

≥ Shell: dagg-fabric

*ProTip: Hold the Command key when opening shell access to launch a new window.*

```
bash-4.4# ls -lrt
total 0
lrwxrwxrwx 1 root root 25 Mar 21 17:08 service-config.yml -> ../data/service-config.yml
lrwxrwxrwx 1 root root 24 Mar 21 17:08 fabric-config.yml -> ../data/fabric-config.yml
lrwxrwxrwx 1 root root 26 Mar 21 17:08 data-aggregator.yml -> ../data/data-aggregator.yml
lrwxrwxrwx 1 root root 16 Mar 21 17:08 config.py -> ../data/config.py
bash-4.4#
```

```
bash-4.4# cat config.py
import importlib, os, sys
from datetime import timedelta

AGGREGATOR_META_DB = os.getenv('AGGREGATOR_META_DB')

REPORTING_DB = os.getenv('REPORTING_DB')

ELASTIC_SEARCH = {
bash-4.4# ls -lrt
total 0
lrwxrwxrwx 1 root root 25 Mar 21 17:08 service-config.yml -> ../data/service-config.yml
lrwxrwxrwx 1 root root 24 Mar 21 17:08 fabric-config.yml -> ../data/fabric-config.yml
lrwxrwxrwx 1 root root 26 Mar 21 17:08 data-aggregator.yml -> ../data/data-aggregator.yml
lrwxrwxrwx 1 root root 16 Mar 21 17:08 config.py -> ../data/config.py
bash-4.4# cat config.py
import importlib, os, sys
from datetime import timedelta

AGGREGATOR_META_DB = os.getenv('AGGREGATOR_META_DB')

REPORTING_DB = os.getenv('REPORTING_DB')

ELASTIC_SEARCH = {
    "host": os.getenv('ELASTIC_SEARCH')
}

ELASTIC_SEARCH_DYNAMIC_TEMPLATE_SETTINGS = True

SHARD_SETTINGS = {
    "number of shards": 2,
    "number of replicas": 1,
    "index.mapping.total_fields.limit": 2000,
    "index.mapping.nested_fields.limit": 500,
    "index.mapping.ignore_malformed": True
}

LOG_PATH = '/var/log/services/dev/' + os.getenv('TENANT_UPPER') + '/'
FILE_NAME = 'ecom-data-aggregator'

#BEGIN Celery configs

FABRIC_SERVICE_URL = "http://127.0.0.1/v1/_fabric/tenants/DEFAULT/services/{0}/service-config"

BROKER_URL = os.getenv('BROKER_URL')

CELERY_RESULT_BACKEND = os.getenv('CELERY_RESULT_BACKEND')

CELERY_EVENT_QUEUE_TTL = 3600

CELERY_TASK_RESULT_EXPIRES = timedelta(days=60)

BROKER_POOL_LIMIT = 0

BROKER_CONNECTION_TIMEOUT = 2

BROKER_CONNECTION_MAX_RETRIES = 5

CELERYBEAT_SCHEDULE = {
    'aggregator': {
        'task': 'app.tasks.aggregator_tasks.queue_aggregators',
        'schedule': timedelta(minutes=5),
        'args': ()
    },
    'indexer failure': {
        'task': 'app.tasks.indexer_tasks.queue_indexers',
        'schedule': timedelta(minutes=60),
        'args': (True,)
    },
    'cleaner': {
        'task': 'app.tasks.cleaner_tasks.perform_clean_up',
bash-4.4# cat data-aggregator.yml
# This is a YAML-formatted file.
# Declare variables to be passed into helm templates.

Service:
  name: ind-prod-data-aggregator
  environment: production
  namespace: ind-prod
  type: backend
  replicas: 1
  service_configmap: true
  image:
    repository: registry.seasam.org/data-aggregator-ind
    tag: v1.0.0
    pullPolicy: Always
  logPath: /var/log/services/
  certsPath: /certs
  memory: 2Gi
  cpu: 1
  vlan: data
  tenant: ind
bash-4.4#
```

```

bash-4.4# cat service-config.yml
apiVersion: vl
data:
  AGGREGATOR_META_DB: postgres://ecomdba_ind_rw:lo@prod-pg-07.swa.samsung.io:5432/ind_dagg
  BROKER_URL: amqp://rmquser:fsf4l3DSa@prod-rmq-01/ind_celery
  CELERY_RESULT_BACKEND: db+postgres://ecomdba_ind_rw:lo@prod-pg-07.swa.samsung.io:5432/ind_celery_results
  ELASTIC_SEARCH: prod-delm-lb.swa.samsung.io
  REPORTING_DB: postgres://ecomdba_ind_rw:lo@prod-pg-07.swa.samsung.io:5432/ind_reporting
  TENANT_LOWER: ind
  TENANT_UPPER: IND
kind: ConfigMap
metadata:
  name: data-aggregator-kv
bash-4.4#

