

Linux comande u DevOps-u

Linux naredbe u DevOpsu

Linux osnove i skriptovanje su jedna od najznačajnijih veština DevOp-a.

Većina kompanija ima svoje okruženje na Linuxu, takođe i mnoge CM alate kao što su - Puppet, Chef i Ansible imaju svoje glavne čvorove na Linux.

Teme koje ćemo ovde da pokrijemo su sledeće :

1. Što je Linux?
2. Zašto je Linux popularan?
3. Linux naredbe u DevOpsu.
4. Skriptiranje školjki
5. Git naredbe.

Šta je Linux?

Linux je operativni sistem otvorenog koda razvijen u zajednici za računare, servere, hostcomputere, mobilne uređaje i ugrađene uređaje. Ima podršku na skoro svim većim računarskim platformama, uključujući x86, ARM itd., što ga čini jednim od najčešće podržanih operativnih sistema.

Linuxov dizajn je sličan UNIX-u, ali je evoluirao da radi na širokom spektru hardvera, od telefona do superkompjuteru. Svaki OS zasnovan na Linuxu sadrži Linux jezgro - koji upravlja hardverskim resursima - i skup softverskih paketa koji čine ostatak operativnog sistema.

Zašto je Linux popularan?

Linux se razlikuje od ostatka operativnih sistema po mnogim važnim aspektima. Neki od njih su sledeći:

1. Prvo, a možda i najvažnije, Linux je besplatan. Ne morate da trošite nikakav iznos da biste ga preuzeli i koristili, za razliku od operativnog sistema Windows.
2. Open source -Linux je open source softver. Kod koji se koristi za kreiranje Linuxa je besplatan i dostupan javnosti za gledanje, uređivanje i -za korisnike sa odgovarajućim veštinama - da doprinesu.
3. Secure - Kada instalirate Linux na svoj sistem, nema potrebe da koristite antivirus! Linux je izuzetno bezbedan sistem. Štaviše, postoji globalna razvojna zajednica koja stalno traži načine da poboljša svoju bezbednost. Svaka nadogradnja OS-a ga čini bezbednijim i robusnijim.
4. Stabilnost i performanse - Linux obezbeđuje veoma visoku stabilnost, pa naime. Nije mu potrebno ponovno pokretanje nakon kratkog vremenskog perioda. Vaš Linux sistem retko usporava ili se zamrzava. Možete da radite bez ikakvih smetnji na linux sistemima. Linux pruža izuzetno visoke performanse na različitim mrežama i radnim stanicama.

Linux komande u DevOps-u

U ovom odeljku ćemo pogledati one najčešće korišćene dok radite u DevOps-u.

Ls

Ova komanda navodi sav sadržaj u trenutnom radnom direktorijumu.

sintaksa:

```
$ ls
```

Naredba	Opis
<code>ls</code>	Navođenjem putanje nakon <code>ls- a</code> , biće prikazano skeniranje sadržaja na toj putanji
<code>ls -l</code>	Oznaka "l" prikazuje sav sadržaj zajedno sa postavkama vlasnika, dozvolama i vremenom Marka (Dugi format)
<code>ls -a</code>	Koristeći zastavicu "a", navodi sav skriveni sadržaj u navedenom direktorijumu

sudo

Ova komanda izvršava samo tu komandu sa privilegijama vrhovnog/superuserca.

sintaksa:

\$ sudo

Naredba	Opis
sudo useradd	Dodavanje novog korisnika
sudo passwd	Postavljanje lozinke za novog korisnika
sudo userdel	Brisanje korisnika
sudo groupadd	Dodavanje nove grupe
sudo groupdel	Brisanje grupe
sudo usermod -g	Dodavanje korisnika u primarnu grupu

cat

Ova komanda može da čita, menja ili objedinjuje tekstualne datoteke.

Takođe prikazuje sadržaj datoteke.

sintaksa:

\$ cat {naziv datoteke}

Naredba	Opis
cat -b	Ovo dodaje brojeve redova praznim redovima
cat -n	Ovo dodaje brojeve redova svim redovima
cat	Ovo stiska prazne redove u jedan red
cat -E	To pokazuje \$ na zemlji retke

grep

Ova komanda traži određenu nisku/reč u tekstualnoj datoteci. Ovo je slično kao 'Ctrl + F', ali se izvodi preko CLI.

sintaksa:

\$ grep {naziv datoteke}

Naredba	Opis
grep -i	Vraća rezultate za nizove koji ne razlikuju velika i mala slova
grep -n	Daje odgovarajuće niske zajedno sa njihovim brojem reda.
grep -v	Daje rezultat redova koji se ne podudaraju sa niskom za traženje.
grep -c	Daje broj redova u kojima se rezultati podudaraju sa niskom za traženje.

