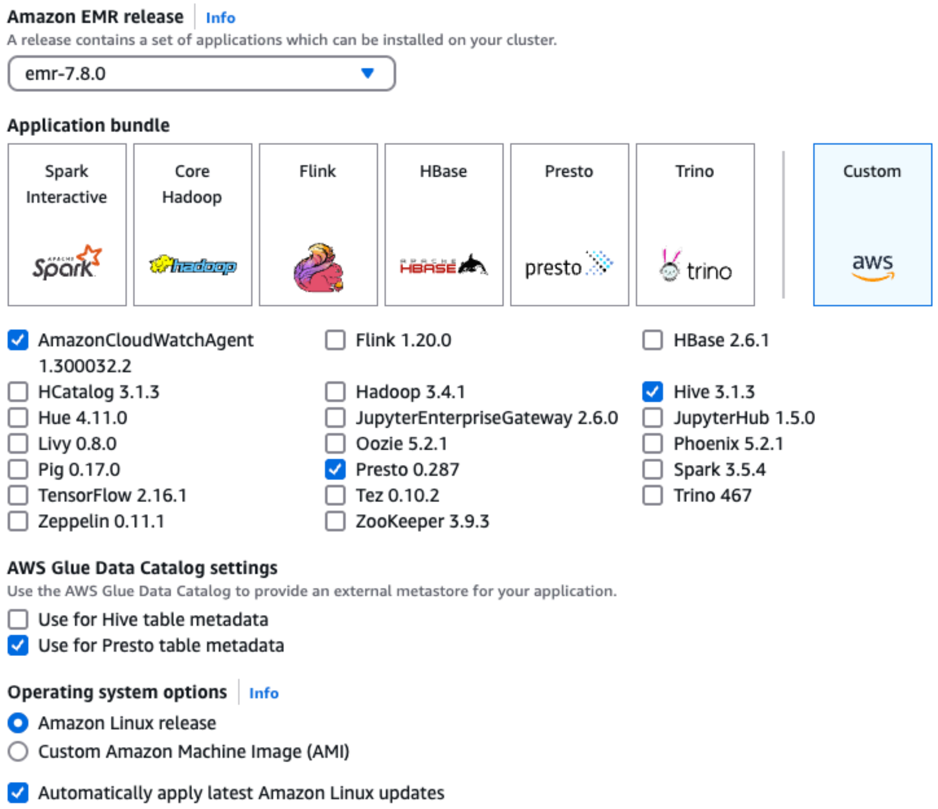
1. Create a Presto cluster
    a. duplicate



Amazon EMR release Info

A release contains a set of applications which can be installed on your cluster.

emr-7.8.0 ▼

Application bundle

☑ AmazonCloudWatchAgent 1.300032.2
☐ HCatalog 3.1.3
☐ Hue 4.11.0
☐ Livy 0.8.0
☐ Pig 0.17.0
☐ TensorFlow 2.16.1
☐ Zeppelin 0.11.1
☐ Flink 1.20.0
☐ Hadoop 3.4.1
☐ JupyterEnterpriseGateway 2.6.0
☐ Oozie 5.2.1
☑ Presto 0.287
☐ Tez 0.10.2
☐ ZooKeeper 3.9.3
☐ HBase 2.6.1
☑ Hive 3.1.3
☐ JupyterHub 1.5.0
☐ Phoenix 5.2.1
☐ Spark 3.5.4
☐ Trino 467

AWS Glue Data Catalog settings

Use the AWS Glue Data Catalog to provide an external metastore for your application.

☐ Use for Hive table metadata
☑ Use for Presto table metadata

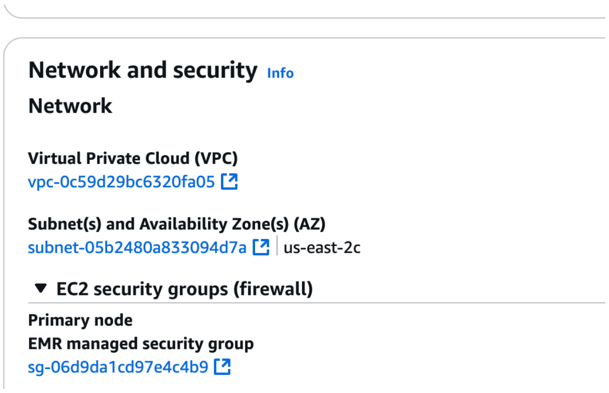
Operating system options Info

● Amazon Linux release
○ Custom Amazon Machine Image (AMI)

☑ Automatically apply latest Amazon Linux updates

    b.
    c. Add ssh in “Security configuration and EC2 key pair” section (should download a .pem file to use in step 3)
    d. Create a new Service Role
    e. Create new Instance Profile with access to all buckets
        i. “All S3 buckets in this account with read and write access”
    f. Be sure to add Permissions to service account and instance profile. Instance profile is not linked from the main page so you can find it here. I do this after the cluster is created.
        i. AmazonS3TablesFullAccess
        ii. AmazonS3FullAccess
        iii. AWSGlueConsoleFullAccess
2. After cluster is created NEED to add inbound routes
    a. Go to primary nodes security group

i.



