

Terraform AWS modules workshop

Anton Babenko
@antonbabenko

January 2020

Anton Babenko

AWS Community Hero / Terraform fanatic since 2015

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

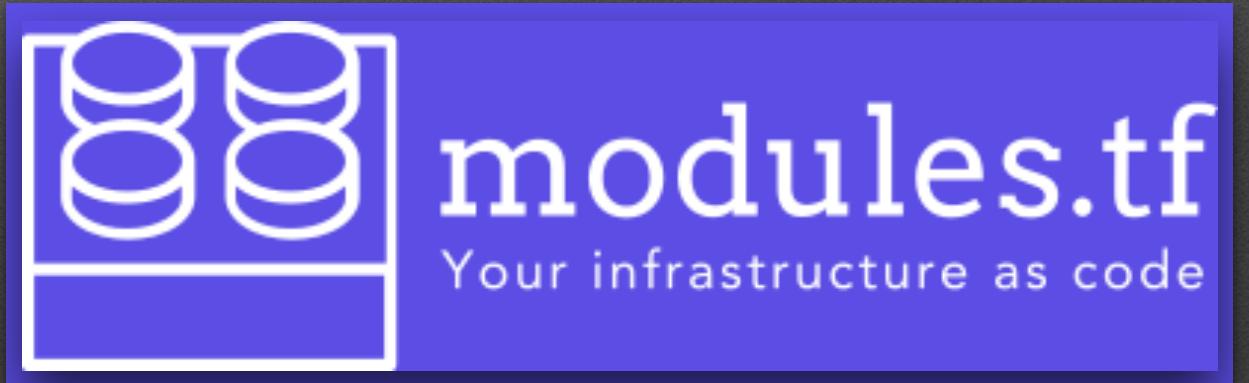
I ❤️ open-source:

- Y [terraform-community-modules + terraform-aws-modules](#)
- Y [antonbabenko/pre-commit-terraform](#) – clean code and documentation
- Y [antonbabenko/modules.tf-lambda](#) – generate Terraform code from visual diagrams
- Y [antonbabenko/terragrunt-reference-architecture](#) – Terragrunt reference architecture
- Y [www.terraform-best-practices.com](#)
- Y [@antonbabenko](#) – Twitter, GitHub, LinkedIn



What do I do?

- Y All-things Terraform + AWS + DevOps
- Y Consulting
- Y Workshops
- Y Trainings
- Y Mentorship



Company: betajob.com

Blog: antonbabenko.com

Interview: <https://www.antonbabenko.com/my-terraform-aws-journey-june-2019/>



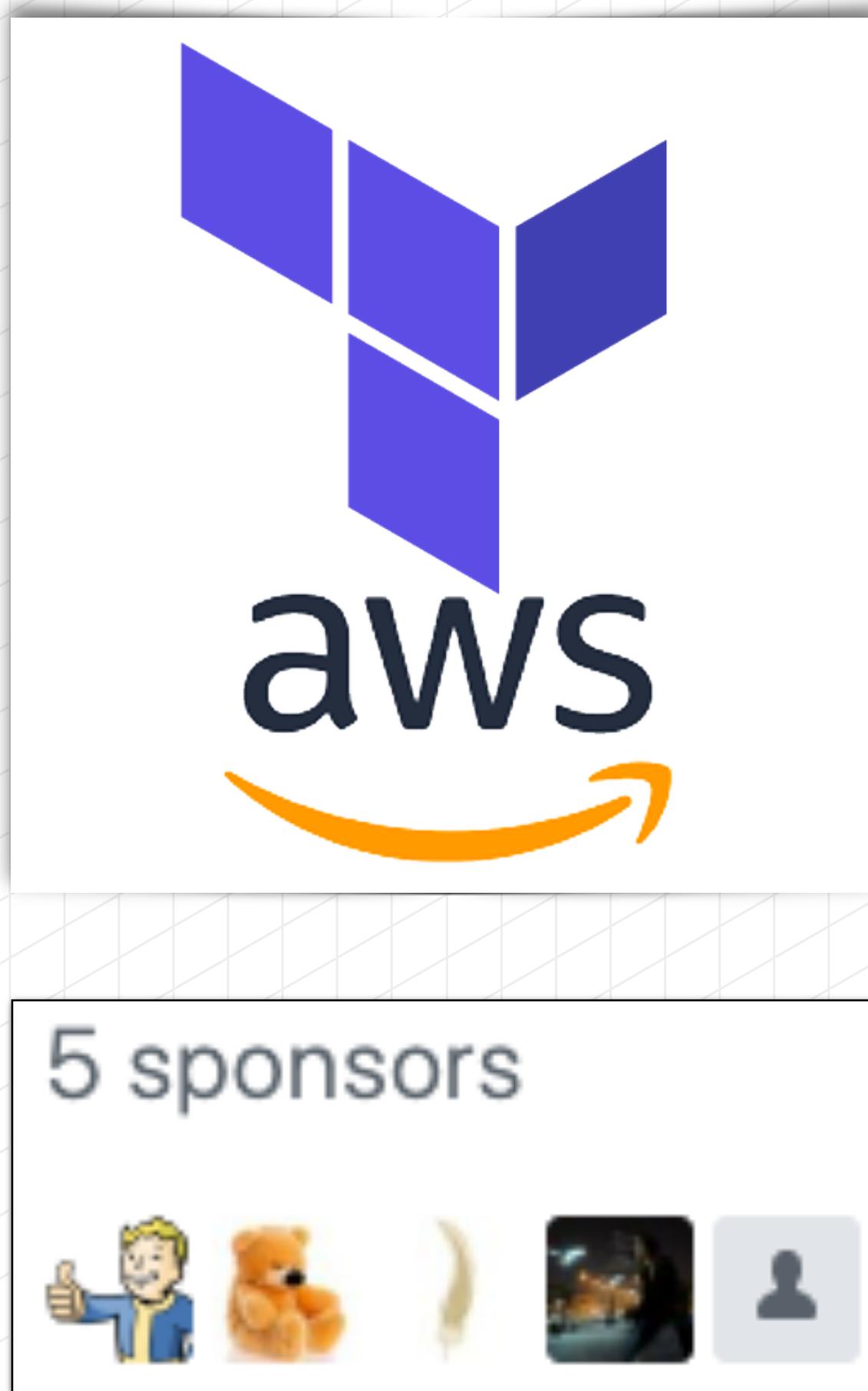
Collection of open-source Terraform AWS modules supported by the community.

More than 7 mil. downloads since September 2017.

