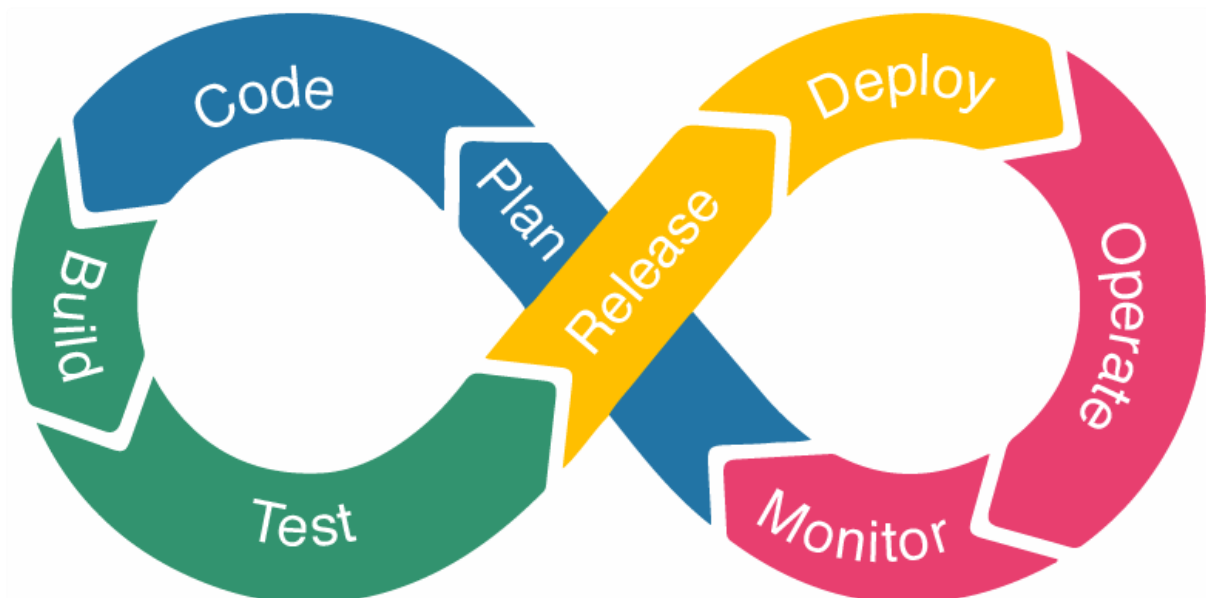


Azure DevOps - Initial Draft

Johan Mallo Bakken

Intro to DevOps

What exactly is DevOps? Is it a set of tools? Well, not really, even though there are many tools that relate to DevOps, the concept itself is not related to any tools. DevOps is rather a way of thinking how software should be developed. It solves several problems from the old methods of software development, and helps in many aspects of the process. Some of the benefits of DevOps are predictability, reproducibility, maintainability, time to market, quality improvement, risk reduction, resiliency, cost efficiency and code partitioning.(1) To help achieve all these benefits, the development of software needs to happen in a certain way. It needs to follow the DevOps lifecycle shown in the figure below.



(2)

Some of these segments might be confusing at first glance, here's a small explanation of each of the stages in the DevOps lifecycle. The application lifecycle starts with the planning phase which is where the scheduling and management of tasks takes place. After this the next stage would be the coding stage where the development and review of the code happen. Both these stages could together be called the development stage. After the development comes the building stage, here continuous integration to build the project and make sure the new changes can correctly be integrated. To do this it's important to implement tests in order to assess the code quality, performance and other risks, this makes the build and test phase closely knit together. In the next phase the release and deployment takes place, here the new functionality is integrated with the prevailing code and deployed in a manner that it does not affect the systems availability and functionality. In the Operate and monitor stage inappropriate system behaviour and bugs are found on the system that's already in production.(3)

What is Azure DevOps?

