



Download Pure Storage Portworx-Enterprise-Professional Exam Dumps Free

Shared by Rosa on 17-06-2026

For More Free Questions and Preparation Resources

Check the Links on Last Page



Question 1

Question Type: MultipleChoice

Which command should an administrator run to initiate the license expansion for a blue-green upgrade in a Portworx cluster?

Options:

- A- pxctl license expand --start
- B- pxctl cluster expand -start
- C- pxctl node expand --start



Answer:

A

Explanation:

Comprehensive and Detailed Explanation From Exact Extract:

During blue-green upgrades or license expansions in Portworx clusters, administrators use the command `pxctl license expand --start` to initiate the license expansion process. This command signals Portworx to begin applying the new license, enabling additional nodes or features as permitted. The process is carefully managed to avoid disruption during the upgrade and ensure that new license entitlements are recognized. The Portworx upgrade and licensing documentation specify this command as the official method for license expansion, ensuring compliance and seamless cluster scaling during complex upgrade workflows. [Pure Storage Portworx License Management Guide](#)source.



Question 2

Question Type: MultipleChoice

Which Portworx component is used to co-locate volumes with pods?

Options:

- A- Autopilot

- B- Stork
- C- Volume Placement Strategy

Answer:

C

Explanation:

Comprehensive and Detailed Explanation From Exact Extract:

Portworx's Volume Placement Strategy ensures that persistent volumes are co-located with the pods that use them, enhancing performance and reducing latency. This strategy involves applying placement rules and constraints that guide Kubernetes scheduler and Portworx storage operations to place data volumes on nodes close to or the same as the pods. Co-location improves I/O throughput and application responsiveness by minimizing network hops between compute and storage resources. While Autopilot automates scaling and Stork manages storage-aware scheduling, Volume Placement Strategy specifically handles volume location relative to workloads. The Portworx architecture documentation highlights this component as critical for optimizing storage efficiency and workload performance in Kubernetes environments running Portworx storagePure Storage Portworx Architecture Docssource.

Question 3

Question Type: MultipleChoice

What step is necessary to start using encrypted PVCs in Portworx?

Options:

- A- Select secret provider.
- B- StorageClass needs the following parameter: secure: enabled.
- C- Configure IO profiles.

Answer:

A

Explanation:

Comprehensive and Detailed Explanation From Exact Extract:

Using encrypted Persistent Volume Claims (PVCs) with Portworx requires that an administrator first configure a secret provider responsible for managing the encryption keys. The secret provider could be an external Key Management System (KMS) such as AWS KMS, Google Cloud KMS, Hashicorp Vault, or Kubernetes Secrets. This step is critical because encryption keys are essential to securely encrypt and decrypt data on volumes. Although enabling encryption in the StorageClass via parameters like `secure: enabled` is necessary to activate encryption on volumes, it is insufficient without a properly configured secret provider to manage the keys. The secret provider ensures keys are securely stored, rotated, and accessed, fulfilling compliance and security requirements. Portworx documentation stresses this as a foundational step to enable encrypted PVCs, highlighting that without a configured secret provider, encrypted volumes cannot be provisioned or used effectively. [Pure Storage Portworx Encryption Docs](#) source.

Question 4

Question Type: MultipleChoice

Which flag in the Portworx StorageCluster spec enables telemetry?

Options:

- A- `spec.autopilot.enabled`
- B- `spec.telemetry.enabled`
- C- `spec.csi.enabled`

Answer:

B

Explanation:

Comprehensive and Detailed Explanation From Exact Extract:

Telemetry in Portworx refers to the automated collection and reporting of cluster performance and health metrics to Pure1 or other monitoring services. To enable telemetry, the `spec.telemetry.enabled` flag must be set to true in the StorageCluster Custom Resource Definition (CRD). This setting activates the telemetry pod on each node, which collects data such as resource usage, storage capacity, and errors, then securely uploads it to Pure Storage's management platform. Enabling telemetry helps administrators gain insights into cluster performance trends, preemptively identify issues, and optimize resource utilization. The Portworx

operator respects this flag during installation and upgrades to ensure telemetry is consistently configured. Neither `spec.autopilot.enabled` (which controls the Autopilot feature) nor `spec.csi.enabled` (which controls CSI driver deployment) affects telemetry settings. Official Portworx documentation highlights this flag as critical for activating health monitoring and analytics features within Portworx clustersPure Storage Portworx Telemetry Guidesource.

Question 5

Question Type: MultipleChoice

An application team is preparing to deploy an ElasticSearch application and wants all Portworx volumes created in 6 specific Kubernetes nodes.

Which Portworx feature should they use to achieve this?

