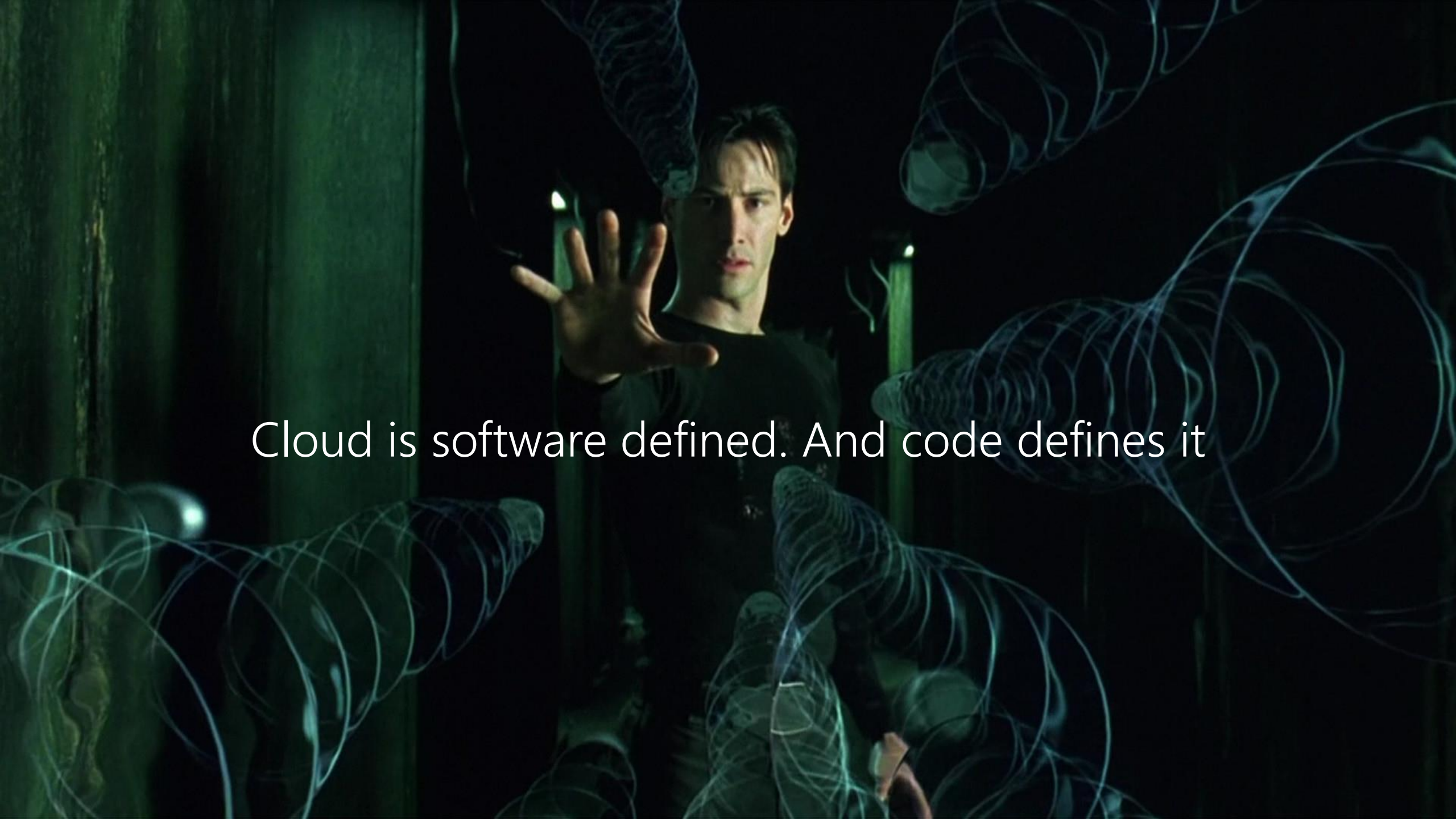
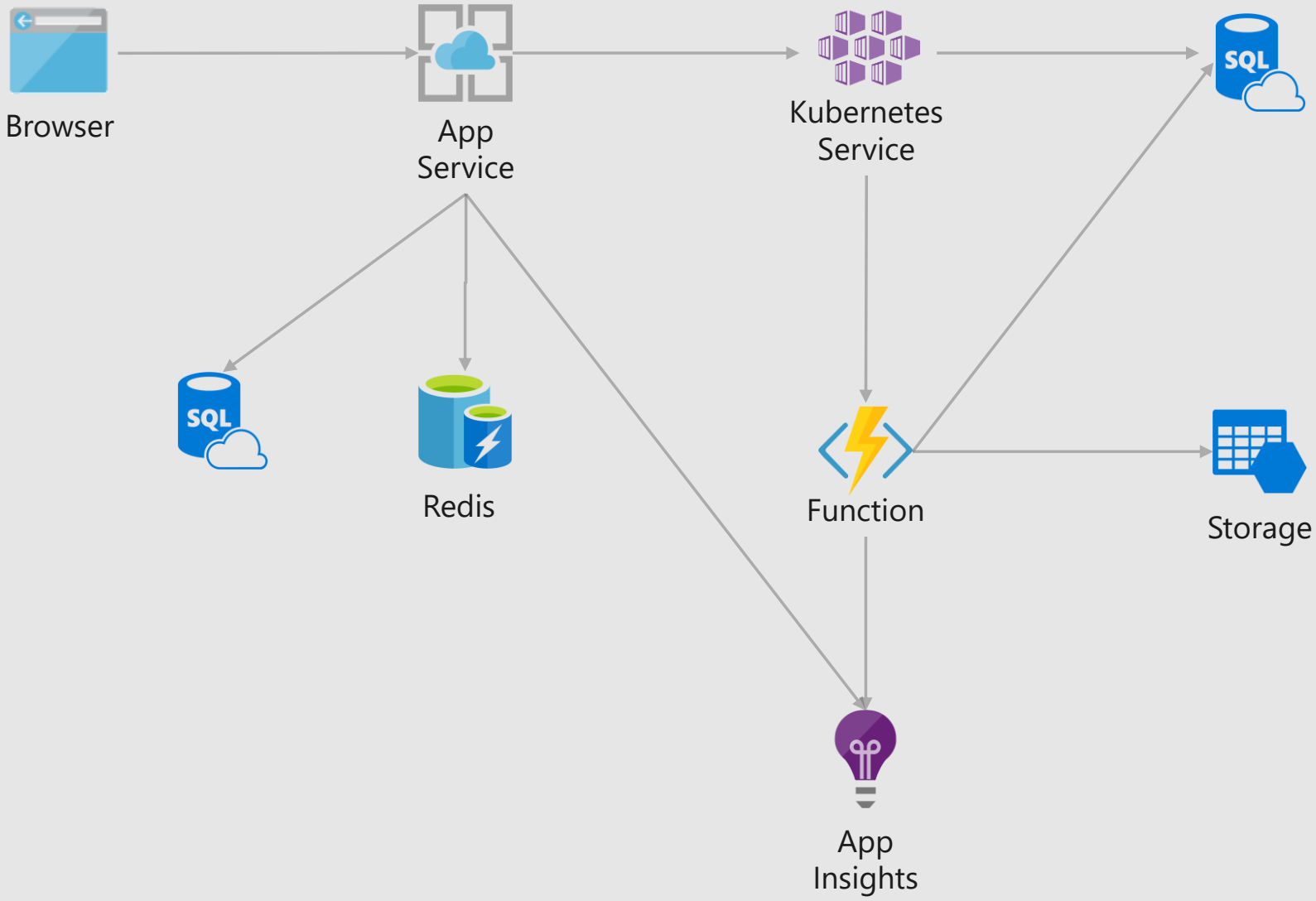
Microsoft

