Terraform AWS modules workshop

Anton Babenko
@antonbabenko

October 2019

Anton Babenko

AWS Community Hero / Terraform 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/tfvars-annotations — update terraform.tfvars using annotations
- antonbabenko/modules.tf-lambda — generate Terraform code from visual diagrams
- antonbabenko/terragrunt-reference-architecture — Terragrunt reference architecture
- www.terraform-best-practices.com
- medium.com/@anton.babenko
- @antonbabenko — Twitter, GitHub, Linkedin

DevOpsDays Oslo — 22-23rd of October, 2019

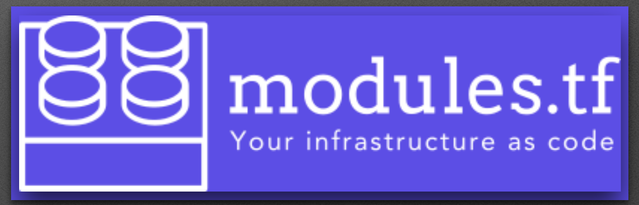
DEVOPSDAYS
OSLO | 2019

http://devopsdays.org/events/2019-oslo/

Registration: https://ti.to/devopsdaysoslo/2019/

What do I do?

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



My interview: https://medium.com/@anton.babenko/my-terraform-aws-journey-hashitimes-interview-73d1b542fcc0

My email: anton@antonbabenko.com

LinkedIn: https://www.linkedin.com/in/antonbabenko

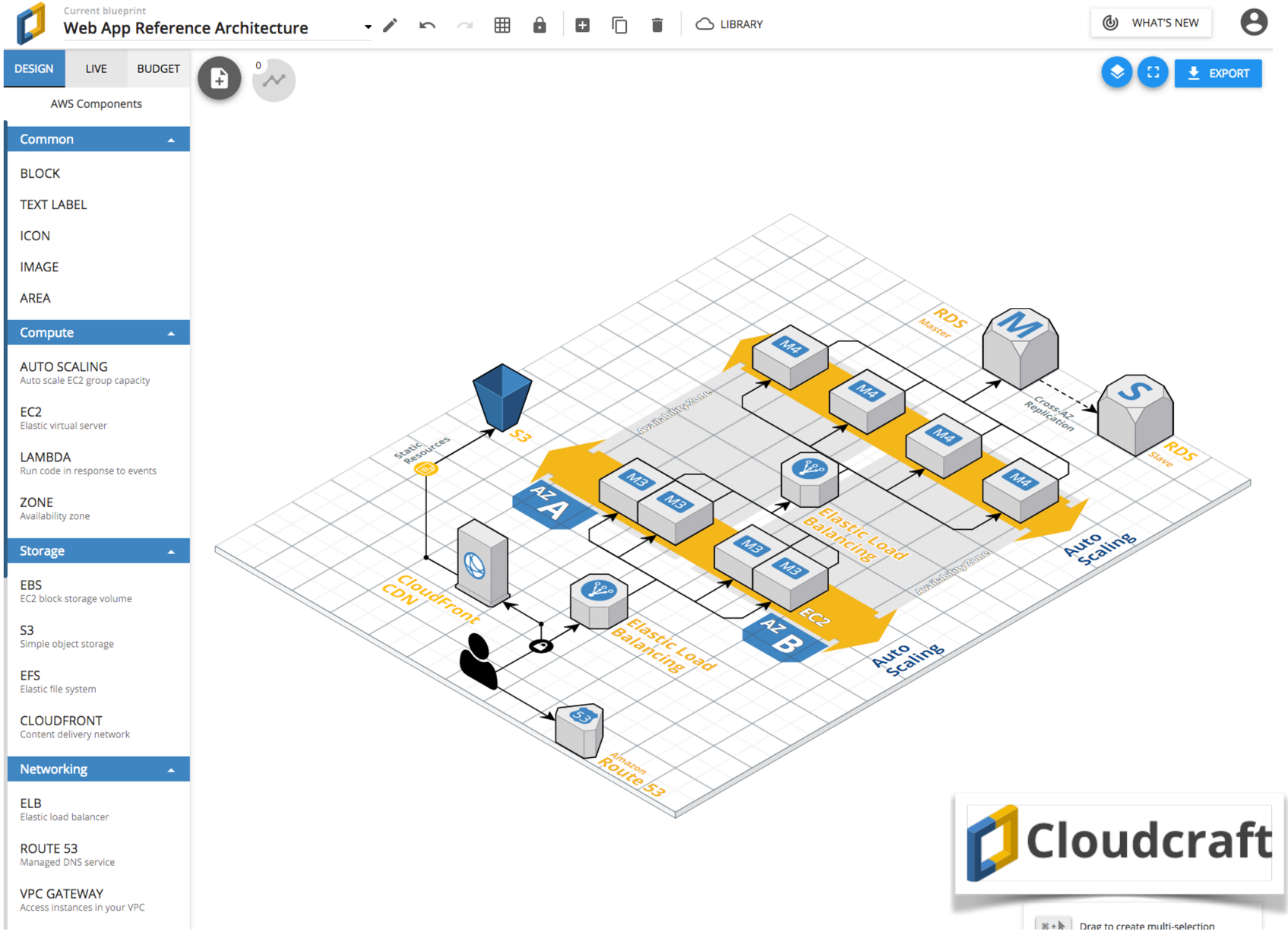
Collection of open-source Terraform AWS modules supported by the community.

