CYSEC

Solution Architecture - VPN hosted in Kubernetes cluster

ARCA Trusted OS

CYSEC

1. Introduction

Connecting devices across the globe via the internet has always been a challenge. Networking methods such as firewall and NAT make it difficult to get all of the devices reachable. Additionally, exposing services to the internet such as a HTTP API without any security can lead to malicious usage. As a solution, new technologies called Virtual Private Network (VPN) have been designed. A VPN offers a way to easily connect multiple nodes over the internet and bypasses the network challenges detailed above. Moreover, the communication can optionally be authenticated and encrypted.

To work, a VPN needs a central point that is reachable for any other nodes and this central point is what we called the “VPN server”. Traditionally, the VPN server is hosted somewhere on the internet in a virtual machine and exposes a public IP. To go further, we could decide to host this VPN server inside a Kubernetes cluster. Kubernetes brings advantages such as:

- Scalability: Resources allocated to the VPN server can be easily reconfigured
- Portability: Moving the workload from a point to another
- High Availability: Automatic fallback mechanism when a failure occurs
- Persistence: Configuration persisted along multiple nodes

Moreover, it allows cluster workload to reach devices that are located outside of the cluster and vice versa.

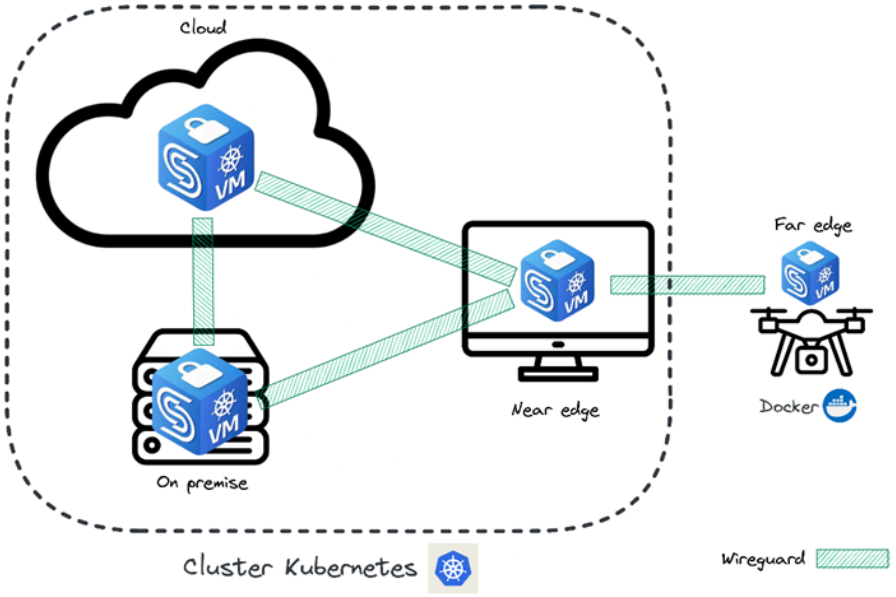

Figure 1. Typical architecture combining Kubernetes cluster and isolated nodes

2. Solution

At CYSEC, we propose a solution architecture that is ready to be deployed on a Kubernetes cluster. Below is the detailed view of the implementation, we're going to cover each part and detail the reasons behind each technical choice.

We use Longhorn as a storage class, but you could use any other alternatives. We simply use Longhorn because it is the default solution deployed as part of our testing components tests suite.

As a VPN technology, we propose to use Wireguard and its container implementation from linuxserver.io. At CYSEC, we believe that Wireguard is currently the best open-source VPN solution for Linux systems. It is integrated directly inside the Linux kernel and requires only a few commands to work. Also, Wireguard uses chacha20-poly1305 as authenticated encryption which offers great performances even without cryptographic hardware acceleration (especially at edge with ARM CPU). Only one instance of this container is required to get a working VPN, that is why we deploy it by using a k8s Deployment with `replicas: 1`.

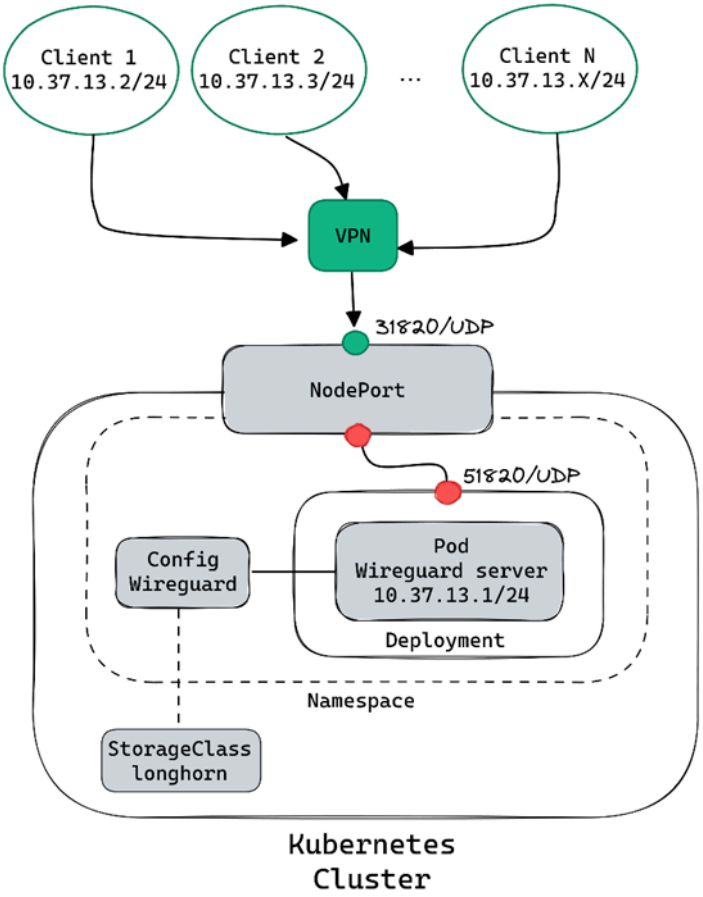

Figure 2. Detailed view of the implementation

One of the main components of a VPN is its configuration (the manifest that declares all the connected devices, their public key and network configurations). It is important to persist it, and this is what we do by persisting the container's `/config` folder inside a storage class volume offered by Longhorn. This is declared directly inside the Deployment file.

The final step is to advertise this pod to the internet so other nodes can connect to it and join the VPN. We achieve this by using k8s Services. There are different types of services and we choose to use NodePort. NodePort exposes a single port to the outside and redirects the traffic toward a workload. This is exactly what we want to achieve because Wireguard needs a single UDP port to work. Other types such as ClusterIP and LoadBalancer could also do the job but in our opinion they generate too much overhead with marginal advantages for this use-case.

If you want to implement this architecture in your own environment, follow our public documentation to reproduce step by step the installation.