

Edition V 0.0.1 Prerelease version

Copyright © by Bdaya Development 2020, Ahmed Khalil.

A division of Bdaya Development

This book is provided “as-is” and expresses the author view and opinions.

GitHub and Microsoft Azure DevOps for Microsoft Corporation Trademarks

Atlassian Bitbucket Trademark

Circle CI Trademark

GitLab Trademark

The Docker whale logo of Docker Inc and Kubernetes logo

Author : Ahmed Khalil

Table of Contents

[1. Introduction 1](#_Toc56496668)

[1.1. DevOps as-is is a Generic Word 1](#_Toc56496669)

[1.1.1. DevOps is a buzz word 1](#_Toc56496670)

[1.1.2. DevOps for teams 1](#_Toc56496671)

[1.1.3. Benefits of DevOps 2](#_Toc56496672)

[1.2. DevOps and The Application Life Cycle 3](#_Toc56496673)

[1.2.1. Planning phase 3](#_Toc56496674)

[1.2.2. Develop Phase 3](#_Toc56496675)

[1.2.3. Deliver Phase 4](#_Toc56496676)

[1.2.4. Operate Phase 4](#_Toc56496677)

[1.3. DevOps Culture 4](#_Toc56496678)

[1.3.1. Collaboration, Visibility and Alignment 5](#_Toc56496679)

[1.3.2. Shifts in scope and accountability 5](#_Toc56496680)

[1.3.3. Shorter release cycles 5](#_Toc56496681)

[1.3.4. Continuous learning 5](#_Toc56496682)

[1.4. DevOps Practices 6](#_Toc56496683)

[1.4.1. Continuous Integration & Continuous Delivery (CI/CD) 6](#_Toc56496684)

[1.4.2. Version Control 6](#_Toc56496685)

[1.4.3. Agile software development 7](#_Toc56496686)

[1.4.4. Infrastructure as code 7](#_Toc56496687)

[2. Version Control 9](#_Toc56496688)

[Managing Boards 13](#_Toc56496689)

Section 1

# Introduction

Enterprises are in a war of saving costs, solving deployment problems and work for make it fast to release its software into production, improving these concepts come in more fuzzy changes that can get the enterprise to the top or its knees, many tools and methods become more effective in these environments such as Docker, Agile, Git, and many others, these tools and others become a de facto standard for enterprises so let’s take a cruise to a new system development and deployment and production, this cruise taking you in infinity mode, this mode is DEVOPS.

## DevOps as-is is a Generic Word

### DevOps is a buzz word

DevOps is a set of practices that merge tow concepts software development and operations, It aims to shorten the systems development life cycle and provide continuous delivery with high software quality.

Shortening the SDLC lifecycle is an old practice, for methods like JAD (Joint application development) and RAD (Rabid application development) and the old solutions of Agile but DevOps is the stable one currently used

DevOps is complementary with Agile software development; several DevOps aspects came from Agile methodology.

From Microsoft: DevOps is the union of people, processes and technology to continually provide value to customers.

The Value and the Customer is the big words when we talk about business, so this is something about domain, it’s not data-driven.

### DevOps for teams

As mentioned in Azure DevOps docs, DevOps enables formerly siloed roles – development, IT operations, quality engineering and security – to coordinate and collaborate to produce better, more reliable products.

By adopting a DevOps culture along with DevOps practices and tools, teams gain the ability to better respond to customer needs, increase confidence in the applications they build and achieve business goals faster.

It’s a goal-oriented technique for shipping a high performance, agility, quality to achieve the customer satisfaction, Improving collaboration and productivity between teem to go faster.

### Benefits of DevOps

Figure ‑ THE BENEFITS OF DEVOPS FOR TEAMS

As it’s agile way, so it’s a customer-centered way to meet the customer experience and satisfaction.

Agile is monotonous activities for Operation teams, so it’s not effective enough to control the development and operations team, it needs something strong to control like DevOps practices.

Less complexity to manage, the framework is so simple enough to work with.

Faster resolution of problem, once the incident or bug found, the bug is solved so fast by managing them in registers.

Faster delivery of features, based on the stories and sprints like SCRUM, the user story is like a feature to achieve in domain driven by the user needs.

