



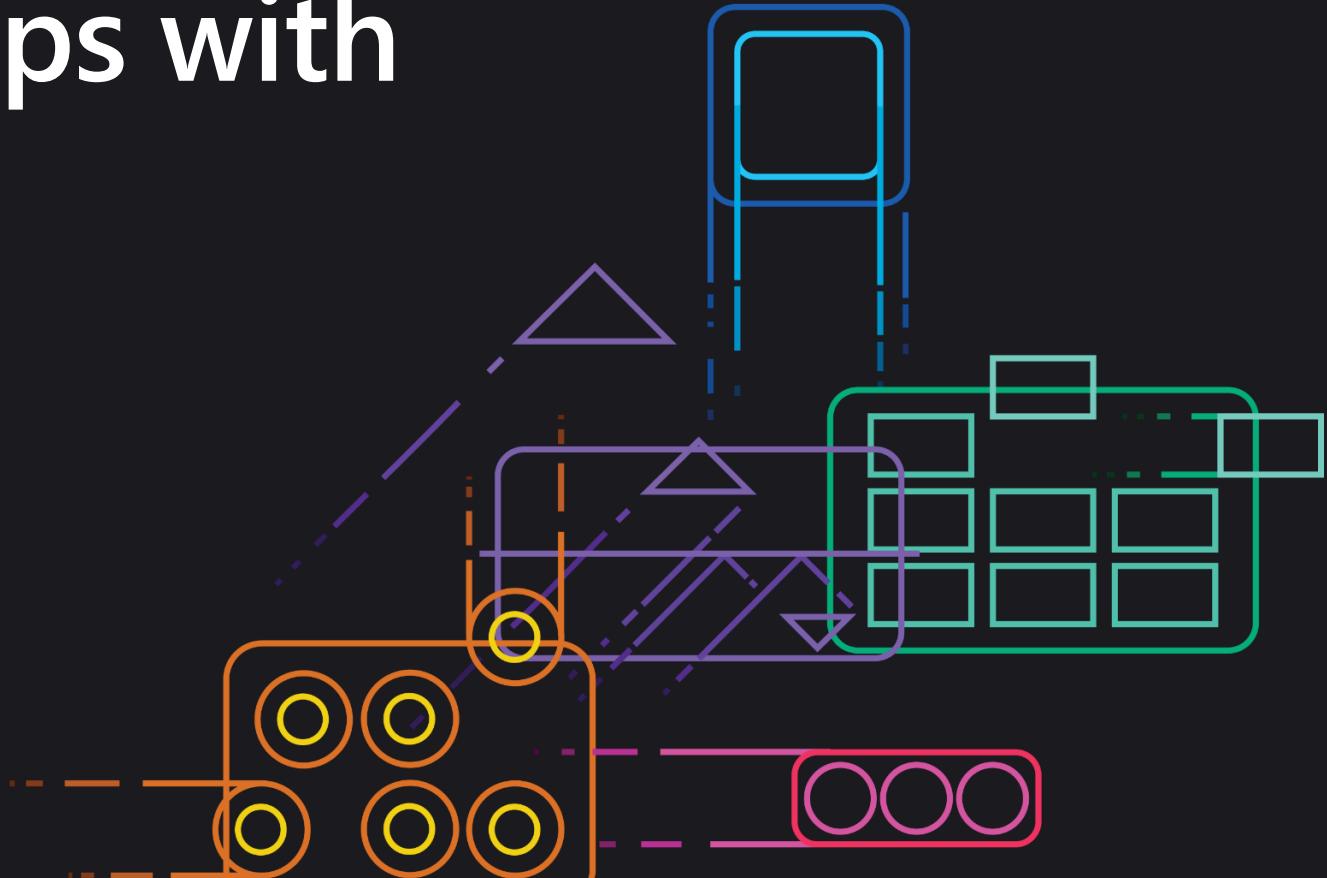
# Accelerating Devops with Azure



Sajeetharan S

**Cloud Solution Architect  
(APAC,OCP)**

*"Azure be the cloud you love"*



# Welcome!

## HOW DO I ASK A QUESTION?

- › If you have a technical or content-related question, please use the Q&A window
- › We will address the questions as they come in

## CAN I VIEW THIS PRESENTATION AFTER THE WEBINAR?

- › There will not be a recording of the session, slides will be shared in the GitHub repository
- › Due to the PII we are not keeping the recording of the session

# Agenda

## Today's Discussion

- What is DevOps?
- Why Devops is important?
- Continuous Integration & Continuous Delivery
- Building Right pipelines
- Github & Azure Devops Better together
- Engineer's guide to a good night sleep
- Demo (Azure Devops, AppCenter)
- QnA and Kahoot fun



# Huh? Devops ? Let me share a story



# What is DevOps?

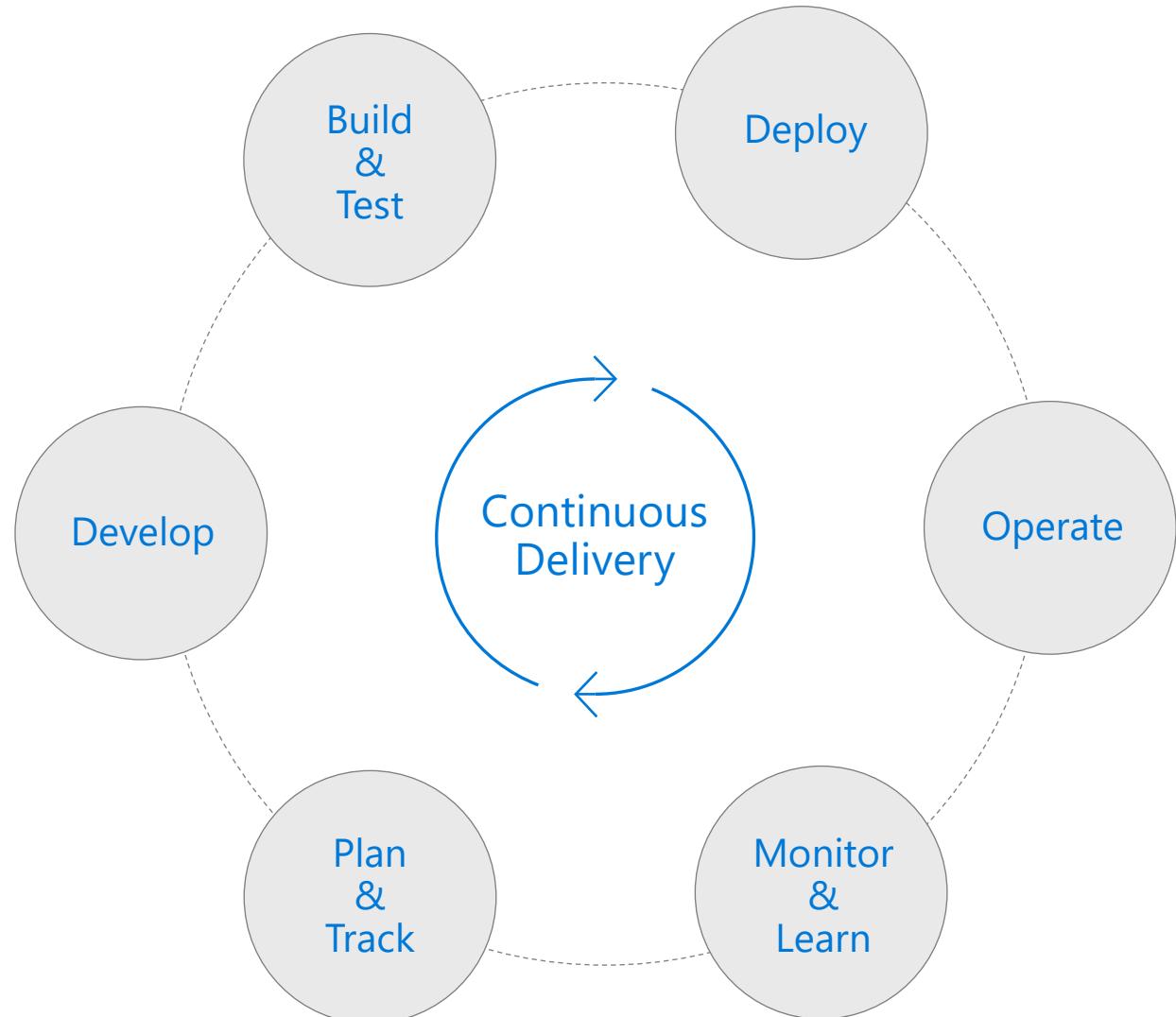
People. Process. Products.

