# Ansible

# Learn by doing

## 1° Deploying Ansible

You have been tasked with putting together a presentation to demonstrate how Ansible may be used to install software on remote hosts automatically. Before the demo, you will need to configure your test systems. You have been provided two hosts called control and workstation. You will need to configure the ansible user on workstation to have sudo access without a password to the automated software installed via Ansible. You must also configure the control host as your Ansible control server by installing Ansible on it as well as configuring the Ansible user with a pre-shared key to login to the workstation host as the ansible user.

Once the basic configuration is complete, you will need to create a simple inventory in /home/ansible/inventory on the control server containing the workstation host. Afterward, you will write a simple playbook in /home/ansbile/git-setup.yml on the control host that installs git on the workstation host. You will need to make sure the playbook works by running it from the control server.

Summary tasks list:

* Install Ansible on the control host.
* Create an ansible user on both the control host and workstation host.
* Configure a pre-shared key for Ansible that allows the user to log in from control to workstation without a password.
* Configure the Ansible user on the workstation host so that Ansible may sudo without a password.
* Create a simple inventory in /home/ansible/inventory consisting of only the workstation host.
* Write and execute an Ansible playbook in /home/ansible/git-setup.yml on the control node that installs git on the workstation host.

1° Install Ansible on Centos

sudo yum install epel-release

sudo yum install ansible

2° Create an ansible user on both the control host and workstation and set them a password

sudo –i

adduser ansible

passwd ansible

3° Configure a pre-shared key for Ansible that allows the user to log in from `control` to `workstation` without a password.

su – ansible

ssh-keygen

ssh-copy-id NameOfServer

Assuming you are logged into control as cloud\_user, run the following commands providing the appropirate passwords when prompted and default options otherwise:

sudo -i -u ansible (provide cloud\_user a sudo password)

ssh-keygen (accept default options by pressing *enter* )

ssh-copy-id workstation (provide ansible user a password)

logout

4° Configure the Ansible user on the workstation host so that Ansible may sudo without a password

sudo visudo

ansible ALL=(ALL) NOPASSWD: ALL

5° Create a simple inventory in /home/ansible/inventory consisting only the workstation host

vim /home/ansible/inventory

6° Write an Ansible playbook in `/home/ansible/git-setup.yml` on the control node that installs `git` on `workstation` then execute the playbook.

---

- name: Installation de git

hosts:

become: yes

tasks:

- name: install git

yum:

name: git

state: latest

…

Dans le dossier en question :

Run ansible-playbook -i inventory git-setup.yml.

Getting Started with Ansible

Additional Information and Resources

Your CIO has greenlit a proof of concept for Ansible in your environment. You are to set up an Ansible control node in a test environment and verify basic functionality. You have three demo hosts, one to be the control node (control1), and two to serve as managed nodes (node1 and node2). You must complete the following steps:

1. Install Ansible on the control node.
2. Configure the ansible user on the control node for ssh shared key access to managed nodes.  
   **Note**: do not use a passphrase for the key pair.
3. Create a simple Ansible inventory on the control node in /home/ansible/inventory containing node1 and node2.
4. Configure sudo access for Ansible on node1 and node2 so that Ansible may use sudo for any command with no password prompt.
5. Verify each managed node can be accessed by Ansible from the control node using the ping module. Redirect the output of a successful command to /home/ansible/output.

Important Notes:

* The user ansible is already present on all servers for your convenience.
* The ansible user has the same password as the cloud\_user.
* /etc/hosts entries are present on control1 for the managed nodes.

**1°Install Ansible on the control node.**

To install Ansible on the control node, run sudo yum install ansible.

**2°Configure the `ansible` user on the control node for ssh shared key access to managed nodes. Do not use a passphrase for the key pair.**

1. To create a keypair for the ansible user on the control host, run the following:
   * sudo su - ansible
   * ssh-keygen (accept all defaults: press **enter** for each prompt)
2. Copy the public key to both node1 and node2.
3. As the ansible user on the control host:
   * ssh-copy-id node1 (accept the host key if prompted, authenticate as ansible user)
   * ssh-copy-id node2 (accept the host key if prompted, authenticate as ansible user)

**3°Create a simple Ansible inventory on the control node in `/home/ansible/inventory` containing `node1` and `node2`.**

On the control host:

1. sudo su - ansible (if not already ansible user)
2. touch /home/ansible/inventory
3. echo "node1" >> /home/ansible/inventory
4. echo "node2" >> /home/ansible/inventory

**4°Configure sudo access for Ansible on `node1` and `node2` such that Ansible may use sudo for any command with no password prompt.**

Log in to node1 as cloud\_user and edit the sudoers file to contain appropriate access for the ansible user:

* ssh cloud\_user@node1
* sudo visudo
* Add the following line to the file and save:

ansible ALL=(ALL) NOPASSWD: ALL

**5°Verify each managed node is able to be accessed by Ansible from the control node using the `ping` module. Redirect the output of a successful command to `/home/ansible/output`.**

* To verify each node, run the following as the ansible user from the control host:
  + ansible -i /home/ansible/inventory node1 -m ping
  + ansible -i /home/ansible/inventory node2 -m ping
* To redirect output of a successful command to /home/ansible/output:
  + ansible -i /home/ansible/inventory node1 -m ping > /home/ansible/output

# File Manipulation with Ansible

Additional Information and Resources

The development team has released the latest build of the enigma software. You have been approached to install the latest version on the QA systems. The development team has provided a task list of how to install the build. Rather than run through the steps on all the servers in your environment, you can use Ansible to run each task on all the servers at one time!

The Ansible control node has been configured for you and each QA server has already been configured for use with Ansible. The default inventory has been configured to include a qa-servers host group which includes the host that requires the new software. Execute the task list provided below using the appropriate Ansible ad-hoc commands.

*SPECIAL NOTE:* Although you will initially log in as the 'cloud\_user', please 'sudo su -' to the 'ansible' user in order to complete the solution. All the SSH keys have been created and exchanged among the hosts in order to allow the tasks to be run appropriately.

**Note:** This URL only works correctly when accessed on the lab servers. The dmain name DNS entry is overridden in /etc/hosts on each lab server. If you attempt to access it from another system, you will reach a parked domain.

Installation task list:

* Download <http://software.xyzcorp.com/enigma.tgz> to /tmp on each host in qa-servers and verify the sha256 checksum via <http://software.xyzcorp.com/enigma-checksum.txt>. (**Note:** There is a bug in ansible get\_url that currently only allows using the literal checksum value instead of a file location for the checksum argument! Thus, you must copy the checksum value from the provided checksum file and then paste that value into the ansible command rather than simply providing the checksum file URL in the command.)
* Extract /tmp/enigma.tgz to /opt/ on all hosts in qa-servers.
* Update the line of text "DEPLOY\_CODE" in /opt/enigma/details.txt to the "CODE\_RED" on each server in qa-servers.
* Set the group ownership of the directory /opt/enigma/secret/ and each file contained within the directory so that the group owner is protected for each host in qa-servers.
* Delete the file /opt/enigma/tmp/deployment-passwords.txt from all servers.

Learning Objectives

1° Become 'ansible' user and then download http://software.xyzcorp.com/enigma.tgz to `/tmp` on each host in qa-servers and verify the sha256 checksum via http://software.xyzcorp.com/enigma-checksum.txt.

2°Extract `/tmp/enigma.tgz` to `/opt/` on all hosts in qa-servers.

3°Update the line of text "DEPLOY\_CODE" in `/opt/enigma/details.txt` to the "CODE\_RED" on each server in qa-servers.

4°Set the group ownership of the directory `/opt/enigma/secret/` and each file contained within the directory so that the group owner is `protected` for each host in qa-servers.

5°Delete the file `/opt/enigma/tmp/deployment-passwords.txt` from all servers.

Correction

**Become 'ansible' user and then download http://software.xyzcorp.com/enigma.tgz to `/tmp` on each host in qa-servers and verify the sha256 checksum via http://software.xyzcorp.com/enigma-checksum.txt.**

**Note:** This URL only works correctly when accessed on the lab servers. The dmain name DNS entry is overridden in /etc/hosts on each lab server. If you attempt to access it from another system, you will reach a parked domain.

