xml

<https://github.com/spring-projects/spring-integration-extensions/blob/master/samples/aws-s3/src/main/resources/META-INF/spring/integration/spring-integration-context.xml>

IMp

<https://stackoverflow.com/questions/33121980/prevent-duplicate-files-polling-using-inboundchanneladapter-annotaion?rq=1>

With proccessor :-

<https://dzone.com/articles/event-driven-microservices-using-spring-cloud-stre>

<https://stackoverflow.com/questions/45125015/create-stream-with-one-source-two-parallel-processors-and-one-sink-in-spring-cl?rq=1>

"headers.targetsystem-folder + '/' + headers.file\_name"

<https://github.com/spring-cloud-stream-app-starters/aws-s3>

<https://docs.spring.io/spring-cloud-stream-app-starters/docs/current/reference/htmlsingle/#spring-cloud-stream-modules-aws-s3-source>

dataflow:> stream create --name ticktock --definition "time --server.port=9000 | log"

GCP ->

<https://cloud.google.com/appengine/docs/flexible/java/using-cloud-storage>

<https://github.com/spring-cloud/spring-cloud-gcp/tree/master/spring-cloud-gcp-storage/src/main/java/org/springframework/cloud/gcp/storage/integration/inbound>

<https://cloud.google.com/java/getting-started/using-cloud-storage>

<https://github.com/googleapis/google-cloud-java/blob/master/google-cloud-examples/src/main/java/com/google/cloud/examples/storage/snippets/StorageSnippets.java>

<https://github.com/spring-cloud/spring-cloud-gcp/tree/master/spring-cloud-gcp-autoconfigure/src/main/java/org/springframework/cloud/gcp/autoconfigure/storage>

<https://github.com/eugenp/tutorials/blob/master/google-cloud/src/main/java/com/baeldung/google/cloud/storage/GoogleCloudStorage.java>

<https://cloud.google.com/storage/docs/reference/libraries#client-libraries-usage-java>

<https://cloud.google.com/appengine/docs/standard/java/googlecloudstorageclient/read-write-to-cloud-storage>

<https://github.com/spring-cloud/spring-cloud-gcp/tree/72d746486fedd19b16ead3cdbbcfb48c3e90c3a9/spring-cloud-gcp-storage>

<https://github.com/spring-projects/spring-integration-java-dsl/wiki/spring-integration-java-dsl-reference>

IMP links

<https://github.com/spring-cloud/spring-cloud-stream-samples/blob/master/sink-samples>

<https://www.programcreek.com/java-api-examples/?code=spring-cloud-stream-app-starters/jdbc/jdbc-master/spring-cloud-starter-stream-source-jdbc/src/main/java/org/springframework/cloud/stream/app/jdbc/source/JdbcSourceConfiguration.java#>

<https://github.com/Dhpandey/SpringCloudPack>

<https://stackoverflow.com/questions/53045991/bind-rabbitmq-consumer-using-spring-cloud-stream-to-rabbitmq-producer>

<https://github.com/spring-cloud/spring-cloud-dataflow-samples>

IMP

<https://labnotes.panderalabs.com/spring-cloud-data-flow-and-docker-kubernetes-99a19f2dbab3>

<https://github.com/eugenp/tutorials/tree/master/spring-cloud-data-flow>

<https://dzone.com/articles/building-and-testing-message-driven-microservices>

<https://dzone.com/articles/event-driven-microservices-using-spring-cloud-stre>

<https://www.baeldung.com/spring-cloud-stream>

<https://www.javainuse.com/spring/cloud-stream-rabbitmq-2>

<https://github.com/spring-cloud/spring-cloud-dataflow-server-kubernetes/blob/master/spring-cloud-dataflow-server-kubernetes-docs/src/main/asciidoc/getting-started.adoc>

Rabbitmq -it can reliably handle 20,000 messages per second with the largest documented deployment

If you are doing more than 100,000 messages per second- go for Kafka



integrating Spring Boot Applications with Messaging Systems like Apache Kafka and RabbitMQ. If you look at these examples these required a lot of configuration code which was Broker specific. For example in case of RabbitMQ integration with Spring Boot we had to write code to create AmqpTemplate Template and Bindings. So if tomorrow the Messaging System changes we will also need to make application code changes.

Spring Cloud helps solve this problem using Spring Cloud Stream. Using Spring Cloud Stream we can develop applications where we do not need to specify the implementation details of the messaging system we want to use. We just need to specify the required binding dependencies and Spring Cloud Stream will the integrate the messaging systems to Spring Boot Application.

if you need to run a single container in Kubernetes, then you need to create a Pod for that container. At the same time, a Pod can contain more than one container.

A Kubernetes pod is a group of containers that are deployed together on the same host. If you frequently deploy single containers, you can generally replace the word "pod" with "container"

A Cluster in a collection of multiple nodes which communicates with each other to perform set of operation at high available rates.

To use Kubernetes, you need to be able to build containers using a Google Cloud container and you need to have a trustworthy place to store images of your containers using container registry. What's important about these tools is that you can manage them using the same tools you're already using. Google Cloud container builder and container registry keep code and images in Google Cloud Storage and you can control access permissions using IAM, just as you can for other GCP resources.

why people choose Kubernetes engine.

* Because it's a managed service, it relieves developers from the operational details of running a cluster.
* It has built-in logging and monitoring capabilities, and Google keeps Kubernetes engine refreshed with successive versions of Kubernetes.
* The Kubernetes engine team periodically performs automatic upgrades of your cluster master to newer stable versions of Kubernetes, and you can enable automatic node upgrades too.
* Cloud Storage is not a file system because each of your objects in Cloud Storage has a URL. Each feels like a file in a lot of ways and that's okay to use the word file informally to describe your objects but still it's not a file system. (it is a object storage)
* You would not use Cloud Storage as the root file system of your Linux box instead Cloud Storage is comprised of buckets you create and configure and use to hold your storage objects.
* The storage objects are immutable, which means that you do not edit them in place but instead you create new versions.
* Cloud Storage always encrypts your data on the server side before it is written to disk and you don't pay extra for that. Also by default, data in-transit is encrypted using HTTPS.

