PDR

06/19/2022

Semester 6

Samir Zalmay
FONTYS HOGESCHOLEN

INHOUD

1. INTRODUCTION

This paper is aimed to demonstrate my learning objectives for semester 6 through the group project, with the group S65-1, and my individual project (Kwetter-Case). I describe what I've done, the decisions I've made, and how I arrived at them in this document. Enterprise software design and deployment are the focus of Semester 6. This sort of software requires a large number of people working on the application at the same time, as well as data transfers.

In Chapter 2, the projects will be further elaborated. The learning outcomes for this semester can be found in chapters 3 through 10. Each chapter will be broken down into subchapters, including sprints, reflection, and references (where the proof/research for the learning goal can be found). The conclusion and my reflection on this semester is provided in the last two chapters of the document.

2. PROJECTS & INFORMATION

As I briefly mentioned in the introduction, I will be working on two projects this semester. The first will be a collaborative effort, while the second will be an individual undertaking. Both projects have been designed in a way that they might be classified as Enterprise Software.

2.1 GROUP PROJECT

With the transition to renewable energy and (large-scale) consumers becoming "producers" as well utility companies must be able to balance their grid: under-capacity can lead to blackouts (consumers/factories not getting electricity) and over-capacity can lead to grid failure (exploding power plants, burnt wires, and so on), both of which result in fines and additional costs.

For the group project, our task is to develop a distributed electricity (market) simulation system for one regional energy grid utility company taking into account the stakeholders' wishes stated here below:

- Consumers (houses, schools, hospitals, etc.) / "Prosumer" (energy feedback to the grid)
- Large-scale consumers (chemical- or steel factories, greenhouses, datacenters, etc.)
- Utility companies (Stedin, Enexis, Liander, Enduris, …)
- Balance Responsible Party: BRP (TenneT)
- Producers: Power Plants (Nuclear, Coal, Bio, Gas, Solar, Wind, …)

The complete project description can be found here.

The group project is written with a Java backend and React Framework (JavaScript) frontend. The project can be found here.

2.2 INDIVIDUAL PROJECT

The goal of the individual project is to practice software development skills and to demonstrate that I can create enterprise software. Fontys also provided the Kwetter project that suits this goal, so I decided to work on the Kwetter case as my individual project. As Fontys says, this project should provide enough challenges to demonstrate my skills in the development of enterprise software considering both functional and non-functional requirements.

The project description can be found here.

The project itself can be found here.

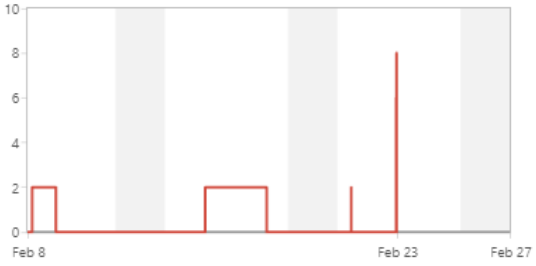
3. DEVELOPING ENTERPRISE SOFTWARE AS A TEAM EFFORT.

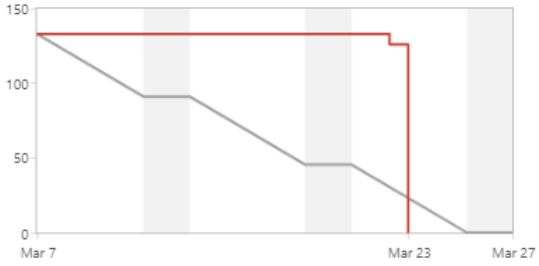
Development (undefined, orienting, beginning, proficient, advanced)

ID	Description	Deadline	Level
3.1	Sprint 0	Ended	Orienting
3.2	Sprint 1	Ended	Orienting
3.3	Sprint 2	Ended	Beginning
3.4	Sprint 3	Ended	Beginning
3.5	Sprint 4	Ended	Proficient
3.6	Sprint 5	19-06-2022	Proficient

3.1 SPRINTS 0 & 1

For this learning goal, everything is related to the group. During these sprints we had to get to know each other first and start to notice each other's strengths and weaknesses. Still, we got familiar quickly and we already established some rules and Setup the Jira board for these sprints. Just for these factors alone, the level should be at least Orienting.





But as you can see by the Jira burndown charts, the Jira board was not that actively/correctly used. However, as a team, we made a lot of progress and got to know each other very well.

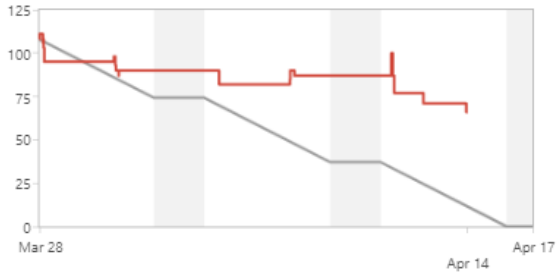
CONTRIBUTION

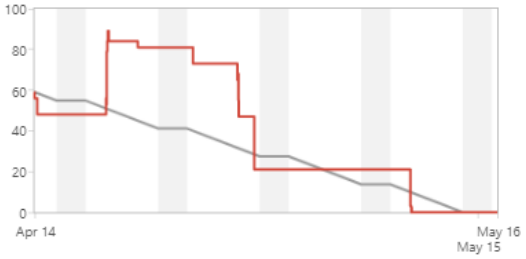
These sprints I mainly worked on helping the team with the documentation and working in the Backend authentication service. For the documentation, we all contributed to the Analysis, Design, and delivery documents.

As for the backend I helped with the authentication and setup the email-verification service.

3.2 SPRINTS 2 & 3

Unfortunately, during sprints 2 & 3 I had some personal issues regarding my health. This way I did manage to fully contribute to the team as much as I wanted to. I also started neglecting my individual work. The team, however, was very supportive and helped me a lot. Even though I missed a lot and did not see much of who worked on what, I still think the bond got stronger and the group grew a lot in general.





