1 CNPA

Generated on: 2026-01-13 15:05:01 **Version:** 1.0

1.1 Table of Contents

1.2 Overview

The **Cloud Native Platform Associate (CNPA)** exam demonstrates knowledge of platform engineering concepts and cloud native technologies.

1.3 Exam Overview

Detail	Information
Exam Format	Multiple Choice
Number of Questions	60
Duration	90 minutes
Passing Score	75%
Certification Validity	3 years
Cost	$250 USD
Retake Policy	1 free retake

1.4 Exam Domains & Weights

Domain	Weight
Platform Engineering Fundamentals	20%
Developer Experience	20%
Infrastructure and Operations	25%
Security and Compliance	20%
Observability and Monitoring	15%

1.5 Key Topics

1.5.1 Platform Engineering

- Internal Developer Platforms (IDP)
- Self-service capabilities
- Golden paths
- Platform as a Product

1.5.2 Developer Experience

- Developer portals
- Service catalogs
- Templates and scaffolding

- Documentation

1.5.3 Infrastructure

- Infrastructure as Code
- GitOps
- Multi-tenancy
- Resource management

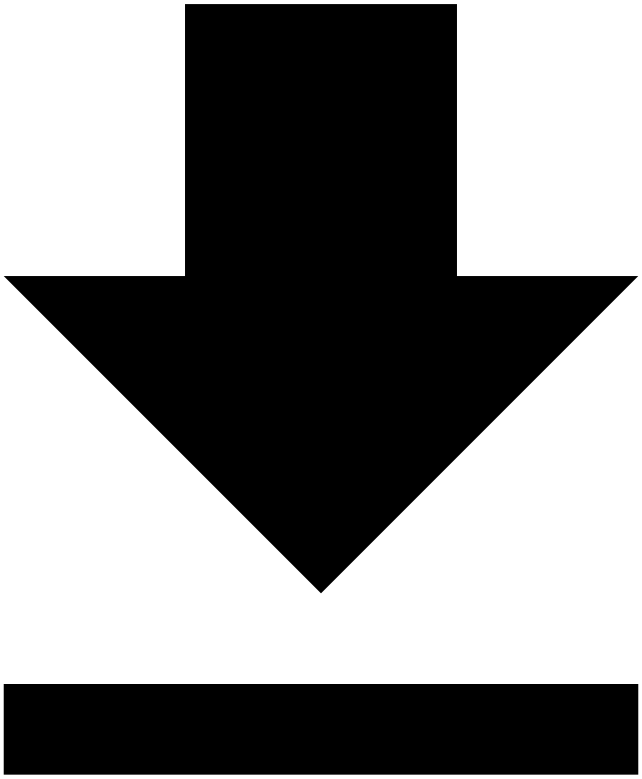
1.6 Study Resources

- Platform Engineering
- CNCF Platforms White Paper
- CNPA Curriculum

1.7 Navigation

- Next: Sample Questions →

1.8 Sample Practice Questions



Download PDF Version

1.9 Practice Resources

- Platform Engineering
- CNCF Platforms White Paper

1.10 Platform Engineering Fundamentals (20%)

1.10.1 Question 1

What is an Internal Developer Platform (IDP)?

Show Solution

An **Internal Developer Platform (IDP)** is a self-service layer that enables developers to: - Deploy applications without deep infrastructure knowledge - Access standardized tooling and workflows - Follow organizational best practices automatically - Reduce cognitive load on development teams

Key components: - Service catalog - Self-service portal - Automated workflows - Golden paths/templates

1.10.2 Question 2

What are “Golden Paths” in platform engineering?

Show Solution

Golden Paths are opinionated, supported ways to build and deploy applications that: - Represent best practices - Are fully supported by the platform team - Reduce decision fatigue for developers - Include security and compliance by default

Example: A golden path for deploying a microservice might include: - Pre-configured CI/CD pipeline - Standard observability setup - Security scanning - Deployment to Kubernetes

1.10.3 Question 3

What is “Platform as a Product”?

Show Solution

Platform as a Product means treating the internal platform like a product: - Developers are customers - Focus on user experience - Gather feedback and iterate - Measure adoption and satisfaction - Have a product roadmap

Key practices: - User research with developers - Documentation and onboarding - Support channels - Regular releases and improvements

---

1.11 Developer Experience (20%)

1.11.1 Question 4

What components make up a good developer portal?

