**CRIANDO UMA APLICAÇÃO REAL DE JUPYTER NOTEBOOK COM GERENCIAMENTO NO KUBERNETES**

**FASES DO PROJETO DE DESENVOLVIMENTO DESTA APLICAÇÃO:**

**FASE 1 - MANIFESTO NAMESPACE & ANNOTATIONS**

PASSO 1 - Execute o comando abaixo, para que criar o arquivo manifesto programado em YAML\YML.

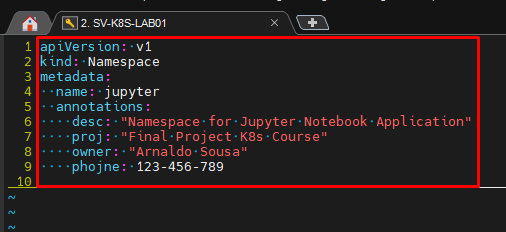
COMANDO: **sudo vim /DIGITE\_O\_DIRETORIO\_COMPLETO/DIGITE\_O\_NOME\_DO\_ARQUIVO.yml**

EXEMPLO:



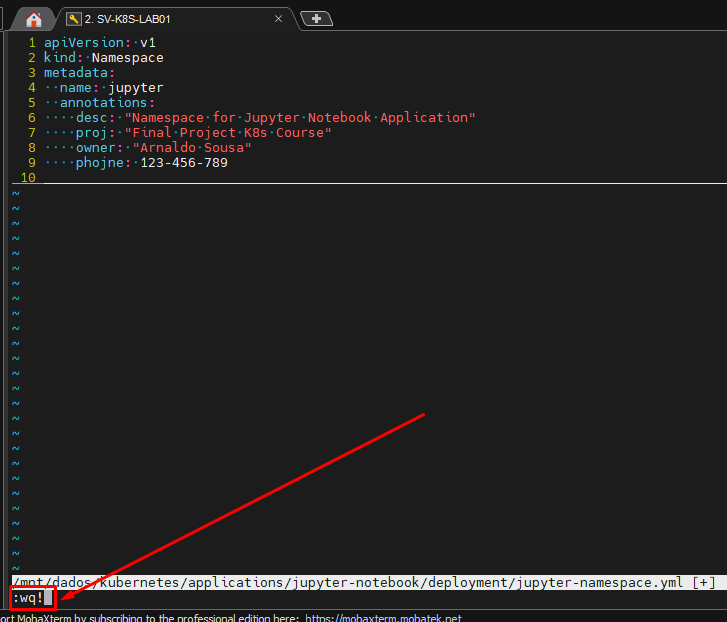
PASSO 2 - Insira o conteúdo desejado dentro do arquivo em questão, se atentando a "IDENTAÇÃO" do conteúdo a ser inserido.

EXEMPLO DE CONTEÚDO:



PASSO 3 - Salve o arquivo em questão, através da combinação de teclas SHIFT + : (DOIS PONTOS DO TECLADO) + wq + ! (Sinal de exclamação).

EXEMPLO:

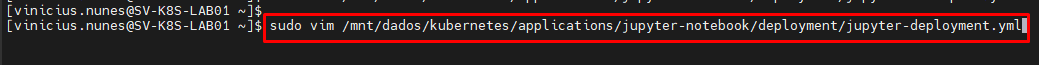


**FASE 2 - MANIFESTO DEPLOYMENT**

PASSO 1 - Execute o comando abaixo, para que criar o arquivo manifesto programado em YAML\YML para os nossos POD's.

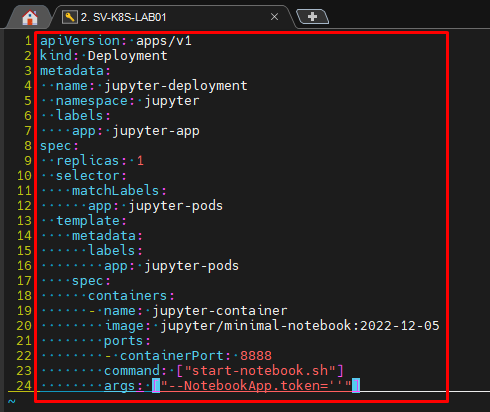
COMANDO:

EXEMPLO:



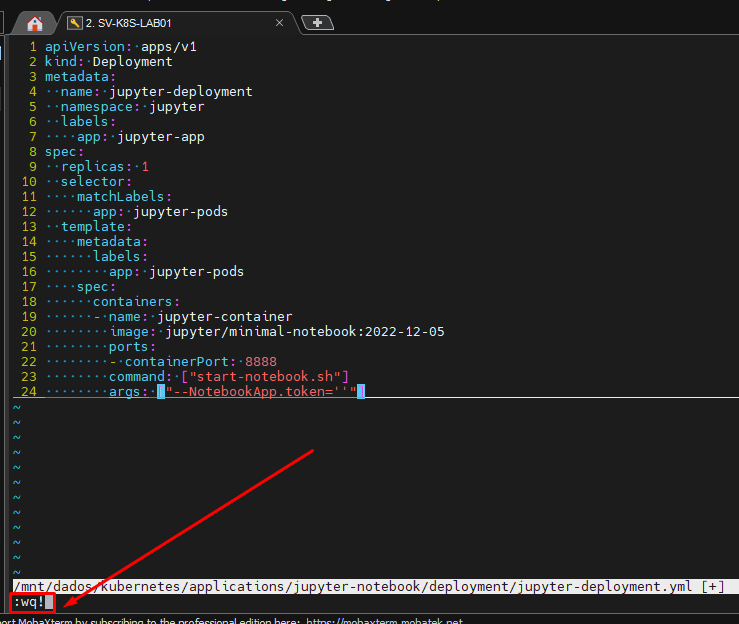
PASSO 2 - Insira o conteúdo desejado dentro do arquivo em questão, se atentando a "IDENTAÇÃO" do conteúdo a ser inserido.

EXEMPLO DE CONTEÚDO:



PASSO 3 - Salve o arquivo em questão, através da combinação de teclas SHIFT + : (DOIS PONTOS DO TECLADO) + wq + ! (Sinal de exclamação).

EXEMPLO:

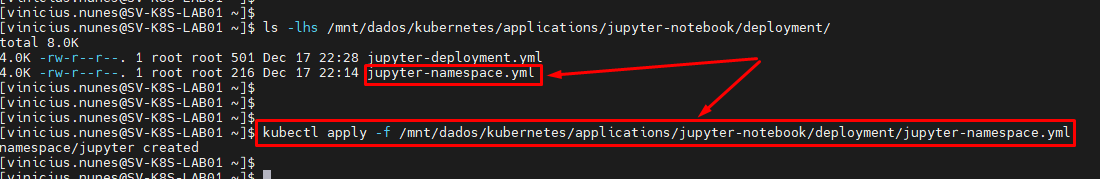


**FASE 3 - IMPLANTANDO APLICAÇÃO**

PASSO 1 - Execute o comando abaixo, para que possamos aplicar as configurações definidas no conteúdo do arquivo manifesto do NAMESPACE, conforme criado anteriormente.

COMANDO: **kubectl apply -f /DIGITE\_O\_CAMINHO\_ABSOLUTO\_DO\_DORETÓRIO/DIGITE\_O\_NOME\_DO\_ARQUIVO.yml**

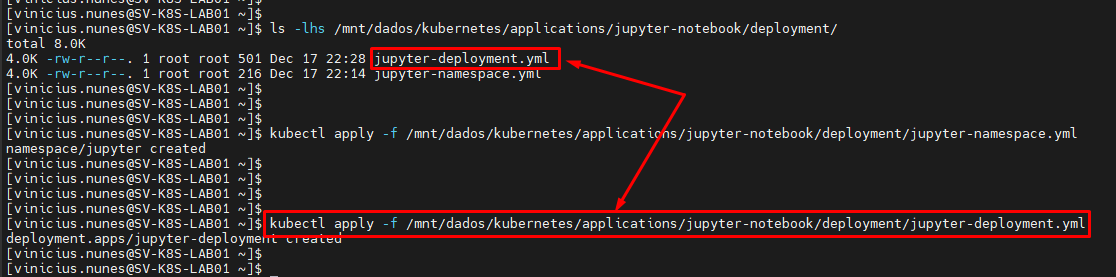
EXEMPLO:



PASSO 2 - Execute o comando abaixo, para que possamos aplicar as configurações definidas no conteúdo do arquivo manifesto do DEPLOYMENT, conforme criado anteriormente.

