DUMPSBOSS.

Cloud Pak for Integration v2021.4 Solution Architect

IBM C1000-147

Version Demo

Total Demo Questions: 8

Total Premium Questions: 87

Buy Premium PDF

https://dumpsboss.co

support@dumpsboss.co

support@dumpsboss.co
dumpsboss.co

Topic Break Down

Topic	No. of Questions
Topic 1, Planning	19
Topic 2, Architecture	40
Topic 3, Installation	2
Topic 4, Configuration	11
Topic 5, Security	7
Topic 6, Operations	6
Topic 7, Troubleshooting	2
Total	87

QUESTION NO: 1

Which active-standby pair of queue managers uses a shared networked file system?

A. Double resilient queue manager

B. Replicated data queue manager

C. Integrated HA queue manager

D. Multi-instance queue manager

ANSWER: D

Explanation:

Multi-instance queue manager is correct because IBM MQ implements this high-availability arrangement with two queue-manager instances that use the same queue-manager data and log directories on shared network storage. One instance owns the queue manager and runs as the active instance, while the other starts as a standby instance. IBM MQ coordinates access to the shared files through locks, ensuring that only the active instance can update queue-manager data at a given time. If the active instance fails or its connection to the shared storage is lost, the standby instance can acquire the required locks and become active. This provides automated local high availability without requiring applications to be redesigned; appropriately configured client reconnection can reconnect to the newly active instance. The shared file system must be accessible from both hosts and must satisfy IBM MQ requirements for reliable shared storage and locking behavior. IBM documents multi-instance queue managers as an active/standby configuration specifically based on shared networked storage. See the IBM MQ documentation on multi-instance queue managers and the configuration guidance for a multi-instance queue manager.

QUESTION NO: 2

What is the recommended minimum number of machines required for the smallest OpenShift v4 cluster?

A. 1 temporary bootstrap machine, 3 control plane machines, and at least 2 worker/compute machines

B. 1 temporary bootstrap machine, 1 control plane machine, and at least 2 worker/compute machines

C. 1 control plane machine and at least 3 worker/compute machines

D. 1 temporary bootstrap machine and 1 control plane machine

ANSWER: A

Explanation:

The recommended minimum configuration is **1 temporary bootstrap machine, 3 control plane machines, and at least 2 worker/compute machines**. A standard highly available OpenShift 4 cluster uses three control plane machines so that the control-plane etcd datastore can maintain quorum and continue operating through the loss of one control-plane node. The bootstrap machine is required only during installation: it provides the initial services needed to bring up the control plane, after which it is removed from the cluster. At least two compute machines provide capacity for application workloads and allow workload placement to remain available when a compute node is unavailable. This is the conventional smallest production-oriented OpenShift topology and is the sizing baseline relevant to Cloud Pak for Integration architecture planning. Red Hat documents the minimum standard cluster architecture as three control-plane nodes and two worker nodes, with a temporary bootstrap host used during installation. See the OpenShift installation architecture overview and the OpenShift minimum resource requirements documentation.

QUESTION NO: 3

What are two characteristics of storage classes that are required for API management capability deployment?

