

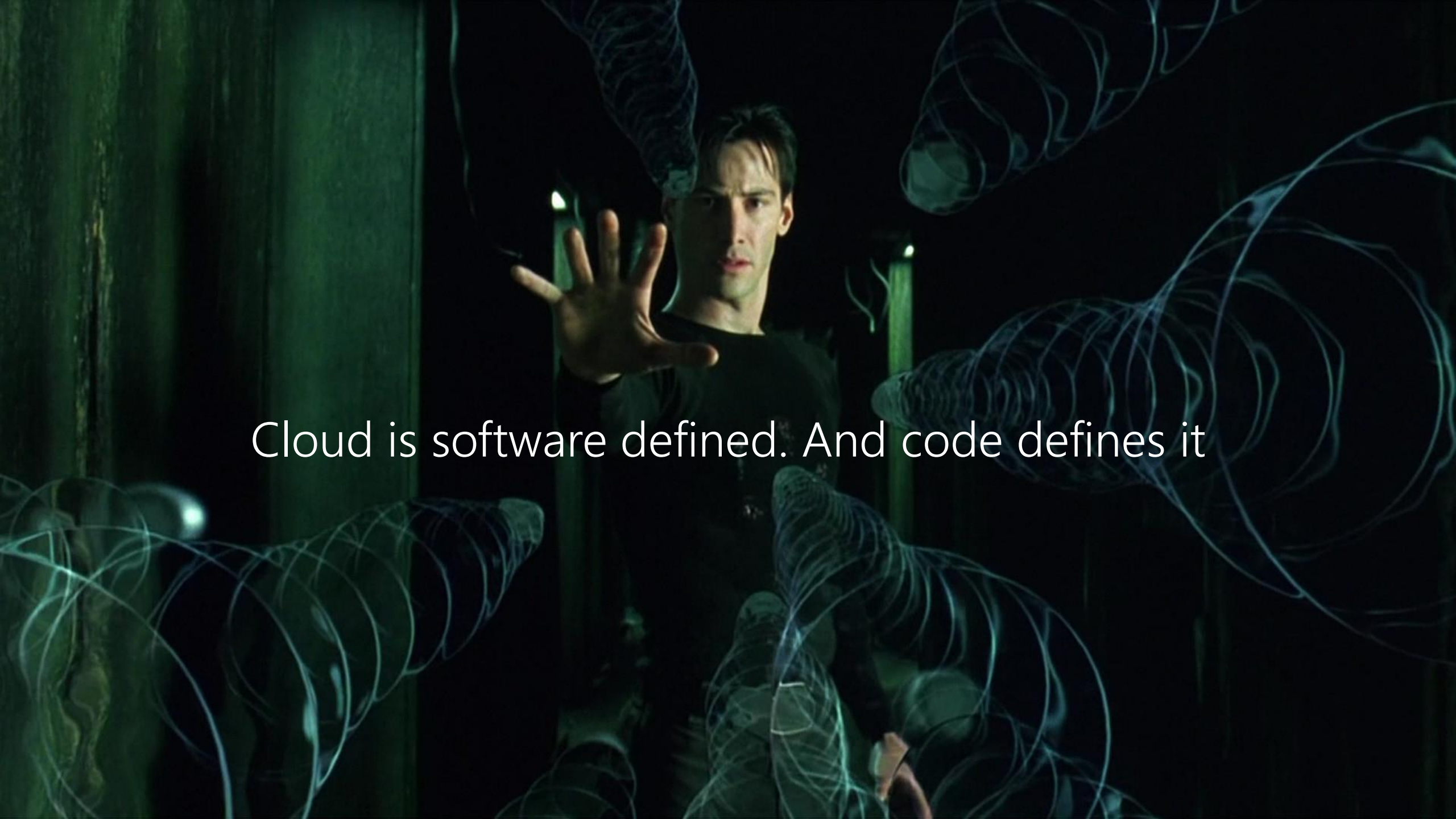


# Infrastructure **As Code** on **Azure**



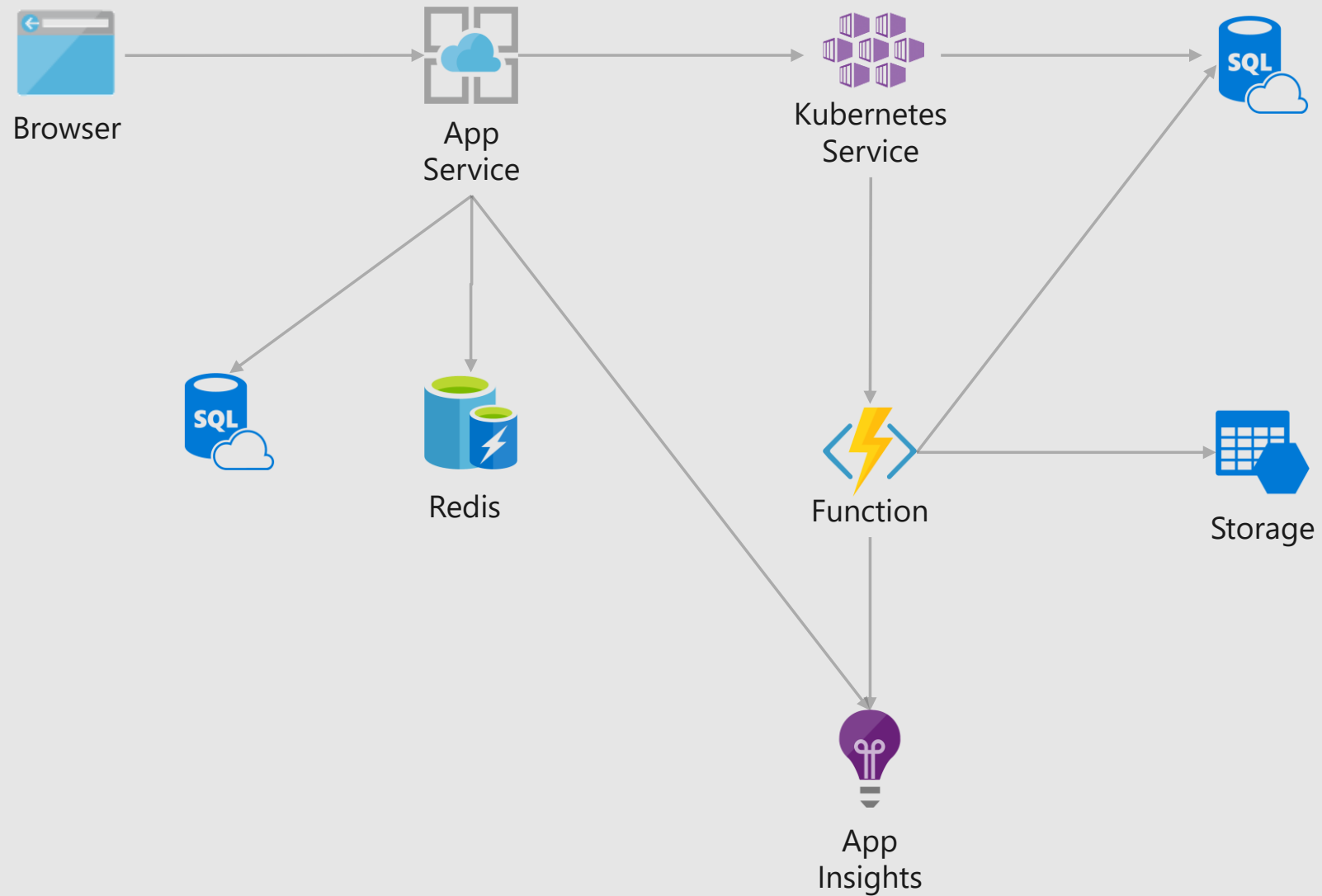















A still from the movie The Matrix showing Keanu Reeves as Neo. He is standing in a dark, industrial-looking environment with green vertical light beams. He has his right hand raised, palm facing forward, in a stopping gesture. The air is filled with glowing blue digital code and wireframe structures. The text "Cloud is software defined. And code defines it" is overlaid in white in the center of the image.

Cloud is software defined. And code defines it

# You want to build this...



# Which turns into this...

 <a href="#">estebandevstore</a>	Storage account	East US
 <a href="#">esteban-ignite-dev-function</a>	App Service	East US
 <a href="#">esteban-ignite-dev-function-sp</a>	App Service plan	East US
 <a href="#">esteban-ignite-dev-logic</a>	Logic app	East US
 <a href="#">esteban-ignite-dev-redis</a>	Redis Cache	East US
 <a href="#">esteban-ignite-dev-sp</a>	App Service plan	East US
 <a href="#">esteban-ignite-dev-sql</a>	SQL server	East US
 <a href="#">esteban-ignite-config-dev-db (esteban-ignite-dev-sql/esteban-ignite-....</a>	SQL database	East US
 <a href="#">esteban-ignite-dev-db (esteban-ignite-dev-sql/esteban-ignite-dev-db)</a>	SQL database	East US
 <a href="#">esteban-ignite-dev-web</a>	Application Insights	East US
 <a href="#">esteban-ignite-dev-web</a>	App Service	East US

# Solving the problem

Version control

CI/CD pipeline

Peer reviews

Dependencies (code and infrastructure)

