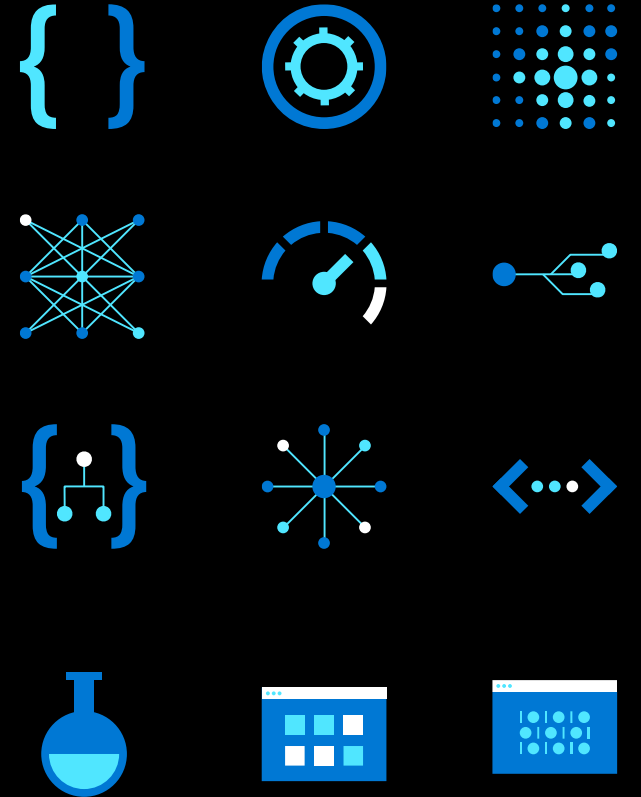


Azure Training Day

Run cloud-native apps with Azure Kubernetes Service



Deploy an app using DevOps and Kubernetes

Part 4 of 4 in the Run cloud-native apps with Azure Kubernetes Service series

About us...

Dave Burnison

 Sr. Specialist

For questions or help with this series

MSUSDev@Microsoft.com

For the lab guides and sample code

<https://github.com/MSUSDEV/Run-Cloud-Native-Apps-With-AKS>

Setting the scene



Overview of the workshop

About the workshop content...

About:

This series is the second half of a longer workshop that teaches how to build a proof of concept (POC) that will transform an existing ASP.NET-based Web application (SimplCommerce) to a container-based application. You can register to view the modules from the first half at <https://aka.ms/web-app-series> You can find all the presentations form the first half at <https://github.com/MSUSDEV/Migrating-web-apps-to-Azure>

At the end of this workshop, you will have a good understanding of container concepts, Docker architecture and operations, Azure Container Services, Azure Kubernetes Services and Azure DevOps tools.

Target Audience:

The workshop is targeted to Cloud Architects, Cloud Solution designers, developers and IT sysadmins, CIO's, CTO's and anybody else who is interested in learning about Azure, containers, application cloud migration and digital transformation.

Focus of the workshop (40%) is getting hands-on experience, complemented with presentations and whiteboard sessions (if in-person delivery).

Time Estimate:

11 hours (+/- 5 hours presentations, 6 hours of optional hands-on labs for attendees)

Workshop Agenda - Presentations

What we will talk about...

Series 1: <https://aka.ms/web-app-series>

- Module 1: Digital App Transformation with Azure
- Module 2: Running Azure Infrastructure and execute Lift & Shift Migrations
- Module 3: Performing proper assessments to smooth Azure Migrations
- Module 4: Why and how migrating databases to Azure PaaS
- Module 5: Migrating to Azure App Services – Azure Web Apps (.NET)

Series 2: <https://aka.ms/cloud-native-series>

- Module 1: Deploying Containers on Azure
- Module 2: Deploying Azure Kubernetes Services
- Module 3: Optimizing Azure Operations and Monitoring
- Module 4: Introduction to Azure DevOps **YOU ARE HERE**

Workshop Agenda – Hands On Labs

From series 1

- **Module 2: Running Azure Infrastructure and execute Lift & Shift Migrations**
 - *Lab 1: Deploy an Azure VM Infrastructure using ARM-Templates*
- **Module 3: Performing proper assessments to smooth Azure Migrations**
 - *Lab 2: Using Azure assessment tools*
- **Module 4: Why and how migrating databases to Azure PaaS**
 - *Lab 3: Migrating SQL Databases to Azure using Database Migration Assistant*
- **Module 5: Migrating to Azure App Services – Azure Web Apps (.NET)**
 - *Lab 4: Publishing application source code to Azure Web Apps using Visual Studio 2019*

Workshop Agenda – Hands On Labs

For this series 2

- **Module 1: Deploying Containers on Azure**
 - *Lab 5: Containerizing applications using Docker and running it in Azure Container Instance and Azure WebApp for Containers*
- **Module 2: Deploying Azure Kubernetes Services**
 - *Lab 6: Deploying Azure Kubernetes Services and running containerized apps from Azure Container Registry*
- **Module 3: Optimizing Azure Operations and Monitoring**
 - *Lab 7: Monitoring and Managing your Azure deployed workloads*
- **Module 4: Introduction to Azure DevOps **YOU ARE HERE****
 - *Lab 8: Deploying Azure DevOps with CI/CD Pipelines and deploy your applications to Azure WebApps, WebApp for Containers, Azure Container Instance and Azure Kubernetes Services*

Technical Requirements

What you need...

- See appendix slides for lab dependencies and / or alternate path for workshop
- Client workstation running recent Windows, Linux or Mac OS and latest internet browser
- Access to ports 80 (HTTP), 443 (HTTPS) and 3389 (Remote Desktop)
- Full Azure subscription (MSDN, AzurePass, Paid subscription, AE, CSP,...), where you have Owner permissions on subscription level
- Lab consumption estimate: \$15-35

Questions and HOL support

For questions or help with this series

MSUSDev@Microsoft.com

For the lab guides and sample code

<https://github.com/MSUSDEV/Run-Cloud-Native-Apps-With-AKS>

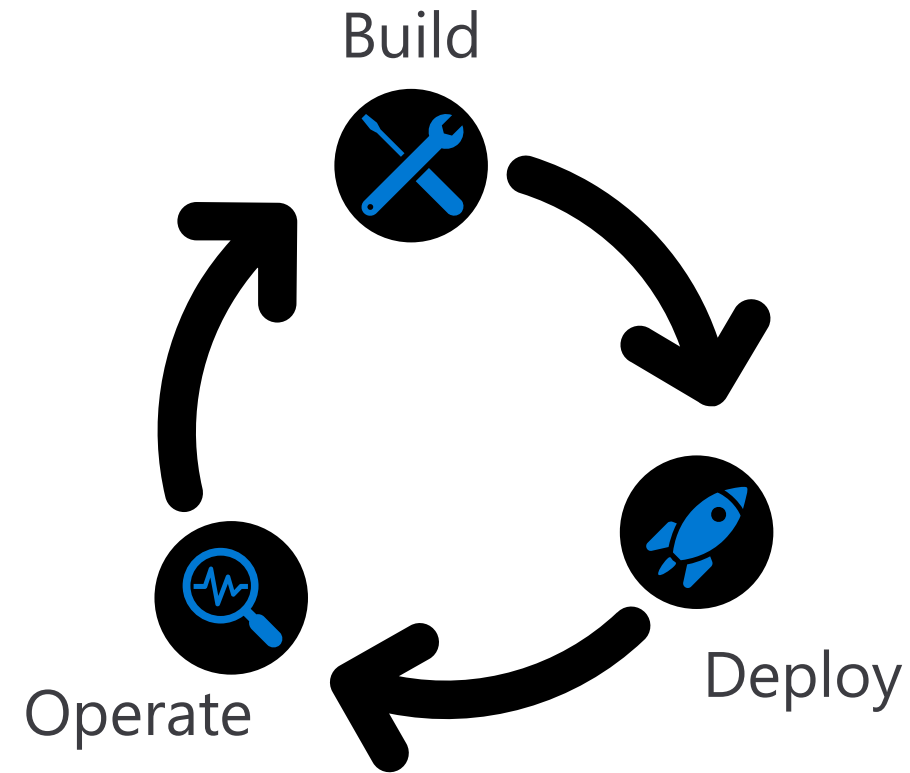
For information about lab dependencies and alternate approach please see the appendix slides at the end of this presentation.

Azure DevOps

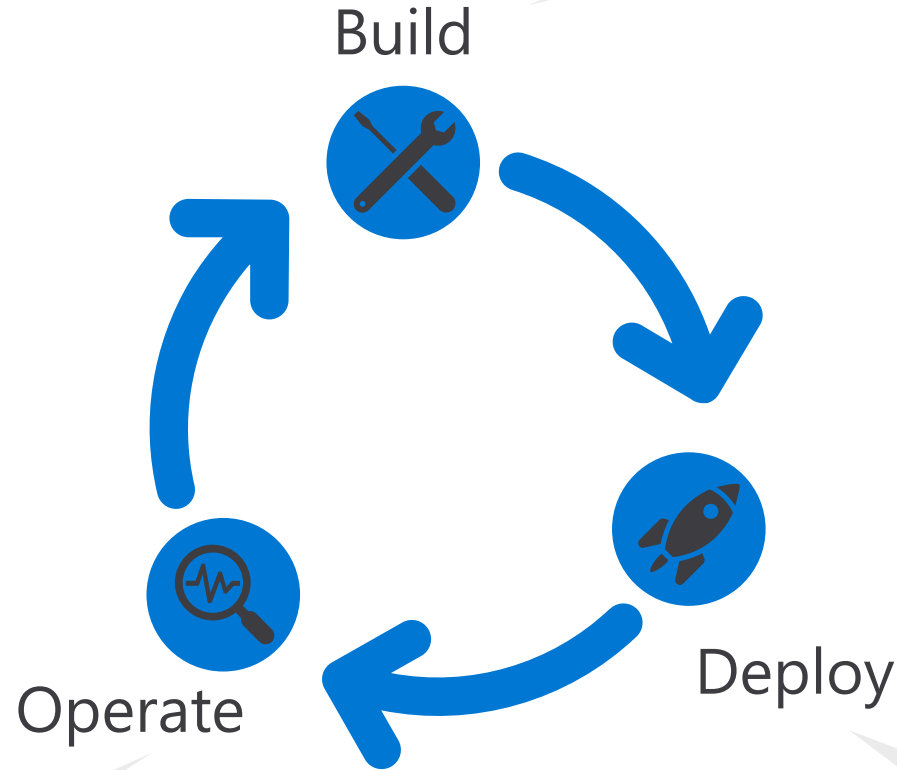
The DevOps methodology

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

Donovan Brown, Microsoft



The DevOps Toolkit (Microsoft)

