



**Apache
Ambari**

Streamline Hadoop DevOps with Apache Ambari

Alejandro Fernandez



2016
**HADOOP
SUMMIT**

**Melbourn
e**

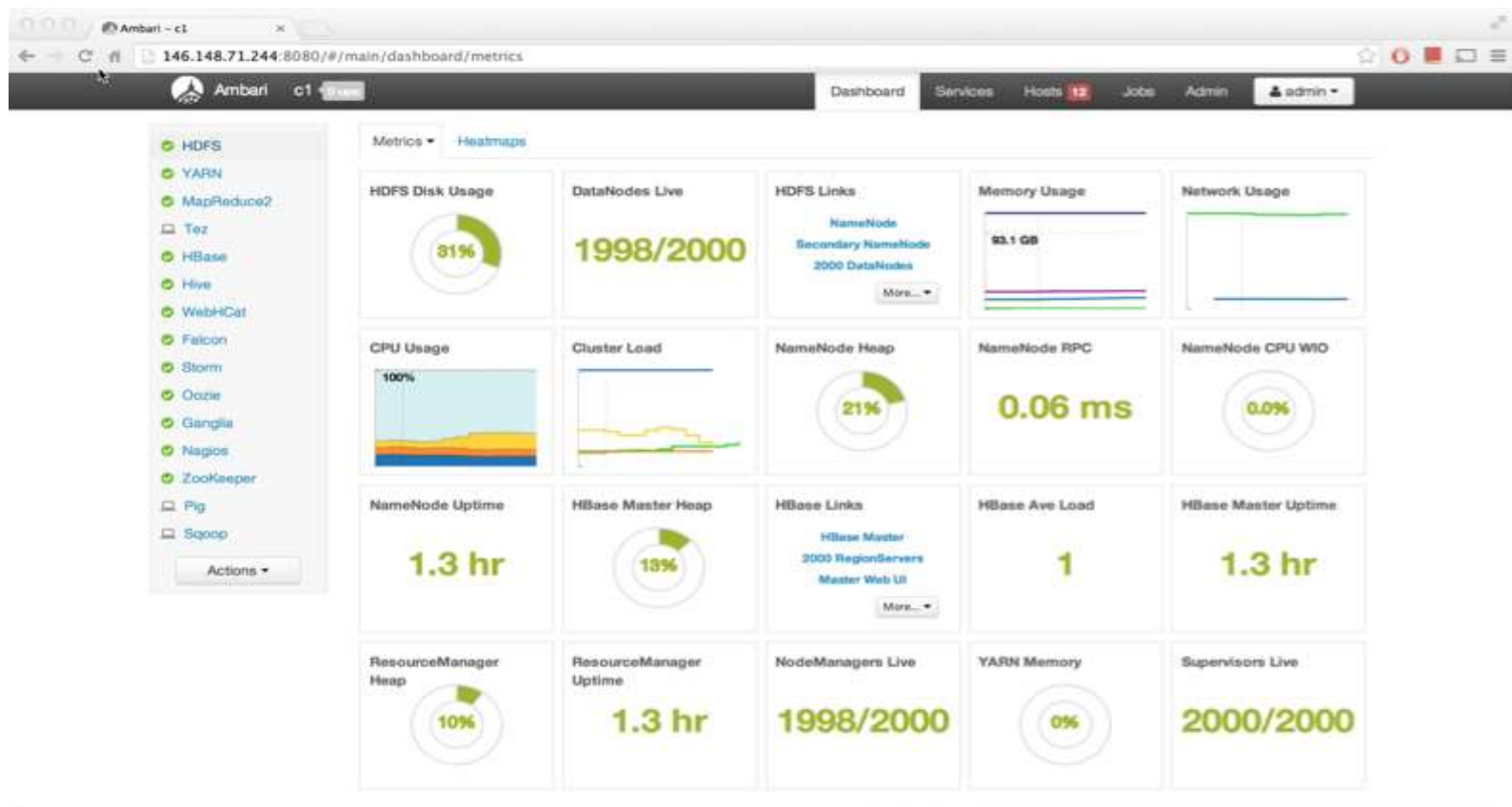
Speaker



Alejandro Fernandez
Sr. Software Engineer @ Hortonworks
Apache Ambari PMC
alejandro@apache.org

What is Apache Ambari?

Apache Ambari is the open source platform to provide, manage and monitor Hadoop clusters.

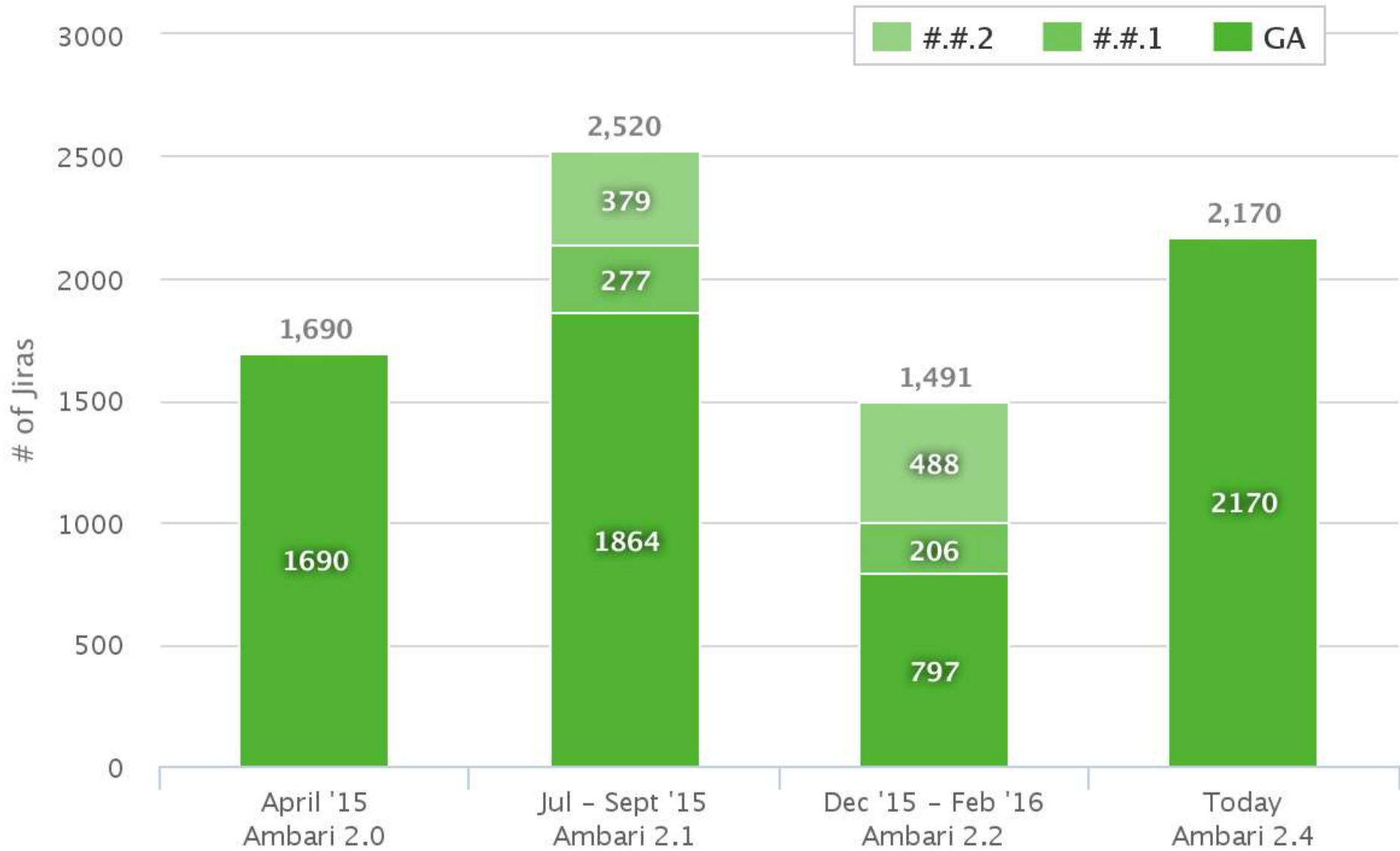




**Apache
Ambari**



Apache Ambari Jiras



4 years old

Exciting Enterprise Features in Ambari 2.4

- New Services: Log Search, Zeppelin, Hive LLAP
- Role Based Access Control
- Management Packs
- Grafana UI for Ambari Metrics System
- New Views: Zeppelin, Storm

