

# Quick Setup on Digital Ocean - Spring 2019

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

Linux Help: [20 Top Most Used & Common SSH Commands](#)

## Create a Droplet with a non-root user

1. Create a new Ubuntu Droplet on DigitalOcean.
2. Select Frankfurt as *datacenter region*
3. Copy your public key into the clipboard. Select "New SSH Key" and paste your key into the TextArea. If you have done this before, you probably already have uploaded a key. Select this key

Initial Server Setup (Replace text in **red** below with your own values)

- Open a batch-terminal (git bash on Windows)
- Log into your server: `ssh root@SERVER_IP_ADDRESS`
- Create a New User: `adduser sammy` (answer questions, starting with the account password)

Add Root Privileges to the new user

As root, run this to add your new user to the *sudo* group: `usermod -aG sudo sammy`

Add Public Key Authentication for the New User (Install the key)

*On your laptop:* Copy the key into the clipboard (start Git Gui, press help->Show SSH key)

*On the server:* as the **root** user, enter the following command to temporarily switch to the new user

```
su - Sammy (now you will be in your new user's home directory)
```

Create a new directory called `.ssh` and restrict its permissions with the following commands:

```
mkdir ~/.ssh
chmod 700 ~/.ssh
```

(Hint: 4 = read, 2 = write, 1 = execute, 4+2+1 = read+write+execute)

Now open a file in `.ssh` called `authorized_keys` with the nano-text editor:

```
nano ~/.ssh/authorized_keys
```

- Paste your public key (which should be in your clipboard) into the editor.
- Hit CTRL-x to exit the file, then y to save the changes that you made, and then ENTER to confirm

Restrict the permissions of the `authorized_keys` file: `chmod 600 ~/.ssh/authorized_keys`

(Hint: 4 = read, 2 = write, 4+2 = read+write)

Type `exit` to return to the root user:

## Test Log In

In a new terminal, log in to your server using the new account: `ssh sammy@SERVER_IP_ADDRESS`

## Swap-file

If you are using a single Droplet for everything (Tomcat, without adjusting java's memory properties, MySQL, NginX etc) you will probably find, from time to time, that your server "dies". If the log-files reveal that you have a memory problem, setting up a swap-file has proved itself as a solution for many students. Use this document as a reference for how to complete this step.

<https://www.digitalocean.com/community/tutorials/how-to-add-swap-space-on-ubuntu-16-04>

## MySQL

This document will not focus on MySQL, mainly because the "best" way, is to install MySQL is on a separate server (droplet) and make sure it's accessible from your droplet with Tomcat and Nginx.

If you only feel like paying for one droplet (which is OK, especially if you are using a swap-file), install MySQL now, using instructions given elsewhere.

# Setup Tomcat

The following description is a simplified way of setting up Tomcat, compared to this document:  
<https://www.digitalocean.com/community/tutorials/how-to-install-apache-tomcat-8-on-ubuntu-16-04>

Logon to your Droplet, using a non-root user with sudo-privileges

## Install Java

```
sudo apt update (apt-get update && apt-get upgrade)
sudo apt install default-jdk
```

## Install Tomcat

- `apt-get install tomcat8 tomcat8-admin`
- `apt-get install havged`

----- Info From the Tomcat Install Process -----

Creating config file /etc/default/tomcat8 with new version

Adding system user `tomcat8' (UID 112) ...

Adding new user `tomcat8' (UID 112) with group `tomcat8' ...

|-----

## Create an admin user for Tomcat, and assign the necessary roles<sup>1</sup>

1. Open `/etc/tomcat8/tomcat-users.xml` with nano as given below:

```
sudo nano /etc/tomcat8/tomcat-users.xml
```

2. Insert the following after line 21 in the file – *please use your own password!*

```
<role rolename="manager-gui"/>
<role rolename="manager-script"/>
<user name="admin" password="xxx" roles="manager-gui,manager-script"/>
```

3. Save the file

4. Restart the Tomcat Server: `service tomcat8 restart`

Set READ-WRITE ACCESS to the logs folder (symlink in `/var/lib/tomcat8/logs`)

```
sudo setfacl -m u:XXX:rwX /var/lib/tomcat8/logs
```

Replace “XXX” with your own user name.