“

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

”

- Donovan Brown



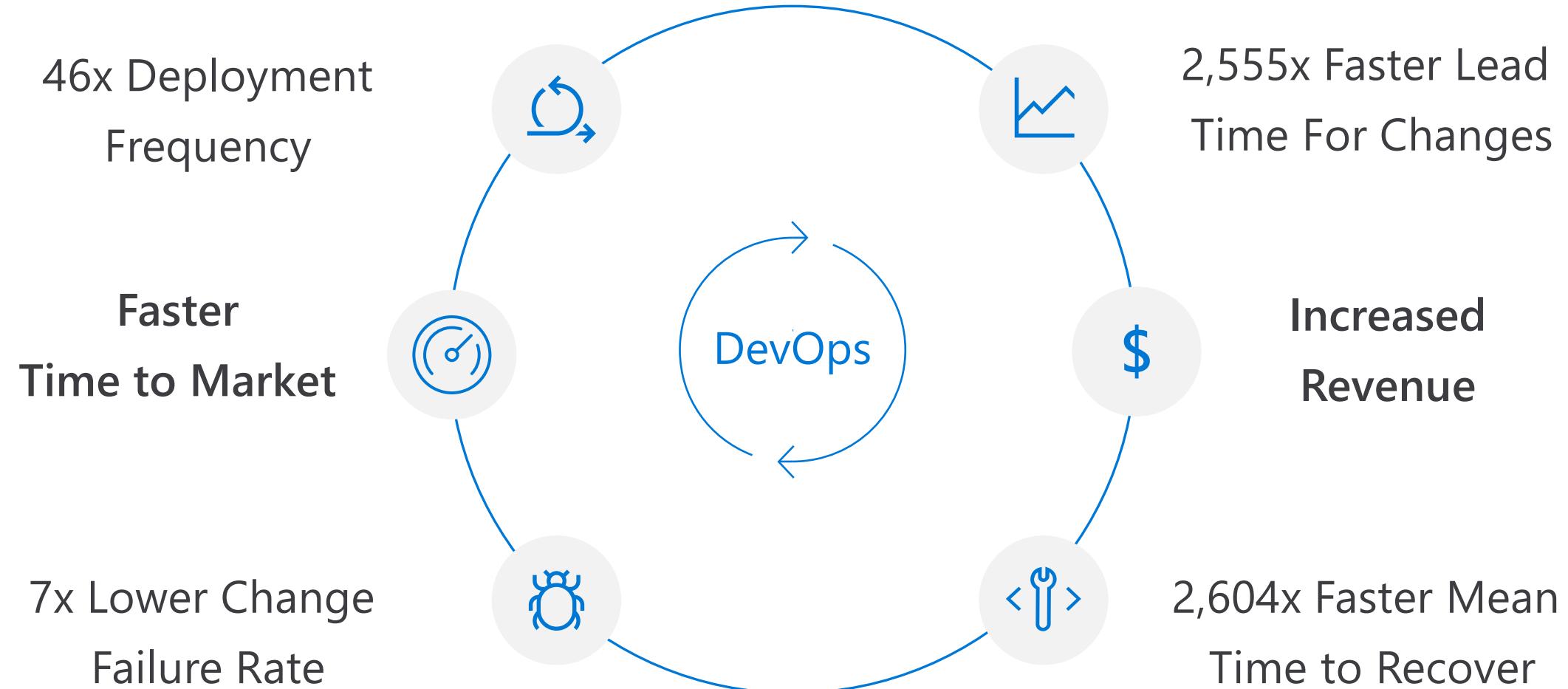
DevOps - Is it for everyone?

*Yes it is!!!*

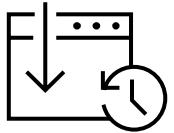
Think about how long would it take you to deliver a change of *one line of code within your application to production*



# High Performance DevOps Companies Achieve...



# What technologies do I need to support DevOps?



**Continuous Integration (CI)**

101010  
010101  
101010

**Continuous Deployment (CD)**



**Continuous Learning & Monitoring**

# Full Solution: People , Process and Tools

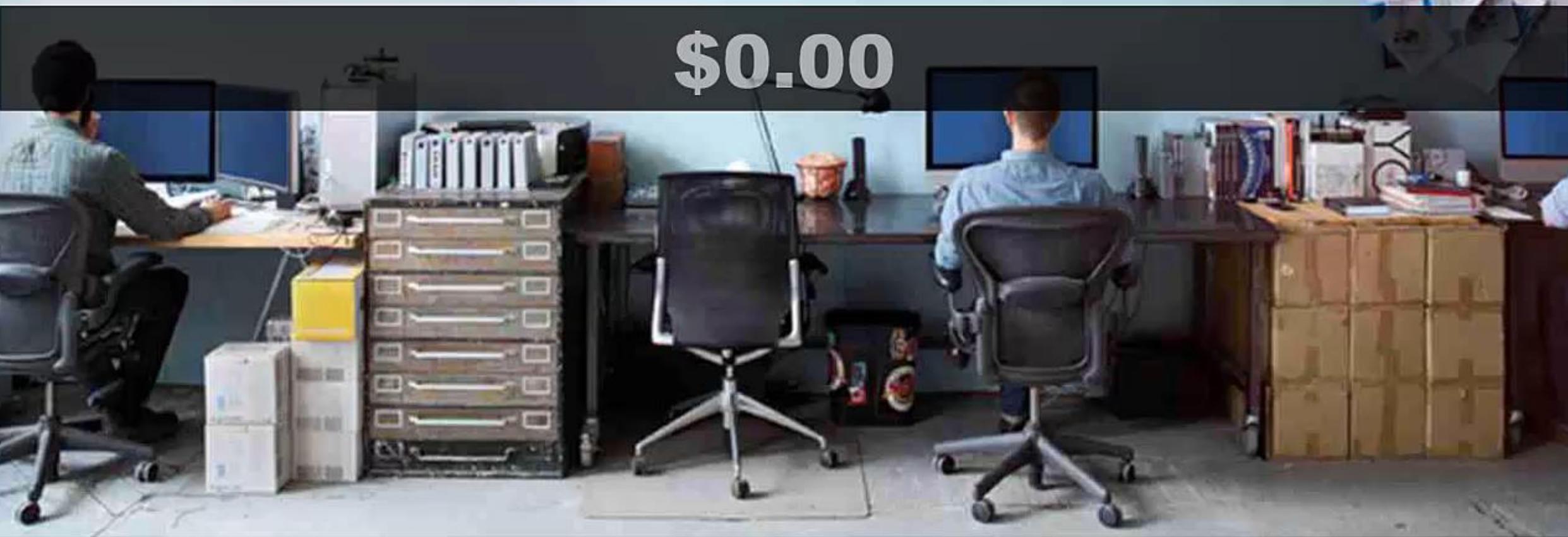




**AUGUST 1, 2012**

**9:29:00 AM**

**\$0.00**



At **10:15 AM** they managed to shut down the entire system.

In **45 minutes** Knight Capital Group  
lost **460 million** dollars.

# What are the Goals of Continuous Integration

**The main purpose of Continuous Integration is to prevent developers stepping over each other code and eliminate integration issues.**

## **CI Goals:**

- #1: Harness collaboration**
- #2: Enable parallel development**
- #3: Minimize integration debt**
- #4: Act as a quality gate**
- #5: Automate Everything!**

# Elements of Continuous Integration

## **Version Control System**

- Application / Test Code
- Configuration
- Database Definition
- Seed / Test Data
- Build / Deployment Scripts

## **Branching Strategy**

- Have a Branching strategy
- Keep changes small
- Commit frequently
- At least once per day for each developer

## **Automated Build**

- Runs after every commit/check-in
- Checks quality of the commit (Shift Left Testing)
- Separate Build machine (Build Server)
- Must run fast

## **Agreement of the Team**

**Continuous integration is a practice, not a tool. It requires a degree of commitment and discipline from your development team.**

