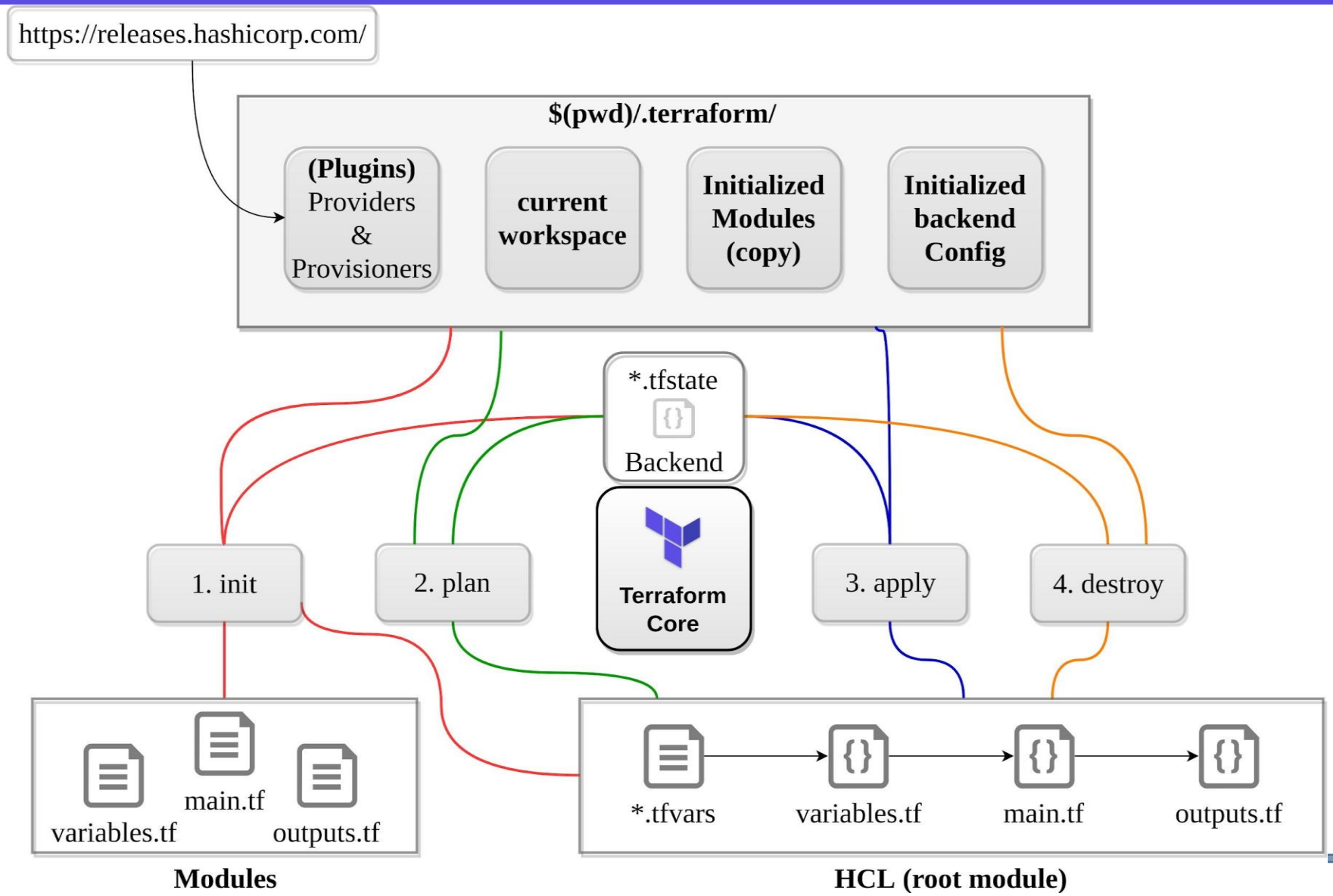
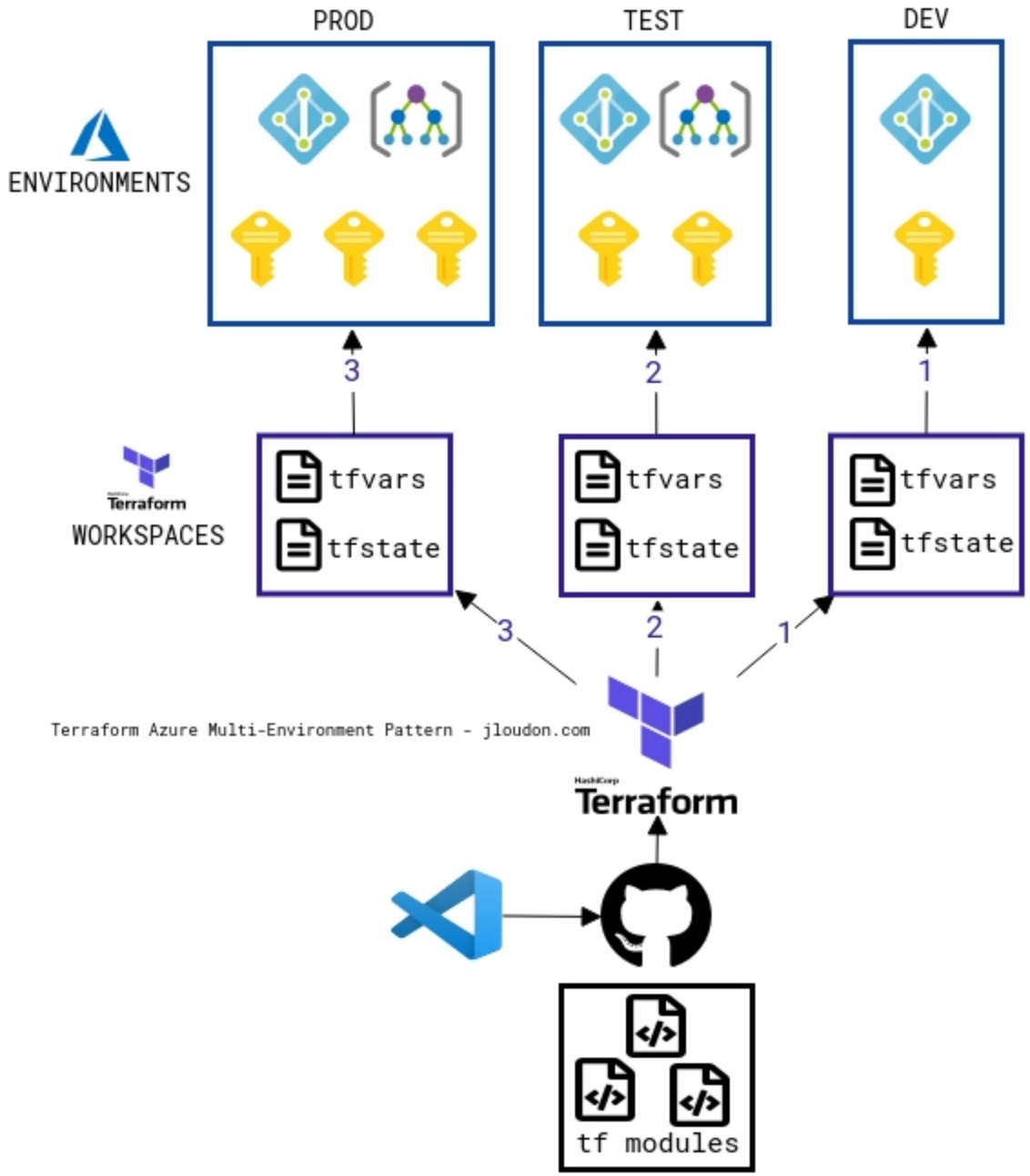
DevOpsSchool
Lets Learn, Share & Practice DevOps