Experiment

# Infrastructure as Code (IaC)

Descriptive model

Leverage version control

Key DevOps practices

Continuous delivery



# Why IaC?

Infrastructure drifts

Inconsistent settings management

Reproduce environments

Manual processes

Hard-to-track configurations

# What do I get with IaC?

Idempotence

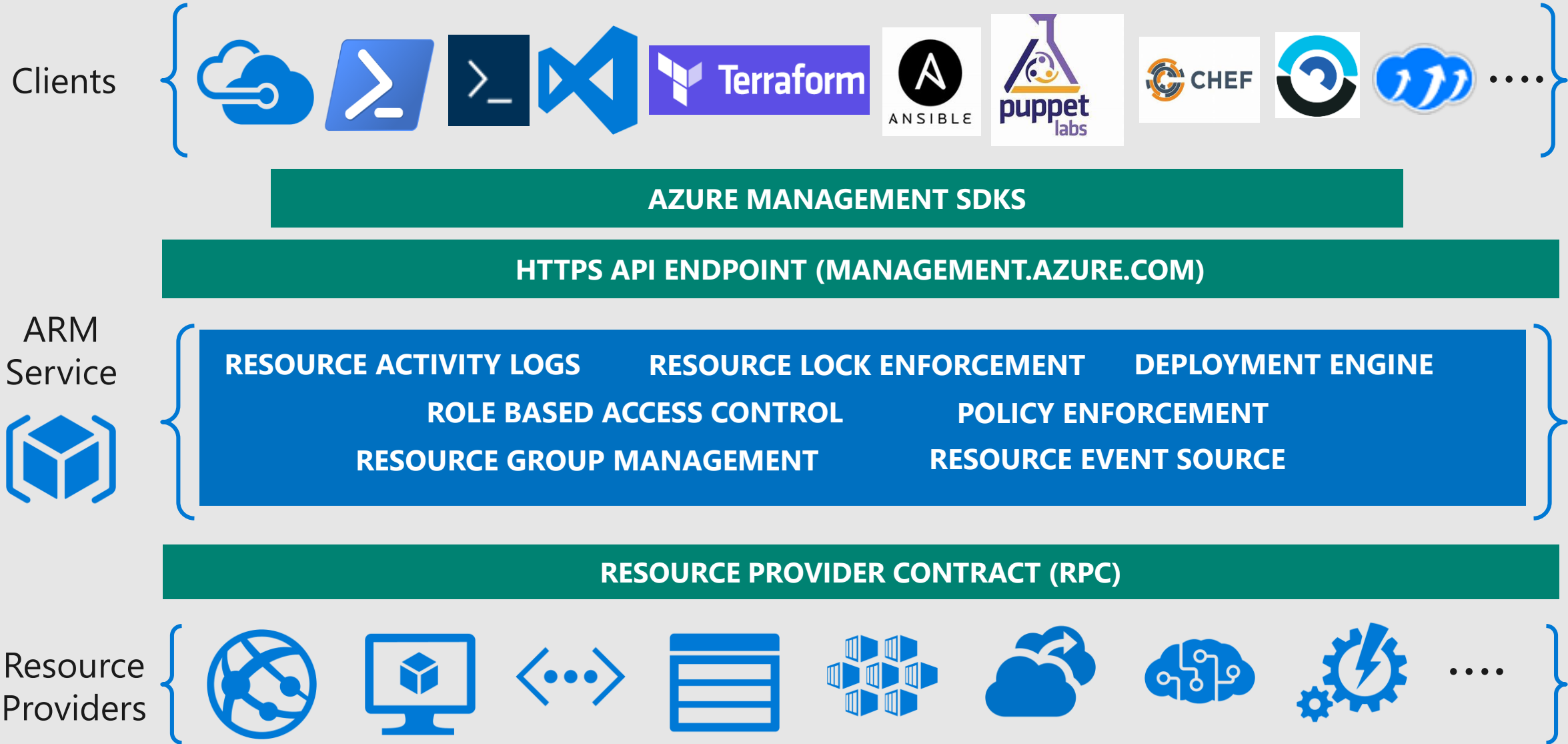
Test in production-like environments

Provision multiple environments

Practice deployments

Deliver stable environments

# What is Azure Resource Manager?



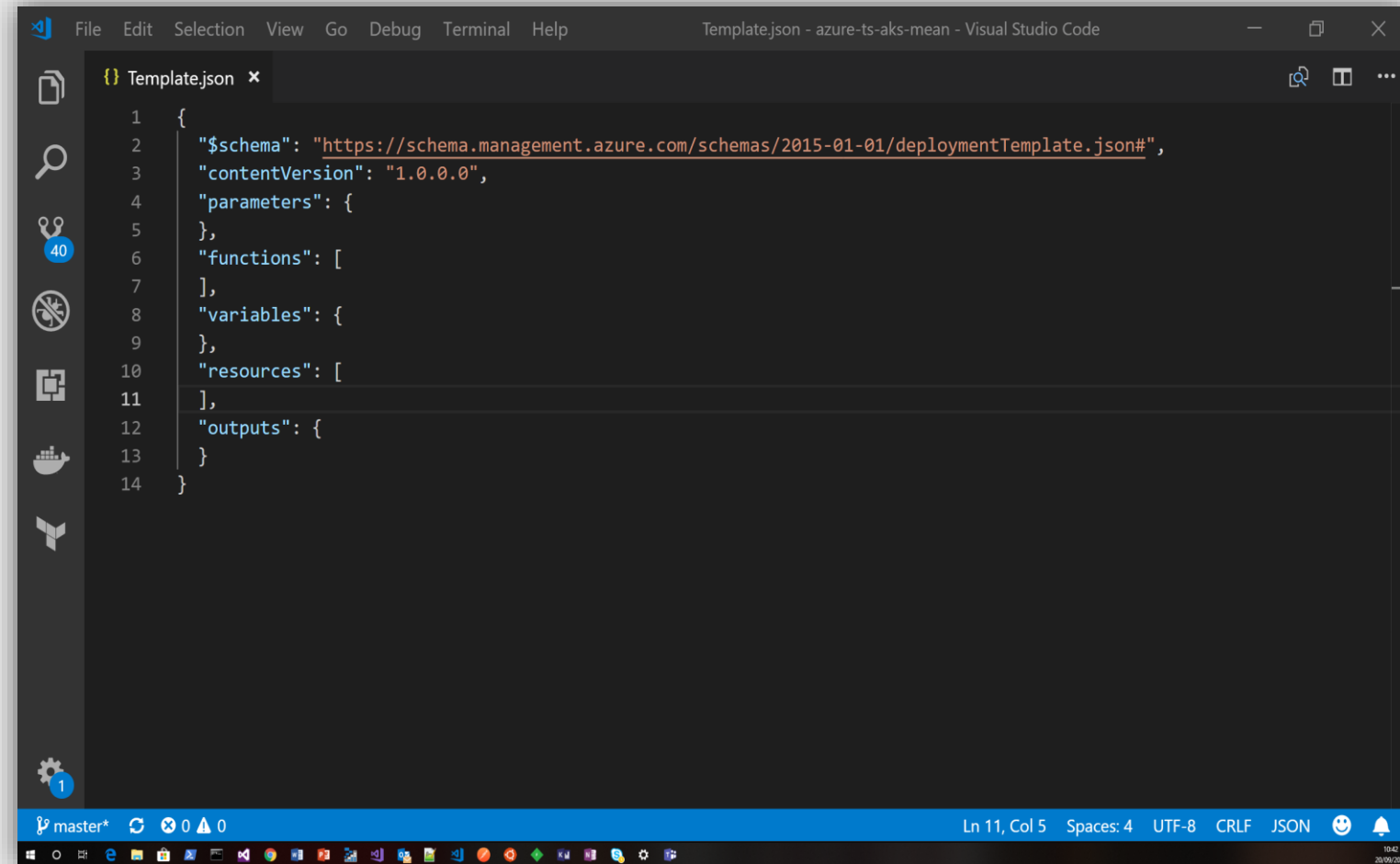
# ARM through REST (direct) or SDK layer

