LAPIN AMK
Lapland University of Applied Sciences

DISTRIBUTED PAIR PROGRAMMING SOLUTION USING DOCKER FOR EDUCATIONAL PURPOSES

Zohios, Andreas

Bachelor´s Thesis
Degree Programme in Business Information Technology
Bachelor of Business Administration

2022

CONTENTS

1 INTRODUCTION 7
2 PURPOSE, OBJECTIVES AND RESEARCH APPROACH 9
2.1 Purpose 9
2.2 Objective 10
2.3 Research Approach 11
3 THEORY AND FUNDAMENTAL INFORMATION 12
3.1 Pair Programming and its Purpose 12
3.2 Docker Implementation in Distributed Pair Programming Solution 13
3.3 Advantages of Docker Implementation 13
3.4 Operating Environment of Given Solution 14
3.5 Commercially Available Alternative Market Solutions 14
3.6 Knowledge Sources 16
4 ETHICAL FOUNDATION, RELIABILITY AND MATERIAL ACQUISITION 17
4.1 Ethics Examination 17
4.2 Reliability 17
4.2.1 Inspecting Reliability 18
4.3 Material Acquisition 18
5 CONCLUSION 20
5.1 Future Improvements 20
BIBLIOGRAPHY 22
APPENDICES 25

LAPIN AMK
Lapland University of Applied Sciences

Abstract of Thesis

Business Information Technology
Bachelor of Business Administration

Author	Andreas Zohios	Year	2022
Supervisor	Pekka Reijonen		
Commissioned by	Pekka Reijonen		
Title of Thesis	Distributed Pair Programming Solution Using Docker for Educational Purposes		
Number of pages	25 + 19		

The objective of this thesis was to create a distributed pair programming environment using docker containers in order to concentrate all the log files into one environment. This environment could be able to be used in coding education, as the teacher could collect the logs from the student pairs, in order to analyse and track student's progress.

The qualitative research method is exclusively used in this thesis. This thesis tried to gather observations, guides, and research conclusions in order to collectively use this information, to suggest a new solution, to avoid most of the crucial weaknesses of other solutions, while it utilizes all the main benefits.

The outcome of this thesis is a guide, which is an amalgamation of both teacher's and a student's guides, on how to setup the given environment on their systems. The complete version of this guide can be found in the appendix section at the end of this thesis. Finally, potential future improvements are discussed, which would provide a future upgrade on the suggested solution.

Keywords	Distributed, Pair Programming, Docker, Container
Other information	The thesis includes a two-section guide.

CONTENTS

1 INTRODUCTION 7
2 PURPOSE, OBJECTIVES AND RESEARCH APPROACH 9
2.1 Purpose 9
2.2 Objective 10
2.3 Research Approach 11
3 THEORY AND FUNDAMENTAL INFORMATION 12
3.1 Pair Programming and its Purpose 12
3.2 Docker Implementation in Distributed Pair Programming Solution 13
3.3 Advantages of Docker Implementation 13
3.4 Operating Environment of Given Solution 14
3.5 Commercially Available Alternative Market Solutions 14
3.6 Knowledge Sources 16
4 ETHICAL FOUNDATION, RELIABILITY AND MATERIAL ACQUISITION 17
4.1 Ethics Examination 17
4.2 Reliability 17
4.2.1 Inspecting Reliability 18
4.3 Material Acquisition 18
5 CONCLUSION 20
5.1 Future Improvements 20
BIBLIOGRAPHY 22
APPENDICES 25

FOREWORD

The commissioner/supervisor Pekka Reijonen served a great assistance by providing frequent suggestions, guidance, and motivation for the making of this Thesis.

Additionally, Tuomo Lindholm provided tremendous assistance for the construction of the Thesis Plan which became the underlying structure that hold together the whole Thesis.

I am very grateful for both contributions. The final result would not have been the same without it.

SYMBOLS AND ABBREVIATIONS

A compiled list explaining all the abbreviations used.

IDE	Integrated Development Environment
SSH	Secure Shell
VS CODE	Visual Studio Code
DFSG	Debian Free Software Guidelines (Debian 2022)
XP	Extreme Programming

1 INTRODUCTION

In the digital age that we live in, the ability to communicate with computers is a necessity that our lives rely on. Many new professions depend on programming and existing programming jobs are becoming exponentially more popular due to the digitalisation of products and services (Codiska 2022). Humans have the constant need to improve and maximize the efficiency of everything they do, and computer programming is no exception. While they are many techniques on how to improve coding efficiency, the most popular methodology is called agile. Agile is an iterative approach to software development and project management which helps teams deliver value to their customers faster and more effectively (Atlassian 2022).

Pair programming is an Agile software development technique which originates from extreme programming (XP). In pair programming, two developers team together in one computer, to design, code and test user stories. Ideally, the two programmers would have equal programming skill lever and they would spend equal time at the keyboard. (Gillis 2021.) Distributed pair programming has the exact same definition as pair programming with the main difference being that the developers are working from a distance (Wang & Luo & Dou 2010).

The main objective of this thesis was to create a distributed pair programming environment in which logging in achieved effortlessly in order to use that environment for educational purposes. Logs in programming can be used for a wide category of processes as they are basically a transcript of all the events that have taken place during the application runtime. Such events can be the inputs and outputs of the application, code executions, errors as well as user actions (Vardalos 2017). All the previously discussed events can be viewed as the metadata of the coding process, and as a result, they can be further analysed and examined to determine all the actions that have taken place in the coding environment. Metadata summarizes basic information about data, which consecutively can make the data easier to read, find and use (Kranz 2020).

During my academic years, I only received a very limited amount of pair programming education due to the social distancing regulations of the COVID-19 pandemic. That is when I realised that there is insufficient number of available solutions for distributed pair programming in education. If students were able to code in a virtual environment in pairs, then, that education part would have never been disrupted and the teacher would have been able to examine student's progress more effectively.

One of the challenges programmers faces when programming in a new virtual environment, has to do with the missing dependencies of the code which introduce environment-dependent problems (Narang 2022). When a developer enters a new programming environment, a big amount of their time is spent on familiarising themselves with the environment and adjusting the environment accordingly to the missing code dependencies. In pair programming, that is always the case for at least one of the two developers (Crouch 2021). The outcome of this thesis will be a complete guide which will enable the reader to create a "ready to code" environment, in which the developers can concentrate exclusively in programming. Furthermore, this environment will be accessible from everywhere and it will be easy to recreate in case of need.

So far, only the benefits for developers in general have been analysed for this solution. To personalize this subject more, there is a need to concentrate on the educational benefits. In pair programming, there is no guarantee that all the work would be done by one student and the other one would be idle. Student activity is important in order to denominate individual progress which is a value that can be evaluated afterwards. That is the reason why, this solution is trying to use a common terminal for both users in order to enable the opportunity to obtain data which could be potentially analysed to showcase student activity. If the work balance seems uneven, the student pair could be changed by the teacher, to achieve a better-balanced pair.

2 PURPOSE, OBJECTIVES AND RESEARCH APPROACH

2.1 Purpose

As mentioned in a study conducted in the University of Macedonia, there are very few studies which concerns the key features of distributed pair programming in education (pair formatting, role contribution, compatibility) (Tsompanoudi & Satratzemi & Xinogalos 2022). As a result, there is a strong need to develop easier methodologies for educators to be able to create virtual workspaces for their students, which promote distributed pair programming and additionally offer easy monitoring for each student's progress. Pair programming is a very important skill to develop and as the remote working practice becomes increasingly more popular, consecutively, distributed pair programming would follow.

As it was previously discussed, logging can be used in order to accurately monitor the most crucial actions inside a software development environment. Typically, logs can be obtained from the user terminal, but, when they are multiple developers working from many different computers, the logs are scattered, and they are impossible to be all collected automatically. In order to fill this void, the thought of collecting all the developers in a common virtual workspace (container) was adopted, which would consecutively result in a common collection point of all terminals, as well as a very efficient working environment. The present subject is the technological gap this thesis is trying to fill.

This thesis, although is aimed at assisting educators, it remains equally educative for anyone who is interested in creating an optimal distributed pair programming environment for themselves. Ideally, the final product of this thesis would be an intelligible guide to follow, in order to create a "ready to code" disposable virtual environment in which two or more developers will be able to access and program together. Additionally, a very important aspect of that environment will be the use of strictly one common terminal, which greatly simplifies the potential future analysis of the metadata gathered by that terminal.

