Ansible Syntactic Sugar ("a2s")

Contents:

a2s is a non-official Syntactic Sugar for Ansible, which without extensions already is an *"Simple, agentless IT automation that anyone can use"*.

Contents:

CHAPTER 1

Installation

If you are new to Ansible, we recommend read our *Installation guides*. TL;DR:

```
# "a2s is installable as an Ansible role and is distributed over Ansible Galaxy
ansible-galaxy install fititnt.syntactic_sugar
```

Quickstart

And then, look at our *Ansible Syntactic Sugar APIs* and *a2s Example Playbooks*. TL;DR:

Listing 1: playbook-basic.yml

```
1   # TODO: improve this quickstart example (fititnt, 2020-01-17 01:11 BRT)
2   # $ ansible-playbook -i myserver.com, playbook-basic.yml
3
4   - hosts: all
5     remote_user: root
6     vars:
7
8       # Create operational system group (if already does not exist)
9       a2s_groups:
10        - name: 'www-data'
11
12      # Create operational system users
13      a2s_users:
14        - name: app
15          groups: ['www-data']
16
17      # Create/delete directories
18      a2s_directories:
19        - path: /var/www/my-app
20          owner: app
21          group: www-data
22        - path: /var/www/my-old-app-folder-to-delete
23          state: absent
24
25      # Install PHP with packages that allow run popular softwares like
26      # Drupal, Joomla, Wordpress & Laravel
27      a2s_install_php:
28        - php7.4-fpm
29        - php7.4-common
30        - php7.4-mbstring
```

(continues on next page)

```
31        - php7.4-mysql
32        - php7.4-bcmath
33        - php7.4-curl
34        - php7.4-gd
35        - php7.4-xml
36        - php-imagick
37
38      # Uncomment next variable only for Windows hosts.
39      # a2s_iswindows: true
40    roles:
41      - { role: fititnt.syntactic_sugar }
```

But playbook-basic.yml, while may help you to undestand a2s (or even Ansible) may be less powerfull than a2s offer. One tricky that a2s uses to avoid be too optionated on defaults is simply create an equivalent to most `a2s_APINAME` as `a2s_APINAME_defaults` and suggest you to keep near your `a2s_APINAME` definitions:

Listing 2: playbook.yml

```
1   # TODO: improve this quickstart example (fititnt, 2020-01-17 01:11 BRT)
2   # $ ansible-playbook -i myserver.com, playbook.yml
3
4   - hosts: all
5     remote_user: root
6     vars:
7
8       # Create operational system group (if already does not exist)
9       a2s_groups:
10        - name: 'www-data'
11
12      # Create operational system users (with some defaults)
13      a2s_users_defaults:
14        groups: ['www-data']
15
16      a2s_users:
17        - name: app
18        - name: drupal-site
19        - name: joomla-site
20        - name: laravel-site
21        - name: wordpress-site
22        - name: fititnt
23          authorized_keys:
24            - key: https://github.com/fititnt.keys
25
26      # Create folders (with some defaults)
27      # "{{ item.myspecialvar }}" is an example of valid default
28      a2s_directories_defauts:
29        path: "/var/www/{{ item.myspecialvar }}"
30        owner: "{{ item.myspecialvar }}"
31        group: www-data
32        mode: '0755'
33        state: present
34
35      a2s_directories:
36        - path: /var/www/my-app
37          owner: app
38        - path: /var/www/my-old-app-folder-to-delete
```

```
39          state: absent
40        - myspecialvar: "drupal-site"
41        - myspecialvar: "joomla-site"
42        - myspecialvar: "laravel-site"
43        - myspecialvar: "wordpress-site"
44
45
46        # Install PHP with packages that allow run popular softwares like
47        # Drupal, Joomla, Wordpress & Laravel
48        a2s_install_php: "{{ a2s__php74 }}"
49
50        # Uncomment next variable only for Windows hosts.
51        # a2s_iswindows: true
52      roles:
53        - { role: fititnt.syntactic_sugar }
```

Important: a2s by design have to explicitly choose not respect some specific Ansible best practices related to large scale projects to allow implement more even more concise Infrastructure As Code. An notable anti-pattern is a2s allowing you bootstrap some features that would be better be done to additional Asible Role with *a2s_templates*.

a2s by design may implement implement features that would never be accepted on the Ansible standard modules because would be redundant with more lines of code or would be better implemented by external extensions.

We may explicitly warn you in some cases, and even help you on how to migrate from a2s to more enterprise approach. We by default namespace our rules using a2s_ to make it easier.

