What Kubernetes does not do.

Kubernetes is not a conventional, comprehensive PaaS (Platform as a Service) technology. Instead, it offers several broadly applicable capabilities common to PaaS products, such as deployment, scaling, load balancing, and the ability for customers to integrate own logging, monitoring, and alerting solutions. This is because Kubernetes functions at the container level, not the hardware level. But Kubernetes is also not a monolithic system, and these built-in default solutions can be changed. Although Kubernetes offers the fundamental components for constructing developer platforms, it maintains user autonomy and flexibility in critical areas.

Kubernetes:

- Kubernetes allows various types of applications, such as stateless, stateful, and data-processing workloads, without imposing restrictions on app choices. If an application runs smoothly in a container, it should perform well on Kubernetes.
- Kubernetes does not build or deploy your program's source code. The way Continuous Integration, Delivery, and Deployment (CI/CD) processes are set up depends on both technical needs and the specific practices and preferences within an organization.
- Application-level services like databases, caches, middleware, data processing frameworks, and cluster storage systems are not included in Kubernetes. However, apps can use portable techniques like the Open Service Broker to access these components, which can be operated on Kubernetes.
- Does not control logging, monitoring, or alerting solutions. It has options to collect and export metrics.
- Kubernetes does not have a required configuration language or system that users must use. It instead allows users to choose and implement their preferred configuration methods and tools.
- Kubernetes exposes a declarative API which means that users define the desired state of the system, Kubernetes makes the actual state match the desired state. Users have the flexibility in how you describe the desired state, e.g. a YAML or JSON file
- Kubernetes is not able to manage every aspect of the underlying machines or servers in your cluster. Managing and orchestrating containers and the apps that run inside of them is the focus of Kubernetes.
- Kubernetes comprises a set of independent, composable control processes that continuously drive the current state towards the provided desired state. This means that Kubernetes does not technically follow the definition of orchestration. Orchestration is a defined workflow, Kubernetes handles processes in a more fluid to continuously move towards the desired state.

What other orchestration tools are available other than Kubernetes?

Docker Swarm

Amazon ECS Elastic Container Service

Apache Mesos

Openstack Magnum

Nomad

Rancher

Google Kubernetes Engine

Azure Kubernetes Service (AKS)

Redhat OpenShift