BIG DATA

Hands on AWS

Identity and Access Management

Identity and Access Management (IAM)

- Web service that controls fine-grained access to AWS resources
- IAM controls who is authenticated and authorized to use resources

IAM user

- Unique identity recognized by AWS services and applications
- Similar to user in an operating system like Windows or UNIX

Identity and Access Management

IAM role

- Set of policies for making AWS service requests
- Trusted entities (e.g., such as IAM users) assume roles
  - Delegate access with defined permissions to trusted entities
  - There is no limit to the number of IAM roles a user can assume

User vs role

- User has permanent long-term credentials and is used to directly interact with AWS services
- Role does not have credentials and cannot make direct requests to AWS services
- Roles are assumed by authorized entities, such as IAM users

Identity and Access Management

Alice (i.e., an IAM user) is a firewoman

- She is the same person with or without her turnout gear
- As a firewoman (i.e., a role)
  - If she speeds to a house fire and passes a police officer, he isn't going to give her a ticket
  - In her role as a *firewoman*, she is allowed to speed to the house fire
- As a private citizen (i.e., another role)
  - When she is off duty, if she speeds past that same police officer, he's going to give her a ticket
  - In her role as a *private citizen*, she is not allowed to speed

AWS

Amazon Web Services (AWS) is a public-cloud platform

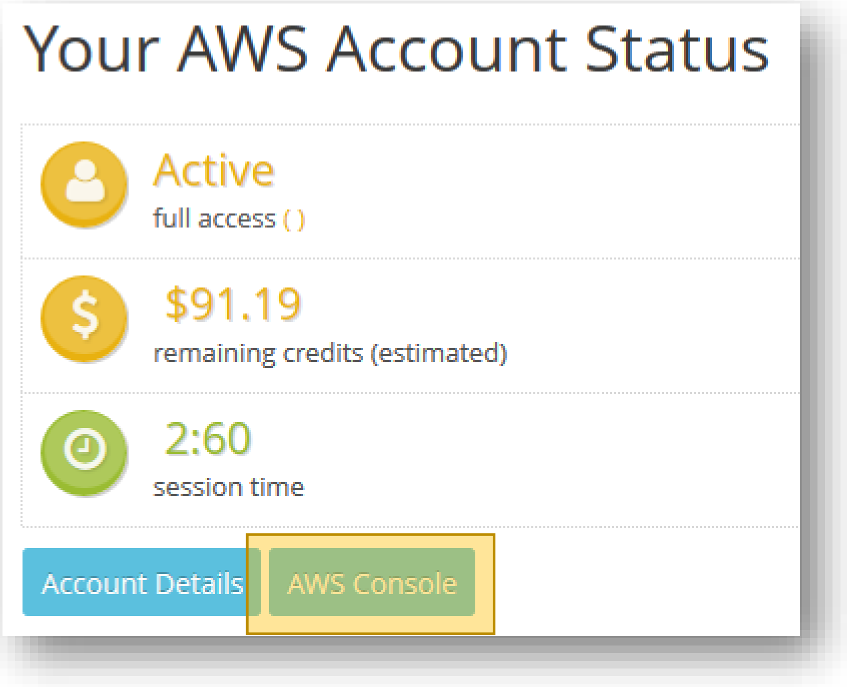
Services can be accessed in multiple ways

- Web GUI: intuitive point and click access without any programming
  - Intuitive interfaces is part of the attraction of cloud services
  - Tedious if the same actions must be performed repeatedly
- (REST) Application programming interface (API)
  - Permits requests to be transmitted via Hypertext Transfer Protocol (HTTPS)
- Software development kits (SDKs) that you install on your computer
  - Access from programming languages such as Python, Java, etc.

AWS Web console

We use the AWS Educate program

- Login with the provided account
- You got 150$ to work on AWS services
- Provisioned services charge even if not used



Your AWS Account Status

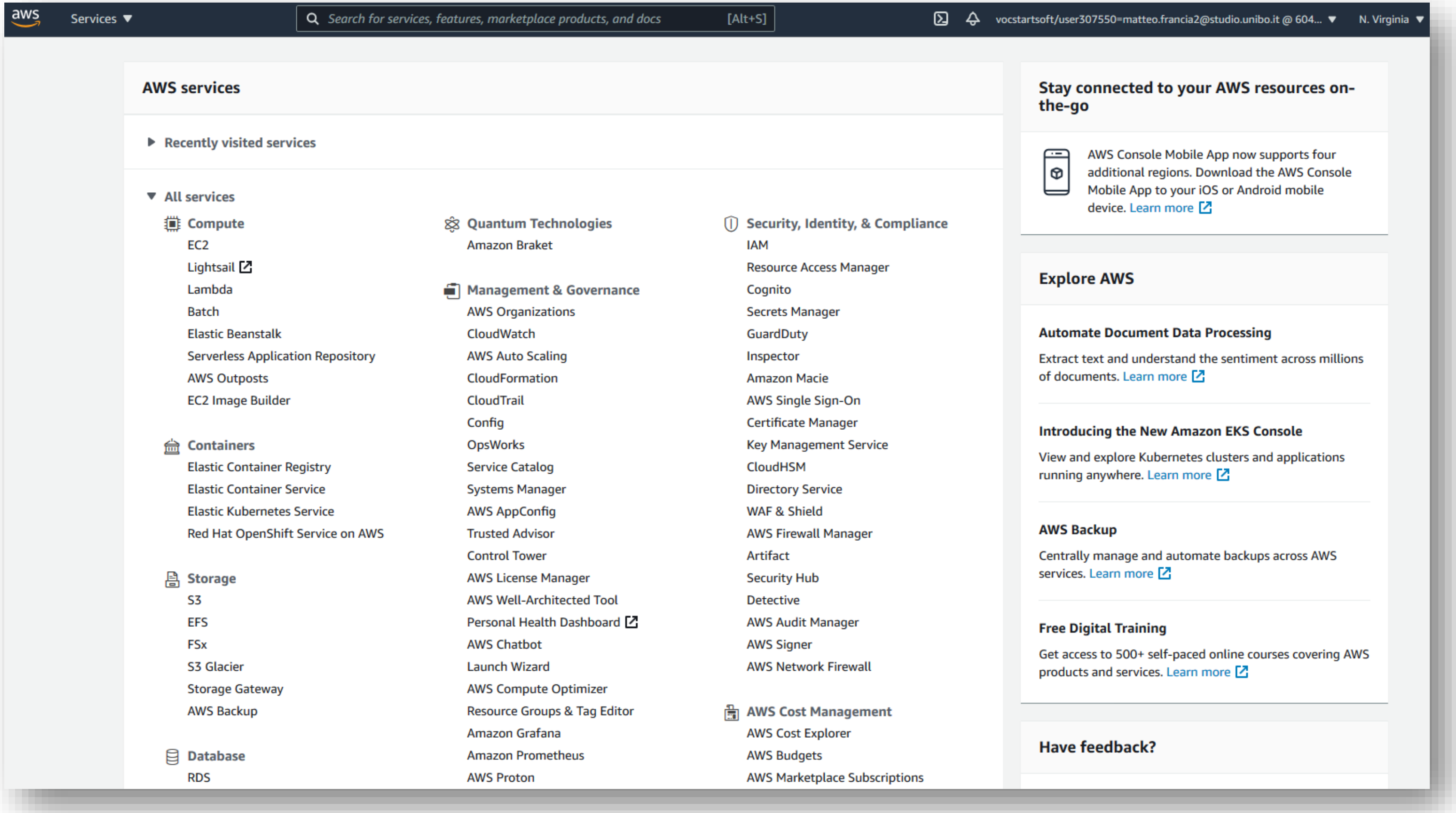
Active
full access ( )

$91.19
remaining credits (estimated)

2:60
session time

https://www.awseducate.com/signin/SiteLogin

AWS Web console



aws
Services ▼
Search for services, features, marketplace products, and docs [Alt+S]
vocstartsoft/user307550=matteo.francia2@studio.unibo.it @ 604... ▼
N. Virginia ▼

AWS services

▶ Recently visited services

▼ All services

Compute
- EC2
- Lightsail
- Lambda
- Batch
- Elastic Beanstalk
- Serverless Application Repository
- AWS Outposts
- EC2 Image Builder

Containers
- Elastic Container Registry
- Elastic Container Service
- Elastic Kubernetes Service
- Red Hat OpenShift Service on AWS

Storage
- S3
- EFS
- FSx
- S3 Glacier
- Storage Gateway
- AWS Backup

Database
- RDS

Quantum Technologies
- Amazon Braket

Management & Governance
- AWS Organizations
- CloudWatch
- AWS Auto Scaling
- CloudFormation
- CloudTrail
- Config
- OpsWorks
- Service Catalog
- Systems Manager
- AWS AppConfig
- Trusted Advisor
- Control Tower
- AWS License Manager
- AWS Well-Architected Tool
- Personal Health Dashboard
- AWS Chatbot
- Launch Wizard
- AWS Compute Optimizer
- Resource Groups & Tag Editor
- Amazon Grafana
- Amazon Prometheus
- AWS Proton

Security, Identity, & Compliance
- IAM
- Resource Access Manager
- Cognito
- Secrets Manager
- GuardDuty
- Inspector
- Amazon Macie
- AWS Single Sign-On
- Certificate Manager
- Key Management Service
- CloudHSM
- Directory Service
- WAF & Shield
- AWS Firewall Manager
- Artifact
- Security Hub
- Detective
- AWS Audit Manager
- AWS Signer
- AWS Network Firewall

AWS Cost Management
- AWS Cost Explorer
- AWS Budgets
- AWS Marketplace Subscriptions

Stay connected to your AWS resources on-the-go

AWS Console Mobile App now supports four additional regions. Download the AWS Console Mobile App to your iOS or Android mobile device. Learn more

Explore AWS

Automate Document Data Processing

Extract text and understand the sentiment across millions of documents. Learn more

Introducing the New Amazon EKS Console

View and explore Kubernetes clusters and applications running anywhere. Learn more

AWS Backup

Centrally manage and automate backups across AWS services. Learn more

Free Digital Training

Get access to 500+ self-paced online courses covering AWS products and services. Learn more

Have feedback?

AWS CLI

CLI interface

- Necessary to install the CLI (version 2)
- See https://docs.aws.amazon.com/cli/latest/userguide/install-cliv2.html

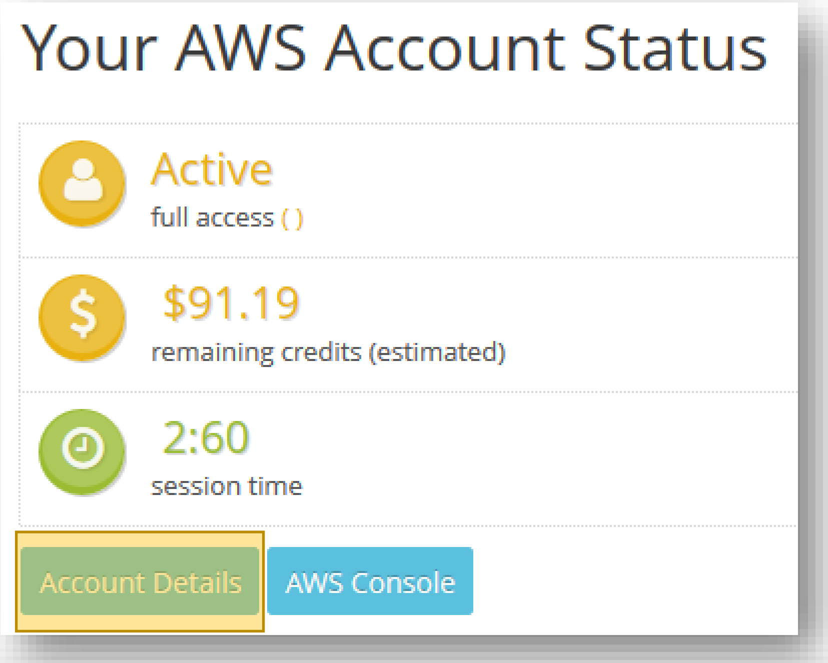
```
Synopsis
********
aws [options] <command> <subcommand> [parameters]


Description
***********
A unified tool to manage your AWS services.
```

https://docs.aws.amazon.com/cli/latest/userguide/install-cliv2-linux.html

AWS CLI

CLI needs credentials to work

- Go back to AWS Educate
- Click on “Account Details”
- Copy the content into the file `~/.aws/credentials`
- Henceforth, we assume that you have set up the credentials file
- Credentials expire after some time; you need a manually refresh

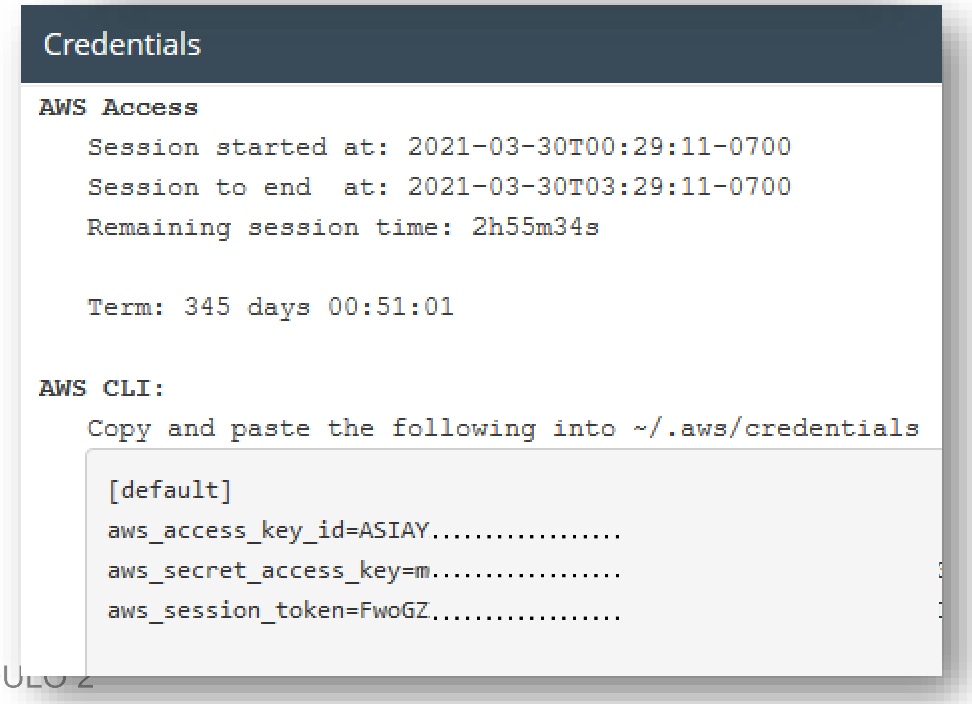


Your AWS Account Status

Active
full access ( )

$91.19
remaining credits (estimated)

2:60
session time



Credentials

```
AWS Access
   Session started at: 2021-03-30T00:29:11-0700
   Session to end  at: 2021-03-30T03:29:11-0700
   Remaining session time: 2h55m34s

   Term: 345 days 00:51:01

AWS CLI:
   Copy and paste the following into ~/.aws/credentials

    [default]
    aws_access_key_id=ASIAY................
    aws_secret_access_key=m................
    aws_session_token=FwoGZ................
```

AWS CLI

Run `` `aws configure` ``

- Confirm AWS Access Key ID (press enter)
- Confirm AWS Secret Access Key (press enter)
- Set Default region name to `` `us-east-1` ``
- Set Default output format to `` `json` ``

It is also possible to configure an AWS profile

- A (named) profile is a collection of settings and credentials
- If profile is specified, its settings and credentials are used to run a command
- When no profile is explicitly referenced, use `` `default` ``
  - We stick to `` `default` ``

Object storage: S3

Create S3 bucket, the following rules apply for naming buckets

- Must be between 3 and 63 characters long
- Can consist only of lowercase letters, numbers, dots (.), and hyphens (-)
- Must be unique within a partition (i.e., a group of regions)

```
$ git clone https://github.com/w4bo/bigdata-aws/
$ cd bigdata-aws/lab01-lambda
$ aws s3api create-bucket --bucket aws-bucket-bigdata2021
$ aws s3 cp datasets/inferno.txt s3://aws-bucket-bigdata2021/inferno.txt
$ aws s3api list-objects --bucket aws-bucket-bigdata2021
```

https://s3.console.aws.amazon.com/s3/home?region=us-east-1#

BIG DATA

Data pipelines on AWS Lambda

Requirements

To start this lecture, you need to

- Activate your AWS Educate account
- Either
  - Install the necessary software
    - git
    - IntelliJ IDEA (with AWS Toolkit and Scala plugins)
    - python
    - java 1.8
    - Docker
    - AWS CLI, AWS SAM CLI
  - Be able to download and run the VM

AWS SAM CLI

Serverless Application Model is a framework to build serverless applications

- A serverless application is a combination of Lambda functions, event sources, etc.
- Install AWS SAM CLI (on Linux)

```
sudo group add docker
sudo usermod –aG docker $USER
newgrp docker
sudo chmod 666 /var/run/docker.sock
wget https://github.com/aws/aws-sam-cli/releases/latest/download/aws-sam-cli-linux-x86_64.zip
unzip aws-sam-cli-linux-x86_64.zip -d sam-installation
sudo ./sam-installation/install
sam --version
```

https://docs.aws.amazon.com/serverless-application-model/latest/developerguide/serverless-sam-cli-install.html

AWS services

AWS Educate (and AWS console)

- https://aws.amazon.com/it/education/awseducate/
- https://console.aws.amazon.com/console/home?region=us-east-1

IAM (authentication)

- https://docs.aws.amazon.com/IAM/latest/UserGuide/iam-ug.pdf

SDK (software API)

- https://docs.aws.amazon.com/sdk-for-java/latest/developer-guide/home.html

Lambda (serverless computing and processing)

- https://docs.aws.amazon.com/lambda/latest/dg/getting-started.html
- https://console.aws.amazon.com/lambda/home?region=us-east-1#/functions

DynamoDB (key-value database)

- https://docs.aws.amazon.com/amazondynamodb/latest/developerguide/Introduction

S3 (object storage)

- https://s3.console.aws.amazon.com/s3/home?region=us-east-1

Case study

Given a dataset of sales per customer

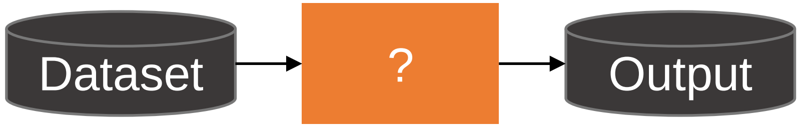
find the products frequently bought together

```
Dataset sample
%%%%%%%%%%%%%%

[ { customerName: Alice, products: [Pizza, Beer, Diaper] },
  { customerName: Bob, products: [Pizza, Beer, Diaper] },
  { customerName: Charlie, products: [Pizza, Cola] } ]
```

Case study

The pipeline involves a single transformation

- A classic mining problem, which one?



