

XXXXXXXXXX

Projekt

„Wysoka dostępność aplikacji w
chmurze publicznej GCP”

dla

Mountkirk Games

Opracowany przez:

XXXXX

XXXXX

XXXXX

Wersja dokumentu: 0.1

Identyfikator projektu: MountkirkGames20191228

Data opracowania: 2019-12-28

Dane Dokumentu

Nazwa Projektu:	Wysoka dostępność aplikacji w chmurze publicznej GCP		
Opracowany Przez:	_____	Nr Wersji:	0.1
Rola:	Kierownik Projektu		

Rola:	Konsultant Techniczny	Data Wersji:	2019-12-28

Rola:	Konsultant Techniczny		
Sprawdzony Przez:			

Rozdzielnik

Od	Data	Telefon/Faks/Email

Do	Działanie*	Termin	Telefon/Faks/Email

Rodzaje Działania: Zatwierdzenie, Przegląd, Do Wiadomości, Do Akt, Wymagane Działanie, Uczestnictwo w Zebraniu, Inne (proszę określić, jakie?)

Historia Zmian

Nr Wersji	Data Wersji	Zmiany Wprowadził(a)	Opis Zmian	Recenzent	Status
0.1	2019-12-28	Robert Kostecki	Utworzenie dokumentu		

Spis treści

1.	Opis środowiska	4
2.	Wymagania.....	4
3.	Założenia	4
4.	Monitoring środowiska.....	4
5.	Koncepcja rozwiązania.....	5
5.1.	Architektura rozwiązania.....	5
5.2.	Zapewnienie wysokiej dostępności.....	6
5.3.	Skalowanie aplikacji.....	8
5.4.	Aktualizacja aplikacji.....	8
6.	PoC.....	9
7.	Harmonogram prac.....	11
8.	Szkolenia.....	11
9.	Testy odbiorowe.....	11

Spis rysunków

Rysunek 1.	Logiczny schemat architektury rozwiązania.....	5
Rysunek 2.	Wysoka dostępność - dwa regiony A-A.....	6
Rysunek 3.	Rysunek 2. Wysoka dostępność - 2 regiony A-P.....	7
Rysunek 4.	Wysoka dostępność - cztery regiony A-A.....	7
Rysunek 5.	Procedura aktualizacji aplikacji.....	8

1. Opis środowiska.

Zarząd MountKirk zdecydował, że wypuszczając swoją nową grę na rynek skorzystają z gotowego rozwiązania w Google Cloud jakim jest Compute Engine. firma postanowiła również wprowadzić nową aplikację dla graczy, którą również chcą oprzeć na tym rozwiązaniu.

Gra okazała się wielkim sukcesem, a aplikacja również przyniosła oczekiwane rezultaty. Firma nie spodziewała się, aż tak wielkiej ilości nowych osób z całego świata, którzy będą chcieli skorzystać z nowych funkcjonalności. W pewnym momencie całe środowisko stało się mało wydajne i użytkownicy końcowi zaczęli dostrzegać, że platformie doskwierają jakieś problemy.

Specjaliści starali się zwiększać na bieżąco parametry maszyn wirtualnych, aby podnieść wydajność, lecz było to mało efektywne oraz zajmowało masę czasu. Realizowany projekt ma dokonać optymalizacji i automatyzacji realizowanych zadań.

2. Wymagania.

MountKirk określił następujące wymagania:

- a) Na ten moment firma nie może zrezygnować z maszyn wirtualnych, dlatego nowa architektura musi korzystać z wirtualnych maszyn postawionych z niestandardowego obrazu, dostarczonego bezpośrednio przez firmę.
- b) Rozwiązanie musi dynamicznie skalować się w górę lub w dół w zależności od aktywności w grze - bez większej ingerencji specjalistów
- c) Gracze korzystający z funkcjonalności firmy pochodzą z całego świata, a w szczególności z Stanów Zjednoczonych oraz Europy. Poprzez odpowiednie umiejscowienie rozwiązania MountKirk chce zredukować opóźnienie jakie występuje dla osób łączących się z US.
- d) Rozwiązanie musi zapobiegać jakiegokolwiek przerwie w dostarczaniu funkcjonalności na wypadek awarii np. regionu Google Cloud.
- e) Rozwiązanie musi umożliwić łatwe i bezpiecznie wdrażanie nowych wersji oprogramowania do instancji bez konieczności wpływania na całe środowisko.

