

Learning objectives

After completing this course, you should be able to:

- Define cloud computing
- Describe the choices that are available to developers when creating cloud applications
- Describe infrastructure as a service, platform as a service, and software as a service
- Describe IBM Cloud
- Distinguish between the various IBM Cloud Platform compute options
 - IBM Cloud Foundry
 - IBM Cloud Kubernetes Service
 - IBM Cloud Function
 - Virtual Machines
 - VMware
- Understand the Twelve-Factor App methodology
- Describe how to create a Cloud Foundry application in IBM Cloud
- Describe how to create services in IBM Cloud
- Explain how to manage your IBM Cloud account with IBM Cloud CLI
- Deploy an application using IBM Cloud App Service (Web Apps)
- Explain containers and the difference between containers and Virtual Machines
- Describe container orchestration (Kubernetes)
- List Kubernetes key capabilities
- Realize the importance of using Kubernetes to prevent vendor lock-in
- Describe the Kubernetes building blocks:
 - Pod
 - Deployment
 - Service
- Describe DevOps
- Describe the capabilities of IBM Continuous Delivery
 - Identify the Web IDE features in IBM Continuous Delivery
 - Describe how to use Git Repos and Issue Tracking
 - Explain the pipeline build and deploy processes
- Create a Git repository to manage your source code
- View and edit code in the Eclipse Orion Web Integrated Development Environment (IDE)
- Build and Deploy code to IBM Cloud using DevOps
- Describe the characteristics of REST APIs
- Describe IBM Watson
- Provide examples of REST APIs using IBM Watson
- List the different database options available in IBM Cloud
- Explain the benefits of IBM Cloudant
- Describe how to design a cloud solution and enrich application using cloud services
- Explain the origin and purpose of the Node.js JavaScript framework
- Write a simple web server with Node.js
- Deploy an IBM SDK for Node.js application on an IBM Cloud account
- Create a Node.js module and use it in your code
- Understand asynchronous callbacks and be able to write the code in a Node.js application
- Create a Hello World Express application
- Use third-party modules in Node.js
- Use React to create interactive web pages
- Explain the core concepts of ECMAScript and React

Prerequisites Skills

- Basic knowledge of JavaScript.
- Basic understanding of HTML
- Basic understanding of the IBM Skills Academy learning environment

Duration

40 hours

Skill level

Basic – Intermediate

Hardware requirements

Classroom (ILT) setup requirements

Processor	2.5 GHz or faster Core 2 Duo
GB RAM	4 GB
GB free disk space	30 GB
Network requirements	Yes
Other requirements	Valid Email address

Notes

The following unit and exercise durations are estimates, and might not reflect every class experience. If the course is customized or abbreviated, the duration of unchanged units will probably increase.

Course Agenda

MODULE I – CLOUD APPLICATION FOUNDATIONS

Course I – Introduction to HTML5 and JavaScript Programming

Duration: 5 hours

Course introduction

Duration: 5 minutes

Unit 1. HTML5 overview

Duration: 45 minutes

Overview	This unit describes some of the overall goals and features of HTML5.
Learning objectives	After completing this unit, you should be able to: <ul style="list-style-type: none">• Describe what HTML does• List the objectives of HTML5• Describe the document types that are supported in HTML5• Describe the document object model (DOM) tree• Describe some of the differences between HTML4 and HTML5• List some HTML document API properties and methods• Describe how scripting is enabled in browsers• Describe browser support for HTML5 features

Unit 2. JavaScript programming for web applications

Duration: 1 hour and 30 minutes

Overview	This unit covers some of the essential features of JavaScript. JavaScript is an important programming language for building the next generation of interactive web applications, including mobile web applications. In this unit, you learn how to use JavaScript to manipulate the document object model elements of an HTML page.
Learning objectives	After completing this unit, you should be able to: <ul style="list-style-type: none">• Describe JavaScript primitives and objects• Explain how variables are declared and used in JavaScript• Describe JavaScript control structures• Describe functions in JavaScript• Describe the document object model (DOM) hierarchy• Describe the window and document objects• Identify the DOM objects that are commonly used in JavaScript applications for working with HTML documents

