CNPA考古題更新 - CNPA最新考證

順便提一下，可以從雲存儲中下載KaoGuTi CNPA考試題庫的完整版：https://drive.google.com/open?id=1ZjOOYy32KaAepuUl0JRIMZE6m9W3Pf6P

您是否在尋找可靠的學習資料來準備即將來的CNPA考試？如果是的話，您可以嘗試KaoGuTi的產品和服務。我們提供最新的Linux Foundation CNPA考古題是經過眾多考生和專家檢驗過的學習指南，保證成功率百分之百的考古題。對于購買CNPA題庫產品的客戶，我們還提供一年的免費更新服務。所以，您不必擔心，Linux Foundation CNPA學習指南不僅讓您更準確的了解考試的出題點，還能讓您更有范圍的學習相關知識，高效率的通過CNPA考試。

Linux Foundation CNPA 考試大綱：

主題	簡介
主題 1	• Platform Observability, Security, and Conformance: This part of the exam evaluates Procurement Specialists on key aspects of observability and security. It includes working with traces, metrics, logs, and events while ensuring secure service communication. Policy engines, Kubernetes security essentials, and protection in CI • CD pipelines are also assessed here.
主題 2	• Platform Engineering Core Fundamentals: This section of the exam measures the skills of Supplier Management Consultants and covers essential foundations such as declarative resource management, DevOps practices, application environments, platform architecture, and the core goals of platform engineering. It also includes continuous integration fundamentals, delivery approaches, and GitOps principles.
主題 3	• IDPs and Developer Experience: This section of the exam measures the skills of Supplier Management Consultants and focuses on improving developer experience. It covers simplified access to platform capabilities, API-driven service catalogs, developer portals for platform adoption, and the role of AI • ML in platform automation.
主題 4	• Platform APIs and Provisioning Infrastructure: This part of the exam evaluates Procurement Specialists on the use of Kubernetes reconciliation loops, APIs for self-service platforms, and infrastructure provisioning with Kubernetes. It also assesses knowledge of the Kubernetes operator pattern for integration and platform scalability.
主題 5	• Continuous Delivery & Platform Engineering: This section measures the skills of Supplier Management Consultants and focuses on continuous integration pipelines, the fundamentals of the CI • CD relationship, and GitOps basics. It also includes knowledge of workflows, incident response in platform engineering, and applying GitOps for application environments.

CNPA考古題更新 - 成功通過Certified Cloud Native Platform Engineering Associate的利刃

獲得CNPA認證是IT職業發展有利保证，而KaoGuTi公司提供最新最準確的CNPA題庫資料，幾乎包含真實考試的所有知識點，借助我們的學習資料，您不必浪費時間去閱讀過多的參考書籍，只需要花費一定的時間去學習我們的Linux Foundation CNPA題庫資料。本站提供PDF版本和軟件本版的CNPA題庫，PDF版本的方便打印，而對于軟件版本的Linux Foundation CNPA題庫可以模擬真實的考試環境，方便考生選擇。

最新的 Cloud and Containers CNPA 免費考試真題 (Q25-Q30):

問題 #25
Which of the following is a primary benefit of using Kubernetes Custom Resource Definitions (CRDs) in a self-service platform model?

- A. CRDs automatically manage the scaling and failover of platform services without additional configuration.
- B. CRDs eliminate the need for Role-based access control (RBAC) configurations in Kubernetes clusters.
- C. CRDs enable platform teams to define custom APIs without modifying the Kubernetes API server code.
- D. CRDs provide built-in support for multi-cloud deployments without additional tooling.

答案：C

解題說明：
Kubernetes Custom Resource Definitions (CRDs) extend the Kubernetes API by allowing platform teams to create and expose custom APIs without modifying the core Kubernetes API server code. Option C is correct because this extensibility enables teams to define new abstractions (e.g., Database, Application, or Environment resources) tailored to organizational needs, which developers can consume through a self- service model.
Option A is incorrect because scaling and failover are handled by controllers or operators, not CRDs themselves. Option B is wrong because RBAC is still required for access control over custom resources.
Option D is misleading because multi-cloud support depends on how CRDs and their controllers are implemented, not a built-in CRD feature.
By leveraging CRDs, platform teams can standardize workflows, hide complexity, and implement guardrails, all while presenting developers with simplified abstractions. This is central to platform engineering, as it empowers developers with self-service APIs while maintaining operational control.
References:- CNCF Platforms Whitepaper- Kubernetes Extensibility Documentation- Cloud Native Platform Engineering Study Guide

問題 #26
Which of the following statements describes the fundamental relationship between Continuous Integration (CI) and Continuous Delivery (CD) in modern software development?

- A. CI and CD are interchangeable terms; they both refer to the process of automating software release management.
- B. CI and CD are entirely separate practices; CI focuses on code quality, while CD focuses on infrastructure management.
- C. CD is a prerequisite for CI; CD automates the deployment of code and CI builds upon this by automating the integration of code changes.
- D. CI is a prerequisite for CD; CI automates the building and testing of code, and CD builds upon this by automating the release process.

答案：D

解題說明：
Continuous Integration (CI) and Continuous Delivery (CD) are complementary practices. Option A is correct:
CI is a prerequisite for CD. CI focuses on automating code integration by building, testing, and validating changes, ensuring code quality and early detection of defects. CD builds upon CI by automating the process of releasing validated builds into staging and production environments, making delivery repeatable and reliable.
Option B incorrectly treats them as entirely separate. Option C reverses the relationship, as CD cannot exist without CI pipelines.
Option D is inaccurate because CI and CD are not interchangeable-they represent distinct stages in the software delivery lifecycle.

