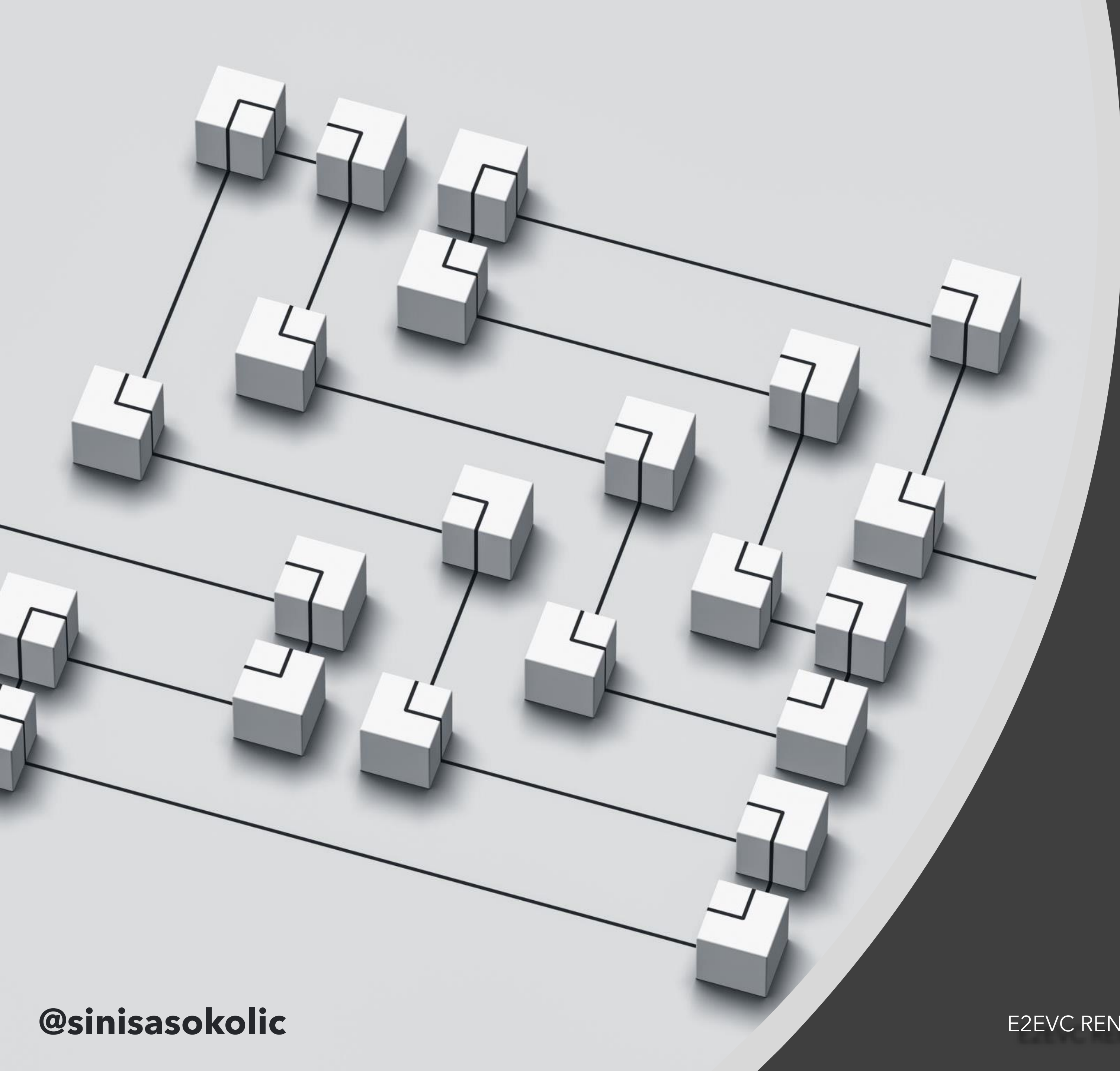


THE INFRASTRUCTURE AS CODE

Deployment Toolchain

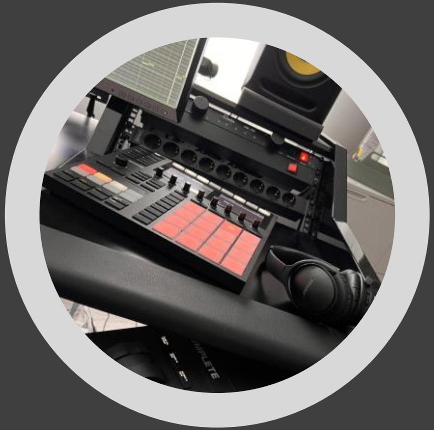
@sinisasokolic

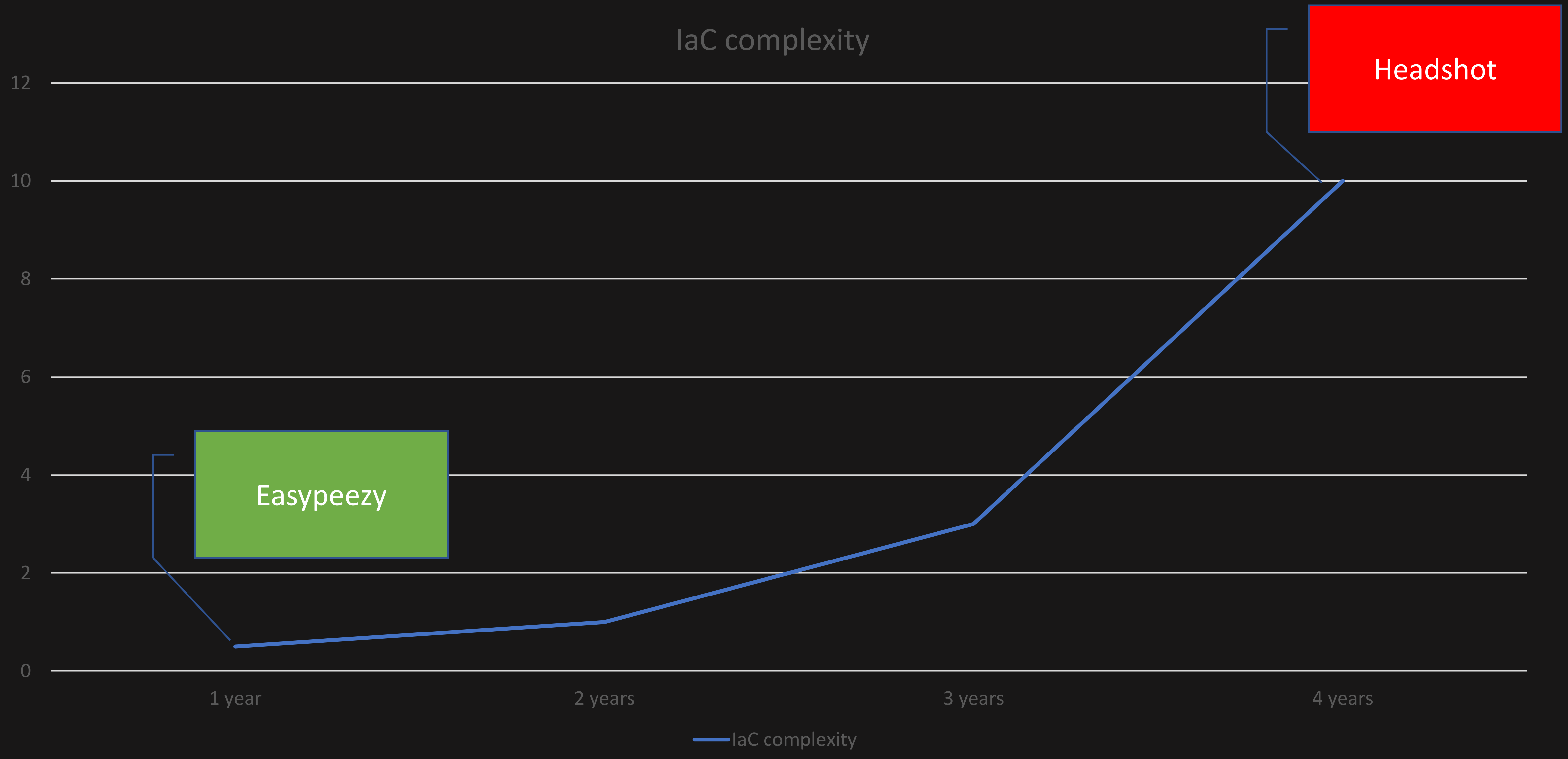
E2EVC RENAISSANCE – BERLIN 2022

Virtualization Conference

WHO AM I…

- My name is Sinisa
- CTO@RIS AG
- Product Owner@XOAP
- Born as Infrastructure Guy
- Trying to become a serious dev




IaC complexity
12
10
8
6
4
2
0
1 year
2 years
3 years
4 years
Easypeezy
Headshot
IaC complexity

