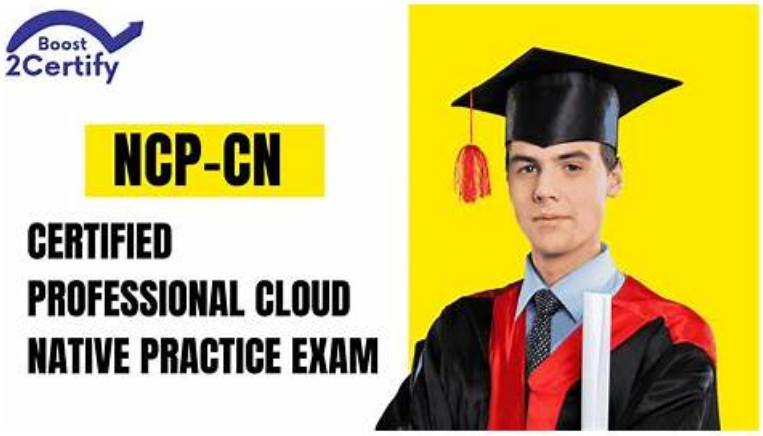
NCP-CN Test Preparation - NCP-CN Exam Cram Review



BTW, DOWNLOAD part of Free4Torrent NCP-CN dumps from Cloud Storage: https://drive.google.com/open?id=18Q7eQW5ANLeI-vy7Us8u2bhCcvexvCSY

In order to gain more competitive advantages when you are going for a job interview, more and more people have been longing to get a NCP-CN certification. They think the certification is the embodiment of their ability; they are already convinced that getting a NCP-CN certification can help them look for a better job. There is no doubt that it is very difficult for most people to pass the exam and have the certification easily. If you are also weighted with the trouble about a NCP-CN Certification, we are willing to soothe your trouble and comfort you.

Nutanix NCP-CN Exam Syllabus Topics:

Topic	Details
Topic 1	• Prepare the Environment for an NKP Deployment: This section of the exam measures the skills of infrastructure engineers and cloud administrators and covers the initial setup tasks needed for NKP deployment. Candidates must demonstrate the ability to seed a private container registry, create a bootstrap Kubernetes cluster, and determine license tiers suitable for clusters. They also need to prepare a bastion host for secure access, build machine images or prepare nodes for deployment, and gather all necessary information to build a cluster on the target cloud or on-premises provider.
Topic 2	• Conduct NKP Fleet Management: This section tests the abilities of platform administrators and cloud operations engineers in managing multiple clusters as a fleet. It focuses on configuring workspaces to organize clusters, deploying workload clusters within these workspaces, and attaching or detaching clusters as needed. Additionally, candidates must be able to configure projects for workload segmentation and manage platform applications that support the overall NKP environment.
Topic 3	• Manage Building an NKP Cluster: This section evaluates the skills of Kubernetes administrators and platform engineers in customizing and deploying NKP clusters. Candidates must show proficiency in tailoring cluster configurations to meet specific requirements and deploying Kommander, the management platform, while applying the appropriate licenses to enable cluster features and management capabilities.
Topic 4	• Perform Day 2 Operations: This part assesses the expertise of site reliability engineers and cluster operators in ongoing cluster management tasks after deployment. It includes configuring authentication and authorization mechanisms, setting up logging systems, and implementing cluster backup and recovery procedures. Candidates also need to demonstrate skills in monitoring cluster performance and health, configuring autoscaling to handle workload changes, and performing lifecycle management functions such as upgrades and maintenance.

>> NCP-CN Test Preparation <<

Free PDF 2026 Latest Nutanix NCP-CN: Nutanix Certified Professional - Cloud Native v6.10 Test Preparation

The simulation of the actual Nutanix NCP-CN test helps you feel the real NCP-CN exam scenario, so you don't face anxiety while giving the final examination. You can even access your last test results, which help to realize your mistakes and try to avoid them while taking the Nutanix NCP-CN Certification test.

Nutanix Certified Professional - Cloud Native v6.10 Sample Questions (Q63-Q68):

NEW QUESTION # 63
A Platform Engineer is deploying a new Kubernetes application in Amazon Web Services (AWS) Elastic Container Service for Kubernetes (EKS) and Azure Kubernetes Service (AKS). The engineer's team has decided to use a custom image instead of the default images provided by AWS or Azure for their clusters. What functionality will the engineer lose in both AWS EKS and Azure AKS by choosing to use a custom image?

