Using Ansible to deploy Nginx, Node, MongoDB

architech

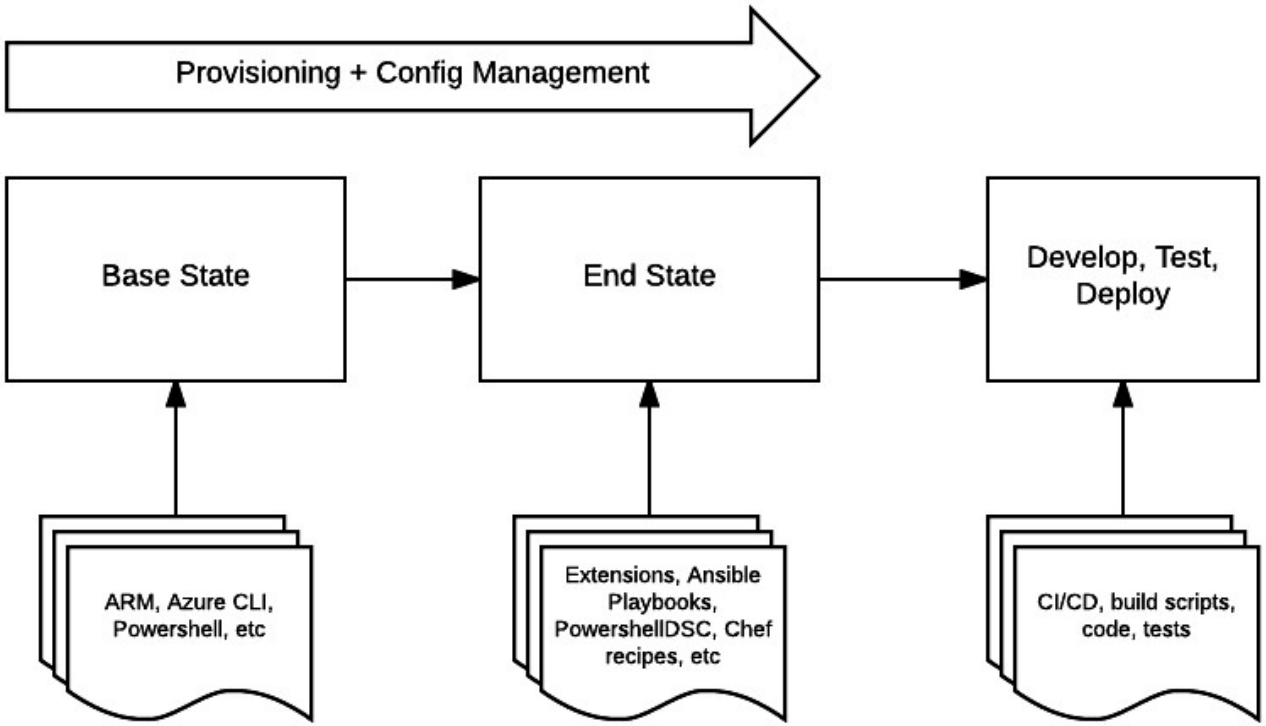
Cloud Computing Lunch & Learn Series

1. Cloud Computing for Non-Techies
2. Overview of Azure IaaS and deploying an HA, secure Linux cluster
3. Overview of Custom Script Extensions and deploying Ansible to a Linux cluster
4. **Deploying nginx, node.js, mongoDB using Ansible to a Linux cluster**
5. Deploying a containerized node.js and mongoDB application to a Linux cluster

Agenda

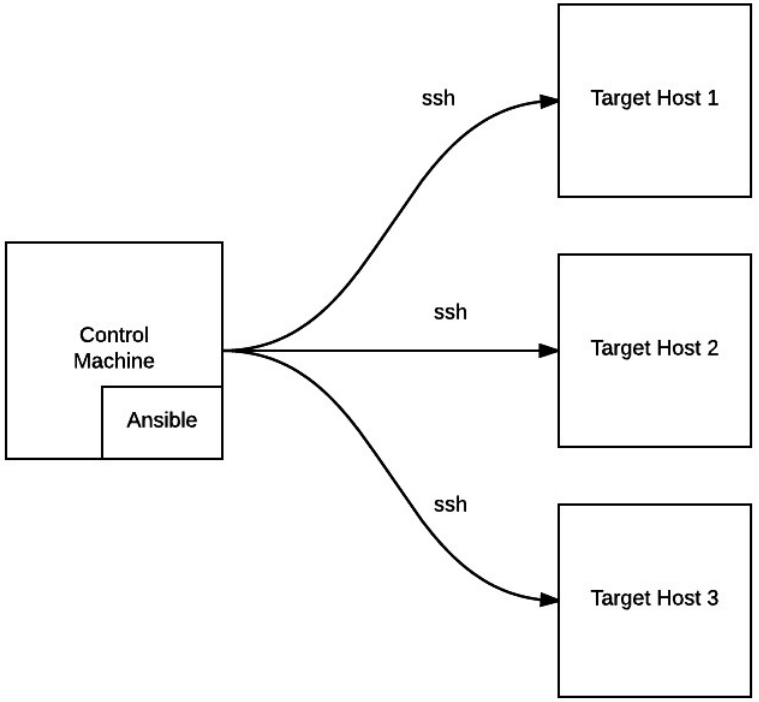
1. Infrastructure Provisioning & Configuration Management process
2. What is Ansible?
    1. Tasks, Plays, Playbooks, Roles
    2. Use-cases
3. Demo – deploying nginx, node, mongoDB to the appserver and database roles

