

## Projekt 1 (5 pkt)

1. Stwórz aplikację webową w wybranej przez siebie technologii, tematyka aplikacji jest dowolna (może to być np. aplikacja umożliwiająca prosty chat grupowy) aplikacja powinna:
  - Posiadać przynajmniej 2 endpointy GET oraz 2 endpointy POST oraz dodatkowy jeden POST i GET do przesyłania/odbierania plików multimedialnych)
2. Aplikacja powinna posiadać oddzielne dwa moduły backend oraz frontend. Oba moduły muszą być hostowane niezależnie.
3. Utwórz konfigurację Dockera dla każdego z modułów
4. Skonfiguruj dostęp do AWS CLI  
Uwaga: dane autoryzacyjne znajdują się w zakładce AWS Details i mogą zmieniać się przy każdym starcie Learner Lab



5. Zainstaluj i skonfiguruj Terraform, a następnie użyj go do utworzenia całej infrastruktury potrzebnej dla aplikacji.
6. Uwierzytelnianie użytkowników zrealizuj przy użyciu AWS Cognito.
7. Do przechowywania plików multimedialnych wykorzystaj AWS S3.
8. Dane aplikacji zapisz w AWS RDS (np. MySQL, PostgreSQL) lub DynamoDB – wybór zależy od rodzaju aplikacji.
9. Uruchom aplikację z wykorzystaniem AWS Fargate (backend i frontend na osobnych instancjach) oraz zweryfikuj jej poprawne funkcjonowanie
10. Skonfiguruj CloudWatch dla swojej aplikacji
11. Dodaj do aplikacji funkcjonalność wykorzystującą AWS Lambda (np. automatyczne przetwarzanie przesyłanych plików, powiadomienia itp.).
12. Cała konfiguracja infrastruktury powinna być możliwa do odtworzenia przy użyciu Terraform.
13. Następnie, ręcznie (poprzez interfejs AWS Management Console) utwórz odpowiednik infrastruktury zdefiniowanej w Terraform i uruchom na niej aplikację – w celu porównania obu podejść.
14. Stwórz raport podsumowujący wykonane zadanie

<https://docs.docker.com/>

[https://docs.docker.com/develop/develop-images/dockerfile\\_best-practices/](https://docs.docker.com/develop/develop-images/dockerfile_best-practices/)

<https://docs.docker.com/develop/develop-images/guidelines/>

<https://docs.docker.com/develop/develop-images/instructions/>

<https://developer.hashicorp.com/terraform/tutorials/aws-get-started/install-cli>

<https://developer.hashicorp.com/terraform/docs>

<https://developer.hashicorp.com/terraform/intro>

<https://developer.hashicorp.com/terraform/language/files>

<https://developer.hashicorp.com/terraform/language/syntax/configuration>

<https://mahira-technology.medium.com/from-zero-to-hero-creating-secure-cognito-identity-pools-using-terraform-without-prior-experience-674a3933dab2>

[https://builder.aws.com/content/32kr9Onkcy9Wrflf46n4QRidDAp/mastering-aws-fargate-beginners-hands-on-tutorial-for-serverless-containers?utm\\_source=chatgpt.com](https://builder.aws.com/content/32kr9Onkcy9Wrflf46n4QRidDAp/mastering-aws-fargate-beginners-hands-on-tutorial-for-serverless-containers?utm_source=chatgpt.com)

<https://aws.plainenglish.io/simple-guide-to-deploy-docker-container-in-ecs-fargate-behind-alb-4fc6b5c12f8b>

[https://github.com/rafalpalak/cloud\\_backend](https://github.com/rafalpalak/cloud_backend)

[https://github.com/rafalpalak/cloud\\_frontend](https://github.com/rafalpalak/cloud_frontend)