



**SMART WEB AGENCY**

5 RUE JACQUES HENRI LARTIGUE  
92130 ISSY-LES-MOULINEAUX  
COURRIEL : [contact@smartweb-agency.com](mailto:contact@smartweb-agency.com)  
SIRET : 90787885400013  
CODE APE : 6312Z

**Administrateur de Système DevOps**

**Zemni Kassas Sarra**

**29 Janvier 2024**

## Table des Matières

<b>1. Présentation de L'auteur .....</b>	<b>3</b>
<b>2. Projet.....</b>	<b>3</b>
2.1. <b>Entreprise .....</b>	<b>3</b>
2.2. <b>Représentation du projet.....</b>	<b>3</b>
2.3. <b>Langages et Framework utiliser dans ce projet.....</b>	<b>4</b>
<b>3.   Projet .....</b>	<b>7</b>
3.1. <b>Automatiser la création de serveurs à l'aide de scripts.....</b>	<b>7</b>
3.2. <b>Automatiser le déploiement d'une infrastructure .....</b>	<b>11</b>
3.3. <b>Sécuriser l'infrastructure .....</b>	<b>15</b>
3.4. <b>Mettre l'infrastructure en production dans le cloud.....</b>	<b>19</b>
<b>4.   Déployer en continu une application.....</b>	<b>25</b>
4.1. <b>Préparer un environnement de test .....</b>	<b>25</b>
4.2. <b>Gérer le stockage des données .....</b>	<b>31</b>
4.3. <b>Gérer des containers.....</b>	<b>33</b>
4.4. <b>Automatiser la mise en production d'une application avec une plateforme .....</b>	<b>35</b>
<b>5.   Superviser les services déployés.....</b>	<b>41</b>
5.1. <b>Définir et mettre en place des statistiques de services.....</b>	<b>41</b>
5.2. <b>Exploiter une solution de supervision.....</b>	<b>43</b>
<b>6.   Conclusion .....</b>	<b>50</b>
<b>7.   Remerciement .....</b>	<b>51</b>
<b>8.   Les résultats obtenus.....</b>	<b>52</b>

## 1. Présentation de L'auteur



Avec Une bonne base en Algorithme et Passionné par la réalisation et le développement, je me suis tout naturellement tourné vers une carrière de développeuse. J'étais étudiante à l'université René Descartes à rue Saints Pères, 75006 Paris en mathématique Informatique et j'ai suivi une formation ERP\GPE en développement web et application .... En effet, Pour compléter mes connaissances et la forger, j'ai suivi cette formation "Administrateur de système DevOps" pour exploiter le domaine de travail et faciliter mon intégration. Je mettrai mon savoir-faire en pratique, ainsi que mes bonnes connaissances en développement et l'infrastructure ... Je suis persuadé que je saurai dûment compléter mes connaissances. Travailleuse, efficace, sociable, organisé et doté d'un esprit d'analyse, je prendrai soin d'assimiler les différents besoins et saurai facilement m'intégrer.

## 2. Présentation du Projet

### 2.1. Entreprise :

Il s'agit d'une société de développement Nommé SMART WEB AGENCY, société par actions simplifiée, immatriculée sous le SIREN 907878854, est en activité depuis 2 ans. Localisée à ISSY-LES-MOULINEAUX (92130), elle est spécialisée dans le secteur d'activité Informatique.

**Activité :** activités des portails internet, activités des agences de publicité, conseil en systèmes et logiciels informatiques, conseil pour les affaires et autres conseils de gestion, le commerce de détail ou de gros de tous types de produits par internet spécialisé comme le commerce de vêtements, textile, chaussures, articles de maroquinerie, bagagerie, bijoux fantaisie, articles de décoration, articles et accessoires de puériculture, jeux et jouets, produits cosmétiques, hygiènes et beauté

**Adresse :** 5 Rue Jacques Henri Lartigue 92130 Issy-les-Moulineaux

### 2.2. Représentation du projet

Au cours de mon stage enrichissant au sein de Smart Web Agency, j'ai eu l'opportunité de plonger au cœur du développement web et d'acquérir une compréhension approfondie des défis auxquels sont confrontées les équipes de développement dans le monde moderne. Fort de cette expérience, je me lance dans un nouveau projet DevOps visant à optimiser nos processus de livraison logicielle, renforcer la collaboration entre les équipes de développement et d'exploitation, et accélérer le déploiement de produit de haute qualité.

#### Contexte du Projet :

Smart Web Agency, en tant qu'entreprise dynamique de développement web, a connu une croissance significative au fil des ans. Cependant, cette croissance rapide a entraîné des défis en matière de gestion des déploiements, de coordination entre les équipes de développement et d'exploitation, et de garantie de la stabilité des applications en production. Mon projet DevOps vise à résoudre ces défis en introduisant des pratiques et des outils modernes qui favoriseront l'efficacité, la fiabilité et la scalabilité de nos opérations de développement.

#### Objectifs Principaux :

**a. Automatisation des Processus de déploiement :**

- Automatiser la création des virtuels machines sous le cloud
- L'automatisation de l'installation de l'infrastructure.
- Mettre en place des pipelines d'intégration continue et de déploiement continu pour automatiser les phases de test, de construction et de déploiement des applications.

**b. Collaboration Renforcée :**

- Faciliter la communication et la collaboration entre les équipes de développement et d'exploitation, favorisant ainsi un cycle de développement plus rapide et plus fluide.
- Préparer un environnement de test.
- Gérer le stockage des données , les containers et automatiser la mise en production d'une application avec kubernetes.

**c. Surveillance et Gestion des Performances :**

- Introduire des solutions de surveillance avancées pour garantir la stabilité des applications en production, tout en permettant une détection proactive des problèmes de performance.

**d. Sécurité DevOps :**

- Intégrer des pratiques de sécurité dès le début du cycle de développement, en veillant à ce que la sécurité soit une priorité tout au long du processus de livraison.

En somme, ce projet DevOps représente une étape cruciale pour améliorer notre agilité, notre fiabilité et notre capacité à répondre rapidement aux demandes du marché. Je suis Convaincu que les résultats de ce projet contribueront significativement à l'évolution positive de Smart Web Agency dans le domaine de développement web.

## 2.3. Langages et Framework utiliser dans ce projet

Pour la réalisation de ce projet, plusieurs technologies ont été utilisés :

**Environnement de développement :**



**Visual Studio Code** : est un éditeur de code extensible développé par Microsoft pour Windows, Linux et macOS. Les fonctionnalités incluent la prise en charge du débogage, la complétion intelligente du code, la refactorisation du code et Git intégrer.

**Base de données :**



**MySQL** : est un système de gestion de bases de données relationnelles. Il est distribué sous une double licence GPL et propriétaire.

**Logiciel :**



**Terraform** est un environnement logiciel « infrastructure as code » publié par la société HashiCorp Configuration Language (HCL). Cet outil permet d'automatiser la construction des ressources d'une infrastructure de centre de données comme un réseau, des machines virtuelles, un groupe de sécurité ou une base de données (Il est aussi possible d'utiliser le langage JSON)



**Ansible** est une plateforme logicielle pour la configuration et la gestion des ordinateurs. Elle combine le déploiement de logiciels (en) multinœuds, l'exécution des tâches ad-hoc et la gestion de configuration. Elle gère les différents nœuds à travers SSH. Les modules communiquent via la sortie standard en notation JSON et peuvent être écrits dans n'importe quel langage de programmation. Le système utilise YAML pour exprimer appelées playbook.



**Docker** Est une plateforme permettant de lancer certaines applications dans des conteneurs logiciels. Il ne s'agit pas de virtualisation, mais de conteneurisation, une forme plus légère qui s'appuie sur certaines parties de la machine hôte pour son fonctionnement. Cette approche permet d'accroître la flexibilité et la portabilité d'exécution d'une application laquelle va pouvoir tourner de façon fiable et prévisible sur une grande variété de machines hôtes, que ce soit sur la machine locale, un cloud privé ou public, une machine nue. Techniquement, Docker étend le format de conteneur Linux standard, LXC, avec une API de haut niveau fournissant une solution pratique de virtualisation qui exécute les processus de façon isolée. Pour ce faire, Docker utilise entre autres LXC, cgroups et le noyau Linux lui-même. Contrairement aux machines virtuelles traditionnelles, un conteneur Docker n'inclut pas de système d'exploitation, mais s'appuie au contraire sur les fonctionnalités du système d'exploitation fournies par la machine hôte.



**Kubernetes** (communément appelé « K8s2 ») est un système open source qui vise à fournir une « plate-forme permettant d'automatiser le déploiement, la montée en charge et la mise en œuvre de conteneurs d'application sur des grappes de serveurs ». Il fonctionne avec toute une série de technologies de conteneurisation, et est souvent utilisé avec Docker. Il a été conçu à l'origine par Google, puis offert à la Cloud Native Computing Foundation. Kubernetes définit un jeu d'outils ("primitives") qui, ensemble, fournissent des mécanismes pour déployer, maintenir et mettre à l'échelle des applications. Ces éléments qui composent Kubernetes sont conçus pour être combinés et extensibles et donc permettre de supporter une grande variété de charge de travail. Cette extensibilité est fournie en grande partie par l'API de Kubernetes, qui est utilisée par les composants internes aussi bien que par les extensions et les conteneurs tournant sur Kubernetes.



**Git** est un logiciel de gestion de versions décentralisé. Il s'agit du logiciel de gestion de versions le plus populaire dans le développement logiciel et web, utiliser sur tous les environnements (Windows, Mac, Linux). Git est aussi le système à la base du célèbre site web GitHub, le plus important hébergeur de code informatique.



**Jenkins** est un outil open source de serveur d'automatisation. Il aide à automatiser les parties du développement logiciel liées au build, aux tests et au déploiement, et facilite l'intégration continue et la livraison continue. Écrit en Java, Jenkins fonctionne dans un conteneur de servlets tel qu'Apache Tomcat, ou en mode autonome avec son propre serveur Web embarqué. Il s'interface avec des systèmes de gestion de versions tels que CVS, Git et Subversion, et exécute des projets basés sur Apache Ant et Apache Maven aussi bien que des scripts arbitraires en shell Unix ou batch Windows. Ses fonctionnalités peuvent être étendues facilement

à l'aide de plugins supplémentaires qui permettra, par exemple, de connecter l'outil à d'autres environnements de test .



**MobaXterm** est votre boîte à outils ultime pour l'informatique à distance. Dans une seule application Windows, il fournit de nombreuses fonctions adaptées aux programmeurs, aux webmasters, aux administrateurs informatiques et à presque tous les utilisateurs qui doivent gérer leurs tâches à distance de manière plus simple. MobaXterm fournit tous les outils réseau distants importants (SSH, X11, RDP, VNC, FTP, MOSH, ...) et les commandes Unix (bash, ls, cat, sed, grep, awk, rsync, ...) sur le bureau Windows.

### Application



**Nagios** (anciennement appelé Netsaint) est une application permettant la surveillance système et réseau. Elle surveille les hôtes et services spécifiés, alertant lorsque les systèmes ont des dysfonctionnements et quand ils repassent en fonctionnement normal. C'est un logiciel libre sous licence GPL. C'est un programme modulaire qui se décompose en trois parties :

Le moteur de l'application qui vient ordonner les tâches de supervision ;  
L'interface web, qui permet d'avoir une vue d'ensemble du système d'information et des possibles anomalies ;

### Système d'exploitation



**Ubuntu** est une distribution Linux fondée sur Debian. Elle est développée, commercialisée et maintenue pour les ordinateurs individuels (desktop), les serveurs (Server) et les objets connectés (Core) par la société Canonical.

Ubuntu est disponible en deux versions, une qui évolue tous les six mois, et une version LTS, pour Long Term Support (« Support long terme ») qui évolue tous les deux ans. Ubuntu se définit comme « un système d'exploitation utilisé par des millions de PC à travers le monde » et avec une interface « simple, intuitive, et sécurisée ». Elle est la distribution la plus utilisée pour accéder aux sites web d'après le site Alexa, et le système d'exploitation le plus utilisé pour les serveurs informatiques

### Plateforme applicative en nuage et le Cloud Computing :



#### Microsoft Azure (Windows Azure) :

Il s'agit d'une offre d'hébergement (applications et données) et de services (workflow, stockage et synchronisation des données, bus de messages, contacts...). La plateforme Azure repose sur l'infrastructure de Microsoft. Un ensemble d'API permet d'utiliser et d'accéder à cette plate-forme et aux services associés. Ces API sont exposées au travers d'un portail web (<https://portal.azure.com> [archive]) qui permet de gérer l'ensemble des services Azure. Cependant, pour une utilisation plus opérationnelle, il est recommandé d'utiliser « Azure (Remote) PowerShell » qui permet d'effectuer des actions plus granulaires, scriptables et pouvant être regroupées dans une boîte à outils.

### 3. Automatiser le déploiement d'une infrastructure dans le cloud

#### 3.1. Automatiser la création de serveurs à l'aide de scripts

Je commence par l'automatisation de la création de la machine virtuel pour chaque développeurs et employeurs dans notre société.

Dans cette étape j'ai proposé à faire la création de machine virtuelle chez Azure à l'aide de python :

- J'ai utilisé mon compte de la plateforme cloud azure pour la création des serveur cloud.
- Appliquant un scripts python et avec des commandes d'Azur CLI (command line interface), pour la création automatique des serveurs des machines virtuelles.
- On premier, j'ai installé Azure CLI et je me suis connecté à mon compte de la plateforme cloud Azure

dans la machine virtuelle Ubuntu on installe azure :

```
*****  
curl -sL https://aka.ms/InstallAzureCLIDeb | sudo bash  
*****  
# connecter au compte azure  
az login -u <username> -p <password>
```

```
zemni@ubuntu:~$ az login -u sarra.kassas@labom2formation.fr -p [REDACTED]  
[{"cloudName": "AzureCloud",  
 "homeTenantId": "373016f8-79a9-4eed-80d2-100ce948d960",  
 "id": "e0cace2d-3d81-4c4a-8787-30d6fc5412fe",  
 "isDefault": true,  
 "managedByTenants": [],  
 "name": "Azure subscription 1",  
 "state": "Enabled",  
 "tenantId": "373016f8-79a9-4eed-80d2-100ce948d960",  
 "user": {  
   "name": "sar  
   "type": "user"  
 }  
}
```

**Installer les packages de bibliothèque Azure nécessaire :** Créez un fichier *requirements.txt* qui répertorie les bibliothèques de gestion utilisées dans cet exemple :

```
azure-mgmt-resource  
azure-mgmt-compute  
azure-mgmt-network  
azure-identity
```

Ensuite, dans votre terminal ou invite de commandes avec l'environnement virtuel activé, installez les bibliothèques de gestion répertoriées dans *requirements.txt* :

```
pip install -r requirements.txt
```

**Écrire du code pour créer une machine virtuelle :** Créez un fichier Python nommé `provision_vm.py` avec le code suivant. Les commentaires expliquent les détails : Ce code utilise l'authentification basée sur CLI (en utilisant AzureCliCredential), car il montre les actions que on peut faire autrement faire directement à l'aide d'Azure CLI. Selon l'environnement, vous devrez peut-être d'abord exécuter az login pour vous authentifier.

Pour utiliser ce type de code dans un script de production (par exemple, pour automatiser la gestion des machines virtuelles), utilisez DefaultAzureCredential (recommandé) ou une méthode basée sur un principal de service, comme décrit dans Comment authentifier des applications Python avec les services Azure.

```

1  # Import the needed credential and management objects from the libraries.
2  import os
3
4
5  from azure.identity import AzureCliCredential
6  from azure.mgmt.compute import ComputeManagementClient
7  from azure.mgmt.network import NetworkManagementClient
8  from azure.mgmt.resource import ResourceManagementClient
9
10 print(
11     "Provisioning a virtual machine...some operations might take a \
12 minute or two."
13 )
14
15 # Acquire a credential object using CLI-based authentication.
16 credential = AzureCliCredential()
17
18 # Retrieve subscription ID from environment variable.
19 subscription_id = os.environ["AZURE_SUBSCRIPTION_ID"]
20
21 # Provision a resource group
22 # from the CLI login.
23 resource_client = ResourceManagementClient(credential, subscription_id)
24 RESOURCE_GROUP_NAME = input("Tapez le nom du Groupe : ") #PythonAzureExample-VM-rg"
25 LOCATION = "westus2"
26
27 # Provision the resource group.
28 rg_result = resource_client.resource_groups.create_or_update(
29     RESOURCE_GROUP_NAME, {"location": LOCATION}
30 )
31
32 print(
33     f"Provisioned resource group {rg_result.name} in the \
34 {rg_result.location} region"
35 )
36

```

AzureCliCredential : Active l'authentification à Microsoft Entra ID à l'aide d'Azure CLI pour obtenir un jeton d'accès.

Os.environ= cette commande va afficher la liste des variables d'environnement de l'utilisateur.

ResourceManagementClient = Cette classe Fournit des opérations pour travailler avec des ressources et des groupes de ressources.

Ensuite il aura un message pour demander d'entrer le nom de groupe, et la saisir.

Et je mais par defaut la location de data vers le « westus2 »

29/01/24

SMART WEB AGENCY

```

36
37 # Network and IP address names
38 VNET_NAME = input("Tapez le nom du Network: ") # "python-example-vnet"
39 SUBNET_NAME = input("Tapez le nom du Sous-réseau: ") # "python-example-subnet"
40 IP_NAME = input("Tapez le nom de l'adresse Ip : ") # "python-example-ip"
41 IP_CONFIG_NAME = input("Tapez le nom de IP-configuration : ") # "python-example-ip-config"
42 NIC_NAME = input("Tapez le nom de NIC: ") # "python-example-nic"
43
44 # Obtain the management object for networks
45 network_client = NetworkManagementClient(credential, subscription_id)
46
47 # Provision the virtual network and wait for completion
48 poller = network_client.virtual_networks.begin_create_or_update(
49     RESOURCE_GROUP_NAME,
50     VNEN_NAME,
51     {
52         "location": LOCATION,
53         "address_space": {"address_prefixes": ["10.0.0.0/16"]},
54     },
55 )
56
57 vnet_result = poller.result()
58
59 print(
60     f"Provisioned virtual network {vnet_result.name} with address \
61 prefixes {vnet_result.address_space.address_prefixes}"
62 )
63
64 # Provision the subnet and wait for completion
65 poller = network_client.subnets.begin_create_or_update(
66     RESOURCE_GROUP_NAME,
67     VNEN_NAME,
68     SUBNET_NAME,
69     {"address_prefix": "10.0.0.0/24"},
70 )
71 subnet_result = poller.result()
72

```

Un message va être afficher : pour demander d'enter le nom de resau , sous reseau , le nom de l'adresse IP et le nom de la configuration de l'adresse IP.

