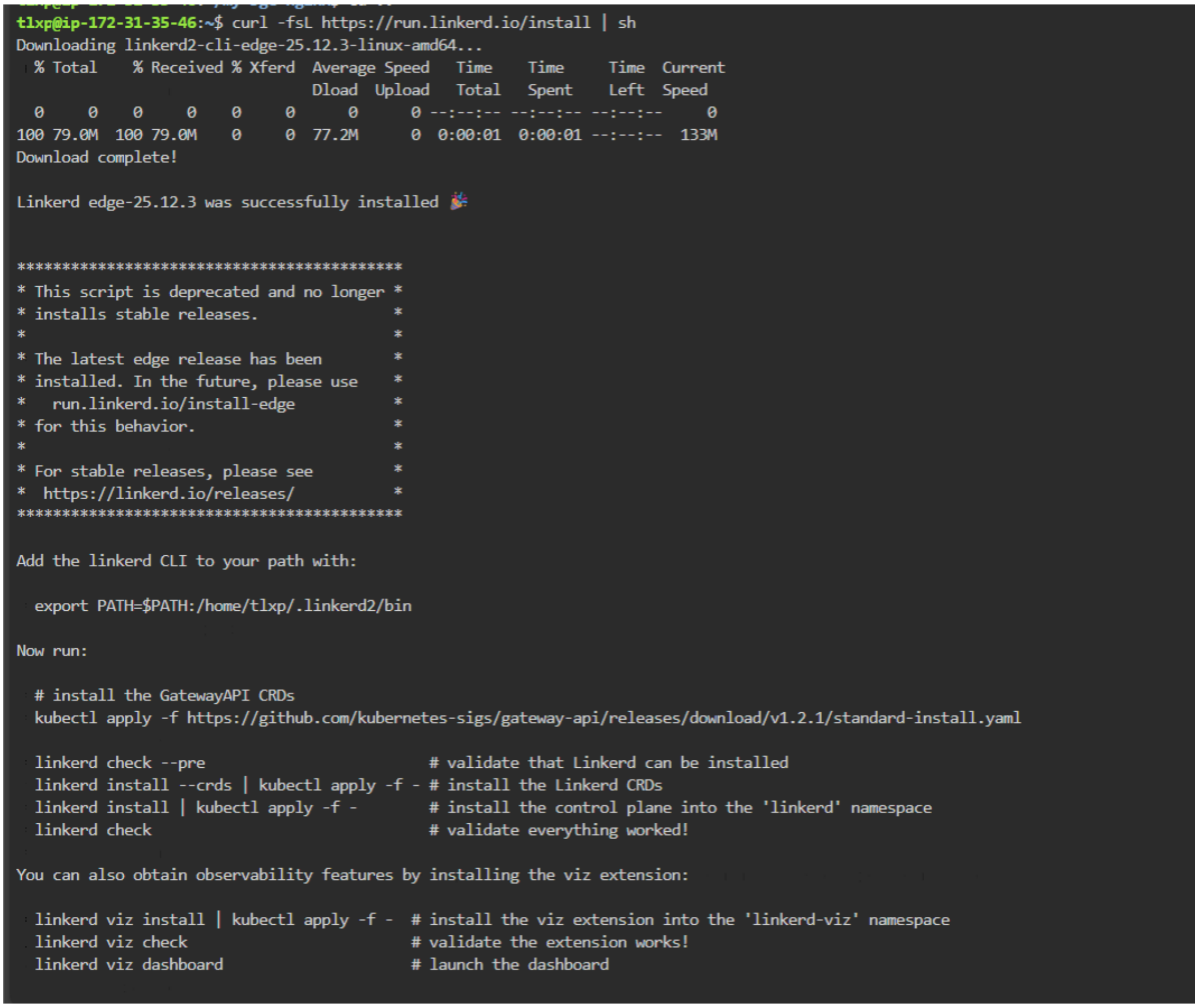
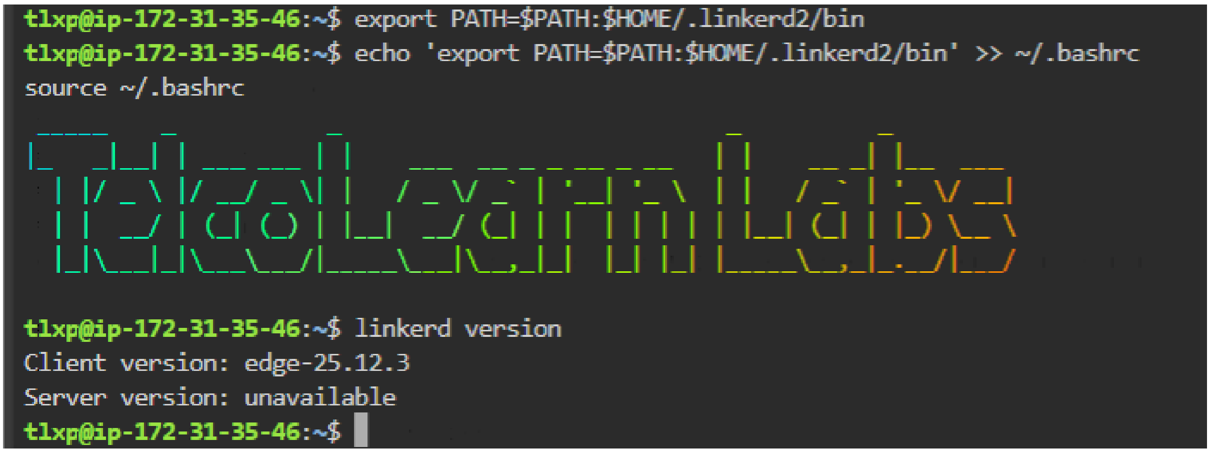
Service Mesh with Linkerd

