# 

**Document Foundation AI (DFAI):**

**CloudBuild for MLOPs & Microservices**



## 

**Table of Contents (TOC)**

## 

[**Objective**](#_a2nf7mr5nytb) **4**

[**ML Model Repository Tags**](#_ss47m092bn7w) **4**

[2.1. Git command for tag push to the source repository](#_4anfk0m3hq6f) 5

[Clone the ml-model repo and git checkout to main branch](#_n2souxko9szf) 5

[**DAGs trigger (pipeline\_01)**](#_i8sydeg1kngu) **5**

[3.1. Overview](#_r6tgw6gzwcln) 6

[**MLOps\_pipeline trigger (pipeline\_02)**](#_7g80zd3dpcd0) **7**

[4.1. Overview](#_gms6u8rz97b6) 7

[4.2. Cloudbuild trigger details](#_uud8f5lg2fbl) 8

[**Model Deployment (pipeline\_03 & 04)**](#_g86ejys4co2) **14**

[5.1. Overview](#_9xoovvtyti9r) 14

[5.2. Document AI Model Deployment](#_8vjzdd9e61f1) 15

[5.2.1. Cloudbuild trigger details](#_9hza4wrfsk75) 15

[5.2.2. Step-by-Step Process for Document AI model Deployment](#_o0y7jlwweoza) 17

[5.3. Vertex AI Model Deployment](#_s8gc3zh1wq9v) 18

[5.3.1. Cloudbuild trigger details](#_952f6rlhy4ax) 18

[5.3.2. Pipeline Code Snippets](#_61yx3grxojt2) 23

[**Python Packages Triggers**](#_6pjf92lni0kd) **25**

[6.1. Cloudbuild trigger details](#_bya42xefv9uk) 25

[**Microservice Triggers**](#_1wfsayold8mp) **30**

[7.1. Microservice Repository Tags](#_e8pmf877dnd6) 31

[7.2. Cloudbuild trigger details](#_24k0o7ep9rjs) 31

## 

# Objective

The Objective of this document is to provide a detailed understanding of cloud buid triggers for Vertex AI, Doc AI processors and Microservices.

## 

# ML Model Repository Tags

Following are the details of the repository:

**Project name – Ibank-docai-devops**

| **Repository Name** | **Tags** |
| --- | --- |
| mlops\_composer | **cdags-v.\*** |
| cdc\_pipeline\_evl | **cdc-ev.\*** |
| cdc\_pipeline\_traning | **cdc-tr.\*** |
| cdc\_pipeline\_update to bq | **cdc-utbq.\*** |
| cde\_pipeline\_evl | **cde-ev.\*** |
| cde\_pipeline\_traning | **cde-tr.\*** |
| cde\_pipeline\_update to bq | **cde-utbq.\*** |
| custom\_od\_mp\_exm | **cuod-exm.\*** |
| custom\_od\_mp\_evalu | **cuod-ev.\*** |
| custom\_od\_mp\_train\_ev | **cuod-traev.\*** |
| custom\_od\_mp\_updatetobq | **cuod-utbq.\*** |

## 2.1. Git command for tag push to the source repository

Clone the ml-model repo and git checkout to main branch

**Step-1:** git add .

**Step-2:** git status

**Step-3:** git commit -am “new changes”

\*\*Copy commit id on above step\*\*

**Step-4:** git push && git tag -am tagname-version commit id -m “version”

# DAGs trigger (pipeline\_01)

# 

## 3.1. Overview

When new changes in composer config files and files are pushed to the source repo automatically trigger the cloud build DAGs trigger and update the composer config file to the dags folder.

**3.2. Cloudbuild trigger details**

Cloudbuild trigger name**:** **composer-dags-trigger-uat**

**Note:** Automatically triggered when changes push to source repo with this

**tag: cdags-uat.\***

| **Tag name** | **Last updated tag name &version** | **Next tag version (recommendation)** |
| --- | --- | --- |
| cdags-uat(.\*)$ | cdags-uat1.0.2 | cdags-uat1.0.3 , |

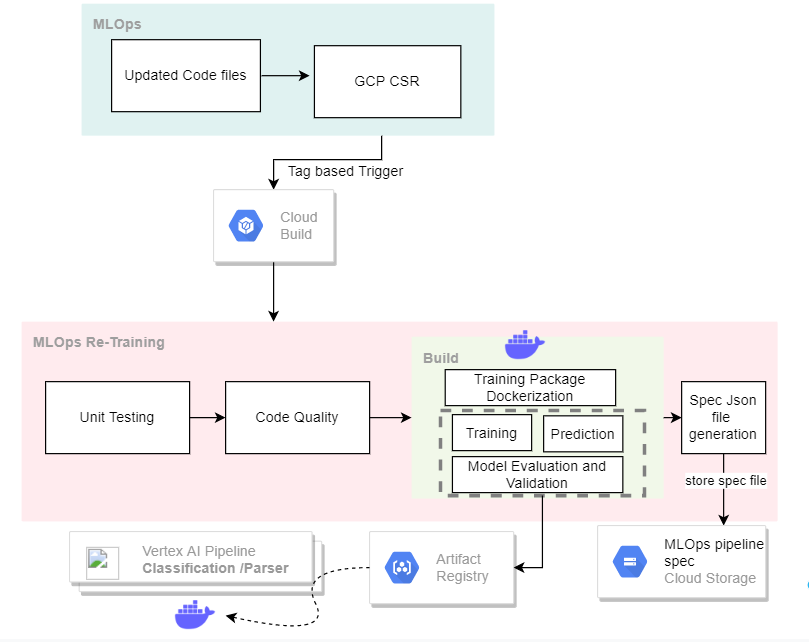
Cloudbuild trigger name**:** **composer-dags-trigger-dev**

**Note:** Automatically triggered when changes push to source repo with this

**tag : cdags-dev.\***

| **Tag name** | **Last updated tag name &version** | **Next tag version (recommendation)** |
| --- | --- | --- |
| cdags-dev(.\*)$ | cdags-dev1.0.1 | cdags-dev1.0.2, |

# MLOps\_pipeline trigger (pipeline\_02)

****

## 4.1. Overview

When new changes in MLOPs pipeline components config files and file pushed to source repo of mlops-model then automatically trigger the cloud build mlops\_pipeline trigger based on tags and & MLOPs pipeline components changes, py coverage and docker image created and push to artifacts registry and generate the spec file of particular pipeline component and stored the spec file into the destination GCS bucket.

