APACHE: BIG_DATA NORTH_AMERICA

Apache Ambari

Streamline Hadoop DevOps with Apache Ambari

Alejandro Fernandez

May 18, 2017

Speaker

Alejandro Fernandez

Staff Software Engineer @ Hortonworks

Apache Ambari PMC

alejandro@apache.org

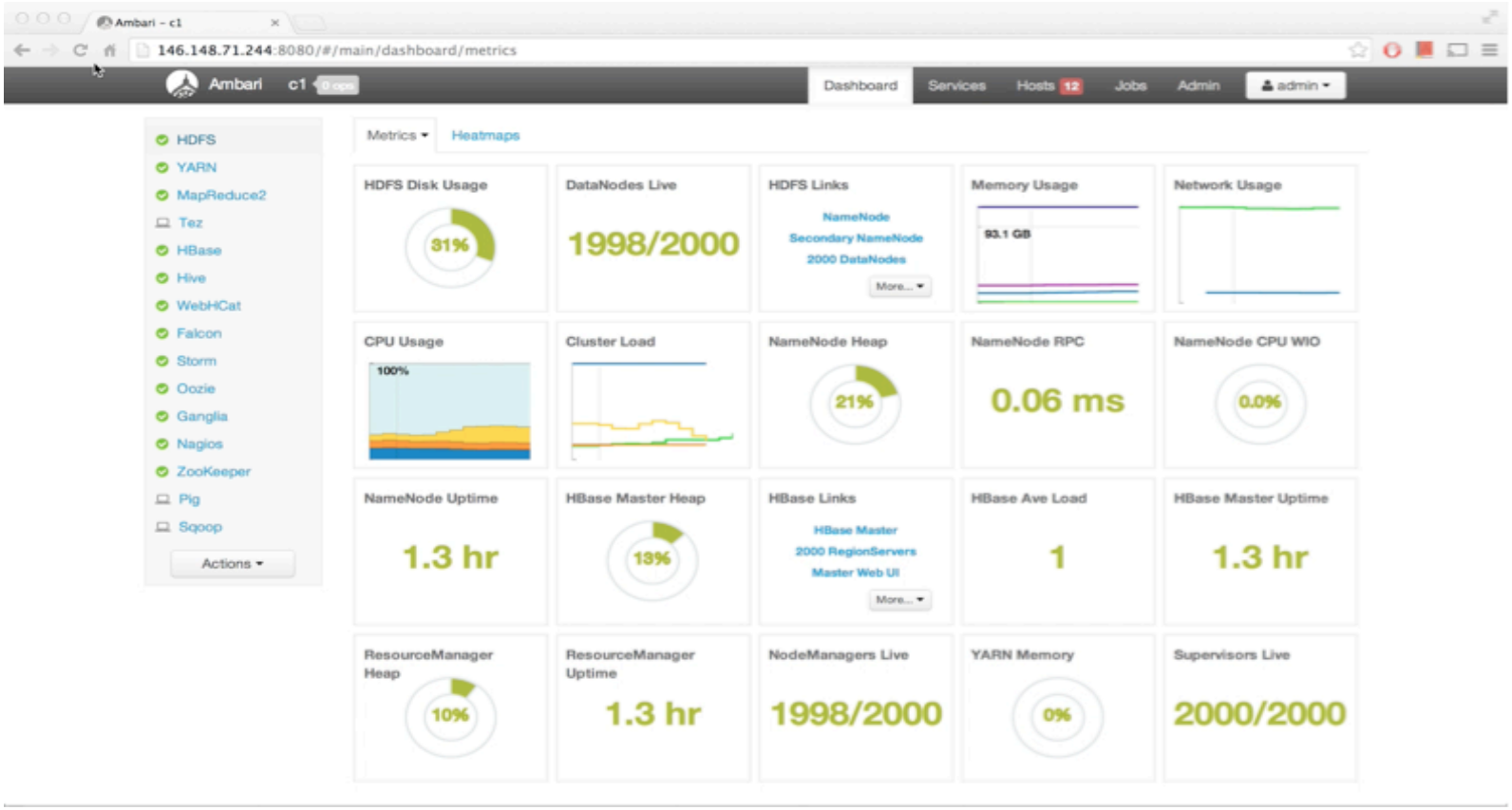
HORTONWORKS®

WHY ARE WE HERE?

“WORKING FROM MIAMI”

What is Apache Ambari?

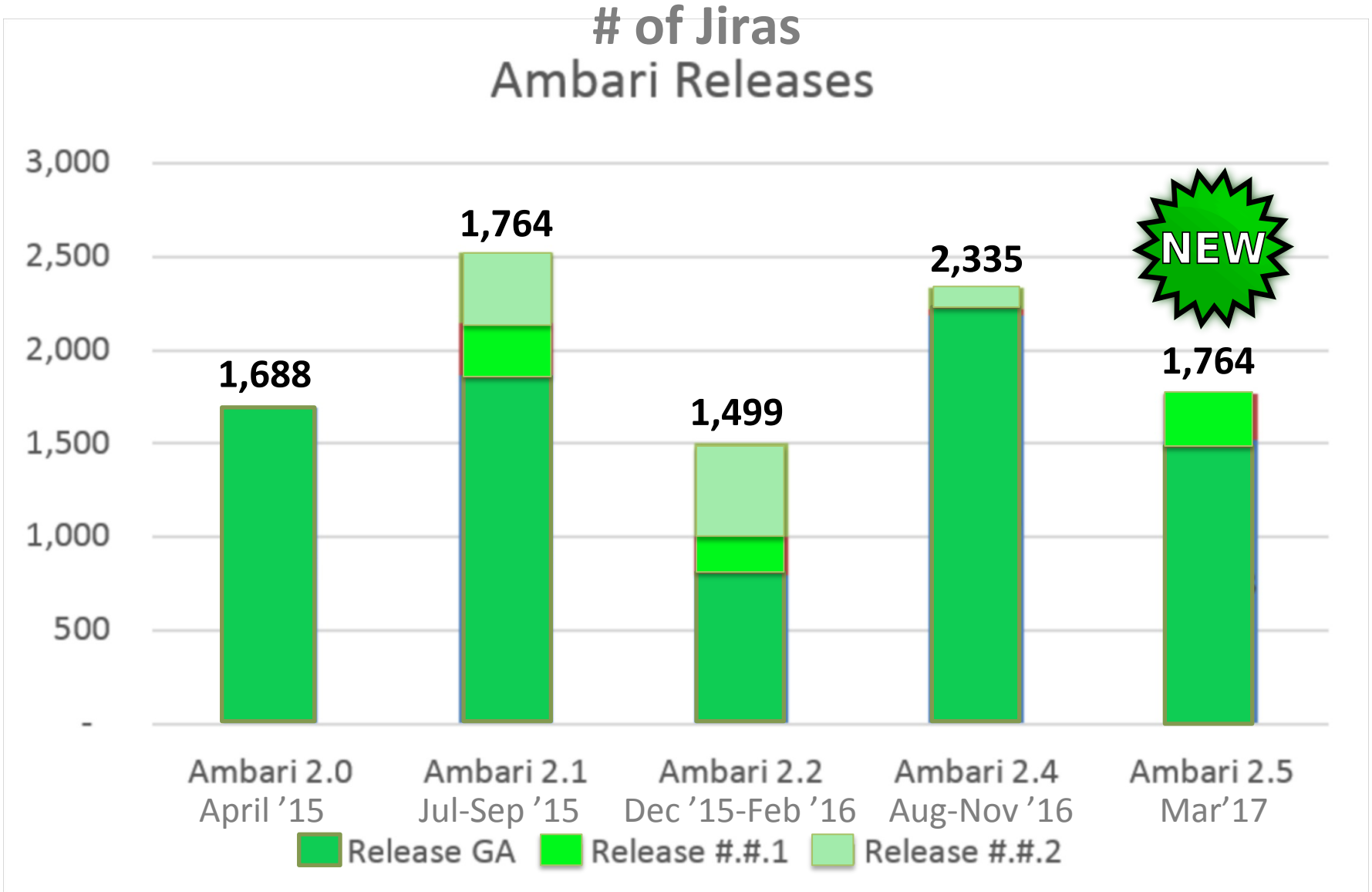
Apache Ambari is the open-source platform to deploy, manage and monitor Hadoop clusters



