Weidmüller

UC20-SL2000-OLAC-EC

2638920000 and others

Quick Start Guide for docker on u-control studio

Instructions for the installation of docker environment and portainer …

QSG0022v04-UC20

Revision history

Version	Date	Change log	Author
01	2020-03	First released version	w010521
02	2021-03	complete revision - added information for V4.x / Debian 10	w011498
03	2021-03	review	w010174
04	2021-06	Separate chapters for different Debian versions information for time synchronisation updated	w011498

Contact

Weidmüller Interface GmbH & Co. KG
Klingenbergstraße 26
32758 Detmold, Germany
T +49 5231 14-0
F +49 5231 14-292083
info@weidmueller.com
www.weidmueller.com

For further support please contact your local sales representative.

Template: V2.1 – 2020-01

Content

1. Warning and disclaimer

Warning

Controls may fail in unsafe operating conditions, causing uncontrolled operation of the controlled devices. Such hazardous events can result in death and / or serious injury and / or property damage. Therefore, there must be provide safety equipment/ electrical safety design or other redundant safety features that are independent from the automation system.

Disclaimer

This Example / Application Note does not relieve you of the obligation to handle it safely during use, installation, operation and maintenance. Each user is responsible for the correct operation of his control system.

By using this program example / application note prepared by Weidmüller, you accept that Weidmüller cannot be held liable for any damage to property and / or personal injury that may occur because of the use.

Note

The application examples do not represent customer-specific solutions, they are simply intended to help for typical tasks. The user is responsible for the proper operation of the described products. This application example does not relieve you of the obligation of safe use, installation, operation and maintenance. Application examples are not binding and do not claim to be complete in terms of configuration as well as any contingencies.

By using this Application Example, you acknowledge that we cannot be held liable for any damages beyond the described liability regime. We reserve the right to make changes to this sample application at any time without notice.

In case of discrepancies between the proposals in the application example and other Weidmüller publications, like manuals, such contents always have more priority to the examples.

We assume no liability for the information contained in this document. Our liability, for whatever legal reason, for damages caused by the use of the examples, instructions, programs, project planning and performance data, etc. described in this application example is excluded.

Security notes

In order to protect equipment, systems, machines and networks against cyber threats, it is necessary to implement (and maintain) a complete state-of-the-art industrial security concept. The customer is responsible for preventing unauthorized access to his equipment, systems, machines and networks. Systems, machines and components should only be connected to the corporate network or the Internet if necessary and appropriate safeguards (such as firewalls and network segmentation) have been taken.

2. Abstract

This guide contains instructions on how to install Docker on u-control studio.

3. Requirements

1. Engineering software u-create studio >= V1.20.2
2. Weidmueller UC20-SL2000-OLAC (or similar) with “open linux”
3. The PLC must have internet access (configuration in chapter 4.1)

4. Workflow

4.1. Controller (basic configuration / DevAdmin)

1. If not done yet, follow the general U-Create Studio Quick Start Guide in order to use u-create studio for creating a new SD card target. This target must contain your public key for ssh access: See chapter “4 Adding a personal SSH key for u-create studio installation” in the U-Create Studio Quick Start Guide.

   Note: In case of u-create studio V4.2, check chapter 4.2.2 before creating the target.

2. Update the PLC with the new target.

3. Open the PLC’s u-control webserver (DevAdmin) via the IP address you assigned during target creation and login.

   The default settings are:

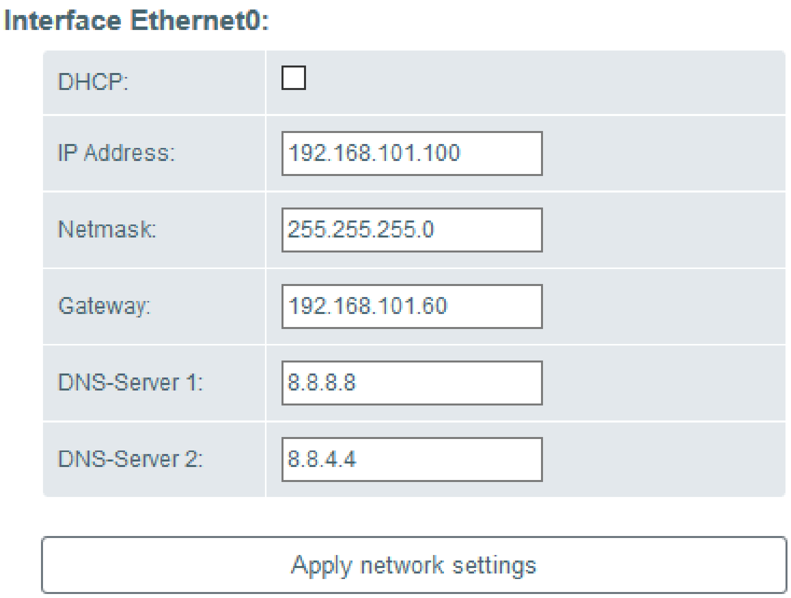
IP	: 192.168.101.100
Username	: Administrator
Password	: tobechanged

