

[Open in app ↗](#)

# Medium



Search



★ Get unlimited access to the best of Medium for less than \$1/week. [Become a member](#)



# Create an Ansible Cluster in Docker Container & Kubernetes Pod

4 min read · Just now



Gowrigayathri

[Listen](#)[Share](#)[More](#)

## What i am Trying to Do?

1. One **Ansible control node** (also called “Ansible master”) controls
2. One or more **Ansible nodes** (also called “managed nodes”) using SSH.
3. Everything runs inside **Docker containers** (or later, inside **Kubernetes Pods**).
4. **Here i am not using pre-built images** – instead, building my own image from scratch.



## Step-by-Step Breakdown :

- 1. **Dockerfile** – Build my own Ansible-ready image

```
FROM ubuntu:22.04

ENV DEBIAN_FRONTEND=noninteractive

# Install dependencies
RUN apt-get update && \
    apt-get install -y python3 python3-pip openssh-server sudo curl vim && \
    apt-get clean && \
    rm -rf /var/lib/apt/lists/*
```

```
# Install Ansible
RUN pip3 install ansible

# Create SSH directory
RUN mkdir -p /var/run/sshd

# Create ansible user and allow sudo
RUN useradd -m ansible && \
    echo "ansible:ansible" | chpasswd && \
    usermod -aG sudo ansible

# Set SSH to start when the container runs
EXPOSE 22
CMD ["/usr/sbin/sshd", "-D"]
```

## ✓ 2. Build the Image

Save the Dockerfile in a folder, then run:

```
docker build -t ansible-img .
```

```
vboxuser@ubuntu24:/ansible-dc-k8s
* ansible-dc-k8s docker build -t ansible-img .
[+] Building 0.4s (8/8) FINISHED
=> [internal] load build definition from dockerfile
=> [internal] transfering dockerfile: 620B
=> [internal] load metadata for docker.io/library/ubuntu:22.04
=> [internal] load build context
=> [internal] transfering context: 2B
=> [1/4] FROM docker.io/library/ubuntu:22.04@sha256:01a3ee0b5e413cefaafffc6abe68c9c37879ae3cced56a8e088b1649e5b2690ee
=> CACHED [2/4] RUN apt-get update && apt-get install -y python3 python3-pip openssh-server sudo curl vim sshpass && apt-get clean && rm -rf /var/lib/apt/lists/*
=> CACHED [3/4] RUN pip3 install ansible
=> CACHED [4/4] RUN mkdir -p /var/run/sshd && useradd -m ansible && echo "ansible:ansible" | chpasswd && usermod -aG sudo ansible
=> exporting to image
=>   using cache layers
=> writing manifest sha256:849b95b4d435a680d86efecff54e721529b5aca077f683c1f5dd4b357da605
=> naming to docker.io/library/ansible-img
* ansible-dc-k8s
```

## ✓ 3. Run the Containers

Let's create 1 master and 2 nodes.

```
docker run -d --name master ansible-img
docker run -d --name node1 ansible-img
docker run -d --name node2 ansible-img
```

#### 4. Setup SSH Between Master and Nodes

Now we allow **Ansible on master** to connect to **node1** and **node2**.

```
# Enter master container
docker exec -it master bash

# Switch to ansible user
su - ansible

# Generate SSH key (press Enter to accept defaults)
ssh-keygen

# Get node1 and node2 IPs from host machine:
# (Run this in another terminal)
docker inspect -f '{{ .NetworkSettings.IPAddress }}' node1
docker inspect -f '{{ .NetworkSettings.IPAddress }}' node2
```

After running above cmds we id's of node1 and node1 then by using those connect the master node and node1 & node2 .

