What is DevOps

Background & History

DevOps = Development & Operations

- **NOT** Development **VS** Operations
- **NOT** Development **OR** Operations

Concept went mainstream in 2009 (Velocity Conference)

- John Allspaw and Paul Hammond Presentation
- 10+ Deploys Per Day: Dev and Ops Cooperation at Flickr

Competing Goals

BUSINESS

Respond to the rapidly changing competitive landscape

Provide stable, reliable, and secure service to the customer

Happy Customers

DEVELOPMENT

New Products

Add new features

Security Updates

Bug Fixes

* Introduce Change

Happy Customers

OPERATIONS

Keep service stable and fast

- Maximize Uptime

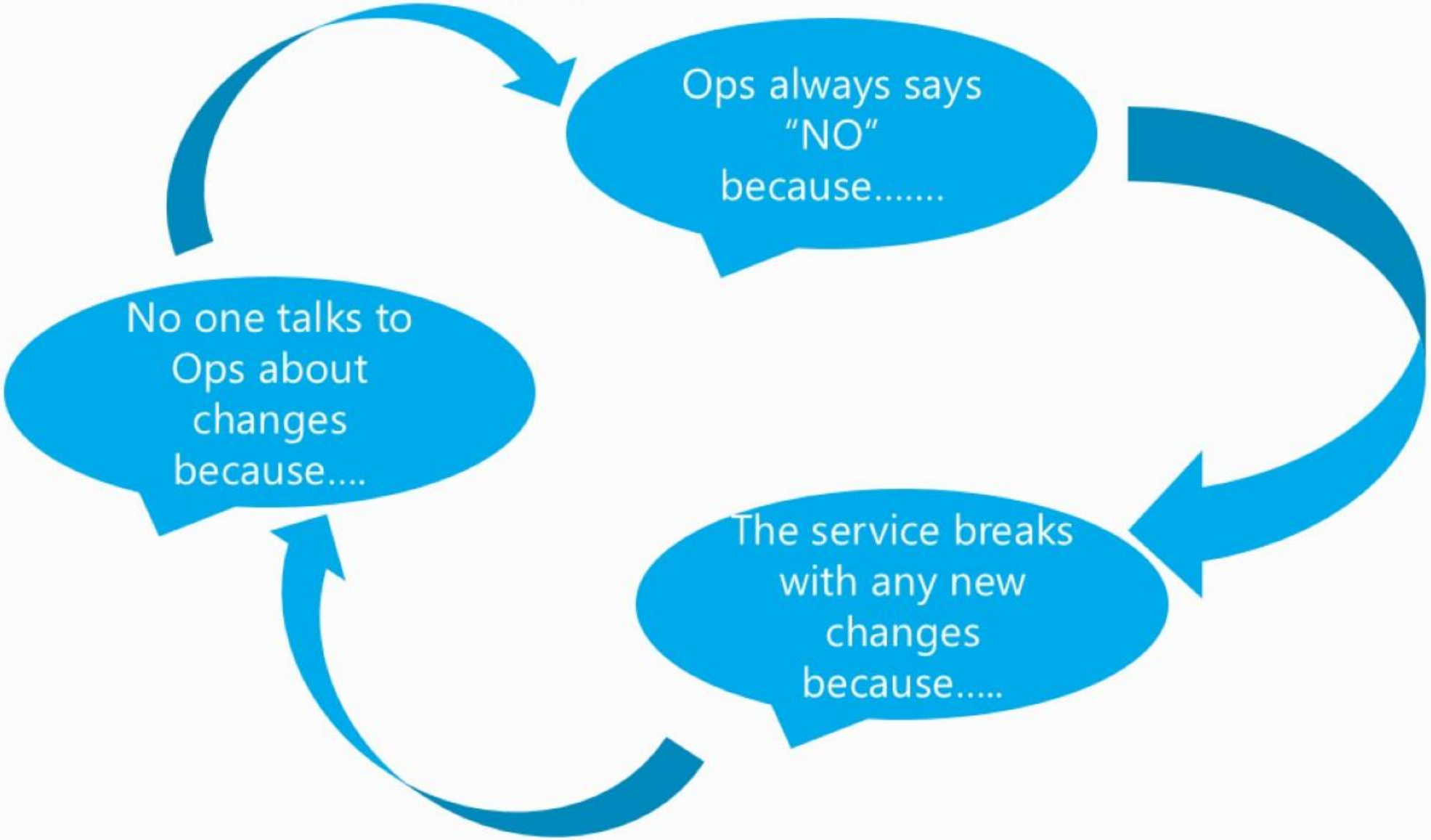
* Minimize Changes

Happy Customers

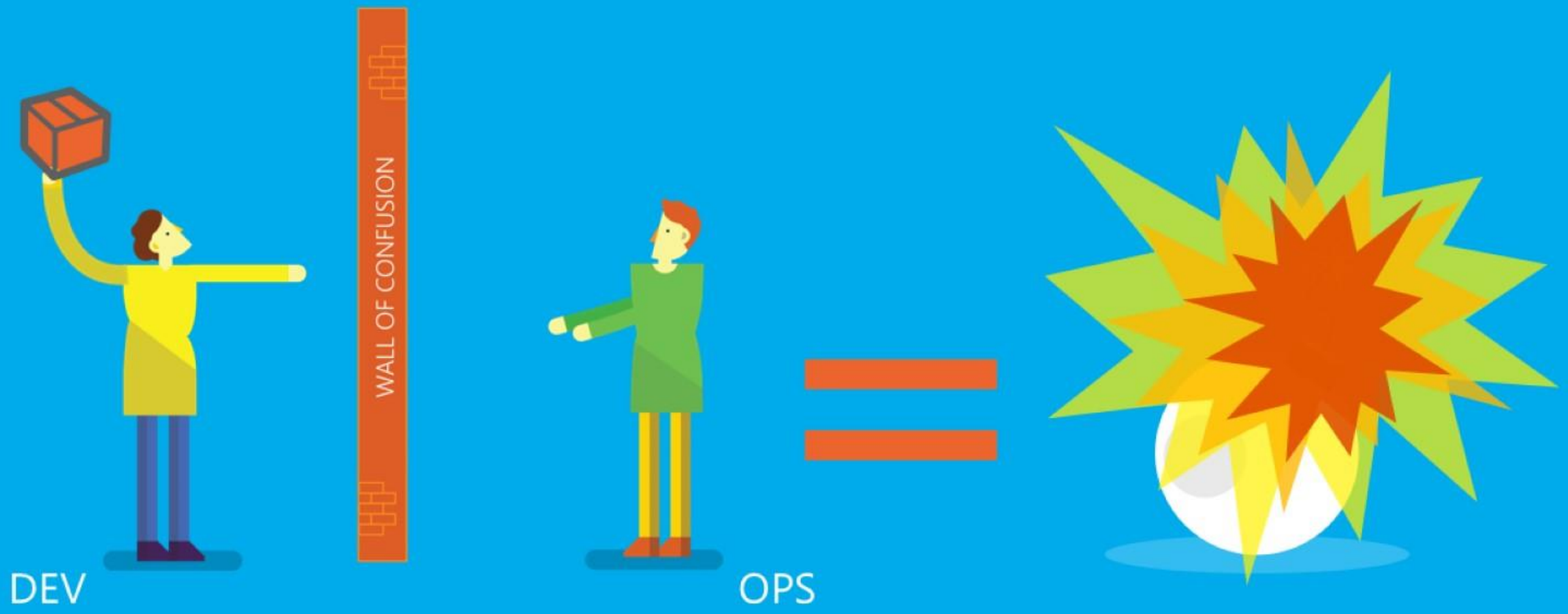
(*Historic*) Stereotypes

DEVELOPMENT	OPERATIONS
It's not my code, it's your machines!	It's not my machines, it's your code!
Ops guy = grumpy old man who always says no!	Assumes all code changes will wreak havoc
Finger-pointy	Finger-pointy

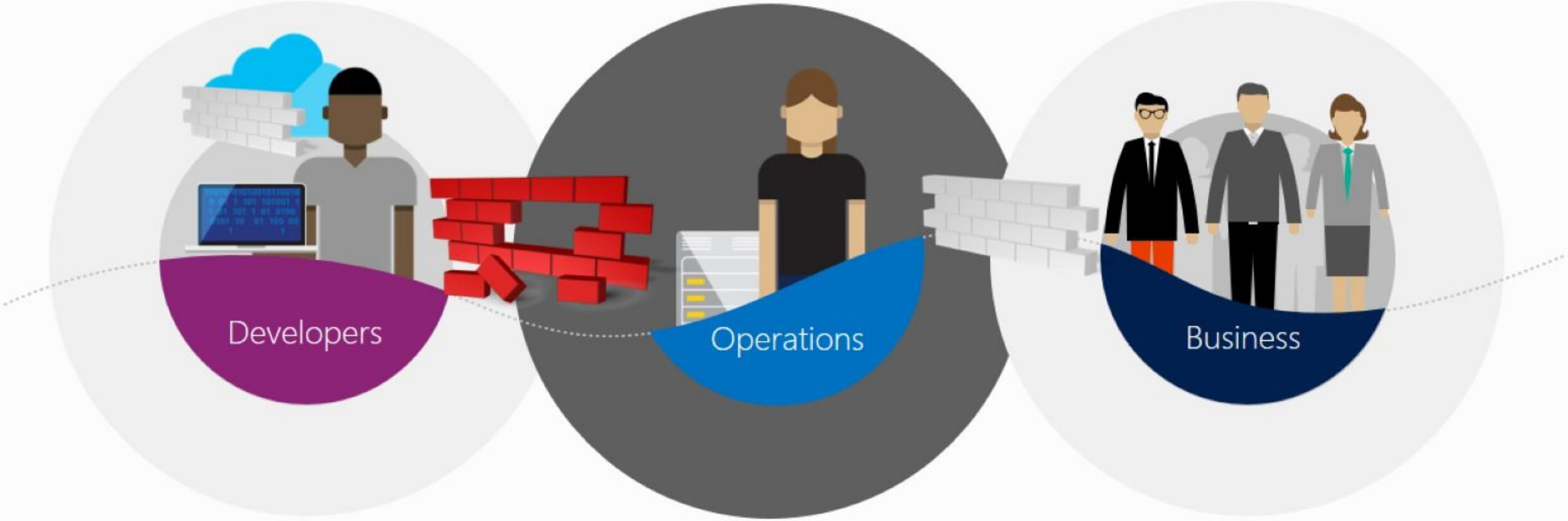
(*Historic*) Stereotypes



(*Historic*) Stereotypes



Consequences of inefficiency



Partial Agility

Downtime Costs

Rework

What DevOps is NOT *(common misconceptions)*

Tool or Set of Tools

Chef, Puppet, Jenkins, Azure ≠ DevOps

DevOps is only for Startups or 'Cloud' companies

DevOps is only for Open Source Software

DevOps replaces Agile

Ops resources need to learn how to code

DevOps is only for cloud deployments

It works for small deployments but ours is complex

What is DevOps

DevOps integrates development and operations teams in order to improve collaboration and productivity by:

- Automating Infrastructure
- Automating Workflows
- Continuously Measuring Service Performance

Core Components of DevOps

People

Process

Tools

Core Components of DevOps (People)

Trust

Don't just say no

Providing devs with access to environment (read-only)

Dev – talk to ops upfront – what are the impacts

Don't hide ideas, failures, features

Trust that everyone is doing their best for the business

Respect

Devs – remember someone will be woken up when something fails/code breaks

No finger pointing

Healthy attitude about failure

Expertise / background / ideas / suggestions

Communication

Don't hide ideas, failures, features

Don't just say no

Provide constructive feedback on current aches and pains

Core Components of DevOps (Process)

- Infrastructure as Code (IaC)
- Continuous Integration
- Automated Testing
- Continuous Deployment
- Release Management
- App Performance Monitoring
- Load Testing & Auto-Scale

- Availability Monitoring
- Change/Config Management
- Feature Flags
- Automated Environment De-Provisioning
- Self Service Environments
- Automated Recovery (Rollback & Roll-Forward)

http://www.itproguy.com/devops-practices/

Core Components of DevOps (Tools)

Automated Infrastructure

Shared version control

One step build

One step deploy

One step build and deploy

Who? When? What?

Tools

Benefits of DevOps

Shorten Cycle Times

Speed up and increase traceability of each release by empowering your development and operations teams with advanced collaboration and automation tools.

Optimize Resources

Efficiently manage environments using technologies that support self-service provisioning in a secure way, in line with your IT governance standards.

Improve Quality and Availability

Capture rich telemetry on application performance and usage so that you can make better decisions on future investments and anticipate issues in production before they impact your service.

DevOps Frame

100 level Overview

People, Process, Business Value, etc.

200 level Practices

Infrastructure as Code

Continuous Integration

300 level Product w/ Practice

Azure Deployment Template with Azure Website + SQL DB

Chef Recipe with Linux VM on Azure

Visual Studio Team Services (git) + Jenkins (build)

Github + Visual Studio Team Services + Gradle

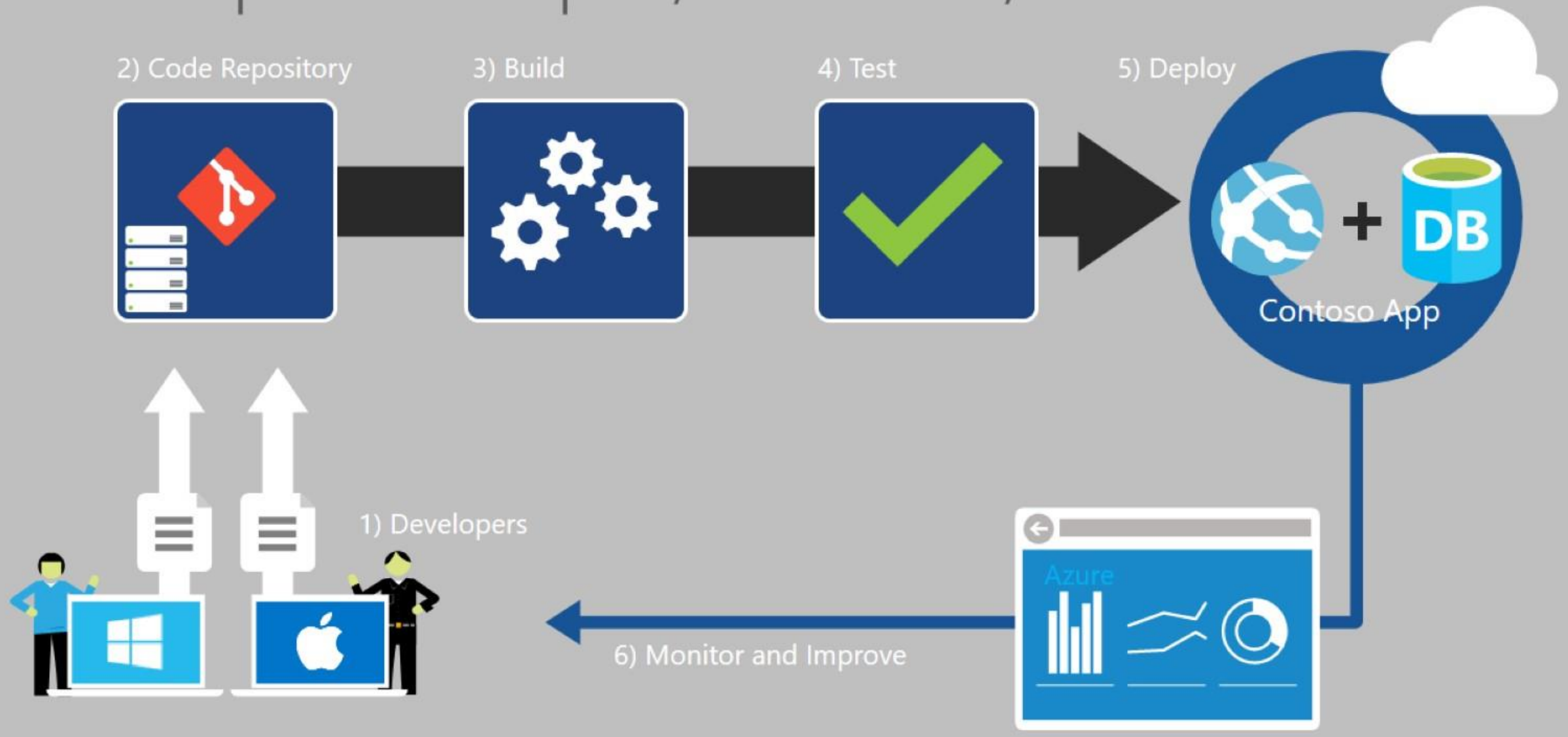
List of DevOps Practices

- Infrastructure as Code (IaC)
- Continuous Integration
- Automated Testing
- Continuous Deployment
- Release Management
- App Performance Monitoring
- Load Testing & Auto-Scale

- Availability Monitoring
- Change/Configuration Management
- Feature Flags
- Automated Environment De-Provisioning
- Self Service Environments
- Automated Recovery (Rollback & Roll-Forward)
- Hypothesis Driven Development
  - Testing in Production
  - Fault Injection
  - Usage Monitoring/User Telemetry

http://www.itproguy.com/devops-practices/

DevOps – People, Process, Tools



What is DevOps?

People. Process. Technology.

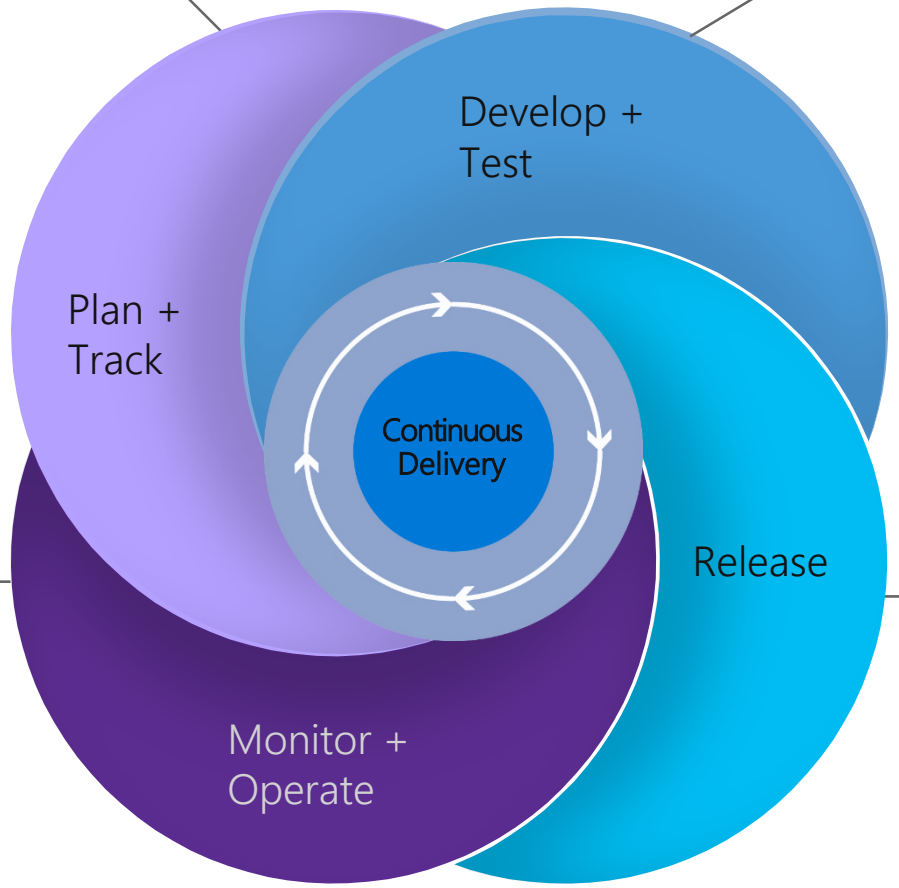
DevOps is the union of people, process, and technology to <u>enable continuous delivery of value to your end users</u>.

