

DevOps and Project Management Trainee

Bertrand Gogniat



Author	
Bertrand Gogniat	
Degree programme	
Business Information Technology	
Report/thesis title	Number of pages
Diary Thesis - DevOps and Project Management Trainee	93
<p>The report reflects 10 weeks of work during my trainee at <i>Microsym</i> company on Azure DevOps and project management. The observation period was from 08.02.2021 to 16.04.2021. My daily work covers various topics such as code automation, continuous delivery, configuration management, scripting and project management. <i>Microsym</i> is planning to use Azure DevOps for their projects, which will allow them to integrate development and project management in one tool. The overall objective is to improve development speed, reduce time to market, enhance collaboration between the teams and increase customer satisfaction.</p> <p>The aim of the trainee is to test and document Azure DevOps functionalities and other kinds of tools in order to ease the migration in the company. Such an initiative can create important cultural changes in companies by modifying the way developers, testers and system administration work together during the development and delivery processes. In this thesis you will experiment the process and the challenges related.</p> <p>At the beginning of my trainee, I had a simple idea of what DevOps concepts were and I had no practical experience. Through these 10 weeks, I come across various topics and tools that allow me to build a stronger comprehension and skills. The experience described in this thesis can provide an introduction to the world of DevOps with concrete use cases and exercises for anyone who is interested in learning more about the topic.</p> <p>The thesis was written in 12 weeks in the diary format, it allows to have an inside on my daily tasks and way of thinking when handling them. The weekly analysis provides theoretical and technical explanations of concepts met during the week, along with a review and justification of my choices made considered on different points of view.</p>	
Keywords	
Azure DevOps, Python, pipelines, configuration management, project management, integration, data reporting	

Table of contents

Acronyms	1
Figures	2
1 Introduction	3
2 Framework.....	5
2.1 Analysis of my current work	5
2.1.1 Description.....	5
2.1.2 Evaluation	7
2.1.3 Development.....	7
2.2 Interest groups at work.....	9
2.3 Interaction skills at work.....	10
3 Diary entries.....	11
3.1 Week 1 (08.02.2021 – 12.02.2021)	11
3.2 Week 2 (15.02.2021 – 19.02.2021)	19
3.3 Week 3 (22.02.2021 – 26.02.2021)	27
3.4 Week 4 (01.03.2021 – 05.03.2021)	35
3.5 Week 5 (08.03.2020 – 12.03.2020)	41
3.6 Week 6 (15.03.2021 – 19.03.2021)	49
3.7 Week 7 (23.03.2021 – 26.03.2021)	56
3.8 Week 8 (29.03.2021 – 02.04.2021)	62
3.9 Week 9 (05.04.2020 – 09.04.2021)	69
3.10 Week 10 (12.04.2020 – 16.04.2021)	76
4 Discussion and Conclusion	83
References.....	87

Acronyms

YAML	Yet Another Markup Language (YAML Ain't Markup Language)
UART	Universal Asynchronous Receiver/Transmitter
CMD	Command Prompt (Microsoft Windows)
CLI	Command Line Interface
REST	Representational State Transfer
API	Application Programming Interface
XSS	Cross-Site Scripting
CI	Continuous Integration
CD	Continuous Delivery
PAT	Personal Access Token
WIQL	Work Item Query Language
ID	Identity Document
SMTP	Simple Mail Transfer Protocol
ARM	Advanced RISC Machine
RISC	Reduced Instruction Set Computing
AMD	Advanced Micro Devices
TTY	TeleTYpewriter
PTS	Pseudo-Terminal Slave
STDIN	Standard Input
STDOUT	Standard Output
GNU GPL	General Public License
OLAP	Online Analytical Processing
OLTP	Online Transactional Processing
PBKDF	Password-Based Key Derivation Function 2
AES	Advanced Encryption Standard
SCM	Source Version Control
XCFE	XForms Common Environment
KDE	K Desktop Environment
ALM	Application Lifecycle Management

Figures

Figure 1 : YAML file using templates (Microsoft.com, 2020d)	16
Figure 2 : Query Editor from Azure Board	20
Figure 3 : Result of the Python script	21
Figure 4 : Mail sent by Python script	22
Figure 5 : Example of WIQL syntax	26
Figure 6 : YAML Pipeline to flash the sensor	27
Figure 7 : Logs of the YAML Pipeline	28
Figure 8 : Resource tag in YAML pipeline	29
Figure 9 : Message published by the Python script in Teams' chat	32
Figure 10 : Request body parameters	36
Figure 11 : Overview of the work done by the script	37
Figure 12 : POST request with JSON body in Postman	40
Figure 13 : Location of water consumption (mock data)	43
Figure 14 : Ansible schema (Shimmercat.com, 2020)	44
Figure 15 : YAML file encrypted with Ansible Vault	51
Figure 16 : Repository "dockerimages" cloned from Bitbucket	52
Figure 17 : Azure DevOps, Lead Time vs. Cycle Time (Microsoft.com, 2020e)	54
Figure 18 : Ansible Tower template for MySQL	58
Figure 19 : Tasks to uninstall MariaDB and Galera packages	59
Figure 20 : WIQL request that retrieves work items assigned to a user	62
Figure 21 : Number of cases per size and state	64
Figure 22 : Progress roll up column example	67
Figure 23 : Tasks.json after modifications	70
Figure 24 : Jira logs, webhook for Azure Function	71
Figure 25 : Jira logs, webhook for Logic App	71
Figure 26 : Azure Functions up and running	72
Figure 27 : Body of the SB POST request	76
Figure 28 : Jira and Silverbucket data mapping fields	77
Figure 29 : Stashed changes schema	82

1 Introduction

The diary journal starts on 08.02.2021 and is going to be filled for 10 weeks until 16.04.2021. The following weeks are going to be dedicated to completing the conclusion and the analysis of all the work done during the period. The final submission is planned on the 23.04.2021.

The purpose of my work at *Microsym*, when I'm writing these lines, is to do research on topics related to Azure DevOps to gain theoretical and practical experience by performing different testing tasks or small projects using the tools described. This is to enable an easier migration to Azure DevOps and to make it the main system of the company for project management and development. The tasks can be quite varied as it is an environment that integrates many complex tools. All our research and code produced are added to a repository in Azure dedicated to the migration and will contain all our work allowing us to share the acquired knowledge with the whole company. The objective of the Azure DevOps migration is to get an embedded development tool that will bring developers, project managers and contributors altogether to complete software development and management.

Most of the work is going to be related to DevOps concepts and implementation, which is described as a set of practices that works to automate and integrate the processes between software development (Dev) and IT operation (Ops). It allows the team to build, test, and release software faster and more reliably. Another concept that I'll face is CI / CD approach, it increases delivery frequency of applications by introducing automation into the development stages. The main concepts related to the CI / CD approach are continuous integration, continuous distribution and continuous deployment. It ensures continuous automation and monitoring throughout the application lifecycle, from the integration and testing phase through to distribution and deployment. We can refer these practices as CI / CD pipeline and are based on agile collaboration between development and operations teams.

My main source of information is come from the Internet and Haaga-Helia online library, that allows me to get more bibliography and references to come along in my analysis. It will help me to make research more precise and getting proper idea of the subject. Due to the pandemic situation, I don't have much contact with my colleagues, but I still get support from my supervisor through Microsoft Teams.

The skills needed for my trainee are general programming skills, project management knowledge such as Agile and Scrum methodologies and software development

comprehension. It mostly corresponds to my background study at HE-ARC and Haaga-Helia University. Also, it is interesting to have a good general understanding of IT in order to adapt to the different topics and understand the needs of the tasks that I'll have to work on.

To introduce the company I'm working for, *Microsym* is a global digital services company that helps companies build innovative software products and transformative digital services. For over twenty years, *Microsym* has taken a customer-centric approach to unlock innovation and deliver exceptional products and service experience. Embedded in its operational DNA, *Microsym's* innovation and design thinking process goes beyond a problem-focused approach to deliver optimum design and functionality for organizations and its customers with Agile and DevOps approach.

In general, I have at my disposal a laptop and accessories to complete my work; also, I can get different devices for specific tasks that I'll have to cope with. The environment of work is mainly done remotely due to the pandemic situation. I'm still able to come at the office one or two days during the week, that allow me to have a few interactions with my colleagues.

2 Framework

2.1 Analysis of my current work

2.1.1 Description

My work tasks can be described as the following list:

- Researching and testing Azure DevOps functionalities
- Creating documentation
- Script programming
- Participating in project management
- Presenting project data

Tasks can have different environments and skills required but will be related to Azure DevOps functionalities or other kinds of tools related to DevOps. I have some general skills to assets the needs of the company, but as tasks can be very various, I'll be most of the time looking to learn technology or skill required to achieve the tasks. Which allows me to gain more knowledge and will be helpful for my future.

Researching and Testing Azure DevOps Functionalities is the main objective of my trainee, most of the tasks are related with Azure tools. When I start working on a new functionality, the first step usually begins with research on the topic through Microsoft documentation, tutorials and video related to the topic. Getting multiple sources here makes the research even more relevant and allows to confirm some practice or information that might not be in the official documentation. Having a good understanding is essential to work with the tools to see its potential and which use case it will fit the best. The second step is to test the functionalities directly on Azure DevOps, where we have a sand box project dedicating for the experimentation and testing. The third step is to maintain some documentation about all the configuration and environment needed for the task with explanations, this point is more described in the next one. The functionality comes usually with a task or specific use case for it in the company, even if it is only for testing purpose, I'll have to demonstrate and present my research and experimentation in the weekly meeting we stand with the team.

As I was talking about the documentation we have to provide, creating documentation is also a significant point during the realisation of the tasks. When providing any kind of code, configuration, or testing environment. It is not mandatory, but it is a very good practice to provide proper documentation and explanations on how scripts, pipelines or functionalities works. It helps me when working and allows to test if I really understand the concepts when I give an explanation in the document. In general, I keep the configuration and how to set

up the environment in my personal document. When it is more a collective project or someone else is going to work on it later, we have to realise a README file in the repository to explain the functioning of the script, pipeline or other kind of code produced. It is essential to provide documents like this for the next person working on the project to understand it and allow him to take over the environment quicker. Also, the documentation will help during the migration to spread the information about the finds and best practice when using the tools.

Another type of work I can have is programming script. Scripts are required to automate different repetitive manual processes that will come around in my tasks. The language usually used for this kind of script is Python. I was not familiar at all with it at the beginning of the trainee, so I needed a period of time to adapt to it. Fortunately, Python is a simple language, it was not too difficult to work with it, especially with my knowledge on Java programming. Also, the tasks are adapted to my Python skills and I'll have support if it becomes harder or out of my skills. I try to be as independent as possible, so I usually look for the best practice on Internet or I follow different tutorials with explanation on the topic. That allows me to be more efficient in my work and don't lose time to ask the colleague about issues I face. If I don't find a solution to my problem, I let my supervisor know and he will give me a hand on it. More advanced scripts would require more time to do researching and understanding more complex concepts; here I need the general functioning more and basic skills to achieve the tasks.

Also my work includes to be involved in the project management side of the company, where I'll be participating in some projects, meetings, and processes. That gives me an overview on the work and an opportunity to take part in some administrative tasks related to it. Also, I'm participating in reviewing project management processes and documentation in order to update them. In addition, it is possible that I'll give a hand to my colleagues to help them using Azure DevOps functionalities or tools for project management. I try to help as much as I can to answer their question related to Azure DevOps. Here I have a role of a facilitator for the migration, that aim to save time by spreading relevant information and practices for the project management tools in Azure DevOps.

Aside participating in project management task, I'm also working on providing different analytics in the Azure Dashboard. The aim is to provide relevant information for projects to help its management and track of progress. These visualizations depend most of the time on a query that is required to create previously, in order to retrieve the information to be processed in the widget of the dashboard. The metrics are usually defined earlier in the task from my supervisor. However, I'm still able to propose or test any other solution that I find relevant. In general, when I'm facing any concept or theory that I don't fully understand, I

start by informing myself and the topics with documentation and use cases. Then I can be completely conscious of the need and propose better explanations and justification when I make choice.

2.1.2 Evaluation

Before starting the trainee, I didn't know much about DevOps and related concepts. I didn't have any experience with any of these tools, because it's not my field of study directly. Having the trainee in combination with the thesis, allows me to practice and have a theoretical input with the weekly analysis and search. This trainee is a great opportunity for me to improve my skills in this area and to get a better understanding of project management. It will take some time to adapt depending on the complexity of the tasks, but that is part of the process. As far as the project management aspects are concerned, I have more background on the different methodologies and frameworks which will be useful for understanding the tasks. However, there will also be more improvements to make as this is only my first experience in this field.

I try to be independent on the realization of the tasks, so it forces me to search for information, best practice and overcome issues that I can face. I usually receive a few explanations on the requirements from my supervisor and I go for it. Sometimes I need more precision on the technical background, so I have a better guideline to follow. If I don't have any or enough knowledge on the topic, I read up for more information on different sources, such as tutorials, forums and books that help me to realize the task. I usually try to achieve what is needed, I only go further if I found some interests to make it better or I found an alternative to do. In addition, I keep track of my work in a document containing all technical explanations and tutorials of the tasks I perform. It allows me to redo the task easily later on if needed or even share it to my colleagues when they need to set a specific environment.

2.1.3 Development

Now that I will finish soon my studies, I'll be more confronted with the professional world. I have more theoretical background and not a lot of concrete professional experience in the IT field. This trainee allows me to have a strong introduction to it and that's what I was looking for. In my opinion, I think personal development is important to consider for the future and objectives that come with it. I like to have a clear idea of the future and to know exactly where I go. I'm looking for more practical experience in a company and this trainee allows me to absorb knowledge and gain new skills in the IT field. DevOps is an interesting subject as it becomes more and more needed in the company software, that will be an advantage as a project manager to have knowledge on this topic.

At the beginning of my trainee, I was not very confident with scripting and had only a vague idea of DevOps and its concept, the trainee and the thesis allowed me to even more detailed on theory and practical side. It is probably not the exact domain where I want to work full time in the future, but it is a relevant subject and promising for the future.

In general, I like when the work is well done, so I don't hesitate to spend more time on something that I want in a specific state. I am persistent in finding a solution to a problem and I can spend a lot of time on details. It is related to the fact that I usually tend to overthink ideas or work to do, it can very time consuming sometimes. However, in general it helps me to have a proper idea on the issue and help me to have a strong opinion on how to do it. Sometimes, I don't pay attention to some aspect of the task that I'm working on, due to a lack of knowledge in those specific fields.

However, it is the way to learn, mistakes are part of apprenticeship and if it needs more time, I'll take the time to complete properly it. I also like to learn from my mistakes, even if it can be frustrating sometimes to make some, but it is important to learn lessons in order to improve and not making them again in the future.

For the future I would like to work as a project manager, or another kind of job related that domain. This trainee allows me to have an introduction on the topic and some practice, but there is still more to learn in the future. I want to improve is also communication and leadership, qualities that are very important for working as project manager. I'm not an authoritative person, I prefer proving people that they can trust me by showing and performing skills. In my opinion, it is a way more impressive way to have authority.

As I like to describe IT like a wide space of more or less complex technologies, there is always something new to learn about. Sometimes it can be hard to not be lost, so many options and it is complicated to know where to start. DevOps and project management are only a small field of what IT is, that's why I'm saying that there will be always something to learn, new technologies, practices, and so on. However, in my case I want to be more focus on project management and leadership for the moment.

2.2 Interest groups at work

Stakeholders are divided in two categories, internal or external to the company. Internal regroups employees, managers, and owner of the firm. On the other hand, the external side represents the environment that will impact the company from outside. It regroups customers, government, shareholders, and partners. Employees have a direct impact in the company because they provide a value and earn an income to support themselves and sometimes along with other benefits monetary or not. They may have a health, safety, personal interest to be in the company.

The company's customers are any kind of company or individual that required software development service for any kind of fields. The aim is to provide a service of quality and bring more value with the new product delivered. The government is a major stakeholder that influence the company with laws and taxes collection from the company as well as from all the people it employs. Governments benefit from the overall Gross Domestic Product that companies contribute to.

Microsym have many different partners that allows them to provide the best value for their customers. Strategic partnerships help to leverage expertise, community relationship, resources and can also add support that strengthens walkability projects from the planning phase through the implementation.

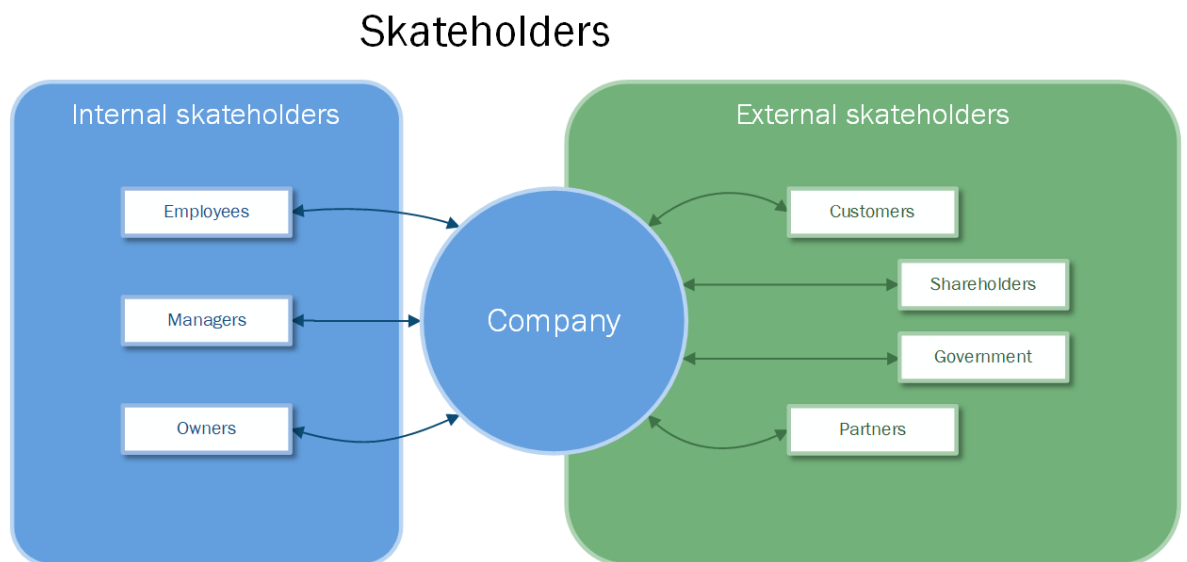


Diagram 1: Skatholders

2.3 Interaction skills at work

Due to the pandemic situation, almost all interactions with my supervisor and colleagues are done remotely on Microsoft Teams. I'm doing my traineeship with two other students and we are working on various tasks mostly related to Azure DevOps. We usually have tasks related to different topics for the Azure DevOps migration. Sometimes, it happens that we collaborate on a same kind of task or we just ask for help if someone has better knowledge on a topic. These collaborations are also constructive for the trainee and future work with teams.

We stand a weekly meeting with my supervisor and colleagues, where we have to share and present the work done during the week. It is interesting to have an insight from the various topics, it also improves the quality of our work with the feedback we got, and it helps for keeping track of progress of the project. Sometimes we have a specific subject presented as a demo by our supervisor that will help us to understand more about a topic and its best practice for an efficient way of working.

These interactions are helpful to keep track of my work and realise what progress is made. Sharing our knowledge and work is also a way to see if we really understand the matter and it allows us to have a good insight on new topics or technologies from the other. It creates a good team spirit even with the complications with the sanitary situation. I'm allowed to go at the office, so I'm doing 1 or 2 days during the weeks. It helps me to have a little break from home. Nonetheless there are only a few workers that come at the office, so I don't have a lot of interaction with my other colleagues or the opportunity to meet them in real life.

As I'm collaborating a bit on the project management side, I may be confronted with some external stakeholders through Teams for the different meetings such as the kick-off, project debrief, steering group or sprint reviews. Later on in my trainee, I hope to be more involved in project management tasks and experience on the field with my colleague and customers.

3 Diary entries

3.1 Week 1 (08.02.2021 – 12.02.2021)

Week's goal is to keep learning about the Azure Pipelines and how to make good YAML files in order to execute the different processes needed for the project. Also, I have to acquire more knowledge about the different functionalities of the Movesense sensor and how it works.

Monday

Last week I've been working on making an Azure Pipeline in order to automate the build of a Movesense application and the flashing process of the sensor. In a nutshell, Movesense is an open sensor development platform created from the sports watch brand Suunto, the sensor is capable of measuring any movement and general information like, temperature, heartbeat and so on.

The building process required a specific environment in order to generate the application files, so the pipeline needs to run a Docker Container. Azure Pipelines are running by agents, an agent is a dedicated machine to execute the different actions specified in the YAML file. For the moment my Azure Pipeline runs with 2 agents, the Hosted Agent (Azure Pipeline agent) and a self-hosted agent (my actual computer) that I configured previously with all the environment required.

Then, I tried to configure the self-hosted agent in order to run the building process, but it was not working because the service didn't have permission to execute "docker.exe". I searched for different ways of fixing this. The best option I found was to set the "Log on" properties of the Azure pipeline service to "Local System Account". After running another time the pipeline, I faced another error. The docker container cannot be started. I don't have any clue why this happens, but as it is not so important for the moment, I moved on the next task for today.

I had to set up a pipeline that could get information from the sensor and display them in the logs. I started by reading the Movesense documentation, where I figure out, I have to use the "wbcmd" application in order to communicate with the sensor. It only works if the programming jig is used and obviously if the sensor is plugged in. I was able to request the information needed through the CMD with the command from the Movesense documentation. However, the pipeline agent would need to run the command, so I copied the "wbcmd.exe" in "C:/Windows/System32/" folder. Now that everything works locally, I

created a pipeline that will execute the command, and then all the information is going to be displayed in the log screen.

Tuesday

Today, I started my journey with a task that its aim is to modify the UART (serial communication) state of the sensor, I have been given some instructions and explanations on how it could be realised. So, I started by documenting myself about the environment and how it was possible to access the property of the sensor to enable or disable the UART. The aim was to use a function in the application file that will change the property. In the settings, a property was needed to be imported.

