



# Developing Applications with Containers

Microsoft Services





# Module 6 – DevOps with Containers

Microsoft Services



# Agenda

- Container Devops Value Proposition
- What is DevOps?
- Azure DevOps Overview
- Containerized Workflow Pipeline
- Azure DevOps: Continuous Integration - Windows/Linux
- Azure DevOps : Continuous Deployment - SF/AKS



# Software Innovation Collision

## **Containers**

Standardization of the application packaging



## **DevOps**

Optimizing how applications  
are built, delivered and  
managed



**Better  
Software**



## **Microservices**

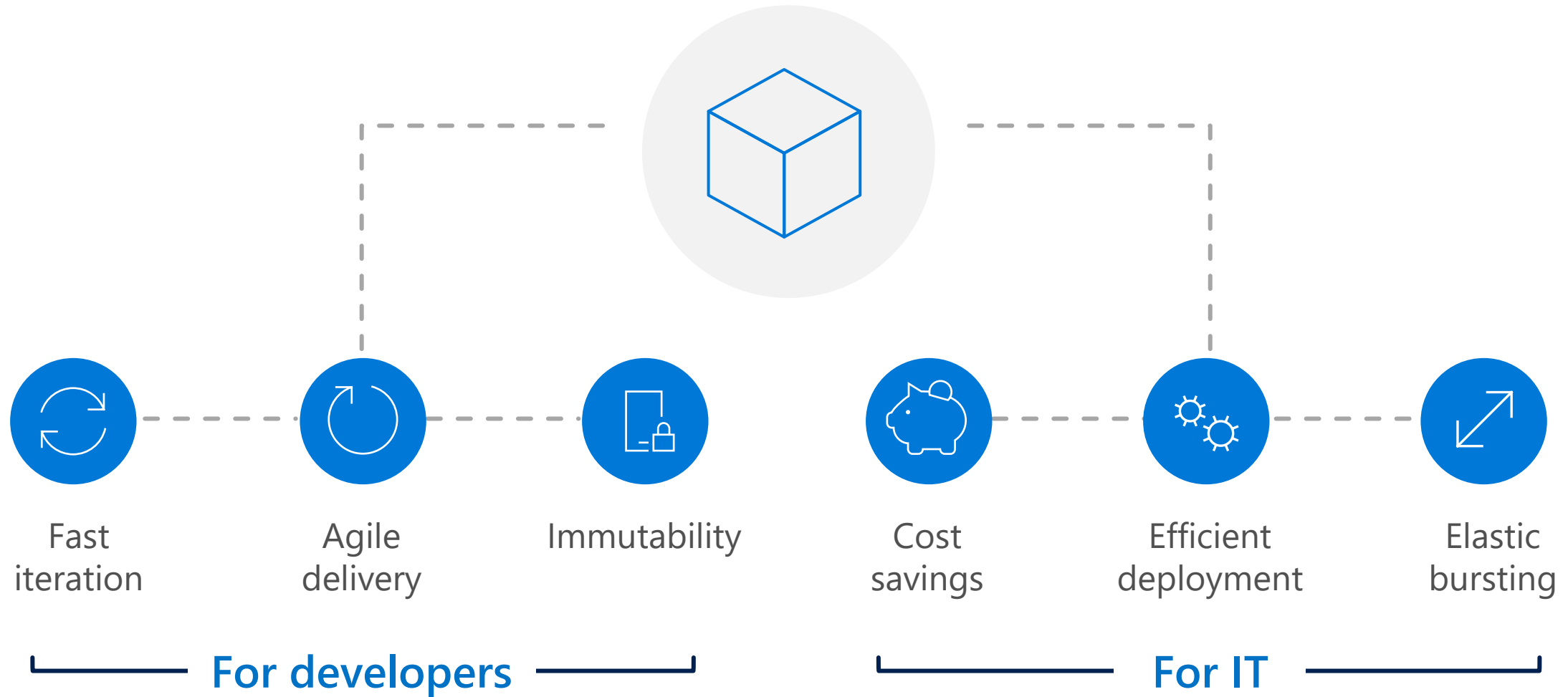
Optimizing how applications  
are composed



