

Multiple Terraform projects in a mono-repo. How to survive a mess?

Dmytro Prekrasnyi



About me



- all into DevOps and coding
- trying to automate everything
- fun of clean code, OSS and IaasC

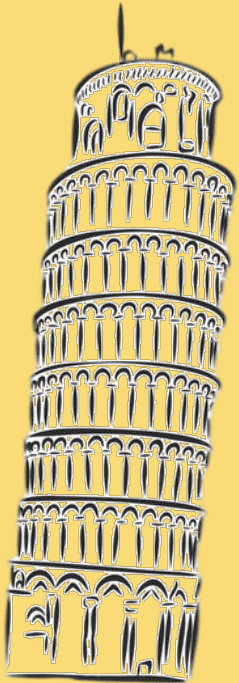
bit.ly/demmonico



Onboarding buddy: all is under control 🤖

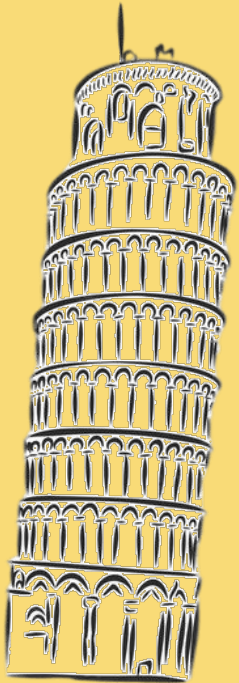


Agenda



- ✓ Prerequisites
- ✓ Problem Deep Dive
- ✓ Exploring existing solutions
- ✓ Introducing Terra Compose
- ✓ Demo
- ✓ Final words

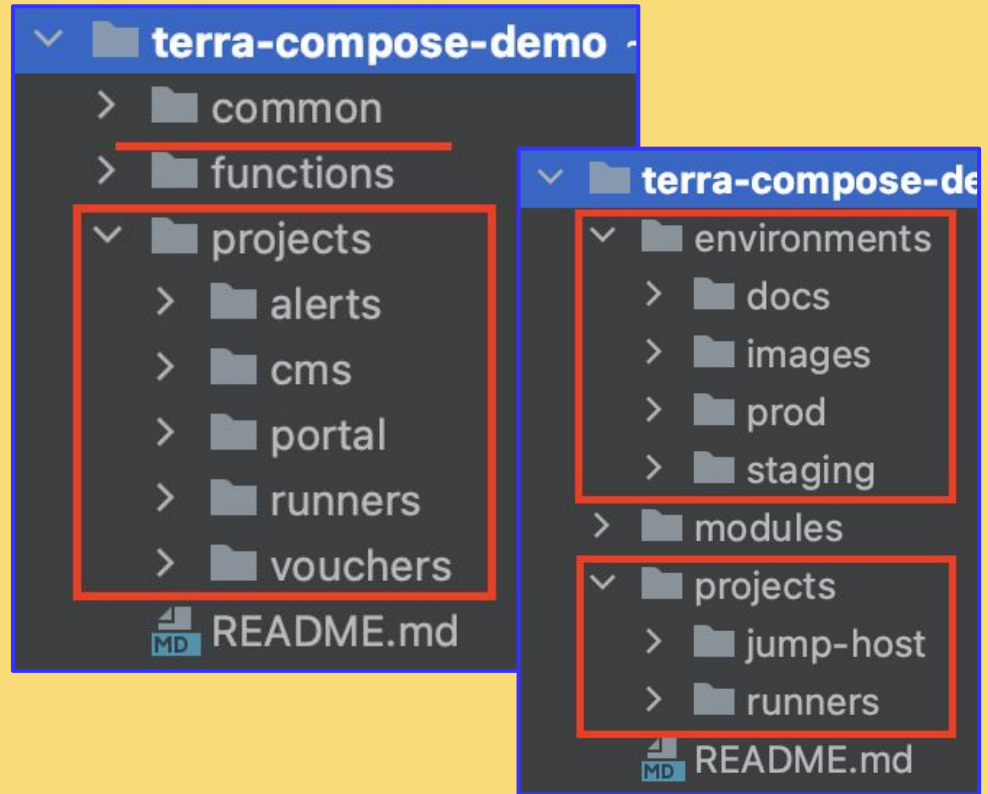
Prerequisites



- ✓ set of projects in a mono-repo
- ✓ various workspaces \approx environments
- ✓ various project structures
- ✓ various Terraform versions

