**Steps to Setup NiFi on CDT cluster namespace**

**NiFi documentation URL:** <https://pages.github.ibm.com/wffh/nifi-umbrella/getting-started/>

**Setup of NiFi consists of two sections:**

1. Prerequisites
2. Installation Steps

**Prerequisites comprises of below steps/actions**

* Kubectl [installed](https://kubernetes.io/docs/tasks/tools/install-kubectl/) and configured
* Helm [installed](https://helm.sh/docs/using_helm/#installing-helm)
* Artifactory API Key for the [TaaS instance](https://na.artifactory.swg-devops.com/artifactory)
* Ingress host configured for your cluster, complete with TLS certificate; If deploying to WH CDT see [Ops Playbook instructions](https://pages.github.ibm.com/wh-ops/wh-ops-playbook/csmo/cloudprereqs/certmanager/)
* A firewall rule to allow traffic to BluePages reverse proxy for your [environment](https://pages.github.ibm.com/wffh/nifi-umbrella/bluepages/#reverse-proxy)
* If using block storage for persistent volumes (recommended), [install the plugin](https://cloud.ibm.com/docs/containers?topic=containers-block_storage#install_block) on your cluster

**Installation Steps comprises of below steps/actions**

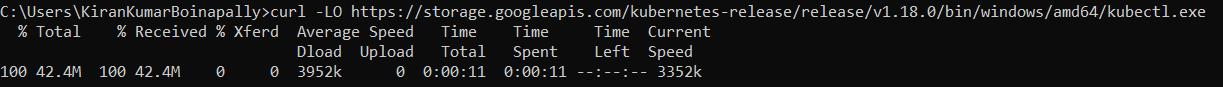
* Deploy Kubernetes Secrets
* Create a Helm override file to customize your deployment
* Add the WFFH Orchestration chart to your helm install
* Install or Upgrade the Helm Chart
* Access the UI

**Prerequisites**

1. **Install and configure kubectl on Windows**

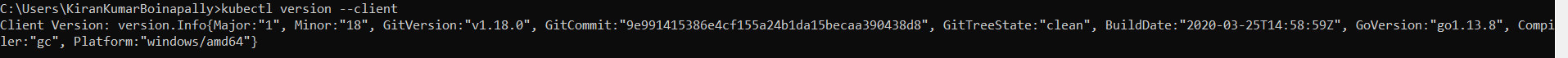
* **Install kubectl binary with curl on Windows**

curl -LO https://storage.googleapis.com/kubernetesrelease/release/v1.18.0/bin/windows/amd64/kubectl.exe



* **Test to ensure the version of kubectl is the same as downloaded:**

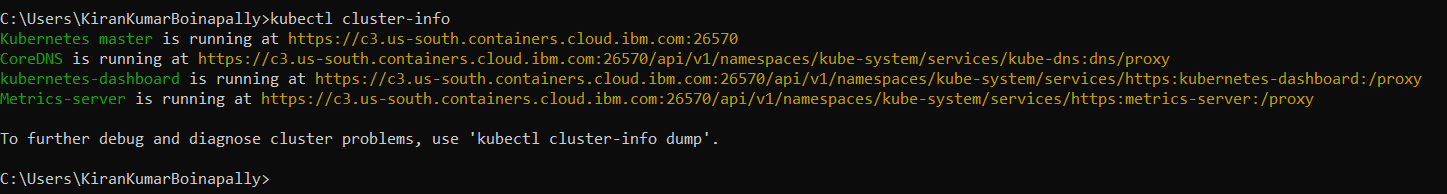
version –client



* **Verifying kubectl configuration**

Check that kubectl is properly configured by getting the cluster state:

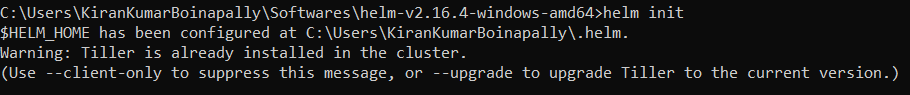
kubectl cluster-info



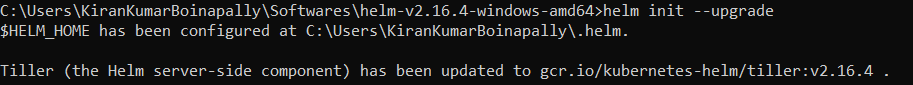
1. **Install and configure Helm on Windows**

Before installing Helm, you must have **Kubernetes** installed.

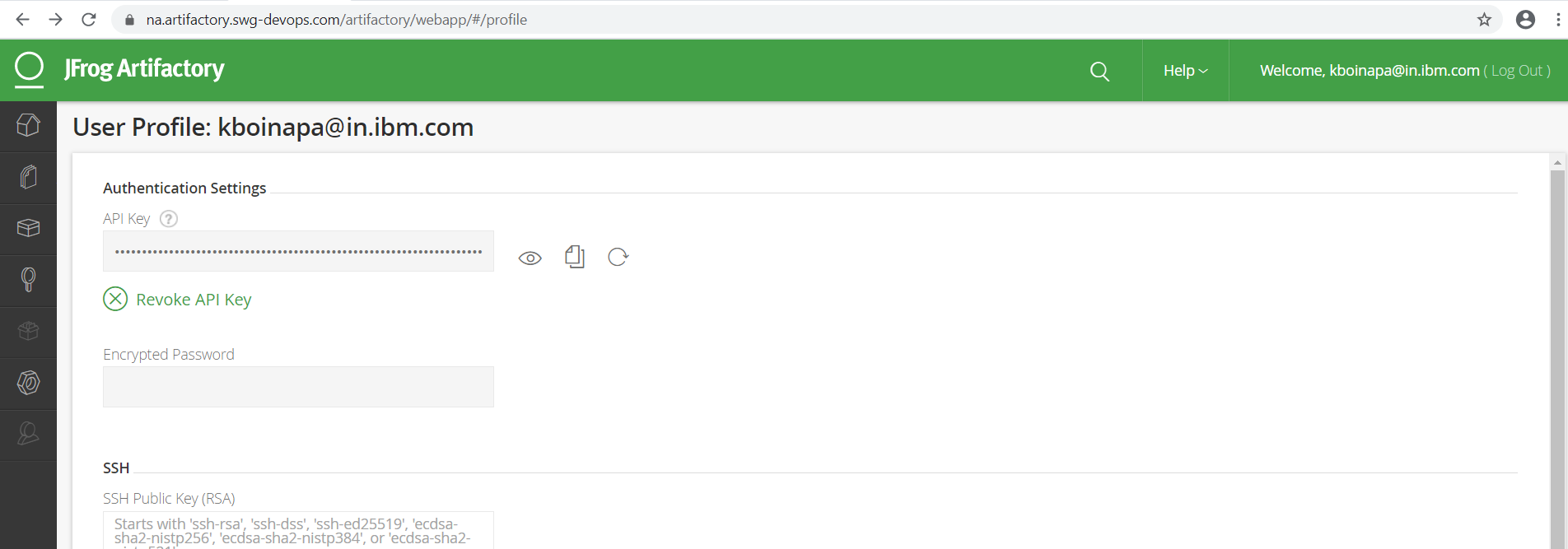
1. Go to <https://github.com/helm/helm/releases> and download **Helm v2.16.4** for Windows.
2. Unzip the **helm-v2.16.4-windows-amd64** zip folder. Helm unzipped folder contains **helm** and **tiller** applications.
3. From Command window change the directory to the unzipped folder (**helm-v2.16.4-windows-amd64**)
4. Run the ‘**helm init**’ command from the CLI command window to install Helm.