Terraform Workspace

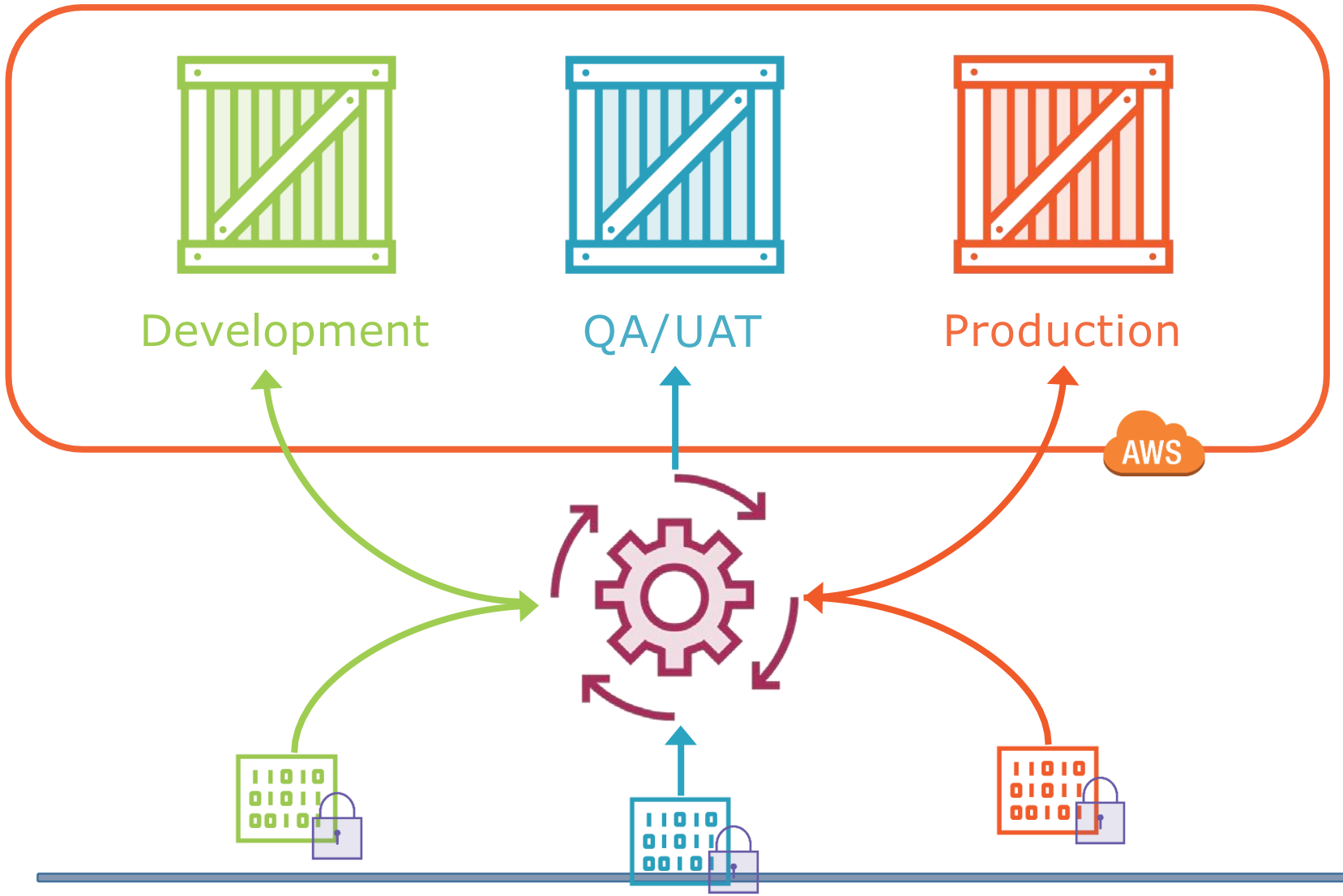
Terraform workflow

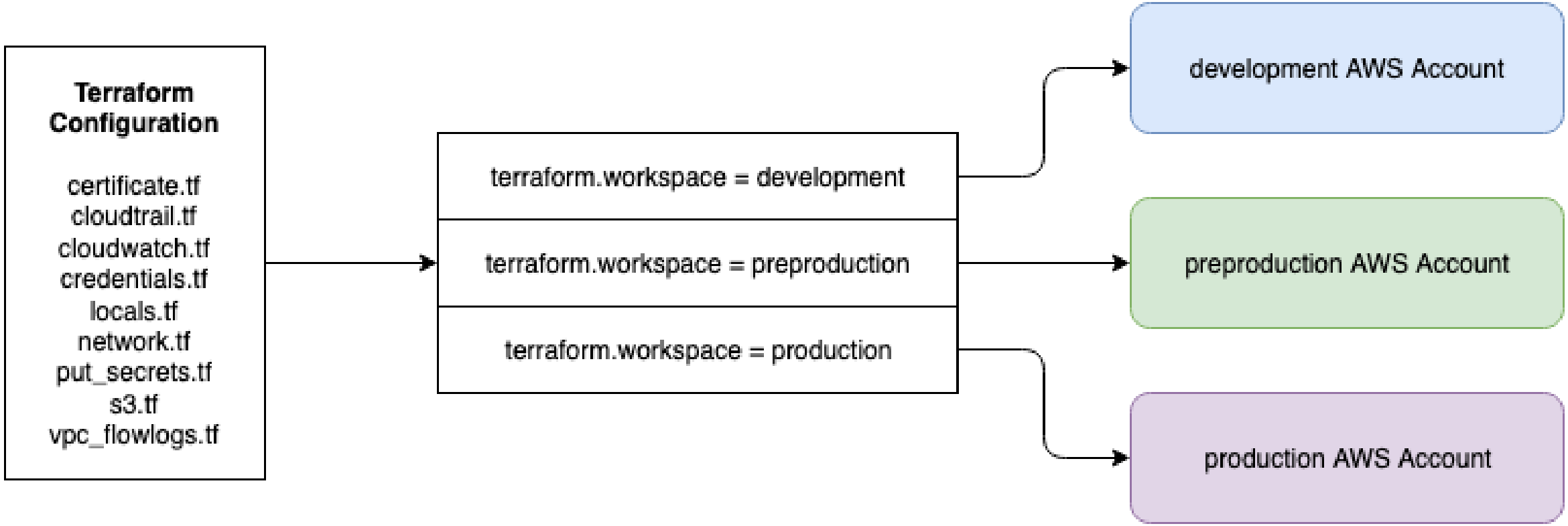



PROD
TEST
DEV
ENVIRONMENTS
3
2
1
Terraform
WORKSPACES
tfvars
tfstate
tfvars
tfstate
tfvars
tfstate
3
2
1
Terraform Azure Multi-Environment Pattern - jloudon.com
Terraform
tf modules