Show Solution

A developer portal typically includes: - **Service Catalog** - List of available services and APIs - **Documentation** - Technical docs, guides, tutorials - **Templates** - Scaffolding for new projects - **Self-Service** - Provisioning resources without tickets - **Search** - Find services, docs, and owners - **Ownership** - Who owns what service

Tools: Backstage, Port, Cortex

1.11.2 Question 5

How do you measure developer experience?

Show Solution

Key metrics: - **DORA Metrics** - Deployment frequency, lead time, MTTR, change failure rate - **Developer Satisfaction** - Surveys, NPS scores - **Time to First Deploy** - How long for new developers to deploy - **Self-Service Adoption** - % of requests handled without tickets - **Cognitive Load** - Number of tools/systems developers must know

1.11.3 Question 6

What is a Service Catalog?

Show Solution

A **Service Catalog** is a centralized inventory of: - All services in the organization - APIs and their documentation - Service ownership and contacts - Dependencies between services - Health and status information

Benefits: - Discoverability - Reduced duplication - Clear ownership - Dependency management

---

1.12 Infrastructure and Operations (25%)

1.12.1 Question 7

How does GitOps relate to platform engineering?

Show Solution

GitOps enables platform engineering by: - **Declarative Infrastructure** - Git as source of truth - **Self-Service** - Developers submit PRs for changes - **Audit Trail** - All changes tracked in Git history - **Automation** - Changes automatically applied - **Rollback** - Easy revert to previous state

Tools: Argo CD, Flux CD

1.12.2 Question 8

What is multi-tenancy in platform context?

Show Solution

Multi-tenancy allows multiple teams to share platform resources:

Isolation levels: - **Namespace-based** - Separate namespaces per team - **Cluster-based** - Separate clusters per team - **Virtual clusters** - vCluster for isolation

Considerations: - Resource quotas - Network policies - RBAC - Cost allocation

1.12.3 Question 9

What Infrastructure as Code tools are commonly used?

Show Solution

Common IaC tools: - **Terraform** - Multi-cloud infrastructure - **Pulumi** - IaC with programming languages - **Crossplane** - Kubernetes-native IaC - **AWS CDK** - AWS infrastructure - **Ansible** - Configuration management

Platform teams often provide: - Terraform modules - Crossplane compositions - Pre-approved configurations

---

1.13 Security and Compliance (20%)

1.13.1 Question 10

How do platforms enforce security policies?

Show Solution

Security enforcement methods: - **Policy as Code** - OPA/Gatekeeper, Kyverno - **Admission Controllers** - Validate/mutate resources - **Supply Chain Security** - Image signing, SBOM - **Network Policies** - Default deny, microsegmentation - **RBAC** - Least privilege access

Platforms embed security in golden paths so developers get security by default.

1.13.2 Question 11

What is shift-left security?

Show Solution

Shift-left security moves security earlier in the development lifecycle:

- **Design** - Threat modeling
- **Code** - SAST, secrets scanning
- **Build** - Image scanning, SBOM
- **Deploy** - Policy enforcement
- **Runtime** - Monitoring, detection

Platform teams enable shift-left by: - Integrating security tools in CI/CD - Providing secure templates - Automating compliance checks

---

1.14 Observability and Monitoring (15%)

1.14.1 Question 12

What observability capabilities should a platform provide?

Show Solution

Platform observability includes: - **Metrics** - Prometheus, Grafana dashboards - **Logs** - Centralized logging (Loki, ELK) - **Traces** - Distributed tracing (Jaeger, Tempo) - **Alerts** - Alertmanager, PagerDuty integration - **Dashboards** - Pre-built for common patterns

Platforms should provide: - Auto-instrumentation - Standard dashboards - Alert templates - SLO tracking

1.14.2 Question 13

How do you implement SLOs in a platform?

Show Solution

SLO implementation: 1. **Define SLIs** - Latency, availability, error rate 2. **Set SLO targets** - 99.9% availability 3. **Calculate error budgets** - Allowed downtime 4. **Monitor and alert** - Track against targets 5. **Report** - Dashboards and reviews

Tools: - Prometheus + recording rules - Sloth for SLO generation - OpenSLO specification

---

1.15 Exam Tips

1. **Understand IDP concepts** - Self-service, golden paths, platform as product
2. **Know developer experience metrics** - DORA, satisfaction, adoption
3. **Understand GitOps** - How it enables platform capabilities
4. **Know policy enforcement** - OPA, Kyverno, admission controllers
5. **Understand observability** - Three pillars, SLOs, error budgets

---

← Back to CNPA Overview
