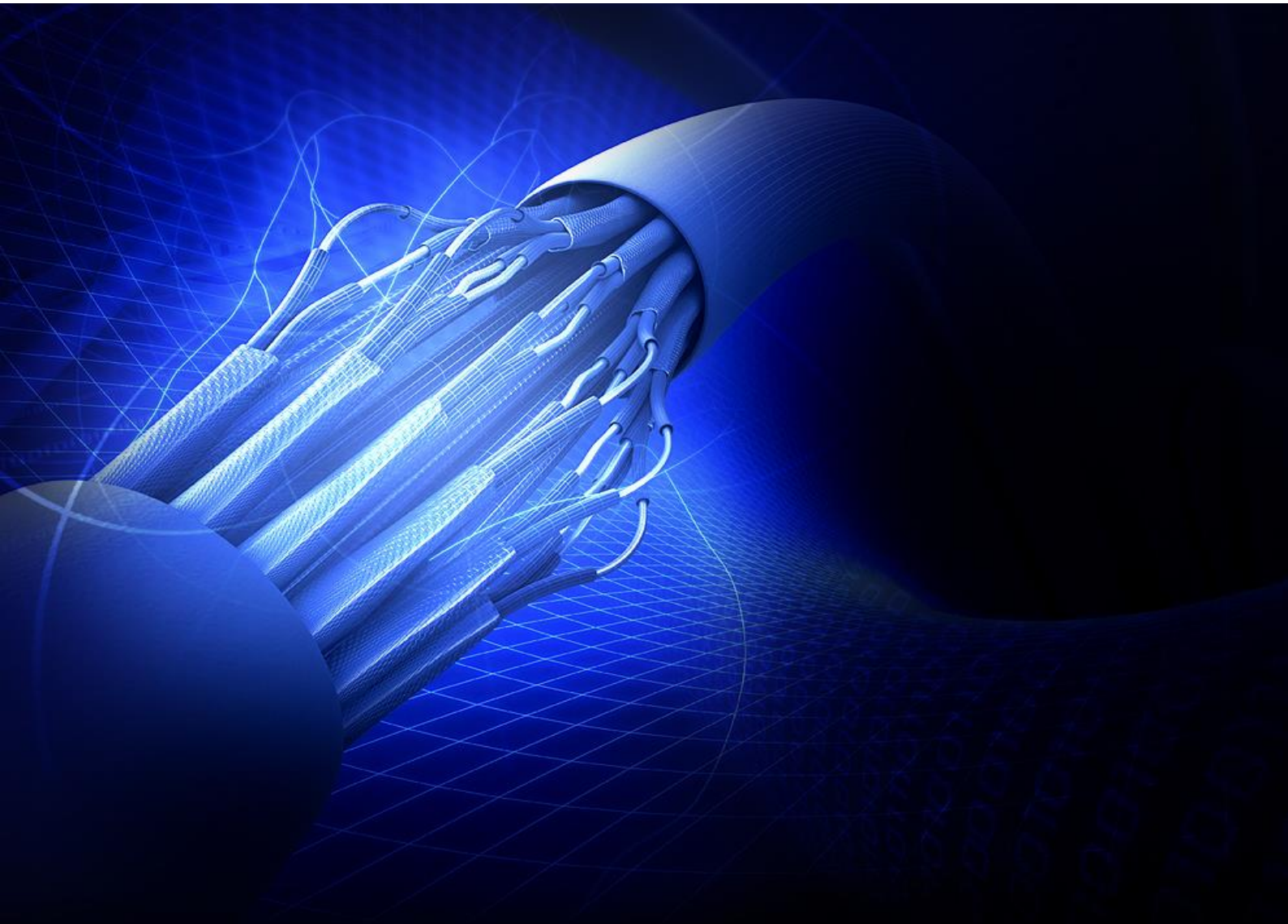


AWS Container Security Project Specification & Plan

Student Name: David Williams
Project Supervisor: Hisain Elshaafi
Student ID: C00263768

Course: Cybersecurity & IT Security
4th Year

Abstract

The project is about AWS container Security. I was planning on making a Security Information & Event Management (SIEM) tool with a focus on network monitoring, intrusion detection, and prevention using a Raspberry Pi cluster. The project shifted its focus towards AWS Cloud Container Security was more intriguing. This decision was driven by the growing significance of cloud and container technologies. Container security brings advantages for both individuals and companies, providing consistent environments, efficient resource utilization, and rapid prototyping for individuals. It also enables application portability, microservices scalability, and streamlined DevOps practices for companies.

Contents

Introduction The Specification

My original plan for this project was originally to build a Security Information & Event Management (SIEM) tool. It would encompass features like network monitoring, intrusion detection, and prevention. All this would have been done on my Home PC while using a raspberry pi cluster. Although both projects share certain similarities, the decision to shift to a focused AWS cloud-based container environment project.

Why Use Container Security and who is it for

Why use container security:

1. For Individuals:
    - Isolation and Consistency: Containers are very practical solutions that offer much better consistent environments for applications ensuring they run across different systems.
    - Efficient Resource usage: While virtual machines offer isolation, cloud containers offer a better solution being that VM's run on its own separate operating system. While cloud containers provide lightweight form of isolation, sharing the host OS kernel. This means less resources are used, making it faster and more efficient in comparison.
    - Rapid Prototyping: Experimenting with container environments makes it easier to use different applications or configurations without affecting the host system.
2. For Companies:
    - Application Portability: From a development perspective, container applications are portable across different environments which are beneficial to companies who may with lots of different software's.
    - Microservices and scalability: Scaling Virtual machines is not as efficient as cloud containers when it comes to scaling and managing individual components of each application.
    - DevOps: Streamlining the process of builds, testing, and deploying applications is a big part of DevOps and has a vital role in their industry.
3. Use Cases:
    - Microservices Architecture: Containers are used to deploy applications using microservices applications making them flexible and scalable.
    - Security Isolations: Containers provide an additional layer of isolation for applications ensuring the impact of the security breaches and issues being limited.
    - Continuous integration & Continuous deployment (CI & CD): DevOps make use of this through development, testing, and deployment.