4. Change to tab “Config” and change values to your needs…

   - add the correct gateway address
   - add at least one DNS Server (Default: Google DNS servers)
   - set the correct date and time
   - set the correct time zone

   Don’t forget to apply new settings!

   Note: It is recommended to attach a buffer battery for the RTC.



Interface Ethernet0:

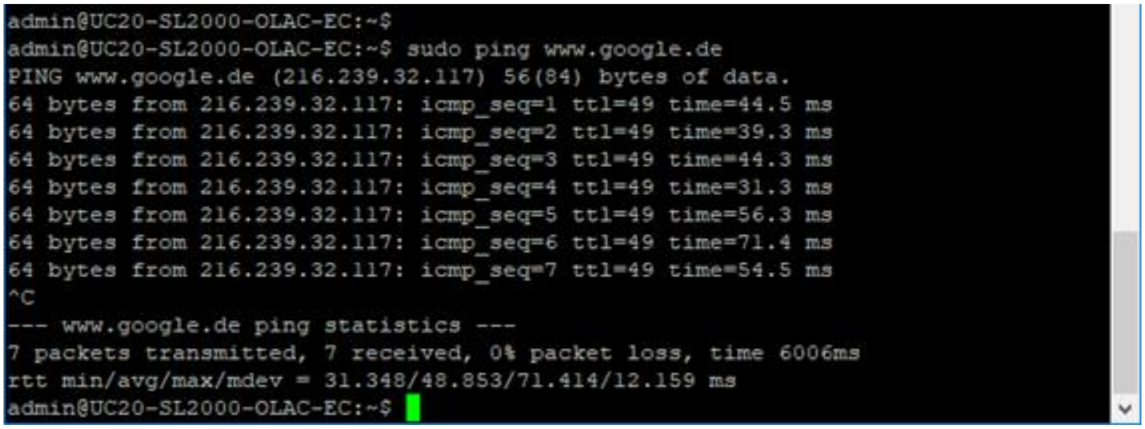
DHCP:	□
IP Address:	192.168.101.100
Netmask:	255.255.255.0
Gateway:	192.168.101.60
DNS-Server 1:	8.8.8.8
DNS-Server 2:	8.8.4.4

4.2. Controller (advanced configuration / ssh)

1. Login to your PLC via ssh (we suggest PuTTY).

2. Check whether you have a functional internet connection to your PLC:

    I. Enter “sudo ping www.google.de” in the shell
    II. Compare the PLC’s output with the picture below.



```
admin@UC20-SL2000-OLAC-EC:~$
admin@UC20-SL2000-OLAC-EC:~$ sudo ping www.google.de
PING www.google.de (216.239.32.117) 56(84) bytes of data.
64 bytes from 216.239.32.117: icmp_seq=1 ttl=49 time=44.5 ms
64 bytes from 216.239.32.117: icmp_seq=2 ttl=49 time=39.3 ms
64 bytes from 216.239.32.117: icmp_seq=3 ttl=49 time=44.3 ms
64 bytes from 216.239.32.117: icmp_seq=4 ttl=49 time=31.3 ms
64 bytes from 216.239.32.117: icmp_seq=5 ttl=49 time=56.3 ms
64 bytes from 216.239.32.117: icmp_seq=6 ttl=49 time=71.4 ms
64 bytes from 216.239.32.117: icmp_seq=7 ttl=49 time=54.5 ms
^C
--- www.google.de ping statistics ---
7 packets transmitted, 7 received, 0% packet loss, time 6006ms
rtt min/avg/max/mdev = 31.348/48.853/71.414/12.159 ms
admin@UC20-SL2000-OLAC-EC:~$
```

4.2.1. U-create studio V1.20.2 (Debian 8)

1. add the Debian repository to the system

```
echo "deb http://deb.debian.org/debian/ jessie main contrib non-free" | sudo
tee \/etc/apt/sources.list.d/debian.list
```

   This will create a new file called debian.list with an entry for the corresponding Debian repository in /etc/apt/sources.list.d/

2. update the package list

```
sudo apt-get update
```

3. Install additional packages

```
sudo apt-get install --no-upgrade \
apt-transport-https \
ca-certificates \
curl \
gnupg2 \
rdate -y
```

4. time synchronization via ‘NTP’

   It is strongly recommended NOT to use any NTP client/server daemon due to negative influences on the realtime behavior.

```
sudo timedatectl set-ntp 0
```

   If you are not able to use a buffer battery for the RTC (real time clock), you can use a workaround in order to adjust date & time after every reboot of the controller.

