

## Postavljanje SYS-STEM ArdLAB laboratorija

### 1 Postavljanje Raspberry PI-a

Da biste postavili Raspberry PI, slijedite niže navedene korake:

- Preuzmite Raspberry PI OS (*operativni sustav*) sa
- Spremite sliku Raspberry PI OS-a na microSD karticu
- Postavite Raspberry PI OS povezivanje i sigurnosne postavke

#### 1.1 Instalirajte Raspberrry PI OS

Preuzmite Raspberry PI Imager za operativni sustav vašeg računala sa Raspberry web stranice ([poveznica](#)). Instalirajte Raspberry PI Imager i obavite sljedeće korake.

##### 1.1.1 Kopirajte Raspberry Pi OS na microSD karticu

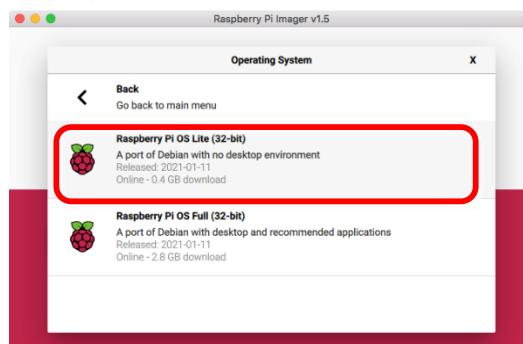
Izaberite „CHOOSE OS“ (IZABERITE OPERATIVNI SUSTAV)



Izaberite „Raspberry Pi OS (Other)“ (Raspberry Pi OS (ostalo))



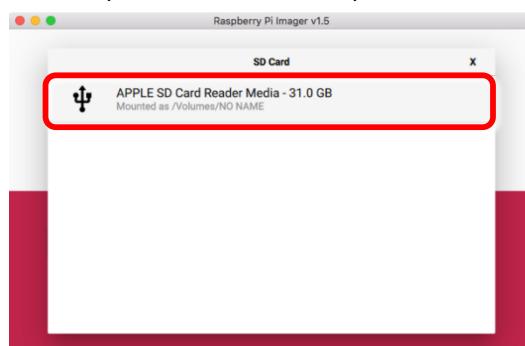
Izaberite „Raspberry Pi OS Lite (32-bit)“



Izaberite „Choose SD Card“ (Izaberite SD karticu)



Izaberite „The SD card to use...“ (Koristiti SD karticu...)

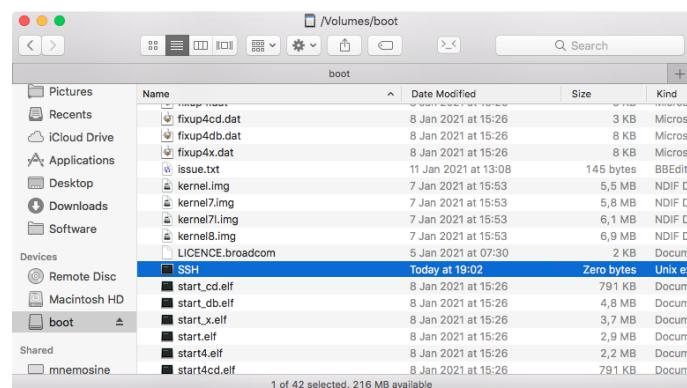


Na kraju, kopirajte sliku na microSD karticu pritiskom na „Write“ (piši)



### 1.1.2 Omogućite SSH prije prvog pokretanja

Da biste omogućili sigurno povezivanje na daljinu (SSH) s Raspberry Pi-em, prije prvog pokretanja potrebno je kreirati praznu SSH datoteku na particiji microSD kartice s koje se pokreće sustav.



## 1.2 Prvo pokretanje sustava i osnovna konfiguracija

Umetnите microSD karticu u svoje računalo i kreirajte datoteku (na Windows sustavu budite vrlo oprezni, jer većina uređivača uvijek dodaje ekstenziju, a zadano ponašanje za Windows sustav jest sakriti poznate ekstenzije)

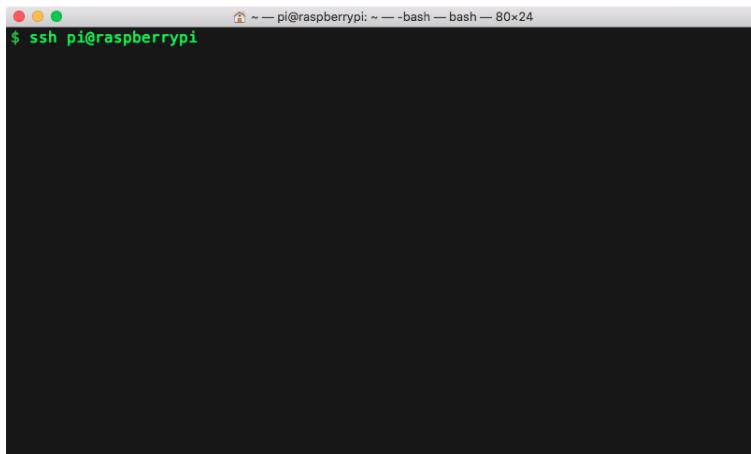
### 1.2.1 Instalirajte microSD karticu u Raspberry Pi i obavite prvo pokretanje

Da biste izbjegli potrebu za povezivanjem monitora i tipkovnice na Raspberry Pi, potrebno je priključiti mrežni kabel na Raspberry Pi. Nakon pokretanja Raspberry Pi-a, trebali biste se moći spojiti uporabom SSH.

