


Creating Pools of 10s or 100s of Virtual Machines

Andrei Savu
ApacheCon NA 2013


Who is this guy?

- Founder of Axemblr.com
 - Organizer of Bucharest JUG (bjug.ro)
 - Apache Whirr PMC, ZooKeeper contributor
 - Passion for DevOps & Data Analysis
 - Connect with me on [LinkedIn](#)
- 

@ Axemblr

- Data Processing Infrastructure
 - Deployment Automation
 - Product: Hadoop On-Demand Appliance
 - Open Source (part of our DNA)
 - Fair amount of consulting (bootstrapping)
- 

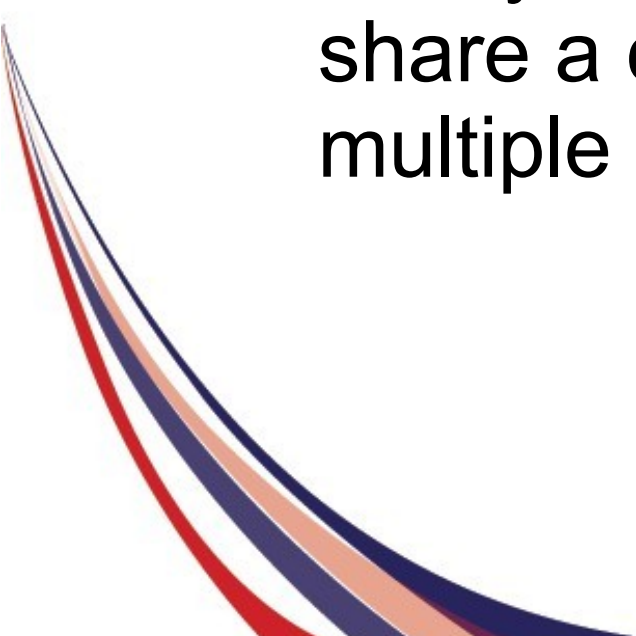
Agenda

- What is Provisionr?
 - Challenges & Architecture
 - Demo (HDFS on EC2)
 - Future @ Apache Incubator
- 


What is Provisionr?

.. and how does it help me create
pools of virtual machines?


What?

- Simple Service for Managing Pools of 10s or 100s of Virtual Machines
 - A way to create clusters of machines that share a common set of characteristics on multiple cloud providers
- 

Characteristics like?

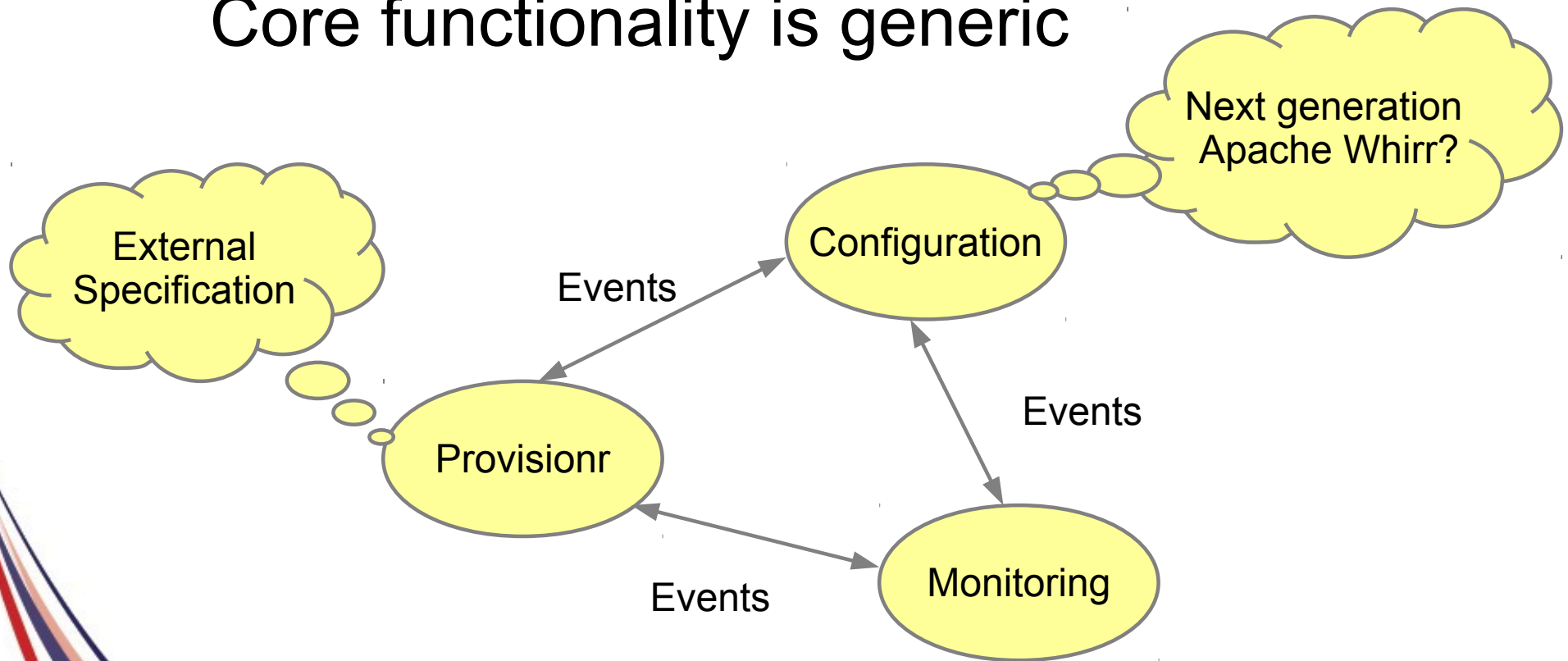
- Operating system
 - Pre-installed packages & binaries
 - Sane DNS settings (forward & reverse dns resolution)
 - NTP settings
 - Network settings
 - Firewall
 - SSH config
 - Admin access
 - VPN access
 - etc.
- 

