

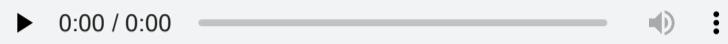
AWS, Cloud Computing, DevOps

 5 Mins Read

## Step-by-Step Guide to Integrate Ansible Dynamic Inventory Plugin for AWS EC2 Instances

www.cloudthat.com

# Step-by-Step Guide to Integrate Ansible Dynamic Inventory Plugin for AWS EC2 Instances


Voiced by [Amazon Polly](#)By [Vineet Negi](#)  September 13, 2022

## Table of Contents

- [Overview](#)
- [Why Ansible Dynamic Inventory?](#)
- [Step-by-Step Guide](#)
- [Method 1: \[Attach Role with EC2 permission\]](#)
- [Method 2: \[AWS Credentials\]](#)
- [Conclusion](#)

## Overview

This blog will talk about how we configure Ansible to get inventory hosts from Amazon Web Services dynamically using the `aws_ec2` plugin.

Ansible is an open-source intelligent automation tool. It is used to automate configuration, provisioning, application deployment, management, and many other manual IT processes. The ansible intelligence is on its configuration management capabilities.

## Why Ansible Dynamic Inventory?

With the rapidly scaling cloud environment, it's difficult to maintain target instances just by using static inventory. Hence Ansible community came up with the concept of dynamic inventory.

We can set up dynamic inventory in two different ways:

1. inventory scripts
2. inventory plugin (recommended)

We will be using the EC2 plugin as it's more up-to-date and easier to set up and manage.

SHARE



Helping organizations transform their IT infrastructure with top-notch Cloud Computing services

- Cloud Migration • Devops • AIML & IoT

[Know More](#)


## Step-by-Step Guide

Let us start working with the ec2 dynamic inventory

1. Install the `aws_ec2` ansible plugin and its dependencies (`boto3` and `botocore`)  
Now a days `aws_ec2` ansible plugin is by default included in the ansible.  
Let's install the dependencies `boto3` and `botocore`.  
\$ `pip3 install boto3 botocore`  
Note: Check the Below link for more information about the [dynamic inventory](#).
2. Setup `ansible.cfg` file

```
[ec2-user@ip-172-31-40-201 ansible]$ pwd
/etc/ansible
[ec2-user@ip-172-31-40-201 ansible]$ cat ansible.cfg
[defaults]
enable_plugins=aws_ec2
[ec2-user@ip-172-31-40-201 ansible]$
```

3. Create inventory `aws_ec2.yml` file



```
keyed_groups:
  - key: tags.Name
filters:
  instance-state-name : running
compose:
  ansible_host: public_ip_address
[ec2-user@ip-172-31-40-201 ansible]$ -
```

```
1 plugin: aws_ec2
2 regions:
3   - "ap-southeast-1"
4 keyed_groups:
5   - key: tags.Name
6 filters:
7   instance-state-name : running
8 compose:
9   ansible_host: public_ip_address
```

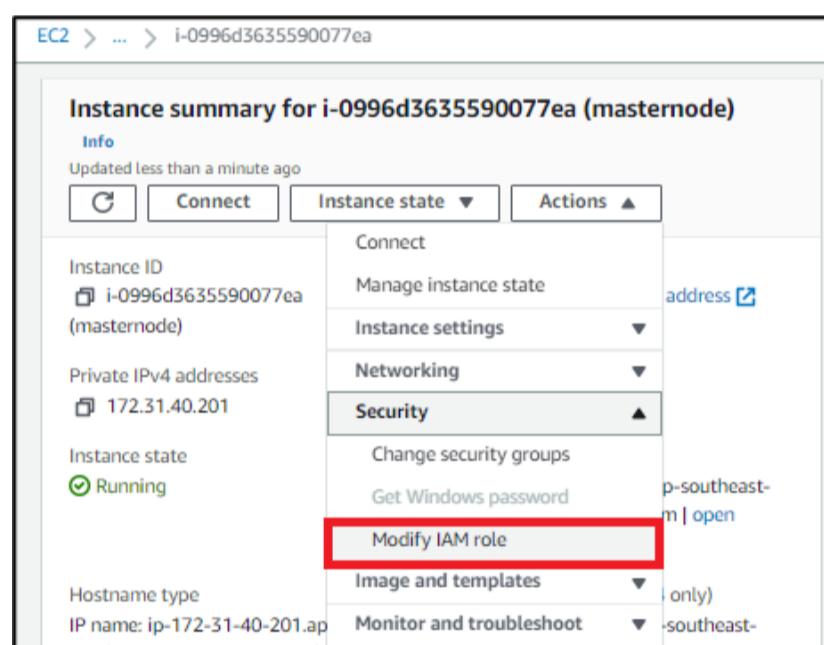
Add the above code to the file we'll be using this access the inventory information.

