# Many Rojects LightnungTalks

I develop software ...

... for customers

... for internal cosee projects

... for open source projects

... for hobby projects

... using ~25 dev environments on my laptop 😳

# context switches in the early days

"My Java / Terraform version is not compatible with the project."

"I don't remember the exact command. Let me check my shell history"

"Wait, I need to open a new shell for this."

"Can you please send me the latest env vars / env file? "Damn, I just applied the Terraform config on the wrong AWS account."

### Let's automate context switches

Manage multiple tool versions

https://asdf-vm.com/

Set environmental variables and secrets

https://direnv.net/

• Save project-specific shell one-liners

gnu.org/software/make/

One solution for all common toolchains 🤝

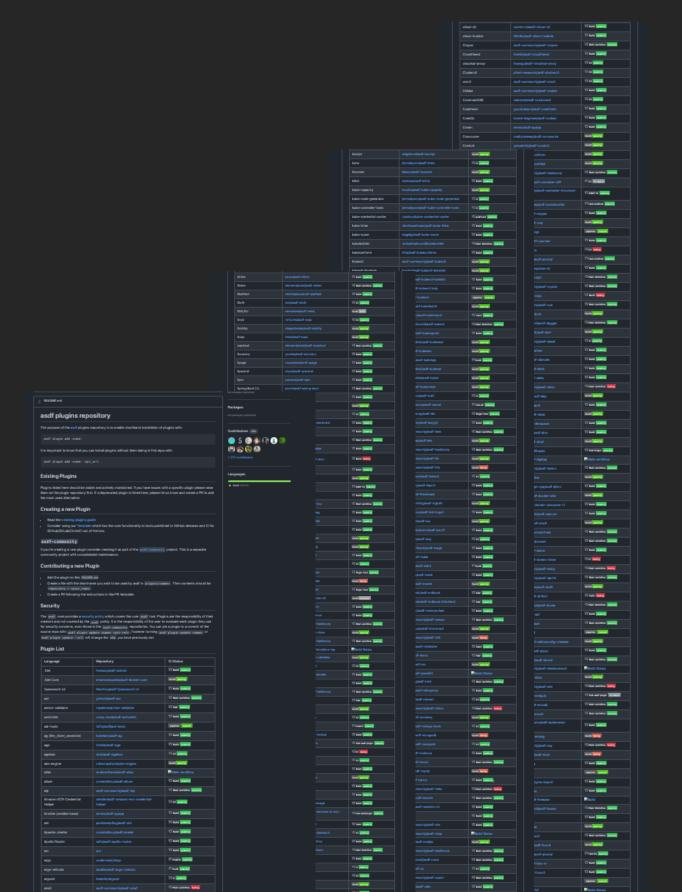
# ASDF Version Manager – Quickstart

```
# Install & hook into your shell & setup completions
<u>asdf-vm.com/guide/getting-started.html#getting-started</u>
# install asdf python plugin
> asdf plugin-add python https://github.com/danhper/asdf-python.git
# download version
> asdf install python 3.11.0
# set version in current directory
> asdf local python 3.11.0
# set default everywhere else
> asdf global python system
```

```
> python --version
Python 3.10.8
> cd asdf-demo
> python --version
Python 3.11.0
> cat .tool-versions
python 3.11.0
```

# ASDF Version Manager

- Adds a shim in front of the executable to redirect all calls to correct version
- Alternative to pyenv, rbenv, goenv, nvm, tfenv, sdkman https://github.com/asdf-vm/asdf-plugins#plugin-list
- ASDF is not a package manager
  - You manage upstream dependencies
  - No shell completions for installed tools



### Direnv – Quickstart

# Install & hook into your shell

```
https://github.com/direnv/direnv#basic-installation
# add env var to .envrc
> echo "export AWS_PROFILE=bwt" > .envrc
direnv: error /path/to/.envrc is blocked.
Run `direnv allow` to approve its content
# grant permission to load .envrc
> direnv allow
direnv: loading /your/path/to/.envrc
direnv: export +AWS_PROFILE
> cd ___
direnv: un
> echo $AWS_PROFILE
```

```
> echo $AWS_PROFILE

> cd direnv-demo
direnv: loading ~/customer/techtalks
/one-shell/tt/vhs/direnv-demo/.envrc
direnv: export +AWS_PROFILE
> echo $AWS_PROFILE
bwt
> cd ...
direnv: unloading
> echo $AWS_PROFILE
> l
```

### Direnv – stdlib

envrc is just a shell script and offers a stdlib with many functions.

```
# load .envrc in a parent directory
source_up
# add directory to PATH
PATH_add .
# load .env file
dotenv local.env
# display currently set versions
asdf current
```

# Makefile > shell script

- Available on any Unix system
- Multiple targets in one Makefile
  - with dependencies
  - with autocompletion
- Less boilerplate for one-liners

```
# use value of env var if set
 STAGE?=default
 check:
     @echo "all checks successful."
 # run target 'check' as dependency
 build: check
     @echo "build done on '$(STAGE)'."
> STAGE=local make build
all checks successful.
build done on 'local'.
```

### Conclusions

- Avoid global configuration (default AWS Profile, ~/.kube/config)
- Instead use environmental variables for local configuration
- Don't add .envrc scripts to git (but add .tool-versions + Makefiles)
- Use ASDF VM if you need 2 or more different versions

Make context switches as easy as changing a directory

cd ../australia
direnv: loading ~/customer/techtalks
/one-shell/tt/vhs/australia/.envrc
direnv: export +KANGOROOS
cd ../america
direnv: loading ~/customer/techtalks

/one-shell/tt/vhs/america/.envrc direnv: export +BAGELS +DONUTS