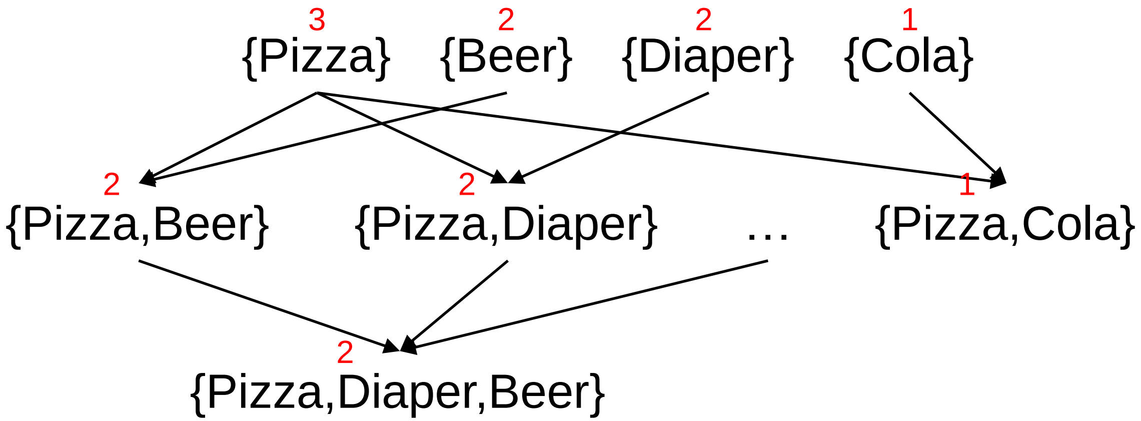
Frequent itemset mining

Find sets of items (i.e., itemsets) frequently appearing together

- **Item**: a product
- **Itemset**: a set of products
- **Frequently**: support above threshold
- **Support**: number of clients buying a set of products

Complexity: $O(2^{|items|})$

```
Dataset sample
%%%%%%%%%%%%


[[Pizza, Beer, Diaper],
 [Pizza, Beer, Diaper],
 [Pizza, Cola]]
```



Case study

$FIM\colon List\big[List[String]\big] \rightarrow List[Set[String]]$

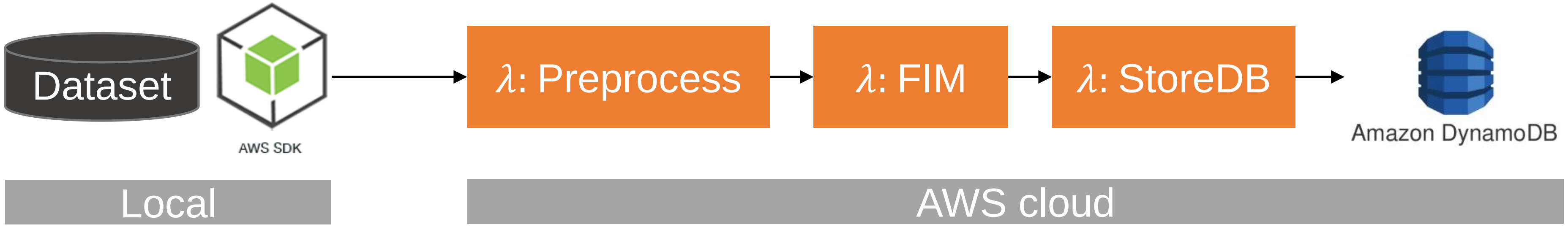
- FIM requires a list of lists as input, but we have nested JSON objects
- We need a pre-processing step

```
Raw dataset sample
%%%%%%%%%%%%%
[ { customerName: Alice, products: [Pizza, Beer, Diaper] },
  { customerName: Bob, products: [Pizza, Beer, Diaper] },
  { customerName: Charlie, products: [Pizza, Cola] } ]
```

```
Processed dataset sample
%%%%%%%%%%%%%
[[Pizza, Beer, Diaper],
 [Pizza, Beer, Diaper],
 [Pizza, Cola]]
```

Finally, we need to store the itemsets in the database

Reference pipeline



NOSQL storage: DynamoDB

Basic DynamoDB components: tables and items

Tables, collection of (data) items

Items, a group of attributes that is uniquely identifiable

- Each table contains zero or more items
  - No limit to the number of items you can store in a table
- Each item in the table has a unique identifier, or primary key
- E.g., in the table \`people\`, each item represents a \`person\`
  - The primary key consists of one attribute (\`fiscalCode\`)

NOSQL storage: DynamoDB

Attributes

- A data element that is not broken down any further
  - E.g., an item in the `people` table contains attributes `fiscalCode` and `lastName`
- Most of the attributes are scalar (have only one value)
- Some of the items have a nested attribute (`address`) up to 32 levels deep

Schemaless

- Other than the primary key, a table is schemaless
  - Neither the attributes nor their data types need to be defined beforehand
  - Each item can have its own distinct attributes

NOSQL storage: DynamoDB

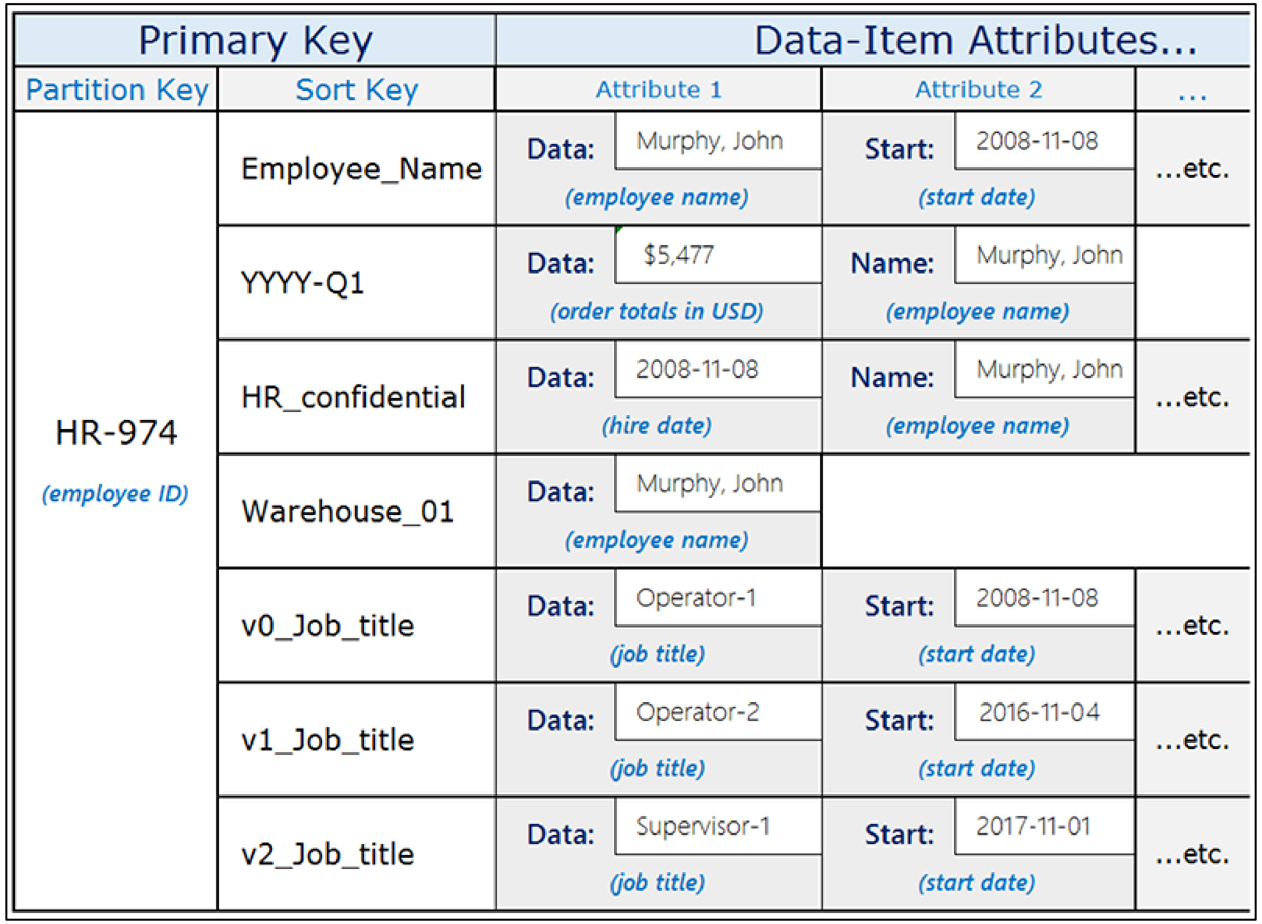
Primary Key

- To create a table, you must specify the primary key of the table
- No two items can have the same key

Two types of primary keys

- Partition key: a simple primary key composed of one attribute (partition key)
  - Keys are inputs to an internal hash function
  - The hash function determines the physical partition in which the item will be stored
  - E.g., access any item in the \`people\` table directly by providing the \`fiscalCode\`
- Composite primary key: partition key and sort key (two attributes)
  - First attribute is the partition key
  - Second attribute is the sort key
  - Items in same partition key value are stored together and sorted by sort key

NOSQL storage: DynamoDB



Primary Key		Data-Item Attributes...		
Partition Key	Sort Key	Attribute 1	Attribute 2	...
HR-974 *(employee ID)*	Employee_Name	Data: Murphy, John *(employee name)*	Start: 2008-11-08 *(start date)*	...etc.
	YYYY-Q1	Data: $5,477 *(order totals in USD)*	Name: Murphy, John *(employee name)*	
	HR_confidential	Data: 2008-11-08 *(hire date)*	Name: Murphy, John *(employee name)*	...etc.
	Warehouse_01	Data: Murphy, John *(employee name)*		
	v0_Job_title	Data: Operator-1 *(job title)*	Start: 2008-11-08 *(start date)*	...etc.
	v1_Job_title	Data: Operator-2 *(job title)*	Start: 2016-11-04 *(start date)*	...etc.
	v2_Job_title	Data: Supervisor-1 *(job title)*	Start: 2017-11-01 *(start date)*	...etc.

https://docs.aws.amazon.com/amazondynamodb/latest/developerguide/bp-gsi-overloading.html

NOSQL storage: DynamoDB

Create a table `frequent-sales` with a composite key

- `dataset`: String
- `timestamp`: String

```
$ aws dynamodb create-table \
    --table-name frequent-sales \
    --attribute-definitions AttributeName=dataset,AttributeType=S AttributeName=timestamp,AttributeType=S \
    --key-schema AttributeName=dataset,KeyType=HASH AttributeName=timestamp,KeyType=RANGE \
    --provisioned-throughput ReadCapacityUnits=1,WriteCapacityUnits=1

$ aws dynamodb list-tables

$ aws dynamodb delete-table --table-name frequent-sales
```

NOSQL storage: DynamoDB

Reading data from DynamoDB might not reflect the results of a recent write

Eventually Consistent Reads (default)

- Response might include stale data
- After short time, the response should return the latest data

Strongly Consistent Reads

- Response includes the most up-to-date data
- A strongly consistent read might not be available if there is a network delay or outage
  - In this case, DynamoDB may return a server error (HTTP 500)
- Strongly consistent reads may have higher latency than eventually consistent reads
- Strongly consistent reads are not supported on global secondary indexes

NOSQL storage: DynamoDB

Provisioned mode: specify the #reads and #writes per second

- You have predictable application traffic or traffic ramps gradually
- You can forecast capacity requirements to control costs

One read capacity unit

- One strongly consistent read per second, two eventually consistent reads per second
- RCUs also depend on the item size (a read is up to 4 KB in size), if item size is 8 KB
  - 2 RCUs to sustain one strongly consistent read per second
  - 1 RCU if you choose eventually consistent reads

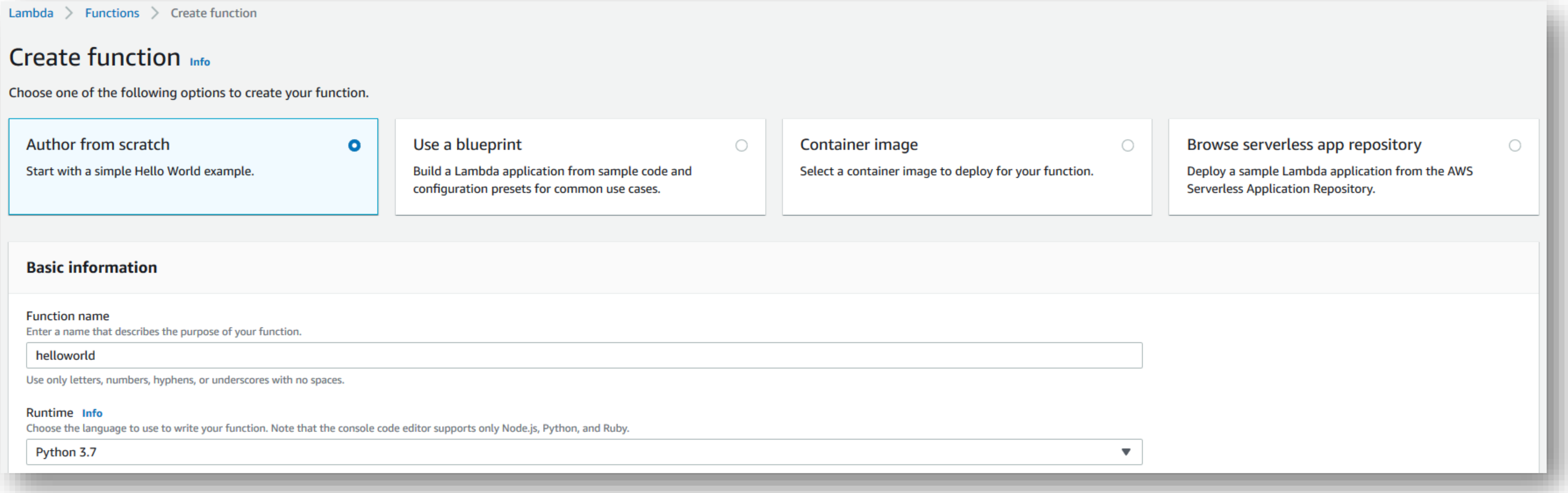
One write capacity unit represents one write per second for an item up to 1 KB in size

NOSQL storage: DynamoDB

Put a new item and get it back

```
$ aws dynamodb put-item
    --table-name frequent-sales
    --item '{"dataset": {"S": "sales"}, "timestamp": {"S": "1611226870"}, "bar": {"S": "foobar"}}'


$ aws dynamodb query
    --table-name frequent-sales
    --key-condition-expression "dataset = :n"
    --expression-attribute-values '{":n":{"S":"sales"}}'
```

Lambda: create a function



Create function Info

Choose one of the following options to create your function.

- ● Author from scratch
  Start with a simple Hello World example.
- ○ Use a blueprint
  Build a Lambda application from sample code and configuration presets for common use cases.
- ○ Container image
  Select a container image to deploy for your function.
- ○ Browse serverless app repository
  Deploy a sample Lambda application from the AWS Serverless Application Repository.

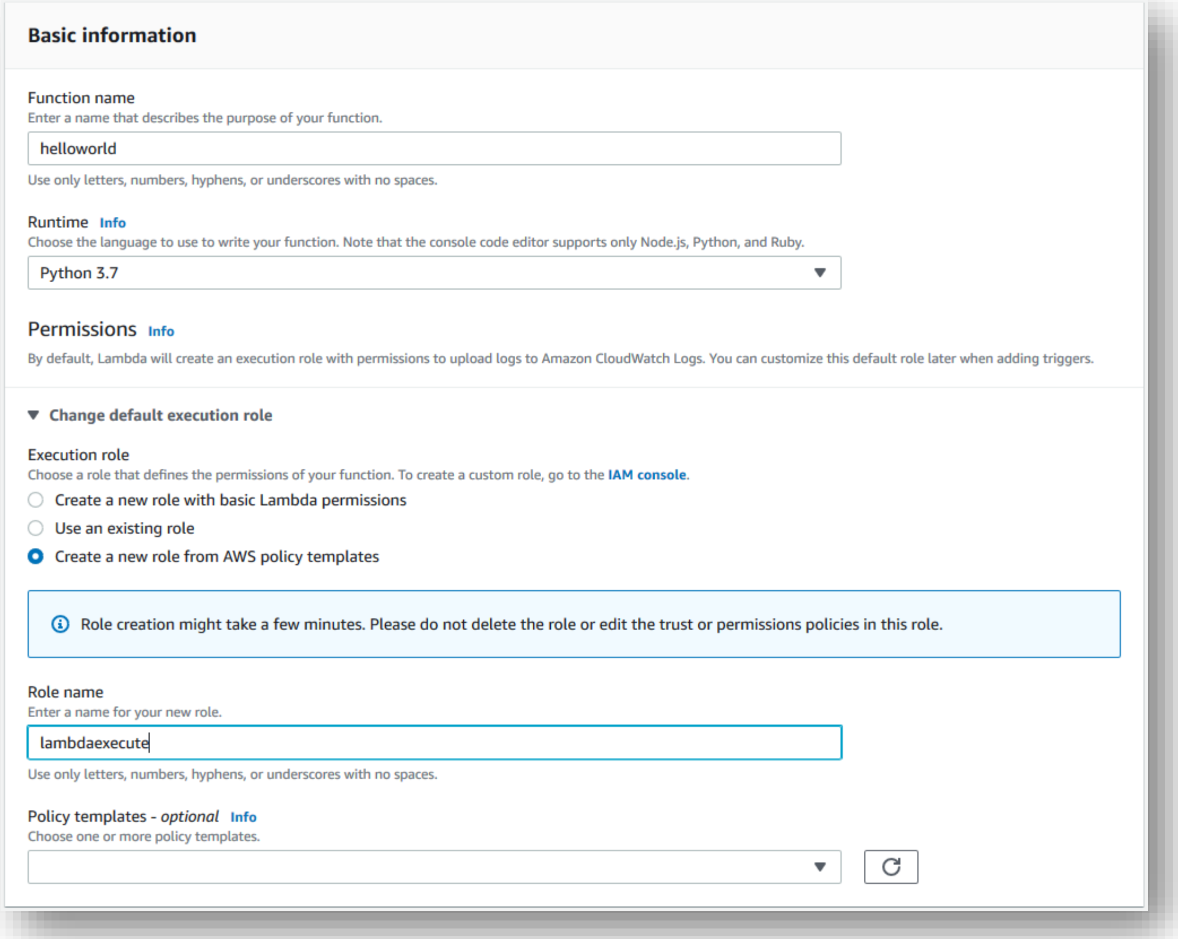
Basic information

Function name
Enter a name that describes the purpose of your function.
helloworld
Use only letters, numbers, hyphens, or underscores with no spaces.