- Hybrid and Multi-Cloud deployments: Organisations with diverse infrastructure operate under many tools, software's and applications which is perfect for containers. Containers can be deployed consistently across the on premise-environment allowing the teams to diversify their operations.

4. How it's used:
    - Container security is of critical importance for Cybersecurity and the owners of the Container environment. Having the tools monitor, detect and prevent potential vulnerabilities or threats withing the containerised environment. The application can be used for various tasks depending on what the organisation desires them for in their internal environment.

What is the project supposed to be?

The project is about Amazon Web Services Containers and cloud-based Securities. The focus is going to be on different technologies that will help define and ensure the success of a secure cloud based containerised environment that allows individuals, developers, and organisations to utilize the full potential of cloud-based technologies. Considering the fast-growing industry that cloud has become and the use of containers. The need to secure them is also just as important in the industry as cloud is used globally with a greater need and desire to move current or older infrastructure onto cloud.

This project report will introduce all the different software's, technologies, applications, and tools that can be used and will be surrounding Cloud containers especially Amazon Web Services (AWS). The importance of cloud and cloud containers is a fast-growing industry over the past 20 years (2013). The goal of this project is to provide security for the container's environment using various tools to ensure nothing can happen to whatever the legitimate user has in the container can be altered, deleted, or interacted with by any malicious actor. The environment will also need to be secured from illegitimate access using Multifactor authentication ensuring extra security.

The reason for going through with this project is the interest and importance it has in the cybersecurity industry and how widely used cloud and containers are used today. Cloud containers are used for a wide range of applications, ranging from:

1. Application Deployment and scalability
2. Microservices architecture
3. Virtual operating systems deployment
4. Continuous Integration/Continuous Deployment (CI/CD)
5. Isolation and security
6. DevOps Practices
7. Hybrid and multi-Cloud Deployment
8. Stateless Services
9. Container Orchestration
10. Resource Efficiency
11. Development and Testing Environments
12. Legacy Applications Modernization
13. Fast Deployment and Scaling

That is to name the many important aspects of cloud containers used in not only individual applications but also organisations. While AWS is the cloud leader today and is where the focus of this projects Containerization will be focused on. The different applications and software's that will help secure and be utilised to achieve the goal of the project.

Technologies involved?

To list some of the different tools that will be potentially used in this project but not all of them! to achieve the outcome I hope to expect will be. The amount used will be a maximum of five technologies and the rest are just to name some that do similar tasks to the main five I choose.

Docker	Kubernetes	Falco	Calico	CloudTrail
Azure Kubernetes Services (AKS)	Lambda	Oracle Cloud infrastructure Container for Kubernetes	Google Kubernetes Engine (GKE)	CloudWatch
Mirantis Kubernetes Engine	Kubernetes	IBM Cloud Kubernetes Service	DigitalOcean	Dynatrace
Logz.io	Better Stack	New Relix	Dynatrace	Elastic Container Service (ECS)
Elastic Kubernetes Service (EKS)	AWS Firelens	PythonAnywhere	Azure App Service	Google App Engine

This is a list of many of the tools, applications and software's that are out there that can be used to effectively develop this project. The focus will be on using technologies that I have no experience with before so focusing on AWS services instead of Azure is the direction to go from not only a researched perspective but also a learning experience.

The overall view of the different applications is to provide Intrusion detection systems (IDS), Intrusion prevention systems (IPS) while having logs of traffic in and out of the container environment while checking when the user logs in and out of the container. Most information provided suggests that docker is very secure and docker works with AWS as well as Kubernetes which makes it a very attractive application to use along with other tools.

Kubernetes is a well-known application that works very well with docker and AWS as well making it increasingly scalable if all three are used together. Many of the other tools have similar combinative benefits that can be used to help secure the environment such as Falco and Calico.

What I hope and want to achieve in this project

My hope: My hope is to learn as much as possible from this project, from cloud services to security, compliance's, policies, and the technology involved. As a student learning Cybersecurity, I find it fascinating and great to learn new technologies and tools even though there is a vast amount of them out there, taking the time to learn is always an achievement of its own.

Another hope is that I complete the project on time for every deadline set while also completing everything I set before me when it comes to the tasks that are involved in this project. From completing each iteration of application, technology, tool that will go into securing the AWS container but also any additional unforeseen application implemented into the project. I do plan to focus on and only on the tools I have chosen but as we all know, sometimes we can get ahead of the projects scope and want more. If I am behind then I will hope I provide enough to pass and secure my future as a Cybersecurity specialist, engineer, technician?

What I want: I want to be able to get far enough with the development of the project to learn enough and be able to use that knowledge for personal and professional goals. I want to take what I learn from this project and be able to put it to use in any company I work for but also if I can help others learn what I have and more! Then I think that will be one of the great things and feelings I can get as helping others is of great value and importance.

Discoveries

Through the vast range of research, I have put into Cloud containerization and security tools. There is much to consider when it comes to what to use and why to use it. To start with, what kind of container should be used and why is it considered the best container. There is also a need to consider the threat detection system, prevention, log router, managed service (EKS) and either Lambda or something similar which is a computer service that lets a user rune code without provisioning or managing the servers.

What will the project deliver?

The project will deliver a few things such as:

1. Core Functionality

   Integration with AWS: This is the foundation of the project which will include the security monitoring for containers using the tools provided.

   Real-time container security alerts: The system should develop to provide and generate real-time alerts based on security events determined within the environment.

   Customisable Policy Rules: Rules need to be in place to ensure security policies are followed and suited for the container environment.

   Integration with AWS log routers and analyzers: This will be crucial for container security through collecting and analysing the traffic.

   Runtime intrusion Detection and Prevention: Tools that are critical to the security for this project in real-time and will help prevent any malicious activity.

   Automation Functionality: Develop a programming function that can automate responses to security events.

2. Non-Core Functionality:

   Multi-Cloud Support: Solutions to extend the support of multiple cloud providers beyond AWS.

   Integration with additional Third-Pary tools: Provided I find tools that can be used to ensure security from third-party sources.

   Historical Data Analysis: Possibly implement the ability to analyse historical security data for trends.

