12-2022

¬¬Challenges and Solutions in the Implementation of DevOps Tools & Security (DevSecOps): A Systematic Review

Gautam Bollieddula
St. Cloud state university

Challenges and Solutions in the Implementation of DevOps Tools & Security (DevSecOps): A Systematic Review

by

Gautam Bollieddula

A Starred Paper

Submitted to the Graduate Faculty of

St. Cloud State University

In Partial Fulfillment of the Requirements

for the Degree of

Master of Science

In Information Assurance

December, 2022

Starred Paper Committee:
Akalanka B. Mailewa, Chairperson
Mark Schmidt
Erich P. Rice

Abstract

DevOps (Development & Operation) is a set of practices that combine software development (Dev) and IT Operations (Ops). It aims to shorten the Software Development Life Cycle (SDLC) process by providing Continuous Integration (CI) and Continuous Deployments (CD). The effort to increase Security in DevOps has resulted in the DevSecOps paradigm. Which is a set of practices, cultural approaches, and tools that bring together software development (Dev), IT Operations (Ops), and security (Sec) to increase an organization's ability to deliver applications and services at high velocity securely. We conducted a Systematic Literature Review of 54 peer-reviewed studies. The Thematic analysis method was applied to analyze the extracted data. We identified challenges related to adopting DevSecOps, Solutions, and Integration. We also identified key gaps by evaluating various solutions against the challenges. The results of the study were classified into DevOps tools and Security tools.

Key Words: Azure, DevOps, DevSecOps, security, systematic literature review

Table of Contents

Page

Chapter Page

List of Figures

Figure Page

CHAPTER I: INTRODUCTION

The term DevOps has come to be applied in many ways. Organizations will use the term 'DevOps' or 'DevOps culture' to mean a certain software development environment where a tight-knit community of engineers, testers and operations personnel share the same goal of continuous delivery. This can take the form of an agile methodology; an open-source toolkit; or even a loose organizational construct. Everything from microservices architecture to interaction with the customers ought to be considered. The term "DevOps" is beginning to show up more and more in sources such as news articles, blogs and marketing materials as businesses recognize that DevOps can help them deliver high-quality software faster with less risk of failure.

A. Problem Domain

Before the Development of DevOps, there was a tendency of organizations to divide work and communicate. Work would be completed by one team and then passed on to another team and then at the end of the project, the security team would be clued. The lack of communication caused confusion and conflict between teams, slowed down production time and introduce more vulnerable products to the consumers affecting the path of value for companies. When software developed in a non DevSecOps Environment security related problems can lead to huge delay and compromise of Assets includes Critical Data (PHI & PII) [1]. This article is intended for organization who are planning or in the process of adopting DevSecOps to be aware of the frequently reportedly problems in this domain.

B. Problem Statement

A security expert and consulting firm, iSEC Partners conducted a survey of software vulnerabilities between January and March and found that NIST estimated that 4.1 billion records are at risk of being compromised because they are not properly protected with encryption, authentication, or other safeguards. A reported 88% of respondents have seen attacks against their software over the past 12 months [22]. The National Institute of Standards and Technology (NIST) says almost 450 million records were exposed due to insecure web applications in 2017 alone [2]. The most recently released NIST Special Publication 800-53 Cyber Security Framework (CSF) is intended to help organizations identify risks, develop controls strategies, and build a roadmap towards cybersecurity maturity using an enterprise architecture perspective [23]. The time it takes a vulnerability to be discovered and disclosed is shrinking. More vulnerabilities are being discovered that affect multiple vendors. Vulnerabilities are being uncovered using new techniques such as reverse engineering or fuzzing. The integration of security into the concept of DevOps has led to the development of DevSecOps whereby at the core, there is the principle of keeping security controls and practices into the DevOps Cycle.

C. Research Questions

To give this research paper a framework, there have been research questions that have been developed that will act as a guide for the research work. This will be critical in solving the problem in question. They are as follows:

1. What are the specific challenges related to adopting DevSecOps?
2. What are the solutions proposed during implementations?

3. What are the opportunities for future research?

D. Objectives and Scope

This study aims to systemize the knowledge about the challenges faced by organizations when adopting DevSecOps and proposed solutions reported in the literature. We also aim to identify the areas that need further research in the future. To achieve automation, security processes and tools must be aligned with the specific needs of the organization. As DevOps is a relatively new activity in software development organizations are still at a phase where processes are in a transitional state. DevSecOps is an umbrella term for approaches for collaboration between IT teams and developers to improve security before, during and after development, deployment, and operation of software products [24]. It therefore requires the alignment of processes and tools from across an organization regardless of their location.

E. Research Contributions

1) *Early adoption of the concept of DevSecOps:* This research contributes to the early adoption of the concept of DevSecOps and the systematic peer reviews that analyze the adoption problems. There is a presentation of empirical research that provides a platform to test the early adoption of DevSecOps as a systematic approach and to assess the effectiveness and suitability of this business operation.
2) *Finding solutions in terms of guidelines, framework, tools, and technologies:* Additionally, it contributes to the growing prospect of finding solutions in terms of guidelines, framework, tools, and technologies. Comprehension of testing and validation mechanisms has not been a priority, despite their importance in information security management. The

detection and prevention of IT threats is paramount, but as stated by IT security professionals, "good security without DevSecOps is like locking the stable door after the horse has bolted." Many organizations have become aware that they need to develop DevSecOps to prevent attacks through DevSecOps methods.

3) *Future Studies:* It is also without a doubt that this is an area of interest that is growing within the research community and that for the purposes of future studies research gaps should be identified. This is in addition to the tools being analyzed here.

F. Expected Results

1. There are loopholes in the DevOps systems in the current situation.

2. DevSecOps is an efficacious and effective way to improve DevOps quality, efficiency, and performance.

3. The DevSecOps approach has potential to increase productivity, performance, and security of applications as well as reduce the overall cost of software supply chain procurement by reducing the time to market.

4. There is a need for a systematic approach to integrate the principles of DevSecOps with the concepts, patterns, and tools based on systems thinking; with this approach we can achieve better results in terms of development processes and security of applications with minimal efforts on the part of analysts and developers in both independent teams associated with each environment (DevOps/IT Operations) working on their specialized tasks.

CHAPTER II: BACKGROUND (LITERATURE REVIEW)

In this section we define tools install, Security tools integration and concepts used in this study. The first part of DevSecOps is:

A. Development and Version Control

These are one of the key tools. In this context we are referencing tools Azure repos, Azure Boards. And in Operations, Continuous Integration and Continuous Delivery are key parts and Azure pipelines fulfill this part. Security tools integration includes SAST (Static Application Security Testing) Integration with Devops [3]. This is a set of tools that are being used to deploy and integrate security into the DevOps lifecycle.

B. Azure Pipelines

Azure Pipelines is a continuous delivery service that allows teams to rapidly deliver software-based changes like code and configuration updates, web pages, and more to their application. It has a built in "blue/green" deployment feature that automatically deploys code to actively running servers when it is ready [25]. In our study we focus on Azure Blue/Green deployment strategy for an individual repository as an example. In this case the team is deploying a change to a running server. DevOps process: The cloud marketplace is full of tools for improving software delivery processes including PaaS (Platform as a Service) offerings from Google and Microsoft [4]. We narrow our research focus to Azure Pipelines because of its integration with Visual Studio Team Services, the Microsoft ecosystem, and its popularity in different industries such as finance, media, manufacturing, and others. We do not consider GitHub's CI or Jenkins as deployments services because of their limited scope for automating tasks other than build.

C. Azure Boards

Azure Boards is an interactive application that teams use as a central hub for their DevOps workflows. It allows all team members to manage software and test deployments, track time and resources, organize tasks, and generate reports. Azure boards integrates with GitHub, Bitbucket, GitLab and Visual Studio Team Services to capture a full-fidelity record of work items and code changes [26]. With the ability to create new work items from commits, pull requests, deployments and other actions, Azure Boards keeps teams connected with their source code. It is used to integrate with CI/CD, it can be used for visually monitoring the status of builds in CI/CD servers. DevOps Azure Boards is designed by developers, for developers.

Every user enters their own account and team, which is used to store their own work and records. DevOps Azure Boards allow the communication of bugs, tasks, and other interesting matters between users within a team. All information is stored in a private gallery of work items that either user can edit or manage [27]. An administrator can always be contacted regarding overall functionality or any other issues relating to the application. DevOps Azure Boards records every small step that completes a task or bug report [28]. The collected information enables complex analysis tools to backtrack how projects were developed, track time spent on specific tasks, identify bottlenecks in development cycles and provide feedback on potential issues with the application.

D. Azure Repos

Is the service that makes all this possible by providing version control, continuous integration (CI), release management (RM), and DevOps services that allow you to scale your

development team without scaling your infrastructure. Azure Boards in the study, we define a scenario where DevOps are ready to deploy code changes to servers by using Azure Boards. So, they need an Azure Repo that they can link with the source Git repo and the build pipeline on Azure Pipeline. They also need an Azure Board that is used as a Scrum Master application.

E. Continuous Integration

Continuous Integration (CI) build automation is a process that automates building, testing, and deploying software components to production under defined conditions (for example, every time a developer commits code). CI can be performed entirely using a build server, or the build server can pass some tests to the integration environment (environment used by Continuous Integration tools), which uses a VM. The development team uses a build (also called check-in) tool (such as Jenkins) [29] which monitors the source code repository and triggers builds when changes are made to the source code [5]. When every developer uses the same software version in their local environment, builds usually succeed and tests are successful. However, if developers use different versions of software from different sources and then submit those changes to a shared repository, builds will fail.

F. Continuous Delivery (CD)

This means making changes continuously and delivering them automatically to end users; it is a service offered by providers like Microsoft in repos Blue Az. Continuous Delivery involves having an automated continuous unit test system in place prior to release and includes a process that verifies every change made to the app. Additionally, it uses automation to build and deploy changes to a production instance (server) before release.

G. DevOps Tools and Security

DevOps Security or DevSecOps is set of practices, cultural approach that brings together Development, Operations, Security to achieve secure application development and increase organization's ability to deliver applications and services securely. One of the DevSecOps tool is Fortify SCA, which was recently acquired by HPE and branded as HPE Fortify SCA Integrity [6]. It is a static application security testing tool, that analyses android and java applications to discover and report security vulnerabilities. Some of the security risks faced by organizations are due to the increase in the number of third-party software applications such as libraries and frameworks used in a organization's code base [30]. These applications can introduce vulnerabilities into an organization's apps. Thus, it is important to identify these vulnerabilities at an early stage, that are not present in the original application code itself. It is difficult to check whether these vulnerabilities are present in the organization's code base or not. Using test automation, one can easily check the application code to ensure that it is free from any vulnerability before deployment [31]. It also comes in handy when a vulnerability is reported after an application has been deployed, on-premises/cloud.

