

# App Deployment with Fabric

Brad Montgomery  
[brad@workforpie.com](mailto:brad@workforpie.com)

# What is Fabric?

- Python (2.5) Library + Command line tool
- Automate your SSH/Bash ***tasks***
- Local & Remote systems
- Run tasks on multiple systems
- It's lightweight & easy to learn.

# What fabric is not.

- Enterprise Configuration Management
- Chef
- Puppet

# Sneak Peak

```
from fabric.api import run
```

```
def host_type():  
    run('uname -s')
```

To run:

```
$ fab -H localhost,user@host host_type
```

# Requirements

- Python (2.5 or greater)
- C compiler available (for the SSH libs)
- *NOTE: Best on Linux/Unix/OS X*
- For more info, see: <http://fabfile.org>

# Installation

- Best bet:
  - `pip install fabric`
  - `easy_install fabric`
- Use your favorite Linux package manager:
  - `fabric` or `python-fabric`
- Download the source, open a shell, run:
  - `python setup.py install`

# Walkthrough

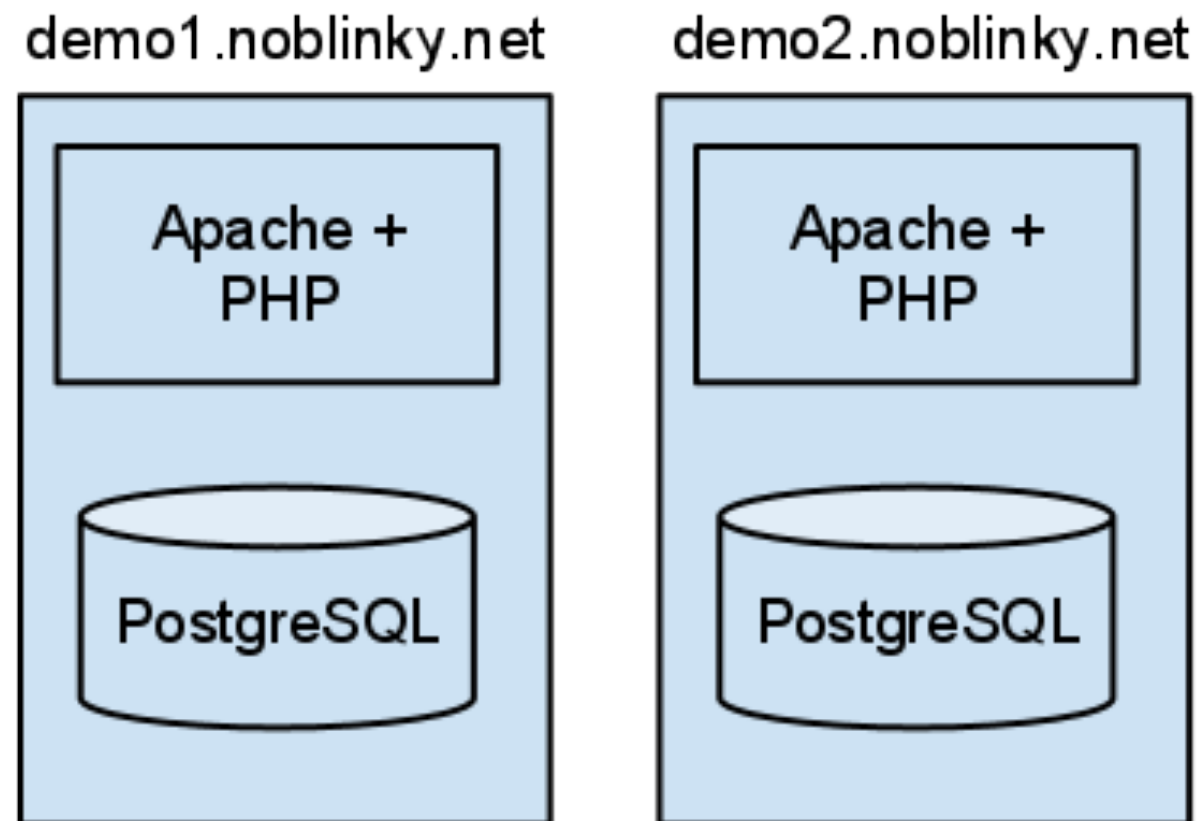
- Install/Upgrade some linux packages
- Pull/Update a PHP app from github
- Update an Apache Config & restart
- Change some file permissions
- Create a database/run some SQL