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

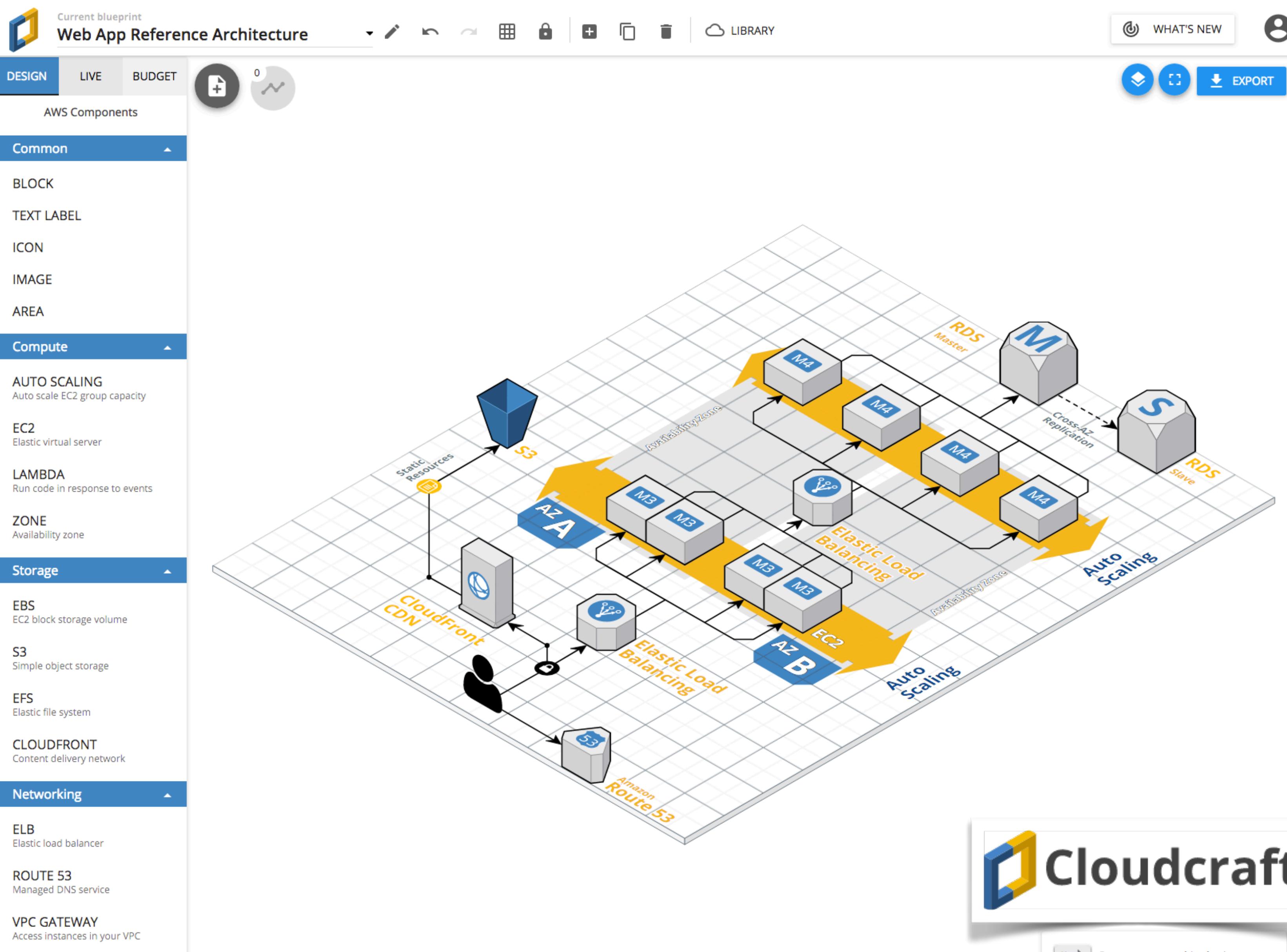
github.com/terraform-aws-modules

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



7+ million provisions, 1000+ pull-requests and issues resolved...

Become a sponsor – github.com/sponsors/antonbabenko



Cloudcraft.co – the best way to draw AWS diagrams

@antonbabenko

cloudcraft.co features

- Manage components in browser (EC2 instances, autoscaling groups, RDS, etc)
- Connect components
- Import live AWS infrastructure
- Calculate the budget
- Share link to a blueprint
- Export as image
- Embed drawing to wiki, Confluence, etc

Agenda

- ▀ Intro into AWS, IaC and Terraform
- ▀ Terraform modules
- ▀ Terraform 0.12
- ▀ Building AWS infrastructure using `terraform-aws-modules`
- ▀ Q&A



Amazon Web Services (**AWS**) is a secure cloud services platform, offering compute power, database storage, content delivery and other functionality to help businesses scale and grow.

<https://aws.amazon.com/>



AWS

Services

Edit

Amazon Web Services

Compute

- EC2**
Virtual Servers in the Cloud
- EC2 Container Service**
Run and Manage Docker Containers
- Elastic Beanstalk**
Run and Manage Web Apps
- Lambda**
Run Code in Response to Events

Storage & Content Delivery

- S3**
Scalable Storage in the Cloud
- CloudFront**
Global Content Delivery Network
- Elastic File System**
Fully Managed File System for EC2
- Glacier**
Archive Storage in the Cloud
- Snowball**
Large Scale Data Transport
- Storage Gateway**
Hybrid Storage Integration

Database

- RDS**
Managed Relational Database Service
- DynamoDB**
Managed NoSQL Database
- ElastiCache**
In-Memory Cache
- Redshift**
Fast, Simple, Cost-Effective Data Warehousing
- DMS**
Managed Database Migration Service

Networking

- VPC**
Isolated Cloud Resources
- Direct Connect**
Dedicated Network Connection to AWS
- Route 53**
Scalable DNS and Domain Name Registration

Developer Tools

- CodeCommit**
Store Code in Private Git Repositories
- CodeDeploy**
Automate Code Deployments
- CodePipeline**
Release Software using Continuous Delivery

Management Tools

- CloudWatch**
Monitor Resources and Applications
- CloudFormation**
Create and Manage Resources with Templates
- CloudTrail**
Track User Activity and API Usage
- Config**
Track Resource Inventory and Changes
- OpsWorks**
Automate Operations with Chef
- Service Catalog**
Create and Use Standardized Products
- Trusted Advisor**
Optimize Performance and Security

Security & Identity

- Identity & Access Management**
Manage User Access and Encryption Keys
- Directory Service**
Host and Manage Active Directory
- Inspector**
Analyze Application Security
- WAF**
Filter Malicious Web Traffic
- Certificate Manager**
Provision, Manage, and Deploy SSL/TLS Certificates

Analytics

- EMR**
Managed Hadoop Framework
- Data Pipeline**
Orchestration for Data-Driven Workflows
- Elasticsearch Service**
Run and Scale Elasticsearch Clusters
- Kinesis**
Work with Real-Time Streaming Data
- Machine Learning**
Build Smart Applications Quickly and Easily

Internet of Things

- AWS IoT**
Connect Devices to the Cloud

Game Development

- GameLift**
Deploy and Scale Session-based Multiplayer Games

Mobile Services

- Mobile Hub**
Build, Test, and Monitor Mobile Apps
- Cognito**
User Identity and App Data Synchronization
- Device Farm**
Test Android, iOS, and Web Apps on Real Devices in the Cloud
- Mobile Analytics**
Collect, View and Export App Analytics
- SNS**
Push Notification Service