Cloud Storage objects are immutable. You can turn on object versioning on your buckets if you wantrabbit

Official link-[https://docs.spring.io/spring-cloud-dataflow-server-kubernetes/docs/current/ref HYPERLINK "https://docs.spring.io/spring-cloud-dataflow-server-kubernetes/docs/current/reference/htmlsingle/" HYPERLINK "https://docs.spring.io/spring-cloud-dataflow-server-kubernetes/docs/current/reference/htmlsingle/" HYPERLINK "https://docs.spring.io/spring-cloud-dataflow-server-kubernetes/docs/current/reference/htmlsingle/" HYPERLINK "https://docs.spring.io/spring-cloud-dataflow-server-kubernetes/docs/current/reference/htmlsingle/" HYPERLINK "https://docs.spring.io/spring-cloud-dataflow-server-kubernetes/docs/current/reference/htmlsingle/" HYPERLINK "https://docs.spring.io/spring-cloud-dataflow-server-kubernetes/docs/current/reference/htmlsingle/" HYPERLINK "https://docs.spring.io/spring-cloud-dataflow-server-kubernetes/docs/current/reference/htmlsingle/" HYPERLINK "https://docs.spring.io/spring-cloud-dataflow-server-kubernetes/docs/current/reference/htmlsingle/" HYPERLINK "https://docs.spring.io/spring-cloud-dataflow-server-kubernetes/docs/current/reference/htmlsingle/" HYPERLINK "https://docs.spring.io/spring-cloud-dataflow-server-kubernetes/docs/current/reference/htmlsingle/" HYPERLINK "https://docs.spring.io/spring-cloud-dataflow-server-kubernetes/docs/current/reference/htmlsingle/" HYPERLINK "https://docs.spring.io/spring-cloud-dataflow-server-kubernetes/docs/current/reference/htmlsingle/" HYPERLINK "https://docs.spring.io/spring-cloud-dataflow-server-kubernetes/docs/current/reference/htmlsingle/" HYPERLINK "https://docs.spring.io/spring-cloud-dataflow-server-kubernetes/docs/current/reference/htmlsingle/" HYPERLINK "https://docs.spring.io/spring-cloud-dataflow-server-kubernetes/docs/current/reference/htmlsingle/" HYPERLINK "https://docs.spring.io/spring-cloud-dataflow-server-kubernetes/docs/current/reference/htmlsingle/" HYPERLINK "https://docs.spring.io/spring-cloud-dataflow-server-kubernetes/docs/current/reference/htmlsingle/" HYPERLINK "https://docs.spring.io/spring-cloud-dataflow-server-kubernetes/docs/current/reference/htmlsingle/" HYPERLINK "https://docs.spring.io/spring-cloud-dataflow-server-kubernetes/docs/current/reference/htmlsingle/" HYPERLINK "https://docs.spring.io/spring-cloud-dataflow-server-kubernetes/docs/current/reference/htmlsingle/" HYPERLINK "https://docs.spring.io/spring-cloud-dataflow-server-kubernetes/docs/current/reference/htmlsingle/" HYPERLINK "https://docs.spring.io/spring-cloud-dataflow-server-kubernetes/docs/current/reference/htmlsingle/" HYPERLINK "https://docs.spring.io/spring-cloud-dataflow-server-kubernetes/docs/current/reference/htmlsingle/" HYPERLINK "https://docs.spring.io/spring-cloud-dataflow-server-kubernetes/docs/current/reference/htmlsingle/" HYPERLINK "https://docs.spring.io/spring-cloud-dataflow-server-kubernetes/docs/current/reference/htmlsingle/" HYPERLINK "https://docs.spring.io/spring-cloud-dataflow-server-kubernetes/docs/current/reference/htmlsingle/" HYPERLINK "https://docs.spring.io/spring-cloud-dataflow-server-kubernetes/docs/current/reference/htmlsingle/" HYPERLINK "https://docs.spring.io/spring-cloud-dataflow-server-kubernetes/docs/current/reference/htmlsingle/" HYPERLINK "https://docs.spring.io/spring-cloud-dataflow-server-kubernetes/docs/current/reference/htmlsingle/" HYPERLINK "https://docs.spring.io/spring-cloud-dataflow-server-kubernetes/docs/current/reference/htmlsingle/" HYPERLINK "https://docs.spring.io/spring-cloud-dataflow-server-kubernetes/docs/current/reference/htmlsingle/"e HYPERLINK "https://docs.spring.io/spring-cloud-dataflow-server-kubernetes/docs/current/reference/htmlsingle/" HYPERLINK "https://docs.spring.io/spring-cloud-dataflow-server-kubernetes/docs/current/reference/htmlsingle/" HYPERLINK "https://docs.spring.io/spring-cloud-dataflow-server-kubernetes/docs/current/reference/htmlsingle/" HYPERLINK "https://docs.spring.io/spring-cloud-dataflow-server-kubernetes/docs/current/reference/htmlsingle/" HYPERLINK "https://docs.spring.io/spring-cloud-dataflow-server-kubernetes/docs/current/reference/htmlsingle/" HYPERLINK "https://docs.spring.io/spring-cloud-dataflow-server-kubernetes/docs/current/reference/htmlsingle/" HYPERLINK "https://docs.spring.io/spring-cloud-dataflow-server-kubernetes/docs/current/reference/htmlsingle/" HYPERLINK "https://docs.spring.io/spring-cloud-dataflow-server-kubernetes/docs/current/reference/htmlsingle/" HYPERLINK "https://docs.spring.io/spring-cloud-dataflow-server-kubernetes/docs/current/reference/htmlsingle/" HYPERLINK "https://docs.spring.io/spring-cloud-dataflow-server-kubernetes/docs/current/reference/htmlsingle/" HYPERLINK "https://docs.spring.io/spring-cloud-dataflow-server-kubernetes/docs/current/reference/htmlsingle/" HYPERLINK "https://docs.spring.io/spring-cloud-dataflow-server-kubernetes/docs/current/reference/htmlsingle/" HYPERLINK "https://docs.spring.io/spring-cloud-dataflow-server-kubernetes/docs/current/reference/htmlsingle/" HYPERLINK "https://docs.spring.io/spring-cloud-dataflow-server-kubernetes/docs/current/reference/htmlsingle/" HYPERLINK "https://docs.spring.io/spring-cloud-dataflow-server-kubernetes/docs/current/reference/htmlsingle/" HYPERLINK "https://docs.spring.io/spring-cloud-dataflow-server-kubernetes/docs/current/reference/htmlsingle/" HYPERLINK "https://docs.spring.io/spring-cloud-dataflow-server-kubernetes/docs/current/reference/htmlsingle/" HYPERLINK "https://docs.spring.io/spring-cloud-dataflow-server-kubernetes/docs/current/reference/htmlsingle/" HYPERLINK "https://docs.spring.io/spring-cloud-dataflow-server-kubernetes/docs/current/reference/htmlsingle/" HYPERLINK "https://docs.spring.io/spring-cloud-dataflow-server-kubernetes/docs/current/reference/htmlsingle/" HYPERLINK "https://docs.spring.io/spring-cloud-dataflow-server-kubernetes/docs/current/reference/htmlsingle/" HYPERLINK "https://docs.spring.io/spring-cloud-dataflow-server-kubernetes/docs/current/reference/htmlsingle/" HYPERLINK "https://docs.spring.io/spring-cloud-dataflow-server-kubernetes/docs/current/reference/htmlsingle/" HYPERLINK "https://docs.spring.io/spring-cloud-dataflow-server-kubernetes/docs/current/reference/htmlsingle/" HYPERLINK "https://docs.spring.io/spring-cloud-dataflow-server-kubernetes/docs/current/reference/htmlsingle/" HYPERLINK "https://docs.spring.io/spring-cloud-dataflow-server-kubernetes/docs/current/reference/htmlsingle/" HYPERLINK "https://docs.spring.io/spring-cloud-dataflow-server-kubernetes/docs/current/reference/htmlsingle/" HYPERLINK "https://docs.spring.io/spring-cloud-dataflow-server-kubernetes/docs/current/reference/htmlsingle/" HYPERLINK "https://docs.spring.io/spring-cloud-dataflow-server-kubernetes/docs/current/reference/htmlsingle/" HYPERLINK "https://docs.spring.io/spring-cloud-dataflow-server-kubernetes/docs/current/reference/htmlsingle/" HYPERLINK "https://docs.spring.io/spring-cloud-dataflow-server-kubernetes/docs/current/reference/htmlsingle/"re