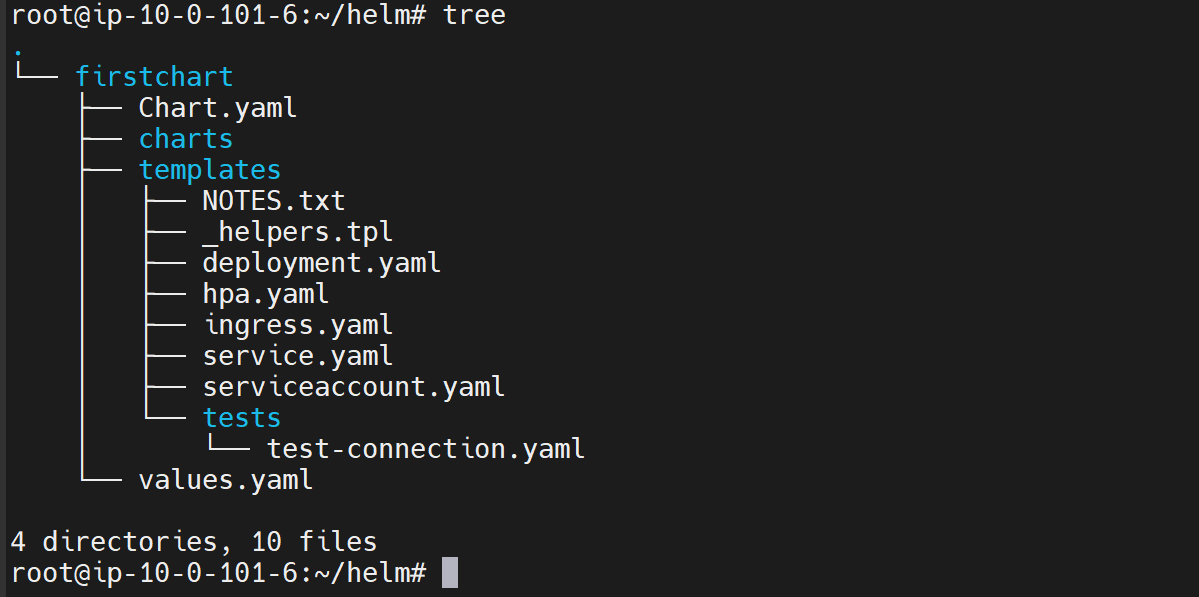
**39. Values.yaml**

--- note – we will learn about the values.yml file, which sits right at the root level of our chart next to the chart.yaml.

--- Open the values dot yaml in your text editor so the values dot yaml

--- tree



--- cat values.yml

# Default values for firstchart.

# This is a YAML-formatted file.

# Declare variables to be passed into your templates.

replicaCount: 1

image:

  repository: nginx

  pullPolicy: IfNotPresent

  # Overrides the image tag whose default is the chart appVersion.

  tag: ""

imagePullSecrets: []

nameOverride: ""

fullnameOverride: ""

serviceAccount:

  # Specifies whether a service account should be created

  create: true

  # Annotations to add to the service account

  annotations: {}

  # The name of the service account to use.

  # If not set and create is true, a name is generated using the fullname template

  name: ""

podAnnotations: {}

podSecurityContext: {}

  # fsGroup: 2000

securityContext: {}

  # capabilities:

  #   drop:

  #   - ALL

  # readOnlyRootFilesystem: true

  # runAsNonRoot: true

  # runAsUser: 1000

service:

  type: ClusterIP

  port: 80

ingress:

  enabled: false

  className: ""

  annotations: {}

    # kubernetes.io/ingress.class: nginx

    # kubernetes.io/tls-acme: "true"

  hosts:

    - host: chart-example.local

      paths:

        - path: /

          pathType: ImplementationSpecific

  tls: []

  #  - secretName: chart-example-tls

  #    hosts:

  #      - chart-example.local

resources: {}

  # We usually recommend not to specify default resources and to leave this as a conscious

  # choice for the user. This also increases chances charts run on environments with little

  # resources, such as Minikube. If you do want to specify resources, uncomment the following

  # lines, adjust them as necessary, and remove the curly braces after 'resources:'.

  # limits:

  #   cpu: 100m

  #   memory: 128Mi

  # requests:

  #   cpu: 100m

  #   memory: 128Mi

autoscaling:

  enabled: false

  minReplicas: 1

  maxReplicas: 100

  targetCPUUtilizationPercentage: 80

  # targetMemoryUtilizationPercentage: 80

nodeSelector: {}

tolerations: []

affinity: {}

--- **note** - As you already know by now, it will have the default values that will be used across the template files under the template’s directory.

--- So, the user need not provide all the values, when he does not provide a value. These default values will be used and he can always override these defaults by providing --values.yml option or --set flag when he does an installation.

--- **replicaCount:** 1 – the number of replicas, it should be creating.

**Image section**

image:

  repository: nginx

  pullPolicy: IfNotPresent

  # Overrides the image tag whose default is the chart appVersion.

  tag: ""

imagePullSecrets: []

nameOverride: ""

fullnameOverride: ""

--- **repository: nginx** – pull this image

--- **pullPolicy: IfNotPresent** – if the image is not present locally then pull the image form the repository.

--- **tag:””** - the tag that should be used when the image is being pulled.

--- **imagePullSecrets: []** - if this image needs any security credentials. Then information can be passed as image pulled secrets right here.

**Service account**

serviceAccount:

  # Specifies whether a service account should be created

  create: true

  # Annotations to add to the service account

  annotations: {}

  # The name of the service account to use.

  # If not set and create is true, a name is generated using the fullname template

  name: ""

--- the service account, which is enabled by default. As you can see, create is true. service account will be created for this chart. When we do an installation, for every installation, there will be a service account created

--- if you turn it to false. That will not be created. You can go to the serviceaccount.yaml file under your templates.

**Service type**

service:

  type: ClusterIP

  port: 80

--- which determines how our pods are exposed out. By default, it uses cluster IP. You can change this to Nodeport and you can also configure ingress if you want to configure ingress. You need to have an ingress controller up and running on your Kubernetes cluster.

**Autoscaler**

autoscaling:

  enabled: false

  minReplicas: 1

  maxReplicas: 100

  targetCPUUtilizationPercentage: 80

--- **note** - You can always enable auto scaling, provided the minimum replicas, maximum replicas, target CPU utilization and so on.