HashiTalks

hugs@hashicorp.com | learn.hashicorp.com | discuss.hashicorp.com

Streamlining infrastructure deployment

With Packer and Terraform

Keerthivasan Kannan

Sr. Cloud Engineer at Mindera

@keerthivasan-kannan


aws certified
Solutions Architect
PROFESSIONAL
aws certified
DevOps Engineer
PROFESSIONAL
Terraform
ASSOCIATE
HashiCorp
CERTIFIED
aws certified
Solutions Architect
ASSOCIATE
aws certified
SysOps Administrator
ASSOCIATE
aws certified
Developer
Associate
aws certified
Security
SPECIALTY


e-vidhai


TECH
PECHU

Agenda

Infrastructure Intro

Creating vs Configure

Terraform for IaC

Packer to Configure

Streamline it in pipeline

Best Practices

_Infrastructure

Infra As Code

Track your change

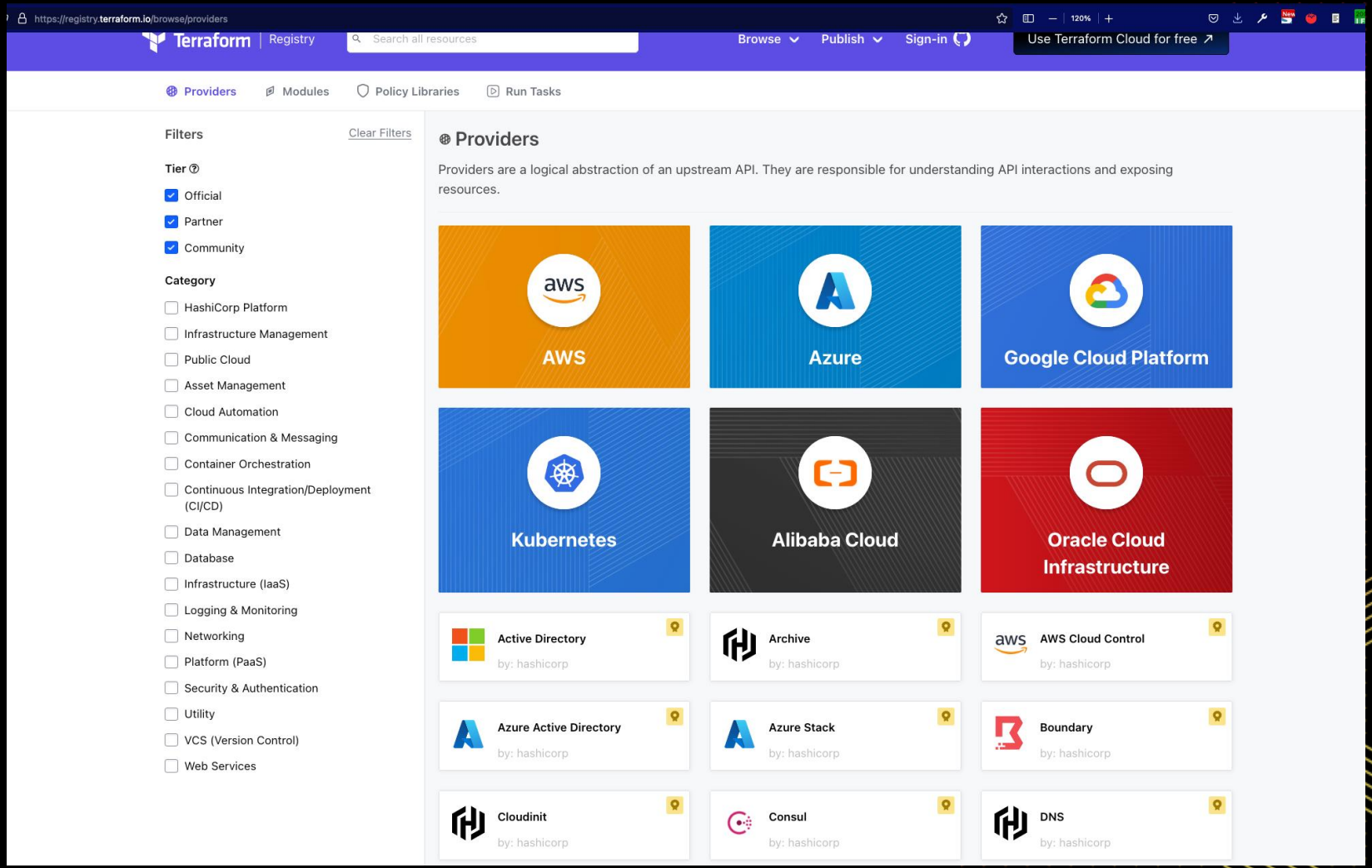
Quick Replication

Automate

Terraform

Supports Multicloud

Not limited to Cloud


https://registry.terraform.io/browse/providers
Terraform | Registry
Search all resources
Browse
Publish
Sign-in
Use Terraform Cloud for free
Providers
Modules
Policy Libraries
Run Tasks
Filters
Clear Filters
Tier
Official
Partner
Community
Category
HashiCorp Platform
Infrastructure Management
Public Cloud
Asset Management
Cloud Automation
Communication & Messaging
Container Orchestration
Continuous Integration/Deployment (CI/CD)
Data Management
Database
Infrastructure (IaaS)
Logging & Monitoring
Networking
Platform (PaaS)
Security & Authentication
Utility
VCS (Version Control)
Web Services
Providers
Providers are a logical abstraction of an upstream API. They are responsible for understanding API interactions and exposing resources.
AWS
Azure
Google Cloud Platform
Kubernetes
Alibaba Cloud
Oracle Cloud Infrastructure
Active Directory
by: hashicorp
Archive
by: hashicorp
AWS Cloud Control
by: hashicorp
Azure Active Directory
by: hashicorp
Azure Stack
by: hashicorp
Boundary
by: hashicorp
Cloudinit
by: hashicorp
Consul
by: hashicorp
DNS
by: hashicorp

