

Install Apache / Verify Status

Install HTTPD service	<code>yum install httpd</code>
Check httpd status	<code>systemctl status httpd.service</code>

Configuring Apache HTTP Server

Inspect Control Script	<code>less /etc/systemd/system/multi-user.target.wants/httpd.service</code> displays the contents of the httpd.service file for the Apache HTTP server in the systemd multi-user target.
Get default start-up state	<code>systemctl get-default</code> shows the default target (runlevel) that the system boots into.
Find config file	<code>find / -name httpd.conf</code>
Inspect config file	<code>nano /etc/httpd/conf/httpd.conf</code> <code>nano [location of config file found using find]</code>
.htaccess & .htpasswd	Note that these are hidden by default, to prevent files being viewed by web clients
ErrorLog	Find and take note of where the errorlog is located (usually <code>logs/error_log</code>)
DocumentRoot	The web document location is usually the <code>/var/www/html</code>

Investigate Processes, Make & Test Apache

List processes and filters for those related to httpd	<code>ps -ef grep httpd</code>
Rules for incoming traffic	<code>iptables -L INPUT</code>
Create index.html file	1) Navigate to appropriate directory <code>e.g.: cd /var/www/html</code> 2) create and edit file using nano <code>e.g.: sudo nano index.html</code>
View access log	<code>cat /var/log/httpd/access_log</code>
Request local page	<code>curl http://localhost</code>

MySQL/MariaDB Installation, Start & Status

MariaDB Installation	<code>yum install mariadb-server</code>
Confirm MySQL/MariaDB is installed	<code>find / -name mysql</code>
Start MariaDB	<code>systemctl start mariadb</code>
Check MariaDB Status	<code>systemctl status mariadb</code>
Confirm servers are running	<code>ps -ef</code> This produces a list of running servers, where you will search for <code>mysql</code> in the far left column (the UID (User ID))
Set new password for mysqladmin root	<code>mysqladmin -u root password [INSERT PASSWORD]</code>

MySQL Config file & Data Directory

Find config file (my.cnf) location	<code>sudo find / -name my.cnf less</code>
Navigate to config file directory	<code>cd [INSERT DIRECTORY]</code> e.g. in my case, my.cnf was located found to be /etc/my.cnf, so <code>cd /etc</code> is used
View contents of config file	<code>cat my.cnf</code>
Locate MySQL Daemon	<code>find / -name mysqld</code>

Create & Populate Database

Enter MariaDB Server	<code>mysql -h localhost -u root -p</code> Then enter password created previously. Note, password will not show any typing.
Create database	<code>`CREATE DATABASE [database name];</code> e.g. <code>CREATE DATABASE food;</code>
Change to created database	<code>USE [database]</code> e.g. <code>`USE food'</code>
Exit MariaDB	<code>quit</code>
Confirm database was created outside of MariaDB	Change to appropriate directory <code>cd /var/lib/mysql</code> Display contents of current directory using <code>ls</code>
Create Table	<code>CREATE TABLE restaurant (name VARCHAR(40), type VARCHAR(40), location VARCHAR(40));`</code> VARCHAR(n) defines variable length
Insert values into restaurant table	<code>INSERT INTO restaurant (name, type, location) values ("Pizzeria", "Italian", "SW10");</code>
Show table	<code>SHOW TABLES; DESCRIBE restaurant; SELECT * FROM restaurant;</code>
Delete value from table	<code>DELETE FROM restaurant WHERE name="Pizzeria" & location="SW10";</code>
Create new user	<code>GRANT SELECT ON food.restaurant TO bayan@localhost IDENTIFIED BY "bayans_password";</code>

Firewalls

Confirm firewalld is running	<code>systemctl status firewalld</code>
Check firewall configuration	<code>firewall-cmd --list-all</code>
Display firewall rules	<code>`iptables -L'</code>
Services/ports available for sshd	<code>systemctl status sshd</code>
Services/ports available for httpd	<code>systemctl status httpd</code>
Services/ports available for vsftpd	<code>systemctl status vsftpd</code> if not installed, use <code>yum install</code> . e.g. <code>yum install vsftpd</code>
Stop firewalld, then check if running to confirm it is infact stopped**	<code>systemctl stop firewalld</code>
Start firewalld	<code>systemctl start firewalld</code>
Add http service to firewall configuration	<code>firewall-cmd --add-service http</code>



Firewalls (cont)

Add ftp service to firewall configuration	<code>firewall-cmd --add-service ftp</code>
iptables rules for accepting traffic for ports 22(SSH), 80(HTTP), and 21 (FTP)	<code>iptables -A INPUT -p tcp --dport 22 -j ACCEPT</code>
	<code>iptables -A INPUT -p tcp --dport 80 -j ACCEPT</code>
	<code>iptables -A INPUT -p tcp --dport 21 -j ACCEPT</code>
Add rules to output chain	<code>iptables -A OUTPUT -m state --state ESTABLISHED, RELATED -j ACCEPT</code>
Dropping default rules for INPUT and OUTPUT traffic	<code>iptables -P INPUT DROP</code>
	<code>iptables -P OUTPUT DROP</code>

SELinux

Install setroubleshoot and httpd	<code>sudo yum install setroubleshoot httpd</code>
Enable httpd	<code>systemctl enable httpd</code>
Start httpd	<code>systemctl start httpd</code>
Check default directory for HTML files	<code>cat /etc/httpd/conf/httpd.conf grep DocumentRoot</code>
Check SELinux permissions / context	<code>ls -lZ index.html</code>
Temporarily disable SELinux enforcement for troubleshooting or testing without changing the permanent configuration.	<code>setenforce 0</code>
Re-enable SELinux enforcement after it has been disabled, restoring its security policies.	<code>setenforce 1</code>
Apply default SELinux to a file:	<code>/sbin/restorecon -v /var/www/html/servlet.html</code>

