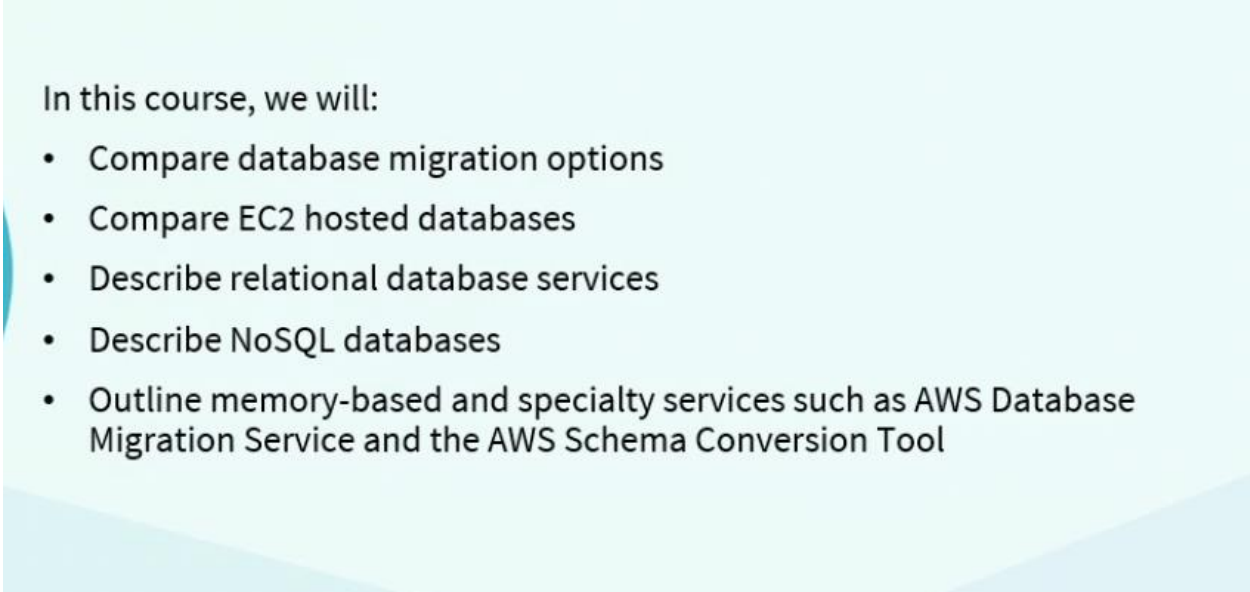
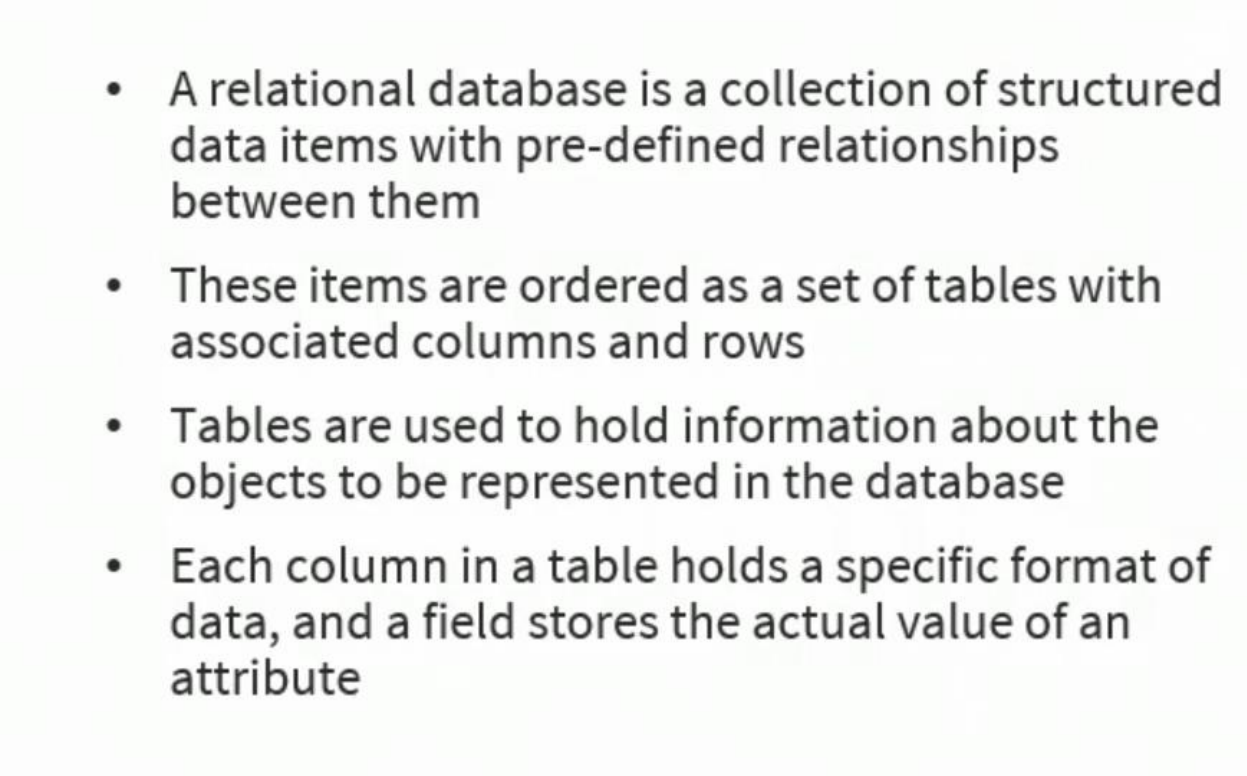
1.7 AWS Database Services



