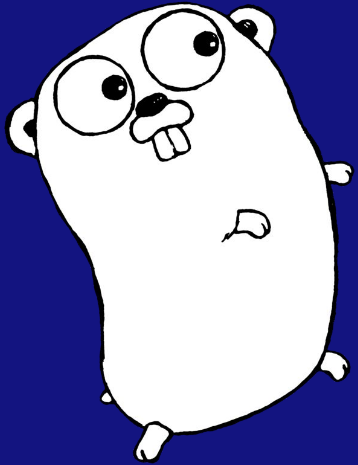
Own CLI tooling in Go



Ondřej Šika
Freelance & SikaLabs s.r.o.

ondrej@sika.io
@ondrejsika

Gophercamp 2025
Brno, 25. 4. 2025

Ondřej Šika

DevOps {Engineer, Consultant, Trainer}

Founder and CEO
of the DevOps company **SikaLabs**.

I build and run infrastructures on Kubernetes, open-source and cloud native tools.

And I do DevOps trainings about it.

@ondrejsika ondrej@sika.io sika.io /in/ondrejsika

Own CLI Tool. Why?

Simplify repetitive tasks

We are lazy so we want to make our work easier

Our example use cases

- Working with Kubernetes
- CLI shortcuts and aliases
- Code generation during development
- Generating data for testing
- Automatic commit messages
- ... and more

Solution?

Shell Scripts 🎉

@ondrejsika ondrej@sika.io sika.io /in/ondrejsika

Shell Scripts 😔

Shell Scripts

- Write once, run forever
- Shell works everywhere
- Easy to write
- All scripts in one Git repo

But ...

But ...

- Is Shell, Bash or ZSH really available everywhere?
- Are they truly easy to write? if it's more than chain of few command
- What about distribution without Git?
- Or without internet?

Another solution?

