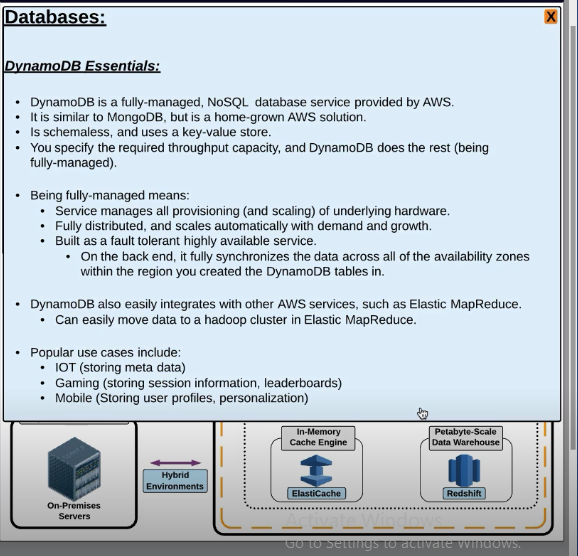
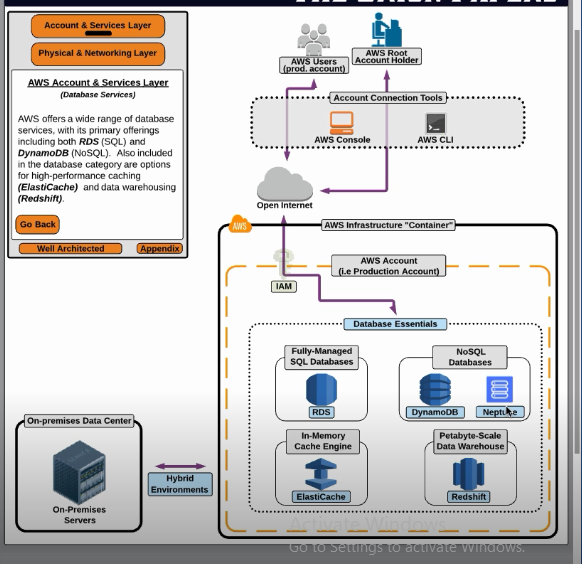
**No SQL Database**

**Dynamodb**

**DYNAMODB**

1. No SQL DB. It is serverless.
2. Serverless architecture
3. We can interact with API
   1. It includes putting, getting, querrying, create, drop, scanning the table
4. Scale out of in as per needs
5. It is schema less. We can modify a table without altering it. Each row can have its own column/attribute/item. Each items is a key value pair
6. Table is a collection of items and Item is a collection of key and values.
7. Capacity AWS provisions for dynamo db table depends on **storage and how many reads/writes** per second. We can definite it when we create a table
   1. We can create auto scaling. When we try to do more read, it scales out
8. DynamoDB makes multiple copies in AZ’x making data HA. It is fault tolerance
9. DynamoDB is a document type of DB. We can store json file in DynamoDB
10. Not a good example when we want to join multiple tables like Relational DB. DDB is a flat table
11. No limit on amount of data we store. No upper limit.
12. Session data can be stored in Dynamodb





**Handson**

1. AWS Console -> Database -> DynamoDB -> Create Table
2. Primary key
   1. Partition key : if this key is not primary create sort key. These two key combination will act as a unique key for the table