# Assumptions

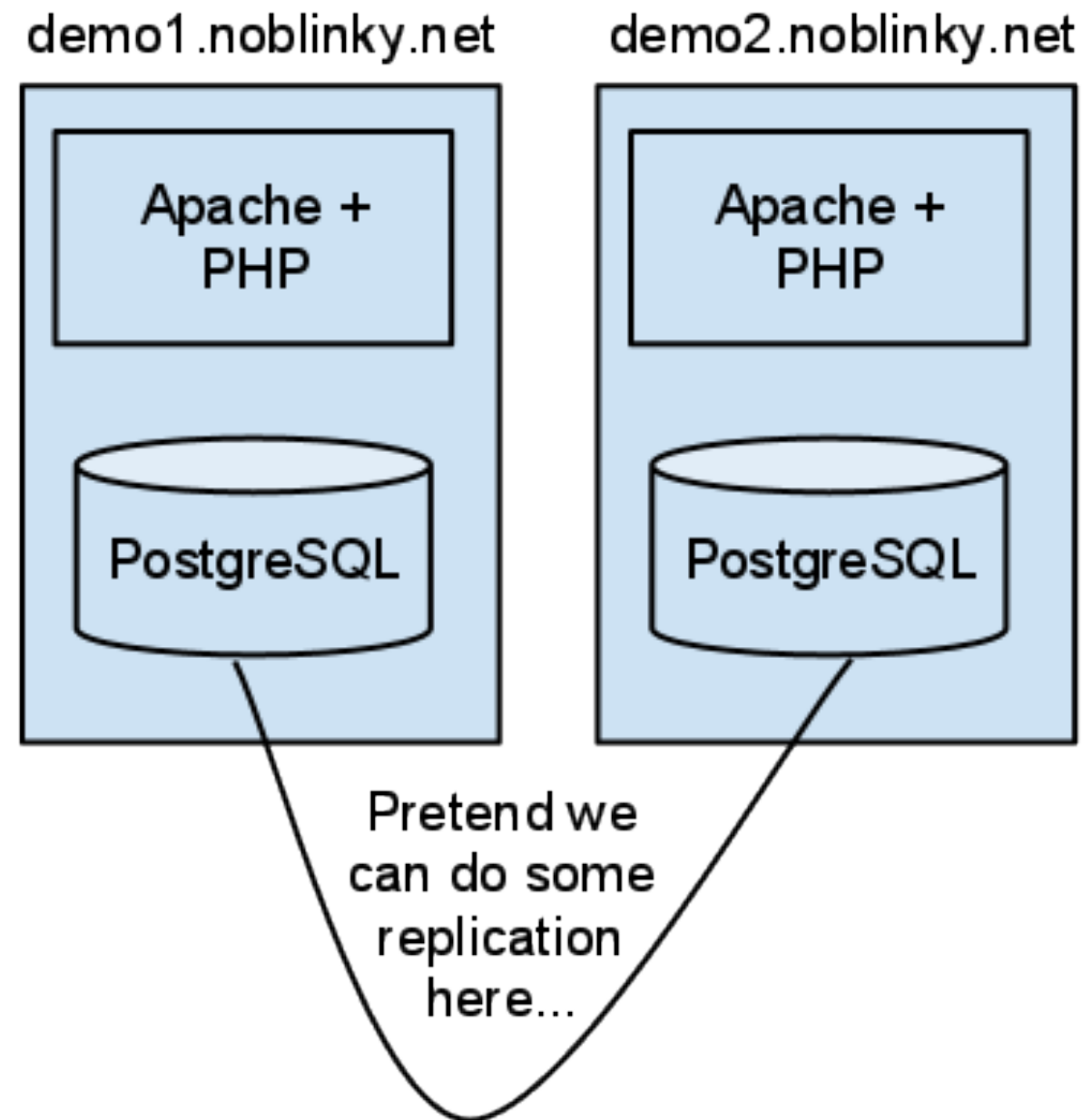
- You've got a Linux Server (ubuntu)
- Running Apache/mod\_php
- Your app's on github
- Servers' have domain names or you've got appropriate names in `/etc/hosts`
- I'll reference: demo1 and demo2