Then inside the master container:

```
# Copy the SSH key to node1
ssh-copy-id ansible@<node1_ip>

# Copy the SSH key to node2
ssh-copy-id ansible@<node2_ip>
```

#### 5. Create Inventory File

```
# /home/ansible/inventory.ini
[nodes]
```

```
<node1_ip> ex: 172.17.0.3
<node2_ip> ex: 172.17.0.4
```

## ✓ 6. Test Ansible Ping

```
ansible -i inventory.ini nodes -m ping
```

You should see:

```
<node1_ip> | SUCCESS => ...
<node2_ip> | SUCCESS => ...
```

```
root@050e816b2c99:/ # ansible -i inventory.ini nodes -m ping
[WARNING]: Platform linux on host 172.17.0.3 is using the discovered Python interpreter at /usr/bin/python3.10, but future installation of another Python interpreter could change the meaning of that path. See https://docs.ansible.com/ansible-core/2.17/reference_appendices/interpreter_discovery.html for more information.
172.17.0.3 | SUCCESS => {
  "ansible_facts": {
    "discovered_interpreter_python": "/usr/bin/python3.10"
  },
  "changed": false,
  "ping": "pong"
}
[WARNING]: Platform linux on host 172.17.0.4 is using the discovered Python interpreter at /usr/bin/python3.10, but future installation of another Python interpreter could change the meaning of that path. See https://docs.ansible.com/ansible-core/2.17/reference_appendices/interpreter_discovery.html for more information.
172.17.0.4 | SUCCESS => {
  "ansible_facts": {
    "discovered_interpreter_python": "/usr/bin/python3.10"
  },
  "changed": false,
  "ping": "pong"
}
root@050e816b2c99:/ #
```

creating ansible cluster in docker container is completed upto here

## ✓ Summary of What i Did

- Built a custom Docker image with Ansible and SSH
- Created 3 containers: 1 master (control node) + 2 nodes

- Configured SSH so Ansible can control the nodes
- Used Ansible to ping nodes to confirm everything works

## Ansible Cluster in Kubernetes Pod

### Architecture

- One Pod with Ansible master container
- Additional Pods for nodes (or use multi-container Pod is needed)

#### Step 1: Use Same Dockerfile (above) and Push to DockerHub

```
docker tag ansible-img <your_dockerhub_username>/ansible-img
docker push <your_dockerhub_username>/ansible-img
```

#### Step 2: Create Deployment YAML for Master

```
# ansible-master.yaml
apiVersion: v1
kind: Pod
metadata:
  name: ansible-master
spec:
  containers:
    - name: master
      image: gayathrigopireddy45/ansible-img
      securityContext:
        privileged: true
      command: ["/usr/sbin/sshd", "-D"]
```

#### Step 3: Create Deployments for Nodes

```
# ansible-node1.yaml
apiVersion: v1
kind: Pod
metadata:
  name: ansible-node1
spec:
```

```
containers:  
- name: node1  
  image: gayathrigopireddy45/ansible-img  
  securityContext:  
    privileged: true  
  command: ["/usr/sbin/sshd", "-D"]
```

```
# ansible-node1.yaml  
apiVersion: v1  
kind: Pod  
metadata:  
  name: ansible-node2  
spec:  
  containers:  
  - name: node2  
    image: gayathrigopireddy45/ansible-img  
    securityContext:  
      privileged: true  
    command: ["/usr/sbin/sshd", "-D"]
```

## Apply the YAML Files to Create Pods

```
kubectl apply -f ansible-master.yaml  
kubectl apply -f ansible-node1.yaml  
kubectl apply -f ansible-node2.yaml # If you created a second node
```

