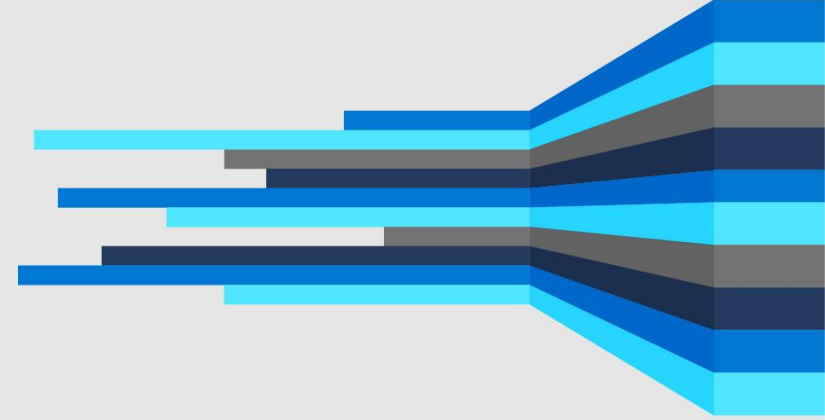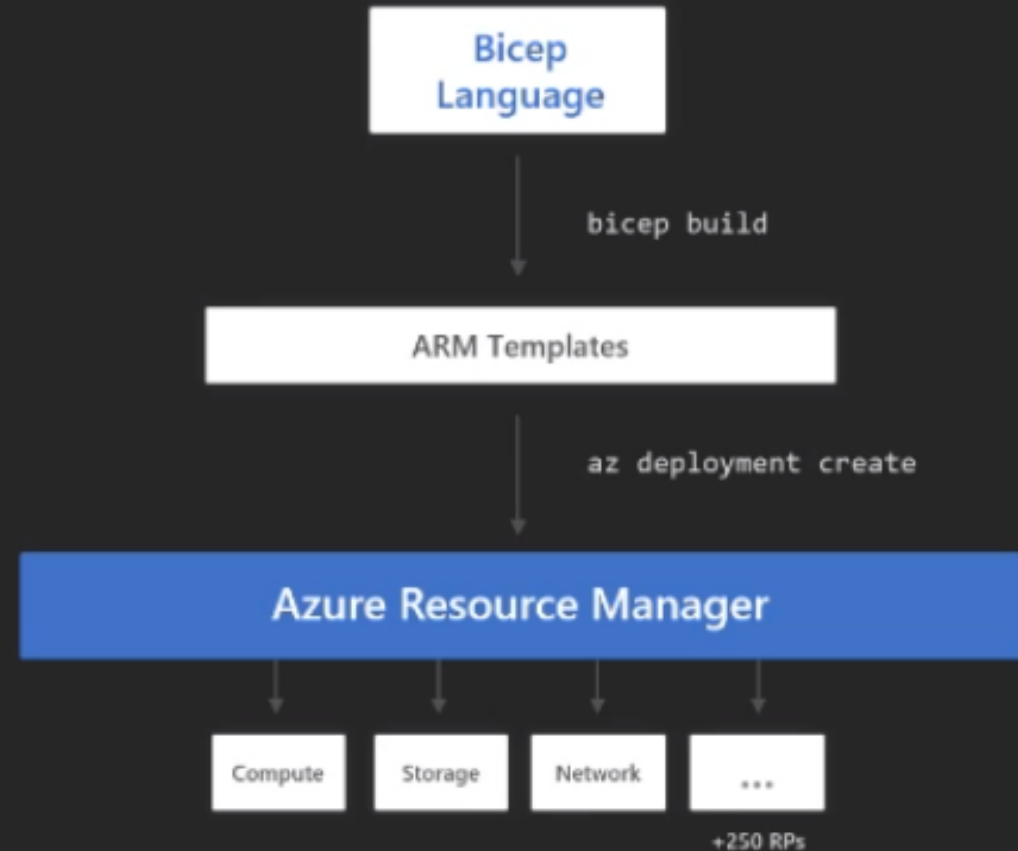# Azure Bicep

Infra as Code in Azure

What is Project Bicep?

Declarative language for
describing and deploying
Azure resources.

Abstraction layer, compiles
into an ARM
Template.

github.com/azure/bicep

Benefits of Azure Bicep

Simple declarative files for creating Azure resources in a reliable, repeatable and auditable way.

Simpler Syntax. No JSON noise

Define Azure resources using a simple language to learn, read and understand.

Declarative Syntax

Declare how the resources should be and Azure Resource Manager "makes it so".

Modularity

Modules to break your bicep code into manageable chunks. Public and Private module registries

Built for tool-ability

First class authoring experience in VSCode. Rich type-safety, intellisense and validation.

Convert existing templates

Easy conversion of ARM Templates to Bicep

https://aka.ms/BicepDocs

Bicep vs other tools

Day zero support for all resource types & API versions
Bicep templates are generated based on resource schema. All preview and GA services can be created using Bicep from Day Zero.

