



Mini-projet : Environnement de développement web avec Docker sous Linux

Énoncé : Création d'un environnement de développement des applications web (par exemple PHP/Laravel et Angular) collaboratif intégrant les outils DevOps (Docker, GitHub et Jira)

Problématique : Dans le cadre d'un développement collaboratif d'une application (équipe de développement au sein d'une entreprise), chaque membre de l'équipe dispose d'un environnement et d'une configuration de machine différents, notamment: les Systèmes d'exploitation hétérogène (linux, windows et Mac) et les versions des API Front-end et Back-end différentes. Par exemple, un membre utilise PHP6/Laravel 6, l'autre PHP7/Laravel 7, pareil pour le front-end, un membre utilise Angular 11 l'autre Angular9 etc. Au moment de fusion des tâches (intégration du projet) il est difficile de le faire fonctionner, de plus, cette hétérogénéité peut entraîner des incohérences dans les résultats, des conflits de dépendances et des difficultés **lors du déploiement** (le serveur de déploiement peut aussi avoir sa propre configuration (Par exemple une Version de PHP/Laravel et de Angular différente de celle utilisée par les membres de développement). Pour résoudre cette problématique, l'utilisation de Docker peut offrir une solution efficace en standardisant les environnements de développement.

Objectif : L'objectif de ce projet est de créer un environnement de développement des applications web avec PHP/Laravel et Angular de manière collaborative en utilisant Docker pour garantir la cohérence des environnements de développement entre tous les membres de l'équipe et les outils de développement modernes et de pratiquer l'intégration de différentes technologies.

Pour cela, vous serez amenés à installer et configurer Docker, créer une image Docker pour un environnement de développement PHP Laravel et Angular, partager cette image avec tous les membres de votre équipe en utilisant Docker Hub avant de commencer le développement de l'application. Ensuite il faut configurer GitHub pour le versionnement du code projet, et configurer Jira pour la gestion des tâches et la répartition des travaux.

Pour l'application web, choisissez n'importe quel sujet, une fois le travail est terminé et validé par le chef d'équipe, l'application doit être déployée sur un serveur Linux contenant la même image Docker.

A- Pourquoi docker ? : L'utilisation de Docker dans le cadre d'un développement collaboratif présente plusieurs avantages importants :

- **Uniformité des Environnements de Développement :** Docker permet de créer des conteneurs légers et portables contenant toutes les dépendances et configurations nécessaires pour exécuter une application. Cela garantit que tous les membres de l'équipe travaillent dans des environnements de développement cohérents, évitant ainsi les problèmes liés aux différences entre les configurations individuelles des machines.
- **Facilité de Déploiement et de Test :** Les conteneurs Docker peuvent être facilement déployés sur n'importe quelle plateforme prenant en charge Docker, ce qui simplifie le



processus de déploiement et réduit les erreurs introduites lors du passage entre les environnements de développement, de test et de production.

- **Gestion des Versions et Reproductibilité** : Les images Docker peuvent être versionnées et partagées via des registres Docker, tels que Docker Hub, ce qui permet de garantir la reproductibilité des environnements de développement. Les développeurs peuvent travailler sur la même version de l'application en utilisant les mêmes images Docker, ce qui facilite la collaboration et la résolution des problèmes.

B- Pourquoi Jira ?

Jira est un outil de gestion de projet largement utilisé dans l'industrie du développement logiciel pour plusieurs raisons importantes :

- **Gestion des Tâches** : Jira offre une plateforme centralisée pour créer, suivre et gérer les tâches et les problèmes liés au développement d'un projet. Cela permet à l'équipe de rester organisée et de prioriser efficacement le travail.
- **Visibilité et Transparence** : Jira fournit des tableaux de bord et des rapports qui offrent une vue d'ensemble du progrès du projet. Cela permet à l'équipe et aux parties prenantes de suivre facilement les avancements, les retards éventuels et les problèmes rencontrés.
- **Collaboration** : Jira facilite la collaboration au sein de l'équipe en permettant aux membres de commenter, de discuter et de travailler ensemble sur les tâches et les problèmes. Les fonctionnalités de notification assurent que les membres de l'équipe sont informés des mises à jour importantes.
- **Intégration avec d'autres Outils** : Jira s'intègre facilement avec de nombreux autres outils de développement logiciel populaires tels que Git, GitHub, Bitbucket, Confluence, etc. Cela permet à l'équipe de centraliser ses outils et de travailler de manière cohérente.

