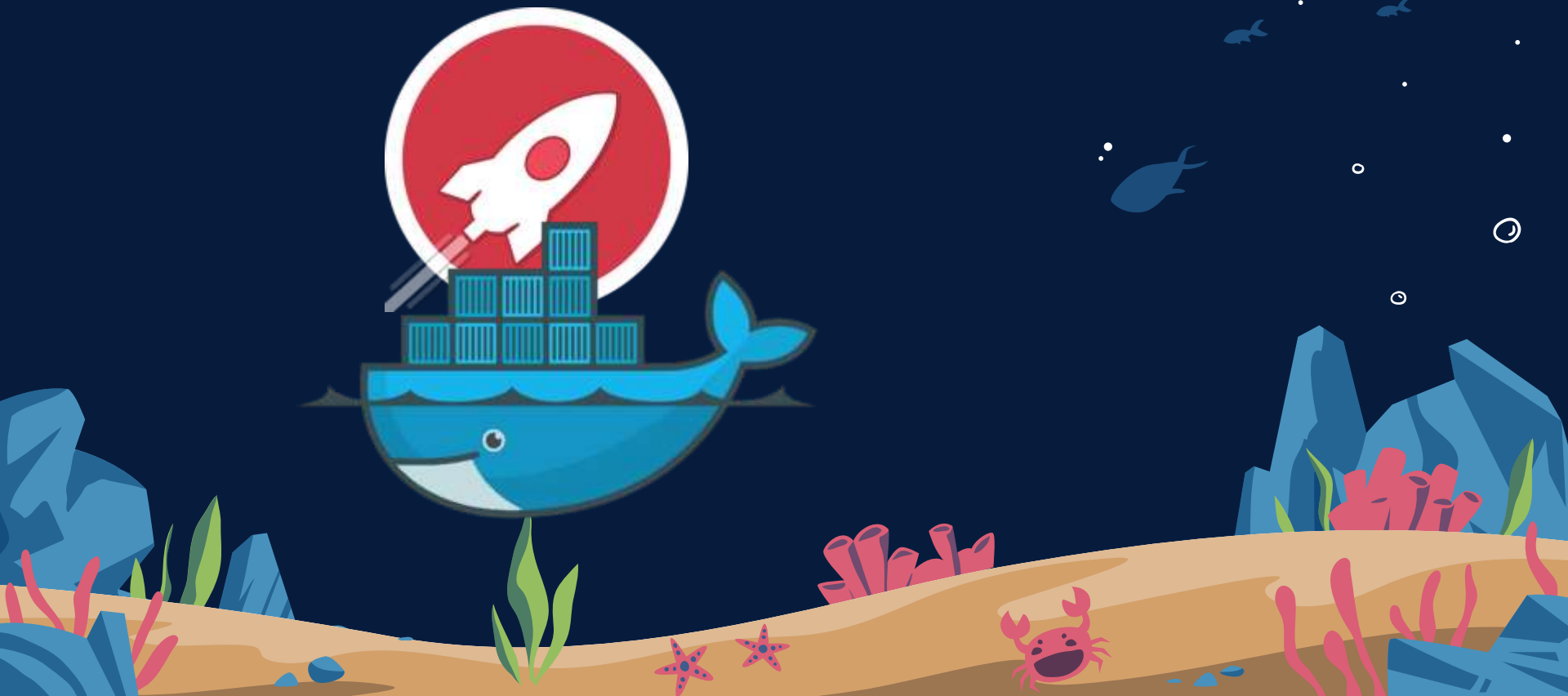


Rest Api With Rocket & Docker!

Let's Gooo.....



