

Auto-scaling Axis2 Web Services on Amazon EC2

By
Afkham Azeez (azeez@apache.org)
WSO2 Inc.



Overview

- The Problem
- A Solution
- Some Concepts
- Design & Implementation Details



The Problem

- Fault tolerance, high availability & scalability are essential prerequisites for any enterprise application deployment
- One of the major concerns is avoiding single points of failure
- There is a high cost associated with achieving high availability & scalability
- **Need to achieve high availability & scalability at an optimum cost**



Solutions

- Traditional solution
 - Buying safety-net capacity
- Better solution
 - Scale-up the system when the load increases
 - Scale-down the system when the load decreases
 - Should not have idling nodes
 - Pay only for the actual computing power & bandwidth utilized



Project Objective

- Building a framework which will auto-scale the number of Axis2 nodes on Amazon EC2, depending on the load



Apache Axis2

- Axis2 is a middleware platform which enables hosting of Web service applications and supports some of the major Web services standards
- Can host Web services written in Java as well as various scripting languages
- Can be deployed in a clustered configuration
- Uses Apache Tribes for clustering
- Axis2 clustering has been adopted to work on EC2



Apache Synapse

- Apache Synapse is designed to be a simple, lightweight and high performance ESB
- Supports load balancing with or without failover
- Supports static & dynamic load balancing
- Uses Apache Axis2

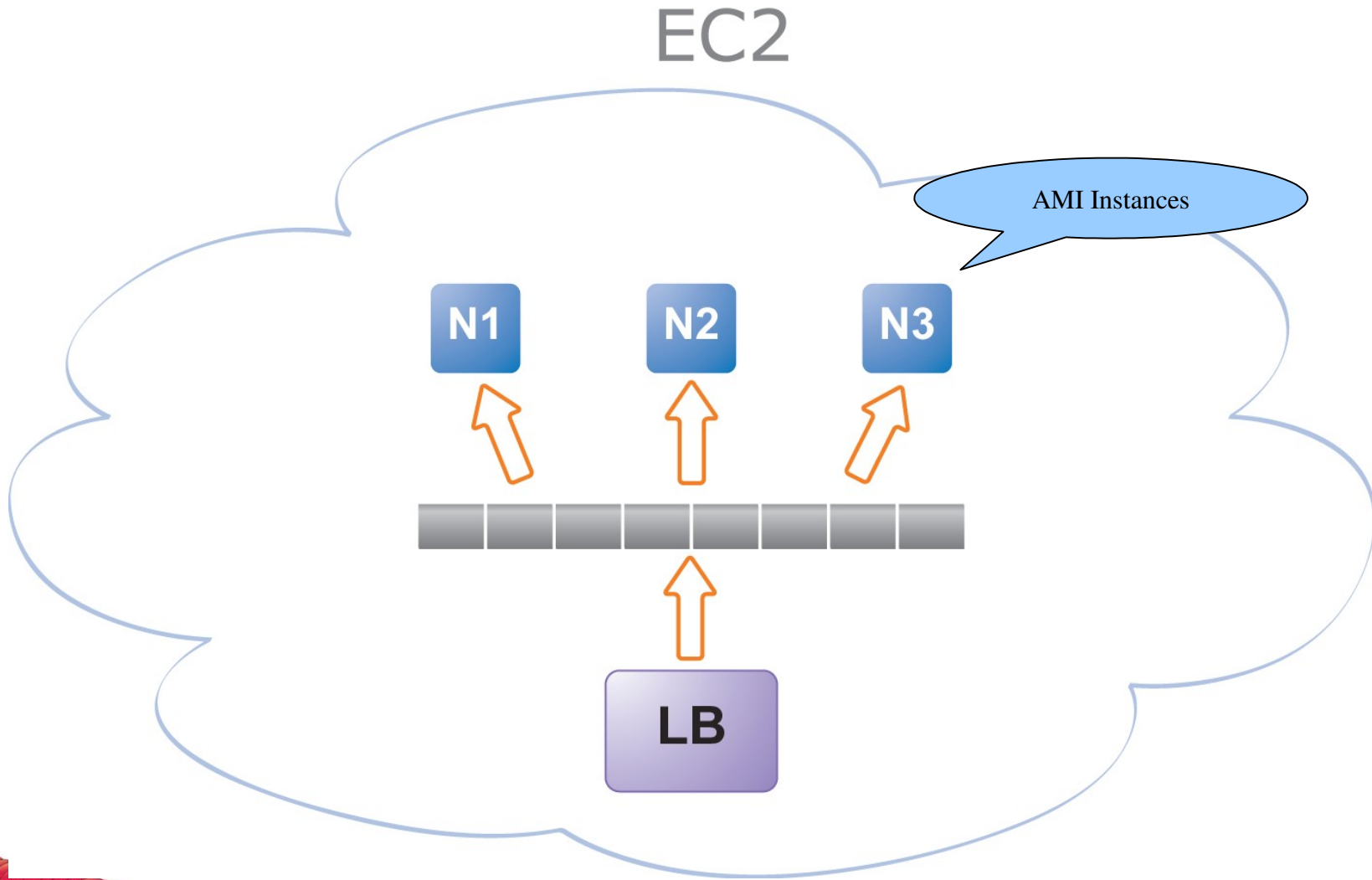


Apache Tribes

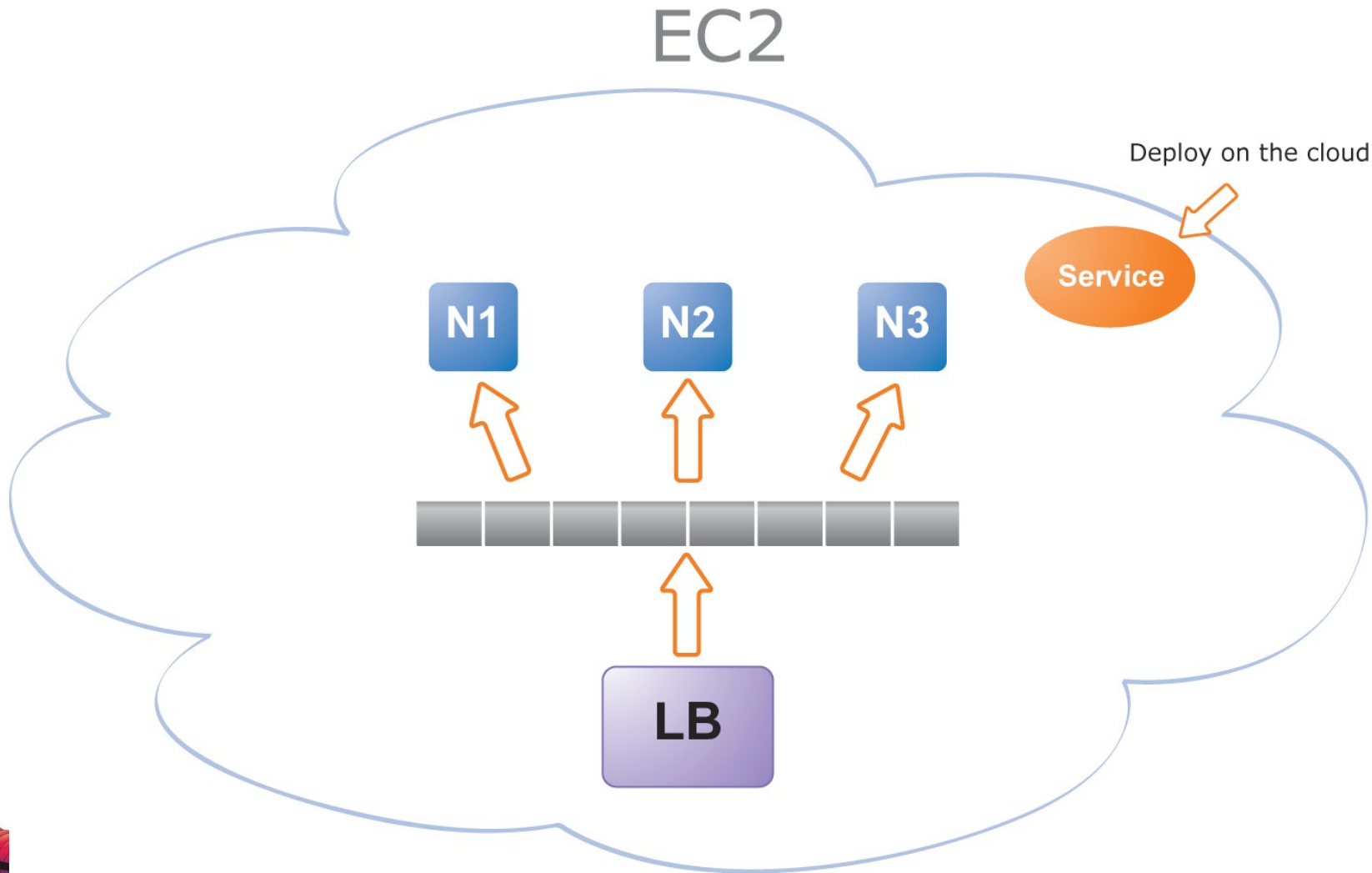
- A messaging framework with group communication abilities
- Allows you to send and receive messages over a network, it also allows for dynamic discovery of other nodes in the network.
- Used by Apache Tomcat & Apache Axis2



