

R&D Workshop @ Red Hat



Tomcat Session Replication in the Cloud

Final presentation

Supervisor: Jean-Frederic Clere

Ismail Senhaji, Guillaume Pythoud



redhat.

unine
UNIVERSITÉ DE
NEUCHÂTEL

Outline

- **Reminder**
- **What has been done**
- **Solution**
- **Demo**
- **Issues and future improvements**

Reminder: Goal

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

- Create a simple test application
- Provide documentation
- Use OpenShift as cloud platform

Reminder: Technology



Tomcat

Open-source Java Servlet Container

Has built-in session replication

But is not cloud-ready

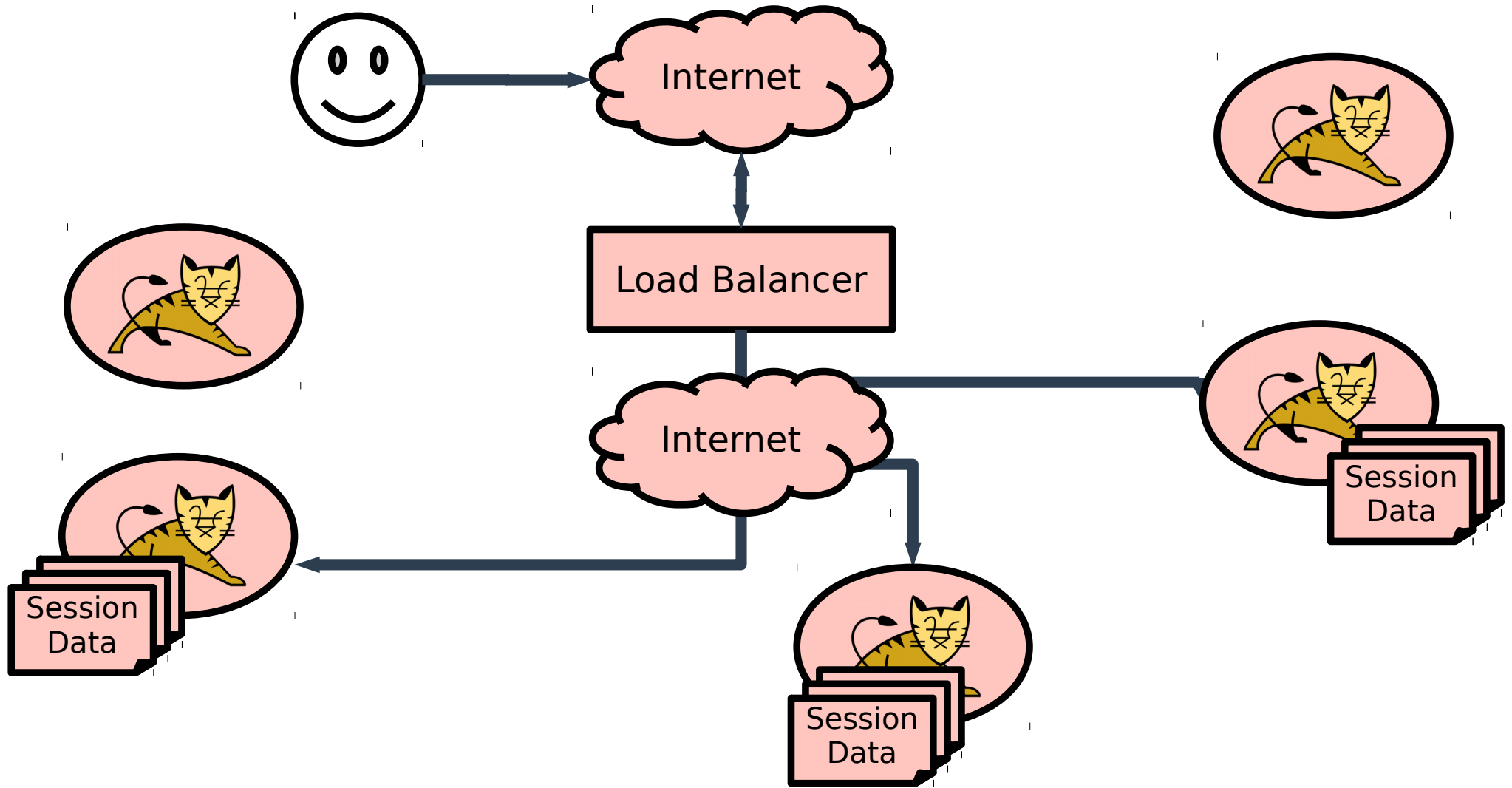


OpenShift

Red Hat Open source cloud/PaaS suite

Built on Docker containers and with Kubernetes as cluster manager.

Reminder: Cloud Architecture



What has been done

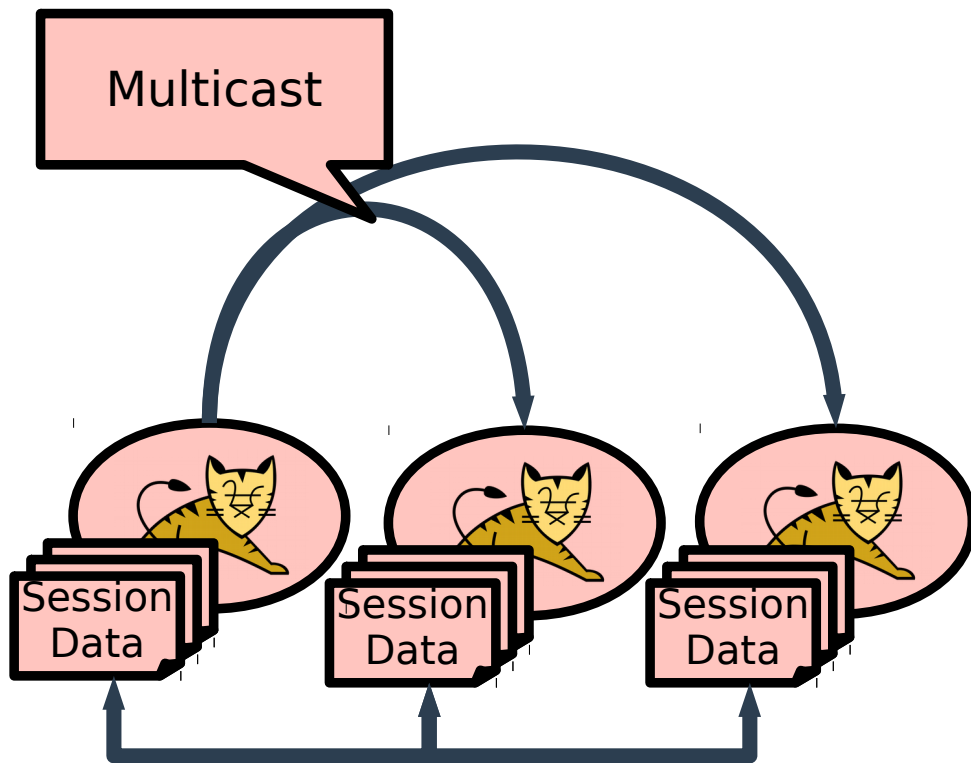
- **First part of the project: discovery of technologies**
- **Exploration of possible solutions**
- **Implementation of test applications:**
 - Based on: tomcat-embedded, spring-boot, external session manager, Infinispan,...
- ➔ **Finally Infinispan shows us a possible solution**

