

Docker

Fast and secure deployment

+ A brief introduction



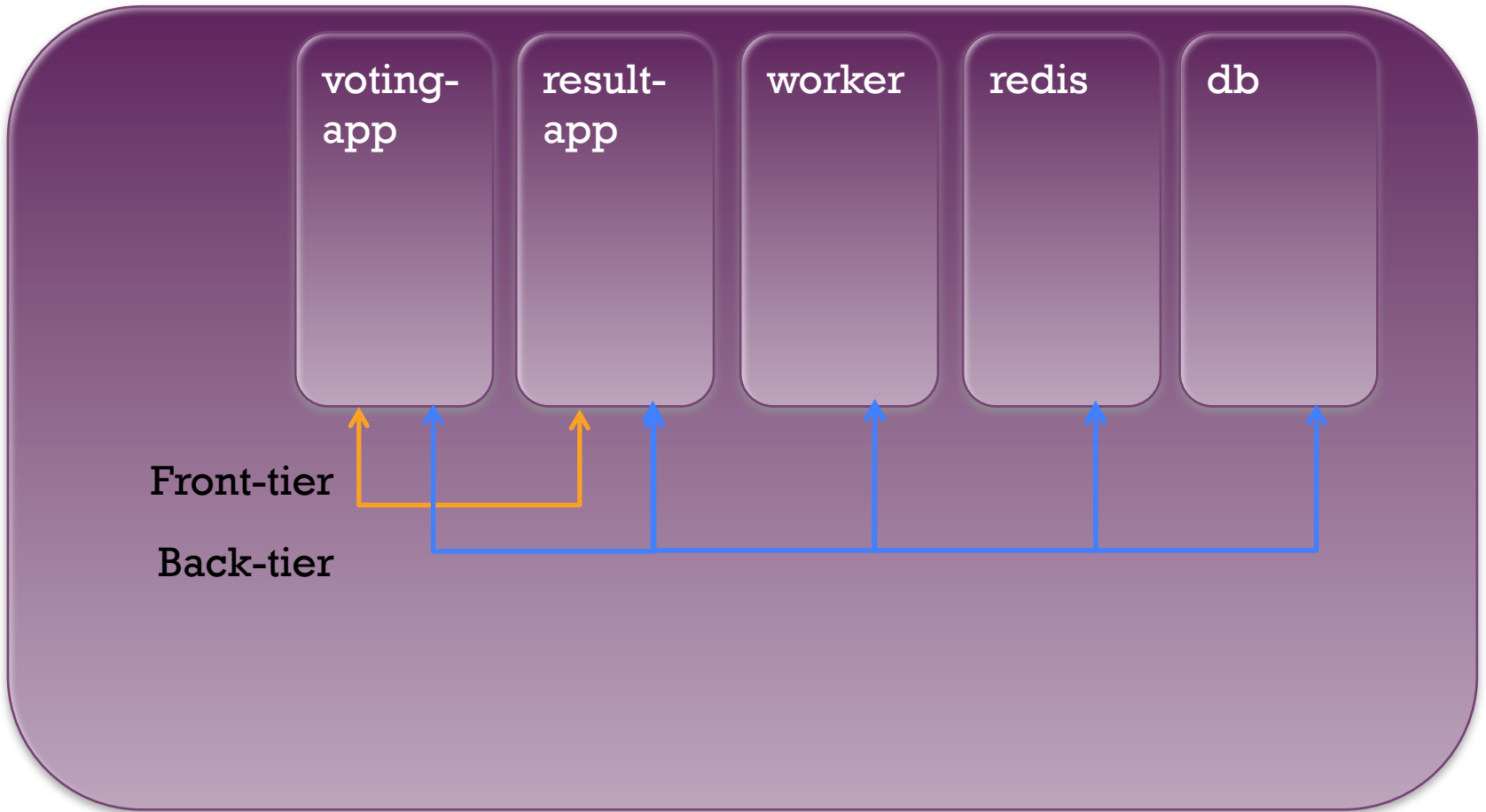
- Docker works with containers
- Containers are 'running' images
- Containers isolate processes from each other
- One image can be the base for many containers -> scalability
- Containers can run on different hosts
- Virtual networks provide connectivity

+ Example Voting App



- Taken from: <https://github.com/docker/example-voting-app>
- A Python webapp which lets you vote between two options
- A Redis queue which collects new votes
- A Java worker which consumes votes and stores them in...
- A Postgres database backed by a Docker volume
- A Node.js webapp which shows the results of the voting in real time
- All services are connected via a virtual network

+ Architecture





docker-compose file snippet'\'



- version: "2"
- services:
 - voting-app:
 - build: ./voting-app/.
 - volumes:
 - - ./voting-app:/app
 - ports:
 - - "5000:80"
 - links:
 - - redis
 - networks:
 - - front-tier
 - - back-tier

+ Dockerfile snippet

- # Using official python runtime base image
- FROM python:2.7-alpine
- # Set the application directory
- WORKDIR /app
- # Install our requirements.txt
- ADD requirements.txt /app/requirements.txt
- RUN pip install -r requirements.txt
- # Copy our code from the current folder to /app inside the container
- ADD ./app
- # Make port 80 available for links and/or publish
- EXPOSE 80
- # Define our command to be run when launching the container
- CMD ["python", "app.py"]

+ Let's go life... Demo



- I'll start the apps...
- Show the host and its processes
- Show the container and its processes
- You can vote via `http:<ip-host>:5000`
- The result is visible via `http:<ip-host>:5001`
- And we can zoom in on the build process