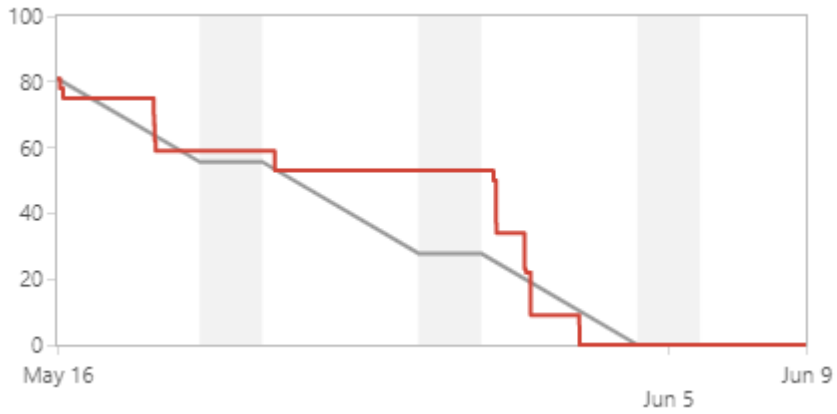
As you can see in the burndown charts, it improved a lot, although it is still not good. In my opinion the group deserves Proficient except for me.

CONTRIBUTION

Like I said before I was not participating well enough during these sprints. I still managed to do some research in Prometheus and Kubernetes and tried deploying them into local cluster.

3.3 SPRINT 4 & FINAL DELIVERY

During this Sprint I felt a bit better and think I managed to take a step ahead again. Our group performed great, and every Jira task got completed accordingly. I managed to also setup the deployments locally and helped Kristian with the google cloud environment. The Jira board speaks for itself this sprint and I think the group grew tremendously in professionalism but also socially. I think the group deserves Proficient if not Advanced for this learning goal. As for myself I am happy with Proficient.



In terms of the final delivery, we decided as a group not to make any new items and instead focus on completing the project. Because the deadline for the individual project is earlier

than the deadline for the group project, we are currently concentrating on the individual projects.

3.4 INDIVIDUAL

Because I work alone, I decided to employ the waterfall method instead of any Agile strategy for my own project. Despite the fact that this is enterprise software that should be scalable (additional user stories may be added throughout the project), this will not be the case for me.

I used the waterfall method to create Analysis and Architecture papers (3.5 References) to aid in the project setup.

3.5 REFERENCES & PROOF

In this chapter you can find the references made and every research/prove document for this learning goal. Each reference/proof is a clickable link that will redirect you to Github.

GROUP

- Project plan
- Architectural design

INDIVIDUAL

- Analysis document
- Software Architecture Document (SAD)

3.6 REFLECTION

The group I was with this semester was a wholesome and supportive group. I was very lucky to have joined them and they helped me a lot in taking my time in some stuff. Regarding the group work I tried my best to keep the pace up and not delay them in any way. I took some tasks that where bigger and managed to get by with the help of Kristian. If I did not have the issues in sprints 2-3-4, then I would have put this learning goal on Advanced.

However, there are some things that could have been improved since I did lack some communication during those sprints, and I also delayed my individual work. For these reasons I put the table on Proficient.

4. CONTEXT BASED RESEARCH

Development (undefined, orienting, beginning, proficient, advanced)

ID	Description	Deadline	Level
4.1	Sprint 0	Ended	Orienting
4.2	Sprint 1	Ended	Orienting
4.3	Sprint 2	Ended	Orienting
4.4	Sprint 3	Ended	Proficient
4.5	Sprint 4	Ended	Proficient
4.6	Sprint 5	19-06-2022	Proficient

4.1 SPRINTS 0 & 1 & 2

During these sprints I started with my emerging trends research plan. For this I set up my questions and that was basically it. During the second sprint I also started with researching Prometheus & Kubernetes for the group project.

I put this learning goal on Orienting since sprint 0, because I already had a lot of experience regarding working with the DOT Framework. However, as I did not provide much research results during these sprints, it will stay on Orienting.

4.2 SPRINTS 3 & 4

During these sprints, I stepped up my game and finished the Prometheus research documentation completely. The library research for Kubernetes is also finished, however, I still need to add the guidance for setting up the local cluster.

I also worked on my individual emerging trends research. Only the last question remains for this research. As I did a lot of research considering the DOT-Framework, I think Proficient is applicable.

4.3 FINAL DELIVERY

As I still have to do a lot for the other learning goals, I did not spend any more time on research, as this is already Proficient. I would have liked to finish my Emerging Trends research in Blockchain, as that affects me currently in my work environment.

4.4 REFERENCES & PROOF

All the research documents for this semester can be found here:

https://github.com/samirz5/Individual_Kwetter/tree/main/Documentation/Research

5. PREPARATION FOR LIFE-LONG-LEARNING

Development (undefined, orienting, beginning, proficient, advanced)

ID	Description	Deadline	Level
5.1	Sprint 0	Ended	Proficient
5.2	Sprint 1	Ended	Proficient
5.3	Sprint 2	Ended	Proficient
5.4	Sprint 3	Ended	Proficient
5.5	Sprint 4	Ended	Proficient
5.6	Sprint 5	19-06-2022	Proficient

Being at the end of my third year, I should be thinking about my future. I should be able to keep up with the new technology and know the path I want take as a Software Engineer. In this chapter I will explain in what thoughts I already have taken, and what my future plans will be.

5.1 SPRINTS 1 & 2 & 3 & 4

I did not think this learning goal would be an issue at the start of the semester because I had already planned my future. This learning goal, in my opinion, could have been on proficient from the start.

I started at Fontys with the intention of becoming a software engineer, and I have not changed my mind since then. During these sprints, I was also working as a software engineer for the startup Bull44, and I loved every minute of it. The project is responsible for updating and maintaining a website that assists users in investing/trading in stocks or cryptocurrencies by utilizing Algorithms that forecast what actions should be taken.

Because the project still needs to go live, my goal for this project was to set up the payment system and assist with CI/CD and deployment. These objectives matched Fontys' present learning goals precisely. Working with Bull44 has confirmed that I already have this learning objective set to proficient, as I want to continue this path for a few years following graduation.

