Linux Administration with Ansible: Advanced Ansible Automation

Templating Configuration Files with Ansible



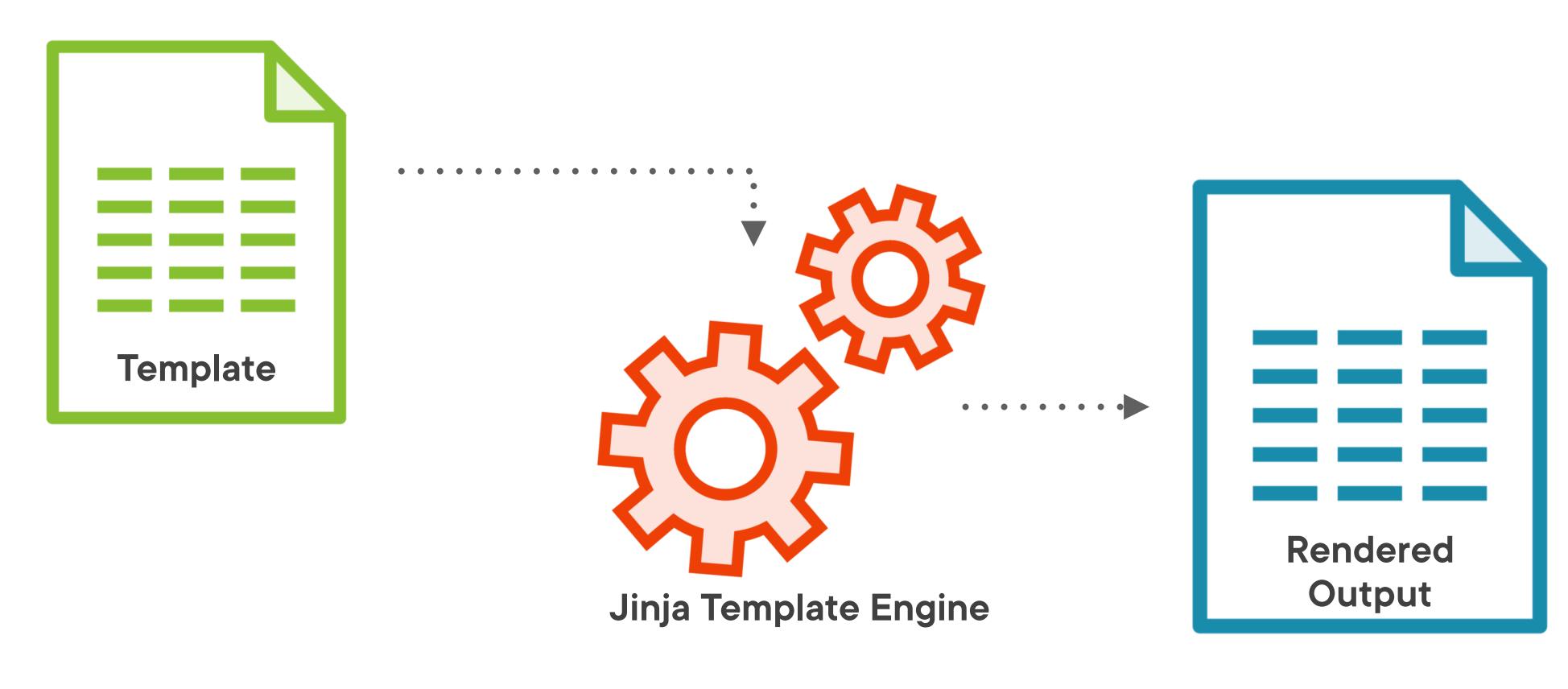
Ugur Oktay
IT Solutions Architect

@uguroktay

Overview

- Review the templating concept
- Variables
- Filters
- If conditional
- For loops
- Globomantics scenario

Templating



Context Data

Jinja Template Engine

Allows dynamic expressions

Allows access to variables

Transparently integrates with Ansible

Completely agnostic to document type

```
# Comments for the configuration file
{# this is a Jinja comment- #}
{% for vhost in apache_vhosts %}
<VirtualHost {{ apache_ip }}:{{ apache_port }}>
  ServerName {{ vhost.servername }}
{% if vhost.alias is defined %}
  ServerAlias {{ vhost.alias }}
{% endif %}
{% if vhost.documentroot is defined %}
  DocumentRoot "{{ vhost.documentroot }}"
{% endif %}
...(truncated)
{% endfor %}
```

Template Example

```
{% ... %} for loops and conditionals
{{ ... }} for variables
{# ... #} for comments not included in the template output
```

Template Module One of the files modules of Ansible

Generates a customized file per each remote host

Templating happens on the Ansible controller

Only the rendered file is copied to remote hosts

Template Task Example

```
- name: template mariadb configuration file
  template:
    src: mariadb_conf.j2
    dest: /etc/my.cnf
    mode: 0664
    owner: root
    group: root
    force: yes
    backup:yes
```

Validation

```
- name: validate new sudoers file with visudo & copy
template:
    src: sudoers
    dest: /etc/sudoers
    validate: /usr/sbin/visudo -cf %s
    backup: yes
```

Render Multiple Templates

```
- name: ensure all template files are rendered
  template:
    src: '{{ item.src }}'
    dest: '{{ item.dest }}'
    loop:
        - { src: 'templates/myapp_cfg.j2', dest: '/home/joe/myapp.cfg' }
        - { src: 'templates/index.html.j2', dest: '/var/www/index.html' }
        - { src: 'templates/config_xml.j2', dest: '/tmp/config.xml' }
```

