

Thank you to our Sponsors

THALES



FUSION5
Business Solutions



Follow us: @ExpertsLiveAUS

Introduction to CARML and how it can help your organization adopt Azure Bicep and Infrastructure as Code

Ahmad Abdalla and Tao Yang

Follow us: @ExpertsLiveAUS



Speakers

Ahmad Abdalla



- Principal Technology Consultant at Microsoft
- Primary focus on Cloud Infrastructure and Automation
- Twitter: @ahmadkabdalla
- GitHub: <https://github.com/ahmadabdalla>

Tao Yang



- MVP – Microsoft Azure
- Twitter: @MrTaoYang
- Blog: blog.tyang.org
- GitHub: <https://github.com/tyconsulting>



Common Azure Resource Modules Library (CARML)

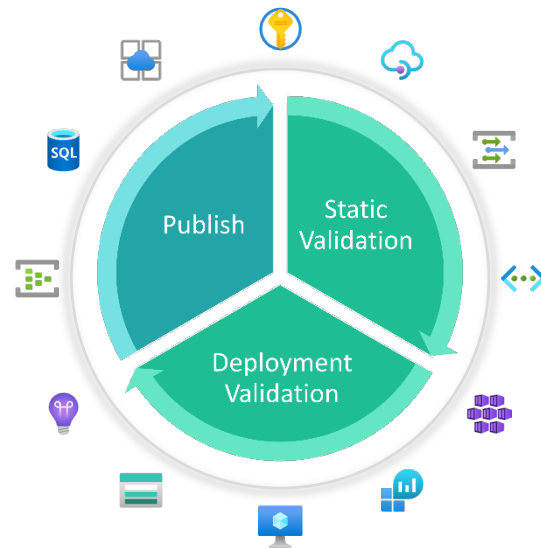
A library of comprehensive, reusable, Bicep-based building blocks to deploy Azure resources.

A continuous integration (CI) framework including static code analysis and deployment validation before publishing known good modules.

Supports GitHub & Azure DevOps.

Value proposition:

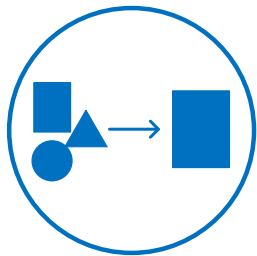
- Enable consistent solution development and delivery.
- It accelerates solution development and delivery over time.



CARML

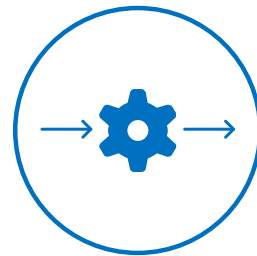


CARML in its simplest form



Resource Modules

"the building blocks that are used to construct solutions"

