

kubectl and the Rancher CLI



What are you Learning?

In this lab you'll learn about different ways to interact with Kubernetes clusters running in Rancher. First, you'll connect to them with Kubectl in the Kubernetes UI. Next, you'll use kubectl with a kubeconfig file, provided by Rancher. Lastly, you'll interface with Kubernetes via the Rancher CLI, with an API Token, provided by Rancher.

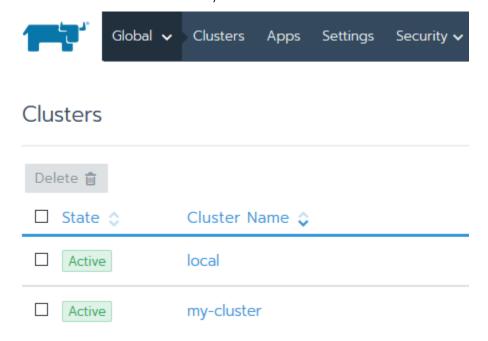
Why is it important?

The Rancher CLI can interact with Kubernetes via kubectl directly, but it also has many of the features provided by the Rancher UI. This is very useful for automating workflows via scripts, or quick access to your clusters without having to log into Rancher via a web browser.

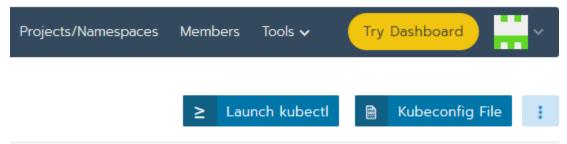
Using kubectl

From the UI

1. Click on the cluster with which you wish to interact.



2. Click on Launch Kubectl



3. Try out a few commands, command-line completion has been activated as well. The click Close

≥ Shell: my-cluster

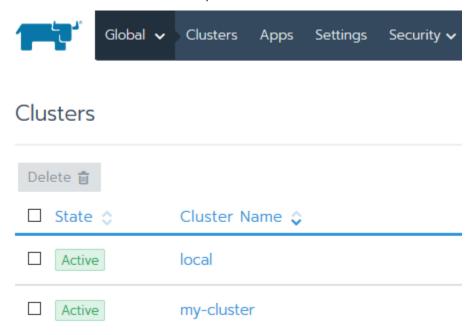
```
# Run kubectl commands inside here
# e.g. kubectl get all
> kubectl get all
                                               EXTERNAL-IP
NAME
                                 CLUSTER-IP
                     TYPE
                                                             PORT (S)
                                                                       AGE
service/kubernetes
                     ClusterIP
                                 10.43.0.1
                                                             443/TCP
                                                                       4d17h
                                               <none>
> kubectl get pods
No resources found in default namespace.
> kubectl get pods -n kube-system
NAME
                                           READY
                                                   STATUS
                                                               RESTARTS
                                                                          AGE
canal-6msbq
                                           2/2
                                                   Running
                                                                          4d17h
                                                                          18h
coredns-7c5566588d-hskbk
                                           1/1
                                                   Running
                                                               0
coredns-autoscaler-65bfc8d47d-924px
                                           1/1
                                                   Running
                                                               0
                                                                          18h
metrics-server-6b55c64f86-tlbn8
                                           1/1
                                                   Running
                                                               0
                                                                          18h
                                           0/1
rke-coredns-addon-deploy-job-mh28z
                                                   Completed
                                                               0
                                                                          18h
rke-ingress-controller-deploy-job-cxt7x
                                           0/1
                                                   Completed
                                                               0
                                                                          4d17h
rke-metrics-addon-deploy-job-jzb4d
                                           0/1
                                                   Completed
                                                               0
                                                                          18h
                                           0/1
                                                   Completed
                                                               0
rke-network-plugin-deploy-job-9qztg
                                                                          18h
> kubectl describe node all-1 .
Name:
                    all-1
Roles:
                    controlplane, etcd, worker
Labels:
                    beta.kubernetes.io/arch=amd64
                    beta.kubernetes.io/os=linux
                    cattle.io/creator=norman
                    kubernetes.io/arch=amd64
                    kubernetes.io/hostname=all-1
                    kubernetes.io/os=linux
                    node-role.kubernetes.io/controlplane=true
                    node-role.kubernetes.io/etcd=true
                    node-role.kubernetes.io/worker=true
Annotations:
                    flannel.alpha.coreos.com/backend-data: {"VtepMAC":"6a:36:27:80:e9:ed"}
```

