Deploying software to a Kubernetes cluster is only the start of operating an application long term. Administrators and developers need to understand the resource consumption patterns and behaviors of their applications, with the goal of providing a scalable and reliable service. This responsibility falls into the hands of the metrics server, a cluster-wide aggregator of resource usage data.

The kubectl tool integrates with the metrics server by inspecting resource consumption of nodes and Pods. In this lab, you will use the top command to check on resource consumption on the node and the Pods that have been set up.

Prerequisite Skills
Comfort building and deploying server-based applications
Familiarity with concepts like load balancers and network storage will be useful, though not required
Experience with Linux, containers, and container runtimes such as Docker Engine

Node Resource Consumption
Once the metrics server has been started, it usually takes a little while to collect the initial set of metrics. You can use the top command to see the list of resources in use by either nodes or Pods.

First, we'll check the resource consumption on all nodes in the cluster by providing the nodes subcommand:

kubectl top nodes

```
$ kubectl top nodes
NAME           CPU(cores)   CPU%   MEMORY(bytes)   MEMORY%
controlplane   129m         6%     1286Mi          33%
node01         176m         8%     1079Mi          28%
$
```

The top command communicates with the metrics server and will display the total CPU and memory in use by the nodes in terms of both absolute units (e.g., cores) and percentage of available resources (e.g., total number of cores).

To query the information for just a single node, append the name of the specific node.

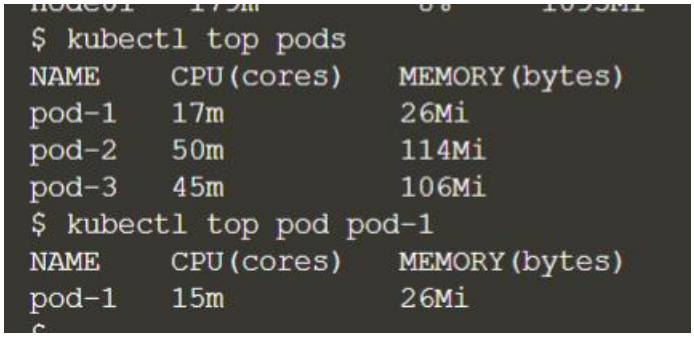
kubectl top node node01

```
$ kubectl top node node01
NAME     CPU(cores)   CPU%   MEMORY(bytes)   MEMORY%
node01   179m         8%     1093Mi          28%
$
```

Pod Resource Consumption
Similarly, you can query similar information on the Pod level. The command top pods will show all Pods and their resource usage. By default it only displays Pods in the current namespace, but you can add the --all-namespaces flag to see resource usage by all Pods in the cluster. kubectl top pods

To query the information for just a single Pod, append the name of the specific Pod. Here, we are checking the resource consumption of Pod pod-1.

kubectl top pod pod-1



kubectl top is a command in the Kubernetes command-line tool kubectl that allows you to retrieve resource utilization information for nodes or pods in a Kubernetes cluster.

To get the resource utilization information for nodes, you can use the following command:
kubectl top node

This will give you an overview of the CPU and memory usage for each node in your cluster.
To get the resource utilization information for pods, you can use the following command:
kubectl top pod

This will give you an overview of the CPU and memory usage for each pod in your cluster.

Note that the kubectl top command requires the metrics-server add-on to be installed and running in your Kubernetes cluster.

kubectl apply -f https://github.com/kubernetes-sigs/metrics-server/releases/latest/download/components.yaml

Once installed, it may take a few minutes for metrics to be collected and displayed by the kubectl top command.

In Kubernetes, a namespace is a virtual cluster that provides a way to partition resources within a cluster. A namespace is a way to create a logical separation of resources in a cluster.

With namespaces, you can have multiple virtual clusters running within a single physical cluster, with each virtual cluster having its own resources, such as pods, services, and storage volumes. Namespaces allow you to organize and manage resources based on their application, team, or environment.

For example, you might create separate namespaces for development, staging, and production environments. This way, you can have different versions of an application running in different namespaces, and you can manage each environment separately.

By default, Kubernetes provides a namespace called "default", where resources are created if no namespace is specified. You can create additional namespaces using the "kubectl create namespace" command, and you can view the list of namespaces in a cluster using the "kubectl get namespaces" command.

The kubectl tool can interact with the metrics server to check CPU and memory resource consumption. You can use the top nodes command to render node-level metrics, or the top pods command to render metrics for the Pods in the current namespace.