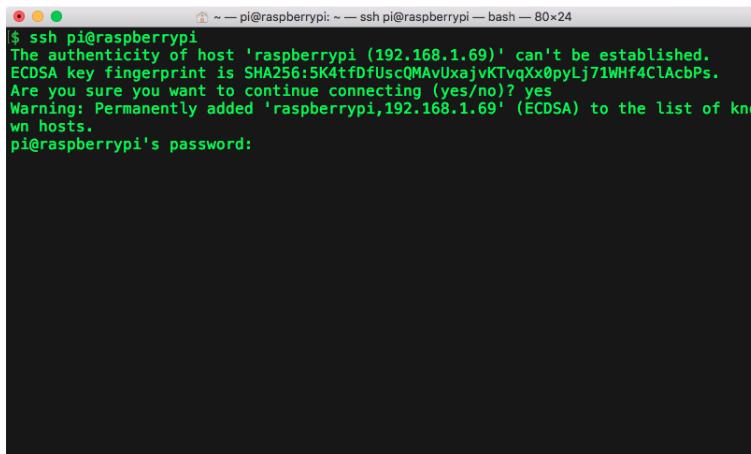
### 1.2.2 Spojite se na Raspberry Pi

- Da biste se spojili na Raspberry Pi:

```
ssh pi@raspberrypi
```



- Kada vam bude postavljeno pitanje o autentičnosti servera, odgovorite sa „yes“ (da) i prijavite se sa zadatom lozinkom „raspberry“.



- Ako se ne možete spojiti, trebat će spajti monitor na Raspberry PI, identificirati IP adresu na konzoli i upotrijebiti nju umjesto raspberrypi, npr.:

```
ssh pi@192.168.1.69
```

- Nakon prijave, trebali biste zamijeniti PI lozinku sa sigurnijom lozinkom. Koristite naredbu:

```
passwd
```

```

pi@raspberrypi: ~ -- ssh pi@raspberrypi -- bash -- 80x24
Are you sure you want to continue connecting (yes/no)? yes
Warning: Permanently added 'raspberrypi,192.168.1.69' (ECDSA) to the list of known hosts.
pi@raspberrypi's password:
Linux raspberrypi 5.4.83-v7l+ #1379 SMP Mon Dec 14 13:11:54 GMT 2020 armv7l

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.
Last login: Sun Feb 7 19:57:22 2021 from 192.168.1.65
-bash: warning: setlocale: LC_ALL: cannot change locale (pt_PT.UTF-8)

SSH is enabled and the default password for the 'pi' user has not been changed.
This is a security risk - please login as the 'pi' user and type 'passwd' to set
a new password.

Wi-Fi is currently blocked by rfkill.
Use raspi-config to set the country before use.

pi@raspberrypi:~ $ passwd

```

- Pokrenite mrežni informacijski program:

```
ifconfig
```

Zabilježite IP adresu eth0 sučelja

```

pi@ardlab-dei: ~ -- ssh pi@raspberrypi -- bash -- 80x24
pi@ardlab-dei:~ $ ifconfig
eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 192.168.1.69  netmask 255.255.255.0  broadcast 192.168.1.255
                inet6 fe80::2826:1161:e4e7:8f0  prefixlen 64  scopeid 0x20<link>
                inet6 2001:8a0:ff30:f700:e2cf:7707:32ef:9e38  prefixlen 64  scopeid 0x0<global>
                        ether dc:a6:32:7d:99:38  txqueuelen 1000  (Ethernet)
                        RX packets 98094  bytes 145123424 (138.4 MiB)
                        RX errors 0  dropped 0  overruns 0  frame 0
                        TX packets 47839  bytes 4259441 (4.0 MiB)
                        TX errors 0  dropped 0  overruns 0  carrier 0  collisions 0

lo: flags=73<UP,LOOPBACK,RUNNING>  mtu 65536
        inet 127.0.0.1  netmask 255.0.0.0
                inet6 ::1  prefixlen 128  scopeid 0x10<host>
                        loop  txqueuelen 1000  (Local Loopback)
                        RX packets 0  bytes 0 (0.0 B)
                        RX errors 0  dropped 0  overruns 0  frame 0
                        TX packets 0  bytes 0 (0.0 B)
                        TX errors 0  dropped 0  overruns 0  carrier 0  collisions 0

pi@ardlab-dei:~ $

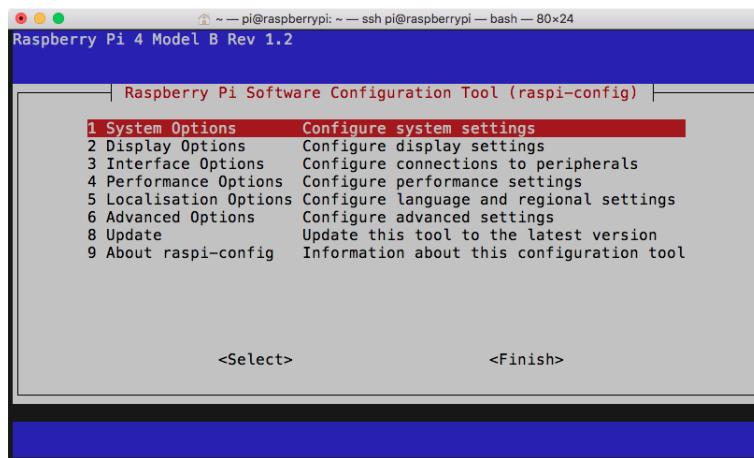
```

- Zatim možete postaviti neke od zadanih konfiguracija na Raspberry Pi-u. Trebali biste koristiti naredbu:

```
sudo raspi-config
```

Naredba sudo dozvoljava pokretanje raspi-config s „root“ (administratorskim) dozvolama.

- Otvara se konfiguracijski program za Raspberry



- Trebali biste konfigurirati sljedeće opcije:

1. Opcije sustava

**S4 Hostname** (ime računala) – Trebali biste promijeniti ime računala Raspberry, npr.: ArdLAB-1

Ako je Raspberry IP adresa javna, to ime potrebno je registrirati na DNS serveru. Ako je Raspberry IP adresa privatna, IP adresa bi trebala biti statična, a pravilo prosljeđivanja porta trebalo bi postaviti u NAT serveru za Raspberry Pi port 7575.

**S1 bežični LAN, nije preporučeno** – možete konfigurirati Raspberry Pi WiFi vezu (trebali biste spojiti pomoću ethernet kabela)

3. Opcije sučelja

**P1 Camera** (kamera) – ako koristite Raspberry Pi kao ArdLAB kameru, trebali biste to ovdje omogućiti

5. Opcije lokalizacije

**L2 Timezone** (vremenska zona) – Konfigurirajte vremensku zonu za vaš server, npr. Europa/Lisbon, Europa/Madrid, Europa/Atena, Europa/Zagreb, itd.

