* Launch Ubuntu Server, used t3.xlarge
* Install minikube and docker, just to simply place all below commands in shell script and run the file.

| #Install VirtualBox:  sudo apt update -y  sudo apt-get install docker.io -y  systemctl start docker  sudo usermod -aG docker $(whoami)  sudo apt install virtualbox -y  sudo apt-get install conntrack -y  #Install kubectl:  sudo apt-get update && sudo apt-get install -y apt-transport-https gnupg2  curl -s https://packages.cloud.google.com/apt/doc/apt-key.gpg | sudo apt-key add -  echo "deb https://apt.kubernetes.io/ kubernetes-xenial main" | sudo tee /etc/apt/sources.list.d/kubernetes.list  sudo apt-get update  sudo apt-get install -y kubectl  #Install Minikube:  curl -Lo minikube https://storage.googleapis.com/minikube/releases/latest/minikube-linux-amd64 && chmod +x minikube && sudo mv minikube /usr/local/bin/  minikube start --kubernetes-version=v1.21.0 --driver=none  mkdir -p /opt/jenkins\_home  chown -R ubuntu:ubuntu /opt/jenkins\_home  minikube addons enable ingress |
| --- |

* Cross verify minikube status and pods, by running kubectl get pods -A
* Now install chectl toll
  + Run the following command in the terminal (this downloads and executes the install.sh script):

bash <(curl -sL https://www.eclipse.org/che/chectl/)

* + Run the following command to verify that the version of chectl that is the expected one.
  + chectl --version
* Install Cert Manager
  + kubectl apply -f https://github.com/cert-manager/cert-manager/releases/download/v1.11.0/cert-manager.yaml
* Install Keycloak

| kubectl apply -f - <<EOF  ---  apiVersion: v1  kind: Namespace  metadata:  name: keycloak  ---  apiVersion: cert-manager.io/v1  kind: Issuer  metadata:  name: keycloak-selfsigned  namespace: keycloak  labels:  app: keycloak  spec:  selfSigned: {}  ---  apiVersion: cert-manager.io/v1  kind: Certificate  metadata:  name: keycloak-selfsigned  namespace: keycloak  labels:  app: keycloak  spec:  isCA: true  commonName: keycloak-selfsigned-ca  privateKey:  algorithm: ECDSA  size: 256  issuerRef:  name: keycloak-selfsigned  kind: Issuer  group: cert-manager.io  secretName: ca.crt  ---  apiVersion: cert-manager.io/v1  kind: Issuer  metadata:  name: keycloak  namespace: keycloak  labels:  app: keycloak  spec:  ca:  secretName: ca.crt  ---  apiVersion: cert-manager.io/v1  kind: Certificate  metadata:  name: keycloak  namespace: keycloak  labels:  app: keycloak  spec:  isCA: false  commonName: keycloak  dnsNames:  - keycloak.$(minikube ip).nip.io  privateKey:  algorithm: RSA  encoding: PKCS1  size: 4096  issuerRef:  kind: Issuer  name: keycloak  group: cert-manager.io  secretName: keycloak.tls  subject:  organizations:  - Local Eclipse Che  usages:  - server auth  - digital signature  - key encipherment  - key agreement  - data encipherment  ---  apiVersion: v1  kind: Service  metadata:  name: keycloak  namespace: keycloak  labels:  app: keycloak  spec:  ports:  - name: http  port: 8080  targetPort: 8080  selector:  app: keycloak  type: LoadBalancer  ---  apiVersion: apps/v1  kind: Deployment  metadata:  name: keycloak  namespace: keycloak  labels:  app: keycloak  spec:  replicas: 1  selector:  matchLabels:  app: keycloak  template:  metadata:  labels:  app: keycloak  spec:  containers:  - name: keycloak  image: quay.io/keycloak/keycloak:18.0.2  args: ["start-dev"]  env:  - name: KEYCLOAK\_ADMIN  value: "admin"  - name: KEYCLOAK\_ADMIN\_PASSWORD  value: "admin"  - name: KC\_PROXY  value: "edge"  ports:  - name: http  containerPort: 8080  readinessProbe:  httpGet:  path: /realms/master  port: 8080  ---  apiVersion: networking.k8s.io/v1  kind: Ingress  metadata:  name: keycloak  namespace: keycloak  annotations:  kubernetes.io/ingress.class: nginx  nginx.ingress.kubernetes.io/proxy-connect-timeout: '3600'  nginx.ingress.kubernetes.io/proxy-read-timeout: '3600'  nginx.ingress.kubernetes.io/ssl-redirect: 'true'  spec:  tls:  - hosts:  - keycloak.$(minikube ip).nip.io  secretName: keycloak.tls  rules:  - host: keycloak.$(minikube ip).nip.io  http:  paths:  - path: /  pathType: Prefix  backend:  service:  name: keycloak  port:  number: 8080  EOF |
| --- |

