

Use Puppet Bolt for agentless execution of tasks and scripts

Table of Contents

1. **Introduction**
 2. **Problem Statement**
 3. **Prerequisites**
 - **Software Requirements**
 - **Hardware Requirements**
 4. **Implementation Steps**
 - **Step 1: Install Puppet Bolt Using Puppetlabs Repository**
 - **Step 2: Run Ad-Hoc Commands with Bolt**
 - **Step 3: Execute Tasks on Remote Nodes**
 - **Step 4: Write and Execute Custom Scripts**
 - **Step 5: Use Plans for Complex Workflows**
 5. **References**
-

Introduction

Puppet Bolt is a versatile, agentless orchestration tool that works well in environments where Puppet agents are not installed. It uses SSH (for Linux) and WinRM (for Windows) to execute tasks and manage configurations directly on nodes.

In this guide, we'll set up Puppet Bolt using the `puppetlabs-release-trusty.deb` package and demonstrate how to use it for task execution and workflows.

Problem Statement

For systems without Puppet agents or for performing ad-hoc operations, managing infrastructure can become cumbersome. Puppet Bolt provides a simple way to manage such environments without requiring a full Puppet setup.

Prerequisites

Completion of all previous lab guides (up to Lab Guide-09) is required before proceeding with Lab Guide-10.

Software Requirements

- Ubuntu 14.04 (Trusty Tahr).
 - Puppet Labs repository set up using `puppetlabs-release-trusty.deb`.
 - SSH access to remote nodes for Linux systems or WinRM for Windows systems.
-

Hardware Requirements

- Control Node (where Puppet Bolt will be installed): 1GB RAM, 1 CPU, 5GB Disk.
- Remote Nodes: Accessible over the network with proper credentials.

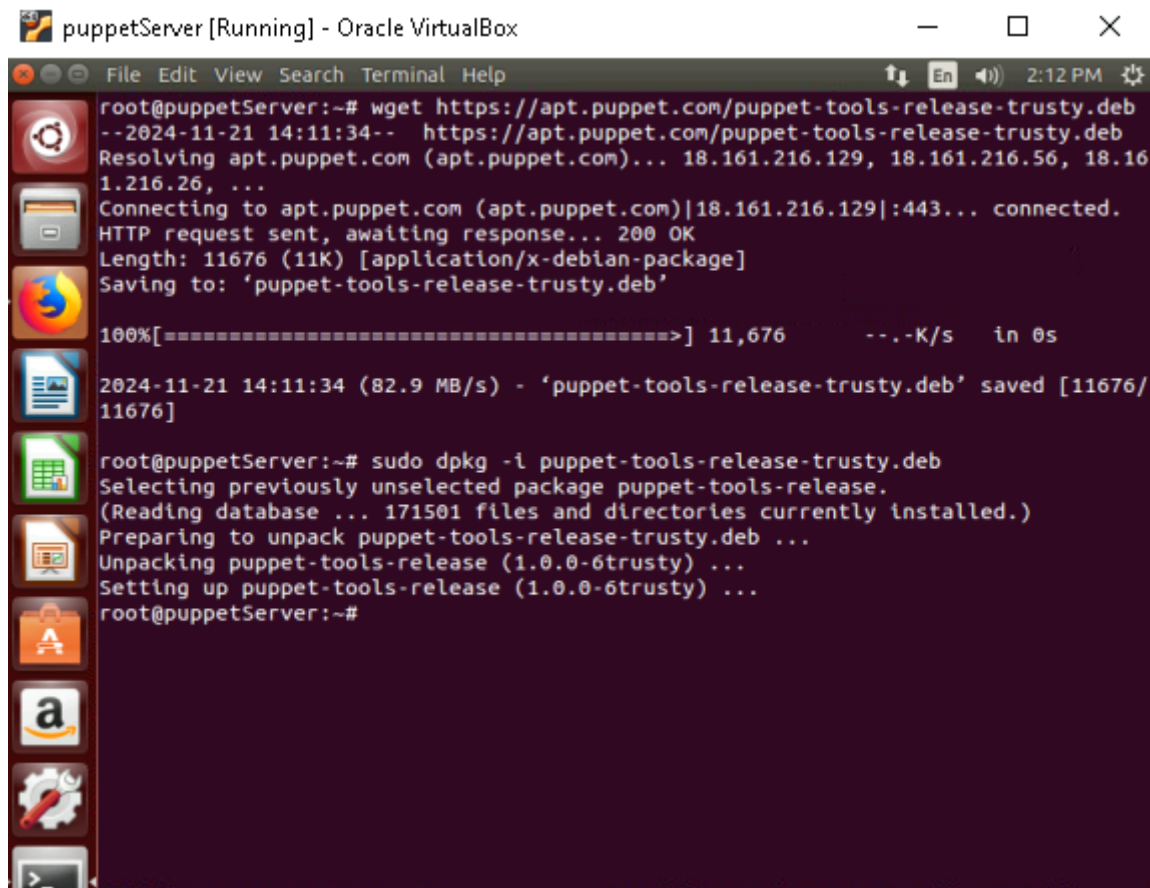
Implementation Steps

Step 1: Install Puppet Bolt Using Puppet Tool Repository

1. Download and Install Puppet Tool Release Package:

If you haven't already installed the Puppet Tool repository, do the following:

```
wget https://apt.puppet.com/puppet-tools-release-trusty.deb
sudo dpkg -i puppet-tools-release-trusty.deb
```



