```
MacBook-Pro:~ zolatech$ minikube version
minikube version: v1.35.0
commit: dd5d320e41b5451cdf3c01891bc4e13d189586ed
MacBook-Pro:~ zolatech$ kubectl version --client
Client Version: v1.32.1
Kustomize Version: v5.5.0
MacBook-Pro:~ zolatech$ minikube start
😄  minikube v1.35.0 on Darwin 15.3 (arm64)
✨  Using the docker driver based on existing profile
👍  Starting "minikube" primary control-plane node in "minikube" cluster
🚜  Pulling base image v0.0.46 ...
🔄  Restarting existing docker container for "minikube" ...
🐳  Preparing Kubernetes v1.32.0 on Docker 27.4.1 ...
🔎  Verifying Kubernetes components...
    ▪ Using image gcr.io/k8s-minikube/storage-provisioner:v5
💡  After the addon is enabled, please run "minikube tunnel" and your ingress resources would be available at "127.0.0.1"
    ▪ Using image registry.k8s.io/ingress-nginx/controller:v1.11.3
    ▪ Using image registry.k8s.io/ingress-nginx/kube-webhook-certgen:v1.4.4
    ▪ Using image registry.k8s.io/ingress-nginx/kube-webhook-certgen:v1.4.4
🔎  Verifying ingress addon...
🌟  Enabled addons: storage-provisioner, default-storageclass, ingress
🏄  Done! kubectl is now configured to use "minikube" cluster and "default" namespace by default
MacBook-Pro:~ zolatech$ █
```

```
                                                              -bash
MacBook-Pro:~ zolatech$ kubectl apply -f /Users/zolatech/Desktop/kubernetes_cluster_resources.yaml
service/lb-service created
deployment.apps/frontend-deployment created
MacBook-Pro:~ zolatech$ kubectl get all -o yaml
apiVersion: v1
items:
- apiVersion: v1
  kind: Pod
  metadata:
    creationTimestamp: "2025-02-09T14:16:35Z"
    generateName: frontend-deployment-776c847ddc-
    labels:
      app: frontend
      pod-template-hash: 776c847ddc
    name: frontend-deployment-776c847ddc-bv972
    namespace: default
    ownerReferences:
    - apiVersion: apps/v1
      blockOwnerDeletion: true
      controller: true
      kind: ReplicaSet
      name: frontend-deployment-776c847ddc
      uid: d6036852-7056-45ff-b6f8-5b83526e20f4
    resourceVersion: "2660"
    uid: c3809ca7-9681-4b52-8333-62fa6f0b5f97
  spec:
    containers:
    - image: nginx
      imagePullPolicy: Always
      name: frontend-container
      resources: {}
      terminationMessagePath: /dev/termination-log
      terminationMessagePolicy: File
      volumeMounts:
      - mountPath: /var/run/secrets/kubernetes.io/serviceaccount
        name: kube-api-access-vkk8m
        readOnly: true
    dnsPolicy: ClusterFirst
```

```yaml
dnsPolicy: ClusterFirst
enableServiceLinks: true
nodeName: minikube
preemptionPolicy: PreemptLowerPriority
priority: 0
restartPolicy: Always
schedulerName: default-scheduler
securityContext: {}
serviceAccount: default
serviceAccountName: default
terminationGracePeriodSeconds: 30
tolerations:
- effect: NoExecute
  key: node.kubernetes.io/not-ready
  operator: Exists
  tolerationSeconds: 300
- effect: NoExecute
  key: node.kubernetes.io/unreachable
  operator: Exists
  tolerationSeconds: 300
volumes:
- name: kube-api-access-vkk8m
  projected:
    defaultMode: 420
    sources:
    - serviceAccountToken:
        expirationSeconds: 3607
        path: token
    - configMap:
        items:
        - key: ca.crt
          path: ca.crt
        name: kube-root-ca.crt
    - downwardAPI:
        items:
        - fieldRef:
            apiVersion: v1
            fieldPath: metadata.namespace
```

```yaml
            fieldPath: metadata.namespace
          path: namespace
status:
  conditions:
  - lastProbeTime: null
    lastTransitionTime: "2025-02-09T14:16:37Z"
    status: "True"
    type: PodReadyToStartContainers
  - lastProbeTime: null
    lastTransitionTime: "2025-02-09T14:16:35Z"
    status: "True"
    type: Initialized
  - lastProbeTime: null
    lastTransitionTime: "2025-02-09T14:16:37Z"
    status: "True"
    type: Ready
  - lastProbeTime: null
    lastTransitionTime: "2025-02-09T14:16:37Z"
    status: "True"
    type: ContainersReady
  - lastProbeTime: null
    lastTransitionTime: "2025-02-09T14:16:35Z"
    status: "True"
    type: PodScheduled
  containerStatuses:
  - containerID: docker://fdd82d24040c61df981156ff737131d2e4a11b9d76a48d6b6f96a38764d5cd7b
    image: nginx:latest
    imageID: docker-pullable://nginx@sha256:91734281c0ebfc6f1aea979cffeed5079cfe786228a71cc6f1f46a228cde6e34
    lastState: {}
    name: frontend-container
    ready: true
    restartCount: 0
    started: true
    state:
      running:
        startedAt: "2025-02-09T14:16:36Z"
    volumeMounts:
    - mountPath: /var/run/secrets/kubernetes.io/serviceaccount
```

