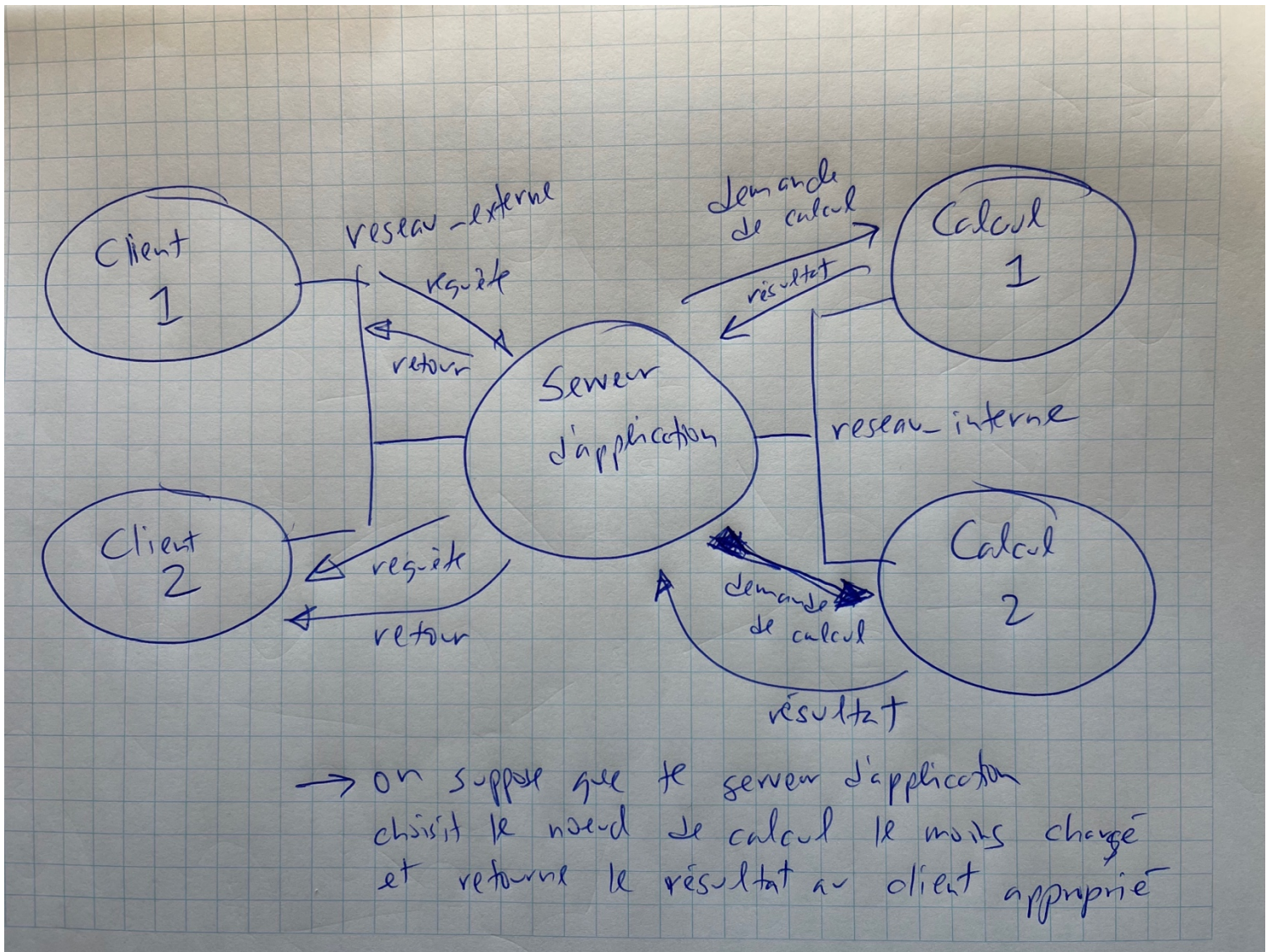


1. Vous êtes consultant pour une entreprise qui se propose de mettre en œuvre un système nuagique destiné à ses clients. On vous demande de proposer un diagramme de déploiement, pour un système qui doit contenir les éléments suivants :
- Deux conteneurs de calcul
 - Un serveur d'application qui est en lien à la fois avec les nœuds de calcul et les clients
 - Deux clients qui accèdent au serveur d'application en utilisant un navigateur web (Firefox, Chrome, Safari, ou autre)



2. Expliquer les avantages et les inconvénients des systèmes monolithiques (centralisés). Pourquoi l'utilisation de technologies nuagiques est maintenant beaucoup plus répandue que les systèmes monolithiques? Existe-t-il des situations où un système monolithique a encore sa place? Si oui, donnez un exemple, si non, expliquez pourquoi.

