

Terraform AWS modules workshop

Anton Babenko
@antonbabenko

April 2019

Anton Babenko

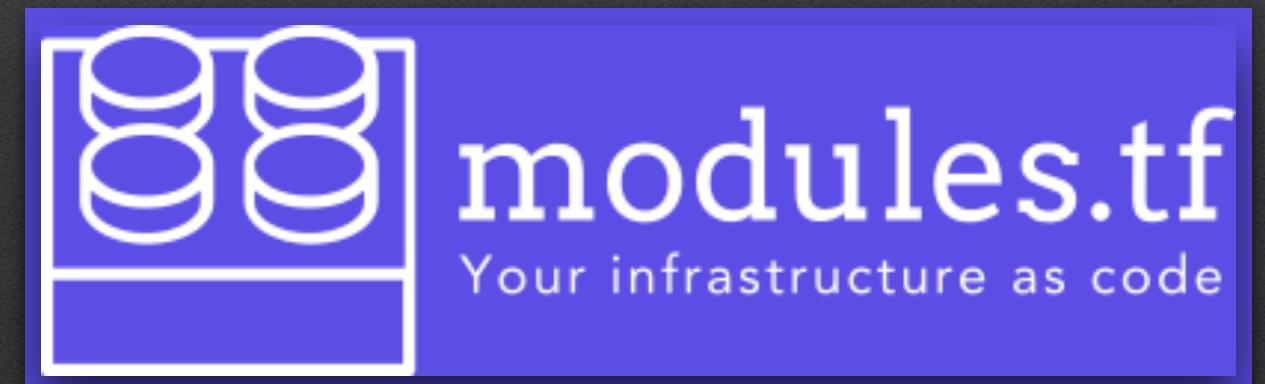
Terraform AWS fanatic since 2015

Organiser of HashiCorp UG, AWS UG, DevOps Norway, DevOpsDays Oslo



I ❤️ open-source:

- [terraform-community-modules + terraform-aws-modules](#)
- [antonbabenko/pre-commit-terraform – clean code and documentation](#)
- [antonbabenko/terraform-docs-as-pdf](#)
- [antonbabenko/modules.tf-lambda – generate Terraform code from visual diagrams](#)
- [www.terraform-best-practices.com](#)
- [medium.com/@anton.babenko](#)
- [@antonbabenko – Twitter, GitHub, LinkedIn](#)





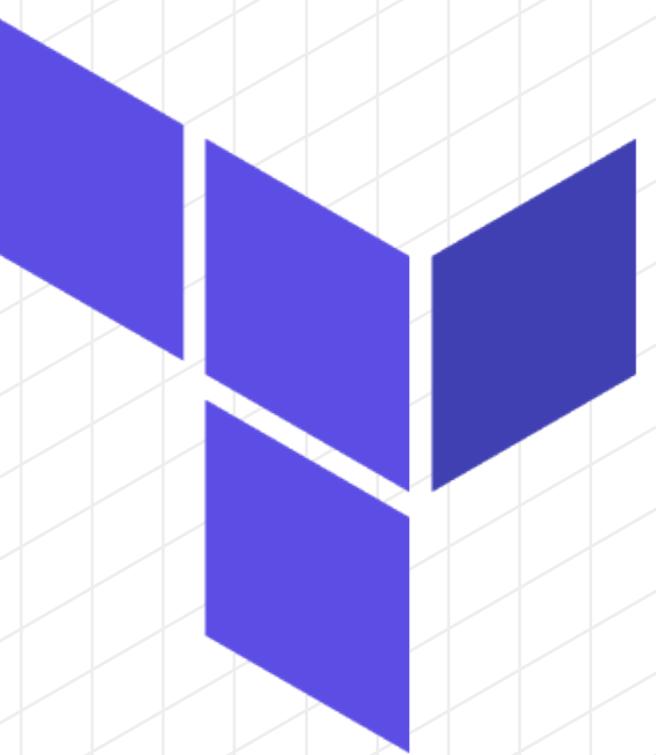
Collection of open-source Terraform AWS modules supported by the community.

More than 2 mil. downloads since September 2017.