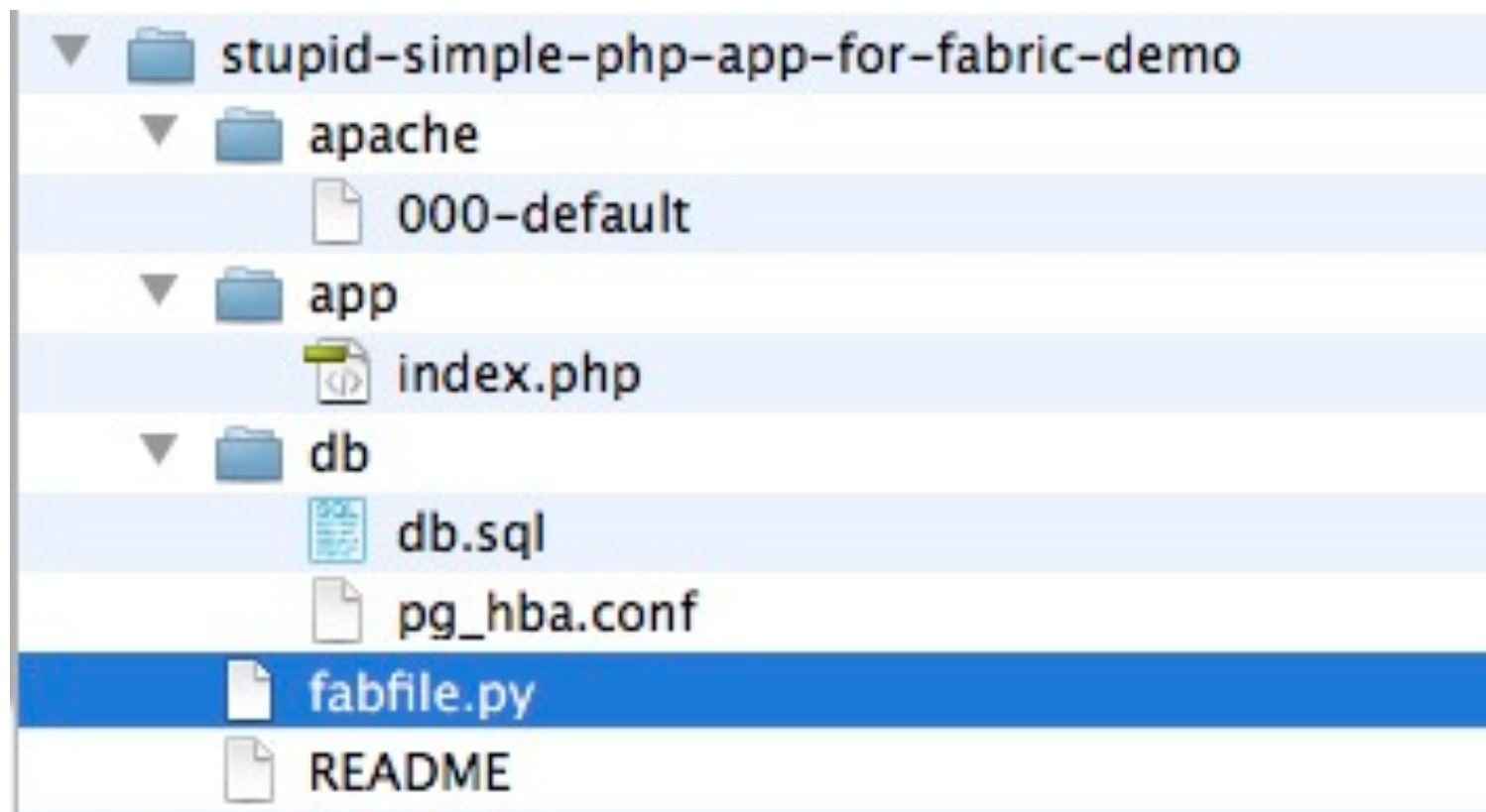
# Demo Arch.



