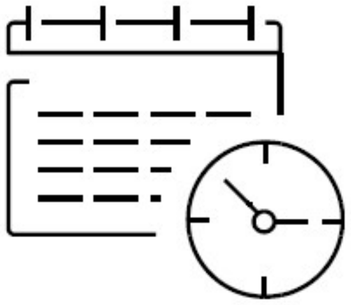
aws open source

AWS Hero Spotlight: Why & how to use Terraform AWS modules



ANTON BABENKO, COMMUNITY HERO
OPEN@
JUNE 16 2021

Anton Babenko

AWS Community Hero / HashiCorp Ambassador / Certified Terraform fanatic since 2015.

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

I 💚 open-source:

- antonbabenko/terraform-aws-devops — my Terraform, AWS, and DevOps projects
- terraform-community-modules + terraform-aws-modules
- antonbabenko/pre-commit-terraform — clean code, documentation, and more
- serverless.tf — Doing serverless on AWS with Terraform
- www.terraform-best-practices.com
- bit.ly/terraform-youtube — Your weekly dose of Terraform (live stream)
- weekly.tf - Terraform Weekly newsletter
- @antonbabenko — Twitter, GitHub, Linkedin

SERVERLESS.TF

Build, deploy, and manage serverless applications and infrastructure on AWS using Terraform

Collection of 40+ open-source Terraform AWS modules supported by the community with over 40 million provisions

(VPC, Autoscaling, RDS, Security Groups, EC2, ELB, ALB, Redshift, SNS, SQS, IAM, EKS, ECS, TGW, S3 bucket, CloudFront, Lambda, API Gateway, EventBridge, AppSync…)

github.com/terraform-aws-modules

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

SERVERLESS.TF

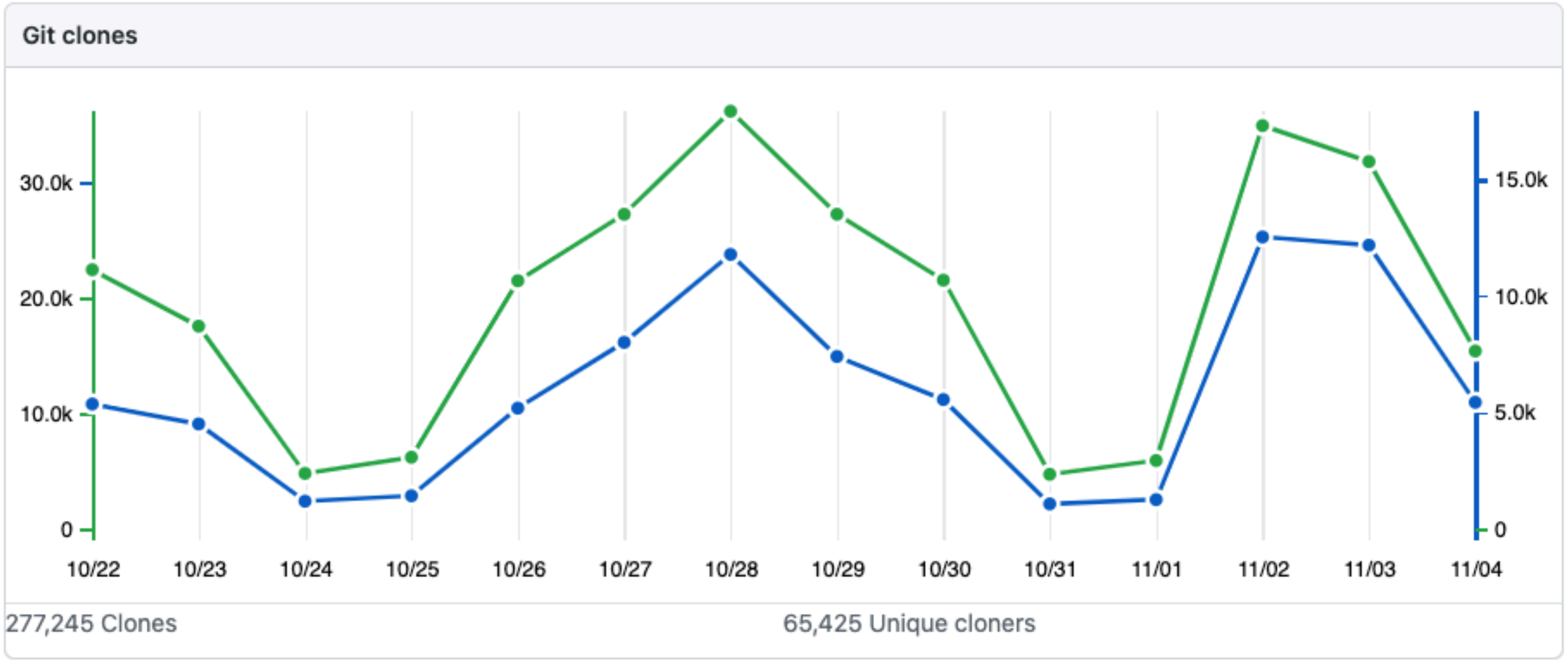
Build, deploy, and manage serverless applications
and infrastructure on AWS using Terraform

18 sponsors are funding antonbabenko's work.