To build a secure internet of things, organizations will have to consider leveraging "IoT gateway", which takes advantage of IoT security tools and standards such as ISO27001 and IEC 62443. As an example, the Link It One IoT Gateway leverages Fortify SCA for security testing of IoT gateways by enabling baseline scanning and risk assessment functionality for IoT gateway devices. "SaaS" is a software as a service [32]. It is one of the most common ways to deliver applications, which runs on a cloud platform. Although the security of SaaS application is still relatively more complex than traditional enterprise applications, it is getting better with use of

modern tools like Fortify SCA that allow quick and easy application scanning. Scans are done on the target web server. It finds vulnerabilities such as XSS, SQL, OSI Layer 7 – web application attack surface and many more by leveraging the methods of code analysis in Fortify SCA.

H. Fortify Static Code Analyzer (SCA) Static Application Security Testing

Fortify SCA is a tool that integrates with Azure pipelines and analyze the source code and identifies the vulnerabilities and provide guidelines to mitigate the issues based on the industrial compliances like healthcare follows FISMA (Federal Information Security Management Act) guidelines and OWASP (Open web applications security project) top 10 issues. Fortify SCA integrates with existing IDEs like Visual Studio, Eclipse and scan the code and generate reports of various threats in the application [33]. It can be integrated with Azure pipelines that scans the build artifacts to generate report on issues based on priority level in analyzer window. Fortify SCA provides many languages for threats assessment like Java, .Net, Ruby, Perl etc.

Fortify's Dynamic Code Analyzer helps you quickly find vulnerabilities in your code. The IDE plug-in enables real-time security analysis as you build applications in C#, C++, Java, PHP and more [34]. Robust security analysis lets you deliver better software, faster. Fortify SCA provides threat level on different categories like SQL Injection, Cross Site Scripting, Broken Authentication and Session Management etc. Fortify SCA offers different types of scanning which includes –

1) Static Application Security Testing (SAST) scans for code issues at the application source. It identifies the potential vulnerabilities in the code like SQL Injection and concludes with the mitigation techniques to handle them. SAST scans the static application only such as web

applications or web services [37]. It can be integrated with Azure pipelines using build task to integrate with analysis engine performing after successful builds in VSTS /Azure DevOps pipeline.

2) Dynamic Application Security Testing (DAST) scans for the vulnerabilities of web applications such as ASP.Net, JAVA and WordPress websites at the runtime using web application firewall rules. DAST scans through web request and generates location on the application where it might be vulnerable to threats like SQL injection, XSS threats etc. [36]. If it is an .net application, then it performs Code Analysis and provides feedback where you can find an issue in your code to fix them with minimum effort.

3) Analysis of third-party libraries: Fortify SCA can analyze third party libraries that might have vulnerabilities before they are used in your application. If you are using a third-party library with your code, then you should manage the library to be secure before use in your application.

Fortify SCA uses from static and dynamic analysis. For static analysis it scans the code, analyzes the security issues, and find out potential vulnerabilities

For dynamic analysis it scans at runtime for potential vulnerabilities in web request. Fortify SCA integrates with Azure DevOps pipelines VSTS) for analyzing the build artifacts like JRE, JVM, PERL etc. [35]. It provides real time scan of applications to check if there is any security issue found in them. Fortify SCA scans through scanner engine and generates report based on priority level as per defined rules.

CHAPTER III: METHODOLOGIES

We included multiple studies to capture all relevant information for our tool's integration. Azure DevOps provides Services including Azure Boards, Azure Pipelines, Azure Repos, Azure Test plans, and Azure Artifacts are collectively known developer services which allow teams to plan work through Agile Boards, collaborate on code development through Azure Repos, Build and deploy applications through Azure Pipelines. This collective Azure Services can be installed on-premises or consumed through the cloud.

Azure DevOps provides Services includes Azure Boards, Azure Pipelines, Azure Repos, Azure Test plans and Azure Artifacts collectively known developer services which allows teams to plan work through Agile Boards, collaborate on code development through Azure Repos, Build and deploy applications through Azure Pipelines. This collective Azure Services can be installed on-premises or consumed through cloud. Azure Devops has its own DevSecOps Framework which can be used by organizations to secure their Azure infrastructure, security, and engineering practices. The Azure Devsecure Framework contains four important components: DevSecOps Organizational Model, DevSecOps Process Model, Developer Security Workflow and Developer Tools. The main aim of this framework is to enhance the digital identity of developers in the organization using their identity. This will allow organizations to define security controls based on the need of different roles within the organization and use them for evaluating risk caused by code vulnerabilities.

A. DevOps Tools Install On-Premises

Azure DevOps Services on-premises: The installer places executables on our servers and runs an installer. The Configuration steps get all the features for our installation.

The system requirements for single server minimum 4GB Ram which supports up to 250 users and supports Windows Server Operating system 2019 & 2016 [7].

When the installation finishes the installer starts the Azure DevOps Server Configuration center and start wizard to install Application tier. Choose Configure Azure DevOps Server and then choose start wizard

The Azure DevOps Server Configuration center runs under the configuration management (CM) domain and the user that runs it is created specifically to run this tool. The App Center administrator and the service administrator, both have access to this tool as well. Application Tier Installer has a configuration center for application tier which contains all the modules of Application tier installation [38]. It also includes 20 pre-built services, each with a setup file that can be used to install all of those components if they are part of an application. Alternatively, one can copy the setup files into their own directory and use them manually.

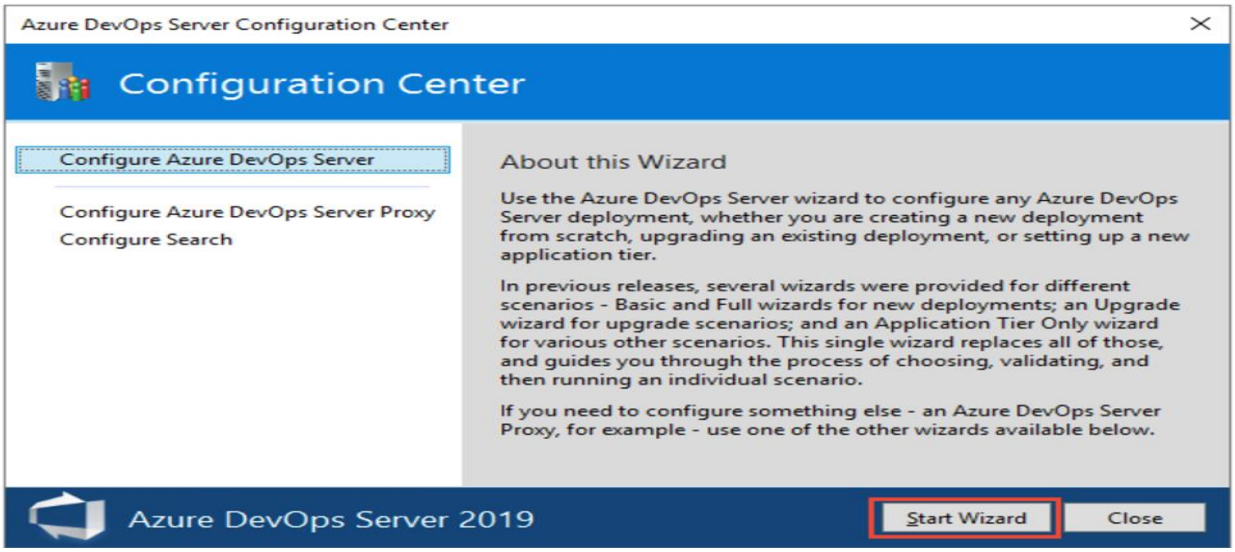

Fig. 0.1. Azure DevOps Server Configuration Centre

After the setting up of application tier, Configuration Tier installs the Network service, which manages Azure DevOps Server configuration. Configuration tier creates a resource as well. When it creates the resource it creates a new container in Azure Active Directory with name "DevOps" [39]. In Azure DevOps Server there are several features like Report Manager, Log Service and Scheduled Tasks for desktop automation using PowerShell.

The Log Service is used to store logs for Azure DevOps Server and configure alert thresholds for tickets.

It also has a scheduler which automatically runs tasks anytime there are changes to different aspects of Azure DevOps server.

SQL Server Instance: Select the SQL server instance as this is for implementation. I did select SQL Server Express.

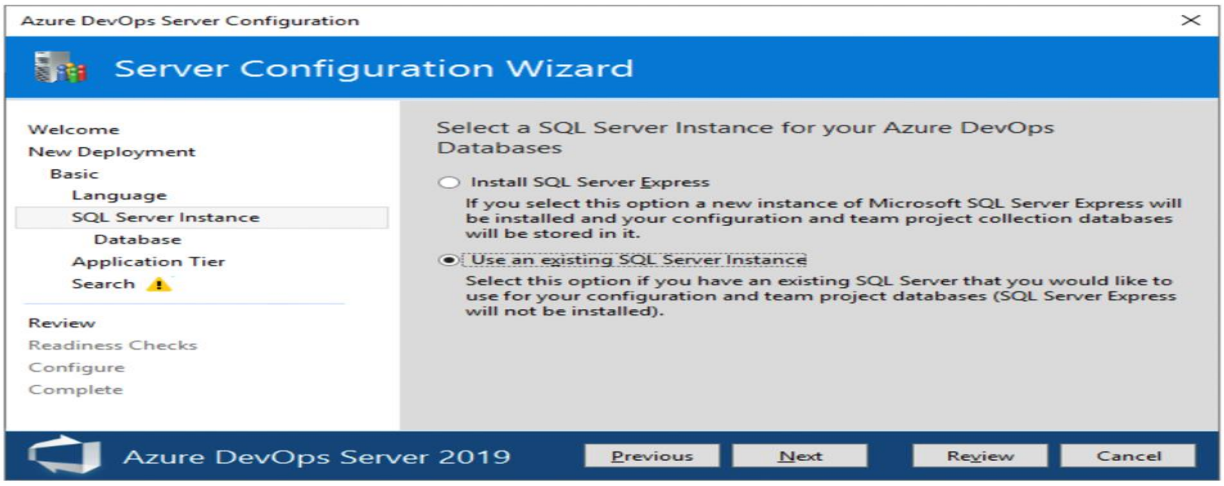

Fig. 0.2. Azure DevOps Server Configuration

Application Tier: Choose the web site settings which includes whether to use HTTP or HTTPS bindings. Use service accounts for production environments and click on review for readiness check.

