

Automated Server Configuration, Deployment, and Maintenance for Multiple Environments: Documentation

Phase I: Prerequisites

Virtual Machine Creation

Environment I: Development

Ansible Server (AMD-01): E2 Small

Operating System: CentOS

Region: us-central1

Web Server (WSD-01): E2 Small

Operating System: Debian

Region: us-east1

Firewall: Allow HTTP Traffic; Allow HTTPS Traffic

Web Server/Database (WDD-01): E2 Small

Operating System: Debian

Region: us-west1

Firewall: Allow HTTP Traffic; Allow HTTPS Traffic

Environment II: Test

Ansible Server (AMT-01): E2 Small

Operating System: CentOS

Region: us-central1

Web Server (WST-01): E2 Small

Operating System: Debian

Region: us-east1

Firewall: Allow HTTP Traffic; Allow HTTPS Traffic

Web Server/Database (WDT-01): E2 Small

Operating System: Debian

Region: us-west1

Firewall: Allow HTTP Traffic; Allow HTTPS Traffic

Environment III: Production

Ansible Server (AMP-01): E2 Small

Operating System: CentOS

Region: us-central1

Web Server (WSP-01): E2 Small

Operating System: Debian

Region: us-east1

Firewall: Allow HTTP Traffic; Allow HTTPS Traffic

Web Server/Database (WDP-01): E2 Small

Operating System: Debian

Region: us-west1

Firewall: Allow HTTP Traffic; Allow HTTPS Traffic

Virtual Machine Root Security

The following two steps should be applied to all servers:

1. Reset Root Password
 - a. SSH into your machines.
 - b. Type `sudo passwd` in the CLI.
 - c. Type in your password when prompted twice. Note that what you are typing will not show up on the screen for security purposes.
2. Set Up SSH_Config File
 - a. Type `sudo`
 - b. On (or around) line 38, uncomment - that is, delete the # on that line - "PermitRootLogin yes."
 - c. On (or around) line 63, uncomment "PasswordAuthentication yes."
 - d. On (or around) line 65, comment out or delete "PasswordAuthentication no."

Phase II: Configuration

Reset Root Password: All Machines

SSH into your machines.

Type the following command in the CLI:

```
$ sudo passwd
```

Type in your password when prompted twice. Note that what you are typing will not show up on the screen for security purposes.

Initial Installs: Ansible Machines (AMD-01, AMT-01, AMP-01)

Type the following commands in the CLI:

```
$ sudo yum install ansible
$ sudo yum install nano
```

When prompted for a response, type "y" or "yes."

Set Up ssh_config File: All Machines

Type the following command:

```
$ sudo nano /etc/ssh/sshd_config
```

On (or around) line 38, uncomment - that is, delete the # - "PermitRootLogin yes."

On (or around) line 63, uncomment "PasswordAuthentication yes."

On (or around) line 65, comment out or delete "PasswordAuthentication no."

Edit the Ansible Config Files: Ansible Machines (AMD-01, AMT-01, AMP-01)

Type the following command:

```
$ sudo nano /etc/ansible/ansible.cfg
```

On (or around) line 71, uncomment "host_key_checking = False."

Edit the Ansible Hosts Files: Ansible Machines (AMD-01, AMT-01, AMP-01)

Type the following command:

```
$ sudo nano /etc/ansible/hosts
```

For AMD-01, insert the following lines of code:

```
# Grouped Development Servers
[devservers]
10.128.0.11
10.142.0.5
```

***NOTE: Replace the IP addresses above with the internal IP addresses of the VMs that YOU created.

For AMT-01, insert the following lines of code:

```
# Grouped Test Servers
[testservers]
10.128.0.13
10.128.0.15
```

***NOTE: Replace the IP addresses above with the internal IP addresses of the VMs that YOU created.

For AMP-01, insert the following lines of code:

```
# Grouped Production Servers
[prodservers]
10.128.0.12
10.142.0.14
```

***NOTE: Replace the IP addresses above with the internal IP addresses of the VMs that YOU created.

Phase III: Deployment

Install Apache, NodeJS, and Git on Development Servers

On AMD-01, type the following commands in the CLI:

```
$ cd ~
$ mkdir ansible-playbooks
$ cd ansible-playbooks
$ nano dev-apache-nodejs-git-install.yml
```

Inside dev-apache-nodejs-git-install.yml, insert the following code:

```
---
- name: Install Apache and NodeJS
  hosts: devwebserver
  tasks:
    - name: Install Apache
      yum:
        name: apache2
        state: present
    - name: Install NodeJS
      yum:
        name: nodejs
        state: present
```

Hit Ctrl+X, Y, and Enter to exit the YAML file.