THE IAC JOURNEY

OR: HOW WE LEARNED THAT WE WERE STUPID

E2EVC RENAISSANCE – BERLIN 2022

Virtualization Conference

TERRAFORM IMPLEMENTATION – v1.0.0

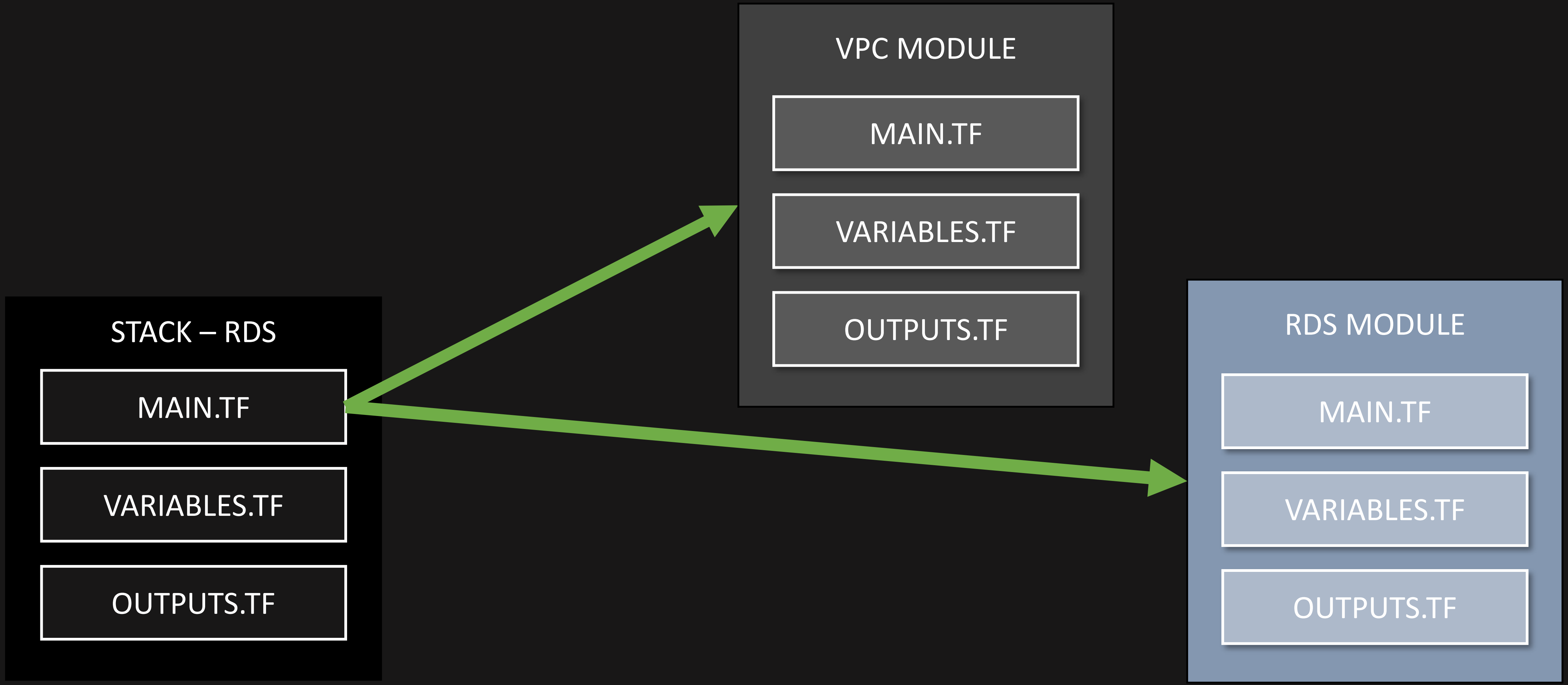
STACK - RDS

MAIN.TF

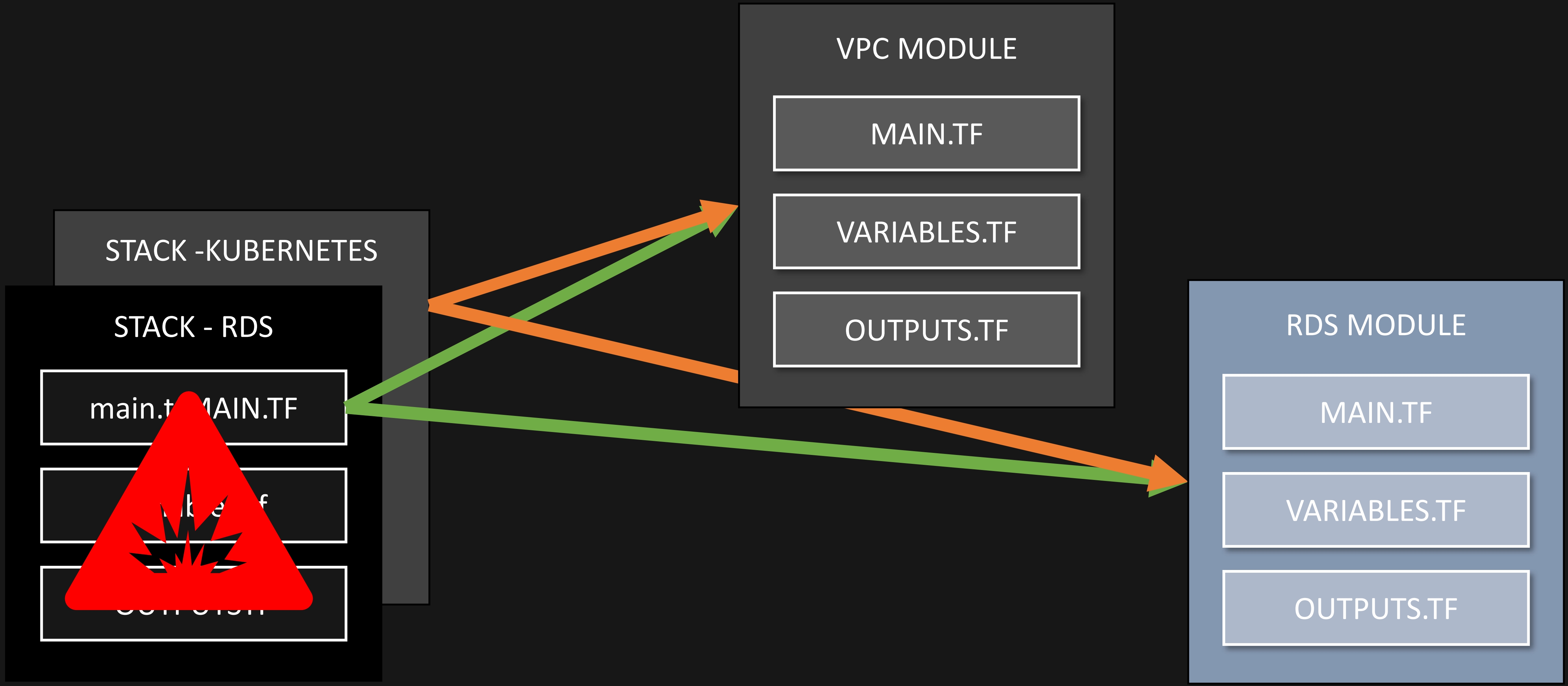
VARIABLES.TF

