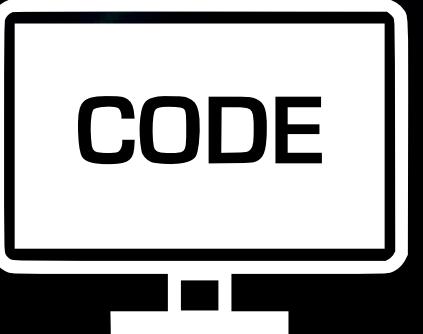




Infrastructure as Code using Terraform on Azure: Implement Managed Identities in Azure Functions

by Jonah Andersson



Hi, I'm Jonah Andersson



Inspire Tech to others!

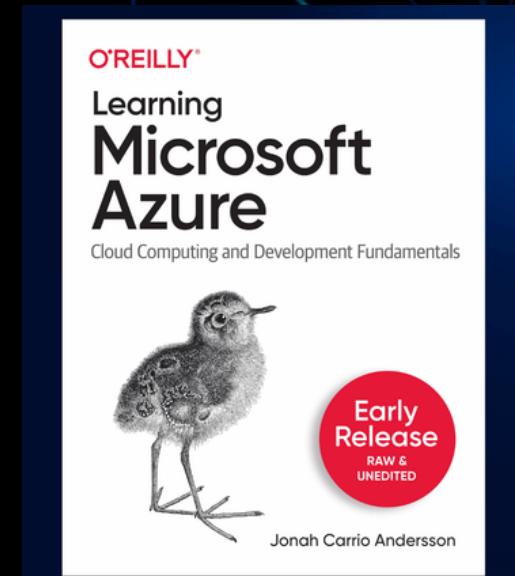
- < Fullstack Developer />
- < Cloud Infrastructure & DevOps Engineer />
- < Book Author />
- < Community Leader />



[/jonahandersson](https://www.linkedin.com/in/jonahandersson)



[@cjkodare](https://twitter.com/cjkodare)



Infrastructure as Code (IaC)

Azure Functions

Managed Identities

Terraform on Azure



/jonahandersson



@cjkodare



Azure Functions



Infrastructure as Code (IaC)



Managed Identities



Terraform on Azure



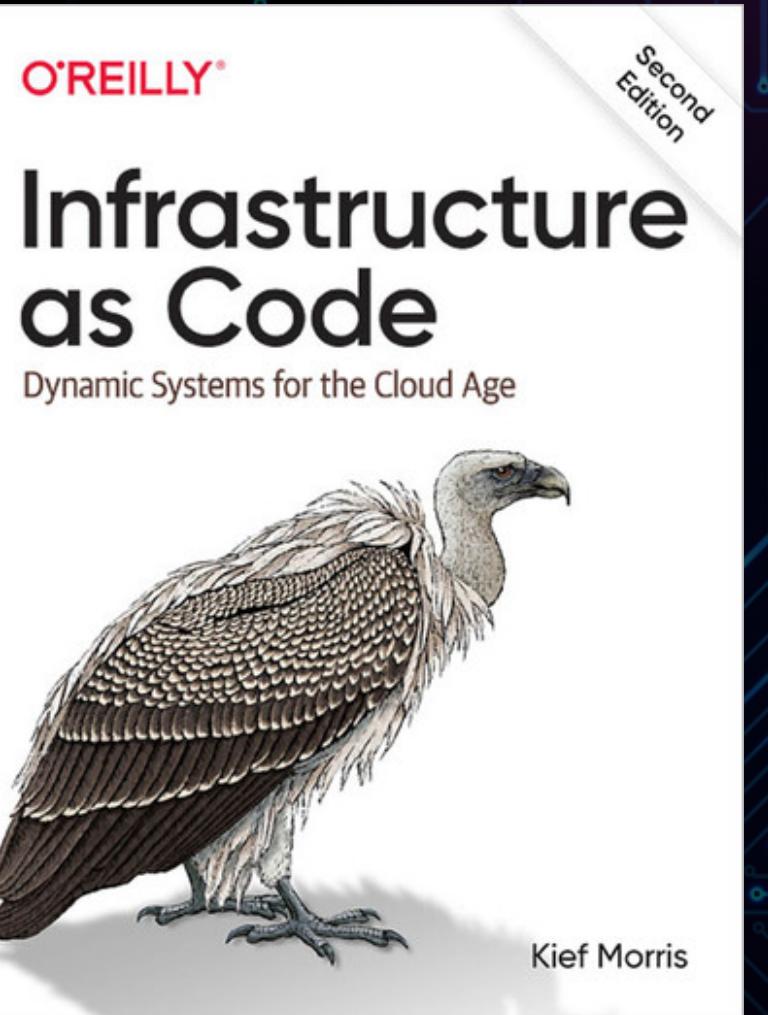
/jonahandersson



@cjkodare

Iron Age vs. Cloud Age

Iron Age	Cloud Age
Cost of change is high	Cost of change is low
Changes represent failure (changes must be “managed,” “controlled”)	Changes represent learning and improvement
Reduce opportunities to fail	Maximize speed of improvement
Deliver in large batches, test at the end	Deliver small changes, test continuously
Long release cycles	Short release cycles
Monolithic architectures (fewer, larger moving parts)	Microservices architectures (more, smaller parts)
GUI-driven or physical configuration	Configuration as Code



Credit: Infrastructure as Code: Dynamic Systems for the Cloud Age by Kief Morris (O'Reilly Media)



/jonahandersson



@cjkodare

“

Infrastructure as Code is a Cloud Age approach to managing systems that embraces continuous change for high reliability and quality.