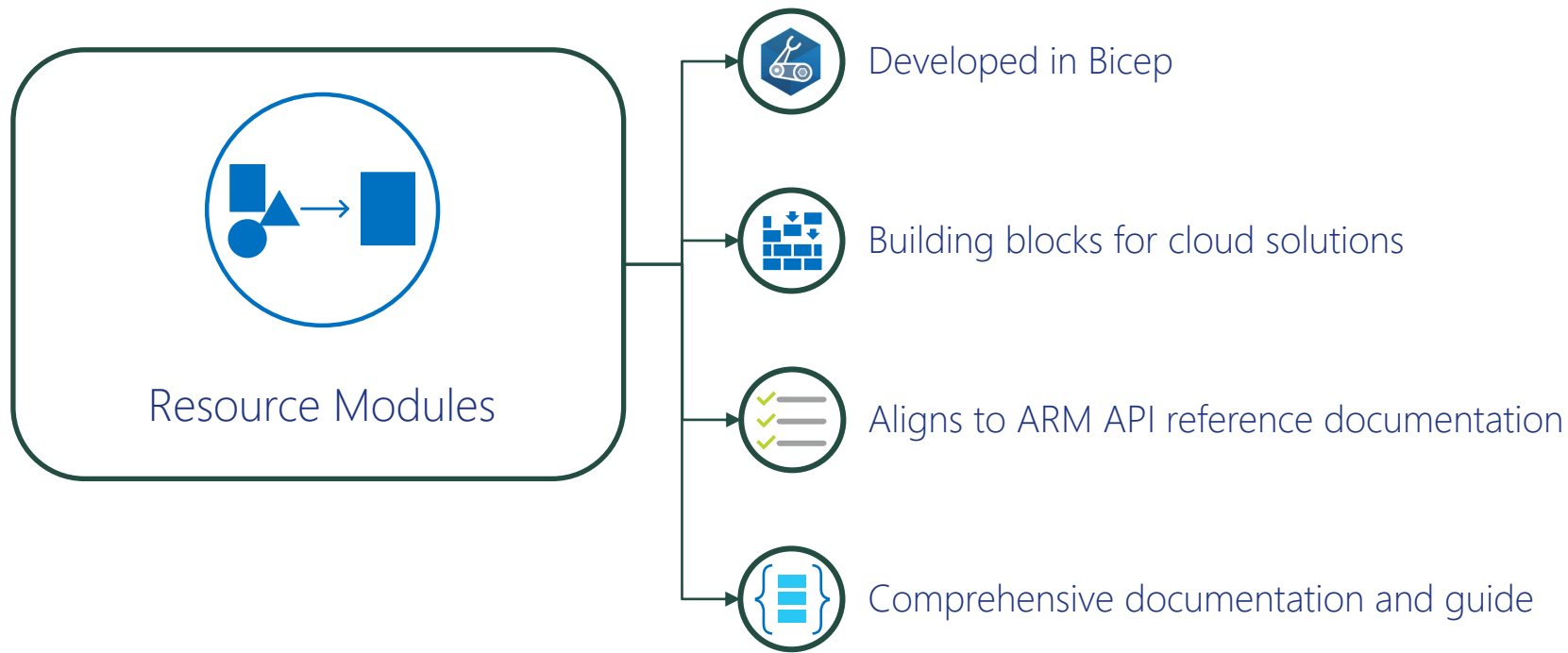


CI Environment

"the factory that validates and makes the building blocks available for users to construct solutions"



CARML in its simplest form





CARML and DevOps toolsets

GitHub Actions / Azure DevOps Pipelines



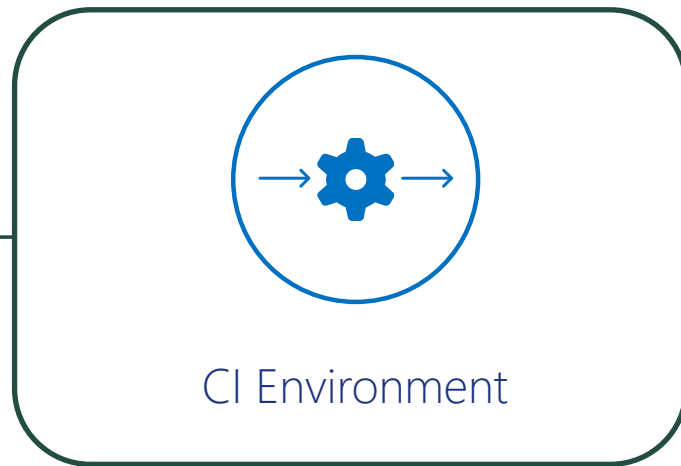
Comprehensive Static validation



Modules validation and deployment tests



Modules publishing and versioning





What does a CARML module look like

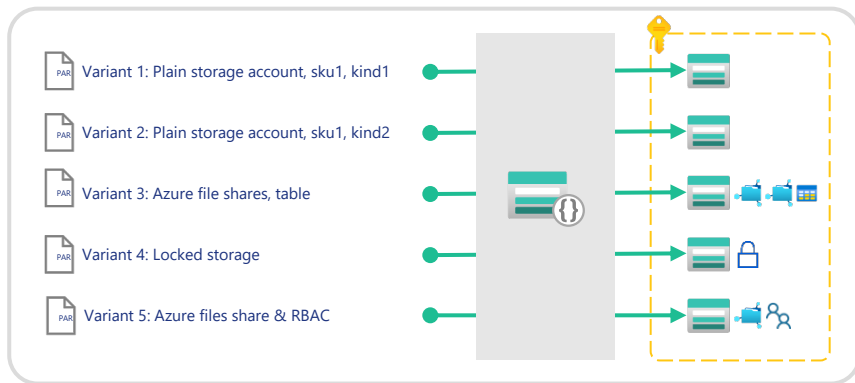
A *reusable building block* for *Infrastructure as Code* deployments

