## horizontal line



Kubernetes y Rancher guide

Configuration for rancher and kubernetes

Creator: ezequiel.mora@globant.com

# Recommended introductory courses before continue

<https://app.pluralsight.com/library/courses/docker-kubernetes-big-picture/table-of-contents>

<https://www.katacoda.com/>

<https://www.youtube.com/watch?v=wVlCbPtH41I&t=22s>

<https://eylearning.udemy.com/?next=%2Fcourse%2Fdocker-and-kubernetes-the-complete-guide%2Flearn%2Flecture%2F11482932%3Fstart%3D15>

The final version of the cloud environments we developed with Mauro are the following

FOR DEV - Listening to the branch "MASTER-DEV" of both repositories:

<https://eytools-dev.corp.globant.com:30085/>

FOR QA - Listening to the branch "QA" of both repositories:

<https://eytools-dev.corp.globant.com:30095/>

Remember to continue pushing new changes to the branches "develop" and carefully do a merge into "MASTER-DEV" and "QA" avoiding these files: Dockerfile, Jenkinsfile, nginx.conf, appsettings.json, environment.ts, environment.prod.ts

IMPORTANT NOTE: Before pushing any commit please test and verify that you webapp or service works against docker.

**Important Links:**

Rancher kubernetes clusters - User your active directory credentials to enter

[https://rancher.corp.globant.com/](https://rancher.corp.globant.com/login)

Request credentials to the Rancher admins.

Jenkins CD / CI Builds deployment - Use your <namesurname> and password <name.surname>

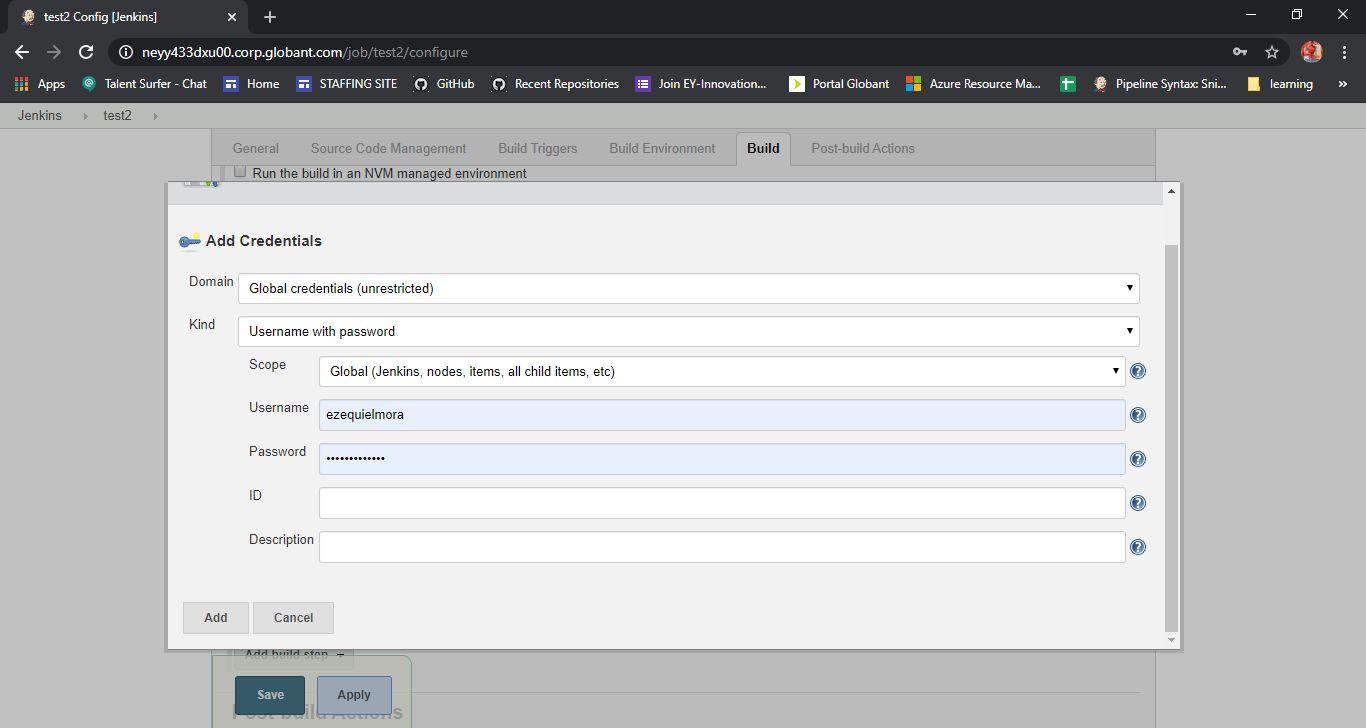
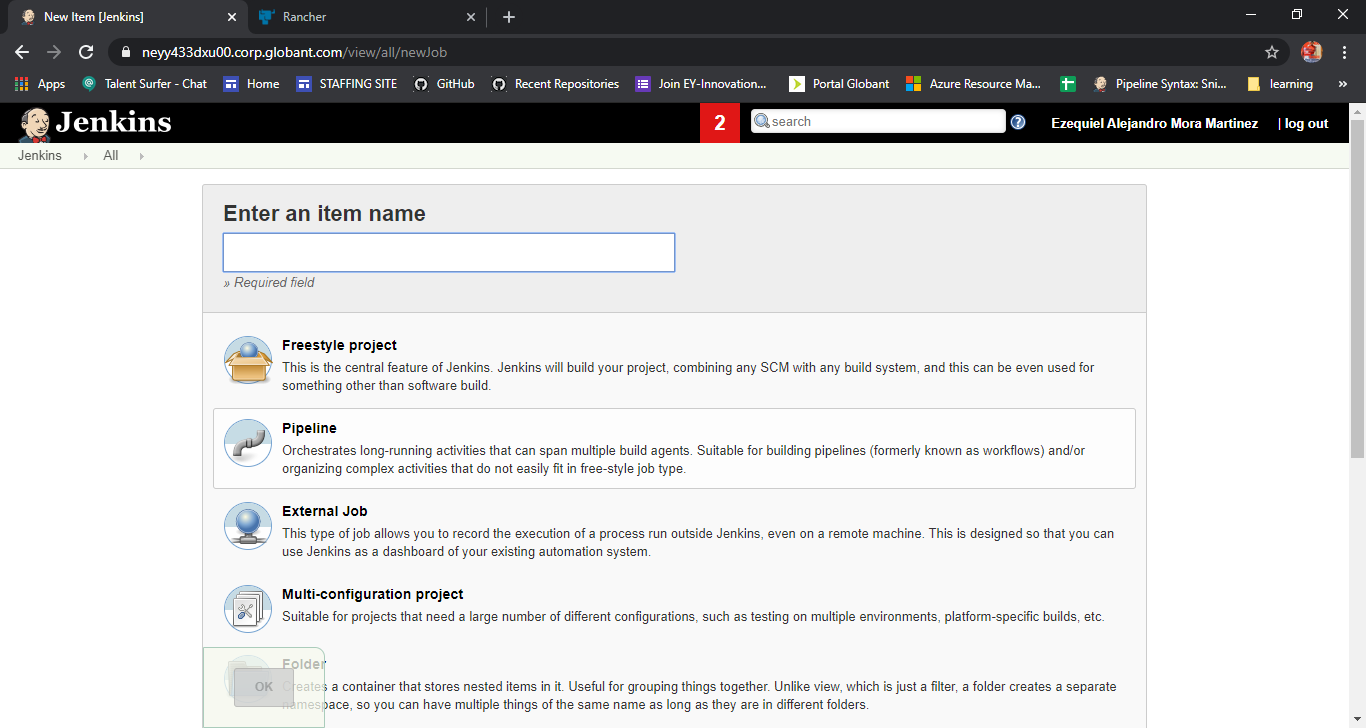
<https://neyy433dxu00.corp.globant.com/>

