



Systems And Network Administration

Topic: Puppet

Group members

Muhammad Faseeh (2012390)

Nabeel Hussain (2012398)

Shayan Hussain (2012407)

Section B

Submitted to

Sir Owaiz Kazmi

Table of Contents

Introduction	3
Who made puppet and who supports it?	3
Which platforms can be managed by using Puppet?	4
What are the advantages of puppet over other tools?	4
History	4
Written in Which language?	5
Why do we use Puppet?	5
How puppet works?	5
Advantages and Disadvantages	6-7
Advantages of Puppet	6
Disadvantages of Puppet	7
Use Cases of Puppet:	8
• Automation of Routine Tasks:	8
• Security and Compliance Management:	8
• Configuration Management:	8
Who use Puppet	8-9
• Enterprise IT Organizations:	8
• Technology Companies:	8
• Financial Institutions:	9
Architecture of puppet:	9-10
Puppet Master:	9
Puppet Master-Slave Communication:	10
Configuration and integration of puppet:	11-15
References:	16

Introduction

In the realm of system administration, the term "puppet" takes on a different meaning, referring to a powerful and widely used configuration management tool. Puppet, in this context, is not a physical object but rather a software tool designed to automate the management and configuration of computer systems within a network.

Puppet is an open-source configuration management system that enables system administrators to define and manage the desired state of their IT infrastructure. It automates the deployment and configuration of software, ensuring consistency and reliability across multiple servers or systems. Puppet is particularly valuable in large-scale, complex IT environments where maintaining uniformity and managing configurations manually can be time-consuming and error-prone.



Who made puppet and who supports it?

Puppet is made by Luke Kanies. It is based on ruby language. Currently puppet is supported by Puppet Labs (Luke Kanies is the CEO of Puppet Labs). GPLv2 is used to license puppet.



Which platforms can be managed by using Puppet?

Puppet can be used to manage Unix and most of the Linux flavors. Even Windows platforms can be handled using puppet. Some of the platforms which is supported by puppet are mentioned below

- HP UX
- BSD
- MACOS X server
- Gentoo
- Debian
- Red Hat/Centos, Fedora
- Mandriva

What are the advantages of puppet over other tools?

- Normally most of the configuration management tool, deploy the required configuration on a machine, and leave them as it is. But puppet keeps on verifying the configuration at a specified interval (which you can modify as per requirement).
- Puppet defines the configurations for a host with the help of a language which is very easy to learn and is only used for that purpose.
- Puppet is used by major players in the industry like Google, red hat etc.
- Larger open-source developer base
- Wide number of platforms are supported in puppet.
- Its works very smooth, even when deployed in a large infrastructure (thousands of hosts to manage)

History

Puppet was started by creator and founder Luke Kanies, who was working as a sysadmin and consultant. He was unable to find the tooling he wanted to use and that his customers could rely on, so he created Puppet as a Ruby-based open-source configuration management language in 2005. The success of this open-source project resulted in the release of a commercial offering, Puppet Enterprise, in February 2011. But as the demands increased and Puppet needed to reform and expand as both a company and an open-source project.

The new leadership that followed took a direction that saw the company develop its professional services, and focus more effort on developer tooling and education while expanding its product range both organically and via acquisitions, striking a difficult balance between the open-source community and its enterprise customer demands. Puppet was acquired by Perforce Software on

May 17, 2022, following the Chef (2020) and Ansible (2015) acquisitions, as the last of the standalone configuration management start-ups. Luke summed up the change that has taken place in the industry: DevOps teams are different now. Companies are looking for a complete solution, rather than wanting to integrate individual best-of-breed vendors.

Written in Which language?

Puppet itself is written in Ruby, while Factor is written in C++.



Why do we use Puppet?

