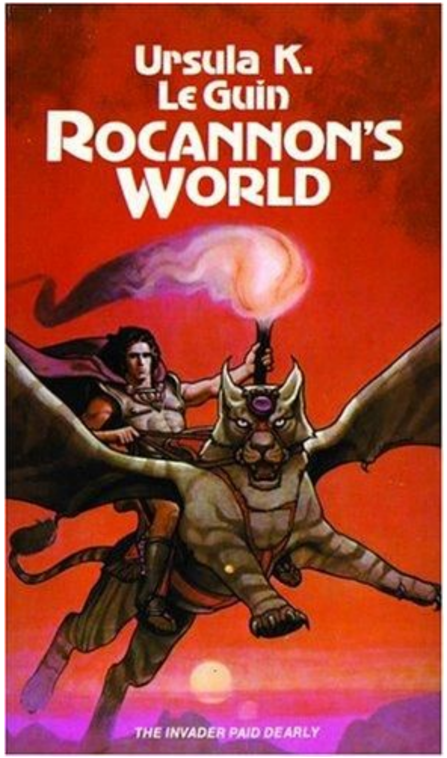
Ansible for Newbies

Greg Horie

github.com/netserf

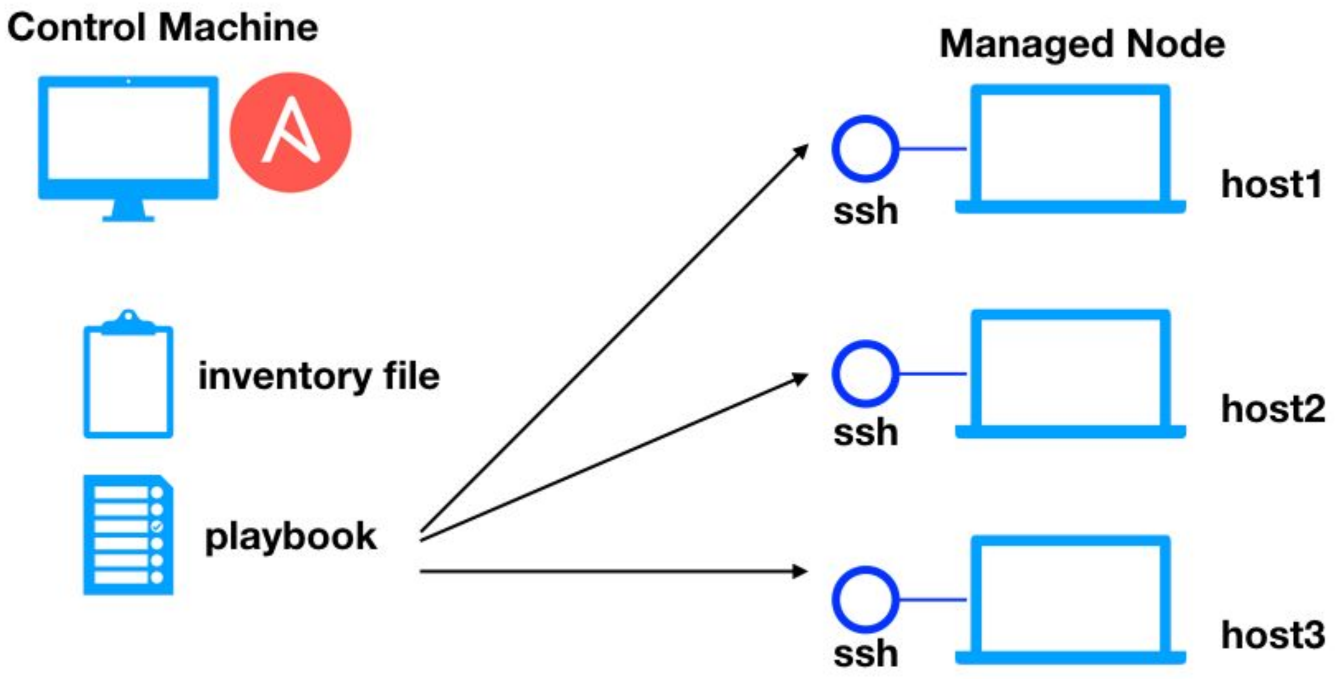
What is Ansible?



What is Ansible?

