



Up-to-date Practice Test with Latest Questions and Answers covering latest syllabus and topics of the exam. Makes you ready to face actual exam.



Consul-Associate-003 Practice Questions
Consul-Associate-003 Practice Test
Consul-Associate-003 Practice Exam
Consul-Associate-003 Exam Questions
Consul-Associate-003 Study Guide



killexams.com

HashiCorp

Consul-Associate-003

Consul Associate (003)

ORDER FULL VERSION



<https://killexams.com/pass4sure/exam-detail/Consul-Associate-003>

Question: 1413

A platform team is using Consul to automate the configuration of a cloud load balancer. They notice that every time a single pod in Kubernetes restarts, the cloud load balancer is updated, causing a brief moment of instability. Which CTS (Consul-Terraform-Sync) configuration options can help mitigate this?

- A. Utilizing 'condition' blocks to filter out transient changes
- B. Increasing the 'consul_kv' timeout
- C. Using a 'quiescence' period in the task configuration
- D. Setting a 'max_parallel' limit on Terraform runs
- E. Validating the 'health_status' before triggering an update

Answer: A,C,E

Explanation: To prevent "churn" in automated infrastructure, CTS provides a quiescence period (or buffer), which waits for a period of stability after a change before running Terraform. This allows multiple events (like several pods restarting) to be batched into a single update. Furthermore, tasks can be configured to only trigger on certain health status transitions or specific conditions, ensuring that transient or irrelevant updates do not cause unnecessary infrastructure changes.

Question: 1414

Consul agents joining via `retry_max_attempts = 0` will attempt infinite retries for both LAN and WAN joins until manual leave, overriding the default finite retry behavior to ensure eventual membership in unstable networks.

- A. False
- B. True

Answer: A

Explanation: False. `retry_max_attempts = 0` (default) means infinite retries with exponential backoff until success or shutdown; however, it applies per-config (LAN/WAN separate), but agents still respect gossip failure detection and can be marked left/failed externally.

Question: 1415

To prepare for a full TLS and gossip encryption audit, the administrator must generate all necessary artifacts using Consul CLI tools. They run commands to create the foundation for TLS. What is the first command that must be executed to establish the private CA needed for signing server and client certificates in a secure Consul deployment?

- A. consul keygen
- B. consul operator keyring
- C. consul tls ca create
- D. consul tls cert create -server

Answer: C

Explanation: The `consul tls ca create` command is the first that must be executed to establish the private CA needed for signing server and client certificates, forming the basis for TLS encryption in Consul agent communication.

Question: 1416

You are deploying Consul in a Kubernetes environment where security is a high priority. You want to enable Gossip encryption. Which steps are necessary when using the Consul Helm chart?

- A. Create a Kubernetes Secret containing the gossip encryption key
- B. Set `global.gossipEncryption.autoGenerate` to true to let the chart handle key creation
- C. Set the `encrypt` flag in the `server.extraConfig` HCL block
- D. Use the `consul-k8s` CLI to manually inject the key into every pod
- E. Reference the secret in the `global.gossipEncryption.secretName` Helm value

Answer: A,B,E

Explanation: The Consul Helm chart provides built-in support for gossip encryption. You can either manually create a Kubernetes Secret and point the chart to it via `secretName`, or you can set `autoGenerate` to true, in which case the Helm installation process will create the secret for you. Both methods are much more manageable than manually injecting keys or using `extraConfig` blocks for this specific purpose.

Question: 1417

Your team is troubleshooting a Consul client agent on a VM that successfully starts but does not appear in the member list of the datacenter despite correct network connectivity to servers. The client configuration includes a `retry-join` address. What additional agent configuration is required to ensure proper participation in the LAN gossip pool and membership propagation?

- A. Set `gossip_lan` interval to a very low value for faster detection.

- B. Explicitly set `bind_addr` to the node's private IP and ensure `client_addr` allows local queries.
- C. Enable `server = true` temporarily on the client.
- D. Configure `bootstrap_expect = 0` explicitly in the client HCL.