Exercise 1. Working with JavaScript in HTML documents

Duration: 45 minutes

Overview	In this exercise, you work with the Web Application Server Developer Tools for Eclipse environment to develop HTML documents and JavaScript functions.
Learning objectives	After completing this exercise, you should be able to: <ul style="list-style-type: none">• Create HTML web pages• Use style statements in HTML documents• Connect scripts to documents• Write JavaScript functions• Create interactive alert and confirm window objects• Use JavaScript to modify the document object model (DOM)

Unit 3. HTML5 features

Duration: 45 minutes

Overview	This unit covers the new structural and form features of HTML5.
Learning objectives	After completing this unit, you should be able to: <ul style="list-style-type: none">• List new elements in HTML5• Describe HTML5 structural elements: section, article, header, footer, figure, figcaption• Describe the attributes of the HTML5 input element: tel, email, datetime, number, range, color

Exercise 2. Working with HTML5 features

Duration: 45 minutes

Overview	In this exercise, you explore some of the features available in HTML5.
Learning objectives	After completing this exercise, you should be able to: <ul style="list-style-type: none">• Create a web page and insert a simple HTML5 form layout• Add new markup elements• Use input types that include attributes such as email to perform client-side validation• Test the application

Unit 4. Course summary

Duration: 5 minutes

Overview	This unit provides a summary of the course, a description of the class evaluation process, and information for future study.
Learning objectives	After completing this unit, you should be able to: <ul style="list-style-type: none">• Explain how the course met its learning objectives• Submit an evaluation of the class• Identify other Web Application Server Education courses that are related to this course• Access the Web Application Server Education website• Locate appropriate resources for further study

MODULE II – CLOUD APPLICATION DEVELOPER

Course I – Essentials of Cloud Application Development

Duration: 19 hours

Course introduction

Duration: 15 minutes

Unit 1. Introduction to cloud computing

Duration: 60 minutes

Overview	This unit provides an overview about cloud computing. It lists characteristics and benefits of cloud computing and describes cloud computing service and deployment models.
Learning objectives	After completing this unit, you should be able to: <ul style="list-style-type: none">• Define cloud computing.• Describe the characteristics of cloud.• Describe the benefits of cloud and the factors that contribute to its growth.• Describe cloud services models (IaaS, PaaS and SaaS).• Describe the cloud deployment options (Private, Public, Hybrid).• Describe cloud native applications and development methods.• Explain the Twelve-Factor App methodology• Describe the choices that developers have when building cloud applications.

Unit 2. Introduction to IBM Cloud

Duration: 120 minutes

Overview	This unit gets you started with IBM Cloud. It provides an overview of IBM Cloud services and the type of applications you can build on IBM Cloud. It explains how to create and manage cloud apps on IBM Cloud and how to manage users and resources.
Learning objectives	After completing this unit, you should be able to: <ul style="list-style-type: none">• Describe IBM Cloud.• Identify the runtimes and services that IBM Cloud offers.• Distinguish among the various compute options on IBM Cloud.<ul style="list-style-type: none">○ IBM Cloud Foundry○ IBM Cloud Kubernetes Service○ IBM Cloud Functions○ Virtual Machines○ VMware• Describe IBM Cloud regions, zones and multi-availability zones.• Describe the IBM Cloud dashboard, catalog, and documentation features.• Work with IBM Cloud resources.• Explain starter kits and Cloud Foundry boilerplates.• Describe how to manage your IBM Cloud users and resources (Optional).• Explain Identity and Access Management (IAM) and Resource Groups (Optional).• Describe how the application route is used to test an application in the browser.• Bind services to an application in IBM Cloud.• Describe the environmental variables that are used with IBM Cloud services.• Explain function as a service.

