

CONTENU

Comment fonctionne Ansible ?

Conditions préalables

Étape 1 — Installation d'Ansible

Étape 2 — Configuration du fichier d'inventaire

Étape 3 — Test de la connexion

Étape 4 — Exécution de commandes ad hoc (facultatif)

Conclusion

APPARENTÉ

Configuration initiale du serveur avec Ubuntu 12.04

[Vue](#) 

Comment installer Ruby on Rails sur Ubuntu 12.04 LTS (Precise Pangolin) avec RVM

[Vue](#) 

// Tutorial //

How to Install and Configure Ansible on Ubuntu 18.04

Published on July 14, 2018 · Updated on April 16, 2020

Ubuntu

Configuration Management

Ansible

Ubuntu 18.04





By [Stephen Rees-Carter](#), [Mark Drake](#) and [Erika Heidi](#)

English ▾



Vous n'utilisez pas Ubuntu 18.04?

Choisissez une autre version ou distribution.

Ubuntu 18.04 ▾

Introduction

Les systèmes de gestion de la configuration sont conçus pour rationaliser le processus de contrôle d'un grand nombre de serveurs, pour les administrateurs et les équipes opérationnelles. Ils vous permettent de contrôler de nombreux systèmes différents de manière automatisée à partir d'un emplacement central.

Bien qu'il existe de nombreux outils de gestion de configuration populaires disponibles pour les systèmes Linux, tels que [Chef](#) et [Puppet](#), ceux-ci sont souvent plus complexes que beaucoup de gens ne le souhaitent ou n'en ont besoin. [Ansible](#) est une excellente alternative à ces options car il offre une architecture simple qui ne nécessite pas



d'installation d'un logiciel spécial sur les nœuds, en utilisant SSH pour exécuter les tâches d'automatisation et des fichiers YAML pour définir les détails de provisionnement.

Dans ce guide, nous allons discuter de la façon d'installer Ansible sur un serveur Ubuntu 18.04 et passer en revue quelques bases de l'utilisation de ce logiciel.

Comment fonctionne Ansible ?

Ansible fonctionne en configurant des ordinateurs clients, appelés *nœuds gérés*, à partir d'un ordinateur sur lequel les composants Ansible sont installés et configurés, appelé *nœud de contrôle Ansible*.

Il communique sur des canaux SSH normaux pour récupérer des informations à partir de systèmes distants, émettre des commandes et copier des fichiers. Pour cette raison, un système Ansible ne nécessite pas l'installation d'un logiciel supplémentaire sur les ordinateurs clients.

C'est l'une des façons dont Ansible simplifie l'administration des serveurs. Tout serveur dont le port SSH est exposé peut être placé sous l'égide de configuration d'Ansible, quelle que soit l'étape à laquelle il se trouve dans son cycle de vie. Cela signifie que tout ordinateur que vous pouvez administrer via SSH, vous pouvez également administrer via Ansible.

Ansible adopte une approche modulaire, vous permettant d'étendre les fonctionnalités du système principal pour faire face à des scénarios spécifiques. Les modules peuvent être écrits dans n'importe quel langage et communiquer en JSON standard.

Les fichiers de configuration sont principalement écrits au format de sérialisation de données YAML en raison de sa nature expressive et de sa similitude avec les langages de balisage populaires. Ansible peut interagir avec les hôtes via des outils de ligne de commande ou ses scripts de configuration, appelés Playbooks.