### 🔍 Verify Pods:

```
kubectl get pods
```

## Get Inside Ansible Master Pod:

```
kubectl exec -it ansible-master -- bash
```

## Get IP Addresses of Nodes

```
kubectl get pods -o wide
```

## SSH Setup (Inside Master Pod)

```
#Generate SSH Key  
ssh-keygen
```

## Copy SSH Key to Node Pods (use their IPs)

```
ssh-copy-id ansible@<node1-ip>  
ssh-copy-id ansible@<node2-ip>
```

## Create Your `inventory.ini` File in Master

```
ansible -i inventory.ini nodes -m ping
```

Output should be:

```
root@ansible-master:/# ansible -i inventory.ini nodes -m ping
[WARNING]: Unable to parse /home/gowrigayathri/ansible-dc-k8s/inventory.ini as an inventory source
[WARNING]: No inventory was parsed, only implicit localhost is available
[WARNING]: provided hosts list is empty, only localhost is available. Note that the implicit localhost does not match 'all'
[WARNING]: Could not match supplied host pattern, ignoring: nodes
+ ansible -dc-k8s kubectl exec -it ansible-master -- bash
root@ansible-master:/# ls
bin  boot  dev  etc  home  lib  lib32  lib64  libx32  media  mnt  opt  proc  root  run  sbin  srv  sys  tmp  usr  var
root@ansible-master:/# ansible -i inventory.ini nodes -m ping
[WARNING]: Platform linux on host 10.244.0.8 is using the discovered Python interpreter at /usr/bin/python3.10, but future installation of another Python interpreter could change the meaning of that path. See https://docs.ansible.com/ansible-core/2.17/reference_appendices/interpreter_discovery.html for more information.
10.244.0.8 | SUCCESS => {
    "ansible_facts": {
        "discovered_interpreter_python": "/usr/bin/python3.10"
    },
    "changed": false,
    "ping": "pong"
}
[WARNING]: Platform linux on host 10.244.0.9 is using the discovered Python interpreter at /usr/bin/python3.10, but future installation of another Python interpreter could change the meaning of that path. See https://docs.ansible.com/ansible-core/2.17/reference_appendices/interpreter_discovery.html for more information.
10.244.0.9 | SUCCESS => {
    "ansible_facts": {
        "discovered_interpreter_python": "/usr/bin/python3.10"
    },
    "changed": false,
    "ping": "pong"
}
root@ansible-master:/#
```


[Edit profile](#)

## Written by Gowrigayathri

1 follower · 2 following

No responses yet


...


Gowrigayathri

What are your thoughts?

## Recommended from Medium



 iris 9e

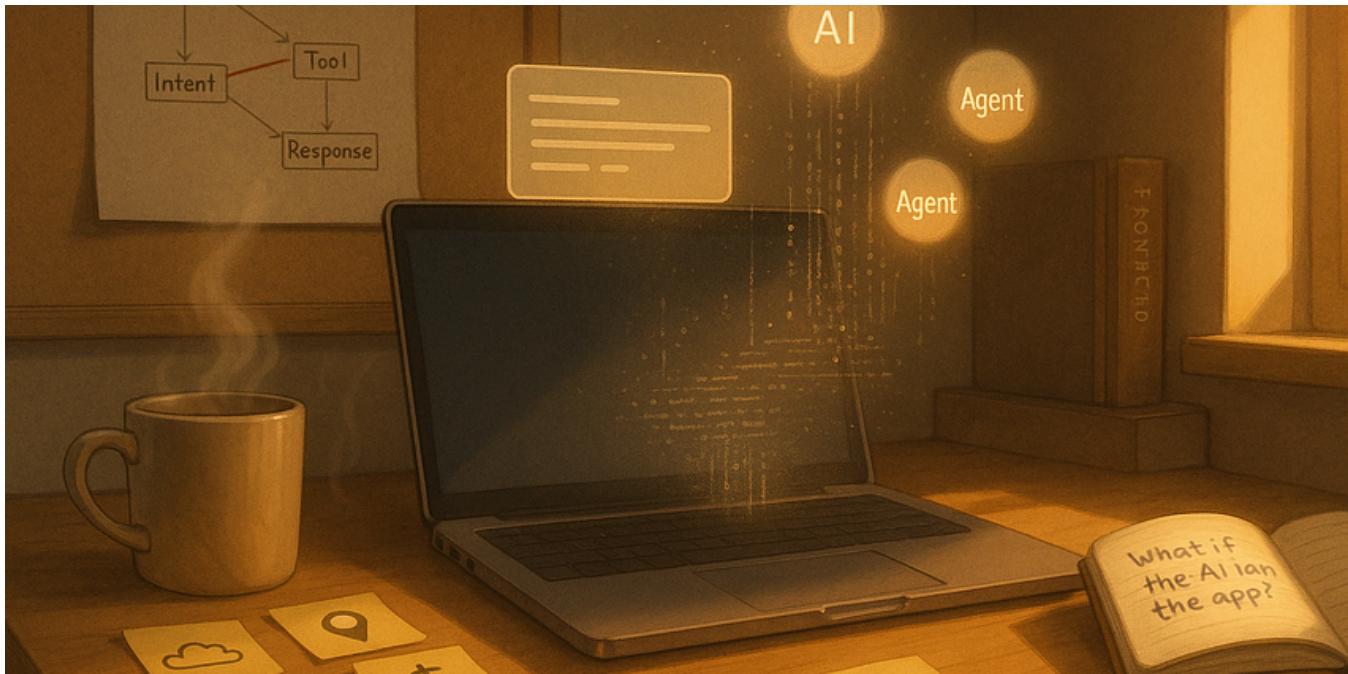
### **you don't owe anyone a version of you that doesn't exist anymore**

