**Item #1**

Create a docker file that starts the jenkins server. It should start with httpPort set to 8090 and prefix="umlJenkins".

Use instructions here and 'docker run' to start the container.

Submitproof you have the container running. In particular, the command

docker logs <container id>

Also, the output to the command:

docker exec -it <container id> env

(Where the container id is taken from "docker ps" output.)

*I have assembled a Dockerfile that builds an image of Jenkins with many of the plugins I have found useful over the years. I also recently started using configuration-as-code to quickly setup the Jenkins server with authentication and users.*

*Total build time was under 9 mins , however this saves a lot of time going back and forth to add plugins and other options on the fly or in the Jenkins settings.*

**RUN LOG**

[23:53:27] week\_2: $ docker run --name jenkins --rm -p 8090:8090 --env JENKINS\_ADMIN\_ID=umladmin --env JENKINS\_ADMIN\_PASSWORD=umlpass jenkins:casc

Running from: /usr/share/jenkins/jenkins.war

webroot: EnvVars.masterEnvVars.get("JENKINS\_HOME")

2021-09-10 03:53:44.291+0000 [id=1] INFO org.eclipse.jetty.util.log.Log#initialized: Logging initialized @859ms to org.eclipse.jetty.util.log.JavaUtilLog

2021-09-10 03:53:44.423+0000 [id=1] INFO winstone.Logger#logInternal: Beginning extraction from war file

2021-09-10 03:53:45.477+0000 [id=1] INFO org.eclipse.jetty.server.Server#doStart: jetty-9.4.42.v20210604; built: 2021-06-04T17:33:38.939Z; git: 5cd5e6d2375eeab146813b0de9f19eda6ab6e6cb; jvm 11.0.12+7

2021-09-10 03:53:45.824+0000 [id=1] INFO o.e.j.w.StandardDescriptorProcessor#visitServlet: NO JSP Support for /umlJenkins, did not find org.eclipse.jetty.jsp.JettyJspServlet

2021-09-10 03:53:45.882+0000 [id=1] INFO o.e.j.s.s.DefaultSessionIdManager#doStart: DefaultSessionIdManager workerName=node0

2021-09-10 03:53:45.882+0000 [id=1] INFO o.e.j.s.s.DefaultSessionIdManager#doStart: No SessionScavenger set, using defaults

2021-09-10 03:53:45.884+0000 [id=1] INFO o.e.j.server.session.HouseKeeper#startScavenging: node0 Scavenging every 600000ms

2021-09-10 03:53:46.483+0000 [id=1] INFO hudson.WebAppMain#contextInitialized: Jenkins home directory: /var/jenkins\_home found at: EnvVars.masterEnvVars.get("JENKINS\_HOME")

2021-09-10 03:53:46.835+0000 [id=1] INFO o.e.j.s.handler.ContextHandler#doStart: Started w.@3a01773b{Jenkins v2.303.1,/umlJenkins,file:///var/jenkins\_home/war/,AVAILABLE}{/var/jenkins\_home/war}

2021-09-10 03:53:46.892+0000 [id=1] INFO o.e.j.server.AbstractConnector#doStart: Started ServerConnector@3f0846c6{HTTP/1.1, (http/1.1)}{0.0.0.0:8090}

2021-09-10 03:53:46.892+0000 [id=1] INFO org.eclipse.jetty.server.Server#doStart: Started @3464ms

2021-09-10 03:53:46.894+0000 [id=24] INFO winstone.Logger#logInternal: Winstone Servlet Engine running: controlPort=disabled

2021-09-10 03:53:47.221+0000 [id=31] INFO jenkins.InitReactorRunner$1#onAttained: Started initialization

2021-09-10 03:53:48.213+0000 [id=38] WARNING hudson.ClassicPluginStrategy#createClassJarFromWebInfClasses: Created /var/jenkins\_home/plugins/jquery-ui/WEB-INF/lib/classes.jar; update plugin to a version created with a newer harness