```

```

bash-4.4# cat fabric-config.yml
catalog:
  app_config:
    cockroachdb:
      connection:
        database: ind_catalog
        host: cockroachdb.swa.samsung.io
        port: "26257"
        ssl:
          certPath: /certs/client.ecomdba_ind.crt
          keyPath: /certs/client.ecomdba_ind.key
          rejectUnauthorized: false
        user: ecomdba_ind
  epp:
    app_config:
      cockroachdb:
        connection:
          database: ind_catalog
          host: cockroachdb.swa.samsung.io
          port: "26257"
          ssl:
            certPath: /certs/client.ecomdba_ind.crt
            keyPath: /certs/client.ecomdba_ind.key
            rejectUnauthorized: false
          user: ecomdba_ind
    order-service:
      app_config:
        postgres:
          connection_string_orders: postgres://ecomdba_ind_rw:lo@prod-pg-01.swa.samsung.io:5432/ind_ecomorders
    partner-api:
      app_config:
        cockroachdb:
          connection:
            database: ind_partner
            host: cockroachdb.swa.samsung.io
            port: "26257"
            ssl:
              certPath: /certs/client.ecomdba_ind.crt
              keyPath: /certs/client.ecomdba_ind.key
              rejectUnauthorized: false
            user: ecomdba_ind
    payment-service:
      app_config:
        postgres:
          connectionstring: postgres://ecomdba_ind_rw:lo@prod-pgs-01.swa.samsung.io:5432/ind_payment
    pricing-engine:
      app_config:
        postgres:
          connectionstring: postgres://ecomdba_ind_rw:lo@prod-pg-03.swa.samsung.io:5432/ind_pricingengine
    redemption-service:
      app_config:
        postgres:
          connection_string_redemption: postgres://ecomdba_ind_rw:lo@prod-pg-11.swa.samsung.io:5432/ind_redemption
    referral-service:
      app_config:
        cockroachdb:
          connection:
            database: ind_referral
            host: cockroachdb.swa.samsung.io
            port: "26257"
            ssl:
              certPath: /certs/client.ecomdba_ind.crt
              keyPath: /certs/client.ecomdba_ind.key
              rejectUnauthorized: false
            user: ecomdba_ind
  service_config:
    lower_tenant_id: ind
    service:
      site_codes: {}
    tenant:
      allowed_locales:
        - en-in
        - en-us
      default_currency: INR
      default_locale: en-in
      locale: en
      service_nginx_base_url: http://ind-prod-nginx
      site_base_url: https://www.samsung.com/in
      tenant_id: IND
  shopping-carts:
    app_config:
      cockroachdb:
        connection:
          database: ind_shoppingcarts
          host: cockroachdb.swa.samsung.io
          port: "26257"
          ssl:
            certPath: /certs/client.ecomdba_ind.crt
            keyPath: /certs/client.ecomdba_ind.key
            rejectUnauthorized: false
          user: ecomdba_ind
  sso:
    app_config:
      cockroachdb:
        connection:
          database: ind_sso
          host: cockroachdb.swa.samsung.io
          port: "26257"
          ssl:
            certPath: /certs/client.ecomdba_ind.crt
            keyPath: /certs/client.ecomdba_ind.key
            rejectUnauthorized: false
          user: ecomdba_ind
  tnc:
    app_config:
      cockroachdb:
        connection:
          database: ind_tnc
          host: cockroachdb.swa.samsung.io
          port: "26257"
          ssl:
            certPath: /certs/client.ecomdba_ind.crt
            keyPath: /certs/client.ecomdba_ind.key
            rejectUnauthorized: false
          user: ecomdba_ind