(VPC, Autoscaling, RDS, Security Groups, ELB, ALB, Redshift, SNS, SQS, IAM, EKS, ECS...)

github.com/terraform-aws-modules

registry.terraform.io/modules/terraform-aws-modules



HashiCorp
Terraform

Write, plan and manage infrastructure as code

www.terraform.io

```
1 variable "aws_region" {
2   description = "Region where resources should be created"
3   default     = "eu-west-1"
4 }
5
6 provider "aws" {
7   region = "${var.aws_region}"
8 }
9
10 resource "aws_s3_bucket" "this" {
11   bucket = "my-bucket-${random_pet.bucket.id}"
12 }
13
14 resource "random_pet" "bucket" {
15   keepers = {
16     aws_region = "${var.aws_region}"
17   }
18
19   length = 1
20 }
21
22 output "this_s3_bucket_id" {
23   description = "ID of S3 bucket"
24   value       = "${aws_s3_bucket.this.id}"
25 }
```

```
1 variable "aws_region" {
2   description = "Region where resources should be created"
3   default     = "eu-west-1"
4 }
5
6 provider "aws" {
7   region = "${var.aws_region}"
8 }
9
10 resource "aws_s3_bucket" "this" {
11   bucket = "my-bucket-${random_pet.bucket.id}"
12 }
13
14 resource "random_pet" "bucket" {
15   keepers = {
16     aws_region = "${var.aws_region}"
17   }
18
19   length = 1
20 }
21
22 output "this_s3_bucket_id" {
23   description = "ID of S3 bucket"
24   value       = "${aws_s3_bucket.this.id}"
25 }
```

```
1 variable "aws_region" {
2   description = "Region where resources should be created"
3   default     = "eu-west-1"
4 }
5
6 provider "aws" {
7   region = "${var.aws_region}"
8 }
9
10 resource "aws_s3_bucket" "this" {
11   bucket = "my-bucket-${random_pet.bucket.id}"
12 }
13
14 resource "random_pet" "bucket" {
15   keepers = {
16     aws_region = "${var.aws_region}"
17   }
18
19   length = 1
20 }
21
22 output "this_s3_bucket_id" {
23   description = "ID of S3 bucket"
24   value       = "${aws_s3_bucket.this.id}"
25 }
```

```
$ terraform init
```

Initializing provider plugins...

- Checking for available provider plugins on <https://releases.hashicorp.com>...
- Downloading plugin for provider "aws" (1.10.0)...
- Downloading plugin for provider "random" (1.1.0)...

Terraform has been successfully initialized!

```
$ terraform apply
```

An execution plan has been generated and is shown below.
Resource actions are indicated with the following symbols:

+ create

Terraform will perform the following actions:

+ aws_s3_bucket.this

 id: <computed>

 acl: "private"

 bucket: "my-bucket-\${random_pet.bucket.id}"

+ random_pet.bucket

 id: <computed>

 keepers.%: "1"

 keepers.aws_region: "eu-west-1"

 length: "1"

Plan: 2 to add, 0 to change, 0 to destroy.

Do you want to perform these actions?

Terraform will perform the actions described above.

Only 'yes' will be accepted to approve.

Do you want to perform these actions?
Terraform will perform the actions described above.
Only 'yes' will be accepted to approve.

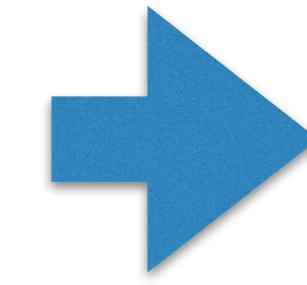
Enter a value: yes

```
random_pet.bucket: Creating...
  keepers.%:      "" => "1"
  keepers.aws_region: "" => "eu-west-1"
  length:        "" => "1"
random_pet.bucket: Creation complete after 0s (ID: seasnail)
aws_s3_bucket.this: Creating...
  acl:           "" => "private"
  arn:           "" => "<computed>"
  bucket:        "" => "my-bucket-seasnail"
aws_s3_bucket.this: Creation complete after 6s (ID: my-bucket-seasnail)
```

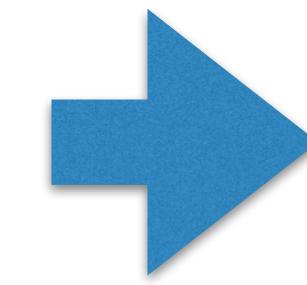
Apply complete! Resources: 2 added, 0 changed, 0 destroyed.