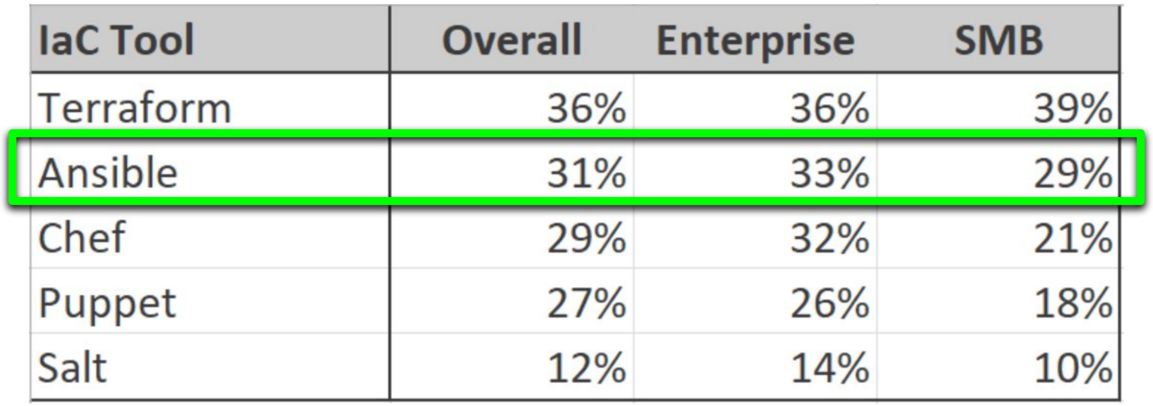
- A procedural IaC tool used to automate IT tasks.
- Developed in 2012 by Michael DeHaan.
    - Open source project using the GPL-3.0 license.
    - Hosted on GitHub.com - https://github.com/ansible/ansible
    - Ansible (the company) was acquired by RedHat in 2015.
    - RedHat acquired by IBM in 2019, so Ansible is curated by IBM.
- How does it work?
    - Integrates with nodes through SSH or APIs.
    - With SSH, it pushes ansible modules to clients then executes them from remote.
- Ansible control machine runs on Linux / Unix hosts.
    - Manages Linux, Unix, and Windows targets.

What is Ansible?



What is IaC?

- IaC = Infrastructure-as-Code
- High-level coding languages used to automate IT infrastructure builds.
- **IaC Tool Adoption**



IaC Tool	Overall	Enterprise	SMB
Terraform	36%	36%	39%
Ansible	31%	33%	29%
Chef	29%	32%	21%
Puppet	27%	26%	18%
Salt	12%	14%	10%

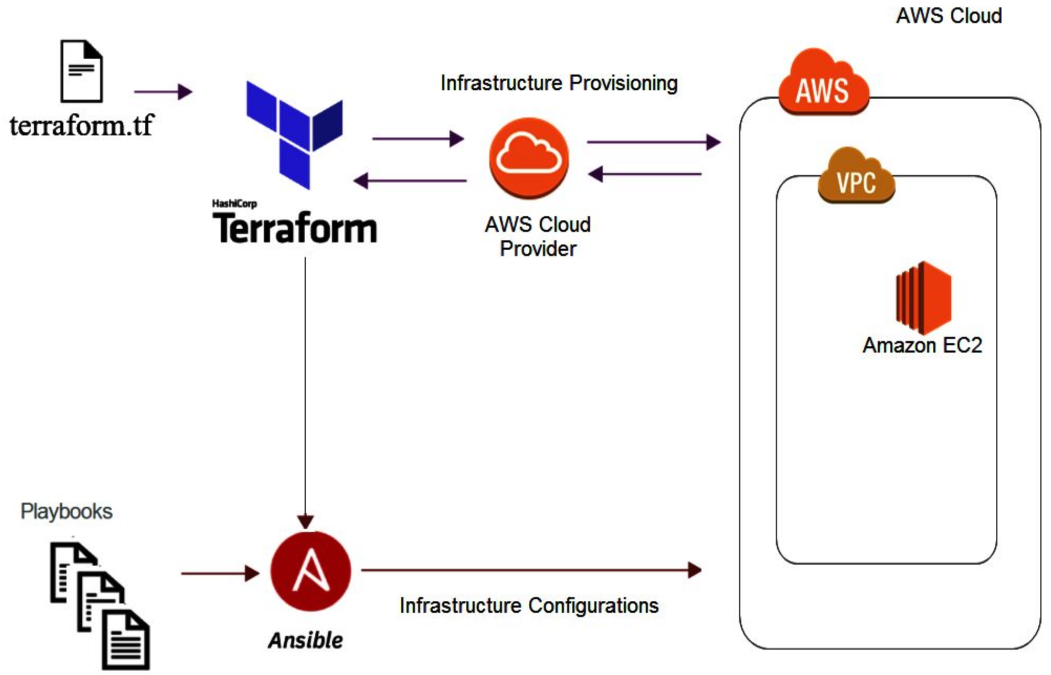
* Taken from Flexera 2021 State of Cloud report.

