ARGOCD IMAGE UPDATER INSTALLATION STEPS:

**Pre-requisites:**

1. Have a minikube/kind cluster up.
2. Install argocd and have the UI working, when port forwarded to port 8080.

- kubectl port-forward svc/argocd-server -n argocd 8080:443

1. Login to the dashboard using “admin” as username and password can be picked from here:

- kubectl get secret -n argocd argocd-initial-admin-secret -o jsonpath='{.data.password}' | base64 --decode

**Steps:**

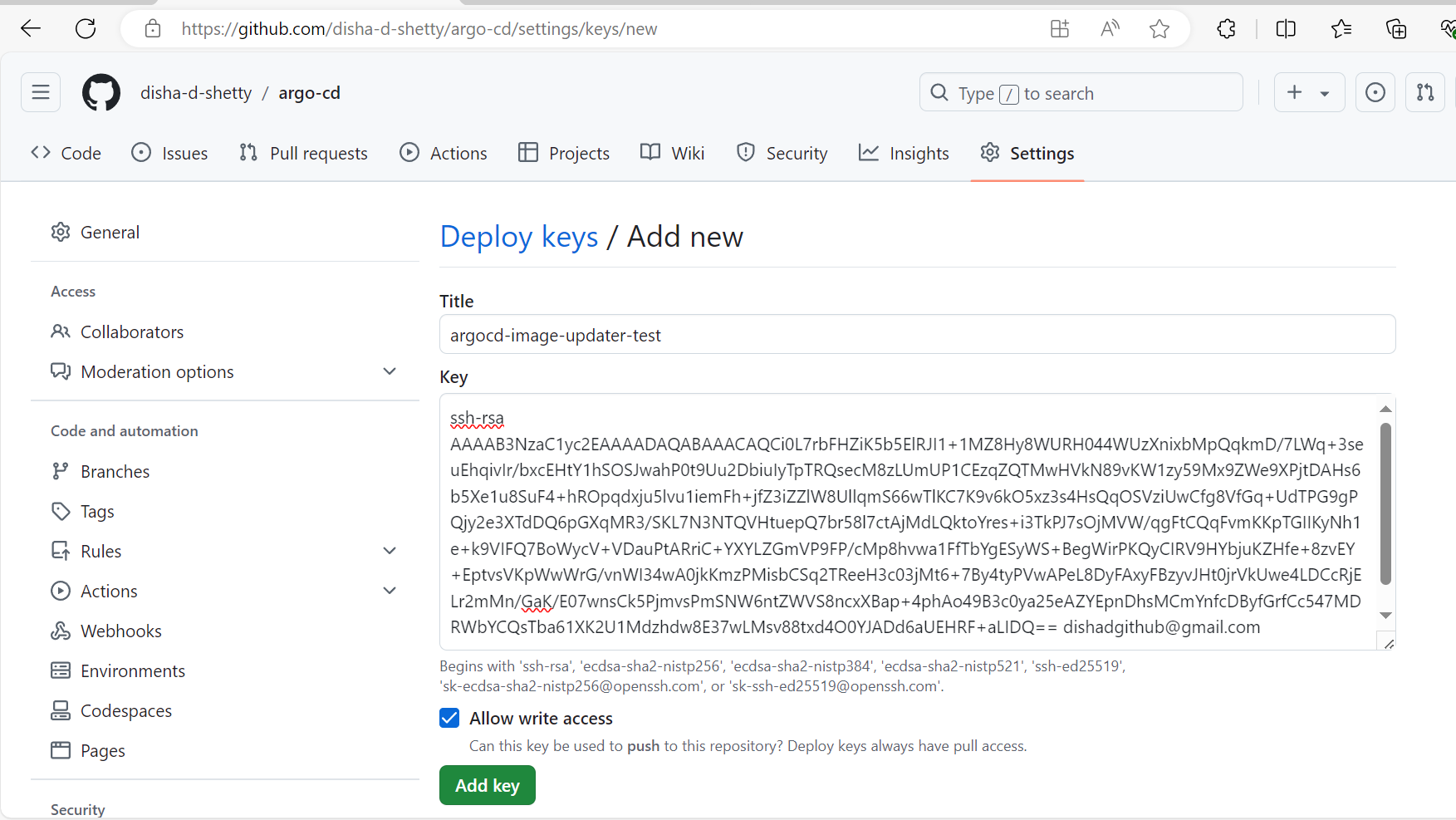
1. Install the required manifest under the “argocd” namespace:

kubectl apply -n argocd -f <https://raw.githubusercontent.com/argoproj-labs/argocd-image-updater/stable/manifests/install.yaml>

