# Day 3

# Agenda

- Advanced Ansible
  - Ansible Dynamic Inventory, Variable Prompt, Role, Troubleshooting, Roles,Tags & Limits, Rolling Update, Local Action, Configuration drift, development team usage
- Advanced Terraform
  - Variable, Software Provisioning with Provisioners, Output attribute, Interpolation Expressions, Locals, Data Sources, Modules, Backends and Remote State, Identity & Access Management (IAM), Autoscaling, Troubleshooting Terraform, Business cases to use Terraform, Terraform for multi-cloud
- Real world experience - use case

# Ansible Dynamic Inventory

If your Ansible inventory fluctuates over time, with hosts spinning up and shutting down in response to business demands, the static inventory solutions described in How to build your inventory will not serve your needs. You may need to track hosts from multiple sources: cloud providers, LDAP, Cobbler, and/or enterprise CMDB systems.

Ansible integrates all of these options via a dynamic external inventory system.

If you use Amazon Web Services EC2, maintaining an inventory file might not be the best approach, because hosts may come and go over time, be managed by external applications, or you might even be using AWS autoscaling. For this reason, you can use the EC2 external inventory script.

# Prompts

When running a playbook, you may wish to prompt the user for certain input, and can do so with the ‘vars_prompt’ section.

A common use for this might be for asking for sensitive data that you do not want to record.

This has uses beyond security, for instance, you may use the same playbook for all software releases and would prompt for a particular release version in a push-script.

The most common strategies for debugging Ansible playbooks are using the modules given below −

## Debug and Register

These two are the modules available in Ansible. For debugging purpose, we need to use the two modules judiciously. Examples are demonstrated below.

## Use Verbosity

With the Ansible command, one can provide the verbosity level. You can run the commands with verbosity level one (-v) or two (-vv).

https://docs.ansible.com/ansible-tower/2.2.0/html/administration/troubleshooting.html

# Role

C:\Users\admin\Downloads\C08412\Ansible 2 for Beginners [Video]\Code section 3\hands-on-ansible-master (1)\hands-on-ansible-master\02-playbooks\02-install-mattermost

/home/osboxes/ansible_training/Ansible1/Code section 5/cheat_sheets/troubleshooting_and_debugging.md

/home/osboxes/ansible_training/Ansible1/Code section 5/02-playbooks/00-simple-playbook-examples/debugging/debug_control_flow.yml

https://docs.ansible.com/ansible-tower/2.2.0/html/administration/troubleshooting.html

# Delegation, Rolling Updates, and Local Actions

https://docs.ansible.com/ansible/latest/user_guide/playbooks_delegation.html

/home/osboxes/ansible-examples/language_features/batch_size_control.yml

# Drift

/home/osboxes/ansible-examples/language_features/ansible_pull.yml

Advanced Terraform

Terraform use case, Variable, Software Provisioning with Provisioners, Output attribute, Interpolation Expressions, Locals, Data Sources, Modules, Backends and Remote State, Identity & Access Management (IAM), Autoscaling, Troubleshooting Terraform, Business cases to use Terraform & Ansible, Terraform for multi-cloud

/home/osboxes/terraform-course

# Terraform use case

https://www.terraform.io/intro/use-cases.html

# Exe

Build solution to deploy enterprise level e-commerce application

# Ansible & Terraform

/home/osboxes/ansible-terraform

/home/osboxes/ansible-packer-terraform-demo

# Multi-cloud terraform

https://www.hashicorp.com/products/terraform/multi-cloud-compliance-and-management/

https://github.com/hajowieland/terraform-kubernetes-multi-cloud

# Real world use case - DevOps transformation

- Current State
- Future State
- Strategy and implementation

# Current State

- People
- Process
  - Cultural issue
  - Waterfall model
- Technology
  - Application running on-prem
  - Little adoption of Azure
  - Provisioning takes weeks
  - Manual provisioning and deployment,
  - Manual testing
  - Errors are high
  - Monolithic application
  - Security issues
  - Least monitoring

# DevOps consulting

- Training
- Coaching
- Implementation

# DevOps consulting

- Training
- Coaching
- Implementation

# End result

- People
  - **Unlearn, DevOps Culture**
- Process
  - **Scrum**
- Technology
  - **Full adoption of Azure, GCP, AWS**
  - **Provisioning takes minutes**
  - **Automated provisioning and deployment,**
  - **Automated testing**
  - **Microservices with Kubernetes**
  - **DevSecOps Security**
  - **Automated monitoring**
  - **SRE**
-