
Généralités sur l'intranet

3.1. Qu'est-ce que l'intranet ?

C'est un réseau informatique au sein d'une même entreprise ou organisation, il se compose d'un ensemble de pages accessibles en réseau et un ensemble d'applications hébergées sur des serveurs.

Les ressources au sein de ce réseau ne sont pas consultables à partir d'autres postes de travail en dehors de l'entreprise. C'est donc l'équivalent d'un Internet propre à une entreprise. Il est utilisé par l'ensemble des travailleurs ou des collaborateurs de cette entreprise, il permet par exemple de :

- Déployer des fonctionnalités et des applications internes à l'entreprise ;
- Discuter à travers un chat ou un forum entre collaborateurs dispersés sur plusieurs sites ;
- Partager les agendas de l'ensemble des collaborateurs ;
- Consulter ses mails ;
- Échanger des documents et des fichiers entre collaborateurs ;
- Rechercher des documents à travers un moteur de recherche interne ;
- Gérer des projets ;
- Fournir un outil d'aide à la décision ;
- Mettre à disposition un annuaire du personnel ;
- Réunir plusieurs collaborateurs pour une visioconférence, ...

Cette liste non exhaustive de fonctionnalités donne un aperçu des possibilités d'un intranet. Un intranet peut-être plus ou moins présent dans les process d'une entreprise : les collaborateurs peuvent, depuis leurs postes de travail respectifs, accéder à des documents internes tels que des comptes-rendus de réunion, des grilles de rémunérations, des demandes de congé... C'est donc une solution pour accéder à l'intégralité du système informatique d'une entreprise au format universel du Web.

Un intranet est hébergé sur un ou plusieurs serveurs et peut être consulté depuis n'importe quel endroit dans le monde.

3.2. Quels objectifs pour l'intranet ?

Un intranet présente de nombreux avantages parmi lesquels on peut citer :

- Une centralisation des données ;
- Une fluidité d'accès à l'information ;
- Une mise à disposition des indicateurs clés & reporting ;
- Une optimisation de gestion des process interne.

3.3. Pourquoi adopter un intranet pour une entreprise ?

La mise en place d'un intranet dans une entreprise permet dans la grande majorité des cas des gains de productivité. Il s'avère également très précieux pour favoriser la collaboration de salariés dispersés sur plusieurs sites.

On peut aussi citer quelques raisons liées à l'adoption d'un intranet :

- Un déploiement et une administration centralisée des courriers électroniques ;
- La rapidité de la consultation et de la mise à jour de documents au format HTML ;
- Une mise à jour centralisée de l'ensemble des documents de l'entreprise ;
- La grande facilité de publication, de diffusion et de consultation de documents ;
- La transparence dans l'organisation du travail et la gestion de projets avec une visibilité sur l'état d'avancement ;
- Une meilleure gestion des ressources de l'entreprise ;
- Une bonne visibilité sur les disponibilités de chacun, ...

Chapitre 4 : Généralités sur Ansible

4.1. Qu'est-ce que l'automatisation ?

L'automatisation informatique, aussi appelée automatisation de l'infrastructure, consiste à utiliser des logiciels pour créer des instructions et des processus reproductibles dans le but de remplacer ou de réduire l'interaction humaine avec les systèmes informatiques. Les logiciels d'automatisation s'exécutent dans les limites de ces instructions, outils ou structures afin de réaliser des tâches avec une intervention humaine minimale, voire nulle.

L'automatisation est un élément clé de l'optimisation de l'environnement informatique et de la transformation numérique. Les environnements informatiques dynamiques et modernes doivent pouvoir évoluer plus rapidement que jamais, et l'automatisation informatique joue là un rôle essentiel.

4.2. Que couvre l'automatisation informatique ?

En théorie, toute tâche informatique peut être automatisée dans une certaine mesure. L'automatisation peut donc être intégrée et s'appliquer à toute tâche, de l'automatisation du réseau à la gestion des configurations et au déploiement d'applications en passant par l'approvisionnement du cloud, de l'infrastructure et des environnements d'exploitation standard.