Answer: B

Explanation: Explicitly setting `bind_addr` to the node's private IP and ensuring `client_addr` allows local queries is correct because the `bind_addr` controls the address used for gossip and RPC communication in the LAN pool, while `client_addr` defines where the agent listens for HTTP/DNS queries; misconfiguration here prevents the client from properly advertising itself via gossip to other members.

Question: 1418

When generating certificates for a Consul cluster using the 'consul tls' command, you have the option to create different types of certificates. Which of the following certificates are typically generated to support a full mTLS implementation?

- A. `consul-agent-ca.pem` (The Root Certificate)
- B. `dc1-client-consul-0.pem` (The Client Certificate)
- C. `consul-cli.pem` (The Operator Certificate)
- D. `consul-gossip-key.pem` (The Symmetric Key)
- E. `dc1-server-consul-0.pem` (The Server Certificate)

Answer: A,B,C,E

Explanation: A full mTLS setup requires a Root CA to sign all other certificates. Server certificates are needed for nodes in the server role, and client certificates are needed for agent nodes. Operator or CLI certificates are used for secure management. Gossip keys are not PEM-encoded certificates but rather symmetric strings, though they are part of the broader security configuration.

Question: 1419

During a Consul 1.17 upgrade in a production cluster, administrators observe that tokens created with templated policies on newer servers fail to resolve correctly on older agents still running 1.16. The cluster uses agent tokens for node registration and service tokens for Connect proxies. What is the recommended strategy to maintain secure ACL functionality across the rolling upgrade without temporarily weakening security?

- A. Use the `global-management` policy on all agent tokens during the upgrade window.
- B. Disable ACLs cluster-wide until the upgrade completes.
- C. Set `default_policy` to "allow" and rely on denylist rules in policies.
- D. Create legacy-style tokens with direct rules embedded until all nodes are upgraded, then migrate to

templated policies.

Answer: D

Explanation: The correct approach is to use legacy-style tokens with embedded rules for compatibility during the rolling upgrade, as newer templated policies are not fully recognized on pre-1.17 agents, while planning a post-upgrade migration to templated policies for simplified management.

Question: 1420

A security audit requires Consul intentions with wildcard source for admin-* services, set to permit for ops-team namespace. Deploying ops-team -> admin-db works, but ops-team-v2 fails despite namespace sync, due to missing service_meta tags in intention spec.

- A. False
- B. True

Answer: A

Explanation: True. Consul wildcard intentions match on exact prefix without meta propagation across namespace boundaries. service_meta must be identical or use namespace scoping in Kubernetes CRDs for cross-ns wildcards to apply.

Question: 1421

In a Consul on Kubernetes deployment using the Helm chart, you need to integrate with an existing external Consul server cluster in a different environment rather than deploying new servers inside the cluster. The goal is to run only client/dataplane components that join the remote servers securely. What Helm values structure enables this external server join pattern without deploying local servers?

- A. Set global.enabled = false, then enable client and connectInject, and specify externalServers.hosts with the remote server addresses.
- B. Set server.enabled = false and use server.join flags in a custom values file.
- C. Enable server.replicas = 0 and configure WAN federation parameters.
- D. Use the consul-k8s CLI with --preset client-only and provide join addresses.

Answer: A

Explanation: Setting global.enabled = false, then enabling client and connectInject, and specifying externalServers.hosts with the remote server addresses is correct because this Helm configuration disables local server deployment, activates client/dataplane mode, and directs the agents to join the provided external server hosts for membership and RPC forwarding as documented in external server integration patterns.

Question: 1422

A policy contains the following rules:

```
service_prefix "app-" { policy = "read" }
```

```
service "app-payment" { policy = "write" }
```

What is the effective permission for a token using this policy when trying to register the service "app-payment"?