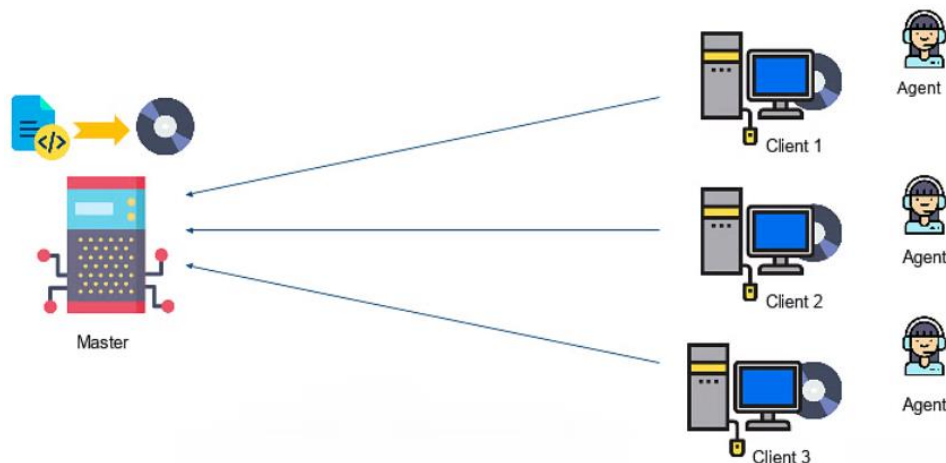
Most reason that companies adopt puppet is because a lot other companies have also adopted puppet and basically there is tried and tested tool which every companies it companies it configuration management needs.

There are following reasons to explain why we use.

- Puppet provides more support for the platform.
- Debian / Ubuntu
- Microsoft Windows
- Redhat, Centos, Fedora
- MACOS X server
- Puppet has better documentation
- Unlike other configuration management tools

How puppet works?

Puppet has a primary-secondary node architecture.



The clients are distributed across the network and communicate with the primary-secondary environment where Puppet modules are present. The client agent sends a certificate with its ID to the server; the server then signs that certificate and sends it back to the client. This authentication allows for secure and verifiable communication between the client and the master.

The factor then collects the state of the clients and sends it to the master. Based on the fact sent, the master compiles the manifests into the catalogs, which are sent to the clients, and an agent executes the manifests on its machine. A report is generated by the client that describes any changes made and is sent to the master.

This process is repeated at regular intervals, ensuring all client systems are up to date. In the next section, let us find out about the various companies adopting Puppet as a part of our learning about what is Puppet.

Advantages and Disadvantages

There is no perfect software program. Every application has advantages and disadvantages, making it suitable for a particular task. It's essential to be aware of these pros and cons before deciding which software to use. Here are some advantages and disadvantages of Puppet.

Advantages of Puppet:

Automation:

When it comes to automation, there are many benefits of using Puppet. One great feature is that you can automate the operating system on remote machines, which saves engineers from manually doing this every time they want a new version or update released in their field. Instead, these tasks are done automatically by adding more advanced scripts with little human intervention necessary.

Such an approach might become dangerous if not implemented correctly, especially considering each specialist writes code according to their style, potentially causing compatibility issues.

Deployment Speed:

The deployment speed of an automated system effectively relieves the network engineer from busy schedules and frees them up to focus on other tasks. Before automation, days or weeks would go by with no progress because every change had its set process, which could take hours depending upon how complicated things got. Automation speeds up deployment times, allowing time for solving other tasks like software development or working with clients.

Support:

Puppet is relatively easy to use. However, sometimes there are problems, and Puppet has technical support whose specialists can save you time and money. Getting help from a professional prevents costly mistakes and fixes your issues quickly. If you're looking for someone to help you with your technology needs, check out technical support.

Disadvantages of Puppet:

Puppet Server Vulnerability:

Centralized network management also has its downsides. Server security should come first. If attackers gain access to the Puppet server, they can configure the client operating systems as they please, like giving commands to encrypt or delete data. In this case, the losses are incalculable.

While these threats exist with any software, the solution is to install modern mechanisms to protect the server from external threats, guided primarily by understanding basic information security principles.

Administrator Qualification:

Maintaining large networks requires the administrator to be highly qualified and attentive. For example, if you make a mistake when compiling a configuration file and do not check it properly, you can negatively impact several servers at once.

Of course, rolling back the system to the previous configuration restores its operation, but any time loss could bring losses to the company. In addition,

ensuring your network specialist is adequately trained and qualified may come at a high cost.

Use Cases of Puppet:

Puppet is designed to automate the management and configuration of IT infrastructure. It is widely used in various use cases to streamline operations, ensure consistency, and enhance efficiency, following are some use cases.

- **Automation of Routine Tasks:**

Puppet automates repetitive tasks, saving time and human error. This includes tasks like software installations, updates, and system configurations.

- **Security and Compliance Management:**

Puppet plays a critical role in enforcing security policies and ensuring compliance with industry regulations. It helps in identifying security vulnerabilities, making it an essential tool for organizations that prioritize data security.