La Classe de `NetworkManagementClient` c'est le Client réseau : fait généralement partie d'une bibliothèque ou d'un SDK (Software Development Kit) qui permet d'interagir avec les services de gestion réseau. Cette classe peut être utilisée pour effectuer diverses opérations liées à la gestion des réseaux, telles que la création, la modification ou la suppression de ressources réseau.

```
72
73     print(
74         f"Provisioned virtual subnet {subnet_result.name} with address \
75 prefix {subnet_result.address_prefix}"
76     )
77
78     # Provision an IP address and wait for completion
79     poller = network_client.public_ip_addresses.begin_create_or_update(
80         RESOURCE_GROUP_NAME,
81         IP_NAME,
82         {
83             "location": LOCATION,
84             "sku": {"name": "Standard"},
85             "public_ip_allocation_method": "Static",
86             "public_ip_address_version": "IPV4",
87         },
88     )
89
90     ip_address_result = poller.result()
91
92     print(
93         f"Provisioned public IP address {ip_address_result.name} \
94 with address {ip_address_result.ip_address}"
95     )
96
97     # Provision the network interface client
98     poller = network_client.network_interfaces.begin_create_or_update(
99         RESOURCE_GROUP_NAME,
100        NIC_NAME,
101        {
102            "location": LOCATION,
103            "ip_configurations": [
104                {
105                    "name": IP_CONFIG_NAME,
106                    "subnet": {"id": subnet_result.id},
107                    "public_ip_address": {"id": ip_address_result.id},
108                }
109            ]
110        }
111    )
112
```

- On crée L'API de nom de groupe qui contient le NETWORK, le subnet et l'adresse IP avec une version IPV4 et une adresse IP « STATIC » elle est statistique.
- Un message s'affiche, qu'il demandera le nom de la machine virtuel le nom de l'utilisateur et le mot de passe.
- Et on ajoute l'API de la machine virtuelle qui contient, le nom de la machine, un système d'exploitation Ubuntu et une version latest, avec un utilisateur et mot de passe

```

109     ],
110   },
111 )
112
113 nic_result = poller.result()
114
115 print(f"Provisioned network interface client {nic_result.name}")
116
117 # Provision the virtual machine
118
119 # Obtain the management object for virtual machines
120 compute_client = ComputeManagementClient(credential, subscription_id)
121
122 VM_NAME = input("Tapez le nom de la virtuel machine : ")      # "ExampleVM"
123 USERNAME = input("Tapez le nom de l'utilisateur : ")          # "azureuser"
124 PASSWORD = input("Tapez le mot de passe de l'utilisateur : ")  # "ChangePa$$w0rd24"
125
126 print(
127   f"Provisioning virtual machine {VM_NAME}; this operation might \
128 take a few minutes."
129 )
130
131 poller = compute_client.virtual_machines.begin_create_or_update(
132   RESOURCE_GROUP_NAME,
133   VM_NAME,
134   {
135     "location": LOCATION,
136     "storage_profile": {
137       "image_reference": {
138         "publisher": "canonical",
139         "offer": "UbuntuServer",
140         "sku": "16.04.0-LTS",
141         "version": "latest",
142       }
143     },
144     "hardware_profile": {"vm_size": "Standard_DS1_v2"},
145     "os_profile": {

```

```

146       "os_type": "Linux",
147       "computer_name": VM_NAME,
148       "admin_username": USERNAME,
149       "admin_password": PASSWORD,
150     },
151     "network_profile": {
152       "network_interfaces": [
153         {
154           "id": nic_result.id,
155         }
156       ],
157     },
158   },
159
160   vm_result = poller.result()
161
162 print(f"Provisioned virtual machine {vm_result.name}")
163
164

```

## Exécuter le script :

`python provision_vm.py`

Nom	Type	Abonnement	Groupe de ressources	Emplacement	État	Système d'exploit.	Taille	Adresse IP publique
vm-dev	Machine virtuelle	Azure subscription 1	group-dev	West Europe	En cours d'exécution	Linux	Standard_B1s	13.69.51.151
vm-ops	Machine virtuelle	Azure subscription 1	group-ops	Canada Central	En cours d'exécution	Linux	Standard_B1s	20.63.39.76
vm-prod	Machine virtuelle	Azure subscription 1	group-prod	West US	En cours d'exécution	Linux	Standard_B1s	168.62.22.99
vm-test	Machine virtuelle	Azure subscription 1	group-test	East US	En cours d'exécution	Linux	Standard_B1s	52.188.71.129

## 3.2. Automatiser le déploiement d'une infrastructure

L'étape suivante je passe à l'automatisation de l'installation et la sécurité de l'infrastructure des machines virtuel :

Je vais installer Ansible sur ma machine à l'aide d'un scripte :

Le fichier install-ansible.sh :

```
#!/bin/bash
sudo apt update
sudo apt -y upgrade
sudo apt install -y python3-pip

sudo apt-add-repository ppa:ansible/ansible
sudo apt update
sudo apt install ansible
sudo nano /etc/ansible/hosts
ansible --version
```

J'exécute ce scripte à l'aide de cette commande : `bash ./install-ansible.sh`

```
zemni@vm-ops:~$ ansible --version
ansible [core 2.16.1]
  config file = None
  configured module search path = ['/home/zemni/.ansible/plugins/modules', '/usr/share/ansible/plugins/modules']
  ansible python module location = /home/zemni/.local/lib/python3.10/site-packages/ansible
  ansible collection location = /home/zemni/.ansible/collections:/usr/share/ansible/collections
  executable location = /home/zemni/.local/bin/ansible
  python version = 3.10.12 (main, Nov 20 2023, 15:14:05) [GCC 11.4.0] (/usr/bin/python3)
  jinja version = 3.0.3
  libyaml = True
zemni@vm-ops:~$ █
```

Ce Playbook Ansible nous fournira une alternative à l'exécution manuelle de la procédure d'installation générale d'un serveur LAMP (Linux, Apache, MySQL et PHP). L'exécution de ce Playbook automatisera donc les actions suivantes sur nos hôtes distants :

- Côté serveur web :
- Installer les package suivantes : apache2, php, python3, vim, Java, Nginx, GIT, Jenkins
- La configuration des Framework et les mettre à jours
- S'assurer que les services sont bien démarrés
  - Côté serveur base de données :
  - Installer les package mysql-server et mysql-client
  - Cree le mot de passe root
  - Cree un utilisateur pour la base de donner avec son mot de passe
  - Cree une base de données et la nommé
  - Autoriser notre serveur web à communiquer avec la base de données
    - Et installer pour tous les serveurs : l'Environnement de développement visuel Studio

Pour ce projet, j'ai décidé de me séparer du fichier inventaire situé par défaut dans `/etc/ansible/hosts` et de créer mon propre fichier `hosts` à la racine du projet, où je me suis permis de séparer le serveur de base de données par rapport au serveur web, voici donc à quoi ressemble notre nouveau fichier inventaire :

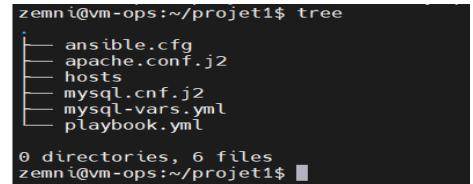
29/01/24

## SMART WEB AGENCY

```
GNU nano 6.2
[all]
vm-dev    ansible_host=192.168.163.147 ansible_user=zemni ansible_password=Reponse ansible_sudo_pass=Reponse
vm-prod   ansible_host=192.168.163.148 ansible_user=zemni ansible_password=Reponse ansible_sudo_pass=Reponse
[web]
vm-dev    ansible_host=192.168.163.147 ansible_user=zemni ansible_password=Reponse ansible_sudo_pass=Reponse
[db]
vm-prod   ansible_host=192.168.163.148 ansible_user=zemni ansible_password=Reponse ansible_sudo_pass=Reponse
[all:vars]
ansible_python_interpreter=/usr/bin/python3
```

- Valeur de la variable inventory, ce qui nous donne le fichier ansible.cfg suivant :

```
GNU nano 6.2                               ansible.cfg
[defaults]
inventory = ./hosts
```



- Variables : Concernant les variables, vous avez le choix entre les placer directement depuis le mot-clé vars, ou vous pouvez les charger depuis un fichier en utilisant le mot-clé vars\_files comme ceci :
- vars\_files: mysql-vars.yml
- 
- Voici le contenu du fichier de variables

```
zemni@vm-ops:~$ ssh-keygen
Generating public/private rsa key pair.
Enter file in which to save the key (/home/zemni/.ssh/id_rsa):
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /home/zemni/.ssh/id_rsa
Your public key has been saved in /home/zemni/.ssh/id_rsa.pub
The key fingerprint is:
SHA256:zAhWjPbxeuISKGhhviPlz921Sg1p00Vx8LYyFGfdC0 zemni@vm-ops
The key's randomart image is:
+---[RSA 3072]----+
| ..o.. o |
| . + =+.BoEo. |
| . o . . +o. |
| . .+ . + o |
| . . o.. So B |
| o o . . * o |
| o o . + o |
| ..o + o |
| ..o.o.o |
+---[SHA256]----+
zemni@vm-ops:~$ cd .ssh
zemni@vm-ops:~/ssh$ ls
authorized_keys  id_rsa  id_rsa.pub
zemni@vm-ops:~/ssh$ ssh-copy-id zemni@192.168.163.147
/usr/bin/ssh-copy-id: INFO: Source of key(s) to be installed: "/home/zemni/.ssh/
id_rsa.pub"
The authenticity of host '192.168.163.147 (192.168.163.147)' can't be established.
ED25519 key fingerprint is SHA256:lRlI+aEBufIG4iynb845g5VojPL/DJdmYIMWmlo0fZs.
This key is not known by any other names
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
/usr/bin/ssh-copy-id: INFO: attempting to log in with the new key(s), to filter
out any that are already installed
/usr/bin/ssh-copy-id: INFO: 1 key(s) remain to be installed -- if you are prompt
ed now it is to install the new keys
zemni@192.168.163.147's password:

Number of key(s) added: 1

Now try logging into the machine, with: "ssh 'zemni@192.168.163.147'"
```

## Les Taches dans le fichier playbook.yml

```
GNU nano 6.2                               playbook.yml
---
- name: Deploy Infrastructure
  hosts: web
  become: true

  tasks:
    - name: Update apt package cache
      apt:
        update_cache: yes

    - name: Mise à jour des paquets
      apt:
        update_cache: yes
      become: yes

    - name: Installation de paquets de base
      apt:
        name: "{{ item }}"
        state: present
      become: yes
      with_items:
        - nginx
        - vim
        - git

    - name: Install required packages
      apt:
        name: "{{ item }}"
        state: present
      with_items:
        - apache2
        - mysql-client
        - php
        - python3
        - python3-pip

    - name: Configure Apache
      template:
```

```
src: apache.conf.j2
dest: /etc/apache2/sites-available/default
notify:
  - Restart Apache

handlers:
  - name: Reload Nginx
    systemd:
      name: nginx
      state: reloaded
  - name: Restart Apache
    service:
      name: apache2
      state: restarted

  - name: Restart MySQL
    service:
      name: mysql
      state: restarted

#####
- name: Installer MySQL sous Ubuntu
  hosts: db
  become: yes
  vars_files: mysql-vars.yml

  tasks:
    - name: Mise à jour des paquets
      apt:
        update_cache: yes

    - name: Installation de MySQL Server
      apt:
        name: mysql-server
        state: present
      - name: Sécuriser l'installation MySQL
```

```

mysql_user:
  name: root
  password: "{{ mysql_root_password }}"
  host: "{{ item }}"
  login_user: root
  login_password: ""

  with_items:
    - "{{ ansible_hostname }}"
    - 127.0.0.1
    - ::1
    - "localhost"

- name: Créer une base de données
  mysql_db:
    name: "{{ mysql_database }}"
    state: present
    login_user: root
    login_password: "{{ mysql_root_password }}"

- name: Créer un utilisateur MySQL
  mysql_user:
    name: "{{ mysql_user }}"
    password: "{{ mysql_user_password }}"
    priv: "{{ mysql_database }}.*:ALL"
    state: present
    login_user: root
    login_password: "{{ mysql_root_password }}"

#####
- name: Installation de Visual Studio Code sous Ubuntu
  hosts: all
  become: yes
  tasks:
    - name: Téléchargement de la clé GPG Microsoft pour VSCode
      apt_key:
        url: https://packages.microsoft.com/keys/microsoft.asc
        state: present
    - name: Ajout du dépôt VSCode
      apt_repository:
        repo: "deb [arch=amd64] https://packages.microsoft.com/repos/vscode stable main"
        state: present
    - name: Mise à jour du cache apt
      apt:
        update_cache: yes
    - name: Installation de Visual Studio Code
      apt:
        name: code
        state: present

- name: Installation de Java sous Ubuntu
  hosts: web
  become: yes
  tasks:
    - name: Mise à jour du cache apt
      apt:
        update_cache: yes
    - name: Installation de Java OpenJDK 11
      apt:
        name: openjdk-11-jdk
        state: present

handlers:
- name: Reload Environment

handlers:
- name: Reload Environment
  become: yes
  command: "systemctl --quiet is-active graphical.target && /usr/sbin/loginctl --no-p

- name: Install Jenkins on Ubuntu
  hosts: your_servers
  become: true
  tasks:
    - name: Import Jenkins GPG key
      apt_key:
        url: https://pkg.jenkins.io/debian-stable/jenkins.io.key
        state: present
    - name: Add Jenkins repository
      apt_repository:
        repo: deb http://pkg.jenkins.io/debian-stable binary/
        state: present
    - name: Update apt package cache
      apt:
        update_cache: yes
    - name: Install Jenkins
      apt:
        name: jenkins
        state: present
    - name: Start Jenkins service
      service:
        name: jenkins
        state: started
        enabled: yes

```

```
zemni@vm-ops:~/projet$ ansible-playbook -i hosts playbook.yml
```

```
zemni@vm-ops:~/projet1$ ansible-playbook -i hosts playbook.yml
PLAY [Deploy Infrastructure] ****
TASK [Gathering Facts] ****
ok: [vm-dev]
TASK [Update apt package cache] ****
changed: [vm-dev]
: TASK [Mise à jour des paquets] ****
changed: [vm-dev]
TASK [Installation de paquets de base] ****
ok: [vm-dev] => (item=nginx)
ok: [vm-dev] => (item=vim)
ok: [vm-dev] => (item=git)
TASK [Install required packages] ****
ok: [vm-dev] => (item=apache2)
ok: [vm-dev] => (item=mysql-client)
ok: [vm-dev] => (item=php)
ok: [vm-dev] => (item=python3)
ok: [vm-dev] => (item=python3-pip)
TASK [Configure Apache] ****
ok: [vm-dev]
PLAY [Installer MySQL sous Ubuntu] ****
TASK [Gathering Facts] ****
ok: [vm-prod]
TASK [Mise à jour des paquets] ****
changed: [vm-prod]
TASK [Installation de MySQL Server] ****
ok: [vm-prod]
TASK [Sécuriser l'installation MySQL] ****
TASK [Sécuriser l'installation MySQL] ****
changed: [vm-prod] => (item=vm-prod)
changed: [vm-prod] => (item=v8_0_1)
changed: [vm-prod] => (item=v5_7)
changed: [vm-prod] => (item=localhost)
TASK [Créer une base de données] ****
changed: [vm-prod]
TASK [Créer un utilisateur MySQL] ****
ok: [vm-prod]
PLAY [Installation de Visual Studio Code sous Ubuntu] ****
TASK [Gathering Facts] ****
ok: [vm-prod]
ok: [vm-dev]
TASK [Téléchargement de la clé GPG Microsoft pour VSCode] ****
changed: [vm-dev]
changed: [vm-prod]
TASK [Ajout du dépôt VSCode] ****
changed: [vm-prod]
changed: [vm-dev]
TASK [Mise à jour du cache apt] ****
changed: [vm-dev]
changed: [vm-prod]
TASK [Installation de Visual Studio Code] ****
PLAY [Installation de Java sous Ubuntu] ****
TASK [Gathering Facts] ****
ok: [vm-dev]
TASK [Mise à jour du cache apt] ****
changed: [vm-dev]
TASK [Installation de Java OpenJDK 11] ****
ok: [vm-dev]
[WARNING]: Could not match supplied host pattern, ignoring: your_servers
PLAY [Install Jenkins on Ubuntu] ****
skipping: no hosts matched
PLAY RECAP ****
vm-dev : ok=14    changed=4      unreachable=0    failed=0    skipped=0
rescued=0  ignored=0
vm-prod   : ok=5     changed=1      unreachable=0    failed=0    skipped=0
rescued=0  ignored=0
zemni@vm-ops:~/projet1$
```

### 3.3. Sécuriser l'infrastructure

Pour la sécurité de l'infrastructure je continue à l'automatiser à l'aide des scripte « Ansible ». Tous d'abord j'ai créé un nouveau dossier pour la sécurité de l'infrastructure. Voici une structure de répertoire exemple pour votre projet :

```
zemni@vm-ops:~/projet2$ tree
.
├── ansible.cfg
└── hosts
    └── playbook.yml
    └── playbook.yml.save

0 directories, 4 files
```

L'automatisation des tâches avec le fichier playbook.yml qui inclut certaines pratiques de sécurité courantes pour les serveurs Ubuntu : Cela inclut des tâches telles que la configuration du pare-feu, la

mise à jour des packages, la désactivation de l'accès root via SSH, et l'installation de fail2ban pour la détection des tentatives d'intrusion :

```

GNU nano 6.2                               playbook.yml.save
[...]
- name: Sécuriser l'infrastructure sous Ubuntu
  hosts: all
  become: true

  tasks:
    - name: Mise à jour du cache des paquets apt
      apt:
        update_cache: yes

    - name: Mise à jour des packages
      apt:
        upgrade: yes
        autoremove: yes

    - name: Install essential security packages
      apt:
        name: "{{ item }}"
        state: present
      with_items:
        - fail2ban
        - ufw

    - name: Configure Uncomplicated Firewall (UFW)
      ufw:
        rule: allow
        port: "{{ item }}"
      with_items:
        - 22 # SSH
        - 80 # HTTP
        - 443 # HTTPS

    - name: Configuration du pare-feu (UFW)
      ufw:
        rule: allow
        name: "OpenSSH"
        ignore_errors: yes

    - name: Désactivation de l'accès root via SSH
      lineinfile:
        path: /etc/ssh/sshd_config
        regexp: '^PermitRootLogin'
        line: 'PermitRootLogin no'
      notify:
        - Restart SSH

    - name: Set up SSH hardening
      copy:
        src: sshd_config
        dest: /etc/ssh/sshd_config
      notify:
        - Restart SSH

    - name: Installation de fail2ban
      apt:
        name: fail2ban
        state: present

    handlers:
      - name: Restart SSH
        service:
          name: ssh
          state: restarted
      #####
      - name: Sécuriser le compte personnel des utilisateurs
        hosts: all
        become: true
        vars:
          created_username: zemni

```

- Installer et configurer Fail2Ban pour détecter les tentatives d'authentification infructueuses et bloquer les adresses IP malveillantes.
- Gestion des identifiants avec le nom d'utilisateur et le mot de passe et Gestion des utilisateurs et des groupes
- Pour ajouter un membre à un groupe supplémentaire, répertorier les groupes supplémentaires dont l'utilisateur est actuellement membre et les groupes supplémentaires dont l'utilisateur doit devenir membre.
- Si l'utilisateur est déjà membre du groupe `groupa` et doit devenir membre de `groupb`:  
`usermod -G groupa,groupb user-name`
- Pour afficher les membres d'un groupe : `getent group group-name`
- Comprenez comment configurer l'authentification classique avec le nom d'utilisateur et le mot de passe en veillant à utiliser des mots de passe forts.