- A. Native monitoring tools, logging, and alerting
- B. Ability to use GPUs and persistent storage
- C. Cluster networking and load balancing
- D. Built-in autoscaling and security capabilities

Answer: D

NEW QUESTION # 64
An administrator has been tasked with deploying NKP as the Kubernetes platform and needs to deploy their first cluster with the following requirements:
* Dark site (no Internet connectivity)
* Nutanix-provided Rocky Linux VM image
* AHV-based clusterWhat are two prerequisites to accomplish the deployment? (Choose two)

- A. Konvoy Image Builder
- B. Air-Gapped Bundle
- C. Self-managed AWS cluster
- D. Existing local container registry

Answer: B,D

Explanation:
The Nutanix Kubernetes Platform (NKP) is designed to simplify the deployment and management of Kubernetes clusters on Nutanix infrastructure, including on-premises AHV-based clusters in dark site environments with no Internet connectivity. The requirements specified in the question-dark site, Nutanix- provided Rocky Linux VM image, and AHV-based cluster-point to a deployment scenario where the environment must be self-contained and rely on Nutanix-specific tools and resources to meet the air-gapped constraints.
According to the Nutanix Kubernetes Platform Administration (NKPA) course, deploying NKP in a dark site environment requires specific prerequisites to ensure all necessary components, such as container images, dependencies, and configuration files, are available without Internet access. The course emphasizes the use of an Air-Gapped Bundle and an existing local container registry as critical components for such deployments.
* Air-Gapped Bundle (Option B):
* The NKPA course explains that for dark site deployments, Nutanix provides an Air-Gapped Bundle, which is a comprehensive package containing all the software components, container images, and dependencies required to deploy and manage an NKP cluster without Internet connectivity. This bundle includes the Kubernetes binaries, NKP platform applications (e.g., Rook Ceph, monitoring tools), and other necessary artifacts.
* The Nutanix Cloud Native (NCP-CN) 6.10 Study Guide specifically states: "For air-gapped environments, the Nutanix Air-Gapped Bundle is required to provide all dependencies, including container images and installation files, to deploy NKP clusters." This bundle is typically downloaded from the Nutanix Support Portal in an Internet-connected environment and then transferred to the dark site for deployment.
* The bundle ensures that the Nutanix-provided Rocky Linux VM image, which serves as the base operating system for the Kubernetes nodes, can be provisioned with all required software components. The NKPA course further notes that the Air-Gapped Bundle is tailored for AHV- based clusters, ensuring compatibility with the Nutanix hypervisor.

* Existing Local Container Registry (Option C):
* In a dark site environment, a local container registry is a prerequisite to store and distribute container images required by the NKP cluster. The NKPA course highlights that NKP relies on container images for Kubernetes components, platform applications, and user workloads. In an air-gapped setup, these images cannot be pulled from public registries like Docker Hub or Quay.
io.
* The course instructs administrators to set up a local container registry (e.g., Harbor, Nexus, or a Nutanix-managed registry) and populate it with the container images included in the Air-Gapped Bundle. The Nutanix Cloud Bible reinforces this, stating: "In air-gapped deployments, a local container registry must be pre-configured to host all required images, which are provided as part of the Nutanix Air-Gapped Bundle."
* The local container registry ensures that the Kubernetes nodes, running on the Nutanix-provided Rocky Linux VM image, can access the necessary images during cluster bootstrapping and operation. The NKPA course provides guidance on configuring the registry and integrating it with the NKP deployment process.
Incorrect Options:
* Konvoy Image Builder (Option A):
* Konvoy Image Builder is a tool associated with D2iQ's Konvoy platform, used to create custom machine images for Kubernetes deployments. While it can be used to build images for Kubernetes nodes, it is not a Nutanix-specific tool nor a prerequisite for NKP deployments. The NKPA course and NCP-CN 6.10 Study Guide do not mention Konvoy Image Builder, as NKP uses the Nutanix-provided Rocky Linux VM image, which is pre-configured for AHV-based clusters. This option is irrelevant to the Nutanix ecosystem.
* Self-managed AWS cluster (Option D):
* A self-managed AWS cluster is unrelated to the requirements of deploying NKP on an AHV- based cluster in a dark site. The question specifies an AHV-based cluster, which is Nutanix's Acropolis Hypervisor running on-premises, not a cloud-based AWS environment. The NKPA course focuses on Nutanix infrastructure (AHV, Prism Central) for NKP deployments and does not include AWS as a supported platform for this scenario. This option is incorrect as it contradicts the deployment environment.
Deployment Context:
* The Nutanix-provided Rocky Linux VM image is a pre-configured operating system image optimized for NKP deployments on AHV. The NKPA course notes that this image includes the necessary kernel settings, drivers, and configurations to run Kubernetes nodes efficiently on Nutanix infrastructure.
* The AHV-based cluster requirement indicates that the deployment leverages Nutanix's hypervisor, managed through Prism Central, to provision and manage the Kubernetes nodes. The Air-Gapped Bundle and local container registry ensure that all software components are available in the dark site, aligning with the NKPA course's guidelines for air-gapped deployments.
References:
Nutanix Kubernetes Platform Administration (NKPA) Course, Section on Preparing the Environment for NKP Deployment.
Nutanix Cloud Native (NCP-CN) 6.10 Study Guide, Chapter on Air-Gapped Deployments.
Nutanix Cloud Bible, NutanixKubernetesPlatform Section: https://www.nutanixbible.com Nutanix Support Portal, Air-Gapped Bundle Documentation: https://portal.nutanix.com Nutanix Kubernetes Platform Deployment Guide: https://www.nutanix.com
═══

