AWS Cheat Sheet

Installation

See https://docs.aws.amazon.com/cli/latest/userguide/getting-started-install.html

For autocompletion see https://docs.aws.amazon.com/cli/latest/userguide/cli-configure-completion.html.

Global flags

Flag	Description
--profile <name>	Run a command using the specified profile
--output <style>	Format output as (json/text/table/yaml)
--help	Show help
--endpoint-url <url>	Override default API endpoint
--region	Override default region
--version	Show version of AWS CLI

Configuration

Command	Description
aws configure	Configure AWS
aws configure sso	Configure AWS IAM Identity Center credentials
aws configure list	Show current configuration data
aws configure set <key> <value>	Set a configuration value
aws configure get <key>	Show a configuration value

Profiles

Command	Description
aws configure --profile <name>	Create a new profile
aws configure list-profiles	Show available profiles

Account information

Command	Description
aws sts get-caller-identity	Retrieve account information

Identity and Access Management (IAM)

Command	Description
aws iam list-users	List users
aws iam list-groups	List groups
aws iam list-roles	List roles
aws iam list-policies	List policies
aws iam create-user --user-name <name>	Create a user
aws iam create-group --group-name <name>	Create a group
aws iam create-role --role-name <name> --assume-role-policy-document <policy>	Create a role
aws iam create-policy --policy-name <name> --policy-document <policy>	Create a policy

Elastic Compute Cloud (EC2)

Command	Description
aws ec2 describe-instances	List instances
aws ec2 run-instances --image-id <id> --instance-type <type>	Create an instance
aws ec2 terminate-instances --instance-ids <id>	Terminate an instance
aws ec2 start-instances --instance-ids <id>	Start an instance
aws ec2 stop-instances --instance-ids <id>	Stop an instance

Elastic Kubernetes Service (EKS)

Command	Description
aws eks list-clusters	List clusters
aws eks describe-cluster --name <name>	Describe a cluster
aws eks create-cluster --name <name>	Create a cluster
aws eks delete-cluster --name <name>	Delete a cluster

Relational Database Service (RDS)

Command	Description
`aws rds describe-db-clusters`	List clusters
`aws rds describe-db-instances`	List instances
`aws rds create-db-cluster-snapshot`	Create a snapshot of a cluster

Simple Storage Service (S3)

Command	Description
`aws s3 ls`	List buckets
`aws s3 ls s3://<bucket>`	List objects in a bucket
`aws s3 mb s3://<bucket>`	Create a bucket
`aws s3 rb s3://<bucket>`	Delete a bucket
`aws s3 cp <src> <dest>`	Copy a file from/to S3
`aws s3 rm s3://<bucket>/<object>`	Delete an object
`aws s3 sync s3://bucket <dir>`	Sync a bucket to a local directory
`aws s3 sync <dir> s3://bucket`	Sync a local directory to an S3 bucket

Other commands

Command	Description
`aws --version`	Show version
`aws help`	Show help including all available commands
`aws <command> help`	Show help for a specific command

Docker Cheat Sheet

Installation

Install Docker Engine (https://docs.docker.com/engine/install/, Linux only) or Docker Desktop (https://docs.docker.com/desktop/, Linux, macOS and Windows).

Container commands

Command	Description
`docker run <image>`	Create and run a new container
`docker run -p 8080:80 <image>`	Publish container port 80 to host port 8080
`docker run -d <image>`	Run a container in the background
`docker run -v <host>:<container> <image>`	Mount a host directory to a container
`docker ps`	List currently running containers
`docker ps --all`	List all containers (running or stopped)
`docker logs <container_name>`	Fetch the logs of a container
`docker logs -f <container_name>`	Fetch and follow the logs of a container
`docker stop <container_name>`	Stop a running container
`docker start <container_name>`	Start a stopped container
`docker rm <container_name>`	Remove a container

Executing commands in a container

Command	Description
`docker exec <container_name> <command>`	Execute a command in a running container
`docker exec -it <container_name> bash`	Open a shell in a running container

Image commands

Command	Description
`docker build -t <image> .`	Build a new image from the Dockerfile in the current directory and tag it
`docker images`	List local images
`docker rmi <image>`	Remove an image

