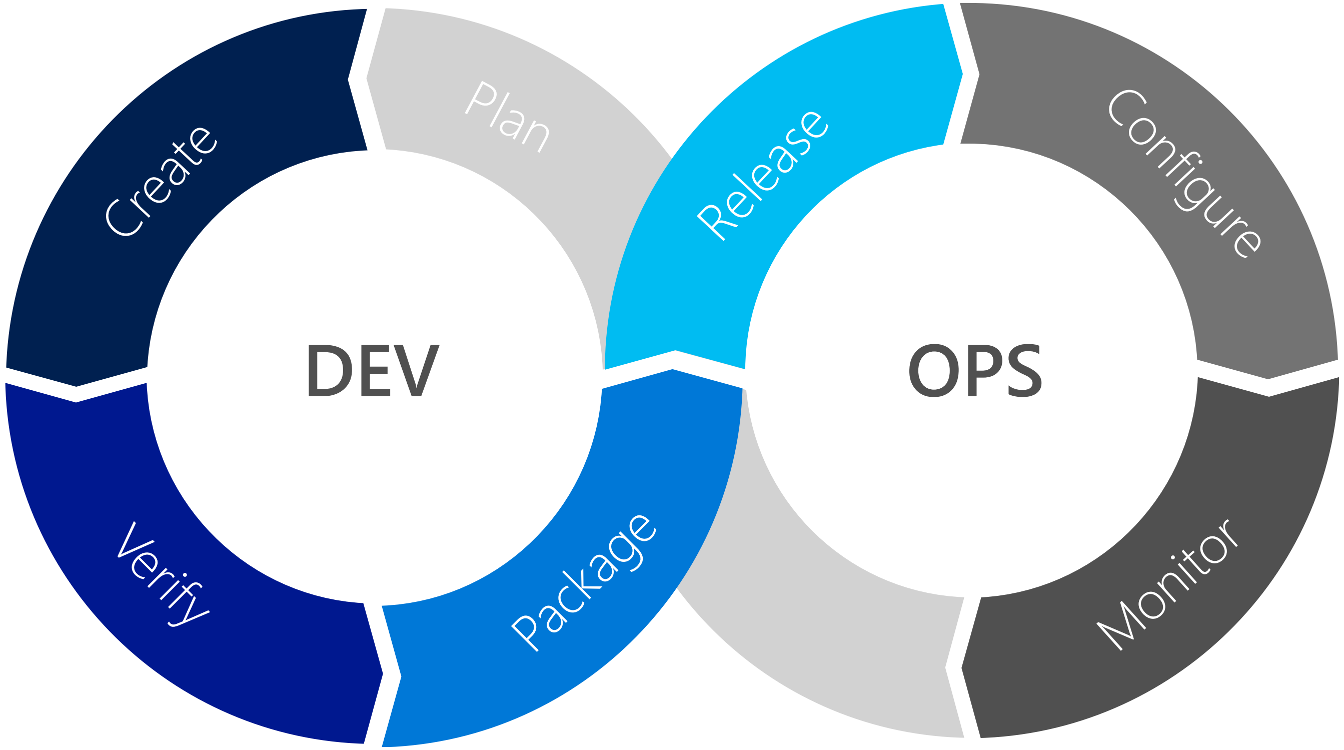
# DevOps: Return of the Security

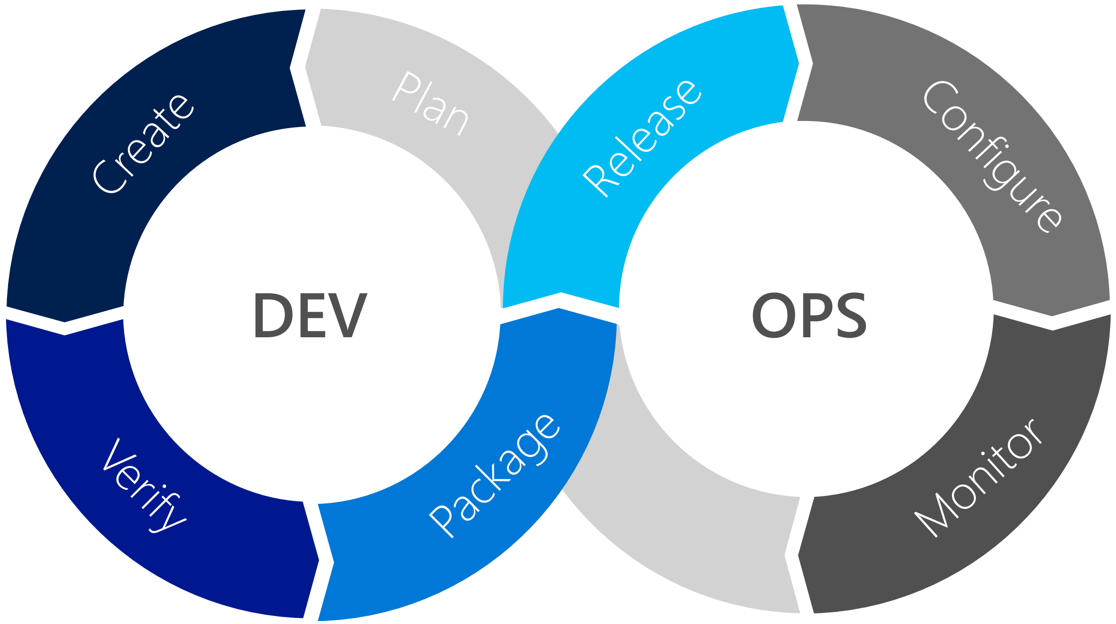
# Speaker

- Mustafa Toroman
- Solution Architect @ Authority Partners
- @toromust
- Microsoft Azure MVP