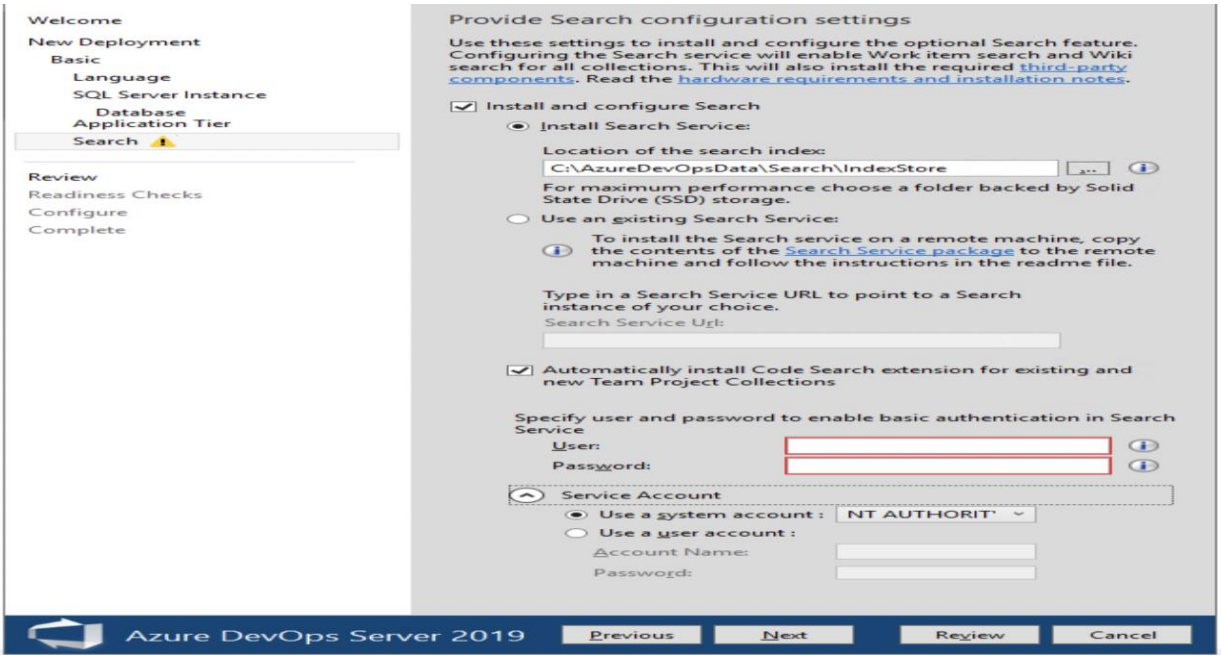

Fig. 0.3. Provide Search Configuration Settings

When checks have passed click on configure:

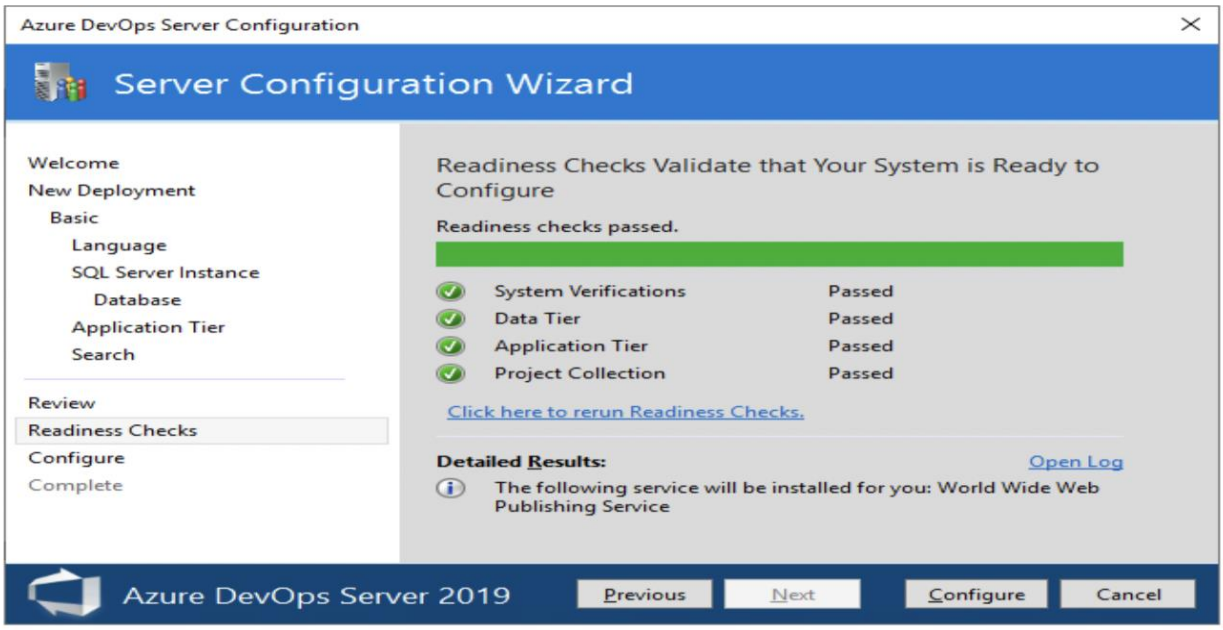

Fig. 0.4 Configuration validation readiness

When configuration completed it shows completed successfully.

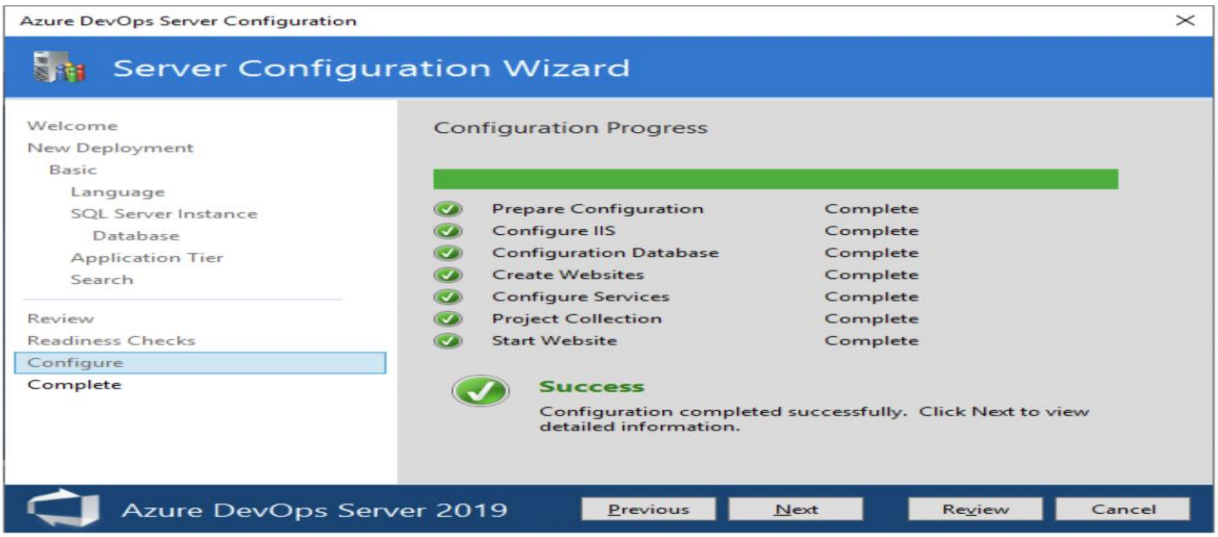

Fig. 0.5. Configuration Progress

B. DevOps Tools Cloud Services

Azure DevOps Services Cloud solutions allow to create an account using outlook email and use the services and its pay per usage. There are different services for various purposes and a lot of configuration options. There is also a pricing calculator which enables you to find the best deals for DevOps Services.

There are many features available within the Azure DevOps Services Cloud, that make this an ideal cloud solution for developers and those who do DevOps with other applications or tools. Some of these features include Continuous Integration; Continuous Delivery; Application Lifecycle Management; and Containerization with Kubernetes on Azure cloud. Microsoft have also announced that they will be releasing continuous integration/continuous deployment workflows soon.

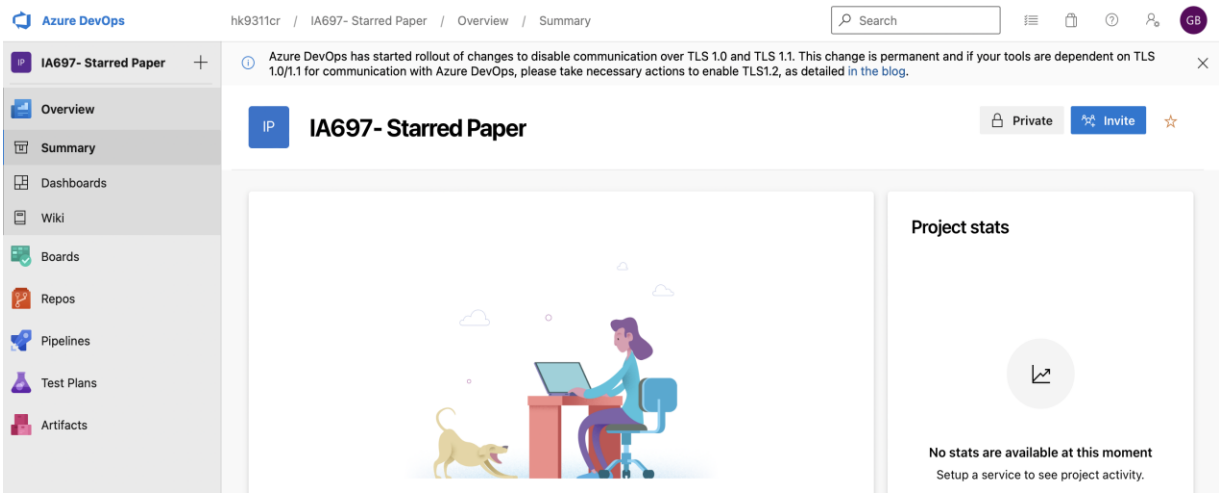

Fig. 0.6 Azure DevOps Services Cloud

C. Azure Repos

Azure Repos are set of version control tools to manage code. Version control system helps to track your changes in the code. As we edit the code version control system take snapshot of the files and this history is maintained permanently and at given point of time you can go back to history to view your changes.

Azure Repos provide two types of version controls

- Team Foundation Version Control (TFVC) Centralized version control
- GIT Distributed version control

Team Foundation Version Control (TFVC): TFVC is a centralized version control system. For this paper I am selecting TFVC repository. TFVC also provides access controls to enable granular level permissions and restrict developers to access on projects they are working or files they are assigned. Below picture I am selecting TFVC repository.