Nakon što izadete iz raspi-config, Raspberry Pi može zatražiti ponovno pokretanje.

### 1.3 Ažuriranje Raspberry Pi OS-a

Sljedeće biste trebali ažurirati Raspberry Pi OS upotrebom sljedećih naredbi u slijedu (također je moguće koristiti apt-get naredbu za ažuriranje/nadogradnju):

```
sudo apt update
sudo apt upgrade
sudo reboot
```

Ove naredbe: *update* (ažuriranje) informacija paketa, preuzimanjem sa konfiguiriranih Internet izvora; *upgrade* (nadogradnja) trenutno instaliranih zastarjelih paketa i na kraju *reboot* (ponovno pokretanje) Raspberry Pi-a

```
pi@ardlab-dei:~ $ sudo apt update
Hit:1 http://archive.raspberrypi.org/debian buster InRelease
Hit:2 http://raspbian.raspberrypi.org/raspbian buster InRelease
Reading package lists... Done
Building dependency tree
Reading state information... Done
28 packages can be upgraded. Run 'apt list --upgradable' to see them.
pi@ardlab-dei:~ $ sudo apt upgrade
Reading package lists... Done
Building dependency tree
Reading state information... Done
Calculating upgrade... Done
The following packages will be upgraded:
bluez-firmware ca-certificates device-tree-compiler file firmware-atheros
firmware-brcm80211 firmware-libertas firmware-misc-nonfree firmware-realtek
iproute2 libgnutls30 libldap-2.4-2 libldap-common libmagic-mgc libmagic1
libraspberrypi-bin libraspberrypi-dev libraspberrypi-doc libraspberrypi0
python-rpi.gpio raspberrypi-bootloader raspberrypi-kernel
raspberrypi-sys-mods rpi-eeprom rpi.gpio-common sudo tzdata unzip
28 upgraded, 0 newly installed, 0 to remove and 0 not upgraded.
Need to get 137 MB of archives.
After this operation, 3575 kB of additional disk space will be used.
pi@ardlab-dei:~ $
```

Instalirajte vim uređivač, za poboljšani vi uređivač

```
sudo apt install vim
```

Kako biste uredili datoteke u terminalu, možete koristiti vi ili nano. Za korisnike koji nisu upoznati s vi, nano uređivač je najbolja opcija. Nakon uredivanja datoteke, pritisnite ctrl+X za izlaz i izaberite odgovarajući odgovor za spremanje izmjena.

```
nano file.txt
```

## 2 Postavljanje ArdLAB laboratorija

Da biste postavili ArdLAB laboratorij, trebate slijediti niže navedene korake

- Instalirajte Docker daemon na Raspberry PI
- Klonirajte ArdLAB repozitorij
- Postavite i instalirajte ArdLAB Docker spremnik(e)
- Kreirajte ArdLAB konfiguracijsku datoteku
- Povežite Arduino UNO pločicu s Raspberry PI-em
- Testirajte postavke

### 2.1 Instalirajte docker na Raspberry Pi

**Docker** je set platformi kao uslužnih proizvoda koji rabe virtualizaciju na razini OS-a za isporuku softvera u paketima koji se nazivaju spremnici (*containers*). Spremnici su međusobno izolirani i grupiraju vlastiti softver, knjižnice i konfiguracijske datoteke; mogu međusobno komunicirati putem jasno definiranih kanala. Obzirom da svi spremnici dijele usluge jedne jezgre (*kernel*) operativnog sustava, koriste manje resursa od virtualnih računala.

Potrebno je slijediti sljedeće korake za instalaciju dockera na Raspberry PI (prema uputama dostupnim na ovoj poveznici [poveznica](#))

- Instalirajte preduvjete

```
sudo apt install apt-transport-https ca-certificates software-properties-common -y
```

- Ponovno pokrenite Raspberry Pi

```
sudo reboot
```

- Preuzmite Docker instalacijski program i pokrenite ga

```
curl -fsSL get.docker.com -o get-docker.sh && sh get-docker.sh
```

- Pružite „pi“ korisniku mogućnost pokretanja Dockera.

```
sudo usermod -aG docker pi
```

- Uvezite Docker CPG ključ

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

- Dodajte docker repozitorij popisu apt izvora

```
sudo sh -c 'echo "deb https://download.docker.com/linux/raspbian/  
buster stable" >> /etc/apt/sources.list'
```

- Ažurirajte Raspberry Pi OS i ponovno pokrenite

```
sudo apt update  
sudo apt upgrade  
sudo reboot
```

- Provjerite je li docker daemon pokrenut (pritisnite q za izlazak iz popisa)

```
systemctl status docker
```

```
pi@ardlab-dei:~ $ systemctl status docker
* docker.service - Docker Application Container Engine
  Loaded: loaded (/lib/systemd/system/docker.service; enabled; vendor preset: e
  Active: active (running) since Mon 2021-02-08 15:33:18 CET; 7min ago
    Docs: https://docs.docker.com
   Main PID: 634 (dockerd)
      Tasks: 11
     CGroup: /system.slice/docker.service
             `--634 /usr/bin/dockerd -H fd:// --containerd=/run/containerd/containerd.sock

Feb 08 15:33:16 ardlab-dei dockerd[634]: time="2021-02-08T15:33:16.270203137Z" l
Feb 08 15:33:16 ardlab-dei dockerd[634]: time="2021-02-08T15:33:16.270460008Z" l
Feb 08 15:33:16 ardlab-dei dockerd[634]: time="2021-02-08T15:33:16.270690452Z" l
Feb 08 15:33:16 ardlab-dei dockerd[634]: time="2021-02-08T15:33:16.271515600Z" l
Feb 08 15:33:17 ardlab-dei dockerd[634]: time="2021-02-08T15:33:17.522143377Z" l
Feb 08 15:33:18 ardlab-dei dockerd[634]: time="2021-02-08T15:33:18.029431303Z" l
Feb 08 15:33:18 ardlab-dei dockerd[634]: time="2021-02-08T15:33:18.400757636Z" l
Feb 08 15:33:18 ardlab-dei dockerd[634]: time="2021-02-08T15:33:18.401986192Z" l
Feb 08 15:33:18 ardlab-dei systemd[1]: Started Docker Application Container Engine.
Feb 08 15:33:18 ardlab-dei dockerd[634]: time="2021-02-08T15:33:18.593151062Z" l
lines 1-19/19 (END)
```

