etude de l'existant:

1-petite intoduction

2-le fonctionement des salles TPs

3-les probémes majeur et les deficultés rencontrés par les profs(selon les reponses au formulaire de sarah)

4-les solutions qui existent déja

5-les probémes de ces solutions

6-la solution proposé

etude theorique:

1-introduction

2-la conteneurisation:

a.definition

Il s’agit d’un type de virtualisation utilisé au niveau des applications.Elle 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 ». Le logiciel ou l'application dans le conteneur peut ainsi être déplacé et exécuté de façon cohérente dans tous les environnements et sur toutes les infrastructures, indépendamment de leur système d'exploitation. Le conteneur fonctionne comme une sorte de bulle, ou comme un environnement de calcul qui enveloppe l'application et l'isole de son entourage. C'est en fait un environnement de calcul portable complet.

b.le principe de la conteneurisation

Le principe repose sur la création de plusieurs espaces utilisateurs isolés les uns des autres sur un noyau commun. On utilise alors le terme de « conteneur » pour désigner une telle instance. Cette séparation repose sur un concept similaire à celui des modules applicatifs cloisonnés, communiquant à l’aide de services et applications web.

c.les avantages de cette technologie

.L’accélération des développements : le développeur travaille dans un cadre restreint à son strict nécessaire ce qui lui épargne le codage de tests d’interactions notamment. Cela favorise aussi la création de bacs à sable et donc une montée en compétence et une capacité d’innovation plus rapides.

.La portabilité et donc l’accélération des déploiements : le conteneur créé est cohérent et ne souffrira pas d’être exécuté sur un autre environnement, tout en étant moins gourmand qu’une VM et peut donc être plus facilement déplacé, copié, relancé.

.L’impact moindre sur les performances du serveur, un conteneur pouvant libérer rapidement les ressources (mémoire, stockage) inutilisées.

Vous gérez un grand nombre d’applications, largement indépendantes, hétérogènes.

Vous cherchez à accélérer le rythme des livraisons de vos applications pour répondre à l’évolution du comportement de vos cibles et à favoriser l’innovation.

Vous cherchez à améliorer l’élasticité de votre architecture, pour la maîtriser ou préparer une forte croissance.

b.les differents utiles de conteneurisation:

La conteneurisation est donc au service de l’agilité, ce qui a été encore favorisé par l’adoption rapide et à grande échelle de solutions comme Docker, de la société française dotCloud, basée sur des technologies open source.Les services informatiques, mais aussi les plateformes Cloud d’Oracle ou Microsoft l’ont adoptée et la communauté Docker Hub met à disposition un grand nombre de conteneurs bac à sable catalysant l’innovation.

D’autres solutions facilitant la conteneurisation existent bien évidemment, sous Linux comme sous Windows, FreeBSD ou Solaris :

LXC (la base historique de la conteneurisation sous Linux),

Rocket (rkt) de CoreOs,

Windows Hyper-V Containers (qui s’apparente à des VM légères),

Docker for Windows,

Oracle Zones,

FreeBSD Jails, etc.

Un deuxième palier a été franchi dans l’accélération de l’adoption des conteneurs à l’ouverture par Google en 2015 de sa solution Kubernetes (K8s), l’orchestrateur largement dominant du marché.

3-kubernete:

a.definition:

Kubernetes (également appelé KS8) est un orchestrateur de containers logiciels open source. Comme son nom l'indique, il est conçu pour orchestrer des architectures de grappe(s) ou cluster(s) de containers logiciels de manière extrêmement agile, en donnant la possibilité de gérer l'allocation des ressources machine sous-jacentes à la volée.

Développé par Google, Kubernetes est la réécriture en Go de Borg, un système de clustering fait maison utilisé par le géant américain depuis des années. Google Search, Maps, Gmail, Youtube, etc. Tous ses services reposent sur des grappes de centaines de containers pilotées par Borg. Via cette brique, Google arrête et redémarre pas moins de 2 milliards de containers chaque semaine.

b.les avantages et les inconvenients de cette technologie:

orchestrer des conteneurs sur plusieurs hôtes ;

optimiser l'utilisation de votre matériel afin de maximiser les ressources requises pour l'exécution de vos applications d'entreprise ;

contrôler et automatiser les déploiements et mises à jour d'applications ;

monter et ajouter des systèmes de stockage pour exécuter des applications avec état ;

mettre à l'échelle des applications conteneurisées et leurs ressources à la volée ;

gérer des services de façon déclarative et garantir ainsi que les applications déployées s'exécutent toujours de la manière dont vous les avez déployées ;

vérifier l'intégrité de vos applications et les réparer automatiquement grâce au placement, au démarrage, à la réplication et à la mise à l'échelle automatiques.