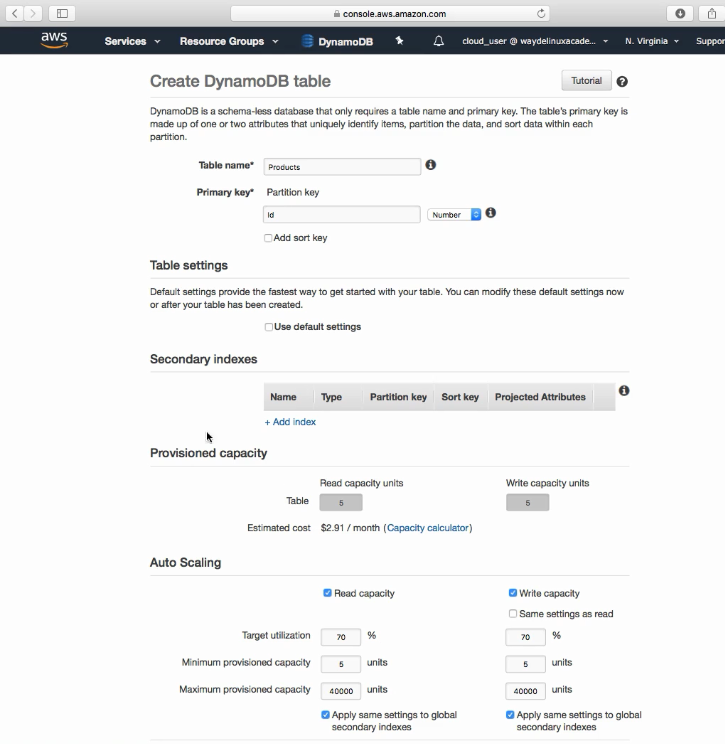
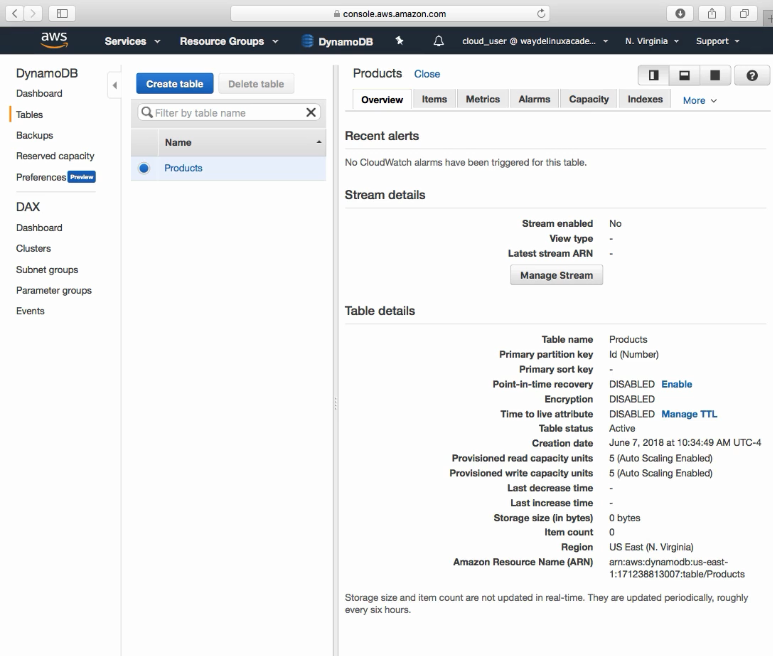
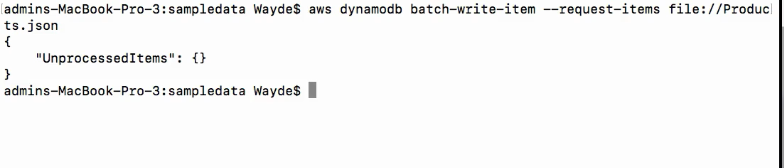