- Languages\Frameworks
  - go, Python, .Net, JavaScript, Java, Ruby, PHP
- Capabilities
  - Long Running operations, paging, retries
  - Connection management
  - Strongly Typed Object Model (Where this makes sense)
    - IDE support
  - Async support
  - Authentication support (e.g. Token Refresh)

# Azure Resource Deployment -Definition

## ARM Templates

- Declarative JSON DSL
- Inputs and Outputs
- Resources
- References
- Dependencies
- Language Expressions
- Nested Deployments



The screenshot shows the Visual Studio Code editor with a file named 'Template.json' open. The editor displays a JSON structure for an ARM template. The left sidebar shows the Explorer, Search, and Run and Debug views. The bottom status bar indicates the current position is Line 11, Column 5, with 4 spaces, UTF-8 encoding, CRLF line endings, and JSON language mode. The taskbar at the bottom shows the Windows Start button and several application icons.

```
1 {
2   "$schema": "https://schema.management.azure.com/schemas/2015-01-01/deploymentTemplate.json#",
3   "contentVersion": "1.0.0.0",
4   "parameters": {
5   },
6   "functions": [
7   ],
8   "variables": {
9   },
10  "resources": [
11  ],
12  "outputs": {
13  }
14 }
```

# Visual Studio, VS code, Portal

The image is a collage of three screenshots related to the Azure Resource Manager Tools extension.

**Top Left Screenshot:** A screenshot of the Visual Studio Code interface showing the 'Custom deployment' page. The breadcrumb navigation is 'Home > New > Marketplace > Everything > Template deployment > Custom deployment'. The page title is 'Custom deployment' with the subtitle 'Deploy from a custom template'. It includes links to 'Learn about template deployment', 'Read the docs', and 'Build your own template in the editor'. Under 'Common templates', there are links to 'Create a Linux virtual machine', 'Create a Windows virtual machine', 'Create a web app', and 'Create a SQL database'. At the bottom, there is a section 'Load a GitHub quickstart template' with a search bar.

**Top Right Screenshot:** A screenshot of the Visual Studio Code 'EXTENSIONS: MARKETPLACE' view. The search bar contains 'azure resource manger'. The extension 'Azure Resource Manager Tools' by Microsoft is highlighted, showing version 0.4.2 and a description: 'Template language support for Azure Resource Manager JSON files'. It also shows 'Azure Resource Manager Snippets' and 'Azure Functions' extensions.

**Bottom Screenshot:** A screenshot of the 'New Project' dialog in Visual Studio. The 'Installed' tab is selected, showing a list of templates. The 'Azure Resource Group' template is selected. The dialog shows the project name 'AzureResourceGroup1', location 'c:\users\exampleuser\documents\visual studio 2017\Projects', and solution name 'AzureResourceGroup1'. It also has checkboxes for 'Create directory for solution' and 'Create new Git repository'.