2021-09-10 03:53:49.545+0000 [id=31] WARNING hudson.ClassicPluginStrategy#createClassJarFromWebInfClasses: Created /var/jenkins\_home/plugins/job-dsl/WEB-INF/lib/classes.jar; update plugin to a version created with a newer harness

2021-09-10 03:53:50.743+0000 [id=36] WARNING hudson.ClassicPluginStrategy#createClassJarFromWebInfClasses: Created /var/jenkins\_home/plugins/git-userContent/WEB-INF/lib/classes.jar; update plugin to a version created with a newer harness

2021-09-10 03:53:51.865+0000 [id=35] INFO jenkins.InitReactorRunner$1#onAttained: Listed all plugins

WARNING: An illegal reflective access operation has occurred

WARNING: Illegal reflective access by com.google.inject.internal.cglib.core.$ReflectUtils$2 (file:/var/jenkins\_home/war/WEB-INF/lib/guice-4.0.jar) to method java.lang.ClassLoader.defineClass(java.lang.String,byte[],int,int,java.security.ProtectionDomain)

WARNING: Please consider reporting this to the maintainers of com.google.inject.internal.cglib.core.$ReflectUtils$2

WARNING: Use --illegal-access=warn to enable warnings of further illegal reflective access operations

WARNING: All illegal access operations will be denied in a future release

2021-09-10 03:53:59.247+0000 [id=35] INFO jenkins.InitReactorRunner$1#onAttained: Prepared all plugins

2021-09-10 03:53:59.257+0000 [id=36] INFO jenkins.InitReactorRunner$1#onAttained: Started all plugins

2021-09-10 03:54:00.816+0000 [id=40] INFO o.j.h.p.r.RepositoryConfiguration#load: no saved repositories found, initializing list using maven central default

2021-09-10 03:54:00.818+0000 [id=40] INFO o.j.h.p.r.RepositoryConfiguration#lambda$setRepositories$0: adding repository [Repository [id=central, url=https://repo1.maven.org/maven2, credentialsId=null]]

2021-09-10 03:54:01.361+0000 [id=39] INFO jenkins.InitReactorRunner$1#onAttained: Augmented all extensions

2021-09-10 03:54:01.770+0000 [id=40] INFO jenkins.InitReactorRunner$1#onAttained: System config loaded

2021-09-10 03:54:03.223+0000 [id=32] INFO jenkins.InitReactorRunner$1#onAttained: System config adapted

2021-09-10 03:54:03.225+0000 [id=31] INFO jenkins.InitReactorRunner$1#onAttained: Loaded all jobs

2021-09-10 03:54:03.228+0000 [id=30] INFO jenkins.InitReactorRunner$1#onAttained: Configuration for all jobs updated

2021-09-10 03:54:03.240+0000 [id=55] INFO hudson.model.AsyncPeriodicWork#lambda$doRun$0: Started Download metadata

2021-09-10 03:54:03.260+0000 [id=55] INFO hudson.util.Retrier#start: Attempt #1 to do the action check updates server

2021-09-10 03:54:03.402+0000 [id=32] INFO jenkins.InitReactorRunner$1#onAttained: Completed initialization

2021-09-10 03:54:03.463+0000 [id=23] INFO hudson.WebAppMain$3#run: Jenkins is fully up and running

2021-09-10 03:54:13.422+0000 [id=55] INFO h.m.DownloadService$Downloadable#load: Obtained the updated data file for hudson.tasks.Maven.MavenInstaller

2021-09-10 03:54:13.515+0000 [id=55] INFO h.m.DownloadService$Downloadable#load: Obtained the updated data file for hudson.plugins.gradle.GradleInstaller

2021-09-10 03:54:13.558+0000 [id=55] INFO h.m.DownloadService$Downloadable#load: Obtained the updated data file for hudson.plugins.groovy.GroovyInstaller

2021-09-10 03:54:13.595+0000 [id=55] INFO h.m.DownloadService$Downloadable#load: Obtained the updated data file for hudson.tasks.Ant.AntInstaller