sort

Ova komanda sortira rezultate pretrage po abecednom redu ili numeričkom broju. Takođe sortira datoteke, sadržaj datoteke i direktorijume.

sintaksa:

```
$ sort {naziv datoteke}
```

Naredba	Opis
sort -r	zastavica daje rezultate obrnutim redosledom
sort -f	zastava vrši sortiranje bez velikih i malih slova
sort -n	Zastavica daje rezultate numeričkim redosledom.

rep

On je komplementaran sa glavnom komandom. Komanda za rep, kao što ime podrazumeva, štampa poslednji N broj podataka date stavke. Podrazumevano štampa poslednjih 10 redova navedenih datoteka. Ako navedete više imena datoteka, podaci iz svake datoteke prethode imenu datoteke.

sintaksa:

rep [OPCIJA] ... [FILE] ...

rep -n 3 stanje.txt ili rep -3 stanje.txt => -n za br. linija

rep +25 stanje.txt

-c da li: Ispisuje zadnji 'num' bajt iz navedene datoteke.

chown

Različiti korisnici u operativnom sistemu imaju vlasništvo i dozvolu da obezbede bezbednost datoteka i postavu ograničenja ko može da promeni sadržaj datoteka. U Linuxu postoje različiti korisnici koji koriste sistem:

1. Svaki korisnik ima neka svojstva povezana sa njima, kao što su ID korisnika i kućni direktorijum. Možemo da dodamo korisnike grupi da bismo olakšali proces upravljanja korisnicima.
2. Grupe mogu da imaju nula ili više korisnika. Navedeni korisnik je povezan sa 'podrazumevanom grupom'. On takođe može da bude član drugih grupa u sistemu.

Vlasništvo i dozvole: Da bismo zaštitili i zaštitili fajlove i direktorijume u Linuxu, koristimo dozvole da bismo kontrolisali šta korisnik može da uradi sa fajlom ili direktorijumom. Linux koristi tri tipa licenci:

1. Pročitajte: Ova dozvola omogućava korisniku da čita datoteke i u direktorijumima i omogućava korisniku da čita direktorijume i poddirektorije uskladištene u njima.
2. Piše: Ova dozvola omogućava korisniku da izmeni i izbriše datoteku. Takođe, omogućava korisniku da izmeni svoj sadržaj (kreira, briše i preimenuje datoteke u njemu) za direktorijume. Ako direktorima ne date dozvolu za izvršavanje, promene neće uticati na njih.
3. Izvrši: Upisivanje dozvole za datoteku izvršava datoteku. Na primer, ako imamo datoteku sa imenom sh, tako da ako joj ne damo dozvolu za izvršavanje, ona neće biti pokrenuti.

Dozvole za tipove datoteka:

1. Korisnik: Ovaj tip dozvole za datoteku utiče na vlasnika datoteke.
2. Grupa: Ovaj tip dozvole za datoteku utiče na grupu u čijem je vlasništvu datoteka. Umesto dozvola za grupu, korisničke dozvole će biti primenjene ako je vlasnik u ovoj grupi.
3. Ostalo: Ovaj tip dozvole za datoteku utiče na sve ostale korisnike u sistemu.

chown Komanda se koristi za menjanje vlasnika datoteke ili grupe. Kad god želite da promenite vlasništvo, možete da koristite komandu chown.

Sintaksa:

```
chown [OPTION] & hellip [VLASNIK] [: [GROUP]] FILE & hellip  
chown [OPCIJA] & hellip -reference = DATUM DATUMA & hellip
```

Primer: Promena vlasnika datoteke:

[Python Pretvoriti Decimalni U Binarni](#)

```
chown owner_name ime_datoteke  
chown master file1.txt
```

gde ga *savladati*; majstorski je drugi korisnik u sistemu. Pretpostavimo da ako ste korisničko ime korisnik1 i želite da promenite vlasništvo u root (gde je trenutni direktorijum korisnik1). koristite 'sudo' pre sintakse.

```
sudo chown root file1.txt
```

chmod

Ova komanda se koristi za promenu dozvola za pristup datotekama i direktorijumima.

Sintaksa:

```
chmod {naziv datoteke}
```

Također

Tokom rada u Linux / Unix sistemu može da se koristi nekoliko fajlova i foldera, neki bi bili vidljivi, a neki ne. **Takođe**, komanda znači "**Otvori listu datoteka**". Ova komanda obezbeđuje listu otvorenih datoteka. U osnovi, on daje informacije da bi se saznale datoteke koje proces otvara. U jednom potezu navodi sve otvorene datoteke u izlaznoj konzoli.

