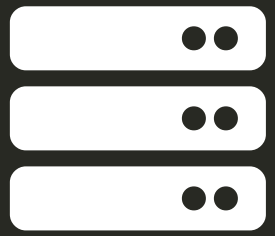
Infrastructure as Code - Terraformujeme cloud

Viliam Púčik
DevOps Tech Lead

ZOOM International

On Premise vs Cloud



On Premise

Cloud

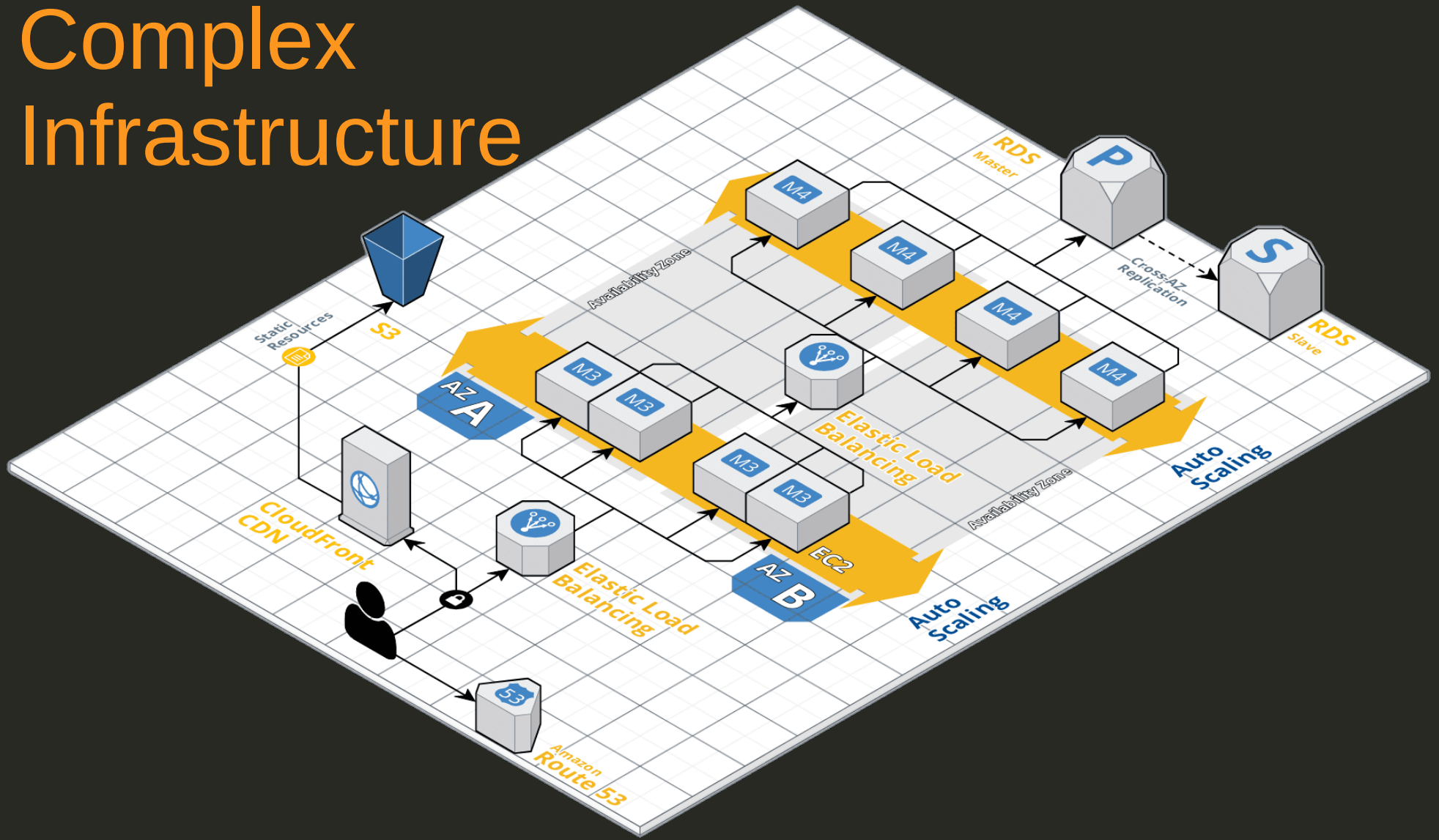
Major Cloud Providers

Amazon
Web
Services