Les fonctionnalités d'automatisation et les applications peuvent elles-mêmes s'étendre à des technologies spécifiques telles que les conteneurs, des méthodes comme le DevOps et des domaines plus vastes comme le cloud, l'edge computing, la sécurité, les tests et la surveillance ou les alertes.

4.3. L'automatisation : les avantages

Une approche globale de l'automatisation informatique peut délester l'équipe de l'entreprise de processus manuels et répétitifs. Elle permet ainsi d'augmenter la productivité des équipes, de réduire le nombre d'erreurs, d'améliorer la collaboration et de consacrer plus de temps à des tâches plus importantes.

4.4. Les taches de l'automatisation

4.4.1. Approvisionnement

L'approvisionnement figure parmi les tâches les plus lourdes, qu'il s'agisse d'approvisionner un système nu ou un cloud privé, hybride ou public. Le fonctionnement des systèmes métier nécessite la mise en place et la configuration d'une infrastructure adaptée. De nos jours, les racks, boîtiers et câbles des datacenters ont presque complètement cédé la place aux ressources virtualisées : réseaux, stockage et datacenters définis par logiciel, machines virtuelles et conteneurs.

La plupart des tâches réalisées aujourd'hui sont définies dans des logiciels, et cette transition vers les logiciels a permis d'accroître considérablement le potentiel d'évolutivité et de capacité. Cette transition permet et, nécessairement, exige aussi la codification de processus et vous aide ainsi à satisfaire les besoins de votre entreprise tout en maîtrisant les coûts et en respectant les délais.

C'est justement à ce niveau qu'intervient l'automatisation. Pourquoi perdre du temps à configurer manuellement ces environnements avec des modèles ? Avec la codification, vous avez à votre disposition un modèle à suivre pour réaliser des tâches. Alors pourquoi ne pas transmettre ces règles à un système automatisé pour qu'il les exécute ? Automatisez les déploiements dans votre datacenter à l'aide de solutions compatibles avec votre infrastructure et vos outils de gestion existants afin de tirer pleinement parti des ressources dont vous disposez et d'atteindre vos objectifs futurs.

4.4.2. Gestion des configurations

Toutes les applications ne sont pas créées de la même manière. Elles peuvent avoir besoin d'éléments qui diffèrent : paramètres, systèmes de fichiers, ports, utilisateurs, etc. Une fois l'approvisionnement automatisé, on doit être en mesure d'indiquer à ces ressources ce qu'elles doivent faire. L'enregistrement de la définition de notre environnement d'applications dans un document, une feuille de calcul, un fichier texte ou même un e-mail ne permettra pas de mettre en place un environnement solide et reproductible pour héberger les applications. Face à la multiplication des systèmes et des instances et à l'augmentation de la complexité, on doit adopter un moyen plus approprié d'enregistrer les informations relatives à vos systèmes pour les gérer plus efficacement.

On a besoin pour cela d'une solution robuste de gestion des configurations, qui permet aux développeurs de définir l'infrastructure (système nu, ressources virtualisées, cloud,

conteneurs, etc.) de manière simple et compréhensible pour tous les membres de votre équipe informatique. Plus l'automatisation de pratiques et de scripts adaptés à la gestion des systèmes sera simple, plus le travail sera facilité.

4.4.3. Orchestration

Il y a de fortes chances que l'on ne déploie pas qu'un service sur une seule machine. L'environnement informatique est probablement bien plus complexe. Il nécessite sans doute de gérer et d'assurer la maintenance d'une multitude d'applications dans plusieurs datacenters et infrastructures. Sans oublier les déploiements de clouds privés, publics et hybrides.

Plus l'environnement informatique est complexe, plus la gestion de tous ses éléments mobiles l'est, elle aussi, et plus on a besoin de combiner des tâches automatisées et leurs configurations dans différents groupes de systèmes et de machines. C'est le principe même de l'orchestration. Cerise sur le gâteau, on a la possibilité de contrôler ces orchestrations avec des solutions d'automatisation robustes. Ceci nous permet d'en effectuer le suivi, de les relier les unes aux autres et d'exécuter des systèmes autonomes et plus avancés, en toute simplicité.

