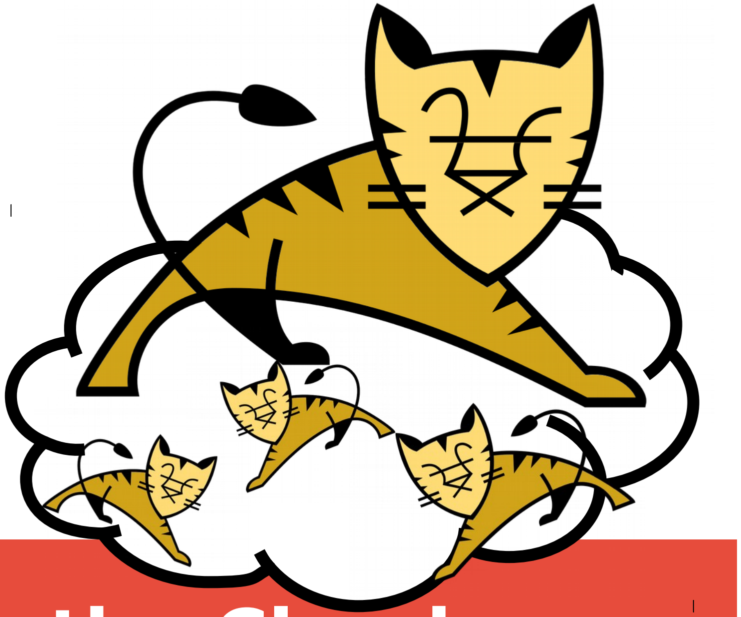
R&D Workshop @ Red Hat



Tomcat Session Replication in the Cloud

Initial presentation

Supervisor: Jean-Frederic Clere

Ismaïl Senhaji, Guillaume Pythoud

Outline

- Goal
- Technology Overview
- Existing Solutions
- Paths to Explore
- Deliverables
- Plan
- Risks

Goal

Extend Tomcat’s session replication to work in a cloud environment

- Create a simple test application using SpringBoot
  - ➔ The app will be distributed and be able to keep session data between nodes.
- Provide documentation (Quick Start Guide)
- Target OpenShift cloud platform

Technology Overview (1)

Tomcat

Open-source Java Servlet Container

Has built-in session replication

But is not cloud-ready

SpringBoot

Packages Java application
+ pre-configured Spring Framework
+ embedded Tomcat Server

Into a ready-to-run, standalone JAR

Technology Overview (2)

