

"Unlock Your Potential: Master RHCSA Certification with Our Dynamic eBook!"



🔍 ≡ Menu

How to Create Templates in Ansible to Create Configurations On Managed Nodes – Part 7

James Kiarie | Last Updated: December 10, 2019 | Read Time: 4 mins | [Ansible](#) | [Leave a comment](#)

In this Part 7 of [Ansible Series](#), you will learn how to create and use templates in Ansible to create customized configurations on managed nodes. Templating in Ansible is an easy and friendly way of pushing custom configurations to managed nodes running different systems with minimal editing of playbook files.

What is Templating in Ansible?

To get a better sense of what a template is, let's consider an IT manager drafting an email to invite his department for a cocktail party. The email is sent to each of the members and also invites them to tag along with their spouses.

The email has been customized such that the body of the email remains the same, but the addressees and the names of their respective spouses vary. The email becomes the template, while the recipients and respective spouses are variables.

That was a generic example. Ansible uses Jinja2 which is a modern templating engine for Python frameworks used to generate dynamic content or expressions. Templating is extremely useful when creating custom configuration files for multiple servers but unique for each of them.

Jinja2 uses the double curly braces `{{ ... }}` to enclose a variable that has been defined. For comments, use `{# #}` and for conditional statements use `{% ... %}`.

Let's assume that you have a data model of VLANs in your network with host systems that you want to push to their respective VLANs as shown.

```
vlan:  
  - id: 10  
    name: LB  
  - id: 20  
    name: WB_01  
  - id: 30  
    name: WB_02  
  - id: 40  
    name: DB
```

To render this configuration, the corresponding jinja2 template called `vlan.j2` would appear as shown. As you can see, the variables `vlan.id` and `vlan.name` have been enclosed in curly braces.

```
vlan {{ vlan.id }}  
  name {{ vlan.name }}
```

Putting it all together in a playbook which places different host machines, this would appear as shown:

```
- hosts  
tasks:  
  - name: Rendering VLAN configuration  
    template:  
      src: vlan.j2  
      dest: "vlan_configs/{{ inventory_hostname }}.conf"
```

Example 1: Configuring Web Servers in Different Distro

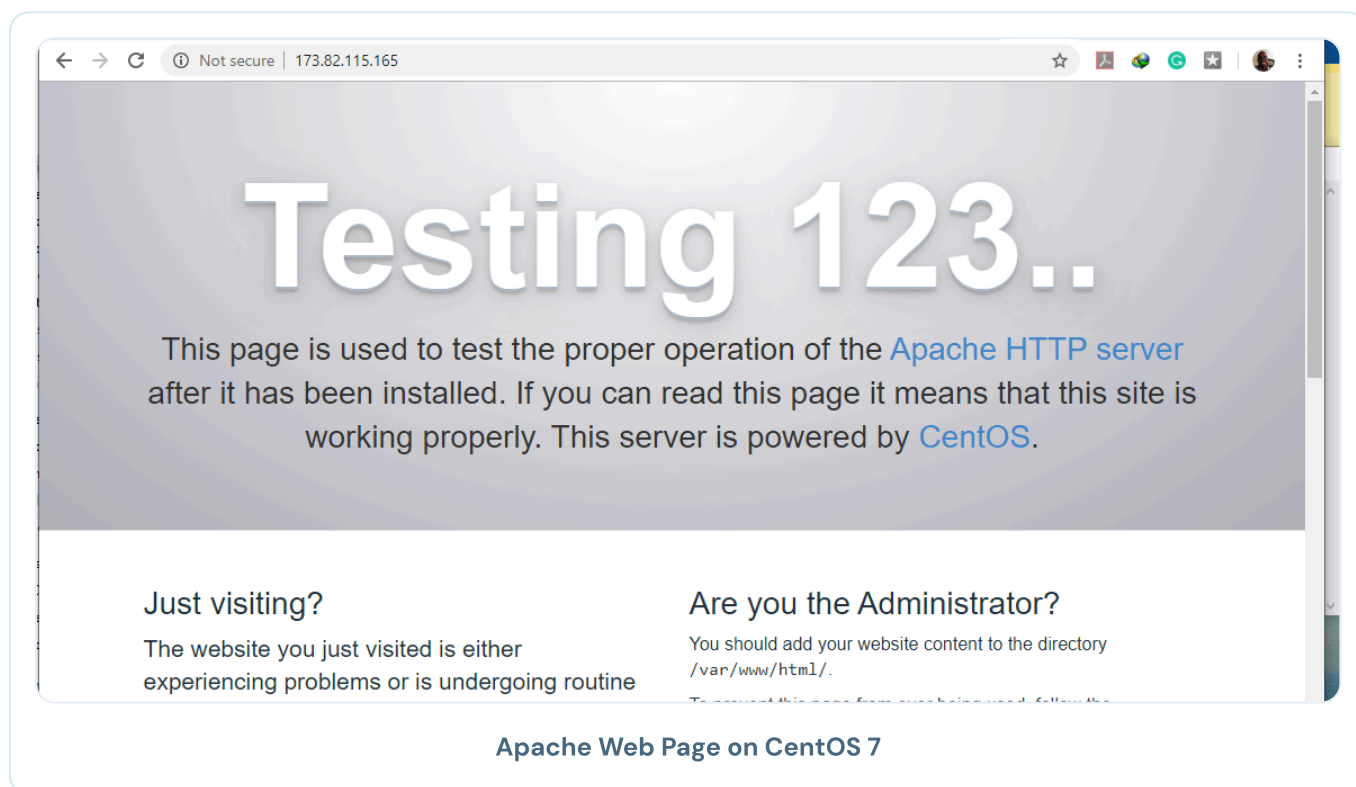
In this example, we shall create `index.html` files that will display information about the hostname & the OS of 2 web servers running CentOS & Ubuntu.