To use this plugin, we need credentials to access other instances. We can do this in two ways.

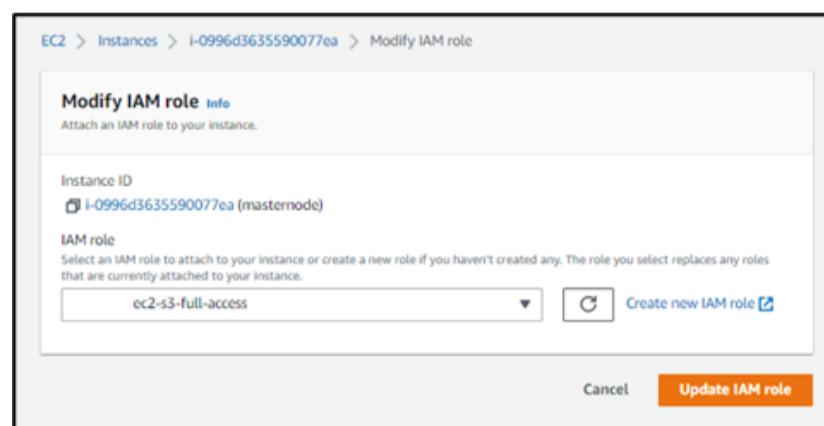
1. Attach Role (aws\_profile) [Recommended]
2. AWS Credentials (aws\_access\_key, aws\_secret\_key)

### Method 1: [Attach Role with EC2 permission]

Go to → Action – Security – Modify IAM (Identity and Access Management) role



Create And Attach role with EC2 permission



Click on Update IAM role and now your ansible have credentials to access the instances information



```

    "_meta": {
      "hostvars": {
        "ec2-13-212-88-103.ap-southeast-1.compute.amazonaws.com": {
          "ami_launch_index": 0,
          "ansible_host": "13.212.88.103",
          "architecture": "x86_64",
          "block_device_mappings": [
            {
              "device_name": "/dev/xvda",
              "ebs": {
                "attach_time": "2022-08-22T09:00:45+00:00",
                "delete_on_termination": true,
                "status": "attached",
                "volume_id": "vol-02344719d4aac20ba"
              }
            }
          ]
        }
      }
    }
  }
}

```

```
1 $ ansible-inventory -i aws_ec2.yml --list
```

## Method 2: [AWS Credentials]

```
1 $ aws configure
```

```
[ec2-user@ip-172-31-40-201 .aws]$ aws configure
AWS Access Key ID [*****]:
AWS Secret Access Key [*****]:
Default region name [ap-southeast-1]:
Default output format [json]:
```

Pass your AWS credentials.

```

[ec2-user@ip-172-31-40-201 ansible]$ ansible-inventory -i aws_ec2.yml --list
[DEPRECATION WARNING]: Ansible will require Python 3.8 or newer on the controller
starting with Ansible 2.12. Current version: 3.7.10 (default, Jun  3 2021,
00:02:01) [GCC 7.3.1 20180712 (Red Hat 7.3.1-13)]. This feature will be removed
from ansible-core in version 2.12. Deprecation warnings can be disabled by setting
deprecation_warnings=False in ansible.cfg.

{
  "_masternode": {
    "hosts": [
      "ec2-13-212-88-103.ap-southeast-1.compute.amazonaws.com"
    ],
    "_meta": {
      "hostvars": {
        "ec2-13-212-88-103.ap-southeast-1.compute.amazonaws.com": {
          "ami_launch_index": 0,
          "ansible_host": "13.212.88.103",
          "architecture": "x86_64",
          "block_device_mappings": [
            {
              "device_name": "/dev/xvda",
              "ebs": {
                "attach_time": "2022-08-22T09:00:45+00:00",
                "delete_on_termination": true,
                "status": "attached",
                "volume_id": "vol-02344719d4aac20ba"
              }
            }
          ]
        }
      }
    }
  }
}