# Continuous Integration Practices



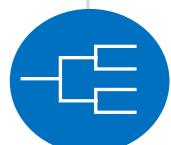
**Multi-Repo**



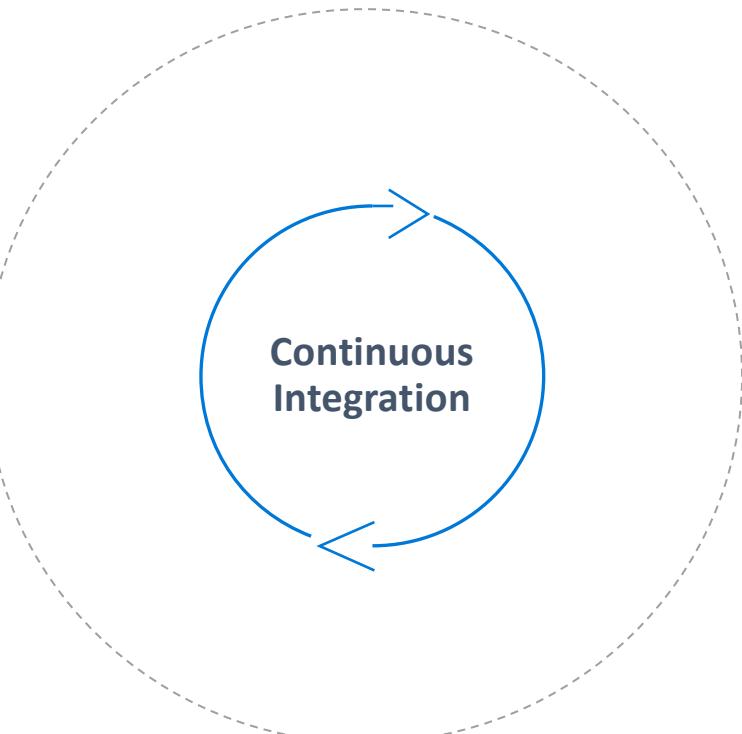
**Trunk Based Branching**



**Branch Policies**



**Parallel Builds**



**Culture & Collaboration**



**Unit Testing**



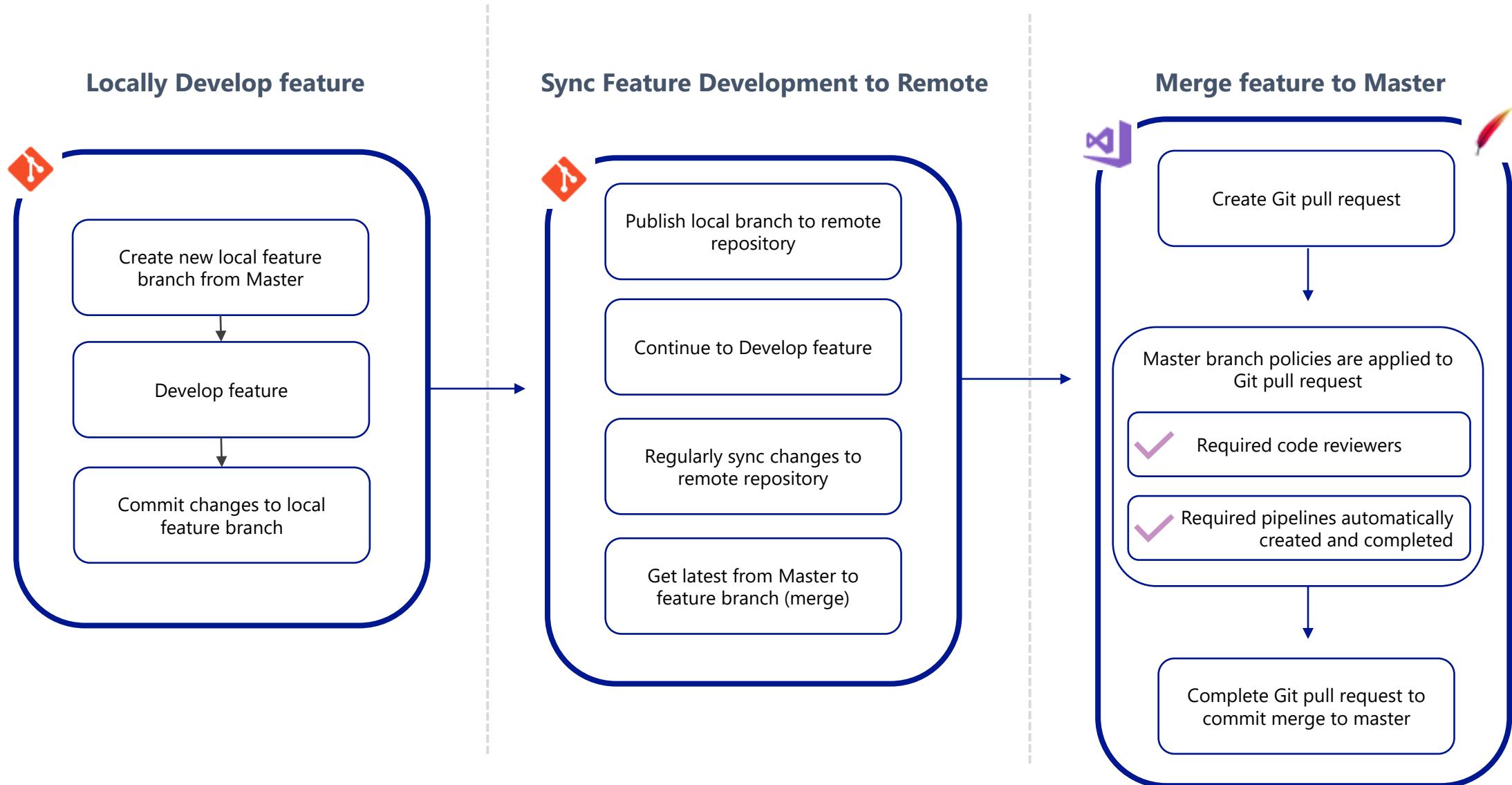
**Pull Requests**



**Security Scanning**



# Continuous integration end to end flow



# Start with this single question

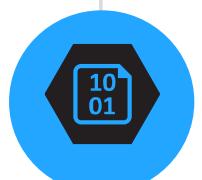
Where code deployments are most painful, you'll find the poorest software delivery performance, organizational performance, and culture.

How **big** is your Deployment Pain to Production?

# Continuous Delivery Practices



Automated Builds



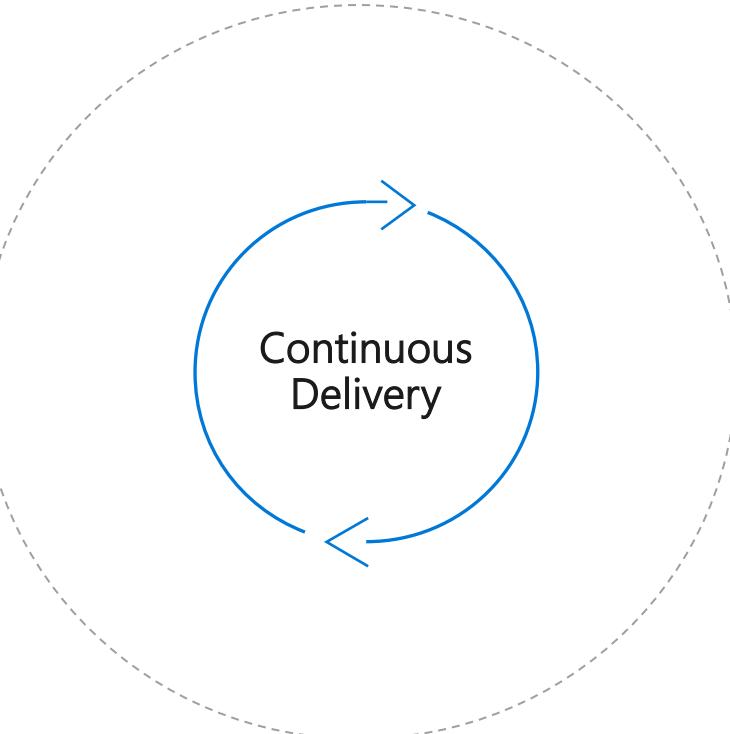
Build Once



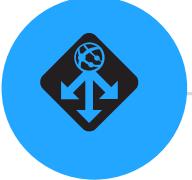
Trunk Always Prod  
Ready



Fix Immediately



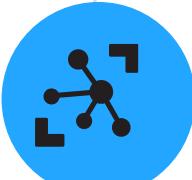
Everything in Source Control



Feature Flags



Blue/Green Deploy

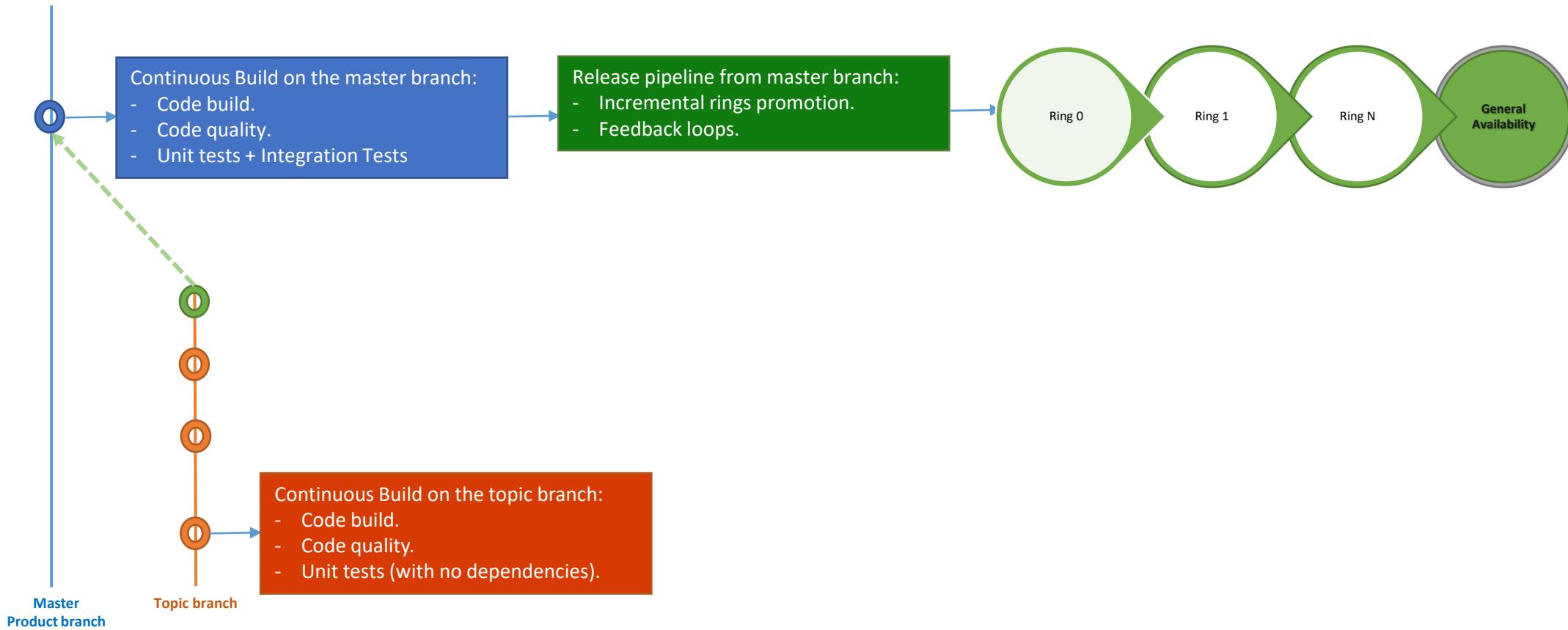


Canary Deployment



# Canary deployments

Make features incrementally available to end users. Enables low-risk collection of feedback



# Blue-green deployments

The management of infrastructure (networks, virtual machines, load balancers, and connection topology) in a [descriptive model](#), using the same versioning as source code

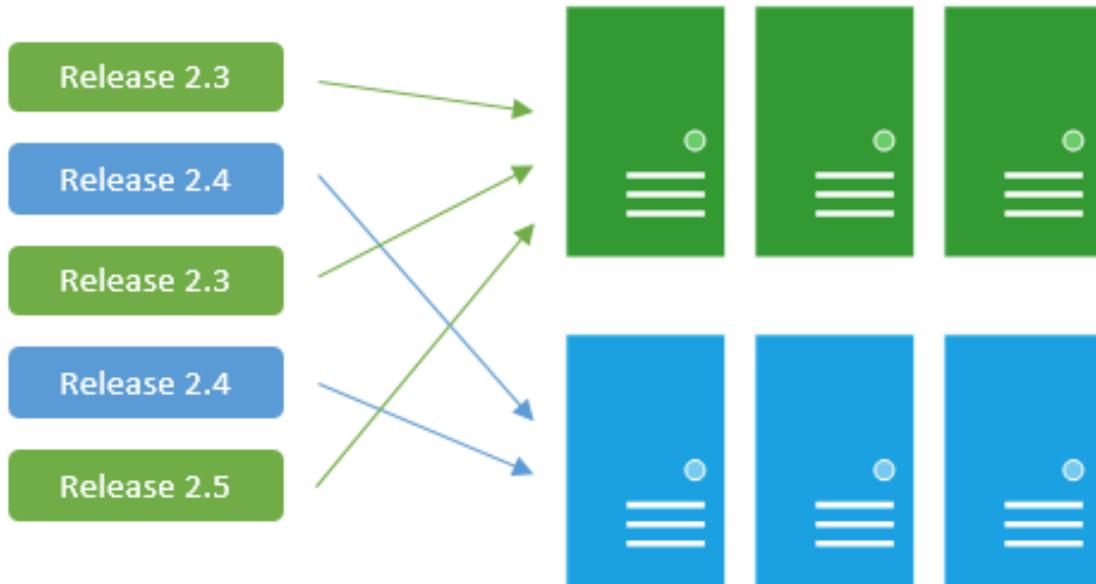


Image Source: <https://octopus.com/docs/deployment-patterns/blue-green-deployments>