Container registry commands

Command	Description
`docker login`	Login to Docker Hub
`docker login <server>`	Login to another container registry
`docker logout`	Logout of Docker Hub
`docker logout <server>`	Logout of another container registry
`docker push <image>`	Upload an image to a registry
`docker pull <image>`	Download an image from a registry
`docker search <image>`	Search Docker Hub for images

System commands

Command	Description
`docker system df`	Show Docker disk usage
`docker system prune`	Remove unused data
`docker system prune -a`	Remove all unused data

Docker Compose

Command	Description
`docker compose up`	Create and start containers
`docker compose up -d`	Create and start containers in background
`docker compose up --build`	Rebuild images before starting containers
`docker compose stop`	Stop services
`docker compose down`	Stop and remove containers and networks
`docker compose ps`	List running containers
`docker compose logs`	View the logs of all containers
`docker compose logs <service>`	View the logs of a specific service
`docker compose logs -f`	View and follow the logs
`docker compose pull`	Pull the latest images
`docker compose build`	Build or rebuild services
`docker compose build --pull`	Pull latest images before building

Dockerfile instructions

Instruction	Description
`FROM <image>`	Set the base image
`FROM <image> AS <name>`	Set the base image and name the build stage
`RUN <command>`	Execute a command as part of the build process
`RUN ["exec", "param1", "param2"]`	Execute a command as part of the build process
`CMD ["exec", "param1", "param2"]`	Execute a command when the container starts
`ENTRYPOINT ["exec", "param1"]`	Configure the container to run as an executable
`ENV <key>=<value>`	Set an environment variable
`EXPOSE <port>`	Expose a port
`COPY <src> <dest>`	Copy files from source to destination
`COPY --from=<name> <src> <dest>`	Copy files from a build stage to destination
`WORKDIR <path>`	Set the working directory
`VOLUME <path>`	Create a mount point
`USER <user>`	Set the user
`ARG <name>`	Define a build argument
`ARG <name>=<default>`	Define a build argument with a default value
`LABEL <key>=<value>`	Set a metadata label
`HEALTHCHECK <command>`	Set a healthcheck command

See https://docs.docker.com/engine/reference/builder/ for the full Dockerfile reference.

Example compose.yaml

Note: this file used to be called `docker-compose.yaml`, but now `compose.yaml` is preferred.

```
services:
  service1:
    image: <image>
    build: .
    volumes:
      - .:/code:ro
    ports:
      - "8000:80"
    environment:
      KEY: value
```

Docker Compose file reference

Key	Description
`name`	Set the name of the project
`services`	A list of services defined in the file
`services.<name>.image`	Set the image to use or build
`services.<name>.build`	Build context and options
`services.<name>.build.context`	Build context (default is the current directory)
`services.<name>.build.dockerfile`	Dockerfile to use (default is `Dockerfile`)
`services.<name>.build.target`	Build stage to use
`services.<name>.build.args`	Build arguments
`services.<name>.command`	Override the default command for the container
`services.<name>.entrypoint`	Override the default entrypoint for the container
`services.<name>.volumes`	Mount volumes in the container
`services.<name>.ports`	Publish container ports to the host
`services.<name>.environment`	Set environment variables in the container
`services.<name>.restart`	Restart policy (no/always/on-failure/unless-stopped)
`services.<name>.scale`	Set the number of containers to run
`services.<name>.networks`	List of networks to connect the container to
`services.<name>.depends_on`	List of services to start before this service
`services.<name>.labels`	Set metadata labels for the container
`networks`	A list of networks defined in the file
`networks.<name>.driver`	Set the network driver
`networks.<name>.external`	Don't create the network, use an existing one
`volumes`	A list of volumes defined in the file
`volumes.<name>.name`	Set the name of the volume
`volumes.<name>.driver`	Set the volume driver
`configs`	A list of configs defined in the file
`secrets`	A list of secrets defined in the file

See https://docs.docker.com/compose/compose-file/ for the full Docker Compose file reference.