Plan & Track

Agile Planning
Dashboards
Kanban Boards
Task boards

Build & Test

Git Source Control
Modern Code Workflow
Continuous Integration
Continuous Testing
Package Management
Open Source Compliance

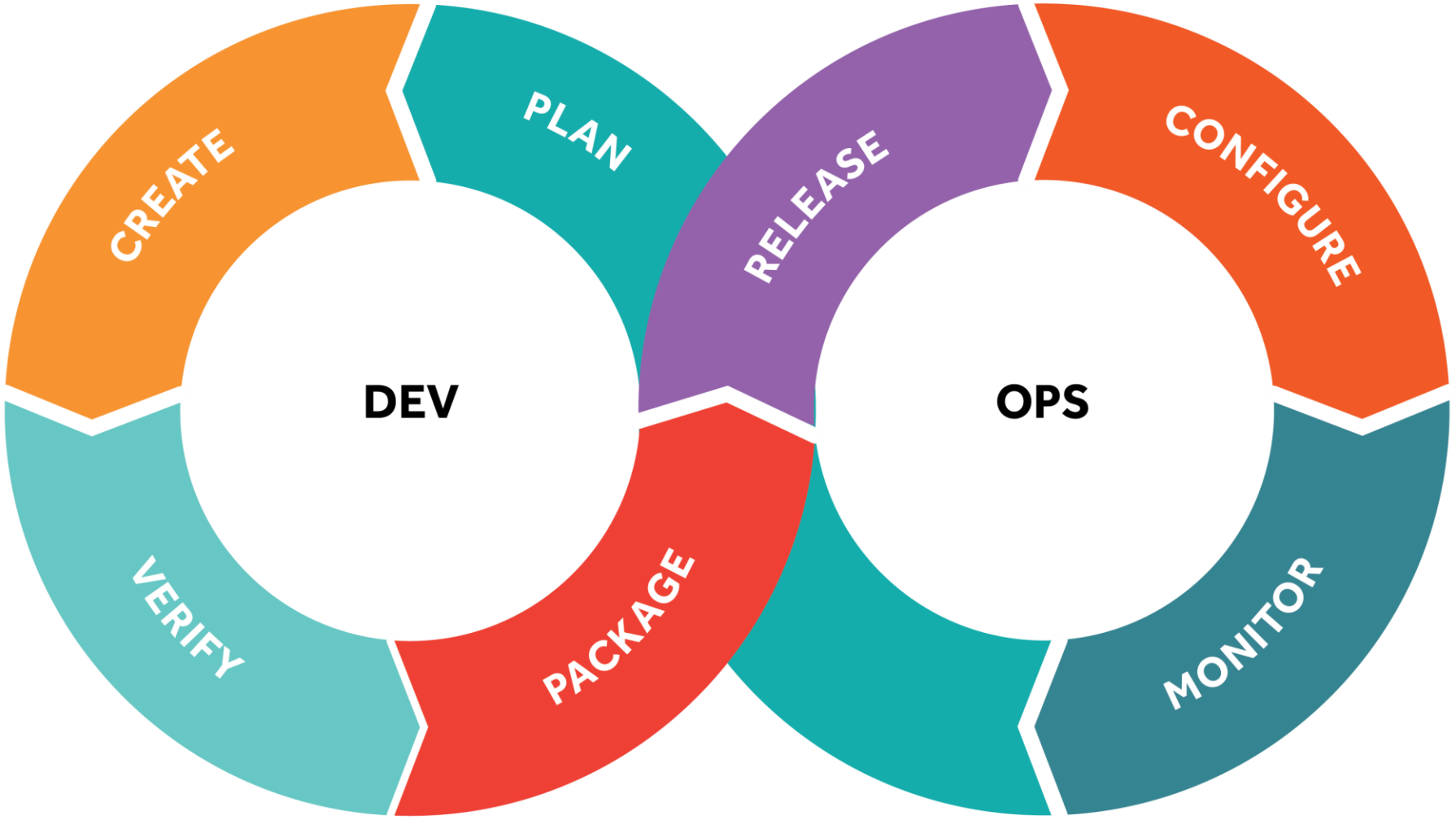


Monitoring & Operations

Telemetry
Diagnostics
Analysis
Compliance

Deployment

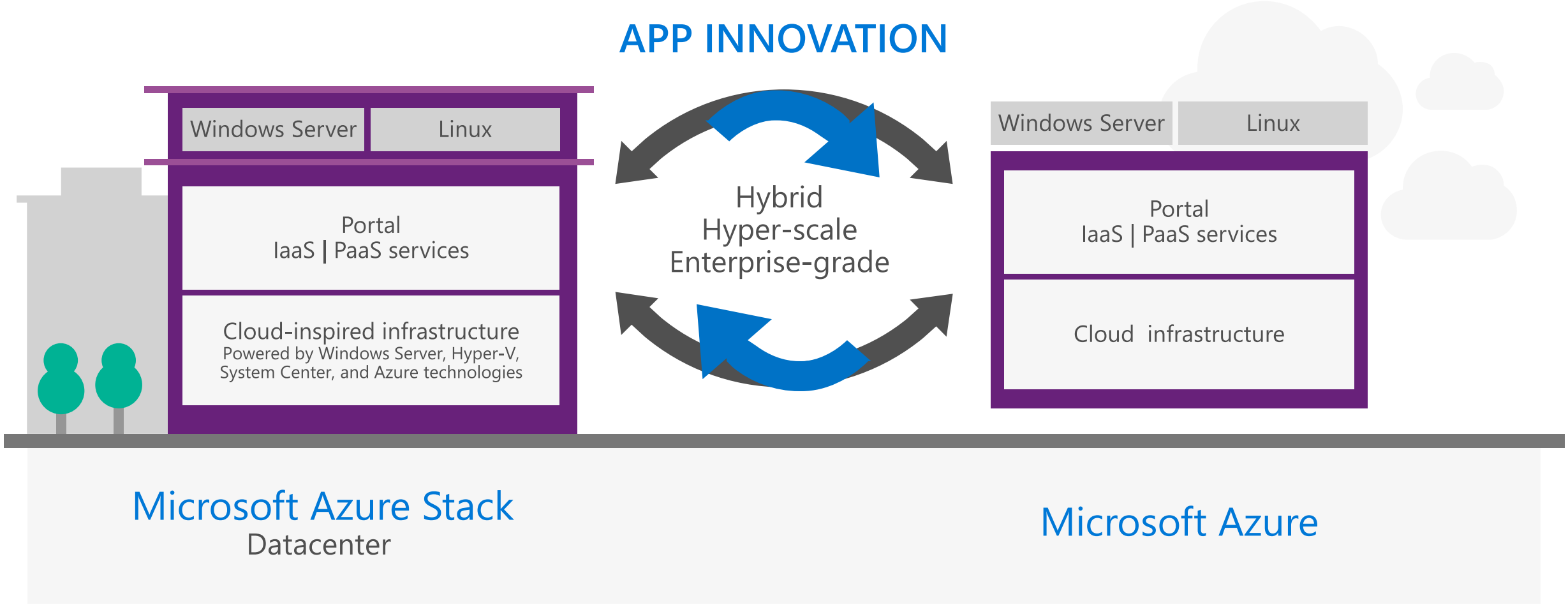
Deployment of app and infrastructure
PaaS, IaaS and Containers

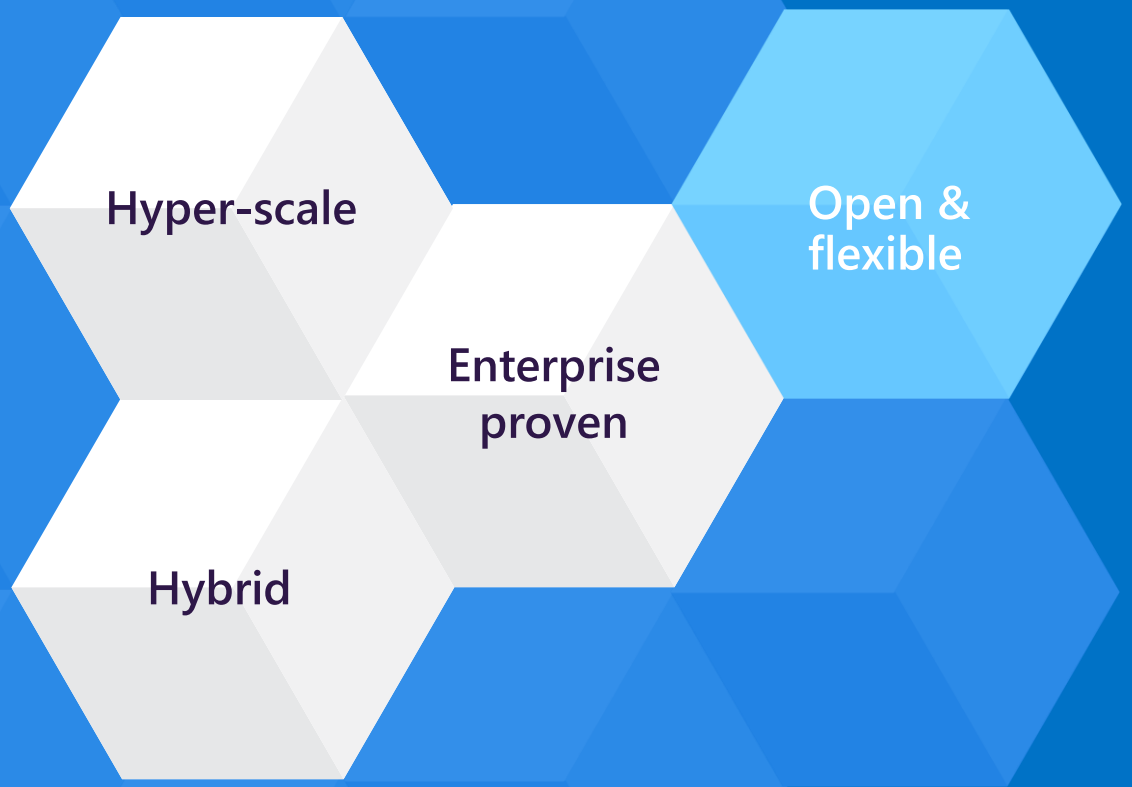

CREATE
PLAN
RELEASE
CONFIGURE
DEV
OPS
VERIFY
PACKAGE
MONITOR

Tools / Products

Power of Azure with the control of the datacenter

Introducing the Microsoft Azure Stack





Open + Flexible

Management

Applications

App Frameworks

Databases & Middleware

Infrastructure

Linux

Visual Studio
Team Services

Team Foundation
Server

Enables automation and Agile & DevOps practices

Open and extensible, supports your tools and technologies

Flexibility with on-premises or cloud service

http://www.visualstudio.com/news/release-archive-vso

Visual Studio Partners and Extensions

65
Visual Studio Code Extensions

5,910
Visual Studio Gallery Extensions

90
Visual Studio Sim-Ship Partners

48
VS Team Services Extensions

Introducing Azure DevOps

Azure Boards

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

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 Repos

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

Azure Test Plans

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

Azure Artifacts

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

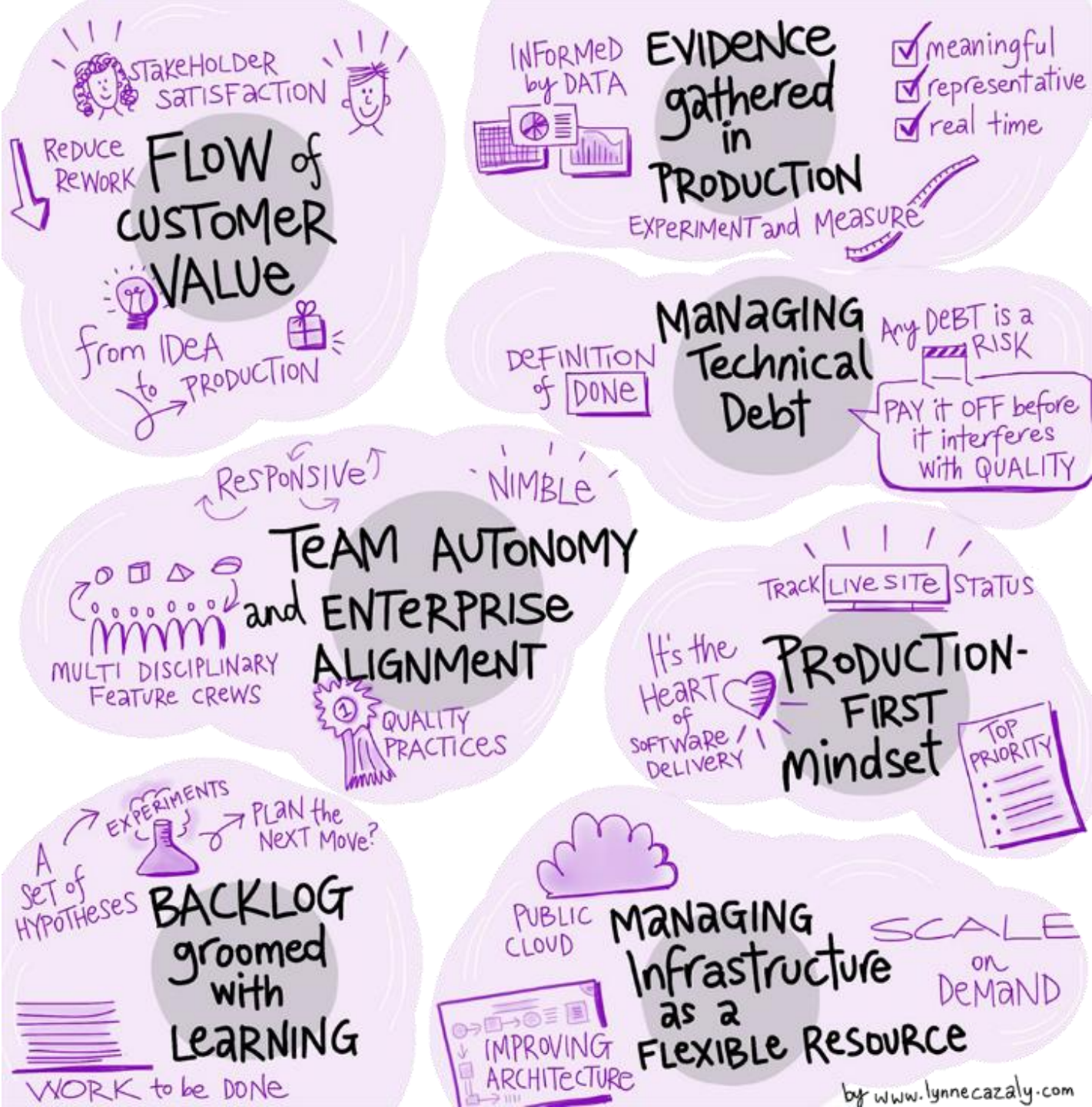
https://azure.com/devops

DevOps Maturity Assessment

OUR 7 HABITS FOR DEVOPS SUCCESS

*Based on our experiences from our own transformation to a DevOps culture, we have identified **DevOps habits** and related **DevOps practices** that will lead to a true shift to DevOps in organizations of any size.*

http://DevOpsAssessment.azurewebsites.net/

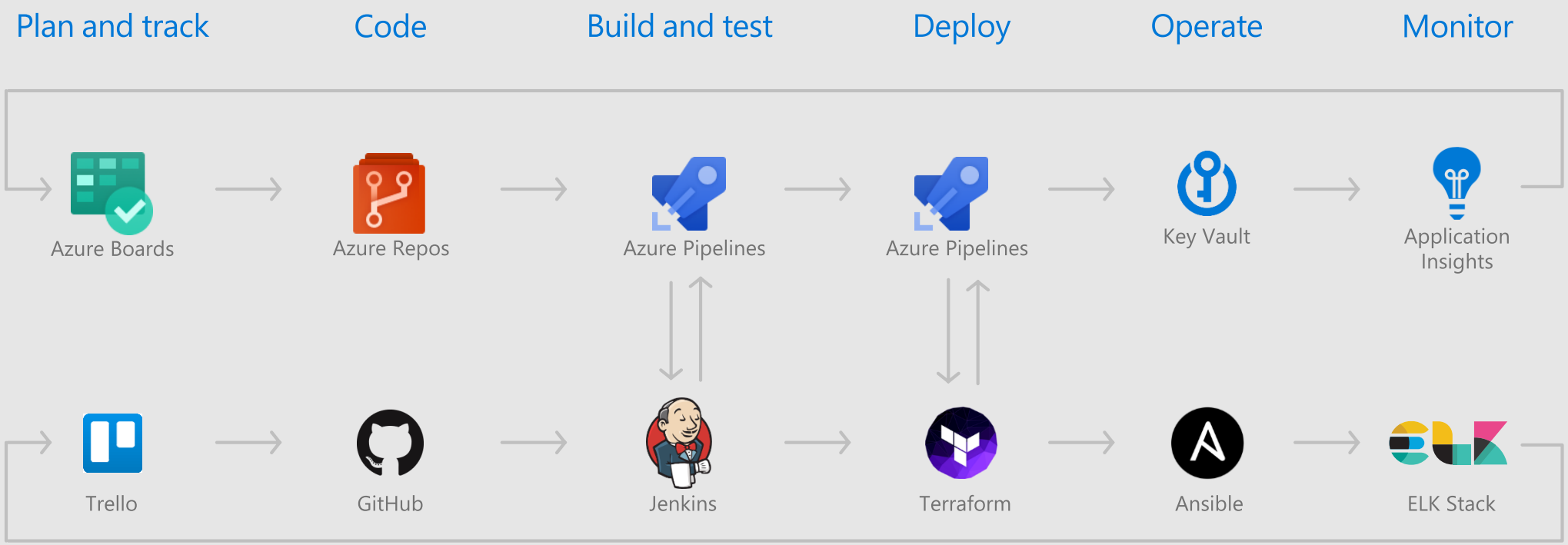


Microsoft's DevOps Journey

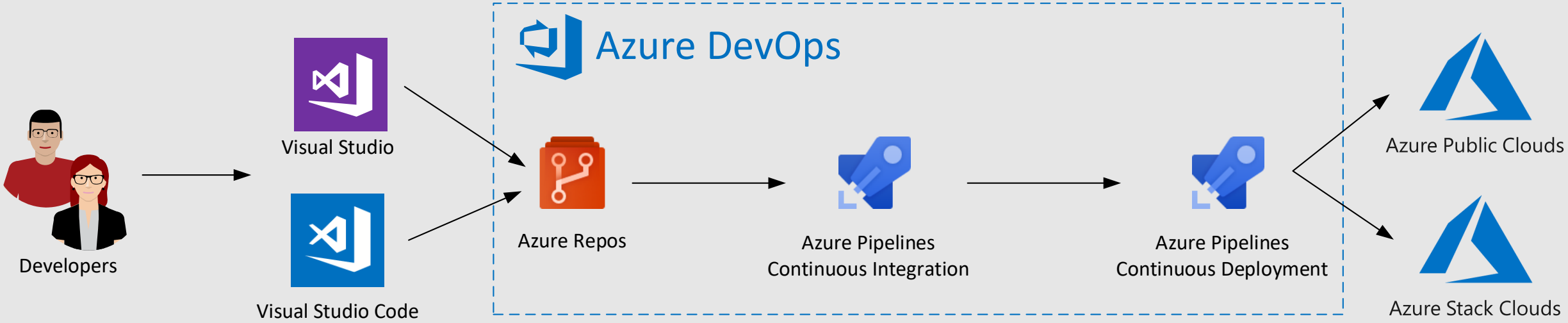
http://aka.ms/OurDevOpsJourney

http://www.itproguy.com/how-microsoft-does-agile-devops-resources/

Azure DevOps framework



Hybrid Pipeline with Azure and Azure Stack



Azure Pipelines

Cloud-hosted pipelines for Linux, Windows and macOS, with unlimited minutes for open source

Any language, any platform, any cloud

Build, test, and deploy Node.js, Python, Java, PHP, Ruby, C/C++, .NET, Android, and iOS apps. Run in parallel on Linux, macOS, and Windows. Deploy to Azure, AWS, GCP or on-premises

Extensible

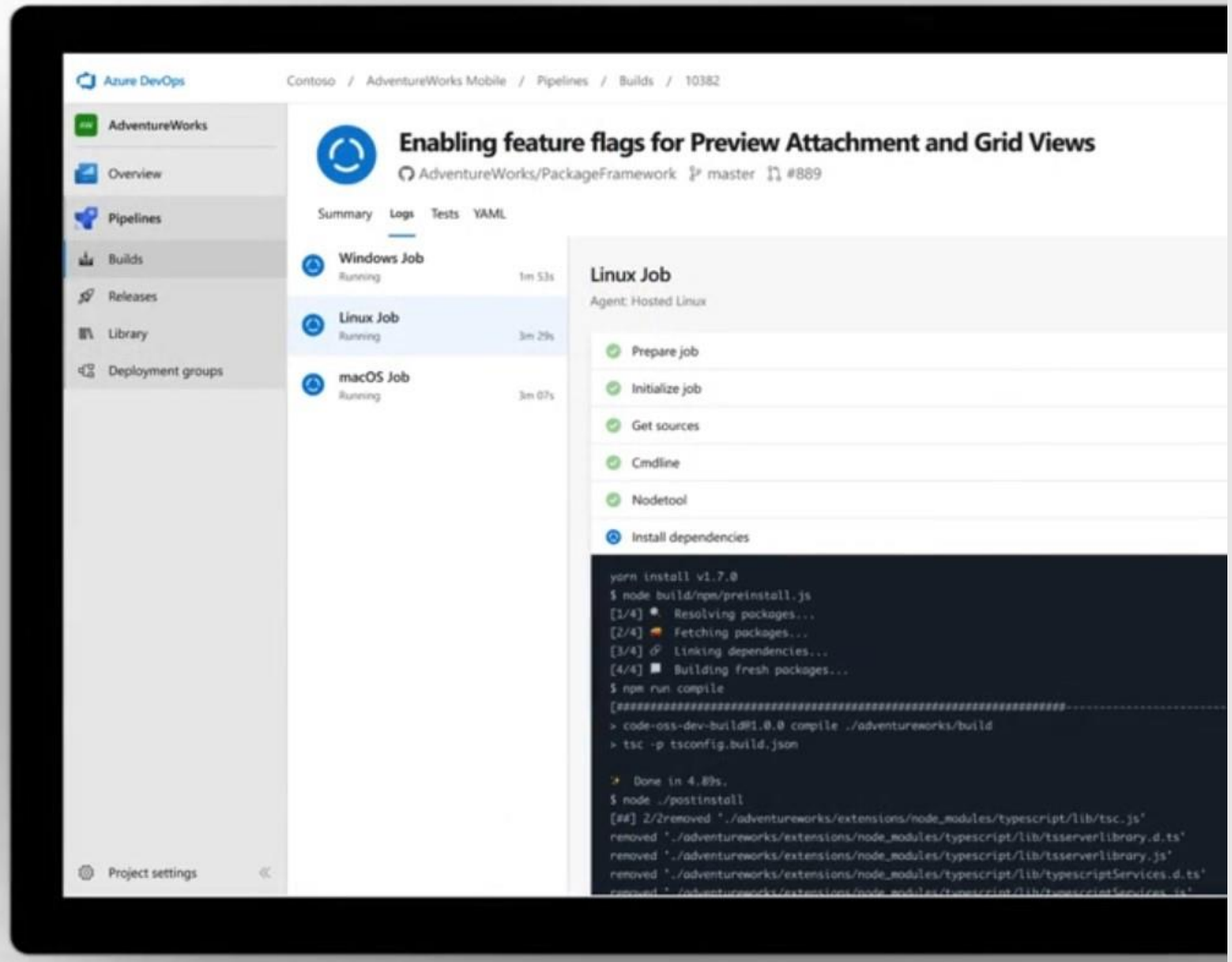
Explore and implement a wide range of community-built build, test, and deployment tasks, along with hundreds of extensions from Slack to SonarCloud. Support for YAML, reporting and more

Containers and Kubernetes

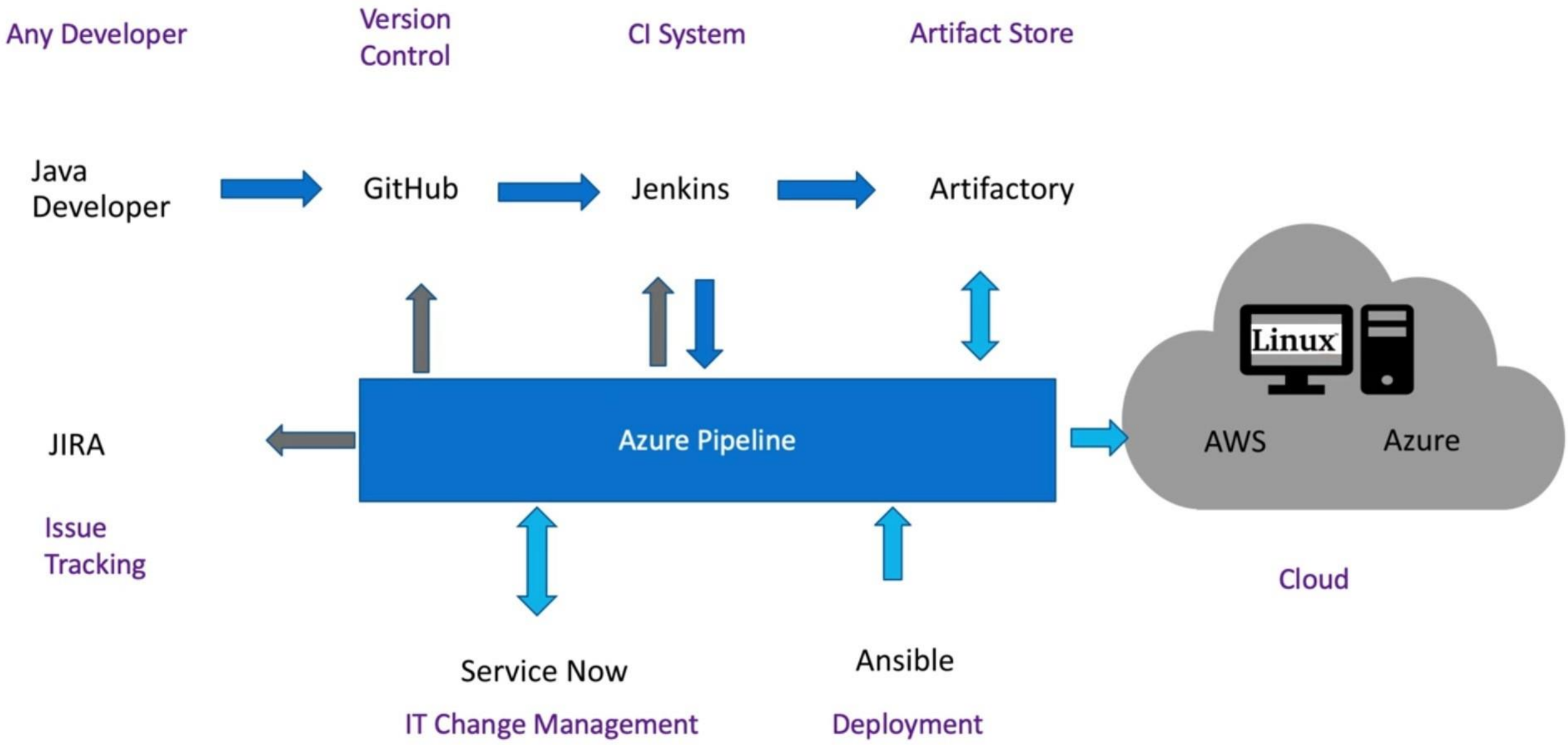
Easily build and push images to container registries like Docker Hub and Azure Container Registry. Deploy containers to individual hosts or Kubernetes.

