

IaC Thunderdome Bicep vs Terraform!

Frank Geisler & Rune Ovlien Rakeie

Eval QR code

AGENDA

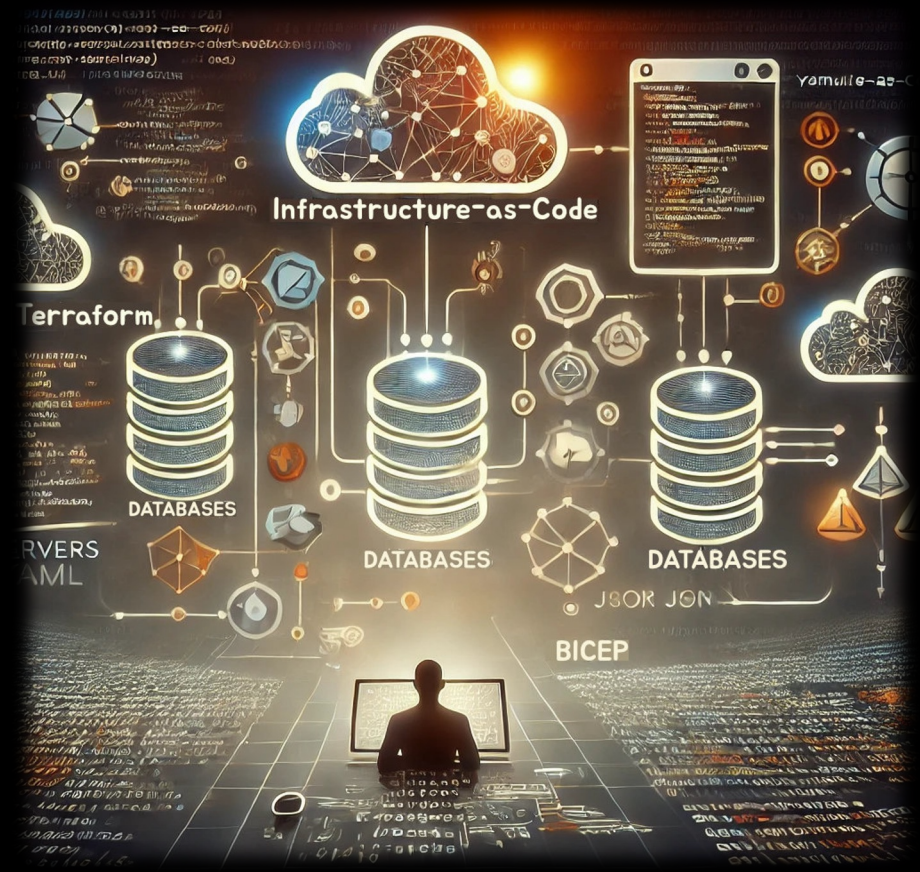
- IaC 101
- Language
- State & backend management
- CLI tools
- CI/CD



IaC 101



What is Infrastructure-as-Code?



Manage and provision computing infrastructure through machine-readable configuration files

IaC can be:

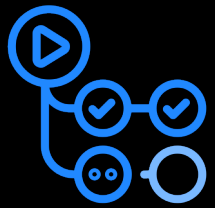
- Versioned
- Automated

Why is IaC important?



Consistency

- Identical
- Every time



Automation

- Full process
- Reduce risk
- Speed up



Version Control

- Track
- Revert
- Collaborate



Scalability

- Easy change
- Re-deploy

Icons: https://flaticon.com

IaC - Declarative vs Imperative



Imperative

- Exact instructions
- Step-by-Step
- Specify what to do
- Traditional programming



Declarative

- Declare the desired outcome
- The tool figures out the rest

Icons: https://flaticon.com

PowerShell = Imperative

```powershell
function Get-NonRetailCustomers {
  param ([array]$Customers)

  # Initialize the non-retail customers array
  $nonRetailCustomers = @()

  # Loop through the array and filter out retail customers
  foreach ($customer in $Customers) {
      if ($customer.Industry -ne "Retail") {
          $nonRetailCustomers += $customer
      }
  }
  return $nonRetailCustomers
}
```

T-SQL = Declarative

```sql
SELECT CustomerID, CustomerName, Industry
FROM Customers
WHERE Industry <> 'Retail';
```

Common IaC tools

Name	Cloud	Type
Terraform	Multi-cloud	Declarative
Bicep	Azure	Declarative
CloudFormation	AWS	Declarative
Pulumi	Multi-cloud	Imperative/Declarative

Language



Terraform

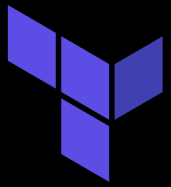
vs.



