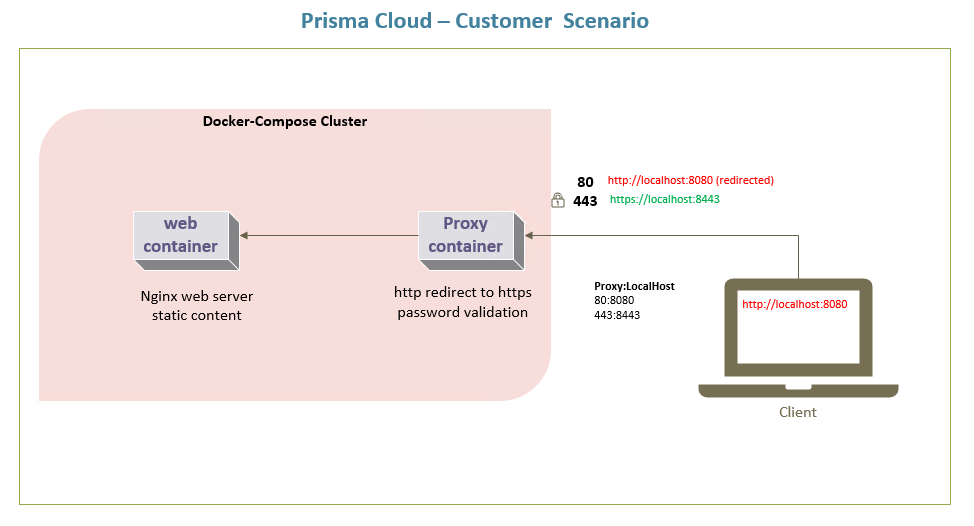
Prisma Cloud SE Customer Scenario Documentation

**Description:** This document outline how to create, secure and validate customer environment which consist of backed web server and front-end reverse-proxy server.

**Figure 1.1 Microservices architecture**

**STEP 1 | Download project files from github.com or from email attachment**

1. ***git clone*** [***https://github.com/idristuna/cse.git***](https://github.com/idristuna/cse.git)

**STEP 2 | Build the web image from Dockerfile**

1. Change directory to the \cse\web
2. Run following command to start to build docker container
3. ***docker build -t web .***

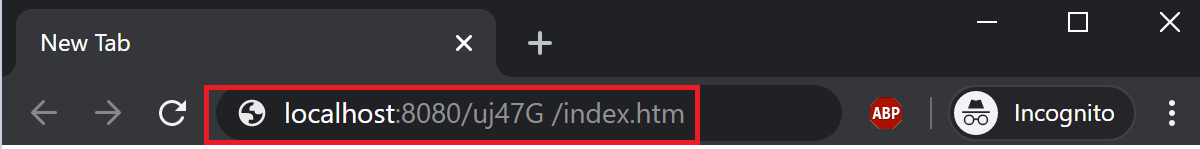
**STEP 3 | Build the reverse proxy image from Dockerfile**

