Cloud **Terraformation**

FROM CLICKOPS TO INFRASTRUCTURE AS CODE

Agenda

ABOUT

THE LANGUAGE

THE PROVIDER

THE ASSIGNMENT

QUESTIONS

Hello from Grand Rapids!

I'm Victor Frye

Your friendly neighborhood developer

Unchallenged #1 Clippy fan

Love the internet and fun
- victorfrye.com/blog
- microsoftgraveyard.com
- shrugman.com

Passion for .NET, React, Azure, Xbox

DevOps specialist at Leading EDJE

Building with Microsoft solutions

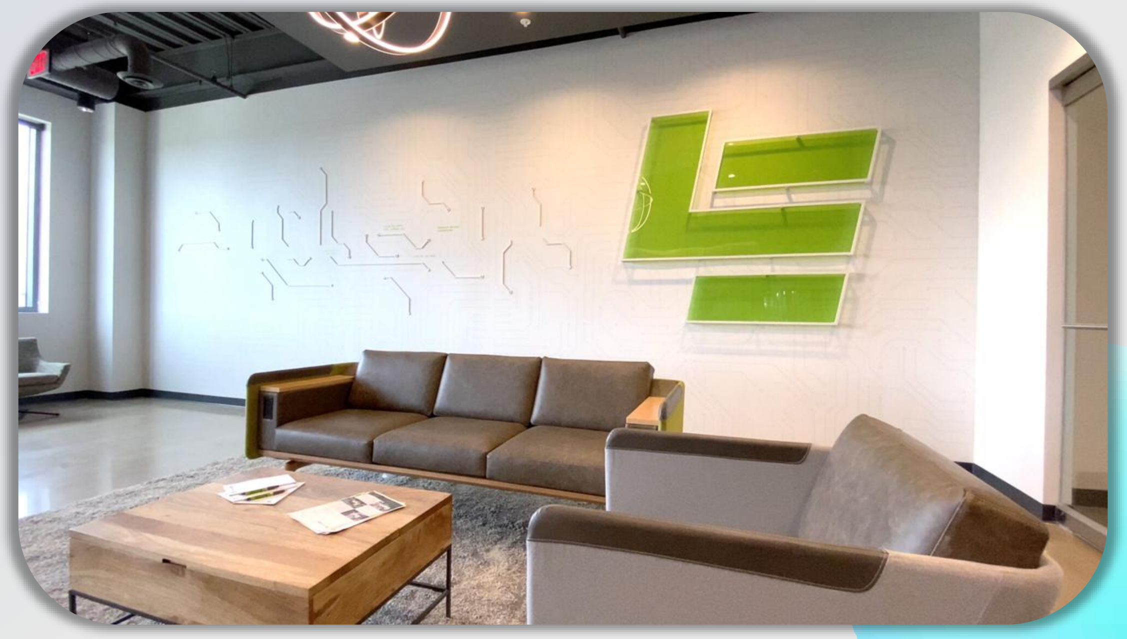
About **Leading EDJE**

Technology consultancy and services
- Strategists, innovators, partners
- Craft bespoke solutions
- Positive disruption of business norm

Real. Fun. Geeks.

FREE lunch and learns

Expert talent
- Cloud infrastructure
- Digital and app innovation
- Data and AI
- Project management



The **ClickOps story**

Build application

What infrastructure?

Click, click, deploy

Cross fingers it works

Please do not change it

The Answer

CODIFY INFRASTRUCTURE

The Language

INFRASTRUCTURE AS CODE

HashiCorp Terraform

An infrastructure as code tool

Source-available

Cloud-agnostic

Declarative

Stateful

Uses Terraform configuration language (HCL)

