
INNOVATING
DEMOCRACY WITH JAVA
EE AND OPEN SOURCE



@ALEXSOTOB - LORDOFTHEJARS.COM



Apache Tomcat

By Alex Soto Bueno and Romain Manni-Bucau

ABOUT APACHE TOMCAT

Apache Tomcat is a pure Java open-source web server that implements the Java Servlet, JavaServer Pages, and Expression Language specifications.

According to the JRebel report, Tomcat is one of the most used web servers in the Java world with more than 50% of the market share.

INSTALLATION

DOWNLOAD

The quickest way to run Tomcat is to download and run a compiled version. Go to <http://tomcat.apache.org/> and in the Download section choose the Tomcat version that fits your requirements and package file depending on your OS.

TOMCAT VERSION	SPECIFICATION
Tomcat 8.0.x	Servlet 3.1 / JSP 2.3 / EL 3.0 / WebSocket 1.1 / Java 7 and later
Tomcat 7.0.x	Servlet 3.0 / JSP 2.2 / EL 2.2 / WebSocket 1.1 / Java 6 and later (WebSocket requires Java 7)
Tomcat 6.0.x	Servlet 2.5 / JSP 2.1 / EL 2.1 / Java 5 and later

To run Tomcat, you have to first install a Java Runtime Environment (JRE). Make sure to install the right version depending on the Tomcat version you want to run (see table above).

```
$ wget http://repo.maven.apache.org/maven2/org/apache/tomcat/tomcat/8.0.24/tomcat-8.0.24.tar.gz
$ tar -zxvf apache-tomcat-8.0.24.tar.gz
$ cd apache-tomcat-8.0.24
```

The root directory is known as CATALINA_HOME. Optionally, Tomcat may be configured for multiple instances by defining CATALINA_BASE for each instance. For a single installation, CATALINA_BASE is the same as CATALINA_HOME.

RUNNING

The main script to start Tomcat is `$(CATALINA_HOME)/bin/catalina.sh` and the most used start-up commands are:

COMMAND	DESCRIPTION
debug [-security]	Starts in a debugger
start [-security]	Starts in a separate window (or in the background)
run [-security]	Starts in the current window (in the foreground)

stop [-force]	Stops server waiting up to 5 seconds
jpda start	-force option kills the process if not stopped after 5 seconds Starts under JPDA debugger - Environment variable JPDA_ADDRESS defines the debug address (often just a port) and JPDA_SUSPEND to suspend execution immediately after start-up

When Tomcat is started in the foreground, it can be stopped by pressing Ctrl+C.

There is a Windows package distribution that installs Tomcat as a Service on Windows operating systems. Most Linux distributions have their own packaging.

DIRECTORY LAYOUT

DIRECTORY	DESCRIPTION
/bin	Stores executable files for Windows/*nix systems to start and stop server.
/conf	Stores configuration files.
/lib	Stores libraries to share between all applications. By default, only Tomcat libraries are in this directory.
/logs	Stores Tomcat log files.
/temp	Stores temporary files created using the Java File API.
/webapps	Deploys .war files or exploded web applications.
/work	Stores intermediate files (such as compiled JSP files) during its work.

THE MOST WIDELY USED
OPEN SOURCE
APPLICATION SERVER,
OVER 50% WORLDWIDE!

Apache Tomcat™

Tomitribe

tomcat.apache.org | tomcat.apache.org | tomitribe.com

Arquillian IN ACTION

Alex Soto Bueno
Jason Porter

MEAP





TAKE WHAT

— You —

NEED



QUESTIONS





TEAM

A close-up, slightly blurred photograph of a red toy car. A child's hand is visible on the right, gripping the yellow steering wheel. Inside the car, there are several colorful blocks: a blue one, a green one, and a yellow one. The background is a soft, out-of-focus light color.

ONLY SERVICES

JAX-RS

ONLY SERVICES

JAX-RS
JSON-P

ONLY SERVICES

JAX-RS

JSON-P

Bean Validation

ONLY SERVICES

JAX-RS

JSON-P

Bean Validation

EJB

ONLY SERVICES

JAX-RS

JSON-P

Bean Validation

EJB

JPA

ONLY SERVICES

JAX-RS

JSON-P

Bean Validation

EJB

JPA

CDI

The background of the image is a wooden wall with vertical planks. On the left, a black metal frame, possibly part of a desk or chair, is visible. On the right, there is a white electrical outlet with a black cord plugged into it. The text 'APPLICATION SERVER' is centered on the wall.