- MCSE, MCP, MCSA, MCITP, MCSD, MCT, MS v-TSP

# DevOps is here…



# DevOps is here… and driving business value

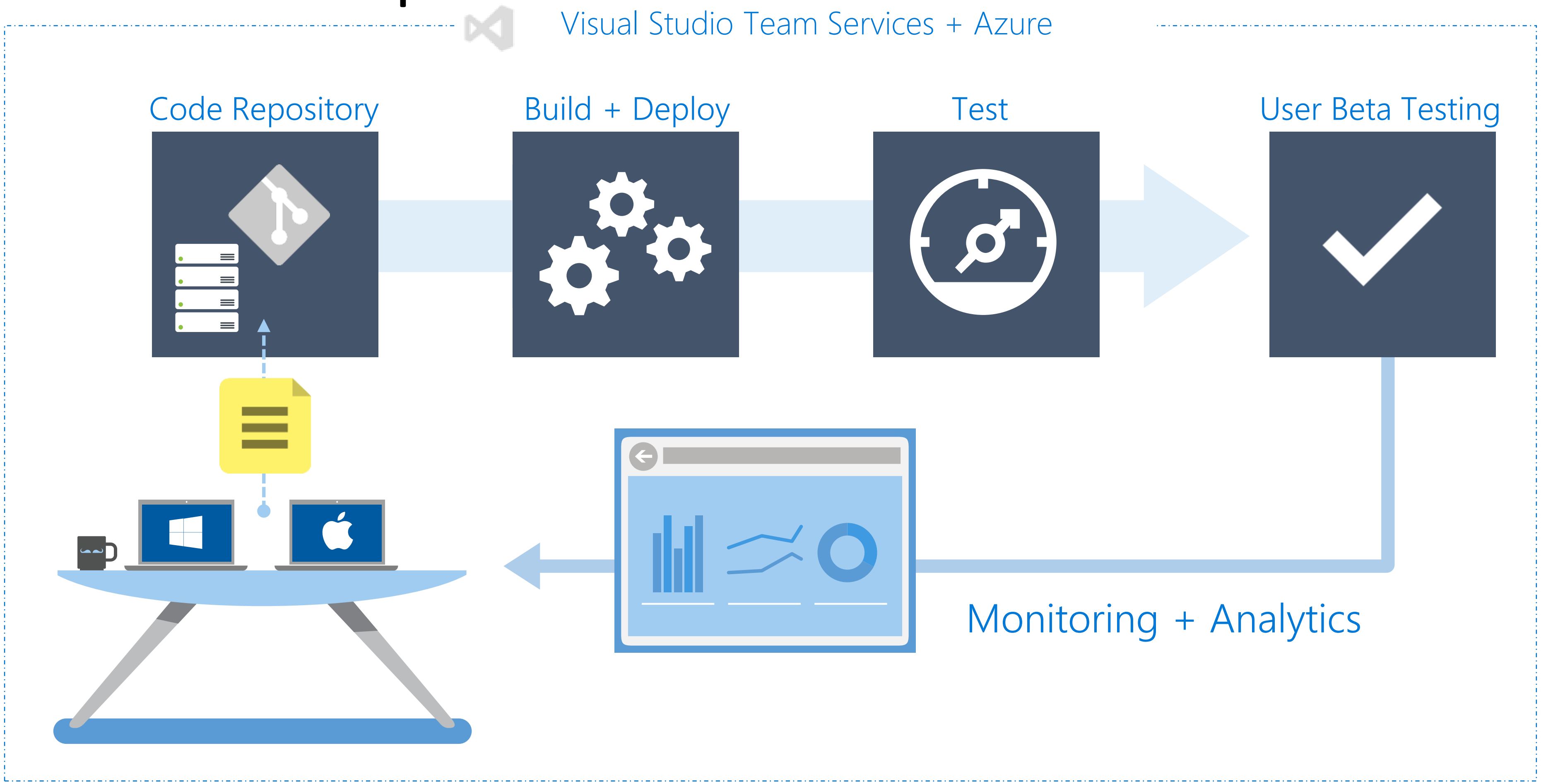


Organizations realizing up to 20% improvements in..