Single Pane of Glass for Hadoop

20.5k commits over 4.5 years by 80 committers/contributors

AND GROWING



Exciting Enterprise Features in Ambari 2.5

Service Features

AMBARI-2330: Service Auto-Restart

AMBARI-19275: Download All Client Configs

AMBARI-7748: Manage JournalNode HA

Security

AMBARI-18650: Password Credential Store

AMBARI-18365: API Authentication Using SPNEGO

Core

AMBARI-18731: Scale Testing on 2500 Agents

AMBARI-18990: Self-Heal DB Inconsistencies

Ambari Metrics System

AMBARI-17859: New Grafana dashboards

AMBARI-15901: AMS High Availability

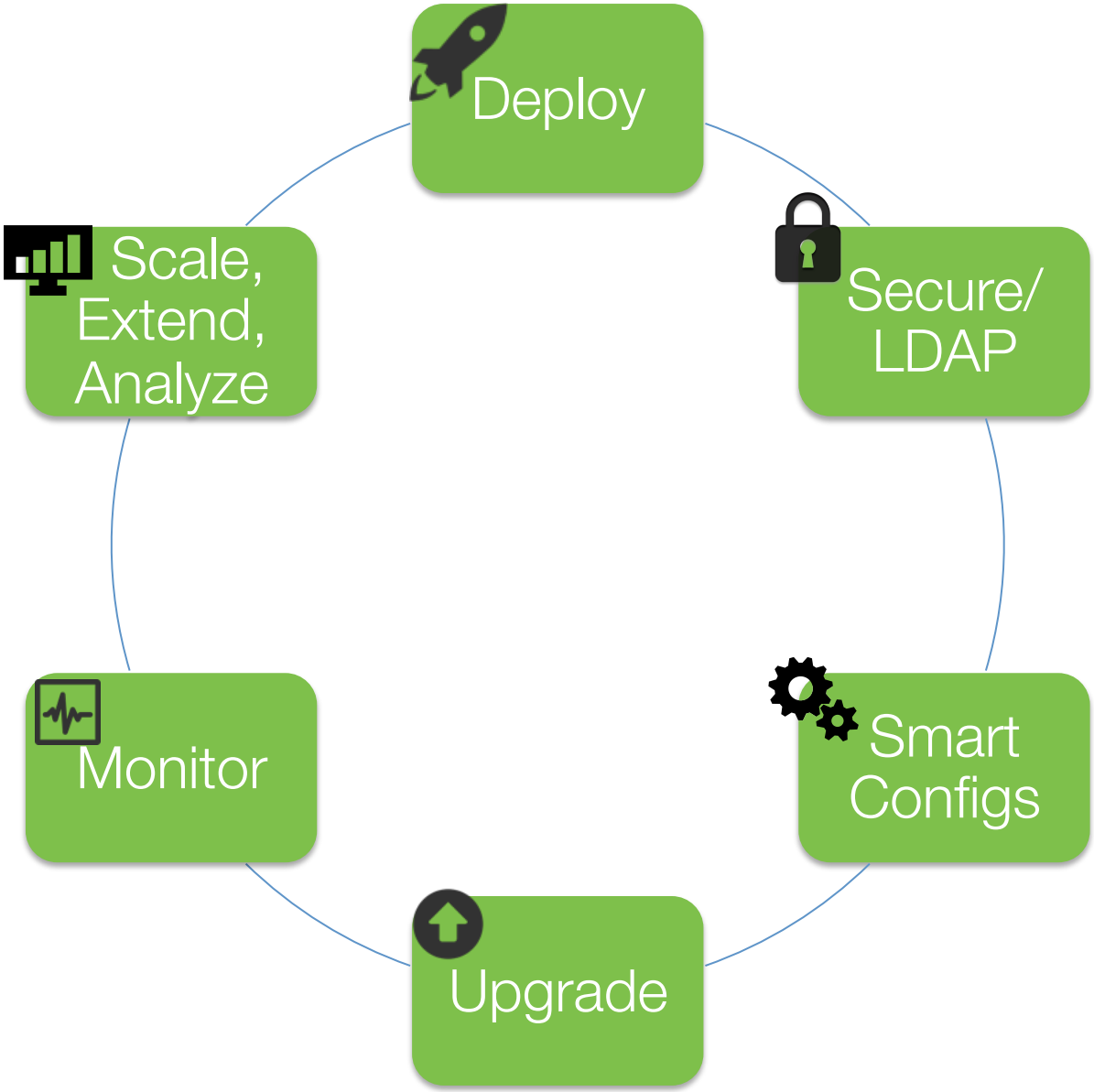
AMBARI-19320: HDFS TopN User and Operation Visualization

Alerts & Log Search

AMBARI-19257: Built-in SNMP Alert

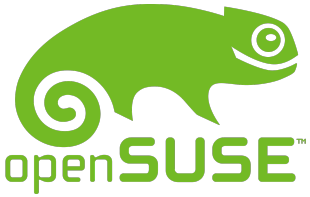
AMBARI-16880: Simplified Log Rotation Configs

Simply Operations - Lifecycle



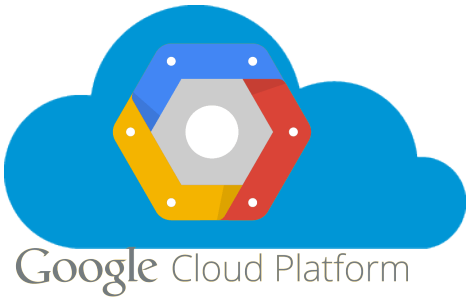
Deploy On Premise

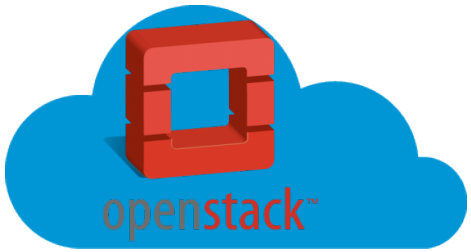
Mix-and-Match



Deploy On The Cloud

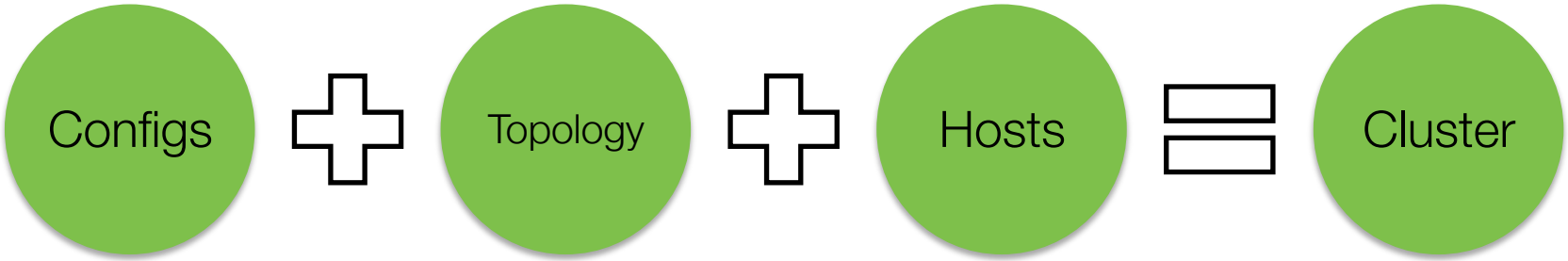
- Certified environments
- Sysprepped VMs
- Hundreds of similar clusters
- Ephemeral workloads





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

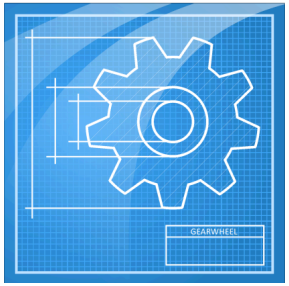
1. POST /api/v1/blueprints/my-blueprint

```
{
 "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

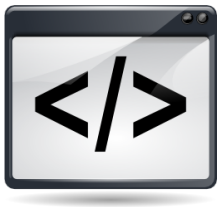


- **Kerberos**, secure out-of-the-box
- **High Availability** is setup initially for NameNode, YARN, Hive, Oozie, etc
- **Host Discovery** allows Ambari to automatically install services for a Host when it comes online
- **Stack Advisor** for config recommendations

Blueprint Host Discovery

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

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

Service Layout



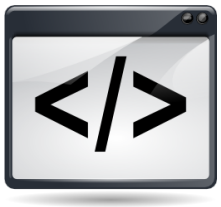
Common Services

- HDFS
  - 2.1.0.2.0
    - configuration
    - package
    - alerts.json
    - kerberos.json
    - metainfo.xml
    - metrics.json
    - widgets.json

Stack Override

- 2.3
  - repos
  - services
    - HDFS
      - configuration
        - hadoop-env.xml
        - hdfs-site.xml
        - ranger-hdfs-audit.xml
        - ranger-hdfs-plugin-properties.xml
        - ranger-hdfs-policymgr-ssl.xml
        - ranger-hdfs-security.xml
      - quicklinks
        - quicklinks.json
      - metainfo.xml
      - widgets.json

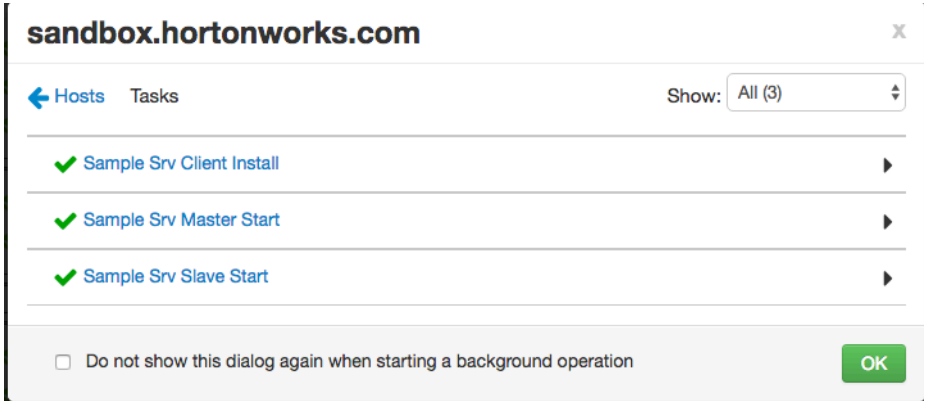
Custom Service



Starter Pack:

- metainfo.xml
- Python scripts: lifecycle management
- Configs: key, value, description, allow empty, password, etc.

- Templates: Jinja template with config replacement
- Role Command Order: dependency of start, stop commands
- Service Advisor: recommend/validate configs on changes
- Kerberos: principals and keytabs, configs to change when Kerberized
- Widgets: UI config knobs, sections
- Alerts: definition, type: [port, web, python script], interval
- Metrics: for Ambari Metrics System



Flink

cassandra

beam

APEX

Custom Service – metainfo.xml

metainfo.xml

```
<service>
    <name>SAMPLESRV</name>
    <displayName>New Sample Service</displayName>
    <comment>A New Sample Service</comment>
    <version>1.0.0</version>
    <components>
        <component>
            <name>SAMPLESRV_MASTER</name>
            <displayName>Sample Srv Master</displayName>
            <category>MASTER</category>
            <cardinality>1</cardinality>
            <commandScript>
                <script>scripts/master.py</script>
                <scriptType>PYTHON</scriptType>
                <timeout>600</timeout>
            </commandScript>
        </component>
        <component>
            <name>SAMPLESRV_SLAVE_OR_CLIENT</name>
            <displayName>Sample Slave or Client</displayName>
            <category>SLAVE | CLIENT</category>
            <cardinality>0+ | 0-1 | 1 | 1+</cardinality>
            <commandScript>
                <script>scripts/slave_or_client.py</script>
                <scriptType>PYTHON</scriptType>
                <timeout>600</timeout>
            </commandScript>
        </component>
    </components>
    ...
```

Custom Service – metainfo.xml

metainfo.xml

```
...
     <customCommand>
          <name>DECOMMISSION</name>
          <commandScript>
              <script>scripts/decommission.py</script>
              <scriptType>PYTHON</scriptType>
              <timeout>1200</timeout>
          </commandScript>
      </customCommand>

      <dependency>
          <name>HDFS/NAMENODE</name>
          <scope>cluster | host</scope>
          <auto-deploy>
              <enabled>true | false</enabled>
          </auto-deploy>
      </dependency>
...