   You need to specify a time server by hostname or IP.

```
example 1:
sudo echo "@reboot root /usr/bin/rdate -ns 192.168.0.1" >> /etc/crontab

example 2:
sudo echo "@reboot root /usr/bin/rdate -ns time.google.com" >>
/etc/crontab
```

5. add the docker repository

```
curl -fsSL https://download.docker.com/linux/debian/gpg | sudo apt-key add -

echo "deb [arch=armhf] https://download.docker.com/linux/debian jessie stable" |
sudo tee \/etc/apt/sources.list.d/docker.list
```

4.2.2. U-create studio V4.2 (Debian 10)

1. add the Debian repository to the system

```
echo "deb http://deb.debian.org/debian/ buster main contrib non-free" | sudo
tee \/etc/apt/sources.list.d/debian.list
```

   This will create a new file called debian.list with an entry for the corresponding Debian repository in /etc/apt/sources.list.d/

2. update the package list

```
sudo apt-get update
```

3. Install additional packages

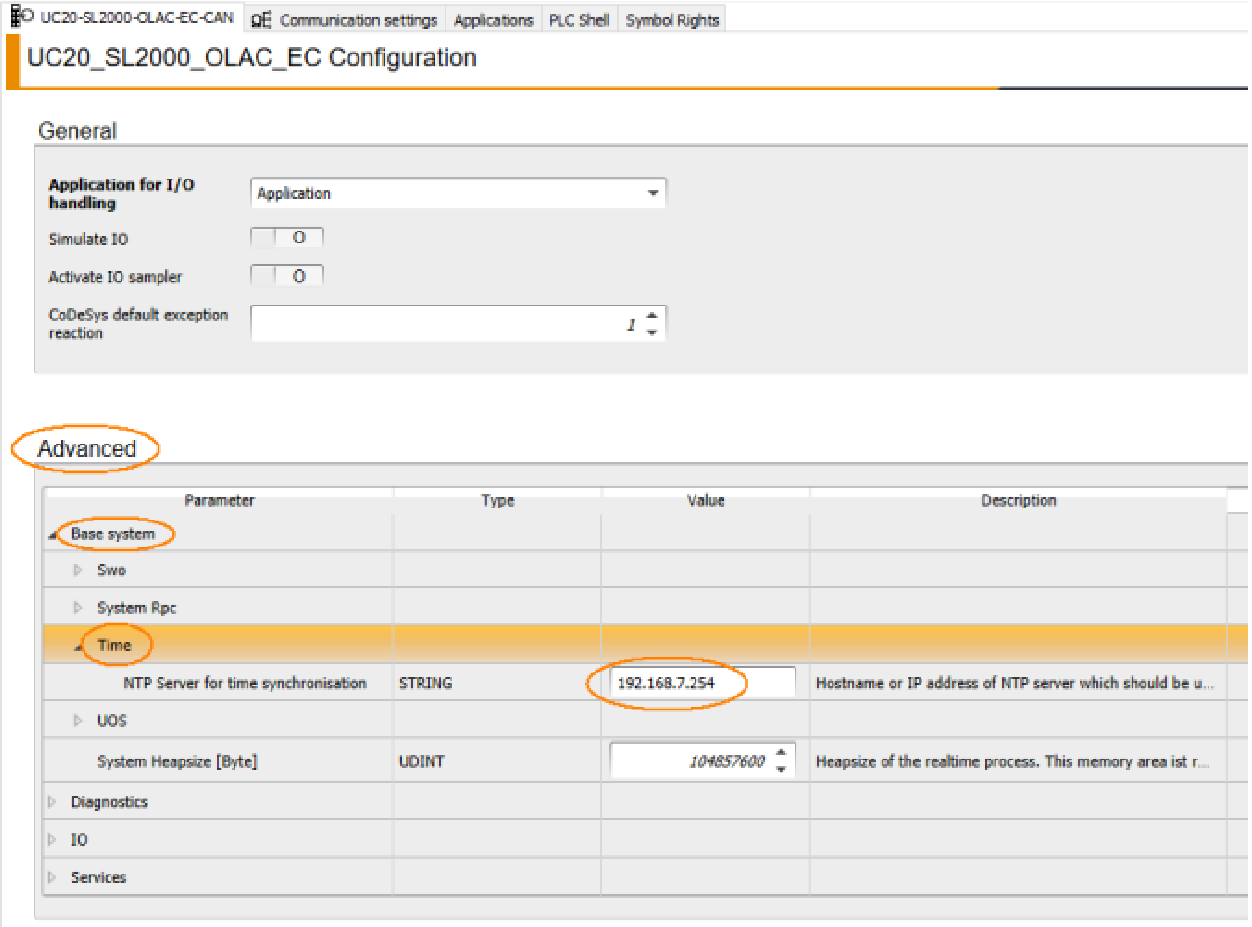
```
sudo apt-get install --no-upgrade \
apt-transport-https \
ca-certificates \
curl \
gnupg2 -y
```

