# commands

сряда, 22 февруари 2023 г. 20:20

RHCSA RHEL

| | |
|---|---|
| man command | |
| -h | |
| --help | |
| -? | |
| ctrl d | logout |
| **Files, Directories** | |
| pwd | current dir |
| cd ~/ | home dir |
| $HOME | |
| cd ..<br>cd-<br>cd ~username<br>cd | moves one dir up<br>moves to previous dir<br>goes to home dir<br>goes go your home dir |
| cp file.txt tonewfile.txt<br>cp -R /home/dir /home/newdir | <br>copies directories with -R |
| mv old_name new_name | renames files; also moves with mv /dir |
| mkdir -m777 -p /dir1/newdir/dir2<br>4<br>2<br>1<br>764 | m sets perm. 777 is rwx<br>-p makes newdir betw. dir1 and dir2<br>r<br>w<br>e<br>users, group, others |
| chmod 764 file.txt | users rwx, group rw, others r |
| chown [option] owner[:group] file(s)<br>chown userName:groupName file<br>chwon user file.txt | <br>changes user and group for file<br>makes user owner of file.txt |
| chgrp grpName fileName | change group |
| rmdir<br>rm -r | rm<br>remove forcefully |
| touch<br>cat > fileName | make empty file |
| file file.txt | get type of file.txt |
| **VIM** | |
| gg | Beginning of file |
| ALT / | end of file |
| u | undo last change. capital u undo all changes |
| / | search |
| :set number | line numbers |
| vim filename +89 | goes on line 89 |
| CTRL Z<br>fg<br>bg<br>jobs | minimizes vim file<br>opens most recent job<br>see job in bg<br>see jobs |
| **gzip, tar compression** | |
| gzip fileName | with gzip compression |
| gunzip fileName.gz | |
| tar -cvf dir.tar /dir<br>tar -cfz dir.tar /dir | uncompressed tar archive<br>gzip tar rachive compression |
| -uvf<br>-rvf | archives and adds to an existing archive<br>-\|\|- |

| | |
|---|---|
| -tvf | lists the contents of a file |
| tar -xf file.tar<br>tar -xvg file.tar -C /dir | extract<br>extart at a directory with -C |
| | |
| **Lists** | |
| df -mh | disk usage in mbs |
| du -sh | currect dir usage |
| head file.txt | show first 10 lines |
| tail -100 file.txt | show last 100 lines (10 by def) |
| wc file.txt | counts words/lines |
| **Find** | |
| locate -i two*words | search -i (not case sens.) * is empty spc |
| find /where -name name.txt<br>find ./ -type d -name dirname | searches for directories |
| grep what file.txt | |
| nd /dir -name x<br>nd /dir -size +100M | list items beginning with x |
| **users, ,groups** | |
| id<br>who<br>finger<br>finger username<br>pinky<br>w<br>whoami | show active user details<br>logged users<br>info on all logged users<br>logged users and activity |
| last | last logins |
| cat /etc/passwd<br>cat /etc/passwd \| cut -d : -f 1 | list users, rights<br>only usenames |
| cat /etc/groups | list groups |
| adduser username<br>userdel username | |
| passwd username | change pw |
| usermod -a -G groupname username | |
| **system** | |
| ssh user@host | |
| ping hostname | |
| hostname -i | hostname ip |
| uname -a; uptime | sys info |
| last reboot | |
| ip a | ipto e na inet |
| **var, commands** | |
| let variable=value | integer val. |
| set | lists variables and functions |
| echo $variable | display value |
| env | display all variables |
| unset varName | removes a var |
| alias name=command<br>unalias name | |
| watch -n interval command<br>watch -n 5 'ntpq -p' | run through an interval<br>execute and display every 5 secs. |
| sleep interval && command | postpone cmnd execution |
| where is command | find binary/source/ manual for command |
| **UTIL** | |

