Bezpieczeństwo rozwiązań chmurowych

Dokumentacja wdrożenia aplikacji WWW

„Event Planner” (AWS, zakres do oceny 3.5)

Autorzy:

Grzegorz Latawiec 169576
Igor Guła 169784
Wiktoria Bzduń 169766

4 stycznia 2026

Spis treści

1 Cel i zakres

Celem pracy było przygotowanie środowiska chmurowego umożliwiającego uruchomienie prostej aplikacji WWW do planowania wydarzeń oraz wdrożenie podstawowych mechanizmów bezpieczeństwa. Zakres obejmuje:

- zarządzanie dostępem (konto nieuprzywilejowane, ograniczenie użycia konta root),
- ograniczenie ekspozycji usług poprzez reguły sieciowe,
- dokumentację wykonanych czynności wraz z dowodami w postaci zrzutów ekranu,
- segmentację sieci (podział na podsieć publiczną i prywatną) oraz pośrednie warstwy dostępu (Bastion Host, Application Load Balancer),
- podstawowy monitoring (CPU, ruch sieciowy) w CloudWatch.

2 Założenia środowiska

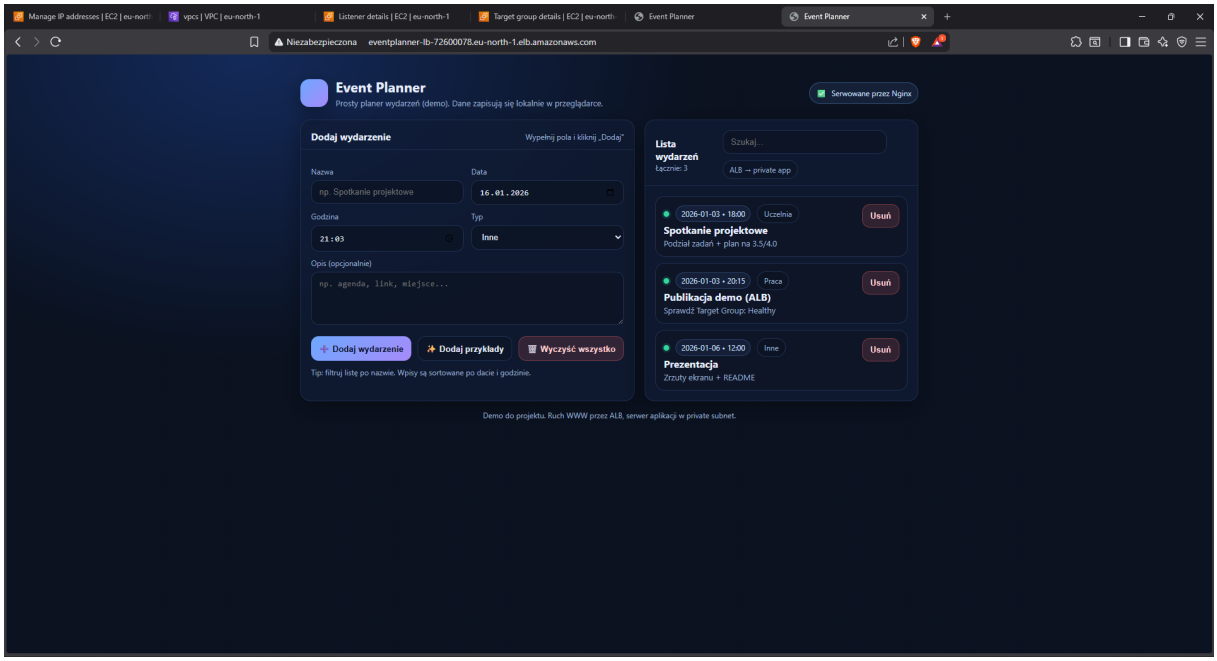
Wdrożenie zrealizowano w usługach Amazon Web Services. Aplikacja ma formę statycznej strony WWW serwowanej przez Nginx na instancji EC2. Dostęp publiczny do aplikacji realizowany jest przez Application Load Balancer, natomiast dostęp administracyjny (SSH) wyłącznie przez Bastion Host. Zastosowano zasadę minimalnych uprawnień w IAM oraz ograniczono otwarte porty do niezbędnych.

3 Opis architektury rozwiązania

Architektura została oparta o sieć VPC z podsiecią publiczną oraz prywatną. W podsieci prywatnej umieszczono serwer aplikacyjny EC2 bez publicznego adresu IP. W podsieci publicznej umieszczono Bastion Host (dla SSH) oraz Application Load Balancer (dla HTTP). Dostęp wychodzący z podsieci prywatnej do internetu został zapewniony poprzez NAT Gateway, co umożliwia wykonywanie połączeń wychodzących bez wystawiania instancji na ruch przychodzący [3].

3.1 Przepływy ruchu

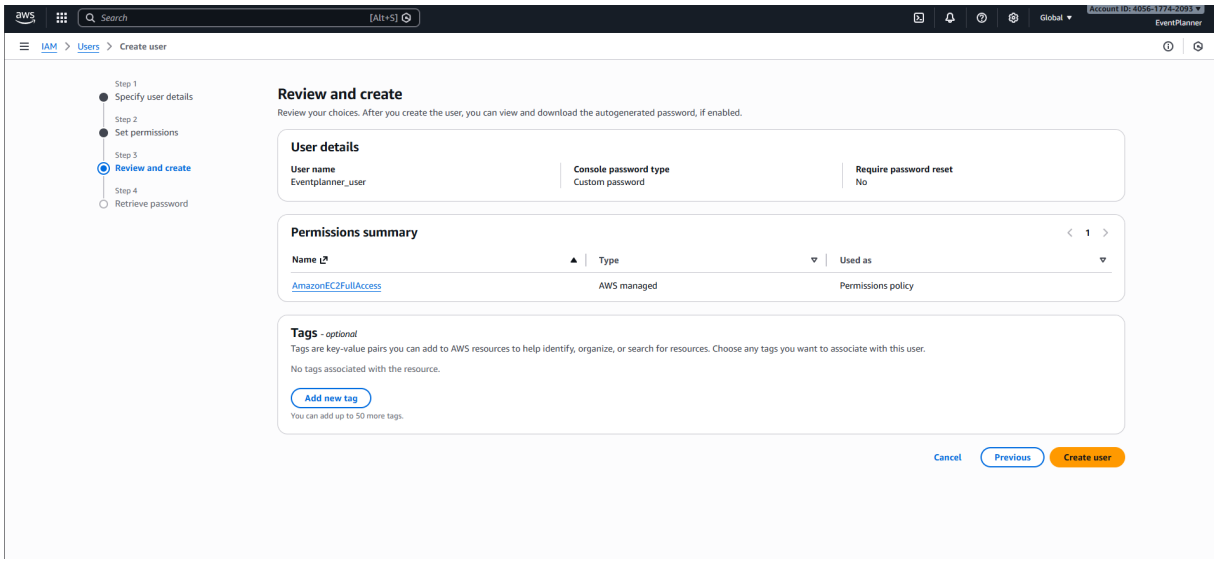
- **Ruch użytkowników (HTTP):** Internet → ALB → serwer aplikacyjny (EC2 w podsieci prywatnej).
- **Zarządzanie (SSH):** stacja robocza → Bastion Host (podsieć publiczna) → serwer aplikacyjny (podsieć prywatna).
- **Ruch wychodzący z podsieci prywatnej:** EC2 → NAT Gateway → Internet (np. aktualizacje systemu).



