Streamline Hadoop DevOps with Apache Ambari

Jayush Luniya

Hadoop Summit, Tokyo

HORTONWORKS®
POWERING THE FUTURE OF DATA™

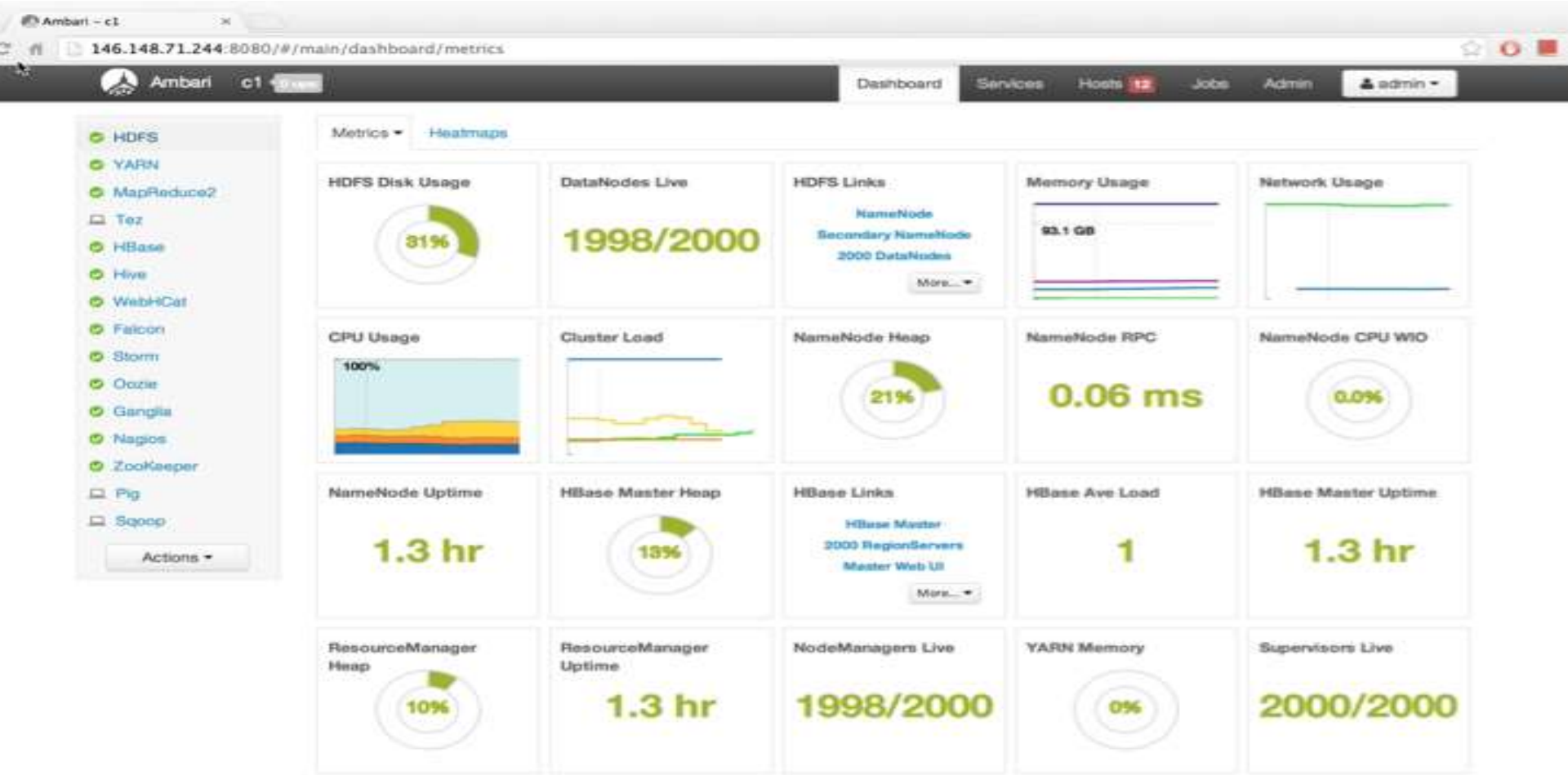
Speaker

Jayush Luniya

Staff Software Engineer @ Hortonworks