Infrastructure As Code on Azure


Cloud is software defined. And code defines it

You want to build this…



Which turns into this…

estebandevstore	Storage account	East US
esteban-ignite-dev-function	App Service	East US
esteban-ignite-dev-function-sp	App Service plan	East US
esteban-ignite-dev-logic	Logic app	East US
esteban-ignite-dev-redis	Redis Cache	East US
esteban-ignite-dev-sp	App Service plan	East US
esteban-ignite-dev-sql	SQL server	East US
esteban-ignite-config-dev-db (esteban-ignite-dev-sql/esteban-ignite-···	SQL database	East US
esteban-ignite-dev-db (esteban-ignite-dev-sql/esteban-ignite-dev-db)	SQL database	East US
esteban-ignite-dev-web	Application Insights	East US
esteban-ignite-dev-web	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

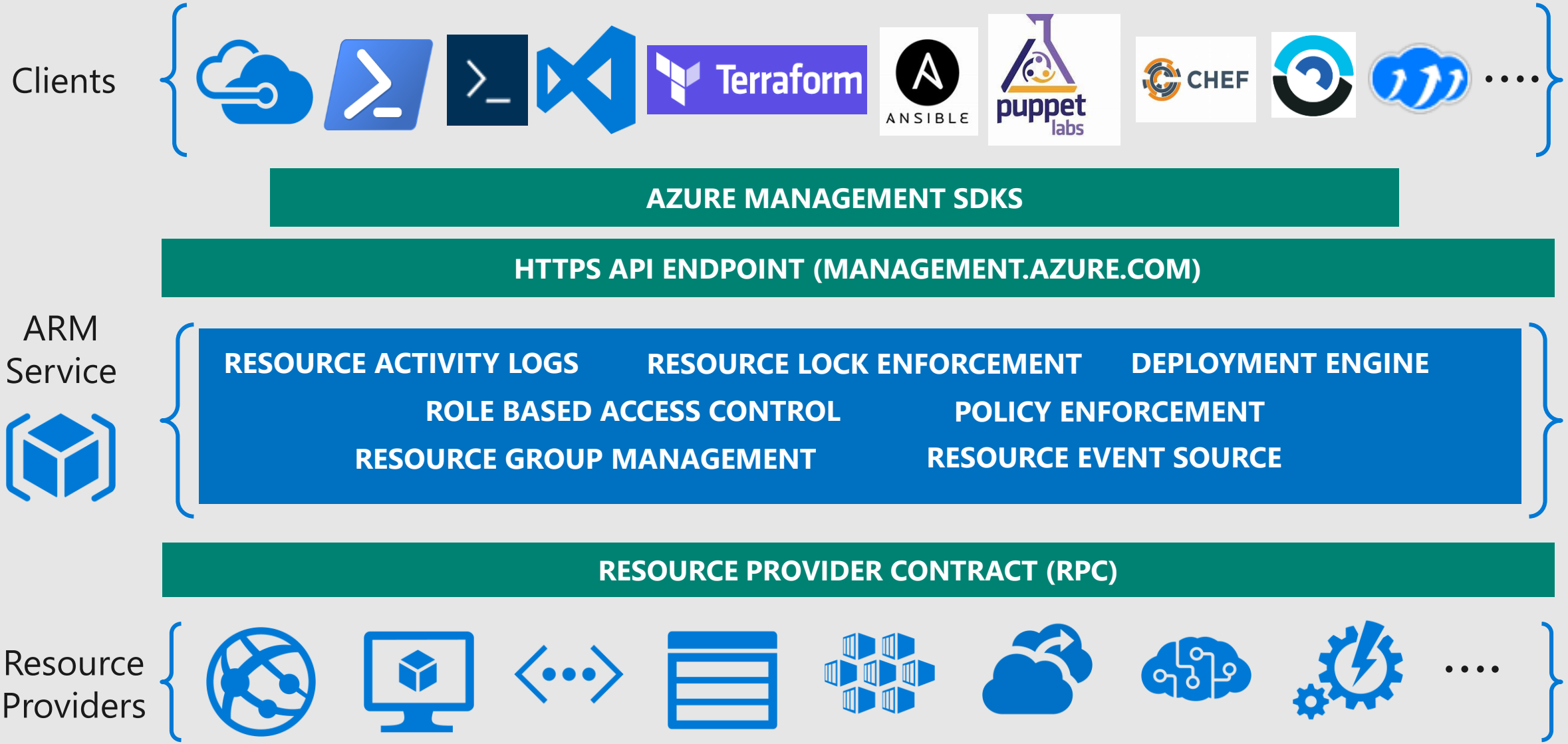
Test in production-like environments

Provision multiple environments

Practice deployments

Deliver stable environments

What is Azure Resource Manager?

