Design Document: Mechanism to fail CD when env var are missing

Issue	Create a mechanism to fail the CD when required environment variables are missing · Issue #257 · Real-Dev-Squad/discord-slash-commands (github.com)
Delivery Date	16th Oct 2024
Design Status	Approved
Document & Feature Owner	Saitharun Burra

Reviewer	**Date**	**Action**
Tejas	10-14-2024	Approved

Table of Content

Overview

This document proposes a solution to check if the required environment variables are configured during the deployment Github Action workflow run and fail the CD pipeline workflow when they are missing.
The current configuration in Github Actions of both staging (Register and deploy Slash Commands) and production (Register and deploy Slash Commands) environments does not fail when the required environment variables are missing.

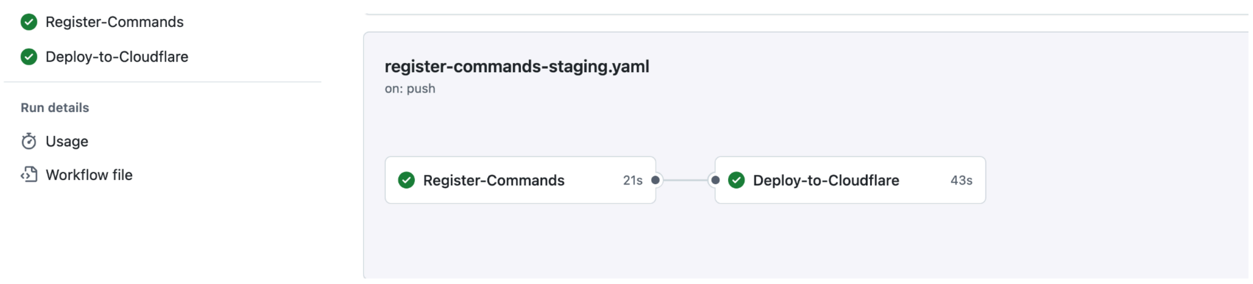
Current Behavior

- Currently all the environment variables are being configured per environment (i.e. staging, production, etc) as environment secrets, which are injected as environment variables in each job
- Even when some of the environment variables are missing, the Github Action finishes the deployment successfully, but the logs contain errors related to the missing environment variables.

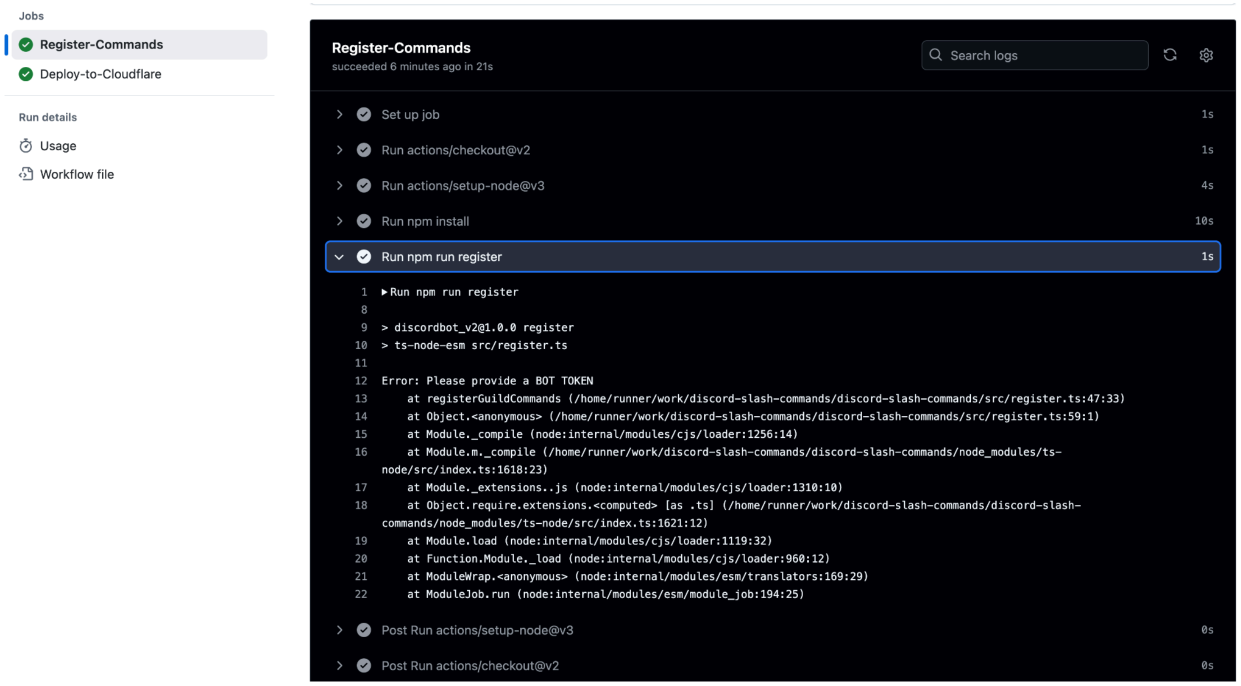
  Sample run (tested on a forked repo in my github account on the `test-main` branch):

