Apache Eagle in Action

Secure Hadoop in Real-time

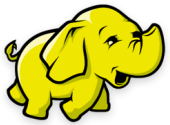
Hao Chen / 陈浩 / hao@apache.org / @haozch

Who is the Guy

Hao Chen / 陈浩

Tech Lead, Sr. Software Engineer @ eBay Cloud Service
hchen9@ebay.com

Co-creator, Committer and PMC @ Apache Eagle
hao@apache.org



Speaker @ Hadoop Summit (SJC, SHA, BJ) ...
http://people.apache.org/~hao

Agenda

- **About Eagle**
- Architecture
- Ecosystem
- Q & A

What's Apache Eagle

Apache Eagle is a distributed real-time monitoring and alerting engine for hadoop from eBay

Open sourced as Apache Incubator Project on Oct 26th 2015

Secure Hadoop in Realtime *a data activity monitoring solution to instantly identify access to sensitive data, recognize attacks/ malicious activity and block access in real time.*

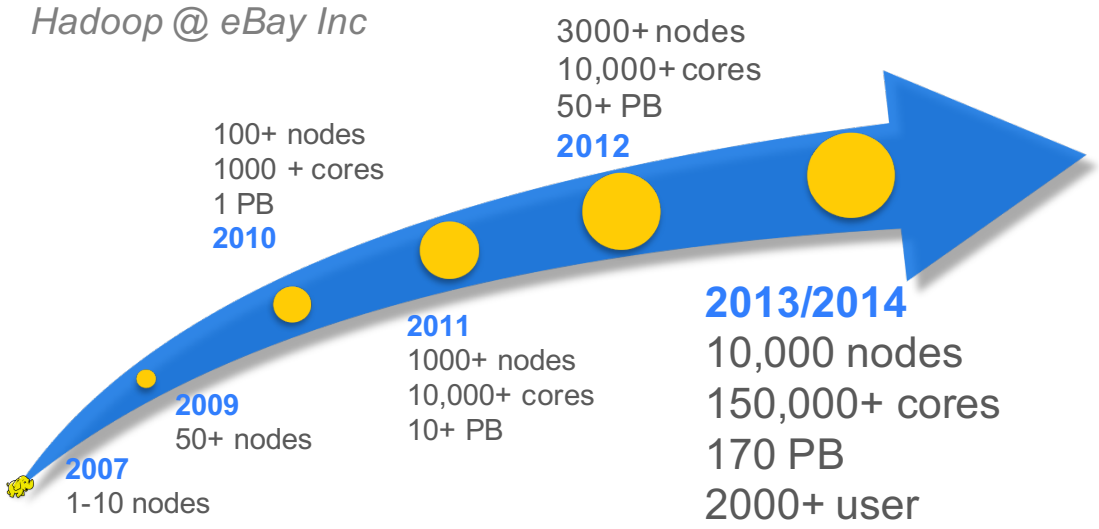
See http://eagle.incubator.apache.org or http://goeagle.io

Apache Eagle History

Donated to Apache Software Foundation (ASF) from eBay at Oct 26th, 2015

Why build Apache Eagle

Eagle was initialized by end of 2013 for hadoop ecosystem monitoring as any existing tool like zabbix, ganglia can not handle the huge volume of metrics/logs generated by hadoop system in eBay.



Hadoop Data
- Security
- Activity

Hadoop Platform
- Heath
- Availability
- Performance

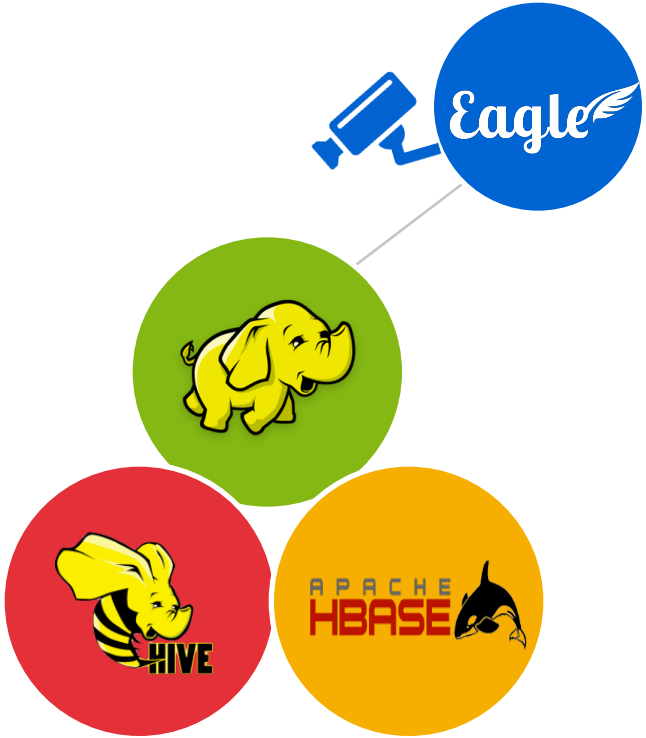
Apache Eagle @ eBay

MONITOR	7	CLUSTERS
	7427	NODES
	160 PB	DATA
PROCESS	10 B+	EVENTS / DAY
	500+	METRIC TYPES
	50,000+	JOBS / DAY
	50,000,000+	TASKS / DAY