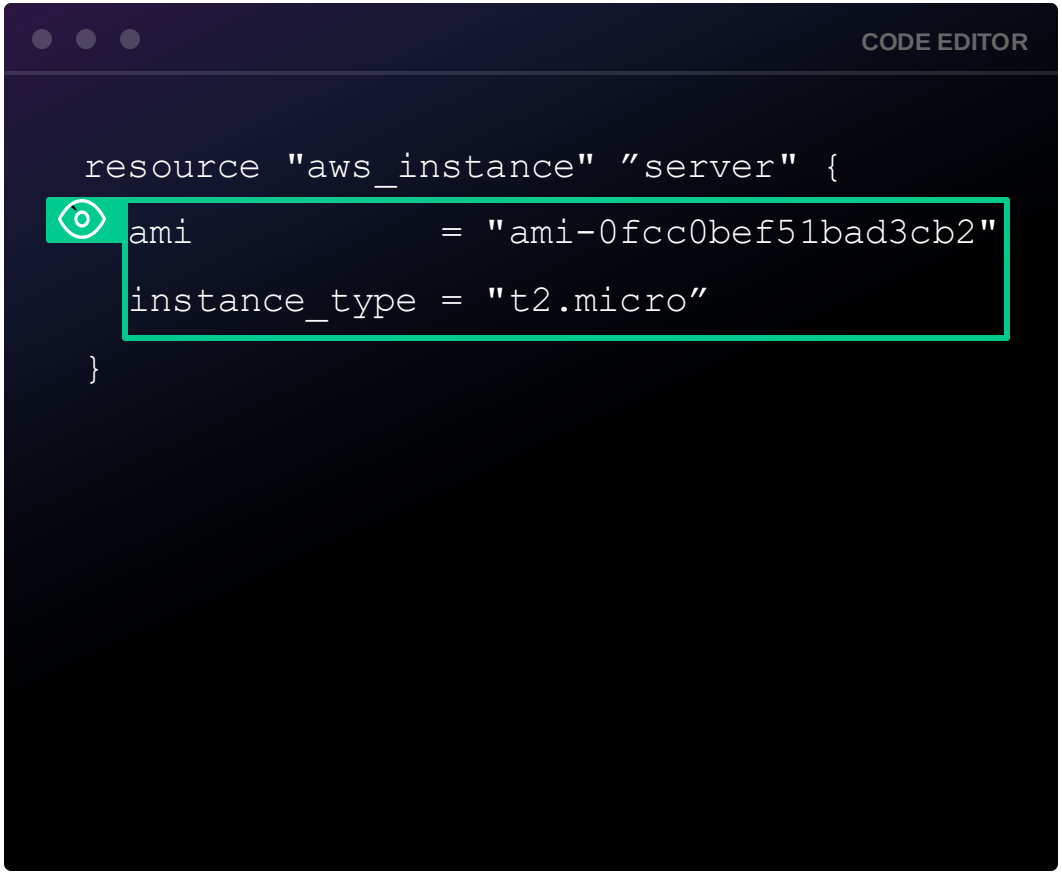
_Infrastructure

_Infrastructure

_Infrastructure

Create

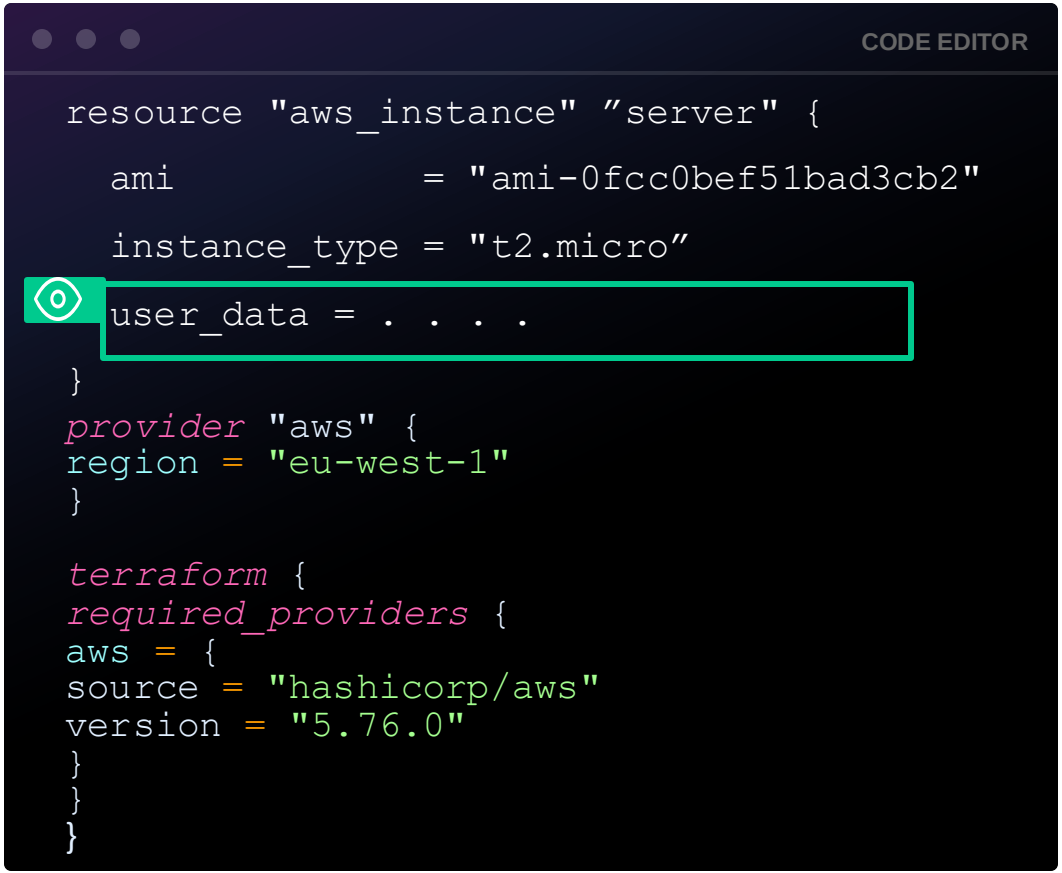
Configure


CODE EDITOR
resource "aws_instance" ”server" {
ami = "ami-0fcc0bef51bad3cb2"
instance_type = "t2.micro”
}

