數據科學與大數據分析

AWS運用介紹
2017.05

環境準備

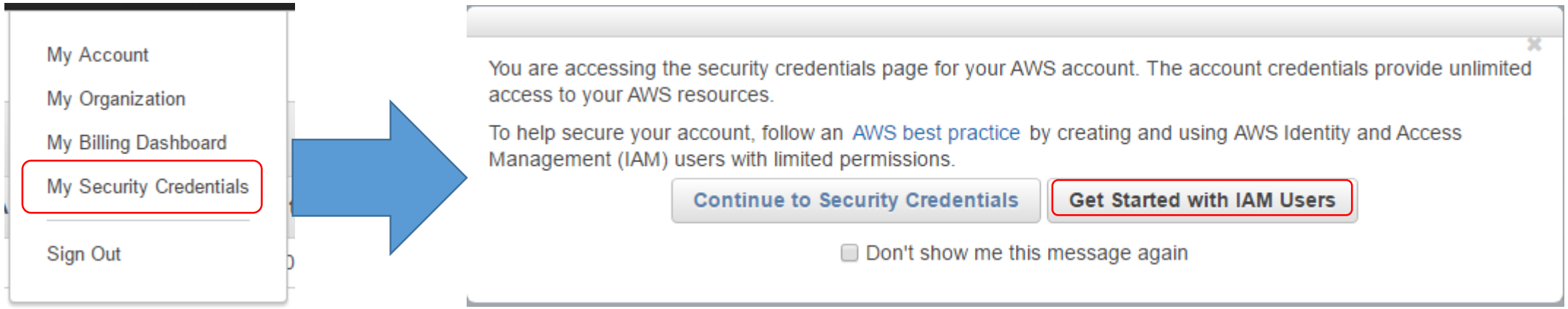
- AWS存取權限設定
- Spark-ec2套件
  - https://github.com/amplab/spark-ec2
  - Branch 2.0
  - 快速佈署Spark AWS Cluster
- 雲端運算範例
  - 如果會用到AWS S3存取檔案，僅能搭配Hadoop 2.4
    反之，任何prebuilt for Hadoop版本皆可
    Spark 1.X/2.X prebuilt for Hadoop 2.4 Spark-2.1.1-bin-hadoop2.4.tgz

spark-ec2預設使用
Hadoop 2.4

- CausalImpact R
  - 單機BSTS分析

AWS存取權限設定

Step1 . 建立 IAM USER



Step2 . 建立 IAM USER之Access Keypairs

Users ➔ UserDemo (create yours) ➔ Security credentials ➔ create access key ➔ accessKeys.csv

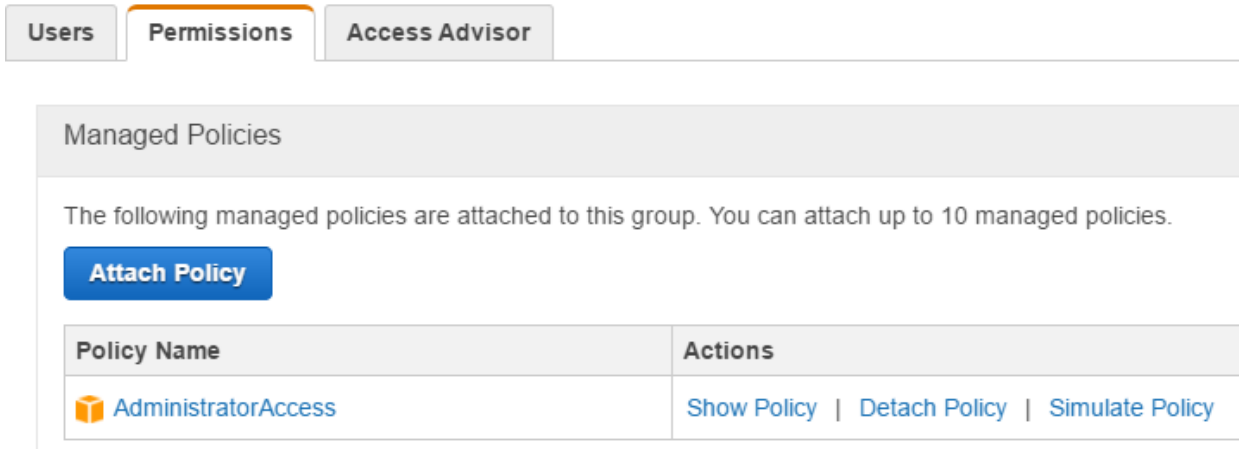
accessKeys.csv 請妥善保存，勿存放於公開空間

```
AWS_ACCESS_KEY_ID=ABCDEFGHIJKLMNOPQRST
AWS_SECRET_ACCESS_KEY=rANDomrANDomrANDomrANDomrANDom
```