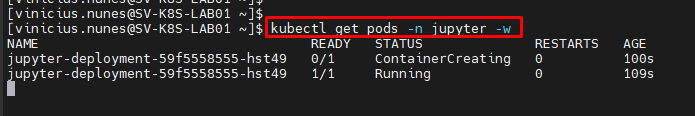
COMANDO: **kubectl apply -f /DIGITE\_O\_CAMINHO\_ABSOLUTO\_DO\_DORETÓRIO/DIGITE\_O\_NOME\_DO\_ARQUIVO.yml**

EXEMPLO:



PASSO 3 - Execute o comando abaixo, para que possamos verificar o POD sendo criado através do NAMESPACE JUPYTER, onde o nome JUPYTER é o nome do namespace que definimos durante o desenvolvimento do conteúdo do arquivco manifesto do NAMESPACE.

EXEMPLO:



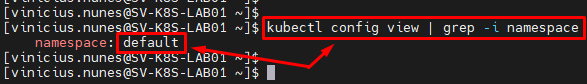
PASSO 4 - Aguarde a conclusão da criação dos novos POD's no novo namespace jupyter.

**FASE 4 - VERIFICANDO IMPLEMENTAÇÃO**

PASSO 1 - Execute o comando abaixo, para que possamos identificar quem é o namespace padrão que está sendo utilizado durante a administração do kubernetes via CLI.

COMANDO: **kubectl config view | grep -i namespace**

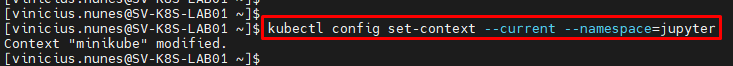
EXEMPLO:



PASSO 2 - Execute o comando abaixo, para que possamos realizar a mudança do namespace padrão, onde iremos inserir no namespace chamado JUPYTER como padrão, pois este é o novo namespace que criamos durante o desenvolvimento do arquivo manifesto do NAMESPACE da aplicação.

COMANDO: **kubectl config set-context --current --namespace=jupyter**

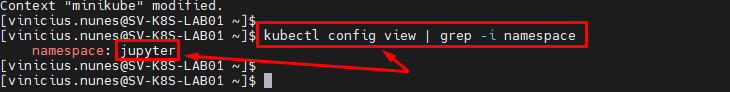
EXEMPLO:



PASSO 3 - Execute o comando abaixo, para que possamos identificar se o namespace padrão que está sendo utilizado durante a administração do kubernetes via CLI já está definido como JUPYTER.

COMANDO: **kubectl config view | grep -i namespace**

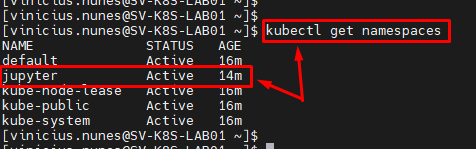
EXEMPLO:



PASSO 4 - Execute o comando abaixo, para validarmos a execução do namespace JUPYTER.

COMANDO: **kubectl get namespaces**

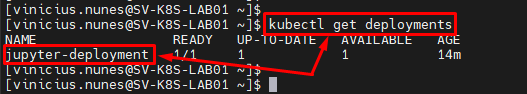
EXEMPLO:



PASSO 5 - Execute o comando abaixo, para validarmos a execução do deployment JUPYTER.

COMANDO: **kubectl get deployments**

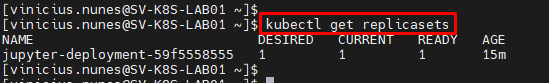
EXEMPLO:



PASSO 6 - Execute o comando abaixo, para validarmos a execução do replicaset JUPYTER.