A. Any storage must be of type File System

B. Depending on the use-case both block and file system storage classes are supported

C. Only block storage classes are supported

D. Block storage for distributed and file system storage for localized

E. Requires an access mode of Read-Write-Many

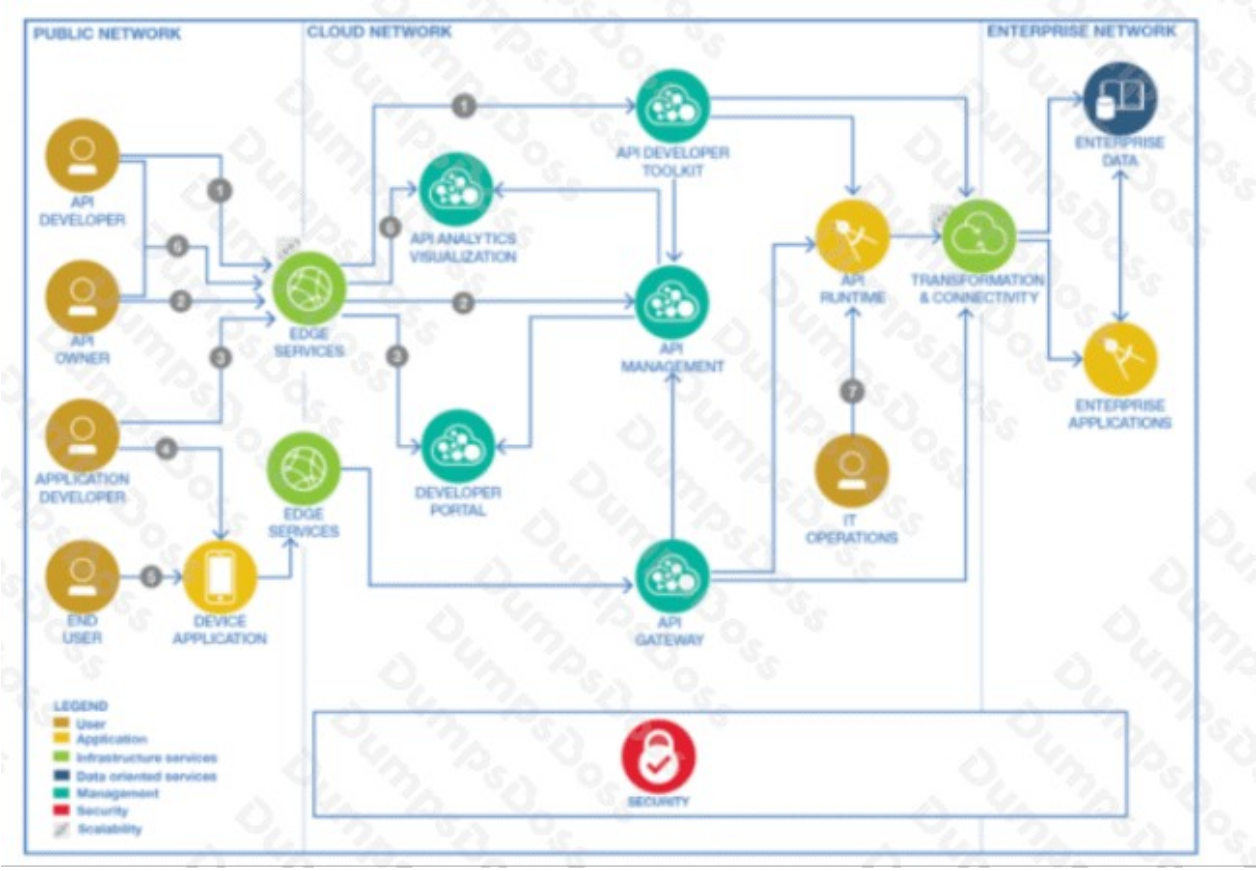
ANSWER: B E

Explanation:

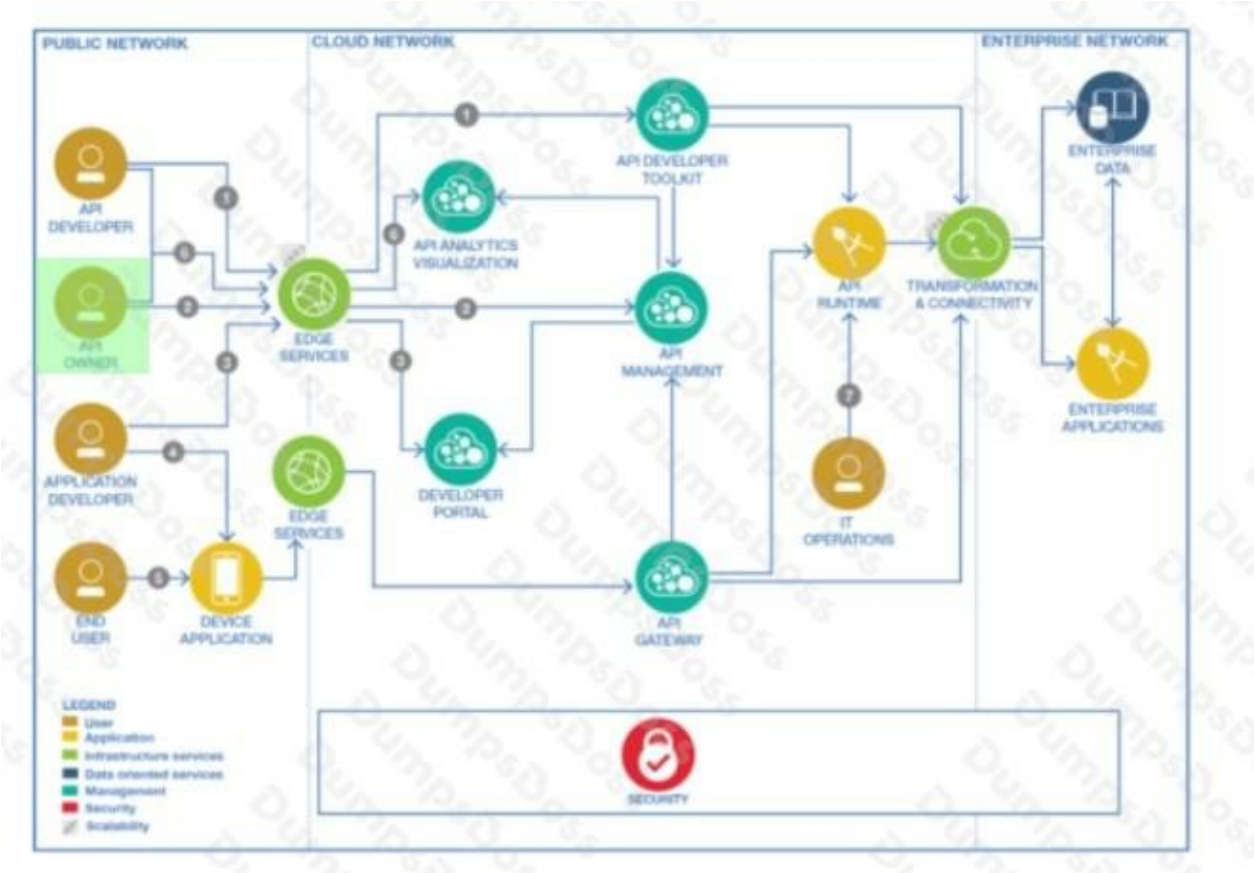
API management capability deployment uses multiple API Connect subsystems and workloads, whose persistence needs are not identical. Consequently, the platform supports both block and file-system storage classes according to the workload and deployment use case. Block storage is appropriate for stateful components that use individually attached persistent volumes, while shared file storage is needed where a volume must be mounted concurrently by more than one pod. The required shared-storage capability is expressed through the `ReadWriteMany` access mode: it permits multiple nodes to mount the same persistent volume with read/write access. A storage environment for an API management deployment must therefore provide suitable storage classes for the relevant block and shared-file use cases, including a class capable of RWX access where shared persistence is required. IBM's API Connect deployment guidance identifies storage as a prerequisite and describes the storage requirements of the API Connect subsystems. Red Hat's Kubernetes storage documentation defines access modes, including RWX, and explains that the mode determines how a volume can be mounted by nodes. See IBM API Connect storage requirements and Red Hat OpenShift persistent storage documentation.

QUESTION NO: 4 - (HOTSPOT)

In the image below, select the user role that can include an API endpoint to existing API products and plans, specifies access control, and publishes the API to the developer portal for external discovery.



ANSWER:



Explanation:

The correct role is **API Owner**. This role is responsible for governing how an API is packaged, exposed, and made available to API consumers. In an API management lifecycle, an API is not simply an implementation endpoint: it needs to be organized into consumable API products, associated with plans that describe the available usage offering, and protected with the appropriate access controls. The API Owner performs these product-oriented and publication-oriented activities so that the API can be safely offered to external developers.

Publishing to the developer portal is particularly an ownership responsibility because the portal is the discovery and self-service experience for consumers. Once published, external application developers can discover the API product, review its documentation, select a plan, and request or obtain the credentials needed to invoke it. The API Owner therefore controls the commercial and governance-facing presentation of the API, including who may consume it and under what plan and policy conditions.

