



Global Knowledge®

CONFIGURATION MANAGEMENT ANSIBLE

Mohamed Taher BEN BAHRI



Objectifs

- Qu'est ce que Ansible?
- Caractéristiques de Ansible
- Push Based vs Pull Based
- Architecture Ansible
- Playbook & Modules Ansible
- Commandes Ad-hoc
- Installation Ansible
- Préparer un fichier de configuration Ansible

Qu'est ce que Ansible?

- Outil d'automatisation informatique, de gestion de la configuration et de provisionnement
- Il utilise des «playbooks» pour déployer, gérer, builder, tester et configurer tout, de environnement serveur complet aux sites Web en passant par les applications de code source compilées sur mesure.

Pourquoi encore un autre outil?

“I wrote Ansible because none of the existing tools fit my brain. I wanted a tool that I could not use for 6 months, come back later, and still remember how it worked.”

- Michael DeHaan, Ansible Founder

“We need to do a rolling deployment of changes that have certain dependencies (including external services).

With Ansible this becomes trivial.

Puppet on the other hand feels like the Wild West.”

- User IUseRhetoric on [reddit.com](https://www.reddit.com)

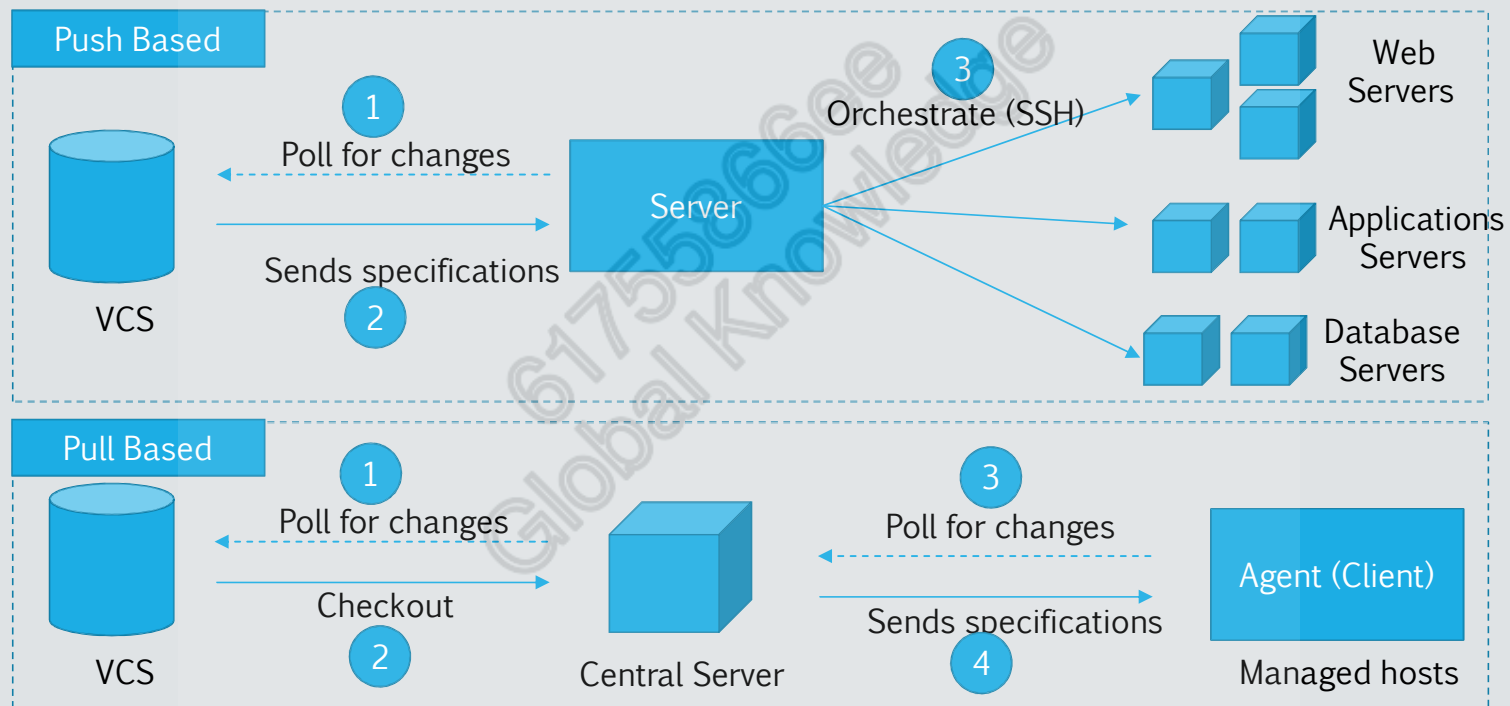
Caracteristiques Ansible

- Sans agent (Agentless) : pas besoin d'installation ni de gestion d'agent
- Construit sur Python et fournit donc beaucoup de fonctionnalités de Python
- Utilise SSH pour des connexions sécurisées
- Suit l'architecture Push pour l'envoi de configurations
- Très facile et rapide à installer, exigences minimales

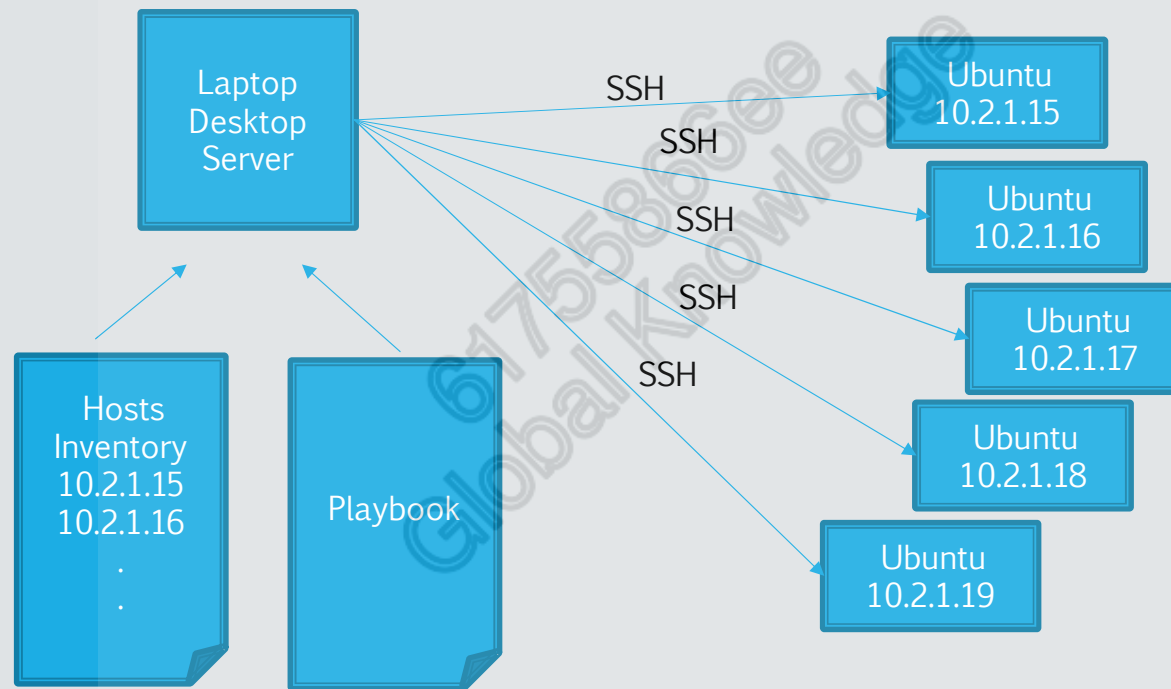
Push Based vs Pull Based

- Des outils comme Puppet et Chef sont basés sur le Pull:
 - Les agents sur le serveur vérifient périodiquement les informations de configuration du serveur central (maître).
- Ansible est basé Push
 - Le serveur central envoie les informations de configuration sur les serveurs cibles
 - Vous contrôlez le moment où les modifications sont effectuées sur les serveurs