Unit 3. Deploying applications to Cloud Foundry on IBM Cloud

Duration: 60 minutes

Overview	This unit introduces Cloud Foundry and describes how to deploy applications to Cloud Foundry on IBM Cloud by using the IBM Cloud CLI. This unit describes the basic structure of Node.js apps.
Learning objectives	After completing this unit, you should be able to: <ul style="list-style-type: none">• Explain Cloud Foundry basic concepts, including:<ul style="list-style-type: none">◦ Organizations and spaces◦ Buildpacks◦ Resiliency◦ Logging and debugging◦ Domains and routes◦ Services binding• Explain the capabilities of the IBM Cloud command-line interface (CLI).• Describe the structure of the sample Node.js application.• Deploy the sample Node.js application by using the IBM Cloud CLI

Exercise 1. Getting started with Cloud Foundry apps on IBM Cloud

Duration: 75 minutes

Overview	This exercise describes how you can deploy a web app without downloading or configuring a runtime environment, or framework or setting up a server. This exercise also covers how to test and run the app when it is deployed.
Learning objectives	After completing this exercise, you should be able to: <ul style="list-style-type: none">• Create an IBM Cloud application by using one of the available run times.• Install the IBM Cloud command-line interface (CLI).• Sign on to IBM Cloud from the CLI.• Deploy an application from a local workstation by using the IBM Cloud CLI.• Test the application with its endpoint after the application is deployed and started.

Unit 4. Adopting a DevOps approach by using IBM Continuous Delivery

Duration: 90 minutes

Overview	This unit introduces the features and functions of the DevOps services on the cloud development platform, IBM Cloud.
Learning objectives	After completing this unit, you should be able to: <ul style="list-style-type: none">• Describe DevOps.• Describe the capabilities of IBM Cloud Continuous Delivery.• Identify the web-based integrated development environment (Web IDE) features in IBM Cloud Continuous Delivery.• Describe how to use source code management (such as Git) and Issue tracking.• Explain how to build and deploy applications using DevOps tools on IBM Cloud.

Exercise 2. Developing IBM Cloud applications with IBM Cloud Continuous Delivery**Duration: 90 minutes**

Overview	In this exercise, you will explore DevOps capabilities in IBM Cloud. You work with the IBM Cloud Continuous Delivery services to explore, develop, build, and deploy IBM Cloud applications.
Learning objectives	After completing this exercise, you should be able to: <ul style="list-style-type: none">• Enable an application to use IBM Cloud Continuous Delivery.• Create a Git repository to manage source code• View and edit code in the Eclipse Orion Web Integrated Development Environment (IDE).• Build and deploy code to IBM Cloud.• Test the application in IBM Cloud.

Unit 5. REST architecture and Watson APIs**Duration: 90 minutes**

Overview	This unit introduces Representational State Transfer (REST) and Resources Representation and JavaScript Object Notation (JSON). It describes how to apply REST architecture concepts to server-side applications. This unit introduces Watson services and provides examples that show how to call Watson services by using REST APIs.
Learning objectives	After completing this unit, you should be able to: <ul style="list-style-type: none">• Describe the main characteristics of REST APIs.• Explain the REST architecture style for designing networked applications.• List best practices to follow when using REST in your application.• Describe the representation format of data in REST.• Explain the advantages of the JSON data format.• Describe the security options for REST in your application• Describe IBM Watson services in IBM Cloud• Provide examples of REST APIs using IBM Watson.

Unit 6. Introduction to data services on IBM Cloud**Duration: 90 minutes**

Overview	This unit provides an overview of the types of data stores that are used in cloud computing. You will also learn about the data services offerings that are available through the cloud development platform.
Learning objectives	After completing this unit, you should be able to: <ul style="list-style-type: none">• Describe different databases types and capabilities• Describe the main types of data services in IBM Cloud.• Explain the benefits of IBM Cloudant.• Access Cloudant databases and documents on IBM Cloud.• Use HTTP APIs to interact with Cloudant database.

Exercise 3. IBM Cloud with Cloudant**Duration: 45 minutes**

Overview	This exercise demonstrates how you can create a Cloudant database service on IBM Cloud without installing or configuring the database instance on your workstation. You use an HTTP API client such as Postman to create, read, update, and delete Cloudant documents. You create indexes and query data by using Cloudant API end points.
Learning objectives	After completing this exercise, you should be able to: <ul style="list-style-type: none">• Create an instance of the Cloudant service on IBM Cloud.• Create service credentials by using IBM Cloud Identity and Access Management (IAM)..• Access the Cloudant documentation.• Explore the features of the Cloudant dashboard.• Create, read, update, and delete Cloudant documents by using HTTP APIs.• Verify the data that is stored in the database by using the Cloudant dashboard.• Create indexes and query Cloudant documents by using HTTP APIs.