3. Założenia.

- a) Biorąc pod uwagę wymagania dotyczące aplikacji odpowiednim rozwiązaniem jest użycie managed instance groups (MIGs).
Użycie MIGs zaadresuje takie wymagania jak:
 - Wysoka dostępność,
 - Skalowalność.
 - Automatyzacja aktualizacji.
- a) Dla zapewnienia lepszej dostępności użyte zostaną grupy regionalne.
- b) Niskie opóźnienia dla graczy zostaną uzyskane poprzez umieszczenie grup w regionach w Europie oraz USA.
- c) Obsługa awarii regionu zostanie zrealizowana poprzez duplikację grupy do zapasowego regionu lub przekierowanie ruchu do istniejącego regionu. Wybór scenariusza zależnie od woli klienta.
- d) Bezpieczne wdrażanie nowych wersji zostanie zrealizowane canary testing/updates co zapewni możliwość wdrożenia aktualizacji dla wybranej grupy maszyn przetestowanie i podjęcie decyzji o wdrożeniu aktualizacji na pozostałe maszyny lub jej wycofanie.

4. Monitoring środowiska.

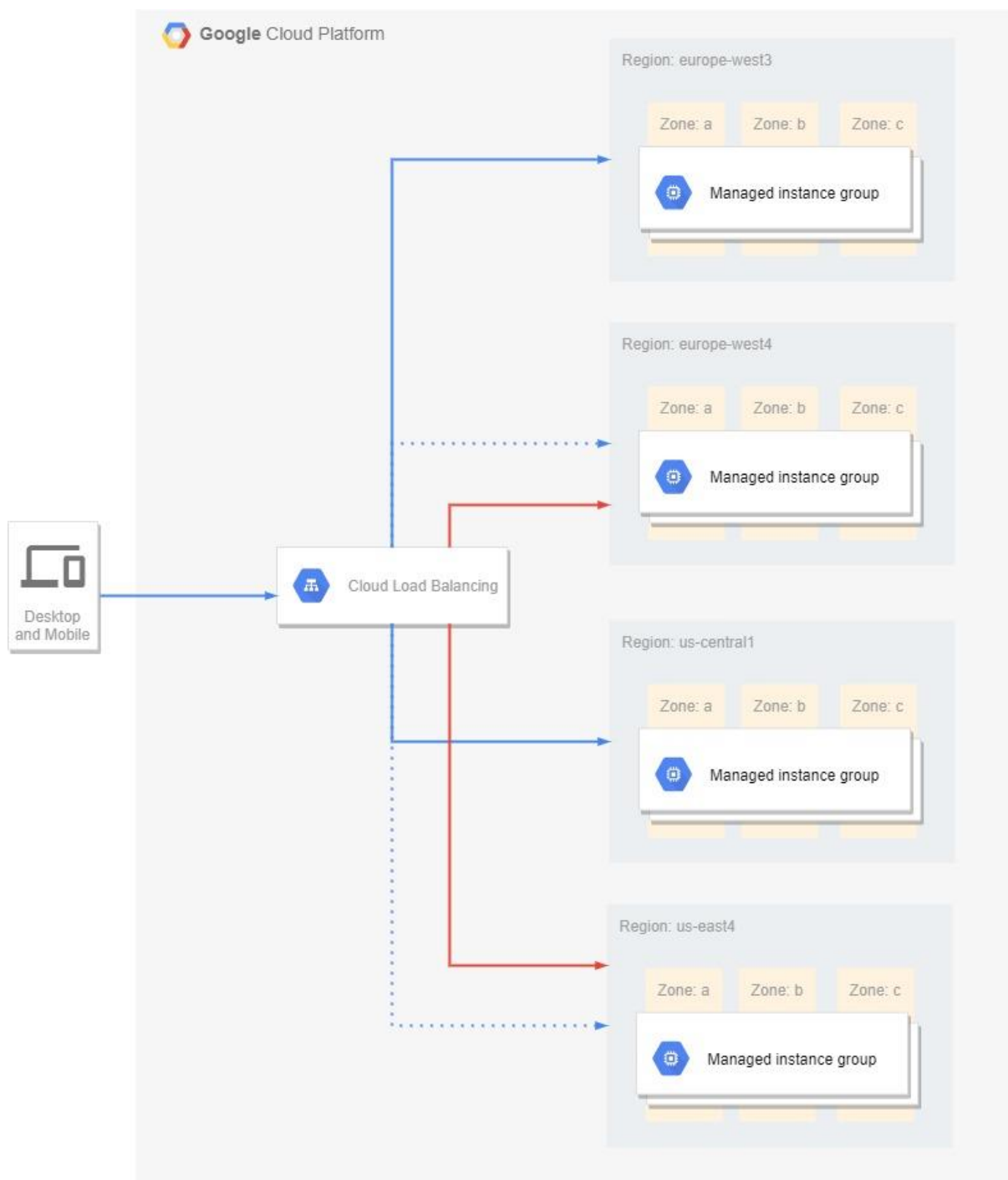
Przed rozwinięciem architektury w projekt techniczny należy dokonać analizy istniejącego środowiska ze szczególnym uwzględnieniem:

- a) Obciążenia środowiska.
- b) Ilości aktywnych instancji.
- c) Struktury aplikacji.

5. Koncepcja rozwiązania.

5.1. Architektura rozwiązania.

Zaproponowana architektura zakłada wykorzystanie podwójnych regionów, z których jeden jest w trybie pasywnym i zostanie wykorzystany w przypadku utraty regionu głównego.

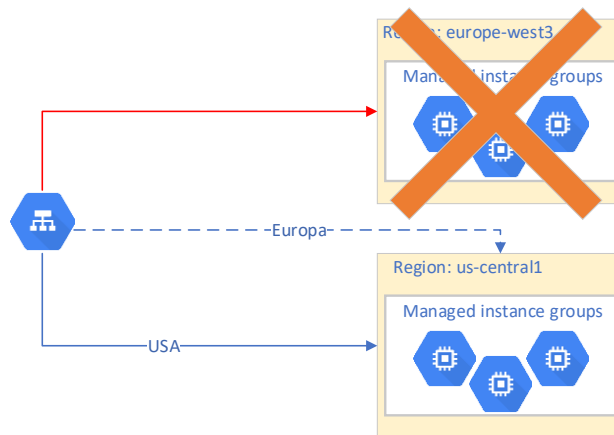


Rysunek 1. Logiczny schemat architektury rozwiązania.

