
Ansible Syntactic Sugar ("a2s")

Jan 18, 2020

Contents:

1	Python	5
2	Ansible	7
3	Ansible Galaxy	9
4	Ansible Syntactic Sugar (“a2s”)	11
5	a2s philosophy	13
5.1	Overview of a2s APIs	13
5.2	Antipatterns of a2s	14
6	a2s APIs	15
6.1	Overview of a2s APIs	15
6.2	Standard APIs	16
6.2.1	a2s_copies	16
6.2.2	a2s_crons	16
6.2.3	a2s_directories	16
6.2.4	a2s_groups	16
6.2.5	a2s_hostname	17
6.2.6	a2s_mysql_dbs	17
6.2.7	a2s_mysql_users	17
6.2.8	a2s_templates	17
6.2.9	a2s_users	17
6.2.10	a2s_users[n]authorized_keys	17
6.3	Extra APIs	18
6.3.1	a2s_etchosts	18
6.4	Installers	18
6.4.1	a2s_install_adminer	18
6.4.2	a2s_install_composer	18
6.4.3	a2s_install_composers	19
6.4.4	a2s_install_php	19
6.5	Devel APIs	19
7	a2s Example Playbooks	21
7.1	Minimal Playbook	21
7.2	Run only these APIs playbook example	21

7.3	Playbook using all Public APIs	22
7.4	Playbook full example with Continuous Integration and testinfra	22
8	a2s TODO	25
8.1	Files modules	25
8.1.1	fetch_module	25
8.1.2	file_module	25
8.1.3	ini_file_module	25
8.1.4	unarchive_module	25
8.2	Other / Ungrouped	26
8.2.1	htpasswd_module	26
8.2.2	known_hosts_module	26
9	a2s alternatives	27
9.1	Ansible without plugins	27
9.2	suitable	27
10	Indices and tables	29

a2s is a non-official [Syntactic Sugar](#) for [Ansible](#), which without extensions already is an “*Simple, agentless IT automation that anyone can use*”.

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
```

And then, look at our *a2s 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
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 understand a2s (or even Ansible) may be less powerful 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_defaults:
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      # Install PHP with packages that allow run popular softwares like
46      # Drupal, Joomla, Wordpress & Laravel
47      a2s_install_php: "{{ a2s__php74 }}"
48
49      # Uncomment next variable only for Windows hosts.
50      # a2s_iswindows: true
51    roles:
```

(continues on next page)

(continued from previous page)

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 stimulate your creativity and allow sub parts of a2s be reused freely.

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 accessed 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).

CHAPTER 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 manager will install python for you.

Tip: If your 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 are installing python 3, if some guide explains how to install pip3, it may be a good idea to do it. Neither Ansible or a2s require pip eventually you may want to install pip and accidentally install pip for python 2 instead of pip3 because most guides may teach you to do with the older one.

You can check how to install python via the [official Python documentation](#). But maybe it's easier to just look on your preferred search engine for “how to install python 3 on *name of my operational system*”,

CHAPTER 2

Ansible

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

Check the [complete guide](#) on how to install Ansible.

CHAPTER 3

Ansible Galaxy

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

Ansible Syntactic Sugar (“a2s”)

a2s is installable as one Ansible role. You can install the latest 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.

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

5.1 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 dictionary (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 underling 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](#)

5.2 Antipatterns of a2s

Note: Undocumented at this moment.

Error: This documentation is incomplete and with obvious markup issues. May take some time to finish the migration from GitHub page README.md to ReadTheDocs and also fix some issues with Sphinx (like how to improve the usability of the navigation bar) (fitint, 2020-01-18 03:27 BRT)

6.1 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 dictionary (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 underling 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](#)”

6.2 Standard APIs

6.2.1 a2s_copies

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

6.2.2 a2s_crons

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

6.2.3 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.
- **Examples**

```
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
```

6.2.4 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"
```

6.2.5 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).

6.2.6 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>>

6.2.7 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>>

6.2.8 a2s_templates

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

6.2.9 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 be omitted.

6.2.10 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 omitted

6.3 Extra APIs

6.3.1 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.

6.4 Installers

6.4.1 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

6.4.2 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
```

6.4.3 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
```

6.4.4 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']` ^(assumes state: present) `[{name: 'php-fpm', state: 'present'}, name: 'php-mysql', 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'] }}"`

6.5 Devel APIs

No documented Devel APIs at this moment.

a2s Example Playbooks

7.1 Minimal Playbook

a2s_

Note: If you run this role without explicitly use any [Public APIs](#public-apis) (variables start 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 }
```

7.2 Run only these APIs playbook example

a2s have so many features that your 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 does not allow running only a subset of a Role without using `-tags` / `--skip-tags` from the 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 a 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 has more than a2s_groups and
# a2s_users defined and not using command line --tags, only a subset of a2s will
```

(continues on next page)

(continued from previous page)

```
# run
- hosts: all
  role: fititnt.syntactic_sugar
  vars:
    a2s_only:
      - 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:
      - a2s_mysql_dbs
      - a2s_mysql_users
```

7.3 Playbook using all 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 }
```

7.4 Playbook full example with Continuous 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 serious 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]`(`molecule/default/tests/test_default.py`).

a2s TODO

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

8.1 Files modules

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

8.1.1 fetch_module

https://docs.ansible.com/ansible/latest/modules/fetch_module.html#fetch-module

8.1.2 file_module

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

8.1.3 ini_file_module

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

8.1.4 unarchive_module

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

8.2 Other / Ungrouped

8.2.1 httpasswd_module

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

8.2.2 known_hosts_module

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

Maybe these ones:

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

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

9.1 Ansible without plugins

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/>

CHAPTER 10

Indices and tables

- `genindex`
- `search`