1. Run the ‘**helm init --upgrade**’ to update the **tiller** with the latest version.

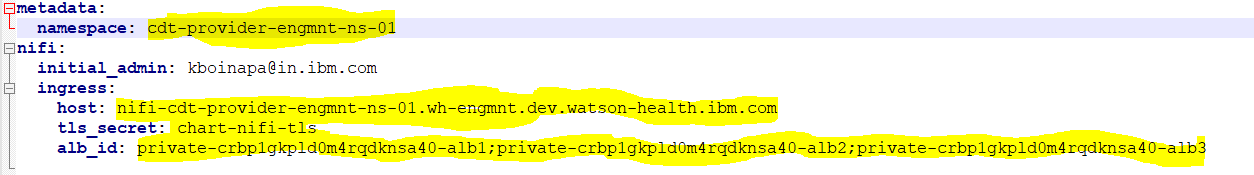


1. **Create the Artifactory API Key using JFrog’s TaaS Instance**
2. Log in to JFrog’s Artifactory’s URL (<https://na.artifactory.swg-devops.com/artifactory/webapp/#/home>) using IBM Blue Pages Id (IBM mail id)
3. By clicking on profile, we can see API Key under Authentication Setting
4. Save or note this Artifactory API Key for future reference



1. **Ingress Setup**

To setup ingress for our CDT cluster namespace, we need to update details of **host**, **tls\_secret**, **alb\_id** and **namespace** in the **override.yaml** file.



Below is the summary of the process in precise/short. Please refer <https://pages.github.ibm.com/wffh/nifi-umbrella/ingress-setup/> for more details on below steps.

**Summary of the Process**

There are several steps to this process:

1. ***Install the TLS certificate as a K8s Secret***
2. ***Request CIS Routes for your cluster***
3. ***Fill out your override file (ALBs, CIS Route, Secret Name)***

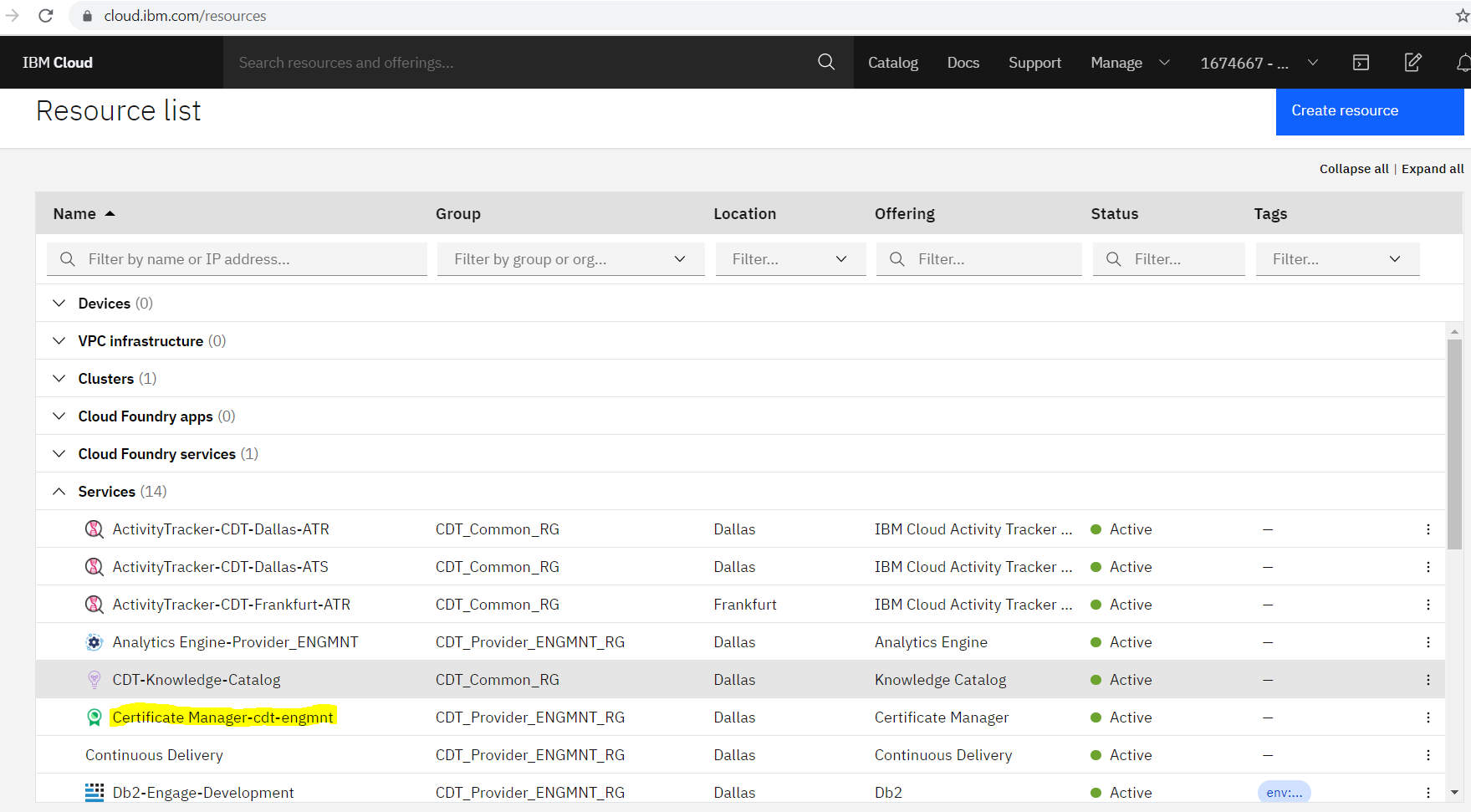
There are a few parts but its pretty straightforward:

* The TLS certificate gives you a subdomain such as .lit.wh-landing.dev.watson-health.ibm.com
* To get something like nifi-dev.lit.wh-landing.dev.watson-health.ibm.com you need to add a CIS route for nifi-dev
* You need to add the details to your override file so the helm chart can create the proper ingress file