- A. Read, because it is the most restrictive permission found
- B. Write, because exact matches take precedence over prefix matches
- C. Write, because the ACL system resolves to the most permissive rule
- D. Read-only, if the default_policy is set to "deny"
- E. Deny, because prefix rules always override exact matches

Answer: B,C,D

Explanation: Consul ACLs follow a "most specific match wins" logic. An exact service name match overrides a prefix match. If no exact match existed, the prefix would apply. If neither existed, the default_policy would be the final arbiter.

Question: 1423

Maintaining a Consul 1.19 cluster with gossip encryption enabled via `encrypt = "key"`, you rotate keys using `consul keyring install` but observe servers failing to join post-rotation with "gossip encryption mismatch." Is it false that `retry_join = ["provider=aws tag_key=Role tag_value=consul-server"]` ignores encryption during rejoin?

- A. False
- B. True

Answer: A

Explanation: False. Cloud auto-join providers like AWS tags still enforce gossip encryption validation during `retry_join`; key rotation requires `consul operator raft list-peers` to confirm all servers install the new key via `keyring` before rejoin. In Consul 1.19+, incomplete propagation causes mismatch, as SERF LAN/WAN layers decrypt incoming gossip independently of join method.

Question: 1424

A company is using Consul service mesh in a multi-datacenter setup with 'Primary' and 'Secondary'

datacenters. They want to ensure that if the 'auth' service in the 'Secondary' datacenter fails, traffic is automatically routed to the 'auth' service in the 'Primary' datacenter. Which configuration entries are needed to achieve this cross-datacenter failover?

- A. A 'service-router' to define the path-based routing to the other DC
- B. A 'service-resolver' for the 'auth' service with a 'failover' block
- C. The 'wan_join' must be healthy between the two Consul clusters
- D. The 'datacenters' list in the failover block must include the 'Primary' DC
- E. Mesh Gateways must be deployed and configured in both datacenters

Answer: B,C,D,E

Explanation: A 'service-resolver' configuration entry is the specific place where failover policies are defined, allowing you to specify a sequence of target datacenters. Mesh Gateways are required to facilitate the actual data plane traffic between the two isolated datacenter networks. The 'datacenters' list within the resolver's failover block tells Consul where to look for healthy instances if none are available locally. A healthy 'wan_join' is the control plane requirement that allows the 'Secondary' datacenter to discover the 'auth' service instances in the 'Primary' datacenter.

Question: 1425

For a DNS-heavy environment, the default token is restricted, and a dedicated DNS token is used. However, some prepared queries and catalog lookups still fail. The DNS token policy uses templated dns but the cluster has custom node metadata. What additional rule type might be needed if metadata-based DNS filtering is in use?

- A. node_prefix rules for read access to node metadata
- B. key_prefix for metadata storage
- C. operator = "read"
- D. No change; templated dns covers everything

Answer: A

Explanation: This scenario requires node_prefix "" { policy = "read" } or equivalent in the DNS token policy when node metadata or coordinates are used in DNS responses or filtering. The builtin/dns templated policy covers core catalog, but node details may need explicit read.

Question: 1426

Consul Dataplane components require persistent storage on each node to maintain local service state across restarts.

- A. True

B. False

Answer: B

Explanation: Unlike client agents lacking persistent storage in ephemeral environments like Kubernetes, Dataplane relies on orchestrator discovery and server catalog sync, eliminating per-node persistence needs for resilience.

Question: 1427

A team queries the service catalog using the HTTP API `/v1/health/service/web?passing=true` and receives instances with Node and Service fields. They notice some instances list "Checks" with Status "critical" despite the passing filter. Additionally, DNS queries for `web.service.consul` return fewer results than the API. Explain the behavior and the correct way to retrieve only passing instances consistently.

- A. The `passing=true` filter applies only to service-level checks, not node checks; use `?filter=Checks.Status=passing` or rely on DNS for strict healthy-only results.
- B. This indicates a bug in the agent sync; restart all clients to reconcile.
- C. The UI shows different results because it aggregates across datacenters.
- D. Prepared queries are interfering with the health endpoint; define a query instead.

