Installing Jenkins on Ubuntu

> **Out of date**
>
> **This content is out of date. See the Ubuntu installation section of the handbook on jenkins.io for Debian and Ubuntu installation instructions**

On Debian-based distributions, such as Ubuntu, you can install Jenkins through `apt-get`.

Recent versions are available in an apt repository. Older but stable LTS versions are in this apt repository.

You need to have a JDK and JRE installed. openjdk-7-jre and openjdk-7-jdk are suggested. As of 2011-08 gcj is known to be problematic - see https://issues.jenkins-ci.org/browse/JENKINS-743.

Please make sure to back up any current Hudson or Jenkins files you may have.

Installation

```
wget -q -O - https://pkg.jenkins.io/debian/jenkins-ci.org.key | sudo
apt-key add -
sudo sh -c 'echo deb http://pkg.jenkins.io/debian-stable binary/ > /etc
/apt/sources.list.d/jenkins.list'
sudo apt-get update
sudo apt-get install jenkins
```

Upgrade

Once installed like this, you can update to the later version of Jenkins (when it comes out) by running the following commands:

```
sudo apt-get update
sudo apt-get install jenkins
```

(aptitude or apt-get doesn't make any difference.)

What does this package do?

- Jenkins will be launched as a daemon up on start. See `/etc/init.d/jenkins` for more details.
- The '`jenkins`' user is created to run this service.
- Log file will be placed in `/var/log/jenkins/jenkins.log`. Check this file if you are troubleshooting Jenkins.
- `/etc/default/jenkins` will capture configuration parameters for the launch like e.g JENKINS_HOME
- By default, Jenkins listen on port 8080. Access this port with your browser to start configuration.

> If your `/etc/init.d/jenkins` file fails to start jenkins, edit the `/etc/default/jenkins` to replace the line
>
> ```
> HTTP_PORT=8080
> ```
>
> by

```
HTTP_PORT=8081
```

Here, 8081 was chosen but you can put another port available.

Deploying on Ubuntu in a cloud (EC2, HP Cloud, OpenStack)

The Ubuntu Jenkins maintainer also maintains the Juju charm deployment/management script for deployment in clouds. It's designed to make it easy to deploy a master with multiple slaves:

```
juju deploy jenkins
juju deploy -n 5 jenkins-slave
juju add-relation jenkins jenkins-slave
```

The default password for the 'admin' account will be auto-generated. You can set it using:

```
juju set jenkins password=mypassword
```

Always change it this way - this account is used by the charm to manage slave configuration. Then feel free to expose your jenkins master:

```
juju expose jenkins
```

Using Linux iptables for port 80 -> 8080

- This enables port forwarding of traffic between ports 80 and 8080. You can keep Jenkins on the default port 8080 and access it with a normal url without installing anything extra.
- `sudo nano /etc/rc.local`
- Then add the following just **before** the exit 0

```
#Requests from outside
iptables -t nat -A PREROUTING -p tcp --dport 80 -j REDIRECT --to-ports
8080
#Requests from localhost
iptables -t nat -I OUTPUT -p tcp -d 127.0.0.1 --dport 80 -j REDIRECT --
to-ports 8080
```

Now reboot or run `sudo /etc/rc.local` to enable port forwarding. Additional info: https://gist.github.com/m5m1th /6870a54717c0387468c3

Setting up an Apache Proxy for port 80 -> 8080

- This configuration will setup Apache2 to proxy port 80 to 8080 so that you can keep Jenkins on 8080.
- `sudo aptitude install apache2`
- `sudo a2enmod proxy`
- `sudo a2enmod proxy_http`

> do not do this next command if you already have virtual hosting setup that depends on the default site. See my comment below - danapsimer

- `sudo a2dissite default`

> If you get `ERROR: Site default does not exist!` then try this instead:
>
> - `sudo a2dissite 000-default`
>
> And if all else fails just have a look if there is a default site set up at all:
>
> - `ls /etc/apache2/sites-enabled/`

- Create a file called `jenkins.conf` in `/etc/apache2/sites-available`

```
<VirtualHost *:80>
        ServerAdmin webmaster@localhost
        ServerName ci.company.com
        ServerAlias ci
        ProxyRequests Off
        <Proxy *>
                Order deny,allow
                Allow from all
        </Proxy>
        ProxyPreserveHost on
        ProxyPass / http://localhost:8080/ nocanon
        AllowEncodedSlashes NoDecode
</VirtualHost>
```

- `sudo a2ensite jenkins`
- `sudo apache2ctl restart`

Setting up an Nginx Proxy for port 80 -> 8080

This configuration will setup Nginx to proxy port 80 to 8080 so that you can keep Jenkins on 8080. Instructions originally found in a GitHub Gist from rdegges: https://gist.github.com/913102

- **Install Nginx**.

```
sudo aptitude -y install nginx
```

- **Remove default configuration**.

```
cd /etc/nginx/sites-available
sudo rm default ../sites-enabled/default
```

- **Create new configuration for Jenkins**. This example uses `cat`, but you can use your favorite text editor. Make sure to replace 'ci. yourcompany.com' with your domain name.
  Note: Sometimes your permissions (umask, etc) might be setup such that this won't work. Create the file somewhere else then copy it into place if you run into that problem.

```
sudo cat > jenkins
upstream app_server {
    server 127.0.0.1:8080 fail_timeout=0;
}

server {
    listen 80;
    listen [::]:80 default ipv6only=on;
    server_name ci.yourcompany.com;

    location / {
        proxy_set_header X-Forwarded-For
$proxy_add_x_forwarded_for;
        proxy_set_header Host $http_host;
        proxy_redirect off;

        if (!-f $request_filename) {
            proxy_pass http://app_server;
            break;
        }
    }
}
^D # Hit CTRL + D to finish writing the file
```

- **Link your configuration** from `sites-available` to `sites-enabled`:

```
sudo ln -s /etc/nginx/sites-available/jenkins /etc/nginx/sites-
enabled/
```

- **Restart Nginx**

```
sudo service nginx restart
```

Where to go from here?

- You might want to make Jenkins visible through Apache, to make it available on port 80 (for example, http://myserver/jenkins/ instead of http://myserver:8080/). See this blog for more details
- You might want to see latest releases of Jenkins Debian packages. See http://pkg.jenkins-ci.org/debian-stable/