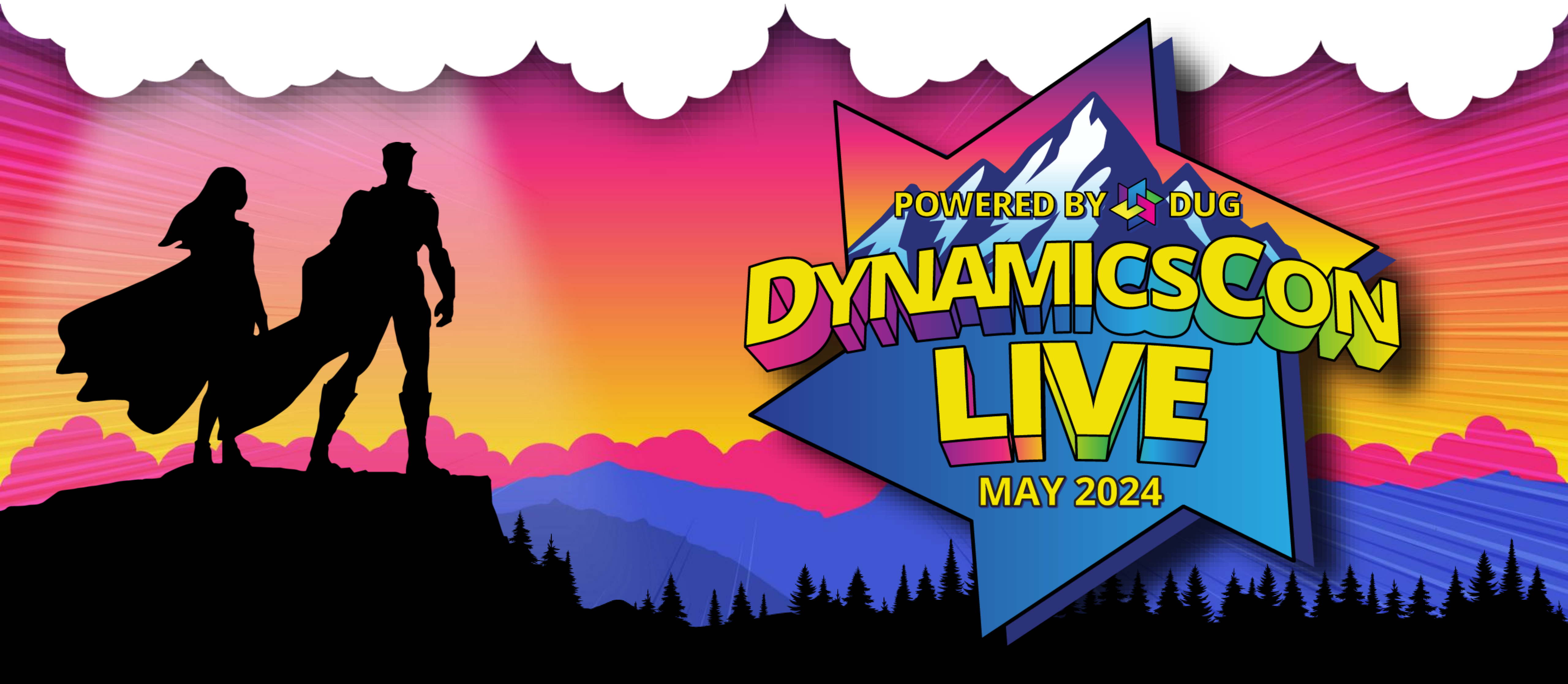


POWERED BY DUG

# DYNAMICSCON LIVE

MAY 2024

live.dynamicscon.com

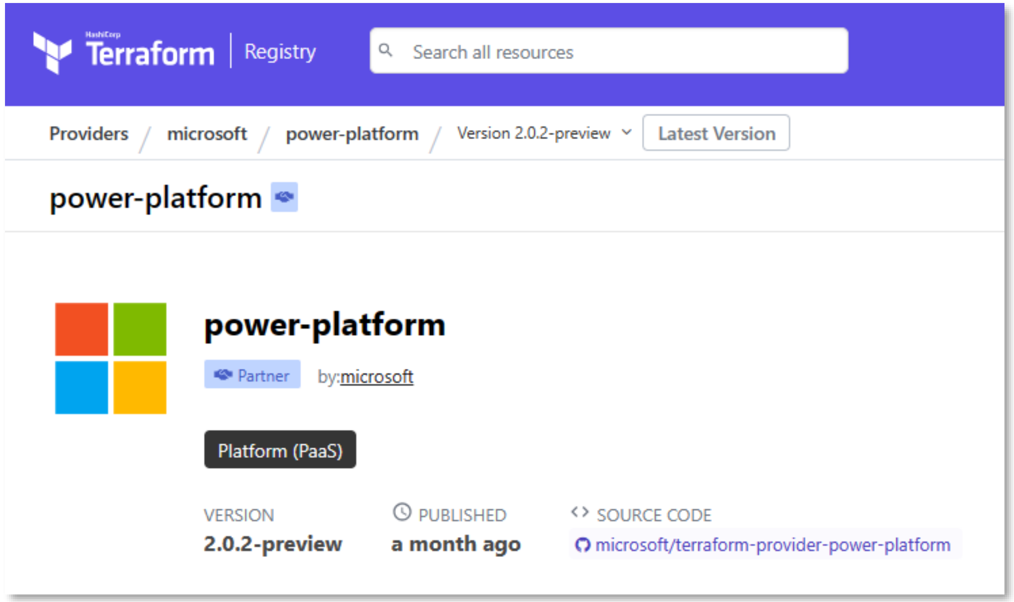
# Business Applications Infrastructure as Code

## (Microsoft's Power Platform Terraform Provider)

Ian Jensen
Microsoft
Industry
Solutions
Engineering

# Introduction

The Power Platform Terraform Provider is an Infrastructure-as-Code (IaC) toolset that enables scalable, repeatable, and observable infrastructure management for D365 and Power Platform infrastructure using the popular Terraform IaC framework.


HashiCorp Terraform | Registry
Search all resources
Providers / microsoft / power-platform / Version 2.0.2-preview
Latest Version

power-platform

power-platform

Partner by:microsoft

Platform (PaaS)

VERSION
2.0.2-preview

PUBLISHED
a month ago

SOURCE CODE
microsoft/terraform-provider-power-platform

# Agenda

- □ Context (there’s a lot, but it’s worth it)
- □ Demo

# Background: Infrastructure as Code (IaC)

Infrastructure as Code (IaC) is a methodology that involves managing and provisioning technical infrastructure through human-readable script files, rather than through manual workflows or interactive configuration tools. IaC tools enable the **scalable automation** of infrastructure deployment, ensuring **consistency** and **repeatability** while treating infrastructure configurations as version-controlled code.

