Quick Setup on Digital Ocean

*You will probably often will find yourself in the situation that you will have to setup a new Ubuntu Server. This repeats all steps from a number of different tutorials, but without most of the explaining text to make if faster to use.*

*Linux Help:* [*20 Top Most Used & Common SSH Commands*](http://www.servermom.org/top-most-used-common-ssh-commands/)

Node.js/Express deployment tutorial here: <https://www.youtube.com/playlist?list=PLDbigcKhXkiW3w8RQ25QRwJD5OhuTj8HU>

## Create an Ubuntu 18.04 droplet with a non-root user

1. If not already done, create a new Ubuntu 18.04 (remember to set 5$) Droplet on DigitalOcean.
2. Select Frankfurt as *datacenter region*
3. If you already have a key select it from the tick box otherwise copy your public key into the clipboard. Select "New SSH Key" and paste your key into the TextArea.
4. Initial Server Setup (Replace text in **red** below with your own values)

*The following is a shortened version of this* [*document*](https://www.digitalocean.com/community/tutorials/initial-server-setup-with-ubuntu-18-04)

Login as root: ssh root@**your\_server\_ip**

Create a new user: adduser **sammy**

Grant administrative privileges to the new user: usermod -aG sudo **sammy**

Copy your local keys from your **root** account into the account for the new user (details [here](https://www.digitalocean.com/community/tutorials/initial-server-setup-with-ubuntu-18-04#if-the-root-account-uses-ssh-key-authentication), if you like):

rsync --archive --chown=**sammy**:**sammy** ~/.ssh /home/**sammy**

## 

## MySQL, Java and Tomcat

The idea is to have a similar setup for all students/classes, so you will do this using a script which also speeds up the time used for the process to less than 10 minutes. We will use the script, **bootstrap.sh** from the instructions given in [Install Virtual Box and Vagrant](https://docs.google.com/document/d/1rZcZK6Tlhh7NFCG0aZcebr0QID_iBKjawoZ9Crccq6A/edit#bookmark=id.vpcsru8co3l0), which were used by vagrant to set up your local guest OS.

1) Create a text-file somewhere (which you can find again) on your local computer called **my-bootstrap.sh**

2) Locate the file **bootstrap.sh** in the folder where you do **vagrant up** (see document linked above). Copy everything from this file into **my-bootstrap.sh** (make sure not to change anything in the vagrant-folder).

3) Open the file **my-bootstrap.sh** in a txt-editor.

Locate the part in the top of the script that defines these values:

#User and Password for the dev-user

DB\_PW="ax2"

DB\_USER\_NAME="dev"

# Password for the hardcoded user: gui\_user

MANAGER\_GUI\_PW="a1234"

#Password for the hardcoded user: script\_user

MANAGER\_SCRIPT\_PW="a1234"

Here you see the database username and password we have used for local development. The MANAGER\_GUI\_PW is the password used to upload projects via Tomcats Manager Interface and MANAGER\_GUI\_PW is the password for the user *script\_user,* used by maven to upload projects.

For local development, it's not a problem that everyone knows these credentials, but for a public server, it would be a SERIOUS PROBLEM.

Change the four values marked above into something only you know. If you plan to use your server for something just a little bit serious, choose strong passwords. Make sure to remember these values (or the location of my-bootstrap.sh, and don’t share it), we can’t help you here.

4) Now SSH into your server, using your non-root user and complete the following steps

Type: sudo nano my-bootstrap.sh (this will create my-bootstrap.sh on your droplet)

Paste everything from your local version of this file into this file in nano . Close the file again.

Now type:

sudo chmod +x my-bootstrap.sh (give executable rights to the file)

sudo ./my-bootstrap.sh (will execute the script)

This will probably take a few minutes, but when complete you will have Java, Tomcat 9 and MySQL installed on your droplet.

