

2.2. Student Handout

Google Kubernetes Engine (GKE) Student Handout

Introduction to Google Kubernetes Engine (GKE)

Objective: Gain a foundational understanding of Google Kubernetes Engine (GKE) and its role in managing Kubernetes clusters on Google Cloud Platform (GCP).

What is GKE?

Google Kubernetes Engine (GKE) is a managed service on Google Cloud that simplifies running and managing Kubernetes clusters.

What is Kubernetes?

Kubernetes is an open-source platform for automating the deployment, scaling, and management of containerized applications.

Key Features of Kubernetes:

1. **Automated Deployment:** Deploy applications consistently across different environments.
 2. **Scaling:** Automatically scale applications up or down based on demand.
 3. **Self-Healing:** Automatically restart failed containers and reschedule them on healthy nodes.
-

Why Choose GKE for Orchestration?

1. Managed Service

- **Cluster Management:** Google handles the setup and management of the underlying infrastructure.
- **Automatic Updates:** GKE keeps your Kubernetes clusters updated with the latest versions.

- **Scaling:** Automatically scales applications based on demand.

2. Integration with Google Cloud Services

- **Google Cloud Storage:** Seamlessly store and retrieve data.
- **Google Cloud Pub/Sub:** Facilitate messaging between services.
- **Google Cloud Monitoring and Logging:** Monitor application health and performance.

3. Cost Efficiency

- **Pay-as-you-go Pricing:** Only pay for the resources you use.
 - **Autoscaling:** Automatically adjust resources based on demand to minimize costs.
-

GCP Setup and Navigation

Step 1: Create a GCP Account

- Visit the [Google Cloud website](#) to create an account.
- Utilize the free tier with \$300 in credits for initial exploration.

Step 2: Navigate the GCP Console

- Use the navigation menu to access services like Compute Engine, Cloud Storage, and Kubernetes Engine.
 - Navigate to the **Kubernetes Engine** section to create and manage clusters.
-

Creating a Kubernetes Cluster on GKE

Step 1: Enable the Kubernetes Engine API

- Access the **Kubernetes Engine** section in the GCP Console.
- Click on **Enable API** to activate the service.

Step 2: Create a Cluster

- Choose a cluster name and region.
- Select the number of nodes for your cluster.

- Click **Create** to initiate the cluster setup.
-

Configuring the Cluster

1. Node Pools

- Create node pools with specific configurations to handle different workloads.

2. Autoscaling

- Enable autoscaling to automatically adjust the number of nodes based on demand.
-

Connecting to Your Cluster Using kubectl

Step 1: Install kubectl

- Follow the [Kubernetes documentation](#) for installation instructions.

Step 2: Connect to Your Cluster

- Use the command:

```
gcloud container clusters get-credentials [CLUSTER_NAME] --zone [ZONE]
```

- This retrieves cluster credentials and configures kubectl.
-

Deploying a Simple Web Application on GKE

Step 1: Create a Deployment

- Write a manifest file to define the desired state of your application.
- Example manifest for an Nginx web server:

```
apiVersion: apps/v1
```

```
kind: Deployment
```

```
metadata:
```

```
name: web-app
```

```
spec:
```

```
replicas: 3
```

```
selector:
```

```
matchLabels:
```

```
app: web-app
```

```
template:
```

```
metadata:
```

```
labels:
```

```
app: web-app
```

```
spec:
```

```
containers:
```

```
- name: web-app
```

```
image: nginx
```

```
ports:
```

```
- containerPort: 80
```

Step 2: Apply the Manifest

- Deploy the application using:

```
kubectl apply -f deployment.yaml
```

Exposing the Application with a LoadBalancer

Step 1: Create a Service

- Example manifest for a LoadBalancer Service:

```
apiVersion: v1

kind: Service

metadata:

name: web-app-service

spec:

type: LoadBalancer

selector:

app: web-app

ports:

- protocol: TCP

port: 80

targetPort: 80
```

Step 2: Apply the Manifest

- Create the Service using:

```
kubectl apply -f service.yaml
```

Monitoring and Logging with Google Cloud's Built-in Tools

1. Monitoring

- Track application performance and set up alerts for issues like high CPU usage.

2. Logging

- Collect logs from applications and infrastructure for troubleshooting and insights.
-

Stackdriver Logging and Monitoring for Kubernetes Clusters

1. Stackdriver Monitoring

- Provides real-time metrics for Kubernetes clusters, such as CPU and memory usage.

2. Stackdriver Logging

- Collects logs from Kubernetes clusters for troubleshooting and insights.
-

Conclusion

Google Kubernetes Engine (GKE) is a robust, managed service for running Kubernetes clusters on Google Cloud. It simplifies deploying, scaling, and managing containerized applications, allowing you to focus on building great software. GKE's integration with Google Cloud services provides real-time insights into application health and performance.

For further questions, feel free to reach out!