

Learn 2025 DevOps/SRE tech stack www.heydevops.in

Our Placements <https://heydevops.in/success-stories>

More Project Videos <https://www.youtube.com/praveensingampalli>

Connect with me <https://www.instagram.com/singam4devops/>

ULTIMATE CHEAT SHEETS / ARCHITECTURE DESIGNS / KUBERNETES NOTES

Git Cheat Sheet

Setup

Set the name and email that will be attached to your commits and tags

```
$ git config --global  
user.name "Danny Adams"  
$ git config --global  
user.email "my-  
email@gmail.com"
```

Start a Project

Create a local repo (omit <directory> to initialise the current directory as a git repo)

```
$ git init <directory>  
Download a remote repo  
$ git clone <url>
```

Make a Change

Add a file to staging

```
$ git add <file>
```

Stage all files

```
$ git add .
```

Commit all staged files to git

```
$ git commit -m "commit  
message"
```

Add all changes made to tracked files & commit

```
$ git commit -am "commit  
message"
```

Basic Concepts

main: default development branch
origin: default upstream repo
HEAD: current branch
HEAD²: parent of HEAD
HEAD⁴: great-great grandparent of HEAD

By @DoableDanny

Branches

List all local branches. Add -r flag to show all remote branches. -a flag for all branches.

```
$ git branch
```

Create a new branch

```
$ git branch <new-branch>
```

Switch to a branch & update the working directory

```
$ git checkout <branch>
```

Create a new branch and switch to it

```
$ git checkout -b <new-  
branch>
```

Delete a merged branch

```
$ git branch -d <branch>
```

Delete a branch, whether merged or not

```
$ git branch -D <branch>
```

Add a tag to current commit (often used for new version releases)

```
$ git tag <tag-name>
```

Merging

Merge branch a into branch b. Add --no-ff option for no-fast-forward merge



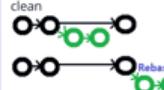
```
$ git checkout b  
$ git merge a
```

Merge & squash all commits into one new commit

```
$ git merge --squash a
```

Rebasing

Rebase feature branch onto main (to incorporate new changes made to main). Prevents unnecessary merge commits into feature, keeping history clean



```
$ git checkout feature
```

```
$ git rebase main
```

Interactively clean up a branches commits before rebasing onto main

```
$ git rebase -i main
```

Interactively rebase the last 3 commits on current branch

```
$ git rebase -i Head~3
```

Undoing Things

Move (&/or rename) a file & stage move

```
$ git mv <existing_path>  
<new_path>
```

Remove a file from working directory & staging area, then stage the removal

```
$ git rm <file>
```

Remove from staging area only

```
$ git rm --cached <file>
```

View a previous commit (READ only)

```
$ git checkout <commit_ID>
```

Create a new commit, reverting the changes from a specified commit

```
$ git revert <commit_ID>
```

Go back to a previous commit & delete all commits ahead of it (revert is safer). Add -hard flag to also delete workspace changes (BE VERY CAREFUL)

```
$ git reset <commit_ID>
```

Review your Repo

List new or modified files not yet committed

```
$ git status
```

List commit history, with respective IDs

```
$ git log --oneline
```

Show changes to unstaged files. For changes to staged files, add --cached option

```
$ git diff
```

Show changes between two commits

```
$ git diff commit1_ID  
commit2_ID
```

Stashing

Store modified & staged changes. To include untracked files, add -u flag. For untracked & ignored files, add -a flag.

```
$ git stash
```

As above, but add a comment.

```
$ git stash save "comment"
```

Partial stash. Stash just a single file, a collection of files, or individual changes from within files

```
$ git rm <file>
```

Remove from staging area only

```
$ git rm --cached <file>
```

View a previous commit (READ only)

```
$ git checkout <commit_ID>
```

Create a new commit, reverting the changes from a specified commit

```
$ git revert <commit_ID>
```

Go back to a previous commit & delete all commits ahead of it (revert is safer). Add -hard flag to also delete workspace changes (BE VERY CAREFUL)

```
$ git reset <commit_ID>
```

Re-apply the stash without deleting it

```
$ git stash apply
```

Re-apply the stash at index 2, then delete it from the stash list. Omit stash@{n} to pop the most recent stash.

```
$ git stash pop stash@{2}
```

Show the diff summary of stash 1. Pass the -p flag to see the full diff.

```
$ git stash show stash@{1}
```

Delete stash at index 1. Omit stash@{n} to delete last stash made

```
$ git stash drop stash@{1}
```

Delete all stashes

```
$ git stash clear
```

Synchronizing

Add a remote repo

```
$ git remote add <alias>  
<url>
```

View all remote connections. Add -v flag to view urls.

```
$ git remote
```

Remove a connection

```
$ git remote remove <alias>
```

Rename a connection

```
$ git remote rename <old>  
<new>
```

Fetch all branches from remote repo (no merge)

```
$ git fetch <alias>
```

Fetch a specific branch

```
$ git fetch <alias> <branch>
```

Fetch the remote repo's copy of the current branch, then merge

```
$ git pull
```

Move (rebase) your local changes onto the top of new changes made to the remote repo (for clean, linear history)

```
$ git pull --rebase <alias>
```

Upload local content to remote repo

```
$ git push <alias>
```

Upload to a branch (can then pull request)

```
$ git push <alias> <branch>
```

Learn 2025 DevOps/SRE tech stack www.heydevops.in

Our Placements <https://heydevops.in/success-stories>

More Project Videos <https://www.youtube.com/praveensingampalli>

Connect with me <https://www.instagram.com/singam4devops/>

Linux Commands Cheatsheet

Basic File Operations		Directory Traversal		Process Management		User Management	
<code>ls -lh file</code>	Display file permissions, size, owner etc	<code>cd or cd ~</code>	Navigate to the user's home directory	<code>ps</code>	Display a snapshot of running processes	<code>id</code>	Displays the user's UID, GID, and groups
<code>touch file</code>	Creates an empty file	<code>cd ..</code>	Navigate to the parent directory	<code>ps aux</code>	Display all processes of all users	<code>whoami</code>	Displays who is currently logged in
<code>cp file1 file2</code>	Copy file1 to file2. File2 can be a directory	<code>cd -</code>	Switch to the previous working directory	<code>finger traw</code>	Print information about user traw	<code>finger</code>	
<code>mv file dir</code>	Move a file to a directory	<code>cd /</code>	Navigate to the root directory	<code>useradd -u 1002 traw</code>	Creates a new user traw with a specific UID	<code>useradd</code>	
<code>mv file1 file2</code>	Rename file1 to file2	<code>cd /tmp</code>	Changes the current directory to /tmp	<code>userdel traw</code>	Deletes the user account named traw	<code>chfn traw</code>	Change a user's personal information
<code>rm file</code>	Delete a file			<code>kill 6732</code>	Terminate a process with PID of 6732	<code>chfn -s bin/zsh</code>	add user traw to the sudo group
<code>ls -lah</code>	List all the contents in a directory			<code>killall proc</code>	Kills all processes named 'proc'	<code>chsh -s /bin/zsh</code>	add user traw to the sudo group
<code>mkdir data</code>	Creates a directory data	<code>df -h</code>	Report file system disk space usage	<code>pidof firefox</code>	Find the process ID of firefox	<code>passwd</code>	Change user password
<code>cp -r dir1 dir2</code>	Copy dir1 and its contents to dir2	<code>du -h /home</code>	Estimate file space usage in the home dir	<code>killall firefox</code>	Find the process IDof firefox	<code>passwd traw</code>	Change user traw's password
<code>rm -rf dir</code>	Delete a directory and its contents	<code>fdisk -l</code>	List available partitions on a disk	<code>bg</code>	Resumes suspended jobs in the background	<code>chsh -s /bin/zsh</code>	Change user shell to zsh
<code>pwd</code>	Print current working directory	<code>fdisk</code>	Create partitions	<code>fg</code>	Brings a suspended job to foreground	<code>su jamee</code>	Switch to user james
<code>stat file</code>	Display attributes of files or directories	<code>fsblk</code>	List block devices	<code>jobs</code>	List active jobs in the current shell		
<code>wc file</code>	Count bytes, words, lines in a file or STDIN	<code>mount /dev/sda /mnt</code>	Mount /dev/sda partition to /mnt directory	<code>renice 12 PID</code>	Changes priority of process with given PID		
<code>file file</code>	Identify (guess) the type of a file.	<code>findmnt</code>	Displays if about all mounted filesystems	<code>pstree</code>	Displays a tree of running processes		
<code>type cd</code>	Find out whether cd binary is built-in , alias or external binary file	<code>fsck /dev/sda</code>	Check a disk partition for errors				
File Viewing		Disk Management		Networking		Access Control Lists	
<code>cat file.txt</code>	Print the contents of a text file	<code>df -h</code>	Report file system disk space usage	<code>ping sysxplore.com</code>	Sends ICMP packets to sysxplore.com	<code>getfacl file</code>	Display ACL permissions of a file or directory
<code>tac file.txt</code>	Prints text in reverse	<code>du -h /home</code>	Estimate file space usage in the home dir	<code>ip addr</code>	Displays all network interfaces information	<code>setfacl -m u:traw:r-x file</code>	Set read/execute ACL perms for the user traw
<code>more file.txt</code>	View large text files one page at a time	<code>fdisk -l</code>	List available partitions on a disk	<code>ifconfig</code>	Shows network interfaces configuration	<code>setfacl -m g:sysops=rw file</code>	Set read/write ACL perms for the group sysops
<code>less data.txt</code>	Same as more but with more features	<code>fdisk</code>	Create partitions	<code>whosys sysxplore.com</code>	Displays domain's registration information	<code>setfacl -x u:traw file</code>	Remove user traw ACL permissions
<code>head -n 5</code>	View the first 5 line of a text file	<code>fsblk</code>	List block devices	<code>route</code>	Display the routing table	<code>setfacl -x u:udevops file</code>	Remove the group devops ACL perms
<code>tail -n 5</code>	View the last 5 line of a text file	<code>mount /dev/sda /mnt</code>	Mount /dev/sda partition to /mnt directory	<code>ss</code>	Display information about network sockets	<code>setfacl -b file</code>	Remove all ACL perms and keep default file permissions
<code>nl file</code>	View text files with their lines numbered	<code>findmnt</code>	Displays if about all mounted filesystems	<code>netstat</code>	Displays network information and statistics		
<code>strings file</code>	Display text that's embedded in a binary file	<code>fsck /dev/sda</code>	Check a disk partition for errors	<code>dig sysxplore.com</code>	Queries DNS, provides domain's DNS info		
Print Text		File Permissions		Compression/Archives		File Transfer	
<code>echo "Hello World"</code>	Print Hello World on the standard output	<code>chmod +x</code>	Set execute permissions to a file	<code>tar -cf backup.tar /home</code>	Creates a tar archive of /home dir	<code>scp file.txt user@rhost:/remote/dir</code>	Copies file.txt to remote host's specified directory
<code>printf "%5d\n" 42</code>	Print formatted text on standard output	<code>chmod u+s script.sh</code>	Set SUID permissions to a file	<code>tar -xf backup.tar</code>	Extract files from "backup.tar" archive	<code>rsync -a /ubuntu /backup/</code>	Synchronizes content from source directory to destination directory, preserving attributes
<code>yes "Hello World"</code>	Print repeated text on the standard output	<code>chmod g+s dir</code>	Set SGID permissions to a directory	<code>tar -zcvf data.tar.gz /home</code>	Creates compressed archive of /home	<code>rsync -a /var/www/web/ user@rhost:/data/backup/</code>	Synchronizes local directory to remote, preserving attributes
<code>seq 1 5</code>	Print a sequence of numbers from 1 to 5	<code>chmod +t dir</code>	Set Sticky Bit permissions to a directory	<code>gunzip data.gz</code>	Uncompress data.gz file		
<code>clear</code>	Clear the terminal screen or window	<code>chgrp devops file.txt</code>	Changes file.txt group owner to devops	<code>zip <dir></code>	Zip the data directory		
		<code>chmod 644 script.sh</code>	Set the file perms to be read/write for the owner, and read-only for group and others	<code>unzip data.zip</code>	Unzip the data.zip file		
		<code>chown traw:sys file</code>	Changes file owner to traw and group owner to sys	<code>gzip data</code>	Compresses "data" into "data.gz", original is removed		
File Search		History		User Group Management		Text Manipulation	
<code>locate file</code>	Searches for files and directories	<code>sudo !!</code>	Execute the previous command with sudo	<code>groups</code>	Print the group membership of a user	<code>grep "linux" file.txt</code>	Search for the word Linux in file.txt
<code>which cd</code>	Searches the location of the cd binary	<code>*cat*`cat</code>	Replace previous cmd with tac	<code>groupadd devops</code>	Create a new group called devops	<code>tr 'a-z' 'A-Z' <file></code>	Translate lowercase chars to uppercase
<code>whereis ls</code>	Find ls' binary docs, and source files	<code>history</code>	Display command history	<code>groupdel devops</code>	Deletes the devops group	<code>rev <file></code>	Print file.txt contents in reverse
<code>find /data -name hello.txt</code>	Searches for "hello.txt" in the /data directory	<code>!S</code>	Last argument of the previous command	<code>groupmod -n sysops sys</code>	Changes sysops group name to sys	<code>sort <file></code>	Sort lines of text by various criteria
		<code>ISO</code>	Execute the 50th command in history			<code>uniq <file></code>	Print only unique lines in file.txt
						<code>vimdiff file1 file2</code>	Line-by-line comparison of two files in vim
						<code>diff <file> file2</code>	Comparison of two files on terminal
						<code>awk '{print \$1}' file.txt</code>	Print the first column of the file.txt
						<code>sed 's/cat/bat/g' file</code>	Substitute all cat occurrence with bat in file

