MUHAMMAD OMER SIDDIQUI 19B-004-SE

ADNAN SAMAD 19B-011-SE

HAFIZ TAHA SHARIF 19B-035-SE

Task 1:

Implement nginx server over scalable deployment/POD.

Task 2:

Use persistent volume for the POD.

Task 3:

Deploy the mysql server with a single replica.

Task 4:

Use volume to store the mysql files.

Task 5:

Develop a single web page which only retrieve the list of mysql table data.

[For volumes you may use the git repo]

***NGINX-PVC***

apiVersion: v1

kind: PersistentVolume

metadata:

  name: pv-volume1

  labels:

    type: local

spec:

  storageClassName: hostpath

  capacity:

    storage: 1Gi

  accessModes:

    - ReadWriteOnce

  hostPath:

    path: "/run/desktop/mnt/host/d/assignment2"

---

apiVersion: v1

kind: PersistentVolumeClaim

metadata:

  name: nginx-pv-claim

spec:

  storageClassName: hostpath

  accessModes:

    - ReadWriteOnce

  resources:

    requests:

      storage: 1Gi

***MY-SQL PVC***

apiVersion: v1

kind: PersistentVolume

metadata:

  name: pv-volume2

  labels:

    type: local

spec:

  storageClassName: hostpath

  capacity:

    storage: 1Gi

  accessModes:

    - ReadWriteOnce

  hostPath:

    path: "/run/desktop/mnt/host/d/assignment2db"

---

apiVersion: v1

kind: PersistentVolumeClaim

metadata:

  name: mysql-pv-claim

spec:

  storageClassName: hostpath

  accessModes:

    - ReadWriteOnce

  resources:

    requests:

      storage: 1Gi

***PHP PVC***

apiVersion: v1

kind: PersistentVolume

metadata:

  name: pv-volume3

  labels:

    type: local

spec:

  storageClassName: hostpath

  capacity:

    storage: 1Gi

  accessModes:

    - ReadWriteOnce

  hostPath:

    path: "/run/desktop/mnt/host/d/assignment2php"

---

apiVersion: v1

kind: PersistentVolumeClaim

metadata:

  name: phpfpm-pv-claim

spec:

  storageClassName: hostpath

  accessModes:

    - ReadWriteOnce

  resources:

    requests:

      storage: 1Gi

A screenshot of a computer

Description automatically generated with medium confidence

***NGINX***

apiVersion: apps/v1

kind: Deployment

metadata:

  name: nginx-server

  labels:

    app: nginx

spec:

  replicas: 2

  selector:

    matchLabels:

      app: nginx

  template:

    metadata:

      labels:

        app: nginx

    spec:

      containers:

      - name: nginx

        image: nginx

        ports:

        - containerPort: 80

          name: "nignx-server"

        volumeMounts:

        - mountPath: "/run/desktop/mnt/host/d/assignment2"

          name: nginx-pv-storage

        - name: nginx-configmap-volume

          mountPath: /etc/nginx/conf.d/default.conf

          subPath: nginx.conf

      volumes:

      - name: nginx-pv-storage

        persistentVolumeClaim:

          claimName: nginx-pv-claim

      - name: nginx-configmap-volume

        configMap:

          name: nginx-configmap

---

apiVersion: v1

kind: Service

metadata:

  name: nginx-service

spec:

  type: NodePort

  selector:

    app: nginx

  ports:

    - protocol: TCP

      port: 80

      targetPort: 80

      nodePort: 30000

***PHP***

apiVersion: apps/v1

kind: Deployment

metadata:

  name: phpfpm

  labels:

    app: phpfpm

    layer: backend

spec:

  replicas: 2

  selector:

    matchLabels:

      app: phpfpm

  template:

    metadata:

      labels:

        app: phpfpm

    spec:

      containers:

      - name: phpfpm

        image: php:7.1-fpm-alpine

        ports:

        - containerPort: 9000

        env:

        - name: MYSQL\_HOST

          valueFrom:

            configMapKeyRef:

              name: nginx-configmap

              key: mysql-host

        - name: MYSQL\_DATABASE

          valueFrom:

            configMapKeyRef:

              name: nginx-configmap

              key: database-name

        - name: MYSQL\_USERNAME

          valueFrom:

            secretKeyRef:

              name: mysql-secret

              key: mysql-user

        - name: MYSQL\_PASSWORD

          valueFrom:

            secretKeyRef:

              name: mysql-secret

              key: mysql-password

        volumeMounts:

        - mountPath: /var/www/html/

          name: phpfpm-pv-storage

      volumes:

      - name: phpfpm-pv-storage

        persistentVolumeClaim:

          claimName: phpfpm-pv-claim

---

kind: Service

apiVersion: v1

metadata:

  name: phpfpm

  labels:

    app: phpfpm

    layer: backend

spec:

  selector:

    app: phpfpm

  ports:

    - protocol: TCP

      port: 9000

      targetPort: 9000

***MYSQL***

apiVersion: apps/v1

kind: Deployment

metadata:

  name: mysqldb

spec:

  selector:

    matchLabels:

      app: mysqldb

  strategy:

    type: Recreate

  template:

    metadata:

      labels:

        app: mysqldb

    spec:

      containers:

      - image: mysql:5.7

        name: mysqldb

        env:

        - name: MYSQL\_ROOT\_PASSWORD

          valueFrom:

            secretKeyRef:

              name: mysql-secret

              key: mysql-root-password

        - name: MYSQL\_DATABASE

          valueFrom:

            configMapKeyRef:

              name: nginx-configmap

              key: database-name

        - name: MYSQL\_USER

          valueFrom:

            secretKeyRef:

              name: mysql-secret

              key: mysql-user

        - name: MYSQL\_PASSWORD

          valueFrom:

            secretKeyRef:

              name: mysql-secret

              key: mysql-password

        ports:

        - containerPort: 3306

          name: mysqldb

        volumeMounts:

        - name: mysql-persistent-storage

          mountPath: /var/lib/mysql

        - name: mysql-initdb

          mountPath: /docker-entrypoint-initdb.d

      volumes:

      - name: mysql-persistent-storage

        persistentVolumeClaim:

          claimName: mysql-pv-claim

      - name: mysql-initdb

        configMap:

          name: nginx-configmap

---

apiVersion: v1

kind: Service

metadata:

  name: mysql-service

spec:

  selector:

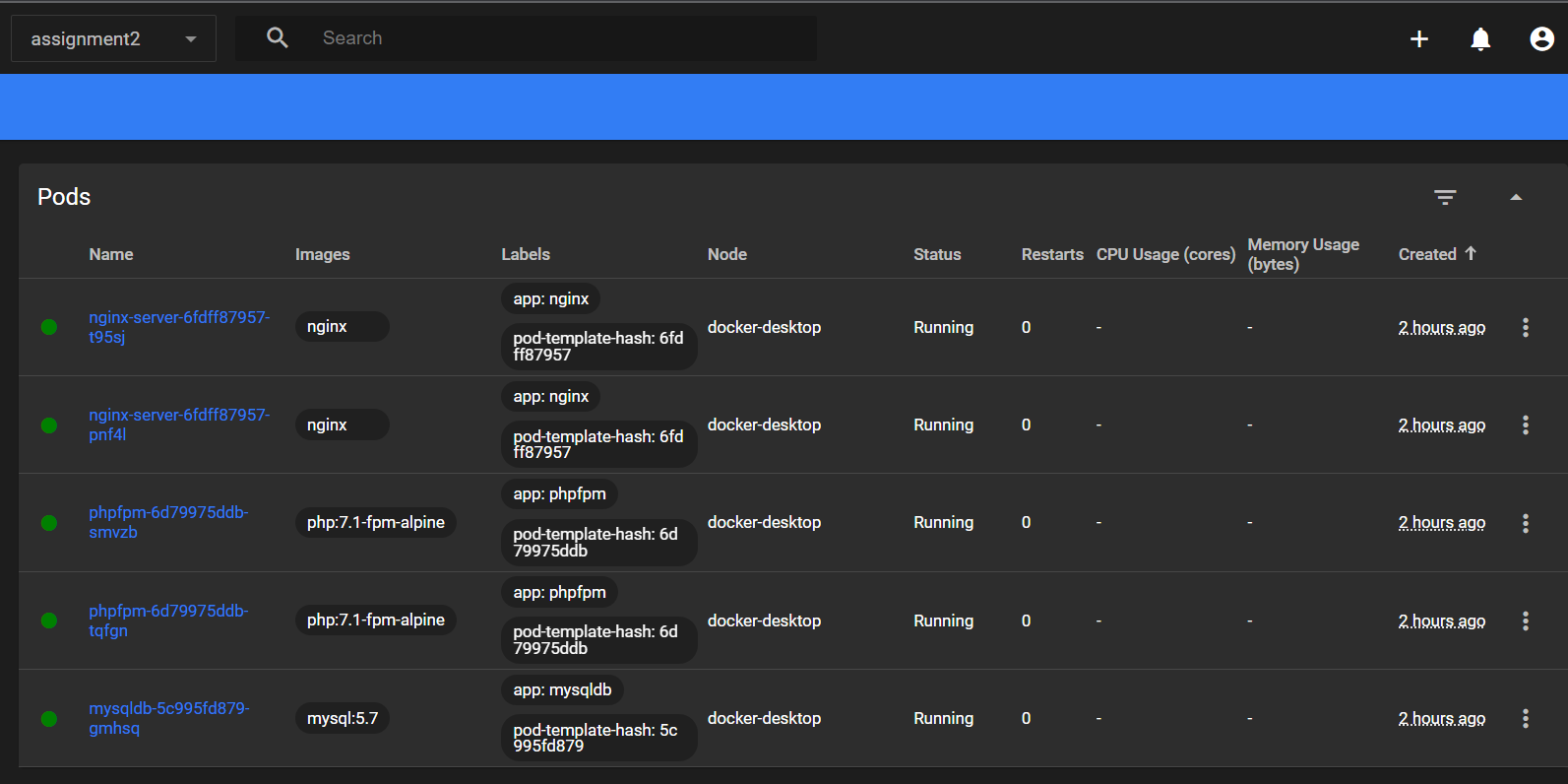
    app: mysqldb

  ports:

    - protocol: TCP

      port: 3306

      targetPort: 3306



A screenshot of a computer

Description automatically generated with medium confidence

A screenshot of a computer

Description automatically generated with medium confidence

***MYSQL SECRET***

apiVersion: v1

kind: Secret

metadata:

  name: mysql-secret

type: Opaque

data:

  mysql-root-password: Y29tcGxleHBhc3N3b3Jk

  mysql-user: T21lclNpZGRpcXVp

  mysql-password: cGFzc3dvcmQ=

Graphical user interface, application

Description automatically generated

***NGINX CONFIGMAP***

apiVersion: v1

kind: ConfigMap

metadata:

  name: nginx-configmap

data:

  mysql-host: mysql-service

  database-name: "studentsdb"

  nginx.conf: |

    server {

      listen 80;

      listen [::]:80;

      access\_log off;

      root /var/www/html;

      index index.php;

      server\_name example.com;

      server\_tokens off;

      location / {

        # First attempt to serve request as file, then

        # as directory, then fall back to displaying a 404.

        try\_files $uri $uri/ /index.php?$args;

      }

      # pass the PHP scripts to FastCGI server listening on wordpress:9000

      location ~ \.php$ {

        fastcgi\_split\_path\_info ^(.+\.php)(/.+)$;

        # Change The Service Name

        fastcgi\_pass phpfpm:9000;

        fastcgi\_index index.php;

        include fastcgi\_params;

        fastcgi\_param SCRIPT\_FILENAME $document\_root$fastcgi\_script\_name;

        fastcgi\_param SCRIPT\_NAME $fastcgi\_script\_name;

      }

    }

  initdb.sql: |

    use studentsdb;

