



SUMMIT
TORONTO

DEV04

Open-source Infrastructure-as-Code with Terraform and AWS

Eric Wright
Technology Evangelist
Turbonomic

Agenda

Why Terraform?

Terraform OSS versus Terraform Cloud

Building our First Simple Deployment

The Super Cool Classroom Use-case

Why Terraform?

“Never send a human to do a machine’s job.”

Agent Smith
The Matrix

Why Terraform?

Codified representation of infrastructure

Can be easily used with version control

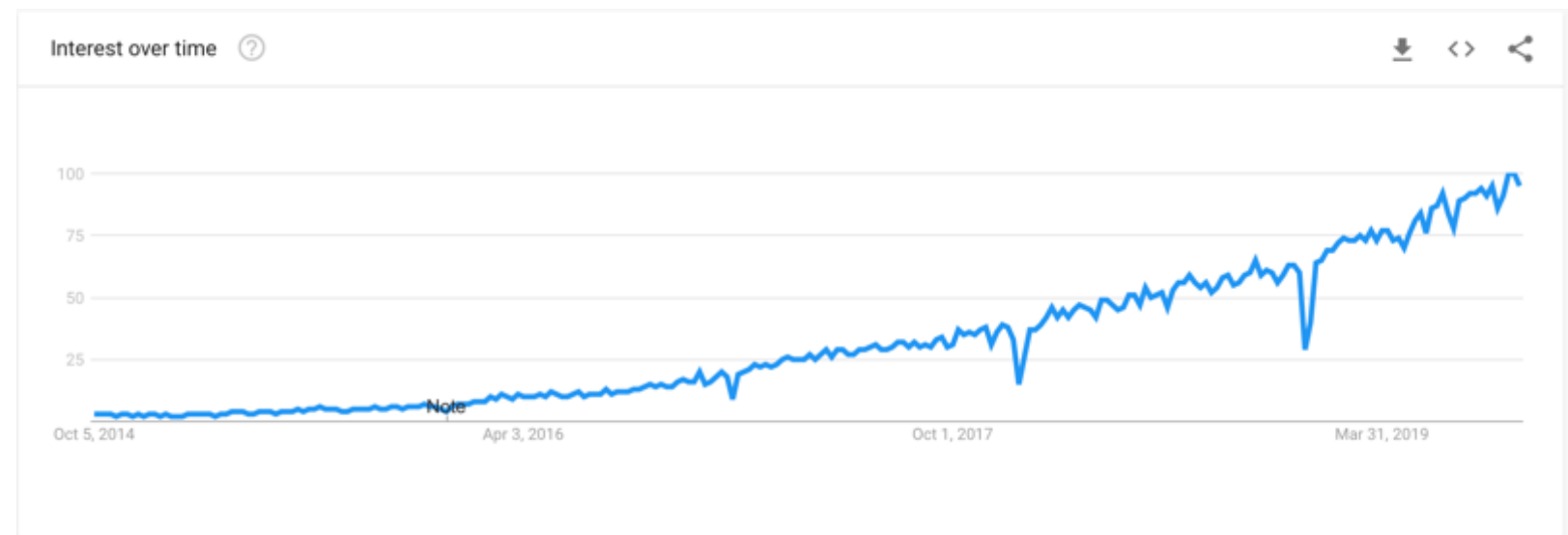
Open Source and widely used

Many providers for everything from infrastructure to applications

Significant service coverage for AWS

DevOps-friendly

Limited code experience needed

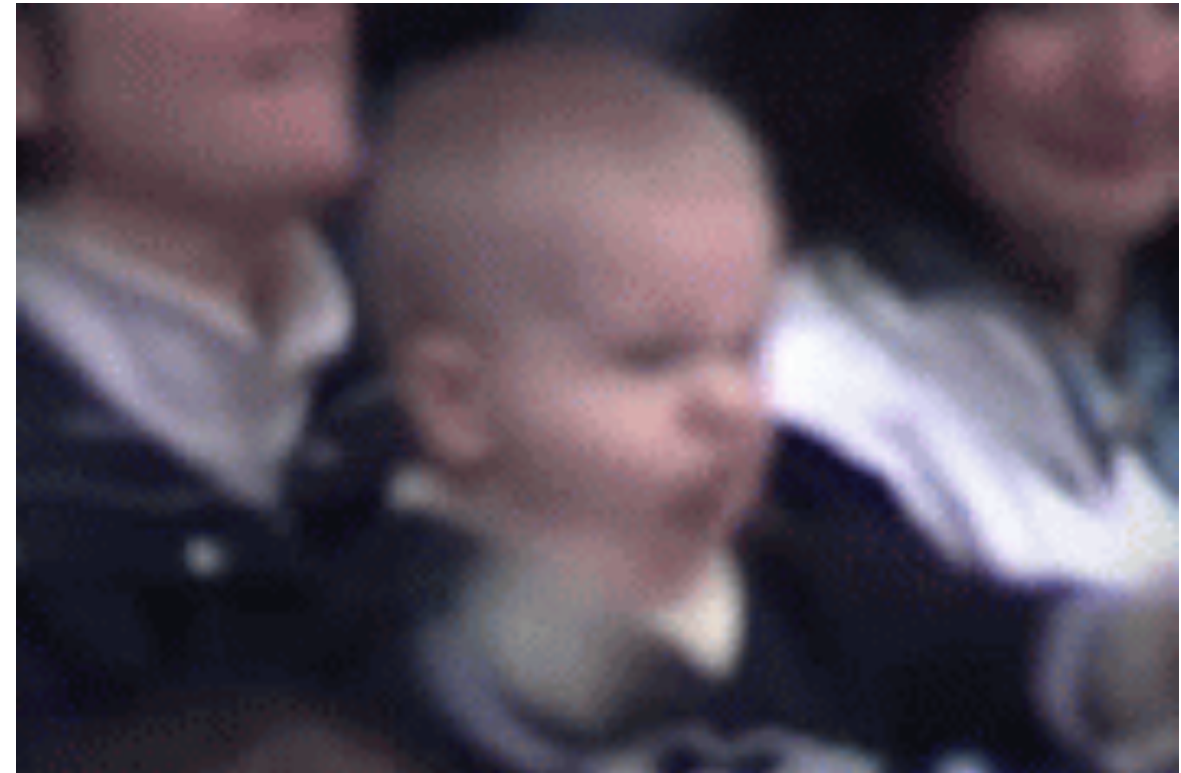


Terraform Cloud versus Terraform OSS

Terraform Cloud FTW!

Stuff that becomes better via Terraform Cloud include...

- Centralized state management
- Collaboration for teams
- Security and governance
- Versioning of state files
- Run local or remote
- Full application UI
- Webhooks and notifications
- Fully API accessible



Interpolate All the Things!

Terraform will order resources as needed (Yay!!)

Interpolated references feed output of one resource dynamically to another resource (Yay again!!)

Disclaimer: Example code for the talk contains some hard-coded references – updates pending to fully variableify it

Second disclaimer: Variableify is not a word



Things You Will Need

- Terraform 0.12 or higher
- Terraform Cloud account
- Create an organization
- Create a workspace
- Create an API token
- AWS client key and secret



Building our First Simple Deployment

Starting With the Basics

EC2 Instance
no, seriously, that's it



Code Snippets – AWS Provider and Terraform Cloud Backend

```
provider "aws" {  
    access_key = "${var.aws_access_key}"  
    secret_key = "${var.aws_secret_key}"  
    region = "${var.aws_region}"  
}  
  
terraform {  
    backend "remote" {  
        organization = "RapidMatter"  
  
        workspaces {  
            name = "terraform-cloud-aws-summit-toronto"  
        }  
    }  
}
```

Code Snippets – Simple EC2 Instance

```
resource "aws_instance" "web" {
  ami = "ami-c55673a0"
  instance_type = "t2.nano"
  key_name = "${var.aws_key_name}"

  tags = {
    Terraform = "true"
    ProvisionedBy = "Project Terra"
  }
}
```

Demo – Simple EC2 Deployment

“Give me six hours to chop down a tree
and I will spend the first four sharpening
the axe.”