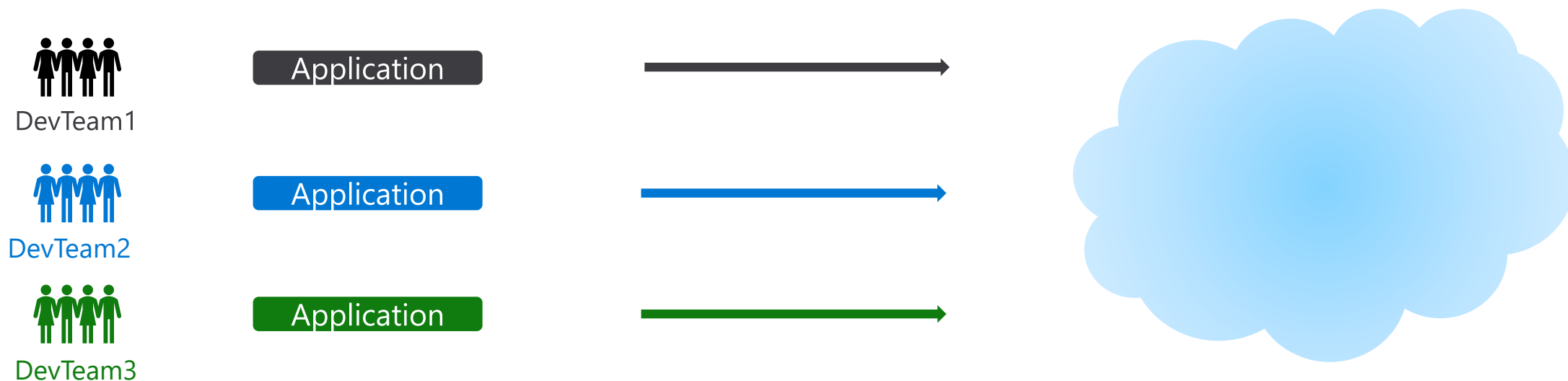


- GitHub
- Visual Studio
- Visual Studio Code

- Azure Monitor
- Azure App Insights
- Azure Security Center

- Azure Boards
- Azure Repos
- Azure Pipelines
- Azure Tests
- Azure Artifacts

Challenge of DevOps: Different approaches, same cloud endpoint



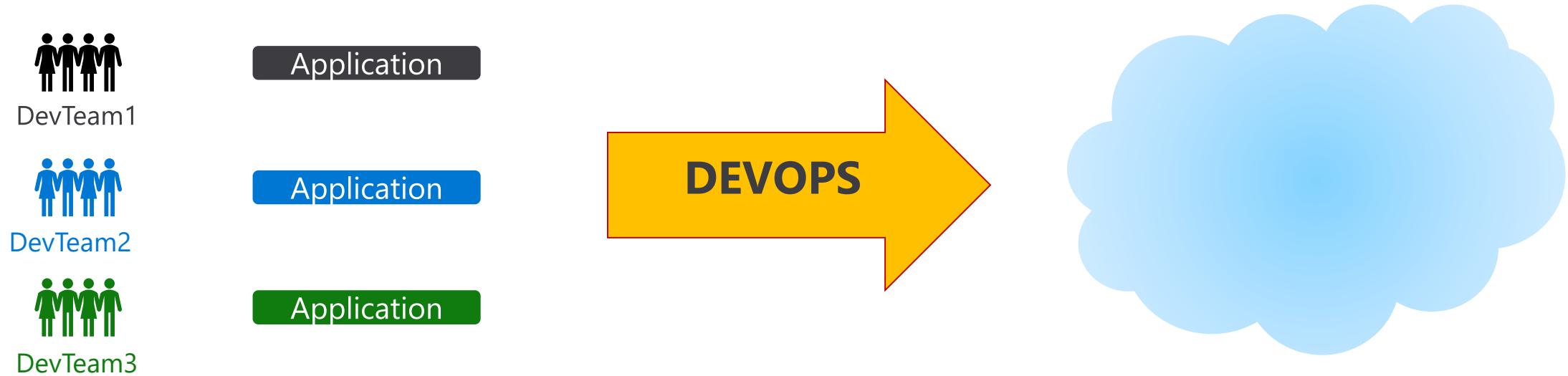
PRO's:

- DevTeams can run independently
- DevTeams are in control
- DevTeams enjoy their work

CON's:

- Different teams = different approaches
- Different teams = different results
- Operational challenges
- Security challenges

Solution of DevOps: Same approach, same toolkit, same cloud endpoint



PRO's:

- DevTeams can run independently
- DevTeams are in control
- DevTeams enjoy their work

More PRO's:

- Different teams = same approach
- Different teams = same results
- Operational benefits
- Governed and Controlled

Introducing Azure DevOps

“... Azure DevOps is a suite of products that allows any organization to do better 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.

Introducing Azure DevOps

“... Azure DevOps is a suite of products that allows any organization to do better 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.

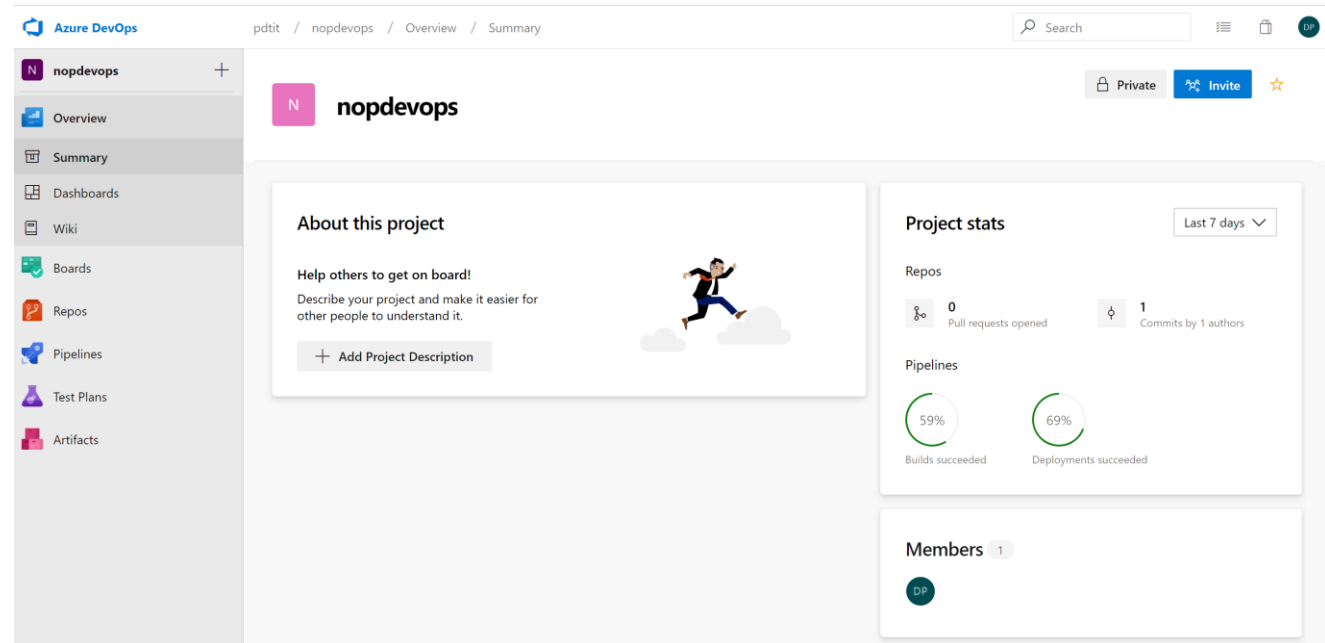


Azure Repos

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

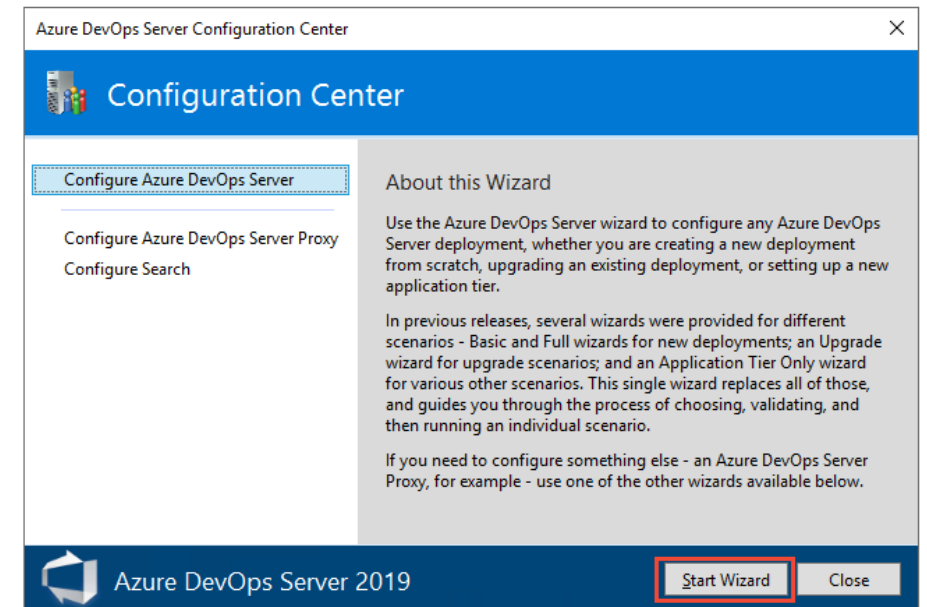
Benefits of Azure DevOps Services

- Quick Set-up
- Maintenance-free operations
- Easy collaboration across domains
- Elastic Scale
- Rock-solid security
- Access to cloud-running build and deployment servers



Azure DevOps Server

- Simplicity of Azure DevOps from the cloud, in your on-prem datacenter
- Data always stays in your own datacenters
- Work process and tracking requirements are handled using XML processes model, instead of using the inheritance process model
- Azure DevOps Build Server supports both on-premises and cloud-hosted builds
- SQL Server and SQL Analysis Server can be added
 - Single Server
 - Dual Servers
 - Multiple Servers
 - Express Install Mode
 - Custom Install Mode



Demo

Creating your first Azure DevOps Project

Introducing Azure DevOps

“... Azure DevOps is a suite of products that allows any organization to do better DevOps, **where you can choose which products you use...**”



JIRA

If you do development tracking with Jira instead of Azure Boards, fine, **Azure DevOps totally integrates with it**



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.



Git / GitHub

If you don't want to use Azure Repos, fine, use Git / GitHub instead; **Azure DevOps totally integrates with it**

Main benefit
is the
openness

Azure Boards

Track work with Kanban boards, backlogs, team dashboards and custom reporting



Drag & Drop Sprint Planning

Flexible work item tracking, using comprehensive traceability to have the perfect environment to manage your development projects and processes



Scrum-ready

Use built-in Scrum boards and planning tools to help your teams run sprints, stand-ups, and planning meetings