After working most of my morning on the task, the use of the function was eventually not needed for the pipeline later. Therefore, I have been given a new task to work on. I had to document all the pipeline I created from last week in a "README.md" file. It contains all the requirements to configure and make the pipeline working. It is also described how to create a self-hosted agent and make it useable for the different jobs needed.

Later in the afternoon, we had the weekly meeting for the last week of work, because it has been postponed. This meeting stands usually on Friday afternoon in order to share all the work that we have been doing during the week. There are two other trainees that also work on the Azure DevOps migration. I started the meeting by presenting my two Azure Pipelines, in a nutshell, the first one is able to build the application packages and then flash the Movesense sensor with it and the second one is able to request the sense information to display it in the log. I have been noticed from my supervisor that there was a step not necessary for the process, called the "check out". It checks if the Git repository is updated in the local agent, but any files from the Azure repository is needed for the flashing process. I have to investigate next day about how I could remove this extra step and make the pipeline better.

Also, my supervisor told me that it was not necessary that the pipeline for the building and flashing process runs on the same agent. It is preferable that only the tasks that need a physical connection are run by the Self-hosted agent. No need to investigate more on the error I was facing yesterday.

Wednesday

I started the day without a specific task, but my supervisor recommended us to read an article about the different features of Azure DevOps Pipelines. In a nutshell the article describes different practices is interesting while editing YAML pipelines using either Azure CLI or Azure DevOps GUI. For now, I feel more comfortable using the graphical interface,

so I can have a better look on my error and the assistant helps to create the different tasks of the pipeline.

Then I received a task where I would have to find a way to create a sort of daily report that contains all tasks that the state has been modified today, either to “Active” or “Closed”. The first thing I start to do is to create a query that would request the items that are required. Next, I used the query to display the information in the Dashboard Overview. The different items are in a row and we can access easily edit or have more information about them by clicking on it.

I shared what I have been doing with my supervisor, he told me that we not exactly what he was looking at. The idea for the task is at the end of a report with all the daily change on the work items in mail or message. After a quick search in the project settings, I found that it is possible to create customized subscription to get notify with a mail when certain conditions are filled. I tested it in a personal Azure DevOps project, I was able to be notified when the state of a task is changed. Unfortunately, it still not really filled the requirement of the task.

The general idea behind the task is to have a daily report from an Azure DevOps project where all the items that the state has been changed are displayed. It provides a general idea of what kind of work has been done and what tasks have started. For the project manager, it is a useful tool to assign future work items and to have a global vision of the project state.

Thursday

My day started by investigating more on the report capabilities of Azure DevOps. After a small discussion my supervisor, he told me to check the Azure DevOps REST API. I study for a while the documentation available to learn about the different possibilities. It provides a lot of information, I’m not sure what I will need for the moment, but I’ll start testing the request available with the Sandbox project. I used the software “Insomnia” to test the API request, it simplifies the way to enter parameters and show different useful information concerning the request.

As it was not my main task of the day, I moved to the next one. It was about reviewing the pipelines I made and moving the repository I was using into another one. The aim here is to have only one repo for the different tasks we are working on. I started by cloning the targeted repo locally, then I copied my folder with the pipeline. I was not really used to Git, so I asked my colleague to help my pushing the repository. For that I had to create a new branch and push all of it on it. When the repository was completely done with all the modification and

the pipeline running properly, I created a pull request to the main, so my supervisor can check my code before pulling into the main branch.

The modification I made to the pipeline was for the checkout that was not needed in the flashing process. After some research, I just had to specify in the YAML file to skip the checkout. Another modification is adding a variable for the new path in the repository, by referring the path in the variable it allows the pipeline to run anywhere.

Just after that, I wanted to experience the release pipeline. So, I created a release pipeline that will consume an artifact created by the building pipeline in order to flash the sensor with the new application built. The task was not complicated because it was basically replicating what I have done in my pipeline to generate the build but only using the GUI from Azure DevOps.

Friday

Last day of the week! I started my day by keeping investigating the Azure DevOps REST API and I found a few requests that would be interesting for the report. Most of the requests allow to get information about the different components, but not directly as a list of work items as I would require.

On the afternoon, we had our weekly meeting where we share everything that we did during the week. Ville presented to us the pipeline he has been doing the past weeks. His pipeline runs a Docker container and add different folders through Git command. He previously had some trouble with the SSH connection, he was able to correct. That was an interesting presentation even if at the end my supervisor wanted a different process for integrating the file in the container. He recommended to do this step in the docker file directly.

For myself I had nothing special to present, but I share what I've been doing since Tuesday. Then we had the occasion to discuss a little more on the report task. He recommends me to create a Python script that will utilize the Azure DevOps REST API to display the information. To check if some state has been changed, I'll need to request to compare which items has changed since the start of the day.

I'm not used to Python language at the moment, so I'll have to learn more about it in order to realise the task. I started following an interesting tutorial to configure my Visual Studio environment to code in Python, I installed different extensions for formatting and debugging the code. Now that the environment is ready, I followed another tutorial about Django to create a simple web application. Developers avoid generating HTML directly in code,

because it opens the app to XSS attacks. Enough for the day, I'll keep learning about Python on the next week with a new tutorial and project more oriented on scripting this time.

Weekly Analysis

This week was a little hard to begin because I've been tested positive to the corona virus on Tuesday. Fortunately, the symptoms didn't stay so long, I was able to still work on my daily task and provide matter for my diary thesis. In general, I'm glad about what I was able to achieve in this week of work.

I had never used the YAML language, and I had barely heard about it. I just know it was used to specify configurations. Fortunately, it is a language that aims to be human friendly and easier to learn compared to others format languages (Leigh Brenecki & Suhas, 2021). The Azure DevOps GUI to edit the YML file is very useful, it contains an assistant that allows to configure the different parameters that you'll need for a task. I'll carry on using it for the future, because the CLI provides less information on the potential error in the code, even though the process to queue a build pipeline is quicker. That would be interesting to reconsider that later when I have more experienced. In general, what I learned is that it is possible to use templates. It is a good way to reduce code duplication when you have the same block of code in the same or multiple pipelines. The idea is to load the template in the YAML file and then reuse the code with different parameters that are specified (Medium.com, 2020). Here an example from the Microsoft documentation:

```
# File: azure-pipelines.yml

jobs:
- template: templates/npm-with-params.yml # Template reference
  parameters:
    name: Linux
    vmImage: 'ubuntu-16.04'

- template: templates/npm-with-params.yml # Template reference
  parameters:
    name: macOS
    vmImage: 'macOS-10.14'

- template: templates/npm-with-params.yml # Template reference
  parameters:
    name: Windows
    vmImage: 'vs2017-win2016'
```

Figure 1 : YAML file using templates (Microsoft.com, 2020d)

My understanding of the Azure Pipelines started to be better, but I still had doubts on the various uses and differences of the Build and Release Pipelines. Azure DevOps supports 2 types of build pipelines:

- **Classic build:** Classic editors use the graphical view and create build pipelines according to the requirement.
- **YAML build:** Require a good understanding of YAML syntax to write declarative scripts to define the build pipeline as a code.

The main purpose of a build pipeline is building the code, testing it, and generating an output package. The source code is not necessarily kept in Azure DevOps it could come from another source control system. On the other hand, the Release Pipelines or deployment pipelines are used to deploy web app, function app, logic app and different types of artifacts to the Azure Platform (Chandrasekara & Herath, 2020).

So, in my case, the pipeline made for the flashing of the sensor is more likely to be a Release pipeline. It has to be based on an artifact generated earlier, here it is the build made previously, that will be used for the flash process. Another important point to notice is that the Release Pipeline could be started by a trigger, for example when the build is generated then it triggers the flash process.

Here are all the concepts of Development (Dev) and Operation (Ops). DevOps is a set of best practice that reduce the barriers between developer, who want to innovate and deliver faster and on the other side, operation, who want to guarantee the stability of production systems and the quality of the system changes (Krief, 2019).

Another concept I wanted to clarify, for my understanding is the concept of CI and CD. So the continuous integration is related to the Build Pipeline, a build can be defined to compile each check-in/commit to the code base and then execute all unit tests to validate the code base to ensure stability of the code base, which is generally known as a CI build. On the other hand, continuous delivery and deployment are related to the Release pipeline. The difference between them is the last stage, continuous delivery ensures that every change can be deployed to production, while having the option to hold the production deployment until manual approval is given. Continuous deployment allows every change to be automatically deployed to the production (Chandrasekara & Herath, 2020).

For the Movesense task, I've been using the official documentation, there are unfortunately not a lot of other documentation available, but I can't count on my supervisor to help me when I can't understand. This week, I had to manipulate the UART of the sensor, which allows the serial communication through USB with the programming jig. The UART is very power hungry even when it is not in use, that why it is a functionality it has to be controlled when manipulating the sensor. The serial communication can be commanded in two ways (Movesense.com, 2020):

- The default factory settings disable the serial communication. To enable it, it needs to turn on using the Movesense settings API.
- The application's App.cpp file also contains setting for turning serial communication on or off. However, this application specific setting will not override the master setting controlled via Settings API

I didn't go deeper on the UART task this week, because eventually it was not needed yet. I might come back on that later.

I'm not so comfortable to code with Python at the moment, but with the tutorial it brought me a good insight for the next week coming. The different objects for this task will be "being able to code a simple python script", "coping with the Azure DevOps API to retrieve the work items needed", "treating the information" and eventually "displaying all of them in form of a report". The script might be run at least once per day, to get information on all the work done that day. It would be an interesting tool, or the project management tasks.

In more general with Python, I learned that is a useful coding language, and even if I won't go very deep in the topic for now, it is good to have some skills for the general knowledge. Its syntax is made more friendly with English keywords that allows anyone to easily understand and start coding with Python (Skillcrush.com, 2021). I'll learn more about it in the following weeks, when I'll face different types of requirements, I would be able to experience coding in Python.

3.2 Week 2 (15.02.2021 – 19.02.2021)

My week goal is mainly based on the Python task for the report, I have to improve my skill to provide a functional script showing the daily change among the Azure Board.

Monday

Objective of the day is to practice coding in Python, I would try to make a simple script for the moment following a baseline. Then if I'll be able to use the Azure DevOps REST API, that would be interesting to start manipulating the data.

I spent most of the day looking at two Python courses. I started by a video that covered all basic concept of the language (Mosh Hamedani, 2019). It helped me to have a better understanding to manipulate Python projects. I'm now using the IDE "PyCharm" specially made to code in Python. The second video was about using a REST API in Python with the library "Requests" (CodingEntrepreneurs, 2020). I was not able to replicate the whole exercise, because the API was not free of charge. However, it gave me a really good insight on how I could use the Azure DevOps API.

Then I started coding my first lines of script, where I faced an authentication problem. The Azure DevOps REST API requires some authentication information in order to access the data. After searching a solution on internet, I found that I have to use a PAT in the request. So, I made a variable "header" containing the authentication information that I added at my request. To generate the personal access token, I went on the Azure DevOps settings and created a new PAT for the request (Stackoverflow.com, 2019a). My first request work properly now that the authentication works. Now I have to study a little more the Azure DevOps REST API in order to get the information I want. For now, I can't retrieve a full list of work items from the Azure Board directly.

My objectives of today are not completely done, but I got good knowledge in Python to make different operations in the code. The tutorial helps me understand and apply some concept or programming that I have but for another language. I realise that Python is very simple to use in comparison to others. For getting all the work items, I didn't find a specific request for it, there must be a way anyway let's check tomorrow for more documentation.

Tuesday

For today I'll carry on doing the Python script, firstly is to find the best option to get all the working items through API requests. Secondly, how to process the data to make them usable and thirdly display the information in an effective and readable way in the command line, at least for now.

After reviewing the different requests, I had tested last week and yesterday in “Insomnia”, I find out that the best option would be to get the result from a query made previously in Azure Board to get the IDs number of the work item. Then I’ll be able to retrieve more information from each task with another request. To get the result of a query, it is needed to use a WIQL request using the ID of the query. I made the following query with all the conditions required:

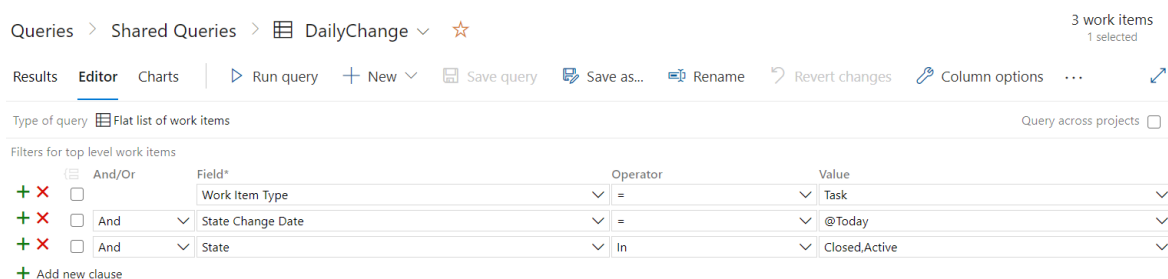


Figure 2 : Query Editor from Azure Board

This query allows me to get all the IDs of the task that fulfilled the conditions. Next step is to request information for each work items. For that, I made a loop on each item to retrieve the following information: "ID", "Name", "State", "AssignedTo", "Date & Time". It was also needed to have a URL to have a direct access to the Azure Board. It is composed this way *“https://dev.azure.com/{Organisation}/{Project}/_workitems/edit/{Id}”*.

Now that I have all information of the work items, I find a way to display all the information with a library called “Tabulate”. It allows to create good-looking table to present the data. In the code, I created an object “table” where I add all the other information of the work item as an object as well. With that, each object will be represented as a line in the table. For the header I used another object where I defined all the headers, I needed inside.

For security reason I researched a way to remove the PAT from the code, because I consider this information important to keep out of the code, especially if it is shared with other people. I found an interesting tutorial to follow, which was about hiding sensitive information in Python code (Drew Seewald, 2020). So, I downloaded the library “dotenv”, I created a file called “.env” in the current folder I copied my PAT directly inside. Now I had to import the library and create a variable that retrieves the personal access token stocked in “.env”. The whole point is to keep sensitive data in another file, if the project uses a GIT repository the file “.env” would be added to the “.ignore”, so it is not pushed and shared in the version control system repository with other people.

Also, for the date format I used the library “datetime”. Unfortunately, the dates retrieved from the request are 2 hours too early than they should be. So, I adapted by adding 2 hours

and I format the date to have a more readable information (Blogspot.com, 2014). Now that everything works, here is a view of what the result looks like from the command line:

ID	Name	State	AssignedTo	Date & Time	URL
522	Another test	Closed	Bertrand Gogniat	2021-02-16, 11:52	https://dev.azure.com/SymbioFinland/SandBox/_workitems/edit/522/
527	Hi From Azure Board	Active	Bertrand Gogniat	2021-02-16, 11:32	https://dev.azure.com/SymbioFinland/SandBox/_workitems/edit/527/
511	I'm only a task	Closed	Bertrand Gogniat	2021-02-16, 08:21	https://dev.azure.com/SymbioFinland/SandBox/_workitems/edit/511/

Figure 3 : Result of the Python script

At the end of the day, I feel a little more comfortable coding in Python, even if the script is simple. I learned about some good practice, for example not having sensitive information in the code and how to hide it. I'm also now more comfortable with the Azure DevOps REST API, I understand the potential to use it and what it is possible to realize with it.

Wednesday

The principal task of the day is to improve the scrip in different ways. I got a feedback from my supervisor and for now the script retrieve the data needed. However, I'll have to check for the PAT, to make sure it uses the minimum permissions. Also, the script should be able to send a mail and possibly a notification or message on Microsoft Team.

I started to check the PAT permission, and I configured previously a full access for testing it. So, I reduced right to the minimum which is ready only on "Work Items, queries and backlogs". It allows a better security of Azure DevOps to reduce to the most restrictive permission, it is a good practice to keep.

Then I investigate on which possibilities there are to send mail in Python. I find out different options, most of them were using the library "smtplib" which allows to send mail through SMTP. I decided to use it because it looked simply to use. After gathering more information, I notice that I would need an SMTP server in order to send the mail. I'm not aware if I can access one from *Microsym* at the moment, so I chose to use the Gmail server for testing it.

I follow some instructions from a YouTube video (Edureka, 2020) to configure it. I had to use my personal Google account and "allows less secure app" option to access it from my Python program. After writing a few lines of code, I was already able to send an email to a group of receivers with the different results of the changed task. Here the result of my testing:

Daily Report



Azure DevOps <[redacted]@gmail.com>
19:32

À: [redacted]@hotmail.com

Here all the task to has its state change today :

527 | Hi from Azure Board | Closed | Bertrand Gogniat | 2021-02-17, 11:50 | [https://dev.azure.com/\[redacted\]/workitems/edit/527/](https://dev.azure.com/[redacted]/workitems/edit/527/)

Figure 4 : Mail sent by Python script

Later the afternoon, I had a meeting with my colleagues Joonas and Ville. We had to talk about the Movesense Pipelines. Now that Ville's pipelines are working, we will need to integrate the process of flashing the sensor. All the environment is in a Docker container that is run by the azure pipeline agent. To flash the sensor, it requires the programming jig physically connected the host. So, my task is now to investigate if there is a way to access the USB port of the host from the docker container.

After some research, I noticed this is a real issue, officially recognized by Microsoft (Microsoft.com, 2020b). It is not possible to access the port of a Windows host through a docker container, this feature is not working. However, there was some alternative proposed like "USB over networks", where it allows the docker environment to access the USB port through an address IP and port to the host. After some research it seems not to be a viable solution for the actual project, also most of the available software are not free of charge.

Thursday

Today I will have a few tasks to look into, first I'll give a small introduction to my colleague about the Azure DevOps for project management and how to manage the work items through an excel file. Secondly, I'll have to configure a self-hosted agent on a Raspberry Pi to run different Azure Pipelines and testing the Docker container that my colleague made with the flash process. The script is not a priority for the moment, I'll focus on configuring the Raspberry Pi.

The presentation went well, we shared different information about the project and more general, how the project management is in *Microsym*. I was quite interesting about this, so I'll be more in contact with Eleonora for tasks related to project management. I created an excel file to retrieve all of the work items for the project, with the add-in "Team" that I configured some time ago. The excel is a powerful way to manage all the work items in a project in Azure DevOps. Updating, creating or removing is a time-consuming task to do through the GUI, especially when there are multiple work items.

On the afternoon, I received the Raspberry Pi. I look if there was any Linux distro better for running Docker and Agent service. There was no result that stood out, some just appeared more often than the other, for example, "CentOs", "Ubuntu" "Alpine linux". We decided to go with Ubuntu as it is well used and documented distro. I'm using the Ubuntu Desktop for Raspberry Pi, I went through all the usual configuration. Then, when I was ready, I started to install the Azure Agent service for arm64 architecture. I followed the official documentation (Microsoft.com, 2020c) to configure the server, pool, work directory and PAT. I also set the agent to run as a service, so no need to run it manually.

As Docker was needed, I installed it based on the docker documentation (Docker.com, 2020a) everything went well and I was able to run docker images. Nevertheless, docker would need more access to be run without "sudo", it is also needed by the agent service. After some research on the net, I was able to fix the issue with a tutorial (Stackoverflow.com, 2018) on how to add docker into groups with higher permission.

Next step I decided to test the images created by my colleague in Azure container registry. For that I had to install Azure CLI and log in with my *Microsym* account. Then docker had to be logged in the container registry and eventually I was able to pull the image and ran it locally. However, I faced an error because of the different CPU architecture of the host and the docker image. Raspberry Pi uses an ARM architecture, and the image required an AMD64. I search on many sources how the issue can be solved, there is no direct way to fix it because it is not possible with the architecture, I find a way to emulate an AMD architecture on the host, unfortunately it is slower to use an emulator but it is the only solution to have something functional. The solution is Qemu, I followed the documentation from Raspberry (Raspberrypi.org, 2019), and I was now able to run the image properly even if there is still the warning about type of CPU architecture.

Friday

The priority task for today is to work on the Raspberry Pi and configure it to run docker images and doing some testing the programming jig. Then, aim is to be able to run a pipeline on this agent, so it can flash the Movesense sensor through . The actual process of flashing has to be done inside the docker images with the file generated earlier in it.

I started to run the docker image and this time I was able to run it, still the warning message about the architecture, but it is working. I used the command line "nrfjprog -i" and got "50125043", this means that the programming jig is connected to the host. Then I use the spell to flash the sensor "nrfjprog -f nrf52 --sectoranducrerase --program Movesense_combined.hex --reset" but I got an error that the command didn't work at all. I checked the log and the docker image couldn't access to the J-link debugger. After some

research I noticed that the flag “--privileged” was needed, it provides more permission to the docker image to access, for example, the devices of the host. Now I’m able to flash the Movesense sensor from inside the image.

After that, I improve a little bit my set-up, I configured a remote desktop connection on the Raspberry Pi. I had to install the software “xrdp” and configured some option for the service to be able to run. Then I got the address IP of the Ubuntu machine with “ifconfig”, I had to install some web tools before that. Now I can use the “Remote Desktop Connection” from Windows to access the Raspberry Pi (Linuxize.com, 2020).

Later the afternoon we had the weekly meeting where we present our work of the week. Saku started to present a small demonstration of the Redis project he was working on by manipulating some data from the DB. Then I introduced the team to the new Azure agent on the Raspberry Pi 4 I configured, I demonstrate the flash process and explained the different events I faced when set the machine up.

My supervisor introduces us to the next step, it will be to create a Release Pipeline to flash the sensor triggered by the artifact previously generated by Ville’s Build Pipeline. The aim is to run the docker image and flash the sensor from it with the devices connected to the host. Almost all the steps have been done previously so it won’t be too hard to combine, but I wonder if it will be easier to just install the required program to flash the sensor on the Raspberry instead of using the docker image.

Weekly Analysis

In general, this week has been fulfilled with news interesting knowledge and nice discoveries. It all started with the Python script where I was able to retrieve all the information needed and display it in the command line. I was then a little confused about what the nature of Python was, a scripting or programming language. A scripting language is interpreted so it is not necessary to compile then to be used. Scripts can be executed directly on the host and it is usually used to automate certain tasks or glue software together. On the other hand, a programming language is usually more complex, and it needs to be compiled before in order to be executed.