## 4.2. Cloudbuild trigger details

Cloudbuild trigger name**:** **mlops-pipeline-cdc-evaluation**

**Note:** Automatically triggered when changes push to source repo with this

**tag: cdc-ev.\***

| **Tag name** | **Last updated tag name &version** | **Next tag version (recommendation)** |
| --- | --- | --- |
| cdc-ev(.\*)$ | cdc-ev1.0.1 | cdc-ev1.0.2 , |

| **Input parameters** | **Parameter description** | **Default parameters** |
| --- | --- | --- |
| **\_IMAGE** | Docker image in artifacts registry | **cdc-pipeline/evaluation** |
| **\_Respository\_url** | Artifacts registry repo URL | **asia-south1-docker.pkg.dev/ibank-docai-devops/ibci-docai-docker-ar** |

Cloudbuild trigger name**:** **mlops-pipeline-cdc-training**

**Note:** Automatically triggered when changes push to source repo with this

**tag: cdc-tr.\***

| **Tag name** | **Last updated tag name &version** | **Next tag version (recommendation)** |
| --- | --- | --- |
| cdc-tr(.\*)$ | cdc-tr1.0 | cdc-tr1.0.1, |

| **Input parameters** | **Parameter description** | **Default parameters** |
| --- | --- | --- |
| **\_IMAGE** | Docker image in artifacts registry | **cdc-pipeline/training** |
| **\_Respository\_url** | Artifacts registry repo URL | **asia-south1-docker.pkg.dev/ibank-docai-devops/ibci-docai-docker-ar** |

Cloudbuild trigger name**:** **mlops-pipeline-cdc-updatetobq**

**Note:** Automatically triggered when changes push to source repo with this

**tag: cdc-utbq.\***

| **Tag name** | **Last updated tag name &version** | **Next tag version (recommendation)** |
| --- | --- | --- |
| cdc-utbq(.\*)$ | cdc-utbq1.0.2 | cdc-utbq1.0.3, |

| **Input parameters** | **Parameter description** | **Default parameters** |
| --- | --- | --- |
| **\_IMAGE** | Docker image in artifacts registry | **cdc-pipeline/updatetobq** |
| **\_Respository\_url** | Artifacts registry repo URL | **asia-south1-docker.pkg.dev/ibank-docai-devops/ibci-docai-docker-ar** |

Cloudbuild trigger name**:** **mlops-pipeline-cde-evaluation**

**Note:** Automatically triggered when changes push to source repo with this

**tag: cde-ev.\***

| **Tag name** | **Last updated tag version** | **Next tag version (recommendation)** |
| --- | --- | --- |
| cde-ev(.\*)$ | cde-ev1.0.1 | cde-ev1.0.2 , |

| **Input parameters** | **Parameter description** | **Default parameters** |
| --- | --- | --- |
| **\_IMAGE** | Docker image in artifacts registry | **cde-pipeline/evaluation** |
| **\_Respository\_url** | Artifacts registry repo URL | **asia-south1-docker.pkg.dev/ibank-docai-devops/ibci-docai-docker-ar** |

Cloudbuild trigger name**:** **mlops-pipeline-cde-training**

**Note:** Automatically triggered when changes push to source repo with this

**tag: cde-tr.\***

| **Tag name** | **Last updated tag name &version** | **Next tag version (recommendation)** |
| --- | --- | --- |
| cde-tr(.\*)$ | cde-tr1.0.1 | cde-tr1.0.2, |

| **Input parameters** | **Parameter description** | **Default parameters** |
| --- | --- | --- |
| **\_IMAGE** | Docker image in artifacts registry | **cde-pipeline/training** |
| **\_Respository\_url** | Artifacts registry repo URL | **asia-south1-docker.pkg.dev/ibank-docai-devops/ibci-docai-docker-ar** |

Cloudbuild trigger name**:** **mlops-pipeline-cde-updatetobq**

**Note:** Automatically triggered when changes push to source repo with this

**tag: cde-utbq.\***

| **Tag name** | **Last updated tag name &version** | **Next tag version (recommendation)** |
| --- | --- | --- |
| cde-utbq(.\*)$ | cde-utbq1.0.2 | cde-utbq1.0.3, |

| **Input parameters** | **Parameter description** | **Default parameters** |
| --- | --- | --- |
| **\_IMAGE** | Docker image in artifacts registry | **cde-pipeline/updatetobq** |
| **\_Respository\_url** | Artifacts registry repo URL | **asia-south1-docker.pkg.dev/ibank-docai-devops/ibci-docai-docker-ar** |

Cloudbuild trigger name**:** **mlops-pipeline-custom-od-evaluation**

**Note:** Automatically triggered when changes push to source repo with this

**tag: cuod-ev.\***

| **Tag name** | **Last updated tag name &version** | **Next tag version (recommendation)** |
| --- | --- | --- |
| cuod-ev(.\*)$ | cuod-ev1.0.1 | cuod-ev1.0.2, |

| **Input parameters** | **Parameter description** | **Default parameters** |
| --- | --- | --- |
| **\_IMAGE** | Docker image in artifacts registry | **custom-od/evaluation** |
| **\_Respository\_url** | Artifacts registry repo URL | **asia-south1-docker.pkg.dev/ibank-docai-devops/ibci-docai-docker-ar** |

