CDS Regression Test Suite

Purpose

Test common CDS use-cases in a realistic environment:
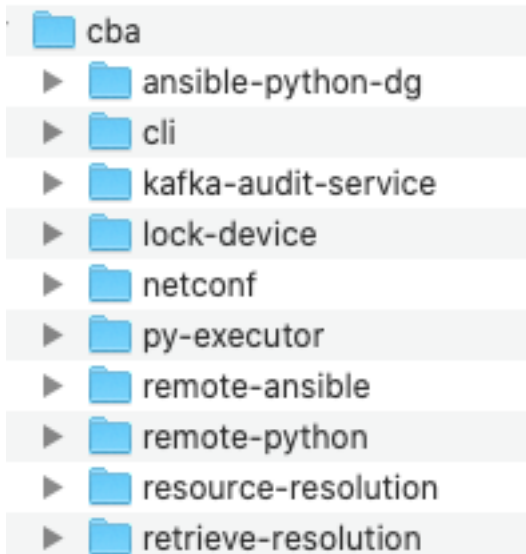
    - Resource resolution

    - Script execution

    - Netconf

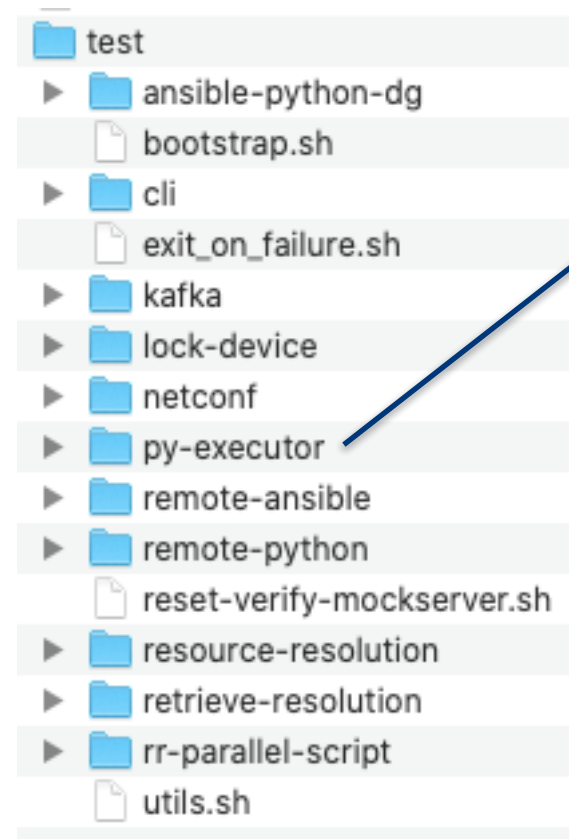   **- CLI**

Tool for development:

