

Manual Creation

This document describes the process of setting up a single server and client to run OpenVZ hosting. There are two machines: the primary and the secondary. The primary machine will only run the Django server, and the secondary machine will be running OpenVZ.

Some Variables

```
PRIMARY="primary.example.com"
SECONDARY="secondary.example.com"
USER="openvz"
IDRSAPUB="http://$PRIMARY/openvz/id_rsa.pub"
DEFAULT_VHOST="default"
DEFAULT_VHOST_URL="http://$PRIMARY/openvz/$DEFAULT_VHOST"
MAPFILE=/etc/apache2/$SECONDARY.map
```

Note



All these commands run as root.

First, the secondary machine needs to install the OpenVZ kernel, vzctl, and vzquota.

Install OpenVZ

```
apt-get install linux-openvz vzctl vzquota
```

Then, reboot into the new kernel to setup Apache proxy for the Virtual Environments.

Install Apache, turn on proxy, rewrite

```
apt-get install apache2
a2enmod proxy
a2enmod rewrite
cd /tmp
wget $DEFAULT_VHOST_URL
mv -f $DEFAULT_VHOST /etc/apache2/sites-available/default
echo export MAPFILE=$MAPFILE >> /etc/apache2/envvar
echo export HOSTNAME=`hostname` >> /etc/apache2/envvar
apache2ctl restart
```

The secondary machine needs to give access to the primary machine. So, the secondary machine needs to create a user with a constant username such as \$USER above. Then, the \$USER needs to have access to run vzctl as root.

Give the primary machine access

```
### Create user
adduser --disabled-password --gecos GECOS --ingroup nogroup $USER
# or for systems not based on Debian
# useradd -m -g nogroup $USER

### Give access to primary machine
mkdir -p /home/$USER/.ssh
chown $USER:nogroup /home/$USER/.ssh
chmod 700 /home/$USER/.ssh
cd /home/$USER/.ssh
wget $IDRSAPUB
cat id_rsa.pub > authorized_keys

### Give access to vzctl on the secondary machine
visudo /etc/sudoers
# Add the following line
# $USER ALL = NOPASSWD: /usr/sbin/vzctl

### OpenVZ configuration
vzsplit -n [MAX number of Virtual Environments] > /etc/vz/conf/ve-vps.basic.conf-sample
```

Now the secondary machine is configured. The next step is to insert the information on the secondary machine into the Django server running in the primary machine.