IBM Cloud Professional Certification Program

Study Guide Series

C1000-099: IBM Cloud Pak for Applications v4.1 Developer exam

Contents

Purpose of Exam Objectives

When an exam is developed, Subject Matter Experts work together to define the role the certified individual will fill. They define the tasks and knowledge that an individual would need to successfully perform this job role doe the product or solution. This creates the foundation for the objectives and measurement criteria, which form the basis of the certification exam. Question writers then use these objectives to develop exam questions.

It is recommended that you review these objectives and ask yourself the following questions:

- Do you know how to complete the task in the objective?
- Do you know why that task needs to be done?
- Do you know what will happen if you do it incorrectly?

If you are not familiar with a task, go through the objective, perform that task in your own environment and read more information on the task. If there is an objective on a task, there is a high likelihood that you WILL see a question about it on the actual exam. Review the recommended learning designed to prepare you to take the certification exam.

After reviewing the objectives in this guide and completing your own research, take the assessment exam. While the assessment exam does not indicate which specific questions were answered incorrectly, it does indicate overall performance by section. This is a good indicator of preparedness or if further preparation is warranted.

High-level Exam Objectives

Section 1 - Design and develop a Cloud Native solution	
1.1	Understand the main elements of a cloud native solution
1.2	Explain Containers and Container Orchestration
1.3	Explain Continuous Integration/Continuous Delivery and DevOps concepts
1.4	Design a Microservices application
Section 2 - OpenShift Container Platform Architecture	
2.1	Understand the OpenShift Container Platform Architecture
2.2	Develop and deploy applications on OpenShift 4.x
2.3	Utilize Developer Tools in OpenShift 4.x
Section 3 - Cloud Pak for Applications Overview	
3.1	Understand the Cloud Pak for Applications value proposition
3.2	Explain IBM Cloud Pak for Applications components and their value proposition

Section 4 - Develop and deploy new applications	
4.1	Develop and deploy serverless applications
4.2	Understand the capabilities of runtimes
4.3	Deploy a new application on OpenShift
4.4	Manage applications with Application Navigator
Section 5 -Develop and deploy new applications with Accelerators for Teams	
5.1	Utilize the available developer tools
5.2	Utilize application stacks
5.3	Set up a local development environment
5.4	Develop on CodeReady
5.5	Develop using Codewind as an Integrated Development Environment
Section 6 - Modernize Applications	
6.1	Describe Application Modernization concepts
6.2	Understand the available Application Modernization tools
6.3	Evaluate a traditional application using Transformation Advisor
6.4	Replatform: Migrate a traditional application to OpenShift
6.5	Repackage: Containerize an application to run on OpenShift
Section 7 - Continuous Integration/Continuous Deployment (CI/CD) and the DevOps lifecycle	
7.1	Understand, utilize and customize OpenShift Pipelines tooling and evolution
7.2	Develop pipelines using OpenShift Pipelines
7.3	Develop pipelines using Jenkins
7.4	Understand and use container registries in OpenShift
7.5	Describe at a high level the content and capabilities provided in Cloud DevOps for IBM Cloud Pak for Applications

Detailed Exam Objectives

Section 1 - Design and develop a Cloud Native solution

1.1. Understand the main elements of a cloud native solution.
SUBTASKS:
1.1.1. Define Cloud Native
1.1.2. Identify the advantages and disadvantages of Cloud Native
1.1.3. Define Cloud Native Architectures
1.1.4. Differentiate between cloud native and traditional applications

Study Material:
https://www.ibm.com/cloud/learn/cloud-native
https://www.ibm.com/cloud/blog/ibm-cloud-podcast-everything-cloud-native

1.1.5. Understanding Cloud Storage
Study Material:
https://www.ibm.com/cloud/learn/cloud-storage

1.2. Explain Containers and Container Orchestration.
SUBTASKS:
1.2.1. Define containers
- 1.2.1.1. Containers vs. VMs
- 1.2.1.2. Benefits of containers
- 1.2.1.3. Use cases

Study Material:
https://www.ibm.com/cloud/learn/containers
https://www.ibm.com/cloud/learn/containers

1.2.2. Define containerization
- 1.2.2.1. Types of containerization
- 1.2.2.2. Security in containerization

Study Material:
https://www.ibm.com/cloud/learn/containerization

1.2.3. Define container orchestration
- 1.2.3.1. Understand platforms (Kubernetes)
- 1.2.3.2. Deploying
- 1.2.3.3. Scaling
- 1.2.3.4. Networking