NEW QUESTION # 65
When deploying an NKP cluster onto air-gapped, pre-provisioned servers, Konvoy Image Builder is utilized to prepare the servers to become NKP cluster nodes.
What does the konvoy-image upload command do as a part of this preparation process?

- A. The command uploads artifacts to the servers such as the container runtime, the OS bundle, and Kubernetes components.
- B. The command is used to upload OS hardening scripts to the server (must be client supplied).
- C. The command uploads artifacts to the servers such as the container runtime, the OS bundle, and Kubernetes components, including optional OS hardening scripts (must be client supplied).
- D. The command is used to create a konvoy userid on the servers, as well as upload artifacts to them such as the container runtime, the OS bundle, and Kubernetes components.

Answer: A

Explanation:
According to the NKPA 6.10 documentation under "Air-Gapped Preparation with Konvoy Image Builder," the konvoy-image upload command uploads essential artifacts to the target servers to prepare them to serve as cluster nodes. These artifacts include:
* Container runtime (containerd)
* OS bundle
* Kubernetes components
It does not involve user creation or OS hardening scripts (those are separate, client-driven processes).
Key reference from documentation:

"The konvoy-image upload command uploads the required artifacts to the target server(s) to prepare them for Kubernetes deployment in air-gapped environments. This includes the container runtime, OS bundle, and Kubernetes binaries." Reference: Nutanix Kubernetes Platform Administration (NKPA) 6.10 - "Air-Gapped Deployment Preparation" NCP-CN 6.10 Study Guide - "Konvoy Image Builder Workflow"

NEW QUESTION # 66
Some time ago, an EKS cluster was attached to be managed with NKP (Fleet Management). Now, a Platform Engineer has been asked to disconnect the EKS cluster from NKP for licensing reasons. After disconnecting the cluster, the developers realized that application changes are still being reflected in the EKS cluster, despite the fact that the EKS cluster was successfully detached from NKP. How should the engineer resolve this issue?

- A. Developers must have some bad configuration in the deployment config files. Ask for revision or call AWS technical support.
- B. Detached cluster must also be deleted from NKP: nkp delete cluster -c detached-cluster-name
- C. Forcefully detach EKS cluster: nkp detach cluster -c detached-cluster-name --force
- D. Detached cluster's Flux installation must be manually disconnected from the management Git repository: kubectl -n kommmander-flux patch gitrepo management -p '{"spec":{"suspend":true}}' -- type merge