Why? (initially)

- Setup on-demand Hadoop clusters (Axemblr)
 - Handles basic setup for large clusters
 - Service config by using 3rd party apps like Ambari or Cloudera Manager
- 

Why? (long term)

Core functionality is generic



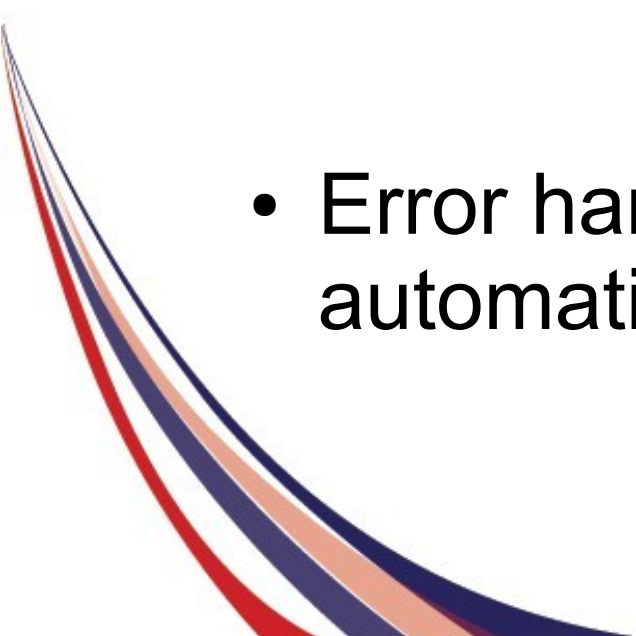
FAQ: Looks like Puppet?

- No
- Provisionr is actually using Puppet
- Focus: Interact with IaaS APIs to start machines in groups with minimal configs (as listed before). Simple & reliable.


Challenges

How is the game different when we work with 50-100+ virtual machines?

