

# Setting up **Microsoft** SQL Server on Ubuntu using **DigitalOcean**



Joshua Marango

Sep 24, 2018 · 4 min read

by Joshua Marango

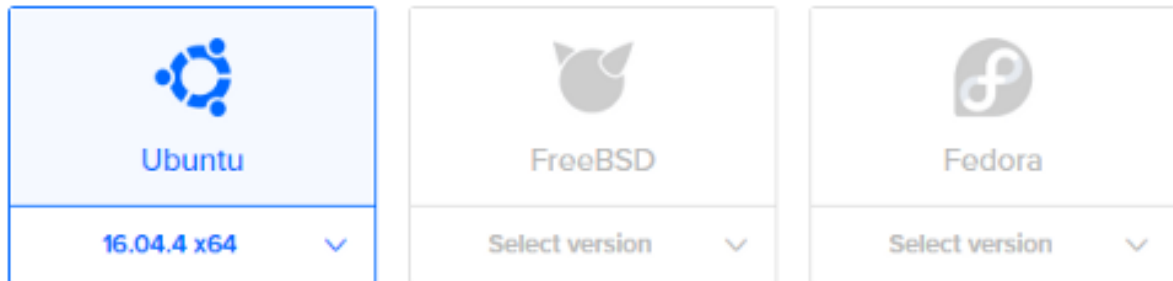
There aren't too many straight-forward tutorials on how to setup a MSSQL database on a machine that is not in the Azure ecosystem. Today I'll show you how easy and straight forward it is, so let's jump right in.

A few things you will need before starting this tutorial:

- **Ubuntu 16.04** — I recommend signing up to **DigitalOcean**, you get **\$100 free credit** so sign up **here** and I'll show you how to create an Ubuntu machine, otherwise you can use docker (locally or cloud based) or any other cloud service you may have.
- **Download SMSS** — this is the GUI we will use to interact with our database(s) as well as writing and performing queries using T-SQL. You can go ahead and download **here** (it's free!).
- **Gitbash** — we'll be using this CMD tool to SSH into our Ubuntu machine. Go ahead and download that **here** (it's free!).

Once you have these three setup we're ready to start. I'll be using [DigitalOcean](#), so make sure sign up and login if you'll be using it too.

## Step 1. Create Ubuntu VM



## Choose a size

### Standard Droplets

Balanced virtual machines with a healthy amount of memory tuned to host and scale applications like blogs, web applications, testing / staging environments, in-memory caching and databases.

MEMORY	vCPUs	SSD DISK	TRANSFER	PRICE
1 GB	1 vCPU	25 GB	1 TB	\$5/mo \$0.007/hr
2 GB	1 vCPU	50 GB	2 TB	\$10/mo \$0.015/hr
3 GB	1 vCPU	60 GB	3 TB	\$15/mo \$0.022/hr
2 GB	2 vCPUs	60 GB	3 TB	\$15/mo \$0.022/hr
1 GB	3 vCPUs	60 GB	3 TB	\$15/mo \$0.022/hr
4 GB	2 vCPUs	80 GB	4 TB	\$20/mo \$0.030/hr

Once you're signed in to your DigitalOcean account, begin by creating a new droplet. Under distributions, click on the Ubuntu image and make sure you select the “**16.04 x64**” option as seen below.

The *minimum requirements for setting up a MSSQL server is 4GB ram*, so select the \$20/mo option.

. . .

## Step 1b. Adding more storage (optional)

*If you would like to add more storage to your image then go ahead and read on, otherwise go to step 2.*

### Add backups

Automatic system-level backups. Use the backup images to revert the server or create new Droplets. Backups cost 20% of the Droplet price.

Enable Backups

### Add block storage

Currently only available in AMS3, BLR1, FRA1, LON1, NYC1, NYC3, SFO2, SGP1 and TOR1.

Block storage lets you add independent storage volumes that can be accessed like local disk and moved from one Droplet to another within the same region.

\$ <input type="text"/> /mo \$ <input type="text"/> /hour	\$10 /mo \$0.015 /hour	\$25 /mo \$0.037 /hour	\$50 /mo \$0.074 /hour	\$100 /mo \$0.149 /hour	\$200 /mo \$0.298 /hour
Enter size in GB	100 GB	250 GB	500 GB	1000 GB	2000 GB

### Choose configuration options:



#### Automatically Format & Mount

We will choose the appropriate default configurations. These settings can be changed later via ssh.



#### Manually Format & Mount

We will still attach the volume. You can then manually format and mount the volume.

Choose a filesystem ☒ Ext4 ☐ XFS

Remove Volume

You could alternatively enter a custom amount of block storage you need.