- Time-to-market
- Software quality
- Frequency of deployment
- Increased revenue
- Customer reach and retention

*Source: Global Study by CA Technologies*

# Complete DevOps Solution



Visual Studio Team Services + Azure

Code Repository

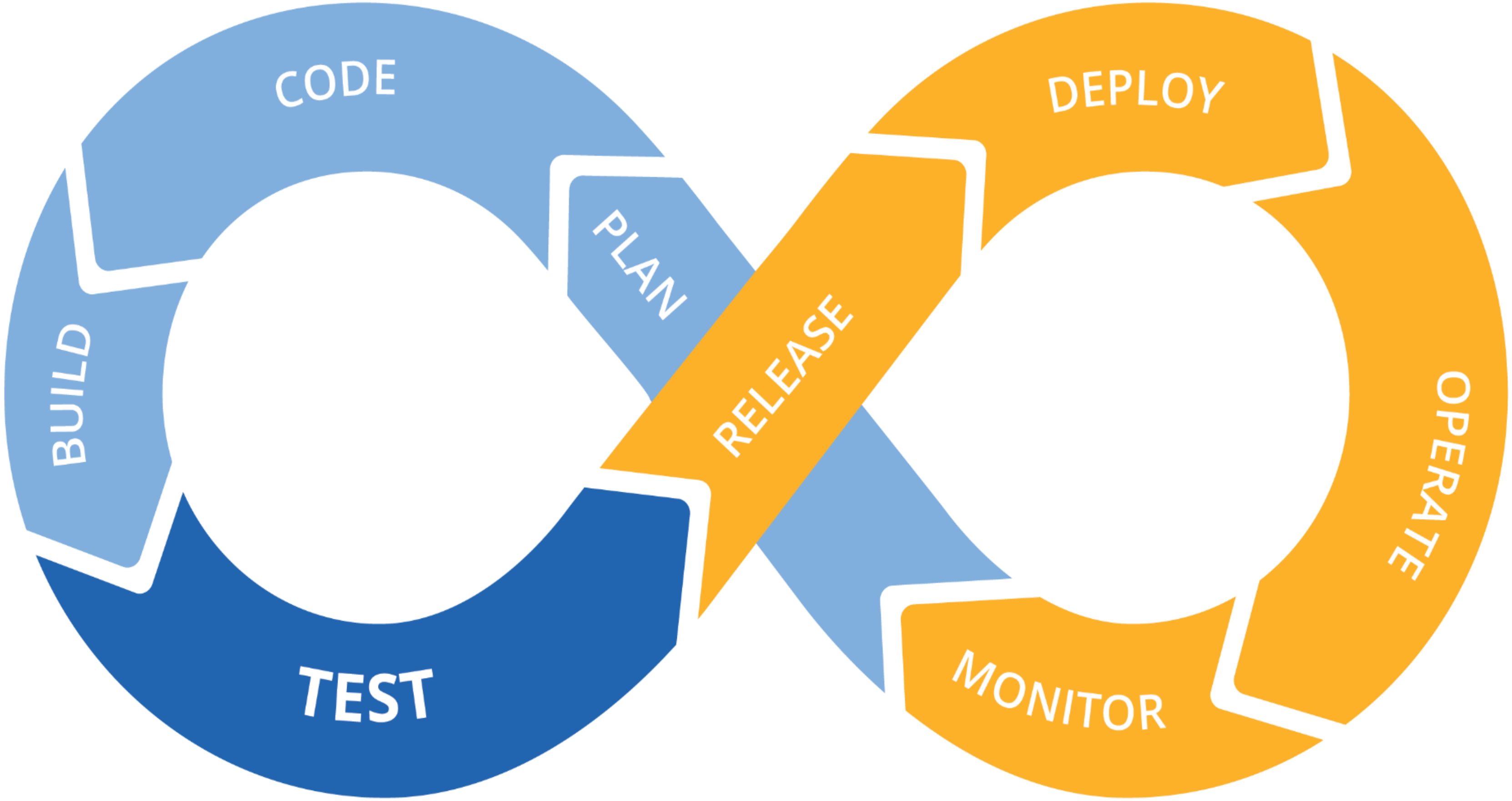
Build + Deploy

Test

User Beta Testing

