THIS IS NOT A CONTRIBUTION

COMMUNITY
THE ASF CONFERENCE
CODE

Revisit Spark Standalone Cluster in Apache Spark 4 with K8s

Dongjoon Hyun, Liang-Chi Hsieh

Agenda

Apache Spark 4 with K8s

Security

REST Submission API

Live Web UI and Customization

Job Management

Cluster Management

Apache Spark 4 with K8s

Apache Spark Release History

Spark 4: The Next Major Release

- Spark 1: 2014.05 (1.0.0) ~ 2016.11 (1.6.3)
- Spark 2: 2016.07 (2.0.0) ~ 2021.05 (2.4.8)
- Spark 3: 2020.06 (3.0.0) ~ 2026.xx (3.5.x)
- Spark 4: 2025.02 (4.0.0, NEW)
  - https://spark.apache.org/versioning-policy.html

Apache Spark 4

Notable User-facing Changes

- **Java: Use Java 17 by default and support Java 21**
- **Scala: Use Scala 2.13 and drop Scala 2.12**
- **Python:**
  - **Support Python 3.12 / 3.13 and drop Python 3.8**
  - **Provide pure Python package (*pyspark-connect)***
- **SQL:**
  - **ANSI mode by default**
  - **Support SQL Script**
- **R: Deprecated**

Apache Spark 4

SPARK-44111: Prepare Apache Spark 4.0.0

- Spark 4 has much more interesting features
  - e.g. *Vectorized IO:* a cross-project collaboration based on Apache Hadoop 3.4, Apache ORC 2, and Apache Parquet 1.14
- Track the progress and details
  - https://issues.apache.org/jira/browse/SPARK-44111

Apache Spark 4

Second Preview (September 2024)

- **Download and documentation**
  - **https://downloads.apache.org/spark/spark-4.0.0-preview2/**
  - **https://spark.apache.org/docs/4.0.0-preview2/**

- **Key changes in resource manager perspective**
  - **Drop *Apache Mesos* support**
  - **Support K8s v1.29 ~ v1.31 and drop 1.28 and below support**
  - **Enhance K8s support in many ways including *Spark Standalone Cluster***

Three ways for better K8s support

Over 300 JIRA issues (SPARK-49524, SPARK-45923, SPARK-45869)

- **Improve K8s support**
  - **Improve the existing direct Spark job submissions and management**

Three ways for better K8s support

Over 300 JIRA issues (SPARK-49524, SPARK-45923, SPARK-45869)

- Improve K8s support
  - Improve the existing direct Spark job submissions and management
- **Spark Kubernetes Operator**
  - **Add Apache Spark K8s Operator for *SparkApp*/*SparkCluster* CRDs**

Three ways for better K8s support

Over 300 JIRA issues (SPARK-49524, SPARK-45923, SPARK-45869)

- Improve K8s support
  - Improve the existing direct Spark job submissions and management
- Spark Kubernetes Operator
  - Add Apache Spark K8s Operator for *SparkApp*/*SparkCluster* CRDs
- **Revisit and Improve Spark Standalone Cluster**
  - **Run *Spark Standalone Cluster* on K8s more seamlessly and safer**
  - **Recommended for repetitive short-life jobs with high frequency submissions**

How to use Spark Cluster in K8s

SPARK-45923 Spark Kubernetes Operator

```
$ helm install spark-kubernetes-operator \
  https://nightlies.apache.org/spark/charts/spark-kubernetes-operator-0.1.0-SNAPSHOT.tgz

$ kubectl apply -f https://raw.githubusercontent.com/apache/spark-
kubernetes-operator/main/examples/cluster-java21.yaml

$ kubectl get sparkcluster
  NAME              CURRENT STATE     AGE
  cluster-java21    RunningHealthy    59s

$ kubectl delete sparkcluster cluster-java21
```

Example *SparkCluster* YAML file

```yaml
apiVersion: spark.apache.org/v1alpha1
kind: SparkCluster
metadata:
  name: cluster-java21
spec:
  runtimeVersions:
    sparkVersion: "4.0.0-preview2"
  clusterTolerations:
    instanceConfig:
      initWorkers: 3
      minWorkers: 3
      maxWorkers: 3
  sparkConf:
    spark.kubernetes.container.image: "apache/spark:4.0.0-preview2-java21"
    spark.master.rest.enabled: "true"
    spark.master.rest.host: "0.0.0.0"
```

Security

Background

Before Spark 4

- **REST Submission Server had no security feature**
- **Spark UIs have *spark.ui.filters* feature, but no built-in authorization filters**

Related Standards

- RFC 1945: Hypertext Transfer Protocol (HTTP/1.0)
  - Section 10.2 Authorization
- RFC 6750: The OAuth 2.0 Authorization Framework: Bearer Token Usage
- RFC 7519: JSON Web Token (JWT)
- RFC 7515: JSON Web Signature (JWS)
  - Appendix A. JWS Examples

Example (1/4)