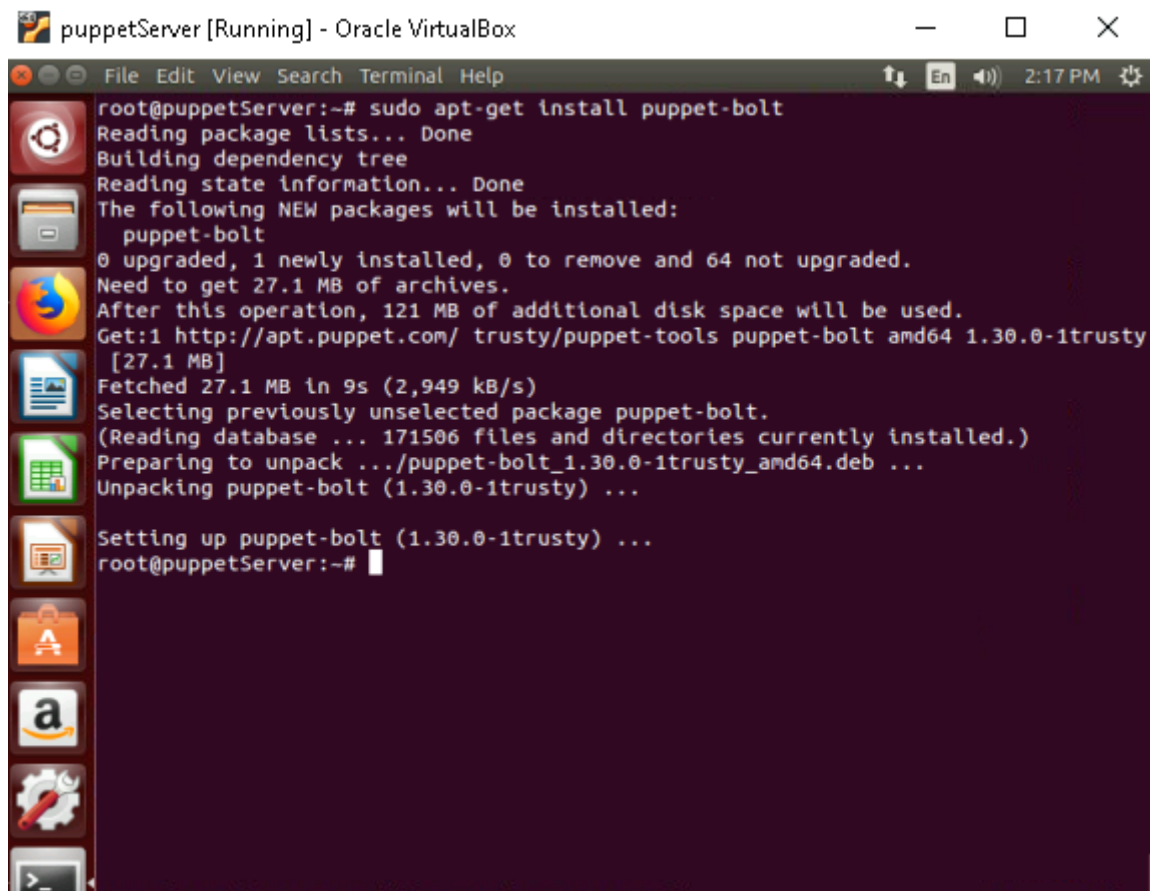
```
puppetServer [Running] - Oracle VirtualBox
File Edit View Search Terminal Help
root@puppetServer:~# wget https://apt.puppet.com/puppet-tools-release-trusty.deb
--2024-11-21 14:11:34-- https://apt.puppet.com/puppet-tools-release-trusty.deb
Resolving apt.puppet.com (apt.puppet.com)... 18.161.216.129, 18.161.216.56, 18.161.216.26, ...
Connecting to apt.puppet.com (apt.puppet.com)|18.161.216.129|:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 11676 (11K) [application/x-debian-package]
Saving to: 'puppet-tools-release-trusty.deb'
100%[=====] 11,676 --.-K/s in 0s
2024-11-21 14:11:34 (82.9 MB/s) - 'puppet-tools-release-trusty.deb' saved [11676/11676]
root@puppetServer:~# sudo dpkg -i puppet-tools-release-trusty.deb
Selecting previously unselected package puppet-tools-release.
(Reading database ... 171501 files and directories currently installed.)
Preparing to unpack puppet-tools-release-trusty.deb ...
Unpacking puppet-tools-release (1.0.0-6trusty) ...
Setting up puppet-tools-release (1.0.0-6trusty) ...
root@puppetServer:~#
```

