Table of Contents

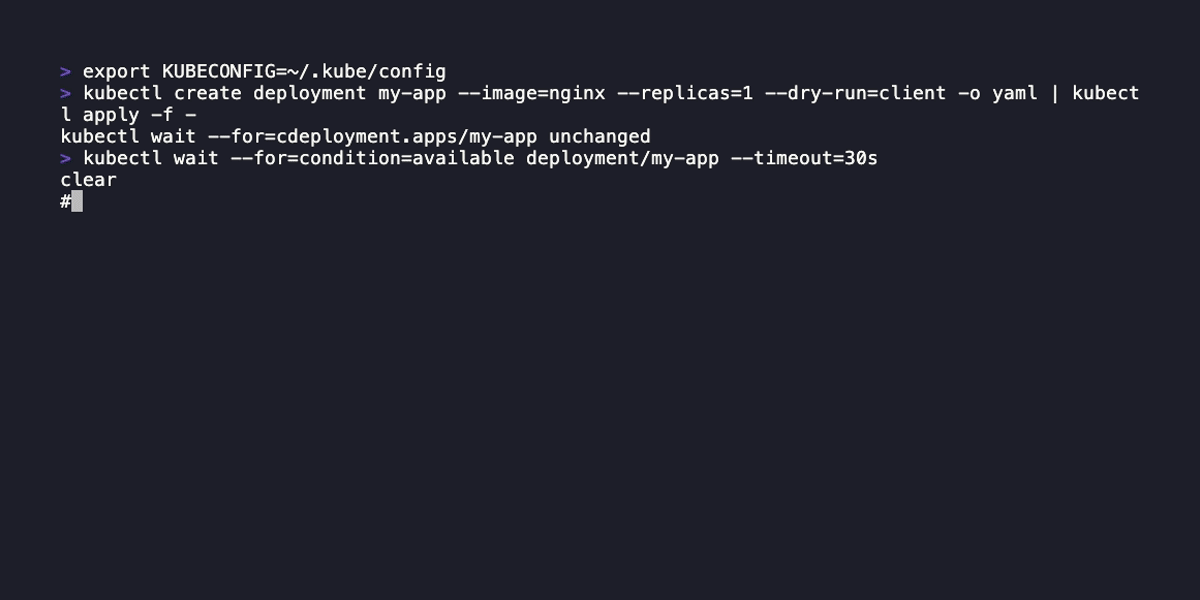
# ktl

ktl is a Kubernetes-focused CLI for logs, Helm workflows, and BuildKit builds.

Core commands:

* Fast pod logs: ktl logs
* Helm preview/apply/delete/revert: ktl apply plan, ktl apply, ktl delete, ktl revert
* Build images with BuildKit: ktl build
* Orchestrate many releases as a DAG: ktl stack
* Secure access to cluster services: ktl tunnel
* HTML viewers: ktl help --ui, ktl apply --ui, ktl delete --ui

## Showcase



Stack Apply Showcase

## Why ktl?

ktl is designed to be a single binary that bridges the gap between **interactive developer workflows** and **headless CI pipelines**. It is suitable for both daily development and rigorous CI/CD steps.

| Tool | Difference |
| --- | --- |
| **ArgoCD / Flux** | These are GitOps operators that run *inside* the cluster. ktl is a CLI that runs *outside* (on your laptop or in GitHub Actions) to render, validate, and apply changes. It complements GitOps by providing a way to “dry run” and debug charts locally before pushing. |
| **Helmfile** | ktl stack offers similar multi-release orchestration but adds a DAG-aware scheduler, concurrent execution, and a rich interactive TUI/HTML viewer for debugging complex dependencies. |
| **Tilt / Skaffold** | These are primarily “inner loop” dev tools that watch files and auto-deploy. ktl focuses on explicit, predictable operations that work exactly the same way in CI as they do on your machine, reducing “it works on my machine” issues. |

**Key Features:** - **Hybrid Runtime**: Works as a rich TUI for devs and a structured JSON/log emitter for CI. - **Unified Stack**: Bundles logging (ktl logs), building (ktl build), and deploying (ktl apply) in one cohesive toolchain. - **Observability**: Built-in HTML viewers for plans, deployments, and help docs.

## Install

Requires Go 1.25.7+.

go install ./cmd/ktl

Or via Makefile:

make build # writes ./bin/ktl  
make install # installs ./cmd/ktl to GOBIN/GOPATH/bin

Other binaries:

go install ./cmd/verify  
go install ./cmd/package

## Quickstart

# Initialize repo defaults  
ktl init  
  
# Tail logs  
ktl logs deploy/my-app -n default  
  
# Preview and deploy a Helm chart (with the viewer)  
ktl apply plan --chart ./chart --release my-app -n default  
ktl apply --chart ./chart --release my-app -n default --ui  
  
# Delete (with the viewer)  
ktl delete --release my-app -n default --ui  
  
# Build an image with BuildKit  
ktl build . -t ghcr.io/acme/app:dev  
  
# Open a tunnel to a service  
ktl tunnel service/my-app 8080:80  
  
# Searchable interactive help  
ktl help --ui

## Verification

ktl provides powerful verification tools for your Kubernetes resources.

### Stack Verification

Verify a stack’s deployment status and health:

ktl stack verify --config stack.yaml

### Configuration Verification

The standalone verify tool checks your manifests against policies and best practices.

go install ./cmd/verify  
  
# Verify a Helm chart  
verify --chart ./chart --release my-app -n default  
  
# Verify a manifest  
verify --manifest ./rendered.yaml

## Docs

* Recipes: docs/recipes.md
* Architecture: docs/architecture.md
* Troubleshooting: docs/troubleshooting.md
* Contributor guardrails: AGENTS.md

## Development

make preflight # fmt + lint + unit tests  
make test # go test ./...  
make fmt # gofmt  
make lint # go vet ./...

See AGENTS.md for contributor guidance.