Git Cheat Sheet

Workflow
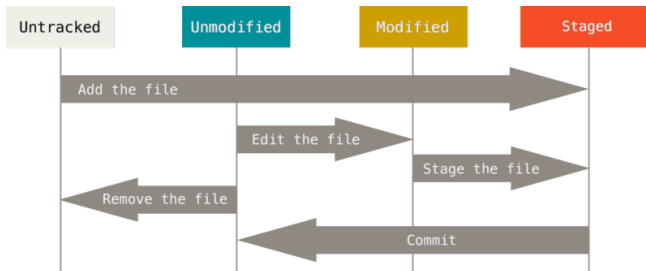
Start a new repository

Command	Description
git init	Create a new repository in the current directory
git clone <url> [<dir>]	Clone a remote repository

History

Command	Description
git log	Show commit logs
git log --oneline	Show commit logs with one line per commit
git show	Show the changes of the last commit
git shortlog	Summarize git log output by author
git blame <file>	Show who changed which line in a file

Local changes

Command	Description
git status	List which files are staged, unstaged, and untracked
git add <file>	Add a file to the staging area
git add -p	Interactively choose hunks of a file to add to the staging area

Local changes (cont.)

Command	Description
git diff	Show unstaged changes
git diff --staged	Show staged changes
git commit	Commit staged changes using the default editor
git commit -m <message>	Commit staged changes with a message provided on the command line
git commit --amend	Amend the last commit
git reset HEAD <file>	Unstage a file
git checkout -- <file>	Discard local changes in a file

Branches

Command	Description
git branch	List local branches
git branch -r	List remote branches
git branch <branch>	Create a new branch
git checkout <branch>	Switch to a branch
git checkout -b <branch>	Create a new branch and switch to it
git merge <branch>	Merge a branch into the active branch
git branch -d <branch>	Delete a branch

Tags

Command	Description
git tag	List tags
git tag <tag>	Create a tag
git tag -a <tag>	Create an annotated tag
git tag -s <tag>	Create a GPG-signed tag
git tag -d <tag>	Delete a tag
git show <tag>	Show the tag data

Collaboration

Command	Description
git remote -v	List remote repositories
git remote add <name> <url>	Add a new remote repository
git fetch <remote>	Download all changes from a remote repository but don't integrate them
git pull <remote>	Download changes and directly merge/integrate them
git pull <remote> <branch>	Download changes and directly merge/integrate a remote branch
git push <remote> <branch>	Upload local changes to a remote repository
git push <remote> --delete <branch>	Delete a remote branch
git push <remote> --tags	Push tags to a remote repository

Stashing

Command	Description
git stash	Stash the changes in a dirty working directory away
git stash list	List all stashes
git stash pop	Apply the last stash and delete it
git stash apply	Apply the last stash but don't delete it
git stash drop	Delete the last stash
git stash clear	Delete all stashes

Writing good commit messages

1. Separate subject from body with a blank line
2. Limit the subject line to 50 characters
3. Capitalize the subject line
4. Do not end the subject line with a period
5. Use the imperative mood in the subject line
6. Wrap the body at 72 characters
7. Use the body to explain what and why vs. how

Helm Cheat Sheet

Helm

Helm is the package manager for Kubernetes. See https://helm.sh/docs/intro/install/ for installation instructions.

Global flags

Flag	Description
`--kube-context <name>`	Name of the Kubernetes context to use
`--namespace <name>`	Namespace to use for this operation

Repository management

Command	Description
`helm repo add <name> <url>`	Add a repository
`helm repo list`	List all added repositories
`helm repo update`	Update the local cache of available charts
`helm repo remove <name>`	Remove a repository
`helm search repo`	List all charts in the repositories
`helm search repo <keyword>`	Search for a chart in the repositories

Installing Helm charts

Command	Description
`helm install <name> <chart>`	Install a chart with a name
`helm install <chart> --generate-name`	Install a chart, auto-generating a name
`helm install <name> <chart> --namespace <namespace>`	Install a chart in a specific namespace
`helm install <name> <chart> --set <key>=<value>`	Install a chart with specific values
`helm install <name> <chart> --values <file>`	Install a chart using a values file
`helm install <name> <chart> --dry-run --debug`	Run a test installation to validate the chart
`helm install <name> <chart> --verify`	Verify the package before installing
`helm install <name> <chart> --dependency-update`	Update dependencies before installing
`helm uninstall <name>`	Uninstall a release
`helm uninstall <name> --keep-history`	Uninstall a release, keeping the history