Best-in-class for open source

Ensure fast continuous integration/continuous delivery (CI/CD) pipelines for every open source project. Get unlimited build minutes for all open source projects with up to 10 free parallel jobs across Linux, macOS and Windows



https://azure.com/pipelines


Any Developer
Version Control
CI System
Artifact Store
Java Developer
GitHub
Jenkins
Artifactory
Linux
JIRA
Azure Pipeline
AWS
Azure
Issue Tracking
Cloud
Service Now
Ansible
IT Change Management
Deployment

Azure Tools

Develop

- Visual Studio Tools
- Visual Studio Cloud Explorer

Deploy and Manage

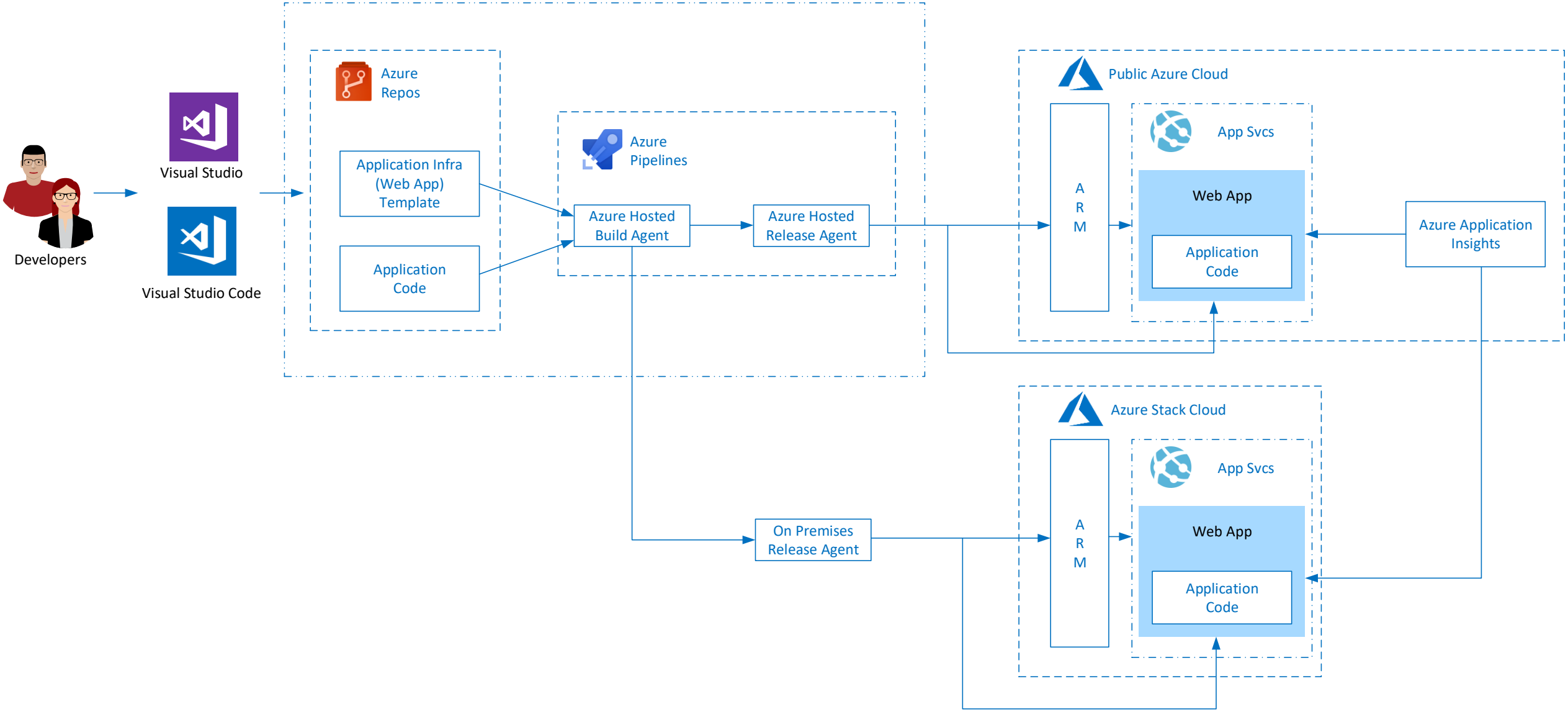
- Azure CLI
- Azure SDKs (.Net, Java etc…)
- Azure PowerShell
- Azure Storage Explorer

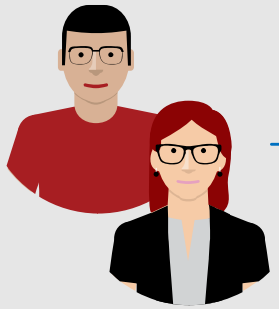
Monitor

- Log Analytics
- Application Insights

Consistent Tooling across clouds

Hybrid DevOps Pipeline - Architecture





Developers

Visual Studio

Visual Studio Code

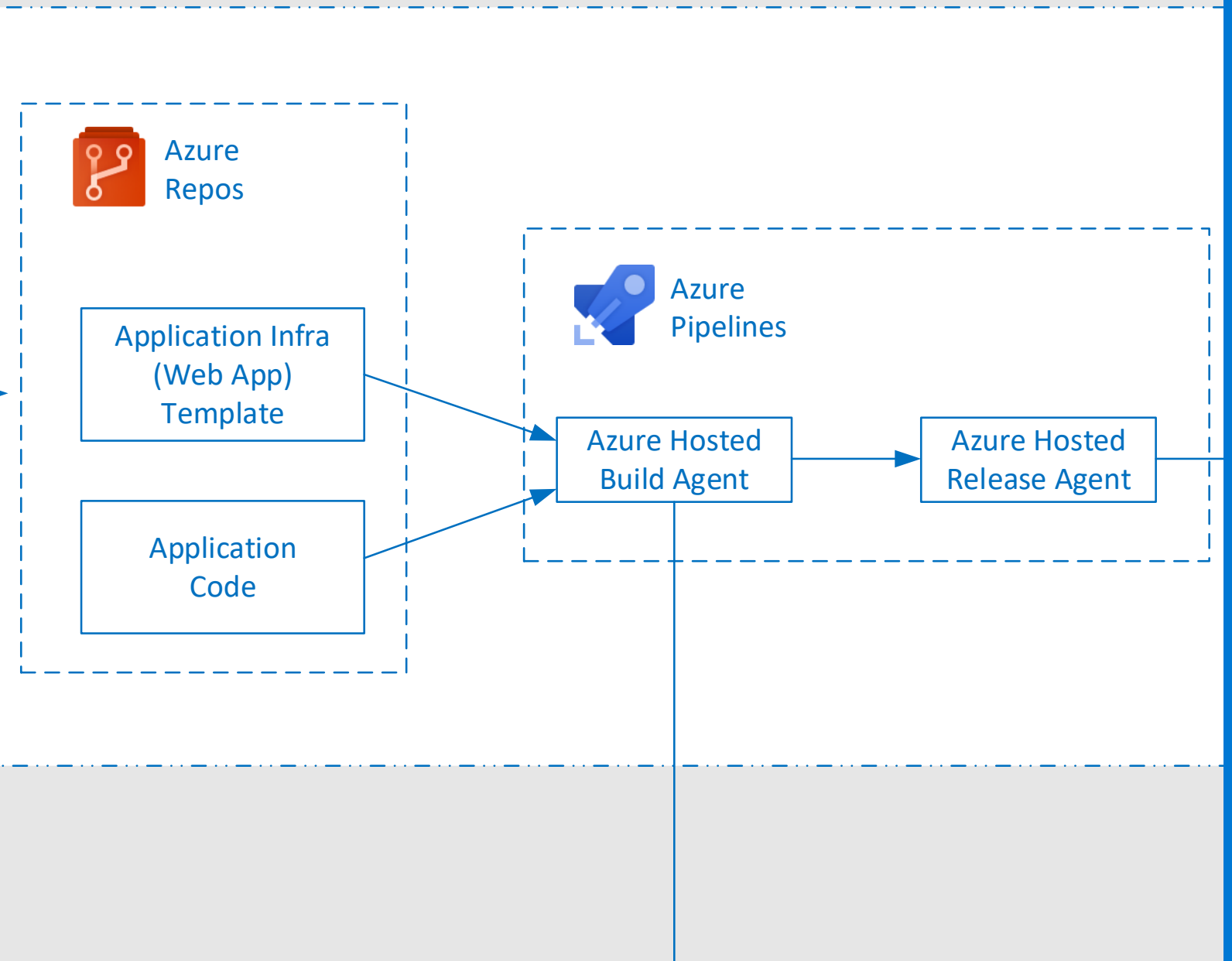
Azure Repos

Application Infra (Web App) Template

Application Code

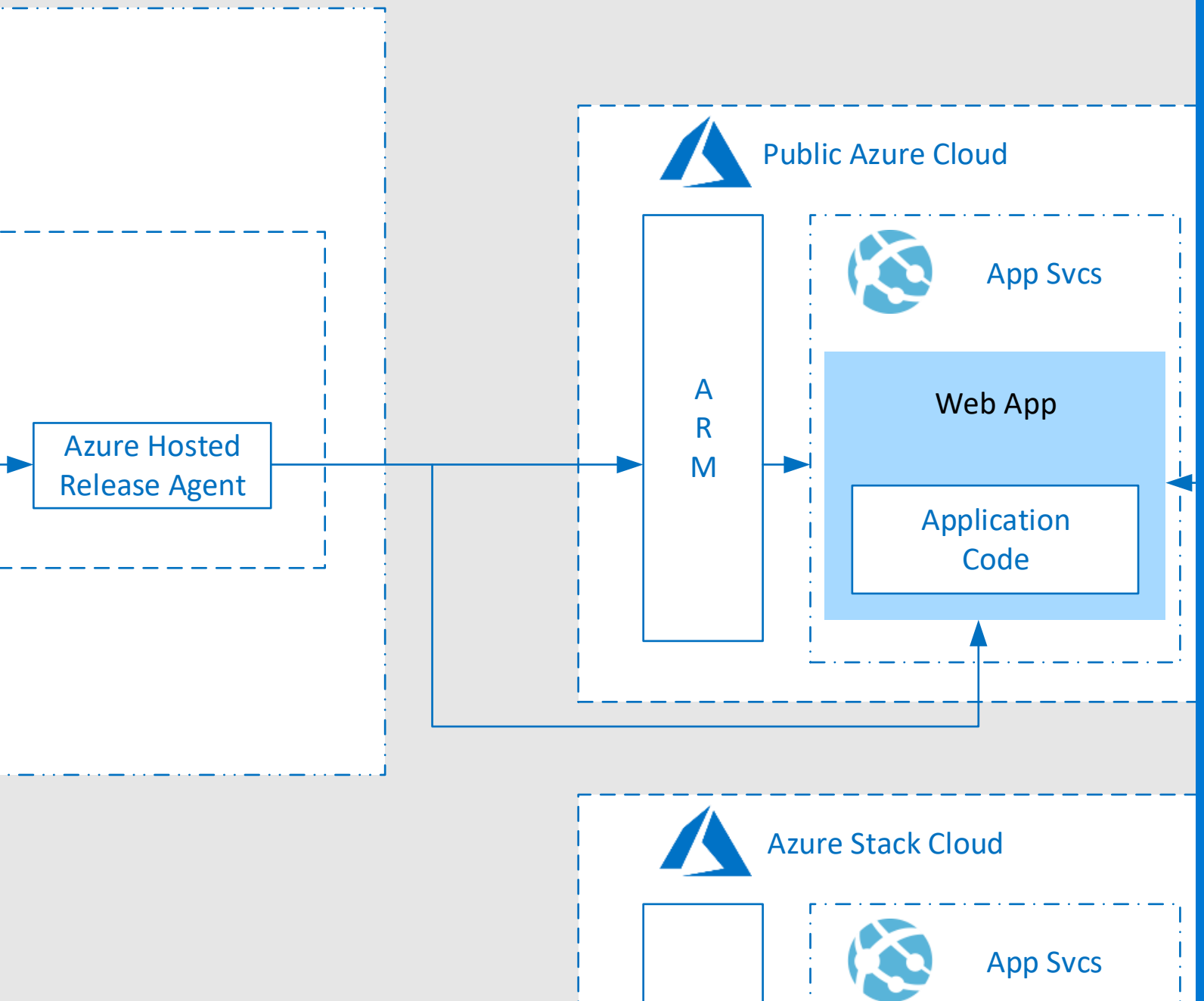
Code

- Use Visual Studio or Visual Studio Code
- Synchronize Repositories to Dev env
- Define Infra for app using ARM templates
- Maintain code for Application separately



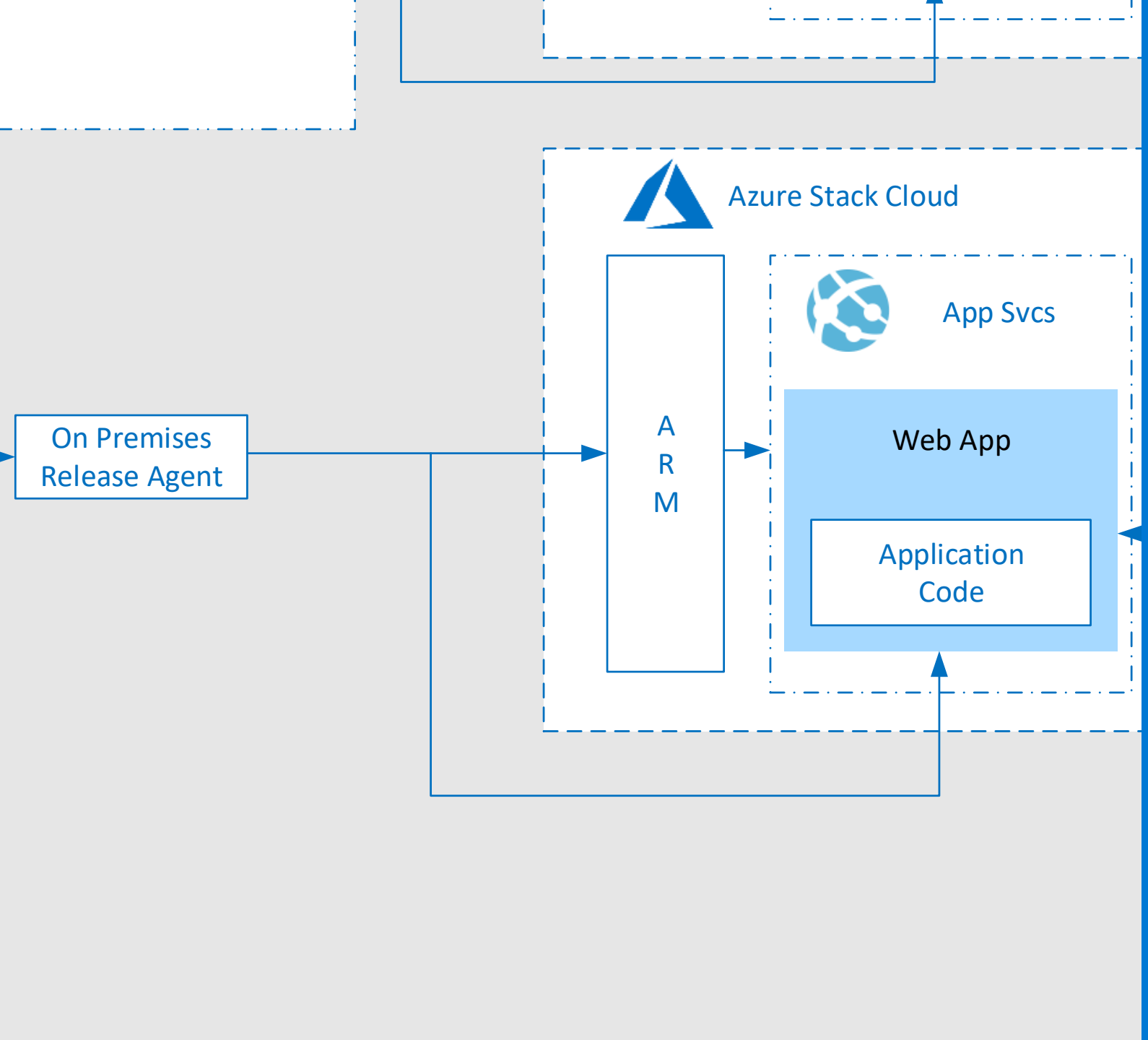
Build

- Build Agent hosted in Azure
- Set up for Continuous Integration



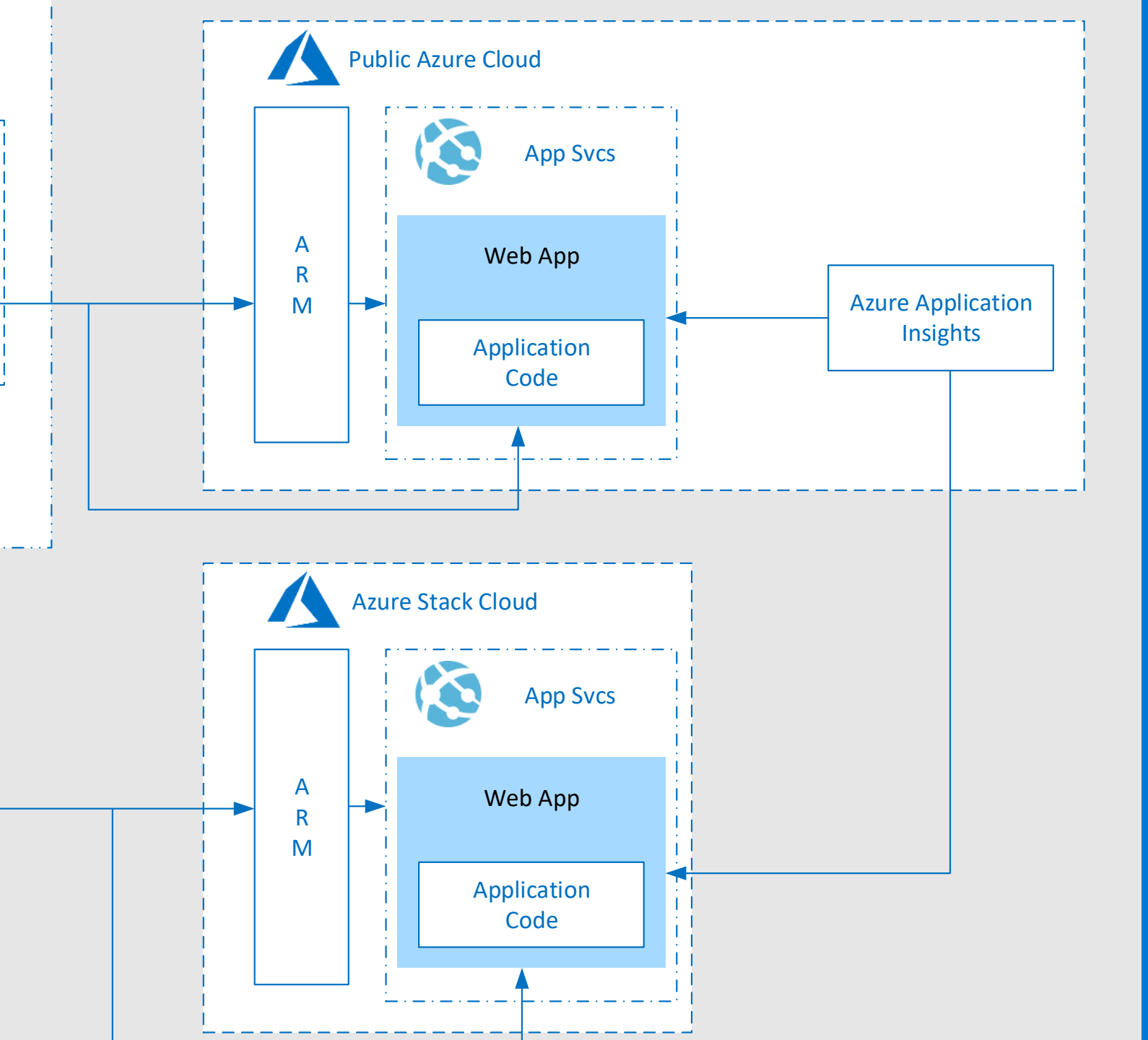
Release to Azure

- Create/Update Web App Infra on App Svcs through ARM template on Azure
- Publish code to Web App



Release to Azure Stack

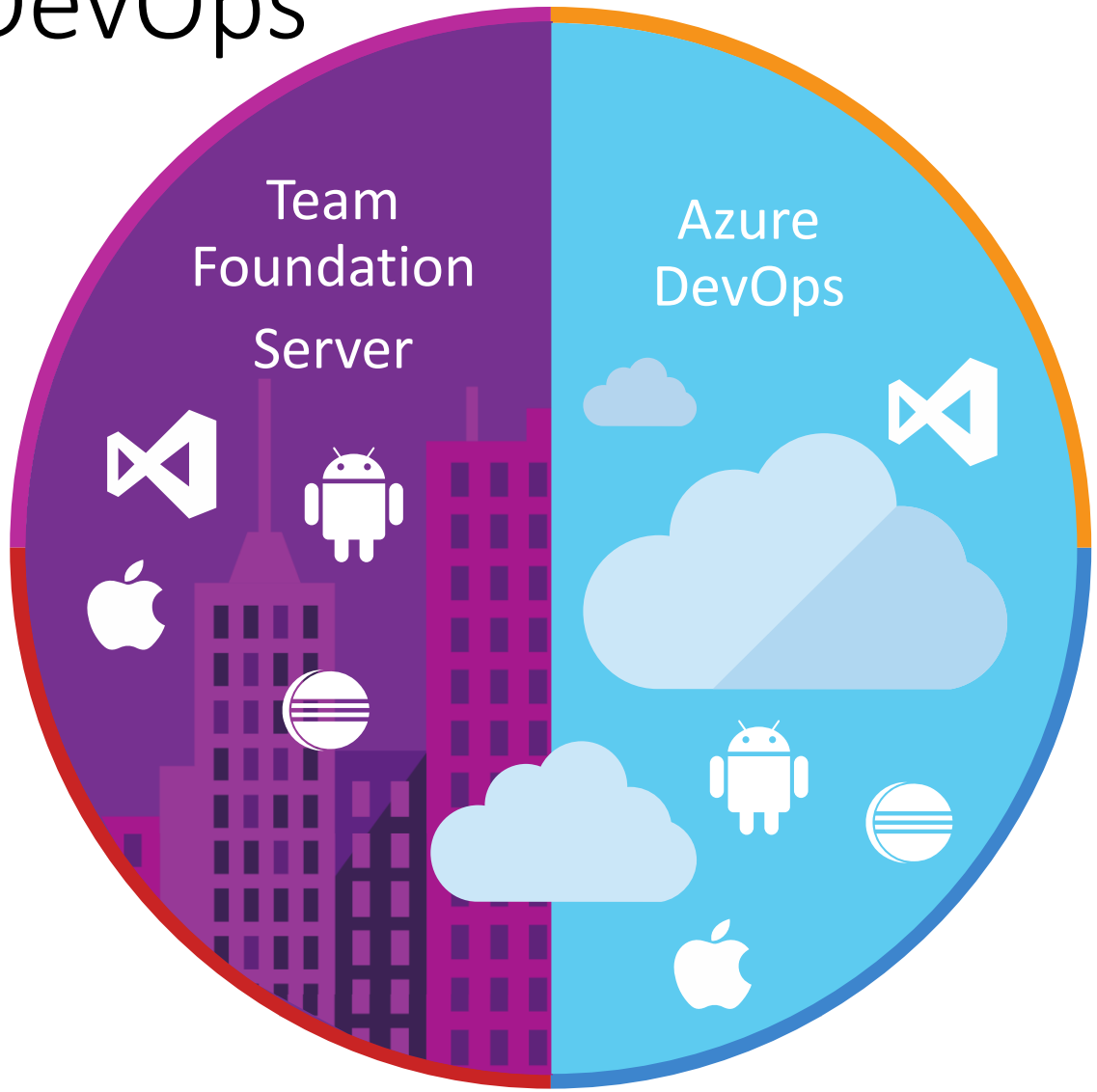
- Release Agent hosted on-prem
- Release Agent has access to Azure Stack
- Create/Update Web App Infra on App Svcs through ARM template on Azure Stack
- Publish code to Web App



