

# **@RunWith(Arquillian.class)** **"Production Near" JavaEE** **Testing**

Andy Gumbrecht  
@AndyGeeDe  
#TomEE



# Don't be a Fmek



# Don't be a Fmek



- Befriend Arquillians – That's är–ki–lē–əns



# Don't be a Fmek



- Befriend Arquillians – That's är–ki–lē–əns
  - They're not going to destroy the Earth





# Don't be a Fmek

- Befriend Arquillians – That's är-ki-lē-əns
  - They're not going to destroy the Earth
- Aslak Knutsen





# Don't be a Fmek

- Befriend Arquillians – That's är–ki–lē–əns
  - They're not going to destroy the Earth
- Aslak Knutsen
  - Project Lead, Co-founder @aslakknutsen





# Don't be a Fmek

- Befriend Arquillians – That's är–ki–lē–əns
  - They're not going to destroy the Earth
- Aslak Knutsen
  - Project Lead, Co-founder @aslakknutsen
- Pete Muir





# Don't be a Fmek

- Befriend Arquillians – That's är–ki–lē–əns
  - They're not going to destroy the Earth
- Aslak Knutsen
  - Project Lead, Co-founder @aslakknutsen
- Pete Muir
  - Co-founder @plmuir







# Don't be a Fmek

- Befriend Arquillians – That's är–ki–lē–əns
  - They're not going to destroy the Earth
- Aslak Knutsen
  - Project Lead, Co-founder @aslakknutsen
- Pete Muir
  - Co-founder @plmuir
- Alex Soto





# Don't be a Fmek

- Befriend Arquillians – That's är–ki–lē–əns
  - They're not going to destroy the Earth
- Aslak Knutsen
  - Project Lead, Co-founder @aslakknutsen
- Pete Muir
  - Co-founder @plmuir
- Alex Soto
  - Arquillian Cube @alexsotob



# Don't be a Fmek



- Befriend Arquillians – That's är–ki–lē–əns
  - They're not going to destroy the Earth
- Aslak Knutsen
  - Project Lead, Co-founder @aslakknutsen
- Pete Muir
  - Co-founder @plmuir
- Alex Soto
  - Arquillian Cube @alexsotob



APACHE TOMEE

# @Test



@AndyGeeDe | #TomEE

**Tomitribe**  
join the tribe

# @Test



- Do you test?



# @Test



- Do you test?
  - Testing facilitates change, with correctness as a side effect



# @Test



- Do you test?
  - Testing facilitates change, with correctness as a side effect
- What type of test?



# @Test



- Do you test?
  - Testing facilitates change, with correctness as a side effect
- What type of test?
  - Unit?





# @Test



- Do you test?
  - Testing facilitates change, with correctness as a side effect
- What type of test?
  - Unit?
  - Integration?



# @Test



- Do you test?
  - Testing facilitates change, with correctness as a side effect
- What type of test?
  - Unit?
  - Integration?
  - Functional?



# @Test



- Do you test?
  - Testing facilitates change, with correctness as a side effect
- What type of test?
  - Unit?
  - Integration?
  - Functional?
  - All of the above?



# @Test



- Do you test?
  - Testing facilitates change, with correctness as a side effect
- What type of test?
  - Unit?
  - Integration?
  - Functional?
  - All of the above? You are on the right path!



# @RunWith(Arquillian.class)



# @RunWith(Arquillian.class)



- Tests should be portable



# @RunWith(Arquillian.class)



- Tests should be portable
  - On any supported container



# @RunWith(Arquillian.class)



- Tests should be portable
  - On any supported container
  - Jboss, IBM WebSphere, Tomcat, TomEE, TomEE, etc.





# @RunWith(Arquillian.class)



- Tests should be portable
  - On any supported container
  - Jboss, IBM WebSphere, Tomcat, TomEE, TomEE, etc.
- Tests should be executable



# @RunWith(Arquillian.class)



- Tests should be portable
  - On any supported container
  - Jboss, IBM WebSphere, Tomcat, TomEE, TomEE, etc.
- Tests should be executable
  - On the developer IDE



# @RunWith(Arquillian.class)



- Tests should be portable
  - On any supported container
  - Jboss, IBM WebSphere, Tomcat, TomEE, TomEE, etc.
- Tests should be executable
  - On the developer IDE
  - On the build tool



# @RunWith(Arquillian.class)



- Tests should be portable
  - On any supported container
  - Jboss, IBM WebSphere, Tomcat, TomEE, TomEE, etc.
- Tests should be executable
  - On the developer IDE
  - On the build tool
  - Within existing test frameworks