Type the following command:

```
$ ansible-playbook dev-apache-nodejs-git-install.yml
```

Install Apache, NodeJS, and Git on Test Servers

On AMT-01, type the following commands in the CLI:

```
$ nano test-apache-nodejs-git-install.yml
```

Inside test-apache-nodejs-git-install.yml, insert the following code:

```
---
- name: Install Apache and NodeJS
  hosts: testwebserver
  tasks:
    - name: Install Apache
      yum:
```

```
    name: apache2
    state: present
- name: Install NodeJS
  yum:
    name: nodejs
    state: present
```

Hit Ctrl+X, Y, and Enter to exit the YAML file.

Type the following command:

```
$ ansible-playbook test-apache-nodejs-git-install.yml
```

Install Apache, NodeJS, and Git on Production Servers

On AMP-01, type the following commands in the CLI:

```
$ nano prod-apache-nodejs-git-install.yml
```

Inside prod-apache-nodejs-git-install.yml, insert the following code:

```
---
- name: Install Apache and NodeJS
  hosts: prodwebservers
  tasks:
    - name: Install Apache
      yum:
        name: apache2
        state: present
    - name: Install NodeJS
      yum:
        name: nodejs
        state: present
```

Hit Ctrl+X, Y, and Enter to exit the YAML file.

Type the following command:

```
$ ansible-playbook prod-apache-nodejs-git-install.yml
```

Install Apache, NodeJS, and Git Troubleshooting

If you receive some sort of error that looks like this:

```
PLAY [Install Apache and NodeJS] *****
TASK [Gathering Facts] *****
[WARNING]: Platform linux on host 10.128.0.14 is using the discovered python interpreter at /usr/bin/python3, but future installation of another Python interpreter could change this. See
https://docs.ansible.com/ansible/2.9/reference_appendices/interpreter_discovery.html for more information.
[WARNING]: Platform linux on host 10.128.0.12 is using the discovered python interpreter at /usr/bin/python3, but future installation of another Python interpreter could change this. See
https://docs.ansible.com/ansible/2.9/reference_appendices/interpreter_discovery.html for more information.
[WARNING]: ansible-py3 is not installed on host 10.128.0.12
TASK [Install Apache] *****
10.128.0.14: FAILED! => "ansible_facts": "os_family": "RedHat", "ansible_update_time": "1700000000", "ansible_update": false, "changed": false, "msg": "''/usr/bin/apt-get -y -o 'Dpkg::Options::=--force-confdef' -o 'Dpkg::Options::=--force-confold' failed: E: Could not get lock /var/lib/dpkg/lock-frontent. It is held by process 41942 (apt-get)!"! Unable to acquire the dpkg frontend lock (/var/lib/dpkg/lock-frontent), is another process using it?!", "stderr_lines": ["E: Could not get lock /var/lib/dpkg/lock-frontent. It is held by process 41942 (apt-get)!", "E: Unable to acquire the dpkg frontend lock (/var/lib/dpkg/lock-frontent), is another process using it?!", "stdout": "", "stdout_lines": []]
10.128.0.12: FAILED! => "ansible_facts": "os_family": "RedHat", "ansible_update_time": "1700000000", "ansible_update": false, "changed": false, "msg": "''/usr/bin/apt-get -y -o 'Dpkg::Options::=--force-confdef' -o 'Dpkg::Options::=--force-confold' failed: E: Could not get lock /var/lib/dpkg/lock-frontent. It is held by process 41917 (apt-get)!"! Unable to acquire the dpkg frontend lock (/var/lib/dpkg/lock-frontent), is another process using it?!", "stderr_lines": ["E: Could not get lock /var/lib/dpkg/lock-frontent. It is held by process 41917 (apt-get)!", "E: Unable to acquire the dpkg frontend lock (/var/lib/dpkg/lock-frontent), is another process using it?!", "stdout": "", "stdout_lines": []]
PLAY RECAP *****
10.128.0.14: 1 task: 1 failed, 0 changed=0, unreachable=0, failed=1, skipped=0, rescued=0, ignored=0
[amkitchen42@prod-ansible-server ~]$
```