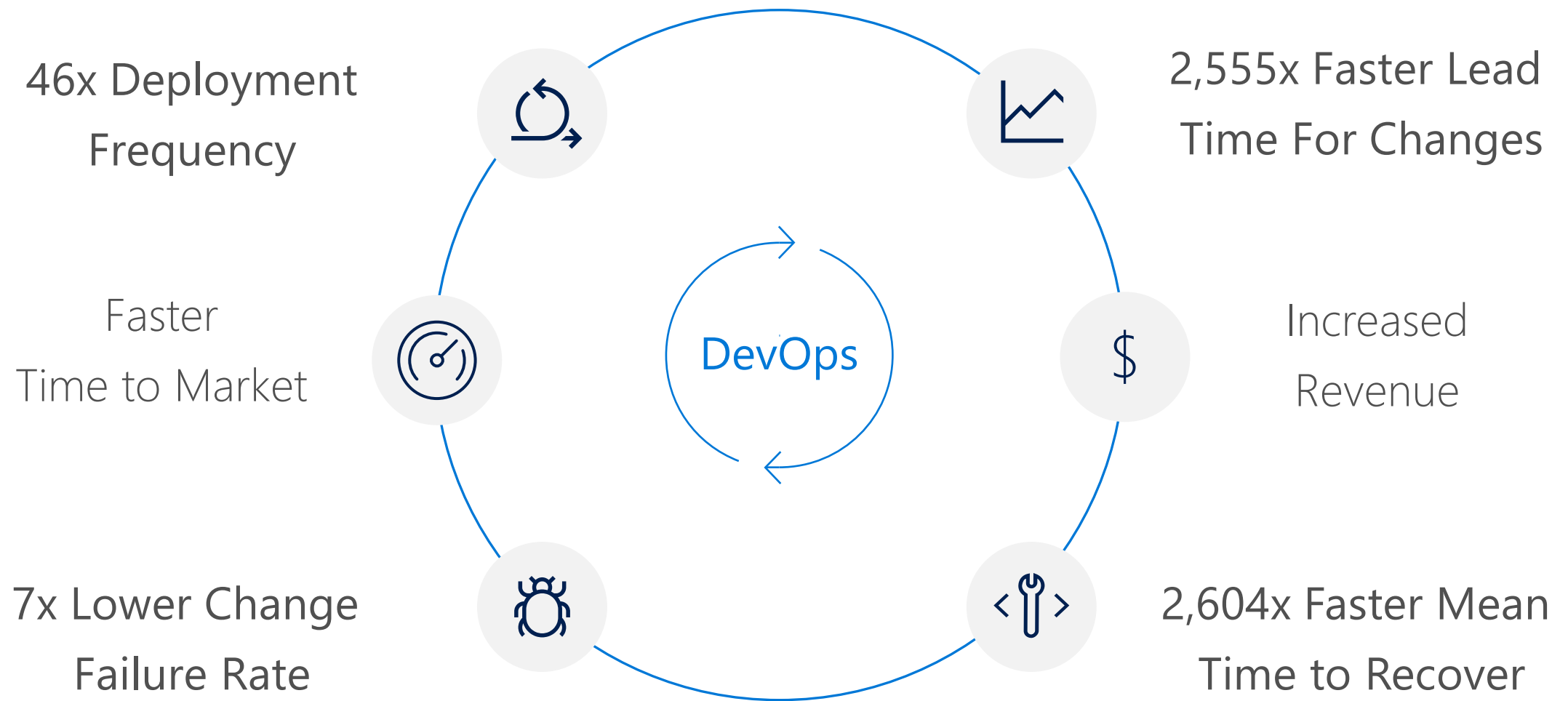
## **Cloud Orchestrators**

Efficient ways to deliver Infrastructure for  
Applications

# Devops View Of The Container Advantage



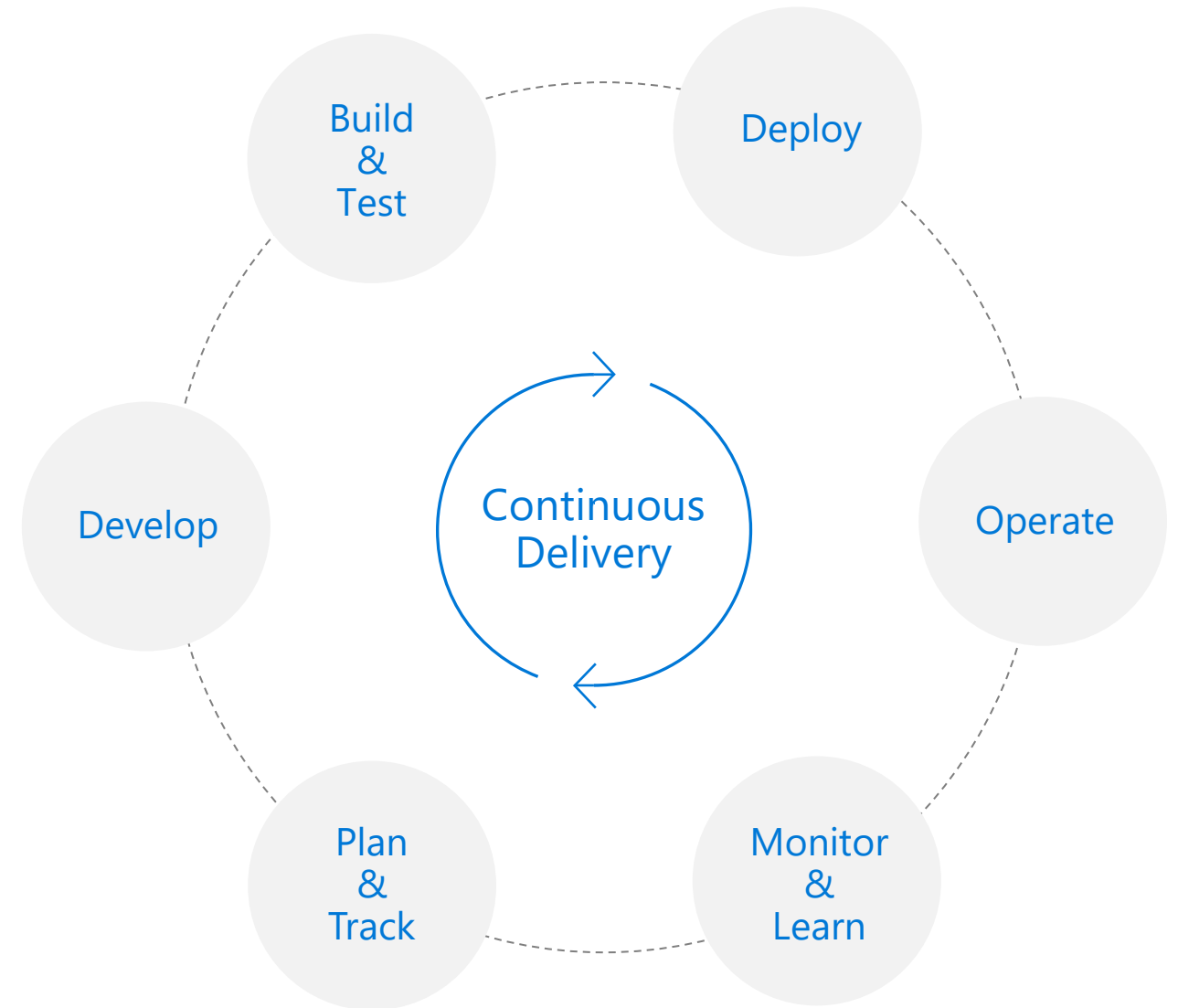
# High Performance Devops Companies Achieve...