The main difference is that scripting is an easier, faster and simpler language and required a host, because scripts are run inside another program. On the other hand, programming language is relatively difficult, requires more code to be written in general and are self-executable, they don't require a host (CodingNinjas.com, 2018). Thus, Python is often described as a scripting language due to his ease of use and its role of glue to control and direct other components. Even though, Python has a simple and flexible feature set, it allows programs to scale up in sophistication as needed. In more general Python can be either a programming or a scripting language depending on and the task and the interpretation of the person that use it (Lutz, 2013).

Also, still related to Python, I have been experiencing the naming convention with PyCharm. It is not an obligation to follow the exact syntax, the code still works, but it improves readability and consistency of the code. I'm used to Java naming convention with the CamelCase for variables and functions, but in Python, it is only used for the class names. For function and variable names, it should be lowercase with words having underscore in between as much as possible. On the other hand, constants are usually defined on a module level and named in all capital letters with underscores separating words. There are also variable names to avoid, "O" (letter "oh" uppercase), "I" (letter "eye" uppercase) as a single, or "l" (letter "el" lowercase) as character (Python.org, 2013).

Later this week my script was able to retrieve data from the Azure Board, for that I had to use a query with the WIQL request, which I didn't really know what it was all about, I just know I was able to get the result from a query. WIQL stands for "Work Item Query Language" consist of a "SELECT" statement that retrieves all columns specified and filling the conditions. This language allows to create queries in order to retrieve work items with specific conditions (Microsoft.com, 2020f). The syntax is similar to SQL, here an example:

```
SELECT [State], [Title]
FROM WorkItems
WHERE [Work Item Type] = 'User Story'
ORDER BY [State] Asc, [Changed Date] Desc
ASOF '6/15/2010'
```

Figure 5 : Example of WIQL syntax

I'm also able to confirm the process I chose to retrieve the different work items was legitimate. In Microsoft documentation, it is specified that there is no way to get full information of work items through a WIQL request. It returns only the work items ID, it requires to perform two steps. Firstly, it is to get the ID from a WIQL request and secondly to retrieve the work items via a "GET" (list or single work item) request (Microsoft.com, 2020f). So that's the steps that I implemented during this week.

The other main task of the week was to install the Raspberry Pi, it has been a long time I haven't played with Ubuntu and I was especially surprised of the good-looking of the interface and how user-friendly it has become. Here, I wanted to know a little more about why we couldn't access a Windows host port from a Docker container and why it is working only with Linux distros. Docker recognize that it is not possible to pass through USB with Docker for Windows, it seems to be due to "Hyper-V" that not offer the ability to bind local hardware resources to VMs because it would limit Live-Migration capability (SocialTechnet.com, 2019). There is a way for Windows host and Windows Docker containers to pass through the USB, but there is not more information about how it works exactly (Linuxize.com, 2020).

Otherwise, sharing Linux hosts' devices with Docker image is way easier. In my case I didn't even have to use the flag "--device" to bind my USB device to make them accessible. When I run the Docker image, it was already accessible. However, it required more permission to be actually used, I find it out with the J-link documentation that the flag "--privileged" was required (Segger.com, 2019). I was able to see the devices connected to the host, but not able to interact with them as I didn't elevate the permission when I ran the Docker image.

3.3 Week 3 (22.02.2021 – 26.02.2021)

Week goal is to set up the Raspberry Pi properly so it can run the docker images from the Azure Pipeline. It is the priority for now, because the hardware is needed for my colleague working on the project. When I'll have more time, the Python script will be the second priority for this week. Also, I have been in contact with Eleonora (Project Manager) and I'll probably work on a new project, but more oriented on the management side.

Monday

Today's task is to create a pipeline that will flash the Movesense sensor from a Docker image. The pipeline is run on the self-hosted agent "Raspberry Pi" which has the programming jig and the J-link debugger plugged in. The flash process has to be done inside the Docker image, firstly because it contains the different tools for the process and secondly for more portability of the system, in other words, the pipeline could be run on any agent that support Docker and the tools will be updated automatically through the Docker image.

I started to create the ".yaml" file for the pipeline. Here I had the choice to use a Release or Build Pipeline, but as it is not really a deployment process, I'm using a Build one for now. In the code, it is required to specify the pool of agent to use and also the container used for the process. Here is a view of the YAML file:

```
1
2  variables:
3    pool: 'Self-hosted'
4    dockerImageName: 'movesense:651'
5
6    pool: $(pool)
7
8  container:
9    image: [REDACTED].azurecr.io/$(dockerImageName)
10   endpoint: ACR
11
12  steps:
13    - task: DownloadBuildArtifacts@0
14      inputs:
15        buildType: 'specific'
16        project: '[REDACTED]'
17        pipeline: '46'
18        buildVersionToDownload: 'latest'
19        downloadType: 'specific'
20        downloadPath: '$(System.ArtifactsDirectory)'
21
22      - script: |
23        nrfjprog -f nrf52 --sectoranducirerase --program Movesense_combined.hex --reset
24        displayName: 'Flash command'
25
```

Figure 6 : YAML Pipeline to flash the sensor

Unfortunately, when I ran the pipeline, I faced an error when the docker commands are used by the agent. The container seems to not be running, but the previous lines show that the Docker container has been started and should be up for the rest of the commands:

```
36 /usr/bin/docker start abac231056ef30c86987287a7339fbff486fe5d0397d557f440a6841dd7a9228
37 abac231056ef30c86987287a7339fbff486fe5d0397d557f440a6841dd7a9228
38 /usr/bin/docker ps --all --filter id=abac231056ef30c86987287a7339fbff486fe5d0397d557f440a6841dd7a9228 --filter status=running --no-trunc --format "{{.ID}} {{.Status}}"
39 abac231056ef30c86987287a7339fbff486fe5d0397d557f440a6841dd7a9228 Up Less than a second
40 /usr/bin/docker exec abac231056ef30c86987287a7339fbff486fe5d0397d557f440a6841dd7a9228 sh -c "command -v bash"
41 Error response from daemon: Container abac231056ef30c86987287a7339fbff486fe5d0397d557f440a6841dd7a9228 is not running
42 ##[error]Docker exec fail with exit code 1
43 Finishing: Initialize containers
```

Figure 7 : Logs of the YAML Pipeline

The logs show the start command has worked but then the container just suddenly stopped and the “docker exec” is not working because it is not running. Furthermore, I tried to run the pipeline with the Azure Pipeline agent, and it worked just properly without issue. At this point I don’t understand what could be wrong with my Raspberry Pi, except the warning message that I still have about the architecture, might be an issue. To find out more about it I did some testing locally, but I didn’t come up with something.

Later in the afternoon I had a call with my Colleague Joonas, we tried to make it work and tried different scenarios locally. Unfortunately, we didn’t come up with something after many tries. I also give a quick testing with my computer as an agent, I had previously configured for, and I got the exact same error, the container doesn’t seem to stay up and running. We learn from a forum that the container could possibly stop after being started if its main finishes, but it wouldn’t make sense in our case as the container work properly with the other agent.

Before the end of the day, I attempt to reinstall Docker on the Raspberry Pi following the viable tutorial I found earlier. I deleted all configurations and general file related to Docker. Then I reinstall it steps by steps and I try again to run the pipeline from Azure DevOps. Unluckily this attempt didn’t work the problem still remains.

Tuesday

Today’s priority is to make the pipeline for the flashing process working. The hardware has to leave in the middle of the week to go for one of my colleagues. Also, today, I’ll have a meeting with Eleonora and Tapio, both of them are project managers, where they will introduce me to the essence of project management in *Microsym* and what kind of updates would have to be made on the actual process in the future.

This morning I tried a different way to run the docker image. As the command “docker start” won’t keep the container running, I tried “docker run” with the flags “-t” (tty) for to allocate a pseudo-TTY and “-d” (detach) to run container in background. The command creates a

container that will stay up and running. I can access and run commands inside with “docker exec”. Now that the process works locally, I’ll create another pipeline with all the commands needed. It consists of (1) pulling the docker image, (2) running it with the different flags, (3) executing the commands inside, (4) stopping the container and (5) removing it with the flag “--force”.

Now I’m facing another issue, to pull the latest docker image, I have to specify its name in a variable of the YAML file. Which means that every time there is a new build available, we have to adapt the value of the variable and trigger the pipeline manually. There are different ways to do it here, one option would be to use the latest docker image available from the build pipeline, which contains the “.hex” file for the flashing so and find a way to trigger it. The other option is to use the Movesense base image, which can be pulled with the tag latest, but the files for the flashing have to be downloaded and copied inside the docker image. For now, I don’t have enough information from my colleague to know exactly what to do, at least the pipeline is working, but just needs some readjustments to work properly.

The rest of the day was about to project management meeting, we discussed about how the project was handled in *Microsym* and which frameworks are used. Some templates would have to be updated for a new process that integrate more details about responsibilities and how projects should be managed. For next week, I’ll have to read some of the documentation and think about what update can be done to improve the project management in general.

Wednesday

The objective of the day is to finish the pipeline and optimize it at the maximum, because the hardware has to leave today for Oulu’s office. It will require to find out how to pull the latest docker image from the Azure container registry with the latest build ID generated by the pipeline.

After some discussion and research with my colleagues, we find out that it was possible to retrieve data from another existing pipeline. To do so, it requires to declare a resource referring to the pipeline that generated the docker image by its name:

```
3 resources:
4   pipelines:
5     - pipeline: resourcevars # identifier for the pipeline resource
6       source: AzureDevOpsReferences # source pipeline definition name
7       trigger: true
```

Figure 8 : Resource tag in YAML pipeline

Now it is possible to access different information about the pipeline. See (Microsoft.com, 2019) for more information about all the value accessible there. With the “`$(resources.pipeline.resourcevars.runID)`” variable, it provides the latest run ID of the build pipeline, which it has been used previously to name the docker image generated. It can now be used in the YAML file. The pipeline is able to pull the latest docker image available in the Azure container registry.

Next step was to present the pipeline running to my supervisor and see if it meets his expectations. It was mostly the case, but he suggested some modifications, for example that the users could specify the image to use by manually enter the “buildId”. Also, I raised an issue about the docker image, which is; each time one of them is pulled from the agent, the oldest ones are not used anymore. It is a problem as the agent is almost running out of disk space.

When the pipeline is triggered, it is possible to assign a value to a variable. So, I configured the variable with the default value as “`$(resources.pipeline.resourcevars.runID)`” and the option “Let users override this value when running the pipeline” on. This variable is accessible in the YAML with `$(buildId)`. Now, when the user wants to use a specific docker image, he can refer its “buildId” instead of the default value and run the pipeline.

The issue about the old docker images was a little more difficult to realize. I first looked for the exact docker command that will remove all the images unused, but I realized that it would be better to keep the latest one so when the pipeline uses the docker pull command, it is not needed to download the complete image each time. Keeping the latest one will allow to download only the layers that have been changed from the last image.

I first find out the docker command to retrieve all the images before the used one:

```
> docker images --filter "before=devopsreferencescontainerregistry.azurecr.io/movesense:$(buildId)" -q
```

It displays only the id (flag “-q”) of all the images before the one specified in the filter. So now I’m able to use “docker rmi” command with the flag “-f” to force all images to be removed. Here the final command used in the YAML pipeline:

```
> docker rmi -f $(docker images --filter "before=devopsreferencescontainer-registry.azurecr.io/movesense:$(buildId)" -q)
```

Unfortunately, there is still an issue when the pipeline is run two times in a row, with the same image. The error is caused by the “docker rmi” command, as there is no result from “docker images” command, it misses a parameter to run properly the first command, so that will cause “error exit with code 1”. It should not cause trouble for the moment as the flashing

process is done anyway and the user will flash only one sensor for testing and each time with a new image. However, it is something to keep in mind if the pipeline begins to be more important for other applications.

Thursday

Today is going to be a little calmer than the others, but still some objective to work on. The first task will be to come up with a working solution to remove the docker image unused from the pipeline and second task will be more related to the project management side, where I'll review some existing documentation and template from *Microsym*.

So, the problem I'm facing is that if the "docker images" command from script doesn't return any result then the "docker rmi" won't work and will raise an error that will cause the pipeline to fail. I got confirmation that the pipeline could be triggered multiple times with the same docker image to flash different sensors, then it is necessary to fix that.

I looked for more information about bash scripts, I find out different ways to test if a command returns some results or not and I combined it with a condition. So now, when there is no result nothing is done and if there are, it will run the "docker rmi" command with all the image id to remove. I was not able to test it with the Raspberry Pi agent, I tried with my Windows agent, but I was not working. I couldn't run the bash script because the agent didn't have enough permission for it. I tried to fix it, by giving administrator access to the service on my local computer, but I still have trouble to make it work. So, I decided to use the Azure Pipeline agent to test my script, but with different commands to see if the condition works properly. It was the case, so next step will be to test it with the Raspberry Pi when it will be available again.

The rest of the day I've been reading some documentation and templates for the project management task, where I learned the process and how to handle projects at *Microsym*. Most of the documents are old and would have to be rethought for the project management task. However, for now I need to search more and get more information before proposing some rethink on the documents. I'll keep on doing little by little.

Friday

Nothing special is planned for today so I decided to carry on doing my Python script for the project management task. I still have to find a solution for publishing message or notifying people through Teams' application about the daily change on projects.

I started by looking on internet different solutions available for that task, I find out "pymsteams" was especially made for this kind of job. To use I had to set up a Webhook on

a teams' channel, then to be able to use it I had to copy the URL of the service into my Python script. The library is very simple to use, after just a while I was already able to post a formatted message in the channel chat. Here an example of what it is looking like:

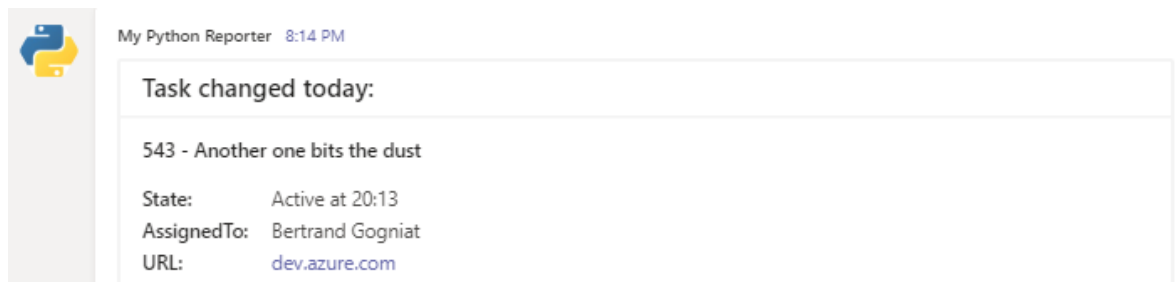


Figure 9 : Message published by the Python script in Teams' chat

Sometime later I got a new task to do from my supervisor, it was about to check and test an Azure Pipeline extension for Microsoft Teams. It allows a channel or a team to subscribe to different information related to the pipeline of a project. The extension requires a Microsoft App account to acquire it. The installation and the configuration of the bot went through easily, and I was able to try some pipeline subscription from the Sandbox project. The way it works is by typing “@Azure Pipeline” in the chat, it allows you to interact with the bot and specify the command you want to use.

On the afternoon we had our weekly meeting, Saku presented his work related to Rabbit MQ, which is an open source message broker software that allows applications to connect to each other. He shows us a simple application to send messages to different types of instances through RabbitMQ. Then I presented a little bit what I was doing with the Python script, it was interesting to get some feedback from the team. I also get the information that next week it is going to be used in a project to help the project management side.

Weekly analysis

This week has been mostly about the Flash pipeline which was needed for the development process of the project. The aim behind using the Movesense docker image is to have all tools and a working environment that could be used on almost any operating system. Also using the Raspberry Pi allows the developer working with the Movesense sensor to have a plug and play environment to start his job.

The solution of using directly the docker commands through a script is not my recommendation and it is probably not a best practice. It would have been better to use the container option and the service connection as described on Figure 6. For multiple reasons, firstly because the task is literally “automated”, log into a container registry by the service connection, pulling the image, creating a docker container, starting it, executing the command required, removing the container and image and finally logging out of the container registry are all the steps are done by it.

As it was not working for a reason that I’m not aware, the other option was to replicate the process directly with the commands through the command line script. Secondly, because it is more secure to use the container resource with the service connection, as the container server name is hidden in the logs. With the solution proposed, it is required to log into the Azure container registry when configuring Docker on the Raspberry Pi. Also, the name of the server is hard-coded in the YAML file. Thirdly, related to the second point, the pipeline won’t be as generic as it should, for example, if there are some changes on the Azure container registry the values used in the commands would also need to be updated.

Now, why the container was not able to stay up and running is a good question. When it was run on the Azure Pipeline agent there was not this problem, only when self-hosted agents were used. It seems that this error has happened to other people as well, but in some different contexts. The cause could be due to the Docker image has an ENTRYPOINT defined in the docker file, but it is not the case here. This instruction tells us which command to run when a container based on this image is started (Goasguen, 2015). Just like with CMD, it is required to specify a command and parameters. In addition, according to some forums, it could also be caused by the agent’s lack of capabilities (VisualStudio.com, 2019). In this example the Linux Alpine version needs some requirements that are not necessarily installed during the configuration of the self-hosted agent. “Bash”, “glibc-based”, “Node.js” are required in Linux-based containers. If this pipeline is going to be more used, it would be interesting to rework it with all the requirements here. That will probably be the case when I’ll have more time for it. As it had a deadline to respect, the solution used is enough for now.

Furthermore, I wanted to clarify the flag “-t” in the docker command, which is described as “allocate a pseudo-TTY” in the documentation, but it doesn’t speak to me at all. So, it is basically a pseudo / virtual terminal (also known as PTS) within the container. STDIN and STDOUT are connected concepts that are simply the input and output streams of a sequence of instructions. Using it with the flag “-i” (for interactive option), allows users to interact with a terminal by entering command and read the output through the standard input / output describe previously (Stackoverflow.com, 2019b). Reflecting that with the commands used in the pipeline, it allows the agent to interact with a terminal to run the script as the input and get the output in Azure pipeline’s logs. It is possible to use the flag “-it”, which combined these two parameters.

Let’s find out what will be what is the aim behind the report task for project management and what advantages could it bring. Obviously, reporting refers to providing a high-level and simple overview of the project, but it provides a view on what is going on and what to do about it for the whole team of the project. There are different types of project management reports, in this case it is more a “Project status report”. Which is allowing stakeholders to get an overall snapshot of the advancement of the project to guide it toward its objectives. It is a simple update designed to keep tracking progress, rise problems and key points to note, just with a quick view (Smartsheet.com, 2021). For now, the Python script only allows to get the daily state changes among the task of a project, but it is a good start to have an overview on the work that has been done. The other type of report can be “Project health report”, “Team availability reports”, “Risk reports”, “Variance report”, “Time tracking report” (Kissflow.com, 2021). I don’t know yet how the script would evolve but will be interesting in digging more in these kinds of reports if needed in the future. Also, in the Azure Dashboard, it is possible to get good overview of the project, but the advantage to have it in Teams is that it is a communication tools and used by everyone in the company.

3.4 Week 4 (01.03.2021 – 05.03.2021)

The horizon of this week is not completely defined for now, but it will concern mostly the project management task with setting up the Python script for an ongoing project and review some of the organization's documentation. Also, reviewing the Flash pipeline when it would set up in its environment.

Monday

This day has been dedicated to documenting the Python script in a README file. This will help other people to understand how it works and ease its use. I've been bringing some minor changes as well, for example some naming correction and hiding the web hook URL in the ".env" with the other sensitive information for more security. Also, I created a new branch on Azure Repository to push my project there, but before that I added a ".gitignore" file to avoid the ".env" file to be pushed with the other files as it contains personal information too.

When someone else is going to clone the project locally, they will have to create their own ".env" file and fill it with the information required for the script. I described all the libraries required and the configuration for the different services (email and webhook). Then, when the script will be functional for a specific project, I'll create a pull request to join the main branch and be up to date with it.

Later in the afternoon I was been asked from my supervisor to search for a solution to show the time elapsed before a work item is closed in the Azure Dashboard overview. After some research I find out in the Microsoft documentation that there are two possibilities to display this kind of information. It is either a "Lead time" widget, which represents the day between the creation and the closing of the item or a "Cycle time", days elapsed between the active and closed status (Microsoft.com, 2020e). Basically, it shows how many days in average to close a type of work item (Epic, Feature, User Story)

This day has been a little less intense, but still interesting on different points. Multiple times I had to rethink how I could make the README file even clearer and with better explanations. It is a good exercise, also to see if we have a good understanding of what we did. The new widget discover today would also be useful for creating a complete dashboard overview in Azure DevOps for projects I'll work on.

Tuesday

The task that my supervisor gives me today is to create a new Python script that will be able to add a new tag on a list of work items. So, script will have to use the id of the work item to

modify it. In the Azure DevOps REST API documentation, I found out that there is a PATCH request, which allows to update the attribute of a work item. The request also needs a JSON file where all the actions are specified:

```
data = [{
    "op": "add",
    "path": "/fields/System.Tags",
    "value": ""
}]
```

Figure 10 : Request body parameters

In this case “op” (operation) has the value “add”, “path” is the location of the tag in the data structure of the work item and “value” for the string value of the new tag. The request was working just fine, but my issue was that it was not keeping the older tags. It was only replacing the value by the one specified, so I will lose some information about the work items.

In my script I have then created another request that will get the information of the work item before and check if it contains tags. If it is the case, the older are added in the string of the new tag of the PATCH request. After some minor adjustment and making my code a little proper (creating functions, check for errors and debugging). I also try with work item id that was not existing, as I didn’t think about it before, it was not working. For the first request I’m now checking if there is a result from the GET request, if it is not the case the work item is skipped and continue with the other ones.

Then at the end of the day, I shared my progress to my supervisor, and I’ll wait tomorrow for some feedback about it. I also pushed the new script on Azure Repository, and I’ll have to document it later. I was glad to have something working at the end of the day, I learned more about Azure DevOps REST API possibilities and it was simple to implement with all the documentation available. Otherwise, my Python skills are getting better, I feel way more comfortable coding and also because I don’t need to search for so much Python documentation during working on the script compared to the beginning.

