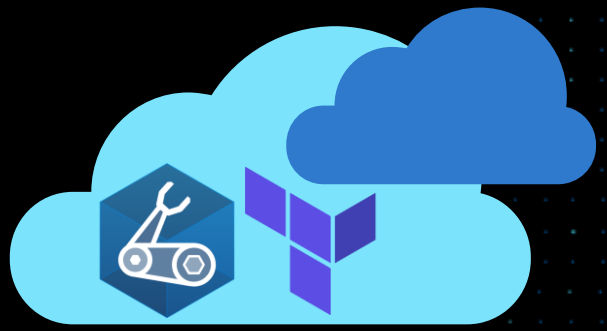
Microsoft

Azure Verified Modules (AVM)



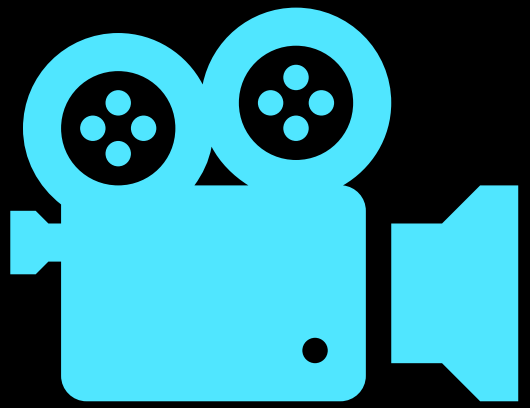
Community Call, 1st July 2025

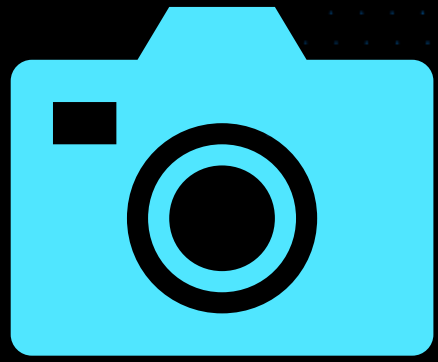
Speakers:
Microsoft: Charlie Grabiaud, Matt White, René Hézser, Jon Chancellor, Jack Tracey, Emily Redmond, Jared Holgate.

External: Nathan Bird (Servent), Aidan Hughes (Servent)

When you join this event, your name, email address and/or phone number may be viewable by other session participants in the attendee list. By joining, you're agreeing to this experience.

Also, this event will be recorded and shared publicly with others, including Microsoft's global customers, partners, employees, and service providers. The recording may include your name and any questions you submit to Q&A.





This meeting is being recorded

Meet the AVM Core Team

Technical SME's

Alex

Bilal

Chris

Erika

Jack

Jared

Jon

Matt

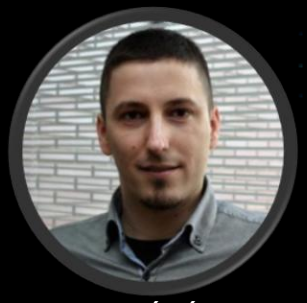
Rainer

Sebastian

René

PM's

Charlie


Máté

Pankaj

Agenda

- AVM adoption at Servent (*Jack hosting Nathan Bird & Aidan Hughes*)
- AVM Terraform updates (*Matt*)
- Terraform Migration Tool (*Jared*)
- AVM Bicep updates (*René*)
- Website updates - Usage Guide (*Jon*)
- Going v1 (*Jack*)
- Support Statement update (*Jack*)
- Copilot experiences for Infra-as-Code (*Emily*)
- Q & A

AVM Adoption at Servent

Jack Tracey

Nathan Bird
Azure Architecture Director

With over 20 years of IT experience and a decade of specialising in Microsoft Azure, Nathan, designs, implements, and leads complex projects, driving innovation and cloud transformation for our customers.

nathan.bird@servent.co.uk
https://github.com/birdnathan
https://www.linkedin.com/in/nathan-bird-servent/

Aidan Hughes
Senior Azure Engineer

Aidan has over 9 years of experience working with Microsoft Azure, specialising in DevOps and Infrastructure-as-Code to deliver complex projects at scale.

Aidan.hughes@Servent.co.uk
https://github.com/arhughes14
https://www.linkedin.com/in/aidanhughes1

Servent

CYBER ESSENTIALS CERTIFIED PLUS

UKAS MANAGEMENT SYSTEMS 8289 ISO 27001 INFORMATION SECURITY MANAGEMENT

UKAS MANAGEMENT SYSTEMS 8289 ISO 9001 QUALITY MANAGEMENT

Why Servent

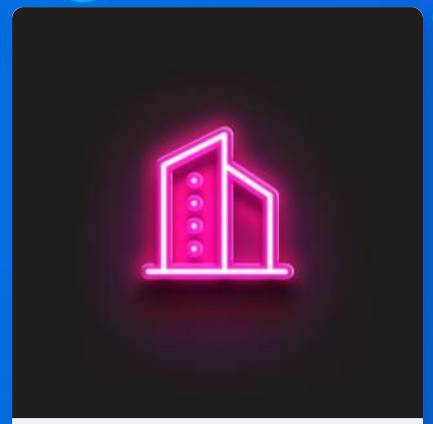
An agile partner who can mobilise in days and not months, with the skills and experience to deliver digital transformation for all businesses.

