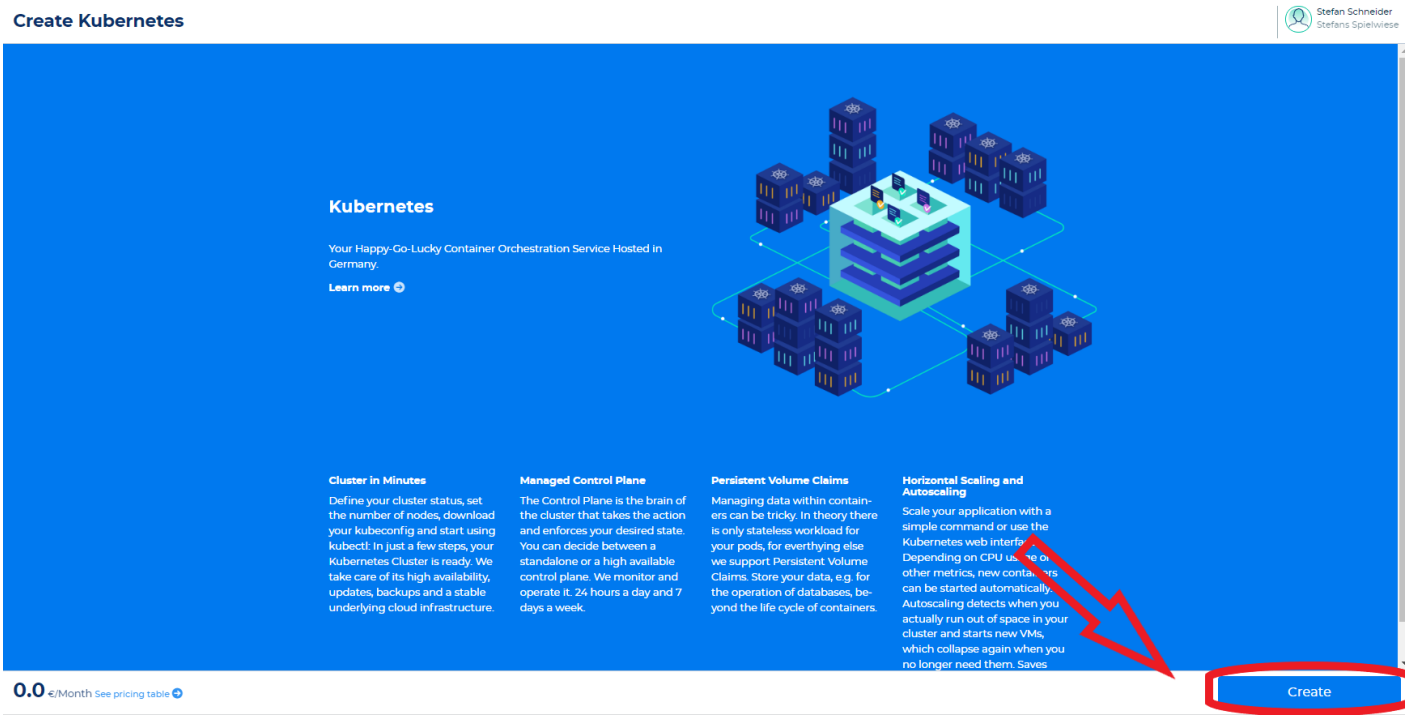