Cloudbuild trigger name**:** **mlops-pipeline-custom-od-preprocess**

**Note:** Automatically triggered when changes push to source repo with this

**tag: cuod-pp.\***

| **Tag name** | **Last updated tag name &version** | **Next tag version (recommendation)** |
| --- | --- | --- |
| cuod-pp(.\*)$ | cuod-pp1.0.1 | cuod-pp1.0.2, |

| **Input parameters** | **Parameter description** | **Default parameters** |
| --- | --- | --- |
| **\_IMAGE** | Docker image in artifacts registry | **custom-od/preprocessing** |
| **\_Respository\_url** | Artifacts registry repo URL | **asia-south1-docker.pkg.dev/ibank-docai-devops/ibci-docai-docker-ar** |

Cloudbuild trigger name**:** **mlops-pipeline-custom-od-training-evaluation**

**Note:** Automatically triggered when changes push to source repo with this

**tag : cuod-traev.\***

| **Tag name** | **Last updated tag name &version** | **Next tag version (recommendation)** |
| --- | --- | --- |
| cuod-traev(.\*)$ | cuod-traev1.0.1 | cuod-traev1.0.2, |

| **Input parameters** | **Parameter description** | **Default parameters** |
| --- | --- | --- |
| **\_IMAGE** | Docker image in artifacts registry | **custom-od/train-eval** |
| **\_Respository\_url** | Artifacts registry repo URL | **asia-south1-docker.pkg.dev/ibank-docai-devops/ibci-docai-docker-ar** |

Cloudbuild trigger name: **mlops-pipeline-custom-od-export-model**

**Note:** Automatically triggered when changes push to source repo with this

**tag: cuod-exm.\***

| **Tag name** | **Last updated tag name &version** | **Next tag version (recommendation)** |
| --- | --- | --- |
| cuod-exm(.\*)$ | cuod-exm1.0.1 | cuod-exm1.0.2, |

| **Input parameters** | **Parameter description** | **Default parameters** |
| --- | --- | --- |
| **\_IMAGE** | Docker image in artifacts registry | **custom-od/model-export** |
| **\_Respository\_url** | Artifacts registry repo URL | **asia-south1-docker.pkg.dev/ibank-docai-devops/ibci-docai-docker-ar** |

Cloudbuild trigger name: **mlops-pipeline-custom-od-updatetobq**

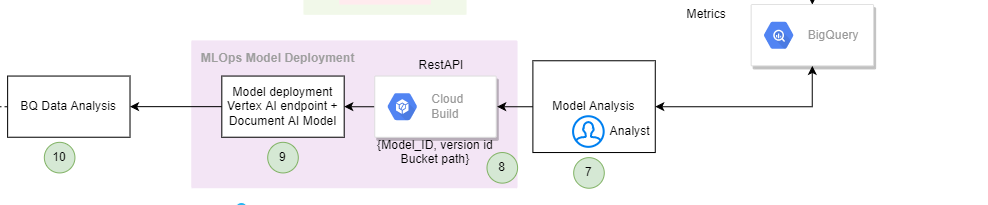
**Note:** Automatically triggered when changes push to source repo with this

**tag : cuod-utbq.\***

| **Tag name** | **Last updated tag name &version** | **Next tag version (recommendation)** |
| --- | --- | --- |
| cuod-utbq(.\*)$ | cuod-utbq1.0.1 | cuod-utbq1.0.2, |

| **Input parameters** | **Parameter description** | **Default parameters** |
| --- | --- | --- |
| **\_IMAGE** | Docker image in artifacts registry | **custom-od/updatetobq** |
| **\_Respository\_url** | Artifacts registry repo URL | **asia-south1-docker.pkg.dev/ibank-docai-devops/ibci-docai-docker-ar** |

# Model Deployment (pipeline\_03 & 04)



## 5.1. Overview

Whenever there is an update in the training dataset, we should perform retraining to create a new model from the latest dataset. After retraining, we will evaluate the model performance by looking at its accuracy metrics and comparing the latest model’s performance with previous model versions. Accuracy metrics will be available in the corresponding Bigquery table. If the latest model’s accuracy metrics are promising, we will deploy the model using a model deployment script.

**Why model deployment is required in the human-in-the-loop part?**

Even if the model accuracy scores look fine, we should validate the latest model with some test documents and make sure that the model has improved its performance, so we require a human to perform this validation of the model.

## 5.2. Document AI Model Deployment

### 5.2.1. Cloudbuild trigger details

Cloudbuild trigger name**:** **model-deployment-mlops-docai-dev**

For ibank-development project model deployment from ibank-docai-devops project cloud build trigger

**Input parameter**

Select the main on branch

| **Input parameters** | **Parameter description** | **Default parameters** | **Mandatory parameters based on requirement** |
| --- | --- | --- | --- |
| Project ID | Must be project id of particular env | Ibank-development |  |
| Project number | Must be project number of particular env | 828122618009 |  |
| Location | Existing processor location | - | **Input needs to change** - Based on processor location |
| Processor ID | Existing processor on docai model | - | **Input needs to change -** based on which processor on docai models |
| Processor version id | Processor ID versions which one need to deploy | - | **Input needs to change -** based on which processor version on docai models |
| User input | Docai model deployment | **docai** |  |

For **ibank-docai-uat** project model deployment from the **ibank-docai-devops** project, cloud build trigger.