HYPERLINK "https://docs.spring.io/spring-cloud-dataflow-server-kubernetes/docs/current/reference/htmlsingle/" HYPERLINK "https://docs.spring.io/spring-cloud-dataflow-server-kubernetes/docs/current/reference/htmlsingle/" HYPERLINK "https://docs.spring.io/spring-cloud-dataflow-server-kubernetes/docs/current/reference/htmlsingle/" HYPERLINK "https://docs.spring.io/spring-cloud-dataflow-server-kubernetes/docs/current/reference/htmlsingle/" HYPERLINK "https://docs.spring.io/spring-cloud-dataflow-server-kubernetes/docs/current/reference/htmlsingle/" HYPERLINK "https://docs.spring.io/spring-cloud-dataflow-server-kubernetes/docs/current/reference/htmlsingle/" HYPERLINK "https://docs.spring.io/spring-cloud-dataflow-server-kubernetes/docs/current/reference/htmlsingle/" HYPERLINK "https://docs.spring.io/spring-cloud-dataflow-server-kubernetes/docs/current/reference/htmlsingle/" HYPERLINK "https://docs.spring.io/spring-cloud-dataflow-server-kubernetes/docs/current/reference/htmlsingle/" HYPERLINK "https://docs.spring.io/spring-cloud-dataflow-server-kubernetes/docs/current/reference/htmlsingle/" HYPERLINK "https://docs.spring.io/spring-cloud-dataflow-server-kubernetes/docs/current/reference/htmlsingle/" HYPERLINK "https://docs.spring.io/spring-cloud-dataflow-server-kubernetes/docs/current/reference/htmlsingle/" HYPERLINK "https://docs.spring.io/spring-cloud-dataflow-server-kubernetes/docs/current/reference/htmlsingle/" HYPERLINK "https://docs.spring.io/spring-cloud-dataflow-server-kubernetes/docs/current/reference/htmlsingle/" HYPERLINK "https://docs.spring.io/spring-cloud-dataflow-server-kubernetes/docs/current/reference/htmlsingle/" HYPERLINK "https://docs.spring.io/spring-cloud-dataflow-server-kubernetes/docs/current/reference/htmlsingle/" HYPERLINK "https://docs.spring.io/spring-cloud-dataflow-server-kubernetes/docs/current/reference/htmlsingle/" HYPERLINK "https://docs.spring.io/spring-cloud-dataflow-server-kubernetes/docs/current/reference/htmlsingle/" HYPERLINK "https://docs.spring.io/spring-cloud-dataflow-server-kubernetes/docs/current/reference/htmlsingle/" HYPERLINK "https://docs.spring.io/spring-cloud-dataflow-server-kubernetes/docs/current/reference/htmlsingle/" HYPERLINK "https://docs.spring.io/spring-cloud-dataflow-server-kubernetes/docs/current/reference/htmlsingle/" HYPERLINK "https://docs.spring.io/spring-cloud-dataflow-server-kubernetes/docs/current/reference/htmlsingle/" HYPERLINK "https://docs.spring.io/spring-cloud-dataflow-server-kubernetes/docs/current/reference/htmlsingle/" HYPERLINK "https://docs.spring.io/spring-cloud-dataflow-server-kubernetes/docs/current/reference/htmlsingle/" HYPERLINK "https://docs.spring.io/spring-cloud-dataflow-server-kubernetes/docs/current/reference/htmlsingle/" HYPERLINK "https://docs.spring.io/spring-cloud-dataflow-server-kubernetes/docs/current/reference/htmlsingle/" HYPERLINK "https://docs.spring.io/spring-cloud-dataflow-server-kubernetes/docs/current/reference/htmlsingle/" HYPERLINK "https://docs.spring.io/spring-cloud-dataflow-server-kubernetes/docs/current/reference/htmlsingle/" HYPERLINK "https://docs.spring.io/spring-cloud-dataflow-server-kubernetes/docs/current/reference/htmlsingle/" HYPERLINK "https://docs.spring.io/spring-cloud-dataflow-server-kubernetes/docs/current/reference/htmlsingle/" HYPERLINK "https://docs.spring.io/spring-cloud-dataflow-server-kubernetes/docs/current/reference/htmlsingle/"n HYPERLINK "https://docs.spring.io/spring-cloud-dataflow-server-kubernetes/docs/current/reference/htmlsingle/" HYPERLINK "https://docs.spring.io/spring-cloud-dataflow-server-kubernetes/docs/current/reference/htmlsingle/" HYPERLINK "https://docs.spring.io/spring-cloud-dataflow-server-kubernetes/docs/current/reference/htmlsingle/" HYPERLINK "https://docs.spring.io/spring-cloud-dataflow-server-kubernetes/docs/current/reference/htmlsingle/" HYPERLINK "https://docs.spring.io/spring-cloud-dataflow-server-kubernetes/docs/current/reference/htmlsingle/" HYPERLINK "https://docs.spring.io/spring-cloud-dataflow-server-kubernetes/docs/current/reference/htmlsingle/" HYPERLINK "https://docs.spring.io/spring-cloud-dataflow-server-kubernetes/docs/current/reference/htmlsingle/" HYPERLINK "https://docs.spring.io/spring-cloud-dataflow-server-kubernetes/docs/current/reference/htmlsingle/" HYPERLINK "https://docs.spring.io/spring-cloud-dataflow-server-kubernetes/docs/current/reference/htmlsingle/" HYPERLINK "https://docs.spring.io/spring-cloud-dataflow-server-kubernetes/docs/current/reference/htmlsingle/" HYPERLINK "https://docs.spring.io/spring-cloud-dataflow-server-kubernetes/docs/current/reference/htmlsingle/" HYPERLINK "https://docs.spring.io/spring-cloud-dataflow-server-kubernetes/docs/current/reference/htmlsingle/" HYPERLINK "https://docs.spring.io/spring-cloud-dataflow-server-kubernetes/docs/current/reference/htmlsingle/" HYPERLINK "https://docs.spring.io/spring-cloud-dataflow-server-kubernetes/docs/current/reference/htmlsingle/" HYPERLINK "https://docs.spring.io/spring-cloud-dataflow-server-kubernetes/docs/current/reference/htmlsingle/" HYPERLINK "https://docs.spring.io/spring-cloud-dataflow-server-kubernetes/docs/current/reference/htmlsingle/" HYPERLINK "https://docs.spring.io/spring-cloud-dataflow-server-kubernetes/docs/current/reference/htmlsingle/" HYPERLINK "https://docs.spring.io/spring-cloud-dataflow-server-kubernetes/docs/current/reference/htmlsingle/" HYPERLINK "https://docs.spring.io/spring-cloud-dataflow-server-kubernetes/docs/current/reference/htmlsingle/" HYPERLINK "https://docs.spring.io/spring-cloud-dataflow-server-kubernetes/docs/current/reference/htmlsingle/" HYPERLINK "https://docs.spring.io/spring-cloud-dataflow-server-kubernetes/docs/current/reference/htmlsingle/" HYPERLINK "https://docs.spring.io/spring-cloud-dataflow-server-kubernetes/docs/current/reference/htmlsingle/" HYPERLINK "https://docs.spring.io/spring-cloud-dataflow-server-kubernetes/docs/current/reference/htmlsingle/" HYPERLINK "https://docs.spring.io/spring-cloud-dataflow-server-kubernetes/docs/current/reference/htmlsingle/" HYPERLINK "https://docs.spring.io/spring-cloud-dataflow-server-kubernetes/docs/current/reference/htmlsingle/" HYPERLINK "https://docs.spring.io/spring-cloud-dataflow-server-kubernetes/docs/current/reference/htmlsingle/" HYPERLINK "https://docs.spring.io/spring-cloud-dataflow-server-kubernetes/docs/current/reference/htmlsingle/" HYPERLINK "https://docs.spring.io/spring-cloud-dataflow-server-kubernetes/docs/current/reference/htmlsingle/" HYPERLINK "https://docs.spring.io/spring-cloud-dataflow-server-kubernetes/docs/current/reference/htmlsingle/" HYPERLINK "https://docs.spring.io/spring-cloud-dataflow-server-kubernetes/docs/current/reference/htmlsingle/" HYPERLINK "https://docs.spring.io/spring-cloud-dataflow-server-kubernetes/docs/current/reference/htmlsingle/"ce/htmlsingle/#kubernetes-getting-started](https://docs.spring.io/spring-cloud-dataflow-server-kubernetes/docs/current/reference/htmlsingle/)