5.2 FINAL DELIVERY

As I previously stated, I had some personal concerns between sprints 2-3 and a little portion of sprint 4 that prevented me from fully focusing on school. As a result, I did not give this learning goal or my future much thought.

However, I was able to climb back up during the last sprints, and after speaking with my teacher, I understood that this learning goal could be useful. The fact that I had picked the wrong minor also prompted me to consider what I want to accomplish next semester, as well as in the future.

After weeks of thought, I finally concluded that I still want to be a software-developer in the same field after graduation. A lot of minors came into play regarding following my path in which I doubted in choosing the following:

- **AI for Society/specialization**
  Always wanted to learn how A.I. works and what we can achieve eventually with A.I.
- **Data Driven Business Lab**
  As I am currently working with KPI's and different data, I would have loved to improve my skills in this region. However, as I probably will not be working at Bull44 after my minor, this one was doubtful.
- **Embrace TEC**
  I thought this was about Technology/Software and improving our skillsets in it, however after a talk with my teacher, I concluded that this was for starters basically.
- **Philosophy and Ethics**
  Always had an interest in Philosophy, however not so much in ethics. I would have loved to follow a semester of Philosophy; however, this does not suit in to my future plannings.
- **Entrepreneurship**
  After graduation I want to work as a Software-Developer for a couple of years to boost my experience. However, I always planned to set up my own small business based on web development for small companies.

There were a lot more minors that piqued my curiosity, but none of them were as well-suited to me as these 5. After much deliberation, I decided on the minor Entrepreneurship as the greatest fit for my career.

5.3 REFLECTION

Even though I did not think about it much this semester, I had done so in previous years. My plans from when I first arrived at this institution have not altered, and I am confident that I will be able to attain them sooner rather than later. However, after making a mistake with my minor, I became intrigued by the wide range of experiences that software development can bring.

Even though I think this learning goal should be on Advanced, as I have my future planned out already. A proficient would suffice, as I did not give much thought about it this semester.

6. SCALABLE ARCHITECTURES

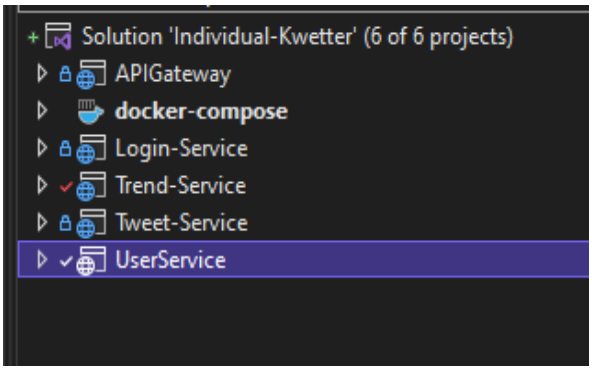
Development (undefined, orienting, beginning, proficient, advanced)

ID	Description	Deadline	Level
6.1	Sprint 0	Ended	Beginning
6.2	Sprint 1	Ended	Beginning
6.3	Sprint 2	Ended	Beginning
6.4	Sprint 3	Ended	Proficient
6.5	Sprint 4	Ended	Proficient
6.6	Sprint 5	19-06-2022	Advanced

This learning stands for me to develop enterprise software based on a chosen distributed architecture that clearly supports scalability for high volume communication and event handling and enables independent life cycle management. The following has been achieved during each sprint.

6.1 SPRINT 0 & 1 & 2

In sprints 1 and 2 I mainly focused on setting up the architecture and focus more on the documentation for now. For the group project I worked on the Architectural Design document together with the rest of the group. We came up with a Microservice architecture that fits best with our current user-stories. I have also contributed in setting up the backend authentication.



As for individual I also focused on the architectural design, but also the first setup with C-sharp and ASP.net core. As I was already familiar with Microservices, API-Gateway and Message bus, the first setup with organizing folders was not that difficult.

6.2 SPRINT 3 & 4

During this period, I was not able to do much as I would like have. However, for the group project I was working with Kubernetes, and had to do some research about it. This is when I

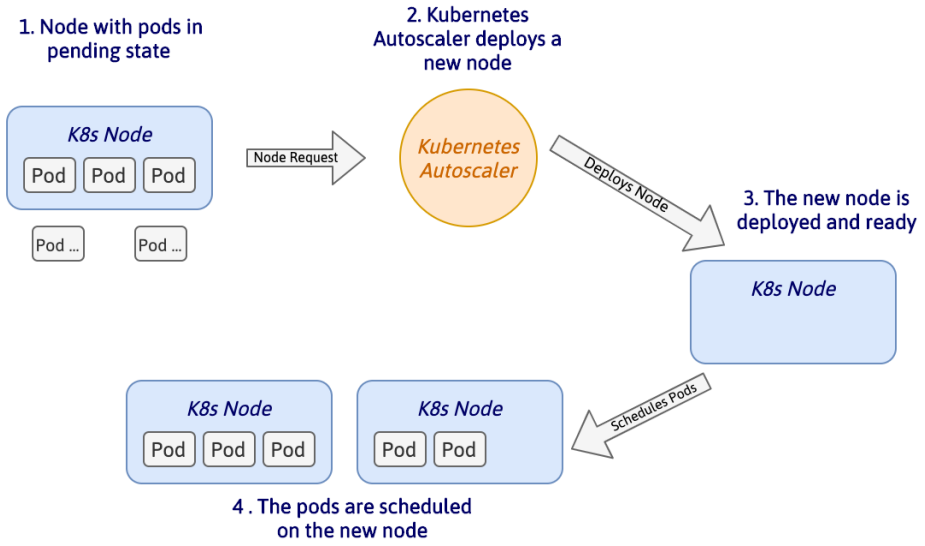
```
private void SendToKafka(string topic, string message)
{
    using (var producer =
        new ProducerBuilder<Null, string>(config).Build())
    {
        try
        {
            var test = producer.ProduceAsync(topic, new Message<Null, string> { Value = message })
                .GetAwaiter()
                .GetResult();
            Console.WriteLine(test);
        }
        catch (Exception e)
        {
            Console.WriteLine($"Oops, something went wrong: {e}");
        }
    }
}
```

stumbled on vertical and horizontal scaling in Kubernetes. For now the documentation for it was not complete and will be finished in the next sprint.