More than 5 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



Cloudcraft.co — the best way to draw AWS diagrams

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

Infrastructure as code makes DevOps possible

Key benefits:

- Treat infrastructure like application code
- Always know what changed
- Validate infrastructure before deployment

https://dzone.com/articles/infrastructure-as-code-the-benefits

Agenda

- Intro into AWS, IaC and Terraform
- Terraform modules
- Terraform 0.12
- Building AWS infrastructure using terraform-aws-modules

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

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

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

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

Try this — http://bit.ly/aws-icon-quiz

@antonbabenko

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

– What is infrastructure?

amazon web services

AWS CloudFormation

Google Cloud

Google Cloud Deployment Manager

Azure

Azure Resource Manager

Configuration Management Tools

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

```hcl
variable "aws_region" {
  description = "Region where resources should be created"
  default     = "eu-west-1"
}

provider "aws" {
  region = var.aws_region
}

resource "aws_s3_bucket" "this" {
  bucket = "my-bucket-${random_pet.bucket.id}"
}

resource "random_pet" "bucket" {
  keepers = {
    aws_region = var.aws_region
  }

  length = 1
}

output "this_s3_bucket_id" {
  description = "ID of S3 bucket"
  value       = aws_s3_bucket.this.id
}
```

```
$ 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.
```

```
  # 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

@antonbabenko

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

```
module "atlantis_alb_sg" {
  source  = "terraform-aws-modules/security-group/aws//modules/https-443"
  version = "v2.0.0"

  name        = "atlantis-alb"
  vpc_id      = "vpc-12345678"
  description = "Security group with HTTPS ports open for everybody (IPv4 CIDR)"

  ingress_cidr_blocks = ["0.0.0.0/0"]
}
```

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

```
module "atlantis" {
  source = "terraform-aws-modules/atlantis/aws"

  name = "atlantis"

  # VPC
  cidr            = "10.20.0.0/20"
  azs             = ["eu-west-1a", "eu-west-1b", "eu-west-1c"]
  private_subnets = ["10.20.1.0/24", "10.20.2.0/24", "10.20.3.0/24"]
  public_subnets  = ["10.20.101.0/24", "10.20.102.0/24", "10.20.103.0/24"]

  # DNS
  route53_zone_name = "terraform-aws-modules.modules.tf"

  # Atlantis app
  atlantis_github_user       = "atlantis-bot"
  atlantis_github_user_token = "examplegithubtoken"
}
```

```
# terraform-aws-atlantis/main.tf

module "vpc" {
  source  = "terraform-aws-modules/vpc/aws"
  version = "v1.32.0"
  # ...
}

module "alb" {
  source  = "terraform-aws-modules/alb/aws"
  version = "v3.4.0"
  # ...
}

module "alb_https_sg" {
  source  = "terraform-aws-modules/security-group/aws//modules/https-443"
  version = "v2.0.0"
  # ...
}