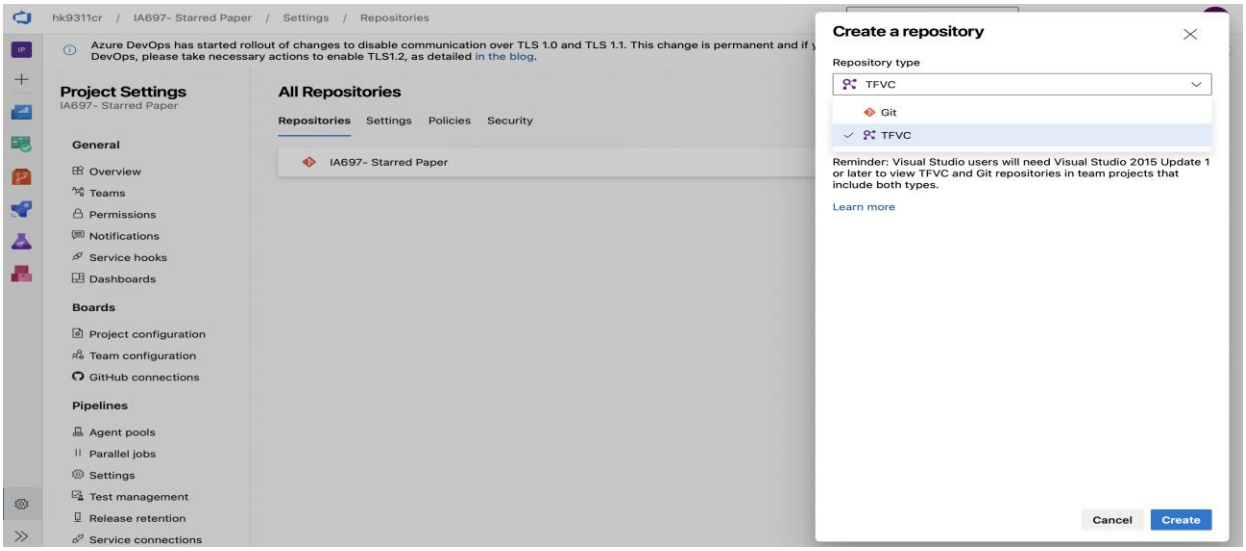

Fig. 0.7 TFVC Repository

I created my project as starred paper22 and created a file with security control access levels.

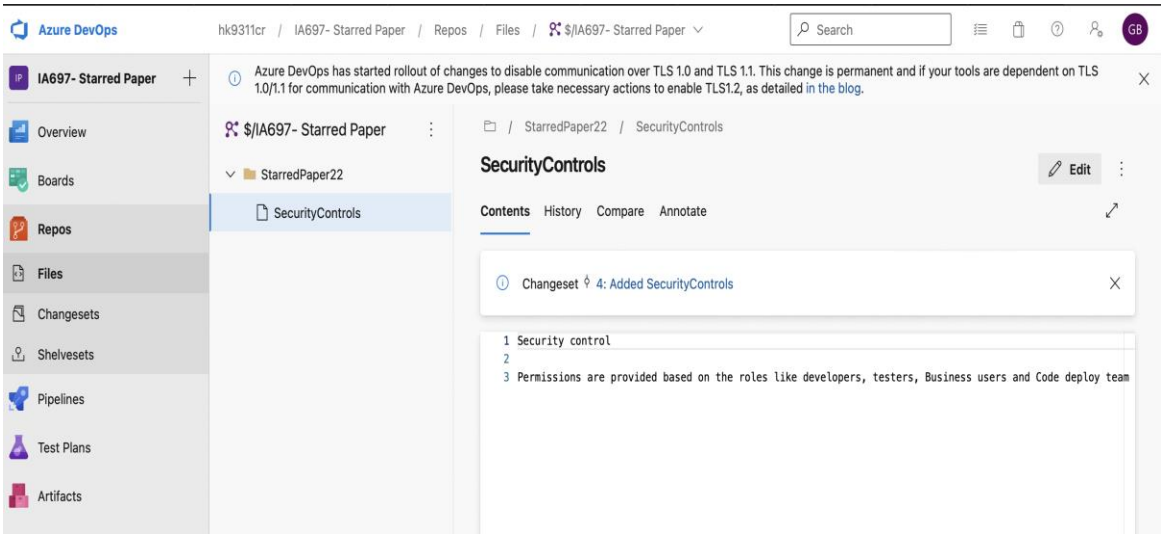

Fig. 0.8. Security Control Access Levels

The picture below shows version control history. At any given point I can go back and get the code that I started in the beginning and do the development again.

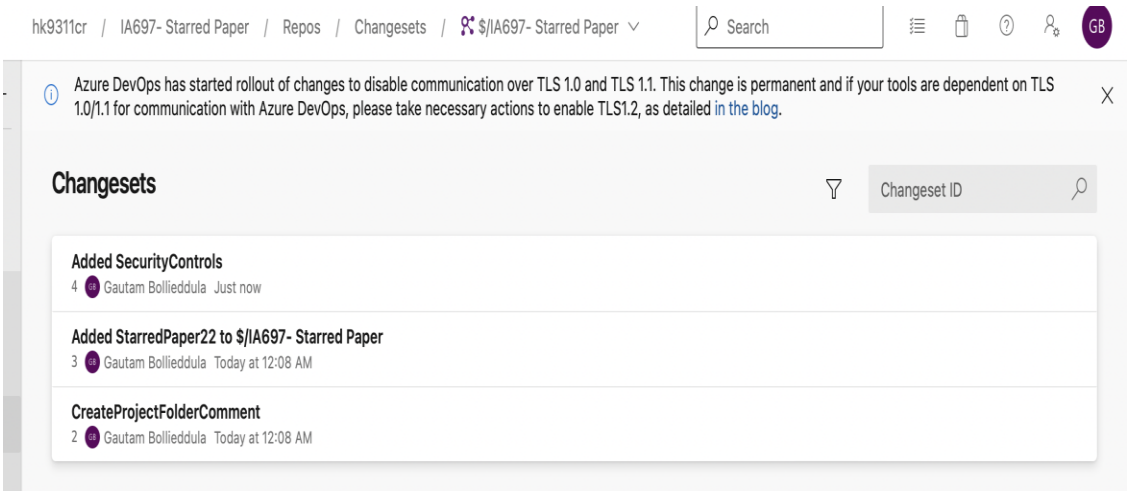

Fig. 0.9. Version Control History

D. Azure Boards

Azure Boards provides software development teams with the interactive and customizable tool. It provides rich capabilities including Agile, Scrum, Kanban process, Calendar \views, configurable dashboard, and integrated reporting. Track User Stories, Bugs, features, Epic Board hubs to view work items as cards and perform quick status updates through drag and drop. The feature is like sticky notes on a physical white board

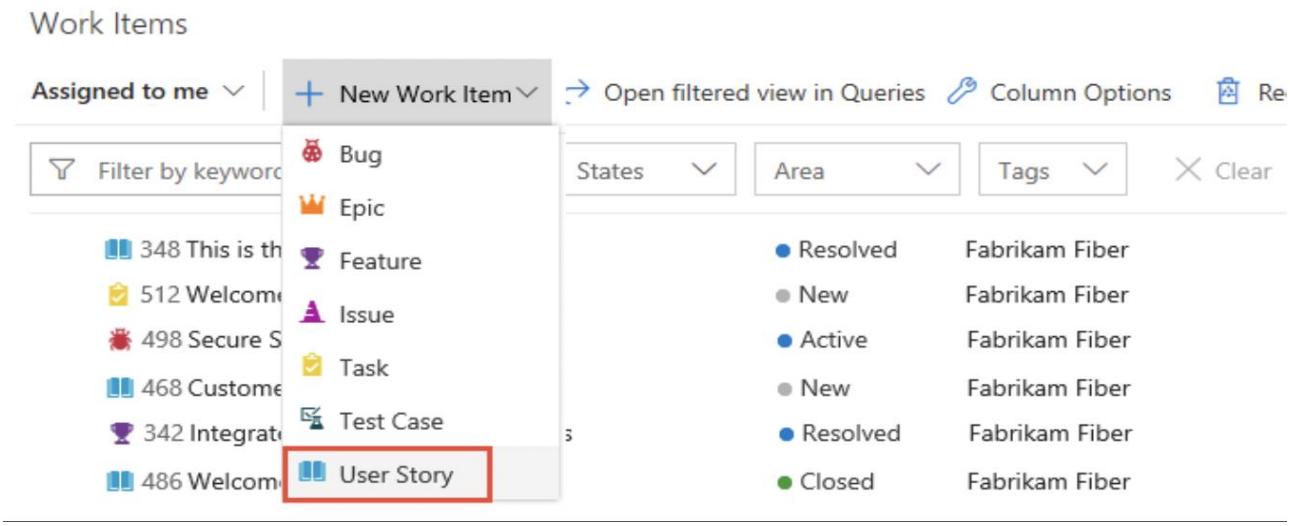

Fig. 0.10. Work Items

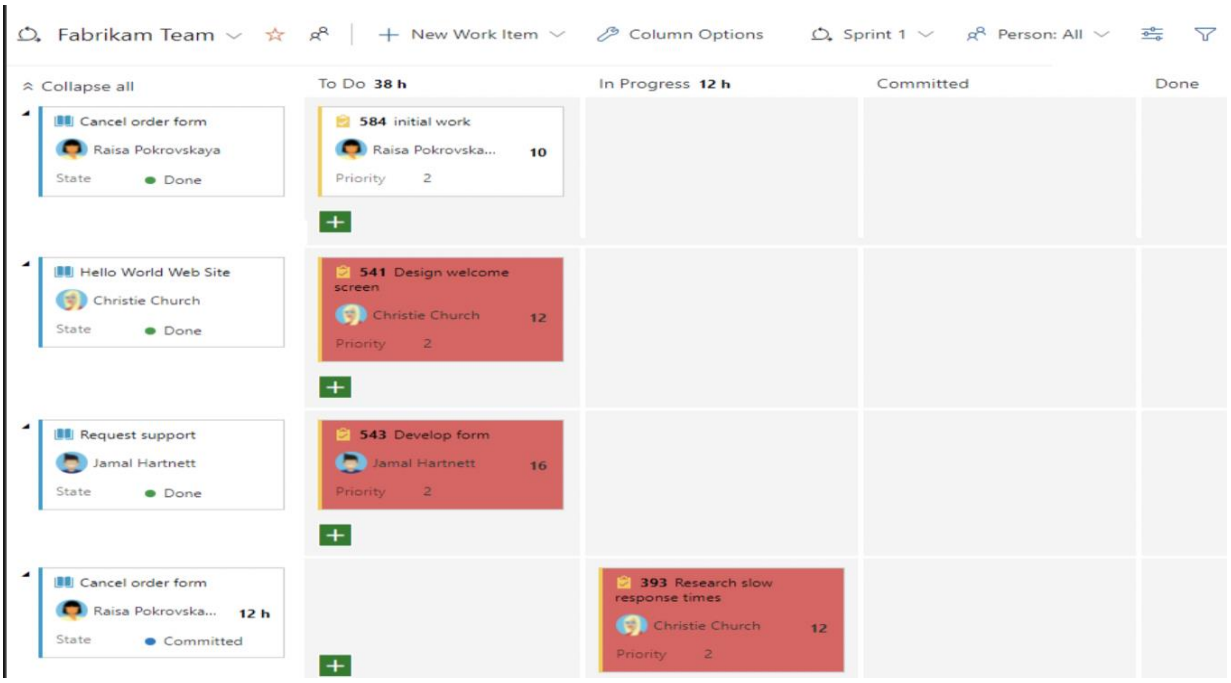

Fig. 0.11. Azure Boards

E. Azure Pipelines

Azure Pipelines is a development environment that enables you to build and manage your production pipelines, with support for on-premises VMs. You can create Pipes that orchestrate data processing across compute clusters and storage systems as well as Azure and OneDrive accounts. When you have complex jobs, it is often better to build the job from code than from the UI. With Azure Pipelines, you can code your workflows using several languages and run them on Windows or Linux environment. Azure Pipelines also provides a wide range of APIs for various services in order to integrate any business logic in your pipelines with other services as shown in this diagram: First, we need to select the version control and in this paper I am selecting TFVC. And select the template. Here we are working on .Net application