Runtime Info
Choose the language to use to write your function. Note that the console code editor supports only Node.js, Python, and Ruby.
Python 3.7 ▼

https://console.aws.amazon.com/lambda/home?region=us-east-1#/functions

Lambda: attaching a role



Basic information

Function name
Enter a name that describes the purpose of your function.
helloworld
Use only letters, numbers, hyphens, or underscores with no spaces.

Runtime Info
Choose the language to use to write your function. Note that the console code editor supports only Node.js, Python, and Ruby.
Python 3.7 ▼

Permissions Info
By default, Lambda will create an execution role with permissions to upload logs to Amazon CloudWatch Logs. You can customize this default role later when adding triggers.

▼ Change default execution role

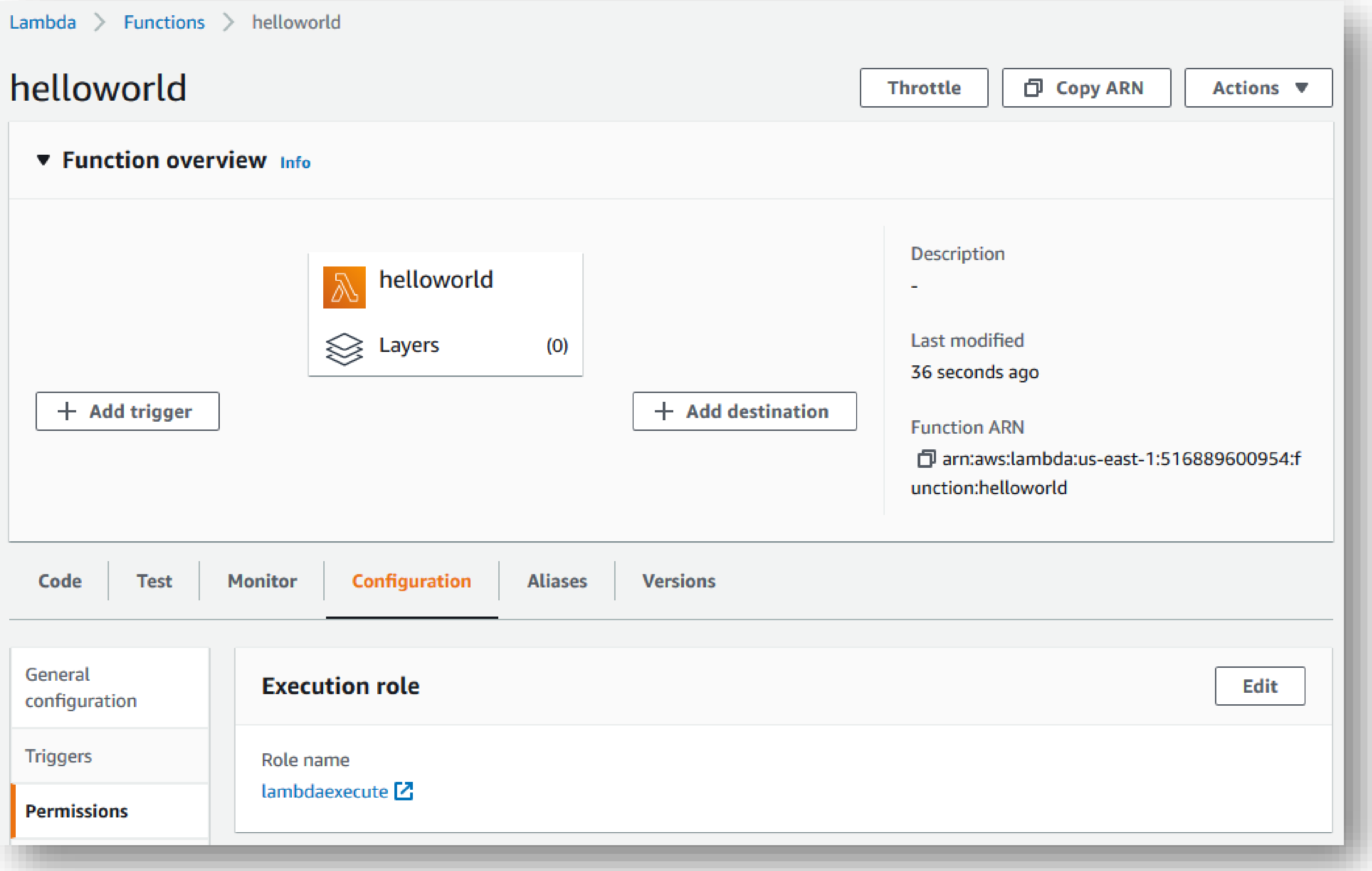
Execution role
Choose a role that defines the permissions of your function. To create a custom role, go to the IAM console.
○ Create a new role with basic Lambda permissions
○ Use an existing role
● Create a new role from AWS policy templates

ⓘ Role creation might take a few minutes. Please do not delete the role or edit the trust or permissions policies in this role.

Role name
Enter a name for your new role.
lambdaexecute
Use only letters, numbers, hyphens, or underscores with no spaces.

Policy templates - *optional* Info
Choose one or more policy templates.
▼

Lambda: attaching a role



Lambda > Functions > helloworld

helloworld

Throttle | Copy ARN | Actions ▼

▼ Function overview Info

helloworld

Layers (0)

+ Add trigger

+ Add destination

Description

-

Last modified

36 seconds ago

Function ARN

arn:aws:lambda:us-east-1:516889600954:function:helloworld

Code | Test | Monitor | Configuration | Aliases | Versions

General configuration

Triggers

Permissions

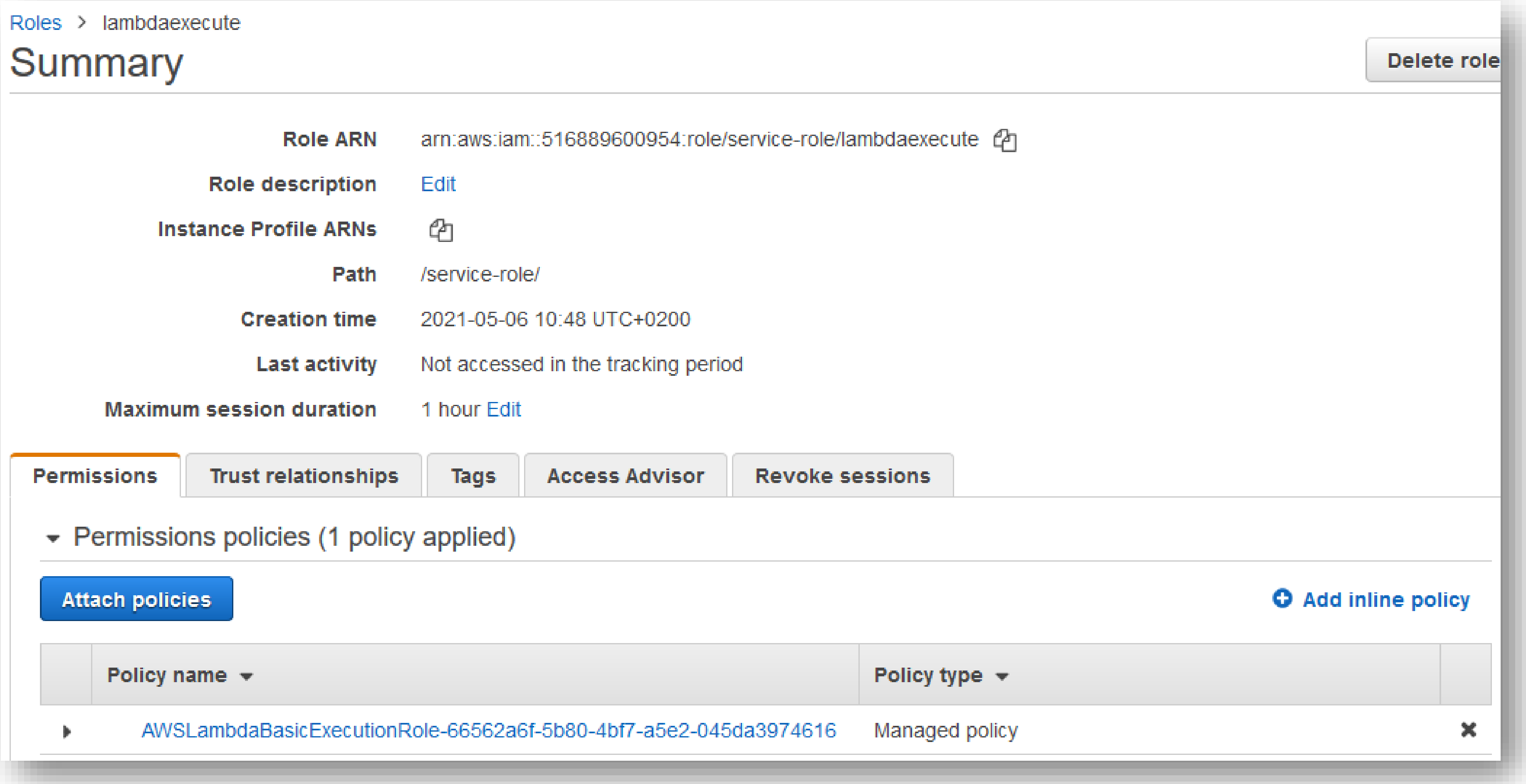
Execution role

Edit

Role name

lambdaexecute

Lambda: attaching a role



Roles > lambdaexecute

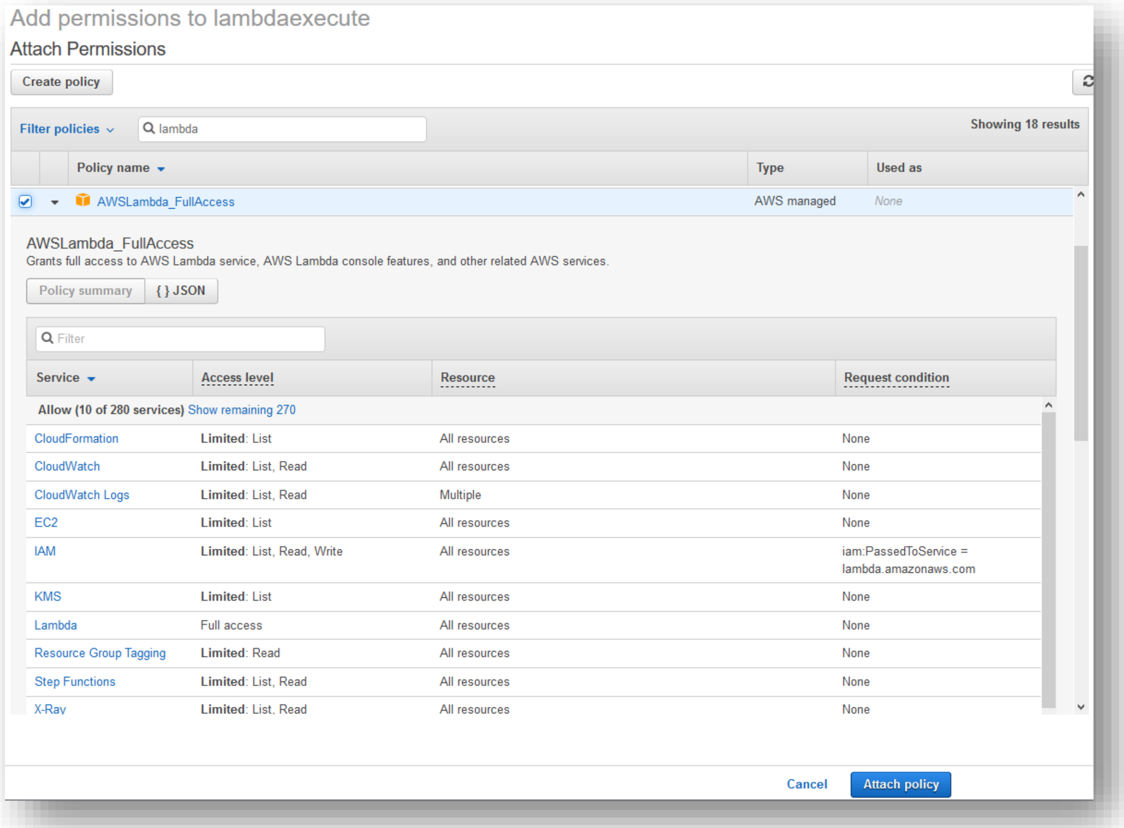
Summary

Role ARN	arn:aws:iam::516889600954:role/service-role/lambdaexecute
Role description	Edit
Instance Profile ARNs	
Path	/service-role/
Creation time	2021-05-06 10:48 UTC+0200
Last activity	Not accessed in the tracking period
Maximum session duration	1 hour Edit

Permissions policies (1 policy applied)

Policy name	Policy type
AWSLambdaBasicExecutionRole-66562a6f-5b80-4bf7-a5e2-045da3974616	Managed policy

Lambda: attaching a role



Add permissions to lambdaexecute

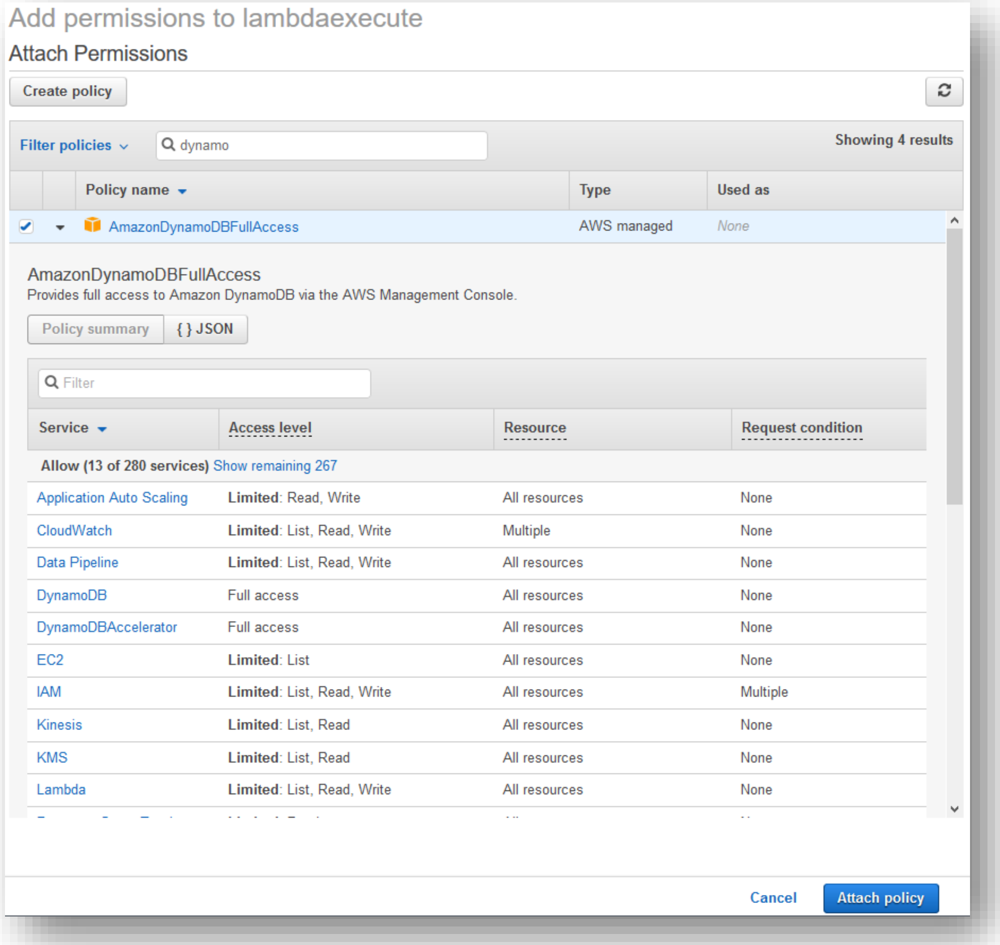
Attach Permissions

		Policy name	Type	Used as
☑	▾	AWSLambda_FullAccess	AWS managed	None

AWSLambda_FullAccess

Grants full access to AWS Lambda service, AWS Lambda console features, and other related AWS services.