Request credential to the Jenkins admins.

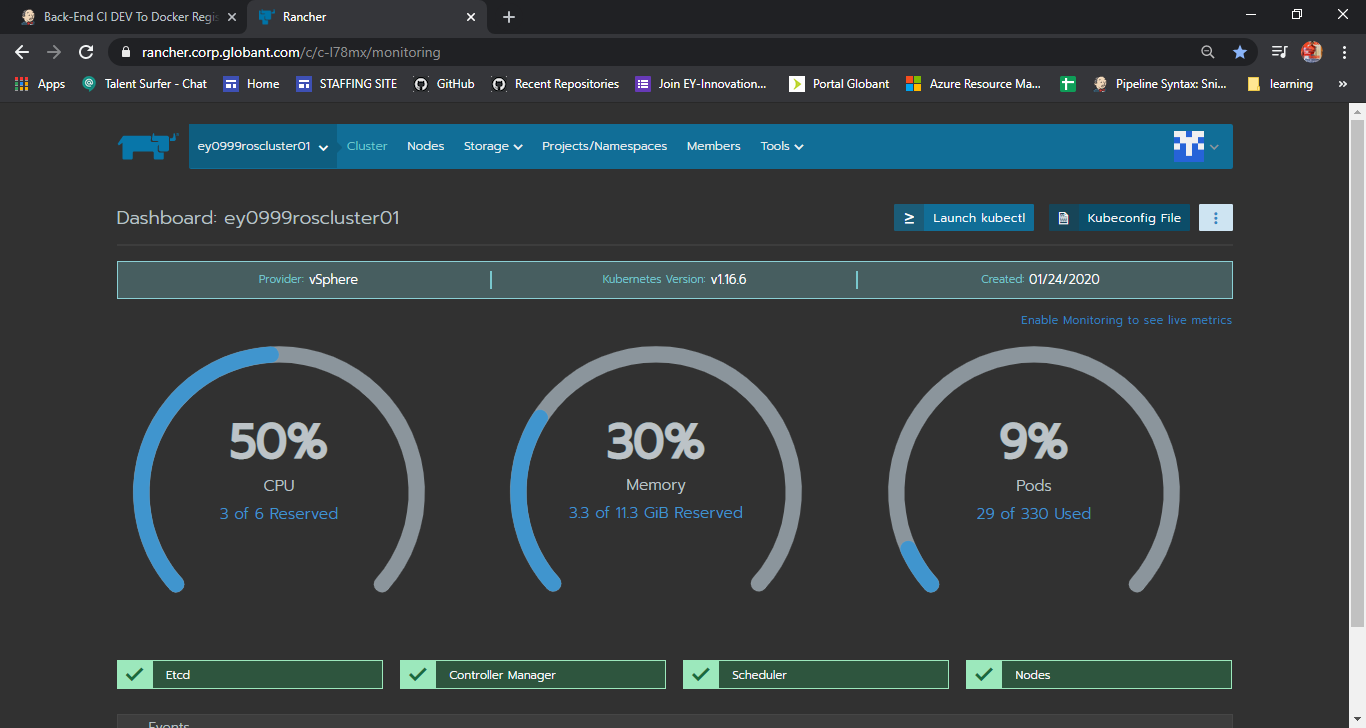
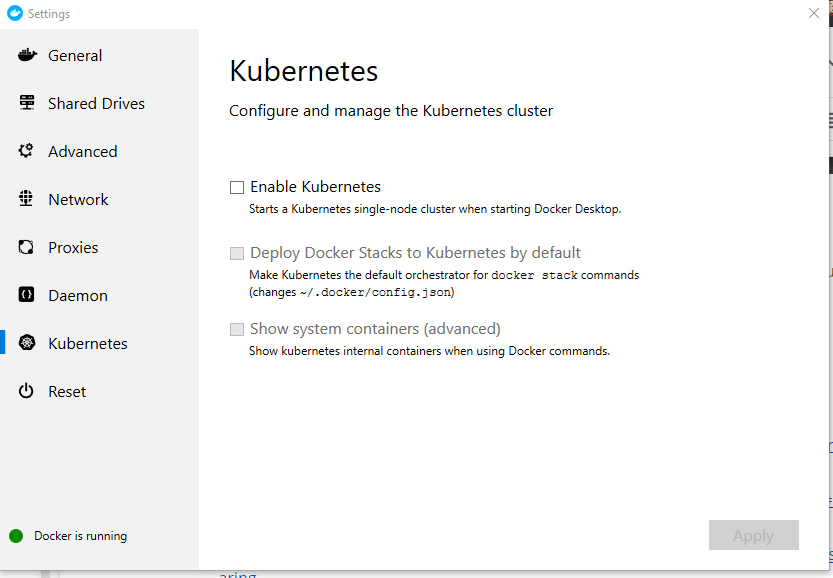
# Note for Readers:

There is no specific order for the step that we will enumerate, these can be performed in different order depending on the configuration that you want to make.

# Jenkins configuration

1. Create the jenkinsfile y push the to the root of the repositories. Jenkinsfile examples can be found in: <https://github.corp.globant.com/ErnstYoungX/TalentSurferBE/blob/MASTER-DEV/Jenkinsfile>
2. Enter to the Jenkins server and go to the left bar and select “Credentials” and create a “Global Credential” for the docker registry server,
3. 
4. Repeat this step and create a credential with your Git username and Password.
5. Add the selected “ID” for that credential and include it in the Jenkinsfile and push that change to the root of the repository
6. Create a “New Item” and select “Pipeline Project”, complete the project using this guide: <https://youtu.be/6tcoRIPBd8s?t=1059>
7. 
8. In the Branch name put the name of the branch that you want Jenkins to listen for continuous deployment for ejemple “QA”.
9. Select the checkbox Poll SCM and inside the text box type “H/5 \* \* \* \*” without the quotes. This way we are telling jenkins to check for new commits every 5 minutes.

# Rancher configuration

1. Enter the Rancher Dashboard and select the main cluster then select the name of the cluster you want to manage.
2. 
3. Then press the button “Kubeconfig File” and download or copy that content to your clipboard.
4. You have to have docker installed and the “Kubernetes Extensions” enabled in docker settings.
5. 
6. Docker and kubernetes cheatsheet and basic commands

<https://www.digitalocean.com/community/cheatsheets/getting-started-with-kubernetes-a-kubectl-cheat-sheet>