Apache Ambari PMC
jluniya@apache.org

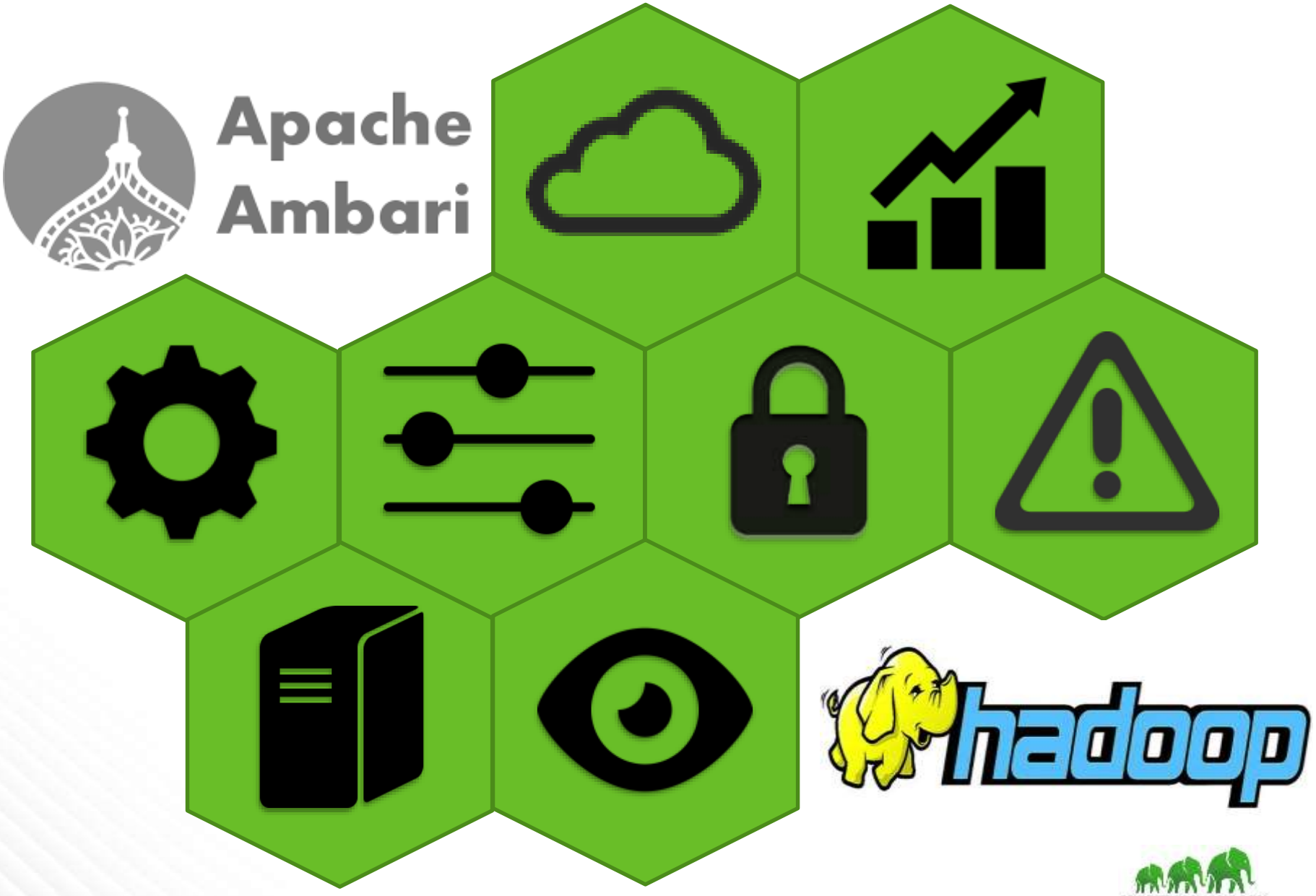
Apache Ambari

Open-source platform to provision, manage and monitor Hadoop clusters

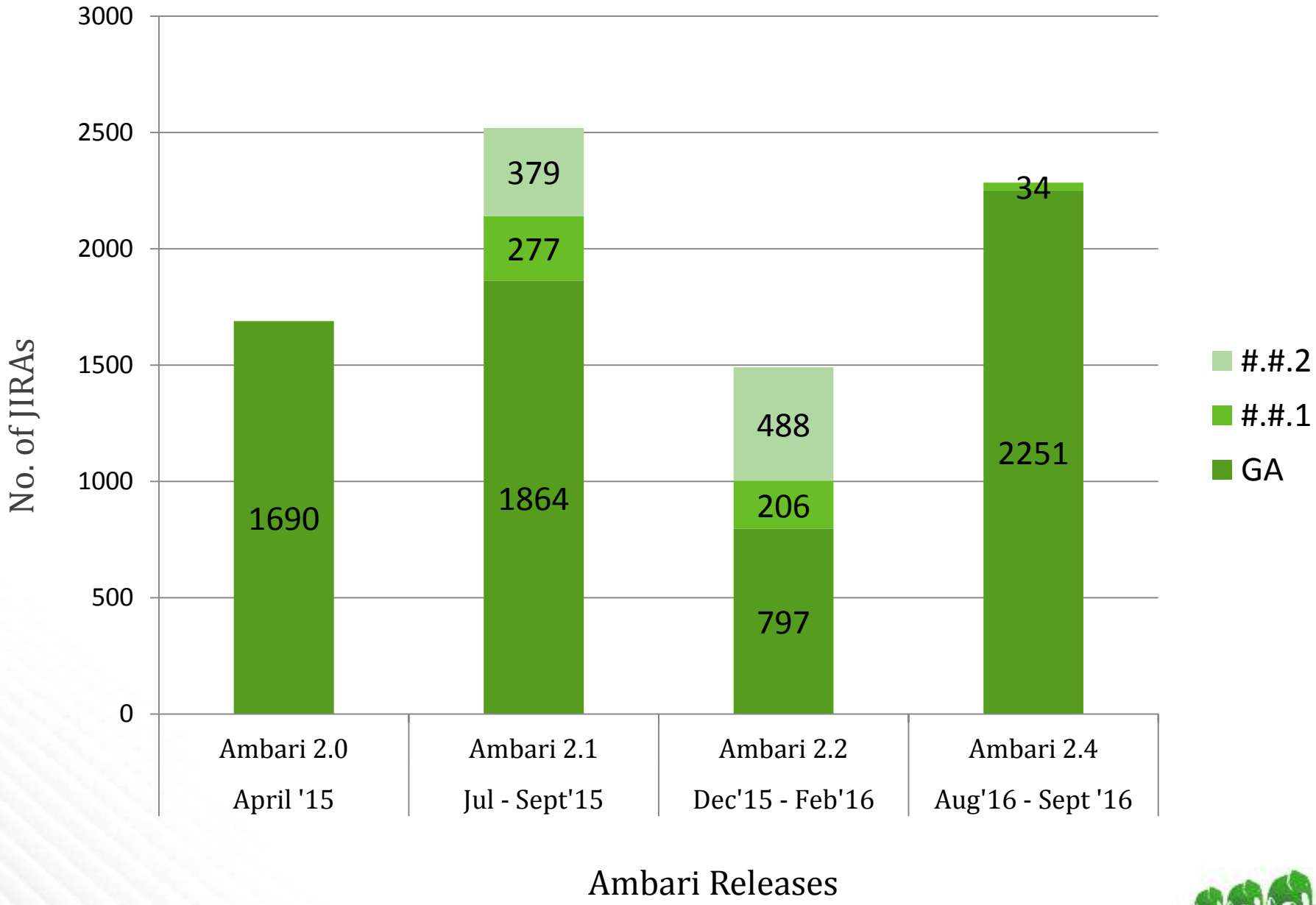


Why Ambari?