For the individual project I worked on coding locally and managed to make Kafka work locally with the Tweet-Service and Trend-Service. Once a tweet is made with a certain trend, this trend will be sent over to the Trend-Service over Kafka.

6.3 FINAL DELIVERY

For the final delivery, we decided as a group to mainly focus on the individual projects first. However, during the lasts sprints I finished up the Kubernetes documentation and explained the vertical and horizontal scaling in Kubernetes.



The image above shows how Kubernetes autoscaling works. The autoscaling can be set on Auto, or if you want to keep the costs at a maximum you can configure the Pod autoscaling in the deployment.yaml files.

In short, the vertical scaling in Kubernetes means ways to change the resources assigned to each node in the cluster (such as CPU or RAM). In most cases, this involves building a new node pool, using machines with various hardware configurations. On pods, vertical scaling refers to dynamically modifying resource demands and restrictions according on current application requirements (Vertical Pod Autoscaler).

Horizontal scaling refers to altering the compute resources of an existing cluster by adding more nodes or raising the replica count of pods, for example (Horizontal Pod Autoscaler).

6.4 REFERENCES & PROOF

In this chapter you can find the references made and every research/prove document for this learning goal. Each reference/proof is a clickable link that will redirect you to Github.

GROUP

- Architectural Document.
  Worked with group in first setup Microservices.
- Kubernetes Research

INDIVIDUAL

- Analysis
  Analysis I made for the Microservices, architecture is based on the Analysis
- SAD
  C4 architecture document updated.

6.5 REFLECTION

Working with Bull44 and having previously attended this semester, I believe my expertise of Scalable Architectures would be proficient straight away. During this semester, I have not changed many of the design decisions I made in the beginning. However, because I was not able to prove this until Sprint 4, the table also includes the word "Beginning."

In my opinion I have the expertise and knowledge to have this learning goal on Advanced, however I lack the proof to show this skillset.

7. DEVOPS

Development (undefined, orienting, beginning, proficient, advanced)

ID	Description	Deadline	Level
7.1	Sprint 0	Ended	Orienting
7.2	Sprint 1	Ended	Orienting
7.3	Sprint 2	Ended	Orienting
7.4	Sprint 3	Ended	Orienting
7.5	Sprint 4	Ended	Proficient
7.6	Sprint 5	19-06-2022	Proficient

For this learning goal we are expected to set up environments and tools which support my chosen software development process. I also need to provide governance for all stakeholders' goals and aim for as much automation as possible, to enable short release times and high software quality.

For this learning goal I already experienced working with:

- **CI/CD** (Jenkins and GitHub workflows)
  Setting up automatic builds and testing.
- **Lighthouse**
  Frontend performance testing.
- **SonarQube**
  For code-quality and vulnerabilities.
- **Elasticsearch and Kibana**
  For monitoring.

So, in my opinion I should already start out with Orienting at least as I already have some pre-knowledge in this learning goal. However, as I had this experience with DevOps, I somehow disregarded this learning goal at the first few sprints, while this should be one of the first to fix.

7.1 SPRINT 1 & 2 & 3

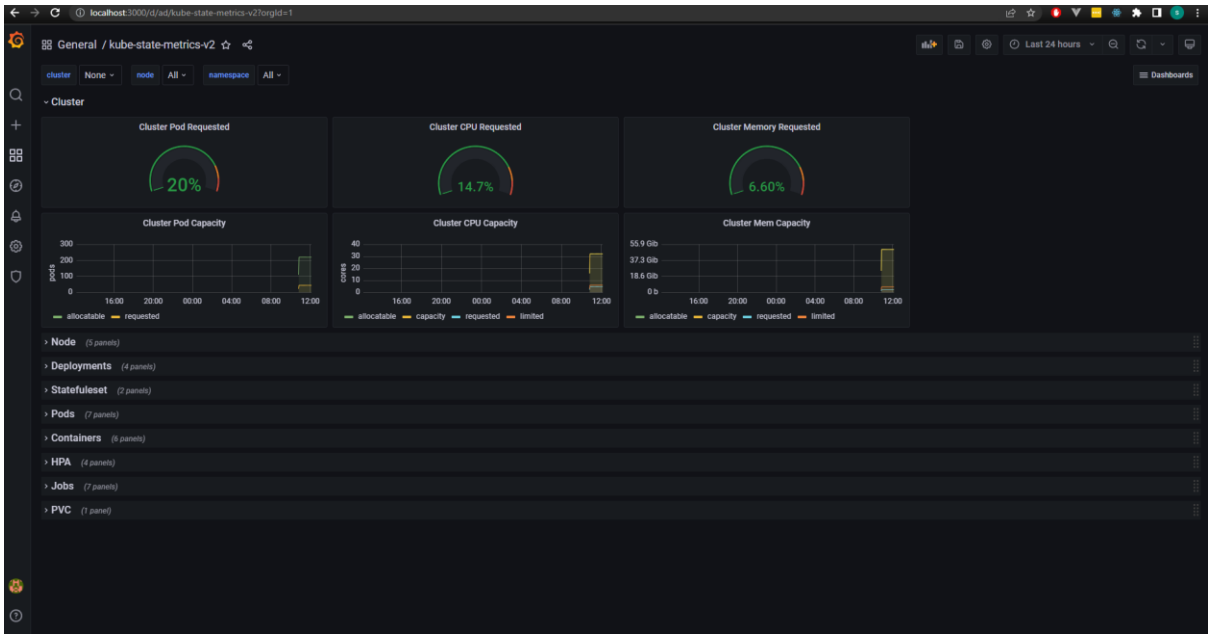
As I stated before, I barely did anything for this learning goal the first few sprints. The only thing I did, was some research into Prometheus/Grafana for the group project.

7.2 SPRINT 4 & 5

In sprint 4 & 5 I finally decided to work on this learning goal. For the individual project I decided to make use of Github Workflows for CI/CD pipelines. I decided for Github Workflows for the simple setup after consideration of Jenkins, Azure pipelines and GitLab pipelines. The Github workflow is currently working for some services, however there are some issues with docker.

