**Lesson 5 Demo 6**

**Deployment of Flask Application with Redis**

**Objective:** Learn to deploy a Flask application with Redis

**Tools required:** Docker Desktop and kubectl

**Prerequisites:** Must have a Docker account or create one at <https://www.docker.com/>

Steps to be followed:

1. Creating a new directory and adding the required files
2. Creating and tagging the Flask image
3. Logging into Docker and pushing the Flask image
4. Creating the Redis and Flask deployments
5. Creating the Redis and Flask services
6. Verifying the Flask application

**Step 1: Creating a new directory and adding the required files**

1. Create and navigate to the directory **redis\_flask** using the following commands:

**mkdir redis\_flask**

**cd redis\_flask**

**A screenshot of a computer

Description automatically generated**

1. Add the code given below to **redis\_flask/app.py**:

**vi app.py**

**from flask import Flask**

**from redis import Redis**

**app = Flask(\_\_name\_\_)**

**redis = Redis(host='redis', port=6379)**

**@app.route('/')**

**def hello():**

**count = redis.incr('hits')**

**return 'Hello from Docker! I have been seen {} times.\n'.format(count)**

**if \_\_name\_\_ == "\_\_main\_\_":**

**app.run(host="0.0.0.0", debug=True)**

**Graphical user interface, text

Description automatically generated**

1. Add the code given below to the **Dockerfile**:

**vi Dockerfile**

**FROM python:3.4-alpine**

**ADD . /code**

**WORKDIR /code**

**RUN pip install -r requirements.txt**

**CMD ["python", "app.py"]**

Text

Description automatically generated

1. Add the following code to **requirements.txt**:

**vi requirements.txt**

**flask**

**redis**

Shape

Description automatically generated with medium confidence

**Step 2: Creating and tagging the Flask image**

1. Create a Flask app image using the following command:

**docker build -t flask\_image .**

**Text

Description automatically generated**

1. Tag the image using the following command:

**docker tag flask\_image:latest <docker-id>/flask-image:flask\_image\_for\_redis**

**Graphical user interface, text

Description automatically generated**

A screenshot of a computer

Description automatically generated

|  |
| --- |
| **Note:** Replace **<docker-id>** with your docker username |

**Step 3: Logging into Docker and pushing the Flask image**

1. Log into Docker using the following command:

**docker login**

Text

Description automatically generated

1. Push the Flask image to the Docker repository:

**docker push <docker-id>/flask-image:flask\_image\_for\_redis**

Text

Description automatically generated

**Step 4: Creating the Redis and Flask deployments**

1. Add the following code to the **redis.yaml** file:

**vi redis.yaml**

**apiVersion: apps/v1**

**kind: Deployment**

**metadata:**

**creationTimestamp: null**

**labels:**

**app: redis**

**name: redis**

**spec:**

**replicas: 1**

**selector:**

**matchLabels:**

**app: redis**

**strategy: {}**

**template:**

**metadata:**

**creationTimestamp: null**

**labels:**

**app: redis**

**spec:**

**containers:**

**- image: redis**

**name: redis**

**resources: {}**

**status: {}**

Text

Description automatically generated

1. Create the Redis deployment using the following command:

**kubectl create -f redis.yaml**

Text

Description automatically generated

1. Add the following code to the **flask.yaml** file:

**vi flask.yaml**

**apiVersion: apps/v1**

**kind: Deployment**

**metadata:**

**creationTimestamp: null**

**labels:**

**app: flask**

**name: flask**

**spec:**

**replicas: 1**

**selector:**

**matchLabels:**

**app: flask**

**strategy: {}**

**template:**

**metadata:**

**creationTimestamp: null**

**labels:**

**app: flask**

**spec:**

**containers:**

**- image: 7337568/flask-image:flask\_image\_for\_redis**

**name: flask-image**

**resources: {}**

**status: {}**

|  |
| --- |
| **Note:** Replace the image repository with yours accordingly. |

Graphical user interface, text

Description automatically generated

1. Create the Flask deployment using the following command:

**kubectl create -f flask.yaml**

Text

Description automatically generated

**Step 5: Creating the Redis and Flask services**

1. Add the following code to the **redis-svc.yaml** file:

**vi redis-svc.yaml**

**apiVersion: v1**

**kind: Service**

**metadata:**

**creationTimestamp: null**

**labels:**

**app: redis**

**name: redis**

**spec:**

**ports:**

**- port: 6379**

**protocol: TCP**

**targetPort: 6379**

**selector:**

**app: redis**

**status:**

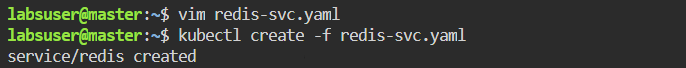
**loadBalancer: {}**

Text

Description automatically generated

1. Create the Redis service using the following command:

**kubectl create -f redis-svc.yaml**



1. Add the following code to the **flask-svc.yaml** file:

**vi flask-svc.yaml**

**apiVersion: v1**

**kind: Service**

**metadata:**

**creationTimestamp: null**

**labels:**

**app: flask**

**name: flask**

**spec:**

**ports:**

**- port: 5000**

**protocol: TCP**

**targetPort: 5000**

**selector:**

**app: flask**

**status:**

**loadBalancer: {}**

Text

Description automatically generated

1. Create the Flask service using the following command:

**kubectl create -f flask-svc.yaml**

Text

Description automatically generated

**Step 6: Verifying the Flask application**

1. Check if the Flask app is working using the following commands:

**kubectl get svc**

**curl 10.107.110.98:5000**

|  |
| --- |
| **Note:** Use your Flask service cluster IP accordingly |

Graphical user interface, text

Description automatically generated