Nakon ponovnog pokretanja, docker usluga trebala bi raditi. Da biste upravljali docker uslugom, možete koristiti sljedeće systemctl opcije:

*start* – pokreće uslugu

*stop* – zaustavlja uslugu

*restart* – ponovno pokreće uslugu, ekvivalentno je sa *stop + start*

*reload* – ponovno učitava konfiguracijske datoteke usluge

*enable* – omogućuje uslugu (automatska pokretanja kod pokretanja sustava)

*disable* – onemogućuje uslugu

- Provjerite informacije o instaliranom dockeru

```
docker info
```

- Instalirajte docker compose

```
sudo apt install docker-compose
```

## 2.2 Kreirajte i pokrenite prvi spremnik

- Da biste isptali Docker instalaciju, kreirajte i pokrenite hello-world spremnik

```
docker run hello-world
```

```
pi@ardlab-dei: ~ $ docker run hello-world
Unable to find image 'hello-world:latest' locally
latest: Pulling from library/hello-world
4ee5c797bcd7: Pull complete
Digest: sha256:31b9c7d48790f0d8c50ab433d9c3b7e17666d6993084c002c2ff1ca09b96391d
Status: Downloaded newer image for hello-world:latest

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/
```

Ako vidite poruku iznad, spremnik je kreiran i pokrenut. Da biste upravljali spremnicima i slikama, možete koristiti sljedeće naredbe:

- Prikažite slike spremnika koje lokalno postoje

```
docker image ls
```

- Prikažite docker stranicu za pomoć

```
docker --help
```

- Prikažite stranicu za pomoć za docker naredbe

```
docker image --help
```

- Uklonite lokalnu sliku (koristite kratku hash (oznaku) ili ime, koristite docker image ls za prikaz imena/oznaka)

Koristite tipku TAB da biste automatski dopunili imena/naredbe.

- Slike korištene u spremnicima nije moguće ukloniti bez uklanjanja spremnika.

```
docker image remove hello-world:latest
```

- Alternativna naredba za uklanjanje slika jest:

```
docker rmi hello-world:latest
```

- Prikažite pokrenute spremnike

```
docker ps
```

- Prikažite sve spremnike

```
docker ps -a
```

- Sada možete ukloniti hello-world spremnik, uporabom imena/oznake

```
docker rm happy_rhodes
```

- Alternativna naredba za uklanjanje spremnika uporabom ključa oznake (hash key)

```
docker rm 2cdaa3512ba
```

Rezultat nekih od prethodnih naredbi:

```
pi@ardlab-dei:~ $ docker image ls
REPOSITORY TAG IMAGE ID CREATED SIZE
hello-world latest 851163c78e4a 13 months ago 4.85kB
pi@ardlab-dei:~ $ docker image remove hello-world:latest
Error response from daemon: conflict: unable to remove repository reference "hello-world:latest" (must force) - container 2cdaa35120ba is using its referenced image 851163c78e4a
pi@ardlab-dei:~ $ docker ps -a
CONTAINER ID IMAGE COMMAND CREATED STATUS
PORTS NAMES
2cdaa35120ba hello-world "/hello" 34 seconds ago Exited (0) 33 seconds ago
pi@ardlab-dei:~ $ docker rm happy_rhodes
happy_rhodes
pi@ardlab-dei:~ $ docker rmi hello-world:latest
Untagged: hello-world:latest
Untagged: hello-world@sha256:31b9c7d48790f0d8c50ab433d9c3b7e17666d6993084c002c2f
f1ca09b96391d
Deleted: sha256:851163c78e4ad68e6fe5391f0894aafdf164d40c4d0a56b4291f0dc2c27cc2c
Deleted: sha256:2536d8d4e4b1baa6515d44eb77a1402d6be0a533e7d191c51cb8428ba5ece3f4
pi@ardlab-dei:~ $
```

## 2.3 Postavite ArdLAB laboratorij

- Klonirajte repozitorij unutar pi početnog direktorija korisnika, pomoću sljedeće naredbe:

```
git clone https://bitbucket.org/pssmatos/ardlab-setup.git
```

Kreira se nova mapa pod nazivom ardlab-setup s konfiguracijskom datotekom spremnika

### 2.3.1 Postavite i instalirajte ArdLAB Docker spremnik

- Promijenite trenutni direktorij u ardlab-setup (možete koristiti tipku TAB da biste automatski dopunili naziv direktorija)

```
cd ardlab-setup
```

- Izlistajte sadržaj direktorija:

```
ls -la
```

Samo je jedna važna datoteka u ovom repozitoriju, docker-compose definicijska datoteka

```
pi@ardlab-dei:~ $ git clone https://pssmatos@bitbucket.org/pssmatos/ardlab-setup
.git
Cloning into 'ardlab-setup'...
remote: Counting objects: 4, done.
remote: Compressing objects: 100% (3/3), done.
remote: Total 4 (delta 0), reused 0 (delta 0)
Unpacking objects: 100% (4/4), done.
pi@ardlab-dei:~ $ cd ardlab-setup/
pi@ardlab-dei:~/ardlab-setup $ ls -la
total 20
drwxr-xr-x 3 pi pi 4096 Feb 12 05:39 .
drwxr-xr-x 12 pi pi 4096 Feb 12 05:39 ..
drwxr-xr-x 8 pi pi 4096 Feb 12 05:39 .git
-rw-r--r-- 1 pi pi 9 Feb 12 05:39 .gitignore
-rw-r--r-- 1 pi pi 177 Feb 12 05:39 docker-compose.yml
pi@ardlab-dei:~/ardlab-setup $
```

- Započnite konfiguraciju pokretanjem spremnika za kreiranje konfiguracijske datoteke koristeći sljedeću naredbu:

```
docker-compose run --rm --entrypoint "python3 constellation.py"
ardlab
```