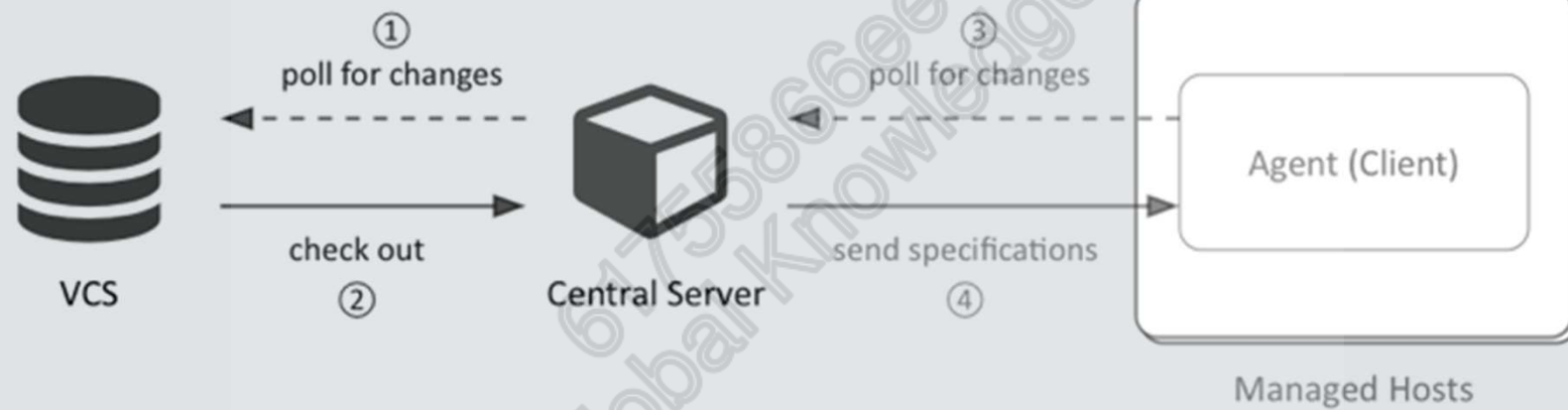
Push Based vs Pull Based



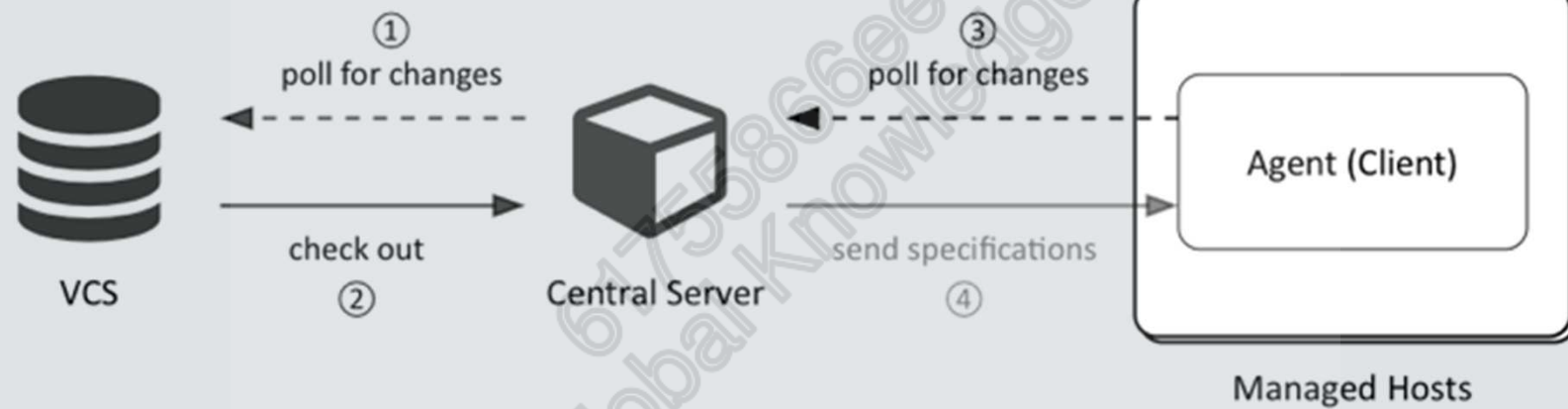
Ansible Agentless Architecture



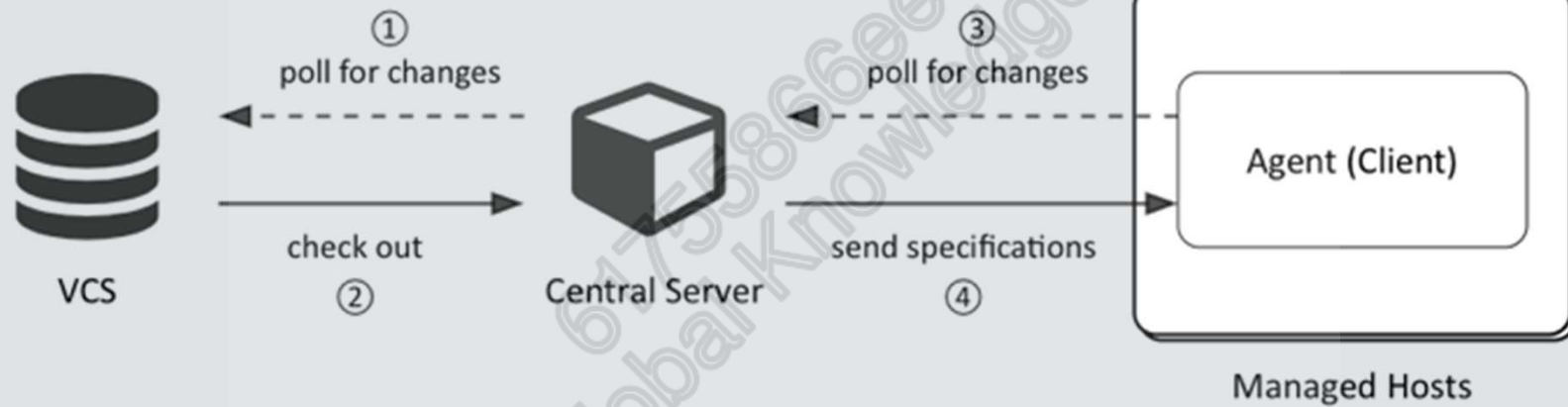
Agent-Based Architecture



