema Conversion Tool (SCT) –

* Standalone application
* SCT is used when converting one DB engine to another
* Larger migrations
  + Including DB -> S3 (Migrations using DMS)
* SCT is NOT used when migrating between DB’s of the same type
* Works with OLTP DB Type – MySQL, MSSQL, Oracle
* Works with OLAP DB Type - Teradata, Oracle, Vertica, Greenplum
* Example use cases:
  + On prem MSSQL -> RDS MySQL
  + On prem Oracle -> Aurora

SCT + DMS + Snowball –

* Larger migrations might be multi-TB in size
  + Moving data over networks takes time and consumes capacity
  + DMS can utilize snowball
* Step 1 – Use SCT to extract data locally and move to a snowball device
* Step 2 – Ship the device back to AWS. They load onto an S3 bucket
* Step 3 – DMS migrates from S3 into the target store
* Step 4 – Change Data Capture (CDC) – can capture changes, and via S3 intermediary they are also written to the target database

**Elastic File System (EFS) Architecture** –

* EFS is am implementation of NFSv4
* EFS Filesystems can be mounted in Linux
* Shared between many EC2 instances
* Private service – accessed via mount targets within a VPC
* Can also be accessed from on-premises via VPN or Direct Connect – Linux OS only
* Uses POSIX permissions - standard for interoperability used within Linux – all distros will understand
* Mount targets exist within a Subnet within a VPC
  + Best practice to have a mount target in every AZ that a VPC uses
* LINUX ONLY
* Performance modes
  + General Purpose – default for 99.9% of uses
    - Latency sensitive
    - Web servers
    - Content management systems
  + Max I/O
    - Scale to high levels of aggregate throughput
    - Trade off of increased latency
    - Highly parallel applications
    - Big data
    - Media processing
    - Scientific analysis
* Throughput Modes
  + Bursting – default configuration
    - Burst pool, throughput scales with data
  + Provisioned
    - Specify throughput requirements separate from amount of data that you store
* Storage Classes
  + Standard - default
    - Frequently accessed files
  + Infrequent IA
    - Cost effective storage for inconsistently used data
* Lifecycle Policies can be used with classes

AWS Backup

* Fully managed, policy based data-protection (backup/restore) service – AWS and Hybrid
* Consolidate management into one place – across multiple accounts and multiple regions
* Supports a wide range of AWS products
  + Compute, Block, File, DB, and Object
    - Storage and Backup
* Backup plans – frequency, window, lifecycle, vault, region copy
* Resources – what resources are backed up
* Vaults – backup destination (container) – assign KMS for encryption – must configure at least one
* Vault lock – Write Once Read Many - WORM Mode
  + 72 hour cool off, then even AWS can’t delete
* On-demand – manual backups created as needed
* PITR – Point in time recovery
* Helps you support your regulatory compliance or business policies for data protection. With AWS Organizations, you can use AWS Backup to centrally deploy data protection policies to configure, manage, and govern your backup activity across your company’s AWS accounts and resources.