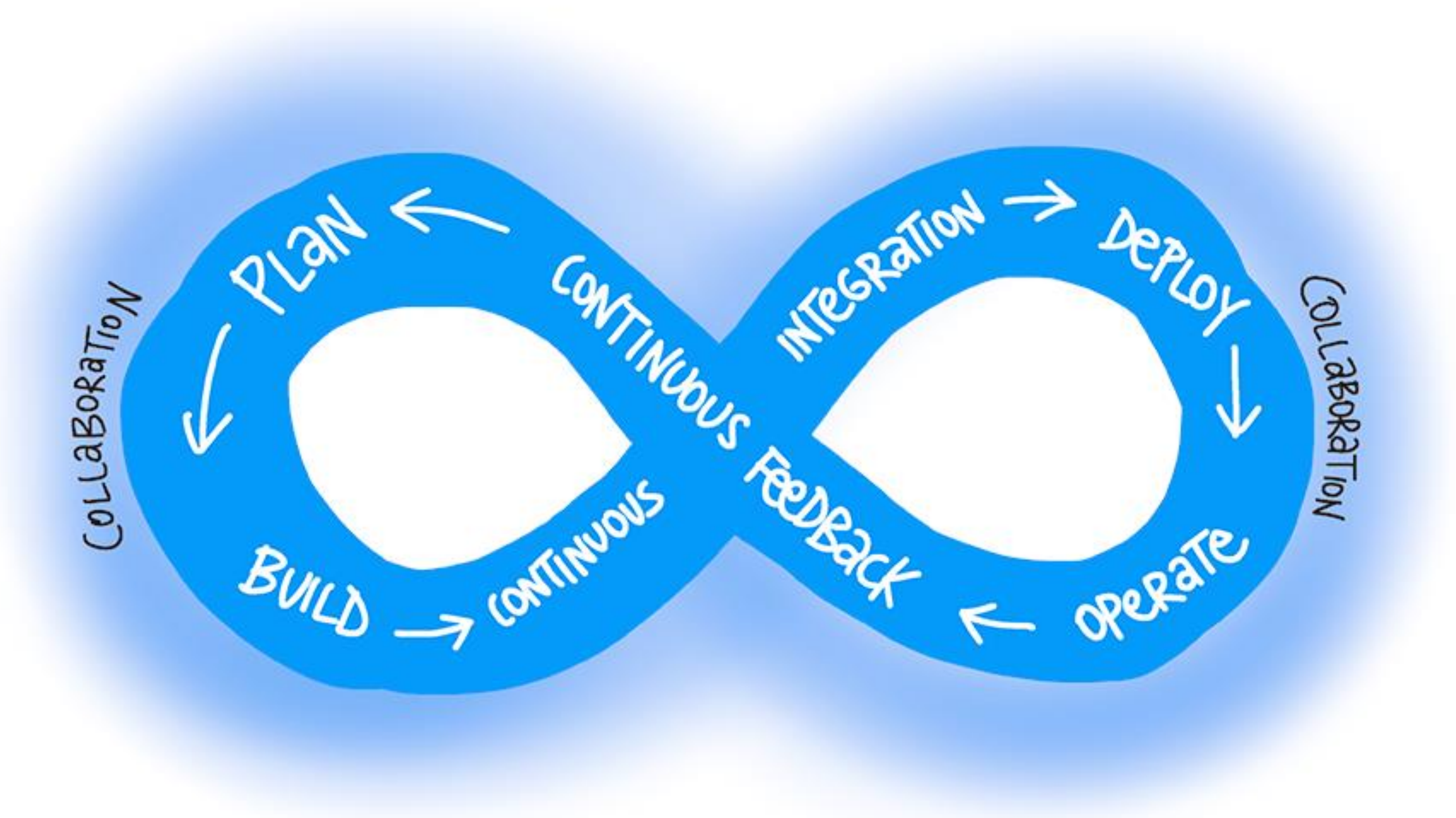
Microsoft Azure

Microsoft Azure
DevDays

Microsoft Azure

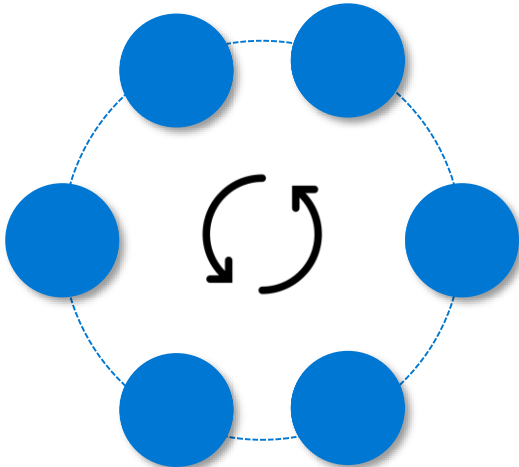
What is DevOps?

What is DevOps?



Why DevOps is needed?

- Before DevOps, the development and operation team worked in complete isolation
- Testing and Deployment were isolated activities done after design-build. Hence, they consumed more time than actual build cycles
- Without using DevOps, team members are spending a large amount of their time in testing, deploying, and designing instead of building the project
- Manual code deployment leads to human errors in production
- Coding and operation teams have their separate timelines and are not in synch causing further delays



What DevOps do?

DevOps helps to

Improve application life cycle management processes to make applications effective, more flexible than other methods of developing software. DevOps allows for the constant development of a product evolving it and refining it when it’s already live.

This Process helps to stay ahead to the companies by constantly adapting their products and releasing them onto the market much faster.

How is DevOps different from traditional IT

Old Process

- After placing an order for new servers, the Development team works on testing. The Operations team works on extensive paperwork as required in enterprises to deploy the infrastructure.

DevOps

- After placing an order for new servers Development and Operations team work together on the paperwork to set-up the new servers. This results in better visibility of infrastructure requirement.

How is DevOps different from traditional IT

Old Process

- Projection about failover, redundancy, data center locations, and storage requirements are skewed as no inputs are available from developers who have deep knowledge of the application.

DevOps

- Projection about failover, redundancy, disaster recovery, data centre locations, and storage requirements are accurate due to the inputs from the developers.

How is DevOps different from traditional IT

Old Process

- Operations team has no clue on the progress of the Development team. Operations team develop a monitoring plan as per their understanding.

DevOps

- The Operations team is completely aware of the progress the developers are making. Operations team interact with developers and jointly develop a monitoring plan that caters to the IT and business needs.

How is DevOps different from traditional IT

Old Process

- Before go-live, the load testing crashes the application. The release is delayed.

DevOps

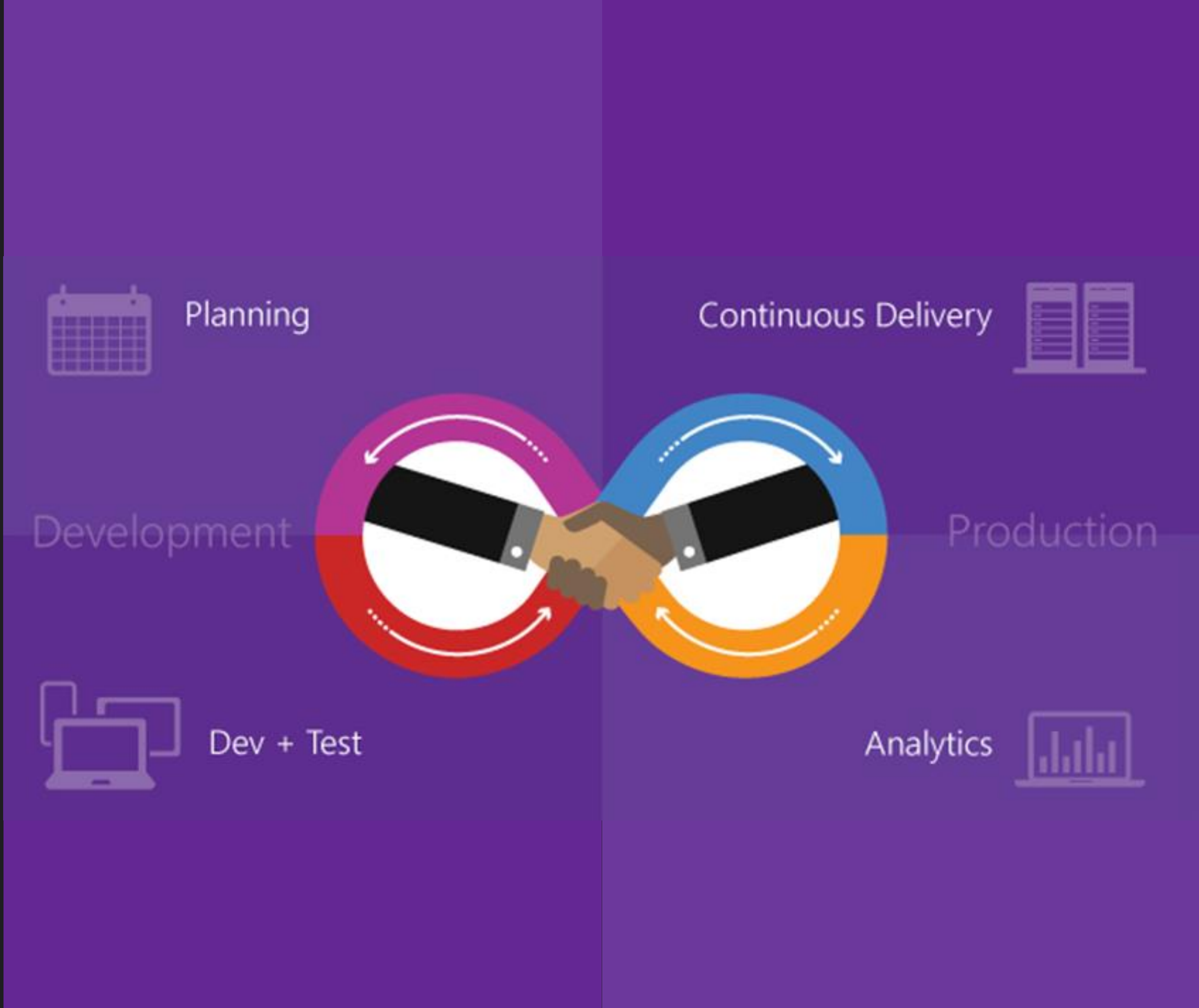
- Before go-live, the load testing makes the application a bit slow. The development team quickly fixes the bottlenecks. The application is released on time.