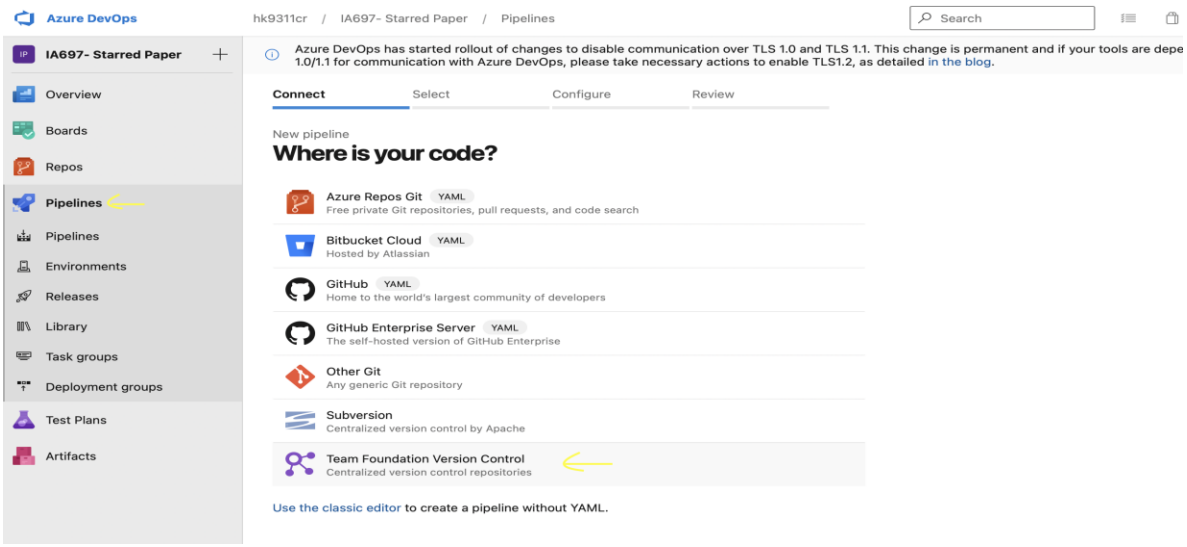

Fig. 0.12 .TFVC .Net Application

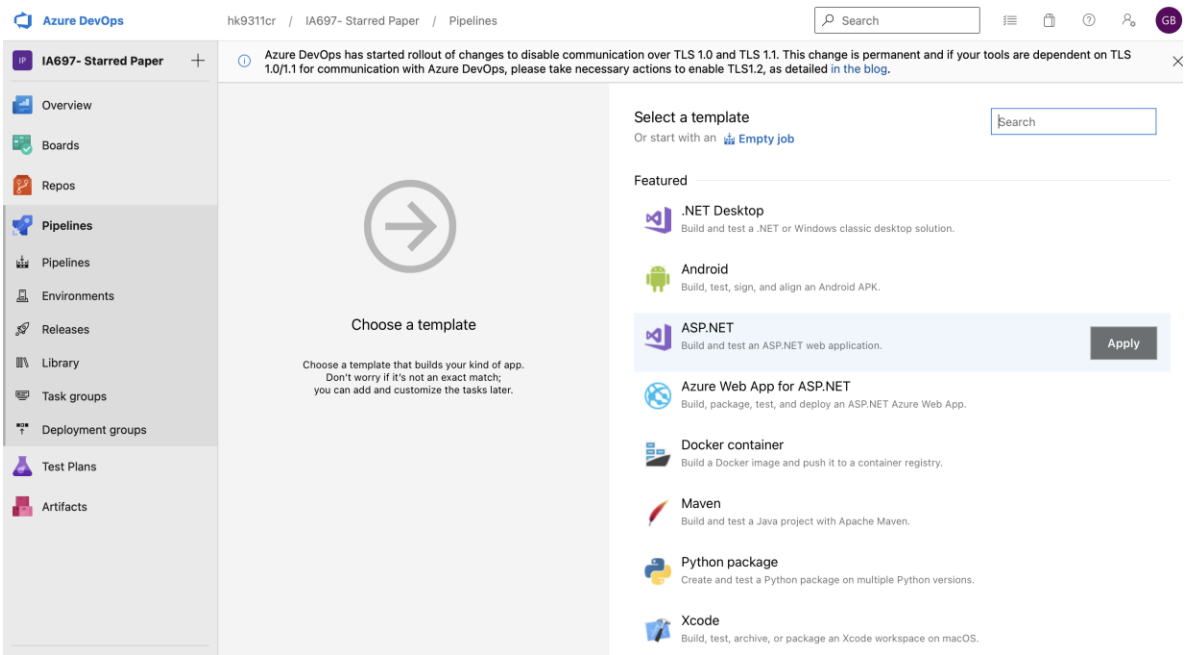

Fig. 0.13. Selecting Template

After selecting .Net Application template below is the pipelines with tasks.

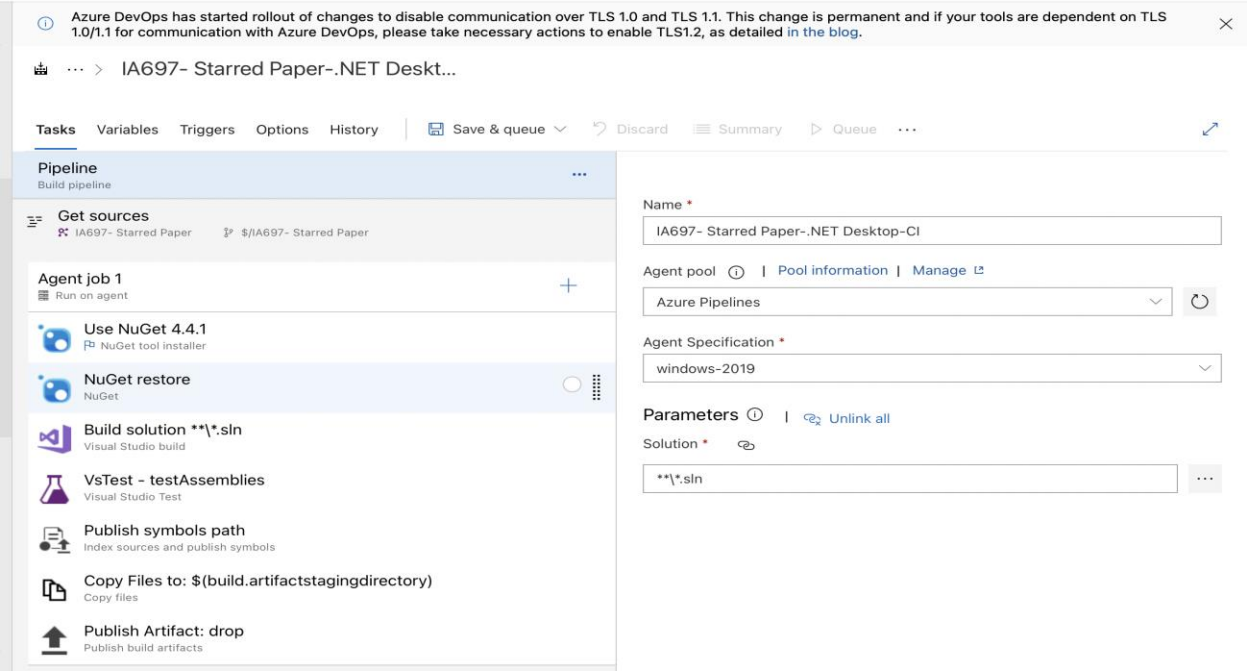

Fig. 0.14. Azure Pipelines with Tasks

There are multiple tasks seen here, like the NuGet package restore that consumes and build NuGet packages in the solutions. Build solution file is the key file which is associated with the project and after building the solution file all the binaries also called as build artifacts are generated.

We can include testing steps and for Continuous deployment we can give the Server path or hosted application path.

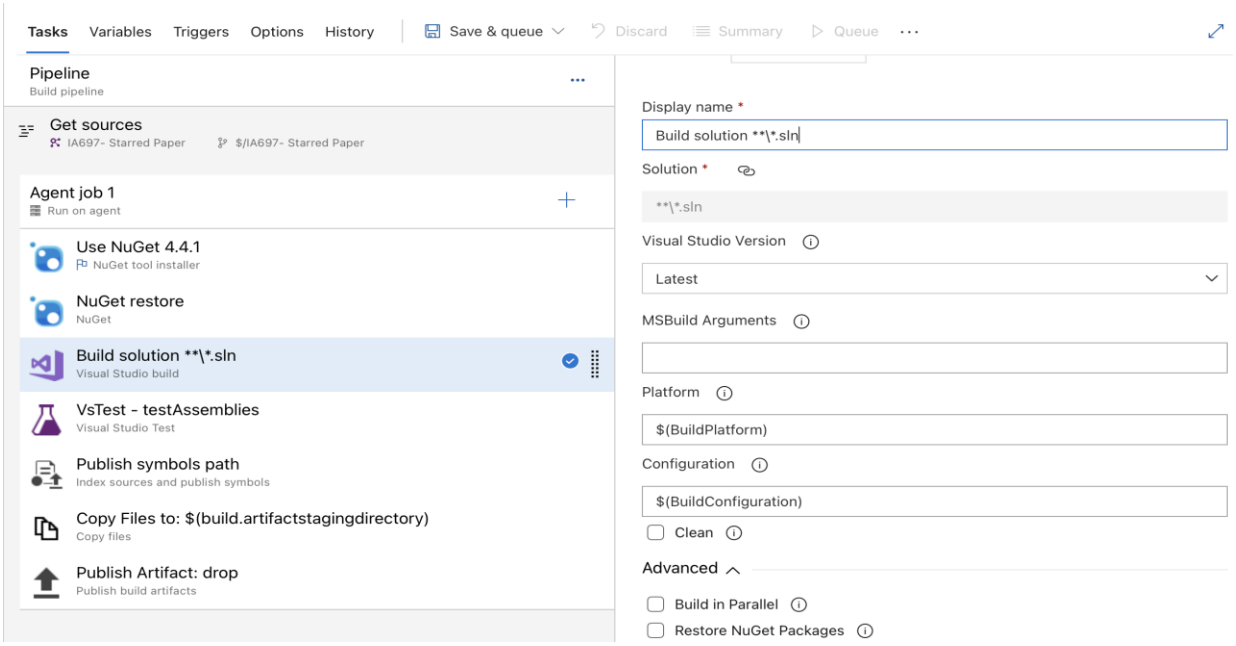

Fig. 0.15. Server Path

We can pass MS Build arguments and select the solution file.