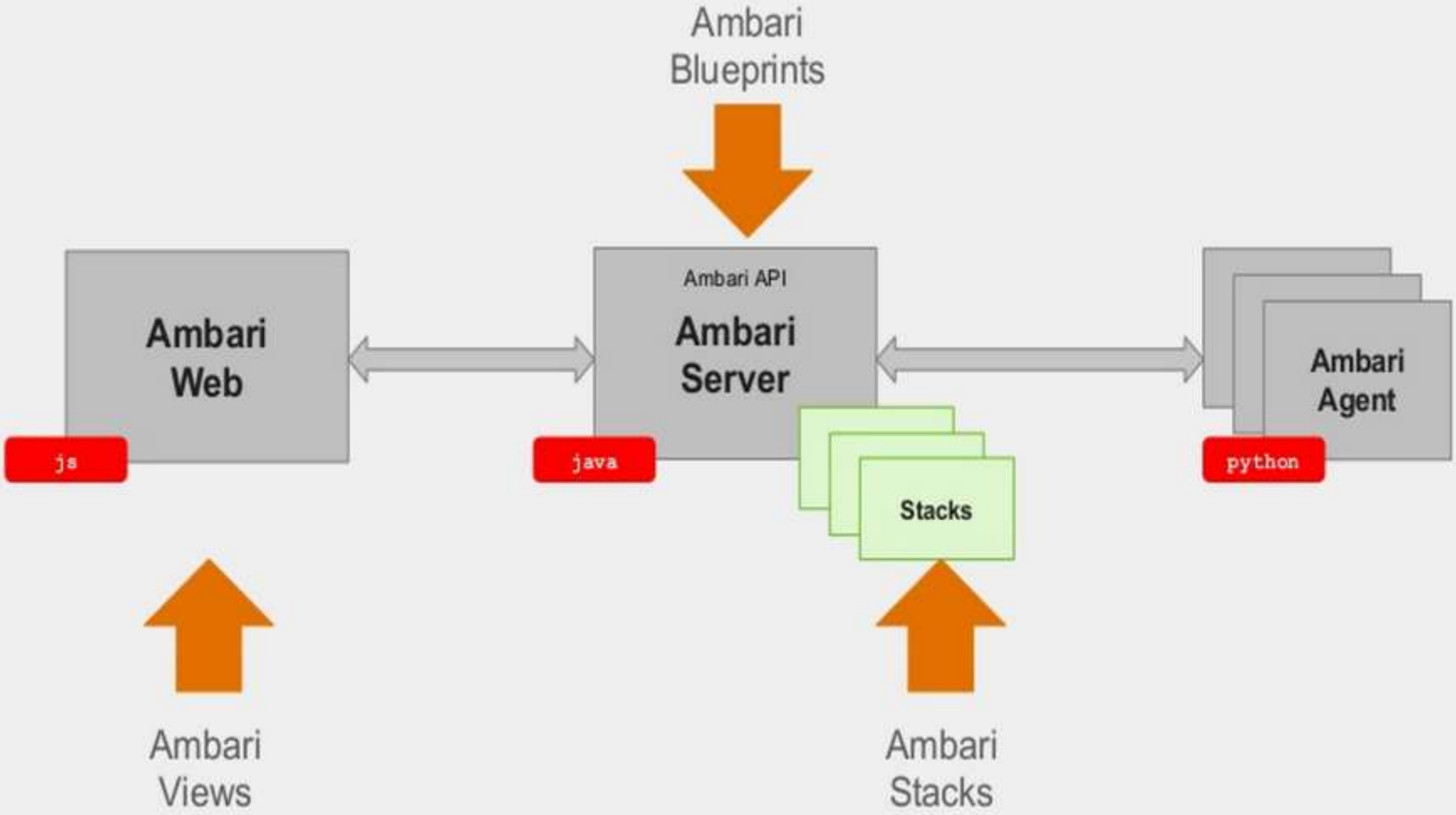
Why Ambari?