Bicep



Language Design and Structure



Strengths & weaknesses

- Readability
- Maturity
- Consistency across clouds
- Modules (reusable blocks of code)

- A bit verbose
- Steep learning curve for complex use
- Limited abstraction

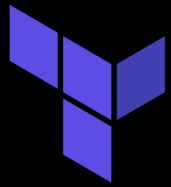


Strengths & weaknesses

- Readability
- Azure-Native
- No state file
- Modules (through nested files)

- Azure-Only
- Not so mature (… but getting there)

Readability (azurerm)



```
resource "azurerm_resource_group" "rg" {
  name     = "MyResourceGroup”
  location = "West Europe”
}

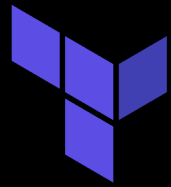
resource "azurerm_virtual_network" "vnet" {
  name                = "my-prod-network”
  location            = azurerm_resource_group.rg.location
  resource_group_name = azurerm_resource_group.rg..name
  address_space       = ["10.20.20.0/24"]
  subnet {
    name             = "subnet1”
    address_prefixes = ["10.20.20.0/25"]
  }
  subnet {
    name             = "subnet2”
    address_prefixes = ["10.20.20.128/25"]
  }
  tags = {
    environment = "Production"  }
}
```

Readability



```
resource vnet 'Microsoft.Network/virtualNetworks@2019-11-01' = {
  name: 'my-prod-network'
  location: 'westeurope'
  properties: {
    addressSpace: {
      addressPrefixes: [
        '10.20.20.0/24'
      ]
    }
    subnets: [
      {
        name: 'subnet1'
        properties: {
          addressPrefix: '10.20.20.0/25'
        }
      }
      {
        name: 'subnet2'
        properties: {
          addressPrefix: '10.20.20.128/25'
        }
      }
    ]
  }
  tags: {
    environment: 'Production'
  }
}
```

Readability (azapi)

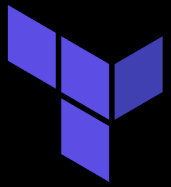


```
resource "azapi_resource" "vnet" {
  type      = "Microsoft.Network/virtualNetworks@2020-06-01"
  parent_id = azurerm_resource_group.rg.id
  name      = "my-prod-network"
  location  = azurerm_resource_group.rg.location

  body = jsonencode({
    properties = {
      addressSpace = {
        addressPrefixes = ["10.20.20.0/24"]
      }
      subnets = [
        {
          name       = "subnet1"
          properties = {
            addressPrefix = "10.20.20.0/25"
          }
        }
        {
          name       = "subnet1"
          properties = {
            addressPrefix = "10.20.20.0/25"
          }
        }

      ]
    }
  })
}
```

Language Design and Structure



Strengths & weaknesses

- Readability
- Maturity
- Consistency across clouds
- Modules (through collection of files)

- A bit verbose
- Steep learning curve for complex use
- Limited abstraction



Strengths & weaknesses

- Readability
- Azure-Native
- No state file
- Modules (through nested files)

- Azure-Only
- Not so mature (… but getting there)
- Less flexible when building modules