O'REILLY®

Second
Edition

Infrastructure as Code

Dynamic Systems for the Cloud Age



Kief Morris



/jonahandersson

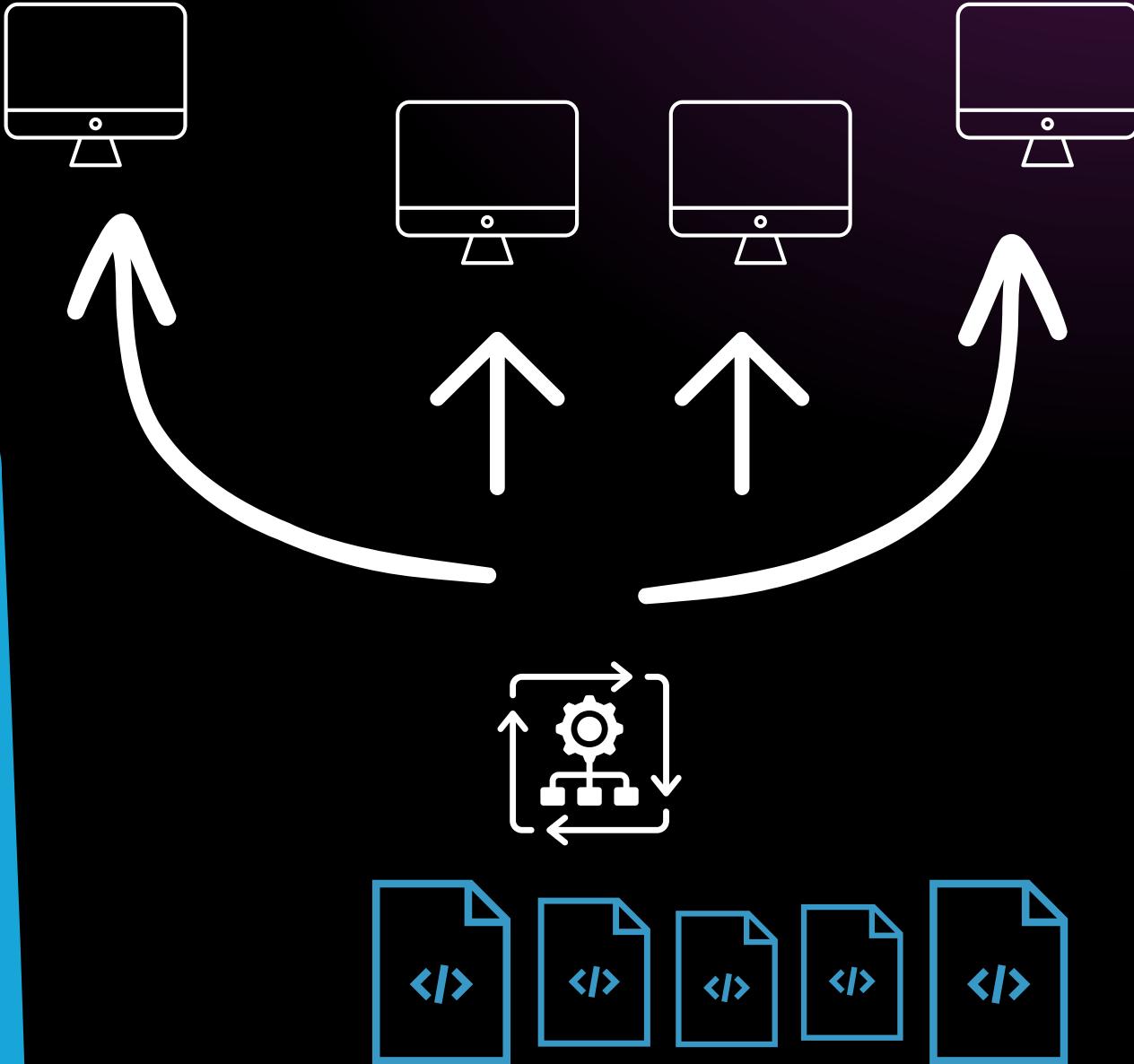


@cjkodare

IaC - The Definition

Infrastructure as code (IaC) tools allow you to **manage infrastructure** with configuration files rather than through a graphical user interface.

IaC allows you **to build, change, and manage your infrastructure in a safe, consistent, and repeatable way** by defining resource configurations that you can version, reuse, and share"



/jonahandersson



@cjkodare

Benefits of Infrastructure as Code (IaC)

- Process automation
- Declarative Paradigm
- Repeatable Deployments
- Easier to scale out/in, configure and change when needed
- Enforce compliance best practices
- Collaboration between teams (DevOps, Dev, etc)
- Help solve the rising complexity of cloud and architecture

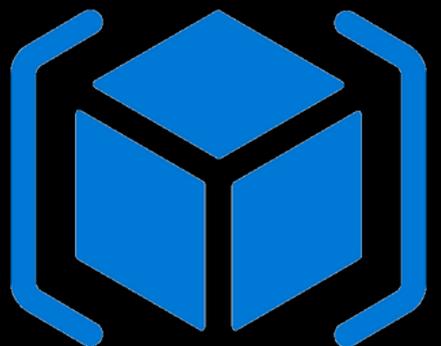


/jonahandersson

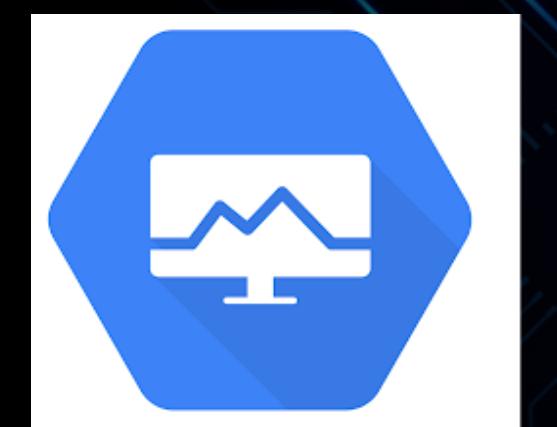


@cjkodare

Infrastructure as Code Tools



Azure Resource Manager (ARM)
Bicep

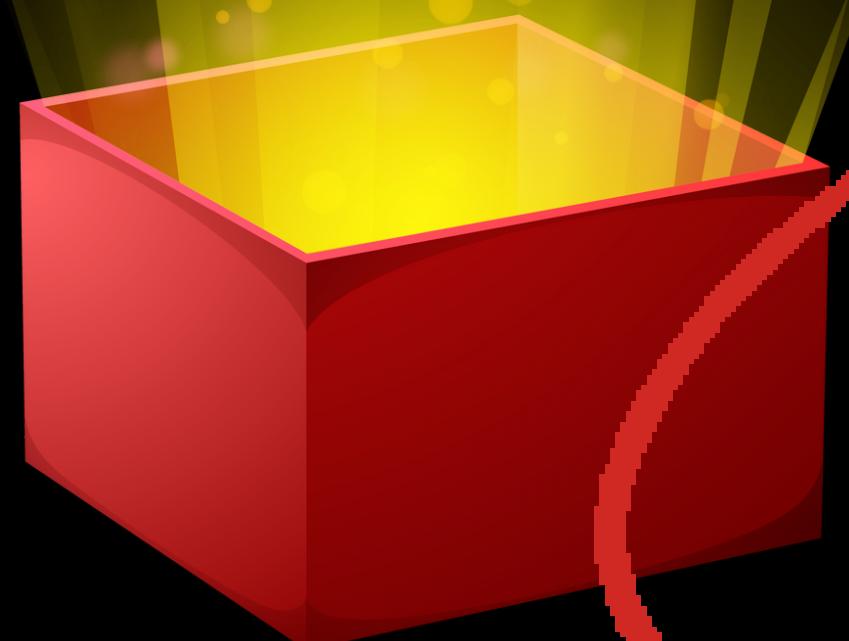




Infrastructure as Code (IaC)



Azure Functions



Managed Identities



Terraform on Azure



Terraform (Hashicorp)

- An open-source IaC (Infrastructure-as-Code) tool
- Used for cloud infrastructure provisioning, automation and management
- Codifies infrastructure in configuration files that describe the desired state for your topology.
- Multicloud support - public, private clouds, and even SaaS services using Terraform providers.