OUTPUTS.TF

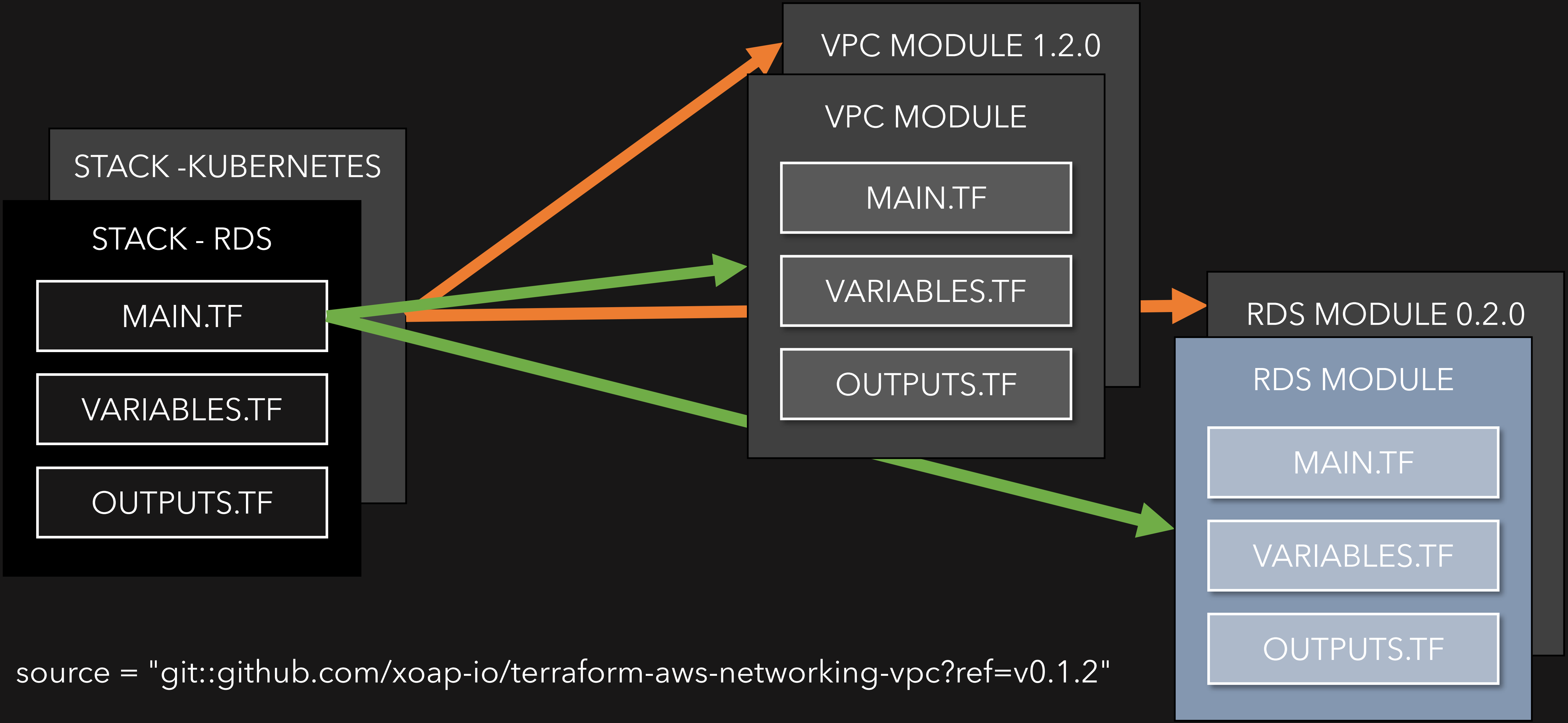
TERRAFORM IMPLEMENTATION – v2.0.0



TERRAFORM IMPLEMENTATION – v2.0.0

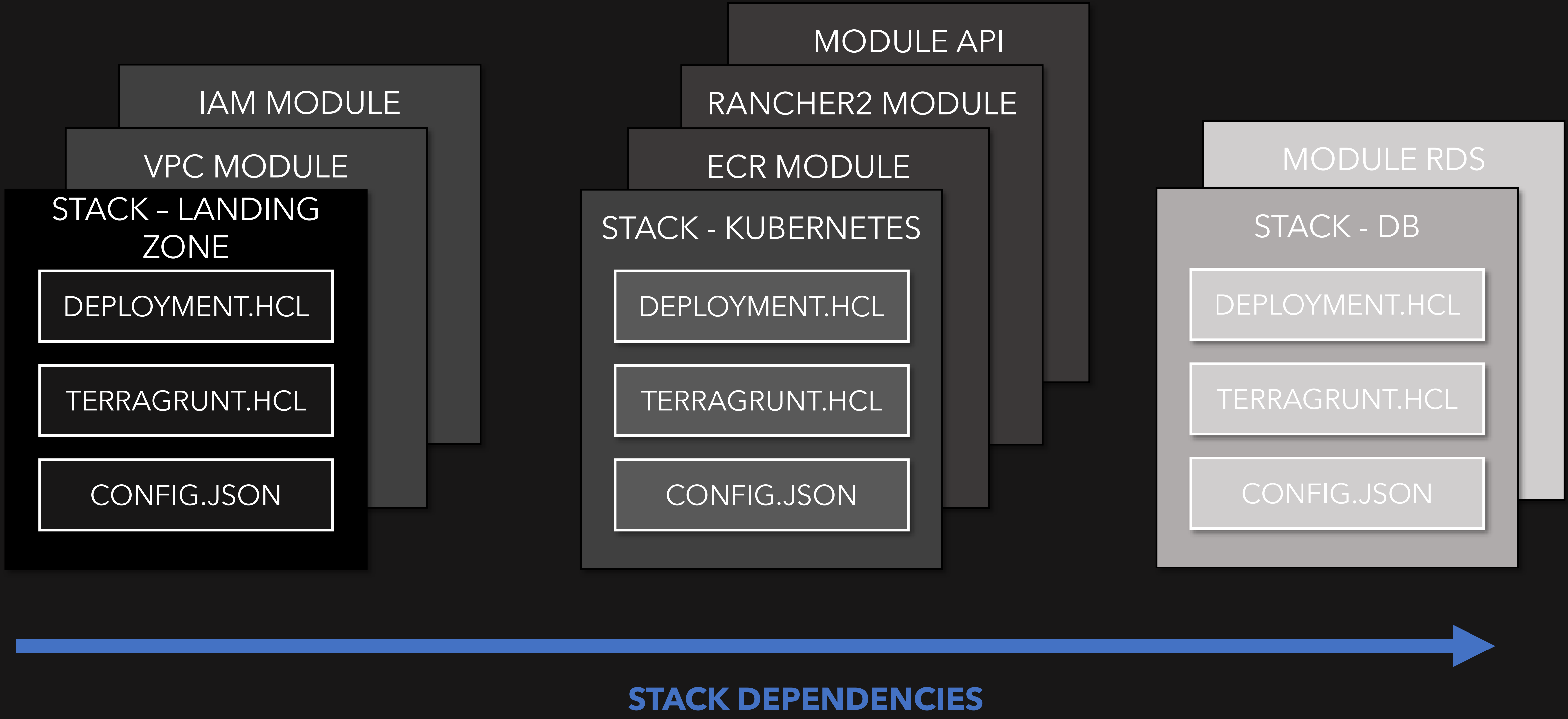


TERRAFORM IMPLEMENTATION – v3.0.0



source = "git::github.com/xoap-io/terraform-aws-networking-vpc?ref=v0.1.2"