    - Spin up an independent instance running a specific image

    - Reproduce bugs
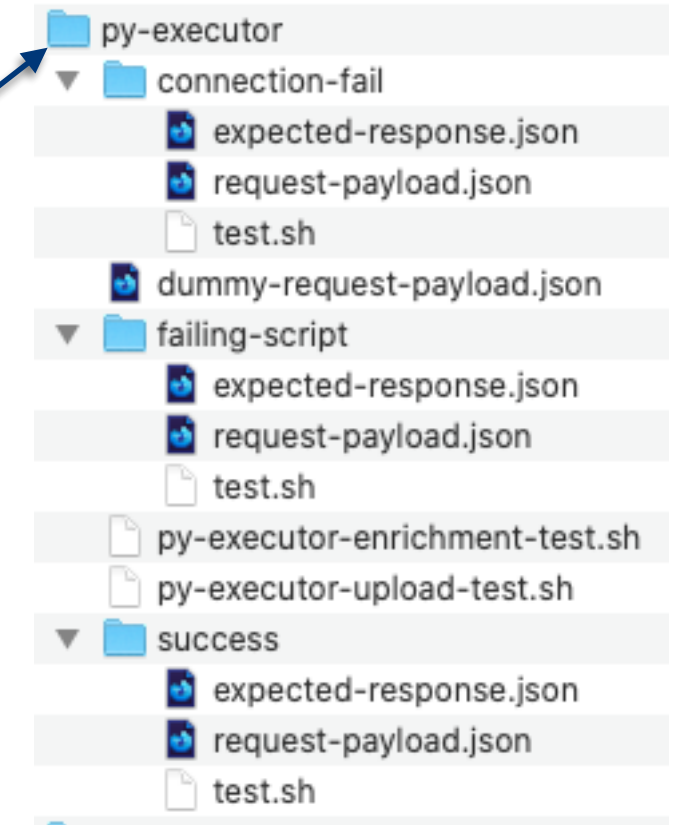
Bell

Repository Structure

CBA Directory
1 per use-case

Tests Directory

Individual Test Directory
Multiple test cases

CBA Directory

- 📁 cba
  - ▶ 📁 ansible-python-dg
  - ▶ 📁 cli
  - ▶ 📁 kafka-audit-service
  - ▶ 📁 lock-device
  - ▶ 📁 netconf
  - ▶ 📁 py-executor
  - ▶ 📁 remote-ansible
  - ▶ 📁 remote-python
  - ▶ 📁 resource-resolution
  - ▶ 📁 retrieve-resolution

Tests Directory

- 📁 test
  - ▶ 📁 ansible-python-dg
  - 📄 bootstrap.sh
  - ▶ 📁 cli
  - 📄 exit_on_failure.sh
  - ▶ 📁 kafka
  - ▶ 📁 lock-device
  - ▶ 📁 netconf
  - ▶ 📁 py-executor
  - ▶ 📁 remote-ansible
  - ▶ 📁 remote-python
  - 📄 reset-verify-mockserver.sh
  - ▶ 📁 resource-resolution
  - ▶ 📁 retrieve-resolution
  - ▶ 📁 rr-parallel-script
  - 📄 utils.sh

Individual Test Directory

- 📁 py-executor
  - ▼ 📁 connection-fail
    - 📄 expected-response.json
    - 📄 request-payload.json
    - 📄 test.sh
  - 📄 dummy-request-payload.json
  - ▼ 📁 failing-script
    - 📄 expected-response.json
    - 📄 request-payload.json
    - 📄 test.sh
  - 📄 py-executor-enrichment-test.sh
  - 📄 py-executor-upload-test.sh
  - ▼ 📁 success
    - 📄 expected-response.json
    - 📄 request-payload.json
    - 📄 test.sh

3

Bell

Test Structure

```bash
#!/bin/bash

CBA_NAME="remote-python"
TEST_NAME="success"
TEST_NUMBER=$RANDOM

mkdir -p "$RESPONSES_DIRECTORY/$CBA_NAME/$TEST_NAME/$TEST_NUMBER/"
RESPONSE_HEADERS_FILE="$RESPONSES_DIRECTORY/$CBA_NAME/$TEST_NAME/$TEST_NUMBER/response-headers"
RESPONSE_PAYLOAD_FILE="$RESPONSES_DIRECTORY/$CBA_NAME/$TEST_NAME/$TEST_NUMBER/response-payload"

REQUEST_PAYLOAD_FILE="$TEST_DIRECTORY/$CBA_NAME/$TEST_NAME/request-payload.json"
EXPECTED_PAYLOAD_FILE="$TEST_DIRECTORY/$CBA_NAME/$TEST_NAME/expected-response.json"

. ./$TEST_DIRECTORY/utils.sh

echo "Running test: $CBA_NAME:$TEST_NUMBER:$TEST_NAME"

echo 'Calling CDS process'
process_cba $REQUEST_PAYLOAD_FILE $RESPONSE_PAYLOAD_FILE $RESPONSE_HEADERS_FILE

echo 'Assert statuscode'
assert_status_code 200 $RESPONSE_HEADERS_FILE

echo 'Assert payload'
assert_cds_response_equals $EXPECTED_PAYLOAD_FILE $RESPONSE_PAYLOAD_FILE
```

Simple bash scripts
1: Send pre-defined request payload
2: Assert status code
3: Assert response payload matches expected payload