100% Azure Focused

As a leading UK Microsoft Partner and early adopter of Azure, we cover the full breadth of Azure services in depth, due to our 100% laser focus on the Azure platform.

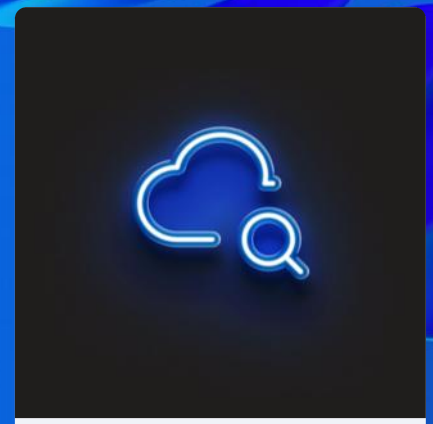
15 Years' Experience

We've been with Azure since the beginning, with 15 years' experience we offer unparalleled expertise and help some of the biggest enterprises in their adoption of the Azure platform.



Enterprise Scale

Our extensive experience with some of the largest enterprise customers enables us to bring enterprise scale thinking to businesses of any size through our Enterprise Cloud approach.



Demystifying the cloud

We work with customers of all shapes and sizes across many industry verticals and help them better understand the Azure platform and how it can transform their Business

Servent

Why do Servent champion AVM's?

Accelerated Delivery and Time Savings

AVMs significantly reduce the time required to deploy Azure infrastructure.

- **Plug and Play**
- **Shrink delivery times by up to 75% (Real-World Impact)**

Enhanced Partner Evangelism

AVMs are not just a technical asset; they're a strategic & scalable enabler for our platform engineering and partner delivery models:

- **Providing a consistent foundation for all customers**
- **Enabling other modular add-ons (e.g. AVD, AVS, Data)**
- **Supporting GitHub and Teams integration for delivery automation**

Microsoft-Supported and Secure

AVMs are maintained by Microsoft and aligned with the Well-Architected Framework (WAF), ensuring:

- **Built-in security best practices**
- **Ongoing updates and support**
- **Confidence in long-term maintainability**

This makes them a strong alternative to community modules, which may lack support or drift from Microsoft's evolving standards.

Simplified Governance and Compliance

AVMs are designed to align with Microsoft's SAFE (Secure, Agile, Fast, and Efficient) deployment model:

- **Microsoft security best practices by default**
- **Modular and composable, enabling teams to quickly assemble infrastructure patterns without reinventing the wheel**
- **Accelerate time-to-value and reduce delivery friction**
- **Reduce duplication, minimize tech debt & ops overhead**

Easier Recruitment and Onboarding

Because AVMs are standardised and well-documented, they:

- **Lower the barrier for new engineers to contribute**
- **Reduce the need for bespoke training**
- **Enable faster ramp-up on projects**

