🚀Deploying a Microservices Application on a Bare-Metal Kubernetes Cluster

## Table of contents

[Overview 1](#_Toc1478857958)

[Prerequisites 1](#_Toc38290647)

[Application overview 1](#_Toc1793360642)

[Service Mesh Options 2](#_Toc1855677627)

[Microservices & Infrastructure 2](#_Toc1486670242)

[Namespace & Sidecar Setup 2](#_Toc722007314)

[Minikube Installation 3](#_Toc1992294491)

[Kubectl Installation 3](#_Toc1758355298)

[Starting the Minikube Cluster 4](#_Toc1233211941)

[Deploying the Pitstop Application 4](#_Toc408577432)

[Verifying the Services 5](#_Toc952112218)

[Final Remarks 5](#_Toc1846509143)

[References 5](#_Toc1068418272)

## Overview

This document outlines the steps undertaken to successfully deploy the Pitstop microservices application on a bare-metal Kubernetes cluster using Minikube. It includes the setup of essential components, installation steps, verification outputs, and final deployment status.

## Prerequisites

* A Linux-based machine (Ubuntu 22.04 used in this case)
* Sudo access
* Internet connectivity

## Application overview

Pitstop is a microservices-based reference application for managing garage operations, including customer, vehicle, and workshop management. This document outlines a robust deployment strategy of Pitstop on a Kubernetes cluster using Minikube with optional support for service meshes like Istio and Linkerd.

It includes:

* Kubernetes manifests for each microservice
* A Docker Compose-based local setup for testing
* A bash script to deploy Pitstop with or without a service mesh
* Guidance for deploying supporting infrastructure (SQL Server, RabbitMQ, Maildev, etc.)

### Service Mesh Options

Pitstop can be deployed in three modes:

1. **Without service mesh**: Direct communication using Kubernetes services
2. **With Istio**: Enables advanced traffic management, security, and observability
3. **With Linkerd**: Lightweight service mesh focused on simplicity and performance

The start-all.sh script accepts the following flags:

./start-all.sh --nomesh # Plain Kubernetes deployment  
./start-all.sh --istio # Deploy with Istio sidecar injection  
./start-all.sh --linkerd # Deploy with Linkerd proxy injection

The script applies Kubernetes manifests with optional postfixes (e.g., -istio, -linkerd) to control sidecar configurations.

### Microservices & Infrastructure

|  |  |  |
| --- | --- | --- |
| **Service** | **Description** | **Dependencies** |
| **RabbitMQ** | Message broker for inter-service communication | - |
| **SQL Server** | Data persistence for all services | - |
| **Mailserver** (MailDev) | Dummy SMTP server to test outgoing mails | - |
| **Logserver** (Seq) | Centralized logging with structured log support | - |
| **CustomerManagementAPI** | Manages customer profiles | SQL Server, RabbitMQ |
| **VehicleManagementAPI** | Vehicle registration and data | SQL Server, RabbitMQ |
| **WorkshopManagementAPI** | Booking and tracking workshop services | SQL Server, RabbitMQ |
| **AuditLogService** | Tracks audit trails for critical operations | RabbitMQ |
| **InvoiceService** | Generates invoices and sends via mail | SQL Server, Mailserver, RabbitMQ |
| **NotificationService** | Sends notifications (email/SMS) | SQL Server, Mailserver, RabbitMQ |
| **TimeService** | Provides time events for periodic operations | RabbitMQ |
| **WorkshopManagementEventHandler** | Event handler for workshop operations | SQL Server, RabbitMQ |
| **WorkmanManagementAPI** | Manages garage staff and their schedules | SQL Server, RabbitMQ |
| **WebApp** | The main user interface (UI) for garage operations | All APIs |

### Namespace & Sidecar Setup

Namespaces and sidecar behavior are handled with manifest variations.

