Azure DevOps: The only ALM you will ever need?

Johan Mallo Bakken, @johanmallobakken, johanmb@kth.se

An essay submission for the course
DD2482 Automated Software Testing and DevOps

Electrical Engineering and Computer Science
KTH Royal Institute of Technology
April 2021

1 Introduction

Envision a scenario where a development team creates a product, let's say a robot. They work very hard on this robot for a long period of time in their climate controlled and pollution free lab. Once they are done with the robot, they bring it over to the operations team so they can test if it meets the requirements. The operations team brings the robot outside, to test it, but at this point a problem arises. The change in environment makes the robot malfunction. The operations team is left with a lot of work and are clearly annoyed as it seems like the development team didn't do much. This is not a desirable scenario for an organization and will cost a lot of time, money and resources to fix. There is clearly a barrier between the development and the operations team that needs to be broken down.

What if the development team and operations team work together in a common space? The development team could develop each block of functionality of the robot, and the operations team could test this functionality immediately. These teams are now working simultaneously, and when a feature is ready for use it can be outputted to build on the final product. The teams have developed a common mindset and share ideas. This is where Azure DevOps comes in, because to further speed up the process, the teams use a set of tools which can automate every stage of the process. Azure DevOps functions as a swiss army knife that offers all the tools necessary and can be applied to every stage of the DevOps lifecycle. [1]

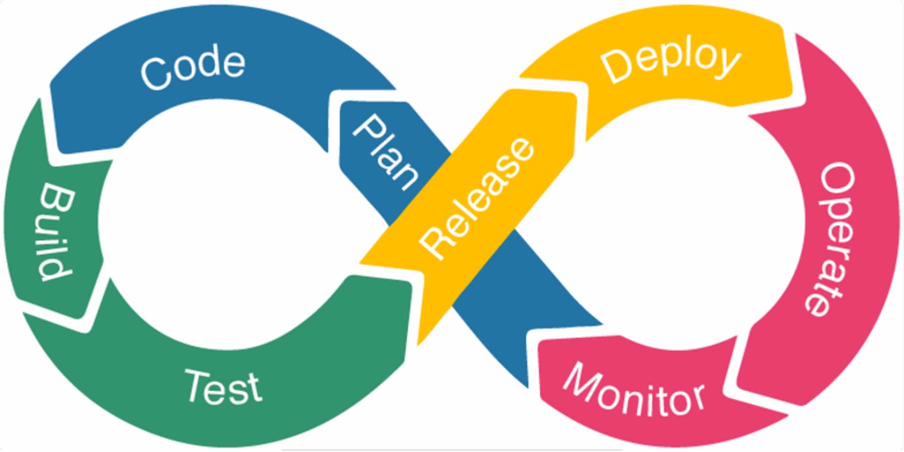


Figure 1: The DevOps lifecycle

[2]

The union of the development team and the operations team is the core of the DevOps approach. DevOps is a way of thinking how software should be developed, which solves many of the issues the old development processes had. The figure above shows the DevOps lifecycle. This is all the stages software goes through when DevOps is used. The left circle covers the development part of the process and the right circle covers the operations part. As you can see in DevOps, the two are closely knit together.

2 Background

2.1 What exactly is Azure DevOps?

As mentioned in the previous section, there are many tools related to DevOps. In this article some of them some of them will be covered, specifically the tools provided by Microsoft's Azure DevOps. More formally defined Azure DevOps is a set of cloud hosted DevOps services that work for any language targeting any platform. These 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. [3] All these tools are packed into Azure DevOps five main services.

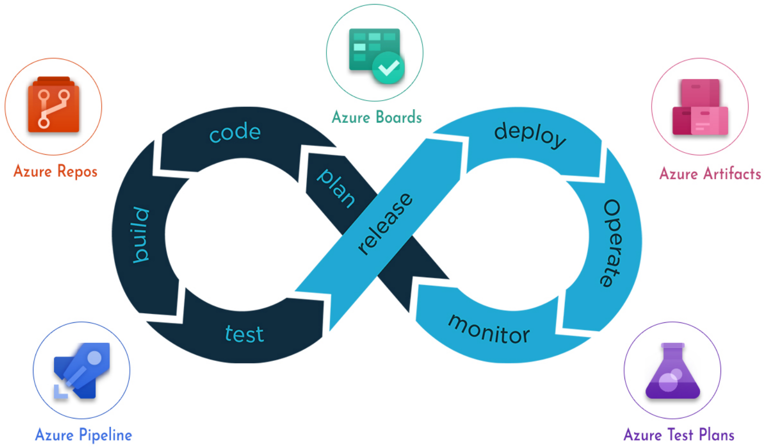


Figure 2: Azure DevOps and its five pillars

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.[4]