2021-09-10 03:54:14.049+0000 [id=55] INFO h.m.DownloadService$Downloadable#load: Obtained the updated data file for hudson.tools.JDKInstaller

2021-09-10 03:54:14.050+0000 [id=55] INFO hudson.util.Retrier#start: Performed the action check updates server successfully at the attempt #1

2021-09-10 03:54:14.058+0000 [id=55] INFO hudson.model.AsyncPeriodicWork#lambda$doRun$0: Finished Download metadata. 10,810 ms

**ENV Output**

[00:03:52] DevOps: $ dei jenkins env

PATH=/opt/java/openjdk/bin:/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin

HOSTNAME=6228a108d214

TERM=xterm

JENKINS\_ADMIN\_ID=umladmin

JENKINS\_ADMIN\_PASSWORD=umlpass

JENKINS\_HOME=/var/jenkins\_home

JENKINS\_SLAVE\_AGENT\_PORT=50000

REF=/usr/share/jenkins/ref

JENKINS\_VERSION=2.303.1

JENKINS\_UC=https://updates.jenkins.io

JENKINS\_UC\_EXPERIMENTAL=https://updates.jenkins.io/experimental

JENKINS\_INCREMENTALS\_REPO\_MIRROR=https://repo.jenkins-ci.org/incrementals

COPY\_REFERENCE\_FILE\_LOG=/var/jenkins\_home/copy\_reference\_file.log

JAVA\_HOME=/opt/java/openjdk

CASC\_JENKINS\_CONFIG=/var/jenkins\_home/casc.yaml

JENKINS\_OPTS=--httpPort=8090 --prefix=/umlJenkins

JAVA\_OPTS=-Djenkins.install.runSetupWizard=false

PYTHONUNBUFFERED=1

HOME=/var/jenkins\_home

**Screen Shots**

Graphical user interface, application

Description automatically generated

Graphical user interface, text

Description automatically generated

Graphical user interface, text, application

Description automatically generated

**ITEM #2**

Create a Kubernetes deployment that runs the MySQL database. The deployment should have a persistent volume. Create a MySQL client that accesses MySQL. Provide evidence that the client can access the database, e.g. a screenshot.

Describe the steps you performed to create the persistent volume, deployment, service, and client. Describe what each of the steps is doing in your own words.

1. First, setup a namespace to make managing the k8s resources easier. Set current context with the new namespace. Verify the namespace is set correctly.

[23:55:21] week\_2: $ k create namespace mysql

namespace/mysql created

[23:55:43] week\_2: $ kubens mysql

Context "docker-desktop" modified.

[23:58:57] week\_2: $ kgetns

namespace: mysql

1. Create a Persistent Volume & claim. The follow shows the apply command.

[00:02:14] week\_2: $ kg pv

NAME CAPACITY ACCESS MODES RECLAIM POLICY STATUS CLAIM STORAGECLASS REASON AGE

mysql-pv-volume 1Gi RWO Retain Bound mysql/mysql-pv-claim manual 9s

[00:02:17] week\_2: $ kg pvc

NAME STATUS VOLUME CAPACITY ACCESS MODES STORAGECLASS AGE

mysql-pv-claim Bound mysql-pv-volume 1Gi RWO manual 14s

1. Apply the deployment file. This gets a request to our waiting claim on “mysql-pv-claim”, deploys the MySql image and sets some reasonable limits for the pod so as not to overwhelm the host server.

[00:03:46] week\_2: $ kap -f k8\_mysql\_dep.yaml

service/mysql created

deployment.apps/mysql created

Submit output of commands:

kubectl describe deployment mysql

Name: mysql

Namespace: mysql

CreationTimestamp: Sun, 12 Sep 2021 00:05:15 -0400

Labels: <none>

Annotations: deployment.kubernetes.io/revision: 1

Selector: app=mysql

Replicas: 1 desired | 1 updated | 1 total | 1 available | 0 unavailable

StrategyType: Recreate

MinReadySeconds: 0

Pod Template:

Labels: app=mysql

Containers:

mysql:

Image: mysql:5.6

Port: 3306/TCP

Host Port: 0/TCP

Limits:

cpu: 2

memory: 1Gi

Environment:

MYSQL\_ROOT\_PASSWORD: password

Mounts:

/var/lib/mysql from mysql-persistent-storage (rw)

Volumes:

mysql-persistent-storage:

Type: PersistentVolumeClaim (a reference to a PersistentVolumeClaim in the same namespace)

ClaimName: mysql-pv-claim

ReadOnly: false

Conditions:

Type Status Reason

---- ------ ------

Available True MinimumReplicasAvailable

Progressing True NewReplicaSetAvailable

OldReplicaSets: <none>

NewReplicaSet: mysql-ff5964794 (1/1 replicas created)

Events:

Type Reason Age From Message

---- ------ ---- ---- -------

Normal ScalingReplicaSet 87s deployment-controller Scaled up replica set mysql-ff5964794 to 1

kubectl describe pvc mysql-pv-claim

Name: mysql-pv-claim

Namespace: mysql

StorageClass: manual

Status: Bound

Volume: mysql-pv-volume

Labels: <none>

Annotations: pv.kubernetes.io/bind-completed: yes

pv.kubernetes.io/bound-by-controller: yes

Finalizers: [kubernetes.io/pvc-protection]

Capacity: 1Gi

Access Modes: RWO

VolumeMode: Filesystem

Used By: mysql-ff5964794-n2pkh

Events: <none>

kubectl get pods -l app=mysql

NAME READY STATUS RESTARTS AGE

mysql-ff5964794-n2pkh 1/1 Running 0 3m7s

Accessing the Database

kubectl run -it --rm --image=mysql:5.6 --restart=Never mysql-client -- mysql -h mysql -p password

Text

Description automatically generated

**ITEM #3**

Define what "DevOps" is. In your definition, give three ways it makes software development faster.

DevOps is philosophy supporting Continuous Development & Continuous Integration (CI/CD). As a methodology, this includes using a variety of tools (containers, images, etc.) to create pipelines to efficiently develop, deploy & maintain code assets.

1. One important way that DevOps makes development faster is by having a repeatable, consistent method in which do deploy & update software. Using version control like GitLab & deployment tools like Jenkins allow the developers to focus on the code and not the process of deployment itself.
2. Another way DevOps supports quick development is by having environments that can be updated instead of needing to start over. An example could be Docker images, there is no need to deploy an entirely new code base when a kernel package is updated. Using Dockerfiles makes it easy to just add the new package (or update an old one) by changing the layer and rebuilding the image for immediate deployment.
3. Finally, use of VCS like Git, coupled with a deployment tool like Harness, allows the team to work on separate tasks, integrating the changes into the code base incrementally. If something fails, you can simply fall back to the last version and fix the issues. When done in a dev/prod type environment, this means only running code is in the prod environment until the team decides its ok to promote the development code.

In this class, we chose to run Jenkins in Kubernetes. We could have run it externally to Kubernetes and only had the results of the pipeline run in Kubernetes. What are 3 pros and 3 cons of running the full Jenkins pipeline in Kubernetes?

Pros:

Keeping Jenkins inside Kubernetes can simply the communication process. Pods can easily be configured to “talk” to each other without having routing issues common between separated servers.

Security is improved between Jenkins and the K8s pods by enabling pod to pod communication without outside routing capability

The whole deployment pipeline from Jenkins to application can be defined in files that are easily deployment with minimal commands

Cons:

Having Jenkins inside K8s could lead to resource constraints, requiring more hardware (or larger cloud services) to meet the demand of both the application and the deployment orchestration tools.

Jenkins tightly coupled to the K8s could make it difficult to deploy pipelines to other servers – such as K8s in the cloud when you need to deploy code on a local server.

Having both the pipeline tool and the application together can lead to difficulty in maintaining the code base. Is the Jenkins instance going to manage both Dev & Prod? Could this lead to accidently deploying non-working code to Prod?