More in Ambari 2.4

Core Features

- Alerts: Customizable props and thresholds (AMBARI-14898)
- Alerts: Retry tolerance (AMBARI-15686)
- Alerts: New HDFS Alerts (AMBARI-14800)
- New Host Page Filtering (AMBARI-15210)
- Remove Service from UI (AMBARI-14759)
- Support for SLES 12 (AMBARI-16007)
- Stability: Database Consistency Checking (AMBARI-16258)
- Customizable Ambari Log + PID Dirs (AMBARI-15300)
- New Version Registration Experience (AMBARI-15724)
- Log Search Technical Preview (AMBARI-14927)

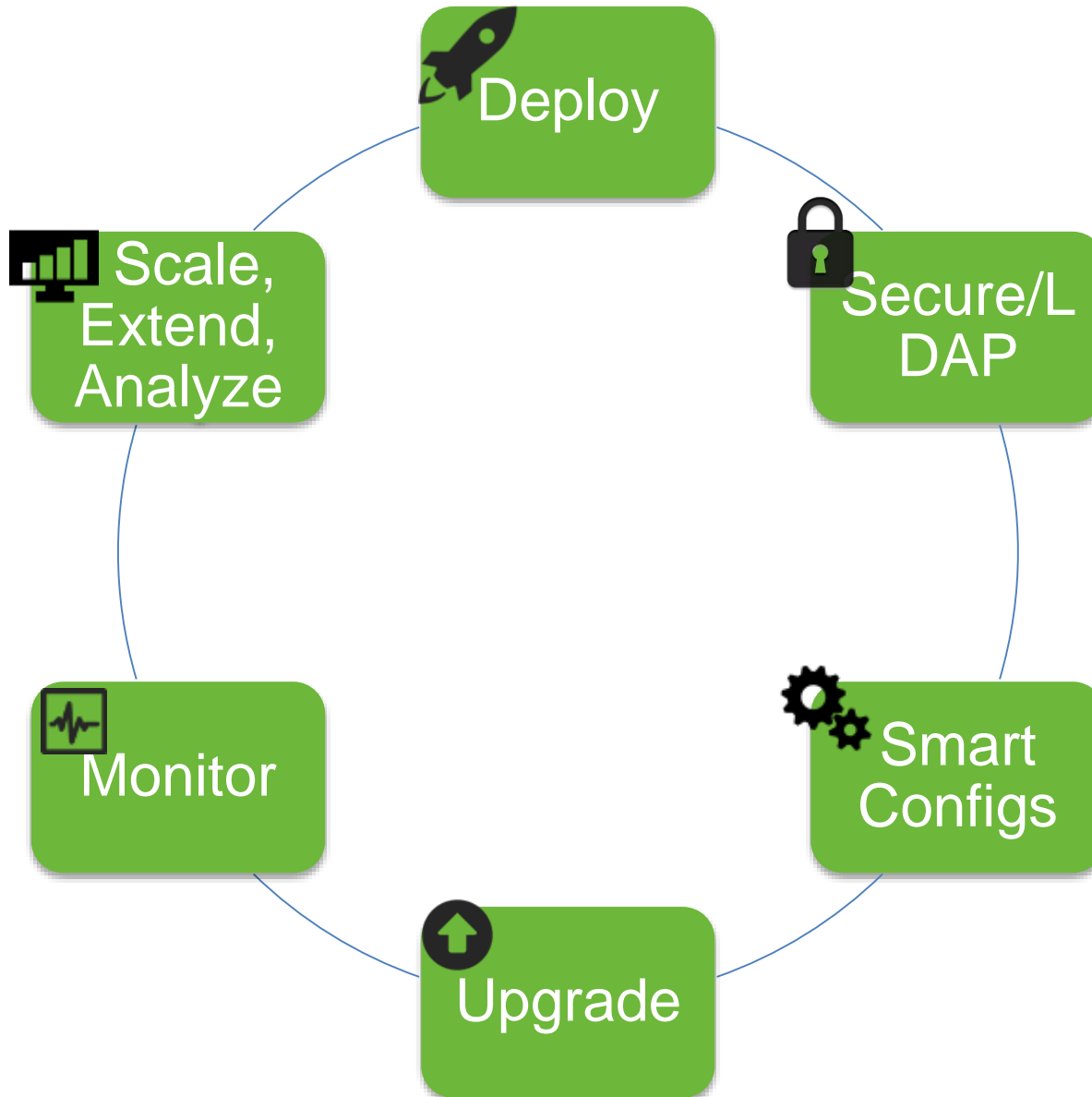
Security Features

- Operational Audit Logging (AMBARI-15241)
- Role-Based Access Control (AMBARI-13977)
- Automated Setup of Ambari Kerberos through Blueprints (AMBARI-15561)
- Automated Setup of Ambari Proxy User (AMBARI-15561)
- Customizable Host Reg. SSH Port (AMBARI-13450)

Views Framework Features

- View URLs for bookmarks (AMBARI-15821), View Refresh (AMBARI-15682)
- Inherit Cluster Permissions (AMBARI-16177)
- Remote Cluster Registration (AMBARI-16274)

Simply Operations - Lifecycle



Deploy On Premise

Ambari UI wizard handles all of these combinations and makes recommendations based on host specs.



Deploy On The Cloud



Certified environments

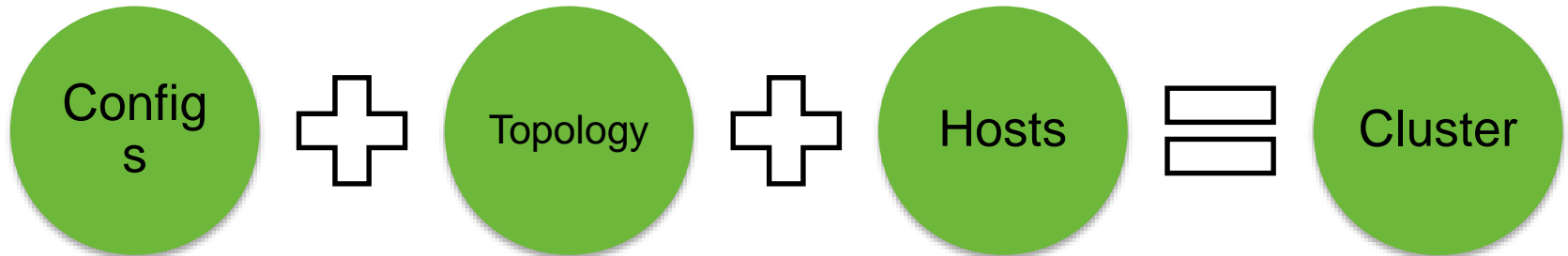
Sysprepped VMs

Hundreds of similar clusters



Deploy with Blueprints

- Systematic way of defining a cluster



- Export existing cluster into blueprint

 `/api/v1/clusters/:clusterName?format=blueprint`

Create a cluster with Blueprints

1. POST /api/v1/blueprints/my-blueprint

```
{
  "configurations" : [
    {
      "hdfs-site" : {
        "dfs.datanode.data.dir" : "/hadoop/1,
          /hadoop/2, /hadoop/3"
      }
    }
  ],
  "host_groups" : [
    {
      "name" : "master-host",
      "components" : [
        { "name" : "NAMENODE" },
        { "name" : "RESOURCEMANAGER" },
        ...
      ],
      "cardinality" : "1"
    },
    {
      "name" : "worker-host",
      "components" : [
        { "name" : "DATANODE" },
        { "name" : "NODEMANAGER" },
        ...
      ],
      "cardinality" : "1+"
    }
  ],
  "Blueprints" : {
    "stack_name" : "HDP",
    "stack_version" : "2.5"
  }
}
```

2. POST /api/v1/clusters/my-cluster

```
{
  "blueprint" : "my-blueprint",
  "host_groups" : [
    {
      "name" : "master-host",
      "hosts" : [
        {
          "fqdn" : "master001.ambari.apache.org"
        }
      ]
    },
    {
      "name" : "worker-host",
      "hosts" : [
        {
          "fqdn" : "worker001.ambari.apache.org"
        },
        {
          "fqdn" : "worker002.ambari.apache.org"
        },
        ...
        {
          "fqdn" : "worker099.ambari.apache.org"
        }
      ]
    }
  ]
}
```

Create a cluster with Blueprints

1. POST /api/v1/blueprints/my-blueprint

```
{
  "configurations" : [
    {
      "hdfs-site" : {
        "dfs.datanode.data.dir" : "/hadoop/1,
          /hadoop/2,/hadoop/3"
      }
    }
  ],
  "host_groups" : [
    {
      "name" : "master-host",
      "components" : [
        { "name" : "NAMENODE" },
        { "name" : "RESOURCEMANAGER" },
        ...
      ],
      "cardinality" : "1"
    },
    {
      "name" : "worker-host",
      "components" : [
        { "name" : "DATANODE" },
        { "name" : "NODEMANAGER" },
        ...
      ],
      "cardinality" : "1+"
    }
  ],
  "Blueprints" : {
    "stack_name" : "HDP",
    "stack_version" : "2.5"
  }
}
```

