

Component load testing Framework

Agenda

- The why
- The what
- The tool stack
 - BZT
 - Jmeter

Demo

- Running locally
 - Jmeter / CLI
- Running in the pipeline
 - Task groups
 - Universal packages
 - Helm chart
- Publishing results

Defining our 'Why'

- App's to prod faster
- Reduced maintenance
- Reduce risk
- Manageable + scalable
- Consistency
- Accountability



Defining our 'what'

- Run locally and at each CD stage
- Uses `docker` and `K8s`
- Use of `Helm` templates (run as a job)
- Includes reporting, metrics and dashboards
- Defined goals and expectations
- Out of the box solution

The Load-testing stack



BlazeMeter



docker



kubernetes



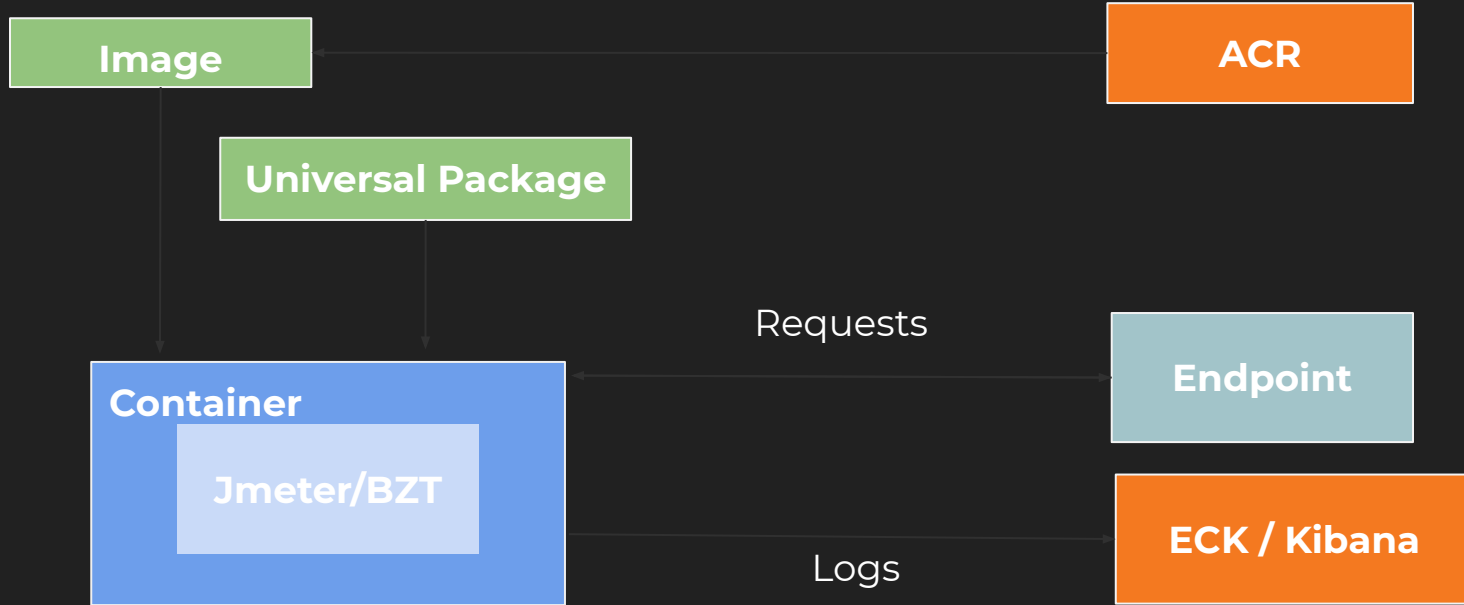
BZT -> Blazemeter taurus

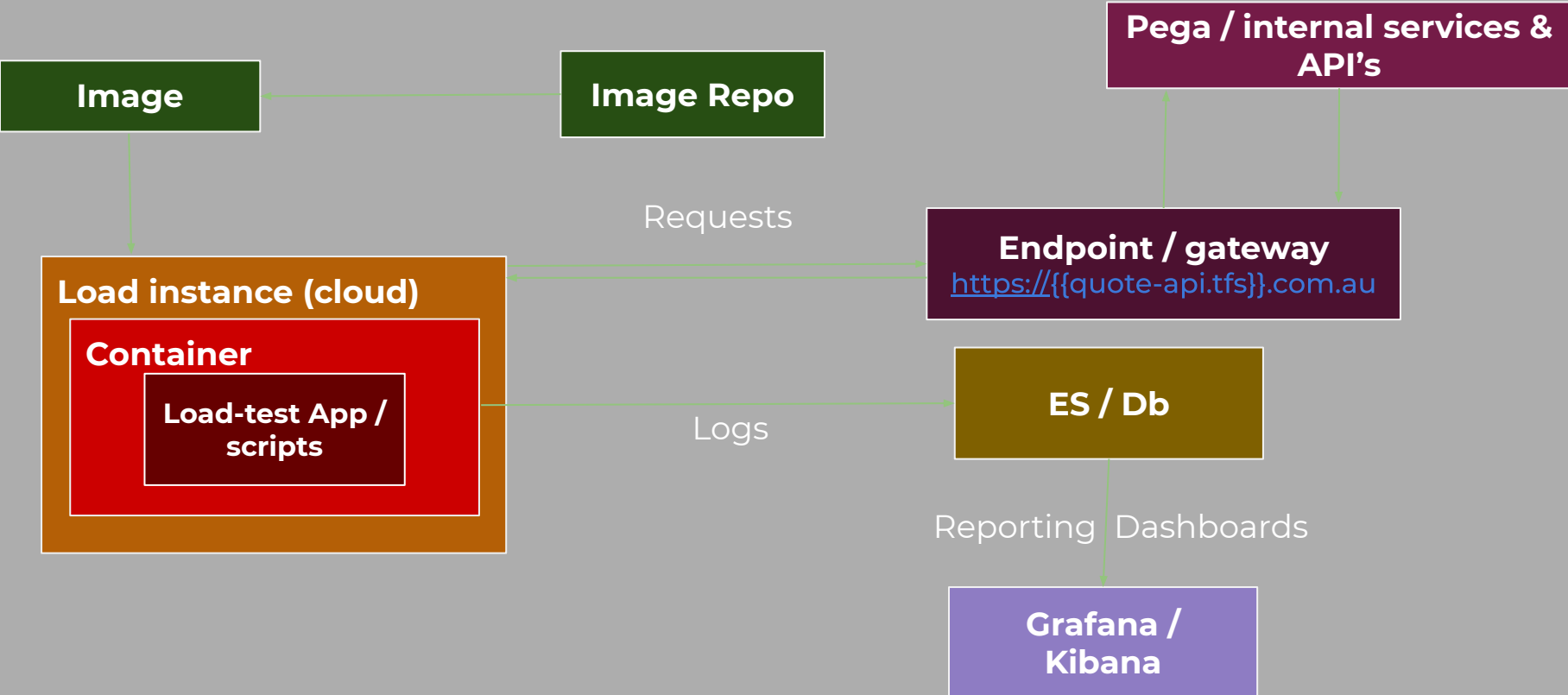
- A wrapper around jmeter
- Provides executors for:
 - Jmeter, Gating, Selenium
- Additional logging
- Test scenarios as yaml files