Terraform


CODE EDITOR
resource "aws_instance" ”server" {
ami = "ami-0fcc0bef51bad3cb2"
instance_type = "t2.micro"
}
provider "aws" {
region = "eu-west-1"
}
terraform {
required_providers {
aws = {
source = "hashicorp/aws"
version = "5.76.0"
}
}
}

Terraform


CODE EDITOR
resource "aws_instance" ”server" {
ami = "ami-0fcc0bef51bad3cb2"
instance_type = "t2.micro”
user_data = ...
}
provider "aws" {
region = "eu-west-1"
}
terraform {
required_providers {
aws = {
source = "hashicorp/aws"
version = "5.76.0"
}
}
}

Terraform

Demo

CODE EDITOR

```
user_data = <<-EOF
#!/bin/bash
yum update -y
yum install -y httpd
systemctl start httpd
systemctl enable httpd
echo "Hello, Apache is running!" >
/var/www/html/index.html
EOF
```

User Data

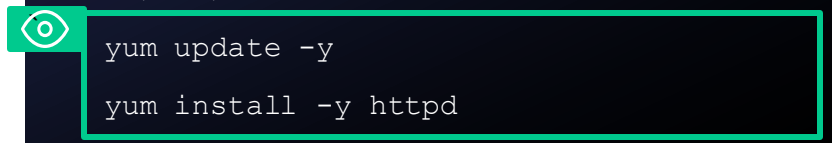
CODE EDITOR

```
user_data = <<-EOF
#!/bin/bash
yum update -y
yum install -y httpd
systemctl start httpd
systemctl enable httpd
echo "Hello, Apache is running!" >
/var/www/html/index.html
EOF
```

User Data

CODE EDITOR

```
user_data = <<-EOF
#!/bin/bash
yum update -y
yum install -y httpd
systemctl start httpd
systemctl enable httpd
echo "Hello, Apache is running!" >
/var/www/html/index.html
EOF
```



User Data

Packer

CODE EDITOR

```
packer {
  required_plugins {
    amazon = {
      version = ">= 1.2.8"
      source  =
"github.com/hashicorp/amazon"
    }
  }
}
```