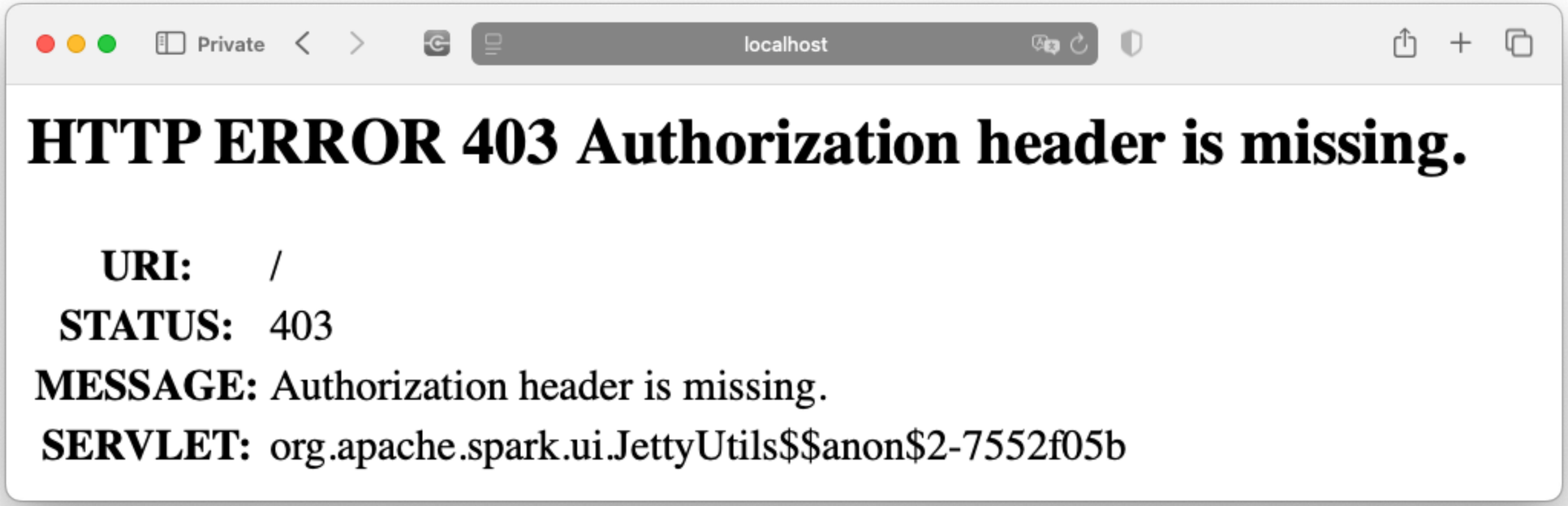
Prepare SECRET_KEY

```
$ jshell
|  Welcome to JShell -- Version 21.0.4
|  For an introduction type: /help intro

jshell> java.util.Base64.getUrlEncoder().encodeToString(
"Visit https://spark.apache.org to download Apache Spark.".getBytes())

$1 ==>
"VmlzaXQgaHR0cHM6Ly9zcGFyay5hcGFjaGUub3JnIHRvIGRvd25sb2FkIEFwYWNoZSBTcGFyay4="
```

Example (2/4)

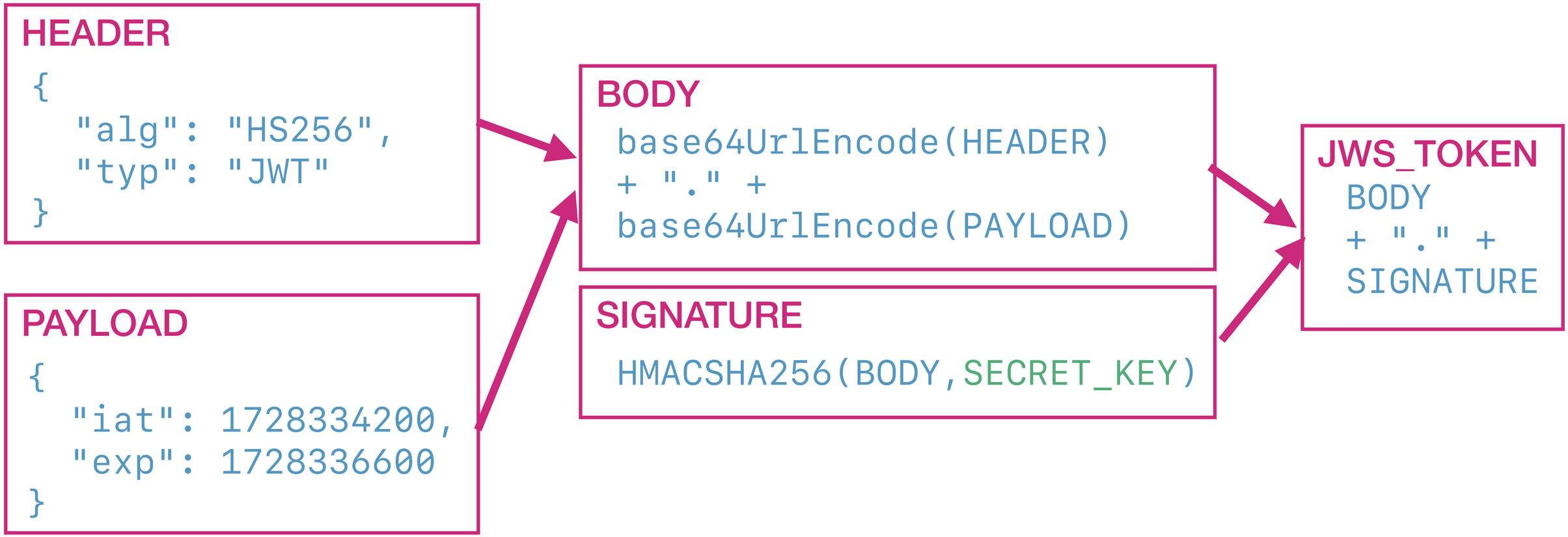
Run *spark-shell* and visit Spark Driver UI (http://localhost:4040)

```
bin/spark-shell \
-c spark.ui.filters=org.apache.spark.ui.JWSFilter \
-c spark.org.apache.spark.ui.JWSFilter.param.secretKey=$SECRET_KEY
```



