Azure DevOps for PaaS Solutions

Table of Contents

[Preface 3](#_Toc67905127)

[Target Audience 4](#_Toc67905128)

[DevOps for Azure PaaS 5](#_Toc67905129)

[Architecture 6](#_Toc67905130)

[Components 7](#_Toc67905131)

[Considerations 9](#_Toc67905132)

# Preface

# Target Audience

This whitepaper is aimed at developers, architects, operations teams, administrators who wants to understand about Azure DevOps for PaaS solutions, high level architecture, various components involved in it and their considerations, their integration with internal and external systems, advantages and limitations. This whitepaper will not make you a fabulous DevOps person or an architect. There is so much more to software engineering than knowing about the process and toolchain. If you read and understand it further, you should feel comfortable with Azure DevOps principles and applying it to PaaS based solution and free to follow your instincts without too much apprehension.

# DevOps for Azure PaaS

DevOps word in itself is a combination of two words - Development and Operations. It is neither an application nor a tool; instead, it is just a culture to promote development and operation process collaboratively. As a result of DevOps implementation, the speed to deliver applications and services has increased. DevOps enables organizations to serve their customers strongly and better in the market. In other words, we can say that DevOps is the process of alignment of IT and development operations with better and improved communication.

Under a DevOps model, development and operations teams does not work in isolation rather these two teams are merged into a single team where the engineers work across the entire application lifecycle, from development and test to deployment to operations, and develop a range of skills not limited to a single function. This ONE team is focussed on practices to automate software development and deployment processes which were historically manual and slow. They usually leverage different tools and technologies which help them to speed up overall process in a reliable way.

DevOps can be used to in different scenarios like infrastructure (IaaS) based solutions when you need complete control over underlying infrastructure, Microservice based applications where you can share the responsibility of with your development team and cloud providers or PaaS based solutions where a service provider delivers a platform for clients to develop, run, and manage applications without having to build and maintain infrastructure

In the following section, we will look into the details of DevOps model for PaaS based solution.

# Architecture

# Components

Following are some of the core architecture components

* **Source code hosting platform** – Project teams can use different source code hosting platforms; however, GitHub and. Azure DevOps repos are commonly used tools to do the same. These are modern tools which provides many benefits over legacy source code hosting platforms like VSS, Subversion etc. For e.g. –
  + Source code available in GitHub repositories can be easily imported into Azure DevOps repos. Although this is not mandatory as you can still maintain the code in GitHub repo and have CI / CD pipeline configured in Azure DevOps.
  + They support automated builds on commit action or on any specific trigger (pull request).
  + Team can browse the source code without completely downloading it in local environment.
  + GitHub also provides Codespaces which are pre-configured development environment to speed up the development.
* **Development Tool** – Similar to different source code hosting platforms, team can use various development tools like Visual Studio Code, IntelliJ, Visual Studio, Eclipse to perform development activities. Most of these tools support seamless integration with code hosting platforms, code analysis tools etc.
* **Azure Board** – Primary web-based tool to manage software projects. Teams can define user stories / work items, issues, bugs, features, tasks related to the projects and these items can be assigned to individual team members. It also provides rich capabilities for querying work items, integrated dashboard and reporting.
* **Azure Pipeline** – Allows team to automatically build source code, perform unit and integration testing, deploy code to various environments. Azure DevOps primarily provides two types of pipelines – build and release. As the name suggests, build pipeline is used to build the source code from different source code hosting platform in a sandboxed environment. It can be integrated with other Azure services like Azure Key Vault to extract the credentials required to pull the source code from source control systems. Release pipeline runs after the build pipeline and is commonly used to deploy the artifacts generated from build pipeline to different environment. Both build and release pipeline act as a CI/CD solution in Azure DevOps.
* **Azure App Service** – Fully managed PaaS offering to build and deploy web apps and APIs on Azure. It supports multiple technologies including .NET, .NET Core, Java, Python, Node etc. App service provides seamless integration with other Azure services and provides DevOps capabilities like CI/CD, multiple environments, configuration management, custom domains etc.
* **Azure Key Vault** – Fully managed PaaS service for securely storing and accessing secrets. These secrets can be keys, password, certificates etc and can replace secrets stored in application configuration file at runtime, thus keeping secrets fully secured for different environments. Azure DevOps components like build and release pipelines can be integrated with Azure Key Vault service to extract required credentials to complete CI / CD process.
* **Azure Active Directory** –
* Terraform –
* Azure Monitor –
* Azure Policy –
* Azure Resource Groups –
* Azure Subscriptions –

# Considerations