	<h2>Top 50 Linux Commands you must know</h2>						
1. <code>ls</code>	11. <code>cat</code>	21. <code>diff</code>	31. <code>kill</code> and <code>killall</code>	41. <code>apt</code> , <code>pacman</code> , <code>yum</code> , <code>rpm</code>			
2. <code>pwd</code>	12. <code>echo</code>	22. <code>cmp</code>	32. <code>df</code>	42. <code>sudo</code>			
3. <code>cd</code>	13. <code>less</code>	23. <code>comm</code>	33. <code>mount</code>	43. <code>cal</code>			
4. <code>mkdir</code>	14. <code>man</code>	24. <code>sort</code>	34. <code>chmod</code>	44. <code>alias</code>			
5. <code>mv</code>	15. <code>uname</code>	25. <code>export</code>	35. <code>chown</code>	45. <code>dd</code>			
6. <code>cp</code>	16. <code>whoami</code>	26. <code>zip</code>	36. <code>ifconfig</code>	46. <code>wheris</code>			
7. <code>rm</code>	17. <code>tar</code>	27. <code>unzip</code>	37. <code>traceroute</code>	47. <code>whatis</code>			
8. <code>touch</code>	18. <code>grep</code>	28. <code>ssh</code>	38. <code>wget</code>	48. <code>top</code>			
9. <code>ln</code>	19. <code>head</code>	29. <code>service</code>	39. <code>ufw</code>	49. <code>useradd</code>			
10. <code>clear</code>	20. <code>tail</code>	20. <code>ps</code>	40. <code>iptables</code>	50. <code>passwd</code>			

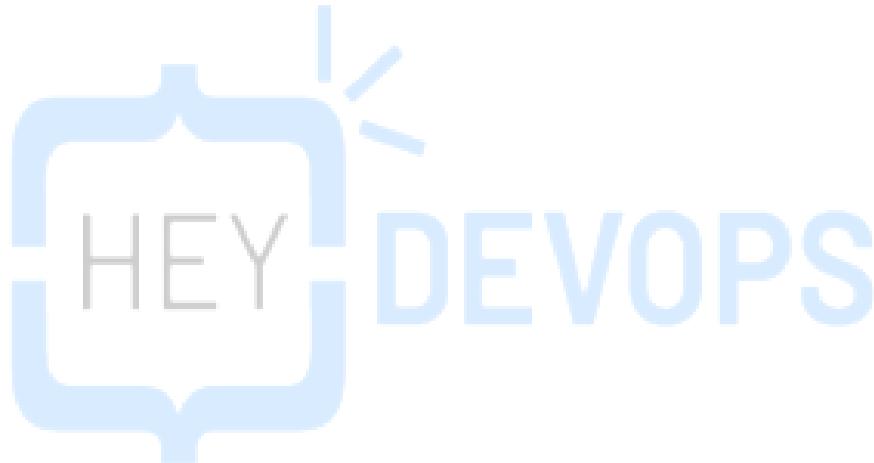
Learn 2025 DevOps/SRE tech stack www.heydevops.in

Our Placements <https://heydevops.in/success-stories>

More Project Videos <https://www.youtube.com/praveensingampalli>

Connect with me <https://www.instagram.com/singam4devops/>

Basic Commands	Networking Commands	Storage Related Commands	Compressing & Uncompressing Commands
<pre>ls Lists all files in the current directory ls -l Lists all files and directories with info about ownership, size, and date modified ls -t Sort the output by last modified date pwd List the path of the current directory cd directory/move/to Switch to a different working directory cd.. Move up one directory level cd / Switch to the root directory clear Clears all text from the terminal screen history Prints out a history of commands touch filename Create a new file vi filename Open a file for editing cat filename Display the contents of a file hostnamectl List system information timedatectl Displays network information date Display the system date & time top List all running processes free -m Display memory usage head filename Displays the first 10 lines of a file tail filename Displays the last 10 lines of a file mv file /newpath Move file to a new location mv filename new_name Rename a file cp filename new_name Copy a file man command_name List information about a command rm filename Delete a file rm -rf directory_name Remove a directory and its contents sudo Run commands with elevated privileges as a regular user mkdir directory_name Create a new directory kill pid Kill a process using its process ID reboot Restart the system shutdown -h now Shutdown the system</pre>	<pre>dig domain Display the DNS of a domain name dig +short Perform a reverse lookup for a host host domain Display the IP of a domain name whosdomain Extensive information for a domain ping ip Check connectivity to an IP address ssh username@ip Securely log into a server wget file Download a file wget -e file Continue a stopped download traceroute domain Trace the route a packet will take telnet domain port Connect to host via a specific port netstat -anlt Display all listening ports route Display the routing table for your machine arp View the contents of the address routing protocol (ARP) table cat /etc/resolv.conf View the DNS server your machine uses top -u www-data tail -n 1 Monitor all incoming traffic on port 80 nmap ip Network discovery for a given IP</pre>	<pre>df -h List storage information for all partitions mount Mount or unmount and ISO or drive umount Unmount or unmount and ISO or drive du -h /home/directory Display directory size in a readable format du -sh /home/directory Get the total size of a directory du -ah --exclude='xxx/' /home/directory Display the disk usage of all files in a directory excluding files with the specified exclusion du -ha -t /home/dir Display disk usage of a directory and list according to time modified fdisk -l Display disk size along with partition information sudo du -x / sort -nr head -20 List the top 20 directories that are over-consuming</pre>	<pre>tar -cvf archive.tar file Compress a file into a tar archive tar -tvf archive.tar Display the contents of a tar archive tar -xvf archive.tar Extract a single file out of a tar archive zip archive.zip file1.txt file2.html file3.jpg Create an archive using multiple files zip -u archive.zip file.txt Add a file to an already zipped file zip -d archive.zip file.txt Delete a file from an archive unzip archive.zip Unzip an archive unzip archive.zip -d dir Extract an archive to a specific directory tar xf archive Extract an archive of any type gzip -c file > archive.gz Create a new gz file. This will delete the original file gzip -c file > archive.gz Create a new gz file, keeping the original</pre>



Learn 2025 DevOps/SRE tech stack www.heydevops.in

Our Placements <https://heydevops.in/success-stories>

More Project Videos <https://www.youtube.com/praveensingampalli>

Connect with me <https://www.instagram.com/singam4devops/>

ifconfig	ip	traceroute	ping
Used to find network details, initialize an interface, assign IP address, enable or disable an interface.	Latest and more powerful version of ifconfig. The utility is used for displaying and manipulating routing, network devices, interfaces.	Network troubleshooting utility for tracing the full path/route of packet from your local system to another network system.	It is used to check the connectivity between two hosts/nodes on a Local Area Network or Wide Area Network. It makes use of the ICMPs to make communicate with end nodes.
netstat	ss	dig	route
Netstat command stands for Network statistics . It displays information about different interface statistics, including open sockets, routing tables, and connection information.	The ss command is a replacement for netstat command. This command gives more information in comparison to the netstat. It is also faster than netstat as it gets all info from kernel userspace.	Dig stands for domain internet groper is a simple DNS lookup utility, that is used to query DNS related info such as A Record, CNAME, MX Record etc. It mainly deals with debug DNS related problems.	Used to displays and manipulate IP routing table for your system.
nslookup	host	arp	iwconfig
This is also another command-line utility to query DNS servers both interactively and non-interactively. It is used to query DNS resource records (RR).	The host command displays domain name for given IP address or vice-versa. It also performs DNS lookups related to the DNS query.	The command arp stands for Address Resolution Protocol. It allows us to view or add content into kernel's ARP table.	Similar to ifconfig, but is dedicated to the wireless interfaces. The command iwconfig configures a wireless network interface. You can view and set basic wi-fi details like SSID and encryption.
hostname	whois	tracepath	curl
The hostname command allows us to set and view /show system's hostname. A hostname is the name of any computer that is connected to a network that is uniquely identified over a network.	The whois command displays information about a website's record. You may get all the information about a website regarding its registration and owner's information.	It is similar to traceroute command, but it doesn't require root privileges. By default, it is installed in Ubuntu. If it's not found in your system you have to install it using your system package manager.	The curl (Client URL) command is mostly used to transfer data over the network and supports various protocols including HTTP, FTP, IMAP, and many others.
wget	mtr	iftop	tcpdump
It is used to download files using HTTP, HTTPS, FTP Protocols. It provides the ability to download multiple files, resume downloads, download in the background, etc.	It is a combination of ping and traceroute utilities and is mainly used for network diagnostics and gives live look at network response and connectivity.	The iftop (Interface TOP) is often used by system admins to monitor stats related to bandwidth and can also be used as a diagnostic tool when you're having issues with the network.	The tcpdump is a packet sniffing and analyzing utility used to capture, analyze and filter network traffic.
iperf	ethtool	scp and sftp	rsync
The iperf is an open-source utility written in C allowing users to perform network performance measurement and tuning.	ethtool is a command-line utility for querying and modifying network interface controller parameters and device drivers.	SCP and SFTP are both file transfer protocols, but they have different functionalities. SCP only allows file transfer, while SFTP allows file access, transfer, and management.	rsync is a fast and versatile command-line utility for synchronizing files and directories between two host over an ssh tunnel.
ifplugstatus	nload	nmcli	bmon
Ifplugstatus command is used to check if the network cable is connected to the network interface. To use the command, you first need to install it.	nload command is used to monitor your network bandwidth. It can show the total amount of data usage and min/max bandwidth usage. To also use it you need to install it first	An easy-to-use, scriptable command-line tool to report network status, manage network connections, and control the NetworkManager.	The bmon is an open-source utility to monitor real-time bandwidth and debug issues by presenting stats in a more human-friendly way.
nc (netcat)	nmap	tshark	vnstat
Referred to as the "Network Swiss Army knife", is a powerful utility used for almost any task related to TCP, UDP, or UNIX-domain sockets. It is used to open TCP connections, listen on arbitrary TCP and UDP ports, perform port scanning	The nmap is a tool to explore and audit network security. It is often used by hackers and security enthusiasts as it allows you to get real-time info on the network, IPs connected to your network in a detailed manner, port scanning, and	TShark is a network protocol analyzer. It lets you capture packet data from a live network, or read packets from a previously saved capture file, either printing a decoded form of those packets to the standard output or writing the packets to	The vnstat utility is mostly used by sysadmins to monitor network traffic and bandwidth consumption (for the most part) as this tool monitors traffic on network interfaces of your system.