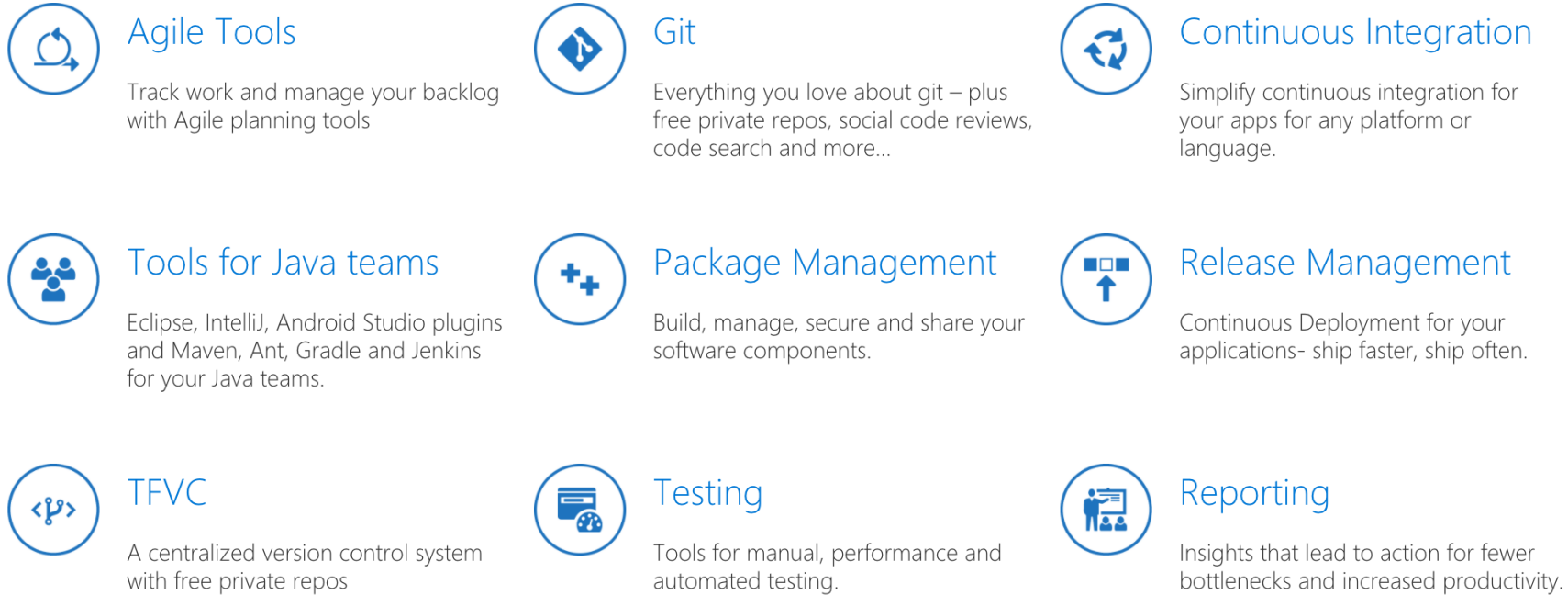
# What is DevOps?



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



# Azure DevOps



Increase productivity with powerful features



Code in any IDE/language and build applications for any target platform.