Skripta bi se trebala pokrenuti i prikazati neke informacije: javnu IP adresu, novu ArdLAB oznaku (*hash code*) i jednu poruku za registriranje te oznake na stranici konstelacije (ili dodavanjem novog laboratorija ili uređivanjem postojećeg)

```
pi@ardlab-dei:~/ardlab-setup $ ls -la
total 20
drwxr-xr-x  3 pi  pi  4096 Feb 12  05:45 .
drwxr-xr-x 12 pi  pi  4096 Feb 12  05:48 ..
drwxr-xr-x  8 pi  pi  4096 Feb 12  05:39 .git
-rw-r--r--  1 pi  pi   9 Feb 12  05:39 .gitignore
-rw-r--r--  1 pi  pi  177 Feb 12  05:39 docker-compose.yml
pi@ardlab-dei:~/ardlab-setup $ docker-compose run --rm --entrypoint "python3 constellation.py" ardlab
Starting ArdLab Service...
After import_module
Hash ID is
Camera is OpenCV
Loading OpenCV camera module...
My public IP address is: 188.83.57.83
No hash, creating new hash 7c634113322042d9865f24675c46dd6b
You must add this hash to the constellation
pi@ardlab-dei:~/ardlab-setup $
```

- Prethodni korak također je kreirao i konfiguracijsku datoteku, sadržanu u konfiguraciji direktorija koju je moguće prikazati upotrebotom:

```
cat config/config.cfg
```

```
pi@ardlab-dei:~/ardlab-setup $ ls -la
total 24
drwxr-xr-x  4 pi  pi  4096 Feb 12  05:49 .
drwxr-xr-x 12 pi  pi  4096 Feb 12  05:48 ..
drwxr-xr-x  8 pi  pi  4096 Feb 12  05:39 .git
-rw-r--r--  1 pi  pi   9 Feb 12  05:39 .gitignore
drwxr-xr-x  2 root root 4096 Feb 12  05:49 config
-rw-r--r--  1 pi  pi  177 Feb 12  05:39 docker-compose.yml
pi@ardlab-dei:~/ardlab-setup $ ls -la config
total 12
drwxr-xr-x 2 root root 4096 Feb 12  05:49 .
drwxr-xr-x 4 pi  pi  4096 Feb 12  05:49 ..
-rw-r--r--  1 root root 402 Feb 12  05:49 config.cfg
pi@ardlab-dei:~/ardlab-setup $ cat config/config.cfg
ARDLAB_HASH="7c634113322042d9865f24675c46dd6b"
CONSTELATION="https://sys-stem.dei.isept.ipp.pt:8080/api/lab/"
# Camera connected to the Raspberry PI, options are:
# "pi"           - Use pi as video provider
# "OpenCV"       - Use OpenCV as video provider
# "v4l2"         - Use Video 4 Linux 2 video provider
#CAMERA_TYPE="pi"
#CAMERA_TYPE="OpenCV"
#CAMERA_TYPE=""
CAMERA_TYPE="OpenCV"
OPENCV_CAMERA_SOURCE="0"
pi@ardlab-dei:~/ardlab-setup $
```

Ovu konfiguracijsku datoteku ćemo urediti u kasnijem koraku, za konfiguraciju imena, porta i kamere vašeg ArdLAB laboratorija.

### 2.3.2 Povežite Arduino UNO s Raspberry PI-em

Sada možete povezati Arduino UNO s Raspberry PI-em koristeći jedan od USB portova

### 3 Konfiguracija dDNS-a i dobivanje SSL certifikata

Za pristupanje ArdLAB laboratoriju s enkripcijom (HTTPS), trebamo koristiti valjani SSL certifikat, kojeg je, ovisno o laboratoriju, moguće dobiti na dva načina.

Ako ArdLAB laboratorij ima fiksnu javnu IP adresu i valjano DNS ime, nastavite na poglavlju 3.3.

U slučaju da vaš ArdLAB laboratorij ima jedno od sljedećih ograničenja, nastavite na odgovarajućem poglavlju:

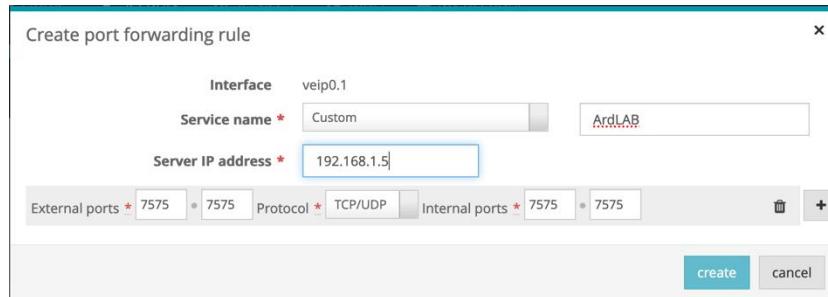
- Fiksna IP adresa, ali ne možete dodijeliti DNS ime (poglavlje 3.2 i 3.3)
- Dinamička javna IP adresa (poglavlje 3.2 i 3.3)
- Privatna IP adresa (poglavlja 3.1, 3.2 i 3.3)

#### 3.1 Dodavanje pravila prosljeđivanja porta vašem ArdLAB laboratoriju

Ako je vaš ArdLAB laboratorij iza NAT usmjerivača, ima privatnu IP adresu, ali laboratoriju neće biti omogućen pristup preko Interneta. Da biste riješili ovaj problem morate dodati *PRAVILO PROSLJEĐIVANJA PORTA* na NAT serveru da biste preusmjerili dolazni promet na ArdLAB komunikacijski port (7575).

Zamolite svog mrežnog administratora da kreira to pravilo, ili ako imate pristup NAT serveru (ili usmjerivaču), konfigurirajte ga sami. Zadani ArdLAB port je 7575 (možete koristiti drugi, ako port 7575 nije dostupan), a IP adresa servera bit će privatna IP adresa ArdLAB laboratorija.

Sljedeća slika prikazuje konfiguraciju takvog pravila na stranici za konfiguraciju usmjerivača.



