

# Security implementation document

Group C

Distributed computing

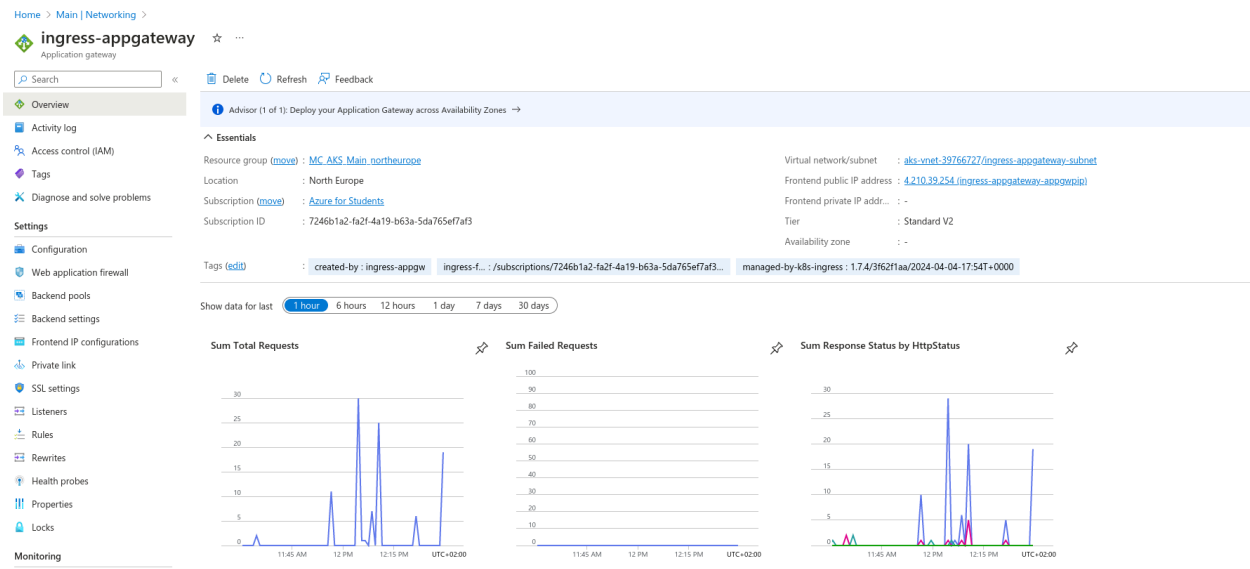
Evgeni, Stefan, Nikolay, Vlad, Naadir, Thomas, Renis

## Introduction

In this document we are going to explain how we managed to implement the researched security methods in our project and how our security works overall. In the security report we mentioned that we are going to implement different levels of security. Azure security, K8s security and Network Security.

## Azure Security & monitoring

In terms of Azure security we implemented a couple of things. One of them is creating the application gateway and configuring it in a way that it prevents malicious traffic as well as DOS attacks.



In here we have also enabled some advanced monitoring options so that we have a bigger overview of the requests and the response time.

The Application gateway also provides us with a WAF and also gives us constant health checks.

[Home](#) > [ingress-appgateway](#) | [Backend settings](#) >

## Backend health

[Refresh](#) [Feedback](#)

### Backend health

By default, Azure Application Gateway probes backend servers to check their health and whether they're ready to serve requests. You can also create custom [Health Probes](#) to mention a specific hostname and path to be probed or a response code to be accepted as Healthy.

The Backend health report is updated based on the respective probe's refresh interval and doesn't depend on the page refresh.

All	Healthy
1 out of 1	1 out of 1

Server (backend pool)	Status	Port (Backend setting)	Protocol	Details	Action
10.244.1.6 (pool-default-backend-service-3000-bp-3000)	Healthy	3000 (bp-default-backend-service-3000-3000)	Http	Success. Received 200 status code	

Also we enabled and followed the AKS recommendations so that we got 100% security checks in Azure.

[Home](#) > [Main](#)

## Main | Microsoft Defender for Cloud

Kubernetes service

Search

- Overview
- Activity log
- Access control (IAM)
- Tags
- Diagnose and solve problems
- Microsoft Defender for Cloud
- Kubernetes resources
  - Namespaces
  - Workloads
  - Services and ingresses
  - Storage
  - Configuration
  - Custom resources
  - Events
  - Run command
- Settings
  - Node pools
  - Cluster configuration
  - Application scaling
  - Networking

For enhanced security capabilities, upgrade your subscription's Microsoft Defender for Cloud 'Containers, Cloud Posture' plans →

Visit Microsoft Defender for Cloud to manage security across your virtual networks, data, apps, and more

Recommendations Security alerts Microsoft Defender for Containers **Off**

0 0

Learn more

[About Microsoft Defender for](#)  
[Explore Container Security cap](#)

### Recommendations

Defender for Cloud continuously monitors the configuration of your Kubernetes services to identify potential security vulnerabilities and recommends actions to mitigate them.