# Azure DevOps



## Azure Boards

Deliver value to your users faster using proven agile tools to plan, track, and discuss work across your teams.



## Azure Test Plans

Test and ship with confidence using manual and exploratory testing tools.



## Azure Pipelines

Build, test, and deploy with CI/CD that works with any language, platform, and cloud. Connect to GitHub or any other Git provider and deploy continuously.



## Azure Artifacts

Create, host, and share packages with your team, and add artifacts to your CI/CD pipelines with a single click.



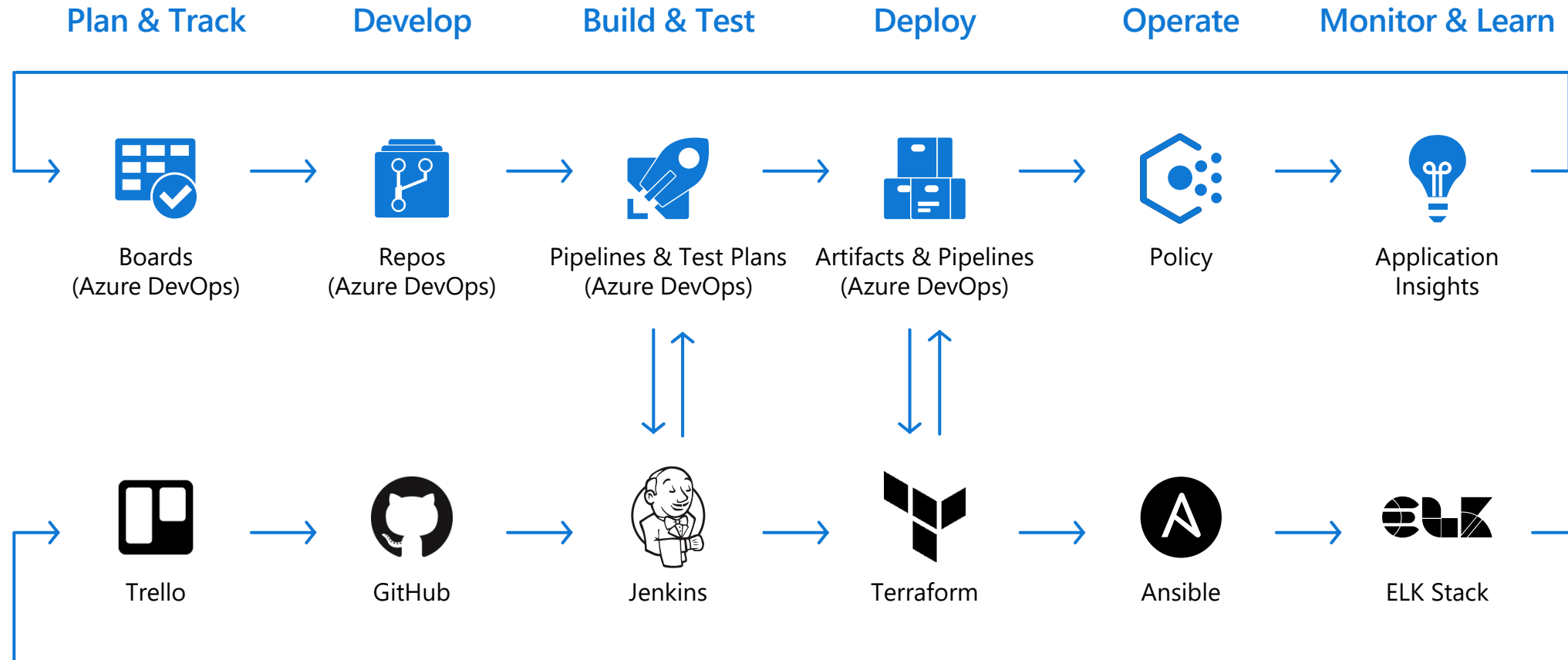
## Azure Repos

Get unlimited, cloud-hosted private Git repos and collaborate to build better code with pull requests and advanced file management.

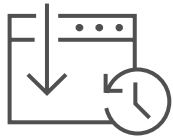


<https://azure.com/devops>

# DevOps on Azure framework



# What Technologies Do I Need To Support Devops?



## Continuous Integration (CI)

---

- Improve software development quality and speed.
- When you use Azure Pipelines or Jenkins to build apps in the cloud and deploy to Azure, each time you commit code, it's automatically built and tested and bugs are detected faster.



## Continuous Deployment (CD)

---

- By combining continuous integration and infrastructure as code (IaC), you'll achieve identical deployments and the confidence to deploy to production at any time.
- With continuous deployment, you can automate the entire process from code commit to production if your CI/CD tests are successful.