- Not an apples-for-apples comparison.
- In practice, these tools are often used together to fully-automate builds.

Procedural vs. Declarative

- IaC tools can be categorized as Procedural or Declarative.
  - Based on how the tool applies changes to the underlying target systems.
- **Procedural**
  - Think script.
  - Step-by-step tasks to automate your build.
  - Ansible & Chef are procedural - Code step-by-step tasks to achieve a desired end state.
- **Declarative**
  - Code says what is desire, not the steps.
  - Hides the process and reveals relationships.
  - Terraform & Puppet are declarative - Code specifies your desired end state.

IaC Tools Working Together



Ansible Pros

- Simple syntax which is easy to learn.
- YAML-based and readable.
- Agentless.
- Makes full deployment automation possible.
- Comprehensive module support.
- Ansible Galaxy - Repository for shared Ansible content.
- Well-suited for system and network engineers.

Ansible Cons

- Not ideal for complex playbooks with extensive logics.
- No state maintenance.
    - Ansible doesn’t track dependencies.
- Does not scale as well for a large fleet of hosts.

ANSIBLE

Ansible Installation

- Recommend: python 3.6+
- Pip installation:

```
$ sudo pip install ansible
```

- Ubuntu installation:

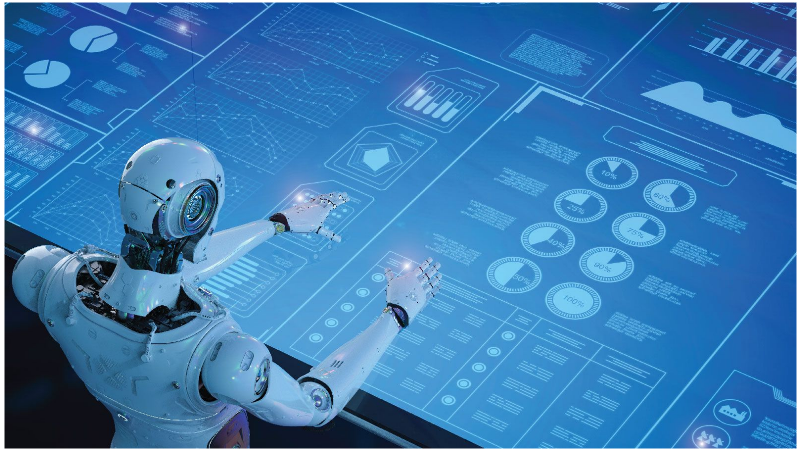
```
$ sudo apt-add-repository ppa:ansible/ansible
$ sudo apt update
$ sudo apt install ansible
```

- CentOS installation:

```
$ sudo yum install epel-release
$ sudo yum install ansible
```

Ansible - Concepts

- Inventory File
- Modules



Ansible - Inventory File

- Ansible works with a list of managed hosts known as the inventory.
- You can execute your playbooks across all or a subset these hosts.
  - You can also group your hosts for easy subset selection.
  - Pattern matching can be used to select subsets of hosts.
- Inventory files can be defined in INI, YAML, or JSON.
- Set your desired default inventory file in **`ansible.cfg`**.

```
[defaults]
inventory = inventory/hosts.ini
```

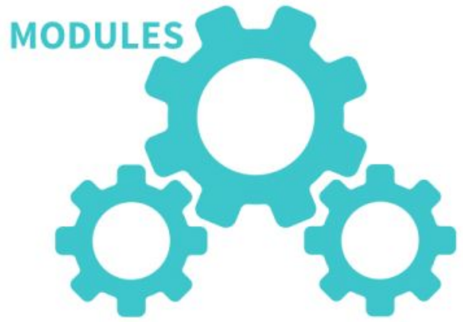
Ansible - Inventory File Example

hosts.ini

```
localhost        ansible_connection=local

[pis]
rpi


[k8s]
kmaster
knode1
knode2
```