DevOpsSchool
Lets Learn, Share & Practice DevOps

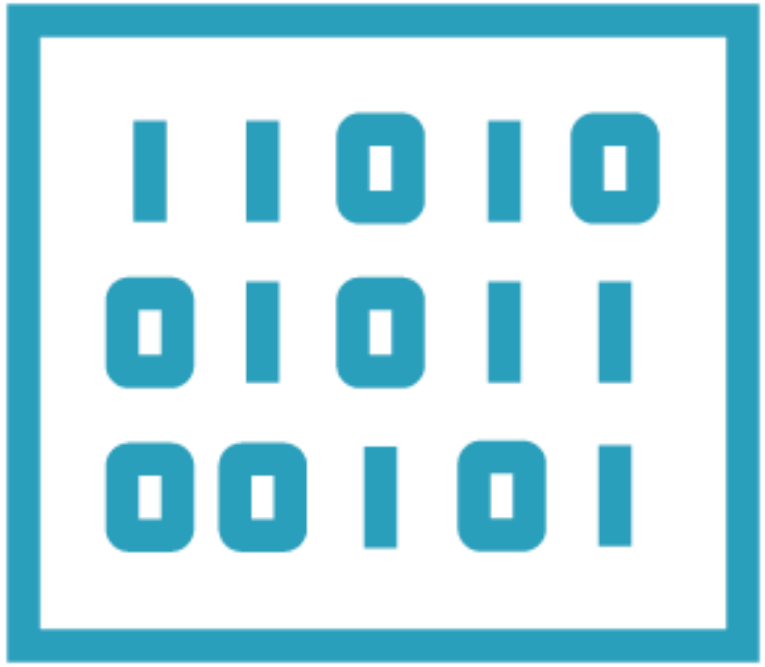
The Scenario




Terraform Configuration
certificate.tf
cloudtrail.tf
cloudwatch.tf
credentials.tf
locals.tf
network.tf
put_secrets.tf
s3.tf
vpc_flowlogs.tf
terraform.workspace = development
terraform.workspace = preproduction
terraform.workspace = production
development AWS Account
preproduction AWS Account
production AWS Account