DevOps Security-Fortify SCA

Azure Fortify SCA integration

Fortify SCA helps to scan for vulnerabilities. Azure Pipelines has tasks and for the fortify static analyzer task we need to browse market extensions. Below picture shows how to browse market extensions.

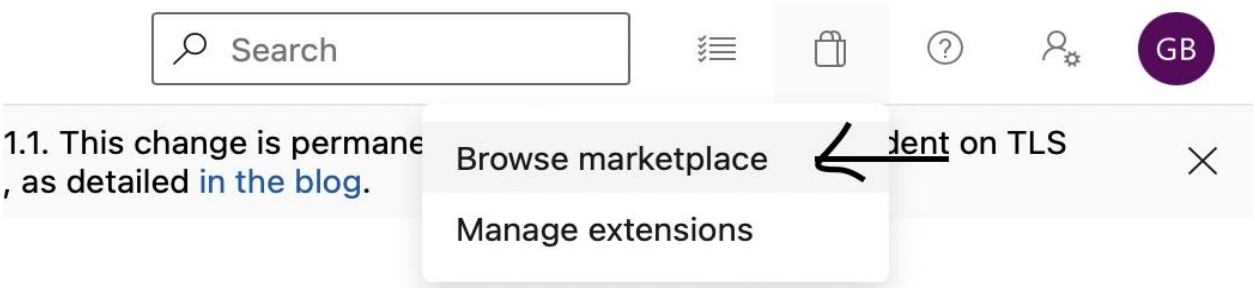

Fig. 0.16. Browsing Market Extensions

In the Market extensions type Fortify and search

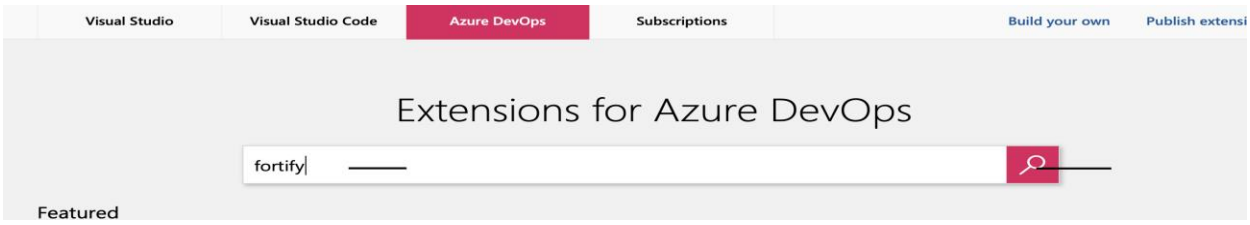

Fig. 0.17. Extensions for Azure DevOps

Once you see the extension for fortify select the one Micro Focus Fortify and install it.

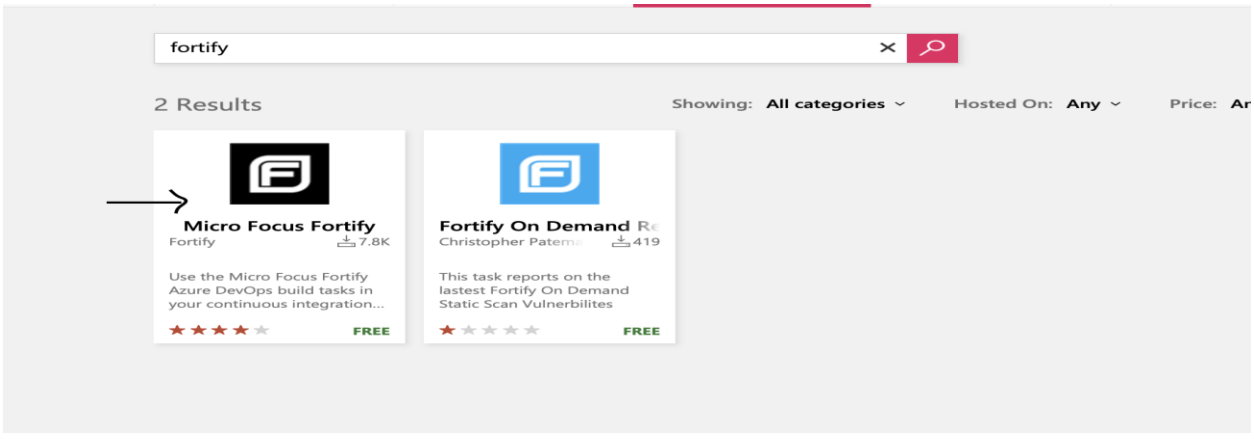

Fig. 0.18. Selecting and Installing Micro Focus Fortify

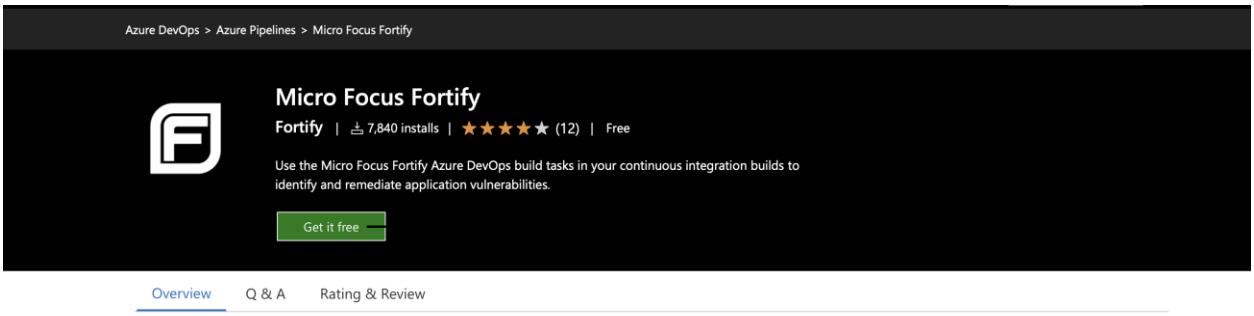

Fig. 0.19. Micro Focus Fortify

Once installed go back to Azure pipelines and add task

Fig. 0.20. Task

And add Fortify task Run Fortify On and pass the parameters for build artifacts and SCA file which has results for vulnerabilities.

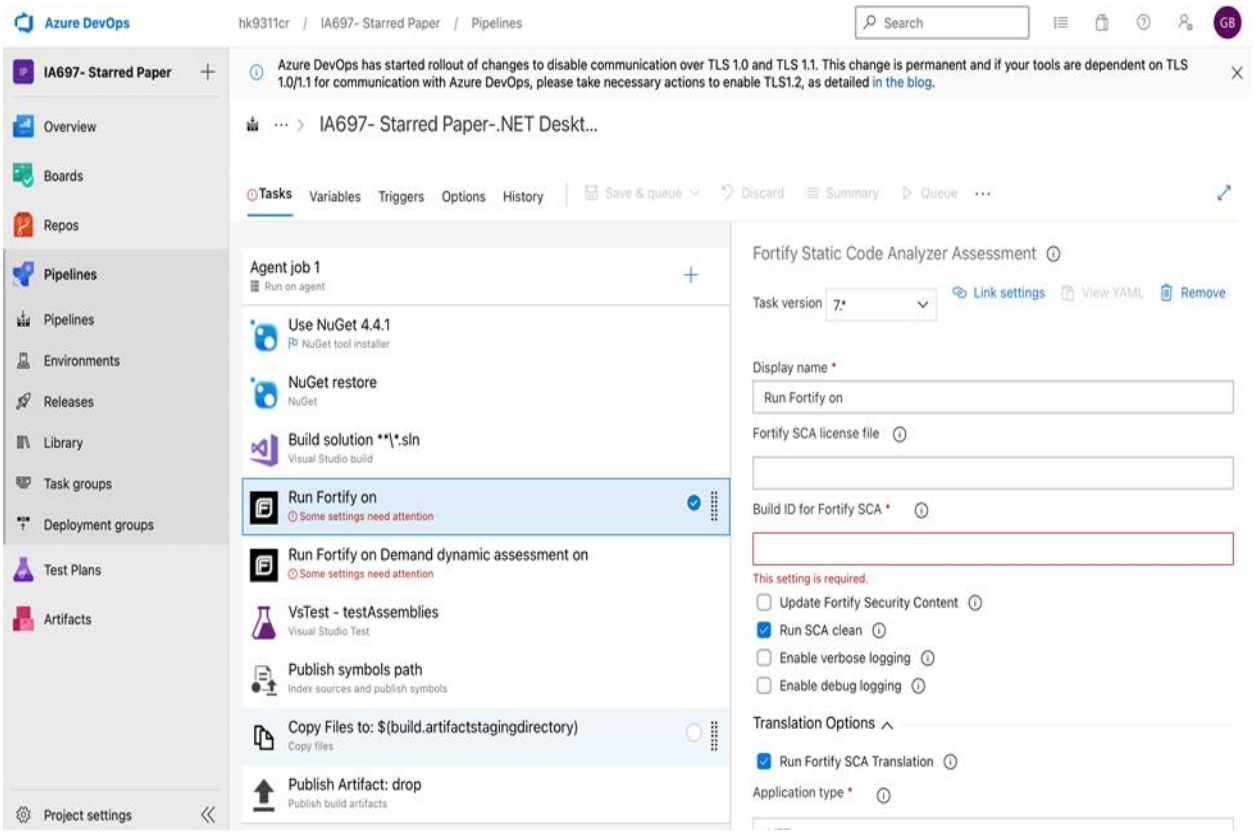

Fig. 0.21. Run Fortify On

CHAPTER IV: RESULTS AND DISCUSSION

Based on the compliance selected there were top ten vulnerabilities that were noticed in the Open Web Application Security Project (OWASP).

A. Sensitive Data Exposure

According to OWASP, sensitive data exposure is one of the ten vulnerabilities identified. In DevSecOps, this is a critical factor. In a DevOps environment, it is crucial to apply security practices throughout the entire process of software development [8]. Data should not be transmitted during certain stages of development or testing. You may want to write an application with many features, but if you do not perform proper testing, that data could be exposed at any stage of the process [40]. Security threats in DevOps include malicious code from human operators, accidental data exposure from developers to clients as well as third parties who may have malicious plans for hackers trying to infiltrate. Their objective is to exploit weaknesses that are present within the system and by exploiting them, they can gain access to sensitive data which could be used for malicious purposes such as identity theft.

DevSecOps" which incorporates security into the DevOps process through automation with high-definition protection from cyberattacks and malicious code. This new way of treating security in DevOps will not only minimize the number of vulnerabilities but also reduce risk of data exposures that other methods create [41]. DevSecOps will allow organizations to manage the entire software process, be it development, testing, or production. By incorporating security in DevOps and using this new method, organizations can minimize the number of vulnerabilities and data exposures while simultaneously increasing the quality of each application release.