Conditions préalables

Pour suivre ce tutoriel, vous aurez besoin des éléments suivants :

- Un nœud de contrôle Ansible : le **nœud de contrôle** Ansible est la machine que nous utiliserons pour nous connecter aux hôtes Ansible et les contrôler via SSH. Votre nœud de contrôle Ansible peut être votre machine locale ou un serveur dédié à l'exécution d'Ansible, bien que ce guide suppose que votre nœud de contrôle est un système Ubuntu 18.04. Assurez-vous que le nœud de contrôle dispose des éléments suivants :



- Un utilisateur non root avec des privilèges sudo. Pour configurer cela, vous pouvez suivre **les étapes 2 et 3** de notre Guide de configuration initiale du [serveur pour Ubuntu 18.04](#). Cependant, veuillez noter que si vous utilisez un serveur distant comme nœud Ansible Control, vous devez suivre **chaque étape** de ce guide. Cela configurera un pare-feu sur le serveur avec et activera l'accès externe à votre profil utilisateur non root, ce qui aidera à sécuriser le serveur distant. ufw
- Une paire de clés SSH associée à cet utilisateur. Pour configurer cela, vous pouvez suivre **l'étape 1** de notre guide sur [Comment configurer les clés SSH sur Ubuntu 18.04](#).

- **Un ou plusieurs hôtes** Ansible : un hôte Ansible est une machine que votre nœud de contrôle Ansible est configuré pour automatiser. Ce guide suppose que vos hôtes Ansible sont des serveurs Ubuntu 18.04 distants. Assurez-vous que chaque hôte Ansible dispose des éléments suivants :

- La clé publique SSH du nœud de contrôle Ansible ajoutée à la clé d'un utilisateur système. Cet utilisateur peut être **root** ou un utilisateur régulier avec des privilèges sudo. Pour configurer cela, vous pouvez suivre **l'étape 2** de [Comment configurer les clés SSH sur Ubuntu 18.04](#). authorized_keys

Étape 1 – Installation d'Ansible

Pour commencer à utiliser Ansible comme moyen de gestion de votre infrastructure de serveur, vous devez installer le logiciel Ansible sur la machine qui servira de nœud de contrôle Ansible.

À partir de votre nœud de contrôle, exécutez la commande suivante pour inclure le PPA (archive de paquets personnels) du projet officiel dans la liste des sources de votre système :

```
$ sudo apt-add-repository ppa:ansible/ansible
```

Copier

Appuyez lorsque vous êtes invité à accepter l'ajout PPA. ENTER

Ensuite, actualisez l'index des packages de votre système afin qu'il connaisse les packages disponibles dans le PPA nouvellement inclus :

```
$ sudo apt update
```

Copier

Suite à cette mise à jour, vous pouvez installer le logiciel Ansible avec :



```
$ sudo apt install ansible
```

Copier

Votre nœud de contrôle Ansible dispose désormais de tous les logiciels nécessaires pour administrer vos hôtes. Ensuite, nous verrons comment ajouter vos hôtes au fichier d'inventaire du nœud de contrôle afin qu'il puisse les contrôler.

Étape 2 – Configuration du fichier d'inventaire

Le *fichier d'inventaire* contient des informations sur les hôtes que vous allez gérer avec Ansible. Vous pouvez inclure de un à plusieurs centaines de serveurs dans votre fichier d'inventaire, et les hôtes peuvent être organisés en groupes et sous-groupes. Le fichier d'inventaire est également souvent utilisé pour définir des variables qui ne seront valides que pour des hôtes ou des groupes spécifiques, afin d'être utilisés dans les playbooks et les modèles. Certaines variables peuvent également affecter la façon dont un playbook est exécuté, comme la variable que nous verrons dans un instant. `ansible_python_interpreter`

Pour modifier le contenu de votre inventaire Ansible par défaut, ouvrez le fichier à l'aide de l'éditeur de texte de votre choix, sur votre nœud de contrôle Ansible : `/etc/ansible/hosts`

```
$ sudo nano /etc/ansible/hosts
```

Copier

Remarque : certaines installations Ansible ne créent pas de fichier d'inventaire par défaut. Si le fichier n'existe pas dans votre système, vous pouvez créer un nouveau fichier à ou fournir un chemin d'inventaire personnalisé à l'aide du paramètre lors de l'exécution de commandes et de playbooks. `/etc/ansible/hosts -i`

