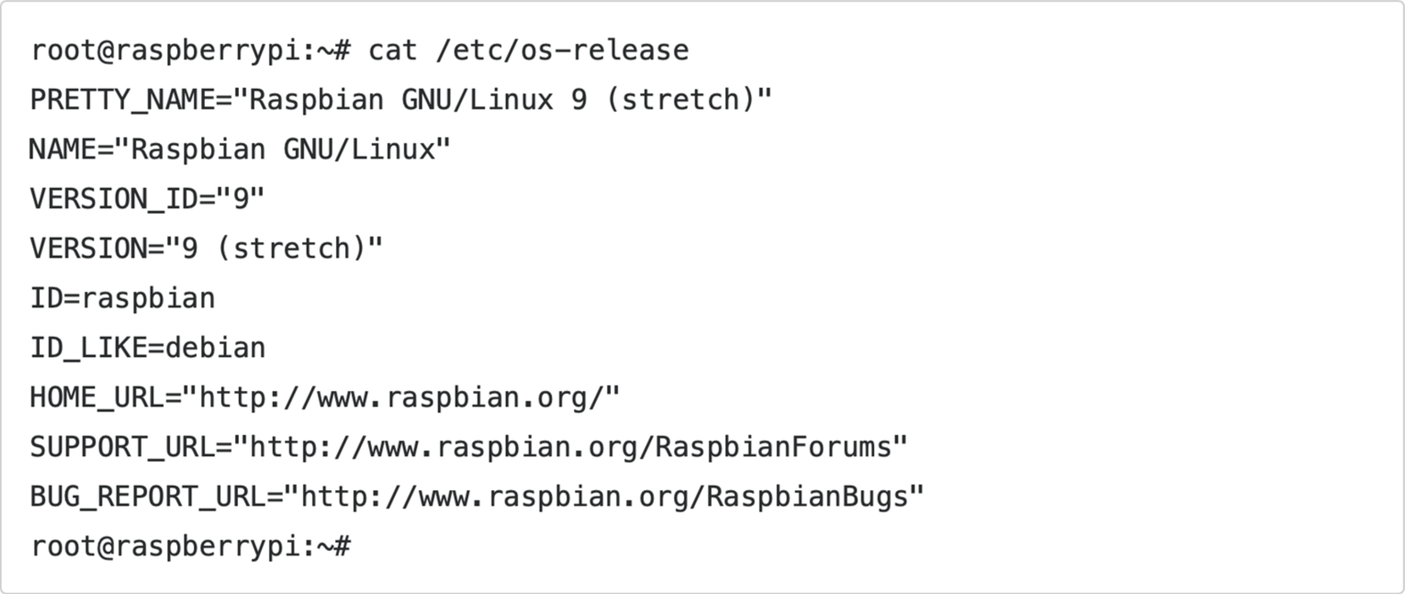
Low-cost CCTV Camera - Hands-on Lab

Verifying Raspbian OS Version



```
root@raspberrypi:~# cat /etc/os-release
PRETTY_NAME="Raspbian GNU/Linux 9 (stretch)"
NAME="Raspbian GNU/Linux"
VERSION_ID="9"
VERSION="9 (stretch)"
ID=raspbian
ID_LIKE=debian
HOME_URL="http://www.raspbian.org/"
SUPPORT_URL="http://www.raspbian.org/RaspbianForums"
BUG_REPORT_URL="http://www.raspbian.org/RaspbianBugs"
root@raspberrypi:~#
```

Low-cost CCTV Camera - Hands-on Lab

Installing Docker 18.09

```
root@raspberrypi:~# curl -sSL https://get.docker.com/ | sh
# Executing docker install script, commit: 40b1b76
+ sh -c apt-get update -qq >/dev/null
+ sh -c apt-get install -y -qq apt-transport-https ca-certificates curl >/dev/null
+ sh -c curl -fsSL "https://download.docker.com/linux/raspbian/gpg" | apt-key add -qq - >/dev/null
Warning: apt-key output should not be parsed (stdout is not a terminal)
+ sh -c echo "deb [arch=armhf] https://download.docker.com/linux/raspbian stretch edge" >
/etc/apt/sources.list.d/docker.list
+ sh -c apt-get update -qq >/dev/null
+ sh -c apt-get install -y -qq --no-install-recommends docker-ce >/dev/null
+ sh -c docker version
Client:
 Version:           18.09.0
 API version:       1.39
 Go version:        go1.10.4
 Git commit:        4d60db4
 Built:             Wed Nov  7 00:57:21 2018
 OS/Arch:           linux/arm
 Experimental:      false
```