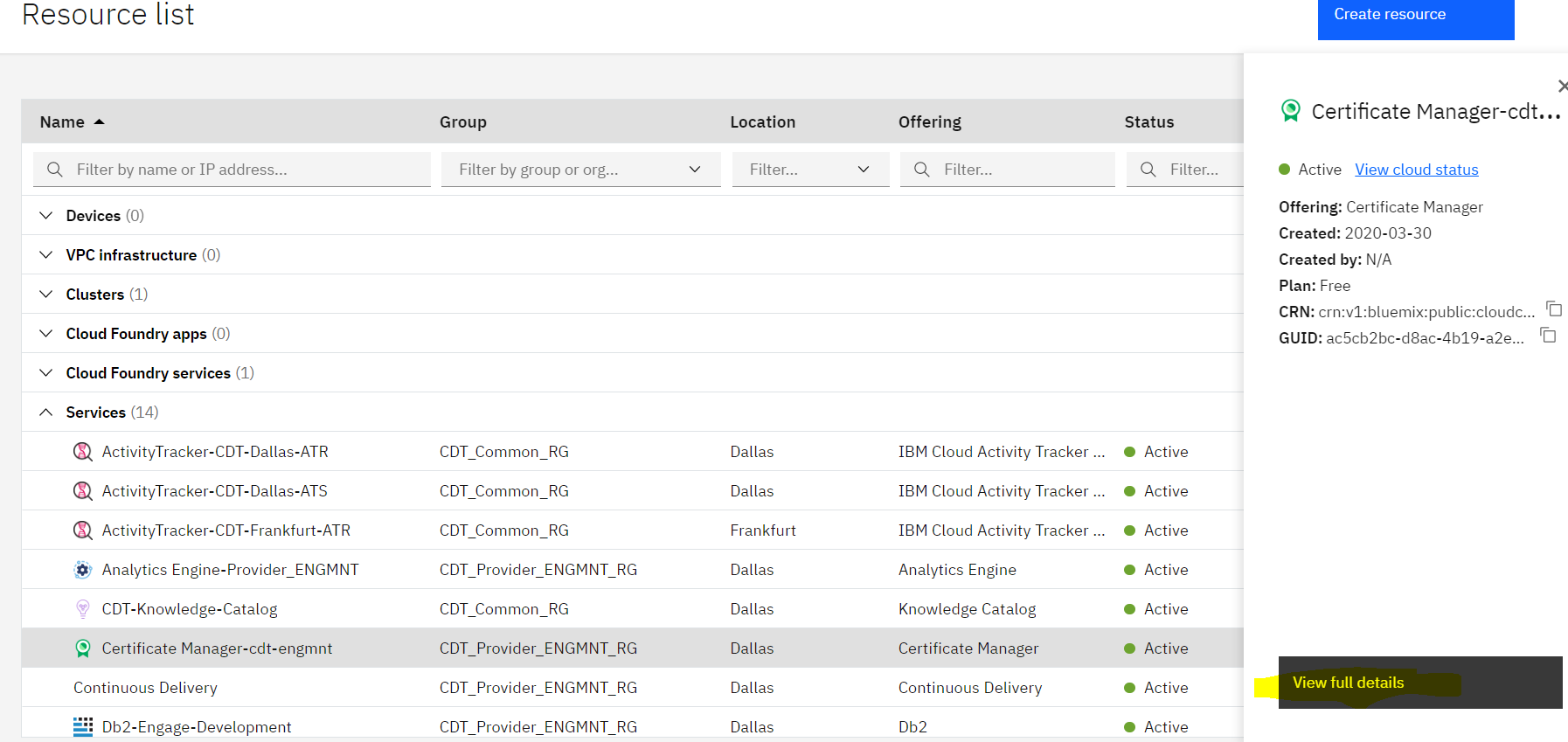
In this example we are setting up nifi-dev.lit.wh-landing.dev.watson-health.ibm.com.

1. **Install the TLS certificate as a K8s Secret:**

* Go to **Certificate Manager** under **Services**

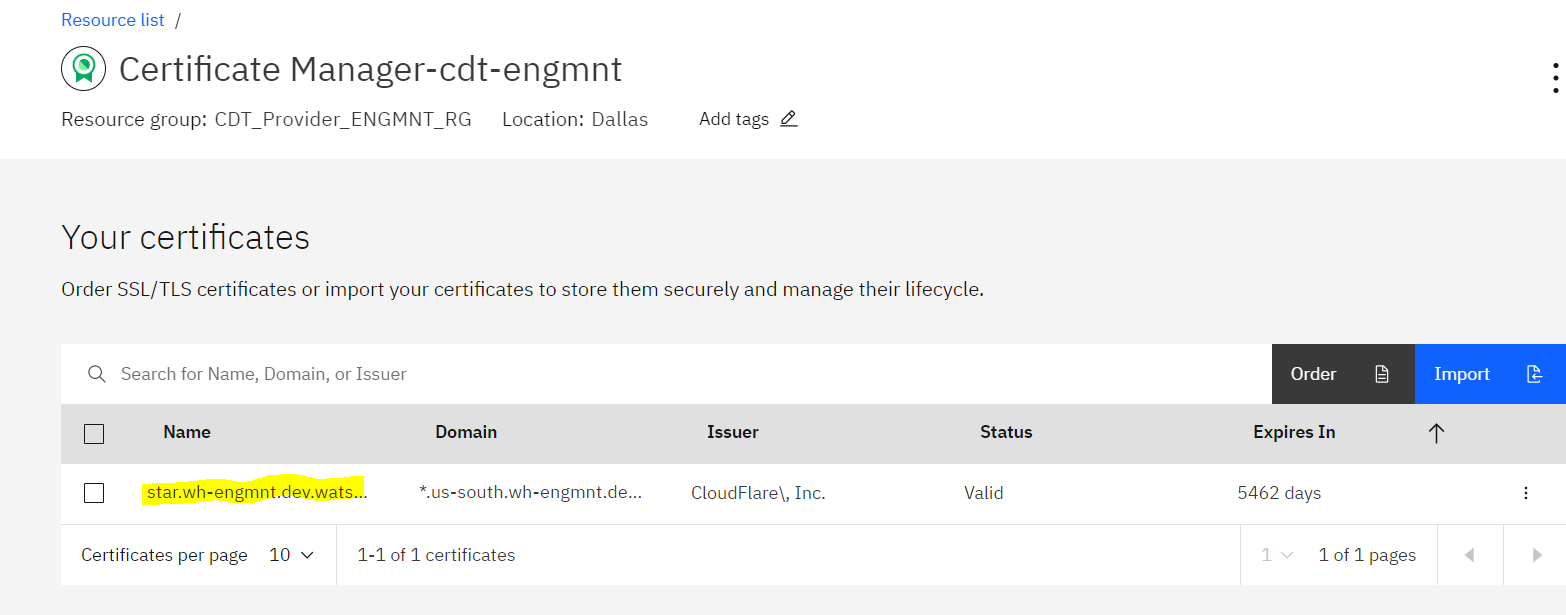


* Click on the **Certificate Manager** and click on **View full details**



* **Download the Certificate**

**Certificate Name:** star.wh-engmnt.dev.watson-health.ibm.com

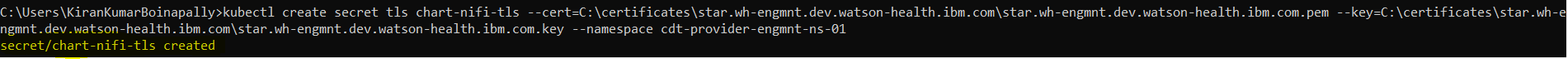


* **Deploy the secrets to the CDT namespace where NiFi will be installed:**

***Unzip the downloaded certificates and run the below command from IBM CDT cloud command window***

**kubectl create secret tls <secretName> --cert=<certificate-directory>/[pem-file-name].pem> --key=< certificate-directory/[key-file-name].key> --namespace <namespaceName>**

kubectl create secret tls chart-nifi-tls --cert=C:\certificates\star.wh-engmnt.dev.watson-health.ibm.com\star.wh-engmnt.dev.watson-health.ibm.com.pem --key=C:\certificates\star.wh-engmnt.dev.watson-health.ibm.com\star.wh-engmnt.dev.watson-health.ibm.com.key --namespace cdt-provider-engmnt-ns-01

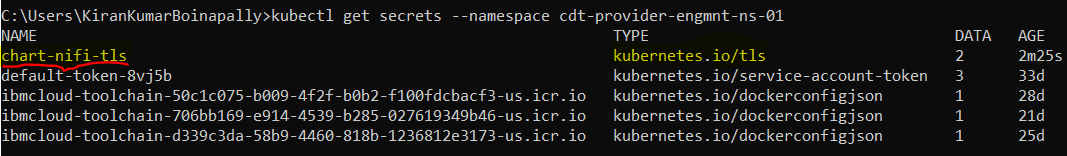


In the above screen shot, we can observe ‘**secret/chart-nifi-tls created**’

* **Validate that secret uploaded successfully or not**

kubectl get secrets --namespace <namespaceName>

kubectl get secrets --namespace cdt-provider-engmnt-ns-01



1. **Request CIS Routes for your cluster**

Raise an RTC ticket same as below for our cluster namespace.