Agenda

- About Eagle
- **Architecture**
- Ecosystem
- Q & A

Apache Eagle Architecture Overview



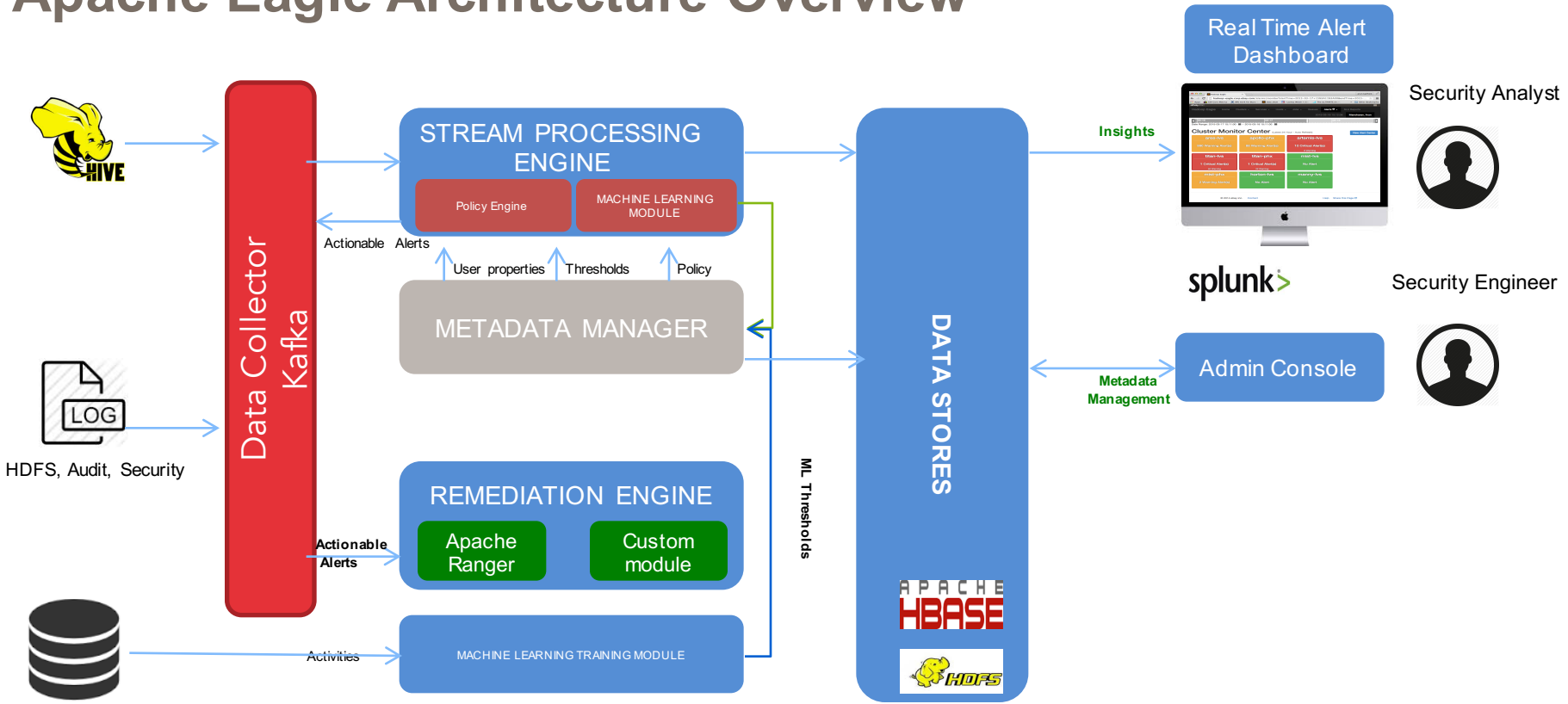
Scalable
Scales to monitor thousands of policies and billions of access events

Extensible
Eagle can be easily extended to monitor other data sources

Real-time
Generates alerts in real time and blocks users with malicious intent

Machine Learning
Create dynamic user profiles based on user behavior

Apache Eagle Architecture Overview



Apache Eagle Architecture Features

- Real-time Data Collection
- Distributed Policy Engine
- Stream Processing DSL
- Scalable Data Storage & Query
- Machine Learning Intergration

NOTE *{NAME}-{NUMBER} like HDFS-6914 means open source project ticket id contributed by us*

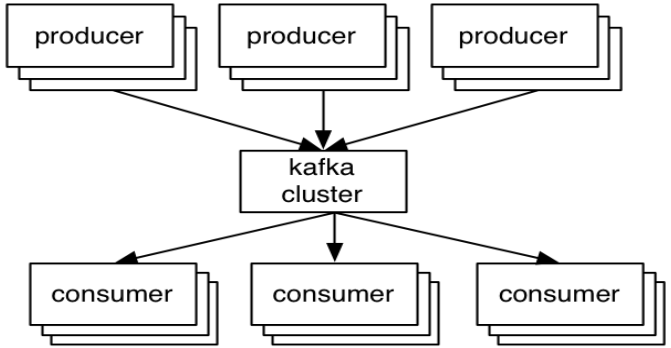
Apache Eagle - Data Collection

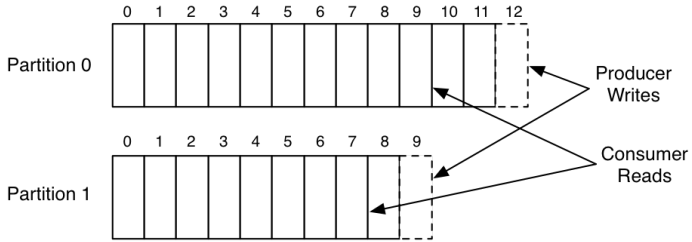
Decoupling with Message Bus

- **Apache Kafka:** high-throughput distributed messaging
- ***Partition:*** *balance between logic and throughput*

Cross-Platform Integration

- **Community Kafka Client** (18+)
  - Python/Go/C/C++/JAVA ..
- **Enhanced Log4j-kafka**
  - KAFKA-2041: Extensible Partition Key
  - KAFKA-2077: Advanced Topic Selector

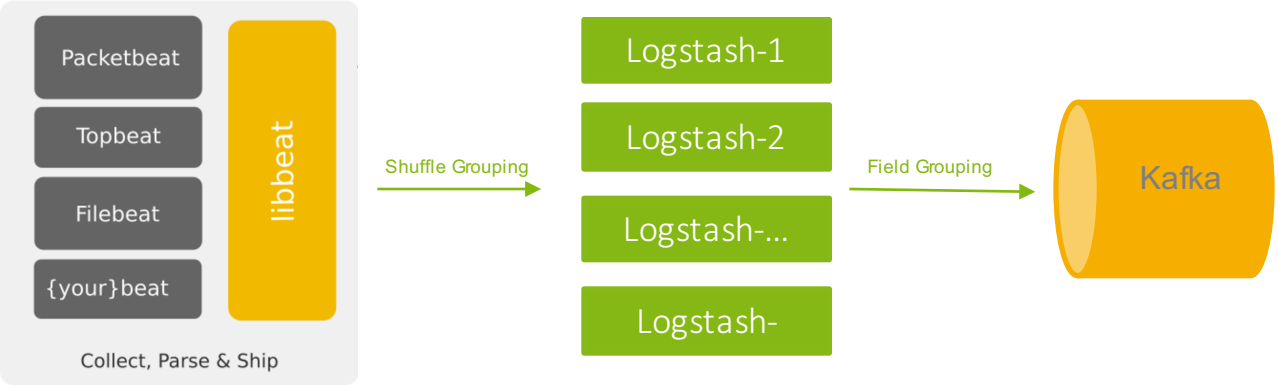




Apache Eagle - Data Collection

Availability: Filebeat + Logstash

Resource consumption balance
Message throughput balance (LOGSTASH-179)



