CLOUD, CONTAINERS & VIRTUALIZATION

21st August '19
Cloud Administration Essential Track
PRESENTERS / CONTRIBUTORS

DUSHYANT DUBARIA
DevOps / Deployment - Cloud Support Associate
Amazon Web Services

JAY SHAH
DevOps & Network Engineering
Southern Methodist University
AGILE / DEVOPS

Agile Development
- Iterative Development
- Scrum, Sprint, Stories
- Velocity

DevOps
- Continuous Integration
- Continuous Deployment
- IT Automation
- Application Management

Business
- Business case
- Requirement
- Use cases
- Features
- Plan
- Go to Market

Developers (application)
- Design
- Code
- Re-factor
- Unit Test
- Bug Fix
- Deploy

IT Operations
- Provision
- Configure
- Orchestrate
- Deploy
- Report
- Monitor

Business Agility
IT Agility
Dev-ops tools for collaboration

- Bamboo
- Hudson
- Subversion
- git
- JIRA
- Maven
- Gradle
- Selenium
- JUnit
- Jenkins
- Docker
- Chef
- Saltstack
- Ansible
- Nagios
- Splunk
- Vagrant
- Puppet

Integration

Plan
Code
Build
Test
Deploy
Operate
Monitor
DevOps Workflow

Local Test → Commit → myRepo → Push → Project Repo → Continuous Integration/Continuous Deployment pipeline

- Build
- Test
- Review/Approve
- Deliver
- Deploy

Monitor and Iterate
LET'S GET OUR HANDS DIRTY

- git
- GitHub
- Vagrant
- Ansible
- kubernetes
- Docker
- Google Cloud
- Azure
- AWS
- Linux Foundation
INFRASTRUCTURE

Web Server: Apache 2
Database Server: Maria DB
Jenkins
Testing Server
Java Installed
Replica of Web Server
Web Development

Virtual Box: Bento/CentOS 7.3
Client: Python Plugins, Ansible

GitHub: To keep a common repository for all developers

Operating System
- Git Bash
- Vagrant
- Visual Basic

Software’s Installed: GitBash, Vagrant, Visual Basic
Infrastructure Provisioning Tool

Vagrant: Automated Virtual Machine Set-Up

Puppet Code

Full Stack

Load Back

Check for Stability

Settings

RAM

OS
Vagrant Workflow

$ vagrant up
Vagrantfile

1. Developer
2. Provisioners
3. VirtualBox
4. Vagrant
5. Virtual Machine
6. SSH
Vagrant

- Automates Infrastructure provisioning
- Uses various providers such as VMware, VirtualBox, Hyper-V, AWS
- Supports variety of Provisioners such as Ansible, Chef, etc.
- Vagrant uses CLI and VagrantFile to operate
- Uses Vagrantbox – which is a packaged environment like a template to provide OS image
- Vagrant is configured and individual directory is created
VagrantFile

# -*- mode: ruby -*-
# vi: set ft=ruby :

# All Vagrant configuration is done below. The "2" in Vagrant.configure
# configures the configuration version (we support older styles for
# backwards compatibility). Please don't change it unless you know what
# you're doing.
Vagrant.configure("2") do |config|
  # The most common configuration options are documented and commented below.
  # For a complete reference, please see the online documentation at
  # https://docs.vagrantup.com.

  # Every Vagrant development environment requires a box. You can search for
  # boxes at https://vagrantcloud.com/search.
  config.vm.box = "bento/centos-7.3"

  # Disable automatic box update checking. If you disable this, then
  # boxes will only be checked for updates when the user runs
  # `vagrant box outdated`. This is not recommended.
  # config.vm.box_check_update = false

  # Create a forwarded port mapping which allows access to a specific port
  # within the machine from a port on the host machine. In the example below,
  # accessing "localhost:8080" will access port 80 on the guest machine.
  # NOTE: This will enable public access to the opened port
  # config.vm.network "forwarded_port", guest: 80, host: 8080
Ansible Terminologies

Ansible Node

- Hosts Inventory File
- Playbook

Web Server

- ssh
Need for Configuration Management

• For those who require Vagrant as an infrastructure-provisioning tool, there should be a way to deploy or customize an application.