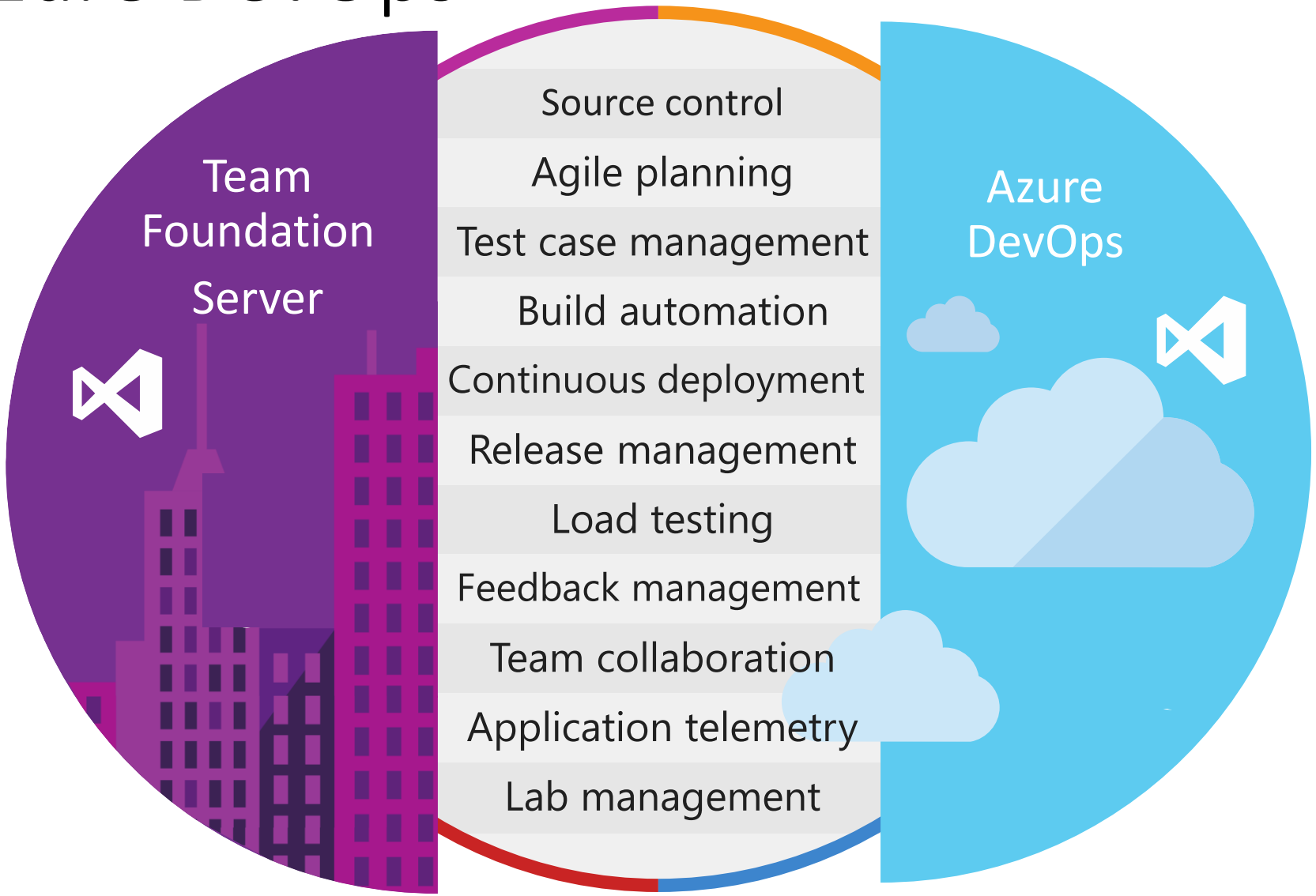
Monitor Applications

- Use Application Insights in Azure to monitor resources in Azure and Azure Stack

TFS vs Azure DevOps



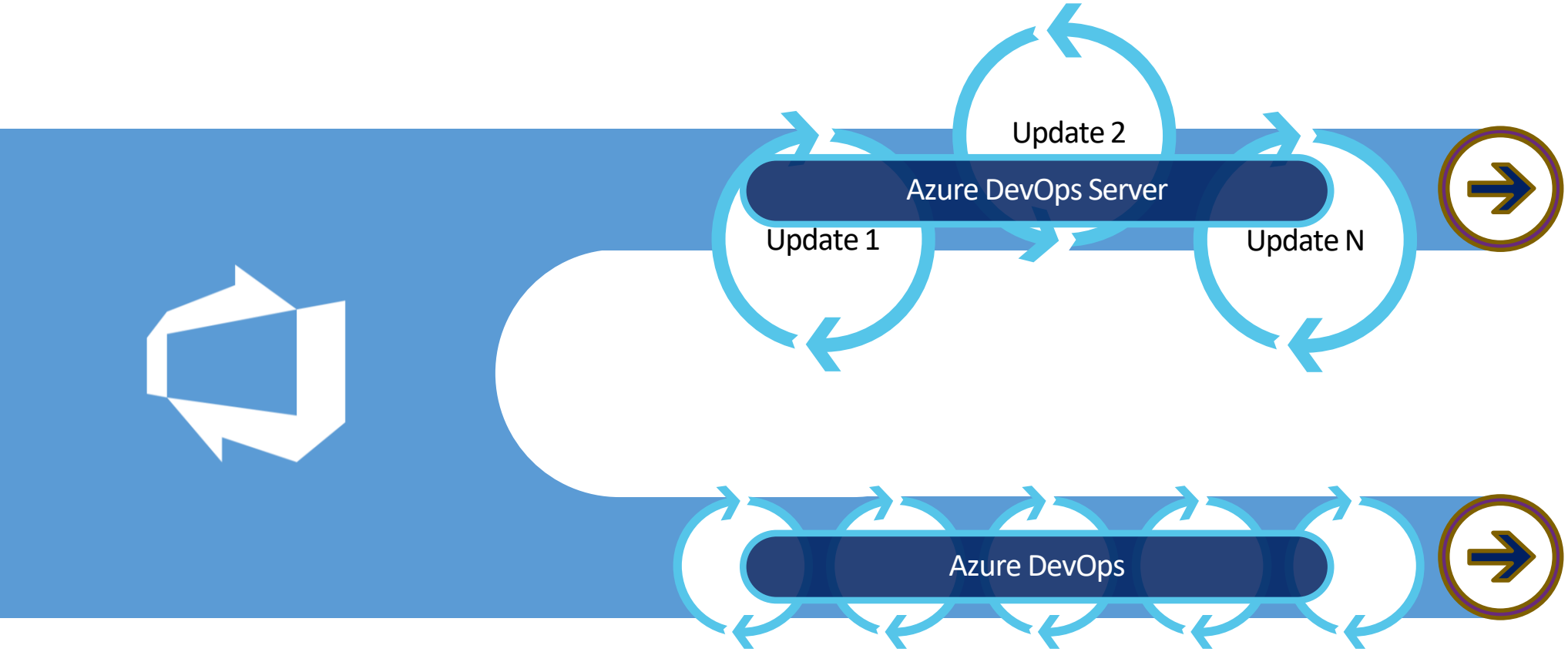
TFS vs Azure DevOps



Code - Cloud first, then move on-premises

One code base with multiple delivery streams

- ✓ Shared abstraction layer, 90% of code is the same
- ✓ Single master branch, multiple release branches
- ✓ Teams uses a 3-week sprint and release cadence



Changes for existing VSTS / TFS customers

The same functionality you know and love today, with greater openness, flexibility and focus

Existing Accounts

- Existing **https://contoso.visualstudio.com** URL continues to work. **https://dev.azure.com/contoso** available for opt-in.
- New **UI opt-in** per user as preview feature. Will start advertising new UI once feedback from new accounts and early adopters has been incorporated.
- Can disable services on a per project basis for new UI
- New Azure branding in communications and documentation (emails, alerts etc).
- Websites and documentation will move from Visual Studio to Azure based locations (with redirects in place).
- Redirects available for some time.
- TFS will remain the on-premises brand until the next major version in 2019. The new UI will be enabled in that release.
- Existing TFS branded information and downloads remain in Visual Studio locations until next release.

Pricing

- **Public project usage is now free.**
- The free tier for Pipelines now includes **1,800** minutes per month, up from 240.
- Pipelines can be used independently from Repos — so if you are only using Pipelines and your repos are hosted on GitHub you don't need to pay for Repos or Boards (Basic) users.

New Accounts

- https://dev.azure.com/contoso based URL.
- New navigation & branding by default.

Migrating from TFS to Azure DevOps

Move from Team Foundation Server to Azure DevOps and bring your data along

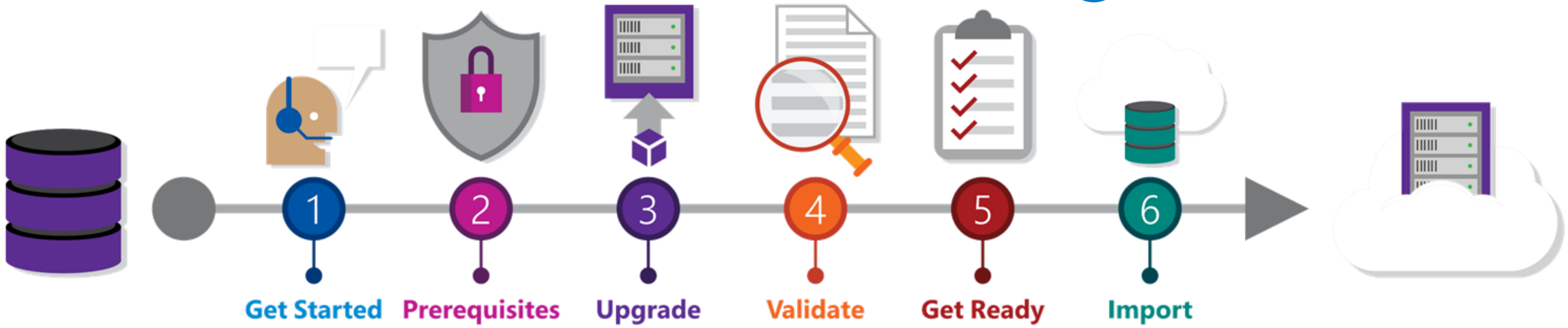
Benefits of Cloud Hosted Azure DevOps Services

- Global availability
- Hosted and maintained by Microsoft with 99.9% uptime guarantee and 24x7 support
- Immediate access to latest features
- Simplified deployment to Azure

TFS Import Service

- Fully supported high fidelity migration path
- Trusted by many large enterprises
- Now faster and easier to use

https://aka.ms/tfsimport



Import Service Requirements

Azure Active Directory

Must be using Azure Active Directory (AAD)

Setup Azure AD Connect to map identities from Active Directory (AD) to AAD

Support for the latest two TFS versions

Two latest versions of TFS are supported for import – includes updates, but not RC releases

Support window is ~6 months per TFS releases

Currently supporting TFS 2018 Update 2 and TFS 2018 Update 3

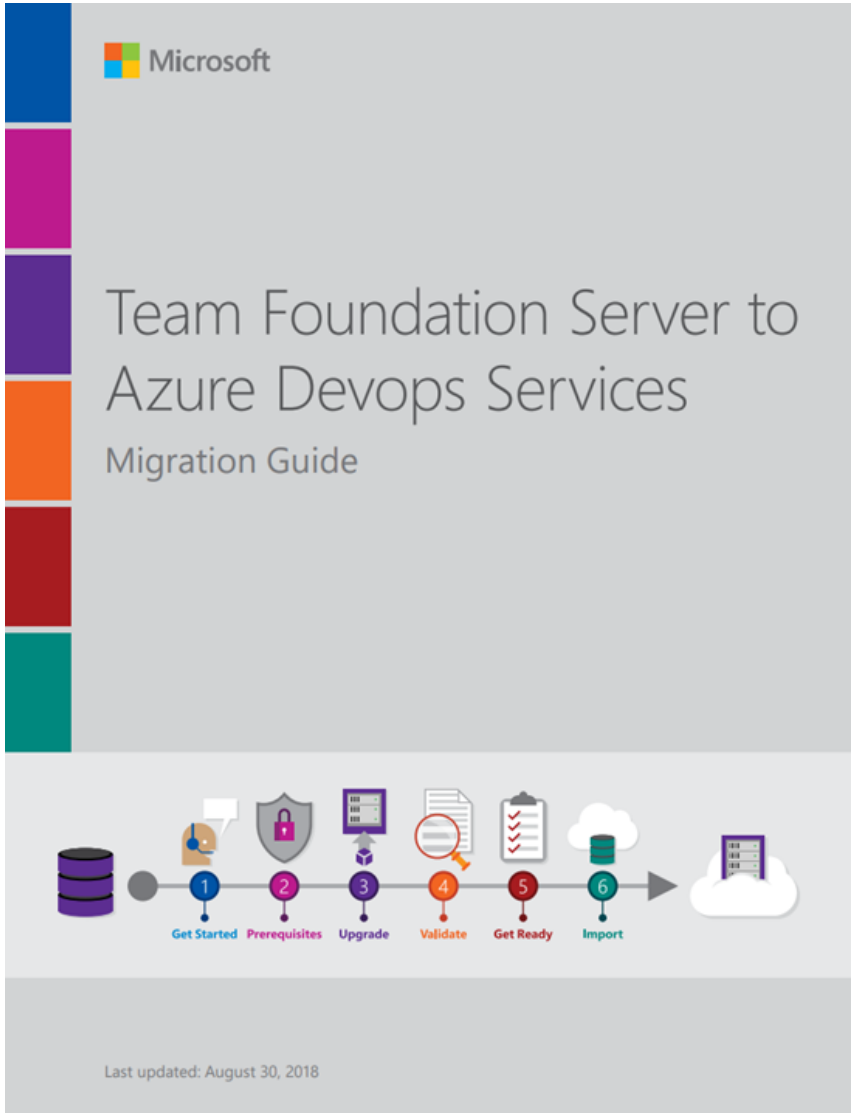
Migrate to Azure DevOps Services

Migration guide & Documentation

Detailed guidelines & documentation can be found at https://aka.ms/tfsimport

Questions? Contact Us!

AzureDevOpsImport@microsoft.com



Azure DevOps Services Pricing

Open Source Projects

Free

Unlimited users and build time

- **Azure Pipelines:** 10 parallel jobs with unlimited minutes for CI/CD
- **Azure Boards:** Work item tracking and Kanban boards
- **Azure Repos:** Unlimited public Git repos

Small Teams

Free

Start free with up to 5 users

- **Azure Pipelines:** Run 1 Microsoft-hosted job for 1,800 minutes per month and 1 self-hosted job for any amount of time
- **Azure Boards:** Work item tracking and Kanban boards
- **Azure Repos:** Unlimited public Git repos
- **Azure Artifacts**: package management
- Unlimited stakeholders

Teams of any size

Starts at $6

per user, per month for Boards & Repos*

Easy pricing that grows with your team

- **Azure Pipelines:** Run 1 Microsoft-hosted job for 1,800 minutes per month and 1 self-hosted job for any amount of time
- **Azure Boards:** Work item tracking and Kanban boards
- **Azure Repos:** Unlimited public Git repos
- **Azure Artifacts**: package management
- Unlimited stakeholders
- Boards & Repos included for Visual Studio subscribers

https://azure.com/pricing/details/devops/

* 5 Boards & Repos users and 5 Artifacts users free. Pipelines with unlimited minutes, Test Plans users and additional Artifacts users also available. Please see the Azure pricing calculator for details.

Manage the full stack

Monitor/Learn	Telemetry	Application Insights
	Diagnostics	Application Insights
	Performance	Application Insights
Approve	Notify Approvers	Release Management
	Notify Approvers	Release Management
Test	Manual Tests	Microsoft Test Manager
	Automated Tests	Coded UI, Web, Load
	Create Test Data	BacPac with SSDT
App	Configure Application	Tokenization
	Install Application	NuGet, WebDeploy, etc
Infra	Configure Environment	Desired State Configuration
	Provision Environment	Azure Resource Manager
Plan	Tracking Work	TFS / VSTS
	Planning Work	TFS / VSTS

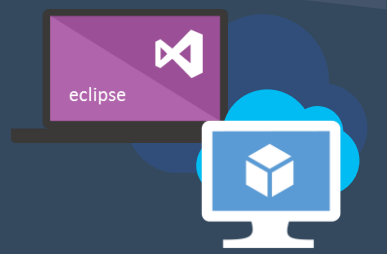
Microsoft Ecosystem

People | Process | Tools

01

Develop

Developer Workstation



Team Collaboration

TFS / VSTS

Workstations - On-Premises | Hybrid | Cloud

02

Build & Test

Build/CI

TFS / VSTS

Release Management for Visual Studio

Test

TFS / VSTS

Microsoft Test Manager

ALM Services - On-Premises | Hybrid | Cloud

03

Deploy

Release

Microsoft System Center

Release Management for Visual Studio

Automation Service

PowerShell | WAML

Azure Resource Management

xPlat Command Line

Environments - On-Premises | Hybrid | Cloud

04

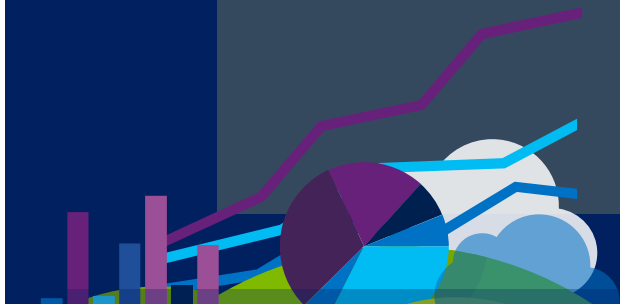
Monitor & Learn

Monitor

Microsoft System Center

VSTS

Application Insights



Monitoring - On-Premises | Hybrid | Cloud

Mixed Ecosystem

People | Process | Tools

01 Develop

Developer Workstation

Team Collaboration

GitHub
CodePlex

02 Build & Test

Build/CI

gradle
GRUNT
Jenkins
Hudson

Test

gradle
GRUNT

03 Deploy

Configuration

puppet labs
CHEF

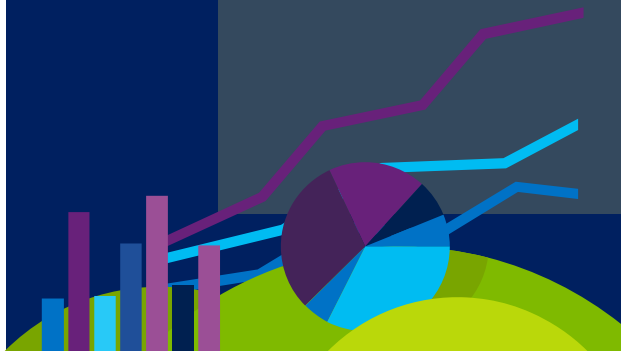
Release

gradle
GRUNT
Jenkins
Hudson
VAGRANT

04 Monitor & Learn

Monitor

Nagios
ZABBIX



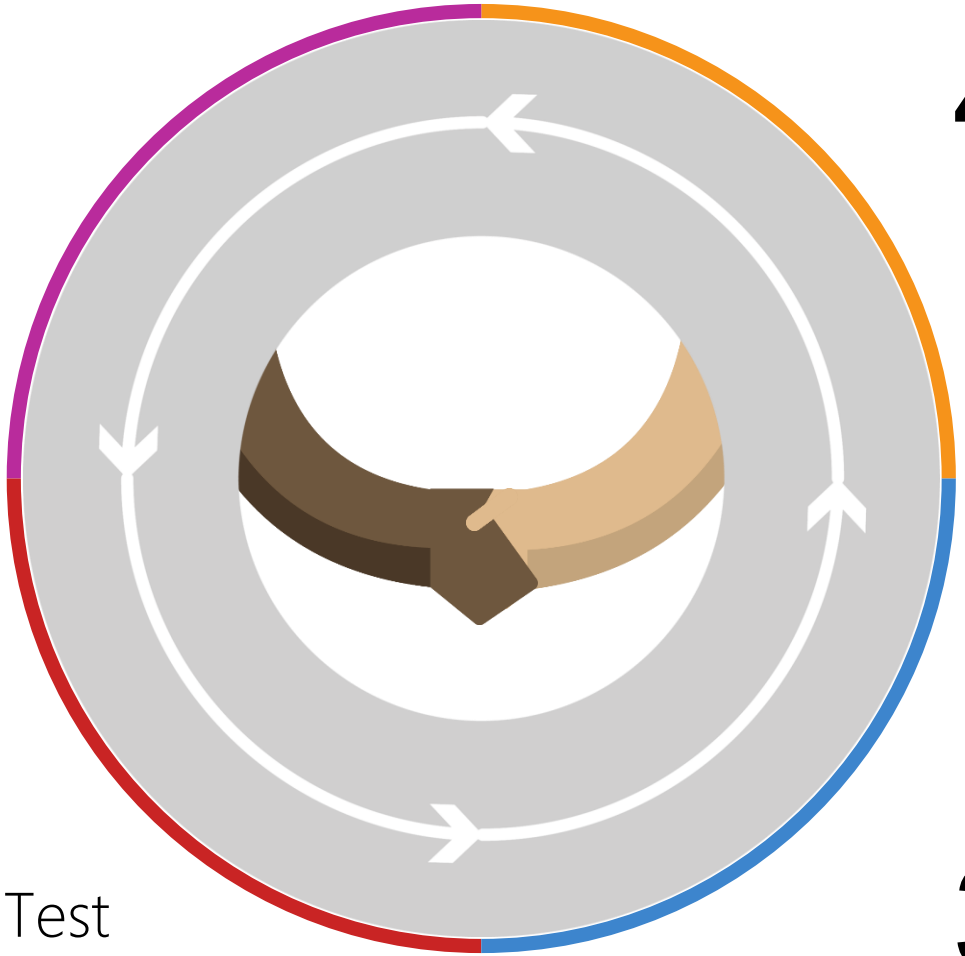
This graphic shows OSS and partner products that are integrated with the Microsoft DevOps solution

