Application Lifecycle with Terraform Cloud and Packer

deploying Vault

Presented by:
Puneet Sahota (he/his)
HashiCorp CSA candidate

About the Presenter: Puneet

- 11 years in the IT industry
- Multiple Clouds experience
- Development and Sales roles
- Avid Cyclist
- Nature Lover
- Occasional Swimmer

Assumptions

- Audience has some knowledge of Terraform OSS
- The Organization is migrating workloads to Cloud
- The team has concerns about security management
- The Organization is a large Media Company

Agenda

What are we solving today?

- Cloud migration journey for the organization
- Cloud collaboration capabilities for the team
- Golden Image creation process for VMs in Cloud

How HashiCorp is going to help you?

- Accelerate your Cloud Journey with Terraform Cloud
- Highly efficient Cloud Image creation process using Packer
- Centralize Secret management using HashiCorp Vault

Let’s solve the problem

Solution: Use case

Use Case:

- Build a Golden AMI with Packer and deploy vault application.
- Use Github Pipeline (Actions) to trigger the packer build on each merge.
- Use Terraform Cloud VCS to deploy Ec2 using the latest Packer Image.

Products used:

- HashiCorp Packer- HashiCorp's Image Build tool
- Terraform Cloud- Managed Terraform with UI
- HashiCorp Vault- We are deploying a vault application in dev mode

HashiCorp Packer:

Create Golden Images for multiple platform

Using single source configuration.

Current Market Challenges:

- Multiple OS and Cloud Platforms.
- Longer time to market for new Images.
- Difficult to manage with DevOps.
- Process compromises Image Security.
- Longer time to deploy the security updates.

HashiCorp Packer: Benefits

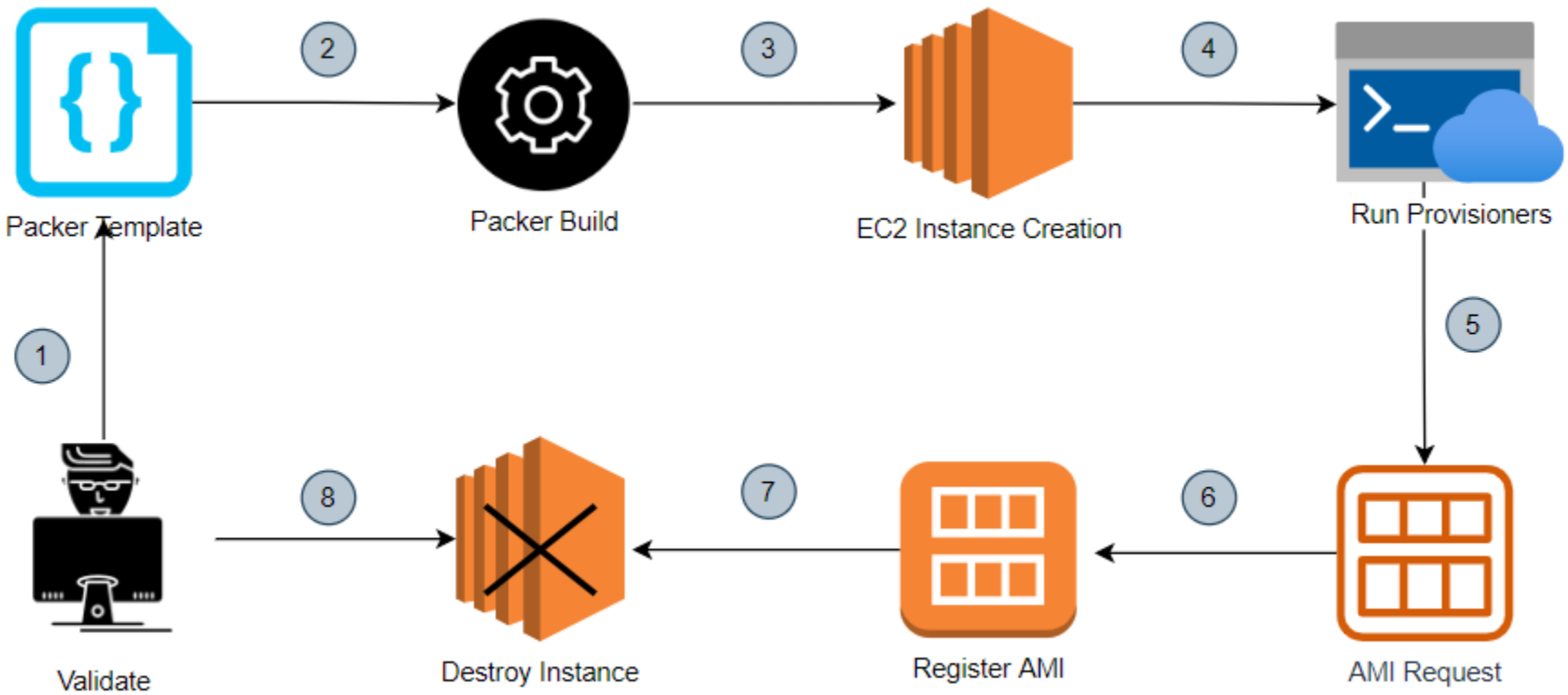
- Version Controlled Images
- Cross Platform consistency
- Automated Image builds
- Increased Dev/Production parity
- Plugins for extending functionality