Wednesday

Today objective is to review the Python script a bit and document the “add_tag.py” on how it works. Also, I’ll read different documents from the company about the project management because I’ll have soon a meeting with the two other project managers.

I added some explanations in the “README.md” file about how the script work and what are the requirements to run it. As the script is using a PATCH request, the PAT I created

earlier is not working because it doesn't have enough permission to execute the request. So, I created a new one from Azure DevOps with "Read, write and manage " access and I added in the ".env" file. I improved the script a bit so when a work item id is not existing in Azure Board, it informs the user in the command line. here what the CMD looks like after running it:

```
512 - <Response [200]>
615 - <Response [200]>
1000 - <Failed, not existing>
```

Figure 11 : Overview of the work done by the script

Now the user has better overview of what has been done and if there is error occurred. He can see all the work item ids that have been added the tag to them and with the response of the request. Response "200" is when the request went well and if there is an error with a work item id it displays "Failed, not existing". Pretty simple but it might evolve in the future when it will be used in real condition.

The rest of the day was about reading the documentation and processes for the project management side. I won't be sharing details about all of that for confidentiality reasons, but now I have a better understanding of the subject and most importantly how *Microsym* manage their projects.

Thursday

For today a new, interesting task has been given to me. It is to investigate on Logic Apps, a service available from the Azure Portal. I haven't been provided much information about it yet, but the first phase is to study about the different use cases and possibilities with the service, then test it with a tutorial and, at the end, probably it will be used for a concrete integration with another application or service.

I have been watching different videos about the topic, so to sum up Azure Logic Apps is a service that allows to develop applications in order to ease the data integration across the company. It is a tool to build powerful visual integrated solutions that connect diverse business app and services together. It can be described as a serverless technology that reduced DevOps, allow faster time to market and implement scalable integrations and workflows in the cloud. Its actions can be triggered by various connectors, for example a simple tweet. The actions could be represented as a box that get an input and transform it into a new output with the operation configured on it. I was not able to test Azure Logic Apps right away, because it has pricing plans and as I was not sure if I were allowed to create one, I preferred to ask and wait for an answer tomorrow.

Most of the use cases I saw using the Azure Logic App were simple integration with Tweeter, where when a new tweet was created it sends some notification through Teams, emails or SMS. I didn't see any concrete or more complex integration for now, but it will also help to have as an example from *Microsym* to work on. Let's find it out tomorrow.

Friday

Today, I have a meeting with Eleonora about the company templates and documentation where she is going to show me some changes that she made and explain to me different things about it. For the rest of the day, I'll be focus on the Azure Logic App and try to implement a working test workflow until the end of the day.

The meeting went well, we discussed about the template of the "Project Plan" and she showed me how she improved it. She explained to me that most of the time this document is not filled out for the organization's project, so it makes it hard to join an ongoing project. As it was too complex and it was containing many not relevant titles, she removed some to make it simpler, so the project manager won't spend so much time to write a project plan. After that she had another meeting and we decided to continue Monday.

Then my supervisor gave me more instructions for the Azure Logic App and gave me access to a MySQL database on Azure Portal. Then, I have downloaded MySQL Workbench, connect to it and I was able to create a new schema with a table for the data. The aim behind the Logic App is to integrate data that comes from sensors that measure temperature of the water into the enterprise environment. I also downloaded Microsoft Power BI to analyze the data later on.

On the afternoon we had our weekly meeting, where my colleague presented their work. Ville was working on another project with a 3d Visualization application that he was able to program different movements for it. Then, Saku was working on object detection with Tensor flow and his program can detect various models. For my turn I didn't present anything as the script was not very relevant to demonstrate here, but we talked about the Logic App and my supervisor gave me more information about it.

The rest of the day I started to create a workflow on Logic App, where I added a trigger with POST request containing a body with mock data. Then when the request receives something, data retrieved are sent by mail to my personal address. The workflow was easy to create as the UI is simple to use, next step for the next week will be to add data into the MySQL database.

Weekly analysis

Overall, this week has been a little less intense in terms of work, but I was still able to discover some new technologies and learn different things. The new functionality of the Python script will be interesting for big projects with a lot of work items in Azure Board, when it is required to add a specific tag to a list of tasks that are not necessarily related. Doing it from the GUI is tedious and takes a long time, but it is still possible to do this way. Another way to do it, that will suit better medium or small projects, is using an excel file to manage the work items. In general, working with the excel file is an efficient way to manage them and especially for the creation of the project structure and all the work items.

This week I faced only a small problem when I was writing the script, as I was using a POST request, I didn't have enough permission with the PAT I was using in the first place. I figured it out quite quickly as the response status codes of the request was "401", which indicates that the request didn't work because the credentials referenced were not valid to authenticate on the selected resource (Mozilla.org, 2021). Then, I created a new PAT with the necessary permission to manage the work items, added it in the ".env" file and the problem was solved.

Later in the week, I was diving into Azure Logic App. It is a topic that I did not have much knowledge at the beginning. After spending some time on research, I was able to come with a good understanding of it. As there are more and more data that are collected in businesses, it is such a necessity to be able to connect it with existing systems. Logic Apps is an integration platform as a service (iPaaS) that allows to automate the process and workflows easily through a simple GUI. It eases the communication among different app, service, systems across the company. It allows developers to be more focus on the app's business logic and functionality than on building, hosting, scaling, managing and maintaining the apps (Microsoft.com, 2020a).

I also encounter many times some content about Azure Functions and how it has some similarities with Azure Logic Apps. I didn't know much Azure Functions I decided to clarify how they are working and what differences they have. First, these two are services that enable serverless workloads. Azure Logic Apps provide serverless workflows to integrate data, while Azure Functions are designed to be a serverless compute service. Orchestrations are made in a different way in each service, in Logic Apps it is a collection of steps or functions that are called "actions" and are executed to accomplish complex tasks (Microsoft.com, 2018). On the other hand, in Azure Functions allows to run event triggered code without to provision explicitly or manage infrastructure and it is required to write code and it is designed to accelerate simplify application development. Logic Apps tend to be easier for non-technical users to manage as it is only monitored in Azure Portal, whereas

Azure Function can also be monitored down to code level via application inside for more technical users (Walkerscott.co, 2020).

It was my first time using MySQL database with the workbench tool for designing and building databases. I already had experience and knowledge with SQL language and Oracle, so I didn't have trouble to set it up as the process is quite similar. Also, MySQL Workbench made is very easy to connect and manage it through the GUI. To compare a little bit with Oracle, MySQL is free and open-source and suit better for smaller businesses. On the other hand, Oracle is licensed for commercial purpose and more suitable for large-scale deployment (Xplenty.com, 2020). Here the choice of a MySQL database is completely adapted as it is, at least for now, only for testing purpose and simple to use.

Then I had used Postman software, which was also the first time that I used it. As specified previously in the diary journal, I have been using Insomnia to test some APIs requests. However, I was advised by my supervisor to use Postman, as it is way more complete software. It is a scalable API testing tool that allows developers to create, share, test and document APIs through a friendly GUI. As it is an interactive and automation tool, it improves testing time, quality of the project and bugs detection (Blazmater.com, 2020). With it, I was able to quickly test my POST request and trigger, the Logic Apps I made. I was testing a simple workflow that when the HTTP request is received, it sends an email with the data contained in the body of it. The body is data sent by the client to the API, in my case it has a JSON structure and I had to specify in the request configuration.

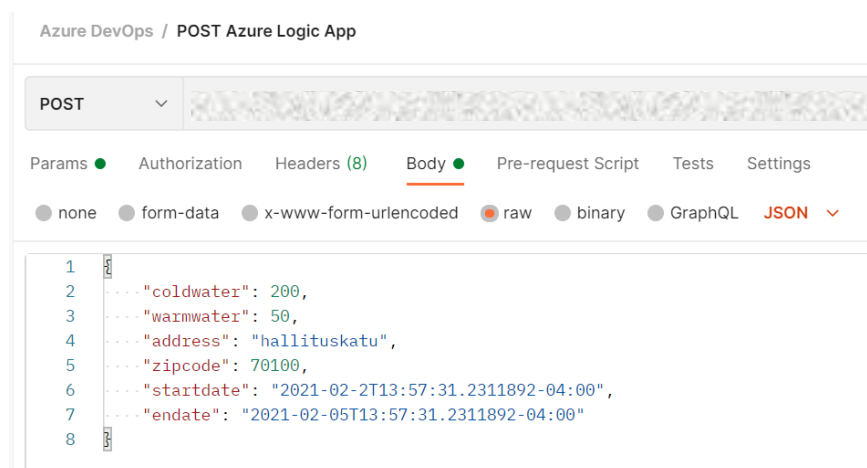


Figure 12 : POST request with JSON body in Postman

3.5 Week 5 (08.03.2020 – 12.03.2020)

This week goal is to set up a complete Logic Apps that are able to get data from a POST request and insert it in a database. The environment is not defined for now, because it might be changed. It is only testing. Otherwise, there will be some improvement and probably a new task for the Python script.

Monday

Today will be dedicated to Logic Apps and doing more research about it. I'll talk to my supervisor to clarify what to do exactly for the task. If I have some extra time, I'll also review a few documents and templates from *Microsym* about the project management.

After a small talk with my supervisor, he sends me a few links to search for more information about using the MySQL connector in Azure Logic App. We came to the conclusion that is not possible to add data to a MySQL database hosted on Azure from Logic App, this feature is not working. It requires to write a supplementary Azure Function to do so. The connector only supports the connection to the databases hosted on an on-premise through an on-premise data gateway.

As this feature is not available, my supervisor told me that he will think more about the task and maybe find a solution outside the box with Cosmos DB. I don't know what to expect, but for now the Logic Apps task is going to be paused for some time.

Tuesday

Today I have been asked to create a report with different previews in Microsoft Power BI. I will first need to insert mock data in the MySQL database based on what my supervisor sent me already. The visualizations should show relevant information about these temperature recordings.

I started by downloading Microsoft BI and installing it, then I imported data by connecting to the MySQL database. Now that all the data are located in one table, I can use for them my different charts. Here I was quite free to realize any kind of report to show the different information. I created a page that will show the minimum and maximum of the temperature recorded for a period of time and based on the localization of the water counter. The recordings come from several places so that is interesting to distinguish all of them.

The second page shows more general information about how the temperature evolve with a graph chart that shows all the recordings of the month. The user can choose the year and the location of the recordings. That's all for it for now, I have to wait for my supervisor

feedback to see if there is enough information and if it is needed to present data in a more specific way.

I'm not able to work on the Logic Apps for now, my supervisor was looking for an alternative of the MySQL database. Maybe we are going to use a Cosmos DB solution to connect with the Logic App, but it is not sure for the moment. This day has been mainly about the presentation of the different data of the database in Power BI and finding the best overview to make. I was already familiar with the tool, so I didn't face any problem during the whole task.

Wednesday

Today's goal is to review the Power BI dashboard made yesterday with the feedback of my supervisor. I'll also add more data in the database so it will be more convenient to test the dashboard.

I didn't understand the nature of the data at the first glance, but my supervisor explained to me that the data are the consumption of water from specific locations. I thought it was the temperature of the water, so the data I displayed on the dashboard are no longer making sense now. In addition, an address was not pointing exactly at the right place on the map, but in a different city with the same street name.

I first started to reorganize the dashboard to have the total of cold and warm water consumption by address and time. On the second page, I renamed the graph to "Water consumption" instead of "Evolution of the temperature". The data are now making more sense with this rearrangement.

Next step I brought some modification in the data, I added a column in the database containing the city of the address. Then I added another column but in Power BI just to concatenate "Address", "ZipCode" and "City". E.g. "Junailijankuja, 00520 Helsinki". Now the map is able to indicate with more precision the exact emplacement of the recording. I also added the total of warm and cold water on the map, so it displayed circles more or less large depending on the consumption :

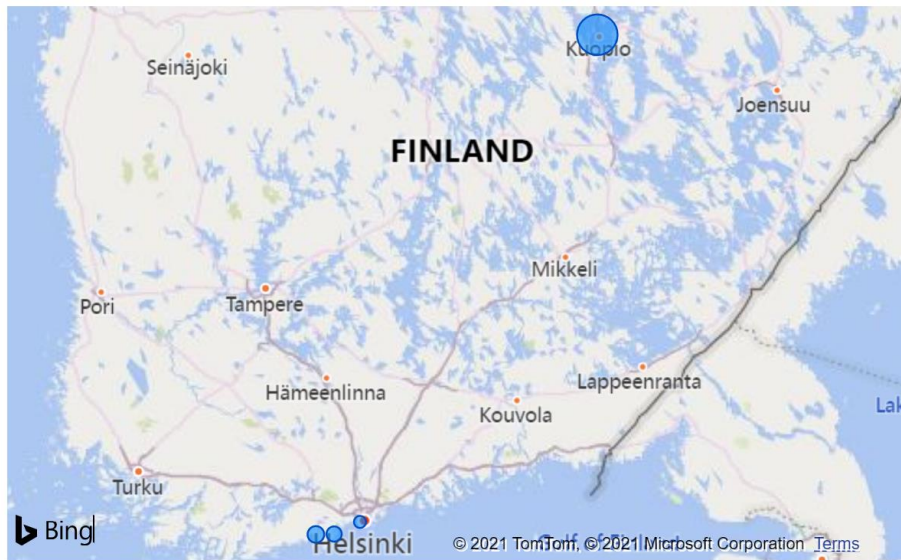


Figure 13 : Location of water consumption (mock data)

Tuesday

For today, I have been assigned to a new task to study. It is about an automation tool called Ansible that I will have to read up on the topic and set up a small testing environment. The aim of the task is to be able to update the version of a MySQL or PostgreSQL database server with the tool and also check if the actions are reversible or not.

I started by looking to an older version of MySQL servers to install, after a few tries to download and installing the database, I kept facing an error during the installation process. I tried different ways to solve the problem by removing existing installations, reinstalling launchers, installing package for Visual Studio but nothing worked. I tried with the newest version and it was the only one that was working properly. As I wanted an older version of a database server, I decided to go for PostgreSQL version 11.11-1, here I faced no error during the installation, and I was able to have my server up and running in a short among time.

Next step is about Ansible. To introduce it a little bit, it is an open-source software that allows configuration management, application deployment, updates on workstations and servers. It is a powerful automation tools for system administration and DevOps, without depending on agent software. The machine with Ansible is called the "Control Node" and the other device that will receive the instructions are named "Managed Nodes". Here a schema:

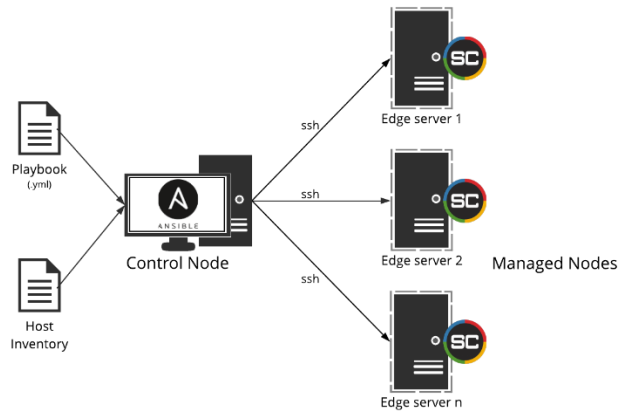


Figure 14 : Ansible schema (Shimmercat.com, 2020)

The control node can send instruction through a “Playbook”, which is a YAML file containing all the instructions that the managed nodes have to execute. The “Host inventory” file contains the structure of the network with hostnames or IPs of the servers.

Then I decided to follow a simple tutorial (Cloud Advocate, 2020) to set up an easy Ansible for Windows hosts. As Ansible doesn’t run on Windows, I add to turn on the feature “Windows Subsystem for Linux” and installed Ubuntu terminal from the Microsoft store. The first requirement was to install “pywinrm”, it allows the control node to connect to the managed nodes that run under Windows. On the hosts Ansible requires minimum “PowerShell 3.0” and “.Net 4.0” installed and a WinRM listener activated.

After taking care of configuring and installing all the requirements, I installed Ansible on the Ubuntu subsystem, then I created a new user on the Windows host that would be needed to execute the instruction for later. Next steps are to add in the inventory our Windows host IP address and finally create a simple playbook that will install the package “Chocolatey” on the managed host. Now I can run the command “ansible-playbook chocolatey.yaml” and Ansible will be installing the package management tool on my local computer. It worked at the first time which was surprisingly satisfying.

At the end of the day, I was quite happy with what I achieved, even if at the first glance I thought that the task will have been done more quickly, I underestimated it. I had no idea what was Ansible at the start of the day and now I have a better understanding of the tool in general, but I need more practice with it. The configuration took some time as well, because I was not sure how to write and what information to provide, hopefully the tutorial helped me.

Friday

Today I'll keep on doing with Ansible task, I'll try to come up with a working set up to configure the PostgreSQL database I created yesterday. I'll also have a meeting with Eleonora to review *Microsym*'s documents related to project management, that will take place during most of the morning.

We started to discuss about the documents, and she presented a report that she was working on. Its aim is to have precise information on how the project went, how the communication was, what didn't work and what kind of improvement that could be made for the next project. There is already a "Project and report", but this one is more general, and it is shared with the customer. Here the idea is to have a more personal review of the project with the team, in order to improve the teamwork and working process.

The process to start a project is to create it in Jira, but as Eleonora describes the tool, it is not very efficient for project management and that's why it is not often use. Most small projects are not even created in Jira tools and are handled just by someone. It is quite hard for me to bring a good insight here as I don't have enough practical experience in general but also experience in *Microsym* projects (management side). However, as we discussed I'll try to come up with new ideas or change for "Kick-off meeting", "project plan" and "debrief report".

On the afternoon we had our weekly meeting, where Saku presented his human detection program and it is now able to count and track the people walking around on a video. Then I introduced them to Ansible and made a small demo of my set-up, I also presented the Logic Apps with the POST request and the Power Bi report I made. The feedback was good so nothing to change on the different tasks I did.

Next step would have been working on the Ansible task, but I don't have enough time to work on it today. However, I went through a few documentations where I learned more about the Ansible Galaxy and how to use it. It is basically a repository for Ansible roles created by the community, so it is not needed to rewrite them from scratch. It can be compared to the NPM package manager for Node.js. It is possible to install them through the Ansible command-line tool and use them in the playbooks.

Weekly Analysis

The week went well with various kinds of tasks in different fields, but it was interesting, and I learned new useful skills. I also have a better understanding on the Azure DevOps migration for the project management side, it will provide a better track of progress and the remaining work. With the actual way to manage the project, the information is spread in different tools and it makes harder for the PM and the team member to have an efficient working environment. It will even more integrated when Microsoft Teams and Azure DevOps will be connected through the bot configured and the python script, the information is going to be more centralized and more easily accessible for all team members. Also, with the actual PM tool, it doesn't provide more information on how the project is going, that's another advantage of Azure Dashboard to be able to display customized and useful analytics about the project.

In the middle of the week, I faced a small problem with MySQL server, I was not able to install an older version of the database. I've been searching if it was a current problem, but it seems that was not the case. I tried many ways to install, but at the end, only the newest version was working. I was not interested in using it as I needed an older version. I decided to not lose more time and that and go for PostgreSQL. Here the choice of the database is not very important as it is only a testing environment, but it will be interesting to know which one will suit the best depending on the requirement.

PostgreSQL offer the usual feature of relational databases plus some unique like inheritance for object-oriented readers, customized data types and it supports geometrical data types such as points, line segments, box, polygons and circles. In general PostgreSQL is highly extensible, it allows to create new functions, new operators and new data types in any language (Douglas & Douglas, 2009). It is a complete open source and community-driven database and mostly used for large systems where reading and writing speeds are necessary. It performs well when executing complex queries and it is following SQL standards.

On the other hand, MySQL is more described as a light, fast and easy relational database to use and more popular than its competitor. Which has the advantage of having more third-party tools or developers /database administrator available (Krasimir Hristozov, 2019). MySQL's source code is available under the GNU general public license and it is owned and maintained by Oracle Corporation. It is essentially used for web projects that need a database for straightforward data transactions and work well in OLTP and OLAP systems when only reading speed is necessary. Examples of important companies using MySQL: Airbnb; Uber; Twitter. For PostgreSQL there are: Netflix; Instagram; Group (Guru99.com, 2021).

To come back on the Azure Logic App and the MySQL connector, we decided not to dig more on the Azure Functions, firstly because it complexifies the process a little bit, but it is still doable. Secondly, because it had an additional cost for the whole task. Each time the process is triggered it would be required to pay for both of them instead of only the Logic App. In this context it won't be an excessive cost as it is only for testing purpose without a lot of requests, but it can become more considerable when it is used on a larger scale.

Along my research about Ansible, I have been confronted a few times with some of its competitors such as "Chef", "Puppet" and "Saltstack". The main difference between Ansible and the other tools is that it doesn't require any agent installation on the client servers. Which is called a native agentless model that allows to save time and makes the configuration and maintenance easier (Ali Raza, 2016). Puppet and Chef use pull configurations, which mean that the agent on the slave node is going to automatically pull all the configuration located on the master server without running any commands. On the other hand, Ansible and Saltstack follow push configuration, where all the configuration present on the central server will be pushed to the nodes. All of these tools use different languages configuration:

- **Chef:** It uses Ruby DSL, which stands for Domain Specific Language
- **Ansible:** It uses YAML
- **Puppet:** It uses its own language, Puppet DSL
- **Saltstack:** It also uses YAML

Chef and Puppet require more programming skills, it makes them respectively developers and system administrators oriented. The advantage of Ansible and Saltstack is to be simpler to program with the YAML language, as described earlier in this thesis, it is a simple and human readable language. This is probably because Puppet and Chef are older than the two other tools. All of them support Linux /Unix and Windows clients but require a Linux / Unix master server. In conclusion, all of them have their own advantages and disadvantages and aim at a different user segments within the same target market. However, Ansible is described as a promising tool with a growing trends (Aayushi Johari, 2019).