Study Material:
https://www.ibm.com/cloud/blog/container-orchestration-explained
https://www.ibm.com/cloud/learn/kubernetes
https://www.ibm.com/cloud/learn/kubernetes

1.3. Explain Continuous Integration/Continuous Delivery and DevOps concepts.
SUBTASKS:

1.3.1. Define continuous integration
- 1.3.1.1. Understand benefits
- 1.3.1.2. Explain CI, Agile, and DevOps
- 1.3.1.3. Define available tools
- 1.3.1.4. Best practices

Study Material:
https://www.ibm.com/cloud/learn/continuous-integration

1.3.2. Define continuous delivery
    1.3.2.1. Best practices
    1.3.2.2. Understand benefits
    1.3.2.3. Understand continuous pipeline/framework
    1.3.2.4. Understand continuous deployment vs continuous delivery
    1.3.2.5. Define available tools
    1.3.2.6. Understand Agile and DevOps
Study Material:
https://www.ibm.com/cloud/learn/continuous-delivery

1.4. Design a Microservices application.
SUBTASKS:
1.4.1. Define Microservices Architecture
    1.4.1.1. Identify drawbacks
    1.4.1.2. Supporting tools

Study Materials:
https://developer.ibm.com/articles/why-should-we-use-microservices-and-containers/
https://www.ibm.com/cloud/architecture/architectures/microservices

1.4.2. Design a Microservices applications
Study Materials:
https://www.ibm.com/cloud/blog/know-developing-applications-microservices
Architecting an Application with Microservices
Implementing Microservices
Using Microservice Application Patterns
Architecting for Microservice Resiliency
Exposing and Versioning Microservice APIs

Section 2 –OpenShift Container Platform Architecture

2.1. Understand the OpenShift Container Platform Architecture.
SUBTASKS:
2.1.1. Understand the high-level components of the OpenShift Container Platform 4.x Architecture
    2.1.1.1. Control Plane
        2.1.1.1.1. RHCOS
    2.1.1.2. Compute Nodes (worker nodes)
        2.1.1.2.1. RHCOS/RHEL

2.1.1.3. Load Balancer nodes

REFERENCES:
https://docs.openshift.com/container-platform/4.3/architecture/architecture.html

2.2. Develop and deploy applications on OpenShift 4.x.
SUBTASKS:
2.2.1. Describe the key components of an OpenShift application
- 2.2.1.1. deploymentconfig resource
- 2.2.1.2. replicaset resource
- 2.2.1.3. configmap resource
- 2.2.1.4. pod/container resources
- 2.2.1.5. statefulset resource
- 2.2.1.6. daemonset resource
- 2.2.1.7. operator resource

2.2.2. Develop and deploy applications on OpenShift
- 2.2.2.1. Use of Operators
- 2.2.2.2. Use of Git
- 2.2.2.3. Use of the Catalog to deploy applications
- 2.2.2.4. Use of the CLI to deploy applications

REFERENCES:
https://docs.openshift.com/container-platform/4.3/applications/application_life_cycle_management/creating-applications-using-cli.html
https://developers.redhat.com/blog/2020/01/17/deploying-applications-in-the-openshift-4-3-developer-perspective
https://docs.openshift.com/container-platform/4.3/operators/olm-what-operators-are.html

2.3. Utilize Developer Tools in OpenShift 4.x.

SUBTASKS:
2.3.1. Develop applications on OpenShift utilizing the different developer tools
- 2.3.1.1. Source to Image tools (S2I)
- 2.3.1.2. CodeReady
- 2.3.1.3. Odo
- 2.3.1.4. Buildah

REFERENCES:
https://docs.openshift.com/container-platform/4.3/builds/understanding-image-builds.html#build-strategy-s2i_understanding-image-builds
https://developers.redhat.com/products/codeready-workspaces/overview
https://docs.openshift.com/dedicated/welcome/index.html
https://docs.openshift.com/container-platform/4.3/builds/custom-builds-buildah.html

Section 3 –Cloud Pak for Applications Overview

3.1. Understand the Cloud Pak for Applications value proposition. SUBTASK(S):

3.1.1. Understand the Cloud Native value propositions for Cloud Pak for Applications.
- 3.1.1.1. Develop cloud native apps – open source, common services, dev tools, DevOps
- 3.1.1.2. Full-stack certification and support
- 3.1.1.3. Supported solution during the transition and re-factoring of existing applications