Sintaksa:

\$ lsof [opcija] [korisničko ime]

Opcije sa primerima:

1. Lista svih otvorenih datoteka: Ova komanda navodi sve datoteke koje je otvorila bilo koja procedura u sistemu.

~ \$ lsof

1. Ovde imajte na nauda da postoje detalji o otvorenim datotekama. ID procesa, korisnik povezan sa procesom, FD (opis datoteke), veličina datoteke sve zajedno daje detaljne informacije o datoteci koju su otvorili komanda, ID procesa, korisnik, njena veličina itd.
 - **FD** predstavlja kao deskriptor datoteke.
 - **cwd** : Trenutni radni direktorij.
 - **txt**: Tekstualna datoteka.
 - **mem** : Memorijska datoteka.
 - **mapa** : Uređaj mapiran u memoriju.

Navedite sve datoteke koje je korisnik otvorio: Postoji nekoliko korisnika sistema i svaki korisnik ima različite zahteve i shodno tome koristi datoteke i uređaje. Ova komanda je korisna za pronalaženje liste datoteka koje je otvorio određeni korisnik.

- **Sintaksa:**

- *ls -l -u korisničko ime*

Uz to ovdje možemo vidjeti vrstu datoteke, a to su:

- **TEBI:** Imenik

- **REG:** Uobičajena datoteka

CHR: Posebna datoteka znakova

ifconfig

ifconfig Komanda (konfiguracija interfejsa) se koristi za konfigurisanje stanovnika mrežnih interfejsa u jezgru. Koristi se u vreme pokretanja da bi se po potrebi podesio interfejs. Nakon toga se obično koristi kada je to neophodno tokom otklanjanja grešaka ili kada je potrebno da podesite sistem. Takođe, ova komanda se koristi za dodeljivanje IP adrese i mrežne maske interfejsu ili omogućavanje ili onemogućavanje određenog interfejsa.

Sintaksa:

ifconfig -a

-s: Prikazivanje kratke liste, umesto detalja.

Sintaksa:

ifconfig -s

naredba id u Linuxu se koristi za otkrivanje imena korisnika i grupa i numeričkih ID-a (UID ili ID grupe) trenutnog korisnika ili bilo kog drugog korisnika na serveru. Ova komanda je korisna za pronalaženje sledećih dole navedenih podataka:

1. Korisničko ime i stvarni ID korisnika.
2. Saznajte određeni korisnički UID.
3. Prikažite UID i sve grupe povezane sa korisnikom.
4. Navedite sve grupe kojima korisnik pripada.
5. Prikažite bezbednosni kontekst trenutnog korisnika.

Sintaksa:

id [OPTION] & hellip [USER]

Opcije:

- **-g** : Ispišite samo efektivni ID grupe.
- **-G** : Ispis svih ID-ova grupe.
- **-n** : Ispisuje ime umjesto broja.
- **-r** : Ispisuje stvarni ID umjesto brojeva.
- **-u** : Ispisuje samo stvarni korisnički ID.
- **-Pomozite** : Prikažite poruke pomoći i izađite.
- **-verzija** : Prikažite podatke o verziji i izađite.

Napomena: Bez ikakve OPCIJE štampa se svaki skup identifikovanih podataka, pa na primer. Numerički ID-i.

Primeri:

1. Da biste odštampali sopstveni ID bez ikakvih opcija:

Znauku

Izlaz prikazuje ID trenutnog korisničkog UID-a i GID-a.

1. Pronađite određeni korisnički ID: Sada pretpostavimo da imamo korisnika sa masterom imena, da bismo pronašli njegov UID koristićemo komandu:

id -u master

1. Nabavite GID određenog korisnika: Opet pod pretpostavkom da pronađemo Master GID, upotrebićemo komandu:

id -g master

1. Poznavanje UID-a i svih grupa povezanih sa korisničkim imenom: U ovom slučaju, koristićemo korisnički "master" da pronađemo UID i sve grupe povezane sa njim, upotrebimo komandu:

id majstor

1. Da biste saznali sve grupe kojima korisnik pripada: Pogledajte UID i sve grupe kojima korisnik "master" pripada:

id -G master

giff

Komanda "diff" se koristi za pronalaženje razlike između dve datoteke. Ova komanda analizira datoteke i štampa linije koje nisu slične. Recimo da imamo dva testa i test1. Razliku između dve datoteke možete pronaći pomoću sledeće komande.

Sintaksa -

```
razl test.txt test1.txt
```

Historycommand se koristi za prikazivanje prethodno izvršene komande. Ova funkcija nije bila dostupna u Born ljusti. Baš i Korn podržavaju ovu funkciju u kojoj se svaka izvedena komanda tretira kao događaj i povezana je sa brojem događaja kojim se mogu opozvati i promeniti ako je potrebno. Ove komande su sačuvane u datoteci istorije. U ljusti istorije bash , komanda prikazuje celu listu komandi.