```

```
1 $ ansible-inventory -l aws_ec2.yml --list
```

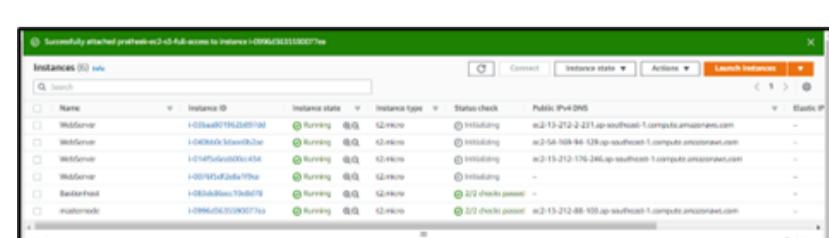
```

{
  "virtualization_type": "hvm",
  "vpc_id": "vpc-51647b36"
}
},
"all": {
  "children": [
    "_masternode",
    "aws_ec2",
    "ungrouped"
  ]
},
"aws_ec2": {
  "hosts": [
    "ec2-13-212-88-103.ap-southeast-1.compute.amazonaws.com"
  ]
}
}

```



Currently, in my **AWS Console** only **one instance** is running, let's launch a few more instances and check the result.



```

"ec2-13-212-2-231.ap-southeast-1.compute.amazonaws.com",
"ec2-13-212-88-103.ap-southeast-1.compute.amazonaws.com",
"ec2-54-169-94-129.ap-southeast-1.compute.amazonaws.com",
"ip-11-0-1-118.ap-southeast-1.compute.internal",
"ip-11-0-2-169.ap-southeast-1.compute.internal"
]
}
[ec2-user@ip-172-31-40-201 ansible]$ -

```

Dynamic Inventory is a working file in this we can see under all we have multiple children these are nothing but tags you are passing to the instances with aws\_ec2 for plugin and ungrouped is by-default children.

To see the graph view for ansible inventory, use this command

```

1 $ ansible-inventory -l aws_ec2.yml --graph
[ec2-user@ip-172-31-40-201 ansible]$ ansible-inventory -l aws_ec2.yml --graph
[DEPRECATION WARNING]: Ansible will require Python 3.8 or newer on the controller
starting with Ansible 2.12. Current version: 3.7.10 (default, Jun 3 2021,
00:02:01) [GCC 7.3.1 20180712 (Red Hat 7.3.1-13)]. This feature will be removed
from ansible-core in version 2.12. Deprecation warnings can be disabled by setting
deprecation_warnings=False in ansible.cfg.

@all:
|-- @_AppServer:
|   |-- ip-11-0-3-33.ap-southeast-1.compute.internal
|-- @_Bastionhost:
|   |-- ip-11-0-1-118.ap-southeast-1.compute.internal
|-- @_Webserver:
|   |-- ec2-13-212-176-246.ap-southeast-1.compute.amazonaws.com
|   |-- ec2-13-212-2-231.ap-southeast-1.compute.amazonaws.com
|   |-- ec2-54-169-94-129.ap-southeast-1.compute.amazonaws.com
|   |-- ip-11-0-2-169.ap-southeast-1.compute.internal
|-- @_masternode:
|   |-- ec2-13-212-88-103.ap-southeast-1.compute.amazonaws.com
|-- @_aws_ec2:
|   |-- ec2-13-212-176-246.ap-southeast-1.compute.amazonaws.com
|   |-- ec2-13-212-2-231.ap-southeast-1.compute.amazonaws.com
|   |-- ec2-13-212-88-103.ap-southeast-1.compute.amazonaws.com
|   |-- ec2-54-169-94-129.ap-southeast-1.compute.amazonaws.com
|   |-- ip-11-0-1-118.ap-southeast-1.compute.internal
|   |-- ip-11-0-2-169.ap-southeast-1.compute.internal
|   |-- ip-11-0-3-33.ap-southeast-1.compute.internal
|-- @ungrouped:

```

Let's connect the inventory with an ansible configuration file.

```

[ec2-user@ip-172-31-40-201 ansible]$ [ec2-user@ip-172-31-40-201 ansible]$ pwd
/etc/ansible
[ec2-user@ip-172-31-40-201 ansible]$ tree
.
├── ansible.cfg
└── aws_ec2.yml

0 directories, 2 files
[ec2-user@ip-172-31-40-201 ansible]$ cat ansible.cfg
[defaults]
enable_plugins=aws_ec2
inventory=aws_ec2.yml
[ec2-user@ip-172-31-40-201 ansible]$

```

Open **ansible.cfg** file and add **inventory=<location\_of\_inventory\_file>**

In my case, config file and dynamic inventory file are in the same location. Hence in the **ansible.cfg** file I have added **inventory=aws\_ec2.yml**.

Let's check whether inventory connects to the configuration file

```

1 $ ansible all --list-hosts

```

The above command lists all the hosts available in the inventory

```

[ec2-user@ip-172-31-40-201 ansible]$ ansible all --list-hosts
[DEPRECATION WARNING]: Ansible will require Python 3.8 or newer on the controller
starting with Ansible 2.12. Current version: 3.7.10 (default, Jun 3 2021,
00:02:01) [GCC 7.3.1 20180712 (Red Hat 7.3.1-13)]. This feature will be removed
from ansible-core in version 2.12. Deprecation warnings can be disabled by setting
deprecation_warnings=False in ansible.cfg.

hosts (7):
    ec2-13-212-193-61.ap-southeast-1.compute.amazonaws.com
    ec2-13-212-176-246.ap-southeast-1.compute.amazonaws.com
    ec2-13-212-2-231.ap-southeast-1.compute.amazonaws.com
    ec2-54-169-94-129.ap-southeast-1.compute.amazonaws.com
    ec2-54-169-225-198.ap-southeast-1.compute.amazonaws.com
    ec2-13-212-88-103.ap-southeast-1.compute.amazonaws.com
    ec2-13-212-113-36.ap-southeast-1.compute.amazonaws.com
[ec2-user@ip-172-31-40-201 ansible]$ -

```

We can even use tags to list the hosts

```

1 $ ansible <tag_name> --list-hosts

```

```

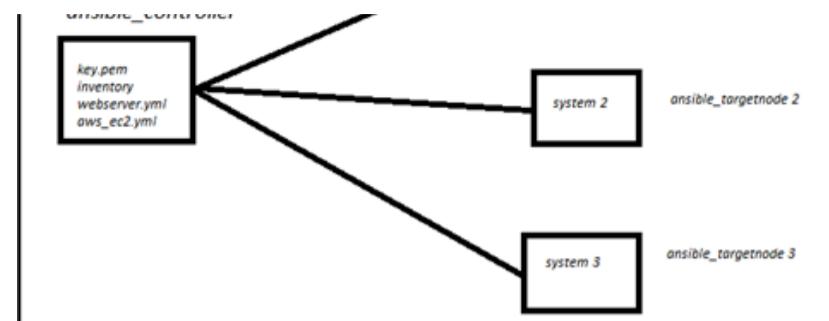
[ec2-user@ip-172-31-40-201 ansible]$ ansible _WebServer --list-hosts
[DEPRECATION WARNING]: Ansible will require Python 3.8 or newer on the controller
starting with Ansible 2.12. Current version: 3.7.10 (default, Jun 3 2021,
00:02:01) [GCC 7.3.1 20180712 (Red Hat 7.3.1-13)]. This feature will be removed
from ansible-core in version 2.12. Deprecation warnings can be disabled by setting
deprecation_warnings=False in ansible.cfg.

hosts (4):
    ec2-13-212-193-61.ap-southeast-1.compute.amazonaws.com
    ec2-13-212-176-246.ap-southeast-1.compute.amazonaws.com
    ec2-13-212-2-231.ap-southeast-1.compute.amazonaws.com
    ec2-54-169-94-129.ap-southeast-1.compute.amazonaws.com
[ec2-user@ip-172-31-40-201 ansible]$

```

As we can use all the results from the **tag\_name Webserver**.