2. Update the APT Cache:

```
sudo apt-get update
```

3. Install Puppet Bolt:

```
sudo apt-get install puppet-bolt
```

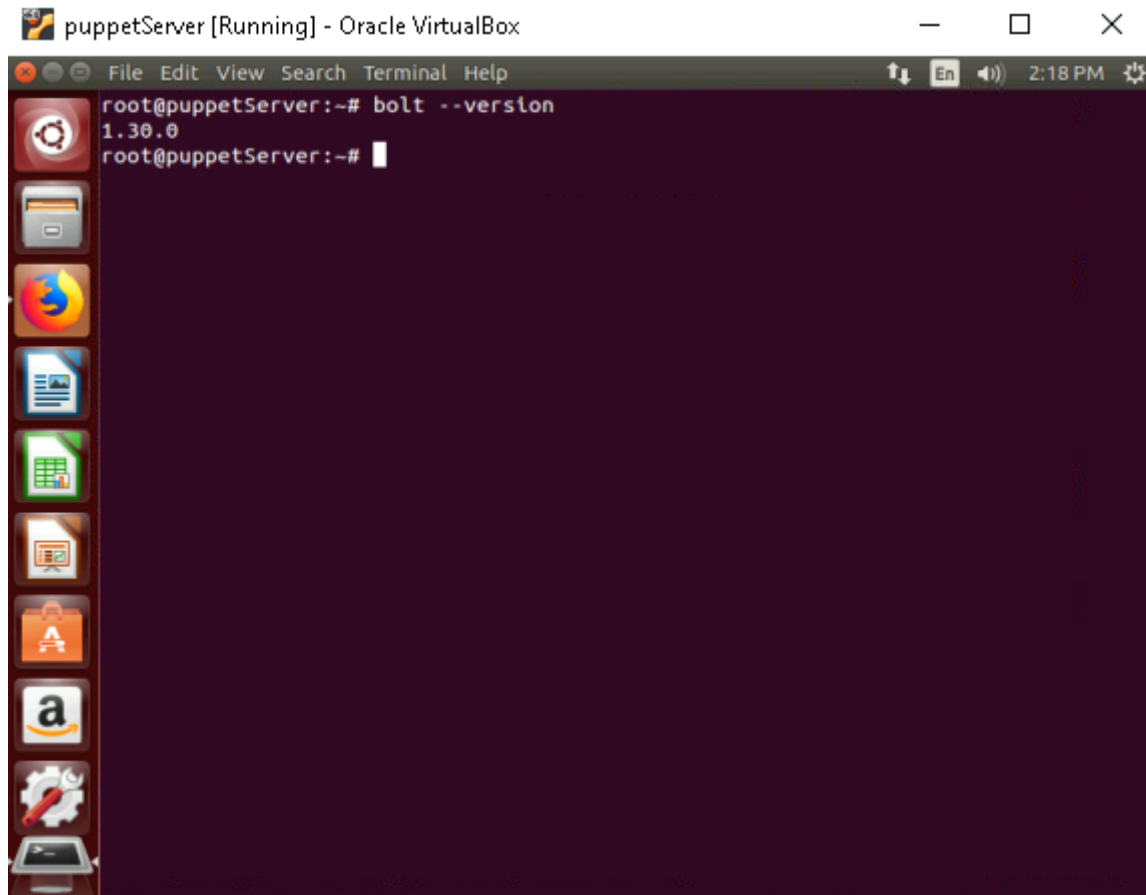


```
puppetServer [Running] - Oracle VirtualBox
File Edit View Search Terminal Help
root@puppetServer:~# sudo apt-get install puppet-bolt
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following NEW packages will be installed:
  puppet-bolt
0 upgraded, 1 newly installed, 0 to remove and 64 not upgraded.
Need to get 27.1 MB of archives.
After this operation, 121 MB of additional disk space will be used.
Get:1 http://apt.puppet.com/ trusty/puppet-tools puppet-bolt amd64 1.30.0-1trusty
  [27.1 MB]
Fetched 27.1 MB in 9s (2,949 kB/s)
Selecting previously unselected package puppet-bolt.
(Reading database ... 171506 files and directories currently installed.)
Preparing to unpack ../puppet-bolt_1.30.0-1trusty_amd64.deb ...
Unpacking puppet-bolt (1.30.0-1trusty) ...
Setting up puppet-bolt (1.30.0-1trusty) ...
root@puppetServer:~#
```

4. Verify Installation:

Confirm that Bolt is installed correctly:

```
bolt --version
```



- **Expected Output:** Displays the installed version of Bolt.

Step 2: Run Ad-Hoc Commands with Bolt

1. Run a Command on a Remote Node:

Use Bolt to execute a simple command (e.g., checking uptime) on a remote node via SSH:

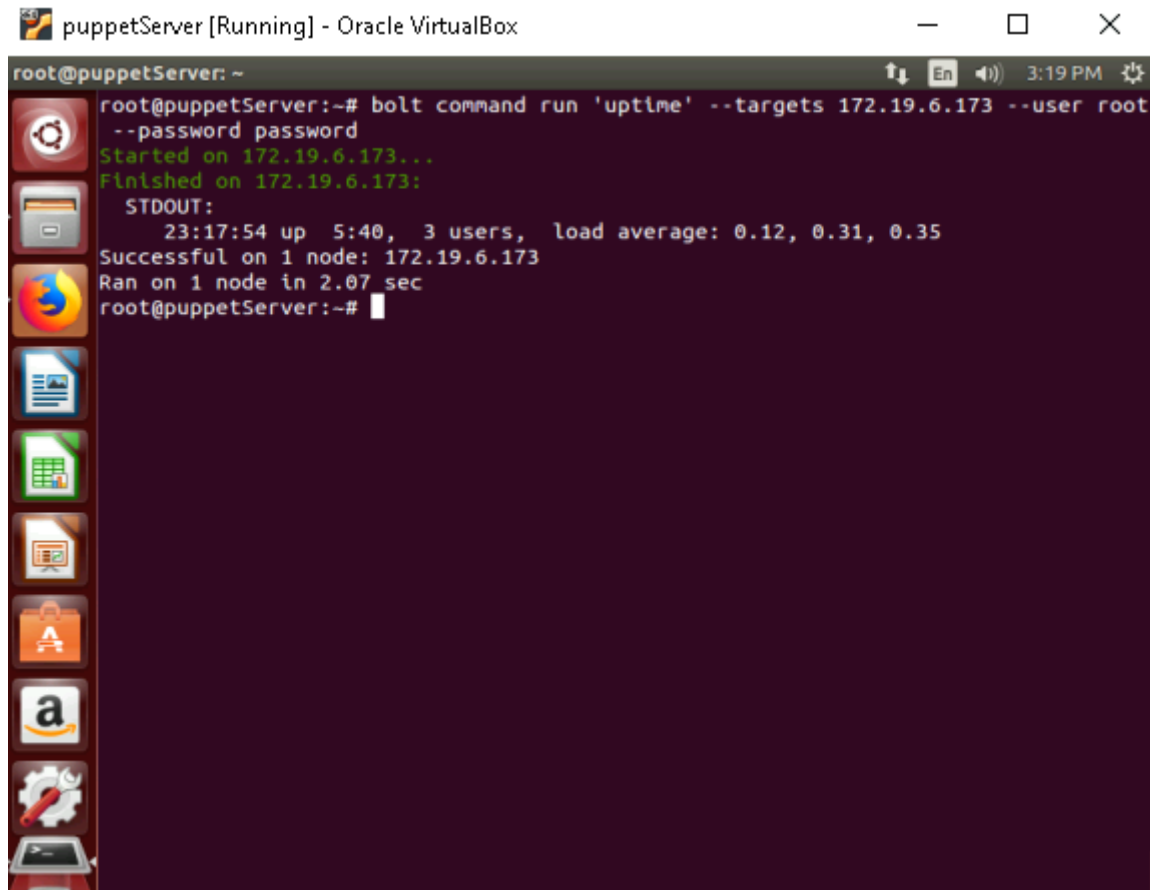
```
bolt command run 'uptime' --targets <TARGET> --user <USERNAME> --password <PASSWORD>
```

Replace:

- **<TARGET>**: IP or hostname of the target node.
- **<USERNAME>**: Username with SSH access.
- **<PASSWORD>**: Password for the username.