2. POST /api/v1/clusters/my-cluster

```
{
  "blueprint" : "my-blueprint",
  "host_groups" : [
    {
      "name" : "master-host",
      "hosts" : [
        {
          "fqdn" : "master001.ambari.apache.org"
        }
      ]
    },
    {
      "name" : "worker-host",
      "hosts" : [
        {
          "fqdn" : "worker001.ambari.apache.org"
        },
        {
          "fqdn" : "worker002.ambari.apache.org"
        },
        ...
        {
          "fqdn" : "worker099.ambari.apache.org"
        }
      ]
    }
  ]
}
```

Create a cluster with Blueprints

1. POST /api/v1/blueprints/my-blueprint

```
{
  "configurations" : [
    {
      "hdfs-site" : {
        "dfs.datanode.data.dir" : "/hadoop/1,
          /hadoop/2,/hadoop/3"
      }
    }
  ],
  "host_groups" : [
    {
      "name" : "master-host",
      "components" : [
        { "name" : "NAMENODE" },
        { "name" : "RESOURCEMANAGER" },
        ...
      ],
      "cardinality" : "1"
    },
    {
      "name" : "worker-host",
      "components" : [
        { "name" : "DATANODE" },
        { "name" : "NODEMANAGER" },
        ...
      ],
      "cardinality" : "1+"
    }
  ],
  "Blueprints" : {
    "stack_name" : "HDP",
    "stack_version" : "2.5"
  }
}
```

2. POST /api/v1/clusters/my-cluster

```
{
  "blueprint" : "my-blueprint",
  "host_groups" : [
    {
      "name" : "master-host",
      "hosts" : [
        {
          "fqdn" : "master001.ambari.apache.org"
        }
      ]
    },
    {
      "name" : "worker-host",
      "hosts" : [
        {
          "fqdn" : "worker001.ambari.apache.org"
        },
        {
          "fqdn" : "worker002.ambari.apache.org"
        },
        ...
        {
          "fqdn" : "worker099.ambari.apache.org"
        }
      ]
    }
  ]
}
```

Create a cluster with Blueprints

1. POST /api/v1/blueprints/my-blueprint

```
{
  "configurations" : [
    {
      "hdfs-site" : {
        "dfs.datanode.data.dir" : "/hadoop/1,
          /hadoop/2, /hadoop/3"
      }
    }
  ],
  "host_groups" : [
    {
      "name" : "master-host",
      "components" : [
        { "name" : "NAMENODE" },
        { "name" : "RESOURCEMANAGER" },
        ...
      ],
      "cardinality" : "1"
    },
    {
      "name" : "worker-host",
      "components" : [
        { "name" : "DATANODE" },
        { "name" : "NODEMANAGER" },
        ...
      ],
      "cardinality" : "1+"
    }
  ],
  "Blueprints" : {
    "stack_name" : "HDP",
    "stack_version" : "2.5"
  }
}
```

2. POST /api/v1/clusters/my-cluster

```
{
  "blueprint" : "my-blueprint",
  "host_groups" : [
    {
      "name" : "master-host",
      "hosts" : [
        {
          "fqdn" : "master001.ambari.apache.org"
        }
      ]
    },
    {
      "name" : "worker-host",
      "hosts" : [
        {
          "fqdn" : "worker001.ambari.apache.org"
        },
        {
          "fqdn" : "worker002.ambari.apache.org"
        },
        ...
        {
          "fqdn" : "worker099.ambari.apache.org"
        }
      ]
    }
  ]
}
```

Blueprints for Large Scale

- **Kerberos**, secure out-of-the-box
- **High Availability** is setup initially for NameNode, YARN, Hive, Oozie, etc
- **Host Discovery** allows Ambari to automatically install services for a Host when it comes online
- **Stack Advisor** recommendations

Blueprint Host Discovery

```
POST /api/v1/clusters/MyCluster/hosts
```

```
[
  {
    "blueprint" : "single-node-hdfs-test2",
    "host_groups" : [
      {
        "host_group" : "slave",
        "host_count" : 3,
        "host_predicate" : "Hosts/cpu_count>1"
      }, {
        "host_group" : "super-slave",
        "host_count" : 5,
        "host_predicate" : "Hosts/cpu_count>2&
          Hosts/total_mem>3000000"
      }
    ]
  }
]
```