Avantages : tous les fonctionnalités sont regroupées dans un même processus, qui peut s'exécuter de façon autonome, sans accès réseau

Inconvénients : Développement et maintenance peuvent être difficiles, surtout s'il y a des dépendances entre les fonctionnalités.

Les technologies nuagiques sont maintenant beaucoup plus répandue puisqu'il est courant d'utiliser des systèmes distribués distants.

Une application monolithique est encore appropriée lorsqu'elle est destinée à un usager unique, qui travaillera localement sur son poste. Par exemple, un traitement de texte, un chiffrier, une application d'édition de photos, etc.

3. Lorsque deux conteneurs communiquent, ils peuvent le faire de façon bloquante ou non-bloquante. Expliquez la différence entre les deux. Quel type d'interaction est préférable? Expliquez votre réponse.

Communication bloquante : le système qui initie l'appel va attendre la réponse du système distant

Communication non-bloquante : le système qui initie l'appel va continuer d'exécuter du code pendant que le système distant prépare et envoie la réponse.

La communication non-bloquante est préférable lorsqu'on veut utiliser au mieux les ressources informatiques. Cependant, il y a un coût à ce choix, il faut s'assurer que le client saura que le serveur a répondu. Il faut donc prévoir soit de faire du « polling » (vérifier périodiquement pour voir si la réponse a été reçue), ou bien utiliser le patron « observateur » (on établit avec le serveur une méthode pour que ce dernier démarre un service de traitement du client lorsque la réponse est envoyée.

4. Dans les technologies de conteneurisation, expliquez le lien entre les « repositories » d'images et l'existence et l'utilisation des systèmes de fichiers à union (union file systems).

Les systèmes de fichiers à union permettent de réutiliser sans danger les éléments communs d'images bâties utilisant des images sous-jacentes identiques.

5. Est-il possible de créer un conteneur à partir d'une image? Est-il possible de créer une image à partir d'un conteneur? Expliquez votre réponse.

Oui, tous les conteneurs sont toujours démarrés à partir d'une image. On peut aussi créer une image à partir d'un conteneur, en utilisant la commande « docker commit ».

Cette façon de procéder pour obtenir une image est intéressante lors du développement d'une nouvelle image, mais lorsque le tout est stable, il est préférable d'utiliser un « Dockerfile », pour permettre l'automatisation et faciliter la modification ultérieure de l'image.

6. Quelle est la différence entre docker et docker-compose?

Docker permet de déployer un conteneur unique. On peut, en y mettant les efforts, créer plusieurs conteneurs, créer un réseau particulier, et « brancher » les conteneurs au réseau, mais pour l'utilisation de plusieurs conteneurs qui interagissent entre eux, c'est un peu fastidieux.

Docker-compose permet de démarrer plusieurs conteneurs qui interagissent, de façon automatisée (à l'aide d'un fichier yml approprié). Le tout est beaucoup plus simple que le déploiement de multiples conteneurs uniquement à l'aide de docker.

7. Expliquer ce que veut dire « l'orchestration » de conteneurs.

L'orchestration, c'est le déploiement de multiples conteneurs, selon des règles de démarrage et d'utilisation bien définies. C'est ce que permettent docker-compose et kubernetes, le premier sur une machine unique, le second sur un ensemble de machines distinctes.

8. Expliquez l'utilité des volumes docker.

Les volumes docker permettent à des conteneurs d'utiliser des données persistantes, de façon efficace.

9. Charlemagne tente d'effacer une image docker, mais le système l'en empêche. Pouvez-vous expliquer la cause la plus probable de cette situation?

Si un conteneur est actif (même s'il est arrêté, mais non terminé), il n'est pas possible d'effacer l'image qu'il utilise.