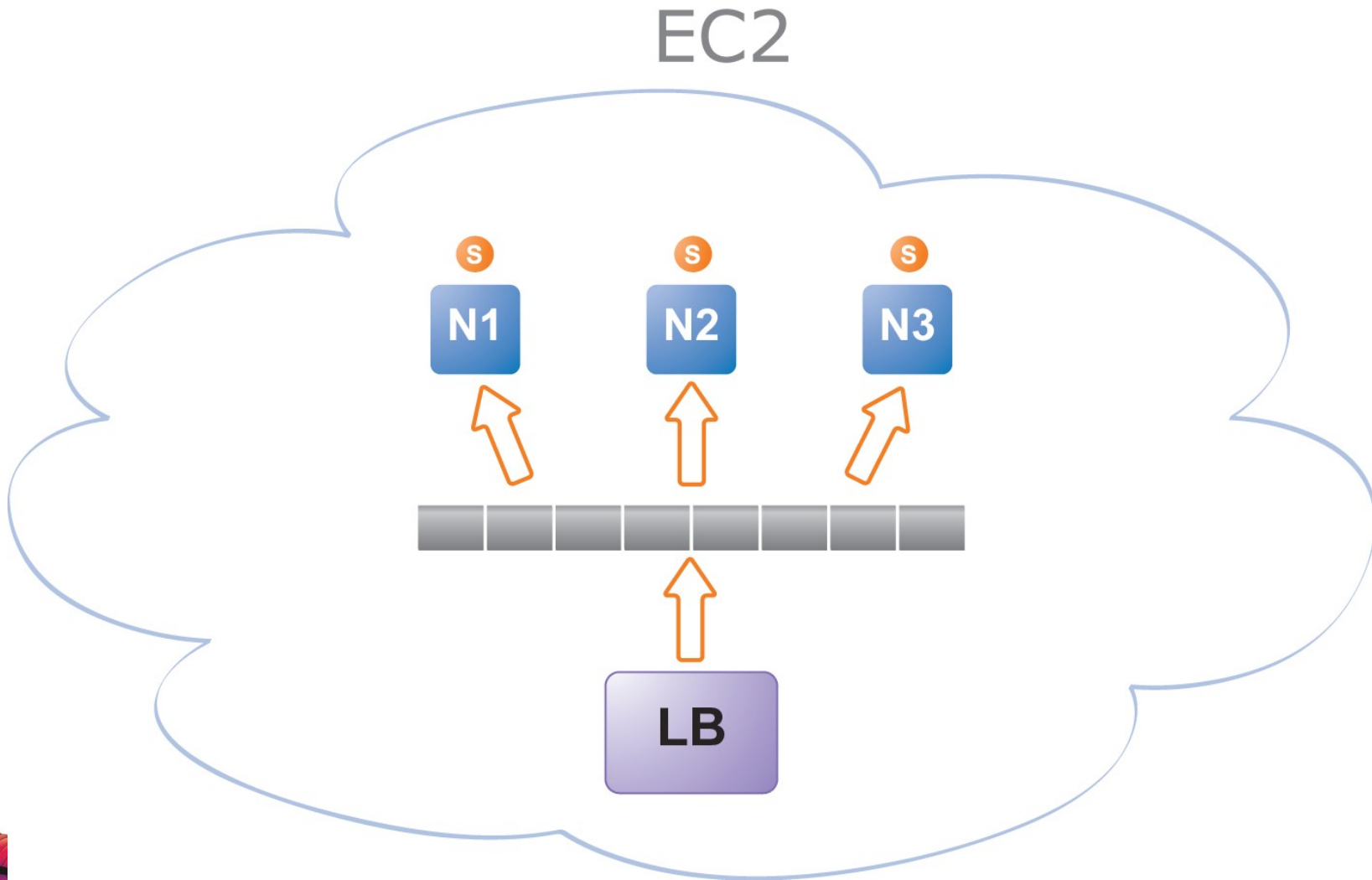
Deploying a Service on the Cloud



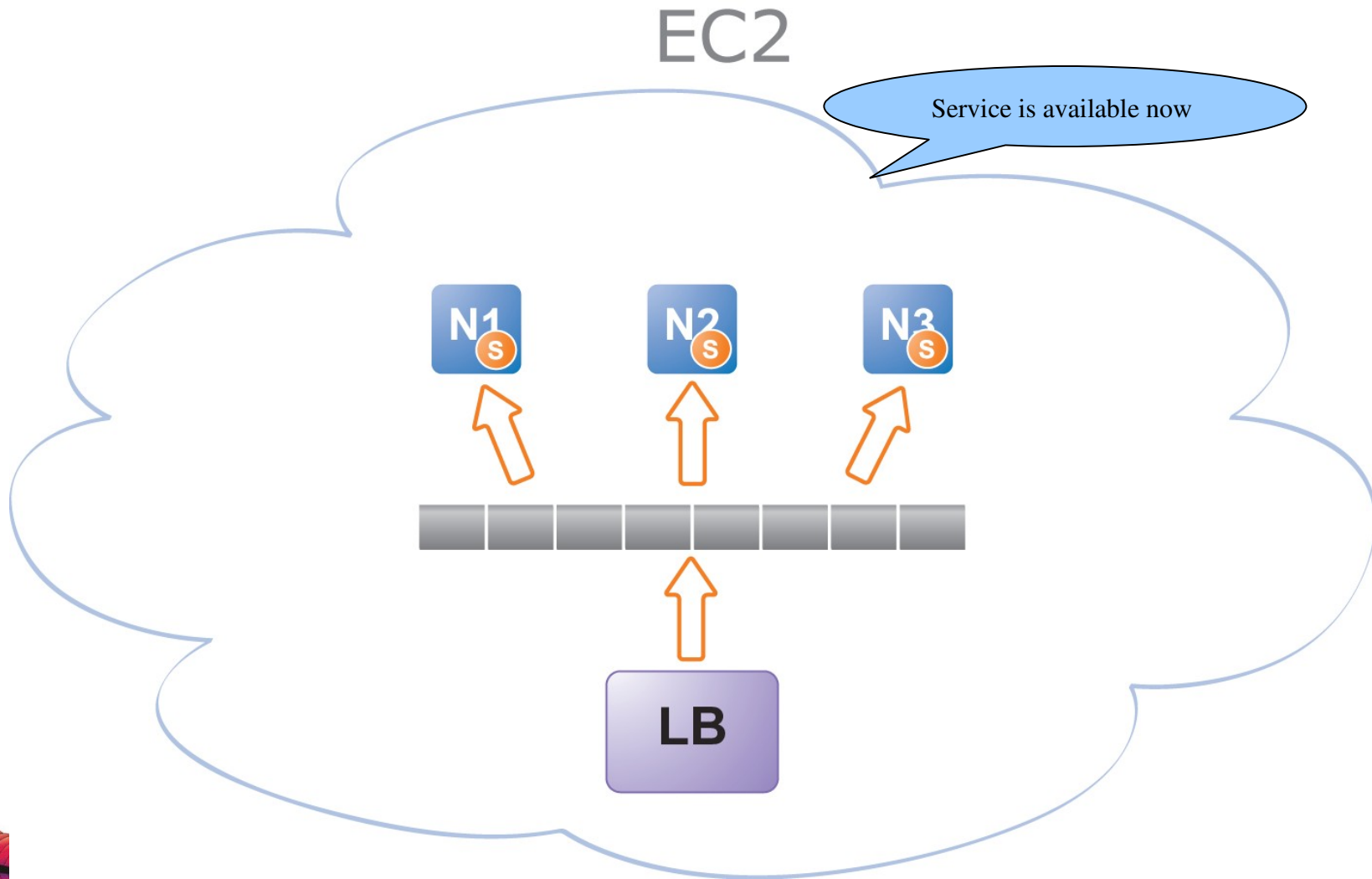
Deploying a Service on the Cloud



Deploying a Service on the Cloud

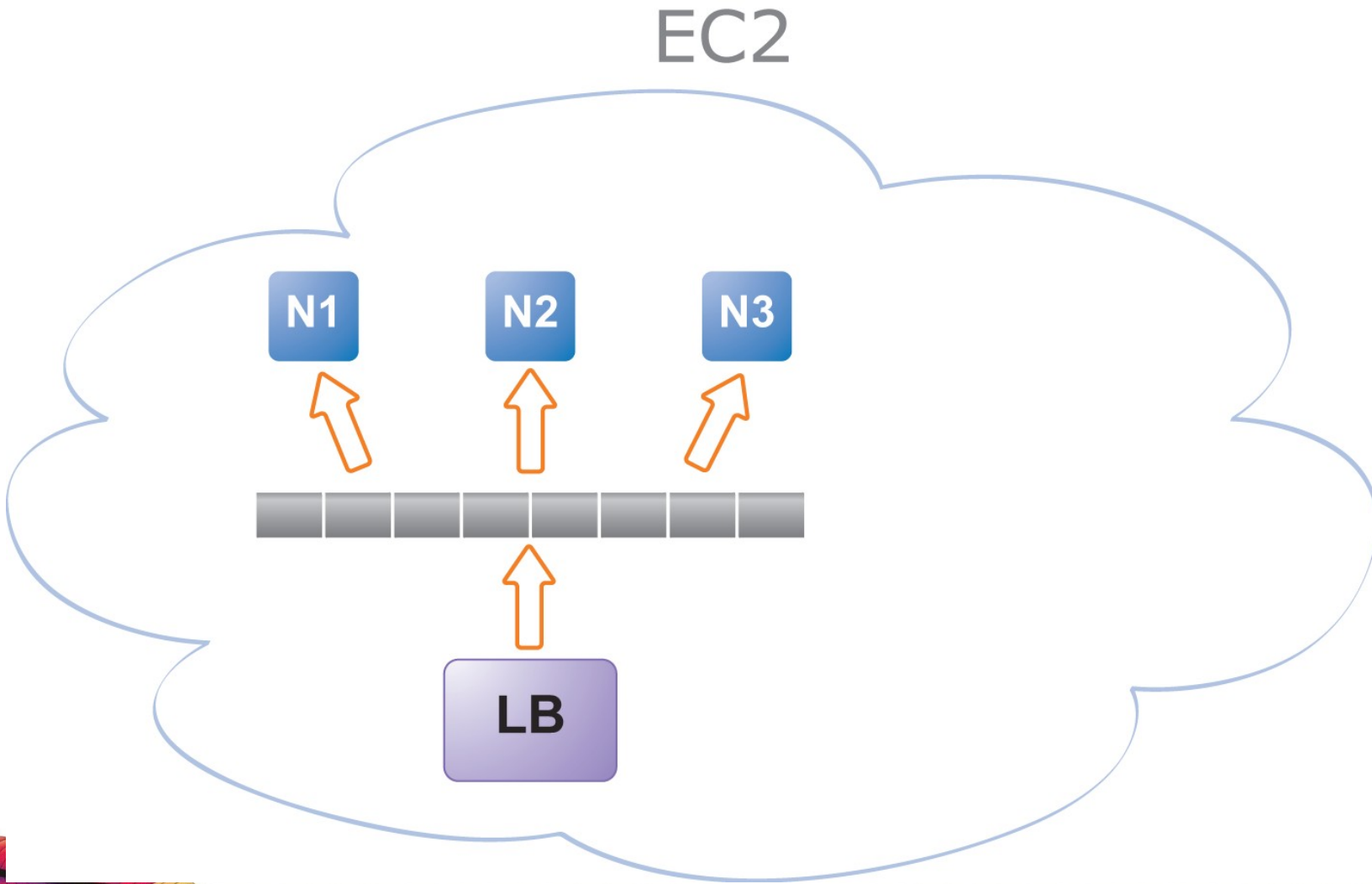


