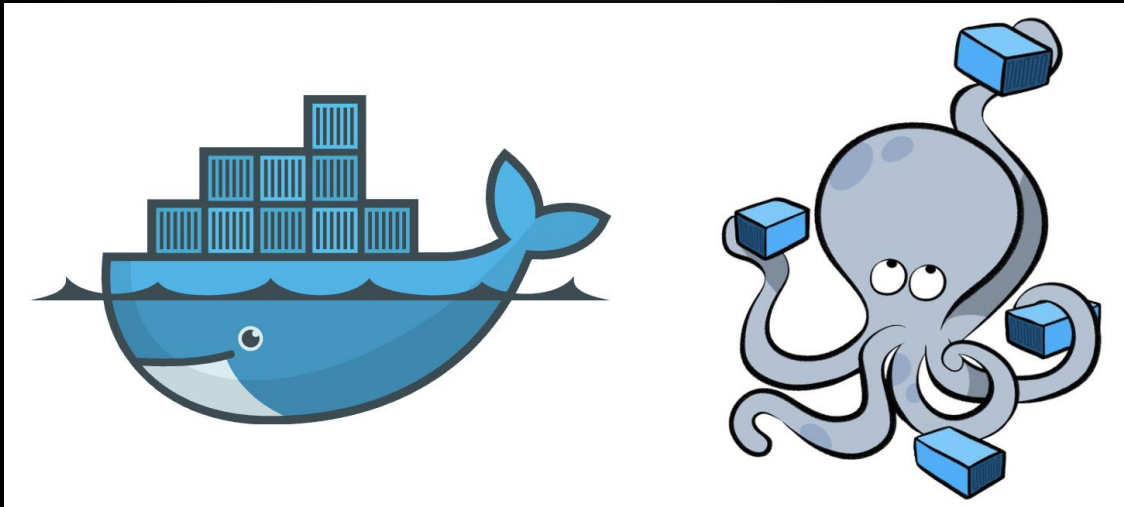


# Multi Container Apps with Docker Compose

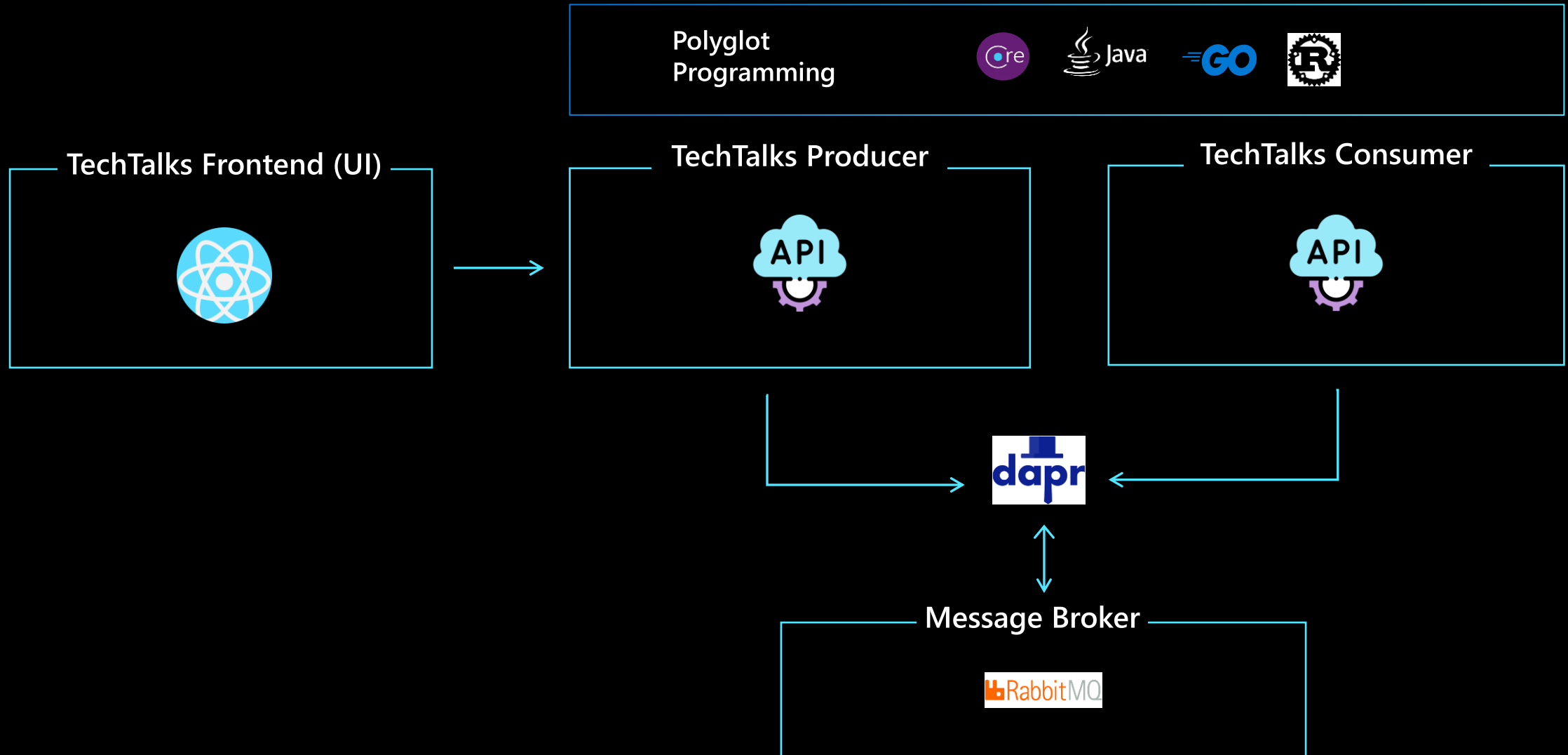
Nilesh Gule @nileshgule



## Cloud Native Ninja



# TechTalks Application Architecture



# Dockerfile – Multistage build

```
FROM mcr.microsoft.com/dotnet/sdk:7.0-alpine AS build-env

WORKDIR /app

COPY producer /producer/
COPY TechTalksModel /TechTalksModel/

WORKDIR /producer
RUN dotnet restore

RUN dotnet publish --configuration Release --output releaseOutput --no-restore

#build runtime image with Alpine base image
FROM mcr.microsoft.com/dotnet/aspnet:7.0-alpine

WORKDIR /producer

COPY --from=build-env /producer/releaseOutput ./

ENTRYPOINT ["dotnet", "producer.dll"]
```

# Docker run

```
docker run --rm -it --hostname rabbitmq --name techtalks-rabbit -p 5672:5672 -p 15672:15672 rabbitmq:3-management
```

## Docker run RabbitMQ

```
docker run -d --hostname rabbitmq `
--name techtalks-rabbit `
-p 5672:5672 `
-p 15672:15672 `
-e RABBITMQ_DEFAULT_USER=avatar `
-e RABBITMQ_DEFAULT_PASS=JamesBond007 `