3.1.2. Understand the Application Modernization value propositions for Cloud Pak for Applications.
- 3.1.2.1. Support and tools – TA, liberty, IBM Support for existing apps, etc.
- 3.1.2.2. License Flexibility – right-size between WAS editions – future proof
- 3.1.2.3. Consistent Platform – RHOCP – run apps anywhere (private, public, hybrid)

3.1.3. Run Existing Applications – WebSphere, JBoss, Node, etc.
3.1.4. Delivery of apps on multiple clouds
3.1.5. Accelerate development with governance and security

REFERENCES:
https://www.ibm.com/docs/en/cloud-paks/cp-applications/4.3?topic=overview
https://www.ibm.com/cloud/cloud-pak-for-applications
https://www.ibm.com/cloud/cloud-pak-for-applications
https://www.ibm.com/cloud/cloud-pak-for-applications
https://www.ibm.com/cloud/cloud-pak-for-applications
https://www.ibm.com/cloud/architecture/files/app-modernization-field-guide.pdf

3.2. Explain IBM Cloud Pak for Applications components and their value proposition.
SUBTASK(S):

3.2.1. Run existing apps – WebSphere Application Server Network Deployment, WebSphere Application Server Base, Liberty, Mobile Foundation, Red Hat Runtimes
3.2.2. Modernize existing apps – Transformation Advisor, App Navigator, Migration Tookit
3.2.3. Build new Cloud Native apps – Accelerators for Teams
3.2.4. Build new Cloud Native apps – Red Hat Runtimes
3.2.5. Platform – Red Hat OpenShift Container Platform
3.2.6. IBM Cloud Pak for Applications DevOps Add-On – Cloud Native and Modernization

REFERENCES:
https://www.ibm.com/cloud/architecture/architectures/application-modernization/overview
https://www.redhat.com/en/products/runtimes
https://access.redhat.com/articles/4394291
https://www.youtube.com/watch?v=6Tdmh1JsoVI

Section 4 –Develop and deploy new applications

4.1. Develop and deploy serverless applications.

SUBTASK(S):
4.1.1. Understand information available from the serverless Custom Resource
REFERENCE:
https://www.ibm.com/docs/en/cloud-paks/cp-applications/4.1?topic=definitions-kabanero-custom-resource
https://www.ibm.com/docs/en/cloud-paks/cp-applications/4.1?topic=definitions-kabanero-custom-resource

4.1.2. Understand the concept of Serverless Computing
REFERENCES:
https://www.ibm.com/cloud/learn/serverless
https://cloud.redhat.com/learn/topics/serverless
https://developer.ibm.com/articles/knative-what-is-it-why-you-should-care/

4.1.3. Create a Cloud-Native App and make it serverless
REFERENCES:
https://github.com/IBM/think2020-cp4a-master-class
https://developer.ibm.com/tutorials/knative-101-labs/

4.2. Understand the capabilities of runtimes.

SUBTASK(S):
4.2.1. Understand the stacks that are available through the Cloud Pak for Applications
REFERENCE:
https://www.ibm.com/docs/en/cloud-paks/cp-applications/4.1?topic=applications-developing-stacks

4.2.2. Develop with the OpenLiberty stack
REFERENCES:
https://www.ibm.com/docs/en/cloud-paks/cp-applications/4.1?topic=stacks-developing-open-liberty-stack

https://cp4apps.cloudnative101.dev/web/1.0.0/e2e-java-openliberty.html

4.2.3. Develop with the Eclipse MicroProfile stack

REFERENCE:
https://www.ibm.com/docs/en/cloud-paks/cp-applications/4.1?topic=stacks-developing-eclipse-microprofile-stack

4.2.4. Develop with the Node.js stack

REFERENCE:
https://www.ibm.com/docs/en/cloud-paks/cp-applications/4.1?topic=stacks-developing-nodejs-stack

4.2.5. Develop with the Node.js Express stack

REFERENCES:
https://www.ibm.com/docs/en/cloud-paks/cp-applications/4.1?topic=stacks-developing-nodejs-express-stack
https://cp4apps.cloudnative101.dev/web/1.0.0/e2e-nodejs-express.html