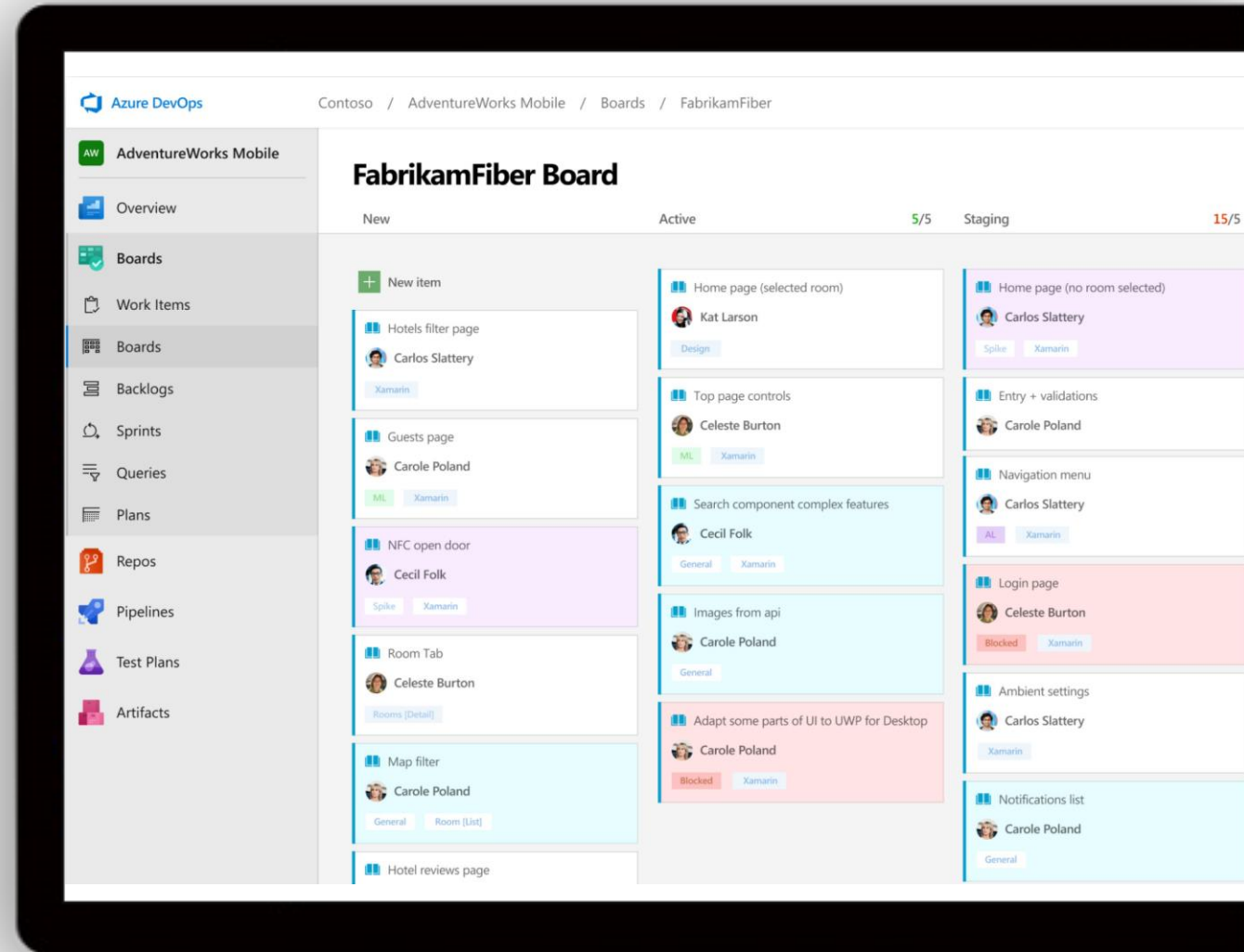


Integration with GitHub

Boost your teams productivity with Boards, Backlogs and Sprints for easy or most complex projects. Connect your GitHub Repo to Azure Boards and start linking commits and PRs to work items

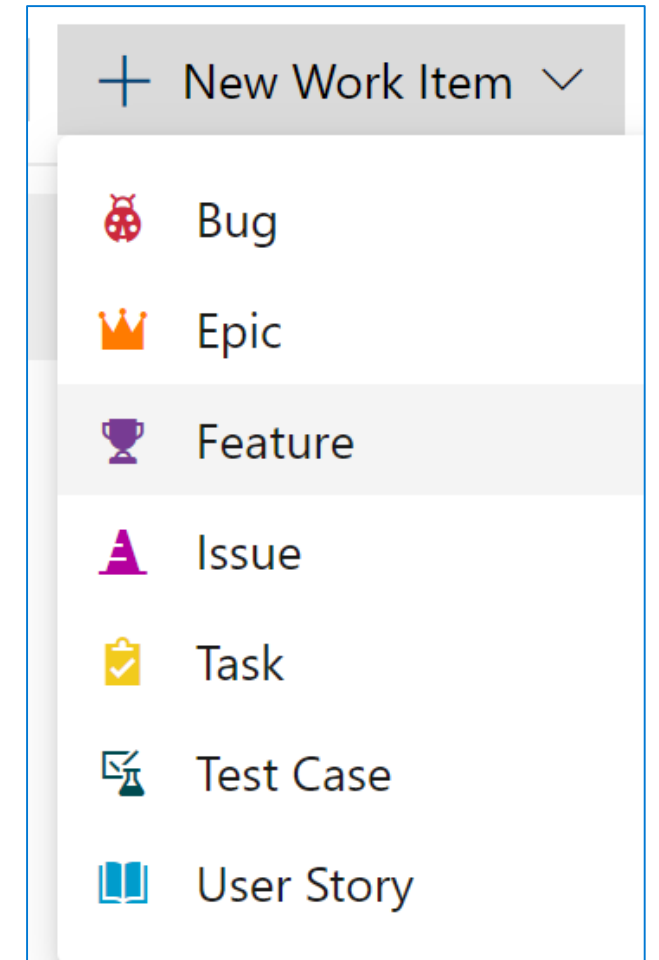
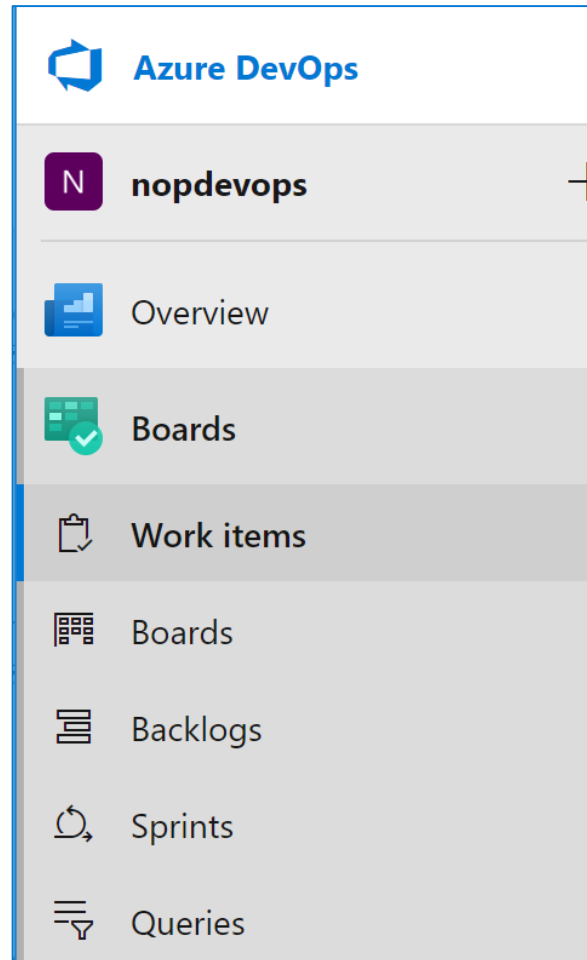


<https://azure.com/boards>



Azure Boards

- Work Items
- Boards
- Backlogs
- Sprints
- Queries



Demo

Azure Boards

Azure Repos

Cloud-hosted, unlimited private Git Repositories for your projects



Support for any Git Client

Securely connect with and push code into your Git Repos from any IDE, editor or Git Client



Collaborate to build better code

Perform more effective Git code reviews with threaded discussion and continuous integration for each change. Use forks to promote collaboration with inner source workflows

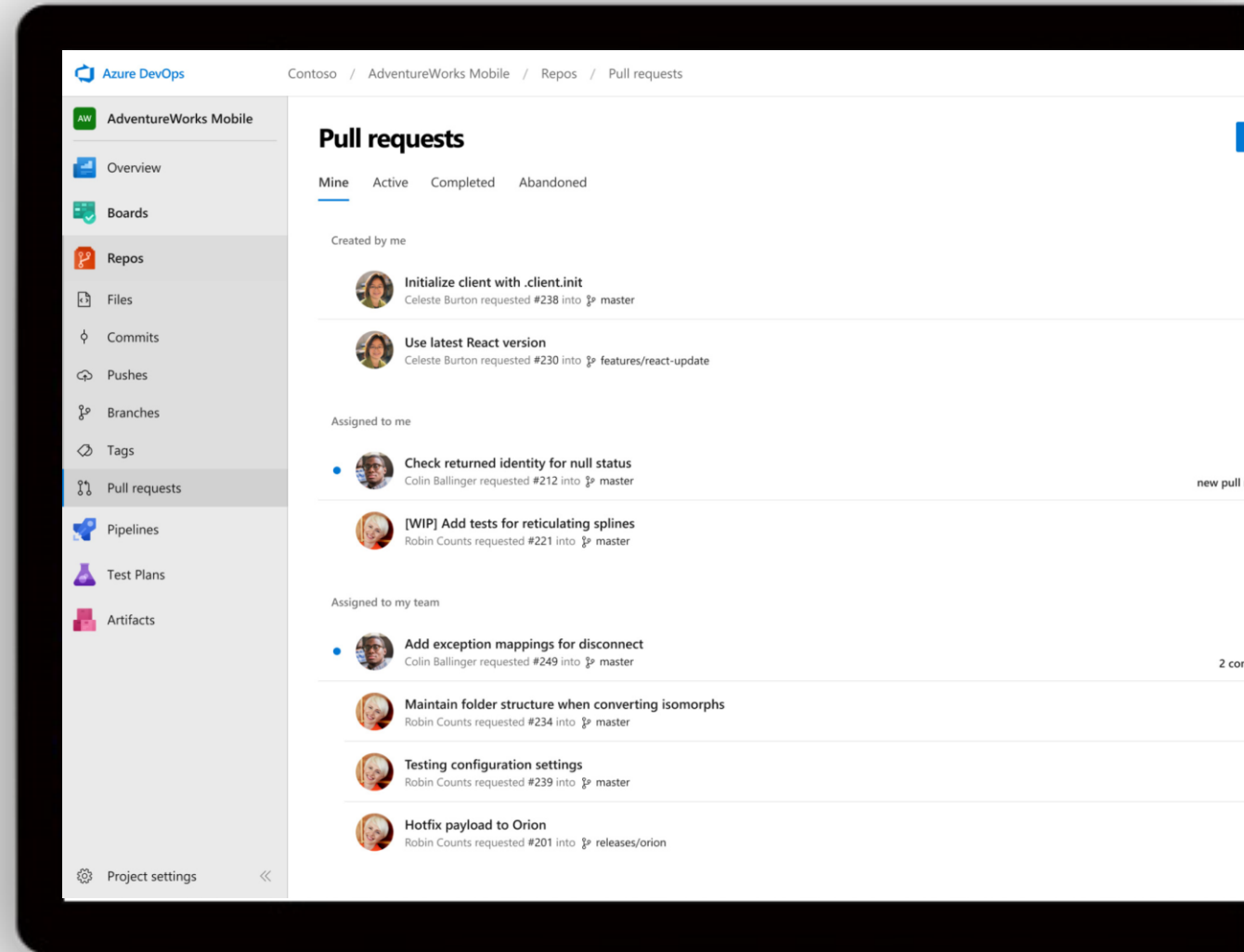


Protect your code with branches

Keep code quality high by requiring signoff, successful builds, and passing tests before pull requests can be merged. Customize your branch policies to maintain your team's standards



<https://azure.microsoft.com/en-us/services/devops/repos/>



Demo

Azure Repos

Azure Pipelines

Cloud-hosted pipelines for Linux, Windows and macOS.