HORTONWORKS



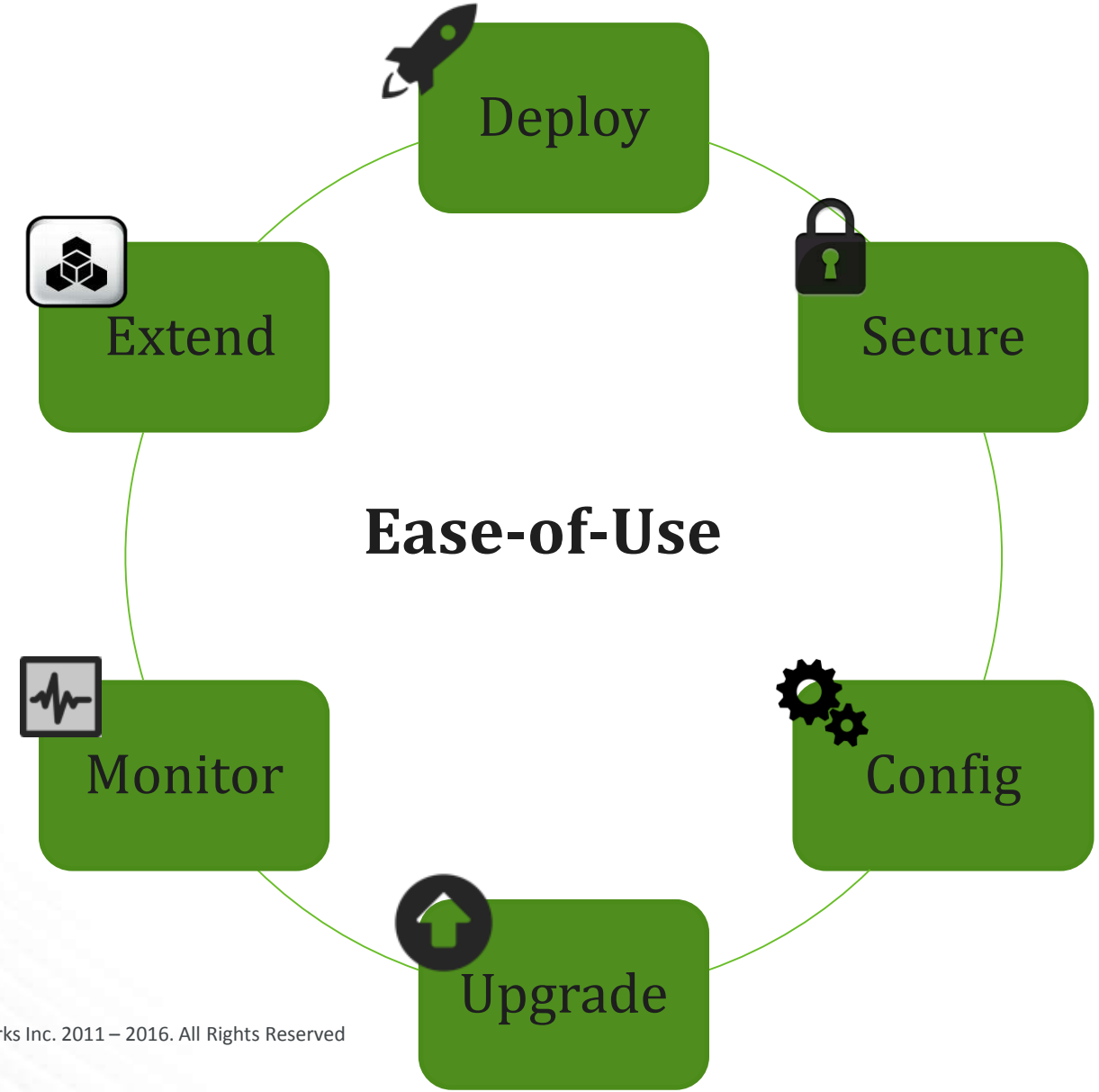
Ambari Architecture



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

Operations - Lifecycle



Deploy

Deploy On Premise

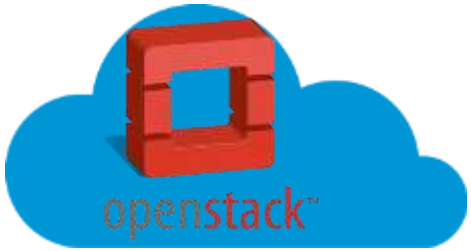
Ambari handles all of these combinations and makes recommendations based on host specs.

Deploy In The Cloud

Certified environments

Sysprepped VMs

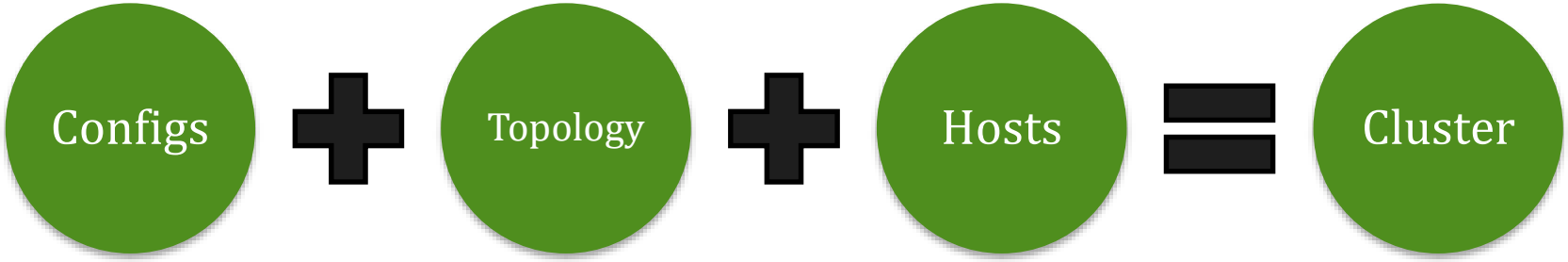
Hundreds of similar clusters



HORTONWORKS

Deploy with Blueprints

- Systematic way of defining a cluster



- Export existing cluster into blueprint
  /api/v1/clusters/:clusterName?format=blueprint