Begin by selecting the amount of storage you wish to add. If you're new to setting up VMs I recommend selecting “Automatically Format & Mount”. That's it! you're all done, you've just added the extra space you need.

*Note: because DigitalOcean are so amazing, you can also enable backups, which will only cost you 20% of the droplet price. So a droplet at \$45/mo could be backed up for only*

*\$9/mo!!!! Crazy right?*

. . .

## Step 2. Setting up SSH keys

Open up your git bash terminal, then key in the following commands:

Note: if you already have initialised an SSH key then go to the retrieving your SSH key section in this step. If you overwrite your key, any service connected to that key (other VMs, gitlab etc.) will be lost and cannot be undone, you have been warned.

Git Bash on Windows / GNU/Linux / macOS:

```
ssh-keygen -o -t rsa -C "your.email@example.com" -b 4096
```

### Retrieving your SSH key

Press enter on everything, create a password if you wish or if your laptop/desktop is used by multiple people. Once that's done go ahead and copy your ssh key by running this command:

**macOS:**

```
pbcopy < ~/.ssh/id_rsa.pub
```

**GNU/Linux (requires the xclip package):**

```
xclip -sel clip < ~/.ssh/id_rsa.pub
```

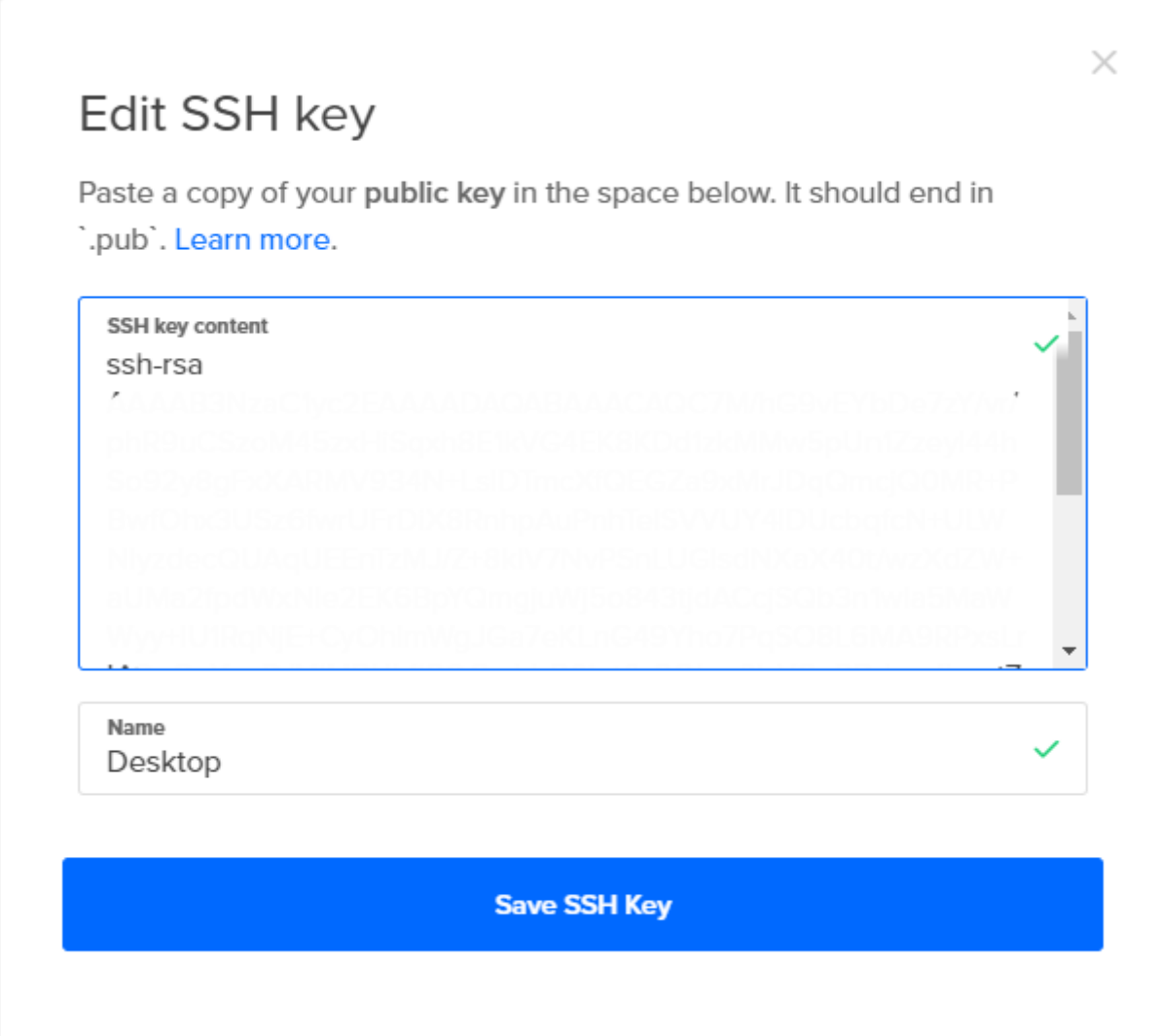
**Windows Command Line:**

```
type %userprofile%\ssh\id_rsa.pub | clip
```