Outputs:

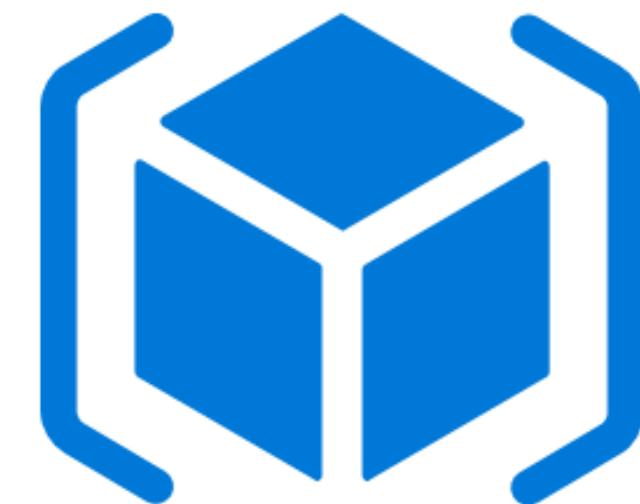
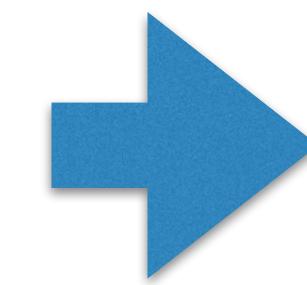
```
this_s3_bucket_id = my-bucket-seasnail
```



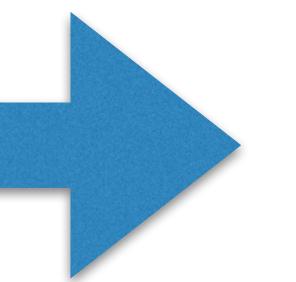
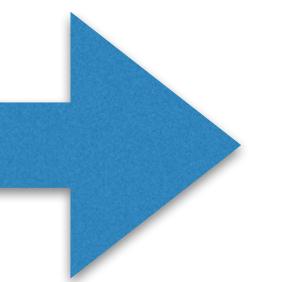
AWS
CloudFormation



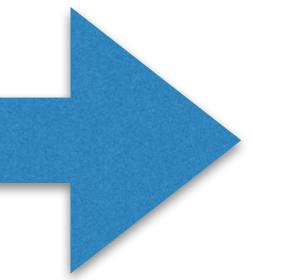
Google Cloud
Deployment Manager



Azure Resource
Manager

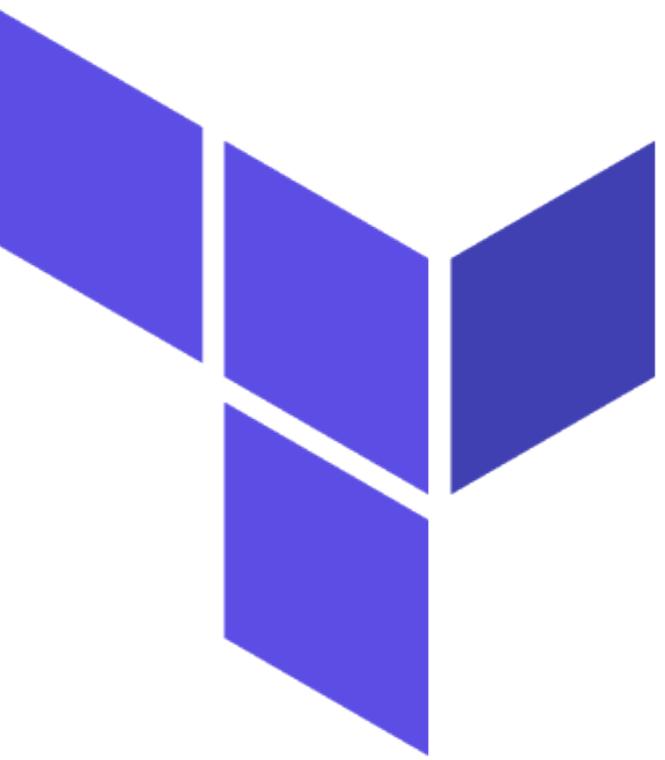
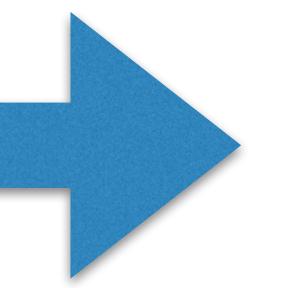
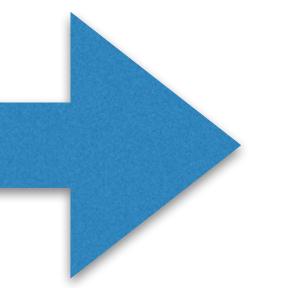
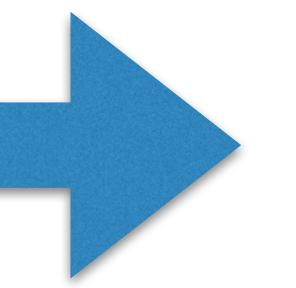


HashiCorp
Terraform





+ more than 100 providers



HashiCorp
Terraform

Why Terraform and not AWS CloudFormation, Azure ARM, Google Cloud Deployment Manager?