Comprehensive Security

Kerberos

- MIT KDC
- Keytab management

LDAP/AD

- User auth
- Sync

Ranger

- Security policies
- Audit
- Authorization

Atlas

- Governance
- Compliance
- Lineage & history
- Data classification

Knox

- Perimeter security
- Supports LDAP/AD
- Sec. for REST/HTTP
- SSL



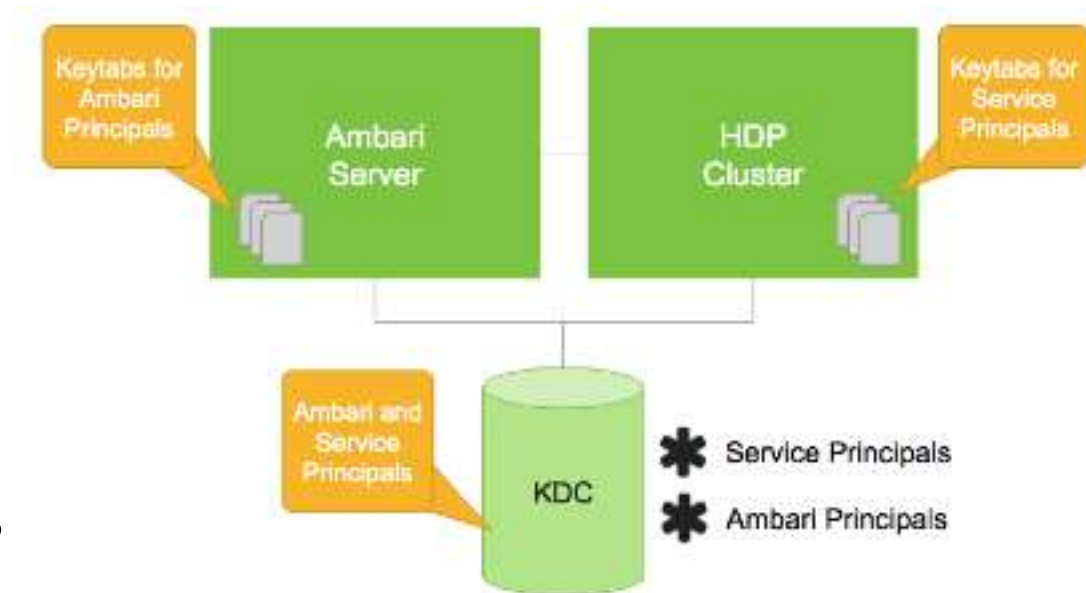
Kerberos

Ambari manages Kerberos principals and keytabs

Works with existing MIT KDC or Active Directory

Once Kerberized, handles

1. Adding hosts
2. Adding components to existing hosts
3. Adding services
4. Moving components to different hosts



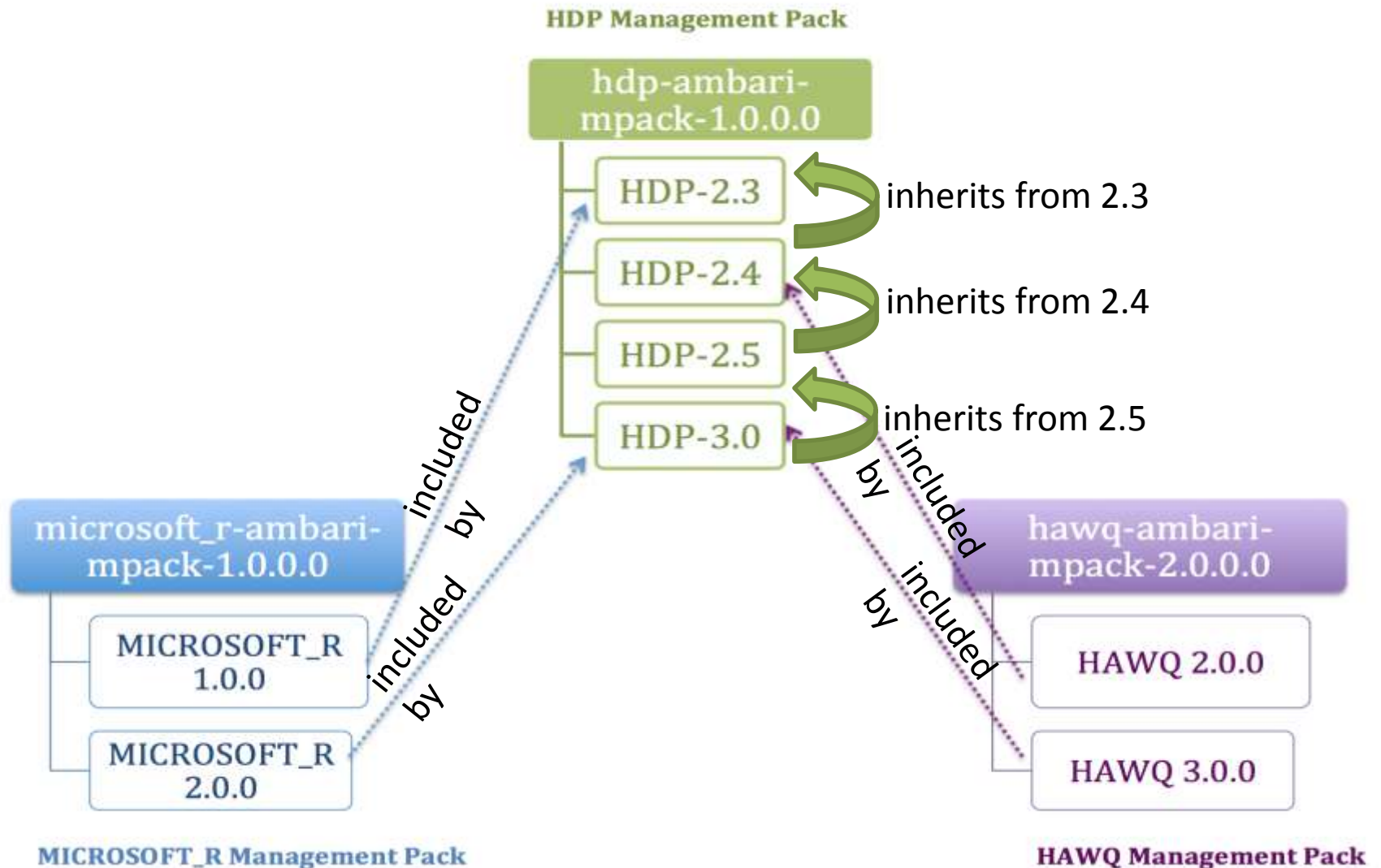
Management Packs

- Improved Release Management:
Decouple Ambari core from stacks



- Support Add-ons:
Release vehicle for 3rd party services, views
Self-contained release artifacts
Stack is an overlay of multiple management packs

Overlay of Management Packs



Management Pack++

Short Term Goals (Ambari 2.4)

- Retrofit in Stack Processing Framework
- Enable 3rd party to ship add-on services

Future Goals

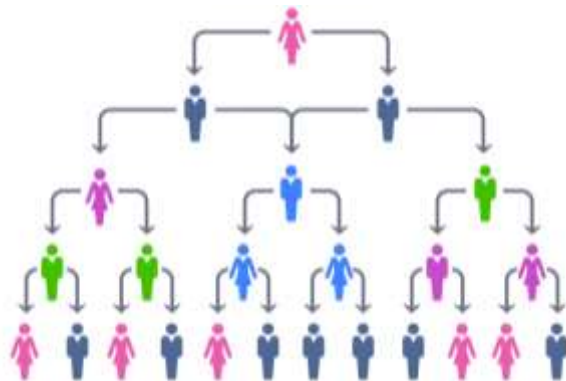
- Management Pack Framework
- Deliver Views

Role Based Access Control (RBAC)

✓ As Ambari & organizations grow, so do security needs



Ambari integrates with external authentication systems & LDAP



RBAC Terms

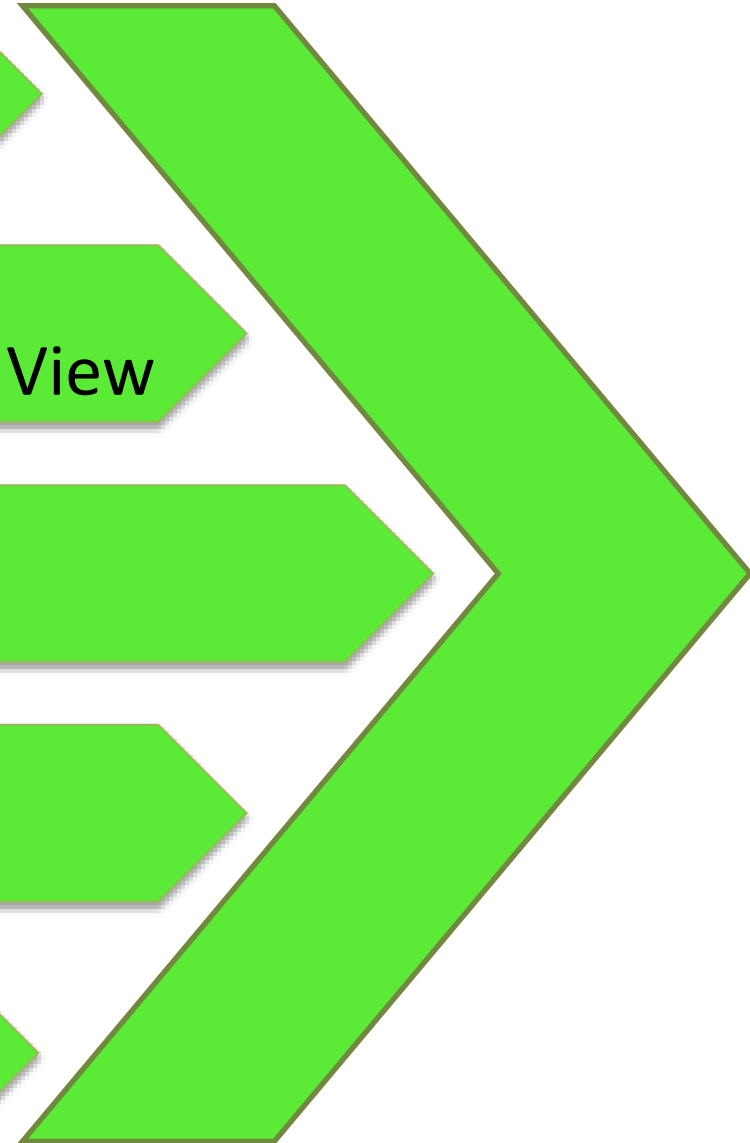
Roles have permissions
e.g., add services to cluster

Roles are applied to Resources. E.g.,
Ambari, particular Cluster, particular View