Example (3/4)

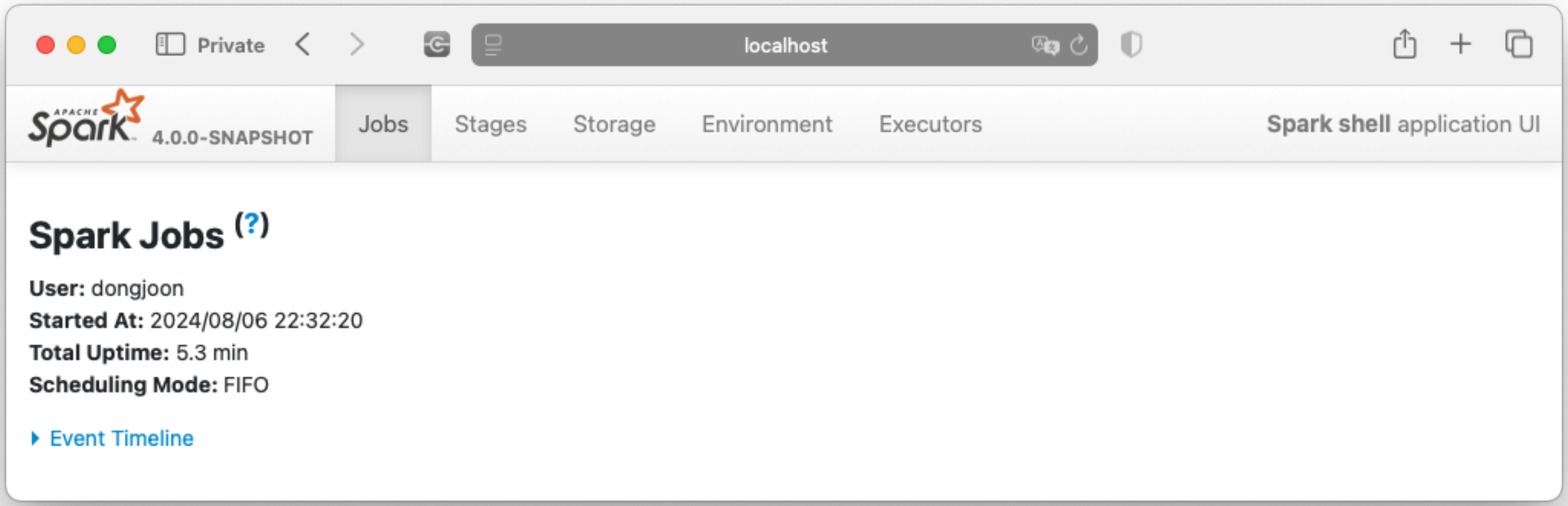
Issue a cryptographically signed JSON Web Token



Example (4/4)

Visit with *Authorization* header

```
$ curl -v -H "Authorization: Bearer $JWS_TOKEN" http://localhost:4040
...
HTTP/1.1 302 Found ...
```



Authorization via a signed JSON Web Token

Support JWSFilter (SPARK-49090)

- **How to use**

```
spark.ui.filters=org.apache.spark.ui.JWSFilter
spark.org.apache.spark.ui.JWSFilter.param.secretKey=$SECRET_KEY
```

- **Protected UIs**
  - Driver UI (port: 4040)
  - Master UI (port: 8080)
  - Worker UI (port: 8081)
  - History Server UI (port: 18080)

Authorization via a signed JSON Web Token

Support *spark.master.rest.filters* (SPARK-49103)

- **How to use**

```
spark.master.rest.enabled=true
spark.master.rest.filters=org.apache.spark.ui.JWSFilter
spark.org.apache.spark.ui.JWSFilter.param.secretKey=$SECRET_KEY
```

