Deploying Flink Jobs as Docker Containers

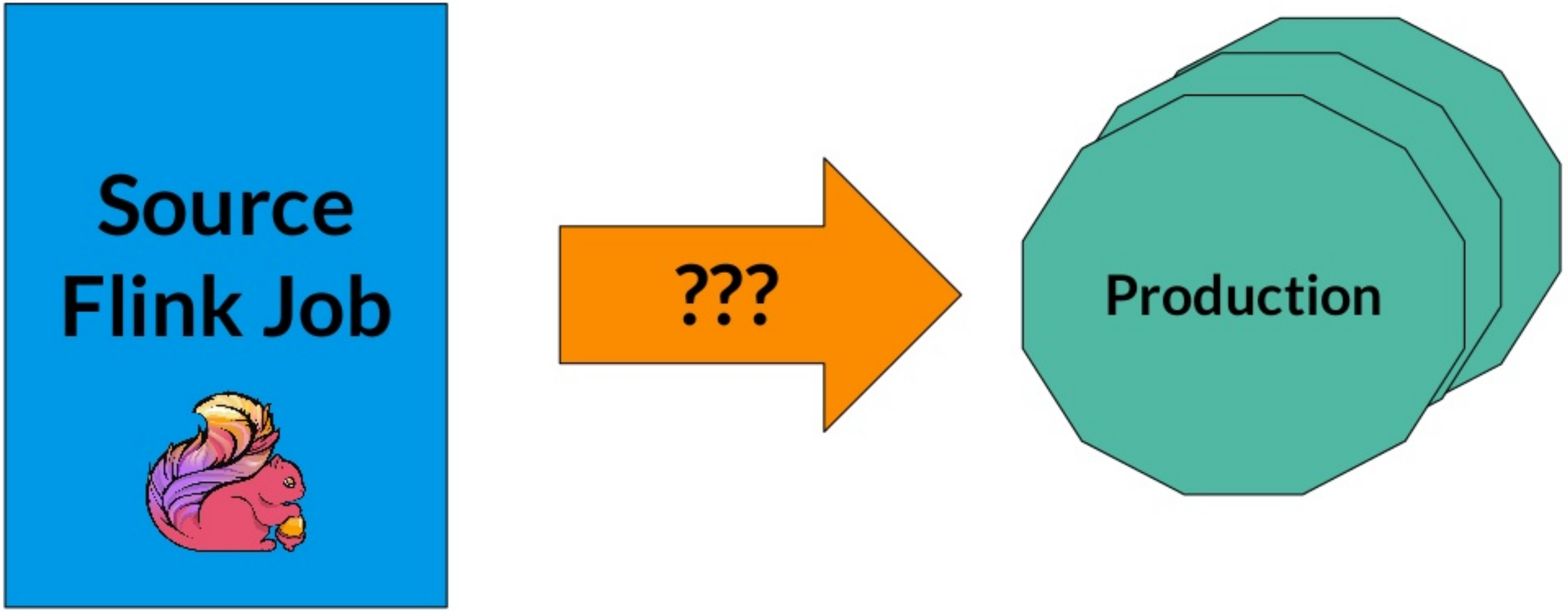
Dominik Bruhn - Director of Platform Engineering Relayr

Flink Forward Sep 12-14 2017 Berlin

Who am I?

- Director of Platform Engineering
- At Relayr
  - Industrial IOT Platform
  - Real Time Sensor Data is processed, stored, analyzed and presented
- Contributor to Apache Flink

What is this talk about?



What are the Requirements?

- Streaming Jobs
- Endless Jobs
- Deployed in Multiple Environments (Configuration Files)
- Single Deployed Artefact
- Deploy to YARN (EMR) cluster
- Repeatable Builds

What is the Idea?