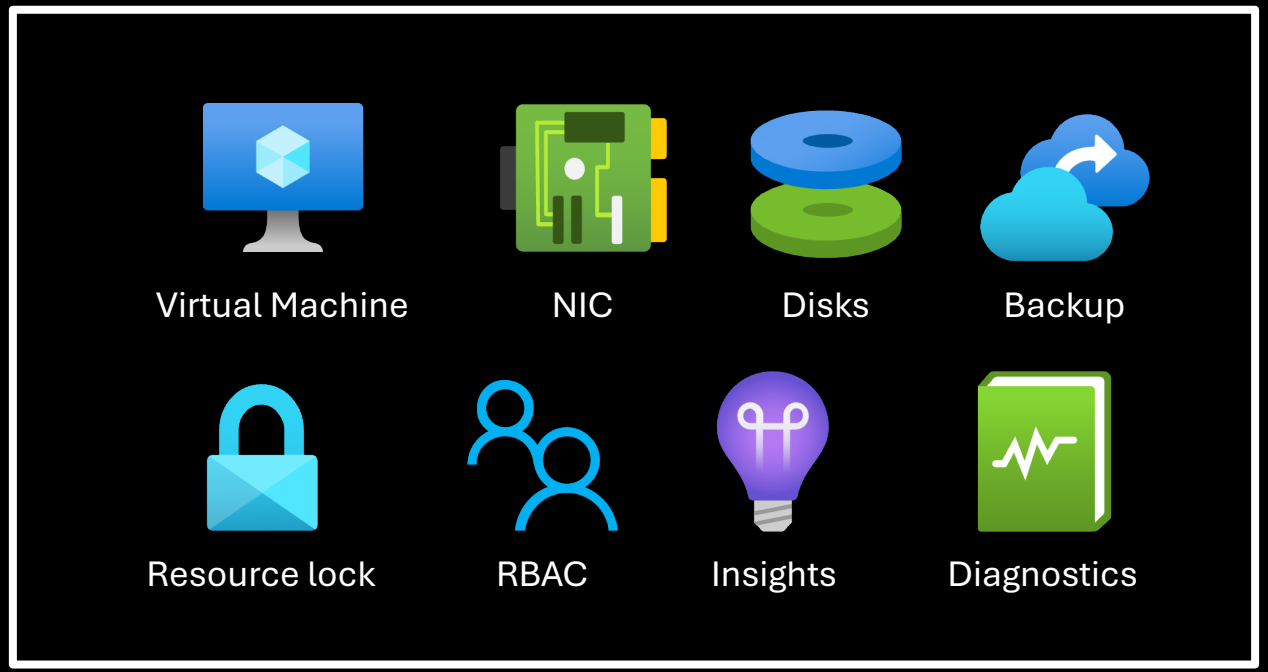
Modules

A sidetrack in the Language section

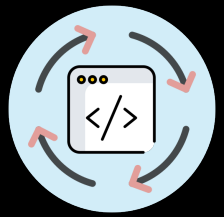


What is a module?

myAwesomeVMmodule



Why use modules?



Reuse of code



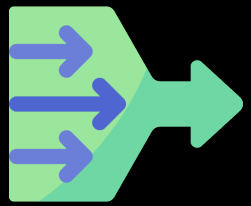
Maintainability



Logical units



Consistency



Simplified code



Version Control

Icons: https://flaticon.com

When (and not) to use modules



- Large projects
- Multiple environments
- Reusable components



- Simple projects
- Highly custom config

Icons: https://flaticon.com

aka.ms/avm

Azure Verified Modules

Search...

Azure Verified Modules

Edit page

Navigation

- Home
- Module Indexes
  - Bicep
    - Resource Modules
    - Pattern Modules
    - Utility Modules
  - Terraform
    - Resource Modules
    - Pattern Modules
    - Utility Modules
- Concepts
  - What, Why, How

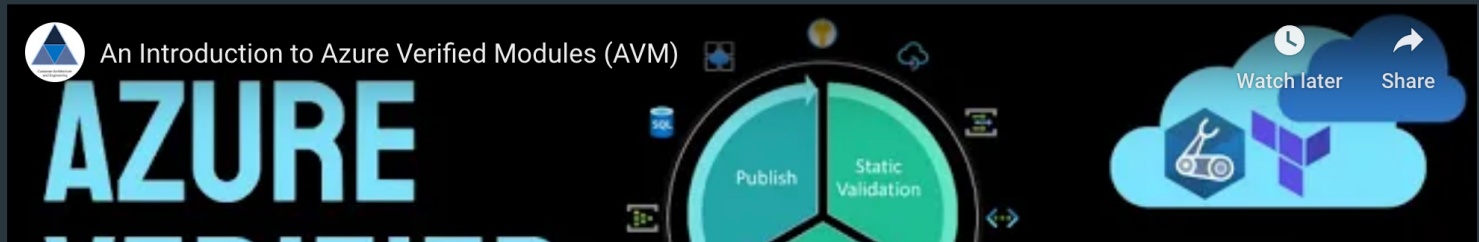
Last updated: 26 Sep 2024

Azure Verified Modules

AVM Community Calls

The AVM team has hosted 2 community calls to date.

To find out more about future calls and the recordings of previous ones please head to the dedicated page!



Navigation

aka.ms/avm

Modules

- Home
- Module Indexes
  - Bicep
    - Resource Modules
    - Pattern Modules
    - Utility Modules
  - Terraform
    - Resource Modules
    - Pattern Modules
    - Utility Modules
- Concepts
  - What, Why, How
- Specifications & Definitions

AVM Community Calls

The AVM team has hosted 2 community calls to date.

To find out more about future calls and the recordings of previou



Language

Let’s continue...



Terraform

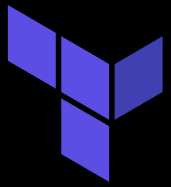
VS.