4.4.4. Déploiement d'application

Qu'on utilise une méthode traditionnelle de déploiement d'applications ou qu'on suive une approche de CI/CD (intégration et déploiement continu), les pipelines de développement doivent s'appuyer sur des systèmes automatisés robustes pour satisfaire les nouvelles exigences. Le succès du déploiement d'applications dépend donc de l'efficacité d'un ensemble de fonctionnalités et de tâches essentielles automatisées, en particulier lors de la phase de tests. L'automatisation peut nous aider à passer des phases de validation et de création à celles de tests et de déploiement d'une manière codifiée, fiable et éprouvée. Cette démarche réduit les risques d'erreur humaine tout en améliorant l'efficacité et la rapidité.

L'automatisation informatique nous permet de déployer des applications en toute confiance, de configurer dès le début les services requis et de lancer des applications et leurs artefacts selon une approche commune, à la fois transparente et compréhensible par tous les membres de l'équipe informatique.

4.4.5. Sécurité et conformité

Définir des politiques de sécurité, de conformité et de gestion des risques, les mettre en application et corriger les problèmes en intégrant ces politiques dans des étapes automatisées à l'échelle de l'infrastructure. Placer la sécurité au cœur de vos processus informatiques et soyez plus proactif grâce à l'automatisation.

La standardisation des workflows et des processus de sécurité facilite la mise en conformité et la réalisation d'audits. On sait exactement comment les politiques sont appliquées et on peut en vérifier la cohérence. On peut également mettre facilement en œuvre de nouvelles exigences de conformité dans l'ensemble de l'environnement informatique.

4.5. Les outils d'automatisation



Figure 12 : Outils d'automatisation

Source : <https://www.veritis.com/>

Il existe sur le marché beaucoup d'outils d'automatisation parmi lesquels on a :

➤ Ansible

Ansible est un logiciel libre et idempotent d'automatisation, de déploiement et de gestion des configurations sous licence GNU/GPL, écrit en Python et est multiplateforme (GNU/Linux, Microsoft Windows).

Ce logiciel est maintenu par son créateur Michael DeHaan et sa communauté d'utilisateur qui comprend, entre autres Red Hat et Hewlett-Packard ;

Il existe une version graphique et payante d'Ansible : Ansible Tower. Cette version est soutenue par RedHat et offre des fonctions de contrôle d'accès basé sur les rôles, de planification des tâches et de gestion graphique des inventaires.

La plus petite unité de valeurs d'Ansible est une tâche décrite en syntaxe YAML, un ensemble de tâches constitue un play et un ensemble de play constitue un playbook. Il est aussi possible d'utiliser la syntaxe Jinja2 dans des templates ou filtres. Dans les tâches ou les templates, il est possible d'utiliser des variables que l'on peut créer soi-même ou utiliser les « facts ».

Ces facts sont un ensemble de variables décrivant les spécificités d'une machine.

Ansible dispose aussi d'un certain nombre de modules pouvant être exécutés directement sur les machines distantes ou par les Playbook. Ces modules sont assez variés et permettent plusieurs interactions.

Les principaux points positifs d'Ansible sont :

- La clarté de la documentation officielle et sa facilité d'installation (Ansible ne nécessite pas d'installation d'agent supplémentaire sur les machines distantes.) ;
- La communication entre les machines se fait par le biais de tunnel SSH. Et le seul prérequis pour permettre cette communication est le déploiement de la clef publique de l'utilisateur Ansible sur les machines distantes et la présence de python 2.7 et sudo sur les machines distantes ;
- Le fait qu'Ansible soit écrit dans un langage assez connu. Cela permet si le besoin se fait sentir de développer plus rapidement et efficacement nos propres modules ;
- Cela est moins contraignant que de déployer un agent supplémentaire sur des machines.