Listing releases

Command	Description
`helm list`	List all releases in the current namespace
`helm list --all-namespaces`	List all releases in all namespaces
`helm list -l <label>=<value>`	List releases with a specific label
`helm list --date`	List releases sorted by date
`helm list --(pending|failed|uninstalled)`	List releases that are in a pending/failed/uninstalled state
`helm status <name>`	Show the status of a release

Managing releases

Command	Description
`helm upgrade <name> <chart>`	Upgrade a release
`helm upgrade <name> <chart> --atomic`	Upgrade a release atomically
`helm upgrade <name> <chart> --dependency-update`	Upgrade a release and update dependencies
`helm upgrade <name> <chart> --version <version>`	Upgrade a release to a specific version
`helm upgrade <name> <chart> --set <key>=<value>`	Upgrade a release with specific values
`helm rollback <release> <revision>`	Rollback a release to a previous revision

Developing charts

Command	Description
`helm create <name>`	Create a new chart
`helm package <chart-path>`	Package a chart directory into a chart file
`helm lint <chart>`	Lint a chart
`helm show all <chart>`	Inspect a chart and list all resources
`helm show values <chart>`	Inspect a chart and show default values
`helm template <name> <chart>`	Render templates locally
`helm template <name> <chart> --set <key>=<value>`	Render templates locally and override values

HTTP Cheat Sheet

Example Request & Response

```
GET /hello.txt HTTP/1.1
Host: www.example.com
User-Agent: Mozilla/5.0
Accept-Language: en, nl
```

```
HTTP/1.1 200 OK
Date: Mon, 27 Jul 2020 12:28:53 GMT
Server: Apache/2.2.22
Content-Length: 12
Content-Type: text/plain

Hello World!
```

Methods

Method	Description
GET (safe, idempotent)	Transfer a current representation of the resource
HEAD (safe, idempotent)	Same as GET, but without transferring content
POST	Process the content of the request
PUT (idempotent)	Update the resource with the content of the request
DELETE (idempotent)	Remove all current representations of the resource
CONNECT	Establish a tunnel to the server
OPTIONS (safe, idempotent)	Describe the communication options for the target resource
TRACE (safe, idempotent)	Perform a message loop-back test along the path to the resource

"Safe" methods are read-only and should not change the server state. "Idempotent" methods can be called multiple times without different outcomes. See RFC 9110 for detailed information about each method.

Request Headers

Header	Description
Referer: <url>	The referring web page URL
User-Agent: <ua>	Information about the client making the request
Authorization: <scheme> <params>	Authenticate using the specified scheme and credentials
Accept: <media-types>	The media types that the client can understand
Accept-Encoding: <algorithms>	The encoding algorithms the client understands
Accept-Language: <langs>	The languages that the client can understand

Response Headers

Header	Description
Location: <url>	The URL of the new resource or redirect target
Allow: <methods>	The methods supported by the resource
Server: <name>	Information about the server software
Retry-After: <s>	Time to wait before a new request
Age: <s>	The age of the cached response (in sec)
Cache-Control: <directives>	Directives for caching mechanisms
Expires: <date>	Date/time after which the response expires

See RFC 9111 for detailed information about caching.

Status Codes

A detailed list of status codes can be found in RFC 9110.

Status Codes: Informational

Code	Description
100 Continue	The server has received the request headers and the client should proceed
101 Switching Protocols	The server will switch protocols to those defined by the response's Upgrade header field

Status Codes: Success

Code	Description
200 OK	The request has succeeded
201 Created	A new resource has been created
202 Accepted	The request was accepted but processing has not been completed
203 Non-Authoritative Information	The response content has been modified by a proxy
204 No Content	The request was fulfilled but no content is returned
205 Reset Content	Request fulfilled; the user agent should reset the view
206 Partial Content	The server has fulfilled a partial GET request

Status Codes: Redirection

Code	Description
300 Multiple Choices	The requested resource corresponds to any one of a set of representations
301 Moved Permanently	The requested resource has been assigned a new permanent URI
302 Found	The requested resource resides temporarily under a different URI

Status Codes: Redirection (cont.)