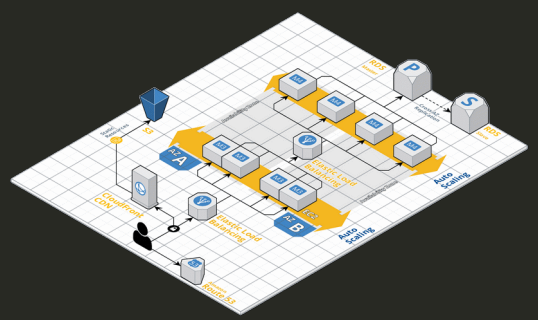
Google
Cloud
Platform

Microsoft
Azure

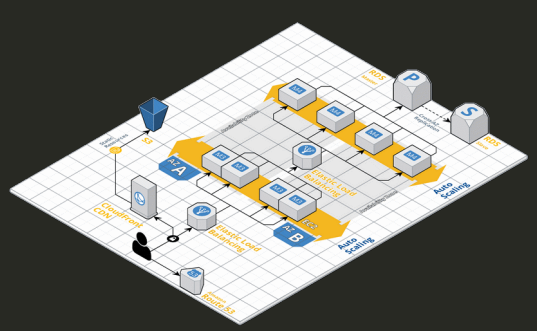
Complex Infrastructure



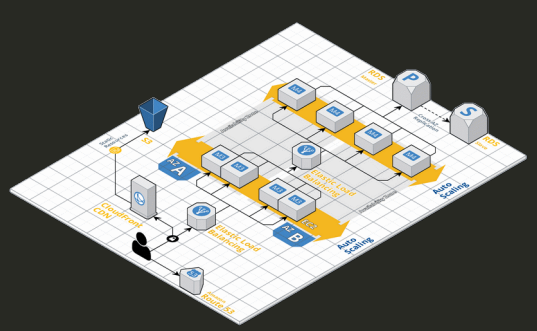
Complex Infrastructure



Development
Environment

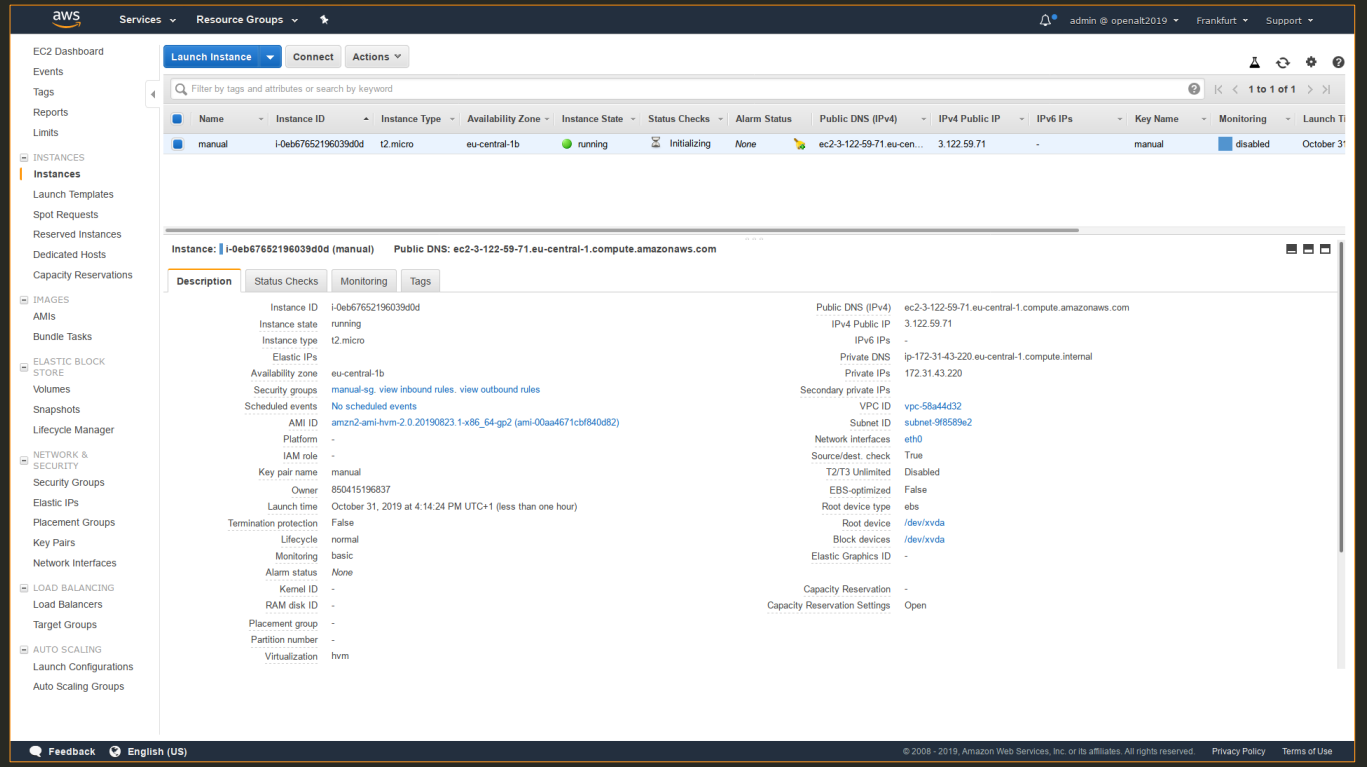