As mentioned in the previous section, there are many tools related to DevOps. And the focus of this article will be on some of them, specifically the tools provided by Microsoft's Azure DevOps service. Azure DevOps is a set of cloud hosted DevOps services that work for any language targeting any platform. These set of DevOps services, aim to provide everything one needs to go from an idea to a working piece of software. This includes access to agile tools that help with the planning phase of projects, code management through Git, and code deployment through its own CI/CD system. (4) All these tools are packed into Azure DevOps five main services.



Azure boards is a set of agile tools for planning projects. With Azure Boards the developers can use agile tools to plan, track and discuss work across the teams. The work becomes trackable through the use of Kanban boards, backlogs, team dashboards and custom reporting.



Azure Pipelines offers automated functionality for building, testing and deploying continuously through a CI/CD system. The system works with any language, platform and cloud and it can be connected to GitHub or any other Git provider to continuously deploy.



Azure Repos offers unlimited, cloud-hosted private Git repos and support for Team Foundation Version Control. This gives projects an interface for advanced file management and allows for a good collaboration between the developers on the code base itself.



Azure Test Plans allows for both manual and exploratory testing. This means one can plan, execute and track scripted tests with actionable defects and end-to-end traceability through manual tests. Or design and execute tests simultaneously through exploratory testing.



Azure Artifacts provides package management and facilitates the usage, creation and publication of packages. Maven, npm, NuGet, and Python packages from public or private sources can easily be created and shared within a team.

(5,6,7,8,9,10)

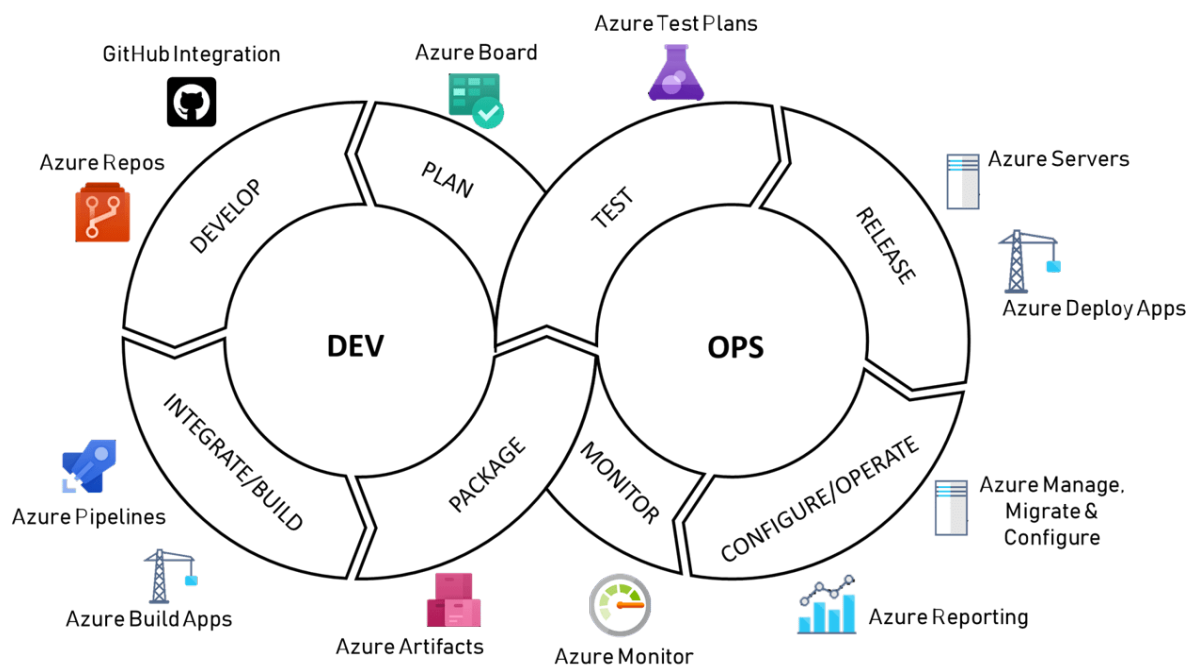
History

Azure DevOps was previously named Team Foundation Server. It was officially announced in November of 2005 and the first version of TFS was released on March 17, 2006. TFS was back then referred to as a workflow collaboration engine that serves as a centralized data warehouse for development project information. Their service has had many releases since then and on September 10th 2018 Microsoft rebranded the service to Azure DevOps.