4. add the docker repository

```
curl -fsSL https://download.docker.com/linux/debian/gpg | sudo apt-key add -

echo "deb [arch=armhf] https://download.docker.com/linux/debian buster stable" |
sudo tee \/etc/apt/sources.list.d/docker.list
```

5. time synchronization via ‘NTP’

   In u-create Studio 4.x, you can configure the NTP server in the engineering software. This has to be done before the SD card target is created.



UC20-SL2000-OLAC-EC-CAN | Communication settings | Applications | PLC Shell | Symbol Rights

UC20_SL2000_OLAC_EC Configuration

General

Application for I/O handling	Application
Simulate IO	O
Activate IO sampler	O
CoDeSys default exception reaction	1

Advanced

Parameter	Type	Value	Description
Base system			
Swo			
System Rpc			
Time			
NTP Server for time synchronisation	STRING	192.168.7.254	Hostname or IP address of NTP server which should be u...
UOS			
System Heapsize [Byte]	UDINT	104857600	Heapsize of the realtime process. This memory area ist r...
Diagnostics			
IO			
Services			

4.3. Docker installation (community edition)

1. Install Docker-ce:

```
sudo apt-get update
sudo apt-get install docker-ce -y
```

2. To avoid the additional “sudo” in front of every docker command, we add our user admin to the new group docker:

```
sudo usermod -aG docker admin
```