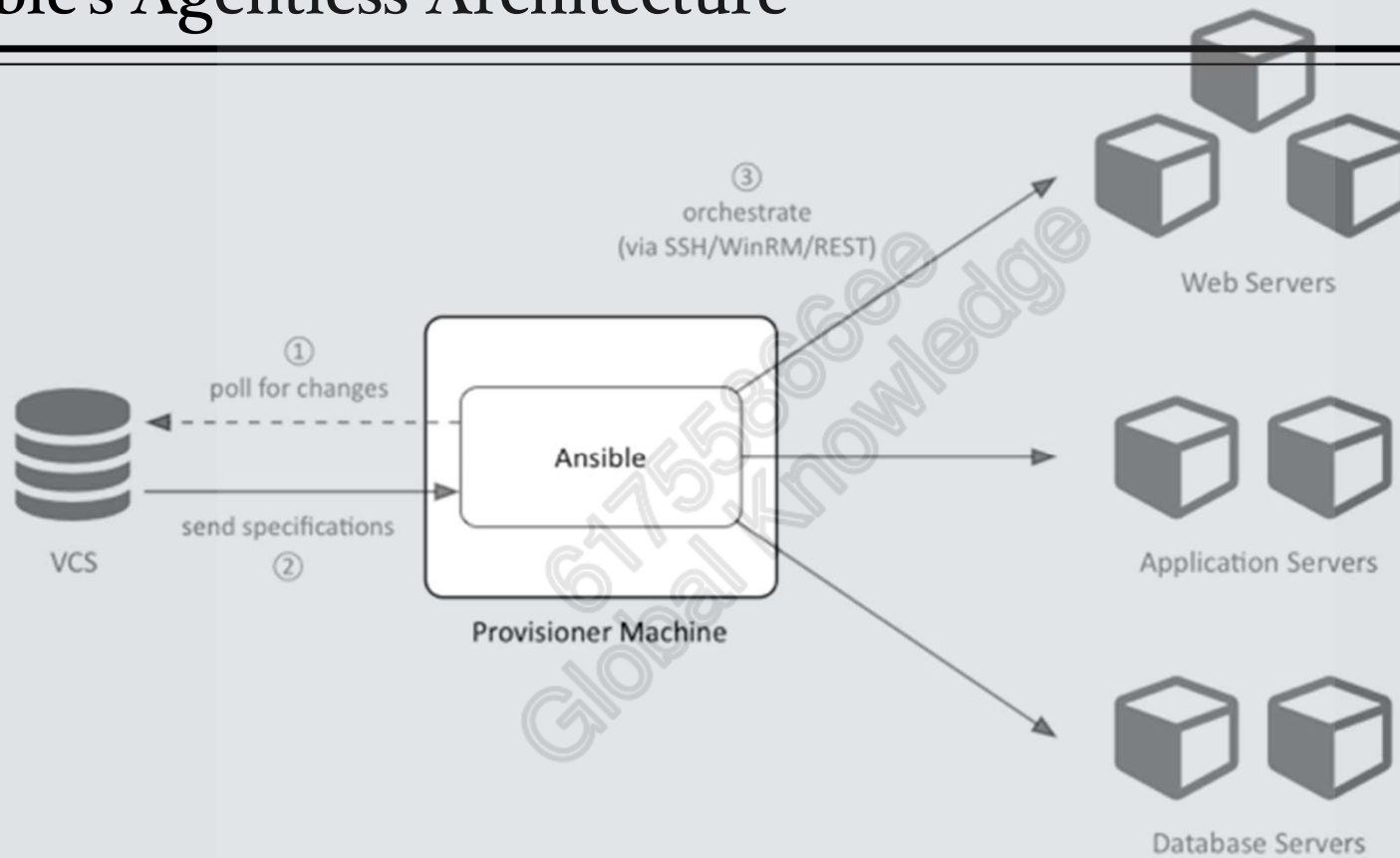
Agent-Based Architecture



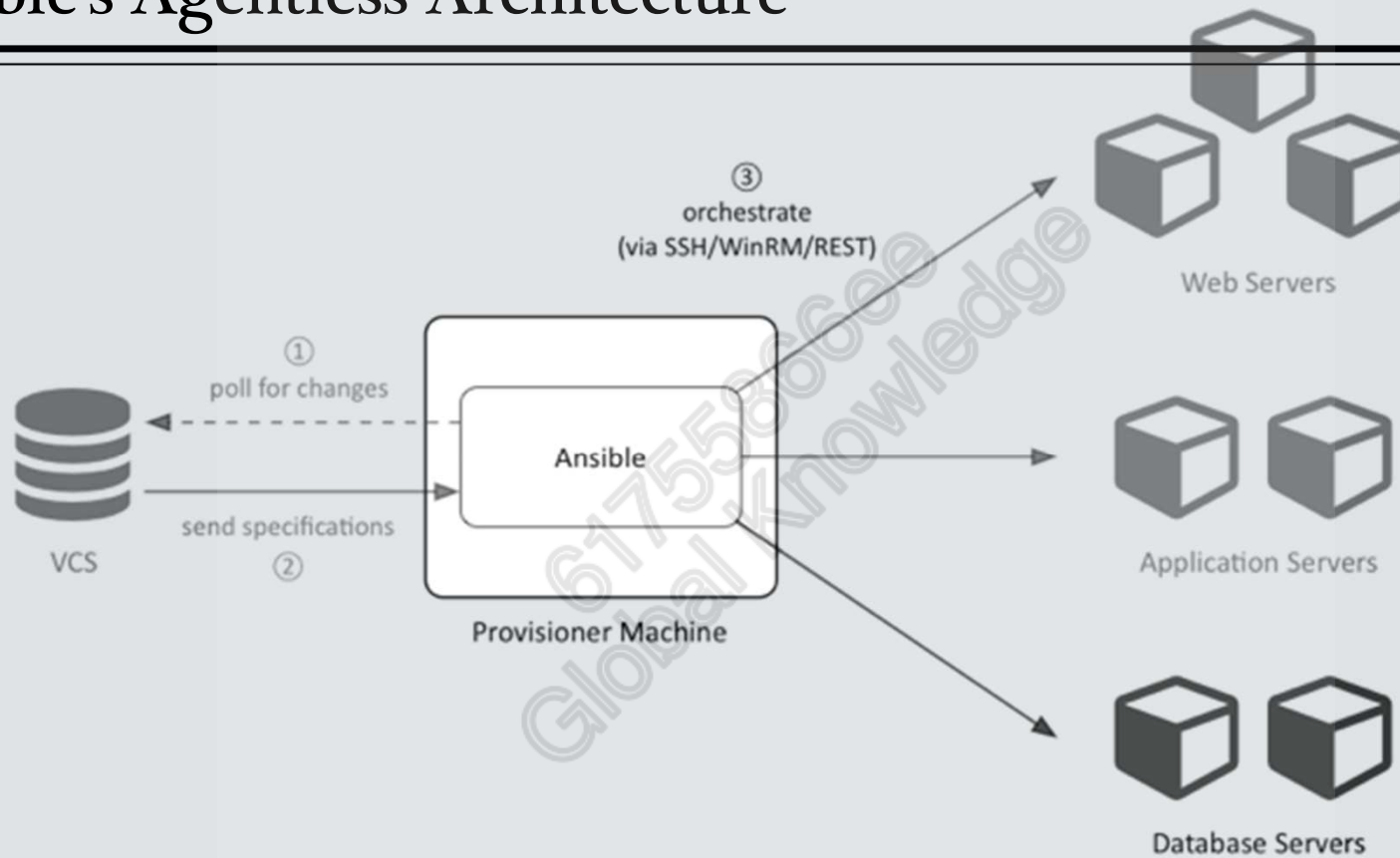
Agent-Based Architecture



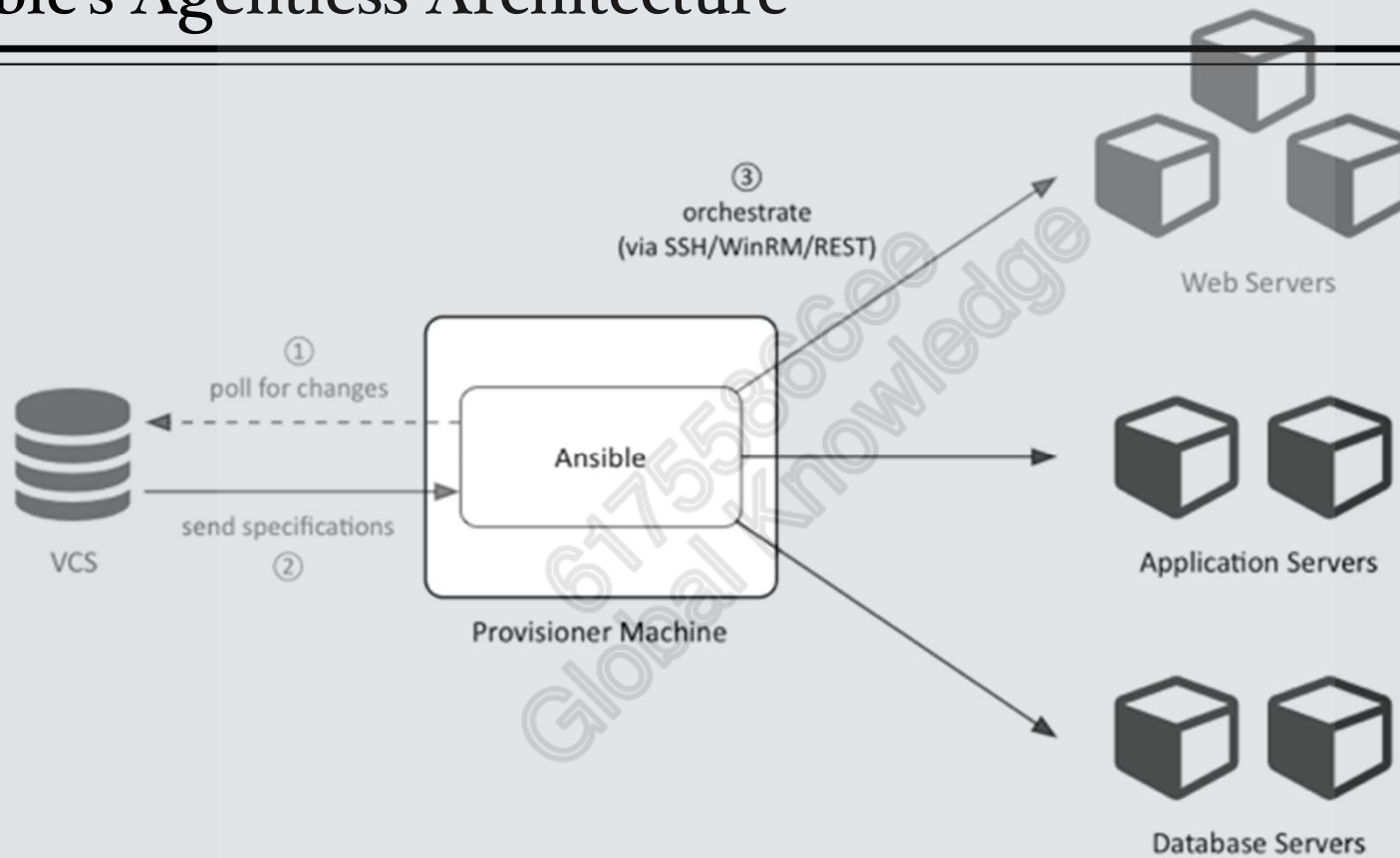
Ansible's Agentless Architecture



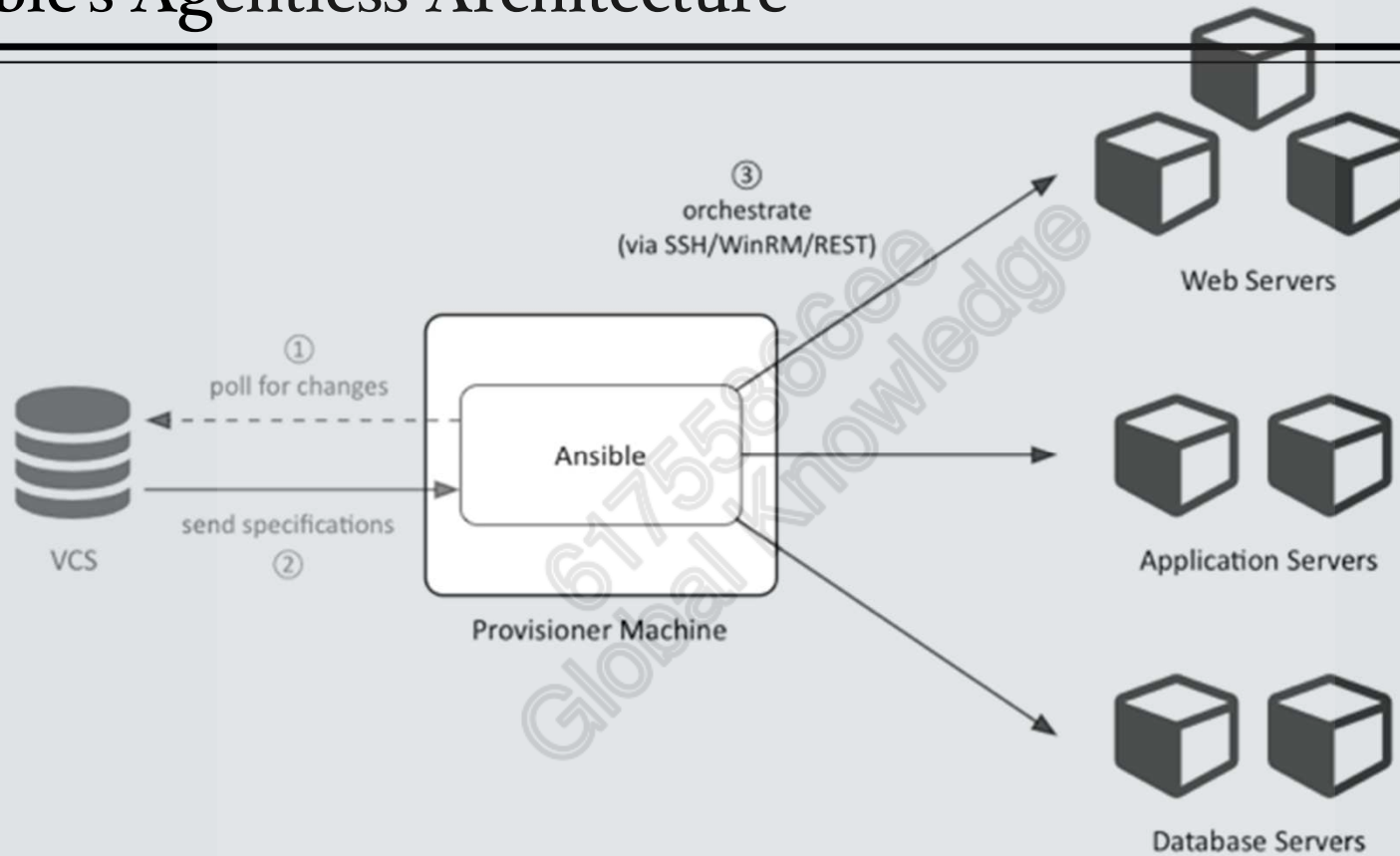
