Section 1 Module: 1.4 - HLDD

Section 1 – IBM Cloud Options

Course 1.4 Containers and Kubernetes

Topics

Section 1: Understanding the Benefits of Containers

- Introduction to Containers
- Docker
- Building Container Images
- Container Registries
- Lab: Running Containers

Section 2: Understanding Kubernetes Architecture

- Container Orchestration
- Kubernetes Architecture
- Intro to Kubernetes Objects
- Basic Kubernetes Objects
- Kubectl CLI
- Using Kubernetes

Section 3: Managing Applications with Kubernetes

- ReplicaSets
- Autoscaling
- Basic Policies Supported by Kubernetes
- Rolling Updates
- ConfigMaps and Secrets
- Persistent Volumes and Persistent Volume Claims
- Service Binding
- Kubernetes Probes

Section 1 Module: 1.4 - HLDD

Course Description

This course provides learners with an introduction to containers, emphasizing their benefits in enterprise devops environments. Docker, a popular container implementation, is introduced, and the high-level Docker architecture is presented. Docker concepts and terminology are explained. Containers are efficient and portable, but they have to be managed and monitored in a business environment. The concept of container orchestration, which addresses the management and monitoring issue, is explained. Finally, Kubernetes, a popular container orchestration solution is introduced.

Section Duration

210 minutes (3 hours, 30 mins)

Learning Objectives (Terminal)

Upon completing this course, the learners will be able to do the following:

- Describe what a container is and how it differs from a virtual machine
- Describe the benefits of using containers
- Describe the key lifecycle states of a container
- Explain the difference between a container and a pod
- Describe the key lifecycle stages of a pod
- Describe the key elements of a container specification
- Describe the key elements of container orchestration with Kubernetes
- Describe the key components of Kubernetes
- Explain the basic policies supported by Kubernetes
- Identify and Describe the different probes available with Kubernetes

Section 1 Module: 1.4 - HLDD

Course Outline

Containers and Kubernetes

Section 1 Development Resource(s)

- Containers and Kubernetes course (new) from IBM
- Containerization video from IBM
- Docker Overview video from IBM
- Blog article on lifecycle / states of a container: https://www.ibm.com/blogs/cloud-computing/2016/02/08/the-6-steps-of-the-container-lifecycle/

Section 1: Understanding the Benefits of Containers

Lesson # and Name	Learning Objective	Time Estimate	Tool
Lesson 1 – Introduction to Containers *Design Strategy:* • Topic 1 – What is a container? *Image/text block;* - A container is an executable unit of software in which application code is packaged, along with its libraries and dependencies, in common ways so that it can be run anywhere, whether on a desktop, on-premises, or in the cloud. • Topic 2 – Shipping Containers *Text on image of cargo ship* - Shipping containers were introduced to standardize the way things are packed and shipped. Now the only	-Explain the definition of a container 1.3.1 Describe how containers differ from virtual machines 1.3.2 Describe the benefits of using containers in cloud computing -Identify the lifecycle of a container 1.3.3. Describe the key lifecycle states of a container	9 minutes	Articulate Rise, Articulate Storyline and IBM video

thing that can go on a cargo ship is a container. Anything that fits in a shipping container can be shipped. By standardizing, shipping became much more predictable and optimized. • Topic 3 – Software Containers *Image & text block* - The containers we're talking about in this course are much the same. They are a standardization for how we package and ship software. • Topic 4 – Container Benefits *Click to Reveal interaction* - Lightweight - Portable and platform independent - Supports modern development and architecture - Improves utilization • Topic 5 – Containers vs. Virtual Machines *Storyline interaction – graphic with hotspots; learner clicks on diagrams to learn about the different layers in each diagram (VO narration required).* - Emphasize the difference between the layers in a container vs. a virtual machine • Topic 6 - Lifecycle/states of container			

- Lifecycle *(process block)*
  ○ Acquire
  ○ Deliver
  ○ Build
  ○ Deploy
  ○ Run
  ○ Maintain
- States *(flipcards)*
  ○ Running
  ○ Pending
  ○ Terminated
- *Sorting activity: Drag and drop each card into the correct category (Lifecycle or States)*
• Topic 7 -Use Cases for Containers
Click and reveal interaction –
- *Microservices*
- *DevOps*
- *Hybrid, multi-cloud*
- *Application modernizing and migration*
• Topic 8 - Containerization explained
Text block introduction followed by video block: may need a note as to what Docker is before the video block) (video - IBM Developer Advocate Sai Vennam explains containerization) (8 mins?)
• Knowledge Check Containerization video – *Storyline series of M/C, T/F, drag/drop questions to review material in this video.*
		4 minutes	

