


# Getting Started


- [Create NWS Managed Kubernetes](#)
- [Install kubectl and kubelogin](#)
- [Deciding on a CNI](#)
- [Connect to the created cluster](#)
- [Starting further clusters in the same K8s project](#)
- [Caution!](#)

# 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](#).

**Create Kubernetes**

**Kubernetes**  
Your Happy-Go-Lucky Container Orchestration Service Hosted in Germany.  
[Learn more](#)



**Cluster in Minutes**  
Define your cluster status, set the number of nodes, download your kubeconfig and start using kubectl. In just a few steps, your Kubernetes Cluster is ready. We take care of its high availability, updates, backups and a stable underlying cloud infrastructure.

**Managed Control Plane**  
The Control Plane is the brain of the cluster that takes the action and enforces your desired state. You can decide between a standalone or a high available control plane. We monitor and operate it 24 hours a day and 7 days a week.

**Persistent Volume Claims**  
Managing data within containers can be tricky. In theory there is only stateless workload for your pods, for everything else we support Persistent Volume Claims. Store your data, e.g. for the operation of databases, beyond the life cycle of containers.

**Horizontal Scaling and Autoscaling**  
Scale your application with a simple command or use the Kubernetes web interface. Depending on CPU usage or other metrics, new containers can be started automatically. Autoscaling detects when you actually run out of space in your cluster and starts new VMs, which collapse again when you no longer need them. Saves

0.0 €/Month [See pricing table](#)

Create

**NETWAYS WEB SERVICES**  
Overview  
Kubernetes  
Myengineer Get Support now  
New Contract  
CLOUD  
Virtual Private Cloud  
Server  
KUBERNETES  
Kubernetes  
Cluster  
DATABASE  
Vitess Cluster  
Database  
APPS  
Icinga 2 Master  
Icinga 2 Satellite  
Rocket.Chat  
Nextcloud  
SuiteCRM  
Gitlab CE  
Request Tracker  
Gitlab EE  
Cachet  
Jitsi

**Kubernetes** 7080-kubernetes-ec03b PREMIUM  
Get started  
Clusters  
Nodegroups  
Servers  
Networks  
PVCs  
Backups  
Snapshots  
Cost Explorer  
Object Storage  
Load Balancers  
Manage Contract

**More Information**  
**First Steps**

- 1 Create a Kubernetes Cluster first
- 2 Install kubectl on your client
- 3 Download your config and save it as `$HOME/.kube/config`
- 4 Check out your Cluster: `kubectl get nodes`

**External Resources**  
Learn more about your product:  
[Documentation](#)  
[Faq](#)  
[Blog](#)  
[Tutorials](#)

**Kubernetes Dashboard**  
Deploy and manage your applications in a web-based dashboard.

**Autoscaling**  
Activating Autoscaling will install additional services in your cluster which will monitor the load of your nodes and spin up new nodes on demand. Once the...

**Monitoring with Prometheus and Grafana**  
If you would like to get an insight into various kinds of metrics, view the current usage of your nodes or setup alerts for some services, Prometheus and Graf...  
You can enable this feature individually for your Kubernetes clusters!