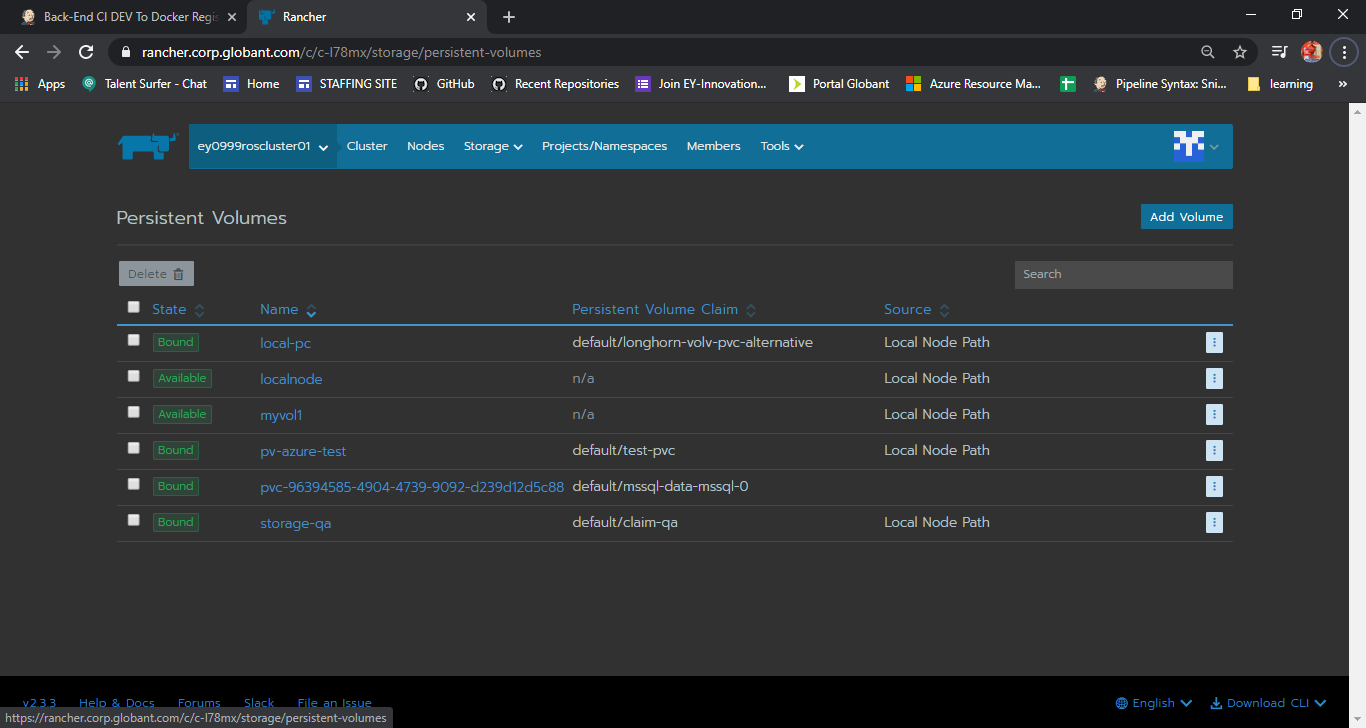
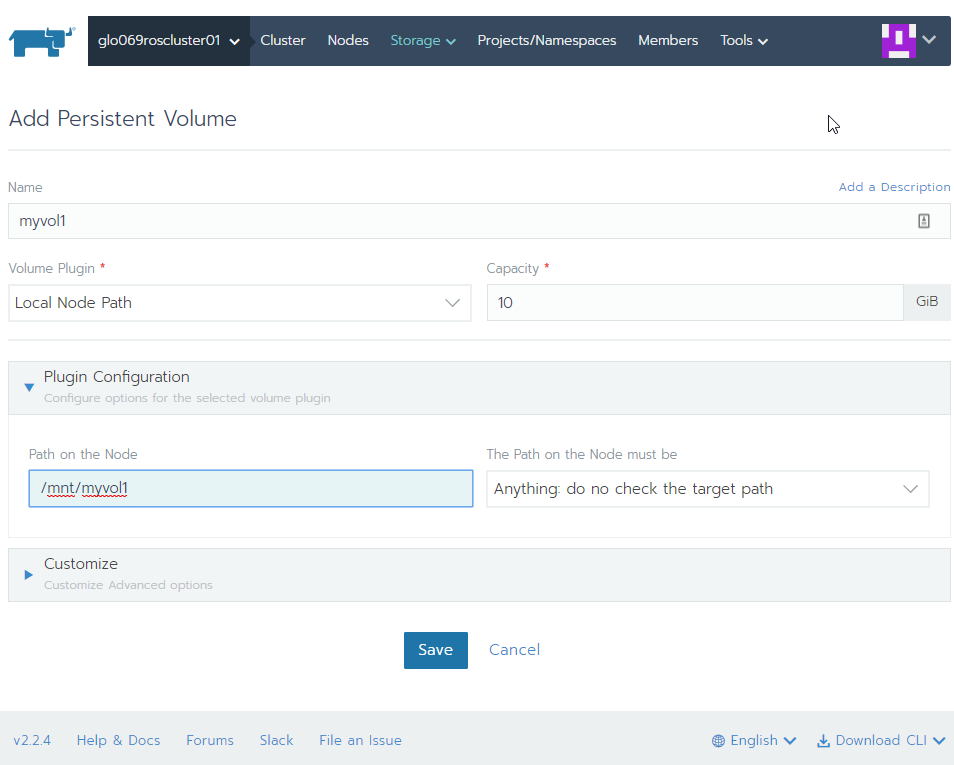
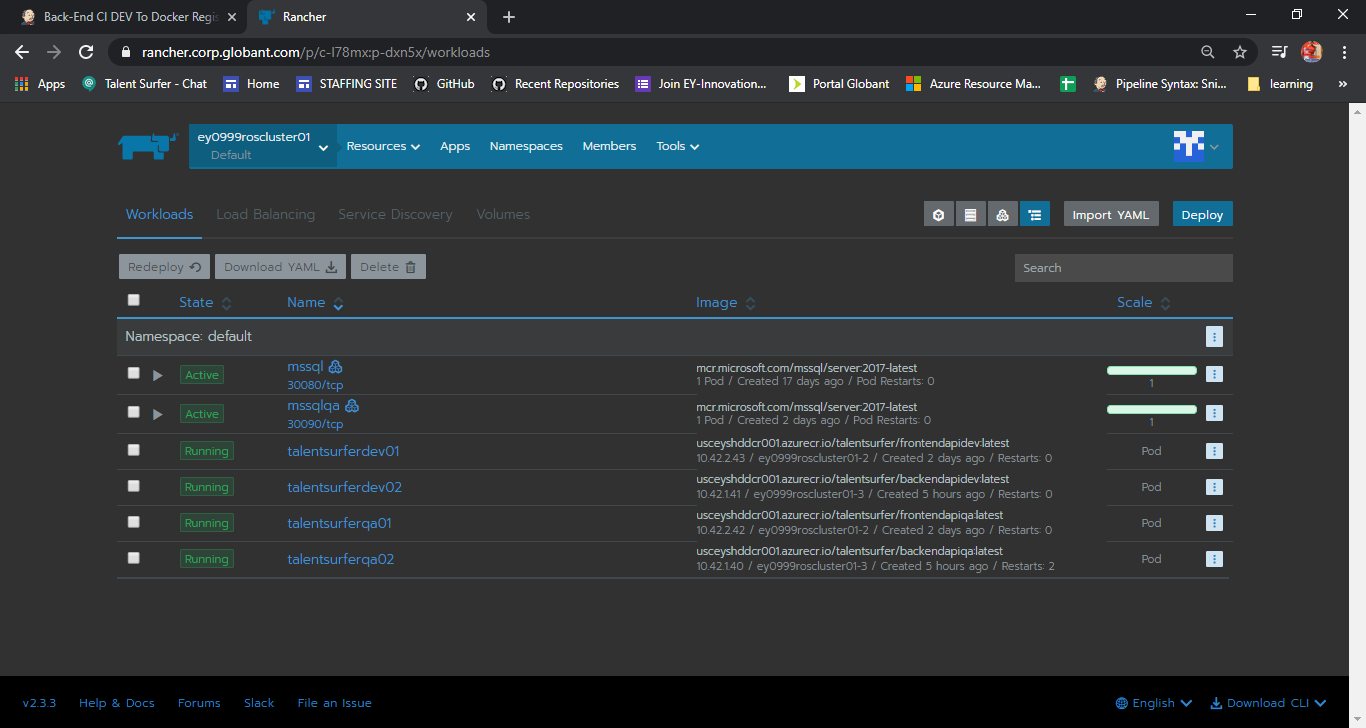
