

SUSE Rancher 2.6

Edge Analytics with SUSE Rancher: Healthcare - Cancer Prediction

Getting Started

Abhinav Sharma, GSoC Contributor

This document provides an introduction to edge analytics in healthcare with SUSE Rancher through the deployment of Cancer Prediction System, an open source, machine learning solution designed to aid medical personnel interpret CT medical images.

This project was part of the Google Summer of Code, a global online mentoring program focused on introducing new contributors to open source software development.

Disclaimer: Documents published as part of the series SUSE Technical Reference Documentation have been contributed voluntarily by SUSE employees and third parties. They are meant to serve as examples of how particular actions can be performed. They have been compiled with utmost attention to detail. However, this does not guarantee complete accuracy. SUSE cannot verify that actions described in these documents do what is claimed or whether actions described have unintended consequences. SUSE LLC, its affiliates, the authors, and the translators may not be held liable for possible errors or the consequences thereof.

Publication Date: 2022-09-11

Contents

- 1 Introduction 3
- 2 Technical overview 4
- 3 Prerequisites 7
- 4 Install Rancher 8
- 5 Install Longhorn 8
- 6 Install Kubeflow Pipelines 11
- 7 Download the Data 13
- 8 Flask Interface 16
- 9 Machine Learning Model 18
- 10 Connect as an external client 25
- 11 Demonstration 26
- 12 Summary 27
- 13 Legal notice 28
- 14 GNU Free Documentation License 29

Cancer is world's second-leading cause of death in the world. It results in development of abnormal cells that divide uncontrollably and have the ability to infiltrate and destroy normal body tissue. Eventually, survival rates are improving and all thanks to cancer screening, treatment and prevention.

Machine Learning is one of the aspect which can be integrated with the modern science and can create wonders. Under the Google Summer of Code, we at SUSE organization have created a Machine larning based Cancer Prediciton Model for early screening.

This Document will demonstrate the detailed approach undertaken to accomplish this project. The project is developed under mentorship of Bryan Gartner, Ann Davis, Brian Fromme and the SUSE organization.

1.2 Scope

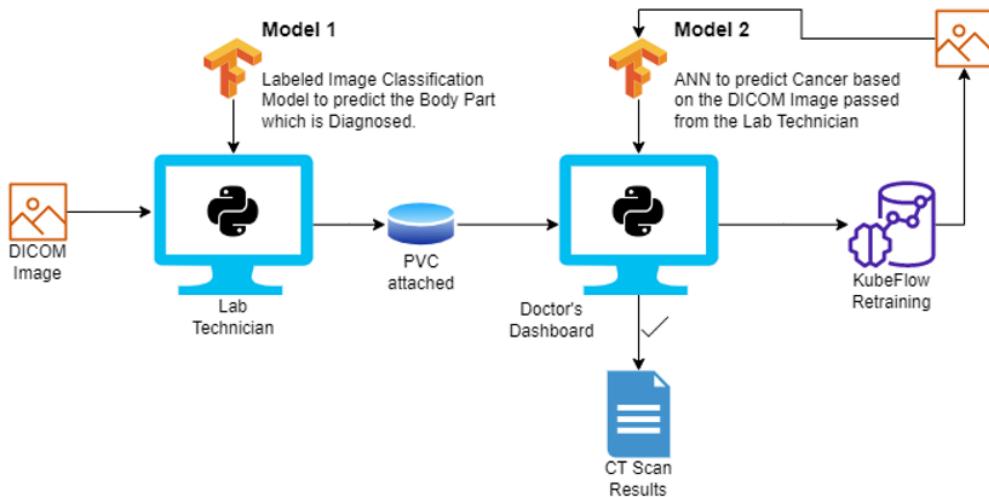
This guide introduces the basic concepts and steps to install, configure, and use the Cancer Prediction System in a SUSE Rancher Kubernetes environment. It uses Convolutional Neural Nets (CNNs) to automatically recognize the complex patterns in imaging data, providing quantitative as well as qualitative assessments of data within a short period of time.

1.3 Audience

This document is intended for Hospital Administrators, Radiologists, Doctors, Data Scientists, Data Engineers, and Data Analysts who are interested in using the Cancer Prediction System in a SUSE Rancher Kubernetes environment. It is also intended for SUSE Rancher users who want to learn how to install and configure the Cancer Prediction System.

2 Technical overview

As a part of GSOC, We have built a Machine Learning Based Cancer Prediciton System. The fundamental goal of the system is the prediciton of Cancer Susceptibility (also known as risk assessment), in this case, we are trying to predict the likelihood of developing a cancer prior to occurence of the disease.



The Cancer Prediction System uses CT Scanned Images in the form of **DICOM** (Digital Imaging and Communications in Medicine) (<https://www.dicomstandard.org/>). The system is designed to be deployed as a Microservice based architecture and is divided into four interfaces:

- Lab Technician Dashboard
- Doctor Dashboard
- Rancher Dashboard
- ML Pipeline Dashboard

2.1 Components and tools

Key Cancer Prediction System components discussed in this guide are:

Flask Application

Flask is a micro web framework also written in Python. It is a lightweight framework used to create web applications easily. We are using flask to serve the Backend and Machine Learning models as APIs.

Kubernetes Architecture

Kubernetes is a portable, extensible, open source platform for managing containerized workloads and services, that facilitates both declarative configuration and automation. We have a kubernetes manifests designed to set up application over the cluster. Our application is compatible with various kubernetes distributions. We have tested our application on SUSE Rancher.

Keras

Keras is an open-source software library that provides a Python interface for designing artificial neural networks. It acts as an interface for TensorFlow library. We are using keras to design [Convolutional Neural Nets \(CNNs\)](https://www.tensorflow.org/tutorials/images/cnn) (<https://www.tensorflow.org/tutorials/images/cnn>) ↗ for our application.

Kubeflow Pipelines

It is used for machine learning pipelines to orchestrate workflows running on our Kubernetes clusters. Kubeflow allows us to focus on writing ML algorithms instead of managing their operations.

Some additional components and tools discussed in this guide include:

- [Longhorn](https://longhorn.io/) (<https://longhorn.io/>) ↗: the cloud native distributed block storage system for Kubernetes.
- [Rancher](https://rancher.com/) (<https://rancher.com/>) ↗: the Kubernetes management platform.
- [kubectl](https://kubernetes.io/docs/reference/kubectl/) (<https://kubernetes.io/docs/reference/kubectl/>) ↗: the command line tool for communicating with the Kubernetes cluster's control plane via the Kubernetes API.
- [kustomize](https://github.com/kubernetes-sigs/kustomize) (<https://github.com/kubernetes-sigs/kustomize>) ↗: a stand-alone tool for programmatically customizing Kubernetes objects.
- [Docker](https://www.docker.com/) (<https://www.docker.com/>) ↗: the container runtime used to build and run the application.

2.2 Process overview

Getting started with Cancer Prediction System is fairly easy. In general, the process is as follows:

1. Clone the project repository:

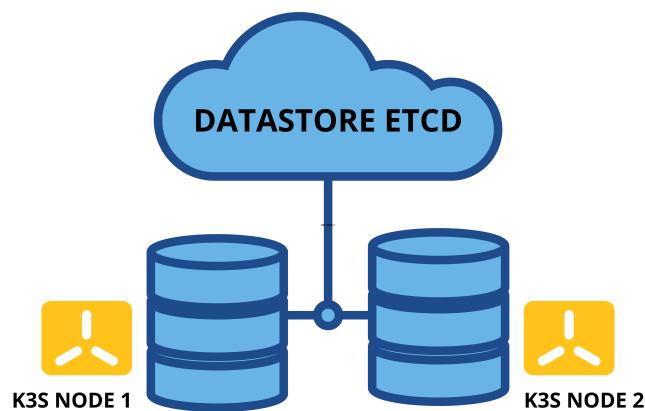
```
git clone https://github.com/abhi-bhatra/ct_image_scanning.git
```

2. Log in to your SUSE Rancher environment and select a managed cluster
3. Apply kubernetes manifests to setup the environment (*namespaces and storage*)
4. Apply kubernetes manifests to deploy the application
5. Access the application via the Rancher Dashboard

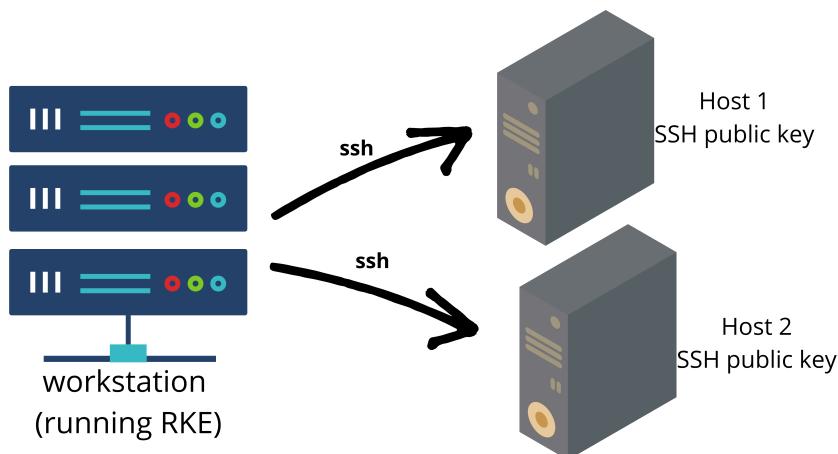
3 Prerequisites

This guide assumes that you have access to an existing Kubernetes cluster managed by SUSE Rancher (<https://rancher.com/docs/rancher/v2.6/en/overview/>). A good place to start getting more details is the Deploying cluster Using Rancher Kubernetes Engine or K3S (<https://medium.com/@abhinavsharma332/deploy-single-node-cluster-using-k3s-or-rke-6fc9e6a38b66>) blog.

