KaozhengPro

IT認證考試題庫

專業平臺

考證寶提供最新考古題與模擬試題
協助您高效通過認證考試

www.kaozhengpro.com

Exam : **EX370**

Title : Red Hat Certified Specialist in OpenShift Data Foundation exam

Version : DEMO

1.You are tasked with creating a project named production-environment in OpenShift using the command line. Provide the step-by-step process to create the project and verify it.

Answer:

Solution:

1. Create the project using the oc CLI: oc new-project production-environment
2. Verify that the project has been created: oc get projects
3. Switch to the new project context: oc project production-environment

Explanation:

The oc new-project command creates a new namespace in OpenShift. Verifying ensures the namespace is ready, and switching context ensures subsequent commands execute in the correct project.

2.You are required to create the same project, production-environment, using the OpenShift web console.

Walk through the steps to achieve this.

Answer:

Solution:

1. Log in to the OpenShift web console.
2. Click on “Projects” from the left-hand navigation menu.
3. Click “Create Project” and fill in the name as production-environment. Optionally, provide a description and display name.
4. Click “Create” and verify the project is listed in the “Projects” view.

Explanation:

The OpenShift web console provides a graphical interface for managing resources. Using the web console allows non-technical users to perform administrative tasks easily.

3.Create a ConfigMap named application-config with the key-value pairs env=production and debug=false.

Walk through the YAML definition, applying it to the cluster, and verifying the result.

Answer:

Solution:

1. Create a ConfigMap YAML file (application-config.yaml):

```
apiVersion: v1
kind: ConfigMap
metadata:
  name: application-config
data:
  env: production
  debug: "false"
```

2. Apply the YAML file:

```
oc apply -f application-config.yaml
```

3. Verify the ConfigMap creation:

```
oc describe configmap application-config
```

Explanation:

ConfigMaps store non-sensitive configuration data for applications. By defining and applying a YAML file, the configuration is maintained in a structured and reusable format.

4.Edit an existing ConfigMap named application-config to change the value of debug to true. Provide step-by-step instructions.

Answer:

Solution:

1. Edit the ConfigMap using the oc CLI: oc edit configmap application-config
2. Change the value of debug to true in the opened YAML editor:

```
data:
   env: production
   debug: "true"
```

3. Save and exit the editor.
4. Verify the changes:

```
oc get configmap application-config -o yaml
```

Explanation:

Editing a ConfigMap using the oc edit command allows you to modify its values in real time. This is useful for making quick updates without reapplying YAML.

5.Create a Secret named database-credentials with the keys username=admin and password=secret12 3. Walk through the process using both the CLI and YAML.

Answer:

Solution:

1. Create the Secret using the CLI:

```
oc create secret generic database-credentials \
   --from-literal=username=admin \
   --from-literal=password=secret123
```

2. Alternatively, create a YAML file (database-credentials.yaml):

```
apiVersion: v1
kind: Secret
metadata:
   name: database-credentials
type: Opaque
data:
   username: YWRtaW4= # Base64 encoded value of 'admin'
   password: c2VjcmV0MTIz # Base64 encoded value of 'secret123'
```

3. Apply the YAML:

```
oc apply -f database-credentials.yaml
```

4. Verify the Secret:

```
oc get secret database-credentials -o yaml
```

Explanation:

Secrets securely store sensitive data in base64-encoded format. The CLI method is quick, while the YAML approach allows for better version control and auditing.