[[Technical Assessment]]

The following tasks should be completed, and all scripts, docker-compose, and ansible files pushed into a public git repository (eg GitHub).
Clear instructions on how to run each of your deliverables should also be included.

There are three sections within the technical assessment. There are also "Bonus Tasks" in each section which are optional.

1. PYTHON OR BASH SCRIPTING
- Write a script that will rename all files that have a '.wav' suffix in a given directory to 'audiofile_DATESTAMP_###.wav'
- where ### is a zero-padded integer that is incremented for each file encountered
- where DATESTAMP is in YYYY-MM-DD format, and is derived from the creation date of the file

Bonus Tasks (optional)
- extend code to be able to work with a different user-defined prefix and suffix - eg 'videofile_DATESTAMP_###.mp4'
- implement argument management, so that directory, prefix and suffix can all be specified at runtime
- implement functionality to rename files in reverse-alphabetical order

2. DOCKER
Create a Docker-Compose multi-container application, that achieves the following goals:
- hosts a simple html page on two separate nginx containers
- the html page should contain the word "blue" on one container and the word "green" on the other
- place a haproxy loadbalancer container in front of nginx containers
- enable a method to view haproxy statistics via web interface

Bonus Tasks (optional)
- configure nginx access logs to be collected and stored in an ElasticSearch docker instance
- configure kibana to allow viewing these logs from ElasticSearch

3. ANSIBLE
create a simple ansible project (consisting of a playbook and role(s)) to configure a remote ubuntu server:
- use ansible to install postgresql on target server
- use ansible to ensure postgresql runs on server startup
- use ansible to enable inbound postgresql network connections from the network range 10.231.0.0/16 (md5)
- use ansible to ensure that SSH logins only work with ssh keypairs (no passwords)

Bonus Tasks (optional)
- use ansible to configure the firewall on the remote server to allow inbound connections on postgres port
- use ansible to configure a scheduled backup (dump) of the postgresql server