<requiredServices>
  <service>HDFS</service>
</requiredServices>
```

Custom Service – metainfo.xml

metainfo.xml

```
...
      <configuration-dependencies>
          <config-type>service-env</config-type>
          <config-type>service-site</config-type>
          <config-type>hdfs-site</config-type>
      </configuration-dependencies>

      <osSpecifics>
          <osSpecific>
              <osFamily>any</osFamily>
              <packages>
                  <package>
                      <name>rpm_apt_pkg_name</name>
                  </package>
              </packages>
          </osSpecific>
      </osSpecifics>
```

Custom Service – Python Script

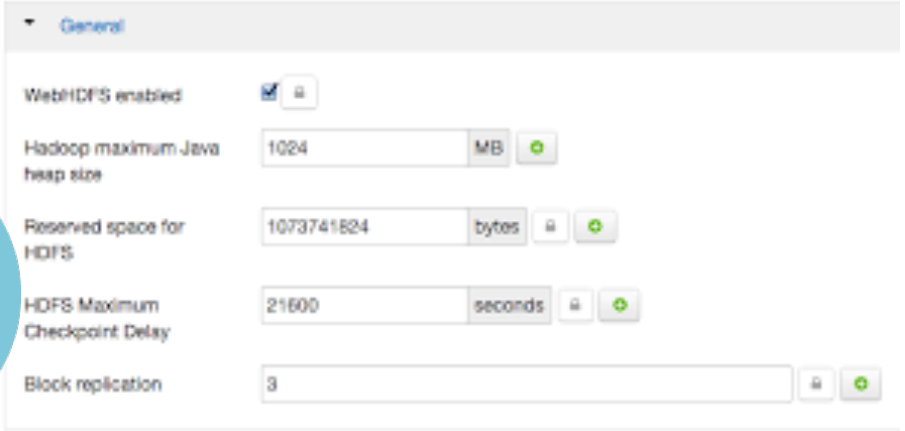
master.py

```
import sys
from resource_management import Script

class Master(Script):

  def install(self, env):
    print 'Install the Sample Srv Master'
  def stop(self, env):
    print 'Stop the Sample Srv Master'
  def start(self, env):
    print 'Start the Sample Srv Master'
  def status(self, env):
    print 'Status of the Sample Srv Master'
  def configure(self, env):
    print 'Configure the Sample Srv Master'

if __name__ == "__main__":
  Master().execute()
```

Stack Advisor

Configurations

Kerberos
HTTPS
Zookeeper Servers
Memory Settings
…
High Availability



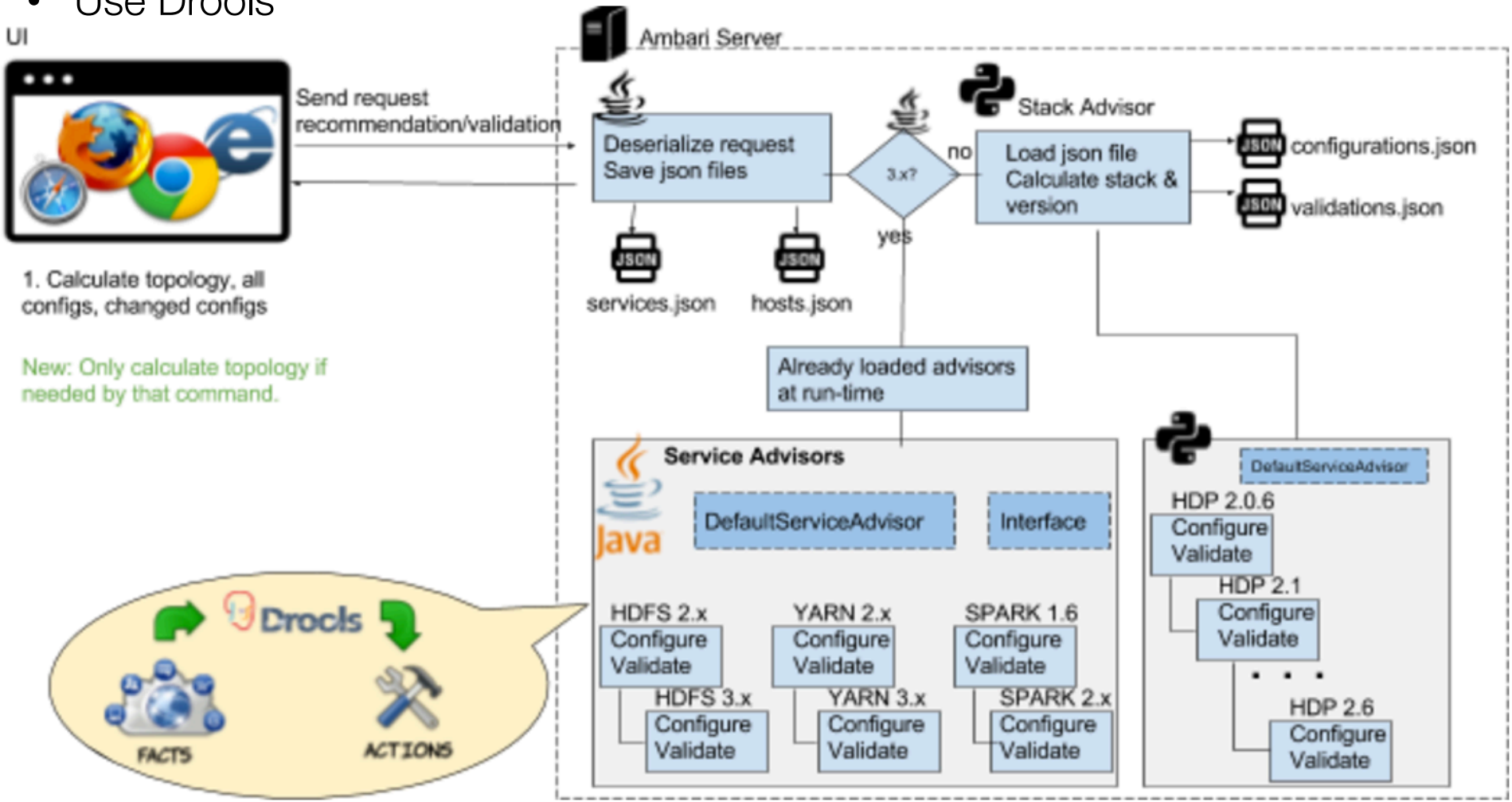
Example

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

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

Service Advisors in Ambari 3.0

- Break up single Stack Advisor into 22 Service Advisors
- Rewrite in Java for stronger checking and faster speed
- Use Drools



Comprehensive Security

Kerberos

- MIT KDC
- Keytab management

LDAP/AD

- User auth
- Sync

Ranger

- Security policies
- Audit
- Authorization

Atlas

- Governance
- Compliance
- Linage & history
- Data classification

Knox

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

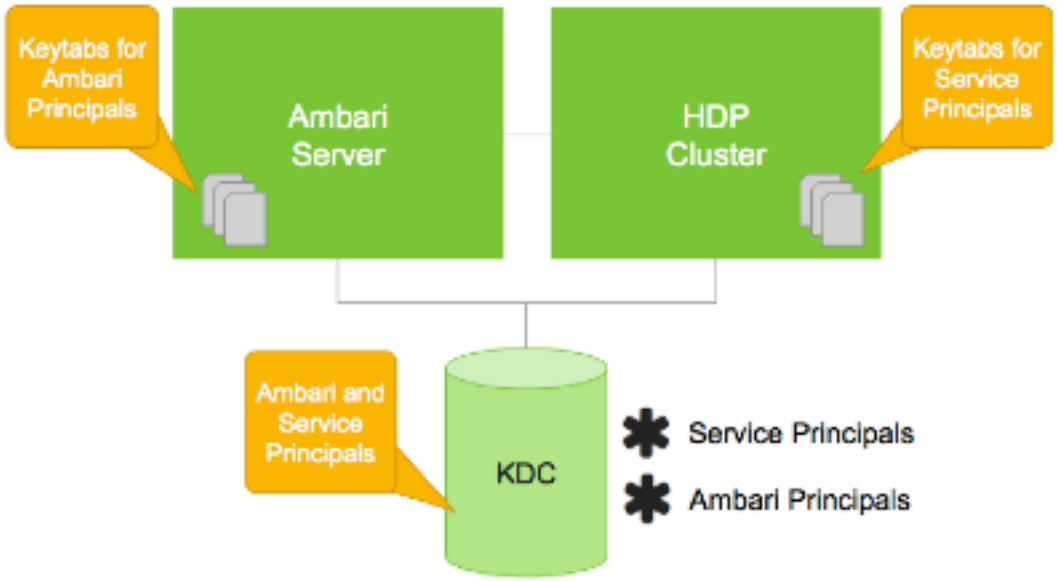
Kerberos

Ambari manages Kerberos principals and keytabs

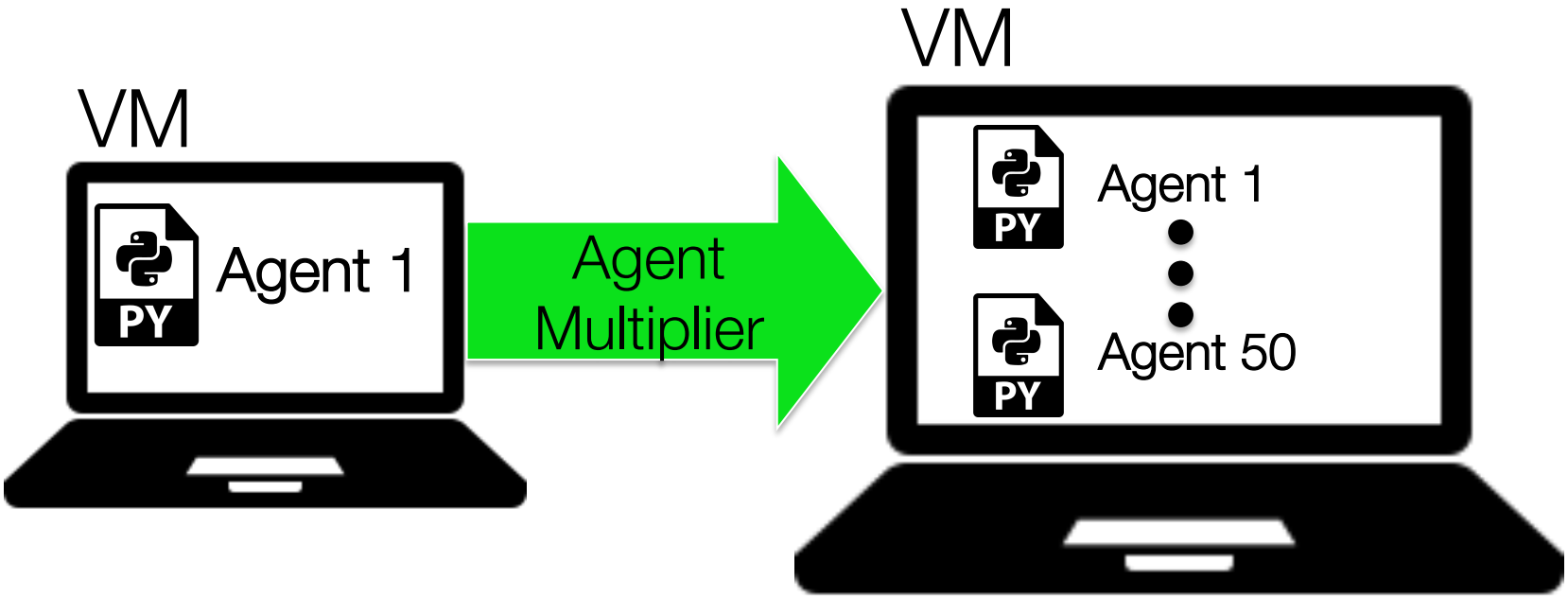
Works with existing MIT KDC or Active Directory

Once Kerberized, handles

- Adding hosts
- Adding components to existing hosts
- Adding services
