Helmfile Declarative Helm releases

CONCEPTS

helmfile.yaml declares which Helm releases belong on a cluster; Helmfile makes the cluster match it.

helmfile.yaml desired state
↓ template + values
Helm releases charts + values
↓ diff / apply / sync
Cluster kube-context

apply runs diff first and syncs only when there are changes; sync always upgrades every selected release.

PROJECT

`helmfile init`
Check helm and install plugins

`helmfile repos`
Add the repositories from the file

`helmfile deps`
Update chart deps and lock files

`helmfile list`
List releases in the state file

`helmfile lint`
Run helm lint on the charts

DEPLOY

`helmfile diff`
Show pending changes

`helmfile apply`
Diff, then sync only if changed

`helmfile apply -i`
Confirm before changing the cluster

`helmfile sync`
Upgrade --install every release

`helmfile sync --wait --timeout <sec>`
Wait until resources are ready

`helmfile template`
Render manifests locally

`helmfile destroy`
Uninstall and purge releases

`helmfile status`
Run helm status per release

DEPLOY (cont.)

`helmfile apply --skip-deps`
Skip repo update and dep build

TARGETING

`helmfile -f <file or dir> <cmd>`
Use another helmfile or directory

`helmfile -e <env> <cmd>`
Choose environment (default: default)

`helmfile -l name=<release> <cmd>`
Select one release

`helmfile -l <key>=<val>,<key2>=<val2> <cmd>`
Labels must all match (AND)

`helmfile -l <k>=<a> -l <k>=<b> <cmd>`
Repeated selectors match any (OR)

`helmfile --kube-context <ctx> <cmd>`
Use a kubectl context

HELMFILE.YAML

Minimal file

```
repositories:
  - name: bitnami
    url: https://charts.bitnami.com/
bitnami
---
releases:
  - name: cache
    namespace: data
    chart: bitnami/redis
    version: <x.y.z>
    values:
      - values/redis.yaml
```

Environments

```
environments:
  default:
    values: [env/default.yaml]
  production:
    values: [env/production.yaml]
---
# in releases:
  installed: {{ eq .Environment.Name
"production" | toYaml }}
```

Dependencies and toggles

```
releases:
  - name: app
    chart: ./charts/app
    needs: [data/cache]
    condition: app.enabled
    labels: {tier: web}
```