**CIS RTC ticket:** <https://nsjazz.raleigh.ibm.com:8050/ccm/resource/itemName/com.ibm.team.workitem.WorkItem/327069>

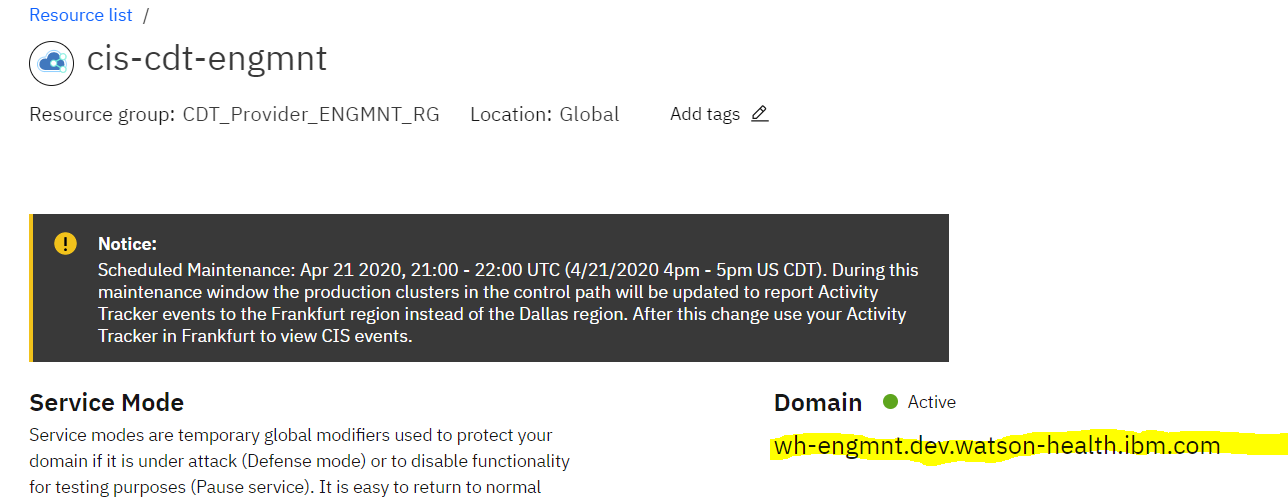
**Description:** I need a CIS Route for nifi-cdt-provider-engmnt-ns-01 with subdomain .wh-engmnt.dev.watson-health.ibm.com. This is needed for NiFi Distribution for Watson Health.

My cluster name space : cdt-provider-engmnt-ns-01. Subdomain :.wh-engmnt.dev.watson-health.ibm.com. Final URL/host should be nifi-cdt-provider-engmnt-ns-01.wh-engmnt.dev.watson-health.ibm.com

**Followed below URL Pattern for host:**

The ingress host we specify will be the base URL for accessing the UI and API from outside the cluster.

**Our domain in our TLS certificate secret is:** wh-engmnt.dev.watson-health.ibm.com



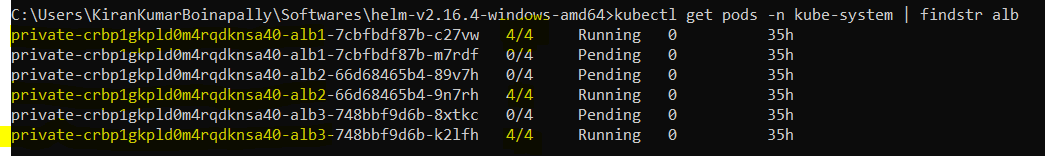
**Our URL pattern is:** nifi-${namespace}.{domain name}

**nifi**-**cdt-provider-engmnt-ns-01**.**wh-engmnt.dev.watson-health.ibm.com**

1. **Steps to add ALBs and Metadata to override.yaml file**

**Get the list of ALBs for my Ingress file**

kubectl get pods -n kube-system | findstr alb



**Add ALBs to override.yaml file**

alb\_id: private-crbp1gkpld0m4rqdknsa40-alb1;private-crbp1gkpld0m4rqdknsa40-alb2;private-crbp1gkpld0m4rqdknsa40-alb3



1. **A firewall rule to allow traffic to BluePages reverse proxy for your environment**

**Reverse Proxy**

Because bluepages.ibm.com is hosted on the internal BlueZone network, a reverse proxy is required to connect from services hosted in IBM Cloud. For WH environments (CDT/STG/PRD), these reverse proxies have been configured on VMWare machines at the following IPs:

CDT: 192.168.128.24

STG: ???

PRD: ???

You will need to ensure connectivity from your Kubernetes cluster to the corresponding proxy server by creating an RTC ticket with the cluster name, proxy IP, and port 636.

**LDAP Groups**

NiFi supports an LDAP Group Provider which allows the administrator to configure NiFi access policies based on LDAP group rather than adding users one by one. However, because of the way the provider queries LDAP for the groups and users, enabling this feature with BluePages comes with a few caveats:

Users must log in with their IBM serial number, which corresponds to the LDAP attribute uid. This includes the nifi.initial\_admin override value. When LDAP groups are disabled, users log in using their IBM email address.

**Raise an RTC ticket same as below for setup firewall rule to allow traffic to BluePages reverse proxy for solution:**

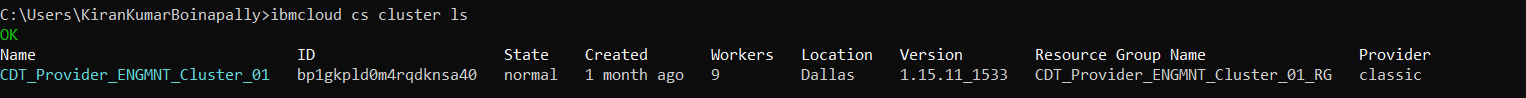
**Description:** Please setup firewall rule to allow traffic to BluePages reverse proxy for solution. This is needed for NiFi Distribution for Watson Health

RTC ticket URL: <https://nsjazz.raleigh.ibm.com:8050/ccm/resource/itemName/com.ibm.team.workitem.WorkItem/326796>

1. **Installing IBM Cloud Block Storage Plugin on our CDT cluster**

Install the IBM Cloud Block Storage plug-in with a Helm chart to set up pre-defined storage classes for Block Storage. You can use these storage classes to create a PVC to provision Block Storage for your apps.

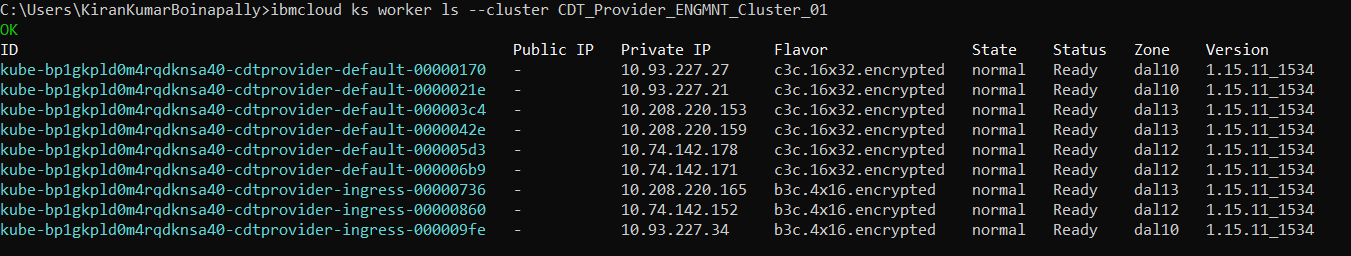
**Before you begin:** [Log in to your account. If applicable, target the appropriate resource group. Set the context for your cluster.](https://cloud.ibm.com/docs/containers?topic=containers-cs_cli_install#cs_cli_configure)



1. Make sure that your worker node applies the latest patch for your minor version to run your worker node with the latest security settings. The patch version also ensures that the root password on the worker node is renewed.

