Etat de l'art : solutions d'orchestration

Référence : **LUSY190** Durée : **0 jours** Certification : **Aucune**

CONNAISSANCES PREALABLES

- Connaître la terminologie et les concepts des architectures informatiques.

PROFIL DES STAGIAIRES

- Architectes, Responsables des infrastructures IT, Chefs de projet, Administrateurs système et/ou réseau, Développeurs....

OBJECTIFS

- Comprendre le fonctionnement des solutions d'orchestration de conteneurs et de leur écosystème pour la mise en oeuvre de plateformes de type CaaS (Container as a Service).

CERTIFICATION PREPAREE

Aucune

METHODES PEDAGOGIQUES

- Mise à disposition d'un poste de travail par stagiaire
- Remise d'une documentation pédagogique numérique pendant le stage
- La formation est constituée d'apports théoriques, d'exercices pratiques, de réflexions et de retours d'expérience
- Le suivi de cette formation donne lieu à la signature d'une feuille d'émargement

FORMATEUR

Consultant-Formateur expert Production et Supervision

METHODE D'EVALUATION DES ACQUIS

- Auto-évaluation des acquis par le stagiaire via un questionnaire
- Attestation des compétences acquises envoyée au stagiaire
- Attestation de fin de stage adressée avec la facture

CONTENU DU COURS

Comprendre les principes fondamentaux de containerisation et du modèle CaaS

- Le besoin : gestion de conteneurs en nombre
- Provisionning et placement des conteneurs
- Monitoring, gestion du failover des conteneurs et la scalabilité
- Gestion des mises à jour
- Contraintes d'une infrastructure de production
- Le modèle CaaS
- Normalisation : OCI, CNCF, CNI, CSI, CRI

Identifier les acteurs majeurs et les usages actuels

- Tour d'horizon des solutions techniques : Kubernetes, Docker Swarm, AWS ECS, AWS ESB, AWS CloudMap
- Exemple d'atelier : démonstration sur AWS

Comprendre la technologie de containerisation et son écosystème

- Les technologies de base : lxc, Docker
- Présentation de lxc : Linux containers, historique, principe de fonctionnement. Les Cgroups.
- L'isolation de ressources, la création d'un environnement utilisateur.
- Positionnement par rapport aux autres solutions de virtualisation.
- Apports de Docker : Docker Engine pour créer et gérer des conteneurs Dockers.
- Plateformes supportées par Docker.
- L'écosystème Docker
- Exemple d'atelier : mise en oeuvre de containeurs Docker

Découvrir le fonctionnement de Kubernetes, orchestrateur de conteneurs

- Présentation Kubernetes, origine du projet
- Fonctionnalités : automatisation des déploiements et de la maintenance des applications en containers, redéploiement, reconnaissance de services, équilibrage de charge, réparation automatique pour la haute disponibilité
- Containers supportés, plateformes utilisant Kubernetes
- Composants de Kubernetes
- Définitions : pods, labels, controllers, services
- L'écosystème Kubernetes : Helm, Ingress, Grafana/Prometheus, Istio, Dashboard
- Distributions et Offre Cloud
- Exemple d'atelier : mise en oeuvre d'une infrastructure Kubernetes avec Helm

Comprendre les interactions avec le Cloud privé/public et le legacy

- Caractéristiques et contraintes des containeurs et de l'interfaçage entre cloud privé/cloud public et le legacy
- Exemple d'atelier : démonstration avec Terraform de déploiements sur une infrastructure complète avec un cloud privé OpenStack, un cloud public AWS et l'infrastructure de serveurs autonomes

Appréhender les principes généraux de sécurité du CaaS, de Kubernetes et de Docker

- Sécurité des technologies de conteneurs
- Analyse des points à risques Docker : le noyau, le service Docker, les containers, ...
- Analyse des types de dangers : déni de service, accès réseau non autorisés, ...
- Mécanismes de protection : pile réseau propre à chaque container, limitations de ressources par les cgroups, restrictions des droits d'accès sur les sockets, politique de sécurité des containers
- Sécurisation des clients par des certificats. Principe, et mise en oeuvre avec openssl. Configuration réseau, sécurité et TLS
- Fiabilité des images déployées dans Docker
- Sécurisation Kubernetes. Accès à l'API Kubernetes. Limitations des ressources. Contrôle des accès réseau
- Restrictions des accès à etcd
- Présentation des bonnes pratiques
- Exemple d'atelier : mise en évidence de failles de sécurité de containers Docker gérés par Kubernetes et des bonnes pratiques à adopter

Identifier les bénéfices et les limites des architectures micro-services en termes techniques et organisationnels

- Apports d'une architecture micro-services, selon les différentes solutions, adéquation des technologies aux différents besoins et risques, limites

*Notre **référent handicap** se tient à votre disposition au 01.71.19.70.30 ou par mail à referent.handicap@edugroupe.com pour recueillir vos éventuels besoins d'aménagements, afin de vous offrir la meilleure expérience possible.*