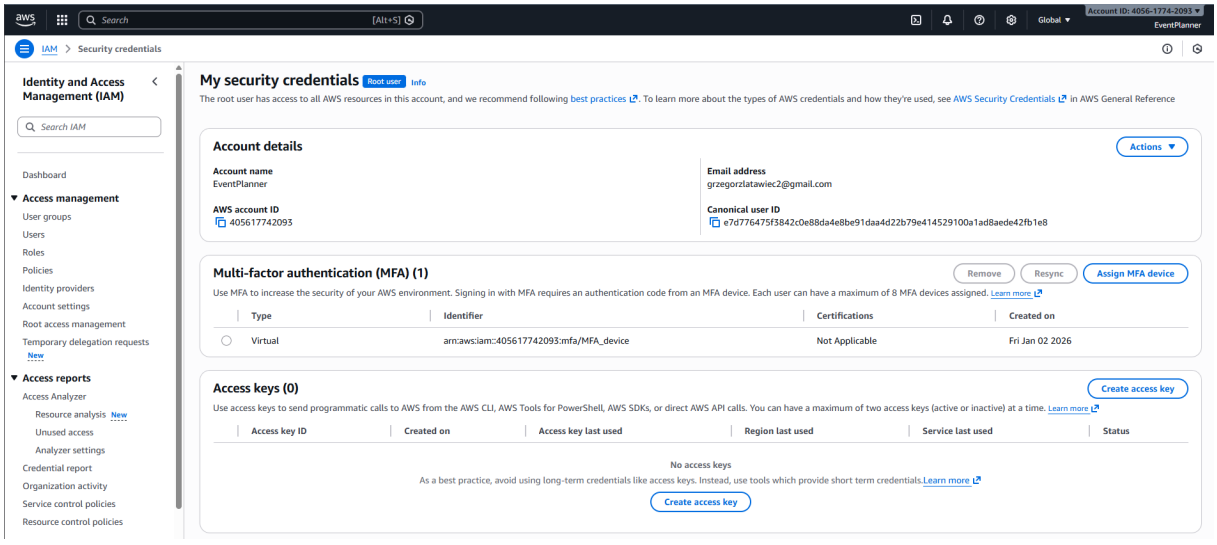
Rysunek 1: Dostęp do aplikacji „Event Planner" przez publiczny adres DNS Application Load Balancer.

4 Zarządzanie dostępem (IAM)

W ramach etapu bazowego utworzono dedykowanego użytkownika IAM wykorzystywanego do prac konfiguracyjnych, zgodnie z zasadą *least privilege* [1]. Ograniczono stosowanie konta root do czynności administracyjnych wymagających jego użycia, a dodatkowo włączono mechanizm uwierzytelniania wieloskładnikowego (MFA), co stanowi zalecaną praktykę dla zabezpieczenia konta [2].



Rysunek 2: Utworzenie dedykowanego użytkownika IAM do prac operacyjnych.



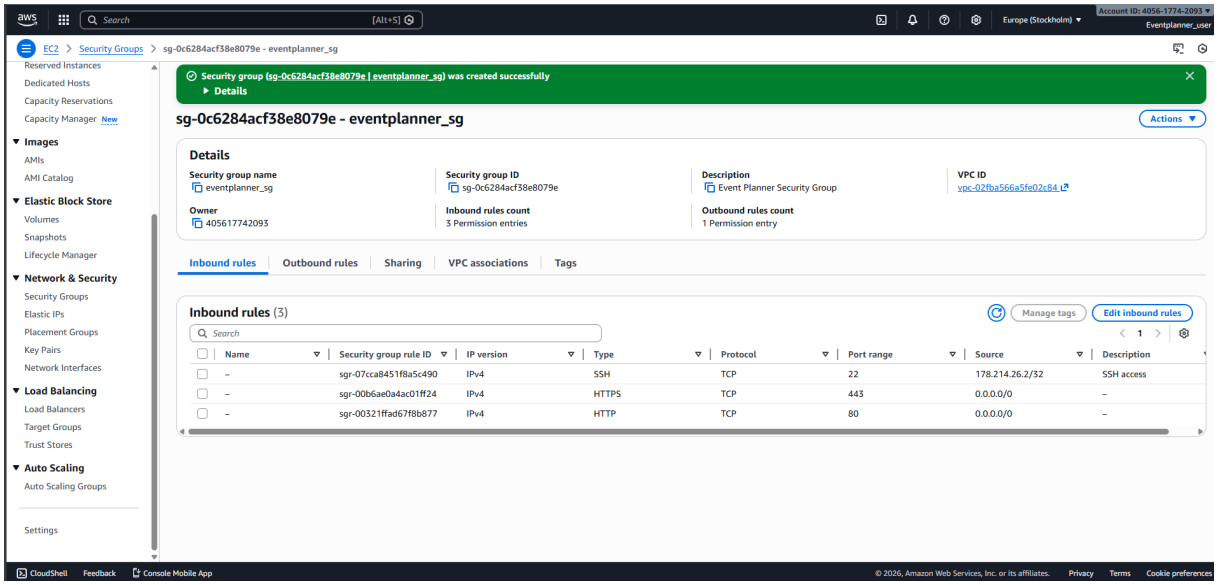
Rysunek 3: Konfiguracja MFA dla użytkownika root (widoczne potwierdzenie przypisanego urządzenia MFA).