#### Organizations tell the story about DevOps

The numbers tell the story. In their “State of DevOps Report,” Puppet details that high-performing DevOps organizations:

* 200 X more frequent deploys
* 24 X faster recovery times
* 3 X Lower change failure rates

## DevOps and The Application Life Cycle

The SDLC is 5 based cycles, as mentioned DevOps shorten these cycles into 4 plan, develop, deliver, operate phases, each phase relies on the other, and each phase is not role-specific but each role is evolved in each phase for some extent at the true environment of DevOps.

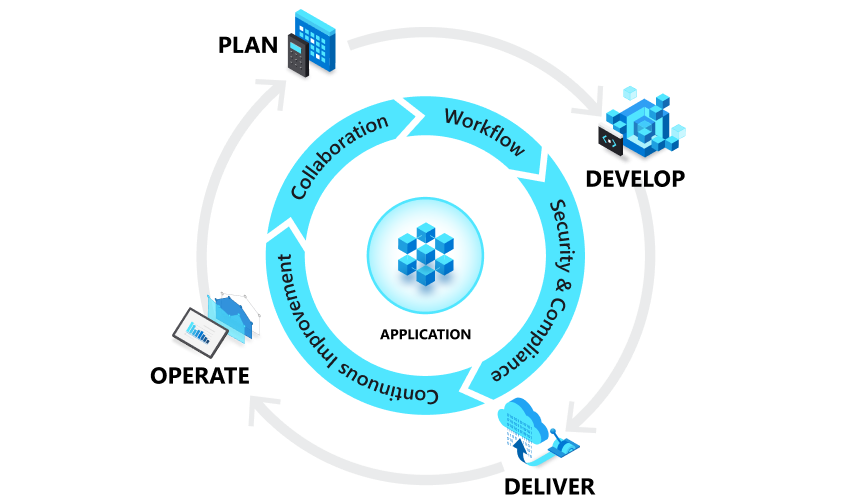


Figure ‑ The lifecycle of DevOps

### Planning phase

In this phase, DevOps team collaborate to ideate, define and describe features and the capabilities of the system they are building , tracking the progress in high and low granularity levels, from single-product tasks to tasks that span portfolios of multiple products.

An enterprise-wide Agile-DevOps transformation typically starts with a Vision followed by many items such as articulation of expected business outcome.

As it’s agile-driven in planning it shares the agile practices of Creating backlogs, tracking bugs, managing agile software development with Scrum, using Kanban boards and visualizing progress with dashboards are some of the ways DevOps teams plan with agility and visibility.

### Develop Phase

This phase include everything about developing, testing, writing, reviewing and the integration of code between teams, it’s relative more about using git and other VCSs to fork, merge, commit, pull, push, for code and project components, as well as building that code into build artifacts that can be deployed into various environments. These techniques is managed from many frameworks like GitHub and other sources like in our organization GitLab.

DevOps teams seek to innovate rapidly without sacrificing quality, stability and productivity.

To do that as mentioned tools they use highly productive tools, automate mundane and manual steps, and iterate in small increments through automated testing and continuous integration.

### Deliver Phase

In other concepts of life like to deliver orders to customers in their homes, also the deliver phase is about sending the tested code as running app in production, and it ready to consume from users.

Deploy to production environment means that app is stable enough and reliable to consume from customers.

The deliver phase also includes deploying and configuring the fully governed foundational infrastructure that makes up those environments.

These days the biggest tool use and become a de facto for most of DevOps base system is Docker environment, based on containerization, virtualization approach. And we will talk about it in next sections.

In the delivery phase, teams define a release management process with clear manual approval stages. They also set automated gates that move applications between stages (Development, QA stage, Production, Retire) until they’re made available to customers.

Automating these processes makes them scalable, repeatable, controlled. This way, teams who practice DevOps can deliver frequently with ease, confidence and peace of mind.

### Operate Phase

To make the production more stable, resilience. The Operate phase work for maintaining, monitoring, troubleshooting application in production environment. This phase aims to zero downtime while reinforcing security and governance.

DevOps teams seek to identify issues before they affect the customer experience and mitigate issues quickly when they do occur. Maintaining this vigilance requires rich telemetry, actionable alerting and full visibility into applications and the underlying system.

## DevOps Culture

While adopting DevOps practices automates and optimizes processes through technology, it all starts with the culture inside the organization - and the people who play a part in it.

A big challenge is required to change, It deeply changes the work of team members and how the collaborate, But when the organization take the step, the team become more productive and high performed.