Application Services

- API Gateway**
Build, Deploy and Manage APIs
- AppStream**
Low Latency Application Streaming
- CloudSearch**
Managed Search Service
- Elastic Transcoder**
Easy-to-Use Scalable Media Transcoding
- SES**
Email Sending and Receiving Service
- SQS**
Message Queue Service
- SWF**
Workflow Service for Coordinating Application Components

Enterprise Applications

- WorkSpaces**
Desktops in the Cloud
- WorkDocs**
Secure Enterprise Storage and Sharing Service
- WorkMail**
Secure Email and Calendaring Service



AWS

Services

Edit

Amazon Web Services

Compute

- EC2 Virtual Servers in the Cloud
- EC2 Container Service Run and Manage Docker Containers
- Elastic Beanstalk Run and Manage Web Apps
- Lambda Run Code in Response to Events

Storage & Content Delivery

- S3 Scalable Storage in the Cloud
- CloudFront Global Content Delivery Network
- Elastic File System Fully Managed File System for EC2
- Glacier Archive Storage in the Cloud
- Snowball Large Scale Data Transport
- Storage Gateway Hybrid Storage Integration

Database

- RDS Managed Relational Database Service
- DynamoDB Managed NoSQL Database
- ElastiCache In-Memory Cache
- Redshift Fast, Simple, Cost-Effective Data Warehousing
- DMS Managed Database Migration Service

Networking

- VPC Isolated Cloud Resources
- Direct Connect Dedicated Network Connection to AWS
- Route 53 Scalable DNS and Domain Name Registration

Developer Tools

- CodeCommit Store Code in Private Git Repositories
- CodeDeploy Automate Code Deployments
- CodePipeline Release Software using Continuous Delivery

Management Tools

- CloudWatch Monitor Resources and Applications
- CloudFormation Create and Manage Resources with Templates
- CloudTrail Track User Activity and API Usage
- Config Track Resource Inventory and Changes
- OpsWorks Automate Operations with Chef
- Service Catalog

Internet of Things

- AWS IoT Connect Devices to the Cloud

Game Development

- GameLift Deploy and Scale Session-based Multiplayer Games

Mobile Services

- Mobile Hub Build, Test, and Monitor Mobile Apps
- Cognito User Identity and App Data Synchronization
- Device Farm Test Android, iOS, and Web Apps on Real Devices in the Cloud
- Mobile Analytics Collect, View and Export App Analytics
- SNS Push Notification Service

Try this – <http://bit.ly/aws-icon-quiz>

Security & Identity

- Identity & Access Management Manage User Access and Encryption Keys
- Directory Service Host and Manage Active Directory
- Inspector Analyze Application Security
- WAF Filter Malicious Web Traffic
- Certificate Manager Provision, Manage, and Deploy SSL/TLS Certificates

Compute

- AppStream Low Latency Application Streaming
- CloudSearch Managed Search Service
- Elastic Transcoder Easy-to-Use Scalable Media Transcoding
- SES Email Sending and Receiving Service
- SQS Message Queue Service
- SWF Workflow Service for Coordinating Application Components

Analytics

- EMR Managed Hadoop Framework
- Data Pipeline Orchestration for Data-Driven Workflows
- Elasticsearch Service Run and Scale Elasticsearch Clusters
- Kinesis Work with Real-Time Streaming Data
- Machine Learning Build Smart Applications Quickly and Easily

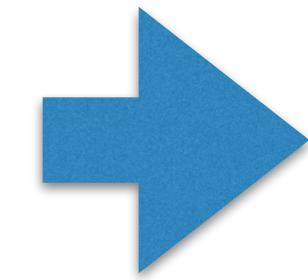
Enterprise Applications

- WorkSpaces Desktops in the Cloud
- WorkDocs Secure Enterprise Storage and Sharing Service
- WorkMail Secure Email and Calendaring Service

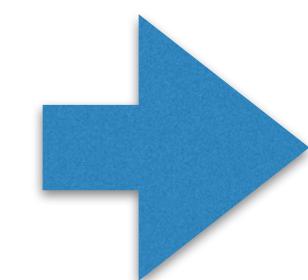
@antonbabenko

“Infrastructure is all what is needed to support the computing requirements of an application.”

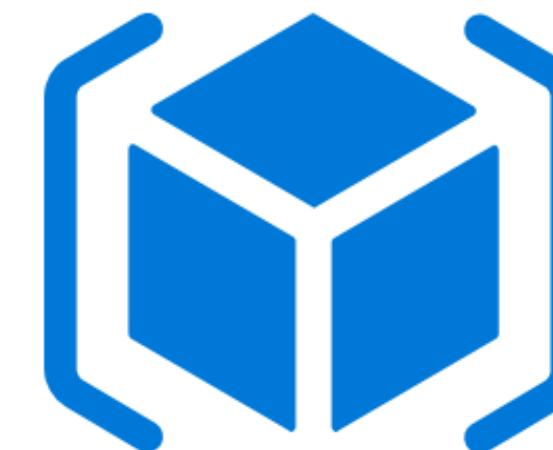
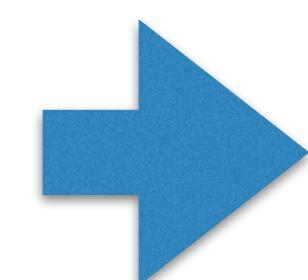
– What is infrastructure?



AWS
CloudFormation

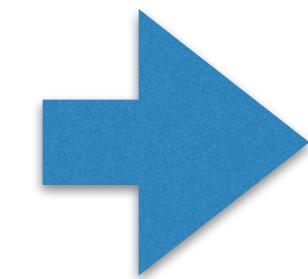


Google Cloud
Deployment Manager



Azure Resource
Manager

Configuration
Management Tools



@antonbabenko

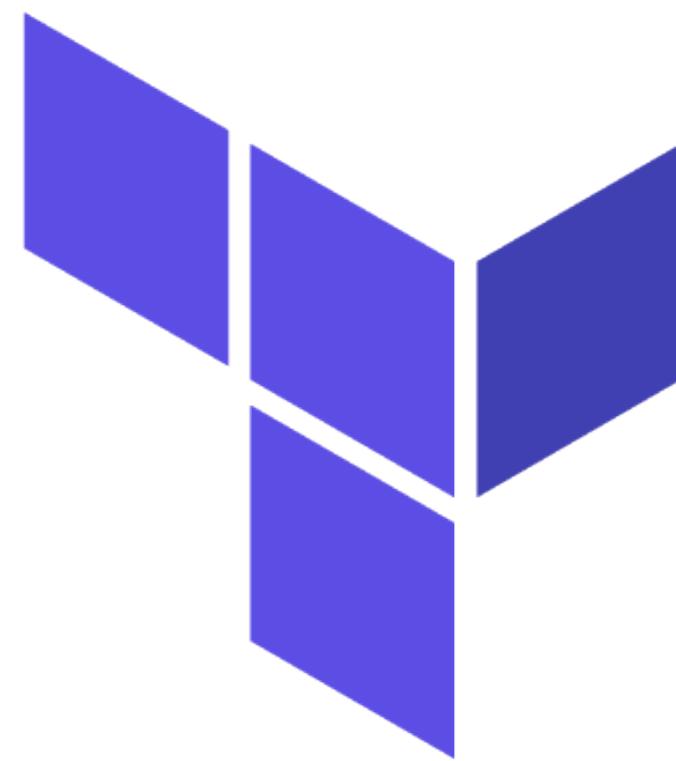


Google Cloud

Azure

Configuration
Management Tools

+ more than 250 providers



HashiCorp
Terraform

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

- Terraform supports 250+ 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, unifies the view of resources
- Supports the modern datacenter (IaaS, PaaS, SaaS)
- Separates planning from execution (dry-run)
- Provides a workflow which is technology agnostic
- Manages anything with an API