5 Bezpieczeństwo sieci (Security Groups oraz host firewall)

Konfiguracja bezpieczeństwa sieciowego została zrealizowana w oparciu o *Security Groups*, które stanowią stanowe listy kontroli dostępu na poziomie instancji. Otworzono wyłącznie porty wymagane do działania systemu:

- port 80/tcp dla HTTP (dostęp do aplikacji przez ALB),
- port 22/tcp dla SSH (dostęp administracyjny ograniczony do źródła).

Dodatkowo na poziomie systemu operacyjnego wykorzystano `ufw` w celu wzmocnienia ochrony hosta (zasada obrony w głąb, *defense in depth*). Przykładowy stan reguł `ufw` przedstawiono na rys. 5.



Rysunek 4: Reguły przychodzące w Security Group (SSH ograniczony do wskazanego adresu IP; ruch WWW na portach 80/443).

```
ubuntu@ip-172-31-47-160:~$ sudo ufw status verbose
Status: active
Logging: on (low)
Default: deny (incoming), allow (outgoing), disabled (routed)
New profiles: skip

To                         Action      From
--                         ------      ----
22/tcp (OpenSSH)           ALLOW IN    Anywhere
80,443/tcp (Nginx Full)    ALLOW IN    Anywhere
22/tcp (OpenSSH (v6))      ALLOW IN    Anywhere (v6)
80,443/tcp (Nginx Full (v6)) ALLOW IN    Anywhere (v6)

ubuntu@ip-172-31-47-160:~$
```

Rysunek 5: Status zapory `ufw` na hoście (dodatkowa warstwa kontroli ruchu).

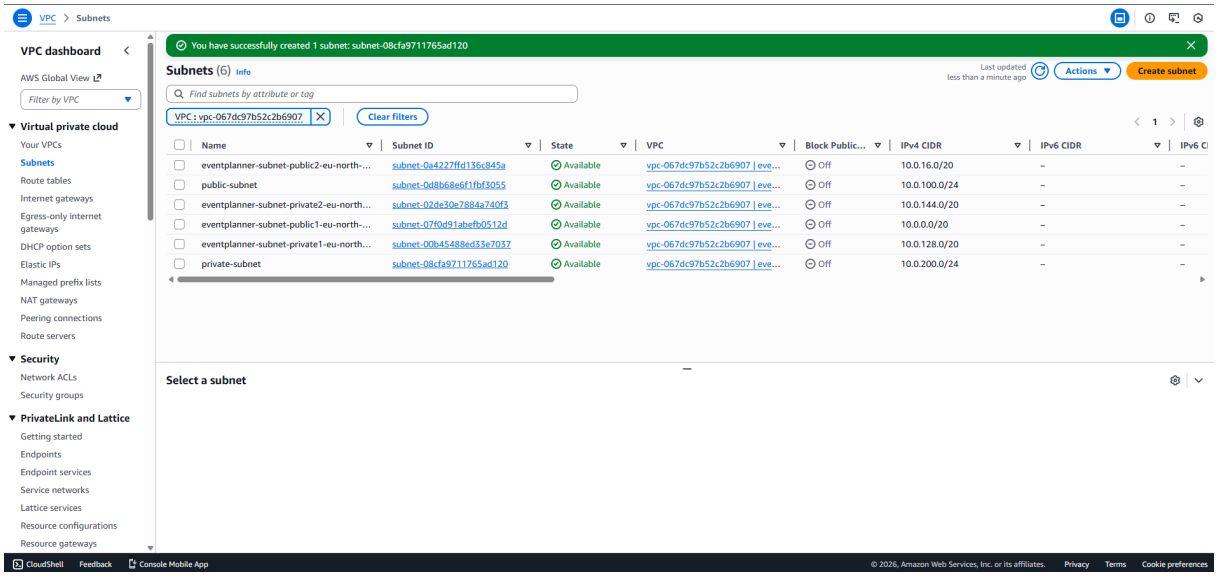
6 Segmentacja sieci (etap 3.5)

W etapie 3.5 wykonano segmentację sieci poprzez umieszczenie serwera aplikacyjnego w podsieci prywatnej bez bezpośredniego dostępu przychodzącego z internetu. Podejście to ogranicza powierzchnię ataku i jest zgodne z zaleceniami projektowania sieci w chmurze.

6.1 Podsieci (Subnets)

W obrębie VPC utworzono podsieci:

- **Public subnet(y)** – przeznaczone dla zasobów wymagających ekspozycji (ALB, Bastion Host),
- **Private subnet** – przeznaczony dla serwera aplikacyjnego EC2 bez publicznego IP.