Flexible, generalized, multi-purpose

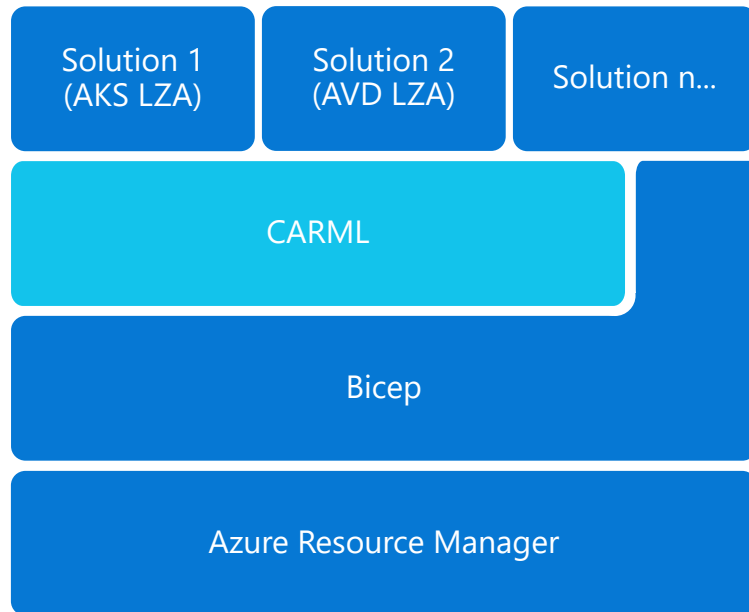
Integrates child resources

Integrates extension resources

Cross references from other modules



Deploys a single or strongly correlated services



How it works

CARML

Consumer of CARML

1

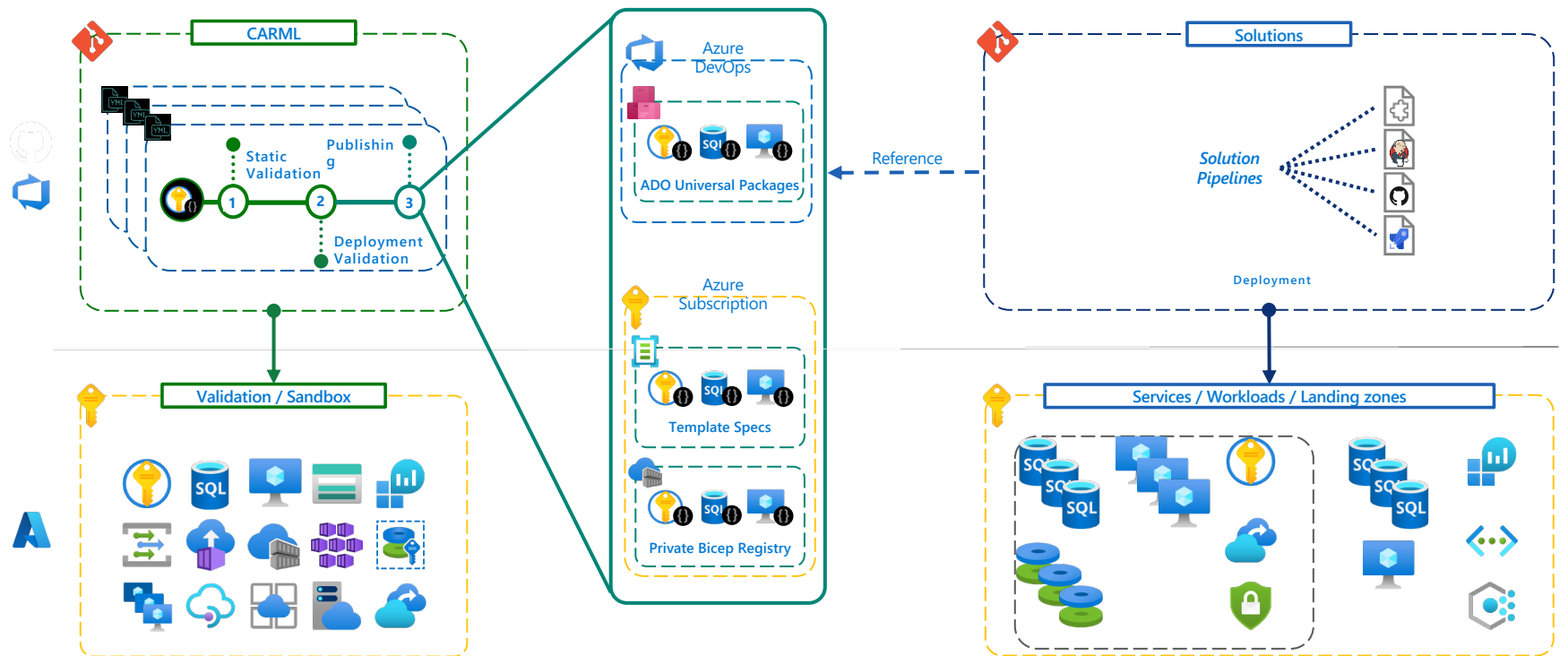
Develop Modules

2

Publish Modules

3

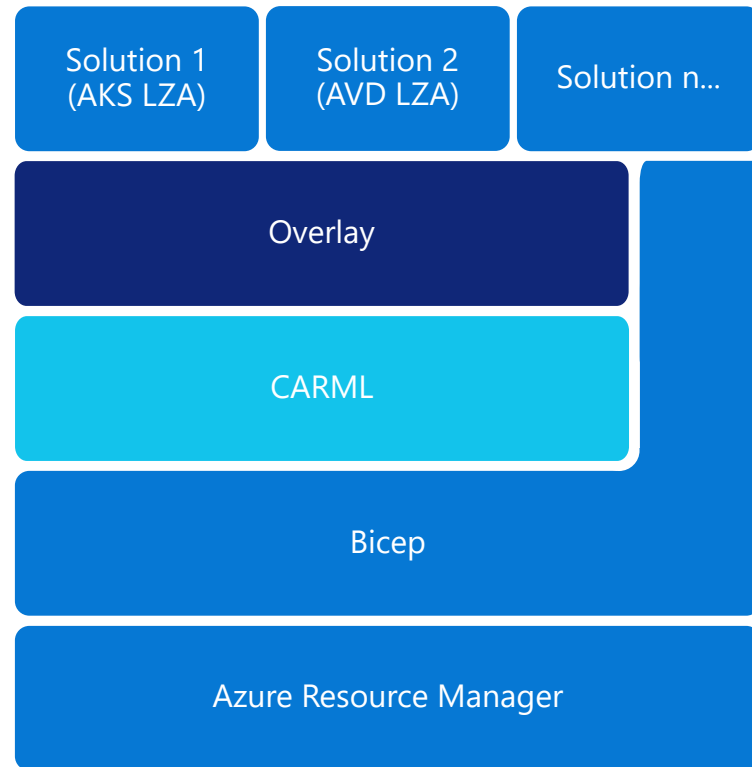
Consume Modules





Overlay Pattern

- Bake in mandatory parameters
- Comply with your security policies and controls
- Thin abstraction, avoids complexity (i.e., solutioning)
- Published for end users to consume
- Helps keep CARML in sync with public changes and minimizes conflicts
- Use Policy to control module version consumption.



Demo



Sponsors





CARML Future

- CARML is evolving into Azure Verified Modules (AVM).
 - Backed by Bicep PG and Customer Success Unit at Microsoft
 - AVM is not just “yet another” IaC project. It’s the One Microsoft approach for IaC, bringing the CSU, ISD, Engineering, etc. together.
 - Will extend to include other IaC tools (Terraform).
- Publishing into the Microsoft Public Bicep Registry.
 - CARML is being tuned for getting the modules into the officially endorsed Microsoft Public Bicep Registry.



References

- CARML: aka.ms/carm1
- Demo: <https://github.com/tyconsulting/ExpertsLiveAU2023-CARML>
- [AzureTar](#) YouTube Series: IaC on Azure with CARML Bicep Modules:
 - [Part 1](#)
 - [Part 2](#)
 - [Demo Repo](#)



Thank you

- Get in touch with the CARML team:
 - [Azure/ResourceModules · Discussions · GitHub](#)



A CARML module structure

✓ key-vault \ vault	
> .bicep	Extension resources (e.g., role assignments)
> .test	Test deployment file(s) for validation
> access-policy	Child resources
> key	Replicated folder structure
> secret	Can be deployed on their own Can be deployed in loops when deployed from the parent
🔗 main.bicep	Core deployment file
{ } main.json	JSON version of the core deployment file
📄 README.md	ReadMe file
{ } version.json	Version file