APPLICATION SERVER

APACHE
TomEE





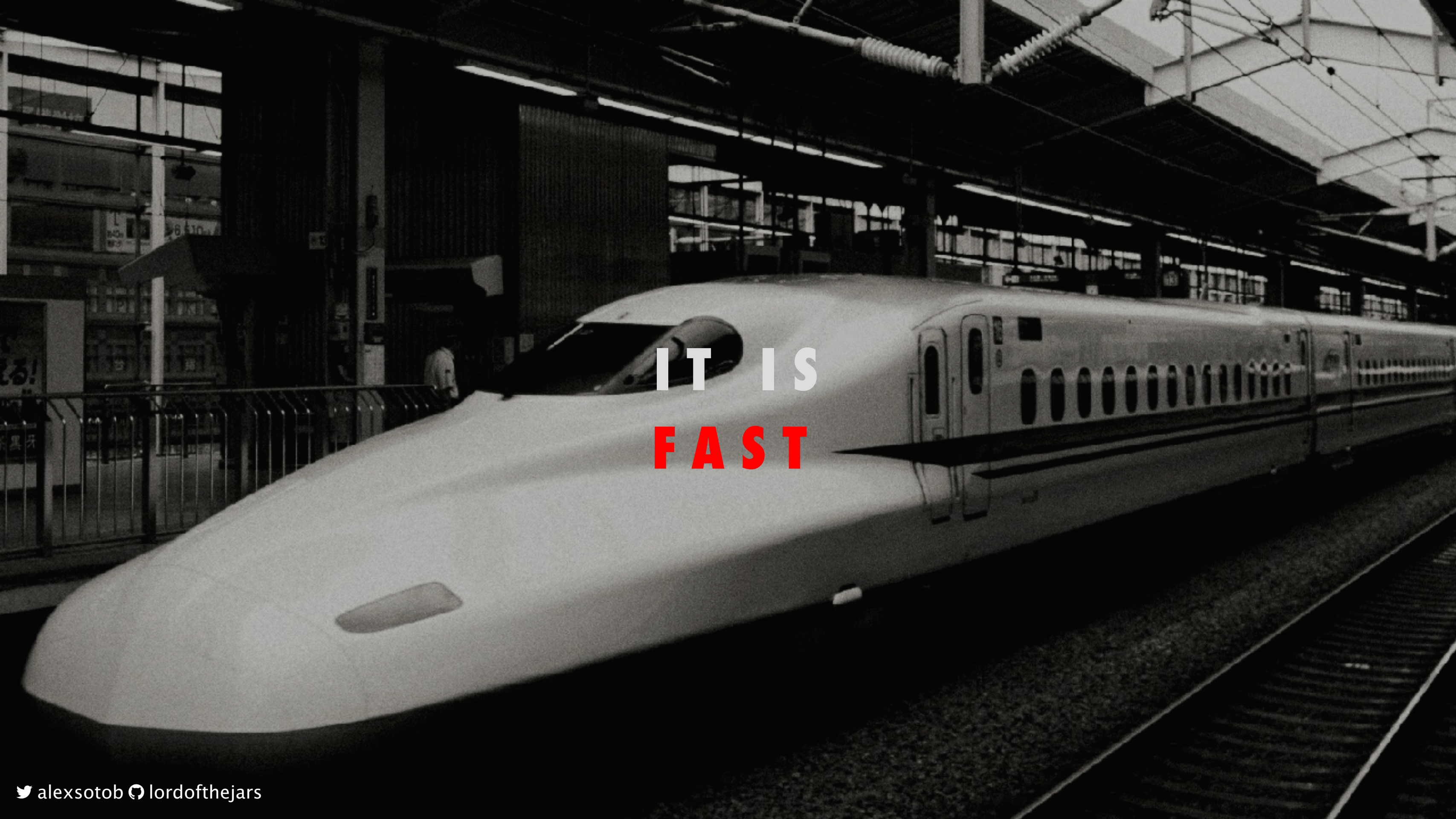
**IT IS A
TOMCAT**




IT IS
CERTIFIED

IT IS
LIGHT





IT IS
FAST



LET'S SEE IN ACTION



JAVAAE

PRACTICE

DEPENDENCIES

```
dependencies {  
    providedCompile group: 'javax', name: 'javaee-api', version: '7.0'  
}
```


JAX-RS

```
@Singleton
@Path("/color")
public class ColorService {

    @PersistenceContext
    EntityManager em;

    @Inject @SHA1MessageDigest
    MessageDigest messageDigest;

    @POST @Path("{color}")
    @Consumes({APPLICATION_JSON})
    public void addColor(
        @PathParam("color") @Size(min = 2, max = 4) String nameColor,
        @Valid Color color) {

        //...
    }
}
```


CDI

```
// Spain SMS JAR
public void sendSMS(@Observes @SMS NotificationEvent event) {
}

// US SMS JAR
public void sendSMS(@Observes @SMS NotificationEvent event) {
}

//Logic
@Inject
@SMS
Event<NotificationEvent> smsEvent;

smsEvent.fire(new NotificationEvent());
```