Install Linkerd CLI on the VM



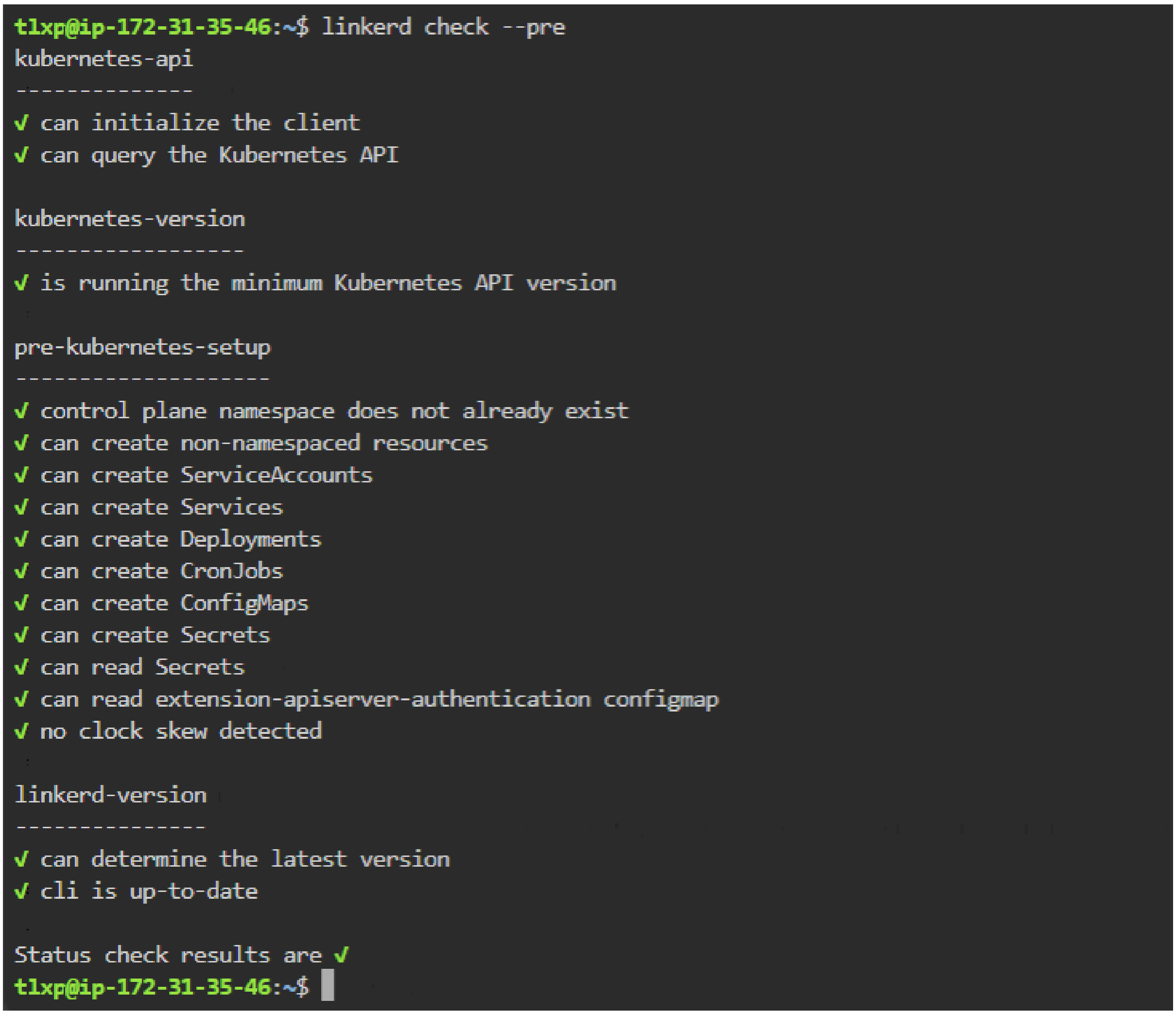
```
tlxp@ip-172-31-35-46:~$ curl -fsL https://run.linkerd.io/install | sh
Downloading linkerd2-cli-edge-25.12.3-linux-amd64...
  % Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
                                 Dload  Upload   Total   Spent    Left  Speed
  0     0    0     0    0     0      0      0 --:--:-- --:--:-- --:--:--     0
100 79.0M  100 79.0M    0     0  77.2M      0  0:00:01  0:00:01 --:--:--  133M
Download complete!

Linkerd edge-25.12.3 was successfully installed 🎉


*******************************************
* This script is deprecated and no longer *
* installs stable releases.               *
*                                         *
* The latest edge release has been        *
* installed. In the future, please use    *
*   run.linkerd.io/install-edge           *
* for this behavior.                      *
*                                         *
* For stable releases, please see         *
*  https://linkerd.io/releases/           *
*******************************************

Add the linkerd CLI to your path with:

  export PATH=$PATH:/home/tlxp/.linkerd2/bin

Now run:

  # install the GatewayAPI CRDs
  kubectl apply -f https://github.com/kubernetes-sigs/gateway-api/releases/download/v1.2.1/standard-install.yaml

  linkerd check --pre                         # validate that Linkerd can be installed
  linkerd install --crds | kubectl apply -f - # install the Linkerd CRDs
  linkerd install | kubectl apply -f -        # install the control plane into the 'linkerd' namespace
  linkerd check                               # validate everything worked!

You can also obtain observability features by installing the viz extension:

  linkerd viz install | kubectl apply -f -  # install the viz extension into the 'linkerd-viz' namespace
  linkerd viz check                         # validate the extension works!
  linkerd viz dashboard                     # launch the dashboard
```

