

## Tworzenie jumphost dla środowiska pod aplikację “Generator Memów”

### Struktura projektu

1. Utwórz katalog (katalog główny), w którym będziesz trzymać wszystkie pliki używane na tym kursie (np. aws-w-praktyce). Jeśli masz już taki katalog, możesz pominąć ten krok.
2. Struktura katalogów będzie wyglądać tak:

```
\${Root-directory}
  \${Project}
    \${Component}
      \commands
      \parameters
      \templates
    \${Component}
      \commands
      \parameters
      \templates
  (...)
```

Dla aplikacji “Generator Memów” - na tym etapie projektu - będziesz mieć następującą strukturę:

```
\${Root-directory}
  \memes-generator
    \network
      \commands
      \parameters
      \templates
    \operations
      \commands
      \parameters
      \templates
```

W kolejnych etapach pojawią się kolejne komponenty, szablony i inne pliki, więc struktura projektu będzie wyglądać mniej więcej tak:

```
\${Root-directory}
  \memes-generator
    \application
      \commands
      \parameters
      \templates
    \data
      \commands
      \parameters
```

```
\scripts
\templates
\network
  \commands
  \parameters
  \templates
\operations
  \commands
  \parameters
  \templates
(...)
```

**Na kolejnych etapach dodawaj niezbędne pliki do już istniejących albo utwórz cały projekt od nowa. W “paczce” dołączanej do lekcji znajduje się komplet plików dla całego projektu - od początku aż do aktualnego etapie projektu**

### Co znajduje się w folderach

commands - przykładowe komendy, których możesz użyć do tworzenia stacków  
parameters - pliki json z parametrami dla danego stacka (z reguły dla danego typu środowiska - dev, test, prod - będzie odrębny plik z konfiguracją, a nazwa środowiska jest w nazwie pliku)  
templates - szablony do tworzenia infrastruktury

### Jak używać plików z przykładowymi poleceniami

1. W terminalu wybierz katalog główny projektu.
2. Na samym początku ustaw zmienne PROJECT, STAGE i REGION.
3. Kolejne zmienne będziesz ustawiać już dla konkretnych stacków: COMPONENT, STACK, TEMPLATE i PARAMETERS.
4. Na podstawie tych zmiennych tworzone są kolejne, czyli TEMPLATE\_FILE i PARAM\_FILE
5. Utwórz, wyświetl, a następnie wykonaj polecenie \$deploy.

Jeśli stworzysz kilka stacków, czynności z pkt. 3, 4 i 5 trzeba powtórzyć dla każdego stacka.

### Stack 1 - Jumphost

Pliki do tej części znajdują się w folderze operations.

#### Pliki

Szablon: memes-generator/operations/templates/jumphost.yaml Plik z parametrami dla środowiska dev: memes-generator/operations/parameters/jumphost-dev.json Plik z przykładowymi poleceniami: memes-generator/operations/commands/deploy-jumphost.sh

### Tworzone zasoby

Utworzymy 3 zasoby:

1. Instancja EC2
2. Instance profile
3. Rola dla instancji

#### **Uwagi dotyczące zasobów**

**Zanim utworzysz stack, utwórz klucz dla instancji o takiej nazwie, jak jest wskazana w pliku z parametrami**

Klucz możesz utworzyć i pobrać w konsoli AWS albo za pomocą AWS CLI:

```
aws ec2 create-key-pair \  
  --key-name $PROJECT-$STAGE-jumphost-key
```

[Dokumentacja do tego polecenia](#)