Ansible - Modules

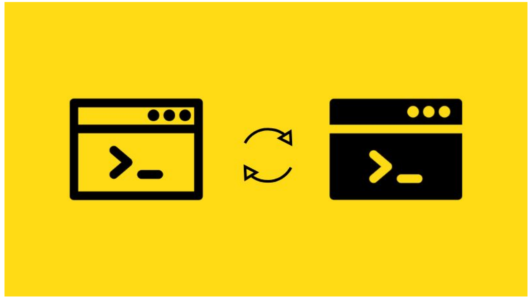
- Ansible ships with a number of task modules called the "module library".
    - Similar to python standard library.
    - Batteries-included philosophy.
- These task modules can be used on the command line or in playbook tasks.
- You can create your own modules or find shared ones on Ansible Galaxy.



Ansible - Ad-hoc Command Demo

```
$ ansible all -m ping
```

```
$ ansible k8s -m command -a "hostname"
```



Note - Inventory has been pre-defined.

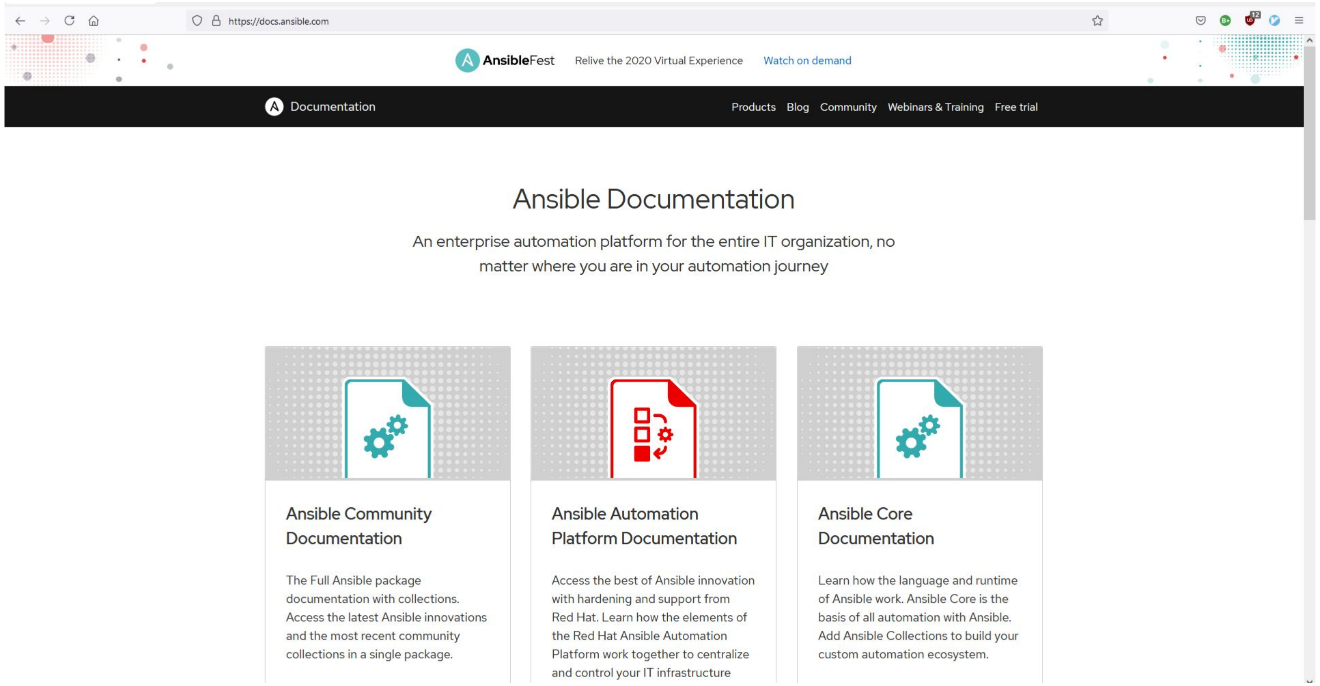
```
$ ansible-inventory --list
```

Ansible - Common Modules

Module Name	Description
copy	Copy a file from the control machine to the target host(s).
template	Similar to the copy module. Processes template at run-time using Jinja2 before the copy.
user	Create / manage Linux users in your target host(s).
package	Install, update, or remove software packages from your target host(s).
service	Manage the target host services using its init system (e.g. systemd).
file	Manage file states - i.e. permissions, ownership, and selinux labels.
command / shell	Execute arbitrary commands on the target system. The shell module also allowing you to use shell operators i.e. >, <, |, &&, ||, etc.