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

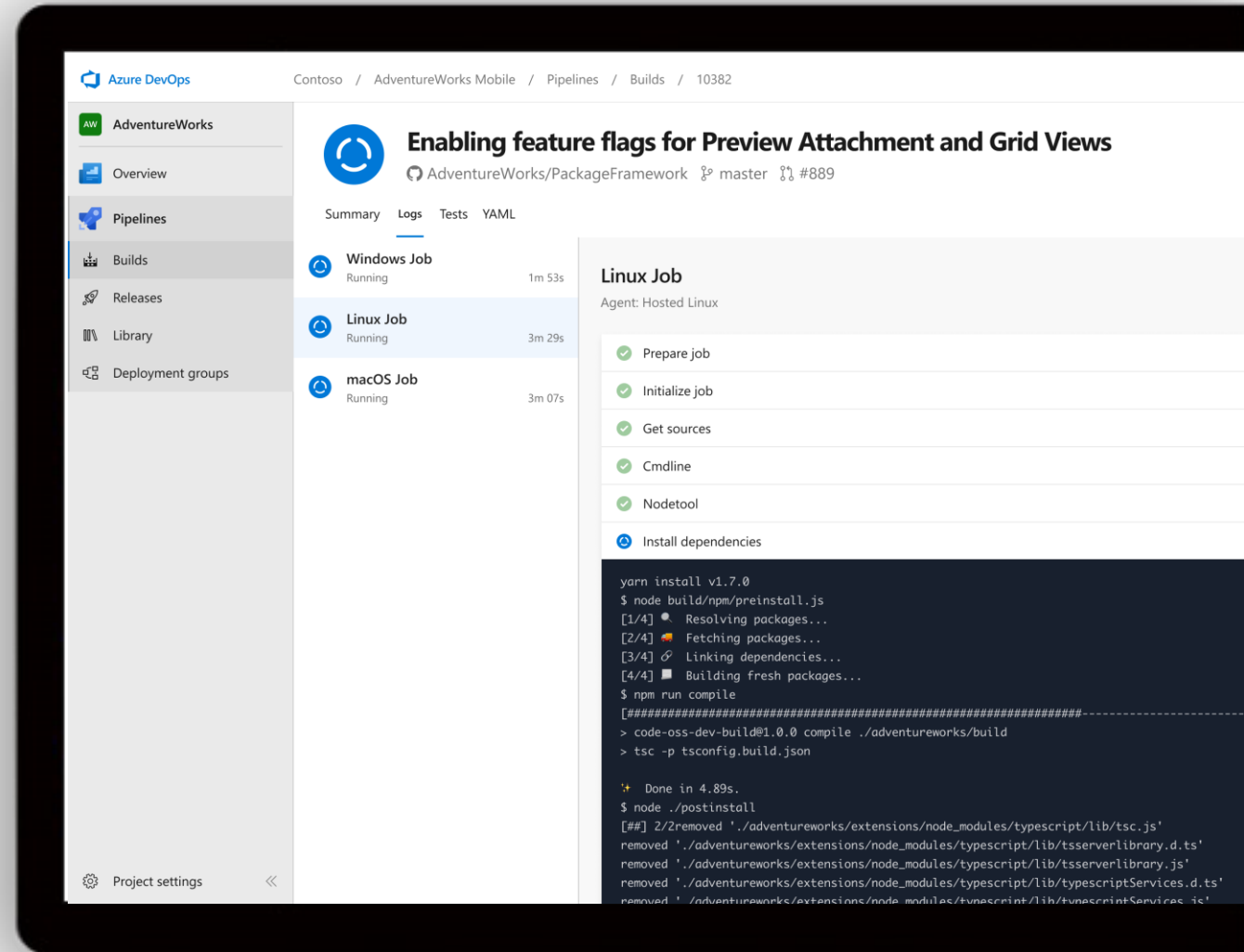


Containers and Kubernetes

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

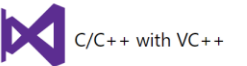
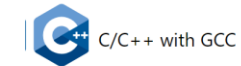
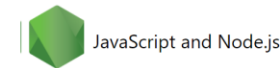


<https://azure.com/pipelines>



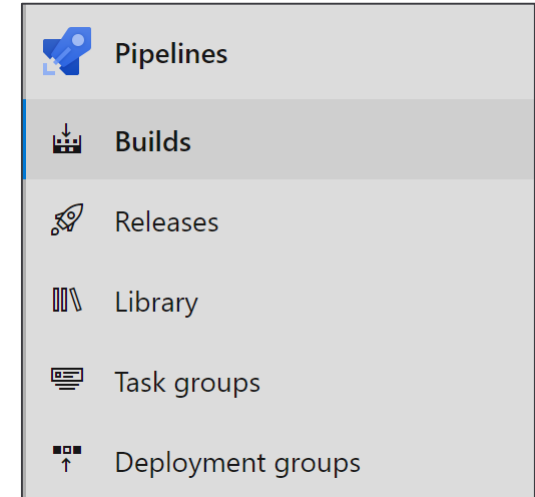
Running a Build with Azure DevOps Pipelines

- Supporting multiple languages
- Prerequisites:
 - A GitHub Account
 - An Azure DevOps Organization
 - Application Source Code
- Based on your source code, Azure DevOps Pipelines “recognizes” the capabilities
- The output of the Pipelines process is a “Azure-Pipelines.yml” file



Building a Pipeline for .NET Core applications

1. New Build Pipeline



Building a Pipeline for .NET Core applications

1. New Build Pipeline
2. Select Source Control environment

Connect


Select

Configure

Review


New pipeline

Where is your code?

 Azure Repos Git


YAML

Free private Git repositories, pull requests, and code search

 Bitbucket Cloud


YAML

Hosted by Atlassian

 GitHub


YAML

Home to the world's largest community of developers


 GitHub Enterprise Server

YAML

The self-hosted version of GitHub Enterprise

 Other Git

Any generic Git repository

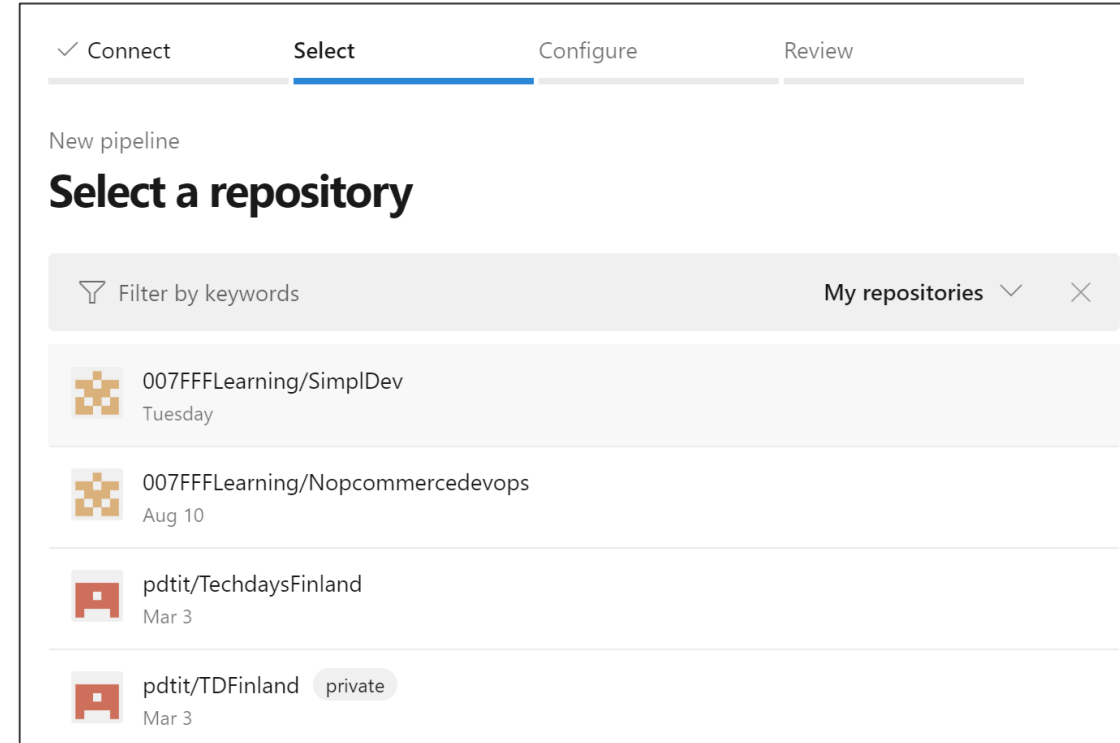
 Subversion

Centralized version control by Apache

[Use the classic editor](#) to create a pipeline without YAML.

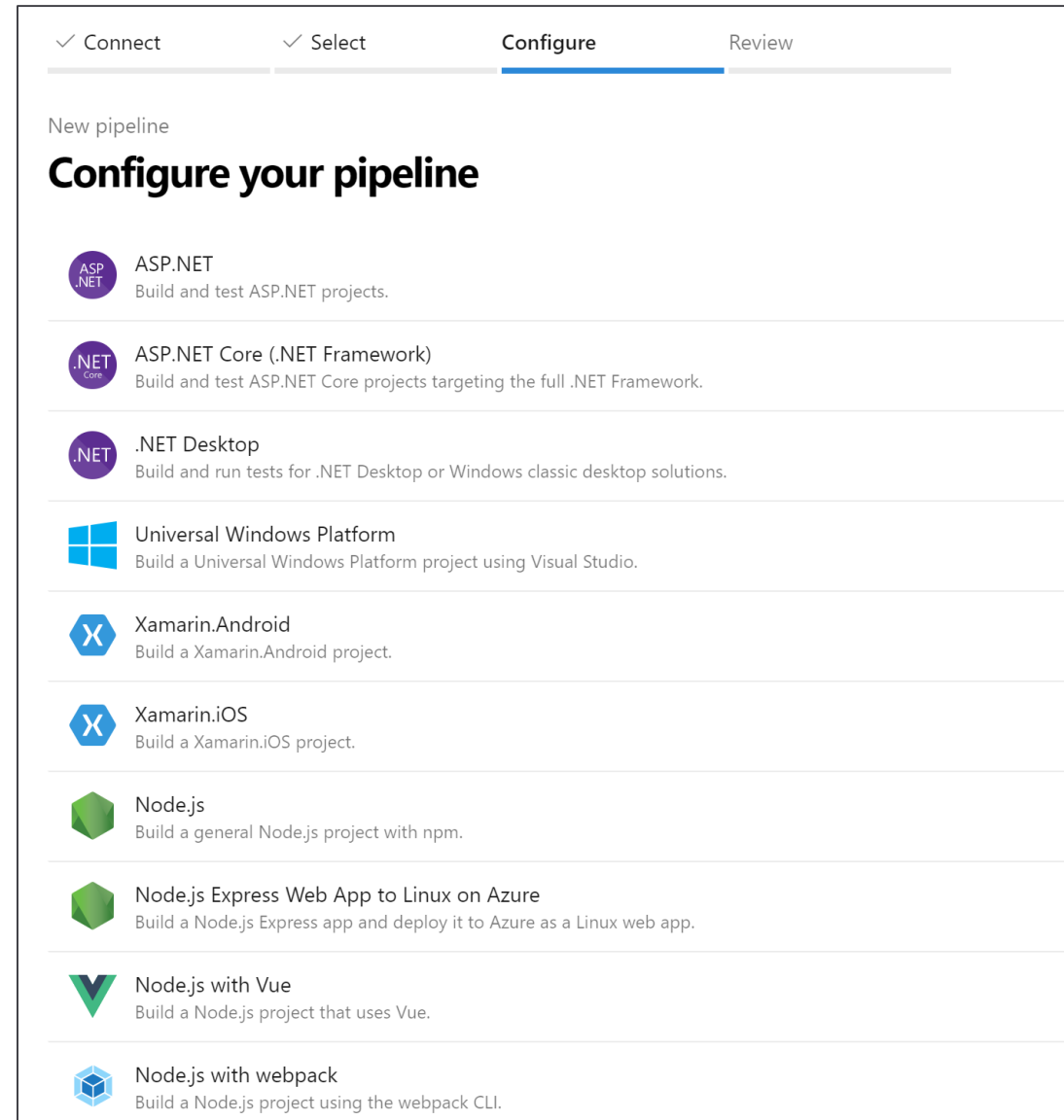
Building a Pipeline for .NET Core applications