# Why Terraform?

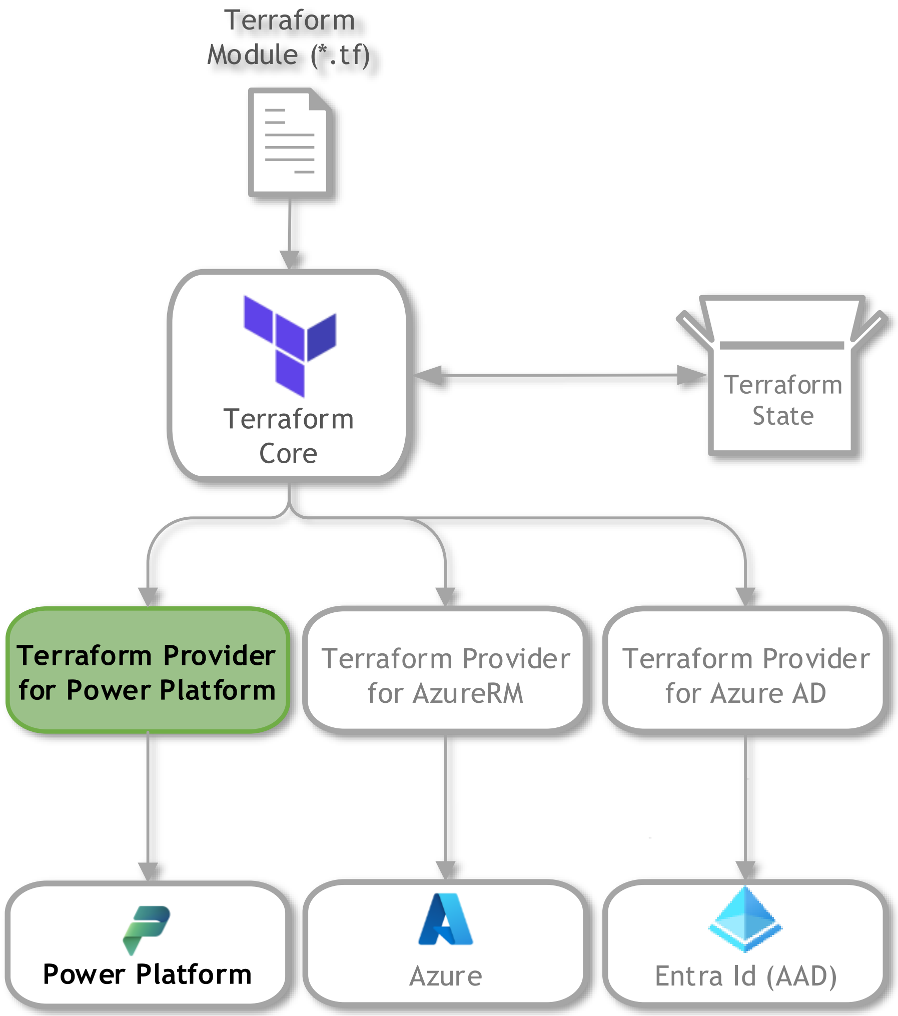
HashiCorp Terraform

ANSIBLE

CHEF

puppet

Microsoft Azure Resource Manager



Terraform Module (*.tf)
Terraform Core
Terraform State
Terraform Provider for Power Platform
Terraform Provider for AzureRM
Terraform Provider for Azure AD
Power Platform
Azure
Entra Id (AAD)

# Basic Terraform Workflow*

## Write

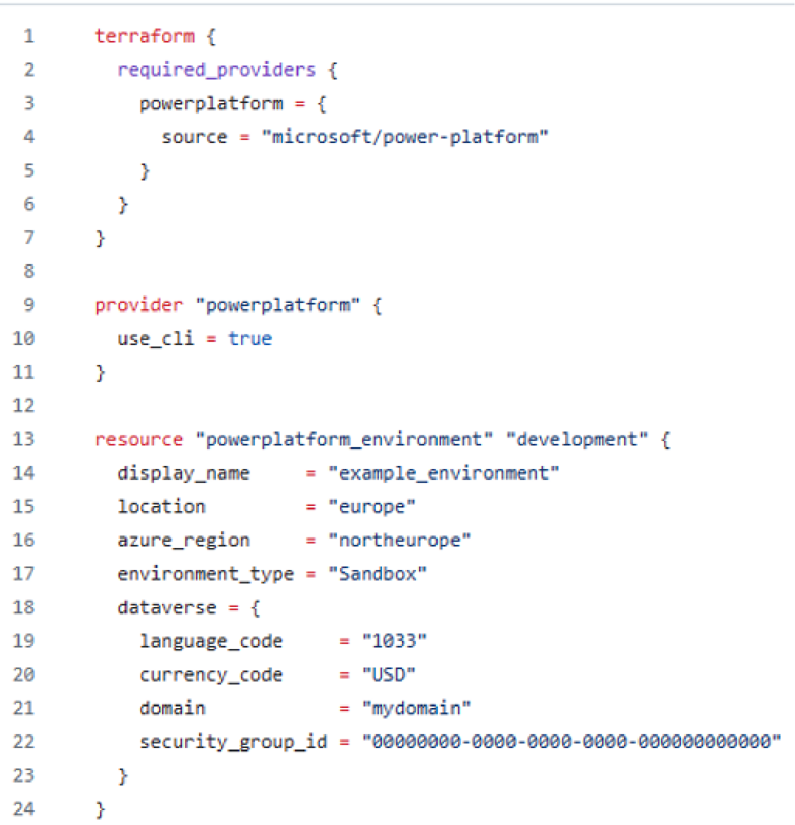
- Define infrastructure **declaratively** in configuration files