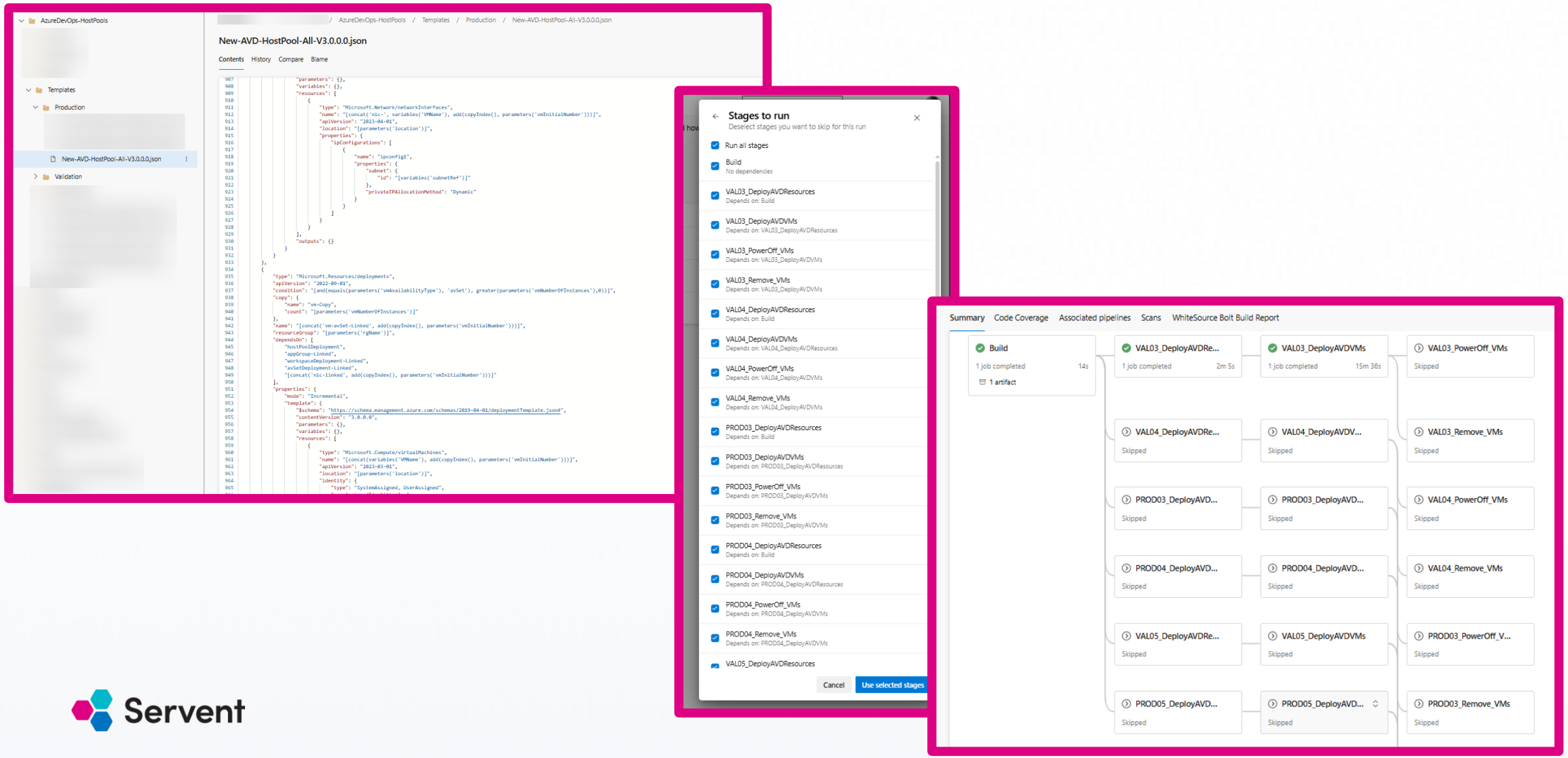
Customer Flexibility and IP Neutrality

AVMs don't lock customers into proprietary IP, which:

- **Makes conversations easier with clients**
- **Supports long-term sustainability**
- **Enables leaner delivery models**

Servent

Before AVM



Servent

```hcl
module "nsg" {
  source              = "Azure/avm-res-network-networksecuritygroup/azurerm"
  version             = "0.4.0"
  providers           = { azurerm = azurerm.avd_lz }
  location            = azurerm_resource_group.rg.location
  resource_group_name = azurerm_resource_group.rg.name
  tags                = var.tags
  name                = "nsg-${local.naming_convention}-${var.suffix}"
  security_rules      = local.nsg_security_rules
}

You, 11 hours ago | 1 author (You)
module "route_table" {
  source              = "Azure/avm-res-network-routetable/azurerm"
  version             = "0.4.1"
  providers           = { azurerm = azurerm.avd_lz }
  resource_group_name = azurerm_resource_group.rg.name
  location            = azurerm_resource_group.rg.location
  tags                = var.tags
  name                = "rt-${local.naming_convention}-${var.suffix}"
  routes              = local.routes
}


You, 11 hours ago | 1 author (You)
module "vnet" {
  source              = "Azure/avm-res-network-virtualnetwork/azurerm"
  version             = "0.8.1"
  providers           = { azurerm = azurerm.avd_lz }
  resource_group_name = azurerm_resource_group.rg.name
  location            = azurerm_resource_group.rg.location
  tags                = var.tags
  name                = "vnet-${local.naming_convention}-${var.suffix}"
  address_space       = var.address_spaces

  You, 11 hours ago | 1 author (You)
  dns_servers = {
    dns_servers = var.dns_servers
  }

  subnets  = local.subnets
  peerings = local.peerings
}
```

Servent

```hcl
module "avd_hostpool" {
  source                                      = "Azure/avm-res-desktopvirtualization-hostpool/azurerm"
  version                                     = "0.4.0"
  providers                                   = { azurerm = azurerm.avd_lz }
  resource_group_name                         = azurerm_resource_group.rg.name
  virtual_desktop_host_pool_location          = azurerm_resource_group.rg.location
  virtual_desktop_host_pool_resource_group_name = azurerm_resource_group.rg.name
  virtual_desktop_host_pool_tags              = var.tags
  virtual_desktop_host_pool_name              = "hp-${local.naming_convention}"
  virtual_desktop_host_pool_type              = var.host_pool_type
  virtual_desktop_host_pool_load_balancer_type = var.host_pool_load_balancer_type
  virtual_desktop_host_pool_maximum_sessions_allowed = var.host_pool_maximum_sessions_allowed
  virtual_desktop_host_pool_validate_environment = var.host_pool_validate_environment
  virtual_desktop_host_pool_custom_rdp_properties = var.host_pool_custom_rdp_properties
  virtual_desktop_host_pool_start_vm_on_connect = var.host_pool_start_vm_on_connect
  virtual_desktop_host_pool_preferred_app_group_type = var.host_pool_preferred_app_group_type
  virtual_desktop_host_pool_friendly_name     = var.host_pool_friendly_name
  enable_telemetry                            = false

}

# Create Azure Virtual Desktop application group
module "avd_application_group" {
  source                                                = "Azure/avm-res-desktopvirtualization-applicationgroup/azurerm"
  version                                               = "0.2.1"
  providers                                             = { azurerm = azurerm.avd_lz }
  virtual_desktop_application_group_name                = "hp-${local.naming_convention}-${local.application_group_type_short}"
  virtual_desktop_application_group_default_desktop_display_name = var.application_group_default_desktop_display_name
  virtual_desktop_application_group_description         = var.application_group_description
  virtual_desktop_application_group_type                = var.application_group_type
  virtual_desktop_application_group_host_pool_id        = module.avd_hostpool.resource.id
  virtual_desktop_application_group_resource_group_name = azurerm_resource_group.rg.name
  virtual_desktop_application_group_location            = azurerm_resource_group.rg.location
  virtual_desktop_application_group_tags                = var.tags
  virtual_desktop_application_group_friendly_name       = "${var.hostpool_name} Windows 11 Desktop"
  enable_telemetry                                      = false
}

# Create Azure Virtual Desktop workspace
module "avd_workspace" {
  source                                = "Azure/avm-res-desktopvirtualization-workspace/azurerm"
  version                               = "0.2.1"
  providers                             = { azurerm = azurerm.avd_lz }
  virtual_desktop_workspace_resource_group_name = azurerm_resource_group.rg.name
  virtual_desktop_workspace_location    = azurerm_resource_group.rg.location
  virtual_desktop_workspace_description = var.workspace_description
  virtual_desktop_workspace_name        = "ws-${local.naming_convention}"
  virtual_desktop_workspace_friendly_name = var.workspace_friendly_name
```

```hcl
module "avd_session_hosts" {
  providers                = { azurerm = azurerm.avd_lz }
  count                    = var.rdsh_count
  source                   = "Azure/avm-res-compute-virtualmachine/azurerm"
  version                  = "0.19.1"
  name                     = "${local.naming_std[avd_session_hosts]}-${format("%02d", count.index + 1)}"
  computer_name            = "${local.naming_std[avd_session_hosts]}-${format("%02d", count.index + 1)}"
  resource_group_name      = data.azurerm_resource_group.rg.name
  location                 = data.azurerm_resource_group.rg.location
  tags                     = local.tags
  os_type                  = "Windows"
  sku_size                 = var.avd_session_hosts_sku_size
  zone                     = random_integer.zone_index.*.result[count.index]
  timezone                 = var.timezone
  source_image_resource_id = data.azurerm_shared_image.avd.id
```