Staging
Environment

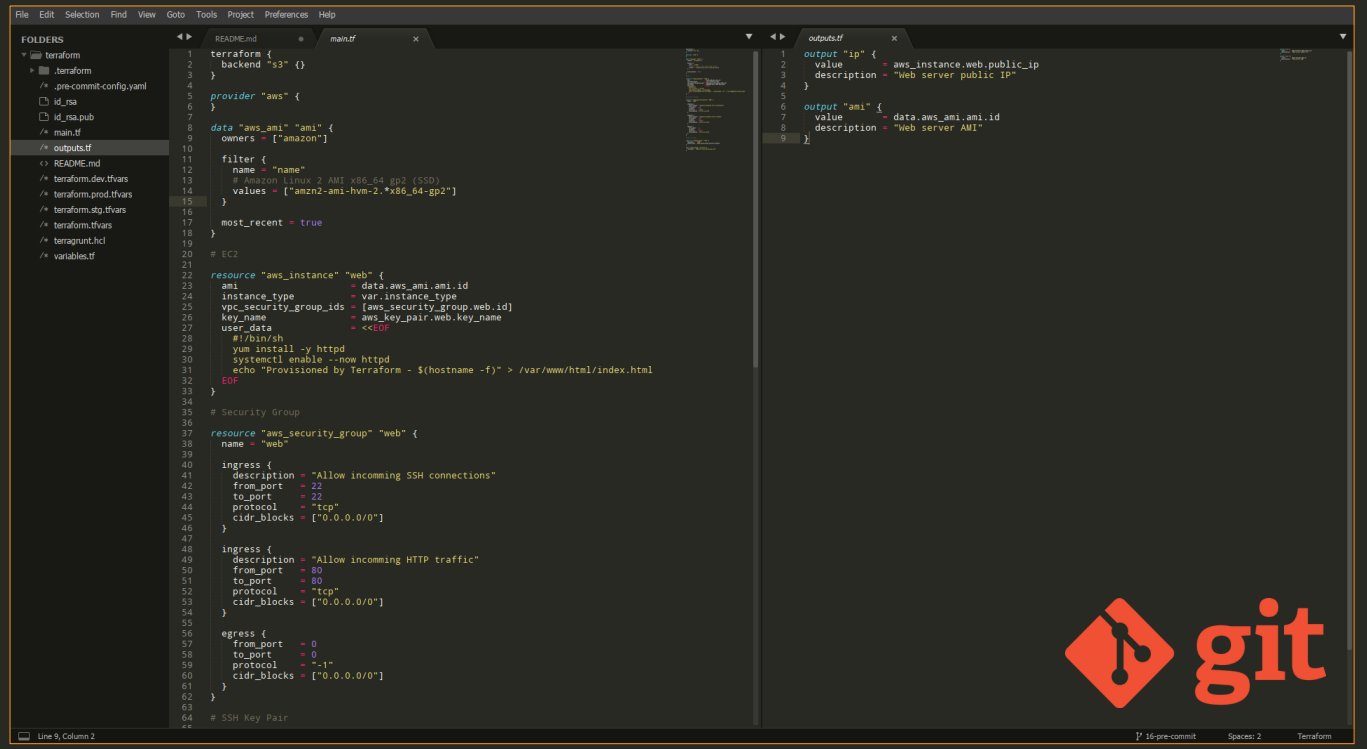


Production
Environment

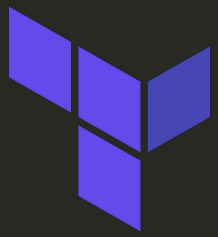
AWS Web Console



Infrastructure as Code



Infrastructure as Code



HashiCorp
Terraform

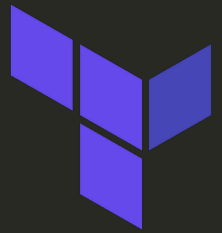
(Open Source)

