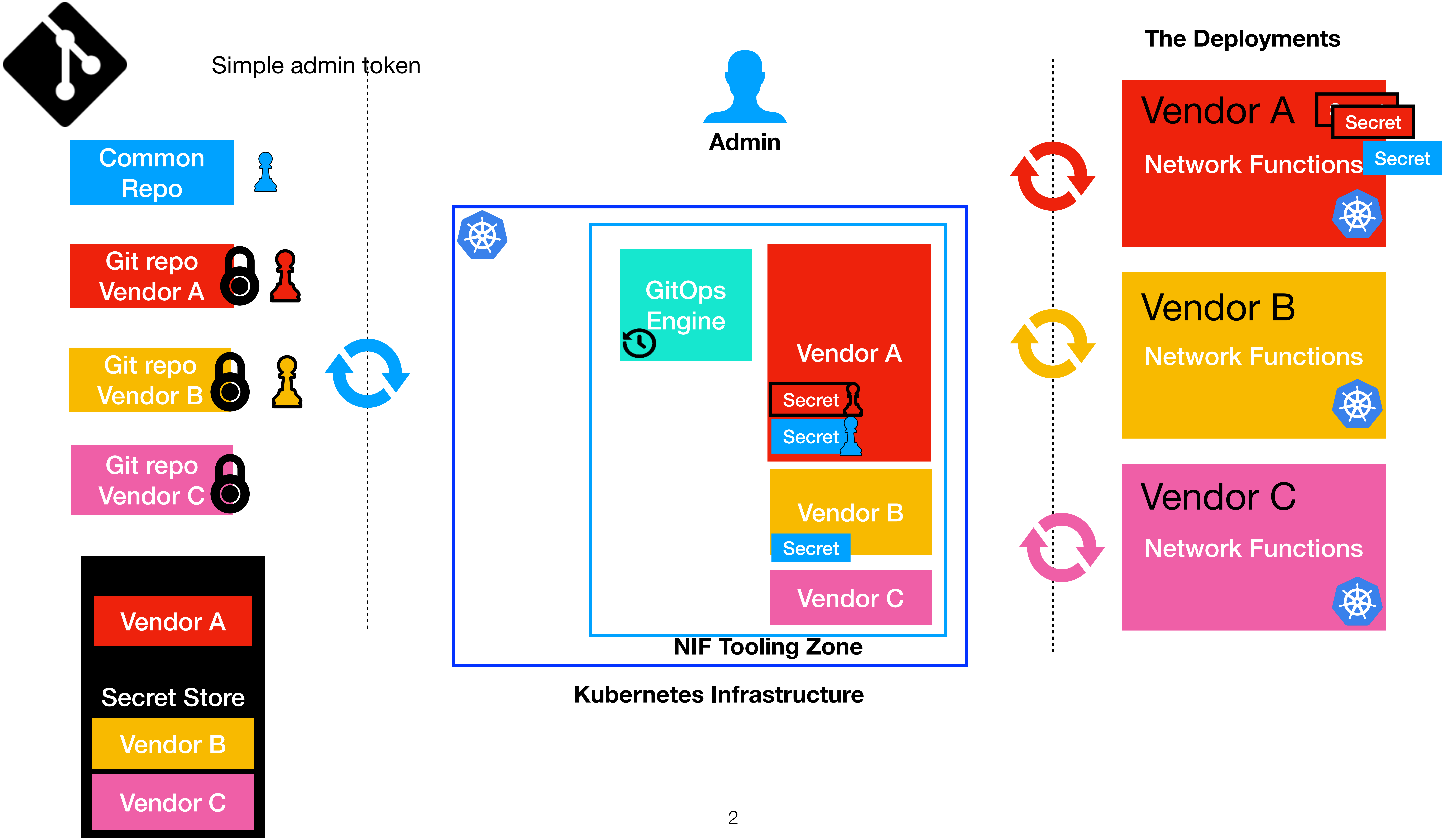
Token Management in GitOps

Token: The Forgettable Security Sidekick

A **token** is like a security superhero—easy to create and ready to save the day! In the world of **GitOps**, tokens are mandatory for smooth operations, ensuring that everything runs securely. However, just like that one friend who always forgets their keys, tokens need regular renewals with the right permissions. So, while they might be your best ally in the battle against unauthorized access, don't be surprised if you find yourself scrambling to remember their rights before they expire!