Cloudbuild trigger name**:**  **model-deployment-mlops-docai-uat**

**Input parameter**

| **Input parameters** | **Parameter description** | **Default parameters** | **Mandatory input parameters based on requirement** |
| --- | --- | --- | --- |
| Project ID | Must be project id of particular env | Ibank-docai-uat |  |
| Project number | Must be project number of particular env | 607434215947 |  |
| Location | Existing processor location | - | **Input needs to change** - Based on processor location |
| Processor ID | Existing processor on docai model | - | **Input needs to change -** based on which processor on docai models |
| Processor version id | Processor ID versions which one need to deploy | - | **Input needs to change -** based on which processor version on docai models |
| User input | Docai model deployment | **docai** |  |

### 5.2.2. Step-by-Step Process for Document AI model Deployment

1. **Step 1:**
   1. **Deploy new version** 
      1. Use a bash script to deploy the new version
         1. passing input parameters like project and processor and version id
         2. generating the gcloud auth token
         3. fetch the deployed processor version
         4. deploying new the processor version
            1. initial state of deployment “state = UNDEPLOYED”
            2. then into the while loop “state= RUNNING”, given the runtime 300 min For while loop
            3. until reaches the “state=Deployed” if not Deployed back to the previous version deployed state
            4. while loop ends
2. **Step 2:**
   1. **Make it as default new deployed version** 
      1. From step 1 deploying the new version set as the default version.



1. **Step 3:**
   1. **Undeploy previous version**
      1. Undeploy the existing default version
2. **Step 4:**
   1. **Updating the Bigquery table**
   2. **Updating to the Bigquery table based on model\_id and processor version id**
   3. **processor id is Model\_id in Bigquery table**
   4. **Processor version id is Processor version id in the Bigquery table**
   5. **Input processor version id deployed and updated as is\_deployed= true**
   6. **Existing default processor version id undeployed updated as Is\_deployed = false**

## 5.3. Vertex AI Model Deployment

## 

For Vertex AI model deployment we have two scenarios, based on requirements user can trigger build according to scenarios 1 & 2 input parameters

### 5.3.1. Cloudbuild trigger details

For  **ibank-development** project model deployment from **ibank-docai-devops** project cloud build trigger.

Cloudbuild trigger name**:** **model-deployment-mlops-vertexai-dev**

**SCENARIO 1: Deploy a new model to an existing endpoint**

**Important note:** While giving the input parameter the endpoint name must be blank

**Input parameters**

| **Input parameters** | **Parameter description** | **Default parameters** | **Mandatory input parameters based on requirement** |
| --- | --- | --- | --- |
| Project ID | Must be project id of particular env | Ibank-development |  |
| Project number | Must be project number of particular env | 828122618009 |  |
| Location | Existing processor location | - | **Input needs to change** - On which locations model to be installed |
| Model name | Name of the vertex ai model |  |  |
| Endpoint ID | Endpoint id of vertex ai endpoints | - | **Input needs to change -** based on existing endpoint id |
| Endpoint name | - | - | **-** |
| Gs-path | Byte model path | - | **Input needs to change -** based on byte model paths |
| User input | Vertex ai model deployment | **vertexai** |  |

**STEPS:**

1. Upload model into model registry
2. Get the model ID of the uploaded model
3. List down deployed model IDs from the existing endpoint id
4. Deploy New model to existing endpoint
5. Undeploy previous models
6. Update

**SCENARIO 2: Deploy a new model to a new endpoint**

**Important note:** While giving the input parameter the endpoint name must be blank

**Input parameters**

| **Input parameters** | **Parameter description** | **Default parameters** | **Mandatory input parameters based on requirement** |
| --- | --- | --- | --- |
| Project ID | Must be project id of particular env | Ibank-development |  |
| Project number | Must be project number of particular env | 828122618009 |  |
| Location | Existing processor location | - | **Input needs to change** - On which locations model to be installed |
| Model name | Name of the vertex ai model |  | **Input needs to change** -  As per the User choose |
| Endpoint ID | - | - | - |
| Endpoint name | Name of the vertex ai endpoint name | - | **Input needs to change** -  As per the User choose |
| Gs-path | Byte model path | - | **Input needs to change -** based on byte model paths |
| User input | vertexai model deployment | **vertexai** |  |

**STEPS:**

1. Upload model into model registry
2. Get the model ID of the uploaded model
3. Create a new endpoint using Endpoint Name
4. Get the Endpoint ID by using endpoint name
5. Deploy a New model to this new endpoint
6. Based on gs\_path input **is\_deployed = true** updated in Biqquery table

For **ibank-docai-uat project** model deployment from ibank-docai-devops project cloud build trigger

Cloudbuild trigger name**:** **model-deployment-mlops-vertexai-uat**

**SCENARIO 1: Deploy a new model to an existing endpoint**

**Important note:** While giving the input parameter the endpoint name must be blank

**Input parameters**

| **Input parameters** | **Parameter description** | **Default parameters** | **Mandatory input parameters based on requirement** |
| --- | --- | --- | --- |
| Project ID | Must be project id of particular env | Ibank-docai-uat |  |
| Project number | Must be project number of particular env | 607434215947 |  |
| Location | Existing processor location | - | **Input needs to change** - On which locations model to be installed |
| Model name | Name of the vertex ai model |  | **Input needs to change** -  As per the User choose |
| Endpoint ID | Endpoint id of vertex ai endpoints | - | **Input needs to change -** based on existing endpoint id |
| Endpoint name | - | - | **-** |
| Gs-path | Byte model path | - | **Input needs to change -** based on byte model paths |
| User input | vertexai model deployment | **vertexai** |  |