---------------------------------------------------------

**Importing dependencies**

---------------------------------------------------------

# Install Kubernetes commandline utility [Local bash]

gcloud components install kubectl

---------------------------------------------------------

**Running commands**

---------------------------------------------------------

# Set the GCP Default Project [Local bash]

gcloud config set project stgcloudcoe

# Set the Default region for deploying your containers [Local bash]

gcloud config set compute/zone us-east1

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

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):

gcloud config set compute/region ***[COMPUTE\_REGION]***

Create a network

First, create a network for your cluster. The following command creates a network, my-net-1:

gcloud compute networks create my-net-1 \

--subnet-mode custom

# Set a VPC Network for cluster [Cloud Shell]

gcloud compute --project=synt-int-cloudcoe-rnd-prd networks create synt-vpc-scdf-network-1 --description="VPC Network for SCDF Kubernetes cluster" --subnet-mode=custom

Created [<https://www.googleapis.com/compute/v1/projects/synt-int-cloudcoe-rnd-prd/global/networks/synt-vpc-scdf-network-1>].

NAME                     SUBNET\_MODE  BGP\_ROUTING\_MODE  IPV4\_RANGE  GATEWAY\_IPV4

synt-vpc-scdf-network-1  CUSTOM       REGIONAL

Instances on this network will not be reachable until firewall rules

are created. As an example, you can allow all internal traffic between

instances as well as SSH, RDP, and ICMP by running:

$ gcloud compute firewall-rules create <FIREWALL\_NAME> --network synt-vpc-scdf-network-1 --allow tcp,udp,icmp --source-ranges <IP\_RANGE>

$ gcloud compute firewall-rules create <FIREWALL\_NAME> --network synt-vpc-scdf-network-1 --allow tcp:22,tcp:3389,icmp

Create a subnet and secondary ranges

Next, create a subnet, my-subnet-1, in the my-net-1 network, with secondary ranges my-pods-1 for Pods and my-services-1 for Services:

gcloud compute networks subnets create my-subnet-1 \

--network my-net-1\

--region us-central1 \

--range 192.168.0.0/20 \

--secondary-range my-pods-1=10.4.0.0/14,my-services-1=10.0.32.0/20 \

--enable-private-ip-google-access

# Set a VPC Subnet for cluster [Cloud Shell]

gcloud compute --project=synt-int-cloudcoe-rnd-prd networks subnets create synt-vpc-scdf-subnet-1 --network=synt-vpc-scdf-network-1 --region=us-east1 --range=10.160.0.0/20

Created [<https://www.googleapis.com/compute/v1/projects/synt-int-cloudcoe-rnd-prd/regions/asia-south1/subnetworks/synt-vpc-scdf-subnet-1>].