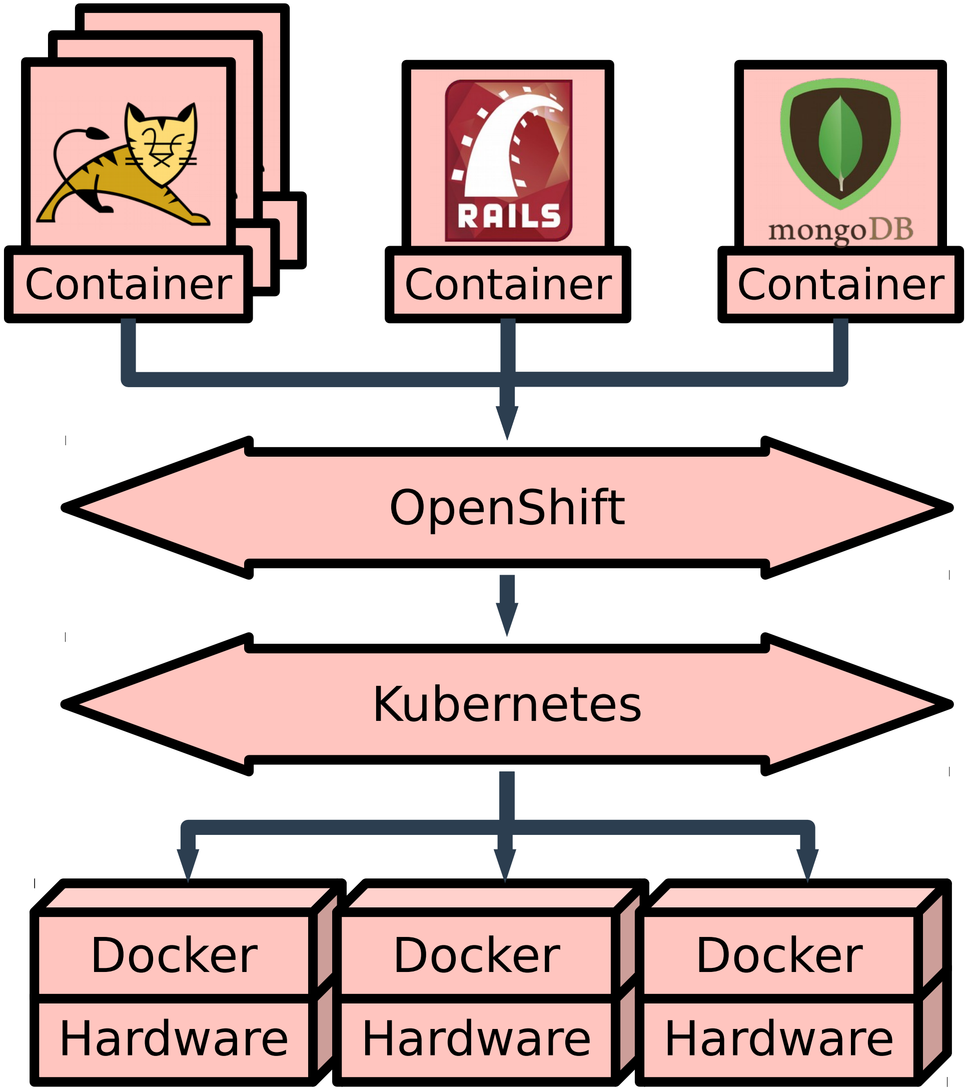
OpenShift

Open source cloud/PaaS suite built on Docker containers and with Kubernetes as cluster manager.
Built by Red Hat.