1. New Build Pipeline
2. Select Source Control environment
3. Select Repo
- 4.



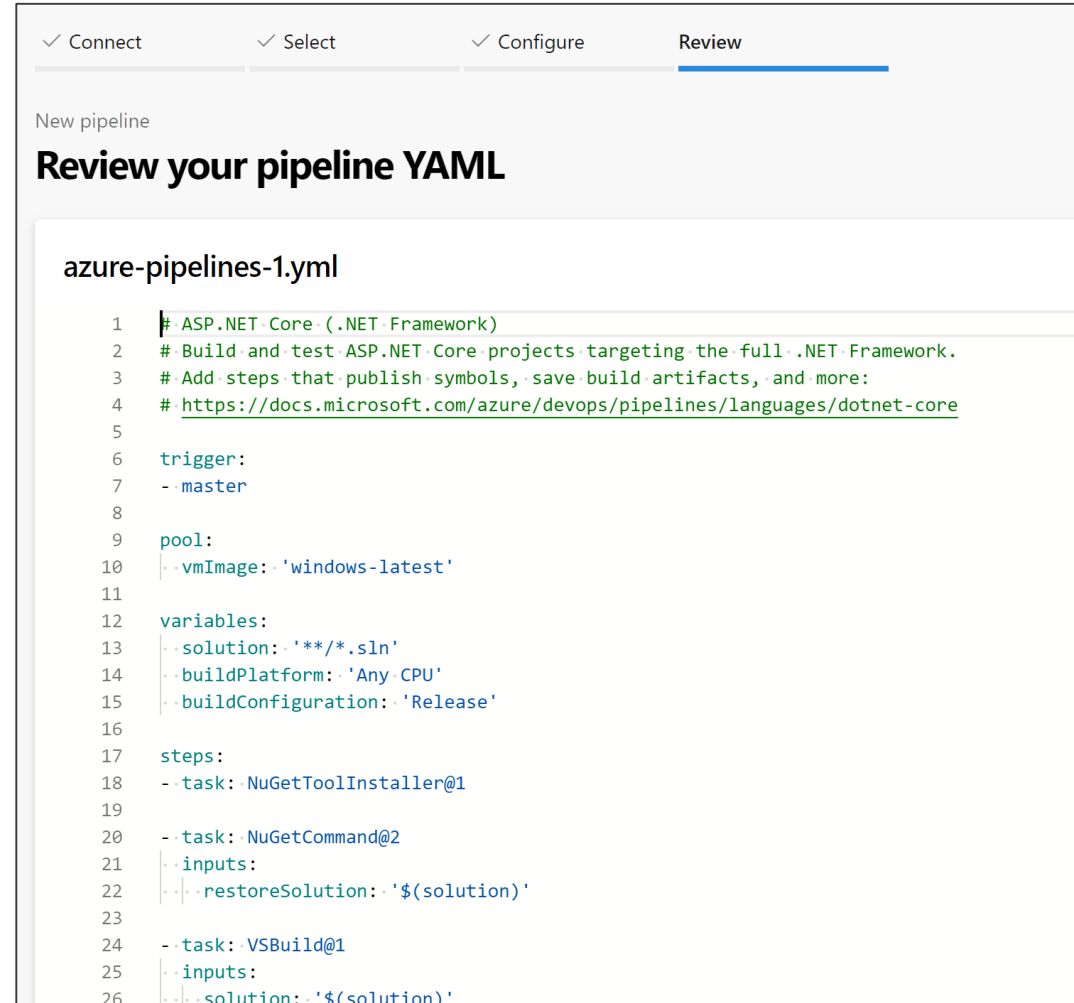
Building a Pipeline for .NET Core applications

1. New Build Pipeline
2. Select Source Control environment
3. Select Repo
4. Azure Pipelines analyzes the source code, and provides “options” for Build



Building a Pipeline for .NET Core applications

1. New Build Pipeline
2. Select Source Control environment
3. Select Repo
4. Azure Pipelines analyzes the source code, and provides “options” for Build
5. This results in a “Azure-pipelines.yml” file



The screenshot shows the 'Review your pipeline YAML' step in the Azure Pipelines configuration wizard. The wizard has four steps: 'Connect', 'Select', 'Configure', and 'Review', with 'Review' being the active step. Below the step indicator, the text 'New pipeline' is visible. The main content area displays a sample YAML file named 'azure-pipelines-1.yml'. The file contains comments in green and configuration in blue. The configuration includes a trigger for the 'master' branch, a pool using the 'windows-latest' VM image, and several variables for solution path, build platform, and build configuration. The steps section defines two tasks: 'NuGetToolInstaller@1' and 'NuGetCommand@2' (which includes inputs for restoring the solution), followed by 'VSBuild@1' (which includes an input for the solution path).

```
1 # ASP.NET Core (.NET Framework)
2 # Build and test ASP.NET Core projects targeting the full .NET Framework.
3 # Add steps that publish symbols, save build artifacts, and more:
4 # https://docs.microsoft.com/azure/devops/pipelines/languages/dotnet-core
5
6 trigger:
7   - master
8
9 pool:
10  - vmImage: 'windows-latest'
11
12 variables:
13   - solution: '**/*.sln'
14   - buildPlatform: 'Any CPU'
15   - buildConfiguration: 'Release'
16
17 steps:
18   - task: NuGetToolInstaller@1
19
20   - task: NuGetCommand@2
21     inputs:
22       - restoreSolution: '$(solution)'
23
24   - task: VSBuild@1
25     inputs:
26       - solution: '$(solution)'
```

Building a Pipeline for .NET Core applications

1. New Build Pipeline
2. Select Source Control environment
3. Select Repo
4. Azure Pipelines analyzes the source code, and provides “options” for Build
5. This results in a “Azure-pipelines.yml” file
6. Create and Run your Build

Linux

Pool: [Hosted Ubuntu 1604](#) · Agent: Hosted Agent

- ✓ Prepare job · succeeded
- ✓ Initialize job · succeeded
- ✓ Checkout · succeeded
- ✓ DotNetCoreInstaller · succeeded
- ✓ dotnet build · succeeded
- ✓ run tests · succeeded
- ✓ Publish Test Results **/*.trx · succeeded
- ✓ Post-job: Checkout · succeeded
- ✓ Finalize Job · succeeded

Demo

Using Azure Pipelines to create your Build

- Source Code to Web Apps
- Source Code to Docker Container

Running a Release with Azure DevOps Pipelines

- Supporting multiple languages
- Start from a template, or blank
- Based on a Pipeline build artifact, or other sources
- Single or multi-staged release scenarios

Featured



Azure App Service deployment

Deploy your application to Azure App Service. Choose from Web App on Windows, Linux, containers, Function Apps, or WebJobs.



Deploy a Java app to Azure App Service

Deploy a Java application to an Azure Web App.



Deploy a Node.js app to Azure App Service

Deploy a Node.js application to an Azure Web App.



Deploy a PHP app to Azure App Service and Azure Database for MySQL

Deploy a PHP application to an Azure Web App and database to Azure Database for MySQL.



Deploy a Python app to Azure App Service and Azure database for MySQL

Deploy a Python Django, Bottle, or Flask application to an Azure Web App and database to Azure Database for MySQL.



Deploy to a Kubernetes cluster

Deploy, configure, update your containerized applications to a Kubernetes cluster.



IIS website and SQL database deployment

Deployment Group: Deploy ASP.NET or ASP.NET Core web applications to an IIS Website and SQL database on physical or virtual machines (VM).

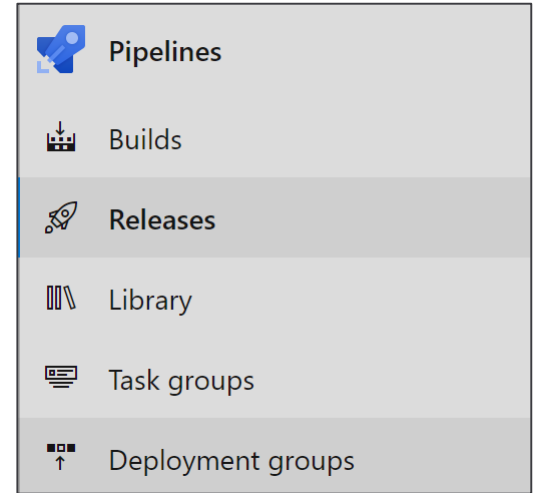
Others



Azure App Service deployment with continuous monitoring

Configuring a Release Pipeline for .NET Core applications

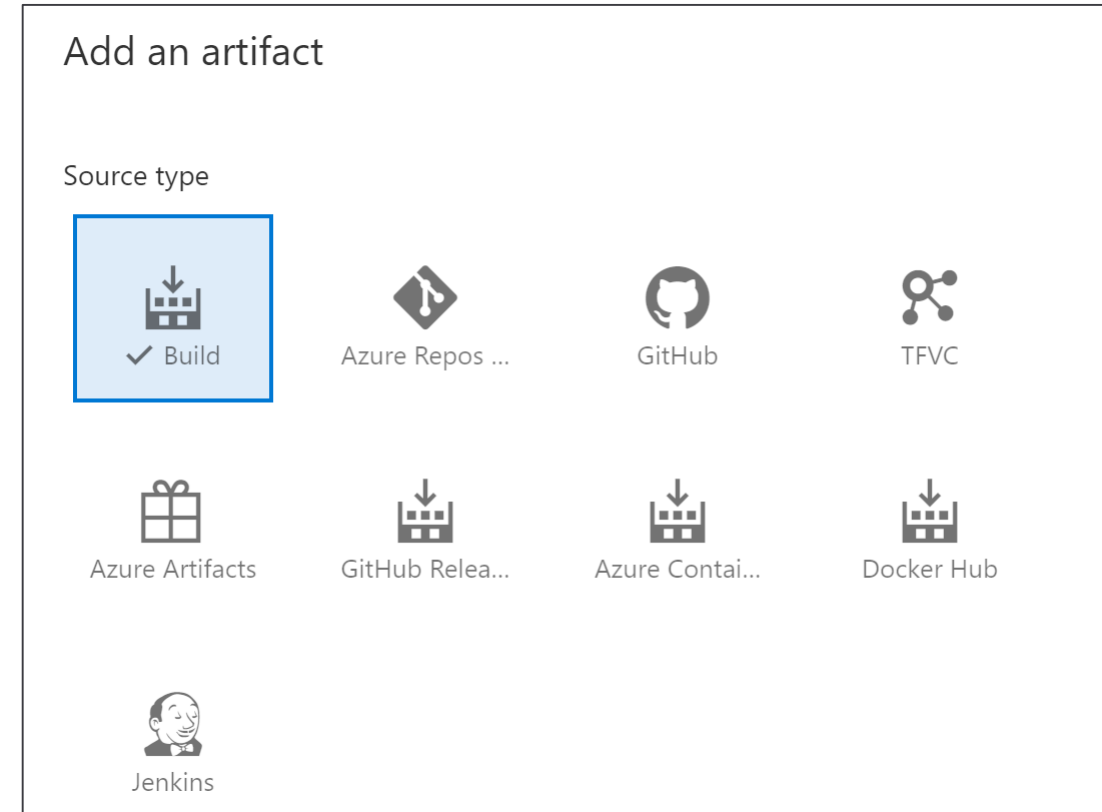
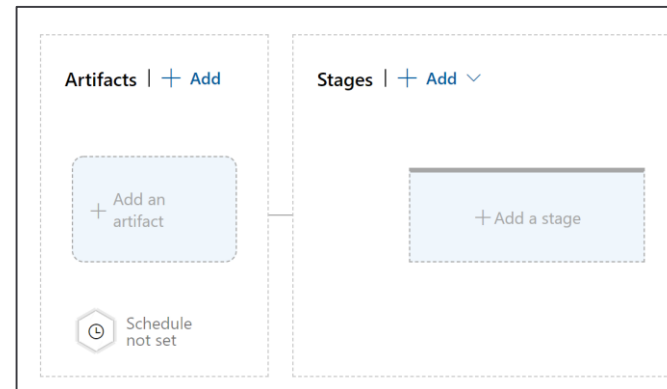
1. New Release Pipeline



Configuring a Release Pipeline for .NET Core applications

1. New Release Pipeline

2. Select Artifacts



Configuring a Release Pipeline for .NET Core applications

1. New Release Pipeline
2. Select Artifacts
3. Define a Stage Template

Featured



Azure App Service deployment

Deploy your application to Azure App Service. Choose from Web App on Windows, Linux, containers, Function Apps, or WebJobs.