Relational Databases



- The rows in the table embody a collection of related values of one object or entity
- Each row in a table could be marked with a unique identifier called a primary key, and rows among multiple tables can be made related using foreign keys
- This data can be accessed in many ways without rearranging the database tables themselves

- Structured Query Language (SQL) is the main interface used to communicate with relational databases
- Data integrity is the general fullness, accuracy, and consistency of data
- Relational databases use a set of constraints called keys to enforce data integrity in the database

- A database transaction is one or more "all-or-nothing" SQL statements that are performed as a series of operations that establish a single logical unit of work
- To ensure data integrity, all database transactions must be ACID compliant:
  - Atomic
  - Consistent
  - Isolated
  - Durable

NoSQL Databases

- Are purpose-built for designated data models
- Have elastic schemas for constructing modern applications
- Are commonly known for their simplicity of development, functionality, and scalable performance

- NoSQL databases use a diversity of data models for accessing and controlling data
- These database types are optimized explicitly for applications that demand large data volume, low latency, and flexible data models
- This is accomplished by bypassing some of the data consistency restrictions of other databases, such as the relational type

NoSQL databases are an excellent solution for many modern applications like mobile, web, and gaming:

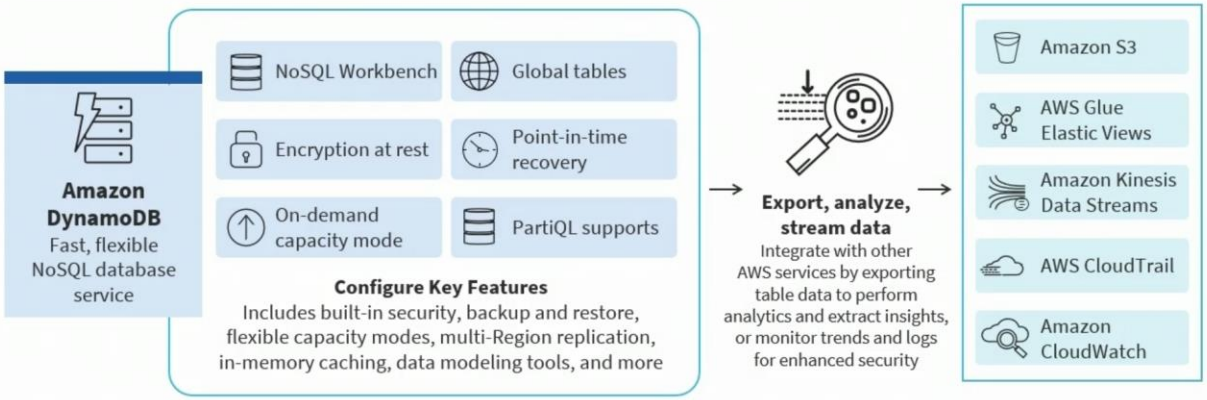
- **Flexibility** - provide flexible schemas that enable faster and more iterative development
- **Scalability** – generally designed to scale out by using distributed clusters of hardware instead of scaling up by adding expensive and robust servers

- **High-performance** – optimized for specific data models and access patterns that enable higher performance than relational databases
- **Highly functional** - offer highly functional application programming interfaces (APIs) and data types that are purpose-built for their respective data models

Amazon DynamoDB

- Is a fast, malleable NoSQL database service for single-digit millisecond performance at any scale
- Is a fully managed, serverless, key-value NoSQL database designed to run high-performance applications
- Delivers built-in security, nonstop backups, automated multi-Region replication, in-memory caching, and powerful data import and export tools