Dedicated to the public domain

The a2s codebase is dedicated to public domain to estimulate your creativity and allow sub parts of a2s be reused freely.

2.1 Quickstart

Warning: This page is a draft.

Table of Contents of APIs

2.1.1 New to Ansible

- https://www.ansible.com/resources/videos/quick-start-video

2.1.2 `a2s` step by step

Topico 1

Listing 3: todo.yml

```
1  # TODO...
```

Topico 2

Listing 4: todo2.yml

```
1  # TODO...
```

2.2 Why a2s may be useful for you

> **Warning:** a2s still an apha release. This page is not complete.

Table of Contents of APIs

- *Why a2s may be useful for you*
    - *Who are likely to NOT need* `a2s`
        - *Who don't need to deploy changes on servers*
        - *Who don't need (and do not want learn) how to automate IT*
    - *I'm new to IT automation*
        - *Ansible is a friendly IT automation tool*
    - *I'm new to Ansible*
        - `a2s` *as jack of all trades Ansible Role*
        - `a2s` *have great examples of common tasks*
    - *I'm a experienced Ansible user*
        - `a2s` *only when you have no better option*
        - `a2s...`
    - *I'm a Ansible extensions developer*
        - *a2s is dedicated to public domain*
        - `a2s...`

2.2.1 Who are likely to NOT need `a2s`

Who don't need to deploy changes on servers

> **Warning:** This pages assumes that you or already work or want to learn how to automate IT and already have some experience on how to do without automation. A great example if you are someone who would be able implement alone (ok use help of search engines) tutorials like How To Install WordPress with LAMP on Ubuntu 18.04 but now want to abstract/automate this.
>
> If you already use software that someone else configure the operational system for you, it's very likely that **at this moment** `a2s` may not be useful.

Who don't need (and do not want learn) how to automate IT

Ansible can be used even for who does not work as sysadmin, DevOps or IT infrastructure in general and, compared to alternatives, may actually save your time even if you don't work full time on a position that benefits from automation.

> TODO: comment more about (fititnt, 2020-01-21 02:53 BRT)

2.2.2 I'm new to IT automation

> TODO: comment more about (fititnt, 2020-01-21 01:26 BRT)

Ansible is a friendly IT automation tool

"It turned out our group of 15 or so people spent about six months trying to automate OpenStack with Puppet, and I spent a lot of time banging my head into a desk. (Later, once AnsibleWorks was founded, one of our Solutions Architects built content to automate OpenStack, similar in scale, in a single week, by himself!)"

–Michael DeHaan - In The origins of Ansible

2.2.3 I'm new to Ansible

> **Tip:** Even if you are very skilled, know some basic automation ("bash history", docker, etc or even Chef/Puppet/Saltstack) **but are in a hurry to implement an infrastructure in production** (think like have just a 2~5 days), in special if this infrastructure it not simple (could be replaced later if you on a code refactoring), dont use `a2s`.

Ok, you can use, but `a2s` starting point. `a2s` can make some implementations simpler, in ones related to filesystem and users/groups manangement.

But in a very short time to learn both Ansible and implement something complex like (here just as example) clusters of FreeIPA, GlusterFS & Galera Cluster (MariaDB/MySQL) you may find ready to use Ansible Roles *near plug and play*. Check out https://galaxy.ansible.com/.

`a2s` as jack of all trades Ansible Role

`a2s` have great examples of common tasks

2.2.4 I'm a experienced Ansible user

`a2s` only when you have no better option

`a2s ...`

2.2.5 I'm a Ansible extensions developer

a2s is dedicated to public domain

In short: while atribution is welcome from ansible users (most because it may help to find documentation) you as open source contributor is free to take take parts of a2s source code without mention.

`a2s ...`

2.3 Installation guides

```
# Too long didn't read:
# "a2s is installable as an Ansible role and is distributed over Ansible Galaxy
ansible-galaxy install fititnt.syntactic_sugar
```

a2s extends **Ansible**. Ansible uses **Python** on the "control node" (for example your notebook running Linux, Mac or Windows via WSL) to control remote machines over **SSH** (these machines could be Windows, Linux, BSD, Raspbery Pi, your home router, etc).

Both **a2s** and Ansible does not require to be installed on controlled nodes, but they require that the control node can be acessed over SSH and have any version of Python installed (even older Python 2 would work fine, the versions installed on control node are more important).

2.3.1 Python

Note: Maybe you already have Python installed on your system. And even if you don't have maybe by installing Ansible your package mananger will install python for you.

