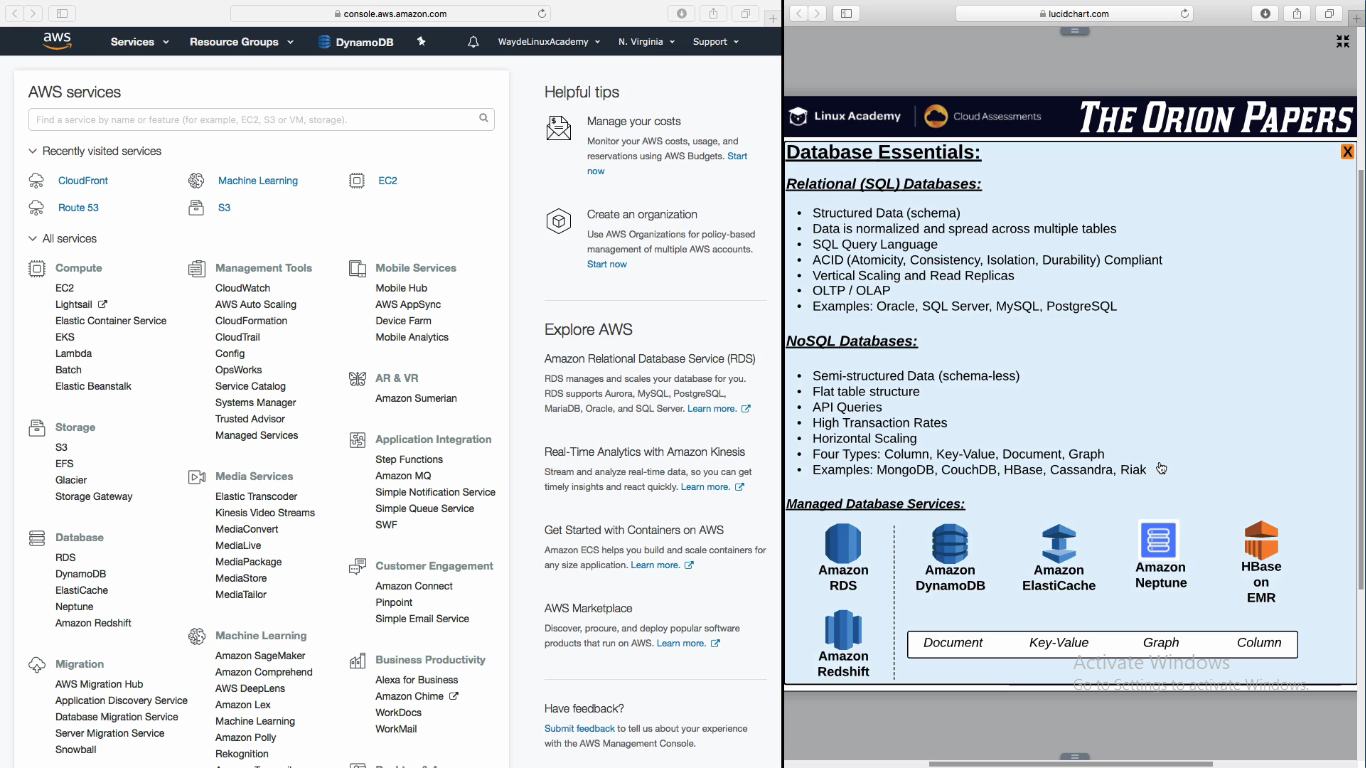
**Relational (SQL) DB**

They are decade old. Designed to run on single machine. Not good solutions for modern web scale app which have high transaction volume which single db can withstand

1. It scales vertically – adding more cpu, storage, memory and it have a limit.
   1. It makes a read replica is a read only copy
2. SQL query language to query data
3. Here we store data’s in multiple tables joined together
4. No ACID compliance. We have to role back ourself

**No SQL Database:**

1. Horizontally scale the cluster. They are not single instance and it is a cluter. Scale in by expanding cluster
2. Use proprietary API for the engine to query data
3. No concept of joining tables together. We store files in a flat table structure
4. Acid Compliance – Atomicity, Consistency, Isolation and Durability
   1. Issue command inside transaction
   2. If that command fails, DB can role back to the state before transaction