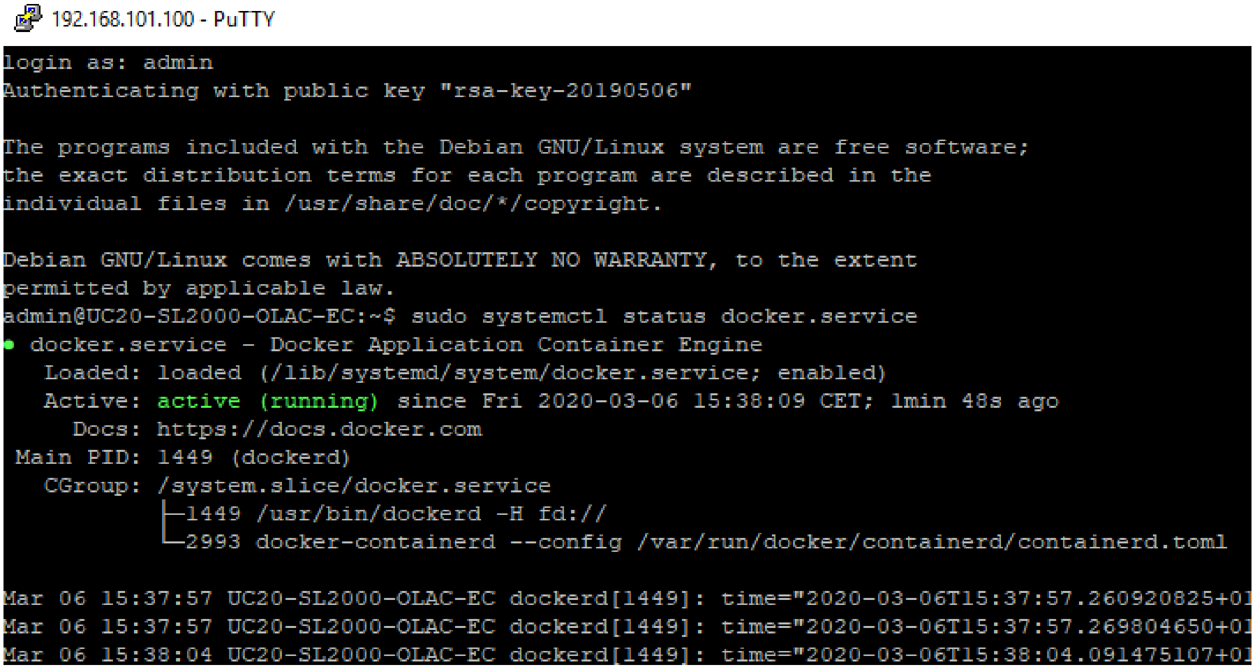
3. Reboot the PLC:

```
sudo reboot
```

4. Check the proper installation of Docker:

```
sudo systemctl status docker.service
```

this should show something similar to the picture below:



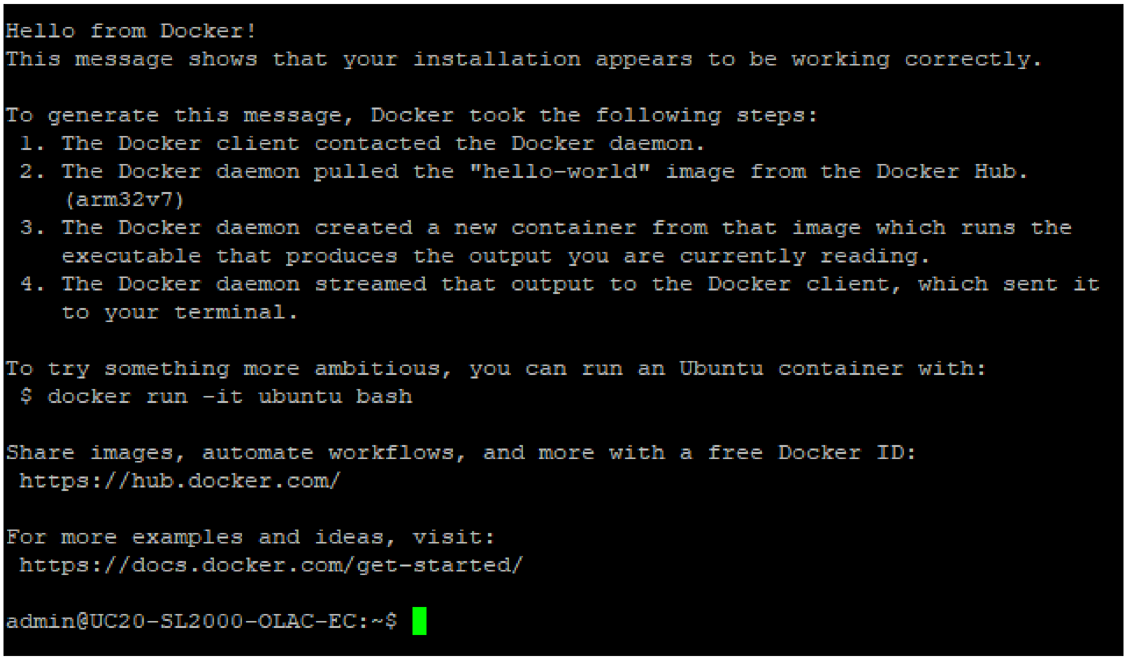
```
login as: admin
Authenticating with public key "rsa-key-20190506"

The programs included with the Debian GNU/Linux system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
admin@UC20-SL2000-OLAC-EC:~$ sudo systemctl status docker.service
● docker.service - Docker Application Container Engine
   Loaded: loaded (/lib/systemd/system/docker.service; enabled)
   Active: active (running) since Fri 2020-03-06 15:38:09 CET; 1min 48s ago
     Docs: https://docs.docker.com
 Main PID: 1449 (dockerd)
   CGroup: /system.slice/docker.service
           ├─1449 /usr/bin/dockerd -H fd://
           └─2993 docker-containerd --config /var/run/docker/containerd/containerd.toml

Mar 06 15:37:57 UC20-SL2000-OLAC-EC dockerd[1449]: time="2020-03-06T15:37:57.260920825+01
Mar 06 15:37:57 UC20-SL2000-OLAC-EC dockerd[1449]: time="2020-03-06T15:37:57.269804650+01
Mar 06 15:38:04 UC20-SL2000-OLAC-EC dockerd[1449]: time="2020-03-06T15:38:04.091475107+01
```

Note:
Since u-control studio has no entry for the DNS server in /etc/.resolv.conf , the docker environment cannot detect any user-defined DNS server.
Therefore the default DNS settings for docker will be used… 8.8.8.8 / 8.4.4.8 (Google).