As for the group project, I made quite some progress. During these sprints I managed to finish the Kubernetes research and deployed 1 Microservice with connection to the Database in my local cluster. Since the cluster was working, I focused afterwards on setting up Prometheus with Grafana on our local cluster. The research documentation for this can be

found in References.



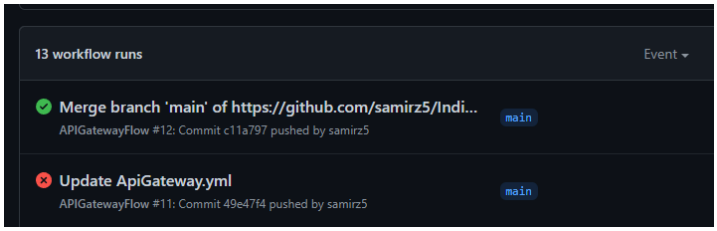
7.3 FINAL DELIVERY

To be quit frank I thought I would have this learning goal on Proficient by just having the workflows from Github implemented. As CI/CD pipelines are the most important ones to have in my opinion. However, I also noticed that I have not show complete proof in the workflows and monitoring yet. In this chapter I will explain every step I have taken in the last couple of weeks to ensure I have this learning goal on at least at Proficient.

For Github workflows the stages for building and testing were working fine. However, with Docker there were still some issues, as I restructured my folders for Github. Some small changes needed to be made for Github workflows to work completely except for deployment. As I am using C# as my backend, I can generate the Dockerfile and Docker-compose with a click. However, C# assumes that each of my microservice is in the root of my Github repository. To fix this, I had to add "Backend" in front of certain COPY and WORKDIR commands.

```
#See https://aka.ms/containerfastmode to understand how Visual Studio uses this

FROM mcr.microsoft.com/dotnet/aspnet:5.0 AS base
WORKDIR /app
EXPOSE 35000

FROM mcr.microsoft.com/dotnet/sdk:5.0 AS build
WORKDIR /src
COPY ["Backend/APIGateway/APIGateway.csproj", "Backend/APIGateway/"]
RUN dotnet restore "Backend/APIGateway/APIGateway.csproj"
COPY . .
WORKDIR "/src/Backend/APIGateway"
RUN dotnet build "APIGateway.csproj" -c Release -o /app/build

FROM build AS publish
RUN dotnet publish "APIGateway.csproj" -c Release -o /app/publish

FROM base AS final
WORKDIR /app
COPY --from=publish /app/publish .
ENTRYPOINT ["dotnet", "APIGateway.dll"]
```

This will bring the issue that docker is not running properly locally. Sadly, I have not found a workaround, and must make this small change every time I push to Github in order for repo to work.



As for the final step for Scalability I have taken is to add some performance tests locally. Unfortunately I could not deploy my project in any Cloud-provider, in which I was stuck in performance testing locally (see references).

7.4 REFERENCES & PROOF

GROUP

- Prometheus/Grafana document.
  Complete research with Library and Workshop methods to monitor your cluster.

INDIVIDUAL

- CI/CD & Performance test document.
  CI/CD steps explained and performance test.

7.5 REFLECTION

As stated at the start of this chapter, DevOps is not something new to me. However, I learned that CI/CD is not enough for DevOps, as monitoring and making sure your product is available 24/7 is just as important. I also did not believe that performance testing has anything to do with DevOps, but to make sure your product is running 24/7 and are automatically scaling, these performance tests are needed.

In general, I have more than enough knowledge and experience to at least get a Proficient for this learning goal. I would like to say Advanced, but I have not managed to actually performance test in the Cloud or add deployment to GitHub workflows as proof for Advanced.

8. CLOUD SERVICES

Development (undefined, orienting, beginning, proficient, advanced)

ID	Description	Deadline	Level
8.1	Sprint 0	Ended	Beginning
8.2	Sprint 1	Ended	Beginning
8.3	Sprint 2	Ended	Orienting
8.4	Sprint 3	Ended	Orienting
8.5	Sprint 4	Ended	Orienting
8.6	Sprint 5	19-06-2022	Proficient

As Fontys describes, for this learning goal I must be able to explain what a cloud platform provider is and can deploy (parts of) my application to a cloud platform. At first, I was a bit confused with this learning goal. The main focus for me for this learning goal was to get my project running in Kubernetes locally, and then make sure that my project is deployed in the cloud.

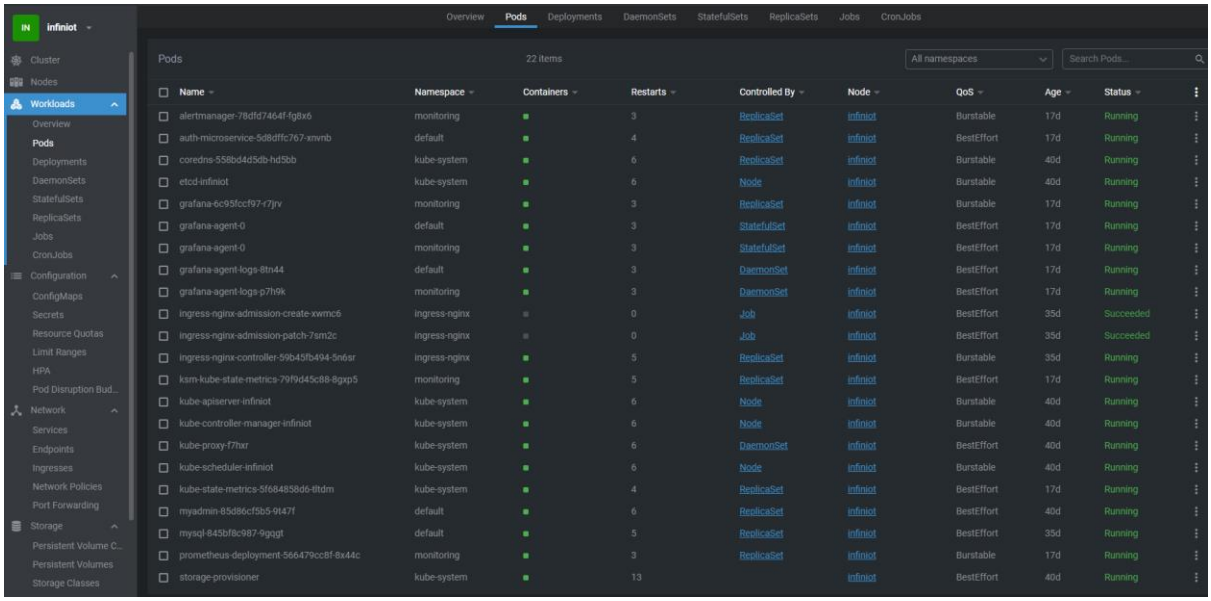
I only realized at the end that the goal for this learning outcome was not to deploy my project completely in the Cloud but knowing what Cloud services are and make sure I deployed at least one of my service or a FaaS/SaaS for my project. This made it easier for me at the end, but at the start I completely got misdirected for this learning outcome.