4.2.6. Develop with the Spring Boot stack

REFERENCES:
https://www.ibm.com/docs/en/cloud-paks/cp-applications/4.1?topic=stacks-developing-springboot-stack
https://cp4apps.cloudnative101.dev/web/1.0.0/e2e-java-spring-boot2.html

4.3. Deploy a new application on OpenShift.
SUBTASK(S):
4.3.1. Know the tasks and pipelines available by default
REFERENCE:
https://www.ibm.com/docs/en/cloud-paks/cp-applications/4.1?topic=applications-building-deploying-pipelines#default-tasks-and-pipelines

4.3.2. Know how to use stacks that are published to internal and private registries in pipelines
REFERENCE:
https://www.ibm.com/docs/en/cloud-paks/cp-applications/4.1?topic=applications-building-deploying-pipelines#using-stacks-that-are-published-to-internal-and-private-registries-in-pipelines

4.3.3. Use pipelines to define and deploy an application stack
4.3.4. Check the status of a pipeline run

REFERENCE:
https://www.ibm.com/docs/en/cloud-paks/cp-applications/4.1?topic=applications-building-deploying-pipelines#running-pipelines

4.3.5. View an application's composition using the Topology View

REFERENCE:
https://docs.openshift.com/container-platform/4.3/applications/application_life_cycle_management/odc-viewing-application-composition-using-topology-view.html

4.3.6. Create Deployments and DeploymentConfigs for an application

REFERENCE:
https://docs.openshift.com/container-platform/4.3/applications/deployments/what-deployments-are.html

4.4. Manage applications with Application Navigator.
4.4.1. Navigate cloud native applications using Application Navigator

REFERENCES:
https://www.ibm.com/docs/en/cloud-paks/cp-applications/4.1?topic=navigator-application-overview
https://www.ibm.com/docs/en/cloud-paks/cp-applications/4.1?topic=navigator-cloud-native-applications

4.4.2. Create Applications in Application Navigator
https://github.com/kappnav/README/blob/master/how-to-create-applications.md

4.4.3. Differentiate between actions, resources and applications

REFERENCES:
https://domain.name.parking.diandongzhi.com/?acct=144&site=kappnav.io
https://github.com/kappnav/README

Section 5 –Develop and deploy new applications with Accelerators for Teams

5.1. Utilize the available developer tools.
SUBTASK(S):

5.1.1. Understand Developer Worker Flow and Tools in Accelerators for Teams
https://developer.ibm.com/blogs/cloud-native-development-grows-up/
[Video] Kabanero Developer Workflow
[Video] Kabanero Guide Tours

5.1.2. Build Cloud Native applications with Accelerators for Teams
https://developer.ibm.com/blogs/cloud-native-apps-kubernetes-kabanero/

5.1.3. Develop containerized applications on CLI using Appsody
https://developer.ibm.com/blogs/introduction-to-appsody/
https://appsody.dev/docs/getting-started/
https://www.ibm.com/docs/en/cloud-paks/cp-applications/4.1?topic=started-developing-command-line-interface-cli
https://www.ibm.com/docs/en/cloud-paks/cp-applications/4.1?topic=started-developing-command-line-interface-cli
Accelerators for Teams Developers Experience

5.1.4. Deploy an application to OpenShift cluster using Appsody
https://cp4apps.cloudnative101.dev/web/1.0.0/appsody-deploy.html
https://developer.ibm.com/tutorials/deploy-appsody-applications-to-openshift/
https://developer.ibm.com/tutorials/deploy-appsody-applications-to-openshift/
https://developer.ibm.com/patterns/create-insurance-quote-application-appsody/
https://developer.ibm.com/patterns/create-insurance-quote-application-appsody/
https://developer.ibm.com/tutorials/configure-an-observable-microservice-with-appsody-openshift-open-liberty/
https://developer.ibm.com/tutorials/use-appsody-develop-application-containerize-migrate-docker-hub/

5.1.5. Build a cloud native application with Codewind and your IDE
https://www.ibm.com/docs/en/cloud-paks/cp-applications/4.1?topic=started-developing-codewind-vs-code
https://developer.ibm.com/tutorials/develop-a-cloud-native-java-application-using-codewind/

5.1.6. Use Appsody and Codewind to build an application on Kubernetes
https://developer.ibm.com/tutorials/kabanero-introduction-to-modern-microservices-development-for-kubernetes/
https://developer.ibm.com/tutorials/kabanero-introduction-to-modern-microservices-development-for-kubernetes/