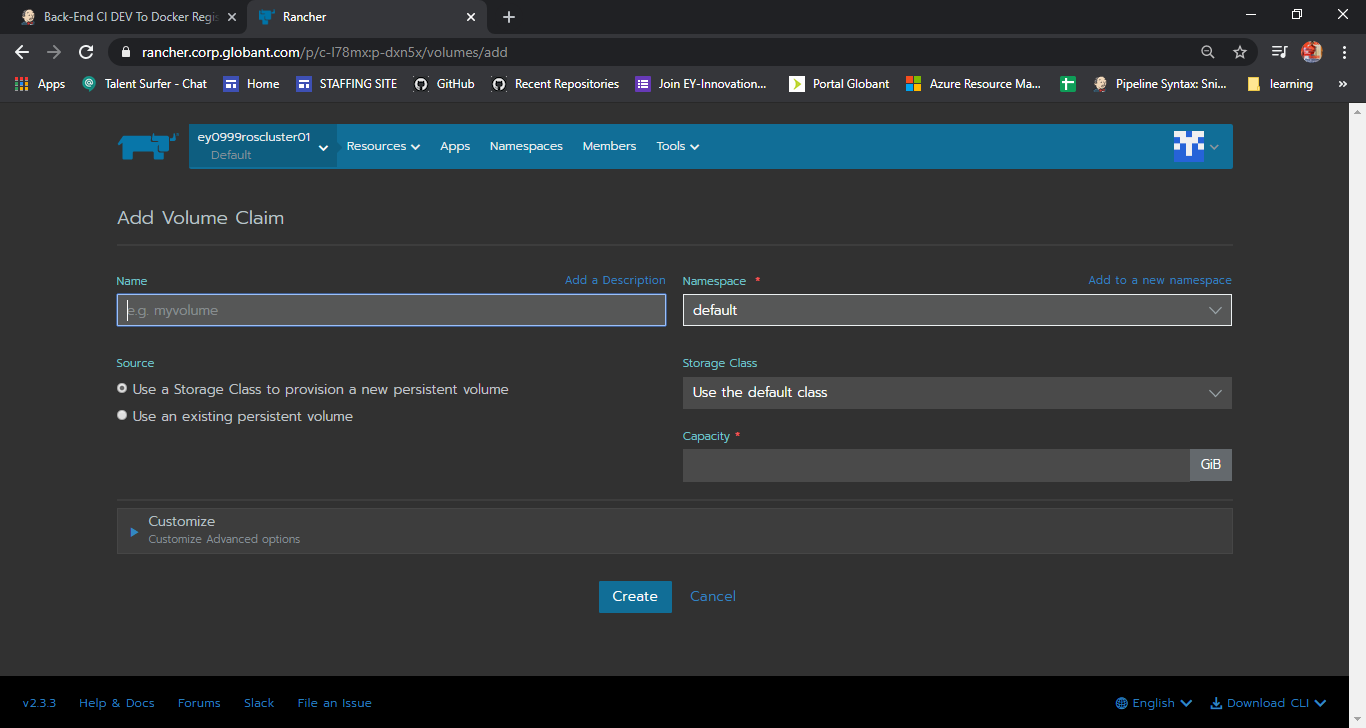
<https://drive.google.com/file/d/1XqHl4z_MJ-WScP0OnKdiPvTj0pOsJBAQ/view?usp=sharing>