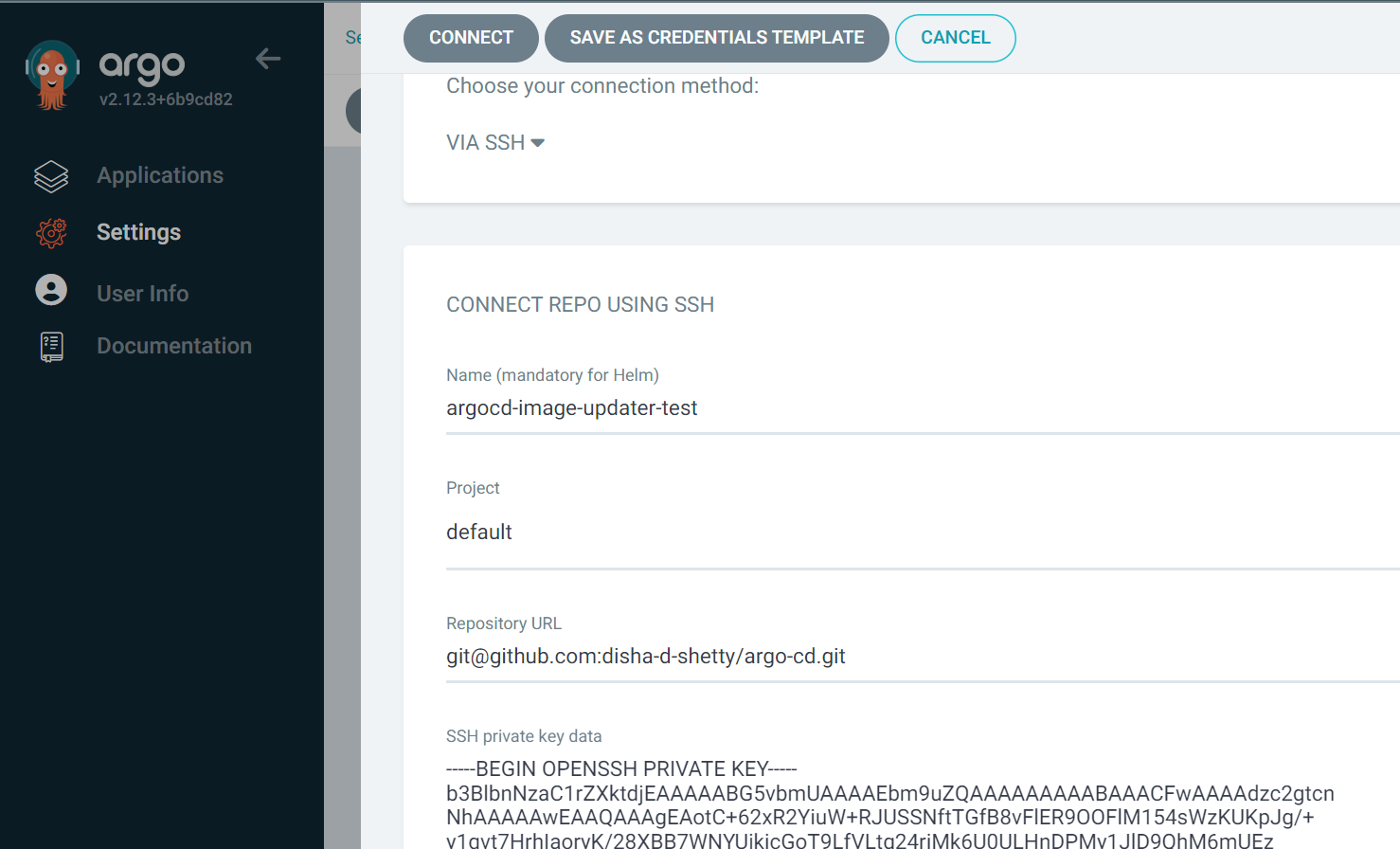
1. To authenticate with github repository and to force write back the updated image, to avoid out out of sync on argocd, connect argocd with github using ssh:
   1. Generate ssh keys:

ssh-keygen -t rsa -b 4096 -C "example@gmail.com" # use your emailid here

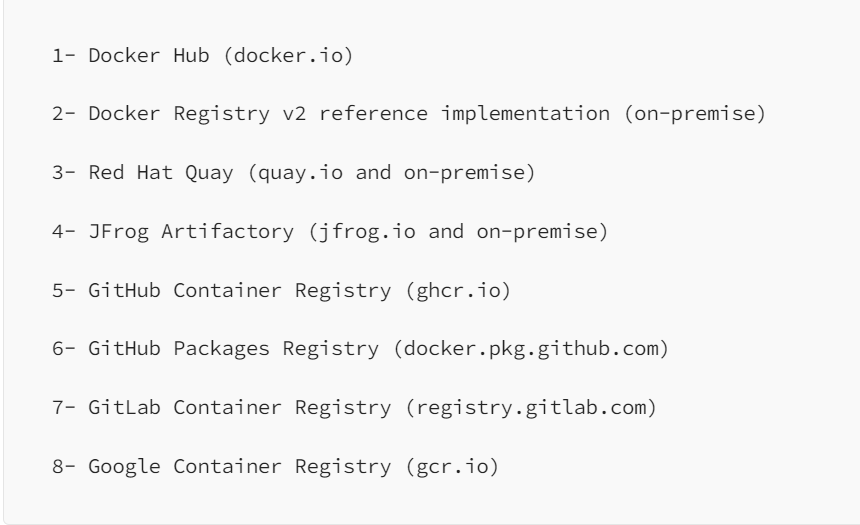
* 1. Add public key (starting from ssh-rsa) to “Deploy-key” section on github repo’s settings and give write access.



* 1. Add the private key to “repository” -> “connect” section of argocd’s settings and provide the correct github repo URL.



1. For image registry connection, image-updater supports only these:



Here docker hub is been used, so create a secret with docker hub creds and store in under “argocd” namespace.

kubectl create secret docker-registry dockerhub-secret   --docker-username=<username>   --docker-password=<password>   --docker-email=<mail-id>   --docker-server=https://registry-1.docker.io -n argocd

1. Edit “argocd-image-updater-config” configmap in “argocd” namespace, provide proper **registry detail** accordingly. Eg:

apiVersion: v1

data:

  log.level: debug

  registries.conf: |

    registries:

      - name: Docker Hub

        api\_url: https://registry-1.docker.io

        credentials: pullsecret:argocd/dockerhub-secret

        defaultns: library

        default: true

kind: ConfigMap

metadata:

  annotations:

    kubectl.kubernetes.io/last-applied-configuration: >

      {"apiVersion":"v1","kind":"ConfigMap","metadata":{"annotations":{},"labels":{"app.kubernetes.io/name":"argocd-image-updater-config","app.kubernetes.io/part-of":"argocd-image-updater"},"name":"argocd-image-updater-config","namespace":"argocd"}}

  creationTimestamp: '2024-10-15T09:10:49Z'

  labels:

    app.kubernetes.io/name: argocd-image-updater-config

    app.kubernetes.io/part-of: argocd-image-updater

  name: argocd-image-updater-config

  namespace: argocd

  resourceVersion: '254917'

  uid: 8f909c7b-7fae-4b53-8fac-8fd4187ded29

1. Create a application.yaml/ through UI with proper **imager updater annotations**. Point the yaml to read the helm manifests.

Eg:

Here the update strategy is set to semver. Many other strategies are supported.

apiVersion: argoproj.io/v1alpha1

kind: Application

metadata:

  name: argocd-test-app

  namespace: argocd

  annotations:

    argocd-image-updater.argoproj.io/image-list: myalias=docker.io/dishadgithub/argocd-test   #Write repository name

    argocd-image-updater.argoproj.io/myalias.pull-secret: pullsecret:argocd/dockerhub-secret  # You can remove this annotation if you have already configured the docker secret and are using the same secret to retrieve all docker images in your deployments.

    argocd-image-updater.argoproj.io/myalias.update-strategy: semver #There are several ways to update the image, but I'm using senver.

    argocd-image-updater.argoproj.io/myalias.semver: ">=0.0.0-0"

    argocd-image-updater.argoproj.io/write-back-method: git  # We don't need the secret as mentioned in argocd doc because we already set it up via argocd console if you're using the same git repo.

    argocd-image-updater.argoproj.io/git-branch: feature/initial-setup #point to branch ehere helm is. For main -"HEAD"

    argocd-image-updater.argoproj.io/myalias.force-update: "true" #true to write on repo

  finalizers:

    - resources-finalizer.argocd.argoproj.io

  labels:

    name: argocd-test-app

spec:

  project: default

  source:

    repoURL: git@github.com:disha-d-shetty/argo-cd.git

    targetRevision: feature/initial-setup

    path: argocd-test-app # path of the kubernetes application code means where you stored deployment.yaml, service.yaml etc

  destination:

    server: https://kubernetes.default.svc  # if you want to deploy application in the same cluster where argocd server is running if thats diffrent then use cluster IP address.