- **Protected UI**
  - **Master REST API (port: 6066)**

Make server logs more secure

Redact *Spark Command* line in launcher logs (SPARK-49197)

- **Before**

```
Spark Command: ...
-Dspark.org.apache.spark.ui.JWSFilter.param.secretKey=ABC_PLAIN_KEY
```

- **After**

```
Spark Command: ...
-Dspark.org.apache.spark.ui.JWSFilter.param.secretKey=****(redacted)
```

K8s Pod Liveness and Readiness Probes

Use *httpHeaders*

```
livenessProbe:
  httpGet:
    path: /
    port: 8080
    httpHeaders:
    - name: Authorization
      value: "Bearer <JWS_TOKEN>"
```

```
readinessProbe:
  httpGet:
    path: /v1/submissions/readyz
    port: 6066
    httpHeaders:
    - name: Authorization
      value: "Bearer <JWS_TOKEN>"
```

Support read-only Master UI

SPARK-46899

- **How to block POST APIs from Master UI**

```
spark.ui.killEnabled=false
spark.decommission.enabled=false
```

REST Submission API

Background

REST Submission API before Spark 4

- Was not documented
- Was not working with K8s port-forwarding
- Behaved differently from normal submission
- Needs more features
- Needs simplification

Documentation and examples

Apache Spark and Apache Spark K8s repository

- Document *REST API* for Spark Standalone Cluster (SPARK-46095)
- Add *submit_pi.sh* REST API example (SPARK-49108)
- Document *SparkCluster* and add *submit-pi-to-prod.sh* (SPARK-49340)
- Add *JavaSparkSQLCli* example (SPARK-45145)
- Add *SparkApplication sql.yaml* example (SPARK-49270)
- Add a symbolic link *spark-examples.jar* in K8s (SPARK-45497)

Add *submit_pi.sh* REST API example (SPARK-49108)

```
curl -XPOST http://$SPARK_MASTER:6066/v1/submissions/create --data \
'{
  "action": "CreateSubmissionRequest",
  "appResource": "",
  "sparkProperties": {
    "spark.submit.deployMode": "cluster",
    "spark.app.name": "SparkPi",
    "spark.executor.cores": "1",
    "spark.cores.max": "2"
  },
  "clientSparkVersion": "",
  "mainClass": "org.apache.spark.deploy.SparkSubmit",
  "environmentVariables": {
    "MASTER": "spark://'$SPARK_MASTER':7077"
  },
  "appArgs": [ "'$PYTHON_FILE'" ]
}'
```

Allow REST API server to choose a host to listen

Support *spark.master.rest.host* (SPARK-46320)

- **How to use**

```
spark.master.rest.enabled=true
spark.master.rest.host=0.0.0.0
```

- **The above allows K8s *port-forward* without side-effects**

```
kubectl port-forward prod-master-0 6066
```

API Improvement

- /create (Improved: SPARK-49033, SPARK-49034, SPARK-49845)
- /kill
- /status
- /readyz (New: SPARK-46368 for K8s health-check)
- /killall (New: SPARK-45843)
- /clear (New: SPARK-45819)

Simplify Submission Request Form

Make *appArgs* and *environmentVariables* optional (SPARK-49845)

- **They were mandatory due to Apache Mesos's requirements**
- **From Spark 4, we can omit them**

```
{
  "action": "CreateSubmissionRequest",
  "appResource": ...,
  "appArgs": ...
  "clientSparkVersion": ...,
  "environmentVariables": ...,
  "sparkProperties": ...,
  "mainClass": "org.apache.spark.deploy.SparkSubmit"
}
```