    CREATE TABLE Students (rollnum varchar(255),firstname varchar(255),lastname varchar(255));

    INSERT INTO Students (rollnum, firstname, lastname) VALUES ("19B-004-SE", "Omer", "Siddiqui");

    INSERT INTO Students (rollnum, firstname, lastname) VALUES ("19B-011-SE", "Adnan", "Samad");

    INSERT INTO Students (rollnum, firstname, lastname) VALUES ("19B-035-SE", "Taha", "Sharof");

Graphical user interface

Description automatically generated

***INDEX.PHP***

<!DOCTYPE html>

<html>

<body>

<?php

echo "<table style='border: solid 1px black;'>";

echo "<tr><th>Roll No.</th><th>Firstname</th><th>Lastname</th></tr>";

class TableRows extends RecursiveIteratorIterator {

    function \_\_construct($it) {

        parent::\_\_construct($it, self::LEAVES\_ONLY);

    }

    function current() {

        return "<td style='width: 150px; border: 1px solid black;'>" . parent::current(). "</td>";

    }

    function beginChildren() {

        echo "<tr>";

    }

    function endChildren() {

        echo "</tr>" . "\n";

    }

}

$servername = $\_SERVER['MYSQL\_HOST'];

$username = $\_SERVER['MYSQL\_USERNAME'];

$password = $\_SERVER['MYSQL\_PASSWORD'];

$dbname = $\_SERVER['MYSQL\_DATABASE'];

try {

    $conn = new PDO("mysql:host=$servername;dbname=$dbname", $username, $password);

    $conn->setAttribute(PDO::ATTR\_ERRMODE, PDO::ERRMODE\_EXCEPTION);

    $stmt = $conn->prepare("SELECT rollnum, firstname, lastname FROM Students");

    $stmt->execute();

    // set the resulting array to associative

    $result = $stmt->setFetchMode(PDO::FETCH\_ASSOC);

    foreach(new TableRows(new RecursiveArrayIterator($stmt->fetchAll())) as $k=>$v) {

        echo $v;

    }

}

catch(PDOException $e) {

    echo "Error: " . $e->getMessage();

}

$conn = null;

echo "</table>";

?>

</body>

</html>

Shape, rectangle

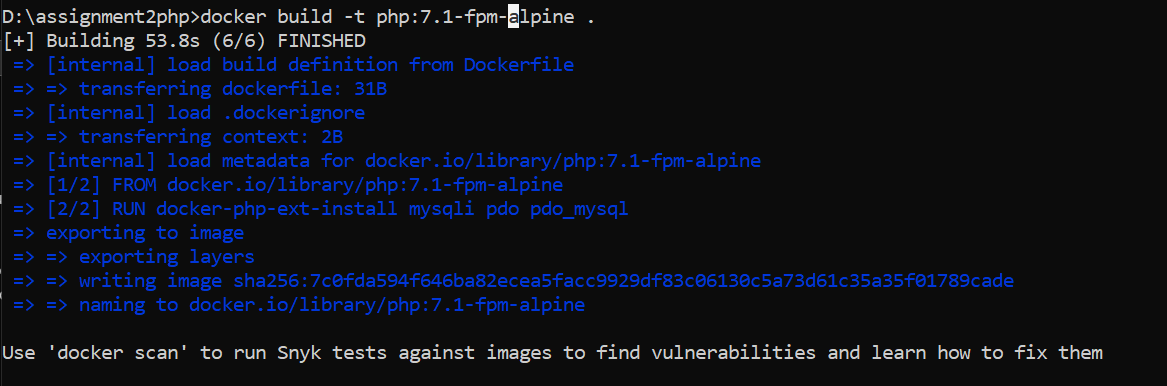
Description automatically generated

***DOCKER FILE***

FROM php:7.1-fpm-alpine

RUN docker-php-ext-install mysqli pdo pdo\_mysql

#  docker build -t php:7.1-fpm-alpine .



A screenshot of a computer

Description automatically generated with medium confidence

Graphical user interface, text

Description automatically generated 