  Sample Github Action run in staging on forked repo

  The above link points to a Github Action workflow which finished successfully when some of the env var were missing

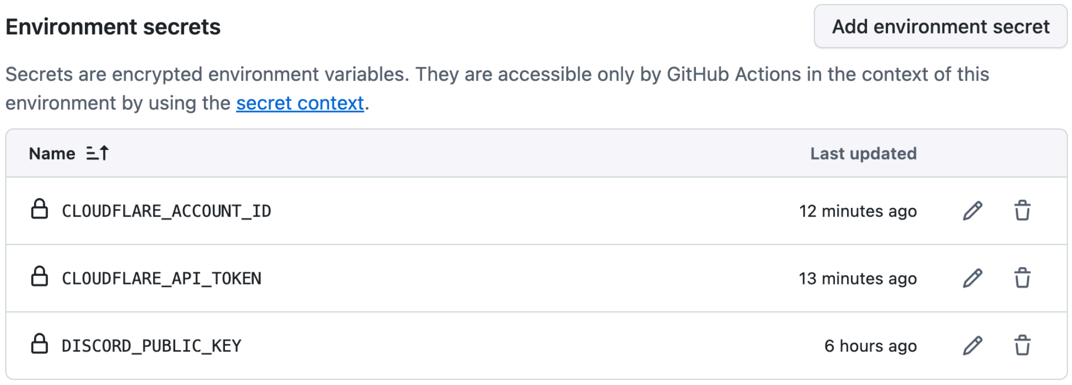


- But upon checking the logs in Register-Commands, there is an error related to the missing BOT TOKEN which is passed as DISCORD_TOKEN environment variable



```
Set up job                                   1s
Run actions/checkout@v2                      1s
Run actions/setup-node@v3                    4s
Run npm install                              10s
Run npm run register                         1s

▸Run npm run register

> discordbot_v2@1.0.0 register
> ts-node-esm src/register.ts

Error: Please provide a BOT TOKEN
    at registerGuildCommands (/home/runner/work/discord-slash-commands/discord-slash-commands/src/register.ts:47:33)
    at Object.<anonymous> (/home/runner/work/discord-slash-commands/discord-slash-commands/src/register.ts:59:1)
    at Module._compile (node:internal/modules/cjs/loader:1256:14)
    at Module.m._compile (/home/runner/work/discord-slash-commands/discord-slash-commands/node_modules/ts-
node/src/index.ts:1618:23)
    at Module._extensions..js (node:internal/modules/cjs/loader:1310:10)
    at Object.require.extensions.<computed> [as .ts] (/home/runner/work/discord-slash-commands/discord-slash-
commands/node_modules/ts-node/src/index.ts:1621:12)
    at Module.load (node:internal/modules/cjs/loader:1119:32)
    at Function.Module._load (node:internal/modules/cjs/loader:960:12)
    at ModuleWrap.<anonymous> (node:internal/modules/esm/translators:169:29)
    at ModuleJob.run (node:internal/modules/esm/module_job:194:25)

Post Run actions/setup-node@v3               0s
Post Run actions/checkout@v2                 0s
```

- Below are the only environment secrets configured for staging environment during the above Github Action run. (BOT TOKEN, i.e. DISCORD_TOKEN is missing.)



Environment secrets

Secrets are encrypted environment variables. They are accessible only by GitHub Actions in the context of this environment by using the secret context.

Name	Last updated
CLOUDFLARE_ACCOUNT_ID	12 minutes ago
CLOUDFLARE_API_TOKEN	13 minutes ago
DISCORD_PUBLIC_KEY	6 hours ago

Expected Behavior

- The CD pipeline should check if all the required environment variables are configured. And it should fail when any of them are missing.

Goals

- Add one more job to the Register and deploy Slash Commands Github Action in sequence before the jobs *Register-Commands* and *Deploy-to-Cloudflare*
- This job should run at the beginning of the Github Action before all other jobs in sequence to check if the required environment variables are configured and fail the workflow when they are not.