## Plan

- Terraform generates an execution plan describing actions it will take based on existing infrastructure and your configuration.
- A state "delta" will automatically be generated for your review

## Apply

- On approval, Terraform performs the proposed operations in the correct order, respecting any resource dependencies.
- Save the Terraform state output from the 'Apply' step, as this is the current known state of your infrastructure
- **SECURE** the file output from the 'Apply' step as sensitive information such as GUIDs, domains, etc. are often included in the Terraform state file.



```
terraform {
  required_providers {
    powerplatform = {
      source = "microsoft/power-platform"
    }
  }
}

provider "powerplatform" {
  use_cli = true
}

resource "powerplatform_environment" "development" {
  display_name     = "example_environment"
  location         = "europe"
  azure_region     = "northeurope"
  environment_type = "Sandbox"
  dataverse = {
    language_code     = "1033"
    currency_code     = "USD"
    domain            = "mydomain"
    security_group_id = "00000000-0000-0000-0000-000000000000"
  }
}
```

# Common Challenges

Enterprises can’t expand Power Platform usage beyond their ability to manage and govern the environments they currently have because “ClickOps” doesn’t scale.

Power Platform is treated as an exception or not used in solutions because enterprise policies require using industry standard IaC tools and processes for production cloud deployments.

ISVs deploy complex software built on Power Platform and Dynamics into customers’, partners’, dealerships’, and franchisees’ tenants and setting up the infrastructure for that software adds cost and latency to onboarding new business

Development teams struggle to build solutions that combine Power Platform and Azure (like Copilots) because infrastructure and ALM tooling is vastly different between the platforms

# Use Cases

**Cross-cloud Deployment**

Utilize existing IaC skillsets to manage infrastructure across Power Platform, D365,  Azure, and third-party clouds.  Modern solutions like Copilots often include resources that span Power Platform and Azure.

**Governance & Compliance**

Establish processes and policies to mitigate risk.  Identify and rectify infrastructure drift, use policy to ensure compliance, approve modifications, and keep track of update history.  Use service principal deployments to implement least-privilege administration.

**Infrastructure Deployment at Scale**

Automate complex deployments, templatize repetitive tasks, and apply changes across many environments.  Provision environments for application dev teams (Dev, Test, Prod, Pipelines, etc.) or common use cases like copilots.