COMANDO: **kubectl get replicasets**

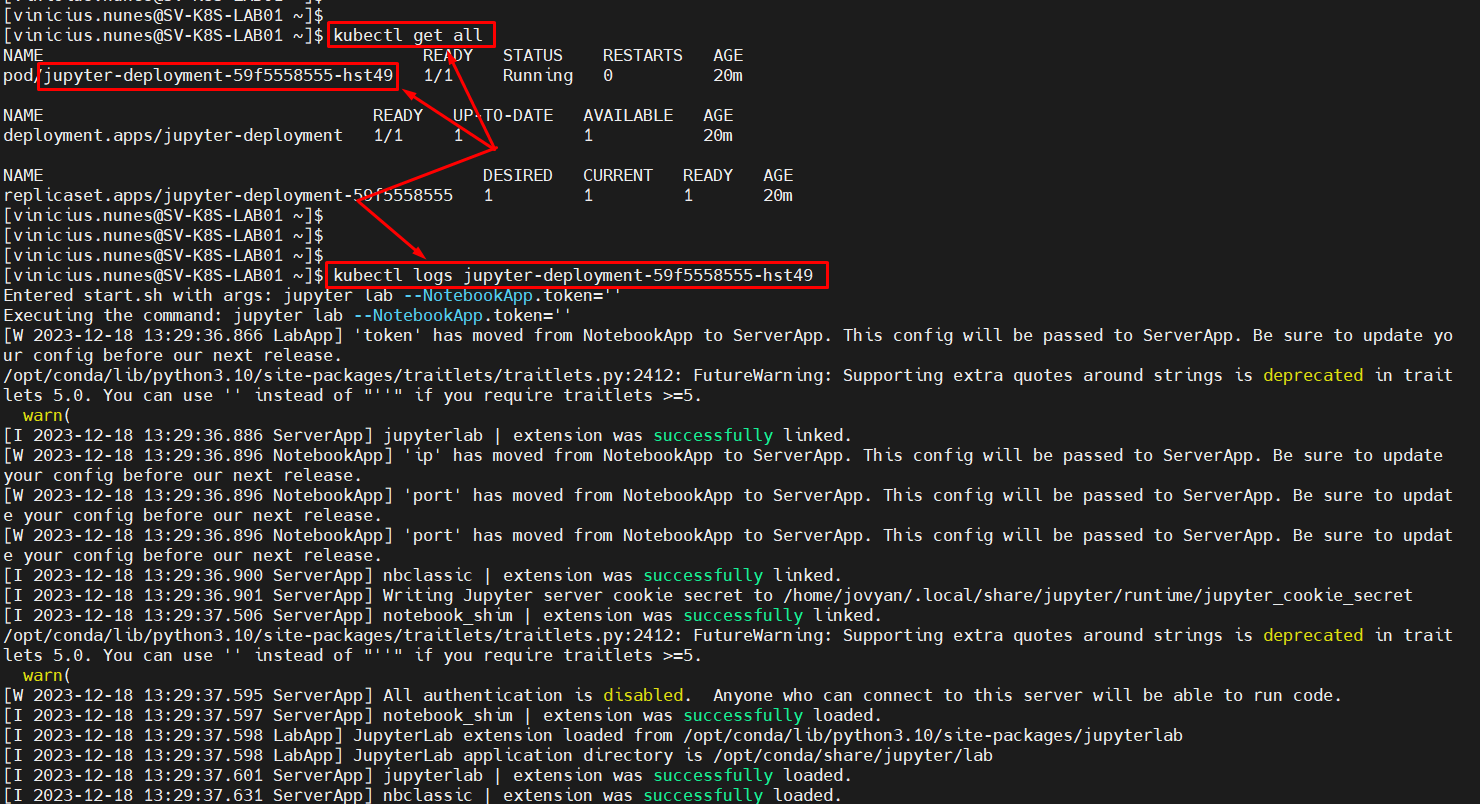
EXEMPLO:



PASSO 7 - Execute o comando abaixo, para validarmos se nas logs do POD do jupyter houve algum erro durante o processo de deployment.

COMANDO: **kubectl logs DIGITE\_O\_NOME\_COMPLETO\_DO\_POD\_DESEJADO**

EXEMPLO:

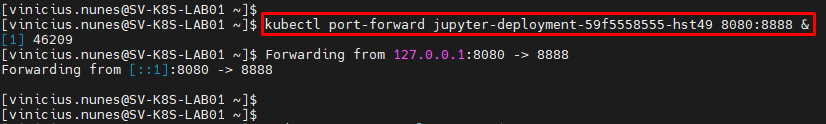


**FASE 5 - TESTANDO APLICAÇÃO**

PASSO 1 - Execute o comando abaixo, para que possamos criar um processo de PORT-FORWARD, ou seja, criar uma espécie de PAT\NAT que traduz uma porta do kubernetes host para uma porta dentro do container em um determinado POD.

COMANDO: **kubectl port-forward DIGITE\_O\_NOME\_COMPLETO\_DO\_POD\_DESEJADO DIGITE\_O\_NÚMERO\_DA\_PORTA\_DO\_KUBERNETES-HOST\_QUE\_DESEJA\_UTILIZAR:8888 &**

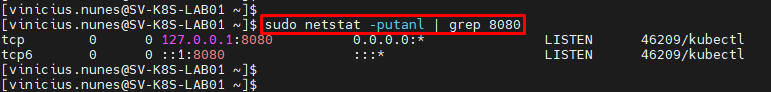
EXEMPLO:



PASSO 2 - Execute o comando abaixo, para que possamos validar se a porta e o processo que definimos no comando anteriormente já estão em execução, ou seja, se a porta já está em estado de "LISTENING".