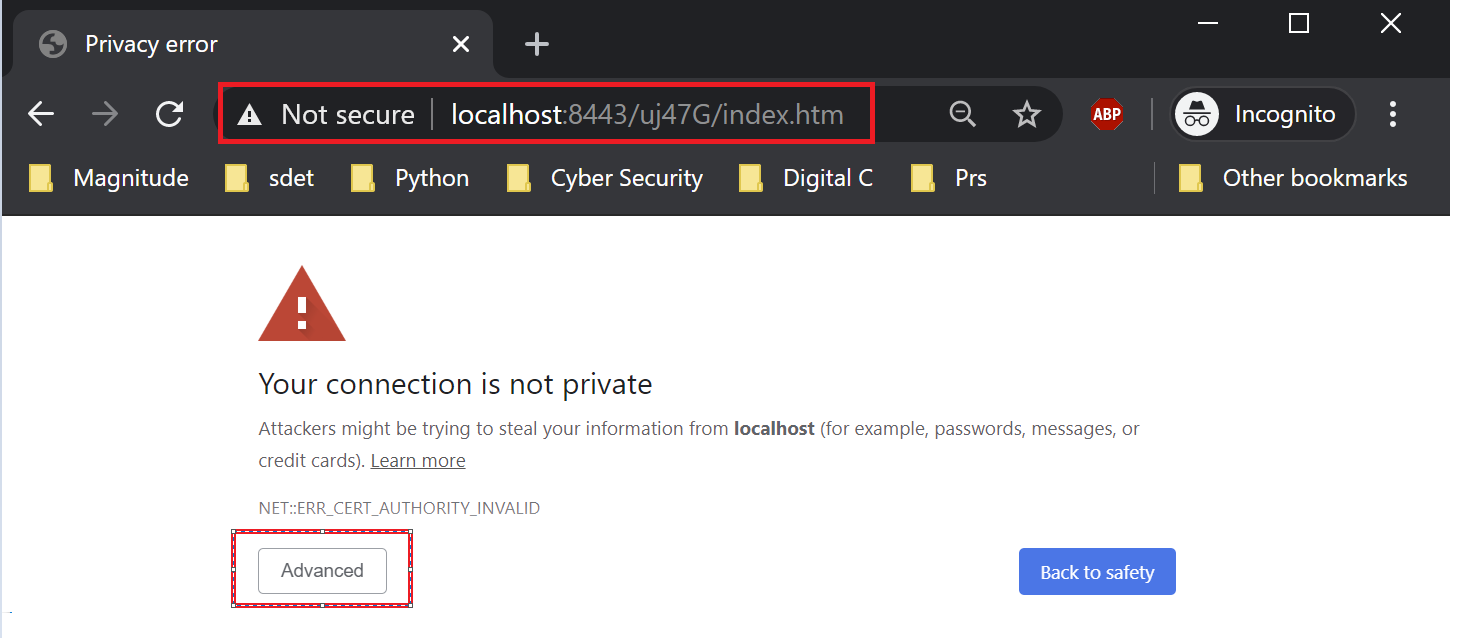
1. Change directory to the \cse\proxy
2. Run following command to build docker image
3. ***docker build -t web .***

**STEP 4 | Build containers from docker-compose.yml file**

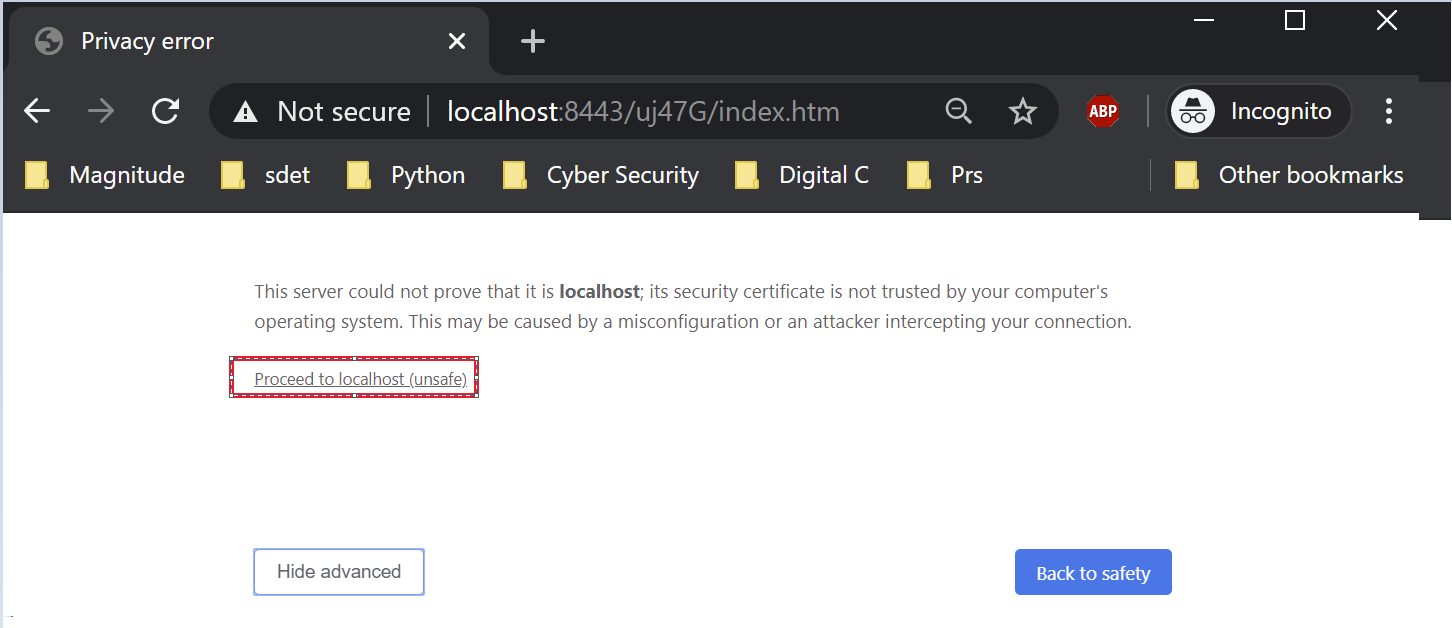
1. Change directory to the \cse
2. Run following command to build and run, web and proxy containers
3. ***Docker-compose up -d***

