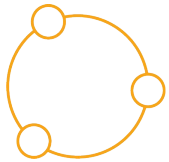


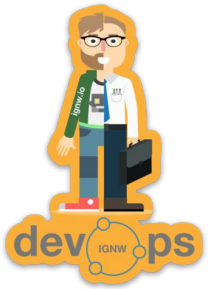
IGNW

www.ignw.io

Introduction to Ansible for Python Devs

by Joe Jacobs - DevOps Engineer

https://github.com/joej164/intro_to_ansible



About the Presenter

Career and Education

- DevOps Engineer at IGNW
- ~ 4 years experience with Python
- First 15 years of my career were as a Telephony Engineer
- Starting the Georgia Tech Online Masters of CS program in January

Personal

- Likes Video Games (PS4)
- Listening to Python Podcasts while driving

About the Presentation

This Presentation is:

- An introduction to Ansible
- My first public speaking session like this
- Hopefully includes live demos

This Presentation is not:

- A deep dive into Ansible
- A whole bunch of Python Code
- Very Long

What is Ansible?

From the Ansible website (https://www.ansible.com)

(Ansible is) Simple, agentless IT automation that anyone can use

From the Presenter:

Ansible is a program that reads yaml files as commands and executes the commands on local or remote systems.

Live Demo - Compile Python 3.8

Devices

Ansible Host - A virtual environment on my Mac

Destination - Centos 7 Minimal VM running on my Mac

Use Case

You have to figure out how to set up your development environment.

Solution

Document it and deploy it in Ansible, so then when you go to Staging or Production, you don’t have to remember what you need.

Why use Ansible?

- Easy to read playbooks (yaml files) for “non programmers”
- Allows for doing infrastructure as code
- Can be as simple or complicated as you’d like
- Allows you to
  - build your dev environment the same way every time
  - sharing your dev environment configuration with other people
  - document your environment in a way you’ll maintain

Things you can do with Ansible

- Manage Linux Machines
- Manage Windows Machines
- Manipulate SSH devices (Network Routers / Switches)
- Manage yourself
- Interact with APIs (Rest or XML)
- Work with XML
- Collect data
- Work with Git Repos
- Much much more...

How to install Ansible

Easy Way (cause we're Python Developers)

- **pip install ansible**
- **pip3 install ansible** (if you're on an older distro with Python installed)
- Recommend doing in a Virtual Environment
- **pipenv install ansible -** Good way to install in a Virtual Env

Another Way

- **apt-get install ansible** (Ubuntu/Debian)
- **yum install ansible** (Fedora/Redhat)

** Demo installing ansible live **I

It’s installed, now what?

Command to type

- **ansible --version** (Verify Ansible is installed, make sure it’s 2.8.x)

Various Components

- **ansible** - Allows to run individual ansible commands from the command line
- **ansible-playbook** - Runs plays from playbooks (yaml files)
- **ansible-galaxy** - Tool for sharing, downloading, and creating Ansible Roles. Roles are collections of playbooks.
- **ansible-vault** - Secrets Management for Playbooks

Running a simple playbook

- **ansible-playbook simple_play.yml**
- Runs on the localhost
- Shows off the format of an Ansible Play
- Shows off how to loop and debug
- Doesn’t do much else

** Live demo and code review **

Adding Conditions

- **ansible-playbook asserting_play.yml**

- You can do testing in Ansible
  - Don’t go overboard, Ansible is not a test framework
  - You can making branching plays though
    - For example, Install Python one way on Ubuntu and another on CentOS
  - **assert that** something exists, matches a condition, or many other things

** Live Demo and code review **

Working with State

- Ansible is (when it can be) Stateful
  - If you state a file exists and it doesn't it will create it
  - If it already exists, it will note it exists and do nothing
- This varies on a module by module basis
- The shell / raw / command modules are examples of non-stateful modules
- The git and file modules are stateful modules
- Check the docs for more details

All the other things you can do

- Deploy VM’s in major cloud providers
- Write your own Ansible modules using Python
- Manage hundreds of remote systems

All the prebuilt modules

https://docs.ansible.com/ansible/latest/modules/list_of_all_modules.html?highlight=modules

Managing Remote Systems

- Connecting to Remote Systems
  - Preferred Method is setting up SSH keys on remote linux boxes
    - Only works for devices that connect via SSH
  - Login / Passwords
    - Standard method if SSH Keys are not possible
    - Can be made more secure by using Ansible Vault

Live Demo (In theory) Deploying a Project

- This takes a while, so may not get through
- Can set up a play to do the following:
  - Install OS Packages
  - Install Python Packages
  - Clone Git Repo
  - Set up Virtual Environment and install Packages in the Virtual Environment

** Live demo and code review **

Centralized Management

Ansible is neat, you have written several plays, you want others to run them but not edit them.

Ansible Tower - https://www.ansible.com/products/tower

- Costs Money

AWX - https://github.com/ansible/awx

- Open Source
- Upstream version of Tower with all features unlocked
- No paid support options (Open Source community only)

Additional Resources

Technical Reference

- https://docs.ansible.com

Training

- https://www.linuxacademy.com

Contact Info

Work Email: joej@ignw.io (if you want to talk business)

Personal Email: joej164@gmail.com (if you just want to ask questions)

Twitter: Nope

Linked In: https://www.linkedin.com/in/joseph-jacobs-92b4b5112