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 history

- Started in 2014 as a tool to "write, plan and manage infrastructure as code"
- Homepage – <http://terraform.io>
- Open-source – <https://github.com/hashicorp/terraform>
- Version 0.11 – November 2017:
 - Stable (soon to be deprecated)
 - HCL 1 (verbose syntax)
- Version 0.12 – May 2019:
 - Stable (we will focus on this)
 - HCL 2, loops, dynamic blocks, and much more...



Infrastructure as code makes DevOps possible



Key benefits:

- Y Treat infrastructure like application code
- Y Always know what changed
- Y Validate infrastructure before deployment
- Y Anyone can build an environment anytime



Terraform 0.12

```
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 the backend...

Initializing provider plugins...
- Checking for available provider plugins...
- Downloading plugin for provider "aws" (hashicorp/aws) 2.26.0...
- Downloading plugin for provider "random" (hashicorp/random) 2.2.0...

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 will be created
+ resource "aws_s3_bucket" "this" {
    + acl          = "private"
    + bucket       = (known after apply)
    + id          = (known after apply)
}

# random_pet.bucket will be created
+ resource "random_pet" "bucket" {
    + id          = (known after apply)
    + keepers     = {
        + "aws_region" = "eu-west-1"
    }
    + length      = 1
    + separator   = "-"
}
```

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

@antonbabenko

```
# random_pet.bucket will be created
+ resource "random_pet" "bucket" {
    + id          = (known after apply)
    + keepers     = {
        + "aws_region" = "eu-west-1"
    }
    + length      = 1
    + separator   = "-"
}
```

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.

Enter a value: yes

```
random_pet.bucket: Creating...
```

```
random_pet.bucket: Creation complete after 0s [id=boar]
```

```
aws_s3_bucket.this: Creating...
```

```
aws_s3_bucket.this: Creation complete after 4s [id=my-bucket-boar]
```

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

Outputs:

```
this_s3_bucket_id = my-bucket-boar
```

Resources about Terraform

 <https://www.terraform.io>

 <http://learn.hashicorp.com>

 <https://www.terraform-best-practices.com>

 + ask me for more

Terraform modules

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

– What is Terraform module?

Resource modules

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

Resource modules

```
1 module "atlantis_alb_sg" {  
2   source  = "terraform-aws-modules/security-group/aws//modules/https-443"  
3   version = "v2.0.0"  
4  
5   name      = "atlantis-alb"  
6   vpc_id    = "vpc-12345678"  
7   description = "Security group with HTTPS ports open for everybody (IPv4 CIDR)"  
8  
9   ingress_cidr_blocks = ["0.0.0.0/0"]  
10 }
```

Infrastructure modules

- Also known as "curated modules" and "company-wide modules"
- Consist of resource modules
- Enforce tags and company standards
- In 0.11 — ~~use preprocessors, jsonnet, cookiecutter~~
- In 0.12 – may implement complex logic (conditions, loops, nested blocks)

Infrastructure modules

```
1 module "atlantis" {
2   source = "terraform-aws-modules/atlantis/aws"
3
4   name = "atlantis"
5
6   # VPC
7   cidr          = "10.20.0.0/20"
8   azs           = ["eu-west-1a", "eu-west-1b", "eu-west-1c"]
9   private_subnets = ["10.20.1.0/24", "10.20.2.0/24", "10.20.3.0/24"]
10  public_subnets = ["10.20.101.0/24", "10.20.102.0/24", "10.20.103.0/24"]
11
12  # DNS
13  route53_zone_name = "terraform-aws-modules.modules.tf"
14
15  # Atlantis app
16  atlantis_github_user      = "atlantis-bot"
17  atlantis_github_user_token = "examplegithubtoken"
18 }
```

```
1 # terraform-aws-atlantis/main.tf
2
3 module "vpc" {
4   source  = "terraform-aws-modules/vpc/aws"
5   version = "v1.32.0"
6   # ...
7 }
8
9 module "alb" {
10  source  = "terraform-aws-modules/alb/aws"
11  version = "v3.4.0"
12  # ...
13 }
14
15 module "alb_https_sg" {
16  source  = "terraform-aws-modules/security-group/aws//modules/https-443"
17  version = "v2.0.0"
18  # ...
19 }
20
21 module "ecs" {
22  source  = "terraform-aws-modules/ecs/aws"
23  version = "v1.0.0"
24  # ...
25 }
```

Terraform 0.12

...and why it is so important?

Terraform 0.12

- Y HCL2 – simplified syntax
- Y Loops ("for")
- Y Dynamic blocks ("for_each")
- Y Correct conditional operators (... ? ... : ...)
- Y Extended types of variables
- Y Templates in values
- Y Links between resources are supported (`depends_on` everywhere)
- Y Read more – <https://www.hashicorp.com/blog/announcing-terraform-0-1-2-beta>

Who are you?

Terraform developers or users?

Terraform developers

- Create and maintain Terraform modules (curated modules)
- Happily using Terraform 0.12 ❤️ ❤️ ❤️
- Implement company's standards (security, encryption, naming, integrations, tagging)
- Maintain "reference architectures" inside of the company

Terraform users (all of us)

- Use Terraform modules by setting correct values
- Terraform 0.12 + HCL2 = ❤
- Domain experts in certain areas (e.g. backend developers)
- May not have "Terraform" skills in profile on LinkedIn