Server-side environment variable replacement

SPARK-49033 and SPARK-49034

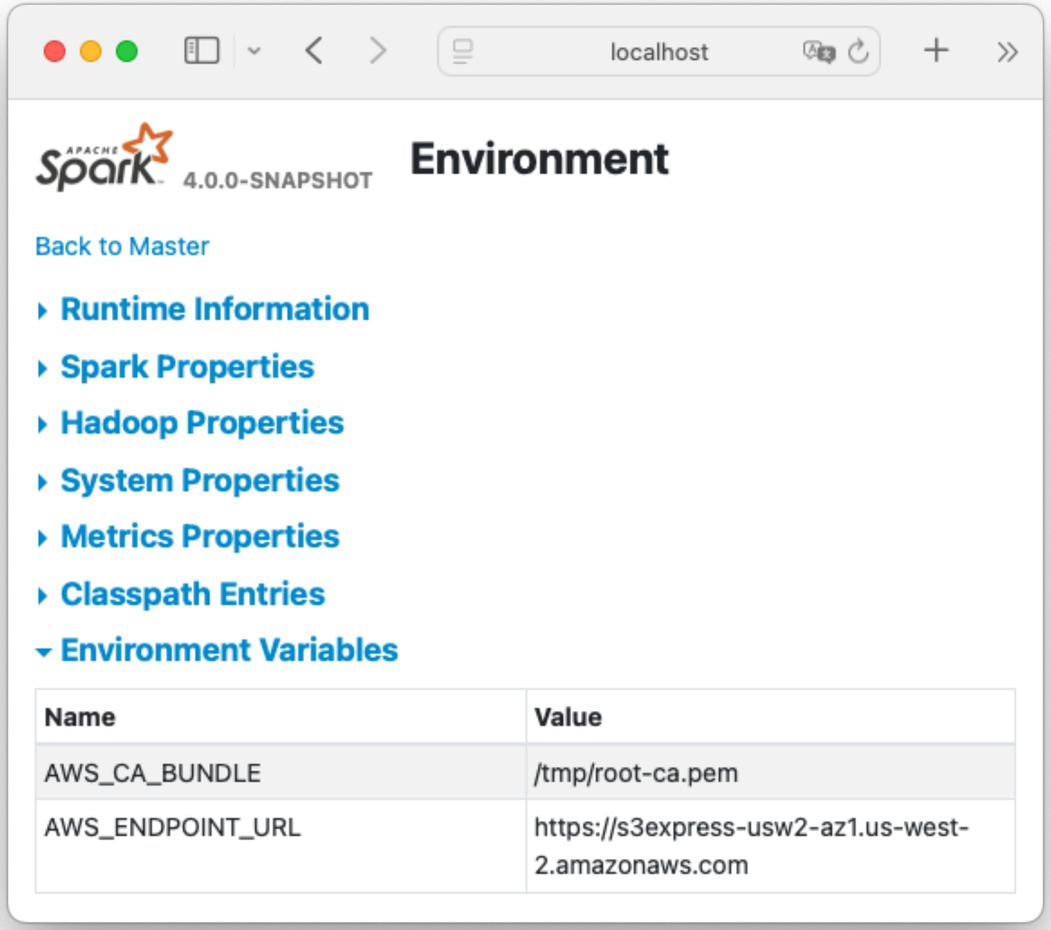
```
"sparkProperties": {
  "spark.hadoop.fs.s3a.endpoint": "{{AWS_ENDPOINT_URL}}",
  ...
},

"environmentVariables": {
  "MASTER": "{{MASTER}}"
  "AWS_CA_BUNDLE": "{{AWS_CA_BUNDLE}}"
},
```

Show *Environment Variables* of Master UI

SPARK-49206

- **How to use**

```
spark.master.ui.visibleEnvVarPrefixes=AWS_
```



Improve Java support

SPARK-45197 Add *JavaModuleOptions* to drivers automatically

- **Since Spark 3.3 (SPARK-36796), *SparkContext* has been adding *JavaModuleOptions* by default to avoid Java module error**
- **From Spark 4, Spark Master adds it automatically for all REST submissions to avoid the following Java module errors**

```
java.lang.IllegalAccessError: ...
module java.base does not export ...
```