Learn 2025 DevOps/SRE tech stack www.heydevops.in

Our Placements <https://heydevops.in/success-stories>

More Project Videos <https://www.youtube.com/praveensingampalli>

Connect with me <https://www.instagram.com/singam4devops/>



Common Network Ports Cheat Sheet

Port	Protocol	Name	Port	Protocol	Name	Port	Protocol	Name
7	TCP/UDP	echo	520	UDP	rip	2103	TCP/UDP	zephyr-clt
9	TCP/UDP	discard	521	UDP	ripng (ipv6)	2104	TCP/UDP	zephyr-hm
19	TCP/UDP	chargen	540	TCP	uucp	2222	TCP	directadmin
20	TCP/SCTP	ftp-data	546	TCP/UDP	dhcpv6	2401	TCP	cvspserver
21	TCP/UDP/SCTP	ftp	547	TCP/UDP	dhcpv6	2483	TCP/UDP	oracle
22	TCP/UDP/SCTP	ssh/scp/sftp	548	TCP	afp	2484	TCP/UDP	oracle
23	TCP	telnet	554	TCP/UDP	rtsp	2809	TCP/UDP	corbaloc
25	TCP	smtp	560	UDP	rmonitor	2967	TCP/UDP	symantec av
42	TCP/UDP	wins replication	563	TCP/UDP	nntp over tls/ssl	3128	TCP/UDP	http proxy
43	TCP/UDP	whois	587	TCP	smtp/submission	3222	TCP/UDP	glbp
49	UDP	tacacs	591	TCP	filemaker	3260	TCP/UDP	iscsi target
53	TCP/UDP	dns	593	TCP/UDP	microsoft dcom	3306	TCP/UDP	mysql
67	UDP	dhcp/bootp	596	TCP/UDP	smsd	3389	TCP	rdp
68	UDP	dhcp/bootp	631	TCP	ipp	3689	TCP	daap
69	UDP	tftp	636	TCP/UDP	ldap over tls/ssl	3690	TCP/UDP	svn
70	TCP	gopher	639	TCP	msdp (pim)	4321	TCP	rwhois
79	TCP	finger	646	TCP/UDP	ldp (mpls)	4333	TCP	msql
80	TCP/UDP/SCTP	http	691	TCP	microsoft exchange	4500	UDP	ipsec nat traversal
88	TCP/UDP	kerberos	860	TCP	iscsi	4899	TCP	radmin
101	TCP	hostname	873	TCP	rsync	5000	TCP	upnp
102	TCP	microsoft exchange iso-tsap	902	TCP/UDP	vmware server	5001	TCP	iperf
110	TCP	pop3	989	TCP	ftps	5004-5005	UDP	rtp/rtsp



Cheatsheet for Docker CLI

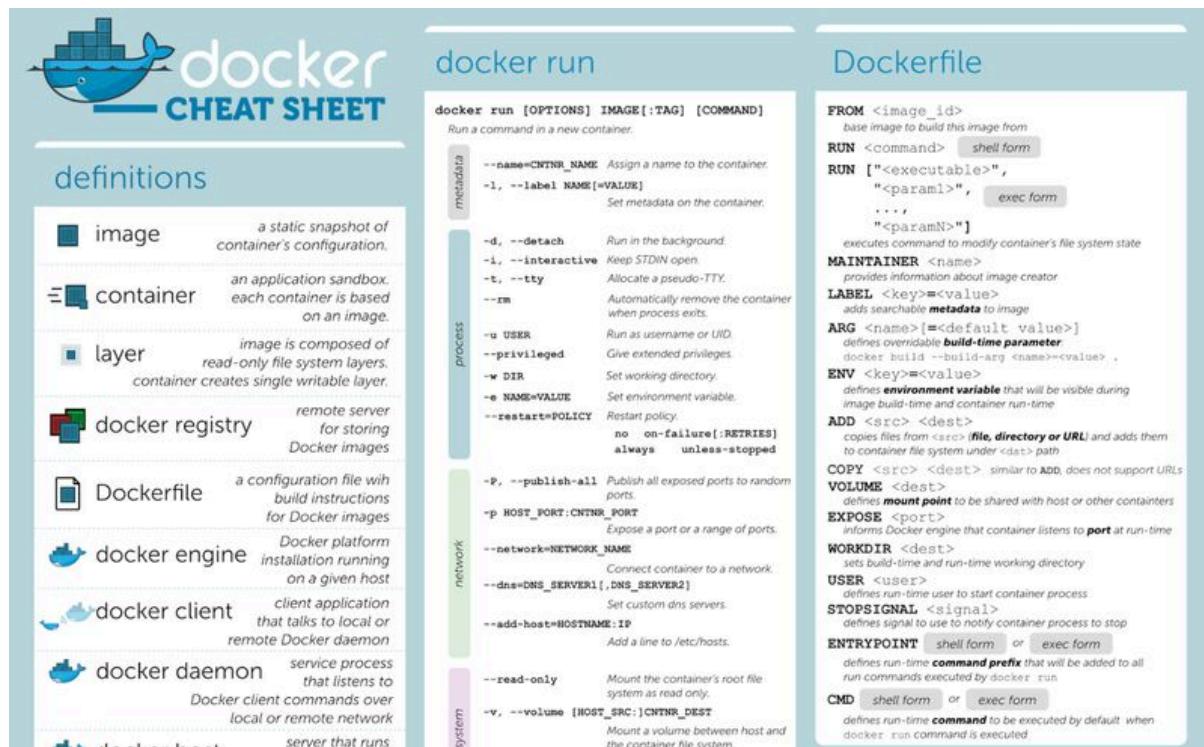
Run a new Container	Manage Containers	Manage Images	Info & Stats
<p>Start a new Container from an Image <code>docker run IMAGE</code> <code>docker run nginx</code></p> <p>...and assign it a name <code>docker run --name CONTAINER IMAGE</code> <code>docker run --name web nginx</code></p> <p>...and map a port <code>docker run -p HOSTPORT:CONTAINERPORT IMAGE</code> <code>docker run -p 8080:80 nginx</code></p> <p>...and map all ports <code>docker run -P IMAGE</code> <code>docker run -P nginx</code></p> <p>...and start container in background <code>docker run -d IMAGE</code> <code>docker run -d nginx</code></p> <p>...and assign it a hostname <code>docker run --hostname HOSTNAME IMAGE</code> <code>docker run --hostname srv nginx</code></p> <p>...and add a dns entry <code>docker run --add-host HOSTNAME:IP IMAGE</code></p> <p>...and map a local directory into the container <code>docker run -v HOSTDIR:TARGETDIR IMAGE</code> <code>docker run -v ~/:/usr/share/nginx/html nginx</code></p> <p>...but change the endpoint <code>docker run -it --entrypoint EXECUTABLE IMAGE</code> <code>docker run -it --entrypoint bash nginx</code></p>	<p>Show a list of running containers <code>docker ps</code></p> <p>Show a list of all containers <code>docker ps -a</code></p> <p>Delete a container <code>docker rm CONTAINER</code> <code>docker rm web</code></p> <p>Delete a running container <code>docker rm -f CONTAINER</code> <code>docker rm -f web</code></p> <p>Delete stopped containers <code>docker container prune</code></p> <p>Stop a running container <code>docker stop CONTAINER</code> <code>docker stop web</code></p> <p>Start a stopped container <code>docker start CONTAINER</code> <code>docker start web</code></p> <p>Copy a file from a container to the host <code>docker cp CONTAINER:SOURCE TARGET</code> <code>docker cp web:/index.html index.html</code></p> <p>Copy a file from the host to a container <code>docker cp TARGET CONTAINER:SOURCE</code> <code>docker cp index.html web:/index.html</code></p> <p>Start a shell inside a running container <code>docker exec -it CONTAINER EXECUTABLE</code> <code>docker exec -it web bash</code></p> <p>Rename a container <code>docker rename OLD_NAME NEW_NAME</code> <code>docker rename 096 web</code></p> <p>Create an image out of container <code>docker commit CONTAINER</code> <code>docker commit web</code></p>	<p>Download an image <code>docker pull IMAGE[:TAG]</code> <code>docker pull nginx</code></p> <p>Upload an image to a repository <code>docker push IMAGE</code> <code>docker push myimage:1.0</code></p> <p>Delete an image <code>docker rmi IMAGE</code></p> <p>Show a list of all Images <code>docker images</code></p> <p>Delete dangling images <code>docker image prune</code></p> <p>Delete all unused images <code>docker image prune -a</code></p> <p>Build an image from a Dockerfile <code>docker build DIRECTORY</code> <code>docker build .</code></p> <p>Tag an image <code>docker tag IMAGE NEWIMAGE</code> <code>docker tag ubuntu ubuntu:18.04</code></p> <p>Build and tag an image from a Dockerfile <code>docker build -t IMAGE DIRECTORY</code> <code>docker build -t myimage .</code></p> <p>Save an image to tar file <code>docker save IMAGE > FILE</code> <code>docker save nginx > nginx.tar</code></p> <p>Load an image from a tar file <code>docker load -i TARFILE</code> <code>docker load -i nginx.tar</code></p>	<p>Show the logs of a container <code>docker logs CONTAINER</code> <code>docker logs web</code></p> <p>Show stats of running containers <code>docker stats</code></p> <p>Show processes of container <code>docker top CONTAINER</code> <code>docker top web</code></p> <p>Show installed docker version <code>docker version</code></p> <p>Get detailed info about an object <code>docker inspect NAME</code> <code>docker inspect nginx</code></p> <p>Show all modified files in container <code>docker diff CONTAINER</code> <code>docker diff web</code></p> <p>Show mapped ports of a container <code>docker port CONTAINER</code> <code>docker port web</code></p>

Learn 2025 DevOps/SRE tech stack www.heydevops.in

Our Placements <https://heydevops.in/success-stories>

More Project Videos <https://www.youtube.com/praveensingampalli>

Connect with me <https://www.instagram.com/singam4devops/>



HEY DEVOPS

Learn 2025 DevOps/SRE tech stack www.heydevops.in

Our Placements <https://heydevops.in/success-stories>

More Project Videos <https://www.youtube.com/praveensingampalli>

Connect with me <https://www.instagram.com/singam4devops/>

Kubernetes cheat sheet _2_.png

Role Based Access Control - Plugins

rakness	Show an access matrix for k8s server resources.
kubectl-who-can	Display who has RBAC permissions to perform actions on different resources in Kubernetes.

Storage - Plugins

kubectl-plugin-pvc	For PVC operations, a simple kubectl binary plugin.
kubectl-dfi	Images on Kubernetes nodes can be listed and shown by their disk resources.
kubectl-df-pv	Allowing administrators to manage persistent volumes with a DF (disk free)-like utility.