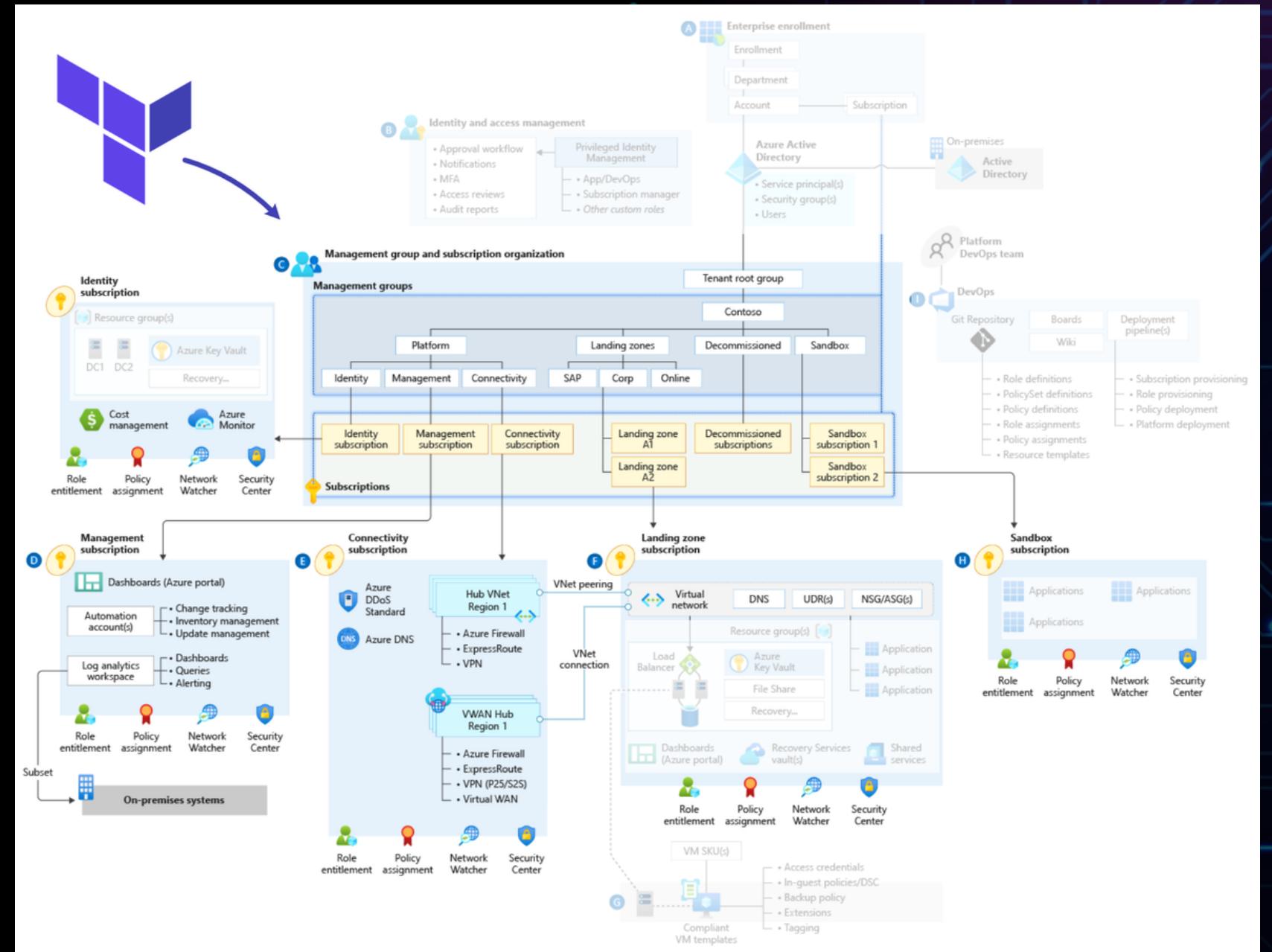
/jonahandersson



@cjkodare

Terraform on Azure

- Automate infrastructure management
- Provision core Azure platform capabilities (ex. CAF on Terraform)
- Provision Azure DevOps pipelines to automate regular infrastructure and application deployments.
- Provision Azure resources