Code	Description
303 See Other	A different resource can provide an indirect reponse
304 Not Modified	The resource has not been modified since the last request
307 Temporary Redirect	Temporary redirect; client must not change the request method
308 Permanent Redirect	Permanent redirect; client must not change the request method

Status Codes: Client Error

Code	Description
400 Bad Request	The server cannot / will not process the request due to a client error
401 Unauthorized	The request requires user authentication
403 Forbidden	The server understood but refuses the request
404 Not Found	No resource was found for the request URI
405 Method Not Allowed	The method is not allowed for the resource
406 Not Acceptable	No representation matches the `Accept-*` headers
407 Proxy Authentication Required	The client must authenticate with the proxy server
408 Request Timeout	The server timed out waiting for the request
409 Conflict	The request conflicts with the current resource state
410 Gone	The requested resource is permanently unavailable
411 Length Required	The server requires a `Content-Length` header
412 Precondition Failed	One or more conditions evaluated to false
413 Content Too Large	The request content is too large
414 URI Too Long	The target URI was too long

Status Codes: Client Error (cont.)

Code	Description
415 Unsupported Media Type	The requested media type is not supported
416 Range Not Satisfiable	The request Range header is not satisfiable
417 Expectation Failed	The server cannot meet the `Expect` header
421 Misdirected Request	The request was directed at a server that is not able to produce a response for this request
422 Unprocessable Content	The server understands the content type but cannot process the content
426 Upgrade Required	The client should switch to a different protocol

Status Codes: Server Error

Code	Description
500 Internal Server Error	The server encountered an unexpected error
501 Not Implemented	The server does not support the functionality required to fulfill the request
502 Bad Gateway	The upstream server returned an invalid response
503 Service Unavailable	The server is currently unable to handle the request
504 Gateway Timeout	The upstream server did not respond in time
505 HTTP Version Not Supported	The server does not support the HTTP version

Kubernetes Cheat Sheet

Installation

Install the `kubectl` command line tool to interact with the Kubernetes API:
https://kubernetes.io/docs/tasks/tools/#kubectl

Enable autocompletion in bash:

```
composer completion bash | sudo tee /etc/bash_completion.d/kubectl
```

Global flags

Flag	Description
`--namespace <namespace>`	The name of the namespace to use
`--context <context>`	The name of the context to use
`--help`	Show information about a given command

Context and configuration

Command	Description
`kubectl config get-contexts`	List all contexts
`kubectl config current-context`	Display the current context
`kubectl config use-context <context>`	Switch to another context
`kubectl config delete-context <context>`	Delete the specified context from the kubeconfig

Display resources

Command	Description
`kubectl get <resource>`	List all resources of this type in the current namespace
`kubectl get <resource> -o wide`	List all resources with more details
`kubectl get <resource> -A`	List all resources of this type in all namespaces
`kubectl get <resource> <name>`	List a particular resource
`kubectl get <resource> <name> -o yaml`	Print a particular resource in YAML format
`kubectl get <resource> <name> -l <key1>=<value1>`	List resources where label `<key1>` contains `<value1>`
`kubectl describe <resource>`	Show detailed information about a resource

Apply configuration manifests

Command	Description
`kubectl apply -f <file>`	Apply a manifest from a file
`kubectl apply -f <dir>`	Apply all manifests in a directory
`kubectl apply -k <dir>`	Apply resources from a kustomize directory

Create resources manually

Command	Description
`kubectl run <name> --image=<image>`	Start a pod
`kubectl create deployment <name> --image=<image>`	Create a deployment
`kubectl expose pod <pod> --port=<port>`	Create a service for an existing pod
`kubectl expose deployment <name> --port=<port>`	Create a service for an existing deployment
`kubectl create ingress <name> --rule=<host/path=svc:port>`	Create an ingress that routes traffic to a service
`kubectl create job <name> --image=<image>`	Create a job
`kubectl create job <name> --from=cronjob/<name>`	Create a job from a cronjob
`kubectl create cronjob <name> --image=<image> --schedule=<schedule>`	Create a cronjob, using a schedule in Cron format
`kubectl create secret generic <name> --from-literal=<key>=<value>`	Create a secret containing `<key>` and `<value>`
`kubectl create secret docker-registry <name> --docker-server=<server> --docker-username=<username> --docker-password=<password>`	Create a secret for a Docker registry