DynamoDB



Memory-Based Databases

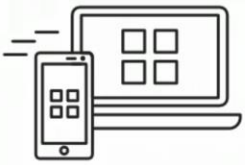
- In-memory databases are purpose-built databases that typically depend on high-speed memory chip clusters for data storage, as opposed to databases that store data on disk or solid-state drives (SSDs)
- In-memory data storage is intended to allow for nominal response times by abolishing the need to access physical disks

- Since all data is stored and managed exclusively in main memory, memory-based databases do risk losing data if there is a process or server failure
- This is often called ephemeral storage:
  - In-memory databases can persist data on disks by storing each operation in a log or in a snapshot
- In-memory databases are best for caching and applications that need microsecond response times or have big spikes in traffic, like gaming leaderboards, session stores, and real-time analytics

Amazon ElastiCache for Redis

- ElastiCache is an extremely fast in-memory data store that provides sub-millisecond latency to enable internet-scale real-time applications
- It is constructed on open-source Redis and compatible with the Redis APIs
- Self-managed Redis applications can function effortlessly with Redis ElastiCache without any code changes
- It joins the speed, ease, and flexibility of open-source Redis with manageability, security, and scalability from AWS

Amazon ElastiCache for Redis

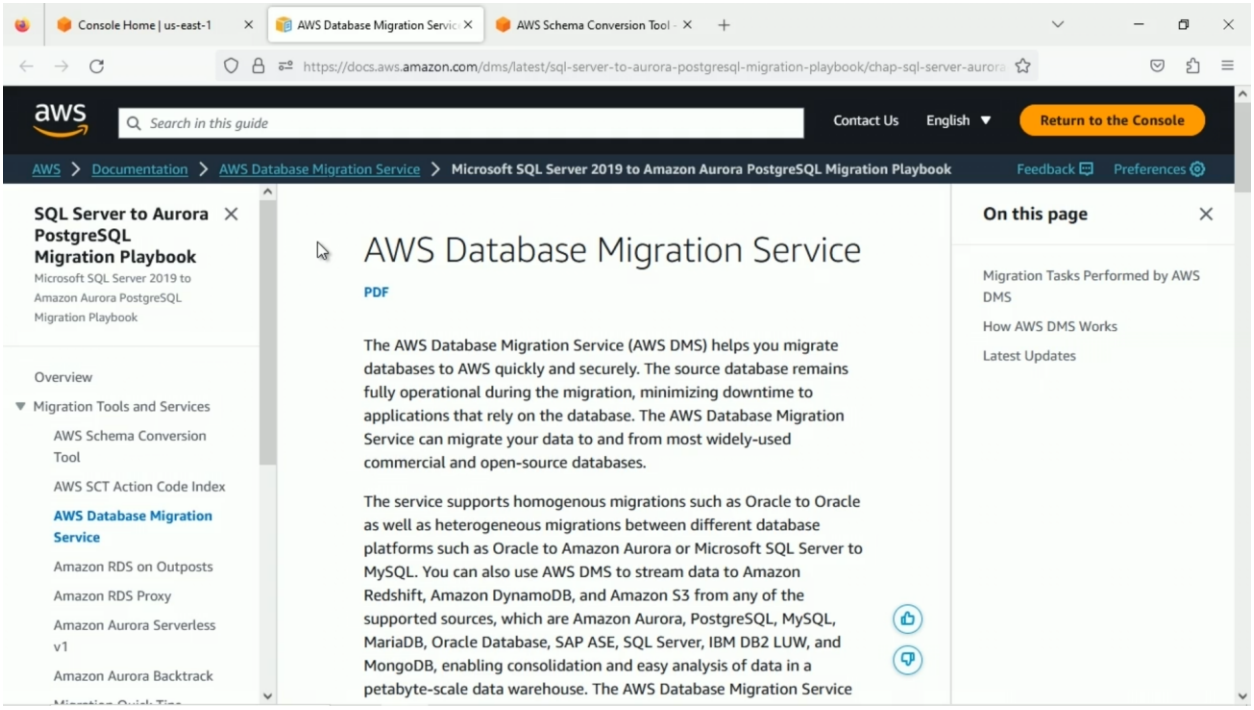


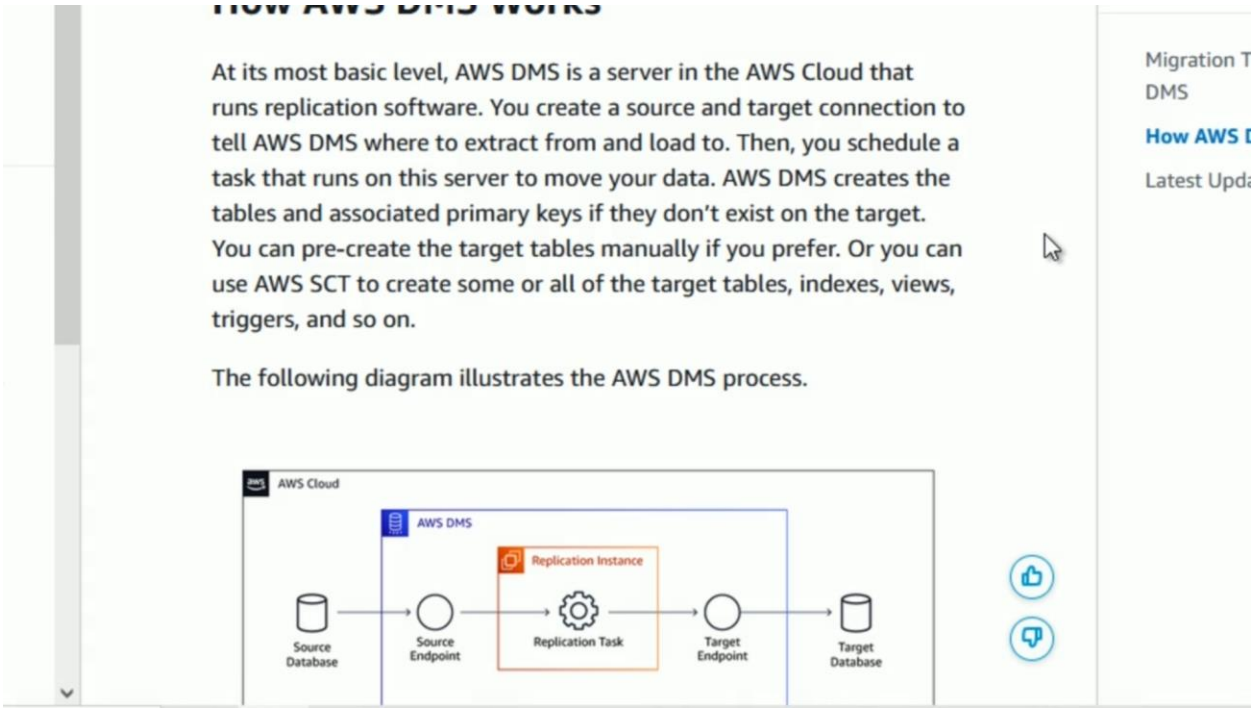
Internet-scale applications
Real-time apps in gaming, ride-hailing, media streaming, dating, and social media need fast data access