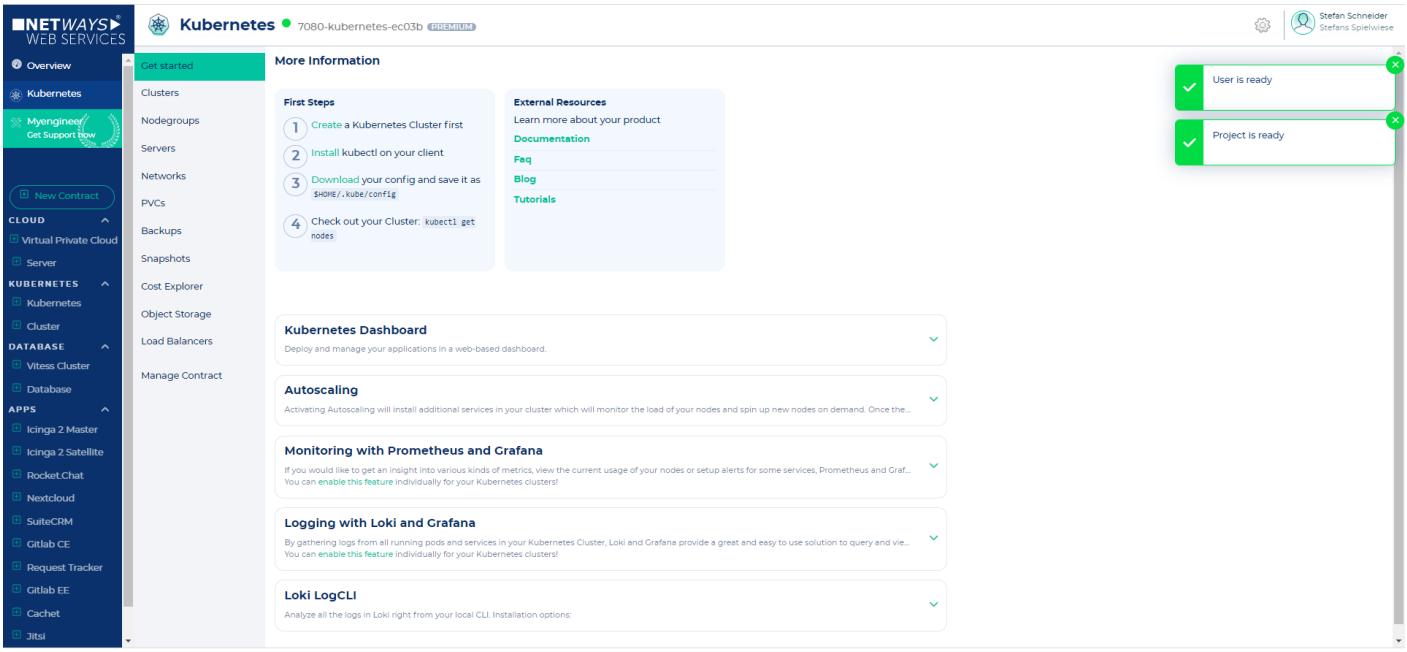
Getting Started

