What is Ansible?

- Ansible is capable of handling many powerful automation tasks with the flexibility to adapt to many environments and workflows. With Ansible user can quickly getup and running to do real work
- Ansible does configuration management
- Ansible can also be used for deployment purpose even

Why Ansible

- Open Source
- .Provisioning
- Configuration Management.
- Application Deployment
- Simple & Compliance.
- Server Orchestration.
- Have almost lot of modules
- Support from RedHat
- Agentless
- Cross Platform
- Dynamic Inventory

YAML

- YAML is a human friendly data serialization standard for all programming languages.
- Data Serialization language: A language used to convert or represent structured data or objects as a series of characters that can be stored on a disk.
- Well Known example are
    - CSV-Comma Separated Values
    - XML - Extensible Markup Language
    - JSON -Java Script Object Notation
    - YAML-Yet Another Markup Language
- Unlike XML which is not easily readable by humans, YAML was created to be human-friendly and integrate easily with modern programming languages.
- Unlike with XML, YAML was intended to simplify the viewing and understanding of config files, log files, object persistence and messaging, it allow the programmer more time programming and less worrying about formatting data.

Basic YAML Syntax Rules

- Document begins with -- -and end with …
- Indentation of lines denotes the structure within the document
- Comments begin with #
- Members of lists begin with -
- Key value pairs use the following syntax.
    - <key>:<value>

Basic YAML Example

```
---

Student-Id: 11223344

First-Name: John

Last-Name: Smith

Phone-numbers:

 - 281.555.7689

 - 713.555.8967

 - 832.555.9980

Addresses:

 - street: 123 Main St.

   city: Houston

   state: TX

…
```

Ansible Installation

1. Create a new file called 'ansible.repo' in

cd /etc/yum.repos.d

vi ansible.repo

2. Copy the following content into it:

[Ansible]

name = ansible

baseurl = https://releases.ansible.com/ansible/rpm/release/epel-7-x86_64/

enabled = 1

gpgcheck = 0

3. Run 'yum update'

rpm -Uvh https://dl.fedoraproject.org/pub/epel/epel-release-latest-7.noarch.rpm

yum -y install ansible

ansible --version

Ansible Terminology

- **Inventory:** A list of hosts, groups and variables
- **Module:** Actually do the work
- **Plugins:** Call back, Actions and other hooks
- **Facts:** Data gathered from target hosts
- **Playbooks**: A collection of plays
- **Plays:** Loops over a list of tasks mapped to a list of hosts
- **Tasks:** Invokes a module to do the work

Ansible Configuration File

Changes can be made and used in a configuration file which will be processed in the following order:

- ANSIBLE_CONFIG (an environment variable)
- ansible.cfg (in the current directory)
- .ansible.cfg (in the home directory)

- /etc/ansible/ansible.cfg

Ansible Host Inventory Parameter

- ansible
- ansible_user=admin
- ansible ssh private_key_file=/opt/ec2.pem
- ansible_python_interpreter=/usr/local/bin/python

Ansible PlayBook

- hosts: dev servers

user: ubuntu

sudo: true

tasks:

- name: Install Web Server

yum: pkg=httpd state=installed

...

Ansible Playbook With Roles

```
---
- hosts: web  #group name from the inventory
  user: ec2-user Server Auth
  sudo: true

  roles:
    - webserver #Role which should be installed on the server
...
```

Ansible Commonly Used Facts

- "ansible_distribution"
- "ansible distribution_version"
- "ansible-distribution-major-version"
- "ansible_os family"
- "ansible_kernel"

- "ansible_domain"
- "ansible_architecture"

Ansible Variables

```
---
- hosts: web  #group name from the inventory
  user: ec2-user Server Auth
  sudo: true

  vars:
    pack_name: httpd
    serv_name: httpd

  tasks:
    - name: Install {{ pack_name }}
      yum: pkg={{ pack_name }} state=installed
    - name: Start {{ serv_name }}
      service: name={{ serv_name }} state=started
...
```

Ansible Variable Files

```
---
- hosts: web  #group name from the inventory
  user: ec2-user Server Auth
  sudo: true

  vars:
    pack_name: httpd

  vars_files:
    - web_vars.yml

  tasks:
    - name: Install {{ pack_name }}
      yum: pkg={{ pack_name }} state=installed
    - name: Start {{ serv_name }}
      service: name={{ serv_name }} state=started
...
```

Ansible Variables from Prompt

```
---
- hosts: web  #group name from the inventory
  user: ec2-user Server Auth
  sudo: true
```

```
  vars:
    pack_name: httpd

  vars_prompt:
    - name: web_pass
      prompt: Web Server Password:

  tasks:
    - name: Install {{ pack_name }}
      yum: pkg={{ pack_name }} state=installed
    - name: Start {{ serv_name }}
      service: name={{ serv_name }} state=started
...
```

Ansible Install Package PlayBook

```
---
- hosts: web  #group name from the inventory
  user: ec2-user Server Auth
  sudo: true


  vars_prompt:
    - name: pack_name
      prompt: Enter the Package Name

  tasks:
    - name: Install {{ pack_name }}
      yum: pkg={{ pack_name }} state=installed

...
```

With more parameters

```
---
- hosts: web  #group name from the inventory
  user: ec2-user Server Auth
  sudo: true
```

```
  vars_prompt:
    - name: pack_name
      prompt: Enter the Package Name
      default: telnet
      private: no

  tasks:
    - name: Install {{ pack_name }}
      yum: pkg={{ pack_name }} state=installed

...
```

Ansible Task Handlers

```
---
- hosts: web  #group name from the inventory
  user: ec2-user Server Auth
  sudo: true

  vars:
    pack_name: httpd

  tasks:
    - name: Install {{ pack_name }}
      yum: pkg={{ pack_name }} state=installed
      notify: Restart HTTPD

  handlers:
    - name: Restart HTTPD
      action: service name=httpd state=restarted
...
```

Ansible Playbook Loops

```
---
- hosts: web  #group name from the inventory
  user: ec2-user Server Auth
  sudo: true
```

```
  tasks:
    - name: Add a list of users
      user: name={{ item }} state=present
      with_items:
        - user1
        - user2
        - user3
...
```

Ansible PlayBook Conditions

```
---
- hosts: web  #group name from the inventory
  user: ec2-user Server Auth
  sudo: true


  tasks:
    - name: Install httpd redhat
      yum: name=httpd state=installed
      when: ansible_distribution == "RedHat"
    - name: Install httpd ubuntu
      apt: name=apache2 state=installed
      when: ansible_distribution == "Ubuntu"

    - name: Start httpd
      service: name=httpd state=started
      when: ansible_distribution == "RedHat"
    - name: Start httpd
      service: name=apache2 state=started
      when: ansible_distribution == "Ubuntu"

...
```

Ansible PlayBook until

```
---
- hosts: web  #group name from the inventory
  user: ec2-user Server Auth
  sudo: true
```

```
  vars:
    pack_name: httpd

  tasks:
    - name: Install {{ pack_name }}
      yum: pkg={{ pack_name }} state=latest
    - name: Verify Service Status
      shell: systemctl status httpd
      register: result
      until: result.stdout.find("active (running)") !=-1
      retries: 5
      delay: 5

...
```