   Cost Monitoring and optimization recommendations: Another possible feature to include that monitors and suggests costs optimization related to security.

3. Users:

   DevOps: They will be the primary users of this system as DevOps use cloud container solutions all the time. They will rely on monitoring and security of the containers, protecting the environment.

   Compliance: Compliance is important for security ensuring that any user has privileged access to the cloud container environment. They may use this tool to ensure their containers comply with regulations.

   System Administrators: The administrator user will benefit from automation and the tools involved that will help secure the environment.

Cloud Architects: Architects may be interested in a different multi-cloud architecture for cloud-agnostics solutions.

Cost Management teams: Team users might be interested in cost monitoring and how its optimizations are provided by the system as part of their own.

Considering the possibility of what this project can deliver, making it all work and come together will be challenging. When users have a hands-on experience with technologies that are becoming industry standards. Having hands-on experience is very important, especially for Cybersecurity specialists and employees within any organisation alike. As cloud computing and containerization increase as a standard within businesses, the need to know how to use them and understand them increases as well. Overall, this project has aspects that fit individual criteria but there with the combination of features in a single Customisable, and potentially multi-cloud compatible solution sets it apart.

Who is going to use this?

There are many individuals and organization teams that can and potentially would / will use this including myself to learn, adapt and enhance their own knowledge on Cloud container environments. The more specific types of people or teams would be:

1. DevOps teams: DevOps have considerable use for Cloud Containers as they use them in their everyday environments to test different configurations. Cloud Containers provide a solution that makes it easy to run in any environment.

2. Security teams: Cloud containers will help gain insights into the security posture of the containerised application that is under development which will be checked and respond to any security incidents or alerts. They can add or remove technologies from their environment while constantly monitoring and logging to see the effects taken.

3. Compliance Teams: There is potential for this tool to offer and ensure that the container cluster complies with their security and regulatory requirements. Stakeholders might interfere with the original containerised environment they use and would prefer an alternative.

4. System Administrators: They can benefit from the automation that will help maintain a secure environment.

5. Cloud Architects: Their interests may lead to the multi-cloud aspect of the project which aligns with architectural considerations involving cloud-agnostics solutions.

This provides a perfect example of who in any organisation would be interested in looking at the project and taking interest in the potential. The stakeholders involved in the development may also see the potential of security of containerised applications if a company does not already have them implemented.

Precedent for this application

Although this was not my original Project plan, I saw the list of projects and this one stood out the most for me and the desire to learn more about containers, especially for AWS intrigued me. As most people know Amazon is a massive company and has got buildings in different countries around the world. Their cloud services can be anywhere that has internet and servers that provide stable networks.

The concept of containers and using them to develop applications where it is efficient when different hardware and software's are involved, desired. The need and increased need of cloud containerised architecture environments is growing fast and has been since docker came to be in existence. Amazons Web Services also has proven that cloud infrastructure is rising fast and has doubled in the last seven years. Microsoft azure has even managed to make billions in profits due to Azure cloud infrastructure being desired and used more over the last couple of years.

How does this project differ?

There are security container architectures out there that use many different software's, applications, and technologies. But none I have come across are doing exactly as this project is trying to achieve so therefore it is something of a new idea. The difference between modern containerization and earlier methods lies in the level of encapsulation and portability. Methods require extensive configurations and understanding of what architecture is being developed for the containerised environment which can make it time consuming.

To list some of the key differences I have noticed from different projects using AWS containers.

The Different projects

- Prevent cloud misconfiguration and vulnerability finder during build time.
- Security monitor for Netflix using botocore and python.
- Detect compliance and security violations across infrastructures as code.
- Practical study plan to become a successful cybersecurity engineer based on roles like Pentest, AppSec using azure-security, API, and study-guides.
- Open-source cloud-native security lake platform (SIEM alternative) for threat hunting, detection & response.

There are a lot more that I have found online and in on GitHub but none are using the identical tools, applications, or software's in combination with one another.

There are key differences between modern containerization compared to earlier existing versions where the methods of encapsulation and portability vary. Older

methods required extensive configuration and manual intervention ensuring applications worked across different container environments. One thing that should be kept in mind is that technologies, applications, and tools get continuous development and changes over the years. They also get bought and taken over by other companies and implementations change and redevelop over the years, if not redeveloped than their use gets changed to focus on a specific use.

The Plan

Amazon Web Services Container Security

Amazon Web Services container security will be the combination of tools and technologies designed to protect the integrity, confidentiality, and availability of the containerised applications. This is important considering the massive growing use of cloud infrastructure, products and services involved in both personal use and organisations alike.
Containers are very popular technologies for packaging and deploying applications in an isolated environment that are persistent when it comes to data and resources. Cloud services such as Elastic Kubernetes service (EKS) or even Elastic Container Service (ECS) are very prominent and well recognised services.

The crucial and important part of this project is protecting the containers and ensuring the best security possible. The focus will be on preventing unauthorised access, data breaches and any other potential security risks that must be prevented.

What's involved in the project:

1. Intrusion Detection and Prevention: The container applications will require real-time monitoring to identify any suspicious activity and respond to it immediately. The Detection and Prevention systems will be critical for preventing unauthorised access, data exfiltration and other security incidents.

2. Runtime Security: Security tools and practices that will protect the container/s from any malicious attempts. Container runtime security is a subset of container security and workload protection, dealing with everything from instantiation to termination. This will focus on monitoring the behaviour of applications at runtime which will look for unusual or potentially malicious activity.

3. Log and event Collection: AWS Firelens can be integrated to the container application to collect and manage logs as well as events generated by the container. This will help identify crucial data for identifying security incidents.

4. Network Security: This will be a case of measuring communications between containers and other resources in the AWS environment.

5. Automated Response: Using a computer service that runs code to automate responses towards security events can provide extra security.

6. Monitoring: To keep a container secure, there has to be monitoring provided at all times to ensure everything in the AWS environment is safe and secure. A user will also need to have access to these

monitoring tools which should provide an accurate and easy to read interface.

