



## How to Use Static and Dynamic Inventories in Ansible – Part 4

James Kiarie | Last Updated: March 26, 2020 | Read Time: 4 mins | [Ansible](#) | [8 Comments](#)

In this Part 4 of [Ansible Series](#), we will explain how to use static and dynamic inventory to define groups of hosts in Ansible.

In an Ansible, managed hosts or servers which are controlled by the Ansible control node are defined in a host inventory file as explained in. A host inventory file is a text file that consists of hostnames or IP addresses of managed hosts or remote servers.

Managed hosts can either be listed as individual entries or categorized under a group name as we shall later see. In Ansible, there are two types of inventory files: Static and Dynamic.

Let's have a look at each one of these and see how we can manage them. By now, we assume that you have already installed [Ansible on your Control node](#), and configured Passwordless SSH connection to your managed hosts.

### Static Host Inventory File

In Ansible, a static inventory file is a plain text file that contains a list of managed hosts declared under a host group using either hostnames or IP addresses.

A host group name is enclosed in square brackets i.e `[group name]`. The managed host entries are later listed below the group name, each on its own line. As discussed earlier, the hosts are listed using either hostnames or IP addresses.

```
[group name]
```

```
Host A ip_address  
Host B ip_address  
Host c ip_address
```

For purposes of illustration, we shall create a static inventory file.

```
# mkdir test_lab && cd test_lab  
# vim hosts
```

### Static Inventory File

```
[webservers]  
173.82.115.165  
  
[database_servers]  
173.82.220.239  
  
[datacenter:children]  
webservers  
database_servers
```

Save the file and exit.

As you can see in the inventory file above, we have created 2 host groups: **webservers** and **database\_servers**. Also, we have created an additional group called **datacenter** that includes a group of host groups denoted by **': children'** suffix as seen above.

Ansible also allows groups of hosts to be placed under a group name. In the inventory file above, the **webservers** and **database\_servers** groups have been placed under the **datacenter**.

**NOTE:** It's not mandatory to place managed hosts in a host group. You can simply list them using their hostnames or IP addresses for example.

```
173.82.202.239  
172.82.115.165
```

```
load_balancer.pnl.com
```

Let's now use a few Ansible commands for referencing the host inventory file. The basic syntax for inventory management is as shown.

```
$ ansible {host-pattern} -i /path/of/inventory/file --list-hosts
```

For example,

```
$ ansible all -i /root/test_labs/hosts --list-hosts
```

```
[root@rhel-8 ~]#  
[root@rhel-8 ~]# ansible all -i /root/test_lab/hosts --list-hosts  
hosts (2):  
    173.82.115.165  
    173.82.220.239  
[root@rhel-8 ~]#
```

List Ansible Hosts

Alternatively, you can use the wildcard character `*` to replace `'all'` argument.

```
$ ansible * -i /root/test_labs/hosts --list-hosts
```

```
[root@rhel-8 ~]#  
[root@rhel-8 ~]# ansible '*' -i /root/test_lab/hosts --list-hosts  
hosts (2):  
    173.82.115.165  
    173.82.220.239  
[root@rhel-8 ~]#
```

List Ansible Hosts Using Wildcards

To list hosts in a group, specify the host group in the place of host-pattern.

```
$ ansible webserver -i /root/test_labs/hosts --list-hosts
```

```
[root@rhel-8 ~]# ansible webservers -i /root/test_lab/hosts --list-hosts
hosts (1):
  173.82.115.165
[root@rhel-8 ~]#
```

#### List Ansible Hosts in Group

## Dynamic Host Inventory File

In a configuration – especially a cloud setup such as AWS where the inventory file keeps constantly changing as you add or decommission servers, keeping tabs on the hosts defined in the inventory file becomes a real challenge. It becomes inconvenient going back to the host file and updating the list of hosts with their IP addresses.

And this is where a dynamic inventory comes to play. So what is a dynamic inventory? A dynamic inventory is a script written in Python, PHP or any other programming language. It comes in handy in cloud environments such as AWS where IP addresses change once a virtual server is stopped and started again.

Ansible already has developed inventory scripts for public cloud platforms such as Google Compute Engine, Amazon EC2 instance, OpenStack, RackSpace, cobbler, among others.

What are the advantages of a dynamic inventory over a static inventory?

- Dynamic inventories do a perfect job of reducing human error as information is gathered using scripts.
- Minimal effort is required in managing inventories.

You can write your own customize dynamic inventory in a programming language of your choice. The inventory should return a format in JSON when appropriate options are passed.

## Utilize an Existing Dynamic Inventory Script

A script that is used to create a dynamic inventory has to be made executable so that Ansible can use it.

To retrieve information about the hosts inside a dynamic inventory script simply run.

```
# ./script --list
```

As pointed earlier, the output should be in JSON in the format below.

A dictionary comprising of groups (i.e webservers, database\_Servers)

- A list of managed hosts per group
- A dictionary of variables

Meta dictionary

- Hosts and hostvars

## Sample Output

---

```
{
  "webservers": {
    "hosts": [
      "webserver1.example.com",
      "webserver2.example.com"
    ],
    "vars": {}
  },
  "database_servers": {
    "hosts": [
      "mysql_db1",
      "mysql_db2"
    ],
    "vars": {}
  },
  "_meta": {
    "hostvars": {
      "mysql_db2": {},
      "webserver2.example.com": {},
      "webserver1.example.com": {},
      "mysql_db1": {}
    }
  }
}
```

```
}  
}
```

## Conclusion

In this article, we have demonstrated how to create both static and dynamic inventories. In summary, a static inventory file is a plain text file containing a list of managed hosts or remote nodes whose numbers and IP addresses remain fairly constant.