Deploy a Java app to Azure App Service

Deploy a Java application to an Azure Web App.



Deploy a Node.js app to Azure App Service

Deploy a Node.js application to an Azure Web App.



Deploy a PHP app to Azure App Service and Azure Database for MySQL

Deploy a PHP application to an Azure Web App and database to Azure Database for MySQL.



Deploy a Python app to Azure App Service and Azure database for MySQL

Deploy a Python Django, Bottle, or Flask application to an Azure Web App and database to Azure Database for MySQL.



Deploy to a Kubernetes cluster

Deploy, configure, update your containerized applications to a Kubernetes cluster.



IIS website and SQL database deployment

Deployment Group: Deploy ASP.NET or ASP.NET Core web applications to an IIS Website and SQL database on physical or virtual machines (VM).

Others



Azure App Service deployment with continuous monitoring

Configuring a Release Pipeline for .NET Core applications


1. New Release Pipeline
2. Select Artifacts
3. Define a Stage Template
4. Complete Stage Template parameters

Stage name

Stage 2

Parameters ⓘ | [Unlink all](#)

Azure subscription * ⓘ | [Manage](#)


007FFFLearning Labs (0a407898-c077-442d-8e17-71420aa82426) ▼ 

ⓘ Scoped to subscription '007FFFLearning Labs'

App type ⓘ

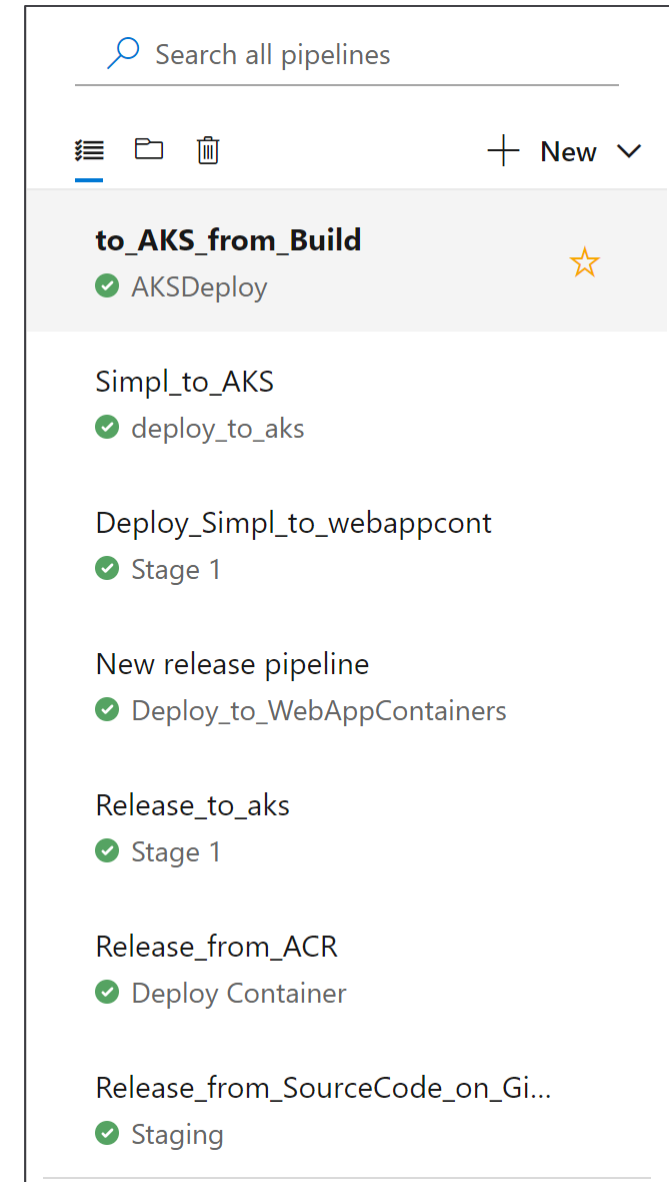
Web App for Containers (Linux) ▼

App service name * ⓘ

simplwebcontdevops ▼ 

Configuring a Release Pipeline for .NET Core applications

1. New Release Pipeline
2. Select Artifacts
3. Define a Stage Template
4. Complete Stage Template parameters
5. Save & Run



Demo

Using Azure Pipelines to create your Release

- From source code Build to Azure WebApps
- From source container to Azure WebApps for Containers
- From source container to AKS

Monitoring a Release Pipeline for .NET Core applications

- Full details of each step in the Release process
- Succeeded / Failed
- Duration of the Pipeline

The screenshot displays the Azure DevOps Release Pipeline monitoring interface. At the top, the pipeline path is shown as 'Release_from_ACR > Release-2 > Deploy Container', with a green 'Succeeded' status badge. Below this is a navigation bar with tabs for 'Pipeline', 'Tasks', 'Variables', 'Logs' (which is selected), 'Tests', and a set of action buttons: 'Deploy', 'Cancel', 'Refresh', 'Download all logs', 'Edit', and a menu icon. The main content area is divided into two panels. The left panel, titled 'Deployment process', shows a list of steps with 'Run on agent' highlighted as 'Succeeded'. The right panel, titled 'Run on agent', provides details for the selected step, showing it was executed on a 'Hosted VS2017' agent. It lists three sub-steps: 'Initialize job', 'Azure Web App on Container Deploy: contnopwebapp', and 'Finalize Job', all of which are marked as 'succeeded'.

Release_from_ACR > Release-2 > Deploy Container ✓ Succeeded

← Pipeline Tasks Variables **Logs** Tests | ☁ Deploy ⛔ Cancel ↻ Refresh ⬇ Download all logs ✎ Edit ⋮

Deployment process
Succeeded

- ✓ Run on agent
Succeeded

Run on agent
Pool: Hosted VS2017 · Agent: Hosted Agent

- ✓ Initialize job · succeeded
- ✓ Azure Web App on Container Deploy: contnopwebapp · succeeded
- ✓ Finalize Job · succeeded

Monitoring a Release Pipeline for .NET Core applications

- Full details of each step in the Release process

```
1 2019-08-11T18:16:38.0429690Z ##[section]Starting: Initialize job
2 2019-08-11
3 2019-08-11
4 2019-08-11
5 2019-08-11
6 2019-08-11
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26

1 2019-08-11T18:16:40.7798636Z ##[section]Starting: Azure Web App on Container Deploy: contnopwebapp
2 2019-08-11T18:16:41.0218887Z =====
3 2019-08-11T18:16:41.0219066Z Task      : Azure Web App for Containers
4 2019-08-11T18:16:41.0219314Z Description : Deploy containers to Azure App Service
5 2019-08-11T18:16:41.0219368Z Version    : 1.0.20
6 2019-08-11T18:16:41.0219430Z Author     : Microsoft Corporation
7 2019-08-11T18:16:41.0219492Z Help       : https://docs.microsoft.com/azure/devops/pipelines/tasks/deploy/azure-rm-web-app-containers
8 2019-08-11T18:16:41.0219597Z =====
9 2019-08-11T18:16:42.4716693Z Got service connection details for Azure App Service:'contnopwebapp'
10 2019-08-11T18:16:42.7336601Z Single-container Deployment to the webapp 'contnopwebapp' as only the image detail was sepcified.
11 2019-08-11T18:16:43.1923367Z Updating App Service Configuration settings. Data: {"appCommandLine":null,"linuxFxVersion":"DOCKER|nopacr1.azurecr.io/nopcon
12 2019-08-11T18:16:45.4827837Z Updated App Service Configuration settings
13 2019-
14 2019-
15 2019-
16 2019-
17 2019-
18 2019-
19 2019-

1 2019-08-11T18:16:53.6649998Z ##[section]Starting: Finalize Job
2 2019-08-11T18:16:53.6694180Z Start cleaning up orphan processes.
3 2019-08-11T18:16:53.6754539Z ##[section]Finishing: Finalize Job
4
```

Demo

Monitoring Azure Pipelines

Azure Test Plans

Test and Ship with confidence using manual and exploratory testing tools



Test across web and desktop

Test your application by executing tests across different environments, from web apps to different desktop applications



Planned manual and automated tests

Plan, execute and track scripted tests with actionable defects and end-to-end traceability. Assess quality throughout the development lifecycle by testing your desktop and web applications

