AppFlow - Deduplicating Archiver

Release 1.0.1.5

2021-06-16

Contents

AppSide

Read the documentation on appside.rtfd.io

For the playbooks, visit the appside-playbooks repository

More Screencasts: Installation, Basic setup, Atlantis setup, Atlantis provision, Project provision

Get an invite and join the conversations!

slack

AppSide is a multitenant environment automation tool based on Ansible.

AppSide is your transparent platform that incorporates the various digital infrastructures into a continuous workflow. At the same time, it creates a consistent environment for the entire development process, saving time and money. AppSide is an open source developer and DevOps application that maps the entire development process (continuous deployment & provisioning) from development to test & staging to production. This ensures that the systems being managed on the different infrastructures are deployed in the same way. At the same time, AppSide creates a consistent environment (on-premise, in the cloud or on bare metal) throughout the development process. The administrators and developers (DevOps) can therefore independently control the entire toolchain. The automation process is created using Ansible playbooks and contains the admin and developer code, which is stored in a git repository (encrypted). The code (AppSide) runs on any UNIX-like operating system where Python can run, including Linux, BSD, and OSX. Orchestration of the AppSide code requires only SSH access to the respective systems. AppSide has been designed and optimized from DevOps for DevOps.

1.1 Features

Provisioning:

- Multitenant architecture (different teams with different environments)
- Supports `development`, `testing`, `staging` and `production`
- All configuration files are encrypted in git with `ansible-vault`
- Provision all nodes with one command

Development:

- Provides a Vagrant based development environment called `atlantis`
- Code locally on any Unix-like system or Windows (cygwin)

Deployment:

- Made for Bedrock projects and bedrock-capistrano deployments
- Deploy and rollback with one command

Infrastructure:

- Builtin Percona XtraDB Cluster 5.6 and GlusterFS support for sharing web uploads on multiple nodes
- Easy development environments with Vagrant
- Easy server provisioning with Ansible (Ubuntu 16.04, PHP 5.6/7)

1.2 Technologies

The technology behind AppSide uses countless best-in-class programs and maps them in a toolbox. The software accesses an extensive repository of various freeware packages. Out-of-the-box, many enterprise features already exist, such as:

- Load balancing
- Apache / PHP
- Web Accelerator / Caching / PageSpeed
- Distributed file system
- Master-Master Database
- Backup & Monitoring Integration
- Jailkit - chroot
- …

1.2.1 Installation

AppSide is hosted on PiP using python3. `pip3 install appside` will install appside. To start using it you first need to *initialize* it: `appside init` follow the onscreen instructions to set it up!

1.2.2 Developers

1.2.2.1 Contribute a new feature

- Create a new issue, e.g. #XX new superfeature

- Create local branch: `git checkout -b XX-new-superfeature`

- Code on it.

- Push it to remote as new branch: `git push -u origin XX-new-superfeature`

- Create new pull request (`base:   master ... compare:   XX-new-superfeature`)

1.2.2.2 Get all vars

```
ansible all -m setup --tree /tmp/facts -i examples/YOUR_TENANT/local/inventory
-a "filter=ansible_distribution*"
```

1.3 Contributing

Contributions are welcome from everyone.

Join us!

1.3.1 Introduction

1.3.2 AppSide

Read the documentation on appside.rtfd.io

For the playbooks, visit the appside-playbooks repository

More Screencasts: Installation, Basic setup, Atlantis setup, Atlantis provision, Project provision

Get an invite and join the conversations!

AppSide is a multitenant environment automation tool based on Ansible.

AppSide is your transparent platform that incorporates the various digital infrastructures into a continuous workflow. At the same time, it creates a consistent environment for the entire development process, saving time and money. AppSide is an open source developer and DevOps application that maps the entire development process (continuous deployment & provisioning) from development to test & staging to production. This ensures that the systems being managed on the different infrastructures are deployed in the same way. At the same time, AppSide creates a consistent environment (on-premise, in the cloud or on bare metal) throughout the development process. The administrators and developers (DevOps) can therefore independently control the entire toolchain. The automation process is created using Ansible playbooks and contains the admin and developer code, which is stored in a git repository (encrypted). The code (AppSide) runs on any UNIX-like operating system where Python can run, including Linux, BSD, and OSX. Orchestration of the AppSide code requires only SSH access to the respective systems. AppSide has been designed and optimized from DevOps for DevOps.

1.3.2.1 Features

Provisioning:

- Multitenant architecture (different teams with different environments)
- Supports `development`, `testing`, `staging` and `production`
- All configuration files are encrypted in git with `ansible-vault`
- Provision all nodes with one command

Development:

- Provides a Vagrant based development environment called `atlantis`
- Code locally on any Unix-like system or Windows (cygwin)

Deployment:

- Made for Bedrock projects and bedrock-capistrano deployments
- Deploy and rollback with one command

Infrastructure:

- Builtin Percona XtraDB Cluster 5.6 and GlusterFS support for sharing web uploads on multiple nodes
- Easy development environments with Vagrant
- Easy server provisioning with Ansible (Ubuntu 16.04, PHP 5.6/7)

1.3.2.2 Technologies

The technology behind AppSide uses countless best-in-class programs and maps them in a toolbox. The software accesses an extensive repository of various freeware packages. Out-of-the-box, many enterprise features already exist, such as:

- Load balancing
- Apache / PHP
- Web Accelerator / Caching / PageSpeed
- Distributed file system
- Master-Master Database
- Backup & Monitoring Integration
- Jailkit - chroot
- . . .

1.3.2.2.1 Installation

AppSide is hosted on PiP using python3. `pip3 install appside` will install appside. To start using it you first need to *initialize* it: `appside init` follow the onscreen instructions to set it up!

1.3.2.2.2 Developers

Contribute a new feature

- Create a new issue, e.g. #XX new superfeature
- Create local branch: `git checkout -b XX-new-superfeature`
- Code on it.
- Push it to remote as new branch: `git push -u origin XX-new-superfeature`
- Create new pull request (`base:  master ... compare:  XX-new-superfeature`)

Get all vars

```
ansible all -m setup --tree /tmp/facts -i examples/YOUR_TENANT/local/inventory
-a "filter=ansible_distribution*"
```

1.3.2.3 Contributing

Contributions are welcome from everyone.

Join us!

1.3.3 Installation

More Screencasts: Basic setup, Atlantis setup, Atlantis provision, Project provision

1.3.3.1 Prerequisites

The preferred method to install AppFlow is `pip`; to install it you must do:

On MacOS (depends on Brew):

```
brew install python3
```

Or if you already have python2 installed do:

```
brew upgrade python
```

On Linux Systems:

```
- Ubuntu/Debian:
sudo apt install python3-pip

- RedHat/Fedora:
sudo dnf install python3-pip
```

After this you will be able to install appflow using:

```
pip3 install appflow
```

this will install the appflow executable you will use.

This installation is incomplete without initializing the playbooks and your tenant. To initialize use:

```
appflow init
```

Then follow the simple instructions to choose your tenant name and default environment.

At this point the installation is **complete**. Now we will illustrate the folder structure.

1.3.3.2 Others

Fix Ansible problems on 14.04

The python3 version Shipped with Ubuntu 14.04 is not enough to use ansible from pip3 that Appflow brings as dependency. We need to remove it and default to the PPA installation:

```
-   sudo pip3 uninstall ansible
-   sudo apt install python2 python2-pip python3 python3-pip git
-   sudo apt-add-repository ppa:ansible/ansible
-   sudo apt install ansible
```

1.3.4 Usage

The following section will cover some basic usages of Appflow.

1.3.4.1 Folder Structure

This will initialize your folders and default playbooks in the appflow config directory. This folder is placed in **$HOME** in **$HOME/.appflow**

You will find here:

```
ls ~/.appflow
.appflow
├── config.yml
├── playbooks
├── tenant
├── tmp
└── vault

4 directories, 1 file
```

What you need to know:

config.yml holds your default config (default tenant, environment) so you do not have to specify them always. To use something different from defaults, AppFlow allows you to specify them during your command: *appflow provision –tenant ANOTHER_TENANT –env ANOTHER_ENV*

playbooks this is where all playbooks are placed, you can contribute to them visiting the repository: Appflow-Playbooks. To keep them updated, you can just type *appflow update*

tenant this is where all your tenants (you can have as many as you wish) will be placed. all TEnants aRE ogranized by name (*~/.appflow/tenant/tenant1, ~/.appflow/tenant/tenant2. . .*) in your tenant you then specify the inventory files for each environment (*~/.appflow/tenant/tenant1/development, ~/.appflow/tenant/tenant1/testing. . .*)

vault this will hold your passwords to decrypt your intentories (**appflow decrypt**, **appflow encrypt**) it's organized in a similar fashon of tenant folder: *~/.appflow/vault/tenant1/* this folder will contain files with the password. These files have to be named with the environment they correspond to: *~/.appflow/vault/tenant1/development. . .*

in your tenant you then specify the inventory files for each environment (*~/.appflow/tenant/tenant1/development, ~/.appflow/tenant/tenant1/testing. . .*)

1.3.4.2 Setting up a new user

More Screencasts: Installation, Atlantis setup, Atlantis provision, Project provision

This sction will refere to the **example tenant** that you will find in this repository: Appflow-Example repository

To setup a new user, you will need to modify the file

```
$HOME/.appflow/tenant/appflow-example/development/group_vars/all
```

There is a section called **conf_users**. Here you will find a series of users that will be configured in your base system.