HashiCorp Packer: Use Cases

- Create Golden Images across Platforms and Environments
- Image factory based on new Commits for Continuous Delivery
- Automate Monthly Patching for New/Existing Workloads
- Immutable Infrastructure in CI/CD Pipelines

HashiCorp Packer: Lifecycle

Packer



HashiCorp Packer: Building Blocks

Packer

- Building Blocks:
  - Source

```
source "amazon-ebs" "PackerGoldenImage" {
  ami_name      = "packer-aws-ubuntu-{{timestamp}}"
  source_ami    = "ami-02541b8af977f6cdd"
  instance_type = "t2.micro"
  region        = var.region
  source_ami_filter {
  owners        = ["amazon"]
  most_recent   = true
  }
  ssh_username = "ec2-user"
}
```

  - Builders

```
build {
  sources = ["source.amazon-ebs.PackerGoldenImage"]
  }
```

  - Variables

```
variable "region"{
    type = string
    default = "us-west-1"
}
```

  - Provisioners

```
  provisioner "file" {
    source      = "hello.txt"
    destination = "/tmp/hello.txt"
  }
```

  - Post processor

```
  post-processor "shell-local" {
    inline = ["echo foo"]
  }
```

HashiCorp Packer: Basic Commands

- Formatting

```
PS C:\PackerCodeBase> packer fmt .\aws-packer.pkr.hcl
```

- Autocomplete

```
PS C:\PackerCodeBase> packer -autocomplete-install
```

- Validating

```
PS C:\PackerCodeBase> packer validate .\aws-packer.pkr.hcl
The configuration is valid.
```

- Building

```
PS C:\PackerCodeBase> packer build .\aws-packer.pkr.hcl
amazon-ebs.amazonImage: output will be in this color.

==> amazon-ebs.amazonImage: Prevalidating any provided VPC info
==> amazon-ebs.amazonImage: Prevalidating AMI Name: packer-aws-
```

- Logging

```
PS C:\PackerCodeBase> $env:PACKER_LOG=1
PS C:\PackerCodeBase> $env:PACKER_LOG_PATH="packerlog.txt"
```