2. Example:

```
bolt command run 'uptime' --targets 172.19.6.173 --user root --password password
```



```
root@puppetServer:~# bolt command run 'uptime' --targets 172.19.6.173 --user root --password password
Started on 172.19.6.173...
Finished on 172.19.6.173:
  STDOUT:
    23:17:54 up 5:40, 3 users, load average: 0.12, 0.31, 0.35
Successful on 1 node: 172.19.6.173
Ran on 1 node in 2.07 sec
root@puppetServer:~#
```

Note: Make sure that you have SSH access to the target node.

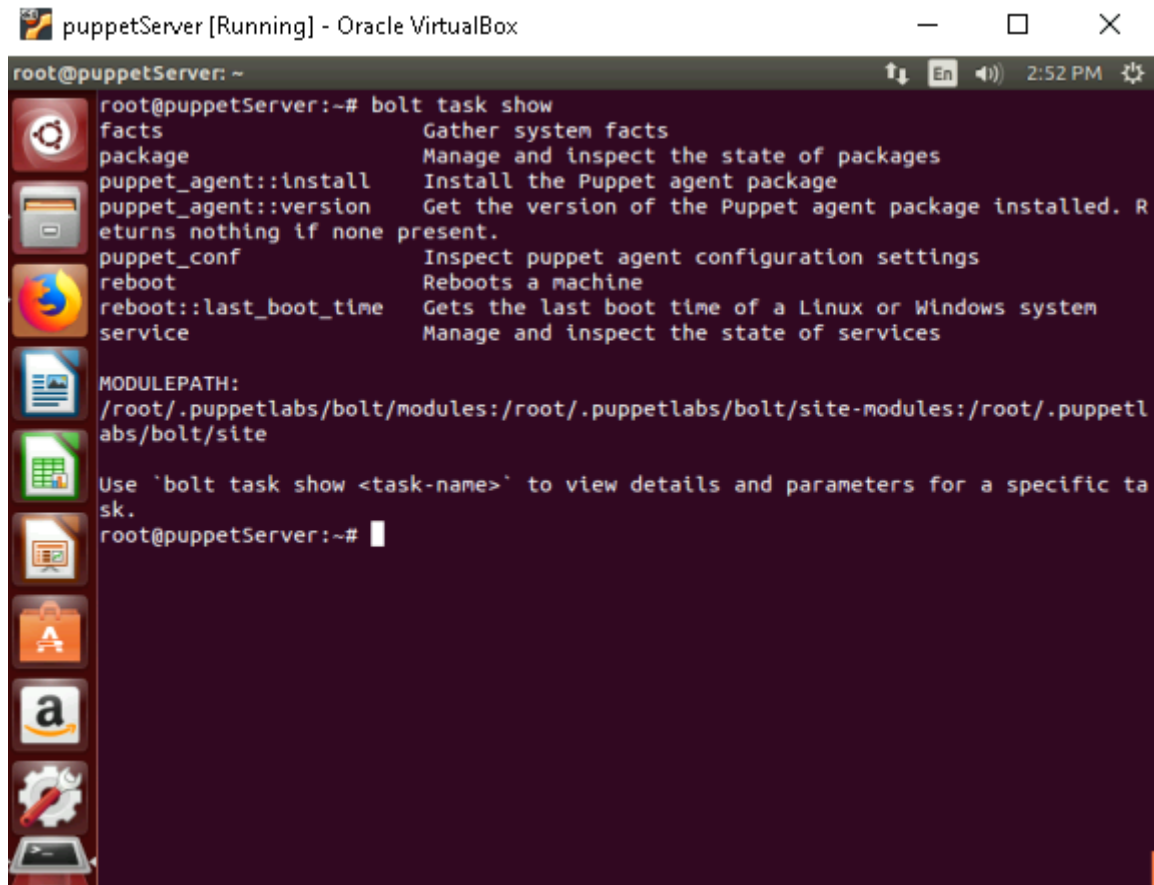
if not installed, install the `openssh-server` package on the target node and enable the `permitrootlogin` in the `/etc/ssh/sshd_config` file(`permitrootlogin yes`) and restart the ssh service.

Step 3: Execute Tasks on Remote Nodes

1. List Available Tasks:

Puppet Bolt includes tasks from Puppet Forge. To list available tasks:

```
bolt task show
```