Test that you can access your Tomcat Installation on port 8080

**Important:** Don't use the “fix” given in earlier descriptions to run Tomcat on Port 80. For now, just use port 8080. Later in this document, we will use Nginx as a reverse proxy that will fix this (and other) problems.

---

<sup>1</sup> *The `manager-gui` role will allow you to add war-files using Tomcat's “Web Application Manager”. The `manager-script` role will allow you to upload war-files using Maven (required for the 3. semester)*

## Before you continue, we (very much) advise you to create a Snapshot

If you have found that you have to set up your droplet(s) over and over, now it's time to consider spending an extra (max) 1 \$ pr. month.

If you have installed everything as explained above, taking a snapshot of what you have done, will allow you to spin up a new droplet from the snapshot in just a few seconds (how cool is that ;-)

*The "cost for snapshots is \$0.05 per gigabyte per month, based on the amount of utilized space within the filesystem. Snapshots that are destroyed before the month is over will be charged hourly, just like Droplets".*

If you are using a 5\$ droplet this would give a max price of 1\$ since you get a maximum of 20Gb for a 5\$ droplet.

If you want to be on the safe side, setup DigitalOcean to provide you with a warning if the monthly price goes over a threshold you have decided.

See here, for all the benefits you get from snapshots (actually they can save you money):

<https://www.digitalocean.com/community/tutorials/digitalocean-backups-and-snapshots-explained#snapshot-uses>

# Set up your Host Name with DigitalOcean

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

## Prerequisites

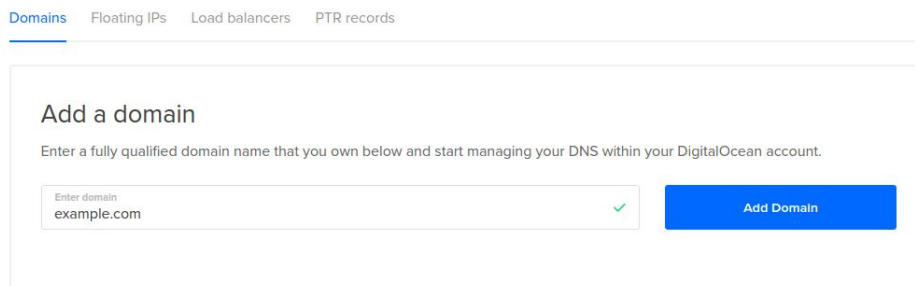
- A droplet with a non-root sudo user + Tomcat installed as described earlier in this document.
- You must have purchased a Registered Domain Name (i.e. mycoolDomainName.dk).
- You can do this many places. For a dk-domain, this is just one of many options:  
<https://www.dandomain.dk/domain/domain-soeg>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

## Networking



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](#) 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>

If you are not using a .dk domain use this [original tutorial](#), 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](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
<b>@</b> (for <a href="http://mycoolDomainName.dk">http:// mycoolDomainName.dk</a> )	IP for your droplet	Create Record
<b>www</b> (for <a href="http://www.mycoolDomainName.dk">http:// www.mycoolDomainName.dk</a> )	Same as above	Create Record

### Create new record

[A](#) AAAA CNAME MX TXT NS SRV CAA

Use @ to create the record at the root of the domain or enter a hostname to create it elsewhere. A records are for IPv4 addresses only and tell a request where your domain should direct to.

HOSTNAME	WILL DIRECT TO	TTL (SECONDS)	
<input type="text" value="Enter @ or hostname"/>	<input type="text" value="Select resource or enter custom IP"/>	<input type="text" value="Enter TTL 3600"/>	<input type="button" value="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](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](#), 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-get update
sudo apt-get 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>

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 40M;
    ...
}
```

## Manage The Nginx Process

See the Original Reference for [how to Manage Nginx](#):

## Get Familiar with Important Nginx Files and Directories

See the Original Reference to find 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 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 /etc/tomcat7/server.xml with nano:

```
sudo nano /etc/tomcat8/server.xml
```

Add address="127.0.0.1" in the Connector configuration part 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 tomcat8 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](#), 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](#) using:

```
sudo certbot renew --dry-run
```

You can also [check what certificates exist](#) using:

```
sudo certbot certificates
```

## Test

1. Verify that you can ONLY access your web-site via SSL (https)
2. Test whether you have got (what you hopefully have) an A-rating for your SSL.

[SSL Server Test](#).