Hewlett Packard Enterprise

HPE Digital Learner Microservices Architecture, Design Patterns, and Deployment Content Pack

HPE Content Pack number	CP024
Content Pack length	28 Hours
Content Pack category	Category 2
Learn more	**View now**

In this course, you will learn concepts of microservices architecture, design patterns for microservices architecture and its advantages and disadvantages. The content covers deployment strategies, tools for configuration, approaches for managing testing, load balancing, scaling out and common practices for securing and monitoring microservices architecture.

Why HPE Education Services?

- IDC MarketScape leader 5 years running for IT education and training*
- Recognized by IDC for leading with global coverage, unmatched technical expertise, and targeted education consulting services*
- Key partnerships with industry leaders OpenStack®, VMware®, Linux®, Microsoft®, ITIL, PMI, CSA, and SUSE
- Complete continuum of training delivery options—self-paced eLearning, custom education consulting, traditional classroom, video on-demand instruction, live virtual instructor-led with hands-on lab, dedicated onsite training
- Simplified purchase option with HPE Training Credits

Audience

IT professionals, developers, software engineers, cloud architects, DevOps and system administrators who are considering the use of service oriented architecture to enhance their service/release delivery solution

Content Pack objectives

This Content Pack provides the information necessary to:

- Describe past and present architectures for microservices
- Describe microservices deployment strategies and essential tools for configuration
- Explain how Docker and Kubernetes work with microservices
- Identify and select the right structural patterns to simplify design solutions
- Create reliable deployments using containers
- Evaluate options to control access to applications

*Realize Technology Value with Training, IDC Infographic 2037, Sponsored by HPE, October 2017

Detailed Content Pack outline

Exploring Microservices

Microservices Architecture

Microservices make it easier to build and maintain applications when they are broken down into several parts. Explore past and present architectures for microservices, including components, design patterns, and inter-process communication.

Outline

- Define a service and its purpose in a service oriented architecture (SOA)
- Identify microservices and their advantages
- Distinguish the architecture behind microservices
- Recognize various microservice processes
- Demonstrate microservices from use cases
- Identify various early versions of microservices
- Define the monolithic approach and the differences of using monolithic over SOA
- Recognize the benefits and costs of SOA
- Distinguish concepts in implementations of SOA
- Recognize early approaches including EAI and CORBA and SOAP
- Describe the process of event-driven architecture with event sourcing
- Describe the Command Query Responsibility Segregation and how to implement queries in a microservice architecture
- Describe various design patterns for microservice-based architectures
- Recognize SOA concepts with Dev Ops and continuous deployment and delivery
- Define dependencies in microservices
- Compare the differences between using modules and services
- Compare the importance of cohesion and coupling microservices
- Define the various communication processes including Direct Client-to-Microservices communication and API Gateway
- Recognize concepts in building IPC
- Recognize the various approaches to microservice architecture from past to present

---

Microservices Containers

There are several types of processes and packages that can be used to build microservice processes. Discover how to use Docker and Kubernetes with microservices.

Outline

- Define Docker and describe its components and relation to the microservice architecture
- Describe the benefits of using Docker with microservices
- Install and configure Docker
- Use the Docker GUI and toolbox
- Define containers in microservices
- Recognize how to use containers with microservices
- Define Dockerfile and how the tool is used to build and automate actions
- Use Docker to create a test database server
- Create a process using Docker to automate several processes
- Identify Kubernetes services
- Describe processes in connecting applications with services
- Configure a Kubernetes cluster
- Recognize how to use Docker with microservices

Microservices Deployment and Continuous Integration

In this course, you will explore deployment strategies for microservices and essential tools for configuration, including Spring Boot and REST. Also, discover how tools, such as Jenkins, can be used to assist in continuous integration (CI) development.

- Identify best practices in microservice deployment
- Describe various deployment implementations
- Recognize the use of service discovery and its benefits
- Describe various service discovery patterns
- Define Spring Boot and its use with microservices
- Describe how to install Spring Boot
- Define spring building with Maven and Gradle
- Define various Spring Boot auto configurations
- Recognize concepts in REST and HTTP architecture
- Define various common REST constraints
- Define continuous integration and how its mapped to microservices
- Describe the Jenkins interface using the Maven plugin
- Describe the process of building and composing containers for the Jenkins UI
- Recognize the process of testing continuous integration
- List benefits of continues and performance testing
- Describe various best practices for designing continuous delivery pipelines
- Recognize the differences between continuous integration and continuous delivery
- Recognize deployment architecture and RESTful services

Microservices Testing Strategies, Scaling, Monitoring, and Security

In this course, you will explore approaches for testing complexity in service architecture, load balancing, scaling out additional instances, and common practices for securing and monitoring microservices architecture.