module "ecs" {
  source  = "terraform-aws-modules/ecs/aws"
  version = "v1.0.0"
  # ...
}
```

Terraform 0.12

…and why it is so important?

Terraform 0.12

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

Who are you?

Terraform users vs developers

Types of Terraform users

- Terraform developers
- Terraform users (everyone else)

Terraform developers

- Write and support Terraform modules
- Implement company's standards (security, encryption, integrations)
- Maintain reference architectures

Terraform users (everyone)

- Use Terraform modules by specifying correct values
- Domain experts
- May not have "Terraform" in LinkedIn profile

Terraform 0.12 for developers

- DevOps&Terraform developers
- Allow to implement flexible/dynamic/reusable Terraform modules

Terraform 0.12 for users

- Terraform users
  - Like HCL2 lightweight syntax more

@antonbabenko

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

More than 5 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

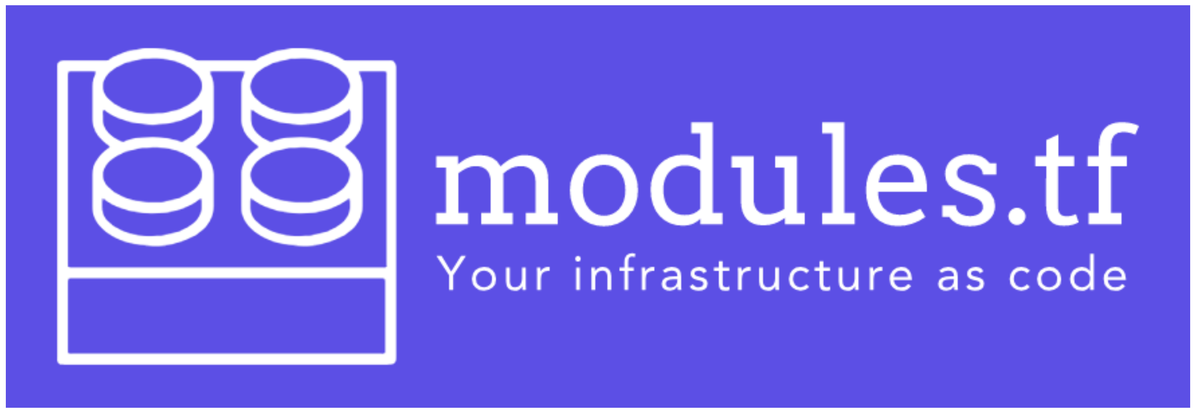
Your turn… What do you want to do?

- 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

Bonus


modules.tf
Your infrastructure as code

Current blueprint