you were not made to stay still just because someone built a home in your old self

6d ago     3.2K     56



...



**JS** In JavaScript in Plain English by GeekSociety

## I Stopped Building Frontends. Now I Use MCP Servers to Let AI Run My Apps

It's 2025, and the way we build applications has fundamentally changed.

Jun 2 4K 155



**LONG** In Long. Sweet. Valuable. by Ossai Chinedum

## I'll Instantly Know You Used Chat Gpt If I See This

Trust me you're not as slick as you think

May 16

7.9K

430



# These 40 Open-Source Tools Will Replace Half Your Tech Stack (And Some Tools are Free)



In Let's Code Future by TheMindShift

## These 40 Open-source Tools Will Replace Half Your Tech Stack (And Some Tools are Free)

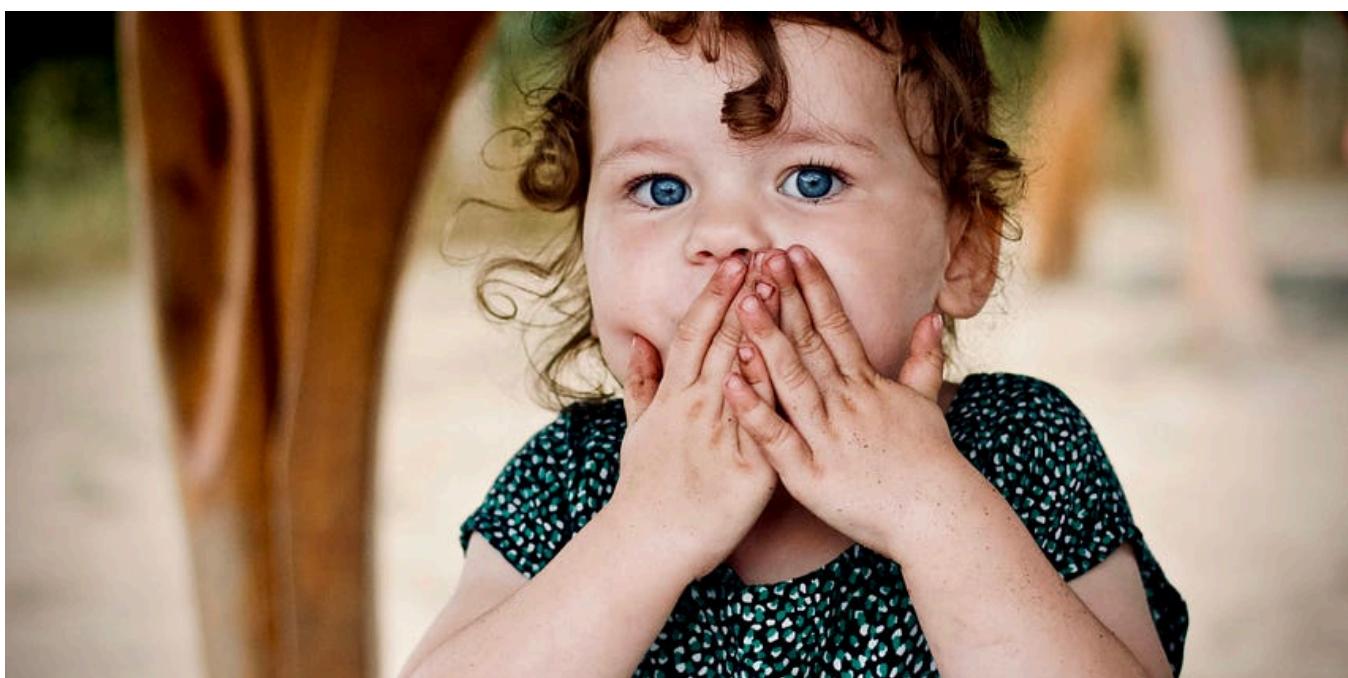
I curated 40 powerful, production-ready open-source tools that every serious developer should have in their toolbox.