Sintaksa:

```
$ history
```

Da biste prikazali ograničeni broj naredbi koje su se prethodno izvršavale, kako sledi:

```
$ hystory 10
```

dd

dd je uslužni program komandne linije za operativne sisteme nalik unix i Unix-u čija je primarna svrha konvertovanje i kopiranje datoteka.

1. Na Unix-u, upravljački programi hardverskih uređaja (kao što su čvrsti diskovi) i posebne datoteke uređaja (kao što su /dev/zero i/dev/random) pojavljuju se u sistemu datoteka baš kao i uobičajene datoteke.
2. dd takođe može da čita i/ili piše iz/u ove datoteke, pod uslovom da se funkcija implementira u njihovim upravljačkim programima
3. Kao rezultat toga, dd se može koristiti za zadatke kao što su pravljenje rezervne kopije sistema za pokretanje čvrstog diska i dobijanje fiksne količine nasumičnih podataka.
4. Dd programa takođe može da izvrši konverziju podataka dok se kopira, uključujući zamenu redosleda bajtova i konvertovanje u ASCII i EBCDIC kodiranje teksta.

Upotreba: Sintaksa dd komandne linije razlikuje se od mnogih drugih Unix programa po tome što koristi *opciju* = sintaksa vrednosti za opcije komandne linije, a ne za opcije standardne *vrednosti ili -Opcija* = formati vrednosti. Dd podrazumevano čita iz stdina i piše u odeljenju, ali ih je moguće promeniti pomoću opcija if (ulazna datoteka) i (izlazna datoteka).

Neki praktični primeri dd komande:

- 1. Za izradu sigurnosne kopije cijelog tvrdog diska:** Za sigurnosno kopiranje cijele kopije tvrdog diska na drugi tvrdi disk spojen na isti sustav, izvršite naredbu dd kao što je prikazano. U ovom primjeru naredbe dd naziv UNIX uređaja izvornog tvrdog diska je / dev / hda, a naziv ciljnog tvrdog diska / dev / hdb.
- 2. # dd ako = / dev / sda od = / dev / sdb**
- 3. "if" predstavlja ulaznu datoteku i "od" predstavlja izlaznu datoteku. Dakle, tačna kopija /dev/sda će biti dostupna u /dev/sdb .**
- 4. Ako postoje greške, gorenavedena komanda neće uspeti. Ako parametru date "Conv. = Noerror", on će i dalje biti kopiran ako postoje greške u čitanju.**

find

Komanda "Pronađi" u UNIX-u je uslužni program komandne linije za šetnju kroz hijerarhiju datoteka. Može se koristiti za pronalaženje datoteka i direktorijuma i izvršavanje narednih operacija na njima. Podržava pretraživanje po datoteci, fascikli, imenu, datumu kreiranja, datumu izmene, vlasniku i dozvolama. Korišćenjem "-exec" u pronađenim datotekama ili fasciklama, mogu se izvršiti druge UNIX komande.

Sintaksa:

```
$ find [odakle početi pretraživati]  
[izraz određuje što pronaći] [-opcije] [što pronaći]
```

Opcije:

1. **-exec CMD: Datoteka koja** se pretražuje zadovoljava gorenavedene kriterijume i vraća 0 kao status izlaza za uspešno izvršavanje komande.
2. **-ok CMD:** Radi isto što i -exec, osim što se prvi put traži od korisnika.
3. **-inum N;** Potražite datoteke sa inode brojem 'N'.
4. **-linkovi N:** Tražite fajlove sa linkovima 'N'.

U LINUX-u postoji uslužni program komandne linije za ovo i to je slobodna **komanda** koja prikazuje ukupnu količinu slobodnog prostora koja je dostupna zajedno sa količinom iskorišćene memorije i zamenom memorije u sistemu, kao i baferima koje koristi jezgro.

To je ono što slobodna komanda radi za tebe.

Sintaksa:

\$free [OPCIJA]

OPCIJA: odnosi se na opcije kompatibilne sa besplatnom naredbom.

Kako free prikazuje detalje memorije povezane s vašim sustavom, za njegovu sintaksu nisu potrebni nikakvi argumenti već samo opcije koje možete koristiti prema svojoj želji.

\$free [OPCIJA]

Opcija: Odnosi se na opcije kompatibilne sa slobodnom komandom.

Kako besplatno prikazuje detalje memorije povezane sa sistemom, njegova sintaksa ne zahteva argumente, već samo opcije koje možete da koristite po vašem ukusu.