Low-cost CCTV Camera - Hands-on Lab

Installing Docker 18.09

```
Server: Docker Engine - Community
 Engine:
  Version:          18.09.0
  API version:      1.39 (minimum version 1.12)
  Go version:       go1.10.4
  Git commit:       4d60db4
  Built:            Wed Nov  7 00:17:57 2018
  OS/Arch:          linux/arm
  Experimental:     false
If you would like to use Docker as a non-root user, you should now consider
adding your user to the "docker" group with something like:

  sudo usermod -aG docker your-user

Remember that you will have to log out and back in for this to take effect!

WARNING: Adding a user to the "docker" group will grant the ability to run
         containers which can be used to obtain root privileges on the
         docker host.
         Refer to https://docs.docker.com/engine/security/security/#docker-daemon-attack-surface
         for more information.
```

Low-cost CCTV Camera - Hands-on Lab

Installing Docker 18.09

```
** DOCKER ENGINE - ENTERPRISE **

If you ’re ready for production workloads, Docker Engine - Enterprise also includes:

  * SLA-backed technical support
  * Extended lifecycle maintenance policy for patches and hotfixes
  * Access to certified ecosystem content

** Learn more at https://dockr.ly/engine2 **

ACTIVATE your own engine to Docker Engine - Enterprise using:

  sudo docker engine activate
```

Low-cost CCTV Camera - Hands-on Lab

Verifying Docker Version

```
root@raspberrypi:~# docker version
Client:
 Version:           18.09.0
 API version:       1.39
 Go version:        go1.10.4
 Git commit:        4d60db4
 Built:             Wed Nov  7 00:57:21 2018
 OS/Arch:           linux/arm
 Experimental:      false

Server: Docker Engine - Community
 Engine:
  Version:          18.09.0
  API version:      1.39 (minimum version 1.12)
  Go version:       go1.10.4
  Git commit:       4d60db4
  Built:            Wed Nov  7 00:17:57 2018
  OS/Arch:          linux/arm
  Experimental:     false
root@raspberrypi:~#
```

Low-cost CCTV Camera - Hands-on Lab

Test Drive Nginx App on Pi Box

```
root@raspberrypi:~# docker run -d -p 80:80 nginx
Unable to find image 'nginx:latest' locally
latest: Pulling from library/nginx
9c38b5a8a4d5: Pull complete
1c9b1b3e1e0d: Pull complete
258951b5612f: Pull complete
Digest:
sha256:dd2d0ac3fff2f007d99e033b64854be0941e19a2ad51f174d9240dda20d9f534
Status: Downloaded newer image for nginx:latest
d812bf50d136b0f78353f0a0c763b6b08ecc5e7ce706bac8bd660cdd723e0fcd
root@raspberrypi:~#
```

