ExamEnroll
EXAM SUCCESS SIMPLIFIED

RedHat

EX380

Red Hat Certified Specialist in OpenShift Automation and Integration

Exam Version: 6.2

QUESTIONS & ANSWERS

DEMO VERSION

(LIMITED CONTENT)

Thank you for Downloading EX380 exam PDF Demo

You can also try our EX380 practice exam software

Visit link below:

https://examsenroll.com/exam/ex380

Question 1. (Single Select)

SIMULATION

Task SIMULATION 1

Node Management – Remove Taint on Worker Node

A: See thesolution below in Explanation

Answer: A

Explanation:

Step 1: Log in to the OpenShift web console with an account that has sufficient cluster administrative privileges.

This Task is performed from the GUI, not the CLI. The lab hint explicitly places this under the worker node details page in the console.

Step 2: Navigate to Compute.

This area contains node-level resources, including control plane and worker nodes.

Step 3: Open Nodes.

Here you can view all nodes currently registered in the cluster.

Step 4: Select the required worker node.

Choose the exact worker node referenced by the lab Task SIMULATION.

Step 5: Open the Details tab.

The taint configuration is managed from the selected node's details view.

Step 6: Locate the Taints section and click Edit.

A taint is used to control pod scheduling. If a worker has a taint, pods without matching tolerations may not schedule there.

Step 7: Remove the unwanted taint entry.

Removing the taint makes the worker eligible again for normal scheduling behavior, depending on the rest of the cluster policy.

Step 8: Click Save.

This commits the change so the node is updated and the scheduler can evaluate it without that taint.

==========

Question 2. (Single Select)

SIMULATION

Task SIMULATION 7

Service Accounts and RBAC – Grant Cluster Reader Role

A: See thesolution below in Explanation

Answer: A

Explanation:

Step 1: Confirm the service account exists in auth-audit.

It must exist before a role can be assigned to it.

Step 2: Run the command:

oc adm policy add-cluster-role-to-user cluster-reader system:serviceaccount:auth-audit:audit

Step 3: Verify the binding is added.

The lab output shows:

clusterrole.rbac.authorization.k8s.io/cluster-reader added: "system:serviceaccount:auth-audit:audit"

Detailed explanation:

This binds the cluster-reader cluster role to the audit service account. The full subject format system:serviceaccount:namespace:name is required because OpenShift RBAC needs the exact service account identity. The cluster-reader role is broader than a project-scoped view role because it allows read-level access across cluster resources. This is appropriate for auditing or inspection use cases where the account must observe but not modify. The distinction between cluster roles and namespaced roles is important: cluster roles apply to non-namespaced resources and broad cluster visibility, while local roles are limited to individual projects. This Task is a classic RBAC operation that combines identity creation with controlled privilege assignment.

==========

Question 3. (Single Select)

SIMULATION

Task SIMULATION 8

Kubeconfig Management – Approve CSR

A: See thesolution below in Explanation

Answer: A

Explanation:

Step 1: Identify the pending certificate signing request.

The lab names it audit-csr.

Step 2: Run the approval command:

oc adm certificate approve audit-csr

Step 3: Confirm approval.

The lab output shows:

certificatesigningrequest.certificates.k8s.io/audit-csr approved

Detailed explanation:

In Kubernetes and OpenShift, a CSR must be approved before the requester can use the signed certificate for authentication. This Task approves the CSR named audit-csr, which is likely associated with the audit user or service account access flow in the lab. Certificate-based authentication is commonly used for kubeconfig access because it enables secure client identity without relying solely on tokens. Until the CSR is approved, the certificate cannot be trusted by the cluster API for authenticated operations. Administrative approval is therefore a gatekeeping step that ensures only intended certificate requests become valid credentials. This Task is part of a broader kubeconfig workflow that continues with setting credentials and defining context.

==========

Question 4. (Single Select)

SIMULATION

Task SIMULATION 9

Kubeconfig Management – Set Credentials in Kubeconfig

A: See thesolution below in Explanation

Answer: A

Explanation:

Step 1: Ensure the client certificate and private key files are available.

The lab uses audit.crt and tls.key.

Step 2: Run the command:

oc config set-credentials audit --client-certificate audit.crt --client-key tls.key --embed-certs --kubeconfig audit.config

Step 3: Confirm the user entry is written.

The lab output shows:

User "audit" set.

Detailed explanation:

This command creates or updates the audit user entry inside the kubeconfig file audit.config. It points the user to a client certificate and private key, and the --embed-certs option stores certificate material directly inside the kubeconfig rather than only referencing external files. That makes the kubeconfig more portable because it can be moved and used without separately copying the certificate files, provided the embedded content is valid. In certificate-based authentication, the private key proves client possession while the certificate presents the approved identity. If the certificate and key do not match, authentication will fail. This step does not yet define what cluster or namespace the user works against; it only defines the credential identity.

==========

Question 5. (Single Select)

SIMULATION

Task SIMULATION 10

Kubeconfig Management – Set Context in Kubeconfig

A: See thesolution below in Explanation

Answer: A

Explanation:

Step 1: Verify the cluster name, namespace, and user name that should be referenced.

The lab uses cluster api-ocp4-example-com:6443, namespace audit-ns, and user audit-sa.

Step 2: Run the command:

oc config set-context audit --cluster api-ocp4-example-com:6443 --namespace audit-ns --user audit-sa --kubeconfig audit.config

Step 3: Confirm context creation.

The lab output shows:

Context "audit" created.

Detailed explanation:

A kubeconfig context ties together three things: a cluster endpoint, a user identity, and optionally a default namespace. This Task creates a context named audit in the file audit.config. Contexts are useful because they simplify repeated administration by letting the user switch between prepared working environments instead of re-entering cluster and namespace details each time. The namespace portion is especially helpful for project-scoped operations, because commands run under that context default to the chosen namespace. Accuracy matters here: if the user name in the context does not match the credentials entry or the cluster name does not exist in the kubeconfig, the context will not function as intended.

==========

For More Information - Visit link below:

https://examsenroll.com/exam/ex380

Thanks for using our Product!

Pass Your Certification With ExamsEnroll Guarantee

For $20 discount, use coupon code:

MINUS20

Sales: sales@examsenroll.com

Support: support@examsenroll.com