# Terraform

## Azure DevOps Tool Integrations

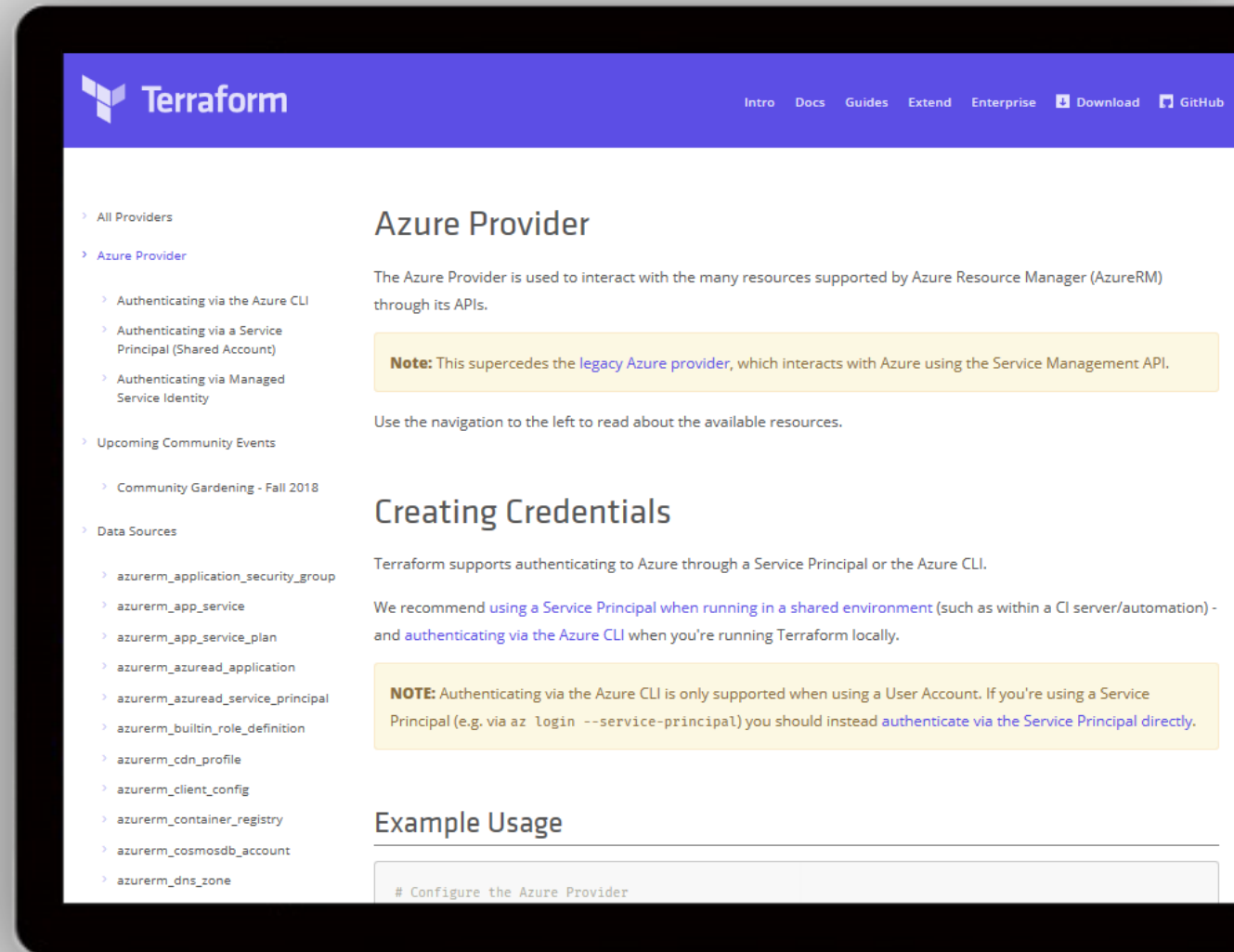
Bringing native Azure support for customers using Terraform

- [Terraform in Azure Cloud Shell](#)
- [Azure Resource Provider](#)
- [Azure Module Registry](#)
- [Documentation Hub for Terraform](#)



[docs.microsoft.com/azure/terraform](https://docs.microsoft.com/azure/terraform)