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. [5]

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.[6]

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 though manual tests. But also design and execute tests simultaneously through exploratory testing.[7]

Azure Artifacts is for 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.[8]

2.2 History

Azure DevOps was previously named Team Foundation Server (TFS). 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. [9, 10]

2.3 Why use Azure DevOps?

After hearing about Azure DevOps services, their advantages are evident. 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. Both small or a big organization, do not need to kludge together different tool conglomerations when using Azure DevOps. 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. A common misconception is that Azure DevOps only is compatible with projects hosted on Azure, this is not the case, it can be applied to projects hosted on any other cloud computing platform. 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. [11]

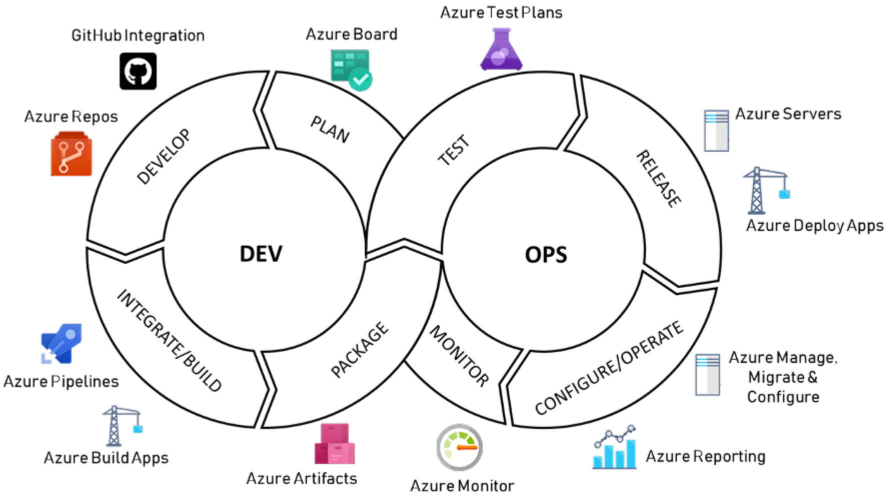


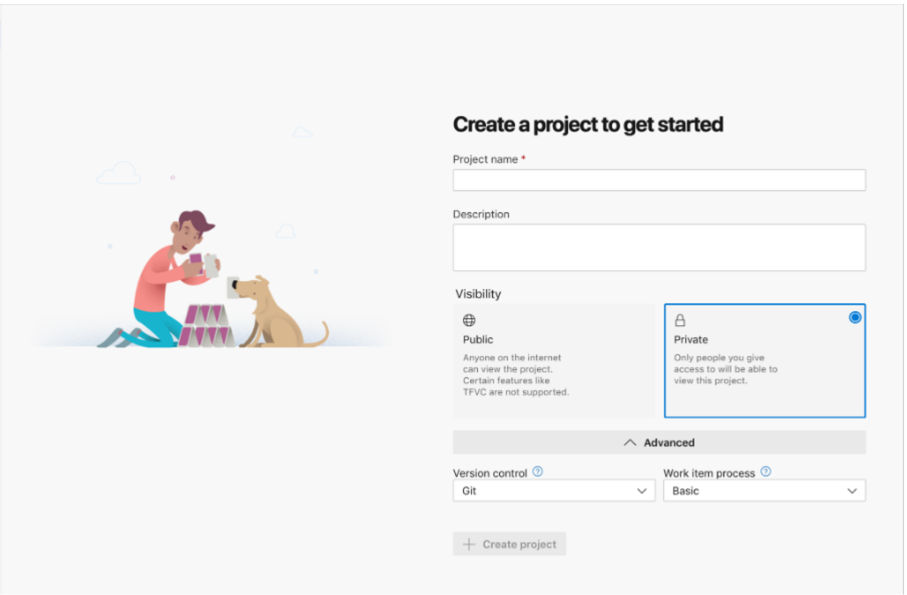
Figure 3: Microsoft's Azure services and their place in the DevOps lifecycle [12]

With Microsoft's recent acquisition of Github in 2018 all tools are provided by the same corporation, which really makes all compatibility issues between them vanish. Microsoft made some small changes to GitHub after their acquisition to better integrate with their tools.[13] In terms of pricing Azure DevOps comes out quite affordable. It's free for teams of up to five people which makes it a reasonable option to start out with. [14] Another reason to start out with Azure DevOps is how simple it is to set up and its ease of use, more about this in the next section.