- Terraform manages 100+ providers, has easier syntax (HCL), has native support for modules and remote states, has teamwork related features, is an open-source project.
- Provides a high-level abstraction of infrastructure (IaC)
- Allows for composition and combination
- Supports parallel management of resources (graph, fast)
- Separates planning from execution (dry-run)

Terraform – universal tool for everything with an API

- GSuite
- Dropbox files and access
- New Relic metrics
- Datadog users and metrics
- Jira issues
- Minecraft, or even order Domino's pizza
- All Terraform providers

Terraform modules

Modules in Terraform are self-contained packages of Terraform configurations that are managed as a group.

Resource modules

- Create resources in a very flexible configuration
- Open-source

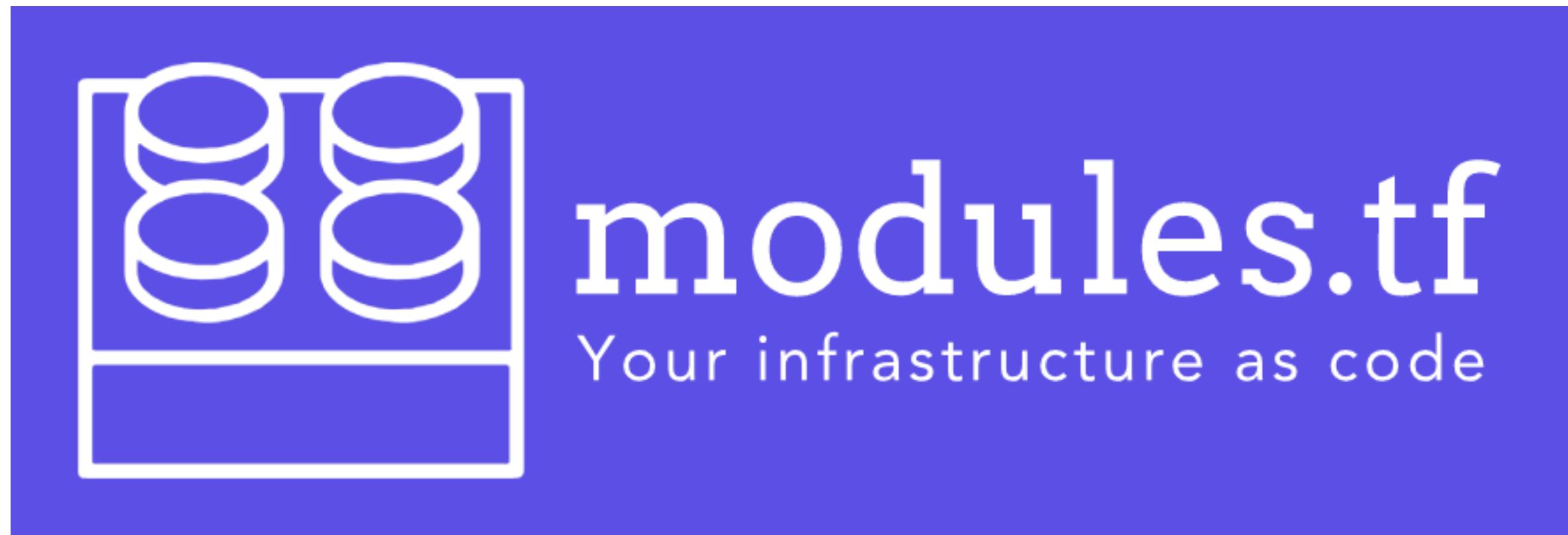
Infrastructure modules

- Consist of resource modules
- Enforce tags and company standards
- Use preprocessors, jsonnet, cookiecutter

Let's start!

- ▀ <https://github.com/antonbabenko/terraform-best-practices-workshop>
- ▀ `git clone git@github.com:antonbabenko/terraform-best-practices-workshop.git`
- ▀ Read "Attendee's checklist" in README.md
- ▀ Follow the agenda in README.md