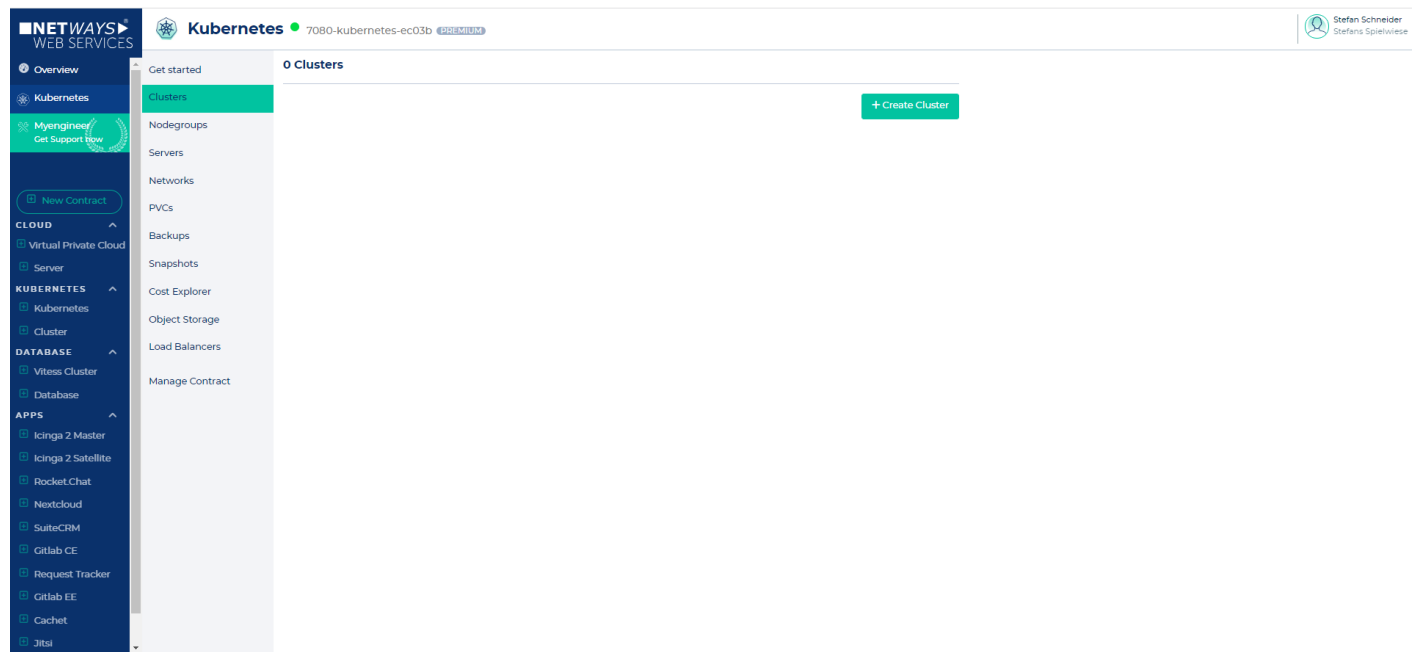
**Logging with Loki and Grafana**  
By gathering logs from all running pods and services in your Kubernetes Cluster, Loki and Grafana provide a great and easy to use solution to query and vie...  
You can enable this feature individually for your Kubernetes clusters!

**Loki LogCLI**  
Analyze all the logs in Loki right from your local CLI. Installation options:

✓ User is ready

✓ Project is ready

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 availability 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](mailto: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

## Control Plane Flavor

Choose a flavor. The chosen flavors are not editable after creation.

S1	S2
----	----

4 vCPU 4 GB RAM 50 GB Storage s1.medium	6 vCPU 8 GB RAM 75 GB Storage s1.large	8 vCPU 16 GB RAM 100 GB Storage s1.xlarge	12 vCPU 32 GB RAM 150 GB Storage s1.xxlarge	16 vCPU 64 GB RAM 200 GB Storage s1.xxxlarge
--	---	--	--	---

## Worker Nodes

Select the initial number of worker nodes. You can change the number of Nodes again later.

-	2	+
---	---	---

## Worker Node Flavor

Choose the flavor of your worker nodes. Later it is possible to add additional Nodegroups to your running cluster with other types of flavors.

S1	S2
----	----

4 vCPU 4 GB RAM 50 GB Storage s1.medium	6 vCPU 8 GB RAM 75 GB Storage s1.large	8 vCPU 16 GB RAM 100 GB Storage s1.xlarge	12 vCPU 32 GB RAM 150 GB Storage s1.xxlarge	16 vCPU 64 GB RAM 200 GB Storage s1.xxxlarge
--	---	--	--	---

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](https://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 its simplicity and Cilium for its 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