Demo 1: Pre-requisites

- AWS Account
- GitHub Account
- Packer 1.7.10 installed locally

Demo 1
HashiCorp Packer Lifecycle

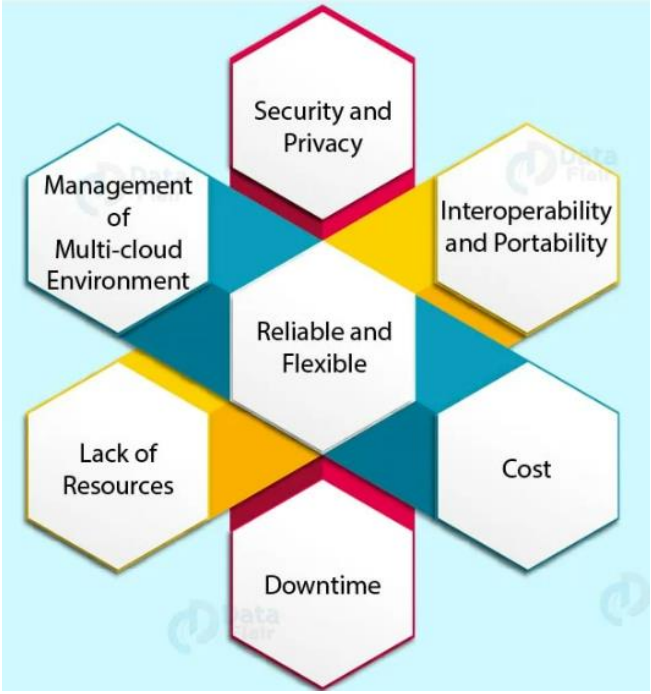
Terraform Cloud:

Online self-service provisioning

and collaboration of cloud resources.