(11,12)

Why use Azure DevOps?

After hearing about Azure DevOps services their advantages are evident, but if we zoom out again and look at the whole application lifecycle we can see the real benefit of using Azure DevOps from beginning to end of a development process. Through Microsoft's Azure services you have all the necessary DevOps tools and an excellent compatibility with their Cloud Computing services. Azure is the second largest Cloud Computing provider in the world after AWS, and a great choice to host services independently of the usage of Azures DevOps tools. The figure below shows how the whole Azure stack can be applied to every step of the application lifecycle and how it all relates to DevOps.(13)



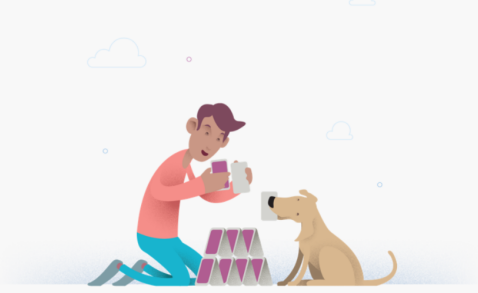
(14)

With Microsoft's recent acquisition of Github in 2018 all tools are provided by the same corporation, which really makes all compatibility issues vanish. Microsoft made some small changes to Github after their acquisition to better integrate with their tools. Some of these changes are that one can access Github through Azure's Active Directory and a Visual Studio subscription that gives access to a Github Enterprise. Also Azure DevOps boards were made available on Github's Marketplace. Also it is now possible to sign in with Github to all the Azure DevOps services, this makes Azure and github closely knit together. (15)

How to use Azure DevOps - Services and Tools

Getting started with Azure DevOps is easy, the first step would be to access the Azure DevOps website and sign up through an outlook account or Github. If it's the first time you log in to the services you will be offered to make an organization, this will make it easy for you to access your DevOps tools through the webpage <https://dev.azure.com/<organization name>>. This will function as a control board for your projects. Once you have created the organization you will be prompted with a screen like the one seen on the image below. This

will allow you to create a project, here you can choose the title, description, visibility and choose between Git and TFVC for the version control. Work item process can be changed based on the project's methodology, here you have the choice between Basic, Agile, CMMI and Scrum. When all fields are filled out you can go ahead and create the project.



Create a project to get started

Project name *

Description

Visibility

☐ Public
Anyone on the internet can view the project. Certain features like TFVC are not supported.

☒ Private
Only people you give access to will be able to view this project.