- Moving components to different hosts



Testing at Scale: 3000 Agents



- Each Agent has own hostname, home dir, log dir, PID, ambari-agent.ini file
- Agent Multiplier can bootstrap 50 Agents per VM
- Tried Docker + Weave before and not very stable for networking

Testing at Scale: 3000 Agents

Testing

- ✓ Scale (server cannot tell the difference)
- ✓ Kerberos
- ✓ Stack Advisor
- ✓ Alerts
- ✓ Rolling & Express Upgrade
- ✓ UI

PERF Stack

Dummy Services

- Happy: always passes
- Sleepy: always timesout
- Grumpy: always fails
- Zookeeper
- HDFS
- YARN
- HBASE

Filter by host and component attributes or search by keyword ...

Name	IP Address	Rack	Cores	RAM	Disk Usage	Load Avg	Versions	Components
0001.ambari.apache.org	192.168.64.101	/default-rack	1 (1)	2.79GB			PERF-1.0	2 Components
0002.ambari.apache.org	192.168.64.101	/default-rack	1 (1)	2.79GB			PERF-1.0	2 Components
0003.ambari.apache.org	192.168.64.101	/default-rack	1 (1)	2.79GB			PERF-1.0	2 Components
0004.ambari.apache.org	192.168.64.101	/default-rack	1 (1)	2.79GB			PERF-1.0	2 Components
0005.ambari.apache.org	192.168.64.101	/default-rack	1 (1)	2.79GB			PERF-1.0	2 Components
0006.ambari.apache.org	192.168.64.101	/default-rack	1 (1)	2.79GB			PERF-1.0	2 Components
[illegible]	[illegible]	[illegible]	[illegible]	[illegible]			PERF-1.0	2 Components
[illegible]	[illegible]	[illegible]	[illegible]	[illegible]			PERF-1.0	2 Components
[illegible]	[illegible]	[illegible]	[illegible]	[illegible]			PERF-1.0	2 Components
[illegible]	[illegible]	[illegible]	[illegible]	[illegible]			PERF-1.0	2 Components

1 - 10 of 3000

Ambari c1 0 ops 0 alerts Dashboard Services Hosts Alerts Admin admin

Snow
Happy
Grumpy
Sleepy
Actions

Summary Configs Service Actions

Group Default (1) Manage Config Groups Filter...

V1 admin 6 minutes ago PERF-1.0

V1 admin authored on Fri, Oct 21, 2016 16:36 Discard Save

Settings Advanced

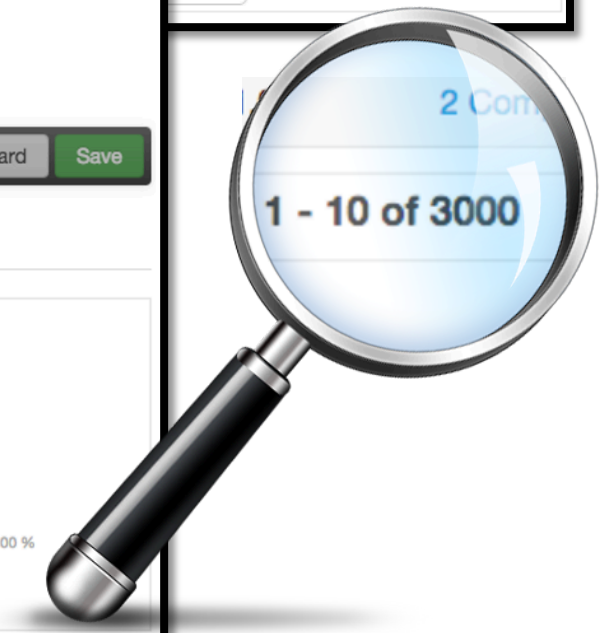
General

Features

Success percentage

100%

0 % 50 % 100 %



Optimize for Large Scale

- Dedicated database server with SSD
- MySQL 5.7 and DB tuning
- Purge old Ambari history: commands, alerts, BP topology, upgrades.

ambari-env.sh