Close

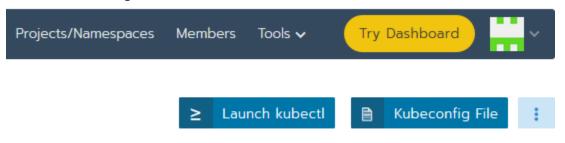
From kubectl

1. If you haven't already install kubectl

2. Click on the cluster with which you wish to interact.



3. Click on Kubeconfig File



- 4. Either copy this file to ~/.kube/config or setup your environment variables to <u>use multiple</u> configuration files.
- 5. At this point you should be able to run similar commands from the CLI.

<pre>\$ kubectl get al</pre>	llall-namespaces	
NAMESPACE	NAME	READY
STATUS REST	TARTS AGE	
cattle-system	<pre>pod/cattle-cluster-agent-86796fdbd8-jslwj</pre>	1/1
Running 0	12d	
cattle-system	pod/cattle-node-agent-929gv	1/1
Running 0	12d	
cattle-system	pod/rancher-fb7c56d9d-f4g8k	1/1
Running 1	12d	
cattle-system	pod/rancher-fb7c56d9d-fv4tc	1/1
Running 2	12d	
cattle-system	<pre>pod/rancher-fb7c56d9d-jqsxs</pre>	1/1
Running 1	12d	

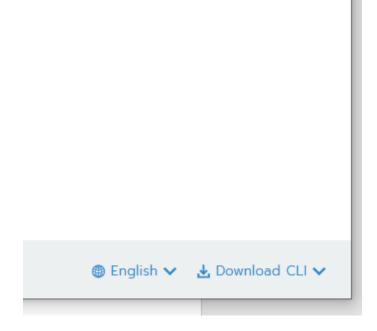
cert-manager	pod/cert-manager-5d9cd85cbb-k6q9z	1/1
Running 1	13d	1 / 1
cert-manager	pod/cert-manager-cainjector-95c885477-w7x4	5 1/1
Running 11	13d	,
cert-manager	<pre>pod/cert-manager-webhook-6ff9487489-q56w2</pre>	1/1
Running 4	13d	
ingress-nginx	<pre>pod/default-http-backend-67cf578fc4-hw7mc</pre>	1/1
Running 0	13d	
ingress-nginx	pod/nginx-ingress-controller-zrvj6	1/1
Running 3	12d	
kube-system	pod/canal-vrvbt	2/2
Running 0	12d	
kube-system	pod/coredns-7c5566588d-gv6jk	1/1
Running 0	12d	
kube-system	pod/coredns-autoscaler-65bfc8d47d-g2bdr	1/1
Running 0	12d	2 /2
kube-system	pod/metrics-server-6b55c64f86-r47vg	1/1
Running 0	12d	0./1
kube-system	pod/rke-coredns-addon-deploy-job-2vvtr	0/1
Completed 0	12d	7 0/1
<pre>kube-system Completed 0</pre>	<pre>pod/rke-ingress-controller-deploy-job-qb6h</pre>	7 0/1
Completed 0 kube-system	pod/rke-metrics-addon-deploy-job-zdf46	0/1
Completed 0	12d	0/ I
kube-system	pod/rke-network-plugin-deploy-job-pljts	0/1
Completed 0	12d	0/1
compiced o	120	
NAMESPACE	NAME TYPE	CLUSTER-
IP EXTERNAL	L-IP PORT(S) AGE	
cattle-system	service/rancher ClusterIP	
10.43.136.179	<none> 80/TCP 12d</none>	
cert-manager	service/cert-manager ClusterIP	
10.43.93.81	<none> 9402/TCP 13d</none>	
cert-manager	service/cert-manager-webhook ClusterIP	
10.43.20.62	<none> 443/TCP 13d</none>	
default	service/kubernetes ClusterIP	10.43.0.1
	43/TCP 13d	
ingress-nginx	service/default-http-backend ClusterIP	
10.43.186.16	<none> 80/TCP 13d</none>	
kube-system	service/kube-dns ClusterIP	
10.43.0.10	<pre><none> 53/UDP,53/TCP,9153/TCP 13d</none></pre>	
kube-system	service/metrics-server ClusterIP	
10.43.2.20	<none> 443/TCP 13d</none>	
NAMECDACE	NAME	DECTRER
NAMESPACE		DESIRED
CURRENT READY	UP-TO-DATE AVAILABLE NODE SELE	