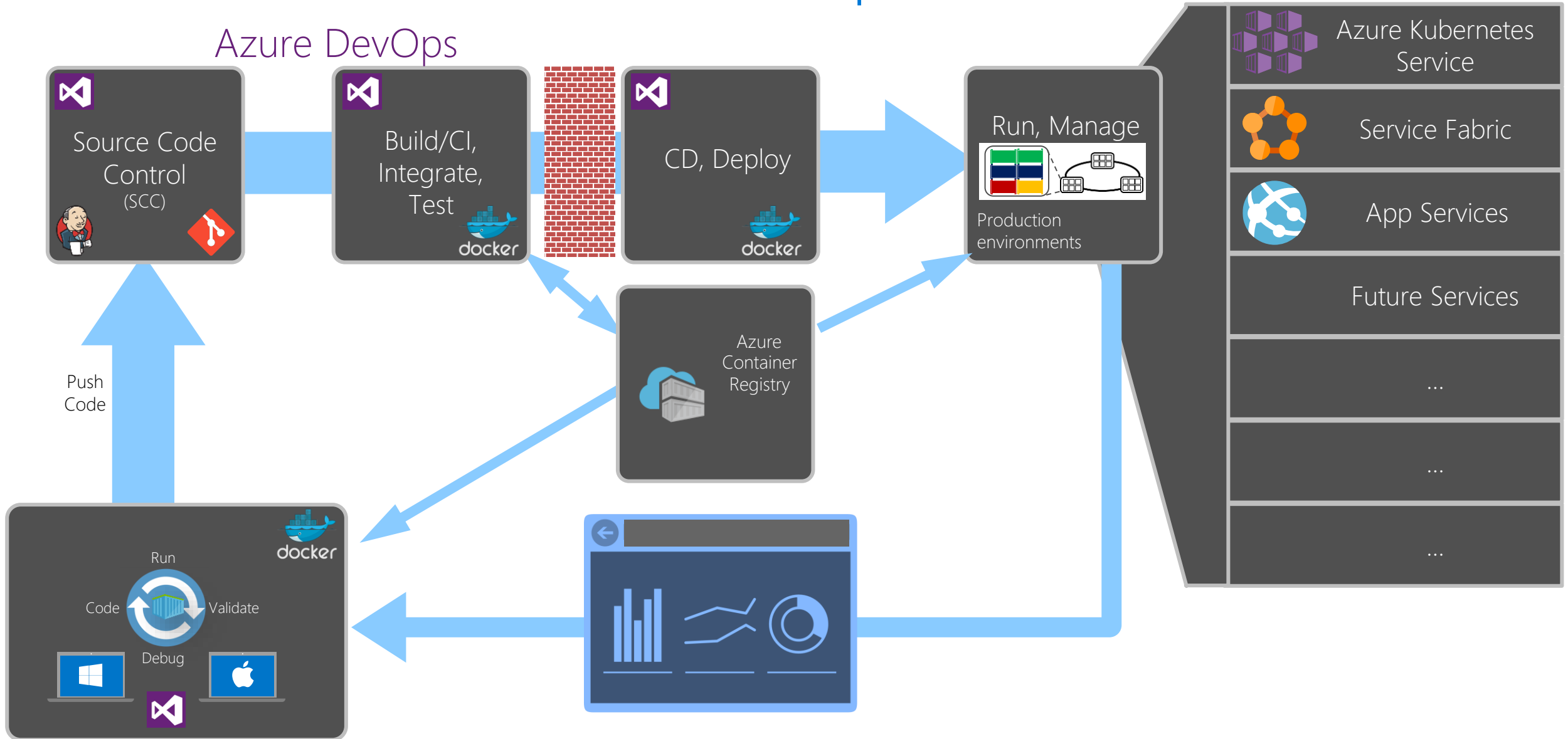
## Continuous Learning & Monitoring

---

- With Azure Application Insights you can identify how your applications are performing and test if the recent deployment made things better or worse.
- Using CI/CD practices, paired with monitoring tools, you'll be able to safely deliver features to your customers as soon as they're ready.

