
aws-vapor

Release 0.1.0

May 05, 2021

Contents:

1	DSL	1
1.1	Template Root	1
1.2	Elements	1
1.3	Utilities	3
2	Utility Functions	7
	Python Module Index	9
	Index	11

1.1 Template Root

```
class aws_vapor.dsl.Template (version: str = '2010-09-09', description: str = ")
    An AWS CloudFormation template builder.

    conditions (element: aws_vapor.dsl.Element, merge: bool = False) → aws_vapor.dsl.Element
    description (description: str) → aws_vapor.dsl.Template
    mappings (element: aws_vapor.dsl.Element, merge: bool = False) → aws_vapor.dsl.Element
    metadata (element: aws_vapor.dsl.Element, merge: bool = False) → aws_vapor.dsl.Element
    outputs (element: aws_vapor.dsl.Element, merge: bool = False) → aws_vapor.dsl.Element
    parameters (element: aws_vapor.dsl.Element, merge: bool = False) → aws_vapor.dsl.Element
    resources (element: aws_vapor.dsl.Element, merge: bool = False) → aws_vapor.dsl.Element
    to_template () → collections.OrderedDict
```

1.2 Elements

```
class aws_vapor.dsl.Element (name: str)
    This is an abstract base class of a template section.

    attributes (name: str, value: Any)
        Map name to value and return self.

    to_template (template: Dict[str, Any])
        Convert mapped key-value pairs into a top level section of an AWS CloudFormation template.

        Parameters template – A template builder.
        Returns Passed a mapping object.
```

class `aws_vapor.dsl.Metadatum` (*name: str*)

The *Metadatum* class is a subclass of *Element*, each instance of which represents details about the template.

class `aws_vapor.dsl.Parameter` (*name: str*)

The *Parameter* class is a subclass of *Element*, each instance of which passes values into your template when you create a stack.

allowed_pattern (*pattern: str*) → `aws_vapor.dsl.Parameter`

Set 'AllowedPattern' to *pattern* and return *self*.

allowed_values (*list_of_values: List[Any]*) → `aws_vapor.dsl.Parameter`

Set 'AllowedValues' to *list_of_values* and return *self*.

constraint_description (*desc: str*) → `aws_vapor.dsl.Parameter`

Set 'ConstraintDescription' to *desc* and return *self*.

default (*value: Any*) → `aws_vapor.dsl.Parameter`

Set 'Default' to *value* and return *self*.

description (*desc: str*) → `aws_vapor.dsl.Parameter`

Set 'Description' to *desc* and return *self*.

max_length (*length: int*) → `aws_vapor.dsl.Parameter`

Set 'MaxLength' to *length* and return *self*.

max_value (*value: int*) → `aws_vapor.dsl.Parameter`

Set *MaxValue* to *value* and return *self*.

min_length (*length: int*) → `aws_vapor.dsl.Parameter`

Set *MinLength* to *length* and return *self*.

min_value (*value: int*) → `aws_vapor.dsl.Parameter`

Set *MinValue* to *value* and return *self*.

no_echo () → `aws_vapor.dsl.Parameter`

Set 'NoEcho' to *true* and return *self*.

type (*name: str*) → `aws_vapor.dsl.Parameter`

Set 'Type' to *name* and return *self*.

class `aws_vapor.dsl.Mapping` (*name: str*)

The *Mapping* class is a subclass of *Element*, each instance of which matches a key to a corresponding set of named values.

add_category (*category: str*) → `aws_vapor.dsl.Mapping`

Create a new top level section of 'Mappings' and return *self*.

Create a new top level section of 'Mappings' if doesn't contain *category*, then set a current selection to *category*.

Parameters *category* – A name of a top level section.

Returns *self*.

add_item (*key: str, value: Any*) → `aws_vapor.dsl.Mapping`

Map *key* to *value* in a current selection and return *self*.

find_in_map (*top_level_key: str, second_level_key: str*) → `Dict[str, Any]`

Call *Intrinsics.find_in_map* and return its return value.

class `aws_vapor.dsl.Condition` (*name: str*)

The *Condition* class is a subclass of *Element*, each instance of which includes statements that define when a resource is created or when a property is defined.

expression (*expression: Dict[str, Any]*) → *aws_vapor.dsl.Condition*
Set *expression* and return *self*.

to_template (*template: Dict[str, Any]*)
Convert *self.attrs* into a top level section of an AWS CloudFormation template.