Bell

Tests are executed by Gitlab jobs

```
remote-python:
  <<: *globals
  stage: run-tests
  script:
    - export
    - ". ./$TEST_DIRECTORY/remote-python/success/test.sh"
    - ". ./$TEST_DIRECTORY/remote-python/failing-script/test.sh"
    - ". ./$TEST_DIRECTORY/remote-python/connection-fail/test.sh"
    - ". ./$TEST_DIRECTORY/exit_on_failure.sh"
```
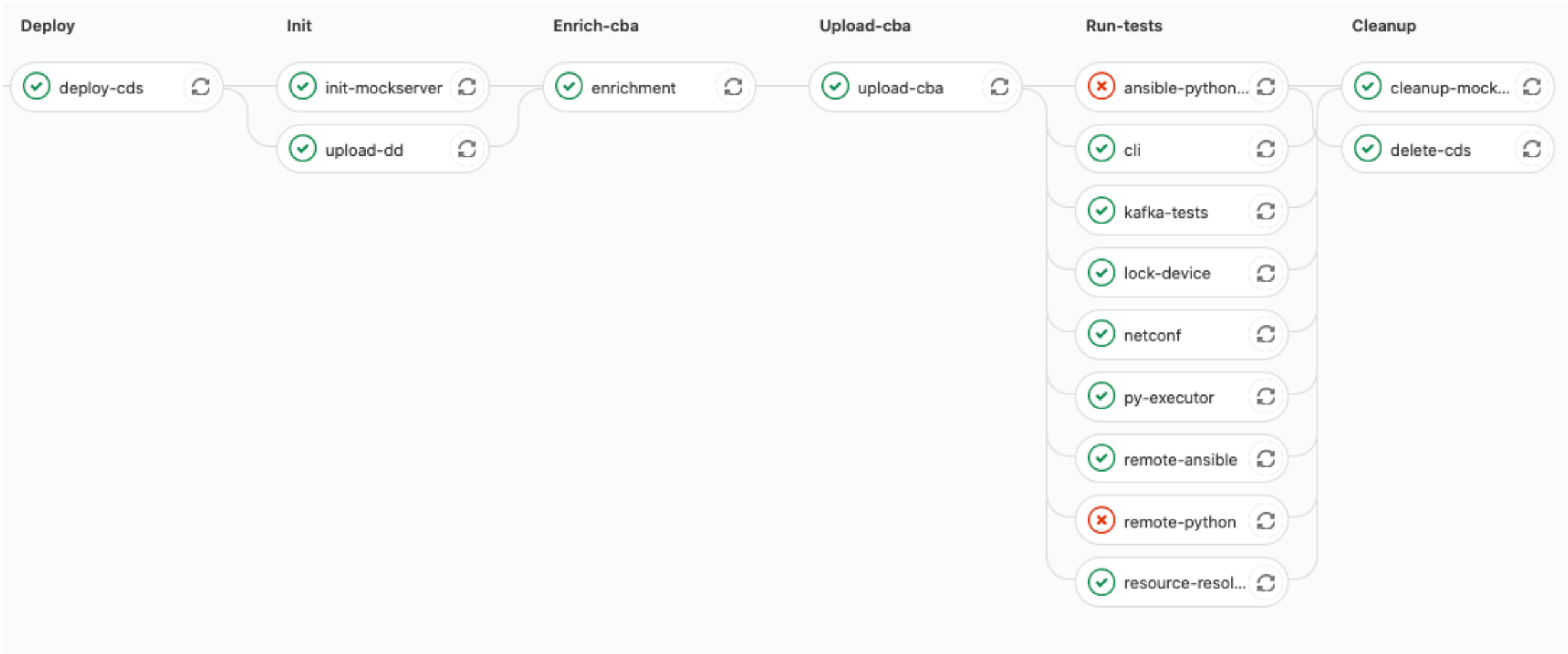
.gitlab-ci.yml