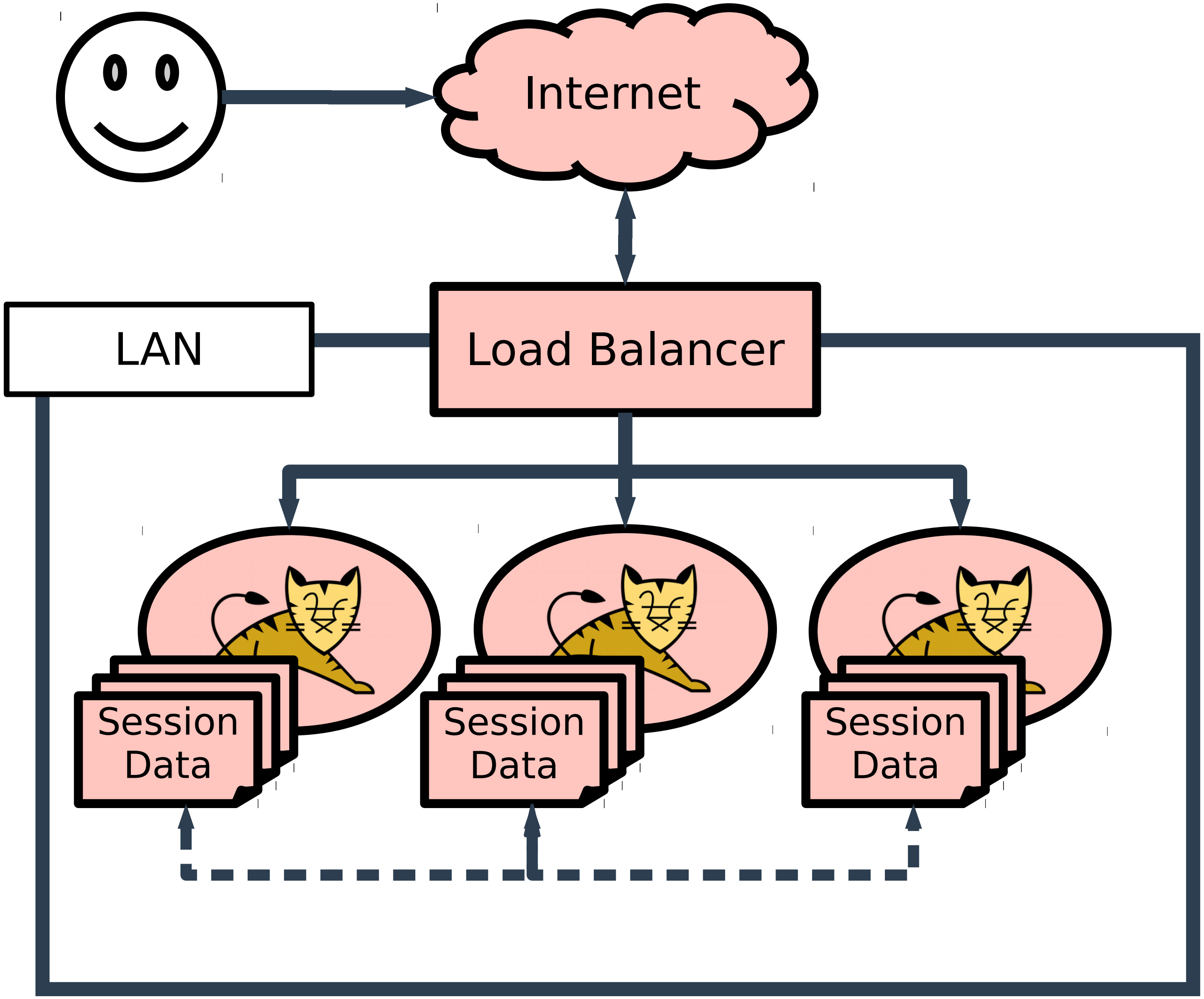
MiniShift

Creates a local OpenShift instance

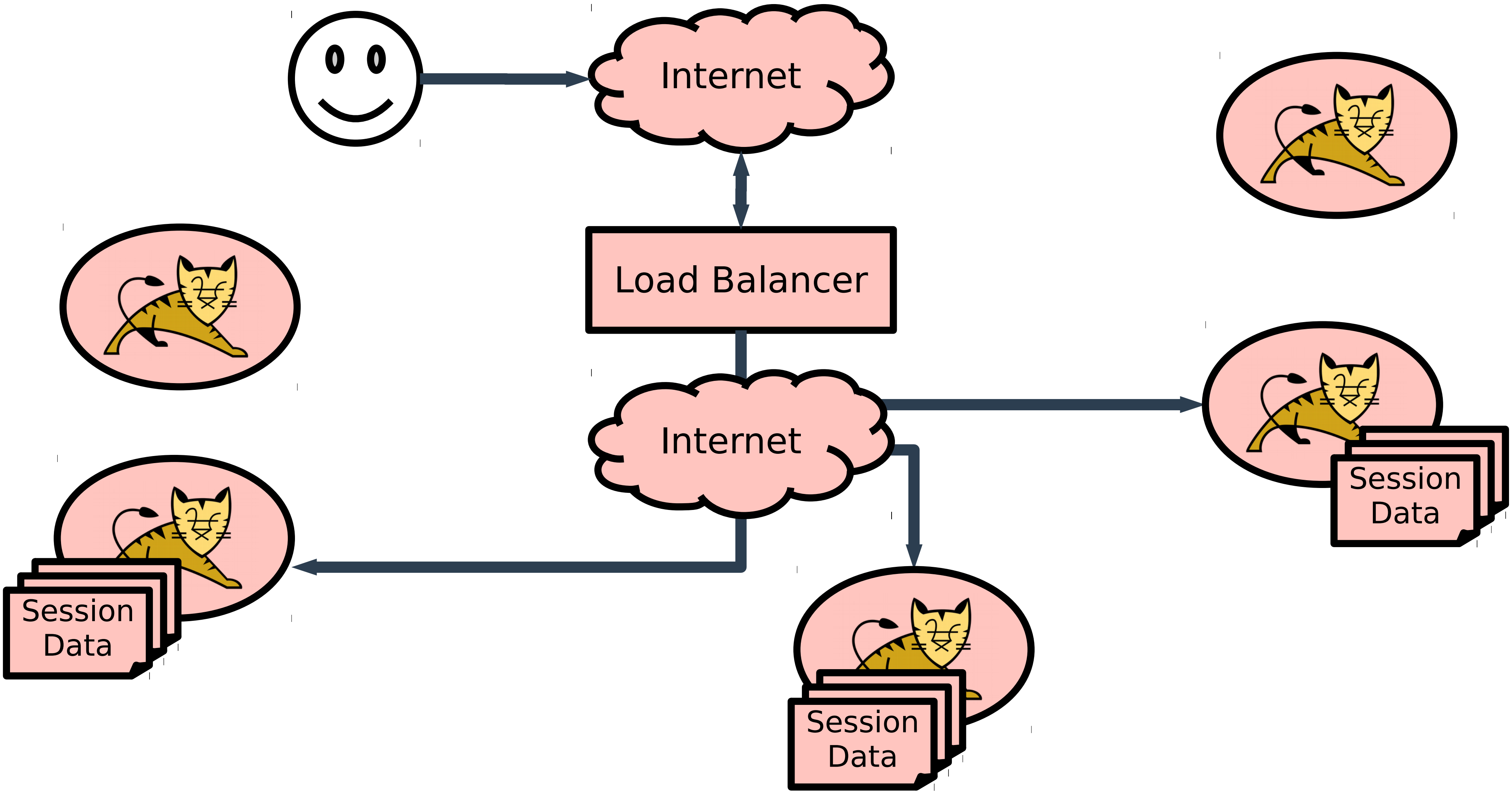
- Useful for testing



Classic Architecture Overview



Cloud Architecture Overview



Existing Solutions (1)

- **Tomcat Session Managers**[1]

1. DeltaManager
   - Replicates (in-memory) session data to all other instances
   - Static: IP addresses of peers hardcoded in Tomcat configuration
   - Dynamic: Peer discovery using multicast
   - Assumes nodes are on secure network
2. PersistanceManager
   - Stores session data in database (uses JDBC)
   - Or filesystem

[1] https://tomcat.apache.org/tomcat-8.5-doc/cluster-howto.html

Existing Solutions (2)

- **External Session Managers**
  - Sessions managed through Spring Framework[1]
    - Session data can be stored on distributed datastore (built-in support for Redis, Mongo, Hazelcast, …)
  - Use Wildfly's session manager[2]
    - Session data is stored in Infinispan / JBoss Data Grid

[1] http://docs.spring.io/spring-session/docs/current/reference/html5/
[2] https://github.com/wildfly-clustering/wildfly-clustering-tomcat

Paths to Explore

- **Plan A: Extend Tomcat’s built-in session management**

  1. DeltaManager:

     ➔ Load list of peers from a shared datastore

  2. PersistanceManager:

     ➔ Store session file in cloud storage

- **Plan B: Use an external session manager**
  - Spring / Wildfly

Deliverables

- **Small Java Web App for session replication testing**
- **App runing in a cloud (demo)**
  - ➔ Testing on Minishift during development
  - ➔ OpenShift, private cloud running on a Raspberry Pi Cluster
- **Documentation and Quick Start Guide**
  - With boilerplate code

Plan

Until midterm

3rd – 10th week

- Read about Tomcat, clustering, Openshift,...
- Write the testing app
- Run the app on a local Tomcat Cluster
- Install and configure Minishift
- Test and choose between different replication methods of Plan A