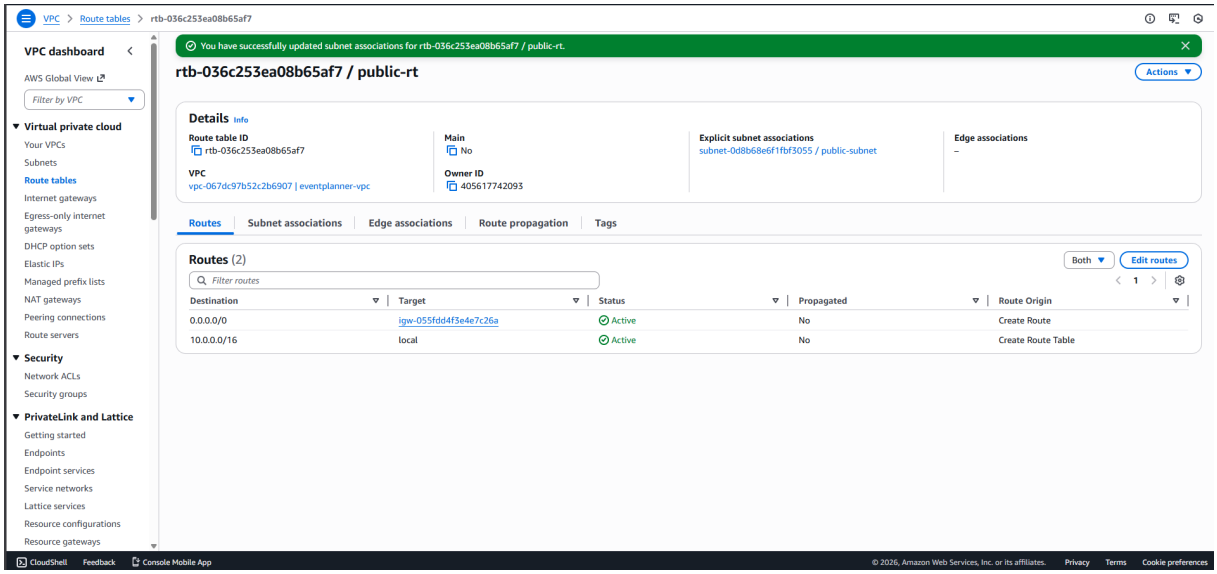


Rysunek 6: Lista podsieci w VPC, w tym podsieć prywatna dla serwera aplikacyjnego oraz podsieć publiczna dla warstwy dostępowej.

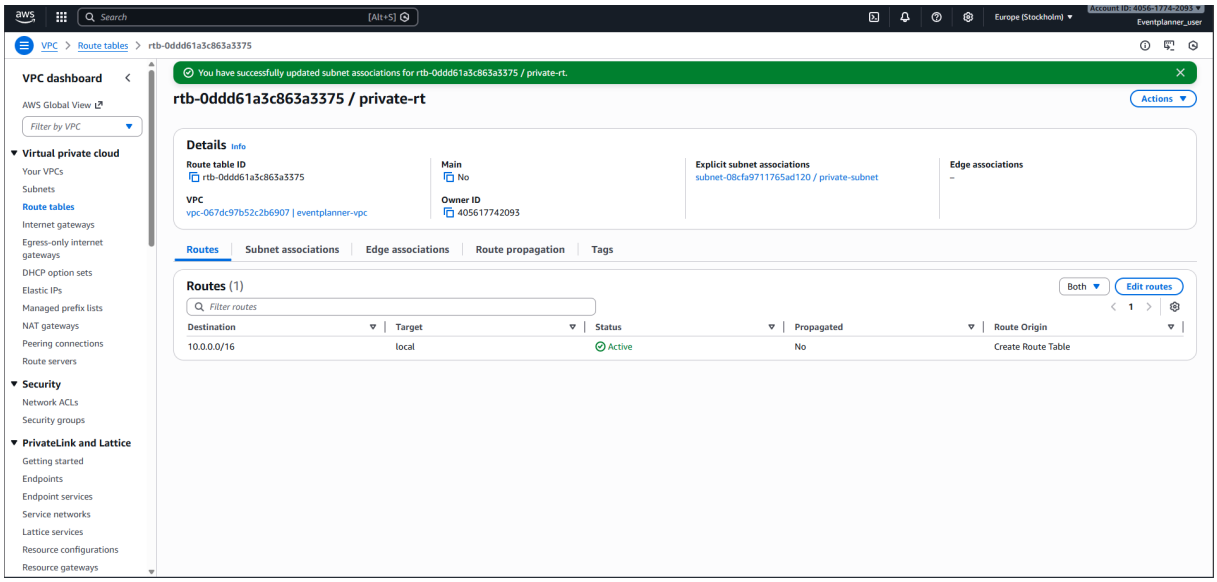
6.2 Routing: Internet Gateway oraz NAT Gateway

Tablice routingu skonfigurowano w sposób rozdzielający ruch publiczny i prywatny:

- tablica routingu dla podsieci publicznej zawiera trasę domyślną do Internet Gateway [4],
- tablica routingu dla podsieci prywatnej zapewnia ruch wewnątrz VPC; dostęp do internetu dla połączeń wychodzących realizowany jest poprzez NAT Gateway [3].



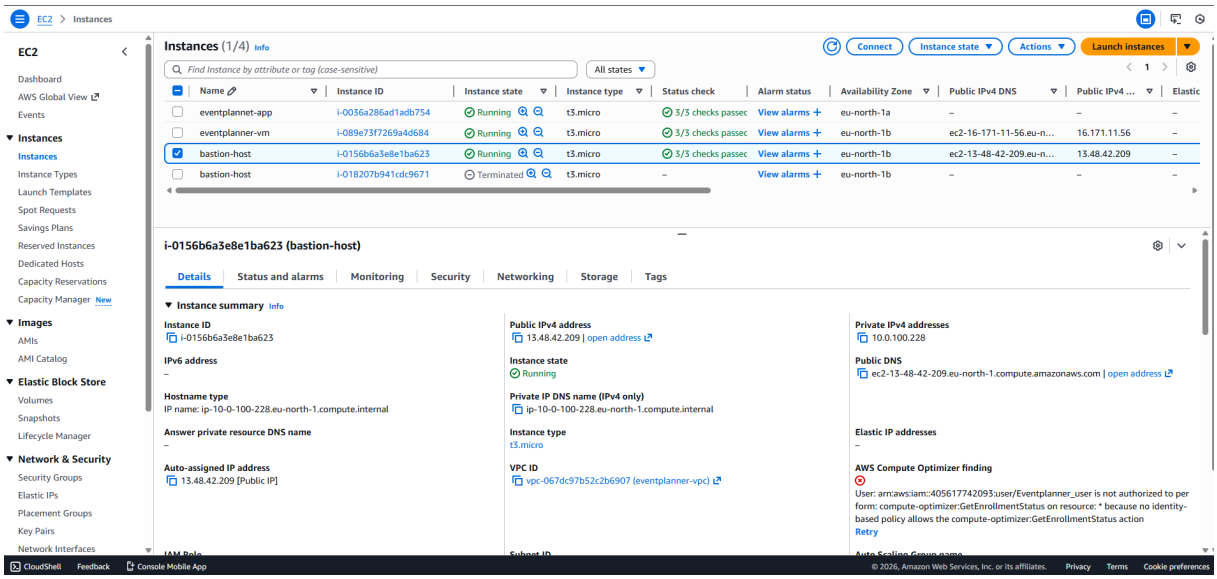
Rysunek 7: Tablica routingu dla podsieci publicznej (trasa 0.0.0.0/0 do Internet Gateway).