Why is DevOps used?

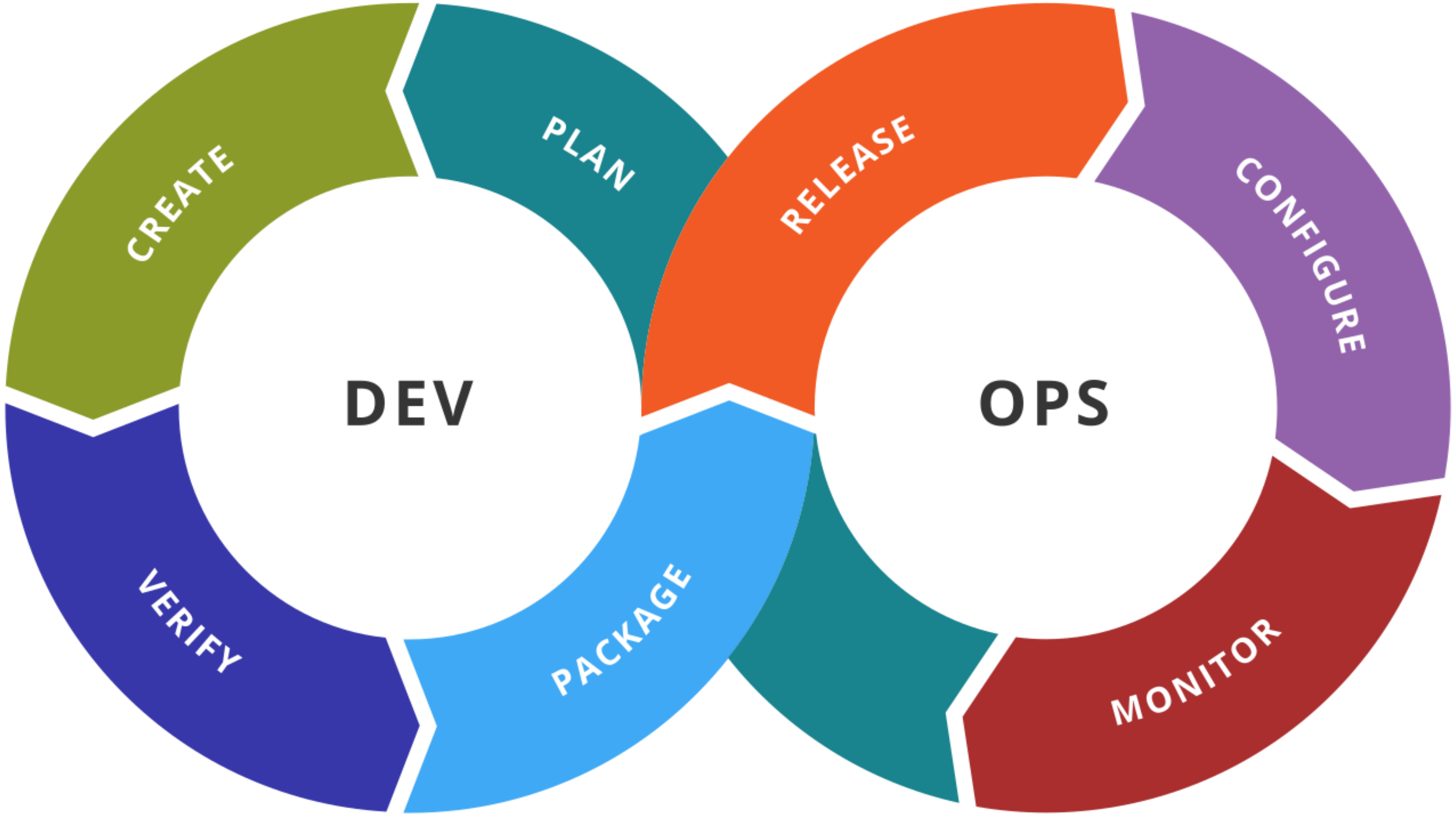
- Predictability
- Reproducibility
- Maintainability
- Time to market
- Greater Quality
- Reduced Risk
- Resiliency
- Cost Efficiency
- Breaks larger code base into small pieces

Microsoft Azure

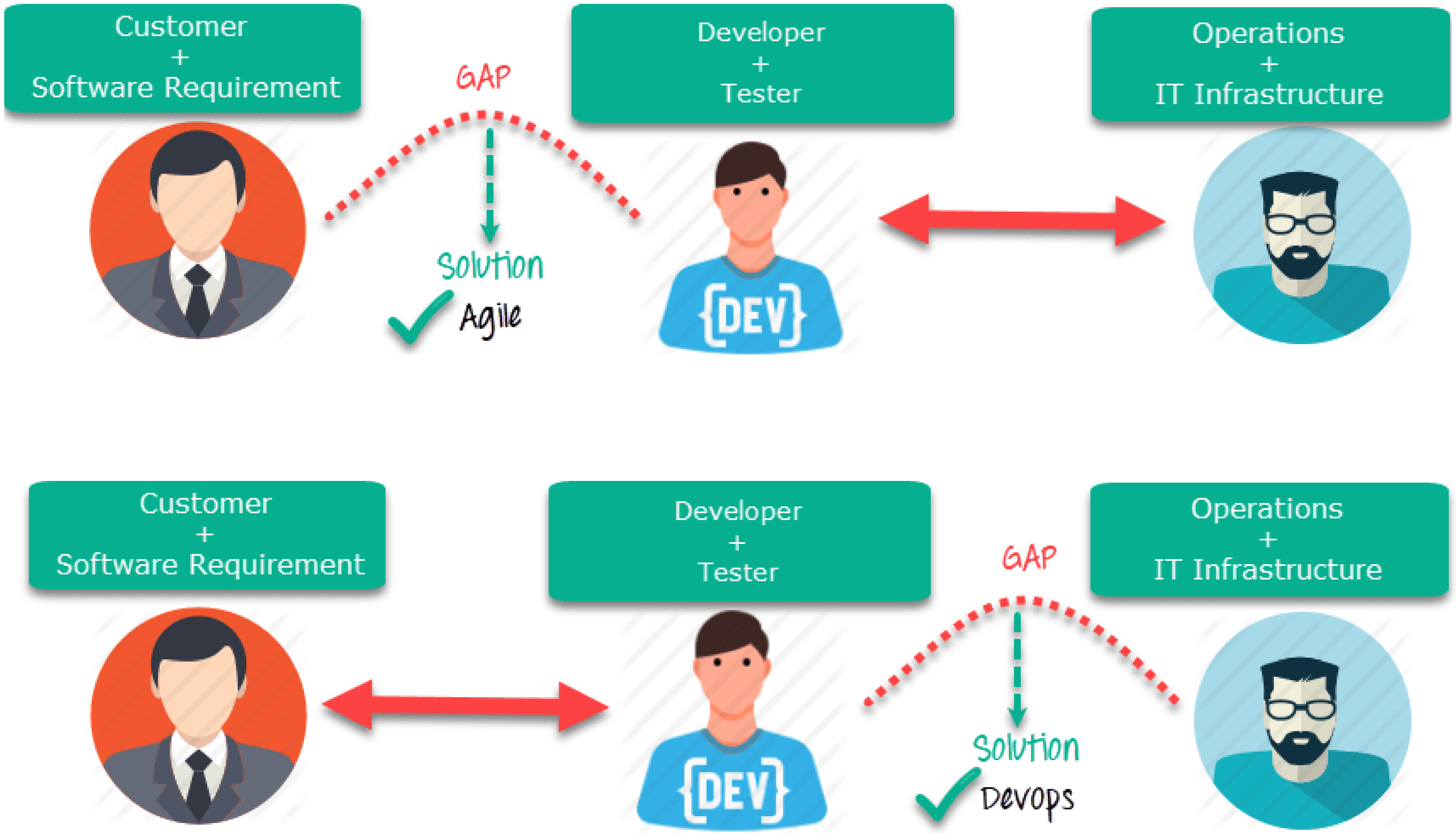
DevOps Lifecycle



DevOps Lifecycle



How is DevOps different from Agile?