➤ Puppet

Puppet est un logiciel, sous licence Apache 2, permettant d'automatiser un grand nombre de tâches d'administration, comme l'installation de logiciels, de services ou encore de modifier des fichiers. Puppet fonctionne sous le mode client /serveur, il a été écrit en Ruby ce qui lui permet d'être multi-plate-forme (BSD, SUN, Linux, Windows). Les échanges entre le maître et les esclaves se font par le biais de communications HTTPS. La société éditant Puppet, Reductive Labs, dispose d'une version commerciale permettant en plus des fonctionnalités de la version open-source :

- d'avoir un support sur Puppet et les modules maintenus par Reductive Labs ;
- d'avoir une interface graphique de gestion ;
- de gérer les machines virtuelles Vmware ;
- ...

En plus de Puppet, Reductive Labs a développé le logiciel nommé Factor. Celui-ci permet de lister les éléments propres aux systèmes administrés, comme le nom de la machine, les adresses MAC et IP, le système d'exploitation ainsi que les variables d'environnement utilisables dans les « templates » de Puppet.

Ces éléments peuvent ensuite être utilisés dans les « templates » Puppet sous la forme de variable. Il est très souvent utilisé par les concurrents pour la même fonction. Comme bon nombre de ses concurrents, Puppet dispose d'une « forge » permettant de partager, rechercher et récupérer des modules.

➤ Chef

Tout comme Puppet, Chef est aussi un outil de gestion de configuration multi-plateforme, écrit en Ruby et sous licence Apache 2. Chef dispose d'un serveur central qui détient les configurations des machines clientes. Comme Puppet, il est nécessaire d'installer un agent sur chaque machine que l'on veut configurer avec ce logiciel.

Après l'installation du client, les machines esclaves vont comparer leur configuration par rapport à celles référencées sur le serveur maître et effectuent si besoin les changements nécessaires pour coller à la configuration décrite dans le serveur maître. Comme le nom du logiciel, l'appellation des outils est en rapport avec la cuisine. Ainsi, l'installation d'un programme est décrite par une recette (« recipes » en anglais).

On trouve par exemple une recette pour « MySQL », une autre pour « Apache » etc. L'ensemble des recettes, pour configurer un logiciel, est contenu dans un livre de recette (« cookbook » en anglais). « Knife », traduction du mot français « couteau » en anglais, est l'outil pour interagir avec Chef. Nous aurons aussi les « rôles » qui désignent l'ensemble des livres de recettes ayant la même finalité.

Par exemple, le rôle « serveur web » décrira les recettes pour installer le serveur web mais aussi la configuration du pare-feu de la machine.

Les ingénieurs PROJIXI Europe intègrent des solutions propriétaires et open source chez des grands comptes, grâce à leurs expériences sur des projets significatifs.

4.6. Qu'est-ce que Ansible ?

Dans l'environnement informatique actuel, les applications d'entreprise peuvent être complexes, évolutives, distribuées, basées sur des composants et sont souvent critiques. Ils peuvent être déployés sur une variété de plates-formes dans le cloud privé, le cloud public ou le cloud hybride. Ils peuvent accéder à des données sensibles, ils peuvent être soumis à des directives réglementaires et à des politiques de sécurité strictes, et doivent cependant être aussi conviviaux que possible. En bref, ces applications sont très complexes. Il est nécessaire de voir la façon dont ces considérations s'harmonisent avec l'utilisation d'outils d'automatisation comme Ansible.

Ansible est un logiciel libre de gestion des configurations qui automatise le déploiement des applications et la livraison continue des mises à jour. Disponible sous licence GPL v3, il s'adosse au protocole de cryptage réseau SSH (pour Secure Socket Shell) pour déployer les mises en production de code via des fichiers décrivant les configurations applicatives cibles au format Json (pour JavaScript Object Notation).

On peut utiliser Ansible pour automatiser trois types de tâches :

- Provisioning : configurer les différents serveurs dont on a besoin dans une infrastructure ;
- Configuration : modifier la configuration d'une application, d'un système d'exploitation ou d'un périphérique, démarrer et arrêter les services, installer ou mettre à jour des applications, mettre en œuvre une politique de sécurité, ou effectuer une grande variété d'autres tâches de configuration ;
- Déploiement d'applications : adopter une démarche DevOps en automatisant le déploiement d'applications développées en interne sur nos environnements de production.