• So, you can find a Vagrant box with Apache installed. But what if you want a Nginx reverse proxy installed on top of that? Or, what if you need some custom users and groups configured, together with some file permissions?

• Shell scripting is an option and it is supported by Vagrant to boot up and configure the machine automatically (instant provisioning). But this may not be your best choice.
Ansible – High Level Design
So, What is Ansible?

- Open Source Configuration Management tool
- An Infrastructure Automation / Configuration Tool
- It uses playbooks to deploy, manage and configure multiple servers.
- Can integrated with other DevOps tools like Jenkins and Docker
- Agent-less
- Written in Python and uses Playbook written in YAML
- YAML- Yet Another Markup Language - It uses JSON files to serialize data in easy format
- Depends on SSH to connect to the managed hosts
- Uses /etc/ansible/hosts file to describe machines and contains IP addresses, hostnames
Installation of Ansible

- We will cover installation on Ubuntu and Centos as they are well known Linux distros. Once installed, Ansible commands are the same on any platform.

- On an **Ubuntu** system issue the following commands to install the package:
  - `sudo apt-add-repository ppa:ansible/ansible`
  - `sudo apt-get update`
  - `sudo apt-get install ansible`

- On a **Centos** (or any Red Hat variant), issue the following commands:
  - `sudo yum install epel-release`
  - `sudo yum install ansible`

- If you want to install it using pip (on either system), run the following commands (make sure Python is installed)
  - `sudo easy_install pip`
  - `pip install ansible`
YAML is short for YAML Ain’t Markup Language. It is used the same as JSON files: to serialize data in an easy format that is readable by both humans and machines.

Data is stored in key/value pairs separated by colons. Items can be grouped if they have the same indentation level.

Arrays (called dictionaries) of data are represented by the placing a dash before each item. Of course all items in an array should be on the same indentation level.

YAML does not allow you to use tabs for indentation. Only spaces are allowed.
Our first playbook will be used to install the Apache web server on the target machine(s).

We strongly recommend that you use a code editor (Notepad++ for Windows, Sublime Text and Textmate for macOS are excellent examples) for editing playbooks. They will ensure that you follow the correct indentation.

Open a new file called webserver.yml and add the following text:

```yaml
- hosts: ubuntu
  become: yes
  tasks:
    - name: Ensure that the Apache web server is installed
      apt:
        name: apache2
        state: present
    - name: Ensure that the service is running and will be started on boot
      service:
        name: apache2
        state: started
        enabled: yes
```
Inventory file for Ansible

- Ansible uses /etc/ansible/hosts file to know about which machines you want configured by Ansible.

- This file contains machine host names, IP addresses, or a mix of them. This depends on how the client machine (the one with Ansible installed) can communicate with target hosts.

- Hosts can be added either in named groups for easier reference, or as standalone entries.

- If you have a number of hosts for which you chose a numbered naming pattern (like web1, web2, web3...web6), you can add them on one line like web[1..6].
Need for Version Control System

- The ability to work on the same project by more than one person at the same time
- Protection against errors and failures by enabling developers to go to and from different versions of the same code
- Code can be branched
- Contains full project history
- Ability to access over the internet. Git can be integrated with GitHub so repository can be accessed from anywhere
Git is the program used to version control your files.

GitHub, on the other hand is a web application. It offers a version control repository.

Git can be integrated with GitHub so that the repository can be accessed over the Internet from anywhere in the world instead of having to create a local repository server.