AWS
Cloud
Formation

GCP
Cloud
Deployment
Manager

Azure
Resource
Manager

Terraform

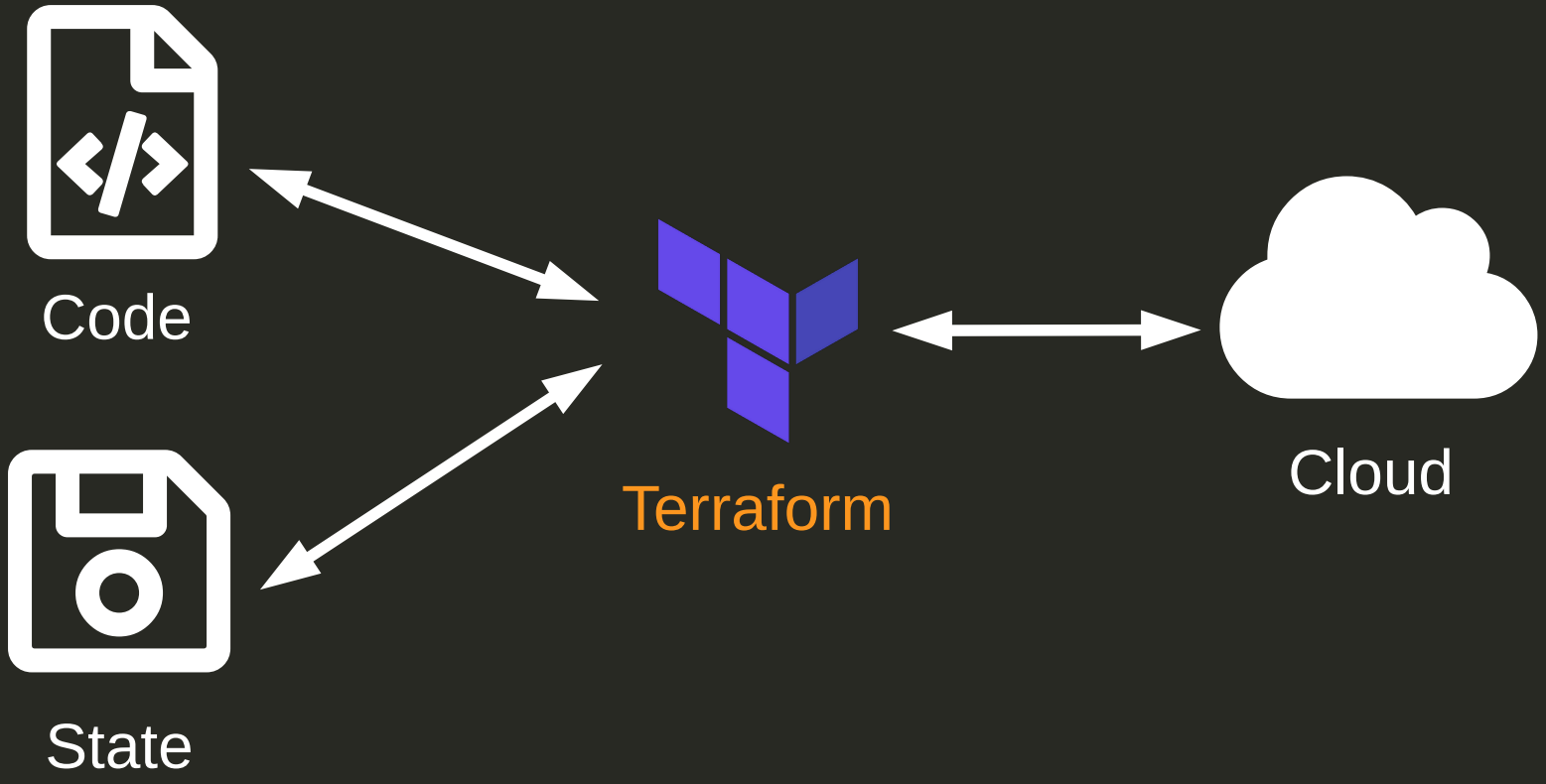


A tool for building, changing, and versioning infrastructure safely and efficiently. Building blocks:

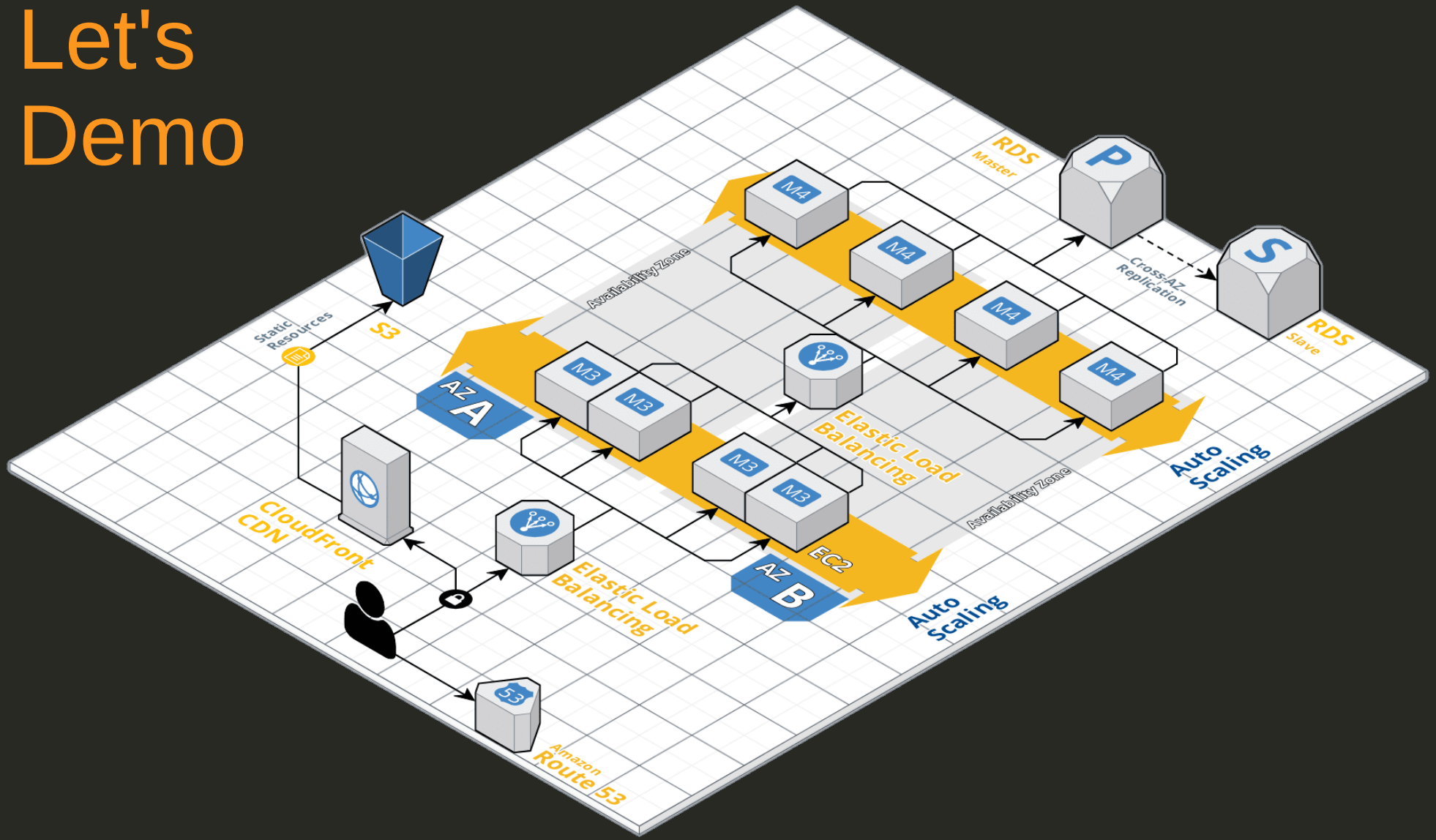
- Providers (AWS, GCP, Azure, MySQL, PostgreSQL, Kubernetes, Helm, GitHub and hundred of others)
- Resources, Data Sources (read-only)
- Input, Local and Output variables
- Expressions and Functions