Bicep



Error Handling and Debugging



Strengths & weaknesses

- Preview changes
- Precise error messages

- State management errors

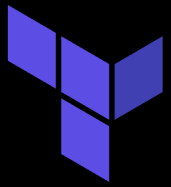


Strengths & weaknesses

- Tight integration with Azure Resource Manager
- Simplified debugging

- Predeployment plan not trustworthy
- Sometimes hard to decrypt error messages
- Different places for error logging

Extensibility and Ecosystem



Strengths & weaknesses

- Vast provider ecosystem
- Strong and active community
  - Community-created modules

- Hard to debug complex modules

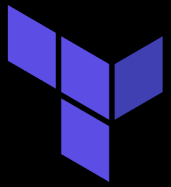


Strengths & weaknesses

- Seamless Azure integration

- Small ecosystem
- No multi-cloud support

Conclusion and Recommendation



Strengths & weaknesses

- Strong multi-cloud support
- Modularity
- Consistency across providers
- Mature
- Backed by a large ecosystem

- Verbose
- Complex for large configurations



Strengths & weaknesses

- Simpler
- Tight integration with Azure

- Azure-only
- Less mature ecosystem

Winner of Language section is...



Terraform

State & backend management



Terraform

vs.



Bicep



What is state management?

How an IaC tool tracks the resources it manages

Prevent accidental:

- Deletion
- Modification
- Recreation

How Terraform manages state



State file

- Stored locally or remote
- A snapshot of your infra

Strengths & weaknesses

- Scalability & collaboration
- Drift detection & planning

- State file management
- Security concerns

Icons: https://flaticon.com



How Bicep manages state



Azure Resource Manager

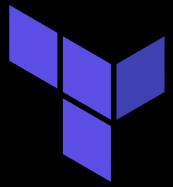
- No state file
- Relies on ARM
- State tracking abstracted

Strengths & weaknesses

- Simplicity
- Incremental deployment

- Lack of visibility
- No (trustworthy) drift-detection

Conclusion and Recommendation



Strengths & weaknesses

- Full control over state
- Powerful drift-detection
- Preview changes before deployment

- Need to manage a state file
- Contains secrets and must be secured
- Risk (though small) of file corruption



Strengths & weaknesses

- Simple to manage
- Seamless experience
- Easy to manage Azure-exclusive projects
- By default secure handling of secrets

- Less visibility
- Buggy and incomplete feature for change previews

Winner of State Management section is...



Bicep

CLI tools



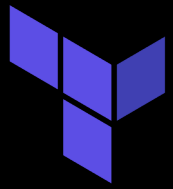
Terraform

vs.



Bicep





Terraform CLI

```
terraform
```

```
terraform plan
terraform apply
terraform destroy
terraform workspace
terraform test
```

Strengths & weaknesses

- Unified interface across providers
- Powerful state & change management
- Modular testing capabilities

- Steep learning curve
- State file dependancy
- Licensing woes? + IBM 😱



Azure CLI or PowerShell



- Azure CLI integration
- PoSh cmdlets

Deploy scopes

- Resource Group
- Subscription
- Management Group
- Tenant

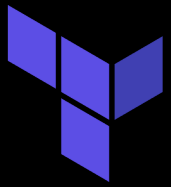
Strengths & weaknesses

- Native integration with Azure
- Simplicity for Azure users
- `What-If` used for change detection
- Free and Open-Source license

- No multi-cloud support
- No built-in testing tools

https://www.cleanpng.com/users/@chellor.html

Conclusion and Recommendation



Strengths & weaknesses

- Works across multiple clouds
- Modular testing capabilities

- Steep learning curve
- Requires careful state managment



Strengths & weaknesses

- Integrated into Azure CLI
- Ideal for Azure-only environments

- Limited to Azure
- No native testing tools

Winner of CLI tools section is...



Terraform

CI/CD



Terraform

VS.



Bicep



What is CI/CD?