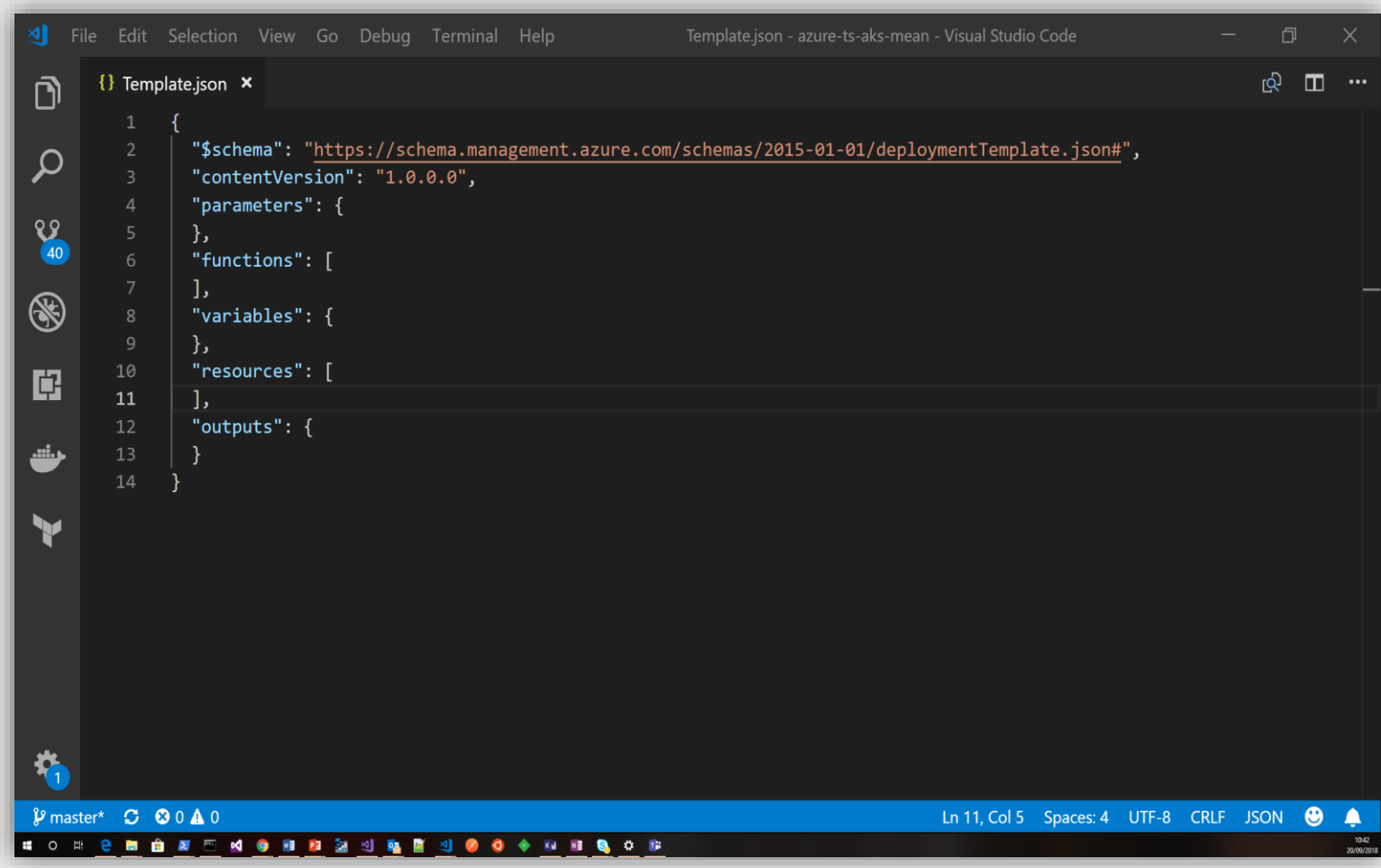


ARM through REST (direct) or SDK layer

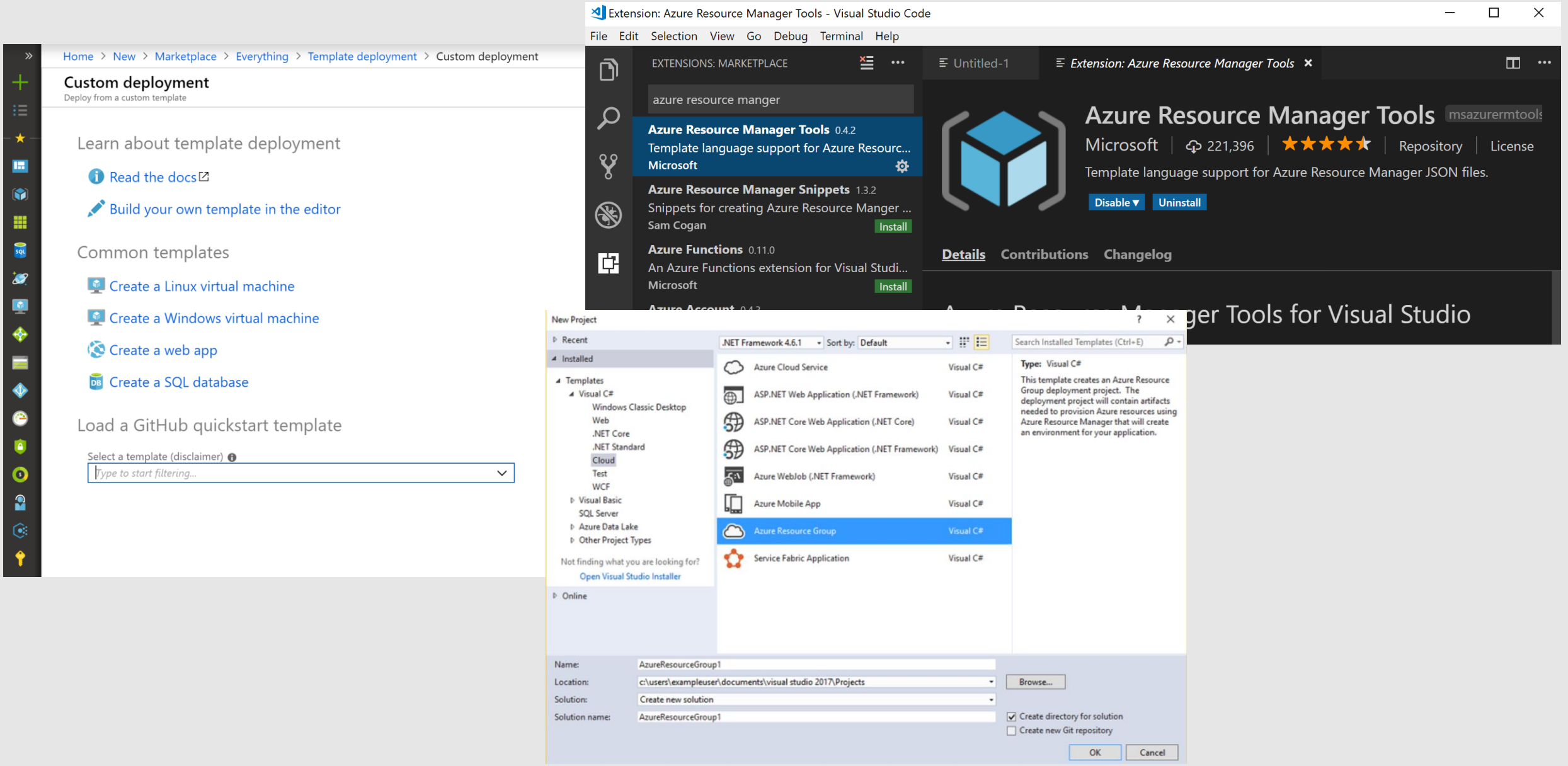
- Languages\Frameworks
  - go, Python, .Net, JavaScript, Java, Ruby, PHP
- Capabilities
  - Long Running operations, paging, retries
  - Connection management
  - Strongly Typed Object Model (Where this makes sense)
    - IDE support
  - Async support
  - Authentication support (e.g. Token Refresh)

