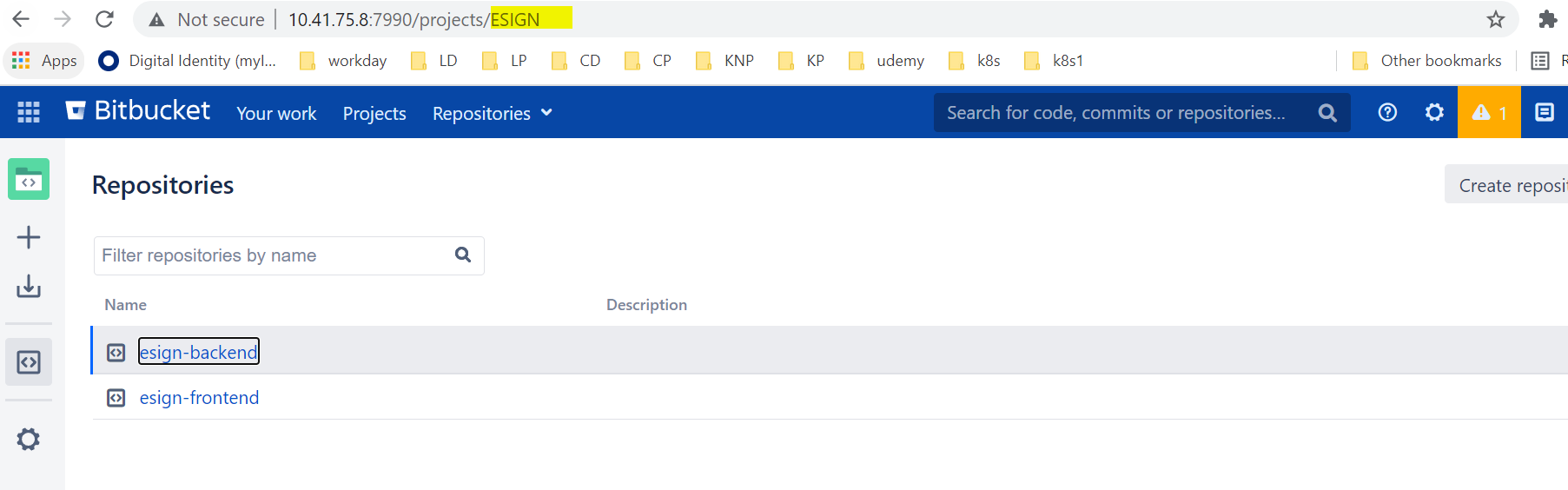
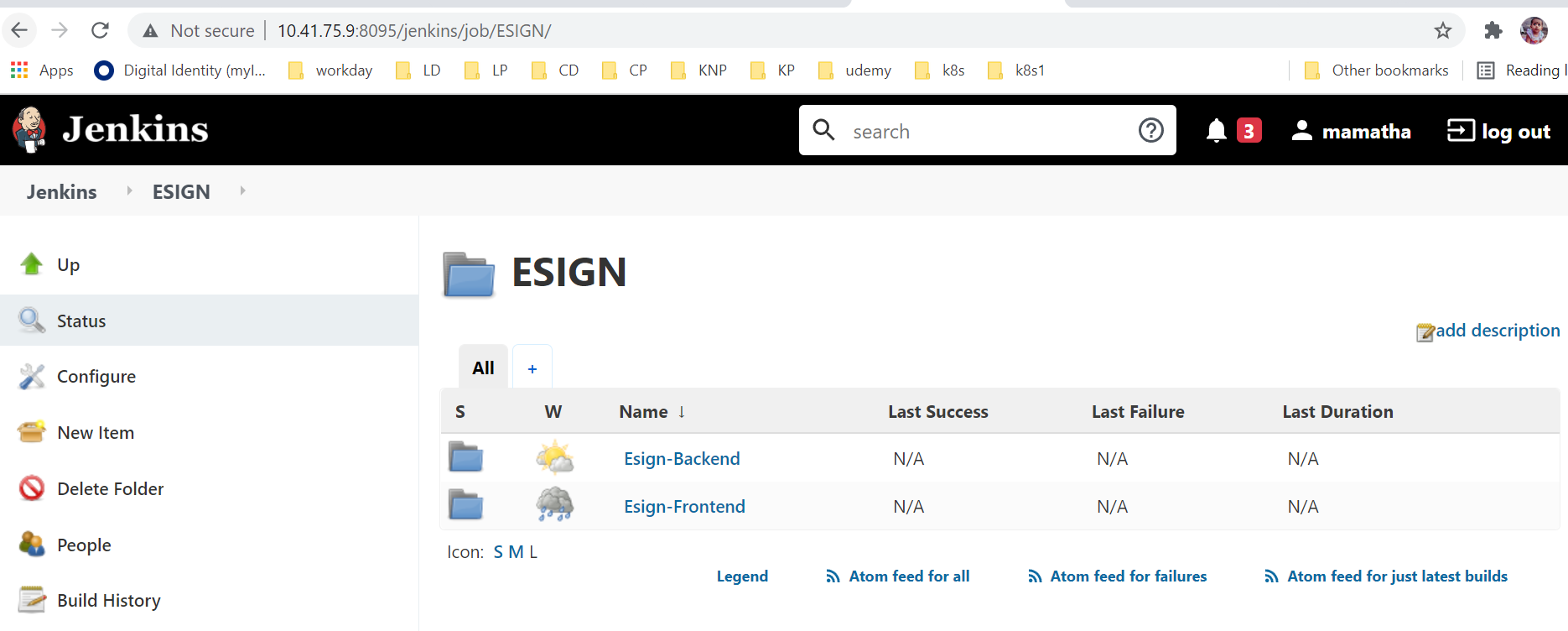
ESIGN