Verify:



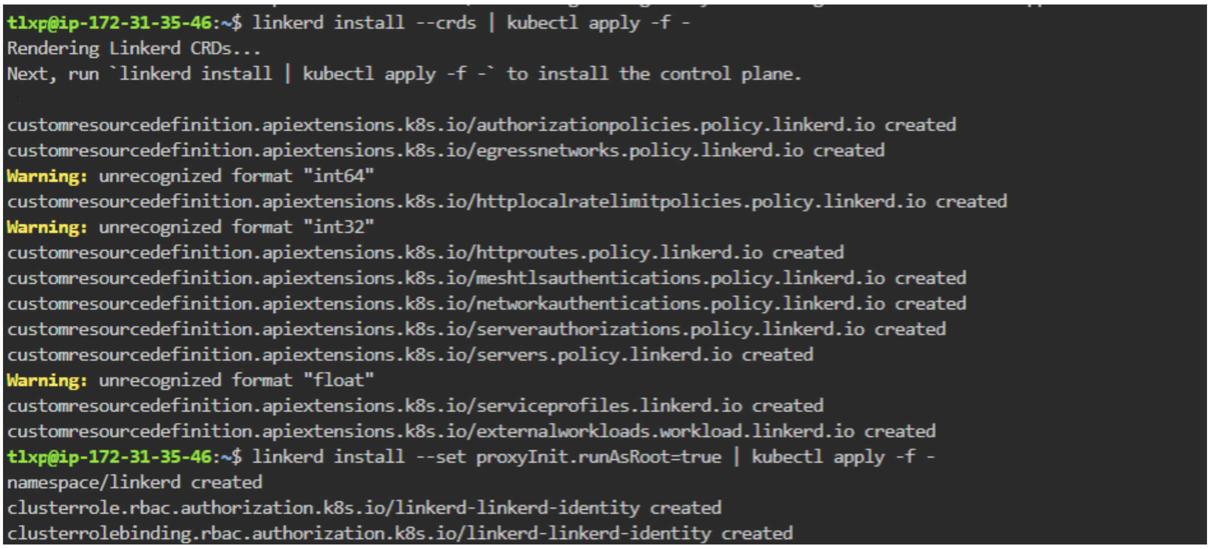
```
tlxp@ip-172-31-35-46:~$ export PATH=$PATH:$HOME/.linkerd2/bin
tlxp@ip-172-31-35-46:~$ echo 'export PATH=$PATH:$HOME/.linkerd2/bin' >> ~/.bashrc
source ~/.bashrc

 _____    _      _                          _          _
|_   _|__| | ___| |    ___  __ _ _ __ _ __ | |    __ _| |__  ___
  | |/ _ \ |/ _ \ |   / _ \/ _` | '__| '_ \| |   / _` | '_ \/ __|
  | |  __/ |  __/ |__|  __/ (_| | |  | | | | |__| (_| | |_) \__ \
  |_|\___|_|\___|_____\___|\__,_|_|  |_| |_|_____\__,_|_.__/|___/

tlxp@ip-172-31-35-46:~$ linkerd version
Client version: edge-25.12.3
Server version: unavailable
tlxp@ip-172-31-35-46:~$
```

