



**Njoo Tuinue viwango!**

July 24-25, 2025 | Nairobi, Kenya

**EAST AFRICA**







**EAST AFRICA**

**24<sup>TH</sup> JULY 2025**

# **DEPLOYMENT STACKS WITH AZURE BICEP**





Who is Samson !? ?

A passionate technologist, mentor and lover of all things cloud computing.



- ❖ **Organizer @Azure Kenya User Group**, a forum that brings together professionals, customers, and enthusiasts in the Azure ecosystem to learn through sharing experiences
- ❖ **Technical Cloud Consultant @BUI Technologies**, designing & implementing cloud solutions for clients around the globe.
- ❖ **Co-founder @Resource Cloud Academy**, Empowering students & technology enthusiasts to differentiate themselves by learning cloud technology.
- ❖ **Microsoft MVP**, Infrastructure as Code

How can you benefit from me?

If your business is looking for better ways to deal with servers, or you want to upskill in cloud technology, or just want to be part of a vibrant cloud community, let's connect 👉 👈

Let's Connect 🍷 🍷 🍷

Phone: 0705917051

Linkedin: Samson Barasa

*"In a world where we can be anything, why should we be average?"*



# Agenda

1. Introduction to Deployment Stacks
2. Why Deployment Stacks?
3. Modes and Configuration
4. Demo Walkthrough
5. Advanced Features
6. CI/CD Integration
7. Getting Started
8. Q&A





# Deployment Stacks

laC





# Why Infrastructure as Code?

- Consistency across environments
- Version control and auditability
- Repeatability and automation
- Easier collaboration and change management







# What is Deployment Stacks

- Deployment Stacks are a new **Azure resource type** (**Microsoft.Resource s/deploymentStacks**) that allow you to manage a group of Azure resources as a single unit.

## How did we get here?

- Azure Blueprints deprecated after 7 years in preview
- Deployment Stacks introduced as modern IaC alternative
- Now GA with support for full resource lifecycle control
- Inspired by Terraform-like state management





# Why Use Deployment Stacks?

- Unified control over related resources
- Auto-delete unmanaged resources
- DenySettings for governance
- Supports subscription & RG-level scopes







# Modes: ActionOnUnmanage

## detachAll

- Unmanaged resources are left intact, but no longer associated with the Deployment Stack.

## deleteResources

- Removes any resource that was part of the stack but is no longer in the template.

## deleteAll

- Deletes **all resources** in the stack and the stack itself when undeployed or updated with empty template.



# ActiononUnmanage

Mode	Ideal Use Case	Risk Level	Resource Retention
detachAll	Test, staging, safety-first updates	<div><div></div>Low</div>	<div><div></div>Keeps resources</div>
deleteResources	CI/CD, prod IaC pipelines	<div><div></div>Moderate</div>	<div><div></div>Deletes removed</div>
deleteAll	Full environment teardown	<div><div></div>High</div>	<div><div></div>Deletes all</div>







# CLI Example: Create a Stack

- az stack group create \
- --name mystack \
- --resource-group myrg \
- --template-file main.bicep \
- --parameters environment=dev \
- --action-on-unmanage deleteResources





# Demo: Stack Lifecycle Walkthrough

1. Create a Stack from Bicep
2. Update Resources
3. Auto-delete Removed Resources
4. Apply DenySettings
5. Tear Down the Stack







# Advanced: Deny Settings

- denyDelete: block deletion of resources
- denyWriteAndDelete: full lock
- Exclude principals (e.g., DevOps SPN)





# CI/CD Integration

- Predictable, repeatable deployments
- Audit-ready infrastructure changes
- Quick rollback via Git history
- Aligns dev, ops & security teams

## How did we get here?

- **Declarative IaC:** Uses Bicep templates tracked in Git
- **Version Control:** All infrastructure changes go through Git commits & PRs
- **Automated Deployments:** Integrates with Azure DevOps / GitHub Actions
- **Rollback Friendly:** Redeploy previous versions via Git revert







# Conclusion

## Deployment Stacks give you:

- The control of Terraform
- The integration of Bicep
- The governance Azure has long needed

**They're ready for production and built for teams that care about structure, safety, and scale.**



[Introduction to Deployment Stacks](#)

[Deployment Stacks – Deep Dive by John Savill](#)

[Running with Deployment Stacks by Bernie White](#)

[Deployment Stacks by Sam Cogan](#)

# Thank You!