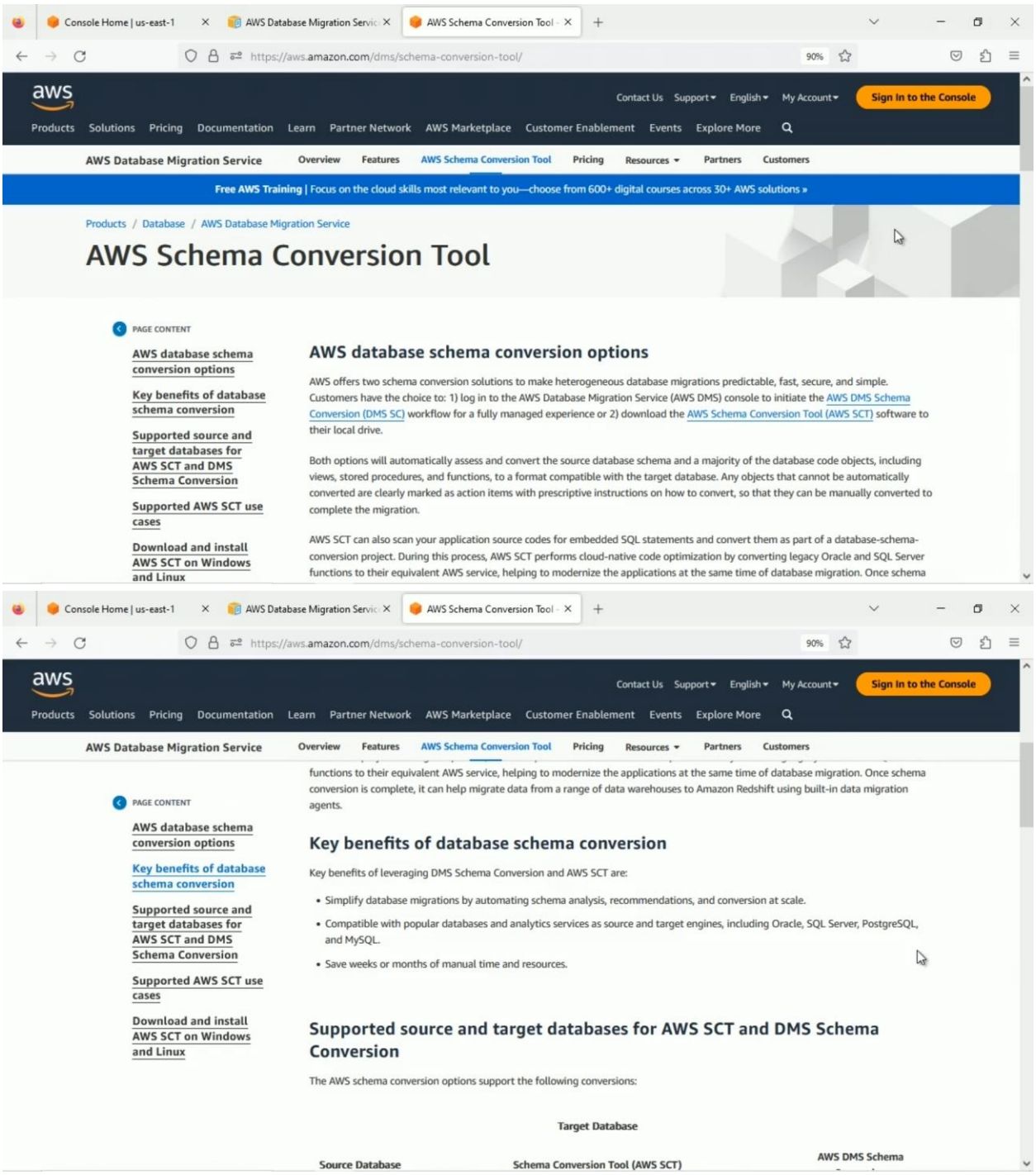
Amazon ElastiCache for Redis
Blazing fast in-memory data store for use as a database, cache, message broker, and queue; store ephemeral data in-memory for sub-millisecond response

Use cases
Real-time transactions, chat, BI and analytics, session store, gaming leaderboards, and cache