rabbitmq:3-management
```

## Docker build

```
docker build `
-f .\Dockerfile-v3-TechTalksProducer `
-t nileshgule/techtalksproducer:v3 .
```

```
docker build `
-f .\Dockerfile-TechTalksConsumer `
-t nileshgule/techtalksconsumer:dotnet .
```

## Docker run

```
docker run --rm `
-it `
--name techtalksproducer `
-p 5000:80 `
-e ASPNETCORE_ENVIRONMENT=Development `
nileshgule/techtalksproducer:dotnet
```

```
docker run --rm `
-it `
--name techtalksconsumer `
-p 5001:80 `
-e ASPNETCORE_ENVIRONMENT=Development `
nileshgule/techtalksconsumer:dotnet
```

# Docker Compose - RabbitMQ

```
version: '3.4'

services:
  rabbitmq:
    image: rabbitmq:3-management-alpine
    ports:
      - "15672:15672"
      - "5671:5671"
      - "5672:5672"
    environment:
      RABBITMQ_DEFAULT_PASS: guest
      RABBITMQ_DEFAULT_USER: guest
```

- ✓ **version**: Docker Compose specification version
- ✓ **Services**: List of services used by application
- ✓ **rabbitmq**: name of the service with all the attributes like
  - ✓ Image
  - ✓ Ports
  - ✓ Environment variables