c-Composants de Kubernetes:

.Le Master de Kubernetes

Le rôle du master est de piloter le fonctionnement du cluster Kubernetes. Il exécute un serveur d’API, un ordonnanceur, divers contrôleurs et un système de stockage pour conserver l’état du cluster et la configuration réseau. Pour plus de fiabilité, il est possible de construire un cluster Kubernetes avec plusieurs masters.

.Les Workers de Kubernetes

Dans un clusteur Kubernetes, les workers sont les machines qui exécutent les workloads applicatifs. Il est possible d’avoir un ou plusieurs types de machines différents adaptés aux différents types de charges de travail sur le cluster.

.Les Pods

Les Pods de Kubernetes sont des éléments fondamentaux de Kubernetes. Ils sont composés d’un ou plusieurs conteneurs, d’un réseau et de volume de peristence partagés. Tout comme les conteneurs, les pods doivent être conçus pour être éphémères.

.Les Services

Comme un pod, un service Kubernetes est un objet REST. Un service est à la fois une abstraction qui définit un ensemble logique de pods et une politique d’accès à cet ensemble. Il existe différents types de services : ClusterIP, NodePort, LoadBalancer et ExternalName.

.Le Replication Controler

Les contrôleurs sont là pour gérer le cycle de vie des ressources. Ils connaissent l’état souhaité d’une ressource, surveillent que l’état est conforme aux attentes et ils font les changements nécessaires pour que l’état actuel soit conforme à l’état désiré.

.La base de données etcd

La “base de données” de Kubernetes est etcd. Les informations de configuration, d’exécution et d’état y sont stockées. C’est le seul composant " stateful " du master, il est crucial pour le bon fonctionnement de Kubernetes.Le stockage dans etcd est toujours cohérent. Il fonctionne sur le principe clé-valeur. Le stockage peut être réparti sur plusieurs machines pour en assurer la fiabilité. Il supporte les pannes de machines. etcd est développé par CoreOS.

.Le Scheduler

Kube-scheduler est l’agent qui affecte les conteneurs à des noeuds. Il prend en compte le taux de charge du noeud, sa capacité et les besoins en ressource. C’est un planificateur, qui vérifie la charge non affectée et trouve les noeuds appropriés.

.Le contrôleur d’API

Kubernetes expose une API via un serveur API. Il est possible de communiquer avec l’API en utilisant un client local appelé kubectl. Il est également possible d’écrire votre propre client ou d’utiliser les commandes curl pour contrôler le cluster via l’API.Lorsqu’un conteneur doit être exécuté, c’est le rôle du kube-scheduler de trouver le noeud approprié. Le kube-scheduler reçoit ses commandes via l’API.

d.le principe de kubernetes(une brève description comment fonctionne kubernetes):

Le projet open source Kubernetes prône la vision d'un système d'information consolidé sur un seul meta-cluster piloté via un orchestrateur de containers unique. Une infrastructure pouvant être, dans le même temps, découpée en sous-clusters géographiques distribués sur plusieurs clouds, privés et/ou publics.Le tout dans une logique de ressources informatiques partagées.Kubernetes déploie des clusters de pods Docker (ou nœuds Docker). Après avoir été installé sur un serveur maître, Kubernetes va pouvoir coordonner les composantes de l'application à gérer (serveurs frontaux, couche applicative, base de données) en les distribuant sur plusieurs pods.Dédié à un traitement donné, un micro-service par exemple, un pod est la plus petite unité d'un cluster Kubernetes. Un pod peut embarqué un ou plusieurs containers étroitement couplés. La fonction "Déploiement" de Kubernetes contrôle l'état de santé du pod et son redémarre le cas échéant. On utilise la commande kubectl create pour créer un Déploiement orchestrant un pod.

e-pourquoi kubernetes:

Kubernetes permet de se soustraire d'un cloud unique en ouvrant la possibilité de répliquer et orchestrer des applications quelle que soit la plateforme ou le cloud provider. Dans sa version 1.6, Kubernetes supporte des clusters allant jusqu'à 50 000 machines.