Ansible Docs

- https://docs.ansible.com/



https://docs.ansible.com

AnsibleFest Relive the 2020 Virtual Experience Watch on demand

Documentation Products Blog Community Webinars & Training Free trial

Ansible Documentation

An enterprise automation platform for the entire IT organization, no matter where you are in your automation journey

Ansible Community Documentation

The Full Ansible package documentation with collections. Access the latest Ansible innovations and the most recent community collections in a single package.

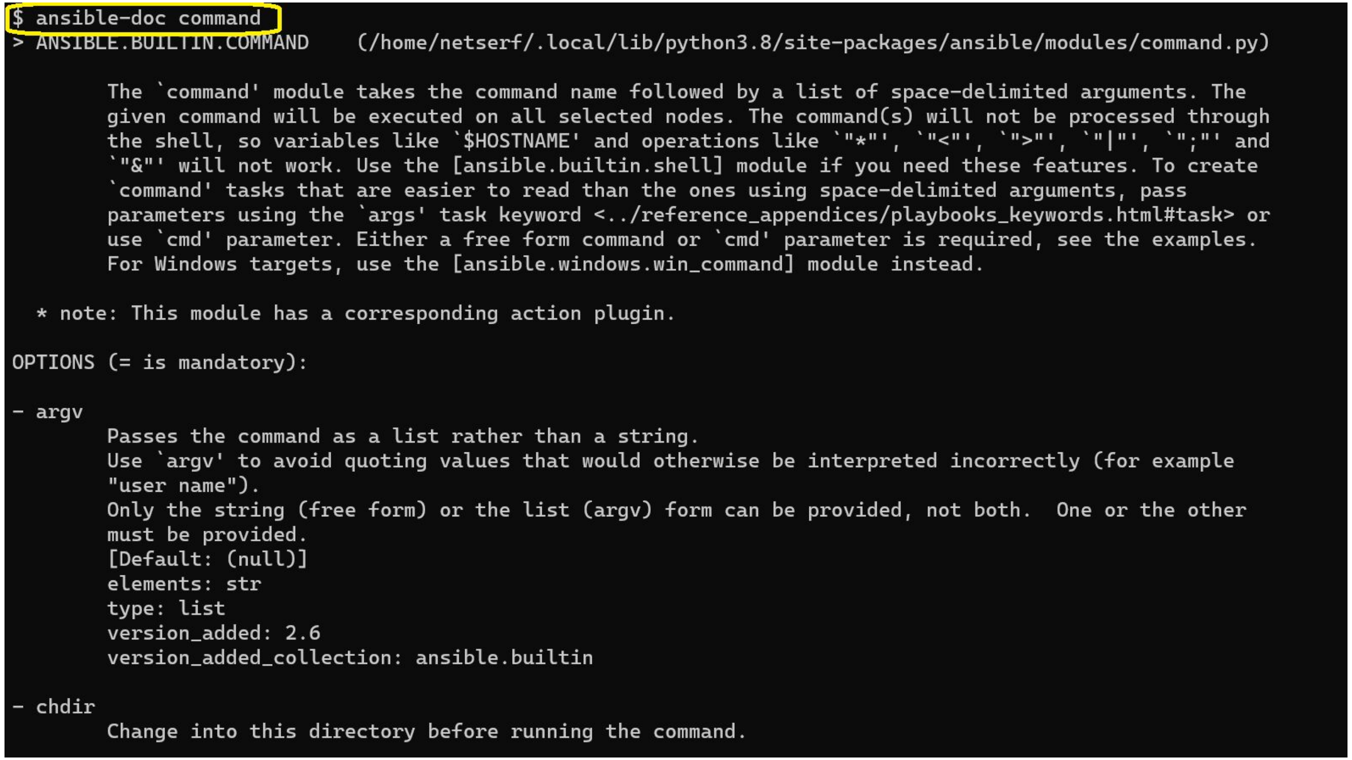
Ansible Automation Platform Documentation

Access the best of Ansible innovation with hardening and support from Red Hat. Learn how the elements of the Red Hat Ansible Automation Platform work together to centralize and control your IT infrastructure

Ansible Core Documentation

Learn how the language and runtime of Ansible work. Ansible Core is the basis of all automation with Ansible. Add Ansible Collections to build your custom automation ecosystem.