****

**Amazon RDS :** Relational DataBase service. Choice of DB services are oracle aql server, my sql etc

**Amazon Redshift : DataWare house service – optimized for analytics**

**Amazon DynamoDB :** Document Database

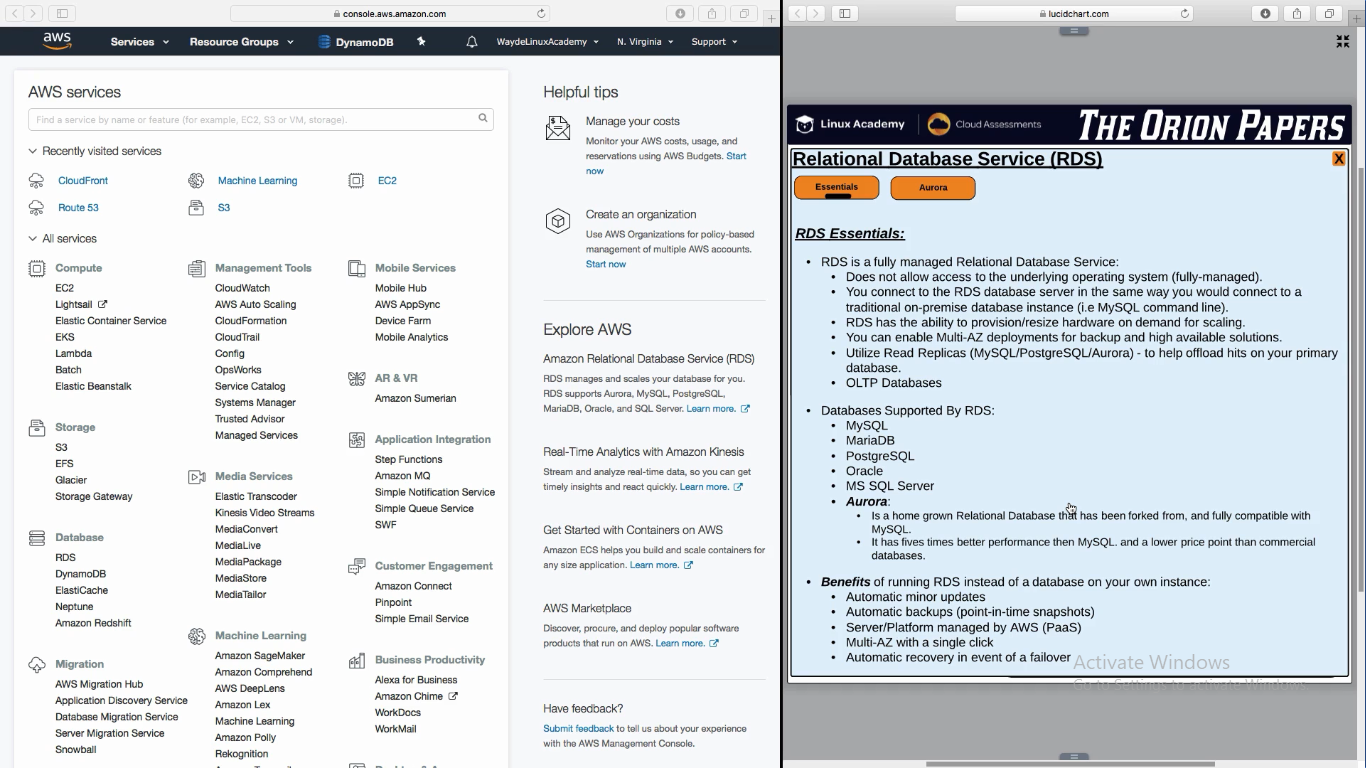
**Amazon ElastiCache :** Key value in memory database