Exec - Plugins

kubectl-enter	Exec into node via kubectl.
kubectl-ssh	A kubectl plugin to ssh into Kubernetes nodes within separate tmux panes.
kubectl-lexec	Kubectl plugin to interactively exec into a pod.
kubectl-ssh	A kubectl plugin to ssh into a pod's node.
kubectl-kcn	A script and kubectl plugin which makes it easier to ssh into nodes within your cluster.
kubectl-exec-all	kubectl plugin to execute a command in all running pods of a resource.
kubectl-nsenter	Simple kubectl plugin to take pod name, SSH onto node and spawn an nsenter shell.
kvaps kubectl-ssh	Tiny plugin for connecting to node in the cluster over SSH.
kubectl-warp	Kubernetes CLI plugin for syncing and executing local files in Pod on Kubernetes.
kubectl-plugin-ssh-jump	A kubectl plugin to SSH into Kubernetes nodes using a SSH jump host Pod.
kubectl-texec	Attach to the pod matching the label with Tmux.

Linting - Plugins

kubectl-lint	Performs linting.
kubectl-neat	Improves the readability of Kubernetes' yaml and json output.
ksort	Sort manifest files according to kind.

Debugging - Plugins

kubectl-debug	Debug your pod by a new container with troubleshooting tools already installed.
kube-profe	Makes continuous profiling easier to perform in Kubernetes.

debugging - plugins

kubectl-sudo	Run Kubernetes commands with the security privileges of another user.
outdated	Find and report outdated images running in a Kubernetes cluster.
kubectl-kubesc	Security risk analysis for Kubernetes resources.

Networking - Plugins

ksniff	An easy way to sniff on Kubernetes pods with tcpdump and Wireshark.
kubectl-swiftnp	A kubectl plugin that renders details about Network Policies.
kubectl-trace	Using the kubectl networking command, you can schedule bpftrace programs on your Kubernetes cluster.
kubectl-capture	A plugin for kubectl that triggers a Sysdig capture.

Pods

Get commands with basic output

kubectl get services	# List all services in the namespace
kubectl get pods -all-namespaces	# List all pods in all namespaces
kubectl get pods -0 wide	# List all pods in the current namespace, with more details
kubectl get deployment my-dep	# List a particular deployment
kubectl get pods	# List all pods in the namespace
kubectl get pod my-pod -o yaml	# Get a pod's YAML

Deleting Resources

kubectl delete -f ./pod.json	# Delete a pod using the type and name specified in pod.json
kubectl delete pod unwanted --now	# Delete a pod with no grace period
kubectl delete pod,service baz foo	# Delete pods and services with same names "baz" and "foo"
kubectl delete pods,services -l name=myLabel	# Delete pods and services with label name=myLabel
kubectl -n my-ns delete pod,svc --all	# Delete all pods and services in namespace my-ns,
kubectl get pods -n mynamespace --no-headers=true awk '/pattern1 pattern2/{print \$1}' xargs kubectl delete -n mynamespace pod	# Delete all pods matching the awk pattern1 or pattern2

Namespace Switching

kubectx	Switch faster between clusters and namespaces in Kubectl	Context/Namespace Switching
kubectl-ns	Simple kubectl plugin to display/switch namespaces	Context/Namespace Switching
kubectl-use	Plugin for simple switch kubernetes contexts and namespaces	Context/Namespace Switching
kubectl-switch	Kubernetes multi-cluster command-line management tool	Context/Namespace Switching
caas-one kubectl-switch	Kubernetes multi-cluster command-line management tool	Context/Namespace Switching
juanvallejo kubectl-ns	Kubernetes multi-cluster command-line management tool	Context/Namespace Switching
km	EKS MFA kubeconfig management tool. Thin wrapper for kubectl	Context/Namespace Switching

Resource CRUDs

ketall	Like kubectl get all, but get really all resources	Resource CRUDs
kubectl-grep	Filter kubernetes resources by matching their names	Resource CRUDs
kubectl-all	Kubectl plugin to list all resources in given namespaces	Resource CRUDs
kubectl-watch	Watches kubernetes resources	Resource CRUDs
kubectl-custom-cols	A kubectl plugin that helps you customise output columns	Resource CRUDs
kubepod	Search pods faster in kubectl	Resource CRUDs
kubectl-resources	Plugin to access kubernetes resources requests, limits and usage	Resource CRUDs
kubectl-free	Show various requested resources on kubernetes nodes	Resource CRUDs
kubectl-match-name	regex matching for resource names	Resource CRUDs
kubectl-eksporter	A simple Ruby-script to export k8s resources	Resource CRUDs

Authentication

kubelogin	Kubectl plugin for kubernetes OpenID Connect authentication (kubectl idc-login)	Authn/Authz
k8s-pixy-auth	K8s plugin to authentication against an OIDC compatible issuer using PKCE (pixy) flow	Authn/Authz
kubectl-login	Kubectl plugin for re-authenticate into OpenID Connect Provider via CLI	Authn/Authz
zj2wry kubectl-logiw	The kubectl plugin is used to login to the kubernetes cluster by OIDC authentication	Authn/Authz
kubectl-pass	Kubectl plugin for integration with pass (the standard unix password manager)	Authn/Authz

Other Commands

kubectl-open-svc-plugin	Kubectl open-sec plugin makes services accessible via their ClusterIP from outside your cluster	Uncategorised
kubectl-doctor	K8s cluster triage plugin - scan your cluster for anomalies (brew doctor equivalent)	Uncategorised
kubectl-view-serviceaccount-kubeconfig-plugin	A kubectl plugin that shows a kubeconfig to access the apiserver with a specified service account	Uncategorised
unfork	kubectl plugin to forked Helm Charts and other k8s resources and unfork them https://www.unfork.io	Uncategorised
kubectl-config-merge	A kubectl plugin for merging multiple kubeconfig files. A cli-runtime example	Uncategorised
konfig	konfig helps to merge, split or import kubeconfig files	Uncategorised
kubectlsafe	Safe operations in kubectl with plugin kubectlsafe	Uncategorised
kubectl-rainbow	Simple binary to colourise output	Uncategorised
kubectl-ansible	An sensible dynamic inventory plugin for kubernetes cluster nodes	Uncategorised
kubectl-config-cleanups	A kubectl plugin for automatically cleaning up your kubeconfig	Uncategorised
kubectl-gitlab-bootstrap	Quickly add a kubernetes cluster to a GitLab project	Uncategorised
kubectl-captain	Kubectl plugin for captain	Uncategorised
kubectl-preflight	Preflight Checks and Support Bundles Framework for Kubernetes Applications	Uncategorised
kubectl-support-bundle	Preflight Checks and Support Bundles Framework for Kubernetes Applications	Uncategorised
kubectl-virt-plugin	Holds all scripts to create packages and manifest files required for publishing with virtual binary as a krew package for kubectl	Uncategorised
kcf	A CLI tool providing you with status and configuration of kubernetes cluster fleet	Uncategorised
kubepug	Kubernetes PreUpGrade (Checker)	Uncategorised

Learn 2025 DevOps/SRE tech stack www.heydevops.in

Our Placements <https://heydevops.in/success-stories>

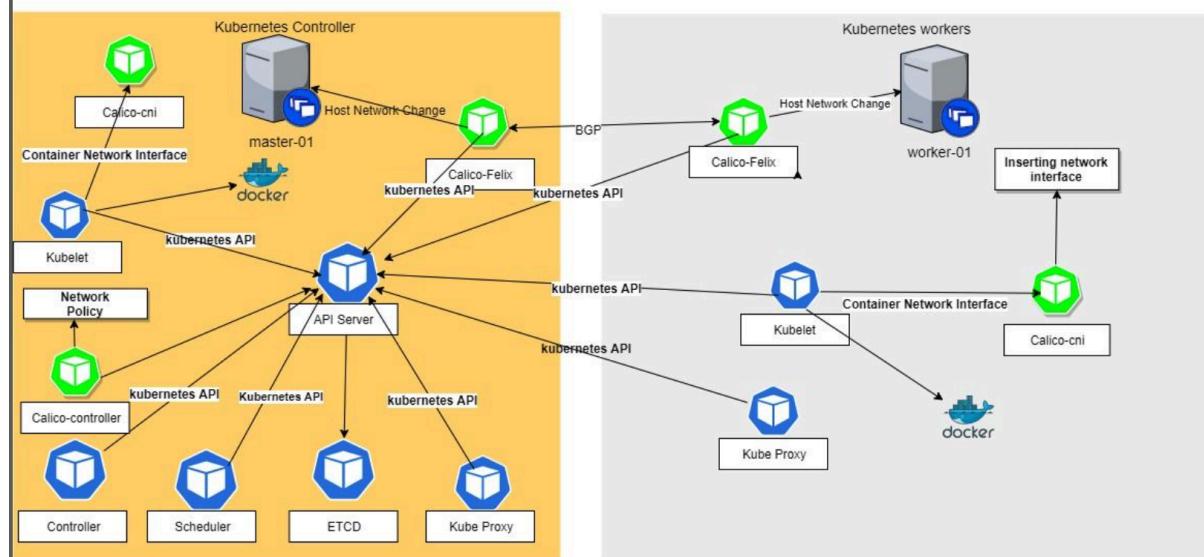
More Project Videos <https://www.youtube.com/praveensingampalli>

Connect with me <https://www.instagram.com/singam4devops/>

Kubernetes Cheatsheet

What is Kubernetes Kapsule and Kubernetes Kosmos?	Viewing and finding resources	DEPLOYMENTS	EVENTS	ROLES	PERSISTENT VOLUME CLAIM
Kubernetes is an open-source platform that enables developers to manage their containerized applications. Kapsule and Kosmo both provide a managed environment for creating, configuring, and running clusters of pre-configured machines.	NODES # Display all node information <code>kubectl get no</code> # Show more information about all nodes <code>kubectl get no -o wide</code> # Display node details with verbose output <code>kubectl describe no</code> # Filter the node with the specified label <code>kubectl get node --selector=[label_name]</code> # Display node (CPU/memory) usage <code>kubectl top node [node_name]</code>	# Display all deployments information <code>kubectl get deploy</code> # Display deployments details <code>kubectl describe deploy</code> # Show more information about all deployments <code>kubectl get deploy -o wide</code>	# Display all events information <code>kubectl get events</code> # Display events information within the namespace kube-system <code>kubectl get events -n kube-system</code> # Lists the specific resources events or the entire cluster <code>kubectl get events -v</code>	# Display all roles information within all namespaces <code>kubectl get roles --all-namespaces</code>	# Display all persistent volume claim information <code>kubectl get pvc</code> # Display all persistent volume claim details <code>kubectl describe pvc</code>
Kapsule clusters are composed solely of Scaleway Instances whereas Kosmos is a managed Multi-Cloud Kubernetes Engine that allows to connect Instances and virtual or dedicated servers from any cloud provider to a single managed Control-Plane.	SERVICES # Display all services information <code>kubectl get svc</code> # Display services details <code>kubectl describe svc</code> # Show more information about all services <code>kubectl get svc -o wide</code> # Display a pod's label <code>kubectl get svc --show-labels</code>	# Display all logs information of a specific pod <code>kubectl logs [pod_name]</code> # Display all logs information of a specific pod for the last 10 lines <code>kubectl logs --since=1h [pod_name]</code> # Display all logs information of a specific pod in a specific container <code>kubectl logs -c <c> [pod_name]</code> # Transfer all logs information of a specific pod in the pod.log file <code>kubectl logs [pod_name] > pod.log</code>	LOGS # Display all secrets information <code>kubectl get secrets</code> # Display all secrets information within all namespaces <code>kubectl get secrets --all-namespaces</code>	SECRETS # Display all secrets information <code>kubectl get secrets</code> # Display all secrets information within all namespaces <code>kubectl get secrets --all-namespaces</code>	STORAGE CLASS # Display all storage class information <code>kubectl get sc</code>
Creating resources # Create resource(s) from file <code>kubectl apply -f [manifest].yaml</code>	PODS # Display all container group information <code>kubectl get po</code> # Show more information about all pods <code>kubectl get po -o wide</code> # Display pod details with verbose output <code>kubectl describe po</code> # View the labels of the container group <code>kubectl get po --show-labels</code> # Display pod usage (CPU/memory) <code>kubectl top pod [pod_name]</code>	# Display all daemon sets information <code>kubectl get ds</code> # Display the detailed state of daemonsets within all namespaces <code>kubectl describe ds --all-namespaces</code> # Transfer all logs information of a specific daemonset in the daemonset.log file <code>kubectl logs [daemonset_name] -n [namespace_name]</code>	DAEMON SETS # Display all daemon sets information <code>kubectl get ds</code> # Display the detailed state of daemonsets within all namespaces <code>kubectl describe ds --all-namespaces</code> # Transfer all logs information of a specific daemonset in the daemonset.log file <code>kubectl logs [daemonset_name] -n [namespace_name]</code>	CONFIG MAPS # Display all config maps information <code>kubectl get cm</code> # Display all config maps within all namespaces <code>kubectl get cm --all-namespaces</code>	MULTIPLE RESOURCES # Display all services and pods information <code>kubectl get svc,po</code> # Display all deploys and nodes information <code>kubectl get deploy,no</code> # Display all the pods, services, statefulsets, etc. in the cluster. Not all the resources are listed using this command. <code>kubectl get all --all-namespaces</code>
Updating resources # Apply a taint that has a key-value of taint-test with a NoSchedule effect <code>kubectl taint nodes [node-name] taint-test:NoSchedule</code> # Mark node as unschedulable <code>kubectl cordon [node-name]</code> # Mark node as scheduleable <code>kubectl uncordon [node-name]</code> # Drain node in preparation for maintenance <code>kubectl drain [node-name]</code>	NAMESPACES # Display all namespace information <code>kubectl get ns</code> # Display namespace details <code>kubectl get ns --export</code>	INGRESS # Display all ingress information <code>kubectl get ing</code> # Display all ingress information within all namespaces <code>kubectl get ing --all-namespaces</code>	SERVICE ACCOUNT # Display all service account information <code>kubectl get sa</code>	PERSISTENT VOLUMES # Display all persistent volumes information <code>kubectl get pv</code> # Display persistent volumes details <code>kubectl describe pv</code>	