Then I also wanted to search for more information about project debriefing. In my opinion, continuous process improvement is very important, especially in the IT world where everything goes fast and can change very suddenly. The debrief session usually takes place whenever the project end or all of the project activities are completed. It allows to have a non-stressful environment and a transparent feedback from the participants. It is a good time to share with each other the experiences, opinion and solutions in a more relaxed climate and helps in identifying issues that may have happened during the project. That will

automatically assist project managers to isolate conflicts or problems, evaluate the team's cohesiveness and improve the team dynamic for future projects. The debrief process should cover the project objective if they were met or not, what problems the participants faced and why, and what things need to be done differently. To keep the debrief constructive, it is needed to avoid blaming or evoking personal conflicts or arguments, it has to be handled behind closed doors. In general, the meeting should not cover any personal gripes and stay constructive in a project and business sense. To sum up debrief session can allow to bring different opportunities or contribute to continuous improvement in the company. It will also create a more high-functioning, strong and united team with a better communication and interaction. Even if it is time consuming and the reward won't come up immediately, its benefit and lesson learned will create more value on the long term, which makes a good argument to do projects debrief (Moirra Alexander, 2016).

3.6 Week 6 (15.03.2021 – 19.03.2021)

Week's goals are to manage the PostgreSQL database version with Ansible and check for rollback possibilities using the tool. There is a meeting planned on Tuesday with my colleagues about the project management review. I'll also take some time to write chapters 1 and 2 of the Thesis.

Monday

To start the week, I decided to dedicate this day to my thesis for the chapters 1 and 2. I was able to complete a good part of it, but I'll take more time to work on it this week.

I also made some research about the possibilities of reversing action with Ansible playbook. To sum up, there are no functionalities built for this kind of task. The only way to reverse the change brought is to write a playbook that actually reverse the actions. However, Ansible offer some functionality to handle failures manually. This is used with "blocks", it allows to define a set of tasks to be executed in the "rescue:" section.

Tuesday

This morning I had a meeting with my supervisor about a new project ongoing. It is about the integration of the pre-sale process in Azure DevOps. That will allow to have a better track on how the pre-sale process goes with different analytics integrated. The case of the pre-sale will be represented by the work item type "Epics", the different stages by the "Features", the general topic by the "User stories" and the actual actions to do, by the "Tasks". Now that I have more information about it, I have time to think about and we will have another meeting on Thursday.

On the afternoon I had a meeting with Eleonora and Tapio, the company's project managers. We have been through the different topics we talk about last time. A point that came up was about the all the documentation contained in Zipper, there are so many documents that it is usually hard to search and see important content and especially related to the project management. There will be a new place on the SharePoint dedicated for the project management document and process.

To sum up the meeting, the project management process has to start earlier in the sale's process. The information and all documentation about projects arrive too late in the hand of the project manager and make it hard to start, especially as the kick-off meeting starts in the same time. There will be a new working process for the process management and a

presentation about it in the following week. Also, the debrief for the project will integrate the customer feedback, in addition with the team's feedback that will help to improve for future projects.

Wednesday

Today will be dedicated to the Ansible task, I'll try come up with the initial task up and running and I'll probably do some testing as well with the managed node.

I found a module for the Postgres server on the Ansible documentation, that will allow my playbook to use operation on my managed node such as creating a database, create user and so on. However, there were some requirements on the Windows host to use the module, it required "pyscopg2" library for python. I really struggle to install it and I went almost out of ideas to make it worked. After some time looking on Ansible documentation, I found out that WinRM was not supporting Python modules and it only works with an equivalent Windows module (Ansible.com, 2021). The module I was trying to use, had no equivalent for Windows.

I decided to completely change my testing environment and I created a virtual machine on Ubuntu with VirtualBox. To access it locally I had to add another network adapter on "host-only", the first one was on "NAT" so I can still access internet for the installation. Then I followed a tutorial (Ashraf Sharif, 2020) that guide me through, the SSH connection setup, using an Ansible Role for MySQL and creating the Ansible Playbook to deploy it on the managed node. It worked properly and I have now the server up and running on my virtual machine.

This day was long and tedious, I spend a lot of time trying to fix the "pyscopg2" installation and on the configuration of the virtual machine. Hopefully, the Ansible tutorial was really helpful, and it guides me through the different steps, I was able to understand much better all the functioning and especially about the Ansible Roles.

Thursday

For today, I have a meeting planned with my supervisor about the Presale project in Azure DevOps. We will go through the project and explain to me what kind of task will be required. Then I'll continue my work of yesterday about the Ansible task.

Eventually the meeting was postponed for another week, but instead we chatted on Teams for a new task related to the project. The aim here is to provide various analytics about the whole Presale process in the Azure Dashboard. I was requested to create a dashboard that

will show the average number of days to complete the different stages of a case and the number of tasks with state “closed” and “active”.

I started by figuring out a way to retrieve all the Tasks related to an Epic in order to use them in a chart. The query has to be a hierarchy, but I discovered the charts in Azure Dashboard don't work with this kind of query. The second option I found was using a tag on each task, that will indicate the name of the Epics (case) and the Features (stage), so the query retrieves proper data. Now I can create a chart showing all the tasks open and closed for a project case. The second visual was simpler, because it exists a widget called “Cycle Time” that will display the average day for a work item to achieve the state “closed”. I created one for each stage, so at the end it will provide a good metric on the process evaluation.

The rest of the day, I have been carrying my testing with Ansible. I set up an Ansible Vault to encrypt all the sensitive information of the scripts. Basically, a vault is a YAML file containing key-value pair information that can be retrieved in the Playbook. When it is created, it asks for a password that will be then use for the encryption method. Here is the file when I tried to open it with nano command:

```
$ANSIBLE_VAULT;1.1;AES256
36333861353839666330356239353963356263353731653631636530353933636230343337653238
3731393738303639333331383761366637373535663037300a616134323338333231326639396161
64353966613932323362326237623835303331653163313266373465393763373634363637623963
6532393563343433380a336664333939383734323036303964636533393238386364626338303736
62623736346537356231313533346263303033376237653934646264626566386662323763386636
65383832356662343332626233333535363733313331376234356537306336656462376130613934
66356539323433383530643931366232653331633865353136373939646432306464643564666234
32386162336133653431356138356638643537616461303565303138613538383432393735613562
3765
```

Figure 15 : YAML file encrypted with Ansible Vault

To use the different values in the playbook or the other file of the role, the key has to be specified, “{{ example_password }}” and a parameter is required when the command is run.

```
> ansible-playbook mysql-playbook.yml --ask-vault-pass -K
```

Then I search more information about the Ansible roles. I created a new role for the MySQL task with the Ansible advised structure and create the different tasks that the playbook will execute when using this role. The tasks will create a database on the MySQL server with a table and then fill it with some data. I executed the playbook and the operation went well and the database was available with the data.

Friday

For today I'll prepare a small demonstration for our weekly meeting using the Ansible Roles and Ansible Vault. Also, I received a new task to work on. It is to code a Python script that is able to clone all repositories from Bitbucket locally on the computer. The script is already

existing, but it misses the cloning part. The Bitbucket repositories won't be any longer used because of the migration to Azure DevOps, all new projects and their repositories are created here. However, it is needed to have a backup of all of the old project code in Bitbucket just in case of special needs.

I started by running the actual script and see the output. All the repositories from a user are shown in the terminal. The first step that I wanted to improve is not using hard-coded information. The user's credentials were simple variables with the value clear in the code, so I installed the library "dotenv", created the ".env" file and filled it with all the sensitive information.

The second step is to clone the repositories locally. Here I found out the library "git" for this kind of operation. It requires the URL of the Bitbucket repository and the path where it will be cloned. When the request is made, a pop-up page for the authentication appears and asks for the user's password. Then, all repositories are cloned in the same level as the script in a folder for the team and then each project has its own folder.

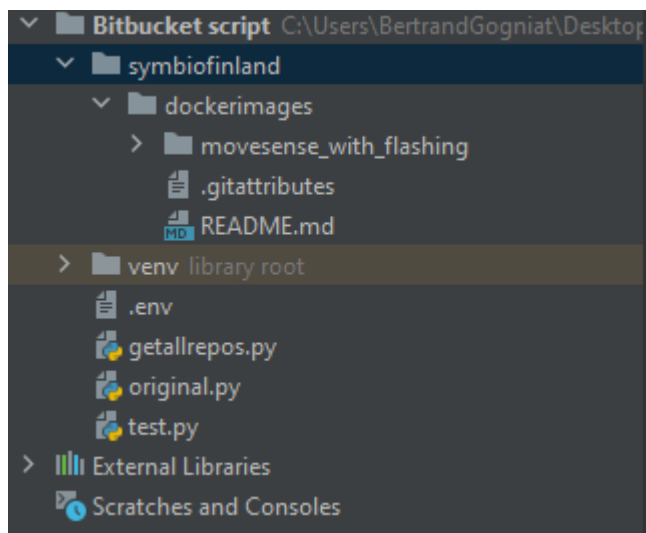


Figure 16 : Repository "dockerimages" cloned from Bitbucket

Weekly analysis

Overall, this week was mostly focusing on the Ansible task, where I was able to test a Role already made and then made one myself. My role was able to create a database and fill it with some mock data. Ansible Vault was also an important point in terms of security, and I was able to test it as well. It allows to encrypt values and data structure such as passwords and private keys within Ansible projects. Sensitive data in vault-encrypted file are automatically decrypted at runtime when the key I provided. Even though it is such a good practice, another side to consider is also, how secure Ansible Vaults can be?

It is scary a bit to push files containing sensitive information on a public or semi-public Git repository, here we rely only on encryption and passwords instead of other kinds of security practice. Ansible Vault requires a password for the encryption key, but it is not directly use as it. It synthesizes a key from the entry using PBKDF2, which is a cryptographic key derivation function to make harder for someone to determine the master password by making repeated tries in a brute-force attacks (Wikipedia.org, 2021). So, after generating a 256-bit key from the password entered by the user, Ansible uses another kind of encryption known as AES, which is a symmetric block cipher that was chosen by the U.S. government to protect classified information.

To answer the question, “How secure Ansible Vaults can be?”, the password given at the beginning needs to be sufficiently long and random. A 32 characters randomly generated password should suffice for the long run (Gabriel Birke, 2017). Along with my research I faced another case scenario, where maybe users want to use multiple passwords to be referenced inside a playbook. Vault ID, appeared after Ansible 2.4 version release, allows this scenario, but before, only one vault password could be used in each playbook run (Ashraf Sharif, 2020).

Another thing I wanted to clarify about Ansible is the Roles and its directory structure. Roles allow to ease sharing code among teams and publicly with Ansible Galaxy. All the configurations and tasks are contained in a specific directory related to its use. Each folder has to contain a “main.yml” file which is filled with important content. Here the folder structure described (Medium.com, 2018):

1. **defaults:** contains default variables for the role, these have the lowest priority of any variables available and can be overridden by any other, including inventory ones.
2. **vars:** other variables for the role and have a higher priority than the default directory
3. **tasks:** contains the instructions or commands to be executed by the role
4. **files:** all files that needs to be copied on the managed node are located here. It is not required to specify a path for resources stored in this directory.

5. **templates**: templates that the role deploys, and it supports modifications.
6. **meta**: metadata for the role, such as its dependencies, support platforms and authors
7. **handlers**: contains handlers, which are tasks that can be flagged when a playbook is running in order to be executed at the end of it.

During my testing phase, I spend a lot of time trying to configure my environment, the “psycpg2” library was not getting installed properly in order to be used by the Ansible playbook. Here the real problem I faced is that I was trying to use the Python Module on a Windows host and as described earlier in the thesis it is not something possible. I should have been more focused on the documentation and the requirements for Windows and Linux hosts. At the end I decided to go for a full-Linux environment, because of the modules I was using and also because there are more tutorials and documentation available with this kind of environment.

The other side task about the dashboard and the pre-sale process was also interesting, it is going to be adapted in Azure DevOps in order to monitor and having more control on the process. The metrics defined for now are going to be the average of days to finish a stage and the actual state of a stage, it’s probably going to change and evolve in the following week. When I was looking for a widget for the dashboard, I had two options to measure the average time to finish a work item “Lead Time” and the “Cycle Time”. After some research on the Microsoft documentation, I found out the difference between the two:

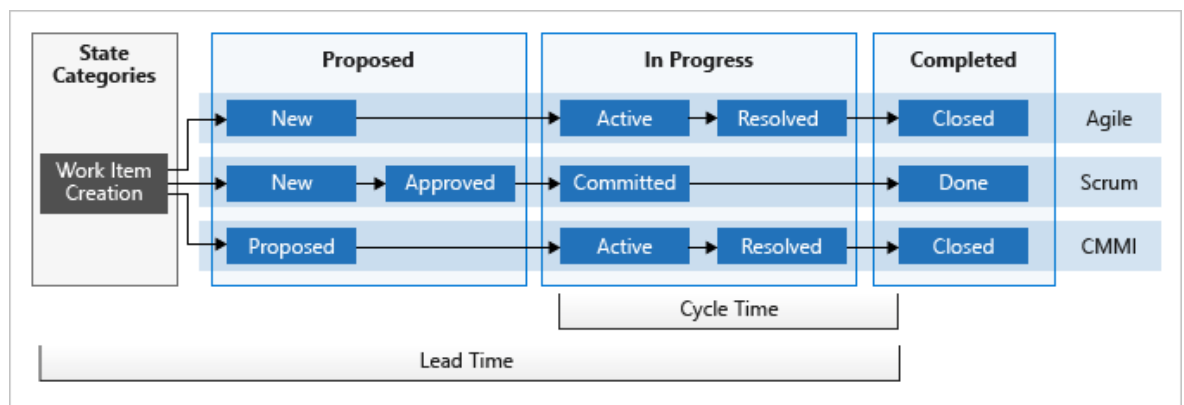


Figure 17 : Azure DevOps, Lead Time vs. Cycle Time (Microsoft.com, 2020e)

Basically, the Lead Time cover the creation, “New” state, until the “Closed” state, where the Cycle Time cover the time elapsed during the “Active” and “Closed” state. In my case the second option was the most appropriate for the analytics to know the completion time of a stage. The actual process for the project in Azure DevOps is configured in Agile, so we have the following work items hierarchy: Epic, Feature, User Story, Task and Bug. Each new case for the pre-sale process will be reflected by an Epic, each stage by a Feature and then each User story will have one or multiple tasks related to it.

This adaptation of the process can be related to Business Management Process (BPM) discipline, which aims to discover, model, analyse, measure, improve, optimize and automate business processes. It is discipline that aims to improve processes over time and standardize them in an organization. It also tends to eliminate errors made by human and process gaps while trying to improving performance and conformity (HappyFox.com, 2021). The pre-sale process is not completely defined yet and will probably change before being totally adapted in Azure DevOps, also more metrics will have to be defined in the future.

3.7 Week 7 (23.03.2021 – 26.03.2021)

This week objective is to discover Ansible Tower and give it a try and get replicate what has been done last week with the new tool. I'll keep on updating my personal documentation about my configurations done for the Ansible tests and also fill the Python script with more comments and probably create a "READ ME" file.

Monday

No entry.

Tuesday

Yesterday I received a new task from my supervisor, so I started working on it today. It is about creating a Windows Forms and fill it with my premade questions for a survey about lead qualification. I'll also plan to work more on chapters 1 and 2 of my Thesis and start to write the abstract.

The creation is pretty straight forward, Windows Form is part of Microsoft 365 and it was quite fast to end the whole form with the 26 questions. I change the default color for the *Microsym's* main color, and I send it for feedback to my supervisor. He wanted some default answer for certain questions, I did some research, and it was not possible with the actual version. It requires the pro-version of the application, which has been recently renamed as "Dynamic 365 Customer voice". Customer voice is a solution for customer feedback management, it provides effective solutions to build and analyse customer surveys and identify metrics they need to improve the experience of their customers. This version has more advanced and complex features than Windows Form, but for this task the basic version was fitting the needs already.

Wednesday

Today I'll spend more time on testing Ansible Tower and set up the environment required. If I have enough time, I'll start testing different roles I created previously and adapt them for this task.

I started by checking on RedHat website how to begin with Ansible Tower. Firstly, I had to create an account in order to get the trial version. Then I received an email for the installation process and the documentation. Here are the different requirements for the installation:

Supported OS:

- Red Hat Enterprise Linux 8.2 or later 64-bit (x86)
- Red Hat Enterprise Linux 7.7 or later 64-bit (x86)

- CentOS 7.7 or later 64-bit (x86)

Hardware requirement:

- Minimum 2 CPUs
- Minimum 4GB RAM on Ansible Tower host
- Recommended: 4GB of memory for each 100 forks (Managed by fork configuration parameters in job templates or system configuration)
- 150 GB VM storage, 10 GB for /var, additional 20 GB for database

Secondly, I set up a virtual machine with Virtual Box on CentOS 7 and configure all the requirements for Ansible Tower. Then I downloaded the zip file locally and unzip it in the document folder. Inside, I had to set up the password for the Ansible Tower admin account (amdin_password), PostgreSQL database user account (pg_password), and Automation Hub user account (automationhub_pg_password) in the inventory file. When it was all ready, I used the command “./setup.sh” to start the installation. The installation is performed by executing Ansible Playbook. Current installation program automatically installs Ansible Engine and any dependencies not already present. Latest stable version of Ansible Engine must be installed on installation server using distribution’s package manager.

Now I was able to get access the Ansible web GUI by typing my IP address or Ansible Tower server name in the URL of the web browser. The interface contains a lot of information and seems quite complex at the first glance. I decided to read up more sources and watch tutorials before starting to test it.

Thursday

Today, I’ll continue to work on Ansible Tower, now that I have the environment functioning, I’ll be able to configure and use my role and playbook made previously. Also, as I saw in yesterday tutorials and documentation, it is possible to directly use a Git repository as a project, so I’ll try to use this functionality as well.

I began by creating a new repository in my GitHub account, where I added my playbook and the two roles for the MySQL database and data. Then I added trouble with my Ubuntu virtual machine, there was not enough disk space. After enlarging the disk, the VM was not working anymore, and I don’t know exactly why but might be related to the change brought the disk. I created a new CentOS 8 VM, added a network adapter so I can access it from the other hosts.

On the host node, I tried first the SSH connection and it was working properly. I also noticed that it is possible to access Ansible Tower from my laptop, but for that I added a network

adapter configure on “Host-only” so now I’m able to access it from my web browser. Ansible Tower works a little differently, there are new concepts to understand a little bit:

- **Templates:** It is the environment containing all the configurations (Credentials, Inventory, Project) and will be run by the agent.
- **Credentials:** It contains the information for different types of connection or password, for SSH connection or Ansible Vault.
- **Project:** That’s where the playbook and roles are contained. It is also possible to get a project from GitHub or other version control system or even imported manually.
- **Inventories:** As view previously, it contains all the managed node IP or hostname.

Here is my template for the testing of MySQL:

The screenshot shows the 'Test MySQL' job template configuration in Ansible Tower. The interface includes several tabs: DETAILS (selected), PERMISSIONS, NOTIFICATIONS, COMPLETED JOBS, SCHEDULES, and ADD SURVEY. The configuration fields are as follows:

- NAME:** Test MySQL
- DESCRIPTION:** (Empty)
- JOB TYPE:** Run
- INVENTORY:** DB server
- PROMPT ON LAUNCH:**
- PROJECT:** MySQL deploy and database
- PLAYBOOK:** mysql-playbook.yml
- CREDENTIALS:** MySQL, MySQL_Vault
- PROMPT ON LAUNCH:**
- FORKS:** 0
- PROMPT ON LAUNCH:**
- VERBOSITY:** 0 (Normal)
- PROMPT ON LAUNCH:**
- JOB TAGS:** (Empty)
- PROMPT ON LAUNCH:**
- INSTANCE GROUPS:** (Empty)
- SKIP TAGS:** (Empty)
- PROMPT ON LAUNCH:**
- SHOW CHANGES:**
- PROMPT ON LAUNCH:**
- OPTIONS:**
 - ENABLE PRIVILEGE ESCALATION
 - ENABLE PROVISIONING CALLBACKS
 - ENABLE WEBHOOK
 - ENABLE CONCURRENT JOBS
 - ENABLE FACT CACHE

Figure 18 : Ansible Tower template for MySQL

During my testing I noticed a small issue when I was modifying and pushing modification on the GitHub repository. The changes were not automatically updated for my Project, that cause many fails when I was running the template. To fix it, I added to manually refresh the project, by clicking on the button “Get latest SCM version”. Now the testing environment works properly, and I was able to access the MySQL database and data in the managed node.

Friday

After some discussion with my supervisor, he asked me if I could have a look on the how to update MySQL and roll back the action with Ansible Tower. So, the aim of today is to make a role that is able to do that, so I can present it during our weekly meeting.

I started doing some research on how to update MySQL with Ansible and from the official documentation, there is no module for that specific task to update an existing MySQL database. Then, I carry on looking for more documentation form MariaDB on how to manually update it on CentOS. I experience the instructions on the managed node and I

was able to pass from versions 10.3 to 10.5. Here I was fairly confused either I should call the database Mariadb or MySQL. On Ubuntu I was using the command “mysql” to connect and here on CentOS I have to use “mariadb”. Now that the update worked as intended, I replicated the whole process in a role, the tasks are simple instructions that execute the commands. Here is the YAML file to uninstall the older version:

```
---
#Uninstall the old version
- name: Stop MariaDB server process
  command: systemctl stop mariadb
  ignore_errors: yes
- name: Uninstall via YUM
  command: yum remove "MariaDB-*" -y
  ignore_errors: yes
- name: Uninstall Galera package as well
  command: yum remove galera-4 -y
  ignore_errors: yes
```

Figure 19 : Tasks to uninstall MariaDB and Galera packages

This role is now pushed on my GitHub repository and I’m able to use it in Ansible Tower. I noticed that there is an option pretty useful when working with a SCM, it is to enable the option “Update revision on launch”. Each time the job is running, Ansible Tower will check for the latest version available on GitHub.

On the afternoon we stand the weekly meeting, where Ville present the program is working on a program to control a robot arm. Then, I gave a small introduction on Ansible Tower interface and the concepts related to it such as Projects, Templates and Credentials. I also explained my workflow with GitHub repository. Later on, I spent the rest of the day writing a “READ ME” file in markdown format, explaining my search on Ansible, the concepts related to it, how I set up my testing environment and more general information with some reference to official documentation. This document will allow another person to quickly understand Ansible and set up a simple project quickly.