Other version control systems are (BitBucket, Azure Repo, CodeCommit, Code Build etc.)
You can download the package from https://git-scm.com/download/

- CentOS: yum –y install git
- Ubuntu: apt-get –y install git
Git Workflow

- **Local**
  - Working directory
  - Staging area
  - Local repo
  - Git Operations:
    - Git add
    - Git commit
    - Git checkout
    - Git merge

- **Remote**
  - Remote repo
  - Git Operations:
    - Git push
    - Git fetch
JENKINS
Jenkins Walk-through:

- It’s a both CI and CD tool
- Supports various Configuration management tool
- It uses Playbook or Cookbooks to facilitate automated delivery of infrastructure configuration
- Supports many plugins
- Allows to create and build jobs and deployments
- It does performance and unit testing
- Gives robust environment for building, integrating and deployment process
**Continuous Integration**: Pull together various software components - build and test as single unit

**Continuous Delivery**: More automation, delivers app code from master to non-prod

**Continuous Deployment**: Full automation, code is deployed to binary artifacts and to live prod servers for users
How It Works?

THE PROJECT BRANCHES

Feature branch

Integration branch

Master branch

PHPUnit

Development

Production
The project is a web-based booking application. It contains the following components:

<table>
<thead>
<tr>
<th>Environment</th>
<th>Number of machines</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>Client</td>
<td>1</td>
<td>Mimics the developer machine used to write the application code. Also used for running Ansible against the other environments.</td>
</tr>
<tr>
<td>Testing</td>
<td>2</td>
<td>The application server hosting the web application and the database server hosting the backend database</td>
</tr>
<tr>
<td>Production</td>
<td>2</td>
<td>The application server, the database server.</td>
</tr>
</tbody>
</table>
Requirements for the Infrastructure

➤ The client machine is a Centos 7 Vagrant machine with the following components:
  ➤ Ansible
  ➤ Git

➤ The application server is a Centos 7 Vagrant machine with the following components:
  ➤ Apache
  ➤ PHP 7
  ➤ Git

➤ The database server is a Centos 7 Vagrant machine with the following:
  ➤ MySQL
  ➤ The CI/CD server is an Ubuntu Vagrant machine containing the following:

➤ Jenkins with the following plugins:
  ➤ SSH host
  ➤ GitHub
  ➤ PHPUnit testing tool

➤ A testing environment will contain a clone of the application and database machines to be used by Jenkins
INFRASTRUCTURE

**Operating System**
Software’s Installed: GitBash, Vagrant, Visual Basic

**Web Server**
- Apache 2

**Database Server**
- MARIA DB

**Client**
- BENTO/ CENTOS 7.3
- Python plugins, Ansible
- GitHub: To keep a common repository for all developers

**Web Development**
- Replica of Web Server
Add the following to the end of the Vagrant file (before the last end keyword):

```ruby
config.vm.define "jenkins" do |jenkins|
  jenkins.vm.box = "bento/ubuntu-16.10"
  jenkins.vm.hostname = "jenkins"
  jenkins.vm.network "private_network", ip: "192.168.33.40"
  jenkins.vm.provision "ansible" do |ansible|
    ansible.playbook = "jenkins.yml"
  end
end
```

Provisioning the Jenkins Machine
Containerization vs Virtualization

Hypervisor-based Virtualization

Container Engine

Operating System

Infrastructure

Hypervisor

Host Operating System

App 1
Bins/Libs

Guest OS

App 2
Bins/Libs

Guest OS

App 3
Bins/Libs

Guest OS

App 1
Bins/Libs

App 2
Bins/Libs

App 3
Bins/Libs

Container virtualization
DEVOPS TROUBLESHOOTING TOOLS

WireShark
Progress Telerik Fiddler
Datadog
SolarWinds
Nagios
Published Paper in ITERA on “Software Defined Networking (SDN) in Telecommunication Industry” at Lexington, Kentucky, Mar’18

Published Paper in ‘IEEE-UEMCON’, on “A Survey of DevOps tools for Networking” at Columbia University, New York, Nov’18

Published Paper in ‘IEEE-CCWC’ on “Building Modern Clouds: Using Docker, Kubernetes & Google Cloud Platform” held at University of Nevada, Jan’19
CHECK OUR LIVE PROJECTS & CONNECT US

- dushyant8858
- shahjay9315

- Dushyant8858
- shah-jay