8.1 SPRINTS 1 & 2 & 3

As stated before, this learning outcome was a bit confusing at the start. The first couple of sprints as a group and individual we did not do a lot for this learning outcome. From sprint 2 and on I started working with Kubernetes for the group project. However, with the personal issues I did not manage to complete anything yet and my personal project got completely neglected in this.

8.2 SPRINT 4 & 5

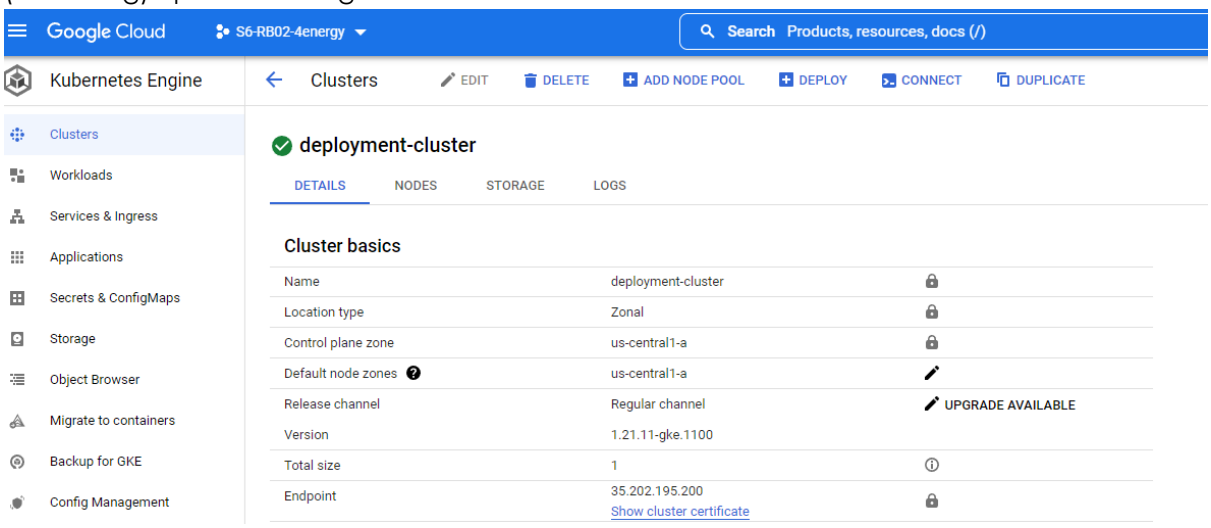
In this sprint I did a lot of research into Kubernetes and how to implement it locally for the group. The picture below show the pods currently running in the local cluster.



As stated, before I actually did not know that Kubernetes is not a part of the Cloud services. However, I am currently working as a software-developer for Bull44, and we are planning to go live on Google Cloud in July. My plan for these sprints was to deploy the complete project in Google Cloud and make it accessible from any computer. This would help me into getting more knowledge in Google Cloud Services and help me with my current work as a Software-Developer.

8.3 FINAL DELIVERY

For the group project I helped Kristian to deploy our group project to the Google Cloud environment. We decided to create the services that are needed for the other groups instead of the whole project, as this will consume a lot of unnecessary data. We managed to setup the deployment in Google Cloud with Mysql-database and the Authentication service (for testing) up and running.



However, we still had some issues regarding the API-Gateway. As the sprint for our group project is not finished yet by the time we have to deliver the Portfolio, we will work on this after the deadline for our individual projects.

For the final delivery I also realized that adding my project to the Google/Azure Cloud environment would be impossible to finish in time. The main problem was that I forgot to ask for an account from Fontys to have some credits in the Google Cloud environment and I did not have any Azure credits left. So, I did not bother to look at it till now, because I thought that setting up my project with Kubernetes locally would be sufficient.

However, after some research in Cloud Services, I somehow managed to get some Azure credits at almost at the end of the deadline, but I was not able to setup my project in Azure. To complete this learning outcome, I decided to add a Function as a Service for my project. This FaaS will be used as a Tweet-purifier. As one of the requirements for this project is that the tweets can not have swear words or hate speech in the tweet (See References Cloud-Service).

8.4 REFERENCES & PROOF

GROUP

- Kubernetes-Documentation

INDIVIDUAL

- Cloud-Service
  Research document with implantation of FaaS in my project.
- Example FaaS
  Uses FaaS and returns ****, the filter works.

8.5 REFLECTION

As I mentioned before, I mainly focused on getting my project into the Cloud, not realizing how many more services there are currently available. The way Cloud Services work gives me a little flashback to NPM with JavaScript. A service/function is written by someone and put online for everybody to use. I am actually quite surprised in how much I have learned for this outcome. However, I realized I still have not seen the amounts of services that are currently provided. For this matter I decided to only put Proficient for this learning goal, as I feel that I can learn a lot more.

