MAYFLOWER

Terraform & AWS

Hands-On, 10th July 2023

Welcome to Terraform

- Infrastructure as Code (**IaC**) tool
- Developed by **HashiCorp** (Vagrant, Vault, Packer)
- Build, change and version **cloud infrastructure**
- Cloud provider agnostic (e.g. AWS, Azure, Google Cloud)
- Automate infrastructure deployment, provisioning and management of cloud resources in any cloud or data center (**CI/CD**)

Imagine aims for the Cloud

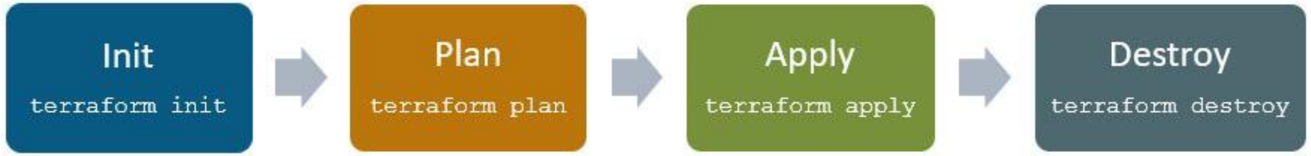
Imagine has developed several web applications for her company that are all based on different tech stacks. Now she likes to deploy these apps into the cloud and hence make them available via public IPs.

She heard of Terraform, an infrastructure as code tool, that automates provisioning of cloud infrastructure – and gets to know this tool in practice.

All acting persons are fictional

Architecture & Working Principle

- Terraform CLI
- **Declare** resources in **HCL**
  (HashiCorp Configuration Language)
- Primary CLI commands: **init**, **plan**, **apply**, **destroy**
- .terraform directory, lock file (add to VCS), state file



Example 1:
AWS Availability Zones

- Pick available AWS Availability Zones for the company AWS account
- Reference on Availability Zones
- Code Example:

availability-zones.tf

Example 2: Random Pet Names

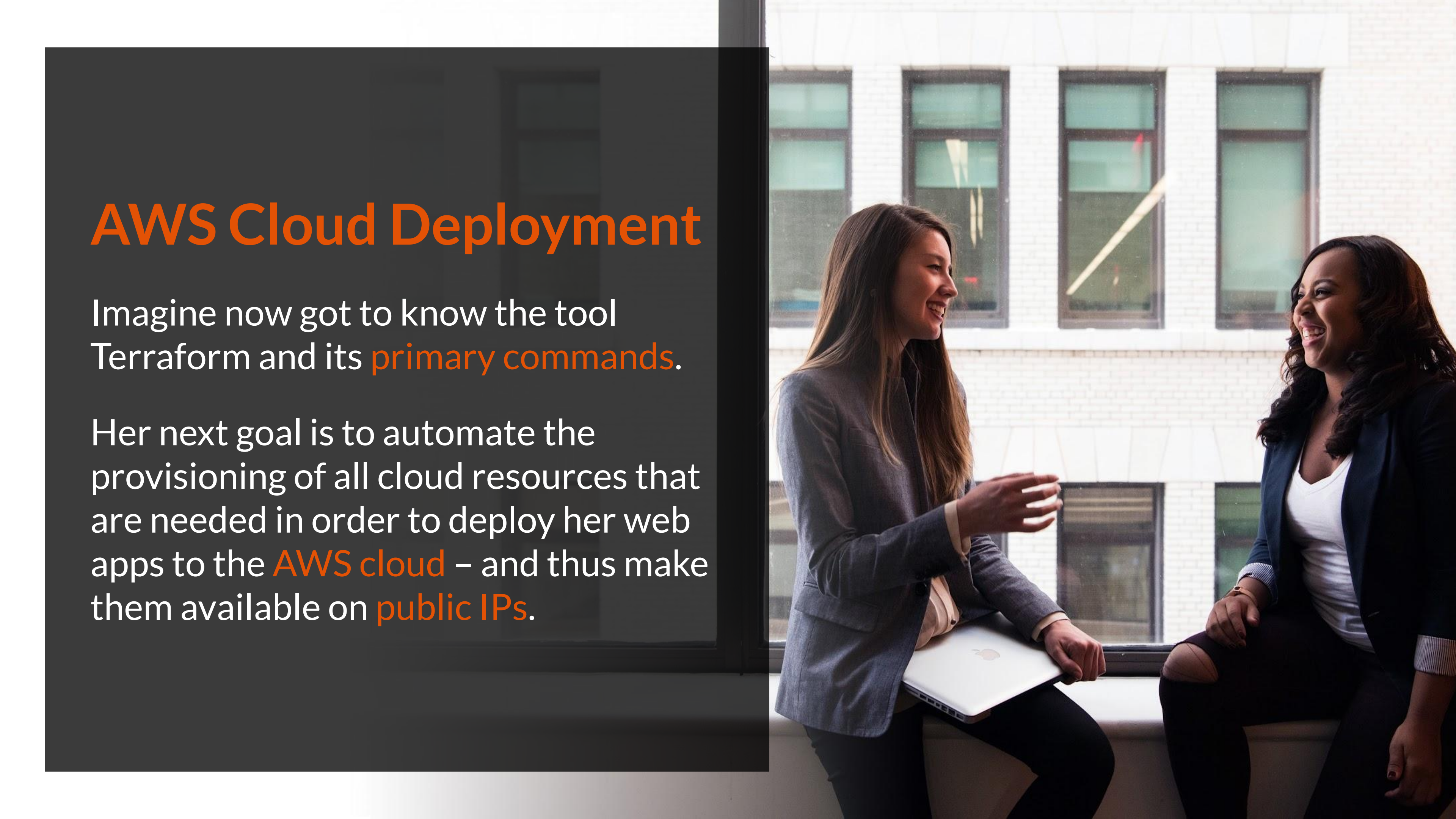
- Generate random pet names that can be assigned to future cloud resources
- Reference on Pet Resource
- Code Example:

random-pets.tf

AWS Cloud Deployment

Imagine now got to know the tool Terraform and its primary commands.

Her next goal is to automate the provisioning of all cloud resources that are needed in order to deploy her web apps to the AWS cloud – and thus make them available on public IPs.



MAYFLOWER

Terraform & AWS Hands-On

Hands-On, 10th July 2023

Hands-On: Big Picture

- Deploy the Node.js web application into a local Docker container
- Run the Docker container locally
- Operate and request the web application inside the locally running Docker container
- Deploy the Docker application container into the AWS cloud

Hands-On: Agenda

1. Requirements
   (tools and account setup)
2. Create new Terraform configuration for deploying the Node.js application container into the AWS cloud
3. More Terraform features
   (Input and Output Variables)
4. Second application container with nginx web application
5. Third application container with php-fpm that overplugs the nginx container
6. More useful Terraform commands
7. Terraform Cloud

Terraform Cloud

- Share and track cloud provisioning projects with the entire team

- Make all planned and performed changes on the cloud infrastructure transparent and accessible to the entire team

- Cloud provisioning automation is run server sided

Upsides of Terraform

Highly extensible

Developer, Würzburg

Open source and unified syntax for cloud infrastructure of different cloud providers

Developer, Munich

Quotations are for illustration purposes only

Downsides of Terraform

No automatic rollback functionality for faulty changes on cloud resources

Developer, Berlin

Supports for various cloud solutions, collaboration and security features only available in fee-paying enterprise plans

Developer, Würzburg

Quotations are for illustration purposes only

Thank you for joining the Terraform & AWS Hands-On

Sources and further Information

Terraform by HashiCorp
Terraform Cloud by HashiCorp
Amazon Web Services
GitHub Repository: Terraform AWS Workshop