Seemingly, many developers would argue that distributed pair programming could easily be achieved simply by using a screensharing software or an IDE plugin. IDE is an abbreviation for an integrated development environment which is a software for building applications which combines developer tools into one graphical user interface (GUI) (RedHat 2019). This thesis analyses those options carefully and compares them with the solution provided. As a result, there is a direct comparison of the most popular market solutions currently available, with all the docker capabilities for distributed pair programming. By doing so, all the key benefits that are granted by using docker are showcased and discussed, which consecutively showcases the benefits obtained by the outcome of this thesis.

Another important question which is answered in this thesis has to do with the collection of log data in the pair programming session. Typically, using different distributed pair programming solutions, the logs obtainability of the sessions could be overly laborious or even impossible in most of the cases. The solution offered in this thesis, aims to minimise the difficulty of this task, by carefully considering how to gather all the metadata in one terminal. By doing so, it could become an academic tool, as more variables can be controlled and monitored. In conclusion, the research question this thesis is trying to answer is the accomplishment and execution of a feasible method for collecting log data from multiple developers, that are working on the same distributed pair programming project.

2.2 Objective

The ideal end result would be a simplified guide’s to produce a distributed pair programming environment for educational purposes using docker. Furthermore, if the guide proves to be fully functional, it will be tested and reported from nonexpert users, in order to showcase the guides descriptive capabilities and functionalities.

Ideally, this guide should have also been tested in a real academic environment for the purpose of showcasing and reporting the student learning experience

inside the constructed digital educational environment suggested in this thesis. Unfortunately, this goal is beyond the scope of this thesis, but it could provide a good research topic for people that are interested to try this solution and receive student feedback.

2.3 Research Approach

They are three main categories of research methodologies used in academic studies. These categories are the quantitative method, the qualitative method and finally there is also a mixed methodology which combines aspects from both. The quantitative approach generally goes by the belief that there is only one truth that needs discovering, and that is the reason why, the data are in the form of numbers from precise measurements transcripts. In contrast, the qualitative approach supports the constructionist perspective (knowledge is created, not discovered, and there are multiple realities based on someone's perspective) and consequently, the data are usually in the form of observations, documents, and transcripts. (Sheppard 2020.)

Due to the nature of the given thesis, the qualitative research method is exclusively used. This thesis is not trying to generate numerical data to be classified, rather, tries to gather observations, guides, and research conclusions in order to collectively use this information, to suggest a new solution, which combines the benefits from many others, while it tries to avoid most of the crucial weaknesses (Enago 2021).

This thesis, undoubtedly falls into the descriptive category, as it tries to accurately paint a picture of the currently available solutions in the market, and its ultimate goal, is to provide a step-by-step guide which simplifies an objectively complex procedure. Moreover, there is a type of classification section which categorises commercially used solutions and sequentially compares it with the solution suggested in this thesis (Neuman 2003). All the previously discussed attributes, infuse together, to produce a descriptive research methodology.

3 THEORY AND FUNDAMENTAL INFORMATION

3.1 Pair Programming and its Purpose

Pair programming is a concept in which two developers are simultaneously working side by side in one computer in order to produce code. In Distributed Pair Programming (henceforth DPP), the whole process remains the same with the pair programming, but the main difference between them is that the developers are sharing a virtual workspace to collaborate instead of being in one computer (Tsompanoudi & Satratzemi 2014, 259-263). Both developers have different roles with one being the “driver” and the other one being the navigator. Generally, the driver is responsible for writing the code and the navigator is asked to verify that the code does not have mistakes and is heading towards the right direction. It should be noted, that both developers are encouraged to switch roles whenever they consider it beneficial (Singh & Andrews 2022). Pair programming is a social skill in its core and undoubtably it needs some getting used to for the newcomers (Wells 1999).

Pair programming purpose is to break the balance between giant flowchart software development and solo programming providing a medium solution (Garber 2020, 23-31). A flowchart is a diagram that depicts a computer algorithm, and it is widely used in multiple fields to document, plan, study, improve and communicate often complex processes in easy-to-understand diagrams (Lucidchart 2022). When programmers are tasked to write complex algorithms, mistakes are more likely to occur. That is the reason why, having a pair of programmers working together becomes so compelling, as mistakes in software engineering can be proven to be very expensive (Garber 2020, 51-54)

Pair programming in encouraged from the agile software development (ASD) paradigm, which has been a basic pillar of software engineering for a greater period than two decades already (Aitken & Ilango 2013). As of the year of this writing, 16% of the companies in the world have started working completely remotely (Kalcheva 2022). Distributed pair programming could be seen as a link between pair programming and remote working.

3.2 Docker Implementation in Distributed Pair Programming Solution

Docker is another fundamental element of the solution, and therefore, it is very important to understand precisely what it is. Simply put, it is a containerized platform which enables developers to package all their coding necessities in order to run code in any environment. A docker container is a standardized unit which can be created to deploy a given environment or application (Chaturvedi 2022). Generally, developers can create containers without the use of Docker using a collection of tools like CRIO-O, CNI and OCI (Shirinkin 2020). Docker provides many simplifications to the whole process, which in conclusion, simplifies the procedure described in this Thesis (IBM Cloud Education 2021).

The main purpose for using docker for pair programming is to obtain a solution for a digital environment which is agile, reliable, repeatable, and easy to reconstruct. Meanwhile, other developers will be able to safely connect to it and contribute changes. In that way, a portable docker image would be acquired which would be secure, and easily accessible via the SSH network protocol. (Petersen 2022.)

3.3 Advantages of Docker Implementation

One of the main advantages obtained using Docker has to do with the bandwidth. For example, one of the most used tools for pair programming, is screensharing. The remote session receives lagged compressed images from the hosting end, instead of the immediate crisp text. This problem becomes even more noticeable when the Internet connection on either end is weak. In that case, the host does not get affected, but the receiver is constantly at a disadvantage which creates communication hazards. (Petersen 2022.)

Another important advantage received from the docker has to do with the digital environment availability. In other cases, if the host needs to leave, the session gets terminated instantly which leaves the receiver in a position which does not allow him to progress. In contrast, if the development environment is on a cloud server, all remote developers have immediate access to the updated progress

which was made previously by the host. It is also important to highlight, that in the event in which the remote developers want to detach from the session and return later, all their changes remain unchanged, and the session is always available for them. (Petersen 2022.)

3.4 Operating Environment of Given Solution

This Thesis subject was inspired due to an ongoing need for an open-source solution which promotes digital remote pair programming in education. The subject was commissioned by Pekka Reijonen, who a teacher in Lapland University of Applied Sciences. This thesis outcome is planned to be used from the teacher in the future.

If the solution will meet all basic requirements needed from the teacher, this solution could potentially also be useful for any developer that wishes to work remotely in a virtual coding environment with other developers. They are many benefits that can be obtained from the given solution, as previously discussed, so the target group of this thesis extends to anyone that can benefit from them.

3.5 Commercially Available Alternative Market Solutions

Amazon provide their solution with AWS Cloud9 which is a third-party product that equips the developers with a range of tools and supports over 40 different programming languages. Although the IDE offers many benefits to the developers, it still does not manage to successfully provide the “follow my lead“ mode that Visual Studio Live Share does, and also charges for each instance which makes it a paid solution. (Yegulalp 2020.)

Codeanywhere is another popular solution and has the unique future that it provides the opportunity for the developers to code on the go as it can run instances in the browser and they become available from computers, tablets as well as phones. It also supports over 75 different languages and most of them can be also executed in-cloud environments. Users can access your instance from links, from real-time collaboration access to your editor as well as from SSH

access to the given project. The charge currently stands at 3$ per user per month. (Yegulalp 2020.)

Undoubtedly, GitHub has been the basic pillar in software engineering for storing and managing code as well as providing detailed version control system. So, there is no surprise that the GitHub team has contributed to offering developers an extraordinary tool for pair programming via the WebRTC protocol called Teletype. The developers experience can be enhanced even further with the use of a vast variety of addons and libraries. Although Teletype is an amazing tool, it still does not provide granular access controls to the admin, and it is impossible to share a running server through it. Although Teletype comes short in these specific features, it is currently one of the best free solutions out there. (Yegulalp 2020.)