Ansible Docs



```
$ ansible-doc command
> ANSIBLE.BUILTIN.COMMAND    (/home/netserf/.local/lib/python3.8/site-packages/ansible/modules/command.py)

        The `command' module takes the command name followed by a list of space-delimited arguments. The
        given command will be executed on all selected nodes. The command(s) will not be processed through
        the shell, so variables like `$HOSTNAME' and operations like `"*"', `"<"', `">"', `"|"', `";"' and
        `"&"' will not work. Use the [ansible.builtin.shell] module if you need these features. To create
        `command' tasks that are easier to read than the ones using space-delimited arguments, pass
        parameters using the `args' task keyword <../reference_appendices/playbooks_keywords.html#task> or
        use `cmd' parameter. Either a free form command or `cmd' parameter is required, see the examples.
        For Windows targets, use the [ansible.windows.win_command] module instead.

  * note: This module has a corresponding action plugin.

OPTIONS (= is mandatory):

- argv
        Passes the command as a list rather than a string.
        Use `argv' to avoid quoting values that would otherwise be interpreted incorrectly (for example
        "user name").
        Only the string (free form) or the list (argv) form can be provided, not both.  One or the other
        must be provided.
        [Default: (null)]
        elements: str
        type: list
        version_added: 2.6
        version_added_collection: ansible.builtin

- chdir
        Change into this directory before running the command.
```

Ansible - Basic Concepts

- Playbooks
- Tasks

Ansible - Playbooks

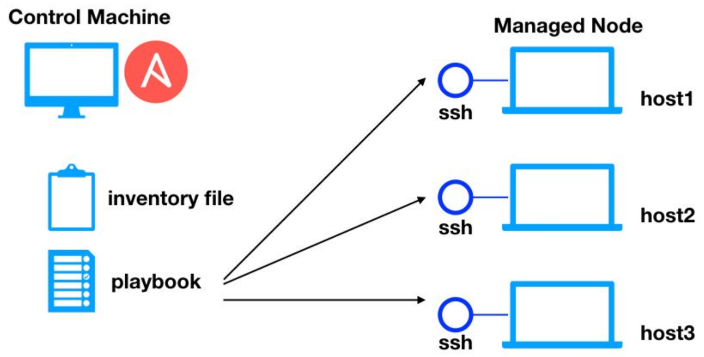
- Playbooks are essentially ansible scripts.
- They compose a set of tasks used to drive your desired automation.
- Example:

hostname.yaml

```
---
- hosts: all
  tasks:
  - name: list the hostname of our inventory
    command:
    cmd: hostname
```

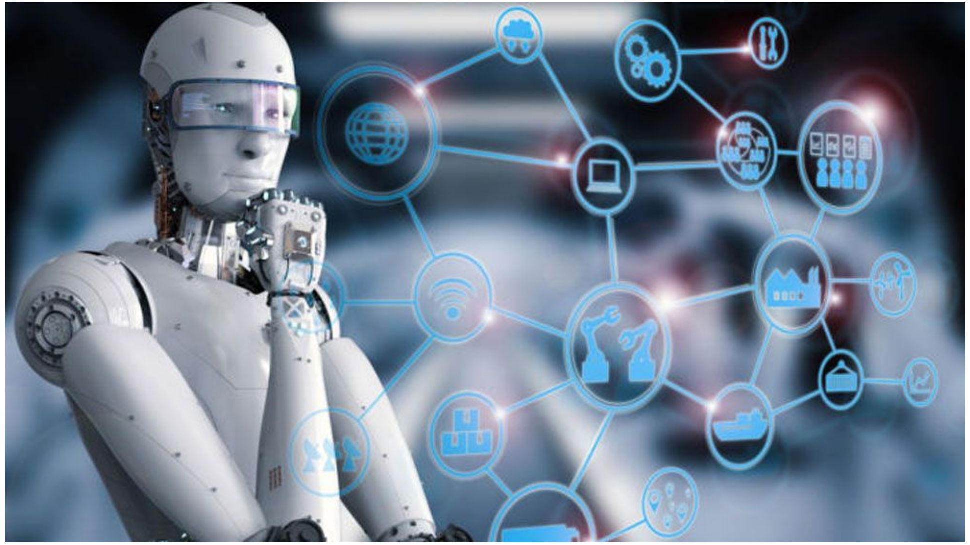
Ansible - Simple Playbook Demo

```
$ ansible-playbook hostname.yaml

$ ansible-playbook hostname.yaml -v

$ ansible-playbook hostname.yaml -v --check
```