TO MEE

THE

RESCUE



CONFIGURATION *of* RESOURCES

RESOURCE CONFIGURATION

conf/tomee.xml — WEB-INF/resources.xml

```
<tomee>
  <Resource id="myds" type="DataSource">
    JdbcDriver = org.hsqldb.jdbc.JDBCDriver
    JdbcUrl = jdbc:hsqldb:mem:my-datasource
    Username = SA
    Password =
    minEvictableIdleTime = 30 minutes
  </Resource>

  <Container id="stateless-container" type="STATELESS">
    PoolSize = 50
    StrictPooling = true
  </Container>
</tomee>
```


PLACEHOLDERS

```
<tomee>
  <Resource id="myds" type="DataSource">
    JdbcUrl = ${DB_URL:-jdbc:hsqldb:mem:}movies
  </Resource>
</tomee>

export DB_URL=jdbc:mysql://localhost/
```


SYSTEM PROPERTIES

```
<tomee>
  <Resource id="myds" type="DataSource">
    JdbcUrl = jdbc:hsqldb:mem:movies
  </Resource>
</tomee>

-Dmyds.JdbcUrl=jdbc:mysql://localhost/
```


CUSTOM RESOURCES

```
<tomee>
  <Resource id="mongoClient" class-name="com.mongodb.MongoClient">
    host localhost
  </Resource>
</tomee>
```

```
@Resource(name = "mongoClient")
private MongoClient mongoClient;
```


A close-up photograph of a baby's face. The baby has light skin and blue eyes. Their mouth is covered by a red and white horizontally striped cloth. The baby is lying on a towel with a rainbow pattern of orange, yellow, and blue stripes. The text "PASSWORD ENCRYPTION" is overlaid on the image, with "PASSWORD" in white and "ENCRYPTION" in red.

PASSWORD ENCRYPTION

PASSWORD CIPHER

```
<Resource id="myds" type="DataSource">  
  //.....  
  Password = xMH5uM1V9vQzVUv5LG7YLA==  
  PasswordCipher = Static3DES  
</Resource>
```

```
<Resource id="myresource" class-name="org.superbiz.VaultGateway">  
  //.....  
  VaultPassword = cipher:Static3DES:xMH5uM1V9vQzVUv5LG7YLA==  
</Resource>
```




TESTING AND JAVA EE

A high-angle, wide shot of a large, diverse crowd of people at an outdoor event, possibly a festival or concert. The people are standing, sitting, and walking on a light-colored paved surface. Many are holding drinks, and some are clapping. The crowd is dense and fills the entire frame. Overlaid in the center is the text "WRITE UNIT TESTS" in a bold, sans-serif font. The word "UNIT" is red, while "WRITE" and "TESTS" are white.

WRITE UNIT TESTS

JUNIT

```
@Singleton
@Path("/color")
public class ColorService {
    @PersistenceContext
    EntityManager entityManager;

    @POST
    public void addColor(...) {}

    @GET
    public Color getColor(...) {}
}
```

```
@Test
public void addAndGetColor() {
    EntityManager em = mock(EntityManager.class);
    //...

    ColorService colorService = new ColorService();
    colorService.entityManager = em;

    colorService.addColor("green");
    assertThat(colorService.getColor(), is("green"));
}
```


A hand-drawn sketch of a box, possibly a shipping container, with its top flaps open. Inside the box, there is a small, dark, insect-like figure with multiple legs and antennae. The drawing is done in a simple, sketchy style with black outlines and some light shading. The text "AVOID SCHRÖDINGER'S CODE" is overlaid on the box.

AVOID SCHRÖDINGER'S CODE

WOULD IT WORK IN **PRODUCTION**?

Entity Manager configuration

WOULD IT WORK IN **PRODUCTION**?

Entity Manager configuration
Transactions

WOULD IT WORK IN **PRODUCTION**?

Entity Manager configuration

Transactions

CDI Qualifiers

WOULD IT WORK IN PRODUCTION?

Entity Manager configuration

Transactions

CDI Qualifiers

CDI Events

WOULD IT WORK IN PRODUCTION?

Entity Manager configuration

Transactions

CDI Qualifiers

CDI Events

JAX-RS

WOULD IT WORK IN **PRODUCTION**?

Entity Manager configuration

Transactions

CDI Qualifiers

CDI Events

JAX-RS

HTTP Servlet Filters



STEPS OF ARQUILLIAN

Choose Application Server

STEPS OF ARQUILLIAN

Choose Application Server

Start Application Server

STEPS OF ARQUILLIAN

Choose Application Server

Start Application Server

(micro)Package the Application

STEPS OF ARQUILLIAN

Choose Application Server

Start Application Server

(micro)Package the Application

Run Tests (In Container or As Client)

STEPS OF ARQUILLIAN

Choose Application Server

Start Application Server

(micro)Package the Application

Run Tests (In Container or As Client)

