

INFRASTRUCTURE AS CODE

For Azure (and a bit about Entra ID)



TROND J. SJØVANG

- ✓ Global CCoE Lead @ Atea
- ✓ Microsoft MVP (Azure – Infrastructure as Code)
- ✓ Doesn't enjoy solving the same problem many times
- ✓ <https://trond.sjovang.no>

DISCLAIMER

Methodology is more important than tools

A lot of platforms and tools are good. Pick the ones you like best

THE JOURNEY



MANUAL TASKS



FRUSTRATION



SCRIPTS



ORCHESTRATION



EVERYTHING AS
CODE

BENEFITS OF INFRASTRUCTURE AS CODE

- Consistent deployments
- Security and standardization
- Facilitate self-service
- Speed!

CHOOSING THE “RIGHT” TOOLBOX

- Declarative
- Idempotent
- Efficient
- Easy to use – nice abstractions over using APIs directly

BICEP

- Because nobody likes ARM templates
json is nice for machines, not humans
- Azure only
(sort of. You can write extensions in C# for anything now)
- First party support (including preview services)
- Deployment Stacks is the preferred way
A bit like blueprints, but not in perpetual preview

TERRAFORM (OR OPENTOFU)

- Now owned by IBM
- OpenTofu is a CNCF supported fork
- Multicloud – but you can't write code once and deploy everywhere
- Stateful – good for efficiency, but corrupted statefiles are not fun
- Lifecycle of providers can be frustrating

AZURE VERIFIED MODULES

- Standarization of modules for Bicep and Terraform
- Sometimes feels a bit overcomplicated
- A set of guidelines and requirements for building modules
- <https://aka.ms/avm>

WHAT IF ... DEVELOPERS COULD USE THE LANGUAGE THEY ALREADY KNOW?

- Pulumi

Write code in Go, Java, JS/TS, Python, or .NET
(or YAML, but why would you want that?)

- .NET Aspire

Model applications in C# -> generate templates for IaC

ENTRA ID – “CONFIGURATION AS CODE”

- Limited support in most languages / tools
Only support for a subset of resources
- The msgraph API is *huge*
- *New!* microsoft/errafom-provider-msgraph
Very early preview, but looks promising

QUESTIONS?



<https://github.com/sjovang/breezy-devconf-2025>