Monitoring + Analytics

# DevOps?



~~DevSecOps or SecDevOps~~

**It‘s just DevOps!!!**

# What are we talking about?

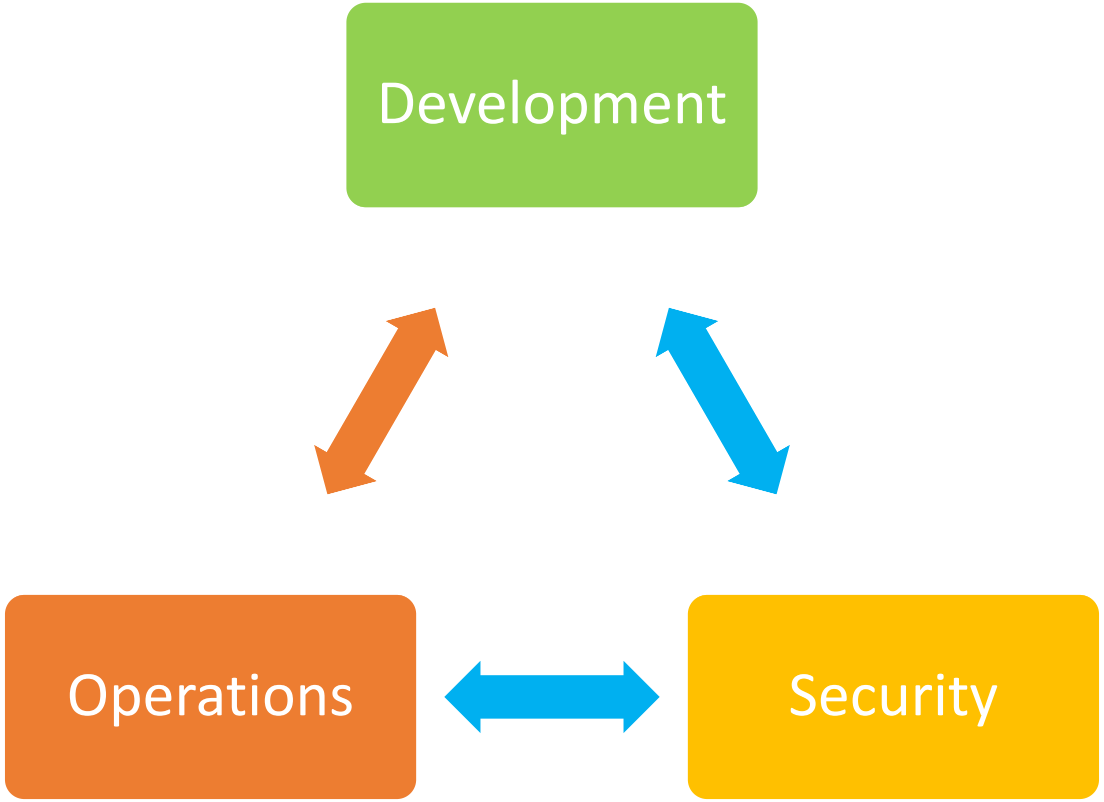
Clear and Concise Communication

Infrastructure as Code->Security As Code

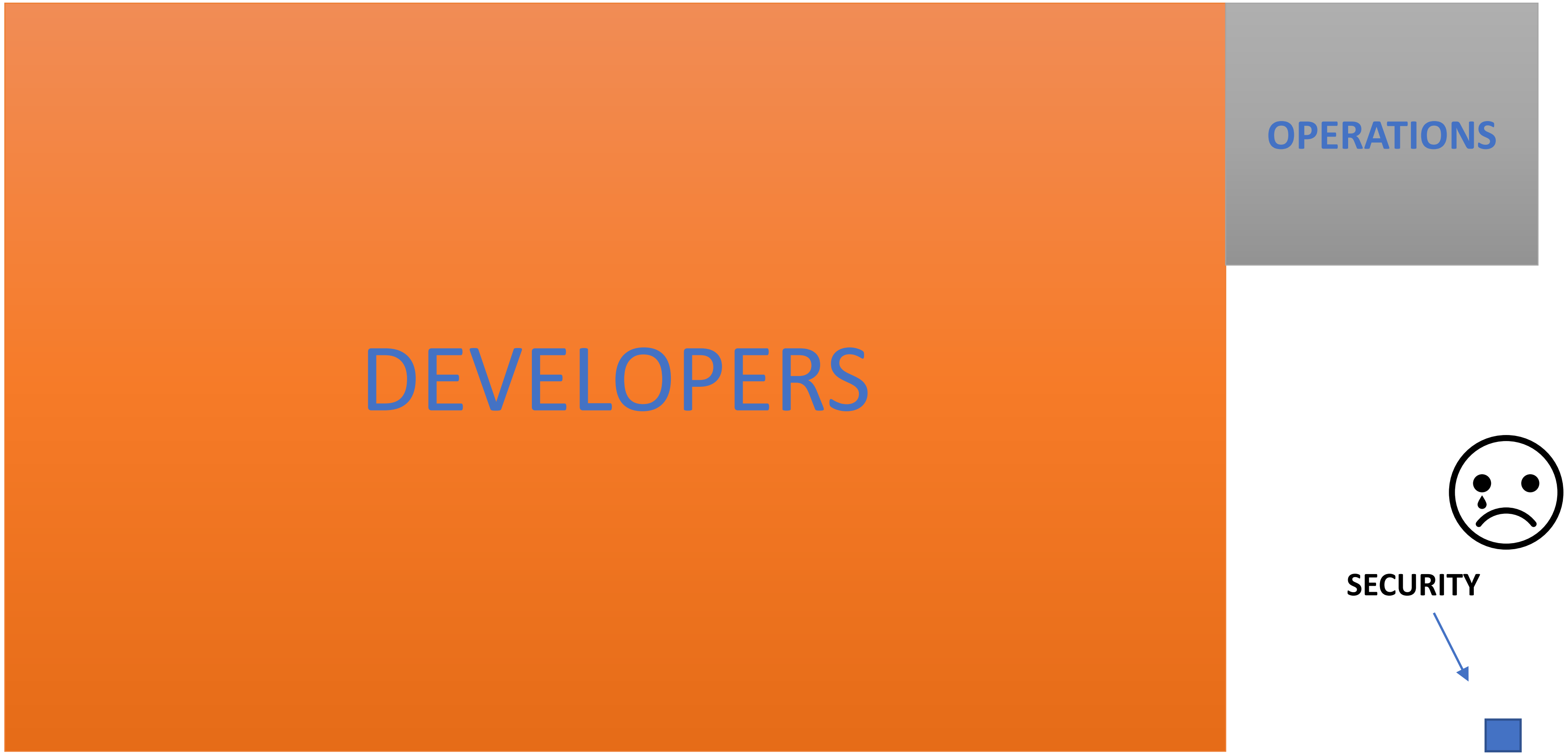
Mandatory Training

