Testowanie utworzonej infrastruktury

Load Balancing

1. Odszukaj URL utworzonego load balancera (np. w Outputs stacka z load balancerem albo w parametrze SSM)
2. Wykonaj następujące zapytania (ale tym razem z terminala na Twoim komputerze - na tym etapie aplikacja powinna już być dostępna z internetu). W miejsce $ALB_URL wstaw adres load balancera:

```
curl -X GET http://$ALB_URL:8080/actuator/health curl -X POST --data '{"text":"Jumphost meme!"}' -H "Content-Type: application/json" http://$ALB_URL:8080/memes curl -X GET http://$ALB_URL:8080/memes?maxItems=5
```

Rezultaty powinny być podobne do tych, które otrzymaliśmy testując aplikację z instancji.

Możesz pobrać utworzonego mema (zastąp `IMAGE_FILE_NAME.jpg` nazwą pliku z Twoim memem):

```
curl http://$ALB_URL:8080/download/IMAGE_FILE_NAME.jpg -o meme.jpg
```

Testowanie autoskalowania instancji

Usunięcie instancji

Pierwszy test, jaki wykonasz, to usunięcie instancji należącej do AutoScaling Group.

1. Wyświetl ID i status instancji posiadające wartości tagów: `Project=memes-generator Stage=dev Component=application`
2. Wykonaj polecenie `aws ec2 terminate-instance` podając ID uruchomionej instancji
3. Ponownie wyświetl uruchomione instancje - liczba uruchomionych instancji z tymi tagami powinna (po kilku chwilach) spaść do 0.
4. Odczekaj kilka minut
5. Ponownie wyświetl ID i status instancji posiadające ww. wartości tagów.

Stress test instancji

Na instancjach, na których zainstalowany jest agent SSM i które mają odpowiednie uprawnienia do komunikacji z AWS System Manager można wykonywać polecenia. Polecenia te (lub grupy poleceń) są zdefiniowane w tzw. dokumentach. Tu użyjemy dokumentu do wykonania stress testu na wszystkich instancjach, gdzie wartości tagów to: `Project=memes-generator Stage=dev Component=application`

Polecenie, którego trzeba użyć:

```
aws ssm send-command \
    --document-name "AWSFIS-Run-CPU-Stress" \
    --document-version "2" \
    --targets '[{"Key":"tag:Project","Values":["memes-
generator"]},{"Key":"tag:Stage","Values":["dev"]},
{"Key":"tag:Component","Values":["application"]}]' \
    --parameters '{"CPU":["0"],"InstallDependencies":
["True"],"DurationSeconds":["600"]}' \
    --timeout-seconds 600 \
    --max-concurrency "50" \
    --max-errors "0" \
    --region eu-west-1
```

Polecenie to uruchomi kod z dokumentu na wszystkich instancjach wybranych na podstawie tagów. Stress test będzie trwał `600` sekund czyli 10 minut. Jest to wystarczający czas, żeby uruchomił się alert, który następnie wywołuje akcję `Scale Out`.

Po uruchomieniu polecenia odczekaj około 7 minut, a następnie zobacz, czy liczba instancji z aplikacją zwiększyła się do 2 (w konsoli AWS lub AWS CLI).

Ponownie odczekaj - około 20 minut. W tym czasie stress test zakończy się, średnie zużycie procesora spadnie poniżej zdefiniowanego progu, a następnie uruchomi się kolejny alert, który wywoła akcję - tym razem `Scale In`. W efekcie liczba instancji powróci do poprzedniej (1)

Dla chętnych:

1. Obejrzyj metryki dla instancji przed, w trakcie i po zakończeniu stress testu (w konsoli AWS) - w usłudze AWS CloudWatch
2. Obejrzyj stan alertów przed, w trakcie i po zakończeniu stress testu (w konsoli AWS - w usłudze AWS CloudWatch)
3. Obejrzyj listę akcji dotyczących autoskalowania