Comparing Database Migration Tools

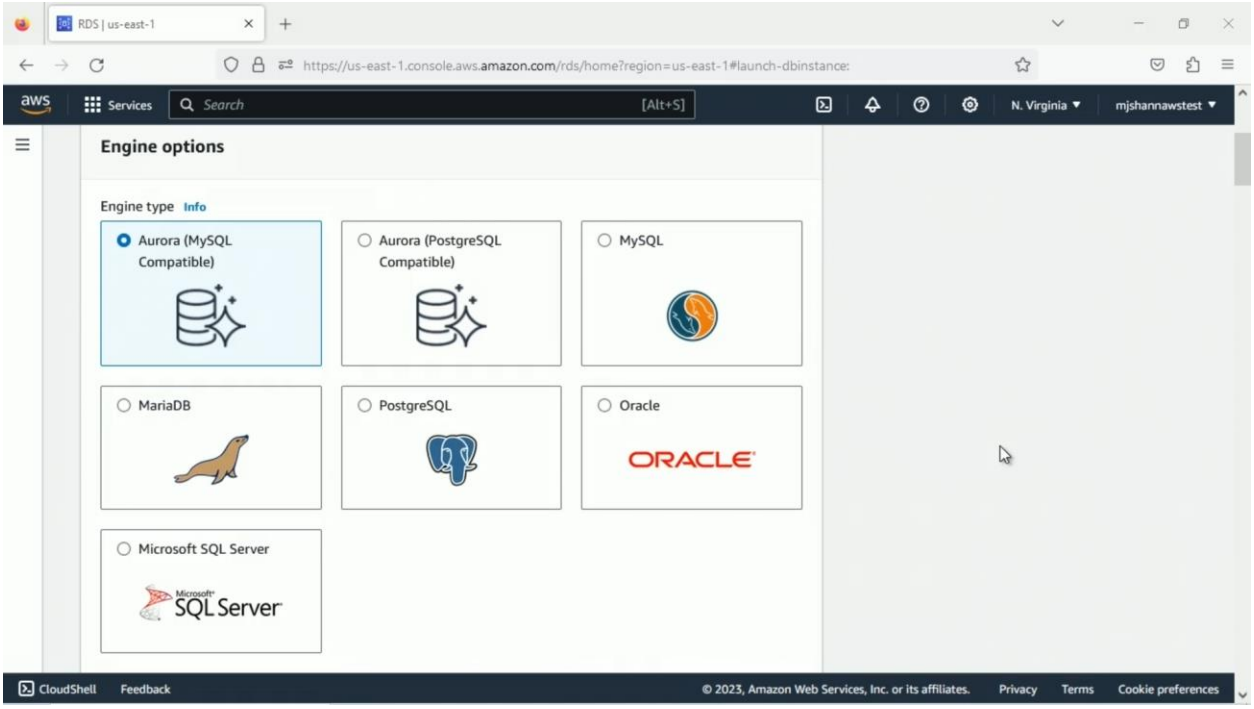




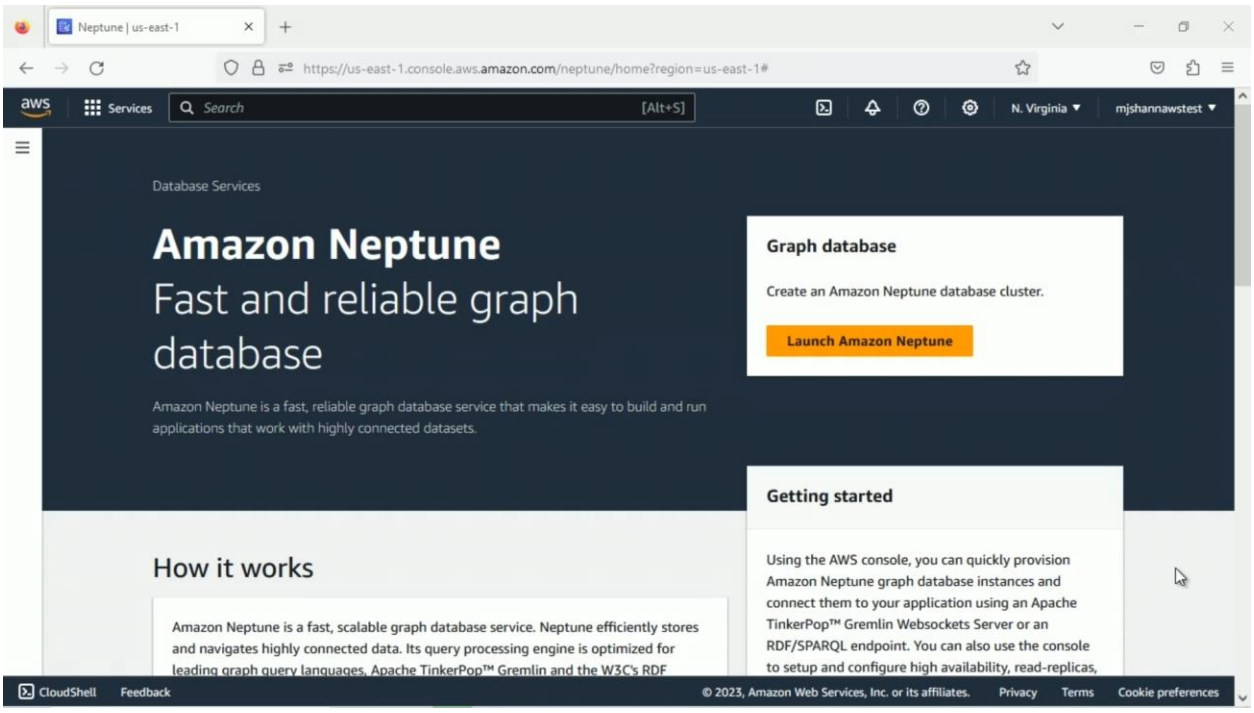