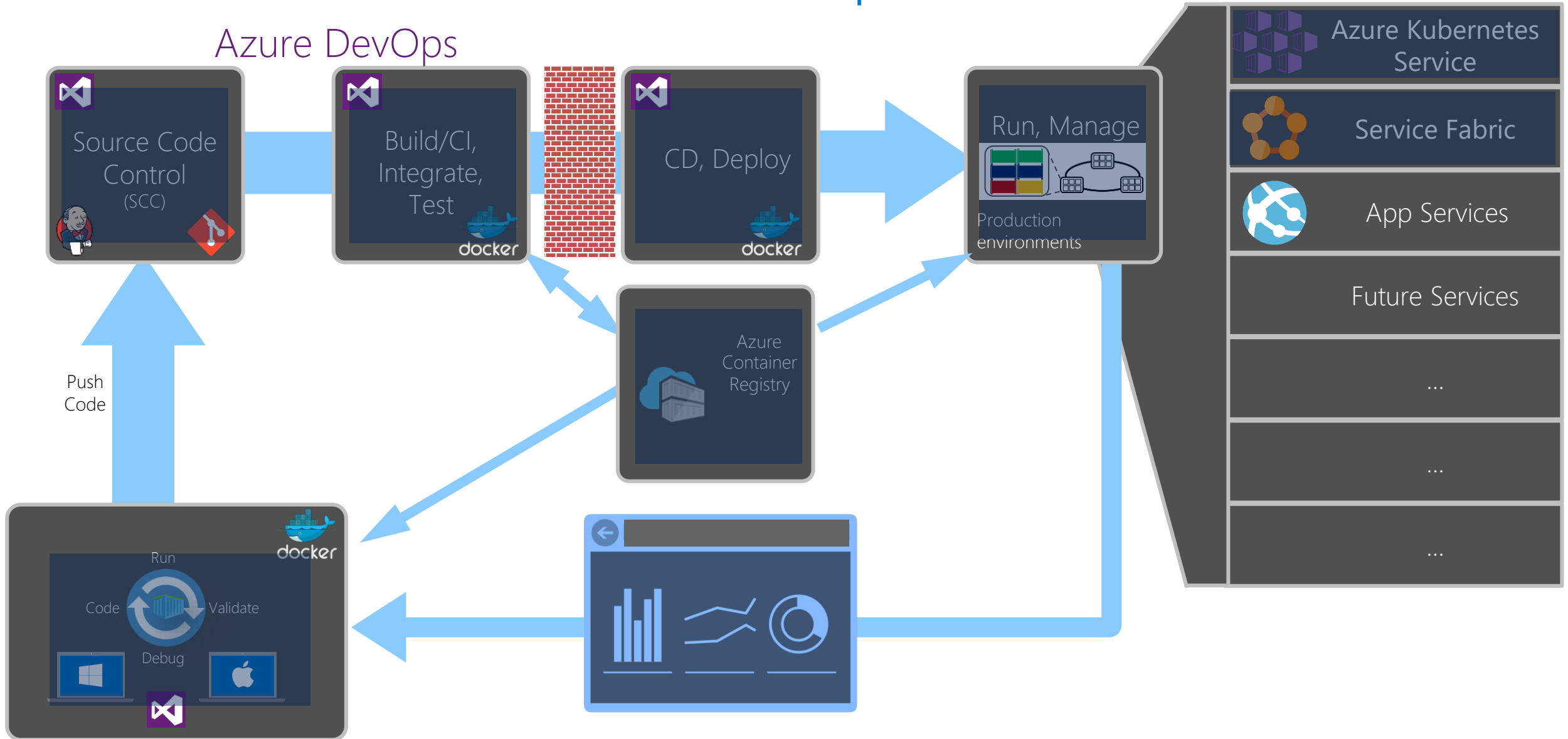
# Containerized Workflow Pipeline

Azure DevOps

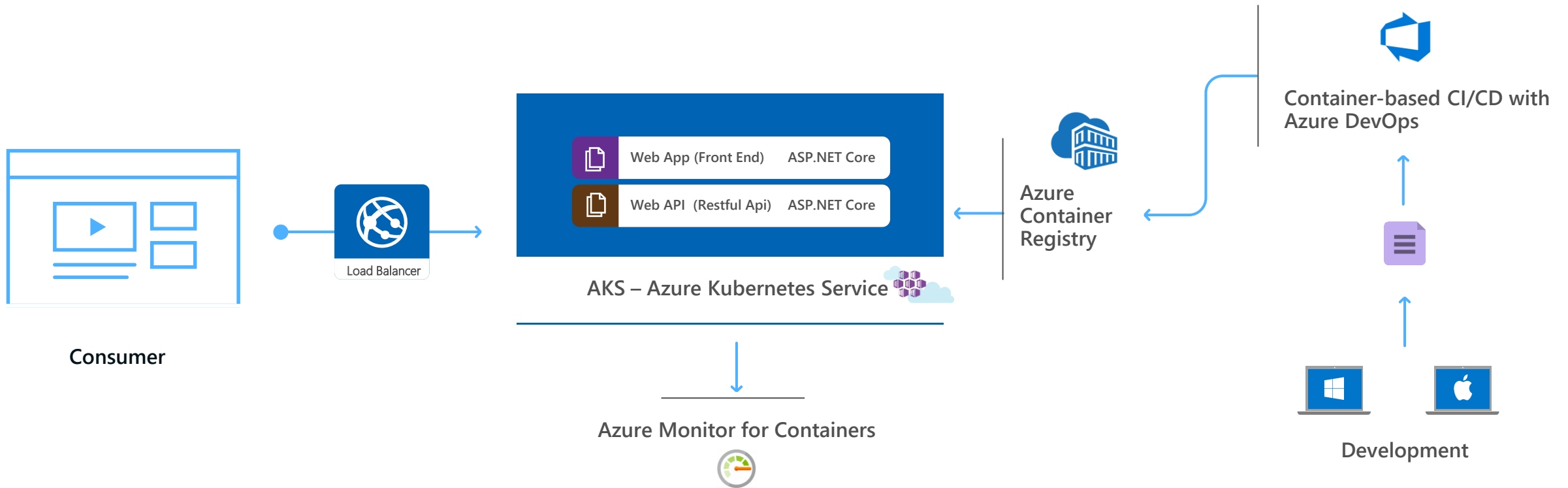


# Containerized Workflow Pipeline With AKS/SF

Azure DevOps



# CI/CD Pipeline | Azure DevOps and AKS





# Azure DevOps Agents

- **Agents:** To build your code or deploy your software you need at least one agent.
  - **Private Agents :** More control to install dependent software needed for builds and deployments
  - **Hosted Agents:** Microsoft provides the agent and takes care of the maintenance and upgrades
- **Linux Agents (Azure Pipeline / TFS 2017):** <https://docs.microsoft.com/en-us/azure/devops/pipelines/agents/v2-linux?view=vsts>
- **Windows Agents (Azure Pipeline / TFS 2017):** <https://docs.microsoft.com/en-us/azure/devops/pipelines/agents/v2-windows>

Agents for pool Hosted Ubuntu 1604 [Download agent](#)


Enabled	Name	State	Current status
<input checked="" type="checkbox"/>	Hosted Agent	Online	Idle


