|  |  |
| --- | --- |
| **Question #** | **Response** |
| 5 | **Image:** Something like a template to create and run containers which comes equipped with any specified dependencies. This is built once via a Dockerfile.  **Container:** The hardware-virtualized runtime that is used to actually house your running application (specified by your Docker image earlier).  **Registry:** A server-side application that allows you to collect, manage and distribute all of your Docker images. |
| 6 | **FROM openjdk:11**  Specify the Docker image used as a basis for your Docker image. This command pulls the image version if it is not detected locally.  **RUN mkdir /app**  Create a new directory in your container image  **COPY out/production/HelloWorldDocker/ /app**  Copy all files (Java Classfile in this case) from one directory to the directory in your container image  **WORKDIR /app**  Set the working directory for any commands issued in the Docker file  **CMD java Main**  Run the pre-compiled Java class copied into your container image |
| 7 | To stop the container (if running in the foreground) open another terminal and enter “docker stop” followed by either the name or id of the container obtainable using the “docker ps” command. To delete the container, use “docker rm “ and the respective container name or id. |
| 8 | A multi-container application is more in line with what you would be building in a production environment, that is a fully fleshed-out application with multiple components deployed and running in different containers |
| 9 | Containers communicate via ad-hoc networks, called bridge networks, bi-directionally. Much like in un-containerized applications, communication is facilitated by the exposition of ports (ports must be mapped to the local machine’s ports upon container launch for this to work) |
| 10 | **docker rm** <container name or id> – add in the “-f” option to force-stop the specified container |
| 11 | **docker build** – Build images from a Dockerfile which can be use to run your containers  **docker network create** – Create a bridge network for running containers to be able to communicate  **docker network connect** – Connect the specified container to the bridge network  **docker-compose up – S**tart up multiple docker containers as specified in the “docker-compose.yml” file. Add in the “-d” flag to run the containers in the background |
| 13 | **gcloud config set project** - Set the environment to use your GCP project  **gcloud config set compute/**zone - Specify the compute zone used for your project  **gcloud container clusters create --num-nodes=1** - Create a GKE Cluster for running your application with 1 node  **gcloud container clusters get-**credentials - Set credentials so that KubeCTL can access your cluster  **gcloud builds submit --tag** - create an image from local project files, add a tag, and commit it to GCR for later deployment  **kubectl create deployment** - Push the created image specified to a GKE cluster specified |
| 17 | **Pods:** Groups of containers  **Service:** Logical grouping of pods which perform the same function meant for ease of discovery and application routing  **Deployment:** A physical grouping of pods which gets assigned to a particular computing cluster  **Node:** The smallest assignable unit of hardware computing power available to be assigned to a cluster |
| 18 | Replicas are simply copies of pods that are identical and simultaneously running – through the use of Replicas, horizontal scaling is achieved |
| 19 | **ClusterIP:** Allows communication between pods within the confines of the service  **NodePort:** Exposes the collection of pods to other services outside of the cluster by adding a cluster-wide port which can be used for external communications  **LoadBalancer:** Exposes the collection of pods specifically to a Load Balancing service provided by the third party cloud Kubernetes Service provider (e.g. Google, AWS, etc.)  **ExternalName:** Often used within Kubernetes to represent an external data source, this service type maps the collection of pods to a registered DNS name as opposed to a specified name. |