Kubernetes Network Architecture with Calico



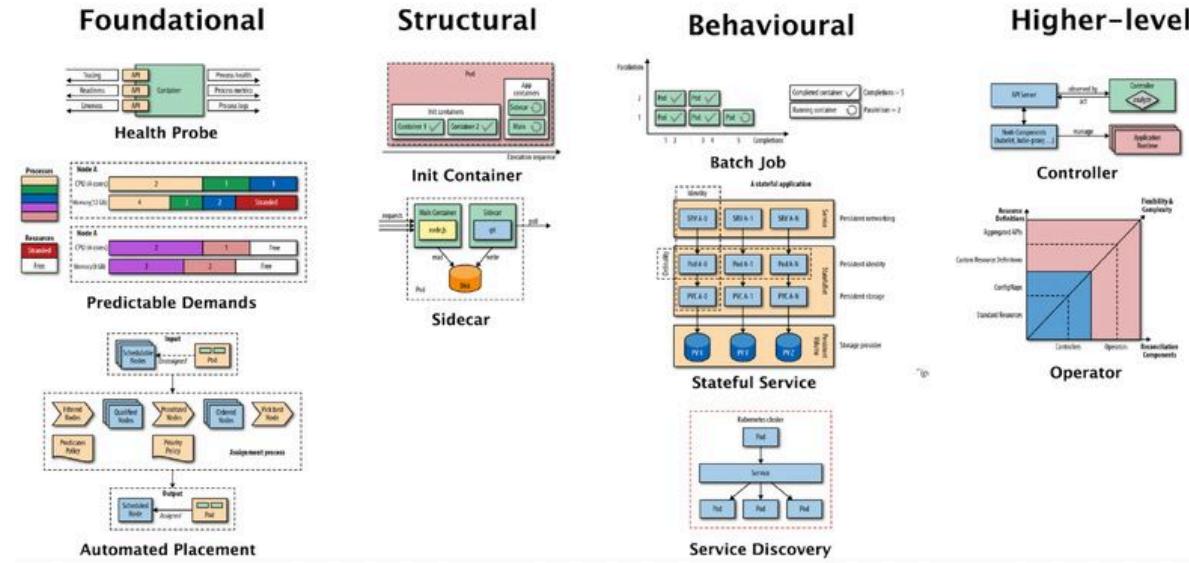
Learn 2025 DevOps/SRE tech stack www.heydevops.in

Our Placements <https://heydevops.in/success-stories>

More Project Videos <https://www.youtube.com/praveensingampalli>

Connect with me <https://www.instagram.com/singam4devops/>

Top 10 Must-Know Design Patterns for Kubernetes Beginners



Python Cheat Sheet

For Beginners & Lazy Experts (1/2)

Data Types

Text Type	<code>str</code>	'I'm a string'
Numeric Types	<code>int</code>	10
	<code>float</code>	10.3
	<code>complex</code>	2 - 3j
Boolean Type	<code>bool</code>	True, False
Sequence Types	<code>list</code>	[1, 2, 'a', 'b']
	<code>tuple</code>	(1, 2, 3)
	<code>range</code>	range(4)
Set type	<code>set</code>	{1, 2, 3}
Mapping type	<code>dict</code>	{'1': 'a', '2': 'b', '3': 'c'}

Data Type Conversions

Integer and Float Conversions	<code>float(2 + 3)</code>	5.0	>>> int(2.0 + 3.0)	5
Real to Complex Data Type Conversion	<code>complex(2, 3.4)</code>	(2+3.4j)	>>> complex(5)	(5j+0)

Data Type Conversion with Strings

<code>int('123')</code>	123	>>> int('456.78')	456
<code>str(15)</code>	'5'	>>> str(123.45)	'123.45'
<code>str(1+2j)</code>	'(1+2j)'		

Type Conversion to Tuples and Lists

<code>tuple([1, 2, 3])</code>	(1, 2, 3)	>>> list((1, 2, 3))	[1, 2, 3]
<code>tuple('AB')</code>	('A', 'B')	>>> list('AB')	['A', 'B']

Type Conversion to Dictionaries and Sets

<code>set([('a', 1, 2, 'b', 2)])</code>	{'a', 1, 2, 'b'}	>>> dict([('a', 1), ('b', 2)])	{'a': 1, 'b': 2}
---	------------------	--------------------------------	------------------

Convert Binary to Decimal

<code>bin(12)</code>	'0b1100'	>>> int('0b11')	3
----------------------	----------	-----------------	---

Convert Hexadecimal to Decimal

<code>hex(12)</code>	'0x8c'	>>> int(0x4f)	79
----------------------	--------	---------------	----

Convert Text to Decimal

<code>ord('a')</code>	97	>>> chr(65)	'A'
-----------------------	----	-------------	-----

Booleans

Booleans as Numbers

<code>>>> True == 1</code>	True	>>> False == 0	True
-------------------------------------	------	----------------	------

Comparison Operators

<code>a == b</code>	is a equal to b?	<code>a != b</code>	is a different to b?
<code>a < b</code>	is a less than b?	<code>a <= b</code>	is a less than or equal to b?
<code>a > b</code>	is a greater than b?	<code>a >= b</code>	is a greater or equal to b?

Membership and Identity Operators

<code>a in b</code>	is a in b?	<code>a is b</code>	are a and b the same object?
<code>a not in b</code>	is a not in b?	<code>a is not b</code>	are a and b different objects?

Boolean Operators

<code>not</code>	returns False if operand is True, True otherwise		
<code>and</code>	returns True if both operands are True, False otherwise		
<code>or</code>	returns False if both operands are False, True otherwise		

Operator Precedence

<code>()</code>	parentheses are evaluated first
<code>**</code>	exponent
<code>*, /, %</code>	multiplication, divisions, and modulo
<code>+, -, <, <, >, ></code>	addition and subtraction
<code>is, is not, in, not in</code>	comparison, identity, and membership operators
<code>not</code>	logical NOT
<code>and</code>	logical AND
<code>or</code>	logical OR

Print Function

<code>print(*a, **b)</code>	a+b
<code>print(*c, *d, sep='?', end='\n')</code>	c+d\n
<code>print(*e, *f)</code>	e+f

User Input

<code>name = input("Enter your name: ")</code>	>>> Enter your name: Promethee
<code>print("Your name is: " + name)</code>	Your name is: Promethee

Decision Structure

<code>if n == 0:</code>	
<code> print("n is zero")</code>	
<code>elif n > 0:</code>	
<code> print("n is strictly positive")</code>	
<code>else:</code>	
<code> print("n is strictly negative")</code>	

Repetition Structures

<code>n = 0</code>	0	<code>for i in range(4):</code>	0
<code>while n < 4:</code>	1	<code> print(i)</code>	1
<code> n += 1</code>	2	<code>print("i = ", i)</code>	2
<code>print("n = ", n)</code>	3		3
<code>n = 4</code>	4		1 = 3

Exceptions

<code>try:</code>		Built-in Exceptions	
<code> # run this code</code>		<code>FileNotFoundError</code>	
<code>except NameError:</code>		<code>IndexError</code>	
<code> # handle error type 1</code>		<code>KeyError</code>	
<code>except NameError:</code>		<code>ModuleNotFoundError</code>	
<code> # handle error type 2</code>		<code>NameError</code>	
<code>except:</code>		<code>SyntaxError</code>	
<code> # handle any other error</code>		<code>TypeError</code>	
<code>else:</code>		<code>ValueError</code>	
<code> # run this code if no error</code>		<code>ZeroDivisionError</code>	
<code>Finally:</code>			
<code> # always run this code</code>			

Modules

<code>>>> import random</code>	
<code>>>> import pi</code>	
<code>>>> print(random.randint(0,9))</code>	2
<code>>>> print(pi)</code>	3.141592653589793

Files

<code>open()</code>	returns a file object	Access modes	
<code>close()</code>	close the file	<code>r</code>	read
<code>read()</code>	returns the file content	<code>w</code>	write
<code>readline()</code>	returns one line from the file	<code>a</code>	append
<code>readlines()</code>	returns a list of lines from the file	<code>x</code>	create

Iterating over Strings

<code>for e in 'ABC123':</code>		<code>print(e, end='')</code>	A+B+C=1+2+3>
<code>for i in range(len('ABC123')):</code>		<code>print('ABC123'[i], end='')</code>	A+B+C=1+2+3=