<https://drive.google.com/file/d/1_IEfftI78SF2yC0q9npQbpWe2WIfDUsx/view?usp=sharing>

<https://drive.google.com/file/d/19Hf2CLqCxY2AGlOb02FbPC54P9jLSAMR/view?usp=sharing>

1. Copy and paste the content of the downloaded kube.config the file C:/Users/<your\_usename>/.kube/config
2. Open the powershell console and create a secret for the docker registry

$kubectl create secret docker-registry azure--docker-server=<https://usceyshddcr001.azurecr.io> --docker-username=USCEYSHDDCR001 --docker-password=rxGDyhkCpyhuP7DHebd9ZDm0i4NvL/vP --docker-email=tuemail@globant.com

1. Create the storageclass using kubectl or the rancher dashboard and create the persisten volumens and asociate it with the Persistent volume claims.
2. <https://docs.google.com/document/d/1NZUkINhS_IyQTwVPZEZessmgQvokLKV8dIZBXUZBEtQ/edit>
3. Install longhorn using this file <https://github.corp.globant.com/ErnstYoungX/TalentSurferKubernetesYAML/blob/master/longhorn.yaml> or instal StorageOs from the “apps” menu of rancher.
4. <https://rancher.com/project-longhorn-now-available-kubernetes/>
5. Navigate to Storage >> Persistent Volumes and create the PV.
6. 
7. 
8. Later create the PVC in YourClusterName >> Default >> Volumes and associate with the storageclass “longhorn” or “fast” if you used storageOs.
9. 
10. In the yaml of the replicasets of mysql change the value of “claimName” for the same PVC we created in rancher.

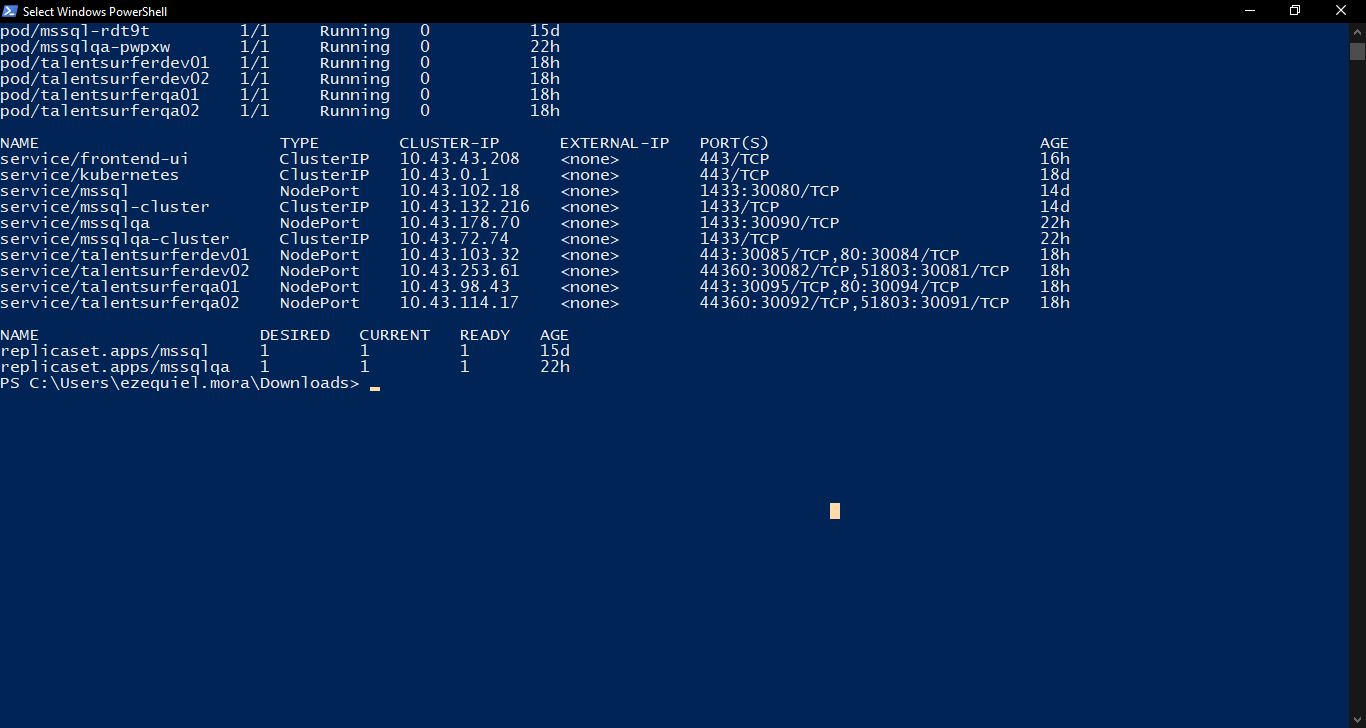
<https://github.corp.globant.com/ErnstYoungX/TalentSurferKubernetesYAML/blob/master/replicaset.yaml>