Customer
+
Software Requirement

Developer
+
Tester

Operations
+
IT Infrastructure

Customer
+
Software Requirement

Developer
+
Tester

Operations
+
IT Infrastructure

Microsoft

Builds in DevOps

Manual Build

- Unseen manual steps often required
- Little or no build quality
- “Works on my machine”
- Slow feedback cycles
- Merge debt accumulates
- Builds and deploy moved to weekends
- Reluctance to try new technologies

VS

Continuous Integration

- Anyone can queue a build
- Consistency
- Includes test and build quality
- Fast feedback on build quality
- Merging code when fresh
- Increasing use of automated tests
- Faster check-in cadence

Build Automation and Continuous Integration

Automated builds are valuable because they:

Validate that code compilation doesn’t just succeed ”on my machine”.

Run as many task as needed, such as scripting, testing, packing, or anything else required.

Publish to a drop folder or network share to be picked up for deployment

Maintain an audit history for builds details, drop details, and associated work items.

In **DevOps, Automated Builds Are an Integral** Part of **Continuous Integration (CI)**

Continuous Integration

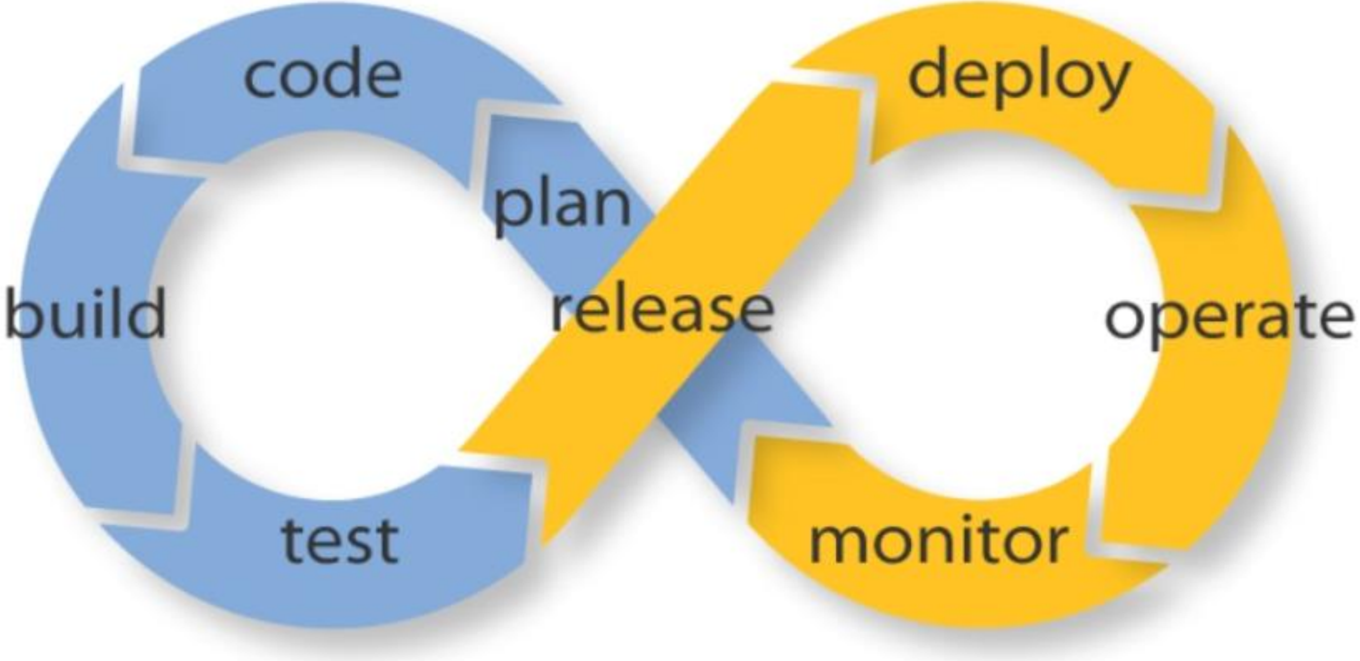
Is the practice of merging all developer working copies to a shared code line several times a day and *validating* each integration with an automated build.

CI *is often defined as having a* ***build*** *with* ***unit tests*** *that executes at every commit or check-in to version control.*

Continuous Integration CI provides many benefits:

1. Improving code quality based on rapid feedback.
2. Triggering for automated testing for every code change.
3. Better managing technical debt and conducting code analysis.
4. Reducing long, difficult and bug-inducing merges
5. Increasing confidence in code long before production.

Continuous Integration and Delivery



“Continuous improvement”

- Keep inventory at a minimum
- Minimize the queue of orders
- Maximize efficiency in the manufacturing process

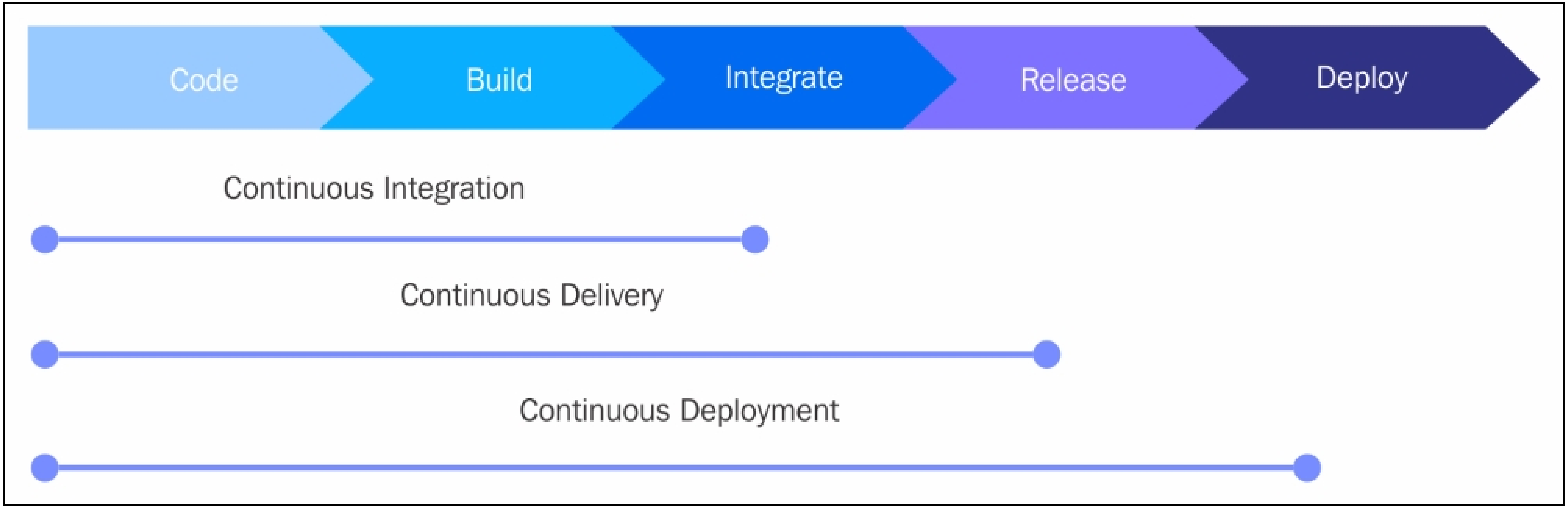
Continuous Integration (prerequisites)

- Using a version repository for source code
- Regular code check in schedule
- Automate testing for the code changes
- Automate the build
- Deploy build in preproduction

Continuous Integration (tools)

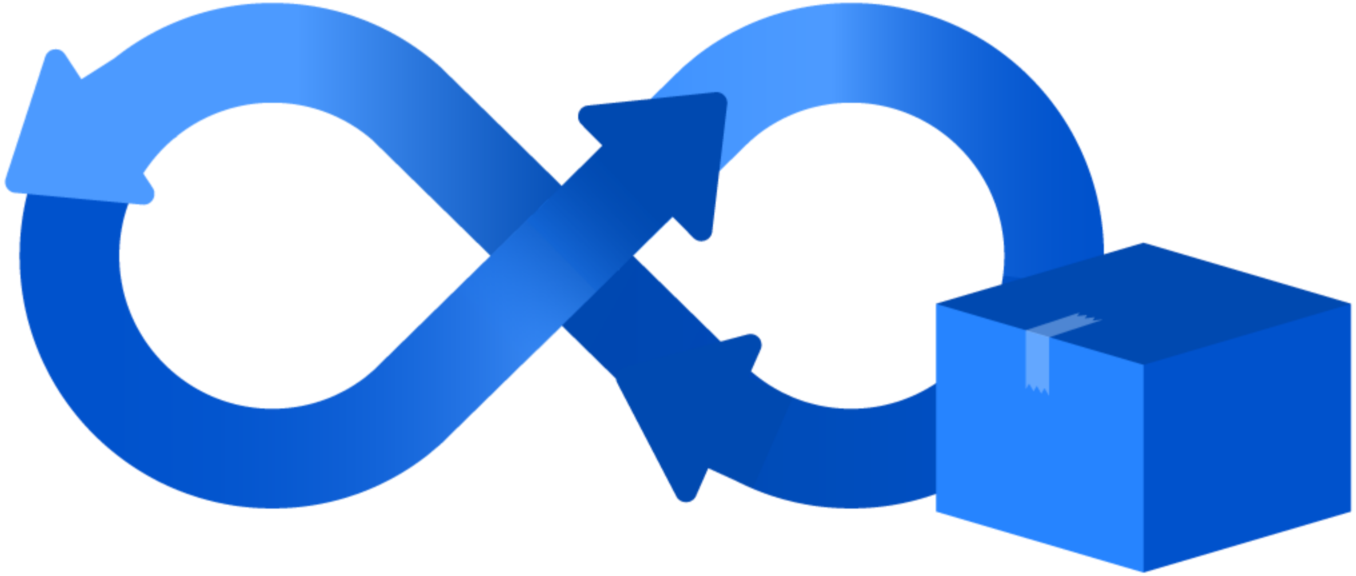
- Jenkins
- TeamCity
- Travis
- Go CD
- Buddy
- Bitbucket
- Chef
- Microsoft Teamcenter
- CruiseControl
- Bamboo
- GitLab CI
- CircleCI
- Codeship