- **Configuration Management:**

Puppet is primarily used for automating and managing the configurations of IT infrastructure. It ensures that the settings and software on servers and other devices are consistent and in the desired state.

Who use Puppet:

Puppet is widely used across various industries and by organizations of different sizes.

- **Enterprise IT Organizations:**

Large enterprises with complex IT infrastructures use Puppet to manage and automate the configurations of servers, networking devices, and other components. It helps ensure consistency and reliability across a large number of systems.

- **Technology Companies:**

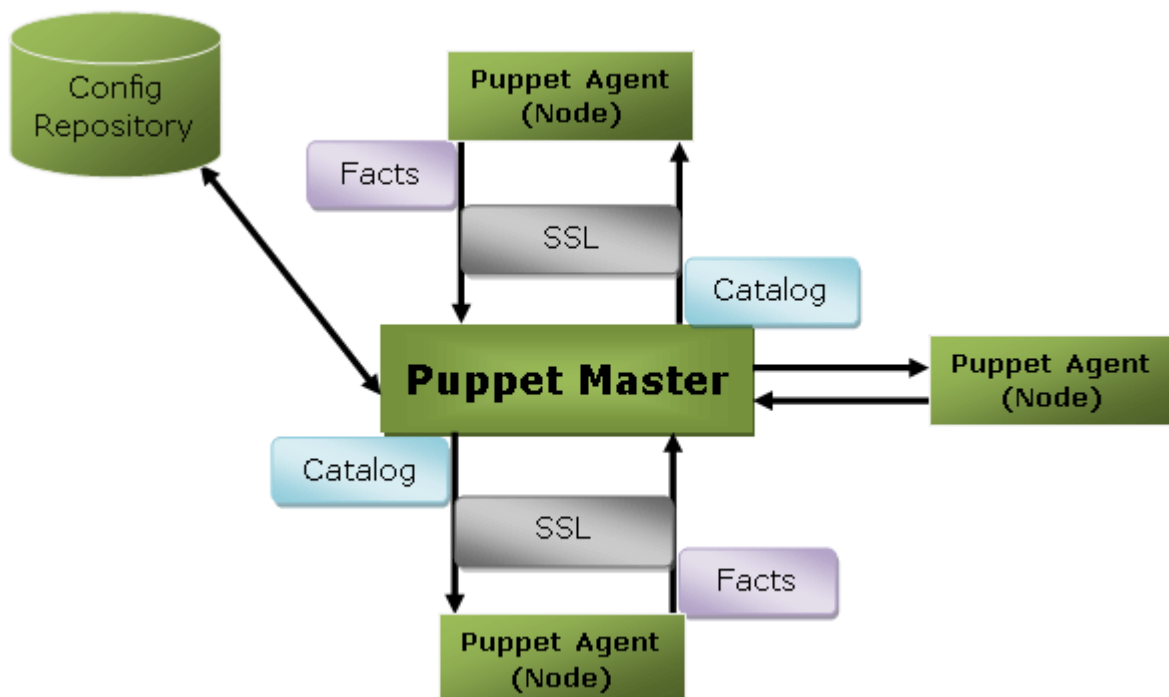
Technology companies, including both startups and established firms, use Puppet for automating the deployment and configuration of their software applications. This is common in environments that utilize DevOps practices.

- **Financial Institutions:**

Financial services organizations, such as banks and insurance companies, use Puppet to manage and secure their IT infrastructure.

Architecture of puppet:

Puppet uses master-slave or client-server architecture. Puppet client and server interconnected by SSL, which is a secure socket layer. It is a model-driven system.



Puppet Master:

Puppet master handles all the configuration related process in the form of puppet codes. It is a Linux based system in which puppet master software is installed. The puppet master must be in Linux. It uses the puppet agent to apply the configuration to nodes.