    namespace: test

  syncPolicy:

    syncOptions:

    - CreateNamespace=true

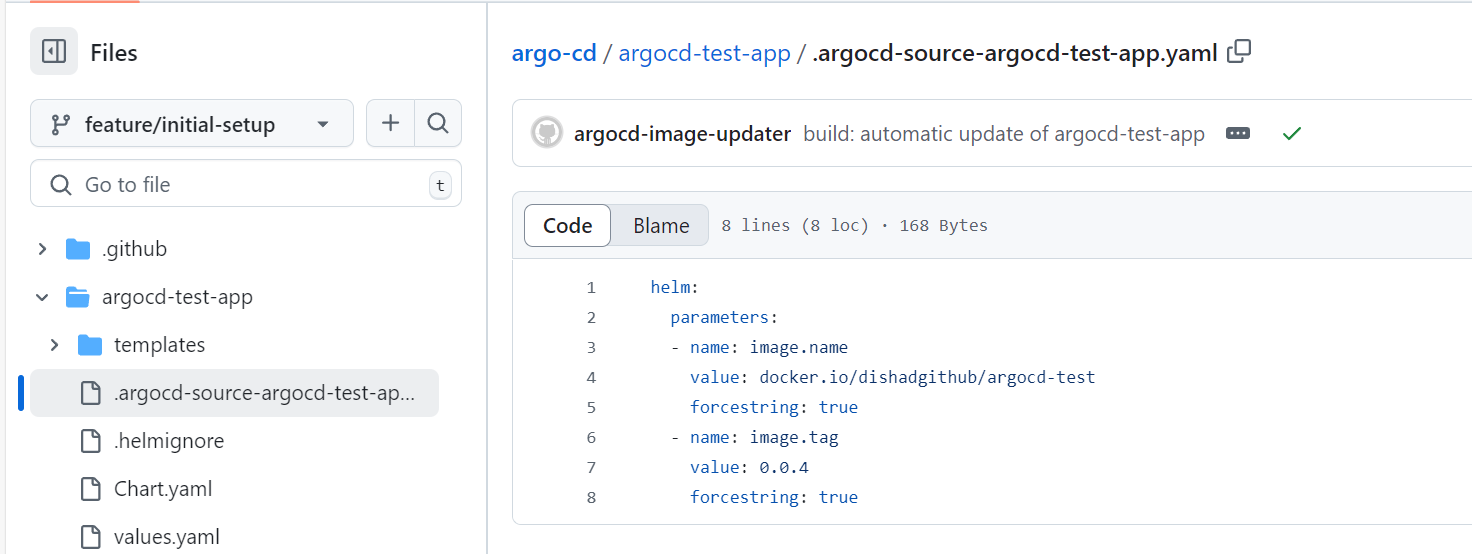
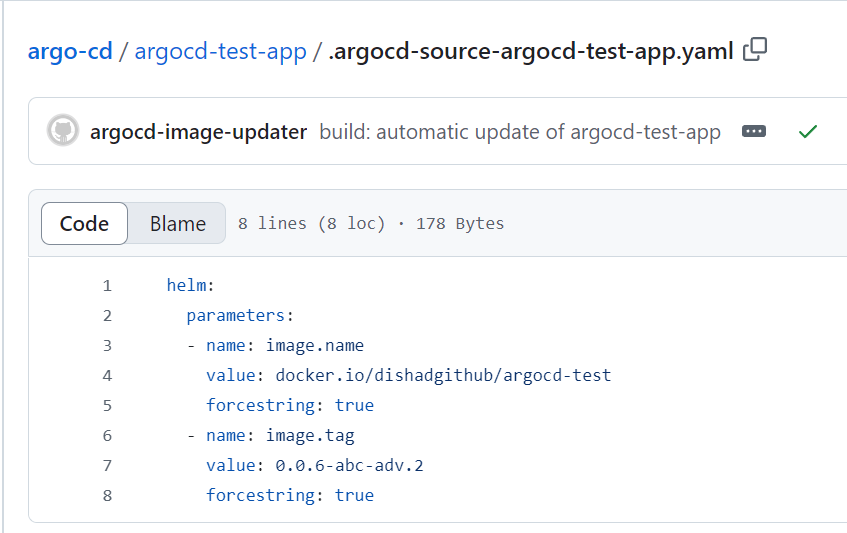
    automated:

      selfHeal: true

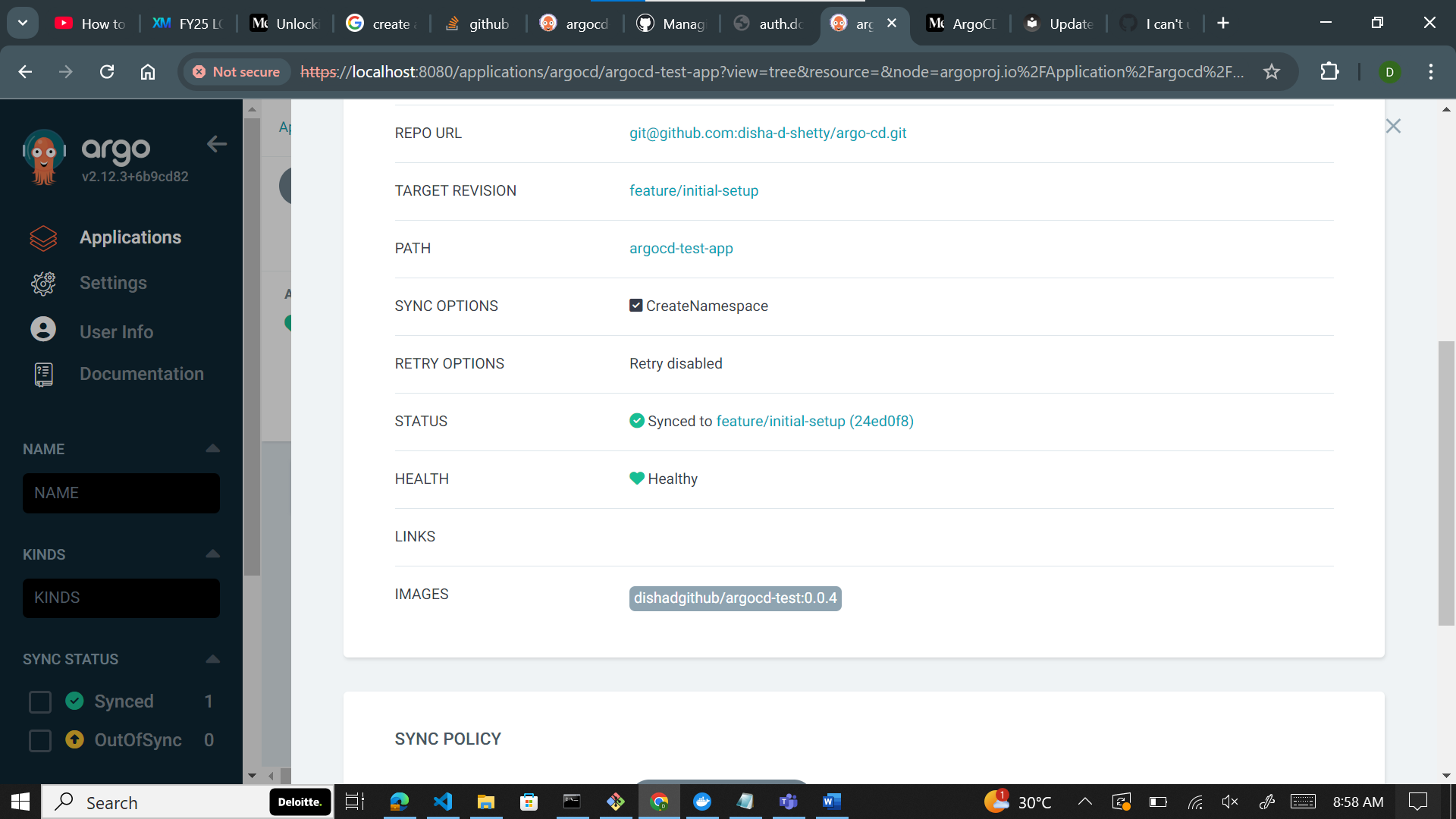
      prune: true

kubectl apply -f application.yaml

1. Once all the changes are deployed and when there is an update in image(tag version satisfying “image list” in application.yaml annotation), there is **automated commit** made by image updater on the helm manifest folder.

On UI, verify the version updation( Here “0.0.2”(hardcoded values.yaml) got updated to “0.0.4”(latest on docker-hub)).



1. To debug image updater issues, use this cmd:

kubectl --namespace argocd logs --selector app.kubernetes.io/name=argocd-image-updater –follow

**Ref Links:**

1. Update-strategies: <https://argocd-image-updater.readthedocs.io/en/stable/basics/update-strategies/>
2. Using GCR: <https://medium.com/@megaurav25/argocd-image-updater-fbc93dc15e34>
3. Docherhub API issue: <https://github.com/argoproj-labs/argocd-image-updater/issues/377>