For Istio:

../istio/disable-default-istio-injection.sh  
kubectl label namespace pitstop istio-injection=enabled

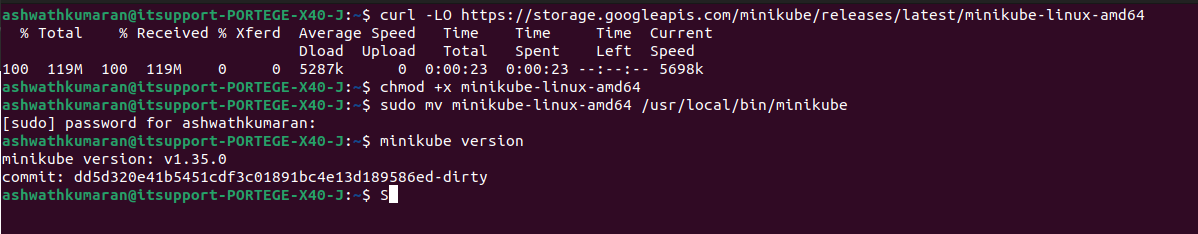
For Linkerd, you can inject using:

linkerd inject deployment.yaml | kubectl apply -f -

## Minikube Installation

Minikube is a lightweight tool that creates a local Kubernetes cluster on your machine, ideal for development and testing. It runs a single-node cluster inside a virtual machine or container. Minikube supports all key Kubernetes features and simplifies learning and experimentation.

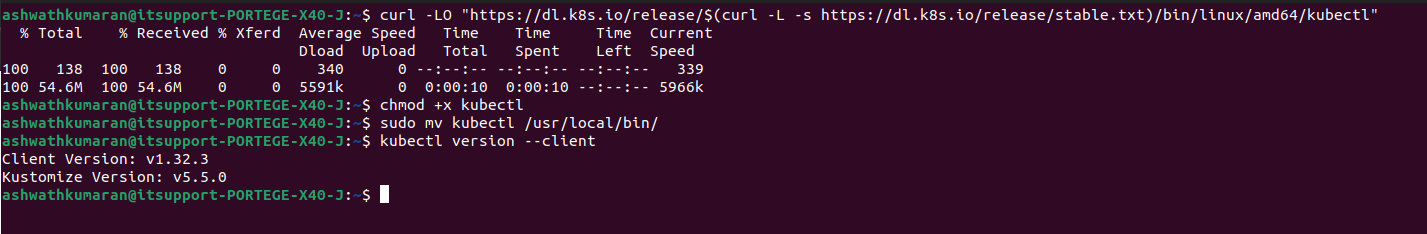
We begin by installing Minikube, which enables a local Kubernetes cluster setup.



## Kubectl Installation

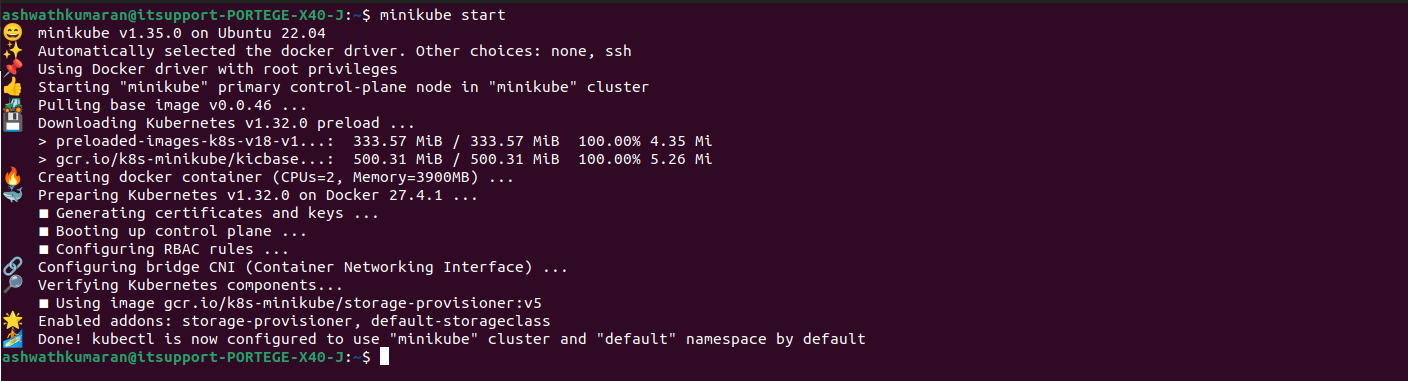
kubectl is the command-line interface for interacting with Kubernetes clusters. It allows users to deploy applications, inspect and manage cluster resources, and view logs. With kubectl, you can control all aspects of your Kubernetes environment efficiently.