Bitbucket: <http://10.41.75.8:7990/projects/ESIGN>



Jenkins: [**http://10.41.75.9:8095/jenkins/job/ESIGN/**](http://10.41.75.9:8095/jenkins/job/ESIGN/)



**BUILD SERVER**:

DEALW11390.emea.zurich.corp

/app/devops/ansible-workspace/playbook/esign

**DEPLOYSERVER**:

DEALA09998.emea.zurich.corp

/datavg/esign/

|  |  |
| --- | --- |
| Here are the IPs for  DEV, UAT and PROD for esign frontend app | Here are the IPs for  DEV, UAT and PROD for esign backend app |
| dev -   [http://10.41.72.150:31028](http://10.41.72.150:31028/) uat -    [http://10.41.72.150:31030](http://10.41.72.150:31030/)  prod - [http://10.41.72.150:31029](http://10.41.72.150:31029/) | http://10.41.72.150:31031/  http://10.41.72.150:31032/  http://10.41.72.150:31033/ |

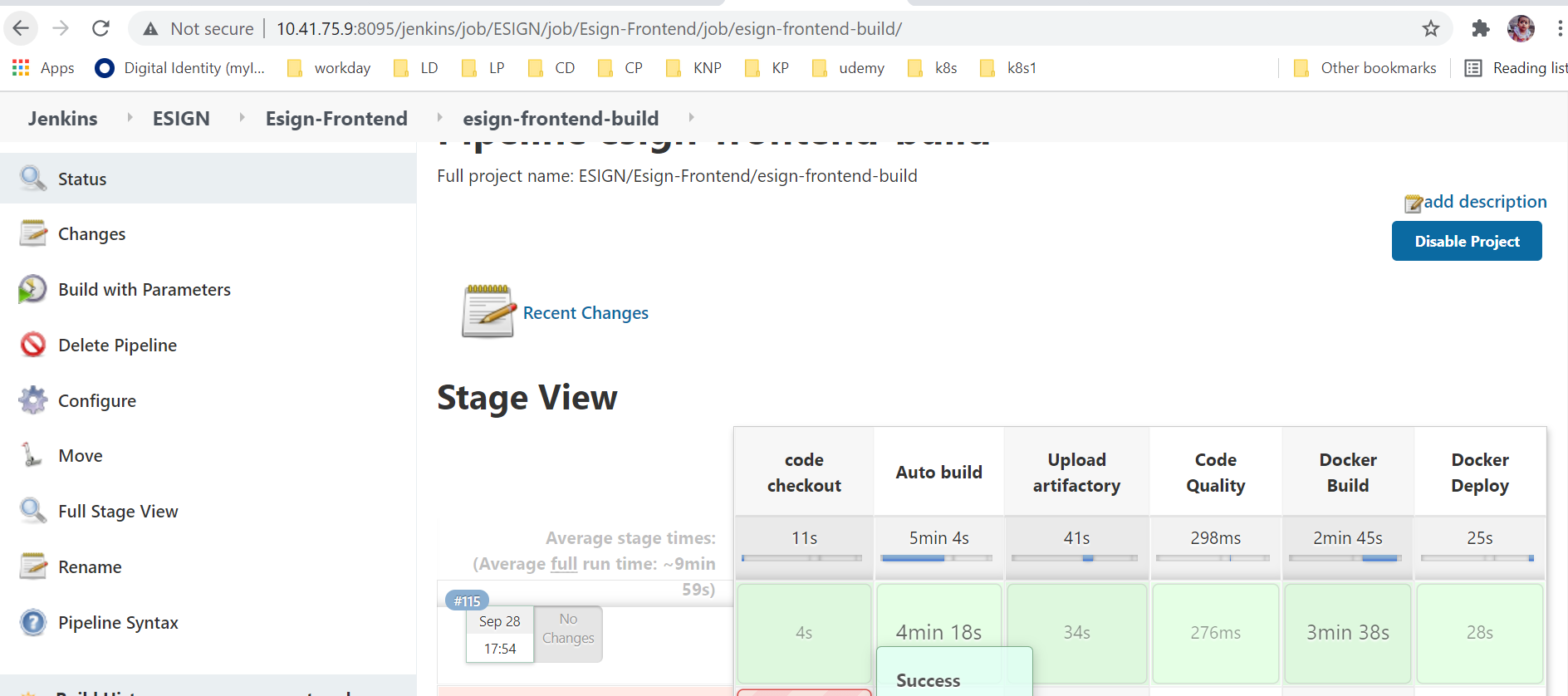
**Esign-Frontend**

Below jobs available in ESIGN (folder)🡪Esign-Frontend(folder)

1. esign-frontend-build
2. esign-frontend-Docker-Build
3. esign-frontend-deploy

**1.esign-frontend-build:**

Below stages available in this job



**code checkout**: Below is default Jenkins workspace path and the code will clone in this location.

**Auto build**: When build occurs the target folder will come into below path.

**Upload artifactory**: This is used Archive jar from the target folder in below location.

We are uploading from below location

To this url <http://10.41.75.8:8061/artifactory>

We need to create the repo in this url <http://10.41.75.8:8061/artifactory>

**Docker Build**: we are passing the parameters to *esign-frontend-Docker-Build*

gitusername and gitpassword are used to checkout the Dockerfile.

ARFTPWD and ARFTUID are used to download jar from artifactory and push the docker image to artifactory

Branchname nothing but version and image version,which version jar has to be downloaded.

**Docker Deploy:** we are passing the parameters to *esign-frontend-deploy*

gitusername and gitpassword are used to checkout the k8s yaml files.

ARFTPWD and ARFTUID are used to pull the image.

Version:Which version of the image that has to be downloaded(for the deploymenet)

Branch:Which branch of the yaml files we have to downloaded it.

**2.esign-frontend-Docker-Build**

cd /app/devops/dockerbuild/esign

curl "http://10.41.75.8:8061/artifactory/esign-frontend/latest/esign-frontend.zip" -o esign-frontend.zip

we are able to download the zip file but not able to unzip in dockerfile. That’y we are downloading and unzipping in job.

cp -rf z-esign-lib /app/devops/dockerbuild/esign/esign-frontend/

Dockerfile available in esign-frontend that’y we are coping unziped file to esign-frontend folder.