Install the MSSQL replicaset using kubectl create -f replicaset.yaml

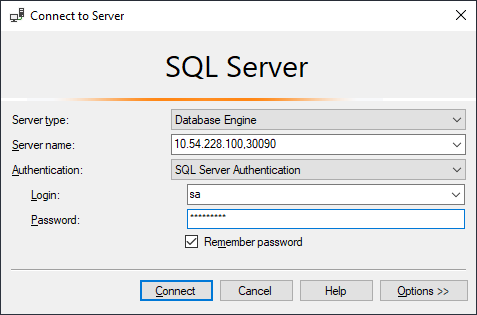
For more examples of different databases please go to: <https://github.com/storageos/deploy/tree/release/1.5.3/k8s/examples>

1. Configure the connectionstring of the backend with the ClusterIP getting by kubectl get all. For example for QA

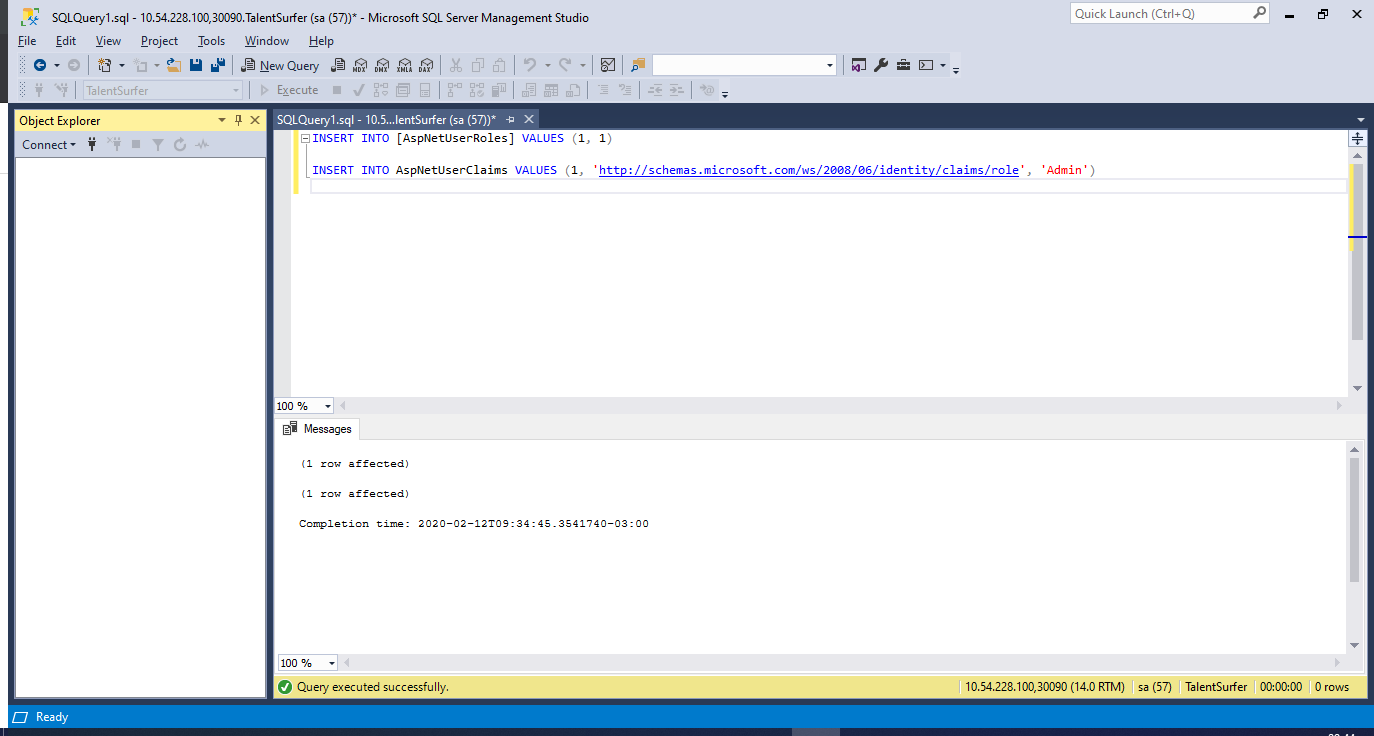
"TalentSurferContext": "Server=10.43.72.74;Database=TalentSurfer;User=sa;Password=Passw0rd1;"



1. We can connect to that database using the nodeport service using the command kubectl get service <nodeport\_service> and search his IP and port in the console.



1. The first time you enter to the database you need to login with an user to execute the following query:



1. Also in the frontend or Angular app, we need to modify the following files environment.ts, environment.prod.ts inserting the domain and port where is pointing our backend.
2. Another necessary step is to create the Return URL in the Google API to enable the SSO Login, this can be done in the Oauth menu of google api and inserting the Return URL of the backend, in Talent Surfer these are:

<https://eytools-dev.corp.globant.com:30082/signin-google>

[https://eytools-dev.corp.globant.com:30092/signin-google](https://eytools-dev.corp.globant.com:30082/signin-google)

Also you need to modify the appsettings.json of the web api to add the “ClientID” and the “ClientSecret” provided by google. For example:

Client ID: [950000135914-7a955fafps3rca2kk9nrd379mdpc2uam.apps.googleusercontent.com](http://950000135914-7a955fafps3rca2kk9nrd379mdpc2uam.apps.googleusercontent.com/)

Secret: tPKm9jcXVDGiorNvi3N4xOvA

NOTE: This step is complex and you may need the help of Martín Méndez.