Prerequisites

- ↳ **multiple projects**
- ↳ various workspaces
- ↳ various structures
- ↳ various versions



Prerequisites

- ↳ multiple projects
- ↳ **various workspaces**
- ↳ various structures
- ↳ various versions

```
# terraform workspace list
default
* live
stage
```

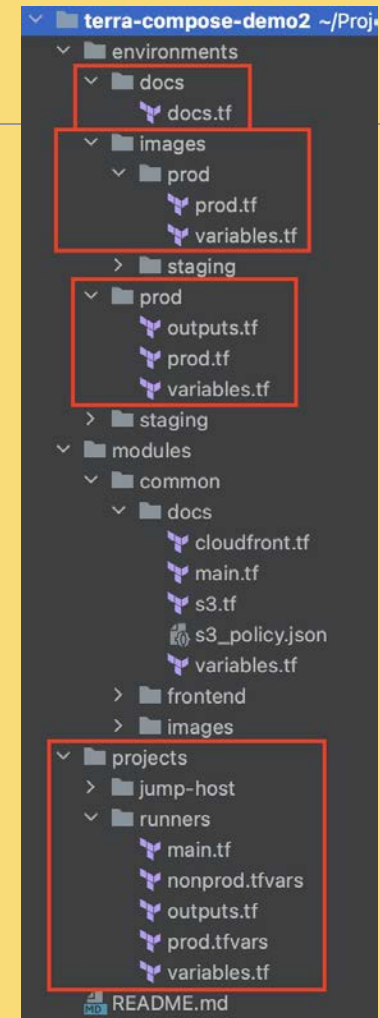
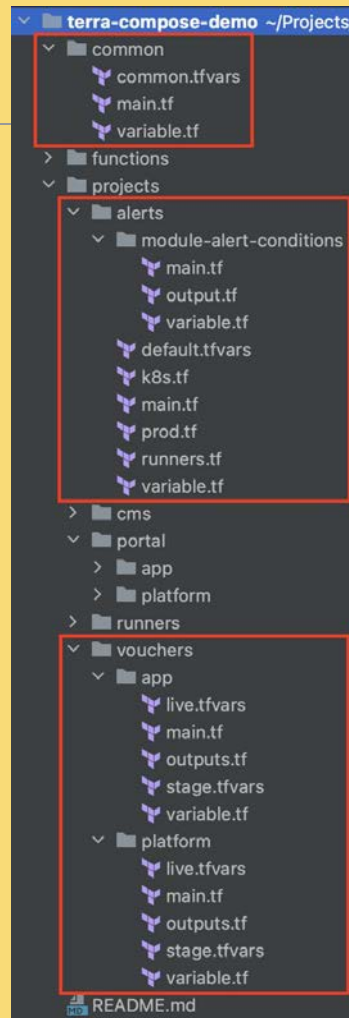
```
# terraform workspace list
default
production
release
* staging
```

```
# terraform workspace list
* default
```

```
# terraform workspace list
default
production
release
staging
* training
```


Prerequisites

- ↳ multiple projects
- ↳ various workspaces
- ↳ **various structures**
- ↳ various versions

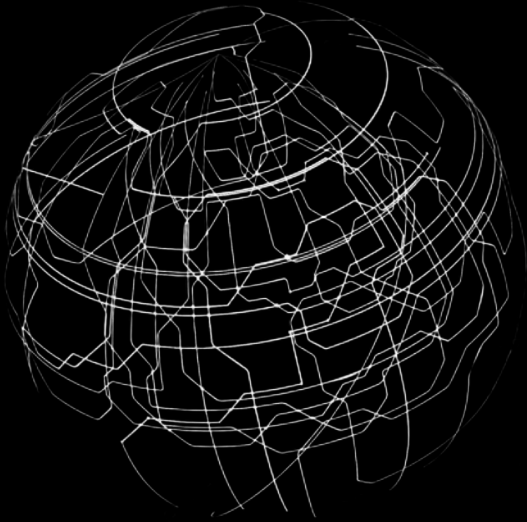


Prerequisites

- ↳ multiple projects
- ↳ various workspaces
- ↳ various structures
- ↳ **various versions**

```
$ cat aliases.yaml | grep tfversion | awk -F ":" '{print $2}' | uniq
"0.12.30"
"1.1.2"
"1.3.9"
$
```

So What??