Korišćenje besplatnih komandi

Slobodnu komandu možete da koristite kao: *\$free*

/ * slobodna komanda bez ikakve

opcija prikazuje iskorišćenu

I slobodan prostor za zamenu.

i fizičke memorije u **KB** * /

Kada se nijedna od opcija ne koristi, slobodna komanda daje izlaz u koloni kao što je prikazano iznad mesta gde je kolona:

- 1. ukupni prikazi** ukupne instalirane memorije (MemTotal i SwapTotal *su* prisutni u / proc / meminfo).
- 2. korišćene ekrane** iskorišćene memorije.
- 3. besplatno prikazuje** neiskorišćenu memoriju.
- 4. deljeni prikazi** memorije koje koriste tmpfovi (Shmen je *prisutan* u /proc/meminfo i prikazuje nulu ako nije dostupan).
- 5. prikaži mi bafer** memorije koji koriste osnovni baferi.
- 6. prikazuje memorijsku keš memoriju** koju koriste keš stranice i table (keširane i table dostupne u /proc/meminfo).
- 7. bafer/keš** prikazuje zbir bafera i keša.

ssh-keygen

Koristite komandu ssh-keygen da biste generisali par ključeva za javnu/privatnu potvrdu identiteta. Tasteri za potvrdu identiteta omogućavaju korisniku da se poveže sa udaljenim sistemom bez davanja lozinke. Ključevi moraju biti generisani za svakog korisnika posebno. Ako generišete parove ključeva kao osnovni korisnik, samo koren može da koristi tastere.

Sledeći primer kreira javni i privatni deo RSA ključa:

```
ssh-keygen -t rsa
```

Koristite opciju `-t` da biste naveli tip ključa koji želite da kreirate. Moguće vrednosti su `"rsa1"` Za protokol verzije 1 i `'dsa'`, `'ecdsa'`, ili `'rsa'` Za protokol verzije 2.

Imate mogućnost da navedete lozinku za šifrovanje privatnog dela ključa. Ako šifrujete lični ključ, morate da unesete lozinku svaki put kada koristite ključ. Na taj način se sprečava napadač koji ima pristup vašem privatnom ključu i kome možete pogrešno da se predstavite i da mu pristupite na svim računarima kojima imate pristup. Napadač još uvek mora da unese lozinku.

ip

Ip komanda u Linuxu je prisutna u mrežnim alatima koji se koriste za obavljanje nekoliko mrežnih administrativnih zadataka. Ova komanda se koristi za prikazivanje ili upravljanje usmeravanjem, uređajima i tunelima. Ova komanda se koristi za izvršavanje nekoliko zadataka, kao što je dodeljivanje adrese mrežnom interfejsu ili konfigurisanje parametara mrežnog interfejsa. On može da izvrši nekoliko drugih zadataka kao što su konfigurisanje i menjanje podrazumevanog i statičnog usmeravanja, podešavanje tunela preko IP-a, listanje IP adresa i podataka o svojstvima, promena statusa interfejsa, dodeljivanje, brisanje i postavljanje IP adresa i usmeravanja.

Sintaksa:

ip [OPTIONS] OBJECT pomoć

Opcije:

-adresa: Ova opcija se koristi za prikazivanje svih IP adresa povezanih sa svim mrežnim uređajima.

IP adresa

-link: Koji se koristi za prikazivanje informacija o sloju veze, preuzeće karakteristike trenutno dostupnih uređaja sloja veze. Svaki mrežni uređaj koji ima učitani upravljački program može se klasifikovati kao dostupan uređaj.

ip veza

nslookup

Nslookup (skraćeno od "Traženje servera imena") je korisna komanda za dobijanje informacija sa DNS servera. To je alatka za mrežnu administraciju za izvršavanje upita nad sistemom imena domena (DNS) da biste dobili ime domena ili mapiranje IP adrese ili bilo koji drugi određeni DNS zapis. Takođe se koristi za rešavanje problema vezanih za DNS.

Sintaksa:

nslookup [opcija]

Mogućnosti nslookup naredba:

- *nslookup google.com:*

nslookup praćen imenom domena će prikazati 'zapis' (IP adresa) domena. Koristite ovu komandu da biste pronašli zapise adresa za svoj domen. Izvršava upit nad serverima imena domena i dobija detalje.

curl

curl je alatka komandne linije za prenos podataka na server ili sa njega, koristeći bilo koji od podržanih protokola (HTTP, FTP, IMAP, POP3, SCP, SFTP, SMTP, TFTP, TELNET, LDAP ili FILE). Ovu komandu vodi Libcurl. Ova alatka je poželjna za automatizaciju jer je dizajnirana da radi bez interakcije korisnika. Može da prenese više datoteka odjednom.

Sintaksa:

uvijanje [opcije] [URL ...]

Najosnovnija upotreba curla je upisivanje naredbe nakon koje slijedi URL.

curl https://www.python.org

-o: sprema preuzetu datoteku na lokalni stroj s imenom navedenim u parametrima.