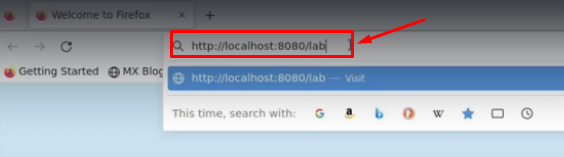
COMANDO: **sudo netstat -putanl | grep DIGITE\_O\_NÚMERO\_DA\_PORTA\_DO\_KUBERNETES-HOST\_QUE\_FOI\_UTILIZADA**

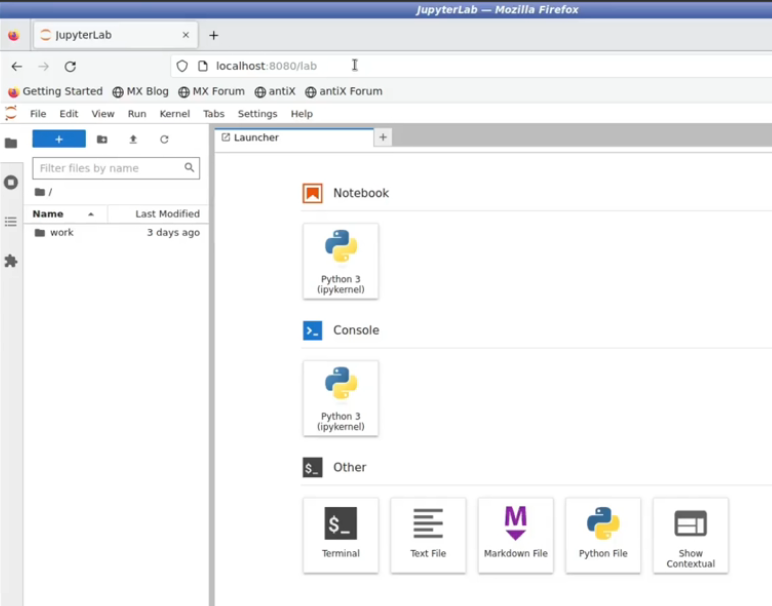
EXEMPLO:



PASSO 3 - Abra o browser e digite o endereço IP ou localhost na porta 8080/lab, para que possamos abrir a console web do Jupyter.

EXEMPLO:



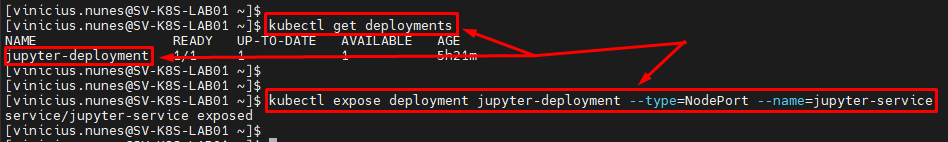


**FASE 6 - MIGRANDO APLICAÇÃO PARA SERVICE**

PASSO 1 - Execute o comando abaixo, para que possamos gerar um serviço no kubernetes, a partir das configurações de um determinado deployment.

COMANDO: **kubectl expose deployment DIGITE\_O\_NOME\_DO\_DEPLOYMENT\_DESEJADO --type=NodePort --name=jupyter-service**

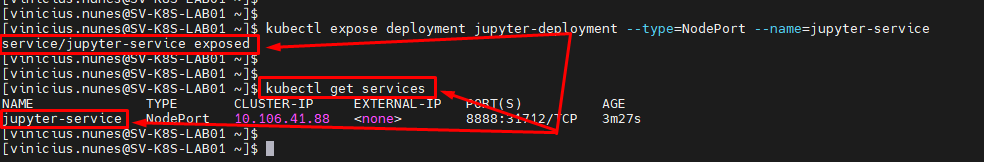
EXEMPLO:



PASSO 2 - Execute o comando abaixo, para que possamos validar se o serviço criado no passo anterior já esta em execução.