### Collaboration, Visibility and Alignment

The healthy DevOps environment come from visibility, The IT Operation team and the development must share their DevOps processes, priorities and concerns to make the whole environment more stable.

These teams must also plan work together as well as align on goals and measures of success as they relate to the business.

### Shifts in scope and accountability

As mentioned the DevOps is not a man job, but the whole team job, the environment is shared, the team is align so the take the ownership and become involved in additional lifecycle phases – not just the ones central to their roles, for example the developers become accountable in all phases not only the plan and develop but also the deploy and operate because the performance and stability is related to the code and the entire production environment.

At the same time, IT operators must include governance, security and compliance in the plan and develop phase.

### Shorter release cycles

DevOps is based on agile techniques, which are based on cycles like SCRUM sprints and many other frameworks in agile adapt these cycles, these cycles have incremental behavior, so the progress is incremental, these cycles make it easier to risk management and planning, which is also reduce system stability.

Shortening the release cycle also allows organizations to adapt and react to evolving customer needs and competitive pressure.

### Continuous learning

DevOps is like a journey, it’s like a room to grow, when the high performance DevOps team establish their growth mindset, the fail fast and incorporate learnings into their process.

This behavior can continually improving, increasing customer satisfaction, and accelerating innovation and market adaptability.

## DevOps Practices

To bring DevOps to life the DevOps team implement certain practices throughout application lifecycle. To accelerate, automate, and improve specific phase.

### Continuous Integration & Continuous Delivery (CI/CD)

Continuous integration is a software development practice in which developer merge code changes frequently into the main code branch. And it employ the continuous testing or the automated testing using some configurations, which run every time the code committed so the code in the main branch is stable.

Continuous delivery is the frequent, automated deployment of application into production environment, this accelerate the deployment strategies and reduce issues, as it frequent pattern of deployment to the same production environment.

This is full automation for organizing development code, testing it and release the app into production, this make the team focus more on building code and removes the overhead of human error in manual and mundane steps.

This make it quicker, less risky. For smaller increments means to more agility, more confident, more productive in their running code.

### Version Control

Version control is a practice to manage the code into series of versions, as a practice, tracking revisions and change history to make code easy to review and recover. Like any system the user consume, it have many version of release and each number has its own features.

The practice is usually implemented using Version Control System like Git which allows multiple developers to collaborate and authoring code. The VCSs usually contain many tools to merge, pull, and manage the process in a same file of development, manage conflicts and roll back changes in early state of failure.

This practice is main fundamental for all DevOps team, help them to work together, divide the tasks between teams, and store all codes, so it easy to recover.

Version control is also a necessary element in other practices such as continuous integration and infrastructure as code.

### Agile software development

Agile is a modern approach for software development that emphasizes team collaboration, customer and user continuous feedback, and high adaptability to change through short release cycles. The feedback from users is the main change director adjusted from the (need and want) of business today engines.

Agile is so different from other traditional frameworks like waterfall or RAD, they have a long release cycles defined by sequential phases.

Kanban and SCRUM are tow most popular frameworks associated with agile.

### Infrastructure as code

IaC defines system resources and topologies in a descriptive manner that allows the team to manage these resources as they code.

Those definitions can also be stored and versioned in VCS,

Section 2

# Version Control

Section 2

# Managing Boards

Section 5

Working With Docker

# Introduction to Docker

# Docker Compose With Aspnet Core and SQL Server

In this chapter, containerizing an Aspnet Core 3.1 app and Sql Server and how to run the migrations as a part of deployment process.

The Aspnet core 3.1 app is Swagger enabled for testing and development purpose, so let’s start working to containerize these services.

First Dockerfile:

FROM mcr.microsoft.com/dotnet/core/aspnet:3.1-buster-slim AS base

WORKDIR /app

EXPOSE 80

EXPOSE 443

FROM mcr.microsoft.com/dotnet/core/sdk:3.1-buster AS build

WORKDIR /src

COPY ["MyApp.API/MyApp.API.csproj", "MyApp.API/"]

RUN dotnet restore "MyApp.API/MyApp.API.csproj"

COPY . .

WORKDIR "/src/MyApp.API"

RUN dotnet build "MyApp.API.csproj" -c Release -o /app/build

FROM build AS publish

RUN dotnet publish "MyApp.API.csproj" -c Release -o /app/publish

FROM base AS final

WORKDIR /app

COPY --from=publish /app/publish .

ENTRYPOINT ["dotnet", "MyApp.API.dll"]