Weekly analysis

This week has been a little less busy than the others, I missed Monday entry because I was on a day off, but I did some work on the chapters 1 and 2 of the Thesis. Overall, I've been mostly occupied with Ansible Tower where I set up an environment for testing the tool. I was able to experiment with CentOS, which I had never used this kind of operating system before. With the GUI it was simpler and looked the same as Ubuntu interface. I have just noticed differences for some commands for example "YUM". It is the default package manager for CentOS, while "APT" is used in Ubuntu.

When I installed the CentOS 7 on the virtual machine, there was no user interface and I didn't want to lose more time on the configuration, so I decided to install the GNOME desktop and the problem was solved pretty quickly. However, along my search, I have been confronted with other kinds of interfaces such as "KDE", "XCFE" and "Mate". I choose GNOME because I heard about it already with other Linux distros, but for the others I didn't know much about. Gnome is described as a desktop environment that is more user-friendly and accessible, which tends to be simpler and more stable. KDE is a desktop environment that allows users to have more basic applications and functions for daily work without relying on the command line. KDE and GNOME are much similar with Windows except they are related to Linux through servers rather than operation system, but KDE consumes more resources than GNOME (Lithmee, 2018).

XCFE is a light desktop environment, which aims to be fast and low on system resources while still being visually appealing and user-friendly. Its lightweight construction conserves both memory and CPU cycles. XFCE has a very small memory footprint and CPU usage compared to some other desktops, such as KDE and GNOME (Ivan Vanney, 2020). On the other hand, MATE desktop is the continuation of GNOME 2 and his part of Linux Mint family also with Cinnamon. It aims to operate with clear features, to be more accessible and to have modest looking, but highly customizable (Rabiya Younas, 2021).

Overall in my case, if it was for a real project, it would have been more interesting to use either none interface and configure the server through the command line or install XFCE as it is a very lightweight solution compared to the others. Also, the user interface was not necessary after the configuration, because when Ansible Tower is up and running the VM is not directly used anymore and can be accessed from any other device of the network over the web browser. However, that small research on the topic helped me to understand a little more on the topic and when it is better to use a specific desktop environment.

Along with my testing and searching on Ansible Tower, my supervisor talk to me about “Infrastructure as Code”, I partially understand it because it is part of DevOps topics, but I didn’t how to properly define it. So here I decided to deep a little bit more into that subject to have a better understanding of the meaning and how it is related to Ansible. Infrastructure as code aims to manage and provisioning IT infrastructure (VMs, networks, connection topology and load balancer) in detailed models for example through configuration files. It emphasizes consistent, repeatable routines for provisioning and changing systems and their configuration (Morris, 2020). The benefit that IaC tries to achieve include rapid delivery of value through IT infrastructure, reduce effort and risk of making changes of infrastructure, providing common tooling across the team, creating reliable secure and cost-effective systems. Here we can already see how Ansible is related to these concepts of automation and configuration of the infrastructure. That allows the team to avoid all kinds of manual configuration on the infrastructure and ensure consistency by specifying the required state of the environment through code (Sam Guckenheimer, 2017).

DevOps apply the same principle but on the source code. The same binary is generated every time from the same source code, so each IaC model will generate the exact same environment each time it is executed. Infrastructure as Code is an essential DevOps practice and it often used associated with continuous delivery (Sam Guckenheimer, 2017). That will allows DevOps teams to work more efficiency with a collection of unified tools and practices in order to deliver solutions with a supporting infrastructure highly reliable, scalable and fast.

Still related to Ansible, during the creation of the roles to update the database server, I left out an important part of the updating process, the backup. In the tutorial that I followed, it was recommended to do backup of the database before starting the update process, which I didn’t replicate in the Ansible role I created. Even if there are not a lot of risks, it is always a good practice to have for this kind of operation. In my case that was only for testing purpose, but for the project with real data, that would have been a serious point to take into account. Also, as I described previously in the Thesis, there is no rollback option with Ansible. The only way to reverse the operation made would have been to reinstall the older version of MariaDB through another playbook especially for that.

3.8 Week 8 (29.03.2021 – 02.04.2021)

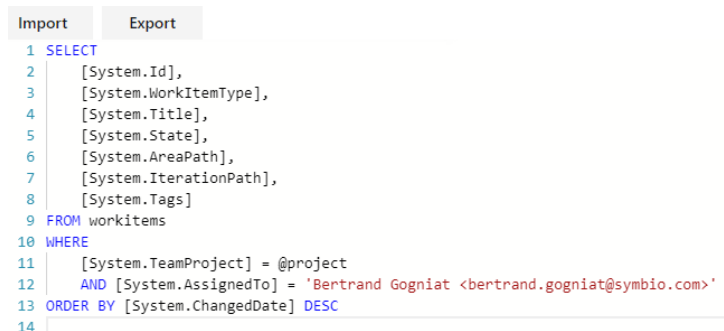
This week's objectives are not completely defined for now, because it will depend on the task I receive. However, from last Friday my supervisor told me about some work on the Logic App and project related to it. This week I have two meetings about the project management process that will take place and also about the presale project.

Monday

This morning I received a new task about Azure Board and the presale project. My supervisor asked me to find a way to display the current stage of a case in the Dashboard in order to keep track of the progress of the process. Here I started by having a look on the different widgets of the Dashboard in Azure DevOps. I didn't find any that will fit for that, so I decided to create a query that retrieves the active feature with the tag of the case's name. Now I can just display the result on the dashboard so we can see the current state of the stage.

I also researched about the work items state, as I noticed it is not updating automatically. For example, if a Feature have all its children with the state "Closed" the Feature won't change to "Closed" state. All the state changes have to be done manually when working with the work item. On a forum I found this information, people were disappointed that there is not this functionality and propose for example to create an Azure Logic App or install a third-party extension to automatically do the job. Also, that could be a task done by a script, let's see tomorrow what my supervisor thinks about it.

I spent the rest of the day testing the Azure REST API to create a query through a script. It is a POST request that requires a JSON body containing the name and the WIQL text for the query. As described previously in the report, the WIQL is very similar to the SQL request. However, to facilitate the creation of it, I installed an extension on my Azure DevOps sandbox project that allows to visualize the WIQL request of a query created through the GUI. Here is an example:



```
1 SELECT
2 [System.Id],
3 [System.WorkItemType],
4 [System.Title],
5 [System.State],
6 [System.AreaPath],
7 [System.IterationPath],
8 [System.Tags],
9 FROM workitems
10 WHERE
11 [System.TeamProject] = @project
12 AND [System.AssignedTo] = 'Bertrand Gogniat <bertrand.gogniat@symbio.com>'
13 ORDER BY [System.ChangedDate] DESC
14
```

Figure 20 : WIQL request that retrieves work items assigned to a user

Now I was able to use it in the script and create a new query with the POST request. I also tested to create a folder in the path and move the query just created inside that folder. It was pretty easy to do as the documentation is straight forward. However, I faced a small problem when using JSON structure for a Python variable. The Python object has to be converted with “`json.dumps(var)`”, but then the POST request was not working because the conversion added double quote instead of just single that was required for the request. I used a library named “`ast`” and I was able to fix the issue with adding that instruction to the conversion “`ast.literal_eval(json.dumps(var))`”.

Tuesday

Today I have a meeting about project management planned and I will continue on working with the dashboard to improve the analytics of the presale project.

During the meeting we have been talking about a new SharePoint page that will regroup all information concerning project management such as a template, project ongoing, lesson learned from past activities and processes described. The page doesn't have a defined structure at the moment, but that something we will work on for the following weeks. Otherwise, my colleagues explained to me how their ongoing project was doing and sharing information related to Azure DevOps. I learned that it is possible to add a column to show the completion of the Epics directly in the backlog. This is a roll-up column with “Progress by all Work Items”, that allows to see the percentage of Epics' progress.

Also, Eleonora came up with an interesting question about a User Story and Tasks that have the same Epic. All work items can only have one parent, so here the option was to duplicate the work items and manages both of them at the same time. I did some research and I find out that it is possible to add a link on a work item for “Related work”, but then the problem was in the backlog the work item was not shown under both Epics. That would be interesting to search further on how to deal with duplicate work items.

Later, I carry on my work on the dashboard and create a few new queries that retrieve the active Features and Epics. With the widget to display the result in the Dashboard, it is possible to see the ongoing stage of each case. Then my supervisor gave me more information about the task and wanted a visualization to see the number of cases for each stage. All stages are retrieved from a query that has some conditions managed with the tag, because the charts in Azure Dashboard don't support hierarchy query. The Features are named “Stage 1”, “Stage 2”, or “Stage 3” for now.

At the end of the day, I learned more tips in Azure DevOps and I was also curious about directly modifying a process of a project in Azure DevOps. I experiment that in a personal project, where I have full access on the configuration. The process currently used is Agile, but it is possible to create a sub process and customize it. So, it allows to create your own work items type and I replicated the hierarchy “Cases”, “Stage”, “User Story” and “Task”. That was maybe not necessarily for the project, but I found it interesting to give it a try.

Wednesday

Today there is the presentation of the new project management process to Sale and Delivery teams. Also, the objective of the day is to improve the Azure Dashboard and show more relevant analytics useful for the pre-sale process.

To sum up the presentation, it went well, and most people agreed on the need here to integrate a project manager in the presale stage. Eleonora explained the different issues related to it about the communication, the beginning of the project and the lesson learned. Also, we present the SharePoint that will be dedicated to project management and fill with processes, documentation and templates.

Later, I spend more time on the dashboard and had a call with my supervisor to review it. The pie charts created in the first place had to be updated every time a new case is created because the tag needs to be added in the list. Using the tag on the Features was the only way to know which stage it was referring, but now it is contained in the title. To avoid using the tag I changed the query to check the title of the stage and now the pie chart can use all tags (which are the cases’ names). Also, we changed the naming convention to “Stage-1/2/3” for the title of Features and to “Size-S/M/L/XL” for Epics’ tag.

I also tried to experiment another kind of widget that allows to display a table of query result. Here I wanted to show the number of cases per size and state. I made a simple query to retrieve all of the Epics and use the pivot table chart as widgets. Here how it looks:

EPICs by State

	Size-XL	Size-M	Size-S	Size-L	Total
Active	3	1	0	1	5
Closed	1	0	1	0	2
New	0	1	0	0	1
Total	4	2	1	1	8

Figure 21 : Number of cases per size and state

It gives a quick overview on the actual state of the different cases and also what has been achieved already. My supervisor liked it and he asked me if that was possible to have the data displayed per year. It is completely possible it just requires changing the query and add a condition for the year, but then I explained that it has to be done for each year in order to be updated.

Thursday

Today I'll be working on a new task that I received this morning. My supervisor asked me to do an integration exercise with Jira and Silverbucket servers. The aim is to create a webhook that is triggered when a new "Issue" is created and then it also creates a ticket in Silverbucket with data sent. For now, it is only on testing purpose, so I'll prepare my own testing environment. Also, on this afternoon we will have our weekly meeting to share our progress made.

I started by pulling a Docker image ready to go with Jira. To keep all the data after turning off the docker container, I had to create a volume and attached to it. Then I was able to configure the Jira server with existing data inside. The webhook was pretty simple to add from "Jira administration" => "System" => "Advanced" => Webhook => and "Add webhook". The service needed a URL in order to POST the request and here I decided to use Azure Logic Apps as it uses for this kind of job integration. To test the URL, I made a simple app to send a mail whenever a POST request is sent.

Unfortunately, that was not working, each time I was creating a new "Issue" in Jira I didn't receive that mail I was supposed to. At that point I was pretty sure that the webhook did not get triggered at all. My supervisor advised me to create a back end on Postman in order to test the webhook. I made some research on how to achieve that and then I created a "Mock server" with the default response body for POST and GET request. I used the server's URL for the webhook in Jira and I created a new "Issue". The request was successful, which meant that the webhook was triggered properly.

Then I had a quick look on the Silverbucket API and I tested a POST request to create a new Project. That was successful so I added that step in the Logic Apps, when it received a POST request from the webhook, the app will then trigger that request to create a new project in the other server. However, the webhook was not working with Logic Apps' URL, I made some research about this kind of issue and I found that on a forum, that might have happened because of the length of the URL (Atlassian.com, 2020). That true that the URL is quite long, but I was surprised that it was causing some issue there.

Along with my research, I've been noticing another kind of software used for integrating web applications named Zapier. I decided to give it a try, as Logic Apps I created a service to catch the webhook and the second step send a mail, because there was no integrated solution for Silverbucket. I create another webhook in Jira with the URL generated and I directly created a new "Issue". That was successful.

On the afternoon we had our weekly meeting, as this week was less intense, it was calmer. Saku presented Flutter, an open-source software development kit used to develop applications for Android, iOS, Linux, Mac and Windows and demonstrated the "hot reload" option. It allows to code and directly have the application rendered directly, very useful feature for development. Then we had a small talk on our April's fool and that was it for the meeting.

At the end of the day, I was a little frustrated that the webhook was not working with Logic Apps, because the other part was functioning and that was the last thing to do. As I described earlier, the issue might be due to the URL length, I'll have more searches on that next week. However, that was interesting to experiment Postman backend and Zapier with the webhook.

Friday

Good Friday

Weekly analysis

Another smaller week with a day off, however, still relevant in terms of new findings and knowledge. I've been experiencing some requests through the Python script to create and manipulate query from Azure Board. That was a requirement for this week of work, but I found it interesting and I'm pretty sure that will be needed in the future of the presale project. I didn't face any real problem here; I've just spent a little more time on understanding how to generate object JSON with a Python object because I didn't want to create an external JSON file for each request. The "json.dumps(var)" converts a Python object into a JSON string (Sathvik Chiramana, 2020) and "ast.literal_eval(json.dumps(var))" was needed to convert the double quote to single one.

At that time, it was working and I didn't look for more explanations on that, but when writing this analysis I've noticed that these steps are not required at all and the Python can be directly used in the request. The "ast.literal_eval" is used to safely evaluate an expression node or a string containing a Python (Python Software Foundation, 2020), in other words, it raises an exception if the input isn't a valid Python data type, so the code won't be executed if it's not. Basically, the script was converting a Python object into JSON string and then back to a Python object, these operations were unnecessary, and I could just use the object for the request. I'll bring the change next time I'll work on the script.

Another interesting discover during this week was the tips from my colleague, the roll-up column in the backlog of Azure board. Roll ups are an aggregation displayed on parent items (such as Epics, Features or even User stories) calculated based on relationships. It allows to have a quick view of the descendant items (all children) in the hierarchy, progress bars or the total of numeric fields (Microsoft.com, 2021). There are two types of roll-up column to add, Progress and Total roll up. **Progress** represents the percentage of completion of child items in the hierarchy as a progress bar, it is based on their state. On the other hand, **Totals** are not depending on state and are used to estimate size. This is an easy way to compare size of two Epics in terms of work. Here is a view of the Sandbox project with the roll-up column "Progress by all Work Items":

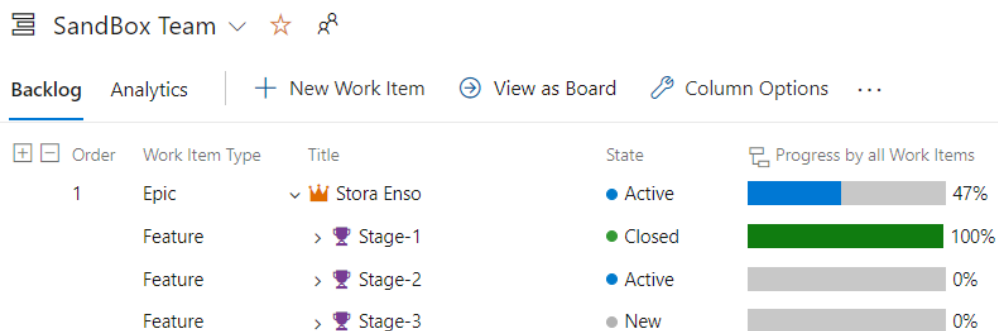


Figure 22 : Progress roll up column example

The main problem I faced this week was about the Logic Apps URL for the webhook in Jira. I was not able to trigger the apps with webhook, due to the length of the URL. Here an example of what it looks like (example not functional):

<https://oooo-45.iiiiiiiiii.logic.azure.com:443/workflows/2a25ab5678534d79b4366ff21593034b/triggers/manual/paths/invoke?api-version=2018-10-01&sr=%2Ftriggers%2Faaual%2Fron&sv=1.0&sig=6tXCV4ulfBjbZxJ1LSTKcZPbPbLDxoPuH-4S9p0EBk8>

Indeed, this URL is quite impressive at the first glance, but it is completely functional because I tested it with Postman. The forum where I read about this problem was suggesting creating an Azure Function in order to shorten the URL, but without more information on how to do that. I'll keep on working on the issue next week.

Nonetheless that was pleasant to use Postman in order to test either if the webhook was working or not. I didn't know about this functionality before and it turned out to be very useful in this case, because I was almost sure that the issue came from the webhook. It was pretty easy to set up with the documentation available.

I had no previous experience with Jira and Silverbucket, and I wanted to know a little more about it and in which case using it. Jira is a project management tool developed by Atlassian to track bugs, resolve issues, and manage project functions. Nowadays, Jira has grown from a simple application into a platform with a suite of other products. Jira software is used for software development teams with Agile methodologies such as Scrum. In addition, Jira can be easily customized and configured to suit special needs. Out-of-the-box, it offers bug tracking functionalities, but can also be configured to act like a helpdesk system or a simple test management suite (Sagar Ravi, 2019). On the other hand, Silverbucket is a Finnish company and a cloud service used for project resource management that allows to make resource plans and have a clear view of a company resource allocation. It contributes to better project planning and execution of projects.

Jira and Azure DevOps are both application lifecycle management (ALM) systems, that aim to help organization to manage their application lifecycle better. Azure DevOps is a tool to implement a DevOps lifecycle in Business, while Jira is a project management for software development teams. Azure tool is fully integrated with Git and CI/CD pipelines, while Jira has Bitbucket which has superior Jira integration built-in CI/CD (Sagar Khillar, 2020). To sum up Azure DevOps covers the whole software development lifecycle with many other embedded tools for versioning control, CI/CD, test plans and Jira is can be described as a project management and an issue tracking system.

3.9 Week 9 (05.04.2020 – 09.04.2021)

For this week the main objective is to complete the Jira and Silverbucket integration with Logic App or another solution available. This week is also a little shorter, but other side task could pop up along, probably related to Azure Board and Dashboard.

Monday

Easter Monday.

Tuesday

This morning I started the day with a meeting with my supervisor, I presented the issue that I was facing with the Logic Apps and what I had already done with Silverbucket and Jira webhook. After some discussion he advised me to use an Azure Function locally instead of the Logic App to integrate the two tools.

I started immediately to set it up with Visual Studio and a tutorial from the Microsoft documentation to create an HTTP trigger Azure Functions. I had to add the module “Azure Development” from Visual Studio installer software and the I was able to create a HTTP trigger with authorization level to “Anonymous”, so any service can access it. Then I was good to go, I ran the function with “F5” and the trigger was ready, listening to this URL “http://localhost:7071/api/WebHookTest”. I tested with a Postman request and it was working properly, but I was a little concern here because the function is coded in C# and I don’t know very well this programming language. So, I search for other possibilities to create such an Azure Function.

Quickly, I found another way to do it on Visual Studio Code and the function is available in multiple language Java, PowerShell, Go and Python. I installed the “Azure Functions” and “Azure Functions core tools” extension and then following almost the same process I created the HTTP function. However, I faced some trouble when I ran it, I was missing some packages in my environment and the script was not able to install them. After a while trying to make it worked, I decided to install them manually with the packages manage Chocolatey and it was successful. Now when the Azure Function is up and running, it is able to receive GET and POST requests from other services.

I created a new webhook in Jira that will trigger when a new Issue is created, and I made one to test. Unfortunately, here the Azure Function didn’t receive the request only the back-end server on Postman receive it, because of the previous webhook. So, I made more research on Internet if another person had a similar case, but I didn’t find something relevant enough to bring a solution.

I noticed that the request of the webhook was containing two parameters in the URL when the request was sent. I thought that was the cause, so I tried to remove that, but without success. The last idea I had for today was to use an HTTPS trigger instead of the HTTP. For that I had to generate a PFX certificate, which is a certificate file protected by a password that contains a certificate chain (intermediate authority certificate for the reliability) and the private key to the certificate (David Mizell, 2019). Then I had to modify the command to run the function, which is located in “.vscode/tasks.json” file.

```
{
  "version": "2.0.0",
  "tasks": [
    {
      "type": "func",
      "command": "host start --port 4034 --useHttps --cert development.pfx --password admin",
      "problemMatcher": "$func-python-watch",
      "isBackground": true,
      "dependsOn": "pipInstall",
      "label": "func: host start",
      "group": {
        "kind": "build",
        "isDefault": true
      }
    }
  ]
}
```

Figure 23 : Tasks.json after modifications

Now the Azure Function uses the URL “https://localhost:4043/api/WebhookTest”. I repeated the process to test it with the webhook and that was not working again. However, with Postman, I was able to send a POST request to the URL, I just add to accept the pop-up saying that the certificate used is not recognized by Postman. At this point I had no more idea to try so I decided to see tomorrow for more searching.

Anyway, the day was still interesting even if I lost a little bit of hope that is going to work. I learned how to create an Azure Function in Visual Studio Code and I was able to make work with Postman. Also, I didn't know about these kinds of PFX certificate, here I didn't really go deeply into the subject, but that was pleasant to test it out with the HTTPS URL. Probably later on, I will have access to a staging Jira server from the company so here I could test it in a real environment.

Wednesday

Once again, the day started with a meeting with my supervisor, we discussed about the Azure Dashboard and how to use the Excel file to add and edit tag on work items. Then I explained that I was out of ideas to make the webhook working and we tried to find out the issue together. After a while he advises me to check for the logs and see if at least the webhook is triggered or not.

