DUMPSARENA

Specialist - Systems Administrator, Elastic Cloud Storage (ECS) Exam

EMC DES-1B31

Version Demo

Total Demo Questions: 9

Total Premium Questions: 91

Buy Premium PDF

https://dumpsarena.co

sales@dumpsarena.co

sales@dumpsarena.co
dumpsarena.co

Topic Break Down

Topic	No. of Questions
Topic 1, ECS Concepts	2
Topic 2, ECS Architecture	14
Topic 3, ECS Installation and Configuration	15
Topic 4, ECS Administration	15
Topic 5, ECS Data Services	40
Topic 6, ECS Monitoring and Troubleshooting	5
Total	91

QUESTION NO: 1 - (SIMULATION)

What is the correct sequence of steps to configure Dell EMC Isilon CloudPools with ECS solution?

ANSWER: See the explanation for the answer

Explanation:

Correct sequence:

1. **Prepare ECS object storage.** Create or identify the ECS namespace, create an ECS object user in that namespace, generate/record the user's S3 access key and secret key, and create the bucket that will hold the CloudPools data. Record the ECS S3 endpoint (VIP/load balancer address and HTTPS port, if applicable).
2. **Add ECS to OneFS as a CloudPools cloud account.** In the Isilon/OneFS CloudPools configuration, create a cloud account using the ECS S3 endpoint and the ECS object user access key/secret key. Import/accept the ECS CA certificate if HTTPS certificate validation requires it.
3. **Create the CloudPool.** Create a CloudPool that uses the ECS cloud account and the ECS bucket. This establishes the target to which OneFS will archive file data.
4. **Create and apply a File Pool Policy.** Create a CloudPools file-pool policy/rule that selects the required files (for example, by path, age, size, or file type) and sets the target storage to the newly created CloudPool. Add/commit the rule in the active file-pool policy.
5. **Run the policy processing job.** Run or allow the SmartPools/FilePool policy job to run so that eligible files are archived to ECS. Verify CloudPools and ECS bucket activity afterward.

In short: `Configure ECS namespace/user/bucket → create OneFS ECS cloud account → create CloudPool → create/activate File Pool Policy → run SmartPools policy processing`.

The CloudPools license must be installed/enabled on the Isilon cluster before performing the OneFS CloudPools configuration.

QUESTION NO: 2

A storage administrator needs to delete an ECS S3 bucket.

What must be done before the bucket can be deleted?

A. Disable the namespace quota

B. Remove all objects from the bucket

C. Delete the object user

D. Remove the namespace from the VDC

ANSWER: B

Explanation:

The bucket must be empty before it can be deleted. ECS follows S3 bucket behavior in requiring all objects to be removed before the bucket deletion operation succeeds. This requirement prevents a bucket container from being deleted while it still contains application data.

If object versioning is enabled, all applicable object versions and delete markers must also be removed before the bucket is empty.

QUESTION NO: 3

Which Centera feature is supported in an ECS 3.0 CAS environment?

A. Shredding (DOD 5015.2)

B. Advance Retention

C. Erasure Coding

D. Single Instance Storage

ANSWER: D

Explanation:

Single Instance Storage is supported by ECS 3.0 when using the Centera Content Addressable Storage (CAS) interface. CAS applications write content as objects identified through content-derived identifiers. This content-addressable approach enables ECS to recognize identical content and retain a single stored instance while allowing the content to be referenced as needed. The capability is especially relevant to Centera migrations because applications designed for Centera's CAS behavior can continue to use content-addressed storage semantics on ECS without requiring each identical item to be physically retained as a separate full copy.

In an ECS CAS environment, the CAS API and its content-addressing model provide the compatibility foundation for this behavior. Single-instance handling reduces unnecessary capacity consumption for duplicate content and is therefore a key supported Centera-related capability in the ECS 3.0 CAS feature set. Dell EMC's Centera-to-ECS migration material identifies the feature compatibility between Centera and ECS, including the supported CAS capabilities. See the Centera to ECS Migration white paper and Dell's ECS documentation portal for ECS product documentation and CAS-related guidance.

QUESTION NO: 4

Which ECS feature allows it to aggregate multiple small data objects queued in memory and then write them in a single disk operation?

A. Box-carting

B. Chunk management

C. Meta-data management

D. Erasure coding

ANSWER: A

Explanation:

Box-carting is the ECS write-optimization feature that accumulates multiple small objects in memory and groups them into a larger write before committing the data to disk. This reduces the number of individual disk operations that would otherwise be required for small-object workloads. By converting many small, random-style writes into a more efficient larger operation, ECS improves disk utilization and write performance while retaining each object's logical identity and associated metadata. The feature is particularly relevant to object-storage workloads, where applications can generate very large numbers of relatively small objects. ECS handles this aggregation internally, so client applications continue to use normal object protocols and do not need to package objects themselves. Dell ECS documentation describes the platform architecture and its object-storage data services, including the mechanisms used to efficiently persist object data across the storage nodes. See the Dell ECS product documentation and the ECS overview reference for additional ECS architecture information.