Together, CI/CD accelerates software delivery, reduces risk, and improves quality. In platform engineering, CI /CD pipelines are critical enablers of developer productivity and efficient operations.
References:- CNCF Platforms Whitepaper- Continuous Delivery Foundation Guidance- Cloud Native Platform Engineering Study Guide

問題 #27
In a Continuous Integration (CI) pipeline, what is a key benefit of using automated builds?

- A. Ensures consistent builds.
- B. Eliminates coding errors.
- C. Reduces code redundancy.
- D. Minimizes server costs.

答案：A

解題說明：
The key benefit of automated builds in a CI pipeline is ensuring consistent and reproducible builds. Option C is correct because automation eliminates the variability introduced by manual processes, guaranteeing that each build follows the same steps, uses the same dependencies, and produces artifacts that are predictable and testable.
Option A (minimizing server costs) may be a side effect but is not the primary advantage. Option B (eliminates coding errors) is inaccurate-automated builds do not prevent developers from writing faulty code; instead, they surface errors earlier. Option D (reduces code redundancy) relates more to code design than CI pipelines.
Automated builds are fundamental to DevOps and platform engineering because they establish reliability in the software supply chain, integrate seamlessly with automated testing, and enable continuous delivery. This practice ensures that code changes are validated quickly, improving developer productivity and reducing integration risks.
References:- CNCF Platforms Whitepaper- Continuous Delivery Foundation Best Practices- Cloud Native Platform Engineering Study Guide

問題 #28
In the context of platform engineering and the effective delivery of platform software, which of the following statements describes the role of CI/CD pipelines in relation to Software Bill of Materials (SBOM) and security scanning?

- A. CI/CD pipelines should integrate SBOM generation and security scanning as automated steps within the build and test phases to ensure early detection of vulnerabilities and maintain a clear inventory of components.
- B. CI/CD pipelines are designed to accelerate the delivery of platform software, and adding SBOM generation and security scanning would slow down the process, so these activities are better suited for periodic audits conducted outside of the pipeline.
- C. CI/CD pipelines are primarily for automating deployments; SBOM generation and security scanning are separate, manual processes performed after deployment.
- D. SBOM generation and security scanning are particularly valuable for application software. While platform software may have different security considerations, these practices are highly beneficial within CI/CD pipelines for applications.

答案：A

解題說明：
Modern platform engineering requires security and compliance to be integral parts of the delivery process, not afterthoughts. CI/CD pipelines are the foundation for delivering platform software rapidly and reliably, and integrating SBOM generation and automated vulnerability scanning directly within pipelines ensures that risks are identified early in the lifecycle.
Option B is correct because it reflects recommended practices from cloud native platform engineering standards: SBOMs provide a transparent inventory of all software components, including dependencies, which is crucial for vulnerability management, license compliance, and supply chain security. By automating these steps in CI/CD, teams can maintain both velocity and security without manual overhead.
Option A downplays the relevance of SBOMs for platform software, which is inaccurate because platform components (like Kubernetes operators, ingress controllers, or logging agents) are equally susceptible to vulnerabilities. Option C dismisses automation in favor of periodic audits, which contradicts the shift-left security principle. Option D misunderstands CI/CD's purpose: security must be integrated, not separated.
References:- CNCF Supply Chain Security Whitepaper- CNCF Platforms Whitepaper- Cloud Native Platform Engineering Study Guide

問題 #29
Which approach is effective for scalable Kubernetes infrastructure provisioning?

- A. Crossplane compositions defining custom CRDs
- B. Imperative scripts using Kubernetes API
- C. Helm charts with the environment values.yaml
- D. Static YAML with kubectl apply

答案：A

解題說明：
The most effective approach for scalable Kubernetes infrastructure provisioning is Crossplane compositions.
Option D is correct because compositions let platform teams define custom CRDs (Composite Resources) that abstract infrastructure details while embedding organizational policies and guardrails. Developers then consume these abstractions through simple Kubernetes-native APIs, enabling self-service at scale.
Option A (Helm with values.yaml) is useful for application deployment but not for scalable infrastructure provisioning across multiple clouds. Option B (imperative scripts) lacks scalability, repeatability, and governance. Option C (static YAML with kubectl apply) is manual and not suited for dynamic, multi-team environments.
Crossplane compositions allow platform teams to curate golden paths while giving developers autonomy. This reduces complexity, ensures compliance, and supports multi-cloud provisioning-all key aspects of platform engineering.
References:- CNCF Crossplane Project Documentation- CNCF Platforms Whitepaper- Cloud Native Platform Engineering Study Guide

問題 #30
.....

人們相信需要一個標準化的、多國的、令人信服的考試來驗證個人在 Linux Foundation上技能的等級。同時，這個考試必須有利於公司雇用 Linux Foundation 方面專業人才。為了實現這壹目的，Linux Foundation 專家機構聯合多方力量設計和完善了 CNPA 認證考試。Linux Foundation 專家機構通過全球的發展使之成為一個倍受公認和廣泛認可的 CNPA 認證考試體系。用戶應該可以自由選擇，在認證 Linux Foundation 最高級工程師這壹關鍵領域不應固定於一個廠商。

2026 KaoGuTi最新的CNPA PDF版考試題庫和CNPA考試問題和答案免費分享：https://drive.google.com/open?id=1ZjOOYy32KaAepuUl0JRIMZE6m9W3Pf6P