<https://github.com/Azure/terraform-azurerm-caf-enterprise-scale>

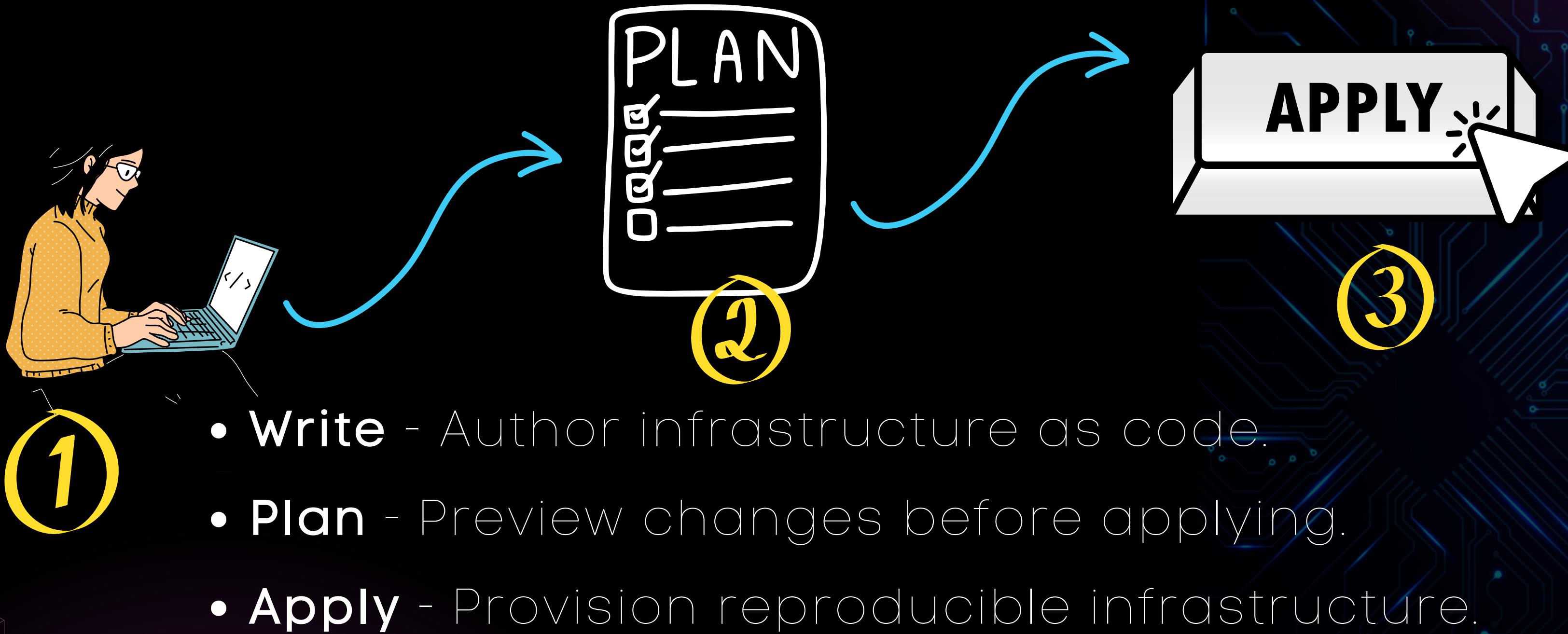


/jonahandersson



@cjkodare

Terraform Workflow

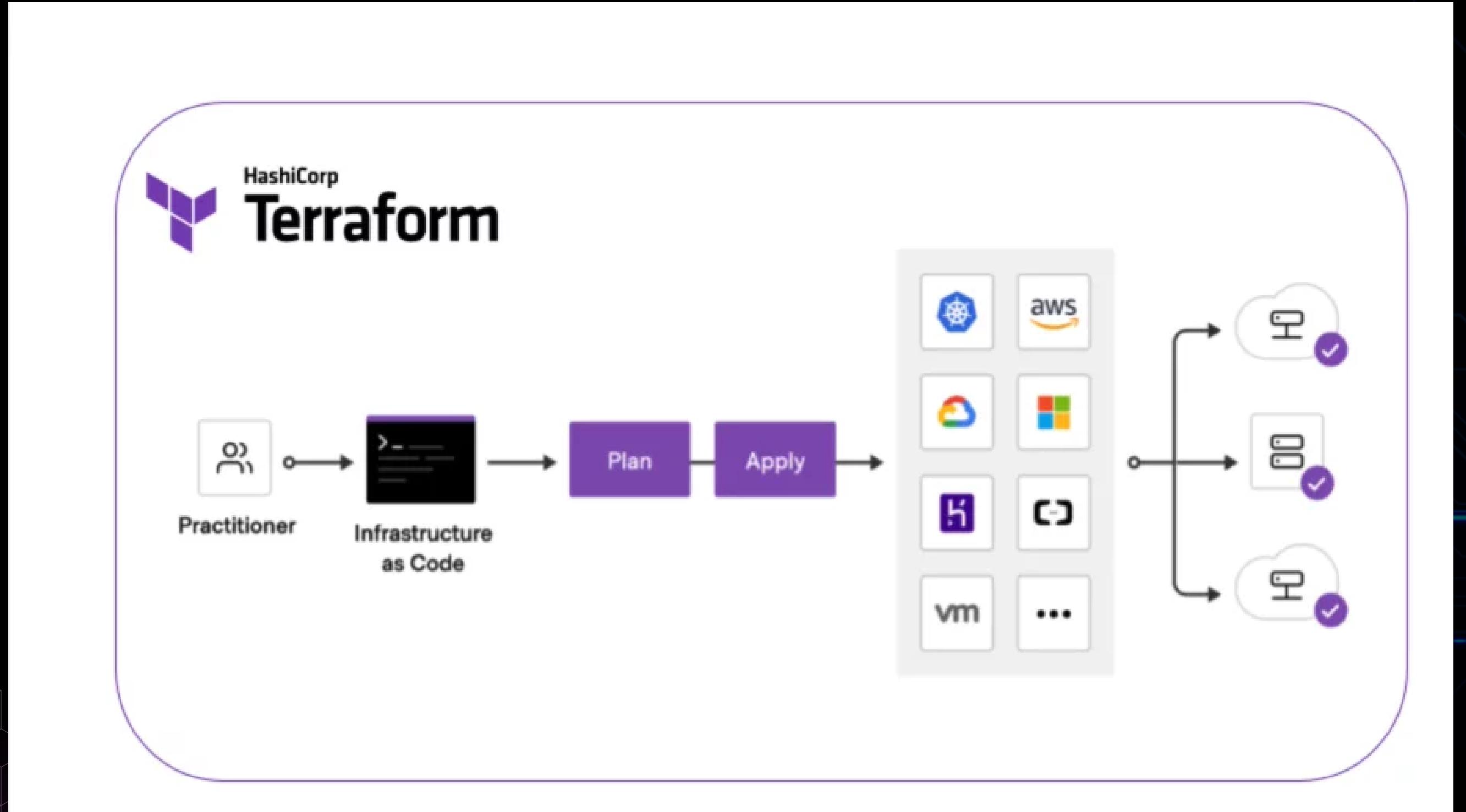


/jonahandersson



@cjkodare

Iac on Terraform's Perspective - Workflow



/jonahandersson



@cjkodare

Infrastructure as Code (IaC)

Azure Functions

Managed Identities

Terraform on Azure



/jonahandersson

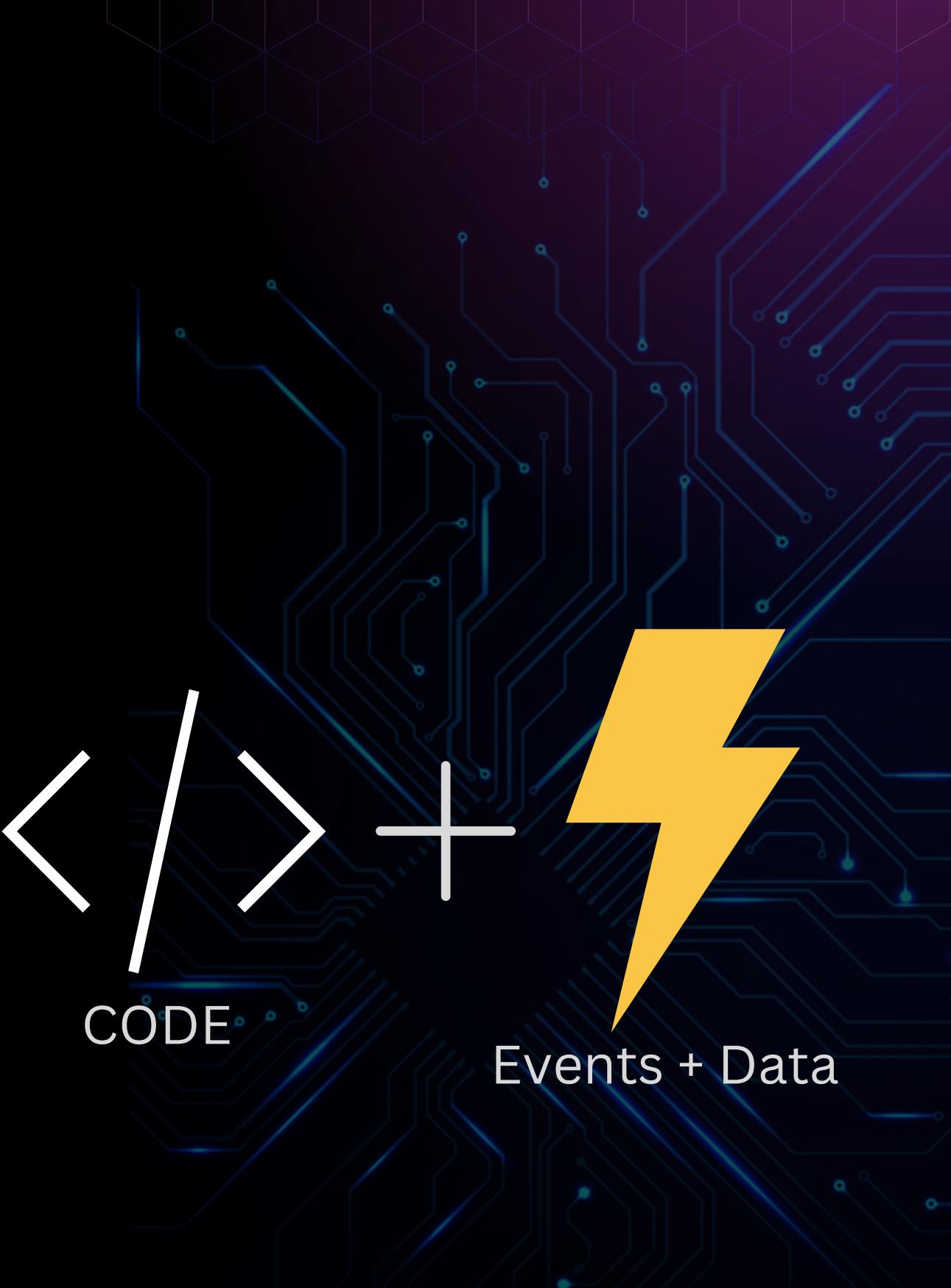


Azure Functions



- Service Compute Solution in Azure
- Allows write less code and be effective
- Maintain less infrastructure
- Event-driven apps and architecture
- Extension: Azure Durable Functions - Code

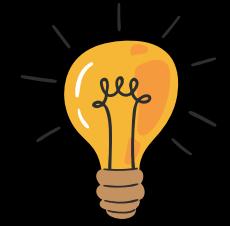
Stateful Workflows/Orchestration of events
by code



/jonahandersson



@cjkodare



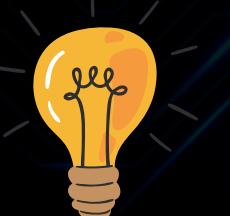
Azure Functions



Infrastructure as Code (IaC)