Ansible's Agentless Architecture



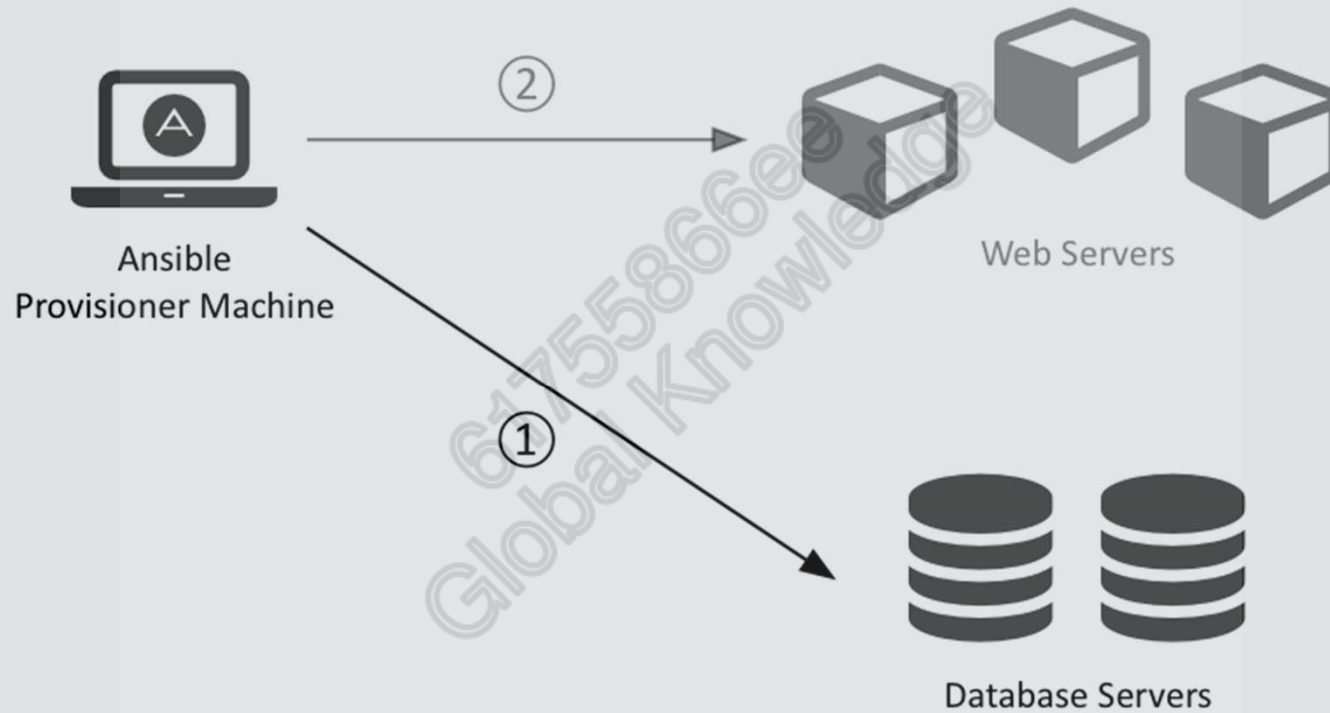
Ansible's Agentless Architecture



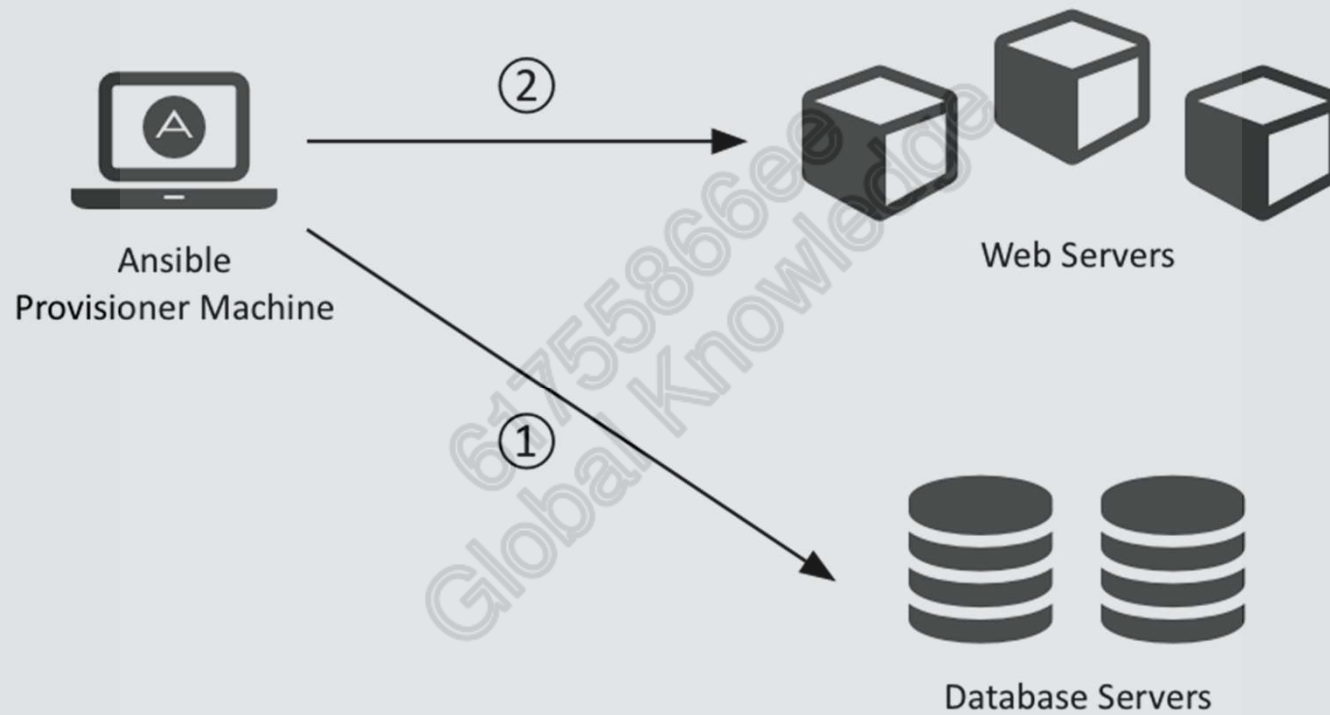
Ansible's Agentless Architecture



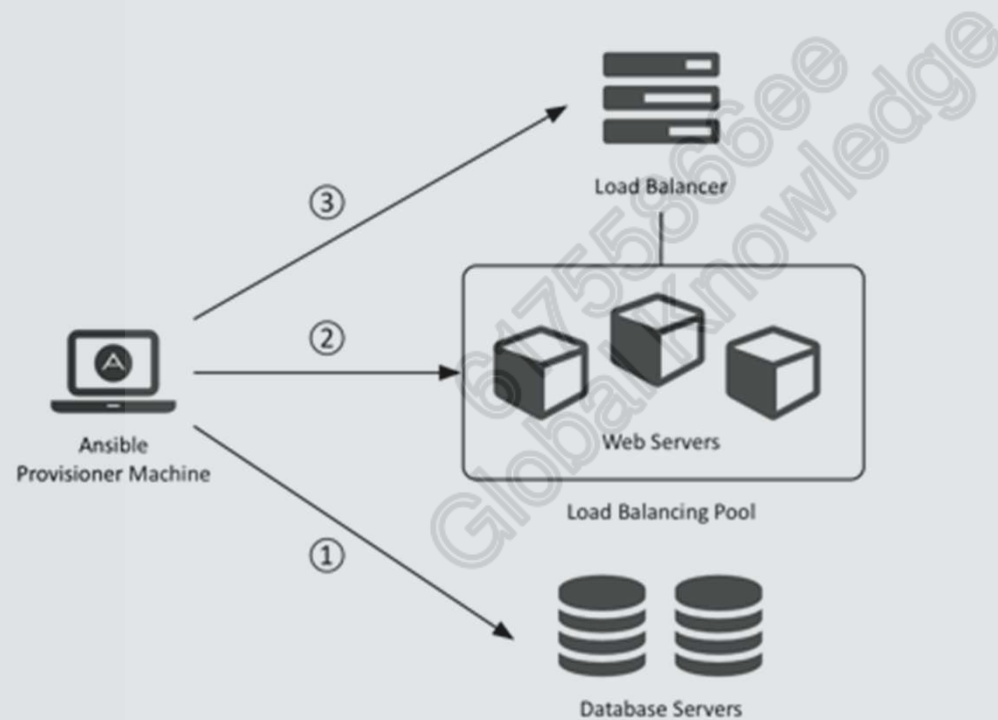