### Verify that servers are running

#### Tomcat

1) Open a browser and enter this url: [http://DROPLET-IP:8080](http://droplet-ip)

You should see the default tomcat page on your server

2) Enter this URL in the browser: [http://DROPLET-IP:8080/manager/html](http://droplet-ip/manager/html)

Enter the user name: gui\_user and the password you selected for MANAGER\_GUI\_PW and you should see the usual Tomcat Web Application Manager

*IMPORTANT!: Do NOT try to change Tomcat to use port 80. Later in this document (topic for week 2), you will see how to change the port number into 80 using a Reverse Proxy and also how to set up your own domain name, and a certificate to use HTTPS.*

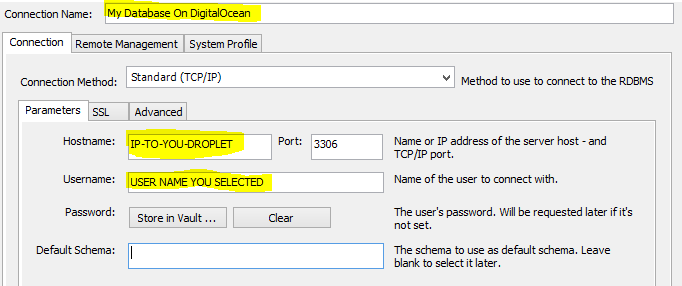
#### MySQL



Open MySQL Workbench on your laptop, and create a new Connection

Add the following values in the window that pops up.

Use the password you created for DB\_PW



Press “Test Connection” and when this work press OK.

IMPORTANT: If you got here, while preparing for day-1 **you are done**.

The remaining parts of this document are meant for week 2

## Set up your Domain Name with DigitalOcean

Ref: <https://www.digitalocean.com/community/tutorials/how-to-set-up-a-host-name-with-digitalocean>

#### Prerequisites

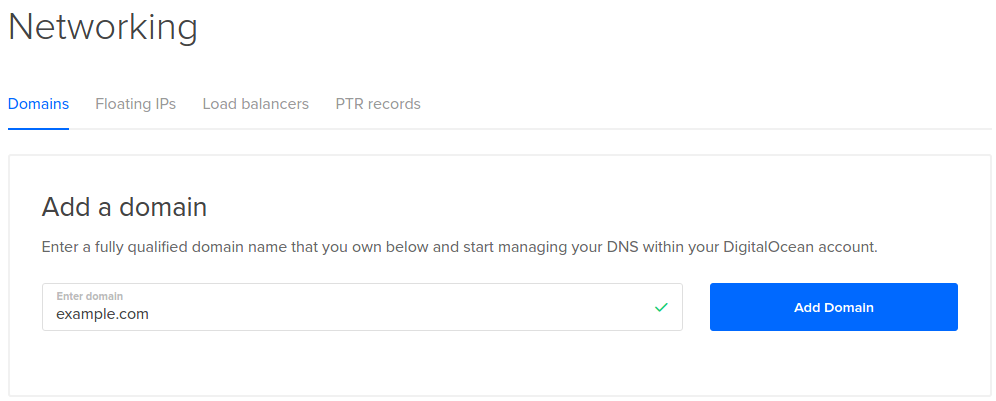
* A droplet with a non-root sudo user + Tomcat installed as described earlier in this document.
* You must have purchased a Registered Domain Name (i.e. mycoolDomainName.dk).
* You can do this many places. For a dk-domain, this is just one of many options: <https://www.dandomain.dk/domain/domain-soeg>

You only need to buy the domain name, NOT a web-hotel, we are using Digital Ocean.

#### Configuring your Domain Part-1

Now we need move into the DigitalOcean control panel.

Within the Networking section, in the Add a domain section, fill in your domain name. Click on the Add Domain button to add the domain. Note: The domain name should not have a "www" at the beginning.



You will reach a page where you can enter all of your site details. We will come back to this in “Configuring your Domain Part-2”, but for now, continue to the next step.

#### Change your Domain Server

See the [original tutorial](https://www.digitalocean.com/community/tutorials/how-to-set-up-a-host-name-with-digitalocean) for “general info” about this, but if you have bought a .dk domain, you do this via this URL: [https://selvbetjening.dk-hostmaster.dk/domæne](https://selvbetjening.dk-hostmaster.dk/dom%C3%A6ne)

If you are not using a .dk domain use this [original tutorial](https://www.digitalocean.com/docs/networking/dns/quickstart/), which also provide a link to specific instructions for many of the popular Common Domain Registrars (like GoDaddy)

#### The following will assume a .dk-domain which are all controlled by dk-hostmaster.dk

Log-in with the credentials you received when you bought the domain-name. Find the menu entry “Redelegate (change nameserver)” and add ns1.digitalocean.com as the hostname.