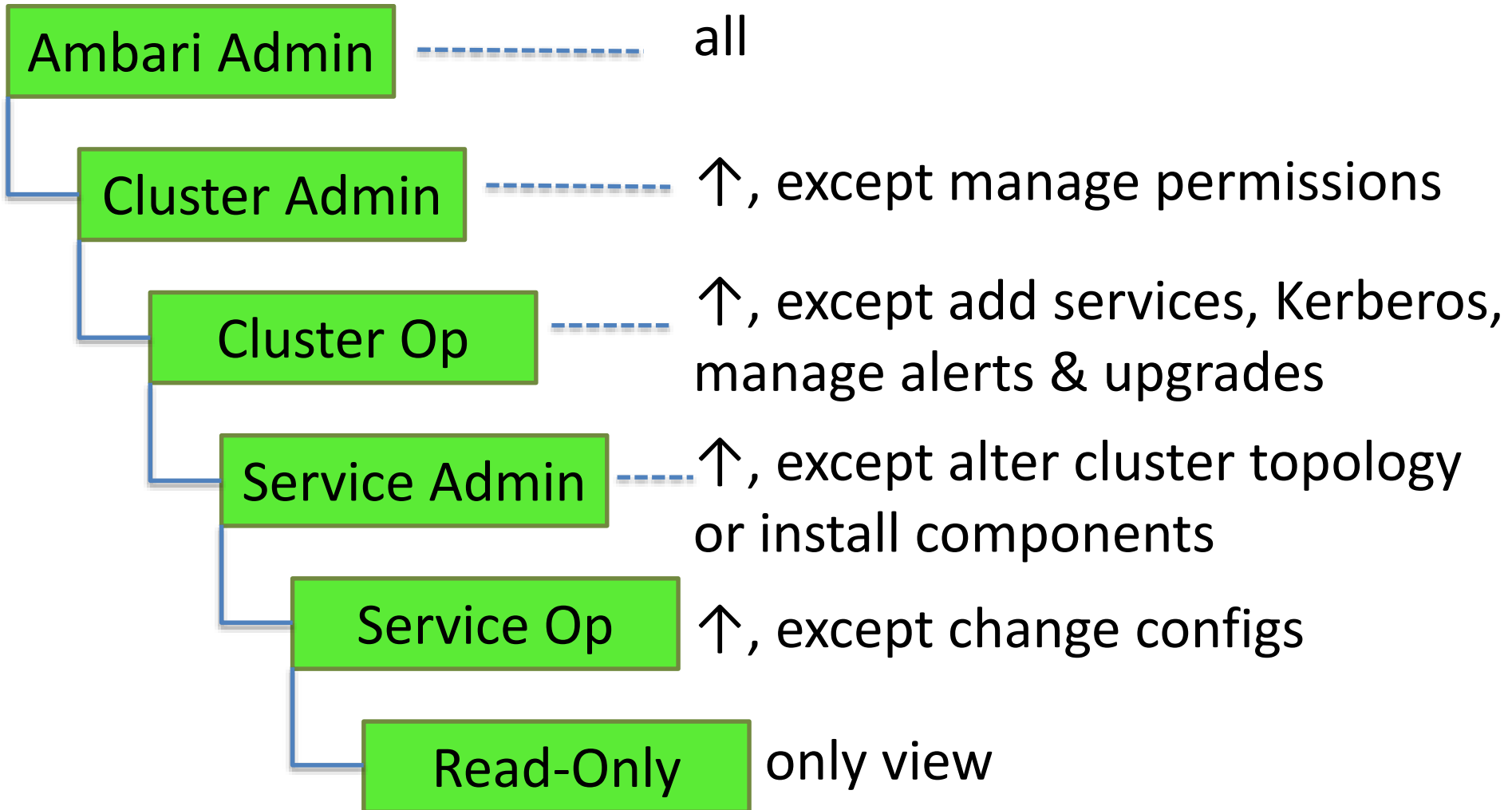
Users belong to groups

A group has a role

Users can also have additional roles

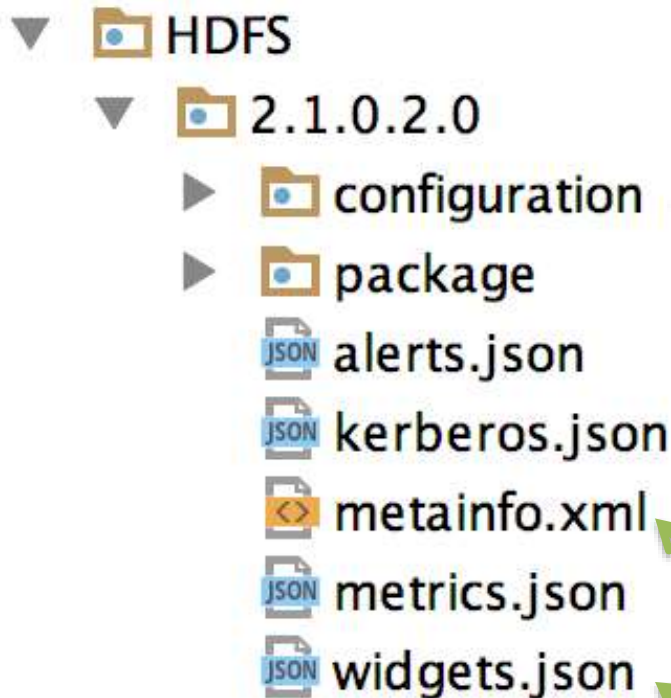


New RBAC Roles

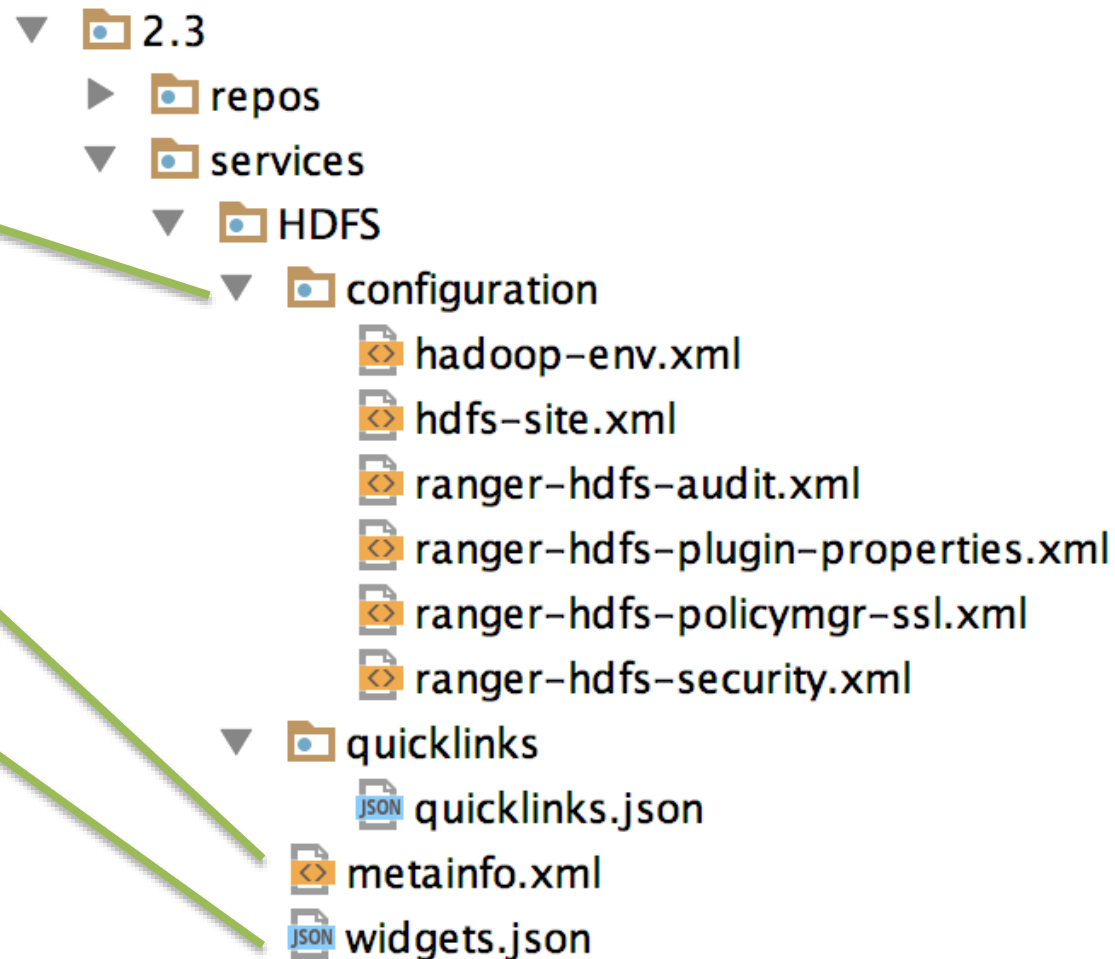


Service Layout

Common Services



Stack Override



Stack Advisor

Configuration

~~R~~erberos

HTTPS

Zookeeper Servers

Memory Settings

...

High Availability



Example

Atlas Servers

atlas.enableTLS = true|false

atlas.server.http.port = 21000

atlas.server.https.port = 21443



**atlas.rest.address =
http(s)://host:port**

Background: Upgrade Terminology

Manual Upgrade

- ✗ The user follows instructions to upgrade the stack
- ✗ Incurs downtime



Background: Upgrade Terminology

Rolling Upgrade

- ✓ Automated
- ✗ Upgrades one component per host at a time
- ✓ Preserves cluster operation and minimizes service impact



Manual Upgrade

- ✗ The user follows instructions to upgrade the stack
- ✗ Incurs downtime



Background: Upgrade Terminology

Express Upgrade

- ✓ Automated
- ✓ Runs in parallel across hosts
- ✗ Incurs downtime



Rolling Upgrade

- ✓ Automated
- ✗ Upgrades one component per host at a time
- ✓ Preserves cluster operation and minimizes service impact