Ligh-weight collector (golang) with daemon

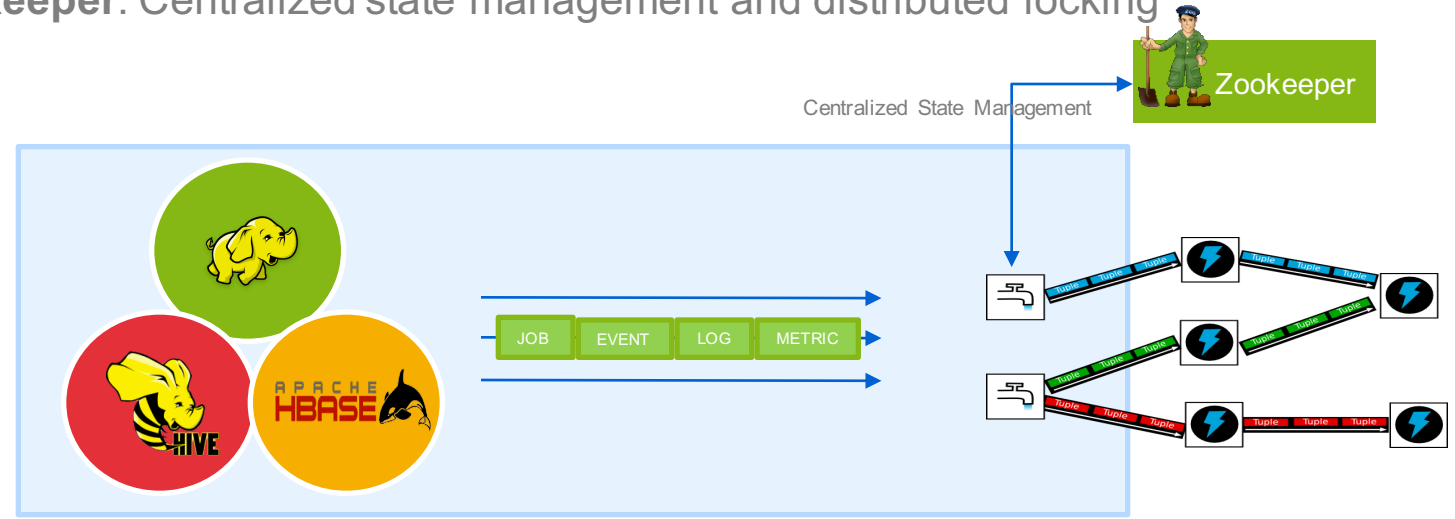
Logstash instances cluster

Distributed Message Bus

Apache Eagle - Data Collection

Scalability: Distributed Real-time Ingestion

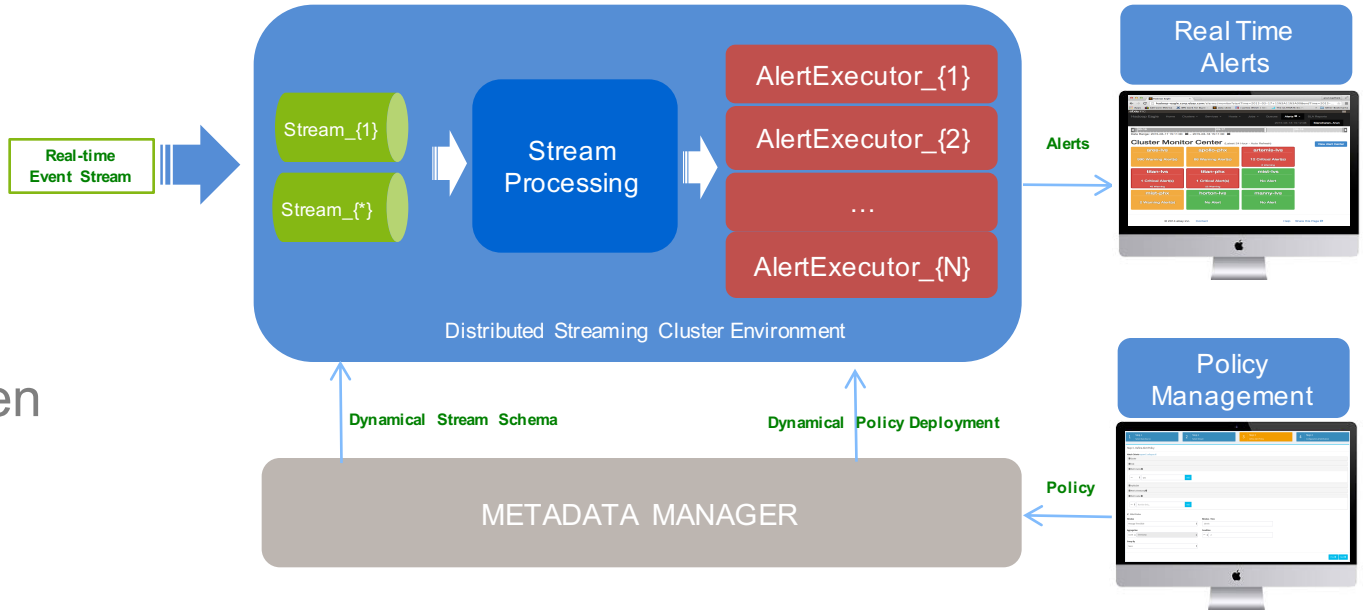
Storm Spout: Distributed crawling for hadoop job, node jmx and service logs, etc.
Zookeeper: Centralized state management and distributed locking



Apache Eagle - Distributed Real-time Policy Engine

Highlights

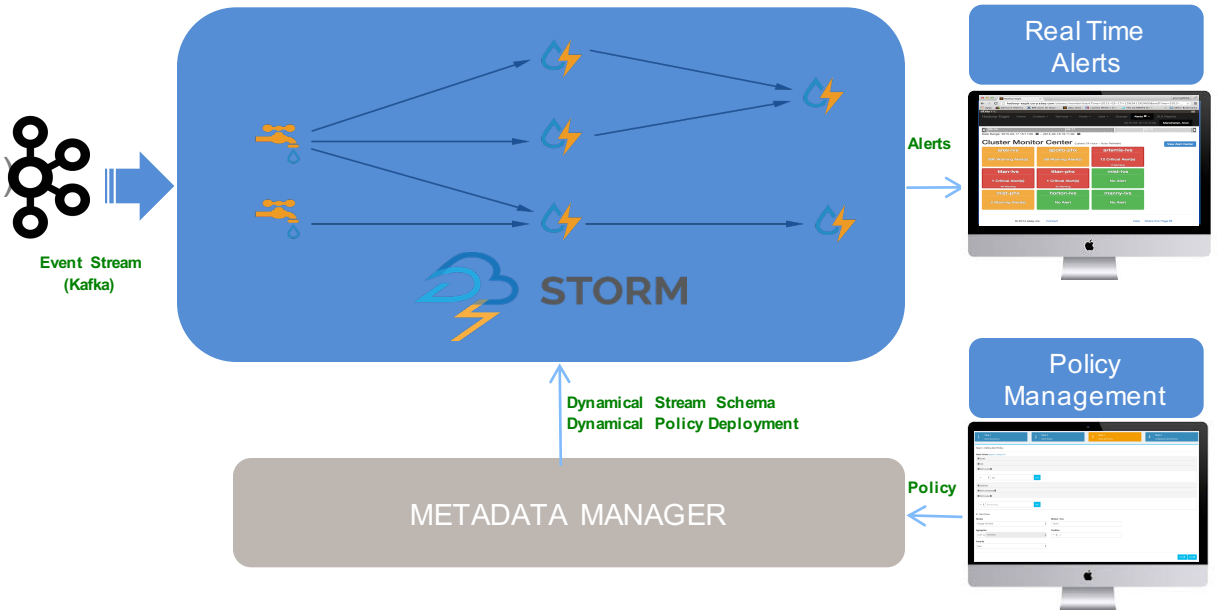
- Real-time
- Usability
- Scalability
- Extensibility
- Metadata-driven