5.2. Zapewnienie wysokiej dostępności.

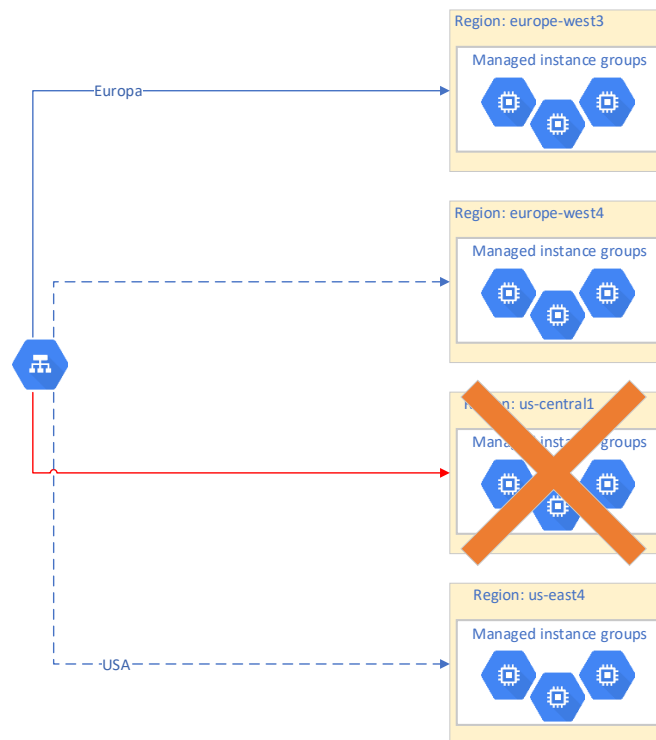
Dostępność aplikacji będzie realizowana na dwóch poziomach:

- a) Mechanizmy MIGs, które zagwarantują poprawną pracę wymaganej ilości maszyn wirtualnych:
 - Wykrywanie awarii serwera wirtualnego co powoduje stworzenie nowej instancji.
 - Sprawdzanie poprawności pracy aplikacji poprzez okresową weryfikację odpowiedzi zapytania http.
 - Rozdzielenie instancji aplikacji w wybranych regionach.
 - Rozrzucanie obciążenia za pomocą usługi Load Balancera.
- b) Mechanizmy Load Balancer-a, który zapewni dostępność w przypadku awarii regionu. Możliwe jest kilka scenariuszy różniące się kosztami.
 - HA oparte na dwóch regionach. Obie regiony obsługują klientów, w przypadku awarii jednego regionu drugi przejmuje całość ruchu.
Główne zalety niski koszt oraz szybkość przełączenia klientów.
Główna wada: wzrost opóźnień dla klientów w regionie, który uległ awarii, brak zapasu w przypadku problemów w pozostałym regionie. Na poniższym schemacie widzimy symulację awarii regionu europejskiego.



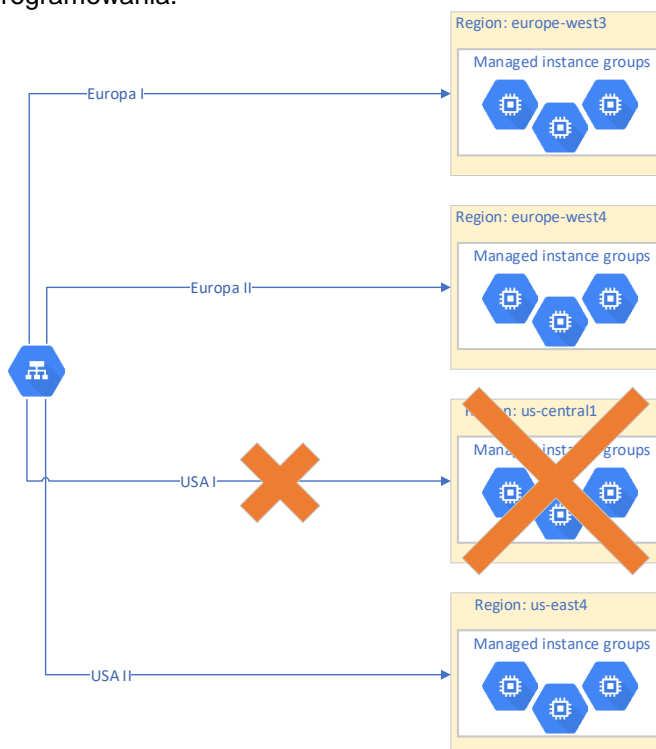
Rysunek 2. Wysoka dostępność - dwa regiony A-A.