Puppet Master-Slave Communication:

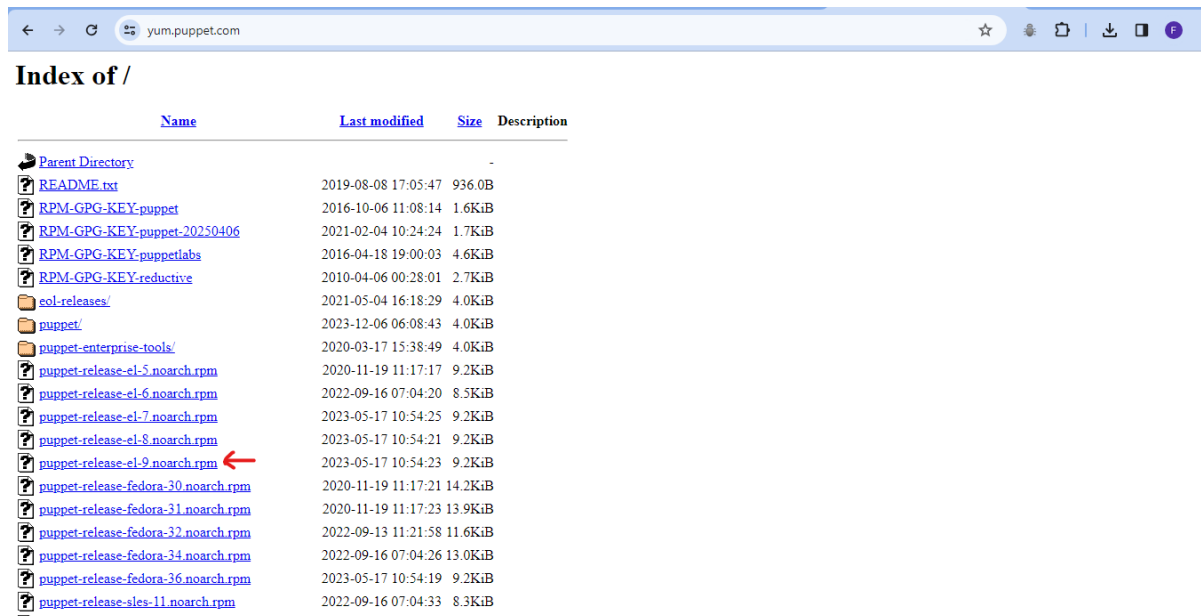
Puppet master-slave communicates via a secure encrypted channel through the SSL (Secure Socket Layer).



- Puppet slave requests for Puppet Master Certificate.
- Puppet master sends the Master Certificate to the puppet slave in response to the client request.
- Puppet master requests to the Puppet slave for the slave certificate.
- Puppet slave sends the requested slave certificate to the puppet master.
- Puppet slave sends a request for data to the puppet master.
- The master sends the data to the puppet slave as per the request.

Configuration and Integration of Puppet:

For configuration and integration of both puppet-server and puppet-agent in our master and client sever we have to select the version of Puppet for which I have used the el-9 version from yum.



Name	Last modified	Size	Description
Parent Directory	-	-	-
README.txt	2019-08-08 17:05:47	936.0B	
RPM-GPG-KEY-puppet	2016-10-06 11:08:14	1.6KiB	
RPM-GPG-KEY-puppet-20250406	2021-02-04 10:24:24	1.7KiB	
RPM-GPG-KEY-puppetlabs	2016-04-18 19:00:03	4.6KiB	
RPM-GPG-KEY-reductive	2010-04-06 00:28:01	2.7KiB	
eol-releases/	2021-05-04 16:18:29	4.0KiB	
puppet/	2023-12-06 06:08:43	4.0KiB	
puppet-enterprise-tools/	2020-03-17 15:38:49	4.0KiB	
puppet-release-el-5.noarch.rpm	2020-11-19 11:17:17	9.2KiB	
puppet-release-el-6.noarch.rpm	2022-09-16 07:04:20	8.5KiB	
puppet-release-el-7.noarch.rpm	2023-05-17 10:54:25	9.2KiB	
puppet-release-el-8.noarch.rpm	2023-05-17 10:54:21	9.2KiB	
puppet-release-el-9.noarch.rpm	2023-05-17 10:54:23	9.2KiB	
puppet-release-fedora-30.noarch.rpm	2020-11-19 11:17:21	14.2KiB	
puppet-release-fedora-31.noarch.rpm	2020-11-19 11:17:23	13.9KiB	
puppet-release-fedora-32.noarch.rpm	2022-09-13 11:21:58	11.6KiB	
puppet-release-fedora-34.noarch.rpm	2022-09-16 07:04:26	13.0KiB	
puppet-release-fedora-36.noarch.rpm	2023-05-17 10:54:19	9.2KiB	
puppet-release-sles-11.noarch.rpm	2022-09-16 07:04:33	8.3KiB	

