Skaffold Kubernetes dev loop

CONCEPTS

skaffold dev is the inner loop: it builds, deploys and tails logs, rebuilds on file change, and deletes what it deployed when you press Ctrl+C. skaffold run deploys once and leaves it.

- **build** images from artifacts
- **test** optional
- **render** manifests with new tags
- **deploy** kubectl / helm / kustomize
- **verify** optional checks

Skaffold only drives your existing tools: Docker or buildpacks to build, kubectl, Helm or Kustomize to deploy. Its own config is skaffold.yaml.

DEV LOOP

`skaffold dev`
Build, deploy, watch, tail logs

`skaffold dev --port-forward=services`
Forward service ports locally

`skaffold dev --no-prune --cleanup=false`
Keep resources after exit

`skaffold dev --auto-build=false`
Rebuild only on request

`skaffold dev -w <image>`
Watch only matching images

`skaffold dev --skip-tests`
Skip tests after build

`skaffold dev --tail=false`
Do not stream logs

RUN, BUILD & DEPLOY

`skaffold run`
Build and deploy once

`skaffold run --tail`
Deploy and stream logs

`skaffold build`
Build images only

`skaffold build -q`
Print only built image names

RUN, BUILD & DEPLOY (cont.)

`skaffold build --push`
Push built images

`skaffold build --file-output <file>`
Save built tags as JSON

`skaffold deploy -a <file>`
Deploy pre-built images

`skaffold test`
Run tests on built images

DEBUG & VERIFY

`skaffold debug`
Dev loop with debug ports

`skaffold debug --port-forward=user,debug`
Forward debug ports

`skaffold verify`
Run verify tests

CONFIG & PROFILES

`skaffold init`
Generate skaffold.yaml

`skaffold init --generate-manifests`
Also create basic manifests

`skaffold init -k <glob>`
Pick Kubernetes manifests

`skaffold fix`
Print config in newest schema

`skaffold fix --overwrite`
Upgrade skaffold.yaml in place

`skaffold diagnose`
Inspect effective config

`skaffold diagnose --yaml-only`
Print merged skaffold.yaml

`skaffold dev -f <file>`
Use another config file

`skaffold dev -p <profile>`
Activate a profile

`skaffold dev -m <module>`
Select config module

RENDER & DELETE

`skaffold render`
Print rendered manifests

RENDER & DELETE (cont.)

`skaffold render -o <file>`
Write manifests to a file

`skaffold render --offline`
Render without a cluster

`skaffold delete`
Remove deployed resources

`skaffold delete --dry-run`
List what would be deleted

TARGETING & IMAGES

`skaffold dev -n <ns>`
Deploy to a namespace

`skaffold dev --kube-context <ctx>`
Use a kube context

`skaffold dev -d <repo>`
Default image repository

`skaffold dev -t <tag>`
Custom image tag

`skaffold dev --platform <os/arch>`
Target platform

`skaffold dev --cache-artifacts=false`
Disable build cache

HOUSEKEEPING

`skaffold version`
Print version

`skaffold completion <shell>`
Shell completion script

`skaffold dev -v <level>`
Set log level

RECIPES

Inner loop on local cluster

```
kubectl config use-context kind-dev
skaffold dev --port-forward=services -n dev
```

Build once, deploy per environment

```
skaffold build -d registry.example/app --push --file-output build.json
skaffold deploy -a build.json -p staging -n staging
skaffold deploy -a build.json -p prod -n prod
```