3.1 K3S Architecture



3.2 RKE Architecture



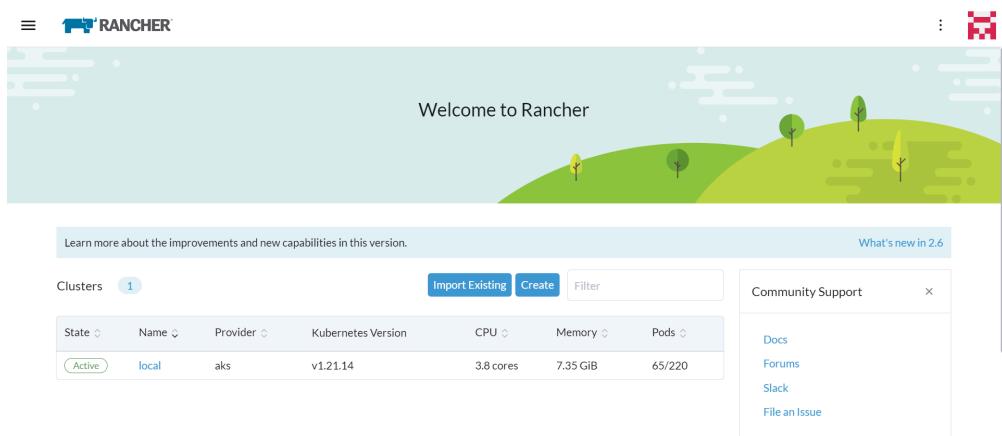


Tip

Be sure docker is installed, as RKE (*Rancher Kubernetes Engine*) clusters need Docker as a prerequisite. Visit Docker website for more information (<https://www.docker.com/get-started/>).

4 Install Rancher

There are various ways to install the Rancher on the existing cluster. This section provides an overview of installing options of Rancher (<https://docs.ranchermanager.rancher.io/v2.5/pages-for-subheaders/installation-and-upgrade>).



Tip

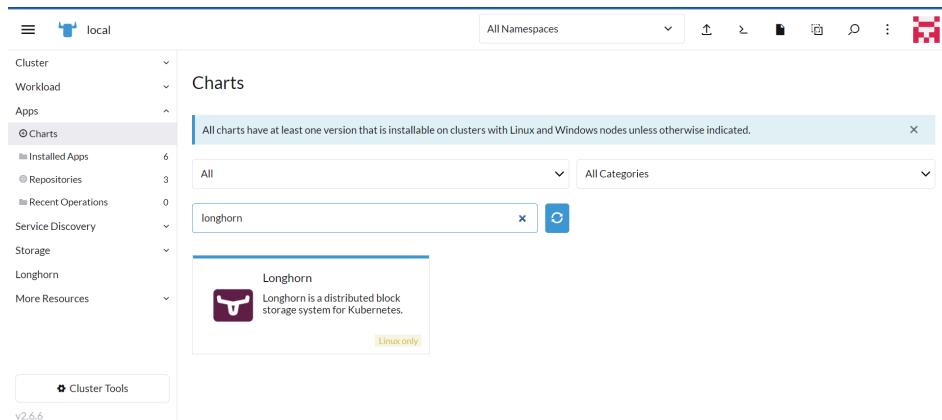
For this guide, we will be using the High-availability Kubernetes Install with the Helm CLI (<https://docs.ranchermanager.rancher.io/v2.5/pages-for-subheaders/install-upgrade-on-a-kubernetes-cluster>) to install Rancher on top of the cluster.

5 Install Longhorn

Before proceeding, review the concepts of persistent volumes, persistent volume claims (PVCs), and storage classes in the Rancher documentation (<https://rancher.com/docs/rancher/v2.6/en/cluster-admin/volumes-and-storage/how-storage-works/>).

Longhorn provides a storage solution that is easy to use, self-healing, and highly available. Longhorn is designed to be deployed on a Kubernetes cluster and provides persistent storage for stateful applications. There are 3 ways to installing Longhorn to Clusters:

1. Using the Apps and Marketplace in Rancher UI (<https://longhorn.io/docs/1.3.1/deploy/install/install-with-rancher/>) ↗



The screenshot shows the Rancher UI interface. On the left, there's a sidebar with a navigation menu: Cluster, Workload, Apps, Charts, Installed Apps (6), Repositories (3), Recent Operations (0), Service Discovery, Storage, Longhorn, and More Resources. The 'Charts' section is selected. In the main area, there's a search bar with 'longhorn' typed into it. Below the search bar, a card for 'Longhorn' is displayed, which is described as 'Longhorn is a distributed block storage system for Kubernetes.' A note at the bottom right of the card says '(Linux only)'. At the bottom left of the interface, there's a 'Cluster Tools' button.

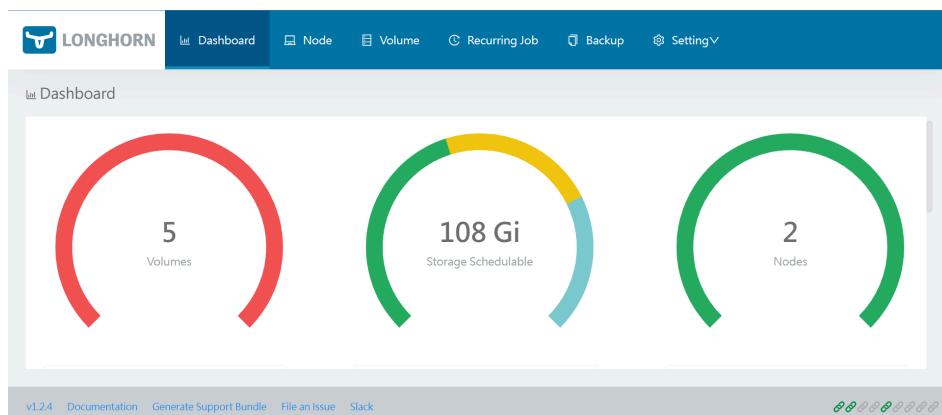
2. Using the kubectl manifests files (<https://longhorn.io/docs/1.3.1/deploy/install/install-with-kubectl/>) ↗

```
kubectl apply -f https://raw.githubusercontent.com/longhorn/longhorn/v1.2.4/deploy/longhorn.yaml
```

3. Using the Helm (<https://longhorn.io/docs/1.3.1/deploy/install/install-with-helm/>) ↗:

```
helm repo add longhorn https://charts.longhorn.io
helm repo update
helm install longhorn/longhorn -name longhorn -namespace longhorn-system
```

4. After successfull installation, you can see the Longhorn UI in the Rancher Dashboard.



The screenshot shows the Longhorn UI dashboard. At the top, there's a header with the Longhorn logo and tabs for Dashboard, Node, Volume, Recurring Job, Backup, and Setting. Below the header, there's a title 'Dashboard'. The dashboard features three large circular progress bars: one red bar labeled '5 Volumes', one green bar labeled '108 Gi Storage Schedulable', and one green bar labeled '2 Nodes'. At the bottom of the dashboard, there's a footer with links: v1.2.4, Documentation, Generate Support Bundle, File an Issue, Slack, and a social media icon row.



Tip

Cancer Prediction System strongly recommends using atleast two-worker nodes cluster.

5.1 Provision local storage

At this point, you can use Longhorn to format and manage the drives in your cluster nodes.