First, I will be downloading puppet package from yum package manager in my master server.

```
[root@puppet ~]# rpm -Uvh http://yum.puppetlabs.com/puppet-release-el-9.noarch.rpm
Retrieving http://yum.puppetlabs.com/puppet-release-el-9.noarch.rpm
warning: /var/tmp/rpm-tmp.6dm5uv: Header V4 RSA/SHA512 Signature, key ID 9e61ef26: NOKEY
Verifying...
Preparing...
Updating / installing...
 1:puppet-release-1.0.0-28.el9
Cleaning up / removing...
 2:puppet-release-1.0.0-28.el7
```

Then after the package is downloaded, I will then install the puppet-server.

```
[root@puppet ~]# yum -y install puppetserver
Puppet Repository el 9 - x86_64
Last metadata expiration check: 0:01:55 ago on Thu 07 Dec 2023 11:28:26 PM PKT.
Dependencies resolved.
-----
```

Package	Architecture	Version	Repository	Size
Installing:				
puppetserver	noarch	7.14.0-1.el9	puppet	72 M
Installing dependencies:				
copy-jdk-configs	noarch	4.0-3.el9	appstream	28 k
java-1.8.0-openjdk-headless	x86_64	1:1.8.0.362.b09-4.el9	appstream	32 M
javapackagesfilesystem	noarch	6.0.0-4.el9	appstream	13 k
lkctp-tools	x86_64	1.0.19-2.el9	baseos	94 k
lua	x86_64	5.4.4-4.el9	appstream	186 k
lua-posix	x86_64	35.0-8.el9	appstream	151 k
puppet-agent	x86_64	7.27.0-1.el9	puppet	21 M
tzdata-java	noarch	2023c-1.el9	appstream	230 k

```
-----
Transaction Summary
-----
Install 9 Packages

Total download size: 126 M
Installed size: 206 M
Downloading Packages:
(1/9): lkctp-tools-1.0.19-2.el9.x86_64.rpm
(2/9): copy-jdk-configs-4.0-3.el9.noarch.rpm
(3/9): lua-5.4.4-4.el9.x86_64.rpm
(4/9): lua-posix-35.0-8.el9.x86_64.rpm
(5/9): javapackagesfilesystem-6.0.0-4.el9.noarch.rpm
(6/9): tzdata-java-2023c-1.el9.noarch.rpm
(7/9): puppet-agent-7.27.0-1.el9.x86_64.rpm
(8/9): puppetserver-7.14.0-1.el9.noarch.rpm
(9/9): java-1.8.0-openjdk-headless-1.8.0.362.b09-4.el9.x86_64.rpm
-----
```

Total	1.2 MB/s 126 MB	01:45
Puppet Repository el 9 - x86_64	1.6 MB/s 1.7 kB	00:00

```
Importing GPG key 0x9E61EF26:
  Userid : "Puppet, Inc. Release Key (Puppet, Inc. Release Key) <release@puppet.com>"
  Fingerprint: D681 1ED3 ADEE B944 1AF5 AASF 4528 B6CD 9E61 EF26
  From : /etc/pki/rpm-gpg/RPM-GPG-KEY-2025-04-06-puppet-release
-----
```

```
(9/9): java-1.8.0-openjdk-headless-1.8.0.362.b09-4.el9.x86_64.rpm
-----
```