Network and security Info

Network

Virtual Private Cloud (VPC)
vpc-0c59d29bc6320fa05

Subnet(s) and Availability Zone(s) (AZ)
subnet-05b2480a833094d7a | us-east-2c

▼ **EC2 security groups (firewall)**

Primary node
EMR managed security group
sg-06d9da1cd97e4c4b9

b. Add inbound routes

i. For Presto

1. Type - CUSTOM TYPE
2. Port - 8889
3. Source - My IP

ii. For SSH

1. Type - CUSTOM TYPE
2. Port - 22
3. Source - My IP

3. Running queries via ssh on terminal

a. Need to change read/write access of pem file to be secure enough

i. chmod 600 <prem_file_name>.pem

b. Ssh

i. ssh -i <prem_file_name>.pem hadoop@<primary node dns>

c. Set configs

i. cd /usr/lib/presto/etc

1. To create if doesn't exist: sudo mkdir -p /usr/lib/presto/etc

ii. sudo nano /usr/lib/presto/etc/config.properties

1.

```
coordinator=true
node-scheduler.include-coordinator=true
http-server.http.port=8889
query.max-memory=5GB
query.max-memory-per-node=1GB
discovery-server.enabled=true
discovery.uri=http://localhost:8889
```

iii. sudo nano /usr/lib/presto/etc/jvm.config

1.

```
-server
-Xmx4G
-XX:+UseG1GC
-XX:G1HeapRegionSize=32M
-XX:+UseGCOverheadLimit
-XX:+ExplicitGCInvokesConcurrent
```

```
-XX:+HeapDumpOnOutOfMemoryError
-XX:+ExitOnOutOfMemoryError
```

iv. sudo nano /usr/lib/presto/etc/node.properties

1.

```
node.environment=production
node.id=presto-1
node.data-dir=/var/presto/data
```

d. Set up Glue

i. sudo mkdir -p /usr/lib/presto/etc/catalog

ii. sudo nano /usr/lib/presto/etc/catalog/hive.properties

1.

```
connector.name=hive-hadoop2
hive.metastore=glue
hive.metastore.glue.region=us-east-2
hive.metastore-glue.datacatalog.enabled=true
```

e. Start Presto

i. sudo /usr/lib/presto/bin/launcher start

ii. Status:

1. sudo /usr/lib/presto/bin/launcher status

iii. Logs:

1. tail -f /var/log/presto/server.log

iv. Check Port 8889

1. netstat -tulnp | grep 8889

f. Enter into presto - NOTE: command might fail at first give it time

i. presto-cli --server localhost:8889 --catalog hive --schema default

g. Run SQL to verify database is there

i. ~~SHOW tables;~~

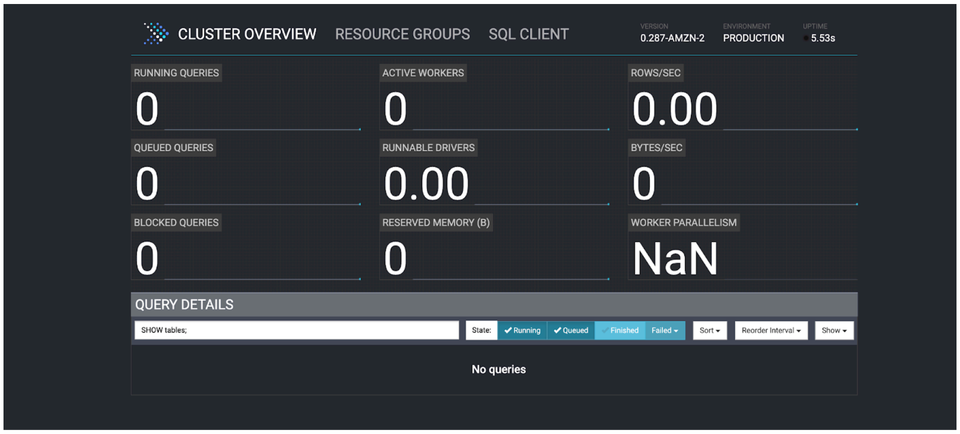
ii. SHOW SCHEMAS FROM hive;

4. Run queries via web interface

a. Go to webinterface

i. http://<primary node dns>:8889

ii.



VERSION 0.287-AMZN-2
ENVIRONMENT PRODUCTION
UPTIME 5.53s

RUNNING QUERIES 0
ACTIVE WORKERS 0
ROWS/SEC 0.00
QUEUED QUERIES 0
RUNNABLE DRIVERS 0.00
BYTES/SEC 0
BLOCKED QUERIES 0
RESERVED MEMORY (B) 0
WORKER PARALLELISM NaN

SHOW tables;

No queries

b. In query details, should be able to run SQL

        i. SHOW tables;

5. Running a Step
    a. Use jar at jars/args-SNAPSHOT
    b. Use args: l_table r_table s3a://584spark-east2/datasets/L10_R10_M1-1_RS1000_SF/JOIN/ <host ip you have to get>
        i. Make sure you change the directory based on what dataset you are running
6. Misc
    a. Jar must include the Presto JDBC Driver
        i.

```
    <!-- Presto JDBC driver -->
     <dependency>
        <groupId>io.prestosql</groupId>
        <artifactId>presto-jdbc</artifactId>
        <version>350</version> <!-- Use the latest or whatever works for your cluster -->
     </dependency>
```

    b. Jar must include the shade plugin
        i.

```
<!-- Shade plugin to create fat JAR -->
        <plugin>
            <groupId>org.apache.maven.plugins</groupId>
            <artifactId>maven-shade-plugin</artifactId>
            <version>3.2.4</version>
            <executions>
            <execution>
                <phase>package</phase>
                <goals>
                <goal>shade</goal>
                </goals>
                <configuration>
                <transformers>
                    <!-- Ensures proper META-INF/services is merged -->
                    <transformer
implementation="org.apache.maven.plugins.shade.resource.ServicesResourceTransformer"/>
                </transformers>
                </configuration>
            </execution>
            </executions>
        </plugin>
```

    c. Java must connect to internal IP of the primary node, not the public. Can find it by sshing into the primary node and running….
        i. hostname -f
        ii. JDBC url
            1. jdbc:presto://<internal ip>/hive/default