Managed Identities



Terraform on Azure



/jonahandersson

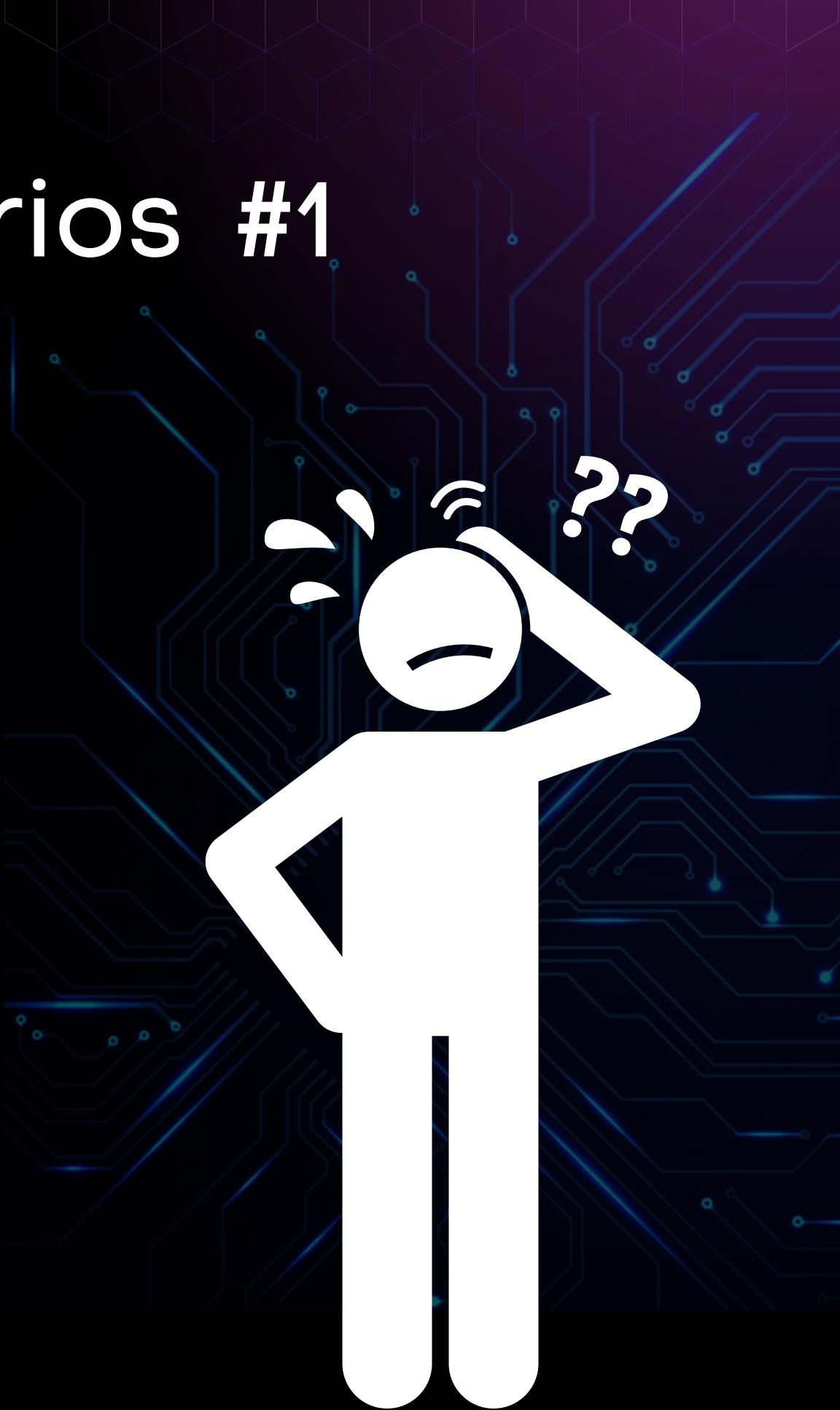


@cjkodare

A Few Common Problems & Scenarios #1



Developers using their own PAT / tokens and credentials to access, develop or manage resources in Azure

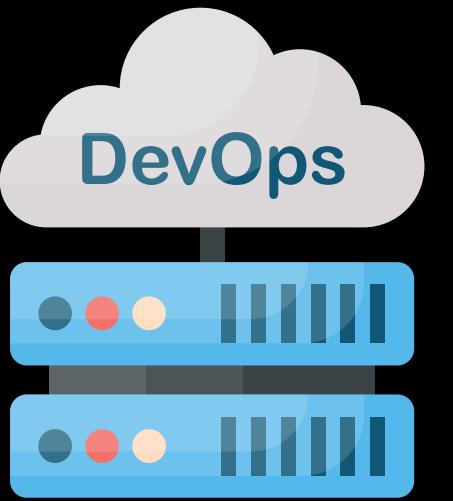


/jonahandersson

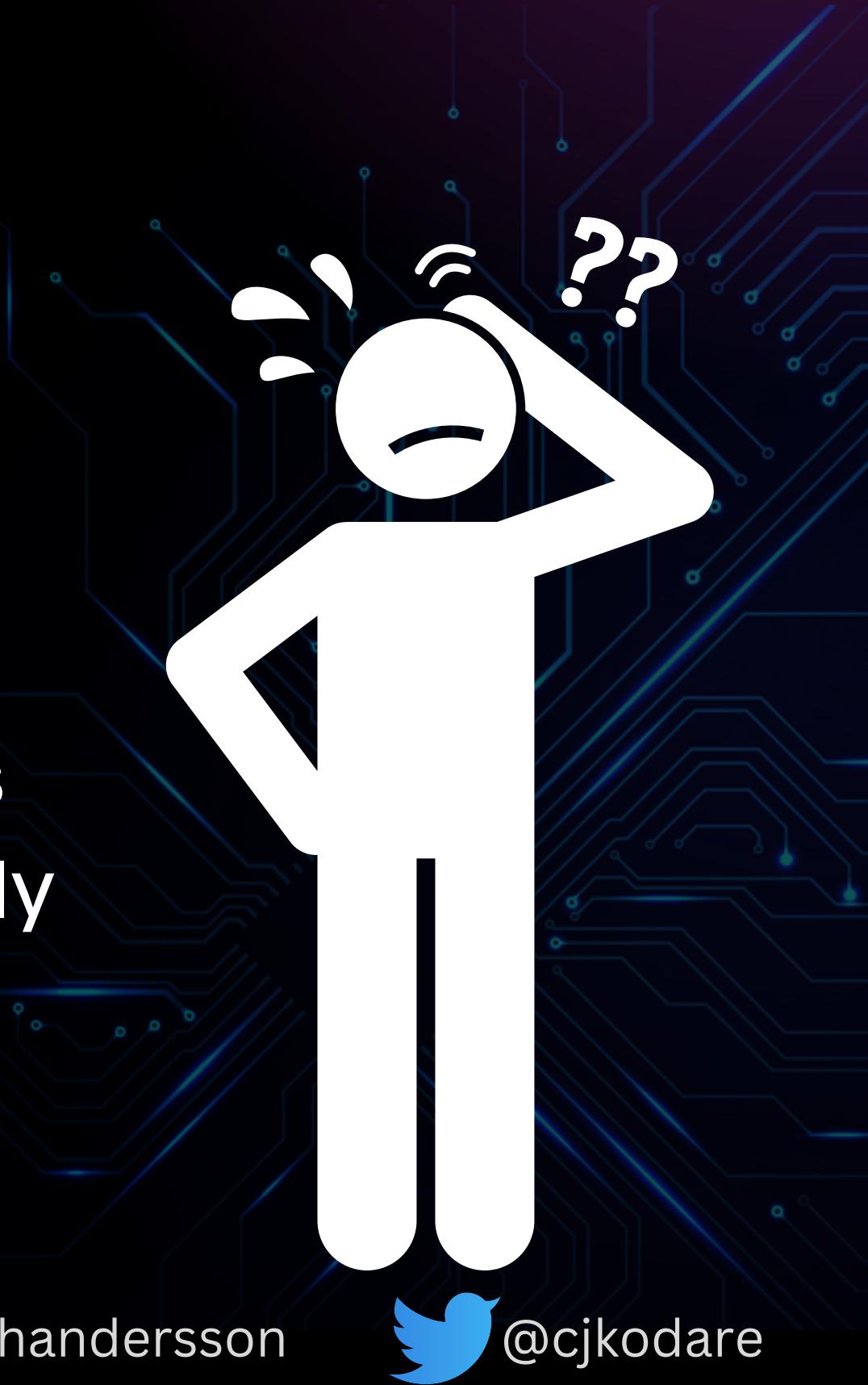


@cjkodare

A Few Common Problems & Scenarios #2



DevOps Pipelines need to use Service Principals (Azure AD) for Service Connections which is usually using client tokens/secrets that expires