Ansible - Basic Concepts

- Variables
- Facts
- Templates



Ansible - Variables

- Simple variables can be hard-coded directly in playbooks.
- Example:

```
 echo.yaml
---
- hosts: all
  vars:
    my_greeting: Hello World!
  tasks:
  - name: a naive use of variables
    command:
    cmd: echo {{ my_greeting }}
```

Ansible - Variable Demo

```
$ ansible-playbook -l kmaster echo.yaml -v
```

Overrides are also possible:

```
$ ansible-playbook -l kmaster echo.yaml \
  -e '{"my_greeting": "Yo World!"}' -v
```

Ansible - Variables

- Better practice is to put variables into the following files and sub-directories:

1. `host_vars/*`
2. `group_vars/*`
3. `group_vars/all.yaml`

- Specified in order of precedence.
- **Notes**
    - Extra vars provided on the command line have the highest precedence.
    - Locally defined playbook vars will also override the externally defined variables.
    - There are more levels of precedence. Check the docs.

Ansible - Variables Demo

Note:

- Use `ansible-inventory --list` to see your variables.

  ```
  $ ansible-playbook -l kmaster echo2.yaml -v
  ```

- Update `group_vars/all.yaml`
- Update `group_vars/k8s.yaml`
- Update `host_vars/kmaster.yaml`
- Notice the precedence.

Ansible - Facts

- Facts are data gathered from remote systems.
  - Includes operating system details, IP addresses, filesystems, cpu, ram, etc.
- Accessed through the ansible_facts variable in a playbook.
- You can create custom facts in the `/etc/ansible/facts.d` directory.
- Examples:

```
$ ansible kmaster -m setup

$ ansible kmaster -m setup -a "filter=ansible_default_ipv6"
```

Ansible - Templates

- Dynamically create files on the target hosts with variable substitution.
- The Jinja2 templating system is used - similar to Flask.

templates/hello_world.j2

```
My Greeting:  {{ my_greeting }}
```

template.yaml

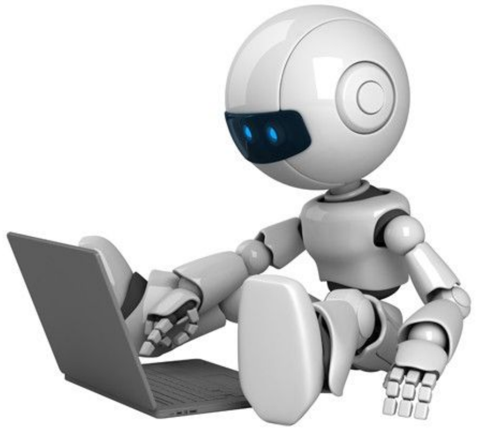
```
- hosts: all
  tasks:
     - name: template example
     template:
     src: hello_world.j2
     dest: /tmp/hello_world.txt
```

```
$ ansible-playbook -l k8s \
  template.yaml -v
```

Playbook Fun

- One more example with a more realistic deployment:

```
$ ansible-playbook nginx.yaml -v
```



Summary

- Ansible is procedural IaC tool used to automate IT tasks.
- Simple to use and easy to learn.
- Great module and community support.
- It makes deployment automation easy.
- Give it a try - It's fun!