DevOps

1 Plan

4 Monitor + Learn

Development



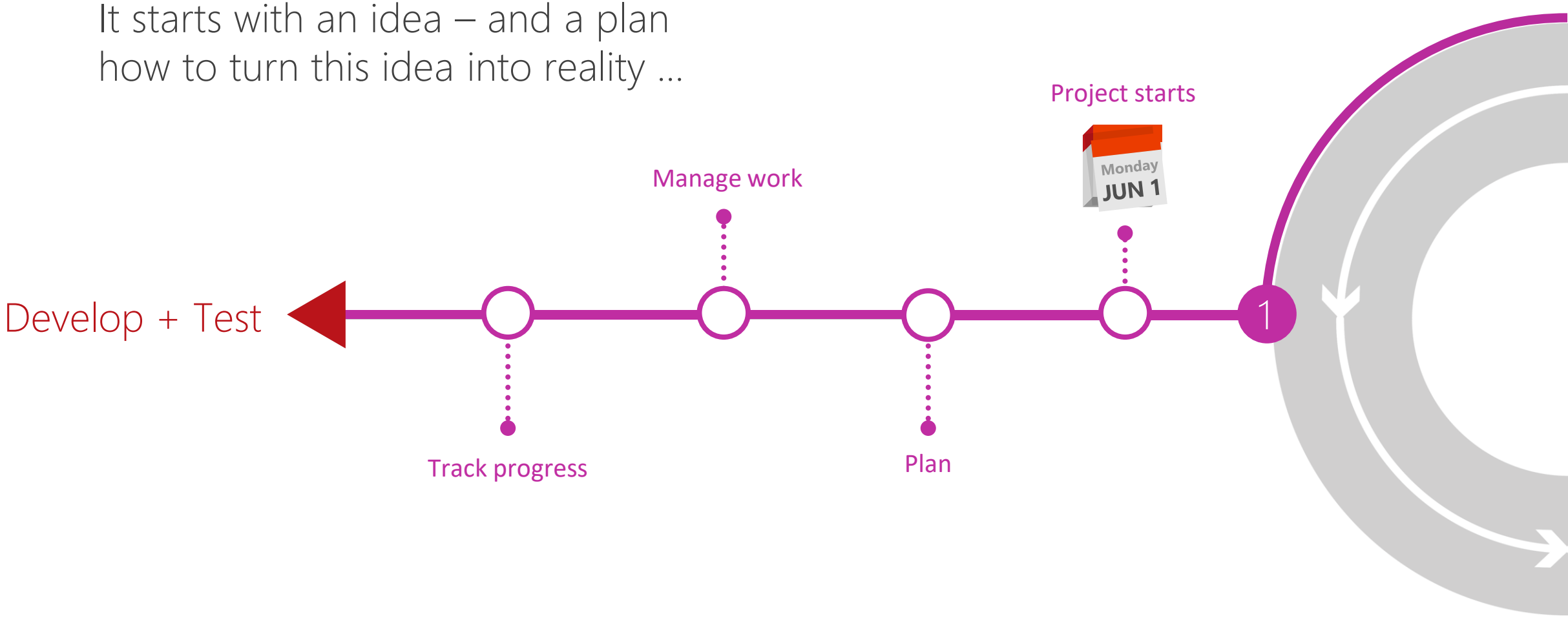
Production

2 Develop + Test

3 Release

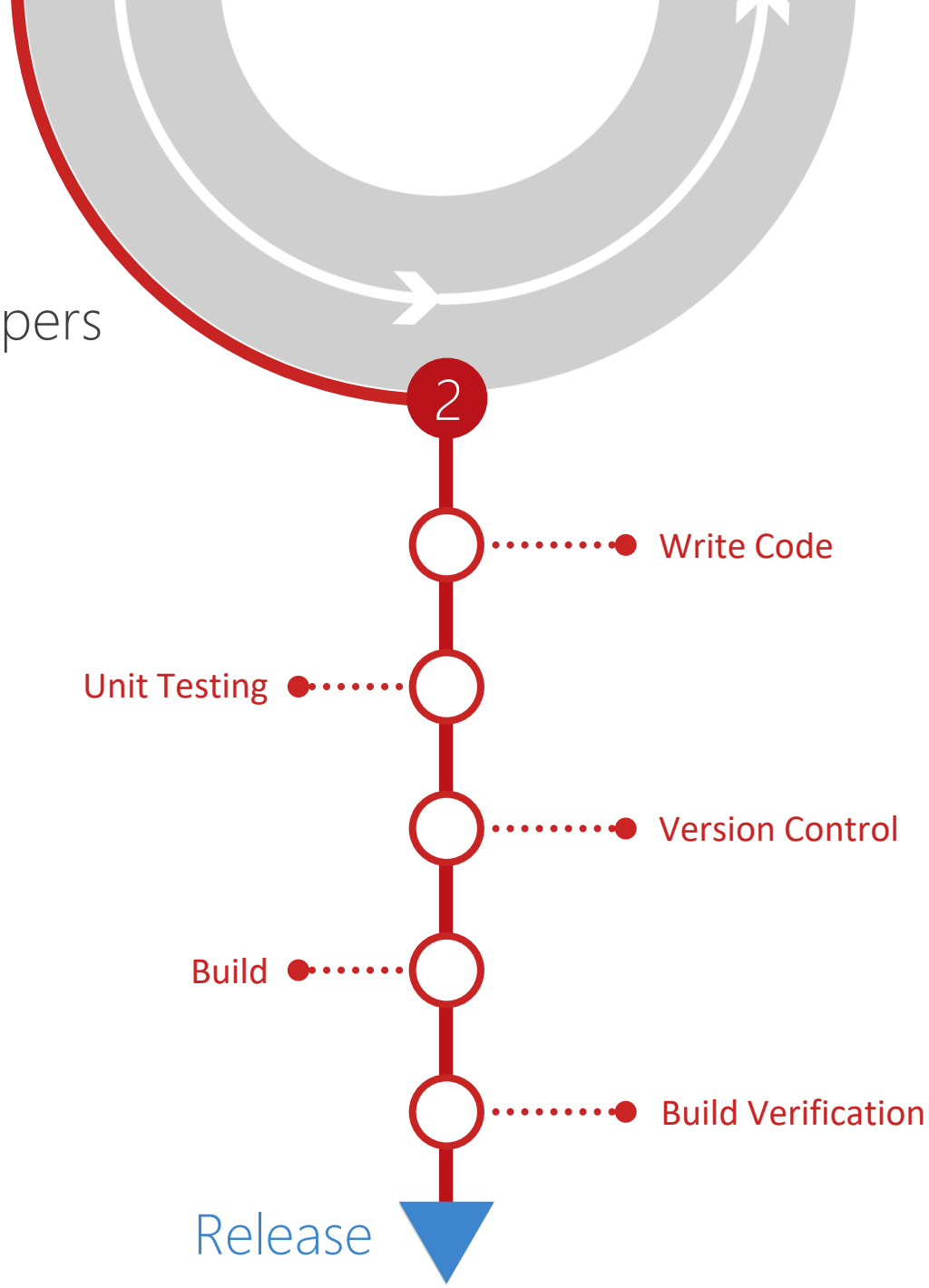
Plan

It starts with an idea – and a plan
how to turn this idea into reality …

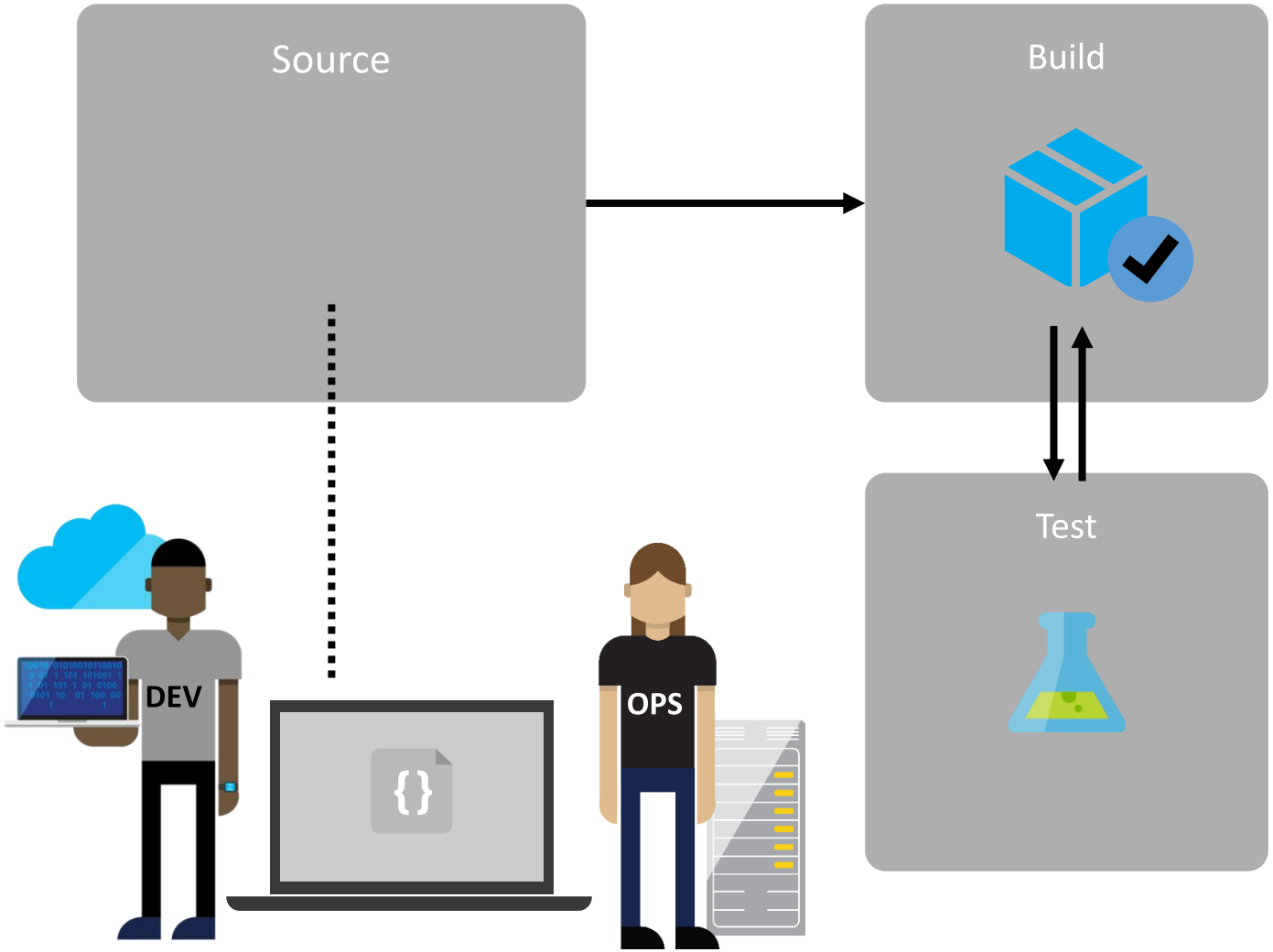


Develop + Test

Once the iteration starts, developers turn great ideas into features …



Continuous Integration



Value

- Accelerate Delivery
- Repeatability
- Optimized Resources

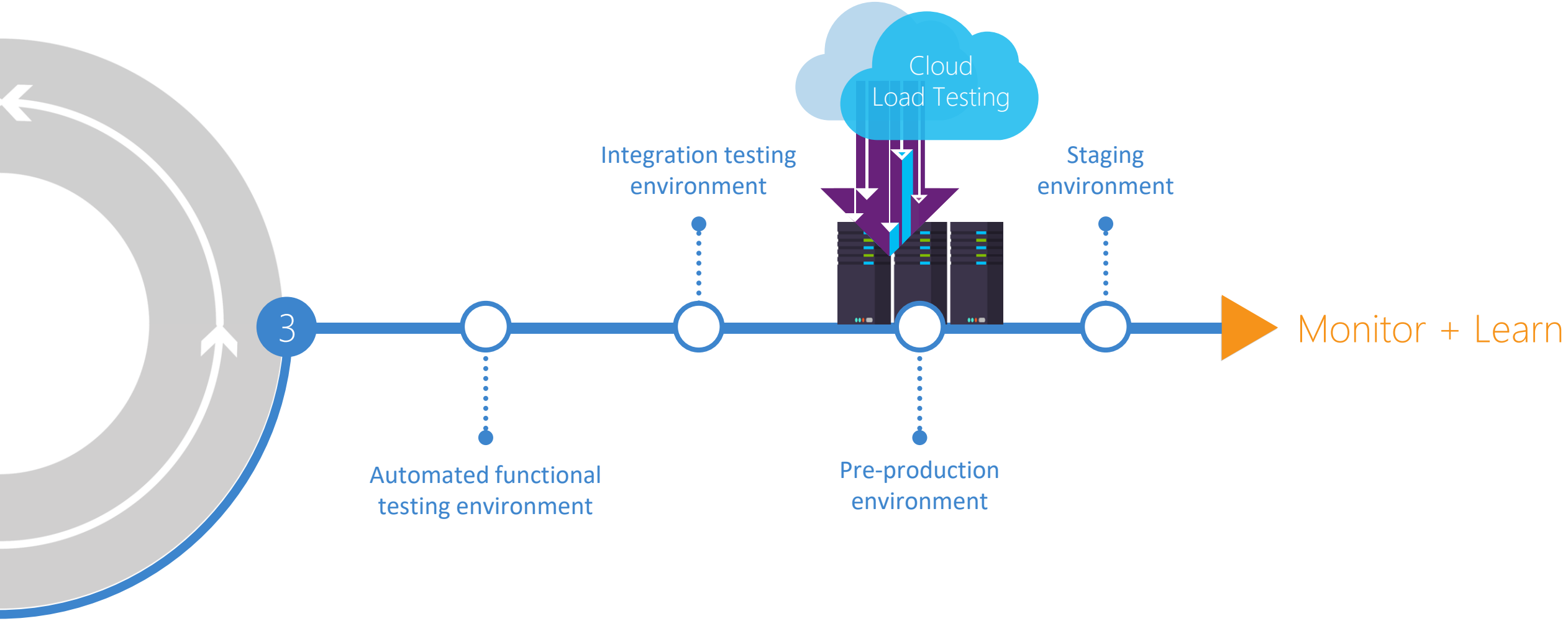
Measure

- More frequent releases
- MTTR
- MTTD

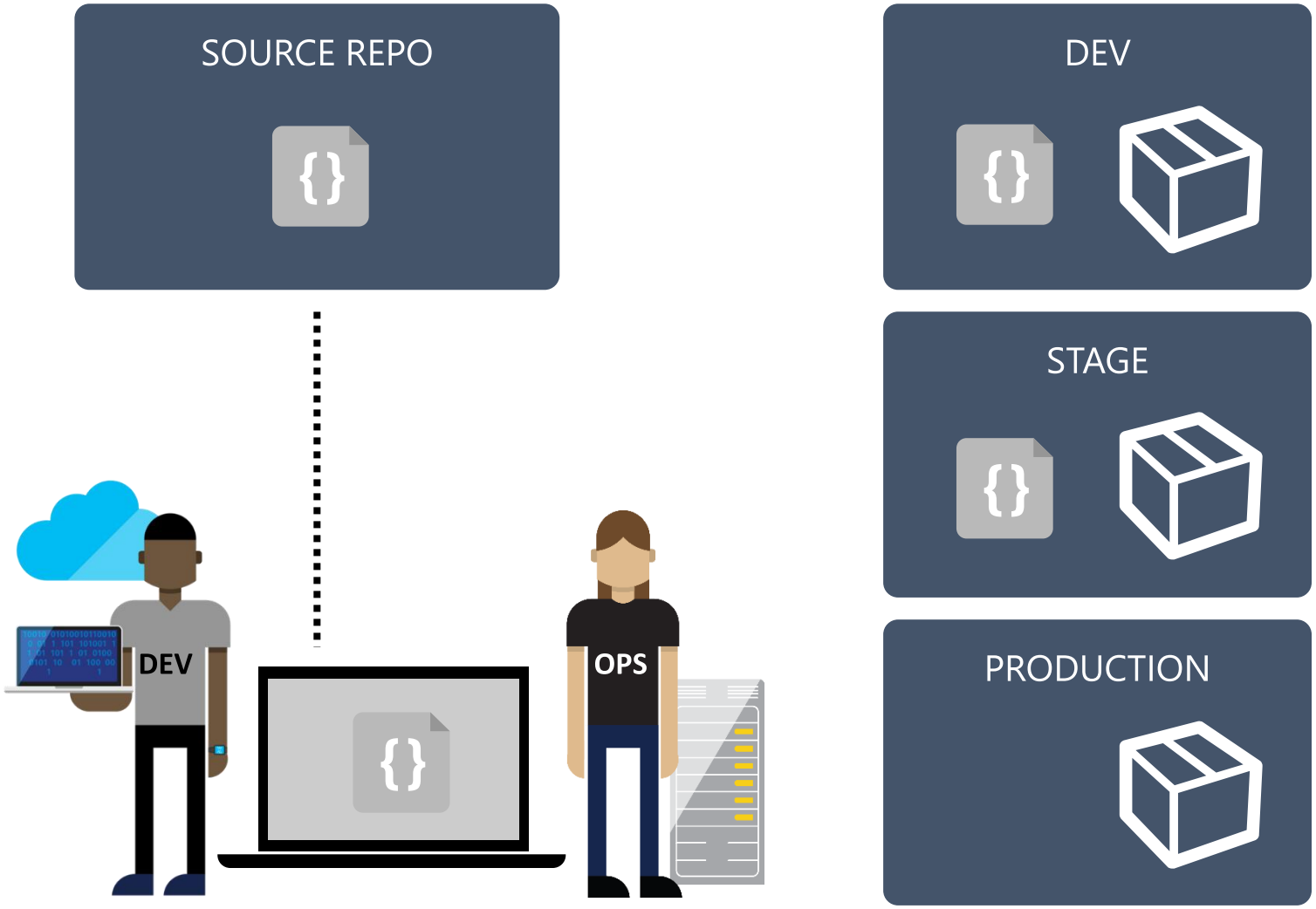
Continuous Integration

Release

When all tests pass, the build is deployed to testing environments for each stage in the release process



Release Management



Value

- Optimized Resources
- Accelerate Delivery

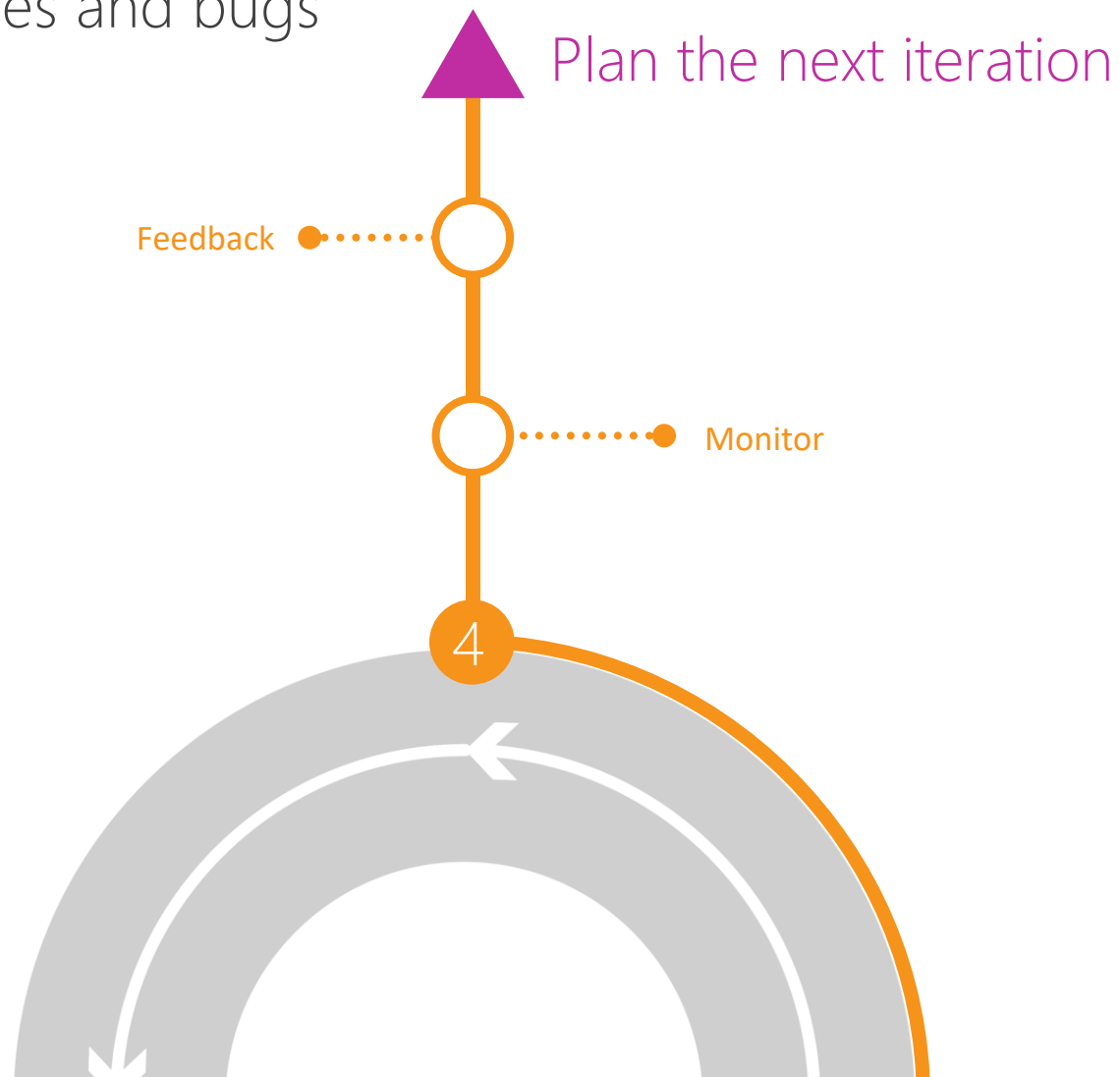
Measure

- Deployment Rate
- MTTR
- Availability

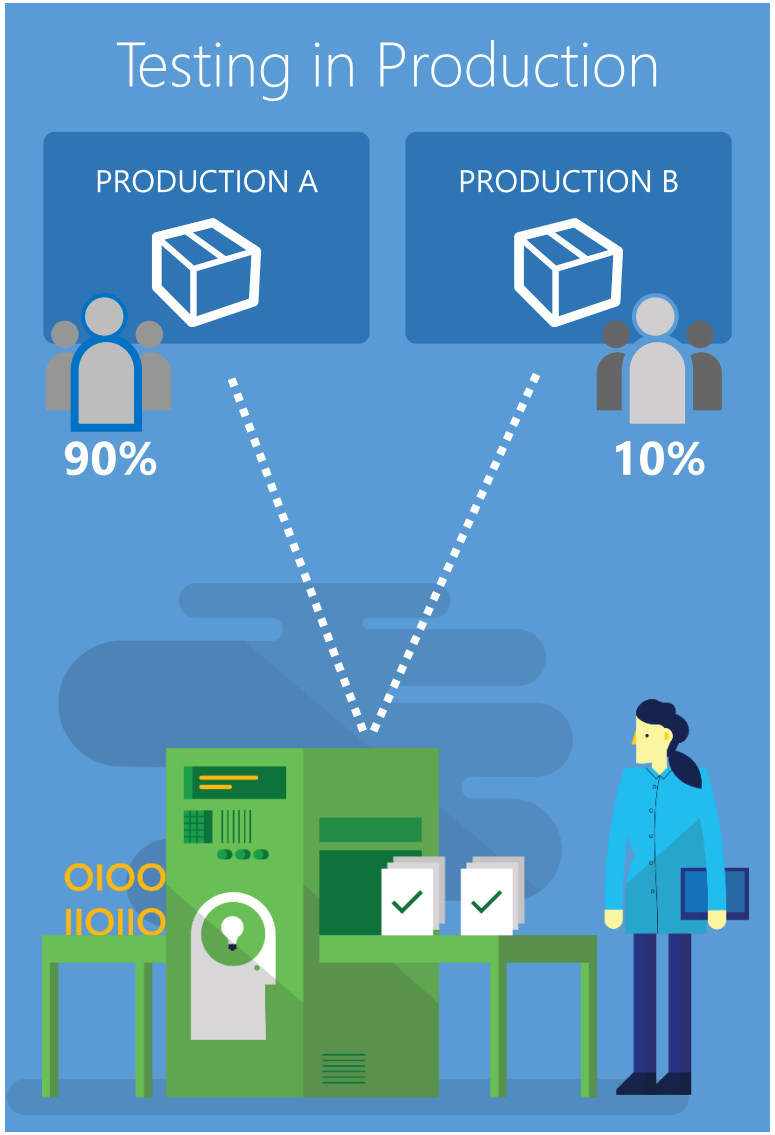
Continuous Deployment & Release Management