| | |
|---|---|
| logrotate | |
| free -mh | |
| ps -ef<br>ps -u username<br>top<br>htop | <br>all processes associated with the user<br><br>improved top |
| pidof processName | get the process ID |
| kill PID | |
| **RHCSA** | |
| ><br>>> | Writes to a file (and replace all in it)<br>Adds at the end of the file |
| ~<br>ls -al ~ > list | Root home dir<br>Saves ls to list at home dir |
| sort < list<br>-S<br>-t | Get input from list and sort it<br>by size<br>last modified |
| ls -smth 2> file 2>&1 | saves errors in a file AND displays them (stderr to sdtout) |
| | |
| grep '^music' file | grep all line starting with music |
| | |
| ls -l - if a file type begins with<br>drw - directory<br>lrw - link<br>-rw - regural file | |
| ls -ltr | lists recently created |
| ls -li | lists inode info |
| | |
| ln | hard link (to the inode) |
| ln -s | soft link (to the file, if it’s renamed the link is gone) |
| | |
| | |
| | |
| | |
| **Ansible** | |
| ssh setup | |
| apt-get update<br>apt-get install openssh-server<br>service sshd start | server must have openssh. could use apt install |
| ls -la .ssh | list ssh keys |
| ssh-keygen -t ed25519 | create key; the create for ansible |
| ssh-copy-id -i ~/.ssh/id_ed25519.pub 10.0.2.4<br>ssh-copy-id -i ~/.ssh/ansible.pub 10.0.2.4 | |
| ssh -i ~/.ssh/ansible 10.0.2.4 | test connecting to the server with ansible key |
| alias ssha='eval $(ssh-agent) && ssh-add' | to save the passphrase for main acc with alias. Add to vim .bashrc |
| git push pull status | |
| git clone git@github.com:usernam/ansible.git | set git rep, ssh |
| git config --global user.name "" | or user.email |
| git status<br>git diff | check for changes |
| git add filename | |
| git commit -m "initial commit - readme file" | |
| git push<br>git push origin main | push changes to files |
| git pull | pull changes |
| **setting up ansible** | |

| | |
|---|---|
| sudo apt update | |
| sudo apt install ansible | |
| vim inventory | write hosts; push to git<br>better - create folder host_vars with separate .yml files |
| ansible all --key-file ~/.ssh/ansible -i inventory -m ping | but can shorten it with setting ansible.cfg -> |
| [defaults]<br>inventory = inventory<br>private_key_file = ~/.ssh/ansible | vim ansible.cfg |
| ansible all -m ping | |
| ansible all --list-hosts | |
| | |
| ansible all -m apt -a update_cache=true --become --ask-become-pass | |
| ansible all -m apt -a "name=snapd state=latest" --become --ask-become-pass | update to the latest version snapd |
| **playbooks**; install apache | |
| vim install_apache.yml | |
| ---<br><br>- hosts: all<br>  become: true<br>  tasks:<br><br>  - name: install apache2 package<br>    apt:<br>      name: apache2<br>      state: latest<br><br>_______________**or multiple**_________________<br>name:<br>        - apache2<br>        - libapache2-mod-php | |
| ansible-playbook --ask-become-pass install_apache.yml | then access it from the web(I had to enable the port): http://127.0.0.1:8080/ |
| | when parameter |
| ansible-playbook --tags apache --ask-become-pass install_apache.yml | tags: apache, apache2, ubuntu |
| **manage files**, added default site html config | |
| - hosts: all<br>  become: true<br>  tasks:<br><br>  - name: install apache2 and php package<br>    apt:<br>      name:<br>        - apache2<br>        - libapache2-mod-php<br>      state: latest<br>      update_cache: yes<br><br>  - name: install mariadb package (Ubuntu)<br>    apt:<br>      name: mariadb-server<br>      state: latest<br>    when: ansible_distribution ==  "Ubuntu"<br><br>  - name: copy default html file for site<br>    tags: apache,apache2,httpd<br>    copy:<br>      src: default_site.html<br>      dest: /var/www/html/index.html | install apache, php, mariadb<br><br>set html file for the apache site<br><br>(tags are optional) |