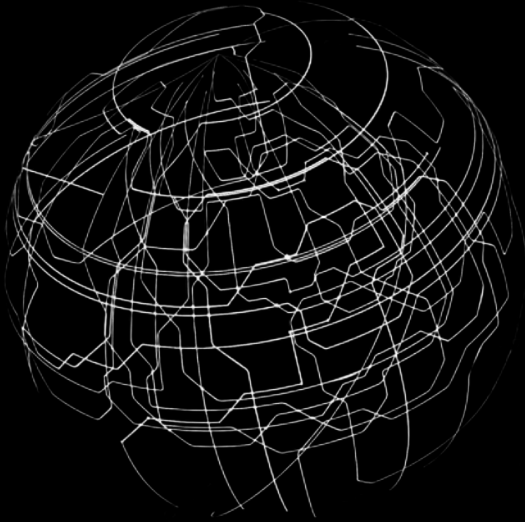
Pain points

- ☁ **different TF versions**
 - ↳ switching TF versions on local and CI
 - ↳ risk of the incompatibility in snippets
- ☁ **different subfolder structures/workspace**
 - ↳ unable to use unified snippets
 - ↳ WS: no visualisation of current workspace
 - ↳ WS: onboard newcomers, troubleshooting
 - ↳ WS: low visibility - are not trackable by VCS

upgrade all-at-once
OR switch projects
1-by-1 => long
creeping upgrade

... and all need to remember!

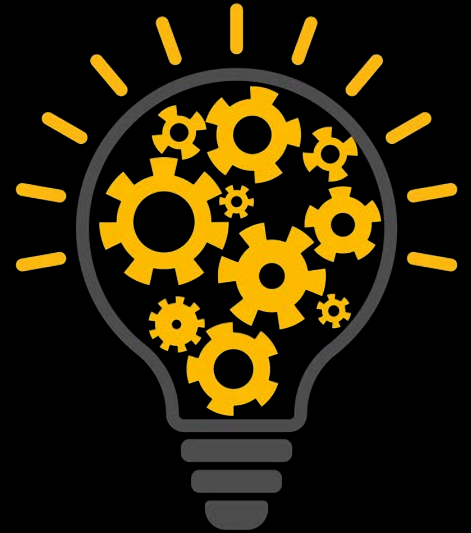
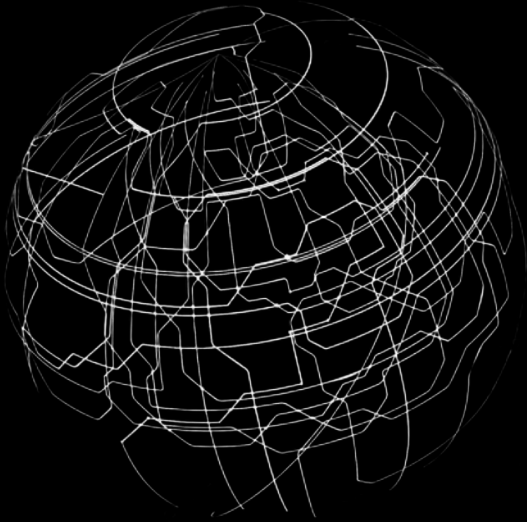
Aren't there any solutions??



Comparing solutions

Pain points	TF Cloud	Terragrunt	Terramate	Terraspace
Versioning: switching TF versions in LDE and CI	±	±	✗	✗
Versioning, structure: snippets incompatibility	✓	✓	?	✗
WS: no visualisation of current workspace	±	✓	✓	✓
WS: onboard newcomers , troubleshooting	✓	?	✗	✗
WS: low traceability - are not trackable by VCS	✗	✓	✓	±
Ability to automate routines , chaining commands	✗	✓	✗	✓
Needs to remember structure, WS, version etc	±	±	✗	✗
Complexity, time-to-implement , price	✗	✗	> Nov 2021	> Jul 2020

Wait, I saw it already...