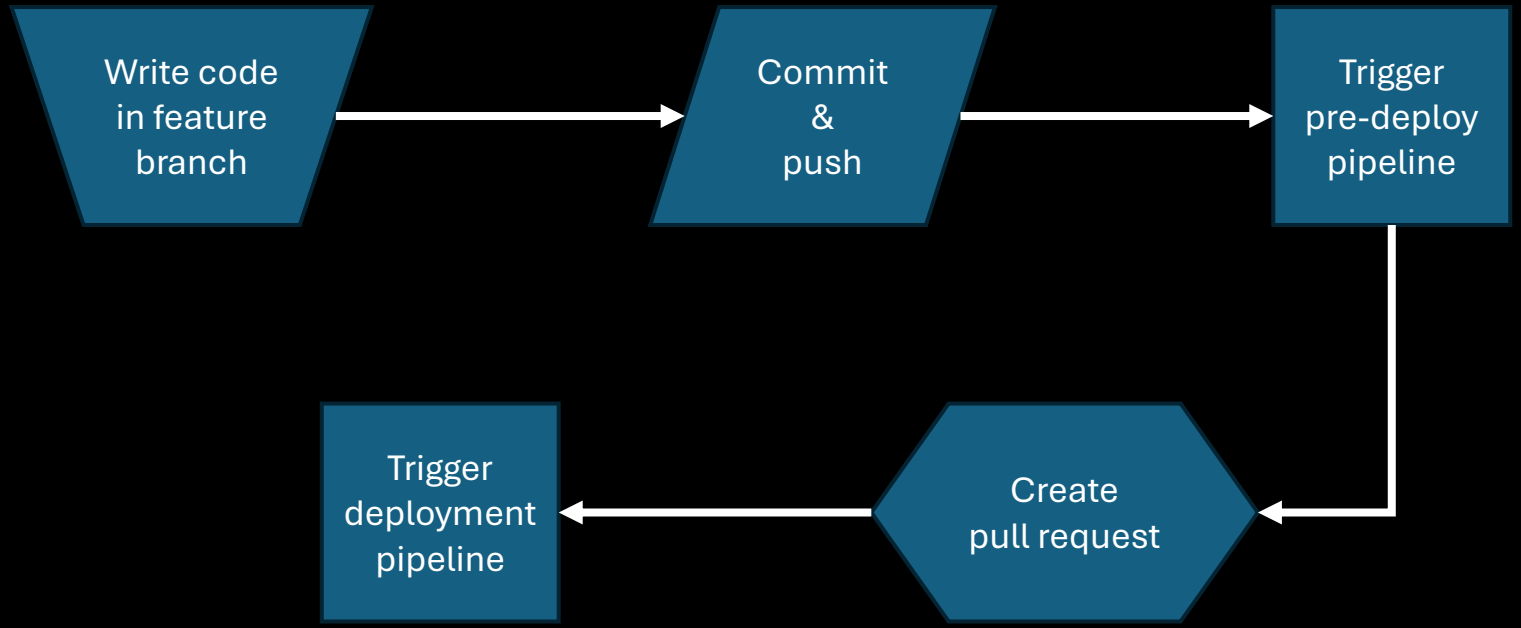
Continuous Integration (CI)

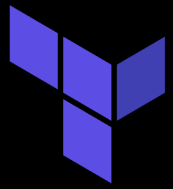
Continuous Deployment (CD)

CI = automatic verification and testing

CD = automatic deployment to target platform

A normal deployment workflow





CI/CD with Terraform

Recommended version control

- git in GitHub
- git in Azure DevOps

Compatible CI/CD tools

- GitHub Actions
- Azure DevOps
- GitLab CI
- CircleCI
- Jenkins

CI/CD testing tools

- TFLint – Config checks
- Terratest – Unit/integration test
- TFSec – Vulnerability scanning
- Trivy – Vulnerability scanning
- Checkov – Vulnerability scanning

Icons: https://flaticon.com



CI/CD with Bicep

Recommended version control

- git in GitHub
- git in Azure DevOps

Compatible CI/CD tools

- GitHub Actions
- Azure DevOps
- GitLab CI
- CircleCI
- Jenkins

CI/CD testing tools

- Bicep linter – Config checks
- PSRule – Unit/integration test
  - in combination with Pester
- Checkov – Vulnerability scanning
- Microsoft Security DevOps – Vulnerability scanning

Icons: https://flaticon.com

Winner of CD/CD section is...



Bicep



Terraform



- Member of the board of PASS Deutschland e.V.
- 30+ years experience with database technology
- MCT, long term MVP, Author

- Metal-Fan and decent BBQ Chef
- Gamer
- Spend time with my daughter

Frank Geisler

CEO
GDS Business Intelligence GmbH

• / frankgeisler

• @FrankGeisler

20 SQL KONFERENZ 2024



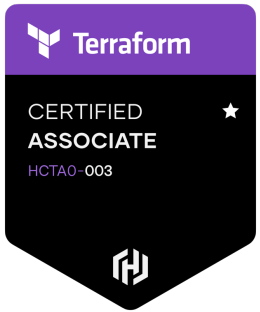
- Organizer of Data Saturday Oslo
- 30+ years experience with database technology and SQL Server specifically
- Developer, DBA, Cloud Architect

- Single-track bicycling through the forests of southern Norway

Rune Ovlien Rakeie

Principal Cloud Architect
Tietoevry

- /runeovlienrakeie
- @datarune.online





Thank you so much!

SQL KONFERENZ 2024