Stop Application Server

ARQUILLIAN TEST

```
dependencyManagement {
    imports {
        mavenBom
        'org.arquillian:arquillian-universe:1.0.0.Alpha1'
    }
}

testCompile group: 'org.arquillian.universe',
            name: 'arquillian-junit',
            ext: 'pom'
testCompile group: 'org.apache.openejb',
            name: 'arquillian-tomee-embedded',
            version: '1.7.2'
testCompile group: 'org.apache.openejb',
            name: 'tomee-jaxrs',
            version: '1.7.2'
```

```
@RunWith(Arquillian.class)
public class ColorServiceIntegrationTest {

    @Deployment public static WebArchive create() {
        return ShrinkWrap.create(WebArchive.class)
            .addClasses(ColorService.class,
                Color.class);
    }

    @ArquillianResource private URL webappUrl;

    @Test @RunAsClient public void postAndGet() {
        final WebClient webClient = WebClient
            .create(webappUrl);

        final Response response = webClient
            .path("color/green")
            .post(null);

        assertThat(response.getStatus(), is(204));

        //...
    }
}
```




LET'S SEE IN ACTION

D O C K E R

— *not used* —

I N P R O D U C T I O N





STEPS OF ARQUILLIAN

Extension for Docker

STEPS OF **ARQUILLIAN**

Extension for Docker

Support for Native, Boot2Docker and Docker-Machine

STEPS OF **ARQUILLIAN**

Extension for Docker

Support for Native, Boot2Docker and Docker-Machine

Docker-Compose definition

ARQUILLIAN CUBE TEST

```
testCompile group: 'org.arquillian.universe',  
            name: 'arquillian-cube',  
            ext: 'pom'
```

```
<arquillian>  
  <extension qualifier="docker">  
    <property name="definitionFormat">COMPOSE</property>  
    <property name="machineName">dev</property>  
    <property name="dockerContainersFile">compose.yml</property>  
  </extension>
```


ARQUILLIAN TESTS

Integration Tests

ARQUILLIAN TESTS

Integration Tests

Functional Tests (Drone/Graphene)

ARQUILLIAN TESTS

Integration Tests

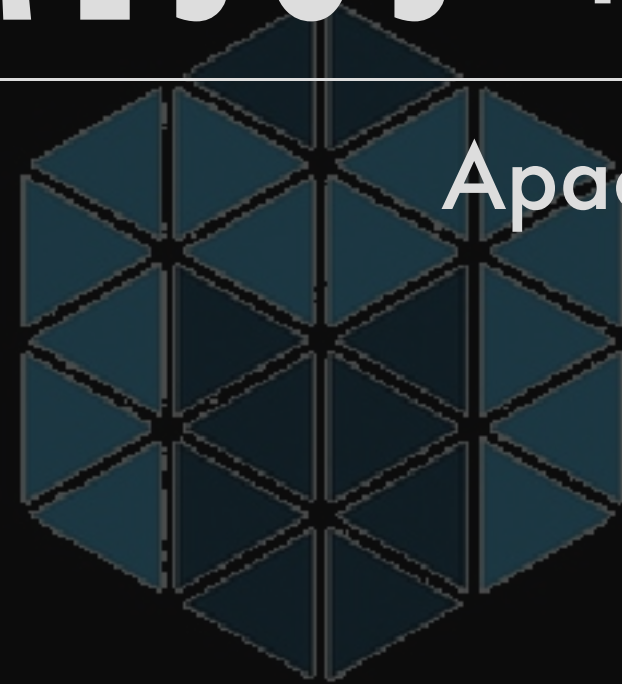
Functional Tests (Drone/Graphene)

BDD Tests (Cucumber)

A wide-angle photograph of a large, modern server room. The room is filled with rows of server racks, some of which are open, revealing internal components. A person is standing in the aisle on the left side of the frame, providing a sense of scale. The room has a high ceiling with a grid of lights and a blue wall in the background. The text "DEPLOYING TO PRODUCTION" is overlaid in the center of the image.