CTOR	AGE				
cattle-system	daemonset.apps,	/cattle-node	-agent	1	
1 1	1	1	<none></none>		
	12d				
ingress-nginx	daemonset.apps,	/nginx-ingre	ss-controller	1	
1 1	1	1	<none></none>		
	13d				
kube-system	daemonset.apps/	/canal		1	
1 1	1	1	kubernete		
s.io/os=linux	13d				
NAMESPACE	NAME			READY	
UP-TO-DATE AVA				ILL ID I	
cattle-system	deployment.apps	s/cattle_clu	stan-agent	1/1	1
1 12d	иертоушенстварр:	s/cattle-tiu	ster -agent	1/1	-
	doployment app	- /nanchan		2 / 2	3
cattle-system 3 12d	deployment.apps	s/rancher		3/3	5
	doubles mont one	-/	0.10	1 /1	1
cert-manager	deployment.apps	s/cer.r-manag	er.	1/1	1
1 13d		,			
cert-manager	deployment.apps	s/cert-manag	er-cainjector	1/1	1
1 13d					
cert-manager	deployment.apps	s/cert-manag	er-webhook	1/1	1
1 13d					
ingress-nginx	deployment.apps	s/default-ht [.]	tp-backend	1/1	1
1 13d					
kube-system	deployment.apps	s/coredns		1/1	1
1 13d					
kube-system	deployment.apps	s/coredns-au	toscaler	1/1	1
1 13d	. , ,				
kube-system	deployment.apps	s/metrics-se	rver	1/1	1
1 13d		,		,	
NAMESPACE	NAME				
DESIRED CURREN		=			
cattle-system			ster-agent-8679	6fdhd8	
1 1	1 120		ster -agent-6075	orabao	
			7cE6d0d		
cattle-system	replicaset.apps		/ C30U3U		
3	3 120		5 10 105 LI		
cert-manager	replicaset.apps		er-5d9cd85cbb		
1 1	1 130				
cert-manager			er-cainjector-9	5c885477	
1 1	1 130				
cert-manager			er-webhook-6ff9	487489	
1 1	1 130				
ingress-nginx			tp-backend-67cf	578fc4	
1 1	1 130	d			

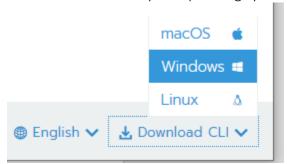
```
kube-system
                replicaset.apps/coredns-7c5566588d
                            13d
          1
kube-system
                replicaset.apps/coredns-autoscaler-65bfc8d47d
                            13d
                replicaset.apps/metrics-server-6b55c64f86
kube-system
                            13d
          1
NAMESPACE
              NAME
COMPLETIONS
              DURATION
                         AGE
              job.batch/rke-coredns-addon-deploy-job
                                                             1/1
kube-system
2s
           12d
              job.batch/rke-ingress-controller-deploy-job
kube-system
                                                             1/1
           12d
3s
             job.batch/rke-metrics-addon-deploy-job
kube-system
                                                             1/1
           12d
3s
              job.batch/rke-network-plugin-deploy-job
                                                             1/1
kube-system
2s
           12d
```

