When managing Apache HTTP Server and Tomcat together using Puppet, you’re typically looking to configure a web server (Apache) that serves as a reverse proxy to an application server (Tomcat). This setup is common in scenarios where Apache handles static content and SSL/TLS termination while Tomcat handles dynamic content and application logic.

Here’s a guide on how to configure and manage both Apache and Tomcat using Puppet, including integrating them for a seamless web application deployment.

**1. Installing Apache and Tomcat**

First, you need to ensure that both Apache and Tomcat are installed on your systems. You can use Puppet to automate the installation and configuration of these services.

**Puppet Manifest for Apache HTTP Server Installation**

puppet

# Install Apache HTTP Server

package { 'httpd':

ensure => installed,

}

# Ensure Apache service is running

service { 'httpd':

ensure => running,

enable => true,

}

# Configure Apache

file { '/etc/httpd/conf.d/your\_site.conf':

ensure => file,

content => "

<VirtualHost \*:80>

ServerName www.example.com

ProxyPass / http://localhost:8080/

ProxyPassReverse / http://localhost:8080/

</VirtualHost>

",

notify => Service['httpd'],

}

**Puppet Manifest for Tomcat Installation**

puppet

# Install Tomcat (assuming Tomcat is available via package manager or you use a specific module)

package { 'tomcat':

ensure => installed,

}

# Ensure Tomcat service is running

service { 'tomcat':

ensure => running,

enable => true,

}

# Configure Tomcat (if needed)

file { '/etc/tomcat/server.xml':

ensure => file,

content => '

<!-- Add or modify Tomcat configurations here -->

',

notify => Service['tomcat'],

}

**2. Configuring Apache as a Reverse Proxy for Tomcat**

To make Apache serve as a reverse proxy for Tomcat, you need to configure Apache to forward requests to Tomcat. This is done using the mod\_proxy and mod\_proxy\_http modules.

**Puppet Manifest for Apache Reverse Proxy Configuration**

puppet

# Ensure mod\_proxy and mod\_proxy\_http are enabled

apache::mod { 'proxy':

ensure => 'present',

}

apache::mod { 'proxy\_http':

ensure => 'present',

}

# VirtualHost configuration for reverse proxy

file { '/etc/httpd/conf.d/your\_site.conf':

ensure => file,

content => "

<VirtualHost \*:80>

ServerName www.example.com

ProxyPreserveHost On

ProxyPass / http://localhost:8080/

ProxyPassReverse / http://localhost:8080/

ErrorLog logs/error\_log

CustomLog logs/access\_log common

</VirtualHost>

",

notify => Service['httpd'],

}

**3. Ensuring Correct File Permissions**

Ensure that the correct file permissions are set for the configuration files and directories used by Apache and Tomcat.

puppet

# Ensure proper ownership and permissions for Apache and Tomcat directories

file { '/etc/httpd/conf.d/your\_site.conf':

owner => 'root',

group => 'root',

mode => '0644',

}

file { '/etc/tomcat/server.xml':

owner => 'tomcat',

group => 'tomcat',

mode => '0644',

}

**4. Managing SSL/TLS**

If you need to configure SSL/TLS for Apache, you can use Puppet to set up certificates and configure Apache to handle HTTPS traffic.

**Puppet Manifest for SSL Configuration**

puppet

# Install mod\_ssl for SSL support

package { 'mod\_ssl':

ensure => installed,

}

# SSL VirtualHost configuration

file { '/etc/httpd/conf.d/ssl.conf':

ensure => file,

content => "

<VirtualHost \*:443>

ServerName www.example.com

SSLEngine on

SSLCertificateFile /etc/pki/tls/certs/your\_cert.crt

SSLCertificateKeyFile /etc/pki/tls/private/your\_key.key

ProxyPreserveHost On

ProxyPass / http://localhost:8080/

ProxyPassReverse / http://localhost:8080/

ErrorLog logs/ssl\_error\_log

CustomLog logs/ssl\_access\_log combined

</VirtualHost>

",

notify => Service['httpd'],

}

**5. Example Complete Puppet Manifest**

Here’s a combined example of a Puppet manifest to install and configure Apache as a reverse proxy for Tomcat, including SSL support:

puppet

# Install Apache HTTP Server and Tomcat

package { ['httpd', 'tomcat']:

ensure => installed,

}

# Ensure services are running

service { ['httpd', 'tomcat']:

ensure => running,

enable => true,

}

# Enable necessary Apache modules

apache::mod { ['proxy', 'proxy\_http']:

ensure => 'present',

}

# Configure Apache for reverse proxy

file { '/etc/httpd/conf.d/your\_site.conf':

ensure => file,

content => "

<VirtualHost \*:80>

ServerName www.example.com

ProxyPreserveHost On

ProxyPass / http://localhost:8080/

ProxyPassReverse / http://localhost:8080/

ErrorLog logs/error\_log

CustomLog logs/access\_log common

</VirtualHost>

",

notify => Service['httpd'],

}

# Configure SSL for Apache

file { '/etc/httpd/conf.d/ssl.conf':

ensure => file,

content => "

<VirtualHost \*:443>

ServerName www.example.com

SSLEngine on

SSLCertificateFile /etc/pki/tls/certs/your\_cert.crt

SSLCertificateKeyFile /etc/pki/tls/private/your\_key.key

ProxyPreserveHost On

ProxyPass / http://localhost:8080/

ProxyPassReverse / http://localhost:8080/

ErrorLog logs/ssl\_error\_log

CustomLog logs/ssl\_access\_log combined

</VirtualHost>

",

notify => Service['httpd'],

}

# Ensure proper file permissions

file { ['/etc/httpd/conf.d/your\_site.conf', '/etc/httpd/conf.d/ssl.conf']:

owner => 'root',

group => 'root',

mode => '0644',

}

file { '/etc/tomcat/server.xml':

owner => 'tomcat',

group => 'tomcat',

mode => '0644',

}

This example ensures both Apache and Tomcat are installed, services are running, Apache is configured as a reverse proxy (with and without SSL), and proper file permissions are set. Adjust the configuration files and paths according to your specific requirements and environment.