

---

# **Ubuntu QA Dashboard Documentation**

***Release 2014.10.30+revno765***

**Andy Doan, Chris Johnston, Joe Talbott**

January 19, 2015



<b>1</b>	<b>Installing the Ubuntu QA Dashboard</b>	<b>3</b>
1.1	Vagrant . . . . .	3
1.2	Grab the source . . . . .	3
1.3	Setting up Vagrant . . . . .	3
1.4	Run the Server . . . . .	4
1.5	Testing . . . . .	4
<b>2</b>	<b>Indices and tables</b>	<b>5</b>



Contents:



---

## Installing the Ubuntu QA Dashboard

---

Before you start hacking on the dashboard, you need to first setup your environment. There are a few different ways to do this.

### 1.1 Vagrant

#### 1.1.1 Requirements

To get started, you'll need:

- Vagrant

```
sudo apt-get install vagrant
```

### 1.2 Grab the source

Create the bazaar environment for the dashboard with:

```
bzr branch lp:qa-dashboard  
cd qa-dashboard
```

### 1.3 Setting up Vagrant

Now we're ready to go, so create an environment for the dashboard:

```
vagrant up
```

That creates a clean environment using vagrant.

After Vagrant finishes setting up, you need to ssh into it:

```
vagrant ssh
```

#### 1.3.1 Start developing

From inside the vagrant environment, you need to go to where the branch is:

```
cd dash
```

You can now work on the dashboard as you would with any other django site.

Vagrant takes care of all of the setup work. You will have a database that is pre-populated with data. Depending on how old the snapshot of the database is, you may need to update the migrations:

```
python manage.py migrate
```

## 1.4 Run the Server

Now all you need to do is run the server:

```
python manage.py runserver 0.0.0.0:8000
```

## 1.5 Testing

The quick way to run the dashboard's tests:

```
./manage.py test
```

Make sure to write tests for all new code your write and all the tests pass.



---

## Indices and tables

---

- *genindex*
- *modindex*
- *search*