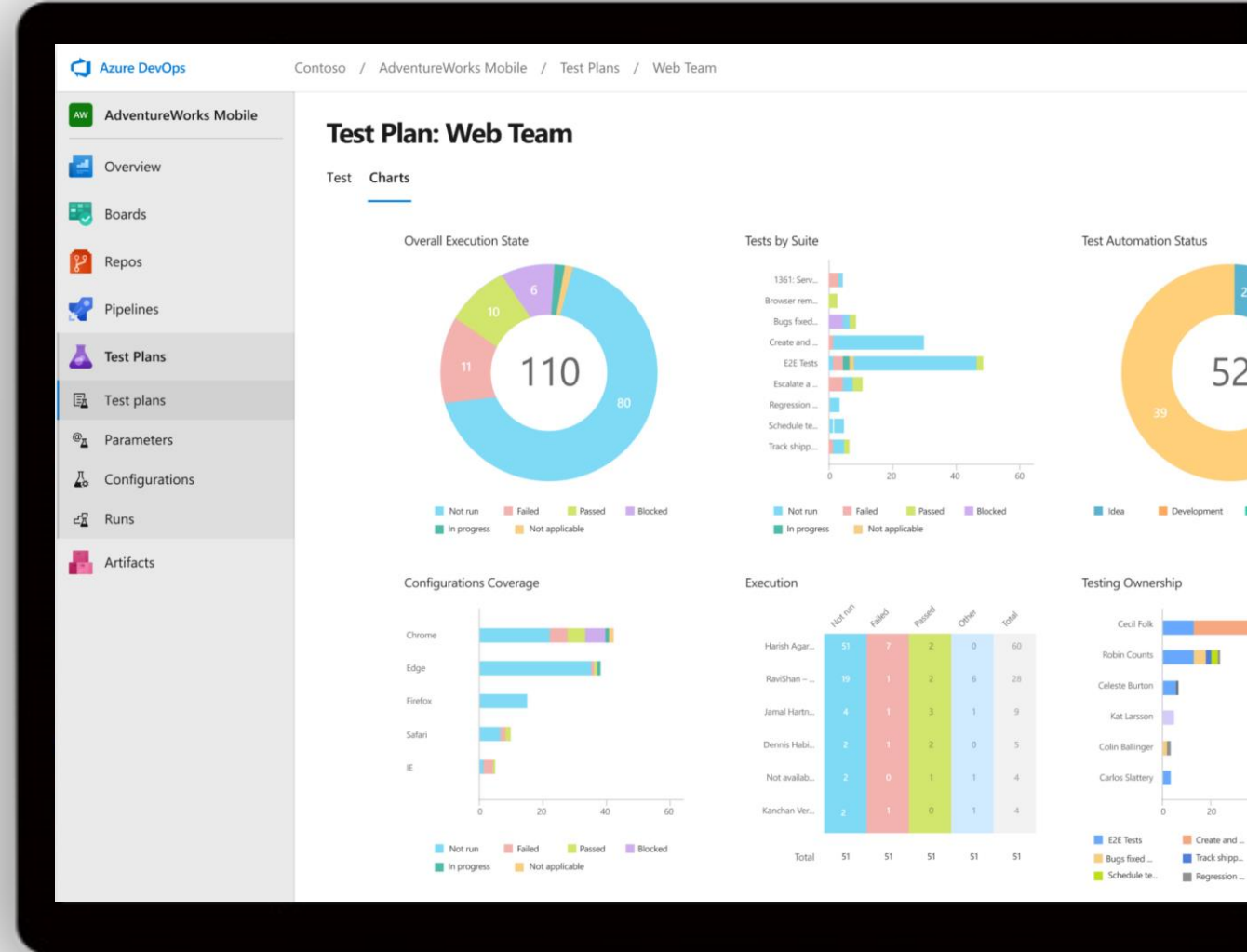


Unit & Functional Testing (VS2019)

Create a Unit Test project; Run Unit Tests with Test Explorer; Start using IntelliTest; Use code coverage to determine how much code is being tested



<https://azure.microsoft.com/en-us/services/devops/repos/>



Azure Artifacts

Create, host and share packages with your team



Support for multiple package languages

Create and share Maven, NPM, NuGet, and Python package feeds from public and private sources



Keep your artifacts organized

Share code effortlessly by storing Maven, NPM, NuGet, Python packages together. And there is no need to store these in Git. Simply store them using Universal Packages in Azure Artifacts

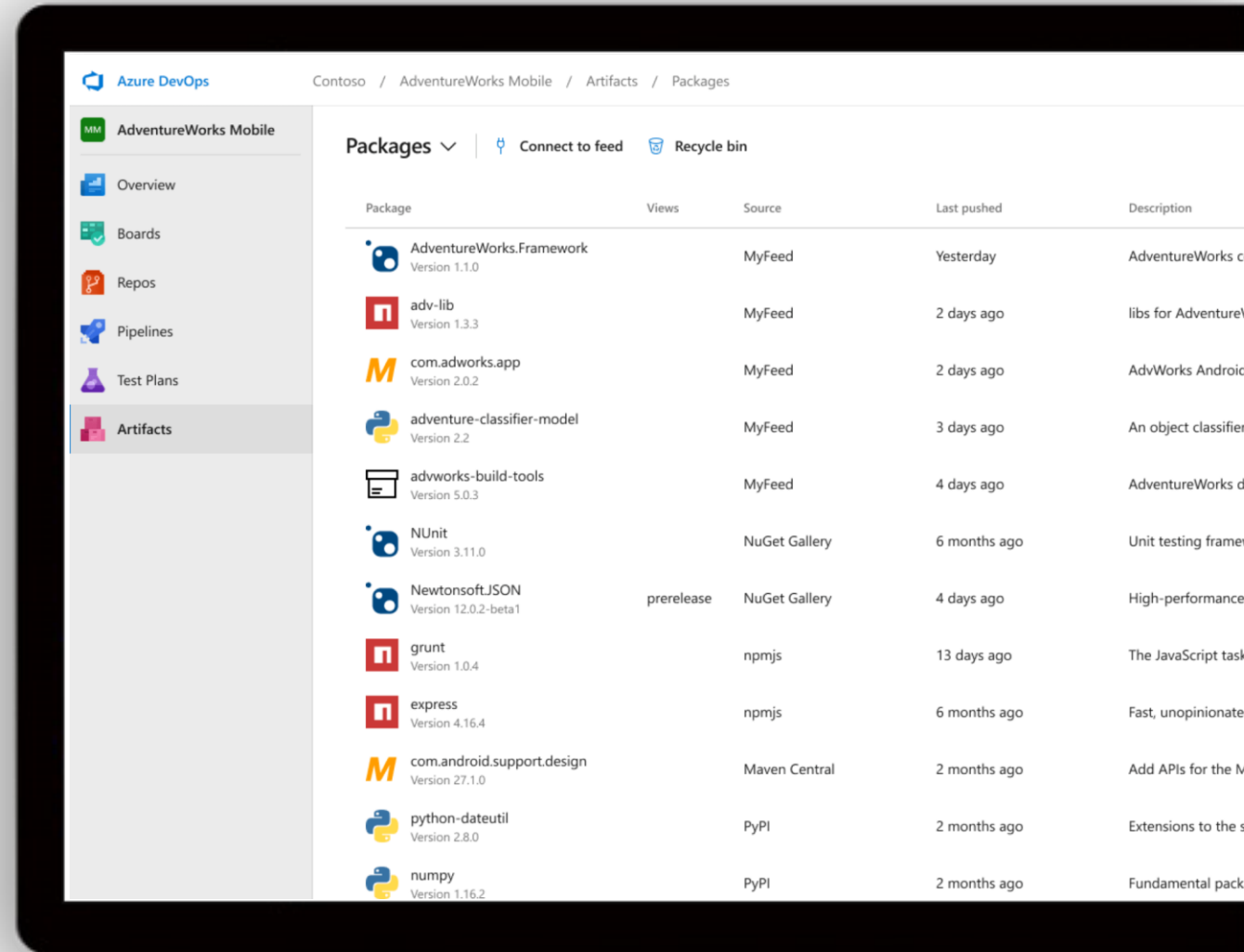


Seamless Package handling

Integrate seamless package handling into your CI/CD Pipeline. Easily access all your artifacts in builds and releases.

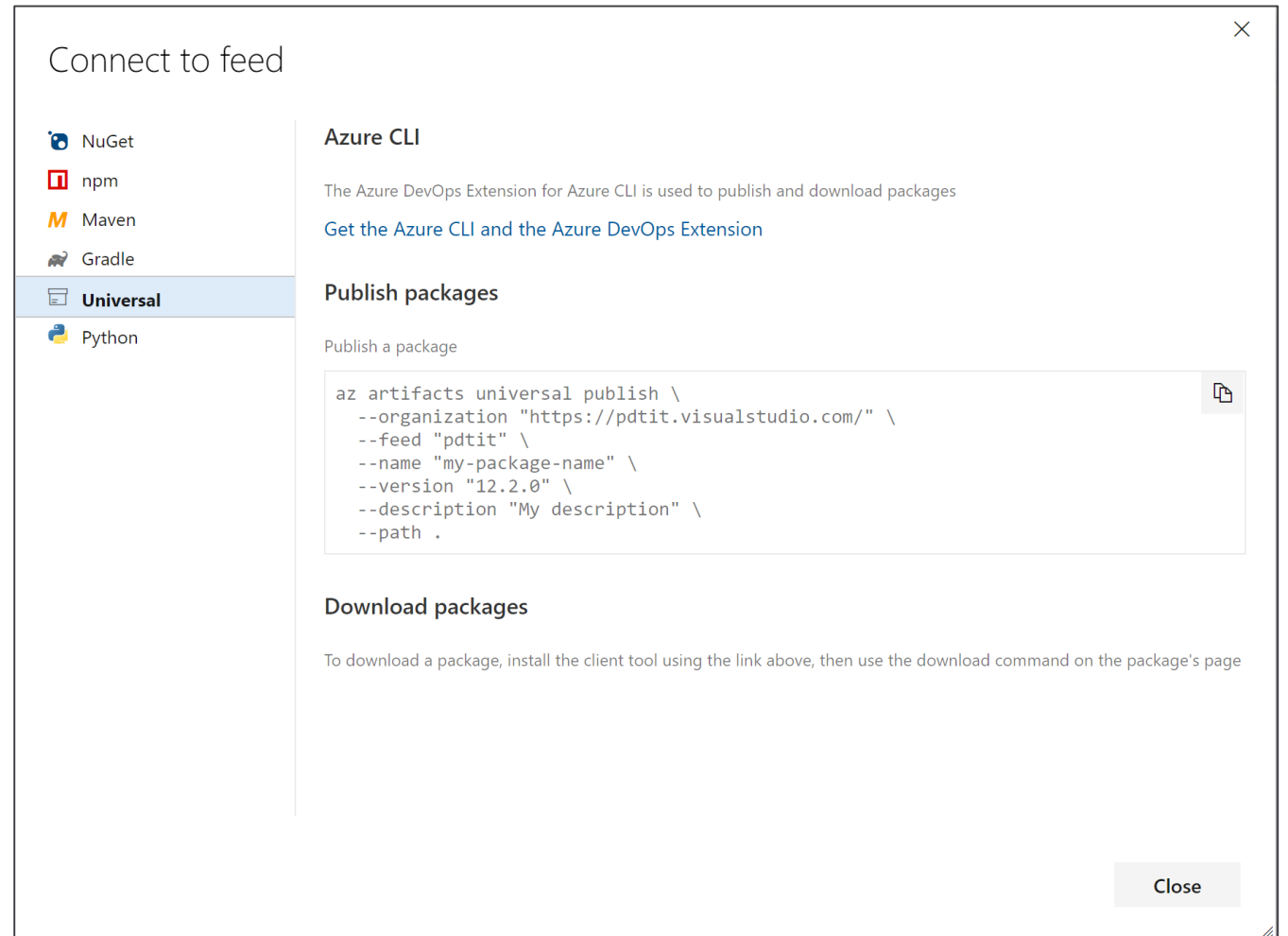


<https://azure.microsoft.com/en-us/services/devops/repos/>



Azure Artifacts

- Based on “feeds”
- Private & Public
- Multiple packages:
 - Maven, NPM, Nuget, Universal





Azure Pipelines

- Free **unlimited** build minutes for public projects
- Up to 10 free parallel jobs across Windows, Linux and macOS