Packer

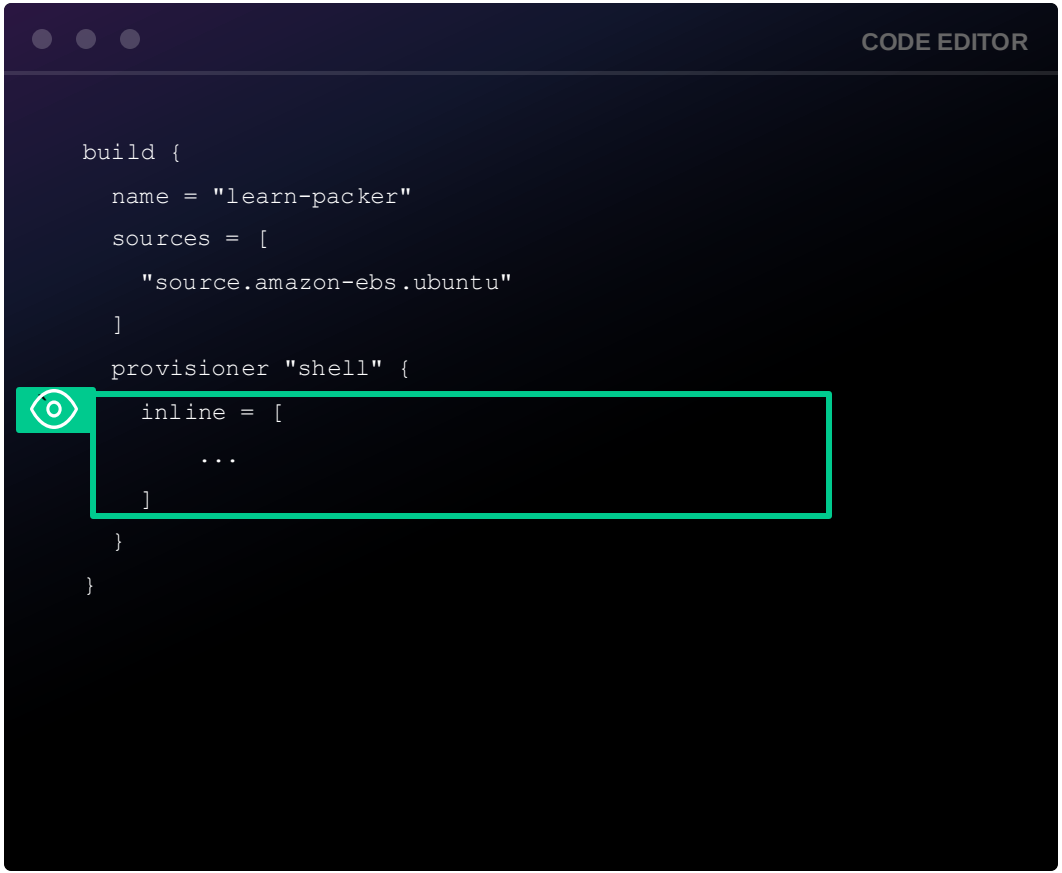
CODE EDITOR

```
source "amazon-ebs" "ubuntu" {
  ami_name      = "vasan-packer"
  instance_type = "t2.micro"
  region        = "us-west-2"
  source_ami_filter {
    filters = {
      name                = "ubuntu/images/*ubuntu-jammy-22.04-
amd64-server-*"
      root-device-type    = "ebs"
      virtualization-type = "hvm"
    }
    most_recent = true
    owners      = ["099720109477"]
  }
  ssh_username = "ubuntu"
}
```

Packer


CODE EDITOR
source "amazon-ebs" "ubuntu" {
ami_name = "vasan-packer"
instance_type = "t2.micro"
region = "us-west-2"
source_ami_filter {
filters = {
name = "ubuntu/images/*ubuntu-jammy-22.04-
amd64-server-*"
root-device-type = "ebs"
virtualization-type = "hvm"
}
most_recent = true
owners = ["099720109477"]
}
ssh_username = "ubuntu"
}

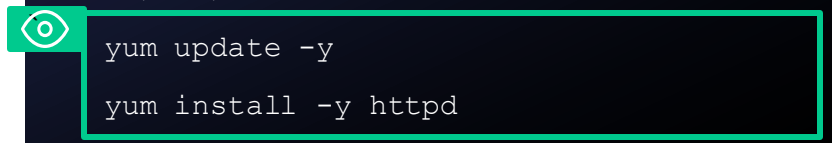
Packer


CODE EDITOR
build {
name = "learn-packer"
sources = [
"source.amazon-ebs.ubuntu"
]
provisioner "shell" {
inline = [
...
]
}
}