```
export AMBARI_JVM_ARGS=$AMBARI_JVM_ARGS' -Xms2048m -Xmx8192m
```

ambari.properties

	10 Hosts	50 Hosts	100 Hosts	> 500 Hosts
agent.threadpool.size.max	25	35	75	100
alerts.cache.enabled	true			
alerts.cache.size	50000		100000	
alerts.execution.scheduler.maxThreads	2		4	

https://community.hortonworks.com/articles/80635/optimize-ambari-performance-for-large-clusters.html

Background: Upgrade Terminology

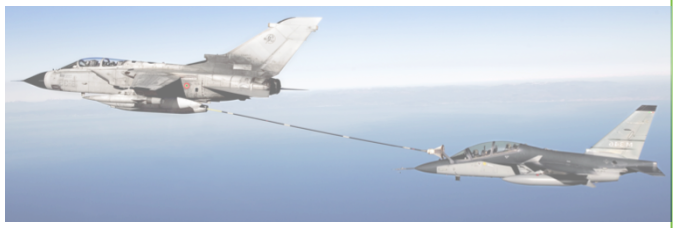
Manual Upgrade

- 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

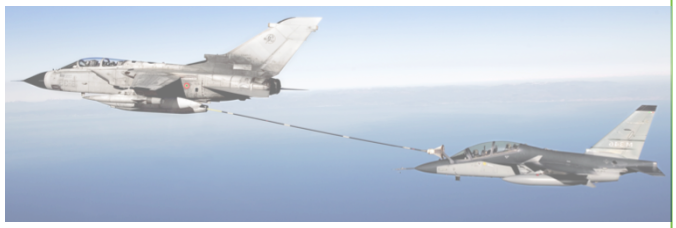
Background: Upgrade Terminology

Express Upgrade

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

Rolling Upgrade

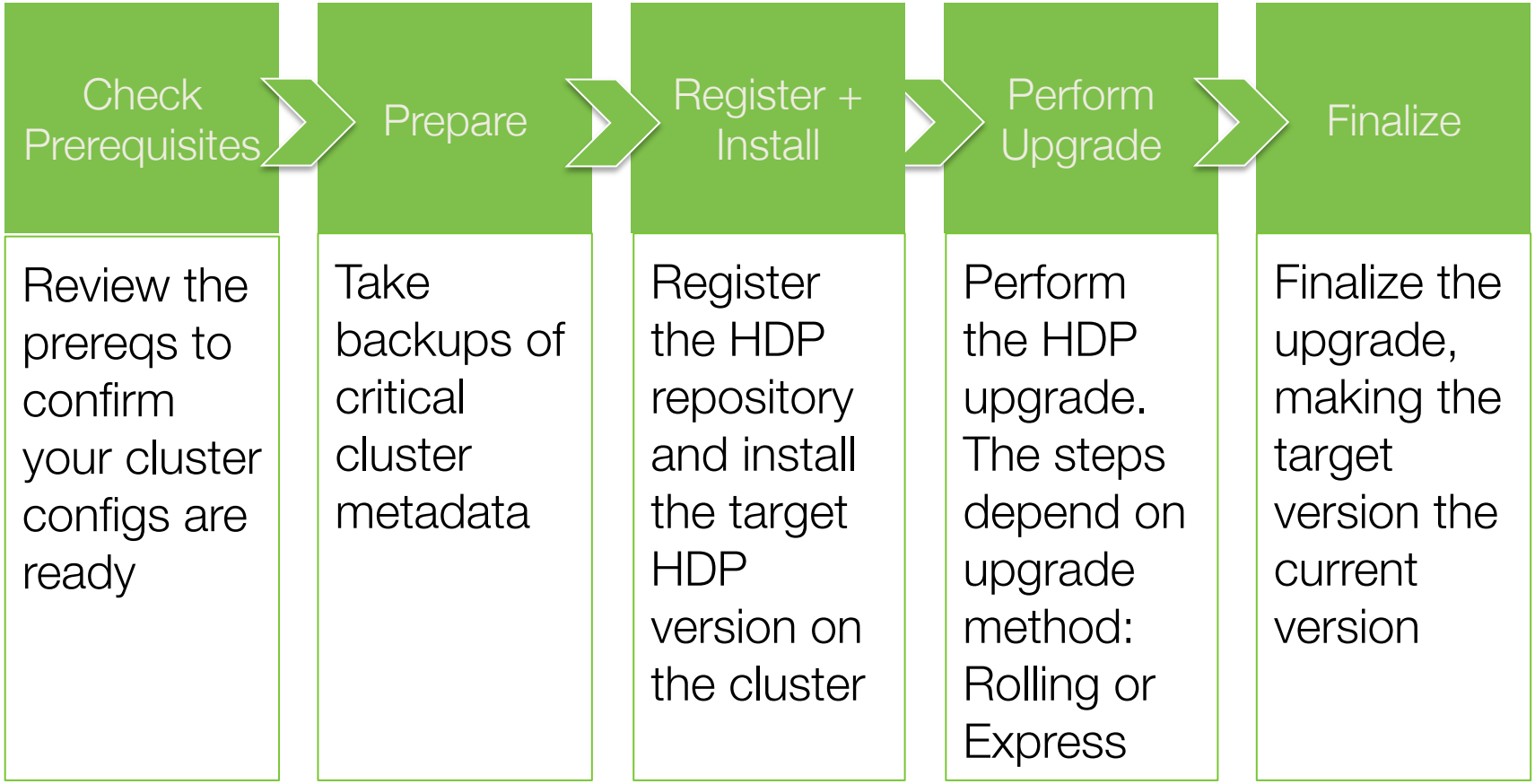
- ✓ Automated
- ✗ Upgrades one component per host at a time