B. XML External Entities

XML External entities is another vulnerability in DevSecOps. Just as in the case of SQL Injection vulnerability, for this vulnerability an attacker will inject a malicious XML document with an external entity such as a reference to a resource on the Internet [42]. This can allow for cross-site scripting that allows the attackers to execute commands and potentially gain access to sensitive information, like credentials or other private data.

DevOps is never without risks. But by using DevSecOps tools like OWASP ZAP and Strut safe, developers can eliminate any security vulnerabilities they create while also making their code more secure and reliable [43]. It is important to note that to be effective, an attacker would need to have some control over the entity being referenced. For example, this could be a server controlled by the attacker, or a misconfigured third-party server. To prevent this vulnerability, it is recommended that developers use W3C XML Schema, instead of an XSD schema [9].

Strict Transport Security (HSTS) is a way for a browser to indicate that it will only communicate with websites in a secure manner [44]. This means the browser will refuse all connections with the less secure variants of the website regardless of whether the Browser supports HSTS.

C. Broken Access Controls

Broken Access Controls in DevOps is a common vulnerability under the DevSecOps topic. In a DevOps environment, access control is often outsourced to third-party applications and people have no insight or control over these external systems [45]. This leaves teams exposed to security and compliance issues as well as data loss due to accidental deletion or overwriting of files in these new systems.

D. Security Misconfiguration

The term "security misconfiguration" is defined as an issue with the application configuration which causes a vulnerability. In the DevOps world, where continuous deployment often occurs, security issues have become more of a concern as there are many points in time where a configuration could be altered and cause an exploit to occur without notice.

Common and costly issues:

Cloud provider misconfiguration - Lost or compromised access keys and credentials can jeopardize your entire infrastructure. This is a result of technical difficulties in managing cloud configurations [46].

Web server misconfiguration - Web server misconfigurations are one of most common issues in the DevOps world because these servers often interact heavily with the application code. The web servers exposed configuration parameters that allow attackers to obtain sensitive information like usernames and passwords, or even use certain functionality of the web application (e.g., email functionality) to attack users [10]. In addition, some applications use hardcoded (static) credentials, which may be inadvertently committed into the code.

E. Cross-site Scripting

Cross-site scripting is another vulnerability identified by OWASP in DevOps. The vulnerability is typically exploited by sending an HTTP request to a vulnerable site that includes arbitrary JavaScript in the Referrer or the Cookie header. If the script is successfully allowed, then the attacker can access data on the server that they would not normally have access to [47]. It can be identified by the browser giving an error message or warning when viewing a particular page while the script is running. Voluntary Cross-site scripting is any XSS attack that is carried out by

a user, willingly. They may be carried out through a malicious payload they have inserted into a form or URL, or they could happen by mistake. The user may not be aware of the consequences of their actions, but they do know they're executing some code on the site.

F. Insecure Deserialization

Insecure Deserialization is a vulnerability that is addressed by DevSecOps. It is due to a lack of security analysis in the development stages. This vulnerability allows attackers to perform unauthorized data access by presenting themselves as a serialization mechanism (for example, XML or JSON) and manipulating/modifying the data in transit or at rest [48]. A deep analysis of typical application architectures reveals several weaknesses that malicious actors can exploit to achieve their goals. For example, in a typical enterprise application architecture using a microservice approach and cloud-based infrastructure, there are different layers that are each responsible for a specific task and communicate with each other in a protocol specific way (typically, using socket connections). As an example, one of these layers could be responsible for user credentials management and the other one for providing some kind of "API" service to external systems [11]. The problem is that by trusting the services from these two layers implicitly (i.e., assuming they don't have security bugs), the application cannot protect itself against malicious actors who attack them.

G. Using Components with Known Vulnerabilities

Using components with known vulnerabilities can be a security threat. It's important to keep your systems up to date with vendor patches. As a side note, there might be conflicts between packages that cause problems and make it difficult for you to use them on the same system. It's

important to resolve these conflicts before implementing any changes [49]. Systems like Icinga 2 Log Collector can help you monitor your systems for security breaches and rule out any potential risk factors or vulnerabilities. The logs are stored safely in the database and can't be accessed without someone logging into the database itself.

H. Insufficient Logging and Monitoring

Without proper monitoring and logging, attackers can evade detection. Without the right tools in place, a DevOps team can’t see what’s happening or take action to prevent new flaws. Monitoring tools need to focus on functions, not tasks. Only by understanding what is happening can you ensure that problems don’t occur again [50]. When upgrading your organization's DevOps toolset should be one of the first things you do to take advantage of DevSecOps. Developing a well-coordinated, successful security program is crucial to getting the most out of new technology. Without proper monitoring and logging, attackers can evade detection. This can dramatically impact both the performance and security of your organization.

I. Broken Authentication

Here, an attacker can bypass authentication, taking advantage of a vulnerability in the application. The most common form of this vulnerability is where local users are trusted with more privileges than they should be allowed. This error has been found in many web applications that have not implemented modern security controls, such as strong passwords and proper authorization. Once an attacker has access to the system, they can gain access to data or perform other malicious activity.

J. Injection

Here, the HTTP parameters are sent, instead of the values from the URL. Hackers can use this to get their data back into your database through the URL parameter. This is quite useful for exploiting SQL injection vulnerabilities and other vulnerabilities that rely on parameters in URLs.

CHAPTER V: DISCUSSION

In comparison with traditional IT, where security was often an afterthought or an add-on, DevSecOps is focused on integrating security into the development phase, rather than treating it as an afterthought. Moreover, it requires security defenders to be extremely proactive instead of reactive. This approach requires DevOps teams to spend more time on security, rather than simply ignoring it. More importantly, security staff will need to be constantly plugged into the development process and not just waiting for an issue to pop up [12]. This makes it easier for them to discover vulnerabilities and work with those behind them to fix it before an attacker does.

A. The Challenges of Implementing Devsecops

Current solutions to these challenges exist but they are often inflexible and require heavy investment. For instance, static code analysis tools can perform static code analysis (i.e., static code analysis is not an entirely automated process - humans still need to verify that the results of a static code analysis match their expectations) [51]. This can take a long time, especially for small applications. Also, these tools often integrate with build or continuous integration systems, and do not always work as expected with container orchestration systems like Docker or Kubernetes.

The NCC Group proposed a solution called SecRules. SecRules uses machine learning and automation to create secure code and to automatically build Docker images that can be used in production environments. SecRules works with tools like static analysis, container orchestration tools (like Docker), continuous integration and build systems, and can integrate with Jenkins, GitLab CI, TeamCity, TravisCI or any other CI or CD system [13].

SecRules is available as a service on Azure as well as a hosted self-hosted solution. By using the hosted option, developers can simply use an API to integrate secure coding into their development lifecycle. This API is accessible through any code editor, IDE, or CI/CD system.

The SecRules API allows developers to integrate SecRules into their development process with a single line of code and offers users the ability to automatically create secure Docker images. SecRules is free for open-source software and the first 1000 builds per month are free for private projects and public (on Azure) projects [52]. It is an agentless, continuous integration and continuous deployment solution.

The technical foundation of SecRules is a combination of machine learning and software analysis. During the development phase, SecRules assists developers to write secure code by training with a large set of secure source code and rule violations (bugs) found in open-source software. Similarly, SecRules can use analysis of existing code to prevent vulnerabilities from entering a code base in the first place [21]. The way that this is done is through static analysis and machine learning: static analysis reports the likely errors or defects in a set of source code based on other projects with similar characteristics; machine learning matches these defects against malicious behavior observed in collected threat data.

Another solution for implementing DevSecOps that has been proposed, apart from SecRules, is the use of a Security Development Lifecycle (SDL). Here, a computer program that is under development is analyzed for security issues [14]. This analysis can take place during every phase of development, using multiple tools. These tools should be custom-made for each phase, as this allows them to best serve their purpose. At the end of each phase there should be some

verification or review that is done by a human, to see if there are any errors in the execution and how it affects the product [53].

The SDL Model is not an actual model, and it is more of a description of the ideal development process. It is suggested that a software developer starts with a problem to solve and specifies the solution in terms of functionality, architecture, etc [20]. This solution can then be evaluated and integrated into the SDL process. Some aspects of the product, such as security, can be added at any time during the development stage but are recommended to be added after each phase has been completed [15].

A traditional SaaS-based DevOps strategy focused on software delivery automation in provisioning new code environments. Hence, DevOps practices and tools are not necessarily always necessary to implement Security Development Lifecycle (SDL).

CHAPTER VI: CONCLUSION

In conclusion, it has been discovered that security solutions are critical in the field of IT, and in this case, DevSecOps. This study has presented the reasons for this. Additionally, the study has proposed a few possible solutions for the challenges faced in implementing DevSecOps systems. One of them is SecRules which uses machine learning and automation to create secure code and automatically build Docker images. Another possible solution is the use of a Security Development Lifecycle (SDL), which may be more applicable for organizations that have well-established SDL processes. This research has also provided insights into the importance of security in software development and the field of IT in general.

Pros: This research has been eye opening on matters to do with web security. Cyber-attacks have been on the rise in recent years and this research provides an opportunity to address these issues systematically [18]. An analysis of the Azure tools has also been instrumental in providing a detailed, step-by step analysis, providing evidence of the working mechanism of the various Azure tools, integrated with the Fortify SCA.

Cons: The research was limited to a few DevOps tools and techniques. This provides a narrow scope through which the research can be handled. It is important for research to cover a wider scope when it comes to active issues in the current world.

A. Future Research

This research has provided opportunity for future research in this field by identifying how to improve existing tools for DevSecOps. Future improvements should consider the increase in digital information and data creation which would require a higher level of security than previously

thought necessary by analyzing possible changes to software development lifecycle management practices such as agile methodologies or product development process models [19].

Future research might improve the security practices of DevSecOps by finding ways to protect IP, data, and digital goods [16]. This research would look at techniques to detect intrusions and cybercrimes within organizations that implement DevSecOps. One possible way to increase the effectiveness of DevSecOps is through innovation in technology and emerging trends in information security [17]. Future research could look at how technology has been applied by DevSecOps practitioners by using cases such as the barcode scan ability (Bartender) and WALA, a Human-driven Aid system for non-expert users.

REFERENCES

[1] L. Leonardo, C. Rocha, F. Kon, D. Milojicic, and P. Meirelles. "A survey of DevOps concepts and challenges," *ACM Computing Surveys (CSUR)* vol. 52, no. 6, pp. 1-35, 2019.

[2] B. Len, I. Weber, and L. Zhu. *DevOps: A software architect's perspective*. Addison-Wesley Professional, 2015.

[3] A. Mann, A. Brown, M. Stahnke, and N. Kersten. *"*State of DevOps report 2018,*"* in *Tech. rep*., 2018.

[4] K. Gene. "A DevOps roadmap for security." 3rd ed. Tech. rep., Signal Sciences, 2020.