| | |
|---|---|
| owner: root<br>group: root<br>mode: 0644 | |
| **manage services** | |
| - name: start apache2 (Ubuntu)<br>service:<br>name: apache2<br>state: started<br>enabled: yes | manage services |
| - name: edit line in file<br>lineinfile:<br>path:<br>/home/danasoft/ansible_tutorial/tutorial12<br>regexp: '^edit here'<br>line: This is the new line | edit line in fine starting with edit here |
| - name: restart apache2 when changed<br>service:<br>name: apache2<br>status: restarted<br>when: apache2.changed | restart when change is detected. at the play should be added:<br>register: apache2 |
| **adding, managing users, bootstrap** | this playbook can be user as bootstrap for new servers |
| cat files/sudoer_user1<br>user1 ALL=(ALL) NOPASSWD: ALL | set user permissions(root |
| cat tutorial13.yml<br>- hosts: all<br>become: true<br>tasks:<br><br>- name: create new user<br>user:<br>name: user1<br>groups: root<br><br>- name: add ssh key for the new user<br>authorized_key:<br>user: user1<br>key: "ssh-ed25519<br>AAAAC3NzaC1lZDI1NTE5AAAAIPOb+a9v8rGtt4Bx+<br>q6u0lvm8fzshdtqR3vEjlLCWn/e ansible"<br><br>- name: add sudoers file for user1<br>copy:<br>src: sudoer_user1<br>dest: /etc/sudoers.d/user1<br>owner: root<br>group: root<br>mode: 0440 | create user as root on the target server |
| cat ansible.cfg<br>[defaults]<br>inventory = inventory<br>private_key_file = ~/.ssh/ansible<br>remote_user = user1 | add user in ansible.cfg(below). now **ansible-playbook can be run without password**! (--ask-become-pass). they are executed as user1 |
| roles, tasks | |
| cat playbook_installs.yml<br>---<br><br>- hosts: all<br>become: true<br>roles:<br>- base<br><br>- hosts: web_servers<br>become: true<br>roles:<br>- web_servers | like OOP. create folders for roles<br><br>base is the ssh key setup. init for new servers<br><br>/web_servers/tasks/main.yml<br>make sure to create the files folder in /web_servers/ |

| | |
|---|---|
|   - db_servers | |
| cat main.yml<br>- name: install mariadb package (Ubuntu)<br>  apt:<br>    name: mariadb-server<br>    state: latest | /db_servers/tasks/main.yml |
| **variables and handlers** | |
| mkdir host_vars/ | |
| cat host_vars/10.0.2.4.yml<br>apache_package_name: apache2<br>apache_service: apache2<br>php_package_name: libapache2-mod-php | add variables here<br>1 .yml for each server<br>if you have CentOS apache2 is httpd, php is php, rename them.<br>Variables stay the same, **see below**. this way code is shorter |
| cat roles/web_servers/handlers/main.yml<br>- name: restart_apache<br>  service:<br>    name: "{{ apache_service }}"<br>    state: restarted | 1 - handler |
| - name: edit line in file<br>  lineinfile:<br>    path: /home/danasoft/ansible_tutorial/tutorial12<br>    regexp: '^This is the new line'<br>    line: This is the new line now 2301<br>  notify: restart_apache | 2 - in the task (roles/web_servers/tasks/main.yml) |
| **templates** | |
| | |
| **docker** | |
| docker pull ubuntu<br>docker run  -it -d ubuntu | <br>create container -iteractive -background |
| docker ps | list containers |
| docker inspect ** | inspect, IP and first numbers of container |
| | |
| Ansible loop | |
| - name: Deploy a web application<br>  hosts: db_and_web_server<br><br>  tasks:<br>    - name: Install dependencies<br>      apt:<br>          name={{ item }}<br>          state=present<br>      with_items:<br>       - python<br>       - python-setuptools<br>       - python-dev | with items - loop for multiple items |
| - name: install python dependencies<br>      pip:<br>        name={{item}}<br>      with_items:<br>        - flask<br>        - flask-mysql | loop for multiple items |
| - name: Copy file<br>      copy:<br>        src: app.py<br>        dest: /opt/app.py | |
| - name: shell command<br>      shell: FLASK_APP=/opt/app.py nohup flask run --host=0.0.0.0 & | |
| separate tasks (reusable) | |
| - name: taskName<br>  hosts: all | |

| | |
|---|---|
| tasks:<br>- include: tasks/deploy_db.yml<br>- include: tasks/deploy_web.yml | |
| separate host_vars inventory | |
| mkdir host_vars | |
| vim db_and_web_server.yml<br><br>ansible_ssh_pass: Passw0rd<br>ansible_host: 192.168.1.14 | |
| move vars to the host_vars inventory | |
| db_name: employee_db<br>db_user: db_user<br>db_password: Passw0rd | |
| | |
| | |
| | |
| | |
| | |
| | |
| | |
| | |
| | |
| | |
| | |
| | |