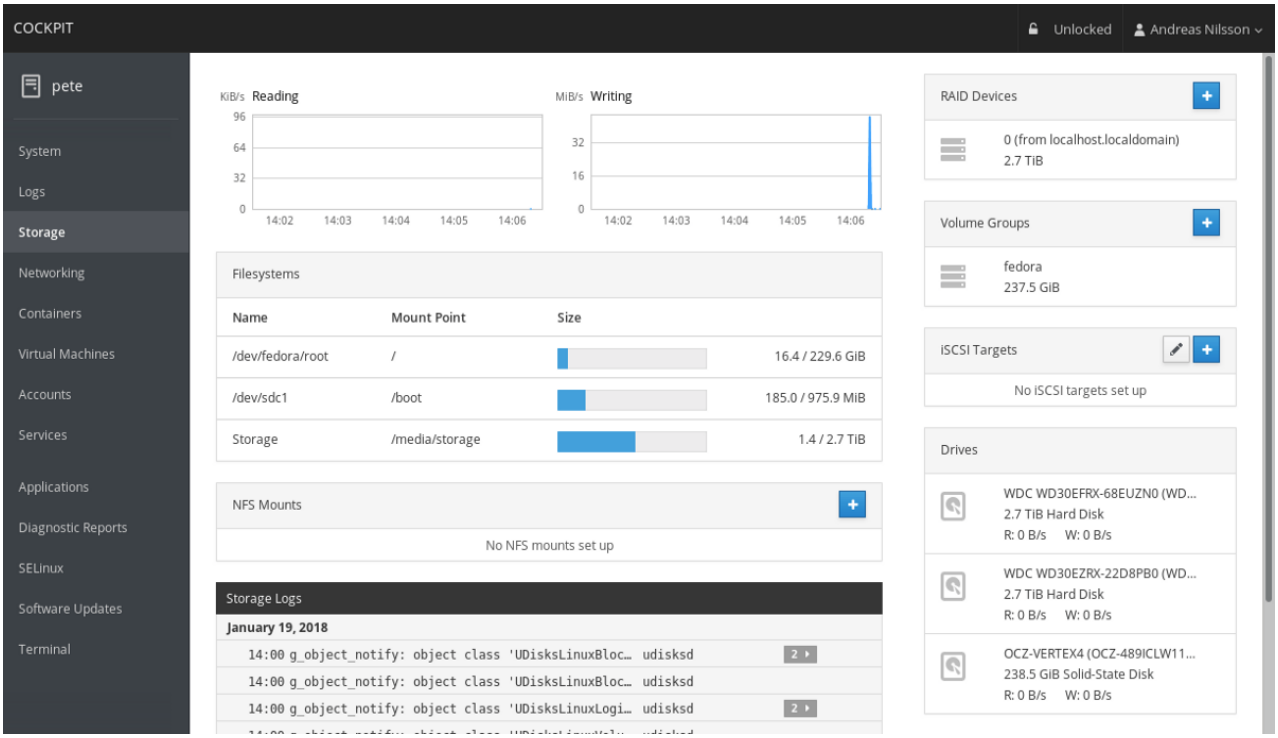
Server setup Part-3: Cockpit Setup with SSL Certificates

szewong.medium.com/server-setup-part-3-cockpit-setup-with-ssl-certificates-2698f1ce2986

10 April 2020

Apr 10, 2020



Cockpit is a very powerful server admin tool. It allows you to monitor server vitals including basic info like CPU, I/O, Memory, etc. You can also see server logs and it even comes with a web-based terminal.

Since it's powerful, we don't want it to be open to the world to see. So we will update ufw accordingly.

Before we start, you should find out what your public IP address is. Simply do a google search on "what's my ip" will give you the answer.

Install cockpit:

```
sudo apt install cockpit -y
```

It's now running and listening to port 9090

Update ufw to allow port 9090 from your IP address

```
sudo ufw allow from [your ip address] to any port 9090
```

You can add additional rules to allow for more IP addresses.

Now you can try going to https://[server IP address]:9090

If it works, login with your username and password.

The following section talks about how to setup your SSL certificate for cockpit.

Cockpit automatically setup a self-signed certificate for SSL. However, if your server is connecting to a DNS name, you may want to use the real SSL so you can connect to it using your domain name rather than IP address.

In general, I am in favor of using real SSL certificates from CAs. As most browsers require https now, SSL certificates have gone a lot cheaper. You should be able to get a wildcard cert for under $10.00/year.

Before we begin, go to your domain name provider and download the SSL certificate, any intermediate certificate and the corresponding private key.

You should have at least 2 files. One or more for certificates and one for the private key.

Open the files in a text editor and you should see the following:

```
-----BEGIN CERTIFICATE-----
d3cuZGlnaWNlcnQuY29tMS0wKwYDVQQDEyRFbmNyeXB0aW9uIEV2ZXJ5d2hlcmUgRFYgVExTIENBIC0gRzE
----END CERTIFICATE-----
```

Your Private key should look like this:

```
-----BEGIN RSA PRIVATE KEY-----
WhYsY/KdAc4Jm+F2ejNOsP6+AoEh2cIIl44WciKDJiuWmNfn4Br7oKIFeWiqzXNh...OcSOb3aexMXMYsNm
----END RSA PRIVATE KEY-----
```

Now create a new text file with all the certificates and private key stack up like this:

```
<Main Cert><Intermediate Cert><Private Key>
```

So your file should look something like:

```
-----BEGIN CERTIFICATE-----
d3cuZGlnaWNlcnQuY29tMS0wKwYDVQQDEyRFbmNyeXB0aW9uIEV2ZXJ5d2hlcmUgRFYgVExTIENBIC0gRzE
----END CERTIFICATE----------BEGIN CERTIFICATE-----
sl7m9scOygAAAXD+2f60AAAEAwBHMEUCIQDAIUHE/pXQReb3xIhzRV1jaTNbPxop...kyMIvB024DelLwIg
----END CERTIFICATE----------BEGIN RSA PRIVATE KEY-----
WhYsY/KdAc4Jm+F2ejNOsP6+AoEh2cIIl44WciKDJiuWmNfn4Br7oKIFeWiqzXNh...OcSOb3aexMXMYsNm
----END RSA PRIVATE KEY-----
```

Save this file as ssl.cert

This file needs to make its way to the server

Since this is a text file, one simple way is to just copy-n-paste the content.

SSH into the server

```
cd /etc/cockpit/ws-certs.dvi ssl.cert
```

Copy the content of the file and save the file.

Restart Cockpit

```
sudo systemctl start cockpit
```

Double-check to see if Cockpit is using the correct certificate:

```
sudo remotectl certificate
```

That's it. Now from your local machine you can run

https://[server name]:9090

This is part of a 3-part series of setting up a basic Ubuntu server.

Part-1: ufw and ssh

Part-2: Server update and timezone

Part-3: Cockpit