To run the dockerfile we can go to this esign-frontend

Now dockerfile and unzipped files are available in esign-frontend

cd esign-frontend

Below command it used to create the docker image and dot(.) will take Dockerfile in esign-frontend folder.

sudo docker build -t arty.zurich.co.uk:5003/esign/latest/esignfrontend:$branchname . --no-cache

Dockerfile:

We need below image for npm install and we pulled image and tagged to our artifactory. So we can use in our Dockerfile.

Docker pull node:alpine3.14

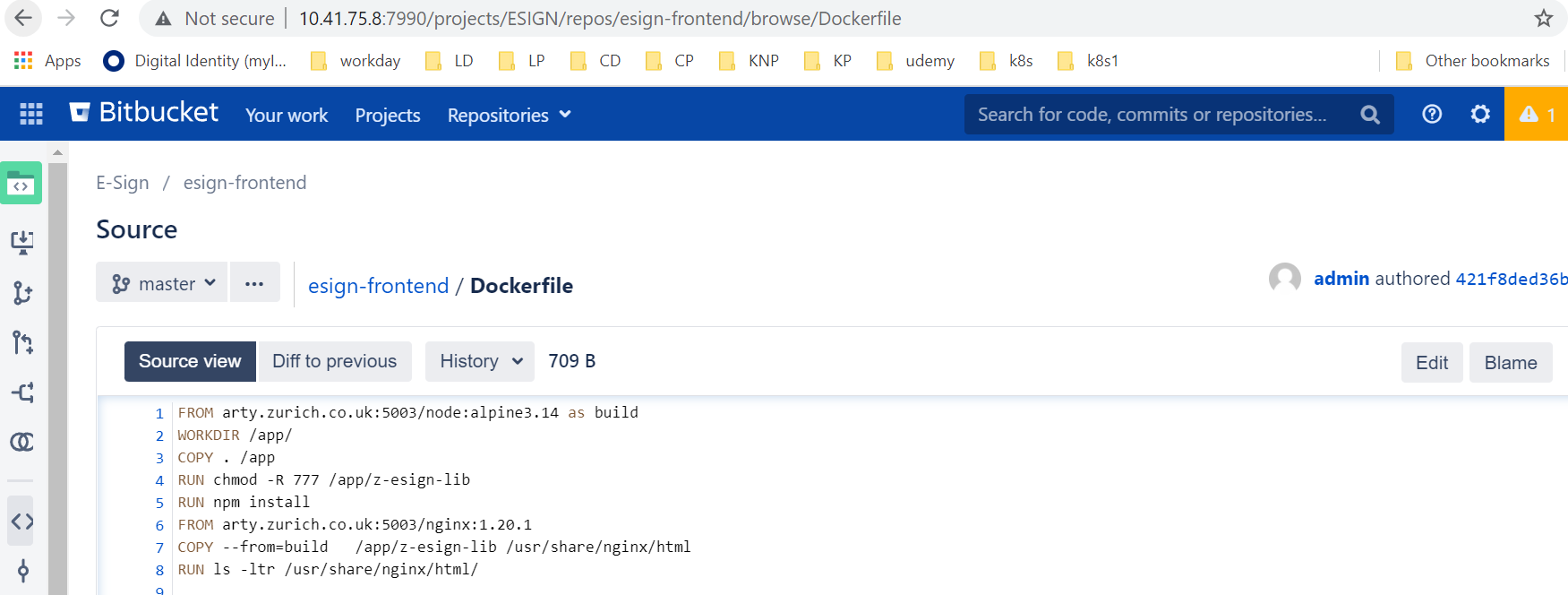
Docker tag node:alpine3.14 arty.zurich.co.uk:5003/node:alpine3.14

Work directory /app/ and copying current files to /app/

We are using Nginx to run our application and we are using nginx image.

