The Basics of Ansible Variables

Posted on September 30, 2017 (http://www.mydailytutorials.com/the-basics-of-ansible-variables/) by Ansible admin (http://www.mydailytutorials.com/author/ansible-admin/)

(http://www.mydailytutorials.com/the-basics-of-ansible-variables/)

Variables are used to store values in programs and as the name suggests the values can be changed throughout the program. These values are essential to deciding the code flow. Ansible variables help to determine how the tasks execute on different systems based on the values assigned to these variables.

Ansible variable names should be letters, numbers, underscores and they should always start with a letter.

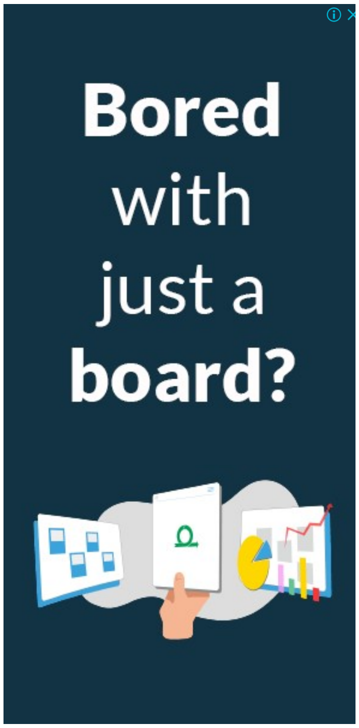
Basic Ansible Variable example

In the playbook, we can give a variable in '**variable_name: variable_value**' format. And we can use the variable_name inside double braces anywhere in the playbook. In the following task, I am declaring a variable **hello** with value **world** and is referenced in one of the tasks. The task will output the value **world**.

```
- hosts: all
  vars:
    hello: world
  tasks:
  - name: Ansible Basic Variable Example
    debug:
      msg: "{{ hello }}"
```

1.2 Ansible List (Array) variables

We can also have an array or list variables. The following task shows how to declare an array variable in Ansible and how to use the values. The **hello** contains 9 values, and each one can be accessed using the index numbers(starting from zer0). The following task will output '**South America**'



Search... SEARCH

RECENT POSTS

How to create files in Ansible (http://www.mydailytutorials.com/ansible-create-files/)

Replacing Strings and Lines in Ansible (http://www.mydailytutorials.com/ansible-replace-example/)

Working with Ansible yum module (http://www.mydailytutorials.com/ansible-yum-module-examples/)

Ansible Pip Module – Managing Python Packages (http://www.mydailytutorials.com/ansible-pip-module/)

Working with Environment Variables in Ansible (http://www.mydailytutorials.com/working-with-environment%e2%80%8b-variables-in-ansible/)

```
- hosts: all
  vars:
    hello:
      - World
      - Asia
      - South America
      - North America
      - Artic
      - Antartic
      - Oceania
      - Europe
      - Africa
  tasks:
  - name: Ansible List variable Example
    debug:
      msg: "{{ hello[2] }}"
```

You can also give the variable values in the following format.

```
vars:
  hello: [Asia, Americas, Artic, Antartic ,Oceania,Europe,Africa]
```

If you need to access all the variables, you can use the **with_items** structure to loop through all the values. For more details n how to use the loop structure for this tutorial (http://www.mydailytutorials.com/working-with-ansible-loop/).

```
- hosts: all
  vars:
    hello: [Asia, Americas, Artic, Antartic ,Oceania,Europe,Africa]
  tasks:
  - name: Ansible array variables example
    debug:
      msg: "{{ item }}"
    with_items:
      - "{{ hello }}"
```

1.2. Ansible Dictionary Variables (Hash)

You can declare dictionaries in Ansible as you would in a python script. And you can access any of the value using the **key.** In the following task, I am declaring a variable python which has 3 key-value pairs inside. The following task would print the whole dictionary structure.

“msg”: {
“Designer”: “Guido van Rossum”,
“Developer”: “Python Software Foundation,”
“OS”: “Cross-platform”
}

```
- hosts: all
  vars:
    python:
      Designer: 'Guido van Rosum'
      Developer: 'Python Software Foundation'
      OS: 'Cross-platform'
  tasks:
  - name: Ansible Dictionary Example
    debug:
      msg: "{{ python }}"
```

CATEGORIES

Ansible (http://www.mydailytutorials.com/category/tutorials/ansible/)

Linux (http://www.mydailytutorials.com/category/tutorials/linux/)

Tutorials (http://www.mydailytutorials.com/category/tutorials/)

You can access the individual values in the dictionary variable using the key name. There are two ways for this, using the dot notation or the bracket notation.

Bracket notation – variable_name[‘key’]

Dot notation – variable_name.key

For example, if you need the designer’s name hen you can do like below. Both will print the name of the designer.

```
  tasks:
  - name: Ansible Hash Example
    debug:
      msg: "{{python['Designer'] }}"
  - name: Ansible Find Example
    debug:
      msg: "{{python.Designer }}"
```

You can also give the dictionary variables in the following format.

```
  vars:
    python:{ Designer: 'Guido van Rossum', Developer: 'Python Software Foundation
',OS: 'Cross-platform'}
```

You can also loop through all the values inside the dictionary using the **with_dict** parameter. The dictionary variable has to be given to the with_dict parameter. The key and value will be available from **item.key** and **item.value** respectively.

```
- hosts: all
  vars:
    python:
      Designer: 'Guido van Rossum'
      Developer: 'Python Software Foundation'
      OS: 'Cross-platform'
  tasks:
  - name: Ansible Dictionary variable Example
    debug:
      msg: "Key is {{ item.key}} and value is {{item.value}}"
    with_dict: "{{ python }}"
```

Issue: For some reason, I was getting the following error when I tried using the with_dict with python variable on the next line.

fatal: [localhost]: FAILED! => {“failed”: true, “msg”: “with_dict expects a dict”}

1.3 Ansible Multiline Variables

It is possible to give variable values in multiple lines using YAML block operators. There are two options for this, depending on whether you need the newline characters to be retained or not.

You can use the ‘|’ operator if you want the new line characters to be included. If you do not want the new line character, and you just want to use it for easy understanding during editing then use ‘>’ operator.

```
- hosts: all
  vars:
    include_newlines_example: |
            The new line charaters
            will appear

    ignore_newlines_example2: >
            The new line character will
            be removed. Useful when editing
            lines
  tasks:
  - name: Ansible varible multiple line Example
    debug:
      msg: "{{ include_newlines_example }}"

  - name: Ansible variables multiline Example
    debug:
      msg: "{{ ignore_newlines_example2 }}"
```

ANSIBLE VARIABLES (HTTP://WWW.MYDAILYTUTORIALS.COM/TAG/ANSIBLE-VARIABLES/)

LONG (HTTP://WWW.MYDAILYTUTORIALS.COM/TAG/LONG/)

‹ USING GREP COMMANDS IN ANSIBLE TASKS
(HTTP://WWW.MYDAILYTUTORIALS.COM/USING-GREP-COMMANDS-ANSIBLE-TASKS/)

WORKING WITH ANSIBLE REGISTER VARIABLES ›
(HTTP://WWW.MYDAILYTUTORIALS.COM/ANSIBLE-REGISTER-VARIABLES/)

Activello Theme by Colorlib (http://colorlib.com/) Powered by WordPress (http://wordpress.org/)