The screenshot shows a 'Create port forwarding rule' dialog box. The 'Interface' field is set to 'veip0.1'. The 'Service name' field has 'Custom' selected, with the text 'ArdLAB' typed into the input field. The 'Server IP address' field contains '192.168.1.5'. Below these, there are two 'External ports' fields, both set to '7575', and two 'Internal ports' fields, also both set to '7575'. A 'Protocol' dropdown is set to 'TCP/UDP'. At the bottom right are 'create' and 'cancel' buttons.

#### 3.2 Dodavanje vašeg Raspberry PI-a DDNS usluzi

Ako vaš ArdLAB laboratorij nema fiksnu IP adresu s valjanim DNS imenom, trebat ćeće kreirati račun kod DDNS pružatelja usluge i kreirati novi DNS registar za ArdLAB.

U ovom primjeru, koristit ćeće Dynu dDNS uslugu (<https://www.dynu.com/>).

Važno je da pružatelj usluge može kreirati „TXT zapise“ upotrebom **ACME Shell skripte: acme.sh dnsapi** (<https://github.com/acmesh-official/acme.sh/wiki/dnsapi>)

**NAPOMENA:** ako imate privatnu adresu ili javnu IP adresu bez DNS imena na svom ArdLAB laboratoriju, idite na sljedeće poglavlje, ako ne, idite na poglavlje 3.3.

### 3.2.1 Kreirajte račun na Dynu dDNS usluzi

- Pristupite <https://www.dynu.com/> stranici:
  - 1 Kliknite na poveznicu „Create Account“ (kreiraj račun) gore desno
  - 2 Popunite svoje podatke
  - 3 Predajte „Create Account“ zahtjev za kreiranje računa
  - 4 Otvorite svoju el. poštu i potražite „Dynu Verification Email“ poruku za verifikaciju i kliknite na URL

Slika 1 - Kreirajte račun na Dynu dDNS usluzi

### 3.2.2 Kreirajte DNS registar za svoj ArdLAB

- Prijavite se na svoj račun na <https://www.dynu.com/> stranici:
  - 1 Pristupite upravljačkoj ploči „Control Panel“ (zupčanik)
  - 2 U upravljačkoj ploči izaberite „DDNS Services“ usluge
  - 3 Na stranici „Dynamic DNS Service“ dinamička DNS usluga, kliknite na „+Add“ za dodavanje
  - 4 Upišite ime svog laboratorija (host) i izaberite „Top Level“ domenu
  - 5 Kliknite na „+ Add“ za dodavanje

Slika 2 - Kreirajte DNS registar za svoj ArdLAB

### 3.2.3 Upravljaljte svojim ArdLAB dDNS registrom

- Pojavit će se „Manage DNS Service“ (upravljanje DNS uslugom) i trebali biste:
  - 1 Promijeniti svoju IPv4 adresu u „invalid“ (nevažeća), npr. 1.1.1.1
  - 2 Kliknite na „Save“ za spremanje da biste ažurirali svoju IPv4 adresu
  - 3 Kliknite na „IP Update Password“ (lozinka za ažuriranje IP adrese) na dnu stranice
- Na stranici „Manage Credentials“ upravljanje vjerodajnicama:
  - 4 Unesite svoju Dynu lozinku
  - 5 Unesite novu IP Update lozinku (ta lozinka će se rabiti na ArdLAB laboratoriju)
  - 6 Kliknite „Save“ za spremanje da biste ažurirali lozinku

Slika 3 - Upravljaljte svojim ArdLAB dDNS registrom

### 3.2.4 Konfigurirajte ArdLAB da automatski ažurira Dynu IP adresu

Sada trebamo konfigurirati Raspberry da objavi javnu adresu ArdLAB laboratorija na Dynu stranici (upute na <https://www.dynu.com/DynamicDNS/IPUpdateClient/RaspberryPi-Dynamic-DNS>).

- U Raspberry početnom direktoriju kreirajte novi direktorij pod nazivom dynudns i napravite sljedeće promjene

```
cd ~
mkdir dynudns
cd dynudns
```

- Kreirajte novu skriptu pod nazivom dynu.sh

```
nano dynu.sh
```

- Dodajte ovaj redak dynu.sh skripti, zamjenjujući HOST s domenom „Domain“ koju ste kreirali na *Dynu* stranici, a IP\_PASS s „IP Update password“ lozinkom koju ste definirali na *Dynu* stranici.

```
echo
url="https://api.dynu.com/nic/update?hostname=HOST&password=IP_PASS
$&myipv4=$(curl -s -4 ifconfig.co)&myipv6=$(curl -s -6
ifconfig.co)" | curl -k -o ~/dynudns/dynu.log -K -
```

Provjerite je li naredba napisana u samo jednom retku i jesu li ime računala i lozinka točni

Pritisnite CTRL+X za pohranu, a Y za potvrdu pohrane dokumenta, zatim ENTER za potvrdu naziva datoteke

- Promijenite dozvole skripte na 700

```
chmod 700 dynu.sh
```

- Na Raspberry Pi terminalu pokrenite sljedeću naredbu da biste dodali skriptu vašem crontab-u

```
crontab -e
```

Kada po prvi put pokrenete ovu naredbu, morate odabrati preferirani uređivač teksta:  
**izaberite nano**

- Zalijepite sljedeći redak na dno uređivača teksta

```
*/5 * * * * ~/dynudns/dynu.sh >/dev/null 2>&1
```



Pritisnite CTRL+X za pohranu, a Y za potvrdu pohrane dokumenta, zatim ENTER za potvrdu naziva datoteke

```
GNU nano 3.2 /tmp/crontab.7nMj51/crontab
# Edit this file to introduce tasks to be run by cron.
#
# Each task to run has to be defined through a single line
# indicating with different fields when the task will be run
# and what command to run for the task
#
# To define the time you can provide concrete values for
# minute (m), hour (h), day of month (dom),
# and day of week (dow) or use '*' in these fields (for 'any').
#
# Notice that tasks will be started based on the cron's system
# daemon's notion of time and timezones.
#
# Output of the crontab jobs (including errors) is sent through
# email to the user the crontab file belongs to (unless redirected).
#
# For example, you can run a backup of all your user accounts
# at 5 a.m. every week with:
# 0 5 * * 1 tar -zcf /var/backups/home.tgz /home/
#
# For more information see the manual pages of crontab(5) and cron(8)
#
# m h dom mon dow   command
*/5 * * * * ~/dynudns/dynu.sh >/dev/null 2>&1

FG Get Help   W Write Out   W Where Is   K Cut Text   J Justify   C Cur Pos   M-U Undo
PX Exit      R Read File   V Replace    U Uncut Text  T To Spell   G Go To Line  M-E Redo
```

Sada imate cronjob koji se pokreće svakih 5 minuta i ažurirat će vašu javnu IP adresu na Dynu dDNS serveru. Možete provjeriti na svojoj Dynu poddomeni da vidite hoće li se IP adresa promijeniti u sljedećih 5 minuta.