Abraham Lincoln

The Super Cool Classroom Use-case

Expanding the Use-case

Repeatable, public-facing
classroom scenario:

EC2 instance from an AMI

Security Group for HTTP/S + SSH

Application Load Balancer

TLS using AWS Certificate Manager

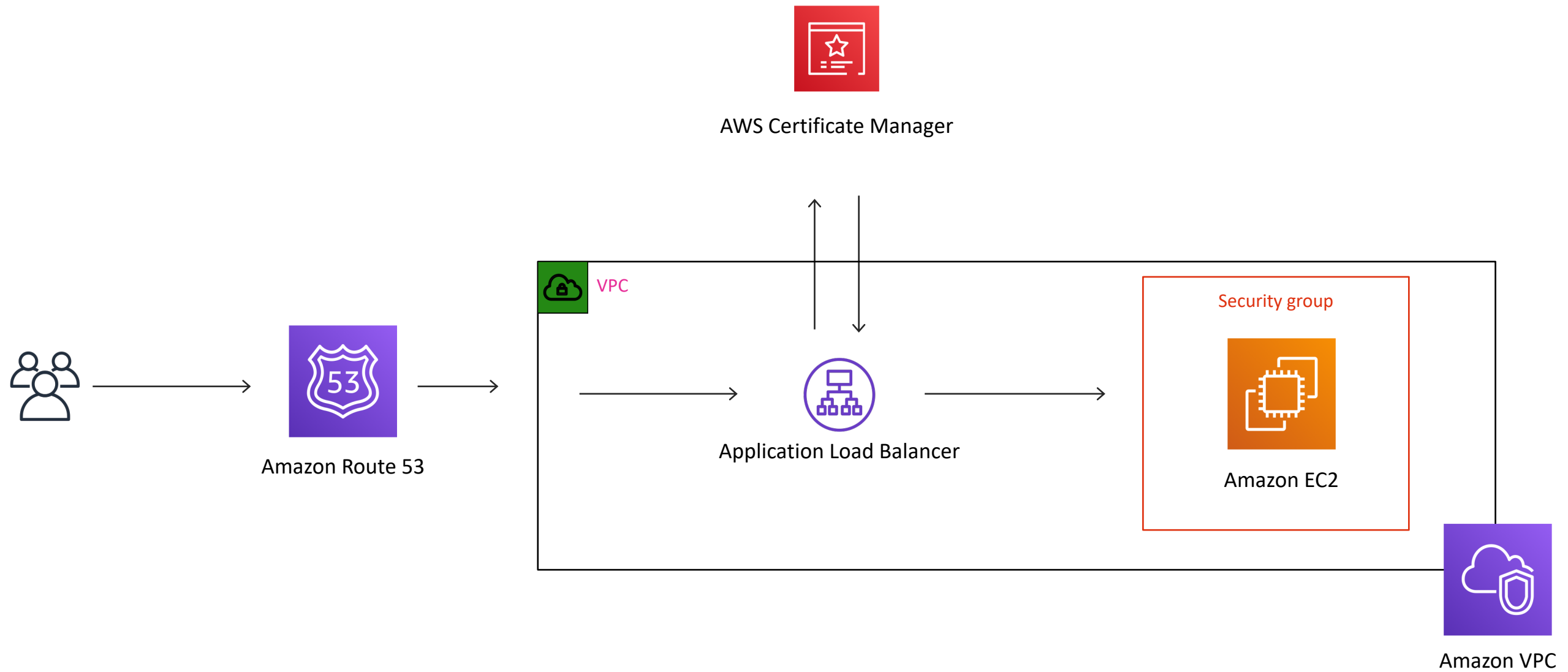
DNS using Route53



All code links and supporting blog available here:

<https://discopos.se/AWSSummit2019DiscoPosse>

The Super Cool Classroom Diagram



Demo – Virtual Classroom Example

Thank you!

Eric Wright
@DiscoPosse
DiscoPosse.com



Please complete the
session survey.