Python 🎉

Python 😔

Python solves one problem but makes others worse

@ondrejsika ondrej@sika.io sika.io /in/ondrejsika

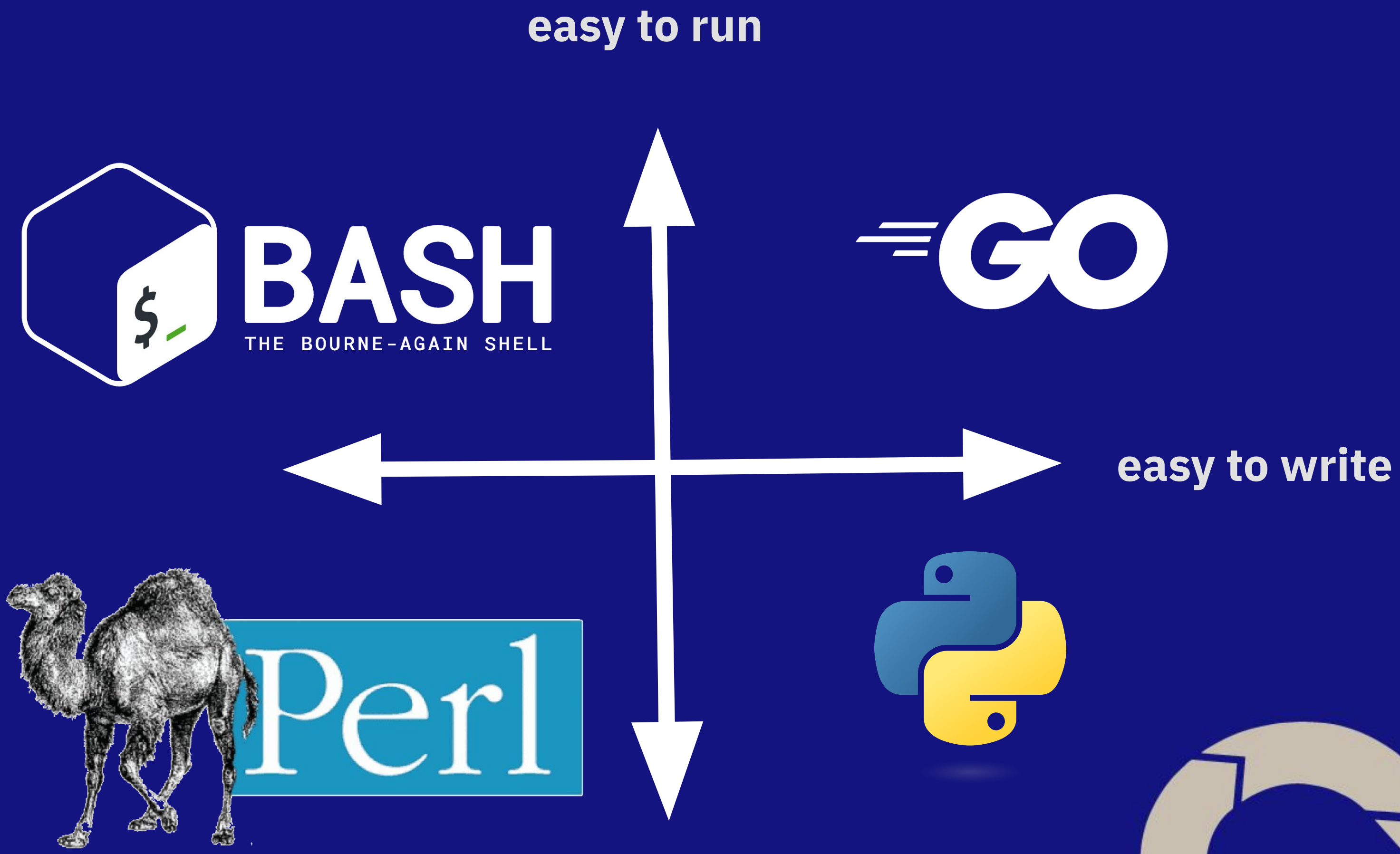
Pros

- Easier and faster development than SH/BASH
- Use of libraries, working with APIs, SDKs...

Cons

- Python isn’t available everywhere
- Different versions of Python, 2 vs 3
- Dependency management isn't ideal (e.g., virtualenv...)
- Still not a single binary for distribution

... solution?

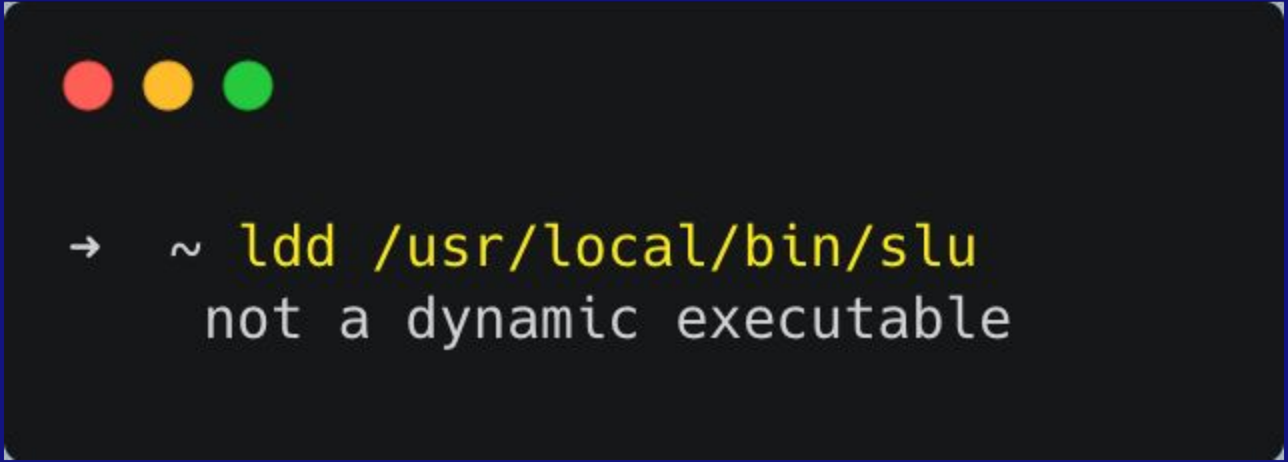

easy to run
BASH
THE BOURNE-AGAIN SHELL
GO
easy to write
Perl

Let's use Go 🎉🎉🎉

Why CLI Tool in Go?

- Statically linked binaries
- Easy development in Go (even for beginners)
- Simple distribution (brew, curl | sh, winget, ...)
- Tooling
    - Goreleaser
    - Cobra
    - Viper

Statically Linked Binaries



```
➜  ~ ldd /usr/local/bin/slu
    not a dynamic executable
```