The example tenant will have the seguent code:

```
REPLACE_USER_NAME:
  state: enabled
  groups: "{{ conf_sudo_nopasswd_group }},{{ conf_www_group }}"
  name: "Basic User"
  home: "/home/REPLACE_USER_NAME"
  shell: "{{ conf_zsh_path }}"
  public_key: |
    REPLACE_SSH_KEY
  private_key:
  is_deployer: yes
  is_mysql_admin: yes

deploy:
  state: enabled
  groups: "{{ conf_www_group }}"
  name: "Capistrano Deploy User"
  home: "/home/deploy"
  shell: "{{ conf_zsh_path }}"
  public_key: |
    REPLACE_SSH_KEY
  private_key: |
  is_deployer: no
  is_mysql_admin: yes
```

As you can see the structure is pretty simple, just REPLACE_SSH_KEY with **YOUR public ssh key** (you can find it in **$HOME/.ssh/id_rsa.pub**).

`private_key` is not needed for a simple user

`is_deployer` will specify if this user is allowed to deploy

`is_mysql_admin` will specify if this user is a mysql admin

User Groups You can also specify in what group the user should (or not) be.

Populate the `groups` line with a list of the groups.

`conf_www_group` is likely needed for a developer and deployer.

`conf_sudo_nopasswd_group` is likely needed for a mantainer and admin user.

And that's it.

If you want to add new users you can simply clone the settings of the example user and modify the confs as described.

1.3.4.3 Setting up a new Project

More Screencasts: Installation, Basic setup, Atlantis setup, Atlantis provision

A project is defined in a tenant as a virtual host.

Setup a vhost

Referring to the *example tenant*, there is a section called `conf_vhosts`. Here you will find:

```
example:
  state: enabled
  servername: dev.example.com
  serveralias:
    - www.dev.example.com
  serveradmin: webmaster@ttss.ch
  db_name: example_wp
  db_user: example_wp
  db_password: We9Diel2
  db_server: 127.0.0.1
  db_host: '%'
  db_prefix: ahCh7Fei
  glusterfs_uploads: example
  type: wp_bedrock
  config_env: true
  env_opt: |
    CFG_URL_DEVELOPMENT='https://dev.example.com'
    CFG_URL_STAGING='https://stage.example.com'
    CFG_URL_PRODUCTION='https://example.com'
    WPMDB_LICENCE='XXXXXXXXXXXXXXXXXXXXXXXXXXXX'
  wp_salt: |
    AUTH_KEY=']+-v`hHqK.M |QO.1|sYEqM5v:^^[3-,]:s?Bbtus9@y+nEbE7+ladg.E|H+<EC|'
    SECURE_AUTH_KEY='94ZoJdn,s:Jy)p-.pH-A3`GtK$BrwZTL6lA-520F=/l90*,i]t-zI|0QZn)Cq#1|'
    LOGGED_IN_KEY='$Jd>Gv{:5}^r||!=.F8*Psg{j_B27TVC{n-R<@9GqF[d`@$WhGd+tf?OiPyN8kcb8'
    NONCE_KEY='46^*wL<)IzG01Y/m_e,|Hb/B-!5:mc#.w{6~@ipSSJc*|67>d[|HJ&OY*|DXjV83'
    AUTH_SALT='!;oV$#%N3WcL*VsW3IkhI0}FtO/fJ`*H}n18.3.2bT5sW/svc-1nKnp~PONKT<B$'
    SECURE_AUTH_SALT='iqyy{?<h`fNX/iQ}on>cmB|/bbRD*nZD;8fGDH5`an_-Qj|:h|yO|two>a-yZ;*x
↪'
    LOGGED_IN_SALT='l82hxF[w)R)L|bqw:a@;x=+geLouagDu:)}ss1k=T:!#.fc:9ZU{hJPEmV`7<BRi'
    NONCE_SALT='R{~-C+p|eJ=mEF,5F$m-|8@<HocSO!e&GNPw{_GTjW]c@to@8[O3RJA7:G-gMu!F'
  htaccess:
  htpasswd_password: false
  ssl_pemfile:
  ssl_haproxy: false
  ssl_pem:
  bkup: false
  bkup_www_hosts:
  bkup_mysql_hosts:
  bkup_cron_schedule:
```

Just replace the values with the one needed in your project. In particular pay attention to the `db_user` and `db_password`. For the `wp_salt` section you can refer to This Website to generate random values for the project.

1.3.4.4 Setting Up Atlantis (14.04)

This procedure has to be executed in the Atlantis VM. To enter it just do

```
- vagrant ssh atlantis
```

1.3.4.4.1 We need percona repo to complete the provisioning

From inside atlantis we have to perform the following commands:

```
-   wget https://repo.percona.com/apt/percona-release_0.1-4.$(lsb_release -sc)_all.deb
-   sudo dpkg -i percona-release_0.1-4.$(lsb_release -sc)_all.deb
-   sudo apt update
-   sudo apt-get install -y percona-xtradb-cluster-server-5.7
```

1.3.4.4.2 Upgrade Packages

```
-   sudo apt update && sudo apt upgrade
-   sudo pip list --outdated --format=columns | grep -v sdist | awk '{print $1}' |␣
→tail -n +3 | xargs -n1 sudo pip install -U
-   sudo pip list --outdated --format=columns | grep -v sdist | awk '{print $1}' |␣
→tail -n +3 | xargs -n1 sudo pip3 install -U
```

1.3.4.5 Setting Up Atlantis (16.04)

More Screencasts: Installation, Basic setup, Atlantis provision, Project provision

1.3.4.5.1 We first need to install Python or ansible will not work

```
- vagrant ssh atlantis -c "sudo apt-get install -y python"
```

1.3.4.5.2 We now need to setup the percona repo and package to install

First we enter atlantis using

```
- vagrant ssh atlantis
```

Then from inside atlantis we have to perform the following commands:

```
-   wget https://repo.percona.com/apt/percona-release_0.1-4.$(lsb_release -sc)_all.deb
-   sudo dpkg -i percona-release_0.1-4.$(lsb_release -sc)_all.deb
-   sudo apt update
-   sudo apt-get install -y percona-xtradb-cluster-server-5.7
-   sudo chown mysql:mysql /run/mysqld
```

note: get ssh pwd for ubuntu user:

```
-   vagrant ssh atlantis -c "echo $(cat ~/.ssh/id_rsa.pub) | sudo tee -a /home/ubuntu/
↪.ssh/authorized_keys"
-   vagrant ssh atlantis -c "sudo passwd ubuntu"
```

1.3.4.6 Provision Atlantis

More Screencasts: Installation, Basic setup, Atlantis setup, Project provision

To provision atlantis we have first to use the `--first-run` flag, because the basic users are not yet configured.

```
- appflow provision --env development --tenant example --limit atlantis --first-run␣
↪True
```

From now on the `--first-run` flag can be ignored as your `id_rsa.pub` key will be used to authenticate.

After the first provision that will setup all the basic packages, users and configs, the most useful `tags` you will use are:

- `--tags env` this will provision `.env`

- `--tags htaccess` this will provision `.htaccess`

- `--tags vhosts` this will provision the projects/vhosts

- `--tags mysql` this will provision both `db` and `users`

- `--tags shell-users` this will provision any new users that will be created afterwards.

Remember, tags can be concatenated, for example

```
- appflow provision --env development --tenant example --limit atlantis,testing --
↪tags htaccess,env,vhosts,mysql
```

1.3.5 Useful Tips

1.3.5.1 Aliases

An useful alias you can add to your `.bashrc` or your `.zshrc` is:

```
vagup(){pushd ~/.appflow/playbooks; vagrant up $@; popd}
vaghalt(){pushd ~/.appflow/playbooks; vagrant halt $@; popd}
vagdestroy(){pushd ~/.appflow/playbooks; vagrant destroy $@; popd}
```

This will make your vagrant managing much faster.

1.3.5.2 Update Playbooks and Vagrantfile

Since the split in `Appflow` and `Appflow-Playbooks`, you can now just update yout playbooks and Vagrantfile using:

```
appflow update
```

1.3.6 FAQs

1.3.6.1 Help

You can always have basic help from `appflow` itself:

`appflow`

Will print a generic help:

```
% appflow
Type:         AppFlow
String form: <__main__.AppFlow object at 0x7f75a19fa080>
Docstring:    Appflow CLI tool.

Type appflow to have a list of available commands.
Type appflow command -- --help to have help for the specified command.

Usage:        appflow
              appflow add
              appflow checkin
              appflow checkout
              appflow decrypt
              appflow encrypt
              appflow get
              appflow init
              appflow provision
              appflow reset
              appflow rm
              appflow set
              appflow ssh
              appflow status
              appflow tags
              appflow update
              appflow vhosts
```

You will have greather help, typing `appflow COMMAND -- --help` This will print a more detailed help for every function you need (add,checking,checkout. . .)