Apache Eagle - Distributed Real-time Policy Engine

Real-time

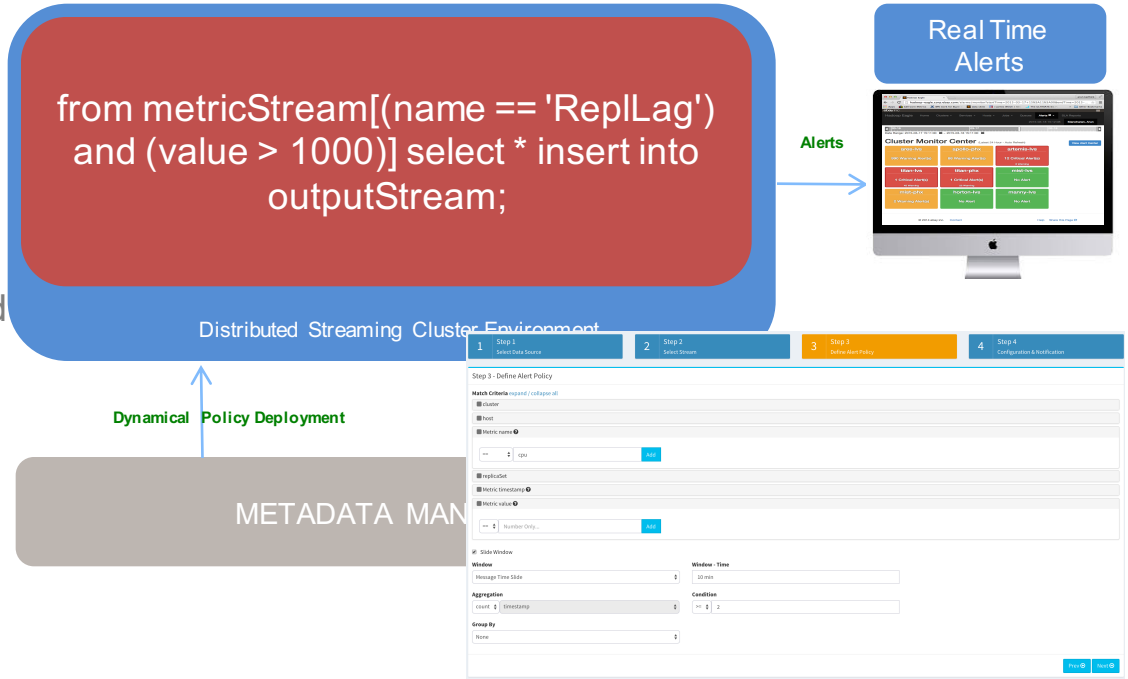
- Kafka-based Distributed Message Bus (Extensible)
- Storm-based Real-time Execution Environment (Extensible)
- Stream events are processed and alerts are evaluated during streaming



Apache Eagle - Distributed Real-time Policy Engine

Usability

- Powerful SQL-Like CEP CQL for Policy Definition
- Dynamical Poilcy Metadata Lifecycle Management (Deployment/Update)
- Easy-to-use Policy management and Alert analytics UI



Apache Eagle - Distributed Real-time Policy Engine

Full-function Streaming CEP CQL: Siddhi on Storm by default

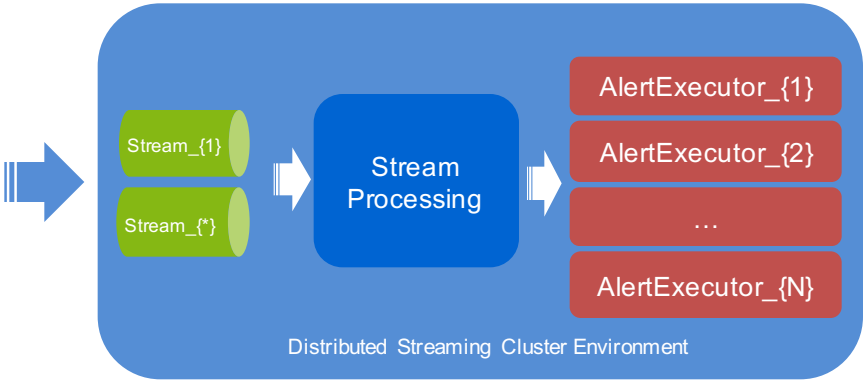
```
hdfsAuditLogEventStream[(src == '/tmp/private')]#window.externalTime(timestamp,10 min) select user, count(timestamp) as aggValue group by user having aggValue >= 5 insert into outputStream;
```

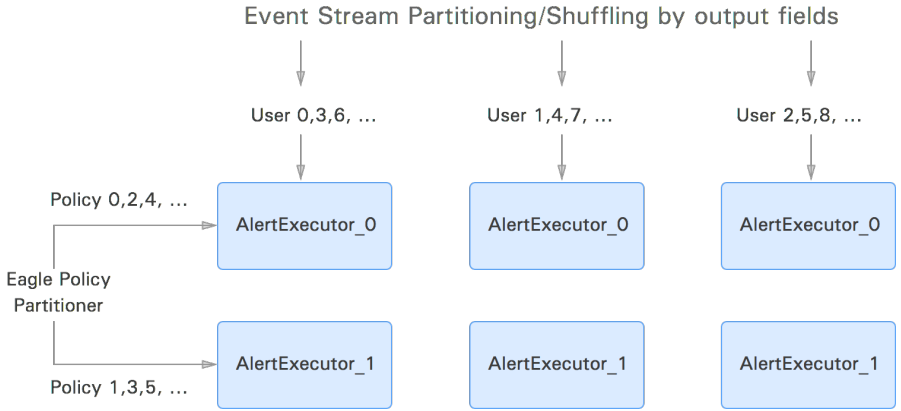
- **Filter**
- **Join**
- **Aggregation**: Avg, Sum , Min, Max, etc
- **Group by**
- **Having**
- **Stream handlers for window**: TimeWindow, Batch Window, Length Window
- **Conditions and Expressions**: and, or, not, ==,!=, >=, >, <=, <, and arithmetic operations
- **Pattern processing**
- **Sequence processing**
- **Event Tables**: intergrate historical data in realtime processing
- **SQL-Like Query**: Query, Stream Definition and Query Plan compilation

Apache Eagle - Distributed Real-time Policy Engine

Scalability: dynamic policy partition by {event} * {policy}

- *N Users with 3 partitions, M policies with 2 partitions, then 3*2 physical tasks*
- *Physical partition + policy-level partition*