Jun 2

1.5K

27



In Age of Awareness by Sam Westreich, PhD

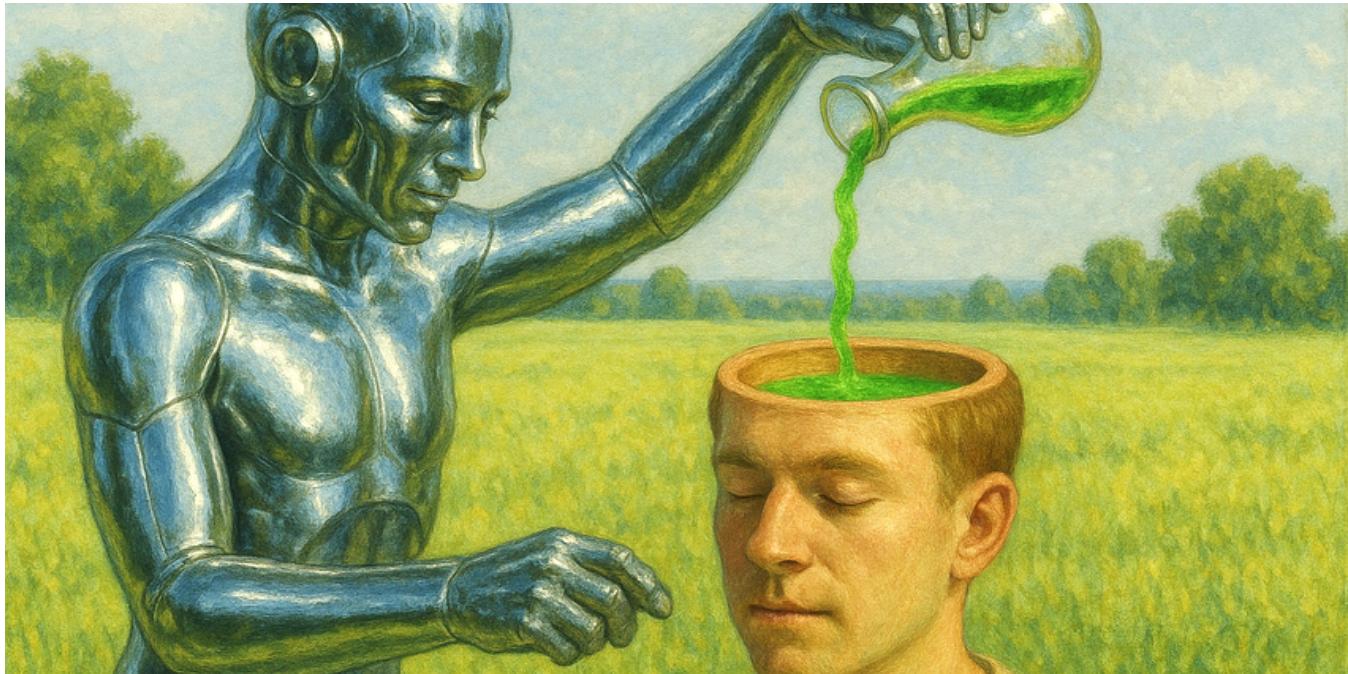
## Why ChatGPT Creates Scientific Citations—That Don't Exist

It looks convincing, even in White House reports, but it's often a sign of shoddy backwards research.

6d ago 5.6K 167



...



 Jordan Gibbs

## ChatGPT Is Poisoning Your Brain...

Here's How to Stop It Before It's Too Late.

Apr 30 17.9K 826



...

See more recommendations