David Pazdera

Combining the power of Azure Verified Modules and private modules in a hybrid setup

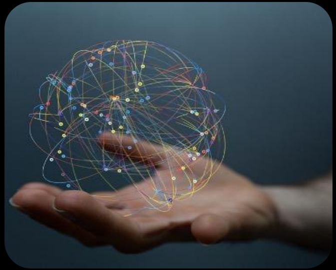
NIC
EMPOWER

About me

- solution architect @ Cegal
- meetups, conferences, ACP, communities (ALZ, Azure Arc, Bicep, AVM, Terraform in Azure)

- GitHub | LinkedIn | Sessionize | SpeakerDeck | X : pazdedav handle
- Blog: azurescholar.cloud

Today’s menu



Concepts

Infrastructure modules refresher (Bicep and Terraform)

AVM

Private Modules Library design

Demo

Featuring Bicep and GitHub combo

Building Private Modules Library

Role-play

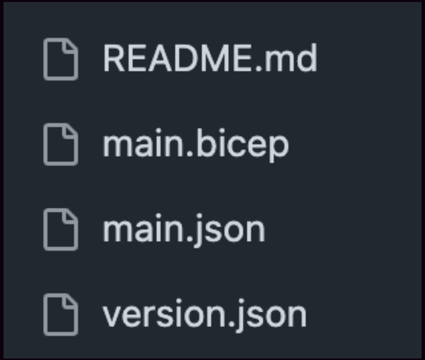
What are infrastructure modules

- composable, reusable files - set of related resources
- used in deployment templates / root modules
- embed your requirements (defined naming conventions and security requirements and policies)
- **contract** = defined input variables / parameters and outputs
- software packages for IaC world (dependency)
- authoring styles: configuration set vs. maximum customization

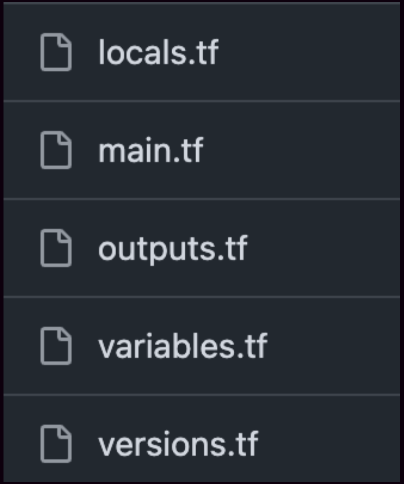
Terminology

User input	Parameters	Variables
Internal variables	Variables	Locals
User output	Outputs	Outputs
Input values files	Parameter files	TFVars files
Provider definition	Extension or Import block	Providers block
Configuration	bicepconfig.json	Terraform block

Module structure



README.md
main.bicep
main.json
version.json



locals.tf
main.tf
outputs.tf
variables.tf
versions.tf

Good practices

- az bicep format
- az bicep lint
- az bicep generate-params
- az bicep restore

- terraform fmt
- terraform validate
- terraform init | terraform get

Module sources

- Local paths
- Bicep registries (pub, priv)
- Template Specs

- Local paths
- Terraform registry (pub, priv)
- GitHub, Bitbucket, generic Git, Mercurial repo
- HTTP URLs
- S3 bucket, GCS bucket
- (package sub-directory)

Consuming modules

```
module hostPool 'br/public:avm/res/desktop-virtualization/host-pool:0.3.0' = {
  scope: resourceGroup('${workloadSubsId}', '${serviceObjectsRgName}’)
  name: 'HostPool-${time}’
  params: {
  }
}
```

```
module "vpc" {
  source  = "terraform-aws-modules/vpc/aws"
  version = "3.18.1"
  name    = var.vpc_name
}
```

Publishing modules

Bicep public registry

- N/A – Microsoft only allows ‘internal’ publishing

Bicep private registry

- ACR instance, permissions, az cli or posh

```
az bicep publish

--file storage.bicep

--target br:exampleregistry.azurecr.io/bicep/modules/storage:v1

--documentation-uri https://www.contoso.com/examplereg.html

--with-source
```