Infinispan and KubePing

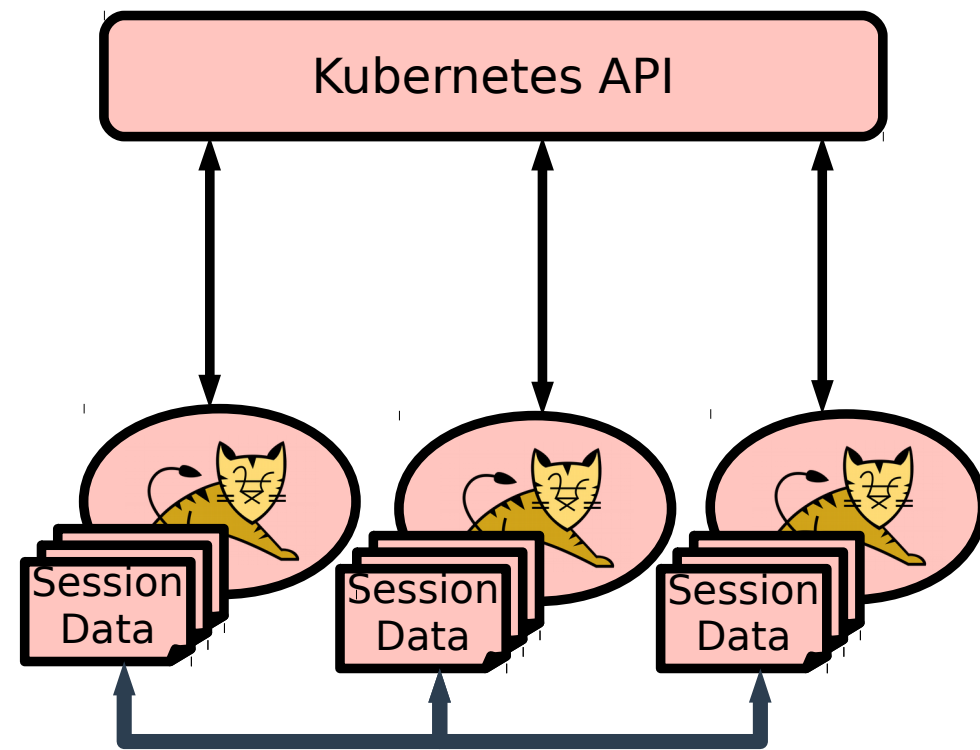
- **Infinispan has a replicated mode (data replicated to all instances)**
 - Can be compared to how DeltaManager in Tomcat works
 - By default, peer discovery through multicast
 - **KubePing** project adds peer discovery for Kubernetes
 - KubePing uses the Kubernetes downward API
- **It does what we wanted to achieve**
 - We studied KubePing's architecture
 - Implemented a similar solution for Tomcat

Solution

Tomcat built-in solution
Peer discovery through multicast
heartbeat messages
Does not work in a cloud environment



Our solution
Peer discovery through Kubernetes
Downward API
Works in OpenShift



Kubernetes API

Tools for managing a
Kubernetes cluster

Accessible from the pods within
the cluster

*GET /api/v1/namespaces/tomcat-
in-the-cloud/pods*

→ Return a JSON representation of all the
pods in the cluster

```
kind: PodList
apiVersion: v1
▼ metadata:
  selfLink: /api/v1/namespaces/tomcat-in-the-cloud/pods
  resourceVersion: 7602
▼ items:
  ▼ 0:
    ▼ metadata:
      name: tomcat-in-the-cloud-1-5xbwm
      generateName: tomcat-in-the-cloud-1-
      namespace: tomcat-in-the-cloud
      ▶ selfLink: /api/v1/namespaces/tomcat-in-the-cloud-1-5xbwm
      uid: ecac3cff-5361-11e7-9a95-3a314e9cf749
      resourceVersion: 7568
      creationTimestamp: 2017-06-17T13:36:10Z
      ▶ labels: Object
      ▶ annotations: Object
    ▶ spec: Object
    ▼ status:
      phase: Running
      ▶ conditions: [3]
      hostIP: 192.168.42.74
      podIP: 172.17.0.3
      startTime: 2017-06-17T13:36:10Z
      ▶ containerStatuses: [1]
    ▶ 1: Object
    ▶ 2: Object
```