	Name	Instance ID	Instance state	Status check	Public IPv4 ...
Ansible_TargetNode	i-036873bebd8e83dc	Running	0/0	2/2 checks passed	54.254.228.184
Ansible_TargetNode	i-09923a692640a3d81	Running	0/0	2/2 checks passed	13.213.13.10
Ansible_TargetNode	i-0dc09a3e951069ca5	Running	0/0	2/2 checks passed	54.255.228.106
Ansible_Controller_NodeB	i-0b034d12ad21e993e	Running	0/0	2/2 checks passed	18.141.228.229

```
[root@ip-172-31-45-101 ansible]# ansible-inventory -i aws_ec2.yml --graph[DEPRECATION WARNING: The --graph option is deprecated. The controller starting with Ansible 2.12. Current version: 3.7.10 (default, Jun 2021, 00:02:01) [GCC 7.3.1 20180712 (Red Hat 7.3.1-13)]. This feature will be removed from ansible-core in version 2.12. Deprecation warnings can be disabled by setting deprecation_warnings=False in ansible.cfg.

@all:
--@Ansible_Controller_NodeB:
|   |--ec2-18-141-228-229.ap-southeast-1.compute.amazonaws.com
--@Ansible_TargetNode:
|   |--ec2-13-213-13-10.ap-southeast-1.compute.amazonaws.com
|   |--ec2-54-254-228-184.ap-southeast-1.compute.amazonaws.com
|   |--ec2-54-255-228-106.ap-southeast-1.compute.amazonaws.com
--@aws_ec2:
|   |--ec2-13-213-13-10.ap-southeast-1.compute.amazonaws.com
|   |--ec2-18-141-228-229.ap-southeast-1.compute.amazonaws.com
|   |--ec2-54-254-228-184.ap-southeast-1.compute.amazonaws.com
|   |--ec2-54-255-228-106.ap-southeast-1.compute.amazonaws.com
|--@ungrouped:
```

```
[root@ip-172-31-45-101 ansible]# tree
.
├── ansible.cfg
└── aws_ec2.yml
```

As above it is visible that I have a private key file name key.pem which helps me to login into the target node as well as configure the web server on the target node.

Webserver.yml is a playbook where the instruction is defined.

```
[defaults]
inventory=./aws_ec2.yml
host_key_checking=false
remote_user=ec2-user
private_key_file=key.pem

[privilege_escalation]
become=true
become_method=sudo
become_user=root
```

The above file is **ansible.cfg** here the remote\_user helps you to connect with the target with this username and private\_key\_file helps to authenticate the user.

The challenge is to configure the webserver we need root permissions and ec2\_user is not a root user hence we are using privilege escalation to use Sudo. Hence with this, we can configure the web server in the target node.



```
- name: add data to website
  copy:
    content: "Hello from Ansible Controller Node"
    dest: "/var/www/html/index.html"

- name: Start the service
  service:
    name: httpd
    state: started
```

The above file is **webserver.yml**. Used to decide where to run this playbook at the top. We it's visible in the screen shot we are running this playbook on Ansible\_TargetNode.

In the Task part, all the instructions are written like installing the Apache webserver to starting the Apache service.

Now let's run the playbook.

```
1 $ ansible-playbook webserver.yml
```

```
[root@ip-172-31-45-101 ansible]# ansible-playbook webserver.yml
[DEPRECATION WARNING]: Ansible will require Python 3.8 or newer on the
controller starting with Ansible 2.12. Current version: 3.7.10 (default, Jun 3
2021, 00:02:01) [GCC 7.3.1 20180712 (Red Hat 7.3.1-13)]. This feature will be
removed from ansible-core in version 2.12. Deprecation warnings can be disabled
by setting deprecation_warnings=False in ansible.cfg.

PLAY [_Ansible_TargetNode] *****

TASK [Gathering Facts] *****
[WARNING]: Platform linux on host ec2-54-254-228-184.ap-southeast-1.compute.amazonaws.com is using the discovered Python interpreter at
/usr/bin/python, but future installation of another Python interpreter could
change the meaning of that path. See https://docs.ansible.com/ansible-
core/2.11/reference_appendices/interpreter_discovery.html for more information.
pk: [ec2-54-254-228-184.ap-southeast-1.compute.amazonaws.com]
[WARNING]: Platform linux on host ec2-13-213-13-10.ap-southeast-1.compute.amazonaws.com is using the discovered Python interpreter at
/usr/bin/python, but future installation of another Python interpreter could
change the meaning of that path. See https://docs.ansible.com/ansible-
core/2.11/reference_appendices/interpreter_discovery.html for more information.
pk: [ec2-13-213-13-10.ap-southeast-1.compute.amazonaws.com]
[WARNING]: Platform linux on host ec2-54-255-228-106.ap-southeast-1.compute.amazonaws.com is using the discovered Python interpreter at
/usr/bin/python, but future installation of another Python interpreter could
change the meaning of that path. See https://docs.ansible.com/ansible-
core/2.11/reference_appendices/interpreter_discovery.html for more information.
pk: [ec2-54-255-228-106.ap-southeast-1.compute.amazonaws.com]
```