C- Pourquoi GitHub ?

GitHub est un choix populaire pour le développement collaboratif d'applications pour plusieurs raisons importantes :

- **Gestion du Code Source** : GitHub offre une plateforme robuste pour héberger et gérer le code source des projets logiciels. Il utilise le système de contrôle de version Git, ce qui permet de suivre l'évolution du code au fil du temps, de gérer les branches et les fusions, et de revenir à des versions antérieures si nécessaire.
- **Collaboration Facilitée** : GitHub facilite la collaboration entre les membres de l'équipe grâce à des fonctionnalités telles que les demandes de tirage (pull requests), les commentaires sur le code, la révision de code et la gestion des autorisations. Cela permet aux développeurs de travailler ensemble de manière efficace, même s'ils sont géographiquement dispersés.
- **Intégration Continue** : GitHub offre des fonctionnalités d'intégration continue (CI) grâce à des outils comme **GitHub Actions** ou l'intégration avec des services externes tels que



Travis CI, CircleCI, etc. Cela permet d'automatiser les tests, les déploiements et d'autres processus pour garantir la qualité du code.

Travail à Faire :

1) Choisissez un sujet de votre application Web et répartir les tâches:

Il vaut mieux choisir un sujet simple et énumérer ses différentes fonctionnalités.

- Désigner un chef de projet : celui-ci doit configurer Jira et répartir les tâches et les valides après révision de code, organiser des réunions pour simuler une entreprise réelle. Intervenir en cas de problèmes.
- Désigner un membre responsable de développement
- Désigner un membre responsable de GitHub/ et de déploiement automatique et Tests Unitaires/intégration et fonctionnels
- Désigner un membre responsable de la configuration d'environnement (Image Docker et BDD) .
- **de plus**, tous les membres doivent participer au développement.

2) Installation et Configuration de Docker sous Linux :

- Installez Docker sur votre système d'exploitation de préférence sous Linux.
- Configurez Docker selon vos besoins
- Création de l'Image Docker pour le Développement PHP/ Laravel, Apache/nginx, Mysql et Angular
- Créez un fichier Dockerfile pour configurer votre environnement de développement avec PHP Laravel et Angular et toutes les bibliothèques nécessaires.
- Assurez-vous que l'image Docker contient tous les outils et dépendances nécessaires pour le développement avec ces technologies.
- Créez un compte sur Docker Hub si vous n'en avez pas déjà un et Partagez de l'Image Docker sur Docker Hub
- Assurez-vous que l'image est partagée avec les membres de votre équipe.
- Tous les membres de l'équipe doivent utiliser la même image docker publiée sur le HUB.

3) Configuration de GitHub pour le Versionnement du Code Projet, chaque membre de l'équipe doit :

- Créez un nouveau dépôt GitHub pour votre projet.
- Configurez Git sur votre machine et liez-le à votre compte GitHub
- Publiez le code de votre projet sur GitHub en suivant les bonnes pratiques de versionnement.

4) Configuration de Jira pour la Gestion des Tâches :

- Créez un compte Jira si vous n'en avez pas déjà un.
- Configurez un projet dans Jira pour votre équipe de développement.
- Définissez les tâches et les sprints nécessaires pour le développement de votre projet.

Livrables :



- Rapport technique : Documentez chaque étape de votre projet, en incluant les commandes utilisées, les configurations effectuées, et les problèmes rencontrés avec leurs solutions.
- Rapport du projet (Conception, tâches, image docker, capture application , etc.)
- Présentation PPT
- Lien du code source de votre projet est correctement publié sur GitHub et l'image Docker partagée sur Docker Hub.

Critères d'Évaluation :

- Complétion réussie de chaque tâche selon les exigences spécifiées.
- Documentation claire et complète de chaque étape du processus
- Présentation propre et organisation du code sur GitHub.
- Configuration fonctionnelle de l'environnement de développement Docker.
- Configuration appropriée de Jira pour la gestion des tâches.
- Partage réussi de l'image Docker sur Docker Hub.

Date butoir : Samedi 11/05/2024