Deploying a Service on the Cloud



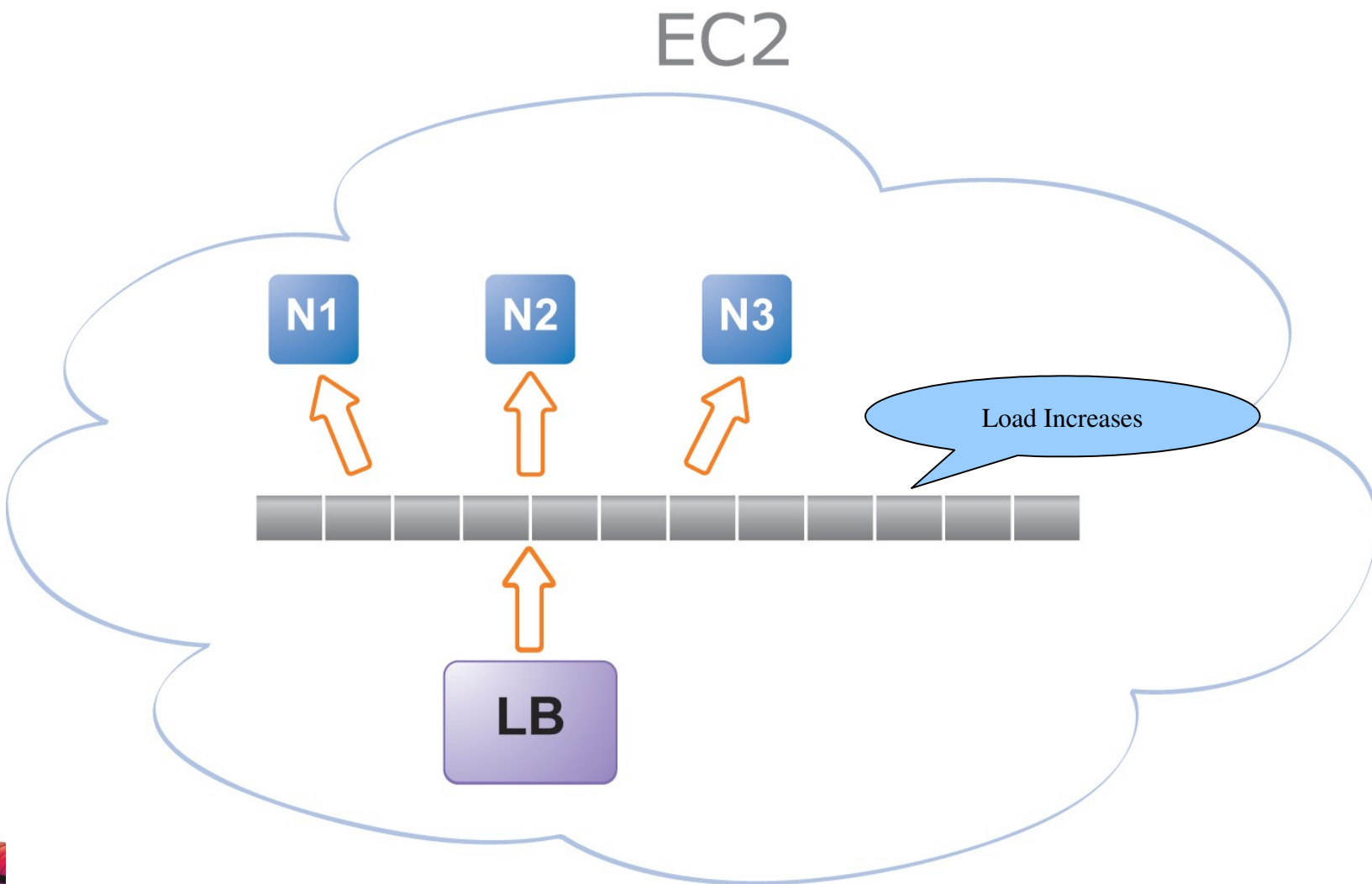


Auto-scaling



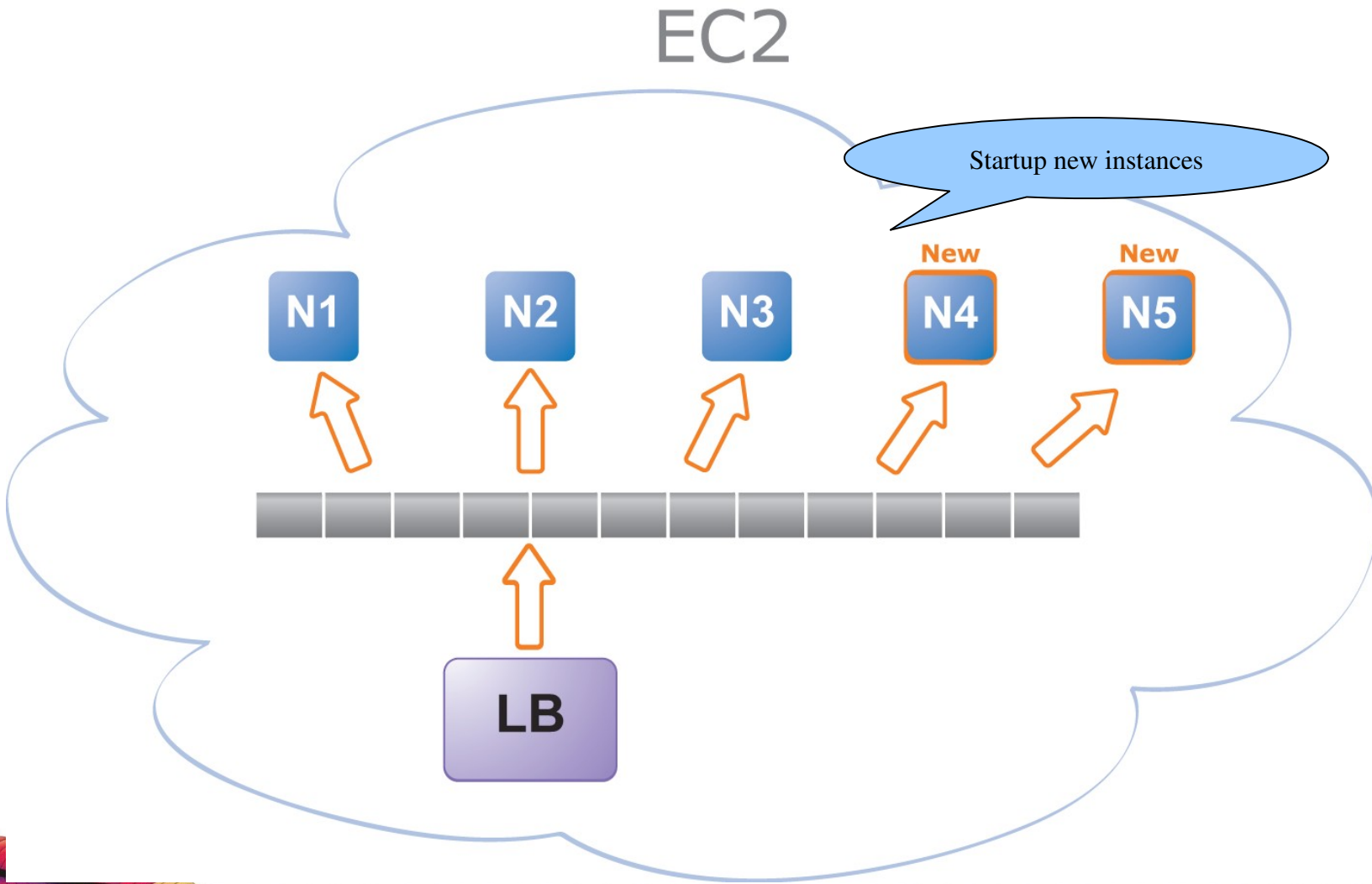


Auto-scaling



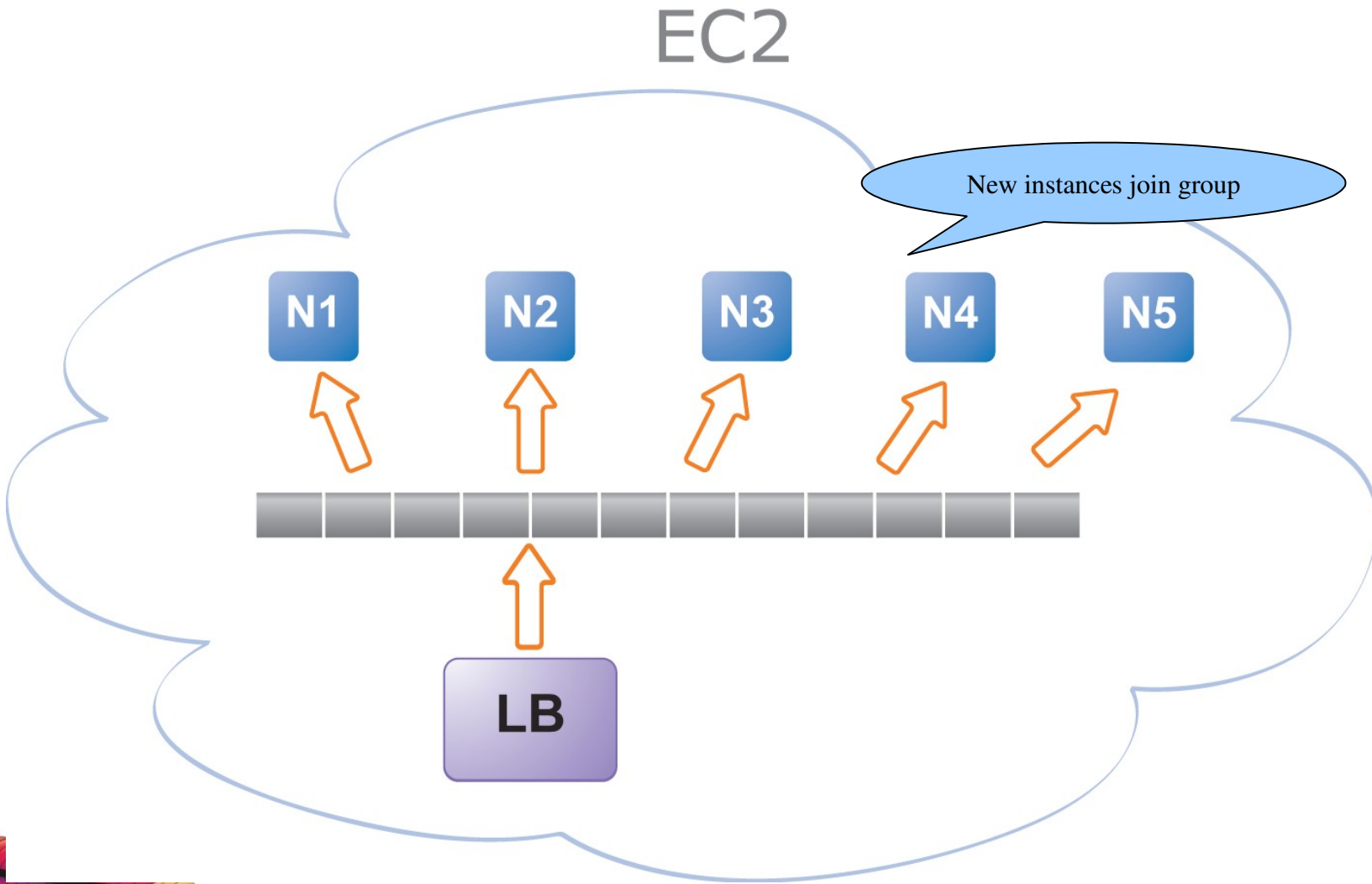


Auto-scaling



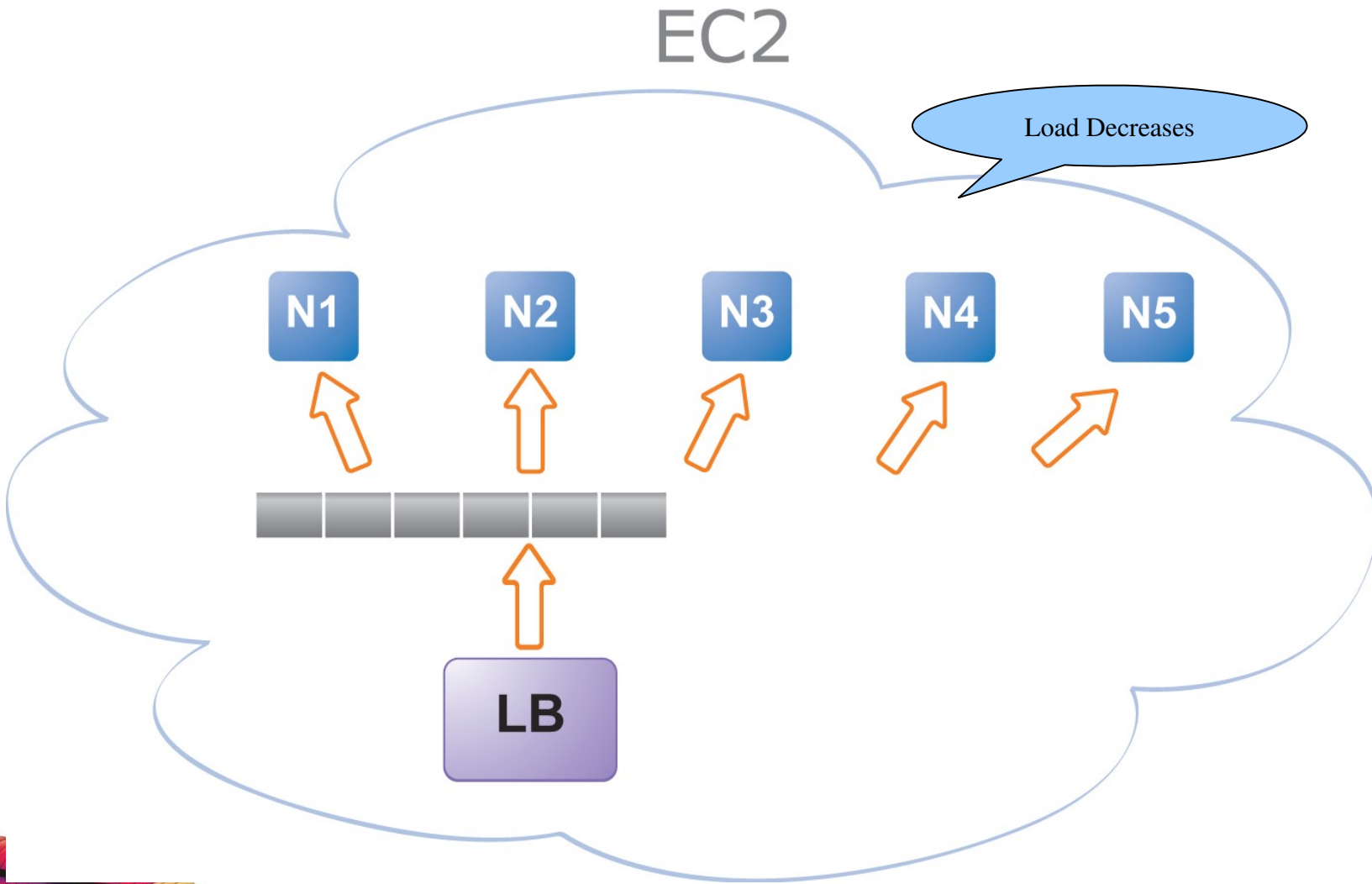