AWS Cloud Container Requirements

Software's: AWS Containers, Elastic Container Service (ECS) or Elastic Kubernetes Services, CloudWatch, CloudTrail, Firelens, Falco, Calico and Lambda.

Docker is no longer supported by Kubernetes but they still work together, so I added them both because when combined with one another, they are very powerful combinations when it comes to security.

Hardware: The computer resources for ECS and EKS which require CPU, Memory (RAM), storage (HDD's or SSD's), network bandwidth and GPU for GUI interface.

All the hardware will be on the Cloud server side, I will only need access to the AWS website and any others that can help me build this. There is a cost for using the service which isn't a problem as long as I only use it when configuring the Cloud containers environment. The application will be turned off when no longer in use outside of making it work. The last thing I will need which is the very important is internet access, thankfully My home has fiber and the college provides internet access as well.

What each software is and does

AWS Containers: Amazon Web Services (AWS) containers allow the user to package applications and their dependencies in an isolated environment. AWS containers ensure consistency across different systems making it highly desirable. There are also services like Elastic Container Services (ECS) and Elastic Kubernetes Services (EKS) that are for managing containerised applications.

Elastic Container Services (ECS): ECS is a management service used for running and stopping containers. ECS is a highly scalable and secure platform for deploying container applications allowing you to focus on developing and deploying.

Elastic Kubernetes Services (EKS): EKS is AWS's managed Kubernetes service. Kubernetes is a well-known open-source container platform that automates the deployment, scaling, and management of containerised applications. EKS also helps simplify the process of running Kubernetes on AWS making it much easier for users.

CloudWatch: This is a monitoring service that provides real-time monitoring of resources and applications. CloudWatch collects and tracks metrics, monitoring log files and sets alarms. CloudWatch is also used to gain visibility into resources, utilization, application performance, and operational health system wide.

CloudTrail: AWS CloudTrail is a service that provides records of actions taken by not only the users but also the services in your AWS account. This is a service that will capture API calls made on your account while including who made them and what

actions were taken during the process. This service helps security, ensuring compliance and troubleshooting.

FireLens: Amazon FireLens is a logging aggregation feature sending logs to various AWS services such as ECS. FireLens can be used to specify log routing to send logs to different AWS services as well as third-party endpoints helping centralise and analyse container logs.

Falco: This is an open-source cloud-native runtime security project to help with security. Falco provides intrusion detection functionality, detecting abnormal application behaviour and unauthorised activity within the containerised environment.

Calico: Calico is another open-source service that provides networking security for containers with load balancing capabilities in Kubernetes. Calico also helps communications between containers while ensuring network policies are enforced, making it a great service for security.

Lambda: Lambda Is an AWS service that is serverless computing letting you run code without provisioning or managing servers. This makes it ideal for anyone that requires their code to be executed in response to events and Lambda scales to handle this which makes it perfect for serverless architectures.

The Milestones of the Project

1. The project Kick off (week 2 – 3, Semester 1):
    a. Select or choose a project provided by Lecture's.
    b. Research the project provided by Lecture's or write a project description of the project you want to work on.
    c. Meet Mentor every week to discuss the choice made and how to proceed forward with the project.

2. Project selection and Documentation (week 4 – 7, Semester 1):
    a. Work on the project Specification and Plan.
    b. Research the different technologies that could be used.
    c. Research the technologies that are competitive with the ones you are planning on using.
    d. Meet with mentor every week to discuss what is needed on the documents.

3. Research document and Labs (week 8 – 11, Semester 1):
    a. Sign up / make account/s to the different services that own the technologies for the project.
    b. Research every technology, application and Software that will be used to develop the project.
    c. Train myself using AWS Cloud quest and Educate lab rooms to help learn more about cloud technologies.
    d. Write the research document.
    e. Meet with mentor to inform of progress on the document and labs.

4. Continue labs and start project development (Christmas break)

a. Over the break, keep up with the labs, learning Cloud containers and security.
b. Start developing the AWS Container.
c. Ensure application works and can be accessed.

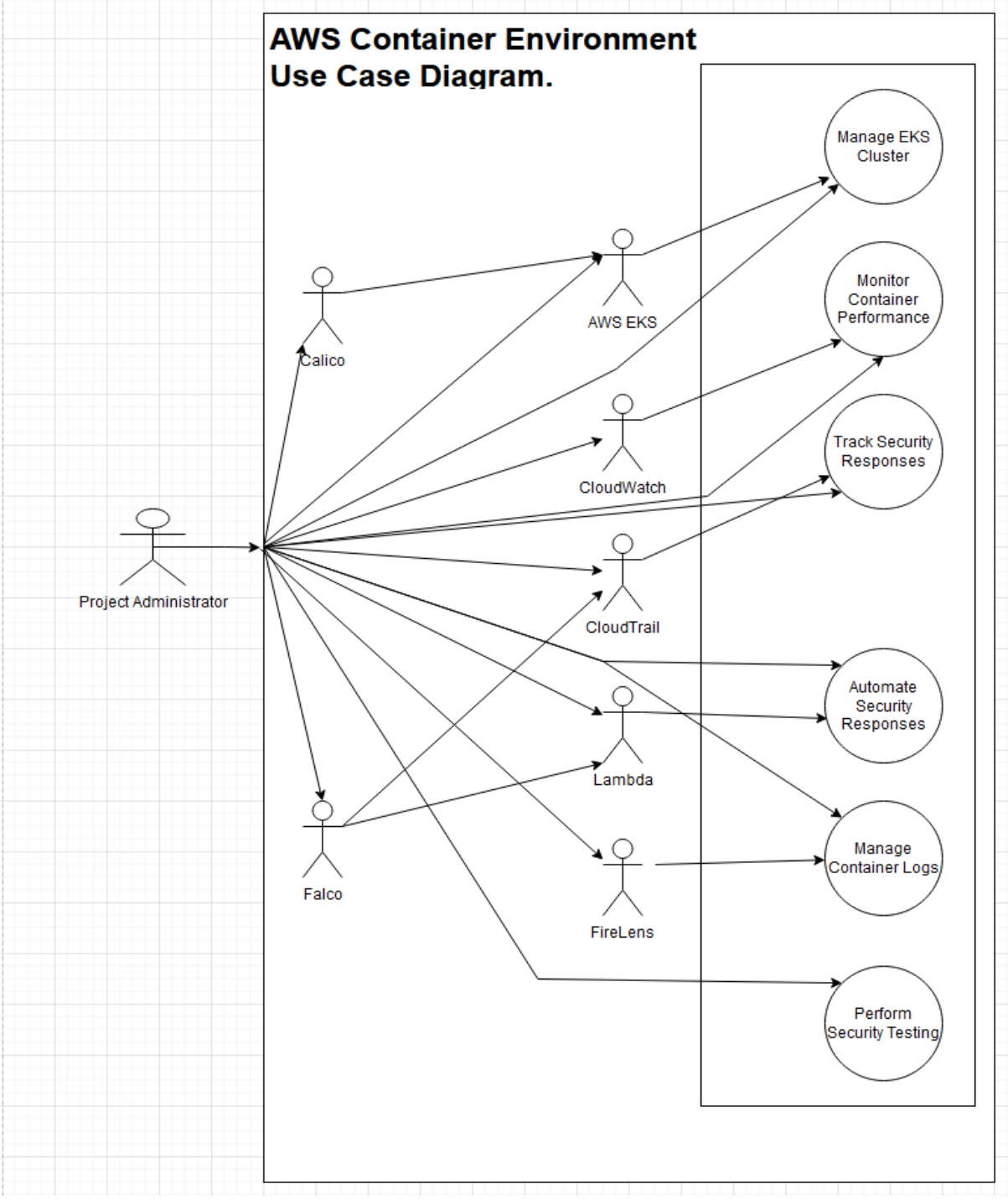
5. Project Presentation & Continue project development (week 1, Semester 2):
    a. Make slides describing different aspects of the project.
    b. Practice what to say during the presentation and time myself.
    c. Check in with mentor before and after the presentation to keep up with the progress.
    d. Continue to configure and setup the AWS container such as create EKS clusters.