# Demo Arch.



# Our App...



# Our App...

```
<!doctype html>
<html>
<head><title>hello world</title></head>
<body>
<h1>
<?php
    echo 'Hello World!';
?>
</h1>
</body>
</html>
```

# Install some packages

Add this function to fabfile.py

```
def install_packages():  
    """ Install a bare minimum LAMP stack """  
    run('apt-get install aptitude')  
    run('aptitude update')  
    run('aptitude upgrade')  
    run('aptitude install libapache2-mod-php5 php5 git')  
    run('aptitude install postgresql php5-pgsql')
```

```
$ fab -H root@demo1 install_packages
```

```
$ fab -H root@demo1,root@demo2 \  
install_packages
```

# Deploy your code

```
def deploy():
    """ Deploy our app by pulling it from github """

    # NOTE: Python's "with" statement is a "context manager".
    # All the following commands will be prefixed with
    # "cd /path/to/ && "

    # Make sure our app directory exists.
    if not exists("/home/web/stupid-simple-php-app-for-fabric-demo/"):
        run("mkdir -p /home/web/")
        with cd("/home/web/"):
            run("git clone git://github.com/bradmontgomery/
                stupid-simple-php-app-for-fabric-demo.git")
    else:
        with cd("/home/web/stupid-simple-php-app-for-fabric-demo"):
            # Fetch & Merge from the remote repo
            run('git pull git://github.com/bradmontgomery/
                stupid-simple-php-app-for-fabric-demo.git')
```

```
$ fab -H root@demo1,root@demo2 deploy
```

# Change file permissions

```
def set_file_permissions():  
    """ Sets the appropriate read/execute permissions """  
    with cd("/home/web/stupid-simple-php-app-for-fabric-demo/"):   
        run("chmod -R 0755 app/")  
        run("chown -R root:www-data app/")
```

```
$ fab -H root@demo1,root@demo2 \  
    set_file_permissions
```

# Change Apache's config

```
<VirtualHost *:80>
    ServerAdmin webmaster@localhost

    DocumentRoot /home/web/stupid-simple-php-app-for-fabric-demo/app
    <Directory />
        Options FollowSymLinks
        AllowOverride None
    </Directory>
    <Directory /home/web/stupid-simple-php-app-for-fabric-demo/app/>
        Options Indexes FollowSymLinks MultiViews
        AllowOverride None
        Order allow,deny
        allow from all
    </Directory>

    ErrorLog ${APACHE_LOG_DIR}/error.log
    LogLevel warn
    CustomLog ${APACHE_LOG_DIR}/access.log combined
</VirtualHost>
```



# Change Apache's config

```
def set_apache_config():  
    """ Replace's apache's default config file """  
    # Delete the existing default config file.  
    if exists("/etc/apache2/sites-enabled/000-default"):  
        run("rm /etc/apache2/sites-enabled/000-default")  
  
    # Symlink the apache config in our app  
    with cd("/etc/apache2/sites-enabled/"):   
        run("ln -s /home/web/.../apache/000-default .")  
        run("apachectl restart")
```