Web App Reference Architecture

LIBRARY

WHAT'S NEW

DESIGN | LIVE | BUDGET

EXPORT

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

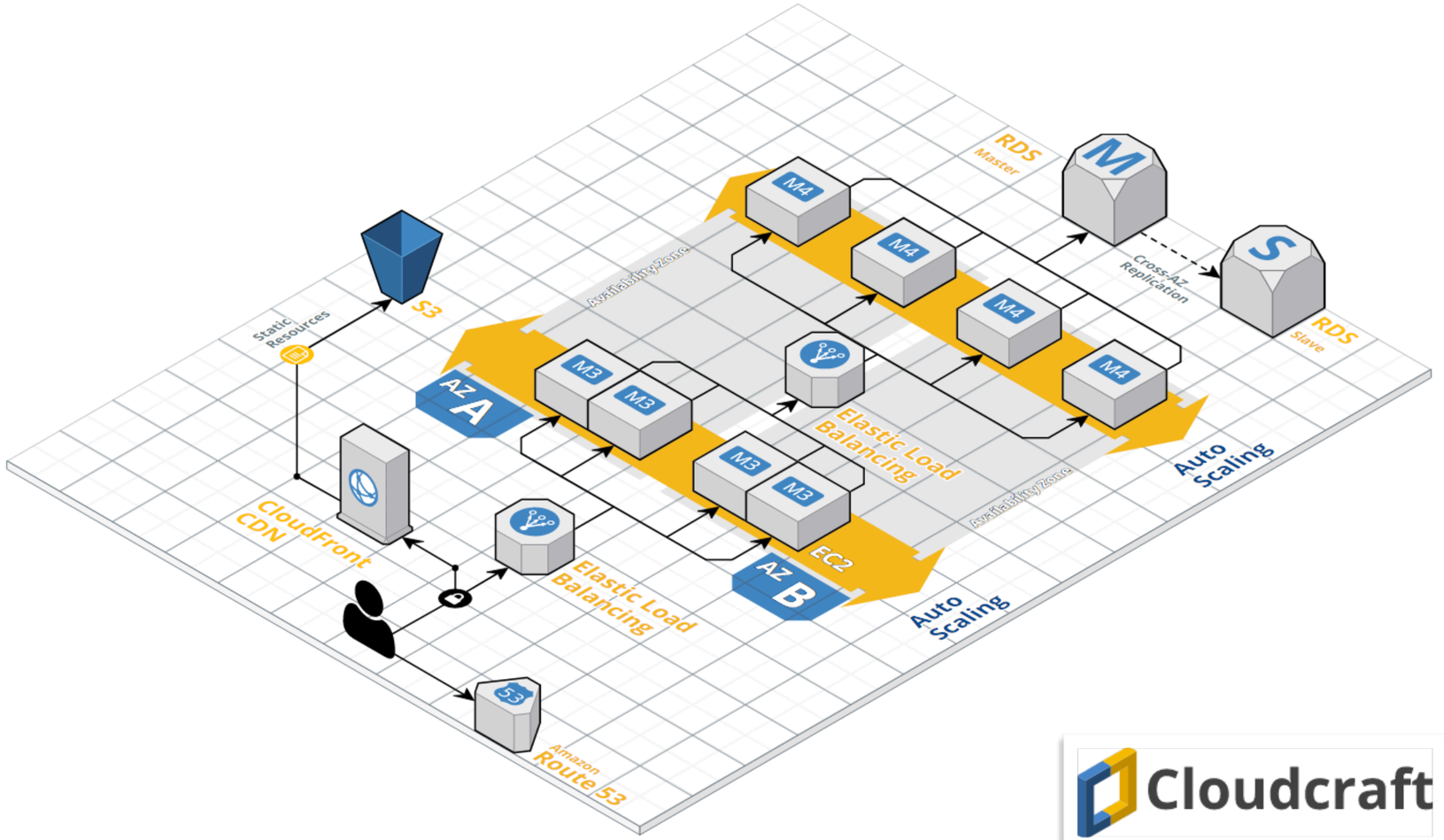
CLOUDFRONT
Content delivery network

Networking

ELB
Elastic load balancer

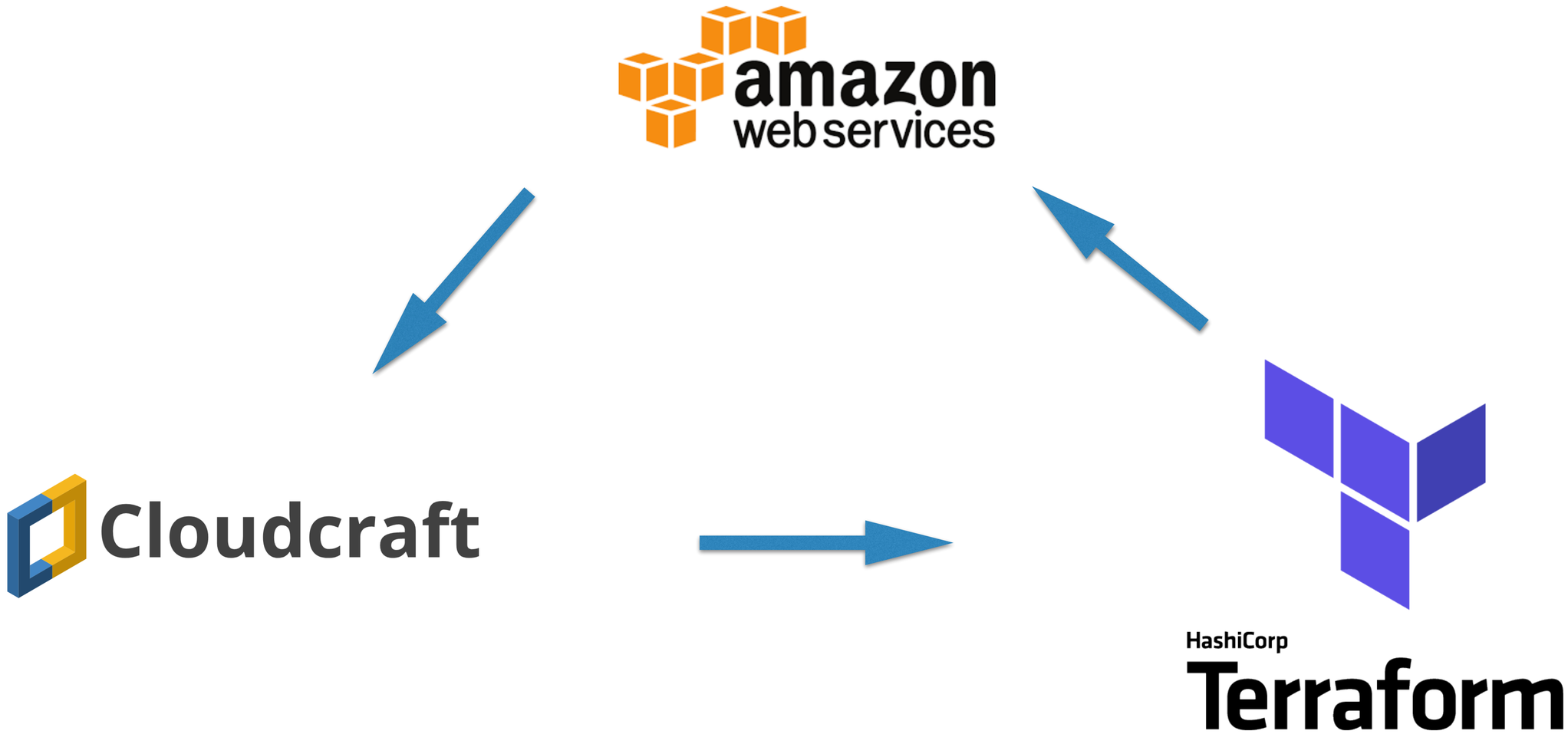
ROUTE 53
Managed DNS service

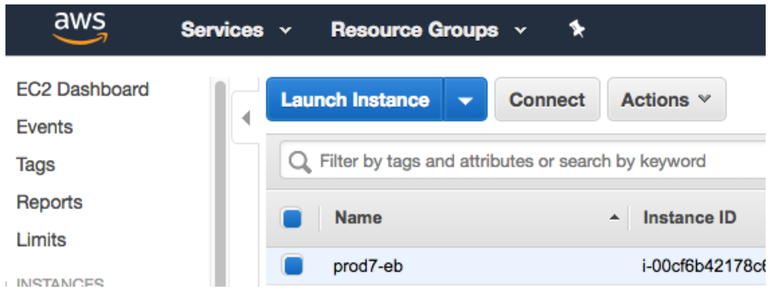
VPC GATEWAY
Access instances in your VPC

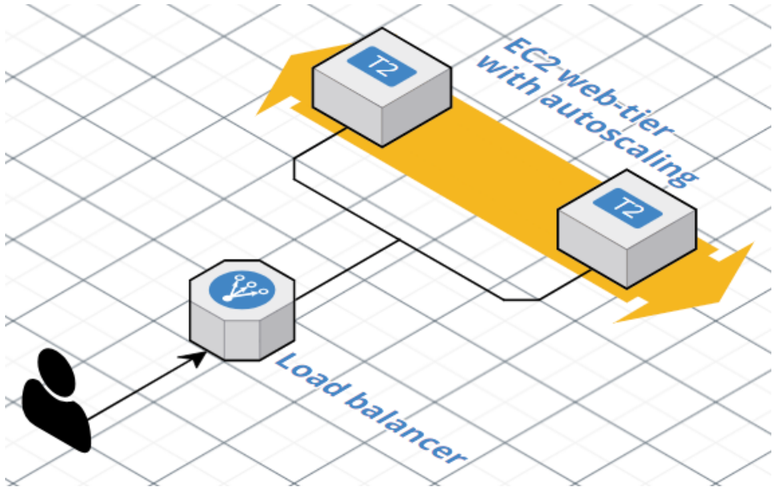


Cloudcraft

Drag to create multi-selection

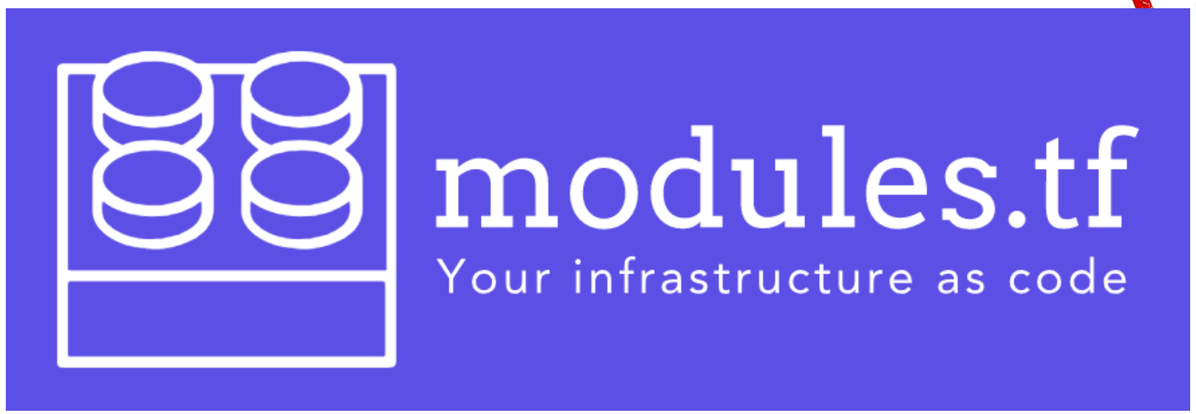

amazon
web services
Cloudcraft
HashiCorp
Terraform