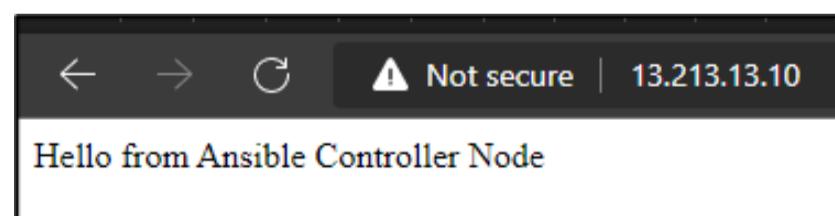
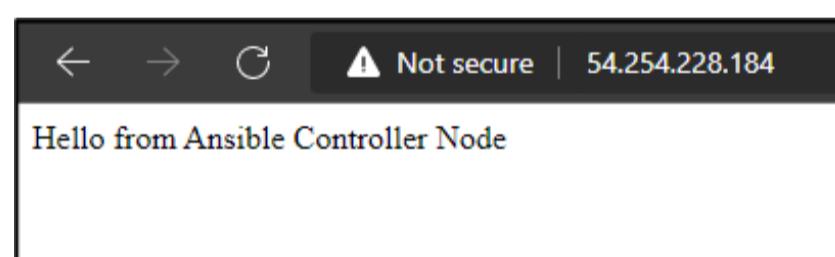
```
TASK [install apache webserver] *****
changed: [ec2-54-255-228-106.ap-southeast-1.compute.amazonaws.com]
changed: [ec2-54-254-228-184.ap-southeast-1.compute.amazonaws.com]
changed: [ec2-13-213-13-10.ap-southeast-1.compute.amazonaws.com]

TASK [Add data to website] *****
changed: [ec2-54-255-228-106.ap-southeast-1.compute.amazonaws.com]
changed: [ec2-54-254-228-184.ap-southeast-1.compute.amazonaws.com]
changed: [ec2-13-213-13-10.ap-southeast-1.compute.amazonaws.com]

TASK [Start the service] *****
changed: [ec2-13-213-13-10.ap-southeast-1.compute.amazonaws.com]
changed: [ec2-54-255-228-106.ap-southeast-1.compute.amazonaws.com]
changed: [ec2-54-254-228-184.ap-southeast-1.compute.amazonaws.com]

PLAY RECAP *****
ec2-13-213-13-10.ap-southeast-1.compute.amazonaws.com : ok=3   changed=3   unreachable=0   failed=0   skipped=0   rescued=0   ignored=0
ec2-54-254-228-184.ap-southeast-1.compute.amazonaws.com : ok=3   changed=3   unreachable=0   failed=0   skipped=0   rescued=0   ignored=0
ec2-54-255-228-106.ap-southeast-1.compute.amazonaws.com : ok=3   changed=3   unreachable=0   failed=0   skipped=0   rescued=0   ignored=0
```

Results of the above command.



In all the target node webserver is configured successfully.

## Conclusion

We learned how to configure dynamic inventory for AWS instances. And how we can use tags to filter the results according to our requirements. Nowadays, dynamic inventory is widely used in many environments as it is automating the inventory. We can also use the dynamic inventory concept in any of the clouds like Azure, GCP, and even for DevOps tools like Docker and Kubernetes. We just need to know the right plugin.



- Cloud Training
- Customized Training
- Experiential Learning

[Read More](#)


## About CloudThat

[CloudThat](#) is also the official AWS (Amazon Web Services) Advanced Consulting Partner and Training partner and Microsoft gold partner, helping people develop knowledge of the cloud and help their businesses aim for higher goals using best in industry cloud computing practices and expertise. We are on a mission to build a robust cloud computing ecosystem by disseminating knowledge on technological intricacies within the cloud space. Our blogs, webinars, case studies, and white papers enable all the stakeholders in the cloud computing sphere.