Example:

```
% appflow provision -- --help                                              ␣
→[12:48:37]
Type:         method
String form: <bound method AppFlow.provision of <__main__.AppFlow object at␣
→0x7fc0f056eb70>>
File:         /usr/local/bin/appflow
Line:         197
Docstring:    Provision your machines.
Syntax is:
appflow provision "machine1,machine2" tag1,tag2 skiptag1,skiptag2
tags: will run only the tags specified
skip_tags: will run all the tags except for the specified ones
limit: limit to only some specified hosts.

Optionally it is possible to specify custom tenant and environment
appflow provision tenant-name env-name...
this is optional and by default will read the
```

(continues on next page)

```
default config in ~/.appflow/config.yml

:type  tenant: string
:param tenant: The name of the tenant.

:type  env: string
:param env: The name of the tenant.

:type  limit: string
Usage:        appflow provision [TENANT] [ENV] [LIMIT] [TAGS] [SKIP_TAGS] [FIRSTRUN]␣
→[LOCAL]
              appflow provision [--tenant TENANT] [--env ENV] [--limit LIMIT] [--tags␣
→TAGS] [--skip-tags SKIP_TAGS] [--firstrun FIRSTRUN] [--local LOCAL]
```

Read carefully the various helps, and in case of doubts head to the Developer section Where you will be able to read each function's Docstring and source code.

1.3.6.2 Let's Encrypt!

```
Issue: Setup Let's Encrypt on a server provisioned with Appflow.
```

```
Solve:

Certificate verification:
    ssh REMOTE-SERVER.NAME "sudo cert-verify.sh"

Create a new Certificate:
    ssh REMOTE-SERVER.NAME "sudo cert-create.sh www.YOUR-URL.DOMAIN"

Then manually renew all the certificates:
    ssh REMOTE-SERVER,NAME "sudo cert-renew.sh"
```

1.3.6.3 Troubleshooting

1.3.6.3.1 [vagrant] Missing Vagrantfile.local.yml

```
Issue: There was an error loading a Vagrantfile. The file being loaded
       and the error message are shown below. This is usually caused by
       a syntax error.

       Path: /Users/foo/Documents/webdev/appflow/Vagrantfile
       Line number: 0
       Message: Errno::ENOENT: No such file or directory @ rb_sysopen - Vagrantfile.
→local.yml`
```

```
Solve: add Vagrantfile.local.yml to the appflow folder with this content:

       synced_folder:
         appflow_folder: "~/Documents/webdev/appflow"
         webdev_folder: "~/Documents/webdev/development"
```

1.3.6.3.2 [vagrant] Vagrant was unable to mount VirtualBox shared folders

```
Issue: Vagrant was unable to mount VirtualBox shared folders.
    This is usually because the filesystem "vboxsf" is not available.
    This filesystem is made available via the VirtualBox Guest Additions
    and kernel module. Please verify that these guest additions are properly
    installed in the guest. This is not a bug in Vagrant and is usually
    caused by a faulty Vagrant box. For context, the command attempted was:

    id -u deploy

    The error output from the command was:

    id: deploy: no such user
```

```
Solve: appflow provision limit=atlantis firstrun=true (password is vagrant).
```

1.3.6.3.3 [vagrant] The box you attempted to add doesn't match the provider you specified

```
Issue: The box you attempted to add doesn't match the provider you specified.
```

```
Solve: vagrant up --provider=virtualbox atlantis
```

1.3.6.3.4 [vagrant] Lost Vagrant reference to VirtualBox VM

```
Issue: Lost Vagrant reference to VirtualBox VM

Solve:
VBoxManage list vms
  "vagrant-atlantis" {xxxxxxxx-xxxx-xxxx-xxxx-xxxxxxxx}
echo xxxxxxxx-xxxx-xxxx-xxxx-xxxxxxxx > ~/appflow/.vagrant/machines/atlantis/
→virtualbox/id
```

1.3.6.3.5 [vagrant] Warning: Authentication failure. Retrying. . .

```
Issue: vagrant Warning: Authentication failure. Retrying...
```

```
Solve: http://stackoverflow.com/a/30792296
```

1.3.6.3.6 [vagrant] an error occurred while downloading the remote file

```
Issue: An error occurred while downloading the remote file.
    The error message, if any, is reproduced below. Please fix this error and try
→again.
```

```
Solve: sudo mv /opt/vagrant/embedded/bin/curl /tmp
```

See also: https://github.com/mitchellh/vagrant/issues/7997

1.3.6.3.7 [boot] An error occurred while mounting /

```
Issue: An error occurred while mounting /.
Keys: Press S to skip mounting or M for manual recovery
```

```
Solve:
Press S and try to see if atlantis boots up.
ssh atlantis
mount -o remount,rw / (optional)
e2fsck /dev/sda1
reboot
```

1.3.7 Changes

1.3.7.1 Appflow 1.0.1.5

Minor update introducing:

- Appflow-playbooks versioning

- Appflow-playbooks selectable branch

- *appflow version* will now yield also the playbooks version: `Appflow Version:  1.0.1.5 Playbooks Version 1.0.0`

1.3.7.2 Appflow 1.0.1.4

Released a little update including

- introduced appflow version command

- introduced appflow provision debug

- fixed newline bug in appflow hosts

to update just use Pip: *pip3 install -U appflow*

1.3.7.3 Screencasts

To complement the documentation, we've added screencasts recorded on asciinema.org

Screencasts include the walkthrought of the basic setups and features of Appflow. They include:

- Installation

- Basic setup

- Atlantis setup

- Atlantis provision

- Project provision

1.3.7.4 Varnish Grace mode

Grace mode has been a long expected feature and went production ready with recent appflow-playbooks commit f0d7f3817ffb1b2354f0c24a98e3dac37b72202d.

This special operation mode in Varnish allows a website to remain online and running also when the backend components, like MySQL or Apache2, are not running. This works beccause Varnish will serve all the web content directly from it's cache.

Backend operations like login to CMS or similar will obviously not work as expected but at least the public facing part of the website will not be offline, so no 404s or similar for end-users.

Grace period is set by default to 6h but can be extended to one week or whatever your preference is, the main advantage of Grace is that your site remains up when bad things happen and you'll get a time buffer for fixing whatever issue happened to the backend.

Grace mode will be enabled by default, for any environment, if you perform:

```
% appflow update
% appflow provision --tags varnish-conf,apache2-conf,mysql
```

It's important to note that Grace mode is active by default from Varnish 5.1 upwards so if you're on 3.x you'll need to upgrade varnish first, this can be done, in development, by:

```
% appflow update
% ssh atlantis "sudo service varnish stop"
% appflow provision --tags varnish,apache2-conf,mysql
```

You also need to set the Varnish version in `group_vars/webservers`:

```
#
# Varnish
#
conf_varnish_version: 51
...
```

a complete config setting for Varnish 5.1 and Grace would look like:

```
#
# Varnish
#
conf_varnish_version: 51
conf_varnish_listen_port: 6081
conf_varnish_listen_admin_port: 60821
conf_varnish_backend_default: |
.host = "127.0.0.1";
.port = "8080";
.max_connections = 800;
conf_varnish_acl_purge: |
"127.0.0.1";
"localhost";
"192.168.80.2";
conf_varnish_vcl_recv: |
{% if conf_lbtier_enable == false %}
remove req.http.X-Forwarded-For;
set req.http.X-Forwarded-For = client.ip;
{% endif %}
```

```
if (req.url ~ "/wp(-login|-admin|-cron|login|-comments-post.php)" ) {
    return (pass);
}

if (req.http.Cache-Control ~ "no-cache") {
    return (pass);
}

# Remove client-side cookies.
set req.http.Cookie = regsuball(req.http.Cookie, "(^|;\s*)(_[_a-z]+|has_
↪js|utmctr|utmcmd.|utmccn.|WT_FPC|_hjIncludedInSample)=[^;]*", "");

# Remove a ";" prefix, if present.
set req.http.Cookie = regsub(req.http.Cookie, "^;\s*", "");

# Are there cookies left with only spaces or that are empty?
if (req.http.cookie ~ "^\s*$") {
    unset req.http.cookie;
}

conf_varnish_vcl_backend_response: |
# Allow stale content, in case the backend goes down.
# make Varnish keep all objects for 6 hours beyond their TTL
set beresp.grace = 6h;
# set beresp.grace = 2m;

conf_varnish_vcl_fetch: |
# set beresp.grace = 2m;

# If the URL is for one of static images or documents, we always want them to be
↪cached.
if (beresp.status == 200 && req.url ~ "\.(ico|jpe?g|jpe|gif|png|webp|svg|css|js)$") {
# Cookies already removed.
# Cache the page for 10 days.
    set beresp.ttl = 10d;
# Remove existing Cache-Control headers.
    remove beresp.http.Cache-Control;
# Set new Cache-Control headers for browser to store cache for 7 days.
    set beresp.http.Cache-Control = "public, max-age=604800";
}

# Cache 404 responses for 15 seconds.
if (beresp.status == 404) {
    set beresp.ttl = 15s;
    set beresp.grace = 15s;
}

conf_varnish_vcl_deliver: |
# For security and asthetic reasons, remove some HTTP headers before final delivery.
unset resp.http.Server;
unset resp.http.X-Powered-By;
unset resp.http.Via;
unset resp.http.X-Varnish;
Once Varnish has been updated and Grace mode has been enabled you could test if it's
↪working correctly:
```

Open one of the web projects you're hosting on Atlantis in the browser, everything should be there.

```
% curl http://atlantis:8080/health.php
MySQL running

% ssh atlantis "sudo varnishadm backend.list"
Backend name                    Admin      Probe
boot.default                    probe      Healthy

% ssh atlantis "sudo service mysql stop"

% ssh atlantis "sudo varnishadm backend.list"
Backend name                    Admin      Probe
boot.default                    probe      Sick
```

Open the previous web project again in the browser, the website should be online as usual via Grace mode.

