**DOCKER REST API**

<https://www.youtube.com/watch?v=a5td09OWFXA>

Go to /lib/systemd/system/docker.service

ExecStart=/usr/bin/dockerd -H fd:// --containerd=/run/containerd/containerd.sock -H=tcp://0.0.0.0:8080

Now we are listening on any IP 0.0.0.0 at port 8080. For API

you can check in docker status

level=info msg="API listen on /run/docker.sock"

level=info msg="API listen on [::]:8080"

Also

ps -ef | grep docker

shows where is docker process.

In postman , import the latest API from <https://docs.docker.com/engine/api/v1.41/> Then you will have the structure ready for you:

BaseURL http://10.xxx.7x.1xx:8080

List containers {{baseUrl}}/containers/json?all=false&limit=14394382&size=false

OR from remote system do:

curl [http://10.xxx.7x.1xx:8080/containers/json](http://10.164.78.179:8080/containers/json)

curl [http://10.xxx.7x.1xx:8080/images/json](http://10.164.78.179:8080/images/json)

Create a redis container

docker run -it --detach --name myredis redis // Don’t use --rm if u want to restart it

List Containers

{{baseUrl}}/containers/json?all=false&limit=14394382&size=false

You can see the volume, port, labels and status of container.

Start a container

{{baseUrl}}/containers/:id/start with params id= 99ff8f7791b9

Stop a container

{{baseUrl}}/containers/:id/stop?t=5 with params id= 99ff8f7791b9

**Secure Docker Host using TLS HTTPS**

[**https://docs.docker.com/engine/security/protect-access/**](https://docs.docker.com/engine/security/protect-access/)

openssl genrsa -aes256 -out ca-key.pem 4096 // passphrase : test

openssl req -new -x509 -days 365 -key ca-key.pem -sha256 -out ca.pem

echo subjectAltName = DNS:MACHNIEHOSTNAME,IP:10.xxx.7x.1xx,IP:127.0.0.1 >> extfile.cnf

echo extendedKeyUsage = serverAuth >> extfile.cnf

openssl genrsa -out server-key.pem 4096

openssl req -subj "/CN=MACHNIEHOSTNAME" -sha256 -new -key server-key.pem -out server.csr

openssl x509 -req -days 365 -sha256 -in server.csr -CA ca.pem -CAkey ca-key.pem -CAcreateserial -out server-cert.pem -extfile extfile.cnf

openssl genrsa -out key.pem 4096

openssl req -subj '/CN=client' -new -key key.pem -out client.csr

echo extendedKeyUsage = clientAuth > extfile-client.cnf

openssl x509 -req -days 365 -sha256 -in client.csr -CA ca.pem -CAkey ca-key.pem -CAcreateserial -out cert.pem -extfile extfile-client.cnf

