# Reflections Cloud City 4 Setup

May 2, 2023, fcohen@starlingwatch.com

### Path and Command Quick Summary

/var/www/node-api/public/

/var/www/node-api/ecosystem.config.js

/home/fcohen/files

/etc/nginx

/var/log/nginx

nginx to access node.js over ssl

pm2 to run node.js as a service

Recorder app is in /var/www/node-api/public/

Recorder download files in /home/fcohen/files

SSL certificates in /etc/ssl/cloudcity

### NodeJS Installation

```
https://docs.aws.amazon.com/sdk-for-javascript/v2/developer-guide/
setting-up-node-on-ec2-instance.html
```

```
curl -o- https://raw.githubusercontent.com/nvm-sh/nvm/v0.39.3/
install.sh | bash
```

```
. ~/.nvm/nvm.sh
```

```
nvm install --lts
```

Now using node v16.20.0 (npm v8.19.4)

```
node -e "console.log('Running Node.js ' + process.version)"
```

Running Node.js v16.20.0

## PM2 Installation

https://medium.com/monstar-lab-bangladesh-engineering/deploying-node-js-apps-in-amazon-linux-with-pm2-7fc3ef5897bb

```
sudo yum update -y
sudo yum install -y gcc gcc-c++ make openssl-devel git
npm install pm2@latest -g
```

https://linux.how2shout.com/how-to-install-node-and-npm-on-amazon-linux-2-aws-ec2/

```
pm2 start ecosystem.config.js
pm2 logs
pm2 save
pm2 start ecosystem.config.js --no-autorestart
```

## Nginx Installation

```
https://www.codegrepper.com/code-examples/whatever/install+nginx+amazon+linux+2
sudo yum install nginx
nginx -v
sudo systemctl restart nginx
sudo systemctl stop nginx
sudo systemctl start nginx
sudo systemctl enable nginx
sudo systemctl status nginx
```

## HTTPS/SSL Setup

nginx ssl installation instruction at:

https://help.zerossl.com/hc/en-us/articles/360058295894-Installing-SSL-Certificate-on-NGINX

logs are at /var/log/nginx

Update /etc/nginx/nginx.conf, /ssl/

**NOTE: You must combine the certificate.crt and ca_bundle.crt files**

/etc/ssl/cloudcity/certificate.crt;

/etc/ssl/cloudcity/private.key;

Then do this from your Mac laptop:

openssl s_client -showcerts -servername cloudcity.starlingwatch.com -connect cloudcity.starlingwatch.com:443 </dev/null

Then copy the second certificate:

-----END CERTIFICATE-----

1 s:/C=AT/O=ZeroSSL/CN=ZeroSSL RSA Domain Secure Site CA

i:/C=US/ST=New Jersey/L=Jersey City/O=The USERTRUST Network/CN=USERTrust RSA Certification Authority

-----BEGIN CERTIFICATE-----

MIIG1TCCBL2gAwIBAgIQbFWr29AHksedBwzYEZ7WvzANBgkqhkiG9w0BAQwFADCB

iDELMAkGA1UEBhMCVVMxEzARBgNVBAgTCk5ldyBKZXJzZXkxFDASBgNVBAcTC0pl

How we Setup code in Reflections3 Server

We have downloaded the Reflections application from Reflections2 server and made required changes. The application directory name is node-api.

After making the changes, we have created a new amazon linux 2 in aws lightsail machine. Then we followed the following steps to deploy the application in the newly created server, we call it Reflections3

SSH into the server

Upload the application directory at /var/www, So the application path is
/var/www/node-api

Navigate to home directory using the command,
>cd

Now we can install nodejs. Please refer NodeJS installation under Installation Section.

We are running the application using pm2 utility. Now we can install pm2. Please refer pm2
utility installation under Installation Section.

After the installation of nodeJS and pm2, we are now ready to test our code. But before that, we
need few things to setup, that we will do in step 7 and 8 below.

With respect to the code, our application stores the .tar files at location
/home/fcohen/files So, we will be creating the corresponding directories using mkdir
command.

[Optional] When we start our application, the application service will be served at port 3000. To
test that, we can open the port under Networking section in aws lightsail portal. (We will not
use this at the end of our setup because we will use nginx to serve the application at port 80)

Now to start the application,
cd /var/www/node-api
pm2 start
Once the application is started it will be served at port 3000, in Reflections3 server it is at http:// 35.89.212.104:3000
Note: - Application might not work as expected because application

uses WebRTC API. WebRTC API expects secured connection i.e, https. For which we will setup self-signed ssl in the coming instructions

Before installing self-signed ssl, let’s install nginx (on which we setup ssl). Please follow the instruction under Nginx installation in Installation Section.

Now to install self-signed ssl, please follow the instruction Self-signed SSL Setup under Installations section
During the setup, please make the modification in the config file at location / section as follows

```
location / {

proxy_pass http://localhost:3000;

client_max_body_size 200M;

}
```

Now the application should be running with secured connect tion i.e, https and at default port.

API Detials

https://35.89.212.104/ - has recorder

https://35.89.212.104/files - will show list of files in html view

https://35.89.212.104/listfiles - will show json format of available files

https://35.89.212.104/<filename.tar> - will serve the tar file

https://35.89.212.104/touch/<filename.tar> - will touch the file