Bonus





DESIGN LIVE BUDGET



AWS Components

Common

BLOCK

TEXT LABEL

ICON

IMAGE

AREA

Compute

AUTO SCALING

Auto scale EC2 group capacity

EC2

Elastic virtual server

LAMBDA

Run code in response to events

ZONE

Availability zone

Storage

EBS

EC2 block storage volume

S3

Simple object storage

EFS

Elastic file system

CLOUDFRONT

Content delivery network

Networking

ELB

Elastic load balancer

ROUTE 53

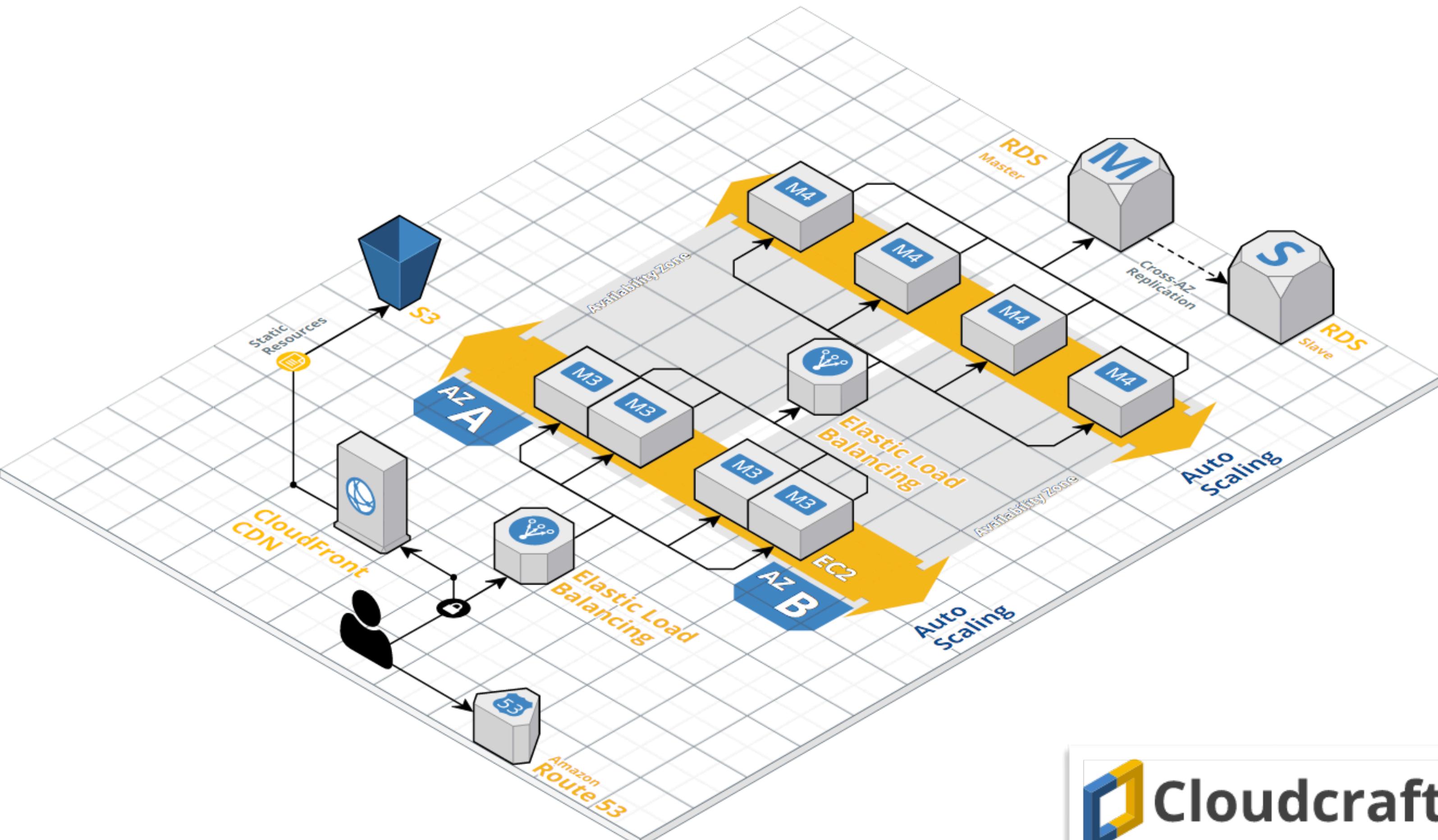
Managed DNS service

VPC GATEWAY

Access instances in your VPC



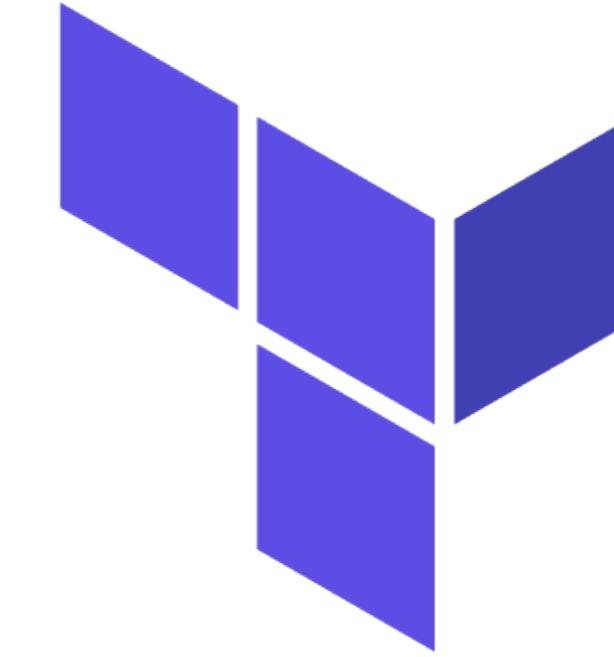
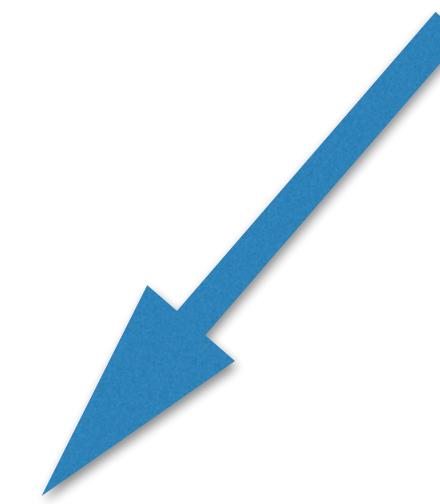
EXPORT