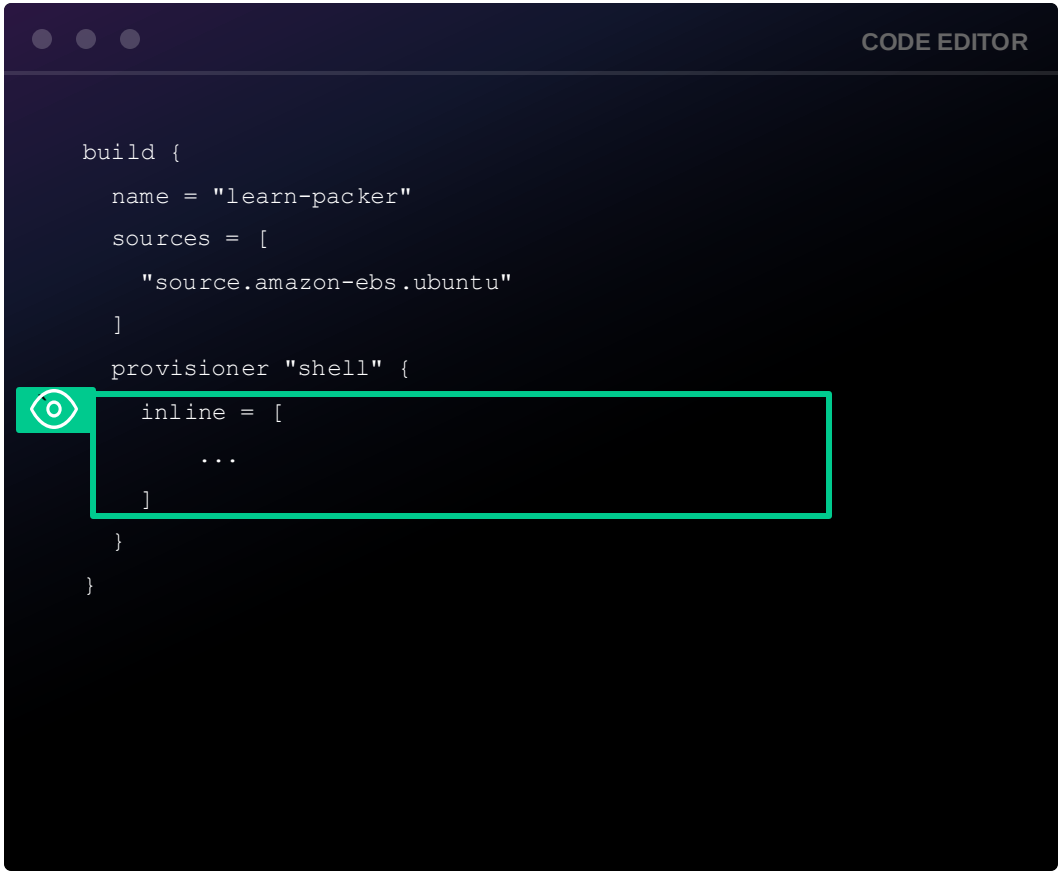
Packer

CODE EDITOR

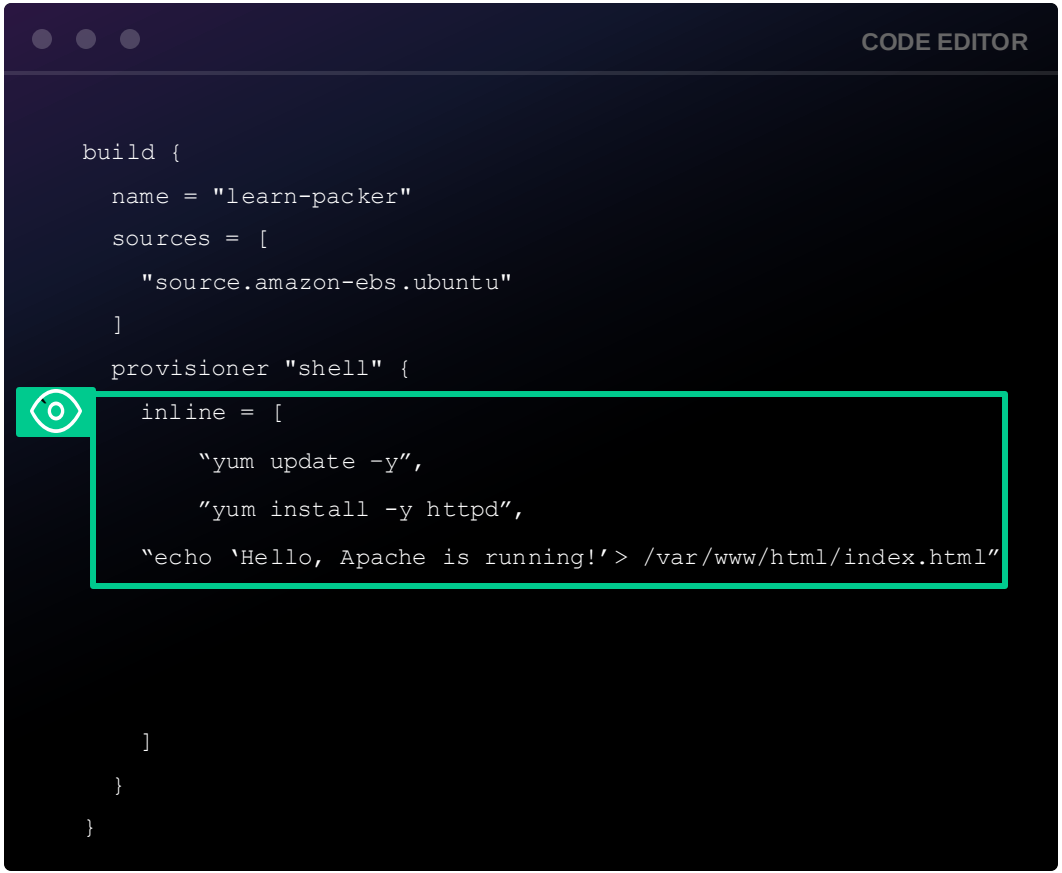
```
user_data = <<-EOF
#!/bin/bash
yum update -y
yum install -y httpd
systemctl start httpd
systemctl enable httpd
echo "Hello, Apache is running!" >
/var/www/html/index.html
EOF
```