**SCENARIO 2: Deploy a new model to a new endpoint**

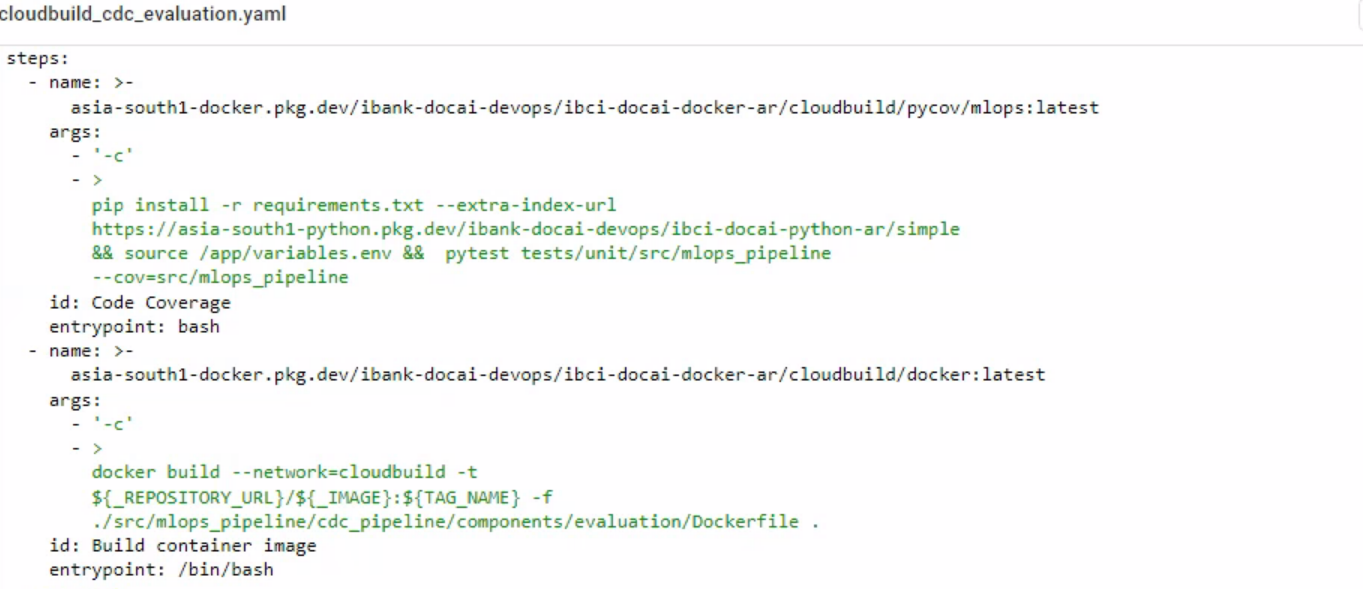
**Important note:** While giving the input parameter the endpoint name must be blank

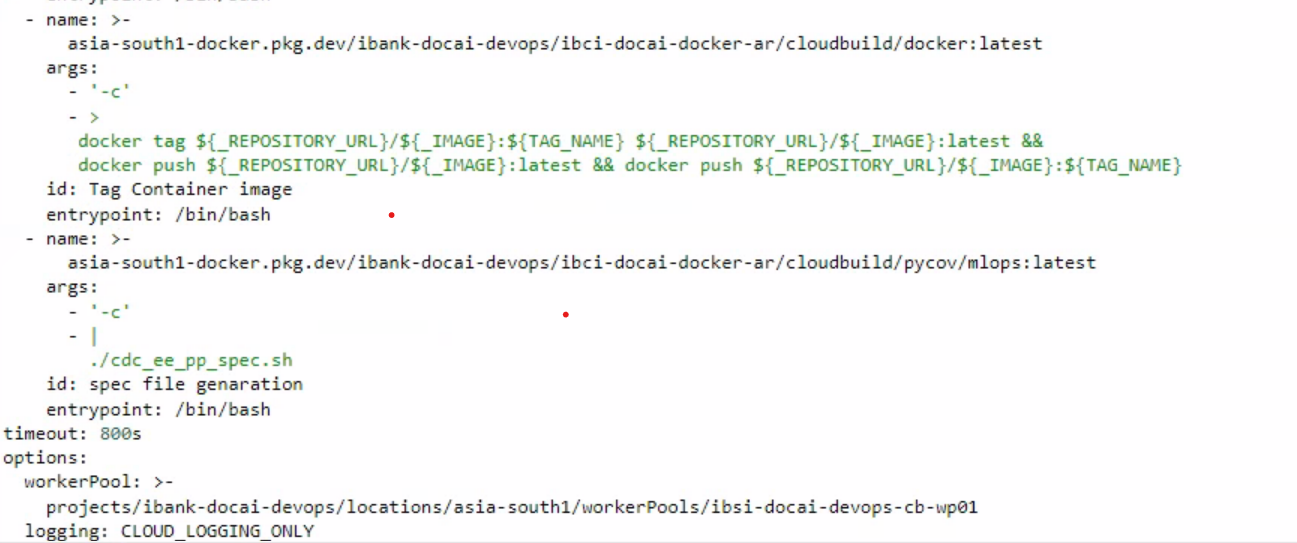
**Input parameters**

| **Input parameters** | **Parameter description** | **Default parameters** | **Mandatory input parameters based on requirement** |
| --- | --- | --- | --- |
| Project ID | Must be project id of particular env | Ibank-docai-uat |  |
| Project number | Must be project number of particular env | 607434215947 |  |
| Location | Existing processor location | - | **Input needs to change** - On which locations model to be installed |
| Model name | Name of the vertex ai model |  | **Input needs to change** -  As per the User choose |
| Endpoint ID | - | - | - |
| Endpoint name | Name of the vertex ai endpoint name | - | **Input needs to change** -  As per the User choose |
| Gs-path | Byte model path | - | **Input needs to change -** based on byte model paths |
| User input | vertexai model deployment | **vertexai** |  |

### 5.3.2. Pipeline Code Snippets

****

****

****

****

****

# Python Packages Triggers

## 6.1. Cloudbuild trigger details

**Cloudbuild Trigger name: docai-docsplit**

**Description:** To upload **docai-docsplit** python package to Artifact Repository.

Push the **ds-v** tag to the ml-models repository to initiate the trigger. For e.g **ds-v1.0.2**

**Input parameter**

| **Input parameters** | **Parameter description** | **Default parameters** |
| --- | --- | --- |
| \_DOCKER\_IMAGE | Docker image required to build the python package | asia-south1-docker.pkg.dev/ibank-docai-devops/ibci-docai-docker-ar/pythonimage:package-slim |
| \_PACKAGE | Python Package Name | docai-docsplit |
| \_REPOSITORY\_URL | Repository in which package would be uploaded | asia-south1-python.pkg.dev/ibank-docai-devops/ibci-docai-python-ar/ |