6. Project Poster and AWS Container (week 2 – 6, Semester 2):
    a. Create the project poster according to specified description based on the guide.
    b. Setup networking, load balancing and auto-scaling policies.
    c. Configure CloudWatch for real-time monitoring.
    d. Enable CloudTrail for auditing and log analysis.
    e. Create Lambda functions for automated response to security events.
    f. Integrate Lambda functions with CloudWatch and other destinations.
    g. Implement the IDS, IPS, Falco and Calico.
    h. Testing the IDS and IPS to see if they are performing correctly.
    i. Meet with mentor to show how the project is coming along as well as the poster.

7. Final Report, Product and Quality (Week 7 – 11, Semester 2):
    a. Conduct thorough testing of the containerized application and security measures.
    b. Validate the effectiveness of security monitoring and response mechanisms.
    c. Perform constant checkups of the application to ensure working effectively.
    d. Perform Quality Assurance checks to ensure the AWS container environment is working as intended.
    e. Create detailed project documentation including architecture diagrams, configurations, and security policies.
    f. Prepare final report for the final date ensuring the quality is perfect.
    g. Check in with mentor to assure everything is going smoothly and make sure that the documentation is up to quality as well as the project.

Conducting regular check-ups with my mentor to ensure that everything is going smoothly throughout the project's development process. Developing the AWS container and the security along side the other aspects of the project such as the poster will help ensure its progress is constantly moving forward.

Use Case Diagram



Manage EKS Cluster: The use case diagram shows that the project Administrator is interacting with the EKS cluster, but also the AWS EKS which also has a use with the EKS cluster.

Monitor Container Performance: The use case involves interaction between the project administrator and CloudWatch.

Track Security Events: The use case Interactions between the project administrator and CloudWatch.

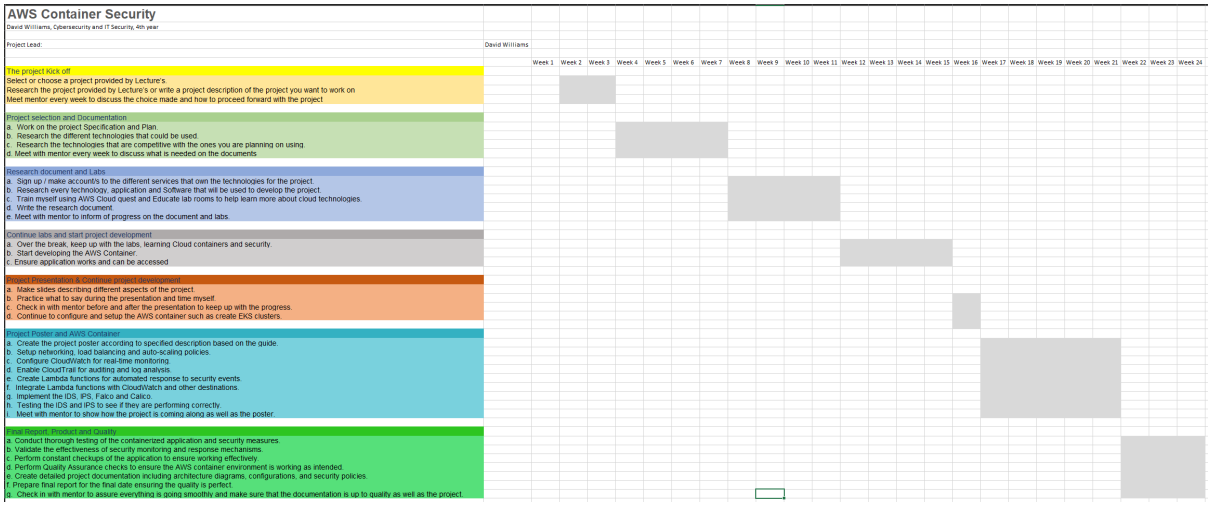
Automation Security Responses: Both Lambda and the project administrators use case points to the automation security response.

Manage Container Logs: FireLens being a log application would be the use case for this while the Project Administrator would use case both.

Intrusion Detection with Falco: Falco as an intrusion detection system would interact with CloudTrail and Lambda for tracking and responding to events. The project administrator would be actively using this to check if working effectively.