Create 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 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 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 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"
      }
    ]
  }
]
```

Secure

Comprehensive Security

Kerberos

- MIT KDC
- Keytab Management

LDAP/AD

- User Auth.
- Sync

Ranger

- Security policies
- Audit
- Authorization

Atlas

- Governance
- Compliance
- Data Classify
- Lineage & History

Knox

- Perimeter Sec.
- LDAP/AD
- Sec. REST/HTTP
- SSL

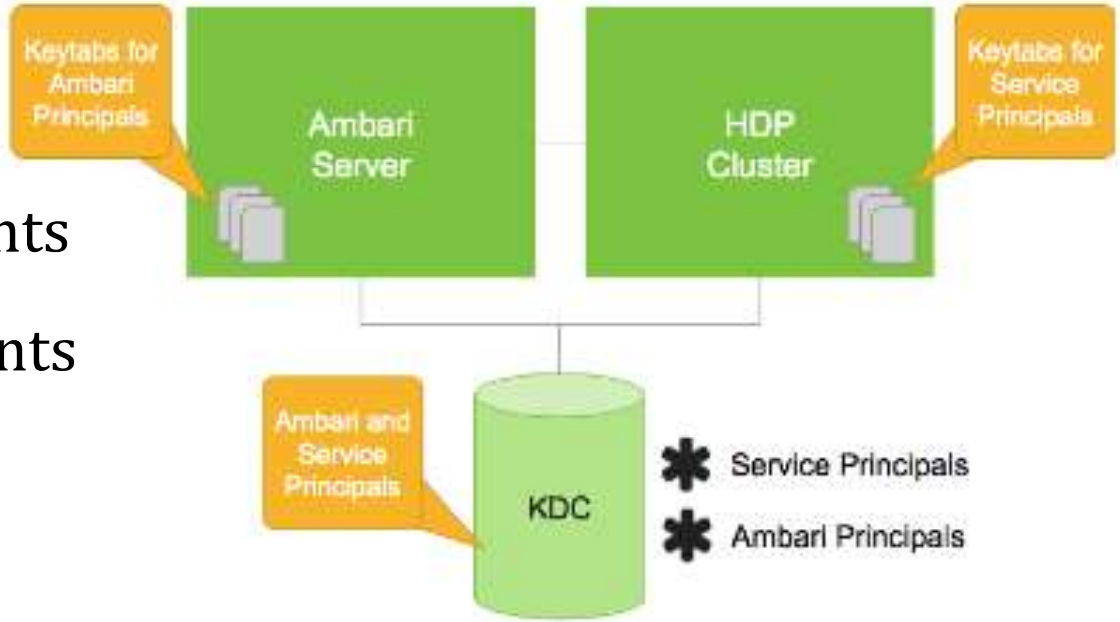
Kerberos

Ambari manages Kerberos principals and keytabs

Works with existing MIT KDC or Active Directory

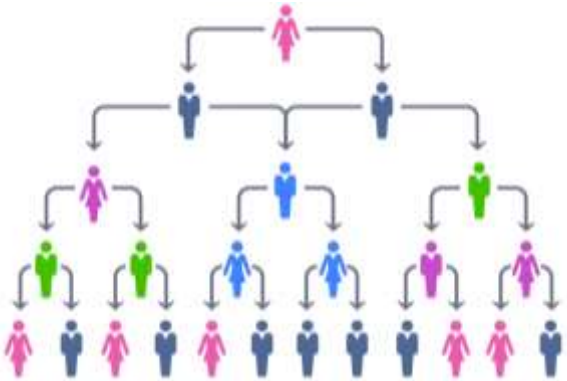
Once Kerberized, handles

- Adding Services
- Adding Hosts
- Adding Host Components
- Moving Host Components



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

- **Ambari Admin** — all
  - **Cluster Admin** — ↑, except manage permissions
    - **Cluster Op** — ↑, except add services, Kerberos, manage alerts & upgrades
      - **Service Admin** — ↑, except alter cluster topology or install components
        - **Service Op** — ↑, except change configs
          - **Read-Only** — only view

Config

Config Management

- Config Groups
  - Different config settings for individual host components
- Config Versioning
  - Revert back to old configs
- Smart Configs
  - Highlight most important configs
- Stack Advisor
  - Recommend configurations

Smart Configs

