Implementing Terraform on AWS

USING THE AWS PROVIDER

Ned Bellavance
FOUNDER, NED IN THE CLOUD

@ned1313 | nedinthecloud.com

Overview

Course prerequisites

Terraform / CloudFormation

Scenario

AWS provider

Course Prerequisites

Terraform Basics

AWS Basics

Terraform for the AWS Admin

CloudFormation	Terraform
JSON / YAML	HCL
Parameters	Variables
Mapping	Local variables
Conditions	Logical operators
Resources	Resources
Intrinsic Functions	Functions
Nested stacks	Modules
Implicit dependency	Automatic dependency

GLOBOMANTICS

GLOBOMANTICS

Elsie Vasquez
Cloud Architect

Jayson Gibson
Security Administrator

Joshua McGee
Software Engineer

Terraform Providers

Versioned

Data sources

Resources

Modules

Authentication

Authentication

Static credentials

Environmental variables

AWS CLI / Shared credentials file

Instance profile

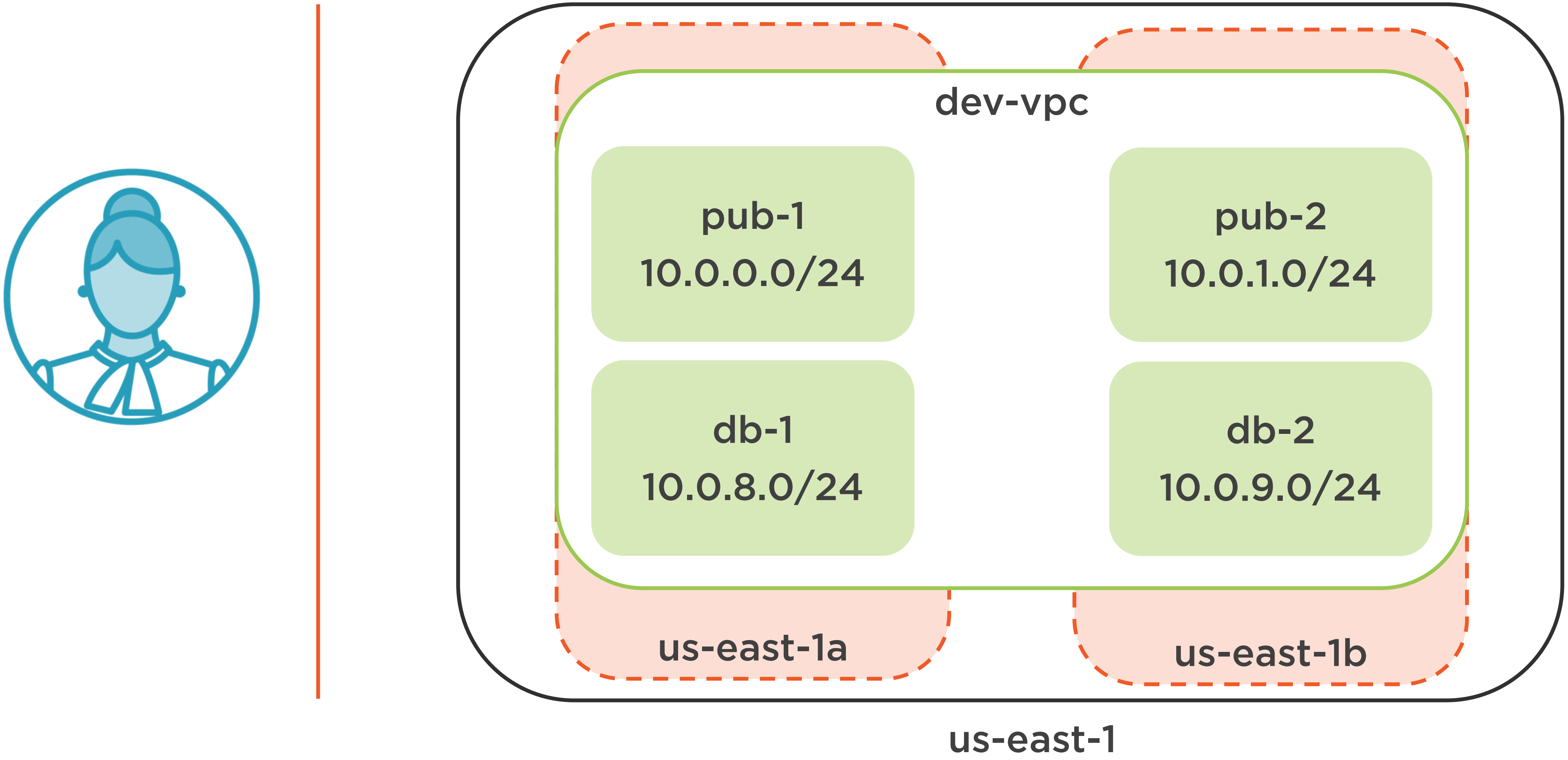
AWS Provider

```
provider "aws" {
  version         = "~> 2.0"
  alias           = "networking"
  region          = var.region
  access_key      = var.access_key
  secret_key      = var.secret_key
}
```

Environment Variables

```
AWS_ACCESS_KEY_ID # Access Key
AWS_SECRET_ACCESS_KEY # Secret Key
AWS_SHARED_CREDENTIALS_FILE # Location of credential file
AWS_PROFILE # Profile name in credentials file
AWS_DEFAULT_REGION # Region to use for provider
AWS_SESSION_TOKEN # Session token for temporary credentials
```

VPC Deployment



Demo

Examine the Terraform files

Deploy the configuration

Review the results

Play along!

- Two AWS accounts
- Full admin access
- AWS CLI
- Terraform and files

Some of the resources deployed in AWS may cost **money**. You’ve been warned.

Summary

Terraform vs. CloudFormation

AWS provider

Authentication and deployment

Coming up:

- Security gets involved
- Multiple account and regions
- Using credential files