2000+ pull-requests and issues resolved!

Kudos to Thierno, Bryant, Sven, Ilya, Max, Alaric, and more than 300 other contributors!

GitHub traffic — terraform-aws-vpc *



* this graph does not include usages from the official Terraform Registry

Why do developers like terraform-aws-modules?

- "Undifferentiated Heavy Lifting"
- Very large community
- Almost all features are shown in examples
- Documentation (by humans and pre-commit-terraform) — anything is better than nothing
- Modules compatibility and patterns used in each module
- Combine modules together to abstract away low-level complexity
- Week-zero support for the stable versions of Terraform

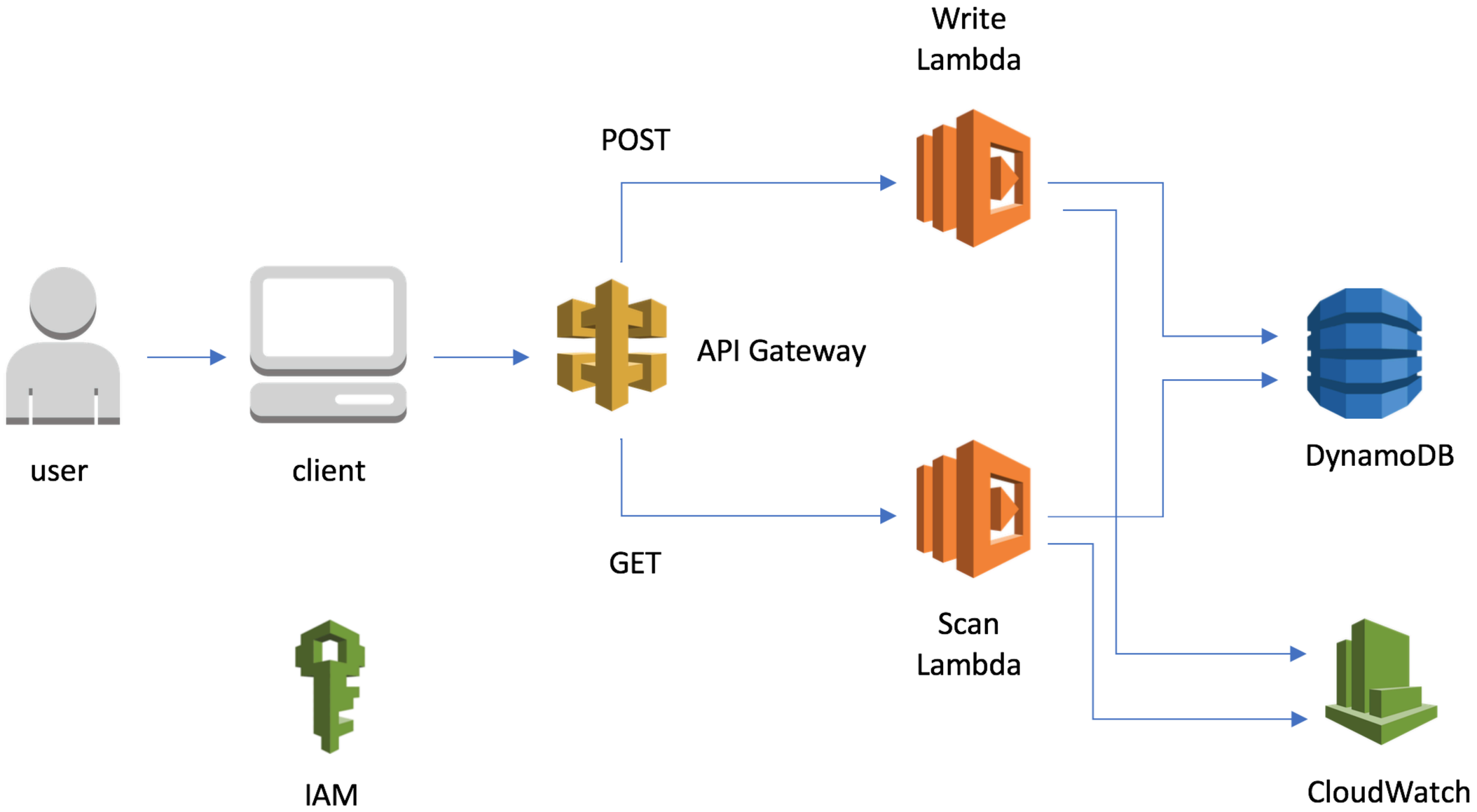
```
[]{15:57}~/Sites/terraform-aws-modules ➲ scc **/*.tf
```