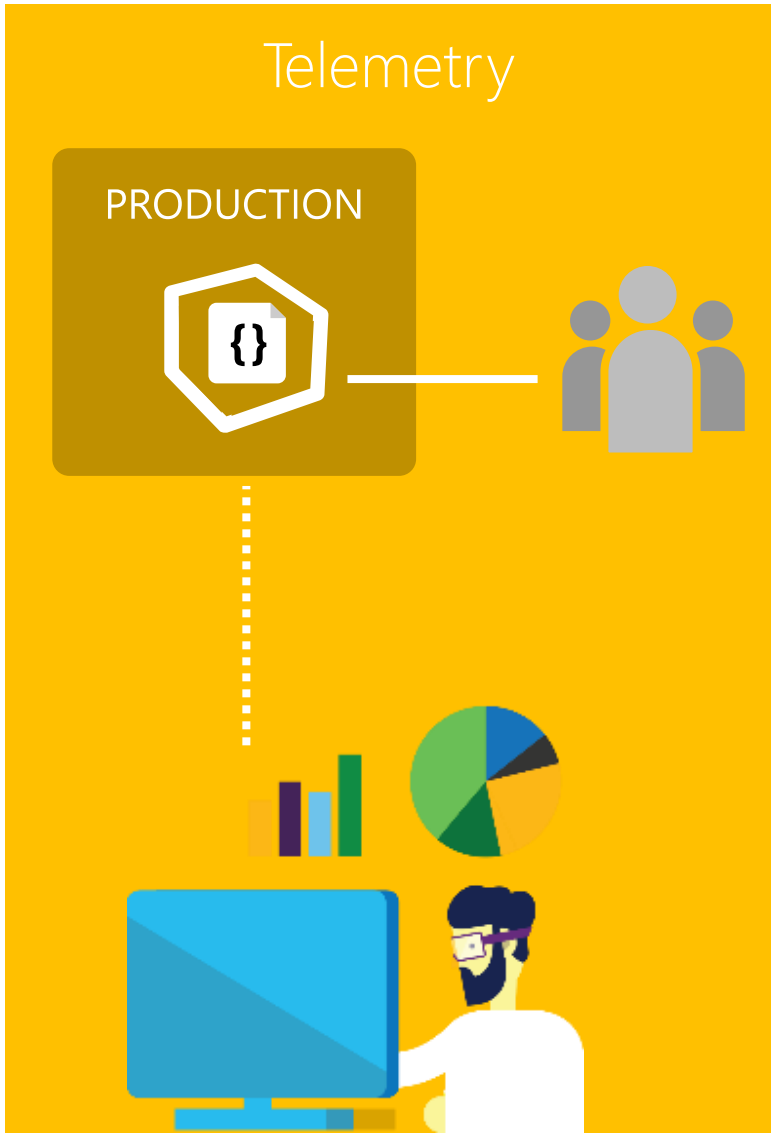
Monitor + Learn

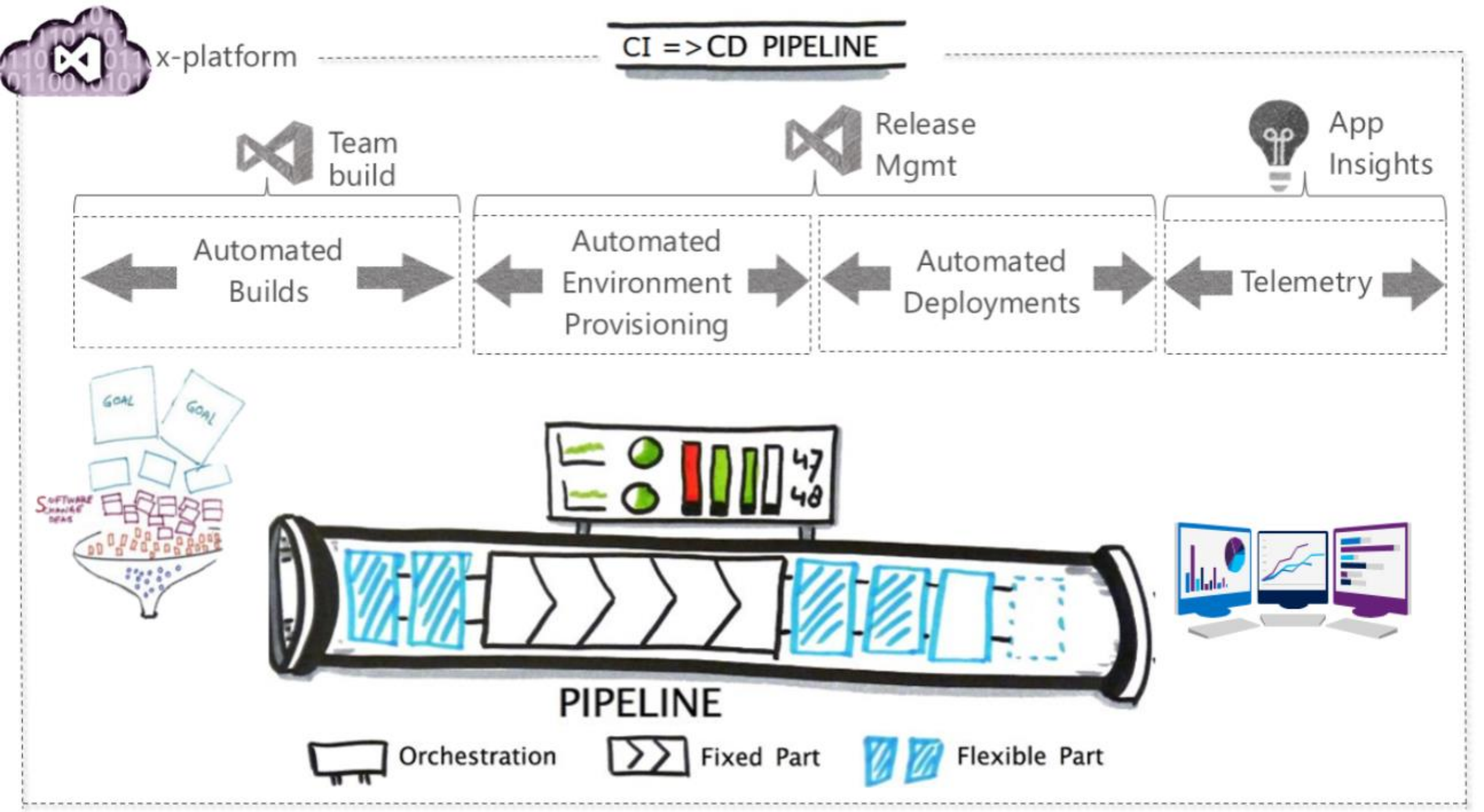
Learn and understand how users use your app, how it reacts and quickly fix issues and bugs



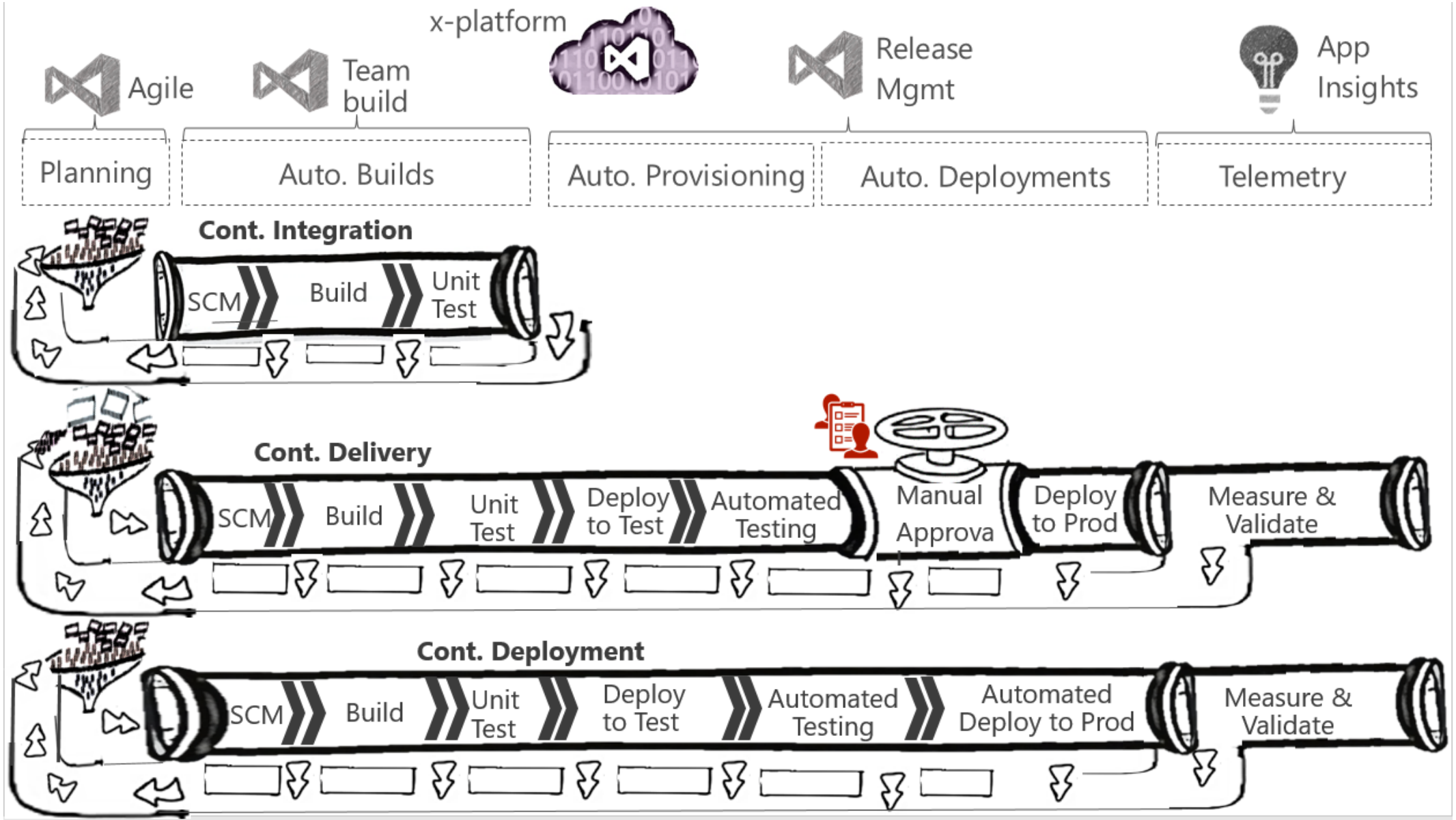
Monitor and Learn






x-platform
CI =>CD PIPELINE
Team build
Release Mgmt
App Insights
Automated Builds
Automated Environment Provisioning
Automated Deployments
Telemetry
GOAL
GOAL
47
48
PIPELINE
Orchestration
Fixed Part
Flexible Part

Continuous Delivery vs. Continuous Deployment



Review

What are the 3 core components of DevOps

- People, Process, Tools

People - What are the 3 areas to focus on when adopting a DevOps culture?

- Trust, Respect, Communication

Process - What are some DevOps practices?

- IaC, CI/CD, Automated Testing, App Perf Monitoring and reporting, Feature Flags

Binary package consumption

Package management allows you to reuse trusted components. An example is NuGet feeds. If you consume and reuse packages consistently, then you can keep your application fresh as those packages are updated. This is particularly important for security and compliance as the packages can be scanned, uniquely identified, and, in the event of new vulnerabilities, automatically updated.

Packages should not usually be included in version control because they may significantly increase the size of version control repositories. Better to have the packages stored in package management, such as internal NuGet feeds.

Continuous integration builds

Continuous integration builds are builds that are triggered upon every check-in of code. They should be quick to complete and include automated tests (such as unit tests). They may or may not drop artifacts upon completion.

Package publishing

Package publishing may tie into binary package consumption. You should use a version that matches the build and deployment number for full transparency (if a deployment breaks, you should be able to find the matching package quickly). The package should be hosted somewhere accessible by projects, such as NuGet feeds.

Automated testing

Testing could include unit, integration, automated UI, web performance, and load tests. Tests should be able to run independently and as often as possible without breaking, such as in continuous integration builds or automated deployments.

Continuous delivery

Continuous delivery means that deployments are triggered automatically after a build completes. The process should include picking up a compiled package of code, setting the target environment in the desired configuration state, and deploying automatically to environments. It may include approvals into QA or production environments.

DevOps in the Cloud

Scale. Scale vertically or horizontally to as many machines as you need, only limited by the amount you wish to pay for machines. This is not always possible on-premises, where an organization might be limited by physical memory available for VMs, or have to go through a difficult process to acquire new machines.

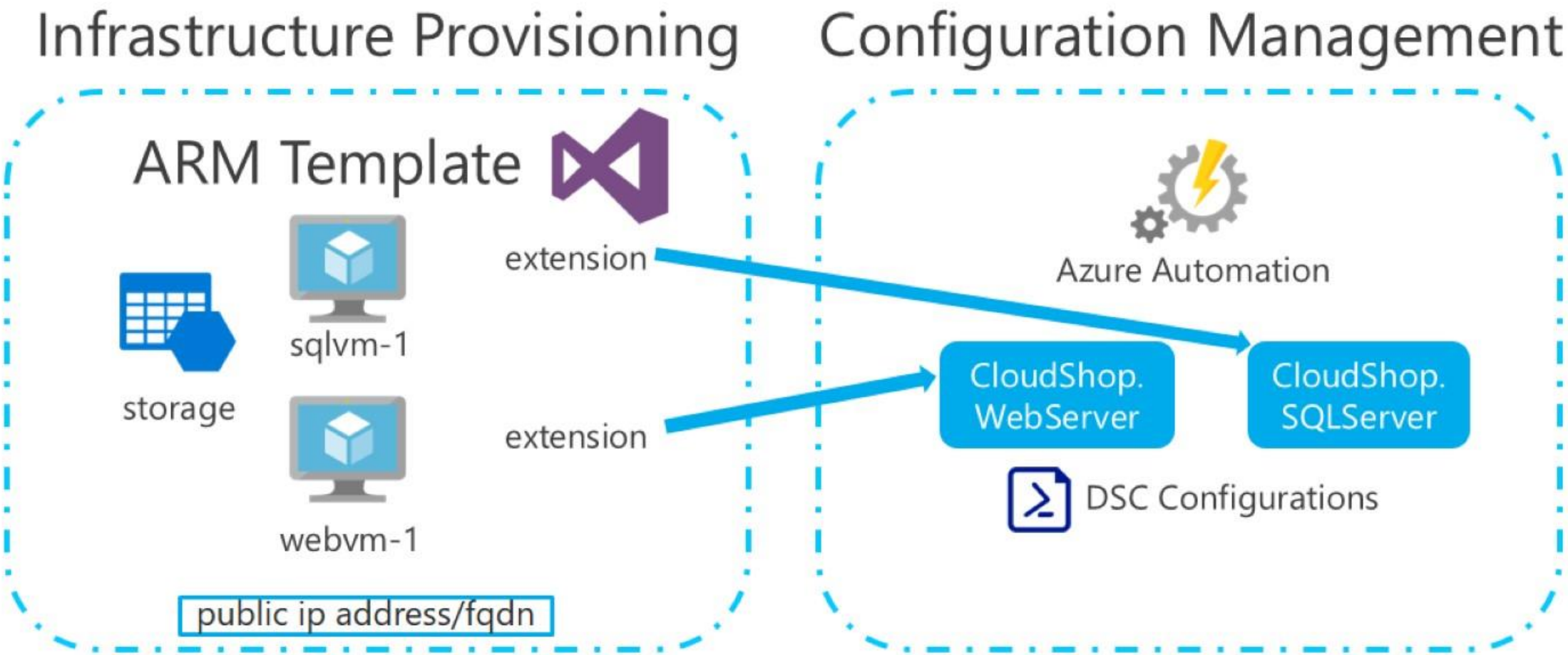
Data storage. Although it varies among cloud providers, the price for storage accounts in the cloud might be less expensive than hosting your own SAN with enough space.

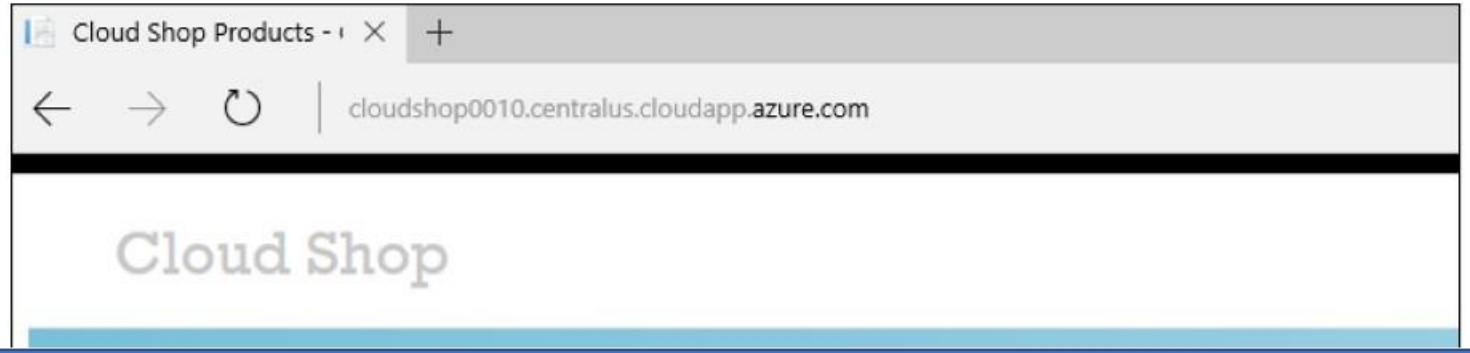
Reliability. Many cloud providers offer a service-level agreement of nearly 100% (potentially 99.9%), and that performance isn't necessarily a guarantee on-premises.

Time and people needed. When on-premises, you might not carefully consider the cost for time and people involved with setting up and maintaining servers. It might take a few days to get a machine provisioned on-premises; in the cloud, it may take only minutes. Instead of assigning roles to start up and maintain machines on-premises, time can instead be used to improve cloud infrastructure or to innovate to make the DevOps process better.

Security. Cloud providers such as Microsoft, although they are larger targets for attacks, are constantly looking for breaches in security with aggressive white hat hackers. That might mean that a cloud is possibly more secure than your on-premises machines.

Cost. Cost can be optimized for variable loads. This, of course, depends on your application architecture and requirements. The cloud makes it very economic to scale out, not necessarily to scale up.


Infrastructure Provisioning
Configuration Management
ARM Template
extension
sqlvm-1
storage
extension
webvm-1
public ip address/fqdn
Azure Automation
CloudShop.
WebServer
CloudShop.
SQLServer
DSC Configurations


Cloud Shop Products -
cloudshop0010.centralus.cloudapp.azure.com
Cloud Shop

Review

What are the 2 approaches to the languages used in IaC solutions?

- Declarative, Imperative

What are the 2 methods for IaC deployment to clients?

- Pull, Push

What are the 3 main benefits to leveraging IaC?

- Cost (reduction), Speed (faster execution), Risk (remove errors and security violations)

Continuous Integration & Delivery/Deployment (CI/CD)

What is Continuous Integration & Delivery/Deployment (CI/CD)?

Continuous Integration

"...is a development practice that requires developers to integrate code into a shared repository several times a day.

Each check-in is then verified by an automated build, allowing teams to detect problems early."

- ThoughtWorks

Continuous Delivery

"...is the practice that usually comes after CI and can be implemented to push a new known good build to a single environment either automatically or via automation capabilities that an authorized user can schedule. "

- TIPROGUY.com

Continuous Deployment

"Usually confused with continuous delivery, continuous deployment means that every change is automatically deployed to production.