Copying unzip folder to nginx.

COPY --from=build /app/z-esign-lib /usr/share/nginx/html



Once the image created we are tagging with different name as well because buckup purpose.

we are pushing the image to build server

removing image from build server

**3.esign-frontend-deploy**

This esign-frontend.yaml available in cd /app/devops/ansible-workspace/playbook/esign.

executing this below command through the ansible playbook and this yaml involking with this below inputs

ansible-playbook esign-frontend.yaml -e "version=$version branchname=$branchname gitusername=$gitusername gitpassword=$gitpassword ARFTUID=$ARFTUID ARFTPWD=$ARFTPWD env=$env" || exit 1

we have some tasks in this yaml file and we are passing required parameters to checkoutservices.sh and deployservices.sh. It will connect to this server

1.Task: It is used to validate the nodename.it will print the nodename.

2.Task: Goto this path cd /datavg/esign/esign-frontend in this **DEALA09998.emea.zurich.corp** server and create checkoutservices.sh file for executing the script inside it.

In checkoutservices.sh file,we are removing exesting service and again cloning.

3.Task: Goto this path cd /datavg/esign/esign-backend in this **DEALA09998.emea.zurich.corp** server and create deployservices.sh file for executing the script inside it.

\*The deployment yml files available in esign-frontend /k8s go to this below path

\*sed command is used to change the IMAGE\_VERSION to image\_version(user declearing version from jobs ) applied in esign-frontend -dev.yaml

\*we are deleting exesting yml and again applying yaml files

\*after that validate wheather the service is completely up or not once it completely up then sent back to the deployment is success.