Security integration with(in) DevOps cycle

# Clear, Concise Communication



# Some numbers



DEVELOPERS

OPERATIONS

SECURITY

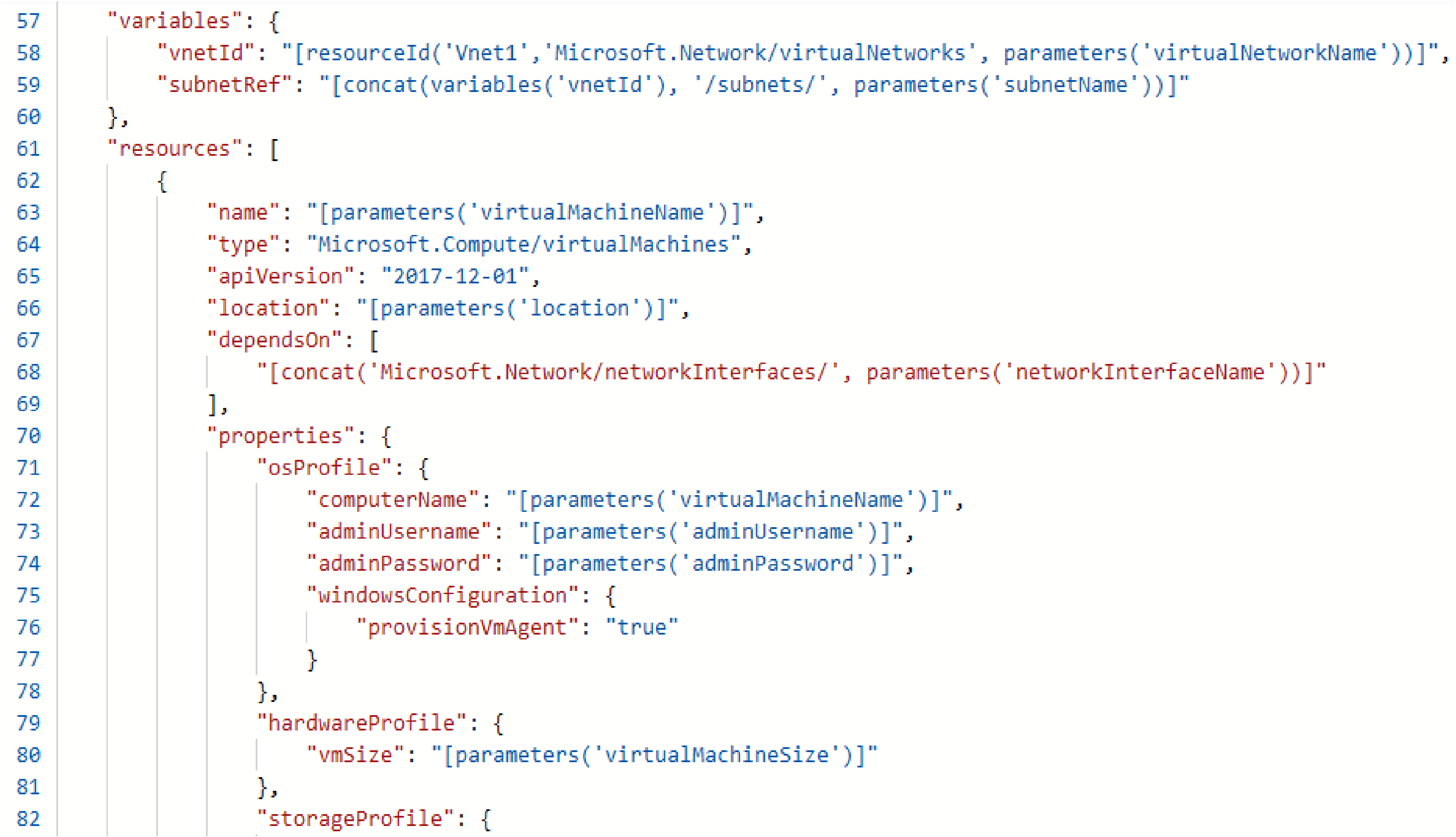
# Effective Communication

- Big NO to:
    - Excel lists and Word (macro) templates
    - (Malicious) attachments
- Big YES to:
    - Boards
    - Git/Versioning