Drop a query if you have any questions regarding Ansible tool and I will get back to you quickly.

To get started, go through our [Consultancy](#) page and [Managed Services Package](#) that is [CloudThat](#)'s offerings.

▷ [Amazon EC2](#) | [Ansible](#) | [AWS](#) | [DevOps](#) | [dynamic inventory](#) | [Kubernetes](#) |

WRITTEN BY [Vineet Negi](#)



Vineet Negi is a Research Associate at CloudThat. He is part of the Kubernetes vertical and has worked on DevOps and many other Cloud Computing technologies. He is an enthusiastic individual who is passionate about exploring all the latest technologies from a learning perspective.

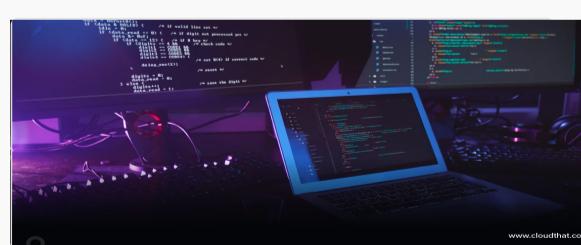
◀ PREV

NEXT ▶

## Comments

Click to Comment

## Related Resources



Tutorials, VMware

How to configure an ESXi Host as an NTP

By Rajiv Gopal

May 31, 2023



Azure, Cloud Computing, Comparision

How Azure Recovery Service Vault Differs from Backup Vault:

By Pankaj P Waghralkar

May 31, 2023



Azure, Cloud Computing, DevOps

Driving DevOps Excellence: Implementing Azure DevOps with.

By Abhilasha D

May 30, 2023





## Get The Most Out Of Us

Our support doesn't end here. We have monthly newsletters, study guides, practice questions, and more to assist you in upgrading your cloud career. Subscribe to get them all!

Yourname@example.com

Subscribe Now



CloudThat is the world's leading training & Consulting services provider on Cloud, DevOps, Security, AI&ML, IoT, and Big Data for midsize and enterprise clients globally.



## Quick links

- [About Us](#)
- [Careers](#)
- [Individual Training](#)
- [Corporate training](#)
- [Consulting](#)
- [Training calendar](#)
- [Test Prep](#)
- [Azure Mastery Pass](#)

## Categories

- [Azure](#)
- [AWS](#)
- [AI & ML](#)
- [IoT](#)
- [DevOps](#)
- [Full Stack Development](#)
- [GCP](#)
- [Microsoft Dynamics](#)
- [Modern Workplace](#)
- [Microsoft Security](#)
- [Power BI](#)
- [Power Platform](#)
- [VMware](#)

## Resources

- [Blogs](#)
- [News and Events](#)
- [Case Study](#)
- [E-Book](#)
- [Webinars](#)

## Contact

102, 4th B Cross, 5th Block  
Koramangala Industrial Area,  
Bangalore, Karnataka - 560095

[sales@cloudthat.com](mailto:sales@cloudthat.com)

India

## India

CLOUDTHAT  
TECHNOLOGIES PRIVATE  
LIMITED  
102, 4th B Cross, 5th Block  
Koramangala Industrial Area,  
Bangalore, Karnataka -  
560095.  
[www.cloudthat.com](http://www.cloudthat.com)

## East India and SAARC

RDB Boulevard, Level 8,  
Plot K-1,  
Block EP & GP, Salt Lake  
City, Sector V, Kolkata – 700  
091.  
[www.cloudthat.com](http://www.cloudthat.com)

## USA

8270 Woodland Center  
Boulevard Tampa, Florida  
33614.  
+1 206-809-1915  
[www.cloudthat.com](http://www.cloudthat.com)

## UK

7B Popin Business Centre  
South Way Wembley  
Middlesex – HA9 0HF.  
[uk.cloudthat.com](http://uk.cloudthat.com)

## Bangladesh

4th Floor, TA - 136,  
Amena Tower,  
Gulshan - Badda Link road,  
Gulshan,  
Dhaka – 1212  
[www.cloudthat.com](http://www.cloudthat.com)