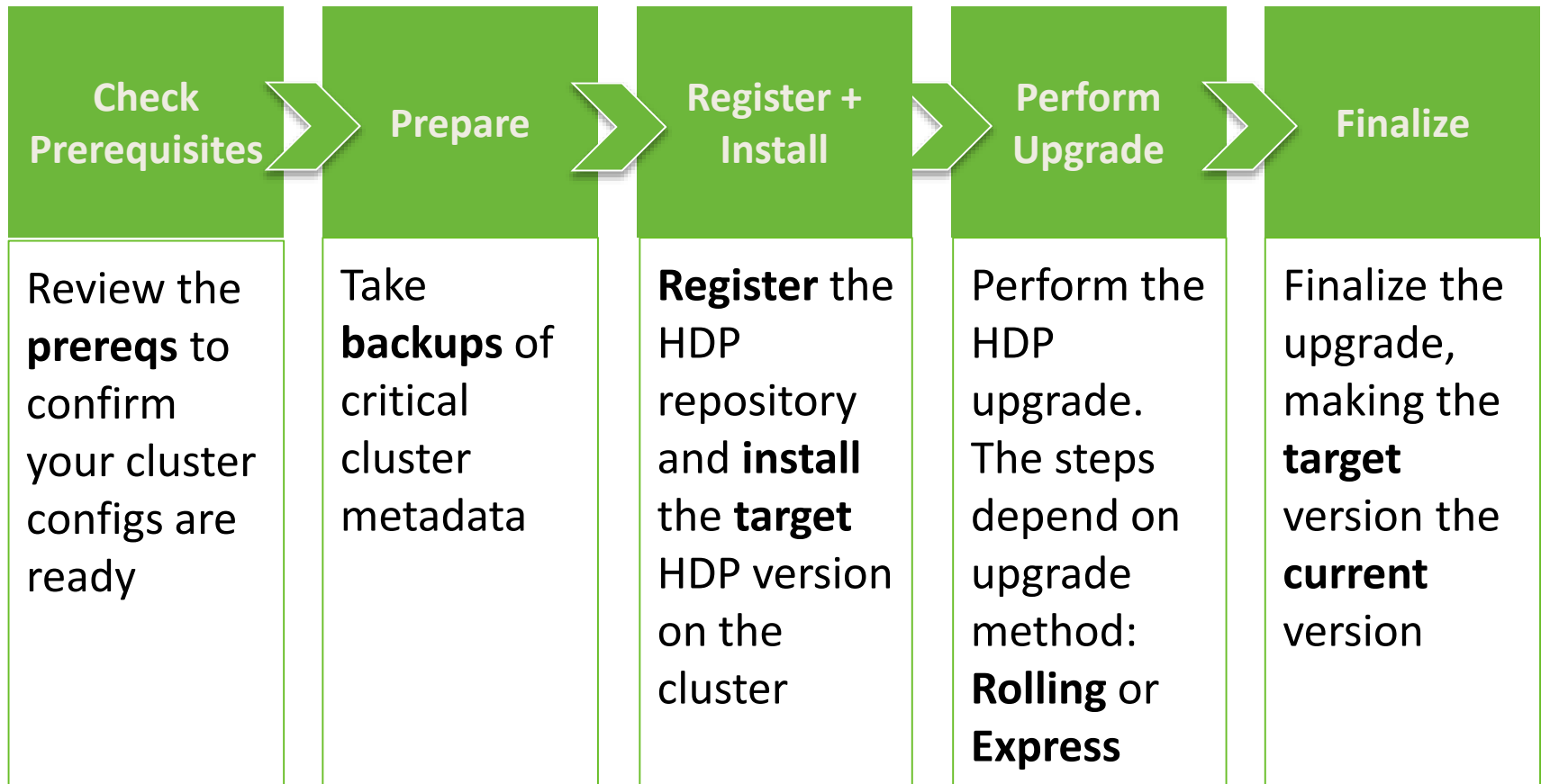


Manual Upgrade

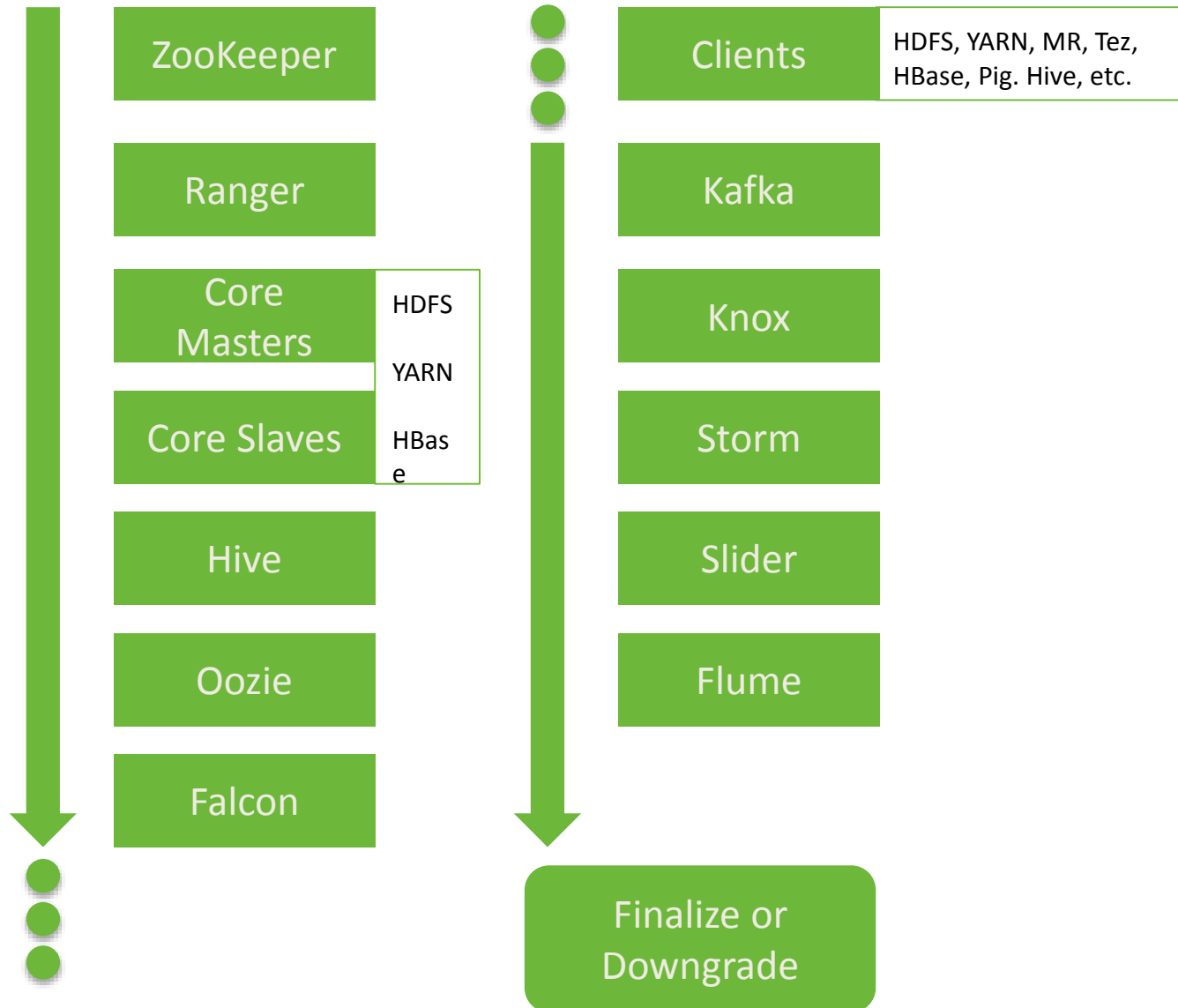
- ✗ The user follows instructions to upgrade the stack
- ✗ Incurs downtime