Lesson 2 – Docker *Design Strategy:* • Topic 1 – What is Docker? - Introduction – *text block*	-Define Docker and what it is -Identify the Docker CLI common commands	7 minutes	Articulate Rise, Articulate Storyline and IBM video

- What is Docker? – *click and reveal interaction*
  - Software platform for building and running containers
  - Open-source community
  - Company that provides enterprise software
• Topic 2 - Docker CLI
Image and text block
• Topic 3 - Common Docker Commands
Accordion interaction
  - build
  - tag
  - images
  - run
  - push
  - pull
  - *Matching activity: match the Docker command to the definition.*
• Topic 4 - Container Runtime
Image and text block
  - Docker is also a container runtime. A container runtime is software that executes containers, much like the docker run command that we just referenced.
  - Let's see how the container runtime fits into a typical stack.
  - *Hotspot interaction: Click on the area on the stack where the container runtime would reside.*
• Topic 5 – Docker Overview – *video block (Note: This is an IBM video where Sai Vennam, a developer advocate at IBM, as he introduces*
	1.3.7: Describe container runtimes		

Docker and the benefits of running an application with containers). • Knowledge Check on Docker Overview video *Storyline series of M/C, T/F, drag/drop questions to review material in Topic 5 Docker Overview video.*		4 minutes	
Lesson 3 – Building Container Images *Design Strategy:* • Topic 1 - Development process of running a container *VYOND video* - Dockerfile - Image - Container. • Topic 2 - Image Layers *Storyline graphic animation with narration* - Docker build container images by using instructions from Dockerfile. - What is Dockerfile - As Docker instructions run, layers are built on image - Once all the Docker instructions from a Dockerfile have been executed, the image is built and complete. Again, this is read-only; making a change would result in a new image. - As we just shared, running an image results in a container. So, when we instantiate this image, we get a running container. • Topic 3 - Dockerfile instructions	- Identify the process to build container images - Define a Dockerfile and how instructions build Docker containers - Describe the differences between containers and images	8 minutes	Articulate Rise, Articulate Storyline and VYOND

Text block for intro text followed by accordion interaction - FROM: Define base image - RUN: Execute arbitrary commands - ENV: Set environment variables - ADD and COPY: Copy files and directories - CMD: Define default command for container execution • Topic 3 - Dockerfile Sample Code – *VYOND video* • End of Lesson Knowledge Check: *Storyline series of M/C, T/F, drag/drop questions to review material in Lesson 3.*		4 minutes	
Lesson 4 – Container Registries *Design Strategy:* • Topic 1 – What is a container registry? *Storyline text/graphics followed by two click/reveal interactions – see below)* - Storage and distribution of named container images - Public or private *(Storyline click / reveal interaction)* - Hosted or self-hosted *(Storyline click / reveal interaction)* • Topic 2 – Storing and Retrieving Images from Registries *Storyline animated graphics with narration* - *Push = storing image* - *Pull = retrieving image* - *Knowledge check – two questions on topic 1 and 1 or 2 questions on topic 2.*	-Identify the purpose of a container registry - Identify the ways in which users interact with registries - Identify how to name an image using standard conventions -Identify additional functionality of the IBM Cloud registry	7 minutes	Articulate Rise and Articulate Storyline

• Topic 3 – Image Naming *Storyline animated graphics with narration* - Hostname - Repository - Tag • Topic 4 – IBM Cloud Container Registry *Storyline animated graphics with narration* - IBM Cloud Container Registry - Kubernetes Service - Docker Hub / Other Registries - How users can interact with registries – GUI, CLI, Local image store			
Lesson 5: Lab Running Containers *Design Strategy:* • Topic: Lab *Steps, Code Snippet, and Image blocks* • The lab will include the following sections: - Use Docker file to create an image - Use repository name "my-app" and tag "v1" - Specify build context (directory where files are used to create your image) - Examine output of the image - Use Docker tag option to create a new tag for this image "second-app1" - Examine output – see two apps - Use Docker run command along with image tag to run it as container		15 minutes	Articulate Rise *Design note: Labs will be documented step-by-step and integrated into the course through the LMS.*