Pres “Continue” and if you see this for *New name servers*, press “Confirm”:

* ns1.digitalocean.com
* ns2.digitalocean.com
* ns3.digitalocean.com

If your domain registrar has verified the three name servers above, everything should be ok.

Open a browser at type this URL: <https://www.whois.com/whois/> or a terminal and type:

nslookup yourdomainname

Enter your domain name “mycoolDomainName.dk” and check the name servers. They are most likely not changed yet, it may take an hour or two for the changes to be reflected on your site.

#### Configuring your Domain Part-2

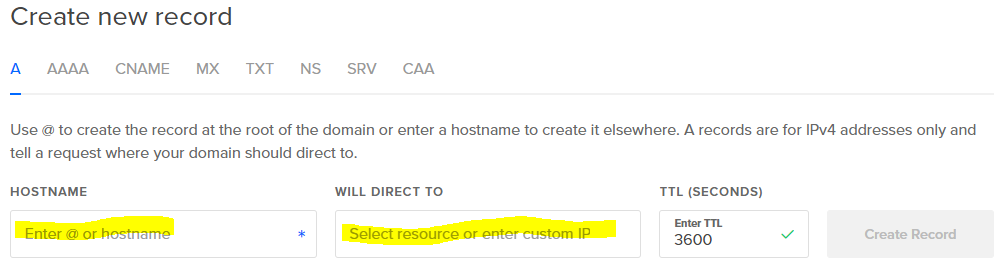
In your browser, navigate back into the DigitalOcean control panel and the Networking section (the one you used in “configuring your Domain Part-1”)

For this part we will add two A Records so your webserver (Tomcat) on your droplet can be accessed by:

* http:// mycoolDomainName.dk or
* http://www.mycoolDomainName.dk

In the section “Create new record” select A Record and enter the following information:

|  |  |  |
| --- | --- | --- |
| Hostname (enter text in bold) | Will Direct to | Press |
| **@** (for http:// mycoolDomainName.dk) | IP for your droplet | Create Record |
| **www** (for http:// www.mycoolDomainName.dk) | Same as above | Create Record |



Verify that you can access your web-server via (8080 only if you have not set up your system to handle port 80):

* http:// mycoolDomainName.dk:8080 or
* http://www.mycoolDomainName.dk:8080

## Installing Nginx

Before you start, make sure you know the purpose(s) of a [Reverse Proxy](https://en.wikipedia.org/wiki/Reverse_proxy), and more specific our purpose.

Prerequisites: A droplet with a non-root sudo user, setup to use you own domain name, + Tomcat installed as described earlier in this document.

**Important:** Make sure you have not followed a tutorial that sets up Tomcat to use port 80, this will be handled by Nginx. This tutorial assumes your Tomcat is running on 8080.

Ref: https://www.digitalocean.com/community/tutorials/how-to-install-nginx-on-ubuntu-16-04

sudo apt update  
sudo apt install nginx

Verify that your firewall is disabled (see original reference to turn it on, but leave it disabled for now)

#### Check that your Nginx Web Server is Running

Check with the systemd init system to make sure the service is running by typing:

systemctl status nginx

Also check via a browser, by typing your domain name (replace domain name): [http://mycoolDomainName.dk](http://mycooldomainname.dk)

This should show the default Nginx landing page.

#### Change the size of documents that can be uploaded via Nginx

Since we are going to upload war-files, you will probably quickly find yourself in a situation where the default restriction placed on files uploads won’t work.

