# How to work with Ansible

## Introduction

Ansible is a simple and powerful automation language for IT applications. The Ansible automation can be managed and controlled with a user interface.

In addition, the Ansible tool does not require special coding skills, the automation is human readable and the tasks are executed in order.

With this tool the configuration of a great number of hosts (depending on the user’s needs) can be done at the same time for all of them. The instances that are automated do not require additional software.

In conclusion, the benefits of using Ansible are productiveness, control over the application lifecycle, security and effectiveness.

## What does Ansible do?

As previously mentioned, Ansible is an automation platform which can be used for a multiple of features:

* *Provisioning:* The process of creating virtual or cloud instances from templates or booting servers or VMs can be automated with Ansible;
* *Configuration management:* A predefined set of servers can be configured using a centralized configuration file;
* *Application deployment:* The lifecycle of an application can be effectively managed using the Ansible tool;
* *Security and Compliance:* The security policy can be defined and integrated into automated Ansible processes so that scanning and remediation would be an integral part with everything that is deployed;
* *Orchestration:* All the multiple automated configurations are managed by Ansible in terms of how they interact and how they make the system work as a whole.

## How does it work?

Ansible is a simple automation engine which does not require additional security infrastructure. Ansible models an IT infrastructure by describing how all the system inter-relate, not just one system at a time.

After connecting to the nodes of the network, using small programs called Ansible modules, Ansible executes the instructions written there and removes them after the process is completed.

This way the libraries of the modules can reside-on any machine and there is no need of servers, daemons or databases.

The way Ansible connects to the remote machines is through SSH by default. Although passwords are supported, it is preferred to use SSH keys.

In order to represent the systems in the network, it is necessary to define an inventory where hosts are listed and variables can be assigned to them in simple text files or directories.

## **How to configure a Ansible server** /Configuration process

The first step in order to configure the Ansible server is to install it. If you want to install it on a virtual machine, a Google Cloud Platform (GCP) or an Amazon Web Services cloud (AWS) that are based on a Fedora distribution (Rat Hat, CentOS etc.), you have to run the following command:

yum –y install ansible

and to be sure that the installation was successful you can run:

ansible --version

If you want to use a machine that runs Debian based distribution, you have to run:

apt-get -y install ansible

In the present configuration it will be presented the steps that need to be followed in order to install Ansible on CentOS virtual machine. Before installing Ansible, it is necessary to install some additional software that Ansible depends on.

Firstly it will be installed the EPEL repository. This additional package provides easy access to install packages for commonly used software. The command used to install the EPE repository is:

sudo yum install epel-release

In order to confirm that the package was successfully installed, the repository list can be refreshed using the following command:

sudo yum repolist

After this step there was necessary to install some Python tools and libraries:

yum install gcc git python-pip python-devel libffi-devel openssl-devel

Another set of tools was downloaded and installed using the following command:

pip install setuptools --upgrade

After installing the per-requisite software, the latest release of Ansible was downloaded and installed:

git clone https://github.com/ansible/ansible.git --recursive

cd ansible

make install

By running the “ansible” command you can check that it was installed correctly.

After having the confirmation that the installation process worked properly, it would be necessary to do a basic connectivity test. At this point when running the “ansible” command without arguments it would display an error because there are not any target hosts specified. Usually, when you work with Ansible, it is necessary to define a hosts file where you detail the target servers that Ansible will communicate with, but a connectivity test can be done only specifying an IP address of a target host as an argument from the command line. In order to do so, it will be necessary to establish SSH key authentication.

This can be done by running the following command on the Ansible machine:

ssh-keygen

This command will generate a new public/private rsa key pair. After this the public key needs to be added to the target host’s ssh configuration file.

After installing and testing the Ansible configuration on the machine, the nest step is to use it to automate different processes on a group of different hosts.

To do so, it is needed to create playbooks. They contain a list of commands intended to be applied on inventories. The inventories are a defined group of remote hosts, for example they can contain a static list of servers, ranges, dynamic lists of servers like AWS, Azure, GCP etc. The playbooks can be used to build entire application environment, to describe the desired state of something. In addition, they don’t require any special coding skills and are human and machine readable.

Another useful component of the Ansible tool are the variables. They can alter the way playbooks run.

After configuring all of components mentioned before, the Ansible can be run following the steps below:

1. Ad-hoc: ansible <inventory> -m
2. Playbooks: ansible-playbook
3. Automation Framework: Ansible Tower

## Architecture diagram

Templates

Playbooks

Modules

System Admin

Delivery Team

Test Team



Pool of running servers

ANSIBLE

Instances Creation

Cloud Provisioning

Configuration Management

Application Deployment

Intra-service orchestration