

[Google Kubernetes Engine \(GKE\)](https://cloud.google.com/kubernetes-engine/) (<https://cloud.google.com/kubernetes-engine/>)
[Documentation](https://cloud.google.com/kubernetes-engine/docs/) (<https://cloud.google.com/kubernetes-engine/docs/>) [Guides](#)

Managing Batch on GKE clusters

Beta

This product or feature is in a pre-release state and might change or have limited support. For more information, see the [product launch stages](https://cloud.google.com/products/#product-launch-stages) (<https://cloud.google.com/products/#product-launch-stages>).

This page shows you how to create and manage Batch on GKE clusters.

Note: You can report bugs and request features at [google-batch-discuss](https://groups.google.com/d/forum/google-batch-discuss) (<https://groups.google.com/d/forum/google-batch-discuss>). For the latest news join [google-batch-announce](https://groups.google.com/g/google-batch-announce) (<https://groups.google.com/g/google-batch-announce>).

Before you begin

To prepare for this task, perform the following steps:

- Ensure that you have enabled the Google Kubernetes Engine API.

ENABLE GOOGLE KUBERNETES ENGINE API ([HTTPS://CONSOLE.CLOUD.GOOGLE.COM/APIs/LIBRARY](https://console.cloud.google.com/apis/library))

- Ensure that you have installed the [Cloud SDK](https://cloud.google.com/sdk/downloads) (<https://cloud.google.com/sdk/downloads>).
- Set your default [project ID](https://support.google.com/cloud/answer/6158840) (<https://support.google.com/cloud/answer/6158840>):

```
gcloud config set project [PROJECT_ID]
```



- If you are working with zonal clusters, set your default [compute zone](https://cloud.google.com/compute/docs/zones#available) (<https://cloud.google.com/compute/docs/zones#available>):

```
gcloud config set compute/zone [COMPUTE_ZONE]
```



- If you are working with regional clusters, set your default [compute region](https://cloud.google.com/compute/docs/zones#available) (<https://cloud.google.com/compute/docs/zones#available>):

```
gcloud config set compute/region [COMPUTE_REGION]
```



- Update `gcloud` to the latest version:

```
gcloud components update
```



★ **Note:** You can override these default settings in `gcloud` commands using the `--project`, `--zone`, and `--region` operational flags.

In Beta, Batch on GKE (Batch) supports only regional clusters. You must create a regional cluster

(https://cloud.google.com/kubernetes-engine/docs/how-to/creating-a-regional-cluster#creating_a_regional_cluster)

and enable Workload Identity.

(<https://cloud.google.com/kubernetes-engine/docs/how-to/workload-identity>).

Note: Cluster scaling is built into Batch. Do not enable the GKE autoscaler when creating a cluster because it does not work with Batch.

Run the following command to create a regional cluster with Workload Identity enabled:

```
gcloud beta container clusters create [CLUSTER_NAME] \  
  --region [COMPUTE_REGION] \  
  --node-locations [COMPUTE_ZONE] \  
  --num-nodes 1 \  
  --machine-type n1-standard-8 \  
  --release-channel regular \  
  --enable-stackdriver-kubernetes \  
  --identity-namespace=[PROJECT_ID].svc.id.goog \  
  --enable-ip-alias
```



Note: Batch on GKE cluster scaling supports only one location. If you specify more than one, your Pods will crash.

Configuring identity and access management

1. Bind your account as the project owner:

```
gcloud projects add-iam-policy-binding [PROJECT_ID] \  
--member user:[EMAIL] --role=roles/owner
```

2. Create a custom role with read permissions on GKE clusters:

```
gcloud iam roles create BatchUser --project [PROJECT_ID] \  
--title GKEClusterReader --permissions container.clusters.get --stage BETA 2>&1
```

where:

- **[PROJECT_ID]** is your Project ID.
- GKEClusterReader is the title of the role.

3. Create a ClusterRoleBinding in your cluster to allow Batch to create Kubernetes Roles:

```
kubectl create clusterrolebinding cluster-admin-binding-[EMAIL] \  
--clusterrole=cluster-admin --user [EMAIL]
```

where **[EMAIL]** is your email address.