- Examine output: There is a “hello world” application that prints a simple message before exiting - Perform Docker push command to push image to registry for distribution and storage - Examine output – each layer of image was pushed successfully!			
	TOTAL TIME	**58 minutes**	

Section 1 Module: 1.4 - HLDD

Section 2 Development Resource(s)

- Containers and Kubernetes course (new) from IBM
- Describe the key lifecycle stages of a pod https://www.openshift.com/blog/kubernetes-pods-life

Section 2: Understanding Kubernetes Architecture

Lesson	Learning Objective	Time Estimate	Tool
Lesson 1 – Container Orchestration *Design Strategy:* • Topic 1 – The need for containerization *Storyline animation to show number of containers growing (use slides from Module 2.1 Container Orchestration - slides 2-4***)** • Topic 2 – Container Orchestration & Lifecycle of Containers *List block (use slide 5 from Module 2.1 Container Orchestration)* • Topic 3 – What is Kubernetes? *Interactive graphic with markers (Use slides 6-7 from Module 2.1 Container Orchestration)* • Topic 4 – What Kubernetes is not. *Accordion interaction (Use slide 8 from Module 2.1 Container Orchestration)* 	1.4.1 Describe the key elements of container orchestration with Kubernetes Explain container orchestration so that we can better understand how to create complex environments of many running containers. Explain a container orchestration tool, Kubernetes	5 minutes	Articulate Rise and Articulate Storyline
Lesson 2 – Kubernetes Architecture *Design Strategy:* • Topic 1 – Kubernetes Components *control plane shown as image followed by tabbed sequence:* - Kubernetes API server	Explain the architecture of the Kubernetes system	6 minutes	Articulate Rise

- Etcd - Kubernetes scheduler - Kubernetes controller manager - Cloud controller manager • Topic 2 – Kubernetes Components *Interactive graphic with markers* - Nodes (kubelet, kube-proxy) - Controllers • Topic 3 – What is a control loop? *text and image side-by-side* • Topic 4 – Controllers *text block*			
Lesson 3 – Intro to Kubernetes Objects *Design Strategy:* • Topic 1 – What are Kubernetes Objects? *Create VYOND video* - Namespaces and names - Diagram of named objects inside of namespaces - Labels and selectors - Diagram of objects inside a namespace with a label • Topic 2 – Kubernetes Master *Accordion interaction*	Explain the purpose of Kubernetes objects 1.4.2.2. Kubernetes Master	9 minutes	Articulate Rise and VYOND
Lesson 4 – Basic Kubernetes Objects *Design Strategy:* • Topic 1 – Pods *Process block showing the progression* - Simplest unit in Kubernetes - Represents processes running in your cluster - Encapsulates a container (or sometimes multiple) - Replicating a Pod serves to scale an application horizontally - States of Pods *Accordion interaction* - Difference between a container and a pod *Create VYOND video*	1.4.2. Describe the key components of Kubernetes 1.4.2.4. Pods 1.3.5. Describe the key lifecycle stages of a pod 1.3.4. Explain the difference between a container and a pod 1.4.2.5. Controllers (Deployment, ReplicaSet)	9 minutes	Articulate Rise and VYOND

• Topic 2 – ReplicaSet *Process block showing the progression* - Maintains a set of identical Pods - Definition consists of: o Selector to identify which Pods it can acquire o Number of replicas o Pod template - Generally encapsulated by a Deployment • Topic 3 – Deployment *Process block showing the progression* - Provides updates for Pods and ReplicaSets - Runs multiple replicas of your application - Suitable for stateless applications - Update triggers a rollout			
Lesson 5 – Kubectl CLI *Design Strategy:* • Topic 1 – Kubectl as a tool *Text block* • Topic 2 – Commands *Create VYOND video* - Explain each command and show examples of how they are used o Imperative o Declarative	Explain the Kubernetes CLI, kubectl	7 minutes	Articulate Rise and VYOND
Lesson 6 – Using Kubernetes *Design Strategy:* • Topic 1 – Create an application *text treatment and images with code snippets and image/text for outputs* - Apply command - Viewing resources - Listing resources - Getting a resource - Creating a resource	Create an application with Kubernetes	5 minutes	Articulate Rise

Section 1 Module: 1.4 - HLDD

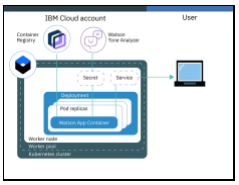
Knowledge Check *Storyline series of M/C, T/F, drag/drop questions to review material in this section (6 questions; 1 from each module)*	< Need to run these past Ryan for input	3 minutes	Articulate Storyline
	TOTAL TIME	**44 min**	