- HA oparte na zdublowanych regionach na kontynentach z czego jeden z pary regionów jest w trybie pasywnym z minimalną ilością maszyn i zostanie wyskalowany w razie potrzeby. Główne zalety podwójne zabezpieczenie w przypadku awarii regionu bez pogorszenia opóźnień, zapewni pracę aplikacji nawet przy awarii 3 regionów jednocześnie.
Główne wady nieznacznie wyższy koszt, wyższa poziom komplikacji infrastruktury, możliwe zakłócenie w jakości świadczonych usług zanim system zapasowy się wyskaluje.



Rysunek 3. Rysunek 2. Wysoka dostępność - 2 regiony A-P.

- Możliwa jest również architektura hybrydowa, kiedy regiony są zdublowane i wszystkie przyjmują ruch. Takie rozwiązanie ograniczy zakłócenia w przypadku awarii jednego regionu, co wynika z tego, że ruch jest cały czas kierowany do „zapasowego regionu” i będzie łatwiej się skalować. W tym scenariuszu wzrasta komplikacja wykonywania aktualizacji oprogramowania.



Rysunek 4. Wysoka dostępność - cztery regiony A-A.

Ze względu na konieczność zachowania wysokiej dostępności oraz łatwości aktualizacji rekomendowane jest wykorzystanie architektury z pasywnymi duplikatami.

5.3. Skalowanie aplikacji.

Skalowanie grupy maszyn dla aplikacji będzie realizowane w sposób horyzontalny u oparciu o obciążenie CPU. W trakcie stabilizacji systemu możliwa zmiana metryk oddających specyficzne wymagania aplikacji,

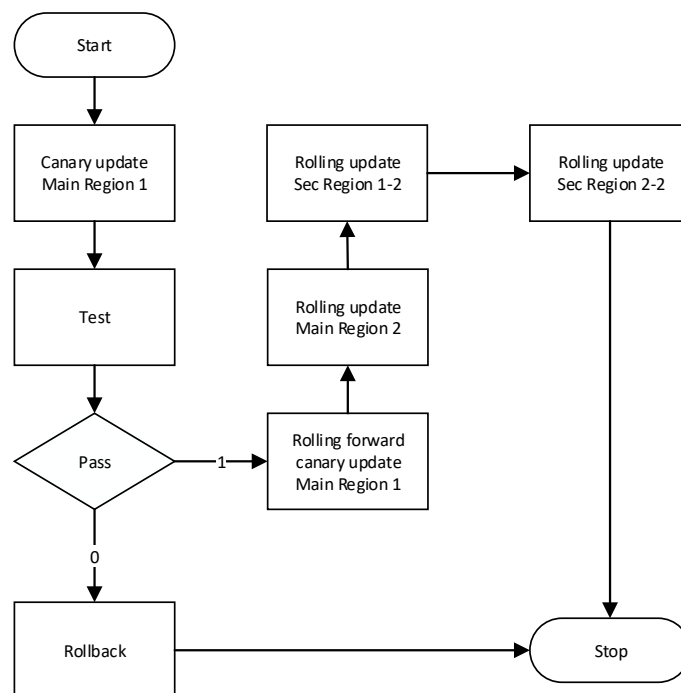
Skalowanie będzie realizowane w ramach grupy w górę i w dół. Parametry cooldown po inicjalizacji nowej maszyny, oraz satbilizacji przed skalowaniem w dół zostaną określne na podstawie obserwacji środowiska i aplikacji.

5.4. Aktualizacja aplikacji.

Realizacja aktualizacji planowana jest w dwóch głównych krokach.

- Wykonanie canary update w wybranym regionie.
Ten typ aktualizacji pozwala na update wybranej ilości instancji w celu wykonania testów na środowisku produkcyjnym. Zależnie od wyniku testów aktualizacje można rozszerzyć na resztę grupy lub ją wycofać.
- Weryfikacja poprawności aktualizacji.
- Wykonanie rolling update w pozostałych regionach.
W przypadku akceptacji aktualizacji, przeprowadza jest pełna aktualizacja pozostałych regionów.