The same concept applies also to production where we have three or more nodes.

1.3.8 Support

Please first read the docs, the existing issue tracker issues and mailing list posts – a lot of stuff is already documented / explained / discussed / filed there.

1.3.8.1 Issue Tracker

If you've found a bug or have a concrete feature request, please create a new ticket on the project's issue tracker.

1.3.9 Development

This chapter will get you started with AppFlow development.

AppFlow is written in Python. Here you will find all the references to the Code

1.3.9.1 appflow

1.3.9.1.1 appflow module

Appflow CLI tool.

Type appflow to have a list of available commands. Type appflow command – –help to have help for the specified command.

class appflow.**AppFlow**

Bases: :class:'object'

Appflow CLI tool.

Type appflow to have a list of available commands. Type appflow command – –help to have help for the specified command.

add(*file*, *key*, *value*)

This will create and then print the key you are specifying. Syntax: appflow get tenant.environment.folder.to.file.searched key.subkey.value

Parameters

- **file** (`string`) – path.to.file (dot encoded) where to set the key.

- **key** (`string`) – The key to search. (this function will add it if not found.)

- **value** (`T`) – the value to set.

checkin(*tenant=''*, *env=''*, *commit='Auto Commit'*)

Git push from yout local tenant repository. This will only push the files that were modified. Before any push, all the files are encrypted.

> **Parameters**
>
> - **tenant** (`string`) – The name of the tenant.
>
> - **env** (`string`) – The name of the tenant.
>
> - **commit** (`string`) – The commit message to use when committing. (default Auto Commit)

checkout(*tenant=''*, *env=''*)

Git pull your local tenant repository. This will download the lates available code. This will also overwrite any unpushed work.

> **Parameters**
>
> - **tenant** (`string`) – The name of the tenant.
>
> - **env** (`string`) – The name of the tenant.

decrypt(*tenant=''*, *env=''*)

Decrypt your local tenant repository

> **Parameters**
>
> - **tenant** (`string`) – The name of the tenant.
>
> - **env** (`string`) – The name of the tenant.

encrypt(*tenant=''*, *env=''*)

Encrypt your local tenant repository

> **Parameters**
>
> - **tenant** (`string`) – The name of the tenant.
>
> - **env** (`string`) – The name of the tenant.

get(*file*, *key=None*)

This will print the key you are searcing (or the whole file if key is not specified) Syntax: appflow get tenant.environment.folder.to.file.searched key.subkey.value

> **Parameters**
>
> - **file** (`string`) – path.to.file (dot encoded) where to search the key.
>
> - **key** (`string`) – The key to search.

init(*tenant=None*, *env=None*)

This will initialize all the folders for Assh. This will also setup autocompletion for the CLI tool.

> **Parameters**
>
> - **tenant** (`string`) – The name of the tenant.
>
> - **env** (`string`) – The name of the tenant.

provision (*tenant=", env=", limit: str = None, tags: str = None, skip_tags: str = None, firstrun: bool = False, local: bool = False, debug: bool = False*)

Provision your machines. Syntax is: appflow provision "machine1,machine2" tag1,tag2 skiptag1,skiptag2 tags: will run only the tags specified skip_tags: will run all the tags except for the specified ones limit: limit to only some specified hosts.

Optionally it is possible to specify custom tenant and environment appflow provision tenant-name env-name... this is optional and by default will read the default config in ~/.appflow/config.yml

> **Parameters**
>
> - **tenant** (*string*) – The name of the tenant.
> - **env** (*string*) – The name of the tenant.
> - **limit** (*string*) – Comma separated list of hosts to provision. (default None)
> - **tags** (*string*) – Comma separated list of tags to exec (default All).
> - **skip_tags** (*string*) – Comma separated list of tags to skip (default None).
> - **firstrun** (*bool*) – if it's first run (default False)
> - **local** (*bool*) – if it's doing a local auto-provision (default False)
> - **debug** (*bool*) – if it's a debug run (default False)

reset (*tenant=", env="*)

Reset your local tenant repository. This will restore the status to the latest git pull. This will also reset any unpushed work.

> **Parameters**
>
> - **tenant** (*string*) – The name of the tenant.
> - **env** (*string*) – The name of the tenant.

rm (*file, key*)

This will remove and then print the key you are specifying. Syntax: appflow get tenant.environment.folder.to.file.searched key.subkey.value

> **Parameters**
>
> - **file** (*string*) – path.to.file (dot encoded) where to remove the key.
> - **key** (*string*) – The key to search.

set (*file, key, value*)

This will modify and then print the key you are specifying. Syntax: appflow get tenant.environment.folder.to.file.searched key.subkey.value

> **Parameters**
>
> - **file** (*string*) – path.to.file (dot encoded) where to set the key.
> - **key** (*string*) – The key to search.
> - **value** (*T*) – the value to set.

ssh (*tenant=", env="*)

This will deploy the ssh keys from your tenant/env to the Assh folders.

> **Parameters**
>
> - **tenant** (*string*) – The name of the tenant.
> - **env** (*string*) – The name of the tenant.

status (*tenant=''*, *env=''*)

> Outputs your local tenant status, any modified files. This is handy to have an overview of what's going to be pushed as a dry run.
>
> > **Parameters**
> >
> > - **tenant** (`string`) – The name of the tenant.
> >
> > - **env** (`string`) – The name of the tenant.

tags (*tenant=''*, *env=''*)

> Show available tags. This is handy to provision only a part of them or skipping some of them.
>
> > **Parameters**
> >
> > - **tenant** (`string`) – The name of the tenant.
> >
> > - **env** (`string`) – The name of the tenant.

update (*branch='master'*)

> Simple function to update Appflow. This is handy for the appflow-git package. You can specify which branch you want to use
>
> > **Parameters branch** (`string`) – The name of the branch (default Master)

version ()

> This will print the appflow version and the current appflow-playbooks informations.

vhosts (*tenant=''*)

> This will setup your /etc/hosts to reflect the configs int your tenant/development host_vars. ** Needs Root Access **
>
> > **Parameters tenant** (`string`) – The name of the tenant.

1.3.9.1.2 Lib package

Submodules

lib.appflow_ansible module

Appflow Ansible utilities. This contains all the functions needed to perform Ansible actions. From provision to encryption/decryption and tag listing.

`lib.appflow_ansible.`**decrypt** (*tenant*, *env*)

> Decrypt the tenant/environment data
>
> > **Parameters**
> >
> > - **tenant** (`string`) – The name of the tenant.
> >
> > - **env** (`string`) – The name of the tenant.
> >
> > **Return type** None
> >
> > **Returns** the function prints to screen the ansible output of the execution.

`lib.appflow_ansible.`**encrypt** (*tenant*, *env*)

> Encrypt the tenant/environment data
>
> > **Parameters**
> >
> > - **tenant** (`string`) – The name of the tenant.

- **env** (*string*) – The name of the tenant.

> **Return type** None

> **Returns** the function prints to screen the ansible output of the execution.

lib.appflow_ansible.**list_tags**(*tenant*, *env*)
> List all available tags for tenant/environment

> **Parameters**

- **tenant** (*string*) – The name of the tenant.

- **env** (*string*) – The name of the tenant.

> **Return type** None

> **Returns** the function prints to screen the available tags.

lib.appflow_ansible.**provision**(*tenant: str*, *env: str*, *limit: str*, *tags: str*, *skip_tags: str*, *firstrun: bool*, *local: bool*, *debug: bool*)
> This will perform the ansible playbook. We pass tenant and environment and all other options as –option xys in order to respect ansible's syntax.

> **Parameters**

- **tenant** (*string*) – The name of the tenant.

- **env** (*string*) – The name of the tenant.

- **limit** (*string*) – Comma separated list of hosts to provision.

- **tags** (*string*) – Comma separated list of tags to exec (default All).

- **skip_tags** (*string*) – Comma separated list of tags to skip (default None).

- **firstrun** (*bool*) – if it's first run (default False)

- **debug** (*bool*) – if it's a debug run (default False)

> **Return type** None

> **Returns** the function prints to screen the ansible output of the execution.

lib.appflow_tools module

Appflow Tools. This contains all the functions needed to perform actions connected to initialization, config deployment and git versioning.

lib.appflow_tools.**git_check_in**(*tenant*, *env*, *commit*)
> Git push. This will affecy only the modified files (see git_status function). Commit message can be specified.

> **Parameters**

- **tenant** (*string*) – The name of the tenant.

- **env** (*string*) – The name of the tenant.

- **commit** (*string*) – The commit message to use when committing.