I made research on how to access the Jira's logs, on the Atlassian documentation I found out a tutorial to generate the log especially for the webhook. In the "Logging and Profiling" menu, I add to configure the logging for the package "com.atlassian.webhooks" and "org.apache.http" with "Logging Level" to "Debug". I triggered a few times the webhook by creating new Issues and then I accessed the docker container with the command, "docker exec -it b7018a7ca4d9 /bin/bash". I used the command "find / -name "atlassian-jira.log" 2>/dev/null" to locate the log file and finally I copied it in the current folder with the command, "docker cp b7018a7ca4d9:/var/atlassian/application-data/jira/log/atlassian-jira.log .". Here are some interesting findings:

```
jspa [o.a.h.i.nio.client.MainClientExec] [exchange: 26] start execution
jspa [o.a.h.client.protocol.RequestAddCookies] CookieSpec selected: ignoreCookies
jspa [o.a.h.client.protocol.RequestAuthCache] Auth cache not set in the context
jspa [o.a.h.i.nio.client.InternalHttpClient] [exchange: 26] Request connection for {}->http://localhost:7071
[exchange: 26] connection request failed
```

Figure 24 : Jira logs, webhook for Azure Function

```
[o.a.h.i.nio.client.MainClientExec] [exchange: 24] Response received HTTP/1.1 401 Unauthorized
[o.a.h.i.nio.client.MainClientExec] Authentication required
[o.a.h.i.nio.client.MainClientExec] prod-45.northeurope.logic.azure.com:443 requested authentication
[o.a.h.i.nio.client.MainClientExec] Response contains no authentication challenges
[o.a.h.i.nio.client.MainClientExec] [exchange: 24] Consume content
[o.a.h.i.nio.client.InternalHttpClient] [exchange: 24] Connection can be kept alive indefinitely
[o.a.h.i.nio.client.MainClientExec] [exchange: 24] Response processed
[o.a.h.i.nio.client.InternalHttpClient] [exchange: 24] releasing connection
[o.a.h.i.nio.conn.ManagedNHttpClientConnectionImpl] http-outgoing-16 172.17.0.3:49562<->13.79.173.49:443[ACTIVE][r:r]
[o.a.h.i.nio.conn.ManagedNHttpClientConnectionImpl] http-outgoing-16 172.17.0.3:49562<->13.79.173.49:443[ACTIVE][r:r]
[o.a.h.i.client.cache.CachingHttpClient] Handling Backend response
[o.a.h.i.client.cache.ResponseCachingPolicy] Response was not cacheable.
anonymous [c.a.webhooks.plugin.PublishTaskFactoryImpl$PublishTaskImpl] Client error - 401 when posting to web hook
anonymous [c.a.webhooks.plugin.PublishTaskFactoryImpl$PublishTaskImpl] Client error - 401 when posting to web hook
```

Figure 25 : Jira logs, webhook for Logic App

On the first figure, we can notice the connection request failed, why still don't know the cause, but we have the confirmation that the webhook is fired and try to request the Azure Function. On the second on, the logs reveal that the request needs an authorization and we got the error 401, which indicates that the endpoint is restricted and would require an authentication, but the webhook failed to provide any.

After more reading and discussion with my supervisor, he sent me a link of a post in a forum that was reporting an issue about Jira webhook that didn't reach the endpoint. It was described that if Jira server is contained inside a Docker container, it is necessary to specify "host.docker.internal" instead of "localhost" in the URL. I immediately brought the changes and after a test, I was receiving a request on HTTP trigger function.

At the end of the day, I was very glad to make it finally worked, that was interesting to have a look on the log and how it can be helpful. I should have done that earlier in order to save more time, I didn't know how far I could go with the logs. Now that I have experienced that I'll keep in mind for the future to debug my issue in a more efficiency way. Also, the IT service didn't want to grant me admin right on the staging Jira server, because there is still sensitive information.

Thursday

Today the aim is to complete the integration with Silverbucket and Jira with the Azure Functions. Now that is working it remains to retrieve the name of the project from the body of the webhook request and use it for a new POST request on Silverbucket.

I added a simple function with the name as parameter. The request requires a token generated in Silverbucket, here I used the ".env" file to store it securely. I create two objects, the headers for the authorization and the body for the configuration of the new project. After some modification on the Azure Function, I was able to complete the integration. Now when the function is up and running, it will receive a request from the webhook whenever a project is created in Jira and will proceed to the creation of it on Silverbucket with the POST request.

```
Functions:
  WebhookTest: [POST] http://localhost:7071/api/WebhookTest

For detailed output, run func with --verbose flag.
[2021-04-08T11:17:13.438Z] Worker process started and initialized.
[2021-04-08T11:17:16.686Z] Host lock lease acquired by instance ID '000000000000000000000000A07DD212'.
[2021-04-08T11:17:25.220Z] Executing 'Functions.WebhookTest' (Reason='This function was programmatically called via the host APIs.
[2021-04-08T11:17:25.354Z] Python HTTP trigger function processed a request.
[2021-04-08T11:17:25.665Z] Code 201 - Project Test created successfully...
[2021-04-08T11:17:25.759Z] Executed 'Functions.WebhookTest' (Succeeded, Id=fc370f2c-e377-4732-8910-5f3110e3a935, Duration=562ms)
```

Figure 26 : Azure Functions up and running

Here we can see that the process went well, the status code 201 indicates that the request has succeeded and has led to the creation of a resource in Silverbucket. Next step is to think about how it is going to be in a production environment, for example version controlling, pipelines to deploy it into some testing/staging environment prior to deploying to production. Also, tomorrow we are going to have a meeting to talk about what data need to be retrieved from Jira into Silverbucket and I'll demonstrate the end to end process I've been working on.

Friday

Today is going to be more interesting for the next steps of the Jira and Silverbucket integration. My supervisor and I will have a meeting with the IT team about the how the Azure Functions will be maintained and what data needs to be retrieved. Also, we will have our weekly meeting to discuss about our work.

This morning my supervisor asked me to check for a generic way to define the manipulations on the data. Framework can be used to ease the mapping of the data and take care of the manual changes. For example, if the output data were in XML instead of JSON format, we won't have to bring that many changes in the code, only some settings in

the framework. After some research on Internet, I found a few frameworks that are widely used such as Pandas, Bonobo, Spark and many others. Pandas was the most interesting, so I decided to do more research on it. To sum up, it is a python library, that allows to manipulate and analyse data. It was first designed as a data analysis tool, it does all in memory and it can be slow when working with big data. We are working here with only a small amount of data, so that will suit well our need.

Later on, we had the meeting with the IT team, here I demo the Azure Functions and give a few explanations on how it works. So, the next steps are going to retrieve more data from the Project card created in Jira, such as “Name of the client company”, “Currency”, “Project key”, “Project manager” and “Start /End date”. The actual workflow is, for each new project, a Project Card is created in Jira and contains all this information, then a webhook create the actual project. The aim here is to retrieve all the information when the project is created by the request of the webhook. However, it doesn’t contain all the data needed so it will require an extra GET request. The Azure Functions can manipulate all the information and finally send a POST request into Silverbucket. Now I have access to the project card in Jira staging server and the Silverbucket testing server.

Weekly analysis

This week I was mostly working on the Jira and Silverbucket integration, where I set up a webhook that allows to send a POST request whenever new project is created to an Azure Function. It is configured as an HTTP trigger function in Python that allows to send another request to Silverbucket in order to create a new project as well. It was my first time to set up an Azure Function, that was quite simple to configure with the help of the Microsoft documentation. The first try I gave was with Visual Studio and the function was only available in C#. I didn't want to spend more time on learning this programming language. I have directly looked if there were Azure Functions in Python, which was the case for Visual Studio Code.

The issue with the Docker container takes me a long time to solve. I spent a lot of time trying to solve it with loads of research and tries. Here, I could have taken another approach to solve the problem, after that was not working, I keep on searching for the same issue and I was not going further. Checking the logs would have been a better idea to do when I was starting to face the issue. My intuition was that the webhook didn't trigger or that it was an incompatibility with the Azure Functions or something like this. When checking the logs, I realised how that was very helpful to target the problem and also talking with my supervisor about it, because he also gave me the way a little bit and redirect my search to the goal. The issue was about the Docker container and the local host connection. Docker has a specific domain name system named "host.docker.internal" that allows to resolve the IP address used by the container to the local host (Docker.com, 2020b). With that change, the webhook was able to reach the Azure Function that was running on the local host address.

Overall, this integration aims to combine different processes in order to achieve the company needs faster and better. In our case it is at smaller scale with only two services and a simple process automated. Business integration is defined as a process where companies put together different services and technologies, so all of them are coordinated and adjusted with the business strategy (Chakray.com, 2020). The data contained in Jira is now managed by Silverbucket as well. It brings many benefits for the company, such as (Adam, 2020)

- optimizing business processes
- reducing costs
- automating workflows
- reducing expense.

These benefits are depending on the scale of the integration, in our case it is only a small one. However, it is still interesting to do to avoid the time and effort spend manually sharing information with the services. System integration can be separated into three categories

(Folio3.com, 2019): DI; EAI; EDI. They stand for Data Integration; Enterprise Application Integration and Electronic Document Integration/Interchange. We are clearly in the EAI, here the Azure Function is playing the role of the middleware that allows integrations of the two services to communicate.

In general, this task allows me to build more knowledge on Azure Functions and the different services used in the company. I'm mostly collaborating with my supervisor, who helps to stay on track and give a guidance, so I don't lose too much time on issues or irrelevant details. Also collaborating with new people from the IT service is another good point, I can learn more about my colleagues and share experience with them. In this case they can be considered as a customer, because we are proposing them this new integration and they have their needs and specifications for this project.

3.10 Week 10 (12.04.2020 – 16.04.2021)

For this last week of entry, the Jira and Silverbucket integration is going to be a big part of the week. The goal of this week is to come up with an Azure Function operational locally with all the data processing (mapping) in order to create a project in Silverbucket. The next step will be to manage its maintainability and deployment in Azure DevOps. This week I also have two meetings planned, one for project management to keep up to date with the news and the other for the presale project.

Monday

Today's objective is to work on the Azure Function and process the data information in order to create POST request in Silverbucket. For that I'll be working with the Jira staging server and some issue that are already existing and the Silverbucket test server, where I can check if the request is working or not. The instructions were pretty clear from the last meeting, so I was able to start working on it directly.

I started to test the different requests with Postman to analyze the data I can retrieve from the Jira server. I need a call back request to get more information about the Project card, because the one sends by the webhook is not enough. For that I used an existing Issue in Jira and I started to map the data needed from it. As the Jira server as some customization for the project card Issue, some fields from the Issue are available as "customfield_XXXX" in the body of the request. So here I had to search which fields corresponded with which data. Here is all the data I need for the POST request:

```
{ } silverbucketPOST.json > ...
1  {
2    "name": "<string>",
3    "code": "<string>",
4    "startdate": "<yyyy-mm-dd>",
5    "enddate": "<yyyy-mm-dd>",
6    "customer": "<customer_id>",
7    "projectlead": "<user_id>",
8    "currency": "<currency_id>"
9  }
```

Figure 27 : Body of the SB POST request

To have a better structure in my project, I created two Python modules that are dedicated for Silverbucket and Jira requests and processing. For the Silverbucket request to create a project, the fields "customer", "currency" and "projectlead" require an ID of an existing value in Silverbucket. These IDs are not related to Jira at all and it provides only the name of it. So for the customer ID I had to create another request to retrieve customers and check if the customer exists or not, in the case that it is not existing it is required to create it otherwise it retrieves the actual customer ID. The same thing for "projectlead" but this time with the

email of the project leader and it must exist already. For the currency there is no possibly to retrieve the different value from the API, so here I decided to test it manually and attribute the following id for the value “EUR” = 1, “USD” = 2 and “SEK” = 3. This one can be probably modified, let’s see with the customer next time for maybe other ways to deal with it.

At the end of the day, I was very happy to make the Azure Function working properly with some testing data. I used Postman to simulate the webhook request and then the Azure Function handle the rest of the process. It didn’t work at the first time, but there were only few minor issues that I was able to fix quickly. Including an error that I faced for the time, circular imports of Python modules, basically my modules were importing each other and that was causing this issue. I still have some modifications to do, I was not working with the real data I used another kind of Issue in Jira, because I had no access to the actual Project Card. However, the most important structure is here, when I’ll get the data structure, I will just have to remap the data with the right field and the request will work again.

Tuesday

Another day on the Azure Function. Today the objective is to create a project in Silverbucket with the Data from an actual Project Card. Yesterday, I was able but only with testing data. Today, after discussion with my colleagues, I’ll have access to project cards in the staging server. I’ll need to process and manipulate the new data, in order to have a coherent state. Here is a resume of the fields in Jira and their equivalent in Silverbucket:

```
1 mapping = {
2     # Project start date
3     "customfield_11348": "startdate",
4     # Project end date
5     "customfield_11349": "enddate",
6     # Project key
7     "customfield_11062": "code",
8     # Summary
9     "customfield_11142": "name",
10    # Name of client company
11    "customfield_11144": "customer",
12    # Project manager
13    "customfield_11060": "projectlead",
14    # Currency
15    "customfield_13034": "currency",
16    # Billable
17    "customfield_12230": "invoiceable",
18    # Pricing Model
19    "customfield_12233": "invoicing_type"
20 }
```

Figure 28 : Jira and Silverbucket data mapping fields

It is also very helpful as a guideline, so I don’t get lost with the naming of the fields. I brought some changes to process the data in order to have the right value for Silverbucket. For example, the field “Billable” has the value “Billable” in Jira, but for the request I need, “Invoiceable” with a Boolean value. After modifying the Azure Function with the new data, the script is now able to get all necessary information from the call-back request and transform them to be usable for the Silverbucket POST request.

I've been in contact with my colleagues about the Azure Functions and they gave more precision on their need. It happens often that a Project Card is updated in Jira, the function needs to apply these changes on Silverbucket as well. Also, a project can be first created in Silverbucket and then in Jira, so it is needed to check if the project is existing or not before. Then, I started to work on the update change, I created another Python module dedicated to it.

At the end of the day, the function was able to use mock data from the Postman request to transform them for a PATCH request. The input data have a different structure from the other request, so here I had to remap fields' names and values. To optimize the script a bit, I decided to move the code to process the data in the "Silverbucket" module as functions, so the update module can use it as well. The update is directly visible on the server and it is working as intended.

Wednesday

As discussed yesterday, the aim for today is to take into account these different processes of updating or creating a project in the Azure Function. A project can be first created in Silverbucket and then the project card can be created later. The script should be able to check if the project already exists in Silverbucket or not. If it is the case, then the project is updated with the information of the project card. The Azure Function will need to check every request from the webhook with the event "created" and check if the project exists or not in Silverbucket.

After a while trying to code, I was not completely sure on how I could check if a project were existing or not. I discussed with my colleague and he explained to me that a project can be created in Silverbucket before Jira, but there is no real link between the two. The project can be created without a code and the name won't necessarily match the one in Jira. So here we have an issue, if we can't assure 100 percent that the project exists or not, we might come across duplicates in Silverbucket.

The solution proposed by colleagues was to have an additional field in the project card with a link to the matching project in Silverbucket. So, it adds a step to the user when creating a new project cards, he/she has to check manually if the project exists or not. That will allow the Azure Function to know what operation to do "create" or "update" if the field is filled or not and which project to update. However, in the case that the user doesn't fill that link and the project exist already in Silverbucket, the Azure Function won't know and will create a duplicate. That the issue I underlined, and we will have to talk more about it tomorrow.

Less coding for today, but important questions were raised about the workflow of the Azure Function. As the main process is not defined exactly and can be various, the function has to adapt to that and it complexifies the integration. That's not an easy process, but it is important to discuss about that and work as a team to make it works. It also raised the question about the data, which service has the master data and centralized them. All of these point needs to be taken into account to go further in the project.

Thursday

Objective for today is to deploy the Azure Function in Function App so I can test it already in its environment. I'll use Postman to send a request to the URL and see what happens and problem that the function face during the process. I'll also add it in an Azure Repository for the versioning control.

I struggle for a while trying to add the Git remote on Visual Studio and I almost lost some files when I was trying to commit and push my project. Some parts of the project disappeared after using the command and I was scared to have lost most of the script. Hopefully after some research I found that commit and the project was back, and I was able to push it on the Azure DevOps repository.

Then I created a Function App in Azure Portal where the function is going to be available. It was pretty straightforward to set up, I asked for the permission before because it involves cost. I wanted to create an Azure Pipeline to deploy to the Function App, but I was not able to set the build with the assistant. It requires more access, so I asked my supervisor and he replied to me that's an operation to do with another kind of account. I used the second option with Visual Studio Code, there is a simple way to deploy the function with the Azure extension.

With the URL generated the Azure Function was available, I tested it with some Postman request to simulate the webhook and I encounter a few issues. The update functionality was not working properly because the function used the project key to update a project in Silverbucket. However, the PATCH request requires its ID. I added function to resolve the project ID with a project key and now the update function was working properly.

After different requests tests, the Azure Function was working as expected. My objective is accomplished for today, I learned how to deploy an Azure Function and have more confidence using Git as well after I had struggled for a while. The function is only tested with mock data from my previous tests with my own Jira server, so there might be additional changes to bring to it when the webhook will be ready. That's because the current Jira server has some customization made in this kind of Issue "Project card".

Friday

Today objective is to provide a proper README file in markdown syntax to describe how the function works, the data mapping and the transformation of these data. I'll also have to recreate the analytics made previously for the presale project test in the official Azure DevOps project, it is going to be pretty straight forward.

I started my day with a small talk with my supervisor about the new dashboard and he also showed me some automation he made for the presale project. Basically, he used Power Automate in Teams to create a new case in the Azure DevOps project and all its structure. That's a significant gain of time here to automate these kinds of repetitive tasks. After that I added the dashboard widget and created the query for the analytics.

I spent the rest of the day working on the README file where I explain all the logic behind the Azure Function and the different operation that are made. The data mapping is very important to understand properly the project and how it works. It can be very confusing because each service has its own terminology for the same kind of data, this will give a good hand to the next people that will work on it.

Weekly analysis

This week objectives have been met; the Azure Function is functioning as expected. Unfortunately, I wanted to test it with the official webhook, but I didn't get news from my colleagues about it. It requires admin privilege for its creation, so I was a little stuck at this stage. The function would probably need some minor changes due to the custom fields of the customized Jira Issue. The data mapping was a little confusing at the beginning of the week, especially in the code. Sometimes I was using a name for a variable that was not reflecting its real context for Silverbucket or Jira data. I tried as much as I can to separate the Silverbucket and Jira transformation code. So, in each file I was using the terminology for this component, so it makes it clearer.

This principle is known as "Separation of concerns" in software development, where the aim is to avoid writing code in just one block, but to break it up into smaller pieces. Each of them should be able to achieve a single distinct operation or job (Alexey Naumov, 2020). We gain in code clarity and the different part of the algorithm are reusable by the other parts, even if that's was not expected at the beginning. In my code I reuse some functions especially for the transformation data, because it requires the same kind of output for the request, so it is used for the Jira data mapping and Silverbucket update module. This principle wants us to put all the function in the form of autonomous modules, where each of them are responsible to achieve a unique set of tasks that are logically related (Alexey Naumov, 2020). The functions less close are then separate and group up the features that serves the same objective.

Also, this week I became a little more confident with Git workflow after struggling to push my project in an existing repository. Here I was not very aware on how to use Git in Visual Studio Code and I thought I lost a major part of the files after a wrong commit. Hopefully, I was able to retrieve the commit and get my work back. The manipulations are pretty simple, but I didn't pay enough attention in what I was doing. I mess up when I wanted to stage the changes; instead I stashed the changes. Stash takes the modified tracked files, stage changes, in the working directory and saves them on a stack of unfinished changes that it is possible to reapply later. It allows to switch branches without committing the current branch (Javatpoint.com, 2020). Here is a more visual example of how Git stash is represented:

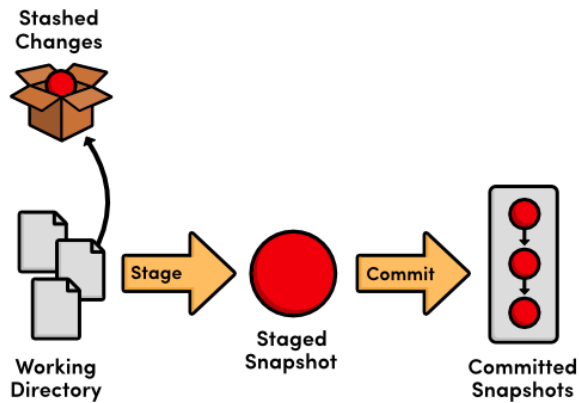


Figure 29 : Stashed changes schema

Basically, stash will store something safely in a hidden place without committing. That was completely my case here and I was not aware of this kind of functionality. The work was not lost, but just out of my vision. As described in this article, it is useful when you need to quickly switch to another task, bug report or request with a clean working directory (Ryan Irelan, 2020). Git stash will store the ongoing state of the working directory in order to have a cleaner directory to work with. It will save the local modifications somewhere safe and get the working directory in the state of the last commit.

This week I also faced some new challenges for the Jira and Silverbucket integration. At the beginning of the case, the functionalities on the paper seemed to be “simple” to implement. It was described as an Azure Function that will create a project in Silverbucket whenever a new Project Card (Issue) in Jira is created. Then a new option was needed to update the changes brought on a Project Card into an existing project in the other service. And finally, a project can be created in Silverbucket earlier in the process by another actor, which means that when the Issue is created in Jira, the function needs to check if the project is existing or not, before to either create or update it.

Now that all the business rules and the process are clearer, there are more conditions to take into account. While I was asking questions about the different cases, we realized that it is going to be a little more complicated. As the process can be triggered in different ways, the Azure Function has to adapt to all of these use cases. This integration is not over yet only a functional part is done; we will need to clarify the scope of the function even more. Here we could have been thinking about these concerns more at the beginning, especially from my side before starting programming. So, this would have given me a proper idea on the need and the scope to figure out a solution earlier in the process.