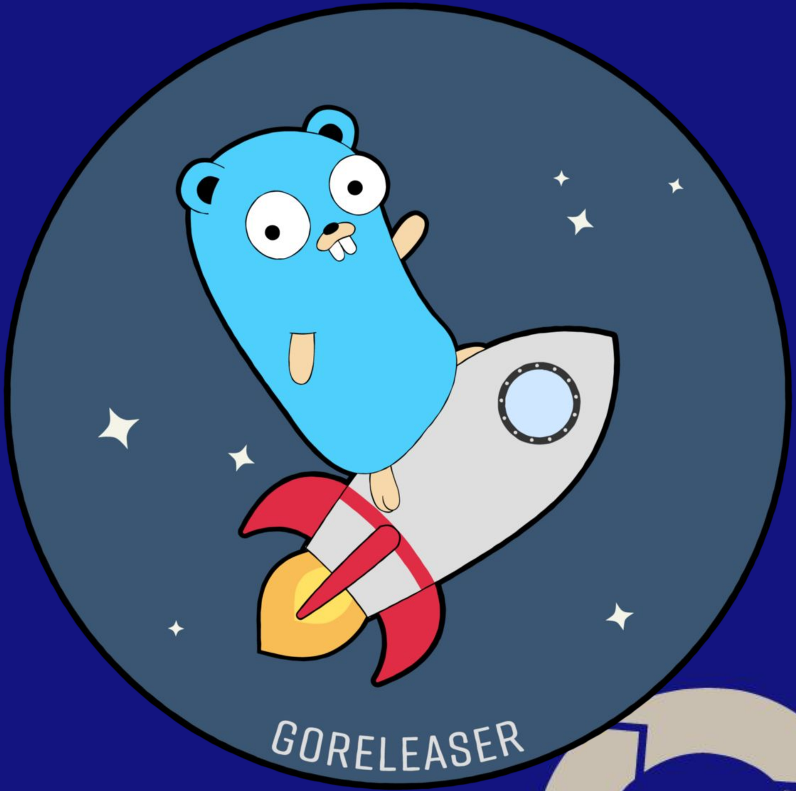
Statically Linked Binaries

- Self contained executable, no external dependencies
- Very easy to build statically linked binaries

```
CGO_ENABLED=0 go build
```

- Binary for every platform and architecture:
  - Linux, Mac, Windows, ...
  - amd64, arm64, ...

Goreleaser



Goreleaser

- Build binaries (for all platforms)
- Build & push Docker image
- Upload binaries to GitHub/Gitlab
- Create Brew and Scoop packages

Go Binary Distribution

Thanks to Goreleaser, distribution is very easy:

- brew install my-org/tap/my-cli-tool
- curl -fsSL https://.../my-cli-tool/master/install.sh | sh
- winget install -e --id my-org.my-cli-tool
- scoop install https://.../scoop-bucket/master/my-cli-tool.json

Cobra & Viper

@ondrejsika   ondrej@sika.io   sika.io   /in/ondrejsika

Cobra

Great library for writing CLI tools

- Used by most CLI tools: Docker, Kubernetes, Helm, ArgoCD, ...
- Very easy and pleasant to use
- Automatically generates completions

  BASH, ZSH, PowerShell, Fish
- Generates documentation

Viper

Library for config files and ENV variables

- setting defaults
- reading from JSON, TOML, YAML, ...
- reading from environment variables
- reading from remote config systems (etcd or Consul)

Completion

Supports: BASH, ZSH, FISH a PowerShell

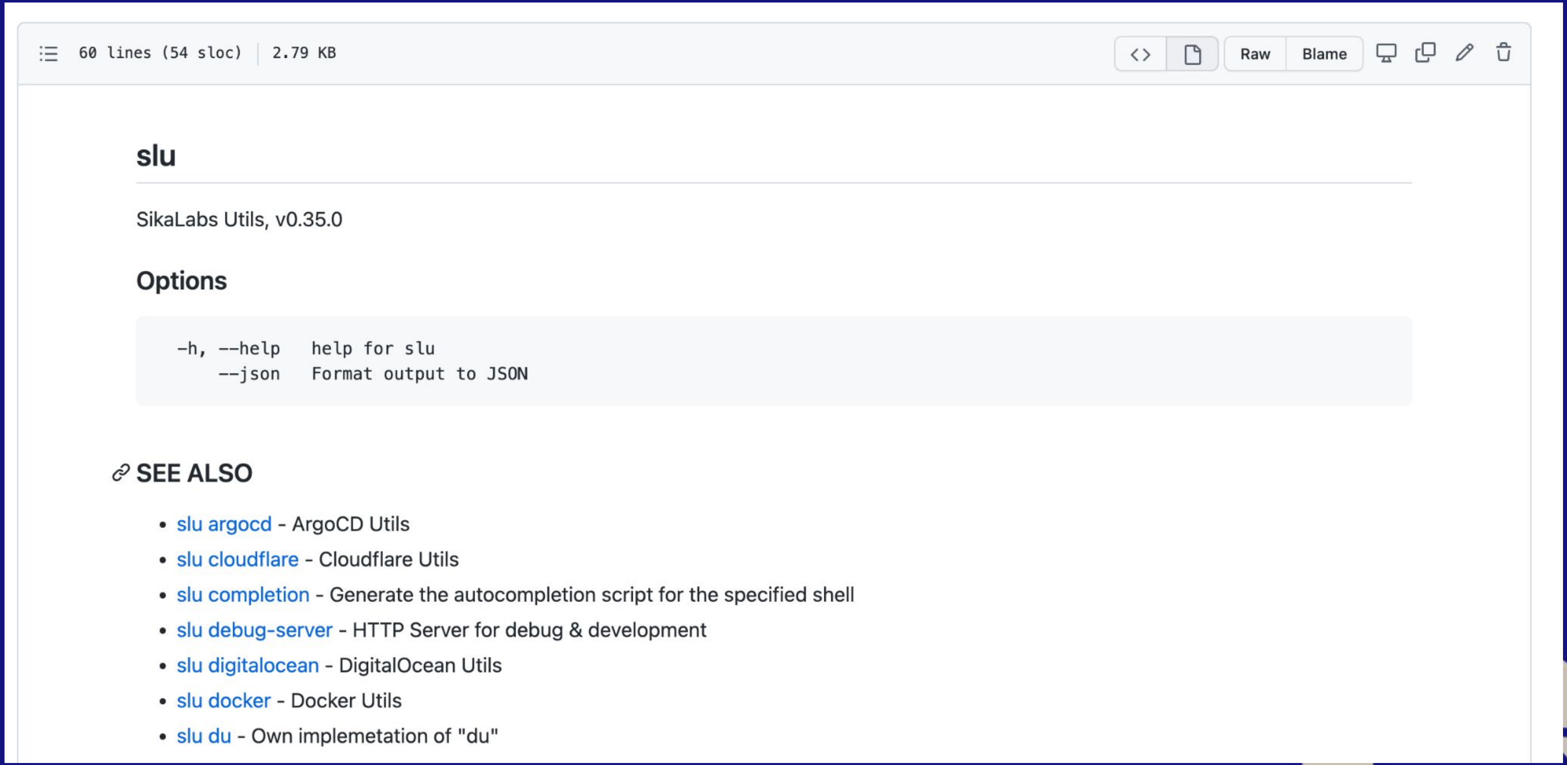
Source in current shell

```
. <(my-cli-tool completion bash)
```