Azure Resource Deployment -Definition

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



Visual Studio, VS code, Portal



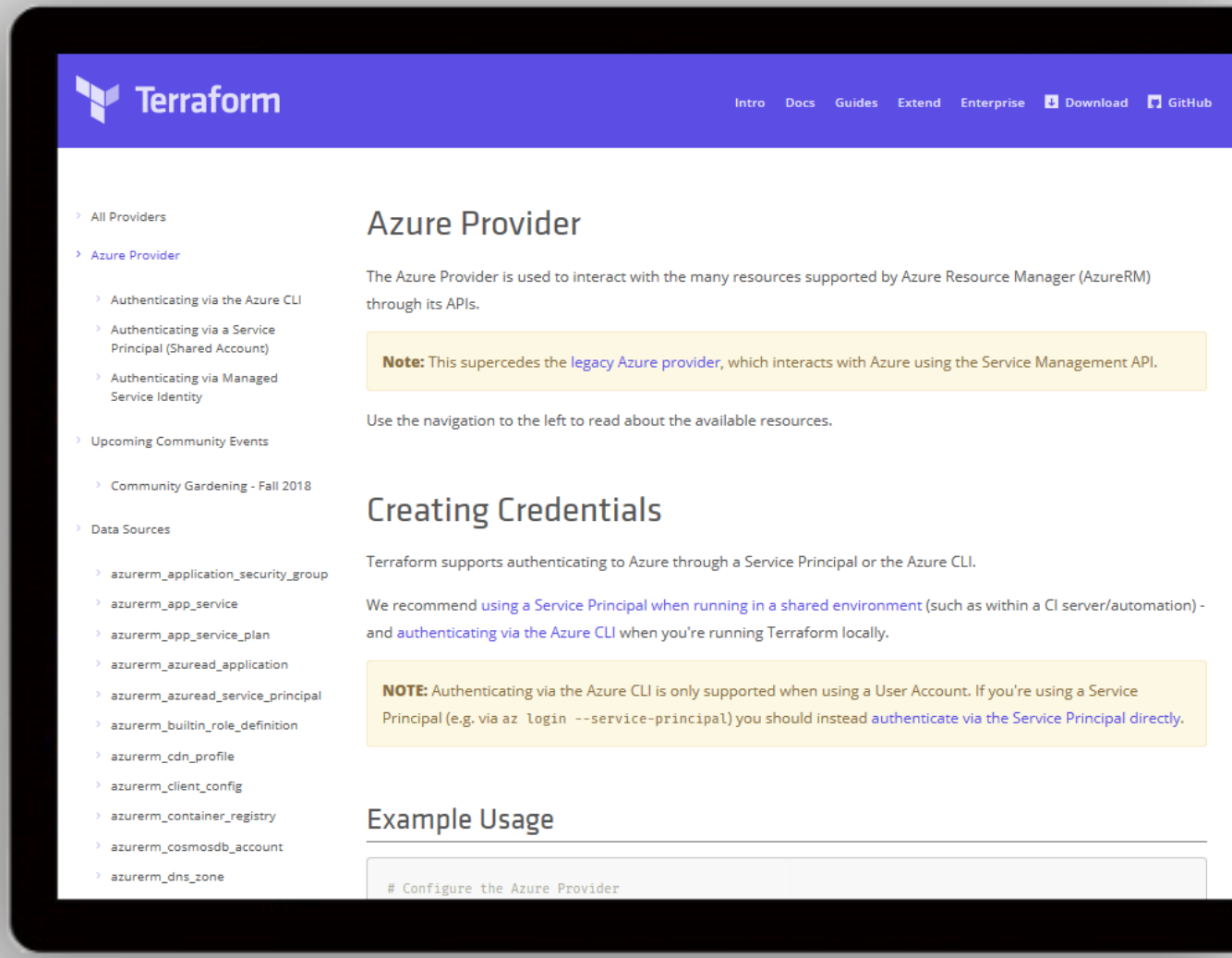
Terraform

Azure DevOps Tool Integrations

HashiCorp Terraform

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

