**Week 11**

**11.1 Serverless Fns**

* npm create cloudflare -- my-app
* We are using Cloudflare workers, GCP has functions, AWS has lambda.
* A worker is mini node JS process.
* AWS lambda is most widely used. In serverless backends, we pay per request. It is not ideal at scale.
* npx wrangler login
* npx wrangler whoami
* Whenever there’s an async call inside the function, the function returns a promise of the return type like Promise<Response> or Promise<number>.
* Change in wrangler.toml the name if you want to start another worker.
* Express does not work on Cloudflare workers.
* Hono is a routing framework like express. Wrangler is a CLI for connecting to cloudflare.

**11.2 AWS Deployments**

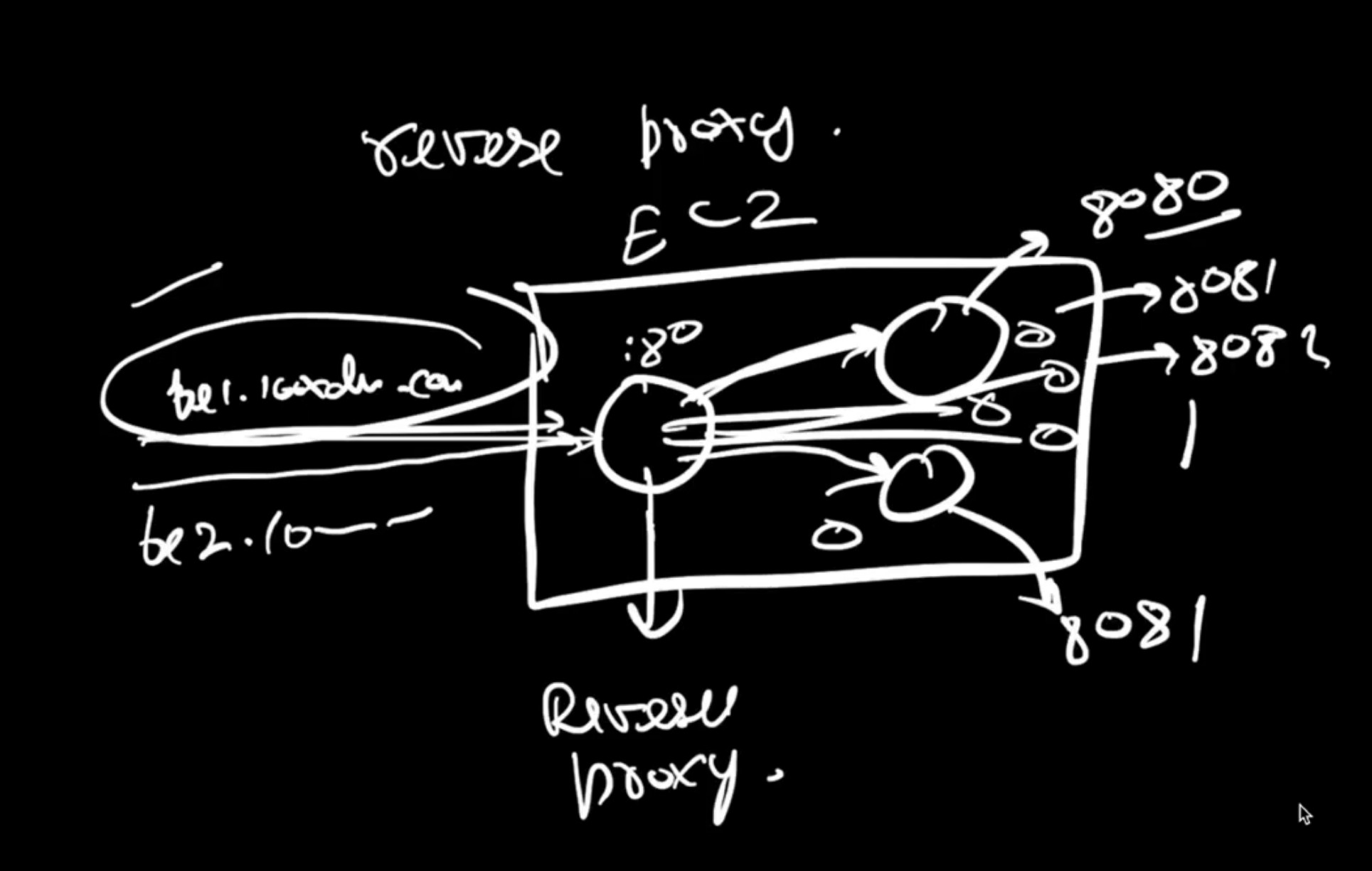
* SSH stands for secure shell.
* SSH is used for connecting to AWS through your mac and deploy your code there. The default port is 22.
* Default port for HTTP is 80, default port for HTTPS is 443.
* Since the default port is 80, you do not need to explicitly write it in the address bar.
* ssh -i saad-test-1.pem ubuntu@13.234.111.39 – for connecting to the server.
* 13.234.111.39 is the IP of your machine.
* chmod 700 saad-test-1.pem. This changes the permissions that the current user logged in to the mac has for the file. It makes it more restrictive. So, other users can’t access.
* Now rerun the connecting command.
* saad-test-1.pem is the certificate file for connecting to the AWS server.
* If you’re EC2 instance does not have access to the internet, it means that your AWS server is not able to a hit a DNS server. DNS server resolves google.com to its IP address. Use sudo vi /etc/resolv.conf to open and edit this file. Paste nameserver 8.8.8.8 and exit the file.
* Sudo mean super user do. I for insert. Esc + : + wq to exit the file.
* For installing nvm on AWS ubuntu:  
  1. curl -o- https://raw.githubusercontent.com/nvm-sh/nvm/v0.39.3/install.sh | bash  
  2. export NVM\_DIR="$HOME/.nvm"