- Ajouter un nouvel utilisateur au système

Utiliser l'adduser pour ajouter un nouvel utilisateur à votre système :

```
adduser <nom user>
```

Vérifier un mot de passe pour l'utilisateur :

- Utilisez la commande usermod pour ajouter l'utilisateur au groupe **sudo** :

```
usermod -aG sudo sammy
```

- Gestion des clés d'accès, l'utilisation des clés SSH pour l'Authentification et la sécurisation de l'accès SSH :

- La sécurisation des accès ssh est essentielle pour protéger votre serveur contre des accès non autorisés.

Utilisation de clés SSH :

Privilégiez l'authentification par clés ssh plutôt que par mot de passe.

Générez une paire de clés sur la machine avec la commande,

```
# Génerez une paire de clés sur chaque client : ssh-keygen -t rsa .
```

Après on tape la commande suivante :

```
# Copier la clé publique vers le serveur : ssh-copy-id -i ~/.ssh/id_rsa.pub
```

Ou bien on copie la clé publique id\_rsa.pub sur le serveur à l'emplacement  
~/.ssh/authorized\_keys pour chaque utilisateur autorisé.

- Désactivation de l'authentification par mot de passe :

Désactivez l'authentification par mot de passe pour renforcer la sécurité.

Et on peut le remplacer par : éditez-le fichier /etc/ssh/sshd\_config et

Assurez-vous que PasswordAuthentication est réglé sur no. Redémarrez le service  
SSH : sudo service ssh restart

- Port SSH non standard : changer le port ssh par défaut (22) pour rendre plus difficile la détection par des attaquants.

- Désactivation de l'accès Root : Désactiver l'accès direct en tant que root via SSH. Utiliser plutôt des comptes d'utilisateurs normaux et utiliser « sudo » pour les priviléges d'administration. Et ajouter le sudo pour l'utilisateur de chaque machine en modifiant le fichier /etc/sudoers

En tapant dans le root la commande suivante : Sudo visudo

# et on ajoute cette ligne

```
root  ALL=(ALL:ALL) ALL
```

```
%admin  ALL=(ALL) ALL
```

```
%sudo  ALL=(ALL:ALL) ALL
```

# Editer le fichier de configuration SSH : sudo nano /etc/ssh/sshd\_config

#Assurer-vous que PermitRootLogin est défini sur no : PermitRootLogin no

#Redémarrez le service SSH : sudo systemctl restart SSH

Installer le Pare-Feu : Pour surveiller, filtrer et contrôler le trafic réseau entrant et sortant d'un réseau informatique. L'objectif principal d'un pare-feu est de renforcer la sécurité en empêchant ou en autorisant sélectivement le trafic en fonction d'un ensemble de règles prédéfinies.

```

tasks:
  - name: Install aptitude
    apt:
      name: aptitude
      state: latest
      update_cache: true

  - name: Setup passwordless sudo
    lineinfile:
      path: /etc/sudoers
      state: present
      dest: "/etc/sudoers.d/{{ new_user }}"
      regexp: '^%sudo'
      line: '%sudo ALL=(ALL) NOPASSWD: ALL'
      validate: '/usr/sbin/visudo -cf %s'

  - name: Create a new regular user with sudo privileges
    user:
      name: "{{ created_username }}"
      state: present
      groups: sudo
      append: true
      create_home: true

  - name: Set authorized key for remote user
    ansible.posix.authorized_key:
      user: "{{ created_username }}"
      state: present
      key: "{{ lookup('file', lookup('env','HOME') + '/.ssh/id_rsa.pub') }}"

  - name: Disable password authentication for root
    lineinfile:
      path: /etc/ssh/sshd_config
      state: present
      regexp: '#?PermitRootLogin'
      line: 'PermitRootLogin prohibit-password'

```

```

- name: Update apt and install required system packages
  apt:
    pkg:
      - curl
      - ufw
    state: latest
    update_cache: true

- name: UFW - Allow SSH connections
  community.general.ufw:
    rule: allow
    name: OpenSSH

- name: UFW - Enable and deny by default
  community.general.ufw:
    state: enabled
    default: deny

```

Dans ce playbook :

- Le cache des paquets apt est mis à jour.
- Les packages sont mis à jour.
- Le pare-feu UFW est configuré pour autoriser le trafic SSH.
- L'accès root via SSH est désactivé en modifiant la configuration du service SSH.
- Fail2Ban est installé pour détecter et prévenir les tentatives d'intrusion.
- Installation de clés SSH et les autorisations
- Création des user et leurs permission SUDO.

```

zemm@vm-ops:~/projet2$ ansible-playbook -i hosts playbook.yml
[WARNING]: Collection ansible.posix does not support Ansible version 2.16.1
PLAY [Sécuriser l'infrastructure sous Ubuntu] ****
TASK [Gathering Facts] ****
ok: [vm-dev]
ok: [vm-prod]

TASK [Mise à jour du cache des paquets apt] ****
changed: [vm-dev]
changed: [vm-prod]

TASK [Install essential security packages] ****
changed: [vm-dev] => (item=fail2ban)
ok: [vm-dev] => (item=ufw)

TASK [Configure Uncomplicated Firewall (UFW)] ****
changed: [vm-dev] => (item=22)
changed: [vm-dev] => (item=80)
changed: [vm-dev] => (item=443)

TASK [Configuration du pare-feu (UFW)] ****
changed: [vm-dev]

TASK [Désactivation de l'accès root via SSH] ****
changed: [vm-dev]

```

```

TASK [Installation de fail2ban] *****
ok: [vm-dev]
changed: [vm-prod]

PLAY [Sécuriser le compte personnel des utilisateurs] *****
TASK [Gathering Facts] *****
ok: [vm-dev]
ok: [vm-prod]

TASK [Install aptitude] *****
changed: [vm-prod]
changed: [vm-dev]

TASK [Installation de fail2ban] *****
ok: [vm-dev]
changed: [vm-prod]

PLAY [Sécuriser le compte personnel des utilisateurs] *****
TASK [Gathering Facts] *****
ok: [vm-dev]
ok: [vm-prod]

TASK [Install aptitude] *****
changed: [vm-prod]
changed: [vm-dev]

TASK [Create a new regular user with sudo privileges] *****
ok: [vm-prod]
ok: [vm-dev]

TASK [Set authorized key for remote user] *****
ok: [vm-dev]
ok: [vm-prod]

TASK [Disable password authentication for root] *****
changed: [vm-prod]
changed: [vm-dev]

TASK [Update apt and install required system packages] *****
ok: [vm-prod]
ok: [vm-dev]

TASK [UFW - Allow SSH connections] *****
ok: [vm-dev]
changed: [vm-prod]

TASK [UFW - Enable and deny by default] *****
changed: [vm-prod]
changed: [vm-dev]

PLAY RECAP *****
vm-dev      : ok=8    changed=2    unreachable=0    failed=0    skipped=0
vm-prod     : ok=8    changed=3    unreachable=0    failed=0    skipped=0
zemni@vm-ops:~/projet2$ 

```

### 3.4. Mettre l'infrastructure en production dans le cloud

Terraform permet la définition, la prévisualisation et le déploiement de l'infrastructure cloud. À l'aide de Terraform, la création des fichiers de configuration à l'aide de la syntaxe HCL . La syntaxe HCL permet de spécifier le fournisseur cloud – tel qu'Azure – et les éléments qui composent l'infrastructure cloud. Après avoir créé les fichiers de configuration, il faut créer un *plan d'exécution* qui permet de prévisualiser les modifications de l'infrastructure avant leur déploiement. Vérifier les modifications et appliquer le plan d'exécution pour déployer l'infrastructure.

Utiliser les emplacements de déploiement Azure pour basculer entre différentes versions de l'application. Cette capacité aide à minimiser l'impact des déploiements interrompus.

L'utilisation des emplacements de déploiement en peut voir tout au long du déploiement de deux applications via GitHub et Azure. Une application est hébergée dans un emplacement de production. La deuxième application est hébergée dans un emplacement intermédiaire. (Les noms « production » et « staging » sont arbitraires. Ils peuvent être ceux qui conviennent à votre scénario.) Après avoir configuré l'emplacements de déploiement, utiliser Terraform pour basculer entre les deux emplacements selon les besoins.

- a. Configurer l'environnement : voir dans les TP précédentes**
- b. Créer et appliquer le plan Terraform**

- Crée un répertoire nommé clouddrive : `mkdir clouddrive`
- Changer de répertoire en clouddriverépertoire : `cd clouddrive`
- Crée un répertoire nommé deploy : `mkdir deploy`
- Crée un répertoire nommé swap : `mkdir swap`
- Utiliser la ls commande bash pour vérifier qu'il est créé avec succès les deux répertoires.

```
Bash      ~ | ⌂ ? ⓘ ⓘ ⓘ
Requesting a Cloud Shell. Succeeded.
Connecting terminal...

jeffrey@Azure:~/clouddrive$ cd clouddrive
jeffrey@Azure:~/clouddrive$ mkdir deploy
jeffrey@Azure:~/clouddrive$ mkdir swap
jeffrey@Azure:~/clouddrive$ ls
deploy  swap
jeffrey@Azure:~/clouddrive$
```

- Changer de répertoire en deploy répertoire : `cd deploy`
- Créer un fichier nommé deploy.tf : `sudo nano deploy.tf`

```
Dossier Professionnel > 🗂 deploy.tf
1 # Configure the Azure provider
2 provider "azurerm" {
3     # The "feature" block is required for AzureRM provider 2.x.
4     # If you're using version 1.x, the "features" block is not allowed.
5     version = "~>2.0"
6     features {}
7 }
8
9 resource "azurerm_resource_group" "slotDemo" {
10     name = "slotDemoResourceGroup"
11     location = "westus2"
12 }
13
14 resource "azurerm_app_service_plan" "slotDemo" {
15     name          = "slotAppServicePlan"
16     location      = azurerm_resource_group.slotDemo.location
17     resource_group_name = azurerm_resource_group.slotDemo.name
18     sku {
19         tier = "Standard"
20         size = "S1"
21     }
22 }
23
24 resource "azurerm_app_service" "slotDemo" {
25     name          = "slotAppService"
26     location      = azurerm_resource_group.slotDemo.location
27     resource_group_name = azurerm_resource_group.slotDemo.name
28     app_service_plan_id = azurerm_app_service_plan.slotDemo.id
29 }
30
31 resource "azurerm_app_service_slot" "slotDemo" [
32     name          = "slotAppServiceSlotOne"
33     location      = azurerm_resource_group.slotDemo.location
34     resource_group_name = azurerm_resource_group.slotDemo.name
35     app_service_plan_id = azurerm_app_service_plan.slotDemo.id
36     app_service_name   = azurerm_app_service.slotDemo.name
37 ]
```

- Initialiser Terraform. : `terraform init`
- Créer le plan Terraform. : `terraform plan`
- Provisionner les ressources définies dans le deploy.tf fichier de configuration. (Confirmez l'action en entrant yes à l'invite.) :
   
    `terraform apply`
- Dans le menu principal du portail Azure, sélectionnez **Groupes de ressources**.
- Dans l'onglet **Groupes de ressources**, sélectionner **slotDemoResourceGroup**.

NAME	TYPE
slotDemoResourceGroup	

Toutes les ressources créées par Terraform.

NAME	TYPE
slotAppService	App Service
slotAppServiceSlotOne	Web App
slotAppServicePlan	App Service plan

### c. Forker le projet de test

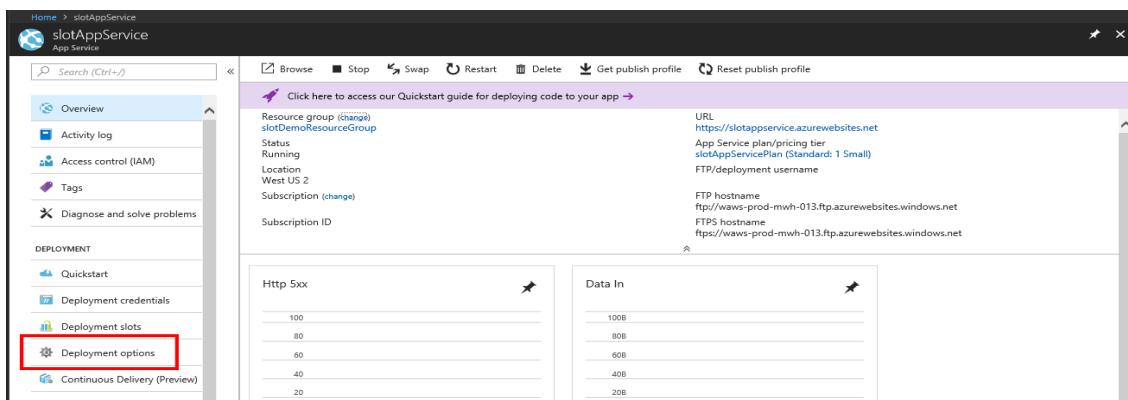
Avant de pouvoir tester la création et l'échange d'emplacements de déploiement, en créer le projet de test depuis GitHub.

1. Accéder au dépôt Awesome-terraform sur GitHub .
2. Forker le dépôt **Awesome-terraform**.

### d. Déployer depuis GitHub vers vos emplacements de déploiement

Après avoir créé le dépôt du projet de test, configurer les emplacements de déploiement en procédant comme suit :

- Dans le menu principal du portail Azure, sélectionner **Groupes de ressources**.
- Sélectionner **slotDemoResourceGroup**.
- Sélectionner **slotAppService**.
- Sélectionner **Options de déploiement**.



Dans l'onglet d'option Déploiement, sélectionner Choisir la source, puis sélectionnez GitHub.

- Une fois qu'Azure a établi la connexion et affiché toutes les options, sélectionner Autorisation.
- Sous l'onglet Autorisation, sélectionner Autoriser et fournir les informations d'identification dont Azure a besoin pour accéder au compte GitHub.
- Une fois qu'Azure a validé les informations d'identification GitHub, un message apparaît et indique que le processus d'autorisation est terminé. Sélectionnez OK pour fermer l'onglet Autorisation.
- Sélectionner Choisir l'organisation et sélectionner votre organisation.
- Sélectionner Choisir un projet.
- Dans l'onglet Choisir un projet, sélectionner le projet Awesome-terraform .
- Sélectionner Choisir une branche.
- Dans l'onglet Choisir une branche, sélectionner master.

The image contains two side-by-side screenshots of the Azure portal's deployment configuration interface.

**Left Screenshot:** The title bar says 'Deployment option Set up deployment option'. It shows a list of source options: Visual Studio Team Services By Microsoft, OneDrive By Microsoft, Local Git Repository By Git, GitHub By GitHub (highlighted with a red box), Bitbucket By Atlassian, Dropbox By Dropbox, and External Repository. Below the list is a note: 'Configure required settings' with a gear icon. At the bottom are 'OK' and 'Cancel' buttons.