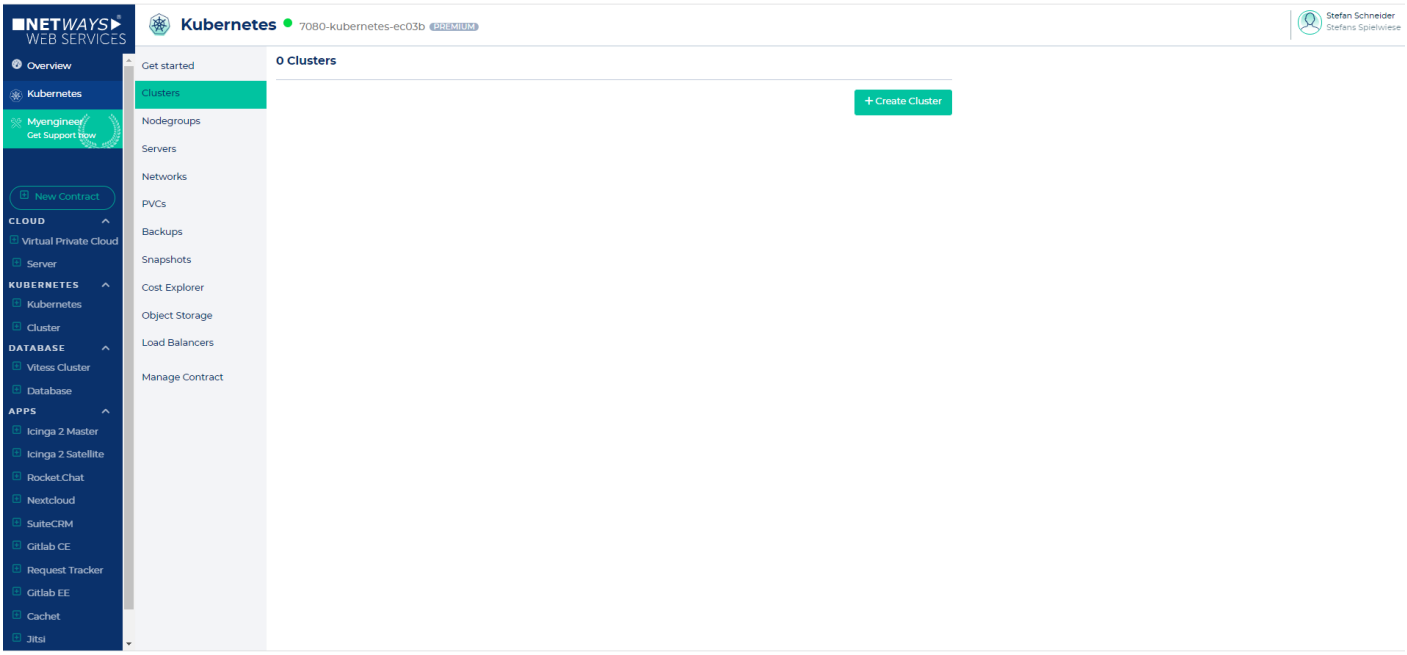
Create NWS Managed Kubernetes

To start our Managed Kubernetes service, you must first create an account on our NWS Customer Interface and provide a valid payment method.





The first step to the cluster would now be to start the first cluster in the Clusters submenu. In the concrete example with the smallest requirements and in version 1.25.2



Create new cluster

Choose a title for your cluster. Leave blank to use the generated name.

Stefans Testcluster

Kubernetes Project

Select one of your existing projects or create a new one.

7080-kubernetes-ec03b | + New Project

Kubernetes Version

Choose your desired Kubernetes version.

1.25.2

Control Plane Nodes

Decide between a setup with a single master node or a high available setup with three master nodes with load balancing.