- ❑ No-downtime Production Deployments
- ❑ Fast recovery by switching routers to previous environment
- ❑ Relies on keeping an existing (blue) version live while a new (green) one is deployed.
- ❑ Uses load balancing to direct increasing amounts of traffic to the green deployment
- ❑ If monitoring discovers an incident, traffic can be rerouted to the blue deployment still running

# Azure DevOps

Better together



Azure Boards



Azure Repos



Azure Pipelines



Azure Test Plans



Azure Artifacts

An end-to-end solution for organizations looking for an enterprise-grade toolchain

Fully Integrated  
with end  
to end  
traceability

Scalable to  
any team  
and project  
size

Highly  
available,  
multi region,  
hybrid  
cloud &  
on-prem

Customer  
Support

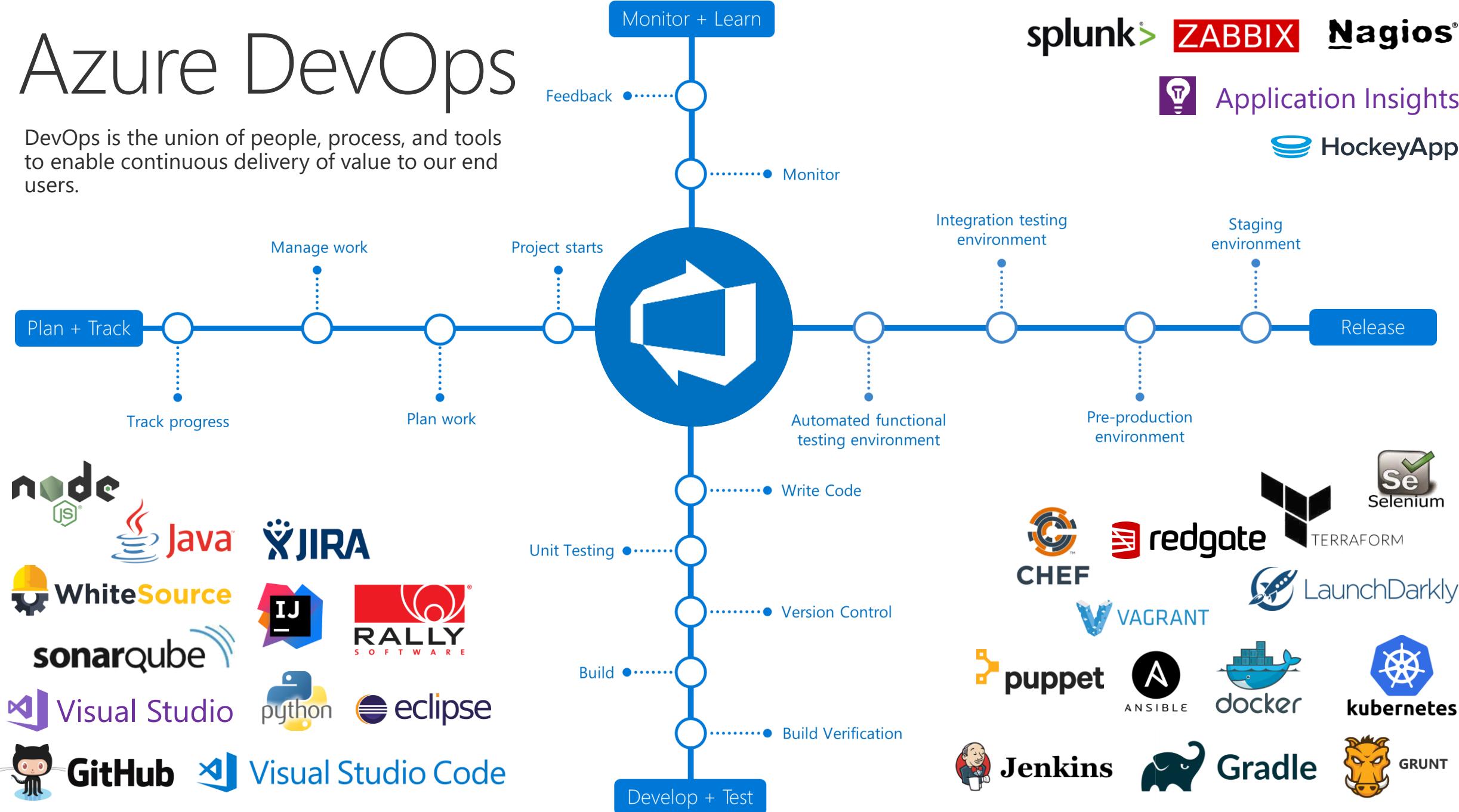
Consistent  
admin  
and access  
control



<https://azure.com/devops>

# Azure DevOps

DevOps is the union of people, process, and tools to enable continuous delivery of value to our end users.





# Azure Pipelines

## Always production deployable

Automatically build and test code  
on every commit

Cloud-hosted pipelines for Linux,  
macOS and Windows

Any language, any platform, any cloud

Native support for containers  
and Kubernetes





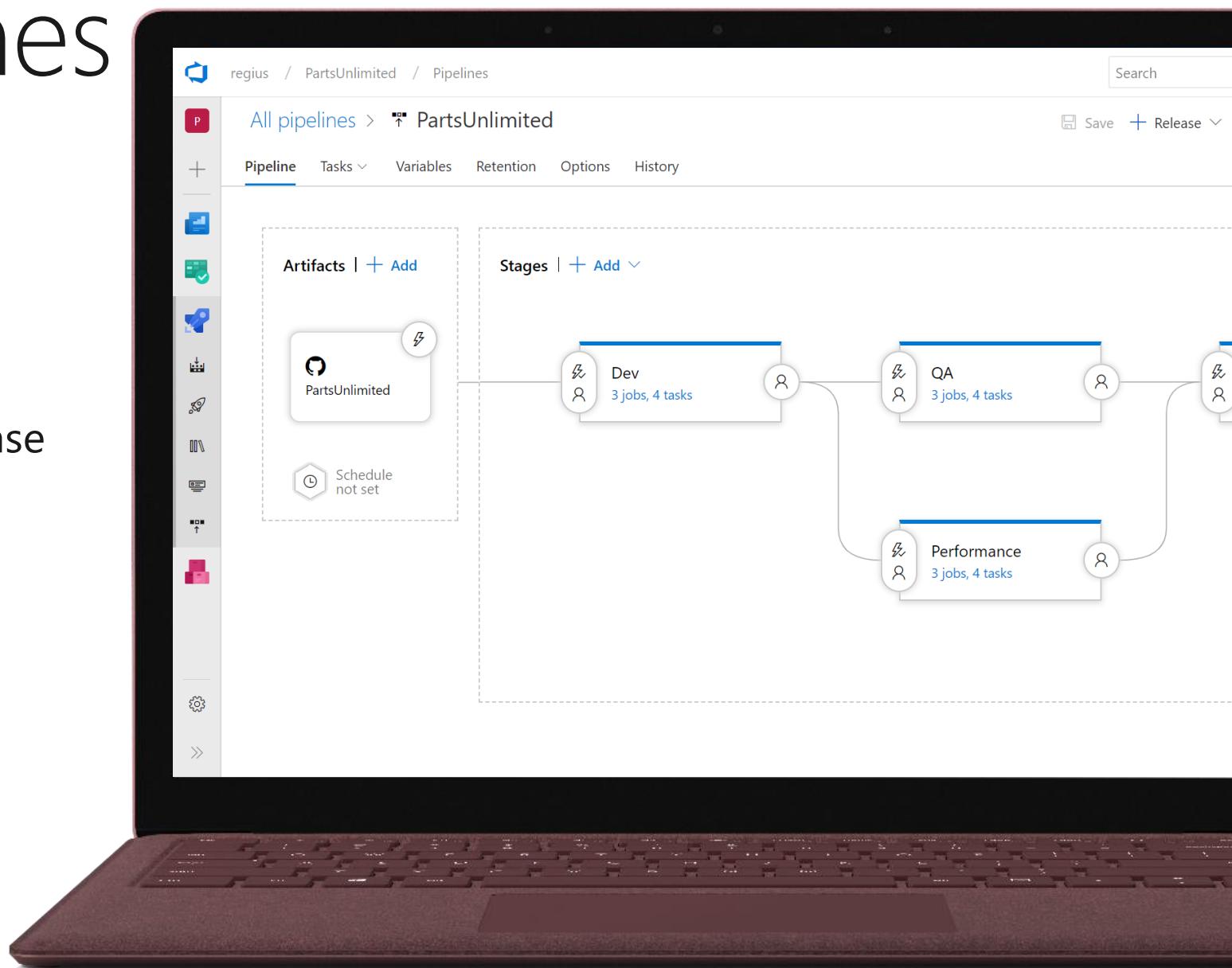
# Release safely. Release often.