5.2. Utilize application stacks.
SUBTASKS:
5.2.1. Differentiate between repositories, stacks, and templates

5.2.2. Develop a custom Application Stack Template
5.2.3. Setup application stacks
5.2.4. Customize an existing application stack
5.2.5. Setup the Appsody CLI and Codewind to work with the same set of Appsody stacks

REFERENCES:
https://developer.ibm.com/tutorials/create-appsody-stack/
https://www.ibm.com/docs/en/cloud-paks/cp-applications/4.1?topic=teams-setting-up-application-stacks
https://www.ibm.com/docs/en/cloud-paks/cp-applications/4.1?topic=teams-setting-up-application-stacks
https://developer.ibm.com/components/websphere-hybrid-edition/
https://developer.ibm.com/tutorials/modify-appsody-stack/
https://cp4apps.cloudnative101.dev/web/1.0.0/codewind-setup-appsody.html

5.3. Set up a local development environment. SUBTASKS:
5.3.1. Install and configure App Modernization tools
    5.3.1.1. Install and configure IBM Transformation Advisor Local
    5.3.1.2. Install and configure WebSphere migration toolkit
5.3.2. Setup Development tools
    5.3.2.1. Install and configure Codewind
    5.3.2.2. Install and configure Appsody
    5.3.2.3. Install and configure RedHat CodeReady Containers
5.3.3. Setup command line tools
    5.3.3.1. Install and configure Git
    5.3.3.2. Setup and configure OpenShift command line interface (oc)
    5.3.3.3. Install and configure a local container environment

REFERENCES:
https://www.ibm.com/support/pages/node/6250961
https://appsody.dev/docs/installing/installing-appsody/
https://crc.dev/crc/
https://docs.docker.com/get-docker/

5.4. Develop on CodeReady.
SUBTASKS:
5.4.1. Describe the difference between OpenShift production and CodeReady
5.4.2. Download and install CodeReady on a local PC
5.4.3. Describe the capabilities of Red Hat CodeReady Workspaces
5.4.4. Import OpenShift application into a workspace

REFERENCES:
https://developers.redhat.com/blog/2019/09/05/red-hat-openshift-4-on-your-laptop-introducing-red-hat-codeready-containers
https://access.redhat.com/documentation/en-us/red_hat_codeready_containers/1.10/html/getting_started_guide/installation_gsg
https://developers.redhat.com/products/codeready-workspaces/overview

5.5. Develop using Codewind as an Integrated Development Environment
SUBTASKS:
5.5.1. Describe Codewind
5.5.2. Understand how Codewind can be used to create cloud-native applications
    5.5.2.1. How to use Codewind locally
    5.5.2.2. How to use Codewind remotely
    5.5.2.3. How to use Codewind as a hosted application on cloud
5.5.3. Understand the requirements for set up of Codewind
5.5.4. Describe the use of templates in Codewind
    5.5.4.1. How to manage templates
    5.5.4.2. How to create templates

REFERENCES:
https://www.eclipse.org/codewind/overview.html
https://www.eclipse.org/codewind/workingwithtemplates.html
https://www.eclipse.org/codewind/learn.html
https://developer.ibm.com/blogs/codewind-introduction/

Section 6 –Modernize Applications

6.1. Describe Application Modernization concepts. SUBTASKS:
6.1.1.Identify the IBM approach to Application Modernization
    6.1.1.1. List Modernization goals
    6.1.1.2. Distinguish Modernization patterns
    6.1.1.3. Describe the 3 R's: Refactor, Replatform, Repackage

REFERENCE:
https://www.ibm.com/cloud/architecture/architectures/application-modernization/

SUBTASKS:
6.1.2.Define methods for Application Modernization
    6.1.2.1. Assess Application portfolio
    6.1.2.2. Align priorities with options
    6.1.2.3. Containerize applications

REFERENCE:

https://www.ibm.com/cloud/architecture/files/app-modernization-field-guide.pdf

SUBTASKS:

6.1.3.Compare Cloud Adoption with Application Modernization

- 6.1.3.1. Determine cultural and organizational change: people
- 6.1.3.2. Define new ways of working: process
- 6.1.3.3. Adopt cloud-native: technology

REFERENCES:

https://www.ibm.com/cloud/architecture/files/ibm-cloud-adoption-transformation-field-guide.pdf
https://www.ibm.com/garage/method/practices/code/chunking-strategy-strangler-pattern