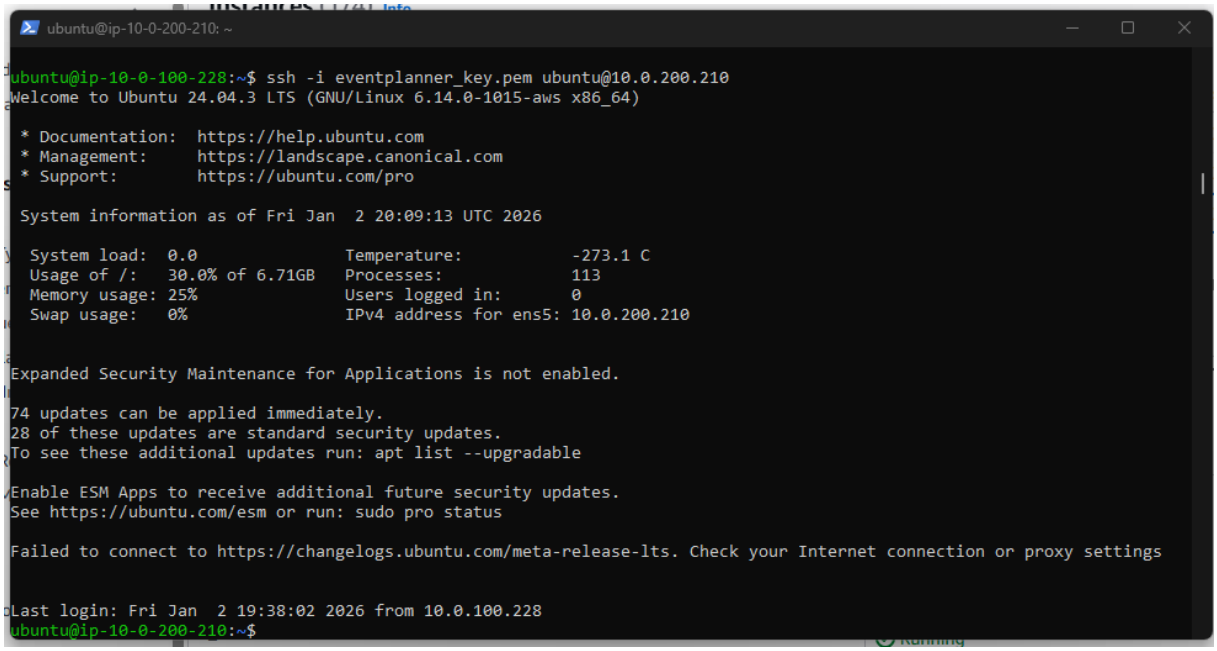
Rysunek 8: Tablica routingu dla podsieci prywatnej (asocjacja z private subnet).

7 Warstwa dostępu: Bastion Host

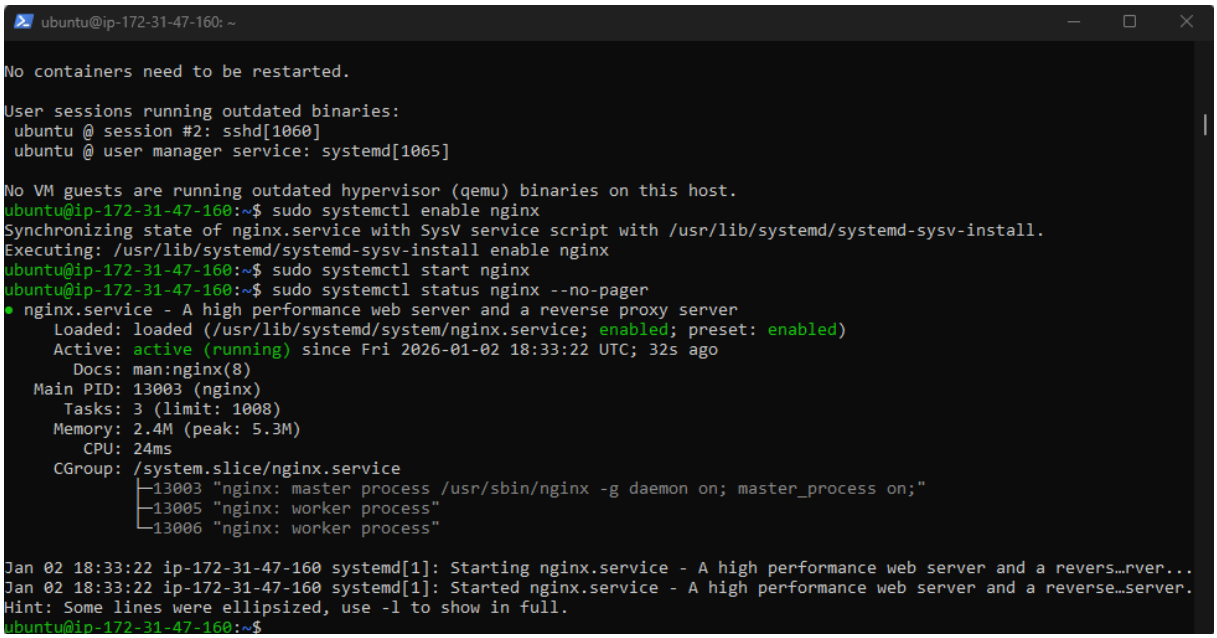
Zarządzanie serwerem aplikacyjnym zrealizowano poprzez Bastion Host umieszczony w podsieci publicznej. Połączenie SSH do serwera aplikacyjnego jest nawiązywane dopiero z Bastiona, co eliminuje konieczność wystawiania portu 22 w podsieci prywatnej na internet.



Rysunek 9: Szczegóły instancji Bastion Host (publiczny adres IPv4).