DevOpsSchool
Lets Learn, Share & Practice DevOps

Multiple Environments



State file commands

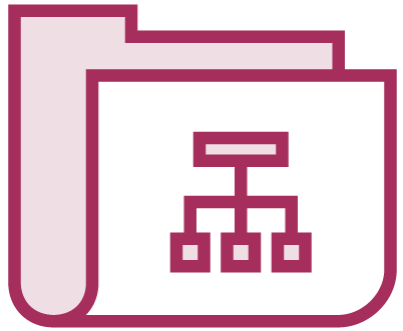
State file storage

Folder structure

Common patterns

DevOpsSchool
Lets Learn, Share & Practice DevOps

State File Example



main_config.tf
variables.tf

dev.state

uat.state

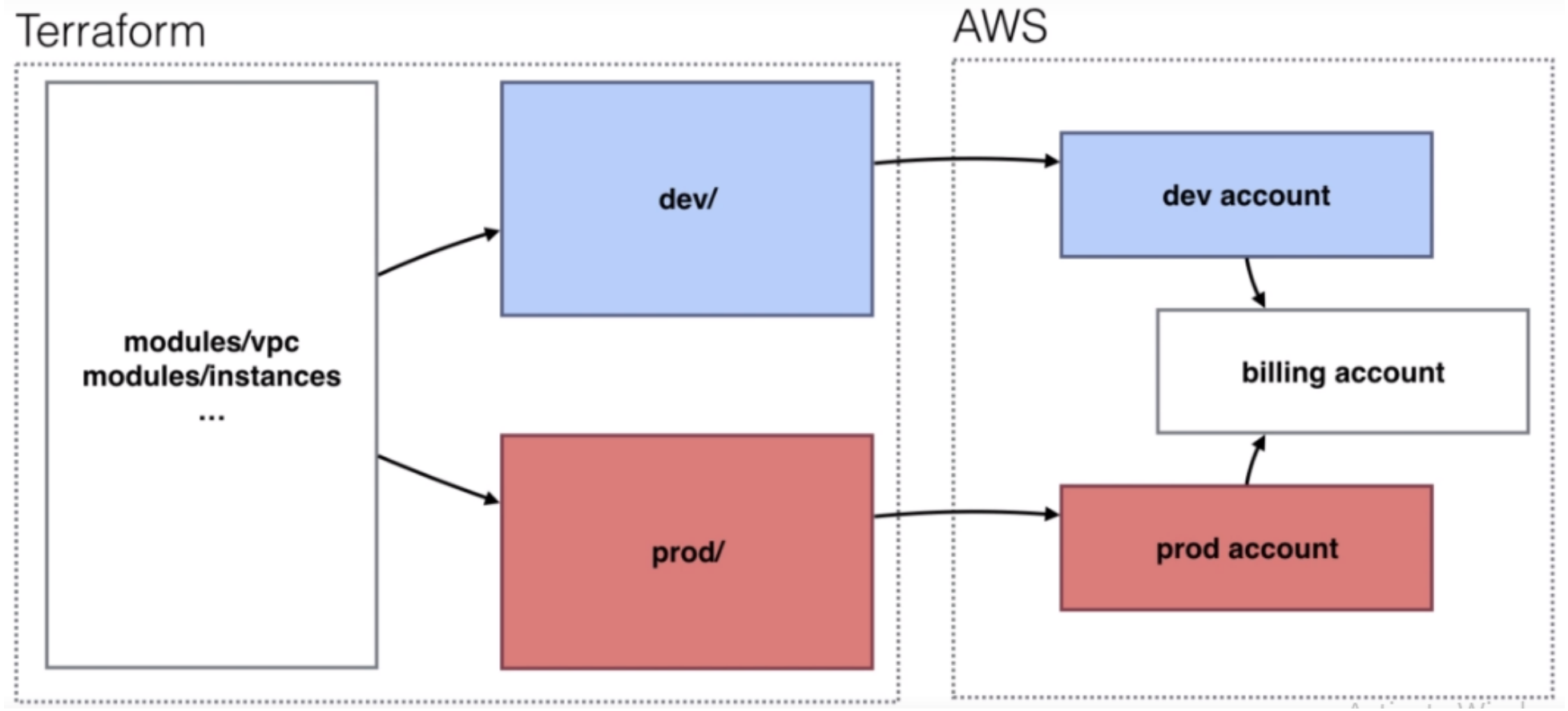
prod.state

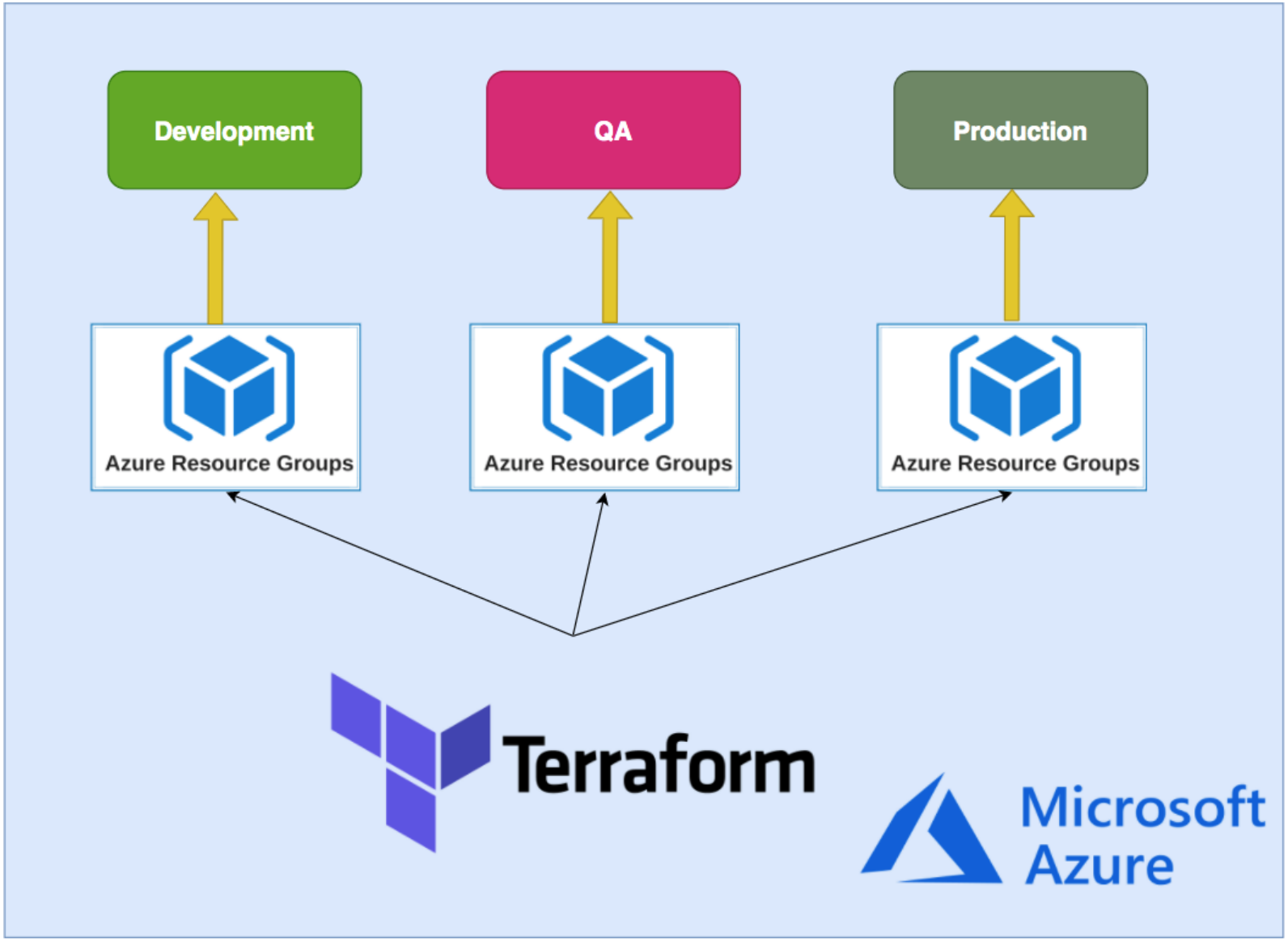
```
C:\>terraform apply -state="..\development\dev.state" `
  -var="environment_name=development"
```