Ansible est un Moteur d'Orchestration. Alors?



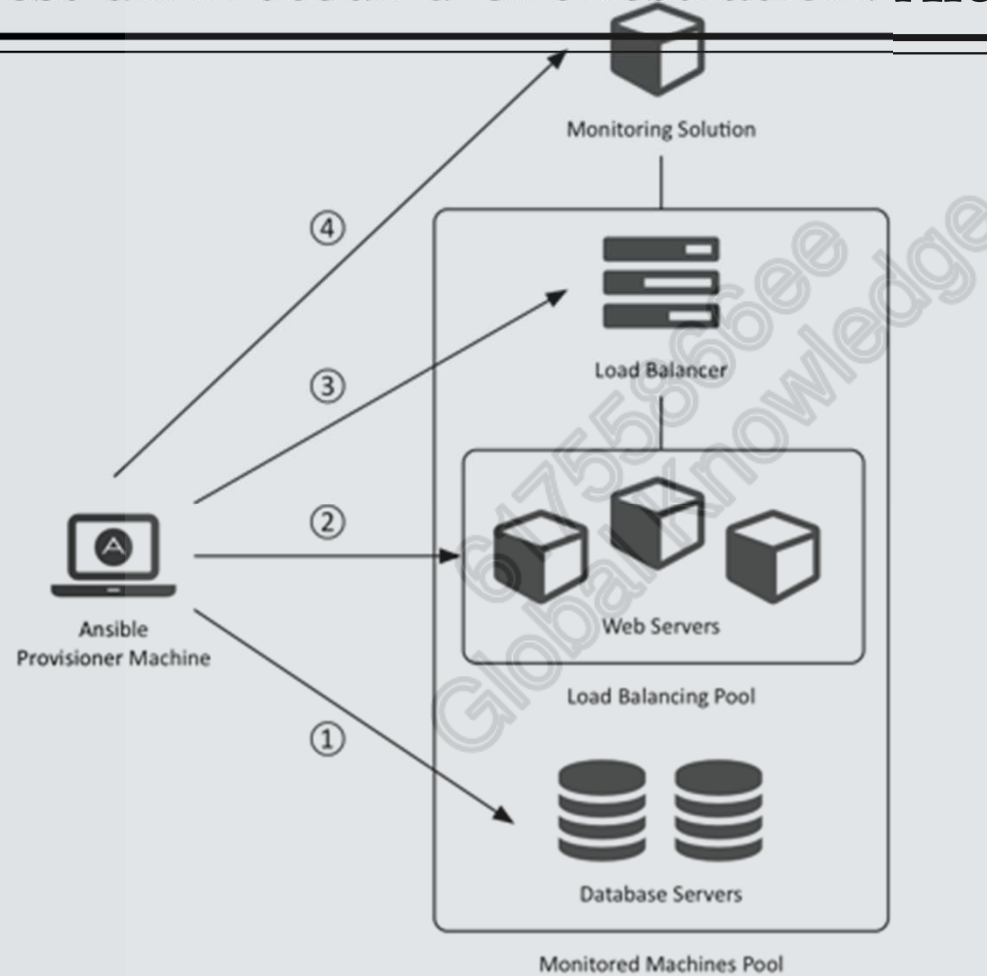
Ansible est un Moteur d'Orchestration. Alors?



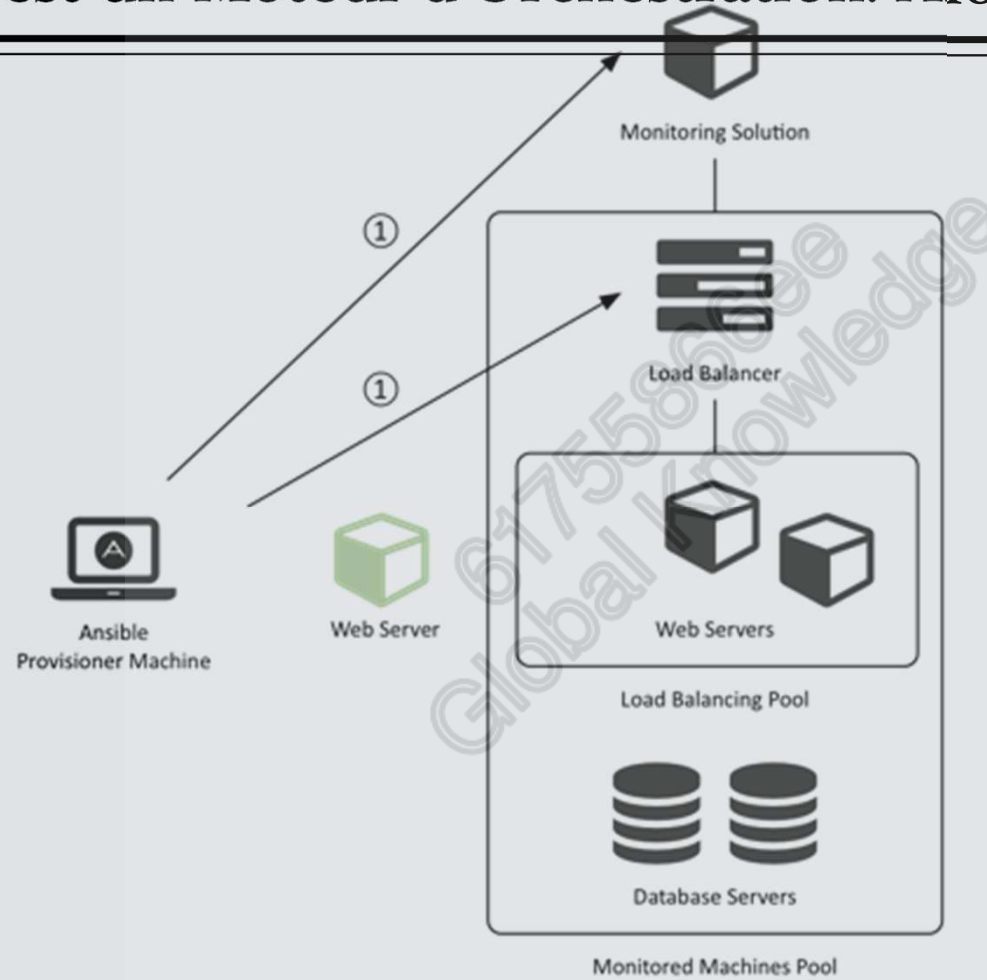
Ansible est un Moteur d'Orchestration. Alors?



