Title: docker-compose up gives 'failed to read dockerfile: error from sender'

Post Body:

Tried to simply dockerize my mongodb and spring boot application. Had a lot of struggles and thought I almost had it running and than the terminal hits me with this error:

```
Building user [+] Building 0.0s (1/2)  => ERROR [internal] load build definition from Dockerfile
```

Dockerfile (which is in a sub-directory; user-module):

```
FROM openjdk:11 EXPOSE 8081 ADD target/user-module-docker.jar user-docker.jar CMD ["java", "-jar", "user-docker.jar"]
```

docker-compose.yml (which is in the main directory that has multiple modules/microservices):

```
version: '3.8'  services:   user:     build: ./user/Dockerfile     restart: unless-stopped     container_name: user-ms     por
```

Like it says that the path specified cannot be found but it literally exists, so where could I have gone wrong?

Accepted Answer:

The error is telling you that the Dockerfile was not found, because the path doesn't exist. That's because it is trying to enter the path as folder.

> The system cannot find the path specified.

This comes because you made a mistake in the compose build syntax. There are 2 ways it can be used.

1. The simple form:

This is using ./users/ as context, expecting a Dockerfile to be in this directory.

```
user:   build: ./user
```

2. The complex form:

```
user:   build:     context: ./     dockerfile: ./users/Dockerfile
```

This lets you separate the context and where the Dockerfile is. In this example, the current folder is used as context, and the Dockerfile is taken from ./users/Dockerfile. It is also useful when you have a different name for your Dockerfile. I.E. Dockerfile.dev.

Note that this is just an example, I don't know if this would make sense in your project. You need to know what context is the correct one.

What do I mean by context?

> The docker build command builds Docker images from a Dockerfile and a “context”. A build’s context is the set of files located in the specified PATH or URL. The build process can refer to any of the files in the context. For example, your build can use a COPY instruction to reference a file in the context.

As example:

```
docker build --file /path/to/Dockerfile /path/to/context
```

Highest Rated Answer: None