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

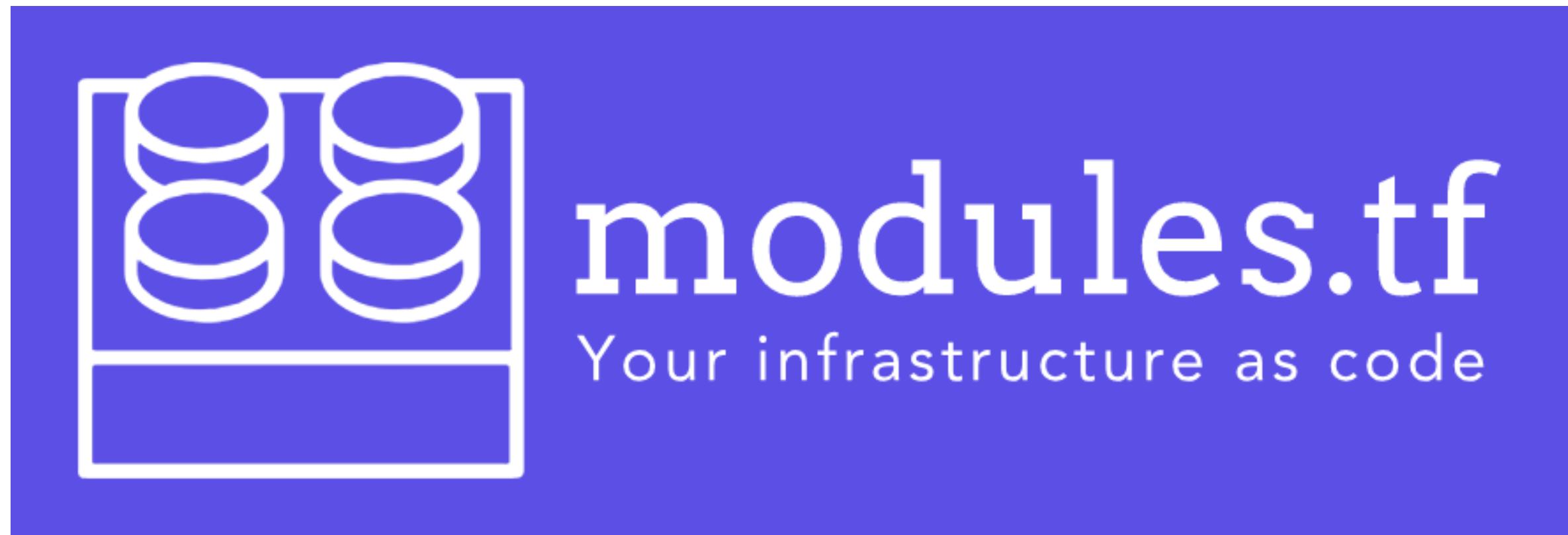
More than 7 mil. downloads since September 2017.

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

github.com/terraform-aws-modules

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

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

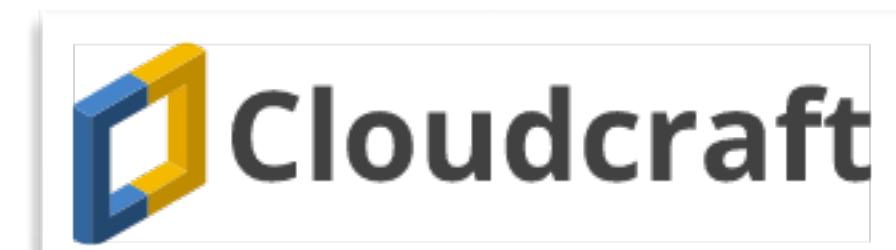
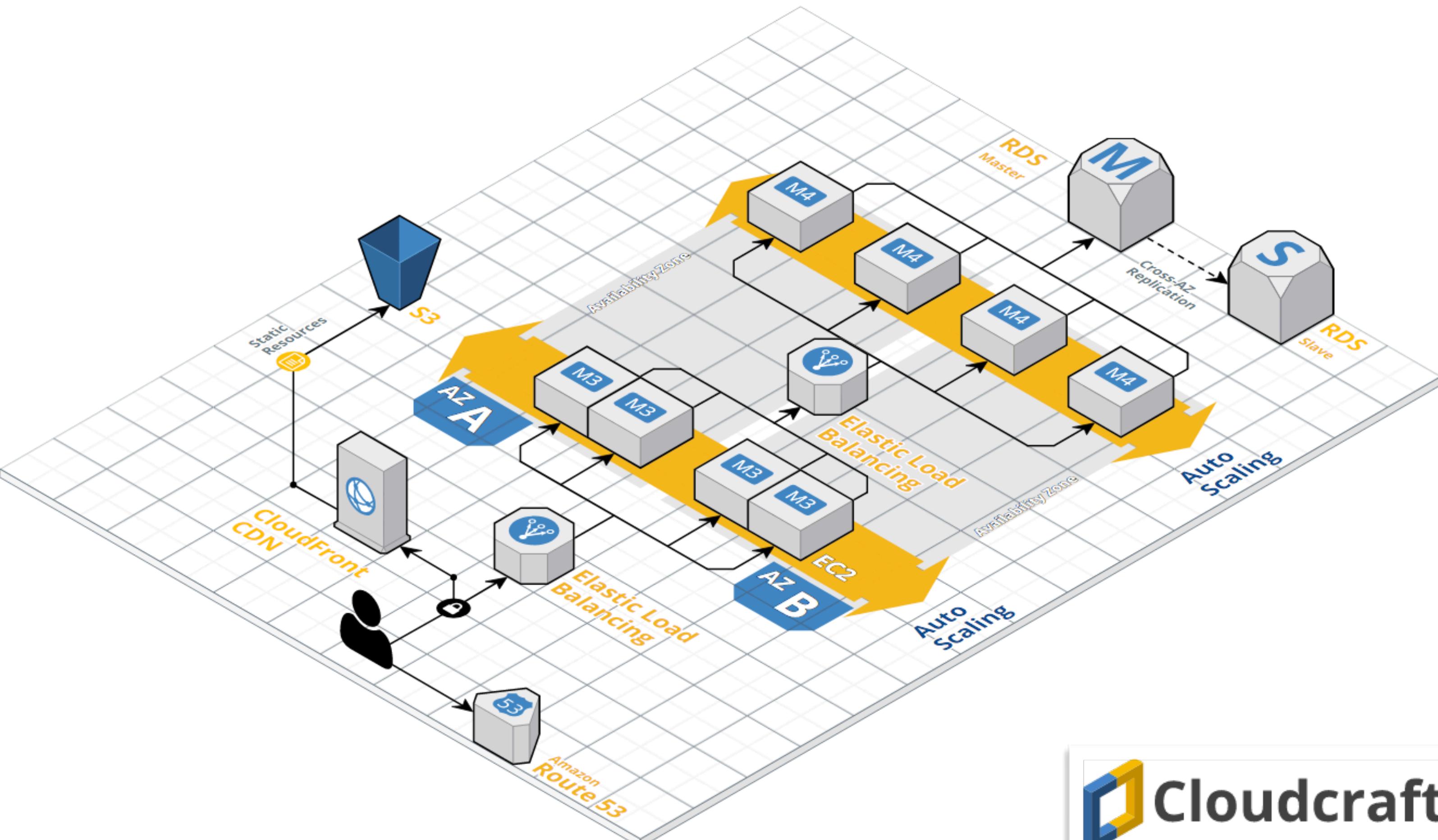
Managed DNS service