1. Provision the storage using the Flask Applications to share data and information on local cluster

```
apiVersion: v1
kind: PersistentVolumeClaim
metadata:
  name: data-pvc
  namespace: concerns
spec:
  accessModes:
    - ReadWriteMany
  resources:
    requests:
      storage: 1Gi
  storageClassName: longhorn
```

2. Create a PersistentVolumeClaim (PVC) for the data volume

```
apiVersion: v1
kind: PersistentVolumeClaim
metadata:
  name: ds-pvc
  namespace: concerns
spec:
  accessModes:
    - ReadWriteMany
  resources:
    requests:
      storage: 1Gi
  storageClassName: longhorn
```

3. You can visualize the Persistent Volumes in the Rancher Dashboard.

<input type="checkbox"/> State	Name	Reclaim Policy	Persistent Volume Claim	Source	Reason	Age
<input type="checkbox"/> (Bound)	pvc-5060c8dc-052b-4bdf-931a-d1b555cf6dfa	Delete	mysql-pv-claim	Longhorn	3.6 days	:
<input type="checkbox"/> (Bound)	pvc-b8c76ee0-6c9d-49fc-b9f5-08e51547e2e4	Delete	data-pvc	Longhorn	6 days	:
<input type="checkbox"/> (Bound)	pvc-c213fa97-fb23-44ed-af03-3cec26fa15e7	Delete	ds-pvc	Longhorn	4.6 days	:
<input type="checkbox"/> (Bound)	pvc-d327ab73-c5f8-4951-88a5-1844ee027734	Delete	minio-pvc	Longhorn	3.6 days	:
<input type="checkbox"/> (Bound)	pvc-f029e569-7368-4803-ac9e-622ff76edb94	Delete	my-longhorn-pvc	Longhorn	29 days	:

6 Install Kubeflow Pipelines

The Kubeflow project provides a straightforward way to deploy best-of-breed open-source systems for ML to diverse infrastructures. Anywhere you are running Kubernetes, you should be able to run Kubeflow. The [Kubeflow documentation \(\)](https://www.kubeflow.org/docs/started/k8s/) provides a detailed guide to installing Kubeflow on Kubernetes.

We will look at how to deploy Kubeflow Pipelines standalone on our local clusters:

1. Set the PIPELINE_VERSION and apply the Kubeflow Pipelines manifest:

```
export PIPELINE_VERSION=1.8.3

kubectl apply -k "github.com/kubeflow/pipelines/manifests/kustomize/cluster-scoped-
resources?ref=$PIPELINE_VERSION"

kubectl wait --for condition=established --timeout=60s crd/applications.app.k8s.io

kubectl apply -k "github.com/kubeflow/pipelines/manifests/kustomize/env/platform-
agnostic-pns?ref=$PIPELINE_VERSION"
```



Note

It will take 15 to 20 mins to deploy the Kubeflow Pipelines on your cluster. You can check the status using `kubectl get all -n kubeflow`

- Once all the services will start, you can see all pods status 1/1 Running. Your output will be somewhat similar to this:

NAME	READY	STATUS
RESTARTS	AGE	
pod/workflow-controller-5667759dd7-fbgrp	1/1	Running
2d3h		0
pod/ml-pipeline-scheduledworkflow-7f8bc78db9-qpx4f	1/1	Running
2d3h		0
pod/ml-pipeline-viewer-crd-8497d9695c-tqmdg	1/1	Running
2d3h		0
pod/ml-pipeline-ui-69bc756bd7-nmzm6	1/1	Running
2d3h		0
pod/metadata-envoy-deployment-6df8bdd989-lc77p	1/1	Running
2d3h		0
pod/minio-5b65df66c9-qt6lk	1/1	Running
2d3h		0
pod/ml-pipeline-persistenceagent-585c4b58d6-mcmctx	1/1	Running
2d3h		1
pod/ml-pipeline-7cc4f8fdf7-b2vjp	1/1	Running
2d3h		2
pod/cache-server-6cddb849-bnd6n	1/1	Running
2d3h		1

- To access the Kubeflow Pipelines UI, you can use the following command:

```
kubectl port-forward -n kubeflow svc/ml-pipeline-ui 8080:80
```

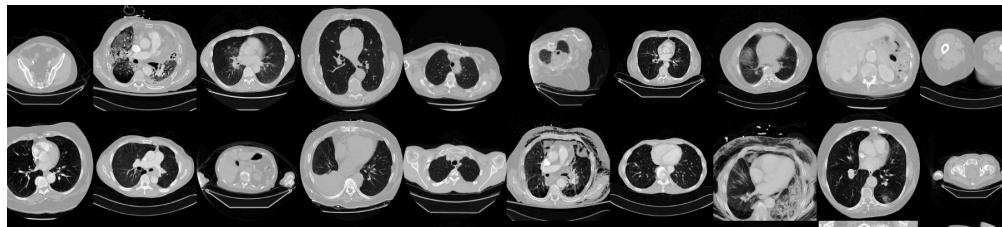
- Click on the pipeline port from the Rancher portal and you will be redirected to the Kubeflow Pipelines UI dashboard.

The screenshot shows the Kubernetes dashboard interface. On the left, there's a sidebar with navigation links like Cluster, Workload, Apps, Service Discovery, Storage, Longhorn, and More Resources. Below the sidebar, it says 'v2.6.6'. The main area lists services under the 'Services' section. One service, 'ml-pipeline-ui', is highlighted with a red box. Other listed services include 'metadata-grpc-service', 'minio-service', 'ml-pipeline', 'ml-pipeline-visualizationserver', and 'mysql'. Each service entry includes its name, status (Active), port information, labels, and creation time.

5. You can see the Kubeflow Pipelines UI dashboard.

The screenshot shows the Kubeflow Pipelines UI dashboard. The left sidebar has links for Home, Notebooks, Tensorboards, Models, Volumes, Experiments (AutoML), Experiments (KFP), Pipelines, Runs, Recurring Runs, and Artifacts. It also mentions 'Privacy + Usage Reporting' and 'build version dev_local'. The main dashboard has tabs for 'Dashboard' (selected) and 'Activity'. Under 'Recent Notebooks', there are two entries: 'kale.log' and 'lost+found'. Under 'Recent Pipelines', there are several entries: 'open-vaccine-model', '[Tutorial] DSL - Control structures', '[Tutorial] Data passing in python components', '[Demo] TFX - Taxi tip prediction model trainer', and '[Demo] XGBoost - Iterative model training'. To the right, there's a 'Documentation' section with links to 'Getting Started with Kubeflow', 'Minikube', 'MicrOKBs for Kubeflow', 'Minikube for Kubeflow', 'Kubeflow on GCP', 'Kubeflow on AWS', and 'Requirements for Kubeflow'.

7 Download the Data



The Cancer Prediction System uses the [CT Medical Images ~ cancer imaging archive with contrast and patient age](https://www.kaggle.com/datasets/kmader/siim-medical-images) (<https://www.kaggle.com/datasets/kmader/siim-medical-images>) dataset from Kaggle.

The dataset is designed to allow for different methods to be tested for examining the trends in CT image data associated with using contrast and patient age. The basic idea is to identify image textures, statistical patterns and features correlating strongly with these traits and possibly build simple tools for automatically classifying these images when they have been misclassified (or finding outliers which could be suspicious cases, bad measurements, or poorly calibrated machines).

Cancer prediction system dataset directory system:

```
/dataset
-- archive/
  -- dicom_dir/
  .
  ID_0001_AGE_0069_CONTRAST_1_CT.dcm
  .
  -- tiff_images/
  .
  ID_0000_AGE_0060_CONTRAST_1_CT.tif
  .
  -- full_archive.npz
  -- overview.csv

-- dataset-classification
  -- Chest-CT/
  -- NonChest-CT/

-- dataset-prediction/
  -- train/
    -- cancer/
    -- non-cancer/
  -- test/
    -- cancer/
    -- non-cancer/
  -- validation/
    -- cancer/
    -- non-cancer/
```

- archive:** This folder comprises of raw dataset downloaded from Kaggle. We use python notebooks to process the data for further used in Machine Learning model.
- dataset-classification:** This is a separate dataset which separates all the DICOM Images as Chest and Non Chest. Currently, our model support Cancer classification on Chest DICOM Images. So, we need to filter our the Non Chest DICOM Images.
- dataset-predictiton:** This is the final dataset used in Machine Learning model. All the ras images are processed into Train, Test and Validation sets. The labels are attached to the DICOM, so images can be classified as Cancer and Non-Cancer Images.