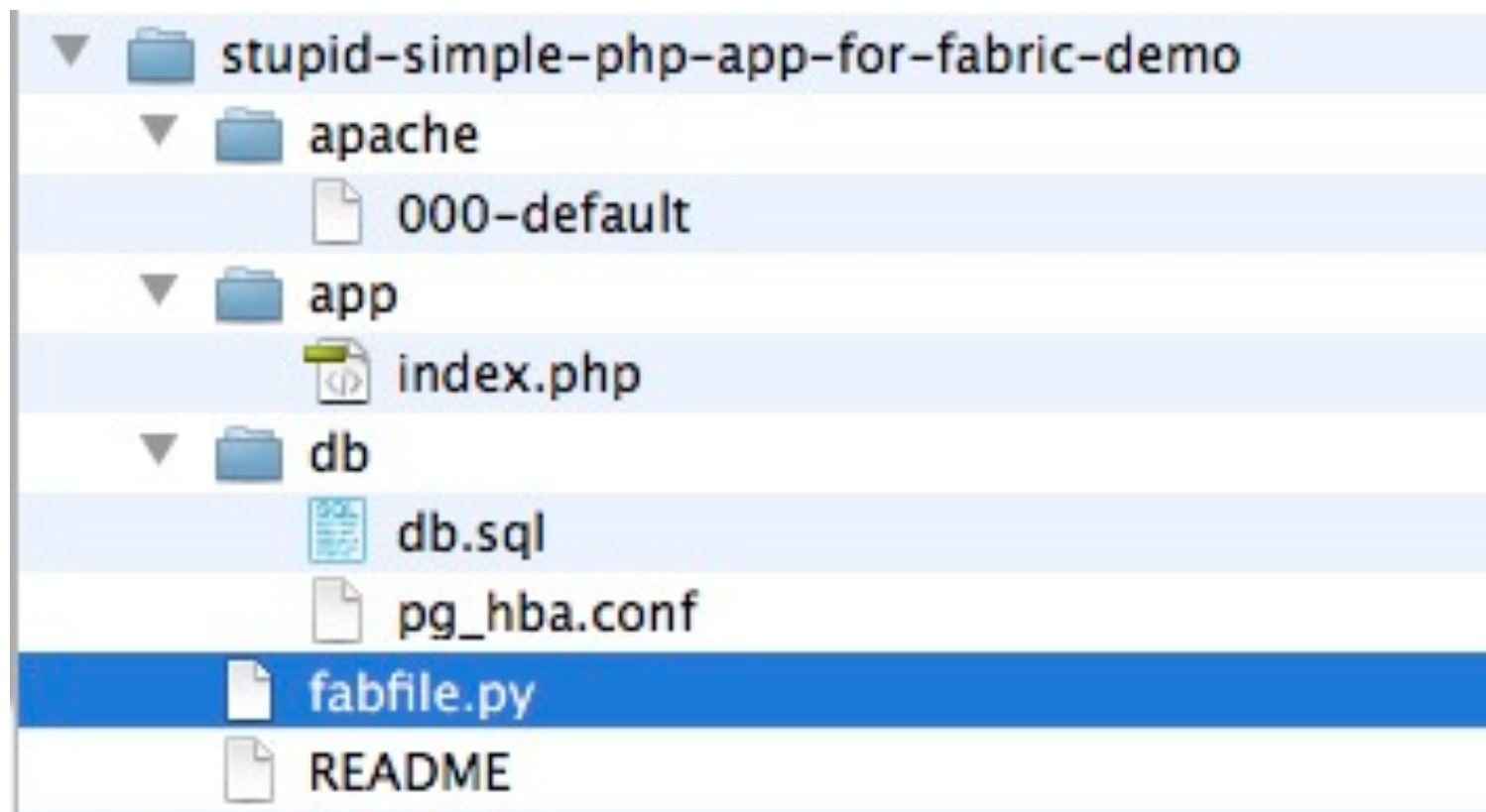
```
$ fab -H root@demo1,root@demo2 \  
    set_apache_config
```

# Backup/Revert

```
def backup():  
    """ Make a tarball snapshot of the deployed codebase """  
    with cd("/home/web/"):   
        run("tar -czf /root/stupid-simple-php-app-for-fabric-demo.tgz  
            stupid-simple-php-app-for-fabric-demo/")  
  
def revert():  
    """ Revert the codebase from a backup """  
    if exists("/root/stupid-simple-php-app-for-fabric-demo.tgz"):  
        with cd("/home/web/"):   
            run("tar -xzf /root/stupid-simple-php-app-for-fabric-demo.tgz")
```

```
$ fab -H root@demo1,root@demo2 backup;  
$ fab -H root@demo1,root@demo2 deploy;  
$ fab -H root@demo1,root@demo2 revert;
```