Auto-scaling



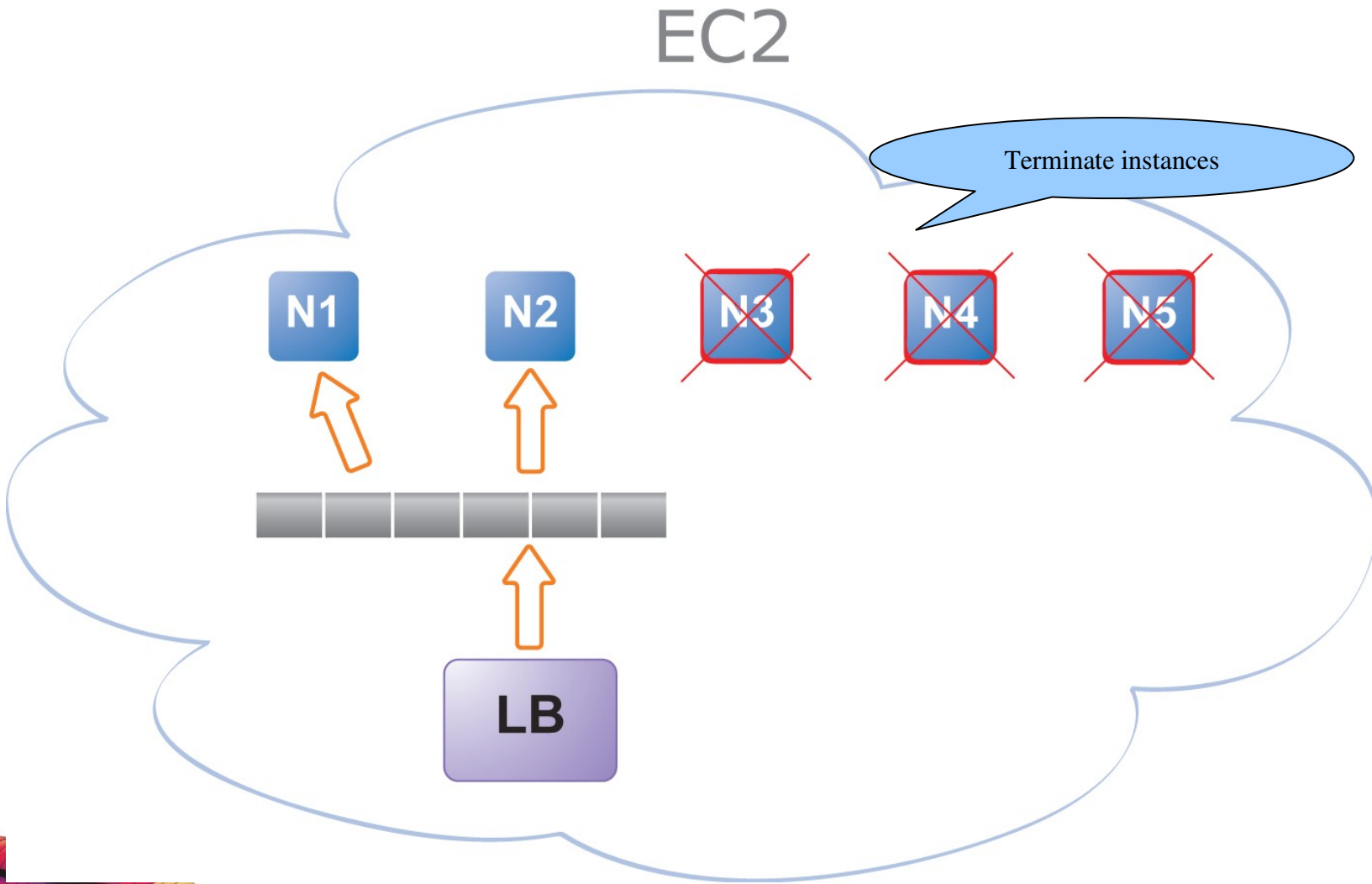


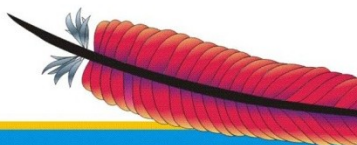
Auto-scaling



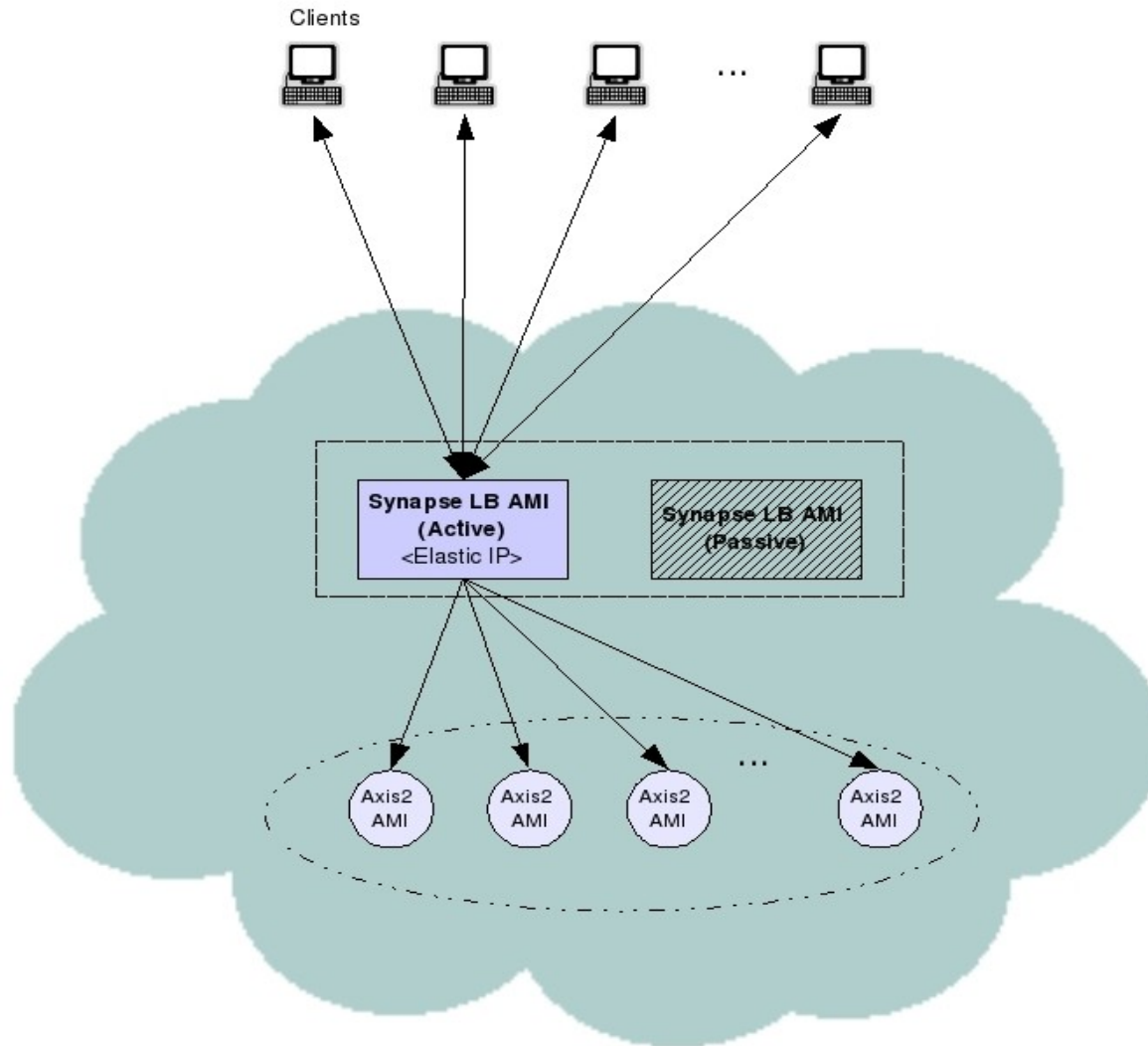


Auto-scaling

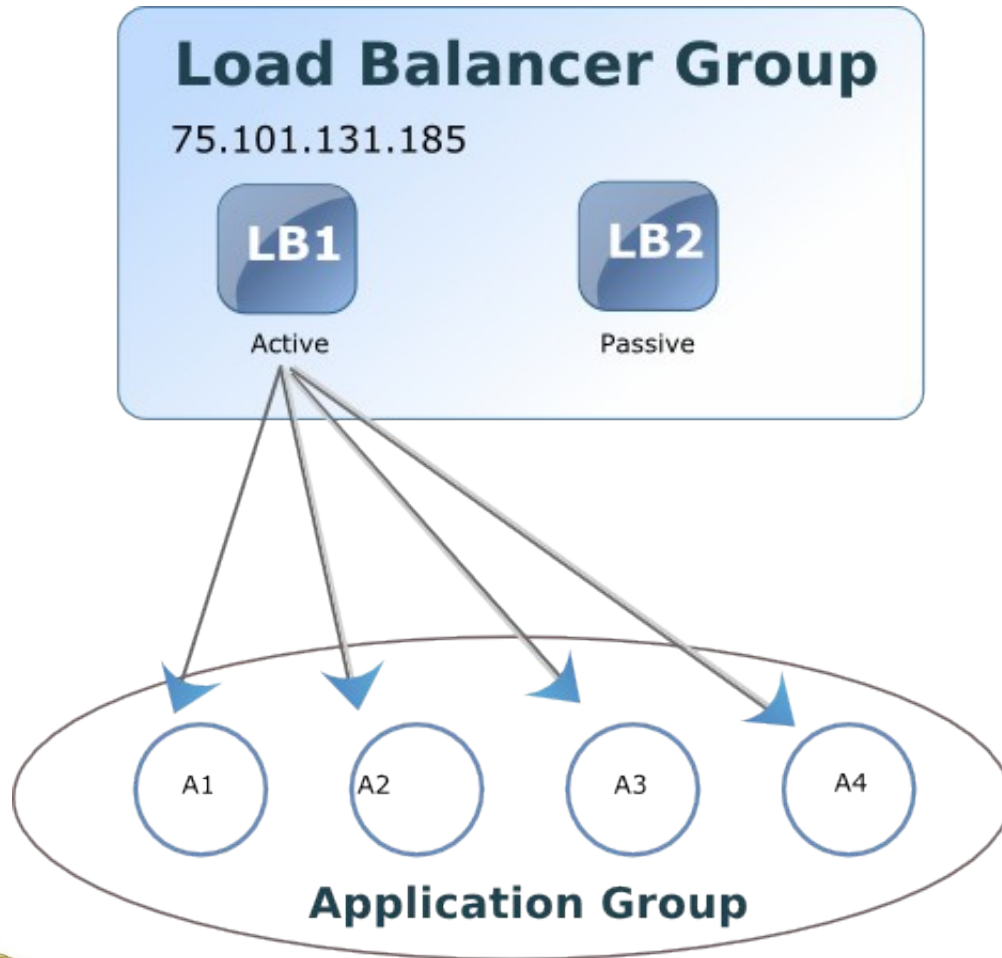




