Ansible – Docker Overview

version 1.2

# Version History

|  |  |  |  |
| --- | --- | --- | --- |
| **Version #** | **Date** | **Revised By** | **Changes** |
| 1.0 | 18/05/2018 | Chetanya Goyal | Initial Draft |
| 1.1 | 21/05/2018 | Chetanya Goyal | Added table of Contents |
| 1.2 | 24/08/2018 | Chetanya Goyal | Added link to infygit project |
|  |  |  |  |
|  |  |  |  |

Table of Contents

[Prerequisite: 2](#_Toc514672438)

[Learning Outcome: 2](#_Toc514672439)

[What is Ansible? 3](#_Toc514672440)

[How Ansible Works? 3](#_Toc514672441)

[Playbook: 4](#_Toc514672442)

[Inventory: 4](#_Toc514672443)

[Understanding YAML: 4](#_Toc514672444)

[Representing List 5](#_Toc514672445)

[List inside Dictionaries 6](#_Toc514672446)

[List of Dictionaries 6](#_Toc514672447)

[Ansible Installation 8](#_Toc514672448)

[Ansible Configuration 9](#_Toc514672449)

[Ansible with Docker 9](#_Toc514672450)

[Requirements Installation 9](#_Toc514672451)

[Configuring Docker with Ansible 10](#_Toc514672452)

# Prerequisite

Understanding of Docker – Refer to ‘DockerFinal.docx’

# Learning Outcomes

By the end of the document you’ll be able to:

1. Understand what is Ansible and how it works
2. How we can configure Docker with Ansible

# What is Ansible?

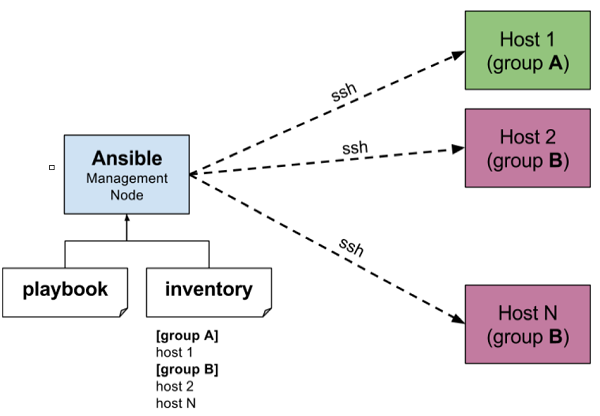
Ansible is an open source automation platform. It is very simple to setup and yet powerful. Ansible can help you with configuration management, application deployment, task automation. It can also do IT orchestration, where you need to run tasks in sequence and create a chain of events which must happen on several different servers or devices

Unlike Puppet or Chef, it doesn’t use an agent on the remote host. Instead Ansible uses SSH which is assumed to be installed on all the systems you want to manage.

Ansible is available for free and runs on Linux, Mac or BSD. Aside from the free offering, Ansible also has an enterprise product called Ansible Tower.

# How Ansible Works?

Ansible works by connecting to your nodes and pushing out small programs, called "Ansible modules" to them. Ansible then executes these modules (over SSH by default, can also use other methods for connection like Kerberos), and removes them when finished. Your library of modules can reside on any machine, and there are no servers, daemons, or databases required.



## Playbook:

Ansible uses playbook to describe automation jobs, and playbook uses very simple language i.e. YAML (It’s a human-readable data serialization language & is commonly used for configuration files, but could be used in many applications where data is being stored) which is very easy for humans to understand, read and write.

## Inventory:

Ansible works against multiple systems in your infrastructure at the same time. It does this by selecting portions of systems listed in Ansible’s inventory, which defaults to being saved in the location /etc/ansible/hosts. You can specify a different inventory file using the -i <path> option on the command line.

# Understanding YAML

Ansible uses YAML because it is very easy for humans to understand, read and write when compared to other data formats like XML and JSON.  
Every YAML file optionally starts with “---” and ends with “...”   
YAML uses simple key-value pair to represent the data. The dictionary is represented in key: value pair.

Note − There should be space between ‘:’ and value.