Agents for pool Hosted VS2017 [Download agent](#)

Enabled	Name	State	Current status
<input checked="" type="checkbox"/>	Hosted Agent	Online	Idle


# Build Pipeline


Build for Multi-Container ASP.NET Core Application using Linux Containers


**Agent job 1** 


 Run on agent


---


 **Build and Push WebAPI Image**  
Docker

 **Build and Push WebApp Image**  
Docker


 **Replace build number in backend-webapi.yaml file co...**  
Command Line


 **Replace build number in frontend-webapp.yaml file**  
Command Line

 **Publish Artifact: frontend-webapp.yaml**  
Publish Build Artifacts


 **Publish Artifact: backend-webapi.yaml**  
Publish Build Artifacts


Build for Multi-Container ASP.NET Core Application using Windows Containers


**Agent job 1** 


 Run on agent


---

 **Build and Push WebAPI Image**  
Docker

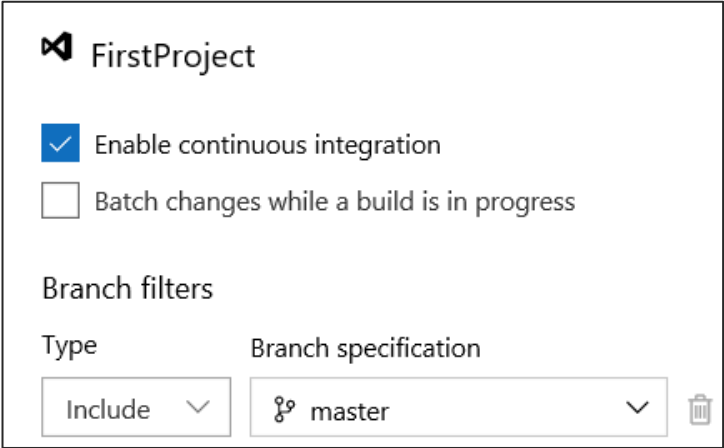
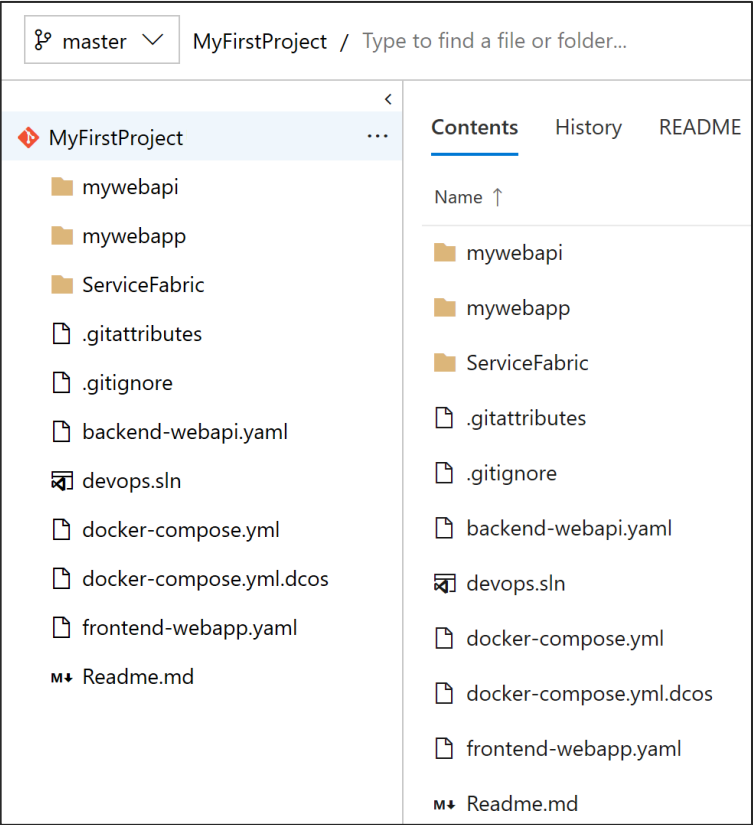
 **Build and Push WebApp Image**  
Docker