NAME                    REGION       NETWORK                  RANGE

synt-vpc-scdf-subnet-1  asia-south1  synt-vpc-scdf-network-1  10.160.0.0/20

# Set the name of container cluster [Cloud Shell]

gcloud beta container --project "synt-int-cloudcoe-rnd-prd" clusters create "synt-gke-scdf-cluster-1" --region "us-east1" --username "admin" --cluster-version "1.13.7-gke.8" --machine-type "n1-standard-1" --image-type "COS" --disk-type "pd-standard" --disk-size "100" --scopes "<https://www.googleapis.com/auth/devstorage.read_only>","<https://www.googleapis.com/auth/logging.write>","<https://www.googleapis.com/auth/monitoring>","<https://www.googleapis.com/auth/servicecontrol>","<https://www.googleapis.com/auth/service.management.readonly>","<https://www.googleapis.com/auth/trace.append>" --num-nodes "2" --enable-cloud-logging --enable-cloud-monitoring --enable-ip-alias --network "projects/synt-int-cloudcoe-rnd-prd/global/networks/synt-vpc-scdf-network-1" --subnetwork "projects/synt-int-cloudcoe-rnd-prd/regions/us-east1/subnetworks/synt-vpc-scdf-subnet-1" --default-max-pods-per-node "110" --addons HorizontalPodAutoscaling,HttpLoadBalancing --enable-autoupgrade --enable-autorepair --enable-autoscaling

WARNING: Starting in 1.12, new clusters will not have a client certificate issued. You can manually enable (or disable) the issuance of the client certificate using the `--[no-]issue-client-certificate` flag.

WARNING: Starting in 1.12, default node pools in new clusters will have their legacy Compute Engine instance metadata endpoints disabled by default. To create a cluster with legacy instance metadata endpoints disabled in the default node pool, run `clusters create` with the flag `--metadata disable-legacy-endpoints=true`.

This will enable the autorepair feature for nodes. Please see <https://cloud.google.com/kubernetes-engine/docs/node-auto-repair> for more information on node autorepairs.

This will enable the autoupgrade feature for nodes. Please see <https://cloud.google.com/kubernetes-engine/docs/node-management> for more information on node autoupgrades.

Creating cluster standard-cluster-1 in asia-south1... Cluster is being health-checked (master is healthy)...done.

Created [<https://container.googleapis.com/v1beta1/projects/synt-int-cloudcoe-rnd-prd/zones/asia-south1/clusters/standard-cluster-1>].

To inspect the contents of your cluster, go to: <https://console.cloud.google.com/kubernetes/workload_/gcloud/asia-south1/standard-cluster-1?project=synt-int-cloudcoe-rnd-prd>

kubeconfig entry generated for standard-cluster-1.

NAME                LOCATION     MASTER\_VERSION  MASTER\_IP       MACHINE\_TYPE   NODE\_VERSION  NUM\_NODES  STATUS

standard-cluster-1  asia-south1  1.11.7-gke.4    35.200.182.184  n1-standard-1  1.11.7-gke.4  6          RUNNING

# Get Credentials for gke cluster [Cloud Shell] Get credentials, so that you can use kubectl to access the cluster:

gcloud container clusters get-credentials synt-gke-scdf-cluster-1 --region us-east1

Fetching cluster endpoint and auth data.

kubeconfig entry generated for standard-cluster-1.

# Git Clone Deployment files [Cloud Shell]

$ git clone <https://github.com/spring-cloud/spring-cloud-dataflow-server-kubernetes>

# Git Clone Deployment files [Cloud Shell]

$cd spring-cloud-dataflow-server-kubernetes

# Change directory to the git-clone-repo [Local Bash]

cd F:\stg-cloud-repo\SpringCloudDataFlow\spring-cloud-dataflow-server-kubernetes

# Change directory to the git-clone-repo [Cloud Shell]

$cd spring-cloud-dataflow-server-kubernetes

/\*Optional

# Start RabbitMQ service [Cloud Shell]

kubectl create -f src/kubernetes/rabbitmq/

deployment.extensions/rabbitmq created

service/rabbitmq created

# Verify RabbitMQ deployment [Cloud Shell]

kubectl get all -l app=rabbitmq

NAME                            READY     STATUS    RESTARTS   AGE

pod/rabbitmq-76df66dc8c-p8zvk   1/1       Running   0          3m

NAME               TYPE        CLUSTER-IP      EXTERNAL-IP   PORT(S)    AGE

service/rabbitmq   ClusterIP   10.228.11.219   <none>        5672/TCP   3m

NAME                       DESIRED   CURRENT   UP-TO-DATE   AVAILABLE   AGE

deployment.apps/rabbitmq   1         1         1            1           3m

NAME                                  DESIRED   CURRENT   READY     AGE

replicaset.apps/rabbitmq-76df66dc8c   1         1         1         3m

 \*/

# Start Kafka service [Cloud Shell]

kubectl create -f src/kubernetes/kafka/

deployment.extensions/kafka-broker created

service/kafka created

deployment.extensions/kafka-zk created

service/kafka-zk created

# Verify Kafka deployment [Cloud Shell]

kubectl get all -l app=kafka

NAME                                READY     STATUS    RESTARTS   AGE

pod/kafka-broker-696786c8f7-ksls2   1/1       Running   0          10m

NAME               TYPE        CLUSTER-IP      EXTERNAL-IP   PORT(S)                      AGE

service/kafka      ClusterIP   10.228.11.201   <none>        9092/TCP                     10m

service/kafka-zk   ClusterIP   10.228.6.212    <none>        2181/TCP,2888/TCP,3888/TCP   10m

NAME                           DESIRED   CURRENT   UP-TO-DATE   AVAILABLE   AGE

deployment.apps/kafka-broker   1         1         1            1           10m

deployment.apps/kafka-zk       1         1         1            1           10m

NAME                                      DESIRED   CURRENT   READY     AGE

replicaset.apps/kafka-broker-696786c8f7   1         1         1         10m

You can use kubectl get all -l app=kafka to verify that the deployment, pod, and service resources are running. You can use kubectl delete all -l app=kafka to clean up afterwards.

