

---

# Spawn Eden Documentation

*Release 0.1a*

**Praneeth Bodduluri**

January 12, 2013



# CONTENTS



# GETTING STARTED:

- We recommend a virtualenv for using these scripts.
- Install the dependencies for the scripts - Fabric and Cuisine.  

```
pip install -r requirements.txt
```
- Use the following commands to install and configure a standalone Eden instance.

**Install :**

```
fab -H targetmachine setup_eden_standalone
```

**Configure :**

```
fab -H targetmachine configure_eden_standalone
```



# GETTING STARTED WITH AMAZON EC2:

The following illustrates the procedure to spawn a standalone Sahana Eden instance on EC2.

- Create a file called **.boto** in your home directory with the following contents with the access key and the secret access key as per your account.

```
[Credentials]
aws_access_key_id = REPLACE_ME_WITH_ACCESS_KEY
aws_secret_access_key = REPLACE_ME_WITH_SECRET_ACCESS_KEY
```

- Upload your public key to be used with the instances created with EC2.

```
fab aws_import_key:key_name=awskey,public_key=path_to_your_public_key,ZONE='us-east-1b'
```

- Create a security group to be used with the instances spawned with EC2.

```
fab aws_create_security_group:name=default,ZONE='us-east-1b'
```

- Create a standalone Sahana Eden instance.

```
fab aws_eden_standalone
```





# FABFILE DOCUMENTATION

## Sahana Eden deployment script

`fabfile.aws_clean(*args, **kwargs)`

Cleans AWS instances in the specific region.

**Parameters** **ZONE** – AWS Zone to list the instances from. **Default ‘us-east-1b’**

`fabfile.aws_create_image(*args, **kwargs)`

Wrapper around boto’s create\_image

### Parameters

- **instance\_id** – ID of the Instance to create an image from.
- **name** – Name of the Image.
- **description** – Description for the Image created. **Default None**
- **no\_reboot** – Shutdown the instance with the given instance\_id while creating the image. **Default False**
- **ZONE** – AWS Zone to list the instances from. **Default ‘us-east-1b’**

`fabfile.aws_create_security_group(*args, **kwargs)`

This function creates security groups with access to the ports given from ALL ips.

### Parameters

- **name** – Name of the new security group we are creating.
- **description** – Description for the new security group we are creating. **Default ‘None’**
- **ports** – List containing ports to allow access from all ips. **Default [80,22,161,443]**
- **ZONE** – AWS Zone to list the instances from. **Default ‘us-east-1b’**

`fabfile.aws_delete_image(*args, **kwargs)`

Deletes the AMI and the EBS snapshot associated with the image\_id

### Parameters

- **image\_id** – The image id of the image to AMI/EBS snapshot to delete.
- **ZONE** – The AWS zone in which the image\_id is located. **Default: ‘us-east-1b’**

`fabfile.aws_eden_standalone(*args, **kwargs)`

Spawns a standalone AWS instance of Eden with Postgres, uwsgi and Cherokee.

Arguments are same as those of `fabfile.aws_spawn()`

`fabfile.aws_import_key(*args, **kwargs)`

Imports a RSA key into AWS

### Parameters

- **key\_name** – Name to store this key as in AWS.
- **public\_key** – Path to the key file to be uploaded. **Note:** only 1024, 2048, and 4096 key lengths RSA accepted.
- **ZONE** – AWS Zone to list the instances from. **Default ‘us-east-1b’**

`fabfile.aws_list(*args, **kwargs)`

Lists out AWS instances launched in the specific region

**Parameters** **ZONE** – AWS Zone to list the instances from. **Default ‘us-east-1b’**

`fabfile.aws_postgres(*args, **kwargs)`

Spawns an AWS instance and installs Postgres with Eden - The uwsgi Eden instance is not started. Eden install on this machine is used only for initialization of DB and migration.

Arguments are same as those of `fabfile.aws_spawn()`

`fabfile.aws_spawn(*args, **kwargs)`

Spawns an AWS instance with the given specs.

### Parameters

- **IMAGE** – The AMI to spawn **Default - ‘ami-cb66b2a2’**
- **INSTANCE\_TYPE** – The type of instance to run. **Default - ‘t1.micro’**

### Following options are allowed:

- m1.small
- m1.large
- m1.xlarge
- c1.medium
- c1.xlarge
- m2.xlarge
- m2.2xlarge
- m2.4xlarge
- cc1.4xlarge
- t1.micro
- **ZONE** – The Zone to spawn the IMAGE. **NOTE:** The IMAGE AMI should exist in the ZONE selected. **Default - ‘us-east-1b’**
- **SECURITY\_GROUP** – The security group to set this AMI up with. **Default - ‘default’**
- **KEY\_NAME** – The SSH key to set this AMI up with. **Default - ‘awskey’**
- **SHUTDOWN\_BEHAVIOR** – Specifies whether the instance stops or terminates on instance-initiated shutdown. **Default - stop**

### Valid values are:

- stop
- terminate

- **NAME** – Set the key “Name” with this value. **Default - ‘changeme’**

`fabfile.aws_tsung(*args, **kwargs)`

Spawns an AWS instance with TSUNG set up to run load testing.

Arguments are same as those of `fabfile.aws_spawn()`

`fabfile.configure_eden_standalone(start_eden=True)`

Configure an installed Eden - Postgres instance - This is to be run after installing Eden with other helpers provided.

**Parameters** `start_eden` – Start uWSGI after configuring eden. **Default True**

`fabfile.drop_eden(path='/home')`

Installs packages necessary for Eden.

**Parameters** `path` – Path to the directory in which a Web2py directory along with Eden is created.

**Default: ‘/home’**

`fabfile.init_env()`

Initializes the Debian env to work with Cuisine.

`fabfile.install_memcached()`

Installs memcached on the remote machine

`fabfile.install_postgres()`

Install Postgres on a remote machine

`fabfile.run_tsung(xml, target, additional_file='', run_name='')`

Runs tsung tests with a given xml against the given target - Replaces localhost in the xml with the target and fetches the reports dir.

**Parameters**

- **xml** – The path to the xml file to upload.
- **target** – The machine to run the test against.
- **additional\_file** – The path to additional file to upload.
- **run\_name** – Prepend the log directory of tsung with this.

**Returns** A tar.gz of the logs directory.

**Example Usage:**

```
fab -i awskey.pem -u root -H machine_with_tsung run_tsung:xml=tsung-tests/test.xml,target=machine_with_tsung
```

`fabfile.setup_eden()`

Sets up Postgres, uwsgi and Cherokee

`fabfile.setup_snmpd()`

Installs snmpd on a given host for monitoring. **NOTE: Allows everyone to connect**

`fabfile.setup_tsung()`

Installs Tsung for Load testing



# PYTHON MODULE INDEX

f

fabfile, ??