docker version

PS C:\Workspace> docker version

Client: Docker Engine - Community

Version: 19.03.5

API version: 1.40

Go version: go1.12.12

Git commit: 633a0ea

Built: Wed Nov 13 07:22:37 2019

OS/Arch: windows/amd64

Experimental: false

Server: Docker Engine - Community

Engine:

Version: 19.03.5

API version: 1.40 (minimum version 1.24)

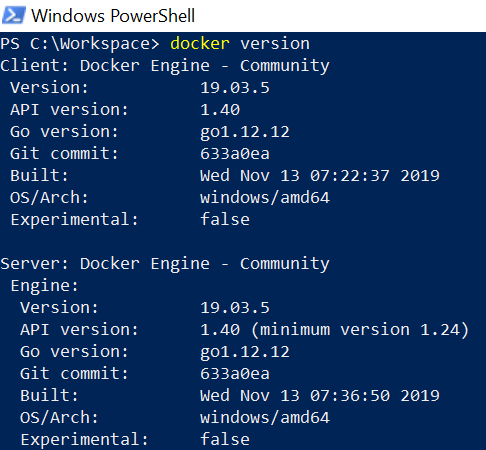
Go version: go1.12.12

Git commit: 633a0ea

Built: Wed Nov 13 07:36:50 2019

OS/Arch: windows/amd64

Experimental: false



docker info

PS C:\Workspace> docker info

Client:

Debug Mode: false

Server:

Containers: 1

Running: 0

Paused: 0

Stopped: 1

Images: 1

Server Version: 19.03.5

Storage Driver: windowsfilter

Windows:

Logging Driver: json-file

Plugins:

Volume: local

Network: ics internal l2bridge l2tunnel nat null overlay private transparent

Log: awslogs etwlogs fluentd gcplogs gelf json-file local logentries splunk syslog

Swarm: inactive

Default Isolation: hyperv

Kernel Version: 10.0 17763 (17763.1.amd64fre.rs5\_release.180914-1434)

Operating System: Windows 10 Enterprise Version 1809 (OS Build 17763.914)

OSType: windows

Architecture: x86\_64

CPUs: 8

Total Memory: 7.838GiB

Name: LTIN136071

ID: JMJ6:IYUM:64KA:WLAA:B542:GFA7:BRZ7:EHEG:JFBD:OJBZ:Q3RV:TFAX

Docker Root Dir: C:\ProgramData\Docker

Debug Mode: true

File Descriptors: -1

Goroutines: 28

System Time: 2020-01-25T13:15:50.5144933+05:30

EventsListeners: 1

Registry: https://index.docker.io/v1/

Labels:

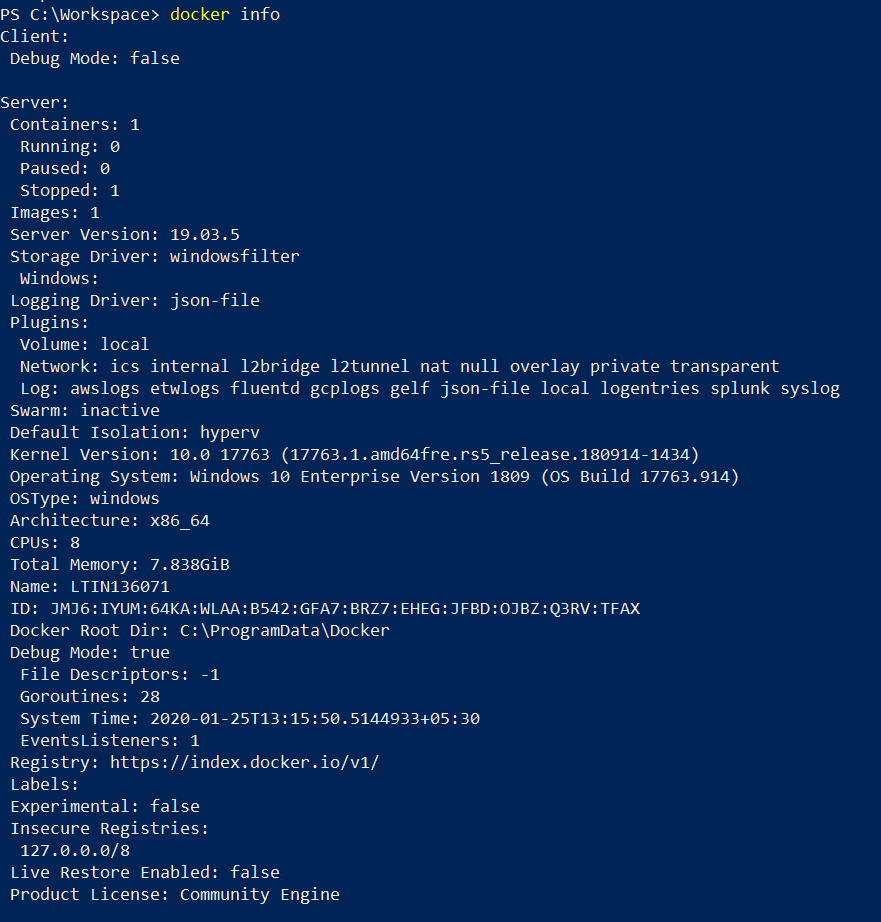
Experimental: false

Insecure Registries:

127.0.0.0/8

Live Restore Enabled: false

Product License: Community Engine



List of docker commands:

Type docker on shell prompt: docker

PS C:\Workspace> docker

Usage: docker [OPTIONS] COMMAND

A self-sufficient runtime for containers

Options:

--config string Location of client config files (default

"C:\\Users\\484260\\.docker")

-c, --context string Name of the context to use to connect to the

daemon (overrides DOCKER\_HOST env var and

default context set with "docker context use")

-D, --debug Enable debug mode

-H, --host list Daemon socket(s) to connect to

-l, --log-level string Set the logging level

("debug"|"info"|"warn"|"error"|"fatal")

(default "info")

--tls Use TLS; implied by --tlsverify

--tlscacert string Trust certs signed only by this CA (default

"C:\\Users\\484260\\.docker\\ca.pem")

--tlscert string Path to TLS certificate file (default

"C:\\Users\\484260\\.docker\\cert.pem")

--tlskey string Path to TLS key file (default

"C:\\Users\\484260\\.docker\\key.pem")

--tlsverify Use TLS and verify the remote

-v, --version Print version information and quit

Management Commands:

builder Manage builds

config Manage Docker configs

container Manage containers

context Manage contexts

image Manage images

network Manage networks

node Manage Swarm nodes

plugin Manage plugins

secret Manage Docker secrets

service Manage services

stack Manage Docker stacks

swarm Manage Swarm

system Manage Docker

trust Manage trust on Docker images

volume Manage volumes

Commands:

attach Attach local standard input, output, and error streams to a running container

build Build an image from a Dockerfile

commit Create a new image from a container's changes

cp Copy files/folders between a container and the local filesystem

create Create a new container

diff Inspect changes to files or directories on a container's filesystem

events Get real time events from the server

exec Run a command in a running container

export Export a container's filesystem as a tar archive

history Show the history of an image

images List images

import Import the contents from a tarball to create a filesystem image

info Display system-wide information

inspect Return low-level information on Docker objects

kill Kill one or more running containers

load Load an image from a tar archive or STDIN

login Log in to a Docker registry

logout Log out from a Docker registry

logs Fetch the logs of a container

pause Pause all processes within one or more containers

port List port mappings or a specific mapping for the container

ps List containers

pull Pull an image or a repository from a registry

push Push an image or a repository to a registry

rename Rename a container

restart Restart one or more containers

rm Remove one or more containers

rmi Remove one or more images

run Run a command in a new container

save Save one or more images to a tar archive (streamed to STDOUT by default)

search Search the Docker Hub for images

start Start one or more stopped containers

stats Display a live stream of container(s) resource usage statistics

stop Stop one or more running containers

tag Create a tag TARGET\_IMAGE that refers to SOURCE\_IMAGE

top Display the running processes of a container

unpause Unpause all processes within one or more containers

update Update configuration of one or more containers

version Show the Docker version information

wait Block until one or more containers stop, then print their exit codes

Run 'docker COMMAND --help' for more information on a command.