# Start MySQL service [Cloud Shell]

kubectl create -f src/kubernetes/mysql/

deployment.extensions/mysql created

persistentvolumeclaim/mysql created

secret/mysql created

service/mysql created

# Verify MySQL deployment [Cloud Shell]

kubectl get all -l app=mysql

NAME                        READY     STATUS    RESTARTS   AGE

pod/mysql-f878678df-zw598   1/1       Running   0          51s

NAME            TYPE        CLUSTER-IP     EXTERNAL-IP   PORT(S)    AGE

service/mysql   ClusterIP   10.91.4.31 <none>        3306/TCP   50s

NAME                    DESIRED   CURRENT   UP-TO-DATE   AVAILABLE   AGE

deployment.apps/mysql   1         1         1            1           51s

NAME                              DESIRED   CURRENT   READY     AGE

replicaset.apps/mysql-f878678df   1         1         1         51s

# Start Redis service [Cloud Shell]

kubectl create -f src/kubernetes/redis/

deployment.extensions/redis created

service/redis created

# Verify Redis deployment [Cloud Shell]

kubectl get all -l app=redis

NAME                         READY     STATUS    RESTARTS   AGE

pod/redis-748db48b4f-7zgrl   1/1       Running   0          44s

NAME            TYPE        CLUSTER-IP     EXTERNAL-IP   PORT(S)    AGE

service/redis   ClusterIP   10.228.12.36   <none>        6379/TCP   44s

NAME                    DESIRED   CURRENT   UP-TO-DATE   AVAILABLE   AGE

deployment.apps/redis   1         1         1            1           46s

NAME                               DESIRED   CURRENT   READY     AGE

replicaset.apps/redis-748db48b4f   1         1         1         46s

Deploy the Metrics Collector.

The Metrics Collector provides message rates for all deployed stream applications. These message rates are visible in the Dashboard UI. Run one of the following commands (depending on your message broker rabbitMQ or KAFKA) to start the Metrics Collector:

/\* Optional

# Deploy RabbitMQ Metrics Collector [Cloud Shell]

kubectl create -f src/kubernetes/metrics/metrics-deployment-rabbit.yaml

deployment.apps/metrics created

 \*/

# Deploy Kafka Metrics Collector [Cloud Shell]

kubectl create -f src/kubernetes/metrics/metrics-deployment-kafka.yaml

deployment.apps/metrics created

# Deploy Metrics Collector Service [Cloud Shell] Create the metrics service:

kubectl create -f src/kubernetes/metrics/metrics-svc.yaml

service/metrics created

# Verify Metrics Collector deployment [Cloud Shell]

kubectl get all -l app=metrics

NAME                           READY     STATUS    RESTARTS   AGE

pod/metrics-6d87c65f79-5hbnx   1/1       Running   0          2m

NAME              TYPE        CLUSTER-IP     EXTERNAL-IP   PORT(S)   AGE

service/metrics   ClusterIP   10.228.13.74   <none>        80/TCP    1m

NAME                      DESIRED   CURRENT   UP-TO-DATE   AVAILABLE   AGE

deployment.apps/metrics   1         1         1            1           2m

NAME                                 DESIRED   CURRENT   READY     AGE

replicaset.apps/metrics-6d87c65f79   1         1         1         2m

You can use kubectl get all -l app=metrics to verify that the deployment, pod, and service resources are running. You can use kubectl delete all -l app=metrics to clean up afterwards.

Optionally, you can deploy [Skipper](http://cloud.spring.io/spring-cloud-skipper/) to leverage the features of upgrading and rolling back Streams, since Data Flow delegates to Skipper for those features

# Deploy Skipper service [Cloud Shell] (optional, recommended)

kubectl create -f src/kubernetes/skipper/skipper-deployment.yaml

deployment.apps/skipper created

# Start Skipper service [Cloud Shell]

kubectl create -f src/kubernetes/skipper/skipper-svc.yaml

service/skipper created

# Verify Skipper deployment [Cloud Shell]

kubectl get all -l app=skipper

NAME              TYPE           CLUSTER-IP    EXTERNAL-IP   PORT(S)        AGE

service/skipper   LoadBalancer   10.228.2.56   <pending>     80:31487/TCP   45s

NAME                      DESIRED   CURRENT   UP-TO-DATE   AVAILABLE   AGE

deployment.apps/skipper   1         0         0            0           1m

NAME                                 DESIRED   CURRENT   READY     AGE

replicaset.apps/skipper-6ddddd5497   1         0         0         1m

# Create Cluster-Admin binding to IAM user [Cloud Shell]

kubectl create clusterrolebinding kamana\_kanjani-cluster-admin-binding --clusterrole=cluster-admin [--user=kamana.kanjani@atos.net](mailto:--user=kshitij.lipare@atos.net)

kubectl create clusterrolebinding kshitij\_lipare-cluster-admin-binding --clusterrole=cluster-admin [--user=kshitij.lipare@atos.net](mailto:--user=kshitij.lipare@atos.net)

clusterrolebinding.rbac.authorization.k8s.io/kshitij\_lipare-cluster-admin-binding created