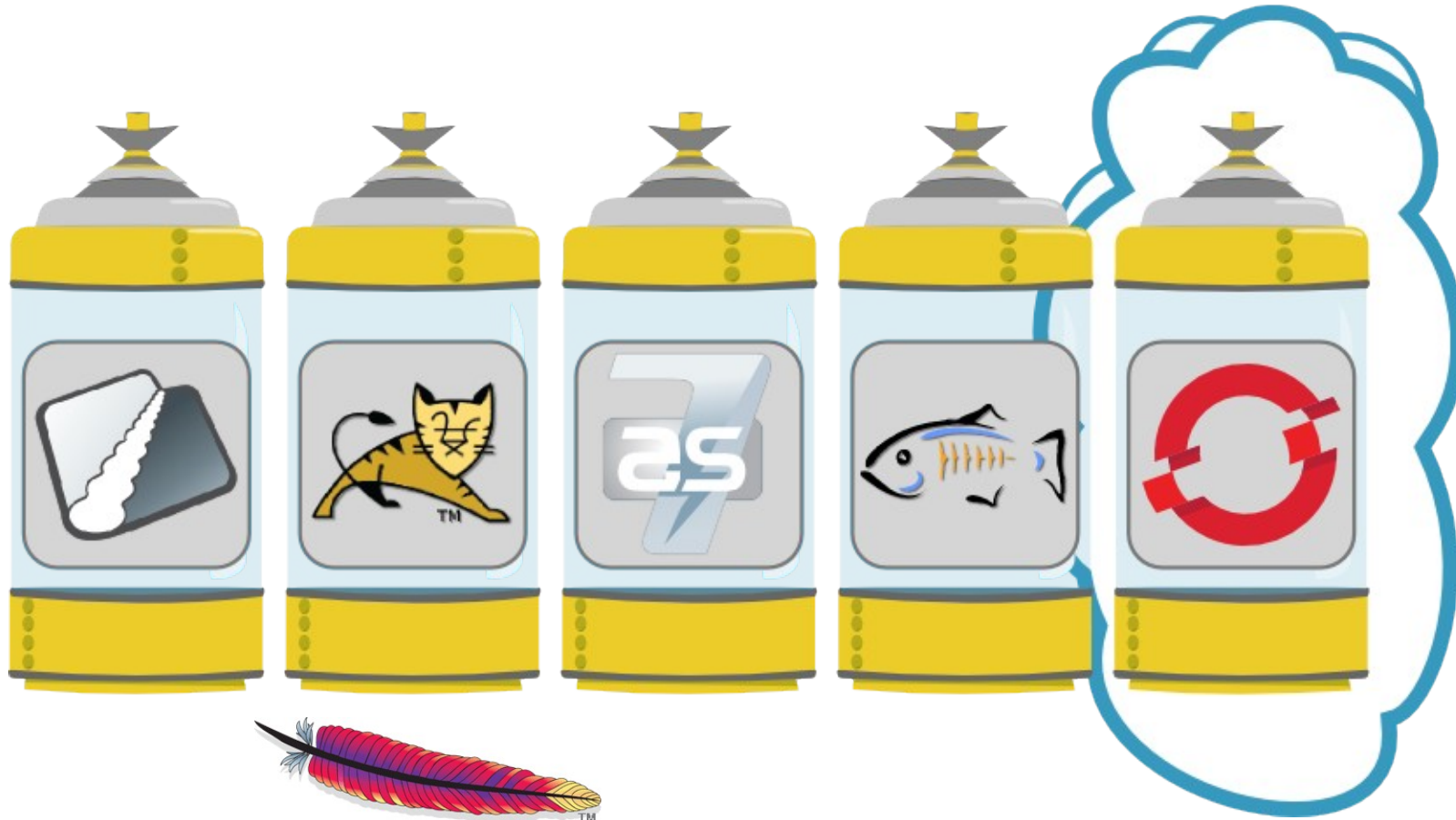
# The Basics





# The Basics

**1** Select a container



# The Basics

2 Start or connect to a container

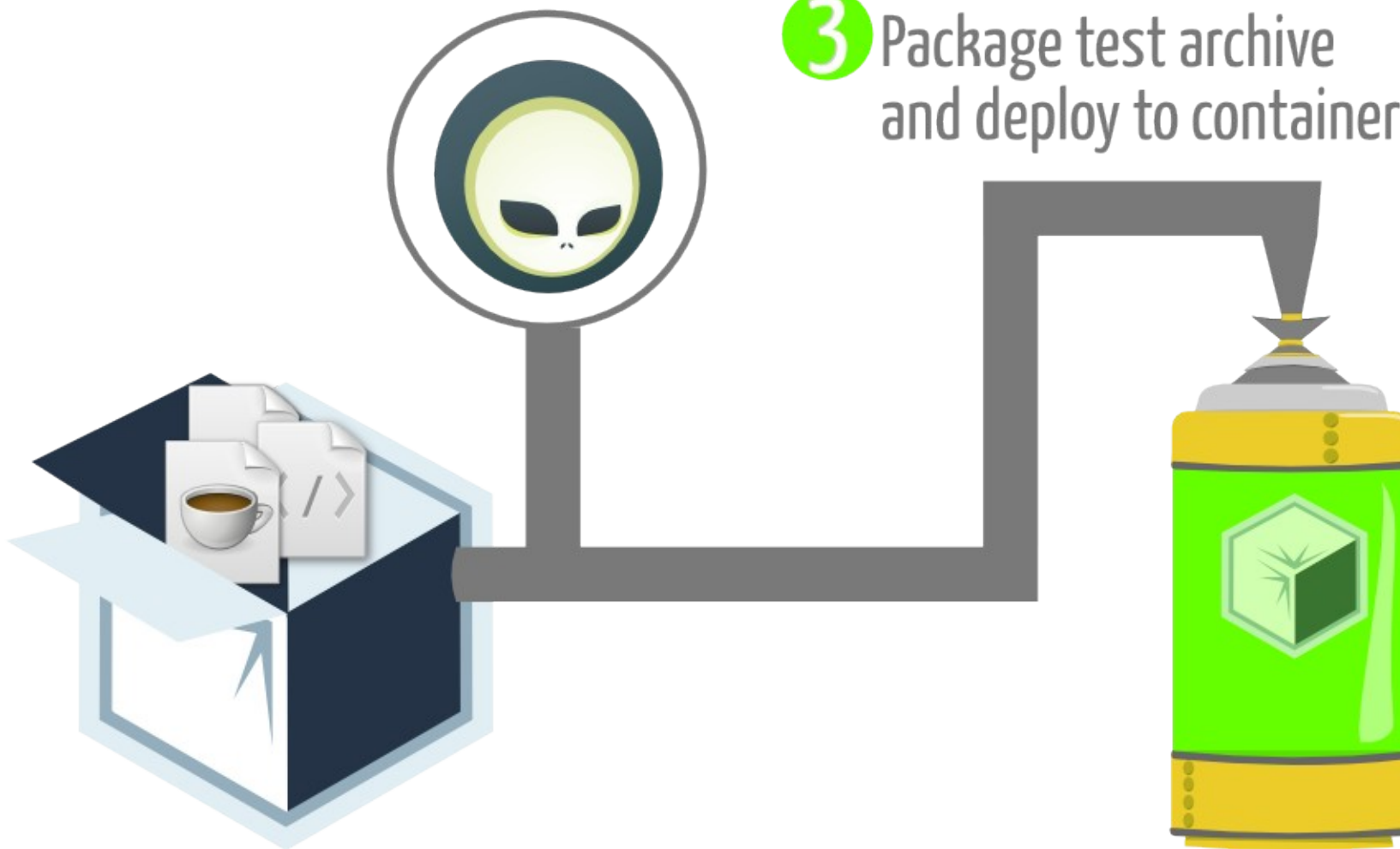






# The Basics

**3** Package test archive  
and deploy to container







# The Basics

3 Package test archive  
and deploy to container

```
ShrinkWrap.create(JavaArchive.class)  
    .addClasses(x)  
    .addPackages(x.z)
```

```
ShrinkWrap.create(WebArchive.class)  
    .addAsLibraries(x)  
    .addAsWebInfResource(x)  
    .setWebXML(z)
```

```
ShrinkWrap.create(EnterpriseArchive.class)  
    .addAsModules(war, jar)  
    .setApplicationXML(x)
```



# The Basics

4 Run test in-container



# The Basics

## 4 Run test in-container



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

    @Deployment
    public static WebArchive deploy() {

        return ShrinkWrap.create(WebArchive.class
            , ArquillianSimple.class.getName() + ".war")
            .addClasses(
                BookService.class,
                Book.class,
                BookBean.class
            )
            .addAsWebInfResource(EmptyAsset.INSTANCE, "beans.xml");
    }