Generate YAML configuration manifests

Command	Description
`kubectl create deployment <name> --image=<image> --dry-run=client -o yaml`	Generate a deployment manifest
`kubectl expose deployment <name> --port=<port> --dry-run=client -o yaml`	Generate a service manifest for a deployment

Edit resources

Command	Description
`kubectl edit <resource> <name>`	Edit a resource in a text editor
`kubectl set image <resource> <name> <container>=<image>`	Update the image of a container in a pod

Set labels and annotations

Command	Description
`kubectl label <resource> <name> <key>=<value>`	Add a label to a resource
`kubectl annotate <resource> <name> <key>=<value>`	Add an annotation to a resource

Delete resources

Command	Description
`kubectl delete <resource> <name>`	Delete a particular resource
`kubectl delete <resource> --all`	Delete all resources of a particular type in the current namespace
`kubectl delete -f <file>`	Delete a resource from a file

Manage deployments

Command	Description
`kubectl rollout status deployment <name>`	Show the status of a deployment rollout
`kubectl rollout history deployment <name>`	View the rollout history of a deployment
`kubectl rollout undo deployment <name>`	Undo a previous rollout deployment
`kubectl rollout restart deployment <name>`	Restart a deployment
`kubectl scale deployment <name> --replicas=<n>`	Scale a deployment to <n> replicas
`kubectl autoscale deployment <name> --min=<min> --max=<max>`	Autoscale a deployment between <n> and <n> replicas

Execute commands

Command	Description
`kubectl exec <pod> -- <command>`	Execute a command in a running pod
`kubectl exec -it <pod> -- sh`	Open a shell in a running pod

View logs

Command	Description
`kubectl logs <pod>`	Print the logs for a pod
`kubectl logs -f <pod>`	Print the logs for a pod and keep streaming

Resource usage

Command	Description
`kubectl top node`	Show resource (CPU/memory) usage of nodes
`kubectl top pod`	Show resource (CPU/memory) usage of pods

Other commands

Command	Description
`kubectl version`	Show the version of the client and server
`kubectl api-resources`	Print the supported API resources on the server

Linux Cheat Sheet

File Commands

Command	Description
cat <file>	Print file contents
touch <file>	Create file (or update timestamp)
cp <src> <dst>	Copy a file
mv <src> <dst>	Move / rename a file
rm <file>	Remove file
head <file>	Print first 10 lines of file
tail <file>	Print last 10 lines of file
tail -f <file>	Print last 10 lines and any new lines added to file

Directory Commands

Command	Description
ls	List files in the current directory
ls <dir>	List files in <dir>
ls -l	List files with details
ls -a	List all files (including hidden)
ls -al	List all files with details
pwd	Print the current/working directory
cd <dir>	Change current directory to <dir>
cd ..	Go up one directory
mkdir <dir>	Create directory
mkdir -p <dir>	Create directory and any missing parent directories
cp -r <src> <dst>	Copy a directory and its contents
mv <src> <dst>	Move / rename a directory
rmdir <dir>	Remove an empty directory
rm -r <dir>	Remove a directory and its contents

Directory Aliases

Alias	Directory
.	Current directory
..	Parent directory
~	Home directory

Link Commands

Command	Description
ln -s <target> <link>	Create symbolic link
ln <target> <link>	Create hard link
readlink -f <file>	Print absolute path of file

Search Commands

Command	Description
grep <pattern> <file...>	Search for pattern in file(s)
grep -r <pattern> <dir>	Search for pattern recursively in a directory
grep -i <pattern> <file>	Search for pattern case-insensitively
grep -v <pattern> <file>	Search for lines not matching pattern
grep -n <pattern> <file>	Search and print line numbers
find <dir> -name <name>*	Find files in <dir> with name starting with <name>
find <dir> -type <type>	Find files in <dir> of type <type>

Superuser Commands

Command	Description
sudo <cmd>	Run command as superuser (root)
su	Switch to superuser (root)
su <user>	Switch to user <user>

File Permission Commands

Command	Description
chmod <mode> <file>	Change file permissions
chmod u+x <file>	Add execute permission for the owner
chmod 644 <file>	Set file permissions to 644 (read/write for owner, read for group and everyone else)
chmod -R g+w <dir>	Add write permission for the group recursively to a directory
chown <owner> <file>	Change file owner
chown -R <owner> <dir>	Change directory owner recursively
chown <owner>:<group> <file>	Change file owner and group
chgrp <group> <file>	Change file group

File Permission Bits

Bit	Description
4	Read (r)
2	Write (w)
1	Execute (x)

First digit represents the owner, the second digit represents the group, and the third digit represents everyone else.