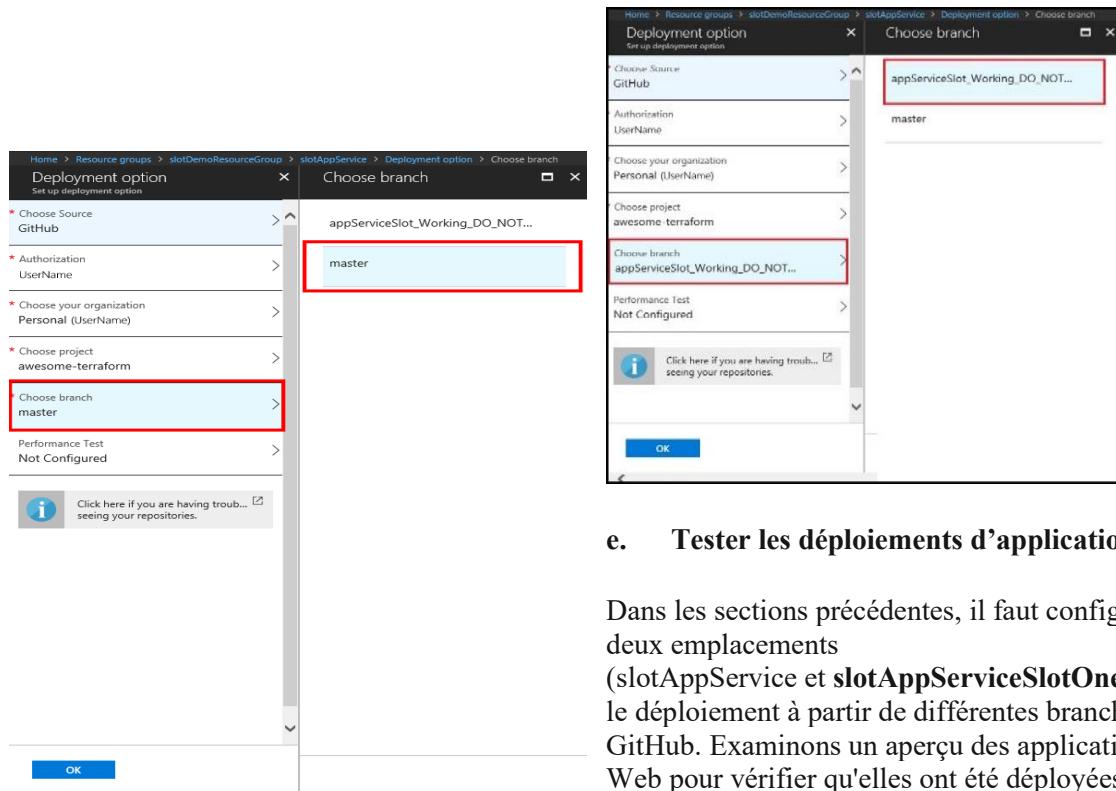
**Right Screenshot:** The title bar says 'Deployment option Set up deployment option'. It shows a list of project selection steps: 'Choose Source GitHub' (highlighted with a red box), 'Authorization UserName', 'Choose your organization Personal (UserName)', 'Choose project Configure required settings' (highlighted with a red box), 'Choose branch Configure required settings' (with a lock icon), and 'Performance Test Not Configured'. Below the list is a note: 'Click here if you are having troub... seeing your repositories.' At the bottom are 'OK' and 'Cancel' buttons.

- Dans l'onglet **Option de déploiement**, sélectionner **OK**.

À ce stade, il faut déployer l'emplacement de production.

Pour déployer l'emplacement intermédiaire, effectuer les étapes précédentes avec les modifications suivantes :

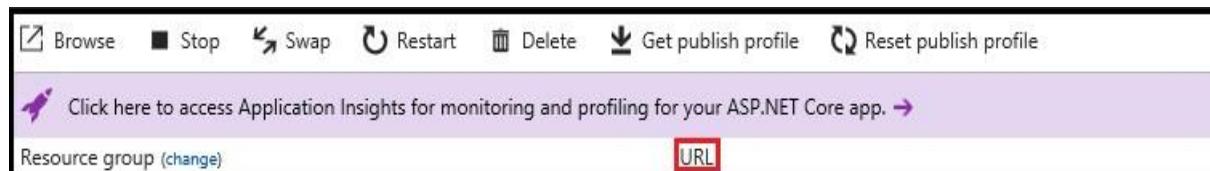
- À l'étape 3, sélectionnez la ressource **slotAppServiceSlotOne**.
- À l'étape 13, sélectionnez la branche de travail.



#### e. Tester les déploiements d'applications

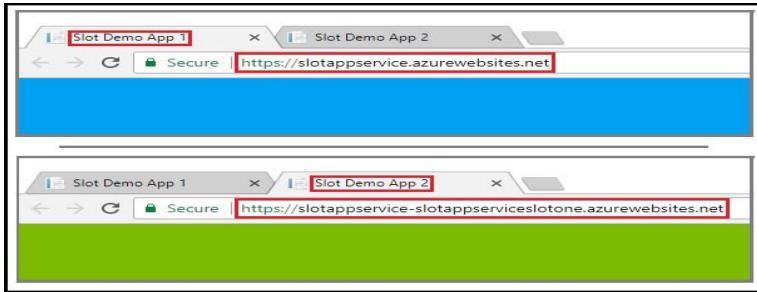
Dans les sections précédentes, il faut configurer deux emplacements (**slotAppService** et **slotAppServiceSlotOne**) pour le déploiement à partir de différentes branches dans GitHub. Examinons un aperçu des applications Web pour vérifier qu'elles ont été déployées avec succès.

- Dans le menu principal du portail Azure, sélectionnez **Groupes de ressources**.
- Sélectionner **slotDemoResourceGroup**.
- Sélectionner **slotAppService** ou **slotAppServiceSlotOne**.
- Sur la page de présentation, sélectionner **URL**.



En fonction de l'application sélectionnée, les résultats suivants :

- Application Web slotAppService** - Page bleue avec un titre de page de **Slot Demo App 1**.
- Application Web slotAppServiceSlotOne** - Page verte avec un titre de page de **Slot Demo App 2**.



## f. Echanger les deux emplacements de déploiement :

Pour tester l'échange des deux emplacements de déploiement, procéder comme suit :

- Basculer vers l'onglet du navigateur qui exécute **slotAppService** (l'application avec la page bleue).
- Revenir au portail Azure dans un onglet séparé.
- Ouvrez Cloud Shell.
- Accéder au répertoire **clouddrive/swap** : `cd clouddrive/swap`
- Dans Cloud Shell, créer un fichier nommé swap.tf. : `sudo nano swap.tf`

```
Dossier Professionnel > swap.tf
1  # Configure the Azure provider
2  provider "azurerm" {
3      # The "feature" block is required for AzureRM provider 2.x.
4      # If you're using version 1.x, the "features" block is not allowed.
5      version = "~>2.0"
6      features {}
7  }
8
9  # Swap the production slot and the staging slot
10 resource "azurerm_app_service_active_slot" "slotDemoActiveSlot" [
11     resource_group_name    = "slotDemoResourceGroup"
12     app_service_name       = "slotAppService"
13     app_service_slot_name = "slotappServiceslotOne"
14 ]
```

- Initialiser Terraform : `terraform init`
- Créer le plan Terraform. : `terraform plan`
- Provisionner les ressources définies dans le swap.tf fichier de configuration. (Confirmer l'action en entrant yes à l'invite.) : `terraform apply`
- Une fois que Terraform a échangé les emplacements, revenez au navigateur. Actualiser la page.

L'application Web dans votre emplacement de préparation **slotAppServiceSlotOne** a été remplacée par l'emplacement de production et est désormais affichée en vert.

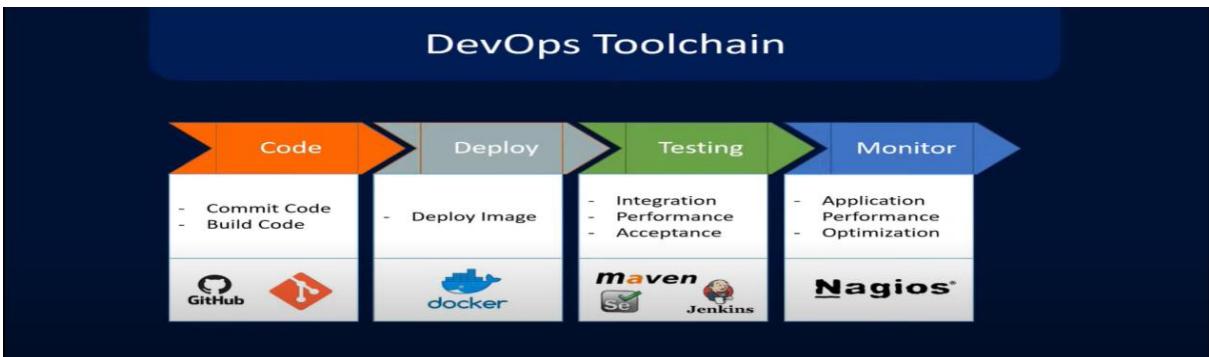


Pour revenir à la version de production d'origine de l'application, réappliquer le plan Terraform que il etait créé à partir du swap.tf fichier de configuration.

`terraform apply`

#### 4. Déployer en continu une application

##### 4.1. Préparer un environnement de test



Dans les travaux précédents : j'ai automatisé l'installation de GIT sur toute les machines et Jenkins sur les machines des devOps pour effectuer les tests. Les développeurs ont codé le projet et chaque développeur a un compte git.

De mon coter j'ai créé le compte principal de GitLab qui réunit tous les groupes des développeurs et chaque groupe à une branche pour le partage de code (on publique que pour les membres du même groupe).

Première étape Configurez votre nom d'utilisateur et votre adresse e-mail dans Git en utilisant les commandes suivantes dans votre terminal :

```
$ git config --global user.name "VotreNom"
$ git config --global user.email "votre@email.com"
```

```
zemni@jenkins:/tmp/git-2.26.2$ git --version
git version 2.26.2
zemni@jenkins:/tmp/git-2.26.2$ git config --list
zemni@jenkins:/tmp/git-2.26.2$ cd
zemni@jenkins:~$ git config --global user.name "Sarra1985"
zemni@jenkins:~$ git config --global user.email "sarraraouf@hotmail.com"
zemni@jenkins:~$ git config --list
user.name=Sarra1985
user.email=sarraraouf@hotmail.com
zemni@jenkins:~$ nano ~/.gitconfig
zemni@jenkins:~$
```

La commande **git init** est utilisée pour initialiser un nouveau référentiel Git dans un répertoire existant. Lorsque vous exécutez cette commande dans un répertoire, Git configure le suivi des changements de ce répertoire, ce qui permet de commencer à utiliser Git pour la gestion de version.

```
zemni@jenkins:~/projet-commerce/SitePhp$ git init
Dépôt Git existant réinitialisé dans /home/zemni/projet-commerce/SitePhp/.git/
zemni@jenkins:~/projet-commerce/SitePhp$ ls
admin           fiche_produit.php  membres.php  profil.php
boutique.php   util              panier.php  README.md
connexion.php  inscription.php  photo        site.sql
zemni@jenkins:~/projet-commerce/SitePhp$ ls -la
total 68
drwxrwxr-x  6 zemni zemni 4096 janv.  6 21:31 .
drwxrwxr-x  4 zemni zemni 4096 janv.  6 21:34 ..
drwxrwxr-x  2 zemni zemni 4096 janv.  6 21:31 admin
-rw-rw-r--  1 zemni zemni 1432 janv.  6 21:31 boutique.php
-rw-rw-r--  1 zemni zemni 1520 janv.  6 21:31 connexion.php
-rw-rw-r--  1 zemni zemni 1783 janv.  6 21:31 inscription.php
drwxrwxr-x  8 zemni zemni 4096 janv.  6 21:34 .
drwxrwxr-x  4 zemni zemni 4096 janv.  6 21:31 inc
-rw-rw-r--  1 zemni zemni 3066 janv.  6 21:31 inscription.php
-rw-rw-r--  1 zemni zemni 3743 janv.  6 21:31 membres.php
-rw-rw-r--  1 zemni zemni 4110 janv.  6 21:31 panier.php
drwxrwxr-x  2 zemni zemni 4096 janv.  6 21:31 photo
-rw-rw-r--  1 zemni zemni 933 janv.  6 21:31 profil.php
-rw-rw-r--  1 zemni zemni 70  janv.  6 21:31 README.md
-rw-rw-r--  1 zemni zemni 5419 janv.  6 21:31 site.sql
zemni@jenkins:~/projet-commerce/SitePhp$ git push --force origin main
Username for 'https://github.com': Sarra1985
Password for 'https://Sarra1985@github.com':
Everything up-to-date
zemni@jenkins:~/projet-commerce/SitePhp$
```

J'ai créé les branches nécessaires pour les différents groupes

La commande ‘git branch’ est utilisée pour afficher, créer ou supprimer des branches dans un référentiel Git.

```
zemni@jenkins:~/projet-commerce/SitePhp$ git branch dev
zemni@jenkins:~/projet-commerce/SitePhp$ git checkout dev
Basculement sur la branche 'dev'
zemni@jenkins:~/projet-commerce/SitePhp$ git status
Sur la branche dev
rien à valider, la copie de travail est propre
zemni@jenkins:~/projet-commerce/SitePhp$ git log
commit f345943787f375439482c8aa5ad7e8a1eeae32ce (HEAD -> dev, origin/main, origin
/HEAD, main)
Author: Sarra1985 <110898069+Sarra1985@users.noreply.github.com>
Date:   Sat Jan 6 22:19:00 2024 +0100

    Add files via upload

commit c59e5dbd805363cdf3bd91ff9b1e6e9a9ad55179
Author: Sarra1985 <110898069+Sarra1985@users.noreply.github.com>
Date:   Sat Jan 6 22:18:23 2024 +0100

    Add files via upload
```

Et pour partager le projet sur une branche spécifique , on peut basculer sur une autre branche avec cette commande : ‘git checkout’ est utilisée pour basculer entre différentes branches et versions du code

```
zemni@jenkins:~/projet-commerce/SitePhp$ git branch ops
zemni@jenkins:~/projet-commerce/SitePhp$ git checkout ops
Basculement sur la branche 'ops'
```

**Utiliser ‘git add .’ :** Pour préparer ces modifications à la validation, utiliser la commande git add . pour ajouter toutes les modifications (y compris les nouveaux fichiers, les modifications et les suppressions) à l'index.

```
zemni@jenkins:~/projet-commerce/SitePhp$ git add .
zemni@jenkins:~/projet-commerce/SitePhp$ git log
```

**Valider les modifications :** Après avoir ajouté les modifications à l'index, utiliser la commande ‘git commit’ pour valider définitivement ces modifications dans le référentiel Git. Cela crée un nouveau commit qui enregistre les changements précédemment ajoutés à l'index.

29/01/24

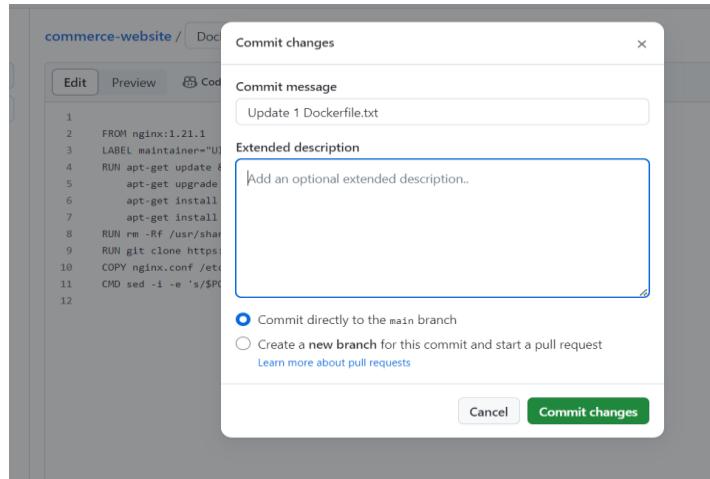
## SMART WEB AGENCY

```
zemni@jenkins:~/projet-commerce/SitePhp$ git commit -m "cree une branche"
[ops 8f2f666] cree une branche
 1 file changed, 2 insertions(+), 1 deletion(-)
zemni@jenkins:~/projet-commerce/SitePhp$ git log
commit 8f2f666e0f58657822e0148d0a3edfe9d068878d (HEAD -> ops)
Author: Sarra1985 <sarraraouf@hotmail.com>
Date:   Sat Jan 6 22:42:30 2024 +0000

  cree une branche

commit f345943787f375439482c8aa5ad7e8a1eeae32ce (origin/main, origin/HEAD, main, dev)
Author: Sarra1985 <110898069+Sarra1985@users.noreply.github.com>
Date:   Sat Jan 6 22:19:00 2024 +0100

  Add files via upload
```



J'ai cloné le projet sur mon espace personnel pour préparer un environnement de test sur Jenkins. La commande ‘git clone’ est utilisée pour créer une copie locale d'un référentiel Git distant. Cela est souvent utilisé lorsqu'en veut obtenir une copie de travail d'un projet distant pour le développement, la collaboration ou l'exploration du code.

```
zemni@jenkins:~/projet-commerce$ git clone https://github.com/Sarra1985/SitePhp
Clonage dans 'SitePhp'...
Username for 'https://github.com': Sarra1985
Password for 'https://Sarra1985@github.com':
remote: Enumerating objects: 58, done.
remote: Counting objects: 100% (11/11), done.
remote: Compressing objects: 100% (11/11), done.
remote: Total 58 (delta 0), reused 9 (delta 0), pack-reused 47
Réception d'objets: 100% (58/58), 441.87 Kio | 1012.00 Kio/s, fait.
Résolution des deltas: 100% (8/8), fait.
zemni@jenkins:~/projet-commerce$ ls
SitePhp
zemni@jenkins:~/projet-commerce$ cd
```

29/01/24

SMART WEB AGENCY

Je connecte au compte Jenkins et je rajoute mes identifiant personnel et les identifiant de GitLab et GitHub

**Bienvenue sur Jenkins !**

Vos jobs Jenkins seront affichés sur cette page. Pour commencer, vous pouvez mettre en place un build distribué ou commencer à créer un projet.