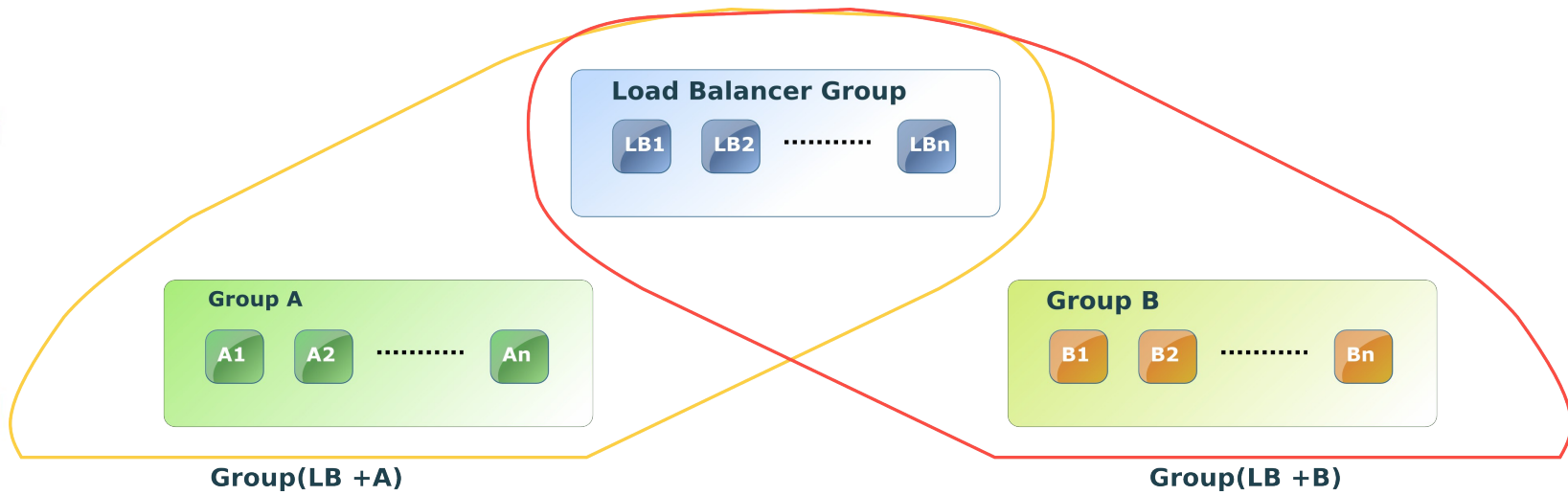
Deployment Architecture



Membership Aware Dynamic Load Balancing



Membership Aware Dynamic Load Balancing

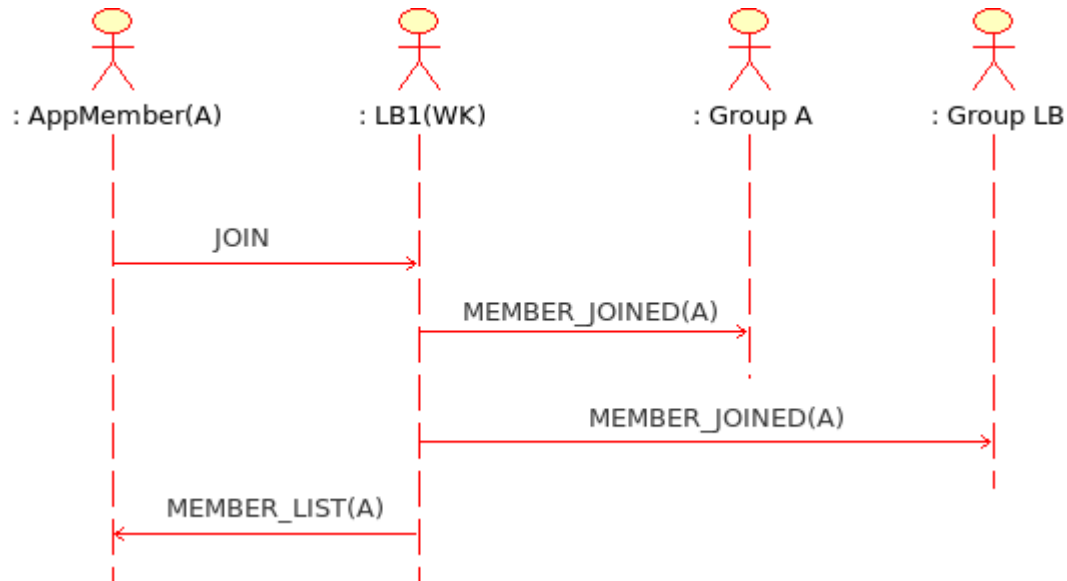


Membership Schemes

- Static
- Dynamic
- Hybrid (WKA based)