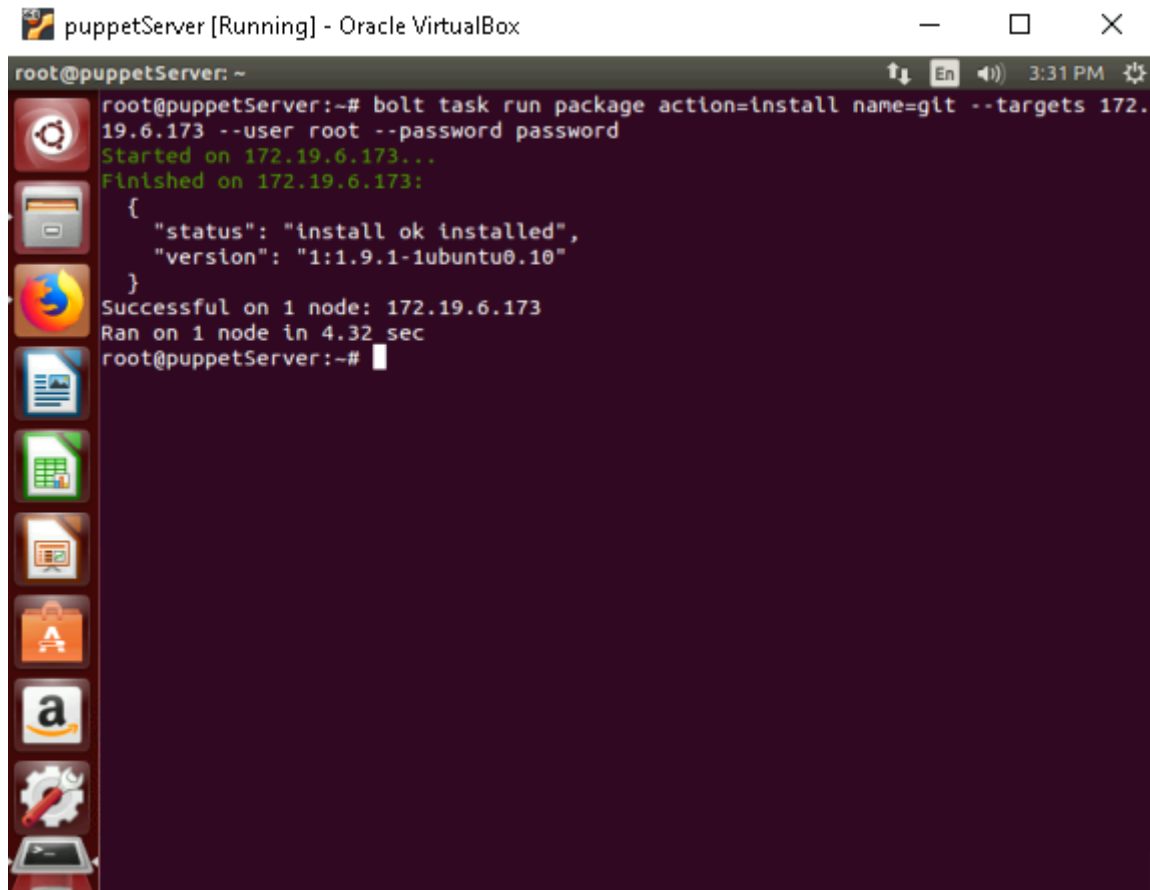


```
root@puppetServer: ~  
root@puppetServer:~# bolt task show  
facts                Gather system facts  
package              Manage and inspect the state of packages  
puppet_agent::install Install the Puppet agent package  
puppet_agent::version Get the version of the Puppet agent package installed. Returns nothing if none present.  
puppet_conf          Inspect puppet agent configuration settings  
reboot               Reboots a machine  
reboot::last_boot_time Gets the last boot time of a Linux or Windows system  
service              Manage and inspect the state of services  
  
MODULEPATH:  
/root/.puppetlabs/bolt/modules:/root/.puppetlabs/bolt/site-modules:/root/.puppetlabs/bolt/site  
  
Use 'bolt task show <task-name>' to view details and parameters for a specific task.  
root@puppetServer:~#
```

2. Run a Predefined Task:

Install a package on a remote node:

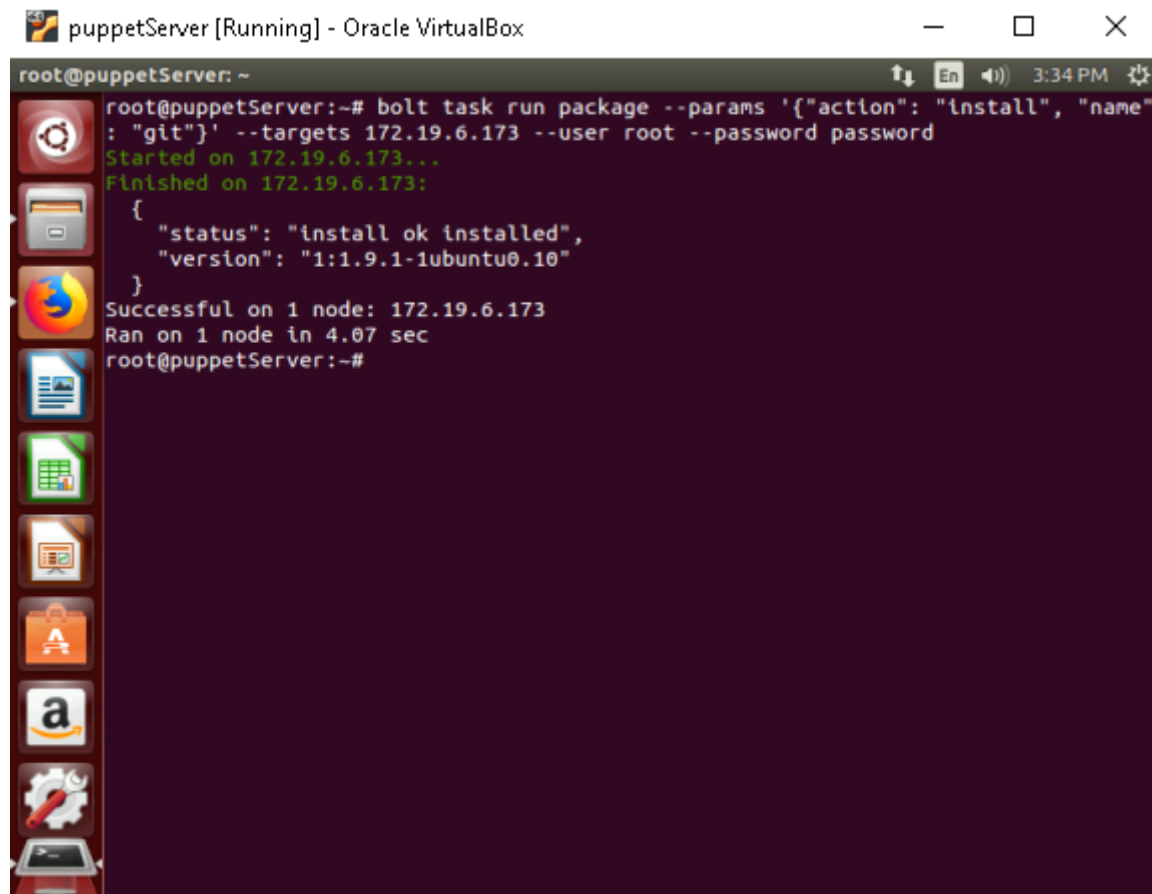
```
bolt task run package action=install name=git --targets 172.19.6.173 --user root --password password
```



```
root@puppetServer: ~  
root@puppetServer:~# bolt task run package action=install name=git --targets 172.19.6.173 --user root --password password  
Started on 172.19.6.173...  
Finished on 172.19.6.173:  
{  
  "status": "install ok installed",  
  "version": "1:1.9.1-1ubuntu0.10"  
}  
Successful on 1 node: 172.19.6.173  
Ran on 1 node in 4.32 sec  
root@puppetServer:~#
```