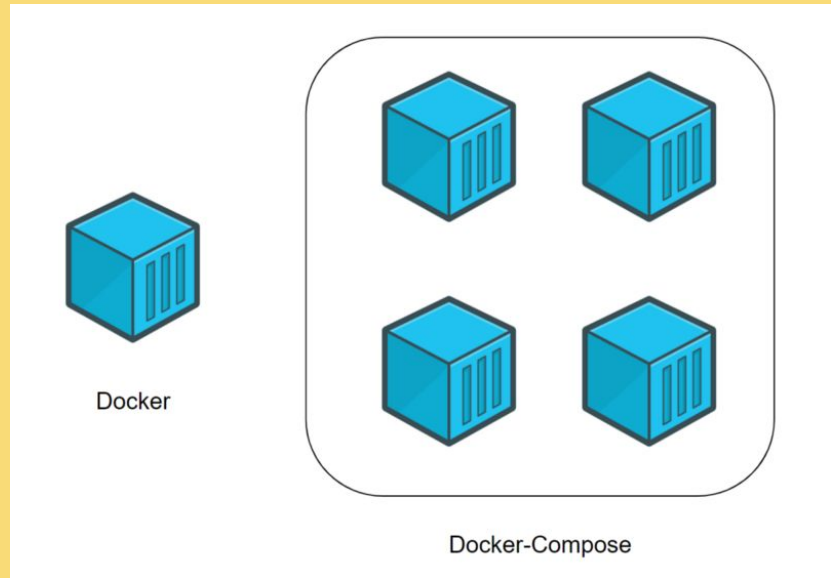
Terra Compose: The idea

Docker ➔ Docker Compose

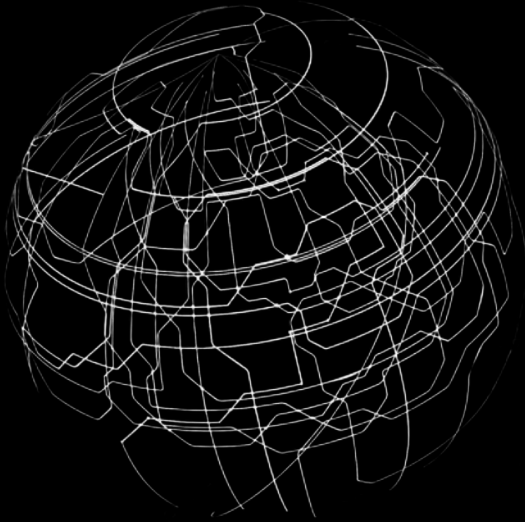
Terraform ➔ ???

Similarity:

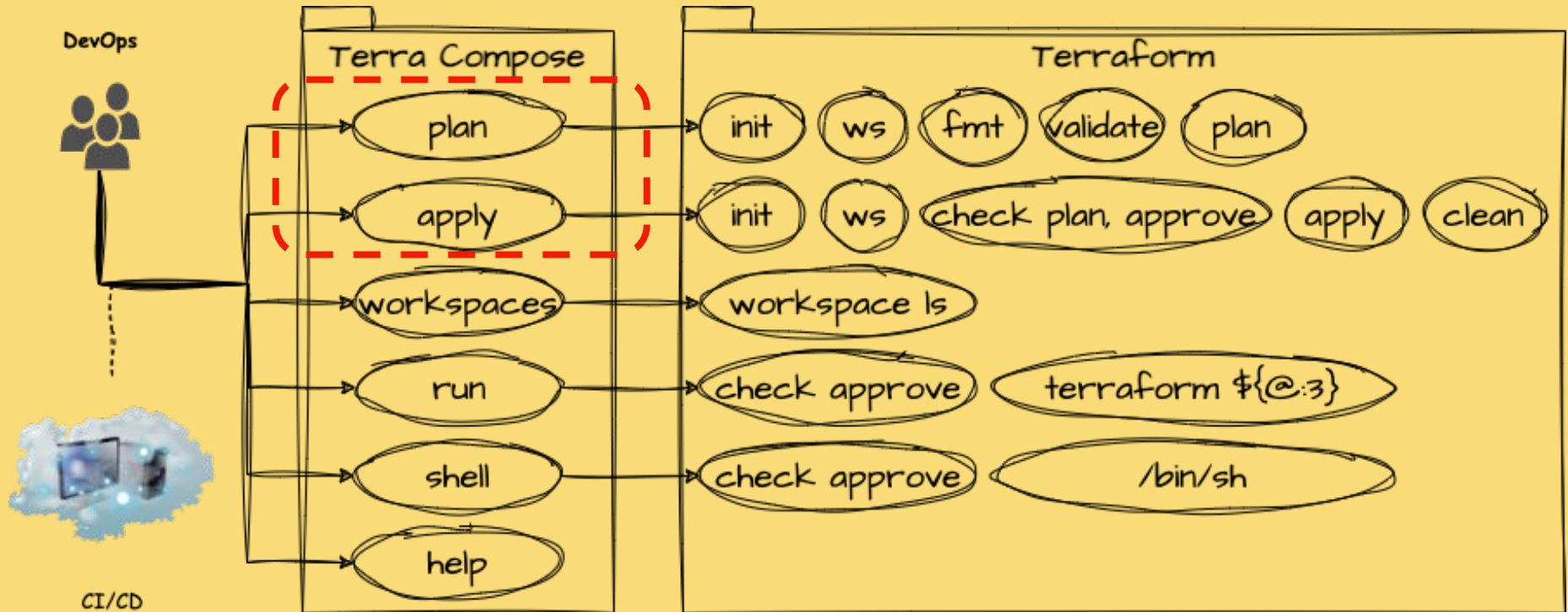
- ✓ same approach
- ✓ config-as-a-code
- ✓ high visibility,
traceability