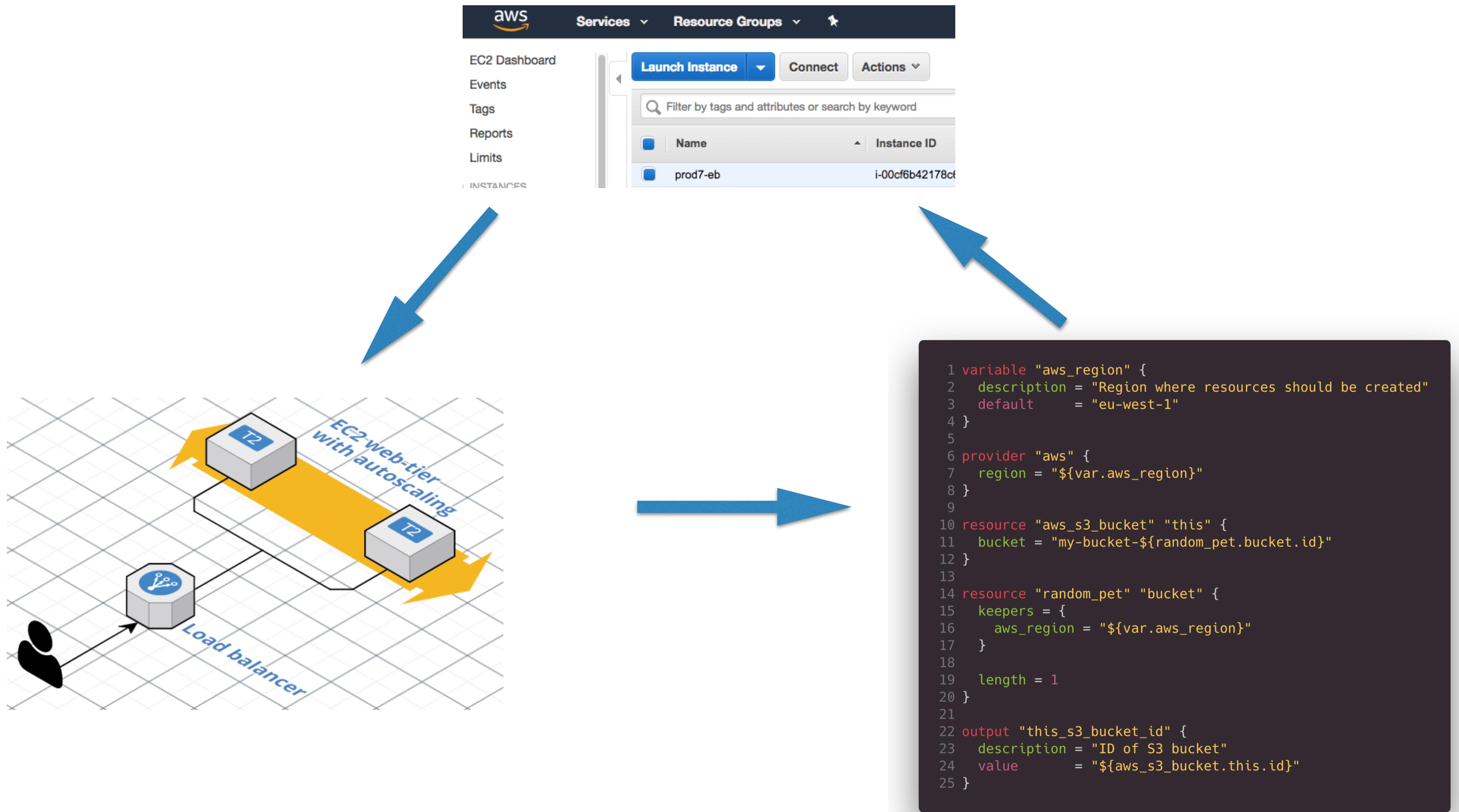
Drag to create multi-selection

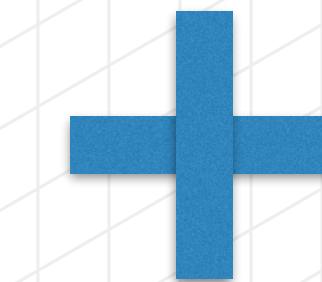
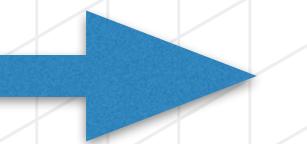


Cloudcraft

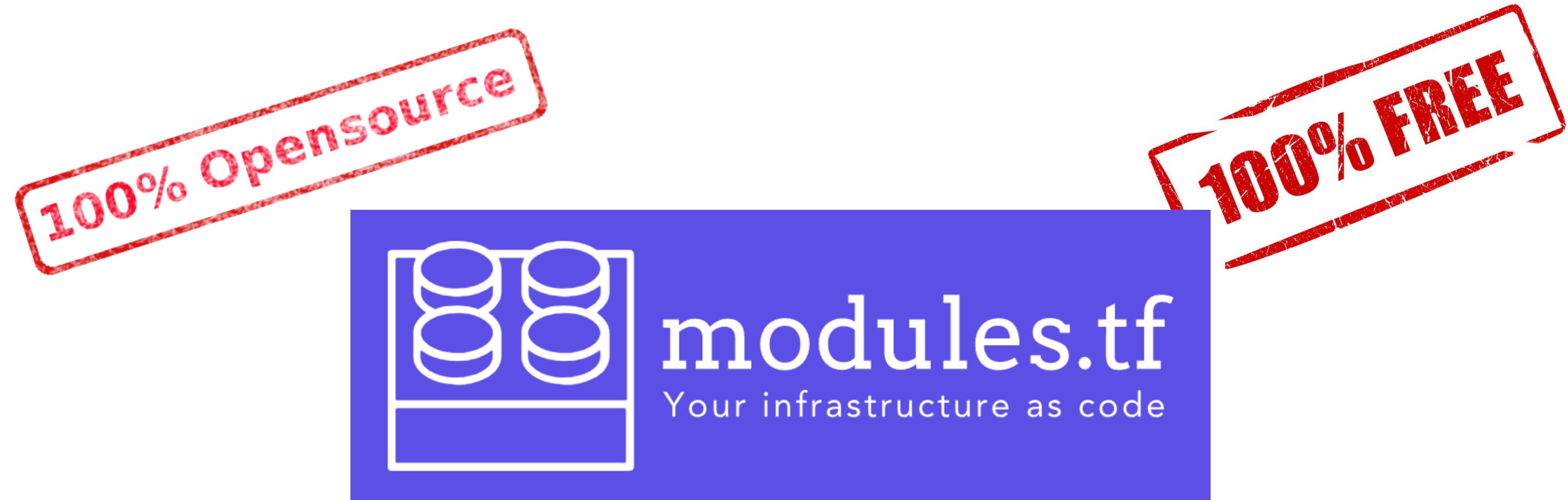


HashiCorp
Terraform





- ✓ [cloudfactory.co](https://cloudfactory.com) – design, plan and visualize
- ✓ [terraform-aws-modules](https://github.com/terraform-aws-modules) – building blocks of AWS infrastructure
- ✓ [Terraform](https://www.terraform.io) – infrastructure as code