# Create a ClusterRoleBinding for user1, user2, and group1 using the cluster-admin ClusterRole

kubectl create clusterrolebinding cluster-admin [-- HYPERLINK "https://www.mankier.com/1/kubectl-create-clusterrolebinding" HYPERLINK "https://www.mankier.com/1/kubectl-create-clusterrolebinding" HYPERLINK "https://www.mankier.com/1/kubectl-create-clusterrolebinding" HYPERLINK "https://www.mankier.com/1/kubectl-create-clusterrolebinding" HYPERLINK "https://www.mankier.com/1/kubectl-create-clusterrolebinding" HYPERLINK "https://www.mankier.com/1/kubectl-create-clusterrolebinding" HYPERLINK "https://www.mankier.com/1/kubectl-create-clusterrolebinding" HYPERLINK "https://www.mankier.com/1/kubectl-create-clusterrolebinding" HYPERLINK "https://www.mankier.com/1/kubectl-create-clusterrolebinding" HYPERLINK "https://www.mankier.com/1/kubectl-create-clusterrolebinding" HYPERLINK "https://www.mankier.com/1/kubectl-create-clusterrolebinding" HYPERLINK "https://www.mankier.com/1/kubectl-create-clusterrolebinding" HYPERLINK "https://www.mankier.com/1/kubectl-create-clusterrolebinding" HYPERLINK "https://www.mankier.com/1/kubectl-create-clusterrolebinding" HYPERLINK "https://www.mankier.com/1/kubectl-create-clusterrolebinding" HYPERLINK "https://www.mankier.com/1/kubectl-create-clusterrolebinding" HYPERLINK "https://www.mankier.com/1/kubectl-create-clusterrolebinding" HYPERLINK "https://www.mankier.com/1/kubectl-create-clusterrolebinding" HYPERLINK "https://www.mankier.com/1/kubectl-create-clusterrolebinding" HYPERLINK "https://www.mankier.com/1/kubectl-create-clusterrolebinding" HYPERLINK "https://www.mankier.com/1/kubectl-create-clusterrolebinding" HYPERLINK "https://www.mankier.com/1/kubectl-create-clusterrolebinding" HYPERLINK "https://www.mankier.com/1/kubectl-create-clusterrolebinding" HYPERLINK "https://www.mankier.com/1/kubectl-create-clusterrolebinding" HYPERLINK "https://www.mankier.com/1/kubectl-create-clusterrolebinding" HYPERLINK "https://www.mankier.com/1/kubectl-create-clusterrolebinding" HYPERLINK "https://www.mankier.com/1/kubectl-create-clusterrolebinding" HYPERLINK "https://www.mankier.com/1/kubectl-create-clusterrolebinding" HYPERLINK "https://www.mankier.com/1/kubectl-create-clusterrolebinding" HYPERLINK "https://www.mankier.com/1/kubectl-create-clusterrolebinding" HYPERLINK "https://www.mankier.com/1/kubectl-create-clusterrolebinding"clusterrole](https://www.mankier.com/1/kubectl-create-clusterrolebinding)=cluster-admin [--user](https://www.mankier.com/1/kubectl-create-clusterrolebinding)=user1 [--user](https://www.mankier.com/1/kubectl-create-clusterrolebinding)=user2 [--group](https://www.mankier.com/1/kubectl-create-clusterrolebinding)=group1

# Create Server Roles [Cloud Shell]

kubectl create -f src/kubernetes/server/server-roles.yaml

role.rbac.authorization.k8s.io/scdf-role created

# Create Server Role Bindings [Cloud Shell]

kubectl create -f src/kubernetes/server/server-rolebinding.yaml

rolebinding.rbac.authorization.k8s.io/scdf-rb created

# Create Server Service Account [Cloud Shell]

kubectl create -f src/kubernetes/server/server-rolebinding.yaml

serviceaccount/scdf-sa created

/\* Optional

# Create RabbitMQ ConfigMap [Cloud Shell]

kubectl create -f src/kubernetes/server/server-config-rabbit.yaml

configmap/scdf-server created

 \*/

# Create Kafka ConfigMap [Cloud Shell]

kubectl create -f src/kubernetes/server/server-config-kafka.yaml

Error from server (AlreadyExists): error when creating "src/kubernetes/server/server-config-kafka.yaml": configmaps "scdf-server" already exists

# Create Server Service Deployment [Cloud Shell]

kubectl create -f src/kubernetes/server/server-svc.yaml

service/scdf-server created

# Create Server Deployment [Cloud Shell]

kubectl create -f src/kubernetes/server/server-deployment.yaml

Error from server (AlreadyExists): error when creating "src/kubernetes/server/server-deployment.yaml": deployments.apps "scdf-server" already exists

# Verify Server Deployment [Cloud Shell]

kubectl get all -l app=scdf-server

NAME                             READY     STATUS    RESTARTS   AGE

pod/scdf-server-d56b88c6-qj6bd   1/1       Running   0          1m

NAME                  TYPE           CLUSTER-IP      EXTERNAL-IP     PORT(S)        AGE

service/scdf-server   LoadBalancer   10.228.12.113   35.244.16.162   80:30571/TCP   1m

NAME                          DESIRED   CURRENT   UP-TO-DATE   AVAILABLE   AGE

deployment.apps/scdf-server   1         1         1            1           1m

NAME                                   DESIRED   CURRENT   READY     AGE

replicaset.apps/scdf-server-d56b88c6   1         1         1         1m

# Locate External IP Address [Cloud Shell]

kubectl get svc scdf-server

NAME          TYPE           CLUSTER-IP      EXTERNAL-IP     PORT(S)        AGE

scdf-server   LoadBalancer   10.228.12.113   35.244.16.162   80:30571/TCP   2m

35.196.177.103---------------------------------------------------------

Deploying Streams with 35.196.35.186

---------------------------------------------------------

# Deploying Streams [Cloud Shell]

wget <http://repo.spring.io/release/org/springframework/cloud/spring-cloud-dataflow-shell/1.7.4.RELEASE/spring-cloud-dataflow-shell-1.7.4.RELEASE.jar>

2019-03-25 03:56:17 (7.16 MB/s) - ‘spring-cloud-dataflow-shell-1.7.4.RELEASE.jar’ saved [23833349/23833349]

---------------------------------------------------------

Starting SCDF Server with Skipper

---------------------------------------------------------

# Initialize the Data Flow server with skipper [Cloud Shell]

java -jar spring-cloud-dataflow-shell-1.7.4.RELEASE.jar --dataflow.mode=skipper

java -jar spring-cloud-dataflow-shell-1.7.4.RELEASE.jar --dataflow.mode=classic

# Login into Data Flow server

dataflow config server --username user --password password --uri <http://34.74.38.0/>

run dashboard

[http://34.74.133.174/dashboard/](http://34.74.133.174/dashboard/%A0)

[http://35.196.34.85 HYPERLINK "http://34.74.77.172/dashboard/" HYPERLINK "http://34.74.77.172/dashboard/" HYPERLINK "http://34.74.77.172/dashboard/" HYPERLINK "http://34.74.77.172/dashboard/" HYPERLINK "http://34.74.77.172/dashboard/" HYPERLINK "http://34.74.77.172/dashboard/" HYPERLINK "http://34.74.77.172/dashboard/" HYPERLINK "http://34.74.77.172/dashboard/" HYPERLINK "http://34.74.77.172/dashboard/" HYPERLINK "http://34.74.77.172/dashboard/" HYPERLINK "http://34.74.77.172/dashboard/" HYPERLINK "http://34.74.77.172/dashboard/" HYPERLINK "http://34.74.77.172/dashboard/" HYPERLINK "http://34.74.77.172/dashboard/" HYPERLINK "http://34.74.77.172/dashboard/" HYPERLINK "http://34.74.77.172/dashboard/" HYPERLINK "http://34.74.77.172/dashboard/" HYPERLINK "http://34.74.77.172/dashboard/" HYPERLINK "http://34.74.77.172/dashboard/" HYPERLINK "http://34.74.77.172/dashboard/" HYPERLINK "http://34.74.77.172/dashboard/" HYPERLINK "http://34.74.77.172/dashboard/" HYPERLINK "http://34.74.77.172/dashboard/" HYPERLINK "http://34.74.77.172/dashboard/" HYPERLINK "http://34.74.77.172/dashboard/" HYPERLINK "http://34.74.77.172/dashboard/" HYPERLINK "http://34.74.77.172/dashboard/" HYPERLINK "http://34.74.77.172/dashboard/" HYPERLINK "http://34.74.77.172/dashboard/" HYPERLINK "http://34.74.77.172/dashboard/" HYPERLINK "http://34.74.77.172/dashboard/"/dashboard/](http://34.74.77.172/dashboard/)