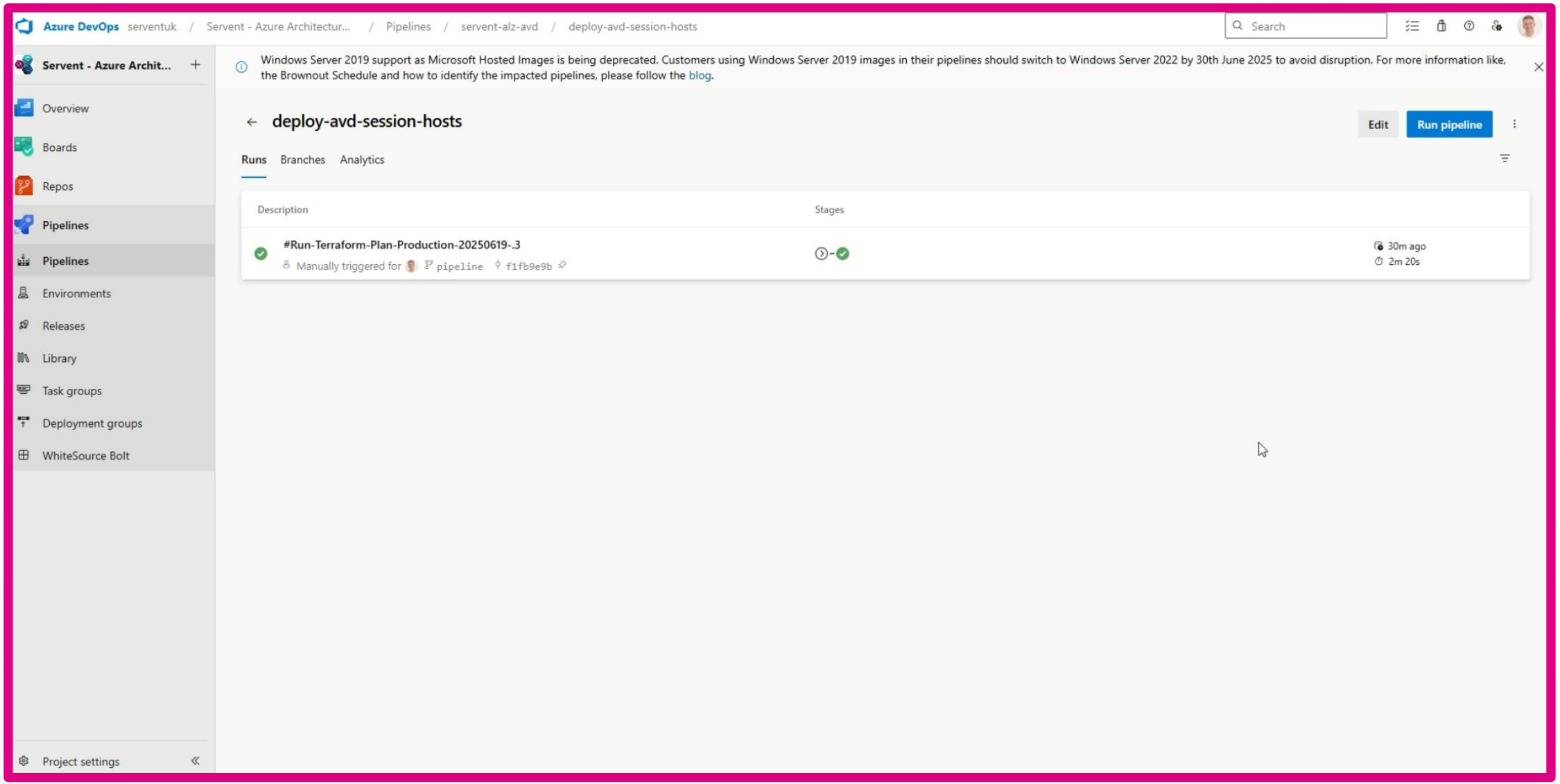
```hcl
module "avd_session_hosts" {

  os_disk = {
    name                 = "osdisk-${local.naming_std[avd_session_hosts]}-${format("%02d", count.index + 1)}"
    caching              = "None"
    storage_account_type = var.os_disk_storage_account_type
  }

  extensions = {
    entra_join = {
      name                       = "entraJoin"
      publisher                  = "Microsoft.Azure.ActiveDirectory"
      type                       = "AADLoginForWindows"
      type_handler_version       = "1.0"
      auto_upgrade_minor_version = true
      settings                   = <<SETTINGS
        {
          "mdmId" : "0000000a-0000-0000-c000-000000000000"
        }
      SETTINGS
    }
    vmext_dsc = {
      name                       = "avd_dsc"
      publisher                  = "Microsoft.Powershell"
      type                       = "DSC"
      type_handler_version       = "2.73"
      auto_upgrade_minor_version = true
      protected_settings         = <<PROTECTED_SETTINGS
        {
          "properties": {
            "registrationInfoToken": "${azurerm_virtual_desktop_host_pool_registration_info.registrationinfo.token}"
          }
        }
      PROTECTED_SETTINGS
      settings                   = <<SETTINGS
        {
          "modulesUrl": "https://wvdportalstorageblob.blob.core.windows.net/galleryartifacts/Configuration_1.0.02714.342.zip",
          "configurationFunction": "Configuration.ps1\\AddSessionHost",
          "properties": {
            "HostPoolName":"${module.avd_hostpool.resource.name}"
          }
        }
      SETTINGS
    }
```