VPC GATEWAY

Access instances in your VPC



EXPORT



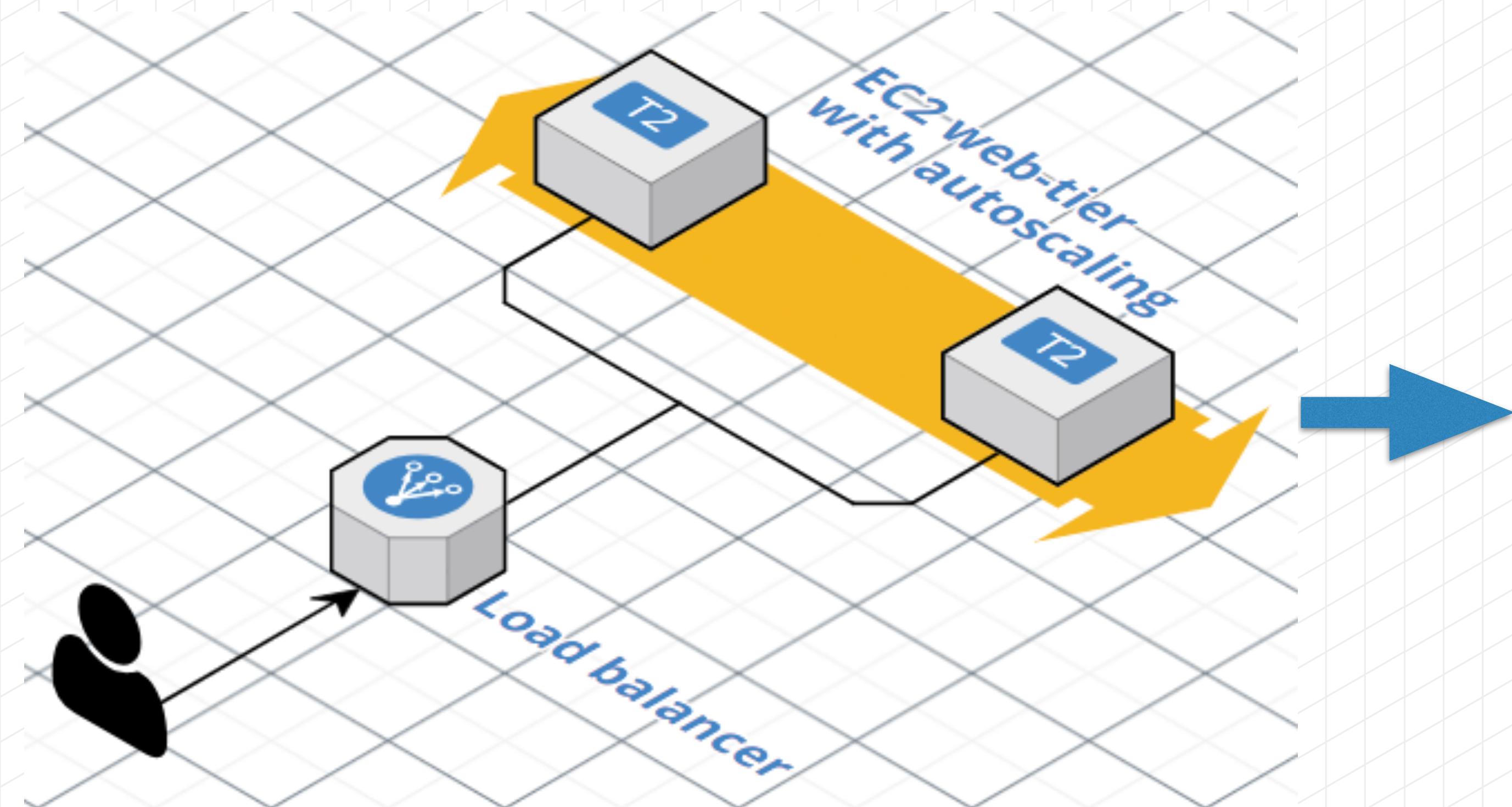
Drag to create multi-selection

Your AWS infrastructure

- Design
- Draw
- Deploy – What to do with deployment?

Requirements & Preferences

- Infrastructure as code
- Reproducible
- Automation
- Proven technologies
- Active community
- Available commercial support
- Follow best-practices
- Reduce copy-paste

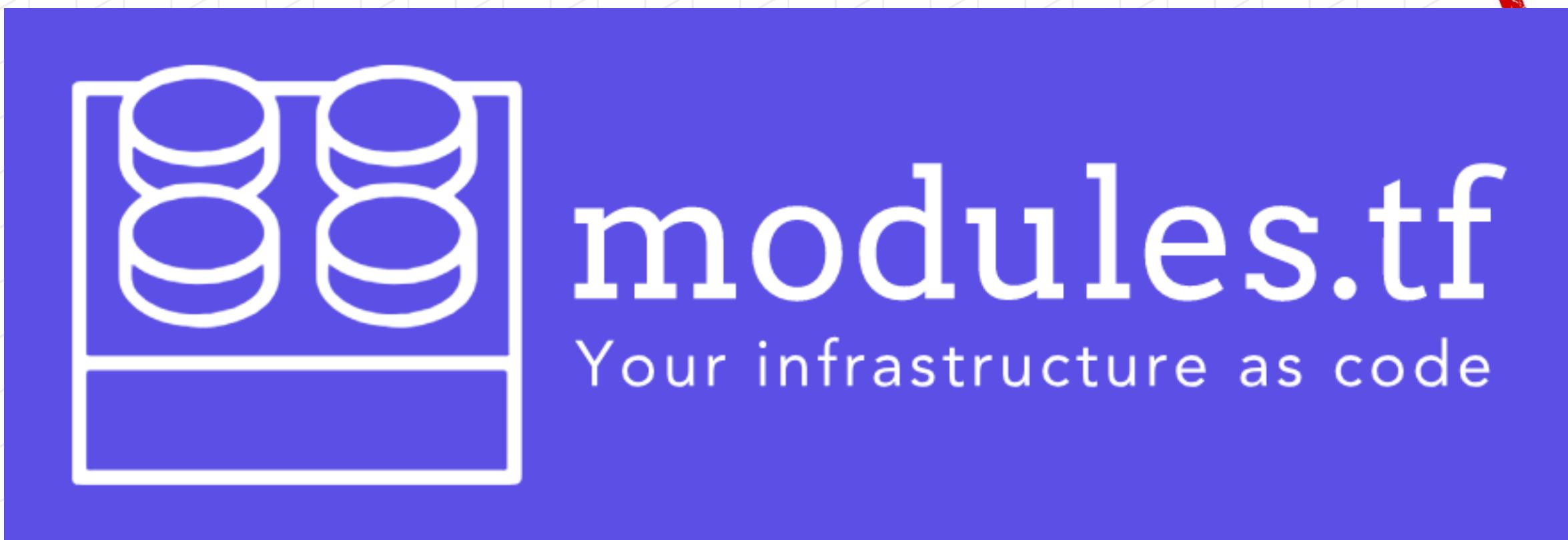


```
1 variable "aws_region" {
2   description = "Region where resources should be created"
3   default      = "eu-west-1"
4 }
5
6 provider "aws" {
7   region = var.aws_region
8 }
9
10 resource "aws_s3_bucket" "this" {
11   bucket = "my-bucket-${random_pet.bucket.id}"
12 }
13
14 resource "random_pet" "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 }
```