Roles

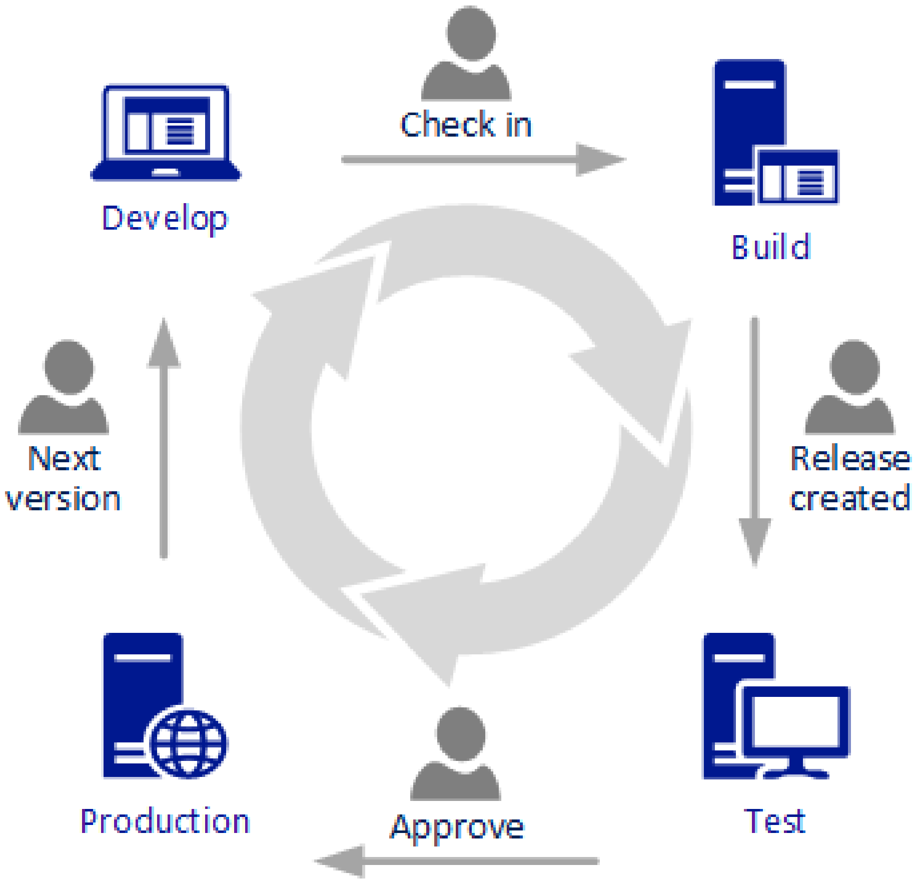

Code
Build
Integrate
Release
Deploy
Continuous Integration
Continuous Delivery
Continuous Deployment

Continuous Delivery

Continuous delivery is the next step of continuous integration in the software development cycle; it enables rapid and reliable development of software and delivery of product with the least amount of manual effort or overhead. In continuous integration, as we have seen, code is developed incorporating reviews, followed by automated building and testing.

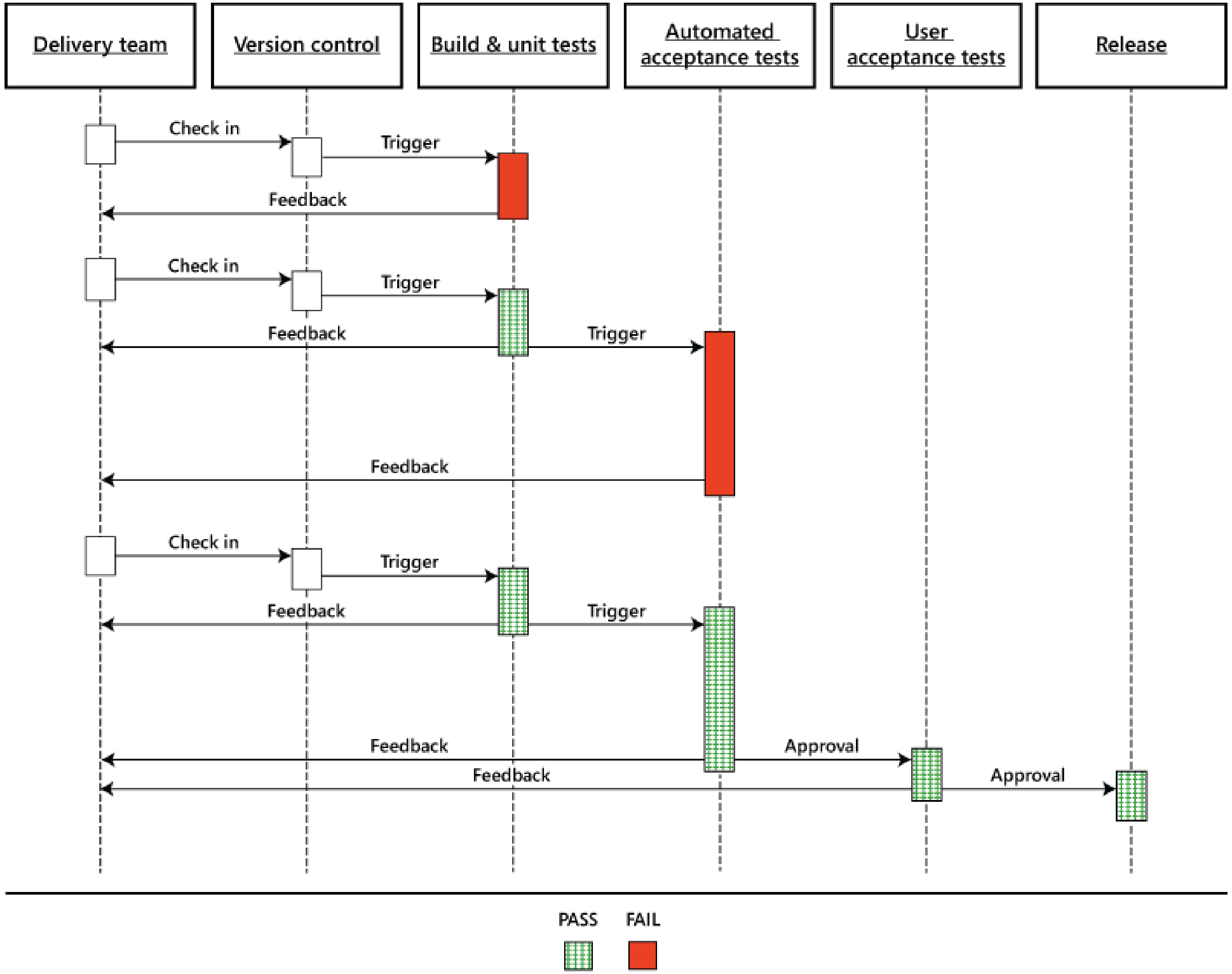


Continuous Delivery



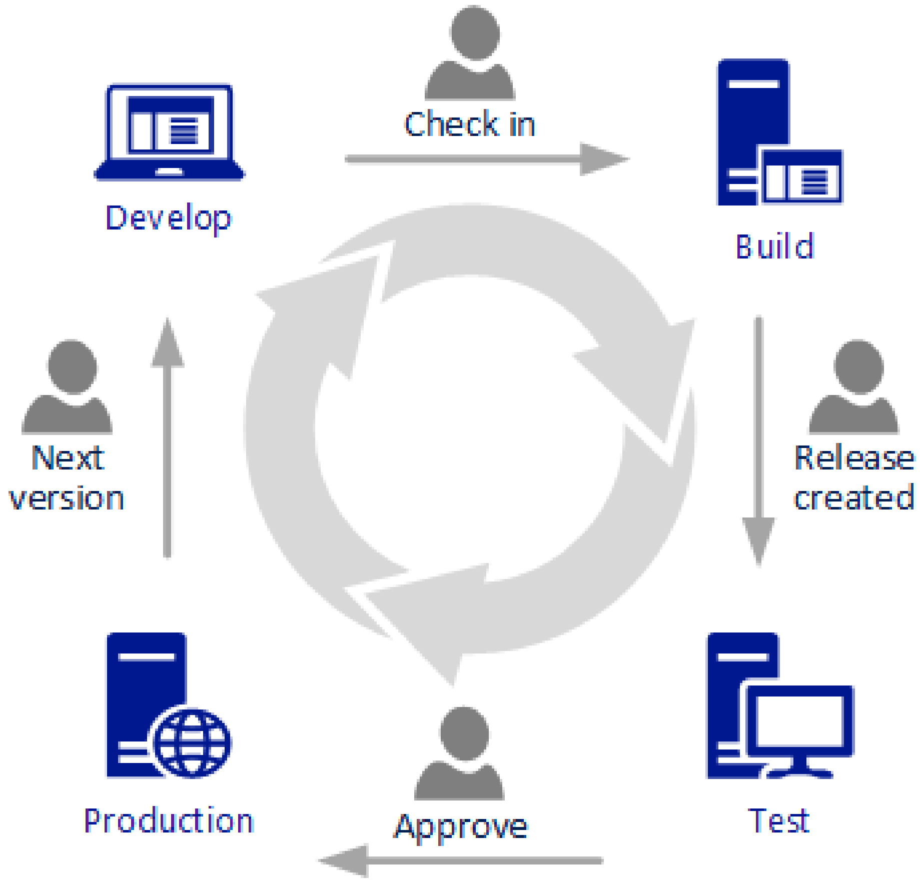
- Encourages Infrastructure as Code and configuration as code.
- Enables automated testing throughout the pipeline.
- Provides visibility and fast feedback cycles.
- Makes going to production a low-stress activity.
- Developed code is continuously delivered
- Code is constantly and regularly reviewed
- High-quality software is deployed rapidly, reliably, and repeatedly
- Maximum automation and minimal manual overhead

Conceptualizing a Release Pipeline



Continuous Deployment

Continuous deployment is the fully matured and complete process cycle of code change, passing through every phase of the software life cycle to be deployed to production environments.