Setup

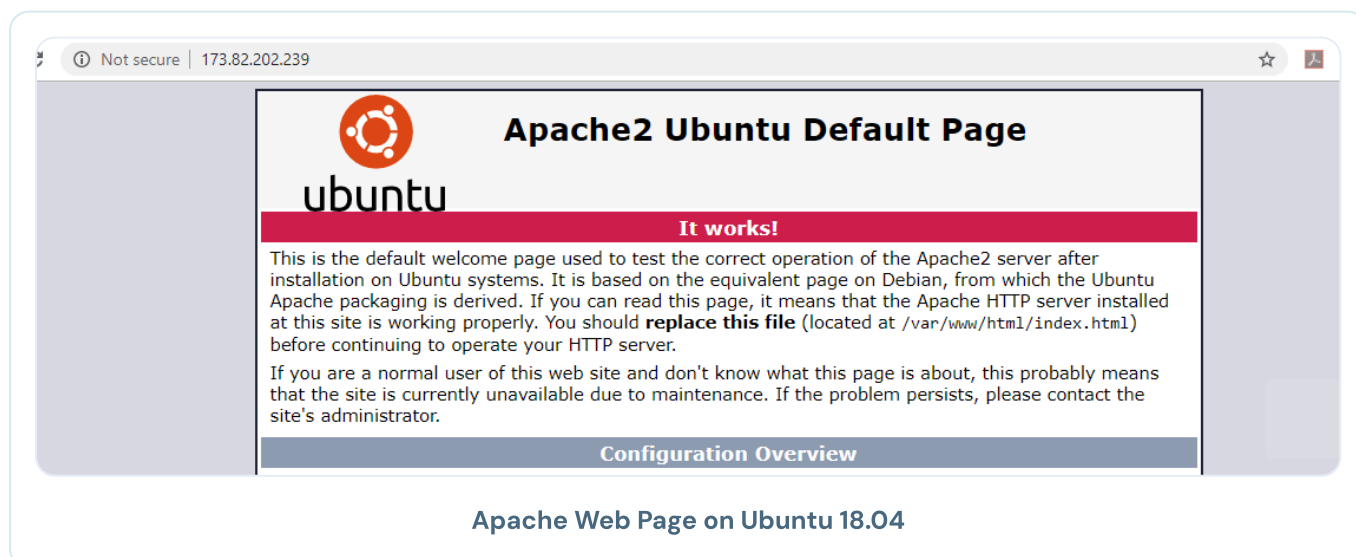
Ubuntu 18 - IP address: 173.82.202.239
CentOS 7 - IP address: 173.82.115.165

Apache webserver has already been installed on both servers.

For CentOS 7



For Ubuntu 18.04



So let's create a playbook `test_server.yml` as shown:

```
---

- hosts: all
  become: yes

  tasks:

    - name: Install index.html
      template:
        src: index.html.j2
        dest: /var/www/html/index.html
        mode: 0777
```

Our Jinja file template is `index.html.j2` which will be pushed to the `index.html` file on each webserver. Always remember to put the extension `.j2` at the end to signify that it's a jinja2 file.

Let's now create the template file `index.html.j2`.

```
<html>
<center>
  <h1> The hostname of this webserver is {{ ansible_hostname }}</h1>
  <h3> It is running on {{ ansible_os_family }}system </h3>
</center>
</html>
```

This template is a basic HTML file where the `ansible_hostname` and `ansible_os_family` are built-in variables that will be substituted with the respective hostnames and operating systems of the individual webserver on the browser.