Learn 2025 DevOps/SRE tech stack www.heydevops.in

Our Placements <https://heydevops.in/success-stories>

More Project Videos <https://www.youtube.com/praveensingampalli>

Connect with me <https://www.instagram.com/singam4devops/>



Terraform Cheat Sheet

Installation		Managing state	
See https://developer.hashicorp.com/terraform/downloads for installation instructions for your platform. Then run <code>terraform -install-autocomplete</code> to enable shell autocompletion.			
Initializing Terraform			
Command	Description	Command	Description
<code>terraform init</code>	Prepare your working directory for other commands	<code>terraform refresh</code>	Update the state to match remote systems
<code>terraform init -upgrade</code>	Upgrade all modules to their latest versions	<code>terraform show</code>	Show the current state
<code>terraform get</code>	Only download and install modules	<code>terraform show <file></code>	Show a saved plan
Making infrastructure changes			
Command	Description	Command	Description
<code>terraform plan</code>	Show changes required by the current configuration	<code>terraform show -json <file></code>	Show the plan in JSON format
<code>terraform plan -out=<file></code>	Write the plan to a file to apply it later	<code>terraform state list</code>	List all resources in the state file
<code>terraform plan -target <resource></code>	Create a plan for a specific module or resource	<code>terraform state show <resource></code>	Show details about a resource
<code>terraform plan -replace <resource></code>	Force the plan to replace a specific resource	<code>terraform state mv <source> <dest></code>	Rename a resource in the state file
<code>terraform plan -var '<key>=<value>'</code>	Set a value for one of the input variables	<code>terraform state rm <resource></code>	Remove a resource from the state file
<code>terraform plan -refresh-only</code>	Inspect resource drift without updating the state file	<code>terraform state replace-provider <from> <to></code>	Replace provider for resources in the state
<code>terraform apply</code>	Create or update infrastructure	<code>terraform import <resource> <remote_id></code>	Import existing infrastructure into Terraform
<code>terraform apply <file></code>	Create or update infrastructure using a plan file	<code>terraform state pull</code>	Pull current state and output to stdout
<code>terraform apply -target <resource></code>	Create or update a specific resource		
<code>terraform apply -replace <resource></code>	Force the replacement of a specific resource		
<code>terraform apply -auto-approve</code>	Skip interactive approval of plan before applying		
Inspecting output values		Tainting resources	
Command	Description	Command	Description
<code>terraform output</code>	Show all output values	<code>terraform taint <resource></code>	(Deprecated) Mark a resource to be replaced
<code>terraform output -json</code>	Show all output values in JSON format	<code>terraform untaint <resource></code>	Mark a resource as no longer tainted
<code>terraform output <name></code>	Show a specific output value		
<code>terraform output -raw <name></code>	Show a specific output value without quotes		
Formatting and validation		Destroying infrastructure	
Command	Description	Command	Description
		<code>terraform destroy</code>	Destroy infrastructure managed by Terraform
		<code>terraform destroy -target <resource></code>	Destroy a specific resource



Helm Cheat Sheet



Helm		Listing releases	
Helm is the package manager for Kubernetes. See https://helm.sh/docs/intro/install/ for installation instructions.		<code>helm list</code>	List all releases in the current namespace
Global flags		<code>helm list --all-namespaces</code>	List all releases in all namespaces
Flag	Description	<code>helm list -l <label>=<value></code>	List releases with a specific label
<code>--kube-context <name></code>	Name of the Kubernetes context to use	<code>helm list --date</code>	List releases sorted by date
<code>--namespace <name></code>	Namespace to use for this operation	<code>helm list --(pending failed uninstalled)</code>	List releases that are in a pending/failed/uninstalled state
Repository management		<code>helm status <name></code>	Show the status of a release
Command	Description		
<code>helm repo add <name> <url></code>	Add a repository		
<code>helm repo list</code>	List all added repositories		
<code>helm repo update</code>	Update the local cache of available charts		
<code>helm repo remove <name></code>	Remove a repository		
<code>helm search repo</code>	List all charts in the repositories		
<code>helm search repo <keyword></code>	Search for a chart in the repositories		
Installing Helm charts		Managing releases	
Command	Description	Command	Description
<code>helm install <name> <chart></code>	Install a chart with a name	<code>helm upgrade <name> <chart></code>	Upgrade a release
<code>helm install <chart> --generate-name</code>	Install a chart, auto-generating a name	<code>helm upgrade <name> <chart> --atomic</code>	Upgrade a release atomically
<code>helm install <name> <chart> --namespace <namespace></code>	Install a chart in a specific namespace	<code>helm upgrade <name> <chart> --dependency-update</code>	Upgrade a release and update dependencies
<code>helm install <name> <chart> --set <key>=<value></code>	Install a chart with specific values	<code>helm upgrade <name> <chart> --version <version></code>	Upgrade a release to a specific version
<code>helm install <name> <chart> --values <file></code>	Install a chart using a values file	<code>helm upgrade <name> <chart> --set <key>=<value></code>	Upgrade a release with specific values
<code>helm install <name> <chart> --dry-run --debug</code>	Run a test installation to validate the chart	<code>helm rollback <release> <revision></code>	Rollback a release to a previous revision
<code>helm install <name> <chart> --verify</code>	Verify the package before installing		
<code>helm install <name> <chart> --dependency-update</code>	Update dependencies before installing		
<code>helm uninstall <name></code>	Uninstall a release		
<code>helm uninstall <name> --keep-history</code>	Uninstall a release, keeping the history		
Developing charts			
Command	Description	Command	Description
<code>helm create <name></code>	Create a new chart	<code>helm package <chart-path></code>	Package a chart directory into a chart file
<code>helm lint <chart></code>	Lint a chart	<code>helm show all <chart></code>	Inspect a chart and list all resources
<code>helm show values <chart></code>	Inspect a chart and show default values	<code>helm template <name> <chart></code>	Render templates locally
<code>helm template <name> <chart> --set <key>=<value></code>	Render templates locally and override values		

Learn 2025 DevOps/SRE tech stack www.heydevops.in

Our Placements <https://heydevops.in/success-stories>

More Project Videos <https://www.youtube.com/praveensingampalli>

Connect with me <https://www.instagram.com/singam4devops/>

KUBERNETES NOTES

```
#####
#####
#####
#####
#####
#
minikube
#####
minikube start
minikube status
minikube stop
# To login to minikube VM from cmd prompt
minikube ssh
# Credentials for minikube VM: docker / tcuser
# Below command will give you kubernetes control plane version (server) and kubectl
version (client)
kubectl version
# To upgrade kubernetes version
minikube start --kubernetes-version=v1.20.0
https://github.com/kubernetes/minikube/releases
https://kubernetes.io/docs/tasks/tools/install-kubectl/#install-kubectl-on-windows
https://www.youtube.com/watch?v=ppgrKs1FNJE

#####
#
kubectl
#####

```

kubectl <command> --help

kubectl rollout --help

kubectl explain object.sub-object

Learn 2025 DevOps/SRE tech stack www.heydevops.in

Our Placements <https://heydevops.in/success-stories>

More Project Videos <https://www.youtube.com/praveensingampalli>

Connect with me <https://www.instagram.com/singam4devops/>

e.g kubectl explain pod.spec.containers

e.g kubectl explain deploy

e.g kubectl explain pod.spec.containers --recursive

alias k=kubectl

--dry-run flag can be appended to run or create commands (imperative commands)

While using kubectl apply can be harder to debug since it's not as explicit, we often use it instead of kubectl create and kubectl replace.

#####

Node

#####

kubectl get no

kubectl get no -o wide

kubectl describe no <node-name>

#####

Pod

#####

kubectl get po

kubectl get po -o wide

kubectl get po <pod name> -o yaml # output in yaml format

kubectl get po <pod name> -o json # output in json format

kubectl describe po <pod name>

kubectl delete po <pod name>

kubectl delete po --all # deletes all pods

kubectl apply -f <pod.yaml> -n <namespace-name>

Learn 2025 DevOps/SRE tech stack www.heydevops.in

Our Placements <https://heydevops.in/success-stories>

More Project Videos <https://www.youtube.com/praveensingampalli>

Connect with me <https://www.instagram.com/singam4devops/>

```
# If you are not given a pod definition file but a pod, you may extract the definition to a file  
using the below command.
```

```
# Then edit the file to make the necessary changes, delete and re-create the pod.
```

```
kubectl get po <pod-name> -o yaml > pod.yaml
```

```
vi pod.yaml
```

```
kubectl delete po <pod-name>
```

```
kubectl apply -f pod.yaml
```

```
# To edit pod properties
```

```
kubectl edit po <pod-name> # opens in vi editor
```

```
kubectl run nginx --image=nginx --dry-run=client -o yaml > pod.yaml
```

```
# To execute a command inside a running container in the pod
```

```
kubectl exec [POD] -- [COMMAND]
```

```
kubectl exec nginx -- ls /
```

```
kubectl exec -it nginx -- /bin/sh # Interactive mode. Get a shell to the container running in  
your Pod
```

```
kubectl exec -it web -c nginx -- /bin/bash
```

```
# In a multicontainer pod, containers are created and destroyed together.
```

```
# Just like containers, initContainers is a property under spec and it has same sub  
properties like name, image etc.
```

```
# Note: kubectl run command creates the pod and not deployment.
```

```
# There is no imperative command like kubectl create po for pod creation.
```

```
# name, labels, annotations and namespace are part of metadata.
```

```
# kubectl get po -A # all pods in all namespaces
```

Learn 2025 DevOps/SRE tech stack www.heydevops.in

Our Placements <https://heydevops.in/success-stories>

More Project Videos <https://www.youtube.com/praveensingampalli>

Connect with me <https://www.instagram.com/singam4devops/>

```
# kubectl get events -A | grep error # all events in all namespaces with errors
```

```
# Pods have a property called restartPolicy whose values can be Always, Never or  
OnFailure.
```

Default value is Always.

```
#####
#
```

commands and arguments

```
#####
#
```

```
# A container only lives as long as the process inside it is alive. If the process inside the  
container is finished or crashes, the container exits.
```

```
# When you specify the command in JSON array format, the first element in the array  
should be executable. e.g command: ["sleep", "4800"]
```

```
# To define a command, include the command field in configuration file.
```

```
# To define arguments for the command, include the args field in configuration file.
```

```
# The command and arguments that you define in the pod configuration file override the  
default command and arguments provided by the container image.
```

```
# If you define args, but do not define a command, the default command is used with your  
new arguments.
```

```
# In a docker file, cmd provides default argument to entrypoint instruction.
```

```
# command replaces the image's ENTRYPOINT instruction, the command that is executed  
to start your container.
```

```
# args replaces the image's CMD instruction. This list of arguments is passed to the  
command specified in the previous field.
```

```
# Example1:
```

```
spec:
```

```
containers:
```

```
- name: command-demo-container
```

```
image: debian
```

```
command: ["printenv"]
```

```
args: ["HOSTNAME", "KUBERNETES_PORT"]
```

```
#     command:
```

Learn 2025 DevOps/SRE tech stack www.heydevops.in

Our Placements <https://heydevops.in/success-stories>

More Project Videos <https://www.youtube.com/praveensingampalli>

Connect with me <https://www.instagram.com/singam4devops/>

- "sleep"

- "5000"

```
# Example3: To run your command in a shell
```

```
command: ["/bin/sh"]
```

```
args: ["-c", "while true; do echo hello; sleep 10;done"]
```

```
# Example4:
```

```
command: ["sleep", "5000"]
```

```
# Example7:
```

```
command: ["ls", "index.html"]
```

Note: Enter the command as is in a shell form. Or in a JSON array format (Preferred). In a JSON array format, the first element should be executable.

```
#
```

```
cat Dockerfile2
```

```
FROM python:3.6-alpine
```

```
RUN pip install flask
```

```
COPY . /opt/
```

```
EXPOSE 8080
```

```
WORKDIR /opt
```

```
ENTRYPOINT ["python", "app.py"]
```

```
CMD ["--color", "red"]
```

On container startup, the following command would run:

```
python app.py --color red
```

Learn 2025 DevOps/SRE tech stack www.heydevops.in

Our Placements <https://heydevops.in/success-stories>

More Project Videos <https://www.youtube.com/praveensingampalli>

Connect with me <https://www.instagram.com/singam4devops/>

```
cat Dockerfile
FROM python:3.6-alpine
RUN pip install flask
COPY . /opt/
EXPOSE 8080
WORKDIR /opt
ENTRYPOINT ["python", "app.py"]
```

On container startup, the following command would run:

```
python app.py
```

```
#
```

If you supply a command but no args for a Container, only the supplied command is used.

The default ENTRYPOINT and the default CMD defined in the Docker image are ignored.

```
#
```

If you supply a command and args, the default ENTRYPOINT and the default CMD defined in the Docker image are ignored.

Your command is run with your args.

```
#
```

If you supply only args for a Container, the default Entrypoint defined in the Docker image is run with the args that you supplied.