Midterm to final

11th – 14th week

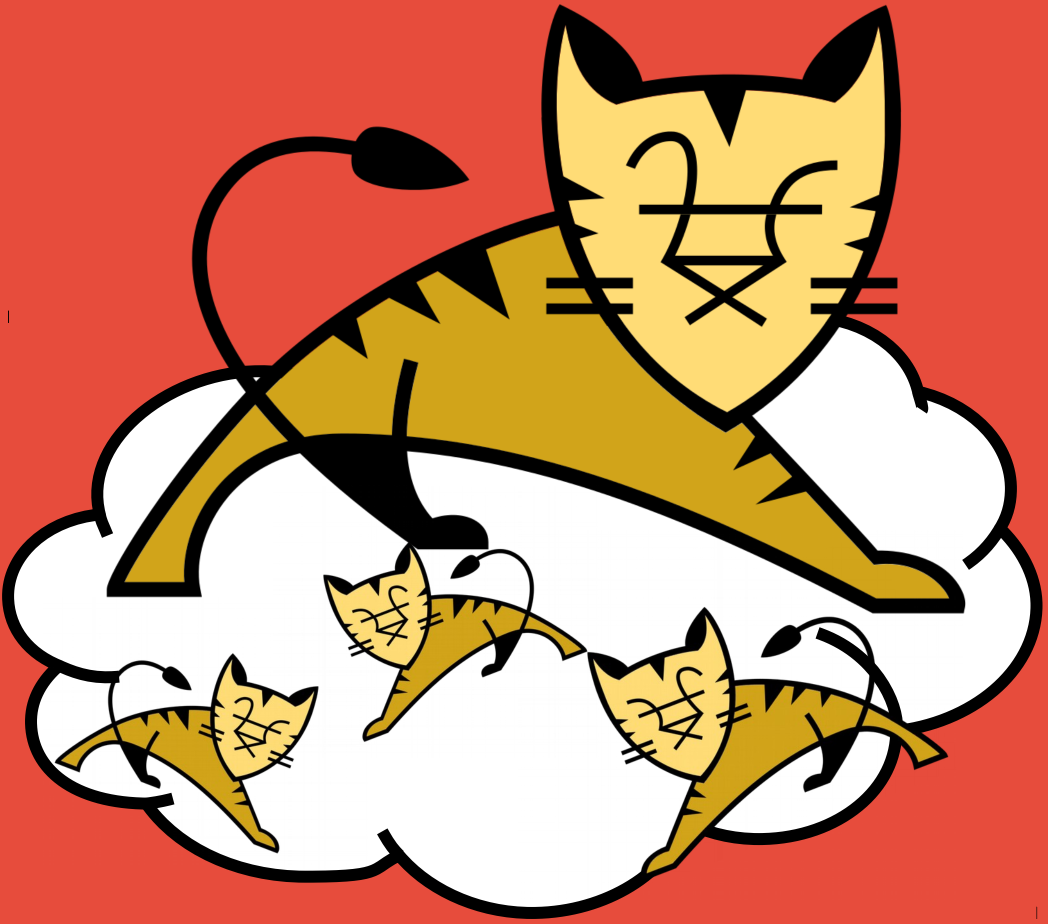
- (Plan B if Plan A failed)
- Install Raspberry Pi Cluster
- Deploy app
- Write documentation & report

https://github.com/iSma/tomcat-in-the-cloud/wiki/Workplan

Risks

- **Plan A is too difficult/impossible**
  - → Fallback solution (Plan B) is known to work
- **Ecosystem (Tomcat, OpenShift, …) is complex**
  - But answers should be somewhere:
  - → Project documentation
  - → Mailing lists
  - → Forums / IRC
  - → Jean-Frédéric Clere’s contacts
  - → Source code

Thank you for your attention!



Ismaïl Senhaji, Guillaume Pythoud