**Infrastructure for ALM**

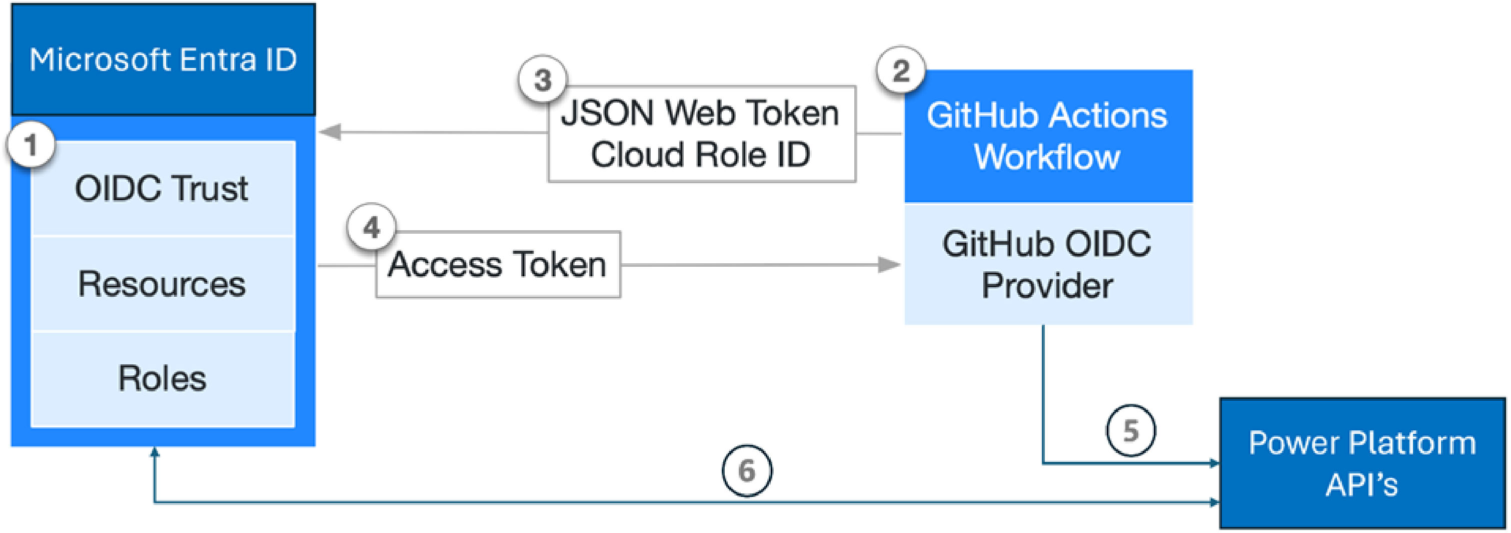
Create and destroy ephemeral dev/test environments as needed.  Achieve greater consistency between production and development environments.  Cross-tenant copying of environment resources for enterprises with non-production tenants for development.

**Tenant Management**

Accelerate infrastructure setup for complex self-hosted products in customer tenants.  Initialize and manage tenants for business partners like dealerships and franchises.

# Least-privilege Administration

- Minimize security risks by using service principals for deployment and management of Power Platform resources.
- Use OIDC Flow Client Credentials in automation scenarios trust CI/CD pipeline (no secrets or user interaction)



# Demo 1 : Power Platform Terraform Provider – Deploy D365 Finance Dev and Test

| | |
|---|---|
| Quickstarts: | aka.ms/PowerPlatformProvider |
| Source Code: | aka.ms/PowerPlatformProviderSource |
| Terraform Registry: | aka.ms/PowerPlatformTerraform |

# Demo 2 : Power Platform Terraform Provider – Deploy Power Platform Environments

| | |
|---|---|
| Quickstarts: | aka.ms/PowerPlatformProvider |
| Source Code: | aka.ms/PowerPlatformProviderSource |
| Terraform Registry: | aka.ms/PowerPlatformTerraform |

# So What?

- D365 and Power Platform resource management can be scripted at scale
- Most of your business applications infrastructure can be defined and managed from a single toolset
  - Not limited to Microsoft products
- **It’s free and open-source!**

# Get Involved!

microsoft/terraform-provider-power-platform

aka.ms/PowerPlatformTerraform

- Read the documentation
- Use the provider to validate your use cases
- Provide feedback through issues or discussions
- Collaborate and contribute PRs
  - Create new or enhance existing resources and data sources
  - Build Terraform modules for common feature sets
  - Additional QuickStart examples
- Open-source Community Project (not maintained or supported by Power Platform engineering teams)

# Q&A