TERRAFORM IMPLEMENTATION – v.4.0.0

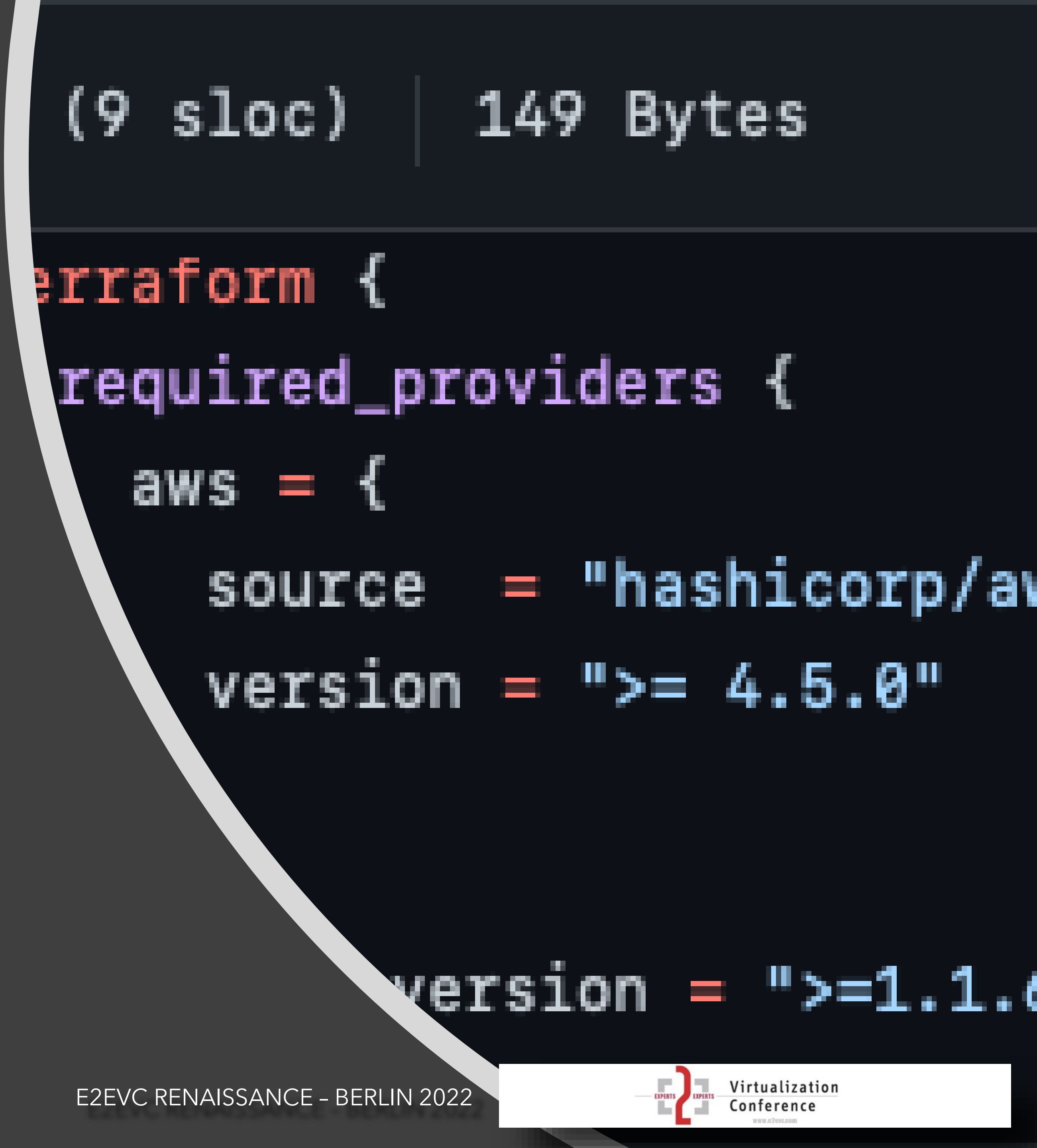


THE IAC JOURNEY

PAIN, HEADACHES, HUMAN ERRORS AND MORE…

99 PROBLEMS BUT VERSIONING AIN´T ONE

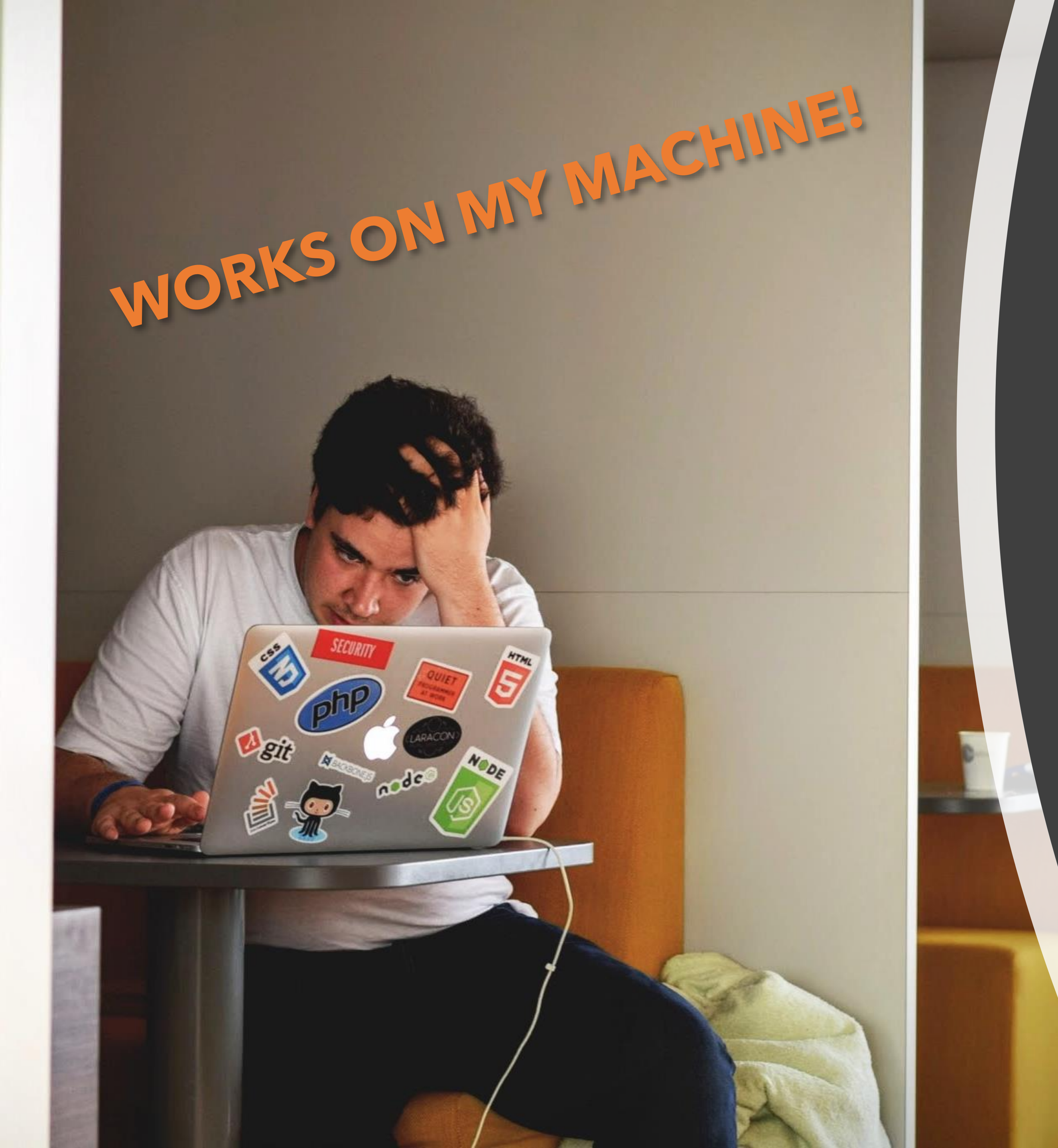
- Don´t use the “latest” tag never, ever
- Everything should have a version assigned or declared
  - Pipelines
  - Modules
  - Stacks
  - Extensions
  - Even: version = ">= 4.5.0” can get you into trouble
    - But it will also bring you in trouble when you hard-code versions
- Put your state files on versioned file shares and make backups



```
(9 sloc)  |  149 Bytes

erraform {
required_providers {
  aws = {
    source  = "hashicorp/aw
    version = ">= 4.5.0"



ersion = ">=1.1.
```