https://www.terraform.io/

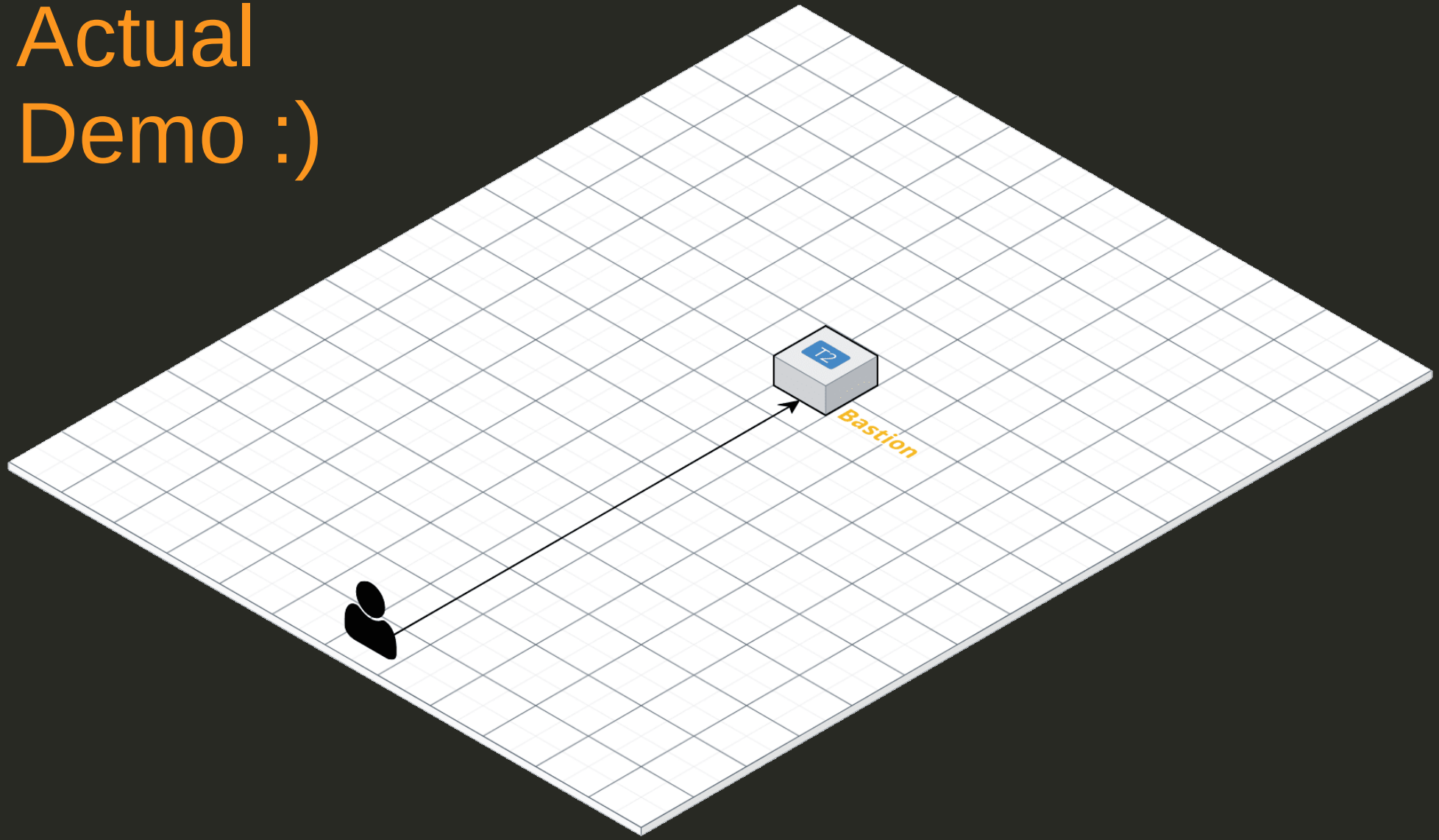
Terraform State



Let's Demo


RDS
Master
P
S
RDS
slave
Cross-AZ
Replication
M4
M4
M4
M4
M3
M3
M3
M3
Availability Zone
Availability Zone
AZ A
AZ B
EC2
Elastic Load
Balancing
Elastic Load
Balancing
Auto
Scaling
Auto
Scaling
Static
Resources
S3
CloudFront
CDN
53
Amazon
Route 53

Actual Demo :)



Terraform Meta Arguments

- depends_on - for specifying hidden dependencies
- count - for creating multiple resource instances according to a count
- for_each - to create multiple instances according to a map - or set of strings
- provider - for selecting a non-default provider configuration
- lifecycle - for lifecycle customizations
- provisioner and connection - for taking extra actions after resource creation

depends_on

```
resource "aws_instance" "bastion" {
  vpc_security_group_ids = [
    aws_security_group.bastion.id,
  ]
  depends_on = [
    aws_instance.web,
  ]
}
```

count

```
resource "aws_instance" "bastion" {
  count = 10
}
```

count

```
variable "web_enabled" {
  type    = bool
  default = false
}
resource "aws_instance" "web" {
  count = var.web_enabled == true ? 1 : 0
}
```

for_each

```
variable "users" {
  type    = list(string)
  default = ["admin", "developer", "manager"]
}
resource "aws_iam_user" "user" {
  for_each = toset(var.users)
  name     = each.key
}
```

provider

```
provider "aws" {}
provider "aws" {
  alias = "staging"
}
resource "aws_instance" "web" {
  provider = aws.staging
}
```

lifecycle

```
resource "aws_instance" "web" {
  lifecycle {
    create_before_destroy = true
    ignore_changes        = [tags]
  }
}
```

lifecycle

```
resource "aws_instance" "web" {
  lifecycle {