Single master setup | High availibility setup

Accessibility

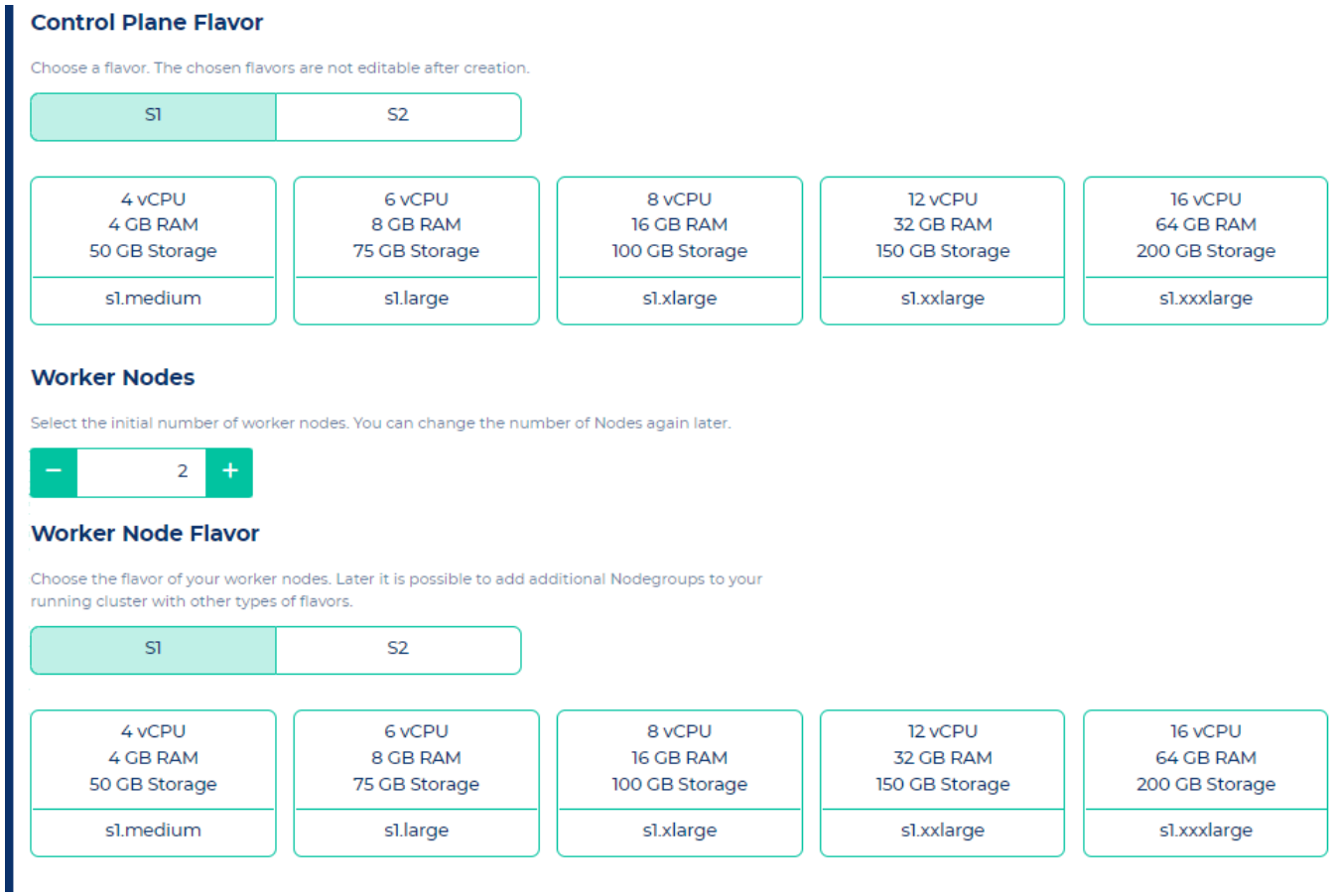
Choose "Private" if you don't want your Kubernetes-API to be accessible over a public floating IP. A private cluster will only be accessible over VPN. To get VPN access, please send your request to nws@netways.de or create a ticket for the MyEngineer.

Public | Private

Custom Subnet

Choose a private subnet the nodes will be launched in. Use CIDR notation. Defaults to 10.0.0.0/24.

10.0.0.0/24



In the background, NWS automations start, create an OpenStack project, create the machines, and configure the cluster with all the necessary components. After 5-10 minutes, the cluster is ready for use.

Install kubectl and kubelogin

kubectl

kubectl is the command-line tool to manage your Kubernetes clusters and is available for Linux, Windows and MacOS. For an easy installation follow the official instructions on kubernetes.io.

kubelogin (kubectl oidc-login)

kubelogin is a plugin that extends kubectl with OpenID Connect. This is mandatory to use NWS-ID with your Kubernetes cluster. Follow the official instructions for easy installation.

Deciding on a CNI

We support two different CNIs, that being Flannel and Cilium. Flannel is known for it's simplicity and Cilium for it's advanced even service mesh like features.