From the Rancher CLI

1. On the bottom right of your screen, there is the Download CLI option.



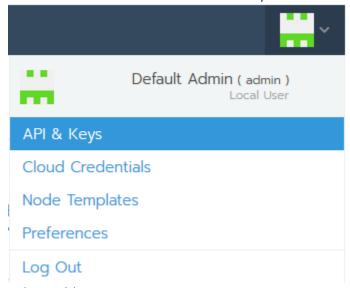
2. Select the version for your operating system.



3. Install this and add it to your path. This will vary for your operating system.

\$ rancher --version
rancher version v2.4.0

4. Now we'll need an API Token. Click on your avatar icon and select API & Keys



5. Select Add Key



Add Key

6. Give it a reasonable description

Add API Key Description API Key for Default Admin

7. Select a reasonable expiration date, consistent with your IT Operations teams' secret rotation policy.

Automatically Expire

- Never
- O A day from now
- O A month from now
- O A year from now
- 8. Select an appropriate scope for this key. No scope will be able to interact with any cluster where you have access. That being said, a cluster-scoped key reflects least-privilege and can access the Authorized Cluster Endpoint. For this lab, use a no scope key.

Scope



Cluster-scoped tokens can be used to interact directly with the Kubernetes API of clusters configured with an Authorized Cluster Endpoint

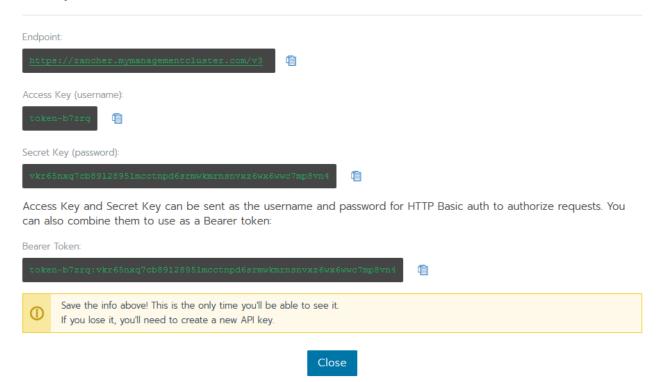
9. Click Create



10. Save these items to a file as this is the last time you will see them. Keep them safe, anyone who has access to this token has access to your credentials and RBAC for the

scope of this API token.

API Key Created



11. Click Close, you'll see your API Token on the list

☐ State ♦	Access Key 💠	Description 💠	Scope 💠	Expires 💠	Created 💠	
Active	helm-token-user- s6t99	token for helm chart deployment	t _{N/A}	Never	12 days ago	i
Active	kubectl-shell-user- s6t99	Access to kubectl shell in the browser	N/A	Never	14 days ago	i
Active	telemetry	telemetry token	N/A	Never	17 days ago	i
Active	token-b7zrq	API Key for Default Admin	Cluster	Never	a few seconds ago	:

12. From your terminal, we'll use the token to login into Rancher and my-cluster. Notice I added a space at the start of my command so the token is not saved to my terminal's history log. Also, if your cluster is using self-signed certificates, you'll want to use the skip-verify option.