- Define the various microservices tests including unit and service testing
- Define the scope of testing end to end
- Recognize the use of performance and cross-functional testing
- Describe the process in implementing service tests
- Define microservices scaling and the various methods
- Describe the process of scaling microservices databases
- Recognize the process of caching microservices
- Define autoscaling and how it is used
- Define various microservices monitoring tools
- Describe the process of monitoring a single server
- Describe the process of monitoring several servers
- Recognize various microservices logging tools
- Define log management and concepts applied to microservices
- Recognize the process of metric tracking across a span of servers
- Recognize general security practices with microservices
- Describe the processes of authentication and authorization
- Describe the process of fine grain authorization
- Describe service to service authentication and authorization
- Recognize the process of testing and scaling microservices end-to-end

Microservices Architecture

Microservices Deep Dive

In this course, you will take an in-depth look at architectures for microservices, including software design patterns, inter-process communication, automation, and microservices in the real world.

- Define what microservices are and what they are used for
- Compare the monolithic approach to the microservices one
- Describe the state of open-source approaches to microservices
- Describe the Azure Fabric approach to microservices
- Identify what polyglot persistence is and how it relates to microservices
- Distinguish when and why to use microservices
- List and describe common communication protocols used in microservices
- Describe key best practices for avoiding troubles with communication
- Explore the pros and cons of virtualization vs. containers for microservices
- Recognize how to implement automated testing in microservices
- Describe some real-world applications of microservices
- Recognize the key technologies and features used to create microservices

Microservices Architecture and Design Patterns

In this course, you will take an in-depth look at software design patterns for microservices. This includes pattern language for microservices, domain driven design, and strategies and best practices for creating successful microservices.

- Describe microservices architecture
- Compare microservices with enterprise architecture
- Describe some alternatives to microservices
- Identify key aspects of what software design patterns are
- Specify some of the recommended patterns on how to compose microservices together
- Describe what a pattern language is and how it applies to microservices
- List and describe the pattern areas recommended for use with microservices
- Specify how the patterns are applied and the typical decisions involved in creating microservices
- Describe how domain-driven design principles can be used with microservices
- Distinguish the ways that proper design can help prevent and deal with failure
- Specify some of the recommended fundamentals for creating useful and robust microservices
- Describe the pattern areas recommended for use with microservices

Microservices and Netflix

In this course, you will take an in-depth look at the Netflix microservices architecture in practice. This includes a look at components such as Eureka, Ribbon, Zuul, and Conductor.

- Describe how Netflix applies microservices to its solution architecture
- List and describe the various tools provided in the Netflix OSS
- Recognize how the plugins provided by Nebula are used
- Describe the key aspects of Spinnaker and its use
- Define how client-side service discovery via Spring Cloud Netflix Eureka works
- List the key features of the Netflix Ribbon library
- Describe the kept features of Netflix Archaius
- Detail the use of containers in Netflix
- Describe how Zuul provides dynamic routing, monitoring, resiliency, and security
- Recognize the role of Conductor in orchestrating various pieces of the Netflix puzzle
- Recall how event sourcing works for Netflix downloads
- Describe some of the key microservices used in Netflix

Microservices Current and Future Trends In this course, you will explore the current state of microservices. This includes coverage of tools and frameworks, current challenges, emerging trends, and some examples of microservice use in the real world.	• Describe the overall state of microservices development • List and describe some of the current security concerns of microservices • Compare the various tools and frameworks currently in use • Define what service meshes are and how they relate to microservices • Recognize some of the current challenges faced by microservices • Compare modularity with microservices	• Describe how microservices are expected to integrate with the IoT • Describe the role that microservices are expected to play in the digital future • Identify the emerging trends in application development • Recall how Amazon uses microservices with AWS • Recognize how microservices were implemented in a real world example • Recognize some of the current and future trends in microservices development