Create environment rings for pre-release testing to scoped users

Staged environment releases

Pre- and post-deployment approvals

Gates can define automatic approval based on satisfied conditions



good

What's in a pipeline?

**Branch  
Policies**



# Protect your branches!

Decide on branching workflow as a team

Limit Merge Types  
(if you can)



Keep main branches free from direct commits!

Avoid committing dependencies

good

What's in a pipeline?

**Branch  
Policies**



# What's in a <sup>good</sup> pipeline?

Branch  
Policies

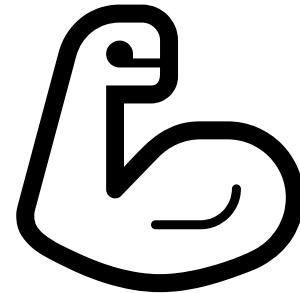
Pull Request  
Policies



# Strengthen your pull requests!

Set Minimum  
number of reviewers

Enforce Linked  
Work Items/Ticket  
Numbers



Check for comment  
resolutions

Auto-add reviewers

Run Pre-Builds!

# What's in a <sup>good</sup> pipeline?

Branch  
Policies

Pull Request  
Policies



# What's in a <sup>good</sup> pipeline?

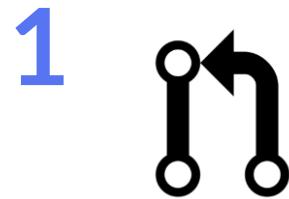
Branch  
Policies

Pull Request  
Policies

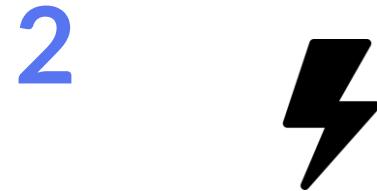
Automated  
Build



# Build Process



Pull Request Merged



New Changes Trigger Build



Successful Build Creates Artifact

# What's in a <sup>good</sup> pipeline?

Branch  
Policies

Pull Request  
Policies

Automated  
Build



# What's in a <sup>good</sup> pipeline?

Branch  
Policies

Pull Request  
Policies

Automated  
Build



# What's in a <sup>good</sup> pipeline?

Branch  
Policies

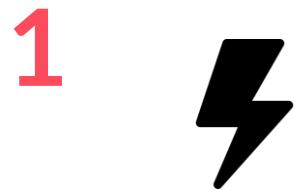
Pull Request  
Policies

Automated  
Build

Automated  
Release



# Release Process



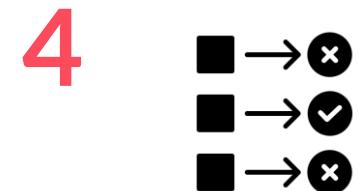
New Build Triggers  
a Release



Pass any  
Pre-Deployment  
Conditions



Deploy to  
environment



Pass any  
Post-Deployment  
Conditions

# What's in a <sup>good</sup> pipeline?

Branch  
Policies

Pull Request  
Policies

Automated  
Build

Automated  
Release



# What's in a <sup>good</sup> pipeline?

**Branch  
Policies**

**Pull Request  
Policies**

**Automated  
Build**

**Automated  
Release**

**Pre- and  
Post-Deployment  
Conditions**



# Secure your deployments!

Add pre- and post-deployment conditions to match your workflow



Consider pre-deployment approvals

Use Artifact filters

Use Gates to check for other statuses

Decide deployment queue settings

# What's in a <sup>good</sup> pipeline?

**Branch  
Policies**

**Pull Request  
Policies**

**Automated  
Build**

**Automated  
Release**

**Pre- and  
Post-Deployment  
Conditions**



# What's in a <sup>good</sup> pipeline?

**Branch  
Policies**

**Pull Request  
Policies**

**Automated  
Build**

**Automated  
Release**

**Pre- and  
Post-Deployment  
Conditions**

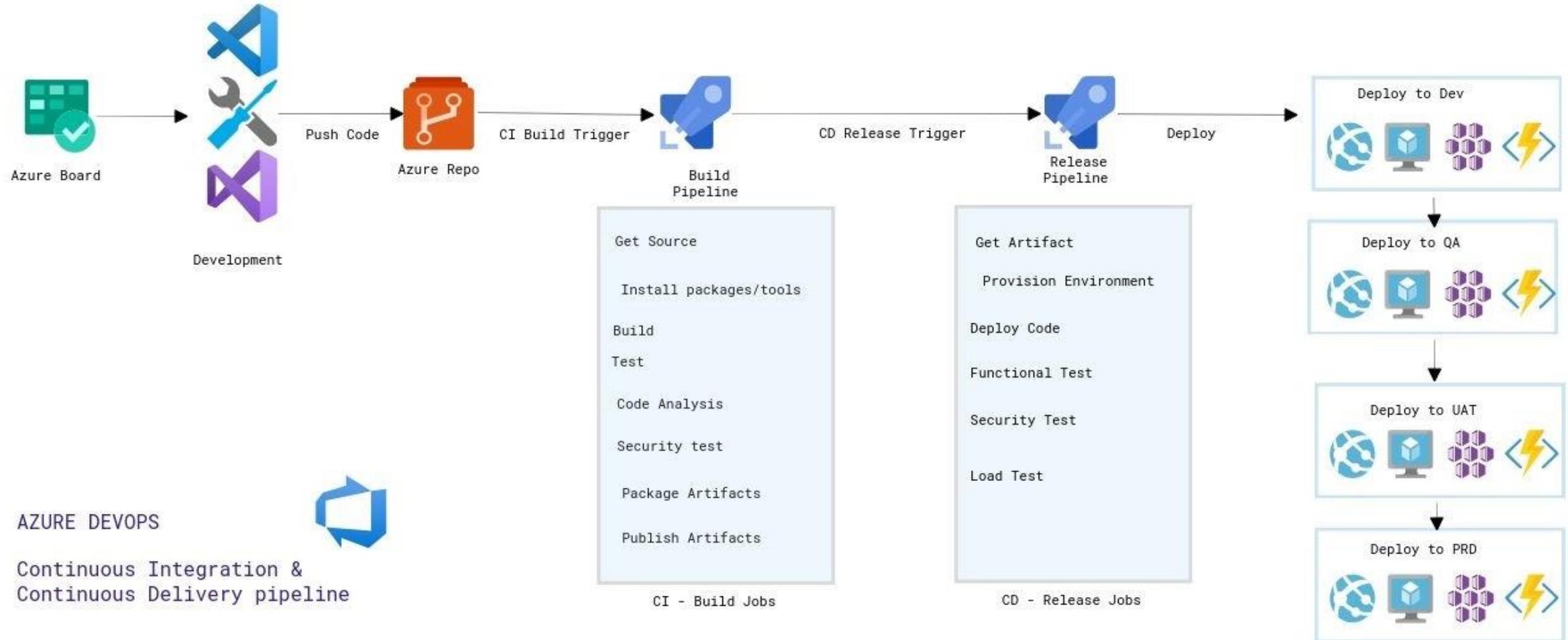


# What's in a <sup>good</sup> pipeline?

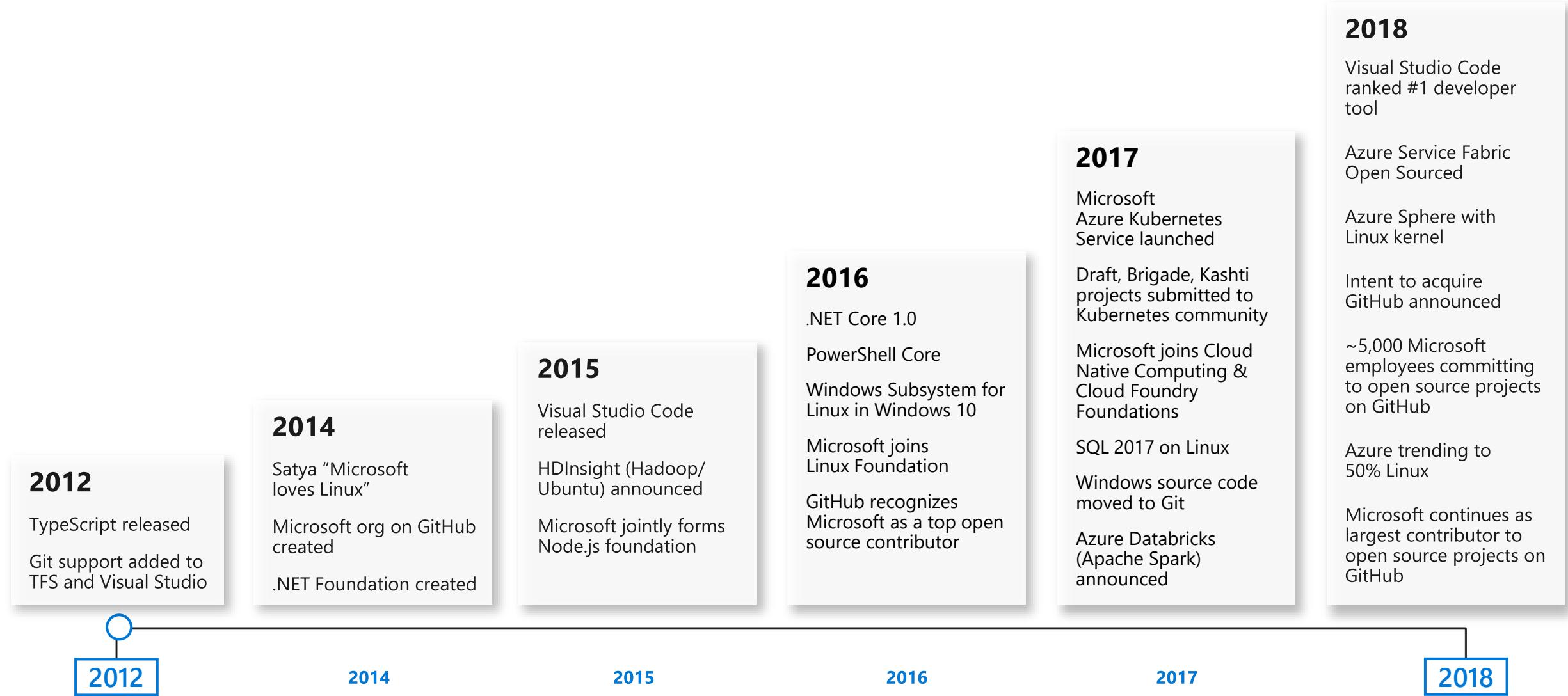
<b>Branch Policies</b>	<b>Pull Request Policies</b>	<b>Automated Build</b>	<b>Automated Release</b>	<b>Pre- and Post-Deployment Conditions</b>	<b>Multi-Stage Deployment</b>
------------------------	------------------------------	------------------------	--------------------------	--	-------------------------------