For example, 640 means read/write for owner, read for group, and no permissions for everyone else.

File Permission User Types

User Type	Description
u	Owner
g	Group
o	Everyone else
a	All

System Information Commands

Command	Description
df	Show file system disk usage
df -h	Show disk usage in MB/GB/TB
du	Show disk usage per directory
du -h	Show disk usage in MB/GB/TB
free	Show memory usage
free -h	Show memory usage in MB/GB/TB
uptime	Show uptime
whoami	Show current user
uname	Show system information
uname -a	Show all system information

Shutdown/Reboot Commands

Command	Description
shutdown	Power off the system after 1 minute
shutdown now	Power off the system immediately
shutdown 16:30	Power off the system at 16:30
shutdown -c	Cancel a scheduled shutdown
reboot	Reboot system
poweroff	Power off system

Process Commands

Command	Description
ps	List processes
ps -ef	List all processes
top	Display processes in real time
kill <pid>	Kill process
pkill <name>	Kill process with name
killall <name>	Kill all processes with name

Secure Shell (SSH) Commands

Command	Description
ssh-keygen	Generate SSH key pair
ssh-add <key>	Add SSH key to SSH agent
ssh-add -l	List SSH keys in SSH agent
ssh-copy-id <user>@<host>	Copy SSH key to <host> for <user>
ssh <host>	Connect to <host> as current user
ssh <user>@<host>	Connect to <host> as <user>
scp <src> <host>:<dest>	Copy local file <src> to <dest> on <host>
scp <host>:<src> <dest>	Copy file <src> from <host> to local <dest>

APT Commands (Debian/Ubuntu)

Command	Description
apt install <pkg>	Install package
apt remove <pkg>	Remove package
apt purge <pkg>	Remove package and config files
apt update	Update package list
apt upgrade	Upgrade packages
apt search <term>	Search for package

Other Commands

Command	Description
echo <msg>	Print message to stdout
man <cmd>	Display manual page for <cmd>
history	Display history of given commands
which <cmd>	Find path to executable <cmd>
whereis <cmd>	Locate binary, source, and manual page for <cmd>

IO Redirection

Command	Description
<cmd> > <file>	Replace contents of file with output
<cmd> >> <file>	Append output to file
<cmd> < <file>	Redirect input from file
<cmd> 2> <file>	Redirect stderr to file
<cmd> &> <file>	Redirect stdout and stderr to file

Pipe Commands

Command	Description
<cmd1>	<cmd2>	Pipe stdout to next cmd
<cmd1> |& <cmd2>	Pipe stderr to next cmd

Bash Shortcuts

Shortcut	Description
Ctrl + C	Stop current command
Ctrl + A	Go to beginning of line
Ctrl + E	Go to end of line
Ctrl + K	Cut current line
Ctrl + L	Clear screen

Directory Structure

Directory	Description
/	Root
/bin	Binary or executable programs
/etc	System configuration files
/home	User home directories
/opt	Optional or third-party software
/tmp	Temporary files
/usr	User programs
/var	Variable data
/var/log	Log files

Terraform Cheat Sheet

Installation

See https://developer.hashicorp.com/terraform/downloads for installation instructions for your platform. Then run `terraform -install-autocomplete` to enable shell autocompletion.