No recommendations to display

There are no security recommendations for this resource

[View all recommendations in Defender for Cloud](#)

### Security alerts

Discover threats at an early stage to quickly respond and prevent future attacks

## Network security

We have enforced strong security rules and policies for access. The K8s API is basically unreachable since we have made it accessible only for specific IPs by setting **Authorized IP ranges**.

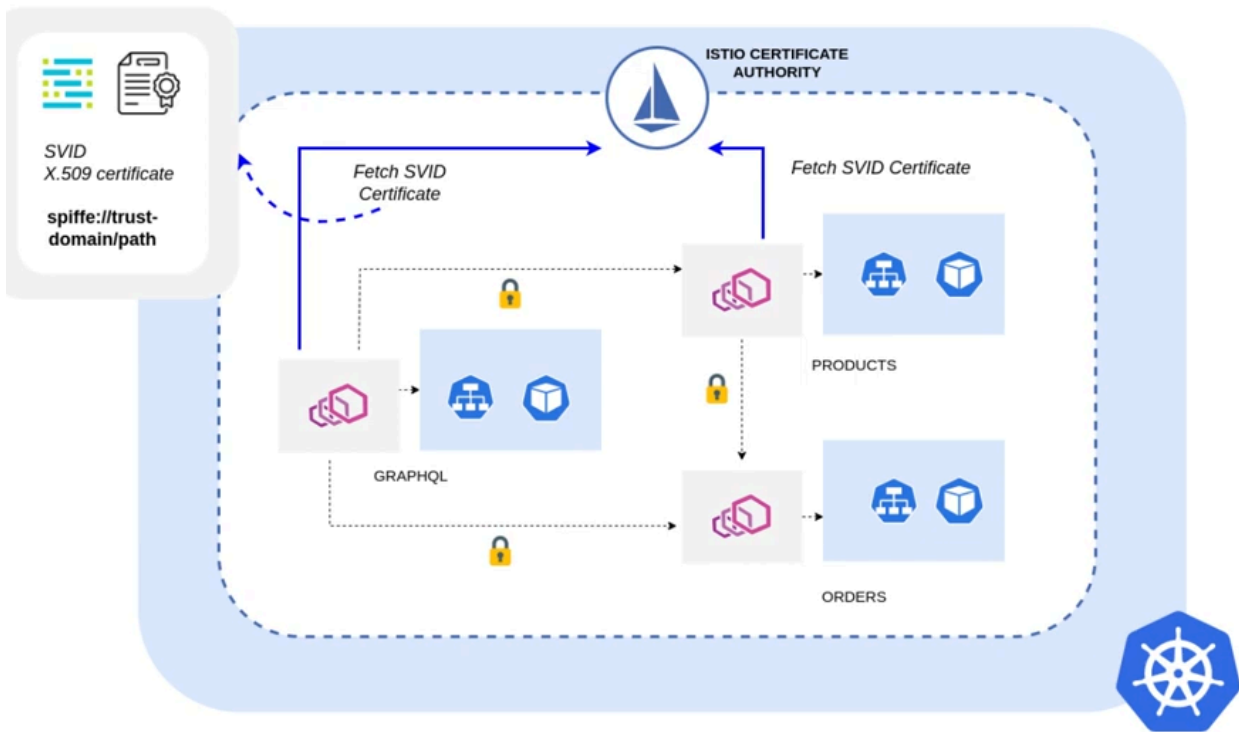
```
zsh zsh
> kubectl get pods
Unable to connect to the server: dial tcp 4.209.61.227:443: i/o timeout
```

This is what unauthorized IP gets when tries to access the AKS API.

## K8s security/pod security

For K8s security we have enabled RBAC authentication for the cluster. That way processes are authenticated by Azure users that have the needed privileges. Since K8s takes care about fallen pods we don't have to worry about backups and ect. The nodes are placed in different V-nets for preventing chain attacks if one of the pods gets compromised.

In the research documents we have talked a lot about encrypted communication pods. For this we have used this K8s package called **Istio**.



This is a brief overview of how Istio mTLS security works. Istio is used as a proxy to check for authentication and as certificate authority , using Spiffe it manages to issue certificates and uses proxy containers to check for authenticated access.

---

Name	State	Reason
<a href="#">backend</a>	✓ Running	
<a href="#">istio-proxy</a>	✓ Running	

Other things that we included in the security part are secrets. We store information like Redis passwords, ports, certificates etc. as K8s secret objects. We extract them from there as values and use them in configuration files.