Project structure

- When starting with terraform on production environments, you quickly realize that you need a decent project structure
- Ideally, you want to separate your development and production environment completely
    - That way, if you always test terraform changes in development first, mistakes will be caught before they can have a production impact
    - For complete isolation, it's best to create multiple AWS accounts, and use one account for dev, another for prod, and a third one for billing
    - Splitting out terraform in multiple projects will also reduce the resources that you'll need to manage during one terraform apply

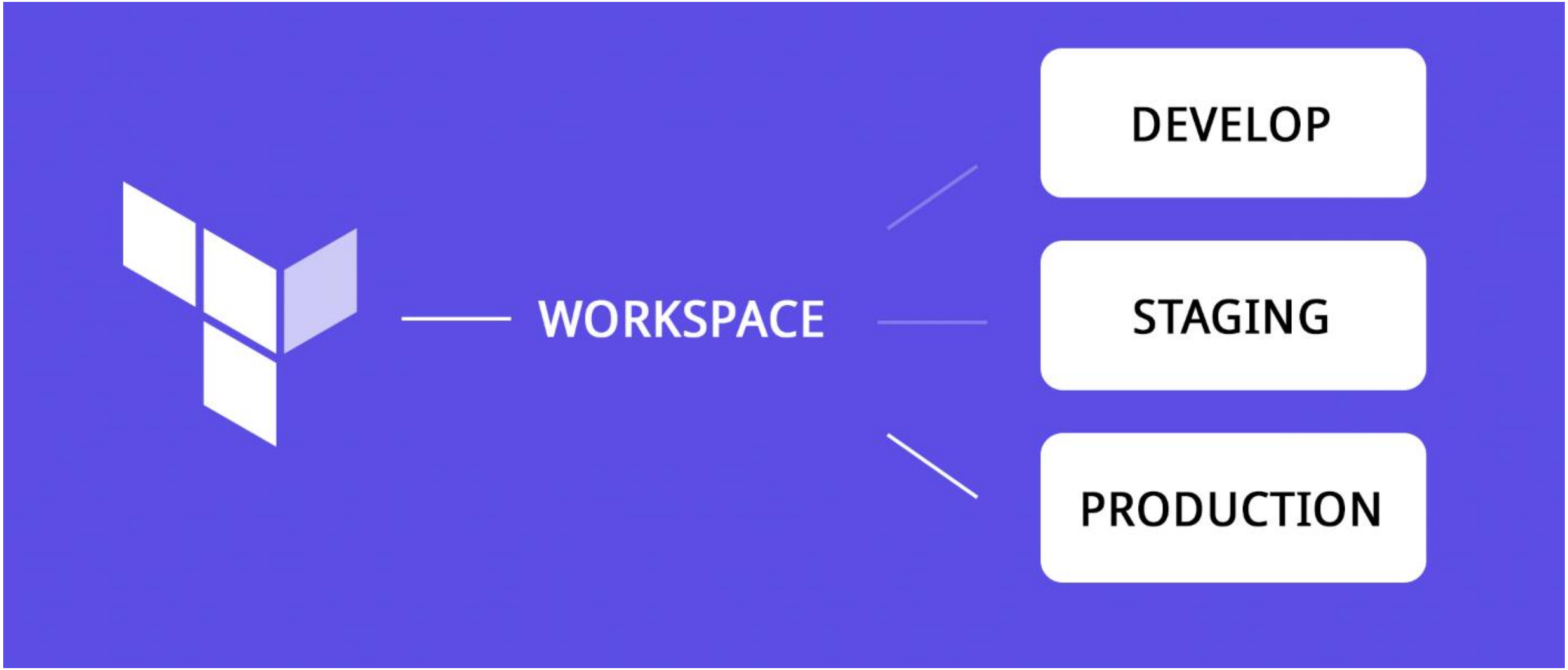
Project structure

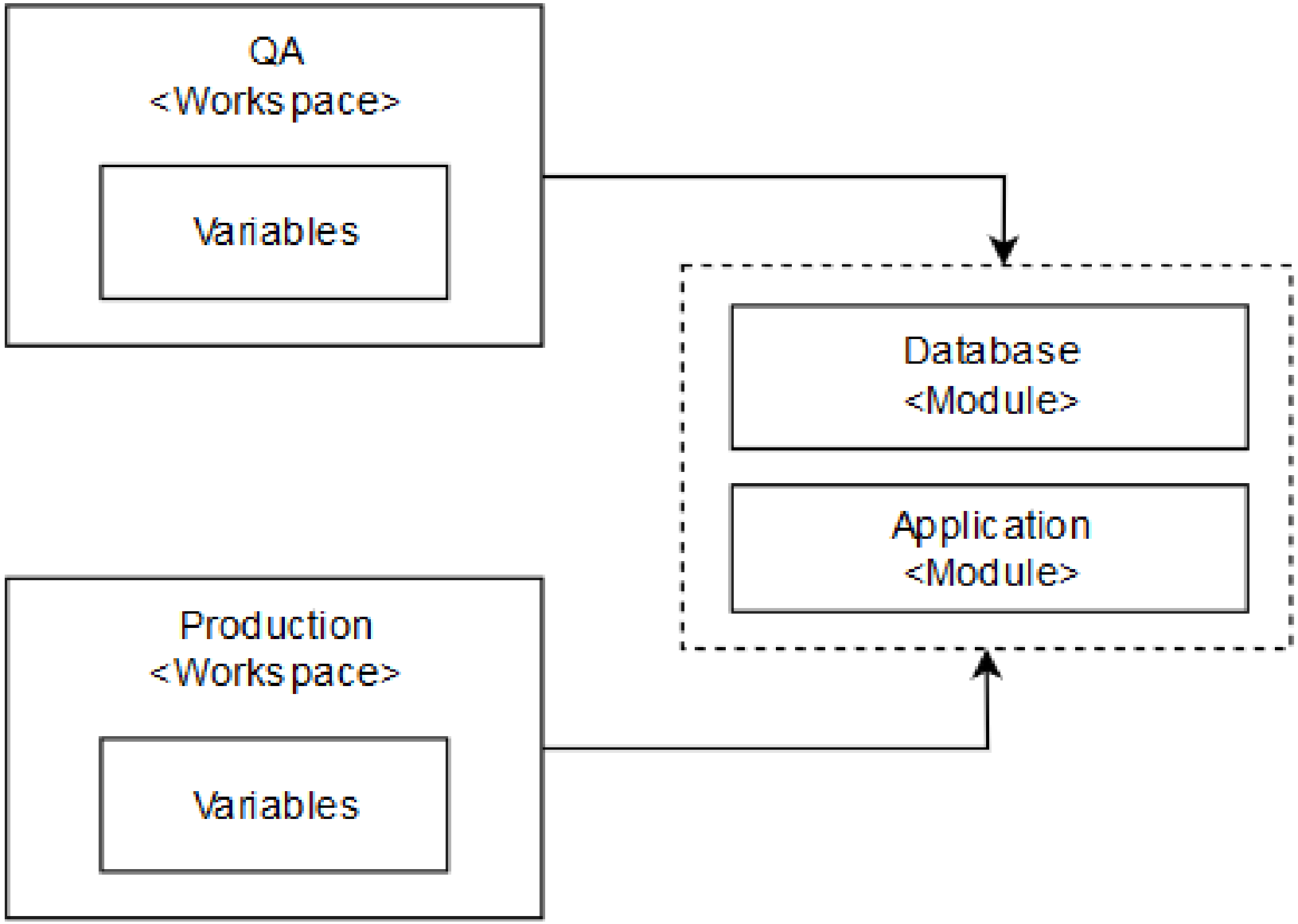