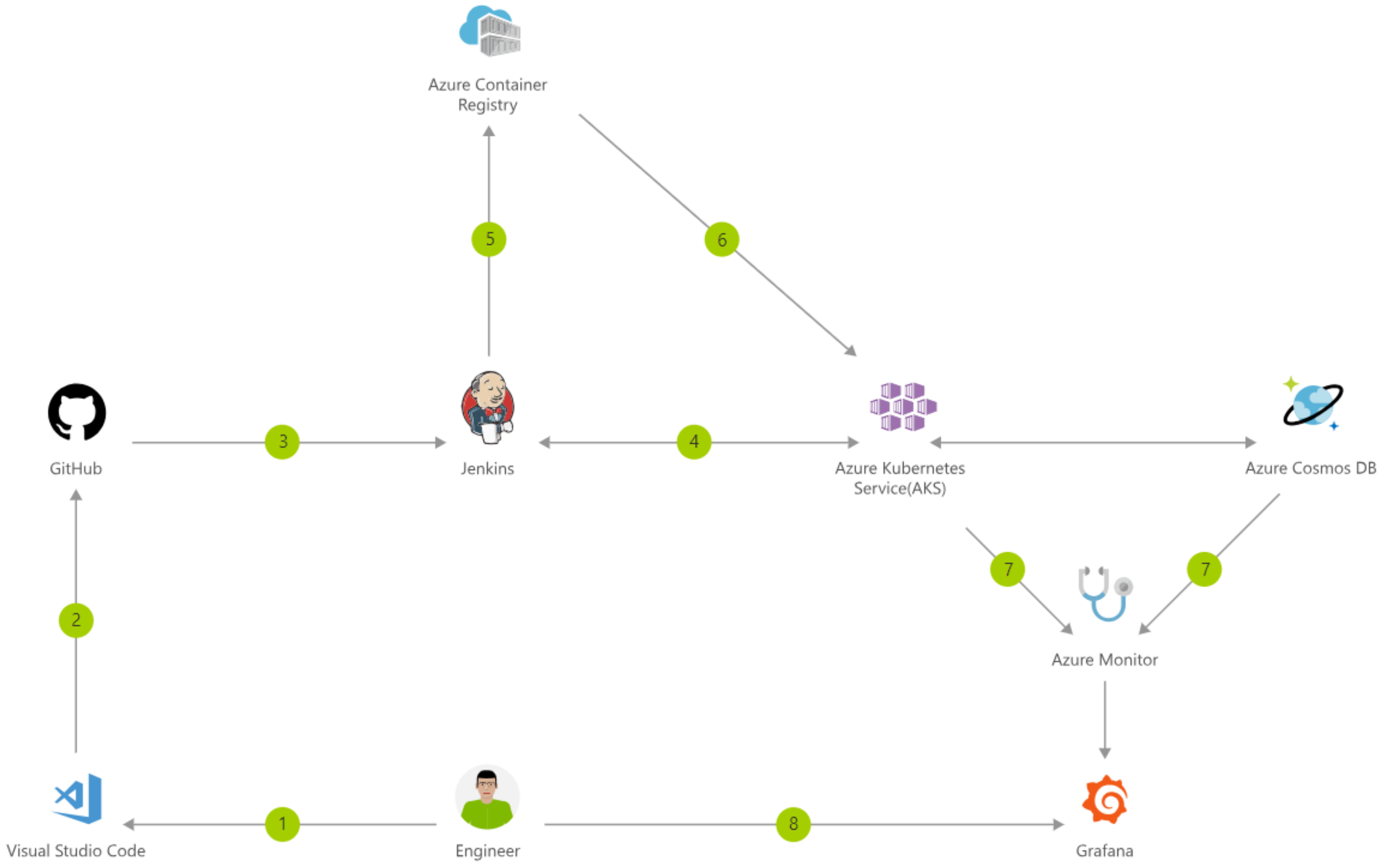
Continuous Deployment (benefits)

- Frequent product releases deliver software as fast as possible
- Automated and accelerated product releases with the code change
- Code changes qualify for production both from a technical and quality view point
- The most current version of the product is ready in shippable format

Continuous Deployment (benefits)

- Deployment modelling reduces errors, resulting in better product quality
- Consolidated access to all tools, process and resource data leads to quicker troubleshooting and time to market
- Effective collaboration between dev, QA, and operation teams leads to higher output and better customer satisfaction
- Facilitates lower audit efforts owing to a centralized view of all phase activities

Container CI/CD Using Jenkins and AKS



Infrastructure as Code (IaC)

Advantages of IaC

- The use of definition files and code to update system configuration is quick
- The version of all the code and changes is less error prone and has reproducible results
- Thorough testing of the deployment with IaC and test systems
- Smaller regular changes are easy to manage, bigger infrastructure updates are likely to contain errors that are difficult to detect
- Audit tracking and compliance are easy with definition files

Advantages of IaC

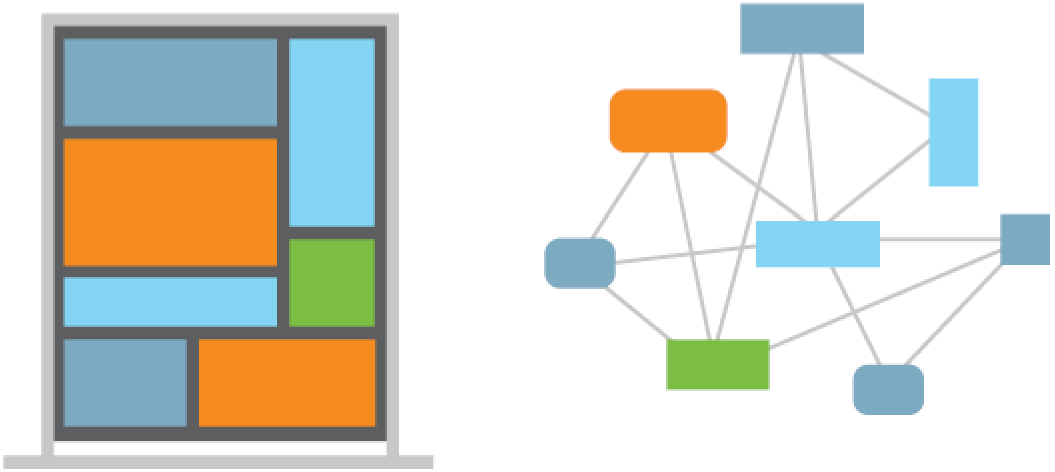
- Multiple servers update simultaneously
- System availability is high, with less down time
- Some tools for IaC are as follows:
- Ansible tower
- CFEngine
- Chef
- Puppet
- SaltStack

Microservices

What are Microservices?

Microservices describes the architectural pattern of composing a distributed application from separately deployable services that perform specific business functions and communicate over web interfaces.