Section 1 Module: 1.4 - HLDD

Section 3 Development Resource(s)

- Containers and Kubernetes course (new) from IBM
- Persistent Volumes and Persistent Volume Claims: https://kubernetes.io/docs/concepts/storage/persistent-volumes/
- Identify and describe the different probes available with Kubernetes: https://kubernetes.io/docs/concepts/workloads/pods/pod-lifecycle/#container-states
- Explain the basic policies supported by Kubernetes: https://unofficial-kubernetes.readthedocs.io/en/latest/concepts/configuration/manage-compute-resources-container/ and https://kubernetes.io/docs/concepts/policy/resource-quotas/ and https://cloud.google.com/blog/products/gcp/kubernetes-best-practices-resource-requests-and-limits

Section 3: Managing Applications with Kubernetes

Lesson # and Name	Learning Objective	Time Estimate	Tool
Introduction • **Topic 0 – Introduction** *Image/text block (show IBM image to right)* - Throughout this module, we'll look at key concepts for managing Kubernetes applications including replicasets, auto-scaling, limitations and resource quotas, rolling updates, ConfigMaps and secrets, persistent volumes and persistent volume claims, service bindings, and Kubernetes probes.	N/A	1 minute	Articulate Rise Note: Introduction image for SME reference: 
Lesson 1 – ReplicaSet *Design Strategy:* • Topic 1 – YAML Descriptor *Image/with markers or hot spots* - Show example YAML Descriptor for already built application. Point out Descriptor.	1.3.6. Describe the key elements of a container specification 1.4.2.5. Controllers	9 minutes	Articulate Rise and VYOND

- Point out matchlables and app. - Point out template specs. • Topic 2 –Deployment *Code snippet block; output icon and output graphic* - Now we will create our deployment using the kubectl CLI and the create or apply command. - Show input and output of using the 'get deploy' command • Topic 3 – Enable public endpoint *Code snippet block; output icon and output graphic* - Explain 'expose' command. - Show input and output of using the 'get deploy' command • Topic 4 – Problems with our current deployment *Vyond video; vyond.com* - How to handle growing demands? - How to handle outages? - How to minimize downtime? - How to add auto restarts on interruptions? • Topic 5 – ReplicaSet Introduction *Text block & list block* - Replicate pods for redundancy - Maintain desired state - Supersede replicacontroller • Topic 6 – ReplicaSet – Deployments - Pods *Text/graphic block* - Point out template, metadata, labels, and app - Loosely coupled yet interconnected - Connected using labels • Topic 7 – ReplicaSet – How to Create One *Code snippet*	(Deployment, ReplicaSet)		

- ReplicaSets are automatically created when you create deployment - Use 'get replicaset' command - By default, generates to one single pod but that can be changed • Topic 8 – ReplicaSet – How to Create One *Code snippet (use slides 11&12 from Module 3.1 Containers and Kubernetes)* - ReplicaSets are automatically created when you create deployment - Use 'get replicaset' command - By default, generates to one single pod but that can be changed - if you describe the pod, you will see that it is "Controlled by:" the same replicaset. • Topic 9 – ReplicaSet from Scratch *Code snippet (use slide 13 from Module 3.1 Containers and Kubernetes) Note block* - Apply a yaml file with the kind attribute set to ReplicaSet. - Point our replicas: 1 and explain. - Note: While it is possible to create a replicaset without a deployment, it is recommended to create a deployment instead. • Topic 10 – ReplicaSet – Scale Existing Deployment *Code snippets and output icon with graphic; text* - First, create deployment - Use the "scale" command - kubectl get pods - kubectl get deploy - "replicas" to 3 - Get pods - Show output • Topic 11 – ReplicaSet – Does it work?			