Containers

```
          - mountPath: /var/run/secrets/kubernetes.io/serviceaccount
            name: kube-api-access-vkk8m
            readOnly: true
            recursiveReadOnly: Disabled
      hostIP: 192.168.49.2
      hostIPs:
      - ip: 192.168.49.2
      phase: Running
      podIP: 10.244.0.20
      podIPs:
      - ip: 10.244.0.20
      qosClass: BestEffort
      startTime: "2025-02-09T14:16:35Z"
- apiVersion: v1
  kind: Pod
  metadata:
    creationTimestamp: "2025-02-09T14:16:35Z"
    generateName: frontend-deployment-776c847ddc-
    labels:
      app: frontend
      pod-template-hash: 776c847ddc
    name: frontend-deployment-776c847ddc-pmqtn
    namespace: default
    ownerReferences:
    - apiVersion: apps/v1
      blockOwnerDeletion: true
      controller: true
      kind: ReplicaSet
      name: frontend-deployment-776c847ddc
      uid: d6036852-7056-45ff-b6f8-5b83526e20f4
    resourceVersion: "2669"
    uid: 06e7b1fd-14e8-495d-bd5f-650fd08acd42
  spec:
    containers:
    - image: nginx
      imagePullPolicy: Always
      name: frontend-container
      resources: {}
```

