

1. Donner la définition de cloud native

- Le Cloud Native décrit une **approche de développement logiciel** dans laquelle les applications sont dès le début conçues pour une utilisation sur le Cloud.
- Le Cloud native se réfère à **une approche de développement et de déploiement de logiciels** exploitant pleinement les avantages du cloud. Cela implique **la décomposition en microservices**, l'utilisation de **conteneurs pour la portabilité et l'élasticité**, ainsi que **l'automatisation** des tâches opérationnelles. Les services cloud natifs sont également utilisés pour optimiser les fonctionnalités. → Les caractéristiques

2. Donner les 4 caractéristiques de cloud computing

- Le service doit être en libre-service à la demande
- Le service doit être mesurable (mesure et affichage de paramètres de consommation)
- Il doit y avoir une mutualisation des ressources
- Il doit être rapidement élastique (adaptation rapide à une variation du besoin)

3. Donner les 4 caractéristiques de cloud native

Microservices - Conteneurisation - Élasticité - évolutivité

1. Décomposition en microservices.
2. Conteneurisation.
3. Évolutivité dynamique.
4. Automatisation et orchestration.

4. Citez les avantages du cloud native

Scalabilité élastique - Haute disponibilité - Déploiement plus rapide - Coût réduit (une optimisation des coûts)

Évolutivité, flexible, une portabilité, une résilience, une agilité

5. Donner la définition de l'approche cloud native

L'approche cloud native est une méthode de développement, de déploiement et de gestion d'applications qui est spécifiquement conçue pour fonctionner de manière optimale dans un **environnement de cloud computing**.

6. Donner les 4 caractéristiques de l'approche cloud native

L'approche cloud native se caractérise par **la décomposition en microservices**, **la conteneurisation**, **l'évolutivité horizontale** et **l'automatisation/orchestration**, ce qui permet d'obtenir une architecture flexible.

7. Citez les avantages de l'approche cloud native

Flexibilité, évolutivité, automatisation réduite

8. A quoi se concentre ou se base l'approche de cloud native ?

Cette approche se concentre sur le **développement d'applications sous la forme de microservices individuels**, qui ne sont pas exécutés « On-Premises » (localement), mais sur **des plateformes agiles basées sur des conteneurs**.

9. L'approche de cloud native repose sur 4 piliers lesquels ? et comment ils sont ?

4 piliers sont : les architectures microservices, les technologies de conteneurs, le processus de développement(devops) et la livraison contenu (continuous delivery)

L'approche Cloud Native repose sur **quatre piliers** qui sont **liés et interdépendants**.



10. Citer les différents services de cloud native

IAAS (Infrastructure as a service) – PAAS(platform as a service) – SAAS(software as a service)

11. Citer les différents modèles de cloud native

Cloud public – cloud privé – cloud hybride

12. Donner 4 fournisseurs cloud

Amazon – google cloud – Microsoft

13. Donner des exemples des fournisseurs de services d'IAAS

Elastic ComputeCloud (EC2) → **AWS** - Compute Engine → **GOOLE CLOUD** - Virtual Machine → **AZURE**

14. Donner des exemples des fournisseurs de services de PAAS

AWS Elastic Beanstalk → **AWS** - Google App Engine → **GOOLE CLOUD** - Azure App Service - azure function App → **AZURE**

15. Donner des exemples des fournisseurs de services de SAAS

Zoom → **AWS** - Google Apps → **GOOLE CLOUD** - Microsoft Office 365 → **AZURE**

16. Définir le conteneur

La **conteneurisation**, est un type de virtualisation, qui consiste à rassembler le code du logiciel et tous ses composants (bibliothèques, frameworks et autres dépendances) de manière à les isoler dans leur propre « **conteneur** » ;

17. Quelle est la différence entre la conteneurisation et la machine virtuelle

- Performance

Chaque machine virtuelle dispose de son propre système d'exploitation

Les applications conteneurisées **partagent** un environnement de système d'exploitation (noyau),