> **Return type** None

> **Returns** the function doesn't have a return statement.

lib.appflow_tools.**git_check_out**(*tenant*, *env*)
> Git pull of the specified tenant/environment folder. un-pushed work can be overwritten by this, so ask for confirmation.

> **Parameters**
>
> - **tenant** (*string*) – The name of the tenant.
>
> - **env** (*string*) – The name of the tenant.
>
> **Return type** None
>
> **Returns** the function doesn't have a return statement.

lib.appflow_tools.**git_reset**(*tenant*, *env*)
  Perform git reset in the specified tenant/environment folder. After this, updates the md5 file to reflect the new status.

> **Parameters**
>
> - **tenant** (*string*) – The name of the tenant.
>
> - **env** (*string*) – The name of the tenant.
>
> **Return type** None
>
> **Returns** the function doesn't have a return statement.

lib.appflow_tools.**git_status**(*tenant*, *env*)
  Return a status of modified files in the tenant/environment folder. this is tracked separately from git, because encryption/decryption of files will always override the git status method.

> **Parameters**
>
> - **tenant** (*string*) – The name of the tenant.
>
> - **env** (*string*) – The name of the tenant.
>
> **Return type** list
>
> **Returns** the function returns a list containing the different lines between the 2 md5 files.

lib.appflow_tools.**git_update_playbooks**(*branch*)
  Git pull the latest version of the playbooks. You can specify which branch you want to use

> **Parameters** **branch** (*string*) – The name of the branch

lib.appflow_tools.**initialize**(*tenant*, *env*)
  Create default dirs, clone playbooks and yaml files for Assh to function properly.

> **Parameters**
>
> - **tenant** (*string*) – The name of the tenant. (ex: mrrobot)
>
> - **env** (*string*) – The name of the tenant.
>
> **Return type** None
>
> **Returns** This function doesn't have a return statement.

lib.appflow_tools.**set_vhosts_hosts**(*tenant*)
  Setup /etc/hosts for tenant. Requires root access to write.

> **Parameters** **tenant** (*string*) – The name of the tenant.
>
> **Return type** None
>
> **Returns** the function doesn't have a return statement.

lib.appflow_tools.**setup_default_config**(*tenant*, *env*)
  Deploy a default config file in ~/.appflow/config.yml

> **Parameters**

- **tenant** (*string*) – The name of the tenant. (ex: mrrobot)

- **env** (*string*) – The name of the tenant.

Return type None

Returns the function prints to screen the ansible output of the execution.

lib.appflow_tools.**setup_ssh**(*tenant*, *env*)
    Deploy Assh configs for tenant/environment.

Parameters

- **tenant** (*string*) – The name of the tenant.

- **env** (*string*) – The name of the tenant.

Return type None

Returns the function doesn't have a return statement.

lib.appflow_utils module

Appflow Utilities. This contains all the generic functions needed to support the rest of the library.

lib.appflow_utils.**add_keys**(*data_dict*, *key*, *value=None*)
    Add keys to dictionary (set also value if specified)

Parameters

- **data_dict** (*dict*) – The dictionary where to search the key.

- **key** (*string*) – The key to search.

- **value** (*string*) – The value to set. (default None)

Return type None

Returns the function doesn't have a return statement.

lib.appflow_utils.**check_string_in_file**(*file_name*, *searched_string*)
    Check if string is in file

Parameters

- **file_name** (*string*) – The file name where to search the string.

- **searched_string** (*string*) – The string to search.

Return type bool

Returns the function returns if the string is found or not.

lib.appflow_utils.**diff_files**(*file1*, *file2*)
    Returns different lines between file1 and file2. Returned data is a list of strings.

Parameters

- **file1** (*string*) – The name of the first file.

- **file2** (*string*) – The name of the second file.

Return type list

Returns the function returns a list containing the different lines between the 2 files.

lib.appflow_utils.**format_string_argument**(*argument*)

    Fire takes multiple arguments (comma separated) as list or tuple. Check argument type and put it to string.

        Parameters **argument** (`tuple or list`) – The argument passed.

        Return type string

        Returns Separated comma strings convertion for lists and tuples.

lib.appflow_utils.**get_appflow_folder**()

    Get directory or appflow.

        Parameters **_file** (`string`) – The name of the script file executed internally.

        Return type string

        Returns the function returns the root of appflow. Needed to then search for playbooks.

lib.appflow_utils.**get_env_color_string**(*env*)

    Color code for the environment variable Needed in provision string.

        Parameters **env** (`string`) – The name of the tenant.

        Return type string

        Returns the function returns the color needed for the corresponding env.

lib.appflow_utils.**get_file_list**(*_dir*)

    Returns a list of files in a directory.

        Parameters **_dir** (`string`) – The name of the directory to explore.

        Return type list

        Returns the function returns the list of files in the folder.

lib.appflow_utils.**get_from_dict**(*data_dict*, *key*)

    Return key-value dictionary

        Parameters

            • **data_dict** (`dict`) – The dictionary where to search the key.

            • **key** (`string`) – The key to search.

        Return type dict

        Returns the function returns a dict containing the key-value pair searched.

lib.appflow_utils.**get_md5_folder**(*tenant*)

    Get directory for the specified tenant md5 files.

        Parameters **tenant** (`string`) – The name of the tenant.

        Return type string

        Returns the function returns the md5_folder searched.

lib.appflow_utils.**get_md5_sum**(*file_name*)

    Return the md5 checksum of the specified file.

        Parameters **file_name** (`string`) – The name of the file to hash.

        Return type string

        Returns the function returns the md5 hash of the file.

lib.appflow_utils.**get_provision_color_string**(*command*, *tenant*, *env*)

    Color code for the provision string

Parameters

- **command** (*string*) – The command to execute.
- **tenant** (*string*) – The name of the tenant.
- **env** (*string*) – The name of the tenant.

Return type string

Returns the function returns the color coded string to print before the execution of the ansible command.

lib.appflow_utils.**get_tenant_dir**(*tenant*)
    Get directory for the specified tenant.

Parameters **tenant** (*string*) – The name of the tenant.

Return type string

Returns the function returns the tenant folder.

lib.appflow_utils.**get_tenant_env_dir**(*tenant*, *env*)
    Get directory for the specified tenant/environment.

Parameters

- **tenant** (*string*) – The name of the tenant.
- **env** (*string*) – The name of the environment.

Return type string

Returns the function returns the tenant/environment folder.

lib.appflow_utils.**get_vault_file**(*tenant*, *env*)
    Get vault file for the specified tenant/environment.

Parameters

- **tenant** (*string*) – The name of the tenant.
- **env** (*string*) – The name of the environment.

Return type string

Returns the function returns the vault file searched.

lib.appflow_utils.**rm_in_dict**(*data_dict*, *key*)
    Remove keys from dictionary

Parameters

- **data_dict** (*dict*) – The dictionary where to search the key.
- **key** (*string*) – The key to search.

Return type dict

Returns the function returns the dictionary with the deleted the key searched.

lib.appflow_utils.**safe_remove**(*file_name*)
    Gracefully delete a file.

Parameters **file_name** (*string*) – The name of the file to delete.

Return type None

Returns the function doesn't have a return statement.

lib.appflow_utils.**set_in_dict**(*data_dict*, *key*, *value*)
Set key-value in dictionary

>   **Parameters**
>
>   - **data_dict** (`dict`) – The dictionary where to search the key.
>
>   - **key** (`string`) – The key to search.
>
>   - **value** (`string`) – The value to set.
>
>   **Return type** None
>
>   **Returns** the function doesn't have a return statement.

lib.appflow_utils.**write_md5_sum**(*file_name*, *md5_store_file*)
Write the modified md5 filename to the md5_store_file

>   **Parameters**
>
>   - **file_name** (`string`) – The name of the file to hash.
>
>   - **md5_store_file** (`string`) – The name of the file where to write the hash.
>
>   **Return type** None
>
>   **Returns** the function doesn't have a return statement.

lib.appflow_utils.**yes_no**(*question*, *default='yes'*)
Get a prompt for asking a question with y/N as accepted answer.

>   **Parameters**
>
>   - **question** (`string`) – The question to ask.
>
>   - **default** (`string`) – The default answer. (default Yes)
>
>   **Return type** bool
>
>   **Returns** the function returns if the answer was yes or no.

lib.appflow_yaml module

Appflow Yaml utilities. This contains all the functions needed to manipulate yaml files. Handy for configs and for tenant setups.

lib.appflow_yaml.**add_value**(*orig_file*, *orig_key*, *value*)
Returns key-value for searched key in file. Key will be created with the value specified. Data is written to file. Returns string in json format.

>   **Parameters**
>
>   - **_file** (`string`) – path.to.file (dot encoded) where to set the key.
>
>   - **key** (`string`) – The key to search. (this function will add it if not found.)
>
>   - **value** (`T`) – the value to set.
>
>   **Return type** json
>
>   **Returns** the function returns a json containing the updated file content.

lib.appflow_yaml.**get_value**(*_file*, *key=None*)
Returns key-value for searched key in file. If key is not specified, returns the whole file. Returns string in json format.

