Constructing a Zero-Trust Kubernetes Cluster

Bachelor Thesis

Submitted in partial fulfillment of the requirements for the degree of

Bachelor of Science in Engineering

to the University of Applied Sciences FH Campus Wien
Bachelor Degree Program: Computer Science and Digital Communications

Author:
Guntram Björn Klaus

Student identification number:
c2110475170

Supervisor:
BSc. MSc. Bernhard Taufner

Date:
dd.mm.yyyy

Declaration of authorship:

I declare that this thesis is my own work and that I did not use any aids other than those indicated or any other unauthorized help (e.g., ChatGPT or similar artificial intelligence-based programs). I certify that this work does not contain any personal data, and that I have clarified any copyright, license or image-law issues pertaining to the electronic publication of this thesis. Otherwise, I will indemnify and hold harmless the FH Campus Wien from any claims for compensation by third parties. I certify that I have not submitted this thesis (to an assessor for review) in Austria or abroad in any form as an examination paper. I further certify that the (printed and electronic) copies I have submitted are identical.

Date: Signature:

Abstract

(E.g. “This thesis investigates...”)

Kurzfassung

(Z.B. “Diese Arbeit untersucht...”)

List of Abbreviations

AKS	Azure Kubernetes Service
EKS	Elastic Kubernetes Service
GKE	Google Kubernetes Engine
IT	Information Technology
IoT	Internet of Things
MFA	Multifactor Authentication
NIST	National Institute of Standards and Technology
ZT	Zero Trust

Key Terms

Kubernetes

Cloud

Zero Trust

Least Privilege

Access Control

Contents

1 Introduction

1.1 Kubernetes and Zero Trust

Over the last years, there have been two significant shifts in enterprise IT systems.

One of these shifts addresses the way companies deploy, scale and maintain the lifecycle of their software services. Applications used to be primarily constructed as one large software unit that bundled all features, business logic, user interfaces and data access components. Increasing necessity for scalability, flexibility and maintainability made organizations transition from such monolithic architectures to microservices, which are smaller, separated but loosely coupled software units implementing one component of the larger system at hand. The release of the Docker container platform in 2013 had a significant impact on making transitions from monoliths to microservices feasible. Kubernetes has since emerged as the go-to choice for managing containerized applications at scale, particularly in cloud-native environments. Its ability to automate tasks, scale applications, and support a wide range of use cases makes it a powerful tool for both large enterprises and small teams.

The second shift pertains to how companies secure their computing infrastructure and the resource hosted on it. While there used to be single, easily identifiable network perimeters in the past, for example a single local area network at a company site, modern infrastructures may consist of multiple internal networks, remote offices, mobile workers, different types of virtualization and cloud services. This circumstance has rendered traditional, static, perimeter-based security no longer appropriate, because transgressing this perimeter once means further, unhindered access into a given system. Under a newer security model labeled "Zero Trust", the aim is to restrict such unhindered movement as much as possible by adhering to certain principles and guidelines, the central one being to never grant implicit trust to any actor on a network - hence the term "Zero" Trust. The principles of ZT have existed way before the term "Zero Trust" was coined. A fundamental piece of liteature, which tries to capture what "Zero Trust" concreteley means, is the NIST Special Publication 800-207, titled "Zero Trust Architecture". It is the point of reference for understanding Zero Trust.

As both topics of container orchestration with Kubernetes and improved security paradigms are evermore gaining traction in enterprise systems, it is of paramount importance to explore how exactly this current new paradigm of Zero Trust can be applied to Kubernetes.

1.2 Related Work

Since the emergence and popularization of the concept of "Zero-Trust", a lot of work has been done on the topic, tying it into various domains of IT: Cloud, on-premise infrastructure, IoT, Hardware, Blockchain and much more.