On the other hand, a dynamic host file keeps changing as you add new hosts or decommission old ones. The IP addresses of hosts are also dynamic as you stop and start new host systems. We do hope that you found this tutorial informative.

🔖 [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](http://www.tecmint.com)

PREVIOUS ARTICLE:

**[How to Setup a DNS/DHCP Server Using dnsmasq on CentOS/RHEL 8/7](#)**

NEXT ARTICLE:

## **fwbackups – A Feature-rich Backup Program for Linux**



**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<sup>®</sup>** **ANSIBLE<sup>®</sup>** **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<sup>®</sup>** **ANSIBLE<sup>®</sup>** **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<sup>®</sup>** **ANSIBLE<sup>®</sup>** **Certification Guide**

## **How to Work with Ansible Variables and Facts - Part 8**

### **How to Work with Ansible Variables and Facts – Part 8**



# **RED HAT<sup>®</sup>** **ANSIBLE<sup>®</sup>** **Certification Guide**

## **How to Create and Use Templates to Create Customized Configuration Files - Part 7**

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

 **8 Comments**

**Leave a Reply**

**derek**

March 10, 2021 at 1:25 am

I love tecmint and this is another worthwhile link with an explanation of network inventory for ansible using JSON (i.e. an ex python step).

Please change "hostvars" to "vars" as my version of ansible issues a warning.

```
[derek@socorro JinaTemplates]$ ansible -i network.json --list-hosts all
[WARNING]: Skipping unexpected key (hostvars) in group (_meta), only "vars",
"children" and "hosts" are valid
hosts (8):
eng-pe2
eng-pe1
spn-2
```

[Reply](#)**Amedix**

March 26, 2020 at 11:54 am

Hi,

Good Ansible articles!

May I suggest a syntax correction?

In the "Dynamic Host Inventory File" section, in the fourth line, change :

A dynamic inventory is a shell script written in Python, PHP or any other programming language.

by

A dynamic inventory is a script written in Python, PHP or any other programming language.

Thanks

[Reply](#)

Admin



**Ravi Saive**

March 26, 2020 at 12:00 pm

@Amedix,

Thanks for the suggestion, corrected in the article...

[Reply](#)

**Matt**

March 6, 2020 at 2:07 pm

Hi James,

After a dynamic inventory script successfully (using `./somescript -list`) how do you use the JSON response as your hosts to execute a playbook?

[Reply](#)

**QA**

December 7, 2019 at 1:31 pm

Yes, but I want to know the content of your Dynamic Inventory Script file, I searched a lot and I'm just a low-level person. I want JSON results like you, please let me know.

I would be happy to be able to get your contact

[Reply](#)



**James Kiarie**

December 5, 2019 at 2:40 pm

Hey QA, The command simply displays the host systems in the script. The `--list` argument is used to display the hosts.

[Reply](#)

**QA**

December 6, 2019 at 1:31 pm

Yes, but I want to know the content of your Dynamic Inventory Script file, I searched a lot and I'm just a low-level person. I want JSON results like you, please let me know.

I would be happy to be able to get your contact

[Reply](#)

**QA**

December 5, 2019 at 9:05 am

what is `./script --list` ? Please let me know..

[Reply](#)

## 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.*



Name \*

Email \*

☐ 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

[5 Command Line Ways to Find Out Linux System is 32-bit or 64-bit](#)

[27 'DNF' \(Fork of Yum\) Commands for RPM Package Management in Linux](#)

[How to Monitor Progress of \(Copy/Backup/Compress\) Data using 'pv' Command](#)

[How to Convert From RPM to DEB and DEB to RPM Package Using Alien](#)

[How to Show Asterisks While Typing Sudo Password in Linux](#)

[35 Practical Examples of Linux Find Command](#)

## Linux Server Monitoring Tools

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

[GoAccess \(A Real-Time Apache and Nginx\) Web Server Log Analyzer](#)

[How to Install Zabbix on RHEL/CentOS and Debian/Ubuntu – Part 1](#)

[How to Install Nagios XI on Ubuntu 22.04](#)



## **Perf- A Performance Monitoring and Analysis Tool for Linux**

## **Icinga: A Next Generation Open Source 'Linux Server Monitoring' Tool for RHEL/CentOS 7.0**

## **Learn Linux Tricks & Tips**

### **How to Count Word Occurrences in a Text File**

### **Understanding Shutdown, Poweroff, Halt and Reboot Commands in Linux**

### **How to Search and Remove Directories Recursively on Linux**

### **How to Copy a File to Multiple Directories in Linux**

### **5 Useful Ways to Do Arithmetic in Linux Terminal**

### **How to Show Asterisks While Typing Sudo Password in Linux**

## **Best Linux Tools**

### **14 Best Tiling Window Managers for Linux in 2024**

### **11 Best Screen Recorders For Linux in 2024**

### **10 Tools to Monitor Linux Disk Partitions and Usage in Linux**

### **10 Top Open Source Artificial Intelligence Tools for Linux**

### **The 27 Best IDEs and Code Editors for Linux**

### **Top 5 Open-Source Enterprise Software for Linux in 2024**

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\*\*](#)