Options:

- A- Stork
- B- Autopilot
- C- Volume placement strategy

Answer:

C

Explanation:

Comprehensive and Detailed Explanation From Exact Extract:

To ensure Portworx volumes for an ElasticSearch application are created only on specific Kubernetes nodes, the Volume Placement Strategy feature is used. This feature allows administrators to define node affinity or anti-affinity rules that restrict volume provisioning to a subset of nodes. By tagging the six nodes with appropriate labels and configuring the StorageClass or volume parameters to respect these labels, Portworx guarantees that volumes will only be provisioned on those nodes. This targeted volume placement is critical for performance optimization, data locality, and compliance with infrastructure constraints. Autopilot automates scaling and Stork manages storage-aware scheduling but does not directly control volume node placement. The Portworx deployment documentation highlights Volume Placement Strategy as the tool for precise volume-to-node mapping in Kubernetes clustersPure Storage Portworx Deployment Guidesource.

Question 6

Question Type: MultipleChoice

What is the primary purpose of Stork in a Kubernetes cluster?

Options:

- A- To manage storage operations and migrations.
- B- To deploy applications automatically.
- C- To monitor network traffic.

Answer:

A

Explanation:

Comprehensive and Detailed Explanation From Exact Extract:

Stork (Storage Orchestrator for Kubernetes) is a Portworx component designed to enhance Kubernetes storage management. Its primary purpose is to orchestrate storage-aware operations, including volume scheduling, migration, backup, and disaster recovery. Stork integrates deeply with Kubernetes to provide application-aware scheduling decisions that respect storage constraints such as volume locality and affinity. It also facilitates migration of stateful workloads by coordinating volume replication and failover. Stork simplifies complex storage workflows in Kubernetes environments, enabling seamless backup and restore of applications and improving overall resilience. Portworx's official documentation highlights Stork as a key enabler for business continuity by managing storage operations and migrations, making it essential for Kubernetes environments running critical stateful workloads with Portworx storagePure Storage Portworx Stork Guidesource.

Question 7

Question Type: MultipleChoice

What command should be used to add a new drive to an existing storage cluster?

Options:

- A- pxctl service pool update ---resize new-disk 150
- B- pxctl service drive add -drive /dev/dm-1 -operation start
- C- pxctl sv nw -a

Answer:

B

Explanation:

Comprehensive and Detailed Explanation From Exact Extract:

Adding a new drive to an existing Portworx storage cluster involves bringing the physical device online for Portworx management. The correct command for this is `pxctl service drive add -drive /dev/dm-1 -operation start`. This command instructs Portworx to recognize and incorporate the new drive specified by the device path (e.g., `/dev/dm-1`) into its storage pool. After this operation, Portworx can use the drive for provisioning volumes or expanding capacity. The `-operation start` flag signals Portworx to initialize and prepare the drive for use. This method is part of Portworx's dynamic storage management capabilities, allowing flexible scaling of storage resources without downtime. Official CLI documentation outlines this command as the supported approach to adding drives to running clusters safely and efficiently. [Pure Storage Portworx CLI Guide](#) source.

Question 8

Question Type: MultipleChoice

If a Portworx node is down and the Kubernetes cluster is healthy, which command should be used to check Portworx alerts on a healthy node?

Options:

- A- pxctl alerts show
- B- journalctl -lfu portworx*
- C- kubectl describe node

Answer:

A

Explanation:

Comprehensive and Detailed Explanation From Exact Extract:

To check Portworx-specific alerts on a healthy node, administrators use the command `pxctl alerts show`. This command displays current alerts raised by Portworx, including warnings and critical notifications about storage, nodes, and cluster health. Even if one node is down, alerts from the healthy nodes can provide insights into cluster-wide issues or the status of the affected node. While `journalctl` displays system and service logs and `kubectl describe node` shows Kubernetes node info, neither provides aggregated Portworx alert data. The Portworx observability documentation recommends using `pxctl alerts show` for focused monitoring and alert management, enabling administrators to respond effectively to operational events within the Portworx cluster. [Pure Storage Portworx Monitoring Guide](#) source.

Question 9

Question Type: MultipleChoice

When upgrading Portworx on Kubernetes using the Operator, what step must the administrator take if using the `px-versions` configmap?

Options:

- A- Create a new namespace for the configmap.
- B- Update the version manifest.
- C- Delete the existing configmap.