Flannel

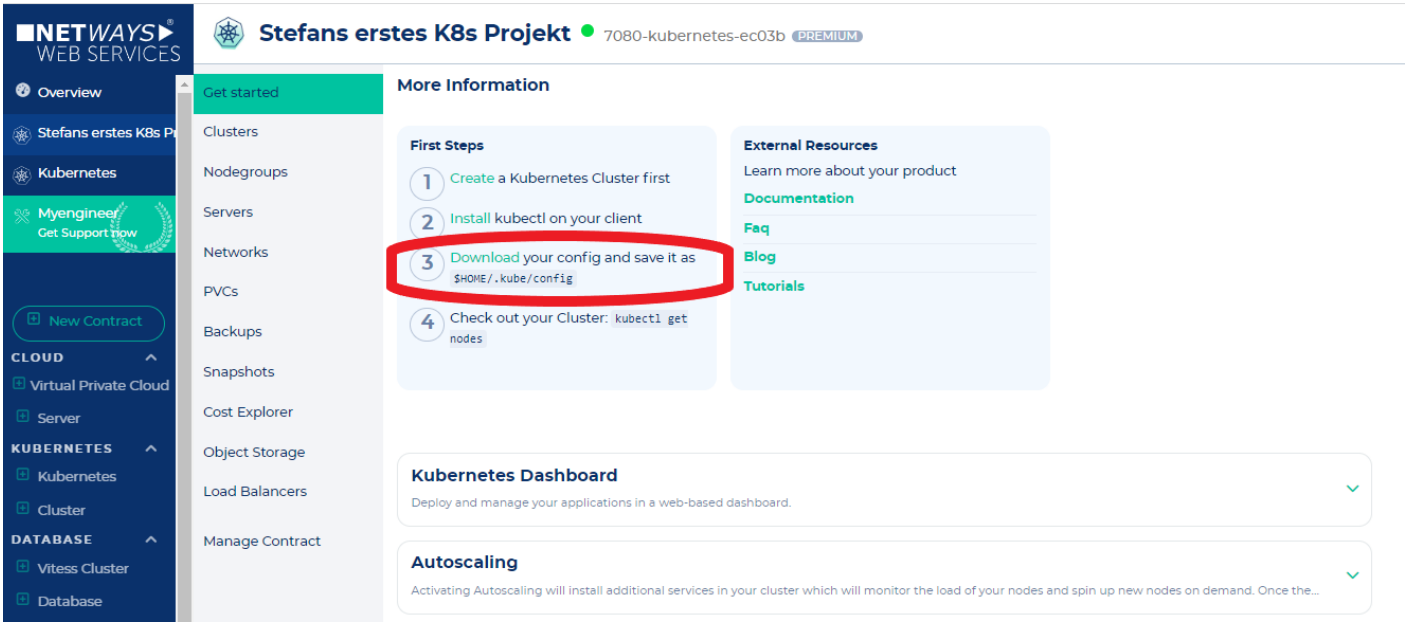
Flannel focuses on the integral part that is the network connection itself. It does not provide any NetworkPolicies or traffic encryption, but it is rock solid when it comes to inter pod communication. That makes it a good choice if you want to chain CNIs and/or add a Service-Mesh on top of it.

Cilium

If you are interested in more advanced CNI features like NetworkPolicies, traffic encryption, mutal TLS and network Observability, Cilium is the right choice for you. It can provide many features that would otherwise neccesstiate a full blown ServiceMesh like Istio. You can find out more on their website: https://cilium.io.

Connect to the created cluster

Now that the cluster is built and kubectl is already installed, it is time to connect to the cluster. This is done by clicking on "Download-Config" in the context menu next to the cluster in the NWS backend.



The just downloaded Config must now be moved to the correct place.