> **Parameters**
>
> - **_file** (*string*) – path.to.file (dot encoded) where to search the key.
> - **key** (*string*) – The key to search.
>
> **Return type** json
>
> **Returns** the function returns a json containing the key-value searched.

lib.appflow_yaml.**rm_value**(*_file*, *key*)

> Returns key-value for searched key in file. Searched key will be removed. Data is written to file. Returns string in json format.
>
> **Parameters**
>
> - **_file** (*string*) – path.to.file (dot encoded) where to remove the key.
> - **key** (*string*) – The key to search.
>
> **Return type** json
>
> **Returns** the function returns a json containing the updated file content.

lib.appflow_yaml.**set_value**(*_file*, *key*, *value*)

> Returns key-value for searched key in file. Searched key will be set with the value specified. Data is written to file. Returns string in json format.
>
> **Parameters**
>
> - **_file** (*string*) – path.to.file (dot encoded) where to set the key.
> - **key** (*string*) – The key to search.
> - **value** (*T*) – the value to set.
>
> **Return type** json
>
> **Returns** the function returns a json containing the updated file content.

Module contents

1.3.9.2 Contributions

. . . are welcome!

Some guidance for contributors:

- discuss about changes on github issue tracker, IRC or mailing list

1.3.10 Authors

1.3.10.1 AppSide authors

- Ivo Marino <ivo.marino@ttss.ch>
- Luca Di Maio <luca.dimaio@ttss.ch>

1.3.10.1.1 AppFlow patches and suggestions

- Dominik Schilling
- Stefan Pasch
- Stefan Kalb
- Houssein Maatouk
- Malik Amrein

1.3.11 License

```
                GNU AFFERO GENERAL PUBLIC LICENSE
                   Version 3, 19 November 2007

 Copyright (C) 2007 Free Software Foundation, Inc. <http://fsf.org/>
 Everyone is permitted to copy and distribute verbatim copies
 of this license document, but changing it is not allowed.

                          Preamble

  The GNU Affero General Public License is a free, copyleft license for
software and other kinds of works, specifically designed to ensure
cooperation with the community in the case of network server software.

  The licenses for most software and other practical works are designed
to take away your freedom to share and change the works.  By contrast,
our General Public Licenses are intended to guarantee your freedom to
share and change all versions of a program--to make sure it remains free
software for all its users.

  When we speak of free software, we are referring to freedom, not
price.  Our General Public Licenses are designed to make sure that you
have the freedom to distribute copies of free software (and charge for
them if you wish), that you receive source code or can get it if you
want it, that you can change the software or use pieces of it in new
free programs, and that you know you can do these things.

  Developers that use our General Public Licenses protect your rights
with two steps: (1) assert copyright on the software, and (2) offer
you this License which gives you legal permission to copy, distribute
and/or modify the software.

  A secondary benefit of defending all users' freedom is that
improvements made in alternate versions of the program, if they
receive widespread use, become available for other developers to
incorporate.  Many developers of free software are heartened and
encouraged by the resulting cooperation.  However, in the case of
software used on network servers, this result may fail to come about.
The GNU General Public License permits making a modified version and
letting the public access it on a server without ever releasing its
source code to the public.

  The GNU Affero General Public License is designed specifically to
ensure that, in such cases, the modified source code becomes available
```

