AWS CLI Amazon cloud CLI

START HERE

`aws configure`
Set up credentials

`aws sts get-caller-identity`
Who am I?

`aws s3 ls`
List buckets

`aws s3 cp <file> s3://<bucket>/`
Upload a file

`aws ec2 describe-instances`
List instances

`aws <svc> help`
Read the docs

`--profile <name>`
Pick an account

`--query '<jmespath>'`
Filter the output

Command aws ec2 describe-... → `profile + region` → **Credentials** env, config, SSO, role → `signed request` → **AWS API** JSON response → `--query, --output` → **Your terminal** json, table, text

CONCEPTS

Shape: aws <service> <operation> [--options]. Add help at any level. Always check sts get-caller-identity before changing anything.

This sheet covers AWS CLI v2. Output goes through a pager by default; AWS_PAGER="" or --no-cli-pager turns it off for scripts.

SETUP AND PROFILES

`aws configure`
Prompt for keys, region and output

`aws configure --profile <name>`
Create or edit a named profile

`aws configure sso`
Set up an IAM Identity Center profile

`aws sso login --profile <name>`
Sign in through the browser

`aws configure list`
Show where each setting comes from

`aws configure get <key>`
Read one config value

Example: aws configure get region --profile prod

`aws configure set <key> <value>`
Write one config value

`aws configure list-profiles`
List all profile names

`export AWS_PROFILE=<name>`
Use a profile for the whole shell

`aws sts get-caller-identity`
Show the account and identity in use

`aws sso logout`
Drop cached SSO tokens

GLOBAL OPTIONS

`--profile <name>`
Use this profile for one command

`--region <region>`
Override the region

`--output json|table|text|yaml`
Choose the output format

`--query '<jmespath>'`
Filter and reshape the result

GLOBAL OPTIONS (cont.)

`--dry-run`
Check permissions without acting (EC2)

`--no-cli-pager`
Print straight to the terminal

`--debug`
Show requests and config lookup

Config lookup order

Source	Example
1. Command flags	`--profile, --region`
2. Environment	`AWS_PROFILE, AWS_REGION`
3. Config files	`~/.aws/config, credentials`
4. Instance or task role	`metadata service`

S3 TRANSFERS

`aws s3 ls`
List buckets

`aws s3 ls s3://<bucket>/<prefix>/`
List objects under a prefix

`aws s3 cp <file> s3://<bucket>/<key>`
Upload a file

`aws s3 cp s3://<bucket>/<key> <file>`
Download an object

`aws s3 cp <dir> s3://<bucket>/ --recursive`
Upload a folder

`aws s3 sync <dir> s3://<bucket>/<prefix>`
Make the target match the source

`aws s3 mv <src> <dst>`
Move or rename objects

`aws s3 rm s3://<bucket>/<key>`
Delete one object

`aws s3 rm s3://<bucket>/<prefix>/ --recursive`
Delete everything under a prefix

`aws s3 mb s3://<bucket>`
Create a bucket

`aws s3 rb s3://<bucket> --force`
Empty and delete a bucket

`aws s3 ls s3://<bucket> --recursive --human-readable --summarize`
Count objects and size

`aws s3 sync <src> <dst> --delete`
Also remove files missing at the source

Run with --dry-run first.

S3 TRANSFERS (cont.)

`--exclude '<glob>' --include '<glob>'`
Filter transfers (later flags win)

`--dryrun`
Preview s3 cp, sync, mv or rm

`aws s3 presign s3://<bucket>/<key> --expires-in <sec>`
Make a temporary download URL

S3 API

`aws s3api list-objects-v2 --bucket <b> --prefix <p>`
List objects with full metadata

`aws s3api head-object --bucket <b> --key <k>`
Show size, type and metadata

`aws s3api get-bucket-policy --bucket <b>`
Read the bucket policy

`aws s3api list-object-versions --bucket <b>`
List versions and delete markers

EC2

`aws ec2 describe-instances`
List instances

`aws ec2 describe-instances --instance-ids <id>`
Show one instance

`aws ec2 describe-instances --filters Name=tag:Name,Values=<name>`
Filter by tag

`aws ec2 start-instances --instance-ids <id>`
Start an instance

`aws ec2 stop-instances --instance-ids <id>`
Stop an instance

`aws ec2 reboot-instances --instance-ids <id>`
Reboot an instance

`aws ec2 terminate-instances --instance-ids <id>`
Delete an instance for good

`aws ec2 run-instances --image-id <ami> --instance-type <type> --key-name <key>`
Launch an instance

`aws ec2 describe-security-groups`
List security groups

IAM

`aws iam list-users`
List users

IAM (cont.)

`aws iam list-roles`
List roles

`aws iam get-user`
Show the calling user

`aws sts assume-role --role-arn <arn> --role-session-name <n>`
Get temporary role credentials

OUTPUT AND QUERIES

`--query 'Reservations[].Instances[].InstanceId'`
Flatten to a list of values

`--query 'Buckets[?starts_with(Name, `<p>`)].Name'`
Filter list items

`--query 'Reservations[].Instances[].{ID:InstanceId, State:State.Name}'`
Build a custom object per item

`--query '[0].<field>'`
Pick the first item's field

`--output text`
Plain tab-separated values for scripts

`--output table`
Readable table for people

`aws <svc> <op> | jq '<filter>'`
Hand JSON to jq instead

--query is JMESPath applied client-side after the response arrives, so it saves output, not API calls. Use backticks for literals inside single quotes.

HELP AND DISCOVERY

`aws help`
List services

`aws <service> help`
List a service's operations

`aws <service> <operation> help`
Show options and examples

`aws --version`
Show CLI version

`aws logs tail <group> --follow`
Stream CloudWatch logs

`aws ecr get-login-password | docker login --username AWS --password-stdin <registry>`
Log Docker in to ECR