

BMC Helix CMDB Administration

24 Hours

Course Description

The BMC Helix Configuration Management Database (CMDB) is a critical component of IT Service Management (ITSM) and plays a pivotal role in managing an organization's assets, configuration items (CIs), and their relationships. This comprehensive course is designed to provide administrators and data management professionals with a deep understanding of BMC Helix CMDB, its architecture, configuration management principles, and data management processes.

Audience

This course is ideal for IT professionals, system administrators, data analysts, and anyone responsible for managing configuration data within an organization. It is suitable for individuals seeking to enhance their skills in BMC Helix CMDB administration, data normalization, reconciliation, and data federation.

Pre-requisite Knowledge/Skills

Participants should have a basic understanding of IT concepts and terminology. Familiarity with IT service management practices is beneficial but not mandatory.

Course Objectives

By the end of this course, participants will:

- Understand the fundamental concepts of Configuration Management and the Configuration Management Database (CMDB).
- Gain insight into the architecture of BMC Helix CMDB and the roles and responsibilities of key stakeholders.
- Explore the Common Data Model (CDM) and its relevance in CMDB data management.
- Learn how to access, navigate, and utilize the BMC Helix CMDB User Interface (UI).
- Master data import techniques using various tools, including Atrium Integrator and cmdbdriver.
- Comprehend the importance of data normalization and its role in maintaining data integrity.
- Acquire skills to create, monitor, and manage normalization and reconciliation jobs.
- Discover the concepts and methods of data federation in BMC Helix CMDB.

Course Outline

The course comprises 24 hours of theory and labs. It's divided into 7 different modules.

Module 1: Introducing the Configuration Management Database (CMDB)

- Introduction to BMC CMDB
- BMC Helix CMDB architecture
- Basics of BMC Helix CMDB
- Key roles and responsibilities
- Overview of the Common Data Model
- The concept of CIs
- Different consumers of BMC Helix CMDB data

Module 2: Accessing and Navigating the BMC Helix CMDB User Interface (UI)

- Accessing BMC Helix CMDB
- Navigating the BMC Helix CMDB Dashboard
- Searching and viewing CIs in the CMDB portal
- Creating a graph query
- Filtering CIs
- BMC Helix CMDB Dashboard features
- Creating and editing CIs

Module 3: Configuration Management and Common Data Model

- Introduction to Common Data Model (CDM)
- The relationships in the CDM
- Documentation and tools for researching the CDM
- Basics of BMC_BaseElement and BMC_Relationship classes
- Most commonly used classes of BMC_BaseElement Subclasses
- Purpose of BMC_BaseRelationship and its individual subclasses
- Navigate the Class Manager
- Viewing a class and a relationship class
- Creating and Modifying a class
- Auditing overview

Module 4: Importing Data

- Process to load data into BMC Helix CMDB
- Concept of Datasets
- Importing data using the Atrium Integrator
- Importing data using Atrium Integrator Spoon.
- Importing data using the Data Management console
- Importing data using cmdbdriver Program

Module 5: Normalizing Data

- Purpose of the Product Catalog
- How the Product Catalog supports multitenancy
- How the Product Catalog helps the BMC Helix CMDB to normalize data
- The role of the Normalization engine
- Normalization modes and process
- Normalization statuses
- How the normalization works with the BMC Helix CMDB to reconcile data
- Creating a normalization job
- Monitoring a normalization job

Module 6: Reconciling Data

- The purpose of reconciling CMDB data
- The structure of a reconciliation job
- BMC tools to reconcile the BMC Helix CMDB data
- How the reconciliation engine works with the BMC Helix CMDB to reconcile data
- Creating a reconciliation job
- Additional reconciliation activities

Module 7: Federating Data

- The concepts and methods of federation
- Identifying when to use the launch and retrieve method