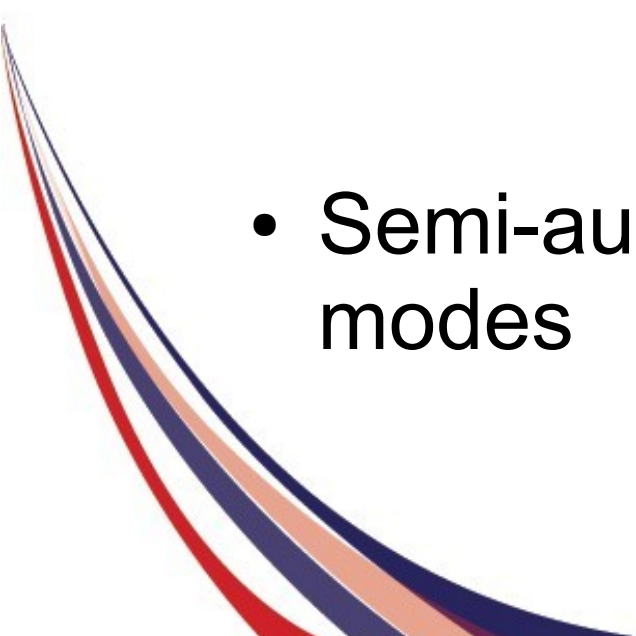
Challenges #1

- API Throttling (batch calls)
 - Concurrency Control (across multiple instances)
 - Error handling, partial failures and automatic retries (idempotency)
- 

Challenges #2

- Granular internal workflows (short transactions)
 - State persistence across restarts and upgrades
 - Audit & Logging
- 

Challenges #3

- Integrating multiple native provider SDKs
 - Provide a plugin architecture (run just a sub-set of all the features)
 - Semi-automated and fully automated modes
- 