DevOps teams encapsulate individual pieces of functionality in microservices and build larger systems by composing the microservices like building blocks

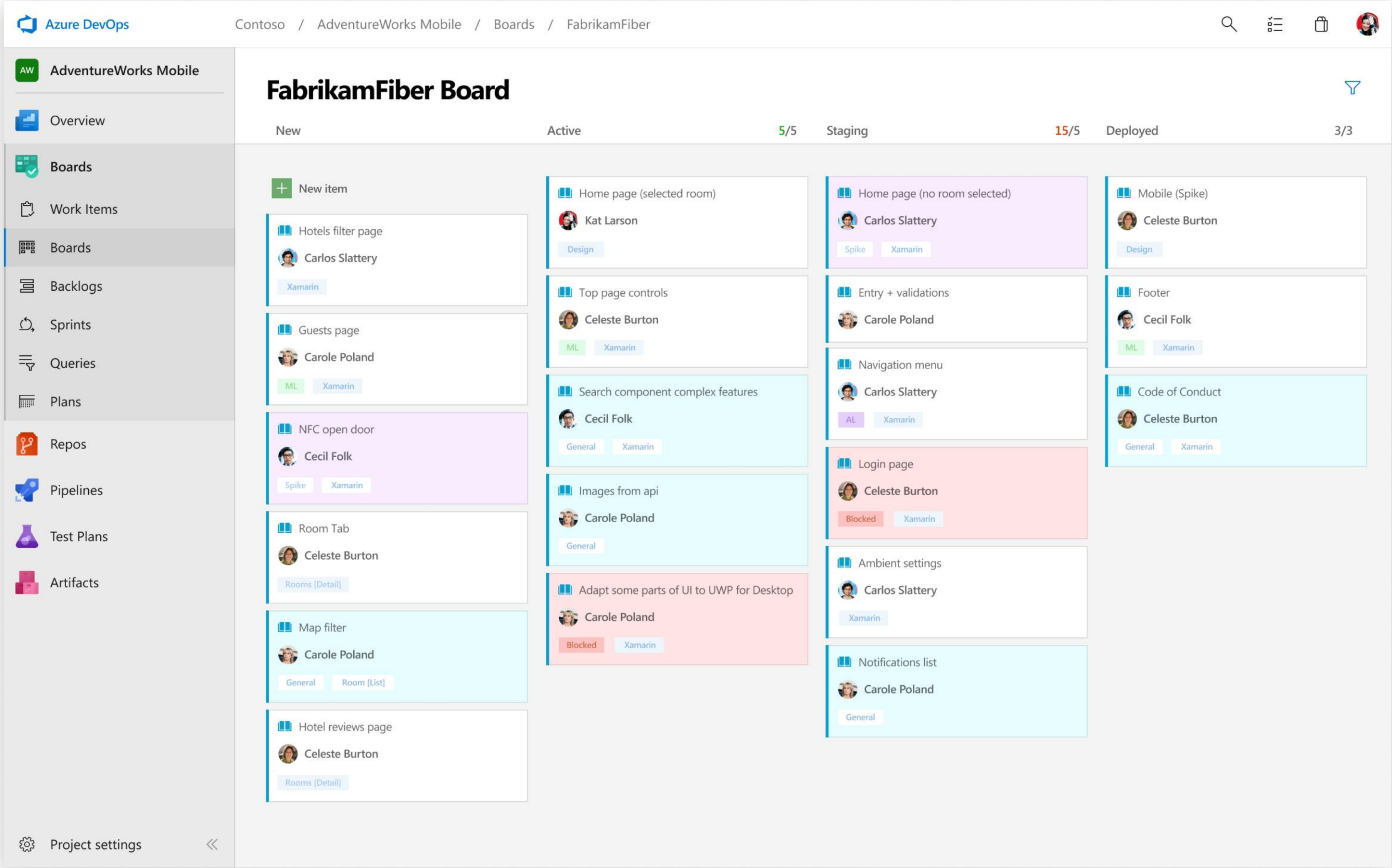


What are Microservices?

- Highly maintainable and testable
- Loosely coupled
- Independently deployable
- Organized around business capabilities.

Azure DevOps Service

Azure DevOps services – Azure Boards



FabrikamFiber Board

New

New item

Hotels filter page
Carlos Slattery
Xamarin

Guests page
Carole Poland
ML Xamarin

NFC open door
Cecil Folk
Spike Xamarin

Room Tab
Celeste Burton
Rooms [Detail]

Map filter
Carole Poland
General Room [List]

Hotel reviews page
Celeste Burton
Rooms [Detail]