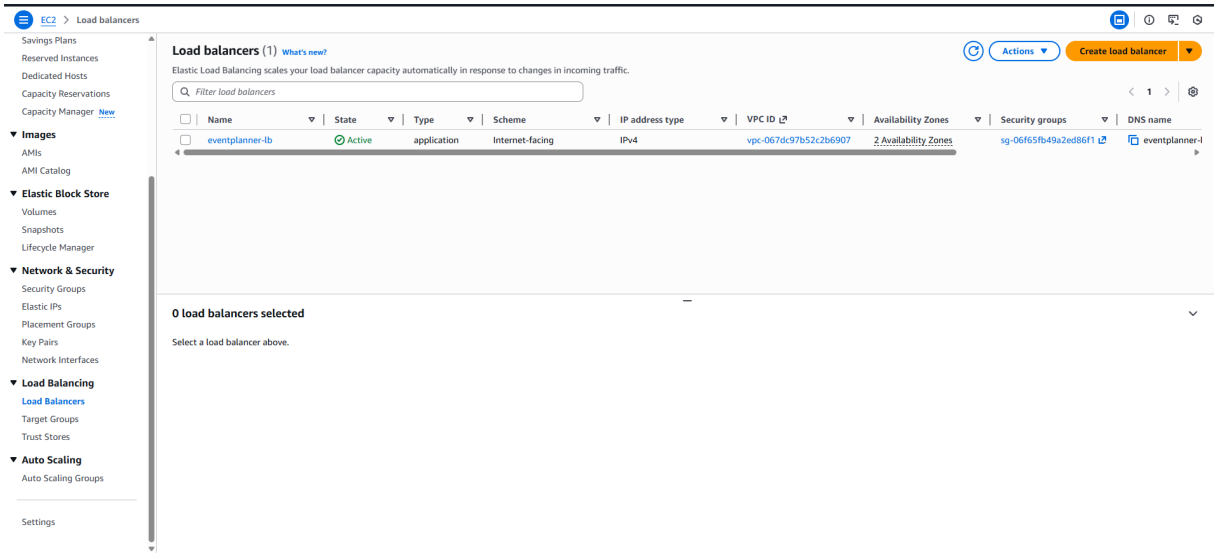
Rysunek 10: Połączenie SSH z Bastion Host do serwera aplikacyjnego w podsieci prywatnej.



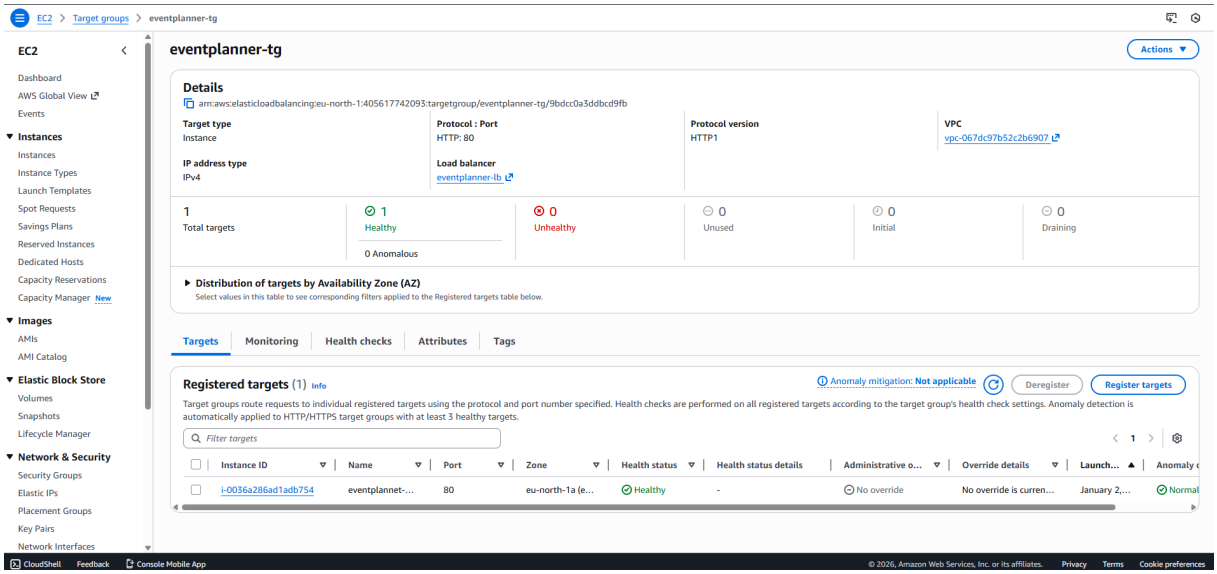
Rysunek 11: Potwierdzenie poprawnego zalogowania do instancji (sesja SSH).

8 Warstwa dostępu: Application Load Balancer

Dostęp użytkowników końcowych do aplikacji realizowany jest przez Application Load Balancer (ALB) w trybie *internet-facing*. ALB przekazuje ruch HTTP do zarejestrowanego Target Group, która wskazuje instancję aplikacyjną jako cel ruchu. Mechanizm zdrowia Target Group umożliwia weryfikację dostępności backendu; status *Healthy* oznacza, że instancja pomyślnie przechodzi testy [6, 5].



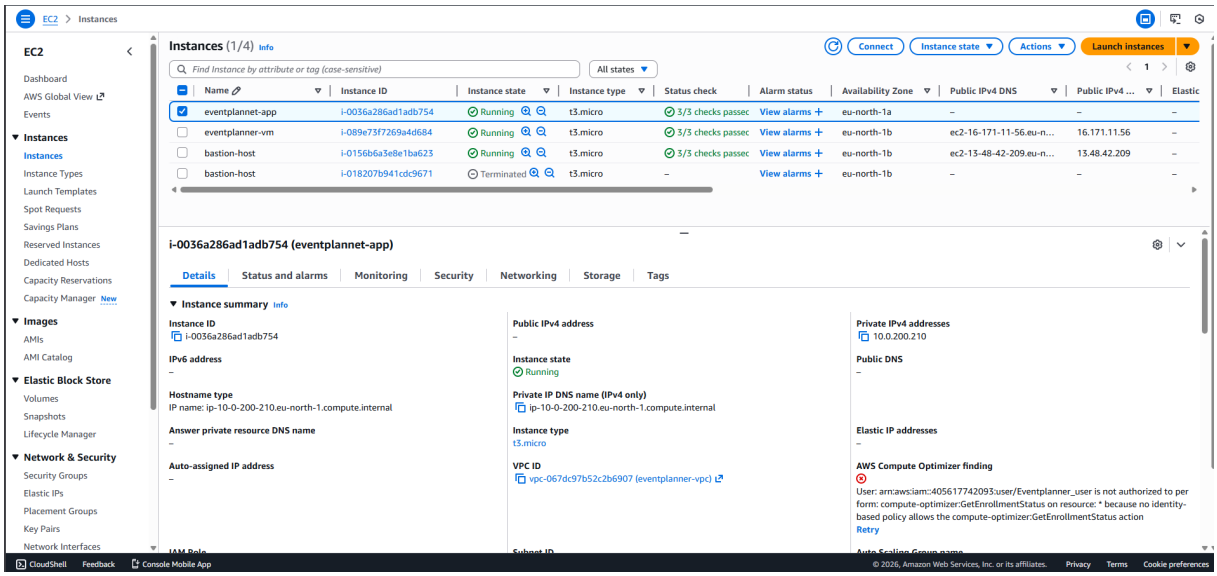
Rysunek 12: Application Load Balancer w stanie *Active*.