```

```

tnc:
  app_config:
    cockroachdb:
      connection:
        database: ind_tnc
        host: cockroachdb.swa.samsung.io
        port: "26257"
        ssl:
          certPath: /certs/client.ecomdba_ind.crt
          keyPath: /certs/client.ecomdba_ind.key
          rejectUnauthorized: false
        user: ecomdba_ind
bash-4.4#

```

```
bash-4.4# head access.log
* Running on http://127.0.0.1:80/ (Press CTRL+C to quit)
127.0.0.1 - - [21/Mar/2023 17:08:54] "GET /v1/fabric/tenants/DEFAULT/services/all/service-config HTTP/1.1" 200 -
127.0.0.1 - - [21/Mar/2023 17:08:55] "GET /v1/fabric/tenants/DEFAULT/services/messenger/service-config HTTP/1.1" 200 -
127.0.0.1 - - [21/Mar/2023 17:08:55] "GET /v1_fabric/tenants/DEFAULT/services/carrier-service/service-config HTTP/1.1" 200 -
127.0.0.1 - - [21/Mar/2023 17:08:55] "GET /v1_fabric/tenants/DEFAULT/services/subscription-service/service-config HTTP/1.1" 200 -
127.0.0.1 - - [21/Mar/2023 17:08:55] "GET /v1_fabric/tenants/DEFAULT/services/rewards/service-config HTTP/1.1" 200 -
127.0.0.1 - - [21/Mar/2023 17:10:28] "GET /v1_fabric/tenants/DEFAULT/services/identity-store/service-config HTTP/1.1" 200 -
127.0.0.1 - - [21/Mar/2023 17:10:28] "GET /v1_fabric/tenants/DEFAULT/services/sso/service-config HTTP/1.1" 200 -
127.0.0.1 - - [21/Mar/2023 17:10:28] "GET /v1_fabric/tenants/DEFAULT/services/fulfillment-service/service-config HTTP/1.1" 200 -
127.0.0.1 - - [21/Mar/2023 17:10:28] "GET /v1_fabric/tenants/DEFAULT/services/exchange-service/service-config HTTP/1.1" 200 -
bash-4.4#
```

```
bash-4.4# cat app.py
#!/usr/bin/python
from flask import Flask, abort, jsonify, request, make_response, json, Response
import functools
import sys
import json
import logging, time
import os, base64
import yaml
from logging.handlers import TimedRotatingFileHandler

app = Flask(__name__)

def get_configfile(service):
    with open("./configs-v2/fabric-config.yml") as stream:
        yaml_data = yaml.safe_load(stream)
    return yaml_data.get(service)

def get_whole_configfile():
    with open("./configs-v2/fabric-config.yml") as stream:
        yaml_data = yaml.safe_load(stream)
    return yaml_data

@app.route('/v1/_fabric/tenants/<tenant>/services/<service>/service-config', methods=['GET'])
def get_service_config(tenant, service):

    if service == 'all':
        json_string = json.dumps(get_whole_configfile(),ensure_ascii = False)
    else :
        json_string = json.dumps(get_configfile(service),ensure_ascii = False)
    #creating a Response object to set the content type and the encoding
    response = Response(json_string,content_type="application/json; charset=utf-8" )
    return response

#return jsonify(get_configfile(service))

if __name__ == '__main__':
    LOGPATH='./logs/access.log'
    logger = logging.getLogger('werkzeug')
    logger.setLevel(logging.INFO)
    handler = TimedRotatingFileHandler(LOGPATH, when="d", interval=1, backupCount=30)
    logger.addHandler(handler)

    app.run(host='127.0.0.1',port=int('80'),debug=False)bash-4.4#
```

*ProTip: Hold the Command key when opening shell access to launch a new window*