On EACH of the hosts in 'qa-servers' inventory, become the 'ansible' user:

sudo su - ansible

As the 'ansible' user (above), run the following commands on each host in the 'qa-servers':

* CHECKSUM=$(curl http://software.xyzcorp.com/enigma-checksum.txt | cut -f1 -d' ')

Ou sinon tu fais :

 curl <http://software.xyzcorp.com/enigma-checksum.txt>

TU aurais ensuite une suite de nombre: 91c5d51081da…… enigma.tgz

Tu copies tout ce qu’il y a avant enigma.

* ansible qa-servers -m get\_url -a "url=http://software.xyzcorp.com/enigma.tgz dest=/tmp/enigma.tgz checksum=sha256:$CHECKSUM"

Ou sinon tu tapes ceci:

* ansible qa-servers -m get\_url -a "url=http://software.xyzcorp.com/enigma.tgz dest=/tmp/enigma.tgz checksum=sha256: 91c5d51081da……"

**Extract `/tmp/enigma.tgz` to `/opt/` on all hosts in qa-servers.**

Run ansible qa-servers -b -m unarchive -a "src=/tmp/enigma.tgz dest=/opt/ remote\_src=yes"

**Update the line of text "DEPLOY\_CODE" in `/opt/enigma/details.txt` to the "CODE\_RED" on each server in qa-servers.**

Run ansible qa-servers -b -m lineinfile -a "regexp=DEPLOY\_CODE line=CODE\_RED path=/opt/enigma/details.txt"

**Set the group ownership of the directory `/opt/enigma/secret/` and each file contained within the directory so that the group owner is `protected` for each host in qa-servers.**

Run ansible qa-servers -b -m file -a "recurse=yes state=directory path=/opt/enigma/secret group=protected"

**Delete the file `/opt/enigma/tmp/deployment-passwords.txt` from all servers.**

Run ansible all -b -m file -a "state=absent path=/opt/enigma/tmp/deployment-passwords.txt"

A revoir:

* + Checksum
  + Remote\_src=yes
  + Regexp dans lineinfile

# Ad-Hoc Ansible Commands

Additional Information and Resources

Some consultants have been employed to perform audits on a number of systems in your company's environment. You must create the user accounts noted in */home/ansible/userlist.txt* and set up the provided public keys for their accounts. The security team has built a jump host for the consultants to access production systems and provided the full key-pair to you so you may set up and test the connection. All hosts in dbsystems will need the provided public key installed so the consultants may use key-pair authentication to access the systems. Also, you must ensure the auditd service is enabled and running on all systems.

To summarize, you must do the following:

1. Create the user accounts noted in /home/ansible/userlist.txt.
2. Copy the authorized\_keys file for each user to the correct location so the new accounts can log in with ssh key authentication.
3. Ensure auditd is enabled and running on all systems.

Important notes:

* For your convenience, Ansible is already on the control node. If you connect to the server by clicking on the Public IP address in your browser, make sure to change to the ansible user with the su - ansible command.
* The user ansible is present on all servers with appropriate shared keys for access to managed servers from the control node. Make sure to use this user to complete the commands.
* The ansible user has the same password as cloud\_user.
* The default Ansible inventory has been configured for you with the appropriate hosts and groups.
* /etc/hosts entries are present on control1 for the managed servers.

Correction

**Create the User Accounts Noted in `/home/ansible/userlist.txt`**

* ansible dbsystems -b -m user -a "name=consultant"
* ansible dbsystems -b -m user -a "name=supervisor"

**Place Key Files in the Correct Location, `/home/$USER/.ssh/authorized\_keys`, on Hosts in `dbsystems`**

* ansible dbsystems -b -m file -a "path=/home/consultant/.ssh state=directory owner=consultant group=consultant mode=0755"
* ansible dbsystems -b -m copy -a "src=/home/ansible/keys/consultant/authorized\_keys dest=/home/consultant/.ssh/authorized\_keys mode=0600 owner=consultant group=consultant"
* ansible dbsystems -b -m file -a "path=/home/supervisor/.ssh state=directory owner=supervisor group=supervisor mode=0755"
* ansible dbsystems -b -m copy -a "src=/home/ansible/keys/supervisor/authorized\_keys dest=/home/supervisor/.ssh/authorized\_keys mode=0600 owner=supervisor group=supervisor"

**Ensure `auditd` Is Enabled and Running on All Hosts**

ansible all -b -m service -a "name=auditd state=started enabled=yes"

1