Network Security with Calico: Interactions between Calico and AWS EKS for ensuring network security within the container system while the project administrator checks and uses this to ensure security.

Gantt Chart Diagram



A Gantt chart diagram to effectively show how the progress should be during the college term over the total weeks that there are. I provided a larger scaled version of this chart below to make it easier to view but unfortunately it is on a side angle.

What the Gantt chart provides:

Visual representation: The chart provides a visual representation of the project from the start to finish date and how each aspect of the project comes in together.

Task dependencies: Tasks that require or need to be completed before others are represented in the Gantt chart which helps the visualization of the project as well.

Resource Allocation: Even though this is a solo project with mentorship helping the process. It helps provide a concise visual of how and where resources can be effectively used and where to put those resources if overlapping happens.

Adjusting Timelines: Unforeseen things happen in everyone's lives making the Gantt chart perfect for allowing adjustments of timelines if and when there are changes in the project scope.

Milestones:

Key Events or Achieve: The milestone that is represented in the Gantt char signifies the achievement of the project.

Decision Points: Decisions are important and implemented to show that there is thought and logical reasoning put in place for those to help proceed through the project and each phase.

Visibility of Progress: While Milestones provide high-level visibility into the projects course of time. They help track progress and points of the project that require focus and when.

Communication Tool: From a business perspective, Gantt charts are essential to help show stakeholders where the business/organisation is in their progress of the project stage and where they should be. Not all projects go to plan but they help give a broad idea of everything.

Setting Objectives: Objectives are important when it comes to goals for any project and that is something that is much needed visually but also when it comes to a sense of accomplishment for the development team.

With all those points expressed, the Gantt chart's visualization should help any involved in the project's development. The comprehensive view of the entire project schedule, milestones, highlighted key points make it so that the individual (myself) of this project take serious consideration on the time frame involved as well as what is involved in the development of the project.

WS Container Security

Williams, Cybersecurity and IT Security, 4th year

t Lead: David Williams

	Week 1	Week 2	Week 3	Week 4	Week 5	Week 6	Week 7	Week 8	Week 9	Week 10	Week 11	Week 12	Week 13	Week 14	Week 15	Week 16	Week 17	Week 18	Week 19	Week 20	Week 21	Week 22	Week 23	Week 24
roject Kick off																								
ct or choose a project provided by Lecture's.																								
arch the project provided by Lecture's or write a project description of the project you want to work on																								
mentor every week to discuss the choice made and how to proceed forward with the project																								
ct selection and Documentation																								
ork on the project Specification and Plan.																								
esearch the different technologies that could be used.																								
esearch the technologies that are competitive with the ones you are planning on using.																								
eet with mentor every week to discuss what is needed on the documents																								
arch document and Labs																								
gn up / make account/s to the different services that own the technologies for the project.																								
esearch every technology, application and Software that will be used to develop the project.																								
ain myself using AWS Cloud quest and Educate lab rooms to help learn more about cloud technologies.																								
rite the research document.																								
eet with mentor to inform of progress on the document and labs.																								
nue labs and start project development																								
ver the break, keep up with the labs, learning Cloud containers and security.																								
art developing the AWS Container.																								
sure application works and can be accessed																								
ct Presentation & Continue project development																								
ake slides describing different aspects of the project.																								
ractice what to say during the presentation and time myself.																								
heck in with mentor before and after the presentation to keep up with the progress.																								
ontinue to configure and setup the AWS container such as create EKS clusters.																								
ct Poster and AWS Container																								
reate the project poster according to specified description based on the guide.																								
etup networking, load balancing and auto-scaling policies.																								
onfigure CloudWatch for real-time monitoring.																								
nable CloudTrail for auditing and log analysis.																								
reate Lambda functions for automated response to security events.																								
tegrate Lambda functions with CloudWatch and other destinations.																								
plement the IDS, IPS, Falco and Calico.																								
esting the IDS and IPS to see if they are performing correctly.																								
eet with mentor to show how the project is coming along as well as the poster.																								
Report, Product and Quality																								
nduct thorough testing of the containerized application and security measures.																								
lidate the effectiveness of security monitoring and response mechanisms.																								
rform constant checkups of the application to ensure working effectively.																								
rform Quality Assurance checks to ensure the AWS container environment is working as intended.																								
eate detailed project documentation including architecture diagrams, configurations, and security policies.																								
epare final report for the final date ensuring the quality is perfect.																								
heck in with mentor to assure everything is going smoothly and make sure that the documentation is up to quality as well as the project.																								

Any Exceptional requirements?

The exceptional requirements for this project are really down to my ability to:

1. Do the research.
2. Implement what I learn from the research to the AWS container security environment.
3. Keep working on the project as much as possible whenever possible.
4. Keep in contact with my mentor, making sure I meet them every week and listen to their feedback. Listen to their advice and apply it to the project because their wisdom is crucial to the success of this project as well.
5. Consistently check that the project is working as intended.
6. To get as far as possible in the product, meaning make all the software's, tools, technologies, applications work for the project and are used.
7. Finish on time on the right dates and never be behind.

The research is going to be another crucial aspect of this project because there is a lot of information out there on the internet. I want to make sure I follow the correct documentation and updated guides to get through any and all aspects of this project to ensure that I deliver a satisfactory project not only for myself but to anyone who is interested in what this project can be.

The Unique requirements: This project is unique as it requires a focus on Security on containerised applications on a high level utilizing many different tools, software's, and applications to ensure it works. Many of these require dedicated learning and time to implement and whilst there are projects out there that are about container security. They do not use or do what this project is about, which makes this project unique and incredibly interesting, and also important.

Why is this project important: Cybersecurity is important no matter what because cyber threats are increasing and have risen massively since covid happened. There are statistics and measures that have been made to show that cybercrime has cost the global working industry billions. Cloud containerization has become a giant in the past 10 years making it a perfect candidate to help secure as a Cybersecurity student but also to help provide a solution to security.

