**Script**

**Risk Assessment**

With this project one of the largest risks to overcome was the lack of knowledge on the tools in which the application we’re deploying was built.

**Cost of the Project**

**Service Infrastructure**

**CI Pipeline**

**AWS**

We used AWS to host our virtual machines as a secure and easy place to run the application.

Before we could set up the infrastructure to automatically deploy the application, we first installed and ran the application manually on the vms. This was so we understood what needed to be done. The dependencies and the deployment order that needed to take place.

Hosting on AWS was also useful so all of us could access, edit and upload files to and from the vms.

**Jenkins Build**

Jenkins is used to automatically update the entire application over its many different vm deployments. It does so using a Jenkinsfile.

The Jenkinsfile will run again every time a new commit is made to the main branch of the git repo. Meaning the updates are automatically applied to the application deployment as and when they are added.

Using Jenkins like this allows for easier, automated deployment and version control. You can easily see what is different and roll back should the need arise.

**Ansible**

Ansible is a tool used to automate the deployment of work throughout the entire infrastructure.

It is an invaluable tool for a system admin as it allows a user to deal with; application deployment, configuration management, intra-service orchestration and basically anything else a system admin would need to use on a daily basis.

The tool we used the most from ansible in this project were its playbook.yaml files. These allowed us to set up vms with the software they required to run. This allowed for our main automotive process in establishing the vms. This also means if a vm ever needs updating or changing, doing so in the playbook file will automate this in conjunction with Jenkins.

**Terraform**

**Docker**

**Docker Swarm**

**NGINX**