Settings | Advanced

Server

HBase Master Maximum Memory
1GB
1 GB 2 GB 2.75 GB

HBase RegionServer Maximum Memory
1GB
1 GB 1.75 GB 2.25 GB

% of RegionServer Allocated to Read Buffers
40%
0 % 40 % 80 %

% of RegionServer Allocated to Write Buffers
40%
0 % 40 % 80 %

Memstore Flush Size
128MB
32 MB 144 MB 256 MB

HBase Region Block Multiplier
4

Number of Handlers per RegionServer
30
5 123 240

Client

Maximum Client Retries
35
5 28 50

Maximum Record Size
1MB
1 MB 15.5 MB 30 MB

Widgets

- Sliders
- Combos
- Toggles
- Spinners
- Lists

Disk

Maximum Region File Size
10GB
1 GB 51 GB 100 GB

HBase Region Major Compaction Interval
7 Days
0 Hours

Maximum Files for Compaction
10

Timeouts

Zookeeper Session Timeout
1 Minutes
30 Seconds

HBase RPC Timeout

Security

Enable Authentication
Simple

Enable Authorization

Phoenix SQL

Enable Phoenix
Disabled

Phoenix Query Timeout

Stack Advisor

Configurations

Kerberos
HTTPS
Zookeeper Servers
Memory Settings
…
High Availability

Example

# Atlas Servers
atlas.enabletTLS = true|false
atlas.server.http.port = 21000
atlas.server.https.port = 21443

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

Upgrade

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

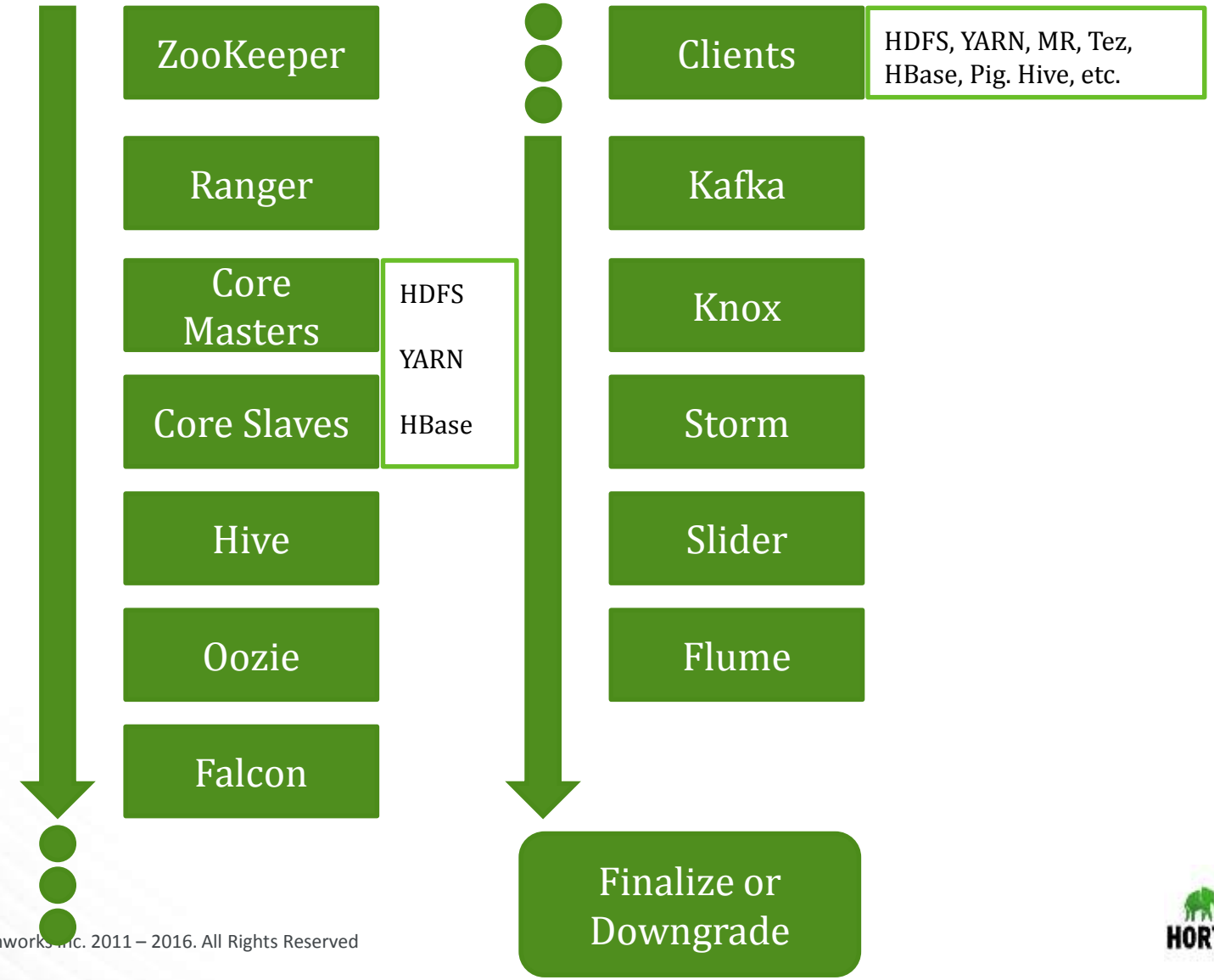
Manual Upgrade

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

Automated Upgrade: Rolling/Express

Check Prerequisites	Prepare	Register + Install	Perform Upgrade	Finalize