Add to .bashrc

```
echo ". <(my-cli-tool completion bash)" >> ~/.bashrc
```

Generated Documentation



slu

SikaLabs Utils, v0.35.0

Options

```
  -h, --help   help for slu
      --json   Format output to JSON
```

SEE ALSO

- slu argocd - ArgoCD Utils
- slu cloudflare - Cloudflare Utils
- slu completion - Generate the autocompletion script for the specified shell
- slu debug-server - HTTP Server for debug & development
- slu digitalocean - DigitalOcean Utils
- slu docker - Docker Utils
- slu du - Own implemetation of "du"

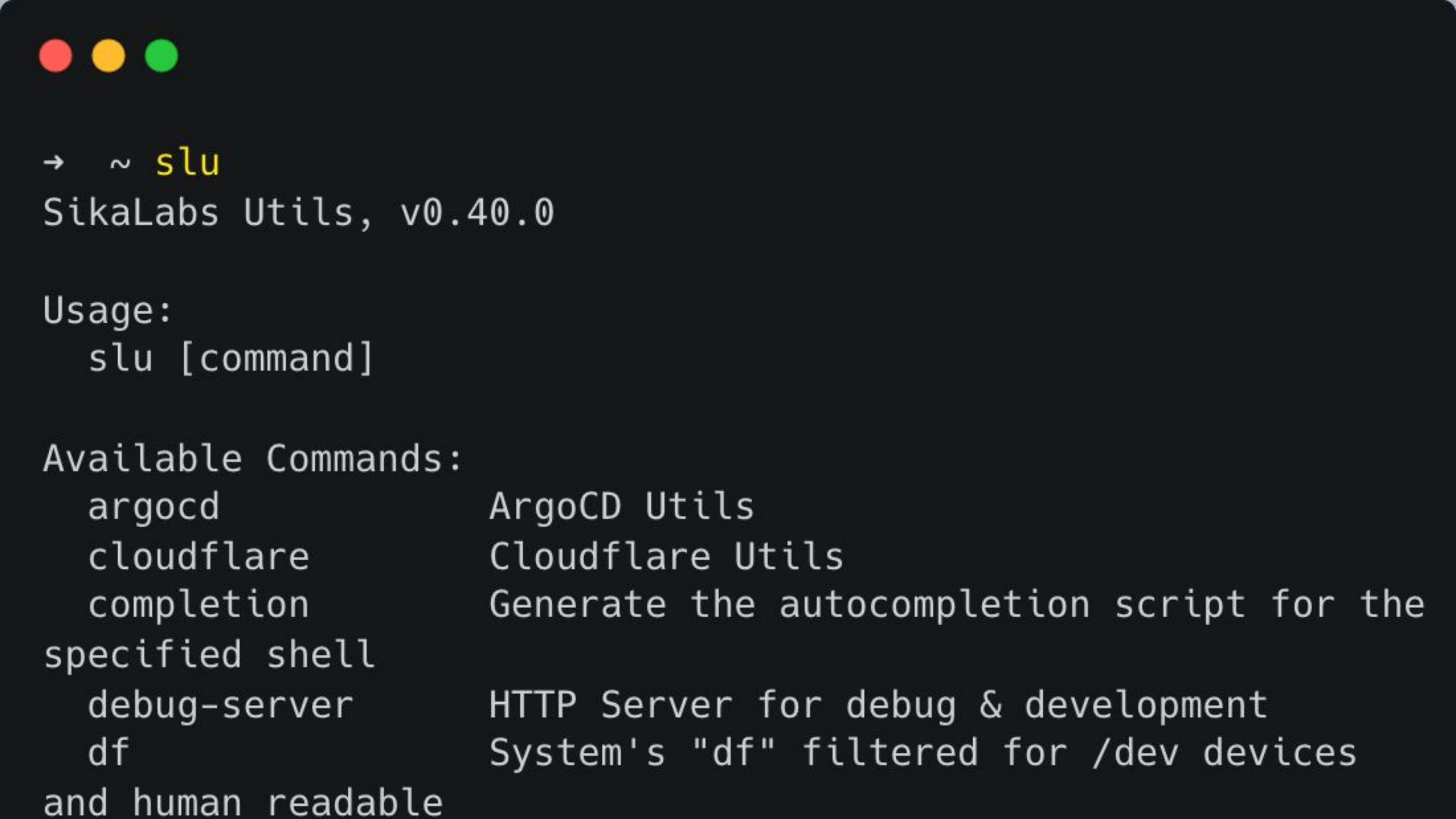
Our CLI Tools in Go

@ondrejsika ondrej@sika.io sika.io /in/ondrejsika

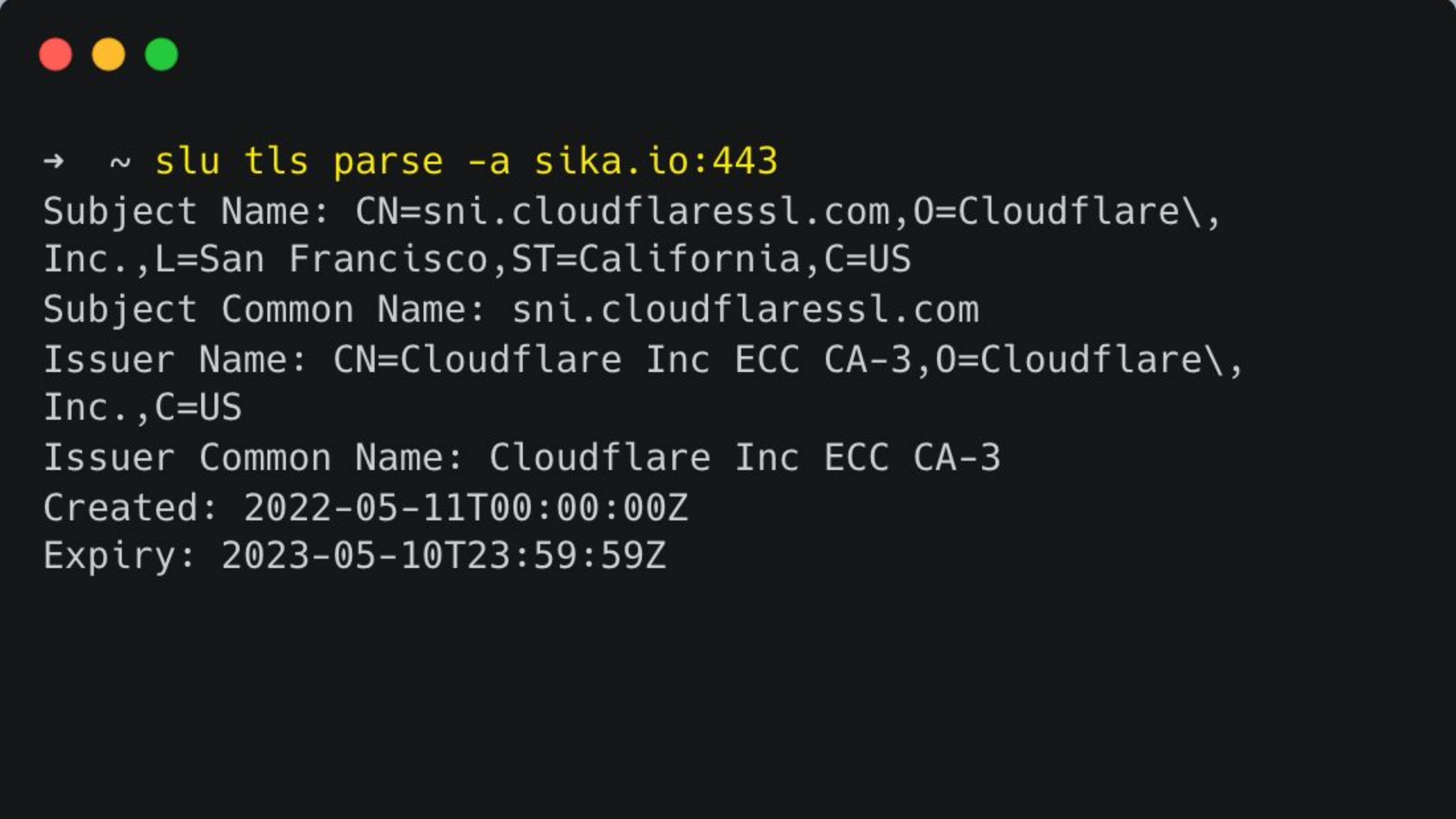
My CLI Tools in Go

- slu - various utilities that might be useful (SikaLabs Utils)
- training-cli - tool to simplify training preparation
- tergum - Backup tool
- gobble - alternative to Ansible

