Using Ansible and Terraform to try out that new toy

AKA prototyping a new tool

What is trying a new tool like for you?

In the bad old days:

- Beg or plead for a new server or VM
- Read docs
- Install software
- An outage forces you to spend 3 days on other work
- Configure software and test
- Reconfigure and test
- Return in 2 weeks after fixing a critical problem with another team
- Try to remember all of your previous work(read your .bash_history) so that the intern can put this into production

SO

Your boss just had a GREAT IDEA*

* After returning from that all expenses paid conference

Tools such as Terraform and Ansible have the following benefits

- Your projected is documented as you work. No more trying to remember that weird setting you found but forgot to document. The change was made via software and there is a history! (via SCM)
- You can repeat your work to validate your process. Once you get your software installed, destroy it all and try, try again.
- You can automate your new tool with continuous integration and deployment tools to works with your current deployment process
- Anyone on the team or in the company can follow this process and collaborate with each other

An (incomplete) introduction to Terraform

“Terraform is a tool for building, changing, and versioning infrastructure safely and efficiently.”

https://www.terraform.io

https://github.com/hashicorp/terraform

Terraform allows you to define your application infrastructure as code with your project.

- Infrastructure as code
  - Versioning of infrastructure
  - Share and reuse infrastructure components
  - Repeatable and automated processes
- Multi-cloud aware
  - OpenStack
  - AWS
  - Google Compute
  - Azure

Elasticsearch Logstash Kibana

^^ The boss's great idea ^^

ELK is hard and will require lots of changes while we figure it out and wow does it need a lot of resources and this could take a whole team!

Trying out ELK on the first day

Let’s create!

- A new SCM repo
- A virtual cluster of instances for each service
- Ansible roles to install, configure and manage each service
- Tests to validate the work

SCM

An SCM is a great way to track the progress and history of your work. In this case a git repo hosted on Github will be used

DO:

- Branch when it makes sense. For Git, branch based on a feature or goal
- Commit early, commit often
- Use pull requests or other code review techniques to keep team members active in the project

DON’T: forget to push your branch to the origin server so that you can collaborate with others

(virtual) Hardware

Using Terraform to describe and implement:

- A VPC and networking for the project
- Security groups
- EC2 instances to run each service in the ELK stack

Ansible Roles

Galaxy to the rescue!

In particular, geerlingguy

Testing the work

The first test is Ansible success. playbooks finish successfully.

You’re making progress if your roles and

Testing with Ansible

Fin

Links:

- Terraform and on Github
- ELK
- This presentation