9. SECURITY BY DESIGN

Development (undefined, orienting, beginning, proficient, advanced)

ID	Description	Deadline	Level
9.1	Sprint 0	Ended	Orienting
9.2	Sprint 1	Ended	Orienting
9.3	Sprint 2	Ended	Orienting
9.4	Sprint 3	Ended	Orienting
9.5	Sprint 4	Ended	Proficient
9.6	Sprint 5	19-06-2022	Proficient

For this learning outcome the goal is to investigate how to minimize security risks for my application, and how to incorporate best practices in my whole software development process. As I followed the specializations Cybersecurity A & B, I already know most of the vulnerabilities concerning web applications. This is the main reason why I put sprint 0 on Orienting.

9.1 SPRINTS 1 & 2 & 3

During these sprints I started with the basic security designs for Authentication and Authorization. In order to have access to the website a user must be logged in to the browser and backend. For this I have setup JWT-authentication in the backend of my Authentication-Service.

```
1 reference
private string GenerateToken(string id, string userName)
{
    var key = new SymmetricSecurityKey(Encoding.UTF8.GetBytes(config["JWT:Key"]));
    var credentials = new SigningCredentials(key, SecurityAlgorithms.HmacSha256);

    var claims = new[]
    {
        new Claim(JwtRegisteredClaimNames.Sid, id),
        new Claim(JwtRegisteredClaimNames.UniqueName, userName),
    };

    var token = new JwtSecurityToken(config["JWTF:Issuer"],
        config["JWT:Issuer"],
        claims,
        expires: DateTime.Now.Add(TimeSpan.FromMinutes(20)),
        signingCredentials: credentials);

    var encoded = new JwtSecurityTokenHandler().WriteToken(token);

    return encoded;
}
```

This will generate a token based on key and claims. The token will then be used in the other services to authenticate the user by adding the [Authorize] above the controllers I am using.

As for saving the user's credentials, BCrypt is used to encrypt the password with salt and verify that the credentials are correct.

```
2 references
public string Authenticate(string userName, string password)
{
    var user = GetUser(userName);
    if (user == null)
    {
        throw new NullReferenceException();
    }

    if (!BCrypt.Net.BCrypt.Verify(password, user.Password))
    {
        throw new UnauthorizedAccessException();
    }

    return GenerateToken(user.Id.ToString(), userName);
}
```

During these sprints I also started in getting my old OWASP top 10 report and reading thorugh it so none of the vulnarbilities apply to my project. Hereby I also updated the document bit by bit.

9.2 SPRINTS 4 & 5 & FINAL DELIVERY

During these sprints I mainly focused on having my OWASP top 10 up to date and made sure that none of the vulnerabilities are applicable to my project. However, there are still some steps that can be taken to make this project more secure (can be read in OWASP10).

I also added the Ocelot Gateway for my project. Instead of authenticating in each service, now the authentication happens in the Gateway. It ensures that all front-end API queries are routed through the Gateway. The Gateway then sends the requests to the corresponding APIs, where they must be processed. As a result, Ocelot Gateway provides a single point of entry for all inbound requests. An ocelot.json file is used for this. All front-end requests are verified and forwarded to the final APIs in this step. A piece of this file's setup is shown below.

Because I'm using microservices, having a Gateway in the project is beneficial. The diagram above illustrates how a request for retrieving, changing, or deleting tweets is routed. The UpstreamPathTemplate is the request that comes from the front-end. You can generate your own URL. The request is routed to the DownstreamPathTemplate when it arrives at the Gateway via this precise URL. This is the backend API's URL to which the message should be delivered.

The RateLimitOptions section of the ocelot.json file is another aspect of the setting. This protects the application from external threats. This is because the maximum number of requests per second is set here. DoS attacks are prevented as a result of this but also brute forcing. It also puts a stop to web scraping. Bots that scrape websites for data may exist, causing an application's performance to suffer. CORS is also in place, which ensures that only certain customers' requests must be approved.

The plan for the final delivery, was to completely update my OWASP top 10 to the new top 10 of 2022 (and in English instead of Dutch). However, since this learning outcome is already proficient, and I did not find the time to change the OWASP top 10, I left it as it was.

9.3 REFERENCES & PROOF

INDIVIDUAL

- OWASP10
  Complete description of the OWASP top 10 of 2017 with every chapter checking whether it is applicable to my project or not. And if yes, what steps can be taken.

REFLECTION

As I already followed Cybersecurity and implemented most of the security designs, this learning outcome should be at least Proficient. I would have stated Advanced if I managed to find the time to completely recheck OWASP top 10 of 2022.

10. DISTRIBUTED DATA

Development (undefined, orienting, beginning, proficient, advanced)

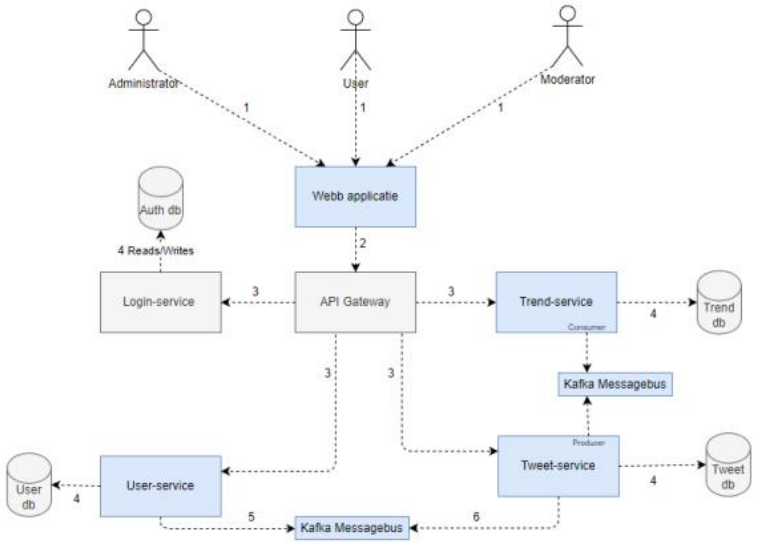
ID	Description	Deadline	Level
9.1	Sprint 0	Ended	Beginning
9.2	Sprint 1	Ended	Beginning
9.3	Sprint 2	Ended	Orienting
9.4	Sprint 3	Ended	Orienting
9.5	Sprint 4	Ended	Orienting
9.6	Sprint 5	19-06-2022	Proficient

This learning outcome was also a bit confusing at the start. As Fontys states "You are aware of specific data requirements for enterprise systems. You apply best practices for distributed data during your whole development process, both for non-functional and functional requirements. You especially take legal and ethical issues into consideration."