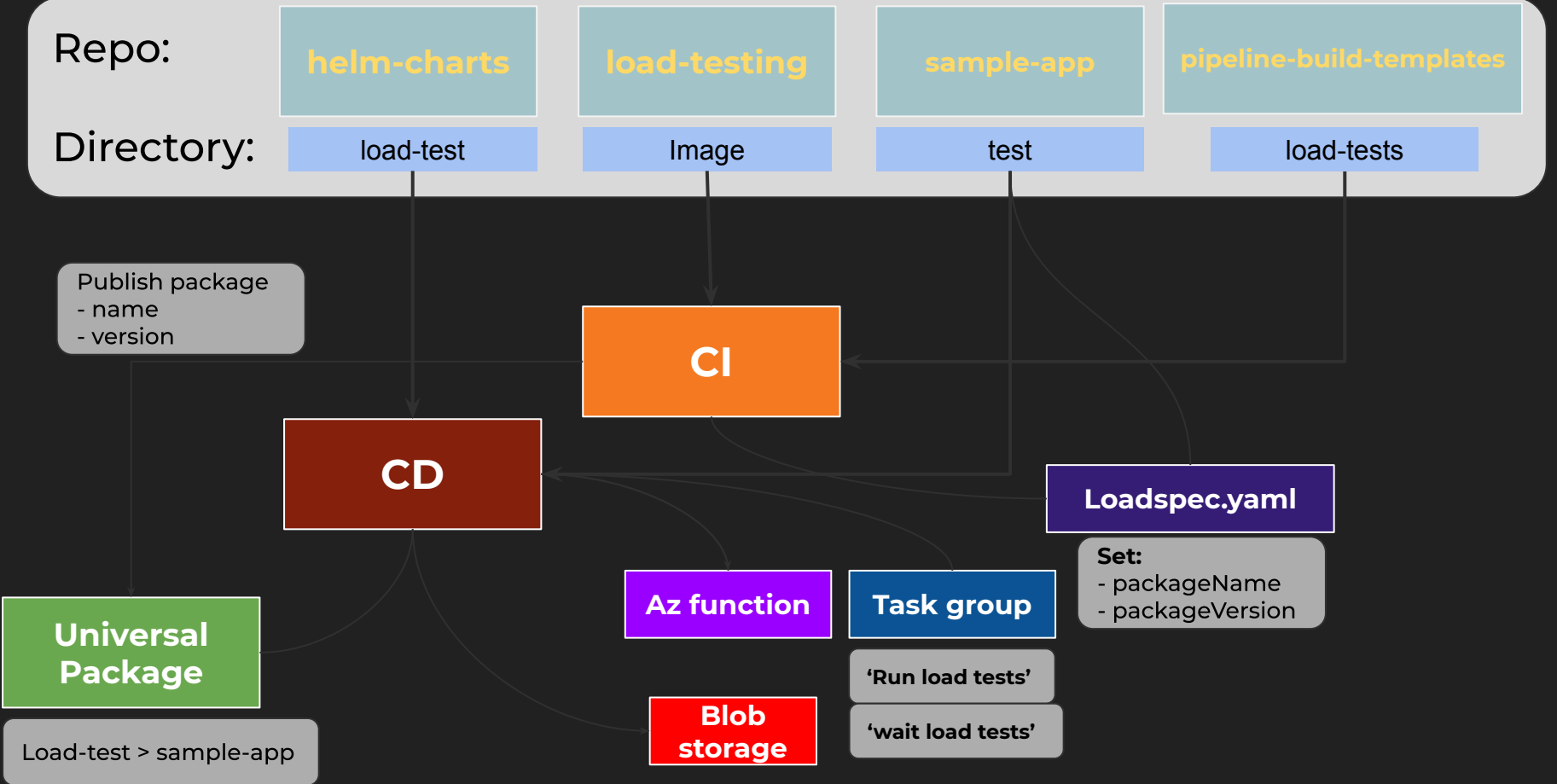
Jmeter

- Open source performance test tool
- Loads of plugins
- Provides a UI for tweaking tests
- Big community

Local Load-tests







Repo:

helm-charts

load-testing

sample-app

pipeline-build-templates

Directory:

load-test

Image

test

load-tests

Package

- name
- version

CI

CD

Loadspec.yaml

Publish Package

Lambda

Task group

Set:

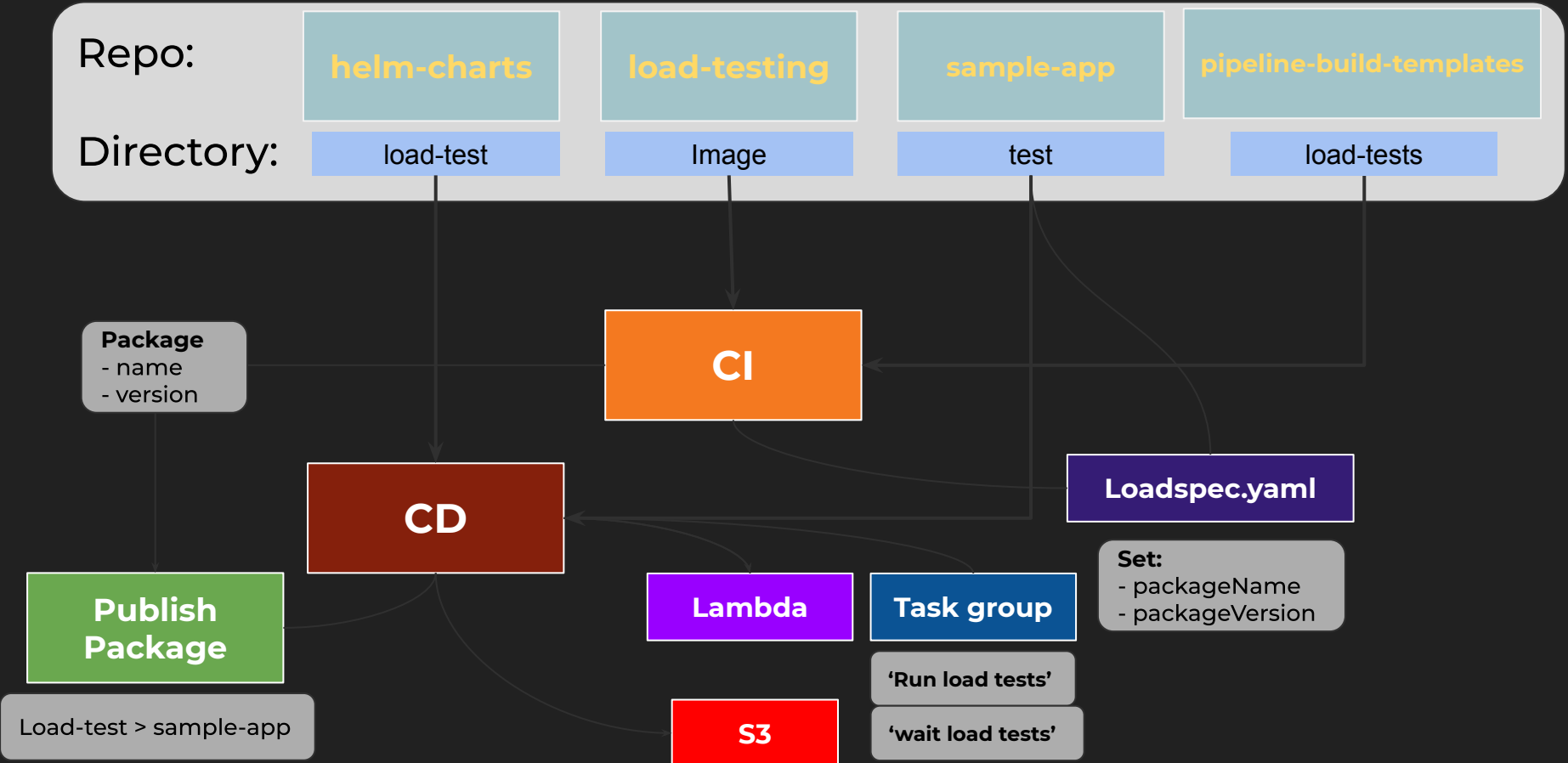
- packageName
- packageVersion

'Run load tests'

'wait load tests'

Load-test > sample-app

S3



Demo

**START WHERE
YOU ARE.**

**USE WHAT
YOU HAVE.**

DO WHAT YOU CAN.

ARTHUR ASHE

SCRATCH PAPER STUDIO

Running locally

- **Jmeter**
 - **Jmx**
 - **Throughput, var's**
- **CLI**
 - **Workdir / logs**
 - **run-project.sh**
- **Blazemeter**
- **Entrypoints**
- **Logging / dashboards**

Running In the build pipeline

- **CI (sampleApp / Image)**
 - **Universal packages**
- **CD SampleApp**
- **Task groups (run and wait)**
- **Callback function**
- **Failure Logs**
- **Dashboard / reporting**

Thankyou

References:

[Jmeter](#)

[Blazemeter](#)

[Backend Listener](#)

[Wiki](#)

[Universal packages](#)