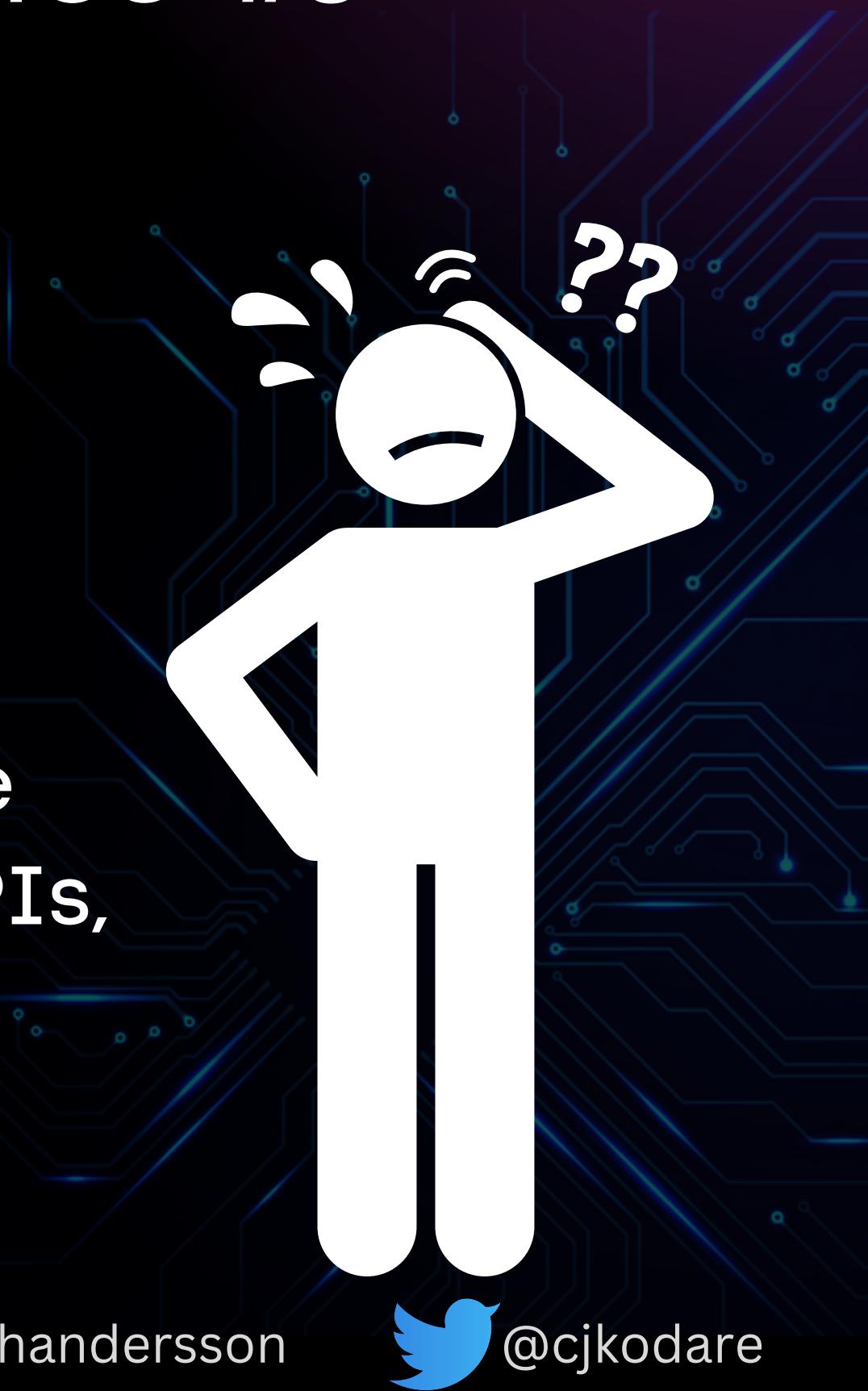
/jonahandersson



A Few Common Problems & Scenarios #3



Azure resources and Azure DevOps needs to be integrated with other services, 3rd-party apps, APIs, etc.



/jonahandersson



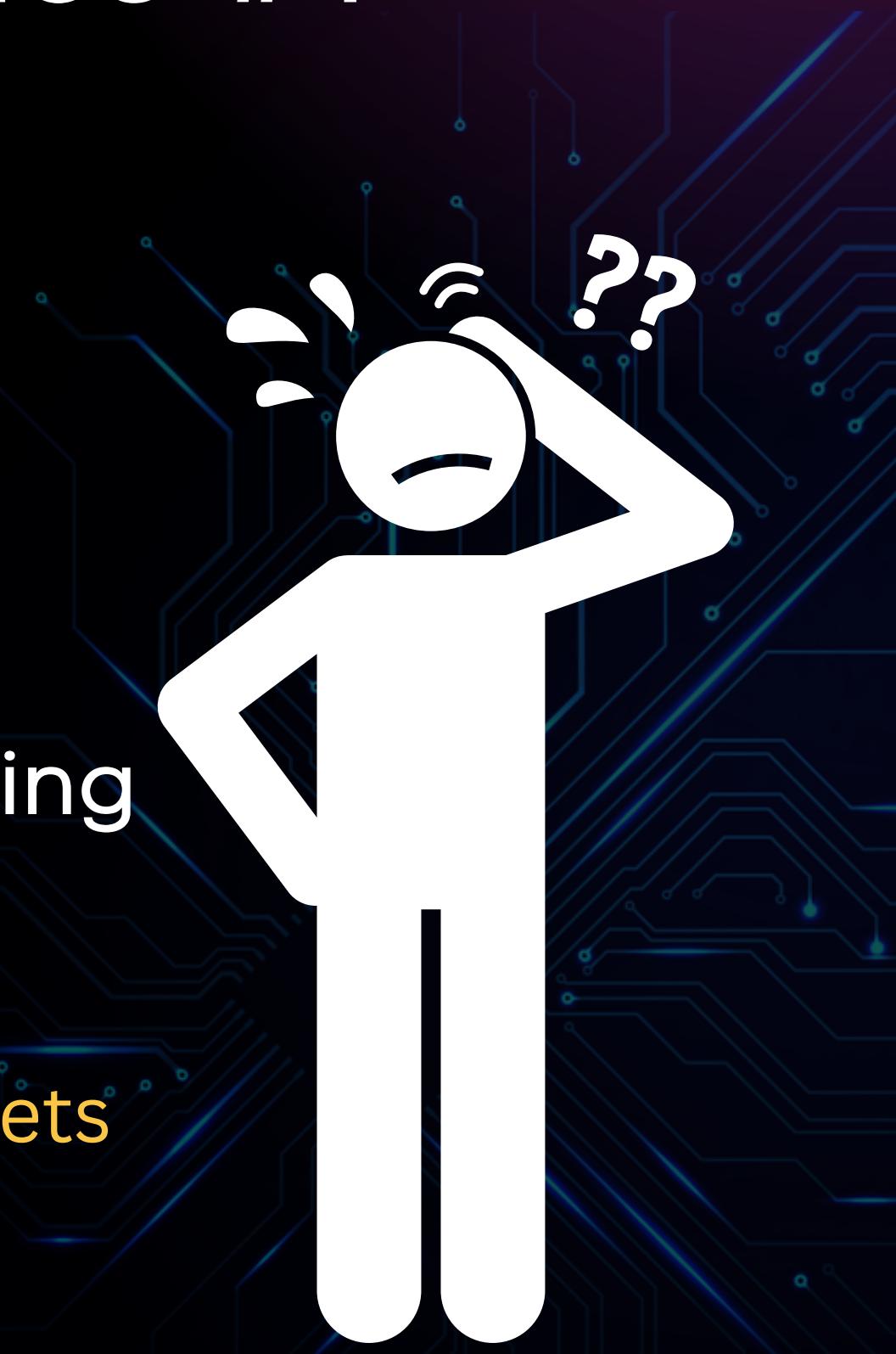
@cjkodare

A Few Common Problems & Scenarios #4



Extra hassle of manually adding, updating, deleting secrets in Azure Key Vault

Note: Azure Key Vault is best solution for storing secrets than putting it hard-coded in your code



/jonahandersson



@cjkodare

Azure AD Identity Objects & Permissions

Types

- Azure AD Users
- Azure Groups
- Enterprise Apps / Applications

Permissions

- RBAC
- Pre-defined roles, etc.
- Management Groups, etc.



/jonahandersson



Azure - Service Principals

Why Needed?