List the current patch version of your worker nodes

**ibmcloud ks worker ls --cluster <cluster\_name\_or\_ID>**



1. From IBM CDT Cloud CLI, change the directory to Helm directory



1. Add the IBM Cloud Helm chart repository to the cluster where you want to use the IBM Cloud Block Storage plug-in

Run the below command

**helm repo add iks-charts https://icr.io/helm/iks-charts**





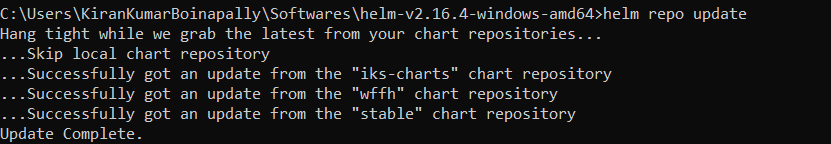
1. Update the Helm repo to retrieve the latest version of all Helm charts in this repo

Run the below command

**helm repo update**







1. Install the IBM Cloud Block Storage plug-in. When you install the plug-in, pre-defined block storage classes are added to your cluster.

**Note:** For installation of this plugin, RTC Work Item 326795 raised against **whc-cdt-sdt-triage** slack channel.

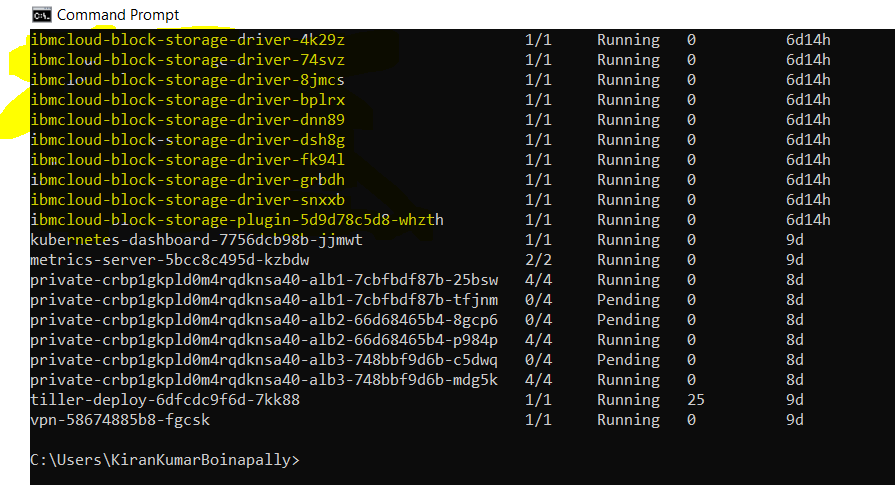
Below are the ticket details:

**Description:** Please install block storage plugins onto the cluster, also enable viewing storage classes.

<https://nsjazz.raleigh.ibm.com:8050/ccm/resource/itemName/com.ibm.team.workitem.WorkItem/326795>

1. Verify that the installation was successful

Got the below result when ran **kubectl get pod -n cdt-provider-engmnt-ns-01**



**Installation Steps**

**A . Deploy Kubernetes Secrets**

***nifi-ca-server-token***

* **Creating nifi-ca-server-token secret**

Contains a single key token which value must be a strong alphanumeric value at least 16 characters long. It is used by the NiFi nodes to authenticate with the CA server when requesting TLS certificates. You generate this value yourself, suggestion would be to use a password generator (such as [1Password](https://ibm.ent.1password.com/home)).

**Example:**

apiVersion: v1

kind: Secret

type: Opaque

metadata:

name: nifi-ca-server-token

stringData:

token: aBcDeFgH1234567890

**Note: I have used above content in the secret.yaml file and applied to the CDT cluster namespace using below command.**

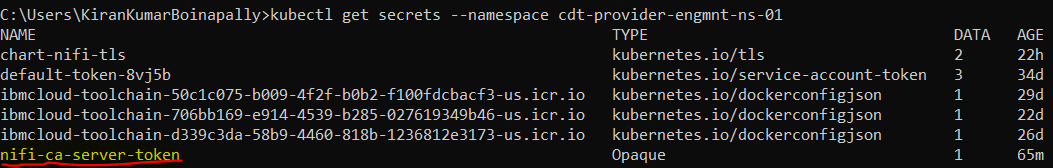
* **Deploying nifi-ca-server-token secret to CDT cluster namespace**

kubectl apply -f ./secret.yaml --namespace cdt-provider-engmnt-ns-01

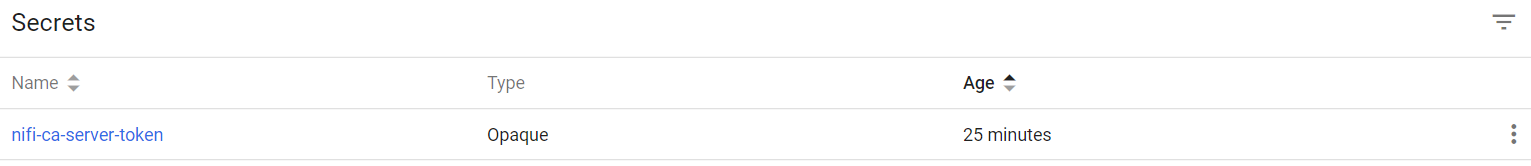


* **Verify nifi-ca-server-token secret applied on CDT cluster namespace**

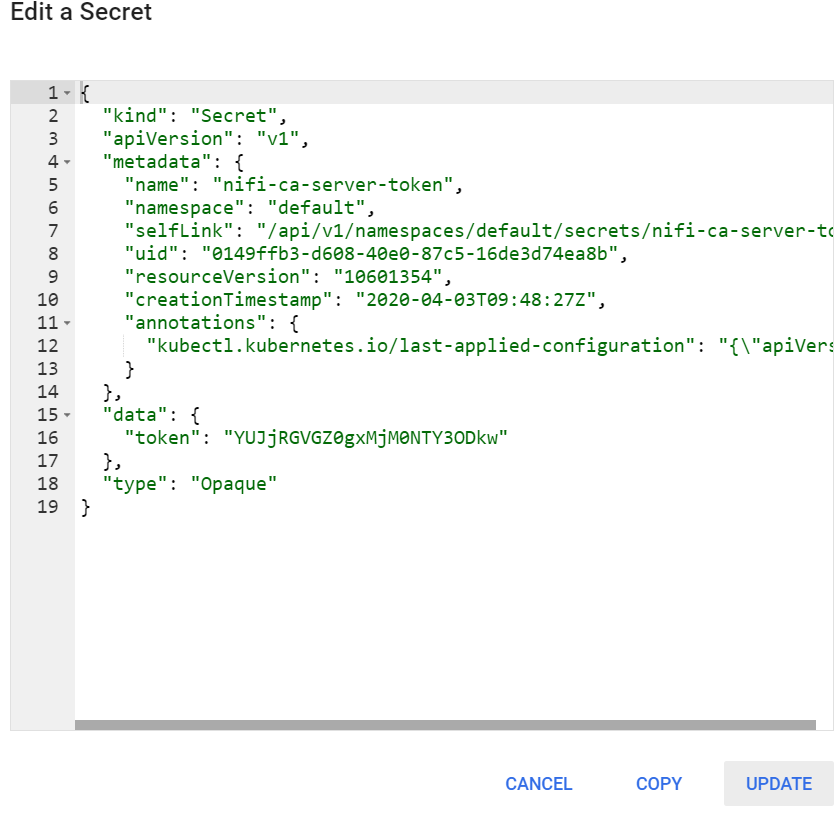
kubectl get secrets --namespace cdt-provider-engmnt-ns-01



* **nifi-ca-server-token secret created in IBM CDT cloud UI**



* **Content of** **nifi-ca-server-token secret in IBM CDT cloud UI**



***nifi-registry-git-credentials***

* **Creating nifi-registry-git-credentials secret**

Contains two keys: **username** and **password**. The username should be the GitHub user that NiFi Registry will use to commit changes to the flow repository, and the password should be a **GitHub API key** with repo permissions.