Sintaksa:

curl -o [ime_datoteke] [URL ...]

Primer:

curl -o hello.zip ftp://speedtest.tele2.net/1MB.zip

tr

Tr u UNIX komandi je uslužni program komandne linije za prevođenje ili brisanje znakova. On podržava niz transformacija, uključujući velike do male, stiskajuće ponavljajuće znakove, brisanje određenih znakova i osnovno pronalaženje i zamenu. Može se koristiti sa UNIX cevima za podršku složenijim prevodima. **Tr označava prevod.**

Sintaksa:

```
$ tr [zastavica] SET1 [SET2]
```

Opcije

-c: dopunjuje skup znakova u nisci, pa naime. operacije se primenjuju na znakove koji nisu u datom skupu-d: brisanje znakova iz prvog skupa iz rezultata.-s: zamenjuje ponovljene znakove navedene u skupu1 jednim pojavljivanjem-t: skraćuje set1

Probne komande

- 1. Kako da konvertujete malim s malim sličnim sličima**To konvertujte iz malim sličnim snimova u velika, unapred definisana skupa u tr.

iptables

Iptable je interfejs komandne linije koji se koristi za podešavanje i održavanje tabela za Netfilter zaštitni zid za IPv4, uključen u Linux jezgro. Zaštitni zid se podudara sa paketima sa pravilima definisanim u ovim tabelama, a zatim preduzima određene radnje na mogućem podudaranju.

1. *Tabele* su naziv za skup lanaca.
2. *Lanac* je skup pravila.
3. *Pravilo* je uslov koji se koristi za podudaranje sa paketom.
4. *Ciljaj* da li je preduzeta radnja kada se moguće pravilo podudara. Primeri cilja su ACCEPT, DROP, QUEUE.
5. *Smernice* su podrazumevana radnja preduzeta u slučaju da se ne podudaraju sa ugrađenim lancima i da mogu biti PRIHVATLJIVE ili DROP.

Sintaksa:

```
iptables --table TABLICA -A / -C / -D ... LANČANO pravilo --jump Target
```

convenient-get

convenient-get je alat komandne linije koji pomaže u rukovanju paketima u Linuxu. Njen glavni zadatak je da preuzme informacije i pakete iz proverenih izvora kako bi instalirao, nadogradio i uklonio pakete zajedno sa njihovim zavisnostima. Ovde APT označava naprednu alatku za *pakovanje* .

Sintaksa:

naredba apt-get [options]

Update: Ova komanda se koristi za ponovno indeksiranje datoteka indeksa paketa iz njihovih izvora. Morate izvršiti ispravku pre nadogradnje.

apt-get update

df, ti

Komanda Df (*ne-disk*) izveštava o količini raspoloživog prostora na disku koji koriste sistemi datoteka. Komanda du (*korišćenje diska*) izveštava o veličinama stabala direktorijuma, uključujući sav njihov sadržaj i veličine pojedinačnih datoteka.

Cilj je da se osigura da ne pređete cenzus od 80 odsto. Ako pređete prag, vreme je da skalirate ili počistite nered, jer resursi ne mogu da se ukrste, a promena u aplikaciji pokazuje neko vrcavo ponašanje.

Da biste proverili čitljiv format:

```
$ sudo df -h
```

Međutim, u većini slučajeva želite da proverite koji deo sistema zauzima mnogo prostora na disku. Koristi sledeću komandu:

```
$ sudo du -h -d 1 / var /
```

htop

htop komanda u Linuxu je uslužni program komandne linije koji omogućava korisniku da interaktivno prati vitalne resurse sistema ili procesa servera u realnom vremenu. Ovo je novije vreme u poređenju sa najvišom komandom i nudi mnoga poboljšanja u odnosu na gornju komandu. Podržava rad miša, koristi izlaz u boji i daje vizuelne indikacije korišćenja cpua, memorije i zamene. htop takođe štampa pune komandne linije za procese i omogućava pomeranje vertikalno i horizontalno za procese, pa na primer komandne linije.

Sintaksa-

htop

1. **-d –odlaganje:** Koristi se za prikazivanje kašnjenja između ispravki, u desetinama sekundi.
2. **-C –bezbojan – bezbojan :** Pokrenite htop u monohromatnom režimu.
3. **-h –Pomoć:** Koristi se za prikazivanje poruke pomoći i izlazak.
4. **-u –user = USERNAME:** Koristi se za prikazivanje samo procesa određenog korisnika.

P.S.

Svaki proces u Linuxu ima jedinstven ID i može se videti pomoću ps komande.