Apply the dataset deployment manifests to download the dataset from storage and process the data for further use in Machine Learning model.

```

apiVersion: apps/v1
kind: Deployment
metadata:
  name: datasetvm
  namespace: concerns
spec:
  replicas: 1
  selector:
    matchLabels:
      app: datasetvm
  template:
    metadata:
      labels:
        app: datasetvm
    spec:
      containers:
        - name: datasetvm
          image: "ubuntu:latest"
          imagePullPolicy: Always
          volumeMounts:
            - name: dataset
              mountPath: /dataset
      env:
        - name: DATASET
          value: "https://rancherdataset.blob.core.windows.net/ct-images/dataset.zip"
          command: ["/bin/sh", "-c"]
          args: ["apt-get update; apt-get install unzip wget -y; wget $DATASET -O /dataset/dataset.zip; unzip /dataset/dataset.zip -d /dataset/dataset; ls -l /dataset/dataset"]
      volumes:
        - name: dataset
      persistentVolumeClaim:

```

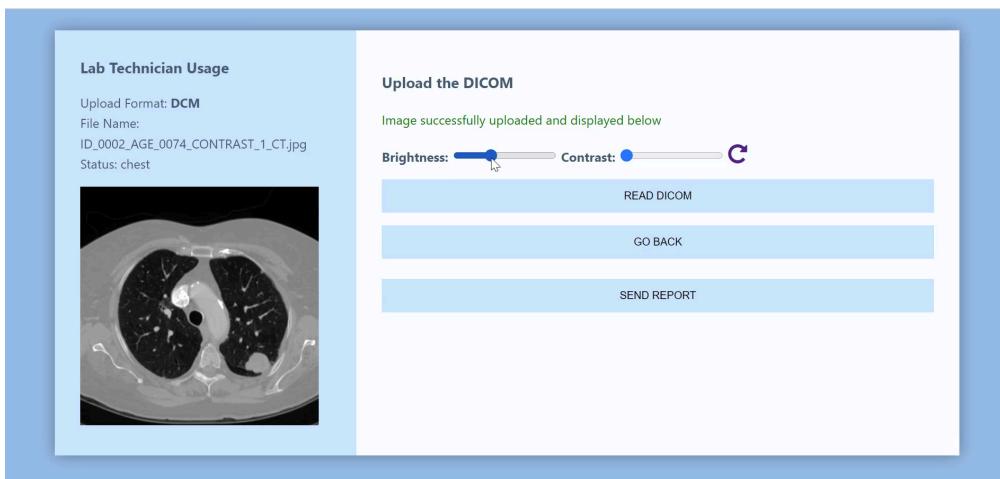
8 Flask Interface

Complete Cancer Prediction System is built on top of Flask. It has two separate applications for the doctor and the radiologist. Directory Structure of the application is as follows:

```
/application
-- doctor_app/
  -- app.py
  -- Dockerfile
  -- classification-model.h5
  -- prediction-model.h5
  -- requirements.txt
  -- static/
    -- styles/
      -- css/
      -- js/
  -- template/
    -- base.html
    -- gallery.html
    -- predict.html
    -- retrain.html
    -- upload.html

-- lab_tech/
  -- app.py
  -- Dockerfile
  -- classification-model.h5
  -- adjust.py
  -- requirements.txt
  -- static/
    -- styles/
      -- css/
      -- js/
  -- template/
    -- base.html
    -- predict.html
    -- send.html
    -- upload.html
```

8.1 Deploy Lab Technician Interface



Lab Technician Interface is responsible for getting DICOM image as input. The person (*Radiologists, Lab Technician, Physicians*) could alter the information such as Contrast, Brightness and Angle of rotation of the DICOM image. They can also read all the information associated with the DICOM image (Modality: CT Scan).

Lab Technician Interface have a streamlined installation process. It can be easily installed through the Manifests.

1. Clone the repository

```
git clone https://github.com/abhi-bhatra/ct_image_scanning.git
```

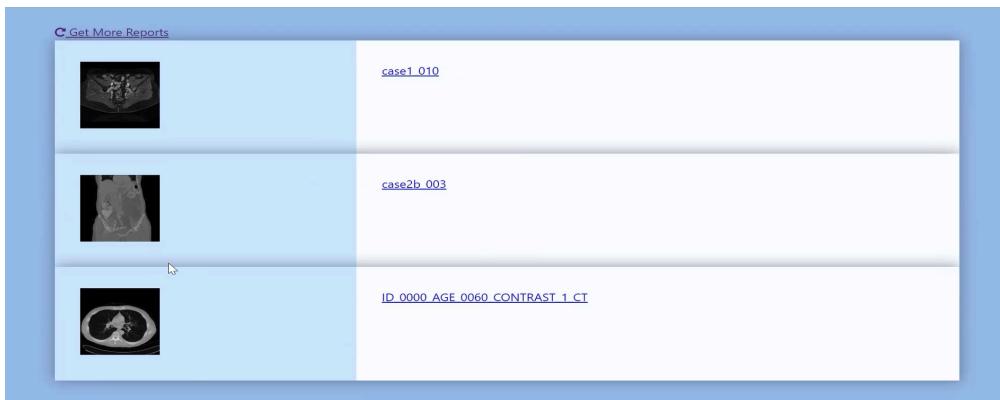
2. Change the directory to k8s/lab-tech

3. Apply the kustomization file to deploy the Lab Technician Interface

```
kubectl apply -k .
```

4. To access the Lab Technician Interface, go to Rancher UI and click on the lab-tech service. Click on the Port button. Click on the port displayed. You will be redirected to the Lab Technician Interface.

8.2 Deploy Doctor Dashboard Interface



Doctor Dashboard is designed for the doctors to examine the report send by the Lab Technician. It receives the report of a patient and displays it to the user, predicting whether or not person is suffering from cancer. If doctor will not be satisfied with the response, they can send the image for the retraining with the correct label attached to it.

Just like Lab Technician Interface, doctor's dashboard Interface installation process is streamline and can be installed through manifests.

1. Clone the repository

```
git clone https://github.com/abhi-bhatra/ct_image_scanning.git
```

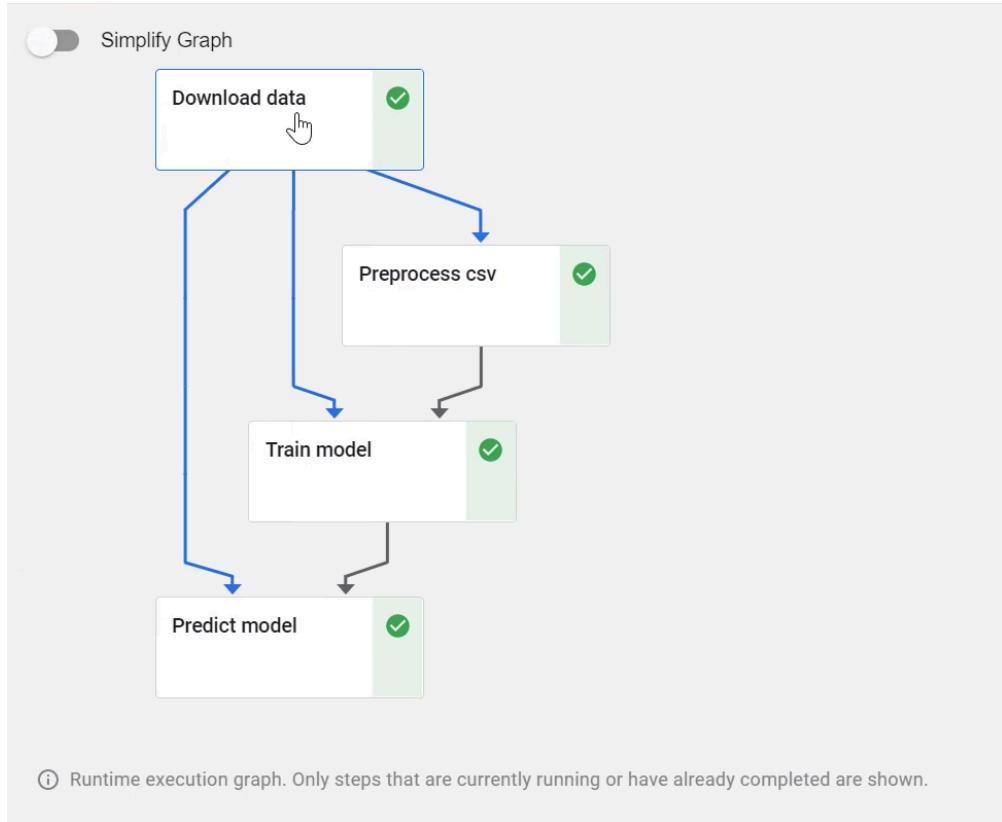
2. Change the directory to k8s/doctor-app/

3. Apply the kustomization file to deploy the Doctor Dashboard Interface

```
kubectl apply -k .
```

9 Machine Learning Model

In the project, Machine Learning is one of the major component used for prediction. We are using Convolutional Neural Network (CNN). It is a deep learning method and has achieved better results in detecting and segmenting specific objects in images in the last decade than conventional models such as regression, support vector machines or artificial neural networks.