Automated Upgrade: Rolling or Express



Process: Rolling Upgrade



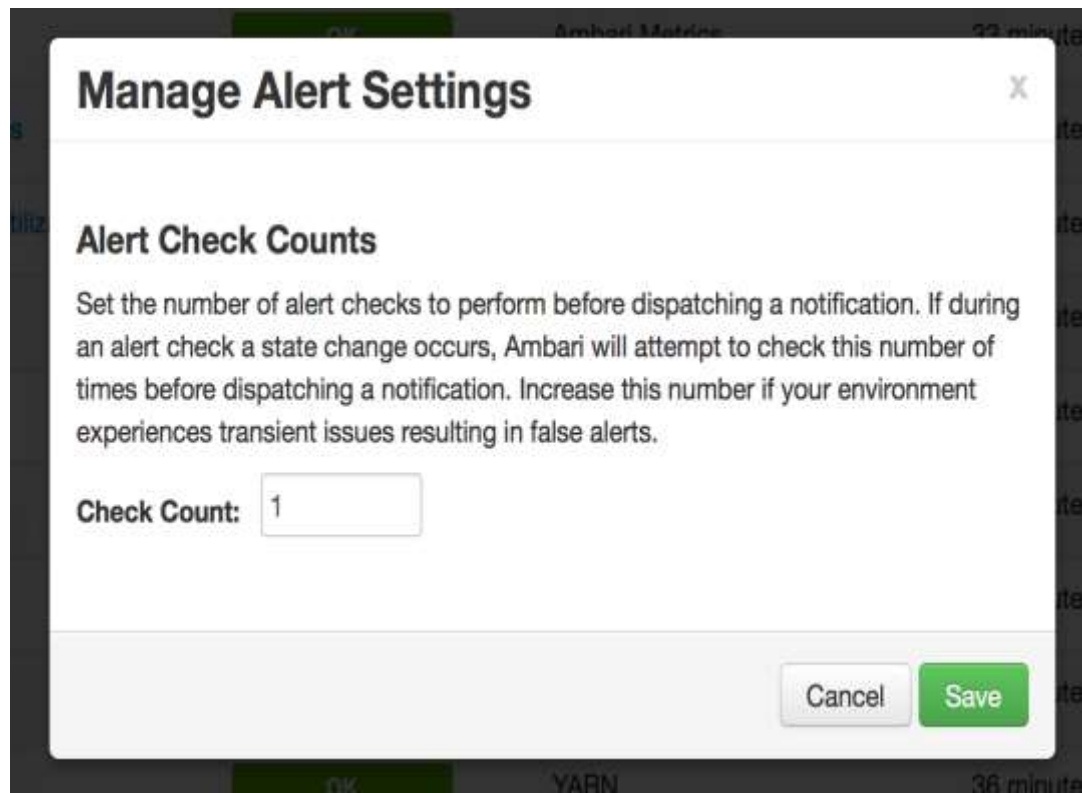
Alerting Framework

Alert Type	Description	Thresholds (units)
WEB	Connects to a Web URL. Alert status is based on the HTTP response code	Response Code (n/a) Connection Timeout (seconds)
PORT	Connects to a port. Alert status is based on response time	Response (seconds)
METRIC	Checks the value of a service metric. Units vary, based on the metric being checked	Metric Value (units vary) Connection Timeout (seconds)
AGGREGATE	Aggregates the status for another alert	% Affected (percentage)
SCRIPT	Executes a script to handle the alert check	Varies
SERVER	Executes a server-side runnable class to handle the alert check	Varies



Alert Check Counts

- Customize the number of times an alert is checked before dispatching a notification
- Avoid dispatching an alert notification (email, snmp) in case of transient issues



Manage Alert Settings [X]

Alert Check Counts

Set the number of alert checks to perform before dispatching a notification. If during an alert check a state change occurs, Ambari will attempt to check this number of times before dispatching a notification. Increase this number if your environment experiences transient issues resulting in false alerts.

Check Count:

Cancel Save

Alerts - Configuring the Check Count

Set globally for all alerts, or override for a specific alert

The image shows a screenshot of the Hadoop alert configuration interface. On the left, a green arrow labeled "Global Setting" points to the "Manage Alert Settings" option in the "Actions" dropdown menu. The dropdown menu also includes "Manage Alert Groups", "Manage Notifications", "Metrics Monitor Status", and "Metrics Collector Process". On the right, a green arrow labeled "Alert Override" points to the "Check Count" field, which is set to "1 (default)".

State:	Enabled
Service:	YARN
Component:	ResourceManager
Type:	WEB
Groups:	YARN Default
Last Changed:	Fri, May 27, 2016 0
Check Count:	1 (default)

Storm Monitoring View

EXECUTOR



0

TASKS



0

SUPERVISOR

100 %

SLOTS

0 %

Slots are Workers (processes).

Nimbus Summary

Host:Port	Status	Uptime
c6405.ambari.apache.org:6627	Leader	5m 32s

Topology Listing

Topology Name	Status	Uptime
No topology found !		

Supervisor Summary

Host	Slots	CPU	Memory	Uptime
c6406.ambari.apache.org	0 %	0 %	0 %	4m 18s
c6404.ambari.apache.org	0 %	0 %	0 %	4m 53s
c6405.ambari.apache.org	0 %	0 %	0 %	5m 3s

Nimbus Configuration

Search By Key



Key	Value
client.jartransformer.class	org.apache.storm.hack.StormShadeTransformer
drpc.invocations.port	3773
logviewer.max.per.worker.logs.size.mb	2048
metrics.reporter.register	org.apache.hadoop.metrics2.sink.storm.StormTimelineMetricsReporter



Grafana for Ambari Metrics



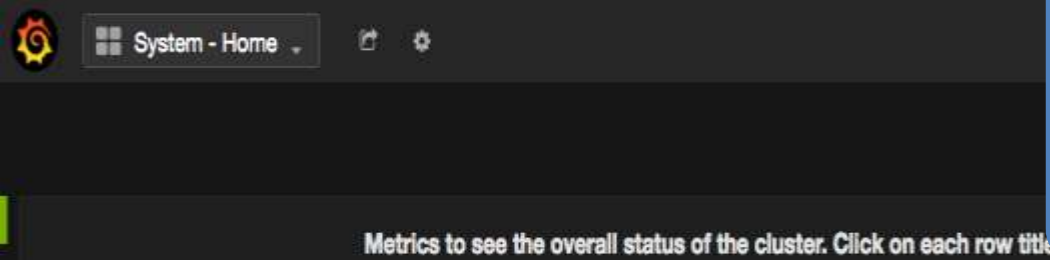
FEATURES

- Grafana as a “Native UI” for Ambari Metrics
- Pre-built Dashboards Host-level, Service-level
- Supports HTTPS

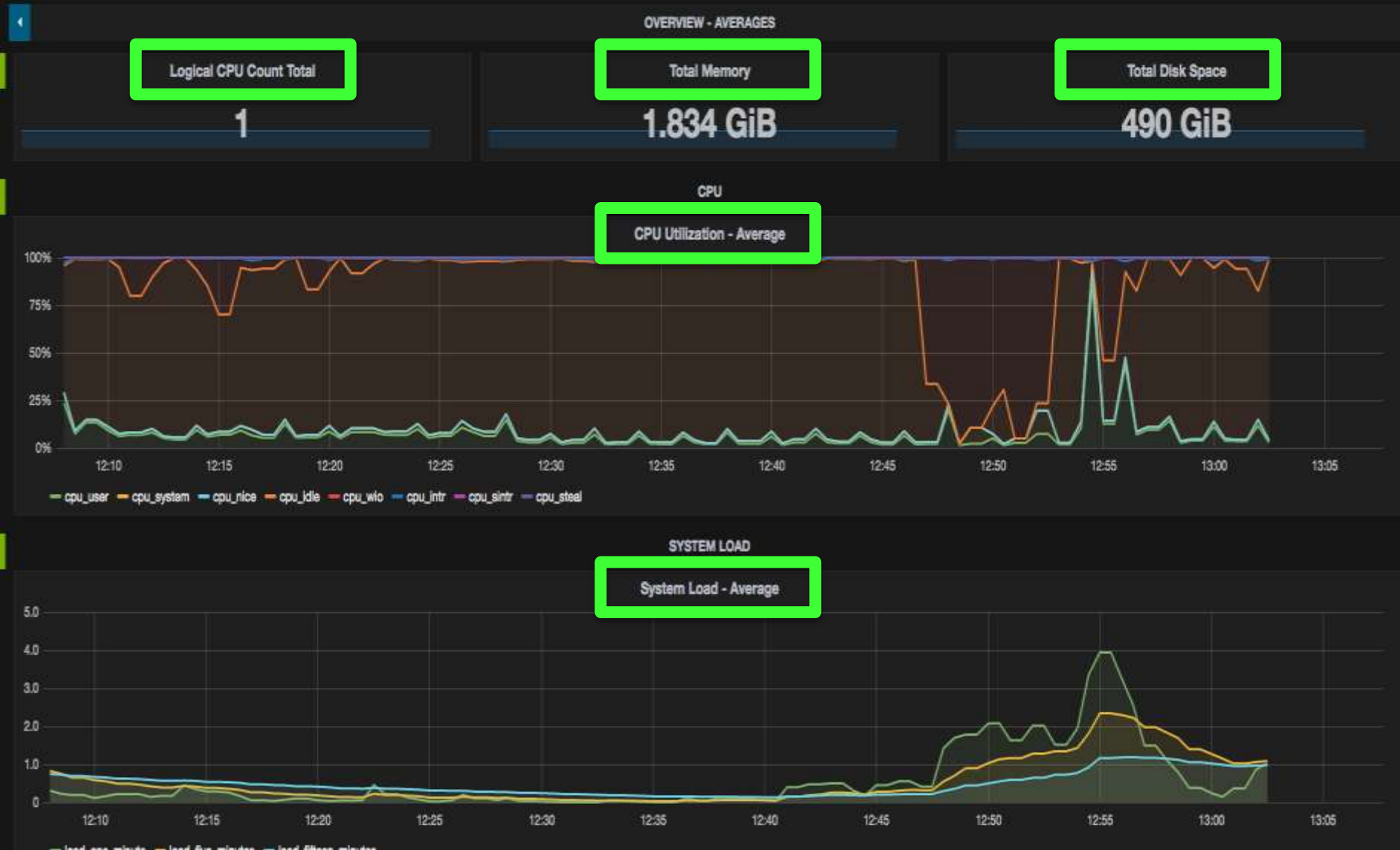


DASHBOARDS

- System Home, Servers
- HDFS Home, NameNodes, DataNodes
- YARN Home, Applications, Job History Server
- HBase Home, Performance



Grafana includes pre-built dashboards for visualizing the most important cluster metrics.





