Building and shipping Go applications with Docker

11-4-2016
GopherConBR, Florianópolis, SC, Brazil

Building

Go without installing go in a disposable container:

```
docker run --rm golang sh -c "go get github.com/golang/example/hello/... && exec hello"
```

Switching golang versions:

```
docker run --rm golang:1.5 sh -c "go get github.com/golang/example/hello/... && exec hello"
```

Install it in your system:

```
docker run -v $HOME/bin:/go/bin  golang go get github.com/golang/example/hello/… /tmp/bin/hello
```

Install it in your system (cross-compile):

```
docker run -e GOOS=darwin -e GOARCH=amd64 -v $HOME/bin:/go/bin  golang go get
github.com/golang/example/hello/… /tmp/bin/hello
```

Shipping

Static link FTW (disable CGO or use musl libc <3)

```
CC=/usr/local/musl/bin/musl-gcc go build --ldflags '-linkmode external -X main.version=123 -extldflags
"-static"’
```

Use lean images:

```
FROM scratch
COPY ./hello /hello
ENTRYPOINT ["/hello"]
```

Build & Ship

Use the docker-compose approach:

```
#builder
builder:
        image: golang
        volumes:
                - $GOPATH/src:/go/src
                - .:/go/bin
        command: go build --ldflags '-linkmode external -extldflags "-static"’


#prod
#Dockerfile from the previous slide
myapp:
        build: .
        ports:
                - “8000:8000”
```

Build & Ship

Use a multi dockerfile approach:

```
#Dockerfile
FROM golang

# Copy the runtime dockerfile into the context as Dockerfile
COPY Dockerfile.run /go/bin/Dockerfile

COPY . /go/src/github.com/golang/example/hello

WORKDIR /go/src/github.com/golang/example/hello

RUN go get -v -d ./...

RUN go build --ldflags '-linkmode external -extldflags
"-static"'  -o /go/bin/hello

# Set the workdir to be /go/bin which is where the binaries
are built
WORKDIR /go/bin

# Export the WORKDIR as a tar stream
CMD tar -cf - .
```

```
#Dockerfile.run

FROM scratch
ADD example /bin/example

WORKDIR /bin

EXPOSE 8080

CMD [“/bin/example”]
```

```
docker build -t builder . && docker run builder |
docker build -t hello -

docker push hello
```