Infrastructure Provisioning & Configuration Management

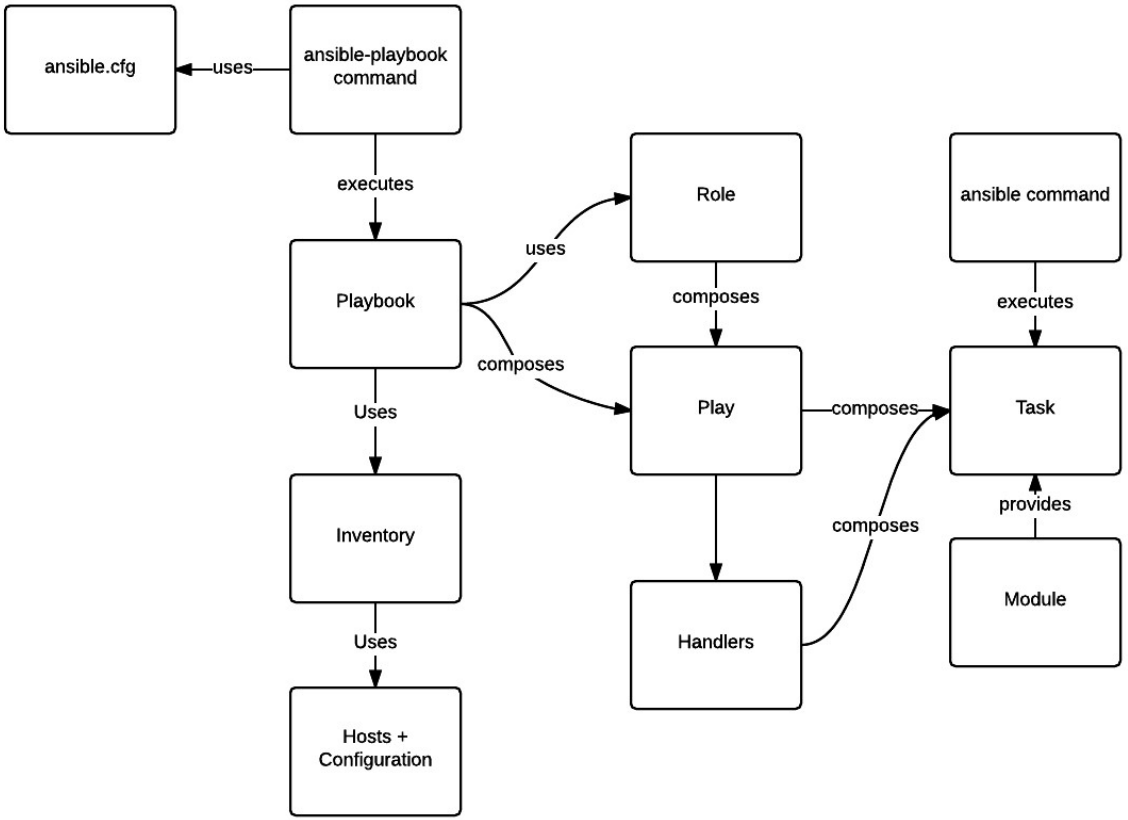


What is Ansible?

- An open-source configuration management tool
- Developed in python so widely available.
- Has an “agent-less” architecture so nothing to install on the target machines.
- Highly modular. Many modules to do common tasks. Can do simple things really easily with minimal setup but scales well to complex use-cases
- Very well documented
- Use a “push” model vs “pull” model of Chef and Puppet.



Task, Plays, Playbooks, Roles…



Common commands

- Quickly check you can reach the target hosts

  `ansible –m ping all`

- List all tasks in your playbook. This also checks the syntax of your playbook.

  `ansible-playbook –list-tasks playbook.yml [ --ask-sudo-pass ]`

- Execute your playbook

  `ansible-playbook playbook.yml`

  `(you specified the inventory location in ansible.cnf)`

  `ansible-playbook -i <inventory path> playbook.yml`

Deploying nginx, node, mongoDB to the Linux Cluster

https://bitbucket.org/architech/ansible-nginx-node-mongodb

Tips & Tricks

- To save time, do the following:
  - Test the playbooks on the target image (obvious right?). It is not enough to have “CentOS” installed on your laptop. Create a VM on Azure and test it there.
  - Set all host values to localhost to increase performance of your testing. When that passes, you are 99% there.
  - Execute the commands that serve as the basis for the Ansible command by hand. You will uncover the settings you need to define. E.g. need to run as root, etc
  - Harden the image first, then apply the configuration and relax policies as needed. Hardening the image after configuration will cause you to spin.
  - Create a simple test to verify the configuration end-end. In my case, I wrote a simple node app to test the load-balancer, nginx proxy, node and of course the database.
  - Always run ansible-playbooks –list-tasks first. This not only allows you to review the order in which the tasks will be executed but also will check the syntax.
  - Use the debug command to print out variable values etc to ensure the proper values are being set.

References

- Ansible Docs - http://docs.ansible.com/ansible/intro.html
- Ansible Up and Running book - https://goo.gl/NQH6k1
- Mastering Ansible on Udemy - https://www.udemy.com/mastering-ansible/learn/v4/overview
- Configuration Management 101 - https://www.digitalocean.com/community/tutorials/configuration-management-101-writing-ansible-playbooks
- Ansible Galaxy - https://galaxy.ansible.com/
- Ansible Tower - https://www.ansible.com/tower