4.7. Quelques notions d'Ansible

➤ Module

Un module est un programme utilisé pour exécuter une tâche ou une commande Ansible. Chaque tâche utilise un module et un seul, qui peut prendre des

arguments pour être exécuté de manière personnalisée. Ansible fournit de nombreux modules, mais vous pouvez créer le vôtre, personnalisé.

➤ Rôle

Un rôle est une structure arborescente constituée de répertoires et de fichiers de configuration YAML, qui vont avoir pour fonction d'installer tel ou tel système. Les rôles peuvent être imbriqués et interdépendants les uns des autres.

Un rôle est donc un ensemble de fichiers organisés dans une structure arborescente.

Le but des rôles est de pouvoir agglomérer des opérations cohérentes (dans les fichiers YAML), afin de pouvoir les réutiliser de façon modulaire. On peut voir un rôle comme un ensemble d'opérations qui ont un rôle commun, comme par exemple le rôle d'installer Apache, ou le rôle de configurer MariaDB.

Les répertoires sont tous optionnels, excepté le répertoire **tasks** qui doit contenir le fichier **main.yml**. Ansible va traiter en premier ce fichier à l'appel d'un rôle.

➤ Tache

Une tâche est une instruction décrite en YAML dans un fichier de configuration. Chaque tâche utilise un module ainsi que quelques éventuels arguments supplémentaires.

➤ Playbook

Un playbook est un fichier de configuration YAML contenant une suite de jeux d'instructions, ou plays en anglais. Chacun peut être constitué d'options, et fait appel à un ou plusieurs rôles. Il permet de décrire une stratégie de déploiement, ou de configuration, en structurant les actions nécessaires.

En utilisant les playbooks, on a la possibilité de conserver le code dans un fichier et de le réutiliser à votre façon, contrairement à la commande ansible qui est volatile

De façon schématique, on peut retenir que :

- Un rôle contient un ou plusieurs fichiers de configuration (YAML) ;
- Un fichier de configuration contient une ou plusieurs tâches ;
- Une tâche fait appel à un module.

4.8. Fonctionnement et avantages d'Ansible

Dans Ansible, il existe deux catégories d'ordinateurs : le nœud maître (master) et les nœuds esclaves (slaves). Le nœud maître est une machine sur laquelle est installé l'outil Ansible. Il doit y avoir au moins un nœud maître, bien qu'un nœud maître de sauvegarde puisse également exister.

Ansible fonctionne en se connectant aux nœuds en SSH et en y poussant de petits programmes, appelés modules. Ces modules sont définis dans un fichier nommé le Playbook. Le nœud maître, se base sur un fichier d'inventaire qui fournit la liste des hôtes sur lesquels les modules Ansible doivent être exécutés.

Ansible exécute ces modules en SSH et les supprime une fois terminer. La seule condition requise pour cette interaction est que le nœud maître Ansible dispose d'un accès de connexion aux nœuds esclaves. Les clés SSH sont le moyen le plus courant de fournir un accès, mais d'autres formes d'authentification sont également prises en charge.

Voici un schéma qui reprend notre explication :