Les applications conteneurisées utilisent donc **moins de ressources** que des machines virtuelles complètes et **réduisent la pression sur la mémoire de l'hôte.**

- Légèreté

Les machines virtuelles traditionnelles peuvent occuper beaucoup d'espace disque

Les conteneurs sont **relativement légers** : ils ne contiennent que les bibliothèques et les outils nécessaires à l'exécution de l'application conteneurisée

Les conteneurs sont donc **plus compacts** que les machines virtuelles et **démarrent plus rapidement**

18. Définition une application monolithique (on-Promise ou locale)

Une application **monolithique** est une application qui est développée en un **seul bloc** (war, jar, Ear, dl ...), avec une même technologie et déployée dans un serveur d'application.

19. Définition le microservice

Les microservices désignent à la fois **une architecture** et **une approche de développement logiciel** qui consiste à décomposer les applications en éléments les plus simples, indépendants les uns des autres.

20. Quelle est la différence entre une approche monolithique et le microservice

Une application monolithique est **une seule unité unifiée**, une architecture microservices la décompose en un ensemble de **petites unités indépendantes**.

Contrairement à une approche monolithique classique, selon laquelle tous les composants forment une entité indissociable, les microservices fonctionnent en synergie pour accomplir les mêmes tâches, tout en étant séparés.

21. Donner la définition API (Application Programming Interface)

Une API (interface de programmation d'application) est un ensemble de définitions et de protocoles qui facilite la création et l'intégration de logiciels d'applications.

22. Citer 3 avantages des APIs

- Pouvoir interagir avec un système sans se soucier de sa complexité et de son fonctionnement
- Souvent spécialisée dans un domaine et sur un use case particulier
- Constituent un moyen simplifié de connecter votre propre infrastructure au travers du développement d'applications cloud-native.

23. Définir l'API REST

REST (Representational State Transfer) comme un **style architectural** et une **méthodologie** fréquemment **utilisés** dans **le développement de services Internet**, tels que les **systèmes hypermédias distribués**

La forme complète de l'API REST (Application Programing Interface Representational State Transfer) est l'**interface de programmation d'applications de transfert d'état représentationnelle**, plus communément appelée service Web API REST.

24. Définir la méthode POST

La méthode POST envoie les données à traiter à une source spécifiée

25. Citer deux avantages des requêtes de la méthode POST

- Ne sont jamais paramétrées,
- Ne restent pas dans l'historique du navigateur et nous ne pouvons pas les mettre en signet.
- N'ont aucune restriction de longueur de données.

26. Définir la méthode GET

La méthode GET de HTTP demande des données d'une source spécifiée

27. Citer deux avantages des demandes de la méthode GET

- Être mises en cache
- Rester dans l'historique du navigateur.
- être marqué.

28. Citer la différence entre require et import

La directive "require" indique à JavaScript **d'importer la totalité du module** demandée, par contre le directif "import" permet de n'importer qu'une **partie spécifique d'un module**.

29. Donner la définition de middleware

Le middleware est une fonction qui a accès au nécessaire, réponse objet et nouvelle fonction dans le cycle demande-réponse

30. Quels sont les avantages des conteneurs (diapo 203)

- **Portabilité** : isolé par rapport au système d'exploitation hôte => s'exécuter de manière cohérente et uniforme sur n'importe quelle plate-forme ou cloud
- **Vitesse** : légers => réduit également le temps de lancement
- **Isolation** : Les équipes de développement peuvent rapidement identifier et corriger les problèmes techniques d'un conteneur défectueux sans provoquer l'arrêt du reste des conteneurs
- **Facilité de la gestion** : L'orchestration des conteneurs permet de faciliter les tâches de gestion, comme le déploiement de nouvelles versions d'applications ou l'évolution d'applications conteneurisées ...

31. Définir devOps

L'adoption des méthodes Agiles qui va permettre d'intégrer les équipes métier dans cette collaboration avec les équipes techniques et de développement. L'idée est de collaborer pour délivrer une itération en améliorant le produit à chaque livraison de façon continue.