Rysunek 13: Target Group z instancją backendową w stanie *Healthy*.

8.1 Izolacja serwera aplikacyjnego

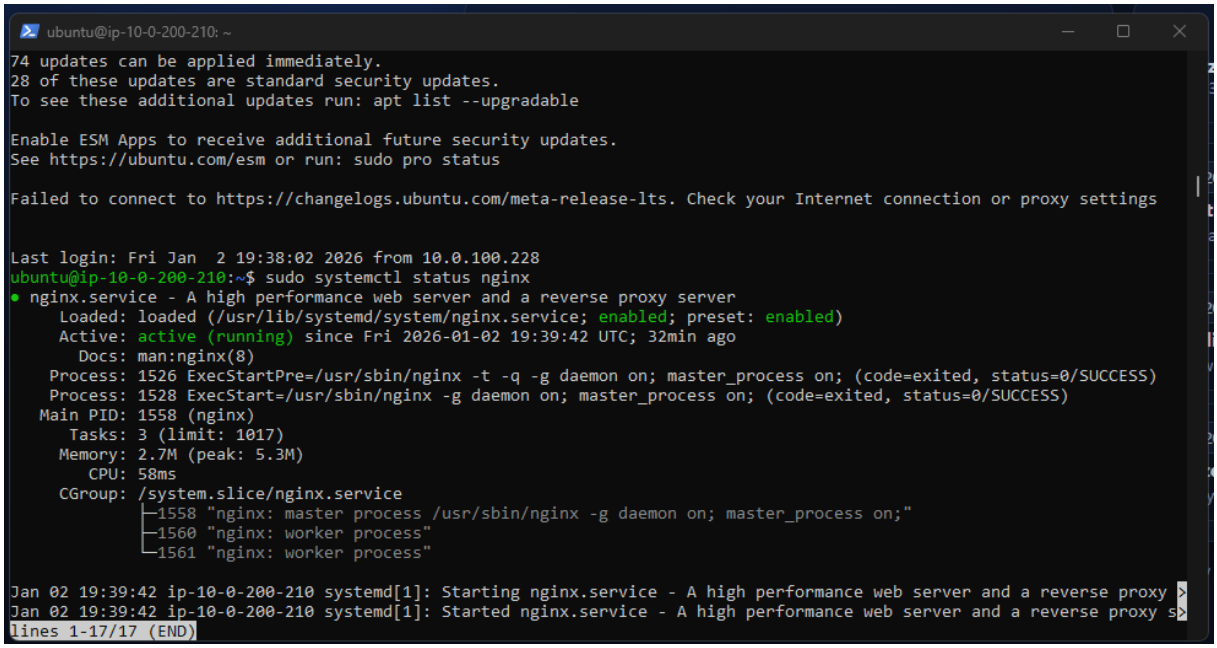
Serwer aplikacyjny nie posiada publicznego adresu IPv4, a ruch HTTP jest dopuszczany wyłącznie z warstwy ALB (co wynika z zastosowania reguł Security Group i separacji podsieci). Potwierdzenie braku publicznego adresu na instancji aplikacyjnej przedstawiono na rys. 14.



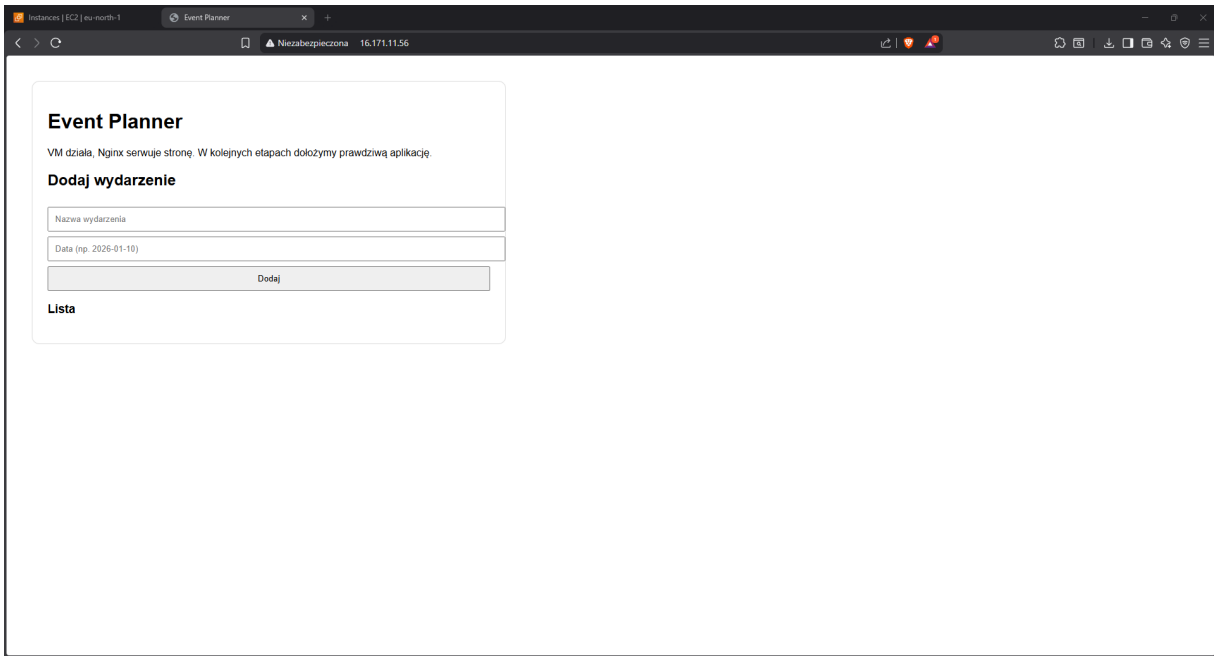
Rysunek 14: Instancja aplikacyjna w podsieci prywatnej (brak publicznego IPv4).