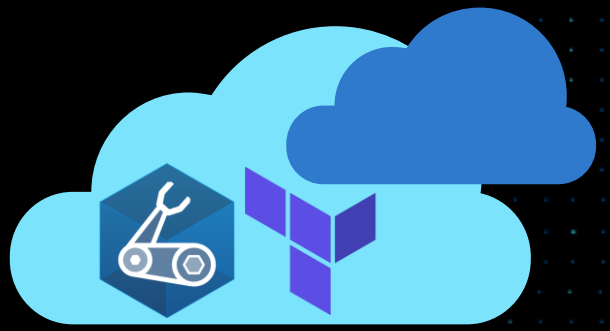
Servent

How do we help our customers adopt AVM?


Azure DevOps serventuk / Servent - Azure Architectur... / Pipelines / servent-alz-avd / deploy-avd-session-hosts
Search
Servent - Azure Archit...
Overview
Boards
Repos
Pipelines
Pipelines
Environments
Releases
Library
Task groups
Deployment groups
WhiteSource Bolt
Project settings
Windows Server 2019 support as Microsoft Hosted Images is being deprecated. Customers using Windows Server 2019 images in their pipelines should switch to Windows Server 2022 by 30th June 2025 to avoid disruption. For more information like, the Brownout Schedule and how to identify the impacted pipelines, please follow the blog.
deploy-avd-session-hosts
Edit
Run pipeline
Runs
Branches
Analytics
Description
Stages
#Run-Terraform-Plan-Production-20250619-.3
Manually triggered for
pipeline
f1fb9e9b
30m ago
2m 20s