(continues on next page)

```
to the community.  It requires the operator of a network server to
provide the source code of the modified version running there to the
users of that server.  Therefore, public use of a modified version, on
a publicly accessible server, gives the public access to the source
code of the modified version.

  An older license, called the Affero General Public License and
published by Affero, was designed to accomplish similar goals.  This is
a different license, not a version of the Affero GPL, but Affero has
released a new version of the Affero GPL which permits relicensing under
this license.

  The precise terms and conditions for copying, distribution and
modification follow.

                   TERMS AND CONDITIONS

  0. Definitions.

  "This License" refers to version 3 of the GNU Affero General Public License.

  "Copyright" also means copyright-like laws that apply to other kinds of
works, such as semiconductor masks.

  "The Program" refers to any copyrightable work licensed under this
License.  Each licensee is addressed as "you".  "Licensees" and
"recipients" may be individuals or organizations.

  To "modify" a work means to copy from or adapt all or part of the work
in a fashion requiring copyright permission, other than the making of an
exact copy.  The resulting work is called a "modified version" of the
earlier work or a work "based on" the earlier work.

  A "covered work" means either the unmodified Program or a work based
on the Program.

  To "propagate" a work means to do anything with it that, without
permission, would make you directly or secondarily liable for
infringement under applicable copyright law, except executing it on a
computer or modifying a private copy.  Propagation includes copying,
distribution (with or without modification), making available to the
public, and in some countries other activities as well.

  To "convey" a work means any kind of propagation that enables other
parties to make or receive copies.  Mere interaction with a user through
a computer network, with no transfer of a copy, is not conveying.

  An interactive user interface displays "Appropriate Legal Notices"
to the extent that it includes a convenient and prominently visible
feature that (1) displays an appropriate copyright notice, and (2)
tells the user that there is no warranty for the work (except to the
extent that warranties are provided), that licensees may convey the
work under this License, and how to view a copy of this License.  If
the interface presents a list of user commands or options, such as a
menu, a prominent item in the list meets this criterion.

  1. Source Code.
```

```
  The "source code" for a work means the preferred form of the work
for making modifications to it.  "Object code" means any non-source
form of a work.

  A "Standard Interface" means an interface that either is an official
standard defined by a recognized standards body, or, in the case of
interfaces specified for a particular programming language, one that
is widely used among developers working in that language.

  The "System Libraries" of an executable work include anything, other
than the work as a whole, that (a) is included in the normal form of
packaging a Major Component, but which is not part of that Major
Component, and (b) serves only to enable use of the work with that
Major Component, or to implement a Standard Interface for which an
implementation is available to the public in source code form.  A
"Major Component", in this context, means a major essential component
(kernel, window system, and so on) of the specific operating system
(if any) on which the executable work runs, or a compiler used to
produce the work, or an object code interpreter used to run it.

  The "Corresponding Source" for a work in object code form means all
the source code needed to generate, install, and (for an executable
work) run the object code and to modify the work, including scripts to
control those activities.  However, it does not include the work's
System Libraries, or general-purpose tools or generally available free
programs which are used unmodified in performing those activities but
which are not part of the work.  For example, Corresponding Source
includes interface definition files associated with source files for
the work, and the source code for shared libraries and dynamically
linked subprograms that the work is specifically designed to require,
such as by intimate data communication or control flow between those
subprograms and other parts of the work.

  The Corresponding Source need not include anything that users
can regenerate automatically from other parts of the Corresponding
Source.

  The Corresponding Source for a work in source code form is that
same work.

  2. Basic Permissions.

  All rights granted under this License are granted for the term of
copyright on the Program, and are irrevocable provided the stated
conditions are met.  This License explicitly affirms your unlimited
permission to run the unmodified Program.  The output from running a
covered work is covered by this License only if the output, given its
content, constitutes a covered work.  This License acknowledges your
rights of fair use or other equivalent, as provided by copyright law.

  You may make, run and propagate covered works that you do not
convey, without conditions so long as your license otherwise remains
in force.  You may convey covered works to others for the sole purpose
of having them make modifications exclusively for you, or provide you
with facilities for running those works, provided that you comply with
the terms of this License in conveying all material for which you do
```

```
not control copyright.  Those thus making or running the covered works
for you must do so exclusively on your behalf, under your direction
and control, on terms that prohibit them from making any copies of
your copyrighted material outside their relationship with you.

  Conveying under any other circumstances is permitted solely under
the conditions stated below.  Sublicensing is not allowed; section 10
makes it unnecessary.

  3. Protecting Users' Legal Rights From Anti-Circumvention Law.

  No covered work shall be deemed part of an effective technological
measure under any applicable law fulfilling obligations under article
11 of the WIPO copyright treaty adopted on 20 December 1996, or
similar laws prohibiting or restricting circumvention of such
measures.

  When you convey a covered work, you waive any legal power to forbid
circumvention of technological measures to the extent such circumvention
is effected by exercising rights under this License with respect to
the covered work, and you disclaim any intention to limit operation or
modification of the work as a means of enforcing, against the work's
users, your or third parties' legal rights to forbid circumvention of
technological measures.

  4. Conveying Verbatim Copies.

  You may convey verbatim copies of the Program's source code as you
receive it, in any medium, provided that you conspicuously and
appropriately publish on each copy an appropriate copyright notice;
keep intact all notices stating that this License and any
non-permissive terms added in accord with section 7 apply to the code;
keep intact all notices of the absence of any warranty; and give all
recipients a copy of this License along with the Program.

  You may charge any price or no price for each copy that you convey,
and you may offer support or warranty protection for a fee.

  5. Conveying Modified Source Versions.

  You may convey a work based on the Program, or the modifications to
produce it from the Program, in the form of source code under the
terms of section 4, provided that you also meet all of these conditions:

    a) The work must carry prominent notices stating that you modified
    it, and giving a relevant date.

    b) The work must carry prominent notices stating that it is
    released under this License and any conditions added under section
    7.  This requirement modifies the requirement in section 4 to
    "keep intact all notices".

    c) You must license the entire work, as a whole, under this
    License to anyone who comes into possession of a copy.  This
    License will therefore apply, along with any applicable section 7
    additional terms, to the whole of the work, and all its parts,
    regardless of how they are packaged.  This License gives no
```

```
    permission to license the work in any other way, but it does not
    invalidate such permission if you have separately received it.

    d) If the work has interactive user interfaces, each must display
    Appropriate Legal Notices; however, if the Program has interactive
    interfaces that do not display Appropriate Legal Notices, your
    work need not make them do so.

  A compilation of a covered work with other separate and independent
works, which are not by their nature extensions of the covered work,
and which are not combined with it such as to form a larger program,
in or on a volume of a storage or distribution medium, is called an
"aggregate" if the compilation and its resulting copyright are not
used to limit the access or legal rights of the compilation's users
beyond what the individual works permit.  Inclusion of a covered work
in an aggregate does not cause this License to apply to the other
parts of the aggregate.

  6. Conveying Non-Source Forms.

  You may convey a covered work in object code form under the terms
of sections 4 and 5, provided that you also convey the
machine-readable Corresponding Source under the terms of this License,
in one of these ways:

    a) Convey the object code in, or embodied in, a physical product
    (including a physical distribution medium), accompanied by the
    Corresponding Source fixed on a durable physical medium
    customarily used for software interchange.

    b) Convey the object code in, or embodied in, a physical product
    (including a physical distribution medium), accompanied by a
    written offer, valid for at least three years and valid for as
    long as you offer spare parts or customer support for that product
    model, to give anyone who possesses the object code either (1) a
    copy of the Corresponding Source for all the software in the
    product that is covered by this License, on a durable physical
    medium customarily used for software interchange, for a price no
    more than your reasonable cost of physically performing this
    conveying of source, or (2) access to copy the
    Corresponding Source from a network server at no charge.

    c) Convey individual copies of the object code with a copy of the
    written offer to provide the Corresponding Source.  This
    alternative is allowed only occasionally and noncommercially, and
    only if you received the object code with such an offer, in accord
    with subsection 6b.

    d) Convey the object code by offering access from a designated
    place (gratis or for a charge), and offer equivalent access to the
    Corresponding Source in the same way through the same place at no
    further charge.  You need not require recipients to copy the
    Corresponding Source along with the object code.  If the place to
    copy the object code is a network server, the Corresponding Source
    may be on a different server (operated by you or a third party)
    that supports equivalent copying facilities, provided you maintain
    clear directions next to the object code saying where to find the
```

```
     Corresponding Source.  Regardless of what server hosts the
     Corresponding Source, you remain obligated to ensure that it is
     available for as long as needed to satisfy these requirements.

     e) Convey the object code using peer-to-peer transmission, provided
     you inform other peers where the object code and Corresponding
     Source of the work are being offered to the general public at no
     charge under subsection 6d.

  A separable portion of the object code, whose source code is excluded
from the Corresponding Source as a System Library, need not be
included in conveying the object code work.

  A "User Product" is either (1) a "consumer product", which means any
tangible personal property which is normally used for personal, family,
or household purposes, or (2) anything designed or sold for incorporation
into a dwelling.  In determining whether a product is a consumer product,
doubtful cases shall be resolved in favor of coverage.  For a particular
product received by a particular user, "normally used" refers to a
typical or common use of that class of product, regardless of the status
of the particular user or of the way in which the particular user
actually uses, or expects or is expected to use, the product.  A product
is a consumer product regardless of whether the product has substantial
commercial, industrial or non-consumer uses, unless such uses represent
the only significant mode of use of the product.

  "Installation Information" for a User Product means any methods,
procedures, authorization keys, or other information required to install
and execute modified versions of a covered work in that User Product from
a modified version of its Corresponding Source.  The information must
suffice to ensure that the continued functioning of the modified object
code is in no case prevented or interfered with solely because
modification has been made.

  If you convey an object code work under this section in, or with, or
specifically for use in, a User Product, and the conveying occurs as
part of a transaction in which the right of possession and use of the
User Product is transferred to the recipient in perpetuity or for a
fixed term (regardless of how the transaction is characterized), the
Corresponding Source conveyed under this section must be accompanied
by the Installation Information.  But this requirement does not apply
if neither you nor any third party retains the ability to install
modified object code on the User Product (for example, the work has
been installed in ROM).

  The requirement to provide Installation Information does not include a
requirement to continue to provide support service, warranty, or updates
for a work that has been modified or installed by the recipient, or for
the User Product in which it has been modified or installed.  Access to a
network may be denied when the modification itself materially and
adversely affects the operation of the network or violates the rules and
protocols for communication across the network.

  Corresponding Source conveyed, and Installation Information provided,
in accord with this section must be in a format that is publicly
documented (and with an implementation available to the public in
source code form), and must require no special password or key for
```

```
unpacking, reading or copying.

  7. Additional Terms.

  "Additional permissions" are terms that supplement the terms of this
License by making exceptions from one or more of its conditions.
Additional permissions that are applicable to the entire Program shall
be treated as though they were included in this License, to the extent
that they are valid under applicable law.  If additional permissions
apply only to part of the Program, that part may be used separately
under those permissions, but the entire Program remains governed by
this License without regard to the additional permissions.

  When you convey a copy of a covered work, you may at your option
remove any additional permissions from that copy, or from any part of
it.  (Additional permissions may be written to require their own
removal in certain cases when you modify the work.)  You may place
additional permissions on material, added by you to a covered work,
for which you have or can give appropriate copyright permission.

  Notwithstanding any other provision of this License, for material you
add to a covered work, you may (if authorized by the copyright holders of
that material) supplement the terms of this License with terms:

    a) Disclaiming warranty or limiting liability differently from the
    terms of sections 15 and 16 of this License; or