Answer: A

Explanation: The `/v1/health/service` endpoint with `passing=true` parameter filters based on the service's aggregated health but may still surface node-level checks or edge cases in output. For precise healthy-only discovery, the DNS interface strictly returns only passing instances (no critical or warning), while API queries can use additional Filter expressions on `Checks.Status`. This ensures clients get reliable endpoints via DNS without manual filtering.

Question: 1428

An organization wants to integrate Consul with their existing Vault cluster to automatically manage the Service Mesh CA certificates. What are the benefits of using the 'vault' CA provider in Consul?

- A. Support for cross-datacenter CA hierarchies where Vault acts as the root of trust
- B. Centralized management of private keys and certificate policies within Vault
- C. Longer-lived certificates to reduce the frequency of rotation and network overhead
- D. Use of Vault's auditing and compliance features for all certificate issuance events
- E. Improved performance as Vault handles the intensive cryptographic operations for mTLS

Answer: A,B,D

Explanation: Integrating Consul with Vault for CA management offloads the security responsibility to a

dedicated tool. Vault provides centralized policy control, audit trails for every certificate issued, and the ability to act as a common root CA across multiple Consul datacenters, which simplifies cross-datacenter service mesh security.

Question: 1429

Interpreting a config with `tls.incoming.use_auto_encrypt = true` and `gossip_verify_outgoing.ca_file = "/vault-ca.crt"`, auto-encrypt provisions gossip-specific certs separate from RPC certs, enabling split-plane security.

- A. True
- B. False

Answer: A

Explanation: Auto-encrypt generates a single pair of client/server certs usable for both RPC and gossip verification when `gossip_verify_outgoing` references the same CA. The shared certs support split verification without separate provisioning.

Question: 1430

In a production Consul deployment, a single server agent node is sufficient to achieve high availability because the Raft algorithm can still guarantee strong consistency and fault tolerance with one node.

- A. False
- B. True

Answer: A

Explanation: False. Raft requires a quorum (a majority of voting nodes) to accept writes and elect a leader; a single server cannot form a quorum and therefore cannot tolerate any node failures. Production Consul deployments should run an odd number of voting server nodes (typically 3, 5, or 7) to ensure fault tolerance and automatic leader election in the event of node or zone outages.

Question: 1431

You are deploying Consul in a Kubernetes environment using the Helm chart. You want to use 'Consul snapshots' for backups. Which of the following are valid ways to manage or trigger snapshots in this environment?

- A. Using the 'consul-k8s' CLI to trigger a snapshot from an external workstation

- B. Configuring the 'snapshotAgent' in the Helm chart to run as a sidecar to servers
- C. Setting up a Kubernetes CronJob that interacts with the Consul HTTP API
- D. Executing 'consul snapshot save' from within a running Consul server pod
- E. Using a 'ServiceAccount' with the necessary permissions to call the snapshot API

Answer: A,C,D,E

Explanation: Backups in Kubernetes can be handled manually by exec-ing into a pod or automated via a CronJob. The CronJob needs a ServiceAccount and an ACL token with 'operator:read' permissions to call the snapshot API endpoint. The consul-k8s CLI also provides helper commands to facilitate these operations from outside the cluster. While a "snapshot agent" exists as an Enterprise feature, the standard Helm chart focuses on the core components and leaves automation of snapshots to Kubernetes native scheduling.

Question: 1432

A Kubernetes administrator is installing Consul using Helm. They want to ensure that only specific nodes in the Kubernetes cluster run the Consul server pods. Which Helm values should be used to control the placement of these pods?

- A. tolerations under the server section
- B. topologySpreadConstraints for the server pods
- C. podAntiAffinity to prevent multiple servers on the same node
- D. nodeSelector under the server section
- E. replicas to set the number of server pods

Answer: A,B,C,D

Explanation: To control pod placement in Kubernetes, nodeSelector is used to pin pods to nodes with specific labels. tolerations are required if the target nodes have taints (such as "dedicated=consul"). To ensure high availability, podAntiAffinity is used to ensure that Kubernetes does not schedule multiple Consul server pods on the same physical host. topologySpreadConstraints can also be used to distribute the pods across different availability zones or racks within the Kubernetes cluster.

