**CRIANDO COMUNICAÇÃO DE CONTAINERS EM DIFERENTES PODS**

PASSO 1 - Execute o comando abaixo, para que possamos criar o arquivo YML do primeiro container de um dos PODs, sendo o container do web server TomCat.

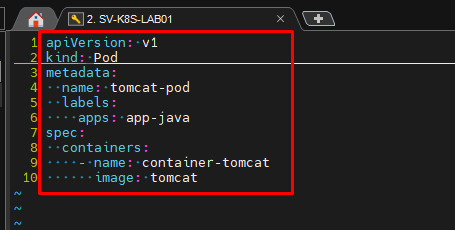
COMANDO: **sudo vim /DIGITE\_O\_DIRETORIO\_COMPLETO/DIGITE\_O\_NOME\_DO\_ARQUIVO.yml**

EXEMPLO:



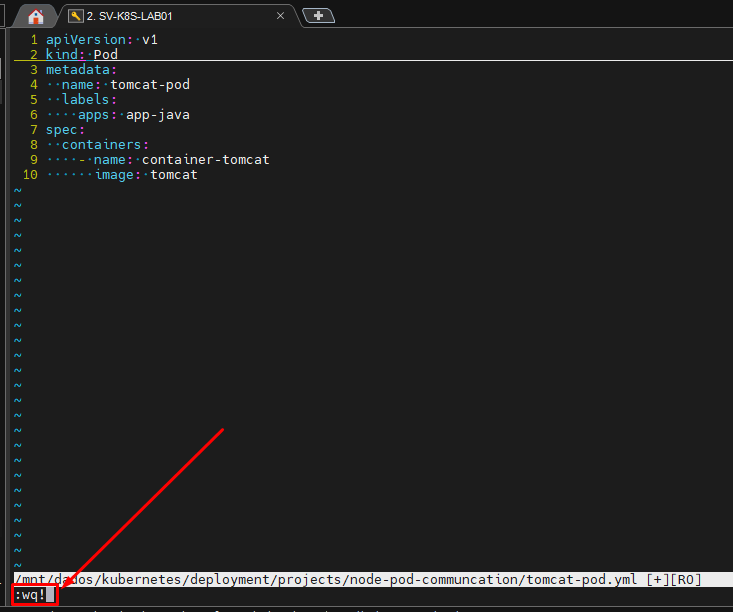
PASSO 2 - Insira o conteúdo abaixo dentro do arquivo em questão, se atentando a "IDENTAÇÃO" do conteúdo a ser inserido.

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 - Agora execute o comando abaixo, para que possamos criar o arquivo YML do segundo container de um dos PODs, sendo o container do SGBD (Banco de Dados) REDIS.

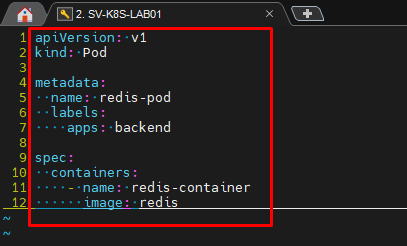
COMANDO: **sudo vim /DIGITE\_O\_DIRETORIO\_COMPLETO/DIGITE\_O\_NOME\_DO\_ARQUIVO.yml**

EXEMPLO:



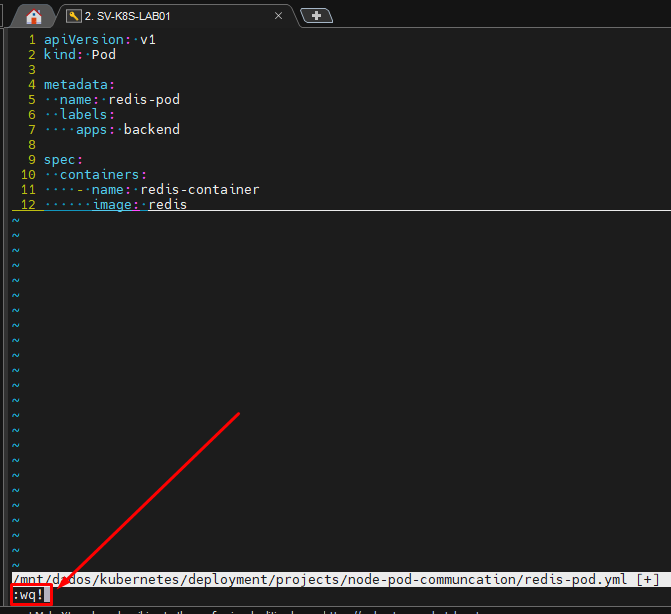
PASSO 5 - Insira o conteúdo abaixo dentro do arquivo em questão, se atentando a "IDENTAÇÃO" do conteúdo a ser inserido.

EXEMPLO:



PASSO 6 - Salve o arquivo em questão, através da combinação de teclas SHIFT + : (DOIS PONTOS DO TECLADO) + wq + ! (Sinal de exclamação).

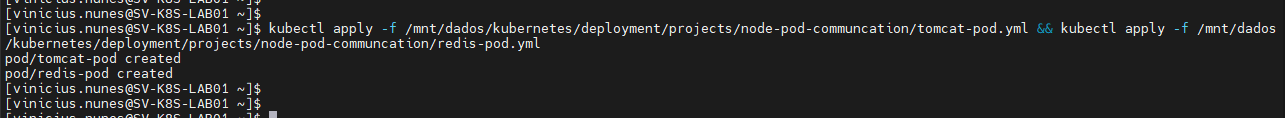
EXEMPLO:



PASSO 7 - Execute o comando abaixo, para que possamos realizar o deployment dos 2 arquivos YML criados anteriormente.

COMANDO: **kubectl apply -f /DIGITE\_O\_DIRETORIO\_COMPLETO/DIGITE\_O\_NOME\_DO\_ARQUIVO\_DO\_PRIMERO\_CONTAINER.yml && kubectl apply -f /DIGITE\_O\_DIRETORIO\_COMPLETO/DIGITE\_O\_NOME\_DO\_ARQUIVO\_DO\_SEGUNDO\_CONTAINER.yml**

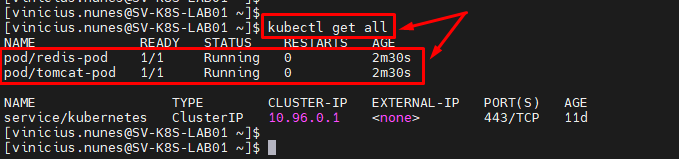
EXEMPLO:



PASSO 8 - Execute o comando abaixo, para validar se os 2 containers em PODs diferentes foram criados.

COMANDO: **kubectl get all**

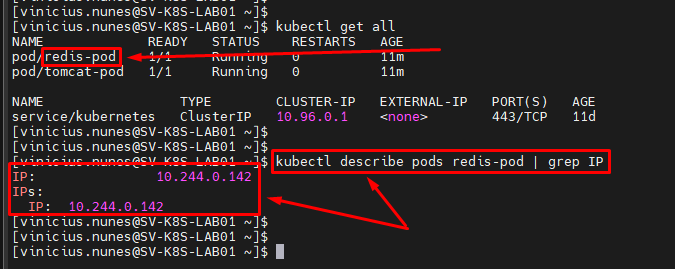
EXEMPLO:



PASSO 9 - Execute o comando abaixo, para que possamos coletar o IP Address do POD do SGBD (Banco de Dados) REDIS, e em seguida, anote o IP Address em questão.