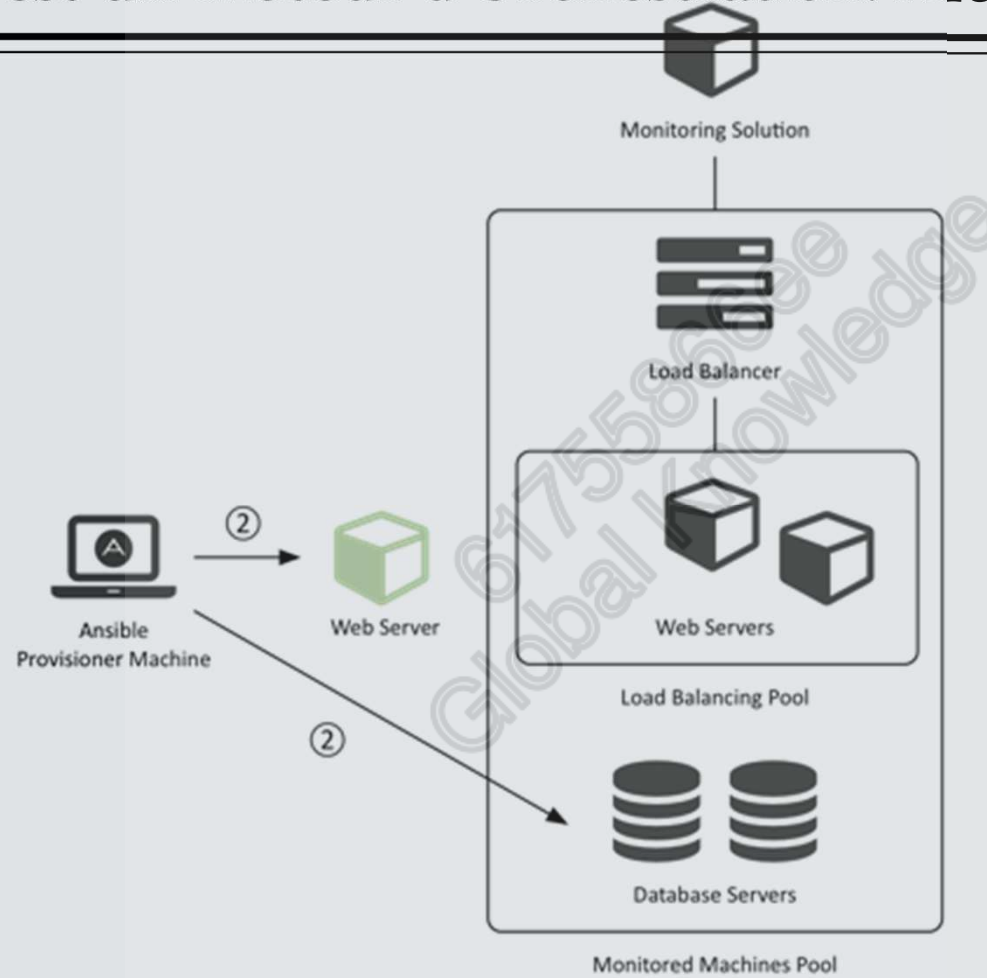
Ansible est un Moteur d'Orchestration. Alors?



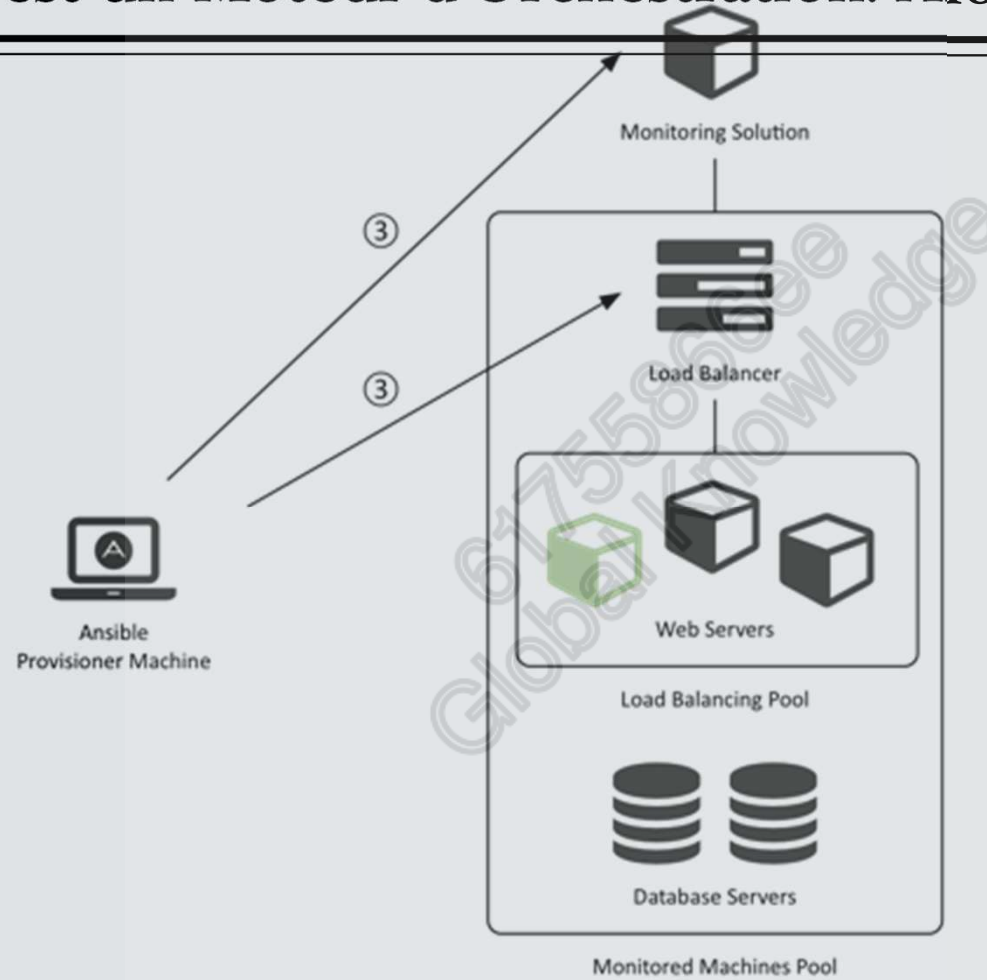
Ansible est un Moteur d'Orchestration. Alors?



