

(1) Should we move to GKE? Why?

*[What are the pros and cons of GKE in relation to **scaling and API use**? And how does this differ from Azure features pros and cons in scaling?] mohammed/shahd*

Google Kubernetes Engine is a management system for Docker containers and container clusters within Google's public cloud services. It serves as a way to deploy, scale and manage kubernetes. It allows for easy cluster creation, load balancing, auto upgrades, logging monitoring and most importantly auto-scaling. Based on the demands of your workloads, GKE will autoscale the cluster's node pools. This is extremely helpful especially when clients are increasing in your company, as it will reduce the manual workload of scaling, and can increase the availability of your workloads when needed. GKE's cluster autoscaler automatically resizes, so there is no need for manual work. All you need to do is specify the minimum and maximum node pool size. Additionally, Cloud Functions service runs code written in Node.js, Java, Python, and C++ which are all APIs that are used in the company. In terms of API use, GKE has two primary API integrations, the google kubernetes engine API and simply the kubernetes engine API. Both integrations have unique features that can be indulged by the user. The google kubernetes engine API is primarily used to configure the users cluster on their google cloud, therefore this api allows the user to create or delete a cluster, configure cluster-level networking, update kubernetes, add or remove or modify node pools in the cluster, setting machine types/node images, configure the location on where cluster is deployed. However, the Kubernetes API allows a heavier approach towards containerized applications and workloads. This includes features such as deploying an application,

scaling an application, configuring intra-cluster networking, configuring containers and pods and controlling when pods are evicted or restarted.

Azure features also allow use of docker containers containing Node.js. Java. Python, and C++ code. Scalability in Azure features is slightly different than GKE, as it has a maximum number of instances unlike GKE which autoscales based off of minimum and maximum nodes. The greatest possible number of instances in Azure functions is 200 in the consumption plan which automatically scales based off of the number of incoming trigger events.

References:

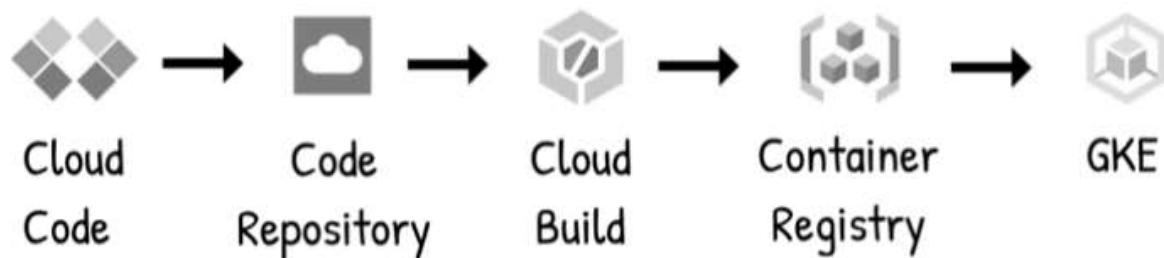
<https://cloud.google.com/kubernetes-engine>

<https://cloud.google.com/kubernetes-engine/docs/quickstart>

What are the list of services that we need to activate on GCP? *shahd*

Moving to the GKE engine would require certain services to be activated on Google Cloud Platform. First, **Cloud Code** is used to write the code that is required. It will then be sent to a source **Code Repository**, which can be done by activating the Code Repository. After the code is in the repository, the build process will take place in **Cloud Build**. After building, the container images will need to be stored. Activating the **Container Registry** service on GCP will help in storing the container images, and then it is ready to deploy on **GKE**. Cloud Console may

also be a service to use on GCP as you can create the cluster there, or you can run it on the command line. This means that we will have to activate Google Kubernetes Engine (GKE), as it will be a useful tool in deployment. Google Cloud Storage is also a service that is provided by GCP which is designed to store large data sets, and also can use SQL or Cloud SQL for database storage. This can be very helpful for data analytics as it provides a large storage area for the databases needed for the company.



References:

https://www.youtube.com/watch?v=R15M1CzgEH4&t=242s&ab_channel=GoogleCloudTech