Manage Logpush with Python

You can manage your Cloudflare Logpush service using Python. In the script below you can find example requests to create a job, retrieve job details, update job settings, and delete a Logpush job.

:::note

The examples below are for zone-scoped datasets. Account-scoped datasets should use `<ACCOUNT_ID>` instead of `<ZONE_ID>`.

:::

```
import json
import requests

url = "https://api.cloudflare.com/client/v4/"

x_auth_email = "<EMAIL>"
x_auth_key = "<API_KEY>"

zone_id = "<ZONE_ID>"
destination_conf = "s3://<BUCKET_NAME>/logs?region=us-west-1"

logpush_url = url + "/zones/%s/logpush" % zone_id

headers = {
  'X-Auth-Email': <EMAIL>,
  'X-Auth-Key': <API_KEY>,
  'Content-Type': 'application/json'
}

# Create job
r = requests.post(logpush_url + "/jobs", headers=headers,
data=json.dumps({"destination_conf":destination_conf}))
print(r.status_code, r.text)
```

```
assert r.status_code == 201
assert r.json()["result"]["enabled"] == False

# Keep id of the new job
id = r.json()["result"]["id"]

# Get job
r = requests.get(logpush_url + "/jobs/%s" % id, headers=headers)
print(r.status_code, r.text)
assert r.status_code == 200

# Get all jobs for a zone
r = requests.get(logpush_url + "/jobs", headers=headers)
print(r.status_code, r.text)
assert r.status_code == 200
assert len(r.json()["result"]) > 0

# Update job
r = requests.put(logpush_url + "/jobs/%s" % id, headers=headers,
data=json.dumps({"enabled":True}))
print(r.status_code, r.text)
assert r.status_code == 200
assert r.json()["result"]["enabled"] == True

# Delete job
r = requests.delete(logpush_url + "/jobs/%s" % id, headers=headers)
print(r.status_code, r.text)
assert r.status_code == 200
```

□□□ #2
□ □□□ □□□ 2026-11-01 02:52:48 UTC
□ □□□ □□□ 2026-11-01 02:52:49 UTC