Answer: D

Explanation:
When an Amazon EKS cluster is attached to NKP for fleet management, NKP uses GitOps principles, leveraging Flux (a GitOps operator) to synchronize application deployments and configurations from a management Git repository to the attached cluster. The NKPA course explains that detaching a cluster from NKP removes it from the NKP management plane, but the Flux installation on the cluster may continue to reconcile with the Git repository, causing application changes to persist post-detachment.
To resolve this, the engineer must manually disconnect the Flux installation by suspending the Git repository reconciliation. The correct command, as per the NKPA course, is: kubectl -n kommmander-flux patch gitrepo management -p '{"spec": {"suspend":true}}' --type merge. This command suspends Flux's synchronization with the management Git repository, stopping further application updates. The Nutanix Cloud Native (NCP- CN) 6.10 Study Guide states: "After detaching an external cluster from NKP, the Flux GitOps operator may continue to apply changes unless its GitRepo resource is suspended using kubectl patch in the kommmander- flux namespace." Incorrect Options:
* A. Forcefully detach EKS cluster: The --force flag is not a standard option for the nkp detach cluster command, and forceful detachment does not address the Flux reconciliation issue.
* B. Detached cluster must also be deleted from NKP: The cluster is already detached, and deletion is not necessary to stop GitOps updates. The issue lies with Flux, not NKP's state.
* C. Developers must have some bad configuration: The issue is not with developer configurations but with Flux's ongoing synchronization, as explained in the NKPA course.
:
Nutanix Kubernetes Platform Administration (NKPA) Course, Section on Fleet Management and GitOps.
Nutanix Cloud Native (NCP-CN) 6.10 Study Guide, Chapter on Detaching Clusters.
Nutanix Cloud Bible, NutanixKubernetesPlatform Section: https://www.nutanixbible.com Flux Documentation: https://fluxcd.io

NEW QUESTION # 67
A Platform Engineer is preparing machine images for NKP through the NIB or KIB process. What is the purpose of doing this?

- A. Tagging the image to be used specifically for NKP
- B. Creating a CAPI-compliant image for use as NKP cluster nodes
- C. Hardening an OS image with client-supplied hardening scripts
- D. Creating a custom user account for NKP admins to ensure access to NKP nodes

Answer: B

Explanation:
The Nutanix Kubernetes Platform (NKP) leverages Cluster API (CAPI) to manage the lifecycle of Kubernetes clusters. When preparing machine images for NKP deployment, the Nutanix Image Builder (NIB) or Kubernetes Image Builder (KIB) process is used to create custom machine images that are compatible with NKP's infrastructure requirements. According to the NKPA course, the primary purpose of this process is to create CAPI-compliant images that can be used as the base for NKP cluster nodes.
The NKPA course explains that NKP uses CAPI to provision and manage Kubernetes clusters, and CAPI requires machine images that meet specific criteria, such as including the necessary Kubernetes components, container runtimes, and operating system

configurations. The NIB/KIB process ensures that the images are pre-configured with these components, making them suitable for use as NKP worker and control plane nodes.
The Nutanix Cloud Native (NCP-CN) 6.10 Study Guide states: "The Nutanix Image Builder (NIB) or Kubernetes Image Builder (KIB) is used to create CAPI-compliant machine images that include the required OS, Kubernetes binaries, and dependencies for NKP cluster nodes." These images are typically based on supported operating systems like Rocky Linux or Ubuntu, as provided by Nutanix, and are customized to include the container runtime (e.g., containerd), kubeadm, and other dependencies required for CAPI-based cluster provisioning. The resulting images are stored in a location accessible to the NKP deployment process, such as a local registry or Nutanix Prism Central.
Incorrect Options:
* A. Hardening an OS image with client-supplied hardening scripts: While hardening the OS is a good practice, it is not the primary purpose of the NIB/KIB process. The NKPA course notes that hardening can be applied as part of image customization, but the core goal is to ensure CAPI compliance, not just hardening.
* B. Creating a custom user account for NKP admins to ensure access to NKP nodes: The NIB/KIB process does not focus on creating user accounts. User access is managed through Kubernetes RBAC or external identity providers, as covered in the NKPA course's authentication section.
* C. Tagging the image to be used specifically for NKP: Tagging may occur as part of image management, but it is not the primary purpose. The NKPA course emphasizes CAPI compliance over tagging.
:
Nutanix Kubernetes Platform Administration (NKPA) Course, Section on Preparing Machine Images.
Nutanix Cloud Native (NCP-CN) 6.10 Study Guide, Chapter on NKP Deployment Prerequisites.
Nutanix Cloud Bible, NutanixKubernetesPlatform Section: https://www.nutanixbible.com Cluster API Documentation: https://cluster-api.sigs.k8s.io

NEW QUESTION # 68
.....

Many IT certification exam dumps providers spend a lot of money and spirit on advertising and promotion about Nutanix NCP-CN exam lab questions but pay little attention on improving products' quality and valid information resource. They prefer low price strategy with low price rather than excellent valid and high-quality NCP-CN Exam Lab Questions with a little more cost. I think high passing rate products is what you need in fact.