# Security as a Code

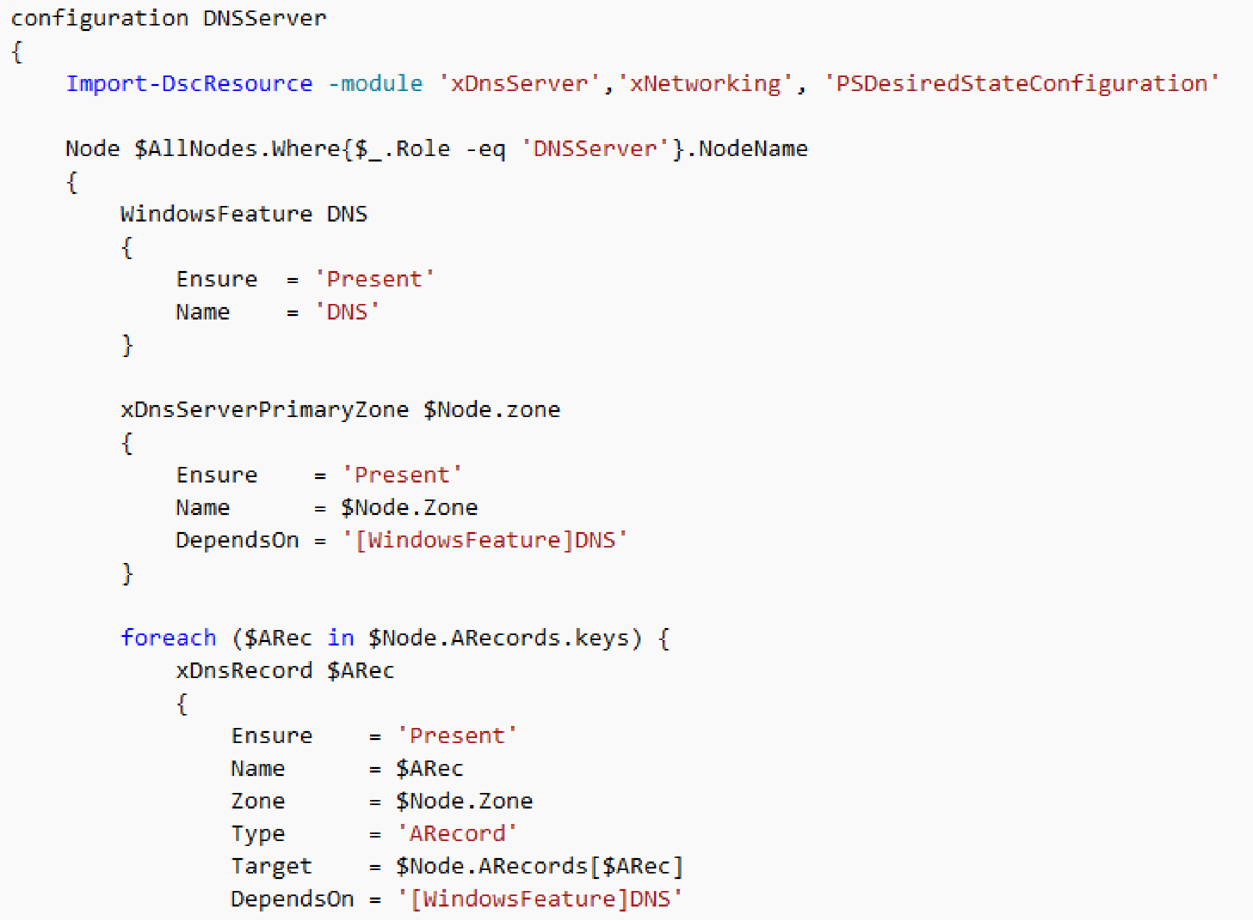
- Application Source Code
- Azure ARM and AWS Cloud Formation
- Server Configuration – Chef, Puppet, DSC
- Secrets Management

# Arm Templates



```
    "variables": {
        "vnetId": "[resourceId('Vnet1','Microsoft.Network/virtualNetworks', parameters('virtualNetworkName'))]",
        "subnetRef": "[concat(variables('vnetId'), '/subnets/', parameters('subnetName'))]"
    },
    "resources": [
        {
            "name": "[parameters('virtualMachineName')]",
            "type": "Microsoft.Compute/virtualMachines",
            "apiVersion": "2017-12-01",
            "location": "[parameters('location')]",
            "dependsOn": [
                "[concat('Microsoft.Network/networkInterfaces/', parameters('networkInterfaceName'))]"
            ],
            "properties": {
                "osProfile": {
                    "computerName": "[parameters('virtualMachineName')]",
                    "adminUsername": "[parameters('adminUsername')]",
                    "adminPassword": "[parameters('adminPassword')]",
                    "windowsConfiguration": {
                        "provisionVmAgent": "true"
                    }
                },
                "hardwareProfile": {
                    "vmSize": "[parameters('virtualMachineSize')]"
                },
                "storageProfile": {
```