Unit 7. Enriching your application with IBM Cloud services**Duration: 60 minutes**

Overview	This unit presents an example of a cloud application to solve a business problem. It introduces functional and non-functional requirements, application architecture, and services on IBM Cloud that you can integrate with your application logic to implement a solution.
Learning objectives	After completing this unit, you should be able to: <ul style="list-style-type: none">• Explain functional and non-functional requirements.• Design a simple architecture for cloud applications.• Identify services listed in the IBM Cloud catalog that you can integrate in your applications, such as:<ul style="list-style-type: none">◦ Database◦ App ID◦ Watson Natural Language Understanding◦ Watson Tone Analyzer◦ Log Analysis◦ Monitoring• Integrate cloud services in a Node.js app by using REST APIs.

Exercise 4. Securing a web application with single sign-on (optional)**Duration: 60 minutes**

Overview	In this exercise, you secure an application by using the App ID service for single sign-on by authenticating your application through trusted server providers.
Learning objectives	After completing this exercise, you should be able to: <ul style="list-style-type: none">• Create an App ID service.• Bind the App ID service to an application to add single sign-on capability.• List and implement different configuration options for the App ID service.

Unit 8. Developing containerized applications on Kubernetes**Duration: 120 minutes**

Overview	This unit introduces containers and containers orchestration. It provides an overview of the Kubernetes platform and describes basic concepts such as Kubernetes architecture, Kubernetes objects, and management of Kubernetes objects.
Learning objectives	After completing this unit, you should be able to: <ul style="list-style-type: none">• Explain containers and the difference between containers and virtual machines.• Describe container orchestration.• List the key capabilities of Kubernetes.• Articulate the importance of using Kubernetes to prevent vendor lock-in.• Describe the Kubernetes building blocks:<ul style="list-style-type: none">◦ Pod◦ Deployment◦ Service• Scale and auto-scale your deployment for high availability.

Unit 9. IBM Cloud Kubernetes Service overview**Duration: 60 minutes**

Overview	This unit introduces IBM Cloud Kubernetes Service and describes how a Kubernetes cluster lets you securely manage the resources that you need to deploy, update, and scale applications.
Learning objectives	After completing this unit, you should be able to: <ul style="list-style-type: none">• Explore IBM Cloud Kubernetes Service on Kubernetes platforms.• Create a Kubernetes cluster by using the IBM Cloud Kubernetes Service.• Create containers and build on the IBM Cloud Container Registry.• Use Vulnerability Advisor to scan for vulnerabilities.• Manage Kubernetes clusters by using the kubectl command-line interface.• Deploy an application from a local workstation by using kubectl.

Exercise 5. Managing IBM Kubernetes Service clusters**Duration: 30 minutes**

Overview	This exercise demonstrates how to create an IBM Kubernetes Service cluster and manage it by using the kubectl CLI.
Learning objectives	After completing this exercise, you should be able to: <ul style="list-style-type: none">• Create an IBM Kubebernetes Service cluster.• Connect to a cluster on IBM Cloud Kubernetes Service.• List the worker nodes in a cluster.

Exercise 6. Deploying an application on Kubernetes

Duration: 90 minutes

Overview	In this exercise, you build a containerized application and deploy it to IBM Cloud Kubernetes Service.
Learning objectives	<p>After completing this exercise, you should be able to:</p> <ul style="list-style-type: none">• Create a containerized Node.js application and build it on IBM Cloud Container Registry.• Explain how the container security analysis capability of Vulnerability Advisor can identify security vulnerabilities by scanning an image.• Create a deployment and scale it.• Expose your application on the internet.