- ✓ Preserves cluster operation and minimizes service impact



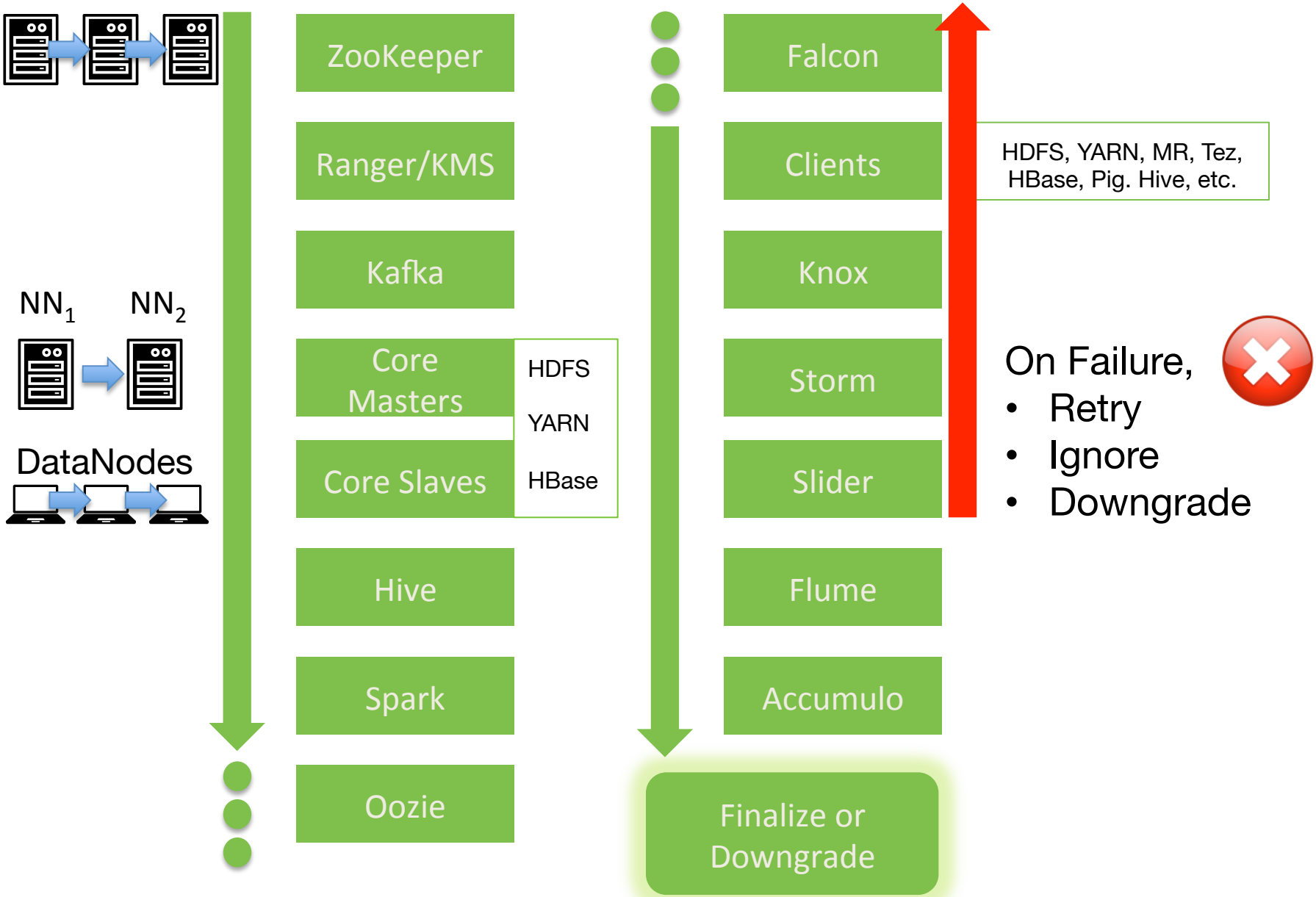
Manual Upgrade

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

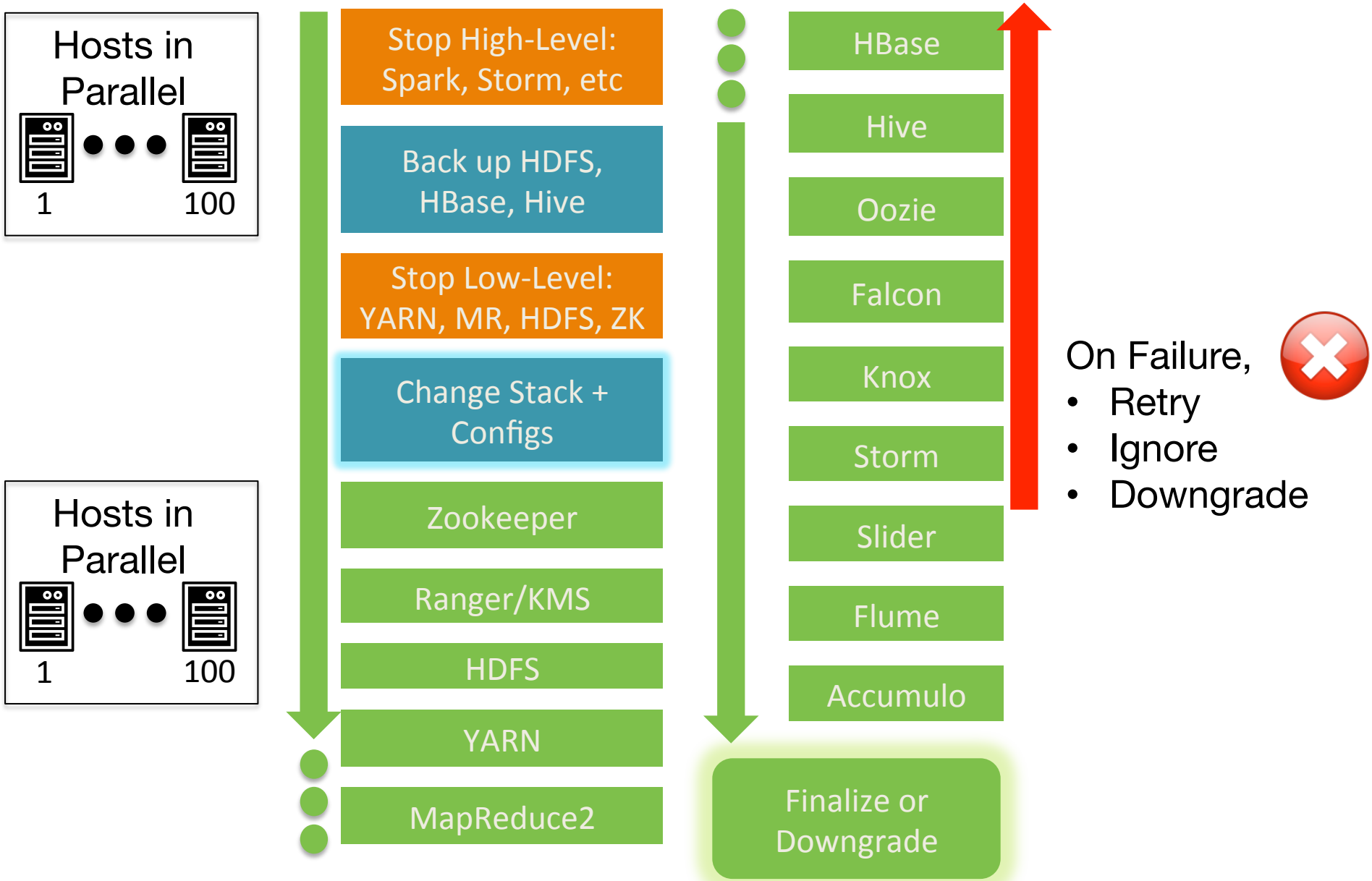
Automated Upgrade: Rolling or Express

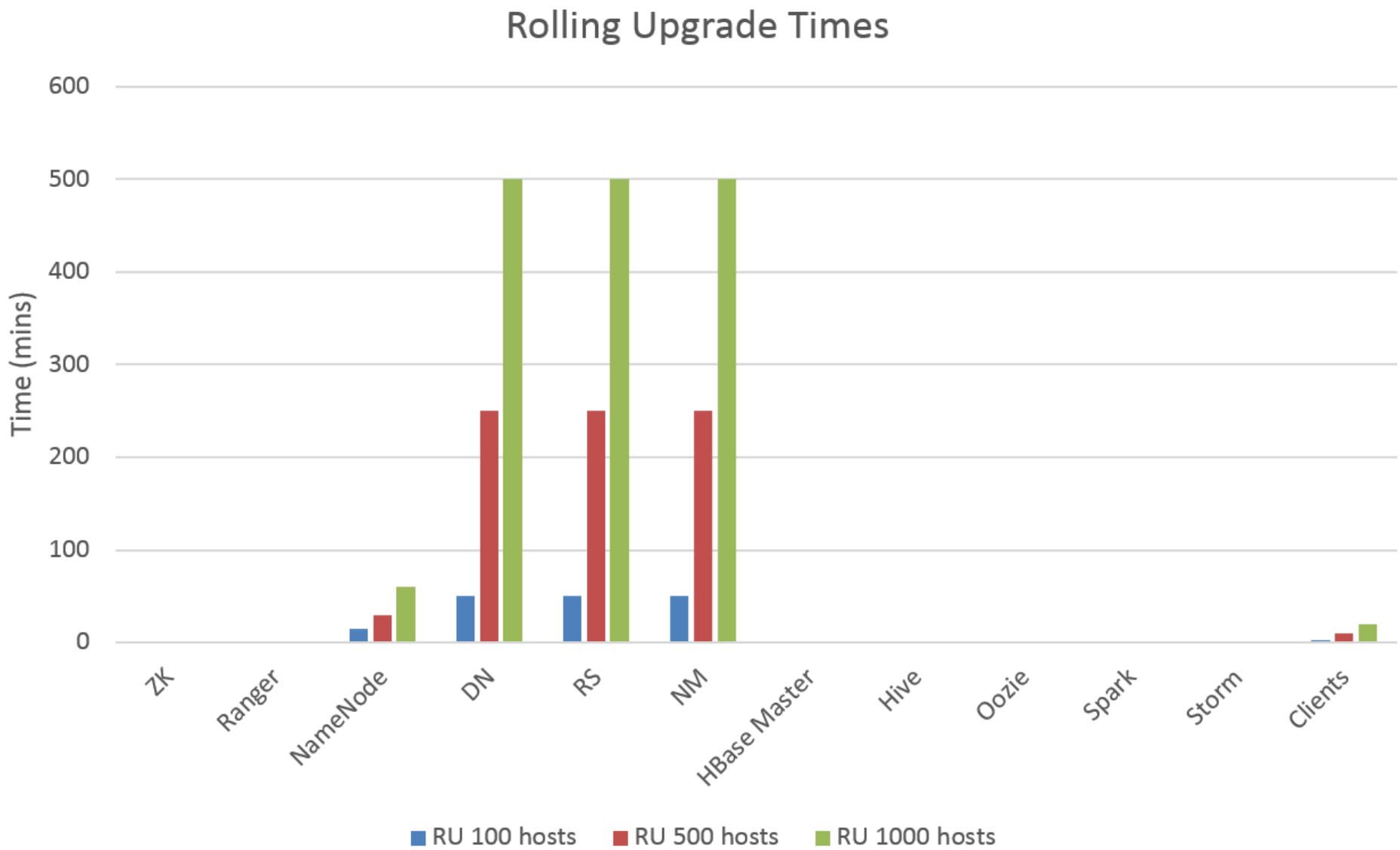


Process: Rolling Upgrade



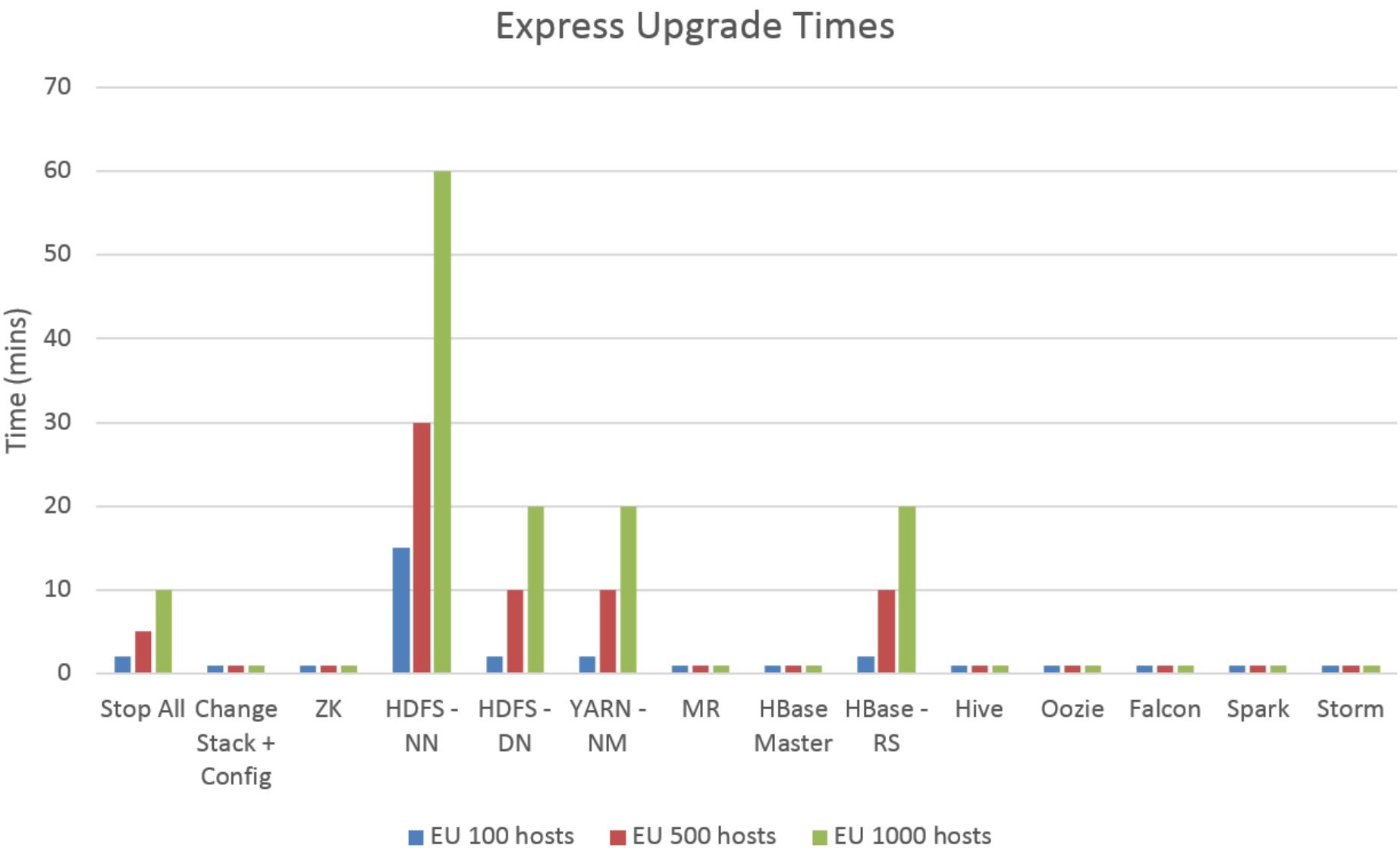
Process: Express Upgrade





Total Time:	2:53	13:16	26:26

Scales linearly with # of hosts



Total Time:	0:32	1:14	2:19
	5.4 X	**10.7 X**	**11.4 X**

Scales linearly with # batches (defaults to 100 hosts at a time)

Upgrade Endpoint

Status:
http://server:8080|8443/api/v1/clusters/$name/upgrades

Navigate
Groups: Core Master, Core Slaves, …, Post Cluster, etc.
Items: Upgrading Zookeeper Server on host namenode.apache.org


