Ansible & Configuration Management at ECDF

Robert Currie Scotgrid F2F @ Durham, Feb 10th 2017

So what does Ansible do?

Ansible is a push based configuration managment tool.

This allows for configuration files to be written using a template language (Jinja) and have their deployment managed using ansible-playbooks (YAML).

Ansible makes use of the standard openssh library to push commands/files to the machine it's deploying. This means all that's required from the host is the ability to connect over ssh.

Ansible can make use of different user accounts but I've been working under the assumption that we will deploy most services using root for now.

Ansible vs Alternatives

We're looking to move resources @ ECDF to VMs. Should we deploy these services using a configuration management tool or do it all manually.

Options considered:

Option	Puppet	Ansible	Salt/Chef/other
Workflow	Pull model	Push model	Usually Pull
Language	Ruby?	Python (v2.5+)	Typically other
Configuration	Puppet Forge	YAML & Jinja	Json/YAML/...

I've previously had some experience with Puppet in but I have been playing with Ansible for a bit and liking what I see.

Reasons why Ansible is Awesome

- Very low barrier to entry

  I was able to hack-at/modify an existing bdii recipe to deploy a bdii test-server in about 1hr*. I thought was nice as it was my first time using bdii or ansible.

- Grouping of hosts is nice and efficient

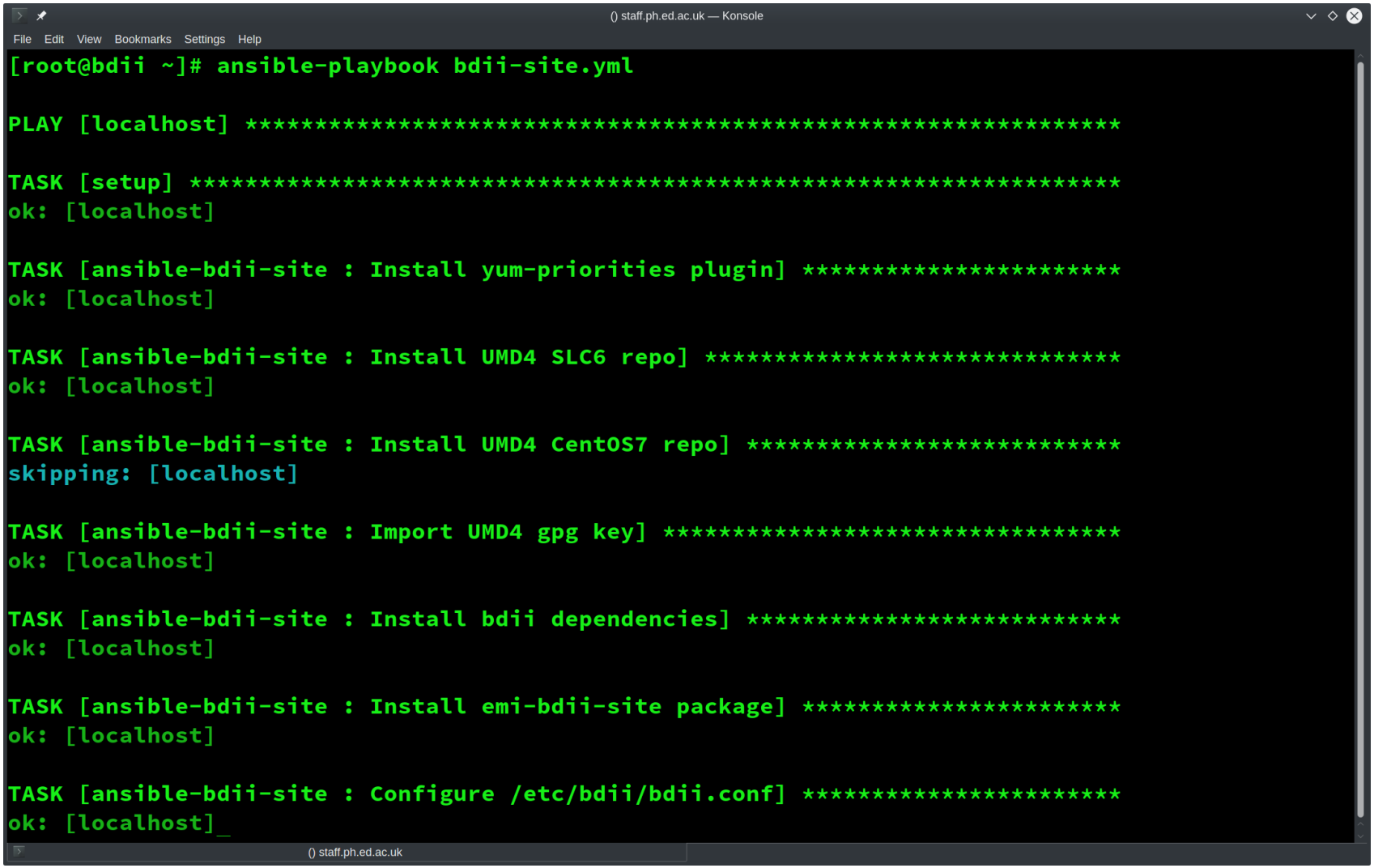
  This makes it easy to know what I want to run and where

- Configuration becomes a bit easier to see

  The configuration system inside Ansible makes it clear what needs to be edited to setup a given service.

*Shamelessly using https://github.com/lancsgrid so thanks Lancaster!

Ansible working



What am I looking to do with Ansible

My personal plan is to use Ansible to be able deploy most services in Edinburgh in the near future.

This is a very nice way of snapshotting the current config of a service even if we deploy it manually.

- Automate the audit of our resources @ ECDF
- Automate some monitoring tasks
- Track the deployment of services
- Centralize/snapshot various service configurations

Maintaining/Monitoring a site

Ansible is used a lot to deploy/monitor sites and give a nice web-UI as to what is going on.
This can be done in one of various ways:

- Semaphore
- Rundeck
- Ansible Tower
- crontab

Ultimately we may not want to expose the site management system via https so a web-UI may be of mixed use.
After playing with Semaphore and Rundeck I'm not entirely convinced they offer much over crontab.

Bookkeeping

The best solution feels like some simple bash scripts run by crontab on a central configuration server.

The plan is to have the configuration stored in gitlab for history purposes and to run the ansible service from this using bash.

Crontab can email us the result of the stderr on a failure to monitor/deploy a service.

Broadly speaking if we maintain a separation between playbooks used for deploying vs monitoring then life gets easier.

Potential cons of using Ansible

Ansible is quite good but potentially has some risks:

- Ansible can check the status of a file/service but this is little guarantee it didn't touch something.
  This is up to the person writing the playbook.
- Storing secrets with Ansible configs requires some care.
  Ultimately I think storing passwords in a secure gitlab isn't too bad.
- Ansible doesn't offer an automatic roll-back.
  If a user makes a playbook which will put the system into a bad state by accident there is little to undo what has just been done.

Conclusion

Personally I think Ansible is the best thing since sliced bread.
Or at least since Puppet.

I've made some simple playbooks for Ansible already and these work as expected.
Low entry barrier makes this a nice tool to use and doesn't require getting certified or spending lots of time/effort.

When in doubt Ansible can be run from the command line and has a "cmd" option to run a raw command over ssh (very useful).

Thanks for listening