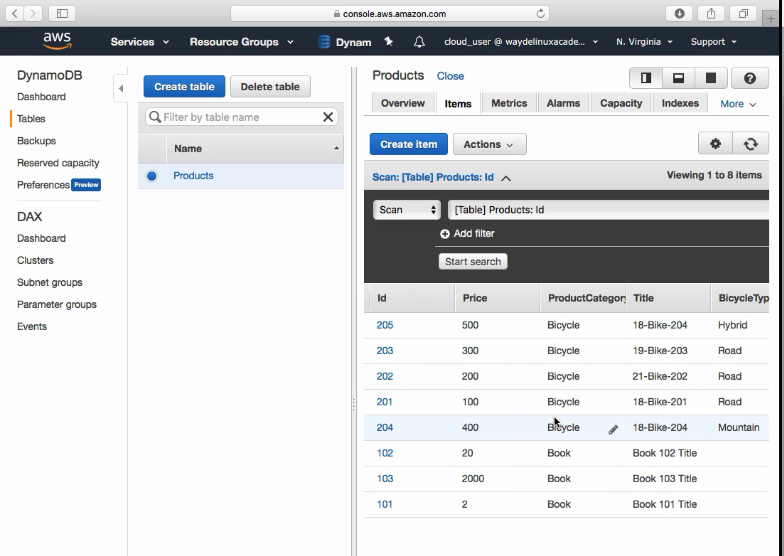
Table is created



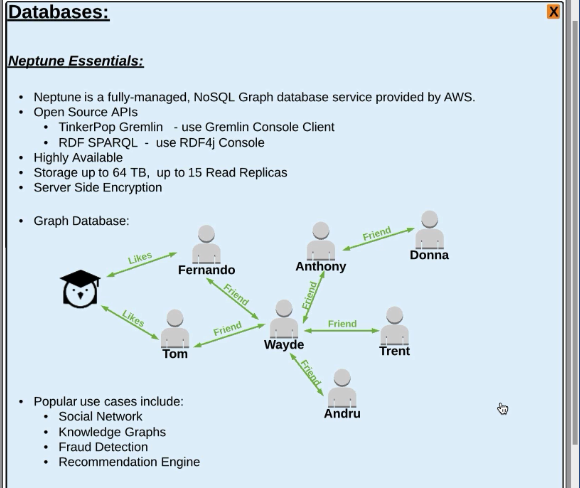
Importing JSON to the table



Items in the table after insertion

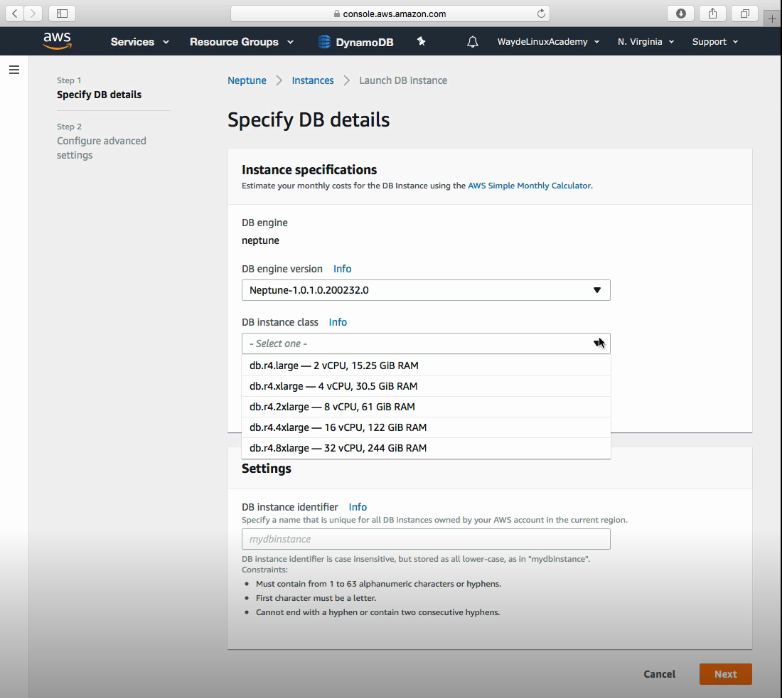


**Neptune**

****

It is not serverless. We have to configure instance and configure storage, read/write etc

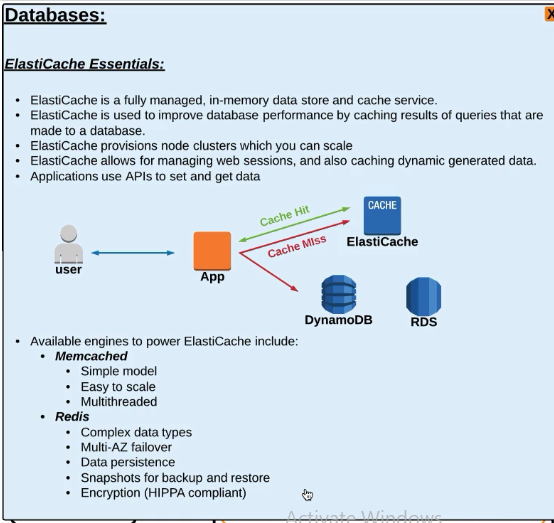
AWSConsole -> DataBase -> Neptune -> Launch Amazon Neptune



