

[Google Discussions](#)

Exam Professional Cloud Security Engineer All Questions

View all questions & answers for the Professional Cloud Security Engineer exam

[Go to Exam](#)

EXAM PROFESSIONAL CLOUD SECURITY ENGINEER TOPIC 1 QUESTION 63 DISCUSSION

Actual exam question from Google's Professional Cloud Security Engineer

Question #: 63

Topic #: 1

[\[All Professional Cloud Security Engineer Questions\]](#)

For compliance reasons, an organization needs to ensure that in-scope PCI Kubernetes Pods reside on `in-scope` Nodes only.

These Nodes can only contain the

`in-scope` Pods.

How should the organization achieve this objective?

- A. Add a nodeSelector field to the pod configuration to only use the Nodes labeled inscope: true.
- B. Create a node pool with the label inscope: true and a Pod Security Policy that only allows the Pods to run on Nodes with that label.
- C. Place a taint on the Nodes with the label inscope: true and effect NoSchedule and a toleration to match in the Pod configuration.
- D. Run all in-scope Pods in the namespace `in-scope-pci`.

[Show Suggested Answer](#)

by  [Tabayashi](#) at April 29, 2022, 2:51 a.m.

Comments

Type your comment...

[Submit](#)

  **Tabayashi** Highly Voted  2 years, 9 months ago

[A] Correct answer. This is a typical use case for node selector.

<https://kubernetes.io/docs/concepts/scheduling-eviction/assign-pod-node/#nodeselector>

[B] The Pod Security Policy is designed to block the creation of misconfigured pods on certain clusters. This does not meet the requirements.

[C] Taint will no longer place pods without the "inscope" label on that node, but it does not guarantee that pods with the "inscope" label will be placed on that node.

[D] Placing the "in scope" node in the namespace "in-scope-pci" may meet the requirement, but [A] takes precedence.

   upvoted 11 times

  **MariaGabiGabriela** 2 years, 7 months ago



I think [A] does not stop other pods from being run in the PCI node, which is a requirement as the question states... I would go with [C]

   upvoted 8 times

  **AzureDP900** 2 years, 2 months ago

A is correct.

   upvoted 1 times

  **gcpengineer** 1 year, 8 months ago

C is correct



   upvoted 4 times

  **gcpengineer** Highly Voted  1 year, 8 months ago

Selected Answer: C

C is the ans as per chatgpt

   upvoted 7 times

  **Rakesh21** Most Recent  5 months, 4 weeks ago

Selected Answer: C

Taints and Tolerations are used in Kubernetes to control which Pods can be scheduled on which Nodes. By applying a taint to Nodes labeled as inscope: true with the effect NoSchedule, you ensure that only Pods that can tolerate this taint can be scheduled on these Nodes. Then, by configuring the in-scope Pods with a matching toleration, you guarantee that only these Pods will land on the Nodes marked as in-scope. This method ensures both that only in-scope Pods run on these Nodes and that these Nodes are used exclusively for in-scope Pods, meeting the compliance requirement.

   upvoted 1 times

  **JohnDohertyDoe** 7 months, 1 week ago

Selected Answer: C

Using a node selector does not prevent other pods from being scheduled in the pci-scope nodes. However a taint and toleration would ensure that only the pods with the toleration can be scheduled in the pci-scope nodes.

   upvoted 1 times

  **pico** 8 months, 1 week ago

Selected Answer: C

why the other options are less suitable:

A. nodeSelector: While nodeSelector can help target pods to specific nodes, it doesn't prevent other pods from being scheduled on those nodes if they fit the node's resources.

B. Node pool and Pod Security Policy: Pod Security Policies are deprecated in newer Kubernetes versions, and node pools alone won't guarantee the required isolation.

D. Namespace: Namespaces provide logical separation but don't inherently enforce node-level restrictions.

   upvoted 1 times

  **rsamant** 1 year, 1 month ago

A

<https://kubernetes.io/docs/concepts/scheduling-eviction/assign-pod-node/>

   upvoted 1 times

  **ArizonaClassics** 1 year, 4 months ago

C. Place a taint on the Nodes with the label inscope: true and effect NoSchedule and a toleration to match in the Pod configuration: This is the best solution. Taints and tolerations work together to ensure that Pods are not scheduled onto inappropriate nodes. By placing a taint on the Nodes, you are essentially marking them so that they repel all Pods that don't have a matching toleration. With this method, only Pods with the correct toleration can be scheduled on in-scope Nodes, ensuring compliance.

👍 ↩ 🚩 upvoted 2 times

🗄 👤 **Meyucho** 2 years, 1 month ago

Selected Answer: C

A nodeSelector configuration is from a pod template perspective. This question ask to PRESERVE some nodes for specific pods, so this is the main utilization for TAINT. This is a conceptual question and the answer is C

👍 ↩ 🚩 upvoted 4 times

🗄 👤 **AwesomeGCP** 2 years, 3 months ago

Selected Answer: A

A. Add a nodeSelector field to the pod configuration to only use the Nodes labeled inscope: true.

👍 ↩ 🚩 upvoted 3 times

🗄 👤 **GHOST1985** 2 years, 3 months ago

Selected Answer: A

nodeSelector is the simplest recommended form of node selection constraint. You can add the nodeSelector field to your Pod specification and specify the node labels you want the target node to have. Kubernetes only schedules the Pod onto nodes that have each of the labels you specify. => <https://kubernetes.io/docs/concepts/scheduling-eviction/assign-pod-node/#nodeSelector>

Tolerations are applied to pods. Tolerations allow the scheduler to schedule pods with matching taints. Tolerations allow scheduling but don't guarantee scheduling: the scheduler also evaluates other parameters as part of its function. => <https://kubernetes.io/docs/concepts/scheduling-eviction/taint-and-toleration/>

👍 ↩ 🚩 upvoted 3 times

🗄 👤 **fanilgor** 2 years, 4 months ago

Selected Answer: C

Basic K8s principles of scheduling workloads.
Taints and tolerations make perfect sense for this use case. Therefore C.

👍 ↩ 🚩 upvoted 2 times

🗄 👤 **Jeanphi72** 2 years, 5 months ago

Selected Answer: A

<https://redhat-scholars.github.io/kubernetes-tutorial/kubernetes-tutorial/taints-affinity.html>

A Taint is applied to a Kubernetes Node that signals the scheduler to avoid or not schedule certain Pods.
A Toleration is applied to a Pod definition and provides an exception to the taint.

<https://kubernetes.io/docs/concepts/scheduling-eviction/taint-and-toleration/>

Node affinity is a property of Pods that attracts them to a set of nodes (either as a preference or a ****hard requirement****).
Taints are the opposite -- they allow a node to repel a set of pods.

👍 ↩ 🚩 upvoted 3 times

🗄 👤 **hybridpro** 2 years, 7 months ago

Answer should be C. "These Nodes can only contain the `in-scope` Pods." - this can only be achieved by taints and tolerations.

👍 ↩ 🚩 upvoted 1 times



Platform

> Home

> Examtopics PRO

> All Exams

> Training Courses