**Amazon Nepture :** Graph Database

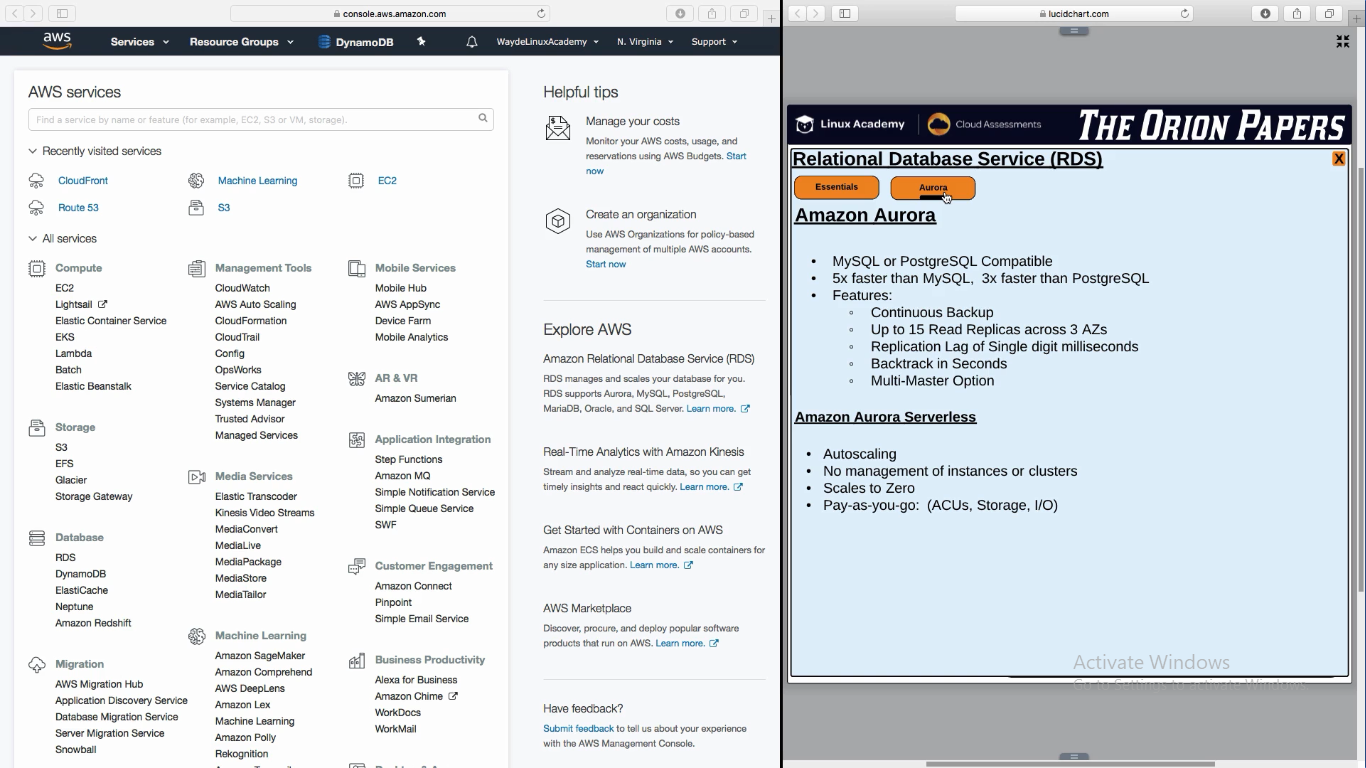
**HBase on EMR :** Column Database

**AMAZON RDS – Relational DataBase**

1. Database engine, security everything will be on the instance. Minor upgrades are done automatically. Major upgrades we will decide when they have to do on our behalf
2. Access we have is on DB level. We cannot ssh to the instance
3. License for SQL (Microsoft) will already be installed in that instance
4. We don’t want to configure access etc as like EC2



