

# Coding Tech Task 1 - Simple Application Deployment

(With this tech task, Afterpay will provide the application (attached), a simple Java/Python/Ruby web application & show application dependencies (like libraries if a framework, etc), preferably on a public github repo)

## Overview

Using scripts & orchestration/provisioning tools we want to be able to easily deploy the following application, running on a Linux OS, into AWS allowing us to connect to the web server on port 80. (Public or Private is your choice)

You will be required to write CaC (Configuration as Code) / IaC (Infrastructure as Code) with tooling for an easy deployment of the application into AWS.

ETA: 2 days.

## Requirements:

- Write scripts (IaC/CaC) using your preferred language & tools to:
  - Configure a Linux OS image (flavour of Linux is your choice) in AWS and:
    - make sure all packages are update to date and all pending security updates are applied against the default OS repositories.
    - disable IPv6 system wide.
    - install the following packages: ntp (NTP daemon)(make sure it is started at boot), telnet, mtr, tree.
    - set the max "open files" limit across all users/processes, soft & hard, to 65535.
  - Create the server from your OS image (as above) in AWS and do any necessary installation/configuration post boot.
  - Select, install & configure an appropriate web server to serve the application.
  - Deploy the provided application on your server.
  - Make the application available on port 80.
  - Ensure the server is locked-down and secure.
- Provide documentation with:
  - Clear instructions for the reviewer to deploy your scripts/code.
  - Requirements for running. (Tooling pre-installed? AWS account?)
  - Explanation of design choices, short comings and assumptions.

# Assessment Criteria (must haves):

- Your scripts in a new git repository showing frequent commits.
- Linux OS image (flavour is your choice).
- Deployed on to AWS.
- Configuration management (Ansible/Chef/Puppet) to install & configure server/application.
- Infrastructure as Code (Terraform/Cloudformation/etc) to deploy infrastructure and/or application.
- Application available on port 80 (using an appropriate web server).
- Server locked-down and secure.
- Anti-fragility: if the server disappears, how does it automatically recover?
- Code / Documentation layout.
- Ease of deployment.
- Simplicity.

## Submission:

Options for submission are:

- Link to **private git repository**: please share with <https://github.com/jameswanafterpay> (Github now supports private repos for free)
- Invite github user **jameswanafterpay** to your private repository
- git-bundle is also acceptable

NB! Please do **not** allow public access to your git repository!