DataStax

sandra-Operations-in-Kubernetes-Assoc

Cassandra Operations in Kubernetes Associate Datastax K8ssandra Exam

For More Information – Visit link below:

https://www.examsempire.com/

Product Version

1. Up to Date products, reliable and verified.
2. Questions and Answers in PDF Format.

Latest Version: 6.0

Question: 1

A cloud-native application is deployed using K8ssandra, featuring a stateless frontend that interacts with the stateful Cassandra cluster via the Stargate microservice API layen If an Ingress Controller is configured to route external traffic to Stargate's REST API endpoint for CRUD operations, which path prefix must the Ingress recognize to route the request to the correct Stargate service port?

- `/v1/auth`
- `/cqlsh/`
- `/v2/keyspaces`
- `/graphql-admin`
- `/metrics`

A. Option A
B. Option B
C. Option C
D. Option D
E. Option E

Answer: C

Explanation:
Stargate provides REST APIs for data access. The REST API provides access to a Cassandra table using information embedded in the REST URL endpoint, and the endpoint path must start with This base path causes the ingress to forward the request to the /v2/ keyspaces. Stargate service, typically on port 8082. Options A, D, and E relate to the Auth API, GraphQL administration, and monitoring, respectively.

Question: 2

A K8ssandraCluster manifest defines the stateful Cassandra data layer in a single datacenter (dc1 The datacenter requires 5Gi of persistent storage per node. Which configuration block correctly specifies this storage request, ensuring the stateless microservices layer relies on durable persistence?

- `spec: cassandra: datacenters: - metadata: name: dc1 storage: requests: 5Gi`
- `spec: cassandra: datacenters: - metadata: name: dc1 storageConfig: cassandraDataVolumeClaimSpec: resources: requests: storage: 5Gi`
- `spec: datacenters: - metadata: name: dc1 size: 3 storage: 5Gi`
- `spec: persistence: cassandraDataVolumeClaimSpec: size: 5Gi`
- `spec: cassandra: config: dataVolumeSize: 5Gi`

A. Option A
B. Option B
C. Option C
D. Option D
E. Option E

Answer: B

Explanation:
The storage requirements for the stateful Cassandra data component are defined using the `storageConfig` block within the `K8ssandraCluster` CRD. The persistent storage request (5Gi) for the Cassandra data volume is specified under `resources: requests: storage: 5Gi` within `cassandraDataVolumeClaimSpec`. Option B follows this required structure for declarative management of persistent storage in K8ssandra.

Question: 3

A stateless microservice client needs to authenticate with the stateful Cassandra cluster via the Stargate API using table-based authentication. The client has successfully retrieved the Cassandra superuser credentials from a Kubernetes Secret managed by the K8ssandra Operator. Which component receives these credentials directly in exchange for the required authorization token?

A. The Cass-operator Deployment/Pod
B. The Stargate Auth API Endpoint
C. The Cassandra Management API on the Seed Node
D. The Reaper Web Interface
E. The underlying Kubernetes API Server

Answer: B

Explanation:
Stargate's architecture includes Authentication Extensions responsible for access control. For table-based authentication, a client provides the Cassandra username and password (retrieved from the K8ssandra-managed Secret) in a POST request to the Stargate Auth API endpoint (e.g., /vl/auth). In response, the client receives an authToken to be used in the X-Cassandra-Token header for subsequent data requests.

Question: 4

A Java client application deployed alongside the stateless frontend connects to the Cassandra backend using the native CQL API provided by Stargate. To ensure proper load balancing and benefit from the separation of coordination (Stargate) and storage (Cassandra) layers, which modification is required for the DataStax Java driver configuration?

A. Replace the Round Robin policy with the Token Aware Policy.
B. Configure the driver to use the Stargate instance addresses as contact points instead of the Cassandra node addresses.
C. Enable Lightweight Transactions (LWTs) for all API interactions.

D.
Set the contact point to the Cassandra Headless service address (`<cluster>-<dc>-service`).
E.
Disable server-side encryption options via `client_encryption_options: enabled: false`.

Answer: B

Explanation:
When using native Cassandra drivers with Stargate, developers must use the addresses of the Stargate instances as contact points rather than the addresses of the Cassandra storage instances. Furthermore, the recommended Load Balancing Policy (LBP) when connecting via Stargate for CQL access is the Round Robin policy, explicitly replacing the Token Aware Policy, because Stargate acts as the coordinator/query planner. Stargate offloads the request coordination overhead from the storage instances.