3 How to use Azure DevOps - Services and Tools

3.1 Quick setup

Setting up an organization and creating a project is easy and can be done in two minutes by accessing the Azure DevOps website. It's very intuitive and you will be guided through the process.



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

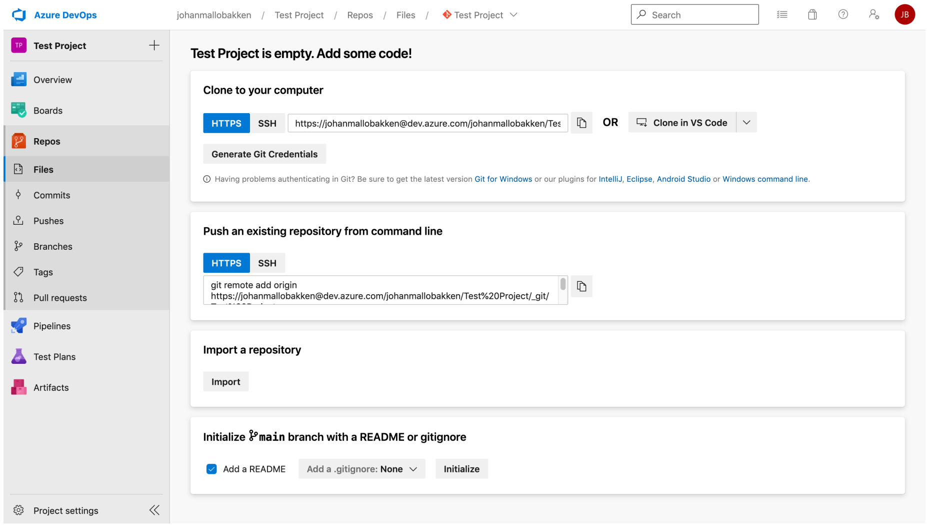
Version control: Git

Work item process: Basic

Figure 4: Project creation page

3.2 Repos

The whole DevOps process is around code, therefore a good place to start is to set up a repository. By clicking on Repos and then Files you will be prompted with four different ways to add code to your project. If you don't have a repository already, Azure DevOps offers the option to initialize the main branch with a README and a gitignore. Here a large list of premade gitignore files is available these are based on the technologies you develop in.



Test Project is empty. Add some code!

Clone to your computer

https://johanmallobakken@dev.azure.com/johanmallobakken/Tes

Having problems authenticating in Git? Be sure to get the latest version Git for Windows or our plugins for IntelliJ, Eclipse, Android Studio or Windows command line.

Push an existing repository from command line

```
git remote add origin
https://johanmallobakken@dev.azure.com/johanmallobakken/Test%20Project/_git/
```

Import a repository

Initialize `main` branch with a README or gitignore

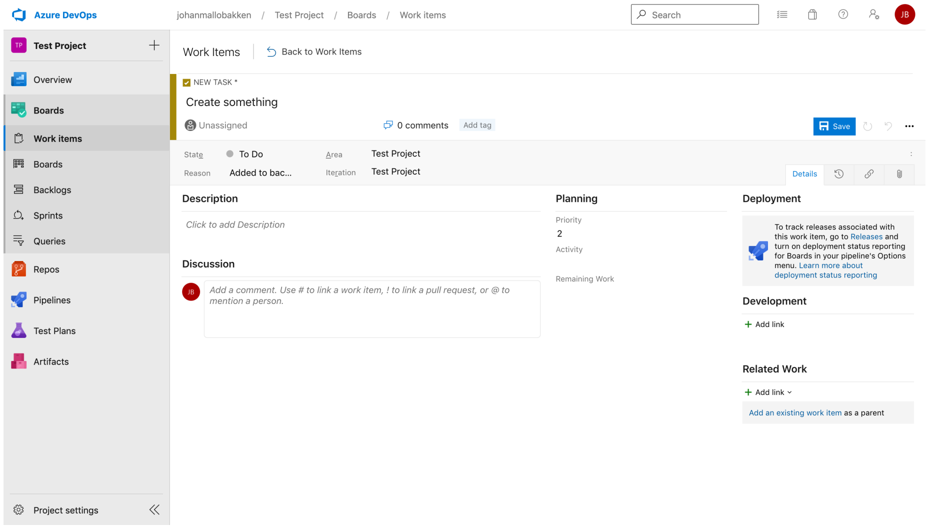
☑ Add a README   Add a .gitignore: None

Figure 5: Ways to add code to an Azure DevOps project

Once the repository is initialized you will get an option to clone it through HTTPS or SSH, just like on Github. The two new files will now be visible in the files page. Now you can go back to the overview section and into the summary page, here you can connect the "About this project" section to the new README file you just created and it will update automatically.