**Cloudbuild Trigger name: docai-fuzzymatch**

**Description:** To upload **docai-fuzzymatch** python package to Artifact Repository.

Push the **df-v** tag to the **ml-models** repo to initiate the trigger. For **e.g df-v1.0.17**

**Input parameter**

| **Input parameters** | **Parameter description** | **Default parameters** |
| --- | --- | --- |
| \_DOCKER\_IMAGE | Docker image required to build the python package | asia-south1-docker.pkg.dev/ibank-docai-devops/ibci-docai-docker-ar/pythonimage:package-slim |
| \_PACKAGE | Python Package Name | docai-fuzzymatch |
| \_REPOSITORY\_URL | Repository in which package would be uploaded | asia-south1-python.pkg.dev/ibank-docai-devops/ibci-docai-python-ar/ |

**Cloudbuild Trigger name: docai-model-stayconnected**

**Description:** To upload **docai-model-stayconnected** python package to Artifact Repository.

Push the **dms-v tag** to **ml-models** repo to initiate the trigger. For e.g **dms-v1.0.13**

**Input parameter**

| **Input parameters** | **Parameter description** | **Default parameters** |
| --- | --- | --- |
| \_DOCKER\_IMAGE | Docker image required to build the python package | asia-south1-docker.pkg.dev/ibank-docai-devops/ibci-docai-docker-ar/pythonimage:package-slim |
| \_PACKAGE | Python Package Name | docai-model-stayconnected |
| \_REPOSITORY\_URL | Repository in which package would be uploaded | asia-south1-python.pkg.dev/ibank-docai-devops/ibci-docai-python-ar/ |

**Cloudbuild Trigger name: docai-preproc**

**Description:** To upload docai-preproc python package to Artifact Repository.

Push the **dp-v tag** to the **ml-models** repo to initiate the trigger. For e.g **dp-v1.0.12**

**Input parameter**

| **Input parameters** | **Parameter description** | **Default parameters** |
| --- | --- | --- |
| \_DOCKER\_IMAGE | Docker image required to build the python package | asia-south1-docker.pkg.dev/ibank-docai-devops/ibci-docai-docker-ar/pythonimage:package-slim |
| \_PACKAGE | Python Package Name | docai-preproc |
| \_REPOSITORY\_URL | Repository in which package would be uploaded | asia-south1-python.pkg.dev/ibank-docai-devops/ibci-docai-python-ar/ |

**Cloudbuild Trigger name: docai-processor**

**Description:** To upload **docai-processor** python package to Artifact Repository.

Push the **dpro-v tag** to the **ml-models** repo to initiate the trigger. For **e.g dpro-v1.0.12**

**Input parameter**

| **Input parameters** | **Parameter description** | **Default parameters** |
| --- | --- | --- |
| \_DOCKER\_IMAGE | Docker image required to build the python package | asia-south1-docker.pkg.dev/ibank-docai-devops/ibci-docai-docker-ar/pythonimage:package-slim |
| \_PACKAGE | Python Package Name | docai-processor |
| \_REPOSITORY\_URL | Repository in which package would be uploaded | asia-south1-python.pkg.dev/ibank-docai-devops/ibci-docai-python-ar/ |

**Cloudbuild Trigger name: qdocai-logger**

**Description:** To upload docai-processor python package to Artifact Repository.

Push the **ql-v tag** to the **ml-models** repo to initiate the trigger. For **e.g ql-v1.0.12**

**Input parameter**

| **Input parameters** | **Parameter description** | **Default parameters** |
| --- | --- | --- |
| \_DOCKER\_IMAGE | Docker image required to build the python package | asia-south1-docker.pkg.dev/ibank-docai-devops/ibci-docai-docker-ar/pythonimage:package-slim |
| \_PACKAGE | Python Package Name | qdocai-logger |
| \_REPOSITORY\_URL | Repository in which package would be uploaded | asia-south1-python.pkg.dev/ibank-docai-devops/ibci-docai-python-ar/ |

**Cloudbuild Trigger name: qdocai-wrapper**

**Description:** To upload **docai-wrappe**r python package to Artifact Repository.

Push the **qw-v tag** to the **ml-models** repo to initiate the trigger. For **e.g qw-v1.0.12**

**Input parameter**

| **Input parameters** | **Parameter description** | **Default parameters** |
| --- | --- | --- |
| \_DOCKER\_IMAGE | Docker image required to build the python package | asia-south1-docker.pkg.dev/ibank-docai-devops/ibci-docai-docker-ar/pythonimage:package-slim |
| \_PACKAGE | Python Package Name | qdocai-wrapper |
| \_REPOSITORY\_URL | Repository in which package would be uploaded | asia-south1-python.pkg.dev/ibank-docai-devops/ibci-docai-python-ar/ |