Code snippets and output icon with graphic; text - kubectl get pods - kubectl delete pod hello-kubernetes-5655b546f8-5mflw - Show that new pod created automatically to replace deleted pod			
Lesson 2: Autoscaling • Topic 1 – Introduction *Text block with heading* - This lesson focuses on how to the autoscale feature to dynamically scale an application and accommodate change in demand. • Topic 2 - Autoscaling *Vyond video; vyond.com* - ReplicaSet works with a set number of pods - Horizontal Pod Autoscaler (HPA) enables scaling up and down as needed - Can configure based on desired state of CPU, memory, etc. - • Topic 3 – Autoscaling – How to Create One *Code snippets; text (use slides 4&5 of Containers and Kubernetes Course Module 3.2)* - kubectl get pods - kubectl get rs - to autoscale, we can simply use the autoscale command with some attributes. - Min is the number of pods - Max is minimum the number of maximum pods - Cpu-percent is the trigger to create new pods. This tells the system, "If the cpu usage hits 10% across the cluster, create a new pod."	1.3.6. Describe the key elements of a container specification	4 minutes	Articulate Rise and VYOND

- Show code snippet with autoscaled deployment set to 2; explain the ReplicaSet is owned by the deployment • Topic 4 – HPA Created Behind the Scenes *Code snippets; text* - A Horizontal Pod Autoscaler (HPA) is created behind the scenes to manage the autoscaling feature of the deployment. • Topic 5 – Autoscaling from Scratch *Code snippets; text* - A Horizontal Pod - You can create a standalone HPA using the following **YAML**, but we recommend using the "autoscale" command on the deploy/rs - Point out kind: HorizontalPodAutoscaler - Point out specs including minReplicas, maxReplicas, and targetCPUUtilizationPercentage: 10 - Note: Even though this is available to you, it is recommended to use the autoscale command instead. *(note block)*			
Lesson 3: Basic Policies Supported by Kubernetes • Topic 1 – Limit Ranges *Image & text block followed by accordion interaction (use slide 2 of Mod 2a of Containers and Kubernetes* - By default, containers run with unbounded compute resources on a Kubernetes cluster. - With resource quotas, cluster administrators can restrict resource consumption and creation on a namespace basis.	1.4.3. Explain the basic policies supported by Kubernetes 1.4.3.1 Limiting container compute resources 1.4.3.2 Limiting pod compute resources	7 minutes	Articulate Rise and VYOND

- A LimitRange is a policy to constrain resource allocations (to Pods or Containers) in a namespace. • Topic 2 – Resource Quotas *Text block and text on image for definition of resource quota (use slide 3 of Mod 2a of Containers and Kubernetes* - A resource quota, defined by a ResourceQuota object, provides constraints that limit aggregate resource consumption per namespace. • Topic 3 – Pod Security Policies *Text block followed by VYOND video. (use slide 4 of Mod 2a of Containers and Kubernetes* - When you specify a Pod, you can optionally specify how much CPU and memory (RAM) each Container needs. (intro) - Resource Types - Resource Requests and limits of Pods and Containers (*VYOND video; vyond.com)* • Topic 4 – Limiting pod compute resources *Text block followed by table of Resource Names and Description (use slide 5 of Mod 2a of Containers and Kubernetes* - Although requests and limits can only be specified on individual Containers, it is convenient to talk about Pod resource requests and limits. A Pod resource request/limit for a particular resource type is the sum of the resource requests/limits of that type for each Container in the Pod.	1.4.3.3 Limiting storage resources		

- You can limit the total sum of storage resources that can be requested in a given namespace. - In addition, you can limit consumption of storage resources based on associated storage-class. - *Insert table of Resource Names and Descriptions here.* • Topic 4a – Example *Text block followed by table of Resource Names and Description (use slide 5 of Mod 2a of Containers and Kubernetes* - For example, if an operator wants to quota storage with gold storage class separate from bronze storage class, the operator can define a quota as follows: - `gold.storageclass.storage.k8s.io/requests.storage: 500Gi` - `bronze.storageclass.storage.k8s.io/requests.storage: 100Gi` - In release 1.8, quota support for local ephemeral storage is added as an alpha feature: - *Insert table here of Resource Name & Description.* • Topic 5 – Limiting Storage Resources *Text block followed by table of (use slide 6 of Mod 2a of Containers and Kubernetes* - Need content here.			
Lesson 4: Probes Available within Kubernetes *Design Strategy:* • Topic 1 – Probes *text block and tabbed interaction* - Readiness Probe - Liveness Probe - Startup Probes for slow starting containers	1.4.4 Identify and describe the different probes available with Kubernetes 1.4.4.1. Readiness Probe 1.4.4.2. Liveness Probe 1.4.4.3. Startup Probes	5 minutes	Articulate Rise

