# Nomad Workload orchestrator

**Job** your spec file → evaluation → **Eval** scheduler run → places → **Allocation** group on a node → runs → **Task** container, process → logs, exec

## CONCEPTS

You submit a job; Nomad evaluates it and places allocations on client nodes. Status commands walk that chain.

job plan exits 0 for no change, 1 when allocations would change, 255 on error. Handy in CI.

Job, alloc and node IDs can be shortened to a unique prefix.

## CONNECT

**export NOMAD_ADDR=**<*url*>
Set the server address

**export NOMAD_TOKEN=**<*secret-id*>
Authenticate with an ACL token

**export NOMAD_NAMESPACE=**<*ns*>
Work in a namespace

**nomad server members**
List servers and leader

**nomad version**
Print the CLI version

## RUN JOBS

**nomad job init**
Write example.nomad.hcl

## RUN JOBS (cont.)

**nomad job validate** <*file*>
Check a job file offline

**nomad job plan** <*file*>
Dry run with a diff

**nomad job run** <*file*>
Submit or update a job

**nomad job run -detach** <*file*>
Submit without monitoring

**nomad job run -check-index** <*n*> <*file*>
Submit only if unchanged since plan

**nomad job run -var** <*key*>=<*value*> <*file*>
Set a job variable

## INSPECT JOBS

**nomad status**
List all jobs

**nomad job status** <*job*>
Summary, allocations, deployment

**nomad job inspect** <*job*>
Show the stored job as JSON

**nomad job history** <*job*>
List versions of a job

**nomad job allocs** <*job*>
List allocations of a job

## CHANGE AND STOP

**nomad job stop** <*job*>
Stop a job

**nomad job stop -purge** <*job*>
Stop and remove from the state

**nomad job scale** <*job*> <*group*> <*count*>
Set a task group count

**nomad job revert** <*job*> <*version*>
Roll back to an older version

**nomad job restart** <*job*>
Restart allocations in place

**nomad job promote** <*job*>
Promote canaries

## ALLOCATIONS

**nomad alloc status** <*alloc*>
Task states and events

**nomad alloc logs** <*alloc*> <*task*>
Print task stdout

**nomad alloc logs -f** <*alloc*> <*task*>
Follow logs

**nomad alloc logs -job** <*job*>
Logs of a random alloc of the job

## ALLOCATIONS (cont.)

**nomad alloc exec** <*alloc*> <*cmd*>
Run a command in a task

**nomad alloc fs** <*alloc*> [<*path*>]
Browse the allocation directory

**nomad alloc restart** <*alloc*>
Restart a task in place

## NODES

**nomad node status**
List client nodes

**nomad node status** <*node*>
Resources and allocations of a node

**nomad node drain -enable** <*node*>
Migrate allocations away

**nomad node drain -disable** <*node*>
Cancel a drain

**nomad node eligibility -disable** <*node*>
Stop new placements

## EVALS AND DEPLOYMENTS

**nomad eval status** <*eval*>
Why a placement failed

## VARIABLES AND NAMESPACES

**nomad var put** <*path*> <*key*>=<*value*>
Write a variable

**nomad var get** <*path*>
Read a variable

**nomad var list** [<*prefix*>]
List variables

## RECIPES

### Plan, then apply exactly that plan

```
nomad job plan web.nomad.hcl
# copy the Job Modify Index from the output
nomad job run -check-index <n> web.nomad.hcl
```

### Drain a node for maintenance

```
nomad node drain -enable -yes <node>
nomad node status <node>
# ...maintenance...
nomad node drain -disable <node>
nomad node eligibility -enable <node>
```

### Debug a failing task

```
nomad job status <job>
nomad alloc status <alloc>
nomad alloc logs -stderr <alloc> <task>
nomad alloc exec -task <task> <alloc> /bin/sh
```