Aktualizacja instancji maszyn wirtualnych wymaga wyłączenia aktualizowanej instancji oraz powołania nowej. Odpowiednia parametryzacja operacji ograniczy możliwy wpływ na użytkowników systemu. Parametry pozwalają między innymi, określić akceptowalną ilość niedostępnych instancji (maxUnavailable) oraz ilość instancji, które zostaną stworzone z wyprzedzeniem i zastąpią aktualizowane istniejące instancje (maxSurge). Parametry należy dobrać biorąc pod uwagę średnią ilość instancji uruchomionych podczas okna aktualizacji oraz istniejące quot-y na posiadanych zasobach. Odpowiednio dobrane parametry zminimalizują zakłócenia w pracy aplikacji przy założeniu jej bezstanowości.



Rysunek 5. Procedura aktualizacji aplikacji.

6. PoC.

Dla celów demonstracyjnych utworzona zostanie grupa w regionie europejskim w celu demonstracji procesu deploymentu, skalowania oraz aktualizacji środowiska.

- a) Przygotowanie template dla PoC
Utworzenie maszyny z Joomla (marketplace).
Utworzenie template z instancji Joomla

```
gcloud compute instance-templates create mountkirk-tmpl-1 \
  --source-instance=joomla-1-vm \
  --source-instance-zone=europe-west1-d
```


```
## Utworzenie maszyny z WordPress (marketplace)
## Utworzenie template z instancji Wordpress.
```

```
gcloud compute instance-templates create mountkirk-tmpl-2 \
  --source-instance=wordpress-1-vm \
  --source-instance-zone=europe-west1-b
```

```
## Wynik
$ gcloud compute instance-templates list
NAME                MACHINE_TYPE  PREEMPTIBLE  CREATION_TIMESTAMP
mountkirk-tmpl-1    g1-small      2019-12-29T09:19:27.514-08:00
mountkirk-tmpl-2    g1-small      2019-12-29T09:16:22.075-08:00
```

- b) Utworzenie MIGs regionalnego dla EU
gcloud compute instance-groups managed create mountkirk-mig-eu-west1 \
 --template mountkirk-tmpl-1 --base-instance-name mountkirk-eu-v1 \
 --size 3 --region europe-west1

<input type="checkbox"/>	Name ^	Zone	Instances	Template	Creation time	Recommendation	Autoscaling	In use by
<input type="checkbox"/>	mountkirk-mig-eu-west1	europe-west1 (3/3 zones)	3	mountkirk-tmpl-1	29.12.2019, 18:46:04		No configuration	
<input type="checkbox"/>	mountkirk-eu-v1-4jgg	europe-west1-d		mountkirk-mig-eu-west1	10.132.0.29 (nic0)	104.155.18.57	SSH	
<input type="checkbox"/>	mountkirk-eu-v1-dv4x	europe-west1-b		mountkirk-mig-eu-west1	10.132.0.28 (nic0)	35.205.225.170	SSH	
<input type="checkbox"/>	mountkirk-eu-v1-nfbm	europe-west1-c		mountkirk-mig-eu-west1	10.132.0.27 (nic0)	34.77.135.71	SSH	



- c) Ustawienie autoscalowania
gcloud compute instance-groups managed set-autoscaling mountkirk-mig-eu-west1 \
 --cool-down-period=90 \
 --target-cpu-utilization 0.2 \
 --max-num-replicas 6 \
 --region europe-west1

Name ^	Zone	Instances	Template	Creation time	Recommendation	Autoscaling	In use by
<input checked="" type="checkbox"/> mountkirk-mig-eu-west1	eu-west-1 (3/3 zones)	3	mountkirk-tmpl-1	29.12.2019, 18:46:04		On: Target CPU utilization 20 %	