**Commencer à créer votre projet**

**Créer un job** +

**Configurer un build distribué**

Mettre en place un agent

Configurer un cloud

En apprendre plus sur les builds distribués

**File d'attente des constructions** File d'attente des constructions vide

**État du lanceur de compilations** 1. Au repos 2. Au repos

REST API Jenkins 2.426.2

**Credentials**

T	P	Store	Domain	ID	Name
System	System	(global)	Sarra1985	/*****	zemni/***** (zemni)
System	System	(global)	3d4148dd-d875-461b-b5d6-70fec1cd9150	zemni/***** (zemni)	zemni/***** (zemni)

**Stores scoped to Utilisateur : zemni**

**Stores from parent**

Je rajoute les API key de HEROKU et le password de DockerHub dans les identifiant global :

**Identifiants globaux (illimité)**

Credentials that should be available irrespective of domain specification to requirements matching.

ID	Nom	Type	Description
dockerhub_password	dockerhub_password	Secret text	dockerhub_password
heroku_api_key	heroku_api_key	Secret text	heroku_api_key

Icône: S M L

Je fais la configuration de l'environnement de test et je fais enter le URL de projet sur Git

## SMART WEB AGENCY

Global Pipeline Libraries

Sharable libraries available to any Pipeline jobs running on this system. These libraries will be trusted, meaning they run without "sandbox" restrictions and may be used in any part of a pipeline.

**Library**

Name: `sarra1985-commerce-sitephp`

Default version: `main`

Cannot validate default version until after saving and reconfiguring.

Load implicitly

Allow default version to be overridden

Include @Library changes in job recent changes

Cache fetched versions on controller for quick retrieval

Retrieval method

Modern SCM

Loads a library from an SCM plugin using newer interfaces optimized for this purpose. The recommended option when available.

Source Code Management

GitHub

Credentials: aucun

Repository HTTPS URL: `https://github.com/Sarra1985/SitePhp.git`

Je crée un pipeline pour l'environnement de test :

Jenkins

Tableau de bord > Tous >

Saisissez un nom

sitephp-test

= Champ obligatoire

Construire un projet free-style

Ceci est la fonction principale de Jenkins qui sert à builder (construire) votre projet. Vous pouvez intégrer tous les outils de gestion de version avec tous les systèmes de build. Il est même possible d'utiliser Jenkins pour tout autre chose qu'un build logiciel.

Construire un projet maven

Construit un projet avec maven. Jenkins utilise directement vos fichiers POM et diminue radicalement l'effort de configuration. Cette fonctionnalité est encore en bêta mais elle est disponible afin d'obtenir vos retours.

Pipeline

Organise des activités de longue durée qui peuvent s'étendre sur plusieurs agents de construction. Adapté pour la création des pipelines (anciennement connues comme workflows) et/ou pour organiser des activités complexes qui ne s'adaptent pas facilement à des tâches de type libre.

Et je fais la configuration de pipeline , et je rajoute le URL de projet jenkinsfile qui est sur mon compte GitHub

Jenkins

Tableau de bord > sitephp-test > Configuration

Configure

General

Description

Ce Pipeline corrige et test le projet et faire une CI/CD de l'application SitePh

Plain text Prévisualisation

Ce build a des paramètres

Do not allow concurrent builds

Do not allow the pipeline to resume if the controller restarts

GitHub project

Project url: `https://github.com/Sarra1985/commerce-website`

Sauver Apply

29/01/24

## SMART WEB AGENCY

Tableau de bord > sitephp-test > Configuration

### Configure

#### Build Triggers

- Construire après le build sur d'autres projets
- Construire dès qu'une dépendance SNAPSHOT est modifiée
- Construire périodiquement
- GitHub hook trigger for GITScm polling
- Scrutin de l'outil de gestion de version

Planning

Avez-vous vraiment dit "chaque minute" avec l'expression " \* \* \* \* "? Peut-être voulez-vous dire "H \* \* \* \*"? Aurait été lancé à dimanche 7 janvier 2024 à 10:27:19 Temps universel coordonné: prochaine exécution à dimanche 7 janvier 2024 à 10:27:19 Temps universel coordonné.

Ignore post-commit hooks

Période d'attente

Déclencher les builds à distance (Par exemple, à partir de scripts)

**Sauver** **Apply**

← → ⌂ [github.com/Sarra1985/commerce-website](https://github.com/Sarra1985/commerce-website)

Sarra1985 / commerce-website

<> Code Issues Pull requests Actions Projects Wiki Security Insights Settings

**commerce-website** Public

main 1 Branch 0 Tags  Add file <> Code

Sarra1985 Update 2 Jenkinsfile.txt 48bf911 · 38 minutes ago 4 Commits

Dockerfile.txt Update 1 Dockerfile.txt 2 hours ago

Jenkinsfile.txt Update 2 Jenkinsfile.txt 38 minutes ago

Tableau de bord > sitephp-test > Configuration

### Configure

#### Pipeline

Definition Pipeline script from SCM

SCM Git

Repositories Repository URL: https://github.com/Sarra1985/commerce-website.git

Credentials aucun Ajouter

Avancé

Add Repository

Branches to build Branch Specifier (blank for 'any'): \*/main

Add Branch

**Sauver** **Apply**

Jenkins

Tableau de bord >

+ Nouveau item Utilisateurs Historique des constructions Administer Jenkins Convert To Pipeline Mes vues Open Blue Ocean

Tous

Nom du projet : sitephp-test

Icone S M L

Legendre Atom feed for all Atom feed for failures Atom feed for just latest builds

File d'attente des constructions

File d'attente des constructions vide

## 4.2. Gérer le stockage des données

J'ai utilisé Mysql dans ce projet, MySQL est un système de gestion de base de données open source, généralement installé dans le cadre de la populaire pile LAMP (Linux, Apache, MySQL, PHP/Python/Perl). Il met en œuvre le modèle relationnel et utilise le langage de requête structuré (mieux connu sous le nom de SQL – Structured Query Language) pour gérer ses données.

Il est déjà installé sur les serveurs qu'ils vont s'occuper de la gestion de la base, notre fichier de la base de données est nommé site.sql

```
zemni@test2:~$ sudo mysql_secure_installation
[sudo] password for zemni:
Securing the MySQL server deployment.

Connecting to MySQL using a blank password.

VALIDATE PASSWORD COMPONENT can be used to test passwords
and improve security. It checks the strength of password
and allows the users to set only those passwords which are
secure enough. Would you like to setup VALIDATE PASSWORD component?

Press y|Y for Yes, any other key for No: y

There are three levels of password validation policy:

LOW   Length >= 8
MEDIUM Length >= 8, numeric, mixed case, and special characters
STRONG Length >= 8, numeric, mixed case, special characters and dictionary
        file

Please enter 0 = LOW, 1 = MEDIUM and 2 = STRONG: 2

Skipping password set for root as authentication with auth_socket is used by default.
If you would like to use password authentication instead, this can be done with the "ALTER USER" command.
See https://dev.mysql.com/doc/refman/8.0/en/alter-user.html#alter-user-password-management for more information.

By default, a MySQL installation has an anonymous user,
allowing anyone to log into MySQL without having to have
a user account created for them. This is intended only for
testing, and to make the installation go a bit smoother.
You should remove them before moving into a production
environment.

Remove anonymous users? (Press y|Y for Yes, any other key for No) : y
Success.
```

Tout d'abord, je connecte à mon serveur MySQL en tant que zemni.

```
mysql> CREATE USER 'zemni'@'localhost' IDENTIFIED BY 'Sarra22919758';
Query OK, 0 rows affected (4,08 sec)
```

```
mysql> █
```

```
zemni@test2:~$ mysql -u zemni -p
Enter password:
Welcome to the MySQL monitor. Commands end with ; or \g.
Your MySQL connection id is 9
Server version: 8.0.35-Ubuntu0.22.04.1 (Ubuntu)

Copyright (c) 2000, 2023, Oracle and/or its affiliates.

Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.
```

```
mysql> show databases;
+-----+
| Database |
+-----+
| devops   |
| information_schema |
| mysql    |
| performance_schema |
| sys      |
+-----+
5 rows in set (0,53 sec)

mysql> CREATE DATABASE site;
Query OK, 1 row affected (0,42 sec)

mysql> use site;
Database changed
    CREATE TABLE IF NOT EXISTS `site`
```

## **Création de la base de données.**

Tout d'abord, je connecte à mon serveur MySQL en tant que zemni. Après exécutez la commande suivante pour créer une base de données site

```

mysql> CREATE DATABASE site;
Query OK, 1 row affected (0,42 sec)

mysql> use site;
Database changed
mysql> CREATE TABLE IF NOT EXISTS `commande` (
    --> `id_commande` int(3) NOT NULL AUTO_INCREMENT,
    --> `id_membre` int(3) DEFAULT NULL,
    --> `montant` int(3) NOT NULL,
    --> `date_enregistrement` datetime NOT NULL,
    --> `etat` enum('en cours de traitement','envoyé','livré') NOT NULL,
    --> ) PRIMARY KEY (`id_commande`)
    --> ) ENGINE=InnoDB DEFAULT CHARSET=latin1 AUTO_INCREMENT=2 ;
Query OK, 0 rows affected, 3 warnings (0,26 sec)

mysql> INSERT INTO `commande`(`id_commande`, `id_membre`, `montant`, `date_enreg
istrement`, `etat`) VALUES
    --> (1, 4, 301, '2015-07-10 14:44:46', 'en cours de traitement');
Query OK, 1 row affected (0,25 sec)

mysql> CREATE TABLE IF NOT EXISTS `details_commande` (
    --> `id_details_commande` int(3) NOT NULL AUTO_INCREMENT,
    --> `id_commande` int(3) DEFAULT NULL,
    --> `id_produit` int(3) DEFAULT NULL,
    --> `quantite` int(3) NOT NULL,
    --> `prix` int(3) NOT NULL,
    --> ) PRIMARY KEY (`id_details_commande`)
    --> ) ENGINE=InnoDB DEFAULT CHARSET=latin1 AUTO_INCREMENT=4 ;
Query OK, 0 rows affected, 5 warnings (1,82 sec)

mysql> INSERT INTO `details_commande`(`id_details_commande`, `id_commande`, `id_
produit`, `quantite`, `prix`) VALUES
    --> (1, 1, 2, 1, 15),
    --> (2, 1, 6, 1, 49),
    --> (3, 1, 8, 3, 79);
Query OK, 3 rows affected (0,01 sec)
Records: 3 Duplicates: 0 Warnings: 0

```

```
mysql> show tables;
+-----+
| Tables_in_site |
+-----+
| commande      |
| details_commande |
| membre        |
| produit       |
+-----+
4 rows in set (0,11 sec)

mysql>
```

Cette base, va être l'utiliser pour le site de commerce qu'elle contient 4 tables :

id_produit	reference	categorie	titre	description	couleur	taille	public	photo
d-23_bleu.jpg	1	11-d-23	tshirt	Tshirt Col V 20   53	Tee-shirt en coton flammé liseré contrastant	bleu	M	m   /dev/tutomooc/site/photo/11-
f-15_rouge.png	2	66-f-15	tshirt	Tshirt Col V rouge 15   230	c'est vraiment un super tshirt en soir&eacute; !	rouge	L	m   /dev/tutomooc/site/photo/66-
g-77_vert.png	3	88-g-77	tshirt	Tshirt Col rond vert 29   63	Il vous faut ce tshirt Made In France !!!	vert	L	m   /dev/tutomooc/site/photo/88-
b-38_jaune.png	4	55-b-38	tshirt	Tshirt jaune 20   3	le jaune reviens &grave; la mode, non? :-)	jaune	S	m   /dev/tutomooc/site/photo/55-
p-33_noir.jpg	5	31-p-33	tshirt	Tshirt noir original 25   80	voici un tshirt noir tr&egrave;s original :p	noir	XL	m   /dev/tutomooc/site/photo/31-
a-65_chemiseblanche.jpg	6	56-a-65	chemise	Chemise Blanche 49   73	Les chemises c'est bien mieux que les tshirts	blanc	L	m   /dev/tutomooc/site/photo/56-
s-63_chemisenoir.jpg	7	63-s-63	chemise	Chemise Noir 59   120	Comme vous pouvez le voir c'est une chemise noir...	noir	M	m   /dev/tutomooc/site/photo/63-
p-79_pullgris2.jpg	8	77-p-79	pull	Pull gris 79   99	Pull gris pour l'hiver	gris	XL	f   /dev/tutomooc/site/photo/77-

### 4.3. Gérer des containers

La création de fichiers Dockerfile pour construire des images, puis l'utilisation de ces images dans les fichiers de configuration Kubernetes pour déployer des conteneurs. Dockerfile pour la configuration d'Apache PHP mysql

```
GNU nano 6.2                               Dockerfile
FROM php:7.3.3-apache

# Install your extensions to connect to MySQL and add mysqli
RUN docker-php-ext-install mysqli
# Install pdo is you need to use PHP PDO

RUN docker-php-ext-enable mysqli
```

Docker-compose :

```
version: '3.3'
services:
  web:
    build:
      context: ./www
      dockerfile: Dockerfile
      container_name: php73
      restart: always
    ports:
      - "8080:80"
    volumes:
      - ./www:/var/www/html/
    depends_on:
      - db
    networks:
      - default
  db:
    image: mysql:8.0
    container_name: mysql_c
    command: --default-authentication-plugin=mysql_native_password
    restart: always
    environment:
      MYSQL_ROOT_PASSWORD: root
      MYSQL_DATABASE: magasin_db
      MYSQL_USER: zemni
      MYSQL_PASSWORD: Reponse
    ports:
      - "3306:3306"
    volumes:
      - ./magasin_db.sql:/docker-entrypoint-initdb.d/magasin_db.sql
      - persistent:/var/lib/mysql
volumes:
  persistent:
```

```
GNU nano 6.2                               index.php
<?php
/*Code php écrit du 29 Avril au 01 Mai 2021 à N'djaména au Tchad par
TARGOTO Christian
Contact: ct@chrislink.net / 23560316682
*/
?>
<?php
$servername = "mysql_c";
$username = "zemni";
$password = "Reponse";
$dbname = "magasin_db";
// Create connection
$conn = mysqli_connect($servername, $username, $password, $dbname);
// Check connection
if (!$conn) {
die("Connection failed: " . mysqli_connect_error());
}

?>
<?php
$mess1="";
$mess2="";
$nom=@$_POST["nom"];
$quantite=@$_POST["quantite"];
$datexp=@$_POST["datexp"];
$numero=@$_POST["numero"];
$id=@$_POST["id"];

$nom2=@$_POST["nom2"];
$quantite2=@$_POST["quantite2"];
$datexp2=@$_POST["datexp2"];
$numero2=@$_POST["numero2"];

?>
<?php
//enregister les depots de produits
if(isset($_POST['benrg'])&&!empty($nom)&&!empty($quantite)&&!empty($datexp)&&!empty($numero))
{
$sql=mysql_query($conn, "insert into tb_action(nom_prod,qte_prod,date_exp,date_expiration) values('$nom','$quantite','$datexp','$numero')");
}
```

Construction et poussée des images Utilisez les commandes suivantes dans les répertoires appropriés de votre projet :

```
zemni@worker1:~$ sudo docker ps
CONTAINER ID        IMAGE               COMMAND             CREATED            STATUS              PORTS                               NAMES
40b7de04ef49        sitephp-web        "docker-php-entrypoi..."   4 hours ago       Up About an hour   0.0.0.0:8080->80/tcp, ::8080->80/tcp   php73
038dd7b4fc2c        mysql:8.0          "docker-entrypoint.s..."   4 hours ago       Up About an hour   0.0.0.0:3306->3306/tcp, ::3306->3306/tcp, 33060/tcp   mysql_c
zemni@worker1:~$
```

```
zemni@worker1:~$ sudo docker images
REPOSITORY      TAG      IMAGE ID      CREATED      SIZE
sitephp-web     latest   3909c424c200  8 hours ago  378MB
mysql           8.0      ba048db12589  3 weeks ago  591MB
zemni@worker1:~$
```

```
zemni@worker1:~$ sudo docker volume ls
DRIVER      VOLUME NAME
local        sitephp_persistent
zemni@worker1:~$
```

## Formulaire d'enregistrement des dépôts de produits

<b>DEPOT</b>	<b>Nom produit</b>	<input type="text"/>
	<b>Quantité produit</b>	<input type="text"/>
	<b>Date expiration</b>	<input type="text"/>
	<b>Numéro magasin</b>	<input type="text"/>
		<b>ENREGISTRER</b>
	<input type="text"/> Identifiant	<b>SUPPRIMER</b>

Formulaire d'enregistrement des retraits de produits

<b>RETRAIT</b>	<b>Nom produit</b>	<input type="text"/>
	<b>Quantité produit</b>	<input type="text"/>
	<b>Date expiration</b>	<input type="text"/>
	<b>Numéro magasin</b>	<input type="text"/>
	<b>ENREGISTRER</b>	

Quantité en stock par produit
Quantité en stock par produit et par date d'expiration
Quantité en stock par magasin et par produit
Quantité en stock par magasin par produit et par date d'expiration

#### Liste des produits déposer ou retirer dans le stock

Nom produit	Quantité	Date expiration	Numéro magasin	Nature action	Date action	Identifiant
SUCRE EN PAQUET	30	18/11/2022	1	DEPOT	2021-04-24 09:43:00	9
SUCRE EN PAQUET	70	18/11/2022	1	RETRAIT	2021-04-24 09:30:51	8
SUCRE EN PAQUET	230	22/11/2023	2	DEPOT	2021-04-24 09:00:23	7
SUCRE EN POUDRE SAC DE 25 KG	370	19/11/2023	2	DEPOT	2021-04-24 08:58:57	6
FARINE DE BLE SAC DE 50 KG	300	05/09/2023	2	DEPOT	2021-04-24 08:57:03	5
RIZ SAC DE 50 KG	220	12/06/2024	1	DEPOT	2021-04-24 08:54:57	4
HUILE EN BIDON DE 25L	180	10/05/2022	1	DEPOT	2021-04-24 08:53:33	3
SUCRE EN PAQUET	4800	18/11/2022	1	DEPOT	2021-04-24 07:53:58	1