Tip: If you system already does not have python installed, but you could choose between python 2 or python 3, **please select python 3**. Ansible actually can work on python 2, but it's older.

Since you already is installing python 3, if some guide guide explain how to install pip3, its may be a good idea do it. Neither Ansible or a2s require pip eventually you may want to install pip and acidentaly install pip for python 2 instead of `pip3` because most guides may teach you do with the older one.

You can check how to install python via the official Python documentation. But maybe is easier to just look on your prefered search engine for "how to install python 3 on *name of my operational system*",

2.3.2 Ansible

Note: TODO: we should document some basic explanation on how to install Ansible

Check the complete guide on how to install Ansible.

2.3.3 Ansible Galaxy

Note: TODO: we should document some basic explanation on how to install Ansible Galaxy

2.3.4 Ansible Syntactic Sugar ("a2s")

`a2s` is installable as one Ansible role. You can install the lastest version with the command:

```
ansible-galaxy install fititnt.syntactic_sugar
```

Note: Ansible roles does not automatic update without you explicitly force this. This behavior may actually not be a bad default behavior.

2.4 Philosophy

Note: `a2s` still a pre-release. But a TL;DR: to explain the idea behind is *"'a2s' be to Ansible what jQuery is to JavaScript"*. So it means:

- `a2s` still YAML and still Ansible.
    - It should not require Python knowledge for one averange Pull Request with a new feature.
    - Ideally should be easy to reuse parts of this code by others.
- `a2s` **"most common data structure"** is a List **"of something"** to be executed in order
    - If **"this something"** is an abstraction to an Ansible Module
        * The name should be plural form
        * The same way Ansible modules may implement parameter aliases (e.g. file_module have dest and name as aliases for path) , `a2s` may implement entire API aliases (e.g a2s_directories, a2s_links are alises to file_module).
        * We point to the official documentation instead of repeating all options, and (if any) document what a2s add to the underlining API.
        * if uses more than one API, both documentation and execution should mention.
    - If **"this something"** is a very customized implementation, it should explain all options. And very likely warn the user if is an API that could be changed on a next major version
- `a2s` should be *"be otimized for humans"* and simpler usages.
    - It does not means that more than simpler usages cannot be otimized, but they may require read more documentation, in special ones related on how to run just part of the a2s per play run.

– What may be more complex than simple to medium implementation is out of scope and we recommend users to use dedicated Ansible roles.

• `a2s` should be namespaced to avoid conflict with Ansible underling modules, with other Ansible Roles and maybe even with more than one version of itself (or someone elses fork) installed.

– *How* this could be done, we're open to ideas.

– `a2s` should respect SemVer and may have major versions more often than most averange software.

2.4.1 Overview of a2s APIs

a2s, released as Ansible role, when not restricted with special variables, will convert the instructions of your Ansible inventory defined on variables started with `a2s_`.

With some exceptions, most of the time, you can expect that:

• **Naming**

– `a2s_APINAMEs` that end with letter `s` accept a list of values, all the others may accept a single value, that may be a string, boolean or a dictionaly (e.g. an "object" like `{ key: 'valye', key2: 'value2 })`

• **Standard APIs reuse same Ansible documentation**

– If you know that the `APINAME` on an `a2s_APINAME(s)` is have and exact Ansible core module, this a2s API will implement the same parameters from the official API

– a2s by default will ignore instead of fail if a param that does not exist on the underlinging Ansible module was defined. Please pay attention to wrong paramenters names.

• **Idempotent by default**

– Using a2s will have at least the the same idempotency of the Ansible underlining modules (which already is very good)

– *"An operation is idempotent if the result of performing it once is exactly the same as the result of performing it repeatedly without any intervening actions. –* Ansible Glossary for idempotency

2.4.2 Antipatterns of a2s

Note: Undocumented at this moment.

2.5 Ansible Syntactic Sugar APIs

Warning: a2s still an apha release. This page is not complete.

Table of Contents of APIs

• *Ansible Syntactic Sugar APIs*

> * *a2s_except_apis_groups*
>
> * *a2s_only_apis*
>
> * *a2s_only_apis_groups*

2.5.1 Standard APIs

a2s_copies

Note: Undocumented at this moment. Check source code at <https://github.com/fititnt/ansible-syntactic-sugar/blob/master/tasks/copy/main.yml>

---

a2s_crons

Note: Undocumented at this moment. Check source code at <https://github.com/fititnt/ansible-syntactic-sugar/blob/master/tasks/cron/main.yml>

---

a2s_directories

- **Short Description**: *Manange directories*

- **Ansible Modules**: - file_module

- **Type of values**: list of dictionaries accepted by the underlining Ansible module. state value will default directory. Other missing values will be ommited.

