I was lead DevOps,

Github was going by the the ALM team- that’s the application lifecyclte team

I helped them move the config,

Name it better, verb noun syntax - **GNU Coding Standards**

Undertand git, understand branches, each time you do a pull your pulling the whole metadata of every commit into your index file and differiantial of every change so offline you can look at every change on that trunk. Forced a little re-write.

Using SourceTree to browse that index stage file ( file that contains every change (branch merge) with its description.

Github branching history

Creating users and ssh keys

SSH Key - .ssh in user profile – ID\_RSA and ID\_RSA\_Pub file

Users – could be stored in the global config file

Python

**Test Stage**

**Build Stage**

Artefact **(python?)**

**Release Stage**

Docker realise image

**Deploy Stage**

Testing

Load testing

Security testing

Acceptance testing

**Statement of Work - Puppet**

**Overview**

* Currently, every Unix OS instance requires at least some OS configuration. For some applications, this can be very simple and is an automated part of the standard build process – e.g. the delivery of access rights to the server.
* The current Puppet environment. as employed by the application teams lacks these controls and consistency as well as reporting.
* Some of the key requirements for this to work are:
  + Fully supported and ICAP controls compliant version control and access control
  + Reporting of deployment status, including versions
  + Dedicated (knowledgeable and consistent) support for creation and maintenance

Goal: To provide a repeatable consistent deployment mechanism for Linux Operating systems and their application configuration and provide a transparent view of current & target state. Enable self service tools to push the configuration to its target state.

**Infrastructure & Governance**

Analyze the current Puppet infrastructure design and implementation and propose and build solutions to:

* Ensure platform is resilient, sustainable & up-to-date
* Introduce release management process for platform and application configuration content
* Analyze current content, looking to optimize current module usage for re-use
* Ensure compliance with Technical Standards
* Engage with risk team for a Risk Assessment, and address any findings

**Content & Configuration**

We already have framework for reporting & asserting configuration in Puppet. We would therefore look to enhance this set of modules to support the additional checks required.

* OS Build and Patch Level
* Core Operating System configuration
* Networks Interfaces configured and conforming to agreed resilient setups
* Firmware levels applied to the hardware (not available today)
* Hardware configuration summary CPU/Memory specifications
* Location, Environment & Application detail summary
* Access and Security components
* This reporting would current and desired state of the servers in scope.
* Puppet would also be used to provide a Service Delivery Quality Assurance process "stamp of approval" that a server has been delivered to specification
* End to end integration of puppet execution with server deployment workflows and request system.

Adding users to the relevant groups

Configuring the private and public key ssh keys

SElinux security stuff – enchanced or lockdown

Firewalld mange iptables

Adding repors to yum