Total	1.2 MB/s 126 MB	01:45
Puppet Repository el 9 - x86_64	1.6 MB/s 1.7 kB	00:00

```
Importing GPG key 0x9E61EF26:
  Userid : "Puppet, Inc. Release Key (Puppet, Inc. Release Key) <release@puppet.com>"
  Fingerprint: D681 1ED3 ADEE B944 1AF5 AASF 4528 B6CD 9E61 EF26
  From : /etc/pki/rpm-gpg/RPM-GPG-KEY-2025-04-06-puppet-release
Key imported successfully
Puppet Repository el 9 - x86_64
Importing GPG key 0xF8D349F:
  Userid : "Puppet, Inc. Release Key (Puppet, Inc. Release Key) <release@puppet.com>"
  Fingerprint: 6F6B 1550 9CF8 E59E 6E46 9F32 7F43 8280 EF8D 349F
  From : /etc/pki/rpm-gpg/RPM-GPG-KEY-puppet-release
Key imported successfully
Running transaction check
Transaction check succeeded.
Running transaction test
Transaction test succeeded.
Running transaction
Running scriptlet: copy-jdk-configs-4.0-3.el9.noarch
Running scriptlet: java-1.8.0-openjdk-headless-1:1.8.0.362.b09-4.el9.x86_64
Preparing :
Installing : tzdata-java-2023c-1.el9.noarch
Running scriptlet: puppet-agent-7.27.0-1.el9.x86_64
Installing : puppet-agent-7.27.0-1.el9.x86_64
Running scriptlet: puppet-agent-7.27.0-1.el9.x86_64
Installing : lua-posix-35.0-8.el9.x86_64
Installing : lua-5.4.4-4.el9.x86_64
Installing : copy-jdk-configs-4.0-3.el9.noarch
Installing : javapackagesfilesystem-6.0.0-4.el9.noarch
Installing : lkctp-tools-1.0.19-2.el9.x86_64
Installing : java-1.8.0-openjdk-headless-1:1.8.0.362.b09-4.el9.x86_64
Running scriptlet: java-1.8.0-openjdk-headless-1:1.8.0.362.b09-4.el9.x86_64
Running scriptlet: puppetserver-7.14.0-1.el9.noarch
Installing : puppetserver-7.14.0-1.el9.noarch
Running scriptlet: puppetserver-7.14.0-1.el9.noarch
usermod: no changes

Running scriptlet: puppet-agent-7.27.0-1.el9.x86_64
Running scriptlet: copy-jdk-configs-4.0-3.el9.noarch
Running scriptlet: java-1.8.0-openjdk-headless-1:1.8.0.362.b09-4.el9.x86_64
Running scriptlet: puppetserver-7.14.0-1.el9.noarch
Verifying : lkctp-tools-1.0.19-2.el9.x86_64
```

Then I will change my master server hostname to puppet

```
[student@szabist ~]$ hostnamectl set-hostname puppet
==== AUTHENTICATING FOR org.freedesktop.hostname1.set-static-hostname ====
Authentication is required to set the statically configured local hostname, as w
ell as the pretty hostname.
Authenticating as: root
Password:
==== AUTHENTICATION COMPLETE ====
[student@szabist ~]$ hostname
puppet
```

After this I will disable my firewall service on my master server

```

Password:
[root@puppet ~]# systemctl stop firewalld

```

And then finally I will start and enable my puppet server service on my master

```

[root@puppet ~]# systemctl start puppetserver
[root@puppet ~]# systemctl enable puppetserver

```

After this I will leave my master server for some time and will focus on my client server. After moving towards my client server, I will again install the same puppet package that I installed in my master server.

```

[root@szabist ~]#
dnf -y install https://yum.puppet.com/puppet-release-el-9.noarch.rpm
CentOS Stream 9 - BaseOS                               2.4 kB/s | 4.5 kB  00:01
CentOS Stream 9 - BaseOS                               719 kB/s | 7.9 MB 00:11
CentOS Stream 9 - AppStream                             5.8 kB/s | 5.0 kB 00:00
CentOS Stream 9 - AppStream                             889 kB/s | 18 MB  00:20
CentOS Stream 9 - Extras packages                       6.0 kB/s | 6.0 kB 00:01
CentOS Stream 9 - Extras packages                       12 kB/s | 15 kB  00:01
Docker CE Stable - x86_64                              9.8 kB/s | 3.5 kB 00:00
Docker CE Stable - x86_64                              43 kB/s | 32 kB  00:00
puppet-release-el-9.noarch.rpm                          11 kB/s | 9.2 kB  00:00
Dependencies resolved.

Package Architecture Version Size Repository
-----
Installing:
puppet-release noarch      1.0.0-28.el9 9.2 k @commandline

Transaction Summary
-----
Install 1 Package

Total size: 9.2 k
Installed size: 3.5 k
Downloading Packages:
Running transaction check
Transaction check succeeded.
Running transaction test
Transaction test succeeded.
Running transaction
  Preparing      :
  Installing     : puppet-release-1.0.0-28.el9.noarch 1/1
  Verifying      : puppet-release-1.0.0-28.el9.noarch 1/1

Installed:
puppet-release-1.0.0-28.el9.noarch

Complete!