6.2. Understand the available Application Modernization tools SUBTASKS:

6.2.1.Analyze the Transformation Advisor architecture

- 6.2.1.1. List the middleware that Transformation Advisor can analyze
- 6.2.1.2. List supported Java versions for the application assessment
- 6.2.1.3. Identify the architecture components a developer will interact with

REFERENCES:

https://www.ibm.com/docs/en/cta?topic=about-transformation-advisor
https://www.ibm.com/docs/en/cta?topic=about-transformation-advisor

SUBTASKS:

6.2.2.Utilize Transformation Advisor artifacts

- 6.2.2.1. Distinguish the migration artifacts produced by Transformation Advisor and their functionality
- 6.2.2.2. Define the process to use Transformation Advisor artifacts

REFERENCES: 做到这个

https://www.ibm.com/docs/en/cta?topic=migration-artifacts
https://www.ibm.com/docs/en/cta?topic=ma-how-deploy-your-app-red-hat-openshift-container-platform

SUBTASKS:

6.2.3. Articulate use of the WebSphere Application Server Migration Toolkit

- 6.2.3.1. Basic Understanding of WebSphere Application Server Migration Toolkit
- 6.2.3.2. How does WS Migration Toolkit fit into the App Modernization Journey?

REFERENCES:

https://www.ibm.com/support/pages/node/6250961#asset/tools-WebSphere_Application_Server_Migration_Toolkit

https://www.ibm.com/cloud/architecture/content/course/explore-application-modernization/appmod-transformation-advisor

6.3. Evaluate a traditional application using Transformation Advisor
SUBTASKS:
6.3.1.Collect Data using Transformation Advisor
- 6.3.1.1. Download the data collector
- 6.3.1.2. Run the data collector
- 6.3.1.3. Understand command line options
- 6.3.1.4. Troubleshoot data collection
- 6.3.1.5. Import data collections
- 6.3.1.6. Use IIB data collector

REFERENCES:
https://developer.ibm.com/recipes/tutorials/using-the-transformation-advisor-on-ibm-cloud-private/
https://www.ibm.com/docs/en/cta?topic=started-using-data-collector
https://www.ibm.com/docs/en/app-connect/11.0.0?topic=tasks-running-transformation-advisor-tool
https://www.ibm.com/docs/en/cta?topic=day-2-operations-troubleshooting

6.3.2.Use the Transformation Advisor reports
- 6.3.2.1. View the analysis results
- 6.3.2.2. Describe the Technology Evaluation Report
- 6.3.2.3. Use the Application Inventory Report
- 6.3.2.4. Explain the importance of Enterprise JavaBeans
- 6.3.2.5. Illustrate the Detailed Migration Analysis Report
- 6.3.2.6. Import data collections
- 6.3.2.7. Use IIB data collector

REFERENCES:
https://www.ibm.com/docs/en/cta?topic=started-finding-your-way-around-ui
https://developer.ibm.com/recipes/tutorials/using-the-transformation-advisor-on-ibm-cloud-private/
https://developer.ibm.com/recipes/tutorials/using-the-transformation-advisor-on-ibm-cloud-private/
https://www.ibm.com/demos/

6.4. Replatform: Migrate a traditional application to OpenShift **SUBTASKS:**
6.4.1.Choose WAS base in containers vs. WAS liberty
- 6.4.1.1. Define Operational modernization
- 6.4.1.2. Describe use-cases for which WAS base is preferable
- 6.4.1.3. List advantages of migrating to WAS base in containers

6.4.2.Build a traditional WebSphere container
6.4.3.Push the container to the OpenShift registry

6.4.4.Deploy the application WAS base container

REFERENCES:
https://www.ibm.com/demos/
https://www.ibm.com/cloud/architecture/architectures/op-modernization-solution

6.5. Repackage: Containerize an application to run on OpenShift

SUBTASKS:
6.5.1.Create a Business App
6.5.2.Create Migration bundle
6.5.3.Update Migration bundle
6.5.4.Test Migration bundle locally 6.5.5.Containerize Liberty Application
REFERENCES:
https://www.ibm.com/cloud/architecture/architectures/runtime-modernization-solution
https://www.ibm.com/demos/

SUBTASKS:
6.5.6.Deploy Containerized Application to OpenShift

REFERENCE:
https://www.ibm.com/docs/en/cta?topic=ma-how-deploy-your-app-red-hat-openshift-container-platform