\$ ranch	er logintoker	token-7vvhc:k8l4n	pmgr97hvjbk6pq9	j4h2fkkzf7l	
65h6zrzc	fbrfjl9hrx29nr8	name rancher-cer	t		
https://	rancher.mymanage	ementcluster.com/v3			
NUMBER	CLUSTER NAME	PROJECT ID	PROJECT NAME	PROJECT	
DESCRIPT	ION				
1	my-cluster	c-mfcvp:p-pw5sv	System	System	
project created for the cluster					
2	my-cluster	c-mfcvp:p-v5dgv	Default	Default	
project created for the cluster					

3	local		<pre>local:p-jcks8</pre>	System	System
project	created	for th	e cluster		
4	local		local:p-nkv4q	Default	Default
project	created	for th	e cluster		

- 13. You'll notice a list of clusters and projects. A Rancher CLI context is the combination of a cluster and a <u>project</u>.
- 14. Select the Default Project on my-cluster, or whatever the name is you gave you your cluster

Select a Project:2

INFO[0239] Saving config to /home/jason/.rancher/cli2.json

15. Notice your credentials are saved in the cli2.json file. This contains your token keep it safe.

```
$ cat ~/.rancher/cli2.json
{"Servers":{"rancher-cert":{"accessKey":"token-
7vvhc","secretKey":"k8l4npmgr97hvjbk6pq9j4h2fkkzf7l65h6zrzcfbrfjl9hr
x29nr
8","tokenKey":"token-
7vvhc:k8l4npmgr97hvjbk6pq9j4h2fkkzf7l65h6zrzcfbrfjl9hrx29nr8","url":
"https://rancher.mymanagementcl
uster.com","project":"c-mfcvp:p-
v5dgv","cacert":""}},"CurrentServer":"rancher-cert"}
```

16. To run kubectl against this context use rancher kubectl

\$ rancher kubec	tl get nodes			
NAME STATUS	ROLES	AGE	VERSION	
all-1 Ready	controlplane,etcd,worker	4d23h	v1.17.5	
\$ rancher kubec	tl get allall-namespaces			
NAMESPACE	NAME			READY
STATUS RES	TARTS AGE			
cattle-system	<pre>pod/cattle-cluster-agent-58</pre>	d884cfcd	-m6rrn	1/1
Running 0	4d23h			
cattle-system	<pre>pod/cattle-node-agent-2qtjs</pre>			1/1
Running 0	4d23h			
cattle-system	pod/kube-api-auth-w9vv6			1/1
Running 0	4d23h			
ingress-nginx	<pre>pod/default-http-backend-67</pre>	cf578fc4	-jtr6r	1/1
Running 0	4d23h			
ingress-nginx	<pre>pod/nginx-ingress-controlle</pre>	r-5ttvj		1/1
Running 0	4d23h			
kube-system	pod/canal-9j7kd			2/2
Running 0	4d23h			
kube-system	pod/coredns-7c5566588d-rtg2	b		1/1
Running 0	4d23h			
kube-system	pod/coredns-autoscaler-65bf	c8d47d-2	ggf5	1/1
Running 0	4d23h			
kube-system	pod/metrics-server-6b55c64f	86-fpwhv		1/1

Running 0	4d23h		
kube-system	pod/rke-coredns-addon-deploy-job-cb79j		0/1
Completed 0	4d23h		0/ I
kube-system	pod/rke-ingress-controller-deploy-job-s	srhn2	0/1
Completed 0	4d23h	01 0 p z	0/1
kube-system	pod/rke-metrics-addon-deploy-job-d9zvx		0/1
Completed 0	4d23h		0/ =
kube-system	pod/rke-network-plugin-deploy-job-662d	j	0/1
Completed 0	4d23h	,	
NAMESPACE	NAME TYPE	CLI	JSTER-
IP EXTERN	AL-IP PORT(S) AGE		
default	service/kubernetes Cluster	[P 10	.43.0.1
<none></none>	443/TCP 4d23h		
ingress-nginx		[P	
10.43.244.173	<none> 80/TCP</none>	4d23h	
kube-system	service/kube-dns Cluster		
10.43.0.10	<pre><none> 53/UDP,53/TCP,9153/TCP</none></pre>	4d23h	
kube-system	service/metrics-server Cluster		
10.43.243.14	<none> 443/TCP</none>	4d23h	
NAMESPACE	NAME	DES.	IRED
CURRENT READ			
CTOR	AGE	1	
cattle-system 1 1	<pre>daemonset.apps/cattle-node-agent</pre>	1	
1 1	1 1 <none></none>		
cattle-system	daemonset.apps/kube-api-auth	1	
1 1	1 1 <none></none>		
	4d23h		
ingress-nginx	daemonset.apps/nginx-ingress-controller	1	
1 1	1 1 <none></none>		
	4d23h		
kube-system	daemonset.apps/canal	1	
1 1	1 1 kubernete		
s.io/os=linux	4d23h		
NAMESPACE	NAME	READY	UP-
TO-DATE AVAI	LABLE AGE		
cattle-system	deployment.apps/cattle-cluster-agent	1/1	1
	23h		
ingress-nginx		1/1	1
	23h		
kube-system		1/1	1
	23h	4.44	
kube-system	deployment.apps/coredns-autoscaler	1/1	1