© Microsoft Corporation



Azure

# Chef

## Azure DevOps Tool Integrations

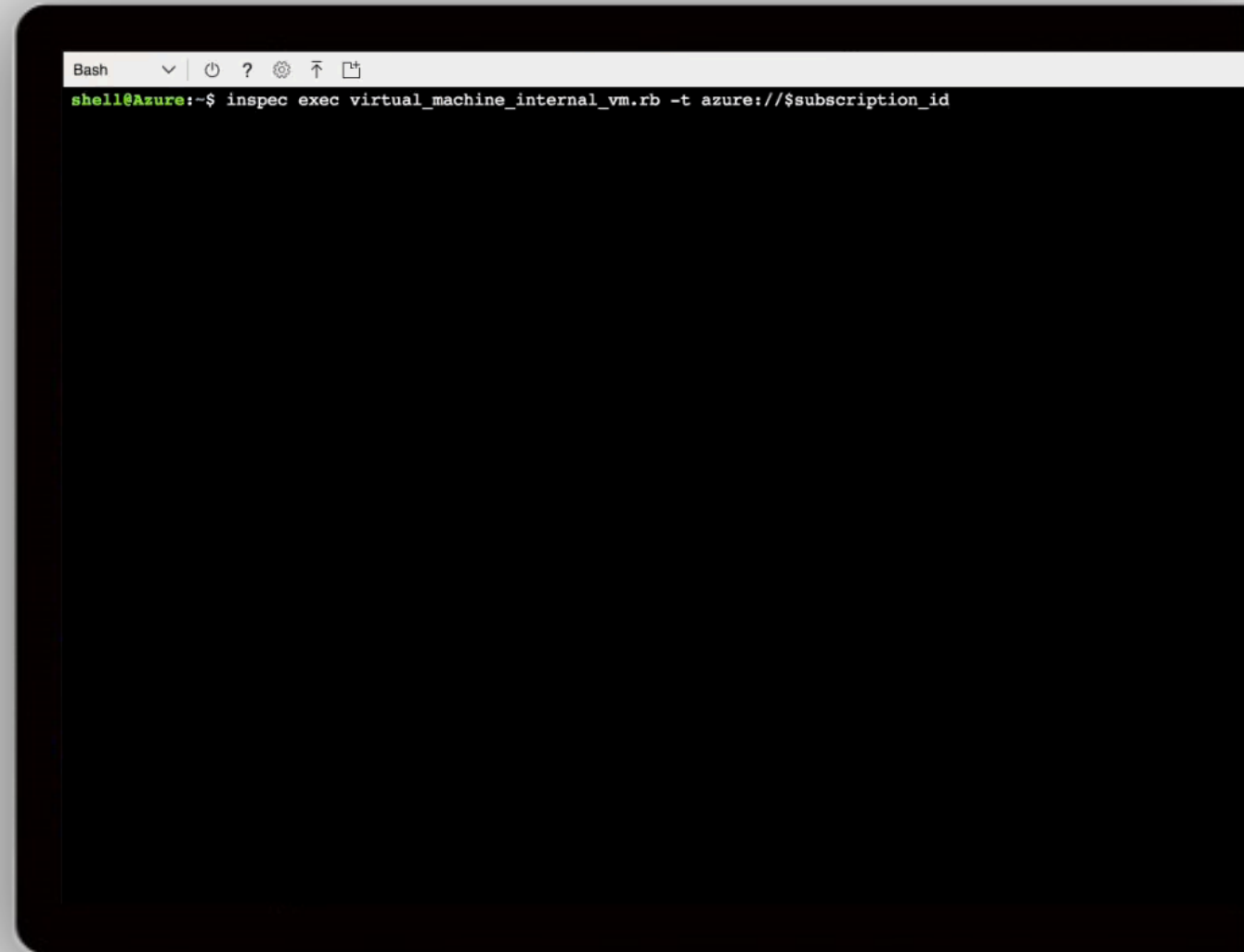


Bringing native Azure support for customers using Chef

- [Documentation Hub for Chef](#)
- [Chef in Azure Cloud Shell](#)
- [Chef VS Code Extension](#)
- [Habitat VS Code Extension](#)
- [InSpec VS Code Extension](#)



[docs.microsoft.com/azure/chef](https://docs.microsoft.com/azure/chef)