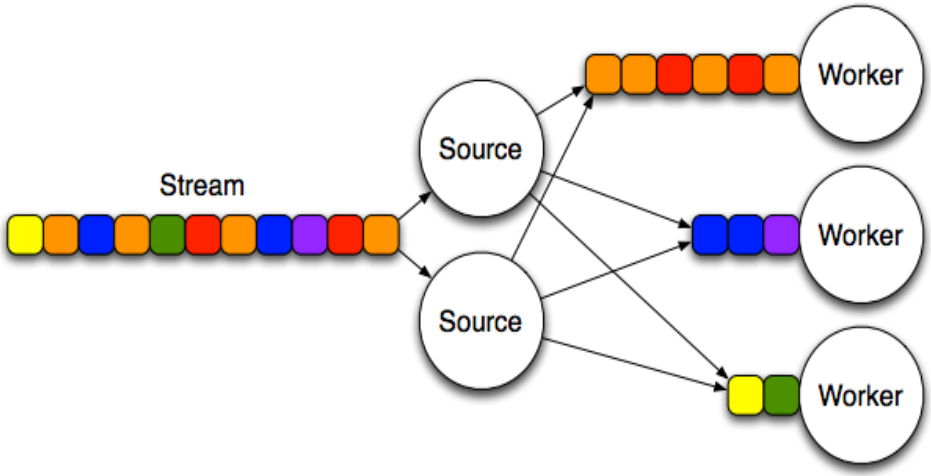




Apache Eagle - Distributed Real-time Policy Engine

Distributed Streaming Partition Problem

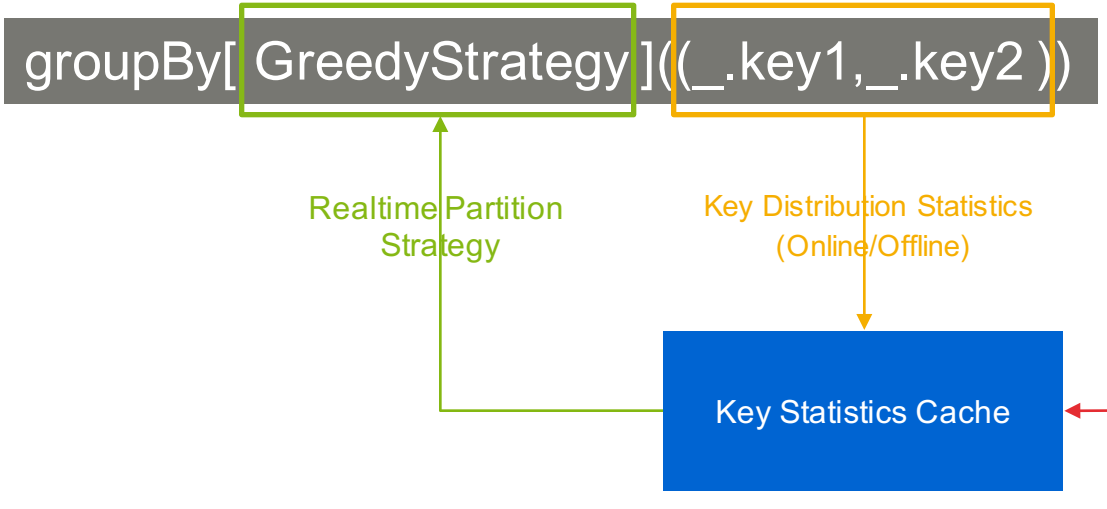
https://en.wikipedia.org/wiki/Partition_problem



$S = \{3,1,1,2,2,1,1\} \rightarrow \begin{cases} S1 = \{1,1,1,1,1\} \\ S2 = \{2,2\} \\ S3 = \{3\} \end{cases}$

Apache Eagle - Distributed Real-time Policy Engine

Distributed Streaming Partition Strategy



Strategy

- Greedy (Online/Offline)
- PoTC
- PKG
- Hashing

Apache Eagle - Distributed Real-time Policy Engine

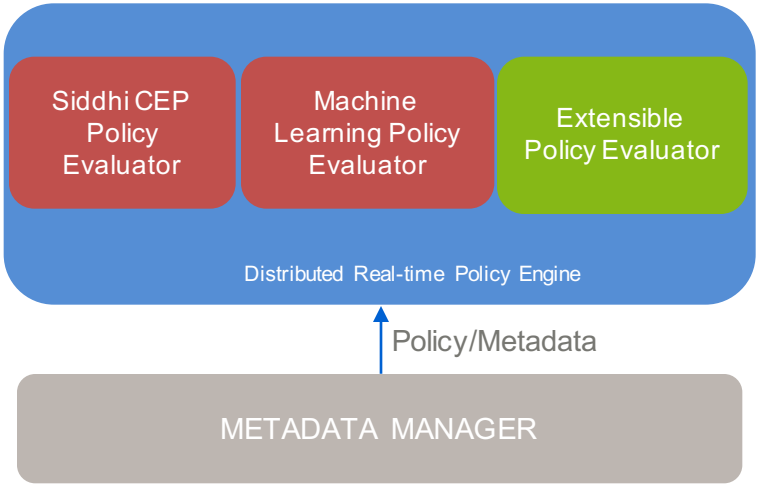
Extensibility

- Support WSO2 Siddhi CEP as first class
- Extensible policy engine implementation

```
public interface PolicyEvaluatorServiceProvider {
        public String getPolicyType(); // literal string to identify one type of policy
        public Class getPolicyEvaluator(); // get policy evaluator implementation
        public List getBindingModules(); // policy text with json format to object mapping
}
```

- Extensible policy lifecycle management

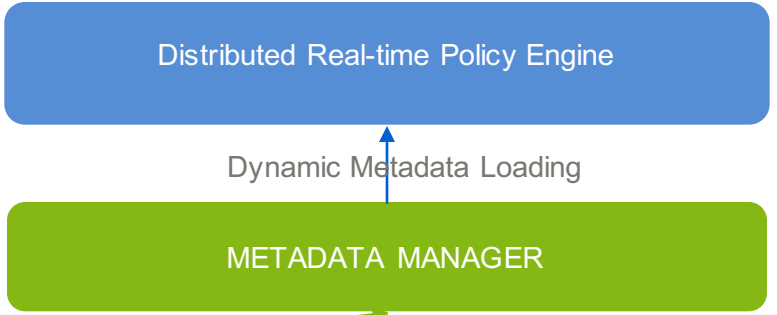
```
public interface PolicyEvaluator {
        public void evaluate(ValuesArray input) throws Exception; // evaluate input event
        public void onPolicyUpdate(AlertDefinitionAPIEntity newAlertDef);// policy update
        public void onPolicyDelete(); // invoked when policy is deleted
}
```



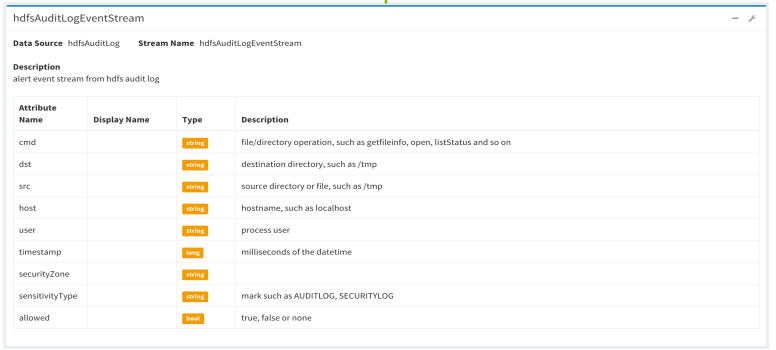
Apache Eagle - Distributed Real-time Policy Engine