# DevOps pipeline



# Microsoft ❤️ Open Source



# GitHub

Building on a foundation of DevOps in Azure, GitHub adds three key ingredients to developer workflows helping ensure your digital transformation success



Open source



Innersource

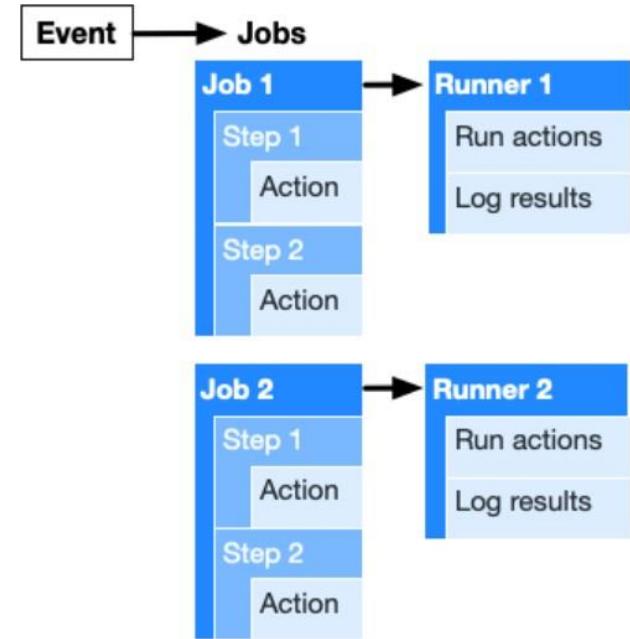


Security and compliance

# GitHub Actions

## Actions

- Automate
- Customize
- Execute software development workflows
- Event driven



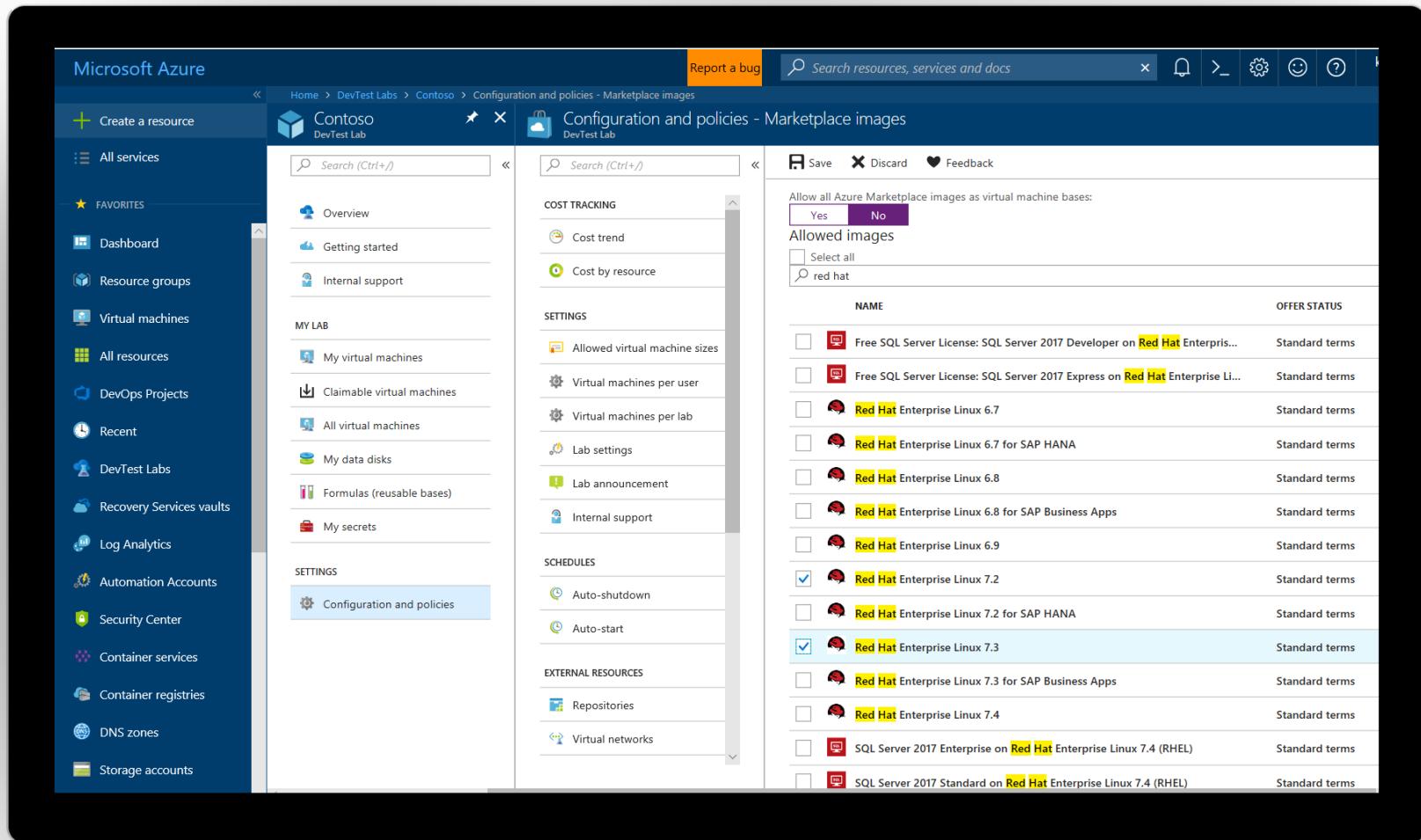
## Starter Workflows

- Workflow -> Automated procedure added to the repository
- Workflow files for getting started

# Self-Service Dev/Test Environments

## Azure Lab Services

- Simplify cloud environment management for developers and testers.
- Enforce policies and control costs with full visibility
- Use templates, custom images and formulas to reproduce environments.
- Orchestrate with Azure Pipelines or integrate using REST API



# Infrastructure and Configuration as Code

Azure Resource Manager, Automation & 3<sup>rd</sup> Party Integrations

→ Infrastructure as Code,  
built-in

→ Azure Config & Automation

→ Support for 3<sup>rd</sup> party and OSS  
tooling such as Terraform,  
Ansible, Chef, Puppet &  
SaltStack



TERRAFORM



ANSIBLE



CHEF



puppet



SALTSTACK

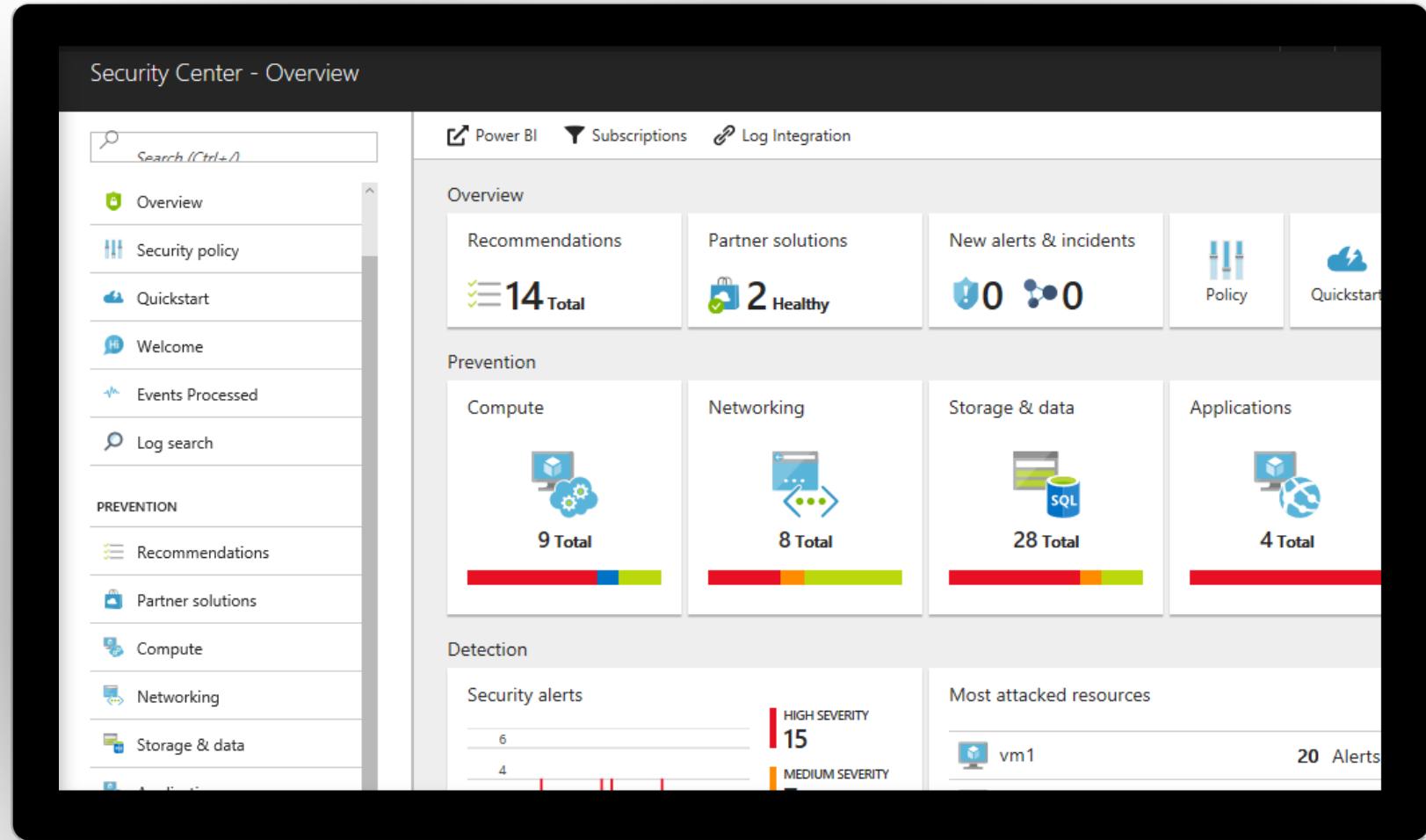
The screenshot shows the Microsoft Azure portal interface. On the left, the 'Resource groups' blade is open, displaying a list of resource groups: AustraliaSEDevelopment, AustraliaSEProduction (selected), autoShutdown, cloud-shell-storage-westus, DefaultResourceGroup-EUS, and securitydata. On the right, the 'AustraliaSEProduction - Automation script' blade is open, showing the 'Automation script' section. It includes tabs for Template, Parameters, CLI, PowerShell, .NET, and Ruby. The 'Template' tab is selected, displaying an ARM template JSON code snippet:

```
$schema: "https://schema.management.azure.com/schemas/2015-01-01/deploymentTemplate.json#"
contentVersion: "1.0.0.0"
parameters:
  "dnszones_onazure.io_name": {
    "defaultValue": "onazure.io",
    "type": "String"
  },
  "NS_@_name": {
    "defaultValue": "@",
    "type": "String"
  },
  "SOA_@_name": {
    "defaultValue": "@",
    "type": "String"
  },
  "_vote_name": {
    "defaultValue": "vote",
    "type": "String"
  },
  "_draft_name": {
    "defaultValue": "draft",
    "type": "String"
  },
  "A_devops_name": {
    "defaultValue": "devops",
    "type": "String"
  },
  "A_*_.draft_name": {
    "defaultValue": "*.draft",
    "type": "String"
  },
```