COMANDO: **kubectl get services**

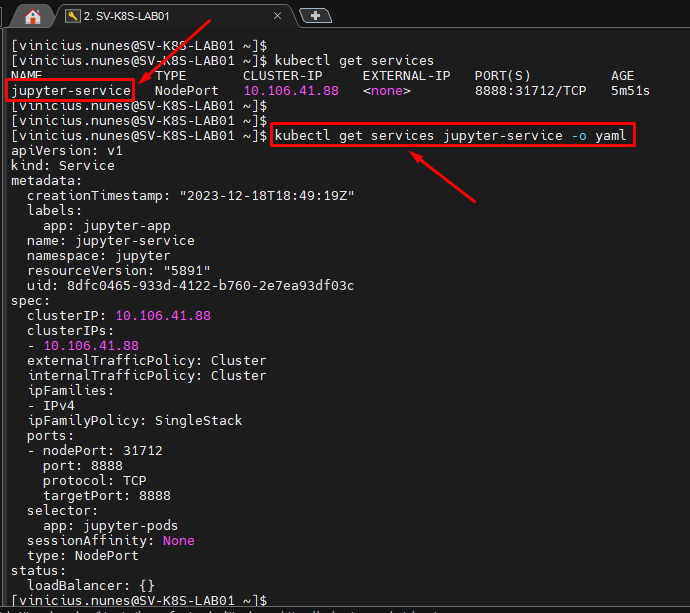
EXEMPLO:



PASSO 3 - Execute o comando abaixo, para que possamos verificar os detalhes em YAML das configurações do serviço criado anteriormente.

COMANDO: **kubectl get services jupyter-service -o yaml**

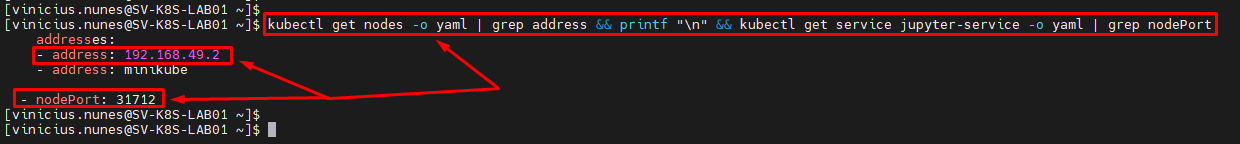
EXEMPLO:



PASSO 4 - Execute o comando abaixo, para que coletar a URL do serviço criado anteriormente, para que possamos acessar essa URL via Web Browser.

COMANDO: **kubectl get nodes -o yaml | grep address && printf "\n" && kubectl get service jupyter-service -o yaml | grep nodePort**

EXEMPLO:



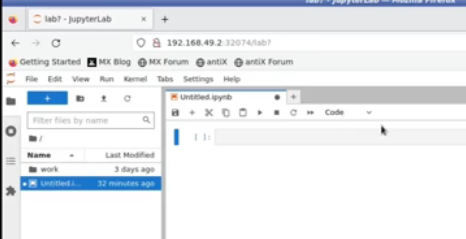
**AGORA COM A SAÍDA DESTACADA ACIMA, MONTE A URL DA SEGUINTE FORMA:** <http://DIGITE_O_ENDEREÇO_IP_DO_STDOUT_ACIMA:DIGITE_O_NÚMERO_DA_PORTA_DO_STDOUT_ACIMA>

EXEMPLO:



PASSO 5 - Abra o Web Browser (navegador) e insira a URL que montamos no passo anterior, onde deverá ser apresentado a interface web da aplicação Jupyter.

EXEMPLO:

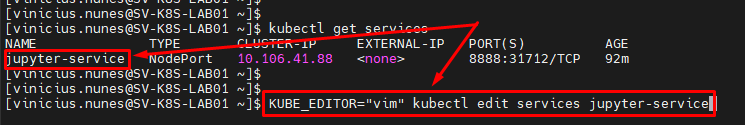


**FASE 7 - MIGRANDO PORTA DA APLICAÇÃO**

PASSO 1 - Execute o comando abaixo, para que possamos acessar o modo de edição do serviço da aplicação, para que possamos seguir com os próximos passos.