4. Create a Google service account:

```
gcloud iam service-accounts create kbatch-controllers-gcloud-sa --display-name  
kbatch-controllers-gcloud-service-account
```

5. Create a Kubernetes service account:

```
kubectl create serviceaccount --namespace kube-system kbatch-controllers-k8s-sa
```

6. Add the following IAM policy bindings:

```
gcloud projects add-iam-policy-binding [PROJECT_ID] \  
--member serviceAccount:kbatch-controllers-gcloud-sa@[PROJECT_ID].iam.gservicea  
--role=roles/container.clusterAdmin
```

```
gcloud projects add-iam-policy-binding [PROJECT_ID] \  
--member serviceAccount:kbatch-controllers-gcloud-sa@[PROJECT_ID].iam.gservicea  
--role=roles/compute.admin
```

```
gcloud projects add-iam-policy-binding [PROJECT_ID] \  
--member serviceAccount:kbatch-controllers-gcloud-sa@[PROJECT_ID].iam.gservicea
```

```
--role=roles/iam.serviceAccountUser
```

```
gcloud iam service-accounts add-iam-policy-binding \  
--role roles/iam.workloadIdentityUser \  
--member "serviceAccount:[PROJECT_ID].svc.id.goog[kube-system/kbatch-controller
```

where **[PROJECT_ID]** is your Project ID.

7. Add the iam.gke.io/gcp-service-account annotation to the Kubernetes service account:

```
kubectl annotate serviceaccount --namespace kube-system kbatch-controllers-k8s-  
iam.gke.io/gcp-service-account=kbatch-controllers-gcloud-sa@[PROJECT].iam.gser
```

Enabling GPUs

If you want to run GPU jobs, run the following command to ensure the proper drivers are installed when GPU nodes are created:

```
kubectl apply -f https://raw.githubusercontent.com/GoogleCloudPlatform/container-eng
```

Installing Batch on GKE

Note: You can go to [google-batch-discuss](https://groups.google.com/d/forum/google-batch-discuss) (<https://groups.google.com/d/forum/google-batch-discuss>) to reach out to Googlers and the community for assistance.

To install Batch, perform the following steps:

1. Download the Batch release from [GitHub](https://github.com/GoogleCloudPlatform/Kbatch/tree/master/releases) (<https://github.com/GoogleCloudPlatform/Kbatch/tree/master/releases>).
2. Extract the tar file:

```
tar zxvf kbatch-[VERSION].tar.gz
```

3. Change to the kbatch directory:

```
cd kbatch
```

4. Add your info to the Config file:

```
vi config/kbatch-config.yaml
```

```
...
ClusterName: [CLUSTER_NAME]
ClusterLocation : [COMPUTE_REGION]
ProjectID: [PROJECT_ID]
Recommender:
  Locations:
    # Note: Only one zone is supported in the Locations list here.
    - [COMPUTE_ZONE]
Actuator:
...
```

5. Create configmaps:

```
kubectl create configmap --from-file config/kbatch-config.yaml -n kube-system
```

6. Create autoscaler machine types:

```
gcloud compute machine-types list --filter="zone:[COMPUTE_ZONE]" --format json
```

```
kubectl create configmap --from-file ./machine_types.json -n kube-system
```

7. Install the Batch custom resource definitions and components:

```
kubectl apply -f install/01-crds.yaml
```

```
kubectl apply -f install/02-admission.yaml
```

```
kubectl apply -f install/03-controller.yaml
```

8. Enable ksub to use your user credentials for API access:

```
gcloud auth application-default login
```

9. Initialize Ksub:

```
./ksub --config --create-default
```

10. Add the default values for **projectID**, **clusterName**, and, if you are not operating in the default namespace, **namespace** in .ksubrc:

```
vi ~/.ksubrc
```

Verifying the Batch installation

1. Verify kbatch-admission Pods are RUNNING:

```
kubectl get pods -n kube-system --selector=app=kbatch-admission
```

2. Verify kbatch-controllers Pods are RUNNING:

```
kubectl get pods -n kube-system --selector=control-plane=kbatch-controllers
```

Once you've verified the Batch installation, you can [run the sample jobs](#)

(https://cloud.google.com/kubernetes-engine/docs/how-to/batch/running-job#running_sample_jobs)

Managing Batch on GKE versions

You can upgrade, downgrade, and uninstall Batch.

