Terraform
Infrastructure as Code

- Pradeep Bhadani

About this Workshop

- Who am I?
- Workshop objective
- What we need?

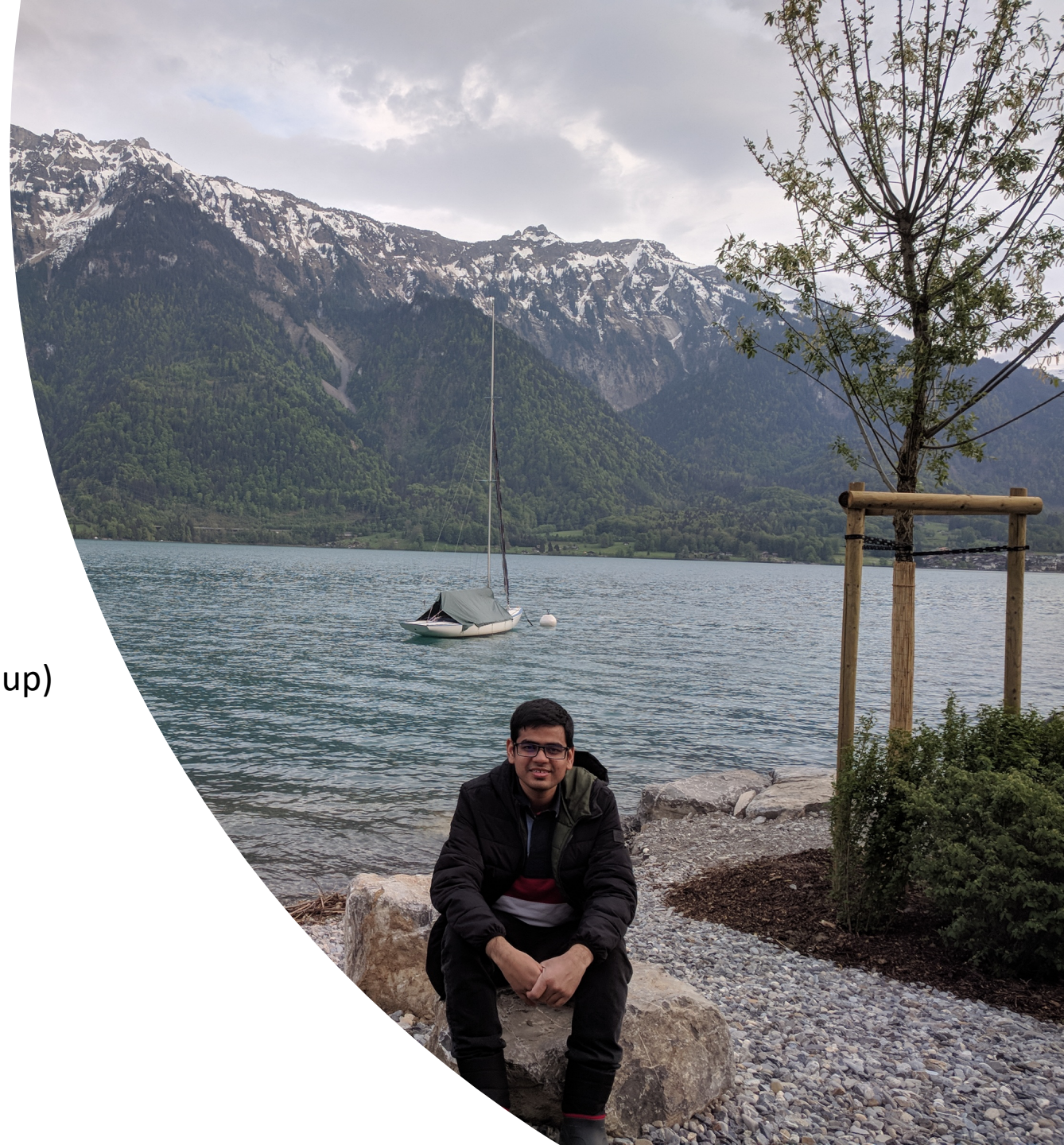
Who am I?

Pradeep Bhadani

Big Data Engineer

Currently working at Hotels.com (brand of Expedia Group)

linkedin.com/in/pradeepbhadani/



Workshop Objective

- Getting started to manage AWS resource as code using Terraform

What we need?

- Access to AWS account
- Text Editor / IDE – Atom or IntelliJ
- Setup Terraform & awscli on workstation

Let’s get started

Legacy World: Problems

- Manual command
- Extensive Documents
- Human error
- Tons of shell scripts
- Time consuming process
- Boring tasks
- Hard to recover from failure
- Difficult to scale

Imagine a World

Imagine a World

~~Manual commands~~	No more manual commands
~~Extensive Documents~~	Self describing documents
~~Human error~~	No Human intervention
~~Tons of shell scripts~~	Not many shell scripts
~~Time consuming process~~	Speedy process
~~Boring tasks~~	Time for fun task
~~Hard to recover from failure~~	Fast recovery
~~Difficult to scale~~	Easy to scale

What is Infrastructure as Code (IAC) ?

- It is an approach to manage Systems, Networks etc.. through Source code.

