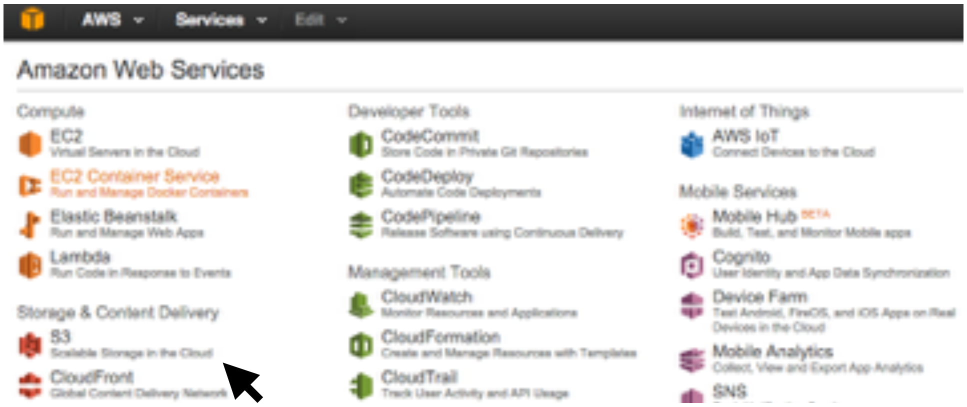
Running stmpy on Amazon Web Services

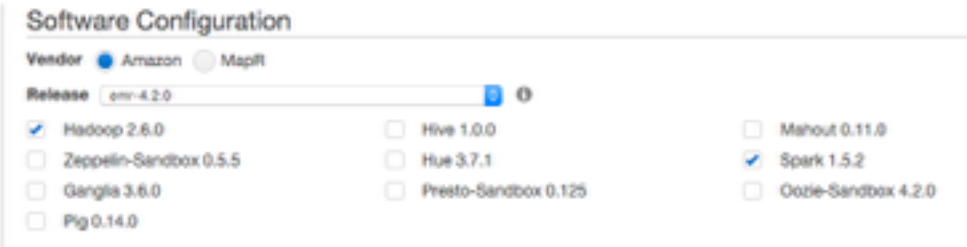
This guide illustrates the configuration process needed to run stmpy on a Spark cluster via Amazon Web Services. It assumes familiarity with the bash shell and the Jupyter/iPython notebook server.

I. Sign up for an account with AWS and log into the console. Set up an RSA key pair for authentication as described in this guide. Save your certificate in a safe place on your computer.

II. Clone the contents of the stmpy repository locally. From the AWS console, enter the S3 screen. Using the S3 interface upload the file *aws_bootstrap.bash* from the stmpy repository to one of your S3 buckets. This bash script contains the bootstrap instructions for the cluster.


AWS
Services
Edit
Amazon Web Services
Compute
EC2
Virtual Servers in the Cloud
EC2 Container Service
Run and Manage Docker Containers
Elastic Beanstalk
Run and Manage Web Apps
Lambda
Run Code in Response to Events
Storage & Content Delivery
S3
Scalable Storage in the Cloud
CloudFront
Global Content Delivery Network
Developer Tools
CodeCommit
Store Code in Private Git Repositories
CodeDeploy
Automate Code Deployments
CodePipeline
Release Software using Continuous Delivery
Management Tools
CloudWatch
Monitor Resources and Applications
CloudFormation
Create and Manage Resources with Templates
CloudTrail
Track User Activity and API Usage
Internet of Things
AWS IoT
Connect Devices to the Cloud
Mobile Services
Mobile Hub BETA
Build, Test, and Monitor Mobile apps
Cognito
User Identity and App Data Synchronization
Device Farm
Test Android, FireOS, and iOS Apps on Real Devices in the Cloud
Mobile Analytics
Collect, View and Export App Analytics
SNS

III. Now go back to the console and enter the EMR (Elastic MapReduce) screen. Click *Create Cluster* and then *Go to advanced options*. In the *Software and Steps* tab, make sure to flag Hadoop and Spark for installation, then click *Next*. Everything else is optional.