Le fichier d'inventaire par défaut fourni par l'installation Ansible contient un certain nombre d'exemples que vous pouvez utiliser comme références pour configurer votre inventaire. L'exemple suivant définit un groupe nommé avec trois serveurs différents, chacun identifié par un alias personnalisé : **server1**, **server2** et **server3**. Veillez à remplacer les adresses IP en surbrillance par les adresses IP de vos hôtes Ansible. [servers]

`/etc/ansible/hosts`

```
[servers]
server1 ansible_host=203.0.113.111
server2 ansible_host=203.0.113.112
server3 ansible_host=203.0.113.113
```



```
[all:vars]
ansible_python_interpreter=/usr/bin/python3
```

Le sous-groupe définit le paramètre host qui sera valide pour tous les hôtes inclus dans cet inventaire. Ce paramètre permet de s'assurer que le serveur distant utilise l'exécutable Python 3 au lieu de (Python 2.7), qui n'est pas présent sur les versions récentes d'Ubuntu.

`all:vars ansible_python_interpreter /usr/bin/python3 /usr/bin/python`

Lorsque vous avez terminé, enregistrez et fermez le fichier en appuyant sur puis et pour confirmer vos modifications. **CTRL+X** Y ENTER

Chaque fois que vous souhaitez vérifier votre inventaire, vous pouvez exécuter :

```
$ ansible-inventory --list -y
```

Copier

Vous verrez une sortie similaire à celle-ci, mais contenant votre propre infrastructure de serveur telle que définie dans votre fichier d'inventaire :

Output

```
all:
  children:
    servers:
      hosts:
        server1:
          ansible_host: 203.0.113.111
          ansible_python_interpreter: /usr/bin/python3
        server2:
          ansible_host: 203.0.113.112
          ansible_python_interpreter: /usr/bin/python3
        server3:
          ansible_host: 203.0.113.113
          ansible_python_interpreter: /usr/bin/python3
  ungrouped: {}
```

Maintenant que vous avez configuré votre fichier d'inventaire, vous disposez de tout ce

Besoin d'une assistance en direct dans les 30 minutes pour les urgences critiques ? Inscrivez-vous au support Premium ! →

Nous embauchons

Blog Docs Obtenir Ventes de l'aide



compte disponible par défaut sur les serveurs nouvellement créés. Si vos hôtes Ansible ont déjà créé un utilisateur sudo régulier, nous vous encourageons à utiliser ce compte à la place.

Vous pouvez utiliser l'argument pour spécifier l'utilisateur du système distant. Lorsqu'il n'est pas fourni, Ansible essaiera de se connecter en tant qu'utilisateur système actuel sur le nœud de contrôle. `-u`

À partir de votre ordinateur local ou de votre nœud de contrôle Ansible, exécutez :

```
$ ansible all -m ping -u root
```

Copier

Cette commande utilisera le fichier intégré d'Ansible [ping module](#) pour exécuter un test de connectivité sur tous les nœuds de votre inventaire par défaut, en vous connectant en tant **qu'utilisateur racine**. Le module testera : `ping`

- si les hôtes sont accessibles;
- si vous avez des informations d'identification SSH valides;
- si les hôtes sont capables d'exécuter des modules Ansible à l'aide de Python.

Vous devriez obtenir une sortie similaire à celle-ci:

Output

```
server1 | SUCCESS => {
    "changed": false,
    "ping": "pong"
}
server2 | SUCCESS => {
    "changed": false,
    "ping": "pong"
}
server3 | SUCCESS => {
    "changed": false,
    "ping": "pong"
}
```

Si c'est la première fois que vous vous connectez à ces serveurs via SSH, il vous sera demandé de confirmer l'authenticité des hôtes auxquels vous vous connectez via Ansible. Lorsque vous y êtes invité, tapez, puis appuyez sur pour confirmer. `yes` `ENTER`

Une fois que vous recevez une réponse d'un hôte, cela signifie que vous êtes prêt à exécuter des commandes Ansible et des playbooks sur ce serveur. "pong"



Remarque : Si vous ne parvenez pas à obtenir une réponse réussie de vos serveurs, consultez notre [Guide de la feuille de triche Ansible](#) pour plus d'informations sur l'exécution des commandes Ansible avec différentes options de connexion.