Course II – Developing Node.js Applications on IBM Cloud

Duration: 11 hours

Course introduction

Duration: 15 minutes

Unit 1. Introduction to server-side JavaScript

Duration: 60 minutes

Overview	This unit introduces server-side JavaScript and Node.js. It describes how to create a Node.js server and implement Node.js modules.
Learning objectives	After completing this unit, you should be able to: <ul style="list-style-type: none">• Explain the origin and purpose of the Node.js JavaScript framework• Write a simple web server with Node.js• Import Node.js modules into your script

Exercise 1. Developing a Hello World Node.js app on IBM Cloud

Duration: 90 minutes

Overview	In this exercise, you create a Node.js Cloud Foundry application on IBM Cloud. You will develop a Node.js-based server application (by using the Eclipse Orion Web IDE) that responds to web browser requests.
Learning objectives	After completing this exercise, you should be able to: <ul style="list-style-type: none">• Create an IBM SDK for Node.js application.• Write your first Node.js application.• Deploy an IBM SDK for Node.js application on IBM Cloud.• Create a Node.js module and use it in your code.

Unit 2. Asynchronous I/O with callback programming

Duration: 90 minutes

Overview	The Node.js SDK relies on callback functions to handle network calls in an asynchronous manner. In this unit, you will learn how to write anonymous callback functions to act upon network events and listen, and intercept network traffic.
Learning objectives	After completing this unit, you should be able to: <ul style="list-style-type: none">• Explain synchronous and asynchronous calls.• Write asynchronous calls code in Node.js applications.• Explain request flows that are sent to Node.js applications that use the http module.

Exercise 2. Understanding asynchronous callback**Duration: 90 minutes**

Overview	This exercise shows how to use callback functions to call an external service. This exercise uses the IBM Watson Language Translator service in IBM Cloud. You create a Node.js module that contains the logic for these calls.
Learning objectives	After completing this exercise, you should be able to: <ul style="list-style-type: none">• Write asynchronous callbacks code in Node.js applications.

Unit 3. Express web application framework**Duration: 90 minutes**

Overview	This unit describes the Express web application framework, which provides a structured way to handle HTTP actions on server resources. You will learn how to write a REST service with Express and parse JSON data from an HTTP message.
Learning objectives	After completing this unit, you should be able to: <ul style="list-style-type: none">• Explain the difference between code that is written in “pure” JavaScript and code that is written with the Express framework.• Explain what Express is and its benefits.• Use Express as a third-party npm package.• Explain the use of middleware functions.• Handle routes and requests..

Exercise 3. Creating your first Express application**Duration: 90 minutes**

Overview	In this exercise, you create an application that uses the Express framework and the IBM Watson Natural Language Understanding service to extract the author name from articles that are published on the web. You provide the web address (URL) of the article to the application, and it outputs the name of the author (or multiple names if the article has multiple authors).
Learning objectives	After completing this exercise, you should be able to: <ul style="list-style-type: none">• Create a Hello World Express application.• Create a simple HTML view for your application.• Explain Express routing.• Use third-party modules in Node.js.• Use the Watson Natural Language Understanding service in your applications.• Use a Git repository in DevOps on IBM Cloud.

Unit 4. Async patterns with ECMAScript

Duration: 60 minutes

Overview	This unit describes async patterns in ECMAScript including callbacks, promises, and async/await.
Learning objectives	After completing this unit, you should be able to: <ul style="list-style-type: none">• Explain async patterns in ECMAScript such as callbacks, promises, and async/await.

Exercise 4. Building a rich UI application by using React and ES8

Duration: 90 minutes

Overview	This exercise guides you through building an interactive and rich client-side application by using React. You also explore the async/await feature of ECMAScript 2017, which is commonly known as ES8, and some features of ES8 through a server-side application by using Node.js.
Learning objectives	After completing this exercise, you should be able to: <ul style="list-style-type: none">• Deploy a React application on IBM Cloud.• Deploy a Node.js application on IBM cloud.• Explain the structure of a React application.• Use ES8 features in Node.js applications.

Unit 5. Building rich UI applications with React (optional)

Duration: 60 minutes

Overview	This unit describes async patterns in JavaScript. It introduces React and basic React concepts such as: <ul style="list-style-type: none">• Components• Props• State• Events• Component lifecycle• Nested components• Lists and keys.
Learning objectives	After completing this unit, you should be able to: <ul style="list-style-type: none">• Explain the React component lifecycle.• Explain React states, props, and events.• Create lists of nested components.• Explain React deployment options on IBM Cloud.