while your ansible playbook is running, type the following command in the CLI:

```
$ nano remove-lock-troubleshoot.yml
```

I encountered this error while trying to install Apache on my production servers, so I deployed this playbook on my prodwebserver host group:

```
---
- name: Remove apt lock file
  hosts: prodwebserver
  tasks:
    - name: remove lock
      file:
        state: absent
        path: "/var/lib/dpkg/lock-frontent"
```

If this error occurs on a different host group, be sure to change the hosts in the YAML file.

Hit Ctrl+X, Y, and Enter to exit the YAML file, then type the following command:

```
$ ansible-playbook remove-lock-troubleshoot.yml
```

Install MariaDB on WDD-01

On AMD-01, type the following command in the CLI:

```
$ nano dev-mariadb-install.yml
```

Inside dev-mariadb-install.yml, insert the following code:

```
---
- name: Install MariaDB
  hosts: 10.128.0.11
  tasks:
    - name: Install MariaDB
      yum:
        name: mariadb-server
        state: present
```

***NOTE: Replace the IP address above with the internal IP address of the VM that YOU created.

Hit Ctrl+X, Y, and Enter to exit the YAML file.

Type the following command:

```
$ ansible-playbook dev-mariadb-install.yml
```

Install MariaDB on WDT-01

On AMT-01, type the following command in the CLI:

```
$ nano test-mariadb-install.yml
```

Inside test-mariadb-install.yml, insert the following code:

```
---
- name: Install MariaDB
  hosts: 10.128.0.13
  tasks:
    - name: Install MariaDB
      yum:
        name: mariadb-server
        state: present
```

***NOTE: Replace the IP address above with the internal IP address of the VM that YOU created.

Hit Ctrl+X, Y, and Enter to exit the YAML file.

Type the following command:

```
$ ansible-playbook test-mariadb-install.yml
```

Install MariaDB on WDP-01

On AMP-01, type the following command in the CLI:

```
$ nano prod-mariadb-install.yml
```

Inside prod-mariadb-install.yml, insert the following code:

```
---
- name: Install MariaDB
  hosts: 10.128.0.12
  tasks:
    - name: Install MariaDB
      yum:
```

```
name: mariadb-server
state: present
```

***NOTE: Replace the IP address above with the internal IP address of the VM that YOU created.

Hit Ctrl+X, Y, and Enter to exit the YAML file.

Type the following command:

```
$ ansible-playbook prod-mariadb-install.yml
```

If you are ever prompted with a yes/no question for these installs, type “y” or “yes.”

Git Clone on Development Servers

On AMD-01, type the following commands in the CLI:

```
$ nano dev-github-clone.yml
```

Inside dev-github-clone.yml, insert the following code:

```
---
- name: Clone repo
  hosts: devwebservers
  tasks:
    - name: delete path
      file:
        state: absent
        path: /var/www/html
    - name: Clone Git Repository
      git:
        repo: https://github.com/ttu-bburchfield/swollenhippofinal.git
        dest: "/var/www/html"
        version: "test"
```

Hit Ctrl+X, Y, and Enter to exit the YAML file.

Type the following command:

```
$ ansible-playbook dev-github-clone.yml
```

Git Clone on Test Servers

On AMT-01, type the following commands in the CLI:

```
$ nano test-github-clone.yml
```

Inside test-github-clone.yml, insert the following code:


```

---
- name: Clone repo
  hosts: testwebservers
  tasks:
    - name: delete path
      file:
        state: absent
        path: /var/www/html
    - name: Clone Git Repository
      git:
        repo: https://github.com/ttu-bburchfield/swollenhippofinal.git
        dest: "/var/www/html"
        version: "test"

```

Hit Ctrl+X, Y, and Enter to exit the YAML file.