Parameters *template* – A template builder.

Returns Passed a mapping object.

class *aws_vapor.dsl.Resource* (*name: str*)

The *Resource* class is a subclass of *Element*, each instance of which declares the AWS resources that you want to include in the stack, such as an Amazon EC2 instance or an Amazon S3 bucket.

add_property (*prop: Dict[str, Any]*) → *aws_vapor.dsl.Resource*
Add a new key-value pair to ‘Properties’ and return *self*.

condition (*condition: aws_vapor.dsl.Condition*) → *aws_vapor.dsl.Resource*
Set ‘Condition’ to *condition* and return *self*.

depends_on (*resource: aws_vapor.dsl.Resource*) → *aws_vapor.dsl.Resource*
Set ‘DependsOn’ to a name of *resource* and return *self*.

metadata (*metadata: Any*) → *aws_vapor.dsl.Resource*
Set ‘Metadata’ to *metadata* and return *self*.

properties (*props: List[Dict[str, Any]]*) → *aws_vapor.dsl.Resource*
Add a list of a new key-value pair to ‘Properties’ and return *self*.

Parameters *props* – A list of a key-value pair.

Returns *self*.

type (*name: str*) → *aws_vapor.dsl.Resource*
Set ‘Type’ to *name* and return *self*.

class *aws_vapor.dsl.Output* (*name: str*)

The *Output* class is a subclass of *Element*, each instance of which declares output values that you can import into other stacks (to create cross-stack references), return in response (to describe stack calls), or view on the AWS CloudFormation console.