Active 5/5

Home page (selected room)
Kat Larson
Design

Top page controls
Celeste Burton
ML Xamarin

Search component complex features
Cecil Folk
General Xamarin

Images from api
Carole Poland
General

Adapt some parts of UI to UWP for Desktop
Carole Poland
Blocked Xamarin

Staging 15/5

Home page (no room selected)
Carlos Slattery
Spike Xamarin

Entry + validations
Carole Poland

Navigation menu
Carlos Slattery
AL Xamarin

Login page
Celeste Burton
Blocked Xamarin

Ambient settings
Carlos Slattery
Xamarin

Notifications list
Carole Poland
General

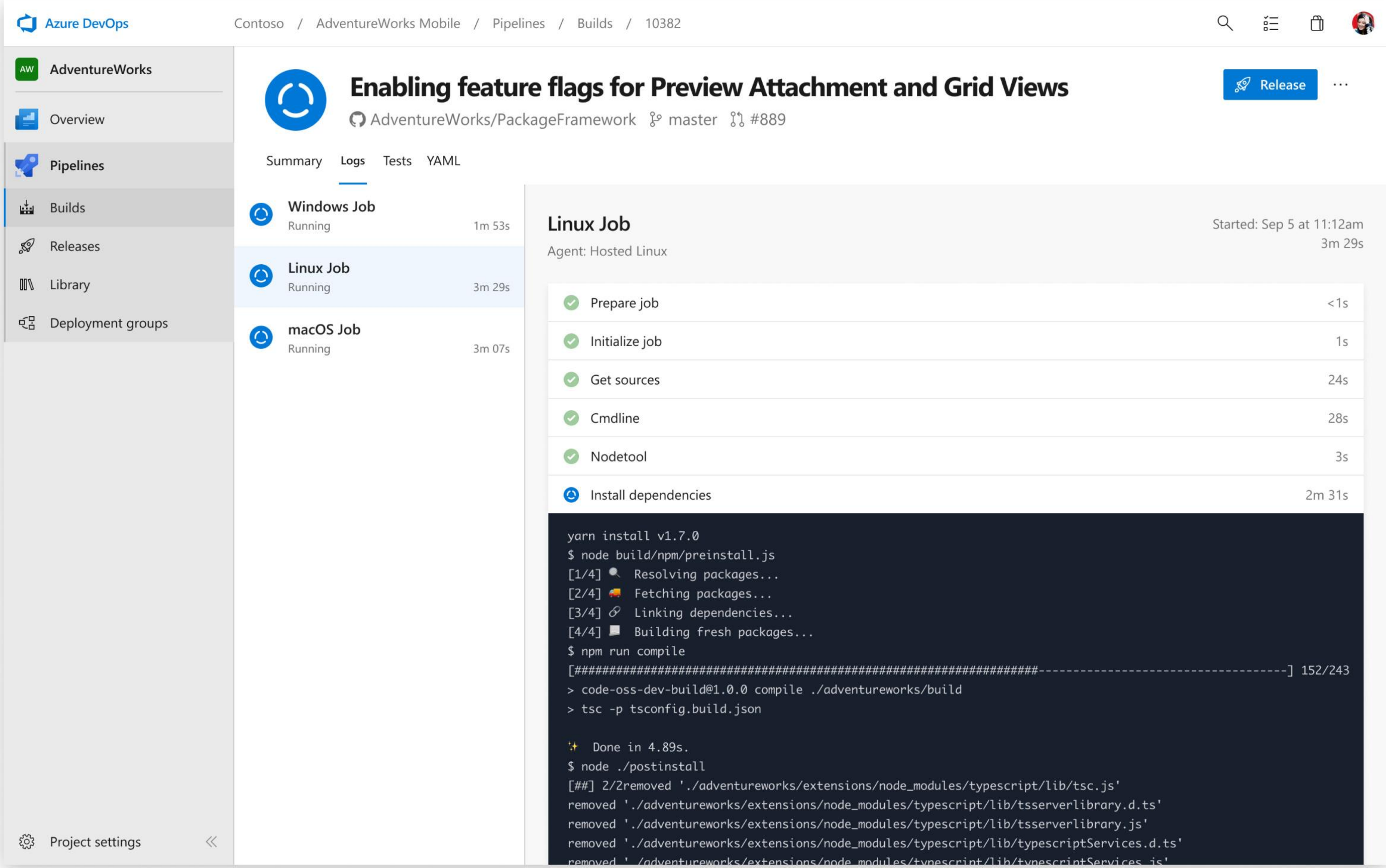
Deployed 3/3

Mobile (Spike)
Celeste Burton
Design

Footer
Cecil Folk
ML Xamarin

Code of Conduct
Celeste Burton
General Xamarin

Azure DevOps services – Azure Pipelines



Enabling feature flags for Preview Attachment and Grid Views

AdventureWorks/PackageFramework   master   #889

Windows Job	Running	1m 53s
Linux Job	Running	3m 29s
macOS Job	Running	3m 07s

Linux Job

Agent: Hosted Linux

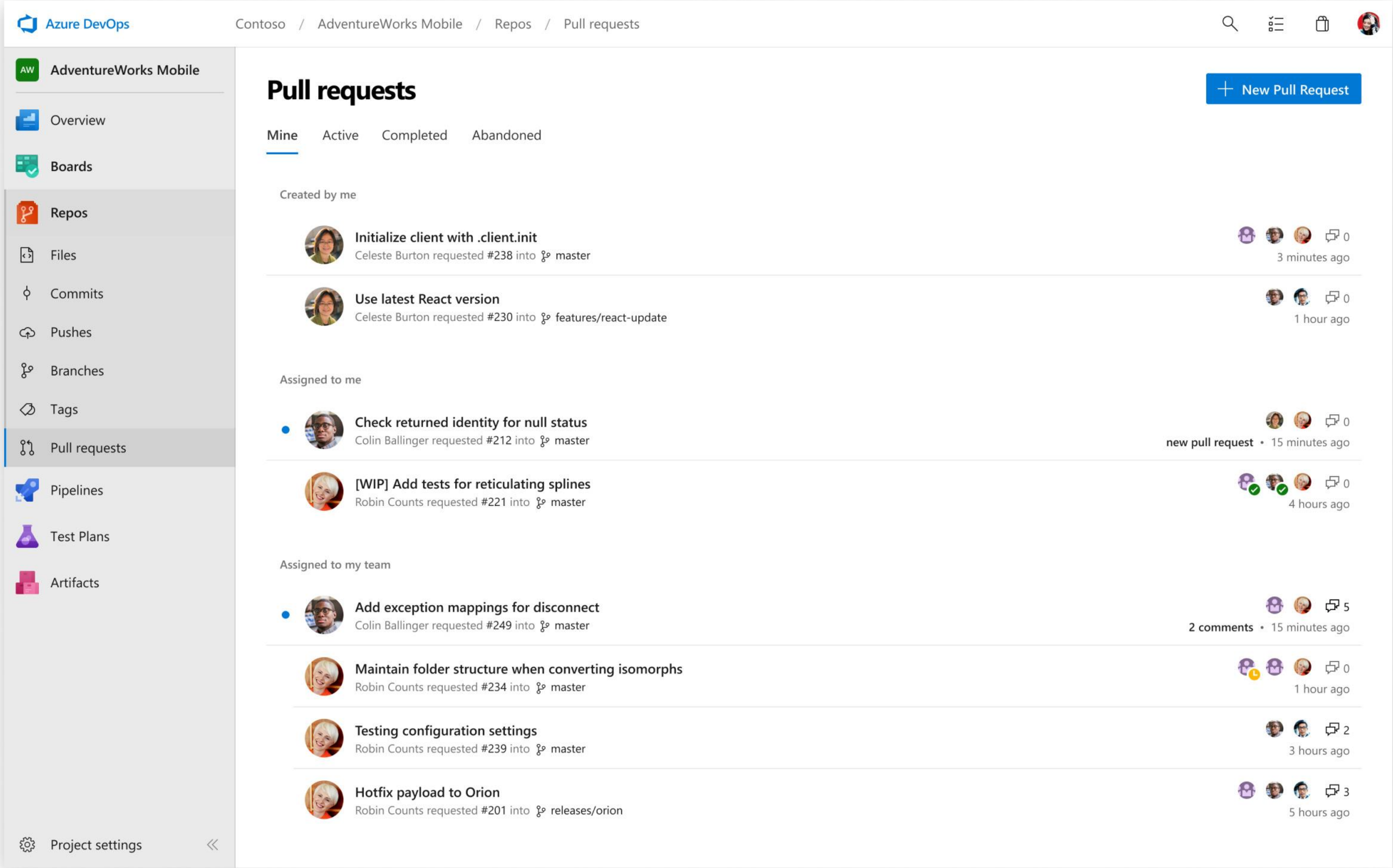
Started: Sep 5 at 11:12am
3m 29s

Prepare job	<1s
Initialize job	1s
Get sources	24s
Cmdline	28s
Nodetool	3s
Install dependencies	2m 31s

```
yarn install v1.7.0
$ node build/npm/preinstall.js
[1/4] 🔍  Resolving packages...
[2/4] 🚚  Fetching packages...
[3/4] 🔗  Linking dependencies...
[4/4] 📃  Building fresh packages...
$ npm run compile
[###################################################################------------------------------------] 152/243
> code-oss-dev-build@1.0.0 compile ./adventureworks/build
> tsc -p tsconfig.build.json

✨  Done in 4.89s.
$ node ./postinstall
[##] 2/2removed './adventureworks/extensions/node_modules/typescript/lib/tsc.js'
removed './adventureworks/extensions/node_modules/typescript/lib/tsserverlibrary.d.ts'
removed './adventureworks/extensions/node_modules/typescript/lib/tsserverlibrary.js'
removed './adventureworks/extensions/node_modules/typescript/lib/typescriptServices.d.ts'
removed './adventureworks/extensions/node_modules/typescript/lib/typescriptServices.js'
```

Azure DevOps services – Azure Repos



Pull requests

Created by me

Initialize client with .client.init
Celeste Burton requested #238 into master
0
3 minutes ago

Use latest React version
Celeste Burton requested #230 into features/react-update
0
1 hour ago

Assigned to me

Check returned identity for null status
Colin Ballinger requested #212 into master
0
new pull request • 15 minutes ago

[WIP] Add tests for reticulating splines
Robin Counts requested #221 into master
0
4 hours ago

Assigned to my team

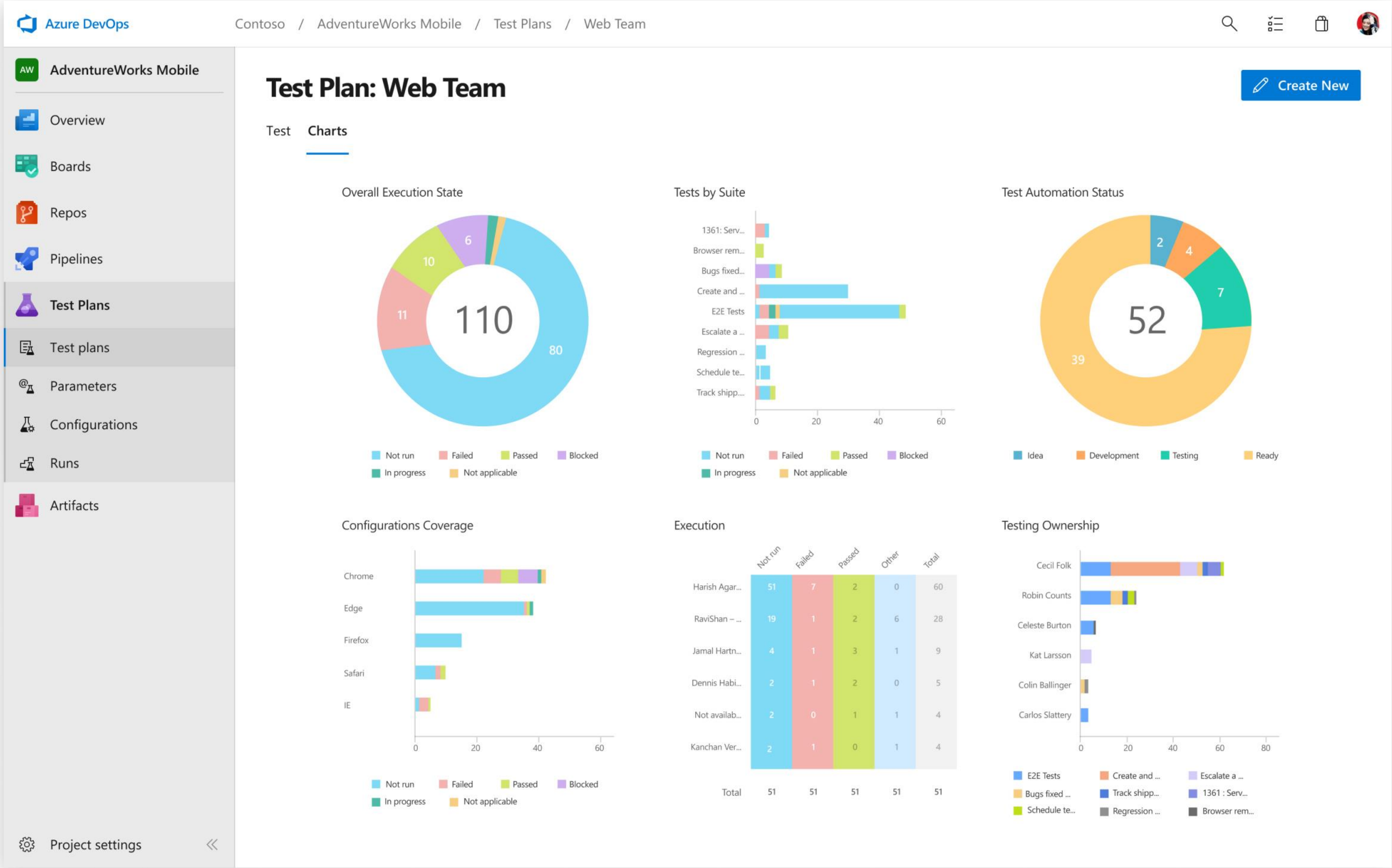
Add exception mappings for disconnect
Colin Ballinger requested #249 into master
5
2 comments • 15 minutes ago

Maintain folder structure when converting isomorphs
Robin Counts requested #234 into master
0
1 hour ago

Testing configuration settings
Robin Counts requested #239 into master
2
3 hours ago

Hotfix payload to Orion
Robin Counts requested #201 into releases/orion
3
5 hours ago

