



# Exam Experience



# About the Exam

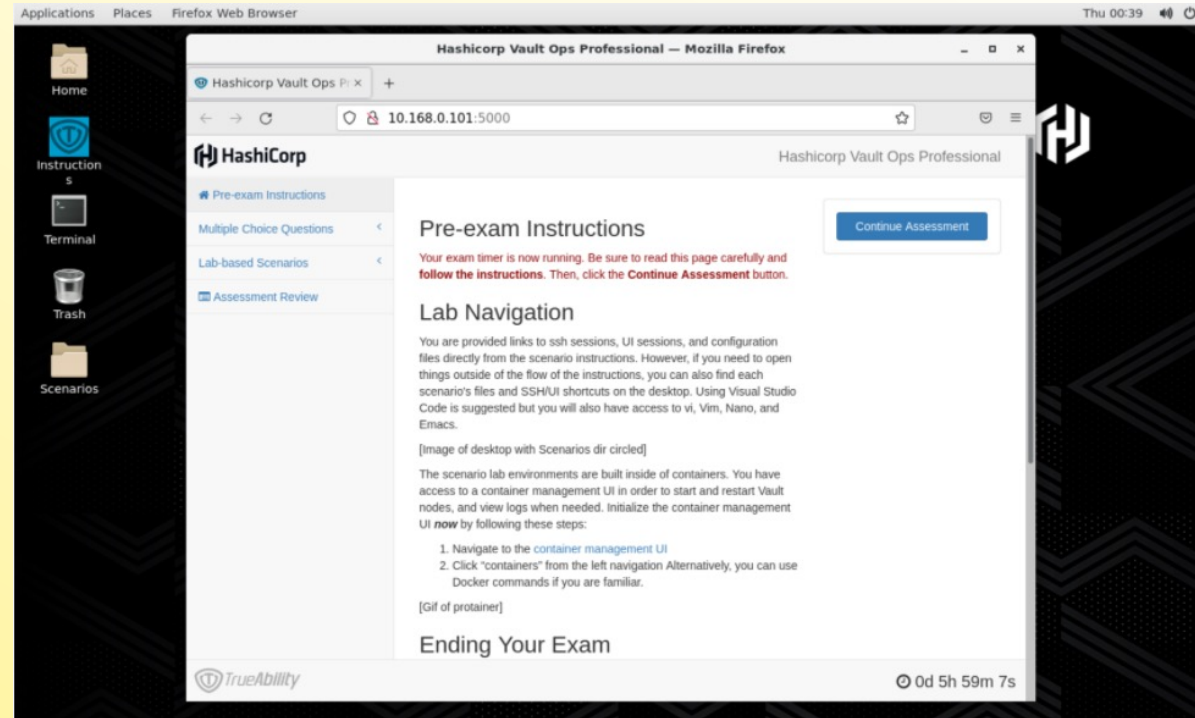


- **Multiple-Choice Questions**
  - Longer, scenario-based questions
- **Hands-On Lab Scenarios**
  - Read a scenario, configure Vault as required
- **Hybrid Scenarios**
  - Read a scenario, look at Vault to determine the answer and select the proper answer from multiple-choice



# Exam Experience

- All assessments are done via browser in the virtual desktop
- Hybrid Questions are under the Lab-Based Scenarios
- You can use Firefox to access Vault documentation



# Multiple-Choice



Your organization is running Vault Enterprise for production workloads. Due to high workloads, your production cluster is cannot keep up with the number of reads requests. As a result, applications are displaying errors to customers and the business is at risk of losing significant revenue.

How can you reconfigure the Vault cluster to provide scale-out read capabilities without taking an outage on any of the Vault cluster nodes?

- a) Increase the memory for each node by changing the underlying hardware
- b) Determine the secrets engine being overused and enable multiple secrets engines to spread the workload across them
- c) Reconfigure the load balancer with another listener, point it to the performance standby nodes, and reconfigure read-only applications to send request to the new listener
- d) Deploy another Vault cluster and configure disaster recovery replication. Redirect some clients to the new cluster.

Answer: C



# Hybrid Question



You are the Vault engineer at your organization. A user recently stated they cannot log in to Vault and interact with the proper Namespace assigned to their team. Based on the policies that are already created in Vault, determine which policy permits them to authenticate to the root namespace and interact with the namespace called mobile-team-a.

Click [HERE](#) to open an SSH session to the Vault cluster node. Determine the correct policy to use and select the answer below.