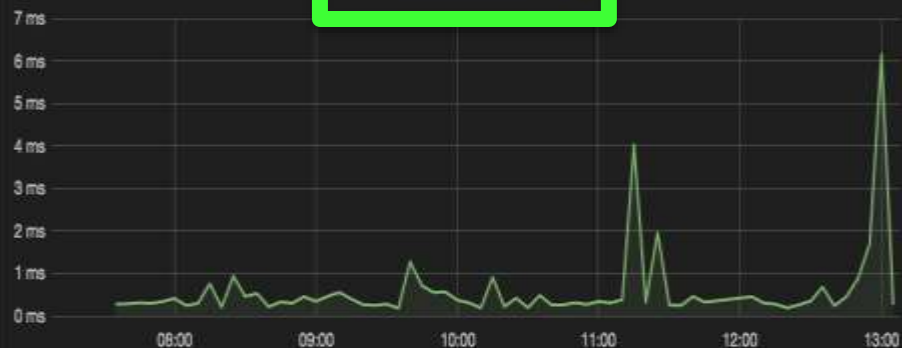
namenode - hosts: All

The HDFS NameNode dashboard highlights file system activity.

Metrics to see the status for the Namenodes on the HDFS cluster. Click on each

RPC CLIENT QUEUE TIME

RPC Client Port Queue Time

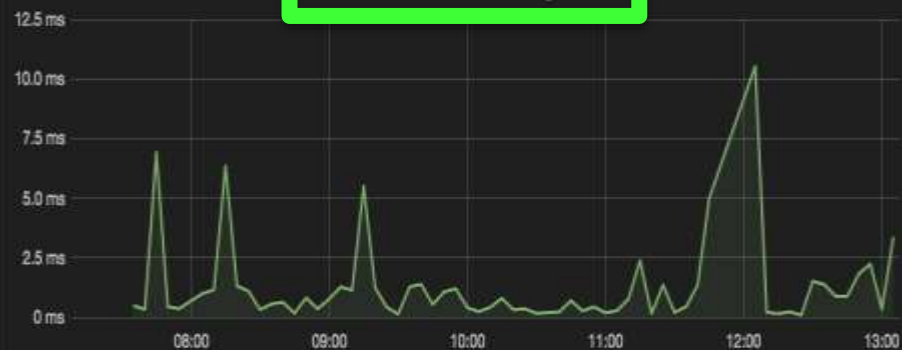


RPC Client Port Queue Num Ops



RPC CLIENT PORT PROCESSING TIME

RPC Client Port Processing Time



RPC Client Port Processing Num Ops



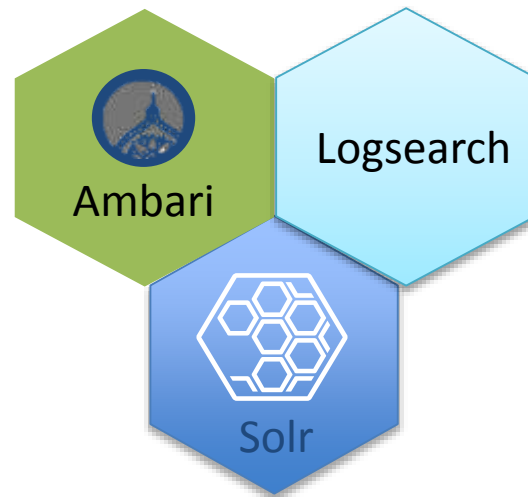
GC COUNT & TIME

GC PAR NEW

GC EXTRA SLEEP & WARNING THRESHOLD EXCEEDED

Log Search

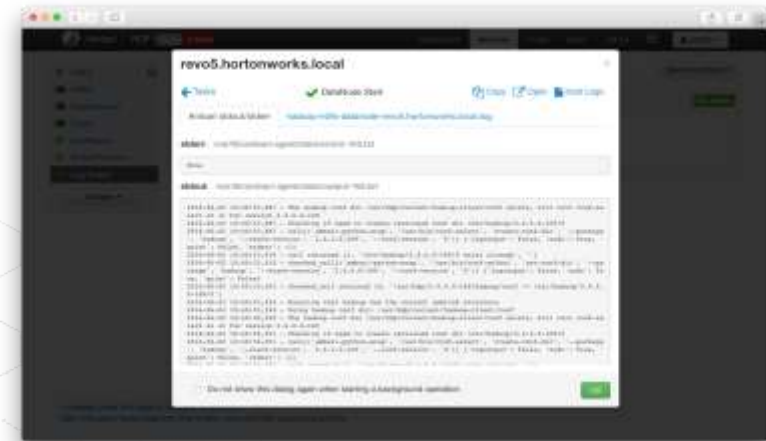
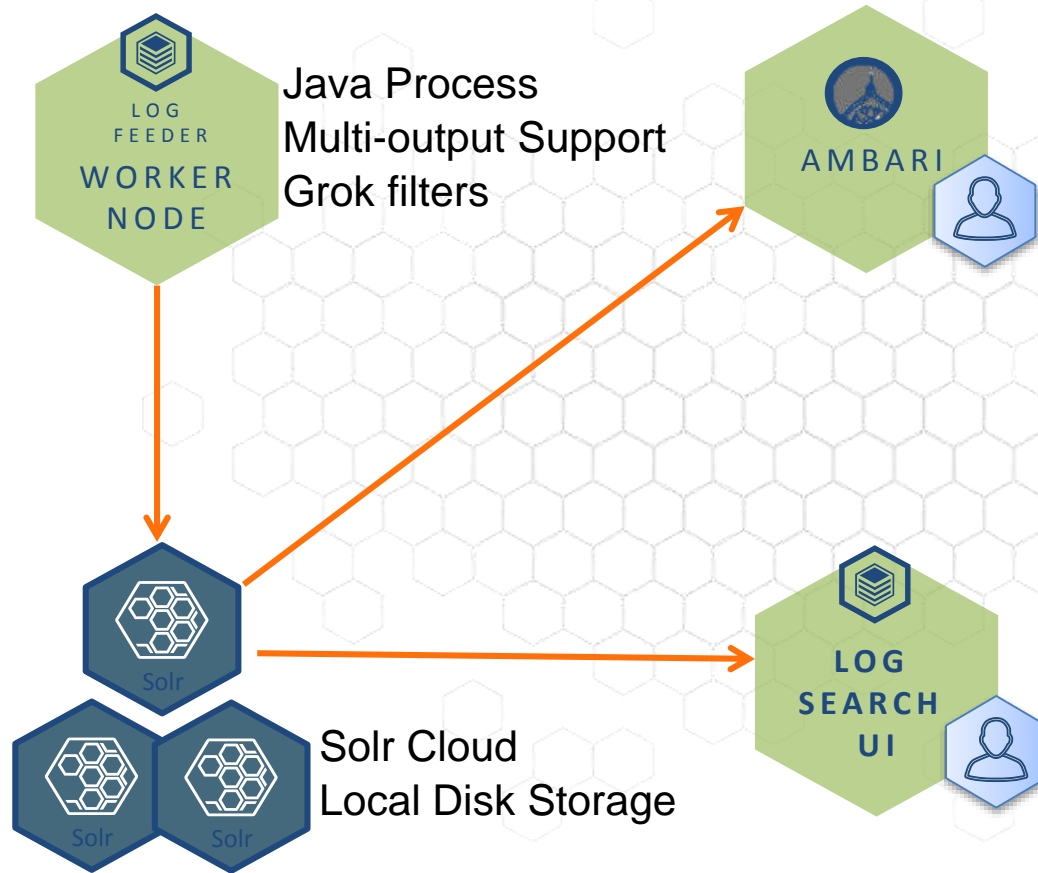
Search and index HDP logs!



Capabilities

- Rapid Search of all HDP component logs
- Search across time ranges, log levels, and for keywords

Log Search



Future of Ambari

- Cloud features
- Service multi-instance (two ZK quorums)
- Service multi-versions (Spark 1.6 & Spark 2.0)
- YARN assemblies
- Patch Upgrades: upgrade individual components in the same stack version, e.g., just DN and RM in HDP 2.5.*.* with zero downtime
- Ambari High Availability

As good as