[5] L. Riungu-Kalliosaari, S. Mäkinen, L.E. Lwakatare, J. Tiihonen, and T. Männistö, "DevOps adoption benefits and challenges in practice: A case study," in *International Conference on Product-Focused Software Process Improvement*, Springer, 2016, pp. 590–597.

[6] H. Myrbakken, and R. Colomo-Palacios. "DevSecOps: A multivocal literature review,*"* in *International Conference on Software Process Improvement and Capability Determination*, Springer, 2017, pp. 17–29.

[7] S. Prince. "The product managers' guide to continuous delivery and DevOps." Mind the product. Englewood Cliffs, NJ, 1970.

[8] M. Shahin, M.A. Ali Babar, and L. Zhu. "Continuous integration, delivery and deployment: A systematic review on approaches, tools, challenges and practices," *IEEE Access*, vol. 84, pp. 1234-1276, 2017.

[9] M. Shahin, M. Zahedi, M.A. Babar, and L. Zhu. "An empirical study of architecting for continuous delivery and deployment," *Empir. Softw. Eng*., vol. 24 (3), pp. 1061–1108, 2019.

[10] I. Fléchais, "Designing secure and usable systems," PhD dissertation, University College London, UK, 2005.

[11] R. Kumar, and R. Gomar, "Modeling continuous security: A conceptual model for automated DevSecOps using open-source software over cloud (ADOC)," *Computers & Security*, vol. 97, p. 101967, 2020.

[12] R. Mao et al., "Preliminary findings about devsecops from grey literature," in *2020 IEEE 20th International Conference on Software Quality, Reliability and Security (QRS)*, 2020, pp. 450-457. IEEE.

[13] M.A. Howard. "A process for performing security code reviews," *IEEE Security & Privacy*, vol. 4, no.4, pp. 74–79, 2006.

[14] J. Peterson. "Dynamic application security testing: DAST basics." Whitesourcesoftware.Com. https://resources.whitesourcesoftware.com/blog-whitesource/dast- dynamic- application- security- testing. (Accessed July 27th, 2022).

[15] S. Alromaihi, W. Elmedany, and C. Balakrishna. "Cyber security challenges of deploying IoT in smart cities for healthcare applications," in *2018 6th International Conference on Future Internet of Things and Cloud Workshops (FiCloudW)*, August 2018, pp. 140-145, DOI: 10.1109/W-FiCloud.2018.00028.

[16] J. Pan, and Z. Yang, "Cybersecurity challenges and opportunities in the new edge computing+ IoT world," in *Proceedings of the 2018 ACM International Workshop on Security in Software Defined Networks & Network Function Virtualization*, 2018, pp. 29-32.

[17] J. Singh, "Cyber-attacks in cloud computing: A case study," *International Journal of Electronics and Information Engineering*, vol. 1, no. 2, pp. 78-87. 2014.

[18] T. Mariarosaria, T. McCutcheon, and L. Floridi. "Trusting artificial intelligence in cybersecurity is a double-edged sword," *Nature Machine Intelligence*, vol. 1, no. 12, pp. 557-560, 2019.

[19] A. Hind, M. Alshurideh, B. Al Kurdi, and S. A. Salloum. "The impact of ethical leadership on employees performance: A systematic review," in *International Conference on Advanced Intelligent Systems and Informatics*, Springer, Cham, 2020, pp. 417-426.

[20] M. T. Javier, C. I. Comesaña, and P. J. García-Nieto. "Machine learning techniques applied to cybersecurity," *International Journal of Machine Learning and Cybernetics*, vol. 10, no. 10, pp. 2823-2836, 2019.

[21] C. Prathamesh, and N. T. Rao. "Teaching cyber security course in the classrooms of NMIMS University," *International Journal of Modern Education and Computer Science (IJMECS)* vol. 13, no. 4, pp 1-15, 2021.

[22] B. Nodeland, S. Belshaw, and M. Saber. "Teaching cybersecurity to criminal justice majors," *Journal of Criminal Justice Education*, vol. 30, no. 1, pp 71-90, 2019.

[23] T. Chee-Wooi, J. Hong, and C. Liu. "Anomaly detection for cybersecurity of the substations," *IEEE Transactions on Smart Grid*, vol. 2, no. 4, pp. 865-873, 2011.

[24] L. Zhiyi, M. Shahidehpour, and F. Aminifar. "Cybersecurity in distributed power systems," *Proceedings of the IEEE*, vol. 105, no. 7, pp. 1367-1388, 2017.

[25] T. Benjamin, and R. Karri. "Challenges and new directions for ai and hardware security," in *2020 IEEE 63rd International Midwest Symposium on Circuits and Systems (MWSCAS)*, Springfield, MA, USA, August 2020, pp. 277-280.

[26] L. Fang, and J. Wang. "A user-centric machine learning for learning support system with adequate cyber security," *Wireless Personal Communications*, pp. 1-22, 2021.

[27] A. Norita, P. Laplante, J. Defranco, and M. H. Kassab. "A cybersecurity educated community," *IEEE Transactions on Emerging Topics in Computing*, 2021.

[28] T. Chee-Wooi, G. Manimaran, and C.-C. Liu. "Cybersecurity for critical infrastructures: Attack and defense modeling," *IEEE Transactions on Systems, Man, and Cybernetics-Part A: Systems and Humans*, vol. 40, no. 4, pp. 853-865, 2010.

[29] E. Zeinab, K. Sadatsharan, N. Sugunaraj, D. F. Selvaraj, S. Plathottam, and P. Ranganathan. "Cybersecurity attacks in vehicular sensors," *IEEE Sensors Journal*, vol. 20, no. 22, pp. 13752-13767, 2020.

[30] G. Justin Scott, J. K. McDonald, J. Balzotti, D. L. Hansen, D. M. Winters, and E. Bonsignore. "Increasing cybersecurity career interest through playable case studies," *TechTrends*, vol. 65, no. 4, pp. 496-510, 2021.

[31] S. Iqbal H., A. S. M. Kayes, S. Badsha, H. Alqahtani, P. Watters, and A. Ng. "Cybersecurity data science: An overview from machine learning perspective," *Journal of Big Data*, vol. 7, no. 1, pp. 1-29, 2020.

[32] L. Yang, and L. Xu. "Internet of Things (IoT) cybersecurity research: A review of current research topics," *IEEE Internet of Things Journal*, vol. 6, no. 2, pp. 2103-2115, 2018.

[33] D. Jessica, J. E. Pérez, M. A. Lopez-Peña, G. A. Mena, and A. Yagüe. "Self-service cybersecurity monitoring as enabler for devsecops," *IEEE Access*, vol. 7, pp. 100283-100295, 2019.

[34] W. Carol, T. Chick, A. Reffett, S. Pavetti, R. Laughlin, B. Frye, and M. Bandor. “DevSecOps pipeline for complex software intensive systems: Addressing the cybersecurity challenges.” *The Journal on Systemics, Cybernetics and Informatics: JSCI*, vol. 18, no. 5, pp. 31-36, 2020.

[35] O. Muñoz, and J. Mejja. “Responsive infrastructure with cybersecurity for automated high availability DevSecOps processes,” in *2019 8th International Conference On Software Process Improvement (CIMPS)*, Leon, Mexico, October 2019, pp. 1-9.

[36] D. Ashenden, and G. Ollis. “Putting the sec in devsecops: Using social practice theory to improve secure software development,” in *New Security Paradigms Workshop*, Basel, Switzerland, July 2019, pp. 34-44. [Online]. Accessed December, 2020).

[37] W. Carol. “CERT GBSD Projects: Designed in Assurance.” Carnegie-Mellon University, May 2019.

[38] R.N. Roshan, M. Zahedi, M. A. Babar, and H. Shen. "Challenges and solutions when adopting DevSecOps: A systematic review," *Information and Software Technology*, no. 141, pp. 106700, 2022.

[39] B. Ahmed, A. Abdelaziz, A. Sayed, L. Elfangary, and H. Fahmy. "Monitoring real time security attacks for IoT systems using DevSecOps: A systematic literature review," *Information*, vol. 12, pp. 154-189, 2021.

[40] E. Luiijf, K. Besseling, and P. de Graaf. “Nineteen national cyber security strategies.” *International Journal of Critical Infrastructure Protection*, vol. 9, no. 1-2, pp. 3-31, 2013.

[41] C. Colliander. "Challenges of DevSecOps," thesis, University Of Helsinki, Helsinki, Finland, 2022.

[42] R. Roshan, M. Zahedi, M. Ali Babar, and H. Shen. "Challenges and solutions when adopting DevSecOps: A systematic review." *Information and Software Technology*, vol. 32, pp. 141-156, 2022.

[43] H. Mitchell. "The Influence of Cybersecurity on Modern Society," *Foundations of Computation and Intelligence*, no. 5882, June 2021.

[44] K. Geers. *Strategic cyber security*. NATO CCDCOE Publications, 2011.

[45] N. Chaillan, & H. Yasar. "Waterfall to DevSecOps in DoD." Carnegie Mellon University Software Engineering Institute Air Force, 2019.

[46] M.D. Cavelty, "Cyber-security*",* in *The Routledge Handbook of New Security Studies,* J. P. Burgess, Ed., Oxfordshire, England, UK: Routledge, 2010, ch. 23, pp. 154-162,

[47] T.A. Chick, A. Reffett, N. Shevchenko, & J. Yankel. 2021. "Modeling DevsecOps to reduce the time-to-deploy and increase resiliency." Carnegie-Mellon University, Pittsburgh, PA, report # AD1121063, 2021.

[48] T.A. Chick. "MBSE for DevSecOps CI/CD Pipeline." Carnegie-Mellon University, Pittsburgh, PA, USA, 2021.

[49] R. Kumar and R. Goyal. "Modeling continuous security: A conceptual model for automated DevSecOps using open-source software over cloud (ADOC)," *Computers & Security*, vol. 97, no. 101967, 2020.

[50] T.E. Gasiba, I. Andrei-Cristian, U. Lechner, & M. Pinto-Albuquerque, *"*Raising Security Awareness of Cloud Deployments using Infrastructure as Code through CyberSecurity

Challenges,*" ARES 21: Proceedings of the 16th International Conference on Availability, Reliability and Security*, no. 63, pp. 1-8, 2021.

[51] H. Myrbakken, & R. Colomo-Palacios. *"*DevSecOps: A multivocal literature review,*" International Conference on Software Process Improvement and Capability Determination*, September, 2017, pp. 17-29.

[52] C. Dongliang, P. Wawrzynski, and L. Zhihan. *"*Cyber security in smart cities: A review of deep learning-based applications and case studies.*" Sustainable Cities and Society*, vol. 66, pp. 534-657, 2021.

[53] A. Osama, N. Moustafa, and B. Turnbull. *"*A review of intrusion detection and blockchain applications in the cloud: Approaches, challenges and solutions.*" IEEE Access*, vol. 8, pp 234-345, 2020.