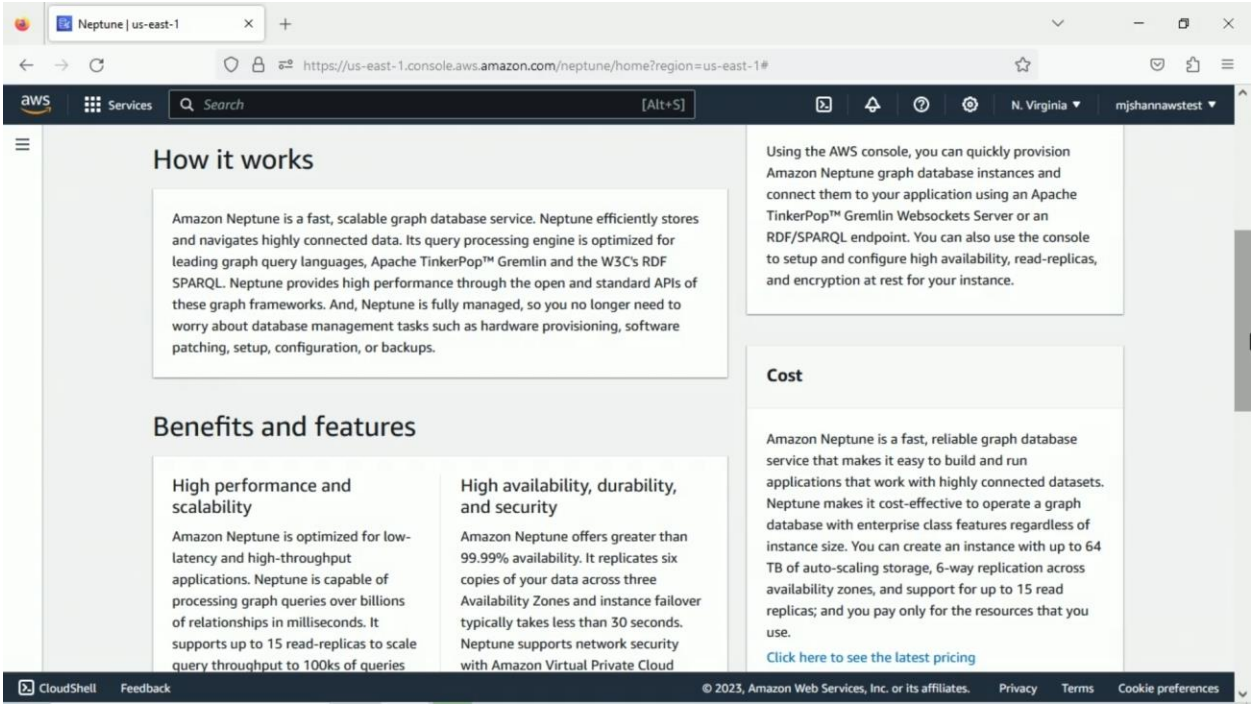
Comparing EC2 Hosted to AWS Managed Databases