Software Configuration

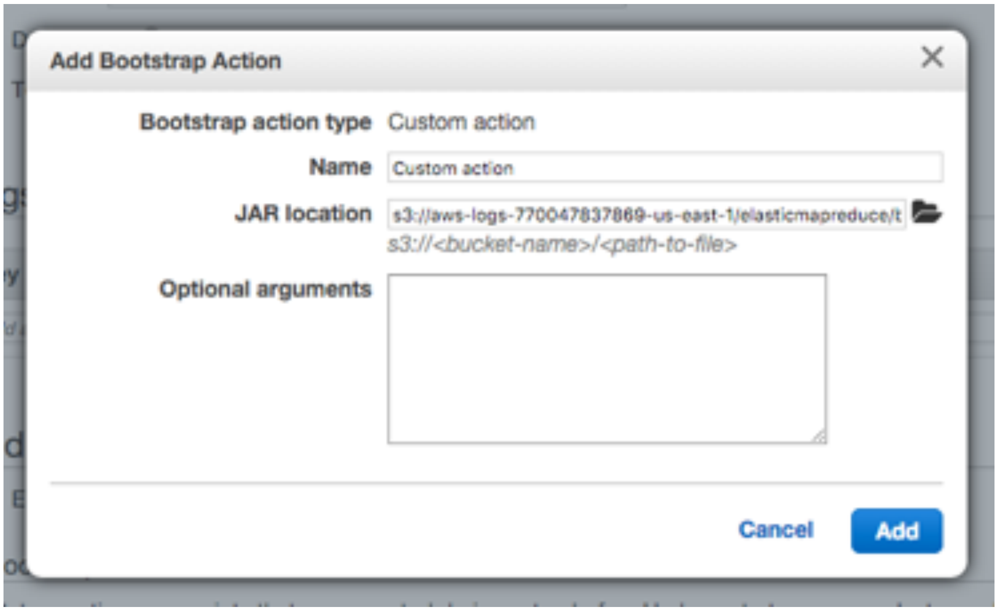
Vendor ● Amazon ○ MapR

Release emr-4.2.0

☑ Hadoop 2.6.0	□ Hive 1.0.0	□ Mahout 0.11.0
□ Zeppelin-Sandbox 0.5.5	□ Hue 3.7.1	☑ Spark 1.5.2
□ Ganglia 3.6.0	□ Presto-Sandbox 0.125	□ Oozie-Sandbox 4.2.0
□ Pig 0.14.0		

IV. In the *Hardware* tab choose the number of machines in your cluster as well as their specifications according to your computing needs. Click *Next*.

V. Now we are in the *General Cluster Settings* tab. Expand the *Bootstrap Actions* section and choose to add a custom bootstrap action from the drop-down menu. Click on the folder icon to select a JAR location. A navigation menu showing your S3 buckets will appear. Use the navigation menu to select the *aws_bootstrap.bash* file that was previously saved. Click *Select,* then *Add.*



Add Bootstrap Action

Bootstrap action type: Custom action

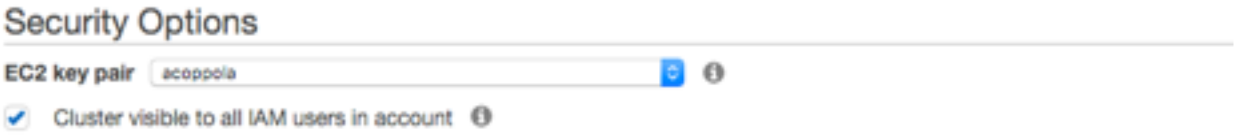
Name: Custom action

JAR location: s3://aws-logs-770047837869-us-east-1/elasticmapreduce/t

s3://<bucket-name>/<path-to-file>

Optional arguments:

VI. In the *Security* tab, make sure to select the key pair you previously created from the *EC2 key pair* drop-down menu. If you don't do this, you will not be able to log into your cluster. Leave all other settings to their defaults.



Security Options

EC2 key pair: acoppola

☑ Cluster visible to all IAM users in account

VII. Now click *Create cluster*. The setup should take fifteen minutes or so. The cluster is ready once it is in a *Waiting* state.

VIII. Once the cluster is ready, open a bash terminal window and navigate to the folder containing the RSA certificate your previously created. Note the master public DNS from the EMR web interface. You can now SSH into the cluster via the following command (substitute values appropriately): `ssh -i <MY_KEY>.pem hadoop@<MY_DNS>`

IX. PySpark is now available in the shell via the pyspark binary, which you can use for interactive or batch computation. From pyspark, you can simply type `import stmpy` to have access to stmpy.

X. You are now ready to use stmpy on AWS! Remember to terminate your cluster when finished in order not to incur unnecessary charges.

Follow the following extra steps if you wish to have access to stmpy via the iPython/Jupyter notebook interface (this functionality is in beta and may have bugs):

XI. From the EMR cluster page, follow the instructions in the *Enable Web Connection* tab to establish an SSH tunnel to the AWS cluster. This will involve installing FoxyProxy in Google Chrome. The tunnel will allow you to access the iPython notebook server and the Spark console from your browser.

XII. Once you have established an SSH connection to your master node, run the following commands in the terminal:

```
export SPARK_HOME=/usr/lib/spark
wget https://dl.dropboxusercontent.com/u/113867121/stmpy/jupyter_setup.bash
bash jupyter_setup.bash
```

XIII. Now type `jupiter notebook` in the terminal. This will start the iPython server. Now use your local browser to navigate to `https://<MY_DNS>:8888`. You can now use the iPython server to start a new notebook in which to conduct your analysis. To access Spark and stmpy, run the following commands at the top of the notebook:

```
import findspark; findspark.init()

import pyspark, stmpy
sc = pyspark.SparkContext(appName=“myAppName”)
```