**Amazons home grown DB engine – AURORA**

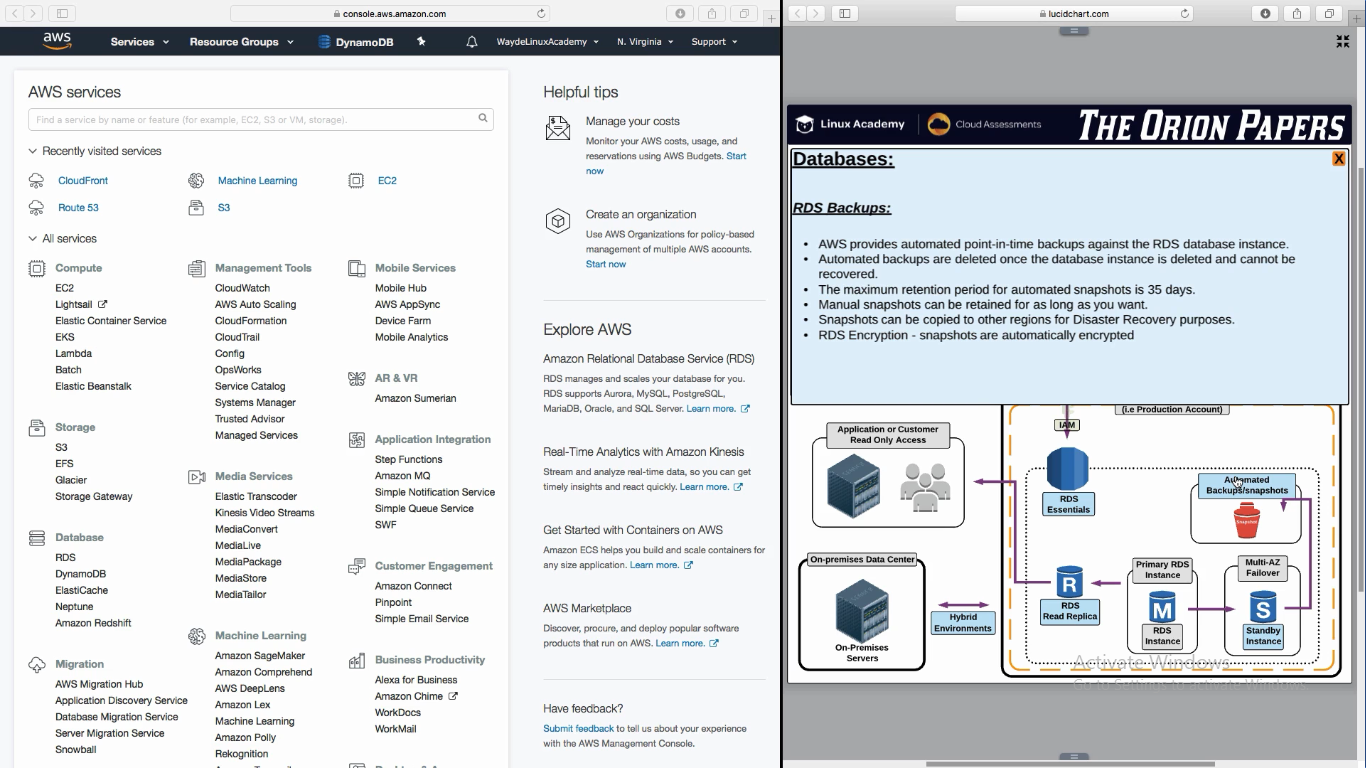
****

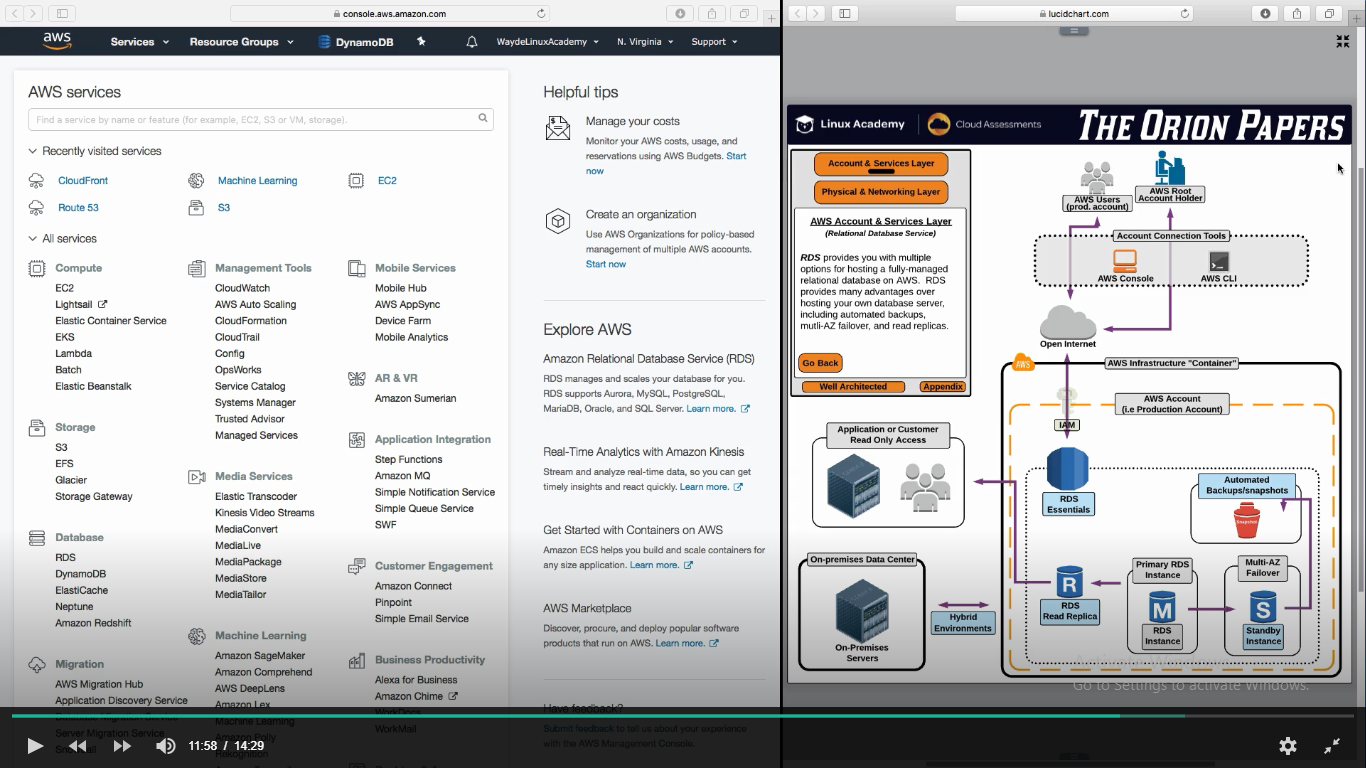
**RDS Backups**

Snapshots are valid till 35 days

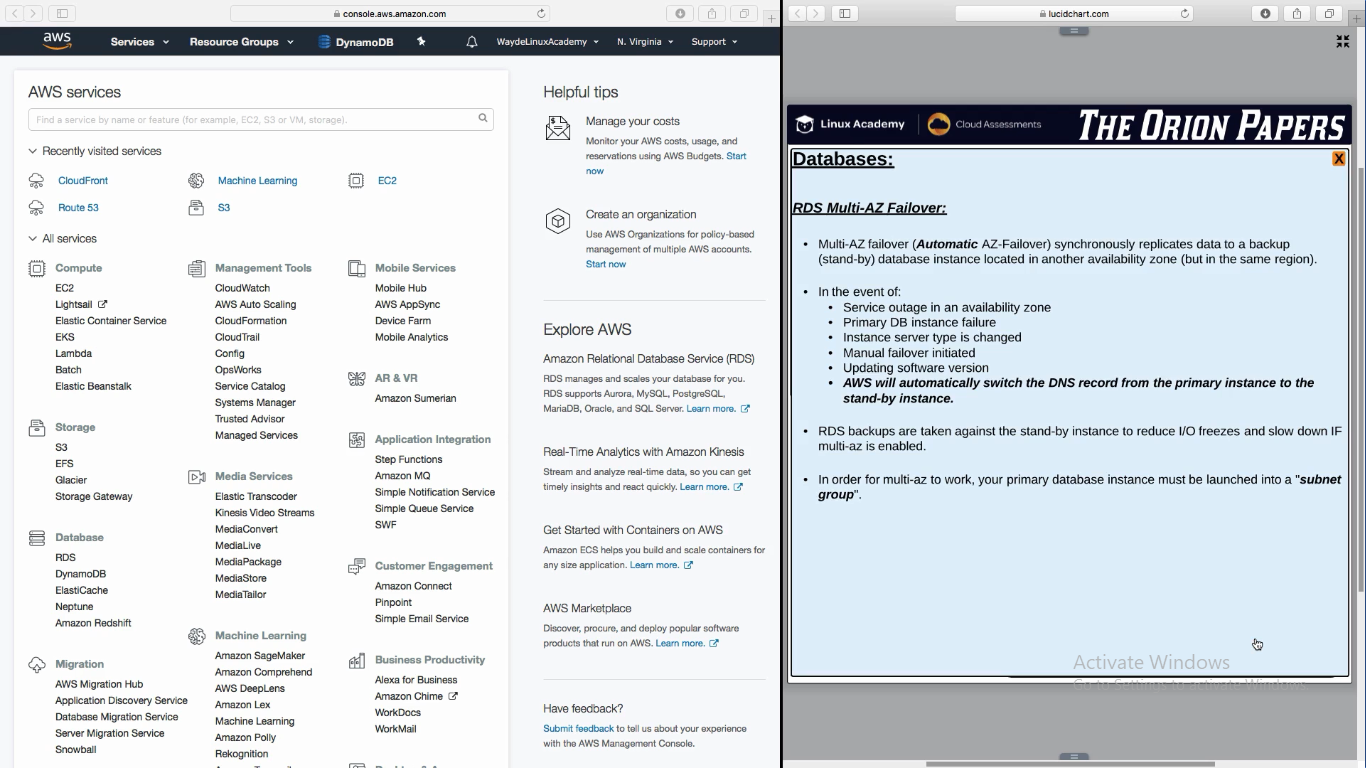
Take manual snapshot if we want more than 35 days. Copy it to a different region (Best practice and DR plan)

It have server side encryption

****



**RDS Multi AZ Failover:**

****

**Read Replica use –** Asynchronous replication

**RDS Multi AZ use –** Synchronous replication

We have a second copy of DB instance. It is in sync with primary. Secondary instance will be in different AZ than primary. We use DNS name to connect to DB (so if primary fails, secondary will become primary. We don’t want to change ip). This is when failover is automatic.

Snapshots are taken from seconday