Serves as Identity Objects for applications
for any apps or entity that is not an Azure
AD user or groups



8

Typical Uses

- Service Connections for Azure DevOps Pipelines
- Integration with Azure resources
- Infrastructure as Code (non Azure) eg. Terraform



/jonahandersson



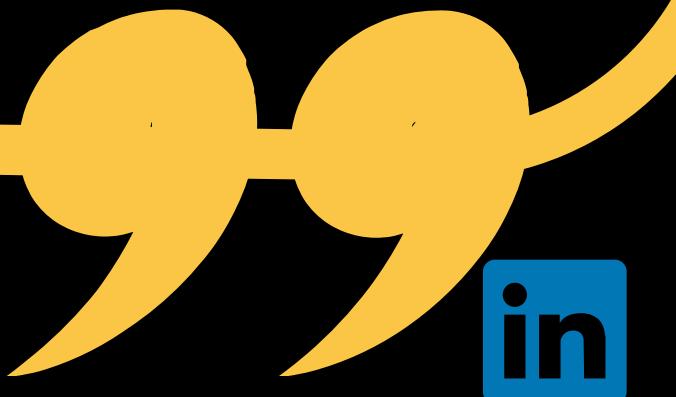
@cjkodare

Azure -Managed Identities (prev. MSI)

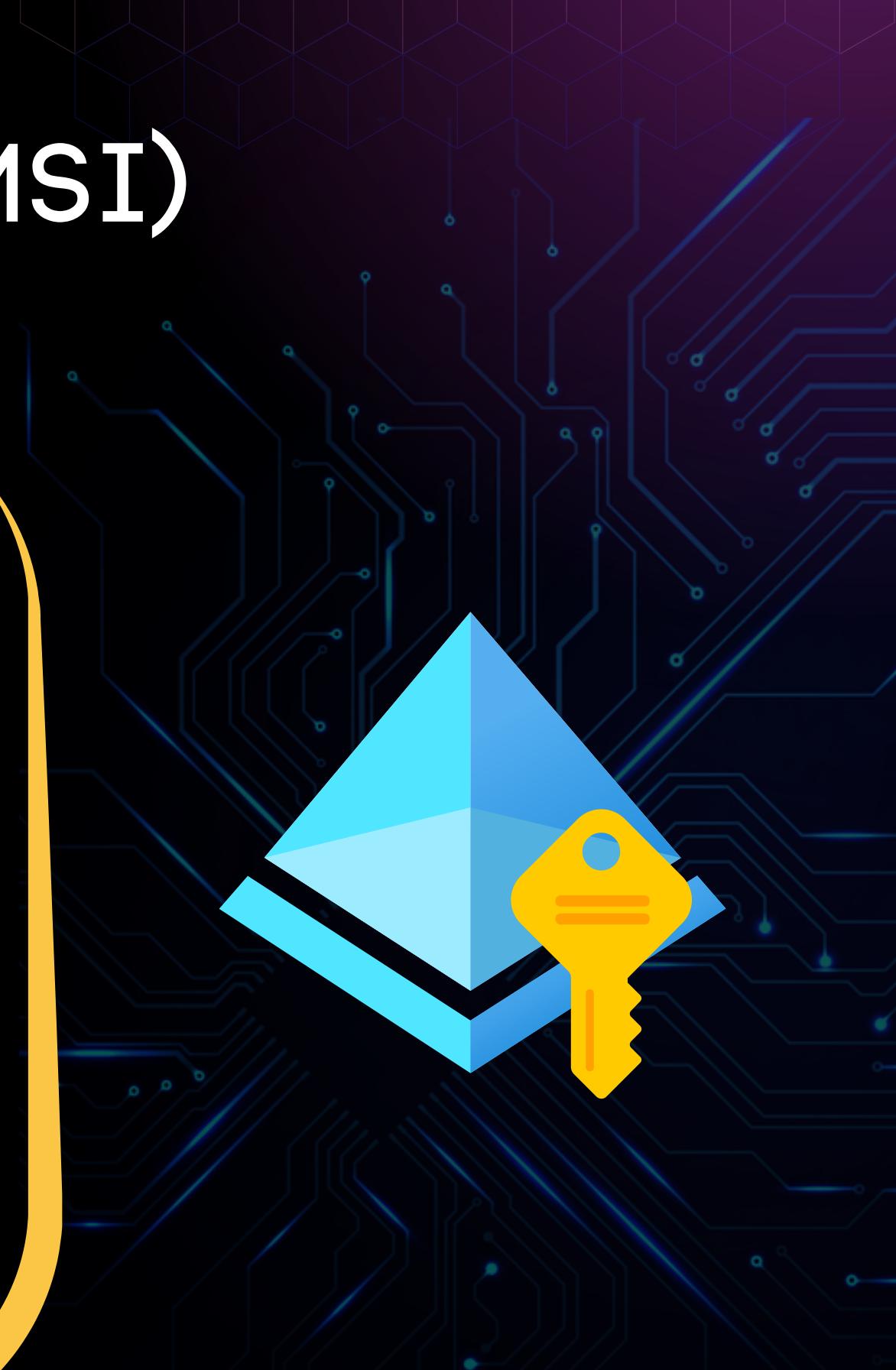
Managed identities provide an automatically managed identity in Azure Active Directory for applications to use when connecting to resources that support Azure AD authentication.

Applications can use managed identities to obtain Azure AD tokens without having to manage any credentials.

Quote from Microsoft Docs



/jonahandersson



Managed Identities Scenario Comparison

Grant storage permissions to Media Services Account



Grant a Media Services Account Key Vault access



Allow other services to access Media Services



Source: Microsoft Learn Docs



/jonahandersson

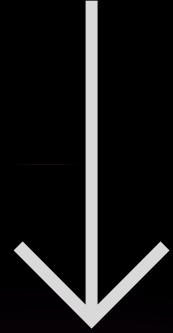


@cjkodare

Managed Identities - Types



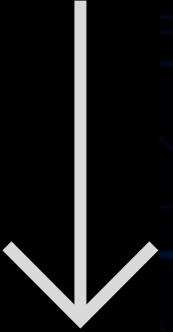
System-Assigned



Associated with an Azure Resource
e.g. Azure VM, App Service, etc.