# Continuous Security

## Azure Security Center

- Gain full visibility and control of your cloud security state
- Leverage ML to Proactively identify and mitigate risks to reduce exposure to attacks
- Quickly detect and respond to threats with advanced analytics



The screenshot shows the Azure Security Center - Overview page. The left sidebar contains navigation links: Search (Ctrl+F), Overview, Security policy, Quickstart, Welcome, Events Processed, Log search, PREVENTION (Recommendations, Partner solutions, Compute, Networking, Storage & data, Applications), and DETECTION (Security alerts, Most attacked resources). The main area is divided into sections: Overview (Recommendations: 14 Total, 2 Healthy, 0 Critical, 0 High, 0 Medium, 0 Low; New alerts & incidents: 0 Critical, 0 High, 0 Medium, 0 Low), Prevention (Compute: 9 Total, Networking: 8 Total, Storage & data: 28 Total, Applications: 4 Total), and Detection (Security alerts: 6 High Severity, 15 Medium Severity, 4 Low Severity; Most attacked resources: vm1 with 20 Alerts).

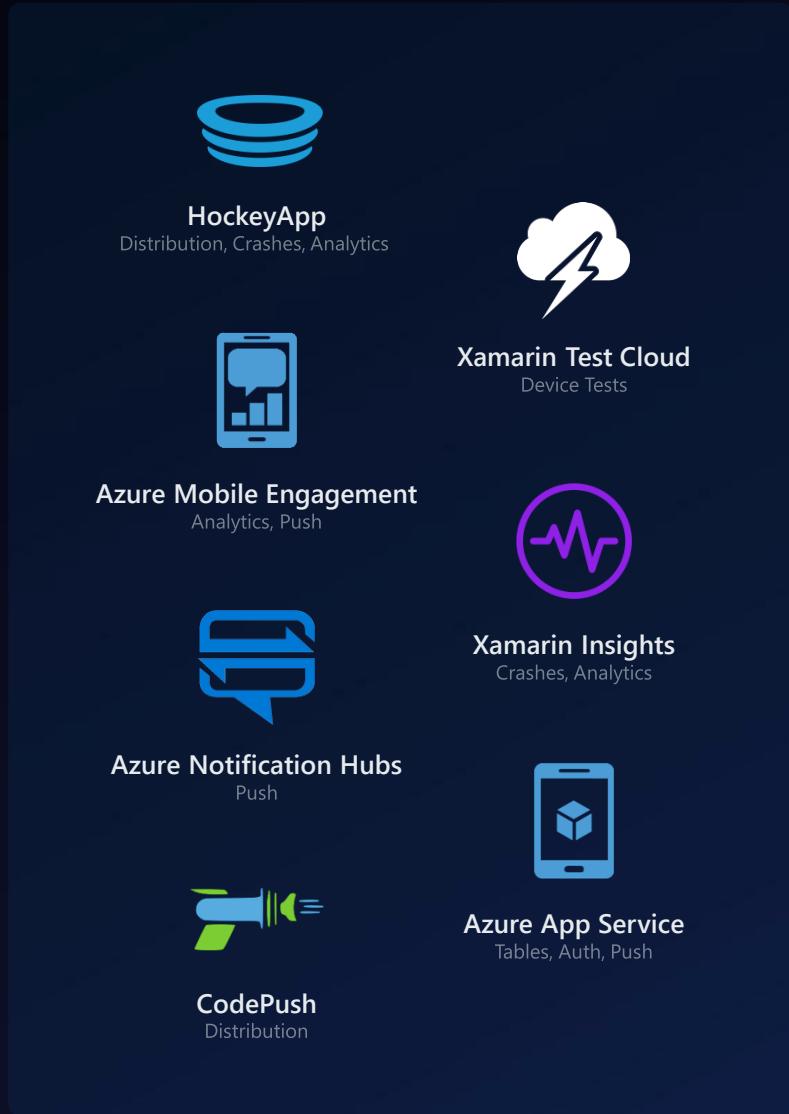
# Smarter Insights, Faster

## Azure Monitor, Application Insights & Log Analytics

- Pre-defined solutions with smart thresholds
- Visualize data in intuitive and customizable dashboards
- Separate the signal from the noise and accelerate root-cause analysis
- Integrate your existing processes & tools like Service Now