Servent

Microsoft



AVM Terraform updates

Matt White

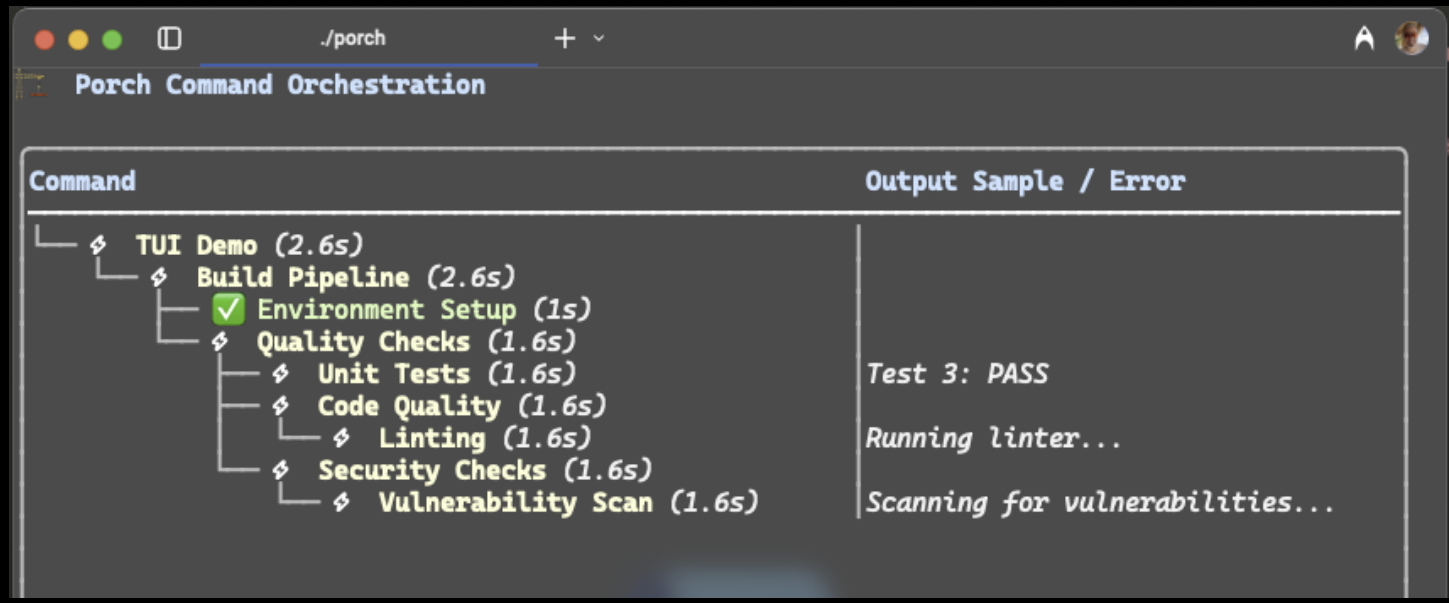
Testing and governance updates

New central repository:

[avm-terraform-governance]

Unified tooling both locally and GH actions

Tagging consistency



Microsoft

Terraform Migration Tool

Jared Holgate

Terraform State Importer Tool

A generic golang tool that can help with importing Terraform state for any Azure (azurerm or azapi) based Terraform Module, including AVM

- Guides you through the process of mapping resources and attributes
- Generates `import` blocks

- Guidance for Azure Landing Zones: https://aka.ms/alz/tf/migrate
- Tool: https://aka.ms/tf/migrate/tool

Terraform State Importer Tool

Stage 1 – Setup

- Target subscriptions and / or management groups
- KQL queries
- Exclusions
- Custom mappings where name alone is not sufficient
- Create the Target Terraform Module

Stage 2 – Resource Mapping

- Run the tool
- Examine the issues.csv
- Fix resource names in the target Terraform module
- Repeat 🔄

- For any unmatched resources decide:
  - Ignore
  - Delete
  - Delete / Recreate
- Save resolved-issues.csv

Stage 3 – Attribute Mapping

- Run the tool with the resolved-issues.csv
  - Generates import blocks
  - Generates delete blocks
- Examine the filtered plan file
- Update any attributes of resources that need to match prior settings
- Repeat 🔄

Run: `terraform apply`

Microsoft

AVM Bicep updates

René Hézser

Changelogs

- Transparency for consumers
  - Module updates only (no updates in tests)
- BCPNFR22
- CI pipeline enforces version entries
  - Existence
  - Content (Changes and Breaking Changes)
  - Version (existing and new version)
- Initial CHANGELOG.md files have been created

Changelog

The latest version of the changelog can be found here.

0.3.2

Changes

- Updated UDTs (RoleAssignments, DiagnosticsSettings, Locks) to use AVM common type
- Configured `replicaSetType` to be exportable

Breaking Changes

- None

0.3.1

Changes

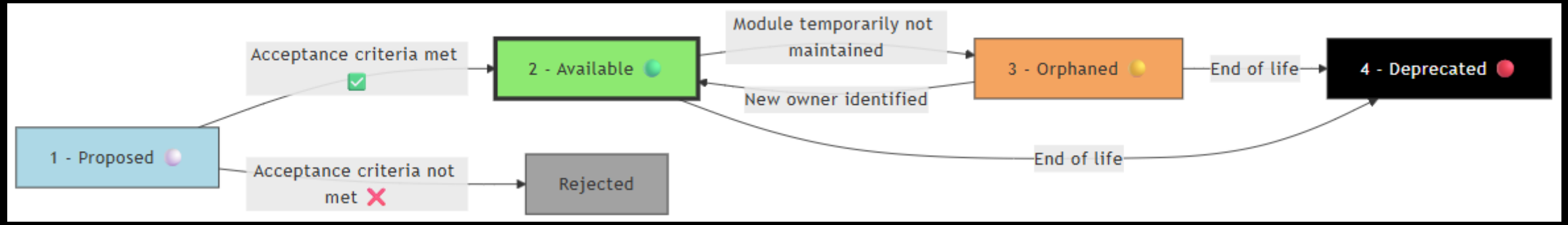
- Removed unused owner metadata

Breaking Changes

- None

More Updates

- Skipping test-cases in the CI (not advised!) - SNFR2 - E2E Testing
- Deprecation of modules - Module Lifecycle



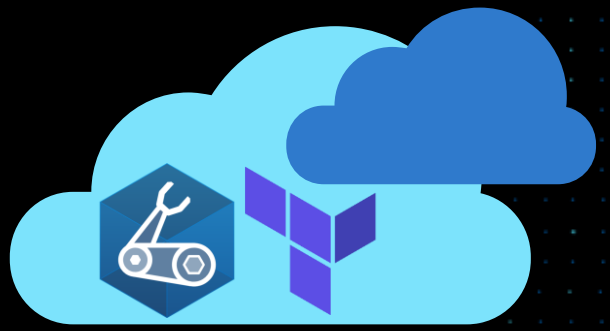
- Tooling in VSCode

```
Use more recent API version for 'Microsoft.Storage/storageAccounts'. '2022-09-01' is 1028 days
old, should be no more than 730 days old, or the most recent. Acceptable versions: 2024-01-
01 bicep core linter(use-recent-api-versions)

View Problem (Alt+F8)   Quick Fix... (Ctrl+.)   Fix using Copilot (Ctrl+I)

    7   resource storageAccount 'Microsoft.Storage/storageAccounts@2022-09-01' = {
```

Microsoft



AVM Website updates

Jon Chancellor

Content Demo

- Solution Development | AVM
- Bicep - Solution Development | AVM
- Terraform - Solution Development | AVM

Microsoft

Going v1.X.X

Jack Tracey

⚠️ But first an important reminder ⚠️

AVM Modules today are ready for production use and are being used by many in production at organizations of all shapes, sizes, and industries!