Type the following command:

```
$ ansible-playbook test-github-clone.yml
```

Git Clone on Production Servers

On AMP-01, type the following commands in the CLI:

```
$ nano prod-github-clone.yml
```

Inside prod-github-clone.yml, insert the following code:

```

---
- name: Clone repo
  hosts: prodwebservers
  tasks:
    - name: delete path
      file:
        state: absent
        path: /var/www/html
    - name: Clone Git Repository
      git:
        repo: https://github.com/ttu-bburchfield/swollenhippofinal.git
        dest: "/var/www/html"
        version: "test"

```

Hit Ctrl+X, Y, and Enter to exit the YAML file.

Type the following command:

```
$ ansible-playbook prod-github-clone.yml
```

Alternatively, you can set up SSH keys to access the github repository that way; however, it takes a lot more work and isn't necessary.

Automation Script

Before running the following script, please complete Phase I the following tasks under Phase II:

1. Reset Root Password
2. Initial Installs
3. Set Up ssh_config

On your ansible server(s), type the following commands:

```
$ cd ~  
$ mkdir scripts  
$ cd scripts  
$ nano server-setup.sh
```

Inside server-setup.sh, insert the following lines of code:

```
#!/usr/bin/env bash  
  
# Set parameters  
environment=$1  
ip1=$2  
ip2=$3  
password=$4  
  
# Assume root so you can edit the config and hosts files  
sudo su  
  
# Set up ansible.cfg  
echo "host_key_checking = False" >> /etc/ansible/ansible.cfg  
  
# Set up hosts file  
echo " #Grouped $environment" >> /etc/ansible/hosts  
echo " [$environment]" >> /etc/ansible/hosts  
echo " $ip1" >> /etc/ansible/hosts  
echo " $ip2" >> /etc/ansible/hosts  
  
echo " [$environment:vars]" >> /etc/ansible/hosts
```

```
echo "  ansible_user=root" >> /etc/ansible/hosts
echo "  ansible_password=$4" >> /etc/ansible/hosts
```

```
#run ansible scripts
mkdir ansible-playbooks
cd ansible-playbooks
```

```
cat > remove-lock-troubleshoot.yml << 'endmsg'
```

```
---
```

- name: Remove apt lock file
- hosts: \$environment
- tasks:
 - name:
 - file:
 - state: absent
 - path: "/var/lib/dpkg/lock-frontent"

```
endmsg
```

```
cat > server-setup.yml << 'endmsg'
```

```
---
```

- name: Install Apache and NodeJS
- hosts: \$environment
- tasks:
 - name: Install Apache
 - yum:
 - name: apache2
 - state: present
 - name: Install NodeJS
 - yum:
 - name: nodejs
 - state: present

```
endmsg
```

```
cat > mariadb-install.yml << 'endmsg'
```

```
---
```

- name: Install MariaDB
- hosts: \$ip2
- tasks:
 - name: Install MariaDB
 - yum:
 - name: mariadb-server
 - state: present

```
endmsg
```

```

cat > github-clone.yml << 'endmsg'
---
- name: Clone repo
  hosts: $environment
  tasks:
    - name: delete path
      file:
        state: absent
        path: /var/www/html
    - name: Clone Git Repository
      git:
        repo: https://github.com/ttu-bburchfield/swollenhippofinal.git
        dest: "/var/www/html"
        version: "test"
endmsg

```

Phase IV: Maintenance

Cron Jobs: Ansible Servers

Type the following command in the CLI:

```
$ crontab -e
```

Hit “i” to edit.

Insert the following lines of code:

```

* * * * * ansible-playbook ~/ansible-playbooks/test-webservers-config.yml
* * * * * ansible-playbook ~/ansible-playbooks/test-web-db-config.yml
0 1 * * * ansible-playbook ~/ansible-playbooks/github-clone.yml

```

To exit, hit esc, then type “:wq!”