 **Replace build number in WebApi Service...**  
PowerShell

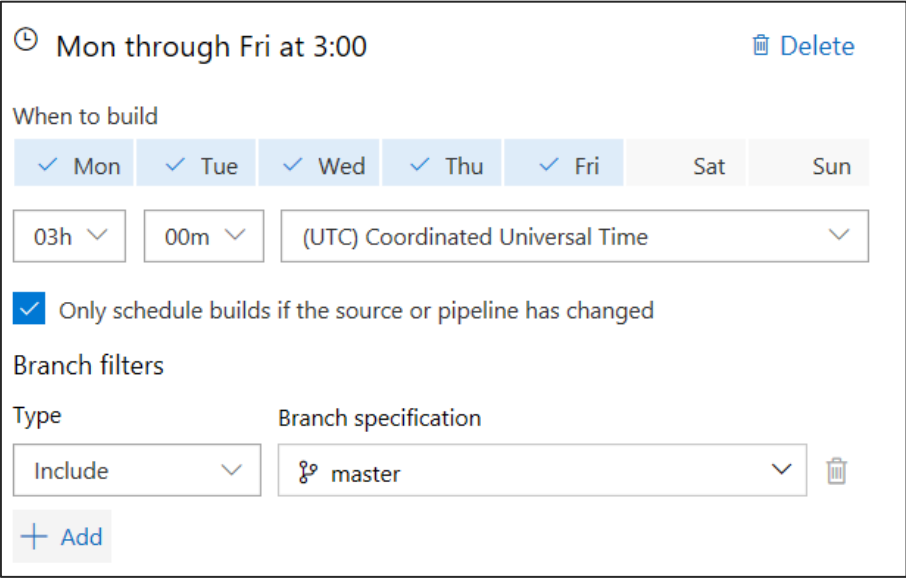
 **Replace build number in WebApp Servic...**  
PowerShell

 **Publish Artifact: drop**  
Publish Build Artifacts

# Continuous Integration Setting



Build → Triggers → CI Enabled



Schedule Build  
Example: Nightly/Daily

Changes to a branch can trigger the build automatically

# Demonstration: *Build Pipeline*


Build Pipeline for Multi-Container Application





# Release Pipeline

Release pipeline for AKS | Kubernetes

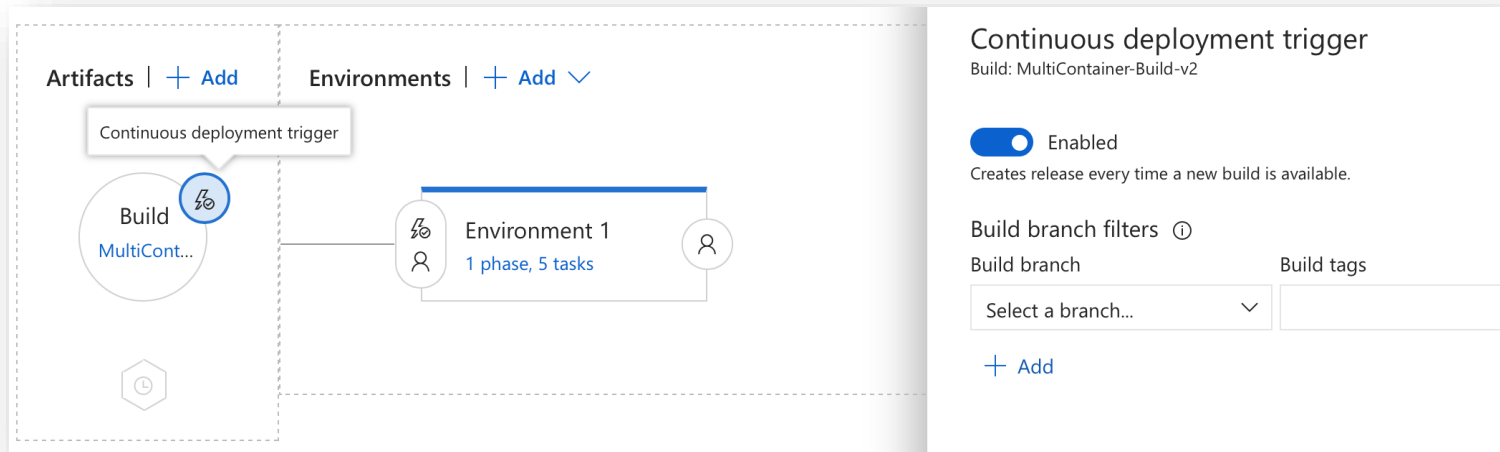
Agent job

 Run on agent

 **kubectl apply webapp**  
Deploy to Kubernetes

 **kubectl apply webapi**  
Deploy to Kubernetes

# Azure Pipeline Continuous Deployment Setting



Assign Build and select continuous deployment trigger

Continuous Deployment on every successful Build



On-Demand Release Kickoff



# Demonstration: *Release Pipeline*

Release Pipeline for AKS

Continuous Deployment  
Setting



# Lab: DevOps With Containers