```

After the package is downloaded, I will then install the puppet-agent service

```

[root@szabist ~]# dnf -y install puppet-agent
Puppet Repository el 9 - x86_64                               916 kB/s | 4.2 MB 00:04
Last metadata expiration check: 0:00:04 ago on Thu 07 Dec 2023 07:25:14 PM FKT.
Dependencies resolved.

Package Architecture Version Size Repository
-----
Installing:
puppet-agent x86_64      7.27.0-1.el9 21 M puppet

Transaction Summary
-----
Install 1 Package

Total download size: 21 M
Installed size: 21 M
Downloading Packages:
puppet-agent-7.27.0-1.el9.x86_64.rpm                       1.9 MB/s | 21 MB 00:10

Total
Puppet Repository el 9 - x86_64                               1.9 MB/s | 21 MB 00:10
Importing GPG key 0x9F61EF26:
  Userid      : "Puppet, Inc. Release Key (Puppet, Inc. Release Key) <release@puppet.com>"
  Fingerprint: D681 1ED3 ADEE B944 1AF5 ABAF 4598 B6CD 9E61 EF26
  From        : /etc/pki/rpm-gpg/RPM-GPG-KEY-2025-04-06-puppet-release
Key imported successfully
Puppet Repository el 9 - x86_64                               1.6 MB/s | 1.7 kB 00:00
Importing GPG key 0xEF8D349F:
  Userid      : "Puppet, Inc. Release Key (Puppet, Inc. Release Key) <release@puppet.com>"
  Fingerprint: 6F6B 1550 9CF8 E59E 6E46 9F32 7F43 8280 EF8D 349F
  From        : /etc/pki/rpm-gpg/RPM-GPG-KEY-puppet-release
Key imported successfully
Running transaction check
Transaction check succeeded.
Running transaction test
Transaction test succeeded.
Running transaction
  Preparing      :
  Running scriptlet: puppet-agent-7.27.0-1.el9.x86_64 1/1
  Installing     : puppet-agent-7.27.0-1.el9.x86_64 1/1
  Running scriptlet: puppet-agent-7.27.0-1.el9.x86_64 1/1
  Verifying      : puppet-agent-7.27.0-1.el9.x86_64 1/1

Installed:

```

Then we will open our `/etc/hosts/` file and will add the master server IP address and hostname in it.

```
GNU nano 5.6.1 /etc/hosts
127.0.0.1 localhost localhost.localdomain localhost4 localhost4.localdomain4
::1 localhost localhost.localdomain localhost6 localhost6.localdomain6
192.168.145.130 puppet
```

After this we will enable our puppet agent service and because of this an SSL certificate will also be generated in our master server.

```
[root@szabist ~]# /opt/puppetlabs/bin/puppet resource service puppet ensure=running enable=true
Notice: /Service[puppet]/ensure: ensure changed 'stopped' to 'running'
service { 'puppet':
  ensure => 'running',
  enable => 'true',
  provider => 'systemd',
}
[root@szabist ~]# systemctl status puppet
● puppet.service - Puppet agent
   Loaded: loaded (/usr/lib/systemd/system/puppet.service; enabled; preset: disabled)
   Active: active (running) since Thu 2023-12-07 23:58:39 PKT; lmin 43s ago
     Docs: man:puppet-agent(8)
  Main PID: 6367 (puppet)
    Tasks: 1 (limit: 10698)
   Memory: 52.3M
      CPU: 6.546s
   CGroup: /system.slice/puppet.service
           └─6367 /opt/puppetlabs/puppet/bin/ruby /opt/puppetlabs/puppet/bin/puppet agent --no-daemonize

Dec 07 23:58:39 szabist systemd[1]: Started Puppet agent.
Dec 07 23:58:43 szabist puppet-agent[6367]: Starting Puppet client version 7.27.0
```

Then after this we will move again to our master-server where we will check for SSL certificate request and after we have found the request, we will sign the certificate and a connection will be established between our master and client server.

```
[root@puppet ~]# /opt/puppetlabs/server/bin/puppetserver ca list --all
Requested Certificates:
  szabist.localdomain (SHA256) 96:7E:B3:54:B4:EB:15:83:C6:28:F8:F1:15:0C:00:EE:A9:70:04:ED:6A:6B:F9:08:4D:9A:99:7C:C3:A1:41:9E
Signed Certificates:
  puppet.localdomain (SHA256) ED:89:BF:62:AF:C4:30:9E:0D:31:CD:2E:02:16:08:00:B2:71:01:AD:9D:B3:59:86:BS:C0:6D:B1:1C:E7:ED:93 alt names: ["DNS:puppet", "DNS:puppet.localdomain"]
  authorization extensions: [pp_cli_auth: true]
[root@puppet ~]# /opt/puppetlabs/server/bin/puppetserver ca sign --certname szabist.localdomain
```