Service	Access level	Resource	Request condition
Allow (10 of 280 services) Show remaining 270			
CloudFormation	**Limited**: List	All resources	None
CloudWatch	**Limited**: List, Read	All resources	None
CloudWatch Logs	**Limited**: List, Read	Multiple	None
EC2	**Limited**: List	All resources	None
IAM	**Limited**: List, Read, Write	All resources	iam:PassedToService = lambda.amazonaws.com
KMS	**Limited**: List	All resources	None
Lambda	Full access	All resources	None
Resource Group Tagging	**Limited**: Read	All resources	None
Step Functions	**Limited**: List, Read	All resources	None
X-Ray	**Limited**: List, Read	All resources	None



Add permissions to lambdaexecute

Attach Permissions

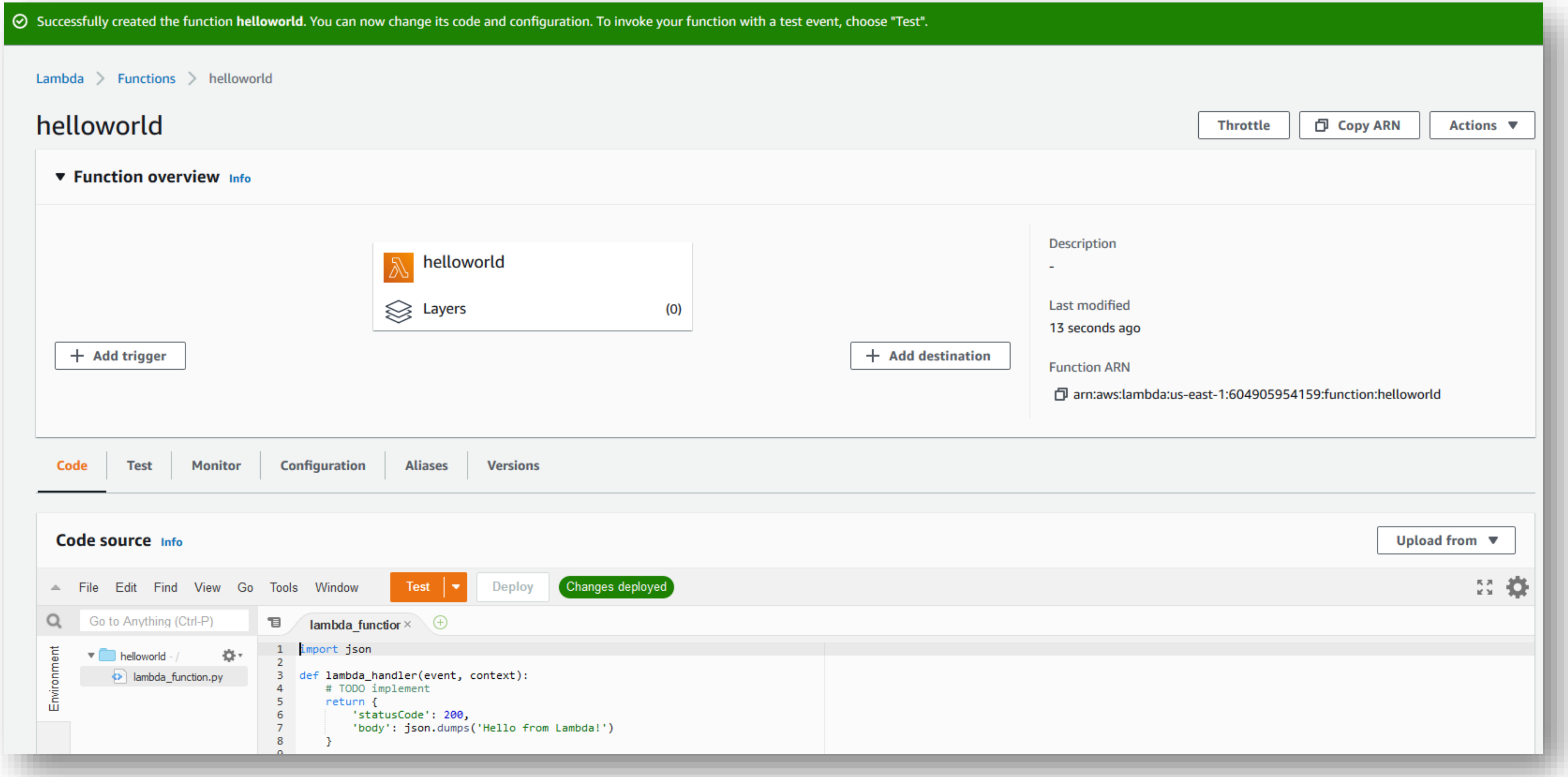
		Policy name	Type	Used as
☑	▾	AmazonDynamoDBFullAccess	AWS managed	None

AmazonDynamoDBFullAccess

Provides full access to Amazon DynamoDB via the AWS Management Console.

Service	Access level	Resource	Request condition
Allow (13 of 280 services) Show remaining 267			
Application Auto Scaling	**Limited**: Read, Write	All resources	None
CloudWatch	**Limited**: List, Read, Write	Multiple	None
Data Pipeline	**Limited**: List, Read, Write	All resources	None
DynamoDB	Full access	All resources	None
DynamoDBAccelerator	Full access	All resources	None
EC2	**Limited**: List	All resources	None
IAM	**Limited**: List, Read, Write	All resources	Multiple
Kinesis	**Limited**: List, Read	All resources	None
KMS	**Limited**: List, Read	All resources	None
Lambda	**Limited**: List, Read, Write	All resources	None

Lambda: create a function



Successfully created the function **helloworld**. You can now change its code and configuration. To invoke your function with a test event, choose "Test".

Lambda > Functions > helloworld

helloworld

Description
-

Last modified
13 seconds ago

Function ARN
arn:aws:lambda:us-east-1:604905954159:function:helloworld

```
import json

def lambda_handler(event, context):
    # TODO implement
    return {
        'statusCode': 200,
        'body': json.dumps('Hello from Lambda!')
    }
```

https://console.aws.amazon.com/lambda/home?region=us-east-1#/functions

Lambda: create a function

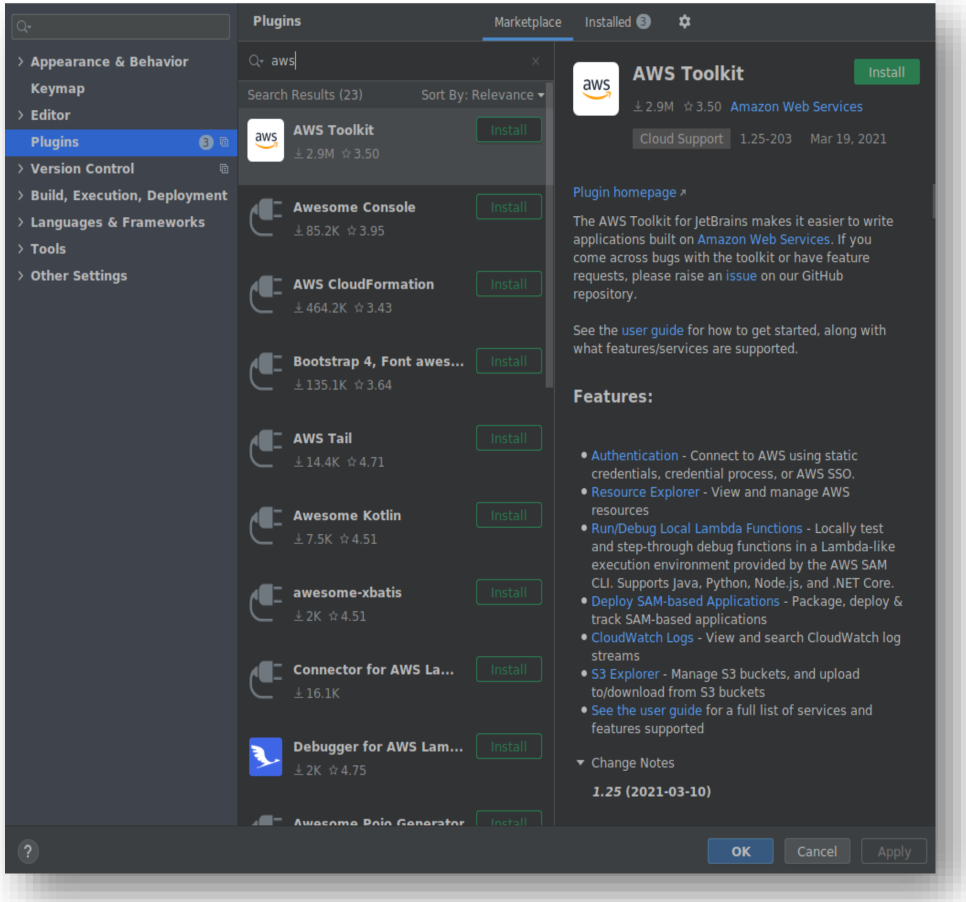
Manually creating the functions is cumbersome

- We must copy and paste code
- No automatic testing
- No debugging
- No IDE support (and not all languages are supported)

Switch to IntelliJ IDEA + AWS Toolkit

AWS Toolkit

- Get the latest IntelliJ IDEA
- Install the `AWS Toolkit`
- Copy the credentials
  ```
  cp ~/.aws/credentials ~/.aws/config
  ```
- Clone the repo
  ```
  git clone https://github.com/w4bo/bigdata-aws/
  ```
- Import `lab01-lambda` as a Gradle project
- Verify that the project builds
  ```
  ./gradlew
  ```



Search Results (23) Sort By: Relevance

AWS Toolkit ⤓2.9M ☆3.50
Awesome Console ⤓85.2K ☆3.95
AWS CloudFormation ⤓464.2K ☆3.43
Bootstrap 4, Font awes... ⤓135.1K ☆3.64
AWS Tail ⤓14.4K ☆4.71
Awesome Kotlin ⤓7.5K ☆4.51
awesome-xbatis ⤓2K ☆4.51
Connector for AWS La... ⤓16.1K
Debugger for AWS Lam... ⤓2K ☆4.75
Awesome Pojo Generator

AWS Toolkit

⤓2.9M ☆3.50 Amazon Web Services

Cloud Support 1.25-203 Mar 19, 2021

Plugin homepage ↗

The AWS Toolkit for JetBrains makes it easier to write applications built on Amazon Web Services. If you come across bugs with the toolkit or have feature requests, please raise an issue on our GitHub repository.

See the user guide for how to get started, along with what features/services are supported.

Features:

- Authentication - Connect to AWS using static credentials, credential process, or AWS SSO.
- Resource Explorer - View and manage AWS resources
- Run/Debug Local Lambda Functions - Locally test and step-through debug functions in a Lambda-like execution environment provided by the AWS SAM CLI. Supports Java, Python, Node.js, and .NET Core.
- Deploy SAM-based Applications - Package, deploy & track SAM-based applications
- CloudWatch Logs - View and search CloudWatch log streams
- S3 Explorer - Manage S3 buckets, and upload to/download from S3 buckets
- See the user guide for a full list of services and features supported

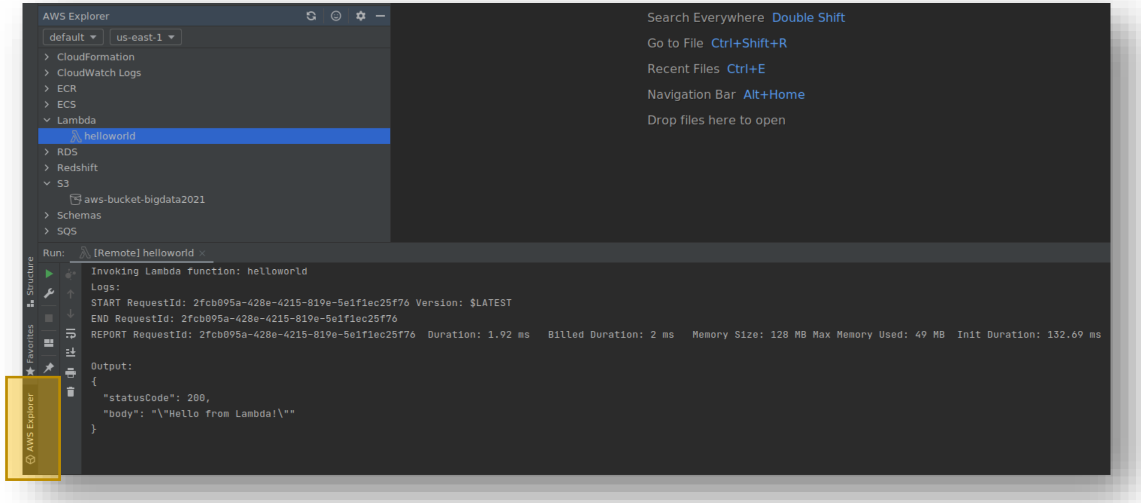
▾ Change Notes

***1.25* (2021-03-10)**

AWS Toolkit

Click on `AWS Explorer`

- You can see the `helloworld` function
- Plus `CloudWatch Logs` and `S3`



```
Invoking Lambda function: helloworld
Logs:
START RequestId: 2fcb095a-428e-4215-819e-5e1f1ec25f76 Version: $LATEST
END RequestId: 2fcb095a-428e-4215-819e-5e1f1ec25f76
REPORT RequestId: 2fcb095a-428e-4215-819e-5e1f1ec25f76  Duration: 1.92 ms   Billed Duration: 2 ms   Memory Size: 128 MB Max Memory Used: 49 MB  Init Duration: 132.69 ms

Output:
{
  "statusCode": 200,
  "body": "\"Hello from Lambda!\""
}
```

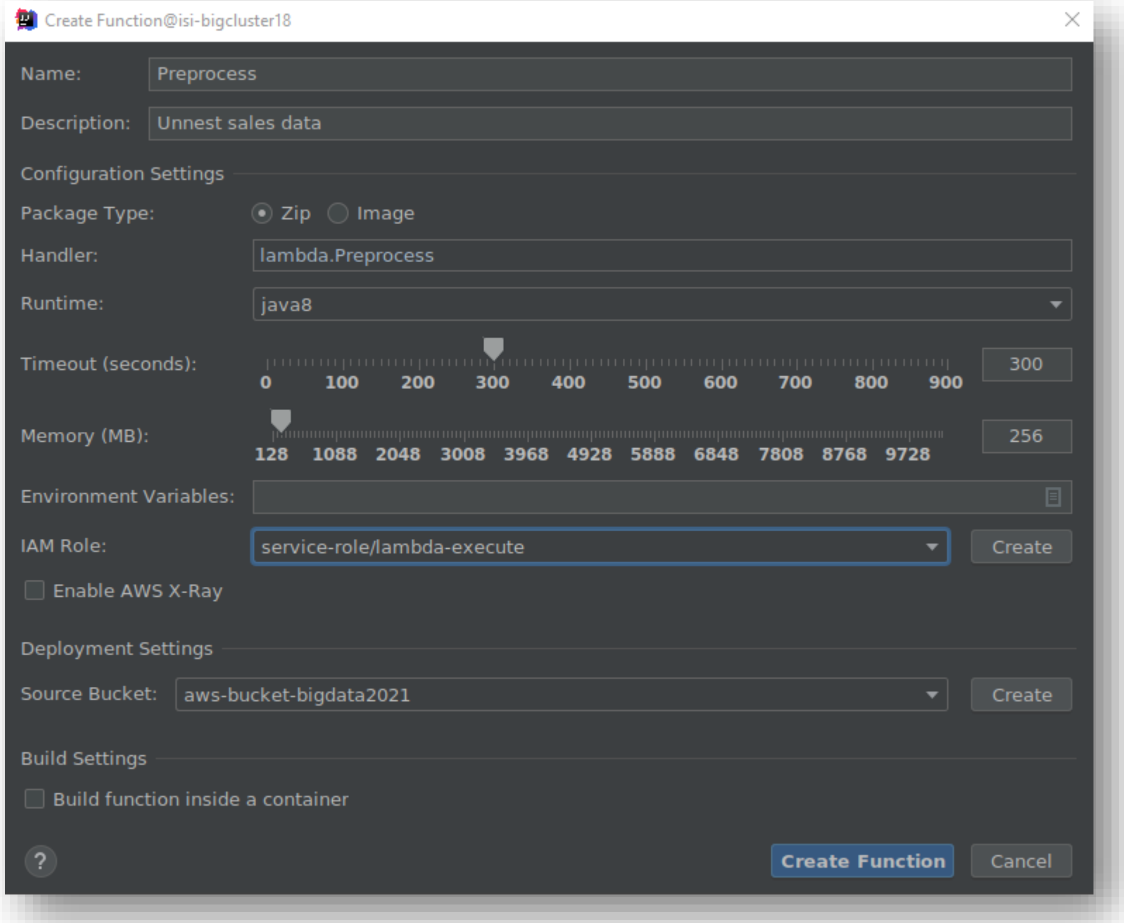
AWS Toolkit

Test the existing code locally

- With Gradle
- Or with local Lambda execution

Deploy a new Lambda function from the existing code

- Right click on AWS Explorer > Lambda
- Select `Create new AWS Lambda…`
- Populate the settings
- `Create the function`



Create Function@isi-bigcluster18

Name: Preprocess

Description: Unnest sales data

Configuration Settings

Package Type: ● Zip ○ Image

Handler: lambda.Preprocess

Runtime: java8

Timeout (seconds): 300

Memory (MB): 256

Environment Variables:

IAM Role: service-role/lambda-execute

□ Enable AWS X-Ray

Deployment Settings

Source Bucket: aws-bucket-bigdata2021

Build Settings

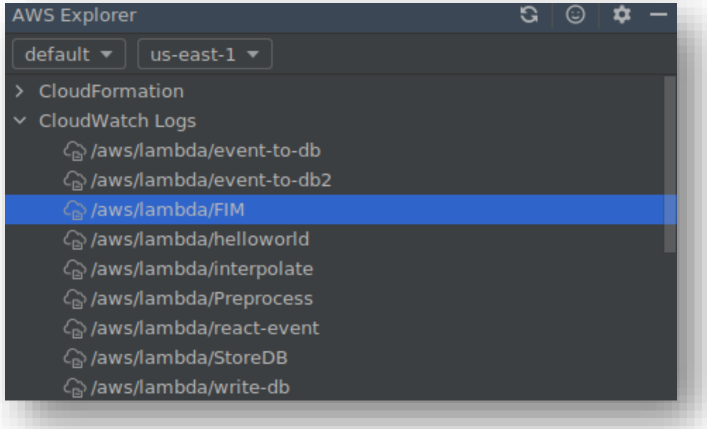
□ Build function inside a container

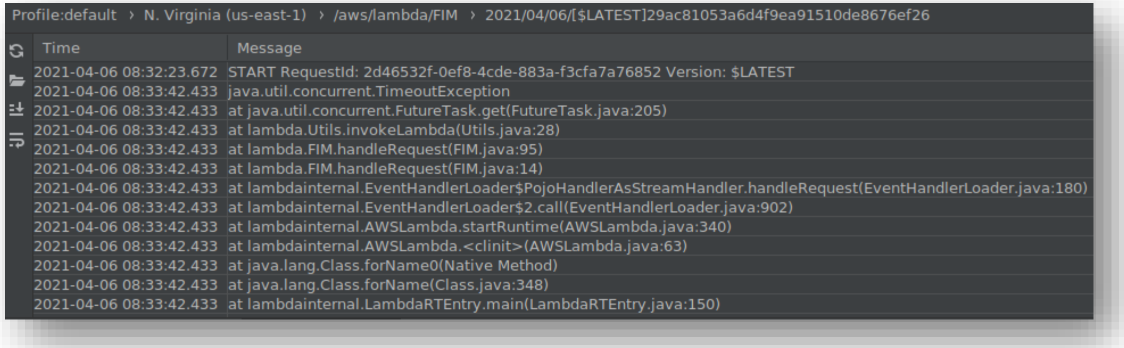
https://aws.amazon.com/lambda/pricing/

AWS Toolkit

Check the log for errors and pricing

- AWS Toolkit > CloudWatch Logs
- Double click on the function name
- Double click on the log entry





Profile:default › N. Virginia (us-east-1) › /aws/lambda/FIM › 2021/04/06/[$LATEST]29ac81053a6d4f9ea91510de8676ef26

```
Time                     Message
2021-04-06 08:32:23.672  START RequestId: 2d46532f-0ef8-4cde-883a-f3cfa7a76852 Version: $LATEST
2021-04-06 08:33:42.433  java.util.concurrent.TimeoutException
2021-04-06 08:33:42.433  at java.util.concurrent.FutureTask.get(FutureTask.java:205)
2021-04-06 08:33:42.433  at lambda.Utils.invokeLambda(Utils.java:28)
2021-04-06 08:33:42.433  at lambda.FIM.handleRequest(FIM.java:95)
2021-04-06 08:33:42.433  at lambda.FIM.handleRequest(FIM.java:14)
2021-04-06 08:33:42.433  at lambdainternal.EventHandlerLoader$PojoHandlerAsStreamHandler.handleRequest(EventHandlerLoader.java:180)
2021-04-06 08:33:42.433  at lambdainternal.EventHandlerLoader$2.call(EventHandlerLoader.java:902)
2021-04-06 08:33:42.433  at lambdainternal.AWSLambda.startRuntime(AWSLambda.java:340)
2021-04-06 08:33:42.433  at lambdainternal.AWSLambda.<clinit>(AWSLambda.java:63)
2021-04-06 08:33:42.433  at java.lang.Class.forName0(Native Method)
2021-04-06 08:33:42.433  at java.lang.Class.forName(Class.java:348)
2021-04-06 08:33:42.433  at lambdainternal.LambdaRTEntry.main(LambdaRTEntry.java:150)
```

Data pipeline

Deploy and execute the HelloWorld.java lambda function

Given the created storage: S3 and DynamoDB

- Deploy the function \`FIM\`
- Deploy the function \`Preprocess\`
- Run ReadDataset.java
- Check that the table \`frequent-sales\` has the FIs for the dataset \`sales\`

Some hints

- Function names are case sensitive
- Some function need more than 128MB of memory
  - Behold! The higher the RAM, the higher the costs

BIG DATA

Amazon EMR

EC2

AWS uses public-key cryptography to secure the login

You can create one using the Amazon EC2 console

- Open the Amazon EC2 console at https://console.aws.amazon.com/ec2/
- In the navigation pane, choose `Key Pairs`
- Choose `Create key pair`
- For `Name`, enter a descriptive name for the key pair
- For `File format`, choose the format in which to save the private key
  - OpenSSH, choose `pem` (`chmod 400 *my-key-pair*.pem`)
  - PuTTY, choose `ppk`
- Choose `Create key pair`
- The private key file is automatically downloaded by your browser

Creating the cluster

Choose the frameworks and applications to install

Data process

- Submit jobs or queries directly to installed applications
- Run steps in the cluster

Submitting jobs

- Connect to the master node over a secure connection
- Access the interfaces and tools that are available on your cluster

Creating the cluster

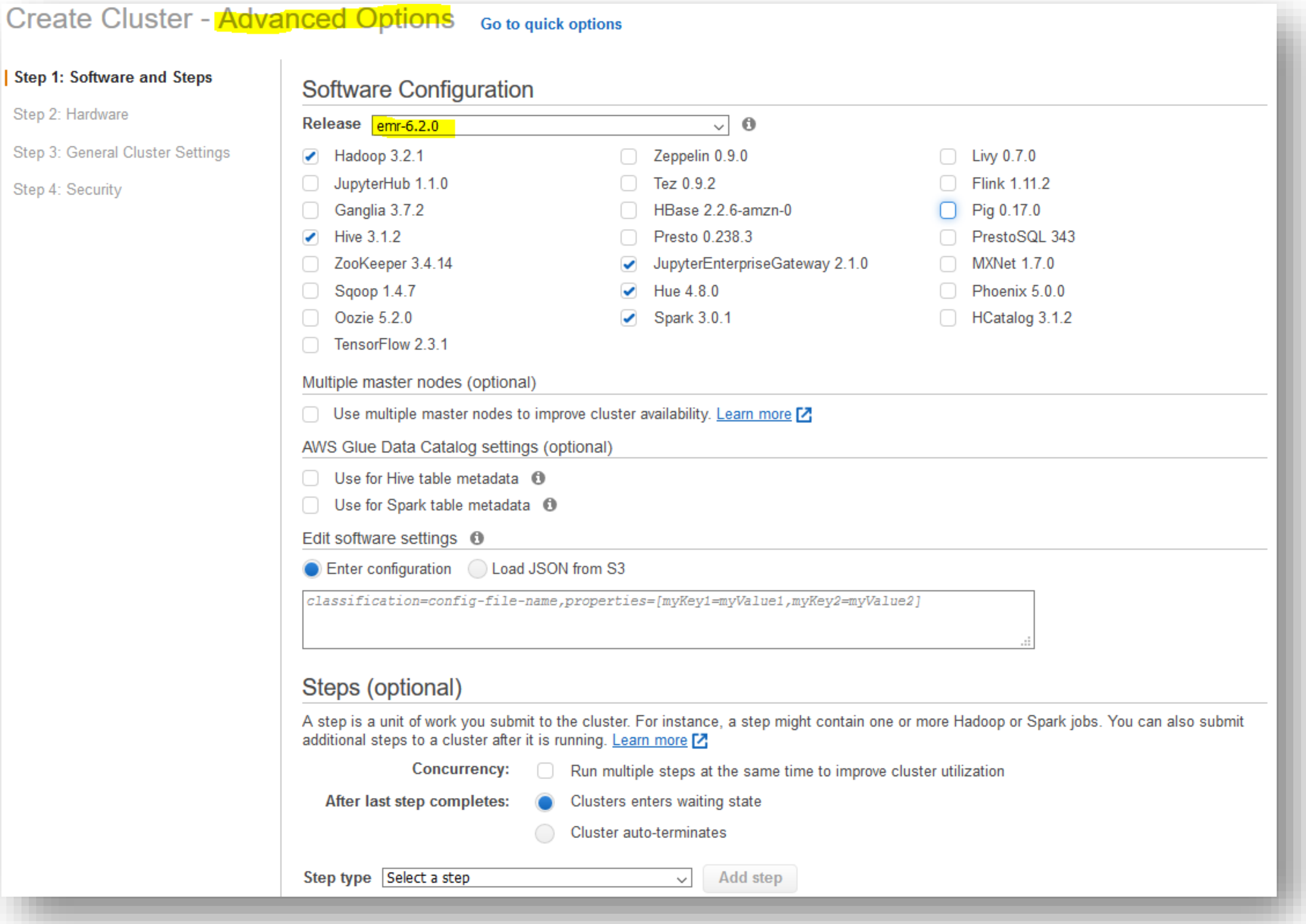
Using CLI (command line interface)

```
aws emr create-cluster \
    --name "My First EMR Cluster" \
    --release-label emr-5.32.0 \
    --applications Name=Spark \
    --ec2-attributes KeyName=myEMRKeyPairName \
    --instance-type m5.xlarge \
    --instance-count 3 \
    --use-default-roles
```

This is more pragmatic, but there are many options to explore

- Let’s stick to AWS Console
- https://console.aws.amazon.com/elasticmapreduce/

Creating the cluster



Create Cluster - Advanced Options   Go to quick options

Step 1: Software and Steps

Step 2: Hardware

Step 3: General Cluster Settings

Step 4: Security

Software Configuration

Release emr-6.2.0

- ☑ Hadoop 3.2.1
- □ JupyterHub 1.1.0
- □ Ganglia 3.7.2
- ☑ Hive 3.1.2
- □ ZooKeeper 3.4.14
- □ Sqoop 1.4.7
- □ Oozie 5.2.0
- □ TensorFlow 2.3.1
- □ Zeppelin 0.9.0
- □ Tez 0.9.2
- □ HBase 2.2.6-amzn-0
- □ Presto 0.238.3
- ☑ JupyterEnterpriseGateway 2.1.0
- ☑ Hue 4.8.0
- ☑ Spark 3.0.1
- □ Livy 0.7.0
- □ Flink 1.11.2
- □ Pig 0.17.0
- □ PrestoSQL 343
- □ MXNet 1.7.0
- □ Phoenix 5.0.0
- □ HCatalog 3.1.2

Multiple master nodes (optional)

□ Use multiple master nodes to improve cluster availability. Learn more

AWS Glue Data Catalog settings (optional)

□ Use for Hive table metadata

□ Use for Spark table metadata

Edit software settings

● Enter configuration ○ Load JSON from S3

```
classification=config-file-name,properties=[myKey1=myValue1,myKey2=myValue2]
```

Steps (optional)

A step is a unit of work you submit to the cluster. For instance, a step might contain one or more Hadoop or Spark jobs. You can also submit additional steps to a cluster after it is running. Learn more

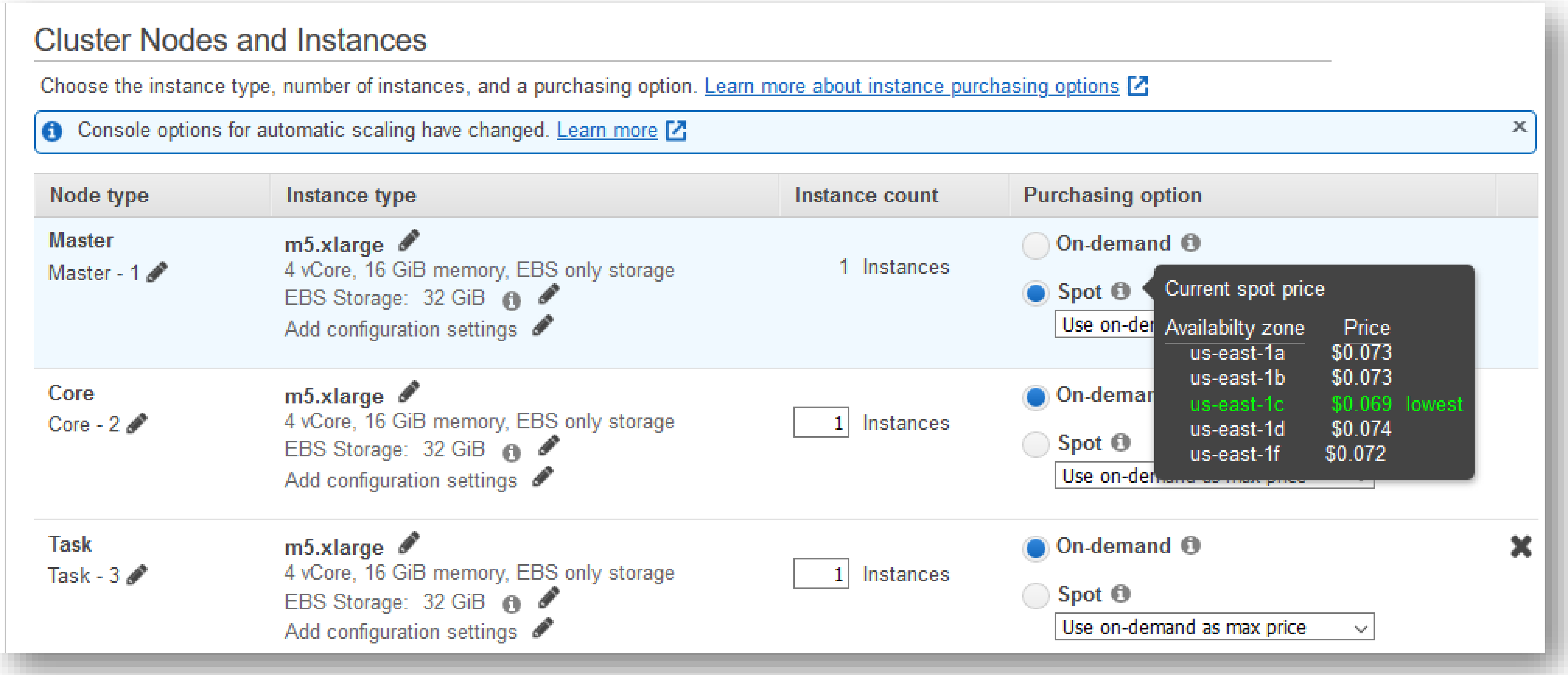
Concurrency: □ Run multiple steps at the same time to improve cluster utilization

After last step completes: ● Clusters enters waiting state

○ Cluster auto-terminates

Step type Select a step   Add step

Creating the cluster



Cluster Nodes and Instances

Choose the instance type, number of instances, and a purchasing option. Learn more about instance purchasing options

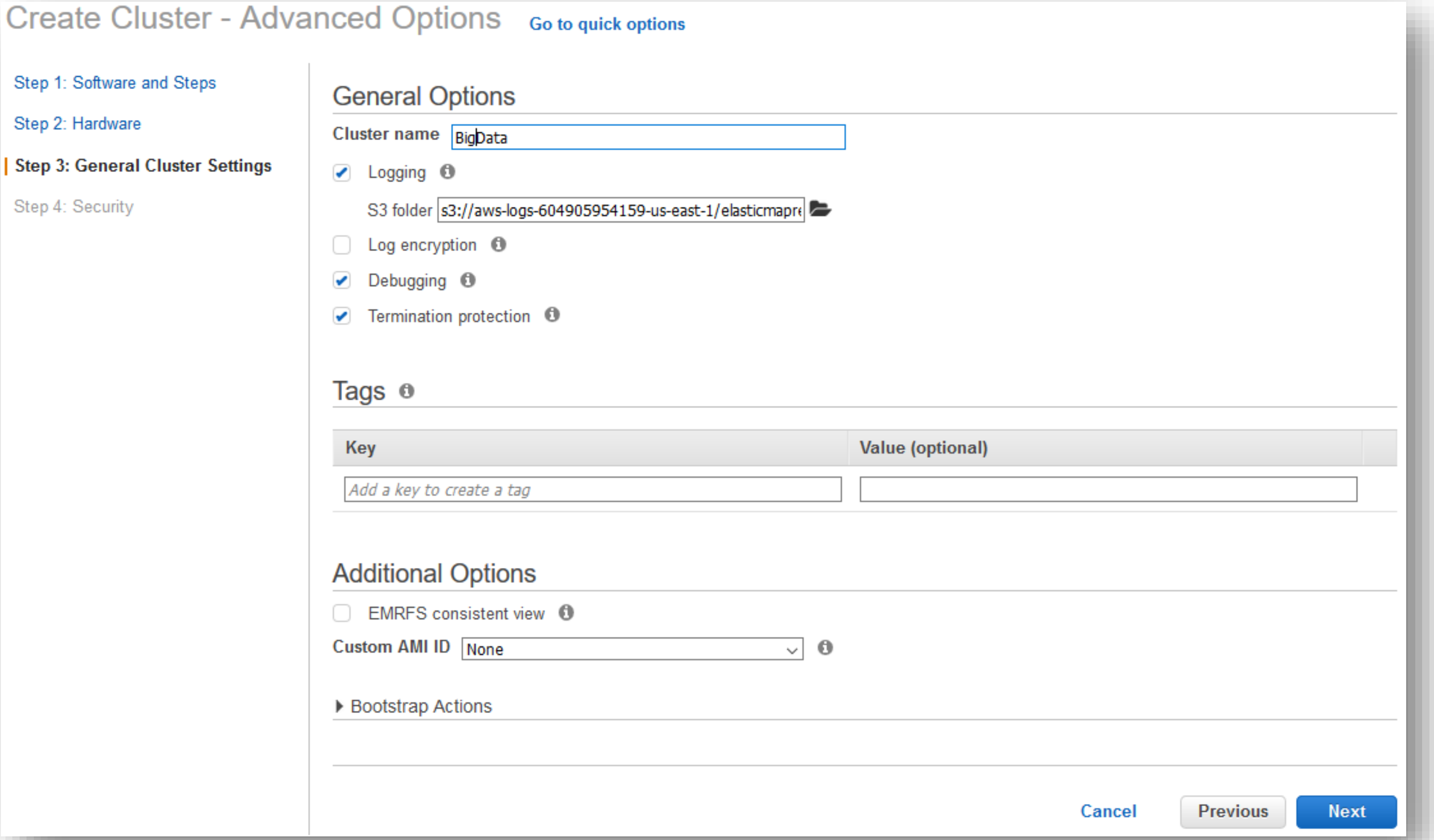
Console options for automatic scaling have changed. Learn more

Node type	Instance type	Instance count	Purchasing option
Master Master - 1	**m5.xlarge** 4 vCore, 16 GiB memory, EBS only storage EBS Storage: 32 GiB Add configuration settings	1 Instances	○ On-demand ● Spot Use on-de
Core Core - 2	**m5.xlarge** 4 vCore, 16 GiB memory, EBS only storage EBS Storage: 32 GiB Add configuration settings	1 Instances	● On-demar ○ Spot Use on-de
Task Task - 3	**m5.xlarge** 4 vCore, 16 GiB memory, EBS only storage EBS Storage: 32 GiB Add configuration settings	1 Instances	● On-demand ○ Spot Use on-demand as max price