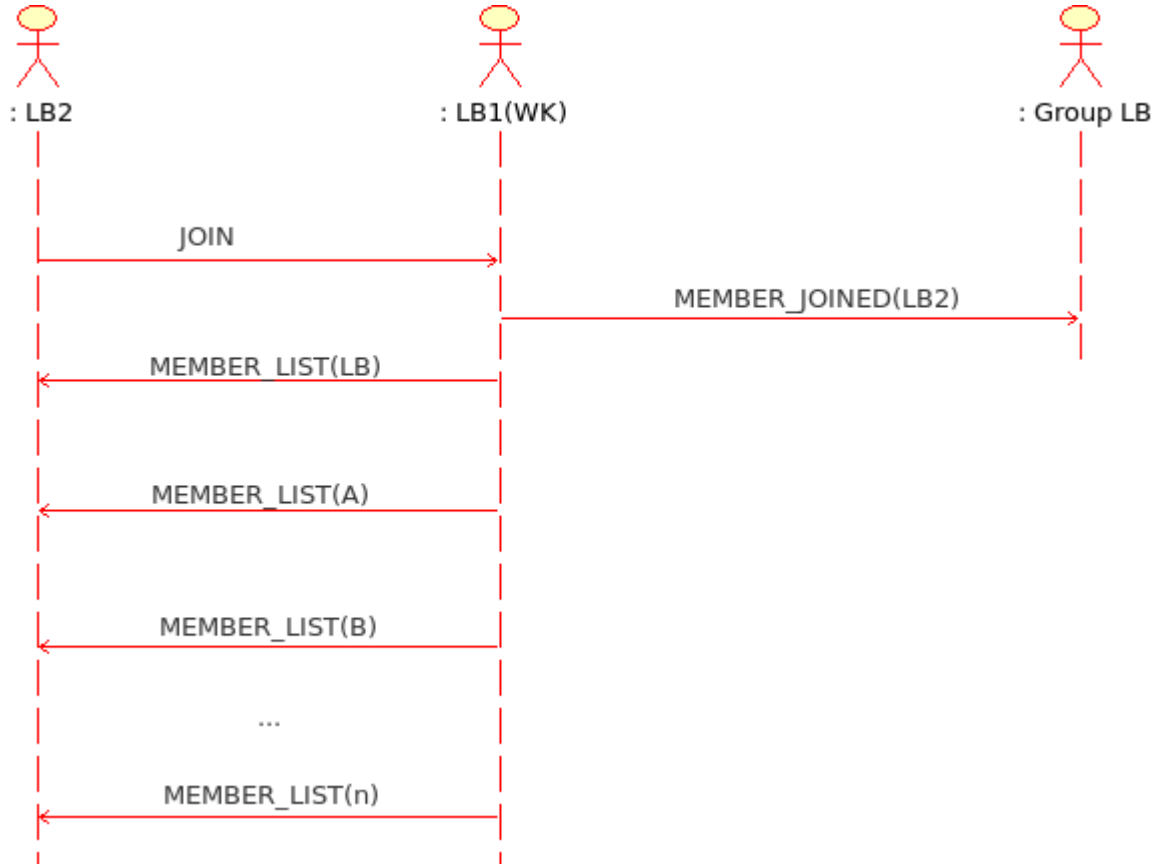
WKA Based Membership (1/3)



Application member joins. The load balancer is also a well-known member

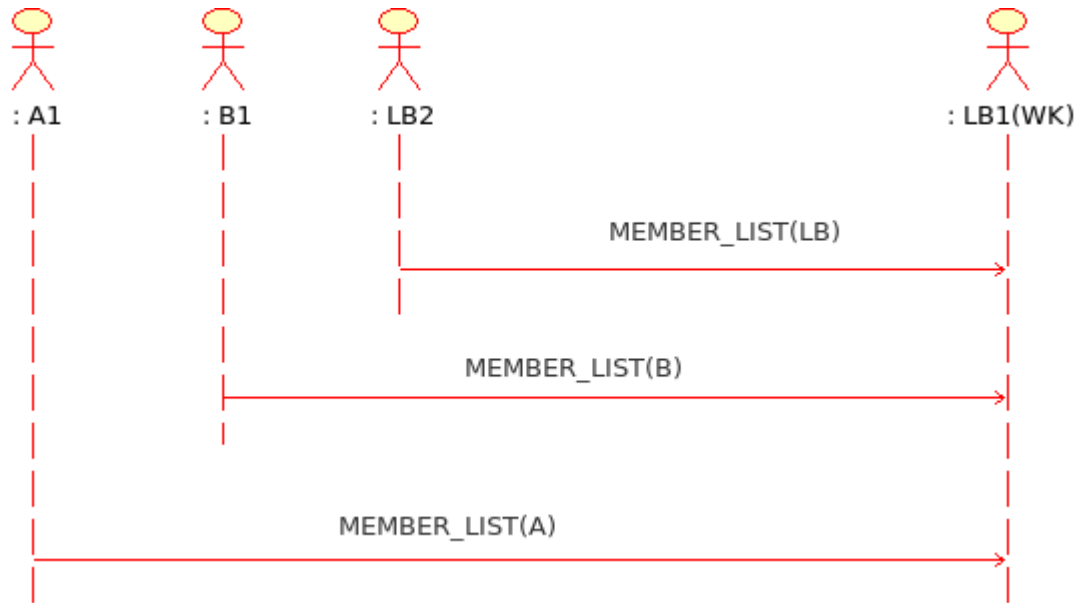


WKA Based Membership (2/3)



A non-WK load balancer joins

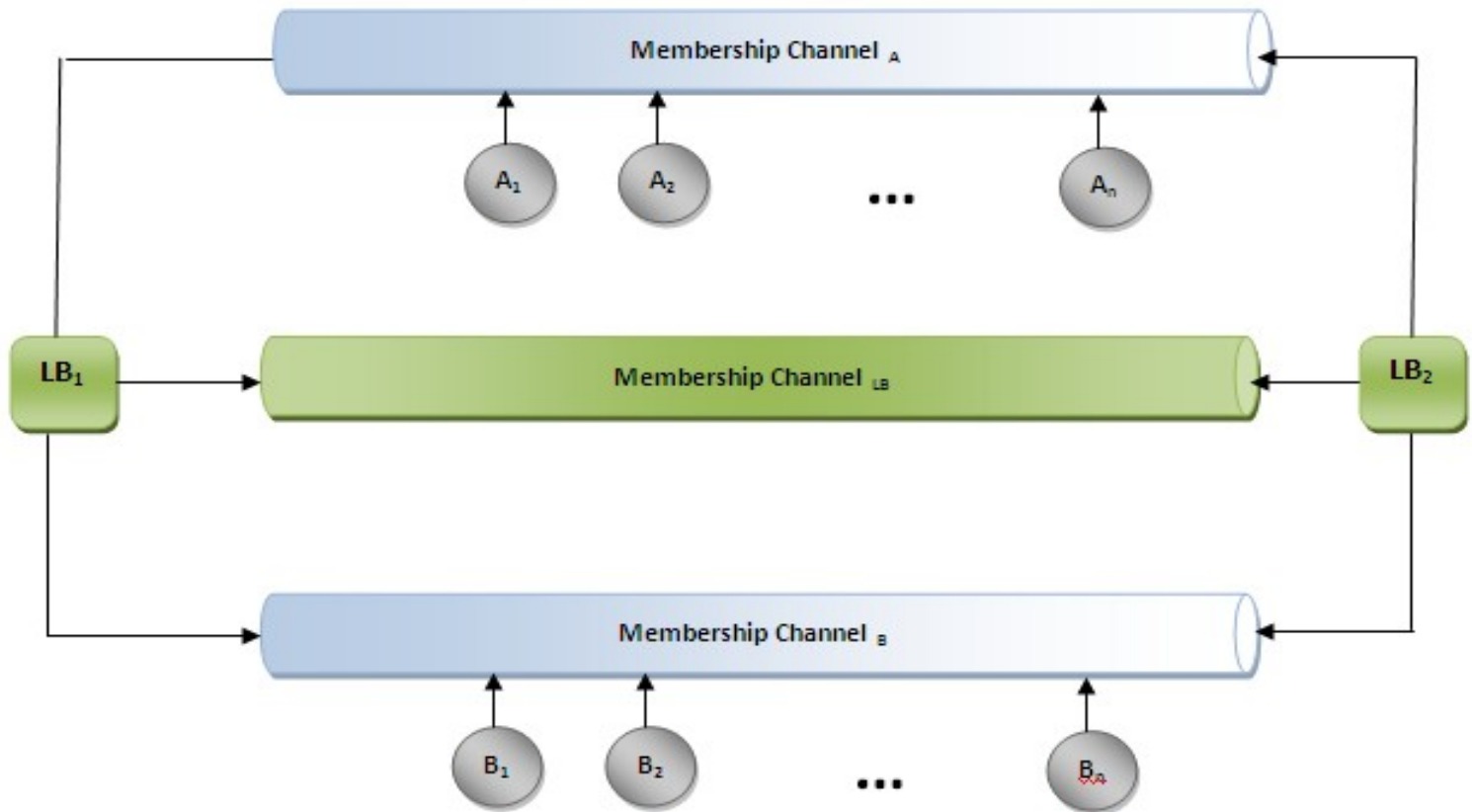
WKA Based Membership (3/3)