Initializing Terraform

Command	Description
`terraform init`	Prepare your working directory for other commands
`terraform init -upgrade`	Upgrade modules/providers to the latest allowed versions
`terraform get`	Only download and install modules

Making infrastructure changes

Command	Description
`terraform plan`	Show changes required by the current configuration
`terraform plan -out=<file>`	Write the plan to a file to apply it later
`terraform plan -target <resource>`	Create a plan for a specific module or resource
`terraform plan -replace <resource>`	Force the plan to replace a specific resource
`terraform plan -var '<key>=<value>'`	Set a value for one of the input variables
`terraform plan -refresh-only`	Inspect resource drift without updating the state file
`terraform apply`	Create or update infrastructure
`terraform apply <file>`	Create or update infrastructure using a plan file
`terraform apply -target <resource>`	Create or update a specific resource
`terraform apply -replace <resource>`	Force the replacement of a specific resource
`terraform apply -auto-approve`	Skip interactive approval of plan before applying

Inspecting output values

Command	Description
`terraform output`	Show all output values
`terraform output -json`	Show all output values in JSON format
`terraform output <name>`	Show a specific output value
`terraform output -raw <name>`	Show a specific output value without quotes

Managing state

Command	Description
`terraform show`	Show the current state in human-readable form
`terraform show <file>`	Show a human-readable state or plan file
`terraform show -json <file>`	Show a state or plan file in JSON format
`terraform state list`	List all resources in the state file
`terraform state show <resource>`	Show details about a resource
`terraform state mv <source> <dest>`	Rename a resource in the state file
`terraform state rm <resource>`	Remove a resource from the state file
`terraform state replace-provider <from> <to>`	Replace provider for resources in the state
`terraform import <resource> <remote_id>`	Import existing infrastructure into Terraform
`terraform state pull`	Pull current state and output to stdout
`terraform refresh`	(Deprecated) Update state to match reality

Tainting resources

Command	Description
`terraform taint <resource>`	(Deprecated) Mark a resource to be replaced
`terraform untaint <resource>`	Mark a resource as no longer tainted

Destroying infrastructure

Command	Description
`terraform destroy`	Destroy infrastructure managed by Terraform
`terraform destroy -target <resource>`	Destroy a specific resource

Formatting and validation

Command	Description
`terraform validate`	Check whether the configuration is valid
`terraform fmt`	Reformat your configuration in the standard style
`terraform fmt -check`	Check whether the configuration is formatted correctly, return non-zero exit code if not

Terraform Cloud / Remote Authentication

Command	Description
terraform login	Log in to Terraform Cloud
terraform login <hostname>	Log in to a different host
terraform logout	Log out of Terraform Cloud
terraform logout <hostname>	Log out of a different host

Managing workspaces

Command	Description
terraform workspace list	List all existing workspaces
terraform workspace show	Show the name of the current workspace
terraform workspace select <name>	Select a different workspace
terraform workspace new <name>	Create a new workspace
terraform workspace delete <name>	Delete an existing workspace

Other commands

Command	Description
terraform providers	Show the providers required for this configuration
terraform force-unlock <lock-id>	Release a stuck lock
terraform console	Try Terraform expressions at an interactive prompt
terraform graph | dot -Tpng > graph.png	Generate a visual graph of Terraform resources
terraform version	Show the current Terraform version
terraform -help	Show help output for Terraform
terraform -help <command>	Show help output for a specific Terraform command

Referencing Named Values

Syntax	Description
<RESOURCE_TYPE>.<NAME>	Reference to a managed resource
var.<NAME>	Reference to an input variable
local.<NAME>	Reference to a local value
module.<MODULE NAME>	Reference to a child module
data.<DATA TYPE>.<NAME>	Reference to a data source

Conditional Expressions

Syntax	Description
condition ? true : false	If condition is true, return true, otherwise return false

Splat Expressions

Syntax	Description
<RESOURCE_TYPE>.<NAME>[*].<ATTRIBUTE>	Return a list of values for the given attribute of all instances of a resource

Resource Meta-Arguments

Argument	Description
depends_on	Explicitly specify resource dependencies
count	Create multiple instances of a resource
for_each	Create an instance of a resource for each element in a map or set
provider	Specify a provider configuration block to use for this resource
lifecycle	Configure the behavior of a resource over its lifetime

Lifecycle Meta-Argument Attributes

Argument	Description
create_before_destroy	Create the new resource before destroying the old one
prevent_destroy	Prevent Terraform from destroying the resource
ignore_changes	Ignore changes to specific resource attributes