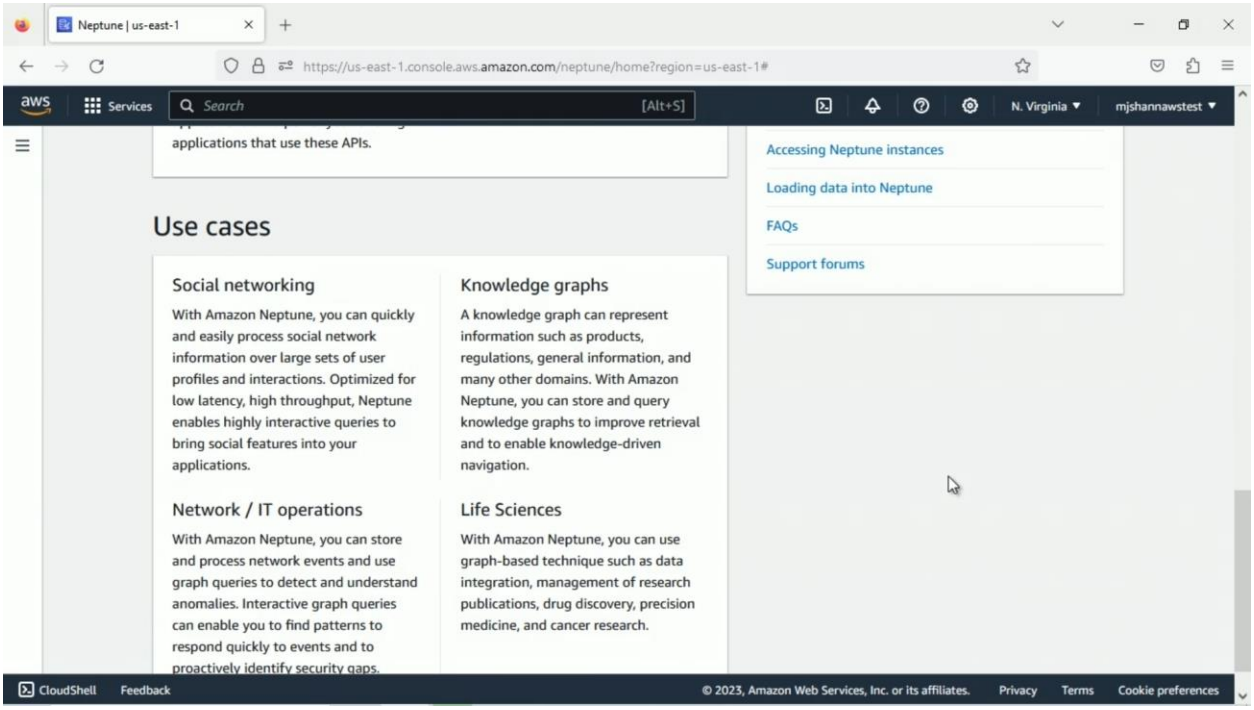
Using Amazon Aurora



Using Amazon Neptune




Neptune | us-east-1
https://us-east-1.console.aws.amazon.com/neptune/home?region=us-east-1#
aws
Services
Search
[Alt+S]
N. Virginia
mjshannawstest
How it works
Amazon Neptune is a fast, scalable graph database service. Neptune efficiently stores and navigates highly connected data. Its query processing engine is optimized for leading graph query languages, Apache TinkerPop™ Gremlin and the W3C's RDF SPARQL. Neptune provides high performance through the open and standard APIs of these graph frameworks. And, Neptune is fully managed, so you no longer need to worry about database management tasks such as hardware provisioning, software patching, setup, configuration, or backups.
Benefits and features
High performance and scalability
Amazon Neptune is optimized for low-latency and high-throughput applications. Neptune is capable of processing graph queries over billions of relationships in milliseconds. It supports up to 15 read-replicas to scale query throughput to 100ks of queries
High availability, durability, and security
Amazon Neptune offers greater than 99.99% availability. It replicates six copies of your data across three Availability Zones and instance failover typically takes less than 30 seconds. Neptune supports network security with Amazon Virtual Private Cloud
Using the AWS console, you can quickly provision Amazon Neptune graph database instances and connect them to your application using an Apache TinkerPop™ Gremlin Websockets Server or an RDF/SPARQL endpoint. You can also use the console to setup and configure high availability, read-replicas, and encryption at rest for your instance.
Cost
Amazon Neptune is a fast, reliable graph database service that makes it easy to build and run applications that work with highly connected datasets. Neptune makes it cost-effective to operate a graph database with enterprise class features regardless of instance size. You can create an instance with up to 64 TB of auto-scaling storage, 6-way replication across availability zones, and support for up to 15 read replicas; and you pay only for the resources that you use.
Click here to see the latest pricing
CloudShell
Feedback
© 2023, Amazon Web Services, Inc. or its affiliates.
Privacy
Terms
Cookie preferences


Neptune | us-east-1
https://us-east-1.console.aws.amazon.com/neptune/home?region=us-east-1#
aws
Services
Search
[Alt+S]
N. Virginia
mjshannawstest
applications that use these APIs.
Use cases
Social networking
With Amazon Neptune, you can quickly and easily process social network information over large sets of user profiles and interactions. Optimized for low latency, high throughput, Neptune enables highly interactive queries to bring social features into your applications.
Knowledge graphs
A knowledge graph can represent information such as products, regulations, general information, and many other domains. With Amazon Neptune, you can store and query knowledge graphs to improve retrieval and to enable knowledge-driven navigation.
Network / IT operations
With Amazon Neptune, you can store and process network events and use graph queries to detect and understand anomalies. Interactive graph queries can enable you to find patterns to respond quickly to events and to proactively identify security gaps.
Life Sciences
With Amazon Neptune, you can use graph-based technique such as data integration, management of research publications, drug discovery, precision medicine, and cancer research.
Accessing Neptune instances
Loading data into Neptune
FAQs
Support forums
CloudShell
Feedback
© 2023, Amazon Web Services, Inc. or its affiliates.
Privacy
Terms
Cookie preferences

1. What is the main interface used to communicate with relational databases?
   - **SQL**

2. What is a fully managed database service built for the cloud that makes it easier to build and run graph applications with built-in security, continuous backups, serverless compute, and integrations with other AWS services?
   - **Amazon Neptune**

3. What is an AWS migration and replication service that helps move your database and analytics workloads to AWS quickly, securely, and with little downtime supporting migration between several database and analytics engines such as Oracle to Amazon Aurora MySQL?
   - **AWS Database Migration Service**

4. When using Infrastructure-as-a-Service at AWS, what is the most common option for deploying Amazon Machine Images?
   - **EC2**

5. Which is an extremely fast in-memory data store that provides sub-millisecond latency to enable internet-scale real-time applications constructed on open-source Redis and compatible with the Redis APIs?
   - **ElastiCache**

6. What is a fast, fully managed, serverless, key-value NoSQL malleable NoSQL database service for single-digit millisecond performance at any scale designed to run high-performance applications?
   - **DynamoDB**

7. Which database platform is supported by Amazon Aurora?
   - **PostgreSQL**