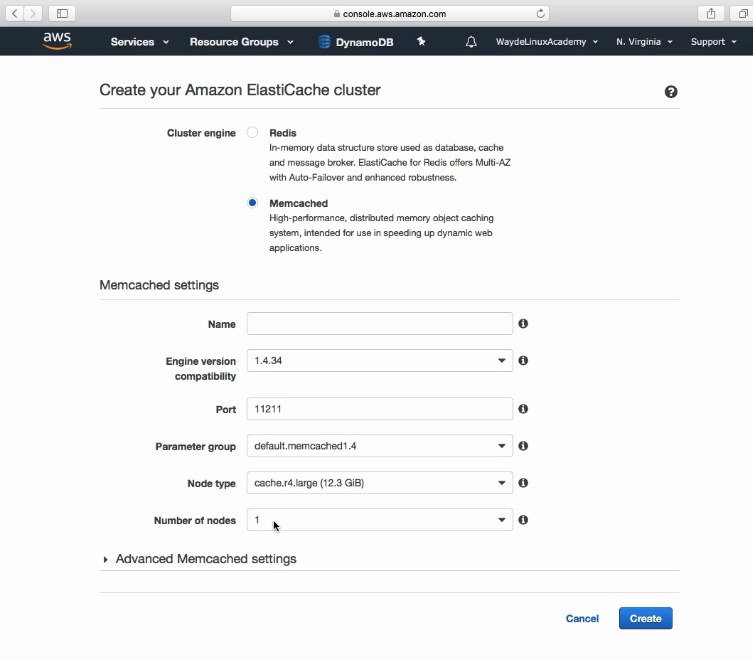
**Elastic Cache** – key/value

It is used to offload a DB (reduce load by memory caching the query result)

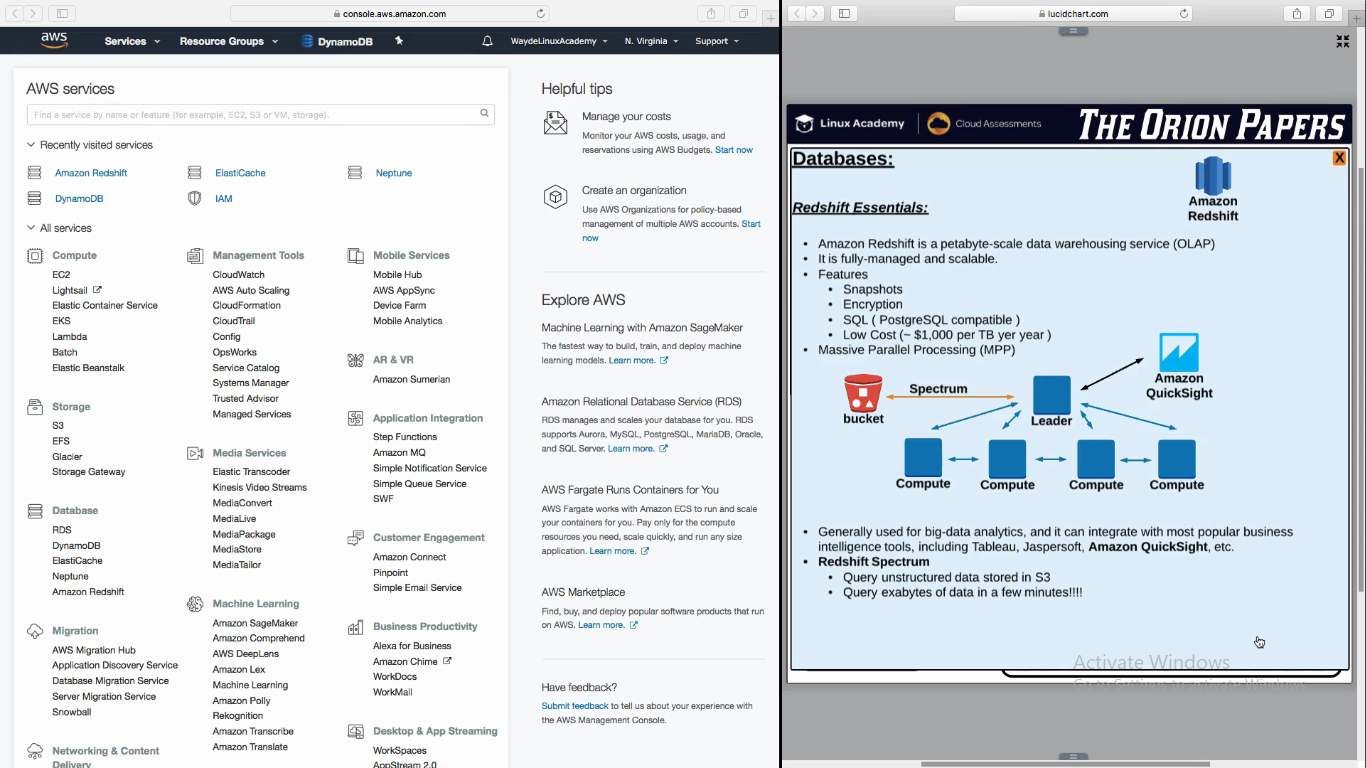
It stores the query value in memory. So that next time if someone use the same query and if the result is unaltered, it returns the value from memory without querying the database again



AWSConsole -> Database -> Elastic Cache -> LaunchCluster



RedShift



Spectrum : Extend query to process date from S3