**Cloudbuild Trigger name: qdocai-utils**

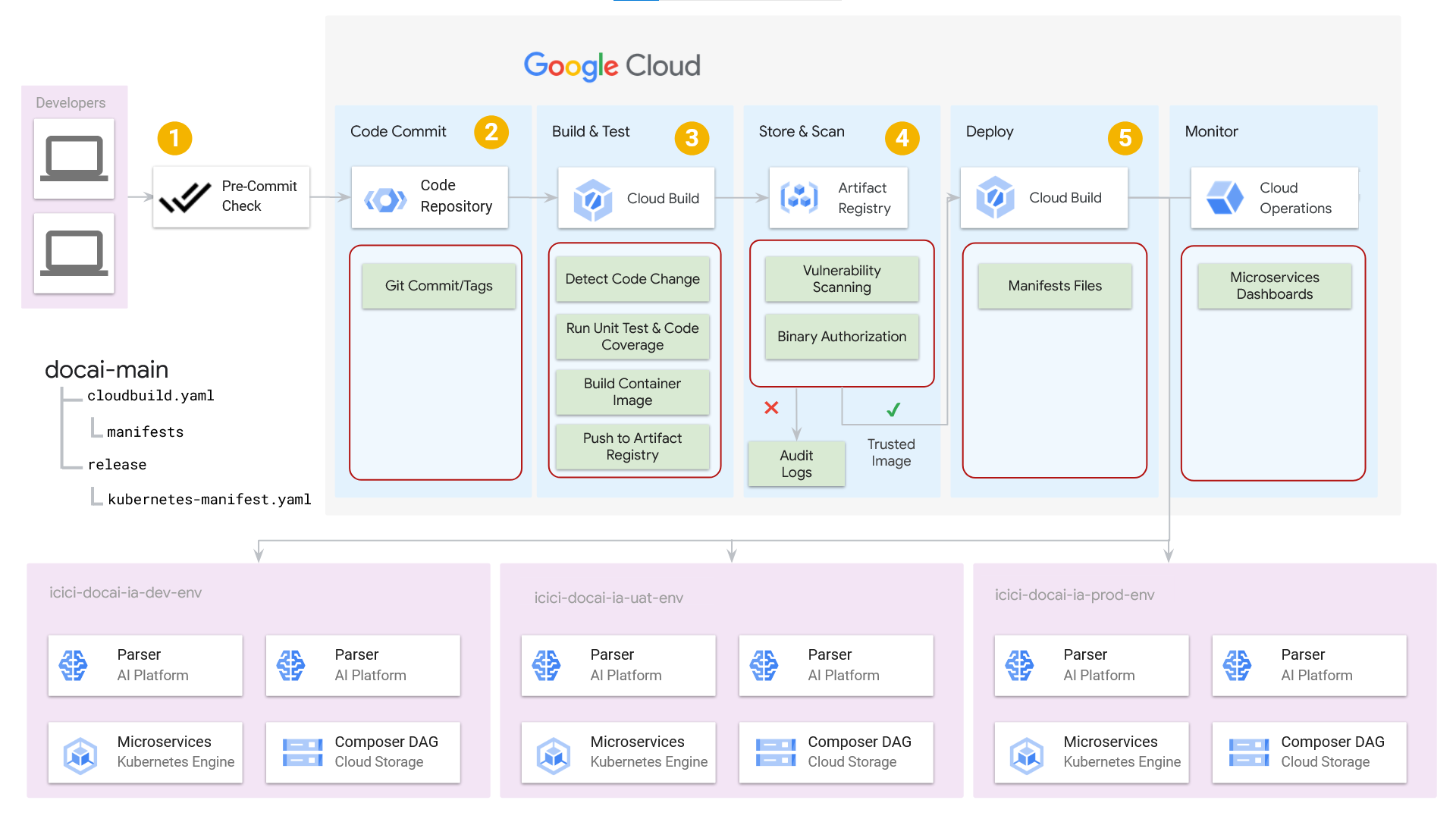
**Description:** To upload docai-utils python package to Artifact Repository.

Push the **qu-v tag** to the **ml-models** repo to initiate the trigger. For **e.g qu-v1.0.12**

**Input parameter**

| **Input parameters** | **Parameter description** | **Default parameters** |
| --- | --- | --- |
| \_DOCKER\_IMAGE | Docker image required to build the python package | asia-south1-docker.pkg.dev/ibank-docai-devops/ibci-docai-docker-ar/pythonimage:package-slim |
| \_PACKAGE | Python Package Name | qdocai-utils |
| \_REPOSITORY\_URL | Repository in which package would be uploaded | asia-south1-python.pkg.dev/ibank-docai-devops/ibci-docai-python-ar/ |

# Microservice Triggers



For every Microservice, we have followed the below steps in CloudBuild Dev Triggers.

* Lint Check
* Code Coverage
* Building Container Image
* Scanning Container Image
* Vulnerability Check
* Push to Artifact Registry
* Binary Auth Attestation
* Deploy on Kubernetes

**Lint Check:** We are using Pylint with a pre-commit tool. We already deployed a container image in which a pylintrc file already exists. Once CloudBuild starts it replaces the current repo pylintrc with the existing one in the container image. Once all the pre-commit passes, this step gets completed.

## 7.1. Microservice Respository Tags

| **Repository Name** | **Tags** |
| --- | --- |
| classification-service | classification-v |
| orchestrator | orchestrator-v |
| image-prep-servce | image-prep-v |
| parser-controller-service | parser-cs-v |
| parser-service | parser-v |
| reader-service | reader-v |
| post-process-service | post-process-v |
| redact-service | redact-v |
| purge-service | purge-v |
| retry-service | retry-v |
| upload-service | upload-v |
| fuzzy-match service | fuzzy-match-v |