Low-cost CCTV Camera - Hands-on Lab

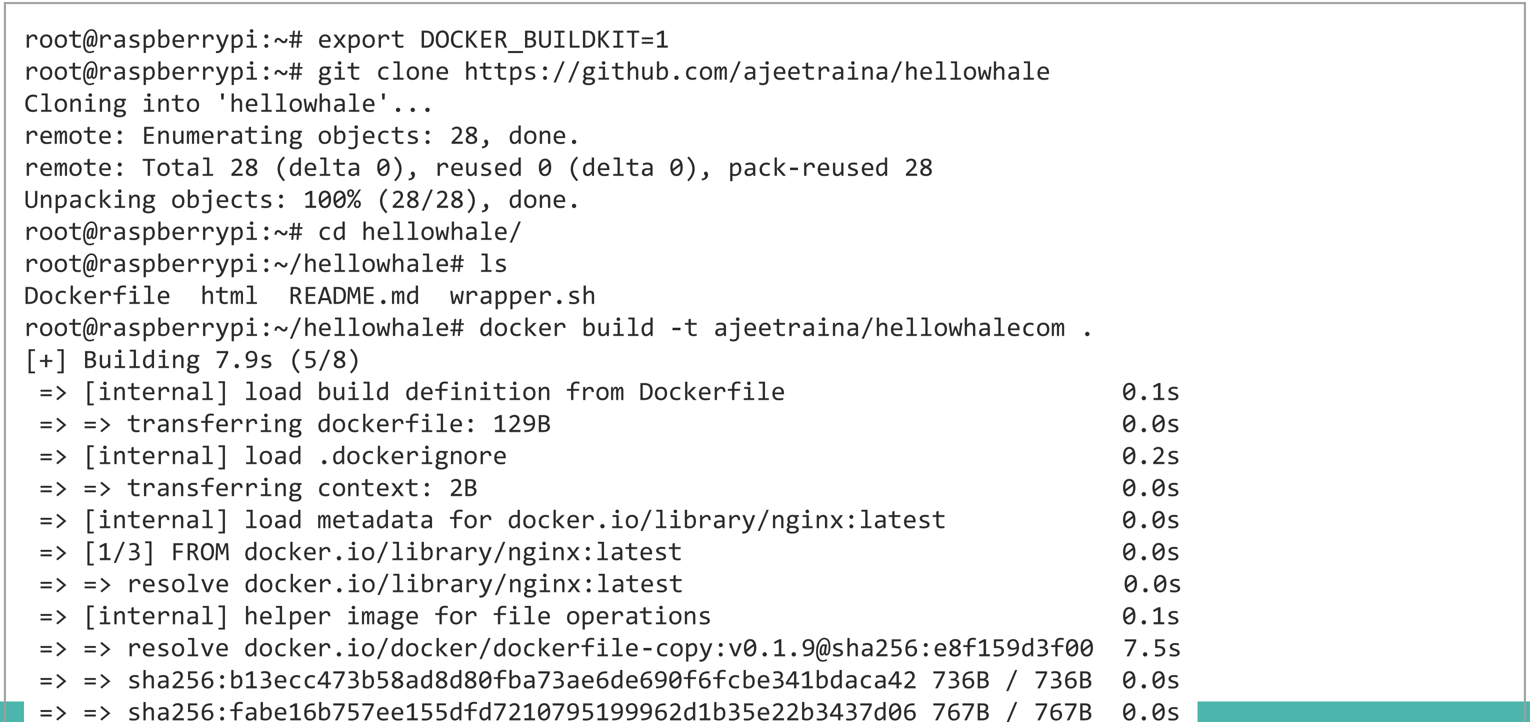
Test Drive Nginx App on Pi Box

```
root@raspberrypi:~# curl localhost:80
<!DOCTYPE html>
<html>
<head>
<title>Welcome to nginx!</title>
<style>
    body {
        width: 35em;
        margin: 0 auto;
        font-family: Tahoma, Verdana, Arial, sans-serif;
    }
</style>
</head>
<body>
<h1>Welcome to nginx!</h1>
<p>If you see this page, the nginx web server is successfully installed and
working. Further configuration is required.</p>

<p>For online documentation and support please refer to
<a href="http://nginx.org/">nginx.org</a>.<br/>
Commercial support is available at
<a href="http://nginx.com/">nginx.com</a>.</p>
```

Low-cost CCTV Camera - Hands-on Lab

BuildKit on Raspberry Pi



```
root@raspberrypi:~# export DOCKER_BUILDKIT=1
root@raspberrypi:~# git clone https://github.com/ajeetraina/hellowhale
Cloning into 'hellowhale'...
remote: Enumerating objects: 28, done.
remote: Total 28 (delta 0), reused 0 (delta 0), pack-reused 28
Unpacking objects: 100% (28/28), done.
root@raspberrypi:~# cd hellowhale/
root@raspberrypi:~/hellowhale# ls
Dockerfile  html  README.md  wrapper.sh
root@raspberrypi:~/hellowhale# docker build -t ajeetraina/hellowhalecom .
[+] Building 7.9s (5/8)
 => [internal] load build definition from Dockerfile                      0.1s
 => => transferring dockerfile: 129B                                      0.0s
 => [internal] load .dockerignore                                         0.2s
 => => transferring context: 2B                                           0.0s
 => [internal] load metadata for docker.io/library/nginx:latest           0.0s
 => [1/3] FROM docker.io/library/nginx:latest                             0.0s
 => => resolve docker.io/library/nginx:latest                             0.0s
 => [internal] helper image for file operations                           0.1s
 => => resolve docker.io/docker/dockerfile-copy:v0.1.9@sha256:e8f159d3f00  7.5s
 => => sha256:b13ecc473b58ad8d80fba73ae6de690f6fcbe341bdaca42 736B / 736B  0.0s
 => => sha256:fabe16b757ee155dfd7210795199962d1b35e22b3437d06 767B / 767B  0.0s
```