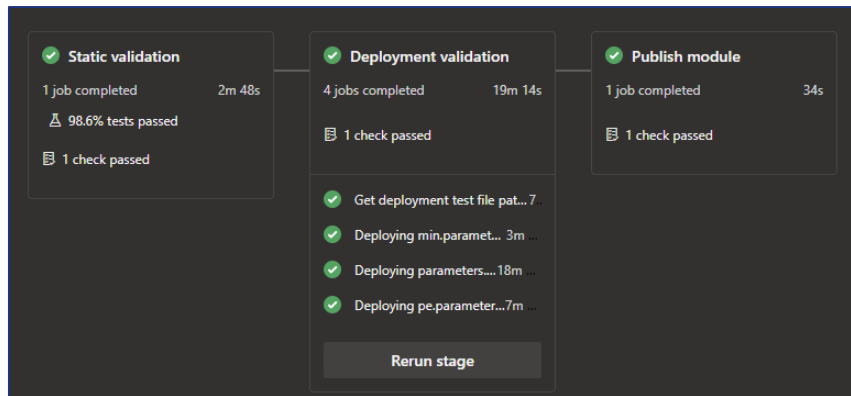


Available flavours for a CI Environment

GitHub Workflows



Azure DevOps Pipelines





Module documentation and versioning

- ReadMe **markdown** files outline what each module contains and how it can be used
- Each module ReadMe **references its child** resource's ReadMe, if applicable
- **ReadMe Generator** utility allows automatic creation/update of the ReadMe from its corresponding core deployment file
- Version file contains **major.minor** module version

```
# Key Vaults [Microsoft.KeyVault/vaults]

This module deploys a key vault and its child resources.

> ## Resource types ...

> ## Parameters ...

> ## Outputs ...

> ## Template references ...

# Key Vault Secret [Microsoft.KeyVault/vaults/secrets]

This module deploys a key vault secret.

> ## Resource Types ...

> ## Parameters ...

> ## Outputs ...

arm > Microsoft.KeyVault > vaults > {} version.json > ...
1  {
2    "$schema": "https://raw.githubusercontent.com/c
3    "version": "0.4"
4  }
```



Contributing to CARML

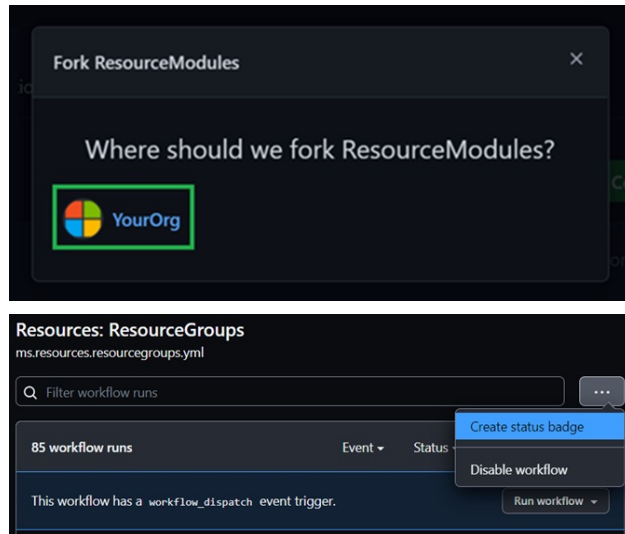


Steps

1. Create / pick an issue at aka.ms/CARML
2. Fork and configure the repository
3. Implement your contribution
 - Depending on the contribution (e.g., [module](#), [pipeline](#)) note the design guidelines in the [CARML wiki](#)
4. Test the code using the pipeline(s) against your environment
5. Open a PR, reference the badge status of your pipeline run and link your issue to it

References

- Read up on the wiki at aka.ms/CARML/wiki





Microsoft Support Policy

CARML is a Microsoft open-source initiative meant to help the community get up to speed with Bicep modules implementation references.

If issues are encountered when deploying these Bicep modules users will be able to engage Microsoft support via their usual channels. Please provide correlation IDs where possible when contacting support to be able to investigate the issue effectively and in a timely fashion.

Following list of issues are within the scope of Microsoft support:

- Underlying Resource or Resource Provider issues when deploying modules (e.g., Management Groups, Policies, Log Analytics Workspace, Virtual WAN, Virtual Network) for any deployment failure.
- Module specific issues (e.g., template errors, internal server errors, etc.).
- Template specific issues (e.g., template publishing, resource removal, etc.).

Any issues that are deemed outside of the above list by Microsoft support and/or requires bugfix in the module or code in the repo, **Microsoft support will redirect user to file the issue on GitHub.**

Project maintainers and community aim to get issues resolved in timely fashion as per community support policy of this repo.