Pre-check Kubernetes Cluster



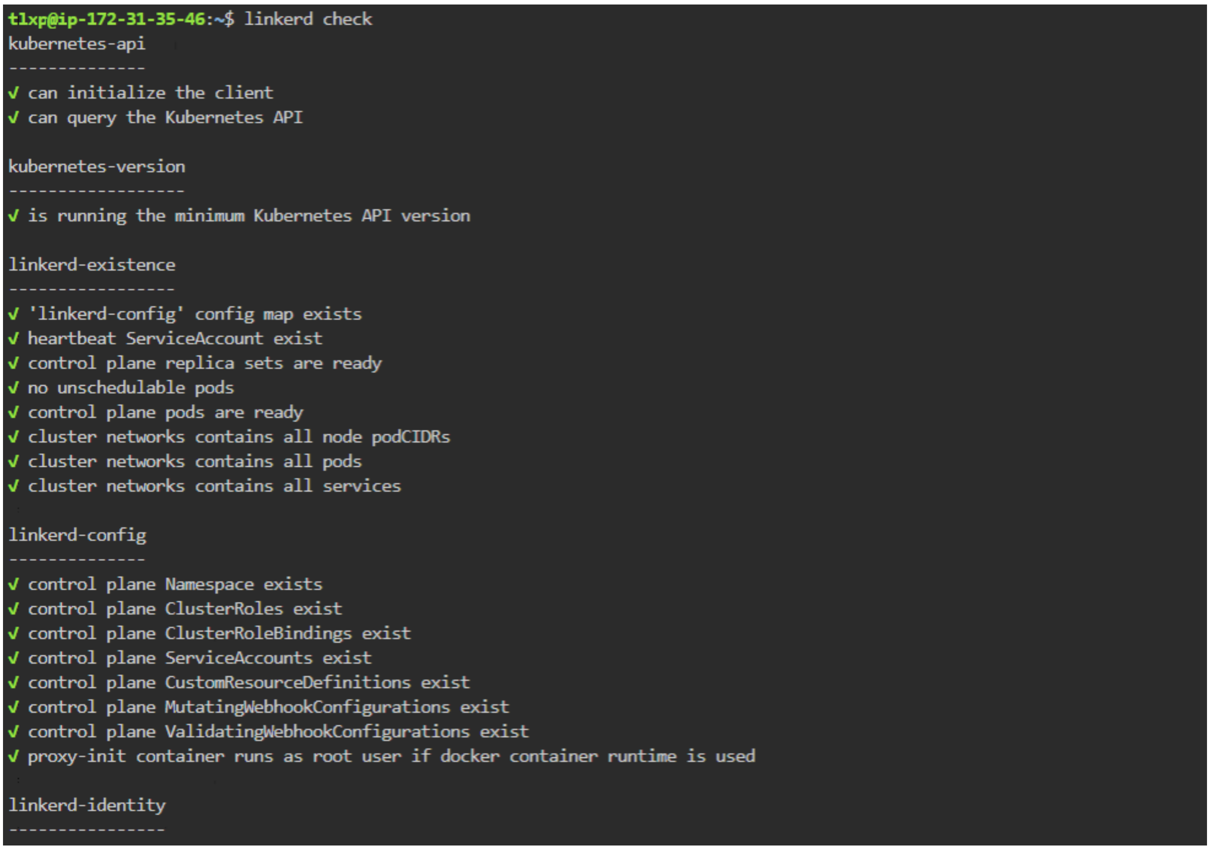
```
tlxp@ip-172-31-35-46:~$ linkerd check --pre
kubernetes-api
--------------
√ can initialize the client
√ can query the Kubernetes API

kubernetes-version
------------------
√ is running the minimum Kubernetes API version

pre-kubernetes-setup
--------------------
√ control plane namespace does not already exist
√ can create non-namespaced resources
√ can create ServiceAccounts
√ can create Services
√ can create Deployments
√ can create CronJobs
√ can create ConfigMaps
√ can create Secrets
√ can read Secrets
√ can read extension-apiserver-authentication configmap
√ no clock skew detected

linkerd-version
---------------
√ can determine the latest version
√ cli is up-to-date

Status check results are √
tlxp@ip-172-31-35-46:~$
```