**Example:**

apiVersion: v1

kind: Secret

type: Opaque

metadata:

name: nifi-registry-git-credentials

stringData:

username: Github-Username

password: GitHubApiKeyGoesHere

**Note: I have used above content in the secret.yaml file and applied to the CDT cluster namespace using below command.**

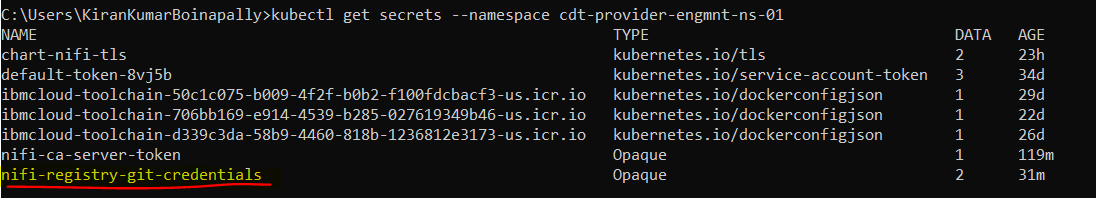
* **Deploying nifi-registry-git-credentials secret to CDT cluster namespace**

kubectl apply -f ./secrets.yaml --namespace cdt-provider-engmnt-ns-01

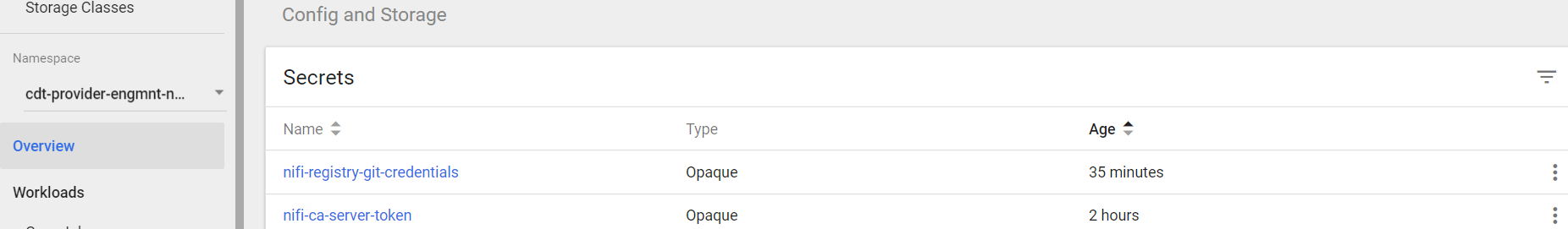


* **Verify nifi-registry-git-credentials secret applied on CDT cluster namespace**

kubectl get secrets --namespace cdt-provider-engmnt-ns-01



* **nifi-registry-git-credentials secret created in IBM CDT cloud UI**



* **Content of** **nifi-registry-git-credentials secret in IBM CDT cloud UI**



**B. Create a Helm override file**

Create an override.yaml file for your cluster. It should look something like below:



**In the above override.yaml file, we have pullSecret under image. If we don’t have below secrets under our cluster namespace, then we need to add them by following below steps.**

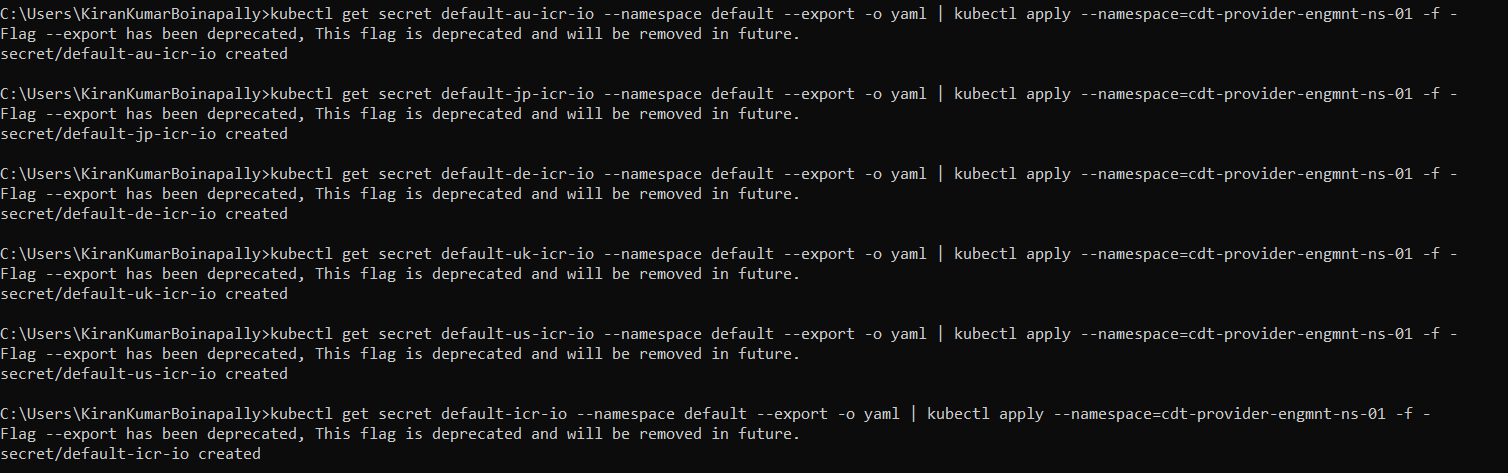
**These secrets pull the respective docker images.**

