**//updated grammatical mistakes**

**# port-map-operator**

A `Load balancer` `Service` type implementation for small home clusters.  
Maps the ports from your router to a Kubernetes cluster nodevia the [Port Control Protocol](https://tools.ietf.org/html/rfc6887).  
It does not perform real load balancing of any kind, but just takes care of the port forwarding so traffic can reach the cluster node.Kubernetes still does its internal service-level load balancing.  
**## Requirements**  
- Kubernetes cluster that can run `Pod’s with `host Network: true`- Router that supports [PCP](https://tools.ietf.org/html/rfc6887) for port mapping- No other controllers implementing `LoadBalancer` `Service` type running in the cluster (to avoid conflicts)  
**## Deployment**  
See the `config` dir.  
Use the `config/default` as a Kustomization base, don’t forget to update the image to a non-rolling docker tag (using rolling tags like `latest`, `nightly`or `master` is not recommended).  
If you have issues with PCP server autodiscovery, you can specify the addressmanually. A typical value would be the address of your router with port `5351`(standard PCP server port), or `5350`.To configure the address, add the argument in the form of`--pcp-server=192.168.1.1:5351` to the container command.  
**## Usage**  
After we install the operator, just create a `Service` with`type: LoadBalancer`, and the operator will map the port and fill in the`externalIP`.  
This is how it should look like:  
```shell$ kubectl get svcNAME         TYPE           CLUSTER-IP     EXTERNAL-IP     PORT(S)          AGEpodinfo      LoadBalancer   10.98.1.2      1.2.3.4         1234:31234/TCP   1h```  
The port map should also be visible in your router UI, for instance at the OpenWRT it can be found on the UPnP page.  
If everything works, you (or anyone on the internet) should be able to reach the service via the IP and the port of the service.In the example above - the service will be available at `1.2.3.4:1234`.  
**## Caveats**  
**### Mapping ports lower than 1024**  
When trying to map ports in the range 0-1024, you may find that the mapping does not work. This is a security measure taken by the PCP servers to prevent abuse.You should be able to tune your PCP server (router) to allow port maps in the 0-1024 for your Kubernetes nodes if you really want to.See the documentation on your PCP server / router for more info.