Language	Files	Lines	Blanks	Comments	Code	Complexity
Terraform	1065	79992	13327	8237	58428	5089
Markdown	1	43	12	0	31	0
Total	1066	80035	13339	8237	58459	5089

```
Estimated Cost to Develop (organic) $1,935,482
Estimated Schedule Effort (organic) 17.676407 months
Estimated People Required (organic) 9.727723
```

```
Processed 2444617 bytes, 2.445 megabytes (SI)
```

Demo: terraform-aws-modules / serverless.tf



```
# Option 1:
# 1. ~40 resources + data-sources
# 2. 200 lines of code
#
# Lambda GET
resource "aws_lambda_function" "lambda_get" {}
resource "aws_lambda_permission" "lambda_get" {}
resource "aws_iam_role" "lambda_get" {}
resource "aws_iam_policy" "lambda_get" {}
resource "aws_iam_policy_attachment" "lambda_get" {}

# Lambda POST
resource "aws_lambda_function" "lambda_post" {}
resource "aws_lambda_permission" "lambda_post" {}
resource "aws_iam_role" "lambda_post" {}
resource "aws_iam_policy" "lambda_post" {}
resource "aws_iam_policy_attachment" "lambda_post" {}

# API Gateway
resource "aws_apigatewayv2_api" "this" {}
resource "aws_apigatewayv2_domain_name" "this" {}
resource "aws_apigatewayv2_api_mapping" "this" {}
resource "aws_apigatewayv2_route" "this" {}
resource "aws_apigatewayv2_integration" "this" {}

# DynamoDB
resource "aws_dynamodb_table" "this" {}
```

```
# Option 2:
# 1. ~40 resources + data-sources
# 2. Add variables and outputs = 1000 lines of code

# variables.tf
variable "lambda_get_memory_size" {
  description = "Memory size for get-record Lambda function"
  default     = 256
}

# Lambda GET
resource "aws_lambda_function" "lambda_get" {
  function_name = "get-record"
  memory_size   = var.lambda_get_memory_size
  # ...
}

resource "aws_lambda_permission" "lambda_get" {}
resource "aws_iam_role" "lambda_get" {}
resource "aws_iam_policy" "lambda_get" {}
resource "aws_iam_policy_attachment" "lambda_get" {}

# Lambda POST
resource "aws_lambda_function" "lambda_post" {}
resource "aws_lambda_permission" "lambda_post" {}
resource "aws_iam_role" "lambda_post" {}
resource "aws_iam_policy" "lambda_post" {}
resource "aws_iam_policy_attachment" "lambda_post" {}

# API Gateway
resource "aws_apigatewayv2_api" "this" {}
resource "aws_apigatewayv2_domain_name" "this" {}
resource "aws_apigatewayv2_api_mapping" "this" {}
```

```
# Option 3:
# 1. Move resources into custom Terraform modules
# 2. Reuse Terraform code for similar resources

# Lambda GET
module "lambda_get" {
  source = "./lambda"

  function_name = "get-record"
  memory_size   = var.lambda_get_memory_size
  # ... values for IAM role, policies, permissions...
}

# Lambda POST
module "lambda_post" {
  source = "./lambda"

  function_name = "post-record"
  memory_size   = var.lambda_post_memory_size
  # ... values for IAM role, policies, permissions...
}

# API Gateway
module "api_gateway" {}

# DynamoDB
module "dynamodb_table" {}
```

```
# Option 4:
# 1. Use terraform-aws-modules
# 2. Much more features than in custom modules
# 3. Your configuration values = 100 lines of code

# Lambda GET
module "lambda_get" {
  source = "terraform-aws-modules/lambda/aws"

  function_name = "get-record"
  memory_size   = 256
  # ... values for IAM role, policies, permissions...
}

# Lambda POST
module "lambda_post" {
  source = "terraform-aws-modules/lambda/aws"

  function_name = "post-record"
  memory_size   = 512
  # ... values for IAM role, policies, permissions...
}

# API Gateway
module "api_gateway" {
  source = "terraform-aws-modules/apigateway-v2/aws"
}

# DynamoDB
module "dynamodb_table" {
  source = "terraform-aws-modules/dynamodb-table/aws"
}
```

Resources

- https://github.com/terraform-aws-modules
- https://registry.terraform.io/
- https://github.com/antonbabenko/serverless.tf-playground

Thanks!

Questions?

Subscribe to "Your Weekly Dose of Terraform" — http://bit.ly/terraform-youtube
and weekly.tf

github.com/antonbabenko

twitter.com/antonbabenko