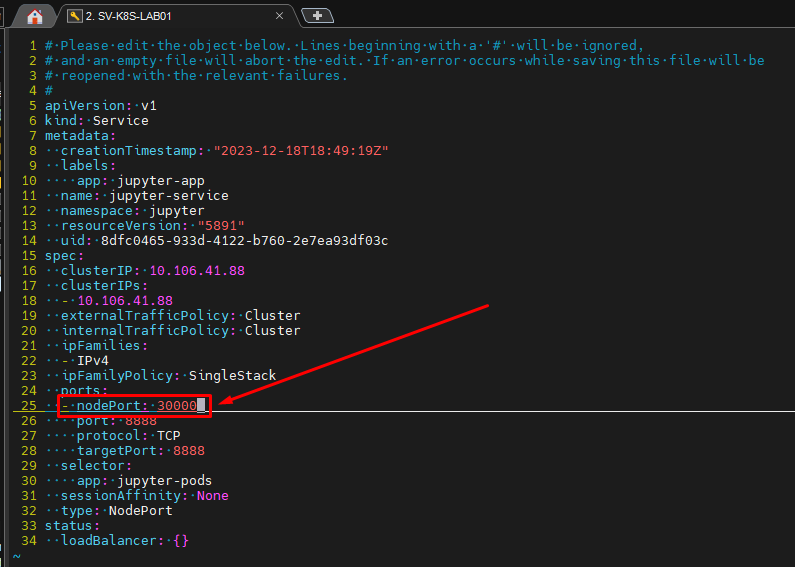
COMANDO: **KUBE\_EDITOR="vim" kubectl edit services jupyter-service**

EXEMPLO:



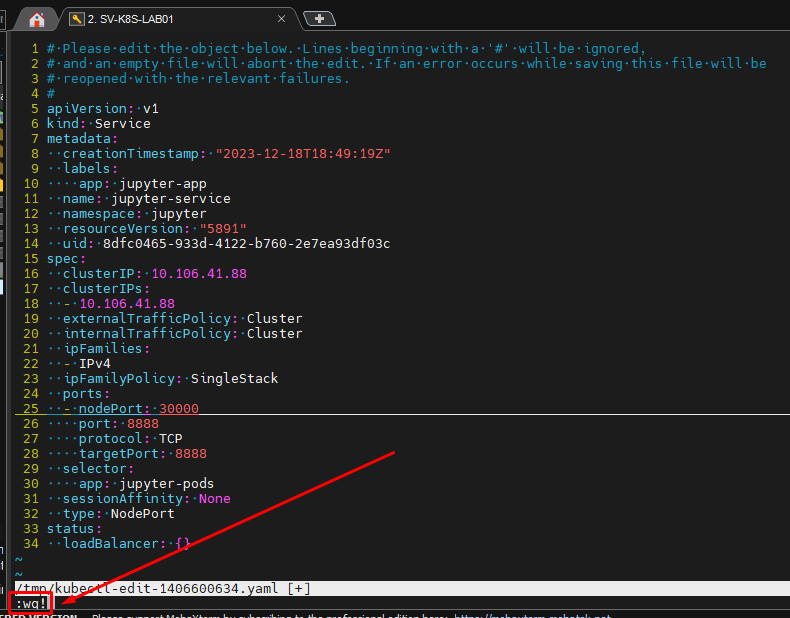
PASSO 2 - No campo destacado abaixo, substitua o valor da porta atual para o valor da porta desejada (30000).

EXEMPLO:



PASSO 3 - Salve o arquivo em questão, através da combinação de teclas SHIFT + : (DOIS PONTOS DO TECLADO) + wq + ! (Sinal de exclamação).

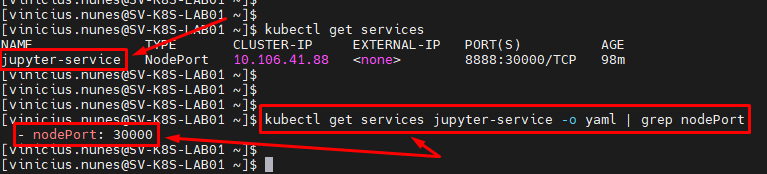
EXEMPLO:



PASSO 4 - Execute o comando abaixo, para que possamos validar se serviço do jupyter já está utilizando a porta 30000, conforme definimos anteriormente.

COMANDO: **kubectl get services jupyter-service -o yaml | grep nodePort**

EXEMPLO:



**FASE 8 - ESCALANDO APLICAÇÃO**

PASSO 1 - Execute o comando abaixo, para que possamos gerar um arquivo YAML através do stdout do conteúdo do serviço jupyter rodando dentro do kubernetes.

COMANDO: **kubectl get services jupyter-service -o yaml > /INSIRA\_O\_DIRETORIO\_ABSOLUTO\_PARA\_O\_ARQUIVO/jupyter-service.yml**

EXEMPLO:



PASSO 2 - Acesse o arquivo gerado anteriormente.