Listing 5: a2s_directories_example.yml

```yaml
# Create directories. a2s_directories_default allows create custom variables
# like (var1 in item.var1) or reuse one variable (like owner in item.owner)
a2s_directories_default:
  path: "/var/www/html/{{ item.var1 }}/"
  owner: "{{ item.var1 }}"
  group: "www-data"

a2s_directories:
  - path: /var/www/html
    recurse: yes
    owner: root
    group: root
    var1: null
  - var1: "app"
  - var1: "site-a"
  - var1: "site-b"
  - var1: "cdn-site-a"
  - var1: "cdn-site-b"
```

a2s_etchosts

- **Short Description**: *List of strings to be added on /etc/hosts file. The current implementation will not will not replace older values.*

- **Ansible Modules**: - None. This is a custom implementation.

- **Type of values**: List of strings.

```
a2s_etchosts:
  - "127.0.0.1  site-a.local"
  - "198.51.100.0  example.org example.com"
```

Note: `a2s_etchosts` is very likely to be improved before a2s stable release.

a2s_fetches

- **Short Description**: *Fetch files from remote machines and storing them locally in a file tree, organized by hostname.*

- **Ansible Modules**: fetch_module

- **Type of values**: list of dictionaries accepted by the underlining Ansible module. Missing values will be ommited.

- **Special defaults override**: `a2s_fetches_default`

Listing 6: a2s_fetches_example.yml

```
1  a2s_fetches_default:
2    dest: /backup/
3  a2s_fetches:
4    - src: /etc/ssh/ssh_host_rsa_key # File required on a new node to keep the␣
   ↪fingerprint
5    - src: /home/root/.ssh/id_rsa
6    - src: /home/root/.ssh/id_rsa.pub
7
8  ## For each host, files will be downloaded prefixed with inventory name
9  ## Paths will be respected. E.g for node "host.example.com" you get:
10 # /backup/host.example.com/etc/ssh/ssh_host_rsa_key
11 # /backup/host.example.com/home/root/.ssh/id_rsa
12 # /backup/host.example.com/home/root/.ssh/id_rsa.pub
```

a2s_files

- **Short Description**: *Manange files. Exlusive feature: Allows create file content from string*

- **Ansible Modules**: file_module and win_file_module or, if `a2s_files[n]content` is defined, copy_module and win_copy_module

- **Type of values**: list of dictionaries accepted by the underlining Ansible module. Other missing values will be ommited.

- **Special defaults override**: `a2s_files_default`

You can consider `a2s_files` with the same parameters of Ansible file_module & win_file_module. The only exception to this rule is if you define a non-empty value to `a2s_files[n]content:` in this case `a2s_files` will use Ansible copy_module & win_copy_module.

Listing 7: a2s_files_example.yml

```
1  a2s_files_default:
2    group: "www-data"
3
4  a2s_files:
5    - path: "/var/www/html/app/index.html"
6      content: "Hello world, app!"
7      owner: "app"
```

a2s_groups

- **Short Description**: *group – Add or remove groups / win_group – Add and remove local groups*
- **Ansible Modules**: - [group](https://docs.ansible.com/ansible/latest/modules/group_module.html) - [win_group](https://docs.ansible.com/ansible/latest/modules/win_group_module.html)
- **Type of values**: list of dictionaries accepted by the underlining Ansible modules. Missing values will be ommited.
- **Examples**

```
a2s_groups:
  - name: "www-data"
  - name: "haproxy"
  - name: "user-to-remove"
    state: "absent"
```

a2s_hostname

- **Short Description**: *hostname – Manage hostname, enforcing [RFC822](https://www.w3.org/Protocols/rfc822/ format to avoid fail*
- **Ansible Modules**: - [hostname](https://docs.ansible.com/ansible/latest/modules/hostname_module.html)
- **Type of values**: Dictionary. Same values of Ansible module hostname.