Some important things to remember 🤔

1. Bicep itself is still v0.X.X but has been supported since v0.3 in production and is used by all still today, pre-V1
2. Terraform only went V1 in 2021 but was up to v0.15.X and had again been used by all pre-V1

V1 != ready for production ❌

V1 == stability and no planned breaking changes ✅

The moment we've all been waiting for 🥁

Now AVM has been around for circa 2 years, and we've seen over **14+ million module deployments**, we are feeling confident in the AVM Framework (specs, processes etc.) that there are unlikely to be any big breaking changes to it. 👌

This means we can now **think** about **V1.X.X** for AVM modules 🥳

However, this **isn't going to be a thing that happens overnight** for all modules and there are still some dependencies and work to do for both the AVM core team to the framework and tooling and for module owners to make their modules V1 ready and complaint 🏋️

So how do we get to and make modules V1?

It will be an **opt-in** process, **per module** (owner is responsible)

- This is so that the module owner is in control and happy that the AVM module is stable and compliant with all the AVM framework tests and also the resource itself in Azure is stable from an API perspective

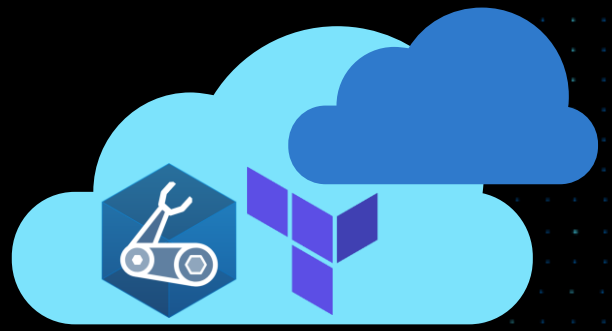
We are aiming to **allow/enable** modules to go to V1.X.X around **CY25H2**, but **still TBC** and will be announced when ready

- The date may move as the AVM core team have several features to get completed to enable, allow, and support V1 modules – e.g. support statement changes, WAF testing, spec adjustments and testing
- Also, there are some external dependencies, e.g. Bicep going V1.X.X itself 🥳 (we need to accommodate any upstream breaking changes from it, if any)

Some rules for going V1.X.X

1. This is **primarily** for resource modules initially
2. Pattern modules can go V1.X.X, **once** all the dependant resource modules are V1.X.X and these versions are used in the pattern module
3. Utility modules are **not** permitted to go V1.X.X just yet as they are still a new concept to AVM
4. All issues/PRs that should be ideally closed for a module prior to V1, or at least responded to sufficiently
   - There may be some that are left open as require more evidence or require a major refactor of the module (if so, this should be done **before going V1.X.X**)
   - Compliance with the new support statement SLAs should be demonstrated also
5. We need to see some active deployment telemetry of the module outside of Microsoft consumption e.g. 20+ customer deployments

Microsoft



Support Statement update

Jack Tracey

We are changing the support statement!

We know today that we haven't been meeting the current SLAs/targets we defined for issues and PRs 😭

And this is leading to frustrated consumers of AVM modules, issue raisers and contributors 😤

Azure Verified Modules are supported by the AVM teams, as defined here, using GitHub issues in the following order of precedence:

1. Module owners/contributors
2. If there is no response within 3 business days, then the AVM core team will step in by:
   - First attempting to contact the module owners/contributors to prompt them to act.
   - If there is no response within a further 24 hours (on business days), the AVM core team will take ownership, triage, and provide a response within a further 2 business days.
3. In the event of a security issue being unaddressed after 5 business days, escalation to the product group (Bicep/Terraform) to assist the AVM core team, will occur to provide additional support towards resolution; if required.

The new support statement is…

Module support is based on type of issue/request:

Note: *Module owners do go on holiday or periods of leave from time to time, in these times the AVM core team will attempt to triage issues based on the below on behalf of module owners.*

For bugs/security issues:
•**5 business days** for a meaningful response and ETA to be provided for fix by module owner (could be past the 5 days)

- For issues that breach the 5 business days, the AVM core team will be notified and will respond to the issue within an **additional 5 business days to assist in triage.**
- For security issues, the Bicep or Terraform Product Groups MAY step in to resolve security issues, if unresolved after a **further additional 5 business days.**

For Feature Requests:
•**15 business days** for a meaningful response and initial triage to understand the feature request. An ETA MAY be provided by the module owner.