# Ansible

## Azure DevOps Tool Integrations

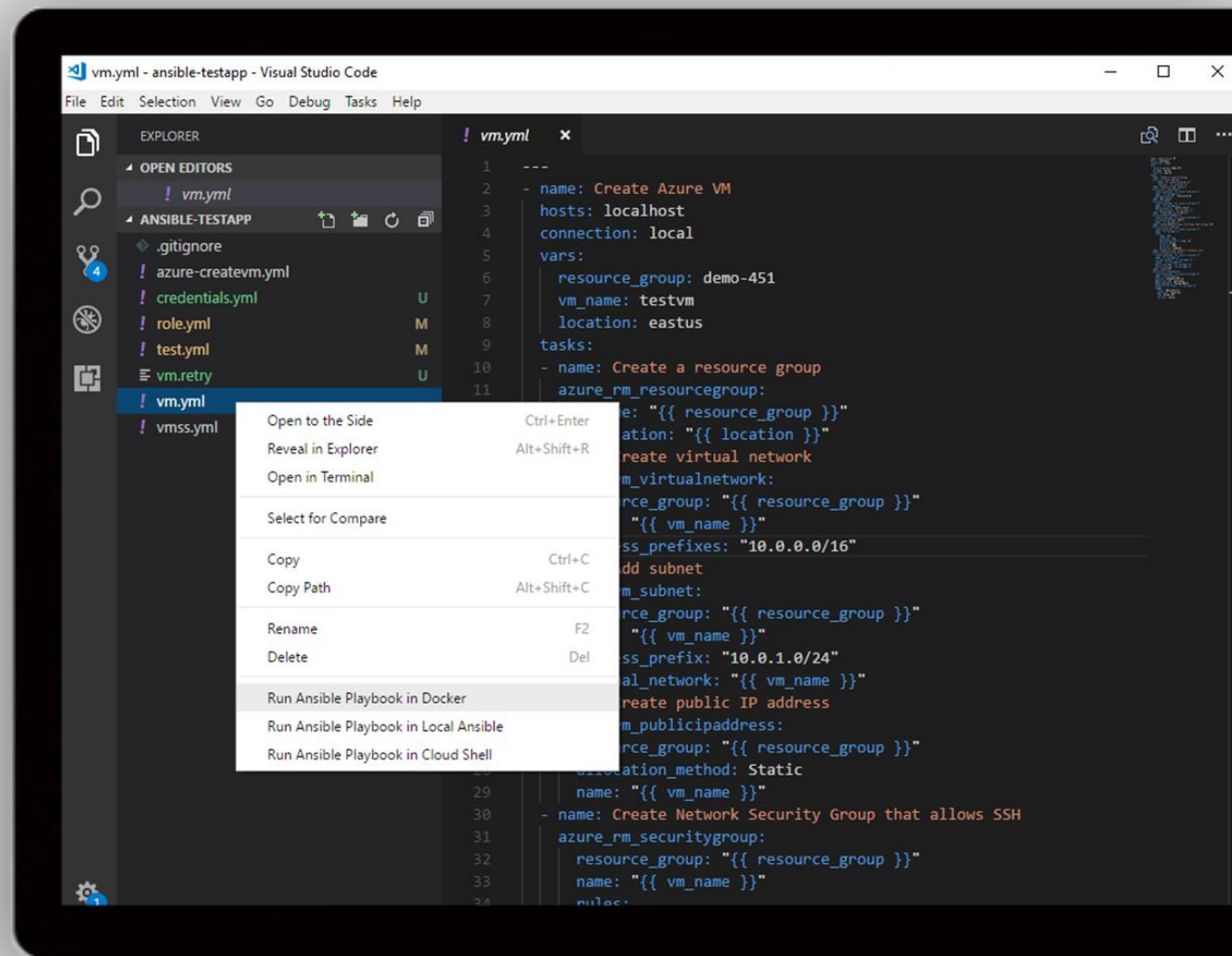


Bringing native Azure support for customers using Ansible

- [Documentation Hub for Ansible](#)
- [Ansible in Azure Cloud Shell](#)
- [Visual Studio Code Extension](#)
- [Azure Modules](#)
- [Azure Preview Modules](#)
- [Azure Playbook Samples](#)



[docs.microsoft.com/azure/ansible](https://docs.microsoft.com/azure/ansible)



# Recent ARM Template and Deployment Improvements



# Azure Resource Graph

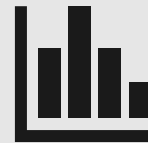
**Query, explore & analyze** cloud resources at scale

## Explore



Perform fast ad hoc **exploration** in large cloud environment

## Query & Analyze



Query & analyze across all of your cloud resources at scale in seconds

## Impact Assessment



Ability to **assess the impact** of applying policies in vast cloud environment

# ARM References

## Useful links

[REST API Browser](#) – an easy way to search and discover REST APIs

[REST Try It](#) – an interactive experience allowing you to try Azure REST APIs directly in your web browser

[Resource provider and data plane Swagger API definitions](#)

[ARM Schemas](#)

[Policy samples](#)

[Template quickstarts](#)

[Resources.azure.com](#)



# Terraform on **Azure**

# Terraform

What is Terraform?

- Open source project
- Cross computing environment templating language
- Provision, Update, and Delete resources
- Authored in HashiCorp Configuration Language (HCL) or JSON

# Terraform Example

```
resource "azurerm_resource_group" "testrg" {  
  name = "resourceGroupName"  
  location = "westus"  
}
```

Resource Group

```
resource "azurerm_storage_account" "testsa" {  
  name = "storageaccountname"  
  resource_group_name = "testrg"  
  location = "westus"  
  account_tier = "Standard"  
  account_replication_type = "GRS"  
}
```

Storage Account

# Dependencies

How are resource dependencies managed?

- Implicit – derived from interpolation
- Explicit – hard coded / explicit dependency

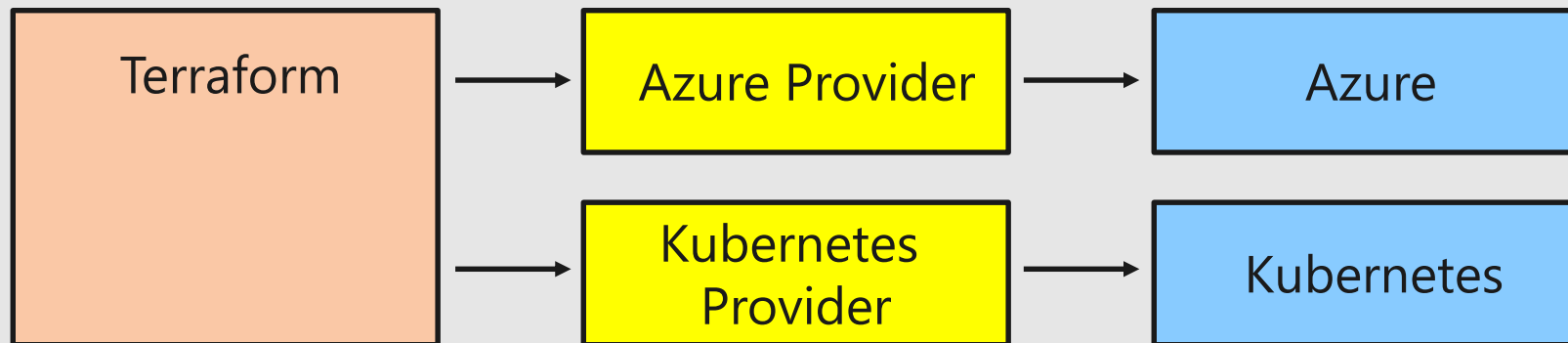
```
resource "azurerm_container_group" "demo-aci" {  
    name = "demo-aci"  
  
    depends_on = ["azure_cosmosdb_account.vote-db"]  
}
```