```
Upgrade: {
    cluster_name: "c1",
    create_time: 1495002598280,
    direction: "UPGRADE",
    downgrade_allowed: true,
    end_time: 1495003244035,
    exclusive: false,
    from_version: "2.5.3.0-37",
    pack: "nonrolling-upgrade-2.5",
    progress_percent: 100,
    request_context: "Upgrading to 2.5.5.0-157",
    request_id: 6,
    request_status: "COMPLETED",
    skip_failures: false,
    skip_service_check_failures: false,
    start_time: 1495002598412,
    suspended: false,
    to_version: "2.5.5.0-157",
    type: "INTERNAL_REQUEST",
    upgrade_type: "NON_ROLLING"
},
```

Upgrade – Debugging with SQL

```
SELECT u.upgrade_id, u.direction, u.from_version, u.to_version,
hrc.request_id, hrc.task_id, substr(g.group_title, 0, 30), substr(i.item_text, 0, 30), hrc.status
FROM upgrade u JOIN upgrade_group g ON g.upgrade_id = u.upgrade_id
JOIN upgrade_item i ON i.upgrade_group_id = g.upgrade_group_id
JOIN host_role_command hrc ON hrc.stage_id = i.stage_id AND hrc.request_id = u.request_id
ORDER BY hrc.task_id;
```

upgrade_id	direction	from_version	to_version	request_id	task_id	title	text	status
1	UPGRADE	2.5.3.0-37	2.5.5.0-157	6	103	Prepare Upgrade	[{"message":"Before continuing, please stop all YARN queues	COMPLETED
1	UPGRADE	2.5.3.0-37	2.5.5.0-157	6	104	Prepare Upgrade	[{"message":"Before continuing, please stop all long-runnin	COMPLETED
1	UPGRADE	2.5.3.0-37	2.5.5.0-157	6	105	Prepare Upgrade	Check Tez Tarball	COMPLETED
1	UPGRADE	2.5.3.0-37	2.5.5.0-157	6	106	Stop Components for High-Level Services	Stopping WebHCat Server on ale-atlas-1.c.pramod-thangali.in	COMPLETED
1	UPGRADE	2.5.3.0-37	2.5.5.0-157	6	107	Stop Components for High-Level Services	Stopping HiveServer2 on ale-atlas-1.c.pramod-thangali.inter	COMPLETED
1	UPGRADE	2.5.3.0-37	2.5.5.0-157	6	108	Stop Components for High-Level Services	Stopping Hive Metastore on ale-atlas-2.c.pramod-thangali.in	COMPLETED
1	UPGRADE	2.5.3.0-37	2.5.5.0-157	6	109	Stop Components for High-Level Services	Stopping NodeManager on 3 hosts	COMPLETED
1	UPGRADE	2.5.3.0-37	2.5.5.0-157	6	110	Stop Components for High-Level Services	Stopping NodeManager on 3 hosts	COMPLETED
1	UPGRADE	2.5.3.0-37	2.5.5.0-157	6	111	Stop Components for High-Level Services	Stopping NodeManager on 3 hosts	COMPLETED
1	UPGRADE	2.5.3.0-37	2.5.5.0-157	6	112	Stop Components for High-Level Services	Stopping ResourceManager on ale-atlas-1.c.pramod-thangali.i	COMPLETED
1	UPGRADE	2.5.3.0-37	2.5.5.0-157	6	113	Stop Components for High-Level Services	Stopping App Timeline Server on ale-atlas-2.c.pramod-thanga	COMPLETED
1	UPGRADE	2.5.3.0-37	2.5.5.0-157	6	114	Stop Components for High-Level Services	Stopping History Server on ale-atlas-3.c.pramod-thangali.in	COMPLETED
1	UPGRADE	2.5.3.0-37	2.5.5.0-157	6	115	Perform Backups	[{"message":"Before continuing, please backup the Hive Meta	COMPLETED
1	UPGRADE	2.5.3.0-37	2.5.5.0-157	6	116	Perform Backups	Snapshot HBASE	COMPLETED
1	UPGRADE	2.5.3.0-37	2.5.5.0-157	6	117	Perform Backups	Prepare HDFS	COMPLETED
1	UPGRADE	2.5.3.0-37	2.5.5.0-157	6	118	Stop Components for Core Services	Stopping RegionServer on 3 hosts	COMPLETED
1	UPGRADE	2.5.3.0-37	2.5.5.0-157	6	119	Stop Components for Core Services	Stopping RegionServer on 3 hosts	COMPLETED
1	UPGRADE	2.5.3.0-37	2.5.5.0-157	6	120	Stop Components for Core Services	Stopping RegionServer on 3 hosts	COMPLETED
1	UPGRADE	2.5.3.0-37	2.5.5.0-157	6	121	Stop Components for Core Services	Stopping HBase Master on ale-atlas-3.c.pramod-thangali.inte	COMPLETED
1	UPGRADE	2.5.3.0-37	2.5.5.0-157	6	122	Stop Components for Core Services	Stopping Kafka Broker on ale-atlas-3.c.pramod-thangali.inte	COMPLETED
1	UPGRADE	2.5.3.0-37	2.5.5.0-157	6	123	Stop Components for Core Services	Stopping DataNode on 3 hosts	COMPLETED
1	UPGRADE	2.5.3.0-37	2.5.5.0-157	6	124	Stop Components for Core Services	Stopping DataNode on 3 hosts	COMPLETED
1	UPGRADE	2.5.3.0-37	2.5.5.0-157	6	125	Stop Components for Core Services	Stopping DataNode on 3 hosts	COMPLETED

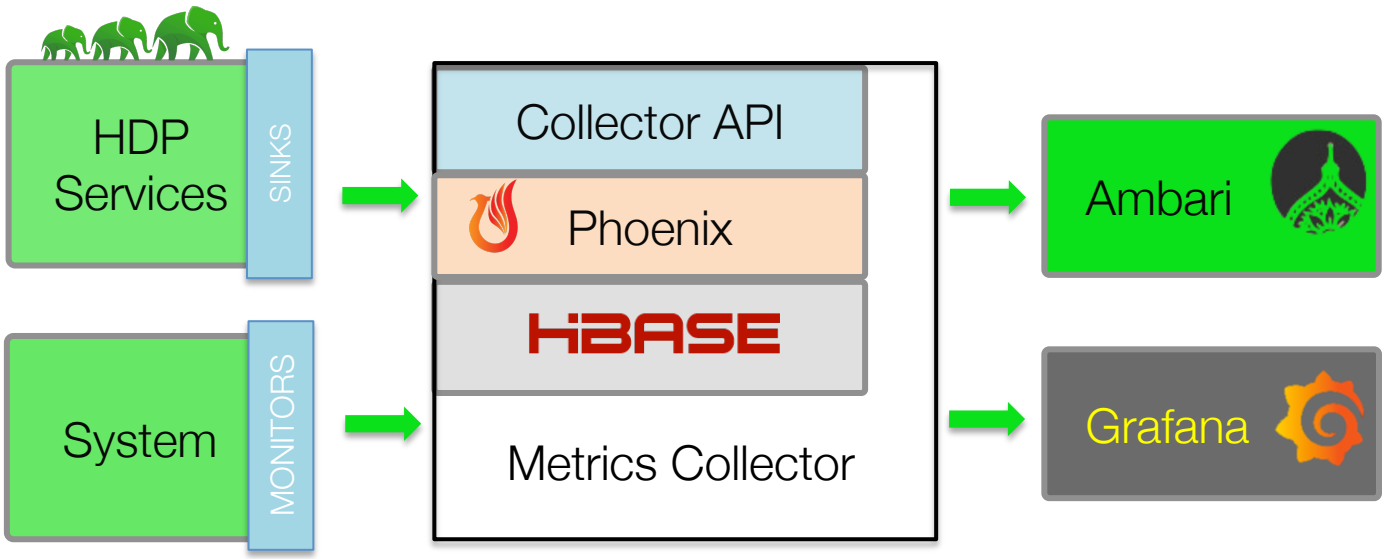
Alerting Framework

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

Motivation Behind Ambari Metrics System

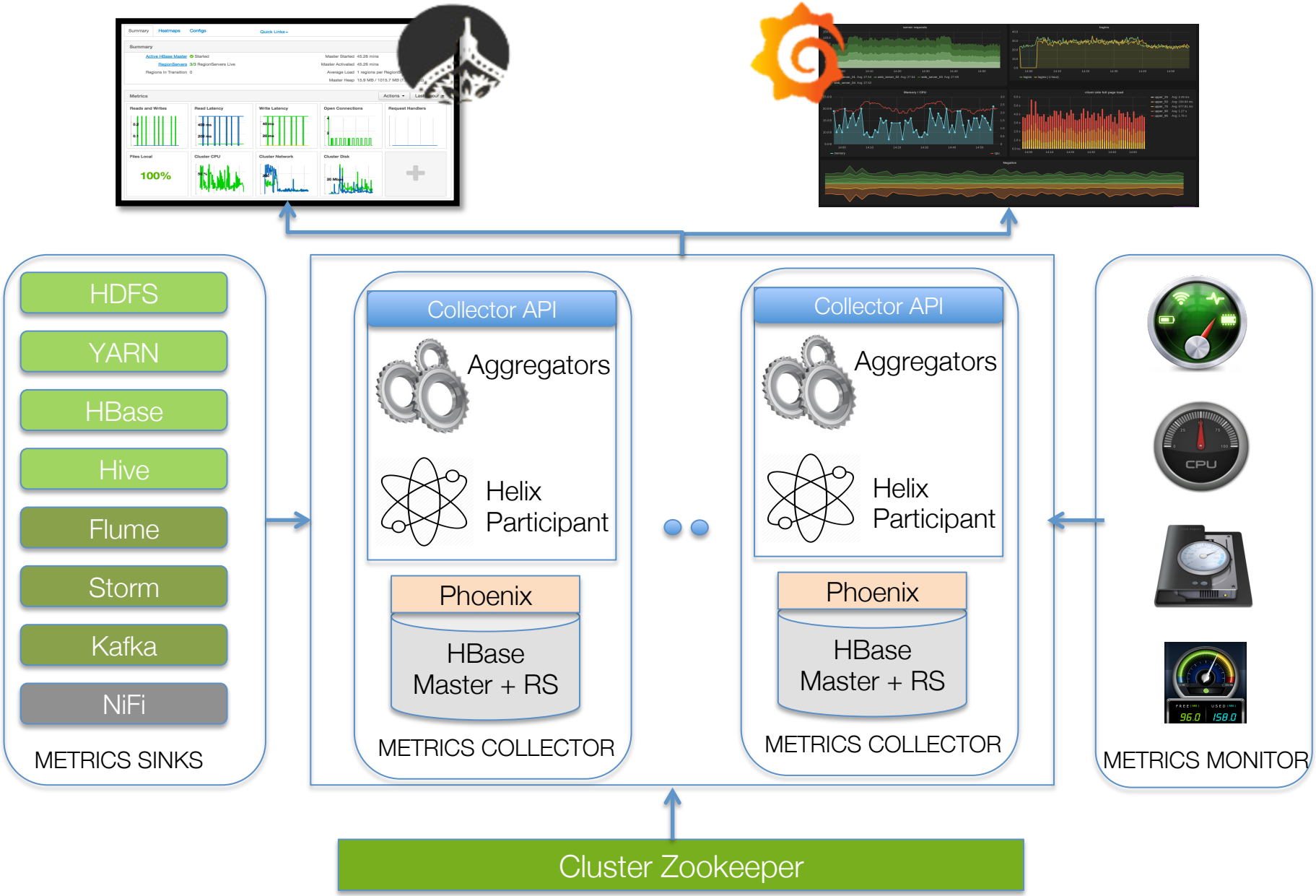
- Limited Ganglia capabilities
- OpenTSDB – GPL license and needs a Hadoop cluster
- Aggregation at multiple levels: Service, Time
- Alerts based on metrics system
- Scale past a 1000 nodes