Now, Let's run the playbook.

```
# ansible-playbook test_server.yml
```

```
[root@rhel-8 ansible]#
[root@rhel-8 ansible]# ansible-playbook test_server.yml

PLAY [all] *****

TASK [Gathering Facts] *****
ok: [173.82.115.165]
ok: [173.82.202.239]

TASK [Install index.html] *****
changed: [173.82.202.239]
changed: [173.82.115.165]

PLAY RECAP *****
173.82.115.165      : ok=2    changed=1    unreachable=0    failed=0    skipped=0    rescued=0
  ignored=0
173.82.202.239     : ok=2    changed=1    unreachable=0    failed=0    skipped=0    rescued=0
  ignored=0
[root@rhel-8 ansible]#
```

Create Ansible Template File

Now let's reload the webpages for both CentOS 7 and Ubuntu webservers.

For CentOS 7

Not secure | 173.82.115.165

The hostname of this webserver is centos-7

It is running on RedHat operating system

Verify Apache Web Page on CentOS 7

For Ubuntu 18.04

Not secure | 173.82.202.239

The hostname of this webserver is ubuntu

It is running on Debian operating system

Verify Apache Web Page on Ubuntu 18.04

As you can see, different information about the hostname and family of OS has been displayed on each server. And that's how cool Jinja2 templating is!

FILTERS:

Sometimes, you may decide to substitute the value of a variable with a string that appears in a certain manner.

Example 1: Make Strings Appear in Uppercase/lowercase

For instance, in the previous example, we can decide to make the Ansible variables appear in Uppercase. To do so, append the value upper to the variable. This way the value in the variable is converted to Uppercase format.

```
{{ ansible_hostname | upper }} => CENTOS 7  
{{ ansible_os_family | upper }} => REDHAT
```



Similarly, you can convert the string output to lowercase by appending the lower argument.

```
{{ ansible_hostname | lower }} => centos 7  
{{ ansible_os_family | lower }} => redhat
```



Example 2: Replace a String with Another

Additionally, you can replace a string with another.

For example:

The movie title is `{{ movie_name }}` => The movie title is Ring.

To replace the output with another string, use the replace argument as shown:

The movie title is `{{ movie_name | replace ("Ring","Heist") }}` => The movie title is Heist.

Example 3: Lists and Sets Filters

To retrieve the smallest value in an array, use the min filter.

```
{{ [ 2, 3, 4, 5, 6, 7 ] | min }}      =>      2
```

Similarly, to retrieve the largest number, use the max filter.

```
{{ [ 2, 3, 4, 5, 6, 7 ] | max }}      =>      7
```

To display unique values, use the unique filter.

```
{{ [ 2, 3, 3, 2, 6, 7 ] | unique }} =>  2, 3
```

Use the random filter to obtain a random number between 0 and the value.

```
{{ 50 | random }} =>  Some random number
```

LOOPS:

Just like in programming languages, we have loops in Ansible Jinja2.

For example, to generate a file containing a list of numbers use the for loop as shown in the example below:

Example 1:

```
{% for number in [0, 1, 2, 3, 4, 5, 6, 7] %}  
  {{ number }}  
{% endfor %}
```

You can also combine the for loop with if-else statements to filter and obtain certain values.

Example 2:

```
{% for number in [0, 1, 2, 3, 4, 5, 6, 7] %}  
  {% if number == 5 %}  
    {{ number }}  
  {% endif %}  
{% endfor %}
```

And that's it for this lecture. Join us in the next topic where we will venture into working with ansible variables and facts.

🔗 [Ansible Exam Guide](#), [Ansible Tips](#)

Hey TecMint readers,

Exciting news! Every month, our top blog commenters will have the chance to win fantastic rewards, like free Linux eBooks such as RHCE, RHCSA, LFCS, Learn Linux, and Awk, each worth \$20!

Learn [more about the contest](#) and stand a chance to win by [sharing your thoughts below!](#)

GIVEAWAY!

Win eBooks



www.tecmint.com

PREVIOUS ARTICLE:

[How to Install Nginx on CentOS 8](#)

NEXT ARTICLE:

[How to Set A Custom Screen Resolution in Ubuntu Desktop](#)



James Kiarie

This is James, a certified Linux administrator and a tech enthusiast who loves keeping in touch with emerging trends in the tech world. When I'm not running commands on the terminal, I'm taking listening to some cool music. taking a casual stroll or watching a nice movie.

Each tutorial at TecMint is created by a team of experienced Linux system administrators so that it meets our high-quality standards.