Publishing modules

Terraform public registry – registry.terraform.io

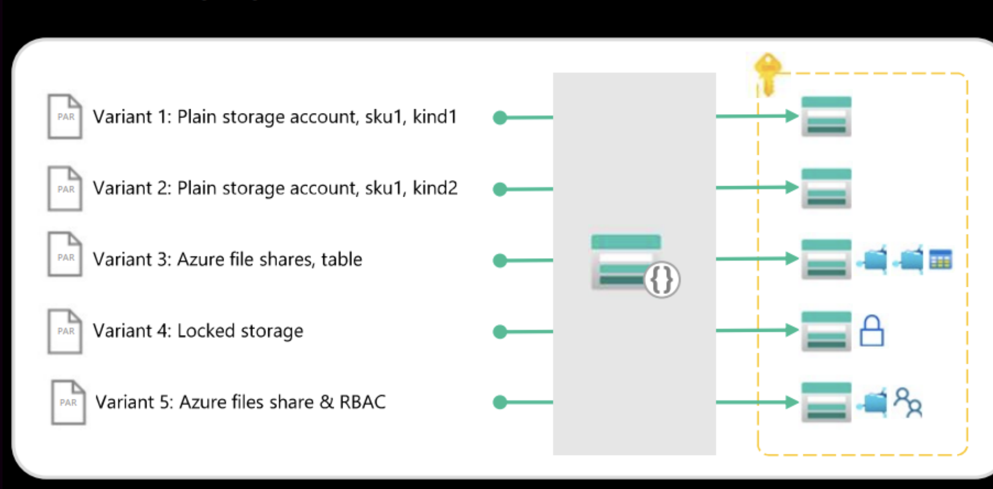
- Compliant GitHub repo (public, naming convention, 1 module per repo, standard module structure, description, x.y.z tags
- sign-in to the registry with GitHub (authorize app)
- tag-based workflow
- Community tier

Terraform private registry - app.terraform.io/example_corp

- Requires Terraform Cloud account
- Connection to VCS provider
- Tag-based vs. branch-based publishing workflow

AVM in a nutshell

- MSFT official initiative to **set the standards for IaC modules**
- Flexible, generalized, multi-purpose with integrated child and extension resources
- Resource and Pattern Modules
- Bicep and Terraform


PAR
Variant 1: Plain storage account, sku1, kind1
PAR
Variant 2: Plain storage account, sku1, kind2
PAR
Variant 3: Azure file shares, table
PAR
Variant 4: Locked storage
PAR
Variant 5: Azure files share & RBAC

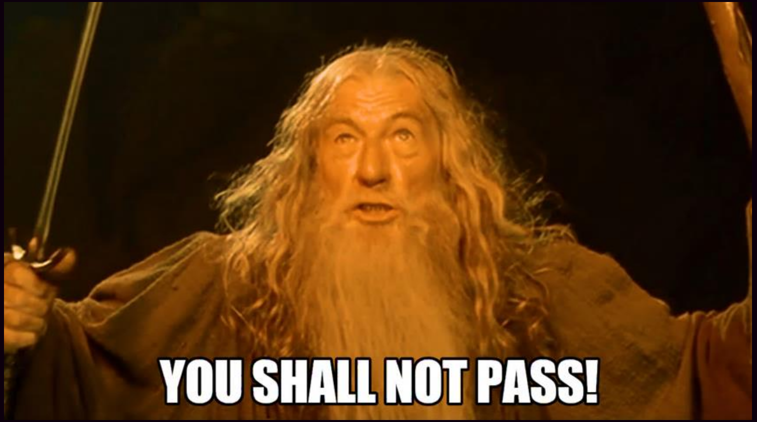
NIC
EMPOWER

Definition of Verified

- supported by MSFT CSS
- aligned to AVM specs with enforced consistency (interfaces)
- up-to-date with product roadmaps
- aligned to WAF High-priority recommendations, Reliability Hub, and APRL
- documented (with examples)
- tested

External contributions

- Modules must be owned by MSFT FTEs
- Create issue for missing module or feature
- Fork the repo and contribute via PR
- All tests must pass