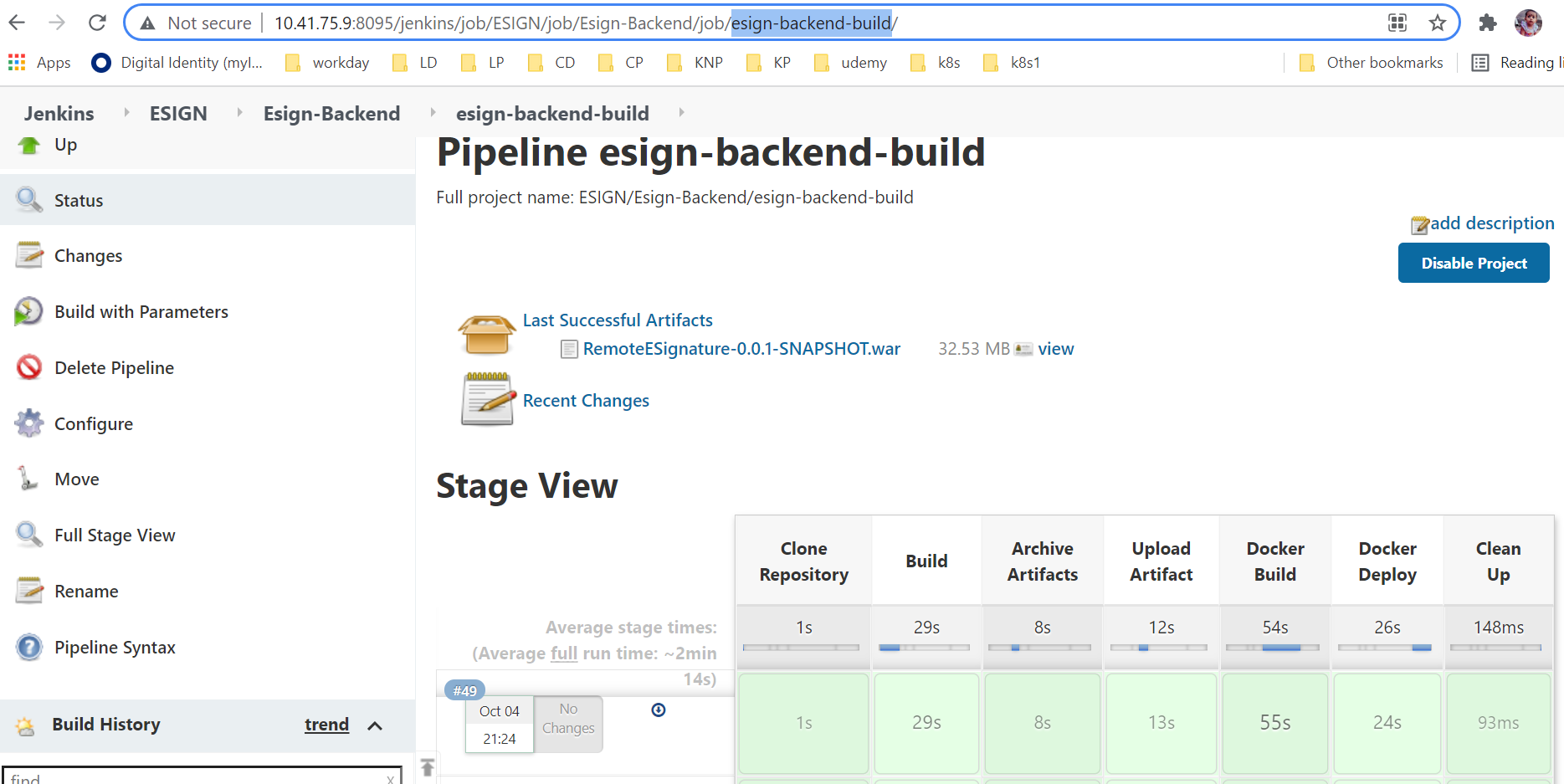
**Esign-Backend**

Below jobs available in ESIGN (folder)🡪Esign-Backend (folder)

1. esign-backend-build
2. esign- backend -Docker-Build
3. esign- backend -deploy

**1.esign-backend-build:**

Below stages available in this job



**2.esign- backend-Docker-Build**

Got to this below location cd /app/devops/dockerbuild/esign/esignbackend

removing existing service and downloading the particular service by using cmd and giving permission to the service.

Going to the esign-backend/k8s folder where we have Dockerfile.

we are building the docker image by using below command and to download the jar from artifactory( to download the particular branchname jar then we can login the artifactory that’s why we gave argumnets).

sudo docker build --no-cache -t arty.zurich.co.uk:5003/esign/latest/esign-backend:$branchname --build-arg BUILDVERSION=$branchname --build-arg ARFT\_UID=$ARFTUID --build-arg ARFT\_PWD=$ARFTPWD --build-arg frontendurl=$frontendurl .

Once the image created we are tagging with different name as well because buckup purpose.

we are pushing the image to build server

removing image from build server

**Dockerfile:**

We created custom image alpineopenjdk:11.0.9.

we are taking input as artifactory id and password and build version and frontendurl.

we created user steel and adding that user with id

Port in dockerfile should match with server port inside src🡪main🡪resources🡪application.properties from Bitbucket where the repo of the code is available. If port not available in application.properties then

We will use default port is 8080 and expose the port and setting the user as well.

RUN curl "http://10.41.75.8:8061/artifactory/esign-backend/latest/RemoteESignature-0.0.1-SNAPSHOT.war" -o RemoteESignature-0.0.1-SNAPSHOT.war

We are using this Artifactory: <http://10.41.75.8:8061/artifactory/>

Downloading the war with latest version and placing this location WORKDIR /app/services from artifactory by using above command

RUN echo "java -Duser.timezone=UTC -Dfile.encoding=UTF-8 -Dspring-boot.run.fork=false -Dprocess.expirationdays=5 -Dzurich.weblife.remoteEsignature.frontendBaseUrl="${frontendurl}" -jar RemoteESignature-0.0.1-SNAPSHOT.war" > /app/services/start.sh

To run the war file we need to pass above arguments and will put it in /app/services/start.sh

Start.sh will be the entry point to build the image.

**3.esign-backend-deploy:**

Same as explained **esign-frontend-deploy**