

SQL Server, Kafka and Container Deployment

Integrated database, messaging and Linux deployment labs

FA-0737

DevOps, Cloud & Infrastructure · Software Development · Data & Analytics

Introduction

Develop practical skills for working with SQL Server and Apache Kafka in Linux-based container environments. Guided modules cover relational queries, database tooling, distributed messaging, client development and deployment considerations.

SQL Server 2019 provides the historical feature context; lab software and client tools are selected for compatibility and current support. The course is intended for learners with existing SQL and programming experience.

Learning outcomes

- Run and inspect containerised services, including ports, volumes and persistent data.
- Use SQL Server tables, queries, stored procedures and T-SQL in guided exercises.
- Explore database notebooks, visualisation, Python integration and migration considerations.
- Configure Kafka topics, partitions, replication and consumer groups in a controlled Linux lab.
- Build and test selected Java or Python producer and consumer examples.
- Explain how Docker and Kubernetes affect service networking, state and deployment.

Prerequisites

- Operational SQL knowledge and experience using SQL Server.
- Working knowledge of Java or Python; examples in the other language may be demonstrated.

- Basic Linux command-line and remote-access skills.
- A workstation and access to the designated compatible lab environment; SQL Server container exercises require a supported x86-64 Linux host.

Training outline

1. Module 1 – Docker and the Deployment Environment

- Overview of SQL Server, Kafka and container-based deployment.
- Docker setup appropriate to the workstation and designated Linux host.
- Images, containers, processes, ports, volumes and basic image/container commands.
- Run PHP/Apache and Nginx examples; map ports and volumes and inspect data persistence.
- Guided exercises and deployment-environment checks.

2. Module 2 – SQL Server on Linux

- Pull and run a compatible SQL Server Linux container image on the supported lab host.
- Connect using Visual Studio Code with the MSSQL extension, or SSMS where appropriate.
- Review SQL commands; create databases and tables, query data and use stored procedures and T-SQL.
- Explore selected SQL Server 2019 features in their version context rather than as the newest release.
- Introduce Kubernetes, big-data concepts and Apache Spark; distinguish their roles from the relational database.
- Practical database exercises.

3. Module 3 – Database Tools, Analysis and Migration

- Use supported SQL tooling and relevant extensions for queries and saved SQL work.
- Explore notebooks, data visualisation, Python integration and query-history workflows using compatible lab tools.
- Demonstrate a supplied machine-learning example and discuss its relationship to database data.
- Consider data portability, migration and related Spark/container technologies.
- Review retired SQL Server 2019 Big Data Clusters as historical architecture, not a currently supported deployment option.
- Exercises and evaluation.

4. Module 4 – Kafka Concepts and First Deployment

- Brokers, clusters, network ports, topics, records, partitions and producer/consumer architecture.
- Use KRaft metadata management for a compatible current Kafka lab; compare ZooKeeper ensembles as historical architecture.
- Prepare Ubuntu/Linux, remote development access and the required runtime; install an appropriate Apache Kafka distribution.
- Send and receive records with console tools; run multiple producers and consumers.
- Create partitioned topics, work with selected partitions and offsets, and inspect log storage.

5. Module 5 – Kafka Clusters, Replication and Consumer Groups

- Configure and start a multi-broker lab cluster; inspect supported administration tools, topic metadata and logs.
- Produce and consume against cluster endpoints; discuss multi-cluster boundaries without assuming automatic cross-cluster replication.
- Create replicated topics and observe replica placement, logs and storage.
- Simulate broker failures and observe recovery in the controlled lab.
- Use explicit consumer groups, inspect partition assignments and understand idle members.
- Assessment of cluster and consumer behaviour.

6. Module 6 – Performance and Client Development

- Run basic producer/consumer performance tests; compare configuration changes and inspect consumer lag.
- Set up Java clients; implement producers, serializers and multiple-producer examples.
- Implement consumers; distinguish consumer offset commits and partition assignment from producer configuration.
- Develop Python producer and consumer examples using a compatible client.
- Discuss container/Kubernetes networking, persistent storage and stateful service identity for database and messaging deployment.
- Hands-on development and assessment.