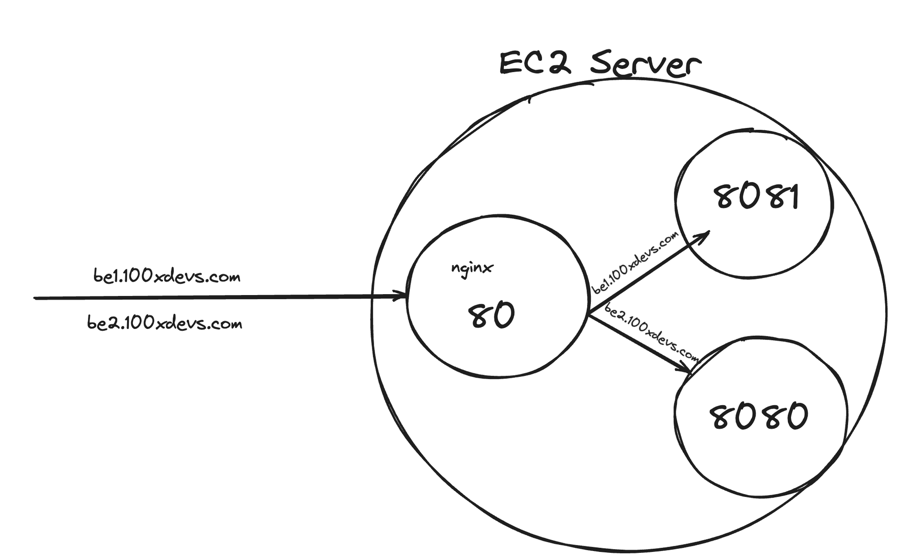
[ -s "$NVM\_DIR/nvm.sh" ] && \. "$NVM\_DIR/nvm.sh" # This loads nvm

[ -s "$NVM\_DIR/bash\_completion" ] && \. "$NVM\_DIR/bash\_completion" # This loads nvm bash\_completion

3. nvm install 20

* This installs node on the ubuntu machine.
* The security group of the instance says only ports 22, 80, 443 are open. We are running backend on 8080.
* Your app should never listen on 80 or 443.
* You do reverse proxy on port 80 and then it can decide whether the backend URL should be sent to 8080, 8081, etc.
* This solves two problems, you do not have to explicitly give port 80 in the URL as it is the default port and multiple backends can connect to port 80 which directs them to different ports.
* sudo rm sudo vi /etc/nginx/nginx.conf – for deleting the default the nginx.conf file.
* sudo vi /etc/nginx/nginx.conf – adding our own reverse proxy (copy format from slides).
* sudo nginx -s reload
* sudo vi /etc/hosts for changing the IP of a domain locally. This way you do not have to buy a domain to test locally.
* lsof -i :8080 – for checking processes are running on port 8080.
* fuser -k 8080/tcp – killing process running on port 8080.





**11.3 Certificate Management**

* Digital Ocean and Vultr are similar to AWS.
* Certbot lets you get SSL certificates for free. Select nginx and Ubuntu 20 from the website: [Certbot (eff.org)](https://certbot.eff.org/)
* sudo snap install –classic certbot
* sudo ln -s /snap/bir/certbot /usr/bin/certbot
* sudo certbot –nginx – This does everything from generating the private key, putting it in the nginx, and changes the conf file to listen on 443.
* This would not work if you are to apply the certificate on a domain you have changed the IP address locally.
* pm2 start index.js – for starting the backend in the background so that you can close the terminal.