Étape 4 – Exécution de commandes ad hoc (facultatif)

Après avoir confirmé que votre nœud de contrôle Ansible est capable de communiquer avec vos hôtes, vous pouvez commencer à exécuter des commandes ad hoc et des playbooks sur vos serveurs.

Toute commande que vous exécuteriez normalement sur un serveur distant via SSH peut être exécutée avec Ansible sur les serveurs spécifiés dans votre fichier d'inventaire. Par exemple, vous pouvez vérifier l'utilisation du disque sur tous les serveurs avec :

```
$ ansible all -a "df -h" -u root
```

Copier

Output

```
server1 | CHANGED | rc=0 >>
Filesystem      Size  Used Avail Use% Mounted on
udev            3.9G   0    3.9G   0% /dev
tmpfs           798M  624K  798M   1% /run
/dev/vda1        155G  2.3G  153G   2% /
tmpfs           3.9G   0    3.9G   0% /dev/shm
tmpfs           5.0M   0    5.0M   0% /run/lock
tmpfs           3.9G   0    3.9G   0% /sys/fs/cgroup
/dev/vda15       105M  3.6M  101M   4% /boot/efi
tmpfs           798M   0   798M   0% /run/user/0
```

```
server2 | CHANGED | rc=0 >>
Filesystem      Size  Used Avail Use% Mounted on
udev            2.0G   0    2.0G   0% /dev
tmpfs           395M  608K  394M   1% /run
/dev/vda1        78G  2.2G   76G   3% /
tmpfs           2.0G   0    2.0G   0% /dev/shm
tmpfs           5.0M   0    5.0M   0% /run/lock
tmpfs           2.0G   0    2.0G   0% /sys/fs/cgroup
/dev/vda15       105M  3.6M  101M   4% /boot/efi
tmpfs           395M   0   395M   0% /run/user/0
```

...



La commande en surveillance peut être remplacée par la commande de votre choix. `df -h`

Vous pouvez également exécuter [des modules Ansible](#) via des commandes ad hoc, de la même manière que nous l'avons fait auparavant avec le module pour tester la connexion. Par exemple, voici comment nous pouvons utiliser le module pour installer la dernière version de sur tous les serveurs de votre inventaire : `ping apt vim`

```
$ ansible all -m apt -a "name=vim state=latest" -u root
```

Copier

Vous pouvez également cibler des hôtes individuels, ainsi que des groupes et des sous-groupes, lors de l'exécution de commandes Ansible. Par exemple, voici comment vous vérifieriez le de chaque hôte du groupe : `uptime servers`

```
$ ansible servers -a "uptime" -u root
```

Copier

Nous pouvons spécifier plusieurs hôtes en les séparant par des deux-points :

```
$ ansible server1:server2 -m ping -u root
```

Copier

Pour plus d'informations sur l'utilisation d'Ansible, y compris sur l'exécution de playbooks pour automatiser la configuration du serveur, vous pouvez consulter notre [Guide de référence Ansible](#).

Conclusion

Dans ce guide, vous avez installé Ansible et configuré un fichier d'inventaire pour exécuter des commandes ad hoc à partir d'un nœud de contrôle Ansible.

Une fois que vous avez confirmé que vous êtes en mesure de connecter et de contrôler votre infrastructure à partir d'une machine contrôleur Ansible centrale, vous pouvez exécuter n'importe quelle commande ou playbook que vous souhaitez sur ces hôtes. Pour les nouveaux serveurs, le playbook de la communauté [Configuration initiale du serveur](#) est un bon point de départ. Vous pouvez également apprendre à écrire vos propres playbooks avec notre guide [Configuration Management 101: Writing Ansible Playbooks](#).