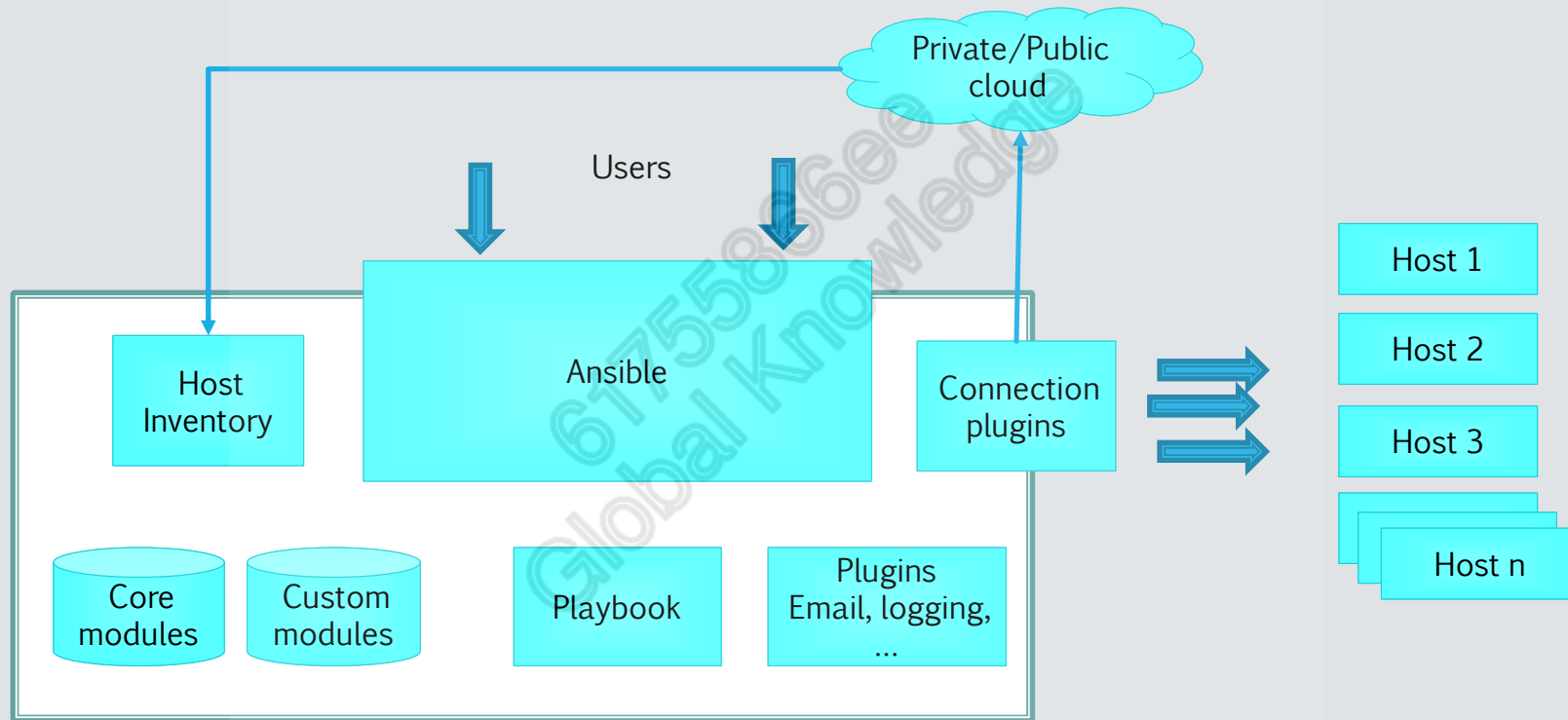
Ansible est un Moteur d'Orchestration. Alors?



Ansible est un Moteur d'Orchestration. Alors?



Ansible Architecture



Inventaire de serveurs

- Il contient la liste de vos hôtes regroupés
- L'emplacement par défaut est /etc/ansible/hosts

```
10.2.1.19  
10.2.1.20  
10.2.1.22
```

```
[webservers]  
10.2.1.19  
10.2.1.20
```

```
[dbservers]  
10.2.1.22
```


Exemple: Inventaire Statique

file: production

Group

[balancers]
www.example.com

Host

[webservers]
www[0-9].example.com

Numeric Range

[dbservers]
db[a:f].example.com

Alphabetic Range

[monitoring]
dynatrace.example.com

Exemple: Inventaire Dynamique

- Scripts Python qui extraient des données de sources dynamiques telles que:
 - Cloud: Amazon, DigitalOcean, Google, OpenShift, OpenStack, etc.
 - Services d'information distribués: LDAP, Etcd, etc.

6175586688
Global Knowledge

Modules Ansible

- Les modules (également appelés plugins de tâches ou de bibliothèques) sont ceux qui sont réellement exécutés dans un playbook
- Ce sont des scripts fournis avec Ansible qui effectuent certaines actions sur les hôtes.
- Exemples:
 - apt: installer et supprimer des paquets à l'aide du gestionnaire de paquets apt
 - copy: copie un fichier de la machine locale aux hôtes
 - file: définit l'attribut d'un fichier, d'un lien symbolique ou d'un répertoire
 - service: démarre, arrête ou redémarre un service

COMMENT UTILISER ANSIBLE?