Low-cost CCTV Camera - Hands-on Lab

BuildKit on Raspberry Pi

```
root@raspberrypi:~/hellowhale# time docker build -t ajeetraina/hellowhale .
[+] Building 0.4s (9/9) FINISHED
 => [internal] load build definition from Dockerfile                       0.1s
 => => transferring dockerfile: 31B                                        0.0s
 => [internal] load .dockerignore                                          0.1s
 => => transferring context: 2B                                            0.0s
 => [internal] load metadata for docker.io/library/nginx:latest            0.0s
 => [internal] helper image for file operations                            0.0s
 => [1/3] FROM docker.io/library/nginx:latest                              0.0s
 => [internal] load build context                                          0.0s
 => => transferring context: 317B                                          0.0s
 => CACHED [2/3] COPY wrapper.sh /                                         0.0s
 => CACHED [3/3] COPY html /usr/share/nginx/html                           0.0s
 => exporting to image                                                     0.1s
 => => exporting layers                                                    0.0s
 => => writing image sha256:5aee990f7e24e7c0f486ed01b4c1f8696ff307f836af1  0.0s
 => => naming to docker.io/ajeetraina/hellowhale                           0.0s
real         0m0.615s
user         0m0.204s
sys          0m0.082s
```

Low-cost CCTV Camera - Hands-on Lab

Verifying Dockerd

```
root@raspberrypi:~/hellowhale# systemctl status docker
◖ docker.service - Docker Application Container Engine
   Loaded: loaded (/lib/systemd/system/docker.service; enabled; vendor preset: e
   Active: active (running) since Tue 2019-02-26 13:01:04 IST; 38min ago
     Docs: https://docs.docker.com
 Main PID: 2437 (dockerd)
      CPU: 1min 46.174s
   CGroup: /system.slice/docker.service
           ├─2437  /usr/b -H unix:// kerd
           ├─2705  /usr/ -proxy -proto tcp -host-ip 0.0.0.0 -host-port 8
           └─4186  /usr/ -proxy -proto tcp -host-ip 0.0.0.0 -host-port 8
```

Low-cost CCTV Camera - Hands-on Lab

Verifying if armv7 hello-world image is available or not

```
docker run --rm mplatform/mquery hello-world
Unable to find image 'mplatform/mquery:latest' locally
latest: Pulling from mplatform/mquery
db6020507de3: Pull complete
5107afd39b7f: Pull complete
Digest: sha256:e15189e3d6fbcee8a6ad2ef04c1ec80420ab0fdcf0d70408c0e914af80dfb107
Status: Downloaded newer image for mplatform/mquery:latest
Image: hello-world
 * Manifest List: Yes
 * Supported platforms:
   - linux/amd64
   - linux/arm/v5
   - linux/arm/v7
   - linux/arm64
   - linux/386
   - linux/ppc64le
   - linux/s390x
   - windows/amd64:10.0.14393.2551
   - windows/amd64:10.0.16299.846
   - windows/amd64:10.0.17134.469
   - windows/amd64:10.0.17763.194
```

Low-cost CCTV Camera - Hands-on Lab

Verifying hellowhale Image

```
root@raspberrypi:~# docker run --rm mplatform/mquery ajeetraina/hellowhale
Image: ajeetraina/hellowhale
 * Manifest List: No
 * Supports: amd64/linux
```

Low-cost CCTV Camera - Hands-on Lab

Verifying Prometheus Image

```
root@raspberrypi:~# docker run --rm mplatform/mquery rycus86/prometheus
Image: rycus86/prometheus
 * Manifest List: Yes
 * Supported platforms:
   - linux/amd64
   - linux/arm/v7
   - linux/arm64
```

Low-cost CCTV Camera - Hands-on Lab

Running Low-cost HD surveillance Camera system using Docker Container -

Cloning the Repository:

```
$ git clone https://github.com/collabnix/docker-cctv-raspbian
$ cd docker-cctv-raspbian
```

Low-cost CCTV Camera - Hands-on Lab

Running Low-cost HD surveillance Camera system using Docker Container -

Building the Docker Image:

```
docker build -t collabnix/docker-cctv-raspbian .
```

Low-cost CCTV Camera - Hands-on Lab

Running Low-cost HD surveillance Camera system using Docker Container -

Configuring Camera Interface:

```
Before you execute run.sh, you need to configure Camera Interface by running
the below command:

# raspi-config

It will open up command-line UI window, choose Interfacing , select Camera
and enable it. Save and exit the CLI window.
```

Low-cost CCTV Camera - Hands-on Lab

Running Low-cost HD surveillance Camera system using Docker Container -

Running the Docker container:

```
root@raspberrypi:~/rpi-motion# sudo modprobe bcm2835-v4l2
root@raspberrypi:~/rpi-motion# ls

$sh run.sh
```

Low-cost CCTV Camera - Hands-on Lab

That’s it. Browse over to http://192.168.1.5:8082(either using Win Laptop or macbook) to open up CCTV cam which initiates the video streaming instantly. Cool, isn’t it?