Additionally, they are some studies that try to create similar environment as this thesis. For instance, the University of Macedonia in Greece, has conducted a study called," Distributed Pair Programming Using Collaboration Scripts: An Educational System and Initial Results" (Tsompanoudi, & Satratzemi & Xinogalos 2015), which analyses in detail the main weaknesses in education from similar distributed pair programming systems and consecutively discusses their own solution which utilizes collaboration scripts. Although these solutions are excellent, they heavily rely in non-open-source resources which breaches the basic requirements of this thesis.

Finally, the most complete solution in the market for pair programming originates from the big tech-giant, Microsoft. Microsoft has created an exclusive feature for their extremely popular IDE (Visual Studio Code) called Live Share. Live Share provides an excellent overall standard for pair programming as your teammates grant access to an instance of your code and they can freely use their own tool configuration to make changes to your environment (Yegulalp 2020).

In conclusion, Live Share from Visual Studio Code is an excellent free solution, but it still lacks benefits that are granted from entering a docker container while using all the great tools provided from Visual Studio Code in it. Docker containers

enhance the availability and configuration options greatly over Live Share although they have very similar usability overall as they both use a common base.

3.6 Knowledge Sources

The knowledge base of this thesis ranges from academic papers, online tutorials for software configuration and educational books which contain technical knowledge. Information has been obtained as well from big cooperation websites which provide fundamental information for their products which are used thoroughly in this Thesis (Docker, IBM, Atom).

In this thesis, there are also references to studies which explain the tremendous importance of distributed pair programming in education in order to justify the great importance of creating new methodologies for such solutions. More specifically, the Department of Computer Science in the University of North Carolina proved that both the metrics used for productivity and quality of code gave slightly better result for the distributed pair programming teams and it showcased an undoubtable positive difference in team communication and student satisfaction. (Stotts , Williams , Nagappan , Baheti , Jen & Jackson 2003, 2-3).

4 ETHICAL FOUNDATION, RELIABILITY AND MATERIAL ACQUISITION

4.1 Ethics Examination

Ethics should always be the core driving force for every decision made within and outside of the business world. Although it could be argued which actions are considered ethical, ethical behavior is described by a given set of widely agreed standards which are almost universal. Even though ethics are a subset of philosophy which categorizes actions according to their wrongness and rightness (Awari & Warjurkar 2022, 1-3,), this thesis tries to objectively be, as universally ethical as it can, by carefully examining righteousness in every step of the way.

In the IT world, there are many different dimensions to ethics to consider in every action taken in the process of creating a software solution. Ethics are important to be examined not only from the IT product provider perspective, but also from the end users desire and skills. This technically means that the creator of a solution is responsible for creating an environment which encourages ethically approved actions within it. This can be achieved when improper usage of computer services is minimized, inappropriate knowledge sharing is circumvented, and software piracy is utterly discouraged. (Awari & Warjurkar 2022, 22-23.) All these values were carefully considered through all decision making thought-out the thesis.

This thesis has managed to follow all the main ethical standards by the exclusive use of open-source software solutions. Open source technically means that the licenses of the software do not restrict parties from selling or giving away the software and conclusively does not require any fees or royalties in order to be used. (Debian Free Software Guidelines (DFSG) 2007)

4.2 Reliability

When referring to reliability, we refer to the ability to reproduce the same results multiple times using the exact same procedure (Dudovskiy 2016). In the context of this thesis, reliability is thoroughly examined, as a nonexpert user is trying to follow the provided guide in order to achieve the exact same result as promised.

Factors that greatly affect the reliability in this thesis concerns mainly the operating system in which the solution takes place. This guide was made with a “Microsoft Windows 10 Home” license and the nonexpert user will try to recreate the solution in a different variation of “Microsoft Windows 10”. Unfortunately, testing the effectiveness of this solution in many different operating systems is beyond the scope of this research.

4.2.1 Inspecting Reliability

In order to test the reliability, the instruction clarity and the effectiveness of the student guide, a non-expert user was tasked to follow it and try to achieve the end result. The outcome of this process would have modified the final form of the student guide if the instructions were proven to be misleading, unclear, or inaccurate. Furthermore, it is acknowledged that one non-expert user is definitely not a proper test sample, but it enough proof for receiving feedback for possible necessary guide alterations.

The non-expert user successfully completed the guide without facing any unexpected or unreported errors. Furthermore, the user was amazed from the detailed instruction descriptions and from the helpful visual representations that the pictures provided. It was confirmed that the omission of thorough instructions for basic software download and installation, did not prove to be a stumble for the user’s experience. It must be noted that the user’s computer complied with all the needed system requirements of the guide, which utterly affected the overall user experience and avoided any additional possible complications.

4.3 Material Acquisition

There is a vast variety of different material sources used for the purpose of this thesis. All the information gathered contributed to the creation of two categories of information, the theoretical and the practical. The theoretical part is formed from the collective material which is used in order to justify the topic choice, compare other industrial solutions with the one produced in this thesis, showcase other studies findings, and illustrate the possible benefits that can be achieved

with the utilization of the provided solution. In contrast, the practical part provides a complete guide on how to implement the solution and uses theory as well as many different guides in order to collectively construct the desired end product.

5 CONCLUSION

This thesis objective was to create a distributed pair programming environment for education purposes inside a container in order to concentrate all the student logs into one common environment. This goal was achieved via the use of Docker for the container creation, Visual Studio Code as the designated IDE and OpenSSH as the connectivity tool. Furthermore, this environment offers much more than just concentrated log collection, as it provides students with a “ready to code”, geographically independent, scalable, and reliable virtual coding workspace. The teacher on the other hand, can create multiple different containers and add specific student pairs in them. As a result, the teacher can always track the progress of the student pair by reading the container logs as well as examining and testing the student’s composed code inside each container.
In contemplation of ensuring the clarity of the guide, a non-expert user tested setting up the student side of the guide. The user effortlessly achieved the desired outcome, which consecutively showcases the simplicity of the whole guided process. Additionally, other commercially used solutions, which would have had similar outcomes as this thesis, were compared in order to showcase all the provided strengths of the given solution.

Although the thesis purpose was achieved, they are still many upgrades that can be done in order to further enhance the accomplished outcome. Some of the most obvious enhancements will be described in the following section and they are highly encouraged to be implemented, as they can provide simplification, more customization as well as increased security for the hosts environment.

5.1 Future Improvements

First of all, the guides, teacher guide and student guide, are specifically made for updated Microsoft Windows computers only. This can be a struggle as many students could have macOS, Linux or other operation systems. The key turning point in the guides, where different operation systems create a huge change, has to do specifically with the SSH client. The first upgrade would be to try and find a

universal SSH client which complies with all (or most) operation systems, so the guide becomes more flexible and extensive.

Another important change would be to setup a key pair authentication system which would greatly increase security when the computers try to make a connection. Furthermore, the guide could introduce a new chapter which educates the students in connectivity automation, in order to avoid inputting the SSH host credential and passwords every time they try to connect to the host in Visual Studio Code.

Finally, the teacher could configure Visual Studio Code accordingly, so when a student connects to the container, they are automatically restricted of accessing all the files on the host system and they are guided directly to the desired directory.

BIBLIOGRAPHY

Aitken, A. & Ilango V. 2013. A Comparative Analysis of Traditional Software Engineering and Agile Software Development. Accessed 29 November 2022 A Comparative Analysis of Traditional Software Engineering and Agile Software Development | IEEE Conference Publication | IEEE Xplore

Atlassian, 2022. The Agile Coach. Accessed 26 November 2022 https://www.atlassian.com/agile#:~:text=Agile%20is%20an%20iterative%20approach,small%2C%20but%20consumable%2C%20increments.

