

## Repozytoria kodu

Tworzenie zasobów CICD zaczniemy od utworzenia repozytoriów kodu

Potrzebujemy 2 repozytoria kodu:

1. Kod infrastruktury dla aplikacji (wraz z parametrami dla poszczególnych środowisk aplikacyjnych)
2. Kod infrastruktury dla narzędzi CICD, za pomocą których będziemy testować, budować i wdrażać infrastrukturę pod aplikację

Parametry dla środowisk można trzymać w odrębnym repozytorium, albo w jednym repozytorium razem z kodem infrastruktury.

### Typy środowisk

W parametrach szablonów dotyczących infrastruktury pod aplikację mamy parametr Stage. Oznacza to typ środowiska pod aplikację. W tym projekcie używamy oznaczeń dev, test i prod dla każdego z tych środowisk.

Ale kod infrastruktury - jak każdy inny kod - musi zostać przetestowany zanim zostanie użyty "na produkcji". Dlatego w parametrach dotyczących zasobów CICD znajdziesz parametr SharedStage i inne oznaczenia środowisk shared-dev, shared-test i shared. Są to środowiska developerskie, testowe i produkcyjne dla kodu tworzącego zasoby CICD oraz kodu infrastruktury. Dopiero przetestowany kod infrastruktury (w różnych konfiguracjach) może być używany do tworzenia zasobów na środowiskach dev, test i prod aplikacji.

W szablonach dotyczących zasobów CICD znajdziesz też inny parametr - TargetStage - jest to parametr określający docelowe środowisko pod aplikację, którego infrastruktura zostanie zbudowana za pomocą danego pipeline.

## Repozytoria kodu

Dla repozytoriów kodu parametr SharedStage zawsze powinien mieć wartość shared.

Rozpakuj do odrębnego folderu paczkę repositories.zip i utwórz repozytoria kodu (jak każdy inny zasób w AWS - za pomocą AWS CloudFormation)

### Repozytorium kodu dla CICD

Pliki

```
repositories/cicd/templates/repository.yaml  
repositories/cicd/parameters/repository-cicd-shared.json  
repositories/cicd/commands/deploy-repository-cicd.sh
```

### Repozytorium kodu infrastruktury pod aplikację

Pliki

```
repositories/cicd/templates/repository.yaml
repositories/cicd/parameters/repository-infra-shared.json
repositories/cicd/commands/deploy-repository-infra.sh
```

## Wstępna konfiguracja repozytorium i utworzenie klucza SSH

Kroki niezbędne do korzystania z AWS CodeCommit są opisane w dokumentacji:

[Linux, macOS, Unix Windows](#)

Jednym z nich jest utworzenie klucza SSH. Klucz należy następnie dodać w konsoli AWS:

Usługa IAM => Users => wybieramy użytkownika => Security Credentials => Upload SSH public key

## Klonowanie repozytoriów

Po utworzeniu repozytoriów należy je sklonować:

```
git clone ssh://git-codecommit.eu-west-1.amazonaws.com/v1/repos/memes-generator-shared-cicd-cicd
git clone ssh://git-codecommit.eu-west-1.amazonaws.com/v1/repos/memes-generator-shared-cicd-infra
```

Repozytoria z kodem powinny być przechowywane w odrębnych katalogach

## Utworzenie głównych gałęzi (branchy) oraz pliku .gitignore

Następnie trzeba utworzyć główne gałęzie. Po przejściu do katalogu w danym repozytorium wykonaj polecenie `git checkout -b main`. Następnie utwórz plik `.gitignore` w którym zdefiniujesz, jakie pliki nie będą uwzględniane w systemie kontroli wersji. Przykład takiego pliku znajdziesz tu: `repositories/cicd/other/.gitignore` Plik powinien być umieszczony w folderze głównym danego repozytorium.

Dodaj, a następnie zakomituuj plik i wypchnij kod do zdalnego repozytorium:

```
git add .
git commit -m "Initial commit"
git push origin main
```

Tę czynność wykonaj dla każdego repozytorium.

## Utworzenie branchy development w repozytorium kodu infrastruktury

Branch development będzie służyć w każdym przypadku do integracji kodu infrastruktury. Kod z tego brancha jest już wdrażany za pomocą narzędzi CICD. Odpowiada więc środowisku `shared-test`

## Uwaga

Kod infrastruktury na środowisko `shared-dev` może być wdrażany za pomocą AWS CLI. W procesie developmentu ten kod jest komitowany do branchy `feature/feature-name`.

Następnie należy zrobić merge do brancha development - kod z tego brancha jest testowany za pomocą narzędzi CICD.

Po przetestowaniu budowania kodu infrastruktury (w różnych konfiguracjach) trafia na branch main. Ten kod może już służyć do budowy infrastruktury dla środowisk aplikacyjnych.

Branch development tworzymy z brancha main za pomocą polecenia `git checkout -b development`.

## **Dodanie kodu infrastruktury do repozytorium**

Ponieważ kod infrastruktury przeszedł już fazę developmentu, możemy go od razu dodać do brancha development.

W tym celu rozpakuj do repozytorium kodu infrastruktury kod z paczki `memes-generator-infra.zip`. Struktura projektu będzie wyglądać tak:

```
\memes-generator-shared-cicd-infra
  \memes-generator
    \application
    \cicd
    \data
    \network
    \operations
```

Po rozpakowaniu dodaj, zakomituuj i wypchnij kod do zdalnego repozytorium:

```
git add .
git commit -m "Added developed infrastructure code"
git push origin development
```

## **Dodanie kodu z zasobami CICD do repozytorium**

Kod z zasobami CICD znajduje się obecnie w fazie developmentu, więc parametr `SharedStage` będzie miał wartość `shared-dev`. Po upewnieniu się, że wszystkie zasoby CICD działają poprawnie, możesz je wypromować na wyższe środowiska (`shared-test` i `shared`)

Tu podobnie należy utworzyć branch `development` (z brancha `main`), a branche dla nowych zasobów CICD - z brancha `development` Utwórz więc branch `feature/cicd-resources-for-infra`.

Kod z paczki `memes-generator-cicd.zip` możesz od razu dodać do brancha `feature/cicd-resources-for-infra`. Nowy kod (lub modyfikacje obecnego) twórz w ramach oddzielnych branchy.

## **Git - więcej informacji**

Więcej informacji o systemie kontroli wersji `git` znajdziesz tu:

[Ściągawka Dokumentacja Pro-Git](#)