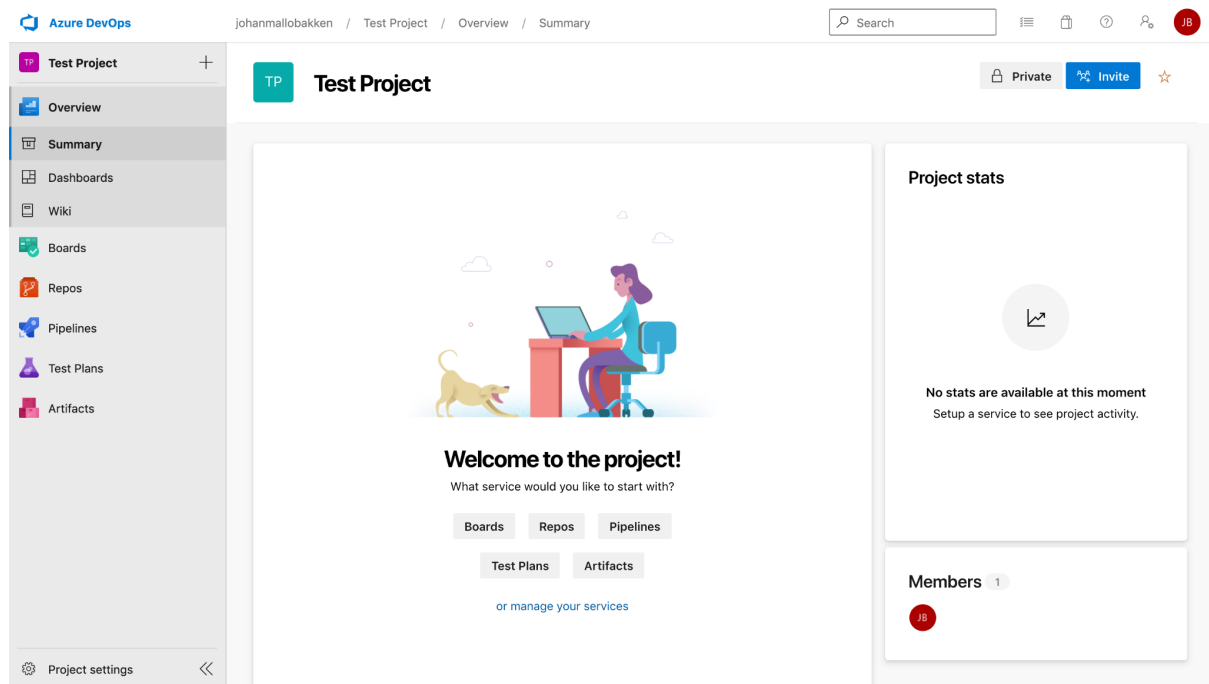
Advanced

Version control ⓘ
Git

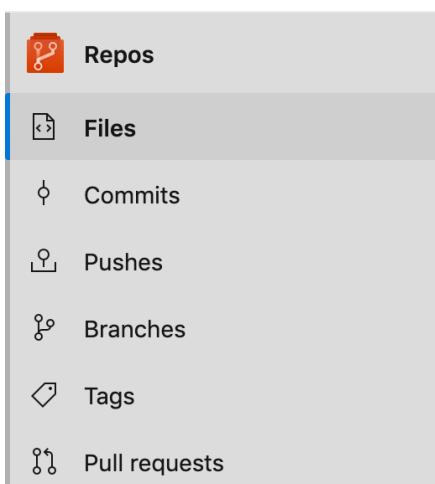
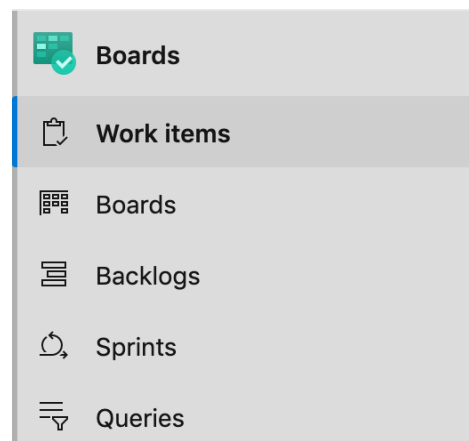
Work item process ⓘ
Basic

+ Create project

Once your project is created it is visible under your organization's projects. When clicking into the project you will be prompted with the projects summary page, this is a customizable page where you can concentrate all the most important information and stats regarding your project. On the left side of the page there is a sidebar which is the main navigation component through the DevOps tools of the project. In addition to the Summary page the overview dropdown also shows dashboards and wiki. The dashboard page can be customized with a set of widgets to gain visibility into the teams progress. The wiki page is meant for the projects wiki to get everyone on the same page.



The DevOps boards section of the sidebar contains five different pages. In the work items page a list of all the projects work items is available. These work items can be created by clicking the plus sign on the top of the sidebar or in the page itself. In the boards section a Kanban style board is located, here you can set your work items in different stages of the development process. In the backlog section you can create new backlogs for the sprints and in the sprints section you have a greater overview and customization of the sprints. In the queries you can group and search the work items with your most precise preference.



In the repos section everything related to version control and Git is located, here you can browse files of the project under the files page. This section gives a great overview of what has changed and when. There is an own page for commits, pushes, branches, tags and pull requests. This makes it a very convenient place to keep track of the source code.

Under the pipelines section you can automate build and release processes by using Azure DevOps wizard. The environments page enables the user to manage deployments and view resource status to get the full end to end traceability. In the releases page all the release pipelines are available. The library page allows you to create and manage variable groups that can be shared across multiple pipelines. Finally task groups help standardize and manage common build and deployment steps for applications while deployment groups allows users to define a logical group of target machines for parallel deployment.