@antonbabenko

100% Opensource



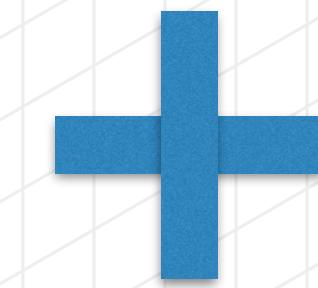
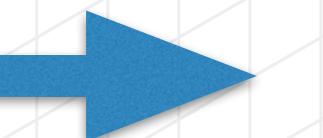
100% FREE

Infrastructure as code generator – from visual diagrams to Terraform

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

Generated infrastructure code – <https://github.com/antonbabenko/modules.tf-demo>

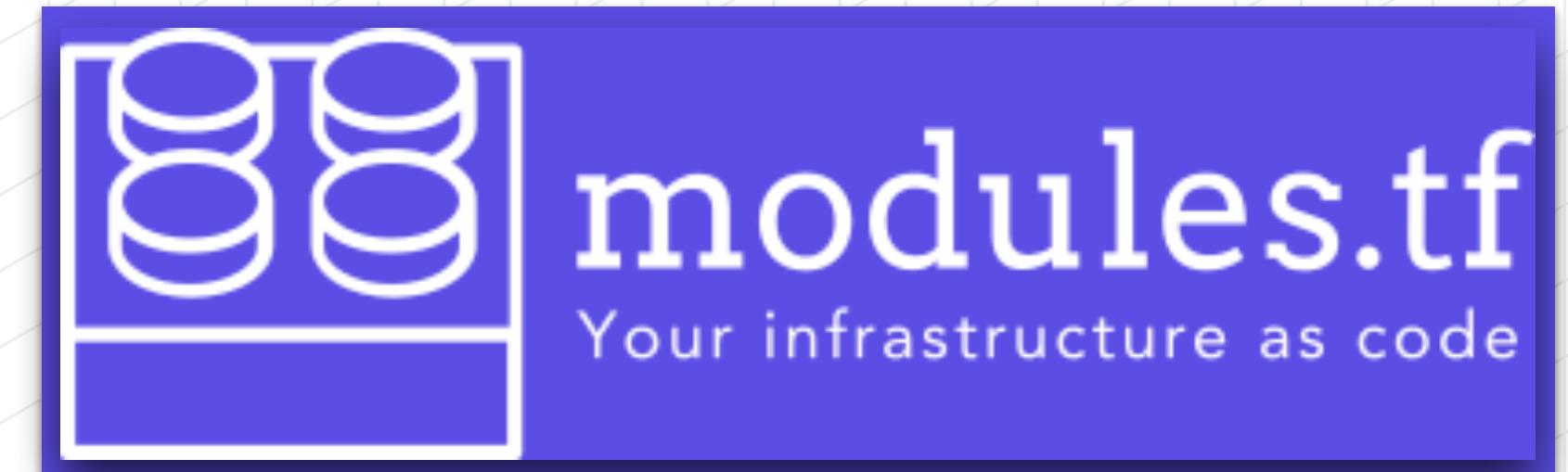
@antonbabenko



- ✓ [cloudcraft.co](#) – design, plan and visualize
- ✓ [terraform-aws-modules](#) – building blocks of AWS infrastructure
- ✓ [Terraform](#) – manage infrastructure as code
- ✓ [Terragrunt](#) – orchestrate Terraform configurations

Demo

- ✓ A picture's worth a thousand words
- ✓ Design, draw and deploy infrastructure as code

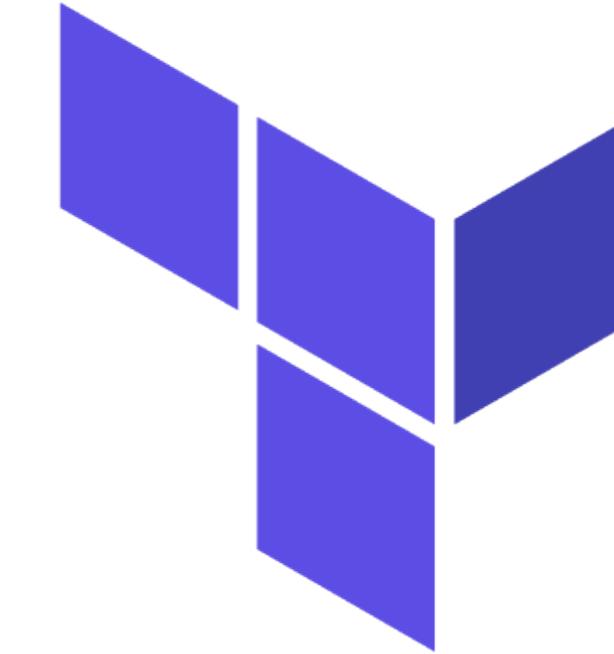
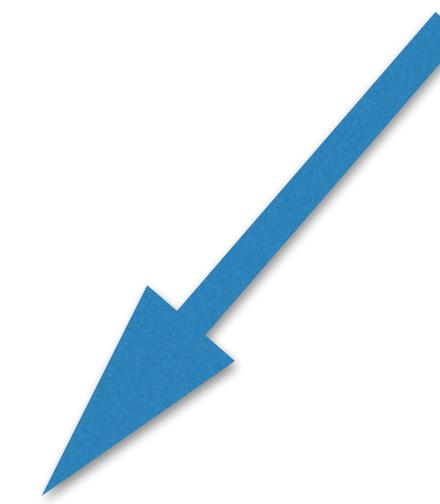