Ako se IP adresa ne ažurira, možete pogledati `~/dynudns/dynu.log` datoteku i provjeriti ima li grešaka. Ako dobijete `badauth`, to znači da ste unijeli pogrešnu domenu ili IP Update lozinku.

### 3.3 Izdavanje i obnova SSL certifikata

Za pristup serveru korištenjem HTTPS-a potreban je valjani SSL certifikat. Za pojednostavljenje dobivanja i obnove certifikata, potreban je alat za automatizaciju certifikata. Koristit ćemo `acme.sh` (ACME Shell skriptu), koja može pribaviti i obnoviti certifikate od različitih tijela za izdavanje certifikata (*Certification Authorities – CAs*) i koristiti nekoliko vrsta izazova (*Challenge Types*) za potvrdu vlasništva domene.

#### 3.3.1 Instalirajte acme.sh skriptu za automatizaciju obnove certifikata

- Instalirajte `acme.sh` skriptu

```
cd ~
curl https://get.acme.sh | sh -s email=my@example.com
```

- Nakon instalacije, morate zatvoriti trenutni terminal i ponovno ga otvoriti kako bi alias stupio na snagu. Ponovno se prijavite i pokrenite sljedeću naredbu

```
acme.sh --version
```

Ako vidite `acme.sh` verziju, skripta je ispravno instalirana i možete nastaviti s generiranjem certifikata.

**Napomena:** ako imate staru „ArdLAB“ verziju, trebat ćete sada ažurirati spremnik.

```
cd ~/ardlab-setup
git pull
docker-compose down
docker-compose pull
docker-compose up --detach
```



### 3.3.2 Generirajte certifikat za ArdLAB na privatnoj IP adresi (upotrebom proslijedivanja porta)

Ako imate privatnu IP adresu i kreirali ste Dynu dDNS zapis, sada trebate generirati certifikat upotrebom DNS izazova (*challenge*) (ako imate javnu IP adresu, nastavite na poglavlju 3.3.3.).

U svom početnom direktoriju pokrenite sljedeće naredbe, zamjenjujući `ClientId` i `Secret` sa svojim API ključevima, dobivenima na Dynu „Manage Dynamic DNS Service“ (upravljanje dinamičkom DNS uslugom) → „api credentials“ (API vjerodajnice) (vidjeti Slika 3.). Također trebate zamijeniti domenu `DOMAIN` iza –d svojom Dynu dDNS domenom.

**Napomena:** trebate samo izvesti vjerodajnice kod prvog pokretanja, jer `acme.sh` sprema ključeve u svoje konfiguracijske datoteke.

- Ti konfiguracijski koraci predstavljeni su na stranici `acme.sh`, „How to use DNS API“ (Kako koristiti DNS API)  
<https://github.com/acmesh-official/acme.sh/wiki/dnsapi#24-use-dynu-api>

```
export Dynu_ClientId=xxxxxxxxx-xxxx-xxxx-xxxx-xxxxxxxxxxxx
export Dynu_Secret=yyyyyyyyyyyyyyyyyyyyyyyyyyyy
acme.sh --issue --dns dns_dynu -d DOMAIN \
--cert-file /home/pi/ardlab-setup/certs/cert.pem \
--key-file /home/pi/ardlab-setup/certs/key.pem \
--fullchain-file /home/pi/ardlab-setup/certs/fullchain.pem \
--reloadcmd "docker restart ardlab-setup_ardlab_1"
```

- Nakon što je naredba izvršena, možete provjeriti svoje izdane certifikate sljedećom naredbom

```
acme.sh -list
```

Sada možete nastaviti na poglavlju 0.

### 3.3.3 Generirajte certifikat na ArdLAB laboratoriju s javnom IP adresom

- Da biste generirali certifikat na ArdLAB laboratoriju s javnom IP adresom, trebate instalirati `socat`

```
sudo apt install socat
```

- Zatim trebate dozvoliti da `socat` prihvati spajanja na ograničenim portovima

```
sudo setcap 'cap_net_bind_service=+ep' /usr/bin/socat
```

- Na kraju možete pokrenuti `acme.sh` kako biste zatražili novi certifikat. Nemojte zaboraviti zamijeniti domenu `DOMAIN` iza –d svojom ArdLAB domenom.

```
acme.sh --issue --standalone -d DOMAIN \
--cert-file /home/pi/ardlab-setup/certs/cert.pem \
--key-file /home/pi/ardlab-setup/certs/key.pem \
--fullchain-file /home/pi/ardlab-setup/certs/fullchain.pem \
--reloadcmd "docker restart ardlab-setup_ardlab_1"
```

- Nakon što je naredba izvršena, možete provjeriti svoje izdane certifikate sljedećom naredbom

```
acme.sh -list
```

## 4 Dodajte ArdLAB Konstelaciji

Da biste dodali ArdLAB laboratorij Sys-Stem ArdLAB Konstelaciji, trebate slijediti sljedeće korake:

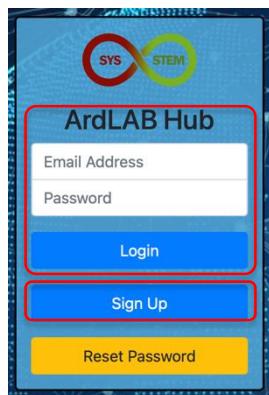
- Kreirajte korisnički račun za ArdLAB Konstelaciju (samo ako ga već nemate)
- Dodajte svoj ArdLAB laboratorij Konstelaciji (to mora odobriti administrator Konstelacije)
- Pokrenite ArdLAB konfiguraciju
- Pokrenite ArdLAB spremnik
- Dodajte skice primjera i testirajte svoj ArdLAB na stranici Konstelacije

### 4.1 Konfigurirajte ArdLAB na Hub-u Konstelacije

Da biste dodali svoj ArdLAB laboratorij Hub-u Konstelacije trebate imati račun, a zatim dodati svoje ArdLAB parametre hub-u.