This division of responsibilities supports a clear API lifecycle: implementation work produces the API interface and backend behavior, while ownership work makes that interface a governed, consumable product. IBM API Connect documentation describes API products and plans as the artifacts used to package APIs and manage consumer access, and it documents publishing those products for developer-portal consumption. See IBM's API product documentation and developer portal publishing documentation for the relevant lifecycle concepts. Therefore, the highlighted API Owner user icon is the correct selection.

QUESTION NO: 5

Which two recovery objectives should be defined when planning disaster recovery for Cloud Pak for Integration workloads?

A. Recovery Time Objective (RTO)

B. Recovery Point Objective (RPO)

C. StorageClass

D. Availability Zone

E. Horizontal Pod Autoscaler

ANSWER: A B

Explanation:

Recovery Time Objective (RTO) and **Recovery Point Objective (RPO)** are fundamental disaster-recovery objectives. RTO defines the maximum acceptable period that a service can be unavailable after an incident. It drives decisions about recovery automation, standby environments, operational procedures, and the speed at which workloads must be restored.

RPO defines the maximum acceptable amount of data loss measured in time. It influences backup frequency, replication design, message retention, database recovery strategy, and the acceptable lag between a primary environment and a recovery environment. Availability zones and storage classes are important architecture considerations, but they are not recovery objectives.

IBM provides Cloud Pak for Integration guidance for disaster recovery planning.

QUESTION NO: 6

Which Istio resource is used to define traffic-routing rules for requests to a service in a Service Mesh?

A. VirtualService

B. DestinationRule

C. Gateway

D. ServiceEntry

ANSWER: A

Explanation:

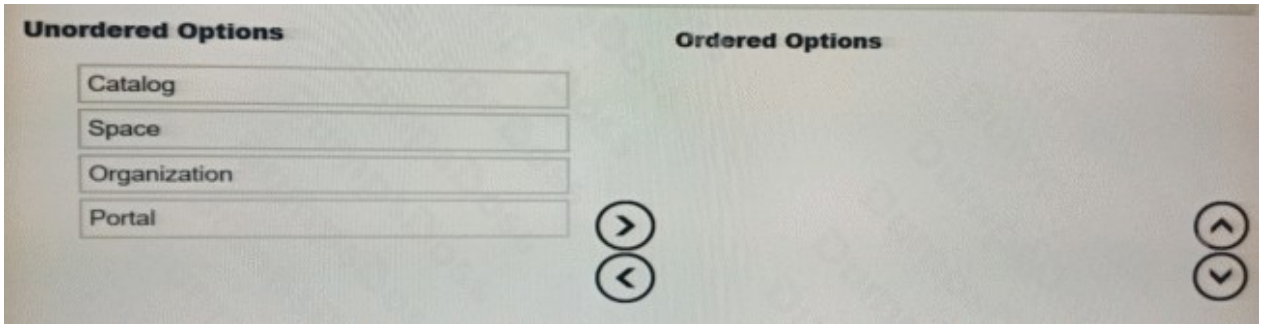
A **VirtualService** defines how requests are routed to a service within an Istio service mesh. It can specify hosts, route destinations, URI or header matching, retries, timeouts, and traffic-splitting behavior. For example, a VirtualService can direct a percentage of requests to a new version of a service while the remainder continue to use the existing version.

A DestinationRule defines policies that apply after traffic reaches a service, such as subsets and load-balancing settings. A Gateway configures ingress or egress traffic at the edge of the mesh, while a ServiceEntry adds an external service to the mesh service registry. See the Istio VirtualService reference.