AWS存取權限設定

Step3 . 新增群組（可限制不同服務存取權限）將IAM USER

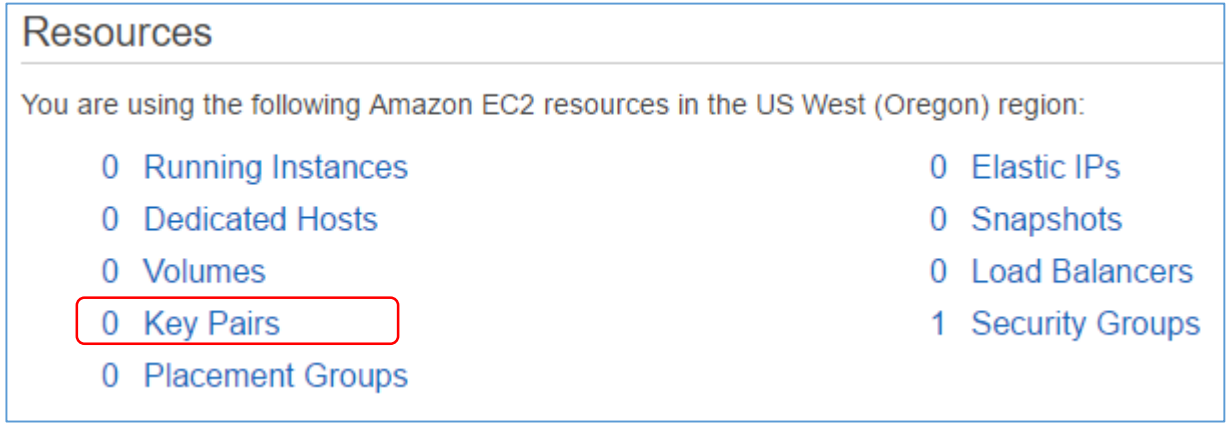
Groups ➔ GroupDemo (create yours) ➔ Permissions ➔ Attach Policy : ##ServiceAccess
(視需求開啟適當權限)