For more details, you can refer to this test:

```
https://kodekloud.com/courses/certified-kubernetes-administrator-with-practice-tests-labs/lectures/12038798
```

```
#####
replicaset
#####
kubectl apply -f <replicaset-definition.yaml>
```

Learn 2025 DevOps/SRE tech stack www.heydevops.in

Our Placements <https://heydevops.in/success-stories>

More Project Videos <https://www.youtube.com/praveensingampalli>

Connect with me <https://www.instagram.com/singam4devops/>

```
kubectl get rs
```

```
kubectl get rs -o wide
```

```
kubectl delete rs <replicaset-name>
```

```
kubectl scale rs <> --replicas=6
```

or

```
kubectl edit rs <replicaset-name>
```

```
kubectl describe rs <replicaset-name>
```

There are 2 ways to edit a rs.

Either delete and re-create the ReplicaSet or

Update the existing ReplicaSet and then delete all PODs, so new ones with the correct image will be created.

```
kubectl edit rs <rs-name>
```

```
kubectl delete po <rs-po-name> # Delete all pods under rs.
```

```
kubectl get rs <rs-name> -o yaml > rs.yaml
```

```
vi rs.yaml
```

```
kubectl apply -f rs.yaml
```

```
kubectl delete po <rs-po-name> # Delete all pods under rs.
```

The value for labels in spec.selector clause and spec.template.metadata should match in replicaset.

To see all the objects at once in current namespace

```
kubectl get all
```

Replicaset has a template.

```
#####
#
```

Deployments

Learn 2025 DevOps/SRE tech stack www.heydevops.in

Our Placements <https://heydevops.in/success-stories>

More Project Videos <https://www.youtube.com/praveensingampalli>

Connect with me <https://www.instagram.com/singam4devops/>

#####

```
kubectl create deploy nginx --image=nginx --replicas=2
```

```
kubectl get deploy
```

```
kubectl get deploy -n <namespace-name>
```

```
kubectl get deploy <deployment-name>
```

```
kubectl get deploy <deployment-name> -o yaml | more
```

```
kubectl get deploy -o wide
```

```
kubectl describe deploy <deployment-name>
```

```
kubectl apply -f deploy.yaml
```

```
kubectl apply -f deploy.yaml --record # to record the change-cause in revision history
```

```
kubectl rollout status deploy <deployment-name>
```

```
kubectl rollout history deploy <deployment-name>
```

```
kubectl rollout history deploy nginx --revision=2 # to get detailed history for a specific revision.
```

```
kubectl rollout undo deploy <deployment-name>
```

```
kubectl rollout undo deploy <deployment-name> --to-revision=1
```

```
#Pause the Deployment to apply multiple fixes and then resume it to start a new rollout.
```

```
#When the deployment is paused, no changes are recorded to revision history.
```

```
kubectl rollout pause deploy <deployment-name>
```

```
kubectl rollout resume deploy <deployment-name>
```

```
kubectl delete deploy <deployment-name>
```

```
# The value for labels in selector clause and pod template should match in deployment.
```

```
kubectl create deploy nginx --image=nginx --dry-run=client -o yaml > deploy.yaml
```

```
kubectl scale deploy <deployment-name> --replicas=3 # To scale up / down a deployment.
```

```
Not recorded in revision history.
```

```
kubectl create deploy <deployment-name> --image=redis -n <namespace-name>
```

```
# RollingUpdateStrategy in deployment details define upto how many pods can be down/up during the update at a time.
```

```
kubectl edit deploy <deployment-name> -n <namespace-name> # to make changes to a running deployment. Another option is to update the yaml file and apply it.
```

```
# Properties to remember
```

Learn 2025 DevOps/SRE tech stack www.heydevops.in

Our Placements <https://heydevops.in/success-stories>

More Project Videos <https://www.youtube.com/praveensingampalli>

Connect with me <https://www.instagram.com/singam4devops/>

spec.strategy.type==RollingUpdate|Recreate

spec.strategy.rollingUpdate.maxUnavailable

spec.strategy.rollingUpdate.maxSurge

#

strategy:

type: RollingUpdate

rollingUpdate:

maxUnavailable: 0

maxSurge: 1

keys follow camel case naming convention. values follow initials in upper case convention.

Deployment has a template.

Whenever you create a new deployment or update images in an existing deployment, a rollout is triggered.

A new rollout creates a new deployment revision.

You can locate a problematic deployment in cluster by checking the READY status for deployments. The no of containers running will be less than desired.

A deployment automatically creates a replicaset which in turn creates pods.

#####

HorizontalPodAutoScaler

#####

kubectl get hpa

kubectl delete hpa <hpa-name>

kubectl autoscale deploy nginx --min=5 --max=10 --cpu-percent=80

Learn 2025 DevOps/SRE tech stack www.heydevops.in

Our Placements <https://heydevops.in/success-stories>

More Project Videos <https://www.youtube.com/praveensingampalli>

Connect with me <https://www.instagram.com/singam4devops/>

#####

service

#####

kubectl apply -f <svc.yaml>

kubectl get svc

kubectl describe svc <service-name> # to get to know about port, target port etc.

kubectl delete svc <service-name>

Copy the label from template section of deployment and paste it under selector section of service.

If you are creating a frontend service for enabling external access to users, set the type to NodePort.

If you are tasked to create a service to enable the frontend pods to access a backend set of pods, set the type to ClusterIP.

In ClusterIP, port refers to port on service aka port exposed by service and targetPort refers to port on pod (container) aka port exposed by container.

LoadBalancer service type only works with supported cloud platforms like GCP, AWS, Azure. In an unsupported environment like VirtualBox, it would have the same effect as setting it to NodePort.

kubectl expose pod redis --port=6379 --name=redis-service --type=ClusterIP

--dry-run=client -o yaml > svc.yaml

kubectl expose pod nginx --port=80 --target-port=8080 --name=nginx-service

--type=NodePort --dry-run=client -o yaml > svc.yaml

kubectl expose deploy <deploy-name> --port=<>

Note: We can't set nodePort using imperative command. So for node port services, use yaml instead of command.

To get endpoints for a service

kubectl get ep <svc-name> # endpoint is nothing but the Pods to which the service is linked.

It contains ip of pod and port.

kubectl edit svc <svc-name> # to edit a service

services, deployments, replicaset and network policy have selector property under spec for selecting pods using labels.

There are 3 ports involved - node port, port and target port. node port is port on the node.

port is port on service. target port is port on pod. port is mandatory.

IP of service is known as Cluster-IP (internal ip).

Service can be accessed by pods using Cluster-IP or service name.

#####

Learn 2025 DevOps/SRE tech stack www.heydevops.in

Our Placements <https://heydevops.in/success-stories>

More Project Videos <https://www.youtube.com/praveensingampalli>

Connect with me <https://www.instagram.com/singam4devops/>

namespace

#####

kubectl get ns

kubectl create ns <namespace-name>

kubectl apply -f <namespace.yaml>

kubectl get po -n kube-system

kubectl describe ns <namespace-name>

kubectl delete ns <namespace-name>

To switch to a namespace permanently

kubectl config set-context \$(kubectl config current-context) --namespace=dev

To view pods in all namespaces

kubectl get po -A

To create a pod in a specific namespace

kubectl run redis --image=redis -n <namespace-name>

kubectl exec -it <pod-name> -n <namespace-name> -- sh

kubectl get all -A # To check all objects in all namespaces

kubectl create ns <namespace-name> --dry-run=client -o yaml

#####

ConfigMap

#####

kubectl get cm

kubectl describe cm <cm-name> # to check key-value pairs in config map

kubectl create cm <cm-name> --from-file=<path to file> # colon or equals to as delimiter between keys and values

kubectl create cm <cm-name> --from-file=<directory>

kubectl create cm <cm-name> --from-literal=<key1>=<value1>
--from-literal=<key2>=<value2>

kubectl apply -f cm.yaml

Properties to remember:

configmapkeyref / env, configMapRef / envFrom, volume

Learn 2025 DevOps/SRE tech stack www.heydevops.in

Our Placements <https://heydevops.in/success-stories>

More Project Videos <https://www.youtube.com/praveensingampalli>

Connect with me <https://www.instagram.com/singam4devops/>

Pods can consume ConfigMaps as environment variables or as configuration files in a volume mounted on one or more of its containers for the application to read.

When ConfigMap is created from a file (kubectl create cm <cm name> --from-file=) and when that ConfigMap is mounted as volume, then the entire file is available at mount point for the pod.

Injected into the Pod.

#####
Secrets
#####
kubectl get secrets

kubectl describe secret <secret-name> # This shows the attributes in secret but hides the values.

kubectl get secret <secret-name> -o yaml # To view the values (encoded).

If you enter the pod where secret is injected, you can see decoded values.

kubectl create secret generic <secret-name> --from-literal=<key1>=<value1> --from-literal=<key2>=<value2>

kubectl create secret generic <secret-name> --from-file=<path to file> # colon or equals to as delimiter between keys and values

Properties to remember:

secretkeyref / env, secretref / envFrom, volume

A secret can be injected into a pod as file in a volume mounted on one or more of its containers or as container environment variables.

opaque and service-account-token are secret types.

For encoding, decoding

echo -n 'string' | base64

echo -n 'encoded string' | base64 --decode

Learn 2025 DevOps/SRE tech stack www.heydevops.in

Our Placements <https://heydevops.in/success-stories>

More Project Videos <https://www.youtube.com/praveensingampalli>

Connect with me <https://www.instagram.com/singam4devops/>

While creating secret with the declarative approach (yaml), you must specify the secret key and value in encoded format.

When we create secret using imperative approach, secret keys and values are encoded on their own (and decoded as well).

Injected into the Pod.

#

<https://www.udemy.com/course/certified-kubernetes-administrator-with-practice-tests/learn/lecture/14827414#overview>

#####
#

Service Account

#####
#

kubectl create sa <sa-name>

kubectl get sa

kubectl describe sa <sa-name>

To fetch token from service account

kubectl describe sa <sa-name> # gives secret name

kubectl describe secret <secret-name> # gives token stored in secret

Create an nginx pod that uses 'myuser' as a service account

kubectl run nginx --image=nginx --serviceaccount=myuser --dry-run=client -o yaml > pod.yaml

kubectl apply -f pod.yaml

When we use service account inside the pod, the secret for that service account is mounted as volume inside the pod.

Learn 2025 DevOps/SRE tech stack www.heydevops.in

Our Placements <https://heydevops.in/success-stories>

More Project Videos <https://www.youtube.com/praveensingampalli>

Connect with me <https://www.instagram.com/singam4devops/>

Property to remember: spec -> serviceAccountName # set at pod level

Injected into the Pod.

For a deployment, can set service account in pod template.

A user makes a request to API server through kubectl using user account.

A process running inside a container makes a request to API server using service account.

A service account just like user account has certain permissions.

#####

Taints (on nodes) and Tolerations (on pods)

#####

To check taints on a node

kubectl describe no <node01> | grep -i "taint"

To create a taint on node01 with key of 'spray', value of 'mortein' and effect of 'NoSchedule'.

Taint effect defines what happens to pods that do not tolerate this taint.

Values for taint effect are NoSchedule, NoExecute and PreferNoSchedule.

kubectl taint no node01 spray=mortein:NoSchedule

The property tolerations under spec has properties like key, operator, value and effect and their values come inside "".

