Cheat sheet:

## Permission Modes

**7 5 5**

**user group world**

**r+w+x r+x r+x**

**4+2+1 4+0+1 4+0+1 = 755**

The permission mode is computed by adding up the following values for the user, the file group, and for everyone else. The diagram shows how.

* **R**ead 4 - Allowed to read files
* **W**rite 2 - Allowed to write/modify files
* e**X**ecute1 - Read/write/delete/modify/directory
  + chmod 600 file
  + chown user file (change ownership)
  + chown group file (change group)

# Shell commands:

|  |  |
| --- | --- |
| ***#*** | Marks a command. |
| ***alias*** | Displays alias. |
| ***bg*** | Resumes job in the background. |
| ***break*** | Resumes execution after the loop. |
| ***breaksw*** | Breaks from a switch command; resumes after the endsw command. |
| ***case*** | Defines a label in a switch command. |
| ***cd*** | Changes directory. |
| ***chdir*** | Changes directory, same as *cd*. |
| ***continue*** | Continues a loop. |
| ***default*** | Specifies the default case in a switch. |
| ***dirs*** | Displays the directory stack. |
| ***echo*** | Writes arguments to the standard output of the shell. |
| ***eval*** | Evaluates a command. |
| ***exec*** | Executes the command in the current shell. |
| ***exit*** | Exits the shell. |
| ***fg*** | Brings a job in the foreground. |
| ***foreach*** | Specifies a looping control statement and execute a sequence of commands until reaching an end command. |
| ***glob*** | Writes arguments to the standard output of the shell, like the echo command, but without the new line. |
| ***goto*** | Continues execution after the specified label. |
| ***hashstat*** | Displays hash table statistics. |
| ***history*** | Displays the history list. |
| ***if*** | Executes a command if condition met. |
| ***jobs*** | Lists active jobs. |

|  |  |
| --- | --- |
| ***kill*** | Sends a signal to a process. *term* (terminate) is the default signal. |
| ***limit*** | Sets or list system resource limits. |
| ***login*** | Logs on. |
| ***logout*** | Logs out. |
| ***nice*** | Changes the priority of commands run in the shell. |
| ***nohup*** | Ignores the hangup signal. |
| ***notify*** | Notifies the user about changes in job status. |
| ***onintr*** | Tells the shell what to do on interrupt. |
| ***popd*** | Pops the top directory off the directory stack and changes to the new top directory. |
| ***pushd*** | Exchanges the top two elements of the directory stack. |
| ***rehash*** | Re-computes the hash table of the contents of the directories in the path shell variable. |
| ***repeat*** | Repeats the execution of a command. |
| ***set*** | Displays or set the value of a shell variable. |
| ***setenv*** | Sets environment variables. |
| ***shift*** | Shifts shell arguments. |
| ***source*** | Reads commands from a script. |
| ***stop*** | Stops a background job. |
| ***suspend*** | Stops the current shell. |
| ***switch*** | Starts a switch. |
| ***time*** | Displays the time used to execute commands. |
| ***umask*** | Shows or set file permissions. |
| ***unalias*** | Removes command alias. |
| ***unhash*** | Disables the internal hash table. |
| ***unlimit*** | Removes limitations on system Resource. |
| ***unset*** | Deletes shell variables. |
| ***unsetenv*** | Deletes environment variables. |
| ***wait*** | Waits for background jobs to complete. |
| ***while …end*** | Executes the commands between the while and matching end statements repeatedly. |
| ***@*** | Displays or set the values of all the shell variables. |

# Yum:

* Yum install
* Yum info
* Yum search

# Ansible:

Site.yml

*# site.yml*

---

- hosts: srv001

remote\_user: vagrant

sudo: true

roles:

- apache

- mysql

- wordpress

- ssl

- hosts: srv002

sudo: true

roles:

- common

Vagrant\_hosts.yml

-

name: srv001

ip: 192.168.56.10

-

name: srv002

ip: 192.168.56.11

* Copy files from ansible to VM:

Template: src=x dest=y

# Apache:

*# roles/apache/main.yml*

---

- name: apache

yum: pkg={{ item }} state=installed

with\_items:

- httpd

- mod\_ssl

- php

- php-xml

- php-mysql

- php-gd

- name: start apache

service: name=httpd state=running enabled=yes (systemctl start httpd.service)

- name: start f

service: name=firewalld state=running enabled=yes

- name: Apply firewall rules

firewalld:

zone=public

service={{ item[0] }}

state=enabled

permanent={{ item[1] }}

with\_nested:

- [ http, https ]

- [ true, true ]

* Set up test files for php -> sudo nano /var/www/html/info.php

Add <?php phpinfo(INFO\_GENERAL); ?>   
service httpd restart

# Mysql: After mysql install - /usr/bin/mysql\_secure\_installation -> prompt root pwd

*# roles/sql/main.yml*

---

- name: install sql

yum: pkg={{ item }} state=installed

with\_items:

- mariadb-server (centos mysql)

- MySQL-python

- name: start sql

service: name=mariadb.service state=running enabled=yes

- name: restart apache after mysql

service: name=httpd state=restarted

* Mysql –u root –p -> log in to test (after secure install)
* $ mysqladmin -u root password NEWPASSWORD (first time)
* Reset root password:
  1. Systemctl stop mariadb.service
  2. Mysqld\_safe –skip-grant-tables –skip networking & (unrestricted access with safe script. Skip = no restrictions
  3. Mysql –u root
  4. Use mysql;
  5. Update user set password=PASSWORD(“new-password”) where user=’root’;
  6. Flush privileges;
  7. Exit
* Reset user password:
  1. Mysqladmin –u root –p ‘old pwd’ password ‘new pwd’

# Firewall:

Centos 7 -> add rules

firewall-cmd --permanent --zone=public --add-service=http

firewall-cmd --permanent --zone=public --add-service=https

firewall-cmd --reload (or restart)

# Wordpress:

*# roles/wordpress/main.yml*

---

- name: Wordpress | Download wordpress

command: wget http://wordpress.org/latest.tar.gz

tags: wordpress

- name: Wordpress | Extract archive

command: /bin/tar xvf latest.tar.gz -C /var/www/html --strip 1

tags: wordpress

- name: add group wordpress

group: name=wordpress

- name: add user wordpress

user: name=wordpress group=wordpress home=/var/www/html

- name: Wordpress | Create the wordpress database

mysql\_db: name=wordpress\_database state=present

tags: wordpress

- name: Wordpress | Fetch random salts for Wordpress config

local\_action: command curl https://api.wordpress.org/secret-key/1.1/salt/

register: wp\_salt

tags: wordpress

- name: Wordpress | Create Wordpress database user

mysql\_user: name=wordpress password=wordpressp priv=\*.\*:ALL host=localhost state=present

tags: wordpress

- name: Wordpress | Copy Wordpress config file

template: src=wp-config.php dest=/var/www/html

tags: wordpress

- name: change ownership wordpress

file: path=/var/www/html owner=wordpress group=wordpress state=directory recurse=yes

WORDPRESS wp-config.php

<?php

*// \*\* MySQL settings - You can get this info from your web host \*\* //*

/\*\* The name of the database for WordPress \*/

define('DB\_NAME', 'wordpress\_database');

/\*\* MySQL database username \*/

define('DB\_USER', 'wordpress');

/\*\* MySQL database password \*/

define('DB\_PASSWORD', 'wordpressp');

/\*\* MySQL hostname \*/

define('DB\_HOST', 'localhost');

/\*\* Database Charset to use in creating database tables. \*/

define('DB\_CHARSET', 'utf8');

/\*\* The Database Collate type. Don't change this if in doubt. \*/

define('DB\_COLLATE', '');

\* Authentication Unique Keys and Salts.

{{ wp\_salt**.**stdout }}

/\*\*#@-\*/

/\*\*

\* WordPress Database Table prefix.

$table\_prefix **=** 'wp\_';

define('WPLANG', '');

/\*\*

\* For developers: WordPress debugging mode.

\*

\* Change this to true to enable the display of notices during development.

\* It is strongly recommended that plugin and theme developers use WP\_DEBUG

\* in their development environments.

\*/

define('WP\_DEBUG', **false**);

*/\* That's all, stop editing! Happy blogging. \*/*

/\*\* Absolute path to the WordPress directory. \*/

**if** ( **!**defined('ABSPATH') )

define('ABSPATH', dirname(**\_\_FILE\_\_**) **.** '/');

/\*\* Sets up WordPress vars and included files. \*/

**require\_once**(ABSPATH **.** 'wp-settings.php');

# General:

* Run automatically at boot:  
  (Chkconfig httpd on) systemctl enable httpd.service
* /var/www in centos not applicable -> use /var/www/html/x
* Cat > file.extension << EOF

>text

>EOF