99 PROBLEMS BUT THE PIPELINE AIN´T ONE

- Pipelines are code either
- Pipelines will crash
  - Because of timeouts
  - Network interrupts
  - Cloud Provider issues
  - Sudden death
- Pipelines should run automatically, there is nothing like the pipeline guy, everybody is in the same boat
- Learn how to taint resources!
  - Terraform taint
- Learn how to adjust state files!
  - Terraform state



99 PROBLEMS BUT THE DEVELOPER AIN´T ONE

- Local versions vs. pipelines versions
  - e.g. running terraform 1.1.1 locally and latest in pipeline
- Prevent approving PRs by the one who requested it
- Prevent pushes to master
- Different coding styles
- Making manual changes in production accounts
  - Will kill your state files and pipelines
- Lack of architecture
- Spaghetti code
- Lack of documentation

99 PROBLEMS BUT THE CLOUD PROVIDER AIN´T ONE

- Check change logs frequently to be notified of provider changes
  - This will kill your pipeline and in worst case state files
- Services issues although status is green
- Service changes without proper notification
- Run pipelines every day
  - This won´t safe you from harm, but will reduce impacts
- Yes, we all are beta testers
  - We killed Opensearch multiple times and had to wait for AWS to fix it
- If Github or Azure repos are unavailable you won´t deploy anything
  - Replicate code and use hosted runners

99 PROBLEMS BUT TESTING AIN`T ONE

- Run pipelines frequently… daily
- Don´t deploy central services without a dev environment (dev Landing zone)
- Destroy, deploy, destroy, deploy…
- Test multiple times
- Developers are the worst testers
- Developers will find a way that test will always succeed
- PRs with multiple approvers
    - Even 4 eye principle sucks!

Virtualization Conference

99 PROBLEMS BUT SECURITY AIN´T ONE

- Don´t use wildcards in policies
- Don´t use weak passwords, even if you just test something

Thank you for letting me test.

- Least privileges, especially for your developers
- Don´t use access keys, use assume role or Vault for secrets
- Use everything that can improve security
- No users in root account, use central security account

99 PROBLEMS BUT OPERATIONS AIN´T ONE

- Never Change a Running System

Is the most dumb thing you can do!

- Don´t run Terraform on Windows:
  - Because not all providers work on it
  - Because a lot of available shell scripts don´t run on it
  - Because case sensitivity on cloud hosted pipelines kills your pipeline
- Having smooth pipelines is now one of your main responsibilities
- Testing too…

