

Lab 1 – nginx intro

Task – Install and setup nginx

Prerequisite – EC2 (t2.micro) with ports 22 and 80 open to the internet.

- 1) SSH connect to your EC2 you have generated
- 2) Run the following commands within the EC2

```
sudo apt update  
sudo apt install -y nginx
```

- 3) Once it is installed using your browser access the public IP of your EC2 and access the nginx website

Stretch goal – Access the nginx website via your terminal using the `curl` command

Lab 2 – nginx conf

Task – Modify the nginx conf to change the web server

Prerequisite – EC2 (t2.micro) with ports 22 and 80 open to the internet with nginx installed

- 1) On the EC2 with nginx installed navigate to where the nginx.conf is stored (location is /etc/nginx)
- 2) Either using vim or nano modify this file (VSCode IDE does not have permissions to modify file)
- 3) Replace the contents of the nginx.conf with the following:

```
events {}

http {
    server {
        listen 80;
        location /hello {
            return 200 "Hello World";
        }
    }
}
```

Stretch goal – As well as /hello when the user enters '<ip>/colour' or '<ip>/food' it should return your favourite colour and food respectively

Lab 3 – nginx reverse proxy

Task – Use nginx as a reverse proxy for a pre built app

Prerequisite – 3x EC2 instances, each with port 22, 80 and 3000 open. Should be called frontend, backend and reverse-proxy respectively

To set up the web app you must SSH into each machine and run the steps for each below

Frontend & Backend

- 1) Update the package manager
- 2) Clone down this repo <https://gitlab.com/Reece-Elder/reverseproxy-nginx>
- 3) Navigate to the frontend or backend folder depending on machine
- 4) Run the setup.bash script (You may need to use chmod +x on script before running)

Reverse-proxy

- 1) Update the package manager
- 2) Clone down the same repo and navigate to the reverse-proxy folder
- 3) Edit the nginx.conf to contain the following (replacing the IP with the public IP of the frontend ec2)

```
location / {  
    proxy_pass http://63.72.210.32;  
}
```

- 4) Run the setup.bash script (using chmod +x if necessary)
- 5) Access the public IP of the reverse proxy EC2 via a browser

Stretch goal – Create another ec2 (spare-nginx) and install nginx on it. When you access the public IP of spare nginx it returns “Hello everyone!”. The reverse-proxy should connect to spare-nginx when the user enters <ip>/proxypass.

Lab 4 – nginx web server

Task – Use nginx to reverse proxy to another nginx web server

Prerequisites – 2x EC2 machines, each with port 22 and 80 open (reverse-proxy and web-server) with nginx installed

Web-server

On the web-server EC2 you should create a directory for your web app files, this should include index.html file with 2 headers, a p tag and an online hosted image.

Modify the nginx.conf to host this .html by using the following code

```
events {}
http {
    server {
        listen 80;
        root /opt/web-app;
        index index.html;
        include /etc/nginx/mime.types;
        location / {
            try_files $uri $uri/ /;
        }
    }
}
```

Move the web app file(s) to /opt/web-app and use `systemctl` to restart nginx

Reverse-proxy

Set up the EC2 to connect to the web-server EC2 when the user enters <ip>/.