IAC Principles

- Consistent Infrastructure
- Easy to reproduce
- Easy to manage
- Ability to repeat
- Handles change in design

Different Tools

- Terraform
- Chef
- Ansible
- Puppet
- Salt

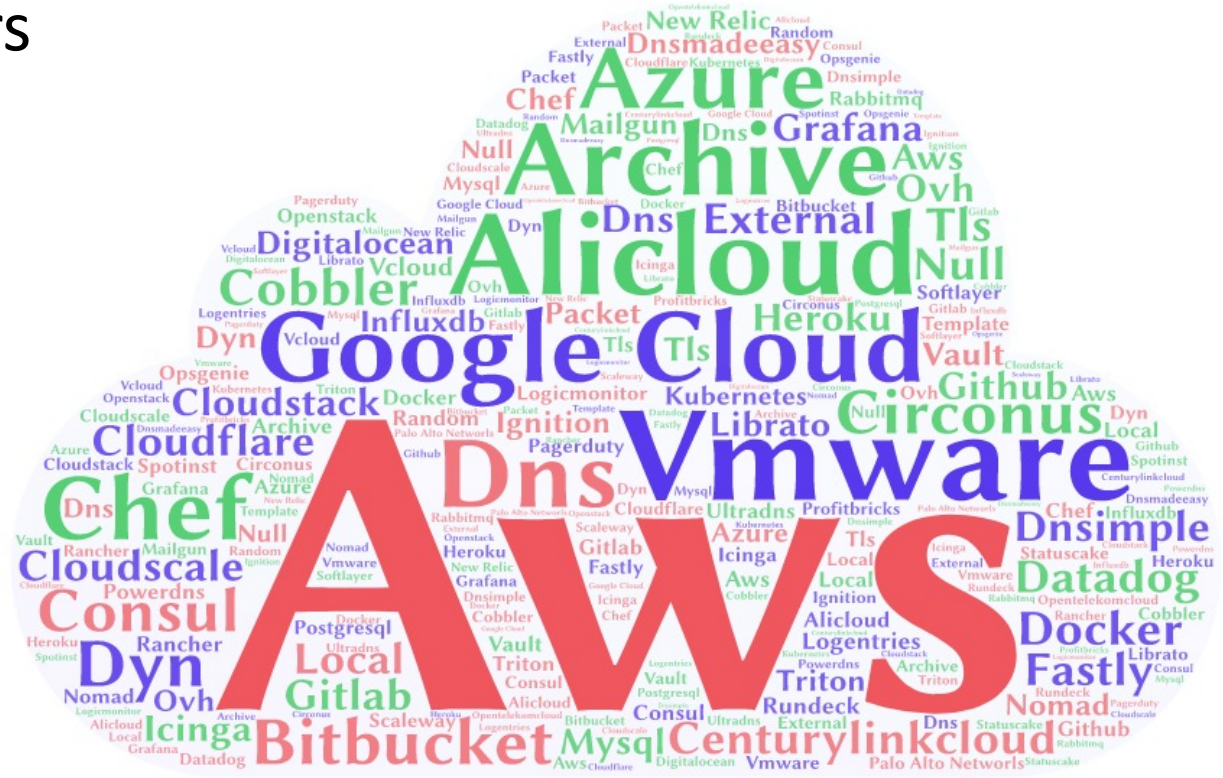
Terraform

Terraform allows to build, change and version our infrastructure in a easy and efficient way

www.terraform.io

Terraform Providers

- 120+ providers



Features

- Infrastructure as Code
- Execution Plan
- State of Infrastructure
- Dependencies
- Resource Graphs
- Allow changes to infrastructure

Benefits of Terraform

- Code reuse
- Easy management of various type of resources
- Tagging resources
- Savings – Time and $$$

Terraform commands

- init
- Plan
- apply
- destroy
- fmt
- console
- ….

https://www.terraform.io/docs/commands/index.html

Lifecycle

> terraform init

> terraform plan

> terraform apply

> terraform destroy

Terraform state

- Local State
  - On your workstation

- Remote State
  - S3
  - Consul
  - Google Cloud Storage

https://www.terraform.io/docs/state/

Setup Terraform

Install Terraform

```
$ brew install terraform
```

Test installation

```
$ terraform -help
```

https://www.terraform.io/intro/getting-started/install.html

Install awscli

Install awscli package

```
$ pip install awscli
```

Test installation

```
$ aws help
```

Configure

```
$ aws configure
AWS Access Key ID [None]: AKIAXXXXXXEXAMPLE
AWS Secret Access Key [None]:XXXXXX/XXXXXXXEXAMPLE
Default region name [None]: us-west-2
Default output format [None]: json
```

Building Infra on AWS

```
provider "aws" {
    access_key = "ACCESS_KEY_HERE"
    secret_key = "SECRET_KEY_HERE"
    region = "us-west-2"
}

resource "aws_instance" ”hello-world" {
    ami = " ami-d874e0a0"
    instance_type = "t2.micro"
}
```

Building Infra on AWS

```
provider "aws" {
    # access_key = "ACCESS_KEY_HERE"
    # secret_key = "SECRET_KEY_HERE"
    profile = ”demo"
    region = "us-west-2"
}

resource "aws_instance" ”hello-world" {
    ami = " ami-d874e0a0"
    instance_type = "t2.micro"
}
```

Other

- Terraform Interpolation: https://www.terraform.io/docs/configuration/interpolation.html

- Terraform modules: https://www.terraform.io/docs/modules/index.html

- Terraform Data Sources:
- https://www.terraform.io/docs/configuration/data-sources.html

Topics

- “Hello World” of Terraform
- Variables & outputs
- Remote execution
  - userdata
- Data source
- Remote State
- Terraform Module

Demo

github.com/pradeepbhadani/tf-techknowday-workshop

Q&A

github.com/pradeepbhadani/tf-techknowday-workshop