3. Run a Task with JSON Parameters:

```
bolt task run package --params '{"action": "install", "name": "git"}' --  
targets 172.19.6.173 --user root --password password
```



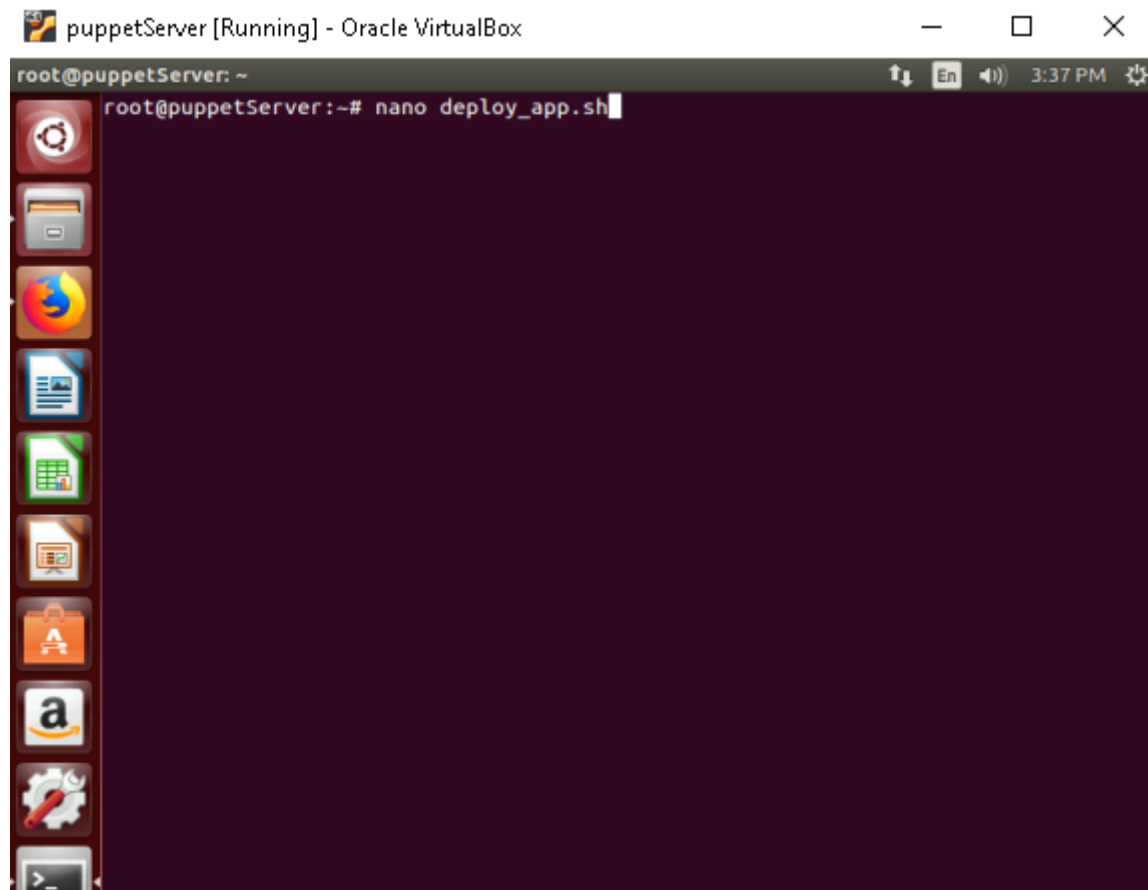
```
root@puppetServer: ~  
root@puppetServer:~# bolt task run package --params '{"action": "install", "name": "git"}' --targets 172.19.6.173 --user root --password password  
Started on 172.19.6.173...  
Finished on 172.19.6.173:  
{  
  "status": "install ok installed",  
  "version": "1:1.9.1-1ubuntu0.10"  
}  
Successful on 1 node: 172.19.6.173  
Ran on 1 node in 4.07 sec  
root@puppetServer:~#
```