Current spot price

Availabilty zone	Price	
us-east-1a	$0.073	
us-east-1b	$0.073	
us-east-1c	$0.069	lowest
us-east-1d	$0.074	
us-east-1f	$0.072	

Creating the cluster



Create Cluster - Advanced Options

Step 3: General Cluster Settings

General Options

Cluster name: BigData

☑ Logging

S3 folder: s3://aws-logs-604905954159-us-east-1/elasticmapre

□ Log encryption

☑ Debugging

☑ Termination protection

Tags

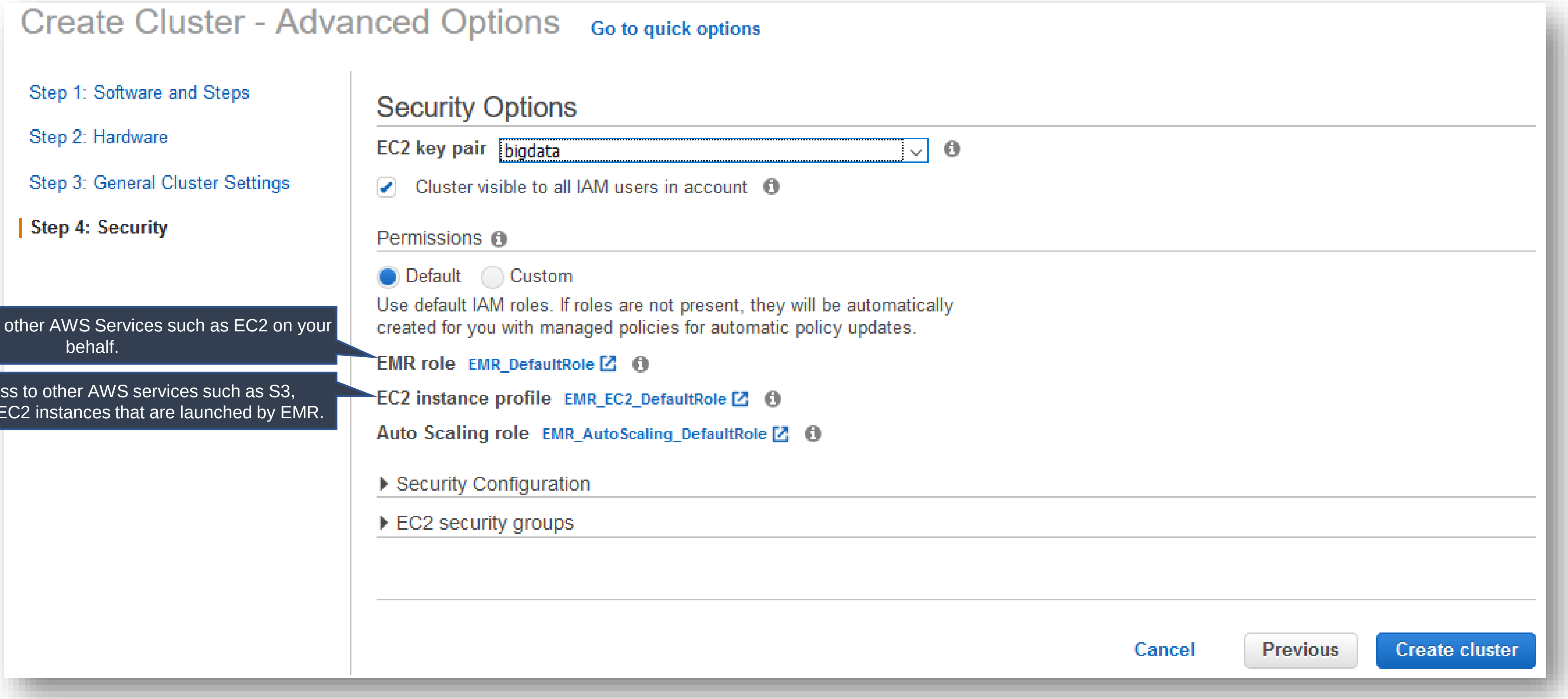
Key	Value (optional)
Add a key to create a tag	

Additional Options

□ EMRFS consistent view

Custom AMI ID: None

Creating the cluster



Create Cluster - Advanced Options   Go to quick options

Step 1: Software and Steps

Step 2: Hardware

Step 3: General Cluster Settings

Step 4: Security

Security Options

EC2 key pair: bigdata

☑ Cluster visible to all IAM users in account

Permissions

● Default ○ Custom

Use default IAM roles. If roles are not present, they will be automatically created for you with managed policies for automatic policy updates.

EMR role: EMR_DefaultRole

EC2 instance profile: EMR_EC2_DefaultRole

Auto Scaling role: EMR_AutoScaling_DefaultRole

Allows EMR to call other AWS Services such as EC2 on your behalf.

Provides access to other AWS services such as S3, DynamoDB from EC2 instances that are launched by EMR.

Creating the cluster

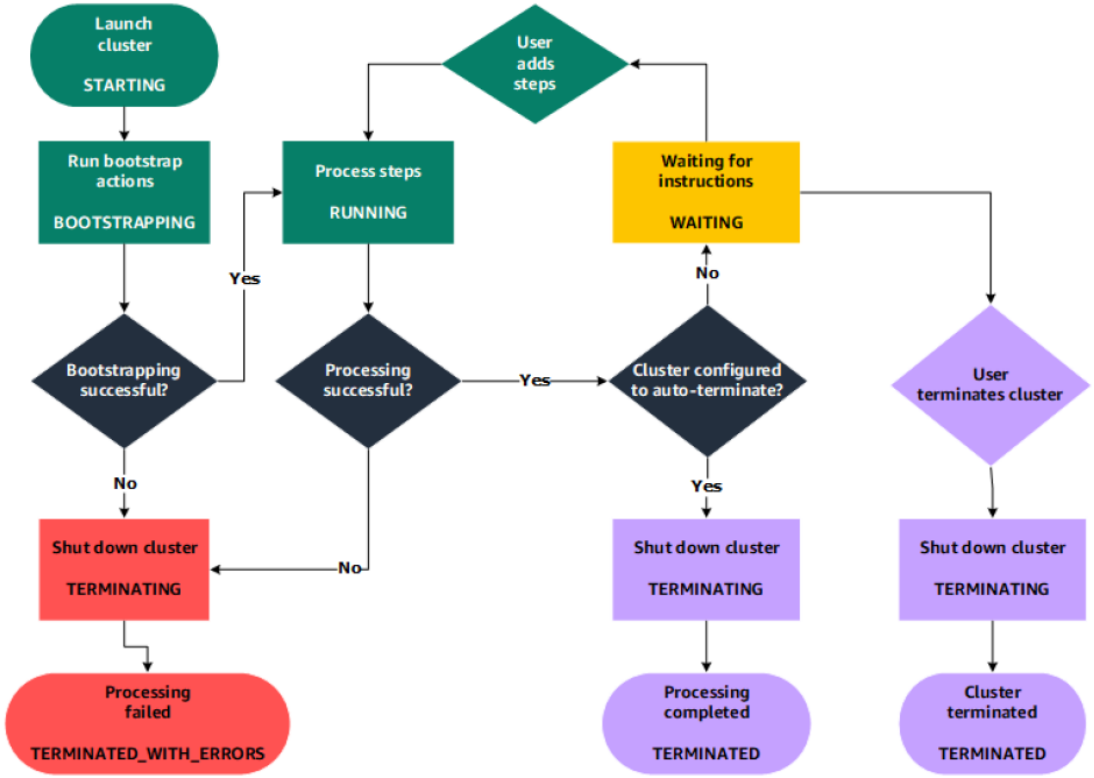
Using CLI (command line interface)

```
aws emr create-cluster --auto-scaling-role EMR_AutoScaling_DefaultRole --termination-protected --
applications Name=Hadoop Name=Hive Name=Hue Name=JupyterEnterpriseGateway Name=Spark --ebs-root-volume-
size 10 --ec2-attributes
'{"KeyName":"bigdata","InstanceProfile":"EMR_EC2_DefaultRole","SubnetId":"subnet-
5fa2f912","EmrManagedSlaveSecurityGroup":"sg-07818b5690a50b3f1","EmrManagedMasterSecurityGroup":"sg-
0e2f5550a2cb98f79"}' --service-role EMR_DefaultRole --enable-debugging --release-label emr-6.2.0 --log-
uri 's3n://aws-logs-604905954159-us-east-1/elasticmapreduce/' --name 'BigData' --instance-groups
'[{"InstanceCount":1,"BidPrice":"OnDemandPrice","EbsConfiguration":{"EbsBlockDeviceConfigs":[{"VolumeSpe
cification":{"SizeInGB":32,"VolumeType":"gp2"},"VolumesPerInstance":2}]},"InstanceGroupType":"MASTER","I
nstanceType":"m4.xlarge","Name":"Master -
1"},{"InstanceCount":1,"BidPrice":"OnDemandPrice","EbsConfiguration":{"EbsBlockDeviceConfigs":[{"VolumeS
pecification":{"SizeInGB":32,"VolumeType":"gp2"},"VolumesPerInstance":2}]},"InstanceGroupType":"CORE","I
nstanceType":"m4.xlarge","Name":"Core - 2"}]' --scale-down-behavior TERMINATE_AT_TASK_COMPLETION --
region us-east-1
```

Cluster lifecycle

Creating a cluster (it takes ~10 minutes)

- A cluster cannot be stopped
- It can only be terminated



Cluster lifecycle

`STARTING`: EMR provisions EC2 instances for each required instance

`BOOTSTRAPPING`: EMR runs actions that you specify on each instance

- E.g., install custom applications and perform customizations

Amazon EMR installs the native applications

- E.g., Hive, Hadoop, Spark, and so on

`RUNNING`: a step for the cluster is currently being run

- Cluster sequentially runs any steps that you specified when you created the cluster

`WAITING`: after steps run successfully

`TERMINATING`: after manual shut down

- Any data stored on the cluster is deleted

Cluster: EMR

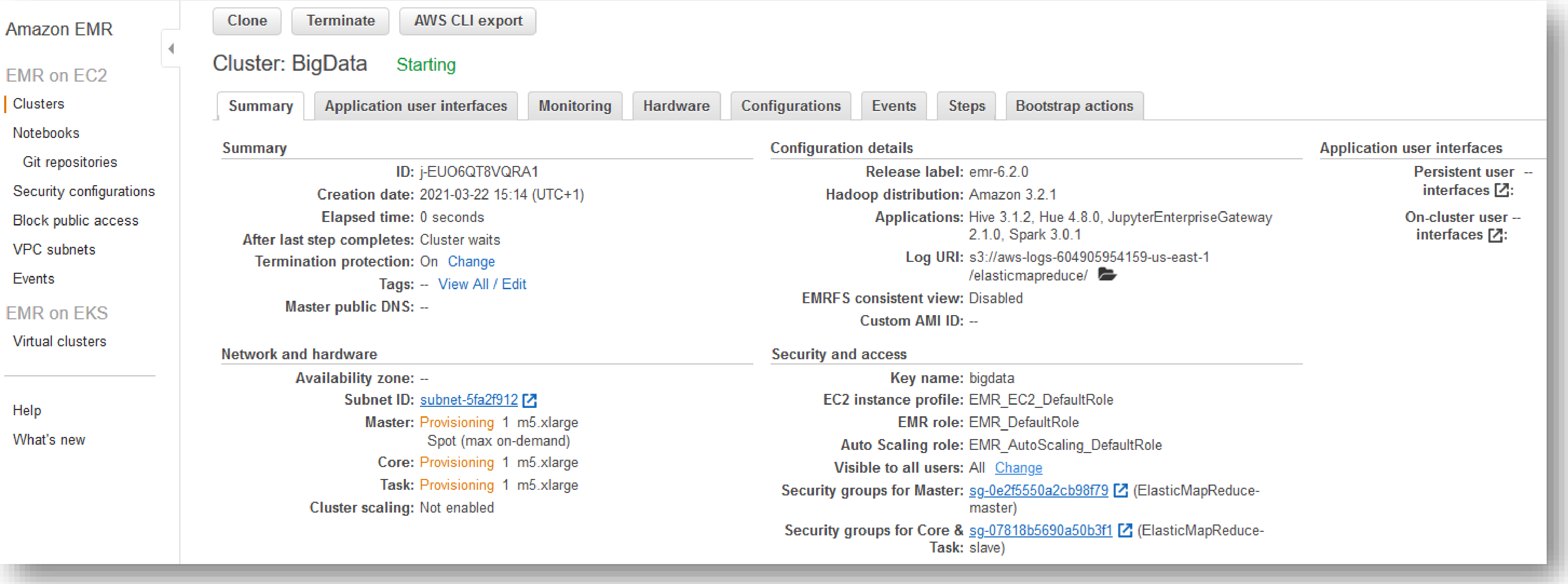
A **step** is a user-defined unit of processing

- E.g., one algorithm that manipulates the data

Step states

- `PENDING:` The step is waiting to be run
- `RUNNING:` The step is currently running
- `COMPLETED:` The step completed successfully
- `CANCELLED:` The step was cancelled before running because an earlier step failed
- `FAILED:` The step failed while running

Running the cluster



Cluster: BigData Starting

Summary

ID: j-EUO6QT8VQRA1
Creation date: 2021-03-22 15:14 (UTC+1)
Elapsed time: 0 seconds
After last step completes: Cluster waits
Termination protection: On Change
Tags: -- View All / Edit
Master public DNS: --

Network and hardware

Availability zone: --
Subnet ID: subnet-5fa2f912
Master: Provisioning 1 m5.xlarge Spot (max on-demand)
Core: Provisioning 1 m5.xlarge
Task: Provisioning 1 m5.xlarge
Cluster scaling: Not enabled

Configuration details

Release label: emr-6.2.0
Hadoop distribution: Amazon 3.2.1
Applications: Hive 3.1.2, Hue 4.8.0, JupyterEnterpriseGateway 2.1.0, Spark 3.0.1
Log URI: s3://aws-logs-604905954159-us-east-1/elasticmapreduce/
EMRFS consistent view: Disabled
Custom AMI ID: --

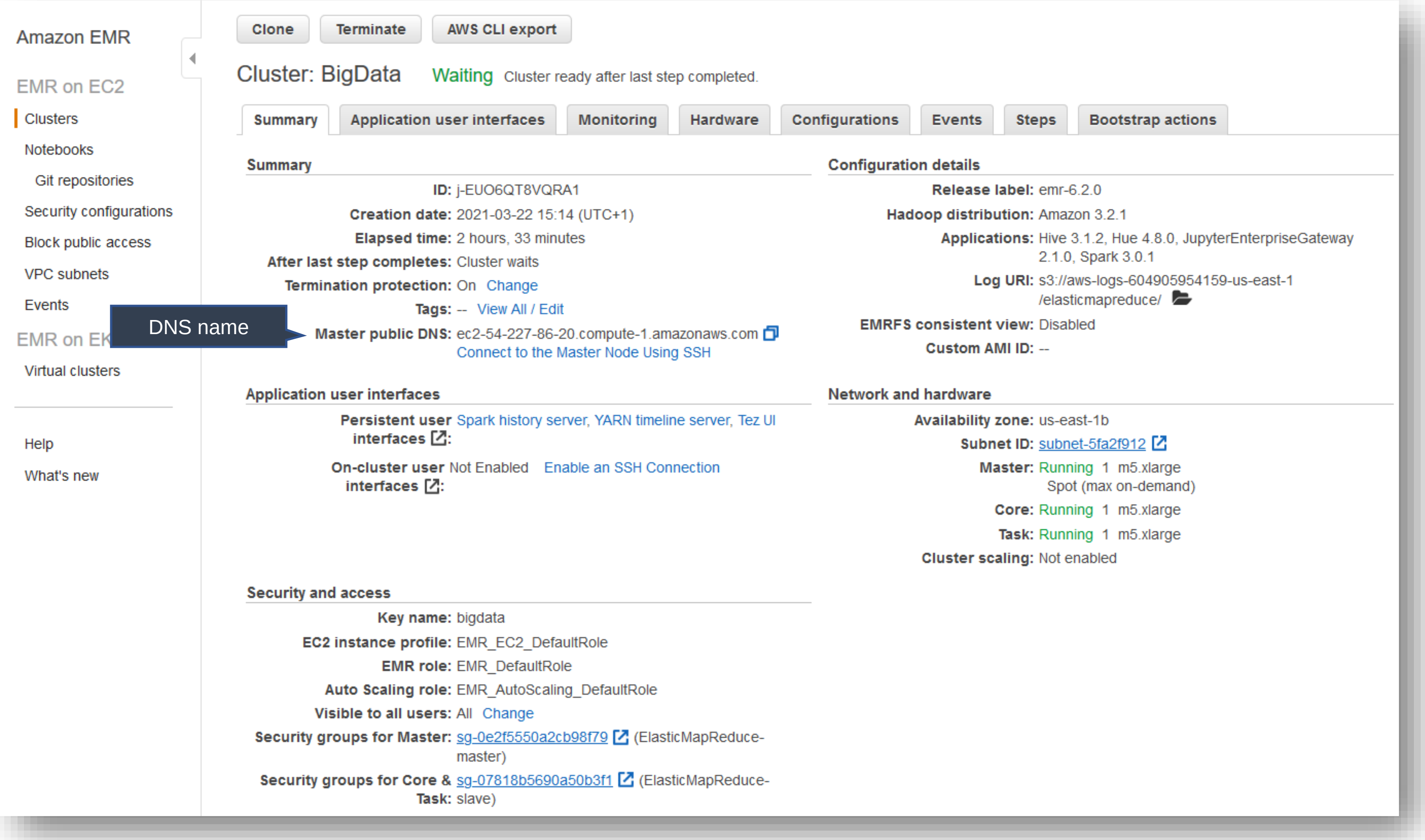
Security and access

Key name: bigdata
EC2 instance profile: EMR_EC2_DefaultRole
EMR role: EMR_DefaultRole
Auto Scaling role: EMR_AutoScaling_DefaultRole
Visible to all users: All Change
Security groups for Master: sg-0e2f5550a2cb98f79 (ElasticMapReduce-master)
Security groups for Core & Task: sg-07818b5690a50b3f1 (ElasticMapReduce-slave)

Application user interfaces

Persistent user interfaces: --
On-cluster user interfaces: --

Running the cluster



DNS name

Cluster: BigData Waiting Cluster ready after last step completed.