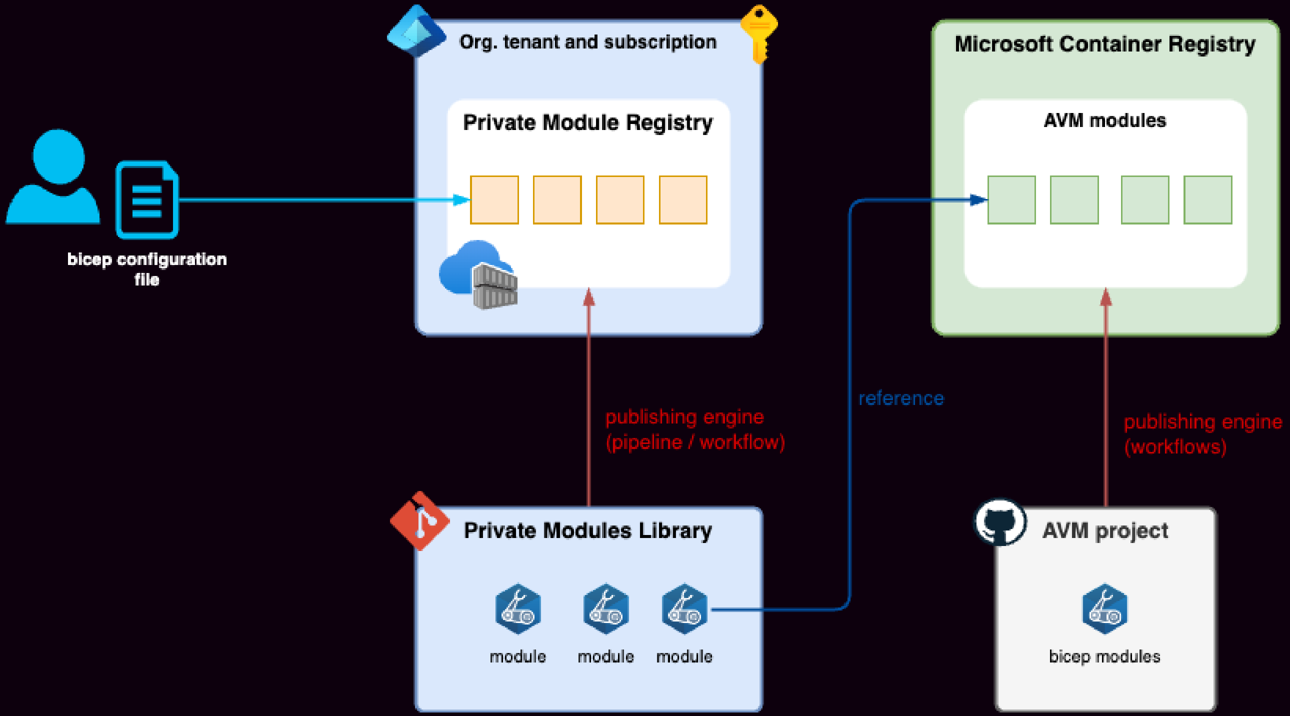


What if you…

- need a specific resource composition / module
- don't want to publish modules externally, but
- don’t want to create and maintain general-purpose resource modules, or
- need to temporarily deviate from AVM to fix a bug / enable feature

Build your own pattern modules but use AVM resource modules

Private Modules Library



NIC
EMPOWER

Building blocks [1/4]

Azure Container Registry

- SKUs:
  - Basic and Standard SKUs uses private as default
- Repositories
- AuthN: Microsoft Entra ID or keys
- AuthZ: RBAC Roles
  - Least privilege: `AcrPull`, `AcrPush`
  - Reader has ‘pull image’ permission
  - Owner and Contributor have ‘push image’ permission

Building blocks [2/4]

Code repository

- Structure
  - Bicep – can use multiple-module single-repo model
  - Terraform – single-module single-repo model
- Branching
  - Main for production version of infra modules
  - Feature branches for updates and new modules

Building blocks [3/4]

CI/CD pipelines

- Tested on both GitHub Action workflows and Azure Pipelines
- Generic scripts / CLI commands – easy to port on other pipelines
- Workflows:
  - CI – linting, validation, testing
  - CD – publishing to ACR

Building blocks [4/4]

Module Web Catalog

- Auto-generated documentation (markdown): PSDocs
- Rendering from markdown to HTML: MKDocs
- Publishing to a web service: Azure Static Apps
- Separate workflow
- Can be integrated with Entra ID