Open this file: **/etc/nginx/nginx.conf** (with nano) and add the following addition(red) to the http-section:

http {

**client\_max\_body\_size 50M;**

…

}

#### Manage The Nginx Process

See the Original Reference for [how to Manage Nginx](https://www.digitalocean.com/community/tutorials/how-to-install-nginx-on-ubuntu-16-04#step-4-manage-the-nginx-process):

#### Get Familiar with Important Nginx Files and Directories

See the Original Reference to find important [Nginx-files and Directories](https://www.digitalocean.com/community/tutorials/how-to-install-nginx-on-ubuntu-16-04#step-5-get-familiar-with-important-nginx-files-and-directories)

#### 

#### 

#### Using Nginx to proxy forward Requests

By now you should have two web-servers running on your droplet (make sure you understand why we do this):

Nginx, listening on port 80: myCoolDomainName.dk

Tomcat, listening on port 8080: myCoolDomainName.dk:8080

Edit Nginx's default server block configuration (/etc/nginx/sites-enabled/default) with nano like this:

sudo nano /etc/nginx/sites-enabled/default

Inside, towards the top of the file, we need to add an upstream block. This will outline the connection details so that Nginx knows where our Tomcat server is listening. Place this outside of any of the server blocks defined within the file:

upstream tomcat {  
 server 127.0.0.1:8080 fail\_timeout=0;  
}

Next, within the server block defined for port 80 ~~443~~, modify the location / block. We want to pass all requests directly to the upstream block we just defined. Comment out the current contents and use the proxy\_pass directive to pass to the "tomcat" upstream we just defined.  
We will also need to include the proxy\_params configuration within this block. This file defines many of the details of how Nginx will proxy the connection:

Also change server\_name (the \_ underscore) with your domain name(s)

server {  
 . . .  
 server\_name myCoolDomainName.dk www.myCoolDomainName.dk;  
 location / {  
 **#**try\_files $uri $uri/ =404;  
 include proxy\_params;

proxy\_pass http://tomcat/;  
 }  
  
 . . .  
}

Red above, is what you have to add.

Remember, whenever you change the config-files, to restart nginx: service nginx restart

Verify that access to the default port will be forwarded to Tomcat.

myCoolDomainName.dk → Shows your Tomcat Server default page.

Please note that Tomcat still listens for external connections on TCP port 8080. Thus, Nginx and its security measures can be easily bypassed. To resolve this problem configure Tomcat to listen on the local interface 127.0.0.1 only. For this purpose open the file /opt/tomcat/conf/server.xml with nano:  
  
sudo nano /opt/tomcat/conf/server.xml

Add address="127.0.0.1" in the Connector configuration part (the Connector NOT commented out) like this:

...  
<Connector address="127.0.0.1" port="8080" protocol="HTTP/1.1"  
 connectionTimeout="20000"  
 URIEncoding="UTF-8"  
 redirectPort="8443" />  
...

Restart Tomcat: sudo service tomcat restart

Verify that you can no longer access Tomcat from “outside” on port 8080.

### Nginx Hints

Remember, after each change in the Nginx-configuration files to restart Nginx:

sudo service nginx restart

If you have any problems, try to stop nginx (if it's running) and type:

sudo nginx -t

This will not start nginx, but parse all config-files

## Setting up your Droplet with SSL

A droplet with Tomcat and Nginx setup as outlined in the previous sections. Most notably, your droplet must be accessible via your domain name (server\_name must be set with your domain name as described in the previous section).

#### Installing and setup a Let’s Encrypt Certificate

Follow the instructions (VERY CLOSELY) in [this document](https://certbot.eff.org/), (select Nginx, and Ubuntu 18.04) but first observe that there are some IMPORTANT additional info below:

For the step: **sudo certbot --nginx**

When you get to this part you will be asked a number of questions:

* Your email (provide a valid one)
* A list of (your) domain names to activate HTTPS for?
* Whether or not to redirect HTTP traffic to HTTPS, removing HTTP access completely (select redirect)

## Automatic renewal

There is nothing to do here. Certbot installed a cron task to automatically renew certificates about to expire.

You can [check renewal works](https://certbot.eff.org/docs/using.html#re-creating-and-updating-existing-certificates) using:

sudo certbot renew --dry-run

You can also [check what certificates exist](https://certbot.eff.org/docs/using.html#managing-certificates) using:

sudo certbot certificates

## Test

1. Verify that you can ONLY access your web-site via SSL (https)

#### Test whether you have got (what you hopefully have) an A-rating for your SSL. [SSL Server Test](https://www.ssllabs.com/ssltest/).