*Continuous delivery means that the team ensures every change **CAN** be deployed to production but may choose not to do it"*

- wikipedia

Continuous Delivery/Deployment Principles

Build Quality In

Work in small batches

Computers perform repetitive tasks, people solve problems

Everyone is responsible

Relentlessly pursue continuous improvement

https://continuousdelivery.com/principles

What are the benefits of CI/CD

Continuous Integration

Integration issues are detected early

Few changes lost when reverting to a previous state

Always a current build for test/demo/release

Frequent check-ins favors modular, less complex code

Continuous Delivery & Deployment

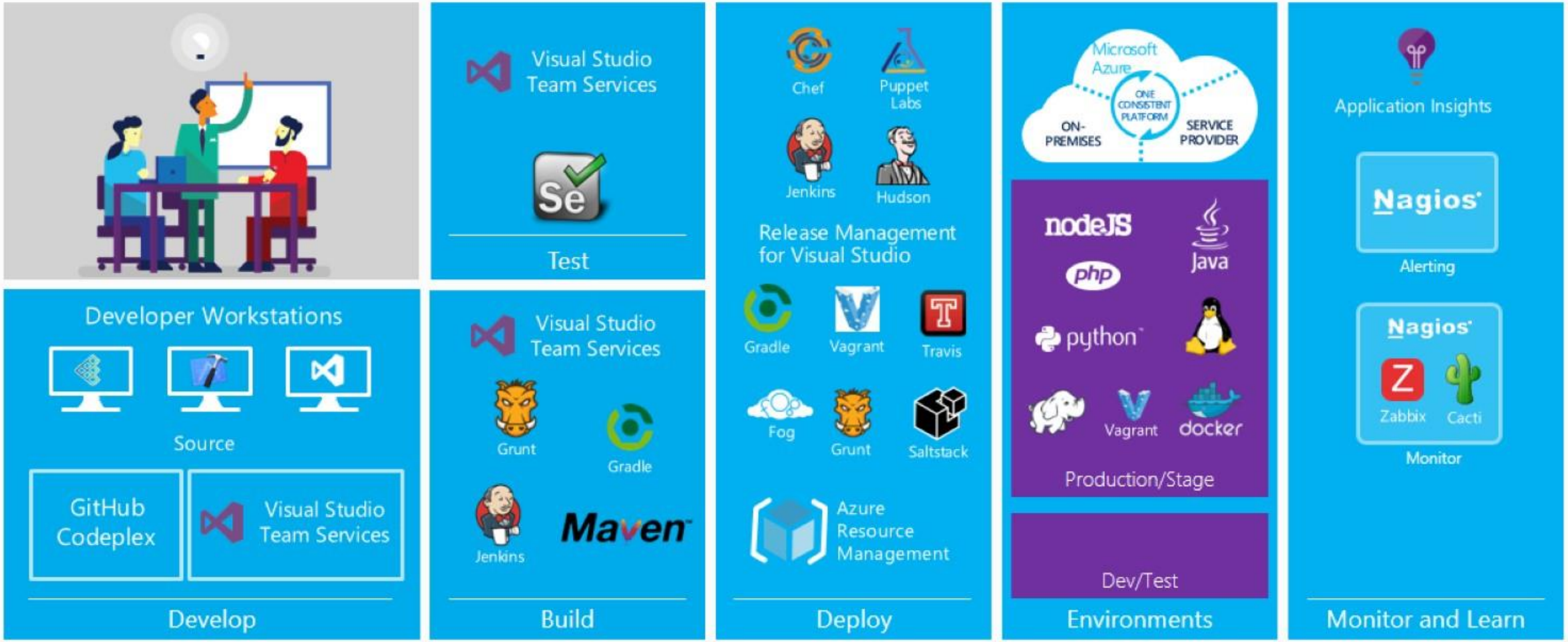
Reduced cost, time and risk
- Rapid delivery of incremental changes
- Improved business competency
- Testing is not a phase

Reduced deployment times
- Small incremental changes, faster business value
- Releasing as frequently as possible

Collaboration
- "Devops culture"
- Devs, testers, systems & db admins, managers

Repeatable and reliable releases
- "Push button" automation
- Deployment to any environment
- Source control for traceability

Tools for CI/CD



Review

What are 5 the CD principles?

- Build Quality In, Work in Small batches, Everyone is responsible, Computers perform repetitive tasks, people solve problems, Relentlessly pursue continuous improvement

Between Continuous Delivery or Deployment, which one involves full automation to Production?

- Continuous Deployment

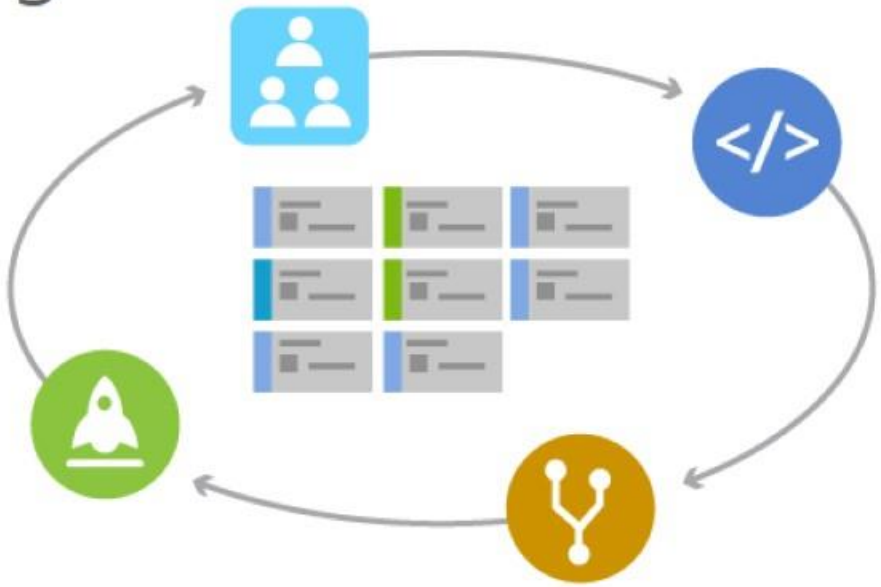
What are some of the benefits of CI/CD?

- Integration issues detected early, collaboration is a must, repeatable and reliable releases, rapid delivery of incremental changes, testing is not a phase

Visual Studio Team Services

Visual Studio Team Services (VSTS)

- Much more than just source control
  - Release and Package management
  - Agile planning tools
  - Continuous Integration and Testing
  - Load Testing
  - Process Customization
  - Reporting



Visual Studio Team Services (VSTS)

- Code in any IDE / Language
- Target any platform
- Integrate familiar tools

VSTS Benefits

- Software-as-a-Service
  - Basically a SaaS-based version of TFS
- Frequent updates

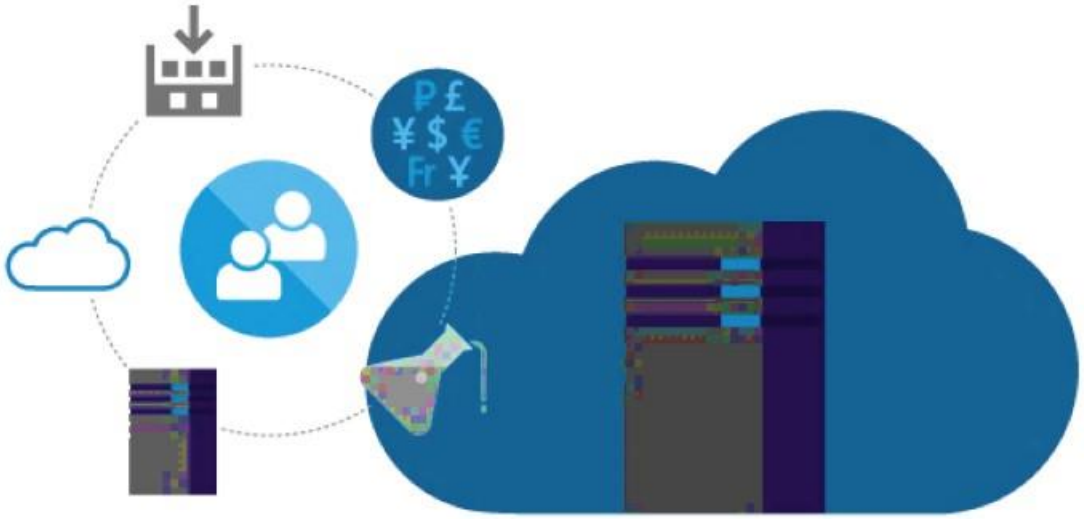
VSTS Licensing

Free Tier

- Up to 5 users
- Unlimited repositories

Paid Tier

- More users for a growing team
- Unlimited repositories



Team Foundation Server (TFS)

- Manually hosted
  - On-premise version of VSTS
- Not as frequent updates
  - ~3 months older than VSTS
- Extended customization options
- You own it, you manage it

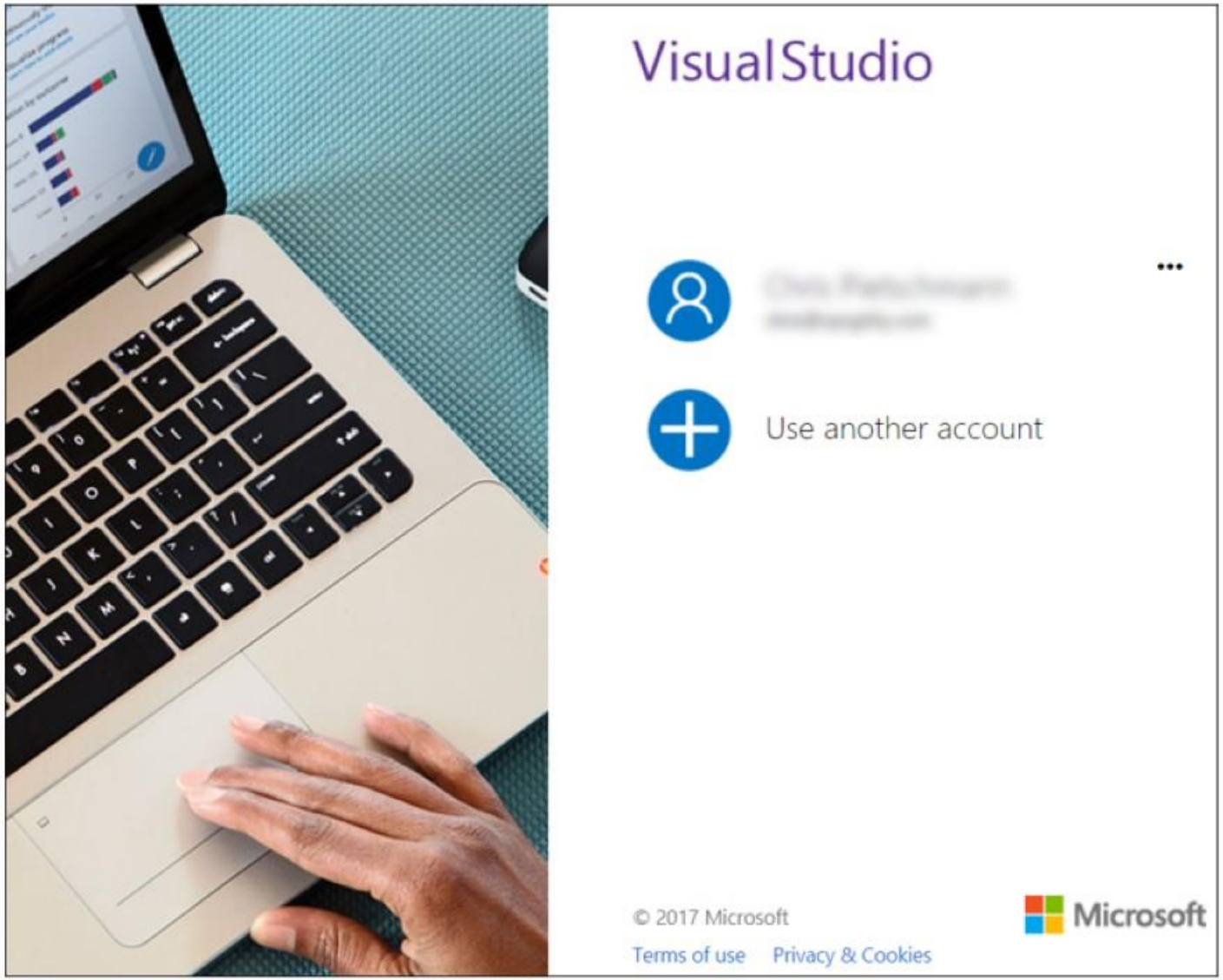
Accessing VSTS

Personal account

Microsoft account (MSA)

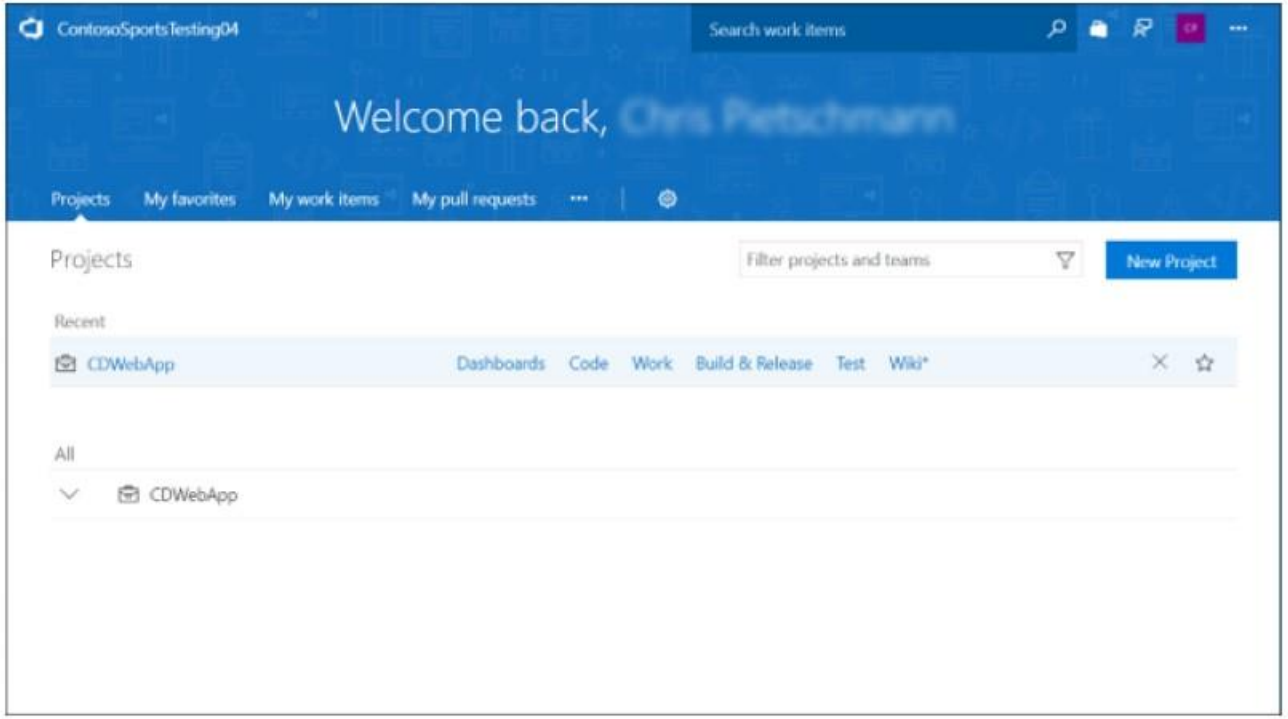
Organization or School account

Azure Active Directory (AAD)



VSTS Account

- Unit of isolation
  - Think of separate database in a multi-tenant environment
- One or more Projects

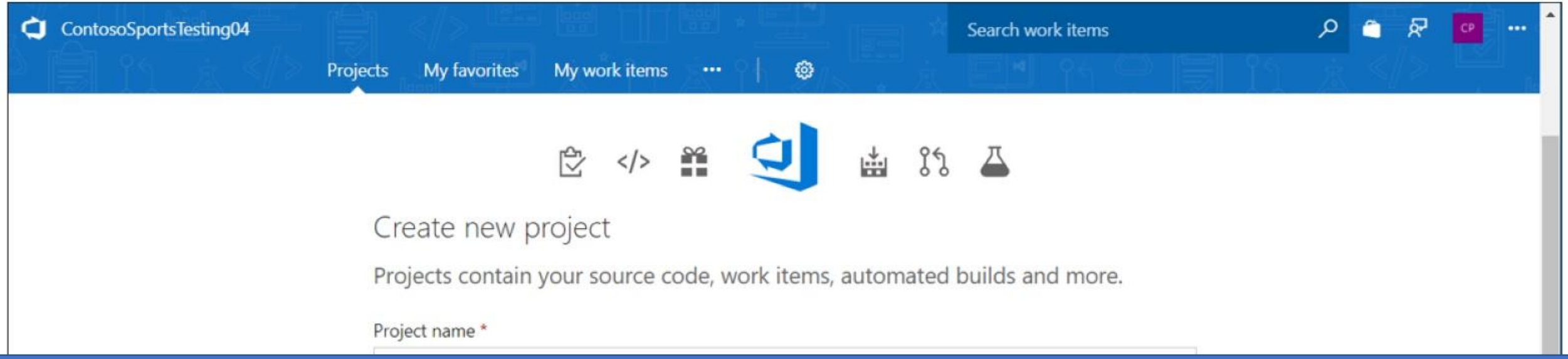


Project

- One VSTS Account with Many Projects
  - Users and licenses are assigned to project
- Project can have…
  - Unlimited Git repos
  - A Single TFVC repo
- Contains Hubs – functional areas
  - Code, Work, Build & Release, Test

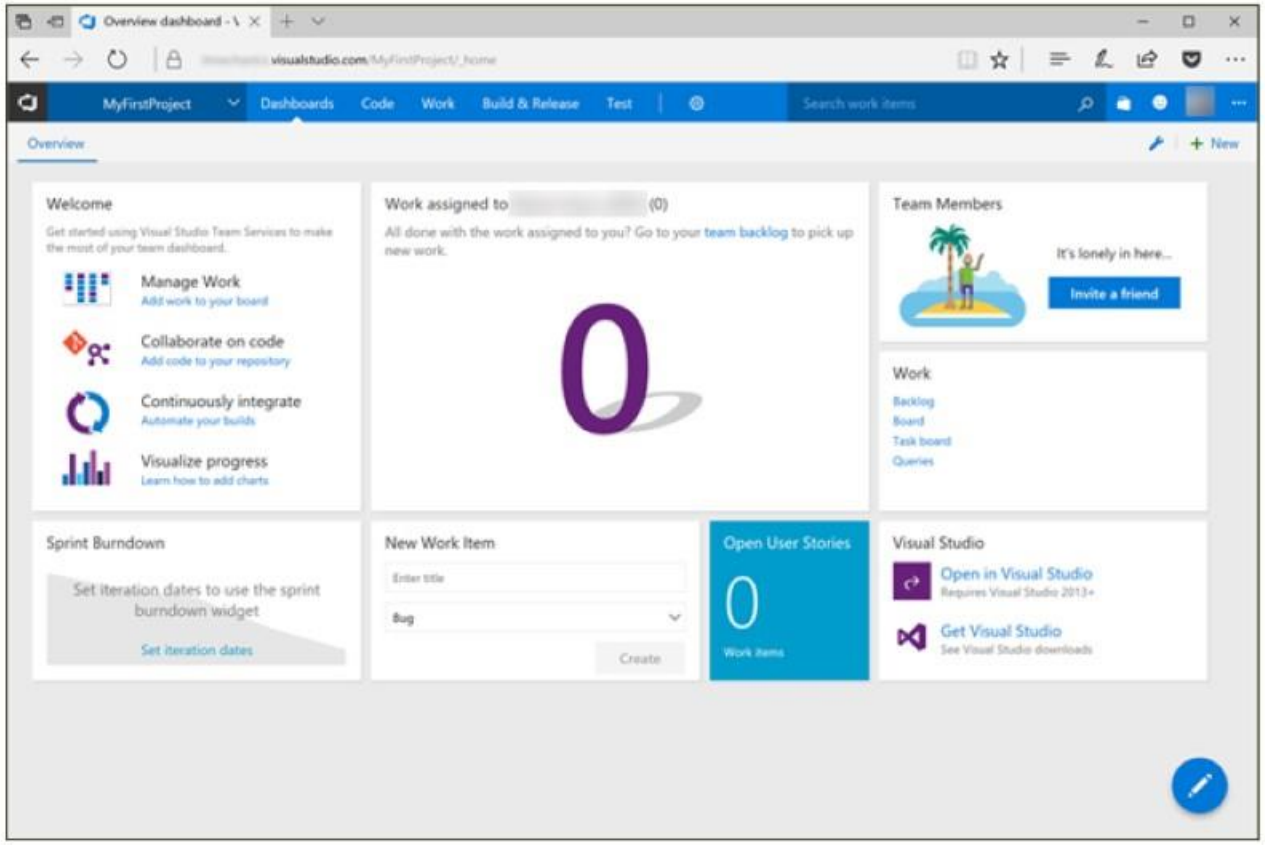
New Project

- Version Control
  - Git or Team Foundation Version Control
- Work item process
  - Agile, CMMI, Scrum



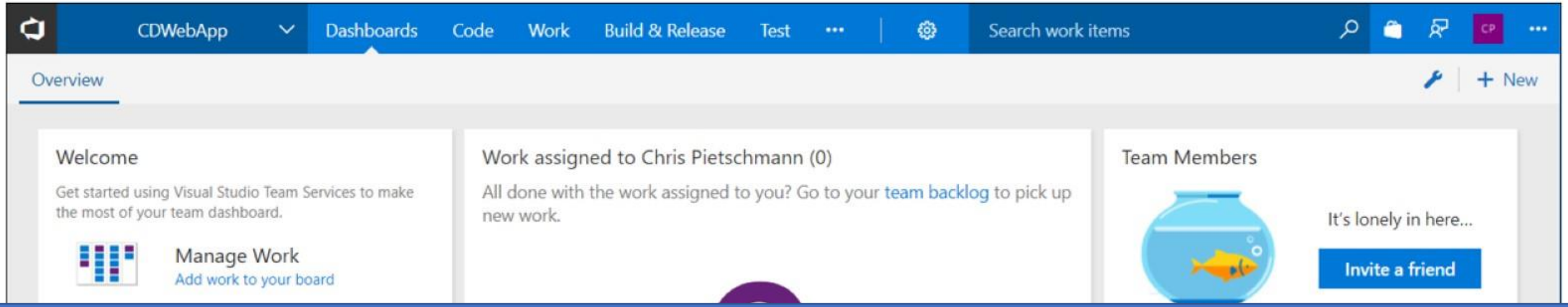
Dashboards

- Created for each project
  - Customize as you want
- Can add multiple



Hubs

- Code
- Work
- Build & Release
- Test



Code

- View Code files, History, Compare, etc.

CDWebApp | Dashboards | Code | Work | Build & Release | Test | Search work items

CDWebApp | Files | History | Branches | Tags | Pull Requests | Clone

master | CDWebApp / CDWebApp / **Web.config**

- CDWebApp
  - CDWebApp
    - App_Start
    - Content
    - Controllers
    - fonts
    - Models
    - Properties
    - Scripts

Contents | History | Compare | Blame | Edit | Rename

```
1  <?xml version="1.0" encoding="utf-8"?>
2  <!--
3    For more information on how to configure your ASP.NET application, please visit
4    http://go.microsoft.com/fwlink/?LinkId=301880
5    -->
6  <configuration>
7    <configSections>
8      <!-- For more information on Entity Framework configuration, visit http://go.microsoft.com/fwlink/?LinkID=237468 -->
9      <section name="entityFramework" type="System.Data.Entity.Internal.ConfigFile.EntityFrameworkSection, EntityFramework, Version=6.0.0.0, Cult
10   </configSections>
11   <connectionStrings>
12     <add name="DefaultConnection" connectionString="Data Source=(LocalDb)\MSSQLLocalDB;AttachDbFilename=|DataDirectory|\aspnet-CDWebApp-2017020
13       providerName="System.Data.SqlClient" />
14   </connectionStrings>
15   <appSettings>
16     <add key="webpages:Version" value="3.0.0.0" />
17     <add key="webpages:Enabled" value="false" />
```

Work

- Task Backlog, Stories, Sprints, Work Items, etc.