Andrea Manzato, at the University of Padua, implements the Zero Trust model in an enterprise environement using solutions provided by Microsoft Azure. It is investigated how the capabilities and configuration options of Microsoft Defender and Active Directory can be leveraged to protect enterprise resources. Attack scenarios on these technologies are simulated and automated remediation actions are presented.
Dr. Wesam Almobaideen's master thesis, at Rochester Institute of Technology Dubai Campus, explores the topic of Zero-Trust specifically in the context of MFA. A framework combining principles of ZT and MFA is designed and evaluated in terms of performance, security and user satisfaction.
In the context of IoT, Cem Bicer at the technical university of Vienna, explores and evaluates ZT for edge networks. The implementation of the thesis follows ZT guidelines as proposed by the National Institute of Standards and Technology (NIST) and additionally places a blockchain network on top of the ZT architecture.
Furthermore, zero trust in and of itself has been put under scrutiny. In "Theory and Application of Zero Trust Security: A Brief Survey", Kang et. al investigates the current challenges faced when making use of Zero Trust, as well as progress that has been achieved so far when doing so. Here, it is noted that research and knowledge on the theory and application of Zero Trust has not yet matured, and more extensive work is still required to obtain a deeper understanding and more accurate implementation of the paradigm in academia and industry.
In his master's thesis at Utrecht university, Michel Modderkolk proposes a more mature model of Zero Trust, branded "Zero Trust Maturity Model (ZeTuMM)". The work makes use of two scietific methods, "comparison analysis" and "focus area maturity modeling", in order to outline a more fully fledged ZT model.

Independent of ZT, Kubernetes security has been explored in the following works. The paper titled "Designing an intrusion detection system for a Kubernetes cluster" by Hristov, P. (2022) focuses on addressing the security challenges associated with cloud-based architectures, particularly those utilizing Kubernetes. The essay highlights the growing reliance on digital platforms, which has led to an increased demand for automation and high-reliability systems. This reliance has, in turn, increased the use of tools and consequently, the number of security concerns.
"A Systematic evaluation of CVEs and mitigation strategies for a Kubernetes stack" by Fred Nordell (2022) offers a detailed examination of CVE's and possible mitigation strategies
The thesis "Kubernetes Near Real-Time Monitoring and Secure Network Architectures" explores the security challenges and mitigation strategies in Kubernetes. It emphasizes the

importance of securing the Kubernetes control plane, implementing secure configurations, and protecting critical components like the etcd data store. Additionally, it highlights the integration of security practices throughout the development, deployment, and operations lifecycle, advocating for a DevSecOps approach that includes security automation, continuous security testing, and security monitoring.
"Testing the Security of a Kubernetes Cluster in a Production Environment" by Giangiulio and Malmberg at KTH Stockholm emphasizes the critical importance of comprehensive security measures for Kubernetes clusters operating in production environments. It highlights the need for robust security practices, including data encryption, proper user and permissions management, and the use of Role-Based Access Control (RBAC) to manage access to the cluster. Additionally, it underscores the significance of maintaining up-to-date Docker images, using minimal Docker images to reduce potential vulnerabilities, and implementing network policies and secrets management to enhance security.
The thesis "A Security Framework for Multi-Cluster Kubernetes Architectures" also aims to address the security challenges in Kubernetes environments, particularly focusing on multi-cluster setups. The study emphasizes the importance of continuous monitoring and management to mitigate security risks, highlighting the need for a robust security framework that balances security enhancements with minimal performance impact.

According to my knowledge as of April 2024, no major work has been done at the intersection of Zero Trust and Kubernetes.

1.3 Objectives and Methodology

As there is no research done on ZT specifically in the context of Kubernetes, the objective of this thesis is to do exactly that. How are ZT principles applied in Kubernetes? How can ZT be achieved in a Kubernetes setup? The aim is to construct a Kubernetes environment according to ZT principles as outlined in NIST Special Publication 800-207. To narrow down the scope of the research, the following limitations are placed.

1) A self-managed cluster (a self-managed control plane) will be instantiated using kubeadm. It will not be a managed cluster like EKS, AKS, or GKE. The cluster will have one master node and two worker nodes.
2) In order to implement ZT guidelines, open source, free, or free versions of otherwise paid-services are used. Least possible vendor lock-in is preferred.

Formulating above mentioned goals in a single, coherent sentence results in the following research question: How can free software projects and native Kubernetes solutions be leveraged to achieve a cloud-agnostic and zero-trust architecture in Kubernetes?