Move certs to /etc/ssl/

secure them

chmod -v 0400 ca-key.pem key.pem server-key.pem

chmod -v 0444 ca.pem server-cert.pem cert.pem

next change in /lib/systemd/system/docker.service

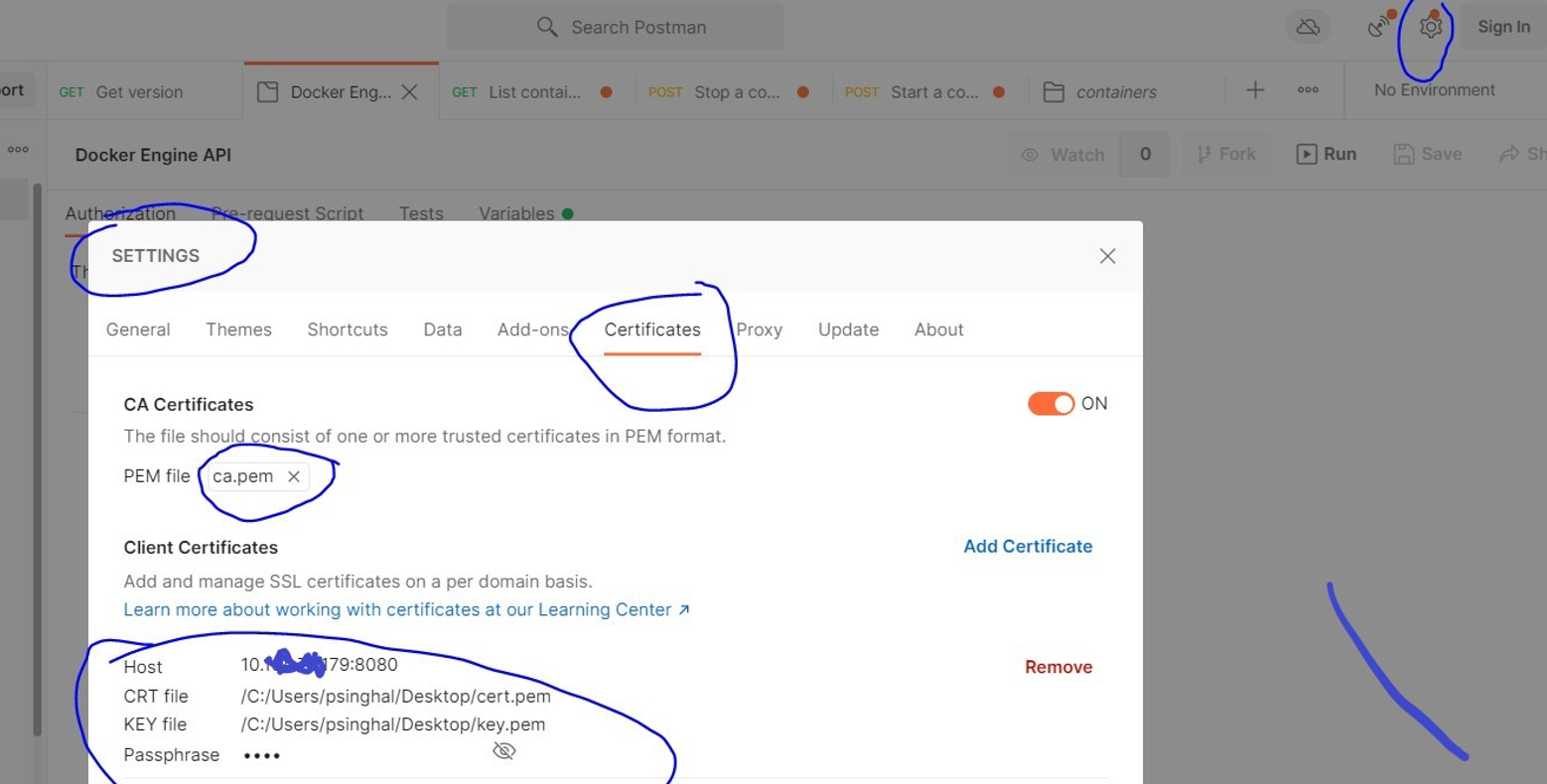
ExecStart=/usr/bin/dockerd -H fd:// --containerd=/run/containerd/containerd.sock -H=tcp://0.0.0.0:8443 --tlsverify --tlscacert=/etc/ssl/certs/ca.pem --tlscert=/etc/ssl/certs/server-cert.pem --tlskey=/etc/ssl/private/server-key.pem

From a differnet machine you can access as :

curl https://10.xxx.7x.1xx:8080/containers/json --cert cert.pem --key key.pem --cacert ca.pem

In Postman at global level you can do settings for certificates for a given HOST. So Add ca.pem to CA certificates and then client key & cert in client certificates section

Then the DockerEngine API will works as is:



You will have to give passphrase, for postman to work.

You may change the connection settings at

/usr/bin/dockerd -H fd:// --containerd=/run/containerd/containerd.sock -H=tcp://**10.xxx.7x.1xx**:8443 --tlsverify --tlscacert=/etc/ssl/certs/ca.pem …

Then the client will be assumed only at VM 10.xxx.7x.1xx. See <https://docs.docker.com/engine/install/linux-postinstall/#configure-where-the-docker-daemon-listens-for-connections>

To restrict access to docker daemon and also the use of firewalls/ufw in here.

How do I authenticate with the V2 API?

The following example script demonstrates authentication with the new V2 API.