DEPLOYING TO PRODUCTION

MESOS + MARATHON

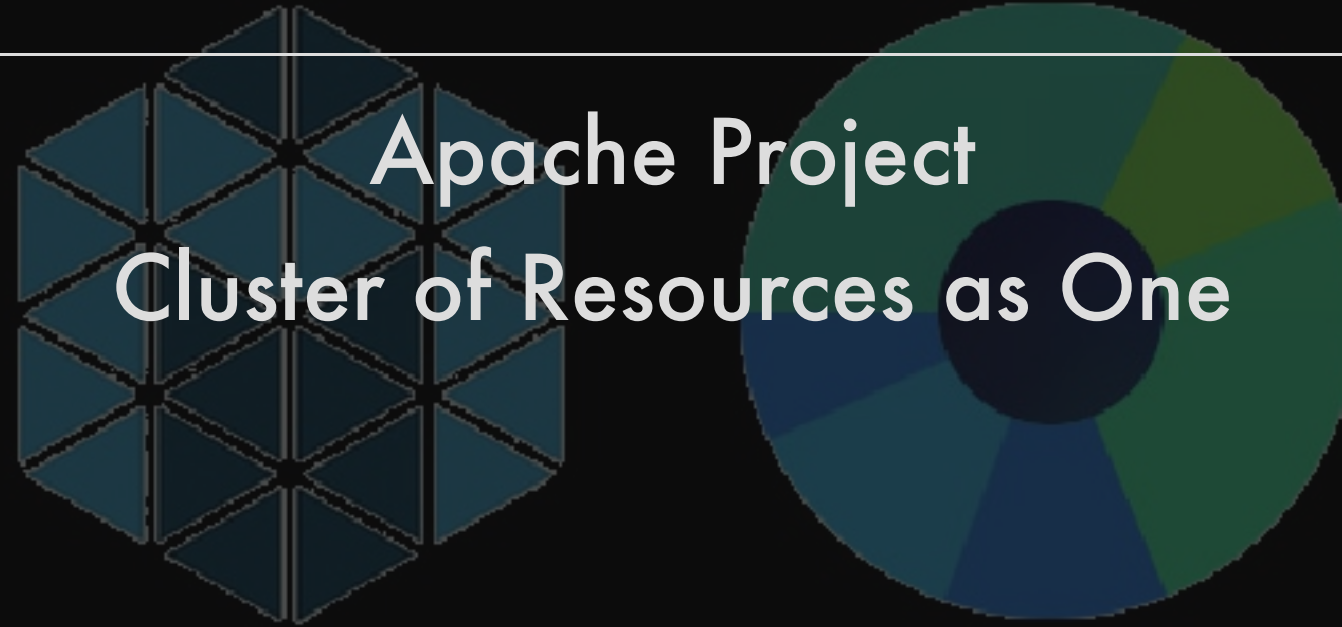


Apache Project

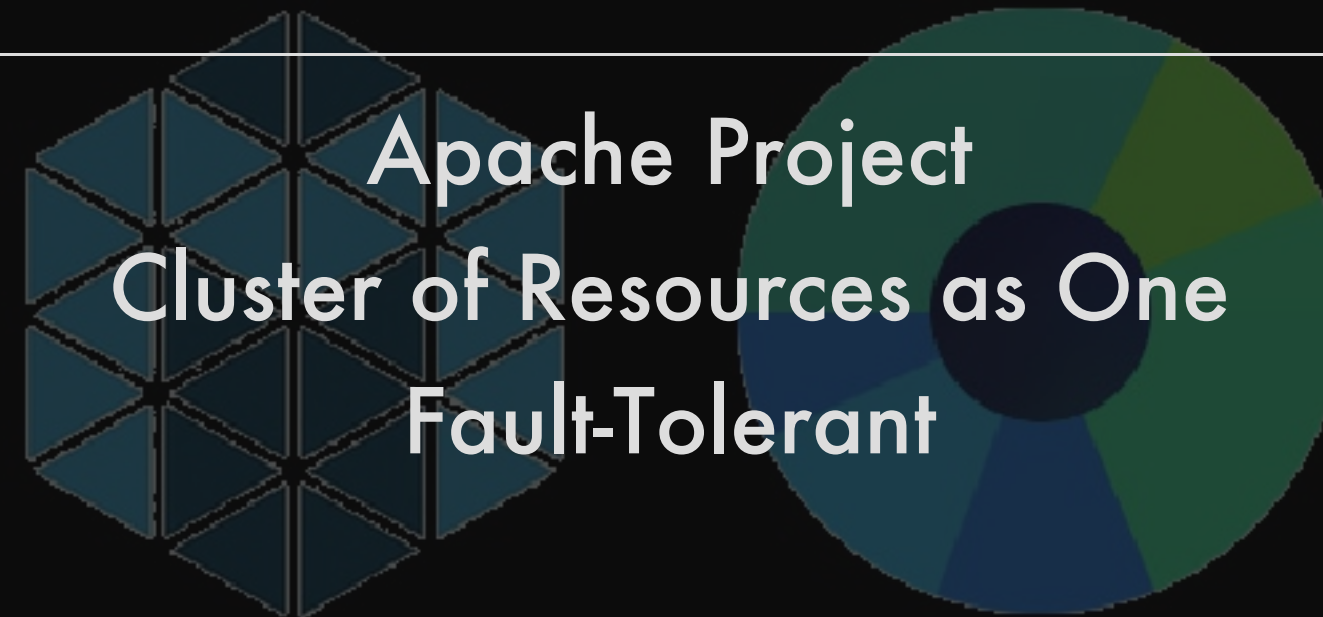