4 Discussion and Conclusion

This diary thesis reflects 10 weeks of my trainee on Azure DevOps, and other technologies related to the subject. At the beginning of the diary thesis, it was supposed to cover mostly the migration project to Azure DevOps, but it turned out to be more focused on studying and testing different technologies that are related to automation and DevOps. That was not an issue at all, but the title of the thesis was no longer reflecting its content. That the reason why it has been renamed “DevOps and Project Management Trainee”. The content is still relevant and related to Azure DevOps and some more general tools and technologies for these tasks. Before starting the trainee, I was already interested in the topic and specially to have a project management experience in a software development company. So, it was a great opportunity for me to gain knowledge and experience on these topics. I thought that I would have to deal with more project management tasks and being more involved on different projects. However, I was still able to experience with project managers different things and being able to help with them with Azure DevOps for their ongoing project.

Doing the thesis at the same time as the trainee was a good combination, even if I was a little overwhelmed and stressed sometimes. It allows me to practice and analyze my work weekly. It helped me to have a better understanding of the technologies, why it is used and what advantage it provides for the company. The weekly analysis was also very useful to review the problem I was facing and get more specific and technical aspect of my work. This experience allowed me to growth not only professionally, but also personally.

Right off the bat, I feel more confident talking about DevOps and other concepts related to it such as CI, CD and IaC. I have a better understanding of the workflow than at the beginning and more experience to back up my knowledge. At the beginning of the trainee, I did not know much about how it was used in companies and what the more detailed advantages of it were. I realized that automation is a significant point in IT and especially for software development where all the phases: plan, code, build, test, deploy and monitor can be automated with pipelines and other kinds of tools. Which improve the time to market, productivity of the team and reliability of the product/service.

In more general terms, this experience was very instructive for my English level as well. I was able to practice more with technical and more specific vocabulary during my research and also when interacting with my colleagues during meetings. That’s something I was looking for as well before starting my trainee, because I was looking for a bit more technical terms for my English. This will be great for my personal future and possibly other international experiences. The weekly analysis and research “forced” me to learn the terms and use them in my writing. Also, working with other trainees was also a good point not only

to practice English, but also because it was motivating. We were able to collaborate on some tasks and help each other, it was important because we all have different skills and sometimes one knows more about a specific subject. Sharing our research and demonstrating our work was interesting for the general knowledge and the improvement of each other.

In addition, having the diary thesis to fill every working day was helpful to target the issues and understanding the struggles I was facing. Just be writing down and explaining why I was blocked; it gave me more ideas on how I could try to solve it. Multiple times when I was writing about the issue, I realized that I could have done things differently or use another approach. This practice of deconstructing the problem is relevant to open your mind and I will probably use more in the future.

As I described earlier, I had no previous concrete experience with DevOps tools, it makes me realize its potential in the IT field. These methods of work are good practice and very helpful to understand when working as a project manager, so that's why I wanted to be aware of that. For project management side, Azure DevOps is also very interesting with the dashboard and the different analytics that are available in there.

Working with Git also was a relevant point, because before I was just using it for my personal project, never with other people on a same one. Now I have a better understanding of the workflow and functionalities. Especially for the stashed stage, I didn't find any use case for it now, but I'm pretty sure in the future I'll be confronted with it. Having a "README" in each project that I was working on was a good practice to keep and even improve it with more details and explanations even for my personal project.

Automating manual and repetitive tasks with Python script was a satisfying task to do and I learned a lot. Through the different tasks, I was trying to improve my code each time. It started with the naming convention in Python which was useful for a better clarity of the code and easier for another person to take on the script later. Also, hard-coding sensitive information was not a good practice and using the environment variable allowed more security for the code, especially when it is shared. The separation of concerns was interesting when working on the Azure Function, I had loads of functions and having only one file for all started to be very confusing and messy. The structure of the script was better organized with functions grouped into modules that was needed for specific operations. Then I experienced many libraries for some manipulations especially the one for the HTTP requests.

In general, holding a personal document where I refer interesting funds and describe the different configurations I had to perform, was good advice from my supervisor. It allowed me to save time several times when I had to redo the environment or any configuration. It will be even more useful in the future if I'm working with the same technologies. I usually had multiple sources of information and sometimes describes the general concept of the technologies. This document is also useful whenever my colleagues are looking for a specific configuration so I can share it right away.

During writing of the thesis, I came across multiple tools and technologies that helped me to understand better how the organization is working. At the beginning I started to work with the pipelines, I had to learn the structure of the file and how to translate the actions needed to be done in YAML syntax. There were multiple pipelines to build the Movesense app within a docker container or just with the tools installed on the agent. Also, to flash the sensor with the artifact generated from the build pipeline. As the flashing required hardware to be connected to the agent, I learned that it was not possible for a Docker container to access the USB port of a Windows host. This issue is well known and there is no other solution than working with a Linux environment. That's when I add to set the Raspberry Pi with the Linux distribution, and then I got the warning about the CPU architecture of the Docker image. The raspberry has an ARM architecture and the Docker image needed an AMD architecture.

Ansible and Ansible Tower were interesting topics, I learned the general concepts of the tools and use playbooks and roles. Even though the task was just an exercise with a testing environment, the approach would be the same to implement it in the company. I see the potential and the advantage that these tools can bring for configuration management. The concept of IaC was well described with this case and allowed me to understand it more. Along with the task, I also learn a little more the CentOS operation system and the different commands. I never used this Linux version before, but it was pretty simple to deal with it as there is a lot of information on Internet from the community.

There are a few issues that I was able to identify during the thesis. For example, when working on the Azure Function for the Silverbucket and Jira integration, we realized the real needs of the process only in the middle of the development. Integration is a significant challenge for business. In my case that was only for a small functionality, but I saw how it can escalate to become a bigger issue when implementing it. I'll be more aware during a project like this, to determine exactly the process and the type of data that will need to be retrieved. Ask more question about the scope and the functioning at the beginning or during the meeting with the customers. Having a scope well defined and determine the challenges that can be faced at the beginning of the process.

In addition, when I was working the Docker container for the Jira server, I faced that problem where Docker Desktop cannot access the local host address for the machine. That was interesting because I had no idea about that kind of issue. After some research and talking with my supervisor, we were able to find out the problem in a forum. It is a case that I'll probably face again in the future, so here it is a good lesson learned.

For further development, I'll probably look for an Azure DevOps certification later on. I very appreciate working with the tool and I'm still interesting to know more about it. There are still a lot of things to learn about, especially for the testing part in the pipeline that I didn't cover in the thesis. My objective is to have a better expertise with the tool and that will be a good advantage in the CV when applying for project management jobs. I like how simple the UI is and all these tools embedded for more complex tasks. On the other hand, Jira was also an interesting tool to work with even if I didn't use it as much as Azure DevOps. I'm glad to have experienced it and see how it was implemented in the company. As Jira is quite famous as well, that would also be interesting to look for more advanced concepts in the tool through different courses or formations.

Overall, analysing my weekly work was bringing more reflection on what I was doing and that allowed me to have a stronger comprehension of the tools, technologies, and topics of my tasks. It helped me to check if I had understood the matter or not, sometimes I realized that I missed something or misunderstand the instructions. So, after explaining it and analysing it I was usually able to fix the issue. For example, when I was working on the Python script to create a query in Azure DevOps, I tried to modify a Python object into a JSON string, but it was not necessary. During the weekly analysis, I was able to target this error and fix it in the code after that. Another example is with Git and the command "Stash changes", here I didn't know what manipulation I did and after researching or more information about the Git functionalities I find out my issue. Reporting my everyday work was a routine that helped me to keep track of my progress. Due to the pandemic and the remote work, the days were sometimes very similar, and it was hard to see the progress made. It was rewarding to see what was achieved previously and the thesis was helping to be focused on my objectives.

References

- Aayushi Johari. (2019). Chef vs puppet vs ansible vs saltstack. Retrieved from <https://www.edureka.co/blog/chef-vs-puppet-vs-ansible-vs-saltstack/>
- Adam, H. (2020). The importance of business integration. Retrieved from <https://www.cleo.com/blog/knowledge-base-business-integration>
- Alexey Naumov. (2020). Separation of concerns in software design. Retrieved from <https://nalexn.github.io/separation-of-concerns/>
- Ali Raza. (2016). Puppet vs. chef vs. ansible vs. saltstack. Retrieved from <https://jetpatch.com/blog/agent-management/puppet-vs-chef-vs-ansible-vs-saltstack/>
- Ansible.com. (2021). Can I run python modules on windows hosts? Retrieved from https://docs.ansible.com/ansible/latest/user_guide/windows_faq.html
- Ashraf Sharif. (2020). An introduction to MySQL deployment using an ansible role. Retrieved from <https://severalnines.com/database-blog/introduction-mysql-deployment-using-ansible-role>
- Atlassian.com. (2020). Jira webhooks not firing. Retrieved from <https://community.atlassian.com/t5/Jira-questions/Jira-Webhooks-not-firing/qaq-p/785220>
- Blazmater.com. (2020). Postman. Retrieved from <https://www.blazemeter.com/blog/how-use-postman-manage-and-execute-your-apis>
- Blogspot.com. (2014). Blog - how to convert date formats. Retrieved from <http://importpython.blogspot.com/2014/07/how-to-convert-date-formats-from.html>
- Chakray.com. (2020). How to successfully achieve business integration. Retrieved from <https://www.chakray.com/how-to-successfully-achieve-business-integration/>
- Chandrasekara, C., & Herath, P. (2020). *Hands-on azure pipelines* (1st ed.). Berkeley, CA: Apress L. P. Retrieved from [https://ebookcentral.proquest.com/lib/\[SITE_ID\]/detail.action?docID=6273755](https://ebookcentral.proquest.com/lib/[SITE_ID]/detail.action?docID=6273755)
- Cloud Advocate. (2020). Set up ansible for windows. Retrieved from <https://www.youtube.com/watch?v=73pgFXLYrfk>
- CodingEntrepreneurs. (2020). Using a REST API service in python. Retrieved from <https://www.youtube.com/watch?v=Sg5VTTBlhqq>

- CodingNinjas.com. (2018). Scripts and programs. Retrieved from <https://www.codingninjas.com/blog/2018/12/08/difference-between-a-programming-language-and-a-scripting-language/>
- David Mizell. (2019). How do I download a PFX file? Retrieved from <https://www.geocerts.com/blog/how-do-i-download-a-pfx-file>
- Docker.com. (2020a). Docker engine on ubuntu. Retrieved from <https://docs.docker.com/engine/install/ubuntu/>
- Docker.com. (2020b). Networking features in docker desktop for windows. Retrieved from <https://docs.docker.com/docker-for-windows/networking/>
- Douglas, K., & Douglas, S. (2009). *PostgreSQL* (2. ed., [Nachdr.] ed.). Indianapolis, Ind: Sams Pub.
- Drew Seewald. (2020). Using dotenv to hide sensitive information. Retrieved from <https://towardsdatascience.com/using-dotenv-to-hide-sensitive-information-in-python-77ab9dfdaac8>
- Edureka. (2020). Send a mail using python. Retrieved from https://www.youtube.com/watch?v=Y_tnWTjTfzY
- Folio3.com. (2019). What is system integration and how do you do it? Retrieved from <https://dynamics.folio3.com/blog/system-integration/>
- Gabriel Birke. (2017). How secure are ansible vaults? Retrieved from <https://lebenplusplus.de/2017/06/09/how-secure-are-ansible-vaults/#:~:text=After%20generating%20a%20256%2Dbit,%2C%20it%20is%20very%20secure.%22>
- Goasguen, S. (2015). *Docker cookbook* (1st ed.) O'Reilly Media, Inc. Retrieved from <https://proquestcombo.safaribooksonline.com/9781491919705>
- Guru99.com. (2021). PostgreSQL vs MySQL. Retrieved from <https://www.guru99.com/postgresql-vs-mysql-difference.html>
- HappyFox.com. (2021). What is BPM? Retrieved from <https://www.happyfox.com/what-is-business-process-management/>

Ivan Vanney. (2020). Debian xfce vs gnome. Retrieved from https://linuxhint.com/debian_xfce_vs_gnome/#:~:text=XFCE%20is%20a%20light%20desktop,PC%20users%20without%20touch%20screen.

Javatpoint.com. (2020). Git stash. Retrieved from <https://www.javatpoint.com/git-stash>

Kissflow.com. (2021). Project management reports. Retrieved from <https://kissflow.com/project/project-management-report/#:~:text=Reporting%20in%20project%20management%20simply,it%20for%20the%20entire%20team.>

Krasimir Hristozov. (2019). MySQL vs PostgreSQL. Retrieved from <https://developer.okta.com/blog/2019/07/19/mysql-vs-postgres#:~:text=Postgres%20is%20an%20object%2Drelational,more%20closely%20to%20SQL%20standards.>

Krief, M. (2019). *Learning DevOps*. Birmingham: Packt Publishing, Limited. Retrieved from [https://ebookcentral.proquest.com/lib/\[SITE_ID\]/detail.action?docID=5969307](https://ebookcentral.proquest.com/lib/[SITE_ID]/detail.action?docID=5969307)

Leigh Brenecki, & Suhas, S. G. (2021). Learn YAML. Retrieved from <https://learnxinyminutes.com/docs/yaml/>

Linuxize.com. (2020). Install xrdp on ubuntu. Retrieved from <https://linuxize.com/post/how-to-install-xrdp-on-ubuntu-18-04/>

Lithmee. (2018). Difference between GNOME and KDE. Retrieved from https://pediaa.com/difference-between-gnome-and-kde/#:~:text=The%20main%20difference%20between%20GNOME_applications%20to%20perform%20daily%20tasks.&text=GNOME%20is%20more%20stable%20and%20user%2Dfriendly.

Lutz, M. (2013). *Learning python* (5. ed. ed.). Beijing [u.a.]: O'Reilly. Retrieved from http://deposit.d-nb.de/cgi-bin/dokserv?id=4281849&prov=M&dok_var=1&dok_ext=htm

Medium.com. (2018). Introduction to ansible roles. Retrieved from https://medium.com/@mitesh_shamra/ansible-roles-1d1954f9932a#:~:text=Roles%20are%20defined%20using%20YAML,file%20which%20contains%20relevant%20content.

Medium.com. (2020). Azure DevOps YAML pipelines. Retrieved from <https://medium.com/swlh/azure-devops-yaml-pipelines-nine-features-you-should-know-about-534f348fc67a>

Microsoft.com. (2018). Logic apps vs azure functions. Retrieved from <https://docs.microsoft.com/en-us/azure/azure-functions/functions-compare-logic-apps-ms-flow-webjobs#:~:text=Functions%20and%20Logic%20Apps%20are,Logic%20Apps%20provide%20serverless%20workflows.&text=An%20orchestration%20is%20a%20collection,to%200accomplish%20a%20complex%20task>.

Microsoft.com. (2019). Azure pipeline resource. Retrieved from <https://docs.microsoft.com/en-us/azure/devops/release-notes/2019/sprint-160-update#pipeline-resource-meta-data-as-predefined-variables>

Microsoft.com. (2020a). Azure logic apps. Retrieved from <https://docs.microsoft.com/en-us/azure/logic-apps/logic-apps-overview>

Microsoft.com. (2020b). Devices in container on windows. Retrieved from <https://docs.microsoft.com/en-us/virtualization/windowscontainers/deploy-containers/hardware-devices-in-containers>

Microsoft.com. (2020c). Pipeline agent. Retrieved from <https://docs.microsoft.com/en-us/azure/devops/pipelines/agents/agents?view=azure-devops&tabs=browser>

Microsoft.com. (2020d). Templates - azure pipelines. Retrieved from <https://docs.microsoft.com/en-us/azure/devops/pipelines/process/templates?view=azure-devops>

Microsoft.com. (2020e). Time widget. Retrieved from <https://docs.microsoft.com/en-us/azure/devops/report/dashboards/cycle-time-and-lead-time?view=azure-devops>

Microsoft.com. (2020f). WIQL syntax. Retrieved from <https://docs.microsoft.com/en-us/azure/devops/boards/queries/wiql-syntax?view=azure-devops#:~:text=A%20query%20defined%20using%20the,by%20using%20a%20logical%200expression>.

Microsoft.com. (2021). Display rollup progress or totals. Retrieved from <https://docs.microsoft.com/en-us/azure/devops/boards/backlogs/display-rollup?view=azure-devops&tabs=agile-process>

Moira Alexander. (2016). Project management: To debrief or not to debrief. Retrieved from <https://www.cio.com/article/3119749/project-management-to-debrief-or-not-to-debrief.html>

Morris, K. (2020). *Infrastructure as code* (second edition ed.). Beijing: O'Reilly. Retrieved from http://bvbr.bib-bvb.de:8991/F?func=service&doc_library=BVB01&local_base=BVB01&doc_number=032361849&sequence=000001&line_number=0001&func_code=DB_RECORDS&service_type=MEDIA

Mosh Hamedani. (2019). Python for beginners. Retrieved from <https://www.youtube.com/watch?v=uQrJ0TkZlc&t=14059s>

Movesense.com. (2020). Docs. Retrieved from <http://www.movesense.com/docs/>

Mozilla.org. (2021). Web HTTP status . Retrieved from <https://developer.mozilla.org/en-US/docs/Web/HTTP/Status/401>

Python Software Foundation. (2020). Ast — abstract syntax trees. Retrieved from <https://docs.python.org/3/library/ast.html>

Python.org. (2013). Style guide. Retrieved from <https://www.python.org/dev/peps/pep-0008/#naming-conventions>

Rabiya Younas. (2021). Linux mint cinnamon vs linux mate vs linux XFCE: Which one is better? Retrieved from <https://ssiddique.info/linux-mint-cinnamon-vs-linux-mate-vs-linux-xfce-which-one-is-better.html>

Raspberrypi.org. (2019). Qemu on raspberry pi. Retrieved from <https://www.raspberrypi.org/forums/viewtopic.php?t=246886>

Ryan Irelan. (2020). Saving changes with git stash. Retrieved from <https://mijingo.com/blog/saving-changes-with-git-stash>

Sagar Khillar. (2020). Difference between azure DevOps and jira. Retrieved from <http://www.differencebetween.net/technology/difference-between-azure-devops-and-jira/#:~:text=piece%20of%20software,-,JIRA%20Software%20is%20well%20suited%20for%20software%20development%20teams%20that,the%20entire%20software%20development%20lifecycle.>

Sagar Ravi. (2019). *Jira quick start guide*. Birmingham: Packt Publishing, Limited. Retrieved from [https://ebookcentral.proquest.com/lib/\[SITE_ID\]/detail.action?docID=5649698](https://ebookcentral.proquest.com/lib/[SITE_ID]/detail.action?docID=5649698)

Sam Guckenheimer. (2017). What is infrastructure as code? Retrieved from [https://docs.microsoft.com/en-us/azure/devops/learn/what-is-infrastructure-as-code#:~:text=Infrastructure%20as%20Code%20\(IaC\)%20is,team%20uses%20for%20source%20code.&text=IaC%20is%20a%20key%20DevOps,in%20conjunction%20with%20continuous%20delivery.](https://docs.microsoft.com/en-us/azure/devops/learn/what-is-infrastructure-as-code#:~:text=Infrastructure%20as%20Code%20(IaC)%20is,team%20uses%20for%20source%20code.&text=IaC%20is%20a%20key%20DevOps,in%20conjunction%20with%20continuous%20delivery.)

Sathvik Chiramana. (2020). Json.dumps() in python. Retrieved from <https://www.geeksforgeeks.org/json-dumps-in-python/>

Segger.com. (2019). J-link docker container. Retrieved from https://wiki.segger.com/J-Link_Docker_Container

Shimmercat.com. (2020). Ansible schema. Retrieved from <https://docs.accelerator.shimmercat.com/tutorials/setup-with-ansible.html>

Skillcrush.com. (2021). What is python. Retrieved from <https://skillcrush.com/blog/what-is-python/>

Smartsheet.com. (2021). Project status report. Retrieved from <https://www.smartsheet.com/creating-an-effective-project-status-report-with-templates#:~:text=A%20project%20status%20report%20is,within%20the%20designated%20time%20frame.>

SocialTechnet.com. (2019). USB pass through. Retrieved from <https://social.technet.microsoft.com/Forums/en-US/26eee626-9df1-4d8a-8b36-0802e71bdaf2/hyperv-usb-passthrough-with-the-windows-10-development-trial-quest?forum=virtualmachingmgrhyperv>

Stackoverflow.com. (2018). Fixing docker permission. Retrieved from <https://stackoverflow.com/questions/48957195/how-to-fix-docker-got-permission-denied-issue>

Stackoverflow.com. (2019a). Call azure DevOps API with requests. Retrieved from <https://stackoverflow.com/questions/60341728/is-there-a-way-to-call-azure-devops-via-python-using-requests>

Stackoverflow.com. (2019b). Docker tty/std-in-out. Retrieved from [https://stackoverflow.com/questions/22272401/what-does-it-mean-to-attach-a-tty-std-in-out-to-dockers-or-lxc#:~:text=A%20pseudo%20terminal%20\(also%20known,a%20shell%20such%20as%20](https://stackoverflow.com/questions/22272401/what-does-it-mean-to-attach-a-tty-std-in-out-to-dockers-or-lxc#:~:text=A%20pseudo%20terminal%20(also%20known,a%20shell%20such%20as%20)

[bash%20.&text=In%20the%20case%20of%20docker,when%20starting%20a%20bash%20shell.](#)

VisualStudio.com. (2019). Docker exit code "1". Retrieved from <https://developercommunity.visualstudio.com/t/pipelines-docker-exec-fail-with-exit-code-1-when-u/804476>

Walkerscott.co. (2020). Azure functions vs azure logic apps. Retrieved from <https://walkerscott.co/2020/03/azure-logic-apps-vs-azure-functions/>

Wikipedia.org. (2021). Pbkdf2. Retrieved from <https://en.wikipedia.org/wiki/PBKDF2>

Xplenty.com. (2020). Oracle vs. MySQL. Retrieved from <https://www.xplenty.com/blog/oracle-vs-mysql/#:~:text=Key%20Differences%20Between%20Oracle%20and%20MySQL,-While%20both%20MySQL&text=MySQL%20provides%20a%20GPL%20software,OS%20X%20and%20z%2FOS.&text=Oracle%20is%20better%20for%20enterprise,Oracle%20requires%20a%20licensing%20fee.>