# Our App...



# Update our App...

```
<h1><?php echo 'Hello World!'; ?></h1>
<hr/>
<?php
    $conn = pg_connect(
        "host=localhost dbname=appdb user=dbuser");

    $result = pg_query("SELECT * FROM things");
    $rows = pg_fetch_all($result);
    if($rows) {
        echo '<ul>';
        foreach($rows as $item) {
            echo '<li>' . $item['name'] . '</li>';
        }
        echo '</ul>';
    }
?>
</body></html>
```

# db/db.sql

```
DROP TABLE things;
```

```
CREATE TABLE things (  
    id serial PRIMARY KEY,  
    name character varying(255)  
);
```

```
INSERT INTO things (name) VALUES ( 'puppies' );
```

```
INSERT INTO things (name) VALUES ( 'kittens' );
```

```
INSERT INTO things (name) VALUES ( 'butterflies' );
```

```
INSERT INTO things (name) VALUES ( 'ninja stars' );
```

# Set up a database

```
def db_setup():  
    """ set up a new database """  
    # Symlink our own config file and restart PostgreSQL  
    with cd("/etc/postgresql/8.4/main/"):   
        # remove the existing config file  
        if exists("pg_hba.conf"):   
            run("rm pg_hba.conf")  
        run("ln -s /home/web/.../db/pg_hba.conf pg_hba.conf")  
        run("/etc/init.d/postgresql restart")  
  
    # Create a superuser named "dbuser" and prompt for a password  
    sudo("createuser -s -P dbuser", user="postgres")  
    # Create a database named "appdb" owned by "dbuser"  
    sudo("createdb -E UTF8 -O dbuser -T template0 appdb", user="postgres")  
    # Run SQL from a file  
    run("psql -U dbuser -d appdb -f /home/web/.../db/db.sql")
```

```
$ fab -H root@demo1,root@demo2 db_setup
```

# Deploy an Update

```
$ fab -H root@demo1,root@demo2 deploy
```

# Maintenance Mode

```
def down():  
    """ put the site into *maintenance mode* """  
  
    # put an "out of order" page in place  
    if exists("/home/web/.../app/down.html"):   
        run("cat /home/web/.../app/down.html >  
            /var/www/index.html")  
  
    # Update Apache config  
    with cd("/etc/apache2/sites-enabled/"):   
        if exists("/etc/apache2/sites-enabled/000-default"):   
            run("rm /etc/apache2/sites-enabled/000-default")  
        run("ln -s /home/web/.../apache/down down")  
        run("apachectl restart")
```

```
$ fab -H root@demo1,root@demo2 down
```



# ~~Maintenance Mode~~

```
def up():  
    """ pull the site out of *maintenance mode* """  
    # replace the production apache config  
    with cd("/etc/apache2/sites-enabled/"):   
        if exists("/etc/apache2/sites-enabled/down"):   
            run("rm /etc/apache2/sites-enabled/down")  
        run("ln -s /home/web/.../apache/000-default 000-default")  
        run("apachectl restart")
```

```
$ fab -H root@demo1,root@demo2 up
```

# host env

Sick of typing this?

```
$ fab -H root@demo1,root@demo2 up
```

# host env

```
$ fab -H root@demo1,root@demo2 up
```

```
from fabric.api import cd, env, run
```

```
env.hosts = [ 'root@demo1@noblinky.net',  
              'root@demo2@noblinky.net', ]
```

```
def host_type():  
    run('uname -s')
```

```
$ fab host_type
```

# For More...

- See fabric's help: `fab --help`
- See available commands: `fab --list`
- <http://fabfile.org/>
- Grab the examples:
  - <https://github.com/bradmontgomery/stupid-simple-php-app-for-fabric-demo>

# Thanks!