Agenda

Apache Spark 4 with K8s

Security

REST Submission API

Live Web UI and Customization

Job Management

Cluster Management

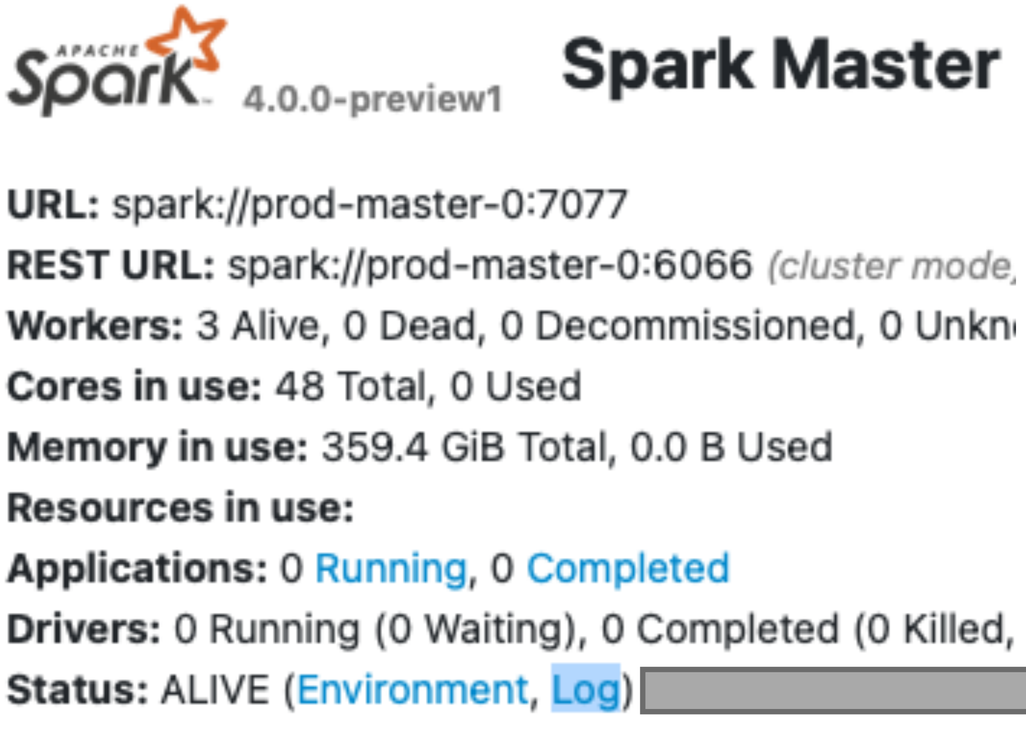
Live Web UI and Customization

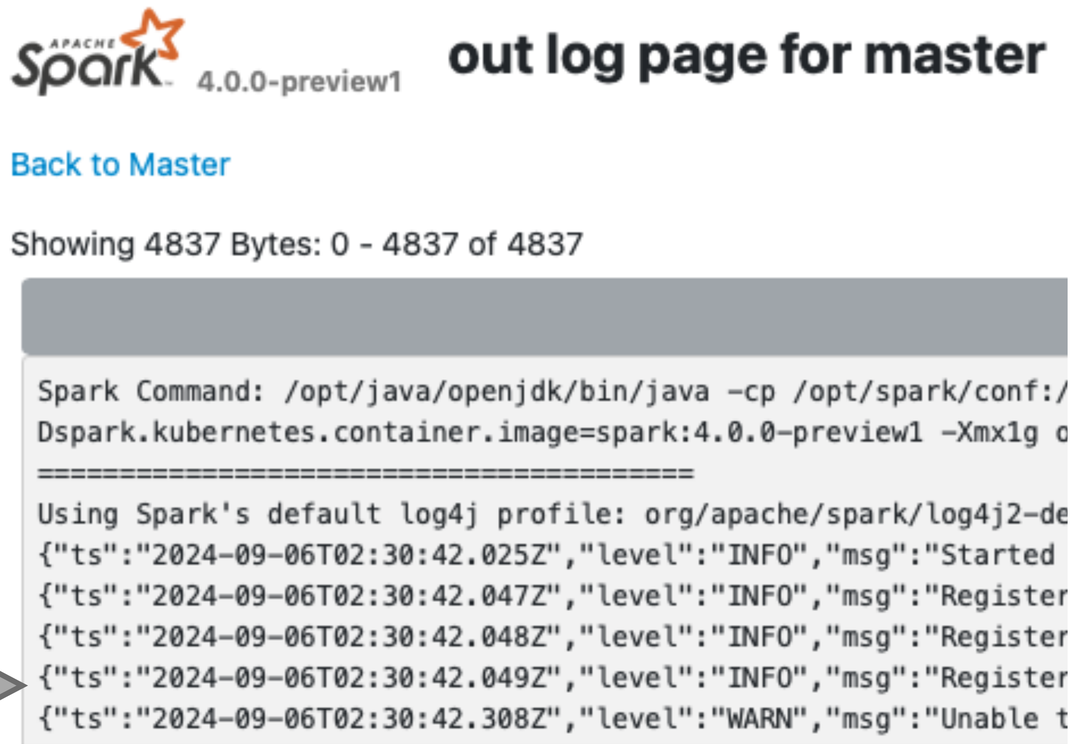
Live Log UIs

- Master Live Log UI (SPARK-46870)
- Worker Live Log UI (SPARK-46868)
- History Server Live Log UI (SPARK-46903)
- Driver Live Log UI (SPARK-44214)

Live Log UIs (Cont.)