Pipelines



Pipelines



Environments



Releases



Library



Task groups



Deployment groups



Test Plans



Test plans



Progress report



Runs



Load test



Artifacts

Towards the end of the main components of the sidebar the Test Plans and Artifacts are located. Test plans offer everything needed to plan and run tests through the test plan and the run page. Also reports on the progress of a project are available after the tests through the progress report page. With Azure artifacts you can browse and add packages by connecting to package managers through feeds by clicking the connect button.

Competitors

Azure DevOps is a unique set of tools and not many others do exactly the same job, or offers this great variety of tools for so many different purposes. This is both a strength and a weakness, although it's very broad and solves many tasks it has been described as a “Jack of all trades, but master of none”.(16) There are other tools out there that are well known for being good at what they do, and they can be put together to solve the same problems as Azure DevOps. Take for example Jenkins, a well known open-source automation server that facilitates CI/CD, even though Azure Pipelines is able to do mostly the same things as Jenkins, it does not offer the same flexibility. Through Jenkins open-source and extensive community support, it can be applied to a larger number of programming technologies. (17)

In addition to not only the pipelines, the boards and agile methodology aspect is also important to compare with other competitors. For this and more purposes JIRA is a well known tool. Again a similar argument for JIRA being more customizable and having more functionality makes it a popular choice for many, but it's not as easy to use and setup as Azure DevOps is. This argument goes on for most other tools and it looks like Azure's biggest downside is that it is too rigid for many purposes and often will not be the first choice for a project starting out, although it's popular among the established enterprises.(18)

Conclusion

Azure is steadily growing and for every year that passes by Microsoft is taking a bigger role in the world of DevOps. Their mission is to empower every person and every organization on the planet to achieve more and through their innovative CEO Satya Nadella I believe Azure DevOps has a bright future ahead of itself. The power of Azure DevOps is evident, it's easy to set up and simple to learn and use. It has many functionalities and tools which makes it sort of a swiss army knife of DevOps. Although there are many options out there, Azure DevOps has proven to be one of the best and will continue improving as Microsoft continues to pursue their mission to empower others. (19)

Sources:

- <https://www.sqlindia.com/what-is-devops-and-how-devops-is-solving-it-problems/> (1)
- https://miro.medium.com/max/781/1*k7IDfXl_KPhjAOEcf2_VNg.png (2)
- <https://www.guru99.com/devops-tutorial.html> (3)
- https://www.youtube.com/watch?v=JhqpF-5E10I&ab_channel=MicrosoftVisualStudio (4)
- <https://azure.microsoft.com/en-us/services/devops/> (5)
- <https://azure.microsoft.com/en-us/services/devops/boards/> (6)
- <https://azure.microsoft.com/en-us/services/devops/pipelines/> (7)
- <https://azure.microsoft.com/en-us/services/devops/repos/> (8)
- <https://azure.microsoft.com/en-us/services/devops/test-plans/> (9)
- <https://azure.microsoft.com/en-us/services/devops/artifacts/> (10)
- <https://www.eweek.com/development/microsoft-announces-release-of-team-foundation-server/> (11)
- <https://azure.microsoft.com/en-us/blog/introducing-azure-devops/> (12)
- <https://www.statista.com/chart/18819/worldwide-market-share-of-leading-cloud-infrastructure-service-providers/> (13)
- <https://i1.wp.com/digitalvarys.com/wp-content/uploads/2019/07/Azure-DevOps.png?w=1921&ssl=1> (14)
- <https://www.cbtnuggets.com/blog/certifications/microsoft/microsofts-purchase-of-github-one-year-later> (15)
- <https://www.7pace.com/blog/azure-devops-vs-github> (16)
- <https://www.incredibuild.com/blog/azure-devops-vs-jenkins-a-comparative-outlook> (17)
- <https://www.amanzia.se/azure-devops-vs-jira/> (18)
- <https://www.microsoft.com/en-us/about> (19)