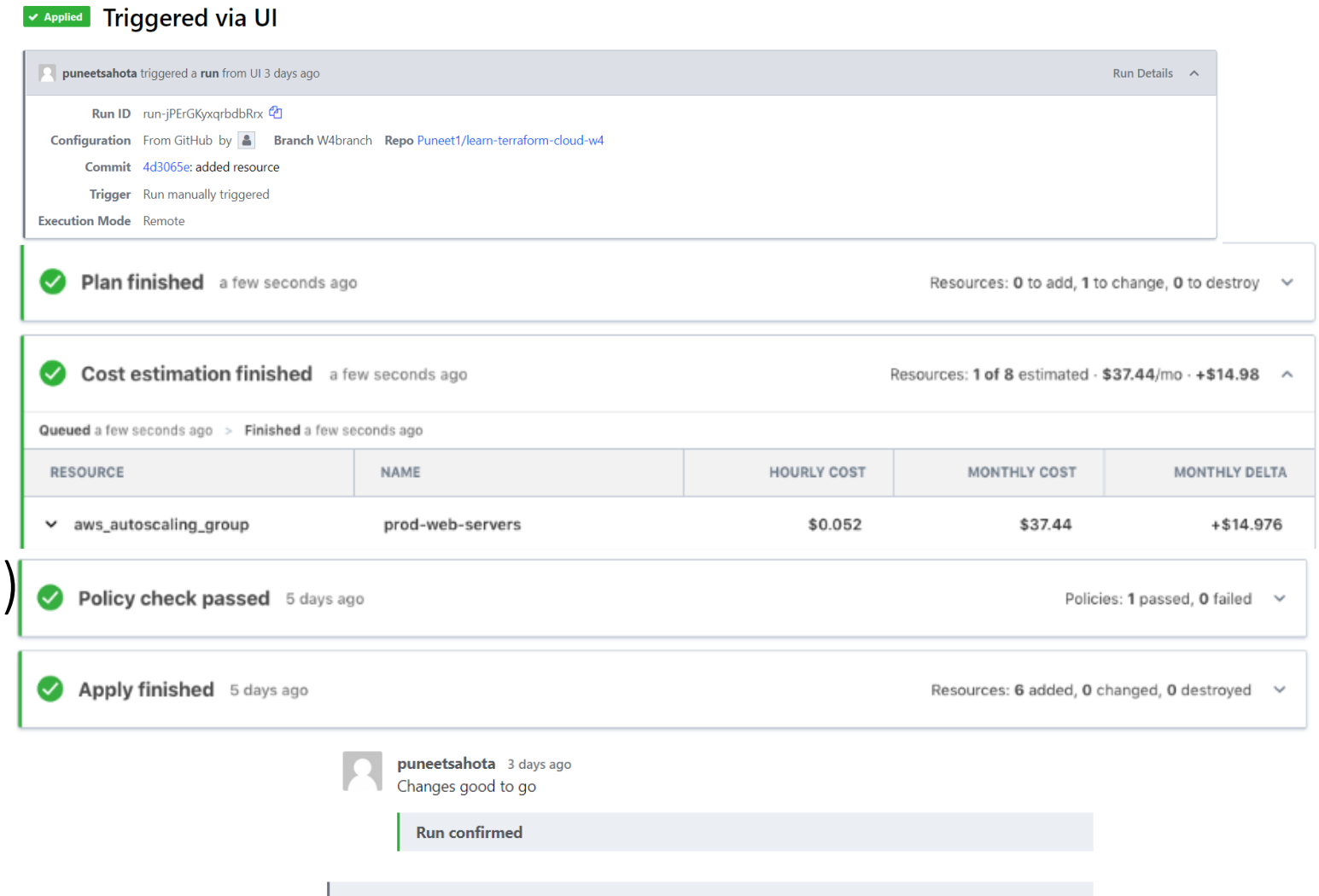
Challenges with current IaC tools

- Not designed to manage hybrid and multi–Cloud environments
- High cost of training for the Infrastructure/SRE/DevOps teams
- Reduced efficiency due to constant shifting among platforms
- Steep learning curve for the developers
- Management Overhead
- Code consistency Issues
- Cloud Costs



Terraform Cloud vs Terraform open-source

- Consistent Environment
- UI based
- VCS Connection
- Estimate cost
- Enforce Policy checks (p)
- Collaborative Runs
- Private Module Registry(p)



Terraform Cloud Workflow

Terraform Cloud: Speed and Agility

- Increase release velocity Up to 5x
- Increased Saving with Collaboration Up to 30%
- Increase IT Ops productivity Up to 75%
- 1000+ providers
- 100+ partners
- **Case Study-** Red-Ventures: https://www.hashicorp.com/case-studies/red-ventures
  - 1500 workspaces across 300+ Teams
  - 400K+ Successful Code Runs
  - $20K saved in Operating Costs Every Month
  - 700+ developers spanning 25+ Business Groups

Demo 2: Pre-requisites

- Terraform Cloud Account
- AWS Account
- GitHub Account

Demo 2
Terraform Cloud: VCS Integration

Terraform Cloud Plans

Free

Cloud

Free

Sign up

OPEN SOURCE FEATURES, PLUS:

State management

Remote operations

Private module registry

Community support

Team & Governance

Cloud

STARTING AT **$20 / user**

Sign up

EVERYTHING IN FREE, WITH OPTIONS TO ADD:

Team management

Sentinel policy as code

Run tasks

Policy enforcement

Bronze support

Business

Cloud

Contact sales

EVERYTHING IN TEAM & GOVERNANCE, PLUS:

SSO, self-hosted agents, audit logs

Custom concurrency

Self-hosted option

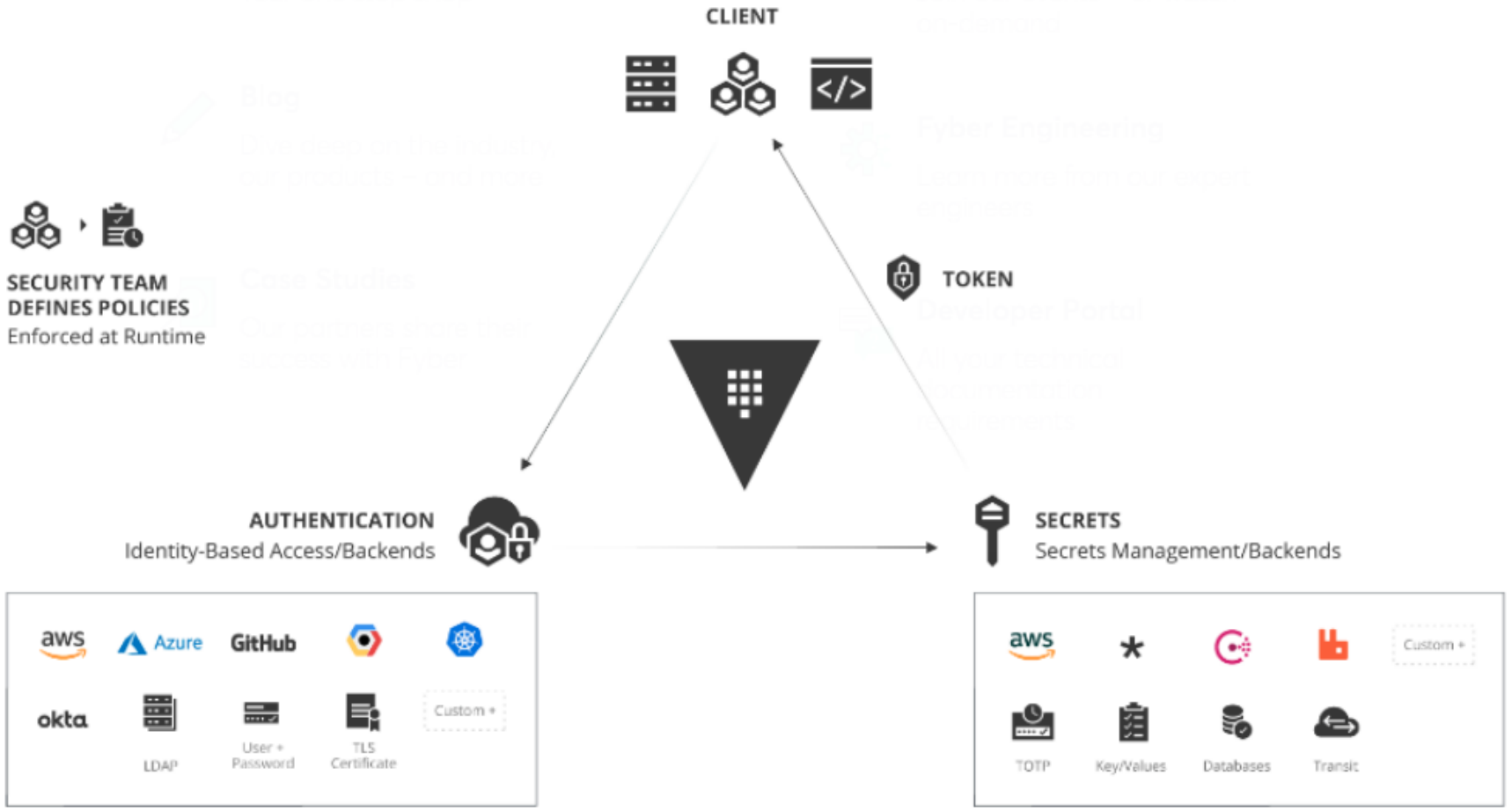
Bronze, Silver, or Gold support

Terraform

HashiCorp Vault:

Manage secrets and Protect sensitive data

High Level Architecture



Features of HashiCorp Vault (Cloud)

- Prevents Secrets Sprawl
- Secrets Storage
- Dynamic Secrets
- Automated Credential rotation
- Encryption as a Service
- API driven Encryption
- Encryption Key rolling
- Plugins with Multiple Cloud Backends

Helpful Information: Next steps

- Github link: https://github.com/Puneet1/PackerGithub_v1
- Explore- https://learn.hashicorp.com/
- Get Help- https://support.hashicorp.com/hc/en-us
- Contact me- Puneet.Sahota@hashicorp.com

Questions?

Thank you