Install Linkerd Control Plane

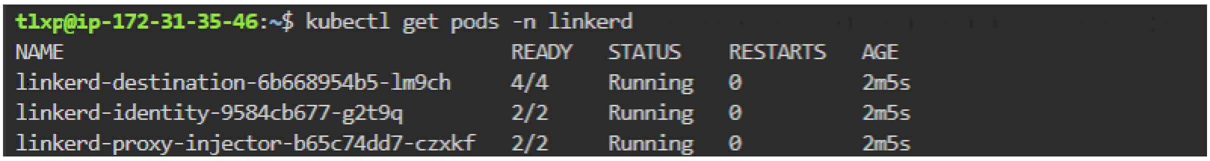


```
tlxp@ip-172-31-35-46:~$ linkerd install --crds | kubectl apply -f -
Rendering Linkerd CRDs...
Next, run `linkerd install | kubectl apply -f -` to install the control plane.

customresourcedefinition.apiextensions.k8s.io/authorizationpolicies.policy.linkerd.io created
customresourcedefinition.apiextensions.k8s.io/egressnetworks.policy.linkerd.io created
Warning: unrecognized format "int64"
customresourcedefinition.apiextensions.k8s.io/httplocalratelimitpolicies.policy.linkerd.io created
Warning: unrecognized format "int32"
customresourcedefinition.apiextensions.k8s.io/httproutes.policy.linkerd.io created
customresourcedefinition.apiextensions.k8s.io/meshtlsauthentications.policy.linkerd.io created
customresourcedefinition.apiextensions.k8s.io/networkauthentications.policy.linkerd.io created
customresourcedefinition.apiextensions.k8s.io/serverauthorizations.policy.linkerd.io created
customresourcedefinition.apiextensions.k8s.io/servers.policy.linkerd.io created
Warning: unrecognized format "float"
customresourcedefinition.apiextensions.k8s.io/serviceprofiles.linkerd.io created
customresourcedefinition.apiextensions.k8s.io/externalworkloads.workload.linkerd.io created
tlxp@ip-172-31-35-46:~$ linkerd install --set proxyInit.runAsRoot=true | kubectl apply -f -
namespace/linkerd created
clusterrole.rbac.authorization.k8s.io/linkerd-linkerd-identity created
clusterrolebinding.rbac.authorization.k8s.io/linkerd-linkerd-identity created
```