Ansible peut être utilisé pour



DeployerApplica
tion



Configuration
Management



Sécurité et
conformité

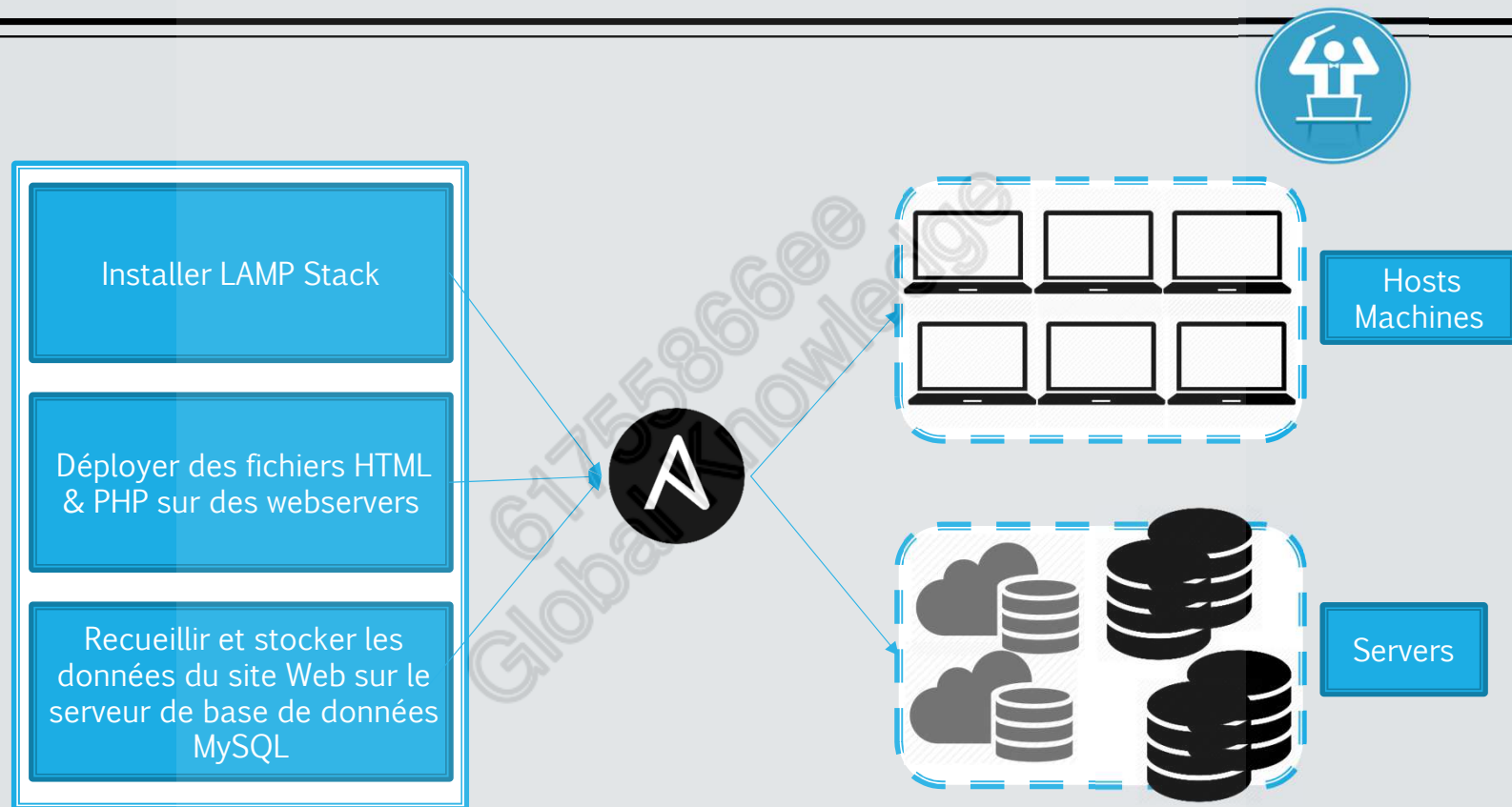


Orchestration



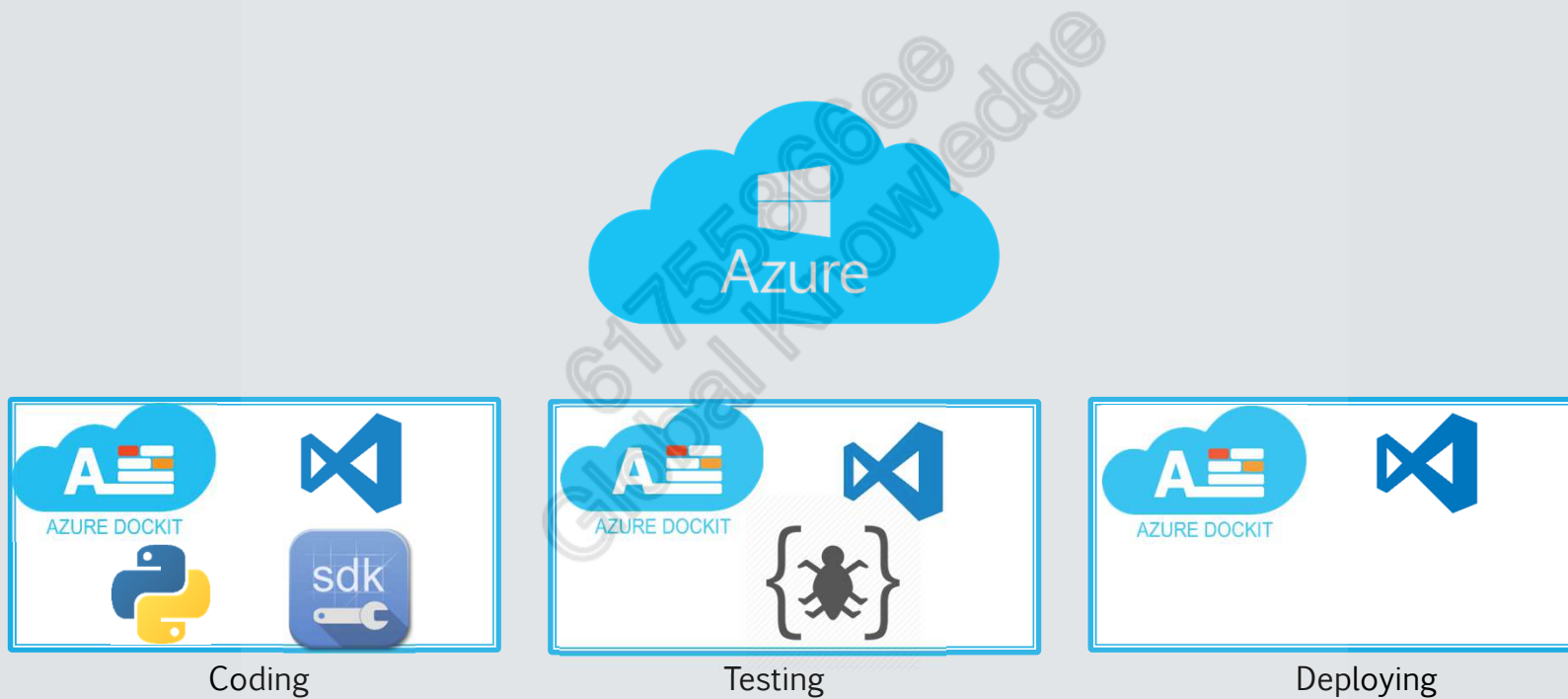
Provisioning

Orchestration



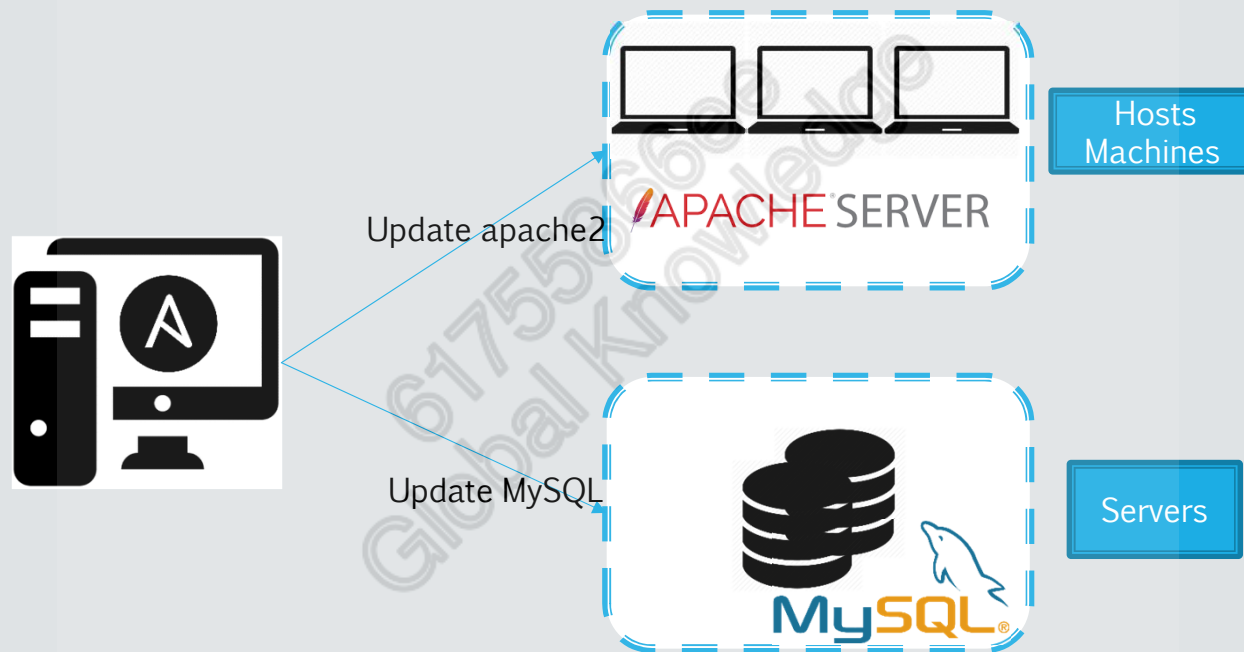
Provisioning

- Mise en service d'une application Web Python hébergée sur Azure



Configuration Management

- LAMP Stack



Ansible: commandes ad-hoc

- Une commande ad-hoc est une commande CLI que vous pouvez taper pour faire quelque chose de très rapide, mais vous ne voulez pas sauvegarder pour plus tard
 - `ansible all -s -n shell -a 'uptime'`
 - => Uptime of all the machines
 - `ansible all -s -n shell -a 'date'`
 - => Date of all the machines
 - `ansible all -s -n shell -a 'cat /etc/redhat-release'`
 - => RH release of all machines
 - `ansible all -s -n shell -a 'mount'`
 - => kind of mount on all the machines
 - `ansible all -s -n shell -a 'service sshd status'`
 - => Check the service status on all the machines

Playbook Ansible

- Le langage de configuration, de déploiement et d'orchestration de Ansible écrit en YAML,
- définissez de manière déclarative vos configurations lisible par l'humain et sont développés dans un langage de base
- Commande pour exécuter le playbook:
 - ansible-playbook fichier.yml

Exemple de Playbook

```
---  
- hosts: webserver  
  tasks:  
    - name: install nginx server  
      apt: pkg=nginx state=installed  
        update_cache=true  
        notify:  
          - start nginx  
  
  handlers:  
    - name: start nginx  
      service: name=nginx state=started
```

Comment écrire un playbook

Hosts

- list

Variables

- list

Tasks

- list

Handlers

- list

Ansible Roles

- Les rôles permettent de charger automatiquement certains fichiers vars_files, tasks et handlers en fonction d'une structure de fichier connue.
- Le regroupement de contenu par rôles permet également de partager facilement des rôles avec d'autres utilisateurs.
- Les rôles s'attendent à ce que les fichiers se trouvent dans certains noms de répertoire.
- Les rôles doivent inclure au moins un de ces répertoires, mais il est parfaitement correct d'exclure ceux qui ne sont pas utilisés.
- Lorsqu'il est utilisé, chaque répertoire doit contenir un fichier main.yml, contenant le contenu pertinent.

Ansible Roles - Directories

- tasks - contains the main list of tasks to be executed by the role.
- handlers - contains handlers, which may be used by this role or even anywhere outside this role.
- defaults - default variables for the role
- vars - other variables for the role.
- files - contains files which can be deployed via this role.
- templates - contains templates which can be deployed via this role.
- meta - defines some meta data for this role. See below for more details.

Ansible Roles - Example project structure:

```
site.yml
webservers.yml
fooservers.yml
roles/
  common/
    tasks/
    handlers/
    files/
    templates/
    vars/
    defaults/
    meta/
  webservers/
    tasks/
    defaults/
    meta/
```

Ansible Roles - Example project structure:

Other YAML files may be included in certain directories.

It is common practice to have platform-specific tasks included from the tasks/main.yml file:

roles/example/tasks/main.yml

- name: added in 2.4, previously you used 'include'

 - import_tasks: redhat.yml

 - when: ansible_os_platform|lower == 'redhat'

- import_tasks: debian.yml

 - when: ansible_os_platform|lower == 'debian'

roles/example/tasks/redhat.yml

- yum: name="httpd" state=present

roles/example/tasks/debian.yml

- apt: name="apache2" state=present

Ansible Vault

- “Vault” is a feature of ansible that allows keeping sensitive data such as passwords or keys in encrypted files
- To enable this feature, a command line tool, *ansible-vault* is used to edit files
- a command line flag *-ask-vault-pass* or *-vault-password-file* is used
 - `ansible-playbook site.yml --ask-vault-pass`