MESOS + MARATHON

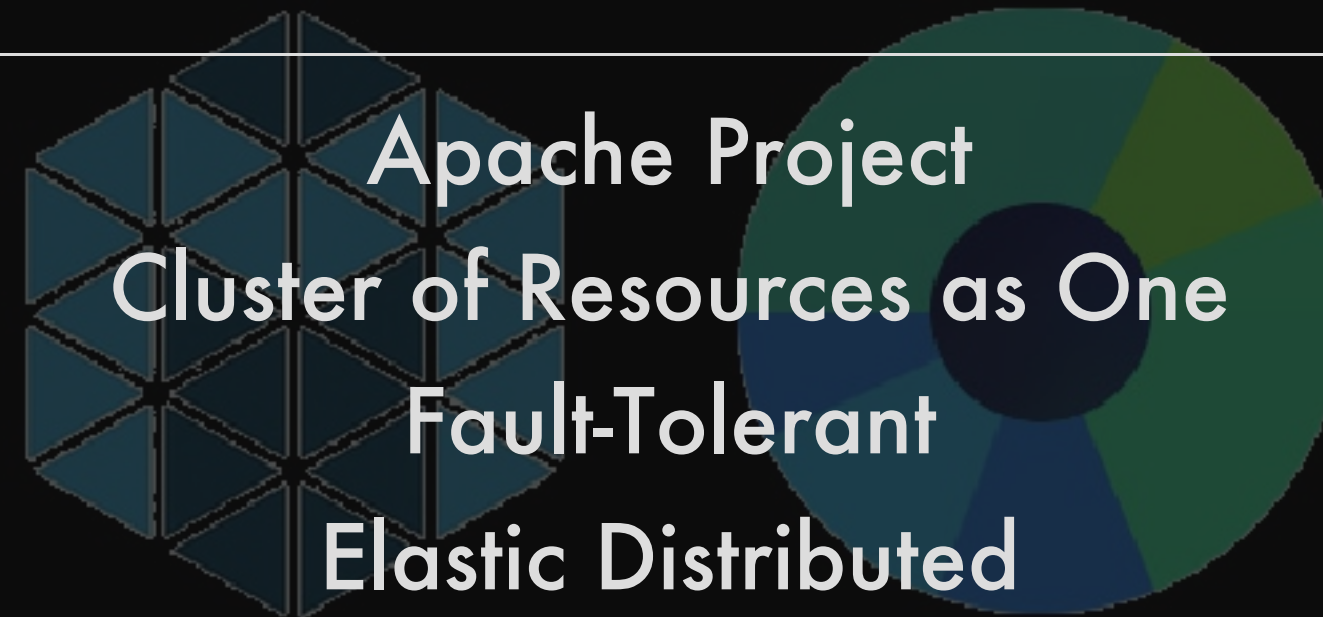
Apache Project
Cluster of Resources as One