Review the **prereqs** to confirm your cluster configs are ready	Take **backups** of critical cluster metadata	**Register** the HDP repository and **install** the **target** HDP version on the cluster	Perform the HDP upgrade. The steps depend on upgrade method: **Rolling** or **Express**	Finalize the upgrade, making the **target** version the **current** version

Process: Rolling Upgrade



Monitor

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 UI

Ambari cl1 0 ops 0 alerts | Dashboard | Services | Hosts | Alerts | Admin | admin

DataNode Storage

Back

OK

Configuration

Edit

Description: This host-level alert is triggered if storage capacity if full on the DataNode. It checks the DataNode JMX Servlet for the Capacity and Remaining properties. The threshold values are in percent.

Check Interval: 2 Minute

Thresholds:

OK	Remaining Capacity:[{0}], Total Capacity:[{2:.0f}% Used, {1}]		
WARNING	75	%	Remaining Capacity:[{0}], Total Capacity:[
CRITICAL	80	%	Remaining Capacity:[{0}], Total Capacity:[

Connection Timeout: CRITICAL 5 Seconds

State:	Enabled
Service:	HDFS
Component:	DataNode
Type:	METRIC
Groups:	HDFS Default
Last Changed:	Wed, Oct 26, 2016 07:59
Check Count:	1 (default)

Instances

Service	Host	Status	24-Hour	Response
All	Any	All		
HDFS	jayush-demo-3.c.pramod-thangali.internal	OK for 23 hours	2	Remaining Capacity:[5367292928], Total Capacity:[41% Used, 91294...

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.

System - Home

Metrics to see the overall status of the cluster. Click on each row titl