Architecture

DynamicMembershipService

RefreshThread

- Call memberProvider.getMembers()
- Filter out already known Member
- Inform listeners of new/dead members

MemberProvider

- init(Properties)
- getMembers(): List<Member>

KubernetesMemberProvider

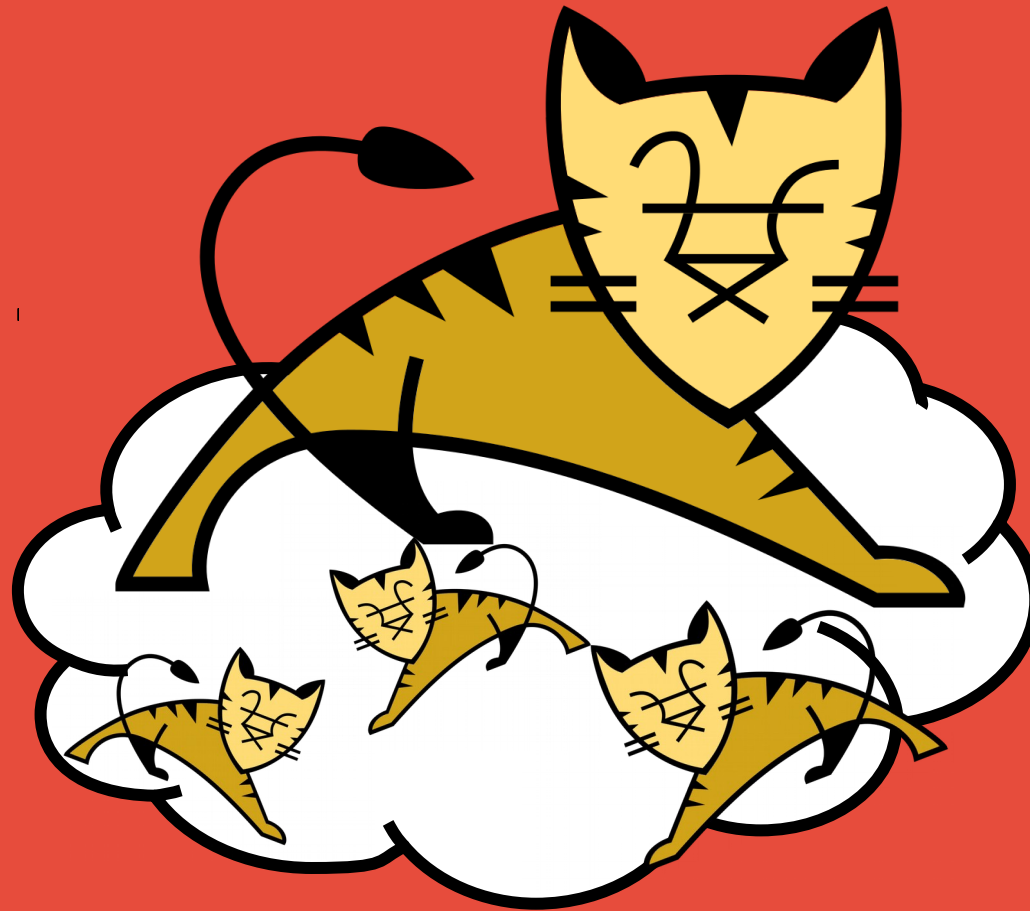
- init():
 - Get URL, cert, ... from environment variables
 - Set startTime
- getMembers():
 - Call api to get pods
 - Filter active pods
 - Compute aliveTime

DEMO

Issues and future improvement

- **Session data is shared within all the applications of a project/namespaces**
 - Performance/security issues
 - Solution: use *labels*
- **Functional in Openshift, but not 100% sure within a pure Kubernetes cluster**
 - Needs testing
- **Needs better error handling**
- **Better tests**
 - Automated scripts
 - Fault injection

Thank you for your attention!



Ismail Senhaji, Guillaume Pythoud