    prevent_destroy = true
  }
}
```

provisioner

```
resource "null_resource" "id_rsa" {
  provisioner "local-exec" {
    working_dir = path.module
    command     = "ssh-keygen -N '' -f id_rsa"
  }
}
```

provisioner

```
resource "aws_instance" "web" {
  provisioner "remote-exec" {
    inline = [
      "sudo systemctl disable httpd",
    ]
  }
}
```

provisioner

```
resource "aws_instance" "web" {
  provisioner "file" {
    source      = "${path.module}conf/httpd.conf"
    destination = "/etc/httpd/conf/httpd.conf"
  }
}
```

provisioner

```
resource "aws_instance" "web" {
  provisioner "file" {
    ...
    connection {
      type = "ssh"
      user = "developer"
      port = 2022
    }
  }
}
```

Terraform Modules

```
module "rds" {
  source  = "terraform-aws-modules/rds/aws"
  version = "2.5.0"
  # insert the 11 required variables here
}
```

https://registry.terraform.io/

Terragrunt

A thin wrapper for Terraform that provides extra tools for working with multiple Terraform modules. For example:

- Creates remote state and locking resources automatically
- Passes extra CLI arguments every time you run certain terraform commands

https://github.com/gruntwork-io/terragrunt

Questions?

Thank you!

https://a.openalt.cz/53

aws-vault

A tool to securely store and access AWS credentials in (development) environments.

- Encrypts AWS keys
- Provides temporary, one time credentials

https://github.com/99designs/aws-vault

Terraform Pre-Commit Framework

Automatically, before each commit:

- Formats Terraform code
- Updates README.md with the description of:
  - terraform input variables
  - terraform output variables

https://github.com/antonbabenko/pre-commit-terraform