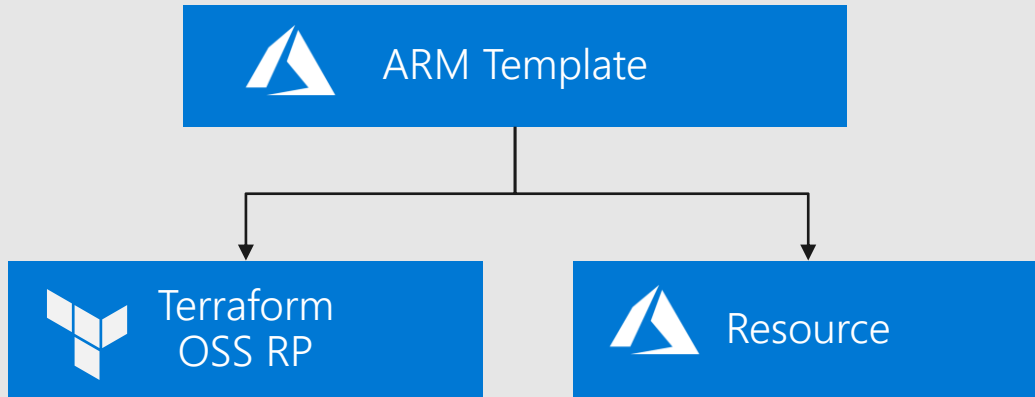
# Providers

What is a Terraform provider?

- Terraform 'extensions' for deploying resources
- Manages cloud / endpoint specific API interactions
- Available for major clouds and other platforms
- Hand authored (azurerm)



# Extending ARM beyond Azure Resources Using The ARM Terraform Resource Provider (private preview)



## [Enable Terraform resources on ARM](#)

Surfaced as an ARM resource

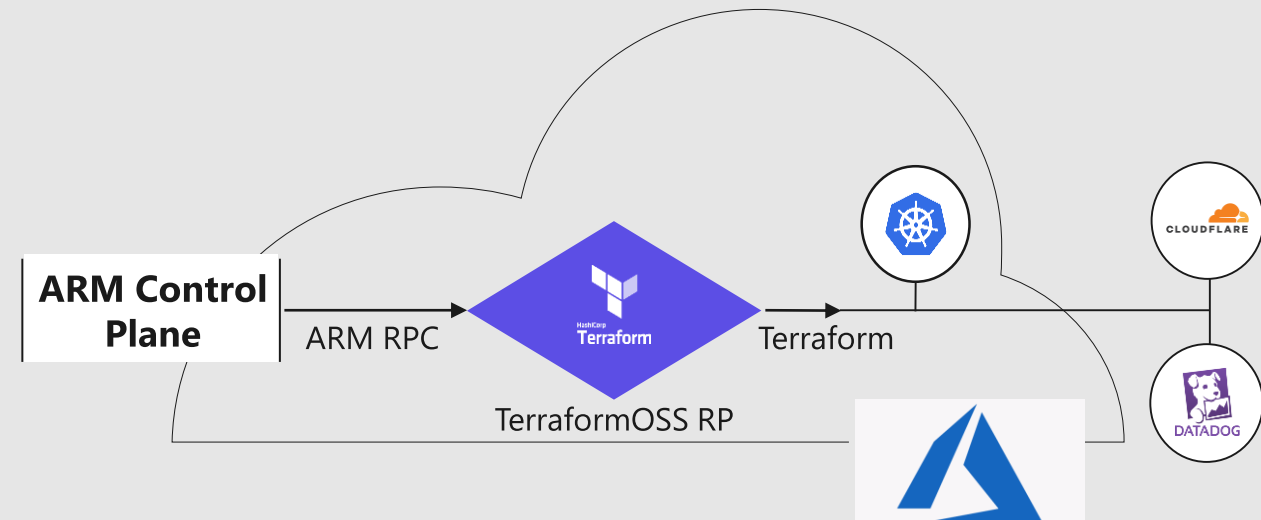
Reach both inside and outside Azure

Allows Azure lifecycle scenarios (RBAC, policy) on Terraform resources

Sign up for private preview: <https://aka.ms/tfossrp>

Currently Supports these Terraform  
Providers:

Kubernetes  
Datadog  
Cloudflare



# Basic Terraform commands

Once we have authored, how do we deploy?

- Terraform init – initializes working directory
- Terraform plan – pre-flight validation
- Terraform apply – deploys and updates resources
- Terraform destroy – removes all resources defined in a configuration

# State / Backend

What is Terraform state and why store it remotely?

Issues with local state:

- No collaboration
- Easy to delete / loose
- State files include secrets

Alternative:

- Store state in a backend (Azure Storage)



Uh...upgrades