condition (*condition: aws_vapor.dsl.Condition*) → *aws_vapor.dsl.Output*

description (*desc: str*) → *aws_vapor.dsl.Output*

export (*name: str*) → *aws_vapor.dsl.Output*

value (*value: Dict[str, Any]*) → *aws_vapor.dsl.Output*

1.3 Utilities

class *aws_vapor.dsl.Attributes*

classmethod *of* (*name: str, value: Any*) → *Dict[str, Any]*

class *aws_vapor.dsl.Intrinsics*

classmethod *base64* (*value_to_encode: Any*) → *Dict[str, Any]*

classmethod *find_in_map* (*map_name_or_mapping: Union[str, Mapping], top_level_key: str, second_level_key: str*) → *Dict[str, Any]*

```
classmethod fn_and (conditions: List[aws_vapor.dsl.Condition] = None) → Dict[str, Any]
classmethod fn_equals (value_1: Any, value_2: Any) → Dict[str, Any]
classmethod fn_if (condition_name: str, value_if_true: Any, value_if_false: Any) → Dict[str, Any]
classmethod fn_not (condition: aws_vapor.dsl.Condition) → Dict[str, Any]
classmethod fn_or (conditions: List[aws_vapor.dsl.Condition] = None) → Dict[str, Any]
classmethod get_att (logical_name_of_resource: str, attribute_name: str) → Dict[str, Any]
classmethod get_azs (region: str = "") → Dict[str, Any]
classmethod import_value (value_to_import: Any) → Dict[str, Any]
classmethod join (delimiter: str, list_of_values: List[Any]) → Dict[str, Any]
classmethod ref (logical_name_or_element: Union[str, Element]) → Dict[str, Any]
classmethod select (index: int, list_of_objects: List[Any]) → Dict[str, Any]
classmethod sub (template: str, dict_of_parameters: Dict[str, Any] = None) → Dict[str, Any]

class aws_vapor.dsl.Pseudos

    classmethod account_id () → Dict[str, Any]
    classmethod no_value () → Dict[str, Any]
    classmethod notification_arns () → Dict[str, Any]
    classmethod region () → Dict[str, Any]
    classmethod stack_id () → Dict[str, Any]
    classmethod stack_name () → Dict[str, Any]

class aws_vapor.dsl.UserData

    classmethod from_files (files: List[Tuple[str, str]], params: Dict[str, Any]) → Dict[str, Any]
    classmethod of (values: List[Any]) → Dict[str, Any]

class aws_vapor.dsl.CfnInitMetadata

    class Authentication (name: str, authentication_type: str)

        access_key_id (value: str) → aws_vapor.dsl.CfnInitMetadata.Authentication
        buckets (list_of_values: List[str]) → aws_vapor.dsl.CfnInitMetadata.Authentication
        password (value: str) → aws_vapor.dsl.CfnInitMetadata.Authentication
        role_name (value: str) → aws_vapor.dsl.CfnInitMetadata.Authentication
        secret_key (value: str) → aws_vapor.dsl.CfnInitMetadata.Authentication
        uris (list_of_values: List[str]) → aws_vapor.dsl.CfnInitMetadata.Authentication
        username (value: str) → aws_vapor.dsl.CfnInitMetadata.Authentication

    class Config (name: str)
```



```

commands (key: str, command: str, env: Dict[str, Any] = None, cwd: str = None, test:
            str = None, ignore_errors: bool = None, wait_after_completion: int = None) →
            aws_vapor.dsl.CfnInitMetadata.Config

files (key: str, content: Union[str, Dict[str, Any]] = None, source: str = None, local_file_path:
         str = None, encoding: str = None, group: str = None, owner: str = None, mode: str =
         None, authentication: str = None, context: str = None, local_file_params: str = None) →
         aws_vapor.dsl.CfnInitMetadata.Config

groups (key: str, gid: int = None) → aws_vapor.dsl.CfnInitMetadata.Config

packages (package_manager: str, key: str, versions: List[str] = None) →
            aws_vapor.dsl.CfnInitMetadata.Config

services (service_manager: str, key: str, ensure_running: bool = None, enabled: bool = None,
            files: List[str] = None, sources: List[str] = None, packages: Dict[str, List[str]] = None,
            commands: List[str] = None) → aws_vapor.dsl.CfnInitMetadata.Config

sources (key: str, url: str) → aws_vapor.dsl.CfnInitMetadata.Config

users (key: str, uid: int, groups: List[str], home_dir: str) →
         aws_vapor.dsl.CfnInitMetadata.Config

class ConfigSet (name, configs: List[CfnInitMetadata.Config])

class Init (config_or_config_sets: List[Union[CfnInitMetadata.Config, CfnInitMeta-
         data.ConfigSet]])

classmethod of (list_of_metadata: List[Union[CfnInitMetadata.Init, CfnInitMeta-
         data.Authentication]]) → collections.OrderedDict

```


CHAPTER 2

Utility Functions

`aws_vapor.utils.combine_user_data` (*files*: *List[Tuple[str, str]]*) → str

Make a multipart/* message from a file content.

Parameters **files** – Paths to a file, a content of which will be used as ‘UserData’.

Returns A multipart/* message attached a file content to.

`aws_vapor.utils.get_property_from_config_file` (*section*: str, *key*: str, *default_value*: str = *None*) → str

Get a property value from a config file.

Parameters

- **section** – A name of a section.
- **key** – A name of a property.
- **default_value** – A value will be returned when a property is not defined.

Returns A property value corresponding to the *key*, which is property name, in the *section*, or *default_value* if the *section* is not defined or the *key* is not defined.

`aws_vapor.utils.inject_params` (*lines*: str, *params*: dict) → List[str]

Replace placeholders with parameters.

Parameters

- **lines** – A file content including placeholders (*{{ ... }}*).
- **params** – A mapping a name of placeholders to a value.

Returns A file content replaced placeholders with parameters.

`aws_vapor.utils.load_from_config_file` (*config_directories*: *List[str]* = *None*) → dict

Load properties from a config file.

Parameters **config_directories** – A path to config directory having ‘config’.
If not specified, locating ‘config’ on *GLOBAL_CONFIG_DIRECTORY* and *LOCAL_CONFIG_DIRECTORY*.

Returns

A mapping of properties loaded from a config file.

example:

```
{
  'section': {
    'key1': 'value1',
    'key2': 'value2'
  }
}
```

`aws_vapor.utils.open_output_file` (*relative_file_path: str*) → `io.TextIOBase`
Open an output file.

Parameters `relative_file_path` – A path to an output file.

Returns A file descriptor of an output file.

`aws_vapor.utils.save_to_config_file` (*props: dict, save_on_global: bool = False*)
Save properties to a config file.