User Data


CODE EDITOR
build {
name = "learn-packer"
sources = [
"source.amazon-ebs.ubuntu"
]
provisioner "shell" {
inline = [
...
]
}
}

Packer


CODE EDITOR
build {
name = "learn-packer"
sources = [
"source.amazon-ebs.ubuntu"
]
provisioner "shell" {
inline = [
“yum update -y”,
”yum install -y httpd”,
“echo ‘Hello, Apache is running!’> /var/www/html/index.html”
]
}
}

Packer

Demo

Best Practices

Keep AMI Centralised

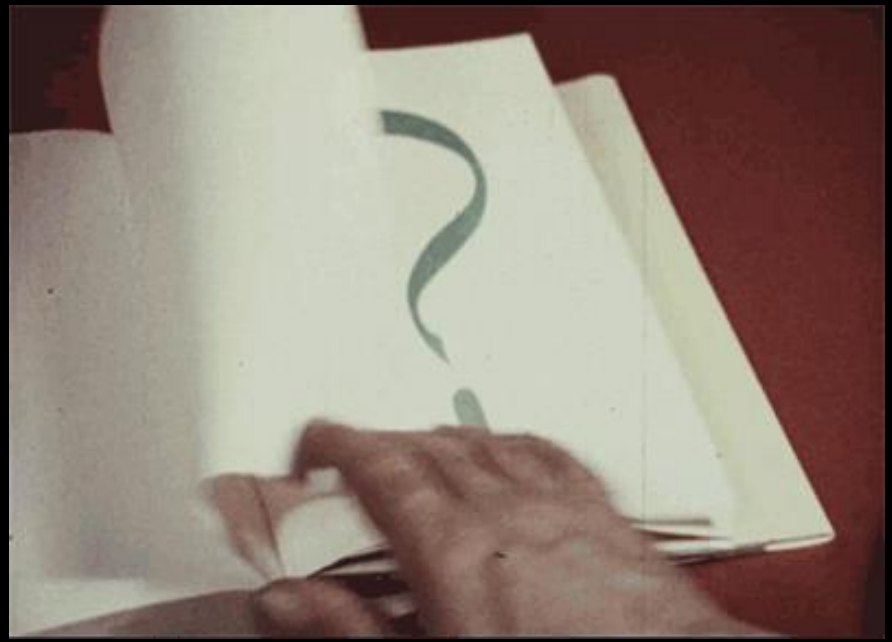
To avoid issues Containerise your ami process

Use Tags on resources for cost tracking

Follow DRY

https://github.com/evidhai/infra-streamline/

Questions



bitly

THANK YOU

Keerthivasan Kannan

Sr. Cloud Engineer at Mindera

@keerthivasan-kannan