SUBTASKS:
6.5.7.Describe modernization of Spring applications

Section 7 –Continuous Integration/Continuous Deployment (CI/CD) and the DevOps lifecycle

7.1. Understand, utilize and customize OpenShift Pipelines tooling and evolution (Jenkins and Tekton)
SUBTASKS:
7.1.1.Describe OpenShift CI/CD concepts
REFERENCE:
https://cloud.redhat.com/blog/cloud-native-ci-cd-with-openshift-pipelines

SUBTASKS:
7.1.2.Understand support for Jenkins in OpenShift 4
REFERENCE:
https://docs.openshift.com/container-platform/4.3/builds/build-strategies.html
(Pipeline build only)

SUBTASKS:
7.1.3.Explain the benefit of custom operators in OpenShift 4
REFERENCE:
https://docs.openshift.com/container-platform/4.3/operators/operator_sdk/osdk-getting-started.html

7.2. Develop pipelines using OpenShift Pipelines (Tekton)

SUBTASKS:
7.2.1.Explain Tekton pipeline concepts
REFERENCES:
https://openshift.github.io/pipelines-docs/docs/0.10.5/con_pipelines-concepts.html
https://openshift.github.io/pipelines-docs/docs/0.10.5/con_pipeline-task.html
https://openshift.github.io/pipelines-docs/docs/0.10.5/con-pipeline-resource-types.html

SUBTASKS:
7.2.2.Creating Tekton CI/CD pipelines using the CLI
REFERENCE:
https://openshift.github.io/pipelines-docs/docs/0.10.5/assembly_creating-applications-with-cicd-pipelines.html

SUBTASKS:
7.2.3.Creating Tekton CI/CD pipelines using the Developer perspective
REFERENCE:
https://openshift.github.io/pipelines-docs/docs/0.10.5/assembly_working-with-pipelines-using-the-developer-perspective.html

SUBTASKS:
7.3. Develop pipelines using Jenkins
7.3.1.Explain how to use Jenkins to manage the application lifecycle on OpenShift
REFERENCE:
https://docs.openshift.com/container-platform/3.11/dev_guide/openshift_pipeline.html

SUBTASKS:
7.3.2.Configure Jenkins images on OpenShift 4.3
REFERENCES:

https://docs.openshift.com/container-platform/4.3/openshift_images/using_images/images-other-jenkins.html (intro only)
https://docs.openshift.com/container-platform/4.3/openshift_images/using_images/images-other-jenkins.html#images-other-jenkins-customize-s2i_images-other-jenkins
https://docs.openshift.com/container-platform/4.3/openshift_images/using_images/images-other-jenkins.html#images-other-jenkins-create-service_images-other-jenkins

7.4. Understand and use container registries in OpenShift (local, private, semi-private)
SUBTASKS:
7.4.1.Explain the function of a container registry in CI/CD processes
REFERENCE:
https://docs.openshift.com/container-platform/4.3/registry/architecture-component-imageregistry.html

SUBTASKS:
7.4.2.Describe the different container registry options available in OpenShift 4.3
REFERENCE:
https://docs.openshift.com/container-platform/4.3/registry/registry-options.html

7.5. Describe at a high level the content and capabilities provided in Cloud DevOps for IBM Cloud Pak for Applications

SUBTASKS:
7.5.1.Describe the contents of the Cloud DevOps for IBM Cloud Pak for Applications

REFERENCE:
https://www.ibm.com/common/ssi/cgi-bin/ssialias?infotype=an&subtype=ca&appname=gpateam&supplier=897&letternum=ENUS220-051

SUBTASKS:
7.5.2.Discuss the capabilities provided by UCV/UCD
REFERENCES:
https://www.ibm.com/docs/en/urbancode-velocity/1.4.x
https://www.ibm.com/docs/en/urbancode-deploy/7.0.4
https://www.ibm.com/docs/en/urbancode-deploy/7.0.4

Next Steps

1. Take IBM Cloud Pak for Applications v4.1 Developer exam assessment test for this exam.

2. If you pass the assessment exam, visit pearsonvue.com/ibm to schedule your testing sessions.
3. If you failed the assessment exam, review how you did by section. Focus attention on the sections where you need improvement. Keep in mind that you can take the assessment exam as many times as you would like ($30 per exam); however, you will still receive the same questions only in a different order.