Join the [TecMint Weekly Newsletter](#) (More Than 156,129 Linux Enthusiasts Have Subscribed)

Was this article helpful? Please [add a comment](#) or [buy me a coffee](#) to show your appreciation.

Related Posts

```
aaronk@tecmint:~$ ansible prod_servers -a "systemctl status firewalld" -u root
[...].235 | FAILED! => {
  "changed": false,
  "module_stderr": "Shared connection to [...].235 closed.\r\n",
  "module_stdout": "/bin/sh: /usr/bin/python: No such file or directory\r\n",
  "msg": "MODULE FAILURE",
  "rc": 127
}
[...].80 | FAILED! => {
  "changed": false,
  "module_stderr": "Shared connection to [...]80 closed.\r\n",
  "module_stdout": "/bin/sh: /usr/bin/python: No such file or directory\r\n",
  "msg": "MODULE FAILURE",
  "rc": 127
}
aaronk@tecmint:~$
```

How to Fix "Shared connection to x.x.xx closed" Ansible Error



RedHat Certified Specialist in Ansible Automation Study Guide

eBook by Tecmint.com

Tecmint's Guide to RedHat Ansible Automation Exam Preparation Guide



RED HAT[®] **ANSIBLE[®]** **Certification Guide**

How to Use Ansible Vault in Playbooks to Protect Sensitive Data - Part 10

How to Use Ansible Vault in Playbooks to Protect Sensitive Data – Part 10



RED HAT[®] **ANSIBLE[®]** **Certification Guide**

How to Create and Download Roles on Ansible Galaxy and Use Them - Part 9

How to Create and Download Roles on Ansible Galaxy and Use Them – Part 9



RED HAT[®] ANSIBLE[®] Certification Guide

How to Work with Ansible Variables and Facts – Part 8

How to Work with Ansible Variables and Facts – Part 8



RED HAT[®] ANSIBLE[®] Certification Guide

How to Use Ansible Modules for System Administration Tasks – Part 6

How to Use Ansible Modules for System Administration Tasks – Part 6

Got Something to Say? Join the Discussion...

Thank you for taking the time to share your thoughts with us. We appreciate your decision to leave a comment and value your contribution to the discussion. It's important to note that we moderate all comments in accordance with our [comment policy](#) to ensure a respectful and constructive conversation.

Rest assured that your email address will remain private and will not be published or shared with anyone. We prioritize the privacy and security of our users.

☐ Save my name, email, and website in this browser for the next time I comment.

Post Comment

Do You Enjoy My Blog?

Support from readers like YOU keeps this blog running. Buying me a cup of coffee is a simple and affordable way to show your appreciation and help keep the posts coming!

Buy Me a Coffee

Linux Commands and Tools

[Fping – A High Performance Ping Tool for Linux](#)

[5 Ways to Empty or Delete a Large File Content in Linux](#)

[8 Parted Commands to Manage Disk Partitions in Linux](#)

[8 Useful X-window \(Gui Based\) Linux Commands – Part I](#)

[PacVim – A Game That Teaches You Vim Commands](#)

[Set Date and Time for Each Command You Execute in Bash History](#)

Linux Server Monitoring Tools

[Linfo – Shows Linux Server Health Status in Real-Time](#)

[linux-dash: Monitors “Linux Server Performance” Remotely Using Web Browser](#)

[Sysstat – All-in-One System Performance and Usage Activity Monitoring Tool For Linux](#)

[How to Install Icinga2 Monitoring Tool on OpenSUSE](#)

[Understand Linux Load Averages and Monitor Performance of Linux](#)

[How to Use ‘fsck’ to Repair Linux File System Errors](#)

Learn Linux Tricks & Tips

[Learn Difference Between “su” and “su -” Commands in Linux](#)

[How to Create a Virtual HardDisk Volume Using a File in Linux](#)

[How to Download and Extract Tar Files with One Command](#)

[Bash-it – Bash Framework to Control Your Scripts and Aliases](#)

[How to Run or Repeat a Linux Command Every X Seconds Forever](#)

How to Disable Shutdown and Reboot Commands in Linux

Best Linux Tools

Top 6 Partition Managers (CLI + GUI) for Linux

14 Best Tiling Window Managers for Linux in 2024

11 Best Free and Low-Cost SSL Certificate Authorities

9 Best Microsoft Excel Alternatives for Linux

10 Best Flowchart and Diagramming Software for Linux

6 Best Whiteboard Applications for Your Linux Systems

Tecmint: Linux Howtos, Tutorials & Guides © 2024. All Rights Reserved.

The material in this site cannot be republished either online or offline, without our permission.

Hosting Sponsored by : [**Linode Cloud Hosting**](#)