Mastering

ANSIBLE

Section 01

What is Ansible and how to install it

Software provisioning evolution

- Long ago, their used to be a clear distinction between a software developer and a system administrator/operator.
- The role of the first was to code and test while the that of the second was to avail the necessary infrastructure and deploy the application.
- As time passed, technology evolved and - as a result - the number of servers that need to be managed increased significantly.
- On the other hand, new development methodologies were introduced like the Agile method.
- Those methodologies required a minimum application life cycle, and more and more environments availed continuously.
- To address those necessities, aiding tools arose to facilitate and shorten the application life cycle, and a new buzzword surfaced: DevOps.

So what is DevOps anyway?

- This is one of the terms for which you will hear different definitions depending on whom you are asking. Just like Big Data and Cloud Computing.
- DevOps in its core is a concept that eliminates the difference between developers and operators (hence the name). Both teams should be able to use a set of tools that helps shorten the application life cycle and ease the overall development/testing/deployment process.
- DevOps tools have many categories. Let's have an example:
  - Version control: like Git
  - Continuous integration: like Jenkins
  - Testing: like JUnit
  - Provisioning and configuration management: Vagrant, Chef, Puppet, Salt and Ansible

Along came Ansible

- Ok, if you are like me: interested in name origins, *Ansible* is a fictional device that was capable of transmitting messages at a speed faster than light. The Sci-Fi author who first coined it stated that it was derived from the word "answerable".
- Back to Ansible, the topic of this course, it is a configuration management and IT automation tool that was released in 2012 by Michael DeHaan.
- He was looking for a tool that can combine the roles done by disparate tools. Configuration management, server deployment, and sending arbitrary commands to a remote server (like through SSH).
- Ansible does all this without the need to install an agent or open extra ports on your servers.

The need for a CM tool

- You might be asking: "and what's wrong with plain old shell scripts?" if you are a veteran sysadmin, you'll sure be more comfortable with the way you manage your servers using SSH and shell scripts.
- However, in the era of cloud computing and the need for "instant provisioning", shell scripting might fall short when addressing the following:
  - Change documentation: while shell scripts allow you to add comments to your commands, this is still far for being a change management document.
  - Idempotence: this is a fancy term that is used to describe the ability of operations to be run multiple times yet yielding the same results. A CM tool should not produce different results run into errors when run multiple times. If you used shell scrips extensively you'd know how challenging this might get.
- A CM tool addresses the above requirements. Additionally, it provides a lot more enhancements that will make automating your infrastructure both easy and fun.

Ansible compared to other tools

- I am not going to recommend Ansible over other tools of the same category as, by doing this, I will be misleading you.
- Choosing the *right* tool for the task at hand is what you need to know. Neither Ansible nor any other tool that's discussed below is a silver bullet (flawless and suitable for any situation).
- Having said that, let's see where Ansible stands among other tools of the same type like Chef for an example:
  - It does not require an agent to be installed on the client. Chef requires a client to be installed.
  - Chef is tightly coupled with Ruby while Ansible uses Python.
  - Chef uses JSON format for configuration files while Ansible uses YAML. Some might argue that YAML is easier than JSON but that's up to you.

We're going to use Vagrant

- Vagrant is one of the DevOps tools that received a lot of attention in the past few years. It works on top of a virtualization platform (like VirtualBox or VMware).
- It uses boxes to provision virtual machines. A box is an already-installed operating system, optionally with built in applications or components.
- Vagrant uses a text file (written in Ruby) called Vagrantfile that defines the characteristics of the machine (like the network interfaces).
- To use Vagrant you have to download and install VirtualBox, VMWare or Parallel virtualization platforms.
- In this class we are going to use VirtualBox. It can be downloaded from https://www.virtualbox.org/wiki/Downloads
- After a successful installation, you can install Vagrant from https://www.vagrantup.com/downloads.html

Our first machine in the lab

- Ok, time to create a test machine to be used in our lab. Vagrant works on a per-directory basis. That is, each machine should have a separate directory for it.
- Create a new directory in your home path called ansible-machines: `mkdir ansible-machines`
- Change directory to server and issue the following command: `vagrant init bento/centos-7.3`
- The above command instructs Vagrant to import a Centos 7.3 box (create a Vagrantfile).
- Let's make some modifications to the Vagrantfile. Add the following before the end keyword at the end of the file:

```
config.vm.define "server" do |server|
  client.vm.hostname = "server"
end
```

- This will ensure that the machine name will be "server" and also the hostname. Now issue `vagrant up` to boot the machine.