Question: 5

In a cloud-native K8ssandra deployment featuring a stateless microservice API (Stargate) interacting with a stateful Cassandra backend, the K8ssandra Operator reconciles the high-level K8ssandraCluster Custom Resource into several native Kubernetes object types. Select the TWO primary Kubernetes resource kinds used to manage the runtime deployment structure for the individual Cassandra data nodes and the individual Stargate API nodes, respectively.

A. ConfigMap
B. Deployment
C. K8ssandraTask
D. StatefulSet
E. ClientConfig

Answer: B,D

Explanation:
For the stateful Cassandra nodes, the Cass Operator (used internally by K8ssandra Operator) creates a StatefulSet to manage persistence and consistent identity. Logical racks are mapped to StatefulSets. For the stateless Stargate component, the Stargate controller creates a Kubernetes Deployment. Deployments are ideal for managing stateless components like Stargate API nodes, allowing easy scaling and rollouts.

Question: 6

An SRE is setting up monitoring for a new K8ssandraCluster named 'demo'. The kube-prometheus-stack has been installed manually in the 'k8ssandra-operator' namespace. Which configuration snippet, when placed in the K8ssandraCluster manifest, correctly enables the automatic creation of Prometheus ServiceMonitors for the Cassandra nodes, Stargate, and Reaper components across the entire cluster?

A.

```
spec:
cassandra:
telemetry:
prometheus: { enabled: true }
stargate:
telemetry: { prometheus: { enabled: true } }
```

B.

```
spec:
cassandra:
telemetry:
prometheus: { enabled: true }
stargate:
telemetry: { prometheus: { enabled: true } }
reaper:
telemetry: { prometheus: { enabled: true } }
```

C.

```
spec:
all-components:
telemetry:
prometheus: { enabled: true }
```

D.

```
spec:
datacenters:
- metadata:
name: dc1
telemetry: { prometheus: { enabled: true } }
```

E.

```
spec:
monitoring:
provider: prometheus
enabled: true
```

Answer: B

Explanation:
To enable the automatic creation of Prometheus `ServiceMonitors` for core K8ssandra components, telemetry must be enabled explicitly under the configuration section for each component at the cluster level (or the datacenter level for Cassandra and Stargate, but Reaper telemetry is cluster-wide). The correct path is `.spec.cassandra.telemetry.prometheus.enabled: true`, `.spec.stargate.telemetry.prometheus.enabled: true`, and `.spec.reaper.telemetry.prometheus.enabled: true`. Option B correctly includes the necessary fields for Cassandra, Stargate, and Reaper.

Question: 7

When telemetry is enabled in K8ssandra, the operator leverages Kubernetes Custom Resources (CRs) to facilitate metrics scraping by Prometheus. For a Cassandra deployment named 'test-dc1', a ServiceMonitor named 'test-dc1 -cass-servicemonitor' is visible. Which specific Prometheus Operator CR is created automatically by the K8ssandra Operator to define how Prometheus locates and pulls metrics from Cassandra pods?

A. PrometheusRule
B. PodMonitor
C. AlertManager
D. ServiceMonitor
E. K8sMetricSource

Answer: D

Explanation:
The K8ssandra Operator integrates with Prometheus by creating ServiceMonitor resources. The ServiceMonitor CRD, provided by the Prometheus Operator, specifies label selectors to identify the Cassandra pods and configures how Prometheus should scrape and relabel the collected metrics. The operator creates a ServiceMonitor for each CassandraDatacenter when telemetry is enabled.

Question: 8

A Site Reliability Engineer (SRE) has installed the 'kube-prometheus-stack' and deployed a 'K8ssandraCluster' successfully. To access the preconfigured visualizations, the SRE must install the dashboards. Which Kubernetes resource type and specific label combination are used to load K8ssandra's preconfigured dashboards into the deployed Grafana instance?

- ○ `Secret` objects labeled `k8ssandra.io/dashboard: enabled`.
- ○ `PersistentVolumeClaim` objects used for Grafana storage.
- ○ `ConfigMap` objects labeled `grafana_dashboard: "1"`.
- ○ `GrafanaDatasource` objects referencing the Prometheus service.
- ○ `Deployment` objects annotated with dashboard URLs.