**Adding default-\*-icr-io secrets to CDT namespace**

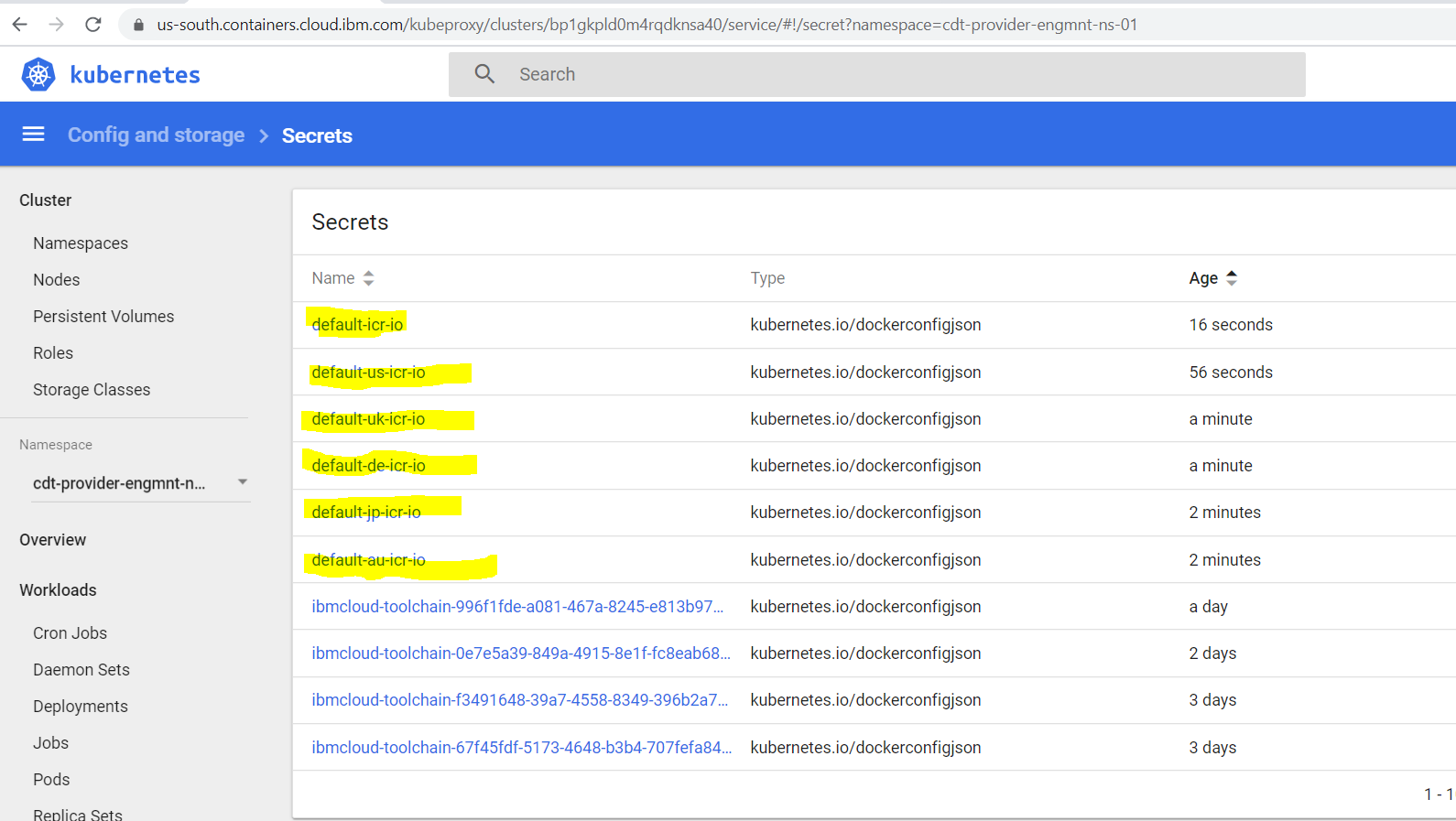
All the default-\*-icr-io secrets allow your namespace access to the IBM Container Registry to pull docker images

Run the below command to add the respective secret individually.

**kubectl get secret default-\*-icr-io --namespace=default --export -o yaml | kubectl apply --namespace=cdt-provider-engmnt-ns-01 -f -**



**Secrets under CDT namespace after adding**



**C. Add the WFFH Orchestration chart to your helm install**

Execute the following commands to add the WFFH repository to your local helm client repository and verify operation:

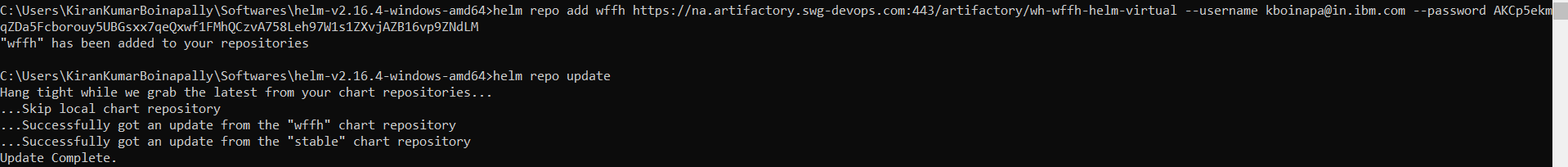
**Add JFrog Artifactory’s User and API Key details:**

helm repo add wffh https://na.artifactory.swg-devops.com:443/artifactory/wh-wffh-helm-virtual --username <YOUR\_ARTIFACTORY\_USER> --password <YOUR\_ARTIFACTORY\_API\_KEY>

helm repo add wffh https://na.artifactory.swg-devops.com:443/artifactory/wh-wffh-helm-virtual --username kboinapa@in.ibm.com --password <ARTIFACTORY\_API\_KEY>

