Are Cluster IP and Pod IP same in Kubernetes ?

No, **Cluster IP** and **Pod IP** are not the same. They serve different purposes and operate at different levels within a Kubernetes cluster. Here's a detailed explanation:

1. Cluster IP

- **What It Is**:
  - The **Cluster IP** is a virtual IP address assigned to a Kubernetes **Service**. It is stable and used for accessing a group of pods that the service manages.
- **Purpose**:
  - To provide a consistent, load-balanced way to access multiple pods running behind a service.
  - Ensures service discovery within the cluster.
- **Scope**:
  - Accessible only within the Kubernetes cluster unless explicitly exposed (e.g., via a NodePort, LoadBalancer, or Ingress).
- **Example**:
  - A service named `my-app-service` might have a Cluster IP `10.96.0.1`. Any pod in the cluster can access it using this IP or the service's DNS name, e.g., `my-app-service.default.svc.cluster.local`.

---

2. Pod IP

- **What It Is**:
  - The **Pod IP** is the IP address assigned to an individual pod by the network plugin (CNI) when the pod is created.
- **Purpose**:
  - To enable direct communication with the pod from other pods or components in the cluster.
  - Each pod has a unique Pod IP within the cluster.
- **Scope**:
  - Pod IPs are ephemeral and tied to the pod's lifecycle. If the pod is deleted and recreated, it gets a new IP.
- **Example**:
  - A pod might have an IP address `192.168.1.15`.

Key Differences

Feature	Cluster IP	Pod IP
Assigned To	A Kubernetes Service	An individual pod
Scope	Stable within the cluster	Ephemeral, tied to pod lifecycle
Purpose	Accessing a group of pods via a Service	Accessing an individual pod directly
Visibility	Internal to the cluster unless exposed	Internal to the cluster
Example	`10.96.0.1`	`192.168.1.15`

Why They Are Different

1. **Abstraction**:
   - Cluster IP abstracts the details of individual pod IPs, enabling you to access services without worrying about pod-specific details.
   - Pod IPs are lower-level and specific to the pod itself.
2. **Stability**:
   - Cluster IP remains constant as long as the service exists.
   - Pod IPs can change when pods are recreated or rescheduled.

Scenario

- A Kubernetes **Service** provides a Cluster IP to allow stable access to multiple pods running behind it.
- Inside the service, requests are routed to the appropriate pod(s) using their Pod IPs.

By distinguishing these concepts, Kubernetes ensures scalability and robustness in managing workloads.