- d) Obciążenie instancji
`sudo stress --cpu 8 --timeout 180`

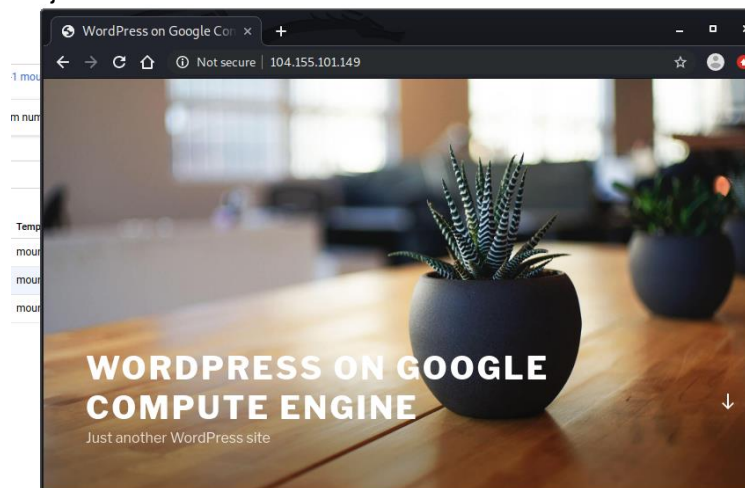
<input type="checkbox"/>	mountkirk-eu-v1-0bcl	eu-west-1-d		10.132.0.33 (nic0)	35.187.88.114	SSH	⌵	⋮
<input checked="" type="checkbox"/>	mountkirk-eu-v1-cbx1	eu-west-1-c	mountkirk-mig-eu-west1	10.132.0.31 (nic0)	104.155.101.149	SSH	⌵	⋮
<input checked="" type="checkbox"/>	mountkirk-eu-v1-dv4x	eu-west-1-b	mountkirk-mig-eu-west1	10.132.0.28 (nic0)	35.205.225.170	SSH	⌵	⋮
<input checked="" type="checkbox"/>	mountkirk-eu-v1-nfbm	eu-west-1-c	mountkirk-mig-eu-west1	10.132.0.27 (nic0)	34.77.135.71	SSH	⌵	⋮
<input checked="" type="checkbox"/>	mountkirk-eu-v1-phmr	eu-west-1-d	mountkirk-mig-eu-west1	10.132.0.30 (nic0)	104.155.18.57	SSH	⌵	⋮
<input checked="" type="checkbox"/>	mountkirk-eu-v1-z85z	eu-west-1-b	mountkirk-mig-eu-west1	10.132.0.32 (nic0)	34.76.68.3	SSH	⌵	⋮

- e) Canary update/testing , podmiana jednej instancji

```
gcloud compute instance-groups managed rolling-action start-update mountkirk-mig-eu-west1 \
  --version template=mountkirk-tmpl-1 \
  --canary-version template=mountkirk-tmpl-2,target-size=30% \
  --max-surge=3 --max-unavailable=0 --region europe-west1
```

<input checked="" type="checkbox"/>	mountkirk-eu-v1-0bcl	29.12.2019, 19:03:54	mountkirk-tmpl-1	eu-west-1-d	10.132.0.33 (nic0)	35.187.88.114	SSH	⌵
<input checked="" type="checkbox"/>	mountkirk-eu-v1-vpfj	29.12.2019, 19:24:36	mountkirk-tmpl-2	eu-west-1-c	10.132.0.34 (nic0)	104.155.101.149	SSH	⌵
<input checked="" type="checkbox"/>	mountkirk-eu-v1-z85z	29.12.2019, 19:03:47	mountkirk-tmpl-1	eu-west-1-b	10.132.0.32 (nic0)	34.76.68.3	SSH	⌵