**Example- A student record:**

--- #Optional YAML start syntax

james:

name: james john

rollNo: 34

div: B

sex: male

… #Optional YAML end syntax

You can also use abbreviation to represent dictionaries.

**Example:**

James: {name: james john, rollNo: 34, div: B, sex: male}

## Representing List:

We can also represent List in YAML. Every element(member) of list should be written in a new line with same indentation starting with “- “ (- and space).

**Example:**

---

countries:

- America

- China

- Canada

- Iceland

…

You can also use abbreviation to represent lists.

**Example:**

Countries: [‘America’, ‘China’, ‘Canada’, ‘Iceland’]

## List inside Dictionaries:

We can use list inside dictionaries, i.e., value of key is list.

**Example:**

---

james:

name: james john

rollNo: 34

div: B

sex: male

likes:

- maths

- physics

- english

…

## List of Dictionaries:

We can also make list of dictionaries.

**Example:**

---

- james:

name: james john

rollNo: 34

div: B

sex: male

likes:

- maths

- physics

- english

- robert:

name: robert richardson

rollNo: 53

div: B

sex: male

likes:

- biology

- chemistry

…

YAML uses “|” to include newlines while showing multiple lines and “>” to suppress newlines while showing multiple lines. Due to this we can read and edit large lines. In both the cases indentation will be ignored.  
We can also represent Boolean (True/false) values in YAML, where boolean values can be case insensitive.

**Example:**

---

- james:

name: james john

rollNo: 34

div: B

sex: male

likes:

- maths

- physics

- english

result:

maths: 87

chemistry: 45

biology: 56

physics: 70

english: 80

passed: TRUE

messageIncludeNewLines: |

Congratulation!!

You passed with 79%

messageExcludeNewLines: >

Congratulation!!

You passed with 79%

…

# Ansible Installation

Ansible only needs to be installed on the control machine, or the machine from which you will be running commands. Make sure that you have Python 2 (versions 2.6 or 2.7) or Python 3 (versions 3.5 and higher) available on the control machine. Note that Windows is not supported as the control machine.

Install latest Ansible in the master node by downloading latest rpm package from <https://releases.ansible.com/ansible/>

# Ansible with Docker

Ansible offers the following modules for orchestrating Docker containers:

1. **docker\_service**

Use your existing Docker compose files to orchestrate containers on a single Docker daemon or on Swarm. Supports compose versions 1 and 2.

1. **docker\_container**

Manages the container lifecycle by providing the ability to create, update, stop, start and destroy a container.

1. **docker\_image**

Provides full control over images, including: build, pull, push, tag and remove.

1. **docker\_image\_facts**

Inspects one or more images in the Docker host’s image cache, providing the information as facts for making decision or assertions in a playbook.

1. **docker\_login**

Authenticates with Docker Hub or any Docker registry and updates the Docker Engine config file, which in turn provides password-free pushing and pulling of images to and from the registry.

1. **docker (dynamic inventory)**

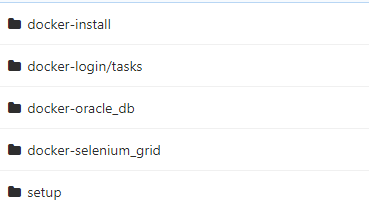
Dynamically builds an inventory of all the available containers from a set of one or more Docker hosts.

# Using Ansible and Docker in Infosys RHEL 7.5 Akaash VMs

Visit <http://infygit/solutions/Ansible-Docker/tree/master> . This is an Ansible projects that can be used to install, configure and use Docker within Infosys Akaash RHEL 7.5 VMs.

## Project structure

The project has following roles in the roles folder:



1. **setup:** Installs git and maven in the remote nodes.
2. **docker-install:** Installs docker-ce 18.03.1, pip, virtualenv and then installs docker-compose python module through pip in a virtualenv
3. **docker-login:** Performs logging in in infy nexus docker repository
4. **docker-oracle\_db:** Creates container of OracleDatabase docker project. The handlers inside this role performs loading of the database dump only after a line ‘Database ready to use. Enjoy!’ is present in the docker container logs.
5. **docker-selenium\_grid:** Creates container of SeleniumGrid docker project