* + Verify keycloak pod is up or not. Once its up get the certificate
    - kubectl get secret ca.crt -o "jsonpath={.data['ca\.crt']}" -n keycloak | base64 -d > keycloak-ca.crt
    - mkdir -p /etc/ca-certificates
    - cp keycloak-ca.crt /etc/ca-certificates/keycloak-ca.crt
    - Configure Minikube to use Keycloak as the OIDC provider:

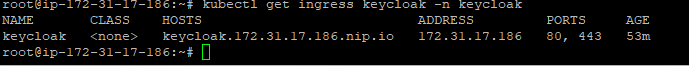
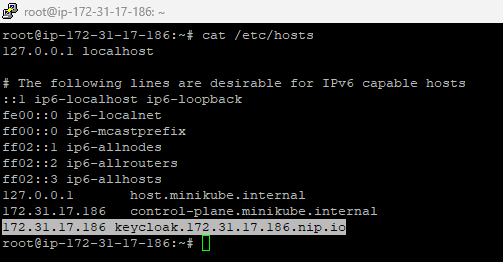
| minikube start \  --extra-config=apiserver.oidc-issuer-url=https://keycloak.$(minikube ip).nip.io/realms/che \  --extra-config=apiserver.oidc-username-claim=email \  --extra-config=apiserver.oidc-client-id=k8s-client \  --extra-config=apiserver.oidc-ca-file=/etc/ca-certificates/keycloak-ca.crt \  --kubernetes-version=v1.21.0 |
| --- |

* Once again verify keycloak pod is ready or not.
  + kubectl wait --for=condition=ready pod -l app=keycloak -n keycloak --timeout=120s
* Configure Keycloak to create the realm, client, and user:

| kubectl exec deploy/keycloak -n keycloak -- bash -c \  "/opt/keycloak/bin/kcadm.sh config credentials \  --server http://localhost:8080 \  --realm master \  --user admin \  --password admin && \  /opt/keycloak/bin/kcadm.sh create realms \  -s realm='che' \  -s displayName='che' \  -s enabled=true \  -s registrationAllowed=false \  -s resetPasswordAllowed=true && \  /opt/keycloak/bin/kcadm.sh create clients \  -r 'che' \  -s clientId=k8s-client \  -s id=k8s-client \  -s redirectUris='[\"\*\"]' \  -s directAccessGrantsEnabled=true \  -s secret=eclipse-che && \  /opt/keycloak/bin/kcadm.sh create users \  -r 'che' \  -s username=test \  -s email=\"test@test.com\" \  -s enabled=true \  -s emailVerified=true && \  /opt/keycloak/bin/kcadm.sh set-password \  -r 'che' \  --username test \  --new-password test" |
| --- |

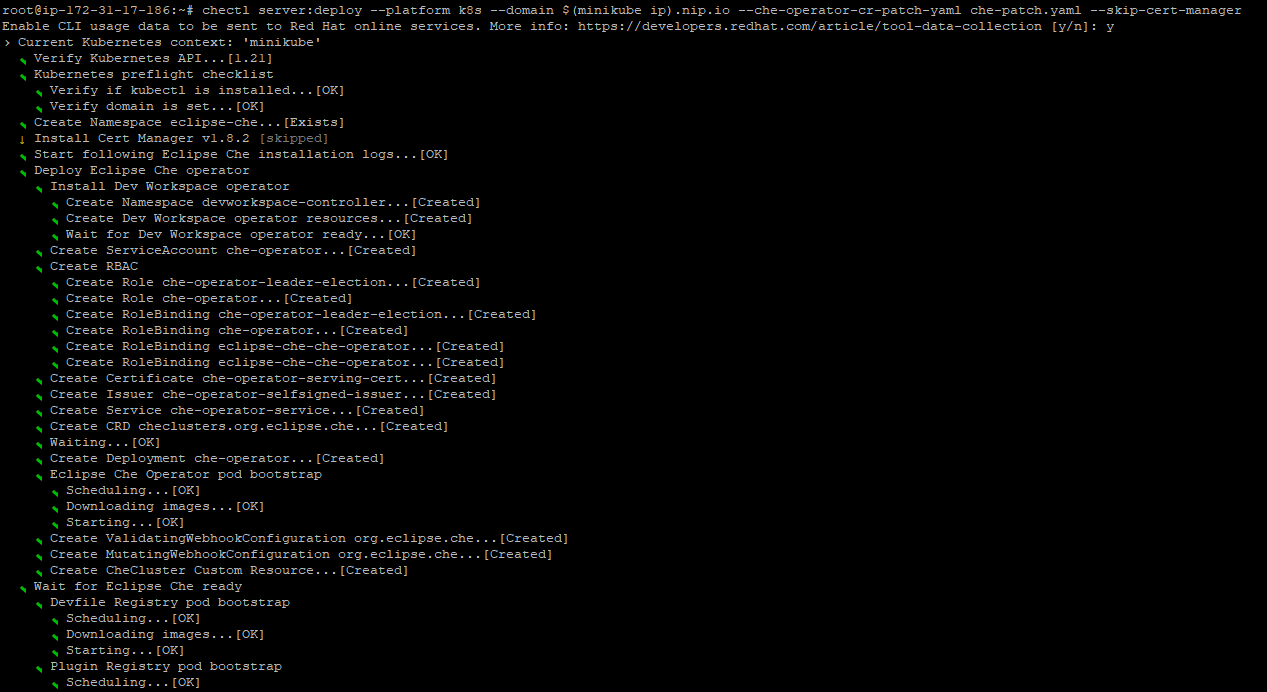
* Copy Keycloak CA certificate into the eclipse-che namespace:

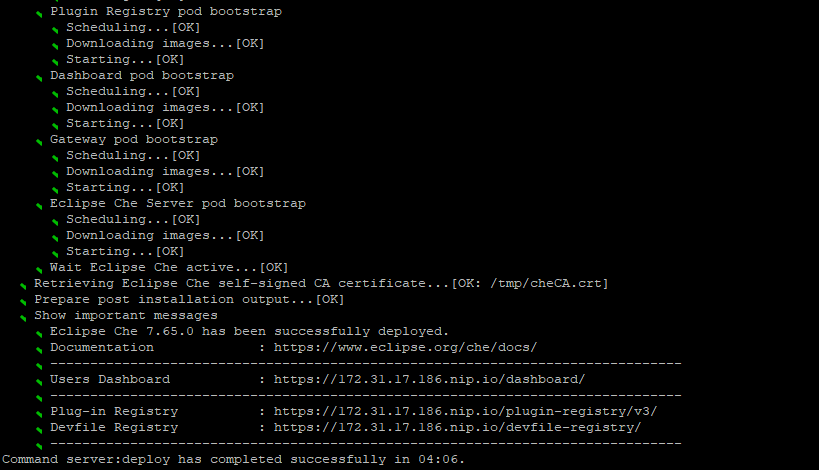
| kubectl create namespace eclipse-che && \  kubectl create configmap keycloak-certs \  --from-file=keycloak-ca.crt=keycloak-ca.crt \  -n eclipse-che && \  kubectl label configmap keycloak-certs \  app.kubernetes.io/part-of=che.eclipse.org \  app.kubernetes.io/component=ca-bundle \  -n eclipse-che |
| --- |

* Add Keycloak host entry in server
  + 
  + Add ingress enty in /etc/hosts
    - 
* Prepare the CheCluster
  + Run below command to create file and create chectl server

| cat > che-patch.yaml << EOF  kind: CheCluster  apiVersion: org.eclipse.che/v2  spec:  networking:  auth:  oAuthClientName: k8s-client  oAuthSecret: eclipse-che  identityProviderURL: "https://keycloak.$(minikube ip).nip.io/realms/che"  components:  cheServer:  extraProperties:  CHE\_OIDC\_USERNAME\_\_CLAIM: email  EOF |
| --- |

* + chectl server:deploy --platform k8s --domain $(minikube ip).nip.io --che-operator-cr-patch-yaml che-patch.yaml --skip-cert-manager