COMANDO: **kubectl describe pods DIGITE\_O\_NOME\_DO\_POD\_DO\_SGBD | grep IP**

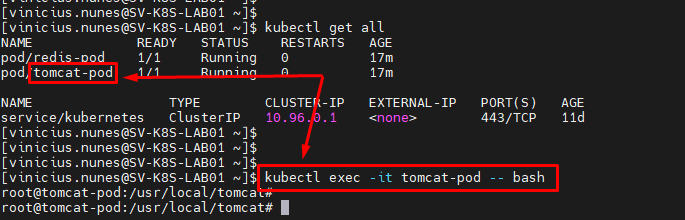
EXEMPLO:



PASSO 10 - Agora execute o comando abaixo, para que possamos acessar o shell do container do POD Tomcat, pois a partir deste, iremos realizar os devidos testes de comunicação entre containers.

COMANDO: **kubectl exec -it DIGITE\_O\_NOME\_DO\_POD\_DA\_APLICAÇÃO -- bash**

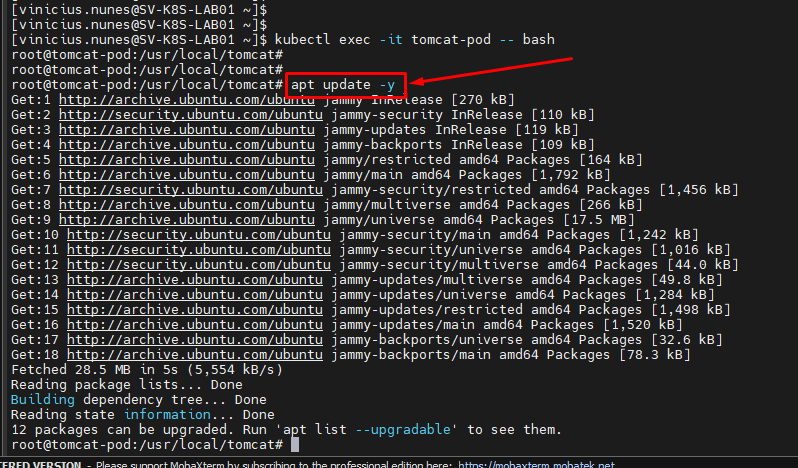
EXEMPLO:



PASSO 11 - Dentro do container do POD da aplicação TOMCAT, execute o comando abaixo, para que possamos realizar a atualização dos repositórios do containers, permitindo que posteriormente realizemos a instalação do pacote do binário ping, para realizarmos o teste de comunicação entre containers em PODs diferentes.

COMANDO: **apt update -y**

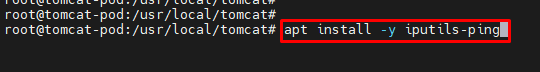
EXMEPLO:



PASSO 12 - Dentro do container do POD da aplicação TOMCAT, execute o comando abaixo, para que possamos realizar a instalação do pacote "iputils-ping" no container, permitindo que realizemos a instalação do pacote do binário ping, para realizarmos o teste de comunicação entre containers em PODs diferentes.

COMANDO: **apt install -y iputils-ping**

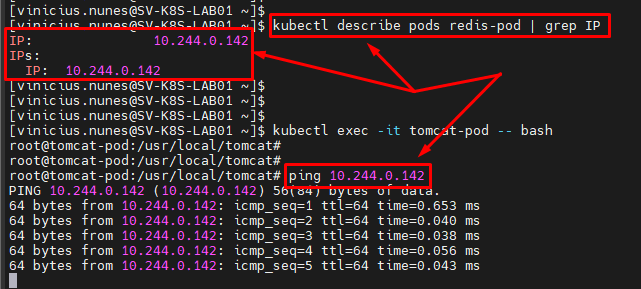
EXEMPLO:



PASSO 13 - Dentro do container do POD da aplicação TOMCAT, execute o comando abaixo, para que possamos realizar o teste de comunicação ICMP entre o container executado no POD da aplicação (Tomcat) com o container executado no POD do SGBD (Redis).

COMANDO: **ping DIGITE\_O\_IP\_ADDRESS\_DO\_POD\_SGBD (Redis)**

EXEMPLO:



PASSO 14 - Pronto, agora basta seguir com a utilização do recurso em questão em seu ambiente.