5. Finally run the first container

```
sudo docker run hello-world
```

You should get a terminal output like this…



```
Hello from Docker!
This message shows that your installation appears to be working correctly.

To generate this message, Docker took the following steps:
 1. The Docker client contacted the Docker daemon.
 2. The Docker daemon pulled the "hello-world" image from the Docker Hub.
    (arm32v7)
 3. The Docker daemon created a new container from that image which runs the
    executable that produces the output you are currently reading.
 4. The Docker daemon streamed that output to the Docker client, which sent it
    to your terminal.

To try something more ambitious, you can run an Ubuntu container with:
 $ docker run -it ubuntu bash

Share images, automate workflows, and more with a free Docker ID:
 https://hub.docker.com/

For more examples and ideas, visit:
 https://docs.docker.com/get-started/

admin@UC20-SL2000-OLAC-EC:~$
```

6. Now you can use your docker environment.
   For additional information have a look at https://www.docker.com/

4.4. Portainer installation (community edition)

“Portainer is comprised of two elements, the Portainer Server, and the Portainer Agent. Both elements run as lightweight Docker containers on a Docker engine or within a Swarm cluster. Due to the nature of Docker, there are many possible deployment scenarios: We have detailed the most common below. Please use the scenario that matches your configuration best.

Note that the recommended deployment mode when using Swarm is using the Portainer Agent.” © portainer.io

1. Login to your PLC via ssh (we suggest to use PuTTY).

   Use the following Docker commands to deploy the Portainer Server; note the agent is not needed on standalone hosts, however it does provide additional functionality if used (see portainer and agent scenario below):

   ```
   $ docker volume create portainer_data
   $ docker run -d -p 8000:8000 -p 9000:9000 --name=portainer --restart=always -v
       /var/run/docker.sock:/var/run/docker.sock -v portainer_data:/data portainer/portainer-ce
   ```

2. Access the portainer.io web interface

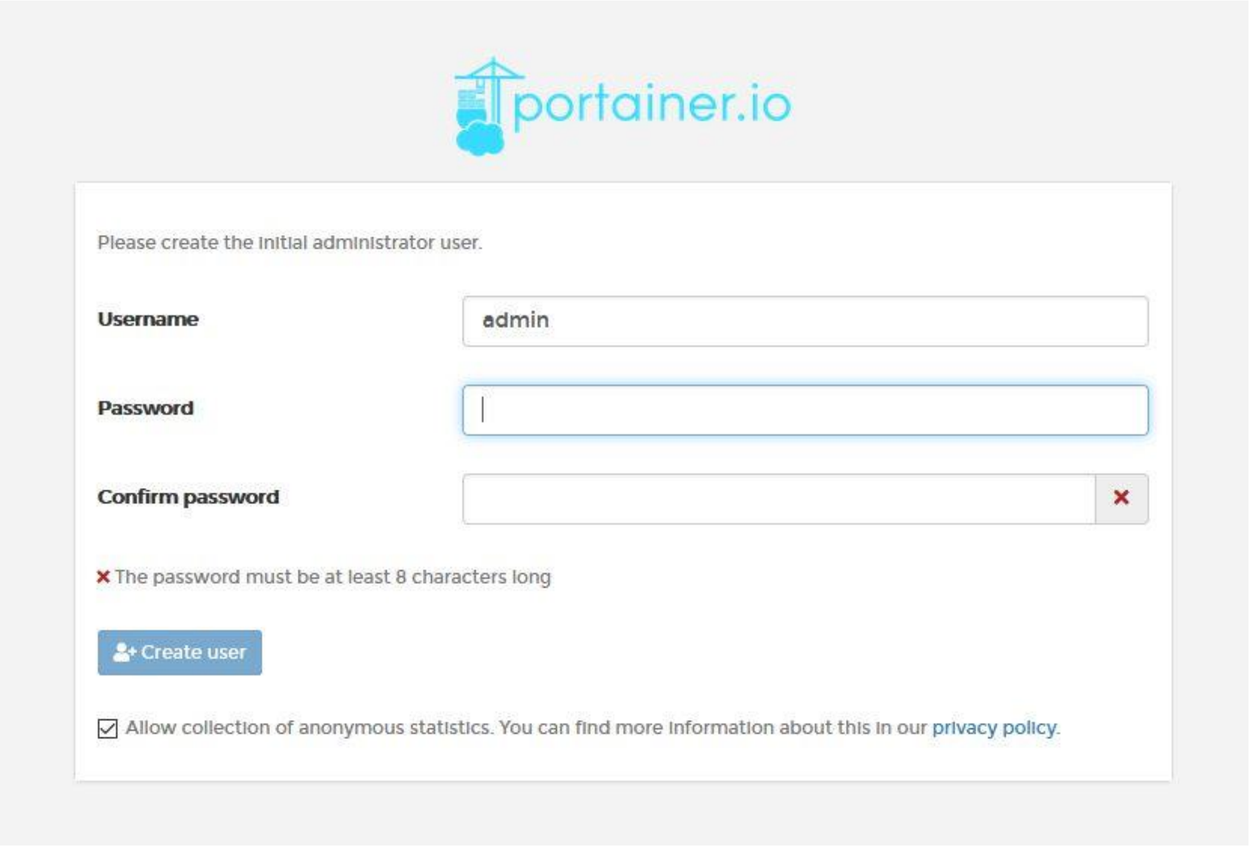
You access the portainer.io web interface via port 9000 of the Docker engine where portainer is running. To do so, open the IP address of your PLC in your browser and add “:9000”. Example:

The IP address of your PLC is 192.168.0.100. Then you use 192.168.0.100:9000 as URL in your Browser to load the portainer.io web interface of your PLC.

Note: Port 9000 is the general port used by Portainer for the UI access. Port 8000 is used exclusively by the EDGE agent for the reverse tunnel function. If you do not plan to use the edge agent, you do not need to expose port 8000 .

Note: the option “-v /var/run/docker.sock:/var/run/docker.sock” can be used in Linux environments only.

3. Create an administrative user. On the first access after installation, the portainer.io web interface prompts you with the “create initial administrator user” dialogue:



Please create the initial administrator user.

Username: admin

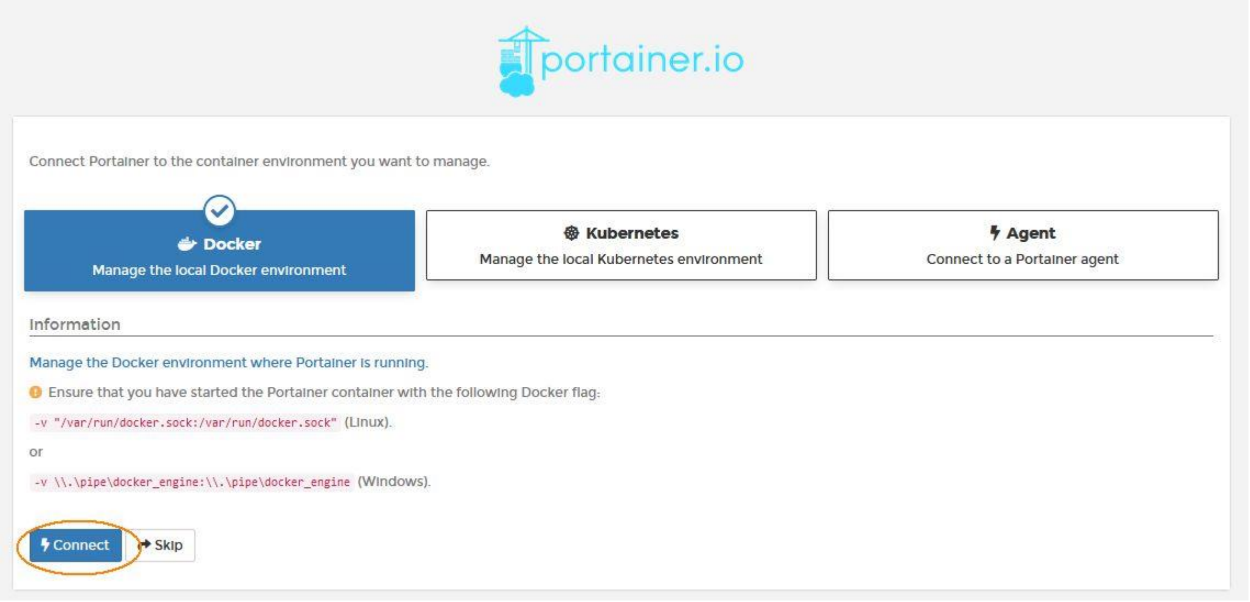
Password:

Confirm password:

✗ The password must be at least 8 characters long

☑ Allow collection of anonymous statistics. You can find more information about this in our privacy policy.