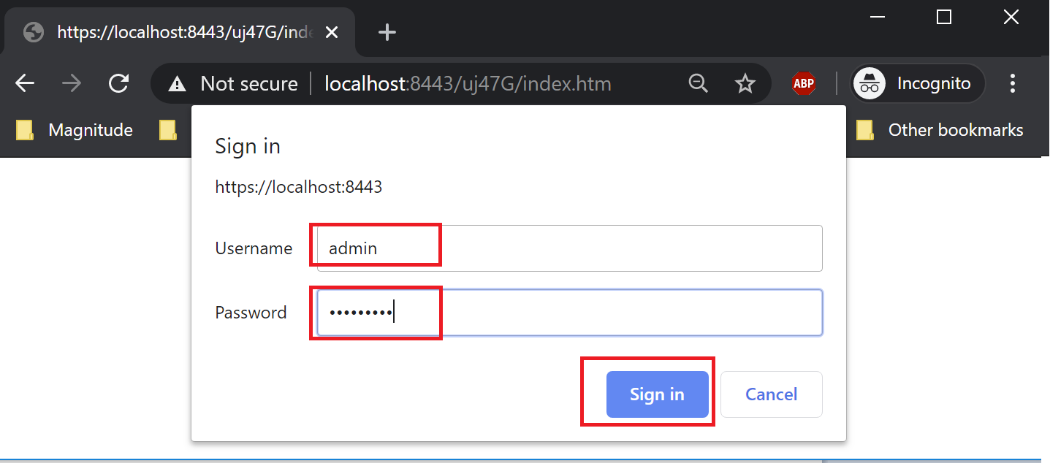
**STEP 4 | Validate scenario requirements and environment status**

1. docker build proxy -validation
   1. run ***docker images*** and validate image named *“proxy” exist*
2. docker build web -validation
   1. run ***docker images*** and validate image named *“web”* exist
3. docker-compose up -d -validation
   1. run***docker container ls*** and validate web and proxy containers exist
4. Connect to proxy’s external HTTPS listener and test paths and authentication
   1. At your browser type [***https:localhost:8443/uj47G/index.htm***](https://localhost:8443/uj47G/index.htm)
5. Connect to the same paths over HTTP instead and test redirection to TLS
   1. At your browser type [**http://localhost:8080/uj47G/index.htm**](http://localhost:8080/uj47G/index.htm)
   2. You should be redirected to the [***https:localhost:8443/uj47G/index.htm***](https://localhost:8443/uj47G/index.htm)



* 1. Click Advanced and **Proceed localhost (unsafe).** Since we have *self-signed certificate* this is expected behavior



* 1. In order to access secret content, **user** is challenged with **password**
  2. To access to the secret content type admin/Password1