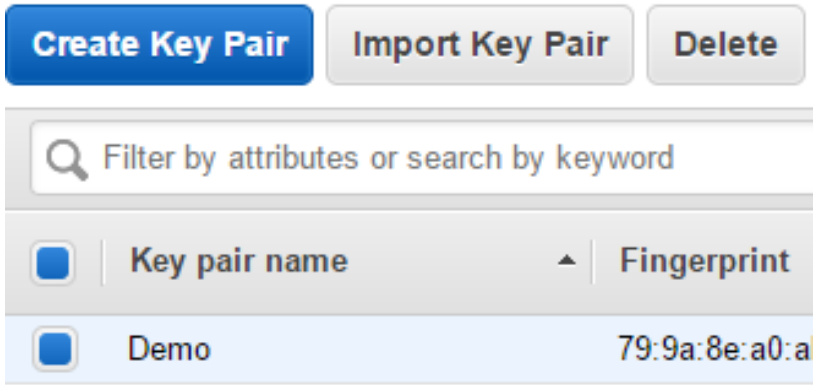


Step4 . 將 IAM USER 加入特定權限群組中

Users ➔ UserDemo ➔ Groups ➔ Add user to groups : UserDemo

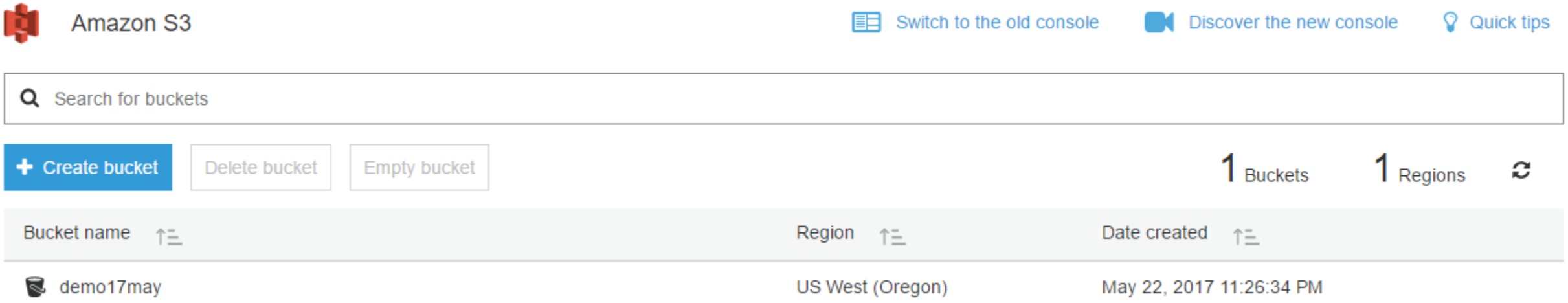
AWS EC2公開金鑰設定



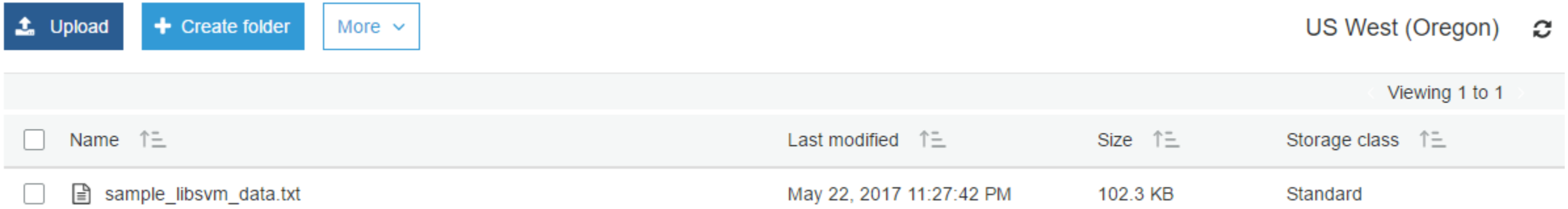


1. 自動下載 **Demo.pem**
   未來用於**ec2**伺服器登入之認證金鑰
2. 將**Demo.pem**存入**Linux OS**中
3. 將**Demo.pem**權限改為**400 (**自己唯讀**)**
   # chmod 400 ./Demo.pem

AWS S3設定



Create bucket (名稱只能小寫，在每個區域之bucket名稱都不可重複使用)
➔ Region (選與EC2服務相同區域) ➔ 上傳檔案



Spark-ec2

- 用於快速佈署Spark AWS Cluster
- Spark 1.X版本以前，內帶Spark-ec2套件
- Spark 2.X版本以後，Spark-ec2已獨立由amplab維護

下載

```
# wget https://github.com/amplab/spark-ec2/archive/branch-2.0.zip
# unzip branch-2.0.zip
```

執行檔位於 **./spark-ec2-branch-2.0/spark-ec2**

Spark-ec2

使用方式

https://github.com/amplab/spark-ec2/tree/branch-2.0

先匯入**accessKeys.csv** **(請妥善保存，勿存放於公開空間)**

```
#  export AWS_ACCESS_KEY_ID=ABCDEFGHIJKLMNOPQRST
#  export AWS_SECRET_ACCESS_KEY=rANDomrANDomrANDomrANDomrANDom
```

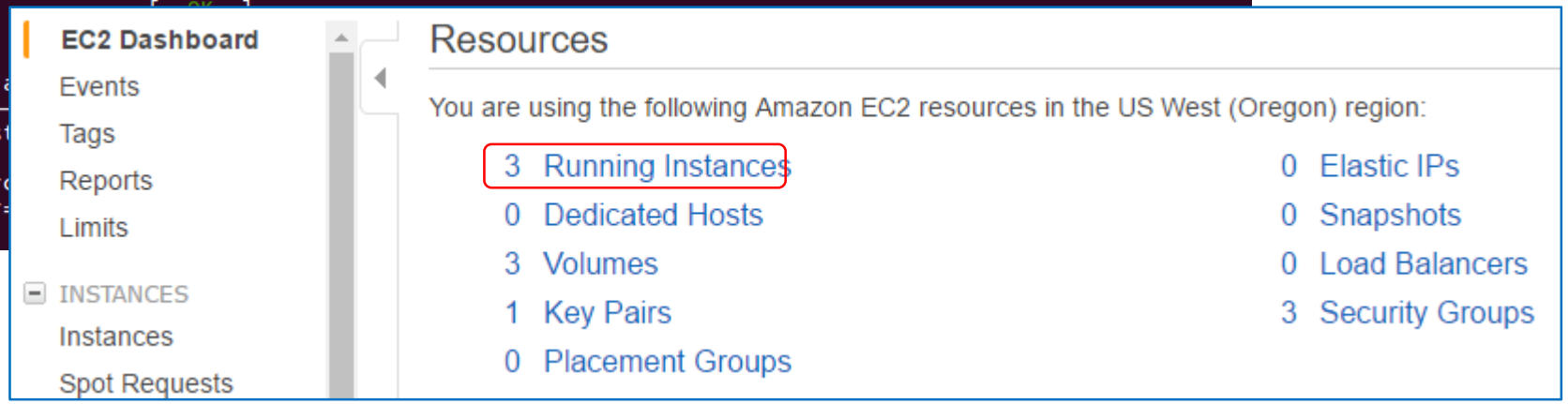
執行佈署

```
#  ./spark-ec2-branch-2.0/spark-ec2
    --key-pair=Demo                          ec2 存取金鑰名稱
    --identity-file=/home/user/Demo.pem      ec2 存取金鑰檔案位置
    --region=us-west-2                       AWS服務提供區域 (全球定價不同)
    --zone=us-west-2a                        AWS服務提供子區域
    --master-instance-type=t2.micro          Master 節點等級
    --instance-type=t2.micro                 Slave 節點等級
    --slaves=2                               Slave(Worker)節點數量
    launch test                              創建Cluster “test”(名稱自取，用完服務後要刪除test)
```