Microservices Design Patterns

Design Thinking Process This course covers how we can adopt design thinking process to identify an approach of leveraging the collective expertise to build solutions.	• Illustrate the significance of design thinking • Identify the design thinking processes • Describe the fundamental objective of design thinking • Apply the design thinking approach to building solutions • Analyze information to generate ideas with team collaboration	• Use the storytelling and analysis approach to build a problem solving process • Recognize peoples' roles and innovations • List customers' and end users' needs and expectations • Create defined and meaningful problem statements • Generate ideas around problem statements • Create prototypes to cover all key aspects, from the ideas to the solution • List and recognize the different types of prototypes • Describe how the design thinking approach is used in the software design process
Prototyping Design Thinking In this course, you will learn how to model prototypes in order to demonstrate the capabilities of systems and solutions.	• Build prototypes • Describe the test mode and its significance • Create and use process • Apply design thinking • Prototype new ideas	• Define business models • Define processes and the various states of process • Define business models and the impact of process states • Describe how and why prototypes are used
Need for Design Patterns and the Principles Guiding Design Patterns This course will focus on identifying common coding flaws and the application of object oriented principles in order to produce quality codes which adhere to the coding standards.	• Recognize the need for design patterns • Compare microservices and design patterns • Illustrate the history of patterns • Recognize the challenges that design patterns help simplify • Define design patterns • Define the core OOAD principles behind design patterns	• Recognize the practice and benefits of design patterns • Describe design patterns in context • Use class and sequence diagrams as notations to depict patterns • Use state machine diagrams as notations to depict patterns • Classify GoF (Gang of Four) pattern categories • List the important object oriented principles and the essential elements of patterns
Design Pattern Classification and Architectural Patterns This course covers the classification of design patterns. You will learn to recognize architectural patterns and apply these patterns in various coding scenarios.	• Classify architectural patterns • Classify creational patterns • Classify behavioral patterns • Classify structural patterns • Recognize when to use specification patterns • Recognize when to use MVP patterns • Recognize when to use MVVM patterns • Recognize when to use integrator patterns	• Recognize when to use lazy load patterns • Recognize when to use event aggregator • Recognize when to use service locator patterns • Recognize how patterns help simplify the right architectural decisions • Describe what a layered pattern is and the benefits of using such a pattern

Selecting Structural Patterns for Simplified Designs

Code simplification leads to a better and reusable design. This course covers different ways of identifying and selecting the right structural pattern to simplify design solutions.

- Recognize the complexities in a design and apply a simplification approach using structural patterns
- Recognize when to use adapter patterns
- Recognize when to use bridge patterns
- Recognize when to use composite patterns
- Recognize when to use decorator patterns
- Recognize when to use prototype patterns
- Recognize when to use proxy patterns
- Recognize when to use façade patterns
- List design challenges and solutions by proposing a structural pattern to implement

Managing Object Construction and Behavior Using Patterns

Object construction plays an important role in defining creational policies. In this course, you will evaluate the detailed and comparative benefits of using object construction and behavioral patterns to simplify logic.

- Recognize the need for various construction policies and apply simplification of logic using behavioral patterns
- Recognize when to use builder patterns
- Recognize when to use factory patterns
- Recognize when to use command patterns
- Recognize when to use interpreter patterns
- Recognize when to use mediator patterns
- Recognize when to use null object patterns
- Recognize when to use observer patterns
- Recognize when to use state patterns
- Recognize when to use strategy patterns
- Recognize when to use visitor patterns
- Identify AntiPattern scenarios and when to avoid them
- List a few design challenges and solutions by proposing a behavioral pattern to implement

Essential Deployment Microservices

Basic Concepts of a Microservices Architecture

In this course, you will learn the basic concepts of the microservices architecture. The properties surrounding a distributed architecture will be covered. The advantages and disadvantages of microservices will also be evaluated.

- Describe the properties that make up a microservice architecture
- Identify the different pieces of a microservice architecture and the function they serve
- Discover the advantages and disadvantages of a microservice architecture
- Identify the individual components of the microservice topology
- Define the level of granularity of microservice components
- Identify communication protocols such as RAC
- Describe how components of a microservice architecture communicate
- Build the API between microservices
- Identify compatibility and interoperability issues
- Use third-party systems for communication and logging
- Use communication in a microservice architecture

Core Concepts of a Service-Oriented Architecture

A service-oriented architecture and microservices are different. In this course, you will learn how a service-oriented architecture (SOA) differs from microservices and the advantages that SOA provides.

- Describe the hierarchical taxonomy of microservice systems
- Identify how microservices communicate and the protocols they use
- Identify the techniques in decoupling functionality in large monolithic services
- Define the problems in breaking down large systems into microservices
- Discover architectural design patterns
- Analyze dependencies between microservice components
- Describe the techniques for logging across the microservice architecture
- Determine how to manage database transactions between microservices
- Analyze distributed services and their reliability
- Define how vendor support is handled in a microservice architecture
- Determine how ownership of microservices is handled throughout the enterprise
- Analyze the API access service
- Analyze the direct access service
- Analyze access design hybrids
- Define how the remote access design pattern works
- Design a microservice

Designing the Microservices Architecture

There is more than one design for microservices architecture. In this course, you will learn about several architecture and design techniques that will provide a template for your microservices design phase.