Azure DevOps services – Azure Test Plans

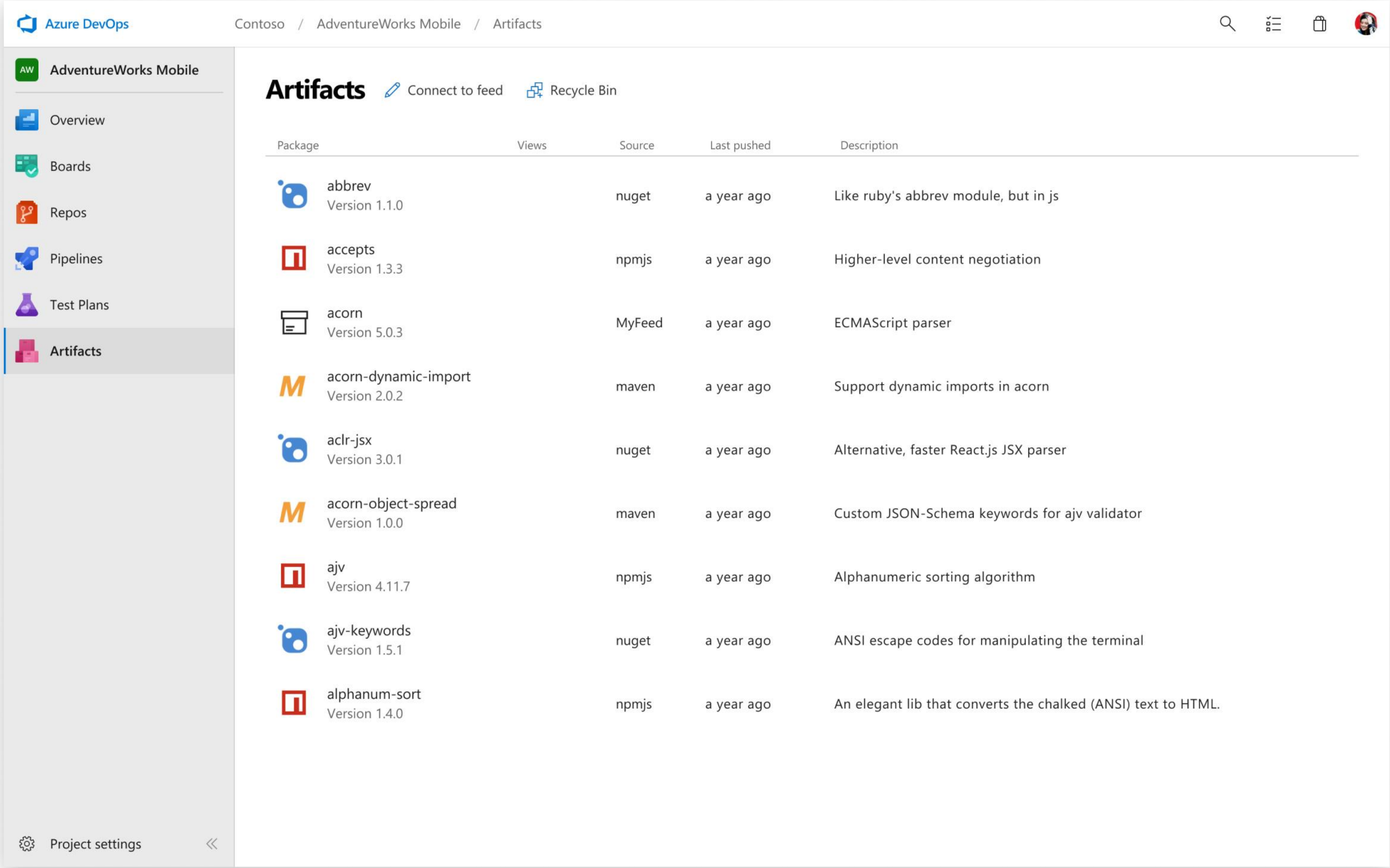


Test Plan: Web Team

Execution

	Not run	Failed	Passed	Other	Total
Harish Agar...	51	7	2	0	60
RaviShan – ...	19	1	2	6	28
Jamal Hartn...	4	1	3	1	9
Dennis Habi...	2	1	2	0	5
Not availab...	2	0	1	1	4
Kanchan Ver...	2	1	0	1	4
Total	51	51	51	51	51

Azure DevOps services – Azure Artifacts



Artifacts

Package	Views	Source	Last pushed	Description
abbrev Version 1.1.0		nuget	a year ago	Like ruby's abbrev module, but in js
accepts Version 1.3.3		npmjs	a year ago	Higher-level content negotiation
acorn Version 5.0.3		MyFeed	a year ago	ECMAScript parser
acorn-dynamic-import Version 2.0.2		maven	a year ago	Support dynamic imports in acorn
aclr-jsx Version 3.0.1		nuget	a year ago	Alternative, faster React.js JSX parser
acorn-object-spread Version 1.0.0		maven	a year ago	Custom JSON-Schema keywords for ajv validator
ajv Version 4.11.7		npmjs	a year ago	Alphanumeric sorting algorithm
ajv-keywords Version 1.5.1		nuget	a year ago	ANSI escape codes for manipulating the terminal
alphanum-sort Version 1.4.0		npmjs	a year ago	An elegant lib that converts the chalked (ANSI) text to HTML.

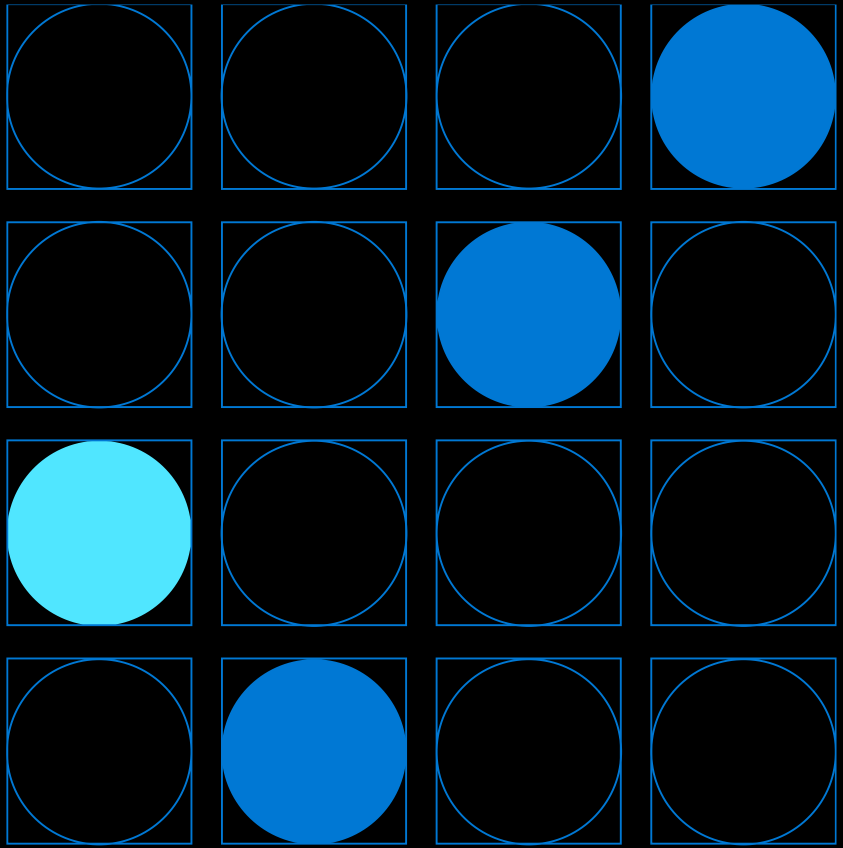
Microsoft

Thank You

Get the Latest...

Microsoft Azure

- Training & Documentation: microsoft.com/learn
- Events & Resources: aka.ms/usdevskills
- Developer newsletter: aka.ms/devsubscribe



facebook.com/MSFTDev.US @MSDEV

Developer Online Immersions

aka.ms/onlinecloudimmersions

For Developers who are getting started in Azure

- ✓ 40 courses running March – June
- ✓ 2-3 hours in Length
- ✓ Instructor Guided Hands-on Labs

3 New Developer titles

1. Deploy a Web Site to Microsoft Azure with Azure App Service

Join us for this live, facilitator-led immersive experience where you will learn how to build and publish Web apps in Microsoft Azure. During this interactive session, you will learn how to create a website through the hosted web app platform in Azure App Service. You will learn, hands-on, how to publish and manage your website easily, without having to work with the underlying servers, storage, or network assets.

2. Create Serverless applications

Join us for this live, facilitator-led immersive experience where you will learn how to build serverless architectures leveraging Azure functions to execute server-side logic. Experience hands-on how Azure functions enable the creation of event driven, compute-on-demand systems that can be triggered by various external events. You will also learn how to chain functions together through bindings to declare the data sources to read and write and let Azure Functions take care of the rest.

3. Getting Started with Azure DevOps and Containers:

Join us for this live, facilitator-led immersive experience where you will learn how Azure DevOps can help you release higher quality, well tested applications, on a shorter release cycle. Part one will focus on creating an Azure DevOps project, exploring the tool suite and learning how Azure DevOps works with your favorite open source tools you use today. Part II focuses on getting you hands on with Containers. Azure Container Instances are the quickest and easiest way to run containers in Azure. This immersion experience will teach you how to create and manage your containers, and how ACI can be used to provide elastic scale for Kubernetes

Dev Days *Deal*

25% Off Developer Courses

Azure
DevOps Engineer

Course title / training titles*	Course ID
Develop Azure Infrastructure as a Service compute Solutions	AZ-203T01-A
Develop Azure Platform as a Service compute solutions	AZ-203T02-A
Develop for Azure storage	AZ-203T03-A
Implement Azure security	AZ-203T04-A
Monitor, troubleshoot, and optimize Azure Solutions	AZ-203T05-A
Connect to and consume Azure, and third-party, services	AZ-203T06-A

Azure
Developer

Course title / training titles	Course ID
Implement DevOps development processes	AZ-400T01
Implement continuous integration	AZ-400T02
Implement continuous delivery	AZ-400T03
Implement dependency management	AZ-400T04
Implement application infrastructure	AZ-400T05
Implement continuous feedback	AZ-400T06
Design a DevOps strategy	AZ-400T07

directions