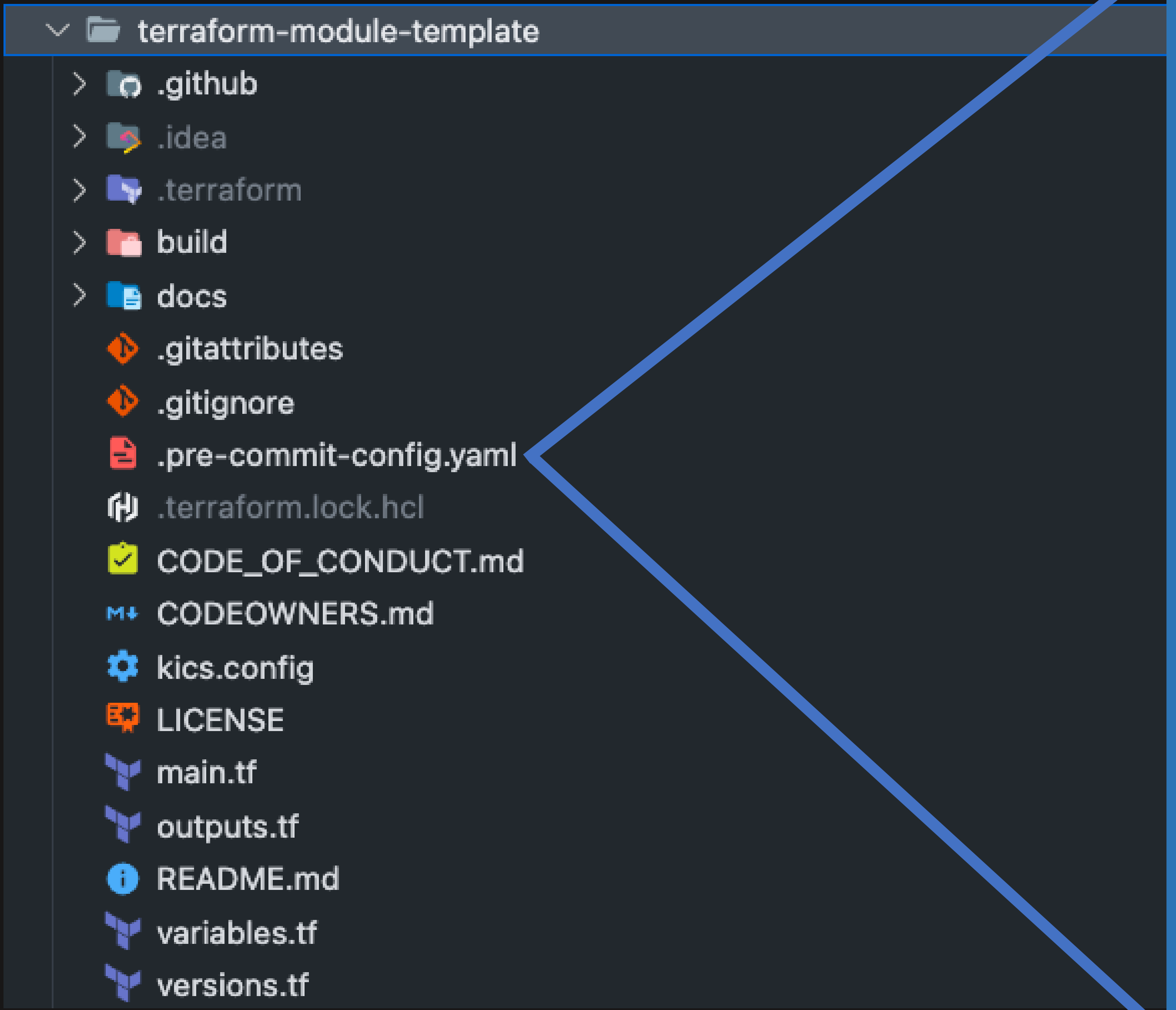
99 PROBLEMS BUT TERRAFORM AIN´T ONE

- Keep your code dry
- Get rid of duplicated backend code
- Keep your Stacks small
  - Define max. 200 resources
- Don´t run tf locally
- Implement State file versioning (locking)
- Don´t change stuff manually
  - If you really need import to State file afterwards
- Plan architecture wisely (Chicken/Egg problem)
- Don´t use Windows

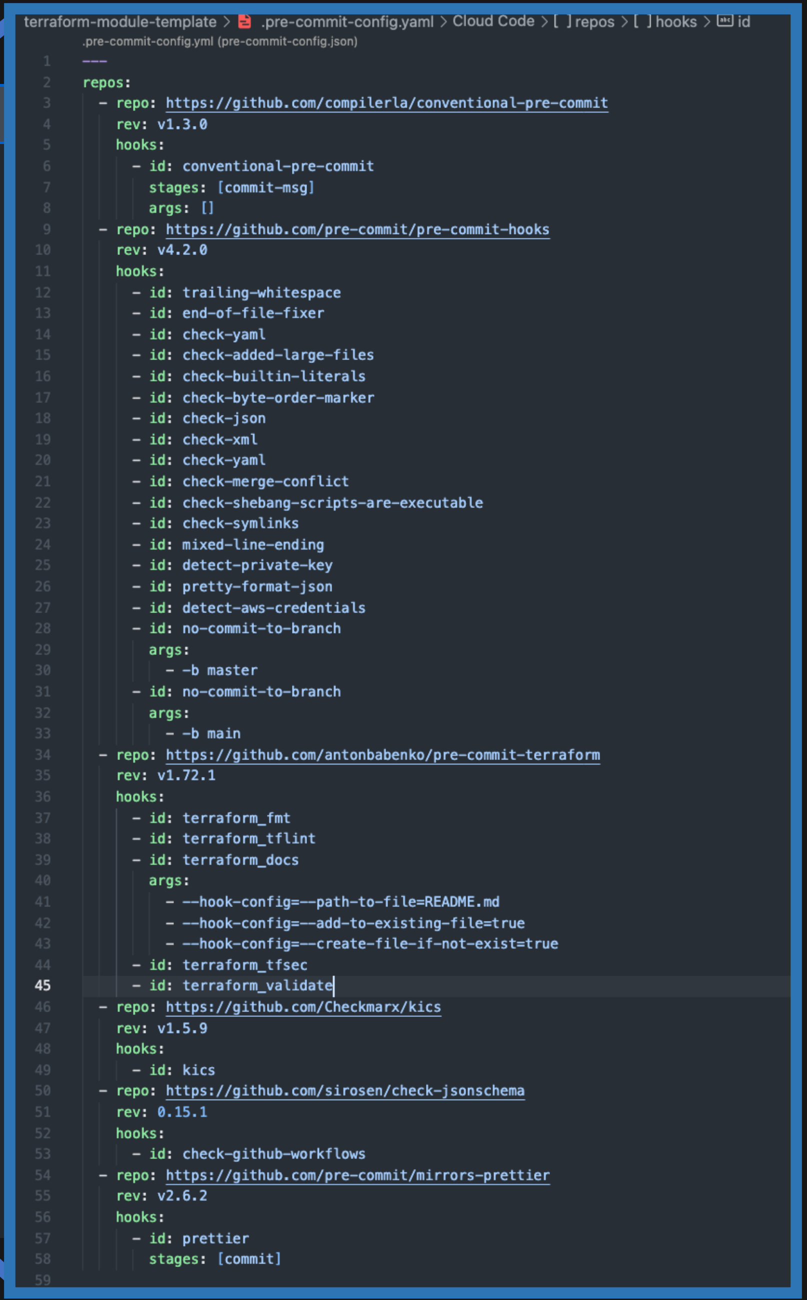
TERRAFORM IMPLEMENTATION – v.0.5.0

- insert to every full version wherever you want

Linting	Commit Conventions	Build Artefacts	Templates
Tagging	Branch Protection	BOM	EOL
Syntax Checks	Prettier	Repo policies	User Stories
URL Checks	tfdocs	Dependabot	Testing
Checkov	tfsec	Kics	Change logs
Infracost	Credential Checking	Landing Zones	Dependency hell