Summary

ID: j-EUO6QT8VQRA1
Creation date: 2021-03-22 15:14 (UTC+1)
Elapsed time: 2 hours, 33 minutes
After last step completes: Cluster waits
Termination protection: On Change
Tags: -- View All / Edit
Master public DNS: ec2-54-227-86-20.compute-1.amazonaws.com
Connect to the Master Node Using SSH

Application user interfaces

Persistent user interfaces: Spark history server, YARN timeline server, Tez UI
On-cluster user interfaces: Not Enabled Enable an SSH Connection

Security and access

Key name: bigdata
EC2 instance profile: EMR_EC2_DefaultRole
EMR role: EMR_DefaultRole
Auto Scaling role: EMR_AutoScaling_DefaultRole
Visible to all users: All Change
Security groups for Master: sg-0e2f5550a2cb98f79 (ElasticMapReduce-master)
Security groups for Core & Task: sg-07818b5690a50b3f1 (ElasticMapReduce-slave)

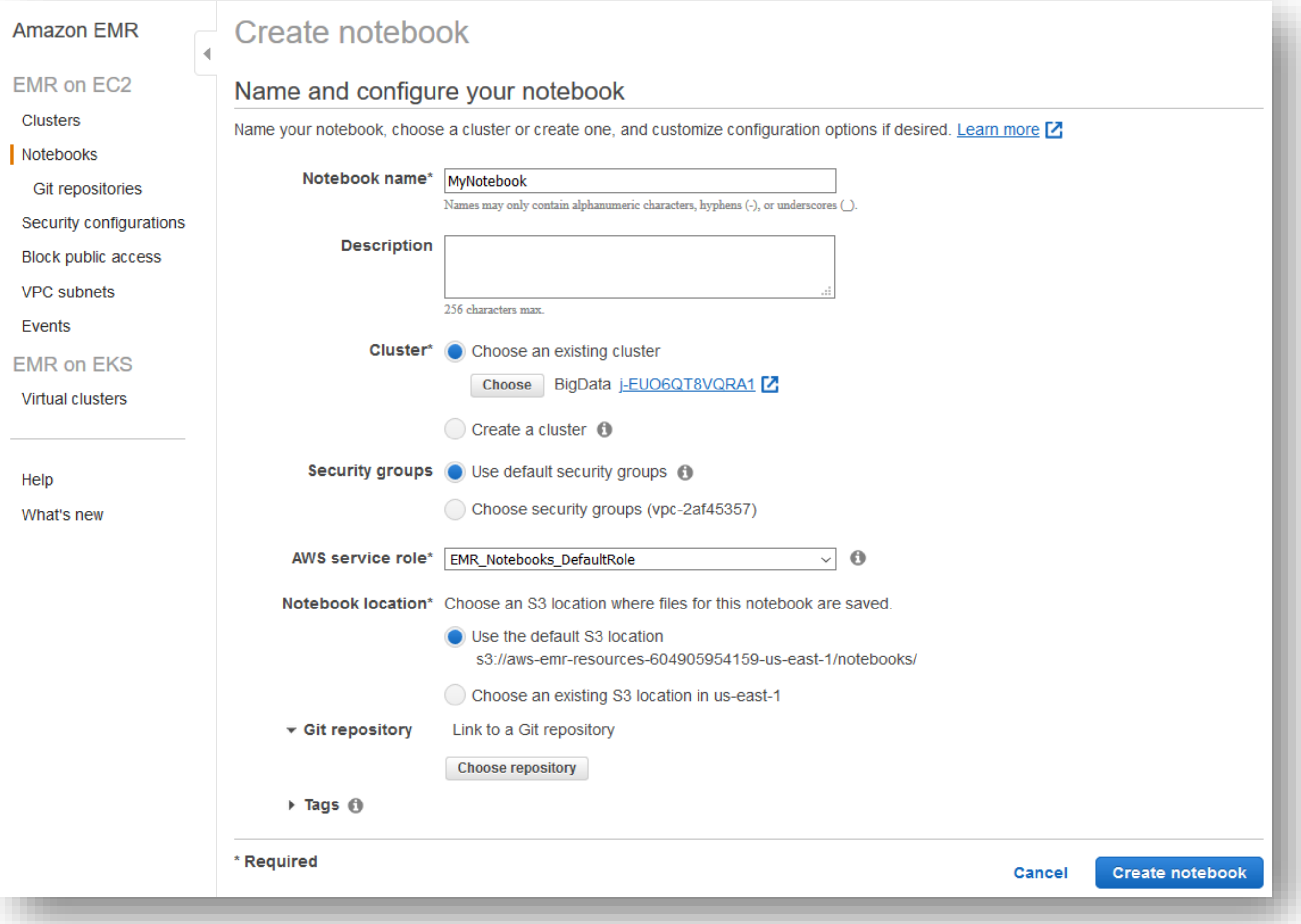
Configuration details

Release label: emr-6.2.0
Hadoop distribution: Amazon 3.2.1
Applications: Hive 3.1.2, Hue 4.8.0, JupyterEnterpriseGateway 2.1.0, Spark 3.0.1
Log URI: s3://aws-logs-604905954159-us-east-1/elasticmapreduce/
EMRFS consistent view: Disabled
Custom AMI ID: --

Network and hardware

Availability zone: us-east-1b
Subnet ID: subnet-5fa2f912
Master: Running 1 m5.xlarge
Spot (max on-demand)
Core: Running 1 m5.xlarge
Task: Running 1 m5.xlarge
Cluster scaling: Not enabled

Creating a notebook



Create notebook

Name and configure your notebook

Name your notebook, choose a cluster or create one, and customize configuration options if desired. Learn more

Notebook name* MyNotebook
Names may only contain alphanumeric characters, hyphens (-), or underscores (_).

Description
256 characters max.

Cluster* ● Choose an existing cluster
Choose BigData j-EUO6QT8VQRA1
○ Create a cluster

Security groups ● Use default security groups
○ Choose security groups (vpc-2af45357)

AWS service role* EMR_Notebooks_DefaultRole

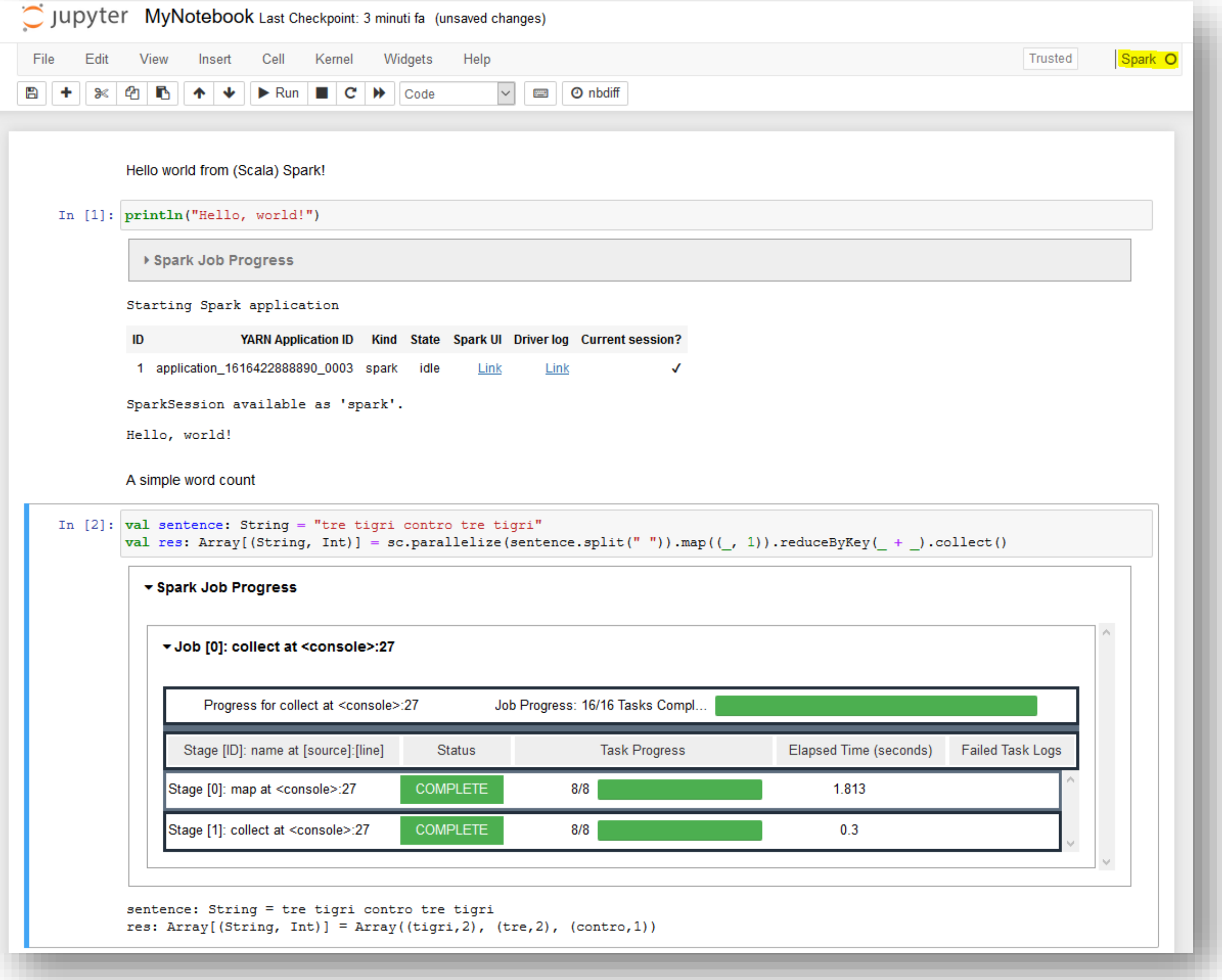
Notebook location* Choose an S3 location where files for this notebook are saved.
● Use the default S3 location
s3://aws-emr-resources-604905954159-us-east-1/notebooks/
○ Choose an existing S3 location in us-east-1

▾ Git repository Link to a Git repository
Choose repository

▸ Tags

* Required

Hello, world!



Hello world from (Scala) Spark!

```
In [1]: println("Hello, world!")
```

▸ Spark Job Progress

```
Starting Spark application
```

ID	YARN Application ID	Kind	State	Spark UI	Driver log	Current session?
1	application_1616422888890_0003	spark	idle	Link	Link	✓

```
SparkSession available as 'spark'.

Hello, world!
```

A simple word count

```
In [2]: val sentence: String = "tre tigri contro tre tigri"
        val res: Array[(String, Int)] = sc.parallelize(sentence.split(" ")).map((_, 1)).reduceByKey(_ + _).collect()
```

▾ Spark Job Progress

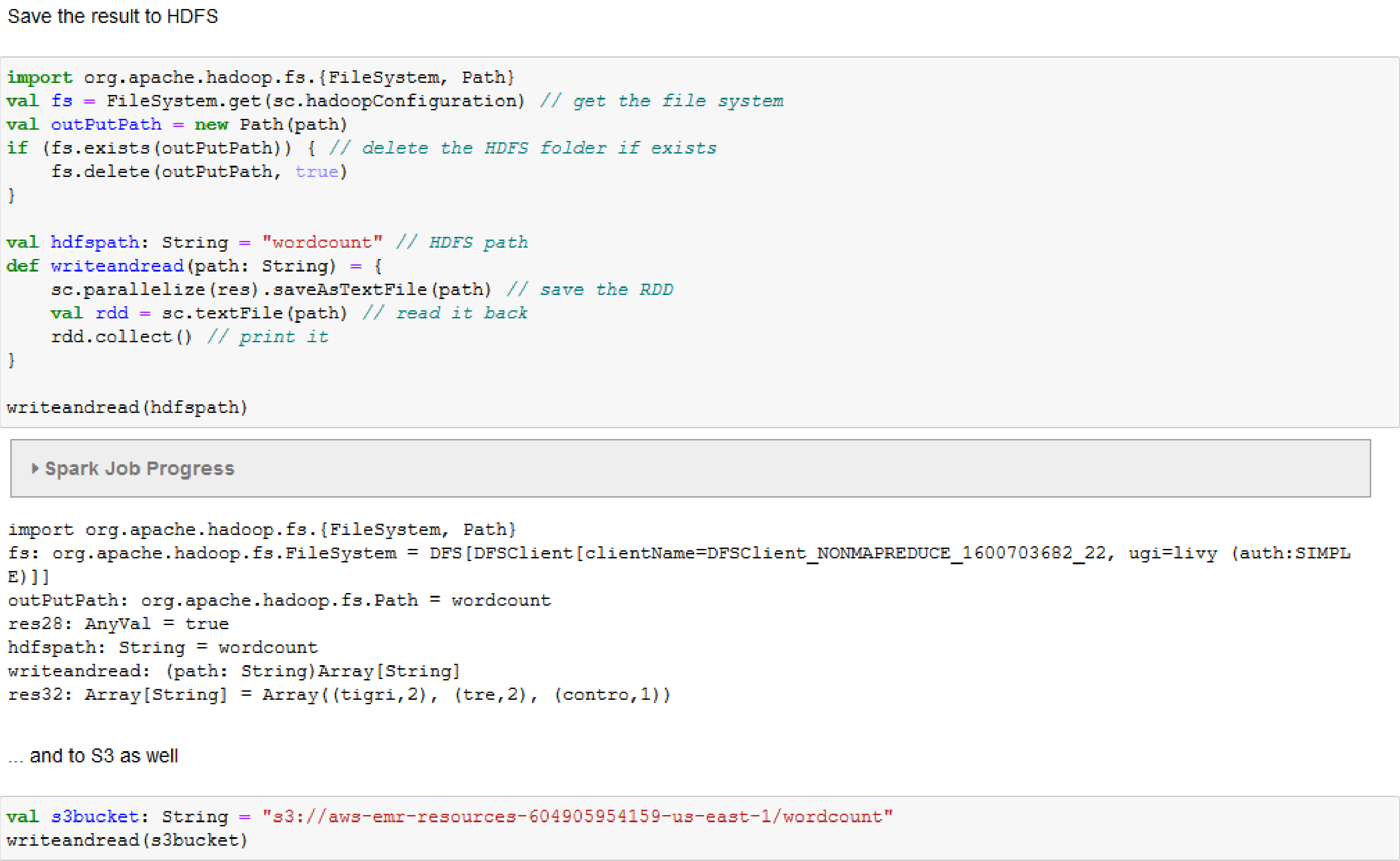
▾ Job [0]: collect at <console>:27

Progress for collect at <console>:27 Job Progress: 16/16 Tasks Compl...

Stage [ID]: name at [source]:[line]	Status	Task Progress	Elapsed Time (seconds)	Failed Task Logs
Stage [0]: map at <console>:27	COMPLETE	8/8	1.813	
Stage [1]: collect at <console>:27	COMPLETE	8/8	0.3	

```
sentence: String = tre tigri contro tre tigri
res: Array[(String, Int)] = Array((tigri,2), (tre,2), (contro,1))
```

Add some storage



Save the result to HDFS

```
import org.apache.hadoop.fs.{FileSystem, Path}
val fs = FileSystem.get(sc.hadoopConfiguration) // get the file system
val outPutPath = new Path(path)
if (fs.exists(outPutPath)) { // delete the HDFS folder if exists
    fs.delete(outPutPath, true)
}

val hdfspath: String = "wordcount" // HDFS path
def writeandread(path: String) = {
    sc.parallelize(res).saveAsTextFile(path) // save the RDD
    val rdd = sc.textFile(path) // read it back
    rdd.collect() // print it
}

writeandread(hdfspath)
```

```
import org.apache.hadoop.fs.{FileSystem, Path}
fs: org.apache.hadoop.fs.FileSystem = DFS[DFSClient[clientName=DFSClient_NONMAPREDUCE_1600703682_22, ugi=livy (auth:SIMPL
E)]]
outPutPath: org.apache.hadoop.fs.Path = wordcount
res28: AnyVal = true
hdfspath: String = wordcount
writeandread: (path: String)Array[String]
res32: Array[String] = Array((tigri,2), (tre,2), (contro,1))
```

... and to S3 as well

```
val s3bucket: String = "s3://aws-emr-resources-604905954159-us-east-1/wordcount"
writeandread(s3bucket)
```

Running a Spark Job

Connect using SSH

Install git

Clone & build the project

```
ssh -i ~/bigdata.pem hadoop@ec2-54-242-176-32.compute-1.amazonaws.com

sudo yum install git -y

git clone https://github.com/w4bo/spark-word-count.git

cd spark-word-count

./gradlew

spark-submit --class it.unibo.big.WordCount build/libs/WordCount-all.jar
                     s3://aws-bucket-bigdata2021/inferno.txt
```

Other services: HUE

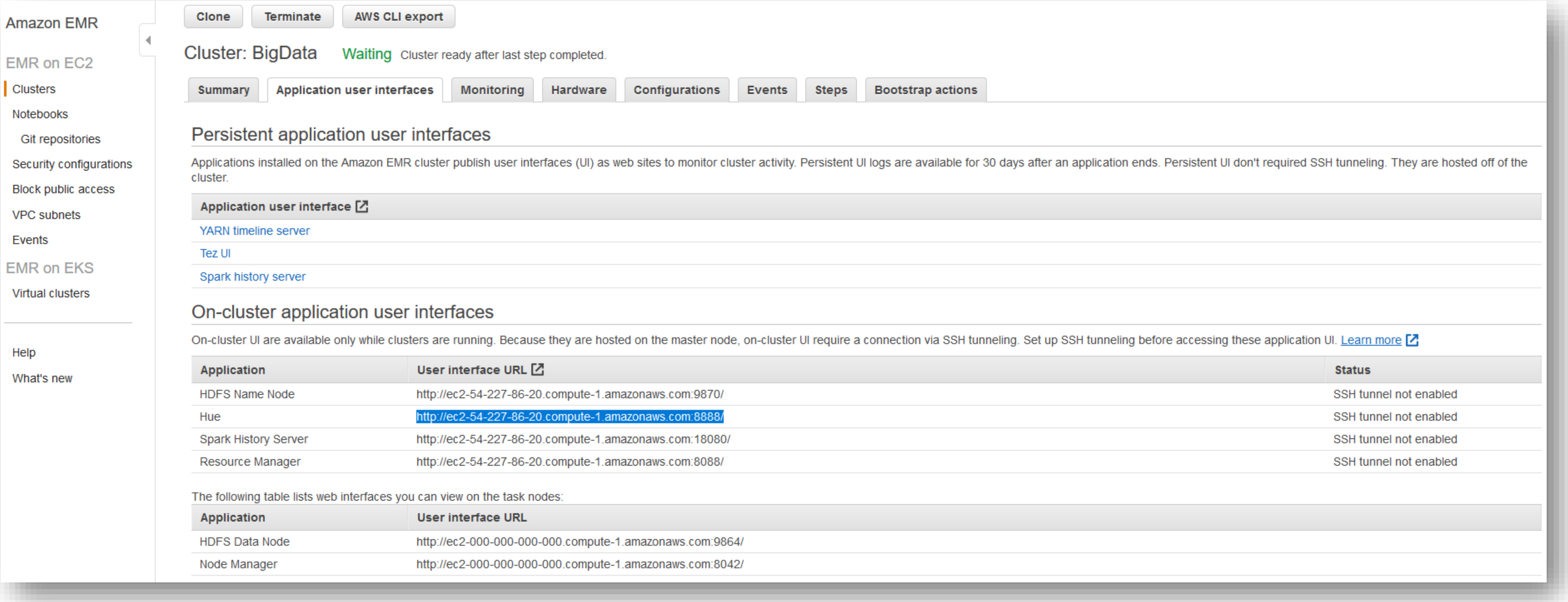
Connecting to Hue

- I.e., connecting to any HTTP interface hosted on the master node of a cluster

To view the Hue web user interface

- Set Up an SSH Tunnel to the Master Node Using Dynamic Port Forwarding
- Type the following address in your browser to open the Hue web interface
  - http://master-public-DNS:8888
  - Where master-public-DNS is the public DNS name of the master node
- If you are the administrator logging in for the first time
  - Enter a username and password to create your Hue superuser account
  - Otherwise, type your username and password and select Create account

Other services: HUE



Cluster: BigData Waiting Cluster ready after last step completed.