modules.tf – generated code

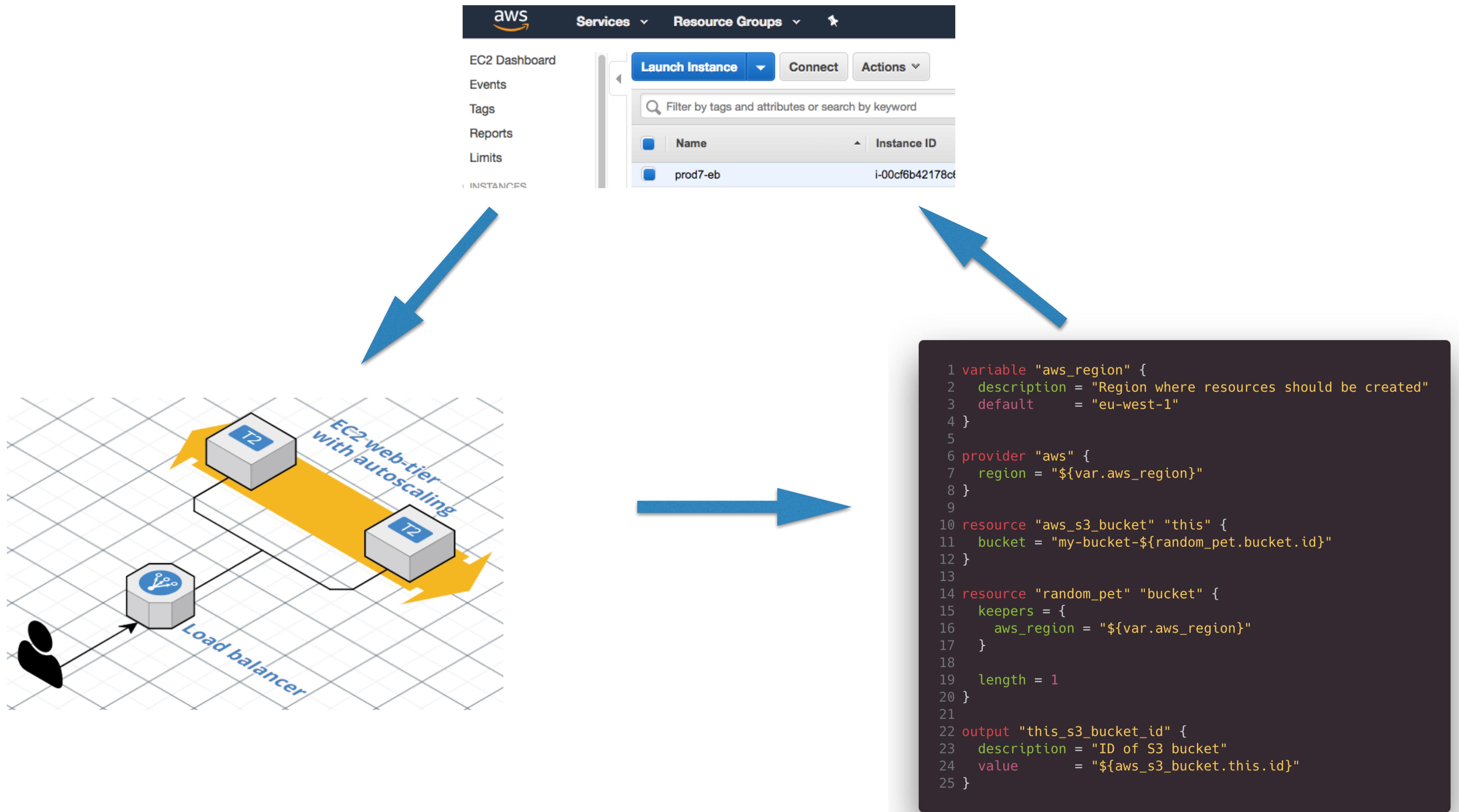
- ✓ Potentially ready-to-use Terraform configurations
- ✓ Suits best for bootstrapping
- ✓ Enforces Terraform best-practices and segregation of duties between Terraform developers and users
- ✓ Batteries included (terraform-aws-modules, terragrunt, pre-commit-terraform)
- ✓ 100% free and open-source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



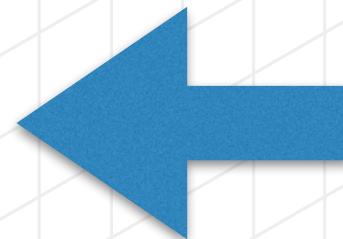
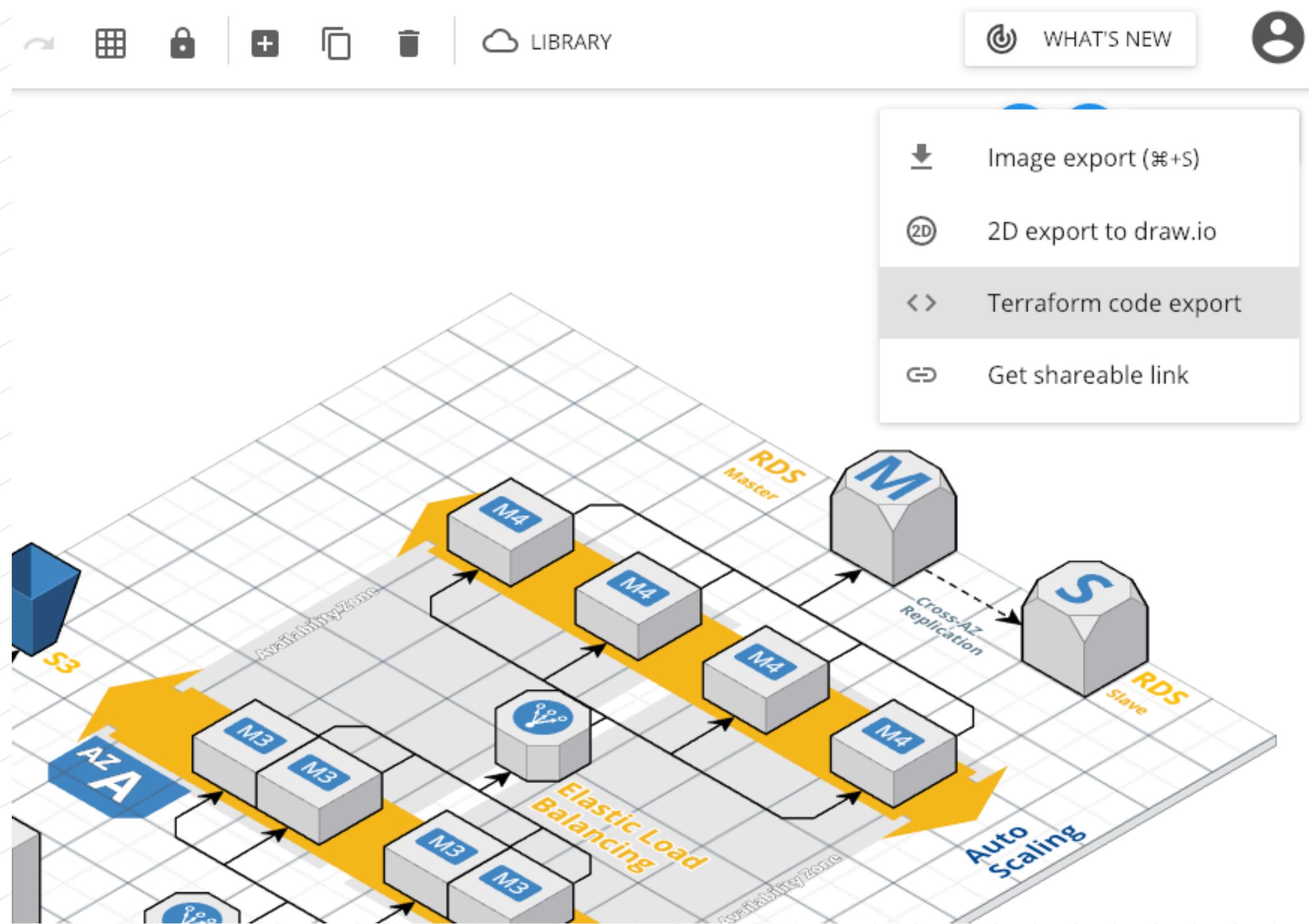
Cloudcraft



HashiCorp
Terraform



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"

Your turn... What do you want to do?

- Build real infrastructure using <https://github.com/terraform-aws-modules>
- or BYOP (Bring Your Own Problem)

Build real infrastructure

- ▀ <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

Thanks!

Questions?

github.com/antonbabenko

twitter.com/antonbabenko