User-Assigned



An object itself and it can be
configured to access multiple Azure
Resources



/jonahandersson

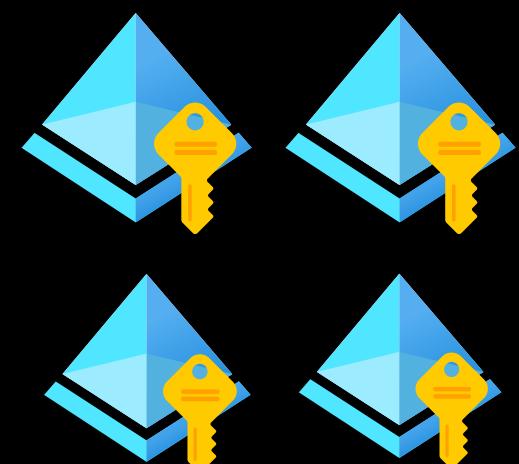


@cjkodare



Managed Identity - User Assigned

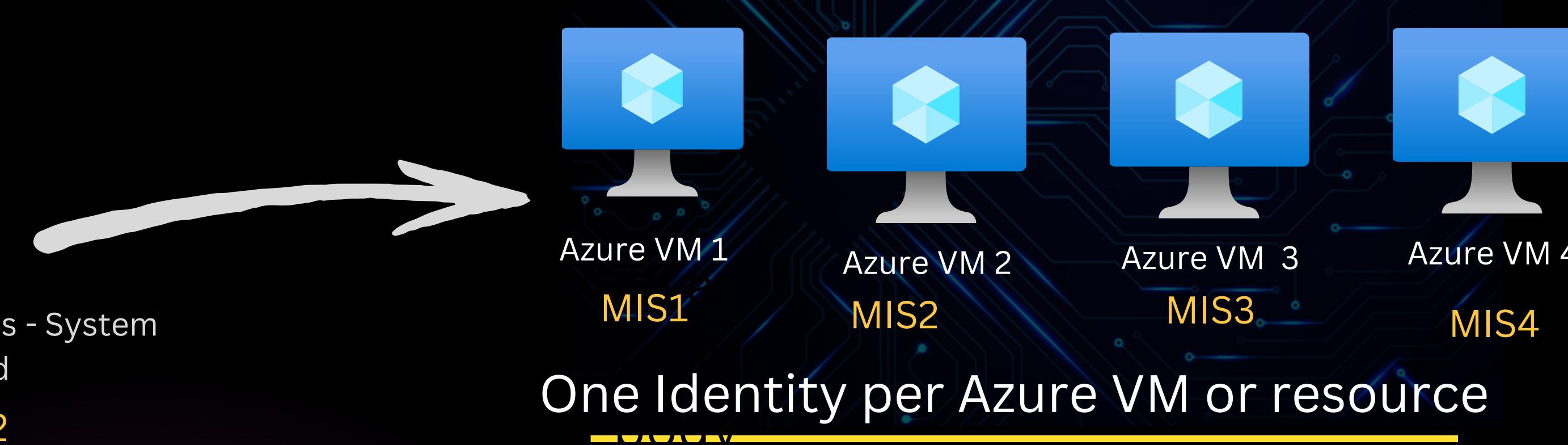
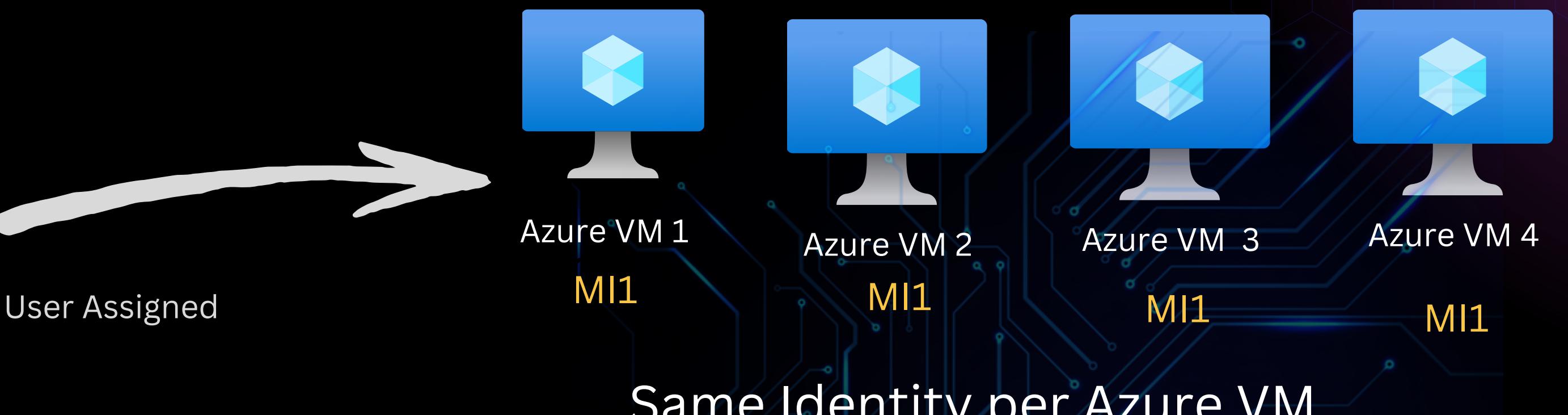
MI1



Managed Identities - System Assigned

MIS1 MIS2

MIS3 MIS4



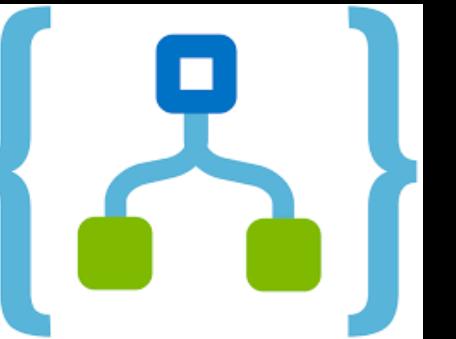
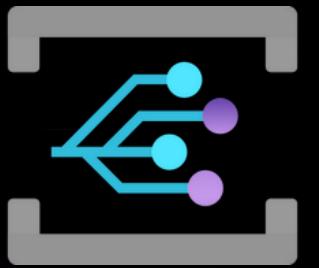
/jonahandersson



@cjkodare

Managed Identities for Azure resources

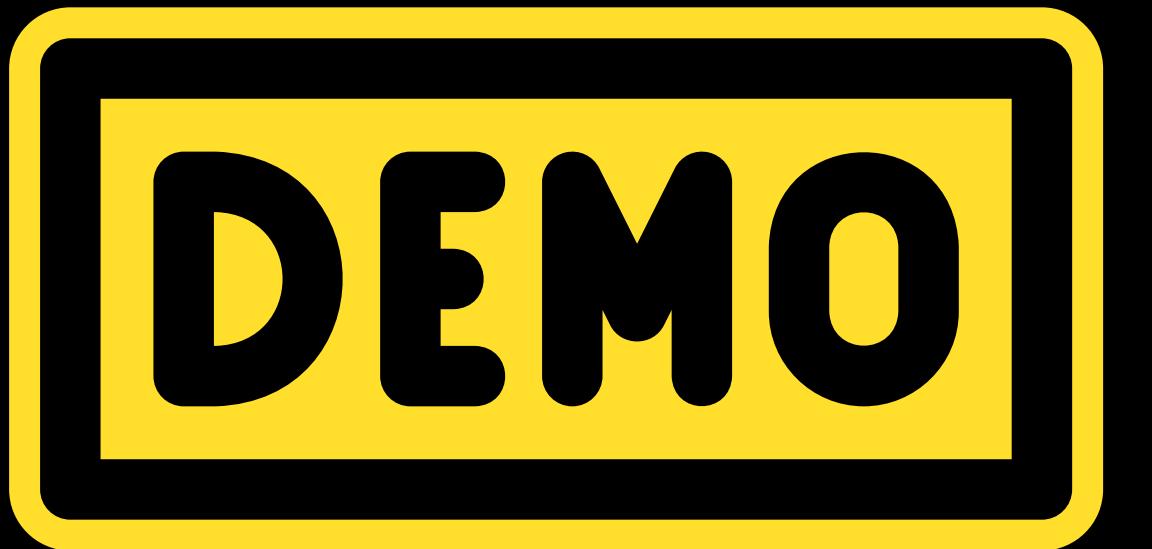
It provides an automatically managed identity in Azure Active Directory.



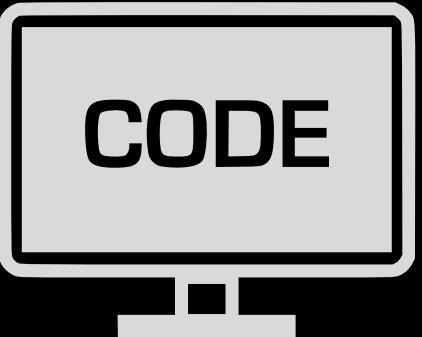
/jonahandersson



@cjkodare



DEMO



/jonahandersson





Terraform on Azure



Bicep on Azure

Learning
by doing!



/jonahandersson



@cjkodare



Azure Functions and Serverless
Microsoft Learn Path



Identity with Azure AD
Microsoft Learn Path

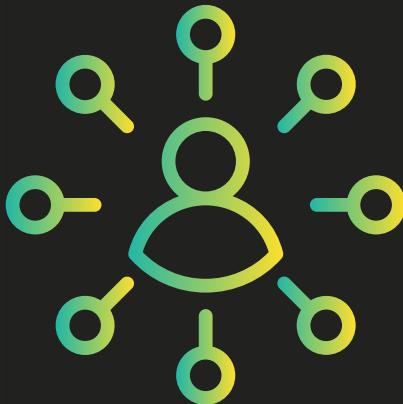
Learning
by doing!



/jonahandersson



@cjkodare



CONNECT

THANK you
VERY much!



Scan QR code to connect to me for questions,
feedback or get a copy of this presentation