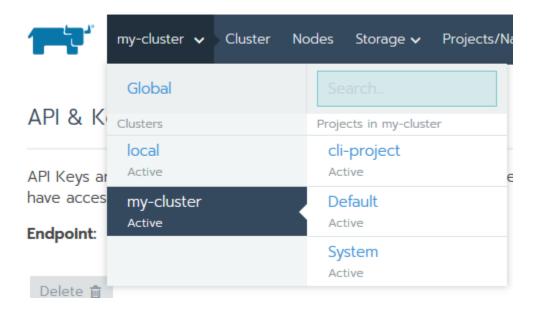
```
4d23h
                deployment.apps/metrics-server
                                                         1/1
                                                                 1
kube-system
            4d23h
NAMESPACE
                NAME
DESIRED
          CURRENT
                    READY
                             AGE
                replicaset.apps/cattle-cluster-agent-58d884cfcd
cattle-system
          1
ingress-nginx
                replicaset.apps/default-http-backend-67cf578fc4
                  4d23h
kube-system
                replicaset.apps/coredns-7c5566588d
                                                                    1
                  4d23h
kube-system
                replicaset.apps/coredns-autoscaler-65bfc8d47d
kube-system
                replicaset.apps/metrics-server-6b55c64f86
                  4d23h
NAMESPACE
              NAME
COMPLETIONS
              DURATION
                         AGE
             job.batch/rke-coredns-addon-deploy-job
                                                              1/1
kube-system
13s
           4d23h
              job.batch/rke-ingress-controller-deploy-job
                                                              1/1
kube-system
           4d23h
kube-system
              job.batch/rke-metrics-addon-deploy-job
                                                              1/1
           4d23h
12s
             job.batch/rke-network-plugin-deploy-job
                                                              1/1
kube-system
10s
           4d23h
17. To always use rancher api access you can alias kubectl.
$ alias kubectl="rancher kubectl"
$ kubectl get nodes
NAME
        STATUS
                 ROLES
                                             AGE
                                                     VERSION
all-1
        Ready controlplane, etcd, worker
                                             4d23h
                                                     v1.17.5
```

Testing That It Works

1. Create a project using the CLI

```
$ rancher project create --description "this project was created vi
a the CLI." cli-project
```

2. Check your web UI to see the new project



This is great! Now your Rancher CLI can modify both Rancher and Kubernetes Clusters. You can learn more about the Rancher CLI commands and their options in the <u>Rancher documentation</u>. Spend some time experimenting.

References

- Install and Setup kubectl https://kubernetes.io/docs/tasks/tools/install-kubectl/
- Configure Access to Multiple Clusters https://kubernetes.io/docs/tasks/access-application-cluster/configure-access-multiple-clusters/
- How the Authorized Cluster Endpoint Works https://rancher.com/docs/rancher/v2.x/en/cluster-admin/cluster-access/ace/
- Project Administration https://rancher.com/docs/rancher/v2.x/en/project-admin/
- Use the Rancher Command Line Interface https://rancher.com/docs/rancher/v2.x/en/cli/