# PowerShell DSC

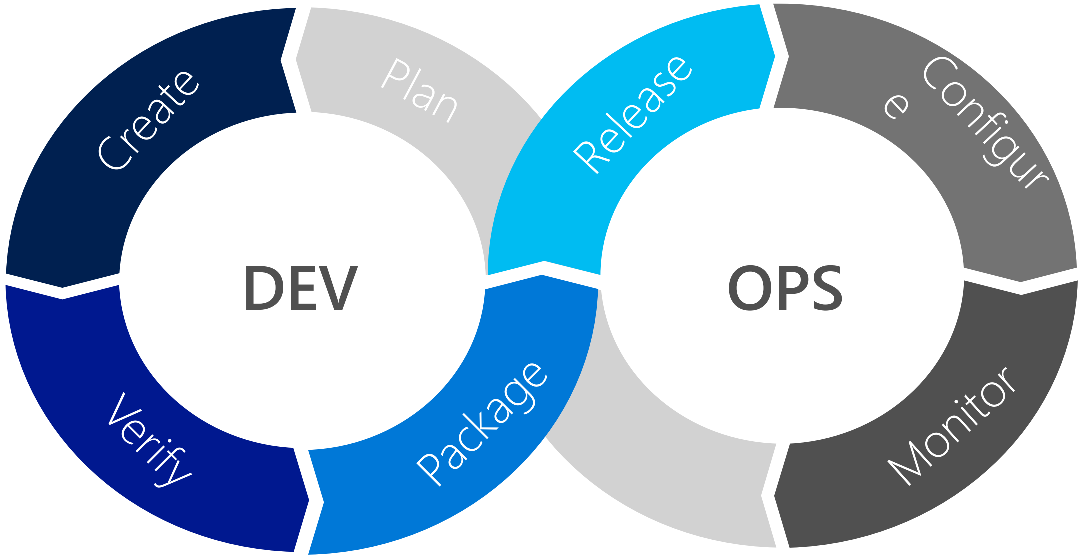


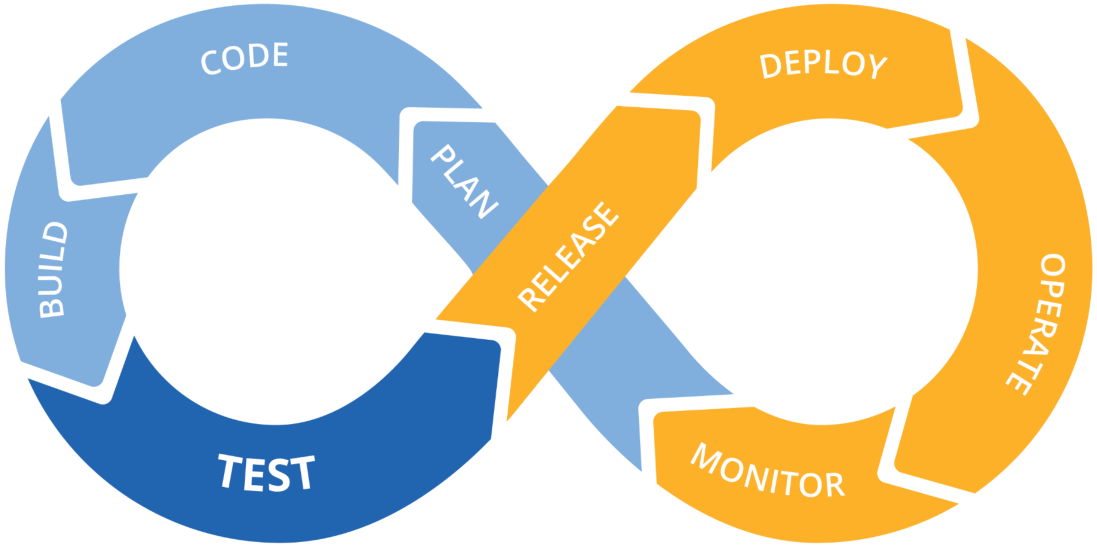
```
configuration DNSServer
{
    Import-DscResource -module 'xDnsServer','xNetworking', 'PSDesiredStateConfiguration'

    Node $AllNodes.Where{$_.Role -eq 'DNSServer'}.NodeName
    {
        WindowsFeature DNS
        {
            Ensure  = 'Present'
            Name    = 'DNS'
        }

        xDnsServerPrimaryZone $Node.zone
        {
            Ensure    = 'Present'
            Name      = $Node.Zone
            DependsOn = '[WindowsFeature]DNS'
        }

        foreach ($ARec in $Node.ARecords.keys) {
            xDnsRecord $ARec
            {
                Ensure    = 'Present'
                Name      = $ARec
                Zone      = $Node.Zone
                Type      = 'ARecord'
                Target    = $Node.ARecords[$ARec]
                DependsOn = '[WindowsFeature]DNS'
```