Metadata-Driven

- Stream Schema: AlertStreamSchemaEntity
- Policy Definition: AlertDefinitionAPIEntity
- Central metadata management
- Dynamic metadata deployment




```
@Table("alertdef")
@ColumnFamily("f")
@Prefix("alertdef")
@Service(AlertConstants.ALERT_DEFINITION_SERVICE_ENDPOINT_NAME)
@JsonIgnoreProperties(ignoreUnknown = true)
@TimeSeries(false)
@Tags({"site", "dataSource", "alertExecutorId", "policyId", "policyType"})
@Indexes({
  @Index(name="Index_1_alertExecutorId",columns = { "alertExecutorID" }, unique = true),
})
public class AlertDefinitionAPIEntity extends TaggedLogAPIEntity{
  @Column("a")
  private String desc;
  @Column("b")
  private String policyDef;
  @Column("c")
  private String dedupeDef;
```



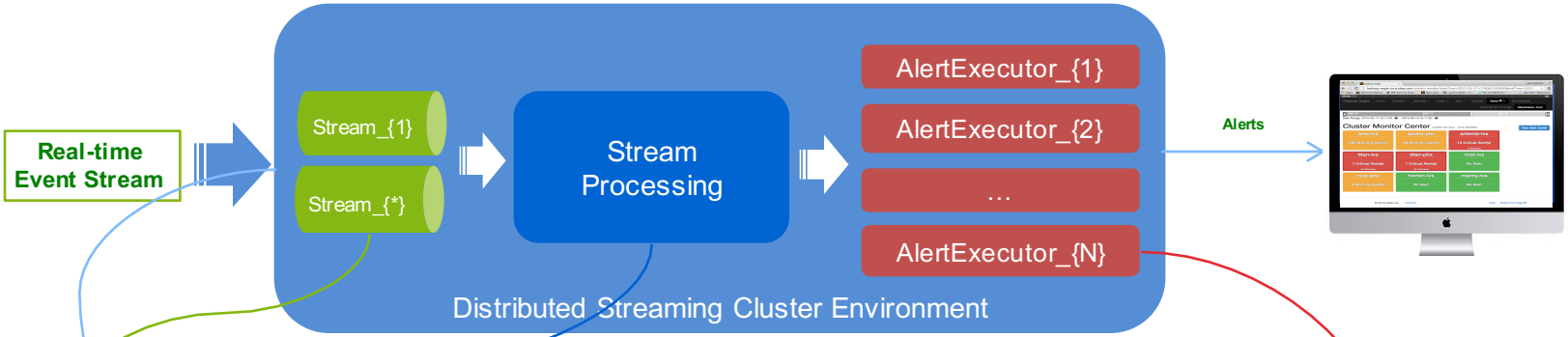
hdfsAuditLogEventStream

Data Source hdfsAuditLog **Stream Name** hdfsAuditLogEventStream

Description
alert event stream from hdfs audit log

Attribute Name	Display Name	Type	Description
cmd		string	file/directory operation, such as getfileinfo, open, listStatus and so on
dst		string	destination directory, such as /tmp
src		string	source directory or file, such as /tmp
host		string	hostname, such as localhost
user		string	process user
timestamp		long	milliseconds of the datetime
securityZone		string	
sensitivityType		string	mark such as AUDITLOG, SECURITYLOG
allowed		bool	true, false or none

Apache Eagle - Fluent Stream Processing DSL




```
val env = ExecutionEnvironment.getStorm()
env.fromKafka(KafkaConfig)
    .flatMap(AuditLogTransformer)
    .groupBy(_.user)
    .flatMap(UserProfileAggregator);
   .alert.persistAndEmail
env.execute()
```

Apache Eagle - Fluent Stream Processing DSL

1. Development 2. Optimization 3. Compile to native app

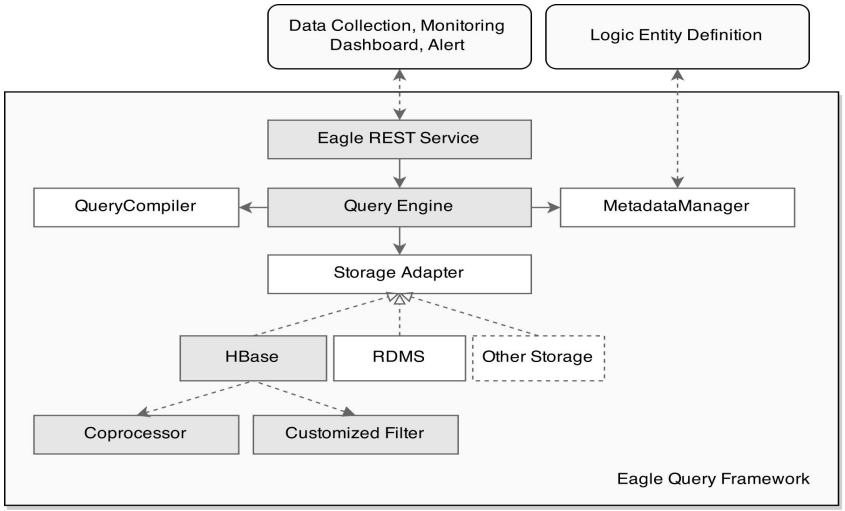
- Physical execution platform independent
- Easily assemble data transformation, filtering, join and alerting DAG in fluent way
- DAG rewrite and optimization
  - StreamUnionExpansion
  - StreamGroupbyExpansion
  - StreamNameExpansion
  - StreamAlertExpansion
  - StreamParallelismConfigExpansion

```
trait StreamProducer{
  filter
  flatMap
  map{1,2,3,4}
  groupBy
  streamUnion  // stream join is hard, not implemented for storm
  alertWithConsumer
}
```

```
StormExecutionEnvironment env =
ExecutionEnvironmentFactory.getStorm(config);
env.newSource(new
KafkaSourcedSpoutProvider().getSpout(config)).renameOutputFields(1)
      .flatMap(new AuditLogTransformer())
      .groupBy(0)
      .flatMap(new UserProfileAggregatorExecutor());
.alertWithConsumer("userActivity","userProfileExecutor")
env.execute();
```