Let's make a quick wrapper!



Terra Compose: Use-cases



Collaboration

✓ TF version update

✓ various structure

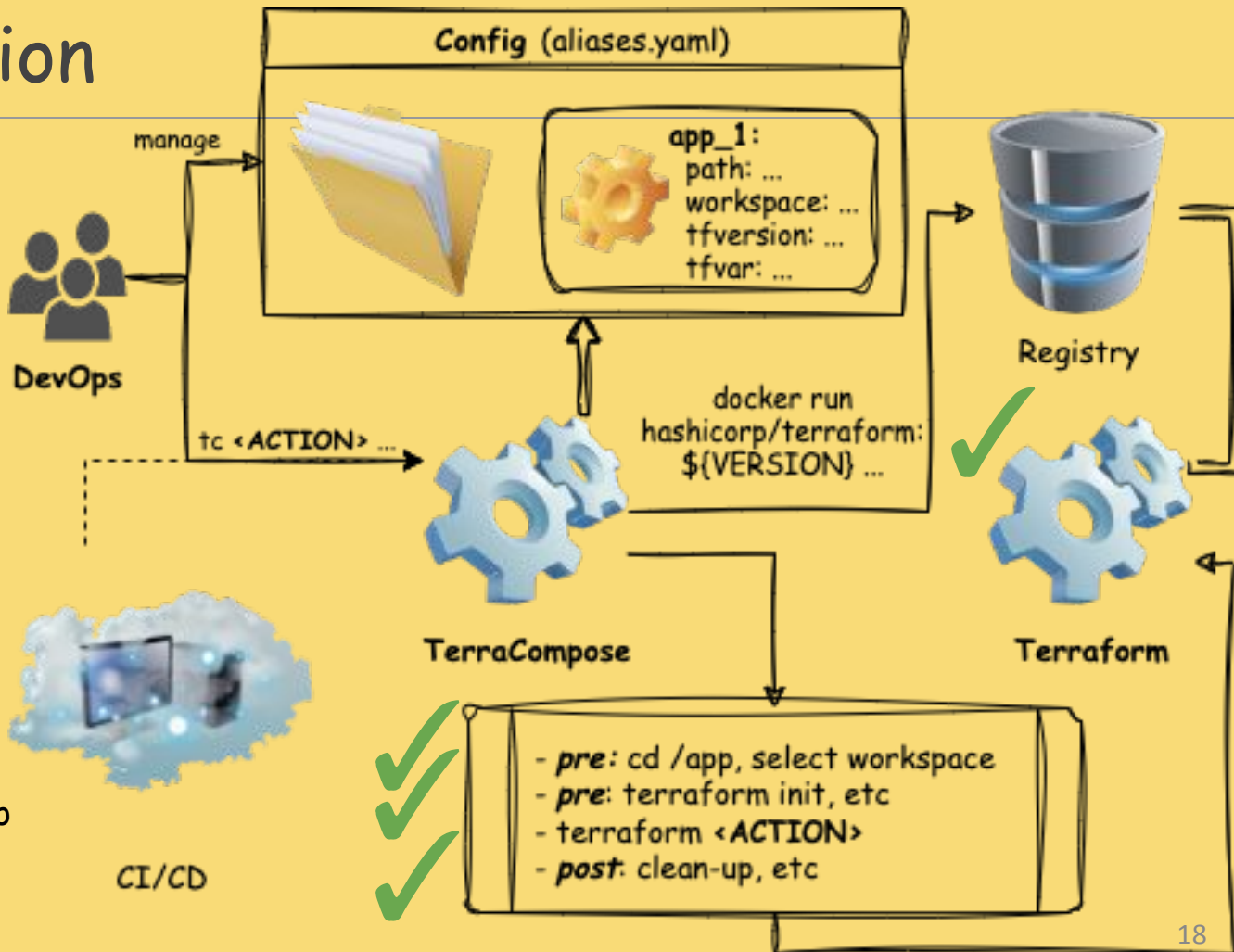
✓ various workspaces

✓ memorizing

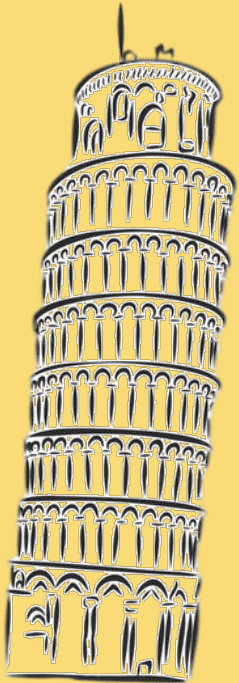
✗ a map needed

✓ chaining TF commands
+ validation

✓ post-actions / clean-up

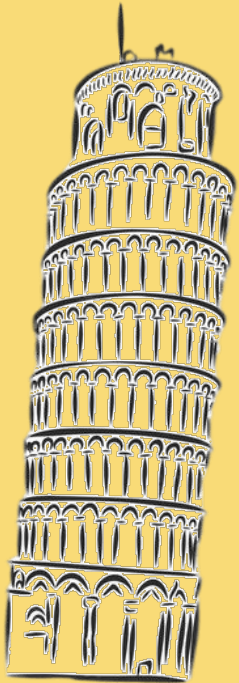


Terra Compose: Capabilities



- ✓ project attributes as-a-code
 - no need to remember -> error free
 - part of a code base -> same visibility
- ✓ composing multiple TF commands:
`tc plan => tf init + tf fmt + tf validate + tf plan`
 - guaranteed validated plan
 - speed-up maintenance
 - => less time for double-checks
 - => less time for printing repeated cmds
 - => less validation failures in CI

Terra Compose: Aliases example



```
default:  
→ tfversion: "0.12.0"
```

```
aliases:  
  common:  
→ path: "common"  
  tfvars: "common"  
  alerts:  
→ path: "projects/alerts"  
  workspace: "prod"  
→ tfversion: "1.3.0"
```

```
  hooks:  
    before_tf_init: "rm -rf ./terraform"
```

Terra Compose: Commands

FORMAT: `<SCRIPT_NAME> <ACTION> <ALIAS>`

`<ACTION>` - TF action you want to run:

- plan
- apply
- workspaces - equal to `workspace list`