The resources of this project: Cybersecurity needs and requires as many tools as possible to enhance security and the tools in this project are perfect for just that. They do what is required of the project specification. Intrusion detection and prevention systems are an incredibly important aspect of security preventing some if not most of the worst security threats possible.

Glossary

ECS – Elastic Container Services

EKS – Elastic Kubernetes Services

AWS – Amazon Web Services

IDS – Intrusion Detection System

IPS - Intrusion Prevention System

CPU – Central Processing Unit

GPU – Graphics Processing Unit

RAM – Random Access Memory

OS - operating system

SIEM – Security Information & event management System

VM – Virtual Machine

HDD – Hard disk Drive

SDD – Solid State Drive

GUI – Graphical User Interface

API – Application Programming Interface

References

Falco and alternatives:

1. *List of best Falco alternatives & competitors 2023* (no date) *TrustRadius*. Available at: https://www.trustradius.com/products/falco/competitors (Accessed: 26 October 2023).

2. *Container security tools* (2023) *Tigera*. Available at: https://www.tigera.io/learn/guides/container-security-best-practices/container-security-tools/ (Accessed: 26 October 2023).

Elastic Kubernetes service and alternatives:

3. (No date) *Top 10 Amazon Elastic Kubernetes Service (Amazon Eks) alternatives ... - G2*. Available at: https://www.g2.com/products/amazon-elastic-kubernetes-service-amazon-eks/competitors/alternatives (Accessed: 26 October 2023).

4. *10 Kubernetes alternatives and why you need them* (2023) *Aqua*. Available at: https://www.aquasec.com/cloud-native-academy/kubernetes-101/kubernetes-alternatives/ (Accessed: 26 October 2023).

5. Slingerland, C. (no date) *10 top Kubernetes alternatives (and should you switch?)*, *10 TOP Kubernetes Alternatives (And Should You Switch?)*. Available at: https://www.cloudzero.com/blog/kubernetes-alternatives (Accessed: 26 October 2023).

6. *Kubernetes alternatives 2023: Top 8 container orchestration tools* (2023) *ServerTribe*. Available at: https://www.servertribe.com/kubernetes-alternatives/ (Accessed: 26 October 2023).

CloudWatch and alternatives:

7. (No date a) *Top 10 amazon CloudWatch Alternatives & competitors | G2*. Available at: https://www.g2.com/products/amazon-cloudwatch/competitors/alternatives (Accessed: 26 October 2023).

8. Hoos, J.T. and A. (2023) *Top 10 CloudWatch alternatives in 2023*, *Better Stack Community*. Available at: https://betterstack.com/community/comparisons/cloudwatch-alternatives/ (Accessed: 26 October 2023).

9. Community, F.D. (2023) *Top 9 CloudWatch alternatives that will make monitoring better: Signoz*, *SigNoz RSS*. Available at: https://signoz.io/blog/cloudwatch-alternatives/ (Accessed: 26 October 2023).

10. MetricFire (no date) *CloudWatch vs. alternatives*, *MetricFire Blog*. Available at: https://www.metricfire.com/blog/cloudwatch-vs-alternatives/ (Accessed: 26 October 2023).

11. Gartner, Inc. (no date) *Top amazon CloudWatch competitors & alternatives 2023: Gartner Peer insights*, *Gartner*. Available at: https://www.gartner.com/reviews/market/application-performance-monitoring-and-observability/vendor/amazon-web-services/product/amazon-cloudwatch/alternatives (Accessed: 26 October 2023).

Lambda alternatives:

12. (No date a) *Top 10 Aws Lambda Alternatives & Competitors* | *G2*. Available at: https://www.g2.com/products/aws-lambda/competitors/alternatives (Accessed: 26 October 2023).

13. Gartner, Inc. (no date b) *Top AWS lambda competitors & alternatives 2023: Gartner Peer insights*, *Gartner*. Available at: https://www.gartner.com/reviews/market/application-platforms-reviews/vendor/amazon-web-services/product/aws-lambda/alternatives (Accessed: 26 October 2023).

14. (No date a) *Aws Lambda Alternatives and similar sites & apps* | *alternative to*. Available at: https://alternativeto.net/software/amazon-web-services-lambda/ (Accessed: 26 October 2023).

15. *Aws Lambda Pricing, alternatives & more 2023* (no date) *Capterra*. Available at: https://www.capterra.com/p/211010/AWS-Lambda/ (Accessed: 26 October 2023).

Calico alternatives:

16. *Top 10 project CALICO alternatives & competitors* | *G2*. Available at: https://www.g2.com/products/project-calico/competitors/alternatives (Accessed: 26 October 2023).

17. *Calico alternatives and reviews (Sep 2023)* (no date) *and Reviews (Sep 2023)*. Available at: https://www.libhunt.com/r/calico (Accessed: 26 October 2023).

18. *Calico alternatives for enterprise businesses in 2023* | *G2*. Available at: https://www.g2.com/products/calico-ai-calico/competitors/alternatives/enterprise (Accessed: 26 October 2023).

Why firelens:

19. Engdahl, S. (2008) *Blogs*, *Amazon*. Available at: https://aws.amazon.com/blogs/containers/under-the-hood-firelens-for-amazon-ecs-tasks/ (Accessed: 26 October 2023).

Guide to docker in AWS:

20. Conn, M. (2021) *Getting started*, *Amazon*. Available at: https://aws.amazon.com/getting-started/hands-on/deploy-docker-containers/ (Accessed: 26 October 2023).

Different container solutions:

21. *Top 10 best container software in 2023* (2023) *Software Testing Help*. Available at: https://www.softwaretestinghelp.com/container-software/ (Accessed: 26 October 2023).

22. (No date a) *Best container software 2023: Compare reviews on 100+* | *G2*. Available at: https://www.g2.com/categories/container-management (Accessed: 26 October 2023).

Container monitoring tools:

23. Rahić A. (2023) *12 Best Docker Container Monitoring Tools [2023 comparison]*, *Sematext*. Available at: https://sematext.com/blog/docker-container-monitoring/ (Accessed: 26 October 2023).

24. Hoos, J.T. and A. (2023) *10 best cloud logging tools in 2023*, *Better Stack Community*. Available at: https://betterstack.com/community/comparisons/cloud-logging-tools/ (Accessed: 26 October 2023).