Apache Eagle - Scalable Data Storage and Query

- Entity Metadata on large-scale NoSQL storage like HBase
- Full-function SQL-Like REST Query
- Optimized rowkey design for time-series monitoring data
- HBase Coprocessor
- Secondary Index




```
@Table("alertdef")
@ColumnFamily("f")
@Prefix("alertdef")
@Service(AlertConstants.ALERT_DEFINITION_SERVICE_ENDPOINT_NAME)
@JsonIgnoreProperties(ignoreUnknown = true)
@TimeSeries(false)
@Tags({"site", "dataSource", "alertExecutorId", "policyId", "policyType"})
@Indexes({
  @Index(name="Index_1_alertExecutorId",columns = { "alertExecutorID" }, unique = true),
})
public class AlertDefinitionAPIEntity extends TaggedLogAPIEntity{
  @Column("a")
  private String desc;
  @Column("b")
  private String policyDef;
  @Column("c")
  private String dedupeDef;
```

```
query=
AlertDefinitionService[@dataSource="hiveQueryLog"]{@policyDef}
```

Apache Eagle – Uniform HBase Rowkey Design

Uniform rowkey design

Rowkey ::= Prefix | Partition Keys | timestamp | tagName | tagValue | …

- Metric

  Rowkey ::= Metric Name | Partition Keys | timestamp | tagName | tagValue | …

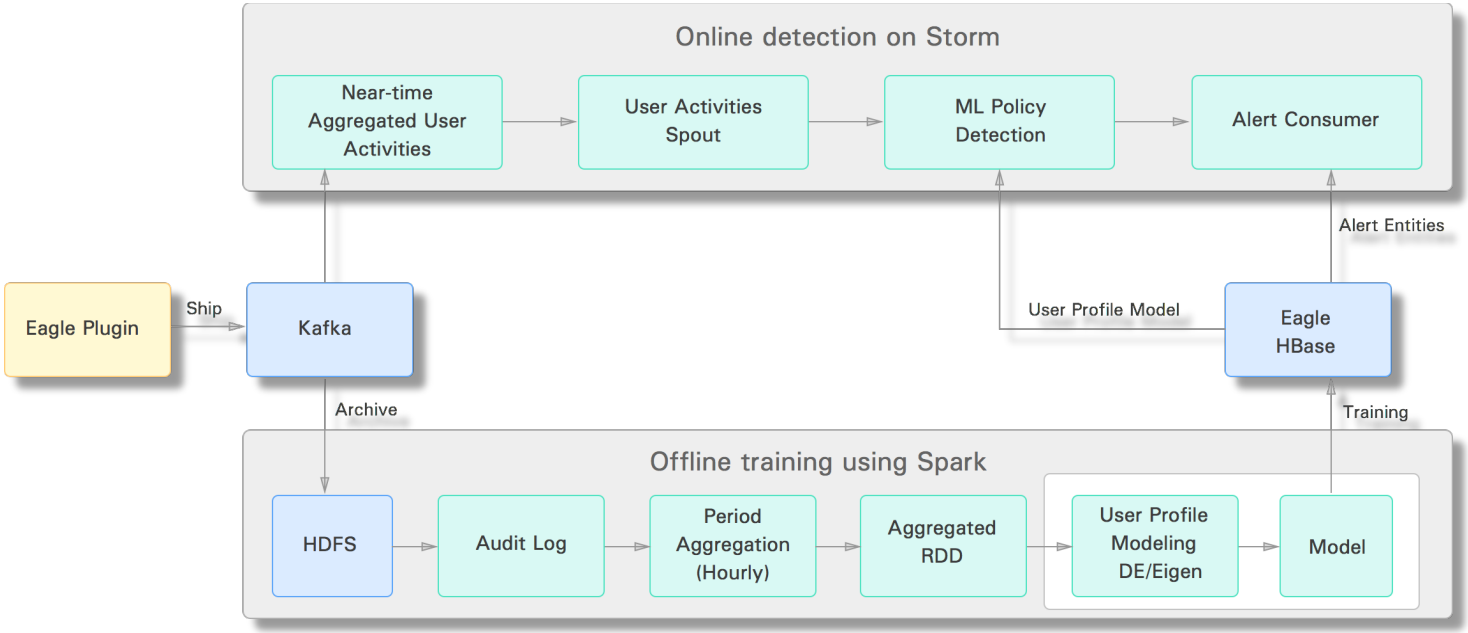
- Entity

  Rowkey ::= Default Prefix | Partition Keys | timestamp | tagName | tagValue | …

- Log

  Rowkey ::= Log Type | Partition Keys | timestamp | tagName | tagValue | …
  Rowvalue ::= Log Content

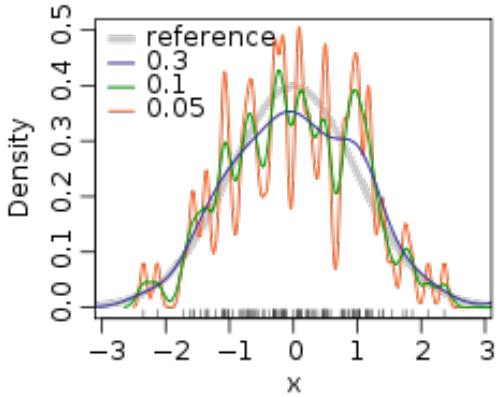
Apache Eagle - Machine Learning Intergration



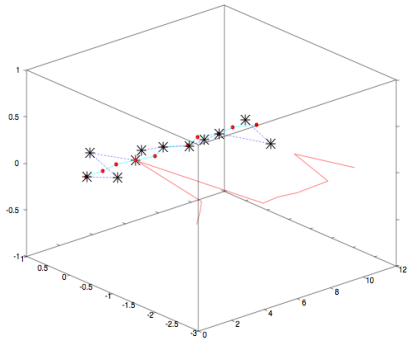
Apache Eagle – User/System Activity Profiling

User Activity Profiling

Offline: Determine bandwidth from training dataset the kernel density function parameters (KDE)
Online: If a test data point lies outside the trained bandwidth, it is anomaly (Policy)



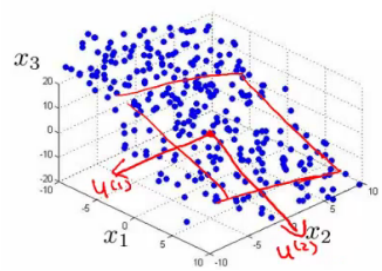
Kernel Density Function


PCs(Principle Components) in EVD
(Eigenvalue Value Decomposition)

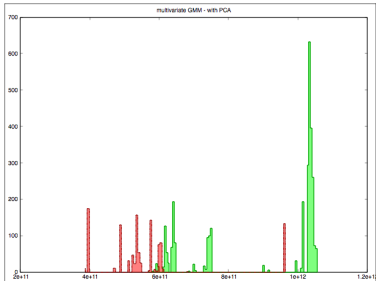
Apache Eagle - Anomaly Metric Predictive Detection

Anomaly Metric Predictive Detection

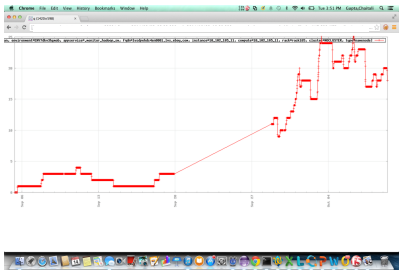
Offline: *Analyzing and combining 500+ metrics together for causal anomaly detections (IG -> PCA -> GMM -> MCC)*
Online: Predictively alert for anomaly metrics

