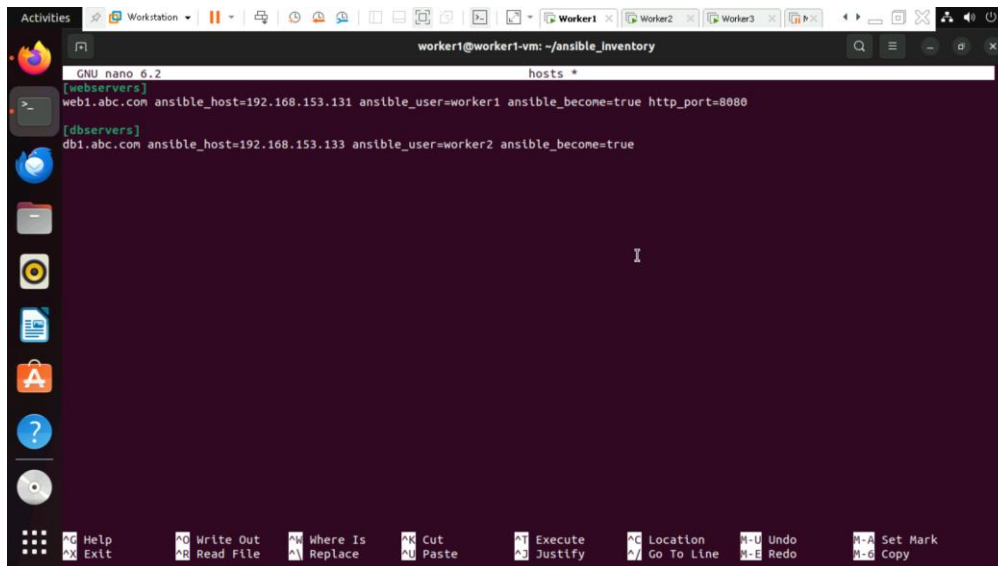


1. Create Inventory File

```
worker1@worker1-vm:~/webserver_playbook$ cd ..  
worker1@worker1-vm:~$ mkdir -p ~/ansible_inventory && cd ~/ansible_inventory  
worker1@worker1-vm:~/ansible_inventory$ nano hosts  
worker1@worker1-vm:~/ansible_inventory$
```



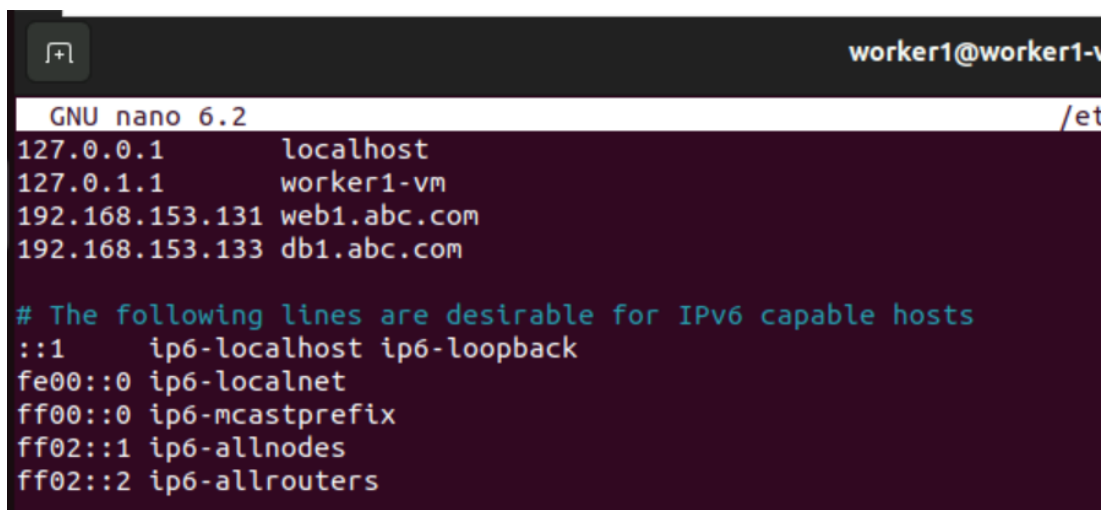
The screenshot shows a terminal window with the nano 6.2 editor open. The file being edited is named 'hosts'. The content of the file is as follows:

```
hosts *  
[webservers]  
web1.abc.com ansible_host=192.168.153.131 ansible_user=worker1 ansible_become=true http_port=8080  
[dbservers]  
db1.abc.com ansible_host=192.168.153.133 ansible_user=worker2 ansible_become=true
```

The terminal window has a title bar that reads 'worker1@worker1-vm: ~/ansible_inventory'. The bottom of the window shows the nano editor's command shortcuts.