Demo

Variable Substitution

https://github.com/uguroktay/advanced_ansible_automation

Filters

```
Used for data manipulation
Separated from the variable by a pipe symbol (|):
    {{ customer_name | capitalize }}
           milena
                           Milena
Data kept in the variable is not altered
Multiple variables can be chained:
    {{ customer_name | capitalize | first }}
           milena
                           Milena
                                      M
```

Filters

```
Some filters accept arguments:

{{ list1 | union(list2) }}

Filters help us provide defaults:

{{ character_set | default('utf8') }}
```

https://docs.ansible.com/ansible/latest/user_guide/playbooks_filters.html https://jinja.palletsprojects.com/en/3.0.x/templates/#builtin-filters

Demo

Filters

-Transform a dictionary into a list

```
{% if <expression> %}
     <some text>
{% endif %}
```

Conditional Statements

Used for conditionally including block of texts in a template

If statement is the primary building block

```
{% if tickets %}
We are going to {{ concert_name }}!
{% endif %}
```

Conditional Statements

Can be used with variables

If block can include static text, variables, control structures or a combination

```
{% if tickets %}
    We are going to {{ concert_name }}!
{% elif picnic %}
    Let's get prepared for picnic!
{% else %}
    This will be a boring weekend!
{% endif %}
```

Conditional Statements

"else" and "elif" can be used to further branch out

Expressions

Tests

Comparisons

Logical Operators

List of Builtin Tests

boolean()	even()	in()	mapping()	sequence()
callable()	false()	<pre>integer()</pre>	ne()	string()
<pre>defined()</pre>	filter()	iterable()	none()	test()
divisibleby()	float()	le()	number()	true()
eq()	ge()	lower()	odd()	undefined()
escaped()	gt()	lt()	sameas()	upper()

https://jinja.palletsprojects.com/en/3.0.x/templates/#builtin-tests

Tests

Can be used to test a variable against a common expression.

```
{% if app_port is defined %}
  Port= {{ app_port }}
{% endif %}
```

Tests

syntax:

{% if <var> is <test_name> %}

```
{% if app_port is defined %}
  Port= {{app_port }}

{% elif max_sessions is defined %}
  Port= {{ default_port }}
  Max_sessions= {{ max_sessions }}

{% elif offline is defined %}
  App is offline!

{% else %}
  You need to define variables!

{% endif %}
```

Tests

Block execution stops at the first expression that evaluates to true

```
{% if "basketball" in sports %}
{% if "berry" in "strawberry" %})
```

Tests

Can be used to verify the existence of list variables

Can be used to verify the existence of characters in a string

```
{% if country == "georgia" %}
Capital city is Tblisi!
{% endif %}

{% if version < 3 %}
Time to upgrade!
{% endif %}</pre>
```

Comparisons

Comparison operators: '==', '!=', '>', '>=', '<', '<='

```
{% if logfile and logpath %})

{% if ansible_facts['os_family'] == "RedHat" and ansible_facts['distribution_version'] | int >= 7 %}

{% if not logfile %})
```

Logical Operators

'and', 'or', 'not'

Demo

Version test

```
{% for item in items %}
{{ item }}
{% endfor %}
```

For Loops

Used for iterating over each item in a sequence

For block can include static text, variables, control structures or a combination

```
{# template.j2 file #}
{# Example of looping inside a template #}
{% for writer in writers %}
  {{ writer }}
{% endfor %}
#Playbook file
- hosts: centos1
  vars:
    writers:
      - Tolstoy
      - Dostoyevski
      - London
      - Sabahattin
      - Austen
  tasks:
  - name: Template Loop example
    template:
      src: template.j2
      dest: /tmp/output.txt
```

[vagrant@centos1 ~]\$ more /tmp/output.txt

Tolstoy

Dostoyevski

London

Sabahattin

Austen

Demo

For Loops

Dealing with Large Config Files

Template File

```
{% for parameter in app_variables %}
{{ parameter.name }}={{ parameter.value }}
{% endfor %}
```

Vars File

```
app_variables:
- { name: 'conn_duration', value: '60'}
- { name: 'max_connections', value: '10000'}
- { name: 'segment_size', value: '1000'}
(truncated)
```

Sign-in

Our story

Robotics Media

Support

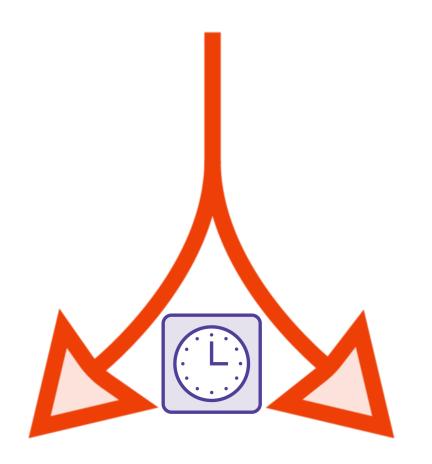
ROBOTICS ALSO A STORY OF LIFE

Life evolves, robotics are no different.











Demo

NTP Template

Ansible Templating

Uses variables and control structures
Integrates with Jinja
Template module runs on the Ansible controller
Jinja template engine allows us to:

- Substitute variables
- Manipulate data
- Use conditional statements
- Use tests, comparisons and logical operators
- Use for loops