Upgrading Batch

Note: Upgrading to the Batch Beta from Alpha is not supported. You must install Batch on a different cluster.

To upgrade to a new minor or patch version, run the following commands:

1. Delete the current admission and controllers .yaml files:

```
kubectl delete -f kbatch-[CURRENT_VERSION]/install/02-admission.yaml \  
kubectl delete -f kbatch-[CURRENT_VERSION]/install/03-controllers.yaml
```

2. Apply the new admission and controllers .yaml files:

```
kubectl apply -f kbatch-[NEW_VERSION]/install/02-admission.yaml \  
kubectl apply -f kbatch-[NEW_VERSION]/install/03-controllers.yaml
```



To upgrade to a new major version, either install the new version on a new cluster, or follow the steps in [Uninstalling Batch](#) (#uninstalling_batch) then install the new major version.

Downgrading Batch

You can only rollback to the previous minor or patch version.

To rollback to a previous version, run the following commands:

1. Delete the current admission and controllers .yaml files:

```
kubectl delete -f kbatch-[CURRENT_VERSION]/install/02-admission.yaml \  
kubectl delete -f kbatch-[CURRENT_VERSION]/install/03-controllers.yaml
```



2. Apply the new admission and controllers .yaml files:

```
kubectl apply -f kbatch-[OLD_VERSION]/install/02-admission.yaml \  
kubectl apply -f kbatch-[OLD_VERSION]/install/03-controllers.yaml
```



Uninstalling Batch

To uninstall Batch, perform the following steps:

1. Verify which version of Batch versions you are running by checking the image tags:

```
kubectl get deployment kbatch-admission -n kube-system -o jsonpath="{..image}" \  
kubectl get statefulset kbatch-controllers -n kube-system -o jsonpath="{..image}"
```



2. Delete the installation bundle from your cluster:

```
kubectl delete -f kbatch-[VERSION]/install/
```



Warning: This command also deletes all Batch data, including resources, in your cluster.

Debugging Batch on GKE using Stackdriver

Batch uses Prometheus as the monitoring tool. You can view your kbatch-controller-service metrics from stackdriver monitoring. The metrics that are generated by Batch services are considered as external metrics (https://cloud.google.com/monitoring/api/metrics_other#prometheus) in Stackdriver.

Custom metrics are a chargeable feature of Stackdriver Monitoring and there could be costs for the custom metrics. For more information on pricing, see Stackdriver Pricing (<https://cloud.google.com/stackdriver/pricing>).

Before you begin

Configuring identity and access management

1. Create a Google service account:

```
gcloud iam service-accounts create kbatch-monitoring-gcloud-sa \  
--display-name kbatch-monitoring-gcloud-service-account
```

2. Create a Kubernetes service account:

```
kubectl create serviceaccount --namespace kube-system kbatch-monitoring-k8s-sa
```

3. Add the following IAM policy bindings:

```
gcloud projects add-iam-policy-binding [PROJECT_ID] \  
--member serviceAccount:kbatch-monitoring-gcloud-sa@[PROJECT_ID].iam.gserviceac  
--role=roles/monitoring.metricWriter
```

```
gcloud projects add-iam-policy-binding [PROJECT_ID] \  
--member serviceAccount:kbatch-monitoring-gcloud-sa@[PROJECT_ID].iam.gserviceac  
--role=roles/monitoring.viewer
```

```
gcloud iam service-accounts add-iam-policy-binding \  
--role roles/iam.workloadIdentityUser \  
--member "serviceAccount:[PROJECT_ID].svc.id.goog[kube-system/kbatch-monitoring
```

where **[PROJECT_ID]** is your Project ID.