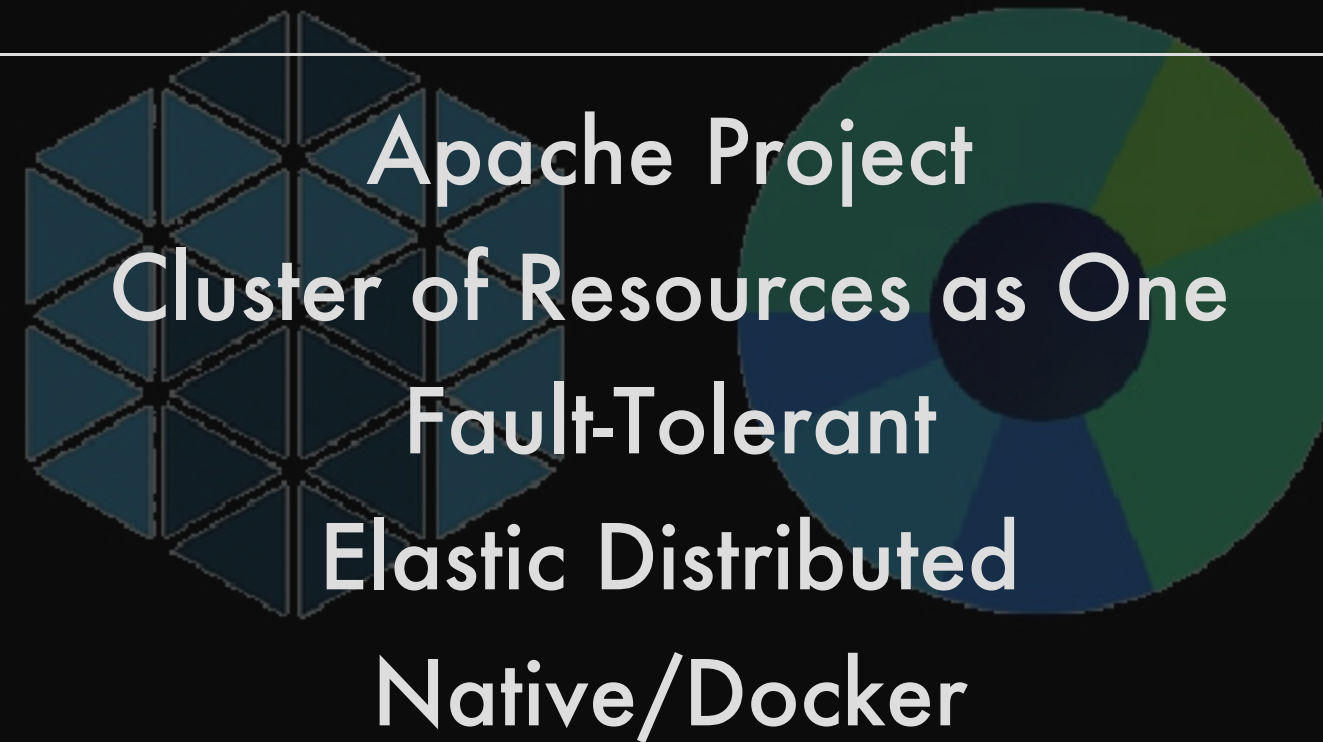
MESOS + MARATHON

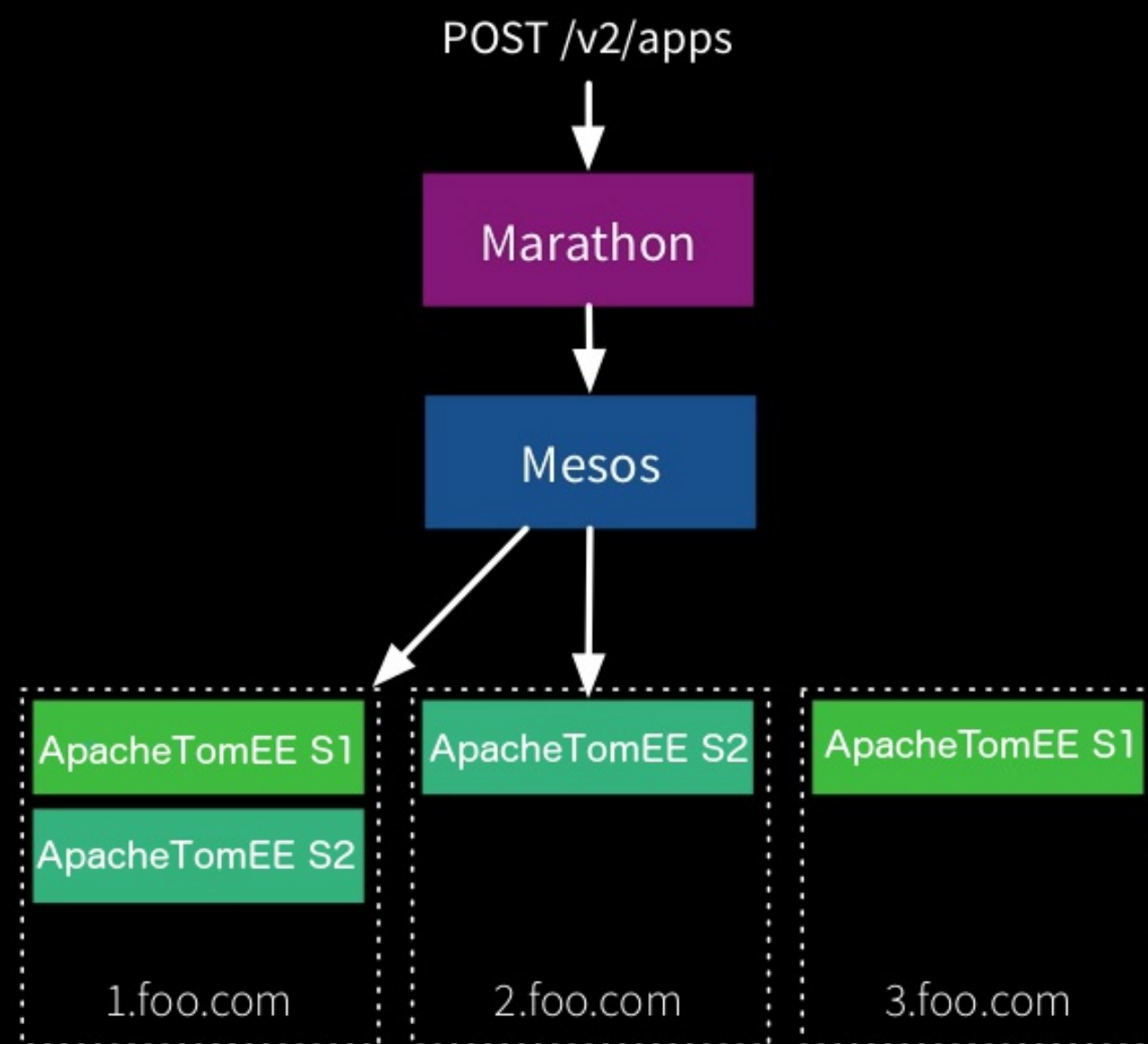


MESOS + MARATHON



MESOS + MARATHON

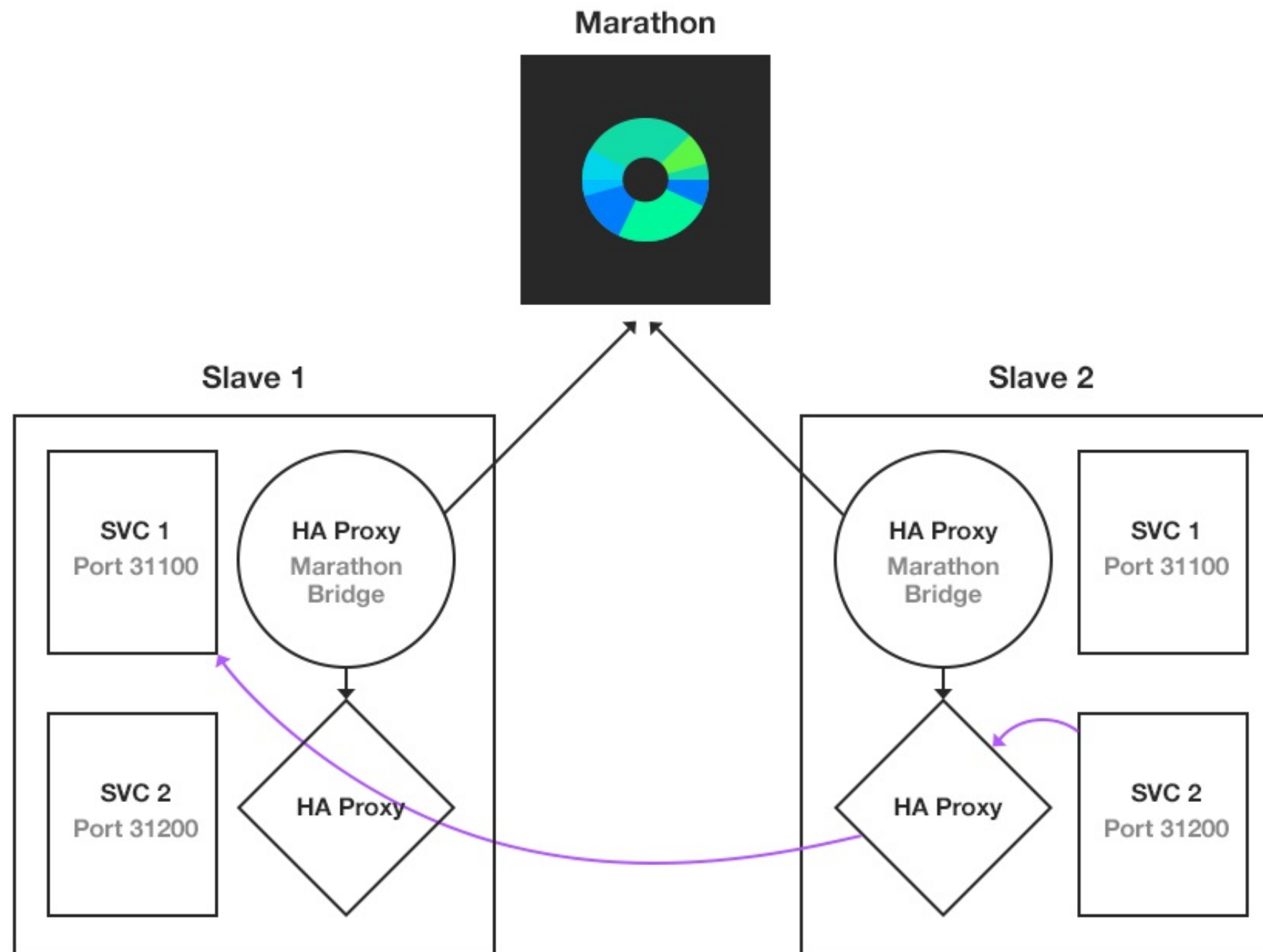




SERVICE DISCOVERY




SERVICE DISCOVERY SCHEMA



R U N A P A C H E T O M E E

```
{
  "id": "projectdemo",
  "cmd": "cd apache-tomee-plus*
        && sed \"s/8080/$PORT/g\" < ./conf/server.xml > ./conf/server-mesos.xml
        && ./bin/catalina.sh run -config ./conf/server-mesos.xml",
  "mem": 256,
  "cpus": 0.5,
  "instances": 1,
  "ports": [10000],
  "constraints": [
    ["hostname", "UNIQUE"]
  ],
  "uris": [
    "http://ftp.cixug.es/apache/tomee/tomee-1.7.2/apache-tomee-1.7.2-plus.tar.gz"
  ]
}
```

LET'S SEE IN ACTION

A serene sunset scene over a calm body of water. The sun is a bright, glowing orb on the horizon, casting a shimmering path of light across the water's surface. The sky is a mix of soft oranges, yellows, and muted blues, with wispy clouds catching the low light. Several birds are captured in flight, their dark silhouettes contrasting against the lighter sky. In the foreground, the silhouettes of people are visible: two figures stand close together in the shallow water, while a few more are scattered further out. The overall mood is peaceful and contemplative.

LET'S WIND DOWN



JAVA EEVS SPRING



http://github.com/lordofthejars/javaee_javaone



asotobu@gmail.com

