

Working with Environment Variables in Ansible

Posted on November 28, 2017 (<http://www.mydailytutorials.com/working-with-environment%E2%80%8Bvariables-in-ansible/>) by Ansible admin (<http://www.mydailytutorials.com/author/ansible-admin/>)

(<http://www.mydailytutorials.com/working-with-environment%E2%80%8Bvariables-in-ansible/>)

Environment Variables in the Local Machine/Ansible Control Machine

First, let us see how to work with the environment variables on the server which acts as the ACM.

Accessing the environment variables

You can access the environment variables available on the local server via the 'lookup' plugins, which allow you to access the system data, and 'env' (<https://docs.ansible.com/ansible/devel/plugins/lookup/env.html>) plugin which is written for accessing the environment variables.

Suppose we need to print the value of the 'HOME' environment variable. We can execute the following task.

```
- hosts: all
  tasks:
    - name: Printing the environment variable in Ansible
      debug:
        msg: "{{ lookup('env','HOME') }}"

output
-----
"msg": "/Users/mdtutorials2"
```

Remember that this task is printing the local servers HOME environment variables. So even if you run it against multiple hosts, it will print the same output.

You can also give multiple environment variables with the same command. If I want to print the home, user and shell variables, then I can do the following.

```
- hosts: all
  gather_facts: False
  tasks:
    - name: Multiple environment variables in a single line ansible task
      debug:
        msg: "{{ lookup('env','USER','HOME','SHELL') }}"

output
-----
"msg": "root,/Users/mdtutorials2,/bin/sh"
```

Storing the value to another variable

You can directly store the value got by lookup plugin in to a variable.

For example, I want to store the output the of the SHELL environment variable in Ansible.

In the following task, the value is saved in to a variable called 'local_shell'.

```
- hosts: all
  vars:
    local_shell: "{{ lookup('env', 'SHELL') }}"
  tasks:
    - debug:
        msg: "{{ local_shell }}"

output
-----
"msg": "/bin/sh"
```

Now you can use this variable along with any conditional statement also as usual. Suppose you might want to run some commands on the ocal server depending on the shell used.

Environment Variables in the Remote Servers/ Target Machines

Since the lookup plugins work only on the local machines, we need some other way to access the variables on the remote servers.

Accessing the variables

The environment variables of the remote servers can be accessed via 'facts' (<http://www.mydailytutorials.com/working-with-ansible-facts-retrieving-facts/>). There is a sublist called 'ansible_env' which has all the env variables inside it.

Note that, the fact-gathering needs to be 'ON' if you want to access this. By default, it is set to true. Or you can set the 'gather_facts' to true explicitly.

Suppose I want to access the 'HOME' variable on the remote servers and print it, I can do the following.

```
- hosts: all
gather_facts: true
tasks:
  - name: Remote server ansible variables
    debug:
      msg: "{{ ansible_env.HOME }}"

output
-----
"msg": "/Users/mdtutorials3"
```

I can also print or dump all the environment variables by not giving any variables name.

```
- hosts: all
gather_facts: true
tasks:
  - name: Printing all the environment variables in Ansible
    debug:
      msg: "{{ ansible_env }}"

output
-----
"msg": {
  "HOME": "/Users/mdtutorials3",
  "LANG": "en_US.UTF-8",
  "LOGNAME": "root",
  "MAIL": "/var/mail/root",
  "PWD": "/Users/mdtutorials3",
  "SHELL": "/bin/sh",
}
```

Comments (2) (<http://www.mydailytutorials.com/working-with-environment%E2%80%8Bvariables-in-ansible/#comments>)

TUTORIAL ([HTTP://WWW.MYDAILYTUTORIALS.COM/TAG/TUTORIAL/](http://www.mydailytutorials.com/tag/tutorial/))

Monitor HAProxy
performance and availability.

The advertisement features a dark blue background with a large, stylized red letter 'A' on the right side. The text is in a white, sans-serif font.