4. Choose Docker as the environment you would like to manage:


portainer.io
Docker
Manage the local Docker environment
Kubernetes
Manage the local Kubernetes environment
Agent
Connect to a Portainer agent

Connect Portainer to the container environment you want to manage.

Information

Manage the Docker environment where Portainer is running.

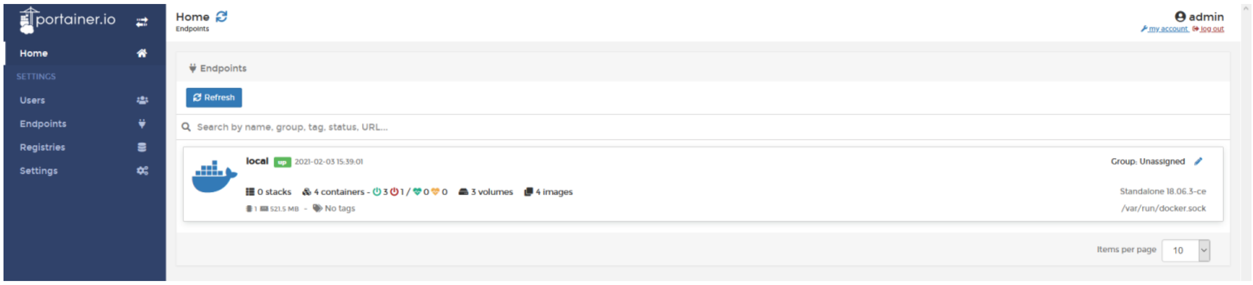
Ensure that you have started the Portainer container with the following Docker flag:

`-v "/var/run/docker.sock:/var/run/docker.sock"` (Linux).

or

`-v \\.\pipe\docker_engine:\\.\pipe\docker_engine` (Windows).

5. This will direct you to the home screen of portainer.io …



local up 2021-02-03 15:39:01

0 stacks 4 containers - 3 1 / 0 0 3 volumes 4 images

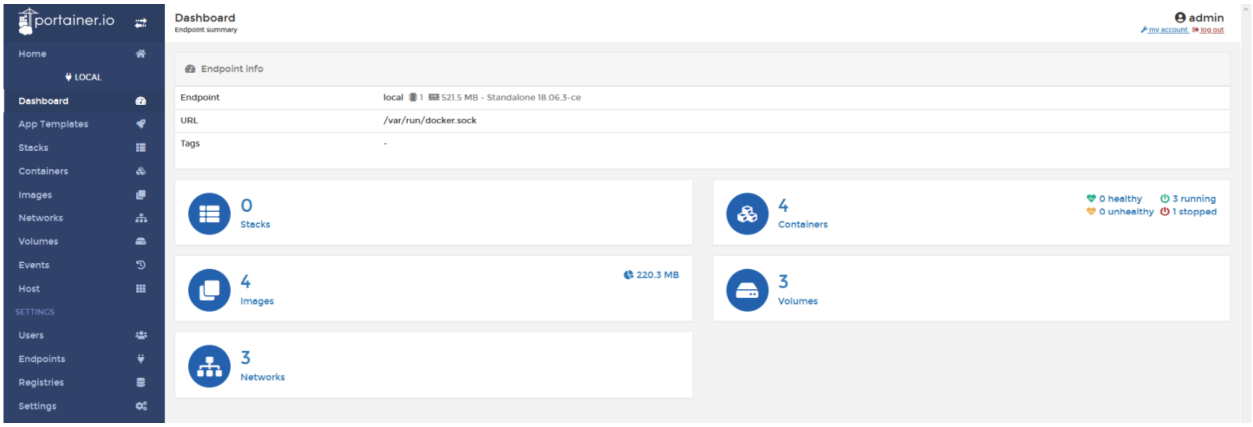
1 521.5 MB - No tags

Group: Unassigned

Standalone 18.06.3-ce

/var/run/docker.sock

6. Click on the “local” environment to go to the LOCAL Dashboard.
   This is the main window to manage your Docker environment.



Endpoint	local 1 521.5 MB - Standalone 18.06.3-ce
URL	/var/run/docker.sock
Tags	-

0 Stacks

4 Containers — 0 healthy, 3 running, 0 unhealthy, 1 stopped

4 Images — 220.3 MB

3 Volumes

3 Networks

ATTENTION:

Please always remember that the UC20-SL2000-OLAC’s resources are limited. It cannot be ruled out that the use of these features can have a negative effect on the real-time behavior of the PLC processing.