Persistent application user interfaces

Applications installed on the Amazon EMR cluster publish user interfaces (UI) as web sites to monitor cluster activity. Persistent UI logs are available for 30 days after an application ends. Persistent UI don't required SSH tunneling. They are hosted off of the cluster.

Application user interface
YARN timeline server
Tez UI
Spark history server

On-cluster application user interfaces

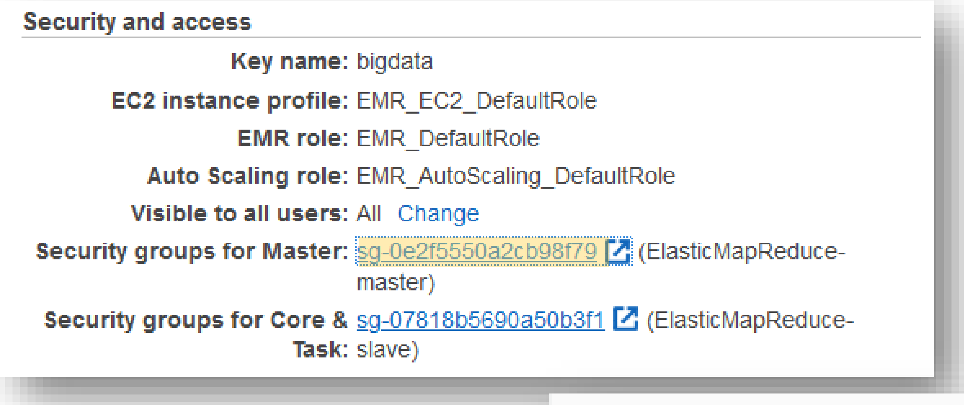
On-cluster UI are available only while clusters are running. Because they are hosted on the master node, on-cluster UI require a connection via SSH tunneling. Set up SSH tunneling before accessing these application UI. Learn more

Application	User interface URL	Status
HDFS Name Node	http://ec2-54-227-86-20.compute-1.amazonaws.com:9870/	SSH tunnel not enabled
Hue	http://ec2-54-227-86-20.compute-1.amazonaws.com:8888/	SSH tunnel not enabled
Spark History Server	http://ec2-54-227-86-20.compute-1.amazonaws.com:18080/	SSH tunnel not enabled
Resource Manager	http://ec2-54-227-86-20.compute-1.amazonaws.com:8088/	SSH tunnel not enabled

The following table lists web interfaces you can view on the task nodes:

Application	User interface URL
HDFS Data Node	http://ec2-000-000-000-000.compute-1.amazonaws.com:9864/
Node Manager	http://ec2-000-000-000-000.compute-1.amazonaws.com:8042/

Set Up an SSH Tunnel



Security and access

Key name: bigdata

EC2 instance profile: EMR_EC2_DefaultRole

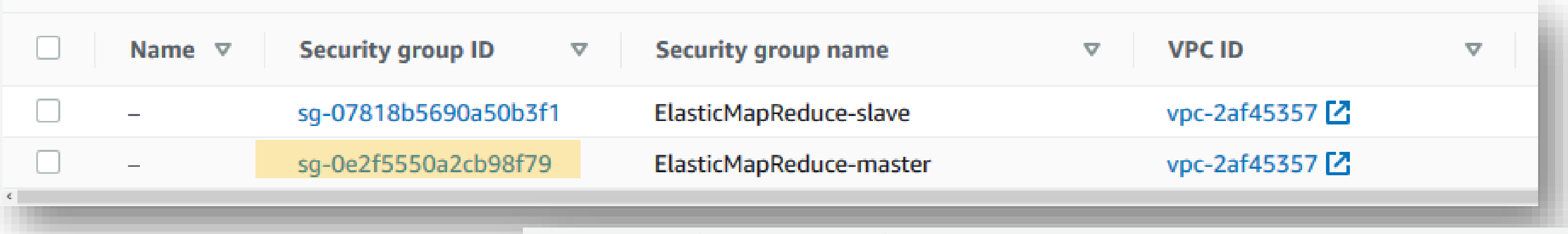
EMR role: EMR_DefaultRole

Auto Scaling role: EMR_AutoScaling_DefaultRole

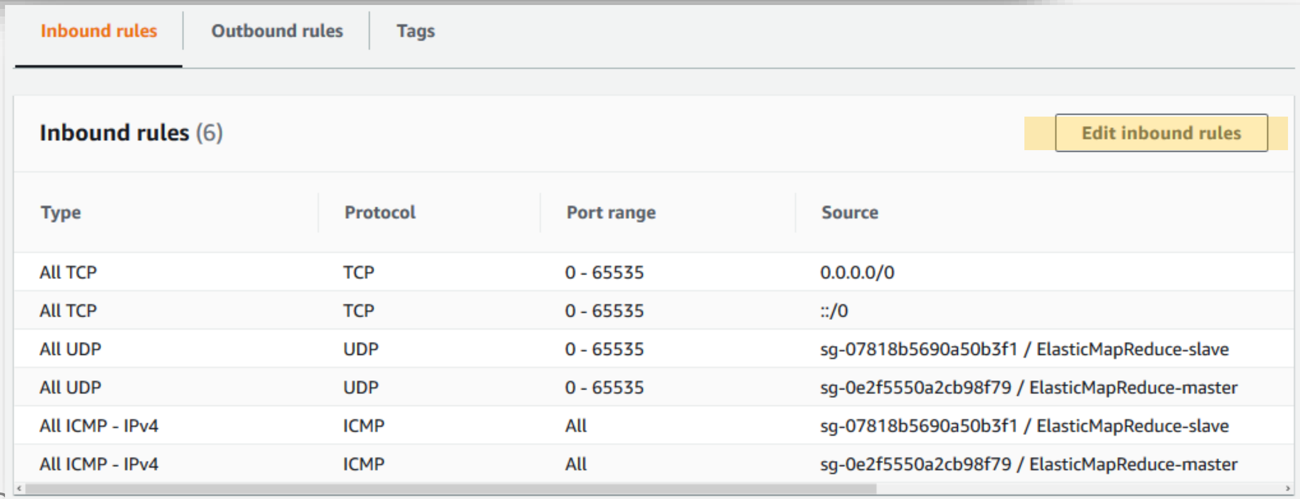
Visible to all users: All Change

Security groups for Master: sg-0e2f5550a2cb98f79 (ElasticMapReduce-master)

Security groups for Core & Task: sg-07818b5690a50b3f1 (ElasticMapReduce-slave)



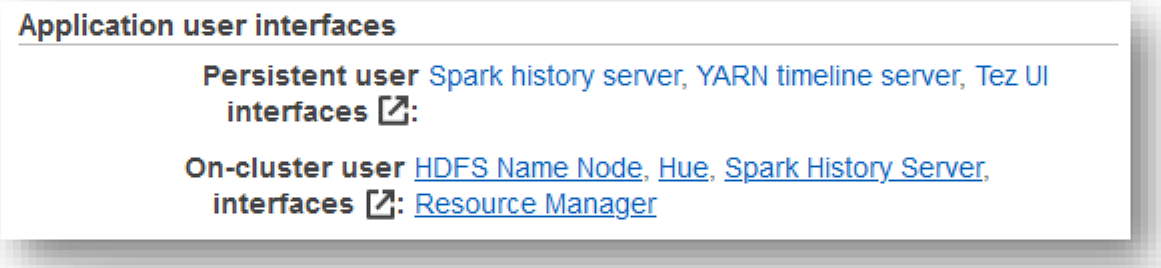
□	Name	Security group ID	Security group name	VPC ID
□	–	sg-07818b5690a50b3f1	ElasticMapReduce-slave	vpc-2af45357
□	–	sg-0e2f5550a2cb98f79	ElasticMapReduce-master	vpc-2af45357



Inbound rules (6)

Type	Protocol	Port range	Source
All TCP	TCP	0 - 65535	0.0.0.0/0
All TCP	TCP	0 - 65535	::/0
All UDP	UDP	0 - 65535	sg-07818b5690a50b3f1 / ElasticMapReduce-slave
All UDP	UDP	0 - 65535	sg-0e2f5550a2cb98f79 / ElasticMapReduce-master
All ICMP - IPv4	ICMP	All	sg-07818b5690a50b3f1 / ElasticMapReduce-slave
All ICMP - IPv4	ICMP	All	sg-0e2f5550a2cb98f79 / ElasticMapReduce-master

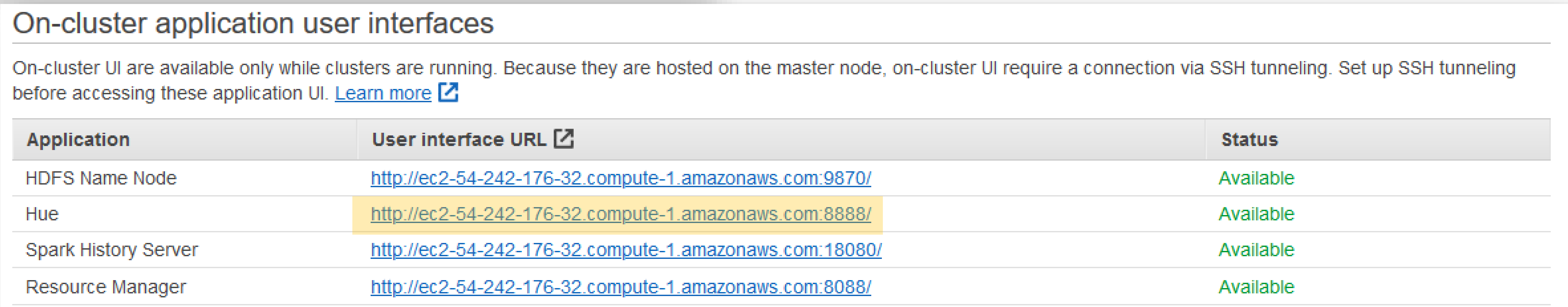
Connect to HUE



Application user interfaces

Persistent user interfaces: Spark history server, YARN timeline server, Tez UI

On-cluster user interfaces: HDFS Name Node, Hue, Spark History Server, Resource Manager

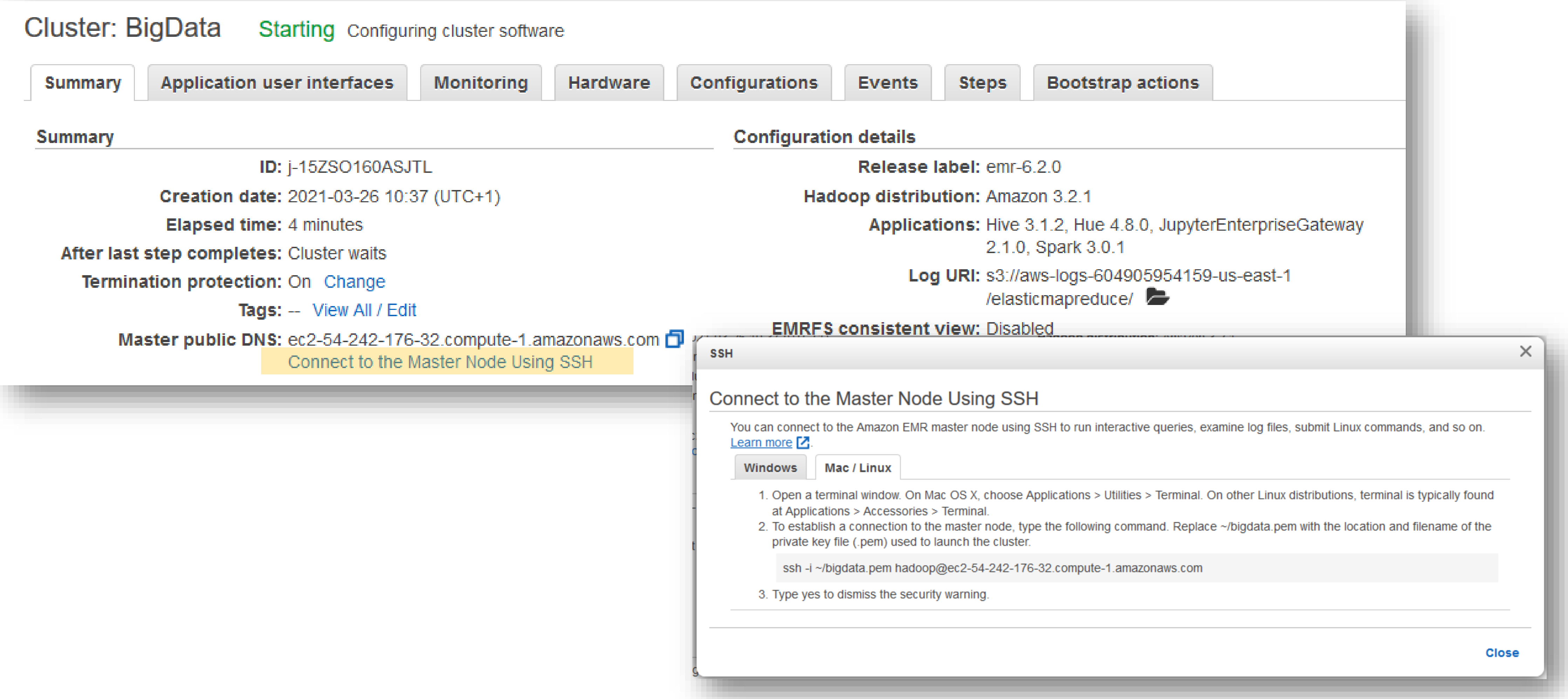


On-cluster application user interfaces

On-cluster UI are available only while clusters are running. Because they are hosted on the master node, on-cluster UI require a connection via SSH tunneling. Set up SSH tunneling before accessing these application UI. Learn more

Application	User interface URL	Status
HDFS Name Node	http://ec2-54-242-176-32.compute-1.amazonaws.com:9870/	Available
Hue	http://ec2-54-242-176-32.compute-1.amazonaws.com:8888/	Available
Spark History Server	http://ec2-54-242-176-32.compute-1.amazonaws.com:18080/	Available
Resource Manager	http://ec2-54-242-176-32.compute-1.amazonaws.com:8088/	Available

Connect using SSH



Cluster: BigData Starting Configuring cluster software

Summary

ID: j-15ZSO160ASJTL
Creation date: 2021-03-26 10:37 (UTC+1)
Elapsed time: 4 minutes
After last step completes: Cluster waits
Termination protection: On Change
Tags: -- View All / Edit
Master public DNS: ec2-54-242-176-32.compute-1.amazonaws.com
Connect to the Master Node Using SSH

Configuration details

Release label: emr-6.2.0
Hadoop distribution: Amazon 3.2.1
Applications: Hive 3.1.2, Hue 4.8.0, JupyterEnterpriseGateway 2.1.0, Spark 3.0.1
Log URI: s3://aws-logs-604905954159-us-east-1/elasticmapreduce/
EMRFS consistent view: Disabled

SSH

Connect to the Master Node Using SSH

You can connect to the Amazon EMR master node using SSH to run interactive queries, examine log files, submit Linux commands, and so on. Learn more.

1. Open a terminal window. On Mac OS X, choose Applications > Utilities > Terminal. On other Linux distributions, terminal is typically found at Applications > Accessories > Terminal.
2. To establish a connection to the master node, type the following command. Replace ~/bigdata.pem with the location and filename of the private key file (.pem) used to launch the cluster.

```
ssh -i ~/bigdata.pem hadoop@ec2-54-242-176-32.compute-1.amazonaws.com
```

3. Type yes to dismiss the security warning.