Pour plus d'informations sur l'utilisation d'Ansible, consultez notre [Guide Ansible Cheat Sheet](#).

le calcul, le stockage, la mise en réseau et les bases de données gérées.

[En savoir plus sur nous →](#)

Obtenez 200 \$ pour essayer DigitalOcean - et faites tout ce qui suit gratuitement!

Créez des applications, hébergez des sites Web, exécutez des logiciels open source, apprenez le cloud computing, etc. - toutes les ressources cloud dont vous avez besoin. Si vous n'avez jamais essayé les produits ou services de DigitalOcean auparavant, nous couvrirons vos premiers 200 \$ dans les 60 prochains jours.

[Inscrivez-vous maintenant pour activer cette offre →](#)

À propos des auteurs



[Stephen Rees-Carter](#) Auteur

Développeur et auteur chez DigitalOcean.



[Mark Drake](#) Auteur

Gestionnaire, Formation des développeurs

Rédacteur technique @ DigitalOcean



[Erika Heidi](#) Auteur

Défenseur des développeurs



Dev/Ops passionné par l'open source, PHP et Linux.

Toujours à la recherche d'une réponse?

[Poser une question](#)

[Search for more help](#)

Was this helpful?

[Yes](#)

[No](#)



Comments

6 Comments

B I U S ⌂ ⌂ H₁ H₂ H₃ ≡ ≡ “,” ⓘ ☰ <>



Leave a comment...

This textbox defaults to using **Markdown** to format your answer.

You can type !ref in this text area to quickly search our full set of tutorials, documentation & marketplace offerings and insert the link!

[Sign In or Sign Up to Comment](#)

[Grant Winney](#) • December 19, 2019

Thanks for this! I setup 2 Ubuntu droplets on DO and was able to get

working by following your tutorial... worked great. I'll be reading your

<https://www.digitalocean.com/community/tutorials/how-to-install-and-configure-ansible-on-ubuntu-18-04>

COOKIE PREFERENCES



working by following your tutorial... worked great. I'll be reading your others, on creating playbooks and such, next.

[Reply](#)

[Comspots](#) • July 24, 2019



Hello Team,

Under one of my Ansible host, I am unable to locate authorized_keys within the `~/.ssh` directory.

KIndly help

[Show replies](#) ▾ [Reply](#)

[Altif](#) • March 26, 2019

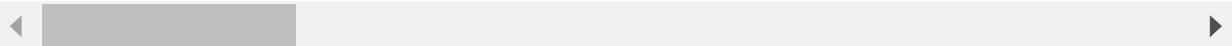


`ansible_ssh_port` deosn't work anymore with ansible 2.0 and later version.

Below link for reference.

https://docs.ansible.com/ansible/2.3/intro_inventory.html

Ansible 2.0 has deprecated the “`ssh`” from `ansible_ssh_user`, `ansible_ssh_host`, and `ansible_ssh_port`. Instead, you can use the `ansible_user`, `ansible_host`, and `ansible_port` parameters.



[Show replies](#) ▾ [Reply](#)

[harireddy090](#) • November 9, 2018



Getting error while doing Ansible-Playbook

host1 | FAILED! ⇒ { “changed”: false, “module_stderr”: “Shared connection to 172.31.8.238 closed.\r\n”, “module_stdout”: “/bin/sh: 1: /usr/bin/python: command not found\r\n”, “msg”: “MODULE FAILURE\r\nSee stdout/stderr for the exact error.” }

127 }

already i have python version Python 2.7.15rc1

[Reply](#)

[Chanoch](#) • October 22, 2018 ^

My experience was that I had to softlink python3 to python for this to work on Ubuntu 18.04 or ansible would complain that it couldn't find /usr/bin/python on each droplet:

```
root@my-droplet$ sudo ln -s /usr/bin/python3 /usr/bin/python
```

I also found that I needed to enter the pass phrase for the ssh cert once per server. After receiving the reply from the first server, I entered the pass phrase even though there was no prompt and this seemed to make it work.

