**OAuth2-Proxy (GOOGLE)**

URL: https://oauth2-proxy.github.io/oauth2-proxy/  
  
STEP 1: SETUP ON GOOGLE CLOUD

* Login into: console.cloud.google.com
* Create a new project: <https://console.developers.google.com/project>
* Choose the new project from the top right project dropdown (only if another project is selected)
* In the project Dashboard center pane, choose "APIs & Services"

Creating OAuth Screen

* Go to "OAuth consent screen", choose ’external’ type and then click on create.
* Provide information like App name, user support email, app logo, app domain, authorized domains, developer email, then click on ‘Save & Continue’
* Set the scope settings (if required)
* Add test users
* Save all the changes.

Creating Credentials

* In the center pane, choose "Credentials" tab.
* Open the "New credentials" drop down
* Choose "OAuth client ID"
* Choose "Web application"
* Application name is freeform, choose something appropriate
* Authorized JavaScript origins is your domain ex: <https://internal.yourcompany.com> (In my case: <https://authtest.saquib.publicvm.com>)
* Authorized redirect URIs is the location of oauth2/callback example: <https://internal.yourcompany.com/oauth2/callback> (In my case: <https://authtest.saquib.publicvm.com/oauth2/callback>
* Choose "Create"
* Take note of the Client ID and Client Secret

Note: It's recommended to refresh sessions on a short interval (1h) with cookie-refresh setting which validates that the account is still authorized.

STEP 2: SETUP FILES ON VIRTUAL MACHINE / LOCAL MACHINE

* Install docker and docker compose plugin  
  curl -fsSL https://get.docker.com -o get-docker.sh

sudo sh get-docker.sh

sudo apt-get install docker-compose-plugin

* Install nginx (for reverse proxy), start nginx service  
  sudo apt install nginx -y && sudo systemctl start nginx
* Install certbot (for issuing ssl certificates)  
  sudo snap install --classic certbot

sudo ln -s /snap/bin/certbot /usr/bin/certbot

* If on linux machine then, generate a random cookie secret with this command

dd if=/dev/urandom bs=32 count=1 2>/dev/null | base64 | tr -d -- '\n' | tr -- '+/' '-\_' ; echo

* If on windows machine then use this powershell command to generate a random cookie secret

# Add System.Web assembly to session, just in case

Add-Type -AssemblyName System.Web

[Convert]::ToBase64String([System.Text.Encoding]::UTF8.GetBytes([System.Web.Security.Membership]::GeneratePassword(32,4))).Replace("+","-").Replace("/","\_")

* Create a docker compose file (replace: app\_url, client\_id, client\_secret, cookie\_secret  
  nano docker-compose.yml  
  paste the below content and save the file

version: '3.8'

services:

  oauth2-proxy:

    image: quay.io/oauth2-proxy/oauth2-proxy:latest

    container\_name: oauth2\_proxy

    ports:

      - "4180:4180"

    environment:

      - OAUTH2\_PROXY\_EMAIL\_DOMAINS=\*

      - OAUTH2\_PROXY\_UPSTREAMS=http://nginx:80

      - OAUTH2\_PROXY\_REDIRECT\_URL=https://<app\_url>/oauth2/callback

      - OAUTH2\_PROXY\_CLIENT\_ID=<client\_id>

      - OAUTH2\_PROXY\_CLIENT\_SECRET=<client\_secret>

      - OAUTH2\_PROXY\_COOKIE\_SECRET=<cookie\_secret>

      - OAUTH2\_PROXY\_PROVIDER=google

      - OAUTH2\_PROXY\_HTTP\_ADDRESS=0.0.0.0:4180

    restart: always

  nginx:

    image: nginx:latest

    container\_name: nginx

    ports:

      - "8080:80"

    depends\_on:

      - oauth2-proxy

    restart: unless-stopped

* Go to /etc/nginx/sites-available & create a new file ex: ‘oauth’  
  cd /etc/nginx/sites-available && sudo nano oauth  
  paste the below content and the save the file

upstream auth\_container {

server localhost:4180;

}

server {

server\_name <app\_url>;

location / {

proxy\_pass http://auth\_container;

proxy\_set\_header Host $host;

proxy\_set\_header X-Real-IP $remote\_addr;

proxy\_set\_header X-Forwarded-For $proxy\_add\_x\_forwarded\_for;

proxy\_set\_header X-Forwarded-Proto $scheme;

client\_max\_body\_size 200M;

}

}

* Soft link sites-available with sites-enabled  
  sudo ln -s /etc/nginx/sites-available/\* /etc/nginx/sites-enabled/
* Generate ssl certificate using certbot (make sure ‘A’ type record is created on dns management portal)  
  sudo certbot --nginx --domain <app\_url> --agree-tos --no-eff-email --non-interactive --redirect --email <your\_email>@gmail.com
* Reload nginx service: sudo systemctl reload nginx
* Go to the directory where docker compose file is present  
  sudo docker compose up -d
* Wait for containers to start properly and then go to the <app\_url> to test the oauth login (google)