After successfully signing the certificate, we will then make a manifest file named as “sample.pp”

```
[root@puppet ~]# touch /etc/puppetlabs/code/environments/production/manifests/sample.pp
```

Then we will move to our client server to test our manifest file

```
[root@szabist ~]# /opt/puppetlabs/bin/puppet agent --test
Info: Using environment 'production'
Info: Retrieving pluginfacts
Info: Retrieving plugin
Info: Caching catalog for szabist.localdomain
Info: Applying configuration version '1701977284'
Notice: Applied catalog in 0.03 seconds
```

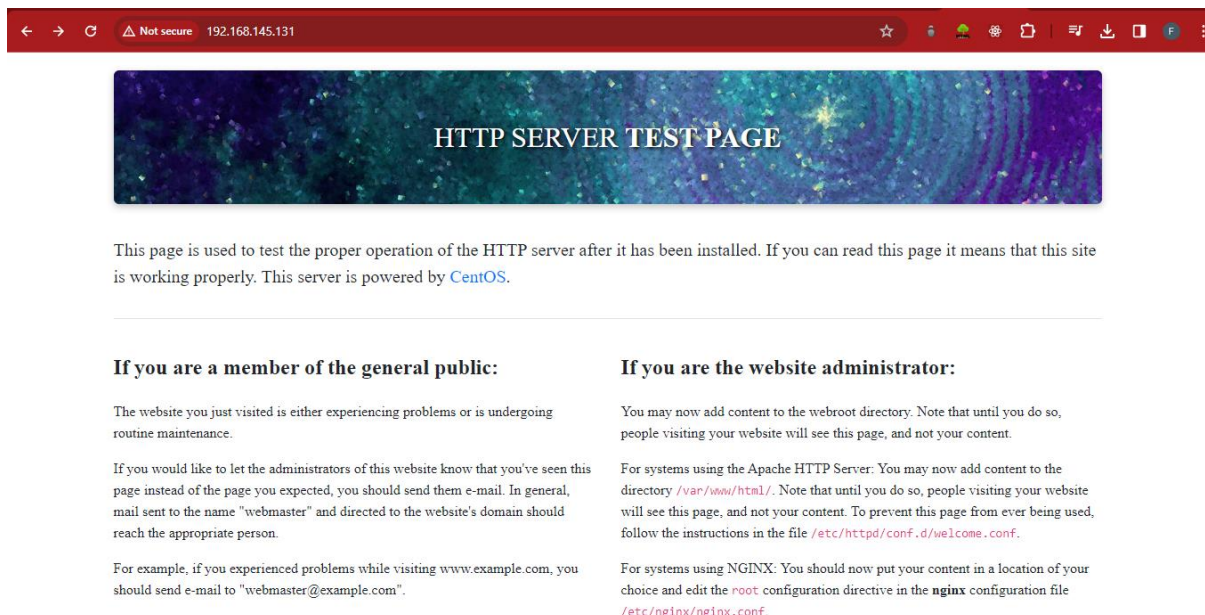
After successful testing is completed, we will again move to our master server and then edit our manifest file where will write manifest to enable http service on our client server.

```
GNU nano 5.6.1 /etc/puppetlabs/code/environments/production/manifests/sample.pp
node 'szabist.localdomain'{
  package { 'httpd' ;
    ensure => installed,
  }
}
```

Then again, we will move to our client server to test our manifest file so that the http service can be enabled in our client and then we will also install Apache service.

```
[root@szabist ~]# /opt/puppetlabs/bin/puppet agent --test
Info: Using environment 'production'
Info: Retrieving pluginfacts
Info: Retrieving plugin
Info: Caching catalog for szabist.localdomain
Info: Applying configuration version '1701977460'
Notice: Applied catalog in 0.68 seconds
[root@szabist ~]# apachectl start
```

Then we will disable our client firewall and test our http service in internet browser.



References:

- <https://www.slashroot.in/puppet-tutorial-introduction-puppet-configuration-management-tool>
- <https://www.browserstack.com/guide/what-is-puppet-devops>
- <https://www.devopsschool.com/blog/what-is-puppet-and-how-it-works-an-overview-and-its-use-cases/>
- <https://www.javatpoint.com/puppet-architecture>
- <https://www.puppet.com/docs/puppet/5.5/architecture.html>
- <https://www.youtube.com/watch?v=F-NGOvYiV9g>