To add to /etc/hosts, check `a2s_etchosts`(#a2s_etchosts).

a2s_mysql_dbs

Note: Undocumented at this moment. Check source code at <https://github.com/fititnt/ansible-syntactic-sugar/blob/master/tasks/mysql/main.yml>

a2s_mysql_users

Note: Undocumented at this moment. Check source code at <https://github.com/fititnt/ansible-syntactic-sugar/blob/master/tasks/mysql/main.yml>

`a2s_templates`

Note: Undocumented at this moment. Check source code at <https://github.com/fititnt/ansible-syntactic-sugar/blob/master/tasks/template/main.yml>

`a2s_users`

- **Short Description**: *user – Manage user accounts / win_user – Manages local Windows user accounts*

- **Ansible Modules**: - [user](https://docs.ansible.com/ansible/latest/modules/user_module.html) - [win_user](https://docs.ansible.com/ansible/latest/modules/win_user_module.html)

- **Type of values**: list of dictionaries accepted by the underlining Ansible modules. Missing values will try `a2s_users_default`, then will be ommited.

- **Special defaults override**: `a2s_users_default`

Listing 8: a2s_users_example.yml

```yml
a2s_users_default:
  create_home: no
  groups: ["www-data"]

a2s_users:
  - name: "app"
  - name: "site-a"
  - name: "site-b"
  - name: "cdn-site-a"
    groups: ["cdns", "www-data"]
  - name: "cdn-site-b"
    groups: ["cdns", "www-data"]
  - name: "php-demo"
  - name: "fititnt"
    create_home: yes
    authorized_keys:
      - key: https://github.com/fititnt.keys
```

`a2s_users[n]authorized_keys`

- **Short Description**: *authorized_key – Adds or removes an SSH authorized key*

- **Ansible Modules**: - [authorized_key](https://docs.ansible.com/ansible/latest/modules/authorized_key_module.html)

- **Type of values**: list of dictionaries accepted by the underlining Ansible modules. Missing required `user` (`a2s_users[n]authorized_keys[m]user`) will default to `name` (`a2s_users[n]name`). Other missing values will be ommited

2.5.2 Extra APIs

`a2s_install_adminer`

`a2s_rclone`

2.5.3 Devel APIs

`a2s_devel_nginx`

Install NGinx with `a2s_devel_nginx:   true`. Used for testing.

Note: This is an internal feature and may not be documented because is not planned to be used for end users. The source code is at: <https://github.com/fititnt/ansible-syntactic-sugar/blob/master/tasks/devel-nginx/main.yml>

Example:

```
- hosts: all
  remote_user: root
  vars:

    a2s_devel_nginx: true

  roles:
    - { role: fititnt.syntactic_sugar }
```

2.5.4 Installers

`a2s_install_adminer`

- **Short Description**: *Install [adminer](https://www.adminer.org/), "Database management in a single PHP file"*
- **Default**: `undefined`
- **Type of value**: Dictionary (name, state)
- **Examples of values**: `{{ a2s__adminer }}`

Variable `{{ a2s__adminer }}` are a *syntactic sugar* for the default parameters

`a2s_install_composer`

- **Short Description**: *Install [composer](https://getcomposer.org/) required by [Ansible composer](https://docs.ansible.com/ansible/latest/modules/composer_module.html) by default on global path*
- **Ansible Modules**: - None. Custom implementation.
- **Type of values**: Boolean, Dictionary

All these examples have the same effect (install on global scope)

```
a2s_install_composer: true
```

```
a2s_install_composer: "{{ a2s__composer }}"
```

```
a2s_install_composer:
 - path: /usr/local/bin
   user: root
   force: false # true force reinstall
   version: '' # use custom version to install
   php: 'php' # php binary to use. If is not 'php' customize here
```

a2s_install_composers

- **Short Description**: *Install [composer](https://getcomposer.org/) required by [Ansible composer](https://docs.ansible.com/ansible/latest/modules/composer_module.html) for more than one user*

- **Ansible Modules**: - None

- **Type of values**: List of Dictionaries

```
a2s_install_composers:
 - "{{ a2s__composer }}" # Global, as root
 - path: '/home/user1/bin'
   user: user1
 - path: '/home/user2/bin'
   user: user2
```

a2s_install_php

- Default: `undefined`

- Type of value: List of Strings; List of Objects (name, state)

- Examples of values: `{{ a2s__php74 }}`, `{{ a2s__php73 }}`, `{{ a2s__php72 }}`, `['php-fpm', 'php-common']` [sup](assumes state: present)[/sup] `[{name: 'php-fpm', state: 'present'}, name: 'php-mssql', state: 'absent']`

> Install a list of PHP packages on the system

Variables `a2s__php74`, `a2s__php73`, `a2s__php72`... are a *syntactic sugar* to install common packages to run Wordpress, Joomla, Drupal and laravel.