# Mandatory Training

- Awareness begins at day Zero
- Cross-disciplined training
- We can’t know everything – but can be taught at least basics

# Back to the Security

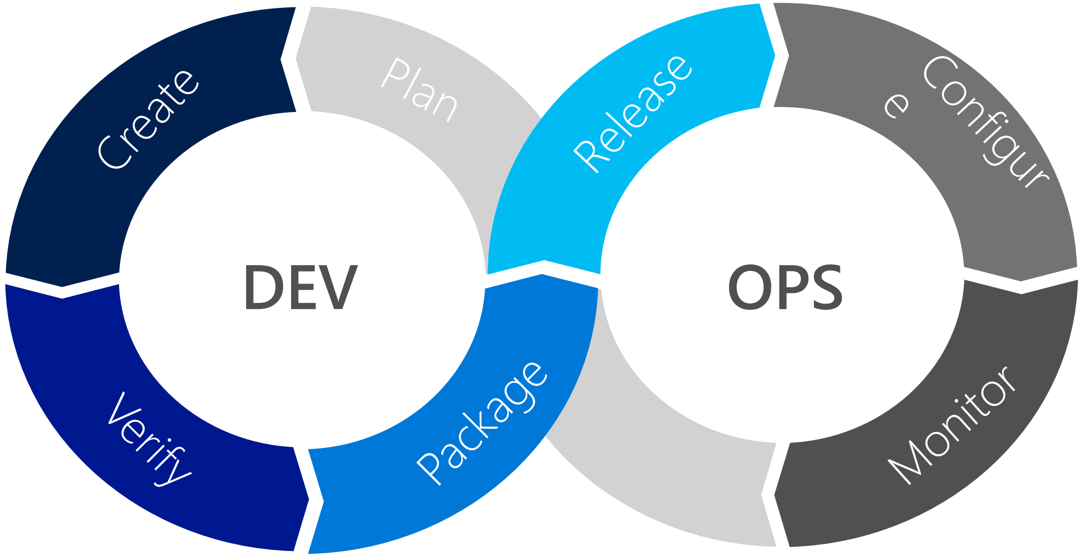


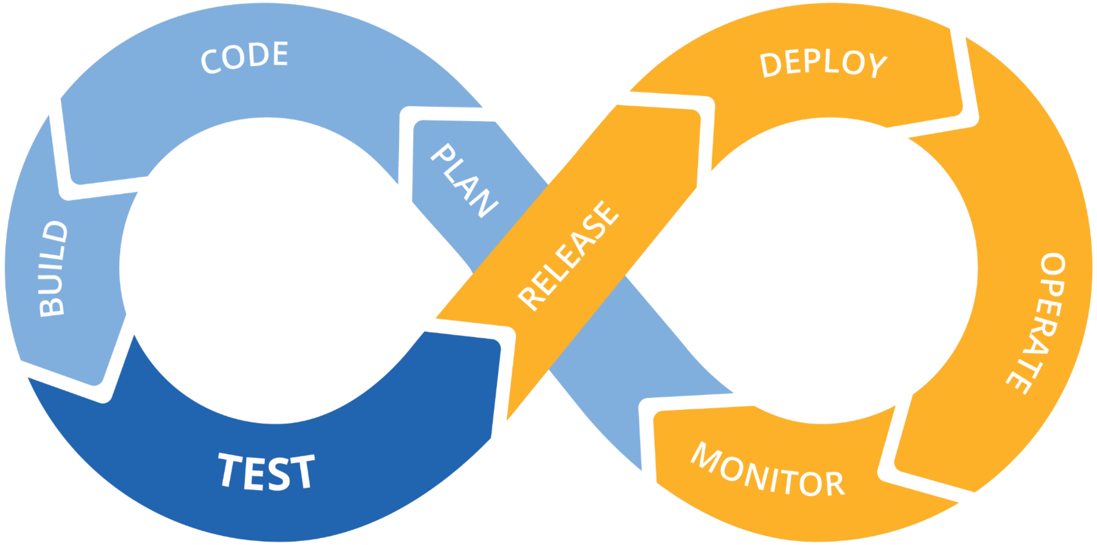


# Alerting and monitoring

- Summary views of critical tasks
- Assurance scans
- RBAC activity
- Security metrics trends and activity
- Runbooks for auto-healing

# Integrate Security to DevOps cycle





# Plan

- Integrate security into sprint planning and reviews
- Consider security stories early

# Code

- Training!
- Test driven development
- Use of the correct tools
- Pull Requests

# Build

- Static code analysis
- Dynamic code analysis
- Dependencies check
- Package management

# Test

- Develop security test cases
- Fuzzing
- Load testing
- DAST or IAST

# Release & Deploy

- Post-deployment security scan
- Post-deployment checkout

# Operate & Monitor

- Monitor logs
- Rescan for vulnerabilities
- Track dependencies
- Patching
- Pen Testing

# Other things to keep in mind…

- Access management
- Use JIT/JEA when possible
- Data security
- Network security scanning
- Physical security

# Q&A?

Thank You!