However, after trying to research distributed data, most of the results led to nothing. This made me even more confused in what was expected here. At first, I thought that distributed data, meant by separating your data in different databases. However, there is also the rules that data needs to be secure and GDPR-compliant.

10.1 SPRINT 1 & 2 & 3

During these sprints I have not accomplished much with this learning goal yet, as this learning outcome was not that clear yet. However, with the group and individual we needed to setup enterprise software. This means a large application that separates their services, including their databases. For distributed data during this period, I made some Architectural choices for the group.



I also had to establish multiple databases, one for each service, for each project. I wasn't sure which database I wanted to use at first. SQLServer, Postgres, and MongoDB were the three options I had set up. I chose to work with MySQL first, as I had a lot of experience with it.

However, I could not make MySql work with .net core somehow and was forced to use another database.

SQL is used by Postgres. Tables are used to hold information, and the tables are interconnected as needed. The tables also have a set timetable that must be followed; this reduces the chances of errors because everything is predetermined. Postgres keeps adding new tools and capabilities to help users control the correctness of their data. Postgres also comes with a lot of documentation, which makes it easy to learn. Postgres is a widely used database server that provides excellent performance, concurrency, and extensibility.

10. 2 SPRINTS 4 & 5 & FINAL DELIVERY

As for the last sprints I focused some more on this learning outcome. I started out by doing some more research (See references GDPR) into secure data and making sure each of my services are correctly separated with their own database. To be GDPR-compliant means ensuring that personal information of users is protected at all cost. If somehow the personal information is leaked, some measures must be ready to be taken instantly. The first measure that is already been taken is to hash and salt any personal data that is used for comparison. I am using BCrypt for this.

For my individual project I made sure that data is segregated well and added a delete user endpoint in which the user can delete everything personal to him/her. Once the user is deleted from the database, a message will be sent to Kafka to delete all user tweets from the Tweet-Service. The image below shows how Kafka consumes the message and talks to the tweetservice to delete all tweets by user id.

```
1 reference
public class KafkaConsumerHandler : IHostedService
{
    private readonly IServiceScopeFactory factory;

    0 references
    public KafkaConsumerHandler(IServiceScopeFactory factory)
    {
        this.factory = factory;
    }
    private readonly string topic = "delete_user_tweets_topic";

    0 references
    public Task StartAsync(CancellationToken cancellationToken)
    {
        using var scope = factory.CreateScope();
        var tweetService = scope.ServiceProvider.GetRequiredService<ITweetService>();

        var conf = new ConsumerConfig
        {
            GroupId = "st_consumer_group",
            BootstrapServers = "localhost:9092",
            AutoOffsetReset = AutoOffsetReset.Earliest
        };
        using (var builder = new ConsumerBuilder<Ignore,
            string>(conf).Build())
        {
            builder.Subscribe(topic);
            var cancelToken = new CancellationTokenSource();
            try
            {
                while (true)
                {
                    var consumer = builder.Consume(cancelToken.Token);
                    Console.WriteLine($"Message: {consumer.Message.Value} received from {consumer.TopicPartitionOffset}");
                    tweetService.DeleteTweetsByUserId(consumer.Message.Value);
                }
            }
            catch (Exception)
            {
                builder.Close();
            }
        }
        return Task.CompletedTask;
    }
```

10. 3 REFERENCES & PROOF

INDIVIDUAL

- GDPR
  Research document in what GDPR is and what measures I have taken and can take.

10.4 REFLECTION

This learning outcome was a bit confusing like stated earlier. It is important to keep data separated when using Microservices. This way we can prevent breaching other databases if one has fallen. However, that's about it regarding Distributing data in my opinion. This learning goal should be elaborated better in the title like Privacy & Data distribution, as only the name Distributing data makes it confusing for me at least.

However, after talks with the teachers, I knew that the more important goal of this learning outcome is to be GDPR-compliant. As I have a lot of experience with work in being GDPR-compliant and researched it again I definitely deserve Proficient.

Contrary to my individual project, with work we have added Terms & Agreement to the frontend. I also added a Cron-Job to check every week, whether a user has logged in, in the last 5 years. If not, then that user-account with all personal information will be deleted. These are two points that I can still add to the project and if added I would have set it to Advanced

CONCLUSION

I had a lot of experience with the current semester as I already did it once and where I am currently working, the project fully overlaps our learning goals for this semester. So, I can say I was surprised to learn a lot of new stuff this semester, but that is Software for you. I realized how important research is and act on this research to benefit in costs and efficiency.

In general, I learned a lot, but also implemented a lot of knowledge that I already had. However, in my opinion the current semester would have been passed with an Advanced if I did not have so many issues during sprints 2-3-4. However, since for the most of my learning goals I filled in Proficient, the conclusion should be that I deserve a Proficient.

RETROSPECTIVE

This semester was very hard for me. At the start I thought I would pass this semester with ease, but during this semester a lot of health issues occurred for me, in which my focus was completely gone. The fact that I have 2 jobs next to my study did not help as well, as I needed to spend a lot of time there to.

However somehow, I managed to pick myself up, and continue with this semester. This because of my group who were very supportive and the teachers that gave me at least a chance to prove to them my knowledge.

But still finding motivation after being so delayed was hard to do. Even the sprints after I struggled a bit with stress, but it luckily did not affect school that much. Even though it was circumstantial with my health issues and some personal problems, I should think of myself some more and quit some stuff when needed. Still, I am glad I continued and was able to produce this much at the end.

Because in this semester I surprisingly learned a lot, especially in Cloud-Services. I always confused this as deploying my own project in a Cloud service, instead of using third-party software of other developers. This opened my mind a bit, and I will explore this in the near future, as this will also be helpful with my work.

In general, I am happy with how I proceeded with school this semester. However, some points that still needs improving, are

- Communication skills (seems it got a bit worse since my internship)
- Documentation skills (just sit and document, as I always delay documentation)
- Planning (better planning with documentation).