

DevOps @ HydraConnect: Deploying Hydra with Ansible and AWS
Code is now found at: <https://github.com/curationexperts/ansible-hydra>

Start screencast, introduce selves & institutions!

Section 1: Anna

Background of collaboration

- Hired to implement a hydra app
- So the first thing I needed to do was set up a hydra environment
- I thought back on all the times I had to do this over and over again for my last project: development environments for me and others, production environment, staging environments, building new production servers which somehow always means starting over instead of being able to migrate anything
- I was tired of slogging through 5 screens full of instructions on a wiki every time I was going to have to do this.
- I wanted to push 'start' and go do something else.
- So I started setting up my dev environment and automated as I went along.

Why Vagrant over other deployment / configuration management applications?

- Lightweight
 - Agentless - no server or client code to install
 - runs locally over SSH
- Low barrier to entry
 - Ansible's focus is simplicity and ease-of-use, which makes it a particularly good choice for smaller organizations or departments.
 - However, it is capable of handling enterprise environments so it will grow with your organization.
- Broad audience
 - No programming languages required - everything is set up through config files. This makes it easier to train new people, whatever background they may be coming from.
 - Easy-to-read, self-documenting deployment and configuration
 - Clear, thorough documentation
- Easy to share (like all deployment managers)
 - Put it under version control
 - use ansible vault

Why use AWS?

- Local constraints
 - In-house IT is windows-only
 - history of bandwidth issues when, e.g. a google doodle causes a spike in traffic.
- Ubiquitous (most importantly, Alicia had used it a lot)

- It works
 - makes backups, upgrades, and maintenance easy
 - flexible in response to growing demands (traffic, space)
 - cost-effective

Section 2: Alicia

The mechanics of using ansible on AWS :

- Set up on AWS from a brand-new account
 - create an account
 - easy with a credit card plus your name, email, snailmail, and phone
 - you're eligible for Free Tier for one year
 - AWS has lots of services (Glacier for long-term storage, S3 for shorter-term storage, etc.) - for ansible scripts you need two - IAM and EC2
 - set up an IAM group
 - the group stores permissions (can the IAM user start new instances? terminate running instances? create backups? delete backups?)
 - easy option is AmazonEC2FullAccess
 - set up an IAM user
 - Ansible uses the IAM user to start up new EC2 instances
 - download the credentials and store safely
 - change IAM user credentials periodically
 - add the user to the group
 - create or upload a keypair
 - this keypair authenticates your ssh connection to your new machine
 - set it in group_vars/all to set the keypair on the remote machine
 - pass the private half on the command line to authenticate
 - create a security group
 - the security group governs port access on your new machine
 - basically iptables - can be open to specific IPs or ranges
 - find the instance ID - basically the OS - ansible assumes a bare bones Ubuntu 14.04 instance - can be a black art - instance IDs are specific to your region
 - find the instance type (size in CPU/memory) you want - easy to change
 - pull it all together in the group_vars/all file:
 - IAM user's access key & secret access key
 - EC2 keypair & security group
 - instance ID
 - instance type
 - encrypt the file with ansible vault if you want to store it in a public code repo

Architecture choices

- This architecture works well on AWS - could make other choices

Vagrant use cases

- prod mimics the system setup for system troubleshooting
- dev as close to possible as the production environment, but still usable for developing.

Possible future improvements

- Handle code deployment with Ansible (replace Capistrano)
- Add tests to the Ansible playbook (output of ruby -v)
- Use dynamic inventory (esp. if you spin up machines in response to demand)