DEVOPS: IT’S NOT DEV AND IT’S NOT OPS

Jesse B. Crawford (NMT ’15)
LEGACY OPERATIONS

Provision Hardware
- Rack and stack
- PI's label cables

Configure System
- Roll over the console
- Click buttons
- Edit text files

Deploy Software
- SCP a WAR
- Insert CD 3

Operate/Maintain
- RDP in once a week
CLOUD OPERATIONS

Provision Resources
- Terraform

Configure System
- Puppet

Deploy Software
- Docker

Operate/Maintain
- Revision Control
INFRASTRUCTURE AS CODE

- Write it like code
- Test it like code
- Control it like code
- Deploy it like code
- Maintain it like code
LEAN ON PROVIDERS

- Infrastructure as a Service (IaaS)
- Platform as a Service (PaaS)
- Application-specific service providers
  - Communications
  - Integrations
  - Data sources
  - Etc.
Abstract operations into software
SELECTED TOOLS
AMAZON WEB SERVICES

- Complete IaaS provider
- Oldest in the game
- Broad portfolio of services
TERRAFORM

- Cloud provisioning system
- Driven by YAML input files
- Support for full set of AWS services and other service providers
- Maintains as well as instantiates
resource "aws_instance" "gitlab" {
  ami = "ami-46c1b650"
  instance_type = "t2.medium"
  key_name = "awstest"

  ebs_block_device {
    device_name = "/dev/sdh"
    volume_size = 5
  }

  provisioner "remote-exec" {
    script = "init_puppet.sh"
    connection {
      type = "ssh"
      user = "centos"
      private_key = "${file("/home/jesse/.ssh/id_rsa/aws")}"  
    }
  }
}
resource "aws_route53_record" "gitlab" {
  zone_id = "Z2RV5O3Z63TSTU"
  name = "gitlab.awsd.jbcrawford.us"
  type = "A"
  ttl = "60"
  records = ["${aws_instance.gitlab.public_ip}"]
}
PUPPET

- Configuration management system
- Driven by Ruby-like DSL
- Highly declarative
- Fairly “enterprise” solution with many moving parts
- Very strong community, open source modules
• Data generally separated from code
  • Data retrieved from external sources or hiera
• Node configuration is based on classification
• Common design pattern is role—profile
PUPPET

- Configuration content should be idempotent
  - Do not `exec`

- Configuration content should be verifiable
  - Do not `exec`

- Configuration content should be as abstracted as possible
  - Do not `exec`
---
classes:
  - 'role::gitlab'

class role::gitlab inherits role::base {
  include docker
  include profile::gitlab
}
class profile::gitlab {
    exec { 'format-storage':
        command => '/usr/sbin/mkfs.ext4 /dev/xvdh',
        unless  => '/sbin/blkid -t TYPE=ext4 /dev/xvdh'
    }

    file { '/var/gitlab':
        ensure => 'directory'
    }

    file { '/etc/gitlab':
        ensure => 'directory'
    }
}
mount { '/var/gitlab':
  device => '/dev/xvdh',
  fstype => 'ext4',
  ensure => 'mounted',
  options => 'defaults',
  atboot => 'true',
  require => [ File['/var/gitlab'],
               Exec['format-storage'] ]
}
file { '/etc/systemd/system/gitlab.service':
    ensure => 'present',
    content => file('profile/gitlab/gitlab.service'),
    notify  => Exec['daemon-reload']
}
[Unit]
Description=Gitlab
After=docker.service
Requires=docker.service

[Service]
TimeoutStartSec=0
Restart=always
ExecStartPre=-/usr/bin/docker stop %n
ExecStartPre=-/usr/bin/docker rm %n
ExecStartPre=/usr/bin/docker pull gitlab/gitlab-ce
ExecStart=/usr/bin/docker run --rm --name %n \\
  --hostname gitlab.awsd.jbcrawford.us \ 
  --publish 443:443 --publish 80:80 --publish 8022:22 \ 
  --volume /etc/gitlab:/etc/gitlab:Z \ 
  --volume /var/gitlab:/var/opt/gitlab \ 
  gitlab/gitlab-ce

[Install]
WantedBy=multi-user.target
DOCKER

- Containerization system... Think of it like lightweight VMs
- Or heavyweight virtualenv
- Contains dependencies and environment
- Enables rapid and reliable deployment
DOCKER

- Implemented using kernel’s namespacing features
  - And a great deal of duct tape

- Uses ‘union file system’ and emphasizes stateless containers

- Containers built procedurally
FROM ubuntu:14.04
MAINTAINER Sytse Sijbrandij

RUN apt-get update -q \
    && DEBIAN_FRONTEND=noninteractive apt-get install -yq --no-install-recommends ca-certificates openssh-server wget apt-transport-https vim nano

RUN echo "deb https://packages.gitlab.com/gitlab/gitlab-ce/ubuntu/`lsb_release -cs` main" > /etc/apt/sources.list.d/gitlab_gitlab-ce.list
RUN wget -q -O - https://packages.gitlab.com/gpg.key | apt-key add -
RUN apt-get update && apt-get install -yq --no-install-recommends gitlab-ce
CONCLUSIONS

- DevOps is *heavily* open-source with enterprise options
  - E.g. Kubernetes vs. Red Hat OpenShift

- Major emphasis on reuse of services
  - But you need to know the right point to start in-housing

- Major emphasis on horizontal scaling over vertical scaling
  - Know which workloads this does not work well for

- Connected at the hip to agile software development

- Security is a hard problem
Shameless plug: I will also take questions about information security and national security careers. Or anything really.

Jesse B. “jeanluc” Crawford

https://jbcrawford.us

jesse@jbcrawford.us
jcrwford@cs.nmt.edu
jbcrawf@sandia.gov
jbcrawf@iastate.edu
ae5jl@arrl.net
jesse.crawford@pcc.edu
jcrawford@sdf.org
jesse.crawford@redcross.org
jesse@waffle.tech