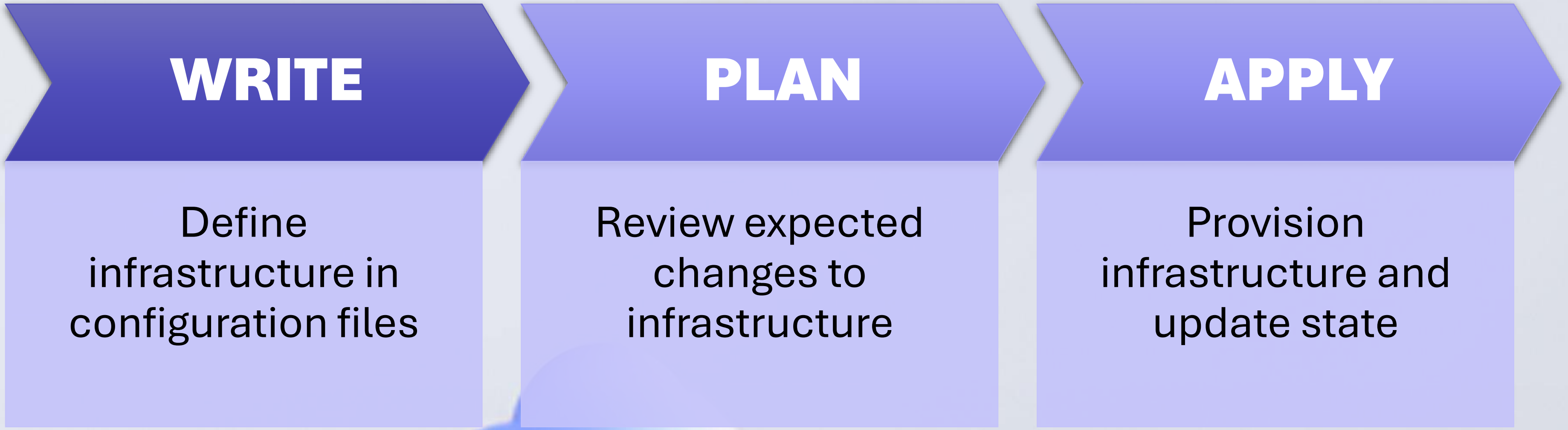
Elements include:
- Blocks (e.g., resources, variables, data)
- Arguments (e.g., name = “value”)
- Expressions (e.g., name = var.example)

```
resource "azurerm_resource_group" "example" {
  name     = "example-resources"
  location = "Central US"
}

resource "azurerm_static_web_app" "example" {
  name                = "example"
  resource_group_name = azurerm_resource_group.example.name
  location            = azurerm_resource_group.example.location
}
```

EXAMPLE CONFIGURATION FOR AN AZURE STATIC WEB APP

Write, Plan, Apply



WRITE

Define infrastructure in configuration files

PLAN

Review expected changes to infrastructure

APPLY

Provision infrastructure and update state

Alternative options

- Bicep
- CloudFormation
- Pulumi
- Ansible

The Provider

YOUR CLOUD SPECIFIC INTERFACE

Multiple providers

- AWS
- AzureRM
- Google
- Kubernetes
- AzAPI
- AzureAD
- MSGraph
- GitHub

Choose your provider

AzureRM

- Automatic API versioning
- Simplicity
- Comprehensive documentation

AzAPI

- Immediate API access
- Targeted resource updates
- Fine-grained control

Terraforming Azure

ClickOps resources in the Azure portal

Configure workload identity federation with Entra ID

Set up Terraform backend with blob storage

Use template exports for Terraform

Automate CI/CD with GitHub Actions

The Assignment

READY, SET, GO!

Install Terraform

DEMONSTRATION

Set up identity

DEMONSTRATION

Create backend

DEMONSTRATION

Configure RBAC

DEMONSTRATION

Export Terraform

DEMONSTRATION

Refactor configurations

DEMONSTRATION

Import state

DEMONSTRATION

Build plan

DEMONSTRATION

Apply plan

DEMONSTRATION

Automate CI/CD

DEMONSTRATION

Did it work?

SHRUGMAN.COM

Thank you

VICTOR FRYE
616-706-7407
VICTORFRYE@OUTLOOK.COM
VICTOR.FRYE@LEADINGEDJE.COM
LEADINGEDJE.COM

VICTORFRYE.COM/BLOG
LINKEDIN.COM/IN/VICTORFRYE
GITHUB.COM/VICTORFRYE/PRESENTATIONS