QUESTION NO: 5

An application written for an EMC Centera environment must connect to ECS by using the CAS interface.

Which file is used by the application to identify the CAS connection information?

A. PEA file

B. License file

C. Certificate signing request

D. Bucket policy file

ANSWER: A

Explanation:

A **PEA file** is used by Centera CAS applications to provide connection information and authentication details for the CAS environment. The PEA file contains the pool entries used by the application to locate the target CAS service and establish communication with ECS.

This allows existing Centera-compatible applications to access ECS through the CAS API without requiring the application to use an S3 or Swift interface. Dell ECS documentation provides release-specific CAS configuration and data-access guidance at Dell ECS Product Documentation.

QUESTION NO: 6

Which system admin user is provided to access the ECS UI for the first time?

A. root

B. admin

C. ecsadmin

D. sysadmin

ANSWER: A

Explanation:

The **root** user is the built-in ECS system-administrator account supplied for initial access to the ECS management UI. During initial deployment and configuration, this account is used to sign in to the portal and perform foundational administration, including configuring authentication providers, creating or managing administrative users, and establishing the access model required for ongoing operations. Because it is a system-level administrative identity, the root account has the privileges needed to administer ECS at the global level rather than being limited to an individual namespace or tenant. After the initial login, administrators should follow their organization's security procedures: protect the root credentials, assign appropriate roles to named administrative accounts, and use externally managed authentication where required. Dell ECS documentation distinguishes system-administrator access from namespace-level user access and describes administration of users, roles, and identity providers as part of ECS security configuration. See the Dell ECS product documentation and the Dell ECS user-management guidance for ECS administration and user-access information.

QUESTION NO: 7

An administrator has configured ECS to use NFSv3. What feature does this allow?

A. Global locking

B. Object storage access

C. Single protocol access

D. File gateway

ANSWER: A

Explanation:

Global locking is enabled by ECS NFSv3 support. ECS presents an NFS namespace backed by its distributed object-storage platform and coordinates lock state across the ECS nodes that can service requests. This distributed, cluster-wide locking behavior is important when multiple NFS clients, potentially connected through different ECS nodes, access the same file. It

ensures that file-lock requests are handled consistently throughout the ECS environment, supporting coordinated concurrent access rather than treating each node as an independent file server.

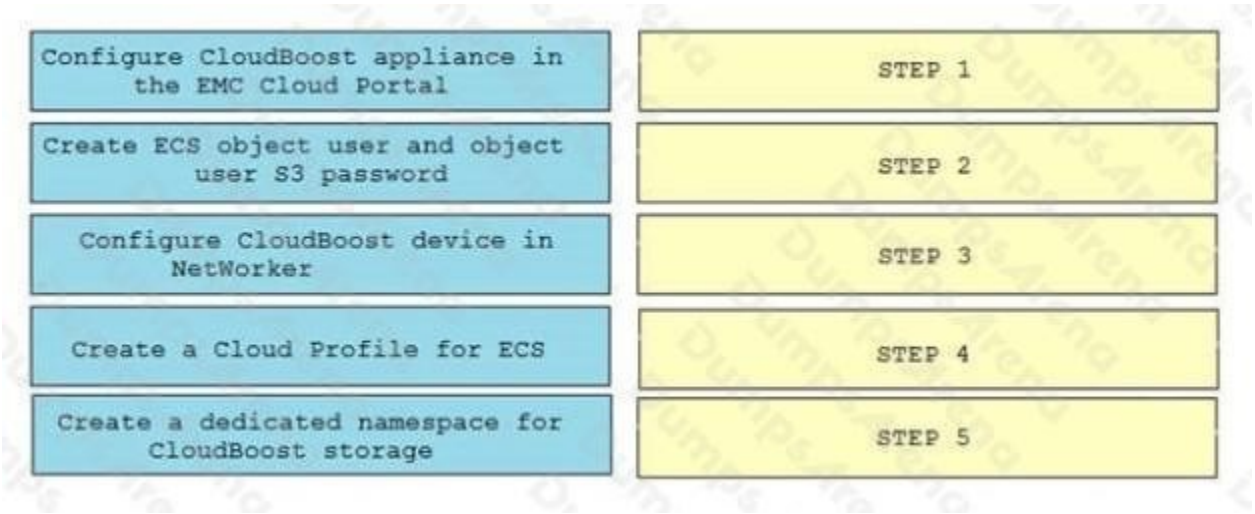
NFS version 3 defines the Network Lock Manager protocol operations used by clients for advisory file locking, including lock acquisition, testing, cancellation, and unlock operations. ECS implements the NFS service with global lock management so the locking view is maintained across the system's global namespace. This makes NFSv3 a suitable access method when applications require traditional file-level locking semantics while storing their data on ECS. See the ECS product documentation available through Dell ECS Support Documentation and the NFS version 3 specification, including lock-management procedures, in RFC 1813.