# Docker Compose – Techtalks Application specific

```
version: "3"

services:
  techtalks.producer:
    image: nileshgule/techtalksproducer:dotnet
    build:
      context: src/dotnet
      dockerfile: Dockerfile-v3-TechTalksProducer

  techtalks.consumer:
    image: nileshgule/techtalksconsumer:dotnet
    build:
      context: src/dotnet
      dockerfile: Dockerfile-TechTalksConsumer
```

- ✓ **build**: settings related to building the image
- ✓ **context**: Build context passed to Docker engine at the time of building the image
- ✓ **dockerfile**: name of the Dockerfile used to build the container image

# Docker Compose Commands

 **build**: builds services

 **up**: create and start containers

 **down**: stops and removes containers, networks

 **start**: start services

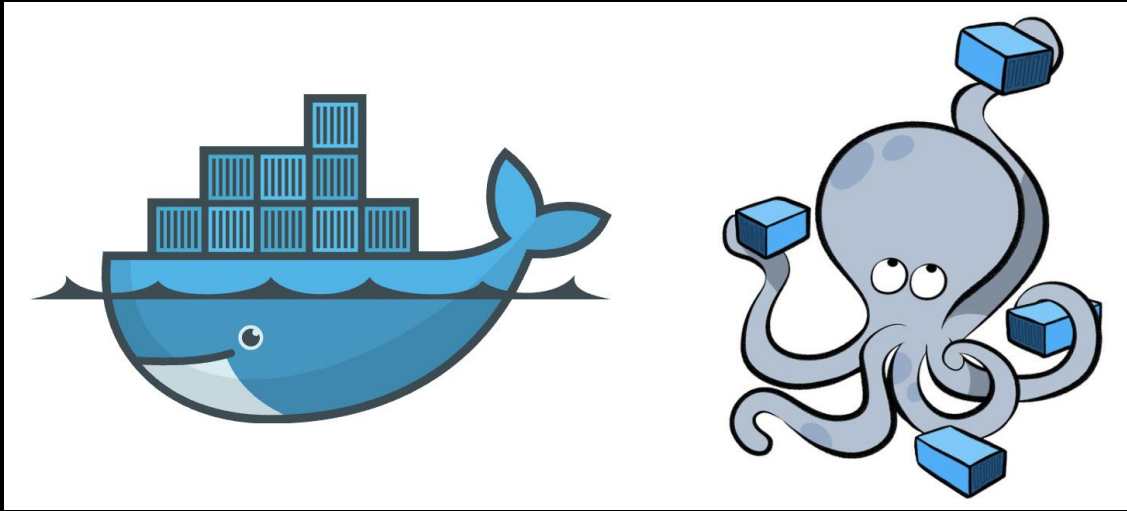
 **stop**: stop services

 **ps**: list containers

## Commands:

build	Build or rebuild services
convert	Converts the compose file to platform's canonical format
cp	Copy files/folders between a service container and the local filesystem
create	Creates containers for a service.
down	Stop and remove containers, networks
events	Receive real time events from containers.
exec	Execute a command in a running container.
images	List images used by the created containers
kill	Force stop service containers.
logs	View output from containers
ls	List running compose projects
pause	Pause services
port	Print the public port for a port binding.
ps	List containers
pull	Pull service images
push	Push service images
restart	Restart service containers
rm	Removes stopped service containers
run	Run a one-off command on a service.
start	Start services
stop	Stop services
top	Display the running processes
unpause	Unpause services
up	Create and start containers
version	Show the Docker Compose version information

# Summary



- ✓ **Local Dev environment:** Easily create all services for an application using Docker Compose
- ✓ **End to end test:** Easy to setup dependencies and run end to end test locally improving developer productivity
- ✓ **Combine compose files:** Combine multiple compose files to form overall application services



# Docker Compose resources

- [Compose Overview](#)
- [Compose Evolution](#)
- [Compose Versioning](#)
- [Devhints Compose cheat sheet](#)
- [Docker Compose gist](#)

# Cloud Native Ninja Resources



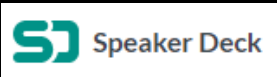
Cloud Native Ninja GitHub repo: <https://github.com/NileshGule/cloud-native-ninja>



## Slides



Slideshare: <https://www.slideshare.net/nileshgule/>



Speaker Deck: <https://speakerdeck.com/nileshgule/>