Installation

- Ansible installation is quiet easy. It only requires Python, development tools, and openssl-devel packages to be installed.
- If you are on a MAC, you can simply run `homebrew install ansible`
- If you are on a Windows I highly recommend installing Vagrant and running a version of Linux to host Ansible.
- If you are using Linux, the following commands need to be issued (the example system is Centos 7.3):
  - First install EPEL repository to avail Python pip: `sudo rpm -ivh http://dl.fedoraproject.org/pub/epel/7/x86_64/e/epel-release-7-9.noarch.rpm`
  - Now that the EPEL repo is available, you can install Ansible directly using `yum -y install ansible`
  - If you prefer to use pip, you can install it using `sudo yum -y install python-pip`
  - Then we need to install the prerequisites. First install the development tools: `sudo yum -y groupinstall "Development Tools"`
  - Finally the `openssl-devel` package: `sudo yum -y install openssl-devel`
  - Now you can use pip to install Ansible as follows: `sudo pip install ansible`
  - You can verify successful installation by running `ansible --version` on the command line.

Setting up an Ubuntu machine

- Let's do the same thing on Ubuntu. I am using Bento Ubuntu 16.10 for this example.
- Create a new directory called ubuntu (you can choose whatever name you like): `mdkir ubuntu`
- From inside this directory, issue the following command: `vagrant init bento/ubuntu-16.10`
- Now issue `vagrant up` command to import the Ubuntu box and boot up the machine.

Installing Ansible on Ubuntu

- For Ubuntu you can the Ansible repository and let the Ubuntu package manager (apt-get) do the rest.
- To add the repository issue the following commands:
```
sudo apt-get update
sudo apt-add-repository ppa:ansible/ansible
```
- Now you can install Ansible by issuing the following command: `sudo apt-get install ansible`
- Verify successful installation by running `ansible --version` on the command line. It should give you the installed version which is 2.3 at the time of this recording.

Ansible inventory file

- Ansible keeps information about the nodes that it manages in a file that is quiet like the /etc/hosts file, which is used to resolve hostnames to IP addresses.
- This file lives in `/etc/ansible/hosts`. A default file gets created automatically when you install Ansible. Open the file and read the comments (comments start with #). You will find that you can add nodes in one of the following ways:
  - Ungrouped: Those are nodes that do not belong to a group. A group is a logical classification of hosts, often by their roles. You can add them one per line either as hostnames or IP addresses.
  - Grouped: You add nodes the same way as in the previous step. However, each group of nodes is preceded by a line containing the group name between square brackets. For example: [loadbalancers] or [appservers]
  - Ansible offers some shortcuts for adding multiple hosts that follow a certain pattern. For example: app01, app02, app03…app09 can be added as app0[1..9].

Setting up the environment

- Let's create a small lab to give Ansible a test drive. For Ansible to work, there should be one or more hosts that you want Ansible to configure.
- First shutdown the client machine by issuing `vagrant halt client`
- Vagrant allows you to create a number of machines simultaneously easily. Just change the part added to Vagrantfile to look as follows:

```
config.vm.define "client" do |client|
   client.vm.hostname = "client"
   client.vm.network "private_network", ip: "192.168.33.10"
 end
 config.vm.define "server" do |server|
   server.vm.hostname = "server"
   server.vm.network "private_network", ip: "192.168.33.20"
 end
```

- The above configuration will create a new machine in the same directory, called server. It will also set a private network (visible to the machines and the host only) with the respective IP addresses.
- Once finished, issue vagrant up to load both machines: client and server

Our first Ansible command

- Ansible uses password-less SSH connectivity. This can be done using public private key authentication method.
- Ok now everything is ready to start using Ansible. Login to the client machine by issuing `vagrant ssh client`
- Now edit the `/etc/ansible/hosts` and add the following:
  ```
  [servers]
  192.168.33.20
  ```
- This adds a new group called servers and inside it one node with the mentioned IP address.
- Issue the following command: `ansible servers -a "cat /etc/hosts"`
- The resulting output should be the contents of the hosts file on the remote node. Congratulations! you have just issued your first Ansible command.