Solution

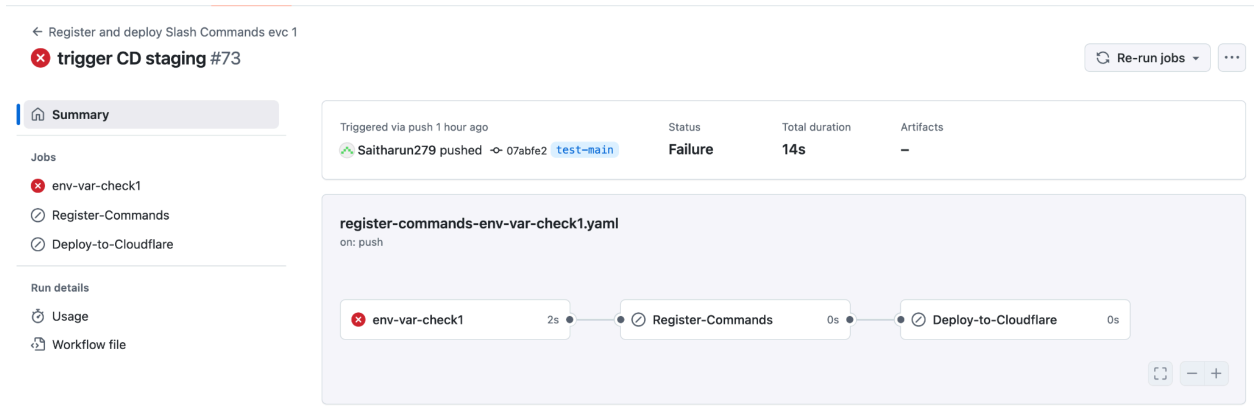
POC: POC implementation in forked repo

Add one more job:

- Adding one more job (called env-var-check1 in POC) to the existing Github Action which runs first in sequence before *Register-Commands* and *Deploy-to-Cloudflare.* This job should check the environment variables.
- It fails the workflow in case they are missing.
- If all the required environment variables are present, then the other jobs would execute.

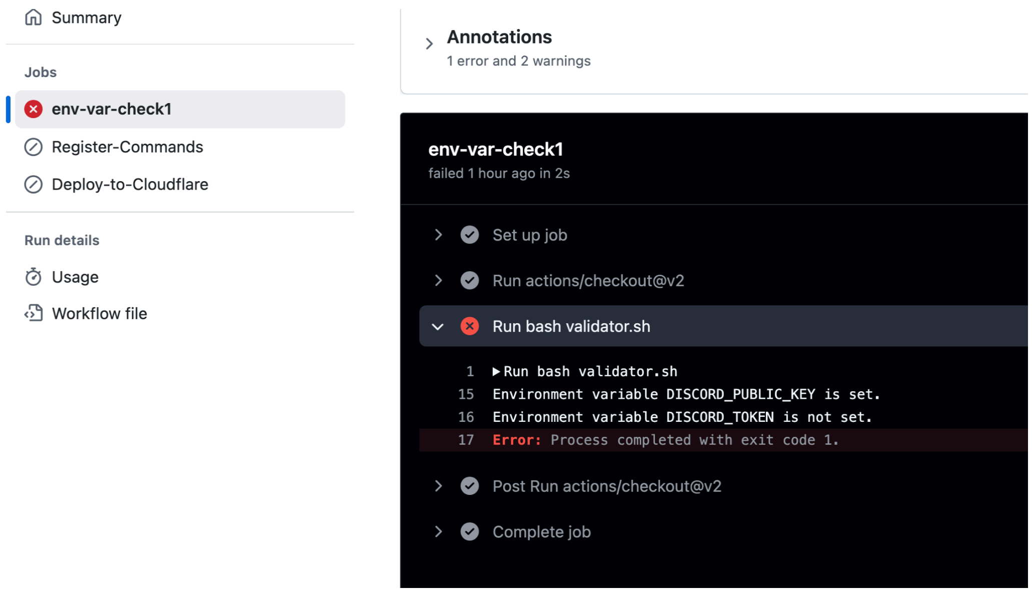
Below picture (from POC) shows the workflow graph, failed job when env var are missing

The suitable name for the job can be: *Environment-Variable-Check instead of env-var-check1* which was used for testing multiple such approaches.



The *Environment-Variable-Check* job configuration:

- The *Environment-Variable-Check* runs a bash script (called validator.sh in POC) with all the required environment variables of the entire workflow injected.
- The bash script reads the names of all the environment variables for the entire workflow from a file (called env-vars.txt in POC)
- Iterates through each of them and checks if they are empty.
- It exits with code 1, if any of the environment variables is empty.



```
▶Run bash validator.sh
Environment variable DISCORD_PUBLIC_KEY is set.
Environment variable DISCORD_TOKEN is not set.
Error: Process completed with exit code 1.
```

Adding one more job at the beginning of the workflow which needs to be finished successfully before the execution of the other jobs in sequence.
This way all the required environment variables can be checked only once at the beginning.

As per the current POC implementation, the file which contains the names of the environment variables has been hard coded in the bash script for now, since currently all the environment variables names are the same in both staging and production environments.
In the future if they require different variables, 2 different files can be created to store variable names per environment. And the file name can be passed as input to the bash script while running the bash script in the respective environment's Github Action.

Conclusion

- To check if required environment variables are present, a new job can be added at the beginning of the workflow. This new job runs a bash script which reads the environment variables from a file. Iterates and checks if each environment variable is set. Raises an error & exits execution otherwise.