OVERVIEW - AVERAGES

Logical CPU Count Total

1

Total Memory

1.834 GiB

Total Disk Space

490 GiB

CPU

CPU Utilization - Average

100%
75%
50%
25%
0%

12:10 12:15 12:20 12:25 12:30 12:35 12:40 12:45 12:50 12:55 13:00 13:05

cpu_user cpu_system cpu_nice cpu_idle cpu_wio cpu_intr cpu_sintr cpu_steal

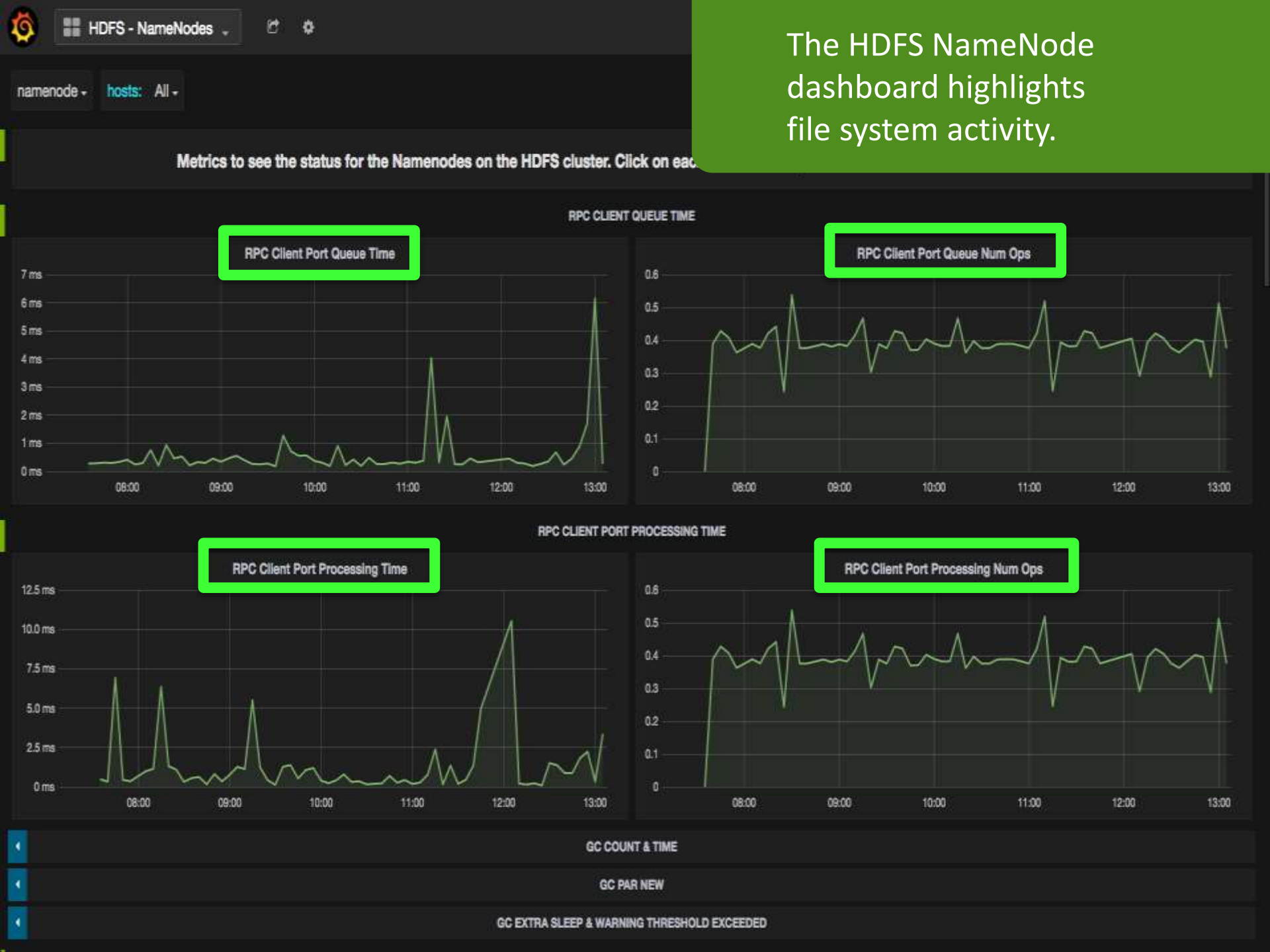
SYSTEM LOAD

System Load - Average

5.0
4.0
3.0
2.0
1.0
0

12:10 12:15 12:20 12:25 12:30 12:35 12:40 12:45 12:50 12:55 13:00 13:05

The HDFS NameNode dashboard highlights file system activity.



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

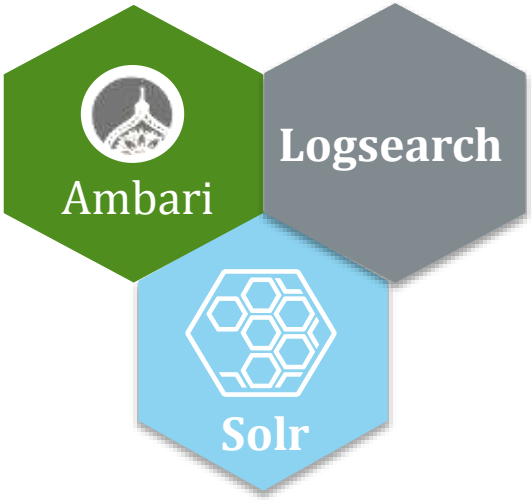
Nimbus Configuration

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

Log Search

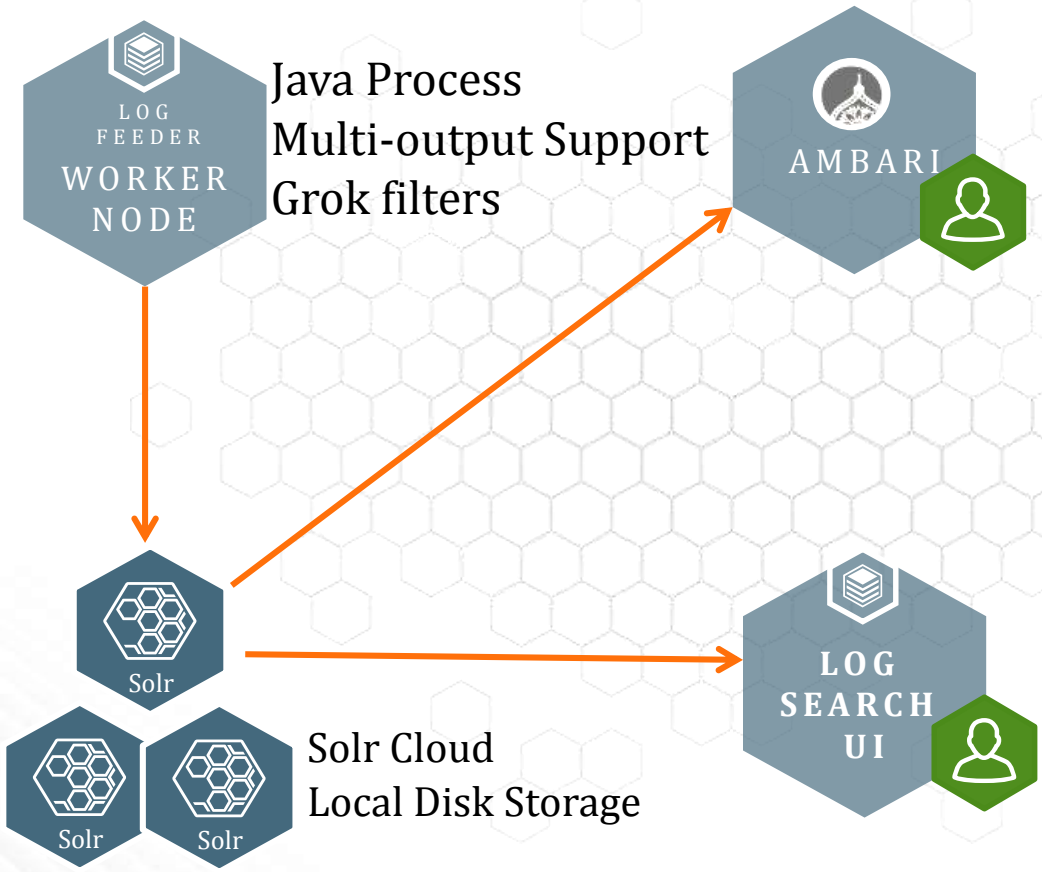
Search and index Hadoop logs!



Capabilities

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

Log Search



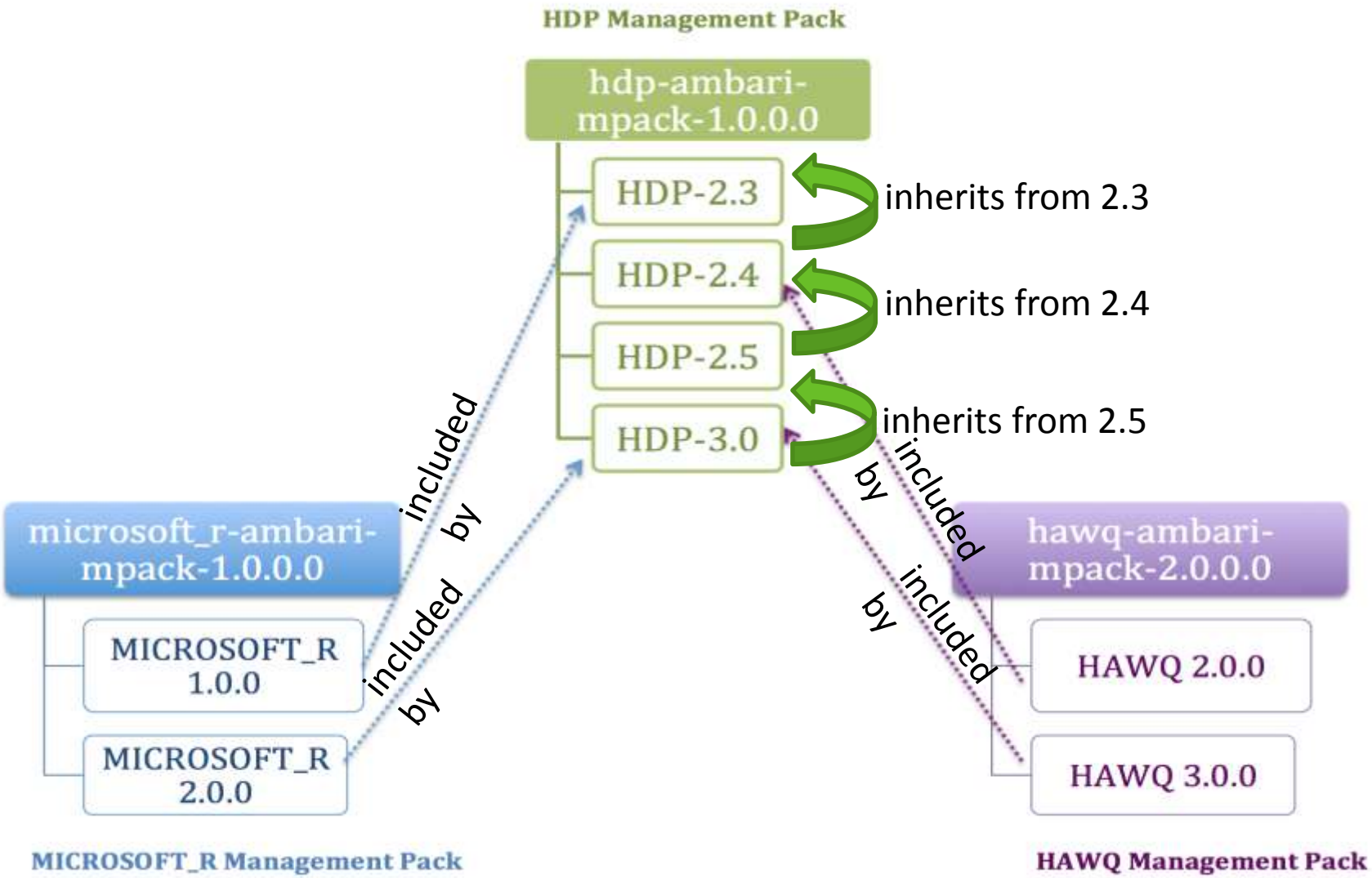
Extend

Management Packs

- Improved Release Management:
  Decouple Ambari core from stacks releases

- 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
- Include Views

Service Level Extensions

- Service Role Command Order
- Service Advisor
- Service Repos
- Service Upgrade Packs

Future

Future of Ambari

- Cloud Focus
- Multiple Service Instance (Two ZK quorums)
- Multiple Service Versions (Spark 1.6 & Spark 2.0)
- YARN Assemblies
- Granular Upgrades: Patch, Component, Service
- Ambari High Availability

Thank You

HORTONWORKS®
POWERING THE FUTURE OF DATA™