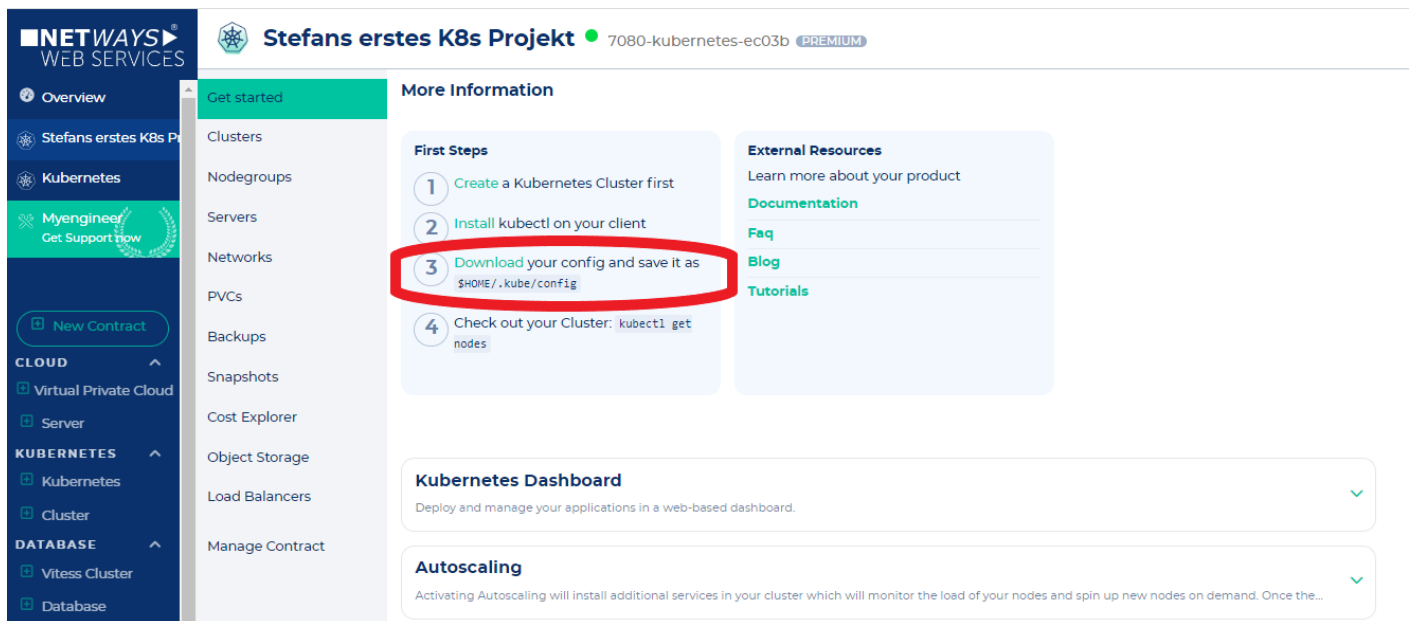
If you are interested in more advanced CNI features like NetworkPolicies, traffic encryption, mutual TLS and network Observability, Cilium is the right choice for you. It can provide many features that would otherwise necessitate a full blown ServiceMesh like [Istio](https://istio.io). You can find out 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
```

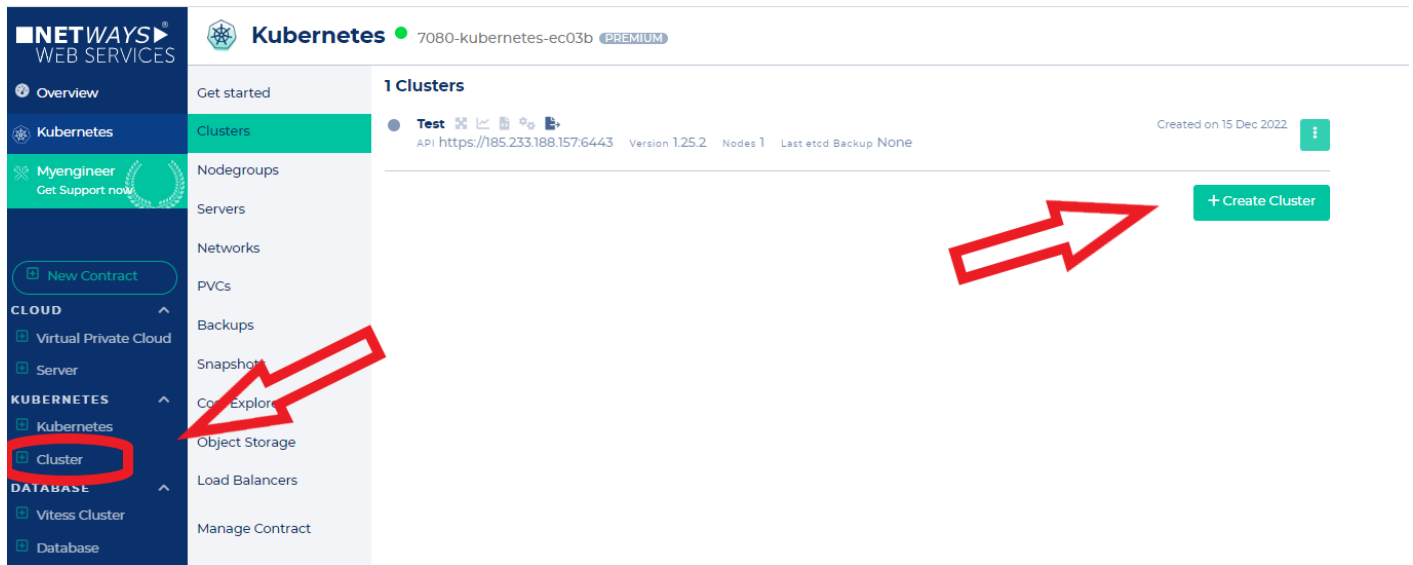
Kubectl should now automatically use the new config. To see if works we can try to list all cluster nodes as done below. If you use NWS-ID, your browser will open for authentication. After that, just switch back to the terminal.

```
$ kubectl get nodes
```

NAME	STATUS	ROLES	AGE	VERSION
workshop-virgin2-ytwudzfjco6-master-0	Ready	master	17h	v1.23.1
workshop-virgin2-ytwudzfjco6-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.