Simple admin token

Admin

The Deployments

Common Repo

Git repo Vendor A

Git repo Vendor B

Git repo Vendor C

Vendor A

Secret Store

Vendor B

Vendor C

GitOps Engine

Vendor A

Secret

Secret

Vendor B

Secret

Vendor C

NIF Tooling Zone

Kubernetes Infrastructure

Vendor A

Secret

Network Functions

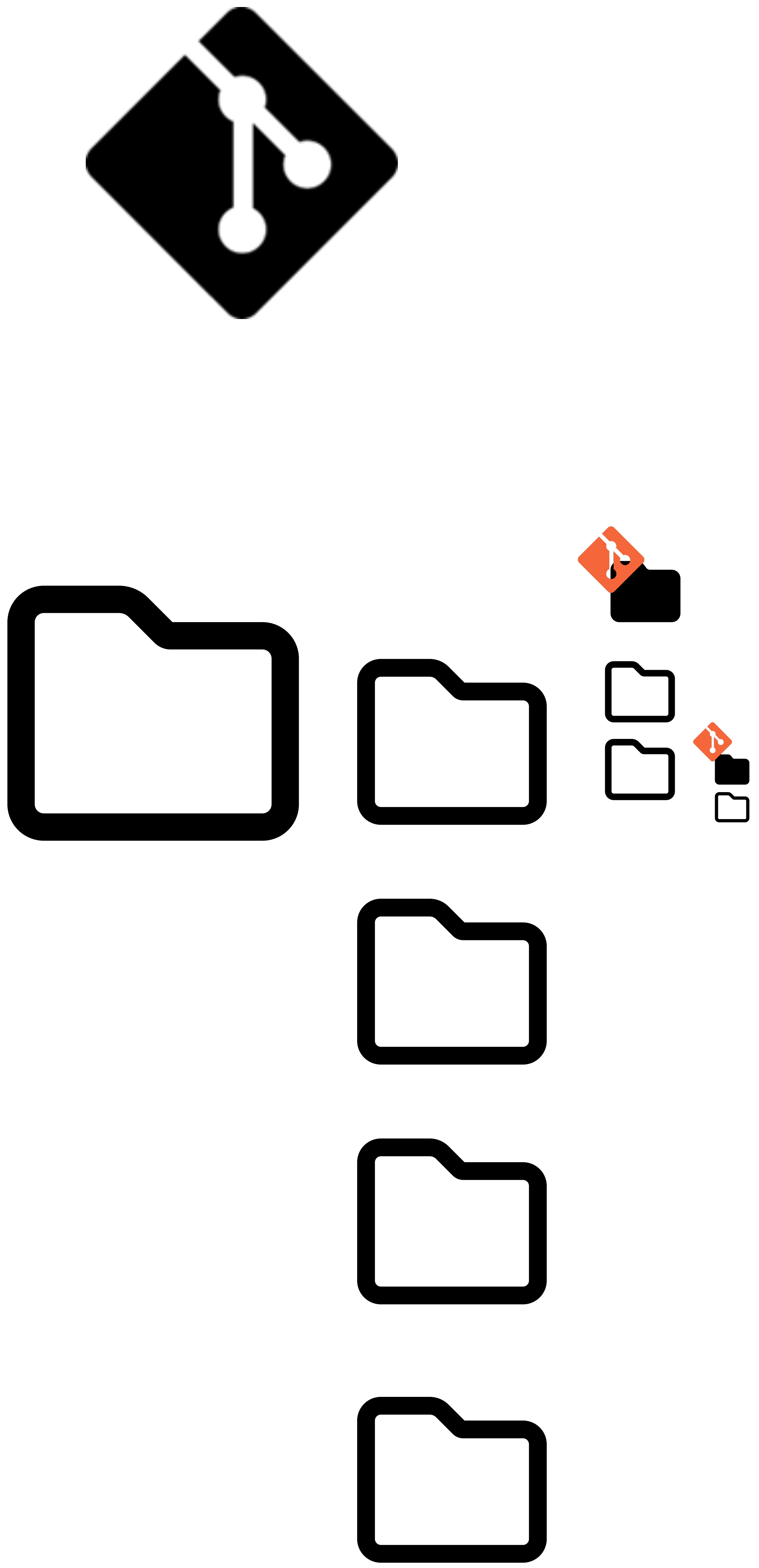
Secret

Vendor B

Network Functions

Vendor C

Network Functions

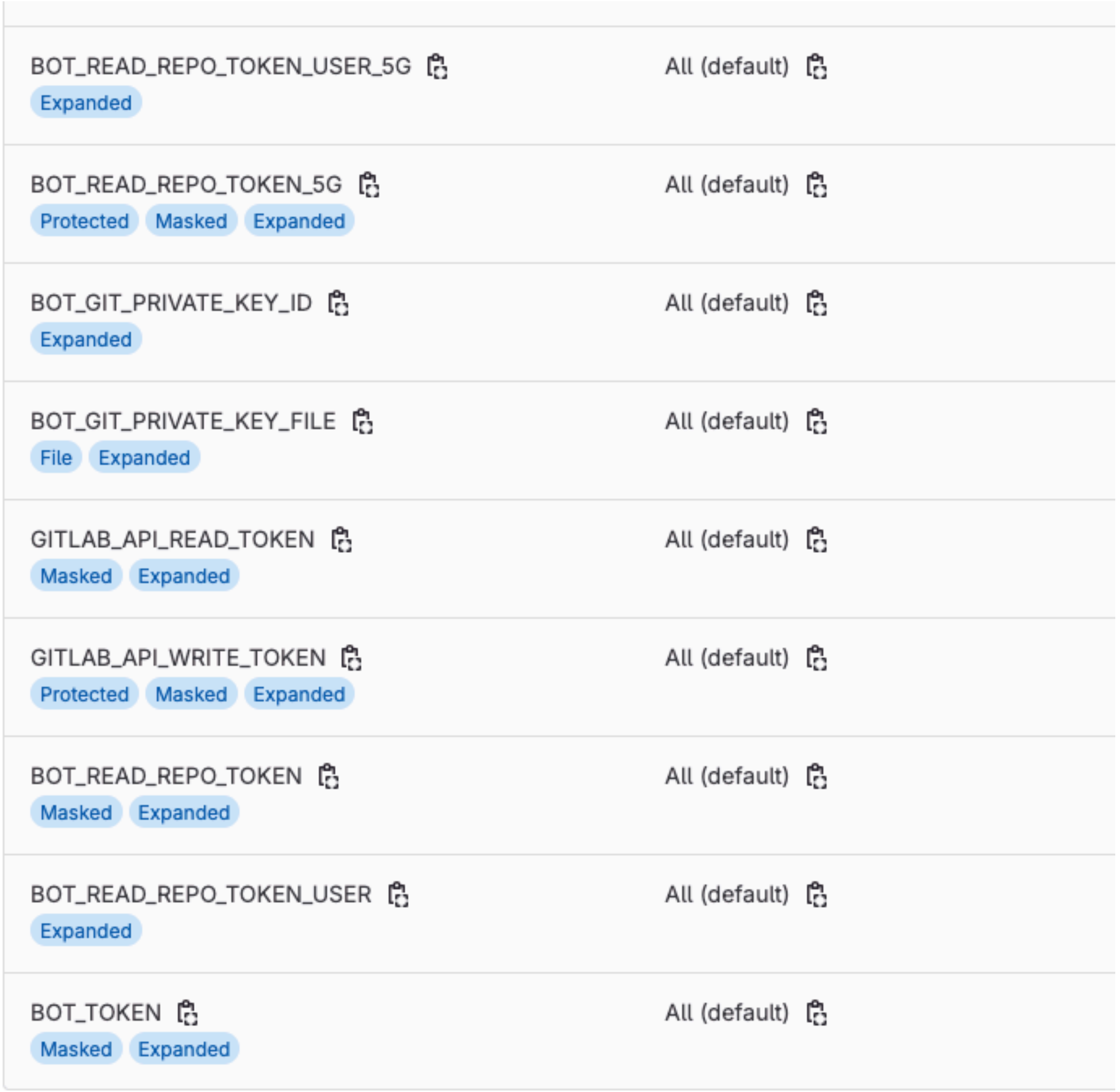


Multi-Repo architecture ~100 repositories for outs
GitOps tooling ~ 200 tokens

For a Vendor functions usually ~1-10 Access token / function:

- private repos
- Security scans
- Semantic release
- Renovate

Security rule: **less privilege**
=> limit nb of group token and
Declare token per repository



BOT_READ_REPO_TOKEN_USER_5G Expanded	All (default)
BOT_READ_REPO_TOKEN_5G Protected Masked Expanded	All (default)
BOT_GIT_PRIVATE_KEY_ID Expanded	All (default)
BOT_GIT_PRIVATE_KEY_FILE File Expanded	All (default)
GITLAB_API_READ_TOKEN Masked Expanded	All (default)
GITLAB_API_WRITE_TOKEN Protected Masked Expanded	All (default)
BOT_READ_REPO_TOKEN Masked Expanded	All (default)
BOT_READ_REPO_TOKEN_USER Expanded	All (default)
BOT_TOKEN Masked Expanded	All (default)

- https://gitlab.tech.orange/oln/nif/cd/tools/gitlab-as-code/-/blob/1-add-initial-project/examples/test.yaml?ref_type=heads