VicPiMakers and Others Slack

- Please let us know if you want an invite to this Slack group

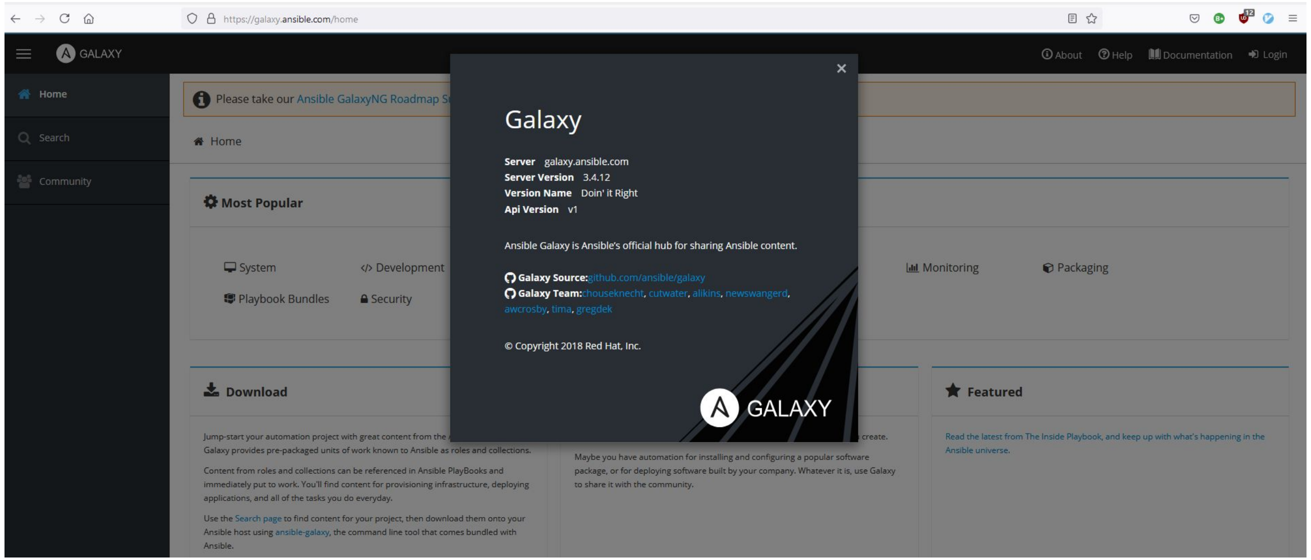
Backup Slides

Ansible - Roles

- Roles are reusable units of deployment.
- Roles abstract a series of tasks.
- They are specific, configurable, and self-contained.
  - e.g. Apache, MySQL and PHP roles may be used to deploy a LAMP stack.
- There are vendor and community supported roles found on Ansible Galaxy.
  - Or you can always build your own.

Ansible Galaxy

- https://galaxy.ansible.com/home



https://galaxy.ansible.com/home

GALAXY

About Help Documentation Login

Home
Search
Community

Please take our Ansible GalaxyNG Roadmap S

Home

Most Popular

System
Development
Monitoring
Packaging
Playbook Bundles
Security

Galaxy

Server galaxy.ansible.com
Server Version 3.4.12
Version Name Doin' it Right
Api Version v1

Ansible Galaxy is Ansible's official hub for sharing Ansible content.

Galaxy Source:github.com/ansible/galaxy
Galaxy Team:chouseknecht, cutwater, alikins, newswangerd, awcrosby, tima, gregdek

© Copyright 2018 Red Hat, Inc.

GALAXY

Download

Jump-start your automation project with great content from the
Galaxy provides pre-packaged units of work known to Ansible as roles and collections.

Content from roles and collections can be referenced in Ansible PlayBooks and immediately put to work. You'll find content for provisioning infrastructure, deploying applications, and all of the tasks you do everyday.

Use the Search page to find content for your project, then download them onto your Ansible host using ansible-galaxy, the command line tool that comes bundled with Ansible.

create.

Maybe you have automation for installing and configuring a popular software package, or for deploying software built by your company. Whatever it is, use Galaxy to share it with the community.

Featured

Read the latest from The Inside Playbook, and keep up with what's happening in the Ansible universe.

- **Warning** - Be cautious with shared code.

Ansible Galaxy Demo

```
$ ansible-galaxy install geerlingguy.apache
```

- Galaxy Docs:
  https://galaxy.ansible.com/geerlingguy/apache

- GitHub Repo:
  https://github.com/geerlingguy/ansible-role-apache

- Example Playbook:

```
- hosts: webservers
  roles:
    - { role: geerlingguy.apache }
```

Ansible Connection Options

```
$ ansible-doc -t connection -l
```