Create a docker image from multiple methods likes Dockerfile, running containers.

Creating a Docker image can be done in several ways, including using a Dockerfile, committing a running container, Multi-stage Dockerfile and using Docker build commands with context.

Method 1: Using a Dockerfile

Creating a Dockerfile

```
# Use an official base image
FROM alpine:latest

# Install any needed packages
RUN apk update && apk add --no-cache python3

# Set the working directory
WORKDIR /app

# Copy the current directory contents into the container at /app
COPY . /app

# Make port 80 available to the world outside this container
EXPOSE 80

# Run app.py when the container launches
CMD ["python", "app.py"]
```

Building Image

```
$ docker build -t image_name  .
```

Method 2: From a Running Container

Start a base container and make changes **:**

```
# Start a container
docker run -it --name mycontainer ubuntu:22.04 bash

# Inside the container, make  changes
apt-get update
apt-get install -y nginx
```

```
# Configure nginx, add files, etc.
exit

# Commit the container to create an image
docker commit mycontainer mynginx:custom

# Optionally add a commit message and author
docker commit -m "Added nginx" -a "our Name" mycontainer mynginx:custom
```

In such a way, we can our Image by modifying running container and commit the changes in it.

Method 3: Multi-stage Dockerfile

```
# Build stage
FROM golang:1.20 AS builder

WORKDIR /app

COPY go.mod go.sum ./

RUN go mod download

COPY . .

RUN go build -o main .


# Final stage

FROM alpine:latest

RUN apk --no-cache add ca-certificates

WORKDIR /root/

COPY --from=builder /app/main .

CMD ["./main"]
```

By using above methods, we can create a Docker image.