Chef

Azure DevOps Tool Integrations

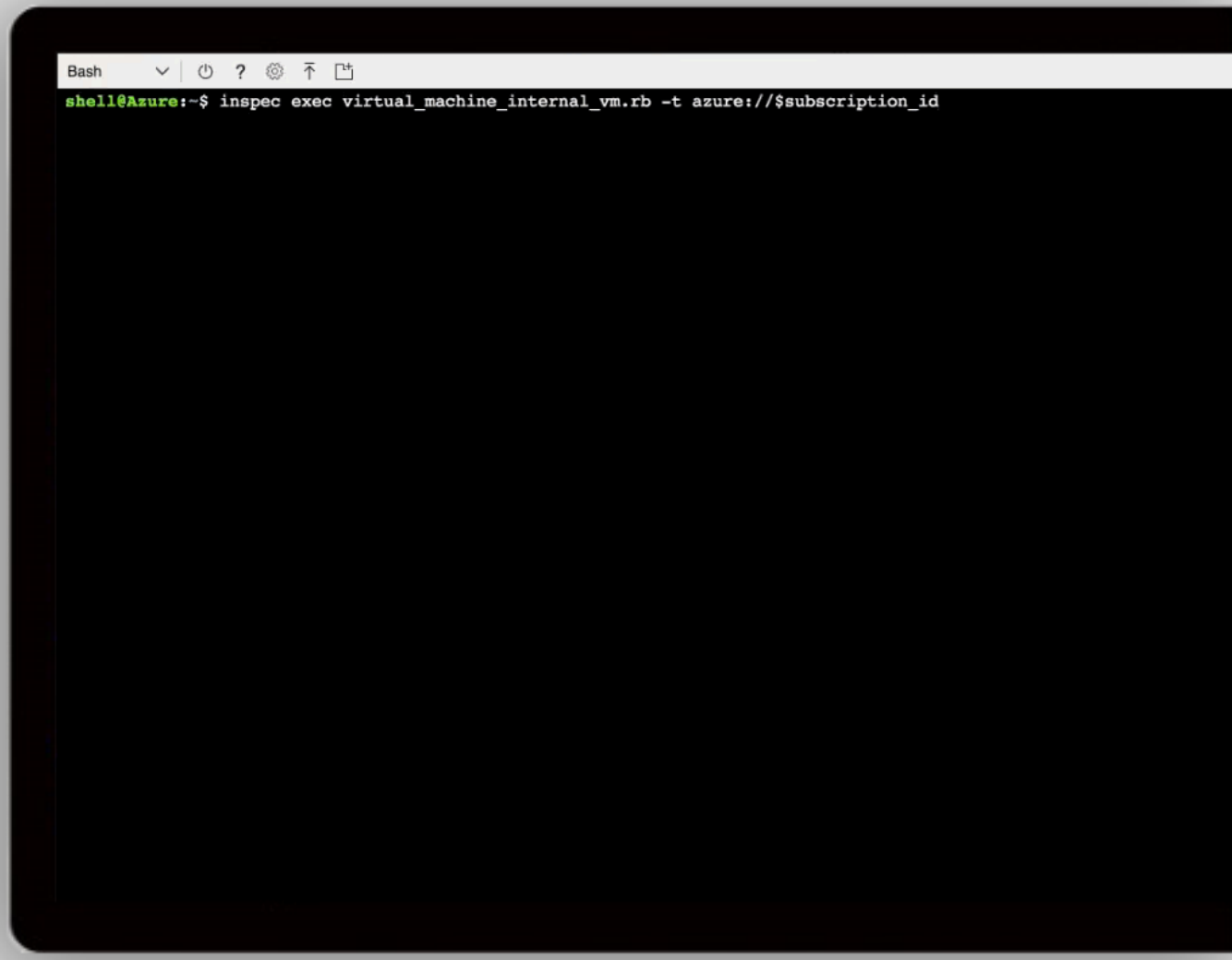
Bringing native Azure support for customers using Chef

- Documentation Hub for Chef
- Chef in Azure Cloud Shell
- Chef VS Code Extension
- Habitat VS Code Extension
- InSpec VS Code Extension