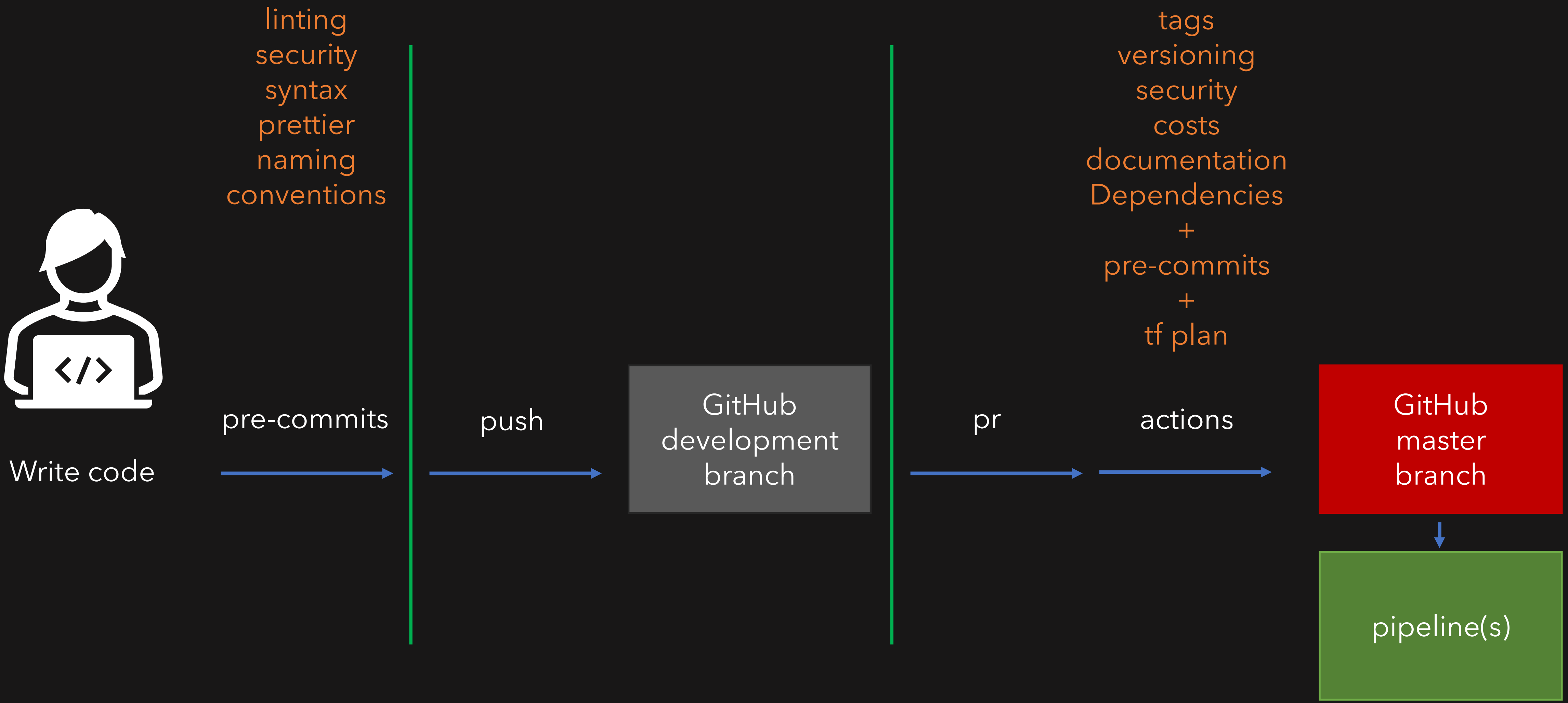
```
terraform-module-template
  .github
  .idea
  .terraform
  build
  docs
  .gitattributes
  .gitignore
  .pre-commit-config.yaml
  .terraform.lock.hcl
  CODE_OF_CONDUCT.md
  CODEOWNERS.md
  kics.config
  LICENSE
  main.tf
  outputs.tf
  README.md
  variables.tf
  versions.tf
```



terraform-module-template > .pre-commit-config.yaml > Cloud Code > [ ] repos > [ ] hooks > id
.pre-commit-config.yml (pre-commit-config.json)

```yaml
---
repos:
  - repo: https://github.com/compilerla/conventional-pre-commit
    rev: v1.3.0
    hooks:
      - id: conventional-pre-commit
        stages: [commit-msg]
        args: []
  - repo: https://github.com/pre-commit/pre-commit-hooks
    rev: v4.2.0
    hooks:
      - id: trailing-whitespace
      - id: end-of-file-fixer
      - id: check-yaml
      - id: check-added-large-files
      - id: check-builtin-literals
      - id: check-byte-order-marker
      - id: check-json
      - id: check-xml
      - id: check-yaml
      - id: check-merge-conflict
      - id: check-shebang-scripts-are-executable
      - id: check-symlinks
      - id: mixed-line-ending
      - id: detect-private-key
      - id: pretty-format-json
      - id: detect-aws-credentials
      - id: no-commit-to-branch
        args:
          - -b master
      - id: no-commit-to-branch
        args:
          - -b main
  - repo: https://github.com/antonbabenko/pre-commit-terraform
    rev: v1.72.1
    hooks:
      - id: terraform_fmt
      - id: terraform_tflint
      - id: terraform_docs
        args:
          - --hook-config=--path-to-file=README.md
          - --hook-config=--add-to-existing-file=true
          - --hook-config=--create-file-if-not-exist=true
      - id: terraform_tfsec
      - id: terraform_validate
  - repo: https://github.com/Checkmarx/kics
    rev: v1.5.9
    hooks:
      - id: kics
  - repo: https://github.com/sirosen/check-jsonschema
    rev: 0.15.1
    hooks:
      - id: check-github-workflows
  - repo: https://github.com/pre-commit/mirrors-prettier
    rev: v2.6.2
    hooks:
      - id: prettier
        stages: [commit]

```

WORKFLOW


Write code
linting
security
syntax
prettier
naming
conventions
pre-commits
push
GitHub
development
branch
pr
tags
versioning
security
costs
documentation
Dependencies
+
pre-commits
+
tf plan
actions
GitHub
master
branch
pipeline(s)

TERRAFORM JOURNEY

	Terraform v.1.0.0	Terraform v.2.0.0	Terraform v.3.0.0	Terraform v.4.0.0	Terraform v.4.5.0
Modules	One Module	Multiple	Multiple	A lot	A lot
Repositories	One Git repo	One Git repo	Multiple Git repos	A lot	A lot
Git Branches	mostly none, tf_vars	Dev,acc,prd, Often none, tf_vars	Dev, acc, prd, Development, PRs	A lot, PRs	A lot, PRs
Collaboration	One individual	One individual	Multiple individuals	Some teams	Multiple Teams
Versioning	No versioning, local state file	No versioning, local state, state in git	Versioning, central multiple state files (maybe locking)	Versioning, multiple state files, locking	Versioning, multiple state files, locking
Infrastructure complexity	One tenant or subscription	One tenant or subscription	Multiple accounts	Landing Zone	Multi Cloud
Planning	none	None, maybe Kanban	Kanban or SCRUM	SCRUM	SCRUM of SCRUMS
Orchestration	Terraform	Terraform	Terraform	Terragrunt	Terragrunt

VISIT THE MASTERCLASS AT

14:00

@SINISASOKOLIC

S.SOKOLIC@RIS.AG