Summit Talk
Automating Network Firewall Rule Creation Using Powershell and CI/CD
Nills Franssens

Runner

Belgian in the US

Cloud Solution Architect @ Microsoft

- Infrastructure
- Networking
- Open Source
Setting the context: Cloud Security

- DDoS protection
- Identity management
- Network security
- Application security
- Encryption
All major cloud providers offer a native L3/L4 firewall capability (Security Group)

Common elements

- Like ACLs
- Instance tagging – security groups
- Stateful
- An object

Differences

- AWS has 2 implementations: security groups and network ACL
- Azure allows IP-based and security group based rules mixed
- Assigned to different levels: AWS is instance level, Azure is instance or subnet level, GCP is VPC level
Customer use case driving this solution.

“Can you help me implement these 500 new network rules TODAY?”
- Anonymous customer
This called for an automated solution.

- Human error.
- What if we needed to do it all over again?
- What if this needs to repeated for a second region?
- We only had half a day to do this right.
We needed to decide whether to use imperative or declarative approaches. Highly parallel, single API call, and source code were the options.
Automation in step 1 in a DevOps Lifecycle.

“DevOps is the union of **people**, **process**, and **products** to enable continuous delivery of value to your end users.”
End-to-end solution overview

Users make changes to ruleset.

CSV is checked in → CI Trigger → Script to transform CSV to ARM → CD Trigger → Artifact stored → Security team needs to approve changes → Azure Deployment

Git check-in → Build process → Release process
Script walkthrough

1. Read CSV input unsorted
2. Creates per-subnet rules hashtable
3. Generates per-subnet IaC artifact
4. Validates IaC artifact
Demo
Outcomes / Benefits

- Implemented 500 firewall rules in half a day
- Flexible way to adapt firewall rules afterwards
- Security review built-in to process
- Standard format for developers to request firewall changes
- Happy customer