Step 4: Write and Execute Custom Scripts

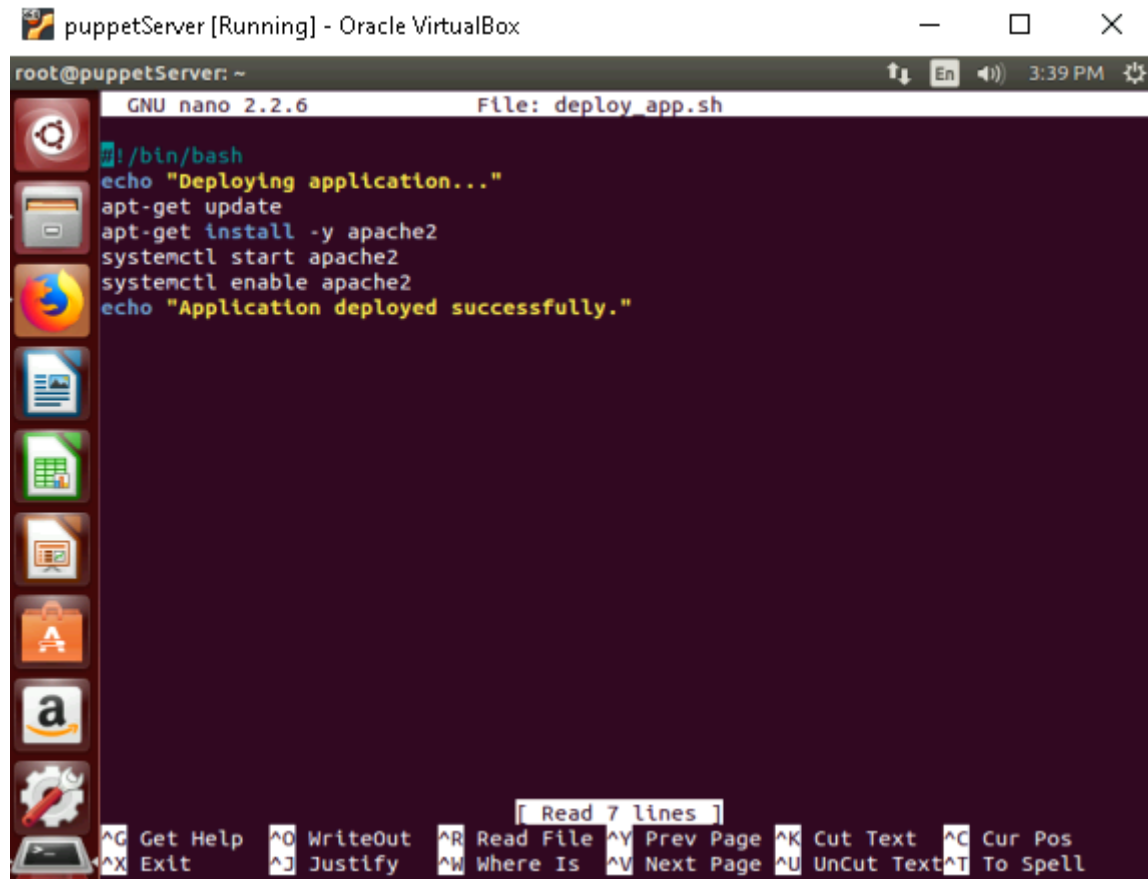
1. Create a Script:

Write a custom script (e.g., `deploy_app.sh`):

```
nano deploy_app.sh
```

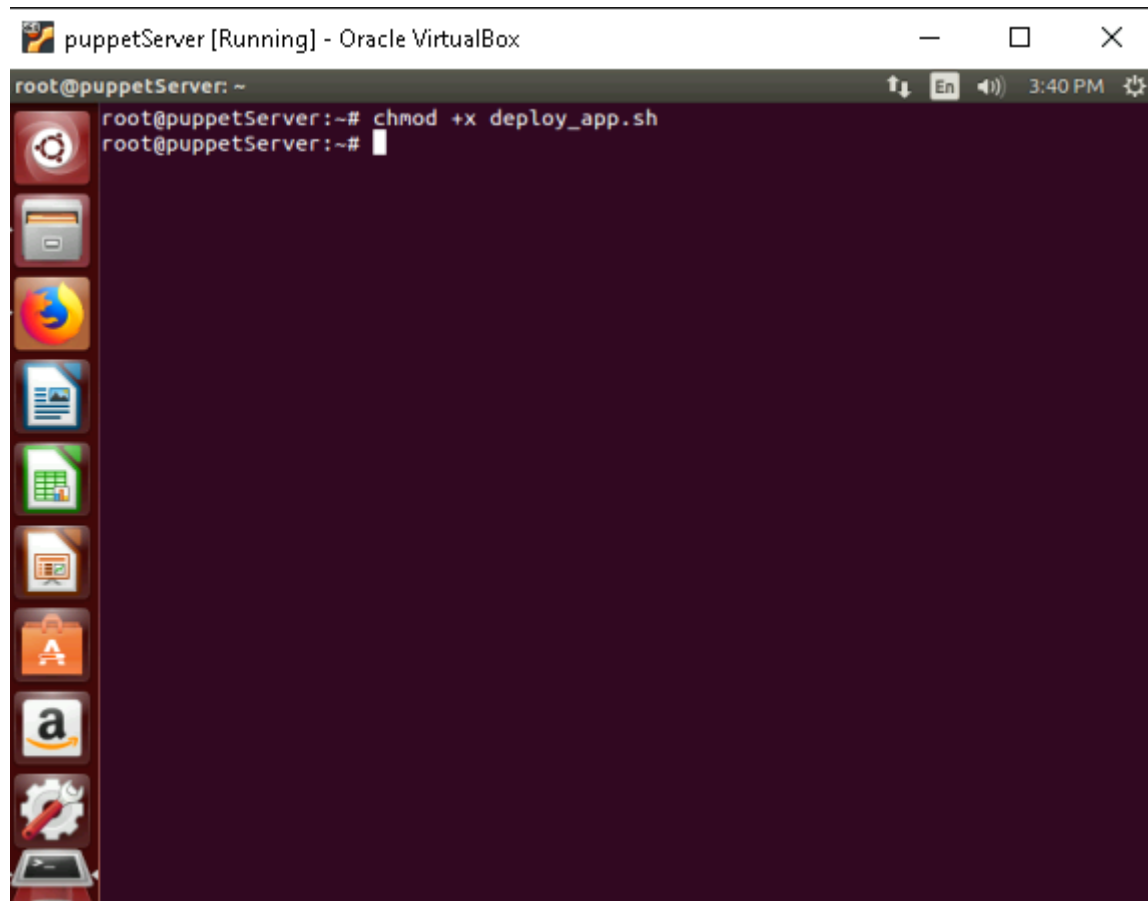
```
#!/bin/bash
echo "Deploying application..."
apt-get update
apt-get install -y apache2
systemctl start apache2
systemctl enable apache2
echo "Application deployed successfully."
```



```
puppetServer [Running] - Oracle VM VirtualBox
root@puppetServer: ~
GNU nano 2.2.6 File: deploy_app.sh
#!/bin/bash
echo "Deploying application..."
apt-get update
apt-get install -y apache2
systemctl start apache2
systemctl enable apache2
echo "Application deployed successfully."
```

