As you **might know**, *CloudControl* is DoDevOps' cloud engineer toolbox, which supplies cloud engineers working in different clouds with common tools in an easy-to-deploy container environment suitable for multiple projects.

I always wasn't quite happy with CloudControl's initialization process. It was very cluttered, didn't point out the relevant information and… well, ugly.

At first, I came up with an idea to use **Dialog** scripts to display some appealing user interface during the process, but that didn't work out mostly because of the Azure flavour's sign up process.

Then I thought, why not give a web client a go? I'm already used to **Vue**, so it wouldn't be that hard. One thing was a problem, though: I'd need a web/api-server that hooks up into the *running* initialization process. I didn't want to move away from shell scripts for the initialization of the flavour and features, because 1) they work quite well and 2) they're not really an effort to write for other people who want to add features.

Well. Two process running alongside tapping into each other *plus* a need for a single binary with a low footprint? That really cried out for **Go**.

As I've commented in another post, I tried starting with Go several times and never succeeded. But those requirements really were a perfect fit. So I tried again.

I downloaded the most current version of Go, checked out the tutorials again and got going.

Then, I needed the first external dependency and I remembered, that Go really was a PITA with this the last time I checked.

But, no. Nevermore. The Go folks finally seemed to agree on a good way to document and implement external dependencies with a go.mod file and dumped the nasty GOPATH. And everything was nicely supported by my IDE-of-choice **IntelliJ Idea**. Just add the required import, let Idea sync the dependencies and you're good to go. Yay!

The rest was classic trial-and-error until I came up with a nice looking Vue frontend (thanks to **vuetify**) and a stable API server (using **gin**) that delivers information from then goroutine, which handles the initialization.

With **Docker multi-stage-builds** it was also possible to build the vue client, the go server and the CloudControl flavour without any dependencies on the client. Oh, and I switched to **GitHub Actions** (because Docker automated builds didn't support build scripts)

So, that was my journey and I'm pretty happy with the end result. If you haven't already, please **check it out** and let me know what you think.

*Originally published to **dev.to***