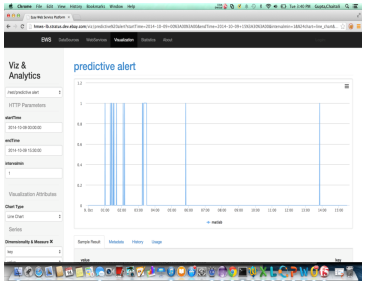


PCA (Principal Component Analysis)



Normal (Green) and Abnormal (Red)
Data and Probability Distribution and Threshold Selection





Anomaly Metric Predictive Detection Case Study

Agenda

- About Apache Eagle
- Architecture
- **Ecosystem**
- Q & A

Apache Eagle Ecosystem

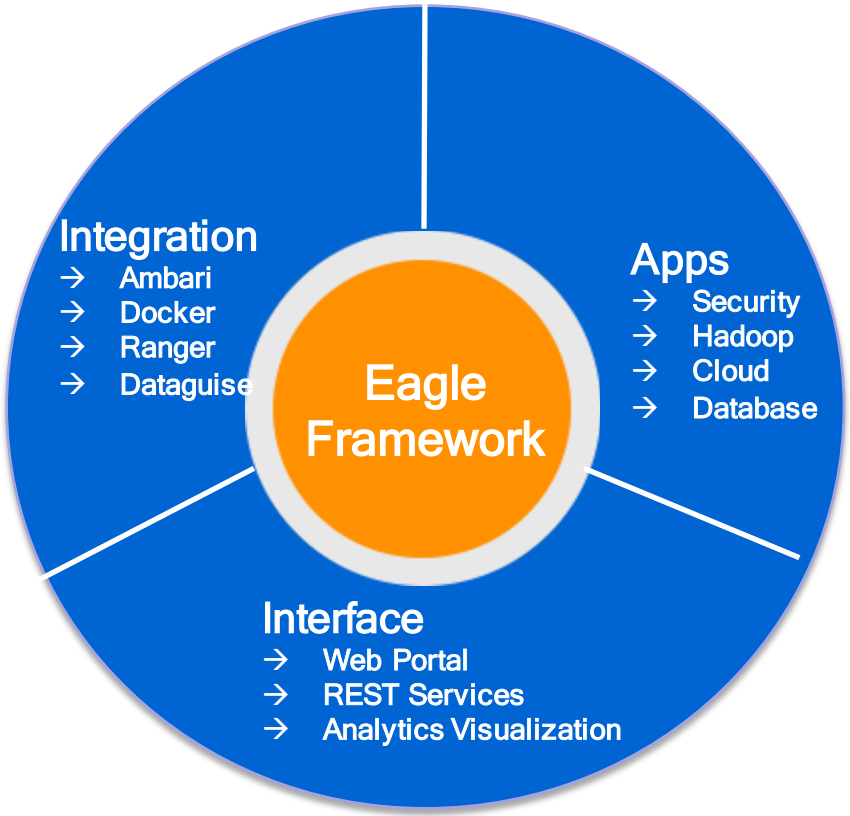
Eagle Framework
Distributed real-time framework for efficiently developing highly scalable monitoring applications

Eagle Apps
Securiy / Hadoop / Cloud / Database

Eagle Interface
REST Service / Management UI / Customizable Analytics Visualization

Eagle Integration
Ambari / Docker / Ranger / Dataguise

Open Source
Community-driven and Cross-community cooperation



Apache Eagle Ecosystem - Security

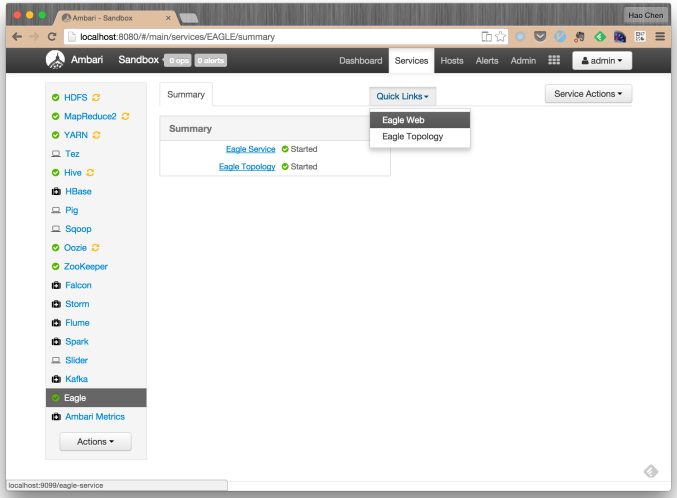
How to Secure Hadoop in Realtime?

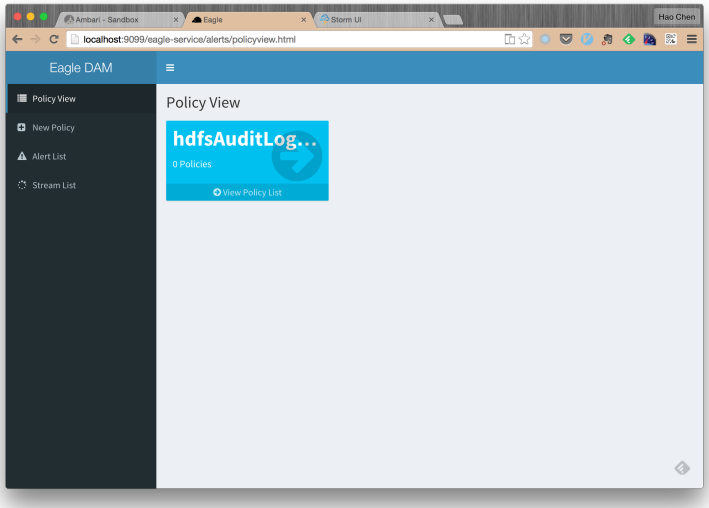
- Apache Eagle
- Apache Ranger
- Apache Knox
- Dataguise

Apache Eagle Ecosystem - Hadoop

Eagle in Apache Amabri: natively be part of hadoop ecosystem

Apache Ambari





Apache Eagle Ecosystem - Docker

Eagle in Docker: natively fly on Cloud/Container

Powered of

- STORM
- KAFKA
- ZOOKEEPER
- HBASE
- HADOOP
- ...

15 + 1 = 1

git clone apache/incubator-eagle eagle-docker

docker pull apacheeagle

Apache Eagle Ecosystem - Open Source

If you want to go fast, go alone.
If you want to go far, go together.

-- African Proverb

Learn more about Apache Eagle

- EAGLE: USER PROFILE-BASED ANOMALY DETECTION IN HADOOP CLUSTER (IEEE)
- EAGLE: DISTRIBUTED REALTIME MONITORING FRAMEWORK FOR HADOOP CLUSTER

Q & A

ebay

http://eagle.incubator.apache.org

apache/incubator-eagle

@TheApacheEagle

@ApacheEagle