- a) operation-policy
- b) automation-ro-policy
- c) training-team-policy
- d) developer-team-a-policy

```
TERMINAL

Welcome to the Vault cluster.

$
```

Answer: C



# Lab-Based Scenarios



- There are **MANY** grading opportunities throughout the test, meaning that you will get partial credit for doing some things correctly even if others are incorrect
- **Validate your configurations/answers**
  - Run a `cat /grading/scenario1/node-a.json` to view the file
  - Log in as a user you created to validate it works and the correct config is there
- Some tasks build on others, so you'll need to complete each scenario in order:
  1. Create a Vault configuration file
  2. Start Vault
  3. Initialize Vault





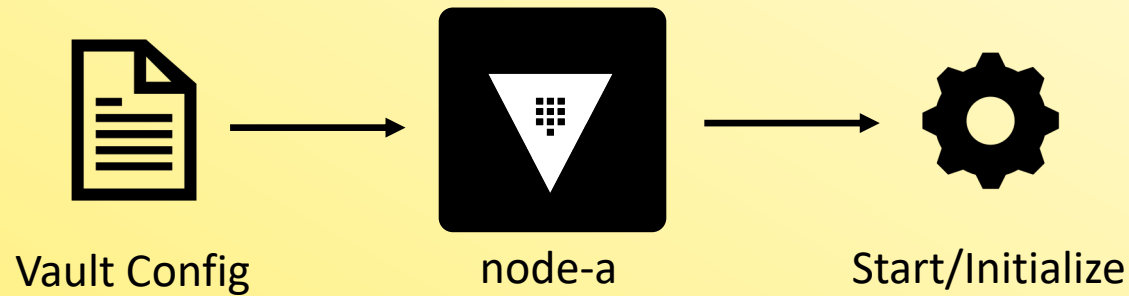
# Lab-Based Scenarios

Example Scenario



## Initialize a Vault node using Integrated Storage

- Demonstrate you can create a Vault server configuration file
- Know how to configure Auto Unseal
- Set up an HA cluster
- Initialize a Vault server using production hardening techniques



\*Information from Study Guide



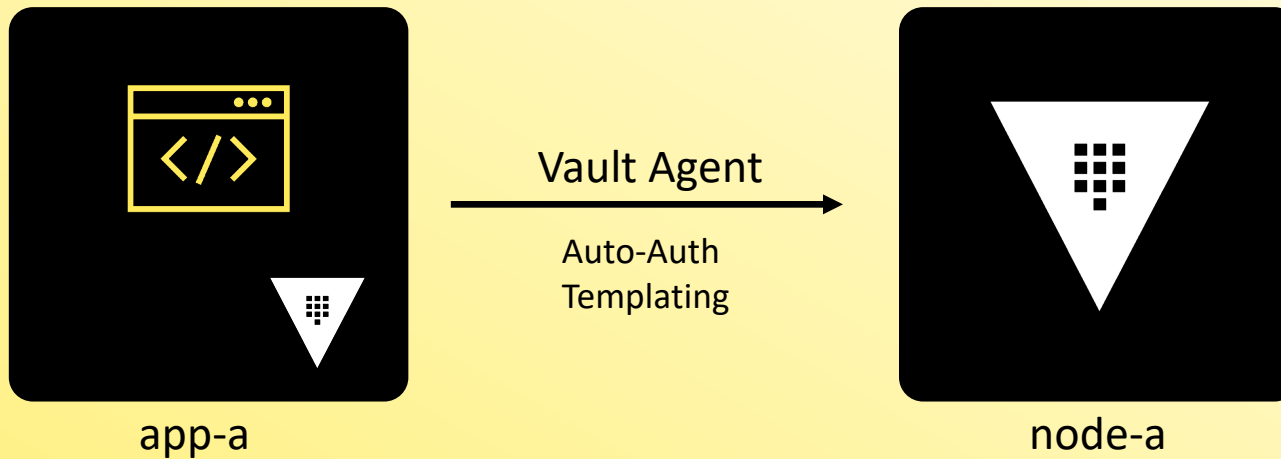
# Lab-Based Scenarios

Example Scenario



## Vault Agent and Templating

- Set up a Vault Agent to authenticate to Vault (*Vault Agent Auth-Auth*)
- Retrieve Secrets (*Vault Agent Templates*)



