

Criando Pod via CLI

```
sudo oc run --generator=run-pod/v1 nginx --image=nginx
```

Criando um secret

```
oc create secret generic secret-cm --from-literal=usuario=usu1 --from-literal=password=secret
```

Criando ConfigMap

```
oc create configmap cf1 --from-literal=usuario=usu1 --from-literal=password=secret
```

Criando app blog via linha de comando cli, de uma imagem - do repositório dockerhub;

```
sudo oc new-app apasoft/blog

Creating resources

imagestream.image.openshift.io "blog" created
deploymentconfig.apps.openshift.io "blog" created
service "blog" created;

sudo oc get is
```

Obs: descrição acima do processo de build da imagem existente;

```
comandos

sudo oc describe is blog - verificando a image;
sudo oc get dc - podemos ver o deploymentconfig;
sudo oc describe dc blog - Detalhe da imagem;
sudo oc get rc -o wide - Informação do deployment
sudo oc get svc - verificando os serviços
sudo oc get all -l app=blog Temos as informações das ações criadas
sudo oc expose svc blog - expomos os serviços criando uma rota
```

Comandos de verificações de pod

Criando um pod com imagem do nginx:

```
sudo oc run --generator=run-pod/v1 nginx --image=nginx
```

vai baixar image do repositório docker existente;

Creating resources

```
sudo oc get pods -o wide - informações do pod
sudo oc describe pod nginx - detalhamento do pod
sudo oc get pod nginx -o yaml - descrição no formato yaml
sudo oc rsh nginx - acessando o pod dentro dele.
```

Obs: O projeto poderá ser deletado através do seguinte comando:

```
sudo oc delete all -l app=deployment-example
```

Secret

CRIANDO SECRET EM ARQUIVO DE SENHA;

```
echo -n 'admin' > username.txt --> criando arquivo com usuário e senha:
```

```
echo -n 'ajba' > password.txt --> Criando arquivo de senha
```

```
cat username.txt
```

```
cat password.txt
```

```
sudo oc create secret generic db-user-pass --from-file=username.txt --from-file=password.txt
```

```
sudo oc describe secret db-user-pass --> modo verificação txt
```

```
sudo oc get secret db-user-pass -o yaml
```

SECRET EM MODO DECLARATIVO;

```
echo -n 'admin' | base64 --> criando criptografia para usuário
```

```
echo -n 'password' | base64 --> Criando criptografia para senha
```

nano secret.yaml : podemos ver especificações criadas no manifesto:

```
sudo oc apply -f secret.yaml
```

DEFININDO VARIALVEL DE AMBIENTE Criamos uma aplicação code.js env_code.js

export=LANGUAGE=English variável no linux printenv - visualizamos Criaremos um Dockerfile para exportar as variáveis de ambiente para este Dockerfile:

```
sudo docker build -t ajorgest/env_code:v2 . -> Usando imagem no repositório docker para
```