Atom, 2022. Accessed 26 November 2022. https://atom.io/

Awari G. K. & Warjurkar S. V. 2022. Ethics in Information Technology: A Practical Guide (1st ed.). CRC Press. 1-2

Awari G. K. & Warjurkar S. V. 2022. Ethics in Information Technology: A Practical Guide (1st ed.). CRC Press. 22-23

Chaturvedi, V. 2022. What Is Docker & Docker Container ? A Deep Dive Into Docker! What Is Docker & Docker Container | A Deep Dive Into Docker | Edureka. Accessed 10 November 2022 What Is Docker & Docker Container | A Deep Dive Into Docker | Edureka

Codiska, 2022. Why Programming is Important, Today, and for the Future? Accessed 26 November 2022 https://codiska.com/articles/why-programming-is-important-today-and-for-the-future/

Crouch S. 2021.Defending your code against dependency problems. Accessed 11 November 2022. https://software.ac.uk/resources/guides/defending-your-code-against-dependency-problems

Debian Free Software Guidelines (DFSG). 2007. Accessed 10 November 2022. https://opensource.org/osd

Debian Social Contract. 2022. Accessed 29 November 2022. https://www.debian.org/social_contract

Dudovskiy J. 2016. The Ultimate Guide to Writing a Dissertation in Business Studies: a step by step assistance

Enago. 2021. Writing the Research Methodology Section of Your Thesis. Accessed 10 November 2022 https://www.enago.com/thesis-editing/blog/writing-research-methodology-section-thesis

Garber, J. 2020. Practical Pair Programming. A Book Apart. 23-24

Garber, J. 2020. Practical Pair Programming. A Book Apart. 23-31

Gillis, A. S. 2021. What is pair programming? Accessed 26 November 2022 https://www.techtarget.com/searchsoftwarequality/definition/Pair-

programming#:~:text=Pair%20programming%20is%20an%20Agile,code%20and%20test%20user%20stories.

IBM Cloud Education. 2021. Docker. Accessed 10 November 2022 https://www.ibm.com/ae-en/cloud/learn/docker

Kalcheva, K. 2022. Remote Work Statistics in 2022 and How they will Influence the Workplace in 2023. Accessed 10 November 2022 https://squaretalk.com/remote-work-statistics/#:~:text=In%202022%2C%2016%25%20of%20companies,remote%20working%20by%20the%20employees%3F

Kranz, G. 2020. Definition Metadata. Accessed 29 November 2022 https://www.techtarget.com/whatis/definition/metadata

Lucidchart 2022. What do you want to do with flowcharts? Accessed 25 November 2022 https://www.lucidchart.com/pages/what-is-a-flowchart-tutorial

Narang, M. 2022. Why Use Docker? Advantages and Benefits. Accessed 29 November 2022 https://www.knowledgehut.com/blog/devops/why-use-docker

Neuman, W., 2003, Social Research Methods: Qualitative and Quantitative Approaches, 5-th Edition, Pearson Education. https://pressbooks.bccampus.ca/jibcresearchmethods/chapter/3-5-quantitative-quantitative-mixed-methods-research-approaches/

Petersen, D. 2022. Cloud Coding: Remote Pairing using Docker. Accessed 10 November 2022 https://www.ctl.io/developers/blog/post/cloud-coding-docker-remote-pairing

RedHat. 2019. What is an IDE? Accessed 11 November 2022. https://www.redhat.com/en/topics/middleware/what-is-ide

Sheppard V. 2020. 3.5 Quantitative, Qualitative, & Mixed Methods. ResearchApproaches. Accessed 11 November 2022.

Shirinkin, K. 2020. Dockerless, part 1: Which tools to replace Docker with and why. Accessed 29 November 2022 https://mkdev.me/posts/dockerless-part-1-which-tools-to-replace-docker-with-and-why

Singh, P. & Andrews, K. 2022. An Introduction to Pair Programming. Accessed 10 November 2022 https://www.qentelli.com/thought-leadership/insights/introduction-pair-programming

Stotts D., Williams L., Nagappan N., Baheti P. & Jen D. & Jackson A. 2003. Virtual Teaming: Experiments and Experiences with Distributed Pair Programming. 2-3.

Tsompanoudi D. & Satratzemi M. & Xinogalos S. 2015. Distributed Pair Programming Using Collaboration Scripts: An Educational System and Initial Results

Tsompanoudi D. & Satratzemi M. & Xinogalos S. 2022. Distributed Pair Programming in Higher Education: A Systematic Literature Review. Accessed 11 November 2022. https://journals.sagepub.com/doi/abs/10.1177/07356331221122884?journalCode=jeca

Tsompanoudi, D. & Satratzemi, M. 2014. A Web-Based Authoring Tool for Scripting Distributed Pair Programming. 259-263

Vardalos, S. 2017. An introduction to logging for programmers. Accessed 25 November 2022 https://www.freecodecamp.org/news/you-should-have-better-logging-now-fbab2f667fac/

Wang, Y. & Luo S. & Dou W. 2010. Analysis and Design of Distributed Pair Programming System. Accessed 29 November 2022 https://www.scirp.org/html/6-8701059_2504.htm

Wells, D. 1999. Pair Programming. Accessed 10 November 2022 http://www.extremeprogramming.org/rules/pair.html.

Yegulalp, S. 2020. 7 tools and services for real-time collaborative coding. Accessed 26 November 2022 https://www.proquest.com/docview/2385081516?accountid=27297&forcedol=true#

APPENDICES

This section contains the complete guide for both setting up a distributed pair programming environment for both the student and the teacher. The attached guide was specifically drafted for the purpose of this Thesis.

The teacher guide contains all the steps required to setup a docker in Visual Studio Code, build a container inside the docker, setup the SSH (Secure Shell Protocol) server, as well as some customization option for setting up the coding environment inside the container. On the other hand, the student guide is a much simpler guide, as it only contains the steps for connecting to the SSH server via the OpenSSH client setup, along with the steps for setting up the VS Code accordingly in order to achieve the connection to the prebuilt container.

Appendix 1. Complete Teacher and Student Guide

LAPIN AMK
Lapland University of Applied Sciences

COMPLETE TEACHER AND STUDENT GUIDE

Distributed Pair Programming Solution Using Docker for Educational Purposes

Zohios, Andreas

Bachelor´s Thesis
Degree Programme in Business Information Technology
Bachelor of Business Administration

2022

CONTENTS

1 INTRODUCTION 28
1.1 System Requirements 29
2 TEACHERS GUIDE 30
2.1 VS CODE and Container Creation Setup 30
2.2 System Requirements Check 35
2.3 SSH Server Setup 36
2.4 Credentials Identification 37
2.5 Enabling Real-time Coding 38
3 STUDENT GUIDE 39
3.1 VS CODE Setup 39
3.2 System Requirements Check 40
3.3 OpenSSH Client Setup 41
3.4 Accessing the Container 42
3.5 Enabling Real-time Coding 43
BIBLIOGRAPHY 44

1 INTRODUCTION

This section of the Thesis serves as the complete guides that needs to be followed in order for the distributed pair programming system to be built. They are two main subsections in this guide, as they are different setup steps that need to be followed from the teacher and the students. The teacher guide contains all the steps required to setup a docker in Visual Studio Code, build a container inside the docker, setup the SSH (Secure Shell Protocol) server, as well as some customization option for setting up the coding environment inside the container. On the other hand, the student guide is a much simpler guide, as it only contains the steps for connecting to the SSH server via the OpenSSH client setup, along with the steps for setting up the VS Code accordingly in order to achieve the connection to the prebuilt container.

The following guides does not require any initial knowledge or expertise in order to get started in neither the teachers nor the student's perspective. In contrast, this Guide, assumes that the reader is capable of downloading and installing basic software on their computers, and as a result, there is no detailed instructions on those steps, as they are considered basic knowledge. All the software that needs to be installed on specific steps, contain links to the official download pages to minimize possible confusions and provide certainty to the user's actions.

It is important to mention that both guides were created in Microsoft Windows 10, and as a result, they are definitely variations according to the operating system of the chosen device. As a final notice, the teacher should follow the guide in a different password protected windows account, than the one personally used, for increased security.

1.1 System Requirements

- A device running at least Windows Server 2019 or Windows 10 (build 1809).
- PowerShell 5.1 or later.
- An account that is a member of the built-in Administrators group.