Verify Installation

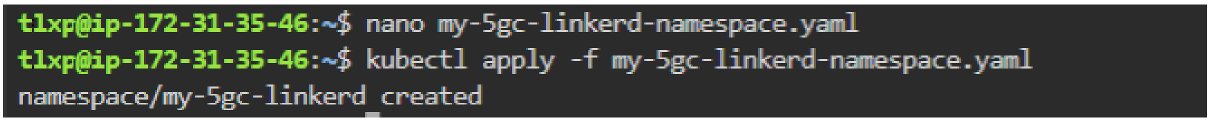


```
tlxp@ip-172-31-35-46:~$ linkerd check
kubernetes-api
--------------
√ can initialize the client
√ can query the Kubernetes API

kubernetes-version
------------------
√ is running the minimum Kubernetes API version

linkerd-existence
-----------------
√ 'linkerd-config' config map exists
√ heartbeat ServiceAccount exist
√ control plane replica sets are ready
√ no unschedulable pods
√ control plane pods are ready
√ cluster networks contains all node podCIDRs
√ cluster networks contains all pods
√ cluster networks contains all services

linkerd-config
--------------
√ control plane Namespace exists
√ control plane ClusterRoles exist
√ control plane ClusterRoleBindings exist
√ control plane ServiceAccounts exist
√ control plane CustomResourceDefinitions exist
√ control plane MutatingWebhookConfigurations exist
√ control plane ValidatingWebhookConfigurations exist
√ proxy-init container runs as root user if docker container runtime is used

linkerd-identity
----------------
```

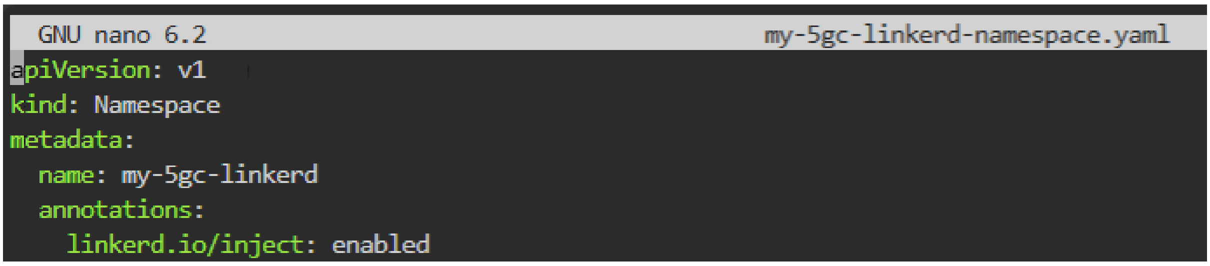


```
tlxp@ip-172-31-35-46:~$ kubectl get pods -n linkerd
NAME                                     READY   STATUS    RESTARTS   AGE
linkerd-destination-6b668954b5-lm9ch     4/4     Running   0          2m5s
linkerd-identity-9584cb677-g2t9q         2/2     Running   0          2m5s
linkerd-proxy-injector-b65c74dd7-czxkf   2/2     Running   0          2m5s
```

Prepare Linkerd-enabled Namespace

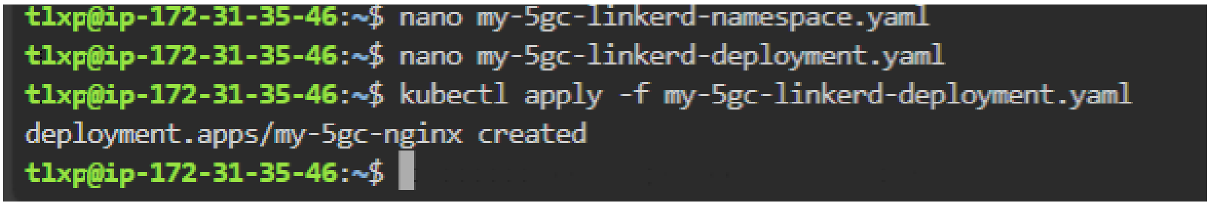


```
tlxp@ip-172-31-35-46:~$ nano my-5gc-linkerd-namespace.yaml
tlxp@ip-172-31-35-46:~$ kubectl apply -f my-5gc-linkerd-namespace.yaml
namespace/my-5gc-linkerd created
```