#### 4.1.1 Kreirajte korisnički račun za ArdLAB Konstelaciju

- Pristupite stranici SYS-STEM ArdLAB Konstelacije (<https://sys-stem.dei.isep.ipp.pt/>)
- Ako već imate račun, prijavite se, ako nemate, koristite gumb Sign-up da biste kreirali novi korisnički račun.



#### 4.1.2 Dodajte svoj ArdLAB Konstelaciji

- Izaberite opciju izbornika „Add your ArdLAB“ (dodajte svoj ArdLAB) → „Add your Laboratory to the Constellation“ (dodajte svoj Laboratorij Konstelaciji).



- Ispunite New ArdLAB obrazac za novi ArdLAB, koristite oznaku koja je prethodno kreirana i pritisnite gumb „Add Laboratory“ (dodajte laboratorij)

Laboratorij treba odobriti jedan od administratora SYS-STEM Konstelacije, stoga prije nego što možete nastaviti sa sljedećom konfiguracijom, trebate pričekati odobrenje (primit ćete el. poruku nakon ArdLAB odobrenja).

## 4.2 Pokrenite ArdLAB konfiguraciju

- Sada možete urediti konfiguracijsku datoteku ArdLAB laboratorija da biste konfiguirali ArdLAB DNS ime, port i kameru

```
cd ~/ardlab-setup
sudo nano config/congif.cfg
```

- Sljedeće je primjer config.cfg s nekim od najčešćih konfiguracija

```
ARDLAB_HASH="a826b23115b84f0d8c0d040b89739b"
CONSTELLATION="https://sys-stem.dei.isep.ipp.pt:8080/api/lab/"
HOSTNAME="ardlab.myInstitution.com"
FORWARD_PORT=7575
# Camera connected to the Raspberry PI, options are:
# "pi" - Use pi as video provider
# "OpenCV" - Use OpenCV as video provider
#CAMERA_TYPE="pi"
CAMERA_TYPE="OpenCV"
OPENCV_CAMERA_SOURCE="0"
```

Parametri koje možete promijeniti su:

**HOSTNAME** – to je ime koje ste registrirali na DNS serveru za svoj ArdLAB

**FORWARD\_PORT** – to je port koji ćete proslijediti na svoj NAT usmjerivač. Trebate dodati ovaj podatak samo ako se port NAT usmjerivača razlikuje od 7575

**CAMERA\_TYPE** vrsta kamere – koristite PI za Raspberry PI kameru i OpenCV za USB kameru

- Kada je Laboratorij odobren trebate ponovno pokrenuti spremnik da biste se „povezali“ s Konstelacijom

```
docker-compose run --rm --entrypoint "python3 constellation.py"
ardlab
```

Ako je naredba uspješna, `constellation.py` skripta će prikazati sljedeće informacije

```
● ○ ● /Volumes/Macintosh HD/Users/matos/Documents/cadeiras/UpSkill/Curso OutSystems/OutSystems/React...
Hash ID is c496af822e6546b8ada11b5a48c765ba
Camera is OpenCV
Loading OpenCV camera module...
My public IP address is: 188.83.57.83
2021-02-12 06:24:33,081 - https://sys-stem.dei.isep.ipp.pt:8080/api/lab/c496af82
2e6546b8ada11b5a48c765ba/188.83.57.83:7575
2021-02-12 06:24:34,090 - {"hash": "c496af822e6546b8ada11b5a48c765ba", "ip": "18
8.83.57.83", "port": 7575}
2021-02-12 06:24:34,091 - status code: 200
2021-02-12 06:24:34,092 - Response: b'{"labId":2,"labName":"New ArdLab DEI","ipA
ddress":"188.83.57.83:7575","country":"Portugal","OwnerId":1,"status":"Active","
labSketches":[],"LabComponents":[{"componentId":1,"name":"LED","description":"A
light-emitting diode (LED) is a semiconductor light source that emits light when
current flows through it"}, {"componentId":2,"name":"LCD Display","description":
"A liquid-crystal display (LCD) is a flat-panel display which display characters
made from a matrix of small pixels,"}, {"componentId":4,"name":"Servomotor","des
cription":"A servomotor is a rotary or linear actuator that allows for precise c
ontrol of angular/linear position, velocity and acceleration"}, {"componentId":5,
"name":"Infrared (IR) Sensor","description":"An sensor capable of pick up infrar
ed signals from an IR-LED or remote control just like a TV or Cable box"}, {"com
ponentId":7,"name":"Potentiometer","description":"A variable resistor with a thir
d adjustable terminal; The potential at the third terminal can be adjusted to gi
ve any fraction of the potential across the ends of the resistor"}],"labLogoImag
eId":"","labSchemaImageId":""}'
pi@ardLab-dei:~/ardlab-setup $
```

#### 4.2.1 Pokrenite ArdLAB spremnik

- Na kraju, možete pokrenuti ArdLAB spremnik da služi kamери i API-ima za programiranje

```
docker-compose up --detach
```

Spremnik će se ponovno pokrenuti kada se pokrene Raspberry PI ili ako se web server spremnika sruši.

- Ako trebate zaustaviti spremnik, koristite naredbu:

```
docker-compose down
```

- Ako u budućnosti trebate ažurirati spremnik, trebate postupiti kako slijedi (nakon zaustavljanja spremnika):

```
docker-compose pull
```

Ažuriranje je tiho i može potrajati nekoliko minuta. Nakon ažuriranja spremnika, morate ponovno pokrenuti spremnik upotrebom prve naredbe u ovom poglavlju.

Možete provjeriti radi li ArdLAB pristupajući sljedećoj poveznici:

<https://ArdLAB-DNS-Name:7575>