```hcl
variable "aws_region" {
  description = "Region where resources should be created"
  default     = "eu-west-1"
}

provider "aws" {
  region = "${var.aws_region}"
}

resource "aws_s3_bucket" "this" {
  bucket = "my-bucket-${random_pet.bucket.id}"
}

resource "random_pet" "bucket" {
  keepers = {
    aws_region = "${var.aws_region}"
  }

  length = 1
}

output "this_s3_bucket_id" {
  description = "ID of S3 bucket"
  value       = "${aws_s3_bucket.this.id}"
}
```

- ✓ cloudcraft.co — design, plan and visualize
- ✓ terraform-aws-modules — building blocks of AWS infrastructure
- ✓ Terraform — infrastructure as code

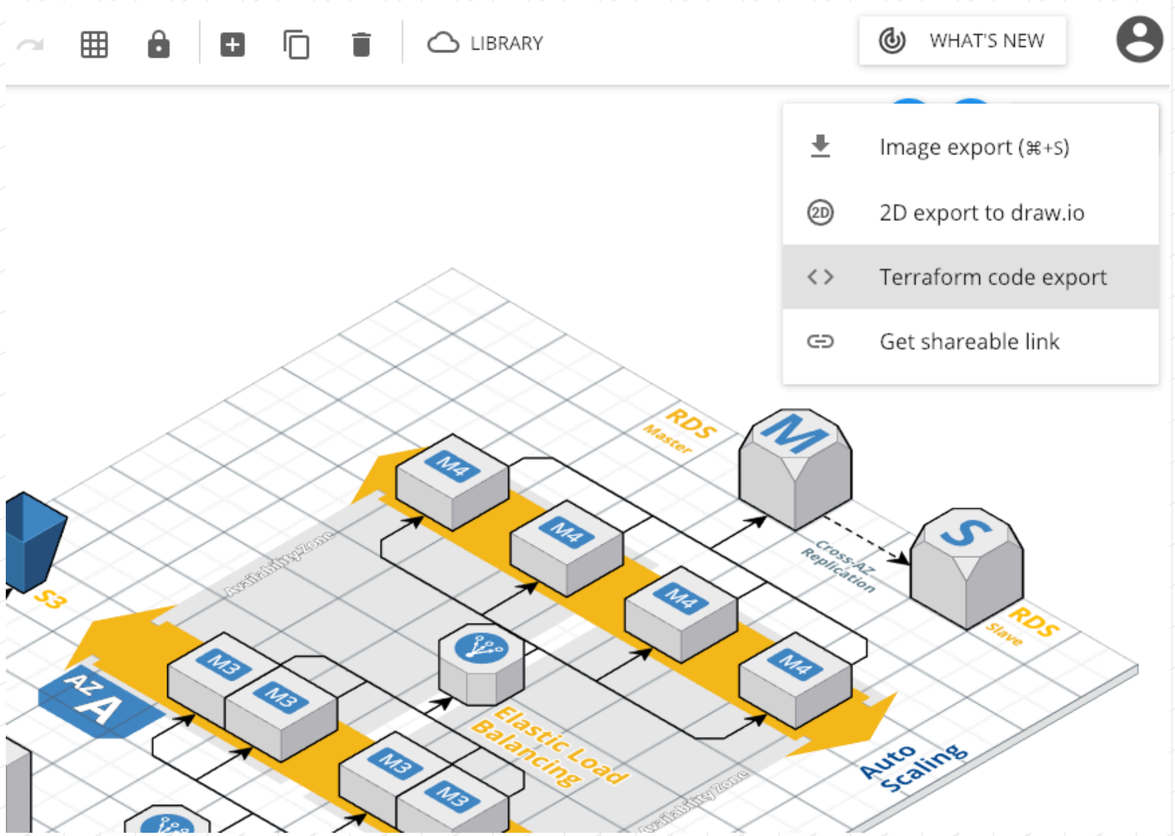


Infrastructure as code generator — from visual diagrams to Terraform

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