**Helm Repository Update**

helm repo update



**D. Install or Upgrade the Helm Chart**

**1. Execute the following command to install or upgrade the helm chart in a specified namespace:**

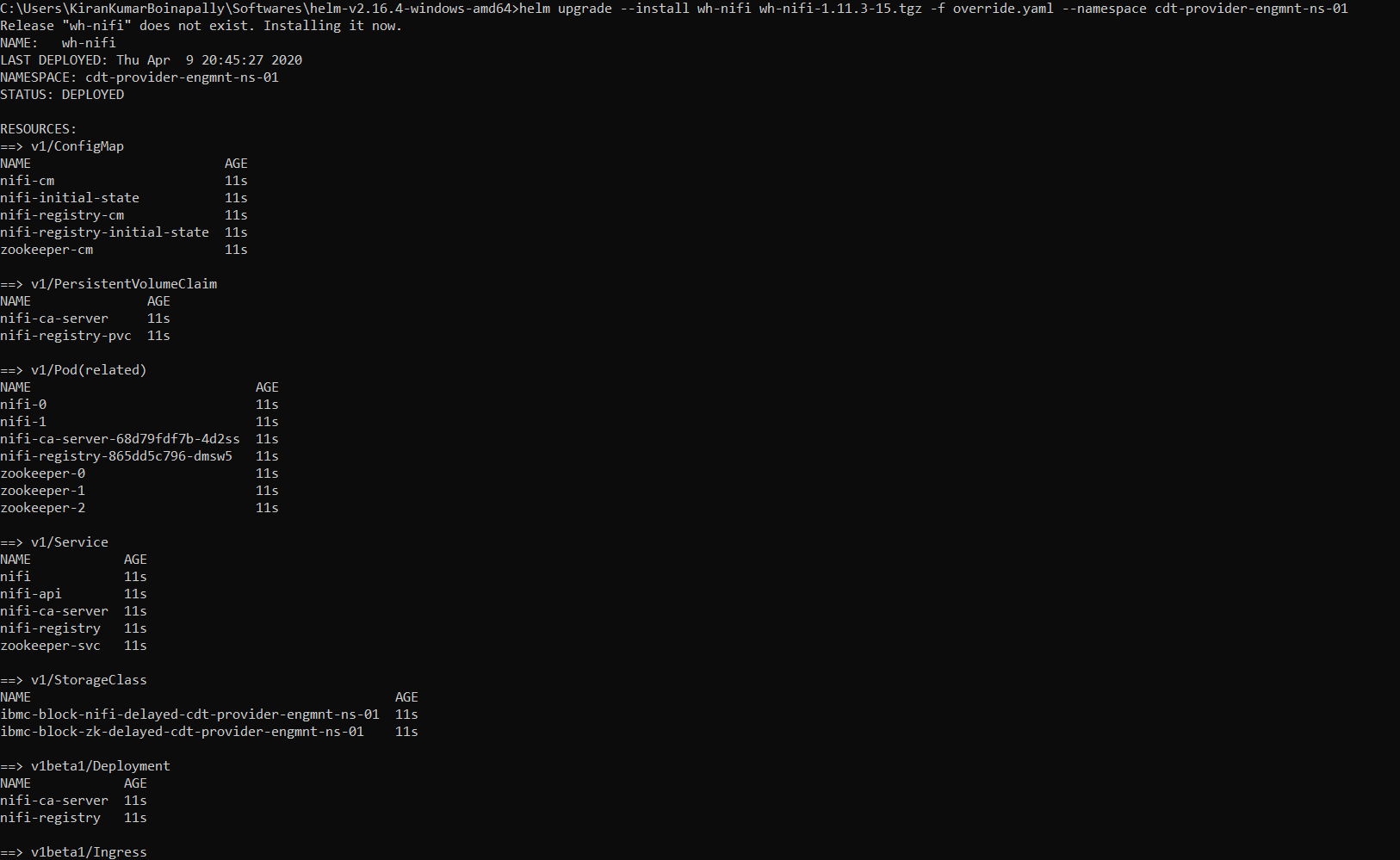
**Syntax:**

**helm upgrade --install wh-nifi wh-nifi-1.11.3-15.tgz -f override.yaml --namespace <my\_namespace>**

**Command:**

helm upgrade --install wh-nifi wh-nifi-1.11.3-15.tgz -f override.yaml --namespace cdt-provider-engmnt-ns-01





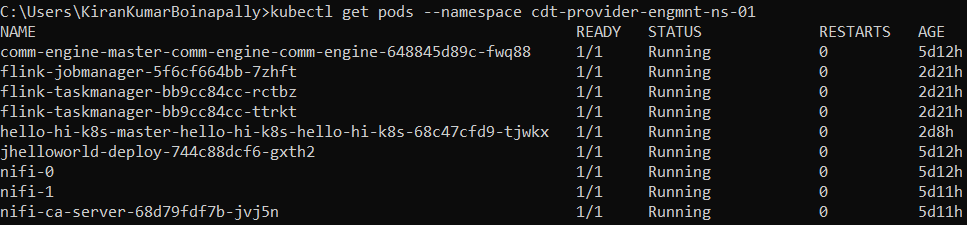
**2. Run below command to check whether the nifi pods are up and running**

**Syntax:**

kubectl get pods --namespace <cluster namespace>

**Command:**

kubectl get pods --namespace cdt-provider-engmnt-ns-01

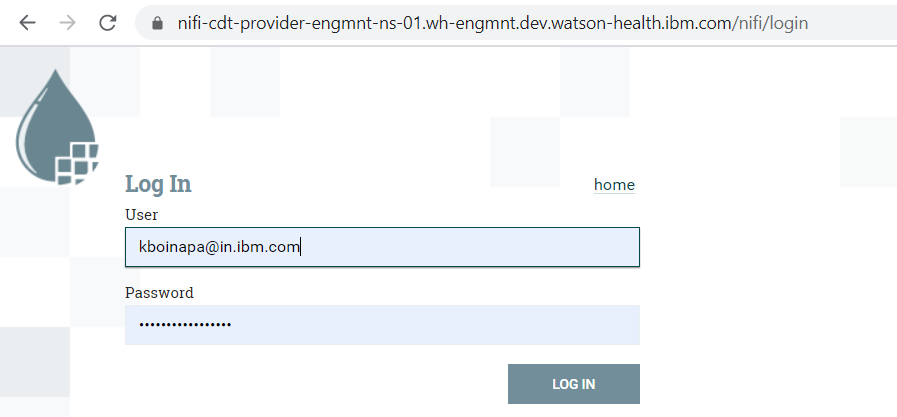


**E. Access the UI**

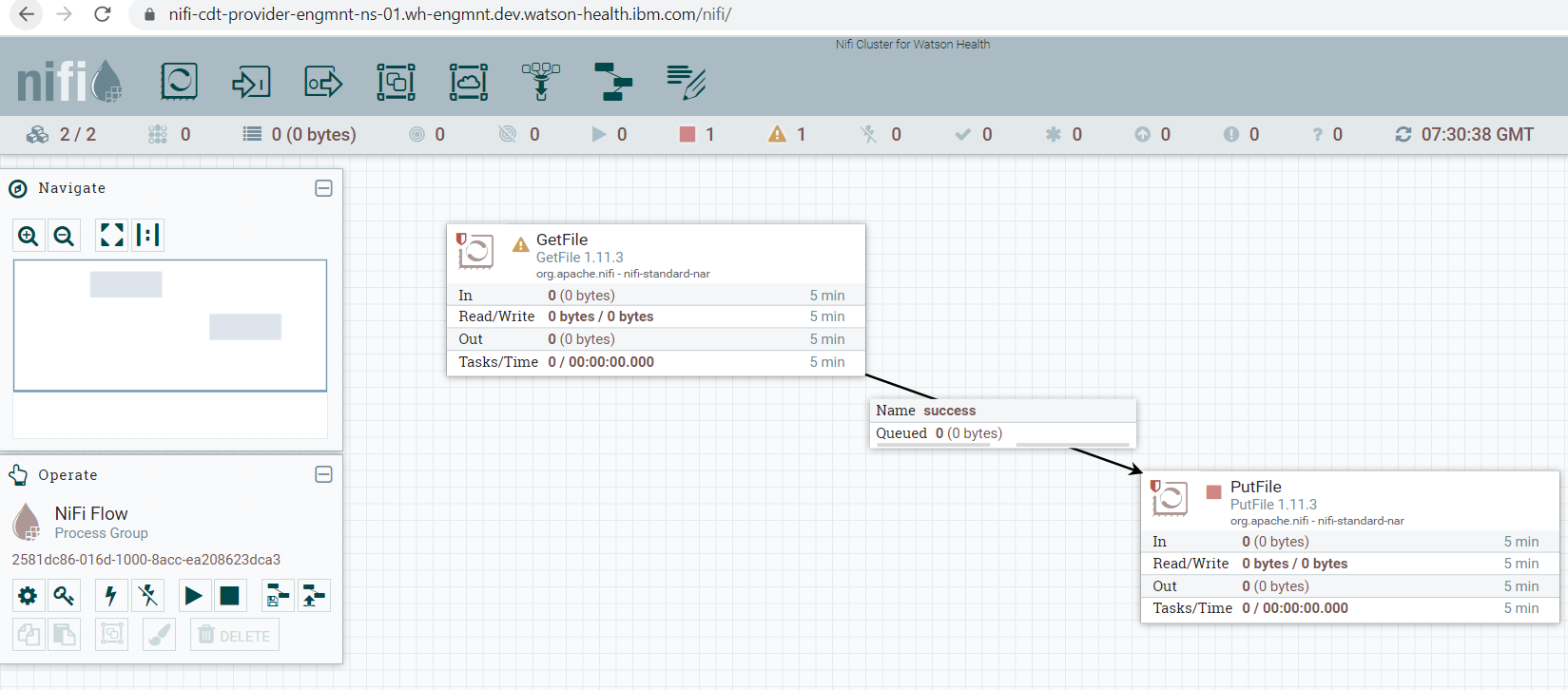
**Access the below URL in the browser:**

<https://nifi-cdt-provider-engmnt-ns-01.wh-engmnt.dev.watson-health.ibm.com/nifi>

**Login Screen:**



**Created GetFile and PutFile processes:**



**Creation of Users:**