- **Master Live Log UI (SPARK-46870)**



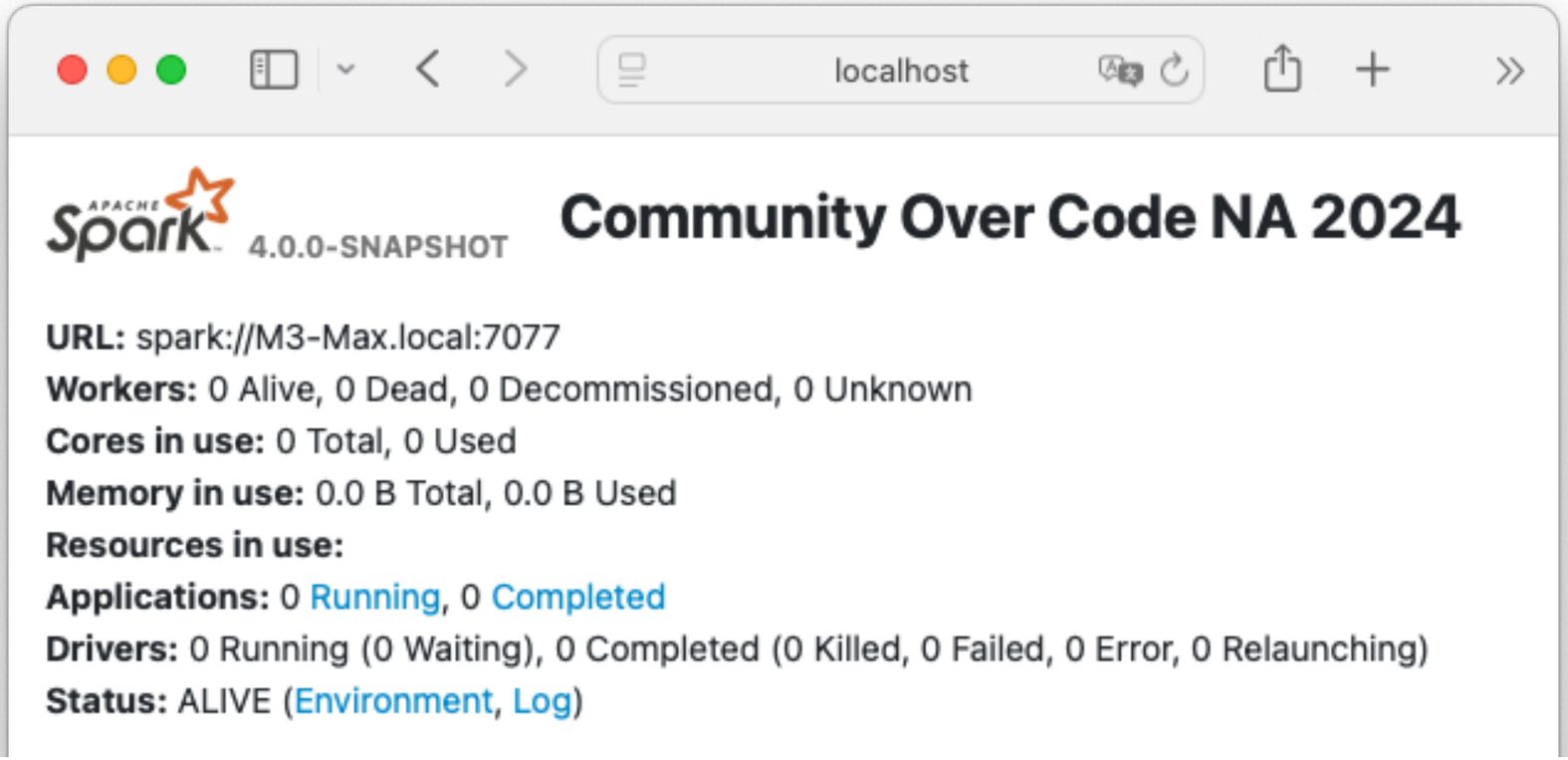


Master UI

- Support custom title of Master UI (SPARK-49007)
- Support *spark.master.ui.historyServerUrl* in *ApplicationPage* (SPARK-45774)
- Show a summary of workers (SPARK-46292)
- Add *Environment* page (SPARK-47894)
- Improve JSON API to support top-level filtering (SPARK-45474)
- Improve JSON API to support */json/clusterutilization* (SPARK-46883)

Master UI (Cont.)

spark.master.ui.title="Community Over Code NA 2024"