- Analytics based on use cases
- Fine grained control over retention, collection intervals, aggregation
- Pluggable and Extensible

AMS Architecture



- Custom Sinks – HDFS, YARN, HBase, Storm, Kafka, Flume, Accumulo
- Monitors – lightweight daemon for system metrics
- Collector – API daemon + HBase (embedded / distributed)
- Phoenix schema designed for fast reads
- Managed HBase
- Grafana support from version 2.2.2

AMS Distributed Collector Arch Details



AMS Features

- Simple POST API for sending metrics.
- Rich GET API to fetch metrics in specific granularity
  - Point in time & series
  - Top N support
  - Rate support
- Performs Host level aggregation as well as time based down sampling
- Highly tunable system
  - Adjust rate of collecting/sending metrics
  - Adjust granularity of data being stored
  - Skip Aggregation for certain metrics
  - Whitelist metrics
- Metadata API that provides information on what metrics are being collected and which component is sending these metrics
- Abstract Sink implementation to facilitate easy integration with metrics collector
- HTTPS Support

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

AMS - Grafana Integration

System - Servers

Zoom Out

Apr 27, 2017 07:21:23 to Apr 27, 2017 14:40:48

Dashboards

HOST | Cluster: c1 | Hosts: All | show: top | 20 | aggregator: avg

System Dashboards

Main Org.

Sign in

Metrics to see the system status per host on the server. Click on each row title to expand on demand to look at various metrics.

This dashboard is managed by Ambari. You may lose any changes made to this dashboard. If you want to customize, make your own copy.

CPU - USER / SYSTEM

CPU Utilization - User

5.0% 4.0% 3.0% 2.0% 1.0% 0%

08:00 09:00 10:00 11:00 12:00 13:00 14:00

CPU Utilization - System

0.50% 0.40% 0.30% 0.20% 0.10% 0%

08:00 09:00 10:00 11:00 12:00 13:00 14:00

Kafka - Hosts

Zoom Out

Last 6 hours

kafka_broker | Cluster: | Hosts: All | show: top | 20 | aggregator: avg

Kafka Dashboards

Metrics for the Kafka cluster on a per broker level. Click on each row title to expand on demand to look at various metrics.

This dashboard is managed by Ambari. You may lose any changes made to this dashboard. If you want to customize, make your own copy.

BYTES / MESSAGES / PARTITIONS

Bytes In & Out /s

94 B 93 B 92 B 91 B 90 B 89 B 88 B

12:00 13:00 14:00 15:00 16:00 17:00

Messages In /s

90 85 80 75 70

12:00 13:00 14:00 15:00 16:00 17:00

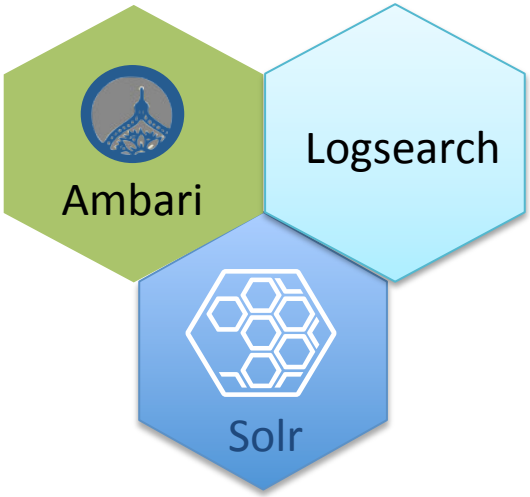
Under Replicated Partitions

98 96 94 92 90 88

12:00 13:00 14:00 15:00 16:00 17:00

Log Search

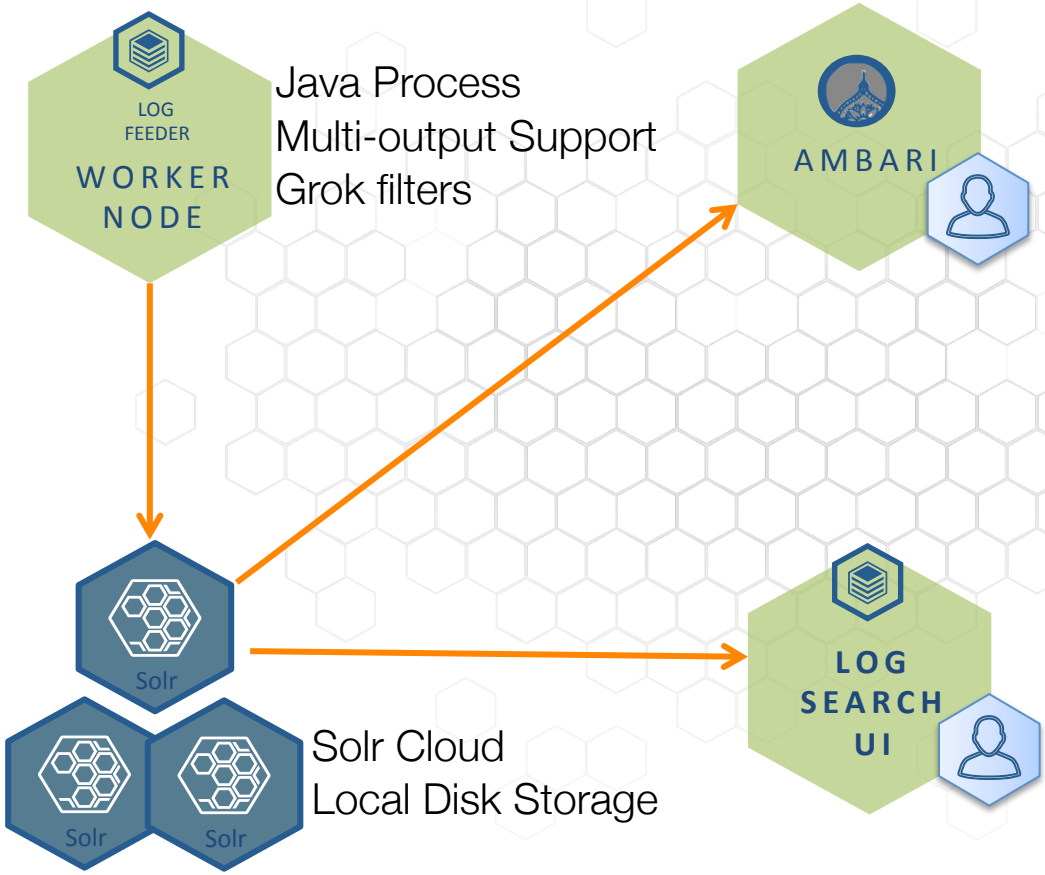
Search and index HDP logs!

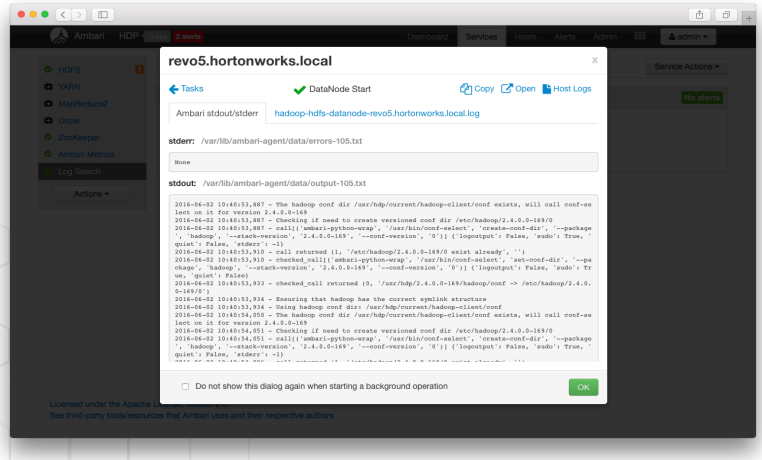


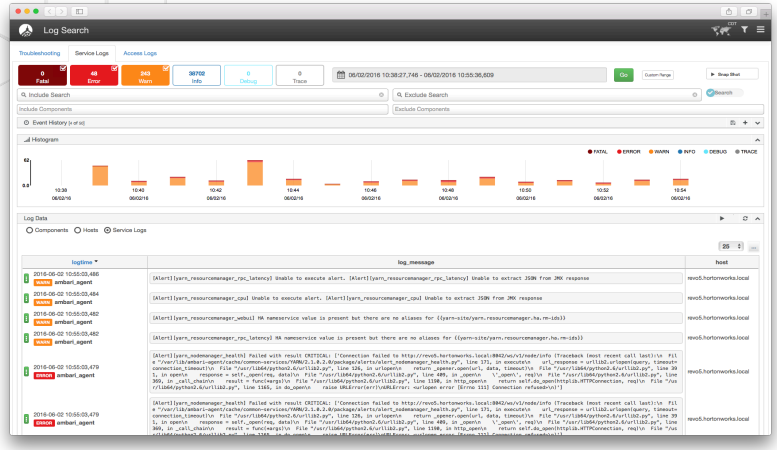
Capabilities

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

Log Search


LOG
FEEDER
WORKER
NODE
Java Process
Multi-output Support
Grok filters
AMBARI
Solr
Solr
Solr
Solr Cloud
Local Disk Storage
LOG
SEARCH
UI


revo5.hortonworks.local
Tasks
DataNode Start
Copy
Open
Host Logs
Ambari stdout/stderr
hadoop-hdfs-datanode-revo5.hortonworks.local.log
stderr: /var/lib/ambari-agent/data/errors-105.txt
stdout: /var/lib/ambari-agent/data/output-105.txt
Do not show this dialog again when starting a background operation
OK


Log Search
Troubleshooting
Service Logs
Access Logs
Include Search
Exclude Search
Histogram
Log Data

Future of Apache Ambari 3.0

- Cloud features
- Service multi-instance (e.g., two ZK quorums)
- Service multi-versions (Spark 2.0 & Spark 2.2)
- YARN assemblies & services
- Patch Upgrades: upgrade individual components in the same stack version, e.g., just DN and RM in HDP 3.0.*.* with zero downtime
- Ambari High Availability

Resources

Contribute to Ambari:
https://cwiki.apache.org/confluence/display/AMBARI/Quick+Start+Guide

Referenced Articles:
https://community.hortonworks.com/articles/43816/how-to-createadd-the-service-stop-the-service.html

https://community.hortonworks.com/articles/80635/optimize-ambari-performance-for-large-clusters.html

Image Sources:
http://www.vacationgetaways4less.com/wp-content/gallery/miami-newport-beachside-hotel-resort-banner/miami-beach-south-beach-night-730x302.jpg

https://ak9.picdn.net/shutterstock/videos/2139614/thumb/1.jpg

Many thanks to the ASF, audience, and event organizers.

THE APACHE® SOFTWARE FOUNDATION