We then install kubectl, the command-line tool for interacting with the Kubernetes cluster.



## Starting the Minikube Cluster

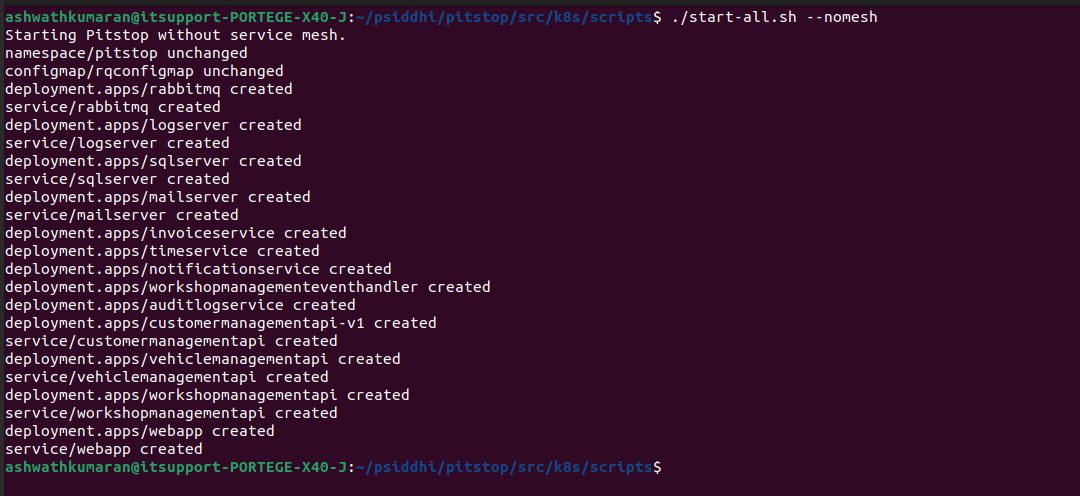
We initialize Minikube to start a local Kubernetes cluster. It is a simple command once executed it then started the control pane in our local machine.



## Deploying the Pitstop Application

Navigate to the Pitstop project directory and execute the deployment script.

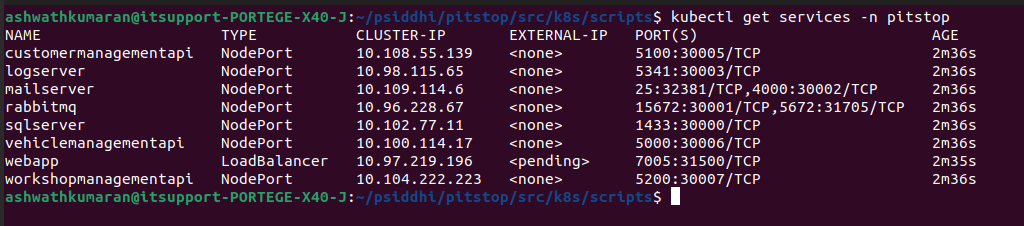
The script sets up all deployments and services required for the Pitstop application in the pitstop namespace.



## Verifying the Services

Check that all services are correctly created and running in the cluster.

All services like customerManagementApi, webapp, sqlserver, and others are listed, including their cluster IPs and ports.



## Final Remarks

* The microservices-based Pitstop application was successfully deployed in a local bare-metal Kubernetes cluster using Minikube.
* All services are accessible via NodePort or LoadBalancer types.
* Minikube's simplicity made the local cluster setup efficient and replicable.

## References

* <https://betterstack.com/community/guides/scaling-docker/kubernetes-getting-started/#kubernetes-core-concepts>
* <https://kubernetes.io/>
* <https://minikube.sigs.k8s.io/docs/start/\>
* <https://kubernetes.io/docs/tasks/tools/install-kubectl-linux/>
* <https://www.youtube.com/watch?v=uUupRagM7m0&list=PL2We04F3Y_41jYdadX55fdJplDvgNGENo>
* <https://github.com/EdwinVW/pitstop/wiki/Run%20the%20application%20on%20Kubernetes>