Answer:

B

Explanation:

Comprehensive and Detailed Explanation From Exact Extract:

During Portworx upgrades via the Kubernetes Operator, if the deployment uses a `px-versions` ConfigMap to manage available Portworx versions, the administrator must update the version manifest within this ConfigMap to include the new version. This update informs the Operator of the target version for upgrades and ensures that the correct container images are pulled and deployed. Simply creating a new namespace or deleting the ConfigMap is insufficient and can cause upgrade failures or inconsistent version deployments. The Portworx Operator upgrade

documentation emphasizes updating the px-versions ConfigMap manifest as a necessary step in orchestrated upgrades, enabling controlled, predictable version management within Kubernetes clustersPure Storage Portworx Upgrade Guidesource.

Question 10

Question Type: MultipleChoice

What configuration steps should a Portworx Administrator perform to ensure that Portworx can use the S3 Object Store using a custom/3rd party (not signed by public CA) certificate?

Options:

- A- Create a Kubernetes secret containing the certificate and reference it in the storagecluster via env variable.
- B- No additional configuration is necessary.
- C- Create a secret containing the certificate and run pxctl certificate import command.

Answer:

A

Explanation:

Comprehensive and Detailed Explanation From Exact Extract:

When integrating Portworx with an S3 Object Store secured by a custom or third-party certificate that is not signed by a public Certificate Authority (CA), administrators must manually provide the relevant CA certificate to Portworx. This involves creating a Kubernetes secret that contains the custom CA certificate and referencing this secret in the StorageCluster manifest through environment variables. This allows Portworx components to trust the certificate during TLS handshake with the S3 endpoint, avoiding connection failures due to untrusted certificates. Without this step, Portworx cannot securely communicate with the object store. The Portworx security and installation documentation highlights this practice as essential for secure Object Store integration in private or regulated environments where internal or custom PKIs are usedPure Storage Portworx Security Guidesource.

Question 11

Question Type: MultipleChoice

What happens if the spec.csi.enabled flag is set to false in the Portworx StorageCluster spec?

Options:

- A- CSI will be installed, but it will not be used.
- B- The cluster will fail to deploy if CSI is disabled.
- C- CSI will not be installed for the storage cluster.

Answer:

C

Explanation:

Comprehensive and Detailed Explanation From Exact Extract:

The spec.csi.enabled flag in the Portworx StorageCluster specification dictates whether the Container Storage Interface (CSI) driver is deployed within the Kubernetes environment. Setting this flag to false means that the CSI driver will not be installed or enabled, effectively disabling the CSI functionality. The CSI driver is responsible for dynamic volume provisioning, attachment, and lifecycle management in Kubernetes clusters. Disabling CSI might be necessary in environments relying on legacy volume plugins or specific operational requirements. When CSI is disabled, Portworx will not support dynamic provisioning or other CSI-dependent features, which could limit functionality for Kubernetes storage operations. Portworx operator documentation explicitly states that disabling CSI **omits the CSI driver installation**, advising users to carefully consider the impact before setting this flag to false, especially in production environments requiring CSI functionality. [Pure Storage Portworx Operator Docs](#) source.

Question 12

Question Type: MultipleChoice

What is the correct procedure to upgrade a Portworx cluster from version 3.0 to 3.1 using the Portworx Operator?

Options:

- A- Execute the 'pxctl cluster upgrade ---version 3.1' command.
- B- Edit the StorageCluster CR and update the .spec.image parameter from portworx/oci-monitor:3.0 to portworx/oci-monitor:3.1.
- C- No manual upgrade is needed as Portworx will automatically upgrade to the latest version.

Answer:

B

Explanation:

Comprehensive and Detailed Explanation From Exact Extract:

Upgrading Portworx clusters managed by the Kubernetes Operator requires a declarative update to the StorageCluster Custom Resource Definition (CRD). Specifically, the administrator must edit the StorageCluster resource and update the .spec.image field to point to the new version image, such as changing portworx/oci-monitor:3.0 to portworx/oci-monitor:3.1. This change instructs the Operator to roll out the new image across the cluster nodes, performing a seamless upgrade with minimal downtime. The pxctl CLI does not perform upgrades in Operator-managed environments; it is primarily for direct cluster management. The Operator ensures orderly upgrade sequencing, node by node, handling pod restarts and health checks. Automatic upgrades without manual intervention are not currently supported to prevent unintentional disruptions. Official Portworx upgrade documentation details this procedure, emphasizing the importance of version pinning and controlled rollout for production stability and rollback capabilities during upgrades. [Pure Storage Portworx Upgrade Guide](#) source.

To Get Premium Files for Portworx-Enterprise-Professional Visit

<https://www.p2pexams.com/products/portworx-enterprise-professional>

For More Free Questions Visit

<https://www.p2pexams.com/pure-storage/pdf/portworx-enterprise-professional>

20%
DISCOUNT

P2P
exams