To do this, we create a directory in the user home that is still required (if it does not exist) and copy the file into it. At the end we adjust the rights. All work is done as a local user:

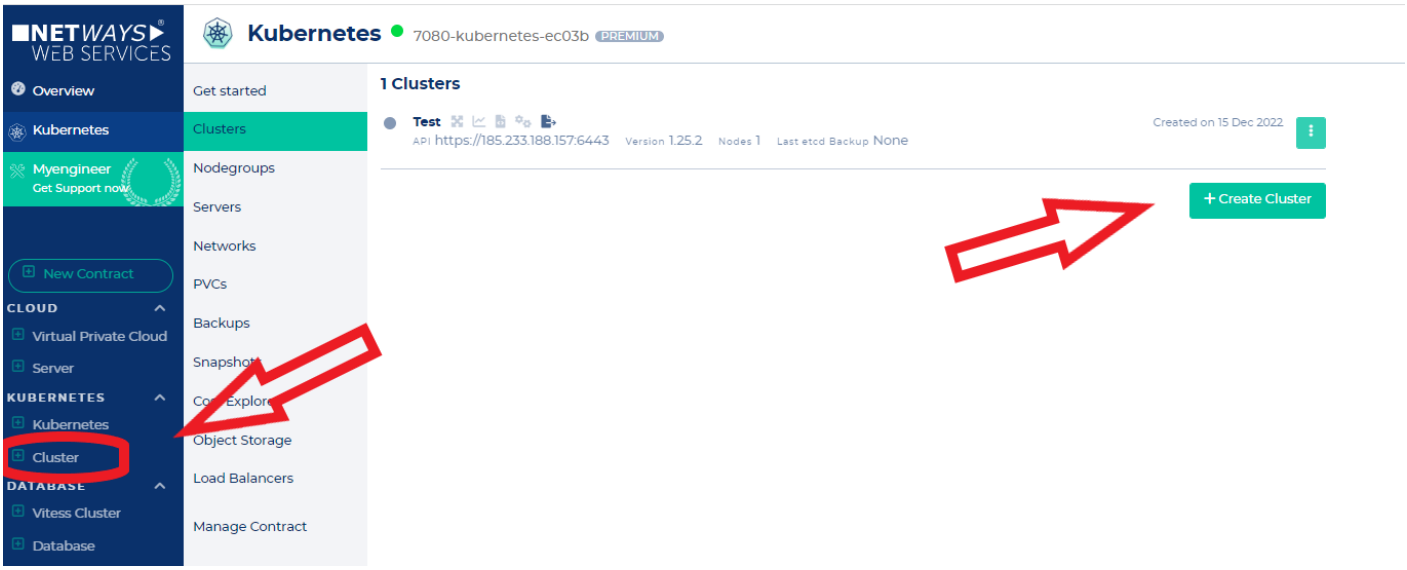
```
mkdir ~/.kube
mv ~/Downloads/config ~/.kube/
chmod 0600 ~/.kube/config
```

Kubectl should now automatically use the new config. To see if works we can try to list all cluster nodes as done below. If you use NWS-ID, your browser will open for authentication. After that, just switch back to the terminal.

```
$ kubectl get nodes
NAME                                  STATUS   ROLES    AGE   VERSION
workshop-virgin2-ytwudzfwjco6-master-0   Ready    master   17h   v1.23.1
workshop-virgin2-ytwudzfwjco6-node-0     Ready    <none>   17h   v1.23.1
```

Starting further clusters in the same K8s project

If you need a staging cluster and a production cluster, this can easily be done in the same Kubernetes app - but you'll need to launch separate clusters for each.



Multiple clusters can be started in the same product, even with the same subnet.
The respective systems of the cluster are also provided normally in the same subnet, but then receive different addresses from the DHCP in the same subnet.
The communication with each cluster is done via its own Config and thus each cluster can also work only with its workers.

Caution!

With NWS Managed Kubernetes you have full control over all resources in your cluster. Please adhere to these rules:

- do not schedule your own pods on master nodes
- keep out of the kube-system namespace *

* unless you want to do stuff that is well documented on this site. For example setting static hostnames in CoreDNS.

Scheduling your own workloads on master nodes could cause API downtimes due to OOM events. Editing or creating resources in the kube-system namespace could break critical cluster services.