No state files
No state management. Can query the state directly from Azure.

Deep integration with Azure Governance Services
Azure Policy remediation for non-compliant resources are done via Bicep. TemplateSpecs can be a source for modules. Bicep Landing Zones and Blueprints

100% free and open source
No paying for 'premium' capabilities and 100% supported by Microsoft support

Automate Portal actions
Use the Azure Portal as Bicep template generator. Export a deployable Bicep file for any portal deployment actions

Demo
Bicep Playground

https://aka.ms/bicepdemo

Demo

Setup your dev environment

https://github.com/Azure/bicep/blob/main/docs/installing.md

Workflow
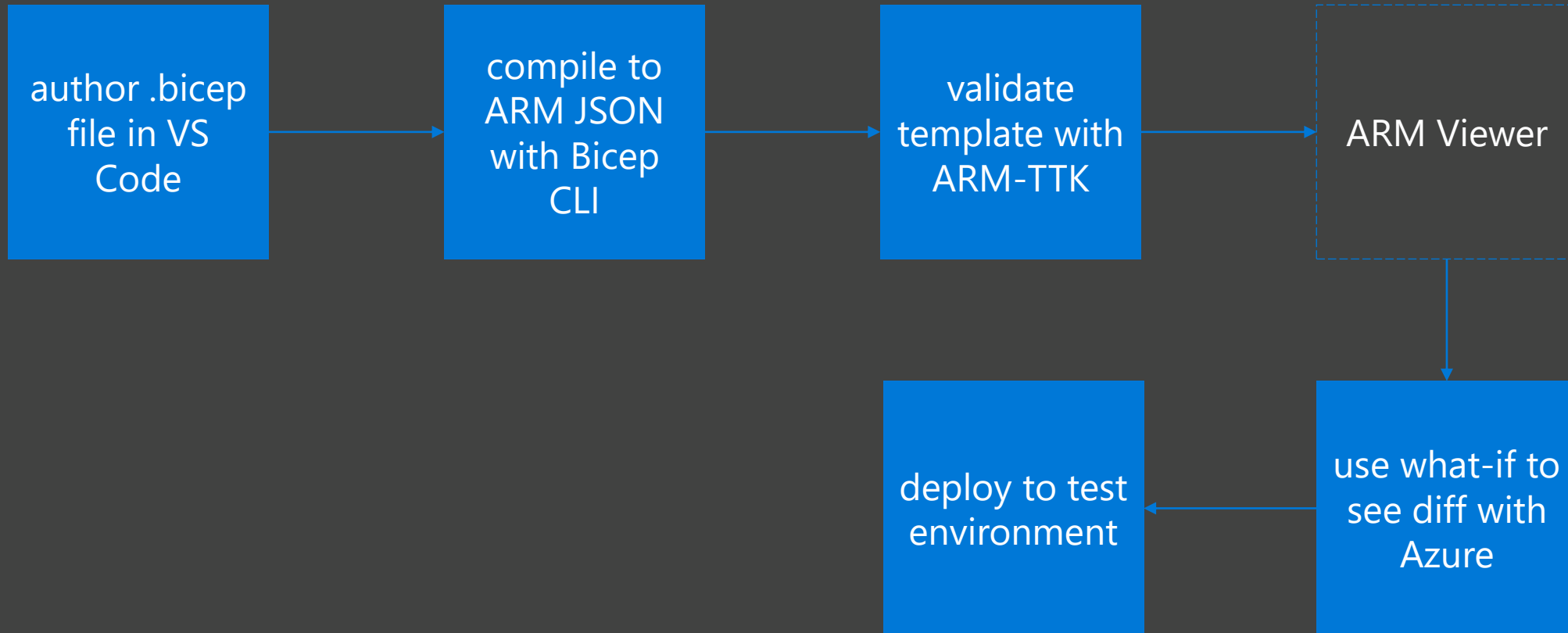
Roadmap

Azure Deployments
ARM + Bicep

1 **Module Registry**
Running infra with verified templates and community modules. Private modules to share across your organization.

2 **Extensibility**
Configure data plane operations and non-Azure services (K8s, GitHub, AAD, ServiceNow)

3 **Bicep Testing Framework & Linter**
Find and fix misconfigurations earlier to prevent risky deployments

4 **Continued Language Enhancements**
includeFile () support, etc.

5 **Stacks**
Improve deployment lifecycle management (rollback, delete, RG expiry)

6 **TemplateSpecs: Staging and Sharing of Templates**
ARM Templates that can package, reused, and shared as a catalog [Public Preview, GA in May '21]

7 **Compliance Blueprints**
One-click deployment solutions to help meet regulations (12 published today ex: ISO, FedRamp, NIST etc.)

8 **Safe deployment guardrails**
Managing change automation so that all code and configuration updates go through well-defined stages and test gates

Public roadmap: github.com/Azure/bicep/projects

All investments and dates are subject to change

Try Azure Bicep today!

github.com/azure/bicep

Follow @biceplang on twitter for updates