

更に上のクオリティ
更に上のサービス!

問題集

ITEXAMPASS

<https://www.itexampass.jp>



1年で無料進級することに提供する

Exam : KCNA

**Title : Kubernetes and Cloud
Native Associate (KCNA)**

Version : DEMO

1.An application that is nearing its usage limit. To increase the amount of users it can handle, you allocate additional memory resources to each instance of the application.

What type of scaling is this?

- A. Horizontal Scaling
- B. Cluster Autoscaling
- C. Recursive Scaling
- D. Vertical Scaling

Answer: D

Explanation:



Vertical Scaling

(Scaling up)

Graphical user interface, diagram

Description automatically generated

Horizontal Scaling

(Scaling out)

2.Which of the following best describes a cloud-native app?

- A. An application where all logic is coded into a single large binary.
- B. An application that publishes an HTTPS web front-end.
- C. An application that takes advantages of cloud computing frameworks and their loosely coupled cloud services.
- D. An application that leverages services that are native to public cloud platforms such as Azure, GCP, and/or AWS.

Answer: C

Explanation:

Cloud-native apps leverage cloud computing frameworks and tend to be microservices based, where individual components of the app are coded as individual.

3.Which project in this list is a leading project in the observability space?

- A. Jaeger

- B. Vitess
- C. Argo
- D. Kubernetes

Answer: A

Explanation:

<https://github.com/cncf/landscape#trail-map>



CLOUD NATIVE TRAIL MAP

The Cloud Native Landscape ([cncf.io](https://github.com/cncf/landscape)) has a large number of options. This Cloud Native Trail Map is a recommended process for leveraging open source, cloud native technologies. At each step, you can choose a vendor-supported offering or do it yourself, and everything after step #3 is optional based on your circumstances.

HELP ALONG THE WAY

A. Training and Certification

Consider training offerings from CNCF and then take the exam to become a Certified Kubernetes Administrator or a Certified Kubernetes Application Developer cncf.io/training

B. Consulting Help

If you want assistance with Kubernetes and the surrounding ecosystem, consider leveraging a Kubernetes Certified Service Provider cncf.io/kcsp

C. Join CNCF's End User Community

For companies that don't offer cloud native services externally cncf.io/enduser

WHAT IS CLOUD NATIVE?

Cloud native technologies empower organizations to build and run scalable applications in modern, dynamic environments such as public, private, and hybrid clouds. Containers, service meshes, microservices, immutable infrastructure, and declarative APIs exemplify this approach.

These techniques enable loosely coupled systems that are resilient, manageable, and observable. Combined with robust automation, they allow engineers to make high-impact changes frequently and predictably with minimal toil.

The Cloud Native Computing Foundation seeks to drive adoption of this paradigm by fostering and sustaining an ecosystem of open source, vendor-neutral projects. We democratize state-of-the-art patterns to make these innovations accessible for everyone.

cncf.io

v20200501



1. CONTAINERIZATION

- Commonly done with Docker containers
- Any size application and dependencies (even PDP-11 code running on an emulator) can be containerized
- Over time, you should aspire towards splitting suitable applications and writing future functionality as microservices

2. CI/CD

- Setup Continuous Integration/Continuous Delivery (CI/CD) so that changes to your source code automatically result in a new container being built, tested, and deployed to staging and eventually, perhaps, to production
- Setup automated rollouts, roll backs and testing
- Argo is a set of Kubernetes-native tools for deploying and running jobs, applications, workflows, and events using GitOps paradigms such as continuous and progressive delivery and MLOps

3. ORCHESTRATION & APPLICATION DEFINITION

- Kubernetes is the market-leading orchestration solution
- You should select a Certified Kubernetes Distribution, Hosted Platform, or Installer: cncf.io/ck
- Helm Charts help you define, install, and upgrade even the most complex Kubernetes application

4. OBSERVABILITY & ANALYSIS

- Pick solutions for monitoring, logging and tracing
- Consider CNCF projects Prometheus for monitoring, Fluentd for logging and Jaeger for Tracing
- For tracing, look for an OpenTracing-compatible implementation like Jaeger

5. SERVICE PROXY, DISCOVERY, & MESH

- CoreDNS is a fast and flexible tool that is useful for service discovery
- Envoy and Linkerd each enable service mesh architectures
- They offer health checking, routing, and load balancing

6. NETWORKING, POLICY, & SECURITY

To enable more flexible networking, use a CNI-compliant network project like Calico, Flannel, or Weave Net. Open Policy Agent (OPA) is a general purpose policy engine with uses ranging from authorization and admission control to data filtering. Falco is an anomaly detection engine for cloud native.

7. DISTRIBUTED DATABASE & STORAGE

When you need more resiliency and scalability than you can get from a single database, Vitess is a good option for running MySQL at scale through sharding. Rook is a storage orchestrator that integrates a diverse set of storage solutions into Kubernetes. Serving as the "brain" of Kubernetes, etcd provides a reliable way to store data across a cluster of machines. TiKV is a high performance, distributed transactional key-value store written in Rust.

8. STREAMING & MESSAGING

When you need higher performance than JSON-RPC, consider using gRPC or NATS. gRPC is a universal RPC framework. NATS is a multi-modal messaging system that includes request/reply, pub/sub and load balanced queues. CloudEvents is a specification for describing event data in common ways.

9. CONTAINER REGISTRY & RUNTIME

Harbor is a registry that stores, signs, and scans content. You can use alternative container runtimes. The most common, both of which are OCI-compliant, are containerd and CRI-O.

10. SOFTWARE DISTRIBUTION

If you need to do secure software distribution, evaluate Notary, an implementation of The Update Framework.

4.To specify a Kubernetes object which language is used?

- A. JSON
- B. Go
- C. YAML
- D. Node
- E. Python

Answer: C

Explanation:

<https://kubernetes.io/docs/concepts/overview/working-with-objects/kubernetes-objects/>

Understanding Kubernetes Objects

This page explains how Kubernetes objects are represented in the Kubernetes API, and how you can express them in `.yaml` format.

Graphical user interface, text

Description automatically generated

5.What is OPA?

- A. Open Permission Agent
- B. Online Policy Audit
- C. Open Policy Agent
- D. Offline Policy Accessor

Answer: C

Explanation:

<https://www.cncf.io/projects/open-policy-agent-opa/>



About

Projects

Training

Community

Blog & News

Join

PROJECTS

Open Policy Agent (OPA)



Open Policy Agent

An open source, general-purpose policy engine.

Open Policy Agent (OPA) was accepted to CNCF on **March 29, 2018** and is at the **Graduated** project maturity level.

Graphical user interface, text, application, email

Description automatically generated