3.3 Boards

Usually before you start coding you will also need to plan. This is what Boards is made for. Here you can do everything from creating Kanban boards to planning Scrum sprints and backlogs. It all comes down to handling work items, which are a central part of Azure DevOps. Work items can be made at any point, by simply clicking the plus button on the top of the sidebar. They can be either tasks, epics or issues and can hold a great amount of information as seen on the image below.



NEW TASK *

Create something

Unassigned

0 comments

State: To Do

Reason: Added to bac...

Area: Test Project

Iteration: Test Project

Description

Click to add Description

Discussion

Add a comment. Use # to link a work item, ! to link a pull request, or @ to mention a person.

Planning

Priority: 2

Activity

Remaining Work

Deployment

To track releases associated with this work item, go to Releases and turn on deployment status reporting for Boards in your pipeline's Options menu. Learn more about deployment status reporting

Development

Related Work

Add an existing work item as a parent

Figure 6: Work item creation page

4 Critical Reflection

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".[15] 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. [16]

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 a tiny bit too rigid for some purposes and often is not be the first choice for a project starting out, although it's popular among the established enterprises, especially the ones already working with the Microsoft stack.[17]

5 Conclusion

So, is Azure DevOps the only ALM you will ever need? Well, although it has most of the features many will ever need for their development needs there are always some niche cases. And for that reason, probably no ALM will ever reach that title. Also, truth being told, products from big corporations are both cursed and blessed in the DevOps world. The community tends to support smaller and more flexible open-source projects. Many would rather not rely on a big corporation for their DevOps tools, although for cloud computing its a whole other world as big corporations offer the highest reliability on their services.

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 many believe Azure DevOps has a bright future ahead of itself. [18] 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 very independent from other technologies. 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.

References

[1] Simplilearn, 2018, *What is DevOps?* https://www.youtube.com/watch?v=mBBgRdlC4sc&ab_channel=Simplilearn

[2] https://miro.medium.com/max/781/1*k7lDfXl_KPhjAOEcf2_VNg.png

[3] Microsoft Visual Studio, 2018, *Introduction to Azure DevOps* https://www.youtube.com/watch?v=JhqpF-5E10I&ab_channel=MicrosoftVisualStudio

[4] Microsoft Azure, 2021, *Azure Boards* https://azure.microsoft.com/en-us/services/devops/boards/

[5] Microsoft Azure, 2021, *Azure Pipelines* https://azure.microsoft.com/en-us/services/devops/pipelines/

[6] Microsoft Azure, 2021, *Azure Repos* https://azure.microsoft.com/en-us/services/devops/repos/

[7] Microsoft Azure, 2021, *Azure Test Plans* https://azure.microsoft.com/en-us/services/devops/test-plans/

[8] Microsoft Azure, 2021, *Azure Artifacts* https://azure.microsoft.com/en-us/services/devops/artifacts/

[9] Darryl K. Taft, 2006, *Microsoft Announces Release of Team Foundation Server* https://www.eweek.com/development/microsoft-announces-release-of-team-foundation-server/

[10] Jamie Cool, 2018, *Introducing Azure DevOps* https://azure.microsoft.com/en-us/blog/introducing-azure-devops/

[11] Felix Richter, 2021, *Amazon Leads 130-Billion dollar Cloud Market* https://www.statista.com/chart/18819/worldwide-market-share-of-leading-cloud-infrastructure-service-providers/

[12] https://i1.wp.com/digitalvarys.com/wp-content/uploads/2019/07/Azure-DevOps.png?w=1921&ssl=1

[13] David Zomaya, 2019, *Microsoft's Purchase of GitHub: One Year Later* https://www.cbtnuggets.com/blog/certifications/microsoft/microsofts-purchase-of-github-one-year-later

[14] Emil Protalinski, 2018, *Microsoft announces Azure DevOps, will succeed Visual Studio Team Services* https://venturebeat.com/2018/09/10/microsoft-announces-azure-devops-will-succeed-visual-studio-team-services/

[15] Tyler Hakes, 2020, *Azure DevOps vs GitHub: Which Toolstack Is Better for Software Teams?* https://www.7pace.com/blog/azure-devops-vs-github

[16] Renana Dar, 2021, *Azure DevOps vs. Jenkins – a 2021 Comparative Outlook* https://www.incredibuild.com/blog/azure-devops-vs-jenkins-a-comparative-outlook

[17] Amanzia, 2020, *Azure DevOps vs. JIRA* https://www.amanzia.se/azure-devops-vs-jira/

[18] Microsoft, 2021 *About* https://www.microsoft.com/en-us/about

All sources have been viewed April 30th 2021