1. *\$ sudo ps aux*

2. **do** = prikaži procese za sve korisnike

3. **u** = prikaži korisnika / vlasnika procedure

4. **x** = takođe prikazuje procese koji nisu povezani sa terminalom

kill

Kill command in Linux (nalazi se u `/bin/ kill`), je ugrađena komanda koja se koristi za ručno prekidanje procesa. Ova komanda šalje signal procesu koji dovrši proces. Ako korisnik ne navede signal koji će biti poslat zajedno sa komandom za ubijanje, onda se podrazumevanom *TERM-u šalje signal koji dovršavanje procesa.*

kill -l : **Da** biste prikazali sve dostupne signale, možete koristiti dole navedenu opciju komande:

Sintaksa: *\$kill -l*

1. Negativne PID vrednosti se koriste za označavanje ID-a grupe procesa. Ako prosledite ID grupe procesa, svi procesi u toj grupi će primiti signal.
 2. PID od -1 je veoma poseban jer označava sve procese, osim kill and init, što je roditeljski proces svih procesa u sistemu.
- Da biste prikazali listu izvršenih procesa, koristite *komandu p.s* i to će vam pokazati pokrenute procese sa njihovim PID brojem. Da bismo utvrdili koja procedura treba da primi signal ubistva, moramo da obezbedimo PID.

Sintaksa:

```
$ ps
```

Da pokažem kako se koristi *pid* sa komandom za ubijanje.

Sintaksa:

```
$ kill pid
```

telnet

Telnet pomaže u -

1. povezivanje sa udaljenim Linux računarom
2. daljinsko pokretanje programa i primena administracije

Sintaksa

1. telnet hostname = "" ili =""
2. Primer:
3. *telnet localhost*

Izvršavanje skripti ljuske

Šta je Šel?

Operativni sistem sadrži mnogo komponenti, ali dve glavne komponente su "Jezgro" i "Ljuska".

Jezgro možete da mislite kao jezgro računara. Omogućava komunikaciju između hardvera i softvera. Jezgro je najdublji deo operativnog sistema, dok je školjka najdalje.

Ljuska u Linux operativnom sistemu uzima unos od korisnika u obliku komandi, obrađuje ga, a zatim daje izlaz. On deluje kao interfejs preko kojeg korisnik radi na programima, komandama i skriptama. Terminal pristupa ljusci i takođe izvršava komande.

Kada je terminal pokrenut, Shell izdaje komandnu liniju (obično \$) u koju je moguće otkucati njegov unos, nakon čega ga terminal izvršava kada pritisnete taster Enter. Terminal zatim prikazuje izlaz komandi.

Školjka se završava kao delikatna unutrašnjost operativnog sistema koji je štiti od slučajnog oštećenja. Stoga, ime je Shell.

Postoje dve glavne školjke u Linuxu:

1. Školjka Bourne : Upit za ovu ljusku je \$, a njezini derivati su sljedeći:

- POSIX ljuska također je poznata kao sh
- Korn Shell također je znao kao sh
- Bourne Again SHell poznat je i pod nazivom bash (najpopularniji)

2. C ljuska: % označava upit za ovu ljusku, a njegove potkategorije su kako slijedi:

- C ljuska je također poznata i kao csh
- Ljuska vrha C također je poznata i kao tcsh

Šta je Shell Scripting?

Skriptiranje ljuske piše niz izvršnih komandi ljuske. Može da kombinuje dugačke i ponavljajuće komandne sekvence u jednu i jednostavnu skriptu. Ovu skriptu možete uskladištiti i izvršiti kad god poželite. Na taj način se značajno smanjuje napor koji zahteva krajnji korisnik.

Slede koraci za kreiranje skripte ljuske -

- Kreiranje datoteke pomoću uređivača teksta kao što ste vi ili bilo koji drugi uređivač. Nazovite datoteku skripte s nastavkom `.sh`
- Pokrenite skriptu s `#!/bin/sh`
- Napišite neki kod.
- Spremite datoteku skripte kao `filename.sh`
- Za izvršavanje skripte upišite `bash filename.sh`

'#!' je operater koji se zove shebang koji usmerava scenario na lokaciju prevodioca. Dakle, ako koristimo "#! / bin / sh "skripta upućuje na bourne-shell.

Sada ćemo stvoriti datoteku pomoću uređivača poput vi i spremiti je s .sh nastavkom. Kopirajte sljedeći program koji dodaje i ispisi zbroj znamenki broja koji je korisnik unio. Zatim pokrenite ovaj program pomoću naredbe bash filename.sh.

```
#! / bin / sh

echo 'Unesite broj'
čitaj Num
g = $ cb

# pohrani zbroj
# znamenke
s = 0

# koristi while petlju za
# izračunajte zbroj
# svih znamenki
dok je [$ Num -gt 0]
čini
# dobiti ostatak
k = $ (($ num% 10))

# dobiti sljedeću znamenku
Num = $ (($ Num / 10))

# izračunaj zbroj
# broj
s = $ (($ s + $ k))

gotovo
echo 'zbroj znamenki od $ g je: $ s'
```

Git naredbe

Što je Git?