\*Information from Study Guide





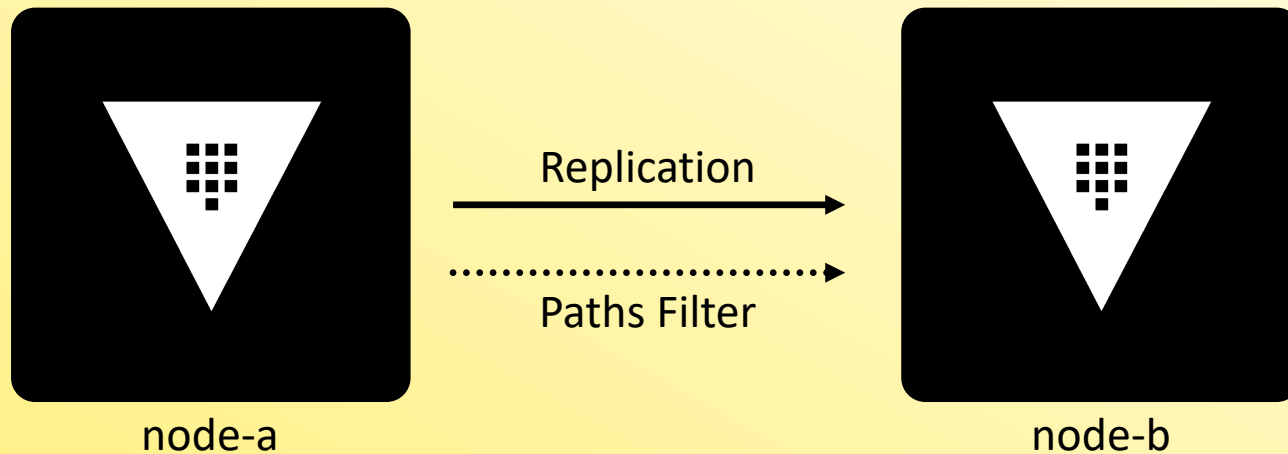
# Lab-Based Scenarios

Example Scenario



## Vault Enterprise Replication

- Enable and Configure DR Replication and Performance Replication
- Know how to configure a paths filter



\*Information from Study Guide



# Using Portainer

Needed for some Lab-Based Scenarios



The screenshot shows the Portainer web interface. On the left is a dark blue sidebar with navigation options: Home, LOCAL, Dashboard, App Templates, Stacks, Containers (highlighted with a mouse cursor), Images, Networks, Volumes, Events, Host, SETTINGS, Users, Environments, Registries, and Settings. The main content area is titled "Container list" and shows a table of containers. Above the table are action buttons: Start, Stop, Kill, Restart, Pause, Resume, Remove, and Add container. A yellow callout box labeled "2. Choose Action" has arrows pointing to the Start, Stop, Kill, and Restart buttons. Another yellow callout box labeled "1. Select Container" has an arrow pointing to the checkbox of the "mysql" container in the table. The table has columns for Name, State, Quick actions, Stack, Image, Created, IP Address, Published Ports, and Ownership. The "mysql" container is highlighted with a red box.