CDWebApp | Dashboards | Code | Work | Build & Release | Test | Search work items

Backlogs | Queries

Features
Backlog items
Current
Sprint 1
Future
Sprint 2
Sprint 3
Sprint 4
Sprint 5
Sprint 6

CDWebApp Team Sprint 1

No iteration dates
Set dates

Backlog | Board | Capacity

Work details On

New | Create query | Column options

Type: Product Backlog Item
Title: Add

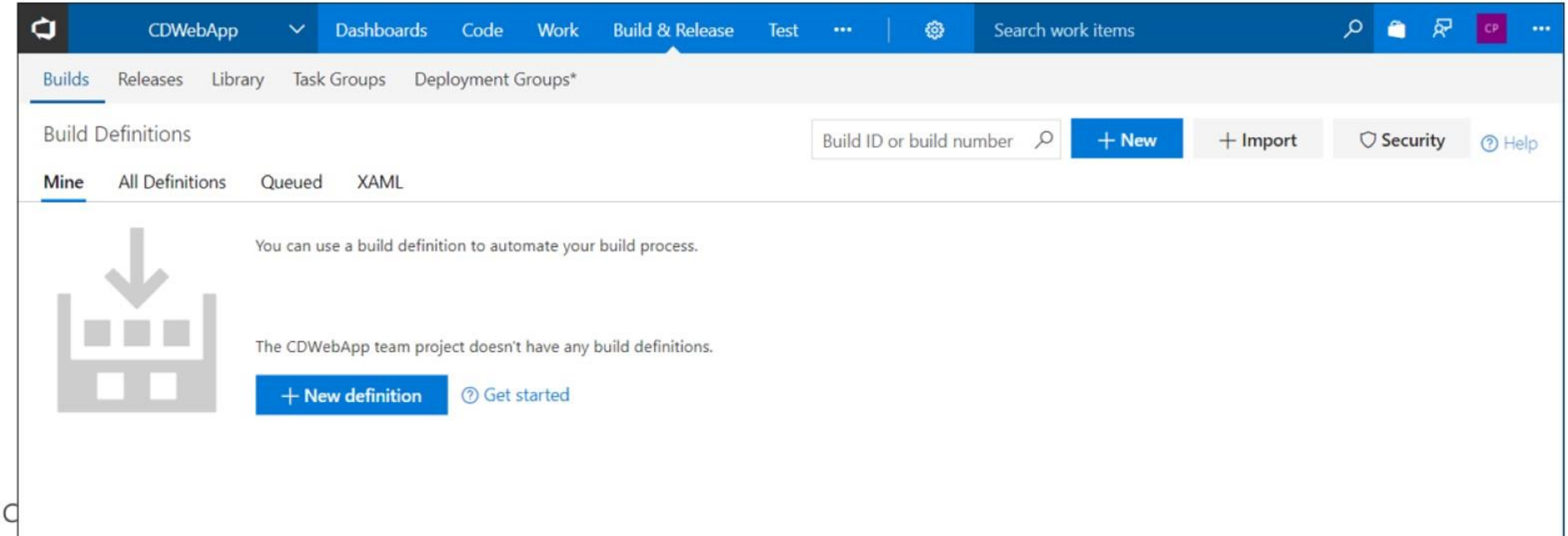
Order	Work Item Type	Title	State	Effort	Value Area
1	Product Backl...	Add Login / Authentication capabilities	Committed		Business
2	Product Backl...	Add User Profile Page	Committed		Business
3	Product Backl...	Add User List Page	Approved		Business
4	Product Backl...	Add Entity 1 List / Search page	Approved		Business
5	Product Backl...	Add Enttity 1 Details / Edit page	Approved		Business
6	Product Backl...	Add "Default" Dashboard view	New		Business

Work details

Work
Work By: Activity
Work By: Assigned To

Build & Release

- Manage Build Definitions
- Manage Release Definitions



Test

- Manage Test Plans and Test Cases

CDWebApp | Dashboards | Code | Work | Build & Release | Test | Search work items

Test Plans | Parameters | Configurations | Runs | Machines | Load test

Sprint 1: MyTest1 (Id: 8)

MyTest1 (1)

Test suite: MyTest1 (Suite ID: 9)

No iteration dates

Tests | Charts

Outcome All | Tester All | Configuration All | View List

New | Add existing | Run | Order tests

Outcome	Order	ID	Title	Configurat...	Tester
Active	1	10	Smoke Test User Profile Page	Windows ...	Chris Piet...

Source Control

VSTS Source Control

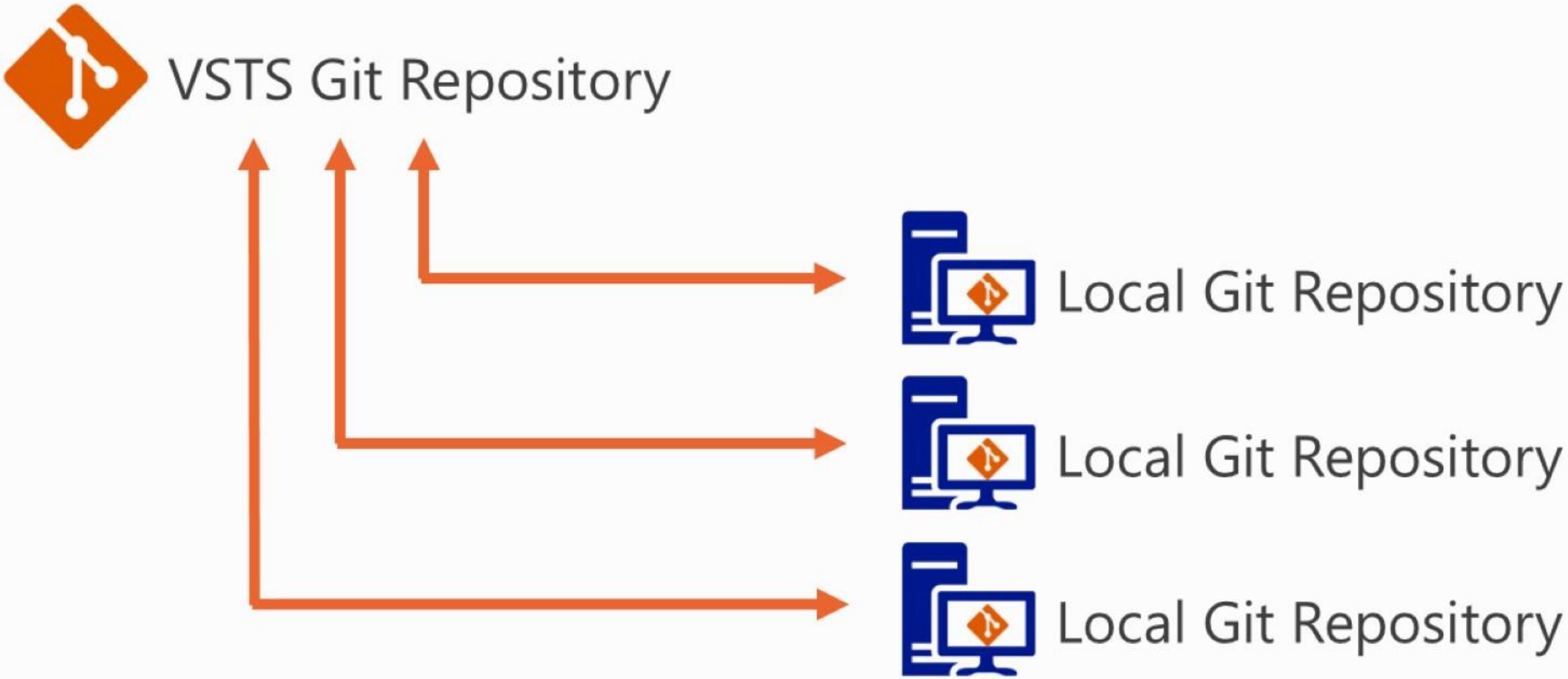
- TFVC
  - Team Foundation Version Control
  - Original source control released with Team Foundation Server (TFS)
  - Use Visual Studio and other Microsoft tooling to work with repository
- Git
  - Distributed Source Control
  - Open Source (http://git-scm.com)
  - Use any Git tooling to work with repository

Git

- Distributed Source Control
  - The Git server hosts the main Git repository
  - Local repositories are used for local “working” copy
    - When ready, changes are sync’d to commit changes up to the server, and share with the rest of the team

git

Git Repositories

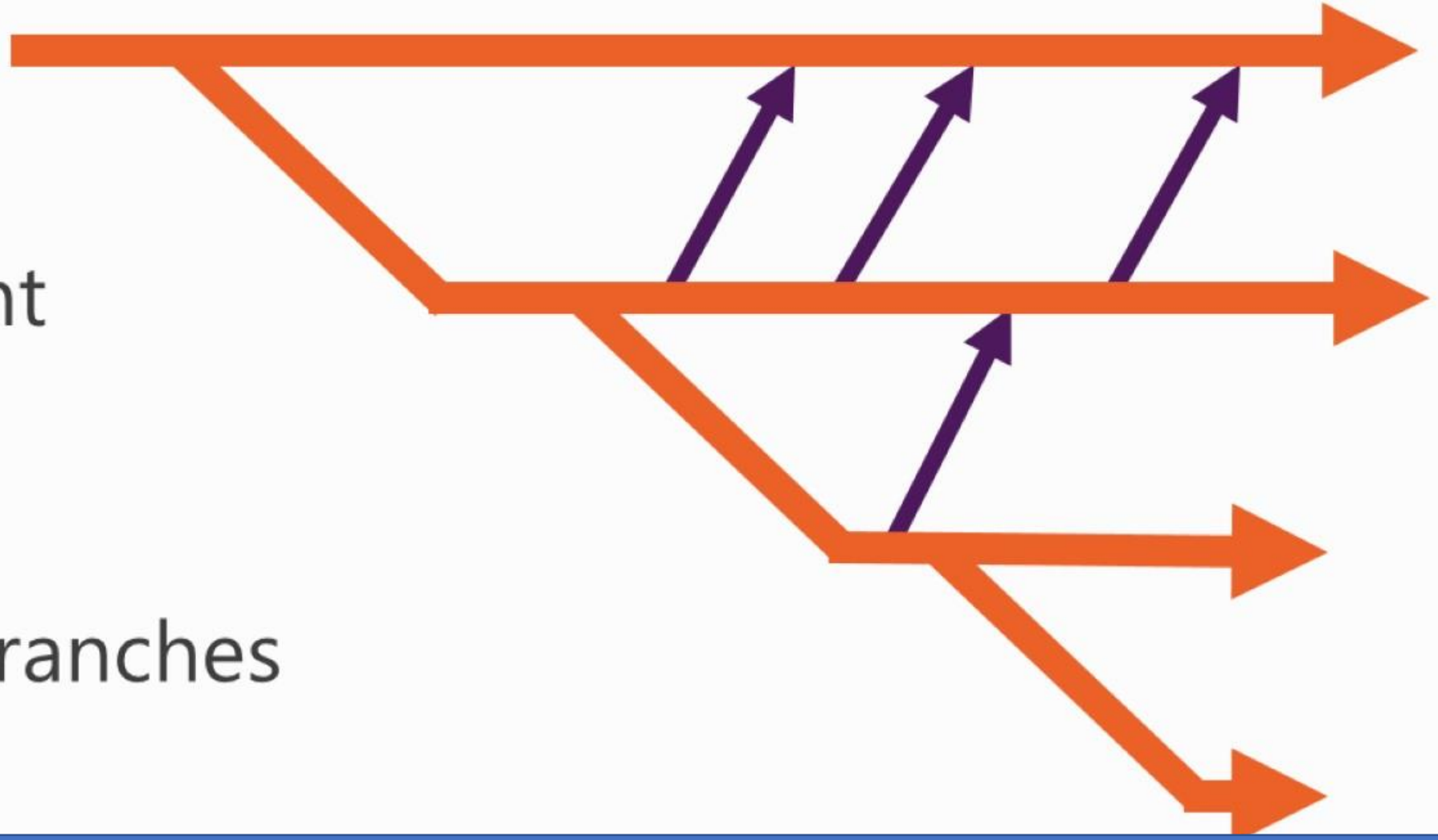


Git Branching and Merging

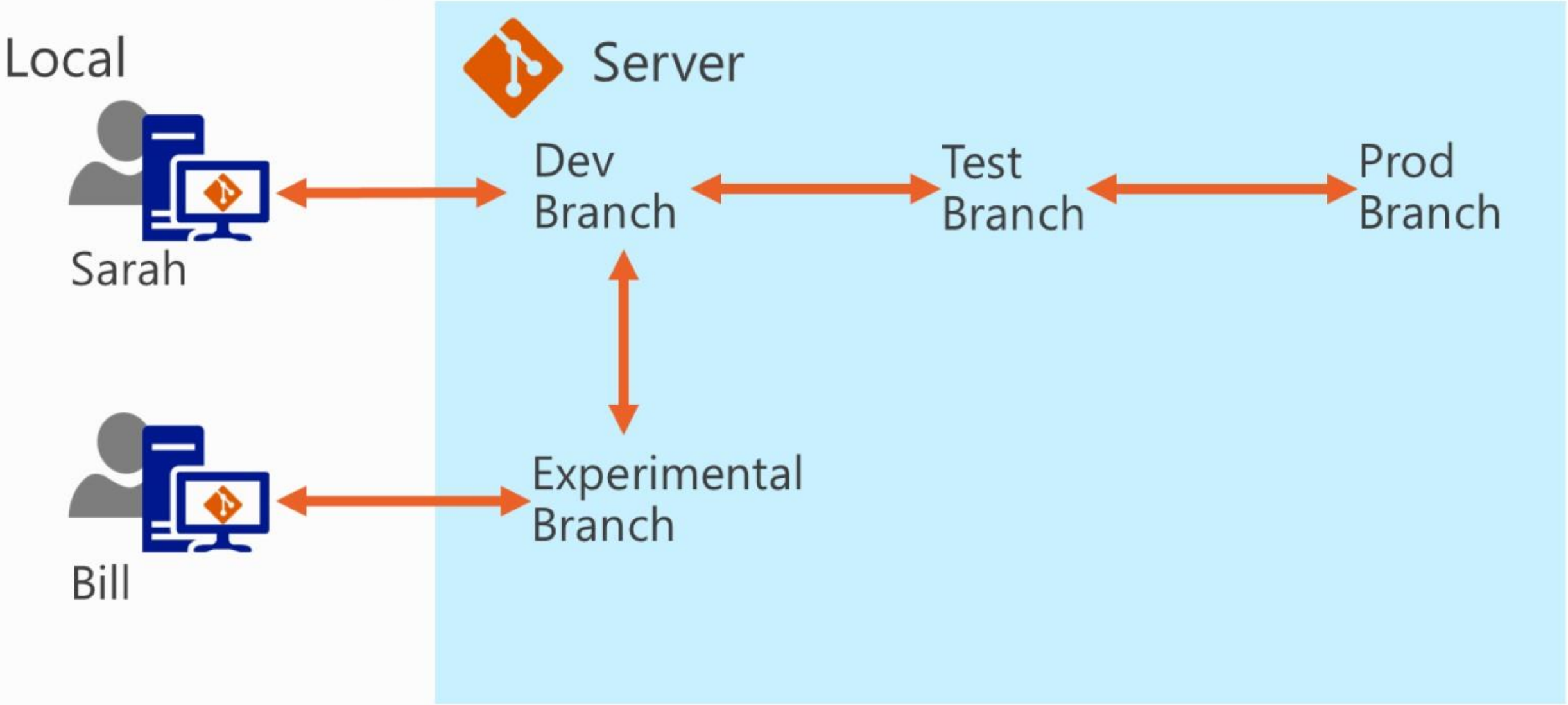
master

development

prototype
and other branches



Branching and Merging Lifecycle



Git Tooling

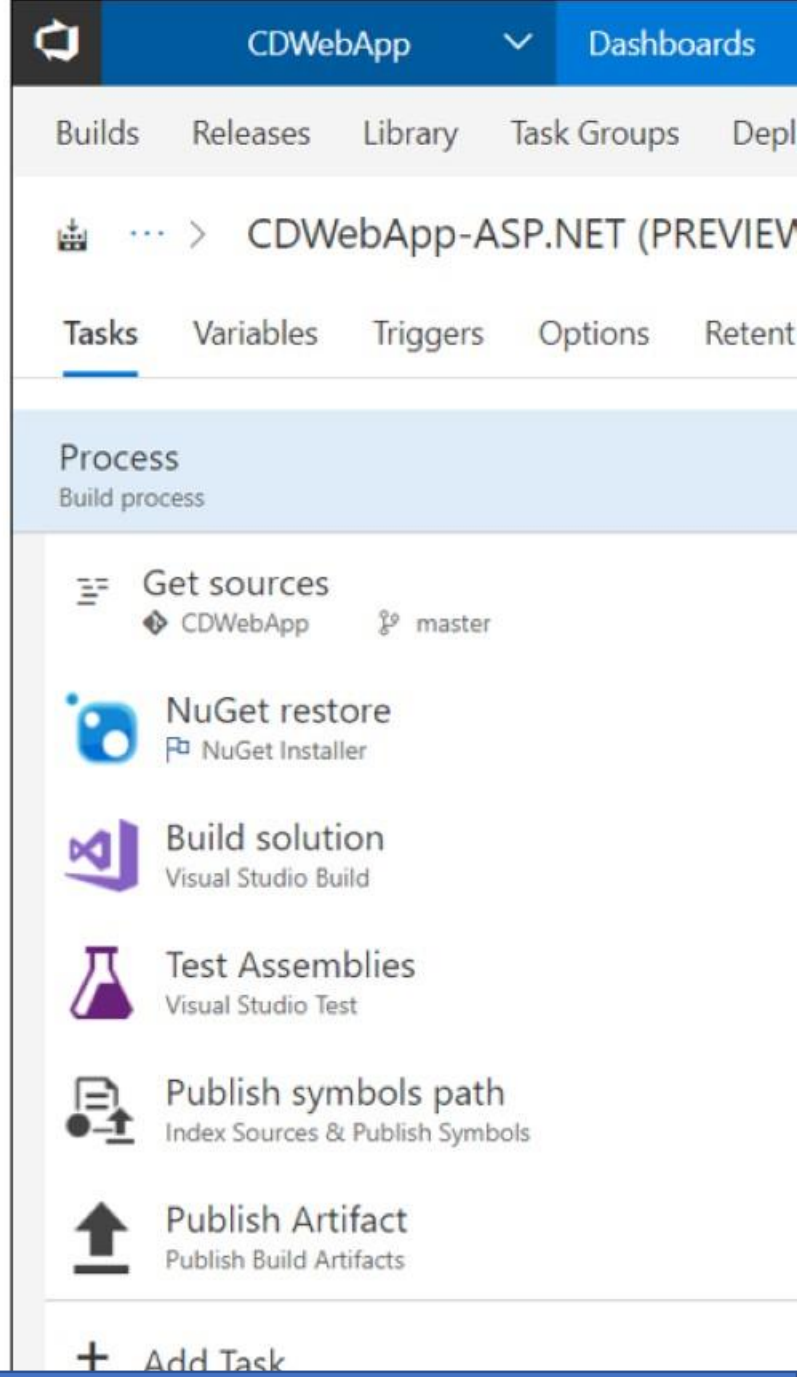
- Git Command-Line
- Visual Studio 2017
- Visual Studio Code
- Many other tools available
  - https://git-scm.com

```
git add .
git commit
git push
```

Build Automation

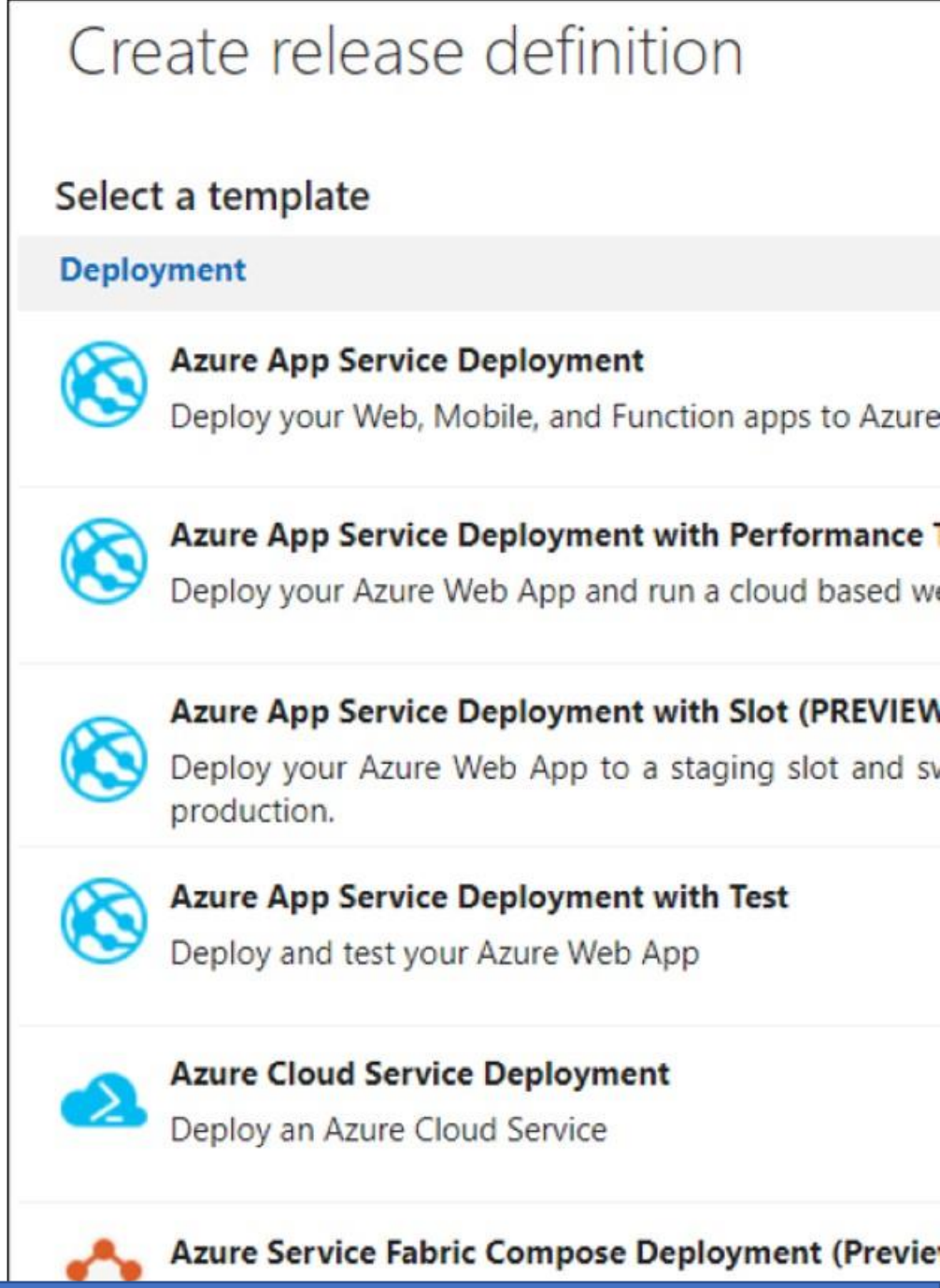
Build Definitions

- Defines the workflow steps to perform
  - Build code, Run Tests, Publish files, and many more…
- Trigger builds…
  - On a Schedule
  - When code is committed to the repository
- Build artifacts used for Release



Release Definitions

- Release the build to environment (Prod, Test, etc)
- Multi-platform support
- Trigger releases…
  - On a Schedule
  - On a Successful Build

Release Definitions

- Many deployment templates to choose from
  - Deploy to…
    - Azure App Service
    - Azure Cloud Service
    - Azure Service Fabric
    - and more…
- Define Custom Release Tasks too!