stern Multi-pod Kubernetes logs

START HERE

`stern <query>`
Tail pods matching regex

`stern <query> -c <regex>`
Pick containers

`stern . -n <ns>`
All pods in a namespace

`stern <query> --tail 20`
Start with last lines

`stern deployment/<name>`
Tail a deployment's pods

`stern <query> -s 15m`
Only recent logs

`stern -l <k>=<v> .`
Pods by label

`stern <query> -i <regex>`
Only matching lines

Pod query regex or kind/name → **`-n -l --node`** → **Pods** namespaces, labels → **`-c -E --container-state`** → **Containers** init, ephemeral → **`-i -e --since`** → **Log lines** merged stream → **`-o --template`** → **Output** colour, JSON, custom

CONCEPTS

stern tails many pods and containers at once, one colour per pod, and picks up new or restarted pods automatically. The query is a regular expression on the pod name, not a glob.

Default is all pods you can see in the current namespace and context; --since defaults to 48h and --tail to all lines, so busy pods print a lot at start.

SELECTING PODS

`stern <query>`
Tail pods whose name matches regex

`stern .`
Tail every pod in the namespace

`stern <kind>/<name>`
Exact resource, e.g. deployment/nginx
Kinds: pod, replicationcontroller, service, daemonset, deployment, replicaset, statefulset, job.

`-n <ns>` / `--namespace <ns>`
Namespace, repeat or comma-separate

`-A` / `--all-namespaces`
Every namespace

`-l <selector>` / `--selector`
Filter by label selector

`--field-selector <expr>`
Filter by field, e.g. spec.nodeName=<node>

`--node <name>`
Pods on one node

`--exclude-pod <regex>`
Skip pods matching regex

`--condition <name[=value]>`
Pods by condition, e.g. ready=false
Only works together with --tail=0 or --no-follow.

SELECTING PODS (cont.)

`-p` / `--prompt`
Pick app.kubernetes.io/instance interactively

CONTAINERS

`-c <regex>` / `--container`
Only containers matching regex

`-E <regex>` / `--exclude-container`
Skip containers, e.g. istio-proxy

`--container-state <state>`
running, waiting, terminated or all

`--init-containers=false`
Leave out init containers

`--ephemeral-containers=false`
Leave out ephemeral containers

TIME AND VOLUME

`--tail <n>`
Last n lines per container
-1 means everything; --tail 0 shows only new lines.

`-s <dur>` / `--since <dur>`
Logs newer than e.g. 15m or 2h

`--no-follow`
Print and exit, do not stream

`-t` / `--timestamps`
Prefix timestamps
Optional value needs an equals sign: -t=short.

`--timestamps=short`
Shorter timestamp format

`--timezone <tz>`
Show timestamps in e.g. Asia/Tokyo

`--max-log-requests <n>`
Limit concurrent log streams

FILTERING LINES

`-i <regex>` / `--include`
Show only matching lines

`-e <regex>` / `--exclude`
Hide matching lines

`-H <regex>` / `--highlight`
Highlight matches, keep all lines

`--only-log-lines`
Drop the pod/container prefix noise

OUTPUT AND COLOUR

`-o raw`
Message text only, no prefix

`-o json`
One JSON object per line

`-o extjson` / `ppextjson`
Extended or pretty-printed JSON

`--template '<go-template>'`
Custom line format
Fields: Message, Timestamp, NodeName, Namespace, PodName, ContainerName, Labels, Annotations.

`-T <file>` / `--template-file`
Template read from a file

`--color auto|always|never`
Force or disable colour
Use always when piping into less -R.

`-d` / `--diff-container`
Separate colour per container

`--pod-colors <sgr,...>`
Custom pod colours

`--container-colors <sgr,...>`
Custom container colours

CLUSTER AND CONFIG

`--context <name>`
Use another kubeconfig context

CLUSTER AND CONFIG (cont.)

`--kubeconfig <path>`
Use another kubeconfig file

`--config <path>`
Config file, default ~/.config/stern/config.yaml
Also settable with the STERNCONFIG environment variable.

`--stdin`
Read logs from stdin, colour only

`--qps <n>` / `--burst <n>`
Throttle API requests (-1 disables)

`--completion bash|zsh|fish`
Print shell completion code

`-v` / `--version`
Print version

`kubectl krew install stern`
Install as kubectl plugin

Put defaults such as tail: 10 or timestamps: short into the config file as flag-name: value pairs.

RECIPES

Errors across a deployment, recent

```
stern deployment/api -n prod --since 15m -i 'error|panic' -e healthcheck
```

Pretty JSON logs through jq

```
stern api -n prod -o raw --tail 0 | jq -R 'fromjson? | select(.level=="error")'
```

Debug a rollout, new lines only

```
kubectl rollout restart deploy/api -n prod
stern api -n prod --tail 0 -t=short -E istio-proxy
```

Custom line with namespace and pod

```
stern . -A -l app=web --template '{{.Namespace}}/{{.PodName}} {{.Message}}{{"\n"}}'
```