PassQueen
a.k.a dumpsqueen.com

Red Hat Certified Specialist in OpenShift Advanced Cluster Management

RedHat EX432

Version Demo

Total Demo Questions: 4

Total Premium Questions: 45

Buy Premium PDF

https://passqueen.com

support@passqueen.com

Topic Break Down

Topic	No. of Questions
Topic 1, Manage the multicluster hub	4
Topic 2, Manage clusters	19
Topic 3, Manage applications	7
Topic 4, Manage policies	15
Total	45

QUESTION NO: 1 - (SIMULATION)

Bind Cluster to ClusterSet

Task information: Add cluster-dev to the development ClusterSet.

ANSWER: See the explanation for the answer

Explanation:

Bind the managed cluster `cluster-dev` to the `development` ManagedClusterSet from the hub cluster:

```
clusteradm clusterset bind --clusterset development --cluster cluster-dev
```

Verify that the cluster has been assigned to the ClusterSet:

```
oc get managedcluster cluster-dev -o jsonpath='{.metadata.labels.cluster\.open-cluster-management\.io/clusterset}{"\n"}'
```

The command should return:

```
development
```

The bind operation sets the `cluster.open-cluster-management.io/clusterset=development` label on `cluster-dev`.

QUESTION NO: 2 - (SIMULATION)

Rotate/renew managed cluster access (agent cert / CSR workflow) and restore connectivity

ANSWER: See the explanation for the answer

Explanation:

On the hub cluster, identify the affected managed cluster and approve only the valid pending CSR(s) generated by that cluster's klusterlet/agent. A managed cluster commonly becomes unavailable when its agent client certificate has expired or is awaiting approval.

```
# Check the managed-cluster state and certificate-related conditions
oc get managedcluster
oc describe managedcluster <managed-cluster-name>

# List CSRs, including requester identity and approval status
oc get csr
oc get csr -o custom-
columns=NAME:.metadata.name,SIGNER:.spec.signerName,USERNAME:.spec.username,STATUS:.status.
conditions[*].type
```

Inspect every candidate before approving it. The CSR must belong to the expected managed cluster and have an OpenShift/Kubernetes client signer, normally `kubernetes.io/kube-apiserver-client`:

```
oc describe csr <csr-name>
# or
oc get csr <csr-name> -o yaml
```

For ACM klusterlet registration and renewal, the requester normally identifies Open Cluster Management, for example a username/group containing `open-cluster-management`, and should correspond to the affected cluster. Do **not** approve unrelated or unexpected CSRs.

Approve the validated pending CSR(s):

```
oc adm certificate approve <csr-name>
```

```
# Confirm that it was approved and issued
oc get csr <csr-name>
```

If several valid CSRs belong to the same affected agent (for example, a bootstrap CSR followed by its renewed client CSR), approve each valid pending CSR:

```
oc get csr --no-headers | awk '$NF == "Pending" {print $1}'
# Inspect each name returned before running:
oc adm certificate approve <validated-csr-name>
```

Then verify recovery from the hub:

```
oc get managedcluster <managed-cluster-name>
oc describe managedcluster <managed-cluster-name>
```

The expected result is that the managed cluster returns to `Available=True` (and is shown as ready/available). If no renewal CSR appears after waiting briefly, check the klusterlet/agent on the managed cluster and restart the relevant agent workload only according to the cluster's installed ACM configuration; do not blindly delete certificate secrets or approve unknown CSRs.

QUESTION NO: 3 - (SIMULATION)

Create a PolicySet and include multiple policies for a baseline

ANSWER: See the explanation for the answer

Explanation:

Create the `baseline-dev` PolicySet in the `team-dev` namespace. A PolicySet contains references to Policy resources in the same namespace.

```
cat <<'EOF' | oc apply -f -
apiVersion: policy.open-cluster-management.io/v1beta1
kind: PolicySet
metadata:
  name: baseline-dev
  namespace: team-dev
spec:
  policies:
    - policy-ensure-audit-namespace
    # Add each additional existing baseline Policy name here, for example:
    # - <another-existing-policy-name>
EOF
```

To include multiple policies, add every required existing Policy name as another item under `spec.policies`; do not use policy YAML filenames or placement-binding names.

```
oc get policyset baseline-dev -n team-dev -o yaml
oc describe policyset baseline-dev -n team-dev
```

The resulting PolicySet is named `baseline-dev` and references `policy-ensure-audit-namespace` plus any other baseline policies created for `team-dev`.

QUESTION NO: 4 - (SIMULATION)

Create MultiClusterHub (CLI Alternative)

Task information: Apply the MultiClusterHub custom resource if not using Web Console.

ANSWER: See the explanation for the answer

Explanation:

Log in to the OpenShift cluster that will be the ACM hub and create the `MultiClusterHub` custom resource in the namespace where the Advanced Cluster Management operator is installed (normally `open-cluster-management`).

```
oc login <hub-cluster-api>
oc project open-cluster-management

cat > multiclusterhub.yaml <<'EOF'
apiVersion: operator.open-cluster-management.io/v1
kind: MultiClusterHub
metadata:
  name: multiclusterhub
  namespace: open-cluster-management
spec: {}
EOF

oc apply -f multiclusterhub.yaml
```

Verify that the operator accepted and reconciles the hub CR:

```
oc get multiclusterhub -n open-cluster-management
oc describe multiclusterhub multiclusterhub -n open-cluster-management
oc get pods -n open-cluster-management -w
```

The required installation object is therefore `MultiClusterHub/multiclusterhub`. The ACM operator watches this resource and deploys the hub components. If the operator was installed into a different namespace, use that namespace in `metadata.namespace` and in the verification commands.