Question: 1433

A production Consul cluster uses separate configuration files for different parameter groups (e.g., one for tls, one for encryption, one for join). When starting the agent with consul agent -config-dir pointing to the directory, in what order or manner are the files processed, and what ensures consistent application of bootstrap and server settings?

- A. All .hcl or .json files in the config directory are loaded and merged, with later files overriding earlier

ones for conflicting keys; place critical settings like `server` and `bootstrap_expect` in a base file loaded first.

- B.** Command-line flags override all directory-loaded files.
- C.** Only a single config file is supported per directory.
- D.** Files are processed alphabetically only, with no merging.

Answer: A

Explanation: Consul loads and merges all configuration files from the `-config-dir` directory (HCL or JSON), with parameters from later-loaded files overriding earlier ones where keys conflict; organizing critical settings like `server = true` and `bootstrap_expect` into an early-loaded base file ensures consistent behavior across the merged configuration.

Question: 1434

A developer is troubleshooting why their service is not appearing in the Consul DNS results. Upon investigation, they find the service is registered but the health check is failing. Which factors could lead to a 'critical' health status in Consul?

- A.** A script-based check is returning an exit code of 2 or higher
- B.** The 'interval' of the check is shorter than the time the script takes to execute
- C.** The TTL (Time To Live) check has not received a status update within the configured window
- D.** The local Consul agent on the node is unable to reach the Consul servers over the gossip protocol
- E.** An HTTP check is receiving a '404 Not Found' response from the health endpoint

Answer: A,C,D,E

Explanation: Consul marks a service as critical if its health checks fail. For scripts, any non-zero exit code other than 1 (warning) is a failure. HTTP checks consider any response outside the 2xx range (and 429) as a failure. If an agent loses contact with the servers, its local services may eventually be marked as stale or critical. TTL checks require periodic pulses from the application, and missing these results in an immediate critical state.

Killexams.com is a leading online platform specializing in high-quality certification exam preparation. Offering a robust suite of tools, including Exam Questions, practice tests, and advanced test engines, Killexams.com empowers candidates to excel in their certification exams. Discover the key features that make Killexams.com the go-to choice for exam success.



Practice Exam Questions Based on Current Exam Objectives

Killexams.com provides practice exam questions aligned with the latest official exam objectives and latest syllabus. Our content is reviewed and updated regularly to reflect recent changes announced by certification vendors. By studying these practice questions, candidates will cover the structure, difficulty level, and topics of the actual exam, helping them prepare more effectively and efficiently.

Comprehensive Practice Exams (PDF Format)

Killexams.com offers multiple-choice questions (MCQs) in easy-to-read PDF format, covering all major domains of the exam. Each PDF contains a structured collection of practice questions and verified answers designed to support focused study. These MCQs help candidates reinforce key concepts, identify knowledge gaps, and improve exam readiness through consistent practice.

Realistic Practice Tests (Online Test Engine & Desktop Test Engine)

To support hands-on preparation, Killexams.com provides practice tests through both an Online Test Engine and a Desktop Test Engine. These tools are designed to simulate a real exam environment, allowing candidates to practice under exam-like conditions, with latest syllabus and topics of the exam. Performance tracking, test history, and result analysis help users evaluate their progress and focus on areas that need improvement.

Risk-Free Purchase Policy

Killexams.com follows a transparent and customer-friendly purchase policy. If users are not satisfied with the study materials, they may request assistance or a refund in accordance with our published terms and conditions. This policy reflects our commitment to customer satisfaction, fairness, and confidence in our preparation resources.

Regularly Updated Content

Our practice question bank is reviewed and updated on an ongoing basis to stay aligned with the latest exam outlines and vendor updates. This ensures candidates are studying up-to-date, relevant material, and preparing with content that reflects current exam expectations, helping them stay confident and well-prepared.