	for slow starting containers		
Lesson 5: Rolling Updates • Topic 1 – Rolling Updates *Text block for intro text.* • Topic 2 – Steps to Prepare your Application for Rolling Updates *Process block with image of code for each (use slides 4/5 of Mod 3.3 Rolling Updates in Containers and Kubernetes course)* - Step 1: add liveness and readiness probes to your deployments. This ensures deployments are marked ready appropriately. - Step 2: Add rolling update strategy to your YAML file - *Storyline file to use graphics animation with VO.* • Topic 3 – Example: Rolling out an update on application *Text and code snippet; text and output image (use slide 6 of Mod 3.3 Rolling Updates in Containers and Kubernetes course)* - Step 1: add liveness and readiness probes to your deployments. This ensures deployments are marked ready appropriately. - Step 2: Add rolling update strategy to your YAML file - *Storyline file to use graphics animation with VO.* • Topic 4 – Rolling Updates: Changing Versions *Text and code snippet; text and output image (use slide 7 of Mod 3.3 Rolling Updates in Containers and Kubernetes course)*	1.3.6. Describe the key elements of a container specification	6 minutes	Articulate Rise

- We have a new image for our application with a different message: “Hello worldv2!” • Topic 4a – Example *Text and code snippets (use slide 8 of Mod 3.3 Rolling Updates in Containers and Kubernetes course)* - First, we need to upload this new image to Docker Hub - Note: Show the simple Docker commands and note that these are unrelated to Kubernetes. *(notes block)* • Topic 5 – Applying the new image to our deployment *Text and code snippet (use slide 8 of Mod 3.3 Rolling Updates in Containers and Kubernetes course)* - First command shows we have 3 pods. - The second command sets the image flag to the updated and tag image on dockerhub. We get a message saying the image has been updated. Let’s see if that actually happened! • Topic 6 – Check the Status of a Rollout *Text and code snippet & output; graphic of app output (use slide 10 of Mod 3.3 Rolling Updates in Containers and Kubernetes course)* - rollout status” command - Show API output: deployment "hello-kubernetes" successfully rolled out”. - *Show app output in a graphic.* • Topic 7 – Rollback to v1 *Text and code snippet & output (use slide 11 of Mod 3.3 Rolling Updates in Containers and Kubernetes course)*			

- First command: kubectl rollout undo deployments/hello-kubernetes - Second command: kubectl get pods • Topic 7a – Rollback to v1 (cont.) *Text and code snippet & output (use slide 12 of Mod 3.3 Rolling Updates in Containers and Kubernetes course)* - First command: » kubectl rollout status deployments/hello-kubernetes - Second command: » kubectl describe pods			
Lesson 6: ConfigMaps and Secrets *Design Strategy:* • Topic 1 – Configuration: - Current state *text block, code snippet and image* • Topic 2 – ConfigMaps and Secrets: Introduction *text box with bullets* • Topic 3 –ConfigMaps: - Environment variable *image of code snippet with section highlighted, text block, code snippet and image* - ConfigMap string literal *process block using code snippets and image* - ConfigMap properties file *text block, code snippets and image* - ConfigMap YAML *text block, code snippets and image* • Topic 4 – Secrets: - Create *text block and code snippets* - Secrets *text block, code snippets image with section highlighted* - Use with environment variables *text block, code snippet and image* - Use with volume mounts *text block and code snippets*	1.3.6. Describe the key elements of a container specification 1.4.2.6. Secrets and ConfigMap Storing configurations in variables Different ways to create a ConfigMap What a Secret is and the different ways to create a Secret	9 minutes	Articulate Rise

Lesson 7: Persistent Volumes and Persistent Volume Claims *Design Strategy:* • Topic 1 – Volumes *Accordion interaction* - PersistentVolume - PersistentVolumeClaim	Describe how to use API resources to manage storage 1.3.6. Describe the key elements of a container specification 1.4.2.7. Persistent Volumes and Persistent Volume Claims	3 minutes	Articulate Rise
Lesson 8: Service Binding • Topic 1: Service binding *text box* • Topic 2: IBM Cloud Services *text box followed by image* • Topic 3: Create a service binding *Process block showing the steps* • Topic 4: Service binding: - Secrets CLI *text block and code snippet* - Secrets dashboard *text/image block* - Use in apps *Process block showing the steps* - Example *text block and code snippet*	1.3.6. Describe the key elements of a container specification Define a service binding Explain how to bind and external service to your deployment	4 minutes	Articulate Rise
	TOTAL TIME	**48 minutes**	