Challenges #4


- Automatic creation of gold images



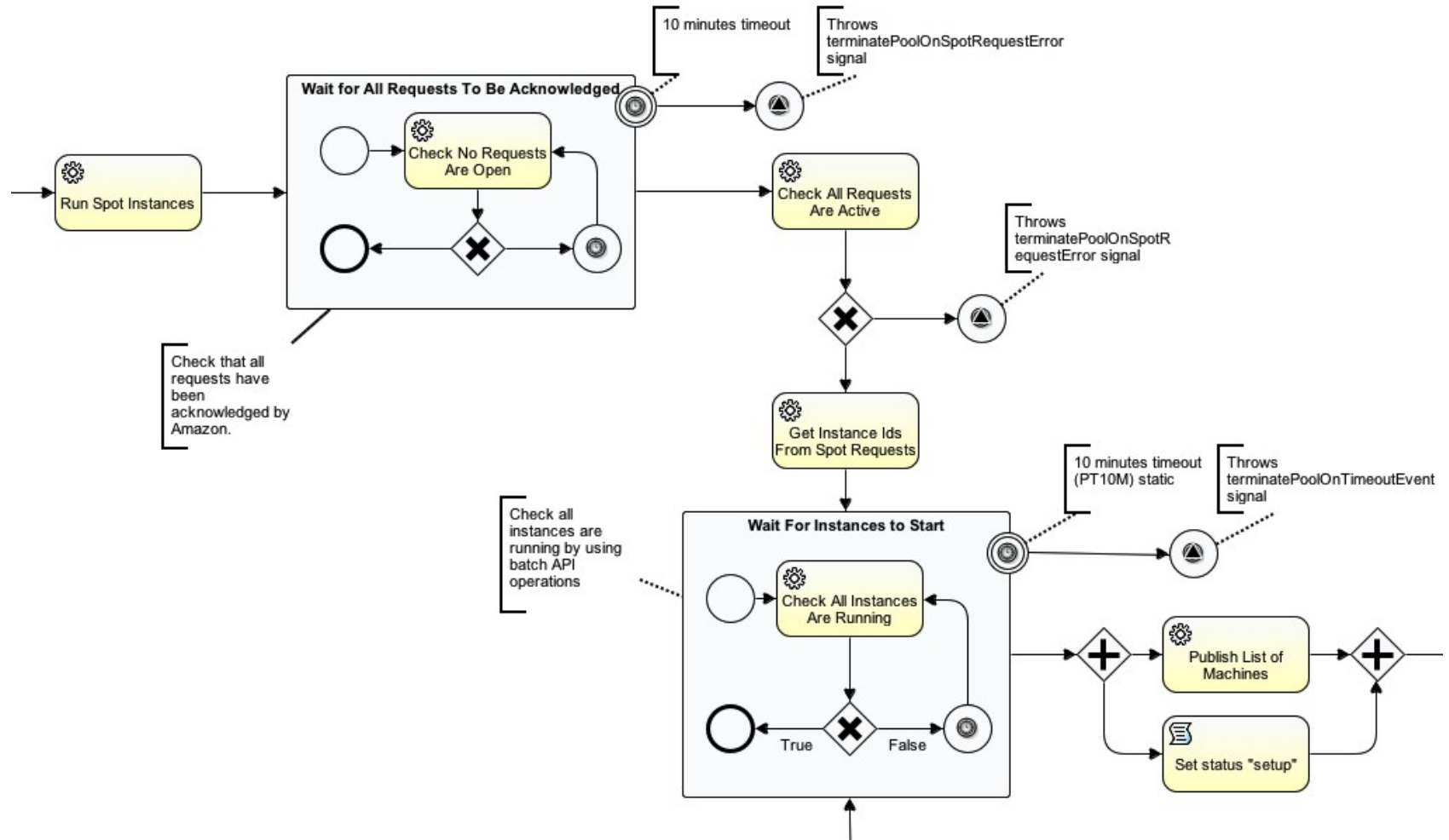
Architecture

Building Blocks, Internals,
Persistence, Packaging, Plugins

Activiti (from Alfresco)

- Light-weight workflow engine (BPM)
 - Has a nice Java API
 - Has a nice set of tools
 - Handles persistence as expected
 - Good error handling (retryable activities)
- 

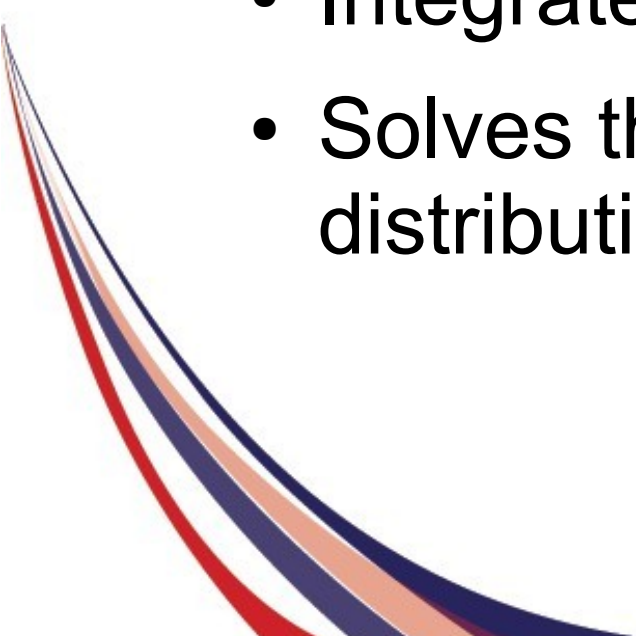
Activiti – Process Execution



Activiti – Interactive View

The screenshot displays the Activiti Explorer web application. The browser address bar shows `localhost:8181/activiti-explorer/#myProcess/404`. The application header includes the Activiti Explorer logo, navigation tabs for `Tasks`, `Processes`, and `Manage`, and a user profile for `Kermit The Frog`. The main content area is titled `Amazon Process (404)` and shows a `Process Diagram`. The diagram starts with a start node leading to a parallel gateway. This gateway splits into two parallel tasks: `Ensure Security ...` and `Ensure Key Pair ...` (the latter is highlighted with a red border). These tasks merge at another parallel gateway, which leads to a `Run On Deman...` task. This task is followed by a `Wait For Instances to Start` container. Inside this container, there is a `Check All Instan...` task, a gateway, and a timer icon. The process then flows to a `Publish List of ...` task, followed by a `Just Wait` task. Below the diagram, the `Tasks` section indicates `There are no tasks for this process instance.` and the `Variables` section shows an empty table with columns `NAME` and `VALUE`. The footer of the application contains the text `© Activiti.org. All rights reserved.`

Apache Karaf

- Using it as an application server
 - Provides an interactive shell
 - Integrated with Activiti
 - Solves the packaging problem (custom distribution)
- 


IaaS SDKs

- AWS SDK for Java
 - <http://aws.amazon.com/sdkforjava/>
- jclouds (for CloudStack)
 - <http://www.jclouds.org/>


Demo Time ([video](#))

Provisionr & Rundeck
CDH4 HDFS cluster on EC2

Summary

- Provisionr solves the problem of creating large pools of virtual machines (100s)
 - Cloud portability by making the machines & the cluster indistinguishable from an application perspective on multiple clouds
- 

You're invited to vote!

- Apache Provisionr proposal ([wiki](#))
 - Check general@incubator.apache.org
 - Feedback at asavu@apache.org
 - Looking for mentors & contributors
- 

Thanks! Questions?

Andrei Savu
asavu@apache.org

Twitter: [@andreisavu](https://twitter.com/andreisavu)