DESCRIPTION

You are working as a DevOps engineer in an IT firm. You have been asked to create a Redis-based Docker image and deploy it on a Kubernetes cluster.

**Background of the problem statement:**

Your organization wants to use Redis in a Kubernetes cluster for the data storage and caching purpose. The development team has asked you to create a Redis-based Docker image using a Dockerfile and deploy this image on a Kubernetes cluster.

You have also been asked to publish this image on your organization's Docker Hub account so that other team members can also access this image.

**You must use the following:**

* Docker CLI: To create the Docker image using a Dockerfile
* Docker Hub: To publish the image
* Kubectl: To deploy the image on a Kubernetes cluster

**Following requirements should be met:**

* Follow the above-mentioned specifications
* Make sure you create an account on Docker Hub to push the Docker image
* Document the step-by-step process involved in completing this task

Solution Guidelines

1. Create Dockerfile:

Graphical user interface, text

Description automatically generated

1. Build docker image:
   1. $ docker build -t <name-for-image> /path/to/dockerfile

Text

Description automatically generated

* 1. Run the following command to verify the image has been created successfully:
     1. docker images

1. Run docker container using the image:
   1. $ docker run -itd --name <give-container-a-name> <image\_id or name>
   2. Validate that the container is running.
      1. 
2. Configure Redis-server: Modify supervised directive from no to systemd as shown below
   1. $ docker exec -it <name-of-container> bash
      1. You should be at the prompt inside the container



* 1. $ vi /etc/redis/redis.conf
     1. Text

        Description automatically generated

1. Start Redis-server and make sure it is running correctly.
   1. $ service redis-server start



* 1. $ redis-cli
     1. 
  2. $ ping
     1. 

1. Commit the changes made to a new image that will be shared with the company Docker hub account
   1. Exit the container to return to host prompt: you will need to exit twice
   2. $ docker commit <container-id/name>
      1. Text

         Description automatically generated with medium confidence
2. Push the image to the company Docker hub account:
   1. Text

      Description automatically generated

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Kubernetes Deployment

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Create EKS cluster (AWS)

1. Create a cluster config yaml file:
   1. Text

      Description automatically generated