Calendar

```yaml
    resources: {}
    terminationMessagePath: /dev/termination-log
    terminationMessagePolicy: File
    volumeMounts:
    - mountPath: /var/run/secrets/kubernetes.io/serviceaccount
      name: kube-api-access-pdm77
      readOnly: true
  dnsPolicy: ClusterFirst
  enableServiceLinks: true
  nodeName: minikube
  preemptionPolicy: PreemptLowerPriority
  priority: 0
  restartPolicy: Always
  schedulerName: default-scheduler
  securityContext: {}
  serviceAccount: default
  serviceAccountName: default
  terminationGracePeriodSeconds: 30
  tolerations:
  - effect: NoExecute
    key: node.kubernetes.io/not-ready
    operator: Exists
    tolerationSeconds: 300
  - effect: NoExecute
    key: node.kubernetes.io/unreachable
    operator: Exists
    tolerationSeconds: 300
  volumes:
  - name: kube-api-access-pdm77
    projected:
      defaultMode: 420
      sources:
      - serviceAccountToken:
          expirationSeconds: 3607
          path: token
      - configMap:
          items:
          - key: ca.crt
```

```
          path: ca.crt
        name: kube-root-ca.crt
    - downwardAPI:
        items:
        - fieldRef:
            apiVersion: v1
            fieldPath: metadata.namespace
          path: namespace
status:
  conditions:
  - lastProbeTime: null
    lastTransitionTime: "2025-02-09T14:16:38Z"
    status: "True"
    type: PodReadyToStartContainers
  - lastProbeTime: null
    lastTransitionTime: "2025-02-09T14:16:35Z"
    status: "True"
    type: Initialized
  - lastProbeTime: null
    lastTransitionTime: "2025-02-09T14:16:38Z"
    status: "True"
    type: Ready
  - lastProbeTime: null
    lastTransitionTime: "2025-02-09T14:16:38Z"
    status: "True"
    type: ContainersReady
  - lastProbeTime: null
    lastTransitionTime: "2025-02-09T14:16:35Z"
    status: "True"
    type: PodScheduled
  containerStatuses:
  - containerID: docker://42d2819c312011cc6e9c332b7a6ff021103f42da5ba4c2b8e77bdfa5c8da1a8b
    image: nginx:latest
    imageID: docker-pullable://nginx@sha256:91734281c0ebfc6f1aea979cffeed5079cfe786228a71cc6f1f46a228cde6e34
    lastState: {}
    name: frontend-container
    ready: true
    restartCount: 0
```

```yaml
        started: true
        state:
          running:
            startedAt: "2025-02-09T14:16:37Z"
        volumeMounts:
        - mountPath: /var/run/secrets/kubernetes.io/serviceaccount
          name: kube-api-access-pdm77
          readOnly: true
          recursiveReadOnly: Disabled
    hostIP: 192.168.49.2
    hostIPs:
    - ip: 192.168.49.2
    phase: Running
    podIP: 10.244.0.19
    podIPs:
    - ip: 10.244.0.19
    qosClass: BestEffort
    startTime: "2025-02-09T14:16:35Z"
- apiVersion: v1
  kind: Service
  metadata:
    creationTimestamp: "2025-02-09T14:08:33Z"
    labels:
      component: apiserver
      provider: kubernetes
    name: kubernetes
    namespace: default
    resourceVersion: "2280"
    uid: 60aa8daa-be9d-4527-8333-2b5465401c4f
  spec:
    clusterIP: 10.96.0.1
    clusterIPs:
    - 10.96.0.1
    internalTrafficPolicy: Cluster
    ipFamilies:
    - IPv4
    ipFamilyPolicy: SingleStack
    ports:
```

```yaml
      ports:
      - name: https
        port: 443
        protocol: TCP
        targetPort: 8443
      sessionAffinity: None
      type: ClusterIP
  status:
    loadBalancer: {}
- apiVersion: v1
  kind: Service
  metadata:
    annotations:
      kubectl.kubernetes.io/last-applied-configuration: |
        {"apiVersion":"v1","kind":"Service","metadata":{"annotations":{},"labels":{"app":"lb-service"},"name":"lb-service","namespace":"default"},"spec":{"ports":[{"port
":80}],"selector":{"app":"frontend"},"type":"LoadBalancer"}}
    creationTimestamp: "2025-02-09T14:16:35Z"
    labels:
      app: lb-service
    name: lb-service
    namespace: default
    resourceVersion: "2632"
    uid: 90da9aa8-2f21-4165-8408-01ee72133d51
  spec:
    allocateLoadBalancerNodePorts: true
    clusterIP: 10.111.2.25
    clusterIPs:
    - 10.111.2.25
    externalTrafficPolicy: Cluster
    internalTrafficPolicy: Cluster
    ipFamilies:
    - IPv4
    ipFamilyPolicy: SingleStack
    ports:
    - nodePort: 32628
      port: 80
      protocol: TCP
      targetPort: 80
    selector:
      app: frontend
    sessionAffinity: None
    type: LoadBalancer
  status:
    loadBalancer: {}
- apiVersion: apps/v1
  kind: Deployment
  metadata:
    annotations:
      deployment.kubernetes.io/revision: "1"
      kubectl.kubernetes.io/last-applied-configuration: |
        {"apiVersion":"apps/v1","kind":"Deployment","metadata":{"annotations":{},"name":"frontend-deployment","namespace":"default"},"spec":{"minReadySeconds":30,"replicas":2,"selector":{"matchLabels":{"app":"frontend"}},"strategy":{"rollingUpdate":{"maxS
":"frontend-container"}]}}}}}
    creationTimestamp: "2025-02-09T14:16:35Z"
    generation: 1
    name: frontend-deployment
    namespace: default
    resourceVersion: "2673"
    uid: f00a0e08-14f3-427c-944a-59c64372902d
  spec:
    minReadySeconds: 30
    progressDeadlineSeconds: 600
    replicas: 2
    revisionHistoryLimit: 10
    selector:
      matchLabels:
        app: frontend
    strategy:
      rollingUpdate:
        maxSurge: 1
        maxUnavailable: 0
      type: RollingUpdate
    template:
      metadata:
        creationTimestamp: null
        labels:
          app: frontend
      spec:
        containers:
        - image: nginx
          imagePullPolicy: Always
          name: frontend-container
          resources: {}
          terminationMessagePath: /dev/termination-log
          terminationMessagePolicy: File
        dnsPolicy: ClusterFirst
        restartPolicy: Always
        schedulerName: default-scheduler
        securityContext: {}
        terminationGracePeriodSeconds: 30
  status:
    conditions:
    - lastTransitionTime: "2025-02-09T14:16:35Z"
```

```
      - lastTransitionTime: "2025-02-09T14:16:35Z"
        lastUpdateTime: "2025-02-09T14:16:35Z"
        message: Deployment does not have minimum availability.
        reason: MinimumReplicasUnavailable
        status: "False"
        type: Available
      - lastTransitionTime: "2025-02-09T14:16:35Z"
        lastUpdateTime: "2025-02-09T14:16:38Z"
        message: ReplicaSet "frontend-deployment-776c847ddc" is progressing.
        reason: ReplicaSetUpdated
        status: "True"
        type: Progressing
      observedGeneration: 1
      readyReplicas: 2
      replicas: 2
      unavailableReplicas: 2
      updatedReplicas: 2
- apiVersion: apps/v1
  kind: ReplicaSet
  metadata:
    annotations:
      deployment.kubernetes.io/desired-replicas: "2"
      deployment.kubernetes.io/max-replicas: "3"
      deployment.kubernetes.io/revision: "1"
    creationTimestamp: "2025-02-09T14:16:35Z"
    generation: 1
    labels:
      app: frontend
      pod-template-hash: 776c847ddc
    name: frontend-deployment-776c847ddc
    namespace: default
    ownerReferences:
    - apiVersion: apps/v1
      blockOwnerDeletion: true
      controller: true
      kind: Deployment
      name: frontend-deployment
      uid: f00a0e08-14f3-427c-944a-59c64372902d
    resourceVersion: "2672"
    uid: d6036852-7056-45ff-b6f8-5b83526e20f4
  spec:
    minReadySeconds: 30
    replicas: 2
    selector:
      matchLabels:
        app: frontend
        pod-template-hash: 776c847ddc
    template:
      metadata:
        creationTimestamp: null
        labels:
          app: frontend
          pod-template-hash: 776c847ddc
      spec:
        containers:
        - image: nginx
```

```
      spec:
        containers:
        - image: nginx
          imagePullPolicy: Always
          name: frontend-container
          resources: {}
          terminationMessagePath: /dev/termination-log
          terminationMessagePolicy: File
        dnsPolicy: ClusterFirst
        restartPolicy: Always
        schedulerName: default-scheduler
        securityContext: {}
        terminationGracePeriodSeconds: 30
  status:
    fullyLabeledReplicas: 2
    observedGeneration: 1
    readyReplicas: 2
    replicas: 2
kind: List
metadata:
  resourceVersion: ""
MacBook-Pro:~ zolatech$ █
```

```
ssh
MacBook-Pro:~ zolatech$ minikube tunnel
✅ Tunnel successfully started

📌 NOTE: Please do not close this terminal as this process must stay alive for the tunnel to be accessible ...

❗ The service/ingress lb-service requires privileged ports to be exposed: [80]
🔑 sudo permission will be asked for it.
🏃 Starting tunnel for service lb-service.
Password:
█
```

← → C ⓘ localhost

Welcome to nginx!

If you see this page, the nginx web server is successfully installed and working. Further configuration is required.

For online documentation and support please refer to nginx.org.
Commercial support is available at nginx.com.

Thank you for using nginx.