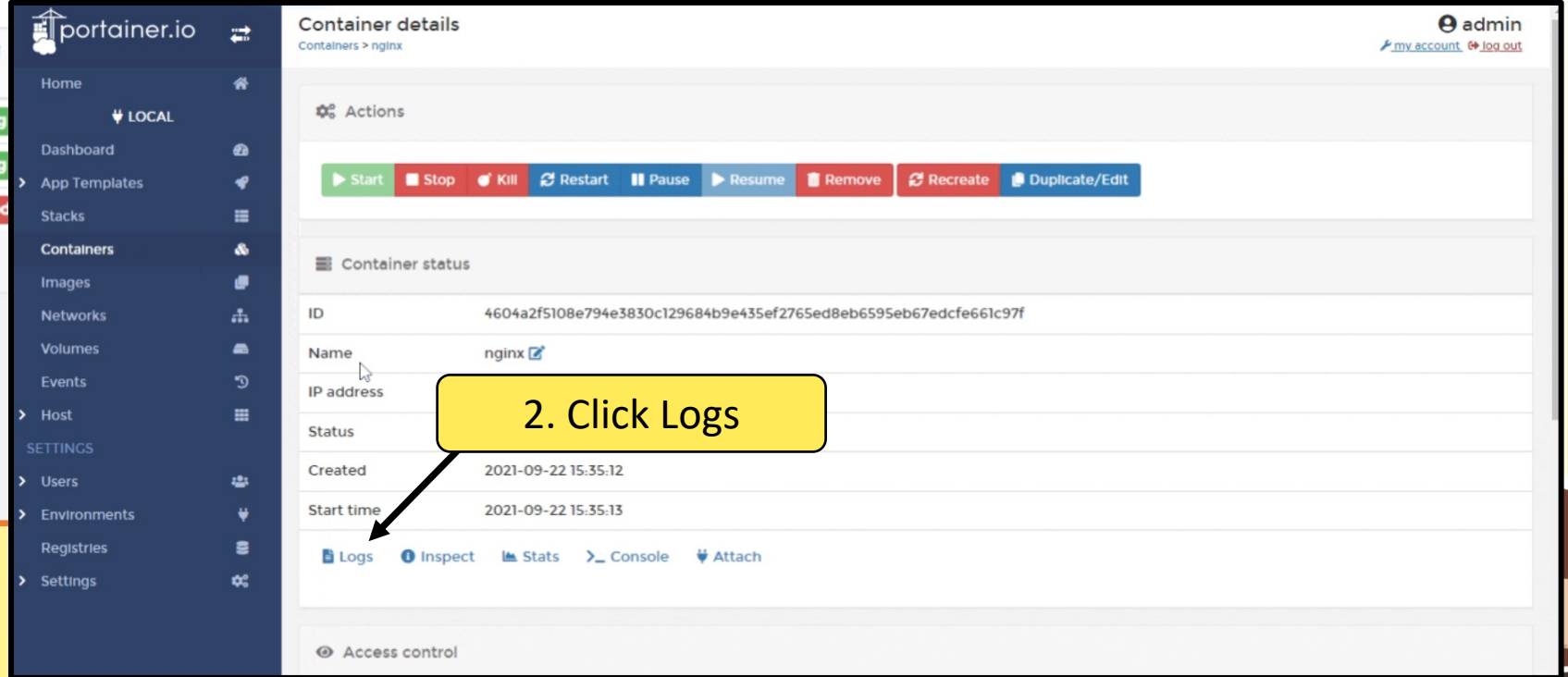
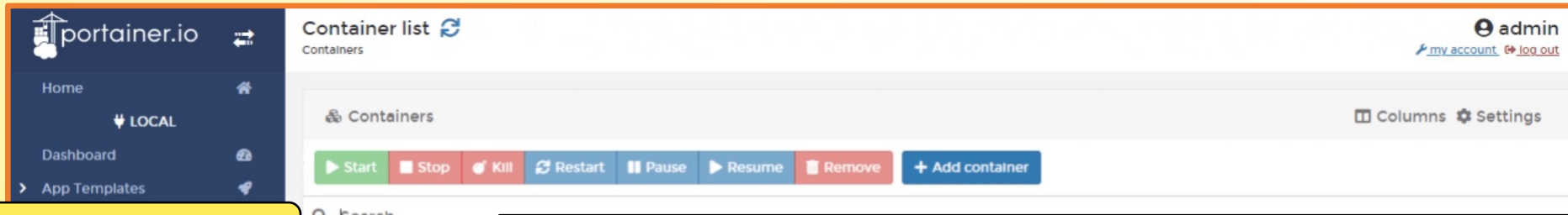
<input type="checkbox"/>	Name	State	Quick actions	Stack	Image	Created	IP Address	Published Ports	Ownership
<input type="checkbox"/>	portainer	running	[Icons]	-	portainerci/portainer.develop	2021-09-22 15:50:09	172.17.0.2	8000:8000 9443:9443	administrators
<input type="checkbox"/>	nginx	running	[Icons]	-	nginx:latest	2021-09-22 15:35:12	172.17.0.3	80:80	administrators
<input type="checkbox"/>	mysql	stopped	[Icons]	-	mysql:8.0	2021-09-22 15:35:33	-	-	administrators

\*Screenshot from docs.portainer.io

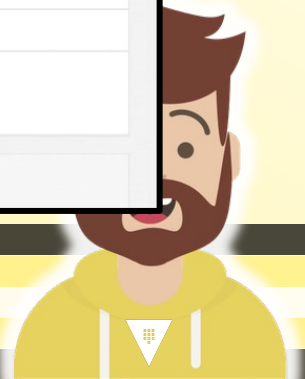


# Using Portainer

Needed for some Lab-Based Scenarios



\*Screenshot from docs.portainer.io



# Using Portainer

Needed for some Lab-Based Scenarios



The screenshot shows the Portainer web interface. On the left is a dark blue sidebar with navigation options: Home, LOCAL, Dashboard, App Templates, Stacks, Containers, Images, Networks, Volumes, Events, Host, SETTINGS, Users, Environments, Registries, and Settings. The main content area is titled "Container logs" and shows "Containers > nginx > Logs". It includes a "Log viewer settings" section with toggle switches for "Auto-refresh logs" (checked), "Wrap lines" (checked), and "Display timestamps" (unchecked). There is a "Fetch" dropdown set to "All logs", a "Search" input field with "Filter...", and a "Lines" input field set to "100". Below these are "Actions" buttons: "Download logs", "Copy", "Copy selected lines", and "Unselect". A yellow callout box with the text "View Logs Here" and an arrow points to the log viewer area. The log viewer itself shows a red-bordered box containing the following log output:

```
/docker-entrypoint.sh: /docker-entrypoint.d/ is not empty, will attempt to perform configuration
/docker-entrypoint.sh: Looking for shell scripts in /docker-entrypoint.d/
/docker-entrypoint.sh: Launching /docker-entrypoint.d/10-listen-on-ipv6-by-default.sh
10-listen-on-ipv6-by-default.sh: info: Getting the checksum of /etc/nginx/conf.d/default.conf
```

\*Screenshot from docs.portainer.io





**END OF  
SECTION**