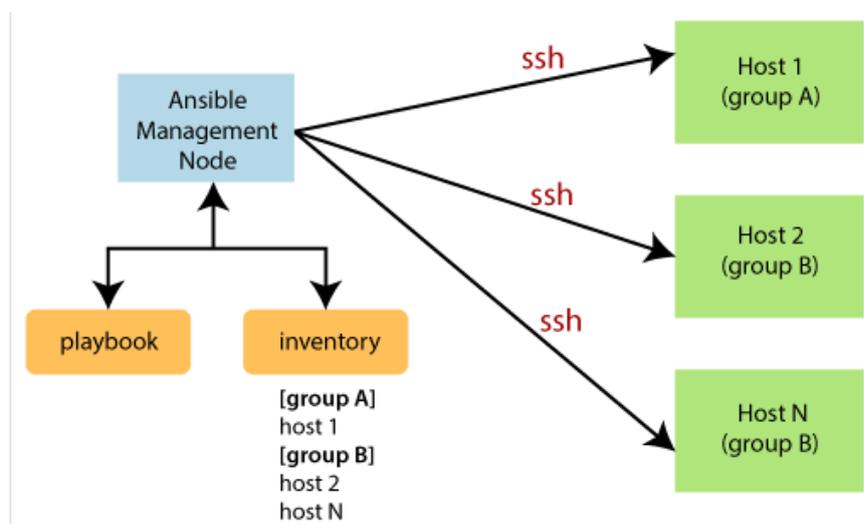


Figure 13 : Fonctionnement d'Ansible

Source : <https://devopssec.fr/>

Voici une liste des avantages d'Ansible :

- Gratuit : Ansible est un outil open source ;
- Simple : Ansible utilise une syntaxe simple écrite en YAML. Aucune compétence en programmation particulière n'est nécessaire pour créer les playbooks d'Ansible. Il est également simple à installer ;
- Puissant : Ansible vous permet de modéliser des workflows très complexes ;
- Flexible : Ansible fournit des centaines de modules prêts à l'emploi pour gérer nos tâches, quel que soit l'endroit où ils sont déployés. On peut réutiliser le même playbook sur un parc de machines Red Hat, Ubuntu ou autres ;
- Agentless : on n'a pas besoin d'installer d'autres logiciels ou d'ouvrir des ports de pare-feu supplémentaires sur les systèmes clients que vous souhaitez automatiser. Ansible réduit encore l'effort requis pour que notre équipe commence à automatiser immédiatement ;
- Efficace : Parce qu'on n'a pas besoin d'installer de logiciel supplémentaire, il y a plus de place pour les ressources d'application sur notre serveur.

4.9. Que peut faire Ansible ?

Ansible peut être utilisé de différentes manières. En voici quelques-unes ci-dessous :

4.9.1. Déploiement d'application

Ansible permet de déployer rapidement et facilement des applications à plusieurs niveaux. On n'a pas besoin d'écrire du code personnalisé pour automatiser les systèmes ; on liste les tâches à effectuer en écrivant un playbook, et Ansible trouvera comment amener les systèmes à l'état dans lequel on veut qu'ils soient. En d'autres termes, on n'aura pas à configurer manuellement les applications sur chaque machine. Lorsque on exécute un playbook à partir de votre machine de contrôle, Ansible utilisera le protocole SSH pour communiquer avec les hôtes distants et exécuter toutes les tâches (Tasks).

4.9.2. Orchestration

Comme son nom l'indique, l'orchestration consiste à amener différents éléments à interagir ensemble sans incohérence. Par exemple, avec le déploiement d'applications, on doit gérer non seulement les services frontend, mais également les services backend comme les bases de données, le réseau, le stockage, etc. on doit également nous assurer que toutes les tâches sont gérées dans le bon ordre. Grâce à Ansible on orchestre les éléments de notre infrastructure à

l'aide des playbooks Ansible, et on peut les réutiliser sur différents types de machines, grâce à la portabilité des modules Ansible.

4.9.3. Sécurité et conformité

Comme pour le déploiement d'applications, des politiques de sécurité de l'entreprise (telles que des règles de pare-feu ou le verrouillage des utilisateurs) peuvent être mises en œuvre avec d'autres processus automatisés. Si on configure les détails de sécurité sur la machine de contrôle et exécute le playbook associé, tous les hôtes distants seront automatiquement mis à jour avec ces détails. Cela signifie qu'on n'aura pas besoin de surveiller chaque machine pour vérifier la conformité de la sécurité en continu manuellement. De plus, tous les identifiants (identifiants et mots de passe des utilisateurs admin) qui sont stockés dans les playbooks ne sont récupérables en brut par aucun utilisateur.

4.9.4. Provisionnement du cloud

La première étape de l'automatisation du cycle de vie de vos applications consiste à automatiser l'approvisionnement de votre infrastructure. Avec Ansible, on peut provisionner des plateformes cloud, des hôtes virtualisés, des périphériques réseau et des serveurs physiques.