2 TEACHERS GUIDE

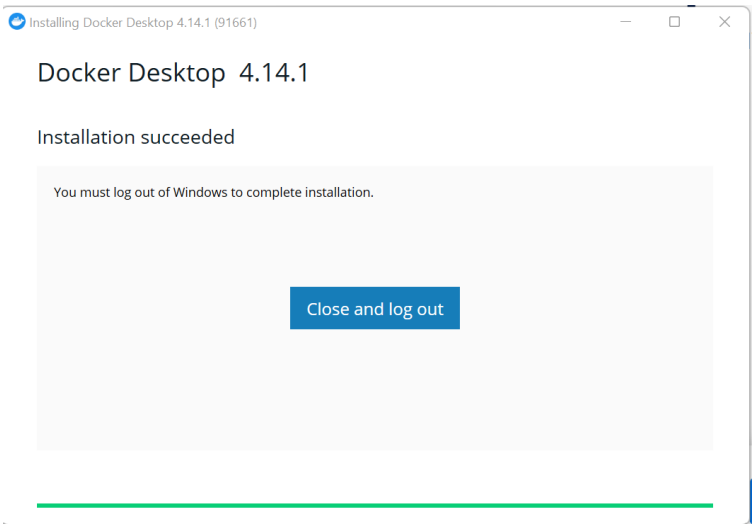
2.1 VS CODE and Container Creation Setup

First of all, the teacher needs to have Visual Studio Code installed in the computer, as it is the preferred IDE (Integrated Development Environment) used for this solution. In case Visual Studio Code is missing from the computer, it can be downloaded and installed throught the official website ”Link 1”.

Link 1: https://code.visualstudio.com/

Once the IDE is ready, we need to install Docker on the computer. Docker Desktop can be downloaded by using the following link ”Link 2”. Currently, the newest Docker Desktop version that can be downloaded is Docker Desktop 4.14.1 and that is the version used in this guide. The installation is very simple but takes some time. After the installation is completed, presh ”Close and log out” PICTURE 1. Log in back into windows and confirm that the installation was successful by opening up the Docker Desktop. Make sure to accept the terms and agreements in the pop up window.

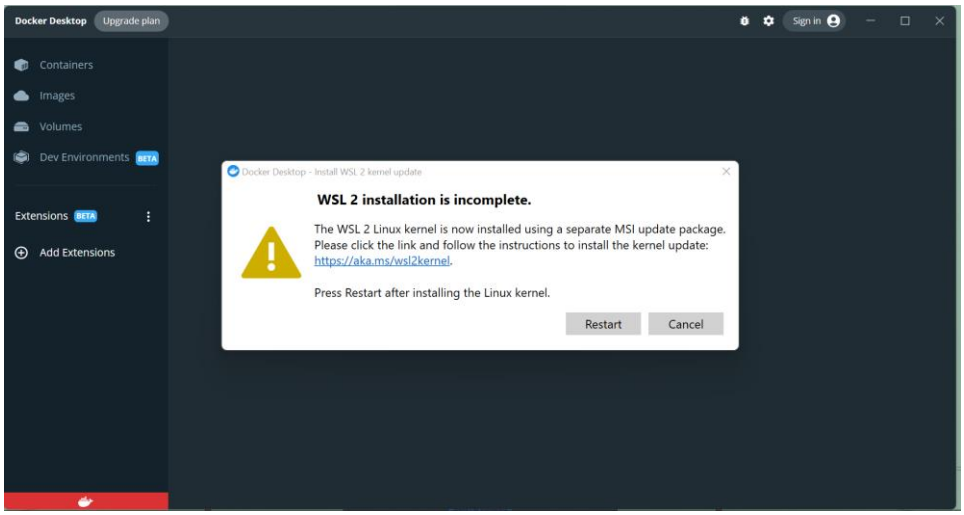
Link 2: https://www.docker.com/products/docker-desktop/



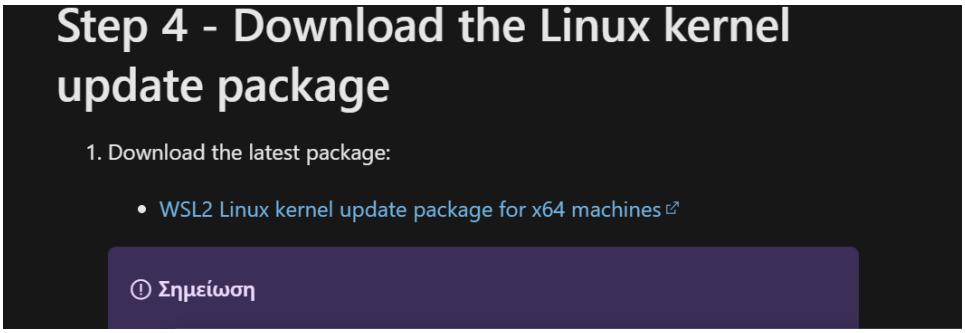
PICTURE 1

If the software was installed successfully, the following error appears PICTURE 2. Make sure to click the link provided in the error message ”Link 3” which will guide you to a webpage to download the WSL2 Linux Kernel update package PICTURE 3. Download and install the package. After the installation, reboot the computer and re-open Docker Desktop to ensure that Docker is running successfully.

Link 3: https://learn.microsoft.com/en-us/windows/wsl/install-manual#step-4---download-the-linux-kernel-update-package



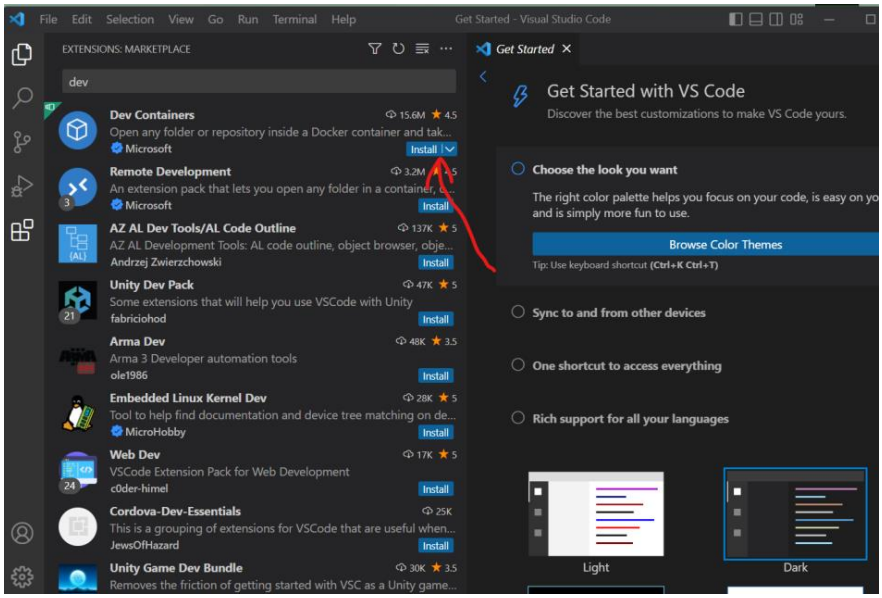
PICTURE 2



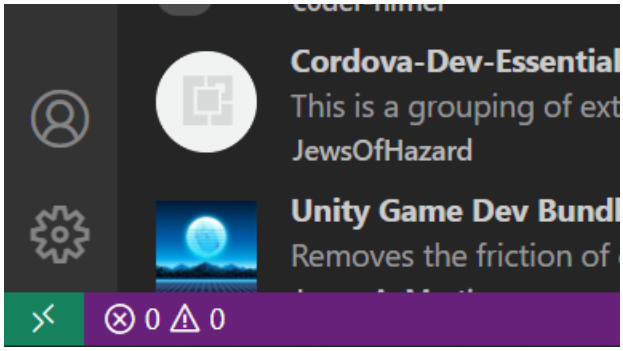
PICTURE 3

The next step is to open Visual Studio Code and go to the extension tab. There, we search and install the ” Dev Containers” plugin PICTURE 4. You can assure

that the installation has been successful by checking the existance of the green symbol in the bottom left corner in VS CODE PICTURE 5.