9.1 Body-part Classification Model

Classification models are a subset of supervised machine learning. A classification model reads some input and generates an output that classifies the input into some category. In our case, model is taking CT-Scan and X-Ray images as input, and images are labelled. The model is a Supervised Learning technique that is used to identify the category of new observations on the basis of training data.

Complete Reference to the Classification Model can be found at: https://github.com/abhi-bhatra/ct_image_scanning/blob/master/machine-learning/ml-model/neural_net.ipynb

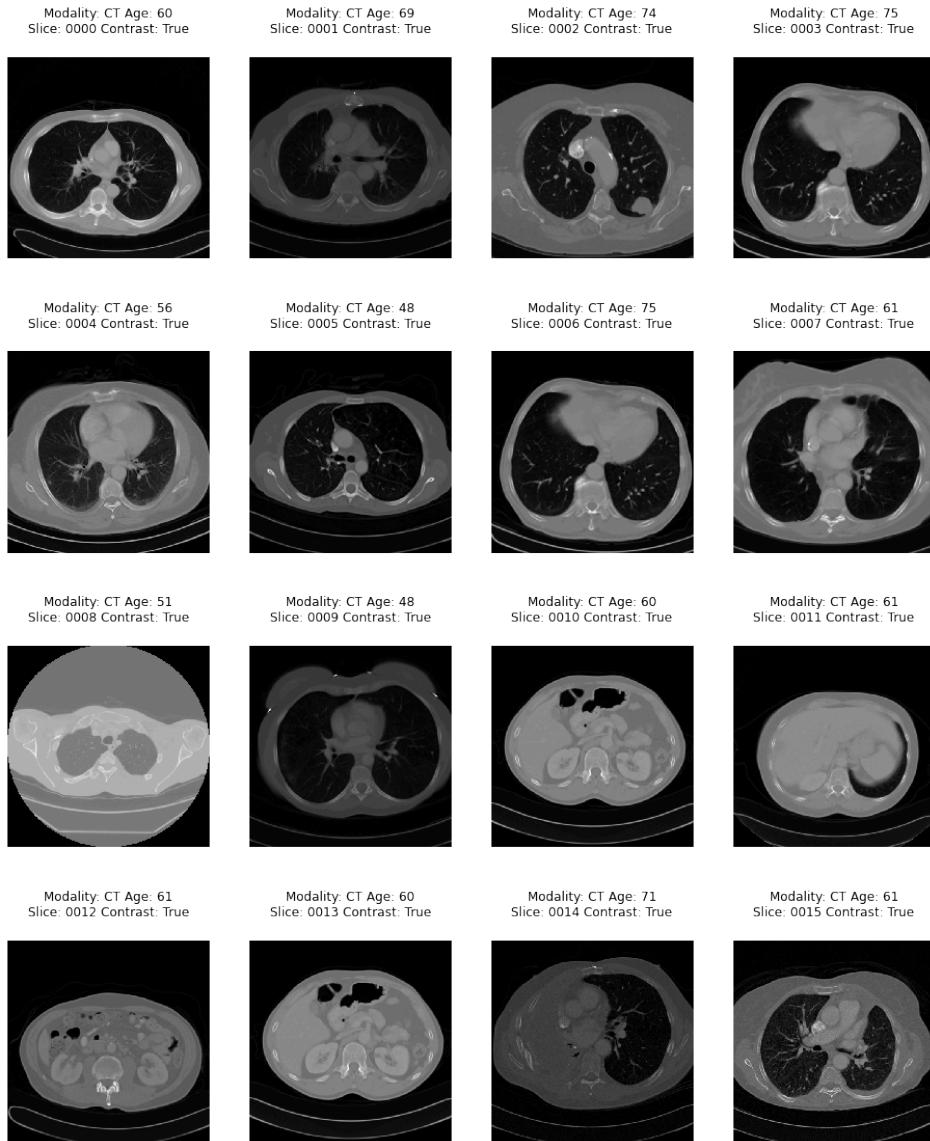
9.2 Cancer Prediction Model

In this model, we use machine learning in cancer diagnosis and detection. We are using Artificial neural networks (ANNs) for detecting and classifying tumors CRT images. We are using Convolutional Neural Network (CNN) for building the model (https://github.com/abhi-bhatra/ct_im-

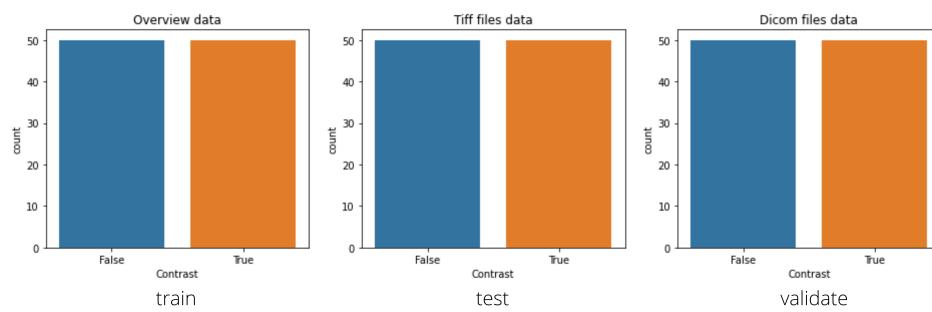
[age_scanning/blob/master/machine-learning/ml-model/cancer_detection.ipynb](#). We are using 3 convolutional layers and 2 dense layers. We are using RMSprop optimizer and binary cross-entropy loss function.

Steps for Prediction model are as follows:

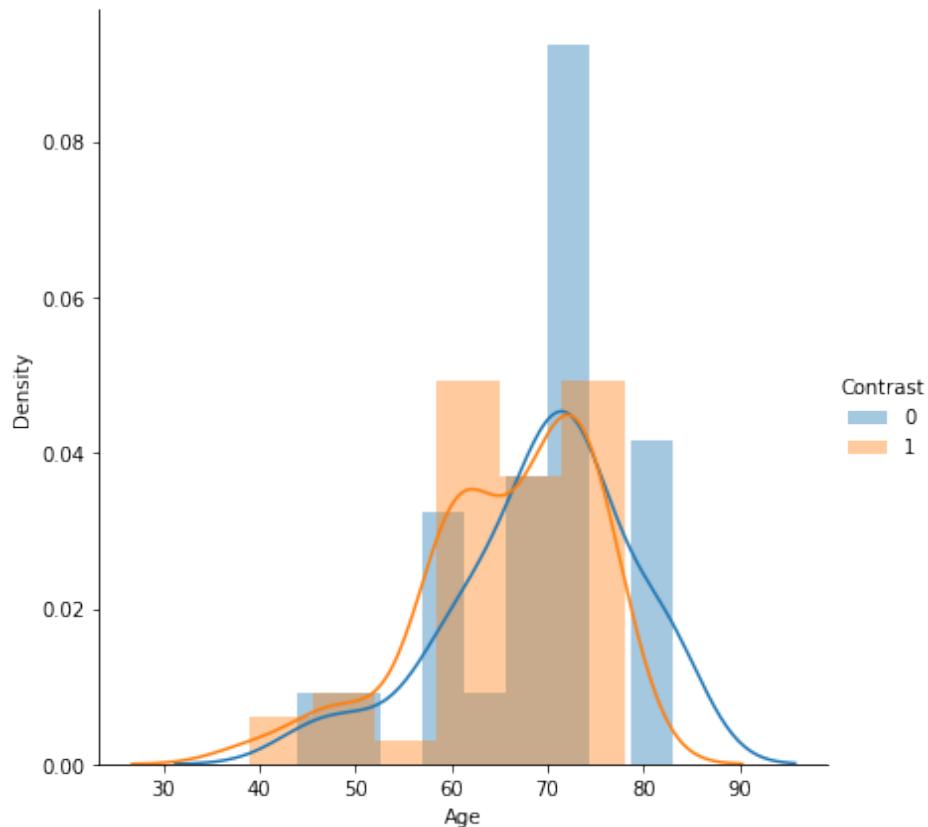
- 1. Preprocessing:** In this step, we are converting the images into grayscale and resizing them to 128x128 pixels. We are also converting the labels into one-hot encoding.