2. Make the Script Executable:

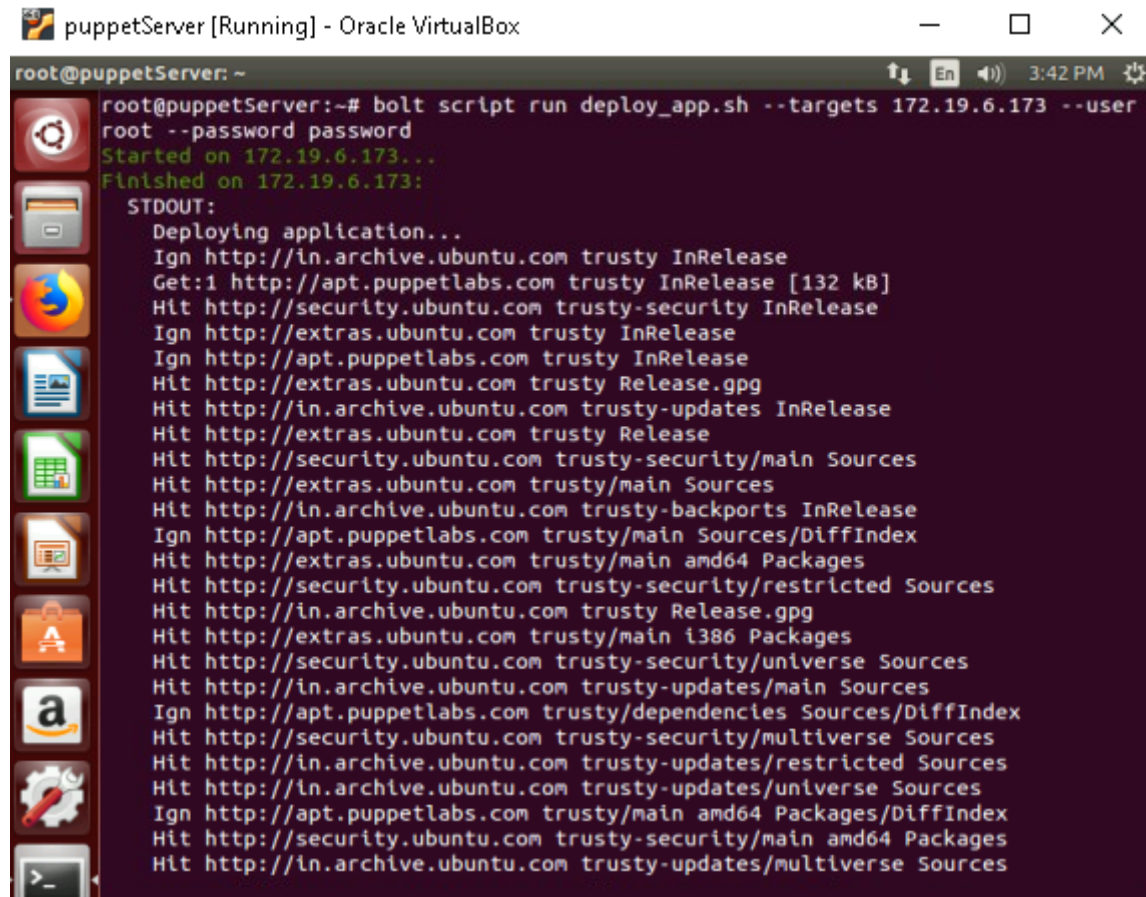
```
chmod +x deploy_app.sh
```



3. Run the Script on Remote Nodes:

Execute the script using Bolt:

```
bolt script run ./deploy_app.sh --targets 172.19.6.173 --user root --  
password password
```



```
root@puppetServer: ~  
root@puppetServer:~# bolt script run deploy_app.sh --targets 172.19.6.173 --user  
root --password password  
Started on 172.19.6.173...  
Finished on 172.19.6.173:  
STDOUT:  
Deploying application...  
Ign http://in.archive.ubuntu.com trusty InRelease  
Get:1 http://apt.puppetlabs.com trusty InRelease [132 kB]  
Hit http://security.ubuntu.com trusty-security InRelease  
Ign http://extras.ubuntu.com trusty InRelease  
Ign http://apt.puppetlabs.com trusty InRelease  
Hit http://extras.ubuntu.com trusty Release.gpg  
Hit http://in.archive.ubuntu.com trusty-updates InRelease  
Hit http://extras.ubuntu.com trusty Release  
Hit http://security.ubuntu.com trusty-security/main Sources  
Hit http://extras.ubuntu.com trusty/main Sources  
Hit http://in.archive.ubuntu.com trusty-backports InRelease  
Ign http://apt.puppetlabs.com trusty/main Sources/DiffIndex  
Hit http://extras.ubuntu.com trusty/main amd64 Packages  
Hit http://security.ubuntu.com trusty-security/restricted Sources  
Hit http://in.archive.ubuntu.com trusty Release.gpg  
Hit http://extras.ubuntu.com trusty/main i386 Packages  
Hit http://security.ubuntu.com trusty-security/universe Sources  
Hit http://in.archive.ubuntu.com trusty-updates/main Sources  
Ign http://apt.puppetlabs.com trusty/dependencies Sources/DiffIndex  
Hit http://security.ubuntu.com trusty-security/multiverse Sources  
Hit http://in.archive.ubuntu.com trusty-updates/restricted Sources  
Hit http://in.archive.ubuntu.com trusty-updates/universe Sources  
Ign http://apt.puppetlabs.com trusty/main amd64 Packages/DiffIndex  
Hit http://security.ubuntu.com trusty-security/main amd64 Packages  
Hit http://in.archive.ubuntu.com trusty-updates/multiverse Sources
```

References

- Puppet Labs Official Repository: [Puppet APT Repositories](#)
 - Puppet Bolt Documentation: [Puppet Bolt Docs](#)
-