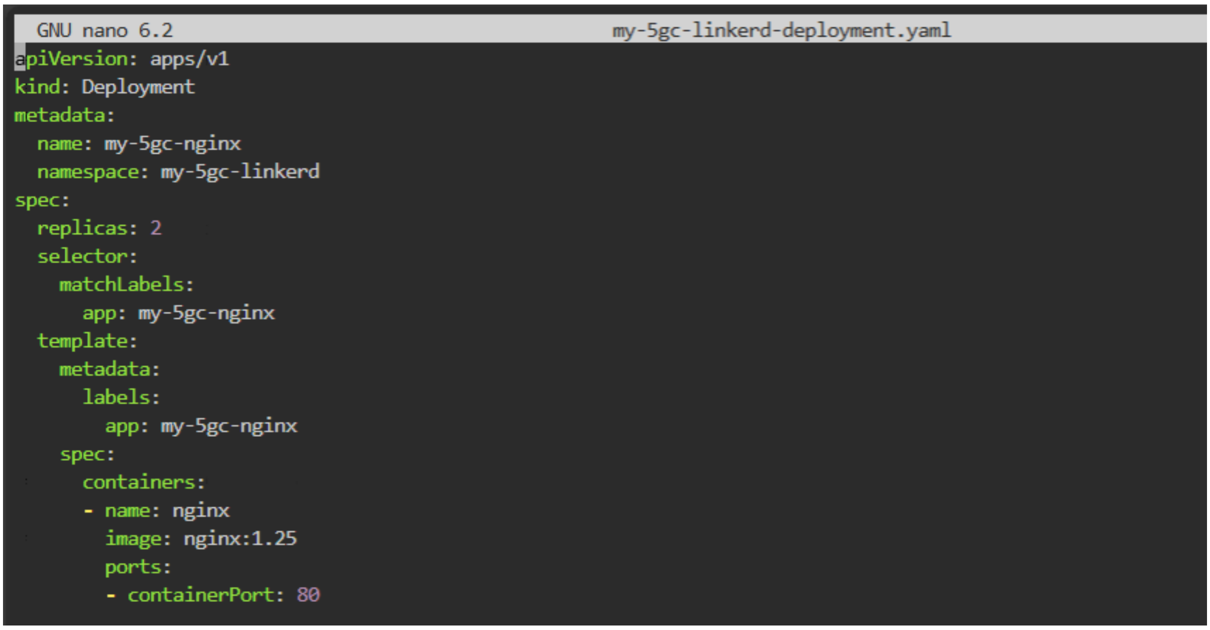


```
  GNU nano 6.2                                    my-5gc-linkerd-namespace.yaml
apiVersion: v1
kind: Namespace
metadata:
  name: my-5gc-linkerd
  annotations:
    linkerd.io/inject: enabled
```

Deploy Nginx into Linkerd Namespace



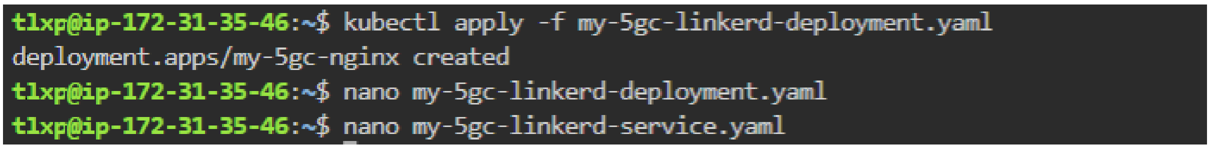
```
tlxp@ip-172-31-35-46:~$ nano my-5gc-linkerd-namespace.yaml
tlxp@ip-172-31-35-46:~$ nano my-5gc-linkerd-deployment.yaml
tlxp@ip-172-31-35-46:~$ kubectl apply -f my-5gc-linkerd-deployment.yaml
deployment.apps/my-5gc-nginx created
tlxp@ip-172-31-35-46:~$
```



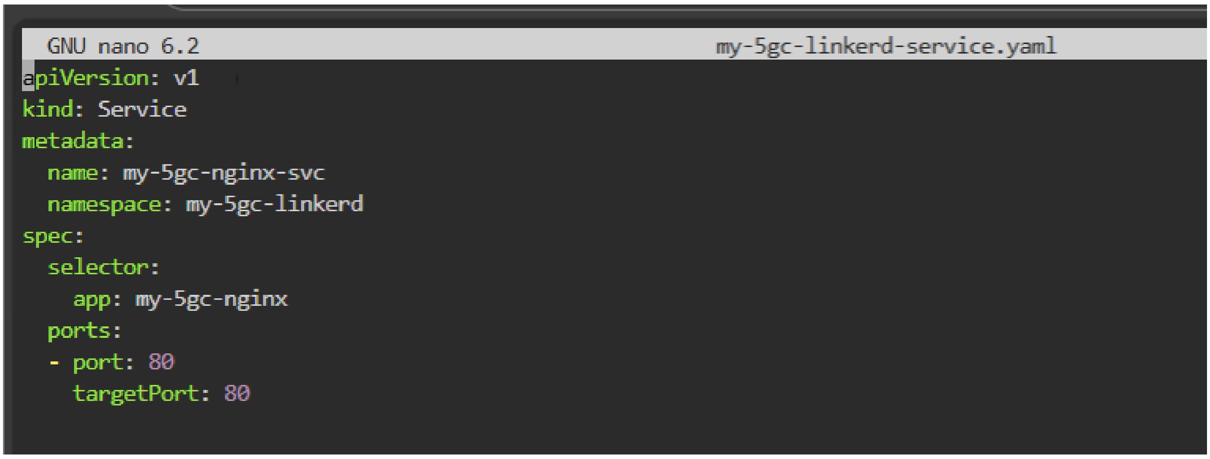
```
apiVersion: apps/v1
kind: Deployment
metadata:
  name: my-5gc-nginx
  namespace: my-5gc-linkerd
spec:
  replicas: 2
  selector:
    matchLabels:
      app: my-5gc-nginx
  template:
    metadata:
      labels:
        app: my-5gc-nginx
    spec:
      containers:
      - name: nginx
        image: nginx:1.25
        ports:
        - containerPort: 80
```