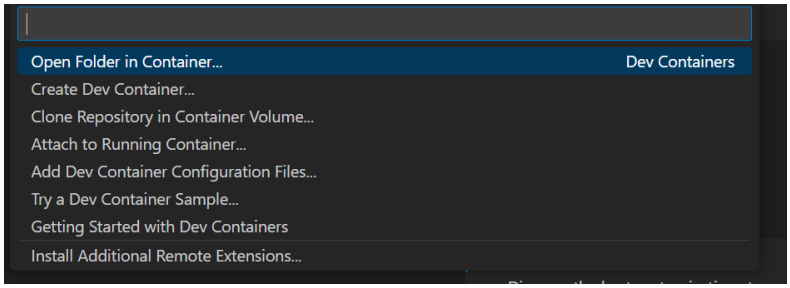
PICTURE 4



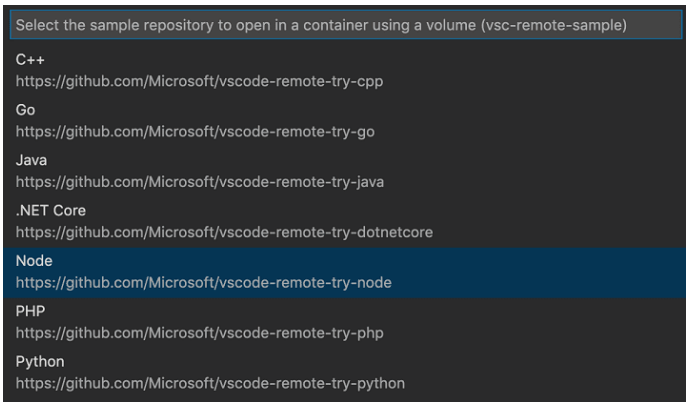
PICTURE 5

Now the system is ready to create a Docker containers. For the purpose on this Guide, a ”Python 3” container running Anaconda will be executed. In order to create this container, click the green symbol on the bottom left and choose the option ”Create Dev Container” PICTURE 7. From the drop out menu, choose Python PICTURE 6. The container creation process takes a couple of minutes so there is no need to worry. Once the container has been made, chose Anaconda Python 3 and then chose once again anaconda and presh ok PICTURE 8. Again, there is some waiting time as shown in PICTURE 9. A pop up window will appear and the ” Yes, I trust the authors” button must be pressed PICTURE 10. After the

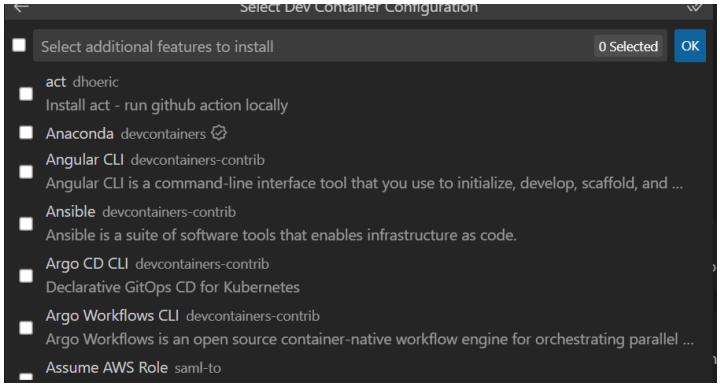
process is finished the container is ready to be used and the green bar should look like in the picture PICTURE 11.



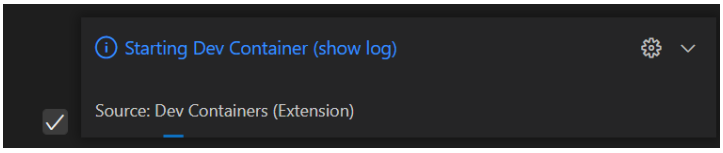
PICTURE 6



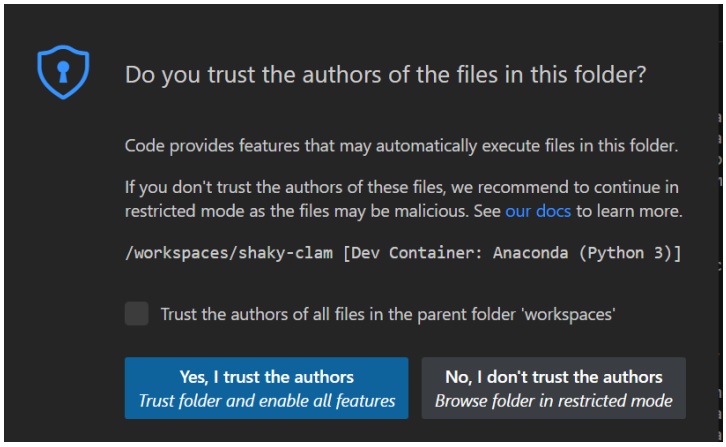
PICTURE 7



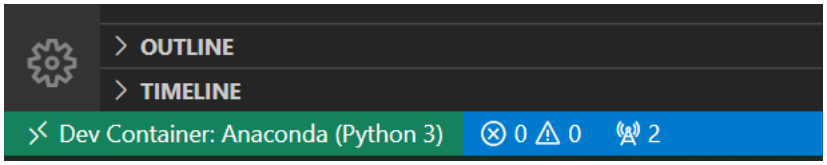
PICTURE 8



PICTURE 9



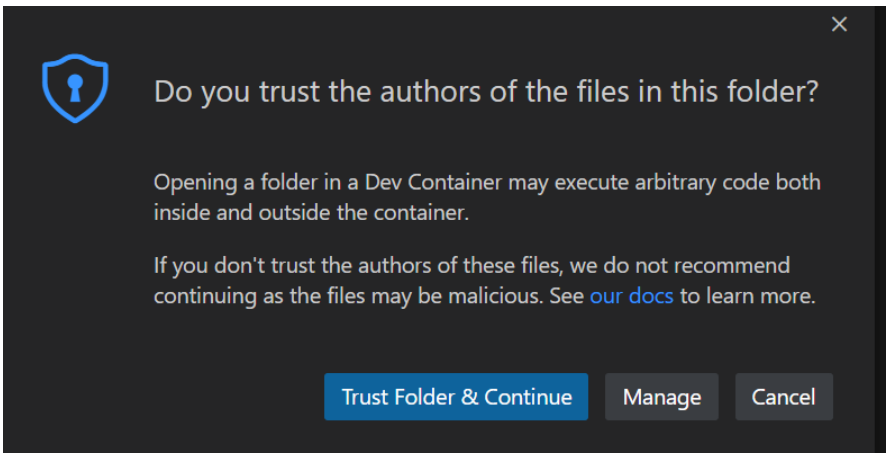
PICTURE 10



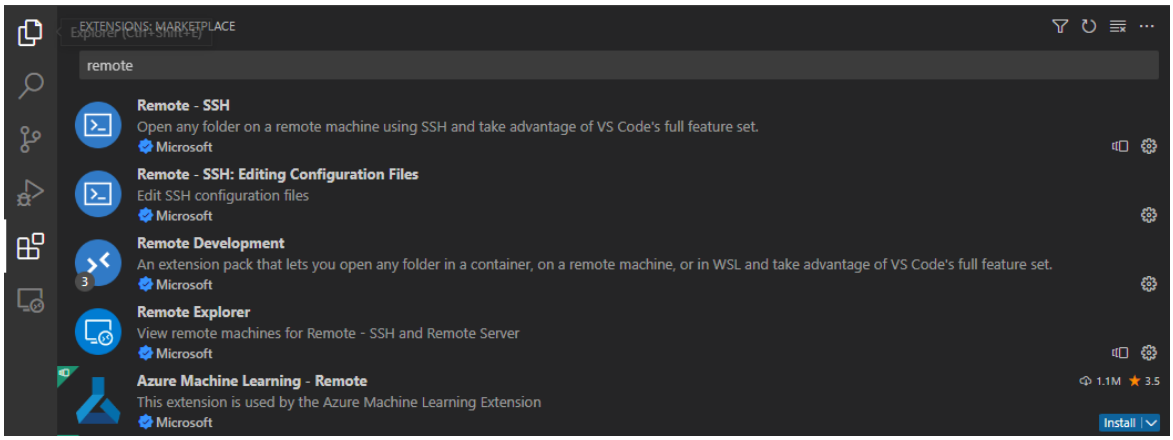
PICTURE 11

Now, a folder should be opened (which would contain our python files) inside the container. For the purpose of testing, a folder must be created and contain a small (.py) file to open inside the container. Once the file is created, click on the green button on the bottom left and select ”Open Folder in Container...” and select the new folder that needs to be opened in the container. From the list shown, select once again Anaconda and then for a final time, anaconda and ok like shown previously. Another pop up window will appear and the ” Press trust foldier and continue ” button should be selected PICTURE 12. The container should be now successfully running with the code selected inside of it.