"If it behaves like a service, package it like a service"

- Docker Container contains the Job + Flink + Some scripting
- Docker Container submits and monitors the Flink Job in the YARN cluster.
- Actual computation happens in the YARN cluster
- Docker Container stays attached to the job

Steps Necessary I - Building

Source Flink Job

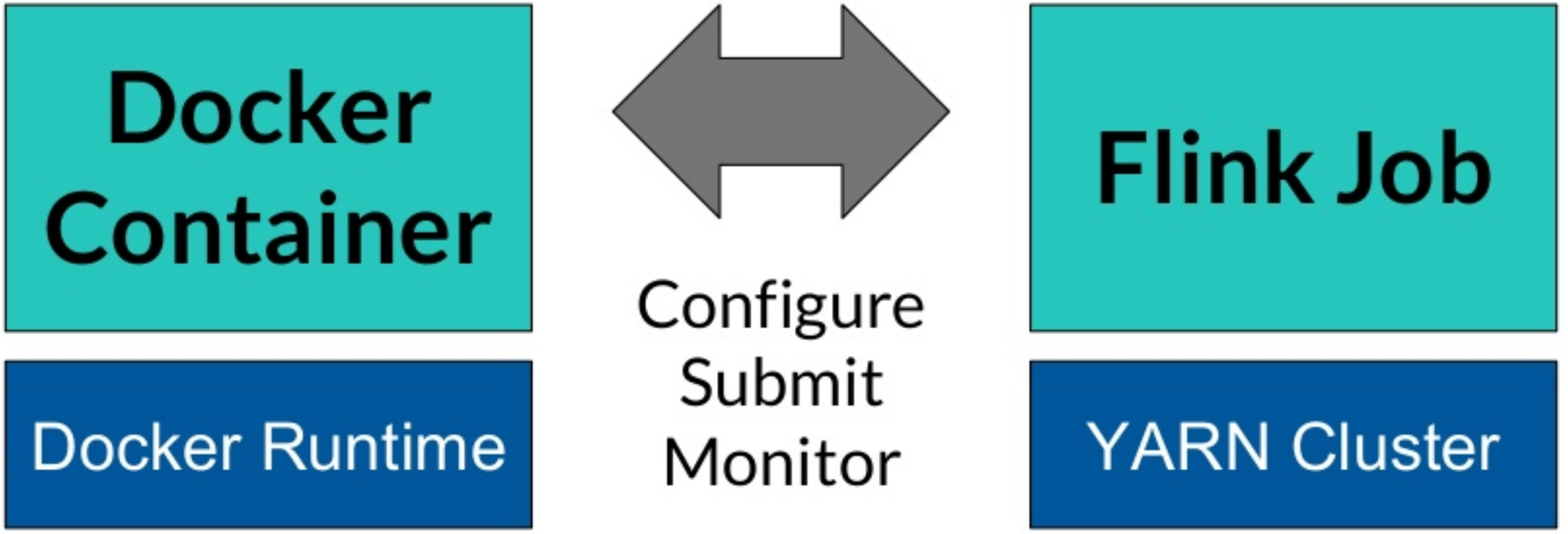
Compile
Test
Package
Upload

Docker Container

Packaging

- Result from compliation: Job Fat JAR
  - Exclude Flink from the Fat JAR
- Package into a Docker Container:
  - Java
  - Apache Flink Distribution
  - Job Fat JAR
  - Configuration File (Templates)
  - If needed: Utilities for Configuration Fetching
  - Entrypoint Shell Script

Steps Necessary II - Executing



What does the Container do?

1. Fetch Configuration Values + YARN Credentials
2. Get YARN Configuration
3. Update Configuration File of Job + Flink
4. List YARN Jobs, find old running
    a. If found, kill
5. Find out the last savepoint on HDFS
6. Start job on YARN from savepoint
7. Stay attached to the Job

What do we get from this?

- Flink Job deployed as all other services
- Adapting to different environment
- Monitoring and failing like other services

What was left out?

- Packaging as stand alone docker container (i.e. for testing)

What could be done in the future?

- Approach independent of YARN on a stand alone Flink cluster.
- Use other resource management tools instead of YARN, i.e. kubernetes.

Thanks for your attention!

Hiring!

relayr.

docker