Bell

Tests are executed by Gitlab jobs

```
186  $ . ./$TEST_DIRECTORY/remote-python/success/test.sh
187  Running test: remote-python:9508:success
188  Calling CDS process
189  Assert statuscode
190  Assert payload
191  $ . ./$TEST_DIRECTORY/remote-python/failing-script/test.sh
192  Running test: remote-python:6372:failing-script
193  Calling CDS process
194  Assert statuscode
195  Assert payload
196  $ . ./$TEST_DIRECTORY/remote-python/connection-fail/test.sh
197  Running test: remote-python:977:connection-fail
198  Calling CDS process
199  Assert statuscode
200  Assert payload
201  $ . ./$TEST_DIRECTORY/exit_on_failure.sh
203  Job succeeded
```
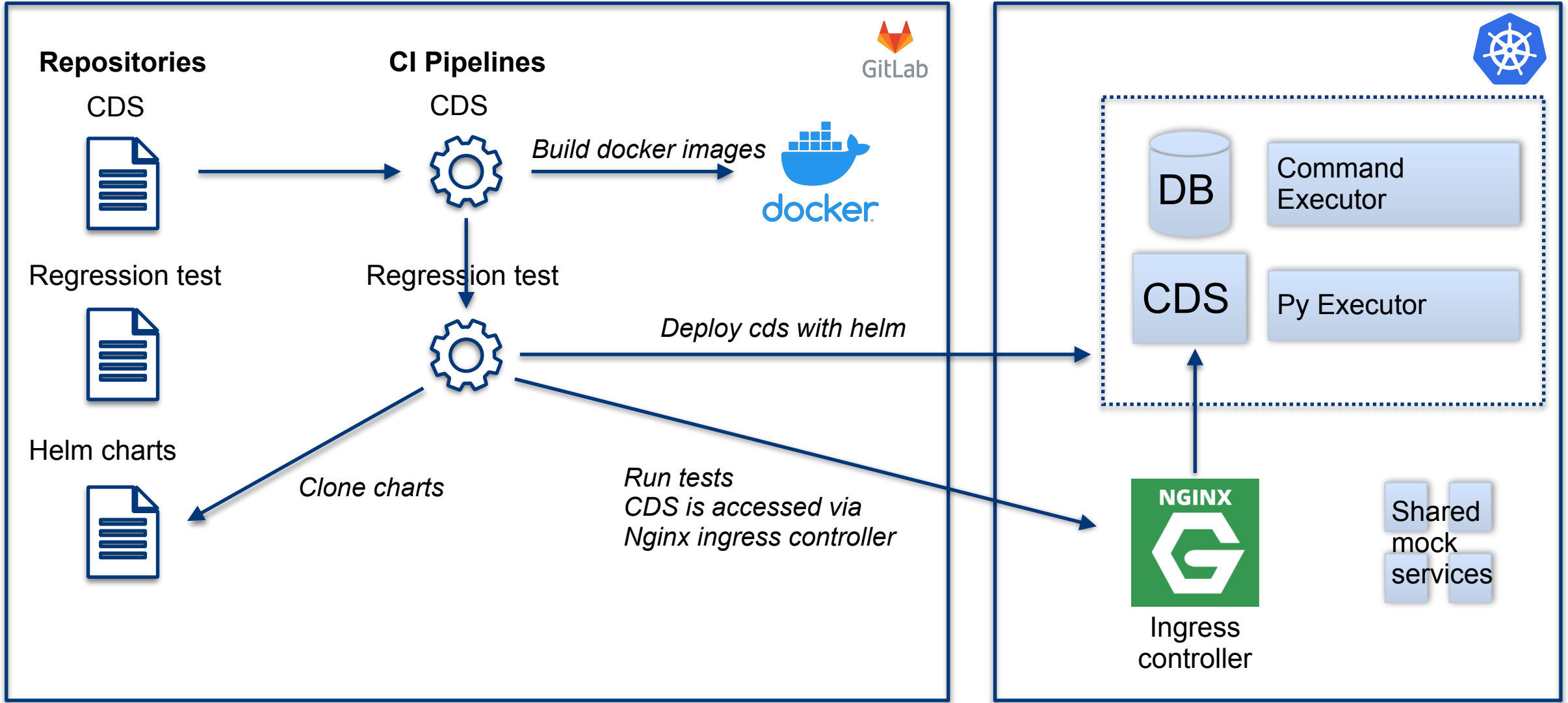
Bell

Tests are executed by Gitlab jobs

```
232  --------------------------------------------------------------
233  Assert payload
234  --------------------------------------------------------------
235  TEST: success
236  assert_json_equals
237  diff:
238  --- /dev/fd/63
239  +++ /dev/fd/62
240  @@ -14,20 +14,15 @@
241     "payload": {
242       "remote-python-response": {
243         "execute-command-logs": [
244  -         "remote executor regression"
245  +         "/bin/sh: 1: Scripts/python/EchoRemotePython.py: Permission denied"
246         ],
247  -       "execute-command-payload": {
248  -         "abc": [
249  -           "xyz",
250  -           "qqq"
251  -         ]
252  -       }
253  +       "execute-command-payload": {}
254       }
255     },
256     "status": {
257  -     "code": 200,
258  -     "errorMessage": null,
259  -     "eventType": "EVENT_COMPONENT_EXECUTED",
260  -     "message": "success"
261  +     "code": 500,
262  +     "errorMessage": "FAILURE : execute-command : [\"/bin/sh: 1: Scripts/python/EchoRemotePython.py: Permission denied\"]",
263  +     "eventType": "EVENT_COMPONENT_FAILURE",
264  +     "message": "failure"
265     }
266  }
```