Finally, an extension for Visual Studio Code should be installed which is called ” Remote Development” . In order to install the extension, we repeat the same proccess as previously with ” Dev Containers ” PICTURE 13.



PICTURE 12



PICTURE 13

2.2 System Requirements Check

Now that the container is running, the SSH server should be configured. First of all, the requirements mentioned in section 1.1 should be met. In order to ensure that they are met, the following instructions must be followed.

First, the Windows version on the chosen computer should be newer than the build 1809. In order to check that, right click on windows start button and select system. There must be clearly stated the OS build of the machine. Next, search “PowerShell” in the windows search bar and right click to open “Windows PowerShell” with administration rights. By doing so, the administrator requirement has been checked and there is only a need to check the PowerShell version. In order to do that, run the following command COMMAND1. The

version of the PowerShell must be over 5.1. If the requirements are met, the creation of the OpenSSH server can begin.

```
$PSVersionTable.PSVersion
```

COMMAND 1

2.3 SSH Server Setup

The OpenSSH client and OpenSSH server should be checked in case they are already in the system. This can be achieved by writing the following command. “COMMAND 2”. In case both client and server are missing, the following commands should be used to install them accordingly COMMAND 3, COMMAND 4.

```
Get-WindowsCapability -Online | Where-Object Name -like 'OpenSSH*'
```

COMMAND 2

```
Add-WindowsCapability -Online -Name OpenSSH.Client~~~~0.0.1.0
```

COMMAND 3

```
Add-WindowsCapability -Online -Name OpenSSH.Server~~~~0.0.1.0
```

COMMAND 4

Once this process is finished, there should be shown that the service is currently online. In order to start the service, the following command should be used ”COMMAND 5”. Then, in order to set the SSH server to automatically open with the windows startup so there is not a constant need to run the previous command, the following command could be used if desired. ”COMMAND 6”. Finally, a firewall rule should be created to enable SSH connection to communicate with other devices. This can be achived with the following command COMMAND 7.

```
Start-Service sshd
```

COMMAND 5

```
Set-Service -Name sshd -StartupType 'Automatic'
```

COMMAND 6

```
if (!(Get-NetFirewallRule -Name "OpenSSH-Server-In-TCP" -ErrorAction SilentlyContinue | Select-Object
Name, Enabled)) {
   Write-Output "Firewall Rule 'OpenSSH-Server-In-TCP' does not exist, creating it..."
   New-NetFirewallRule -Name 'OpenSSH-Server-In-TCP' -DisplayName 'OpenSSH Server (sshd)' -Enabled
True -Direction Inbound -Protocol TCP -Action Allow -LocalPort 22
} else {
   Write-Output "Firewall rule 'OpenSSH-Server-In-TCP' has been created and exists."
}
```

COMMAND 7

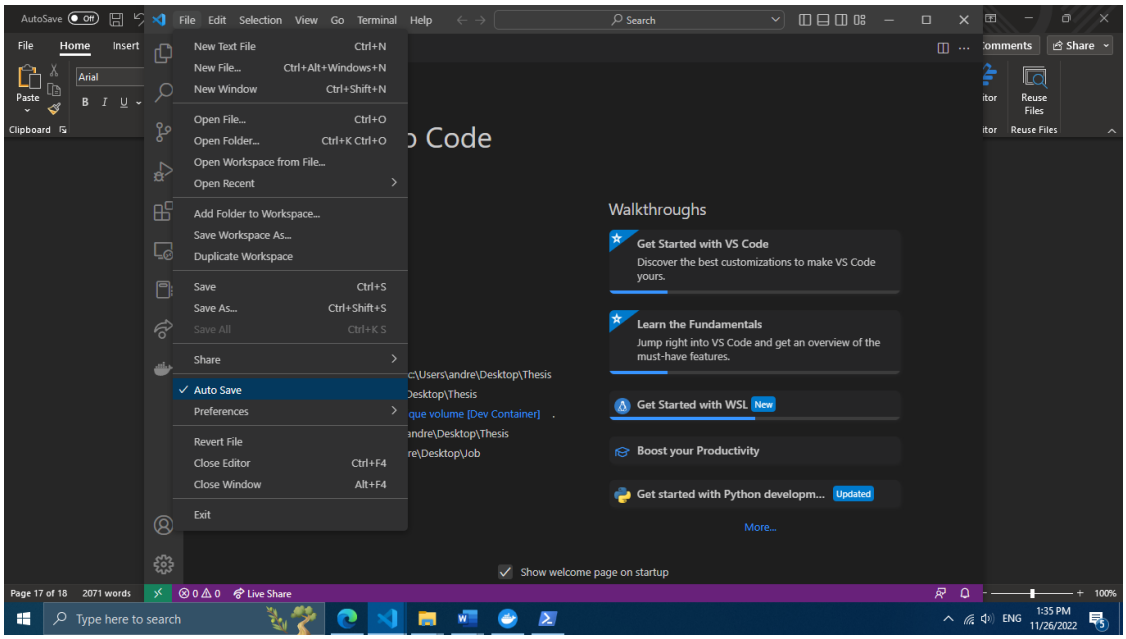
Now the whole process in finished as the container and the OpenSSH server are both successfully running.

2.4 Credentials Identification

Finally, the SSH credentials must be clarified so they can be provided to the students who request access to the server. They are three credentials needed, the domain, the nickname and the IP address of the computer. The nickname can be seen by going to ” This PC ”, ”Local Disk C”, ”Users” (This location could varry according to the drive used to store Windows OS). Next, in order to find the domain, search on Windows ” System Information ” . The domain can be seen next to ”System Name”. Finally, the IP address can be identified by going back to Powercell and writting ”ipconfig”. The desired IP is shown next to Ipv4 Address, under the according connection category that the computer uses to connect to the internet.

2.5 Enabling Real-time Coding

When changes occur in a coding file, they can only be shown to the other users once the modifications have been saved. In order to approach the feel of real-time coding for both ends, automatic saving is recommended to be enabled in Visual Studio Code. This can be achieved by clicking on the file tab in VS CODE and clicking the “Autosave” mode like shown in PICTURE 14.



PICTURE 14

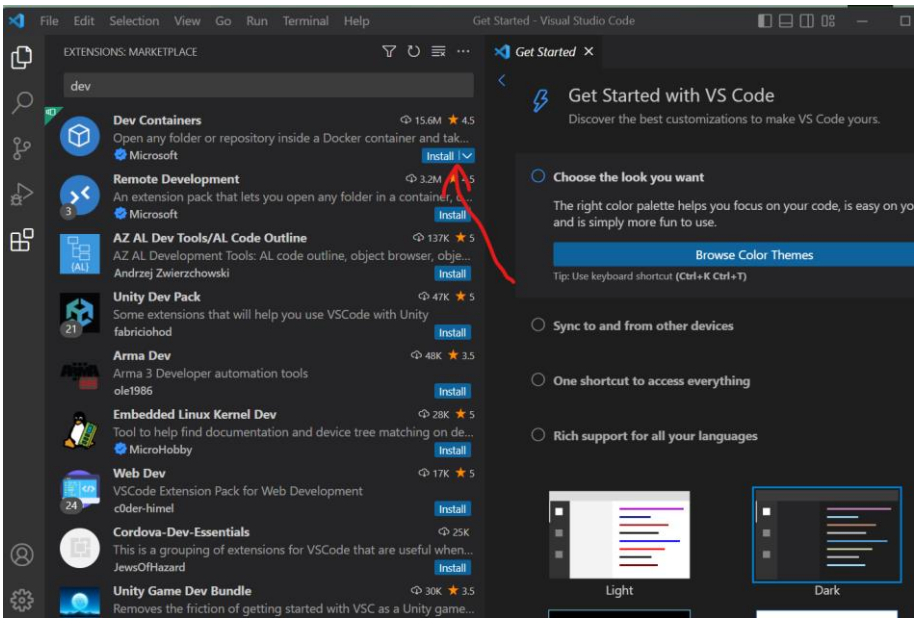
3 STUDENT GUIDE

3.1 VS CODE Setup

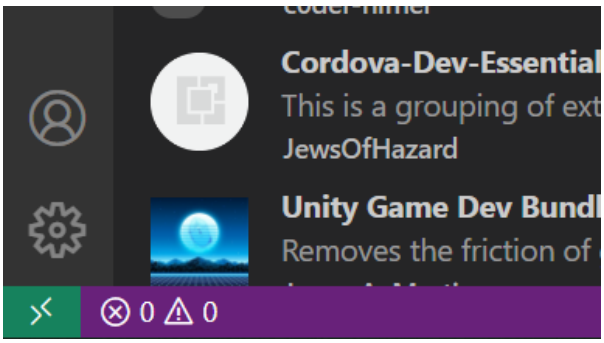
First of all, the student needs to have Visual Studio Code installed in the computer, as it is the preferred IDE (Integrated Development Environment) used for this solution. In case Visual Studio Code is missing from the computer, it can be downloaded and installed throught the official website ”Link 1”.