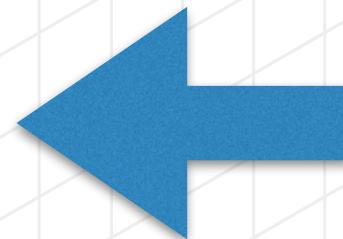
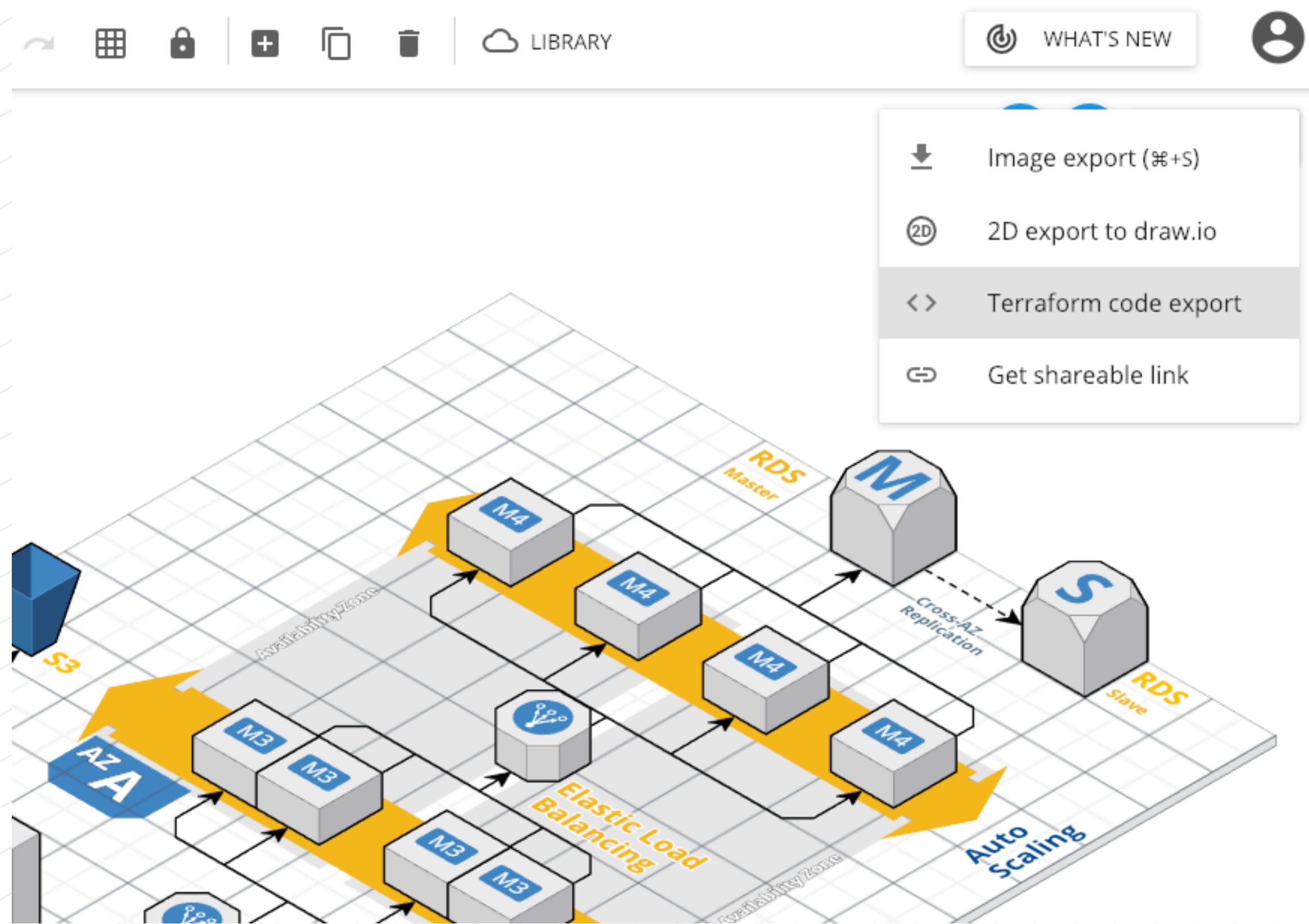


Infrastructure as code generator – from visual diagrams to Terraform

<https://github.com/antonbabenko/modules.tf-lambda>

Demo video: <https://www.youtube.com/watch?v=F1Ax1zfZbiY>

Try it yourself!



1. Go to cloudcraft.co
2. Sign up, sign in (free account)
3. Draw your AWS infrastructure
4. Click "Export"
5. Click "Terraform code export"

modules.tf – generated code

- ✓ Potentially ready-to-use Terraform configurations
- ✓ Suits best for bootstrapping
- ✓ Enforces Terraform best-practices
- ✓ Batteries included (`terraform-aws-modules`, `terragrunt`, `pre-commit`)
- ✓ 100% free and open-source (<https://github.com/antonbabenko/modules.tf-lambda>)
- ✓ Released under MIT license

Thanks!

Questions?

github.com/antonbabenko

twitter.com/antonbabenko