Important: *AVM is open-source, therefore, contributions are welcome via Pull Requests from anyone in the world* 🌐

More Info & Sharing With Customers (if asked)

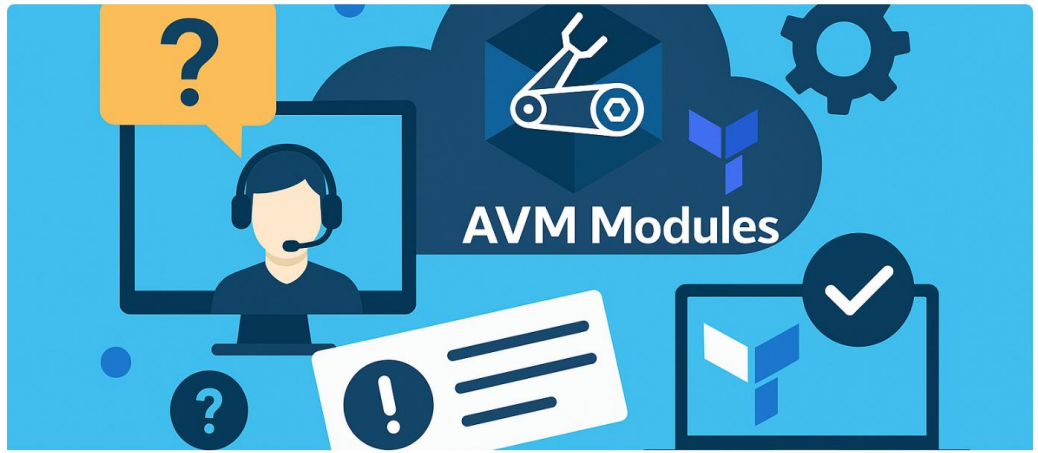
Checkout our blog post:

[Tech Community | Azure Verified Modules: Support Statement & Target Response Times Update]

Register Sign In

Azure Tools Blog

Search this community



AZURE TOOLS BLOG 4 MIN READ

Azure Verified Modules: Support Statement & Target Response Times Update

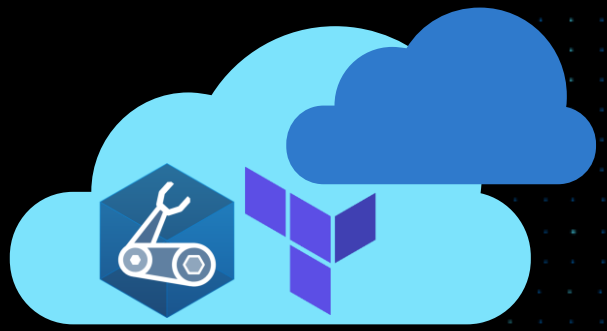
jtracey93msft MICROSOFT
Jun 09, 2025

In preparation for allowing V1.X.X AVM modules, we are updating our support statement and target response times

We are announcing an update to the Azure Verified Modules (AVM) support statement. This change reflects our commitment to providing clarity alongside timely and effective support for our community and AVM module consumers.

Microsoft

Copilot experiences for Infra-as-Code



Emily Redmond

Update: Bicep AI Progress

1. We have released initial Bicep copilot support in the **Azure Portal** and through the **VSCode extension** **GitHub Copilot for Azure**
   GitHub Copilot for Azure demos:
   - “Define a storage account in Bicep using the most recent API and descriptive placeholders for any unique values”
   - Define a storage account in Bicep using Azure Verified Modules AVM

1. We plan to release further Bicep agent support, likely through MCP, in the coming months and are working with AVM team to improve AVM support

Call for Feedback: AVM Scenarios and Tools

User Scenario
1. Generate IaC for a defined list of resources and **default to AVM modules** for resources wherever possible
2. Generate IaC code for a defined list of resources and **use AVM modules for specific resources when requested by the user**
3. Use AI chat to **find and explain potentially relevant AVM modules**

Call for Feedback: Bicep AI

Please continue to share your demos and feedback!

You can reach me to share Azure IaC copilot feedback here: emilyredmond@microsoft.com

Resources:

- Feedback form: https://forms.office.com/r/ZdyVFhhNFs
- Bicep AI Overview: https://github.com/Azure/bicep/issues/17205

Q & A

Questions from GH

JoelMiller74 last week

My company moved from AWS to Azure without knowing about Well Architected Framework, Cloud Adoption Framework, or the Azure Landing Zones. Since joining the company, I have been hoping to add the Azure Landing Zones to their tenant but I know you are going to AVM v1.0 and moving all of the ALZ Bicep installs to use new AVM modules.

I would love to hear if I can implement the current ALZ using Bicep now and later hopefully UPGRADE those to the new ALZ Bicep using AVM or it would be too different to have an upgrade path. I don't want to wait for the release but I would if there is no upgrade path from the current ALZ install to the new ALZ using Bicep AVM.
-Thanks!

Getting Involved – aka.ms/AVM

AVM is open for everyone to contribute

- Devolved ownership, not centralised!
- AVM welcomes contributors from all over the world!

Learn

- AVM Resources – aka.ms/AVM/resources
  - Labs, blog posts, podcasts, videos and more
- Leverage aka.ms/AVM/specs & aka.ms/AVM/contributing
- Stay informed – aka.ms/avm/monthly/latest

Contribute

- Identify which proposed modules you would like to contribute to:
  - Bicep AVM modules looking for contributors
  - Terraform AVM modules looking for contributors
- Propose a new module: aka.ms/AVM/ModuleProposal