Git je besplatan, distribuirani sistem kontrole open source verzije. Ovaj alat upravlja svime, od malih do veoma velikih projekata sa brzinom i efikasnošću. Linus Torvalds ga je stvorio 2005. Git ima funkcionalnost, performanse, bezbednost i fleksibilnost koja je potrebna većini timova i pojedinačnih programera.

Alatke kao što je Git omogućavaju komunikaciju između tima za razvoj i operacije. Kada razvijate veliki projekat sa ogromnim brojem saradnika, veoma je važno da imate komunikaciju između saradnika dok pravite izmene u projektu. Poruke predavanja u Gitu igraju veoma važnu ulogu u komunikaciji između tima. Delovi koje smo svi postavili leže u sistemu za kontrolu verzija kao što je Git. Da biste uspeeli u programu DevOps, morate imati svu komunikaciju u kontrolnoj verziji. Stoga, Git igra važnu ulogu u uspehu na DevOps-u.

Git naredbe

git init

Upotreba : `git init [ime depoa]`

Ova komanda kreira novo skladište.

git config

Upotreba : `git config --global user.name "[ime]"`

Upotreba : `git config --global user.email "[adresa e-pošte]"`

Ova komanda postavlja ime autora i e-adresu. Ovo su korisne informacije sa obavezama.

Git Klon

Upotreba : `git clone [url]`

Ova komanda vam omogućava da nabavite kopiju skladišta iz postojeće URL adrese.

git dodaj

Upotreba: `git add [datoteka]`

Ova naredba dodaje datoteku u scensko područje.

Upotreba: `git dodaj *`

Ova naredba dodaje jedno ili više mjesta za postavljanje.

git poćiniti

Upotreba: *git commit -m "[Type commit message]"*

Ova komanda trajno upisuje ili zapisuje datoteku u istoriju verzija.

Upotreba: *putty commit -a*

Ova komanda obavezuje sve datoteke koje ste dodali pomoću komande "git add", kao i datoteke koje ste promenili od tada.

status git

Upotreba: *git status*

Git statuscommand prikazuje stanje radnog direktorijuma i administrativnu oblast. Ova komanda vam omogućava da vidite promene koje su u stepenu, one koje nisu inscenirane i koje Git ne prati.

git show

Upotreba: *git show [predaj]*

Ova komanda prikazuje metapodatke i promene sadržaja u navedenom commitu.

go rm

Upotreba: *git rm [ypedu]*

Ova komanda briše datoteku iz vašeg radnog direktorijuma i postepeno je briše.

Daljinski upravljač za Git

Upotreba: *git daljinsko dodavanje [ime promenljive] [Veza sa udaljenim serverom]*

Ova komanda povezuje lokalno skladište sa udaljenim serverom.

git push

Upotreba: *git push [promenljivo ime] master*

Ova komanda šalje prosleđene promene u glavnu granu u udaljeno skladište.

Upotreba: *git push [promenljivo ime] [grana]*

Ova komanda šalje obavezu grana na udaljeno skladište.

Upotreba: *git push - sve [promenljivo ime]*

Ova komanda gura sve grane u vaše udaljeno skladište.

Upotreba: *git push [promenljivo ime]: [ime grane]*

Ova komanda briše granu na udaljenom skladištu.

git pull

Upotreba: *git pull [Link ka depou]*

git grit

Upotreba: *putty branch*

Ova komanda navodi sve lokalne ogranke u trenutnom skladištu.

Upotreba: *git grana [naziv grane]*

Ova komanda kreira novu granu.

Upotreba: *git grana -d [naziv grane]*

Ova komanda briše granu funkcije.

git naplata

Upotreba: *git odjava [ime grane]*

Ova komanda vam omogućava da se prebacite sa jedne grane na drugu.

Upotreba: *git checkout -b [naziv grane]*

Ova komanda kreira novu granu i takođe se prebacuje na nju.

idi objedini

Upotreba: *git join [ime grane]*

Ova komanda objedinjuje navedenu istoriju grane u trenutnu granu.

Git Rebase

Upotreba: *git rebase [ime grane]*

git rebase master - Ova komanda će premestiti sav naš rad sa sadašnje grane na glavnu.

Ovo nas dovodi do kraja bloga o Linux komandama u DevOps-u. Ovde sam pokušao da obuhvatim što više komandi. Ovaj blog će vam definitivno pomoći da započnete putovanje sa DevOps-om