- run - proxy all arguments directly to `terraform`.
 Allows running any custom command
- shell - run a shell inside the terraform docker container.
 Allows running any custom commands inside
- help - this help information

`<ALIAS>` - a unique shortcut to project/environment pair, like `runner_prod`.

Demo



TerraCompose



All done?



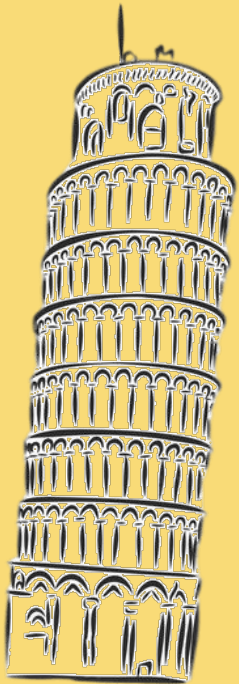
TerraCompose



Terra Compose: Solved ~~pain~~ points

Pain points	Terra Compose	
Versioning: switching TF versions on LDE and CI	✓	just an attribute in yaml
Versioning, structure: snippets incompatibility	✓	partially solved
WS: no visualisation of current workspace	✓	yaml
WS: onboard newcomers , troubleshooting	✓	IaaS/CaaS works well 🙌
WS: low traceability - are not trackable by VCS	✓	
Ability to automate routines , chaining commands	±	partially solved, chaining is still hardcoded...
Needs to remember structure, WS, version etc	✓	yaml
Complexity, time-to-implement , price	✓	simple yaml, ~1h, OSS

Terra Compose: Future



- 📌 re-think command interface
- 📌 make actions customisable
- 📌 re-write in Go ???
- 📌 ...



📍 <https://github.com/demmonico/terra-compose>

Thank you 🤘 🙌



TerraCompose