#### 4.4. Automatiser la mise en production d'une application avec une plateforme

J'ai automatisé l'installation de docker et Kubernetes avec Ansible

C'est l'arborescence de dossier d'installation,

```
zemni@ansible:~$ ls
ansible.cfg  install-ansible.sh  kube-projet  pod_network_setup.txt
zemni@ansible:~$ cd kube-projet
zemni@ansible:~/kube-projet$ ls
hosts  kube-dependencies.yml  master.yml  workers.yml
zemni@ansible:~/kube-projet$ ls -la
total 24
drwxrwxr-x 2 zemni zemni 4096 Oct 25 15:17 .
drwxr-x--- 7 zemni zemni 4096 Jan  7 15:12 ..
-rw-rw-r-- 1 zemni zemni  914 Oct 25 11:38 hosts
-rw-rw-r-- 1 zemni zemni 1168 Oct 25 13:56 kube-dependencies.yml
-rw-rw-r-- 1 zemni zemni  921 Oct 25 15:11 master.yml
-rw-rw-r-- 1 zemni zemni  496 Oct 25 15:17 workers.yml
```

Le fichier hosts qui contient toutes les informations de master et des workers pour l'automatisation :

```
zemni@ansible:~/kube-projet$ cat hosts
[all]
master ansible_host=192.168.163.136 ansible_user=zemni ansible_password=Reponse  ansible_sudo_pass=Reponse  ansible_sudo_password=Reponse
worker1 ansible_host=192.168.163.139 ansible_user=zemni ansible_password=Reponse  ansible_sudo_pass=Reponse  ansible_sudo_password=Reponse
worker2 ansible_host=192.168.163.138 ansible_user=zemni ansible_password=Reponse  ansible_sudo_pass=Reponse  ansible_sudo_password=Reponse

[masters]
master ansible_host=192.168.163.136 ansible_user=zemni ansible_password=Reponse  ansible_sudo_pass=Reponse  ansible_sudo_password=Reponse

[workers]
worker1 ansible_host=192.168.163.139 ansible_user=zemni ansible_password=Reponse  ansible_sudo_pass=Reponse  ansible_sudo_password=Reponse
worker2 ansible_host=192.168.163.138 ansible_user=zemni ansible_password=Reponse  ansible_sudo_pass=Reponse  ansible_sudo_password=Reponse

[all:vars]
ansible_python_interpreter=/usr/bin/python3
zemni@ansible:~/kube-projet$ cat kube-dependencies.yml
```

```
ans@centos8_1:~/kubescripts$ cat kube-dependencies.yml
---
- hosts: all
  become: yes
  tasks:
    - name: create Docker config directory
      file:
        path: /etc/docker
        state: directory

    - name: changing Docker to systemd driver
      copy:
        dest: "/etc/docker/daemon.json"
        content: |
          {
            "exec-opts": ["native.cgroupdriver=systemd"]
          }

    - name: install Docker
      apt:
        name: docker.io
        state: present
        update_cache: true

    - name: install APT Transport HTTPS
      apt:
        name: apt-transport-https
        state: present

    - name: add Kubernetes apt-key
      apt_key:
        url: https://packages.cloud.google.com/apt/doc/apt-key.gpg
        state: present

    - name: add Kubernetes' APT repository
      apt_repository:
        repo: deb http://apt.kubernetes.io/ kubernetes-xenial main
        state: present
        filename: 'kubernetes'
```

```
- name: install kubelet
  apt:
    name: kubelet
    state: present
    update_cache: true

- name: install kubeadm
  apt:
    name: kubeadm
    state: present

- hosts: masters
  become: yes
  tasks:
    - name: install kubectl
      apt:
        name: kubectl
        state: present
        force: yes
```

**Le fichier kube-dependencies.yml il depend de l'installation sur toute les VM.  
Le fichier master.yml s'occupe des installation et configuration que du master**

```
zemni@ansibile:~/kube-projet$ cat master.yml
---
- hosts: masters
  become: yes
  tasks:
    - name: initialize the cluster
      shell: kubeadm init --control-plane-endpoint=192.168.163.136 --pod-network-cidr=192.168.0.0/16 >> cluster_initialized.txt

    args:
      chdir: $HOME
      creates: cluster_initialized.txt

    - name: create .kube directory
      become: yes
      become_user: zemni
      file:
        path: $HOME/.kube
        state: directory
        mode: 0755

    - name: copy admin.conf to user's kube config
      copy:
        src: /etc/kubernetes/admin.conf
        dest: /home/zemni/.kube/config
        remote_src: yes
        owner: zemni

    - name: install Pod network
      become: yes
      become_user: zemni
      shell: kubectl apply -f https://raw.githubusercontent.com/coreos/flannel/master/Documentation/kube-flannel.yml >> pod_network_setup.txt
      args:
        chdir: $HOME
        creates: pod_network_setup.txt
zemni@ansibile:~/kube-projet$ cat workers.yml
```

Le fichier worker.yml est

```
---
- hosts: masters
  become: yes
  gather_facts: false
  tasks:
    - name: get join command
      shell: kubeadm token create --print-join-command
      register: join_command_raw

    - name: set join command
      set_fact:
        join_command: "{{ join_command_raw.stdout_lines[0] }}"

- hosts: workers
  become: yes
  tasks:
    - name: join cluster
      shell: "{{ hostvars['master'].join_command }} >> node_joined.txt"
      args:
        chdir: $HOME
        creates: node_joined.txt
|
```

```

Oct 20 19:30:46 ansible sshd[1397]: Accepted password for zemni from 192.168.163.1 port 51121 ssh2
Oct 20 19:30:46 ansible sshd[1397]: pam_unix(sshd:session): session opened for user zemni(uid=1000) by (uid=0)
zemni@ansible:~$ ssh-keygen
Generating public/private rsa key pair.
Enter file in which to save the key (/home/zemni/.ssh/id_rsa):
/home/zemni/.ssh/id_rsa already exists.
Overwrite (y/n)? y
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /home/zemni/.ssh/id_rsa
Your public key has been saved in /home/zemni/.ssh/id_rsa.pub
The key fingerprint is:
SHA256:coexX+YFZtvrbg+wuSwdjpwD1Q/gZ402QzdlM7YTymgo zemni@ansible
The key's randomart image is:
+---[RSA 3072]----+
|          o0*+ |
|          +#*oo |
|          +.++.. |
|          . S X .o |
|          oE+ @o |
|          .=oR. |
|          .o+o+. |
|          oB*... |
+---[SHA256]----+
zemni@ansible:~$ ssh-copy-id zemni@192.168.163.131
/usr/bin/ssh-copy-id: INFO: Source of key(s) to be installed: "/home/zemni/.ssh/id_rsa.pub"
/usr/bin/ssh-copy-id: INFO: attempting to log in with the new key(s), to filter out any that are already installed
/usr/bin/ssh-copy-id: INFO: 1 key(s) remain to be installed -- if you are prompted now it is to install the new keys
zemni@192.168.163.131's password:
Number of key(s) added: 1

Now try logging into the machine, with: "ssh 'zemni@192.168.163.131'"
and check to make sure that only the key(s) you wanted were added.

zemni@ansible:~$ ssh-copy-id zemni@192.168.163.132
/usr/bin/ssh-copy-id: INFO: Source of key(s) to be installed: "/home/zemni/.ssh/id_rsa.pub"
/usr/bin/ssh-copy-id: INFO: attempting to log in with the new key(s), to filter out any that are already installed
/usr/bin/ssh-copy-id: INFO: 1 key(s) remain to be installed -- if you are prompted now it is to install the new keys
zemni@192.168.163.132's password:

```

```

zemni@ansible:~/kube-projet$ nano hosts
zemni@ansible:~/kube-projet$ ansible-playbook -i hosts kube-dependencies.yml
PLAY [all] ****
TASK [Gathering Facts] ****
ok: [worker2]
ok: [master]
ok: [worker1]

TASK [Create Docker config directory] ****
changed: [worker2]
changed: [master]
changed: [worker1]

TASK [Changing Docker to systemd driver] ****
changed: [worker1]
changed: [master]
changed: [worker2]

TASK [Uninstall Docker] ****
changed: [master]
changed: [worker2]
changed: [worker1]

TASK [Install API Transport HTTPS] ****
changed: [worker2]
changed: [worker1]
changed: [master]

TASK [Add Kubernetes apt-key] ****
changed: [worker1]
changed: [worker2]
changed: [master]

TASK [Add Kubernetes' APT repository] ****
changed: [master]
changed: [worker2]
changed: [worker1]

```

```

TASK [Install kubelet] ****
ok: [worker2]
ok: [master]
ok: [worker1]

TASK [Install kubeadm] ****
ok: [worker2]
ok: [worker1]
ok: [master]

PLAY [masters] ****
TASK [Gathering Facts] ****
ok: [master]

TASK [Install kubectl] ****
ok: [master]

PLAY RECAP ****
master      : ok=11   changed=0    unreachable=0    failed=0    skipped=0    rescued=0    ignored=0
worker1     : ok=9    changed=0    unreachable=0    failed=0    skipped=0    rescued=0    ignored=0
worker2     : ok=9    changed=0    unreachable=0    failed=0    skipped=0    rescued=0    ignored=0

```

Les étapes générales pour passer de Docker Compose à Kubernetes pour la mise en production :

1. Définir des manifestes Kubernetes : Créez des fichiers de configuration Kubernetes (manifestes) pour décrire les services, les déploiements, les secrets, les volumes persistants, etc. nécessaires à votre application.
2. Configurer les ressources de stockage : Si votre application utilise des bases de données, assurez-vous de configurer correctement les ressources de stockage persistant dans Kubernetes pour assurer la persistance des données.
3. Déployer sur Kubernetes : Utilisez les outils Kubernetes (kubectl, Helm, etc.) pour déployer vos applications et services sur un cluster Kubernetes.

4. Configurer les services Kubernetes : Configurez les services Kubernetes pour exposer votre application au trafic externe si nécessaire. Vous pouvez utiliser des services de type LoadBalancer, NodePort, ou Ingress, selon vos besoins.
5. Gérer les secrets : Utilisez Kubernetes Secrets pour gérer les informations sensibles telles que les informations d'identification de la base de données.
6. Surveiller et ajuster : Utilisez les outils de surveillance Kubernetes pour suivre les performances de votre application et ajustez les ressources en conséquence.

Il est important de noter que la transition de Docker Compose à Kubernetes peut nécessiter des ajustements dans la manière dont vous définissez les services, les réseaux et les volumes, car les concepts et les abstractions sont différents entre les deux. Cependant, Kubernetes offre une mise en production plus évolutive et robuste pour les applications distribuées.

Pour un déploiement PHP et MySQL dans Kubernetes :

### **1. Deployment pour PHP (deployment-php.yaml) :**

```
GNU nano 6.2                                     deployment-php.yaml
apiVersion: apps/v1
kind: Deployment
metadata:
  name: php-deployment
spec:
  replicas: 3
  selector:
    matchLabels:
      app: php
  template:
    metadata:
      labels:
        app: php
    spec:
      containers:
        - name: php73
          image: sitephp-web
          ports:
            - containerPort: 8080
```

Service pour PHP (service-php.yaml) :

```
GNU nano 6.2                                     service-php.yaml
apiVersion: v1
kind: Service
metadata:
  name: php-service
spec:
  selector:
    app: php
  ports:
    - protocol: TCP
      port: 8080
      targetPort: 8080
  type: LoadBalancer # Change to NodePort or ClusterIP as needed
```

Deployment pour MySQL (deployment-mysql.yaml) :

```
GNU nano 6.2                                     deployment-mysql.yaml
apiVersion: apps/v1
kind: Deployment
metadata:
  name: mysql-deployment
spec:
  replicas: 1
  selector:
    matchLabels:
      app: mysql
  template:
    metadata:
      labels:
        app: mysql
    spec:
      containers:
        - name: mysql-container
          image: mysql:8.0
          env:
            - name: MYSQL_ROOT_PASSWORD
              value: root
            - name: MYSQL_DATABASE
              value: magasin_db
            - name: MYSQL_USER
              value: zemni
            - name: MYSQL_PASSWORD
              value: Reponse
          ports:
            - containerPort: 3306
          env:
            # Ajoutez vos variables d'environnement ici
        volumeMounts:
          - name: mysql-persistent-storage
            mountPath: /var/lib/mysql
        volumes:
          - name: mysql-persistent-storage
            emptyDir: {}
```

Service pour MySQL (service-mysql.yaml) :

```
GNU nano 6.2                                     service-mysql.yaml
apiVersion: v1
kind: Service
metadata:
  name: mysql-service
spec:
  selector:
    app: mysql
  ports:
    - protocol: TCP
      port: 3306
      targetPort: 3306
  type: LoadBalancer  # Change to ClusterIP LoadBal
```

Pour configurer correctement les ressources de stockage persistant dans Kubernetes, vous avez besoin d'un fichier YAML décrivant les ressources de stockage persistant (Persistent Volume et Persistent Volume Claim).

```
GNU nano 6.2                                     persistent-volume.yaml
# persistent-volume.yaml

apiVersion: v1
kind: PersistentVolume
metadata:
  name: mysql-app-pv
spec:
  capacity:
    storage: 10Gi  # Ajustez la capacité en fonction de
  accessModes:
    - ReadWriteOnce
  hostPath:
    path: "/data"  # Choisir un chemin approprié sur le
```

```
GNU nano 6.2                                     persistent-volume-claim.yaml
apiVersion: v1
kind: PersistentVolumeClaim
metadata:
  name: mysql-app-pvc
spec:
  accessModes:
    - ReadWriteOnce
  resources:
    requests:
      storage: 10Gi  # Ajustez la capacité en fonction de
```

Le fichier YAML pour définir un autoscaling dans Kubernetes utilise généralement un objet **HorizontalPodAutoscaler** (HPA). Voici un exemple simple de fichier YAML pour un HPA :

```
GNU nano 6.2                                     autoscaling.yaml
apiVersion: autoscaling/v1
kind: HorizontalPodAutoscaler
metadata:
  name: app-autoscaler
spec:
  scaleTargetRef:
    apiVersion: apps/v1
    kind: Deployment
    name: php-deployment
  minReplicas: 2
  maxReplicas: 5
  targetCPUUtilizationPercentage: 80
```

29/01/24

SMART WEB AGENCY

```
zemni@worker1:~/k8s-projet$ tree
.
├── autoscaling.yaml
├── deployment-mysql.yaml
├── deployment-php.yaml
├── persistent-volume-claim.yaml
├── persistent-volume.yaml
└── service-mysql.yaml
    └── service-php.yaml

0 directories, 7 files
zemni@worker1:~/k8s-projet$
```

Je passe à l'étape suivante , c'est la création de namespace

```
zemni@master:~$ kubectl create namespace m2ins
zemni@master:~$ kubectl create secret -n m2ins
```

Kubectl get po -A

```
#output
kube-flannel kube-flannel-ds-zjnqh      1/1   Running  0      21s
kube-system coredns-5dd5756b68-cpj9      1/1   Running  0      6m54s
kube-system coredns-5dd5756b68-ddxkb      1/1   Running  0      6m54s
kube-system etcd-master.m2i.com          1/1   Running  0      7m12s
kube-system kube-apiserver-master.m2i.com 1/1   Running  0      7m15s
kube-system kube-controller-manager-master.m2i.com 1/1   Running  0      7m16s
kube-system kube-proxy-75dbj            1/1   Running  0      6m54s
kube-system kube-scheduler-master.m2i.com 1/1   Running  0      7m12s
```

#Print token

```
export KUBECONFIG=/etc/kubernetes/admin.conf
```

```
sudo kubeadm token create --print-join-command
```

```
sudo kubeadm join 10.0.0.13:6443 --token q4axum.3rn8ua2lhk9cng01 --discovery-token-ca-cert-hash
sha256:360384b8bf0f665602f171c3f1da691cc2c00d811bfa6816b4a9de1fe24325a4
```

\*\*\*\*\*dans master\*\*\*\*\*

```
kubectl get nodes
```

```
kubectl get nodes -o wide
```

```
zemni@master:~$ kubectl get nodes -o wide
```

NAME	STATUS	ROLES	AGE	VERSION	INTERNAL-IP	EXTERNAL-IP	OS-IMAGE	KERNEL-VERSION	CONTAINER-RUNTIME
master.m2i.com	Ready	control-plane	6h25m	v1.28.2	192.168.163.141	<none>	Ubuntu 22.04.3 LTS	5.15.0-88-generic	containerd://1.7.2
worker1.m2i.com	Ready	<none>	96m	v1.28.2	192.168.163.140	<none>	Ubuntu 22.04.3 LTS	5.15.0-88-generic	containerd://1.7.2
worker2.m2i.com	Ready	<none>	61m	v1.28.2	192.168.163.142	<none>	Ubuntu 22.04.3 LTS	5.15.0-88-generic	containerd://1.7.2

```
zemni@master:~$ sudo kubeadm certs check-expiration
```

[sudo] password for zemni:

[check-expiration] Reading configuration from the cluster...