## Git Bash on Windows / Windows PowerShell:

```
cat ~/.ssh/id_rsa.pub
```

Back to our [DigitalOcean](#) page, we can now add our new (or existing) SSH key to our DigitalOcean Ubuntu image as shown below.

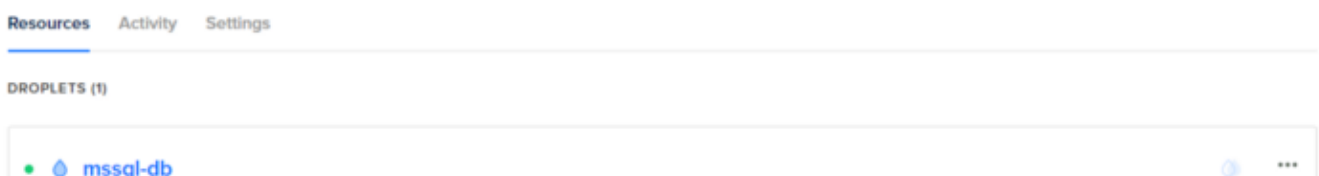


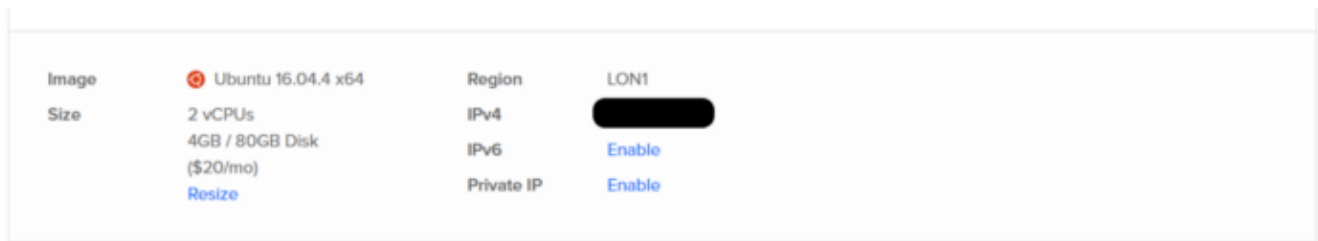
Make sure to give your SSH key name a memorable name.

. . .

## Step 3. Installing Microsoft SQL Server

Your new Ubuntu image should look similar to this one:





Scaling? You can always resize your droplet as needed.

Now we're going to SSH into our machine and begin installing MSSQL Server.

Open up the Git terminal and enter the following:

```
ssh root@{copy your IP Address here}
```

Congratulations you're now in. Now complete the following steps:

### Enter superuser mode.

```
sudo su
```

### Import the public repository GPG keys:

```
curl https://packages.microsoft.com/keys/microsoft.asc | apt-key add -
```

### Exit superuser mode.

```
exit
```

### Run the following commands to install SQL Server:

```
sudo apt-get update  
sudo apt-get install -y mssql-server
```

**After the package installation finishes, run the configuration script and follow the prompts.**

```
sudo /opt/mssql/bin/sqlservr-setup
```

**Once the configuration is done, verify that the service is running:**

```
systemctl status mssql-server
```

. . .

#### **Step 4. Opening up MSSQL Server in SSMS**

Go ahead and open up SSMS. In the login window copy your IP address, in the user enter “SA” (without quotation marks) and then enter the password configured in the installing of MSSQL in Step 3.



SQL Server Management Studio

## Congratulations, you've now installed MSSQL Server on an Ubuntu machine 😊

In the next short tutorial I'll be showing you how to ditch T-SQL (soft of) and start using Nodejs to create tables, drop tables, data types, data models and so much more!

*Let me know, would you guys like me to supplement these short tutorials with videos?*

[Deployment](#) [Ubuntu](#) [DevOps](#) [Microsoft](#) [Database](#)

[About](#) [Help](#) [Legal](#)

Get the Medium app