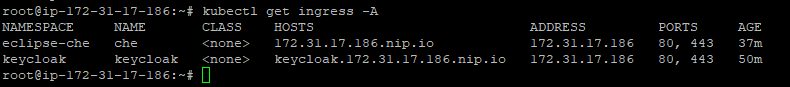




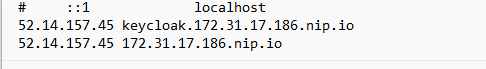
Now you can verify eclipse che pods



Get ingress and add in your local host file

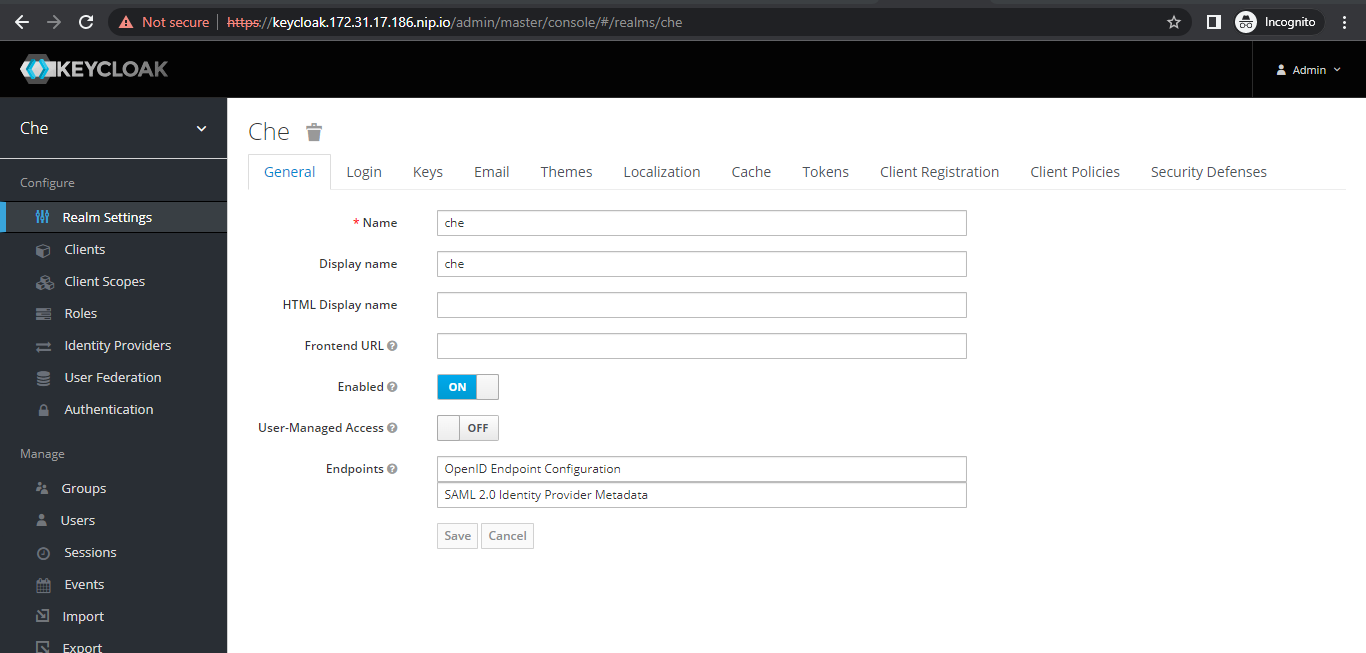


Localhost file in laptop

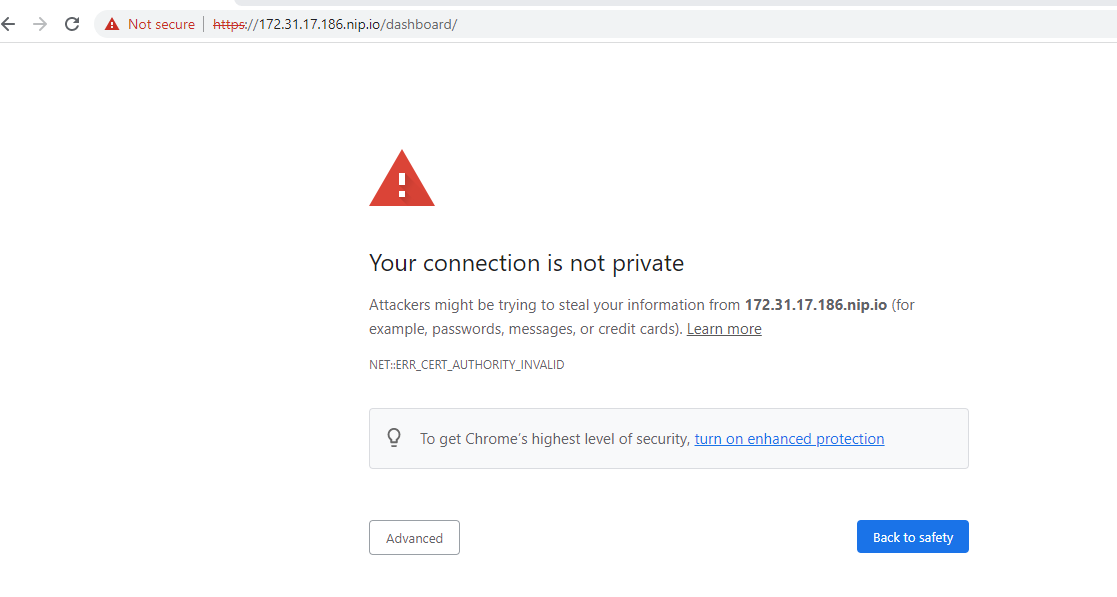


Now you can login Keycloak and eclipse-che from your local laptop with ingress url

Keycloak



Eclipse che



Now its redirect to keycloak, login with username **test** and password **test**