[check-expiration] FYI: You can look at this config file with 'kubectl -n kube-system get cm kubeadm-config -o yaml'

CERTIFICATE	EXPIRES	RESIDUAL TIME	CERTIFICATE AUTHORITY	EXTERNALLY MANAGED
admin.conf	Oct 31, 2024 16:41 UTC	364d	ca	no
apiserver	Oct 31, 2024 16:41 UTC	364d	ca	no
apiserver-etcd-client	Oct 31, 2024 16:41 UTC	364d	etcd-ca	no
apiserver-kubelet-client	Oct 31, 2024 16:41 UTC	364d	ca	no
controller-manager.conf	Oct 31, 2024 16:41 UTC	364d	ca	no
etcd-healthcheck-client	Oct 31, 2024 16:41 UTC	364d	etcd-ca	no
etcd-peer	Oct 31, 2024 16:41 UTC	364d	etcd-ca	no

```

etcctl-server      Oct 31, 2024 16:41 UTC 364d      etcd-ca      no
front-proxy-client Oct 31, 2024 16:41 UTC 364d      front-proxy-ca  no
scheduler.conf     Oct 31, 2024 16:41 UTC 364d      ca          no

CERTIFICATE AUTHORITY EXPIRES             RESIDUAL TIME EXTERNALLY MANAGED
ca          Oct 29, 2033 16:41 UTC 9y        no
etcd-ca     Oct 29, 2033 16:41 UTC 9y        no
front-proxy-ca Oct 29, 2033 16:41 UTC 9y       no

```

Utilisation de kubectl pour déployer les ressources directement :

- Assurez-vous que votre cluster Kubernetes est prêt et que kubectl est configuré pour s'y connecter.
- Appliquez les fichiers de configuration YAML pour les déploiements, services et ressources de stockage persistant :

```

zemni@worker1:~/k8s-projet$ kubectl apply -f persistent-volume.yaml
persistentvolume/mysql-app-pv created
zemni@worker1:~/k8s-projet$ kubectl apply -f persistent-volume-claim.yaml
persistentvolumeclaim/mysql-app-pvc created
zemni@worker1:~/k8s-projet$ kubectl apply -f deployment-php.yaml
deployment.apps/php-deployment created
zemni@worker1:~/k8s-projet$ kubectl apply -f service-php.yaml
service/php-service created
zemni@worker1:~/k8s-projet$ kubectl apply -f deployment-mysql.yaml

```

```

zemni@worker1:~/k8s-projet$ kubectl apply -f deployment-mysql.yaml
deployment.apps/mysql-deployment created

```

```

zemni@worker1:~/k8s-projet$ kubectl apply -f service-mysql.yaml
service/mysql-service created

```

```

zemni@worker1:~/k8s-projet$ kubectl apply -f autoscaling.yaml
horizontalpodautoscaler.autoscaling/app-autoscaler created

```

```

zemni@worker1:~/k8s-projet$ kubectl get pod -o wide
NAME                               READY   STATUS    RESTARTS   AGE
mysql-deployment-7dbdf55595-rgc4m  0/1    Pending   0          21m
php-deployment-5d94ff8d6-64fh     0/1    Pending   0          57m
php-deployment-5d94ff8d6-grpxj    0/1    Pending   0          57m
php-deployment-5d94ff8d6-xdq14   0/1    Pending   0          57m

```

```

zemni@worker1:~/k8s-projet$ kubectl get svc
NAME         TYPE        CLUSTER-IP   EXTERNAL-IP  PORT(S)   AGE
kubernetes   ClusterIP  10.96.0.1   <none>        443/TCP   65m
mysql-service LoadBalancer 10.101.107.192 <pending>    3306:31278/TCP 18m
php-service   LoadBalancer 10.103.128.154 <pending>    8080:31175/TCP 59m

```

## 5. Superviser les services déployés

### 5.1. Définir et mettre en place des statistiques de services

Dans le cadre de l'optimisation de notre infrastructure informatique, nous cherchons à définir et mettre en place des statistiques de services pour la supervision Nagios. Cette initiative vise à renforcer notre capacité de surveillance des services déployés, assurant ainsi une gestion proactive et efficace de notre environnement informatique. La collaboration avec l'équipe de travail sera cruciale pour garantir le succès de cette démarche.

**Objectifs :**

1. **Définir les Statistiques de Services :** Élaborez une liste détaillée des services critiques et non critiques qui nécessitent une surveillance constante. Identifiez les paramètres clés à surveiller pour chaque service afin de garantir des performances optimales.
2. **Mise en Place de Nagios :** Assurez-vous que Nagios est correctement déployé et configuré pour collecter les données pertinentes. Configurez les notifications pour les événements critiques et mettez en place un tableau de bord intuitif pour une visualisation facile des statistiques de services.
3. **Collaboration avec l'Équipe de Travail :** Impliquez l'équipe de travail dans le processus en identifiant les responsables de chaque service. Établissez des protocoles clairs pour la communication des incidents et définissez des seuils d'alerte appropriés.

### Étapes de Mise en Place :

1. **Analyse des Besoins :** Rencontre avec les parties prenantes pour discuter des exigences spécifiques en matière de surveillance des services.
2. **Configuration de Nagios :** Paramétrage des hôtes, services, et notifications dans Nagios, en tenant compte des spécifications définies.
3. **Formation de l'Équipe :** Organisez des sessions de formation pour l'équipe de travail sur l'utilisation de Nagios et la compréhension des statistiques de services.
4. **Tests et Validation :** Effectuez des tests approfondis pour vous assurer que les statistiques sont correctement collectées et que les notifications sont fonctionnelles.
5. **Documentation :** Créez une documentation détaillée sur la configuration de Nagios, les statistiques de services, et les procédures d'intervention en cas d'incident.

La définition et la mise en place de statistiques de services pour la supervision Nagios représentent une étape cruciale pour renforcer la stabilité et les performances de notre infrastructure informatique. La collaboration étroite avec l'équipe de travail garantira une gestion proactive des services déployés, minimisant ainsi les temps d'indisponibilité et assurant une expérience utilisateur optimale.

Les éléments d'infrastructure que nous pourrions superviser avec Nagios. On a noté que les besoins spécifiques peuvent varier en fonction de l'environnement et des objectifs. Ces exemples peuvent servir de point de départ :

1. **Serveurs :**
  - Utilisation du processeur
  - Utilisation de la mémoire RAM
  - Utilisation de l'espace disque
  - État des services essentiels (par exemple, Apache, MySQL, SSH)
2. **Réseau :**
  - Latence réseau
  - Utilisation de la bande passante
  - Erreurs sur les interfaces réseau
  - État des connexions VPN
3. **Stockage :**
  - Espace disque disponible sur chaque serveur
  - État des disques durs (SMART)
  - Performances des systèmes de stockage partagé
4. **Base de données :**
  - Connexions à la base de données
  - Temps de réponse des requêtes
  - Utilisation de la mémoire par la base de données
  - Disponibilité des bases de données
5. **Sécurité :**

- Détection d'intrusion
- Statut des correctifs de sécurité
- Journalisation des événements

#### 6. Applications Web :

- Temps de réponse des pages web
- Disponibilité des services web
- Erreurs HTTP

#### 7. Services Cloud :

- Utilisation des ressources cloud (CPU, RAM, stockage)
- Statut des instances
- Surveillance des services cloud spécifiques (par exemple, AWS EC2, Azure VM)

#### 8. Services Système :

- État des services système (par exemple, cron, syslog)
- Statut des tâches planifiées
- Utilisation du système de fichiers

#### 9. Équipements Réseau :

- État des routeurs et commutateurs
- Latence sur les liaisons réseau
- Surveillance des ports réseau

#### 10. Équipements Physiques :

- Température des serveurs
- État des alimentations électriques
- État des ventilateurs

Ces exemples couvrent un large éventail d'aspects de l'infrastructure informatique. L'idée est d'identifier les points critiques qui peuvent impacter les performances, la disponibilité, et la sécurité de votre environnement, et de configurer Nagios en conséquence pour superviser ces aspects.

### 5.2. Exploiter une solution de supervision

Dans la machine principale, celle qu'elle va superviser les autres machines et leur infrastructure.

```
zemm1@nagios1:~$ sudo systemctl status nagios
● nagios.service - Nagios Core 4.4.6
   Loaded: loaded (/lib/systemd/system/nagios.service; enabled; vendor preset: enabled)
   Active: active (running) since Thu 2023-11-30 19:17:18 UTC; 3min 11s ago
     Docs: https://www.nagios.org/documentation
 Process: 2825 ExecStartPre=/usr/local/nagios/bin/nagios -v /usr/local/nagios/etc/nagios.cfg (code=exited, status=0/SUCCESS)
 Process: 2826 ExecStart=/usr/local/nagios/bin/nagios -d /usr/local/nagios/etc/nagios.cfg (code=exited, status=0/SUCCESS)
 Main PID: 2827 (nagios)
   Tasks: 6 (limit: 970)
  Memory: 2.0M
    CPU: 832ms
   CGroup: /system.slice/nagios.service
           └─2827 /usr/local/nagios/bin/nagios -d /usr/local/nagios/etc/nagios.cfg
             ├─2828 /usr/local/nagios/bin/nagios --worker /usr/local/nagios/var/rw/nagios.gh
             ├─2829 /usr/local/nagios/bin/nagios --worker /usr/local/nagios/var/rw/nagios.gh
             ├─2830 /usr/local/nagios/bin/nagios --worker /usr/local/nagios/var/rw/nagios.gh
             ├─2831 /usr/local/nagios/bin/nagios --worker /usr/local/nagios/var/rw/nagios.gh
             └─2832 /usr/local/nagios/bin/nagios -d /usr/local/nagios/etc/nagios.cfg

nov. 30 19:17:18 nagios1 nagios[2827]: qh: Socket '/usr/local/nagios/var/rw/nagios.gh' successfully initialized
nov. 30 19:17:18 nagios1 nagios[2827]: qh: core query handler registered
nov. 30 19:17:18 nagios1 nagios[2827]: qh: echo service query handler registered
nov. 30 19:17:18 nagios1 nagios[2827]: qh: help for the query handler registered
nov. 30 19:17:18 nagios1 nagios[2827]: wproc: Successfully registered manager as @wproc with query handler
nov. 30 19:17:18 nagios1 nagios[2827]: wproc: Registry request: name=Core Worker 2830;pid=2830
nov. 30 19:17:18 nagios1 nagios[2827]: wproc: Registry request: name=Core Worker 2829;pid=2829
nov. 30 19:17:18 nagios1 nagios[2827]: wproc: Registry request: name=Core Worker 2828;pid=2828
nov. 30 19:17:18 nagios1 nagios[2827]: wproc: Registry request: name=Core Worker 2831;pid=2831
nov. 30 19:17:19 nagios1 nagios[2827]: Successfully launched command file worker with pid 2832
```

On connecte au site <https://localhost/nagios> et on entre le nom d'utilisateur et le mot de passe

29/01/24

SMART WEB AGENCY

① 192.168.163.156/nagios

**Se connecter**

http://192.168.163.156  
Votre connexion à ce site n'est pas privée

Nom d'utilisateur

Mot de passe

**Se connecter** **Annuler**

Non sécurisé | 192.168.163.156/nagios/

### Nagios®

Current Network Status

Last Update: Thu Nov 21 15:24 UTC 2023  
Last updated every 30 seconds

Nagios® Core™ 4.6.6 - www.nagios.org

Logins

Logins History For all hosts

View Notifications For All Hosts

View Host Status Detail For All Hosts

Current Status

Tactical Overview Map (Legacy)

Hosts

Services

Host Groups

Summary Grid

Service Groups

Summary Grid

Problems

Services (Unhandled)

Hosts (Unhandled)

Network Outages

Quick Search:

Host Status Totals		Service Status Totals	
Up	Down	Unreachable	Pending
1	0	0	0
All Problems	All Types	All Problems	All Types
0	1	0	0

**Service Status Details For All Hosts**

Host	Service	Status	Last Check	Duration	Attempt	Status Information
localhost	Current Load	OK	11-30-2023 21:47:16	0d 1h 40m 8s	1/4	OK - Charge moyenne: 0.25, 0.22, 0.25
localhost	Current Users	OK	11-30-2023 21:47:55	0d 1h 39m 30s	1/4	UTILISATEURS OK - 2 utilisateurs actuellement connectés sur
localhost	HTTP	OK	11-30-2023 21:48:31	0d 1h 38m 53s	1/4	HTTP OK: HTTP/1.1 200 OK - 10945 octets en 0.100 secondes de temps de réponse
localhost	PING	OK	11-30-2023 21:49:13	0d 1h 38m 15s	1/4	PING OK - Paquets perdus = 0%, RTA = 0.07 ms
localhost	Root Partition	OK	11-30-2023 21:49:46	0d 1h 37m 38s	1/4	DISK OK - free space: 2792 MB (35.92% mode=<3%>)
localhost	SSH	OK	11-30-2023 21:49:25	0d 1h 37m 0s	1/4	SSH OK - OpenSSH_8.9p1 Ubuntu-1ubuntu4.4 (protocol 2.0)
localhost	Swap Usage	OK	11-30-2023 21:51:01	0d 1h 41m 23s	1/4	SWAP OK - 100% free (1567 MB sur un total de 1380 MB)
localhost	Total Processes	OK	11-30-2023 21:48:56	0d 0h 39m 28s	1/4	PROCS OK - 197 processus avec ETAT = R/S/Z/T

Results 1 - 8 of 8 Matching Services

```
zemni@nagios1:/usr/local/nagios/etc$ cd etc
zemni@nagios1:/usr/local/nagios/etc$ ls -la
total 104
drwxrwxr-x 3 nagios nagios 4096 déc. 4 15:58 .
drwxr-xr-x 8 root root 4096 déc. 2 09:59 ..
-rw-rw-r-- 1 nagios nagios 13719 déc. 4 12:37 cgi.cfg
-rw-r--r-- 1 root root 50 déc. 2 10:03 htpasswd.users
-rw-rw-r-- 1 nagios nagios 46010 déc. 4 14:38 nagios.cfg
-rw-r--r-- 1 nagios nagios 12851 déc. 4 15:13 nrpe.cfg
-rw-r--r-- 1 root root 1024 déc. 4 15:58 nrpe.cfg.swp
drwxrwxr-x 2 nagios nagios 4096 déc. 4 15:56 objects
-rw-rw---- 1 nagios nagios 1313 déc.. 2 09:59 resource.cfg
zemni@nagios1:/usr/local/nagios/etc$
```

```
contacts.cfg  switches.cfg  timeperiods.cfg
zemni@nagios4:/usr/local/nagios/etc/objects$ ls -l
total 56
-rw-rw-r-- 1 nagios nagios 6864 déc. 6 15:16 commands.cfg
-rw-rw-r-- 1 nagios nagios 1797 déc. 6 14:08 contacts.cfg
-rw-rw-r-- 1 nagios nagios 4777 déc. 6 14:08 localhost.cfg
-rw-r--r-- 1 root root 1405 déc. 6 17:20 nagios5.cfg
-rw-rw-r-- 1 nagios nagios 3001 déc. 6 14:08 printer.cfg
-rw-rw-r-- 1 nagios nagios 3484 déc. 6 14:08 switch.cfg
-rw-rw-r-- 1 nagios nagios 12533 déc. 6 14:08 templates.cfg
-rw-rw-r-- 1 nagios nagios 3512 déc. 6 14:08 timeperiods.cfg
-rw-rw-r-- 1 nagios nagios 4074 déc. 6 14:08 windows.cfg
zemni@nagios4:/usr/local/nagios/etc/objects$ _
```

## Configuration des heures de supervision

Il est nécessaire d'indiquer les heures de supervisions active. Il est possible de créer des périodes concernant les heures et les jours ouvrés... mais pour superviser des serveurs, il est préférable d'activer la supervision H 24 et 7 jours par semaine.

Le fichier /usr/local/nagios/etc/objects/timeperiod.cfg n'est pas à créer. Mais il faut vérifier que 24x7 existe bien...

```

define timeperiod {
    name          24x7
    timeperiod_name 24x7
    alias         24 Hours A Day, 7 Days A Week
    sunday       00:00-24:00
    monday       00:00-24:00
    tuesday      00:00-24:00
    wednesday    00:00-24:00
    thursday     00:00-24:00
    friday       00:00-24:00
    saturday     00:00-24:00
}

# This defines a timeperiod that is normal workhours for
# those of us monitoring networks and such in the U.S.
define timeperiod {
    name          workhours
    timeperiod_name workhours
    alias         Normal Work Hours
    monday       08:00-17:00
    tuesday      08:00-17:00
    wednesday    08:00-17:00
    thursday     08:00-17:00
    friday       08:00-17:00
}

```

## Configuration des groupes d'hôtes

Sous nagios on peut créer un autre fichier nommé groups.cfg pour les nouveaux groupes , ce qui permet de créer des groupes d'hôtes pour simplifier la gestion de ces derniers. Le fichier /usr/local/nagios/etc/objects/groups.cfg est à créer.

```

define hostgroup {
    hostgroup_name      linux-servers
    alias               Linux Servers
}

define hostgroup {
    hostgroup_name      linux-web-servers
    alias               Linux web Servers
}

```

## Déclaration des hôtes

Nagios nécessite que l'on déclare les hôtes à superviser et les groupes si on crée pas le fichier groups.cfg . Elle se fait de façon simplifiée avec le fichier /usr/local/nagios/etc/objects/localhost.cfg qui est à créer.