Notes:

* jq required
* Need to set user/pass

Authentication by username, password: <https://hub.docker.com/support/doc/how-do-i-authenticate-with-the-v2-api>

#!/bin/bash

set -e

# set username and password

UNAME="<^>username<^^>"

UPASS="<^>password<^^>"

# get token to be able to talk to Docker Hub

TOKEN=$(curl -s -H "Content-Type: application/json" -X POST -d '{"username": "'${UNAME}'", "password": "'${UPASS}'"}' https://hub.docker.com/v2/users/login/ | jq -r .token)

# get list of namespaces accessible by user (not in use right now)

#NAMESPACES=$(curl -s -H "Authorization: JWT ${TOKEN}" https://hub.docker.com/v2/repositories/namespaces/ | jq -r '.namespaces|.[]')

# get list of repos for that user account

REPO\_LIST=$(curl -s -H "Authorization: JWT ${TOKEN}" https://hub.docker.com/v2/repositories/${UNAME}/?page\_size=10000 | jq -r '.results|.[]|.name')

# build a list of all images & tags

for i in ${REPO\_LIST}

do

# get tags for repo

IMAGE\_TAGS=$(curl -s -H "Authorization: JWT ${TOKEN}" https://hub.docker.com/v2/repositories/${UNAME}/${i}/tags/?page\_size=10000 | jq -r '.results|.[]|.name')

# build a list of images from tags

for j in ${IMAGE\_TAGS}

do

# add each tag to list

FULL\_IMAGE\_LIST="${FULL\_IMAGE\_LIST} ${UNAME}/${i}:${j}"

done

done

# output list of all docker images

for i in ${FULL\_IMAGE\_LIST}

do echo ${i} done

**DOCKER REST API Authorization in Docker Daemon:**

Install as in <https://github.com/casbin/casbin-authz-plugin>

Install GO if not there,

$ apt install golang-go # install go language

$ mkdir /usr/local/go

$ export GOPATH=/usr/local/go

$ go get github.com/casbin/casbin-authz-plugin

$ cd $GOPATH/src/github.com/casbin/casbin-authz-plugin

$ make

$ sudo make install

Run anywhere …

mkdir /var/lib/docker/plugins/casbin/

$ Copy casbin-authz-plugin in /var/lib/docker/plugins/casbin/

$ mkdir examples

$ cp basic\_model.conf examples/.

$ cp basic\_policy.csv examples/.

$ ./casbin-authz-plugin

Working Example:

/var/lib/docker/plugins/casbin/casbin-authz-plugin# ls

casbin-authz-plugin examples go.mod go.sum LICENSE main.go Makefile plugin README.md systemd

vi /lib/systemd/system/casbin-authz-plugin.service 🡪

[Unit]

Description=Docker RBAC & ABAC Authorization Plugin based on Casbin

Before=docker.service

After=network.target casbin-authz-plugin.socket

Requires=casbin-authz-plugin.socket docker.service

[Service]

ExecStart=/var/lib/docker/plugins/casbin/casbin-authz-plugin/casbin-authz-plugin

WorkingDirectory=/var/lib/docker/plugins/casbin/casbin-authz-plugin

$ systemctl daemon-reload

$ systemctl enable casbin-authz-plugin

$ systemctl start casbin-authz-plugin

sudo systemctl edit --full docker.service

$ Edit vi /etc/systemd/system/docker.service to add details :

ExecStart=/usr/bin/dockerd -H fd:// **--authorization-plugin=casbin-authz-plugin** --containerd=/run/containerd/containerd.sock -H=tcp://0.0.0.0:8443 --tlsverify --tlscacert=/etc/ssl/certs/ca.pem --tlscert=/etc/ssl/certs/server-cert.pem --tlskey=/etc/ssl/private/server-key.pem

systemctl restart casbin-authz-plugin

*systemctl restart docker*

See logs for *casbin* at:

journalctl -xe -u casbin-authz-plugin -f

See <https://casbin.org/editor/> and change your policy accordingly.

& API support <https://casbin.org/docs/en/management-api>

Authentication Flow: <https://docs.docker.com/engine/extend/plugins_authorization/>

Generate a Random Key:

openssl rand -base64 32

OR

head -c32 /dev/urandom | base64