## 

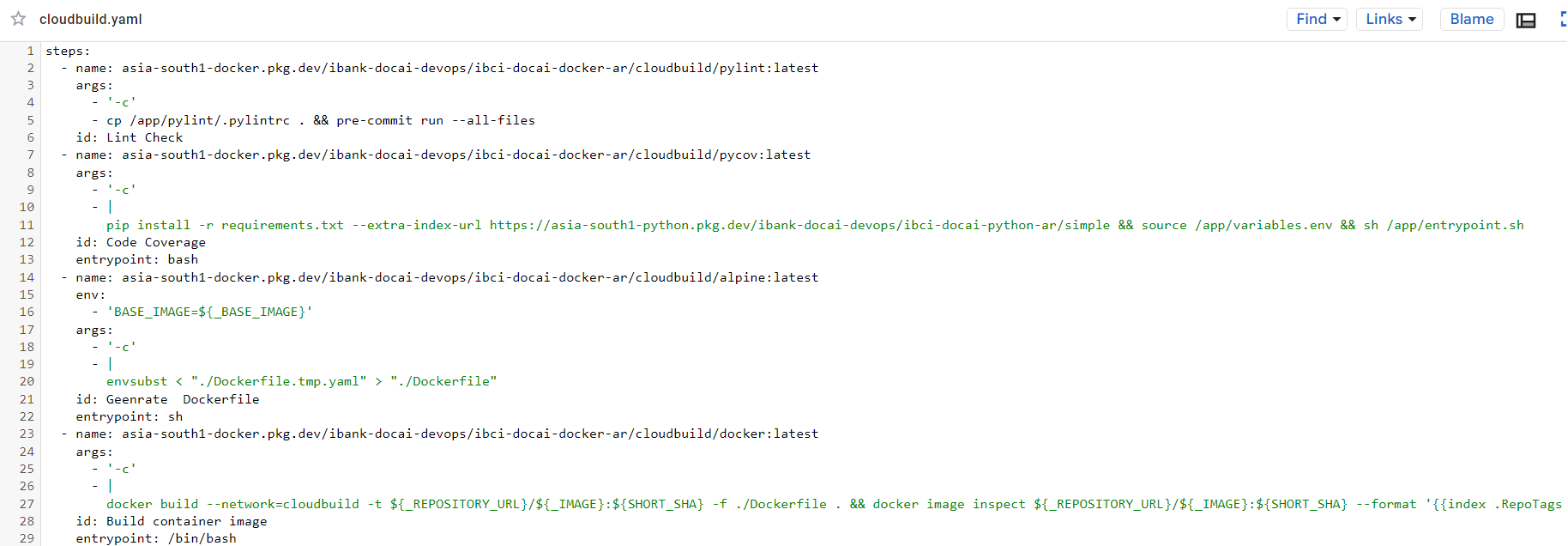
## 7.2. Cloudbuild trigger details

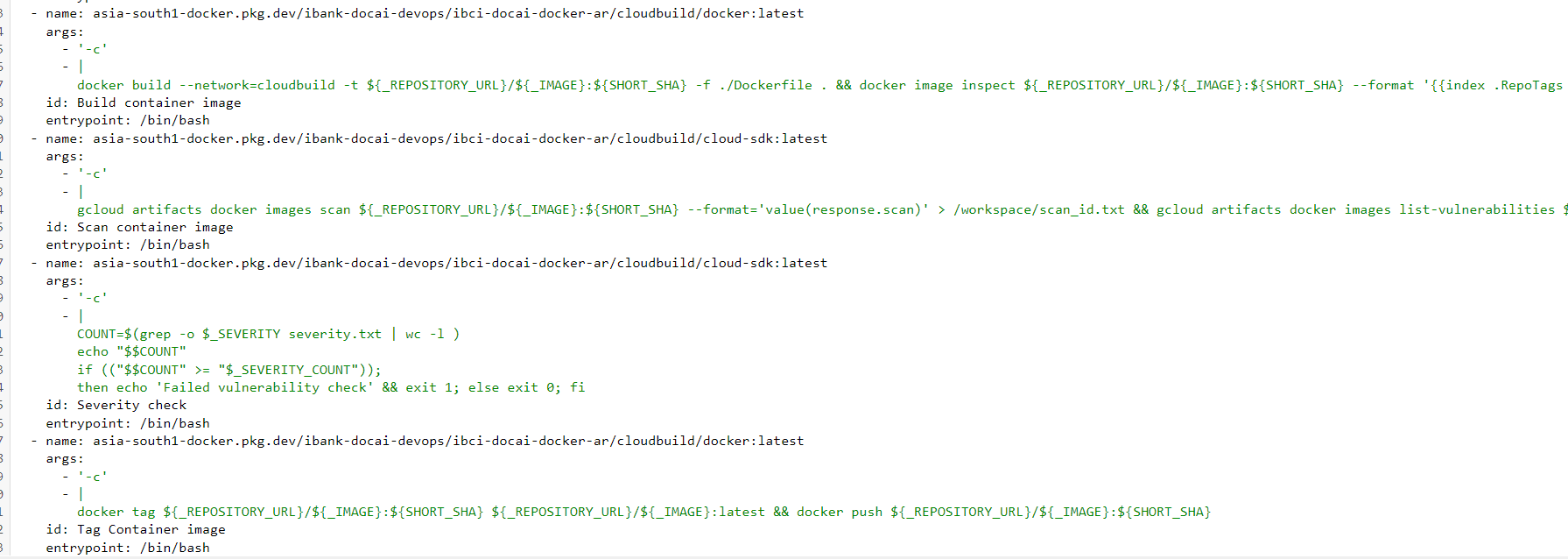
Cloudbuild trigger name**:** **image-prep-trigger-dev**

**Substitution Variables**

| **Input parameters** | **Parameter description** | **Sample Parameters** |
| --- | --- | --- |
| \_App\_name | **Required for Kubernetes Manifest template, Microservice Name** | **image-prep** |
| \_Base\_image | **Base Image for Microservices** | **asia-south1-docker.pkg.dev/ibank-docai-devops/ibci-docai-docker-ar/pythonimage:slim** |
| \_Cluster\_Name | **GKE cluster Name** | **ibsi-docai-dev-gke01** |
| \_Cluster\_Region | **GKE cluster Region** | **asia-south1** |
| \_Image | **Required for Artifact registry** | **image-prep** |
| \_Project | **GKE Cluster Project ID** | **ibank-development** |
| \_Repository\_url | **Artifact registry URL** |  |
| \_Request\_cpu | **Microservice Request CPU on GKE** | **200m** |
| \_Request\_memory | **Microservice Request Memory on GKE** | **1Gi** |
| \_Severity | **Vulnerability needs to checked in Container Image** | **CRITICAL** |
| \_Severity\_count | **No of Vulnerability which can be passed** | **4** |

**Parameter**





**On-demand Scanning API**

Using On-Demand Scanning as part of your Cloud Build pipeline allows you to block builds if the container image has vulnerabilities with a severity matching a predefined level.

We are using Cloud Build to build the container image from the source code, scan it for vulnerabilities, check the severity levels of the vulnerabilities in it, and push the image to Artifact Registry if there are no or allowed number of vulnerabilities of a specific severity level.