



## ***Utilidad K9S***

**Ricoh Sistemas**

**Ver. 1.0, 2022-04-22**

# Índice de contenidos

- 1. *Objetivo* .....1
- 2. *Documentación de referencia* .....2
- 3. *Instalación* .....3
- 4. *Manejo de la herramienta* .....4

# 1. Objetivo

Instalación y configuración del software K9SCli el cual permite la administración de varios cluster de kubernetes desde un punto centralizado.

## 2. Documentación de referencia

- <https://k9scli.io/>
- <https://github.com/derailed/k9s>
- <https://k9scli.io/topics/skins/>

## 3. Instalación

*Creamos el usuario*

```
useradd -m -d /home/k9suser k9suser
```

*Creamos una carpeta para descargar los fuentes*

```
mkdir source  
cd source  
git clone https://github.com/derailed/k9s.git .
```

*Se instalan las Herramientas de Desarrollo*

```
yum -y groupinstall "Development Tools"  
yum -y install go
```

*Instalamos*

```
sudo install ./k9s /usr/bin
```

*Instalar el metric server en el entorno de Kubernetes para que se pueda mostrar estadísticas de CPU y Memoria.:*

```
kubectl apply -f https://github.com/kubernetes-sigs/metrics-server/releases/latest/download/components.yaml  
-O metrics-server-components.yaml  
kubectl patch deployment metrics-server -n kube-system --type 'json' -p '[{"op": "add", "path":  
"/spec/template/spec/containers/0/args/-", "value": "--kubelet-insecure-tls"}]'
```

## 4. Manejo de la herramienta

Desde una usuario que tenga instalado kubectyl y el fichero de kubernetes config en su perfil, ejecutar

k9s

Una que se haya accedido al interface gráfico de la herramienta existen un conjunto de opciones para la gestión de componentes para lo cual se debe iniciar una linea de comandos con el simbolo ":" y a continuación la opción correspondiente que se puede autocompletar con el tabulador ejemplos:

- pods
- deploys
- services

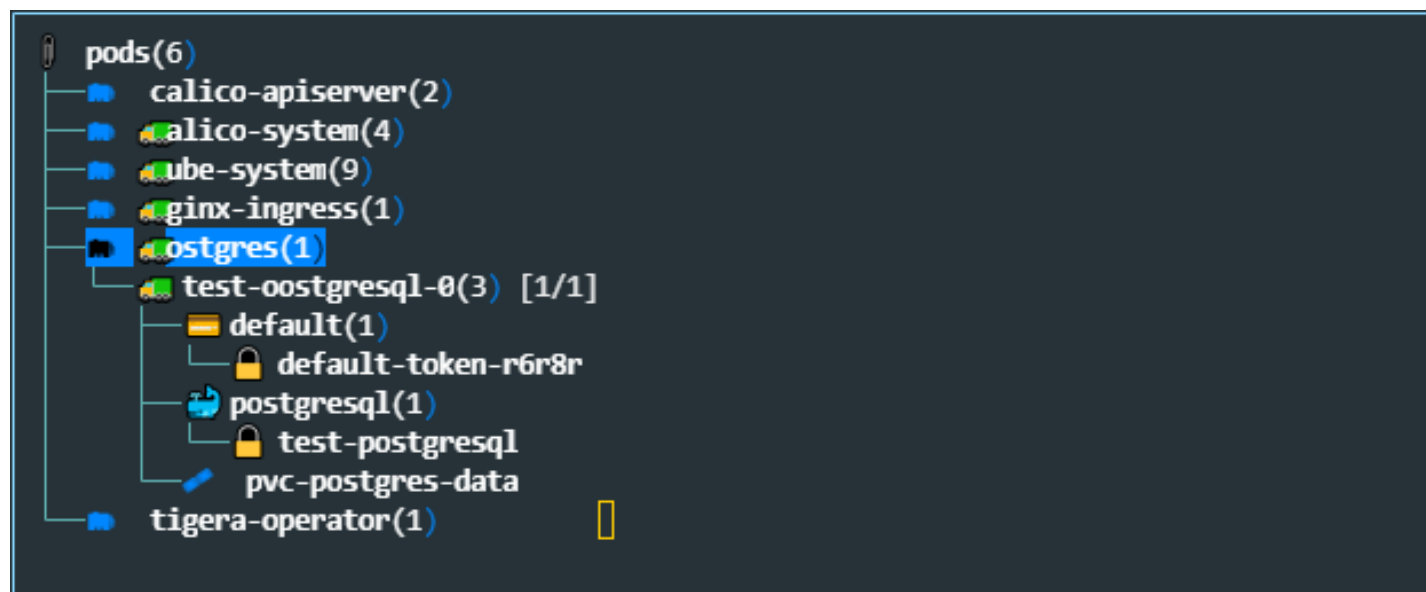
NAMESPACE	NAME	PF	READY	RESTARTS	STATUS	CPU	MEM	%CPU/R	%CPU/L	%MEM/R	%MEM/L	IP	NODE	AGE
calico-apiserver	calico-apiserver-6c6f86b484-pkg9x	•	1/1	11	Running	5	62	n/a	n/a	n/a	n/a	192.168.25.233	k8s-server1	56d
calico-system	calico-kube-controllers-77c48f5f64-k8679	•	1/1	9	Running	3	42	n/a	n/a	n/a	n/a	192.168.25.234	k8s-server1	56d
calico-system	calico-node-fp2ef	•	1/1	14	Running	34	215	n/a	n/a	n/a	n/a	128.0.2.100	k8s-server1	56d
calico-system	calico-node-rj5nk	•	1/1	9	Running	23	234	n/a	n/a	n/a	n/a	128.0.2.101	k8s-server1	56d
calico-system	calico-typha-7899c895bc-wm4nc	•	1/1	20	Running	2	43	n/a	n/a	n/a	n/a	128.0.2.100	k8s-server1	56d
kube-system	coredns-64897985d-b999d	•	1/1	8	Running	1	21	1	n/a	30	12	192.168.25.236	k8s-server1	56d
kube-system	coredns-64897985d-dq6f4	•	1/1	8	Running	1	30	1	n/a	43	17	192.168.25.235	k8s-server1	56d
kube-system	etcd-k8s-server1	•	1/1	8	Running	17	67	17	n/a	67	n/a	128.0.2.100	k8s-server1	56d
kube-system	kube-apiserver-k8s-server1	•	1/1	9	Running	61	445	24	n/a	n/a	n/a	128.0.2.100	k8s-server1	56d
kube-system	kube-controller-manager-k8s-server1	•	1/1	11	Running	11	81	5	n/a	n/a	n/a	128.0.2.100	k8s-server1	56d
kube-system	kube-proxy-d8865	•	1/1	7	Running	1	37	n/a	n/a	n/a	n/a	128.0.2.101	k8s-server1	56d
kube-system	kube-proxy-kjcp5	•	1/1	8	Running	1	30	n/a	n/a	n/a	n/a	128.0.2.100	k8s-server1	56d
kube-system	kube-scheduler-k8s-server1	•	1/1	12	Running	3	38	3	n/a	n/a	n/a	128.0.2.100	k8s-server1	56d
kube-system	metrics-server-7cf8b65d65-cl6dl	•	1/1	0	Running	3	27	3	n/a	13	n/a	192.168.194.92	k8s-server1	17h
nginx-ingress	nginx-ingress-9bz1q	•	1/1	6	Running	1	41	n/a	n/a	n/a	n/a	192.168.194.81	k8s-server1	52d
postgres	test-postgresql-0	•	1/1	1	Running	4	29	1	n/a	11	n/a	192.168.194.89	k8s-server1	49d
tigera-operator	tigera-operator-59fc55759-jdwmn	•	1/1	17	Running	3	61	n/a	n/a	n/a	n/a	128.0.2.100	k8s-server1	56d

Adicionalmente por cada uno de los elementos a gestionar existen un conjunto de acciones que son ejecutadas con la letra correspondiente

<0> all	<a> Attach	<l> Logs
<1> calico-system	<ctrl-d> Delete	<p> Logs Previous
<2> kube-system	<d> Describe	<shift-f> Port-Forward
<3> postgres	<e> Edit	<s> Shell
<4> calico-apiserver	<?> Help	<f> Show PortForward
<5> default	<ctrl-k> Kill	<y> YAML

Además se pueden utiliza un conjunto de utilidades adicionales como son:

- xray (componente): Muestra en forma de arbol los componentes asociados y dependencias. Por ejemplo "xray pods"



- pulses: Un pequeño dashborad con el estado del cluster