QUESTION NO: 8 - (DRAG DROP)

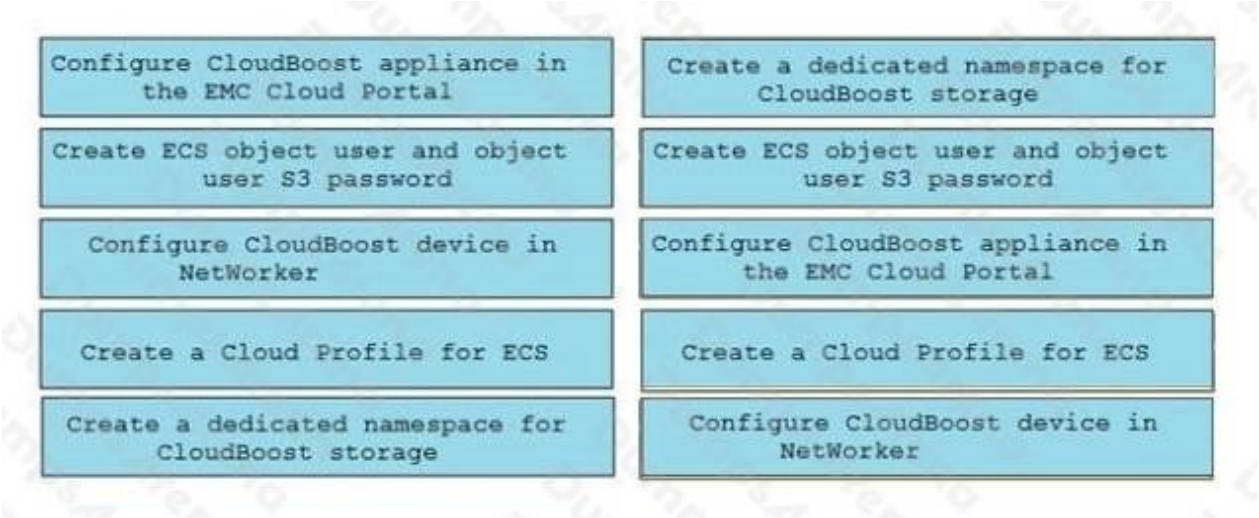
DRAG DROP

What is the correct sequence of steps to configure a NetWorker CloudBoost with ECS solution?

Select and Place:



ANSWER:



Explanation:

A CloudBoost with ECS deployment is built from the storage layer upward, so the sequence starts by creating a dedicated ECS namespace for CloudBoost storage. The namespace provides the isolated logical location in ECS where CloudBoost stores its data. Establishing this boundary first ensures that every later configuration item is associated with the intended ECS storage area.

Next, create the ECS object user and generate the object user's S3 password. CloudBoost accesses ECS through the S3-compatible object interface, so the user identity and its S3 credentials must exist before the appliance or NetWorker can authenticate to ECS. These credentials authorize the solution to use the namespace created for CloudBoost.

After the ECS namespace and access credentials are ready, configure the CloudBoost appliance in the EMC Cloud Portal. At this stage, the appliance can be supplied with the ECS endpoint, namespace context, and S3 authentication information needed to connect to object storage. The appliance is the CloudBoost component that manages deduplicated backup data before it is stored in ECS.

With the appliance configuration established, create a Cloud Profile for ECS in NetWorker. The cloud profile records the ECS connection details for NetWorker and provides the storage definition that NetWorker uses when integrating the CloudBoost environment. Finally, configure the CloudBoost device in NetWorker. This device is the NetWorker storage resource that ties backup operations to the already configured CloudBoost appliance and ECS-backed cloud profile.

This dependency order makes the storage destination, authentication, appliance connectivity, NetWorker cloud definition, and backup device available in the order each subsequent component needs them. Dell's current documentation entry points for the platforms are available through the ECS documentation page and the NetWorker support documentation page.

QUESTION NO: 9

An ECS is configured as backend storage for OpenStack Cinder. Which type of data will it store?

A. Disk and server images

B. OpenStack Swift client access information

C. Multimedia files

D. Volume data backups

ANSWER: D

Explanation:

Volume data backups is correct. In this integration, ECS provides object storage used by the OpenStack Cinder Backup service. When a Cinder volume backup is created, Cinder reads the volume's block data and writes the resulting backup data to the configured object-storage backup target. ECS can provide this target through its compatible object protocols, including the OpenStack Swift API. The stored content is therefore the backup representation of Cinder volumes, rather than the active block volumes themselves.

This design separates primary block storage from backup storage and lets Cinder restore a volume by retrieving the backup data from the object-storage repository. Cinder's administration documentation describes volume backups as being stored through a configured backup driver and identifies object storage, including Swift-based storage, as a backup destination. Dell ECS is designed to expose object-storage services, including Swift compatibility, making it suitable for this Cinder backup use case. See the OpenStack Cinder volume backup documentation and Dell's ECS product information.