To do so, NIST Special Publication 800-207 is used as a primary point of reference. The paper will be thoroughly read and summarized in chapter two. Here, an in depth explanation for the importance of ZT shall be given. Core Kubernetes concepts which help understand the subsequent practical setup are also explained in chapter 2. After explaining the pillars of ZT and Kubernetes, technologies and solutions are researched and picked, which will then be practically implemented to achieve the corresponding ZT principle inside the Kubernetes environment. Code snippets shall be presented. Throughout the practical setup, cross references to the NIST publication shall be made. To evaluate the setup, test namespaces will be created and simple test applications will be deployed inside them. User accounts will be created. At the end of the paper we shall critique the environment and answer the question if a setup, which does a 100% justice to ZT, is even possible.

2 Concepts

2.1 Kubernetes

Kubernetes Architecture (Components, Nodes, Pods,), Important for this thesis: CNI, Namespaces

Any enterprise environment can be designed with zero trust tenets in mind. Most organizations already have some elements of zero trust in their enterprise infrastructure or are on their way through implementation of information security and resiliency policies and best practices. Several deployment scenarios and use cases lend themselves readily to a zero trust architecture. For instance, ZTA has its roots in organizations that are geographically distributed and/or have a highly mobile workforce. That said, any organization can benefit from a zero trust archit

2.2 Zero Trust

Summary of NIST 800-207 publication, important concepts

Definitions:

View of the network:

In brief, a Zero Trust Architecture (ZTA) approach to network planning and deployment operates on these core assumptions:

No implicit trust: Treat the entire enterprise network as untrusted. All assets should behave as though attackers are present, necessitating secure communication through authentication and encryption.

External Device Consideration: Acknowledge that networked devices may not be owned or controlled by the enterprise, including those used by visitors or under BYOD policies. Authentication and authorization are required for these devices to access enterprise resources.

Trust Evaluation: Reject the notion of inherent trust for any resource. Continuously assess each asset's security posture before granting access to enterprise resources, emphasizing the need for device authentication alongside subject credentials.

Infrastructure Diversity: Recognize that not all enterprise resources reside on enterprise-owned infrastructure. This encompasses cloud services and remote enterprise subjects, which may rely on external networks for connectivity.

Local Network Skepticism: Assume a lack of trust in local network connections for remote subjects and assets. All connection requests should be treated with suspicion, requiring robust authentication and authorization measures for secure communication.

Consistent Security Policy: Enforce consistent security policies and postures for assets and workflows transitioning between enterprise and external infrastructure. This applies to devices moving across networks and workloads migrating between on-premises and cloud environments.

Policy Engine: The PE is in charge of deciding whether or not to allow access to a resource for a particular resource. To give, deny, or revoke access to a resource, the PE employs enterprise policy together with input from external sources (such as threat intelligence services and CDM systems, which are detailed below) as input to a trust algorithm (see to Section 3.3 for further information). The component of the policy administrator is matched with the PE. The decision is made by the policy engine, which also documents its approval or rejection. The policy administrator then puts the decision into action. In the context of Kubernetes: Trivy for example

Policy Administrator (PA) is the component handling initiation and/or termination of the communication line between a resource and a subject. Any session-specific authentication, credential, or token that a client uses to get access to an enterprise resource would be generated by it. It is directly related to the PE and depends on its final determination of whether to approve or reject a session. The PA sets up the PEP to permit the session to begin if the request has been authenticated and the session is allowed. The PA notifies the PEP to terminate the connection in the event that the session is rejected or an earlier approval is overturned.

The policy enforcement point is in charge of permitting and overseeing connections between a subject and an enterprise resource is known as a policy enforcement point (PEP). The PEP interacts with the PA in order to transmit requests and/or obtain updates on PA policies. Although there is just one logical component in ZTA, it can be divided into two parts: the resource side (such as the gateway component in front of the resource that regulates access) and the client (such as an agent on a laptop) or a single portal component that serves as a gatekeeper for communication channels. The trust zone containing the enterprise resource is located beyond the PEP.

3 Creating the cluster

Build the cluster according to 800-207, describe configuration and code snippets, reference the ZT principle, referene the 800-207 section or section from other paper

4 Discussion and Future Work

5 Conclusion

Bibliography

List of Figures

List of Tables

Appendix

(Hier können Schaltpläne, Programme usw. eingefügt werden.)