- 2. Splitting the dataset:** We are splitting the dataset into training, validation and testing sets. We are using 80% of the dataset for training, 10% for validation and 10% for testing.

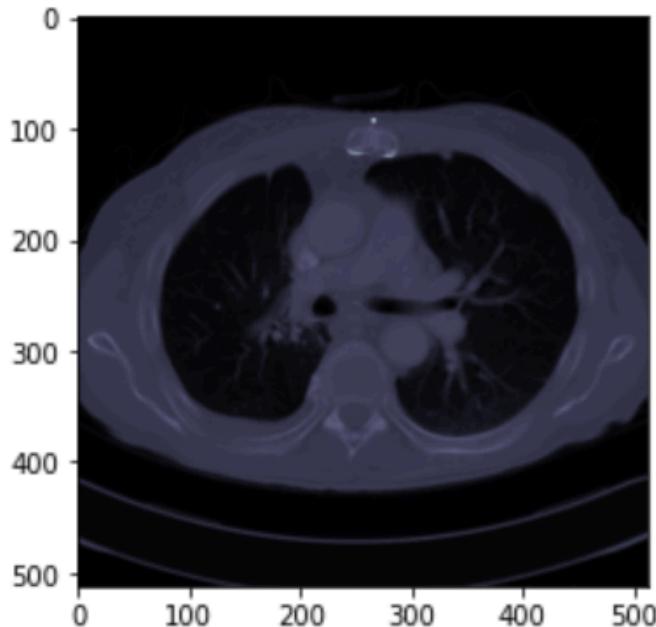


3. **Building the model:** We are using Convolutional Neural Network (CNN) for building the model. We are using 3 convolutional layers and 2 dense layers. We are using Adam optimizer and categorical crossentropy loss function.



4. **Training the model:** We are training the model for 200 epochs. We are using 32 as the batch size.

97.59% The patient has Cancer disease
<matplotlib.image.AxesImage at 0x26d1d32c8b0>

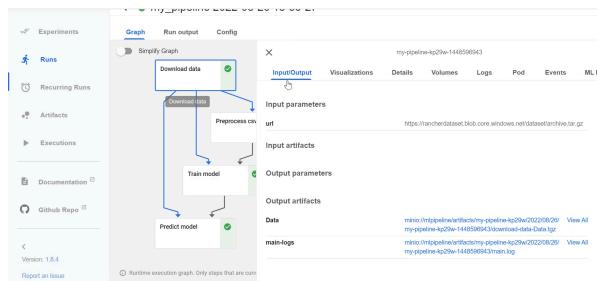


9.3 Retraining Model

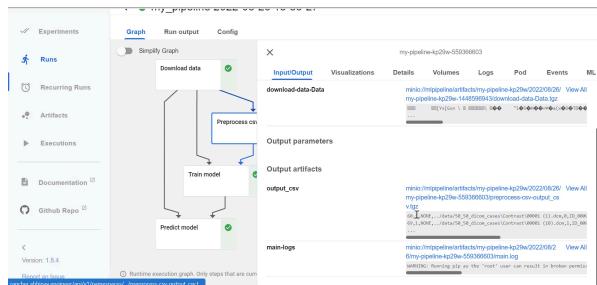
Kubeflow Pipelines is a platform for building and deploying portable, scalable machine learning (ML) workflows based on Docker containers.

Retraining the Machine Learning Model is an automated process. We have used Kubeflow Pipelines to automate the process of retraining the model. It comprises of the following steps:

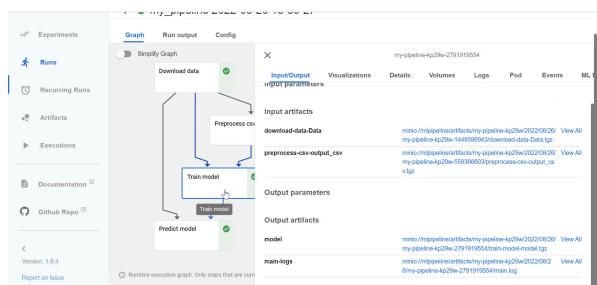
1. **Data Collection:** The data is collected from the dataset-prediction folder. The data is split into Train, Test and Validation sets. The labels are attached to the images. The data is then stored in the dataset folder.



2. Data Preprocessing: The data is preprocessed to remove the noise from the images. The images are converted into grayscale images. The images are then resized to 512x512. The data is then stored in the preprocessed folder.



3. Trained Model: The model is trained on the preprocessed data. The model is saved in the trained-model folder.



4. Prediction: The model is then converted into a prediction model. The prediction model is saved in the prediction-model folder.

Complete Reference to the Retraining Model can be found at: https://github.com/abhi-bhatra/ct_image_scanning/tree/master/machine-learning/kubeflow

9.4 Verify installation

After you have performed all of the above installation methods, validate that the Cancer Prediction System is installed and running.

1. List the pods running in the 'cancers' namespace.

```
kubectl get all --namespace cancers
```

This should produce output like the following:

NAME	READY	STATUS	RESTARTS	AGE
pod/datasetvm-5db64d7549-cflmf	1/1	Running	0	9s
pod/doctor-app-d5b856997-64gnt	1/1	Running	0	10s
pod/labtech-app-6c54f58874-jdmcw	1/1	Running	0	12s

NAME	TYPE	CLUSTER-IP	EXTERNAL-IP	PORT(S)	AGE
service/doctor-svc	NodePort	10.0.66.40	None	5002:30001/TCP	23s
service/labtech-svc	NodePort	10.0.145.57	None	5001:32009/TCP	24s

NAME	READY	UP-TO-DATE	AVAILABLE	AGE
deployment.apps/datasetvm	1/1	1	0	13s
deployment.apps/doctor-app	1/1	1	0	23s
deployment.apps/labtech-app	1/1	1	0	23s

NAME	DESIRED	CURRENT	READY	AGE
replicaset.apps/datasetvm-5db64d7549	1	1	1	13s
replicaset.apps/doctor-app-d5b856997	1	1	1	10s
replicaset.apps/labtech-app-6c54f58874	1	1	1	21s

2. Verify that you see doctor-app- and labtech-app- pods in the 'Running' state and the service/doctor-svc and service/labtech-svc assigned to NodePort addresses.
3. Open the Kubeflow Console.

You can temporarily forward traffic from the Kubeflow Pipeline Console service to your local machine by issuing:

```
kubectl port-forward -n kubeflow svc/ml-pipeline-ui 8080:80
```

This will produce output like the following:

```
Forwarding from 127.0.0.1:8080 -> 3000
Forwarding from [::1]:8080 -> 3000

Connect using: http://localhost:8080
```

4. To access the application in your Web browser, open the Rancher portal at <https://localhost>:



Note

If you are using a different port on Rancher, put :PORT_NUMBER with the port you are using.

10 Connect as an external client

By default, external applications cannot access the Cancer Prediction System.

With SUSE Rancher, you can set up either load balancers or ingress controllers to redirect service requests. Load balancers can only handle one IP address per service, while ingress works with one or more ingress controllers to dynamically route service requests.

It is recommended that you configure your cluster with an ingress.



Note

Ingress and ingress controllers residing in RKE-launched clusters are powered by [NGINX](https://www.nginx.com/) (<https://www.nginx.com/>) .

Below is a sample ingress resource for Cancer Prediction System:

```
apiVersion: networking.k8s.io/v1
kind: Ingress
metadata:
  name: ingress-minio
  namespace: concerns
  annotations:
    kubernetes.io/ingress.class: "nginx"
    ## Remove if using CA signed certificate
```

```

nginx.ingress.kubernetes.io/proxy-ssl-verify: "off"
nginx.ingress.kubernetes.io/backend-protocol: "HTTPS"
nginx.ingress.kubernetes.io/rewrite-target: /
nginx.ingress.kubernetes.io/proxy-body-size: "0"
nginx.ingress.kubernetes.io/server-snippet: |
  client_max_body_size 0;
nginx.ingress.kubernetes.io/configuration-snippet: |
  chunked_transfer_encoding off;

spec:
  tls:
    - hosts:
        - cancerpred.example.com
      secretName: cancerpred-tls
  rules:
    - host: cancerpred.example.com
      http:
        paths:
          - path: /doctor
            pathType: Prefix
            backend:
              service:
                name: doctor-svc
                port:
                  number: 443
          - path: /labtech
            pathType: Prefix
            backend:
              service:
                name: labtech-svc
                port:
                  number: 443

```

11 Demonstration

Below is a demonstration of the Cancer Prediction System.