Parameters

- **props** – A mapping of properties.
- **save_on_global** – A flag whether or not a new configuration will be saved globally.

a

`aws_vapor.utils`, [7](#)

A

[access_key_id\(\)](#) (*aws_vapor.dsl.CfnInitMetadata.Authentication* *aws_vapor.dsl.Parameter method*), 2
[account_id\(\)](#) (*aws_vapor.dsl.Pseudos class method*), 4
[add_category\(\)](#) (*aws_vapor.dsl.Mapping method*), 2
[add_item\(\)](#) (*aws_vapor.dsl.Mapping method*), 2
[add_property\(\)](#) (*aws_vapor.dsl.Resource method*), 3
[allowed_pattern\(\)](#) (*aws_vapor.dsl.Parameter method*), 2
[allowed_values\(\)](#) (*aws_vapor.dsl.Parameter method*), 2
[Attributes](#) (*class in aws_vapor.dsl*), 3
[attributes\(\)](#) (*aws_vapor.dsl.Element method*), 1
[aws_vapor.utils](#) (*module*), 7

B

[base64\(\)](#) (*aws_vapor.dsl.Intrinsics class method*), 3
[buckets\(\)](#) (*aws_vapor.dsl.CfnInitMetadata.Authentication method*), 4

C

[CfnInitMetadata](#) (*class in aws_vapor.dsl*), 4
[CfnInitMetadata.Authentication](#) (*class in aws_vapor.dsl*), 4
[CfnInitMetadata.Config](#) (*class in aws_vapor.dsl*), 4
[CfnInitMetadata.ConfigSet](#) (*class in aws_vapor.dsl*), 5
[CfnInitMetadata.Init](#) (*class in aws_vapor.dsl*), 5
[combine_user_data\(\)](#) (*in module aws_vapor.utils*), 7
[commands\(\)](#) (*aws_vapor.dsl.CfnInitMetadata.Config method*), 4
[Condition](#) (*class in aws_vapor.dsl*), 2
[condition\(\)](#) (*aws_vapor.dsl.Output method*), 3
[condition\(\)](#) (*aws_vapor.dsl.Resource method*), 3
[conditions\(\)](#) (*aws_vapor.dsl.Template method*), 1

[constraint_description\(\)](#) (*aws_vapor.dsl.Parameter method*), 2

D

[default\(\)](#) (*aws_vapor.dsl.Parameter method*), 2
[depends_on\(\)](#) (*aws_vapor.dsl.Resource method*), 3
[description\(\)](#) (*aws_vapor.dsl.Output method*), 3
[description\(\)](#) (*aws_vapor.dsl.Parameter method*), 2
[description\(\)](#) (*aws_vapor.dsl.Template method*), 1

E

[Element](#) (*class in aws_vapor.dsl*), 1
[export\(\)](#) (*aws_vapor.dsl.Output method*), 3
[expression\(\)](#) (*aws_vapor.dsl.Condition method*), 2

F

[files\(\)](#) (*aws_vapor.dsl.CfnInitMetadata.Config method*), 5
[find_in_map\(\)](#) (*aws_vapor.dsl.Intrinsics class method*), 3
[find_in_map\(\)](#) (*aws_vapor.dsl.Mapping method*), 2
[fn_and\(\)](#) (*aws_vapor.dsl.Intrinsics class method*), 3
[fn_equals\(\)](#) (*aws_vapor.dsl.Intrinsics class method*), 4
[fn_if\(\)](#) (*aws_vapor.dsl.Intrinsics class method*), 4
[fn_not\(\)](#) (*aws_vapor.dsl.Intrinsics class method*), 4
[fn_or\(\)](#) (*aws_vapor.dsl.Intrinsics class method*), 4
[from_files\(\)](#) (*aws_vapor.dsl.UserData class method*), 4

G

[get_att\(\)](#) (*aws_vapor.dsl.Intrinsics class method*), 4
[get_azs\(\)](#) (*aws_vapor.dsl.Intrinsics class method*), 4
[get_property_from_config_file\(\)](#) (*in module aws_vapor.utils*), 7
[groups\(\)](#) (*aws_vapor.dsl.CfnInitMetadata.Config method*), 5