    @Test
    public void test(){

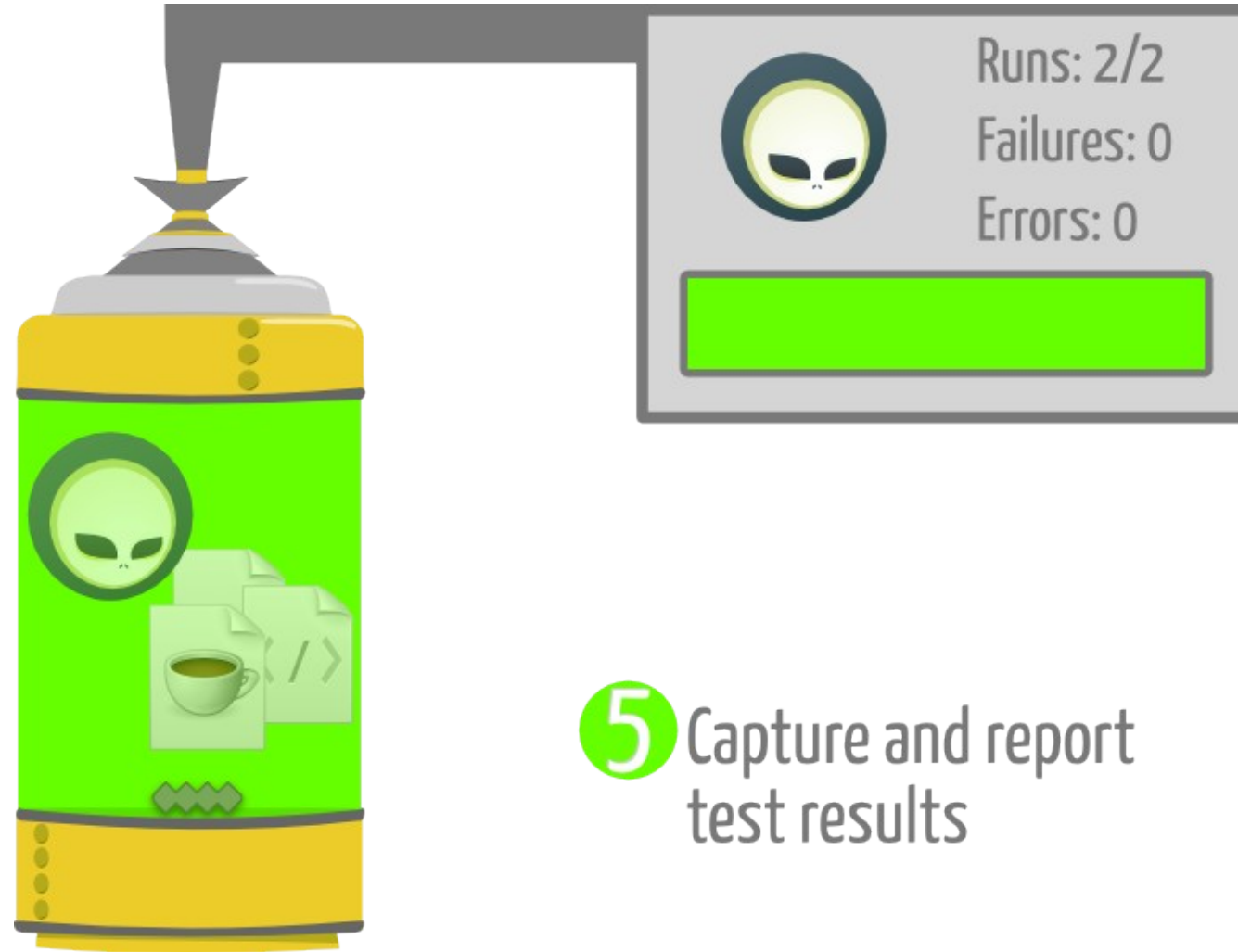
    }

}
```





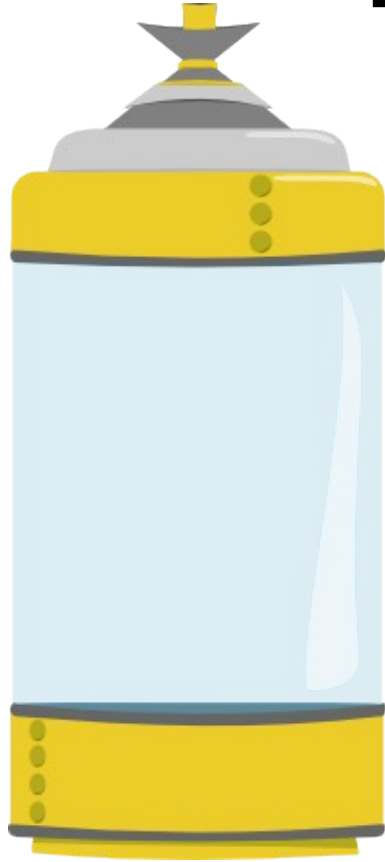
# The Basics



5 Capture and report test results



# The Basics



- 6 Undeploy test archive and stop or disconnect from container



# **@RunWith(Arquillian.class)** **"Production Near" JavaEE** **Testing**

<https://github.com/AndyGee/JAX>

Let's see some code...