habitat BY CHEF

INSPEC BY CHEF

docs.microsoft.com/azure/chef

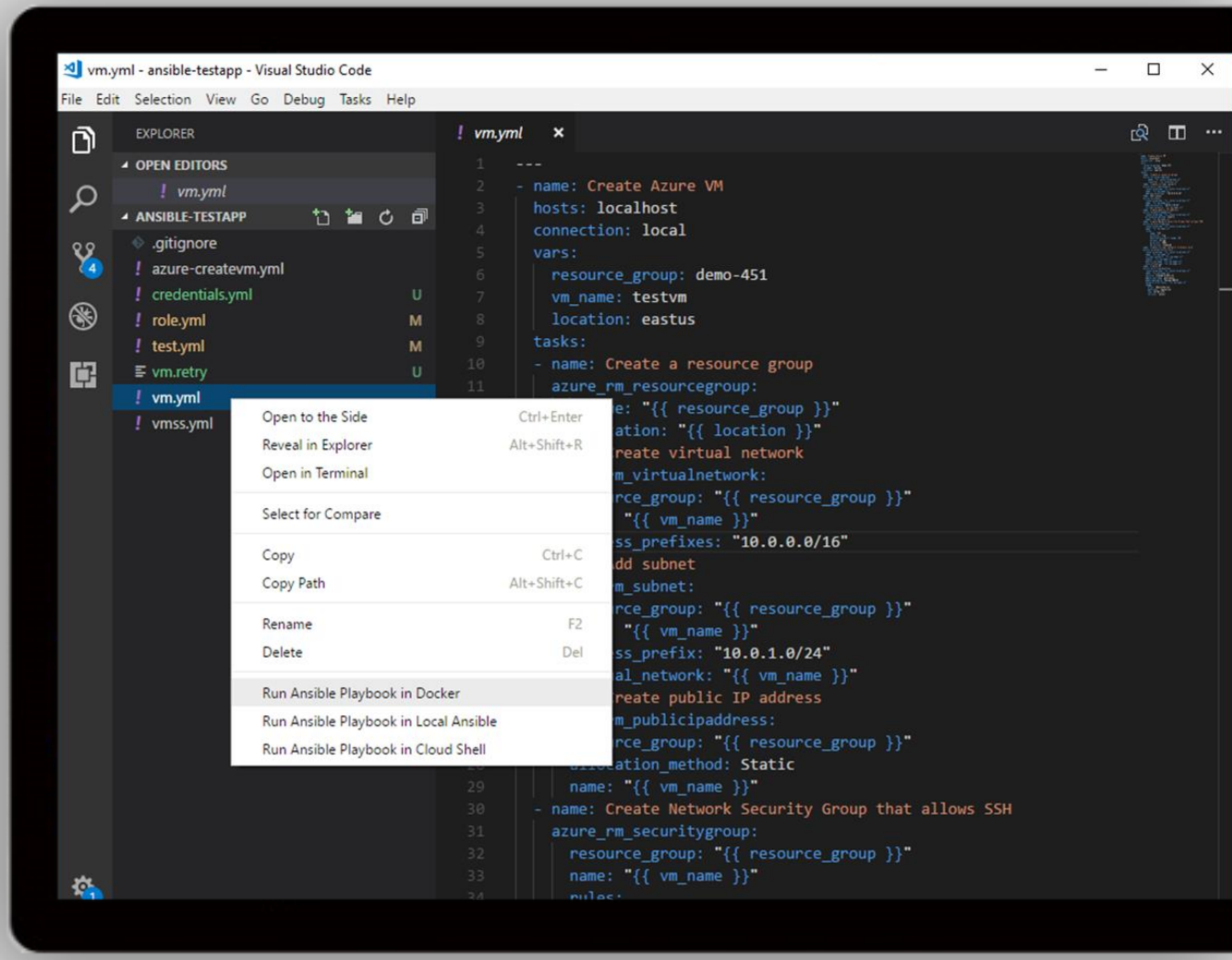


Ansible

Azure DevOps Tool Integrations

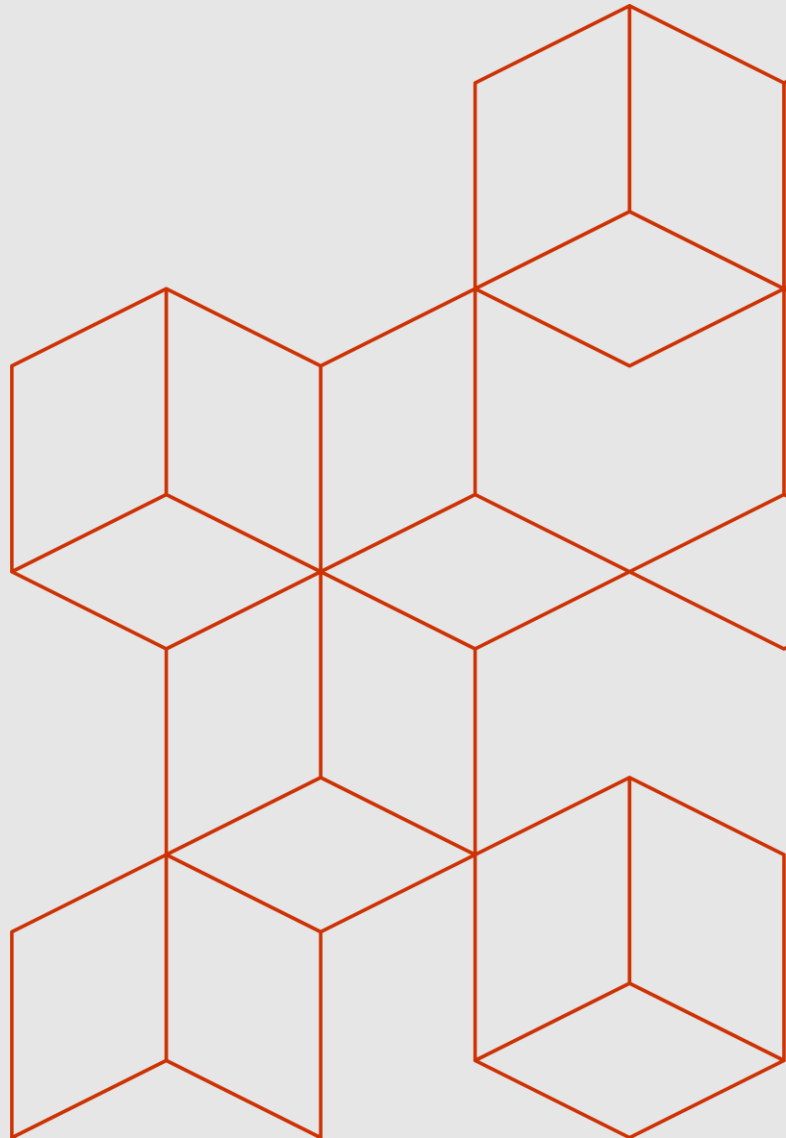
Bringing native Azure support for customers using Ansible

- 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

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

Microsoft

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

Resource Group

Storage Account

Dependencies

How are resource dependencies managed?