9 Konfiguracja usługi WWW na serwerze aplikacyjnym (Nginx)

Na instancji aplikacyjnej uruchomiono serwer WWW Nginx, który dostarcza stronę „Event Planner”. Z punktu widzenia wymagań projektu istotne było wykazanie, że usługa działa oraz jest uruchomiona jako usługa systemowa. Stan usługi przedstawiono na rys. 15. Dodatkowo zweryfikowano dostępność aplikacji przez przeglądarkę (rys. 1 oraz rys. 16).



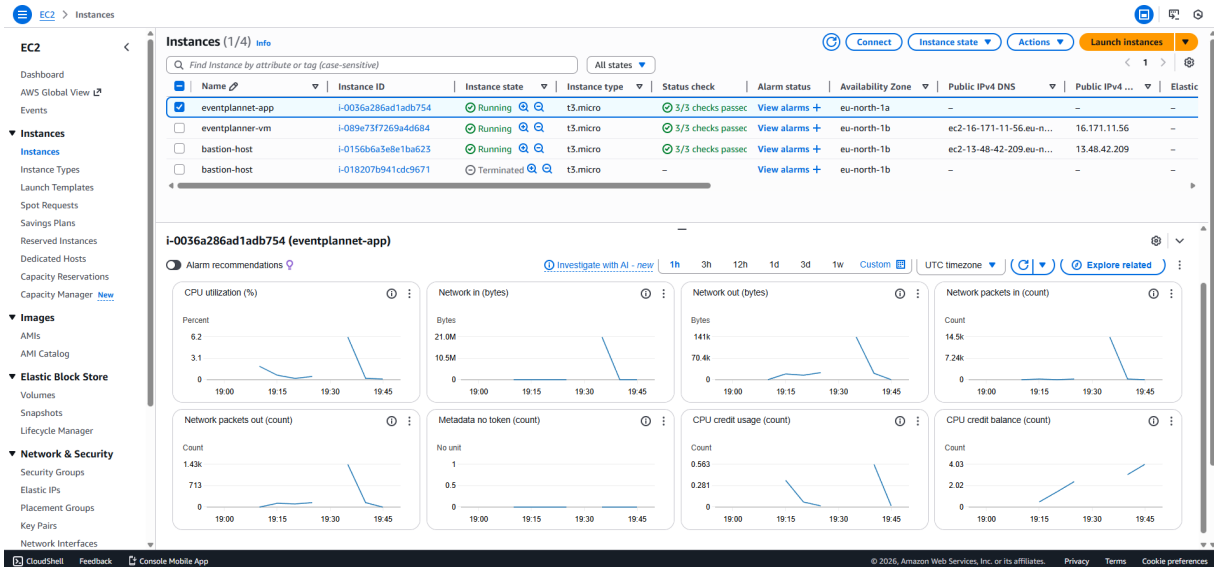
Rysunek 15: Stan usługi Nginx na serwerze (aktywna jednostka systemd).



Rysunek 16: Weryfikacja działania strony serwowanej przez Nginx (widok w przeglądarce).

10 Monitoring (Amazon CloudWatch)

Włączono podstawowy monitoring zasobów w Amazon CloudWatch. Monitorowane są metryki instancji EC2, w szczególności `CPUUtilization` oraz metryki ruchu sieciowego (`NetworkIn`, `NetworkOut`) [7]. Przykładowy widok metryk w konsoli przedstawiono na rys. 17.



Rysunek 17: Podstawowe metryki instancji EC2 w CloudWatch (CPU oraz ruch sieciowy).

11 Wnioski

Zrealizowane wdrożenie spełnia wymagania do oceny 3.5 poprzez zastosowanie:

- dedykowanego użytkownika IAM i ograniczenie użycia konta root (MFA),
- minimalizacji ekspozycji usług poprzez reguły Security Group oraz dodatkowo `ufw`,
- segmentacji sieci (public/private subnet),
- pośredniej warstwy dostępu do aplikacji przez ALB oraz do administracji przez Bastion Host,
- podstawowego monitoringu zasobów w CloudWatch.

Przyjęta architektura ogranicza bezpośredni dostęp do zasobów w podsieci prywatnej, utrzymując jednocześnie funkcjonalność systemu. W praktyce stanowi to typowy wzorzec wdrożeniowy dla prostych usług WWW w środowisku chmurowym.

Literatura

[1] AWS, *Security best practices in IAM*. Dostęp: https://docs.aws.amazon.com/IAM/latest/UserGuide/best-practices.html.

[2] AWS, *Root user best practices for your AWS account*. Dostęp: https://docs.aws.amazon.com/IAM/latest/UserGuide/root-user-best-practices.html.

[3] AWS, *NAT gateways – Amazon VPC User Guide*. Dostęp: https://docs.aws.amazon.com/vpc/latest/userguide/vpc-nat-gateway.html.

[4] AWS, *Internet gateways – Amazon VPC User Guide*. Dostęp: https://docs.aws.amazon.com/vpc/latest/userguide/VPC_Internet_Gateway.html.

[5] AWS, *Target groups for your Application Load Balancers*. Dostęp: https://docs.aws.amazon.com/elasticloadbalancing/latest/application/load-balancer-target-groups.html.

[6] AWS, *Health checks for Application Load Balancer target groups*. Dostęp: https://docs.aws.amazon.com/elasticloadbalancing/latest/application/target-group-health-checks.html.

[7] AWS, *CloudWatch metrics for Amazon EC2 instances*. Dostęp: https://docs.aws.amazon.com/AWSEC2/latest/UserGuide/viewing_metrics_with_cloudwatch.html.