Expose the Service

Create service YAML

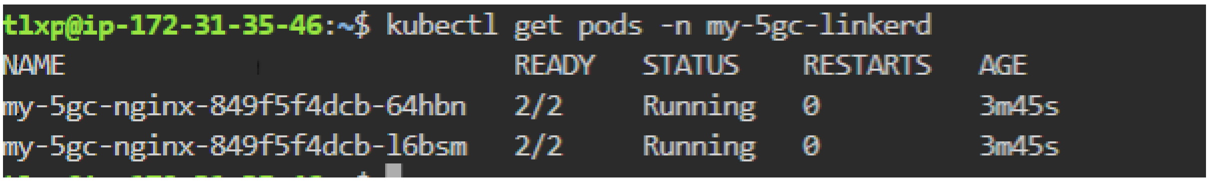


```
tlxp@ip-172-31-35-46:~$ kubectl apply -f my-5gc-linkerd-deployment.yaml
deployment.apps/my-5gc-nginx created
tlxp@ip-172-31-35-46:~$ nano my-5gc-linkerd-deployment.yaml
tlxp@ip-172-31-35-46:~$ nano my-5gc-linkerd-service.yaml
```



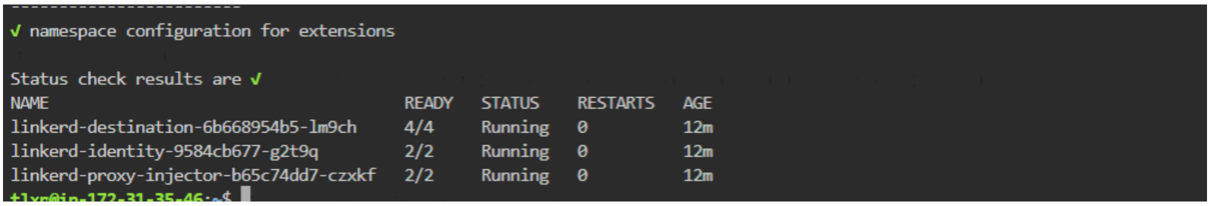
```
  GNU nano 6.2                                   my-5gc-linkerd-service.yaml
apiVersion: v1
kind: Service
metadata:
  name: my-5gc-nginx-svc
  namespace: my-5gc-linkerd
spec:
  selector:
    app: my-5gc-nginx
  ports:
  - port: 80
    targetPort: 80
```

Verify Linkerd Sidecar Injection



```
tlxp@ip-172-31-35-46:~$ kubectl get pods -n my-5gc-linkerd
NAME                              READY   STATUS    RESTARTS   AGE
my-5gc-nginx-849f5f4dcb-64hbn     2/2     Running   0          3m45s
my-5gc-nginx-849f5f4dcb-l6bsm     2/2     Running   0          3m45s
```

Observe Traffic via Linkerd



```
------------------------
√ namespace configuration for extensions

Status check results are √
NAME                                   READY   STATUS    RESTARTS   AGE
linkerd-destination-6b668954b5-lm9ch   4/4     Running   0          12m
linkerd-identity-9584cb677-g2t9q       2/2     Running   0          12m
linkerd-proxy-injector-b65c74dd7-czxkf 2/2     Running   0          12m
tlxp@ip-172-31-35-46:~$
```

Newer versions of Linkerd require the Kubernetes Gateway API CRDs to be installed before installing Linkerd CRDs and the control plane.