1. Now it is time to load the certificate, for that please study the following material:
   1. <https://stackoverflow.com/questions/4691699/how-to-convert-crt-to-pem>
   2. <https://medium.com/@paraspatidar/add-self-signed-or-ca-root-certificate-in-kubernetes-pod-ca-root-certificate-store-cb7863cb3f87>
   3. <https://wiki.jenkins.io/display/JENKINS/Kubernetes+Continuous+Deploy+Plugin>
   4. <https://rancher.com/docs/rancher/v2.x/en/k8s-in-rancher/certificates/>
   5. <https://rancher.com/docs/rancher/v2.x/en/k8s-in-rancher/load-balancers-and-ingress/ingress/>
2. In the case of the angular app the certificate is added using the .crt file in the nginx.conf file and for API the certificate must be converted to .pfx with openssl and add this one as an environment variable in your dockerfile.
3. For more references check the following files:

* Dockerfile for Backend - <https://github.corp.globant.com/ErnstYoungX/TalentSurferBE/blob/MASTER-DEV/EY.TalentSurfer.Api/Dockerfile>
* Nginx.conf for Frontend - <https://github.corp.globant.com/ErnstYoungX/TalentSurferInternalModuleUI/blob/MASTER-DEV/config/nginx.conf>

1. <https://msandbu.wordpress.com/2012/10/15/convert-from-crt-to-pfx-with-openssl/>

The example command to generate the .pfx using the \*.corp.globant.com certificate is the following

openssl pkcs12 -export -out globant\_certificate.pfx -inkey corp.globant.com.key -in corp.globant.com.crt -certfile gd\_bundle-g2-g1.crt

Note: In order to work it is necessary to add the path C:\Program Files\OpenSSL-Win64\bin as an environment variable of the system.

1. Add CORS module to the Startup.cs and as a decorator in each Class Controller of the API

<https://stackoverflow.com/questions/44379560/how-to-enable-cors-in-asp-net-core-webapi>

<https://stackoverflow.com/questions/31942037/how-to-enable-cors-in-asp-net-core>

<https://www.c-sharpcorner.com/article/enabling-cors-in-asp-net-core-api-application/>

1. Clone the YAML FILES from <https://github.corp.globant.com/ErnstYoungX/TalentSurferKubernetesYAML>
2. **IMPORTANT NOTE: If you create the yaml files from scratch don’t forget to define the memory and cpu limits and the memory and cpu request for the pods and replicasets, the recommendation is to set between 200 and 300 mili cpu and between 25 and 50 MiB for the memory.**
3. Deploy the yamls files with kubectl create -f \*.yaml and delete with kubectl delete -f \*.yaml
4. The quickest and easiest method is install kubectl in the jenkins server and add in the jenkinsfile another new “stage” adding a argument with “--kubeconfig kube.config” downloaded from the Rancher Cluster,

<https://kubernetes.io/docs/tasks/tools/install-kubectl/>

<https://kubernetes.io/docs/tasks/access-application-cluster/configure-access-multiple-clusters/>

NOTE: as an alternative you can try Automatic deploy to kubernetes with jenkins plugins

<https://plugins.jenkins.io/kubernetes-cli>

And then Create Credentials of the type “Kind: Username y password” and passing the ID in the stage using the plugin documentation.

NOTE: If you want to validate a certificate for an app (excluding the NET CORE apps since the procedure is different) you need to open the rancher dashboard and select your cluster,

on your cluster , go to RESOURCES / SECRETS / CERTIFICATES, load the private key + cert there, and then on your project, go to LOAD BALANCING, and create an ingress there using the new cert (and a valid domain to the cert).

# Support RoadMap - Bitacora de soporte

Depending the case the best support tools are the #rancher slack channel. And for infrastructure problems you can contact Felipe Martínez or for development problems and credentials contact Martín Méndez.

## Important

First try to resolve your problems using the support channel of the company and only try to contact Felipe or Martín for more difficult problems to solve.

# Support in case of problems

## Felipe Martínez

Part of the Linux virtual server team. Knowledge of rancher administration but not in kubernetes.

## Martín Méndez

Subject matter expert of globant. Knowledge of kubernetes and complex software problems.

## Rancher Slack Channel - #Rancher (speak in english here)

Here you can ask for an Rancher cluster and ask for help in generic problems. To request a cluster you need to verify your Project Name and PM name in glow and send this information to the global channel, always asking in english.

## Ticketing tool (speak in english here)

The ticket tool is the tool where we should ask for some permissions like DNS, Domains, or web certificates for the cloud deployment of our services and apps. The specific type of support that we need to follow is the following:

My IT Support » My Virtual Server » Linux Virtual Server » General Support

In order to deploy Talent Surfer, Here we need to:

* Ask for Domain or DNS: Send our cluster IP exposed with kubectl describe service <service\_name> and ask for a custom sub domain in the \*.corp.globant.com domain. Ask for the \*.corp.globant.com certificate signed for CA. Here we need to send a ticket with our project name, PM name and time period that we will use the certificate.

NOTA: Depending on the ticket that you requested, the admins ask you for more information, and maybe you have to reply to the ticket with that information.