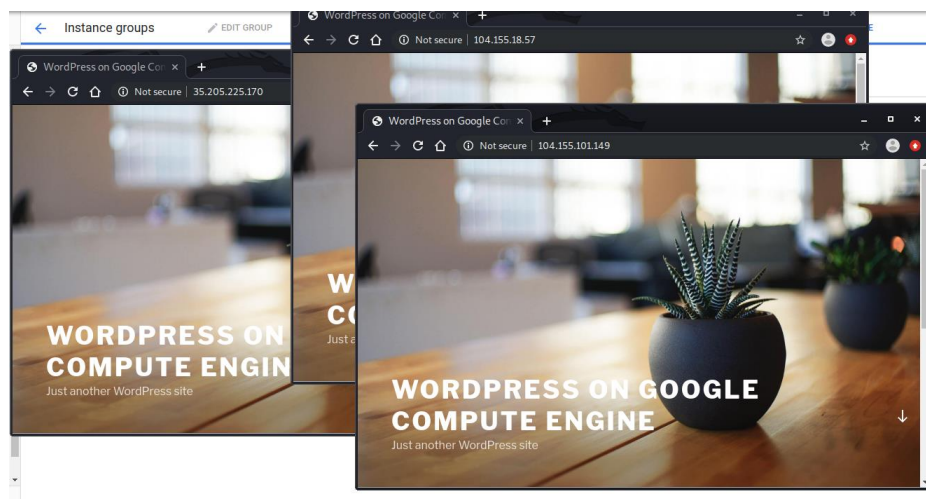
- f) Testowanie wersji.



- g) Aktualizacja pozostałych maszyn. Utworzone zostają dwie nowe a następnie usunięte dwie ze starym oprogramowaniem.

```
gcloud compute instance-groups managed rolling-action start-update mountkirk-mig-eu-west1 \
  --version template=mountkirk-tmpl-2 \
  --max-surge=3 --max-unavailable=0 --region europe-west1
```

<input checked="" type="checkbox"/>	mountkirk-eu-v1-0bcl	eu-west-1-d		mountkirk-mig-eu-west1	10.132.0.33 (nic0)	35.187.88.114	SSH	⌵	⋮
<input checked="" type="checkbox"/>	mountkirk-eu-v1-k1hf	eu-west-1-b		mountkirk-mig-eu-west1	10.132.0.35 (nic0)	35.205.225.170	SSH	⌵	⋮
<input checked="" type="checkbox"/>	mountkirk-eu-v1-rs68	eu-west-1-d		mountkirk-mig-eu-west1	10.132.0.36 (nic0)	104.155.18.57	SSH	⌵	⋮
<input checked="" type="checkbox"/>	mountkirk-eu-v1-vpfj	eu-west-1-c		mountkirk-mig-eu-west1	10.132.0.34 (nic0)	104.155.101.149	SSH	⌵	⋮
<input checked="" type="checkbox"/>	mountkirk-eu-v1-z85z	eu-west-1-b			10.132.0.32 (nic0)	34.76.68.3	SSH	⌵	⋮
<input checked="" type="checkbox"/>	mountkirk-eu-v1-k1hf	29.12.2019, 19:34:25	mountkirk-tmpl-2	eu-west-1-b	10.132.0.35 (nic0)	35.205.225.170	SSH	⌵	
<input checked="" type="checkbox"/>	mountkirk-eu-v1-rs68	29.12.2019, 19:34:26	mountkirk-tmpl-2	eu-west-1-d	10.132.0.36 (nic0)	104.155.18.57	SSH	⌵	
<input checked="" type="checkbox"/>	mountkirk-eu-v1-vpfj	29.12.2019, 19:24:36	mountkirk-tmpl-2	eu-west-1-c	10.132.0.34 (nic0)	104.155.101.149	SSH	⌵	



7. Harmonogram prac.

8. Szkolenia.

9. Testy odbiorowe.