2. Add to /etc/hosts

```
worker1@worker1-vm: ~/ansible_inventory  
worker1@worker1-vm:~/ansible_inventory$ sudo nano /etc/hosts  
[sudo] password for worker1:  
worker1@worker1-vm:~/ansible_inventory$
```

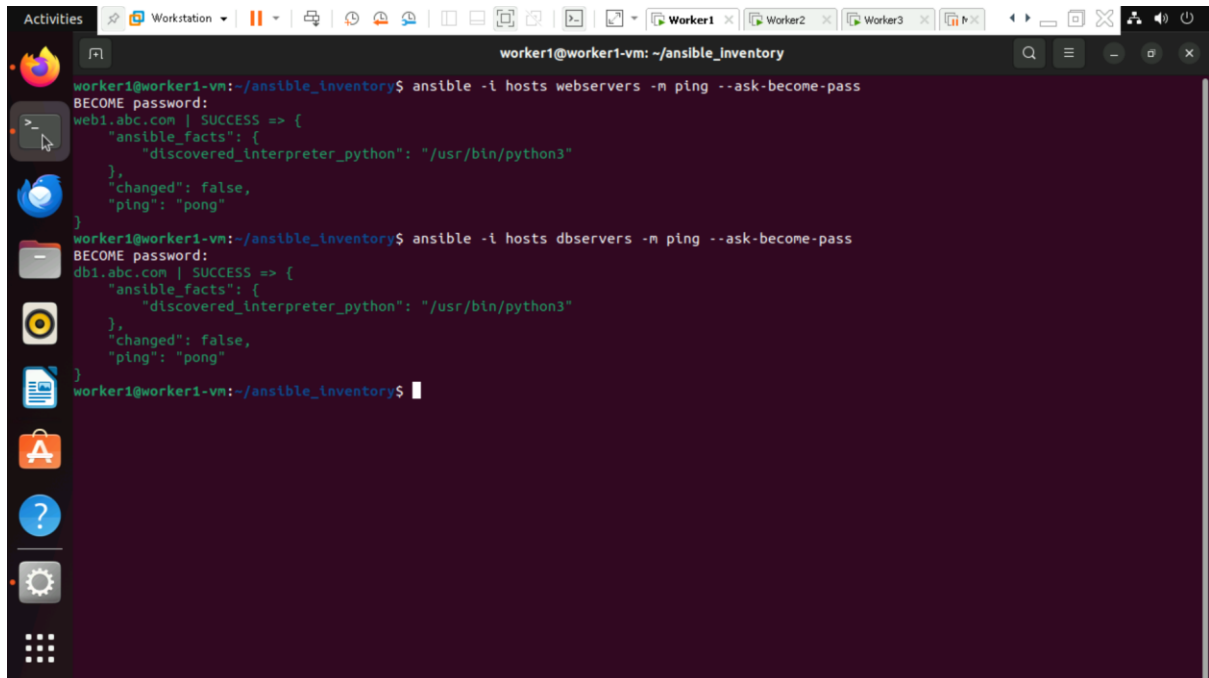


The screenshot shows the nano 6.2 editor editing the /etc/hosts file. The content of the file is:

```
GNU nano 6.2 /et  
127.0.0.1 localhost  
127.0.1.1 worker1-vm  
192.168.153.131 web1.abc.com  
192.168.153.133 db1.abc.com  
  
# The following lines are desirable for IPv6 capable hosts  
::1 ip6-localhost ip6-loopback  
fe00::0 ip6-localnet  
ff00::0 ip6-mcastprefix  
ff02::1 ip6-allnodes  
ff02::2 ip6-allrouters
```

The terminal window title is 'worker1@worker1-vm: ~/ansible_inventory'.

3. Test and Output:



The image shows a terminal window titled "worker1@worker1-vm: ~/ansible_inventory". The terminal displays two Ansible commands and their outputs. The first command is `ansible -i hosts webservers -m ping --ask-become-pass`, which prompts for a password and returns a success message for `web1.abc.com`. The second command is `ansible -i hosts dbservers -m ping --ask-become-pass`, which also prompts for a password and returns a success message for `db1.abc.com`. The terminal output for both commands is identical, showing a JSON structure with `ansible_facts`, `discovered_interpreter_python`, `changed`, and `ping` fields.

```
worker1@worker1-vm:~/ansible_inventory$ ansible -i hosts webservers -m ping --ask-become-pass
BECOME password:
web1.abc.com | SUCCESS => {
  "ansible_facts": {
    "discovered_interpreter_python": "/usr/bin/python3"
  },
  "changed": false,
  "ping": "pong"
}
worker1@worker1-vm:~/ansible_inventory$ ansible -i hosts dbservers -m ping --ask-become-pass
BECOME password:
db1.abc.com | SUCCESS => {
  "ansible_facts": {
    "discovered_interpreter_python": "/usr/bin/python3"
  },
  "changed": false,
  "ping": "pong"
}
worker1@worker1-vm:~/ansible_inventory$
```