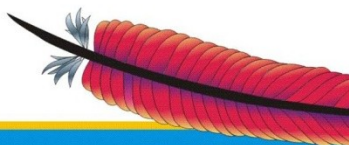


A well-known load balancer rejoins after crashing

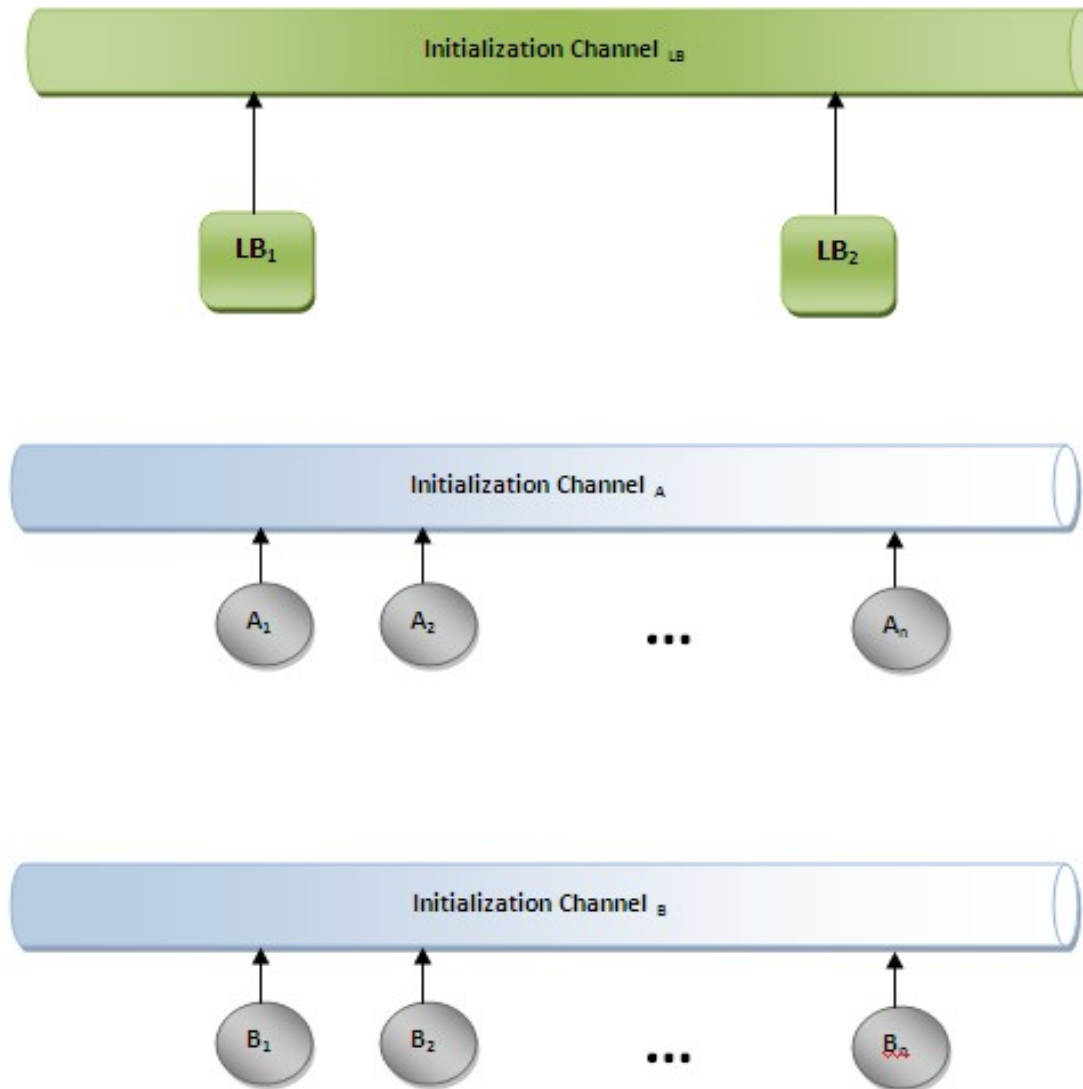


Membership Channel Architecture





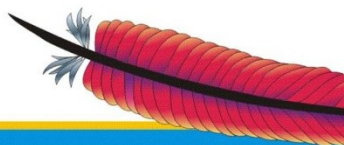
Initialization Channel Architecture



Synapse Configuration - axis2.xml

```
<cluster class="org.apache.axis2.clustering.tribes.TribesClusterManager" enable="true">
  <parameter name="AvoidInitiation">>false</parameter>
  <parameter name="membershipScheme">wka</parameter>
  <parameter name="domain">apache.synapse.domain</parameter>
  <parameter name="synchronizeAll">>true</parameter>
  <parameter name="maxRetries">10</parameter>
  <parameter name="mcastAddress">228.0.0.4</parameter>
  <parameter name="mcastPort">45564</parameter>
  <parameter name="mcastFrequency">500</parameter>
  <parameter name="memberDropTime">3000</parameter>
  <parameter name="localMemberPort">4000</parameter>
  <members>
    <member>
      <hostName>${ELASTIC_IP}</hostName>
      <port>4000</port>
    </member>
  </members>
  <loadBalancer enable="true">
    <applicationDomain name="apache.axis2.app.domain"
      handler="org.apache.axis2.clustering.DefaultLoadBalanceEventHandler"/>
  </loadBalancer>
  <contextManager class="org.apache.axis2.clustering.context.DefaultContextManager"
    enable="false">
    <listener class="org.apache.axis2.clustering.context.DefaultContextManagerListener"/>
    <replication>
      <defaults>
        <exclude name="local_*/>
        <exclude name="LOCAL_*/>
      </defaults>
      <context class="org.apache.axis2.context.ConfigurationContext">
        <exclude name="UseAsyncOperations"/>
        <exclude name="SequencePropertyBeanMap"/>
      </context>
      <context class="org.apache.axis2.context.ServiceGroupContext">
        <exclude name="my.sandesha.*/>
      </context>
      <context class="org.apache.axis2.context.ServiceContext">
        <exclude name="my.sandesha.*/>
      </context>
    </replication>
  </contextManager>
</cluster>
```

Apache
CON



Leading the Way
of Open Source

Synapse Configuration - synapse.xml

```
<definitions xmlns="http://ws.apache.org/ns/synapse">  
  
  <task class="org.wso2.ec2autoscale.LoadAnalyzerTask" name="LoadAnalyzer"...>  
  
    <sequence name="main" onError="errorHandler">  
      <in>  
        <autoscaleIn/>  
        <send>  
          <endpoint name="dynamicLB">  
            <dynamicLoadbalance failover="true"  
              algorithm="org.apache.synapse.endpoints.algorithms.RoundRobin">  
              <membershipHandler  
                class="org.apache.synapse.core.axis2.Axis2LoadBalanceMembershipHandler">  
                <property name="applicationDomain" value="apache.axis2.app.domain"/>  
              </membershipHandler>  
            </dynamicLoadbalance>  
          </endpoint>  
        </send>  
        <drop/>  
      </in>  
  
      <out>  
        <autoscaleOut/>  
        <!-- Send the messages where they have been sent (i.e. implicit To EPR) -->  
        <send/>  
      </out>  
    </sequence>  
  
    <sequence name="errorHandler">  
      <autoscaleOut/>  
      <makefault response="true">  
        <code value="tns:Receiver" xmlns:tns="http://www.w3.org/2003/05/soap-envelope"/>  
        <reason value="COULDN'T SEND THE MESSAGE TO THE SERVER."/>  
      </makefault>  
      <send/>  
    </sequence>  
  
</definitions>
```