    b) Requiring preservation of specified reasonable legal notices or
    author attributions in that material or in the Appropriate Legal
    Notices displayed by works containing it; or

    c) Prohibiting misrepresentation of the origin of that material, or
    requiring that modified versions of such material be marked in
    reasonable ways as different from the original version; or

    d) Limiting the use for publicity purposes of names of licensors or
    authors of the material; or

    e) Declining to grant rights under trademark law for use of some
    trade names, trademarks, or service marks; or

    f) Requiring indemnification of licensors and authors of that
    material by anyone who conveys the material (or modified versions of
    it) with contractual assumptions of liability to the recipient, for
    any liability that these contractual assumptions directly impose on
    those licensors and authors.

  All other non-permissive additional terms are considered "further
restrictions" within the meaning of section 10.  If the Program as you
received it, or any part of it, contains a notice stating that it is
governed by this License along with a term that is a further
restriction, you may remove that term.  If a license document contains
a further restriction but permits relicensing or conveying under this
License, you may add to a covered work material governed by the terms
of that license document, provided that the further restriction does
not survive such relicensing or conveying.
```

```
  If you add terms to a covered work in accord with this section, you
must place, in the relevant source files, a statement of the
additional terms that apply to those files, or a notice indicating
where to find the applicable terms.

  Additional terms, permissive or non-permissive, may be stated in the
form of a separately written license, or stated as exceptions;
the above requirements apply either way.

  8. Termination.

  You may not propagate or modify a covered work except as expressly
provided under this License.  Any attempt otherwise to propagate or
modify it is void, and will automatically terminate your rights under
this License (including any patent licenses granted under the third
paragraph of section 11).

  However, if you cease all violation of this License, then your
license from a particular copyright holder is reinstated (a)
provisionally, unless and until the copyright holder explicitly and
finally terminates your license, and (b) permanently, if the copyright
holder fails to notify you of the violation by some reasonable means
prior to 60 days after the cessation.

  Moreover, your license from a particular copyright holder is
reinstated permanently if the copyright holder notifies you of the
violation by some reasonable means, this is the first time you have
received notice of violation of this License (for any work) from that
copyright holder, and you cure the violation prior to 30 days after
your receipt of the notice.

  Termination of your rights under this section does not terminate the
licenses of parties who have received copies or rights from you under
this License.  If your rights have been terminated and not permanently
reinstated, you do not qualify to receive new licenses for the same
material under section 10.

  9. Acceptance Not Required for Having Copies.

  You are not required to accept this License in order to receive or
run a copy of the Program.  Ancillary propagation of a covered work
occurring solely as a consequence of using peer-to-peer transmission
to receive a copy likewise does not require acceptance.  However,
nothing other than this License grants you permission to propagate or
modify any covered work.  These actions infringe copyright if you do
not accept this License.  Therefore, by modifying or propagating a
covered work, you indicate your acceptance of this License to do so.

  10. Automatic Licensing of Downstream Recipients.

  Each time you convey a covered work, the recipient automatically
receives a license from the original licensors, to run, modify and
propagate that work, subject to this License.  You are not responsible
for enforcing compliance by third parties with this License.

  An "entity transaction" is a transaction transferring control of an
organization, or substantially all assets of one, or subdividing an
```

```
organization, or merging organizations.  If propagation of a covered
work results from an entity transaction, each party to that
transaction who receives a copy of the work also receives whatever
licenses to the work the party's predecessor in interest had or could
give under the previous paragraph, plus a right to possession of the
Corresponding Source of the work from the predecessor in interest, if
the predecessor has it or can get it with reasonable efforts.

  You may not impose any further restrictions on the exercise of the
rights granted or affirmed under this License.  For example, you may
not impose a license fee, royalty, or other charge for exercise of
rights granted under this License, and you may not initiate litigation
(including a cross-claim or counterclaim in a lawsuit) alleging that
any patent claim is infringed by making, using, selling, offering for
sale, or importing the Program or any portion of it.

  11. Patents.

  A "contributor" is a copyright holder who authorizes use under this
License of the Program or a work on which the Program is based.  The
work thus licensed is called the contributor's "contributor version".

  A contributor's "essential patent claims" are all patent claims
owned or controlled by the contributor, whether already acquired or
hereafter acquired, that would be infringed by some manner, permitted
by this License, of making, using, or selling its contributor version,
but do not include claims that would be infringed only as a
consequence of further modification of the contributor version.  For
purposes of this definition, "control" includes the right to grant
patent sublicenses in a manner consistent with the requirements of
this License.

  Each contributor grants you a non-exclusive, worldwide, royalty-free
patent license under the contributor's essential patent claims, to
make, use, sell, offer for sale, import and otherwise run, modify and
propagate the contents of its contributor version.

  In the following three paragraphs, a "patent license" is any express
agreement or commitment, however denominated, not to enforce a patent
(such as an express permission to practice a patent or covenant not to
sue for patent infringement).  To "grant" such a patent license to a
party means to make such an agreement or commitment not to enforce a
patent against the party.

  If you convey a covered work, knowingly relying on a patent license,
and the Corresponding Source of the work is not available for anyone
to copy, free of charge and under the terms of this License, through a
publicly available network server or other readily accessible means,
then you must either (1) cause the Corresponding Source to be so
available, or (2) arrange to deprive yourself of the benefit of the
patent license for this particular work, or (3) arrange, in a manner
consistent with the requirements of this License, to extend the patent
license to downstream recipients.  "Knowingly relying" means you have
actual knowledge that, but for the patent license, your conveying the
covered work in a country, or your recipient's use of the covered work
in a country, would infringe one or more identifiable patents in that
country that you have reason to believe are valid.
```

```
   If, pursuant to or in connection with a single transaction or
arrangement, you convey, or propagate by procuring conveyance of, a
covered work, and grant a patent license to some of the parties
receiving the covered work authorizing them to use, propagate, modify
or convey a specific copy of the covered work, then the patent license
you grant is automatically extended to all recipients of the covered
work and works based on it.

   A patent license is "discriminatory" if it does not include within
the scope of its coverage, prohibits the exercise of, or is
conditioned on the non-exercise of one or more of the rights that are
specifically granted under this License.  You may not convey a covered
work if you are a party to an arrangement with a third party that is
in the business of distributing software, under which you make payment
to the third party based on the extent of your activity of conveying
the work, and under which the third party grants, to any of the
parties who would receive the covered work from you, a discriminatory
patent license (a) in connection with copies of the covered work
conveyed by you (or copies made from those copies), or (b) primarily
for and in connection with specific products or compilations that
contain the covered work, unless you entered into that arrangement,
or that patent license was granted, prior to 28 March 2007.

   Nothing in this License shall be construed as excluding or limiting
any implied license or other defenses to infringement that may
otherwise be available to you under applicable patent law.

   12. No Surrender of Others' Freedom.

   If conditions are imposed on you (whether by court order, agreement or
otherwise) that contradict the conditions of this License, they do not
excuse you from the conditions of this License.  If you cannot convey a
covered work so as to satisfy simultaneously your obligations under this
License and any other pertinent obligations, then as a consequence you may
not convey it at all.  For example, if you agree to terms that obligate you
to collect a royalty for further conveying from those to whom you convey
the Program, the only way you could satisfy both those terms and this
License would be to refrain entirely from conveying the Program.

   13. Remote Network Interaction; Use with the GNU General Public License.

   Notwithstanding any other provision of this License, if you modify the
Program, your modified version must prominently offer all users
interacting with it remotely through a computer network (if your version
supports such interaction) an opportunity to receive the Corresponding
Source of your version by providing access to the Corresponding Source
from a network server at no charge, through some standard or customary
means of facilitating copying of software.  This Corresponding Source
shall include the Corresponding Source for any work covered by version 3
of the GNU General Public License that is incorporated pursuant to the
following paragraph.

   Notwithstanding any other provision of this License, you have
permission to link or combine any covered work with a work licensed
under version 3 of the GNU General Public License into a single
combined work, and to convey the resulting work.  The terms of this
```

```
License will continue to apply to the part which is the covered work,
but the work with which it is combined will remain governed by version
3 of the GNU General Public License.

  14. Revised Versions of this License.

  The Free Software Foundation may publish revised and/or new versions of
the GNU Affero General Public License from time to time.  Such new versions
will be similar in spirit to the present version, but may differ in detail to
address new problems or concerns.

  Each version is given a distinguishing version number.  If the
Program specifies that a certain numbered version of the GNU Affero General
Public License "or any later version" applies to it, you have the
option of following the terms and conditions either of that numbered
version or of any later version published by the Free Software
Foundation.  If the Program does not specify a version number of the
GNU Affero General Public License, you may choose any version ever published
by the Free Software Foundation.

  If the Program specifies that a proxy can decide which future
versions of the GNU Affero General Public License can be used, that proxy's
public statement of acceptance of a version permanently authorizes you
to choose that version for the Program.

  Later license versions may give you additional or different
permissions.  However, no additional obligations are imposed on any
author or copyright holder as a result of your choosing to follow a
later version.

  15. Disclaimer of Warranty.

  THERE IS NO WARRANTY FOR THE PROGRAM, TO THE EXTENT PERMITTED BY
APPLICABLE LAW.  EXCEPT WHEN OTHERWISE STATED IN WRITING THE COPYRIGHT
HOLDERS AND/OR OTHER PARTIES PROVIDE THE PROGRAM "AS IS" WITHOUT WARRANTY
OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, BUT NOT LIMITED TO,
THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR
PURPOSE.  THE ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE PROGRAM
IS WITH YOU.  SHOULD THE PROGRAM PROVE DEFECTIVE, YOU ASSUME THE COST OF
ALL NECESSARY SERVICING, REPAIR OR CORRECTION.

  16. Limitation of Liability.

  IN NO EVENT UNLESS REQUIRED BY APPLICABLE LAW OR AGREED TO IN WRITING
WILL ANY COPYRIGHT HOLDER, OR ANY OTHER PARTY WHO MODIFIES AND/OR CONVEYS
THE PROGRAM AS PERMITTED ABOVE, BE LIABLE TO YOU FOR DAMAGES, INCLUDING ANY
GENERAL, SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES ARISING OUT OF THE
USE OR INABILITY TO USE THE PROGRAM (INCLUDING BUT NOT LIMITED TO LOSS OF
DATA OR DATA BEING RENDERED INACCURATE OR LOSSES SUSTAINED BY YOU OR THIRD
PARTIES OR A FAILURE OF THE PROGRAM TO OPERATE WITH ANY OTHER PROGRAMS),
EVEN IF SUCH HOLDER OR OTHER PARTY HAS BEEN ADVISED OF THE POSSIBILITY OF
SUCH DAMAGES.

  17. Interpretation of Sections 15 and 16.

  If the disclaimer of warranty and limitation of liability provided
above cannot be given local legal effect according to their terms,
```

```
reviewing courts shall apply local law that most closely approximates
an absolute waiver of all civil liability in connection with the
Program, unless a warranty or assumption of liability accompanies a
copy of the Program in return for a fee.

                     END OF TERMS AND CONDITIONS

            How to Apply These Terms to Your New Programs

  If you develop a new program, and you want it to be of the greatest
possible use to the public, the best way to achieve this is to make it
free software which everyone can redistribute and change under these terms.

  To do so, attach the following notices to the program.  It is safest
to attach them to the start of each source file to most effectively
state the exclusion of warranty; and each file should have at least
the "copyright" line and a pointer to where the full notice is found.

    <one line to give the program's name and a brief idea of what it does.>
    Copyright (C) <year>  <name of author>

    This program is free software: you can redistribute it and/or modify
    it under the terms of the GNU Affero General Public License as published
    by the Free Software Foundation, either version 3 of the License, or
    (at your option) any later version.

    This program is distributed in the hope that it will be useful,
    but WITHOUT ANY WARRANTY; without even the implied warranty of
    MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.  See the
    GNU Affero General Public License for more details.

    You should have received a copy of the GNU Affero General Public License
    along with this program.  If not, see <http://www.gnu.org/licenses/>.

Also add information on how to contact you by electronic and paper mail.

  If your software can interact with users remotely through a computer
network, you should also make sure that it provides a way for users to
get its source.  For example, if your program is a web application, its
interface could display a "Source" link that leads users to an archive
of the code.  There are many ways you could offer source, and different
solutions will be better for different programs; see section 13 for the
specific requirements.

  You should also get your employer (if you work as a programmer) or school,
if any, to sign a "copyright disclaimer" for the program, if necessary.
For more information on this, and how to apply and follow the GNU AGPL, see
<http://www.gnu.org/licenses/>.
```

Python Module Index

a

l

Index