# A unified destination at Microsoft for hosted developer services.



PREVIOUSLY: SEPARATE SERVICES ACROSS MICROSOFT

NOW: ONE UNIFIED SERVICE

# Github or Azure Devops

## Source Control



CloudRepos



GitHub repos

Based on Git

## CI / CD



Cloud Pipelines



GitHub Actions

Based on the Azure  
Pipelines technology

## Code packages



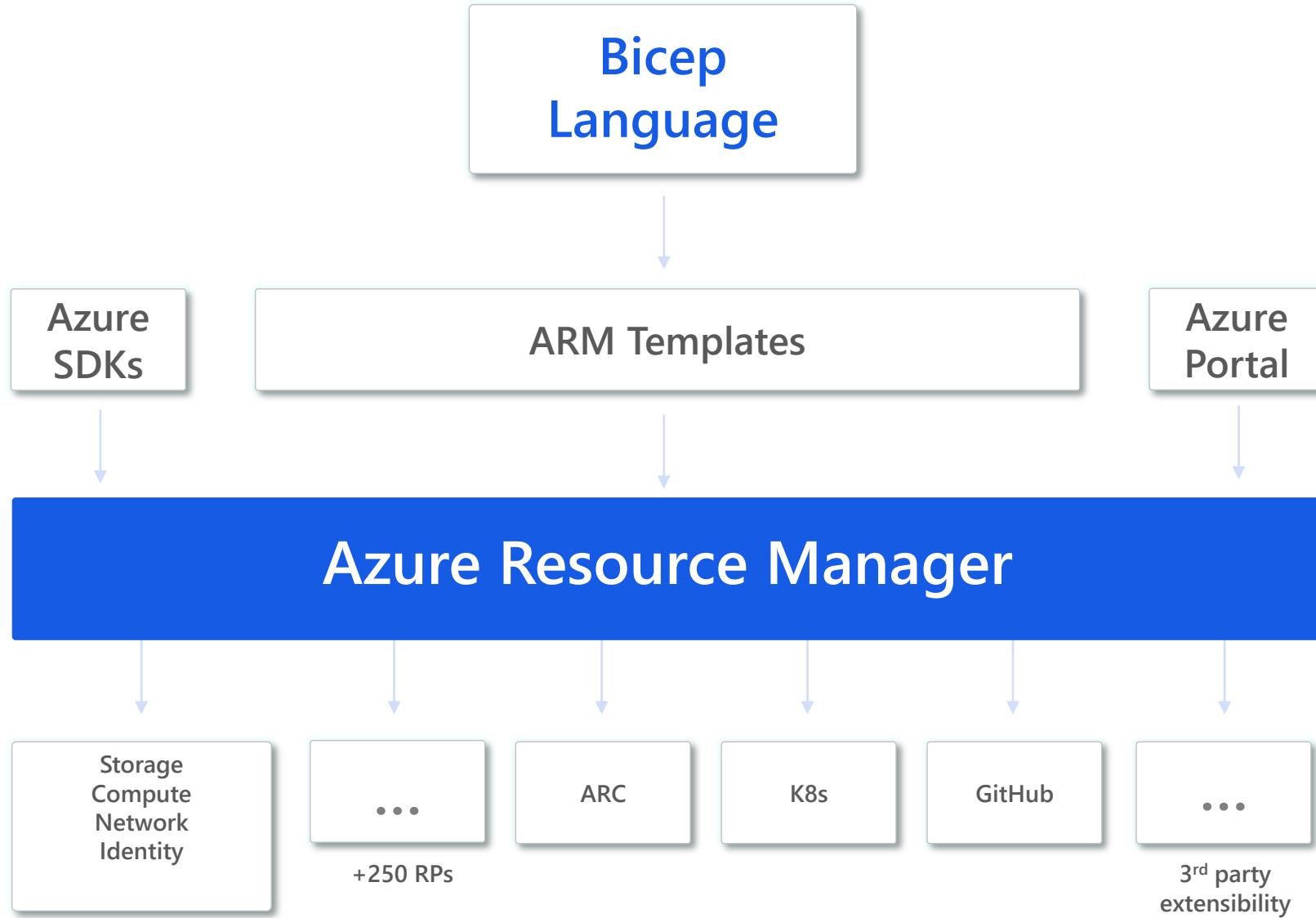
Cloud Artifacts



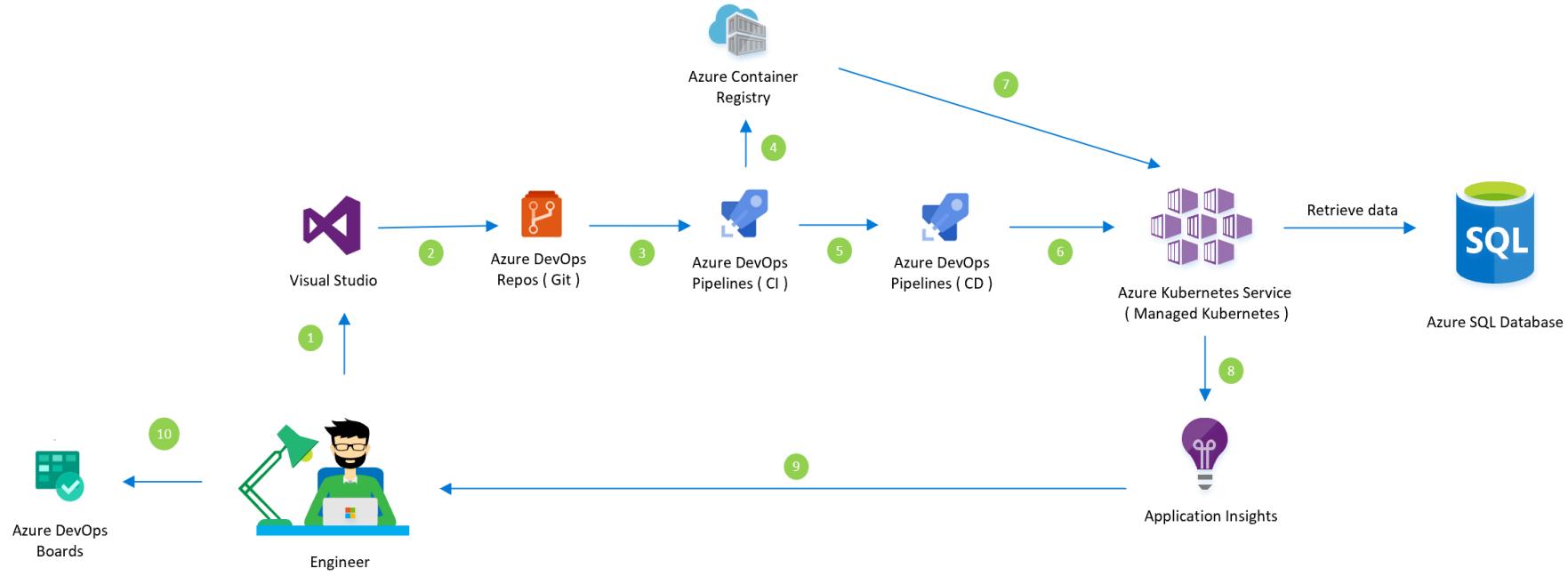
GitHub Package  
Registry

Based on standard, open  
package formats

# Project 'Bicep'



# Demo 1: Deploy a multi-container application to AKS and debug with Devspaces



1 Change Application Source code

2 Commit code change

3 Continuous integration triggers application build, container image build and unit tests

4 Container image pushed to Azure Container Registry

5 Continuous deployment triggers orchestrates deployment of application artefacts with environment specific parameters

6 Deploy application to Azure Kubernetes Service ( AKS )

7 Launch container by using container image from Azure Container Registry ( ACR )

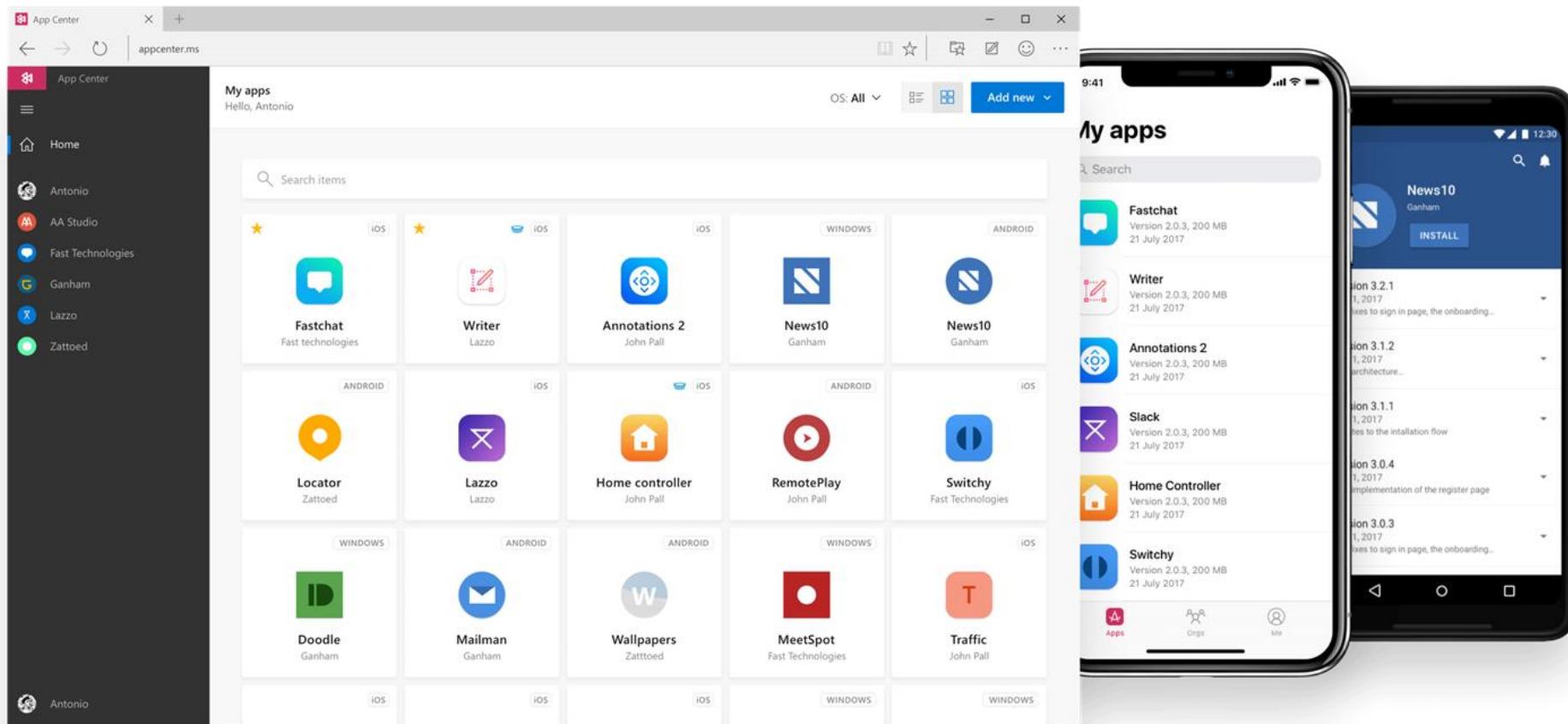
8 Application Insights collects and analyses health, performance and usage data

9 Review health, performance and usage information

10 Update backlog item

Try this demo : <https://github.com/sajeetharan/DevOps-and-Containers>

# Demo 2: AppCenter



# Next steps:



## Exam AZ-400: Designing and Implementing Microsoft DevOps Solutions

<https://docs.microsoft.com/en-in/learn/certifications/exams/az-400>

# Upcoming DevOps Events

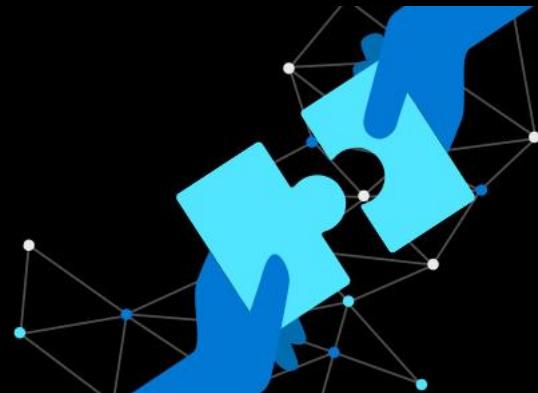
## Remote Engineering DevOps Summit

Agility and Adaptiveness for Remote Engineering Teams

Oct 28 – 29 : 10:00 AM–12:00 PM UTC+8

## DevOps Virtual Summit

Collaborate. Innovate. Accelerate.



Nov 25 – 26 : 9:00 AM–1:15 PM UTC+8

Register at:

<https://info.microsoft.com/ww-landing-remote-engineering-devops-summit-splash.html>

Scan QR Code here:



Register at:

[aka.ms/APACDevOpsSummit\\*](http://aka.ms/APACDevOpsSummit)

Scan QR code here:



\*Link will go live on 26<sup>th</sup> Oct



Let's have some fun! Join Kahoot here