Demo time…

https://github.com/pazdedav/private-modules-library

Personas

Josh

- Cloud engineer
- module creator

Jane

- Software engineer
- module consumer

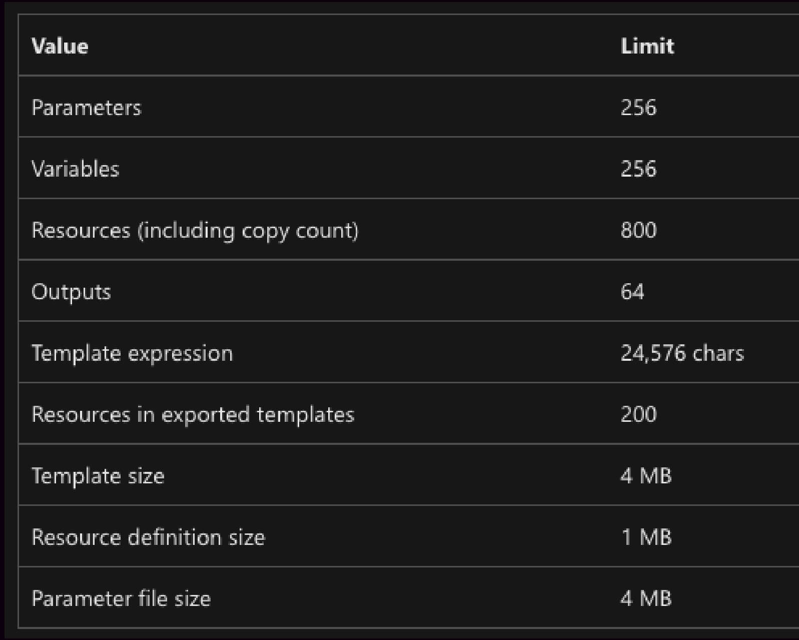
Challenges

Challenges 1/2

- **access management to registry**
  - adding MIs to ACR in ‘vending machine’
  - group memberships for engineers
- **lifecycle management – upstream modules**
  - change feed
  - all or some
  - test before publish
  - publishing *cascade*

Challenges 2/2

- flexibility can lead to complexity and verbosity
  - e.g., storage-account module (json) has 5281 lines of code
  - authoring and debugging
  - template size limits
- external dependency - software supply chain

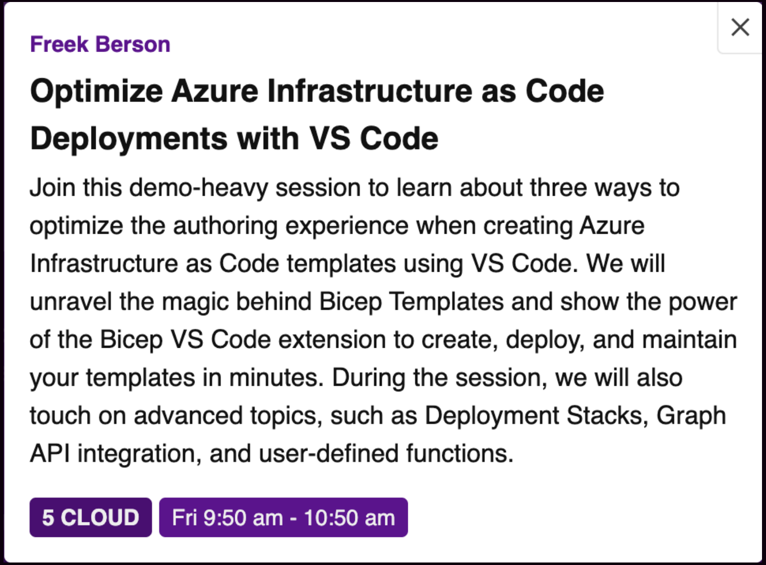


Value	Limit
Parameters	256
Variables	256
Resources (including copy count)	800
Outputs	64
Template expression	24,576 chars
Resources in exported templates	200
Template size	4 MB
Resource definition size	1 MB
Parameter file size	4 MB

Want to learn more?

- aka.ms/avm
- aka.ms/learnbicep

- https://github.com/pazdedav/private-modules-library



Freek Berson

Optimize Azure Infrastructure as Code Deployments with VS Code

Join this demo-heavy session to learn about three ways to optimize the authoring experience when creating Azure Infrastructure as Code templates using VS Code. We will unravel the magic behind Bicep Templates and show the power of the Bicep VS Code extension to create, deploy, and maintain your templates in minutes. During the session, we will also touch on advanced topics, such as Deployment Stacks, Graph API integration, and user-defined functions.

5 CLOUD | Fri 9:50 am - 10:50 am

Thank you for coming…

NIC
EMPOWER