



System Administration & Security

COMP 175 | Fall 2021 | University of the Pacific | Jeff Shafer



ANSIBLE

Ansible
(IT Automation)



Creating Infrastructure

Manually creating instances

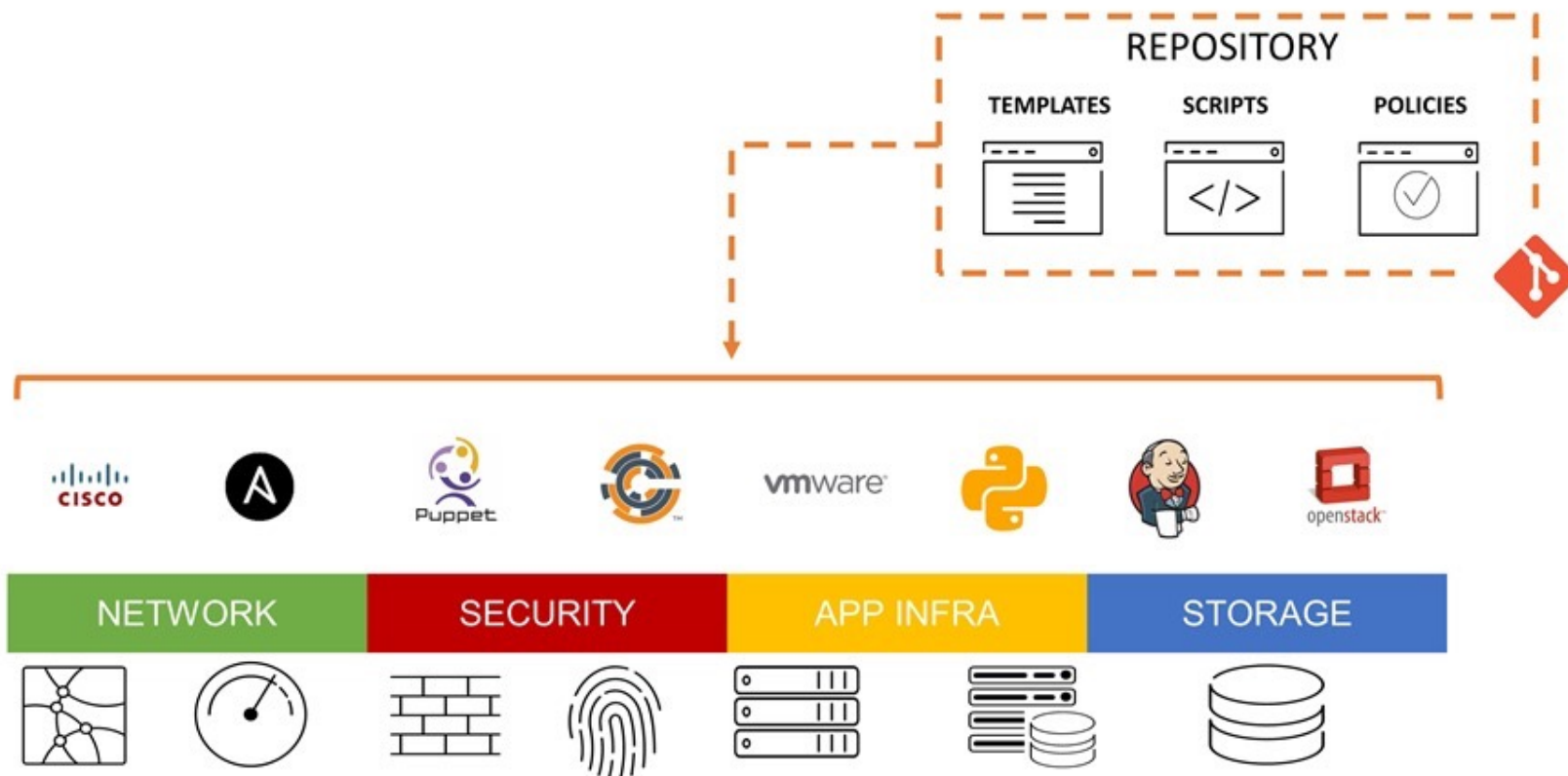
Manually installing software

Manually changing settings

Infrastructure as Code

- Write “code” that defines
 - Instances that need to exist
 - Software that needs to be installed
 - Settings that need to be set
- Track code in version control
- Press a button and let a tool ensure that current infrastructure state matches desired state (in code)
- Benefits
 - Faster deployments (automation, not human)
 - Greater reliability (consistency, reduce human errors)

INFRASTRUCTURE as CODE



Infrastructure Automation Tools

- Ansible: <https://www.ansible.com/> (“Playbooks”)
- Chef: <https://www.chef.io/> (“Recipes”)
- Puppet: <https://puppet.com/> (“Manifests”)
- Terraform: <https://www.terraform.io/>
- SaltStack: <https://www.saltstack.com/> (VMware)
- AWS CloudFormation:
<https://aws.amazon.com/cloudformation/>

Infrastructure Automation Tools

Different Styles

- **Declarative**
 - “Functional”
 - Code specifies what the eventual state of the system should be
- **Imperative**
 - “Procedural”
 - Code specifies what specific steps must occur (in sequence)

Different Architectures

- **Push**
 - Control server pushes configuration to managed systems
- **Pull**
 - Managed systems pull configuration from control server

Ansible Overview

You
Need
TO
LEARN



Idempotency / Desired State

- Most Ansible modules check whether the desired final state has already been achieved
 - State achieved? Exit without performing any actions!
 - **Repeating the task does not change the final state**
 - Key benefit over *a pile of shell commands*
- **'Idempotent' modules**: Whether you run a playbook once, or multiple times, the outcome should be the same

Ansible & AWS



**Create AWS EC2
Linux Instance using
Ansible**

Wrap-Up

➤ Questions?

➤ Concerns?

➤ **This Week**

➤ **Lab 14** – Ansible

➤ **Next Week**

➤ Tuesday – *Lab Work Day*

➤ Thursday – No School

➤ **Last Week**

➤ Tuesday – Backups

➤ Thursday – Monitoring