ApacheCon



Leading the
of Open Sou

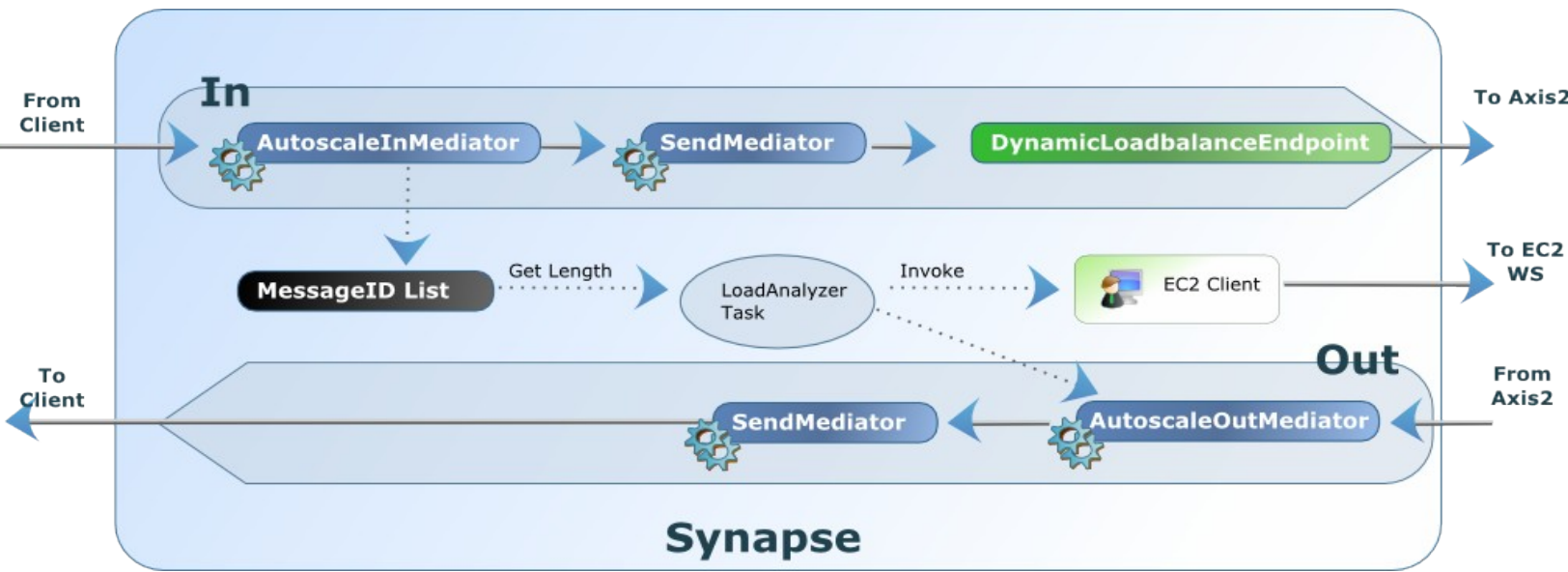
Synapse Configuration

- AutoscaleInMediator
- AutoscaleOutMediator
- Autoscale Task



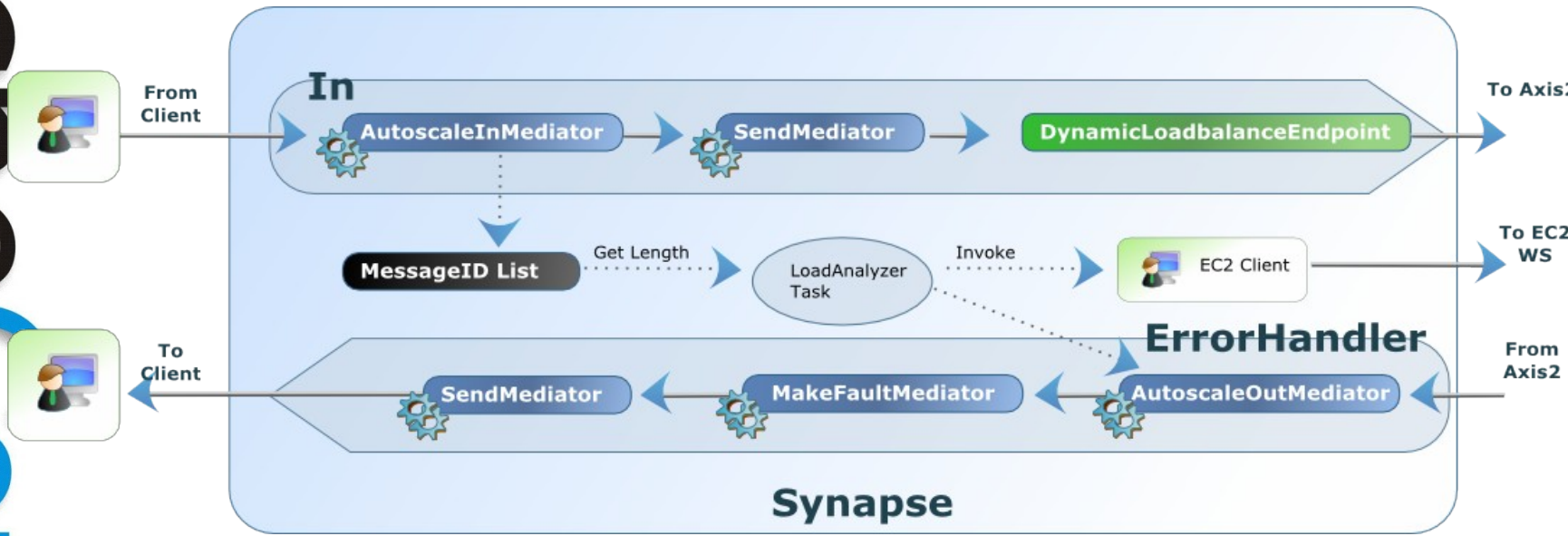
Synapse Configuration

Normal Flow



Synapse Configuration

Fault Flow



Synapse Autoscale Task

```
<task class="org.wso2.ec2autoscale.LoadAnalyzerTask" name="LoadAnalyzer">
  <property name="ec2PrivateKey" value="pk.pem"/>
  <property name="ec2Cert" value="cert.pem"/>
  <property name="instanceType" value="m1.small"/>
  <property name="applicationGroup" value="autoscale-app"/>
  <property name="loadBalancerGroup" value="autoscale-lb"/>
  <property name="instanceAdditionalInfo" value="EC2 autoscale instance"/>
  <property name="key" value="autoscale"/>
  <property name="applicationPayload" value="/tmp/axis2/axis2-payload.zip"/>
  <property name="loadBalancerPayload" value="/tmp/payload.zip"/>
  <!--
  The elasticIP property can be overridden by the ELASTIC_IP environment var,
  which is set in the payload -->
  <property name="elasticIP" value="75.101.131.185"/>

  <property name="messageExpiryTime" value="60000"/>

  <property name="maxAppInstances" value="10"/>
  <property name="minAppInstances" value="2"/>

  <property name="minLoadBalancerInstances" value="2"/>

  <property name="scaleUpListLength" value="15"/>
  <property name="scaleUpObservationRounds" value="12"/>
  <property name="instancesPerScaleUp" value="1"/>

  <property name="scaleDownListLength" value="5"/>
  <property name="scaleDownObservationRounds" value="50"/>

  <trigger interval="5"/>
</task>
```

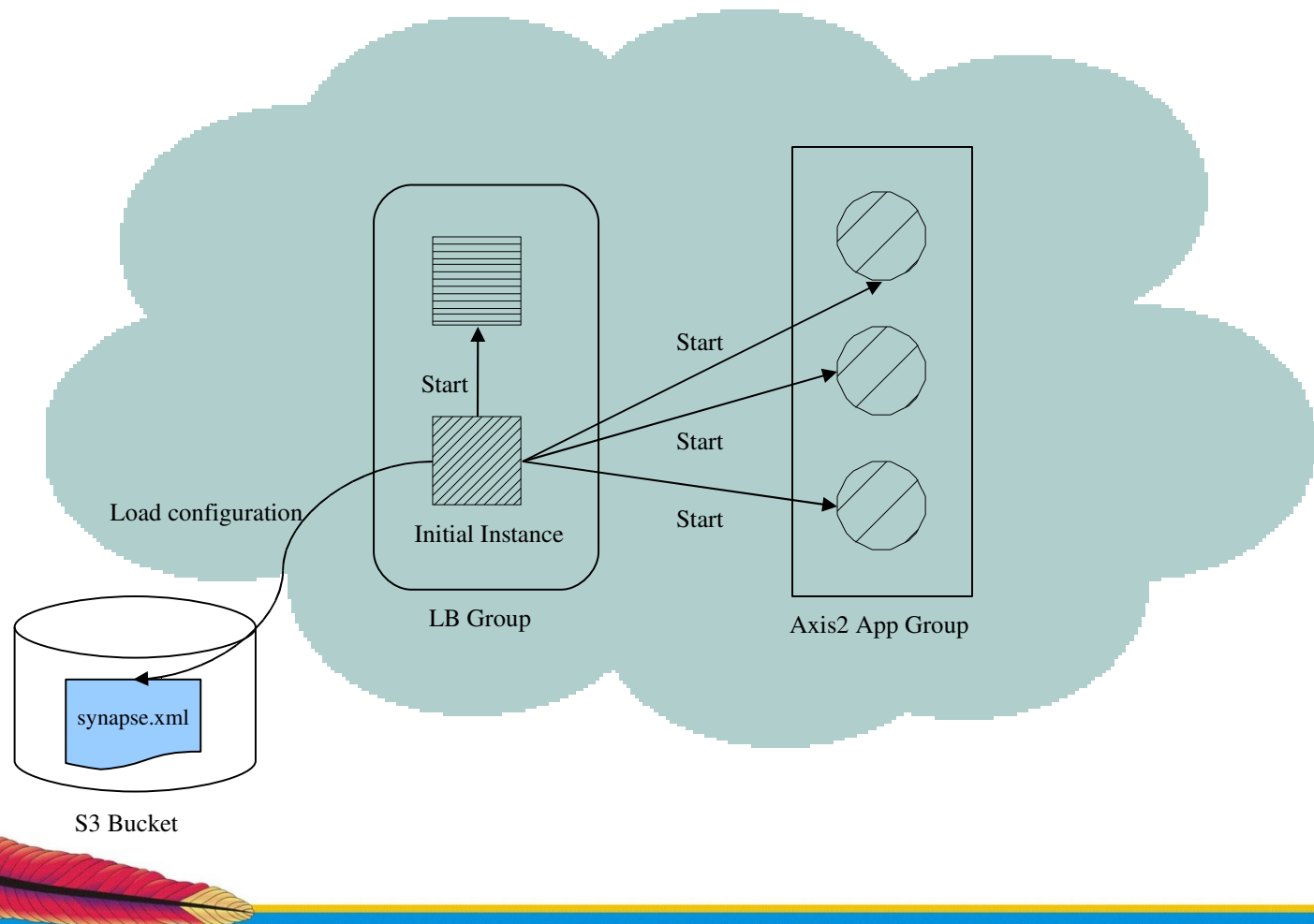


Synapse Autoscale Task

- Sanity Check
- Autoscaling
 - Scale up
 - Scale down



Booting up



Axis2 Configuration - axis2.xml

```
<cluster class="org.apache.axis2.clustering.tribes.TribesClusterManager" enable="true">
  <parameter name="AvoidInitiation" true</parameter>
  <parameter name="membershipScheme">wka</parameter>
  <parameter name="domain">apache.axis2.app.domain</parameter>
  <parameter name="synchronizeAll">true</parameter>
  <parameter name="maxRetries">10</parameter>
  <parameter name="mcastAddress">228.0.0.4</parameter>
  <parameter name="mcastPort">45564</parameter>
  <parameter name="mcastFrequency">500</parameter>
  <parameter name="memberDropTime">3000</parameter>
  <members>
    <member>
      <hostName>${ELASTIC_IP}</hostName>
      <port>4000</port>
    </member>
  </members>
</cluster>
<contextManager class="org.apache.axis2.clustering.context.DefaultContextManager"
  enable="false">
  <listener class="org.apache.axis2.clustering.context.DefaultContextManagerListener"/>
  <replication>
    <defaults>
      <exclude name="local_*/>
      <exclude name="LOCAL_*/>
    </defaults>
    <context class="org.apache.axis2.context.ConfigurationContext">
      <exclude name="UseAsyncOperations"/>
      <exclude name="SequencePropertyBeanMap"/>
    </context>
    <context class="org.apache.axis2.context.ServiceGroupContext">
      <exclude name="my.sandesha.*/>
    </context>
    <context class="org.apache.axis2.context.ServiceContext">
      <exclude name="my.sandesha.*/>
    </context>
  </replication>
</contextManager>
</cluster>
```

Apache
CON

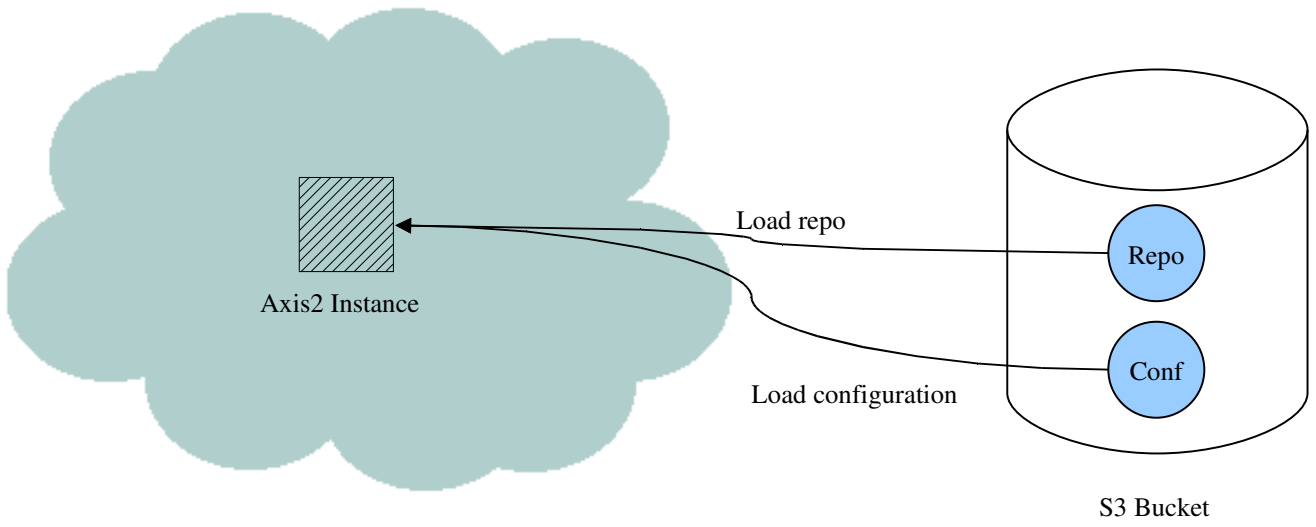


Leading the
of Open Sou

More Implementation Details

- Single AMI – autoscalews
- Start
 - `ec2-run-instances ami-a03fdbbc9 -k autoscale -f payload.zip -g autoscale-lb`
- Payload
 - Extract params from payload
 - Env variables
- Axis2 & Synapse Configuration files & Repositories
 - Maintained on S3
- Fault Tolerance
 - Monitoring cron job
 - Java Service Wrapper daemons
 - Future: Use monit
 - Future: Axis2/Synapse agent to check process status

Axis2 Configuration & Repository



**Questions
Suggestions
Improvements**



ApacheCon

Thank You