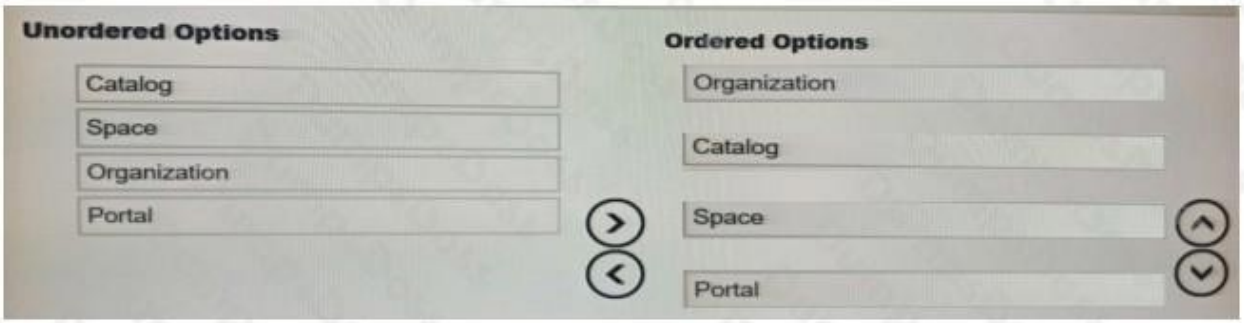
QUESTION NO: 7 - (DRAG DROP)

Select all that apply

What is the correct order to configure TLS profiles for an API's access and encryption? List the API Connect components to be configured, from the first to the last one.



ANSWER:



Explanation:

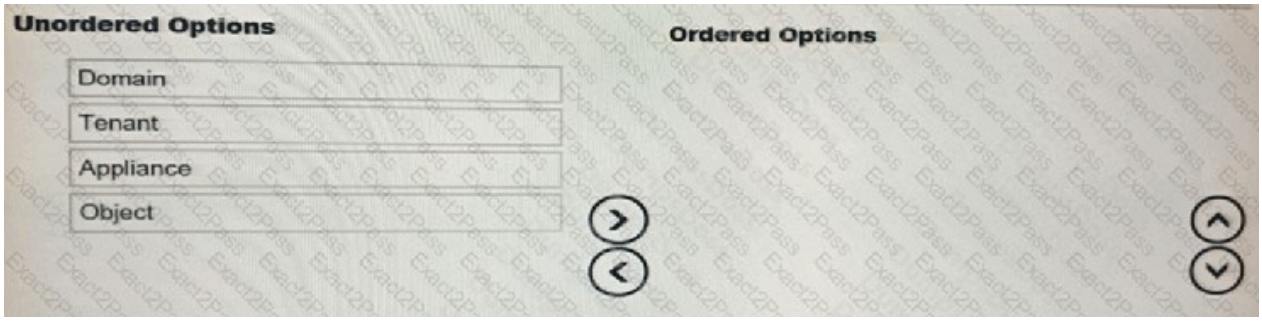
The correct sequence is **Organization, Catalog, Space, Portal**. API Connect configuration follows its resource hierarchy, beginning at the provider organization. The organization is the owning administrative boundary for provider-side API assets and settings, so it establishes the top-level context in which TLS profile configuration can be governed. Configuring it first ensures that certificate and TLS-related controls are defined within the appropriate provider organization.

Next, configuration proceeds to the catalog. A catalog belongs to an organization and is the environment in which APIs are published and made available, such as a development, test, or production environment. TLS profiles and their use for API access are therefore refined at the catalog level after the organizational context exists. A space is a subdivision of a catalog, used to separate teams or API work within that catalog. Its TLS-related configuration logically follows the catalog because a space cannot exist independently of its parent catalog.

Finally, the portal is configured. The Developer Portal is associated with the catalog and provides the consumer-facing experience for discovering APIs and accessing published API documentation and applications. Completing portal setup after the organization, catalog, and space context ensures that its secure access configuration is aligned with the API environment it exposes. This order follows the API Connect resource model and supports consistent inheritance and administration of security settings. IBM's API Connect documentation describes the provider organization, catalog, and space hierarchy in its API Connect concepts, and documents TLS configuration under Configuring TLS profiles.

QUESTION NO: 8 - (SIMULATION)

What is the correct order of dependencies in the DataPower Gateway services point of view as they are nested into each other?



ANSWER: See the explanation for the answer

Explanation:

The correct nesting order, from the outermost scope to the innermost configuration item, is:

In the DataPower Gateway services point of view, an appliance can contain tenants; each tenant contains domains; and domains contain configuration objects, such as services, handlers, policies, and cryptographic objects.