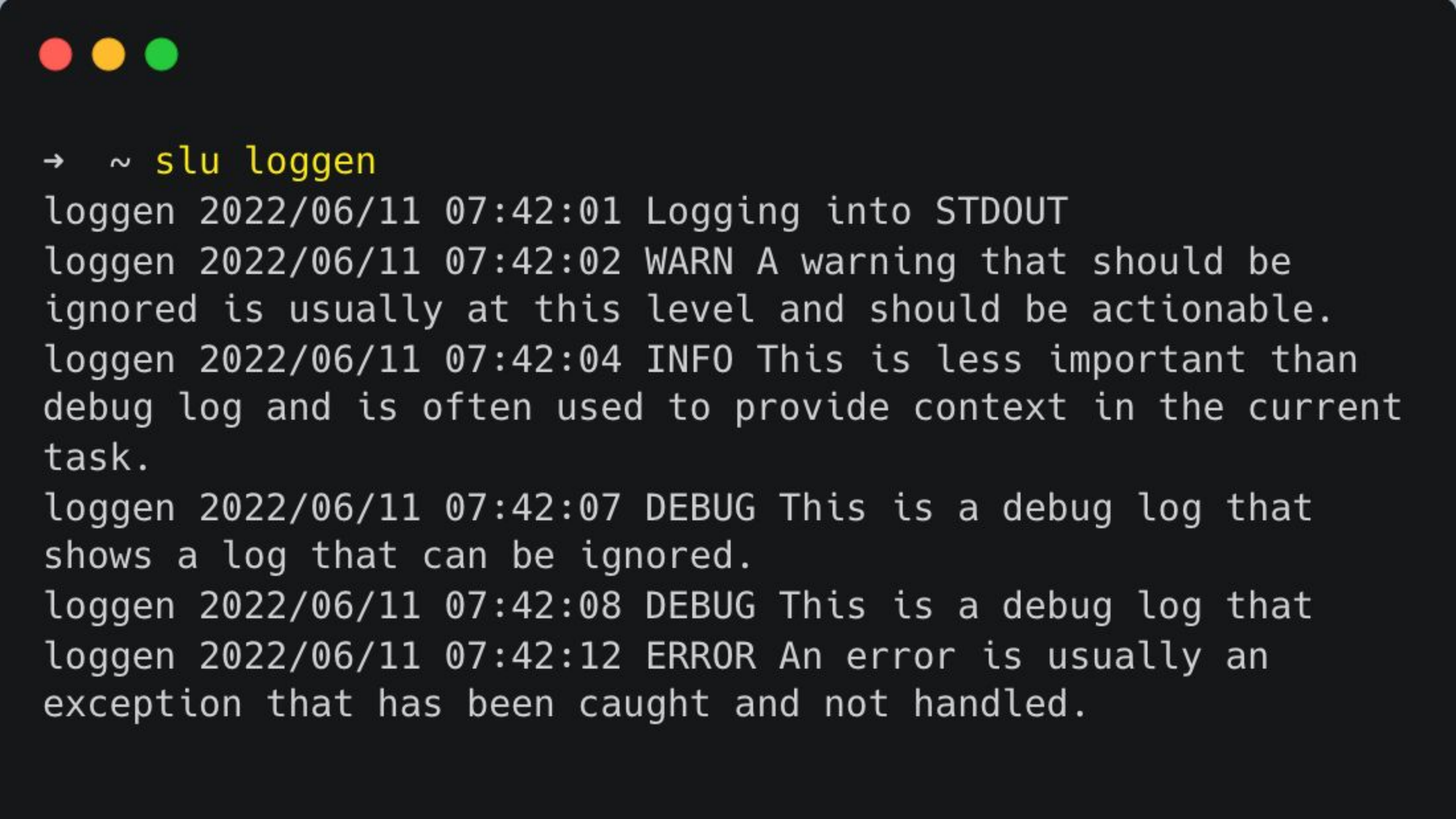
github.com/sikalabs/slu



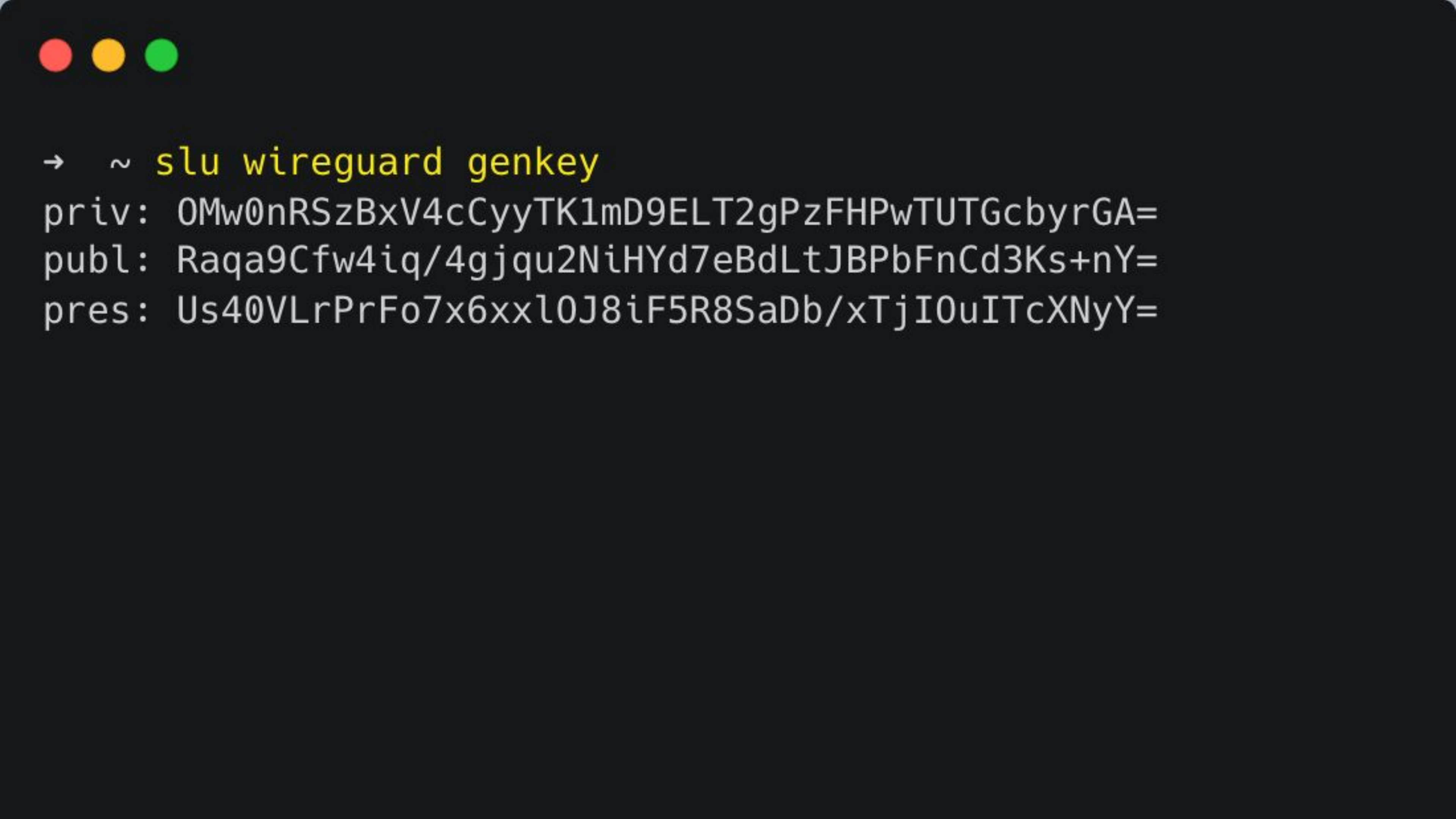
```
➜  ~ slu
SikaLabs Utils, v0.40.0

Usage:
  slu [command]

Available Commands:
  argocd          ArgoCD Utils
  cloudflare      Cloudflare Utils
  completion      Generate the autocompletion script for the
specified shell
  debug-server    HTTP Server for debug & development
  df              System's "df" filtered for /dev devices
and human readable
```



```
→  ~ slu tls parse -a sika.io:443
Subject Name: CN=sni.cloudflaressl.com,O=Cloudflare\,
Inc.,L=San Francisco,ST=California,C=US
Subject Common Name: sni.cloudflaressl.com
Issuer Name: CN=Cloudflare Inc ECC CA-3,O=Cloudflare\,
Inc.,C=US
Issuer Common Name: Cloudflare Inc ECC CA-3
Created: 2022-05-11T00:00:00Z
Expiry: 2023-05-10T23:59:59Z
```