- Describe how databases work in a microservice architecture
- Identify the methods for writing reports from log files
- Discover how event handling works
- Analyze consistency patterns and techniques
- Define the architecture of monolithic systems
- Analyze legacy architecture
- Identify the challenges in planning for microservice systems
- Describe the business justification for microservices
- Analyze the ways to work with parallel systems
- Define the issues when working with nonfunctional iteration management
- Discover new ways to work with the new architecture
- Use a microservice design

Microservice Refactoring Patterns

There are different ways to refactor. In this course, you will learn techniques to mitigate the risk and increase the success of the transformation from a monolithic architecture to a microservice architectural style

- Analyze the adaptation refactoring pattern
- Analyze the migration refactoring pattern
- Identify how hybrid refactoring is performed
- Identify which tools can be used in refactoring
- Describe the skillset needed for code rewrites
- Define the microservices life cycle
- Discover transformation alternatives to legacy code
- Describe how to retire and remigrate microservices
- Refactor legacy code

REST API & Microservices

Getting Started with REST and .NET Core

In this course, you will learn the basics of .NET Core and how to use it to create and test applications.

- Configure Visual Studio Code to work with .NET Core
- Identify REST design principles and patterns
- Use Postman to test and verify your REST services
- Design a simple REST service
- Configure routing for your REST application
- Set up a database connection for your REST application
- Install and set up Ninject and use it in applications
- Build a controller for your REST application
- Implement logging for your REST application
- Set up a smoke test of your application to verify functionality
- Set up a mocking framework for your API to run tests
- Build and run unit tests to ensure API functionality
- Build a simple REST API using .NET Core Web API

Building an API Project in ASP.NET

In this course, you will learn how to build a sample API project in ASP.NET using Web API, and how to create a fully functioning service from start to finish.

- Design a RESTful API for microservices
- Create models and design API URLs for accessing services
- Build models that will be used for applications
- Configure logging for your service and catch errors as they occur
- Connect to an external database in ASP.NET
- Set up Entity Framework to work with applications
- Create a database context to access data with microservices
- Create type mappings to improve the data integrity of a system
- Create and prepare a controller for microservice
- Configure an API with attribute-based routing for finer control over routing
- Implement dependency injection using Ninject
- Publish an application to a server
- Build a model and connect it to a database

Building a Microservice in .NET Core

In this course, you will learn the fundamentals of building a REST Microservice in .NET Core, from designing the API to deploying it.

- Design an API using .NET Core
- Create basic tests to validate functionality
- Design and build a model for a .NET Core API
- Use Rhino to inject a mock repository for testing and development
- Create a data store for an application
- Configure Entity Framework Core for use in .NET Core applications
- Create a database context to use in your applications
- Create read and update data using Entity Framework Core
- Use log4net to log events from your application
- Use Postman to verify the functionality of an API
- Deploy a microservice in a container for development or production
- Deploy a .NET Core microservice to a server
- Test an API with mocking and unit testing

Advanced Data Access with EF Core

In this course, you will explore the Entity Framework Core in depth, and learn how to add database connectivity to your .NET Core microservices.

- Connect to a sample Northwind database with .NET Core
- Configure and use SQLite for small database projects
- Use different methods to access data providers with EF Core
- Build attributes to fine tune your models
- Use and implement the EF Core Fluent API
- Build models with EF Core to satisfy application requirements
- Query and modify a model to retrieve data for an application
- Use LINQ queries to retrieve data
- Query and sort entities using LINQ and EF Core
- Work with filters to refine retrieved data
- Implement features of EF Core with sets
- Build and customize LINQ extension methods for specific applications
- Build a model with EF Core and query data from it

Deploying Your Microservice in Containers

In this course, you will explore containerizing and deploying your microservice with Docker. This creates reliable and repeatable deployments.

- Install Docker as a container manager for applications in Windows
- Use Docker components within Visual Studio
- Run a basic Docker image in Windows
- Build Docker images for development and deployment
- Manage and control Docker images using built-in tools
- Migrate image from development to your server
- Deploy a microservice to an image
- Utilize Microsoft Azure services to deploy a container
- Configure and use logging for your Docker images
- Monitoring images at any time using Docker
- Updating a microservice on an existing image
- Implement best practices for microservice lifecycle
- Build and run a simple image with Docker

Securing Your Microservices

Every application needs security. In this course, you will learn how to make sure that certain people can access your application and that you have control over what they can do.

- Configure options using cookie and forms authentication with services
- Use bearer tokens for authentication in your application
- Work with OpenID to establish secure authentication
- Use OAuth for authorization with your user accounts
- Implement individual accounts in your Web API
- Use ASP.NET Core Identity to establish user accounts in an application
- Use roles to authorize users to perform certain operations
- Use policies to authorize users to perform certain operations
- Secure applications against Cross-Site Request Forgery (CSRF)
- Implement Windows authentication for applications
- Deploy authorization filters to fine tune authentication schemes
- Use SSL to secure a Web API
- Implement a bearer token system in a Web API

Learn more at
www.hpe.com/ww/digitallearner

www.hpe.com/ww/digitallearner-contentpack

Interested in purchase of this Content Pack as a stand-alone WBT? Contact Us for information on purchasing this Content Pack for individual use.

Follow us:

CP024 A.00, January 2019

Hewlett Packard
Enterprise