25. Slingerland, C. (no date) *The 15 best container monitoring tools (reviewed 2023)*, *The 15 Best Container Monitoring Tools (REVIEWED 2023)*. Available at: https://www.cloudzero.com/blog/container-monitoring-tools (Accessed: 26 October 2023).

26. *Configure logging drivers* (2023) *Docker Documentation*. Available at: https://docs.docker.com/config/containers/logging/configure/ (Accessed: 26 October 2023).

What are cloud containers used for:

27. *What are containers? | google cloud* (no date) *Google*. Available at: https://cloud.google.com/learn/what-are-containers (Accessed: 26 October 2023).

28. *What are containers in DevOps? benefits, use cases* (no date) *KnowledgeHut*. Available at: https://www.knowledgehut.com/blog/devops/devops-containers (Accessed: 26 October 2023).

29. *What is microservices architecture? | google cloud* (no date) *Google*. Available at: https://cloud.google.com/learn/what-is-microservices-architecture (Accessed: 26 October 2023).

30. *What is a CI/CD pipeline?* (no date) *Red Hat - We make open-source technologies for the enterprise*. Available at: https://www.redhat.com/en/topics/devops/what-cicd-pipeline (Accessed: 26 October 2023).

31. Tweedie-Yates, S. (2023) *Container isolation: Is a container a security boundary? Cloud native applications security*. Available at: https://blog.aquasec.com/container-isolation (Accessed: 26 October 2023).

32. *Multicloud and Hybrid Cloud Solutions: Microsoft Azure* (no date) *Multicloud and Hybrid Cloud Solutions | Microsoft Azure*. Available at: https://azure.microsoft.com/en-us/solutions/hybrid-cloud-app/ (Accessed: 26 October 2023).

33. Zaman, S. (2023) *Multi-Cloud vs Hybrid Cloud: What's the Difference? Folio3 Cloud Services*. Available at: https://cloud.folio3.com/blog/multi-cloud-vs-hybrid-cloud/ (Accessed: 26 October 2023).

34. *Stateful vs stateless* (no date) *Red Hat - We make open-source technologies for the enterprise*. Available at: https://www.redhat.com/en/topics/cloud-native-apps/stateful-vs-stateless (Accessed: 26 October 2023).

35. Person (2023) *Stateless vs stateful containers: What's the difference and why does it matter? Contino: Global transformation consultancy*, *Contino*. Available at: https://www.contino.io/insights/stateless-vs-stateful-containers-whats-the-difference-and-why-does-it-matter (Accessed: 26 October 2023).

36. *What is container orchestration | google cloud* (no date) *Google*. Available at: https://cloud.google.com/discover/what-is-container-orchestration (Accessed: 26 October 2023).

37. *What is container orchestration?* (no date) *IBM*. Available at: https://www.ibm.com/topics/container-orchestration (Accessed: 26 October 2023).

38. *What is container orchestration?* (2023) *Cisco*. Available at: https://www.cisco.com/c/en/us/solutions/cloud/what-is-container-orchestration.html (Accessed: 26 October 2023).

39. Rosen, C. (2019) *Containers are efficient-now, make them even better*, *IBM Blog*. Available at: https://www.ibm.com/blog/containers-are-efficient-now-make-them-even-better/ (Accessed: 26 October 2023).

40. Zimmergren, T. (2023) *Boost efficiency and cost savings with new sustainability guidance in the Cloud Adoption Framework: Azure blog:*

Microsoft Azure, *Azure Blog*. Available at: https://azure.microsoft.com/en-us/blog/boost-efficiency-and-cost-savings-with-new-sustainability-guidance-in-the-cloud-adoption-framework/ (Accessed: 26 October 2023).

41. *Development and testing on Azure: Microsoft Azure* (no date) *Development and Testing on Azure | Microsoft Azure*. Available at: https://azure.microsoft.com/en-us/solutions/dev-test/ (Accessed: 26 October 2023).

42. (No date) *The growing trend: Cloud development environments emerging as ...* Available at: https://venturebeat.com/business/the-growing-trend-cloud-development-environments-emerging-as-enterprise-essentials/ (Accessed: 26 October 2023).

43. *How to move legacy applications to containers and Kubernetes* (no date) *Red Hat - We make open-source technologies for the enterprise*. Available at: https://www.redhat.com/en/resources/moving-legacy-applications-to-containers-overview (Accessed: 26 October 2023).

44. (No date a) *IBM developer*. Available at: https://developer.ibm.com/articles/containerization-of-legacy-applications/ (Accessed: 26 October 2023).

45. DevOps (2023) *What are the key challenges and best practices for scaling containers in the cloud? How to Scale Containers in the Cloud: Challenges and Best Practices*. Available at: https://www.linkedin.com/advice/3/what-key-challenges-best-practices-scaling-containers-cloud (Accessed: 26 October 2023).

46. *Scaling an application | Google Kubernetes Engine (GKE) | google cloud* (no date) *Google*. Available at: https://cloud.google.com/kubernetes-engine/docs/how-to/scaling-apps (Accessed: 26 October 2023).

Cloud leader:

47. Law, M. (2023) *Top 10 biggest cloud providers in the world in 2023*, *Technology Magazine*. Available at: https://technologymagazine.com/top10/top-10-biggest-cloud-providers-in-the-world-in-2023 (Accessed: 26 October 2023).

Cost Management Teams:

48. Kate Eby | April 25 (no date) *The Ultimate Guide to Cost Management*, *Smartsheet*. Available at: https://www.smartsheet.com/ultimate-guide-to-cost-management-and-templates (Accessed: 26 October 2023).

Solutions and challenges:

49. Gandhi, R. (2019) *The benefits of containerization and what it means for you*, *IBM Blog*. Available at: https://www.ibm.com/blog/the-benefits-of-containerization-and-what-it-means-for-you/ (Accessed: 26 October 2023).

GitHub sources:

50. *Build software better, together* (no date) *GitHub*. Available at: https://github.com/topics/aws-security (Accessed: 27 October 2023).