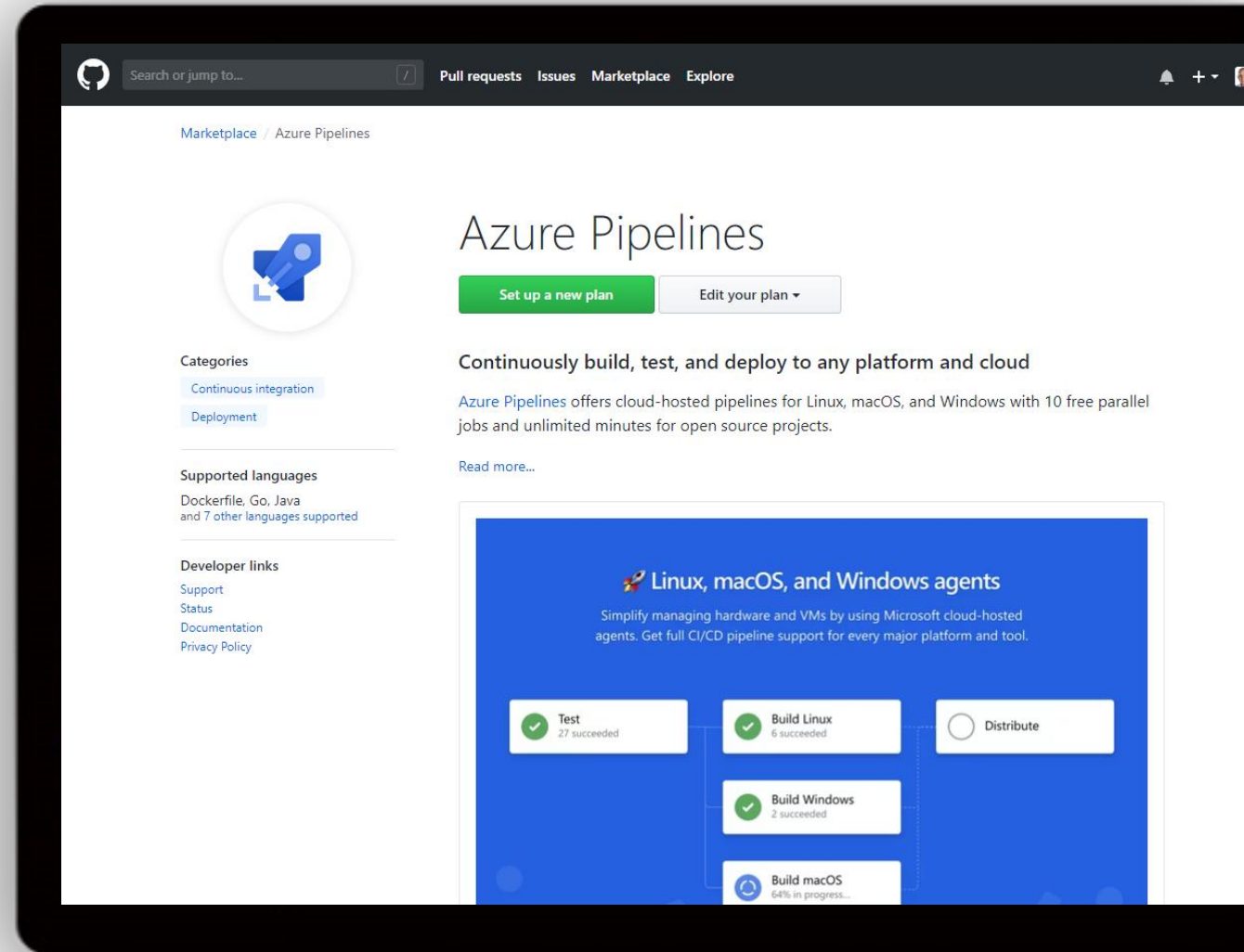
 <https://azure.com/pipelines>

Microsoft ❤️ Open Source



Integrated with GitHub

Azure Pipelines available now to any developer from the GitHub Marketplace



Azure DevOps Summary

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

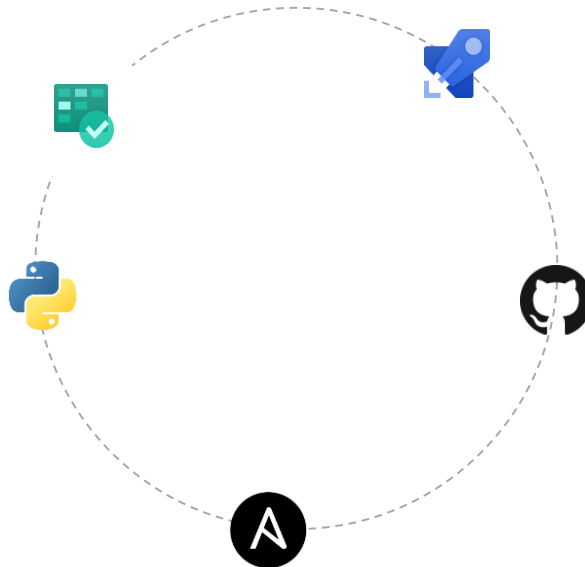


<https://azure.com/devops>

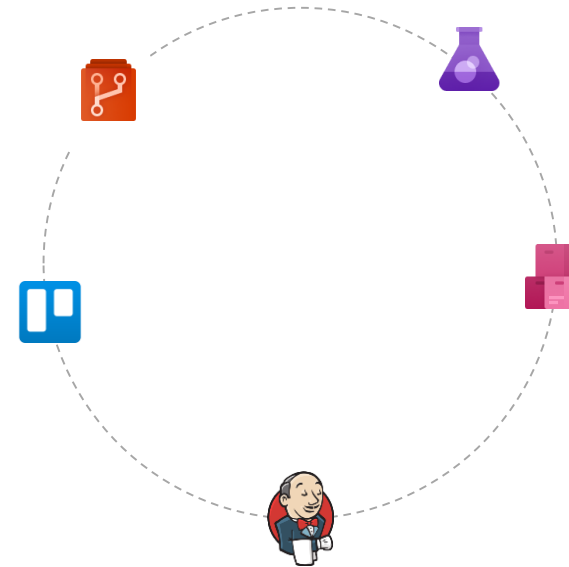
Azure DevOps: Choose what you love

Your tools, languages, and clouds

Any cloud, any platform, whether public, hybrid, on-premises



Developers are still in control of the language and process they want to keep using...



Azure DevOps integrates with existing dev tools, or can replace most of them... you choose!!

Section Take-Aways

1. Azure DevOps helps organizations in adopting and using DevOps... easier and better
2. Azure DevOps covers all aspects of the DevOps cycle: Repos, Build, Release, Artifacts,... end-to-end CI/CD Pipeline
3. Azure DevOps provides massive flexibility, by integrating with your existing Open Source tools you already use today

Questions Landing Spot

“...If you want good answers,
ask better questions...”

© Randy Glasbergen

Appendix

Lab dependencies and alternate path

Alternate path

To avoid the SQL dependencies from the first series labs

- SimplCommerce offers a Docker container that uses a built-in non-SQL database
- <https://hub.docker.com/r/simplcommerce/ci-build>
- If you run this locally (or in ACI or AKS) it will spin up the web app and give you the option to select “sample products” (phones or fashion)

Create container instance

✓ Validation passed

Basics Networking Advanced Tags Review + create

Basics

Subscription	007FFFLearning Labs
Resource group	simplwebappRG
Region	(Europe) North Europe
Container name	nodbcontainer
Image type	Public
Image name	simplcommerce/ci-build
OS type	Linux
Memory (GiB)	1.5
Number of CPU cores	1
GPU type	None
Number of GPU cores	0

Networking

Include public IP address	Yes
Ports	80 (TCP)

Advanced

Restart policy	On failure
Command override	[]

Tags

(none)

Full workshop

Dependencies from first series

The lab guide assumes that the learner has completed the first 3 labs from the previous series “[Migrate a web app to Azure](#)” and the first 3 labs from this series

[Deploying VM baseline using ARM Templates](#)

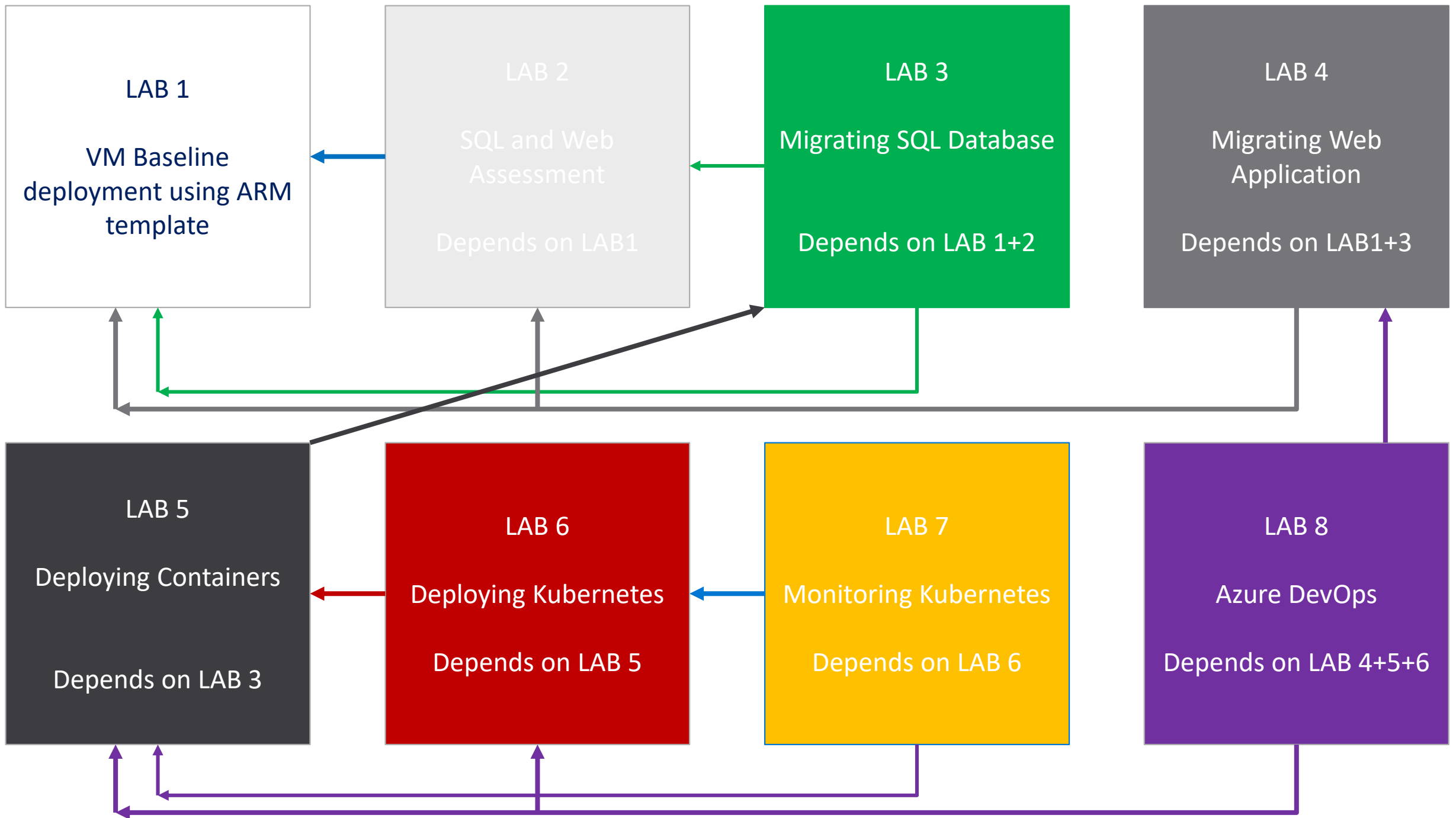
[Performing Assessments](#)

[Migrating SQL Databases](#)

[Deploying and running Azure containers](#)

[Deploying and running Azure Kubernetes Services](#)

[Optimize Operations and Monitoring](#)



Thank You