In Ansible is possible append arrays values with + (objects you use `| combine()`), e.g `a2s_php_install: "{{ a2s__php74 + ['php7.4-dev', 'php7.4-ldap'] }}"`

2.5.5 Groups of a2s APIs

a2s-devel

a2s-extra

database

files

notifications

packaging-languages

---

packaging-os

system

2.5.6 Control a2s workflow

a2s_except_apis

a2s_except_apis_groups

a2s_only_apis

a2s_only_apis_groups

2.6 Ansible Syntactic Sugar Constants

> **Warning:** a2s still an apha release. This page is not complete.

2.7 a2s Example Playbooks

2.7.1 Minimal Playbook

a2s_

Note: If you run this role without explicitly use any [Public APIs](#public-apis) (variables starte with `a2s_` that are not defaults) this Ansible role will make no changes on your system.

```
# Note: this playbook will run without errors, but will not make changes. You
#       need to specify variables
- hosts: all
  roles:
    - { role: fititnt.syntactic_sugar }
```

2.7.2 *Run only these APIs* playbook example

a2s have so many features that you play may become bigger. As explained in [Ansible Playbook Tags](https://docs.ansible.com/ansible/latest/user_guide/playbooks_tags.html) the Ansible default behavior do not allow run only a subset of a Role withtout using –tags / –skip-tags from command line:

> *"There is no way to 'import only these tags'; you probably want to split into smaller roles/includes if you find yourself looking for such a feature."*

> —Ansible Playbook Tags

BUT with a2s you can!. We provide one workaround using an special variable `a2s_only`. The best way to use is not define `a2s_only` on your inventory, but only when importing this role.

```
# On this a2s example, even if your inventory have more than a2s_groups and
# a2s_users defined and not using command line --tags, only a subset of a2s will
# run
- hosts: all
  role: fititnt.syntactic_sugar
    vars:
    a2s_only_apis:
        - a2s_groups
        - a2s_users
```

```
# Instead of using a2s_only, is possible to specify what a2s apis to not run.
# On this example ones related to database are skipped
- hosts: all
  role: fititnt.syntactic_sugar
    vars:
      a2s_except_apis:
        - a2s_mysql_dbs
        - a2s_mysql_users
```

2.7.3 Playbook using several Public APIs

```
- hosts: all
  remote_user: root
  vars:

    a2s_directories:
      - path: /var/www/my-app
        owner: app
        group: www-data
        mode: '0755'
      - path: /var/www/my-old-app-folder-to-delete
        state: absent

    a2s_groups:
      - name: group1
      - name: group2

    a2s_users:
      - name: user1
      - name: user2
        groups:
          - group2
      - name: fititnt
        authorized_keys:
          key: https://github.com/fititnt.keys

    # a2s_iswindows: true # Uncomment next variable only for Windows hosts.
  roles:
    - { role: fititnt.syntactic_sugar }
```

2.7.4 Playbook full example with Continuos Integration and testinfra

Check the contents of molecule/default/playbook.yml and our Travis-CI panel at <https://travis-ci.com/fititnt/ap-application-load-balancer-extras>.

If you are not developing an interface that other people depend on or have a very serius infrastructure or several people making changes, well written Ansible playbooks may already be great. If this is not sufficient, check our testinfra tests file at molecule/default/tests/test_default.py.

2.8 a2s TODO

Note: This page is a draft. May change later.

2.8.1 Files modules

https://docs.ansible.com/ansible/latest/modules/archive_module.html

ini_file_module

https://docs.ansible.com/ansible/latest/modules/ini_file_module.html

unarchive_module

https://docs.ansible.com/ansible/latest/modules/unarchive_module.html#unarchive-module

2.8.2 Other / Ungrouped

htpasswd_module

https://docs.ansible.com/ansible/latest/modules/htpasswd_module.html

Maybe these ones:

git_module https://docs.ansible.com/ansible/latest/modules/git_module.html#git-module

2.9 a2s alternatives

Note: This page is a draft. May change later.

2.9.1 Ansible without plugins

2.9.2 suitable

- https://www.reddit.com/r/ansible/comments/8lguqj/anyone_using_the_python_api_wrapper_named_suitable/

- https://suitable.readthedocs.io/en/latest/

2.9.3 redhat-cop/infra-ansible

Description from mainainers: ***Tooling / Ansible to support the many aspects of infrastructure installation, setup and configuration.***

- https://github.com/redhat-cop/infra-ansible

2.10 Indices and tables

- genindex

- search