Remove the taint on master, which currently has the taint effect of NoSchedule

kubectl taint no master node-role.kubernetes.io/master:NoSchedule-

Remove from node 'foo' all the taints with key 'dedicated'

kubectl taint no foo dedicated-

#####

Logging

#####

The standard output of a container can be seen using the logs command.

kubectl logs -f <pod-name> <container-name> # follow the logs

Learn 2025 DevOps/SRE tech stack www.heydevops.in

Our Placements <https://heydevops.in/success-stories>

More Project Videos <https://www.youtube.com/praveensingampalli>

Connect with me <https://www.instagram.com/singam4devops/>

```
kubectl logs <pod-name> --previous # dump pod logs for a previous instantiation of a  
container
```

```
#####
```

Monitoring

```
#####
```

```
kubectl top no
```

```
kubectl top po
```

To get name of pod that is consuming most CPU.

```
kubectl top pod --namespace=default | head -2 | tail -1 | cut -d " " -f1
```

```
kubectl top po --sort-by cpu --no-headers
```

```
#####
```

Jobs

```
#####
```

```
kubectl create job busybox --image=busybox -- /bin/sh -c "echo hello;sleep 30;echo world"
```

```
kubectl get jobs
```

```
kubectl logs busybox-qhcnx # pod under job
```

```
kubectl delete job <job-name>
```

In case of pods, default value for restart property is Always and in case of jobs, default value for restart property is Never.

Job has pod template. Job has 2 spec sections - one for job and one for pod (in order).

A pod created by a job must have its restartPolicy be OnFailure or Never. If the restartPolicy is OnFailure, a failed container will be re-run on the same pod. If the restartPolicy is Never, a failed container will be re-run on a new pod.

Job properties to remember: completions, backoffLimit, parallelism, activeDeadlineSeconds, restartPolicy.

By default, pods in a job are created one after the other (sequence). Second pod is created only after the first one is finished.

Learn 2025 DevOps/SRE tech stack www.heydevops.in

Our Placements <https://heydevops.in/success-stories>

More Project Videos <https://www.youtube.com/praveensingampalli>

Connect with me <https://www.instagram.com/singam4devops/>

#####

CronJobs

#####

kubectl get cj

Create a cron job with image busybox that runs on a schedule and writes to standard output

kubectl create cj busybox --image=busybox --schedule="*/1 * * * *" -- /bin/sh -c "date; echo Hello from Kubernetes cluster"

In a cronjob, there are 2 templates - one for job and another for pod.

In a cronjob, there are 3 spec sections - one for cronjob, one for job and one for pod (in order).

Properties to remember: spec -> successfulJobHistoryLimit, spec -> failedJobHistoryLimit

#####

Ingress

#####

kubectl get ingress # To check details about Ingress Resource

kubectl describe ingress <>

kubectl edit ingress <ingress name>

kubectl apply -f <ingress.yaml>

Ingress setup requires an ingress controller (deployment), a node port ingress service (for accessing ingress controller from outside the cluster) and a config map.

They all are in same namespace.

The ingress resource (type is ingress), application(deployment) and service (for accessing deployment) are in different namespace.

In order for the Ingress resource to work, the cluster must have an ingress controller running.

Unlike other types of controllers which run as part of the kube-controller-manager binary, Ingress controllers are not started automatically with a cluster.

Kubernetes supports AWS, GCE and nginx ingress controllers.

Ingress resource defines rules and Ingress controller fulfills those rules.

Learn 2025 DevOps/SRE tech stack www.heydevops.in

Our Placements <https://heydevops.in/success-stories>

More Project Videos <https://www.youtube.com/praveensingampalli>

Connect with me <https://www.instagram.com/singam4devops/>

URL: <https://kubernetes.io/docs/tasks/access-application-cluster/ingress-minikube/>

<https://medium.com/@Oskarr3/setting-up-ingress-on-minikube-6ae825e98f82>

#####

Volumes

#####

kubectl get pv

kubectl get pvc

kubectl delete pvc <pvc-name>

kubectl delete pv <pv-name>

Properties to remember:

spec -> volumes -> name:

emptyDir: {}

spec -> containers -> volumeMounts -> name:

mountPath:

spec -> volumes -> name:

hostPath -> type: Directory

path:

spec -> containers -> volumeMounts -> name:

mountPath:

pvc is to be injected into the pod. Pods use pvc as volume and pod's containers mount that volume.

pvc remains in pending state until it is bound to a pv.

storageClassName and accessModes must match between pv and pvc. If no storage class is specified in PV, then there should be no storage class in PVC as well.
storage size must also be in range.

Learn 2025 DevOps/SRE tech stack www.heydevops.in

Our Placements <https://heydevops.in/success-stories>

More Project Videos <https://www.youtube.com/praveensingampalli>

Connect with me <https://www.instagram.com/singam4devops/>

persistent volumes have reclaim policy whose values are recycle (data in volume to be purged), retain (data and volume to be retained) and delete (volume to be deleted).

reclaim policy is invoked when pvc is deleted. once pvc is deleted, future of pv depends on reclaim policy.

property is spec -> persistentVolumeReclaimPolicy.

PVs use labels and PVCs use selectors for selecting the PVs.

PVs are cluster wide and PVCs are namespaced.

#####

Labels

#####

kubectl get po|deploy|all --show-labels

kubectl label po nginx env=lab # Add a label to a pod

kubectl label deploy my-webapp tier=frontend # Add a label to a deployment

kubectl label no node01 size=large # To label nodes

kubectl label po nginx env- # Remove the label

kubectl label po nginx env=lab1 --overwrite # to overwrite a label

Duplicate keys can't be used in labels.

#####

Selectors

#####

kubectl get po --selector=app=App1

kubectl get po --selector=app!=App1

kubectl get all --selector=env=prod

kubectl get po --selector=env=prod,bu=finance,tier=frontend # equivalent of && in programming languages

#####

Environment Variables

Learn 2025 DevOps/SRE tech stack www.heydevops.in

Our Placements <https://heydevops.in/success-stories>

More Project Videos <https://www.youtube.com/praveensingampalli>

Connect with me <https://www.instagram.com/singam4devops/>

#####

```
kubectl run nginx --image=nginx --env=app=web # Create an nginx pod and set an  
environment variable
```

env and envFrom property is an array.

```
# env property takes two properties - name and value. value takes only string and will  
always come in double quotes.
```

#####

Annotations

#####

```
kubectl annotate po nginx desc="Hello World"
```

```
kubectl annotate po nginx author=Avnish
```

```
kubectl annotate po nginx desc- # Remove this annotation from the pod
```

Duplicate keys in annotations are not allowed.

#####

Security Context

#####

```
# Properties to remember: securityContext -> runAsUser and securityContext -> capabilities  
-> add
```

Note: securityContext -> capabilities -> add property takes array as value. For example:

add:

```
- "NET_ADMIN"
```

```
- "SYS_TIME"
```

OR

```
add: ["NET_ADMIN", "SYS_TIME"]
```

Note: runAsUser takes only numeric values.

Learn 2025 DevOps/SRE tech stack www.heydevops.in

Our Placements <https://heydevops.in/success-stories>

More Project Videos <https://www.youtube.com/praveensingampalli>

Connect with me <https://www.instagram.com/singam4devops/>

Can be set at pod level as well as container level.

If you configure at pod level, the security settings will carry over to all the containers with in the pod.

If you configure at both pod level and container level, the settings on container will override the settings on pod.

#####
#

requests and limits | Resource Quotas

#####
#

We can set quotas for the total amount of memory and cpu that can be consumed by all the containers running in a namespace.

Quotas are specified using ResourceQuota object.

Once ResourceQuota is there, Every Container must have a memory request, memory limit, cpu request, and cpu limit.

We can configure default memory requests and limits for a namespace using LimitRange object in namespace.

If a Container is created in a namespace that has a default memory limit, and the Container does not specify its own memory limit, then the Container is assigned the default memory limit.

We can configure default CPU requests and limits for a namespace using LimitRange object.

If a Container is created in a namespace that has a default CPU limit, and the Container does not specify its own CPU limit, then the Container is assigned the default CPU limit.

Requests and Limits can be set for each of the containers in the Pod. If not set, they take default values.

kubectl run nginx --image=nginx --requests="cpu=100m,memory=256Mi"
--limits="cpu=200m,memory=512Mi"

cpu can also be specified as 1,2,0.1 etc. Here 1 count of cpu is equivalent to 1 vCPU in AWS or 1 GCP core. 0.1 count = 100m where m is milli.

It can go as low as 1m.

minimum usage through requests and maximum usage through limits.

Requests are what the container is guaranteed to get. If a container requests a resource, Kubernetes will only schedule it on a node that can give it that resource.

Limits, on the other hand, make sure a container never goes above a certain value. The container is only allowed to go up to the limit, and then it is restricted.

Resource quota is specified at namespace level. Resource limits (requests and limits) are specified at container level.

If a pod tries to exceed resources beyond its specified limit, then in case of cpu, Kubernetes throttles cpu.

Learn 2025 DevOps/SRE tech stack www.heydevops.in

Our Placements <https://heydevops.in/success-stories>

More Project Videos <https://www.youtube.com/praveensingampalli>

Connect with me <https://www.instagram.com/singam4devops/>

A container can't use more cpu resources than its limit. But a container can use more memory resources than its limit.

If a pod tries to use more memory resources than its limit constantly, pod will be terminated.

kubectl get quota

kubectl describe quota -n <namespace-name>

Good explanation in Golden book.

#####

Network Policy

#####

kubectl get netpol

kubectl describe netpol <name>

Note: While creating network policy, make sure that not only network policy is applied to the correct object but also that it allows access from (ingress) / to correct object (egress). labels and selectors are used.

An empty podSelector selects all pods in the namespace.

#####

Probes

#####

Set at the level of containers

readinessProbe is to check whether the application is ready.

livenessProbe is to check whether the application is live (running).

Properties to remember:

livenessProbe -> httpGet -> port

-> path

livenessProbe -> exec -> command

Learn 2025 DevOps/SRE tech stack www.heydevops.in

Our Placements <https://heydevops.in/success-stories>

More Project Videos <https://www.youtube.com/praveensingampalli>

Connect with me <https://www.instagram.com/singam4devops/>

livenessprobe -> tcpSocket -> port

livenessProbe -> initialDelaySeconds

livenessProbe -> PeriodSeconds

livenessProbe -> failureThreshold

readinessProbe -> httpGet -> port

-> path

readinessProbe -> exec -> command

readinessprobe -> tcpSocket -> port

readinessProbe -> initialDelaySeconds

readinessProbe -> PeriodSeconds

readinessProbe -> failureThreshold

#####
#####

Notes:

1.) securityContext -> runAsUser is available at pod as well as container level.

 securityContext -> capabilities -> add only available at container level.

2.) volumeMounts has properties like name, mountPath and readOnly (true|false)

4.) Deployment doesn't have good example in kubernetes.io, so remember this:

Learn 2025 DevOps/SRE tech stack www.heydevops.in

Our Placements <https://heydevops.in/success-stories>

More Project Videos <https://www.youtube.com/praveensingampalli>

Connect with me <https://www.instagram.com/singam4devops/>

spec:

replicas: 5

strategy:

type: RollingUpdate

rollingUpdate:

maxSurge: 7

maxUnavailable: 3

5.) The labels in pod template and selector clause in deployment must match.

6.) If you are using a port in readiness probe or liveness probe, then that port must be exposed.

7.) If unit for cpu request is given as 200m then use that ("200m"). If it is given as .2, then use that ("0.2")

8.) name in volumes and volumeMounts must be same.

9.) env is used with configMapKeyRef. In doc, search by configMapKeyRef. name property of env is used as such. Used to import one key from configmap.

envFrom is used with configMapRef. In doc, search by configMapRef. Used to import the entire configmap.

When configmap is mounted as volume, then no need to think of env or envFrom. Just think of volumes and volumeMounts.

10.) env is used with secretKeyRef. In doc, search by secreyKeyRef. name property of env is used as such. Used to import one key from secret.

envFrom is used with seretRef. In doc, search by secretRef. used to import the entire secret.

When secret is mounted as volume, then no need to think of env or envFrom. Just think of volumes and volumeMounts.

11.) To check connectivity, you can use curl or netcat.

nc -v -w 2 -z ip port

curl http://ip:port