Bell

Gitlab pipeline

Gitlab CI + Kubernetes

Repositories

CDS

CI Pipelines

CDS

Build docker images → docker

Regression test

Regression test

Deploy cds with helm

Helm charts

Clone charts

*Run tests
CDS is accessed via
Nginx ingress controller*

DB   Command Executor

CDS   Py Executor

NGINX
Ingress controller

Shared mock services

Bell

Ingress enables parallel execution

```
jc@ ~ () $ kubectl -ncds-regression get pods
NAME                                          READY   STATUS             RESTARTS   AGE
cds-844821-cds-844821-f9cb7fd67-29dl4         0/1     Running            0          108s
cds-844821-cds-844821-f9cb7fd67-2f7l4         0/1     PodInitializing    0          108s
cds-844821-cds-844821-f9cb7fd67-ltrfl         0/1     Running            0          108s
cds-844821-cds-db-0                           1/1     Running            0          108s
cds-844822-cds-844822-7796c8474f-h7b66        0/1     Running            0          62s
cds-844822-cds-844822-7796c8474f-vpx5q        0/1     PodInitializing    0          62s
cds-844822-cds-844822-7796c8474f-w9c4c        0/1     PodInitializing    0          62s
cds-844822-cds-db-0                           1/1     Running            0          61s
cds-844823-cds-844823-58d4dc4cf-vfc66         0/1     Init:0/2           0          3s
cds-844823-cds-844823-58d4dc4cf-vt5bv         0/1     Init:0/2           0          3s
cds-844823-cds-844823-58d4dc4cf-w4pzg         0/1     Init:0/2           0          3s
cds-844823-cds-db-0                           0/1     Init:0/1           0          3s
cds-cds-pyexec-844821-547d7849f-njxqw         0/1     ContainerCreating  0          24s
cds-cds-pyexec-844822-77458fbc76-cp74w        0/1     ContainerCreating  0          3s
cds-ce-844821-cds-ce-75c8d8dd6-mtn8h          0/1     Running            0          30s
cds-ce-844822-cds-ce-6f45c7c9cf-cv9nw         0/1     ContainerCreating  0          9s
mockserver-mockserver-587f447cf6-skt4h        1/1     Running            0          93d
odltesttool-odltesttool-6459c5b446-tfnrf      1/1     Running            0          93d
sshmock-sshmock-6964c9d6d5-lmc5q              1/1     Running            0          93d
tiller-deploy-77bf97b4fd-trfx5                1/1     Running            0          6d11h
```

```
jc@ ~ () $ kubectl -ncds-regression get ingress
NAME                   HOSTS   ADDRESS          PORTS   AGE
cds-844821-ingress     *       10.195.177.152   80      2m19s
cds-844822-ingress     *       10.195.177.152   80      93s
cds-844823-ingress     *       10.195.177.152   80      34s
mockserver-ingress     *       10.195.177.152   80      92d
```

```yaml
apiVersion: extensions/v1beta1
kind: Ingress
metadata:
  annotations:
    kubernetes.io/ingress.class: nginx
    nginx.ingress.kubernetes.io/proxy-read-timeout: "300"
    nginx.ingress.kubernetes.io/proxy-request-buffering: "off"
    nginx.ingress.kubernetes.io/proxy-send-timeout: "300"
    nginx.ingress.kubernetes.io/rewrite-target: $1
    nginx.ingress.kubernetes.io/ssl-redirect: "false"
  creationTimestamp: "2021-01-13T01:50:02Z"
  generation: 1
  name: cds-844821-ingress
  namespace: cds-regression
  resourceVersion: "138885546"
  selfLink: /apis/extensions/v1beta1/namespaces/cds-regression/ingresses/cds-844821-ingress
  uid: e316a6bf-4b9f-45b7-8eb3-09a07acbd685
spec:
  rules:
  - http:
      paths:
      - backend:
          serviceName: cds-844821-http
          servicePort: 8080
        path: /cds/cds-844821(/.*$)
```

Bell

Questions