- I**
- `import_value()` (*aws_vapor.dsl.Intrinsics class method*), 4
 - `inject_params()` (*in module aws_vapor.utils*), 7
 - `Intrinsics` (*class in aws_vapor.dsl*), 3
- J**
- `join()` (*aws_vapor.dsl.Intrinsics class method*), 4
- L**
- `load_from_config_file()` (*in module aws_vapor.utils*), 7
- M**
- `Mapping` (*class in aws_vapor.dsl*), 2
 - `mappings()` (*aws_vapor.dsl.Template method*), 1
 - `max_length()` (*aws_vapor.dsl.Parameter method*), 2
 - `max_value()` (*aws_vapor.dsl.Parameter method*), 2
 - `metadata()` (*aws_vapor.dsl.Resource method*), 3
 - `metadata()` (*aws_vapor.dsl.Template method*), 1
 - `Metadatum` (*class in aws_vapor.dsl*), 1
 - `min_length()` (*aws_vapor.dsl.Parameter method*), 2
 - `min_value()` (*aws_vapor.dsl.Parameter method*), 2
- N**
- `no_echo()` (*aws_vapor.dsl.Parameter method*), 2
 - `no_value()` (*aws_vapor.dsl.Pseudos class method*), 4
 - `notification_arns()` (*aws_vapor.dsl.Pseudos class method*), 4
- O**
- `of()` (*aws_vapor.dsl.Attributes class method*), 3
 - `of()` (*aws_vapor.dsl.CfnInitMetadata class method*), 5
 - `of()` (*aws_vapor.dsl.UserData class method*), 4
 - `open_output_file()` (*in module aws_vapor.utils*), 8
 - `Output` (*class in aws_vapor.dsl*), 3
 - `outputs()` (*aws_vapor.dsl.Template method*), 1
- P**
- `packages()` (*aws_vapor.dsl.CfnInitMetadata.Config method*), 5
 - `Parameter` (*class in aws_vapor.dsl*), 2
 - `parameters()` (*aws_vapor.dsl.Template method*), 1
 - `password()` (*aws_vapor.dsl.CfnInitMetadata.Authentication method*), 4
 - `properties()` (*aws_vapor.dsl.Resource method*), 3
 - `Pseudos` (*class in aws_vapor.dsl*), 4
- R**
- `ref()` (*aws_vapor.dsl.Intrinsics class method*), 4
 - `region()` (*aws_vapor.dsl.Pseudos class method*), 4
 - `Resource` (*class in aws_vapor.dsl*), 3
 - `resources()` (*aws_vapor.dsl.Template method*), 1
- S**
- `role_name()` (*aws_vapor.dsl.CfnInitMetadata.Authentication method*), 4
 - `save_to_config_file()` (*in module aws_vapor.utils*), 8
 - `secret_key()` (*aws_vapor.dsl.CfnInitMetadata.Authentication method*), 4
 - `select()` (*aws_vapor.dsl.Intrinsics class method*), 4
 - `services()` (*aws_vapor.dsl.CfnInitMetadata.Config method*), 5
 - `sources()` (*aws_vapor.dsl.CfnInitMetadata.Config method*), 5
 - `stack_id()` (*aws_vapor.dsl.Pseudos class method*), 4
 - `stack_name()` (*aws_vapor.dsl.Pseudos class method*), 4
 - `sub()` (*aws_vapor.dsl.Intrinsics class method*), 4
- T**
- `Template` (*class in aws_vapor.dsl*), 1
 - `to_template()` (*aws_vapor.dsl.Condition method*), 3
 - `to_template()` (*aws_vapor.dsl.Element method*), 1
 - `to_template()` (*aws_vapor.dsl.Template method*), 1
 - `type()` (*aws_vapor.dsl.Parameter method*), 2
 - `type()` (*aws_vapor.dsl.Resource method*), 3
- U**
- `uris()` (*aws_vapor.dsl.CfnInitMetadata.Authentication method*), 4
 - `UserData` (*class in aws_vapor.dsl*), 4
 - `username()` (*aws_vapor.dsl.CfnInitMetadata.Authentication method*), 4
 - `users()` (*aws_vapor.dsl.CfnInitMetadata.Config method*), 5
- V**
- `value()` (*aws_vapor.dsl.Output method*), 3