<http://35.196.177.103/dashboard/>

/\* Optional

# Register the Docker Images of RabbitMQ

# Source

app register --type source --name time --uri docker://springcloudstream/time-source-rabbit:1.3.1.RELEASE --metadata-uri maven://org.springframework.cloud.stream.app:time-source-rabbit:jar:metadata:1.3.1.RELEASE

app register --type source --name sourcedocker --uri docker://kamanakanjani/sourcekafka:latest

app register --type source --name sourcedocker --uri docker://kamanakanjani/sourcenew:latest

app register --type standalone --name standaloneapp --uri docker://kamanakanjani/sourcedone:latest

app register --type source --name sourcefinal --uri docker://kamanakanjani/sourcefinal:latest

app register --type processor --name processorfinal --uri docker://kamanakanjani/processorfinalpp:latest

app register --type sink --name sinkfinal --uri docker://kamanakanjani/sinkfinal:latest

stream create --name processorfinal --definition 'sourcefinal | processorfinal | sinkfinal'

stream deploy --name processorfinal

app register --type source --name sources3 --uri docker://kamanakanjani/sources3final:latest

app register --type sink --name sinks3 --uri docker://kamanakanjani/sinks3final:latest

stream create --name s3done --definition 'sources3 | sinks3'

stream deploy --name s3done

OR

dataflow:>app register --name myprocessor --type processor --uri [file:///Users/example/myprocessor-1.2.3.jar](file://\Users\example\myprocessor-1.2.3.jar)

# Sink

app register --type sink --name log --uri docker://springcloudstream/log-sink-rabbit:1.3.1.RELEASE --metadata-uri maven://org.springframework.cloud.stream.app:log-sink-rabbit:jar:metadata:1.3.1.RELEASE

 \*/

# Register the Docker Images of Kafka

# Source

app register --type source --name time --uri docker://springcloudstream/time-source-kafka-10:1.3.1.RELEASE --metadata-uri maven://org.springframework.cloud.stream.app:time-source-kafka-10:jar:metadata:1.3.1.RELEASE

app register --type source --name sources3 --uri docker://kamanakanjani/sources3:latest

app register --type sink --name sinks3 --uri docker://kamanakanjani/sinks3:latest

stream create --name s3done --definition 'sources3 | sinks3'

stream deploy --name s3done

app register --type source --name awssources3 --uri docker://kamanakanjani/awssources3finaldone:latest

app register --type sink --name awssinks3 --uri docker://kamanakanjani/awssink3finaldone:latest

stream create --name awss3done --definition 'awssources3 | awssinks3'

stream deploy --name awss3done

app register --type source --name sourcekafka --uri docker://kamanakanjani/sourcekafka:latest

app register --type sink --name sinkkafka --uri docker://kamanakanjani/sinkkafka:latest

stream create --name kafkalogdone --definition 'sourcekafka | sinkkafka'

stream deploy --name kafkalogdone

Successfully registered application 'source:time'

# Sink

app register --type sink --name log --uri docker://springcloudstream/log-sink-kafka-10:1.3.1.RELEASE --metadata-uri maven://org.springframework.cloud.stream.app:log-sink-kafka-10:jar:metadata:1.3.1.RELEASE

Successfully registered application 'sink:log'

stream create --name loggings --definition 'time | log'

stream deploy --name loggings

# Check Deployment of Streams on Pods [Cloud Shell (Dataflow)]

! kubectl get pods -l role=spring-app

command is:kubectl get pods -l role=spring-app

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| NAME | READY | STATUS | RESTARTS | AGE |
| ticktock-log-v1-7c9cb4f89b-n7n29 | 1/1 | Running | 0 | 1m |
| ticktock-time-v1-859fc54c86-lj2ng | 1/1 | Running | 0 | 1m |

# View Logs for Pod Deployment [Cloud Shell (Dataflow)]

! kubectl logs ticktock-log-v1-7c9cb4f89b-n7n29

# View detailed information about the pods

kubectl describe pods/ticktock-log-v1-7c9cb4f89b-n7n29

---------------------------------------------------------

Deploy Tasks

---------------------------------------------------------

# Register an Application called Timestamp

app register --type task --name timestamp --uri docker:springcloudtask/timestamp-task:1.3.1.RELEASE --metadata-uri maven://org.springframework.cloud.task.app:timestamp-task:jar:metadata:1.3.1.RELEASE

Successfully registered application 'task:timestamp'

# Create a Task called Task1 for timestamp application

task create task1 --definition "timestamp"

Created new task 'task1'

# Launch Task Task1

task launch task1

# View Task list

task list

|  |  |  |
| --- | --- | --- |
| Task Name | Task Definition | Task Status |
| Task1 | Timestamp | UNKNOWN |

# View Task execution

task execution list

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Task Name | ID | Start Time | End Time | Exit Code |
| Task1 | 1 |  |  |  |

app register --name mysource --type source --uri <https://storage.googleapis.com/springclouddataflowbucket/source-0.0.1-SNAPSHOT.jar>

app register --name myprocessor --type processor --uri <https://storage.googleapis.com/springclouddataflowbucket/processor-0.0.1-SNAPSHOT.jar>

app register --name mysink --type sink --uri <https://storage.googleapis.com/springclouddataflowbucket/sink-0.0.1-SNAPSHOT.jar>

stream create --name logkafkadone--definition 'sourcedocker | sink'

stream deploy --name logkafkadone

stream create --name logging --definition 'source | processor | sink'

stream deploy --name logging

stream create --name logdata --definition 'sourcedocker | processordocker | sinkdocker'

stream deploy --name logdata

stream create --name logdata --definition 'time | log'

stream deploy --name logdata

<https://cloud.google.com/kubernetes-engine/docs/tutorials/hello-app>

kubectl get svc

kubectl get all