It is probably better to configure the certificate at playbook level rather than globally in any case.

[Show replies](#) ▾ [Reply](#)

[ebostran10312010](#) • October 12, 2018 ^

hello dear people how can i get here your_server_ip (alias ansible_ssh_host = your_server_ip) Find Thnx

[Reply](#)



This work is licensed under a Creative Commons Attribution-NonCommercial-
International License

<https://www.digitalocean.com/community/tutorials/how-to-install-and-configure-ansible-on-ubuntu-18-04>

COOKIE PREFERENCES

Essayez DigitalOcean gratuitement

Cliquez ci-dessous pour vous inscrire et obtenir **200 \$ de crédit** pour essayer nos produits sur 60 jours!

[S'enregistrer →](#)

Sujets populaires

[Ubuntu](#)

[Principes de base de Linux](#)

[JavaScript](#)

[Réagir](#)

[Python](#)

[Sécurité](#)

[MySQL](#)

[Docker](#)

[Kubernetes](#)

[Parcourir toutes les balises de rubrique](#)

[Hébergement géré gratuit →](#)

[Tous les tutoriels →](#)

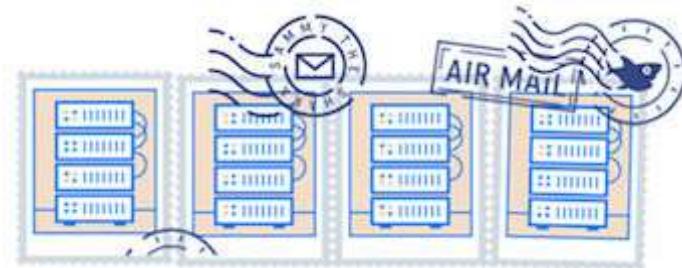
[Questionne](#)

[Forum Q&R](#)

[Poser une question](#)



Prise en charge de DigitalOcean



GET OUR BIWEEKLY NEWSLETTER

Sign up for Infrastructure as a
Newsletter.



HOLLIE'S HUB FOR GOOD

Working on improving health and
education, reducing inequality,
and spurring economic growth?

We'd like to help.



BECOME A CONTRIBUTOR

 COOKIE PREFERENCES

You get paid; we donate to tech
nonprofits.

Featured on Community [Intro to Kubernetes](#) [Learn Python 3](#) [Machine Learning in Python](#)
[Getting started with Go](#)

DigitalOcean Products [Virtual Machines](#) [Managed Databases](#) [Managed Kubernetes](#) [Block Storage](#)
[Object Storage](#) [Marketplace](#) [VPC](#) [Load Balancers](#)

Welcome to the developer cloud

DigitalOcean makes it simple to launch in the cloud and scale up as you grow – whether you're running one virtual machine or ten thousand.

[Learn More](#)



Company Products Community Solutions Contact

About	Products Overview	Tutorials	Website Hosting	Support
Leadership	Droplets	Q&A	VPS Hosting	Sales
Blog	Kubernetes	CSS-Tricks	Web & Mobile Apps	Report Abuse
Careers	App Platform	Write for DOnations		System Status
Customers				

COOKIE PREFERENCES

Partners	Functions	Currents	Game
Channel Partners	Cloudways	Research	Development
Referral Program	Managed Databases	Hatch Startup Program	Streaming
Affiliate Program	Spaces	deploy by DigitalOcean	VPN
Press	Marketplace	Shop Swag	SaaS Platforms
Legal	Load Balancers	Research	Cloud Hosting for Blockchain
Security	Block Storage	Program	Startup Resources
Investor Relations	Tools & Integrations	Open Source	
DO Impact	API	Code of Conduct	
	Pricing	Newsletter	
	Documentation	Signup	
	Release Notes	Meetups	
	Uptime		

© 2023 DigitalOcean, LLC. All rights reserved.