On peut ajouter les groupes et les hosts dans la configuration de fichier localhost.cfg dans le serveur

```

# Define a host for the local machine
define host {
    use           linux-server ; Name of host template to use
    ; This host definition will inherit all variables
    ; in (or inherited by) the linux-server host template
    host_name     localhost
    alias         localhost
    address       127.0.0.1
}

#####
# HOST GROUP DEFINITION
#####

# Define an optional hostgroup for Linux machines

define hostgroup {
    hostgroup_name      linux-servers ; The name of the hostgroup
    alias               Linux Servers ; Long name of the group
    members             localhost ; Comma separated list of hosts that belong to this hostgroup
}

```

## Configuration des contacts

Un contact sous Nagios représente la personne devant être notifiée en cas de souci. Le fichier /usr/local/nagios/etc/objects/contacts.cfg permet de le déclarer.

```
define contact {
    contact_name      nagiosadmin          ; Short name of user
    use               generic-contact       ; Inherit default values from generic-contact template (defined above)
    alias             Nagios Admin         ; Full name of user
    email             sarraraouf@hotmail.com ; <***** CHANGE THIS TO YOUR EMAIL ADDRESS ****>
}

#####
# CONTACT GROUPS
#####
#####

# We only have one contact in this simple configuration file, so there is
# no need to create more than one contact group.

define contactgroup {
    contactgroup_name   admins
    alias              Nagios Administrators
    members            nagiosadmin
}
```

## Configuration des commandes

Et pour ajouter des Framework à Controller, j'ajoute des commands et chaque bloc de commandes on spécifier le nom du Framework dans le fichier commands.cfg Le fichier /usr/local/nagios/etc/objects/commands.cfg permet de le déclarer.

```
GNU nano 6.2                               commands.cfg

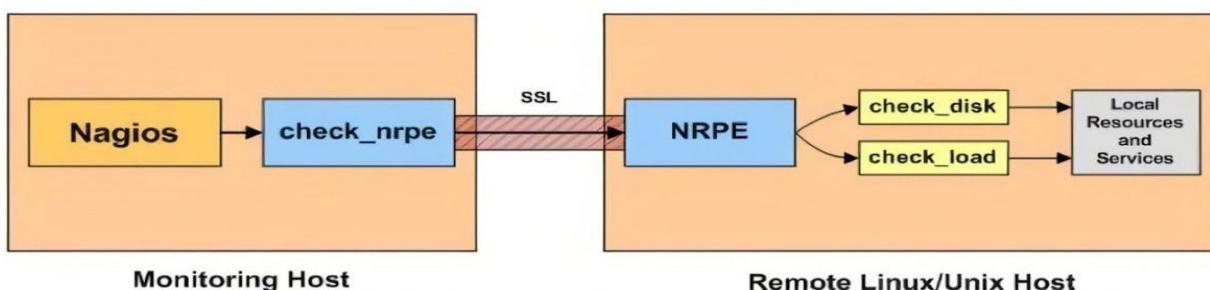
define command {
    command_name  check_local_procs
    command_line   $USER1$/check_procs -w $ARG1$ -c $ARG2$ -s $ARG3$
}

define command {
    command_name  check_local_users
    command_line   $USER1$/check_users -w $ARG1$ -c $ARG2$ 
}

define command {
    command_name  check_local_swap
    command_line   $USER1$/check_swap -w $ARG1$ -c $ARG2$ 
}

define command {
    command_name  check_local_mrtgtraf
    command_line   $USER1$/check_mrtgtraf -F $ARG1$ -a $ARG2$ -w $ARG3$ -c $ARG4$ -e $ARG5$
```

NRPE est l'acronyme pour Nagios Remote Plugin Executor. NRPE sera installé sur le serveur Nagios Core ainsi que sur les serveurs que vous surveillerez. Il est important de faire la différence entre ses deux



J'installe le NRPE et les plugins sur toute les machines, les serveur moniteur et les autres serveurs à surveiller

## **Sur la machine moniteur on fais les taches suivantes :**

1+ Et on va configurer le fichier :

```
zemni@nagios4:/usr/local/nagios/etc/objects$ sudo nano commands.cfg
```

Et j'ajoute ce code :

```
define command{
    command_name check_nrpe
    command_line $USER1$/check_nrpe -H $HOSTADDRESS$ -c $ARG1$
}
```

2+ Je vais ajouter dans le fichier /usr/local/nagios/etc/nagios.cfg:

```
# You can specify individual object config files as shown below:
cfg_file=/usr/local/nagios/etc/objects/commands.cfg
cfg_file=/usr/local/nagios/etc/objects/contacts.cfg
cfg_file=/usr/local/nagios/etc/objects/timeperiods.cfg
cfg_file=/usr/local/nagios/etc/objects/templates.cfg
cfg_file=/usr/local/nagios/etc/objects/nagios5.cfg
# Definitions for monitoring the local (Linux) host
cfg_file=/usr/local/nagios/etc/objects/localhost.cfg
```

3+ J'ajoute la ligne suivante :

'Cfg\_file=/usr/local/nagios/etc/objects/nagios5.cfg'

4+ Et en ajoute le fichier dans le répertoire '/usr/local/nagios/etc/objects' et on édite le fichier nagios5.cfg

Dans le fichier /usr/local/nagios/etc/objects/nagios5.cfg

```
define host{
    use          linux-server
    host_name   nagios5
    alias       nagios-client
    address     192.168.163.163
}

define service {
    use          generic-service
    host_name   nagios5
    service_description  Users
    check_command check_nrpe!check_users
}

define service {
    use          generic-service
    host_name   nagios5
    service_description  CPU load
    check_command check_nrpe!check_load
}

define service {
    use          generic-service
    host_name   nagios5
    service_description  Disk /
    check_command check_nrpe!check_sda1
}

define service {
    use          generic-service
    host_name   nagios5
    service_description  Zombie Processes
    check_command check_nrpe!check_zombie_procs
}

define service {
    use          generic-service
    host_name   nagios5
```

## **Sur la machine distante on fait les taches suivantes :**

Sur les serveurs distants j'installe également le NRPE et les plugins avec Ansible. Ce fichier playbook.yml

```

---
- name: Install NRPE on Target Servers
  hosts: your_target_servers
  become: yes

  vars:
    nrpe_version: "3.2.1" # Mettez à jour la version en fonction de vos besoins
    nagios_server_ip: "your_nagios_server_ip" # Remplacez par l'IP de votre serveur Nagios

  tasks:
    - name: Update apt cache (for Debian/Ubuntu)
      apt:
        update_cache: yes
      when: ansible_os_family == 'Debian'

    - name: Install required packages
      package:
        name:
          - wget
          - build-essential
          - openssl
          - libssl-dev
      become: yes

    - name: Download NRPE source code
      get_url:
        url: "https://github.com/NagiosEnterprises/nrpe/archive/nrpe-4.1.0.tar.gz"
        dest: "/tmp/nrpe-4.1.0.tar.gz"
        mode: 0644

    - name: Extract NRPE source code
      ansible.builtin.unarchive:
        src: "/tmp/nrpe-4.1.0.tar.gz"
        dest: "/tmp/"
        remote_src: yes
        creates: "/tmp/nrpe-nrpe-4.1.0"

    - name: Configure NRPE
      ansible.builtin.command:
        cmd: "./configure --enable-command-args --with-nagios-server=192.168.163.162"
        chdir: "/tmp/nrpe-nrpe-4.1.0"

    - name: Compile and install NRPE
      ansible.builtin.make:
        chdir: "/tmp/nrpe-nrpe-4.1.0"

    - name: Install NRPE binaries and configuration files
      ansible.builtin.make:
        chdir: "/tmp/nrpe-nrpe-4.1.0"
        args: "install-plugin"

    - name: Cleanup temporary files
      ansible.builtin.file:
        path: "/tmp/nrpe-4.1.0.tar.gz"
        state: absent

    ansible.builtin.command:
      cmd: "rm -rf /tmp/nrpe-nrpe-4.1.0"

    - name: Allow NRPE in the firewall
      ansible.builtin.firewalld:

```

```
service: nrpe
permanent: yes
state: enabled
when: ansible_service_mgr == 'firewalld'
```

Je fais ses étapes pour mettre en place l'installation de NRPE sur un serveur distant.

Test : Et je teste sur la machine de contrôle ce code pour tester la connexion

```
zemni@nagios4:~/usr/local/nagios/libexec/check_nrpe -H 192.168.163.163
NRPE v4.1.0
```

Pour confirmer l'installation de nrpe

```
zemni@nagios4:~/usr/local/nagios/etc$ ls -la
total 104
drwxrwxr-x 3 nagios nagios 4096 déc. 4 15:58 .
drwxr-xr-x 8 root root 4096 déc. 2 09:59 ..
-rw-rw-r-- 1 nagios nagios 13719 déc. 4 12:37 cgi.cfg
-rw-r--r-- 1 root root 50 déc. 2 10:03 htpasswd.users
-rw-rw-r-- 1 nagios nagios 46010 déc. 4 14:38 nagios.cfg
-rw-r--r-- 1 nagios nagios 12851 déc. 4 15:13 nrpe.cfg
-rw-r--r-- 1 root root 1024 déc. 4 15:58 nrpe.cfg.swp
drwxrwxr-x 2 nagios nagios 4096 déc. 4 15:56 objects
-rw-rw---- 1 nagios nagios 1313 déc. 2 09:59 resource.cfg
zemni@nagios4:~/usr/local/nagios/etc$
```

Et je commence la configuration :

1+ j'ajoute ses lignes

Allowed\_hosts=127.0.0.1, ::1, 192.168.163.162

Server\_address=0.0.0.0

Dont\_blame\_nrpe=1

```
GNU nano 6.2                               /usr/local/nagios/etc/nrpe.cfg
allowed_hosts=127.0.0.1,::1,192.168.163.162
server_address=0.0.0.0

# COMMAND ARGUMENT PROCESSING
# This option determines whether or not the NRPE daemon will allow
# to specify arguments to commands that are executed. This option
# if the daemon was configured with the --enable-command-args config
# option.
#
# *** ENABLING THIS OPTION IS A SECURITY RISK! ***
# Read the SECURITY file for information on some of the security
# of enabling this variable.
#
# Values: 0=do not allow arguments, 1=allow command arguments
dont_blame_nrpe=1
```

Et dans le même fichier j'ajoute ses lignes

```
GNU nano 6.2                               nrpe.cfg *
command[check_users]=/usr/local/nagios/libexec/check_users -w 5 -c 10
command[check_load]=/usr/local/nagios/libexec/check_load -r -w .15,.10,.05 -c .30,.25,.20
command[check_sdai]=/usr/local/nagios/libexec/check_disk -w 20% -c 10% -p /dev/sdai
command[check_zombie_procs]=/usr/local/nagios/libexec/check_procs -w 5 -c 10 -s Z
command[check_total_procs]=/usr/local/nagios/libexec/check_procs -w 150 -c 200

# The following examples allow user-supplied arguments and can
# only be used if the NRPE daemon was compiled with support for
# command arguments *AND* the dont_blame_nrpe directive in this
# config file is set to '1'. This poses a potential security risk, so
# make sure you read the SECURITY file before doing this.

### MISC SYSTEM METRICS ###
#command[check_users]=/usr/local/nagios/libexec/check_users $ARG1$
#command[check_load]=/usr/local/nagios/libexec/check_load $ARG1$
#command[check_disk]=/usr/local/nagios/libexec/check_disk $ARG1$
command[check_swap]=/usr/local/nagios/libexec/check_swap $ARG1$
command[check_cpu_stats]=/usr/local/nagios/libexec/check_cpu_stats.sh $ARG1$
command[check_mem]=/usr/local/nagios/libexec/custom_check_mem -n $ARG1$

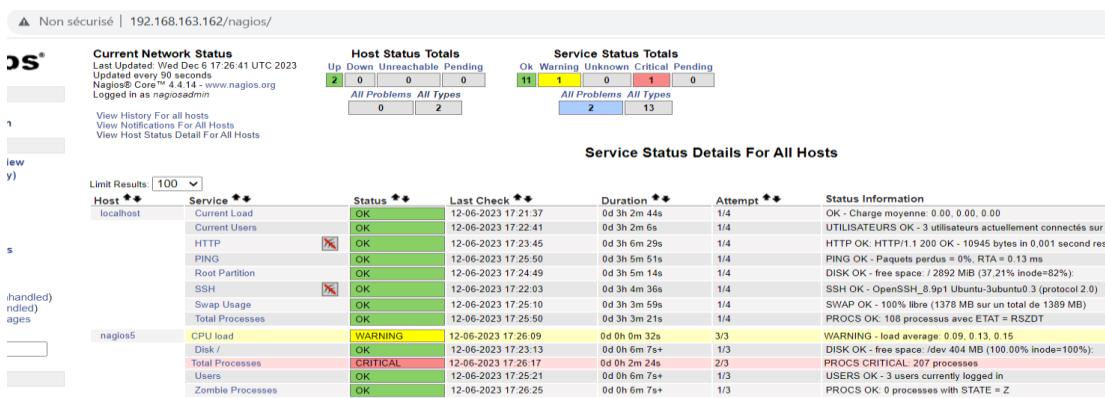
### GENERIC SERVICES ###
command[check_init_service]=sudo /usr/local/nagios/libexec/check_init_service $ARG1$
command[check_services]=-p $ARG1$

### SYSTEM UPDATES ###
#command[check_yum]=/usr/local/nagios/libexec/check_yum
command[check_apt]=/usr/local/nagios/libexec/check_apt

### PROCESSES ###
#command[check_all_procs]=/usr/local/nagios/libexec/custom_check_procs
#command[check_procs]=-p $ARG1$

### OPEN FILES ###
#command[check_open_files]=/usr/local/nagios/libexec/check_open_files.pl $ARG1$

### NETWORK CONNECTIONS ###
#command[check_netstat]=-p $ARG1$ $ARG2$
```



## La sonde check\_mysql\_query

Cette sonde permet de surveiller l'activité du site web concernant les commentaires de type spam, cela permet de remonter une alerte de type "warning" dès que 4 commentaires sont postés ou plus ont été postés dans les 4 dernières heures et une alerte de type "critical" dès que 10 commentaires ou plus ont été postés également dans les 4 dernières heures.

Ajout de la sonde côté serveur

Fichier :

```
define service {
    use generic-service
    hosts_name wordpress
    service_description Wordpress Comments
    check_command check_nrpe!check_mysql_query
}
```

Activation de la sonde côté clients Avant de pouvoir commencer à utiliser cette sonde, il faut créer un utilisateur pouvant se connecter à la base de données mariadb en exécutant les commandes suivantes : Modifier le fichier nrpe.cfg sur le client hébergeant Mariadb.

Fichier : ... command[check\_mysql\_query]=/usr/lib/nagios/plugins/check\_mysql\_query -u nagios \ -p Azerty11 -d wp\_database -P 3306 -q "SELECT COUNT(\*) FROM wp\_comments \ WHERE comment\_date BETWEEN DATE\_SUB(NOW(), INTERVAL 4 HOUR) AND NOW()" -w 4 -c 10

## 6. Conclusion

Mon stage chez Smart Web Agency a été une expérience enrichissante qui m'a permis d'approfondir mes connaissances en matière de DevOps et de me plonger dans le monde dynamique du développement et de l'exploitation conjoints. Au cours de ces derniers mois, j'ai eu l'opportunité de travailler sur des projets variés, d'appliquer les meilleures pratiques DevOps, et de collaborer étroitement avec une équipe passionnée et talentueuse.

L'un des aspects les plus stimulants de mon stage a été la mise en œuvre et l'optimisation de pipelines d'intégration continue et de déploiement continu. Cela m'a permis de comprendre l'importance de l'automatisation dans le cycle de vie du développement logiciel, améliorant ainsi l'efficacité des processus et la qualité des livrables. J'ai également acquis une solide expérience dans la gestion des

infrastructures, en utilisant des outils tels que Kubernetes, Docker , Ansible et Terraform pour créer des environnements fiables et évolutifs.

La culture DevOps prônée par Smart Web Agency m'a enseigné l'importance de la collaboration entre les équipes de développement et d'exploitation. Les itérations rapides, les retours fréquents et l'approche axée sur la résolution rapide des problèmes ont été des aspects cruciaux de cette culture, contribuant à une amélioration continue des processus et des produits.

Ce stage a été une étape déterminante dans mon parcours professionnel. J'ai acquis des compétences techniques avancées, mais surtout, j'ai appris à travailler au sein d'une équipe DevOps et à comprendre l'impact positif de cette approche sur la livraison de logiciels de qualité. Je suis reconnaissant envers Smart Web Agency pour cette expérience formatrice, et je suis prêt à appliquer ces compétences nouvellement acquises dans mes projets futurs.

## 7. Remerciement :

Je tiens à exprimer ma profonde gratitude envers le Centre de Formation M2I pour le parcours enrichissant que j'ai suivi en tant qu'Administrateur de Système DevOps. Cette expérience a été des plus formatrices, et je suis reconnaissant(e) d'avoir eu l'opportunité de bénéficier de vos enseignements et de votre expertise.

Votre dévouement, votre passion pour le domaine, ainsi que la qualité de vos cours ont grandement contribué à mon apprentissage et à ma croissance professionnelle. Les compétences acquises au cours de cette formation ont indéniablement renforcé mes connaissances dans le domaine de l'administration système et du DevOps.

Par la même occasion, je tiens à exprimer ma gratitude envers la société SMART WEB AGENCY, où j'ai eu l'opportunité de réaliser mon stage. Les défis passionnants auxquels j'ai été confronté(e) et le soutien constant de l'équipe ont grandement contribué à ma compréhension pratique des concepts abordés en formation.

Je suis conscient(e) du privilège que j'ai eu de bénéficier de ces enseignements de qualité et des opportunités offertes par M2I et SMART WEB AGENCY. Ces expériences ont été déterminantes pour mon parcours professionnel, et je suis impatient(e) de mettre en pratique les compétences acquises.

Je tiens à vous exprimer ma reconnaissance sincère et à saluer l'ensemble de l'équipe pédagogique de M2I ainsi que SMART WEB AGENCY pour leur contribution significative à mon développement professionnel.

## 8. Les résultats obtenus

Nom Prénom	Contrôle 20231004_ECF		Contrôle 20231016_ECF	
ZEMNI ep.KASSAS Sarra	Examen_M3_M30_31_Azure_20230 908	18.2	Examen_M43_Projet fin de cursus_Fiche 8_20231017	17
	Examen_M32_34_GCP_20230915	16	Examen_M19_Virtualisation	8
	Examen_xx_Supervision_20230922	16	Examen_M15_M16_M17_Shell_Linux Admin_Linux Reseau	30/40 15
	Examen_M36_G Kubernetes_20231004_QCM_Noté	20	Examen_M18_Linux Python	20
	Examen_M39_GCP Design and Process_20231004_QCM_Noté	17	Examen_M14_Linux_Fondamentaux	17/20
	Examen_M5_Anglais_20231002	20	Examen_M29_Powershell	15
	Examen_M23_Terraform_20231009	20	Examen_M13_Reseau	16/20
	Examen_M3_Eco_citoyen Green IT RGPD	15,3 3	Examen_M20_Fondamentaux DevOps	15
	Examen_M8_Connaissance du cloud	10,2	Examen_M21_GIT	16
	Examen_M9_Gestion de projet	Abs ente	Examen_M22_ANSIBLE	19
	Examen_M10_Agile	ABS	Examen_M27_Cloud gouvernance et sécurité	20
	Examen_M11_Python	12	Examen_M25_Docker	14
	Examen_M12_SQL	17	Examen_M26_Kubernetes	16
	Examen_M28_Windows server	14		