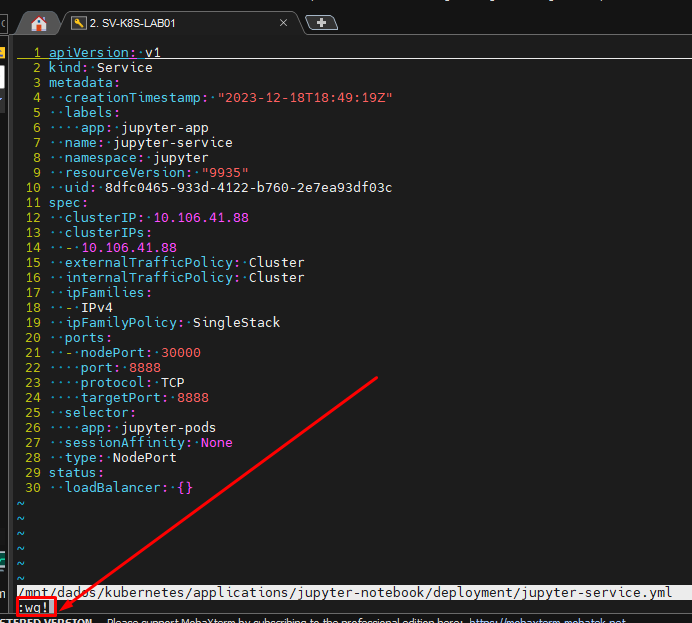
COMANDO: **sudo vim /DIGITE\_O\_CAMINHO\_ABSOLUTO\_ONDE\_ESTA\_ARMAZENADO\_O\_ARQUIVO/jupyter-service.yml**

EXEMPLO:



PASSO 3 - Salve o arquivo em questão, através da combinação de teclas SHIFT + : (DOIS PONTOS DO TECLADO) + wq + ! (Sinal de exclamação).

EXEMPLO:



PASSO 4 - Execute o comando abaixo, para que possamos acessar o arquivo de deployment da aplicação jupyter, onde este arquivo foi criado anteriormente, ainda durante os primeiros passos das FASE 1 DO PROJETO.

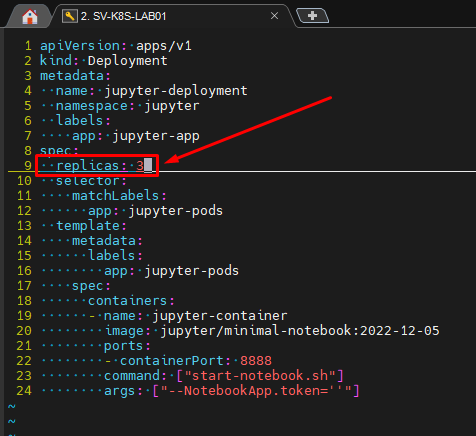
COMANDO: **sudo vim /DIGITE\_O\_CAMINHO\_ABSOLUTO\_ONDE\_ESTA\_ARMAZENADO\_O\_ARQUIVO/jupyter-deployment.yml**

EXEMPLO:



PASSO 5 - Conforme campo destacado abaixo, insira o valor 3, para que possamos criar 3 POD's replicados quando este arquivo for executado.

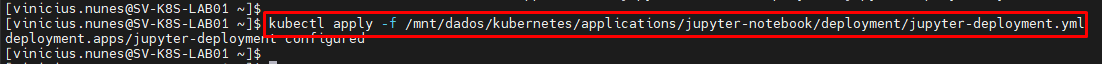
EXEMPLO DE CONTEÚDO:



PASSO 6 - Execute o comando abaixo, para que possamos executar o arquivo manifesto criado anteriormente.

COMANDO: **kubectl apply -f /DIGITE\_O\_CAMINHO\_ABSOLUTO\_DO\_DORETÓRIO/DIGITE\_O\_NOME\_DO\_ARQUIVO.yml**

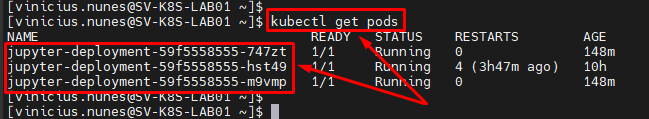
EXEMPLO:



PASSO 7 - Execute o comando abaixo, para que possamos validar se a quantidade de POD's foi alterada, devendo ser apresentado agora 3 POD's, devido ao replicaset.

COMANDO: **kubectl get pods**

EXEMPLO:



PASSO 8 - Agora basta seguir com a utilização dos recursos em seu ambiente.