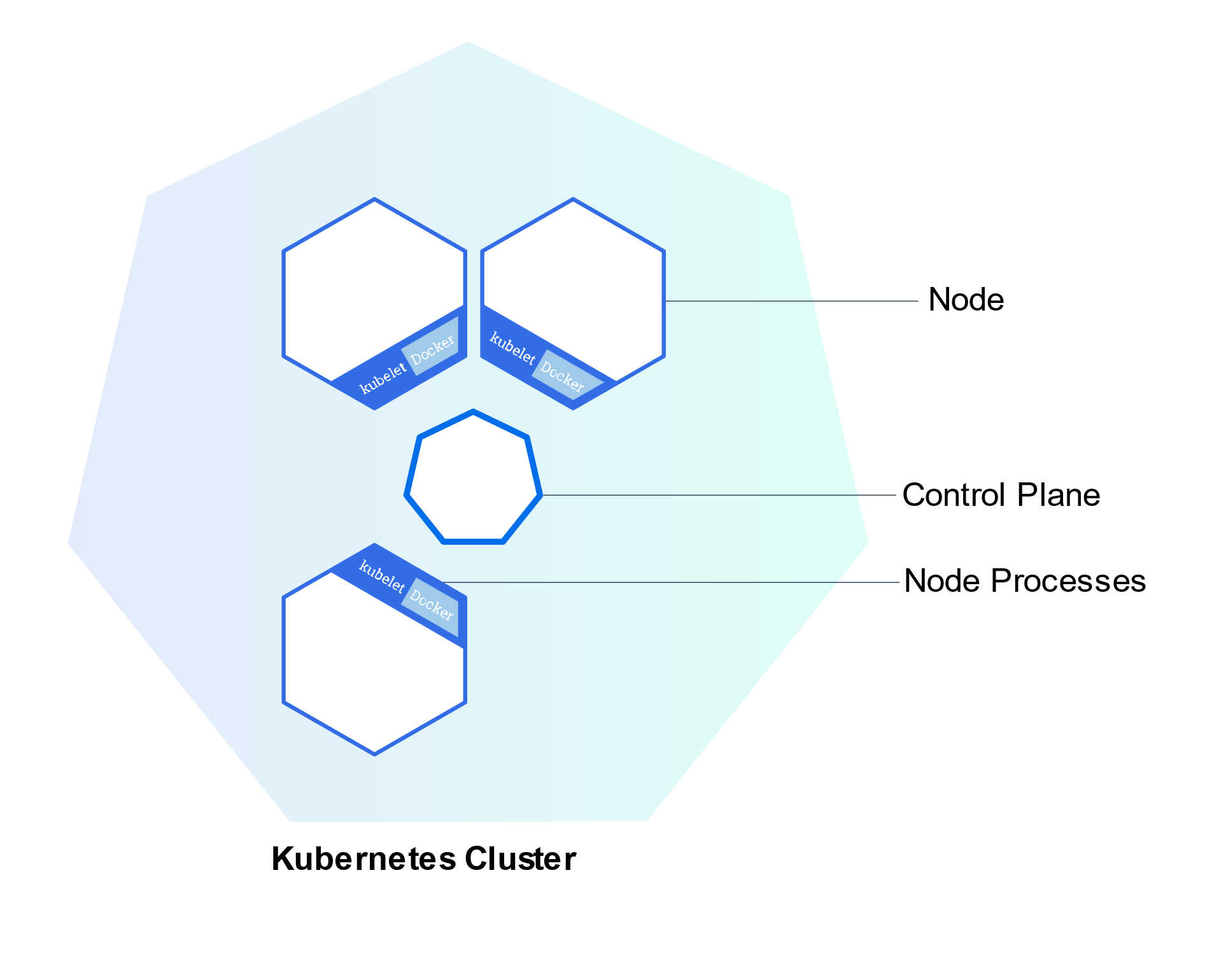
**What is Kubernetes Cluster?**

쿠버네티스는 컴퓨터들을 연결하여 단일 형상으로 동작하도록 컴퓨팅 클러스터를 구성하고 높은 가용성을 제공하도록 조율한다. 사용자는 쿠버네티스의 추상화 개념을 통해 개별 머신에 얽매이지 않고 컨테이너화된 애플리케이션을 클러스터에 배포할 수 있다. 이렇게 새로운 배포 모델을 활용하려면, 애플리케이션을 개별 호스트에 독립적인 방식으로 패키징할 필요가 있다. 즉, 컨테이너화가 필요하다. 예전 배치 모델인 설치형 애플리케이션이 특정 머신의 호스트와 밀접하게 통합되는 패키지 인 것에 비해, 컨테이너화된 애플리케이션은 유연성(Flexible)과 가용성(Available)이 훨씬 높다. 쿠버네티스는 이러한 애플리케이션 컨테이너를 클러스터에 분산 시키고 스케쥴링하는 일을 더욱 효율적으로 자동화한다. 쿠버네티스는 오픈소스 플랫폼이며 운영 수준의 안정성(production-ready)을 제공한다.

쿠버네티스 클러스터는 두 가지 현태의 자원으로 구성된다.

* 컨트롤 플레인은 클러스터를 조율한다.
* 노드는 애플리케이션을 구동하는 작업자(worker)이다.

**Cluster Diagram**



컨트롤 플레인은 클러스터 관리를 담당한다. 컨트롤 플레인은 애플리케이션을 스케쥴링하거나, 애플리케이션의 향상성을 유지하거나, 애플리케이션을 스케일링하고, 새로운 변경사항을 순서대로 반영(rolling out)하는 일과 같은 클러스터 내 모든 활동을 조율한다.

노드는 쿠버네티스 클러스터 내 워커 머신으로 동작하는 VM 또는 물리적 컴퓨터다. 각 노드는 노드를 관리하고 쿠버네티스 컨트롤 플레인과 통신하는 kubelet이라는 에이전트를 갖는다. 노드는 컨테이너 운영을 담당하는 containerd 또는 도커와 같은 툴도 갖는다. 운영 트래픽을 처리하는 쿠버네티스는 최소 세 대의 노드를 가져야한다.

애플리케이션을 쿠버네티스에 배포하기 위해서는, 컨트롤 플레인에 애플리케이션 컨테이너의 구동을 지시하면된다. 그러면 컨트롤 플레인은 컨테이너를 클러스터의 어느 노드에 구동시킬지 스케쥴한다. 노드는 컨트롤 플레인이 제공하는 Kubernetes API를 통해서 컨트롤 플레인과 통신한다. 최종 사용자도 쿠버네티스 API를 사용해 클러스터와 직접 상호작용(interact)할 수 있다.

쿠버네티스 클러스터는 물리 및 가상 머신 모두에 설치 될 수 있다.