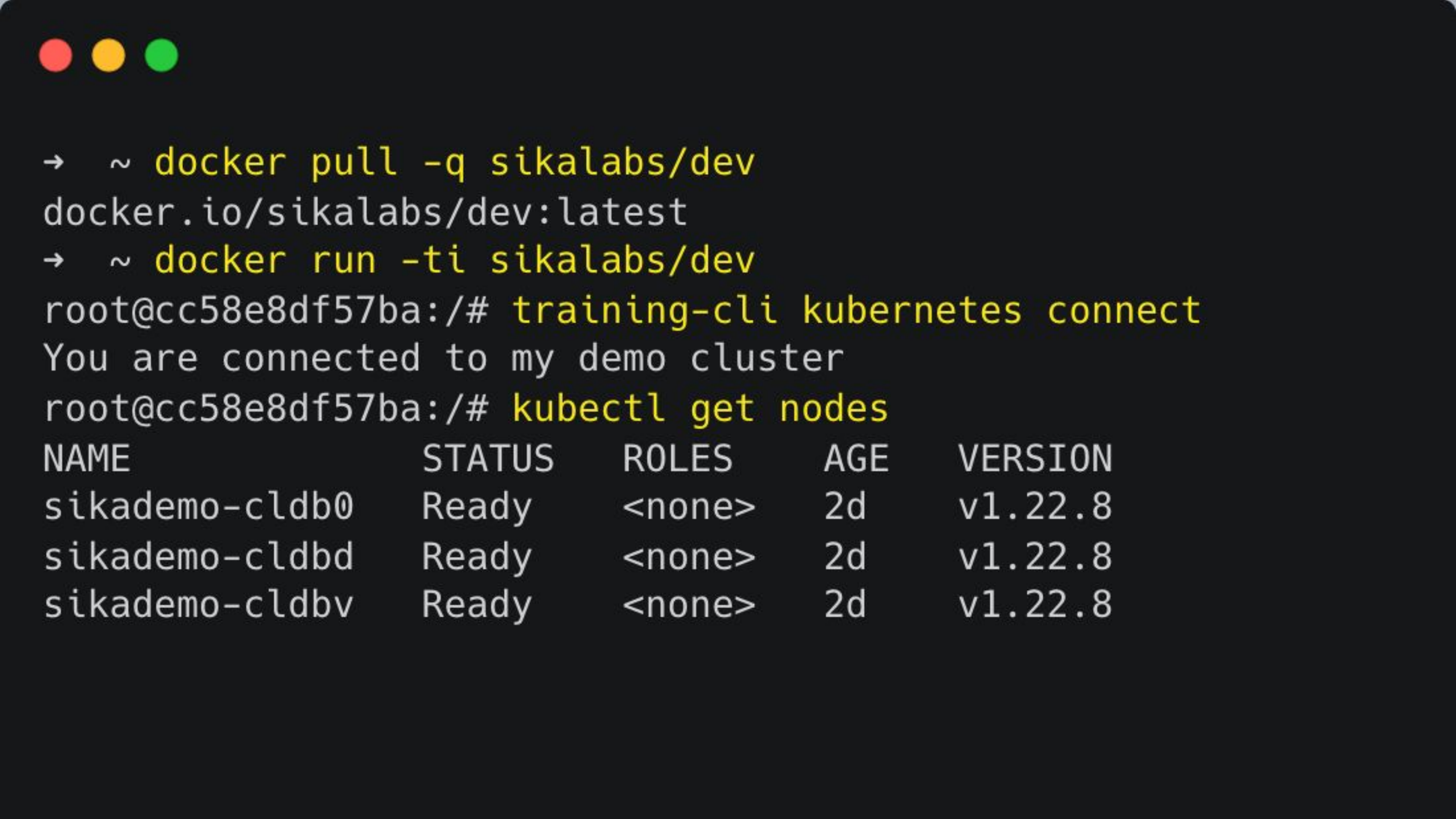


```
➜  ~ slu loggen
loggen 2022/06/11 07:42:01 Logging into STDOUT
loggen 2022/06/11 07:42:02 WARN A warning that should be
ignored is usually at this level and should be actionable.
loggen 2022/06/11 07:42:04 INFO This is less important than
debug log and is often used to provide context in the current
task.
loggen 2022/06/11 07:42:07 DEBUG This is a debug log that
shows a log that can be ignored.
loggen 2022/06/11 07:42:08 DEBUG This is a debug log that
loggen 2022/06/11 07:42:12 ERROR An error is usually an
exception that has been caught and not handled.
```



```
➜  ~ slu wireguard genkey
priv: OMw0nRSzBxV4cCyyTK1mD9ELT2gPzFHPwTUTGcbyrGA=
publ: Raqa9Cfw4iq/4gjqu2NiHYd7eBdLtJBPbFnCd3Ks+nY=
pres: Us40VLrPrFo7x6xxlOJ8iF5R8SaDb/xTjIOuITcXNyY=
```

github.com/ondrejsika/training-cli



```
➜  ~ docker pull -q sikalabs/dev
docker.io/sikalabs/dev:latest
➜  ~ docker run -ti sikalabs/dev
root@cc58e8df57ba:/# training-cli kubernetes connect
You are connected to my demo cluster
root@cc58e8df57ba:/# kubectl get nodes
NAME                STATUS   ROLES    AGE   VERSION
sikademo-cldb0      Ready    <none>   2d    v1.22.8
sikademo-cldbd      Ready    <none>   2d    v1.22.8
sikademo-cldbv      Ready    <none>   2d    v1.22.8
```

LIVE DEMO TIME

Summary

- CLI tools in Go are great – proven in practice
- Easy to use and distribute
- Simple and fast to develop

Thank you for your attention.

@ondrejsika   ondrej@sika.io   sika.io   /in/ondrejsika

Email

ondrej@sika.io

Twitter

@ondrejsika

LinkedIn

/in/ondrejsika

Slides

sika.link/slides