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

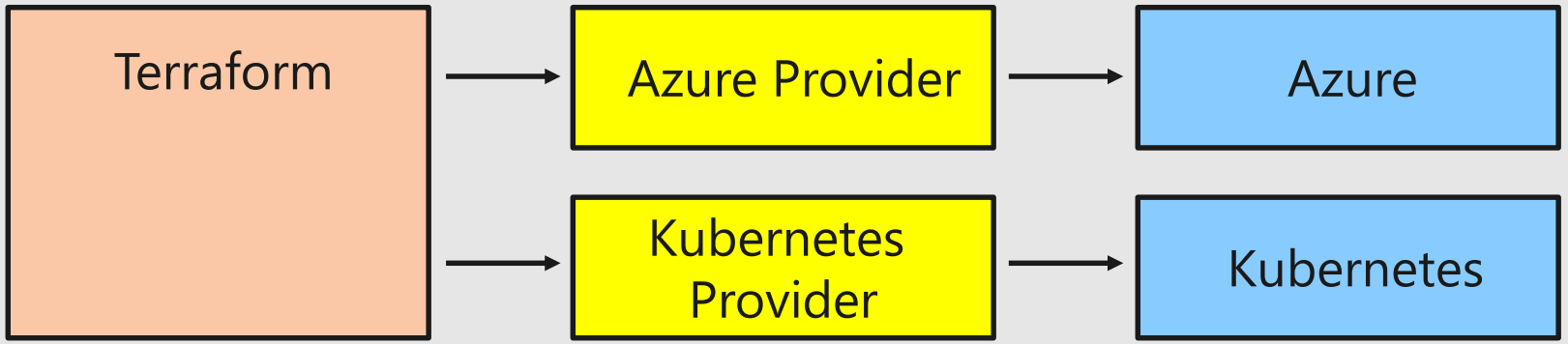
```
resource "azurerm_container_group" "demo-aci" {
    name = "demo-aci"

    depends_on = ["azure_cosmosdb_account.vote-db"]
}
```

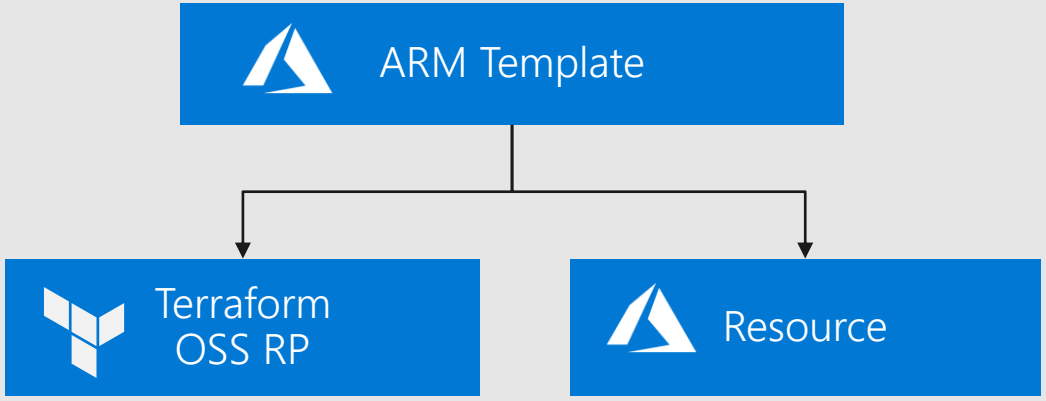
Providers

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

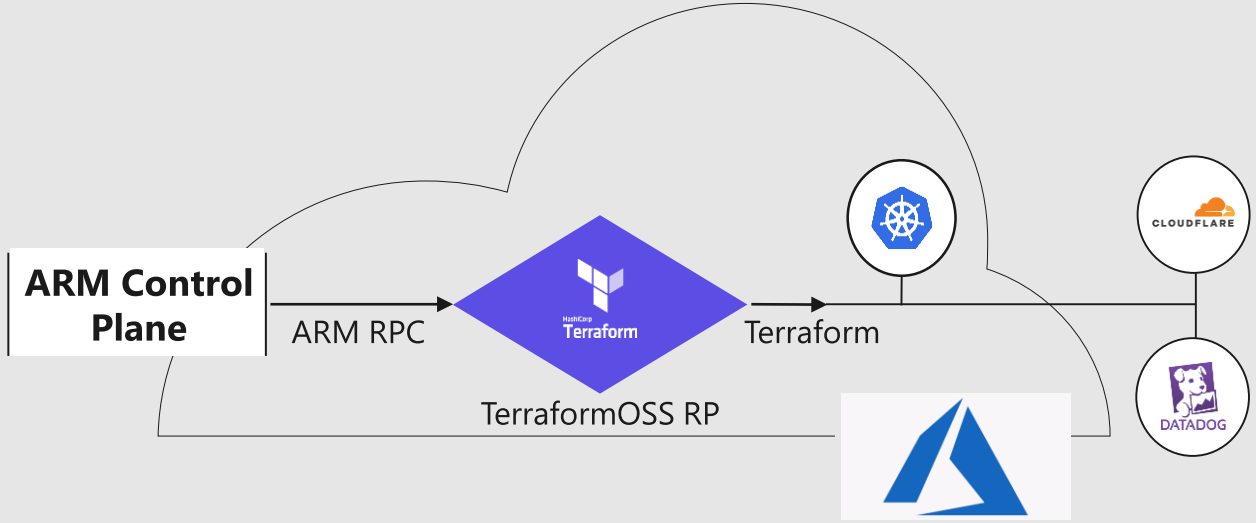
Reach both inside and outside Azure

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