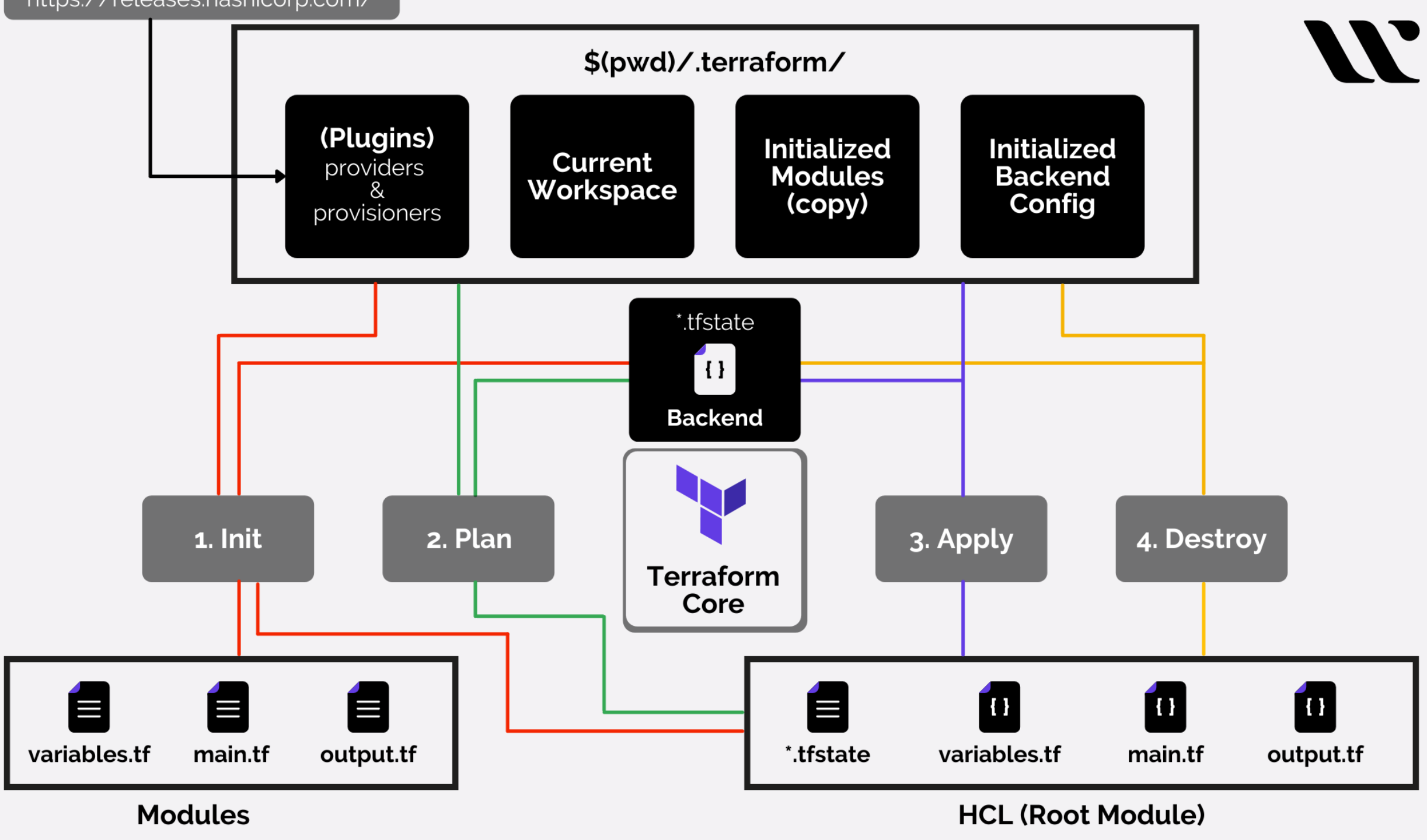

Development
QA
Production
Azure Resource Groups
Azure Resource Groups
Azure Resource Groups
Terraform
Microsoft Azure

DevOpsSchool
Lets Learn, Share & Practice DevOps

Solutions


WORKSPACE
DEVELOP
STAGING
PRODUCTION


QA
<Workspace>
Variables
Database
<Module>
Application
<Module>
Production
<Workspace>
Variables


https://releases.hashicorp.com/
$(pwd)/.terraform/
(Plugins)
providers
&
provisioners
Current
Workspace
Initialized
Modules
(copy)
Initialized
Backend
Config
*.tfstate
{ }
Backend
1. Init
2. Plan
Terraform
Core
3. Apply
4. Destroy
variables.tf
main.tf
output.tf
Modules
*.tfstate
variables.tf
main.tf
output.tf
HCL (Root Module)

https://www.devopsschool.com/blog/terraform-workspace-explained/