4. Add the `iam.gke.io/gcp-service-account` annotation to the Kubernetes service account:

```
kubectl annotate serviceaccount --namespace kube-system kbatch-monitoring-k8s-s  
iam.gke.io/gcp-service-account=kbatch-monitoring-gcloud-sa@[PROJECT].iam.gserv
```

5. Get the admin tools:

```
git clone https://github.com/GoogleCloudPlatform/Kbatch.git
```

6. Go to the monitoring directory:

```
cd admintools/monitoring
```

Deploy prometheus service

1. To deploy the prometheus service run the following command:

```
kubectl apply -f prometheus.yaml
```

2. To validate the Prometheus deployment, run the following command:

```
kubectl get pod -n kube-system | grep 'kbatch-prometheus'
```

The output is similar to this:

```
kbatch-prometheus-deployment-97bc6b97b-m4q9h      1/1      Running      0
```

Installing the Stackdriver collector

Next, deploy the sidecar container as the Stackdriver collector. Sidecar exports the Prometheus metrics to Stackdriver.

Note: After deploying the sidecar container, you will be charged for the custom metrics.

1. To deploy the stackdriver collector run the following command:

```
sh ./setup_metrics_export_to_sd.sh
```

2. To validate the Stackdriver collector installation, run the following command:

```
kubectl -n kube-system get deployment kbatch-prometheus-deployment -o=go-template
```

When the Prometheus sidecar is successfully installed, the output of the script lists the image used from the container registry.

```
sidecar exists. Image: gcr.io/kbatch-images/stackdriver-prometheus-sidecar:0.6.
```

Otherwise, the output of the script shows:

```
stackdriver-prometheus-sidecar does not exist.
```

Viewing metrics

1. Go to Metrics Explorer.

[GO TO METRICS EXPLORER \(HTTPS://APP.GOOGLE.STACKDRIVER.COM/\)](https://app.google.stackdriver.com/)

2. Go to Resources > Metrics Explorer. In the **Find resource type and metric** field, select the one with prefix external/prometheus/.

For example, you might select external/prometheus/kbatch_scheduling_dep.

You can add multiple metrics in one Workspace.

Disable the Stackdriver collector

To disable the sidecar container run the following command from the kbatch directory.

```
sh ./disable_metrics_export_to_sd.sh
```

Clean up

To stop running Batch services in a GKE cluster, run the following commands:

```
kubectl delete deployment kbatch-admission --namespace=kube-system  
kubectl delete statefulset kbatch-controllers --namespace=kube-system
```

To delete the GKE cluster that has Batch installed, run the following command:

```
gcloud container clusters delete [CLUSTER_NAME] --region [REGION]
```



To delete the Filestore instance, run the following command:

```
gcloud beta filestore instances delete [FILESTORE_INSTANCE_ID] \  
  --project=[PROJECT_ID] --location=[FILESTORE_ZONE]
```



where **[FILESTORE_INSTANCE_ID]** is your Filestore Instance ID, **[PROJECT_ID]** is your Project ID, and **[FILESTORE_ZONE]** is your zone.

To delete the project that has Batch installed, run the following command:

```
gcloud projects delete [PROJECT_ID]
```



What's next

- [Run sample jobs](https://cloud.google.com/kubernetes-engine/docs/how-to/batch/running-job) (https://cloud.google.com/kubernetes-engine/docs/how-to/batch/running-job)
- [Manage Batch on GKE resources](https://cloud.google.com/kubernetes-engine/docs/how-to/batch/managing-resources) (https://cloud.google.com/kubernetes-engine/docs/how-to/batch/managing-resources)
- [Manage Batch on GKE users](https://cloud.google.com/kubernetes-engine/docs/how-to/batch/managing-users) (https://cloud.google.com/kubernetes-engine/docs/how-to/batch/managing-users)
- [Learn more about Batch on GKE and its resources](https://cloud.google.com/kubernetes-engine/docs/concepts/batch-reference) (https://cloud.google.com/kubernetes-engine/docs/concepts/batch-reference)

Except as otherwise noted, the content of this page is licensed under the [Creative Commons Attribution 4.0 License](https://creativecommons.org/licenses/by/4.0/) (https://creativecommons.org/licenses/by/4.0/), and code samples are licensed under the [Apache 2.0 License](https://www.apache.org/licenses/LICENSE-2.0) (https://www.apache.org/licenses/LICENSE-2.0). For details, see our [Site Policies](https://developers.google.com/terms/site-policies) (https://developers.google.com/terms/site-policies). Java is a registered trademark of Oracle and/or its affiliates.

Last updated November 15, 2019.