Create File a Main



Make sure you have set up your rust application with the Rocket framework. Here is the src/main.rs file:

```
main.rs

#[macro_use] extern crate rocket;

use rocket::serde::json::Json;
use serde::Serialize;

#[derive(Serialize)]
struct Message {
    message: String,
}

#[get("/")]
fn index() -> Json<Message> {
    Json(Message {
        message: "Hello, Bayu Widia Santoso!".to_string(),
    })
}

#[launch]
fn rocket() -> _ {
    rocket::build().mount("/", routes![index])
}
```

Create a Dockerfile

Create a dockerfile in the same directory as your app folder :



```

Dockerfile

# Use the official Rust image as the base image
FROM rust:1.74 AS builder

# Set the working directory inside the container
WORKDIR /app

# Copy the Cargo.toml and Cargo.lock files
COPY Cargo.toml Cargo.lock ./

# Copy the source code
COPY src ./src

# Build the application in release mode
RUN cargo build --release

# Use a smaller, final image for deployment
FROM ubuntu:22.04

# Set environment variable for Rocket to use the correct IP address
ENV ROCKET_ADDRESS=0.0.0.0
# ENV ROCKET_PORT=8000

# Copy the built binary from the builder stage
COPY --from=builder /app/target/release/rust-rocket-docker-hello /usr/local/bin/

# Expose the port Rocket will run on
EXPOSE 8000

# Set the default command to run the application
CMD ["rust-rocket-docker-hello"]
```

Create a docker-compose



Create a docker-compose in the same directory as your app folder :

```
docker-compose

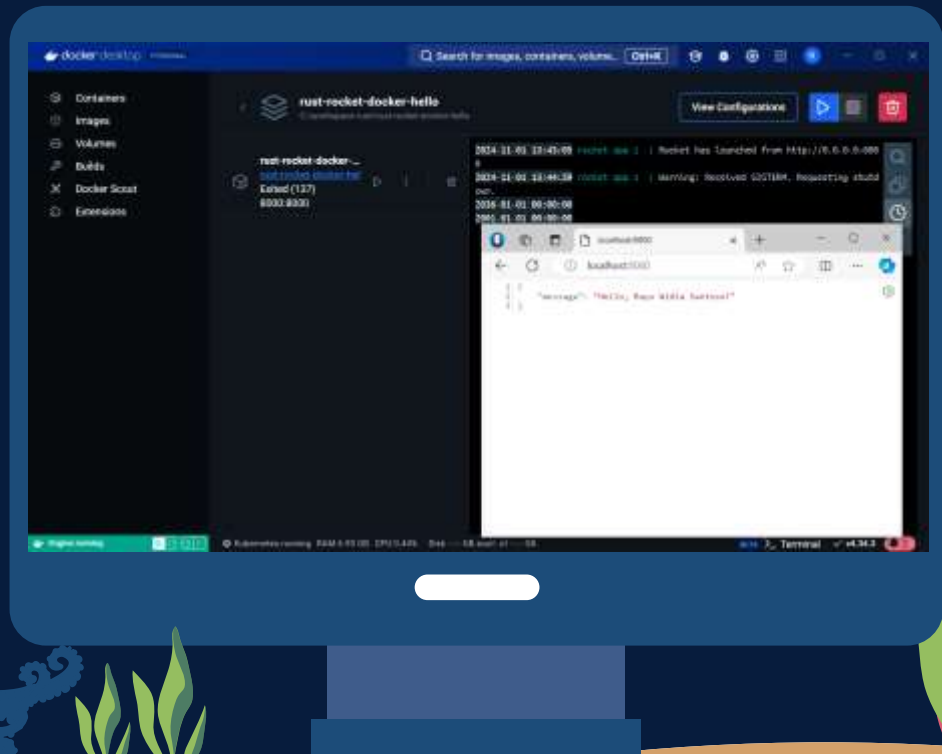
version: '3.8'

services:
  rocket-app:
    build: .
    ports:
      - "8000:8000"
    environment:
      - ROCKET_ADDRESS=0.0.0.0
      # - ROCKET_PORT=8000
```

Docker Desktop



- *# Rebuild the Docker image*
- `docker-compose down`
- `docker-compose build --no-cache`
- `docker-compose up`





SEE YOU...