Link 1: https://code.visualstudio.com/

The next step is to open Visual Studio Code and go to the extension tab. There, search and install the ” Dev Containers” plugin PICTURE 1. You can assure that the installation has been successful by checking the existance of the green symbol in the bottom left corner PICTURE 2.



PICTURE 1



PICTURE 2

3.2 System Requirements Check

All the requirements mentioned in section 1.1 should be met. In order to ensure that they are met, the following instructions must be followed.

First, the Windows version on the chosen computer should be newer than the build 1809. In order to check that, right click on windows start button and select system. There must be clearly stated the OS build of the machine. Next, search “PowerShell” in the windows search bar and right click to open “Windows PowerShell” with administration rights. By doing so, the administrator requirement has been checked and there is only a need to check the PowerShell version. In order to do that, run the following command COMMAND 1. The version of the PowerShell must be over 5.1. If the requirements are met, the process to connect to the OpenSSH server can begin.

```
$PSVersionTable.PSVersion
```

COMMAND 1

3.3 OpenSSH Client Setup

The OpenSSH client should be checked in case it is already in the system. This can be achieved by writing the following command inside PowerShell which is opened in administrator rights COMMAND 2. In case that the client is missing, the following commands should be used to install it COMMAND 3.

```
Get-WindowsCapability -Online | Where-Object Name -like 'OpenSSH*'
```

COMMAND 2

```
Add-WindowsCapability -Online -Name OpenSSH.Client~~~~0.0.1.0
```

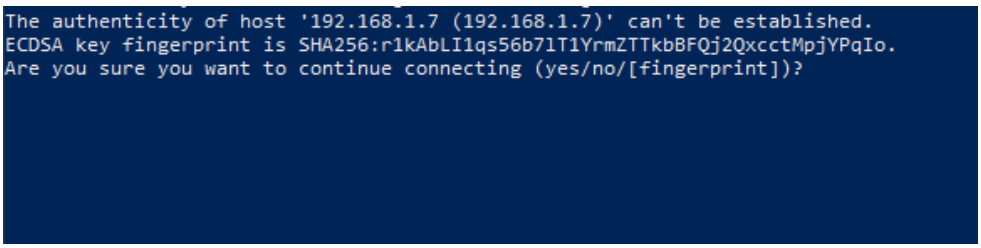
COMMAND 3

Once the installation is finished, there should be shown that the service is currently online.

Now, the system is ready to connect with another computer. That can be achieved by running the following command COMMAND 4, where the domain, username and servername would all be provided from the teacher. If the credentials are written correctly, there should be the following output PICTURE 3 in which it should be typed “yes “. Finally, the teacher should also provide a password for accessing his computer. A new terminal opens up if the connection has been established.

```
ssh domain\username@servername
```

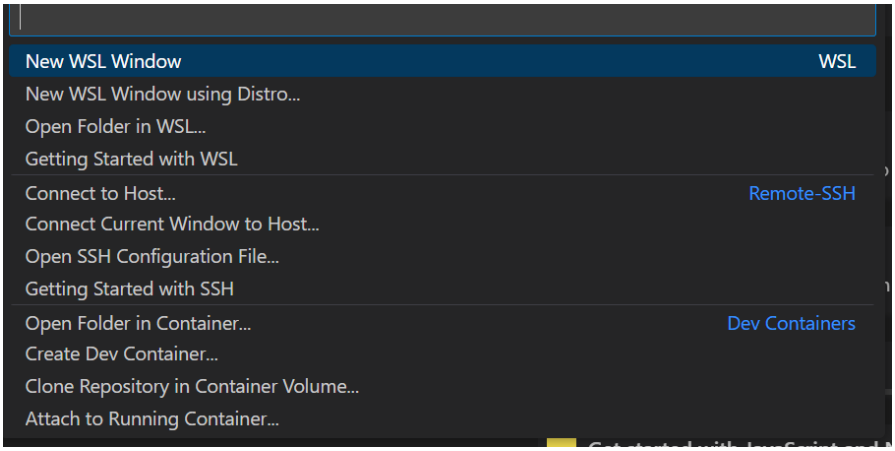
COMMAND 4



PICTURE 3

3.4 Accessing the Container

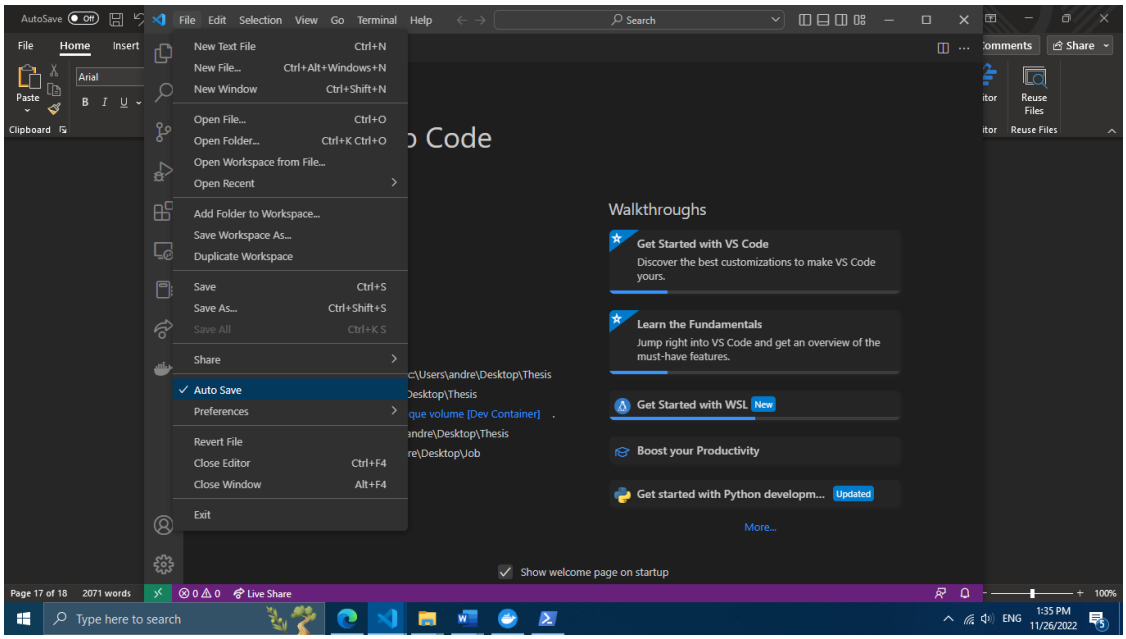
In order to access the container, go back to VS CODE and press the green bottom left button. Then, press “Connect to Host” PICTURE 4 and type once again “username@servername” with the same credentials from previously. There is a chance, that the computer does not identify the Operating system of the container and it should be chosen manually to “Windows”. The final step is to rewrite the password which was used previously. Now, the access has been granted to use the container made from the host.



PICTURE 4

3.5 Enabling Real-time Coding

When changes occur in a coding file, they can only be shown to the other users once the modifications have been saved. In order to approach the feel of real-time coding for both ends, automatic saving is recommended to be enabled in Visual Studio Code. This can be achieved by clicking on the file tab in VS CODE and clicking the “Autosave” mode like shown in PICTURE 5.



PICTURE 5

BIBLIOGRAPHY

Throughout the Guide, a combination of all the links provided bellow were used. All links provided were accessed in 24/11/2022.

https://code.visualstudio.com/docs/devcontainers/tutorial

https://code.visualstudio.com/docs/devcontainers/containers

https://code.visualstudio.com/docs/remote/ssh#_ssh-host-setup

https://learn.microsoft.com/en-us/windows-server/administration/openssh/openssh_install_firstuse?tabs=gui