<https://www.youtube.com/watch?v=Hy00j1eoQ1c>

12 Summary

In this guide, you have explored how Artificial Intelligence is used for Cancer detection. This application has a streamline process of deploying into your SUSE Rancher Kubernetes landscape for consumption by your cloud native applications through a consistent API across all infrastructure platforms.

Below are a few resources to help you continue your exploration of Cancer Prediction System and SUSE Rancher:

- SUSE Technical Reference Documentation: Kubernetes (<https://documentation.suse.com/trd/kubernetes/>) ↗
- SUSE & Rancher Community (<https://www.suse.com/community/>) ↗
- Preparing for the next wave of transformation (<https://www.suse.com/c/preparing-for-the-next-wave-of-transformation/>) ↗
- Analytical Edge Ecosystem Workload for Healthcare Space (https://github.com/abhi-bhatra/ct_image_scanning/tree/master) ↗
- Getting Started with Sample workload over Rancher (<https://medium.com/@abhinavsharma332/deploying-wordpress-over-rancher-cb9539b1d7da>) ↗
- Empowering Machine Learning Application on Rancher (<https://medium.com/@abhinavsharma332/empowering-machine-learning-applications-on-rancher-f4e368a9009>) ↗
- Orchestrate Machine Learning Model with Kubeflow (<https://medium.com/@abhinavsharma332/orchestrate-machine-learning-model-with-kubeflow-11945e7801b5>) ↗

13 Legal notice

Copyright © 2006–2021 SUSE LLC and contributors. All rights reserved.

Permission is granted to copy, distribute and/or modify this document under the terms of the GNU Free Documentation License, Version 1.2 or (at your option) version 1.3; with the Invariant Section being this copyright notice and license. A copy of the license version 1.2 is included in the section entitled "GNU Free Documentation License".

SUSE, the SUSE logo and YaST are registered trademarks of SUSE LLC in the United States and other countries. For SUSE trademarks, see <https://www.suse.com/company/legal/>.

Linux is a registered trademark of Linus Torvalds. All other names or trademarks mentioned in this document may be trademarks or registered trademarks of their respective owners.

Documents published as part of the series SUSE Technical Reference Documentation have been contributed voluntarily by SUSE employees and third parties. They are meant to serve as examples of how particular actions can be performed. They have been compiled with utmost attention to detail. However, this does not guarantee complete accuracy. SUSE cannot verify that actions described in these documents do what is claimed or whether actions described have unintended consequences. SUSE LLC, its affiliates, the authors, and the translators may not be held liable for possible errors or the consequences thereof.

14 GNU Free Documentation License

Copyright © 2000, 2001, 2002 Free Software Foundation, Inc. 51 Franklin St, Fifth Floor, Boston, MA 02110-1301 USA. Everyone is permitted to copy and distribute verbatim copies of this license document, but changing it is not allowed.

0. PREAMBLE

The purpose of this License is to make a manual, textbook, or other functional and useful document "free" in the sense of freedom: to assure everyone the effective freedom to copy and redistribute it, with or without modifying it, either commercially or noncommercially. Secondarily, this License preserves for the author and publisher a way to get credit for their work, while not being considered responsible for modifications made by others.

This License is a kind of "copyleft", which means that derivative works of the document must themselves be free in the same sense. It complements the GNU General Public License, which is a copyleft license designed for free software.

We have designed this License in order to use it for manuals for free software, because free software needs free documentation: a free program should come with manuals providing the same freedoms that the software does. But this License is not limited to software manuals; it can be used for any textual work, regardless of subject matter or whether it is published as a printed book. We recommend this License principally for works whose purpose is instruction or reference.

1. APPLICABILITY AND DEFINITIONS

This License applies to any manual or other work, in any medium, that contains a notice placed by the copyright holder saying it can be distributed under the terms of this License. Such a notice grants a world-wide, royalty-free license, unlimited in duration, to use that work under the conditions stated herein. The "Document", below, refers to any such manual or work. Any member of the public is a licensee, and is addressed as "you". You accept the license if you copy, modify or distribute the work in a way requiring permission under copyright law.

A "Modified Version" of the Document means any work containing the Document or a portion of it, either copied verbatim, or with modifications and/or translated into another language.

A "Secondary Section" is a named appendix or a front-matter section of the Document that deals exclusively with the relationship of the publishers or authors of the Document to the Document's overall subject (or to related matters) and contains nothing that could fall directly within that overall subject. (Thus, if the Document is in part a textbook of mathematics, a Secondary Section may not explain any mathematics.) The relationship could be a matter of historical connection with the subject or with related matters, or of legal, commercial, philosophical, ethical or political position regarding them.

The "Invariant Sections" are certain Secondary Sections whose titles are designated, as being those of Invariant Sections, in the notice that says that the Document is released under this License. If a section does not fit the above definition of Secondary then it is not allowed to be designated as Invariant. The Document may contain zero Invariant Sections. If the Document does not identify any Invariant Sections then there are none.

The "Cover Texts" are certain short passages of text that are listed, as Front-Cover Texts or Back-Cover Texts, in the notice that says that the Document is released under this License. A Front-Cover Text may be at most 5 words, and a Back-Cover Text may be at most 25 words.

A "Transparent" copy of the Document means a machine-readable copy, represented in a format whose specification is available to the general public, that is suitable for revising the document straightforwardly with generic text editors or (for images composed of pixels) generic paint programs or (for drawings) some widely available drawing editor, and that is suitable for input to text formatters or for automatic translation to a variety of formats suitable for input to text formatters. A copy made in an otherwise Transparent file format whose markup, or absence of markup, has been arranged to thwart or discourage subsequent modification by readers is not Transparent. An image format is not Transparent if used for any substantial amount of text. A copy that is not "Transparent" is called "Opaque".

Examples of suitable formats for Transparent copies include plain ASCII without markup, Texinfo input format, LaTeX input format, SGML or XML using a publicly available DTD, and standard-conforming simple HTML, PostScript or PDF designed for human modification. Examples of transparent image formats include PNG, XCF and JPG. Opaque formats include proprietary formats that can be read and edited only by proprietary word processors, SGML or XML for which the DTD and/or processing tools are not generally available, and the machine-generated HTML, PostScript or PDF produced by some word processors for output purposes only.

The "Title Page" means, for a printed book, the title page itself, plus such following pages as are needed to hold, legibly, the material this License requires to appear in the title page. For works in formats which do not have any title page as such, "Title Page" means the text near the most prominent appearance of the work's title, preceding the beginning of the body of the text.

A section "Entitled XYZ" means a named subunit of the Document whose title either is precisely XYZ or contains XYZ in parentheses following text that translates XYZ in another language. (Here XYZ stands for a specific section name mentioned below, such as "Acknowledgements", "Dedications", "Endorsements", or "History".) To "Preserve the Title" of such a section when you modify the Document means that it remains a section "Entitled XYZ" according to this definition. The Document may include Warranty Disclaimers next to the notice which states that this License applies to the Document. These Warranty Disclaimers are considered to be included by reference in this License, but only as regards disclaiming warranties: any other implication that these Warranty Disclaimers may have is void and has no effect on the meaning of this License.

2. VERBATIM COPYING

You may copy and distribute the Document in any medium, either commercially or noncommercially, provided that this License, the copyright notices, and the license notice saying this License applies to the Document are reproduced in all copies, and that you add no other conditions whatsoever to those of this License. You may not use technical measures to obstruct or control the reading or further copying of the copies you make or distribute. However, you may accept compensation in exchange for copies. If you distribute a large enough number of copies you must also follow the conditions in section 3.

You may also lend copies, under the same conditions stated above, and you may publicly display copies.

3. COPYING IN QUANTITY

If you publish printed copies (or copies in media that commonly have printed covers) of the Document, numbering more than 100, and the Document's license notice requires Cover Texts, you must enclose the copies in covers that carry, clearly and legibly, all these Cover Texts: Front-Cover Texts on the front cover, and Back-Cover Texts on the back cover. Both covers must also clearly and legibly identify you as the publisher of these copies. The front cover must present the full title with all words of the title equally prominent and visible. You may add other material on the covers in addition. Copying with changes limited to the covers, as long as they preserve the title of the Document and satisfy these conditions, can be treated as verbatim copying in other respects.

If the required texts for either cover are too voluminous to fit legibly, you should put the first ones listed (as many as fit reasonably) on the actual cover, and continue the rest onto adjacent pages.