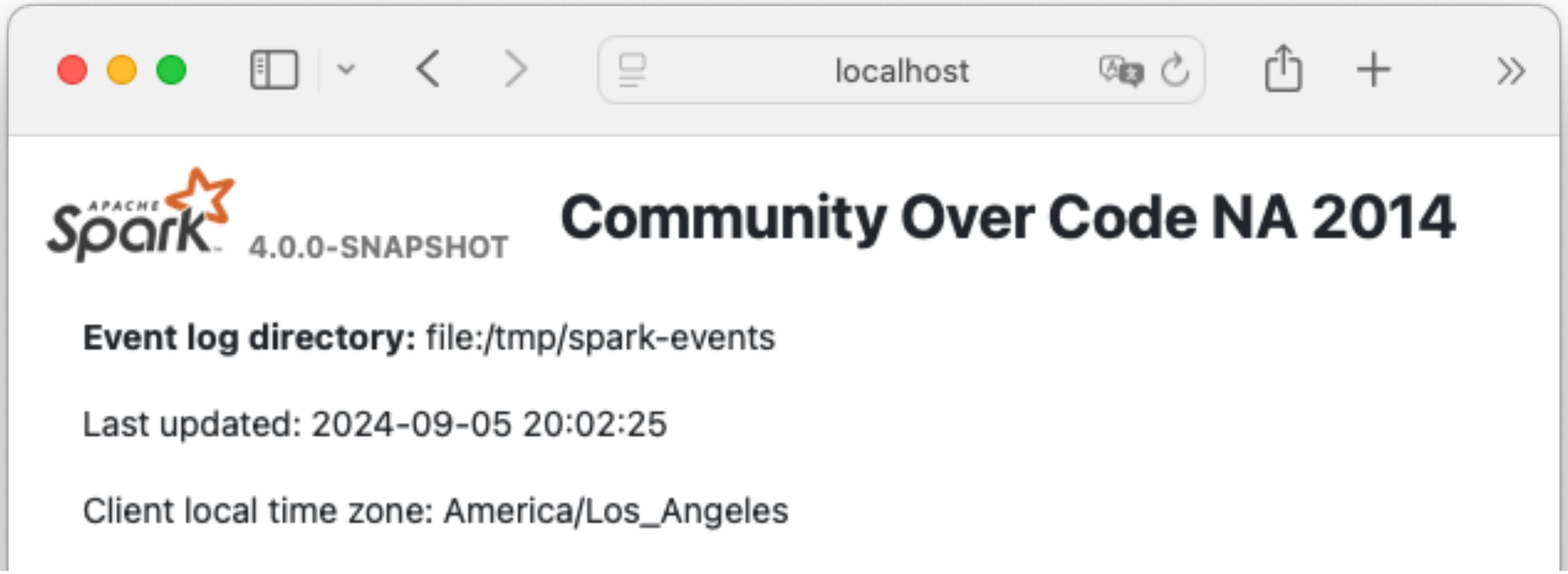


History Server UI

- Support custom title of History Server UI (SPARK-49128)
- Show driver log location (SPARK-46907)

History Server UI (Cont.)

spark.history.ui.title="Community Over Code NA 2024"



Various ID customization

- Support *spark.worker.idPattern* (SPARK-45867)
- Support *spark.deploy.driverIdPattern* (SPARK-45753)
- Support *spark.deploy.appIdPattern* (SPARK-45754)
- Support *spark.deploy.appNumberModulo* to rotate app number (SPARK-45785)
- Support *spark.master.useAppNameAsAppId.enabled* (SPARK-45756)

Job Management

Support *spark.deploy.maxDrivers*

SPARK-45174

- **Like *spark.mesos.maxDrivers*, allow users to limit the maximum number of running drivers in a Spark cluster**

```
spark.deploy.maxDrivers=100 (default: Int.MaxValue)
```

Support *spark.driver.timeout*

SPARK-47207

- **Terminate Spark drivers with *SparkExitCode.DRIVER_TIMEOUT* (124) if it runs over the limit**

```
spark.deploy.timeout=10min (default: 0min)
spark.plugins=org.apache.spark.deploy.DriverTimeoutPlugin
```

Rename *spark.deploy.(spreadOut → spreadOutApps)*

SPARK-46797

- **Whether to spread apps out across nodes or consolidate onto less nodes**

```
spark.deploy.spreadOutApps=false (default: true)
```

Support *spark.deploy.spreadOutDrivers*

SPARK-46800

- **Like *spark.deploy.spreadOutApps*, allow users to control Spark driver distribution in Spark clusters**

```
spark.deploy.spreadOutDrivers=false (default: true)
```

Support *spark.deploy.workerSelectionPolicy*

SPARK-46881

- **A policy to assign executors on one of the assignable workers**

Policy	Description
CORES_FREE_ASC	Choose a worker with the least free cores
CORES_FREE_DESC	Choose a worker with the most free cores (default)
MEMORY_FREE_ASC	Choose a worker with the least free memory
MEMORY_FREE_DESC	Choose a worker with the most free memory
WORKER_ID	Choose a worker with the smallest worker id

Cluster Management

Spark Master HA (1/4)

Use HTTP response for Readiness Check (SPARK-46368)

- **How to use**

```
curl -vv http://localhost:6066/v1/submissions/readyz
```

```
< HTTP/1.1 200 OK
{
  "action" : "ReadyzResponse",
  "message" : "",
  "serverSparkVersion" : "*",
  "success" : true
```

```
< HTTP/1.1 503 Service Unavailable
{
  "action" : "ErrorResponse",
  "message" : "Master is not ready.",
  "serverSparkVersion" : "*"
}
```

Spark Master HA (2/4)

Improve recovery time

- Support *spark.deploy.recoveryTimeout* (SPARK-46348)
- Improve *Master* to recover quickly in case of zero workers and apps (SPARK-46664)

Spark Master HA (3/4)

Support compression in persistence engine (SPARK-46216)

- **How to use**

```
spark.deploy.recoveryCompressionCodec=snappy
```

- **Up to 4x faster than *FileSystemPersistenceEngine* without compression**

Spark Master HA (4/4)

Support *RocksDB* as a persistence engine (SPARK-46258)

- **How to use**

```
spark.deploy.recoveryMode=ROCKSDB
```

- **10x faster than *FileSystemPersistenceEngine* without compression(= Spark 3)**
  - **2.5x faster than *FileSystemPersistenceEngine* with the best compression**

Use with HorizontalPodAutoscaler

With new Spark job management configs

- ***Collaborate Spark Job management with K8s HPA***

```
spark.deploy.spreadOutDrivers=false
spark.deploy.spreadOutApps=false
spark.deploy.workerSelectionPolicy=WORKER_ID
```

- ***Use Apache Spark Kubernetes Operator to handle HPA generation and management***
  - ***https://github.com/apache/spark-kubernetes-operator/***

Thank you!