A. Option A
B. Option B

C. Option C
D. Option D
E. Option E

Answer: C

Explanation:
K8ssandra provides preconfigured Grafana dashboards. These dashboards are passed to Grafana as configuration using Kubernetes ConfigMaps. The ConfigMaps must be explicitly labeled with grafana_dashboard: "1" (or equivalent depending on the stack version/setup) for Grafana to pick them up for visualization.

Question: 9

K8ssandra v1 .5.0+ introduced the ability to switch metric collection from the legacy Metric Collector for Apache Cassandra (MCAC) to the new Management API for Apache Cassandra (MAAC) endpoint. When MAAC is used (MCAC disabled), OS level metrics (CPU, disk) are no longer automatically provided by the Cassandra agent. Which component and configuration setting is recommended by K8ssandra to replace the OS level metrics previously supplied by MCAC's underlying tool ('collectd')?

- ○ Deploying the Prometheus Operator's `node_exporter` directly onto Kubernetes worker nodes.
- ○ Enabling the Vector agent sidecar and configuring it with the `host_metrics` source and the `enrich_host_metrics` transform.
- ○ Setting `spec.cassandra.telemetry.mcac.includeOsMetrics: true` while still keeping MCAC disabled.
- ○ Modifying the `ServiceMonitor` to scrape the Kubernetes kubelet endpoints for pod resource usage.
- ○ Using the `jmx_exporter` sidecar which natively supports OS metrics via JMX hooks.

A. Option A
B. Option B
C. Option C
D. Option D
E. Option E

Answer: B

Explanation:
MCAC relied on to provide OS level metrics, which is now deprecated in favor of Kubernetes-native solutions. When MCAC collectd is disabled (using MAAC for Cassandra metrics), K8ssandra recommends enabling the Vector agent sidecar and configuring it using the host_metrics source. This metric source, along with the transform, collects and enriches OS level metrics (disk space, CPU, load average, etc.) and routes them to Prometheus.

Question: 10

A K8ssandra operator observes that Prometheus is collecting an excessive number of time series data points, particularly table-level metrics, which is impacting the performance of the monitoring stack. The operator needs to implement efficient metric filtering. What TWO distinct methods are supported by

K8ssandra/Prometheus architecture for limiting the metrics collected, depending on the metrics agent in use? (Select TWO)

- ☐ Using the `spec.cassandra.telemetry.mcacMetricFilters` field to supply explicit `allow` or `deny` rules (e.g., `deny:org.apache.cassandra.metrics.Table`) directly in the `K8ssandraCluster` manifest.
- ☐ Configuring the `kube-prometheus-stack` Prometheus resource with `metric_relabel_configs` that target and drop metrics based on Kubernetes service labels.
- ☐ Updating the `reaper_db` keyspace configuration to set `enable_metrics_logging: false`.
- ☐ Defining Prometheus relabeling rules within a Kubernetes `ConfigMap`, which is referenced by the new MAAC metrics endpoint configuration.
- ☐ Modifying the `CassandraDatacenter` object to include a `jvm.options.metrics_blacklist` YAML snippet.

A. Option A
B. Option B
C. Option C
D. Option D
E. Option E

Answer: A,D

Explanation:

K8ssandra supports two primary methods for controlling metric volume based on the collection agent: 1. **Legacy MCAC:** Metrics filtering rules for MCAC can be defined directly in the `K8ssandraCluster` manifest using **`spec.cassandra.telemetry.mcacMetricFilters`**. This accepts explicit `allow` or `deny` rules (e.g., `deny:org.apache.cassandra.metrics.Table`). This validates Option A. 2. **New Metrics Endpoint (MAAC):** The new metrics endpoint introduced in v1.5.0+ supports configuration override via a ConfigMap, which is used to define filtering through standard Prometheus **relabeling rules**. This validates Option D. Option B describes a standard Prometheus function but not the specific K8ssandra integration mechanism for agent-side filtering. Option C and E use incorrect or non-existent configuration paths.

Visit us at: https://www.examsempire.com/cassandra-operations-in-kubernetes-associate

Thank You for Trying Our Product

Special 16 USD Discount Coupon: NSZUBG3X

Email: support@examsempire.com

Check our Customer Testimonials and ratings available on every product page.

Visit our website.

https://examsempire.com/