If you publish or distribute Opaque copies of the Document numbering more than 100, you must either include a machine-readable Transparent copy along with each Opaque copy, or state in or with each Opaque copy a computer-network location from which the general network-using public has access to download using public-standard network protocols a complete Transparent copy of the Document, free of added material. If you use the latter option, you must take reasonably prudent steps, when you begin distribution of Opaque copies in quantity, to ensure that this Transparent copy will remain thus accessible at the stated location until at least one year after the last time you distribute an Opaque copy (directly or through your agents or retailers) of that edition to the public.

It is requested, but not required, that you contact the authors of the Document well before redistributing any large number of copies, to give them a chance to provide you with an updated version of the Document.

4. MODIFICATIONS

You may copy and distribute a Modified Version of the Document under the conditions of sections 2 and 3 above, provided that you release the Modified Version under precisely this License, with the Modified Version filling the role of the Document, thus licensing distribution and modification of the Modified Version to whoever possesses a copy of it. In addition, you must do these things in the Modified Version:

- A. Use in the Title Page (and on the covers, if any) a title distinct from that of the Document, and from those of previous versions (which should, if there were any, be listed in the History section of the Document). You may use the same title as a previous version if the original publisher of that version gives permission.
- B. List on the Title Page, as authors, one or more persons or entities responsible for authorship of the modifications in the Modified Version, together with at least five of the principal authors of the Document (all of its principal authors, if it has fewer than five), unless they release you from this requirement.
- C. State on the Title page the name of the publisher of the Modified Version, as the publisher.
- D. Preserve all the copyright notices of the Document.

- E. Add an appropriate copyright notice for your modifications adjacent to the other copyright notices.
- F. Include, immediately after the copyright notices, a license notice giving the public permission to use the Modified Version under the terms of this License, in the form shown in the Addendum below.
- G. Preserve in that license notice the full lists of Invariant Sections and required Cover Texts given in the Document's license notice.
- H. Include an unaltered copy of this License.
- I. Preserve the section Entitled "History", Preserve its Title, and add to it an item stating at least the title, year, new authors, and publisher of the Modified Version as given on the Title Page. If there is no section Entitled "History" in the Document, create one stating the title, year, authors, and publisher of the Document as given on its Title Page, then add an item describing the Modified Version as stated in the previous sentence.
- J. Preserve the network location, if any, given in the Document for public access to a Transparent copy of the Document, and likewise the network locations given in the Document for previous versions it was based on. These may be placed in the "History" section. You may omit a network location for a work that was published at least four years before the Document itself, or if the original publisher of the version it refers to gives permission.
- K. For any section Entitled "Acknowledgements" or "Dedications", Preserve the Title of the section, and preserve in the section all the substance and tone of each of the contributor acknowledgements and/or dedications given therein.
- L. Preserve all the Invariant Sections of the Document, unaltered in their text and in their titles. Section numbers or the equivalent are not considered part of the section titles.
- M. Delete any section Entitled "Endorsements". Such a section may not be included in the Modified Version.
- N. Do not retitle any existing section to be Entitled "Endorsements" or to conflict in title with any Invariant Section.
- O. Preserve any Warranty Disclaimers.

If the Modified Version includes new front-matter sections or appendices that qualify as Secondary Sections and contain no material copied from the Document, you may at your option designate some or all of these sections as invariant. To do this, add their titles to the list of Invariant Sections in the Modified Version's license notice. These titles must be distinct from any other section titles.

You may add a section Entitled "Endorsements", provided it contains nothing but endorsements of your Modified Version by various parties—for example, statements of peer review or that the text has been approved by an organization as the authoritative definition of a standard.

You may add a passage of up to five words as a Front-Cover Text, and a passage of up to 25 words as a Back-Cover Text, to the end of the list of Cover Texts in the Modified Version. Only one passage of Front-Cover Text and one of Back-Cover Text may be added by (or through arrangements made by) any one entity. If the Document already includes a cover text for the same cover, previously added by you or by arrangement made by the same entity you are acting on behalf of, you may not add another; but you may replace the old one, on explicit permission from the previous publisher that added the old one.

The author(s) and publisher(s) of the Document do not by this License give permission to use their names for publicity for or to assert or imply endorsement of any Modified Version.

5. COMBINING DOCUMENTS

You may combine the Document with other documents released under this License, under the terms defined in section 4 above for modified versions, provided that you include in the combination all of the Invariant Sections of all of the original documents, unmodified, and list them all as Invariant Sections of your combined work in its license notice, and that you preserve all their Warranty Disclaimers.

The combined work need only contain one copy of this License, and multiple identical Invariant Sections may be replaced with a single copy. If there are multiple Invariant Sections with the same name but different contents, make the title of each such section unique by adding at the end of it, in parentheses, the name of the original author or publisher of that section if known, or else a unique number. Make the same adjustment to the section titles in the list of Invariant Sections in the license notice of the combined work.

In the combination, you must combine any sections Entitled "History" in the various original documents, forming one section Entitled "History"; likewise combine any sections Entitled "Acknowledgements", and any sections Entitled "Dedications". You must delete all sections Entitled "Endorsements".

6. COLLECTIONS OF DOCUMENTS

You may make a collection consisting of the Document and other documents released under this License, and replace the individual copies of this License in the various documents with a single copy that is included in the collection, provided that you follow the rules of this License for verbatim copying of each of the documents in all other respects.

You may extract a single document from such a collection, and distribute it individually under this License, provided you insert a copy of this License into the extracted document, and follow this License in all other respects regarding verbatim copying of that document.

7. AGGREGATION WITH INDEPENDENT WORKS

A compilation of the Document or its derivatives with other separate and independent documents or works, in or on a volume of a storage or distribution medium, is called an "aggregate" if the copyright resulting from the compilation is not used to limit the legal rights of the compilation's users beyond what the individual works permit. When the Document is included in an aggregate, this License does not apply to the other works in the aggregate which are not themselves derivative works of the Document.

If the Cover Text requirement of section 3 is applicable to these copies of the Document, then if the Document is less than one half of the entire aggregate, the Document's Cover Texts may be placed on covers that bracket the Document within the aggregate, or the electronic equivalent of covers if the Document is in electronic form. Otherwise they must appear on printed covers that bracket the whole aggregate.

8. TRANSLATION

Translation is considered a kind of modification, so you may distribute translations of the Document under the terms of section 4. Replacing Invariant Sections with translations requires special permission from their copyright holders, but you may include translations of some or all

Invariant Sections in addition to the original versions of these Invariant Sections. You may include a translation of this License, and all the license notices in the Document, and any Warranty Disclaimers, provided that you also include the original English version of this License and the original versions of those notices and disclaimers. In case of a disagreement between the translation and the original version of this License or a notice or disclaimer, the original version will prevail.

If a section in the Document is Entitled "Acknowledgements", "Dedications", or "History", the requirement (section 4) to Preserve its Title (section 1) will typically require changing the actual title.

9. TERMINATION

You may not copy, modify, sublicense, or distribute the Document except as expressly provided for under this License. Any other attempt to copy, modify, sublicense or distribute the Document is void, and will automatically terminate your rights under this License. However, parties who have received copies, or rights, from you under this License will not have their licenses terminated so long as such parties remain in full compliance.

10. FUTURE REVISIONS OF THIS LICENSE

The Free Software Foundation may publish new, revised versions of the GNU Free Documentation License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns. See <http://www.gnu.org/copyleft/>.

Each version of the License is given a distinguishing version number. If the Document specifies that a particular numbered version of this License "or any later version" applies to it, you have the option of following the terms and conditions either of that specified version or of any later version that has been published (not as a draft) by the Free Software Foundation. If the Document does not specify a version number of this License, you may choose any version ever published (not as a draft) by the Free Software Foundation.

ADDENDUM: How to use this License for your documents

Copyright (c) YEAR YOUR NAME.

Permission is granted to copy, distribute and/or modify this document
under the terms of the GNU Free Documentation License, Version 1.2

or any later version published by the Free Software Foundation;
with no Invariant Sections, no Front-Cover Texts, and no Back-Cover Texts.
A copy of the license is included in the section entitled “GNU
Free Documentation License”.

If you have Invariant Sections, Front-Cover Texts and Back-Cover Texts, replace the “ with...
Texts.” line with this:

with the Invariant Sections being LIST THEIR TITLES, with the
Front-Cover Texts being LIST, and with the Back-Cover Texts being LIST.

If you have Invariant Sections without Cover Texts, or some other combination of the three,
merge those two alternatives to suit the situation.

If your document contains nontrivial examples of program code, we recommend releasing these
examples in parallel under your choice of free software license, such as the GNU General Public
License, to permit their use in free software.