執行指令時
勿斷行

Spark-ec2

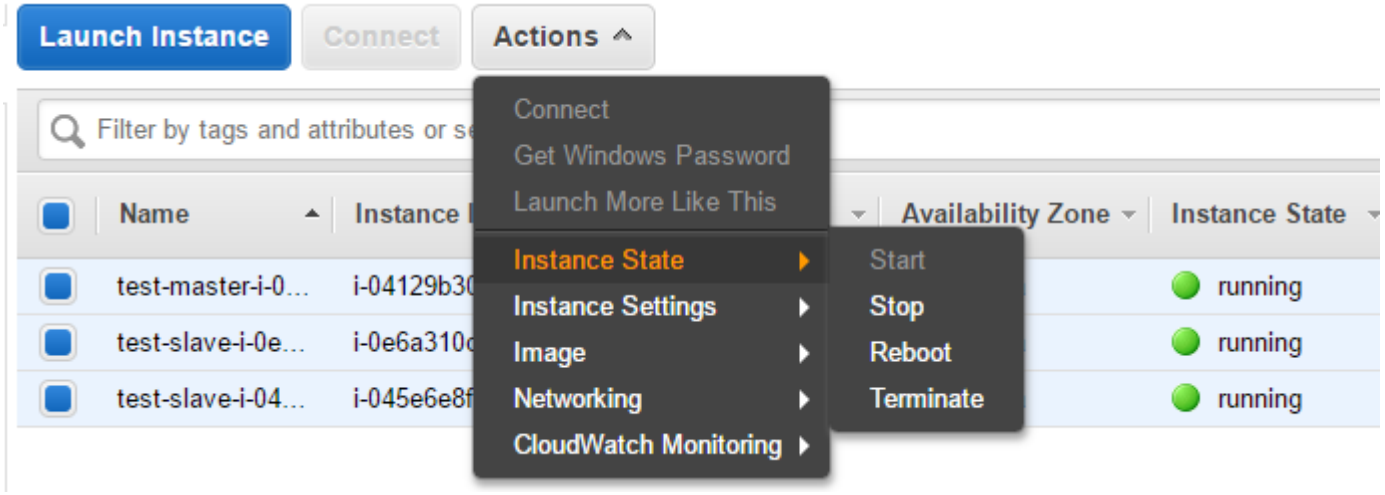
```
starting org.apache.spark.deploy.master.Master, logging to /root/spark/logs/spark-root-org.apache.spark.deploy.master.Master-1-ip-172-31-40-71
.us-west-2.compute.internal.out
ec2-                .us-west-2.compute.amazonaws.com: org.apache.spark.deploy.worker.Worker running as process 5512.  Stop it first.
ec2-                .us-west-2.compute.amazonaws.com: org.apache.spark.deploy.worker.Worker running as process 5515.  Stop it first.
[timing] spark-standalone setup:  00h 00m 27s
Setting up rstudio
spark-ec2/setup.sh: line 110: ./rstudio/setup.sh: No such file or directory
[timing] rstudio setup:  00h 00m 00s
Setting up ganglia
RSYNC'ing /etc/ganglia to slaves...
ec2-                .us-west-2.compute.amazonaws.com
ec2-                us-west-2.compute.amazonaws.com
Shutting down GANGLIA gmond:                               [  OK  ]
Starting GANGLIA gmond:                                    [  OK  ]
Shutting down GANGLIA gmond:                               [FAILED]
Starting GANGLIA gmond:                                    [  OK  ]
Connection to ec2-                .us-west-2.compute.amazonaws.com closed.
Shutting down GANGLIA gmond:                               [FAILED]
Starting GANGLIA gmond:                                    [  OK  ]
Connection to ec2-                .us-west-2.compute.amazonaws.com closed.
ln: creating symbolic link `/var/lib/ganglia/conf/default.json': File exists
Shutting down GANGLIA gmetad:                              [  OK  ]
Starting GANGLIA gmetad:                                   [  OK  ]
Stopping httpd:
Starting httpd:
[timing] ganglia setup:  00h 00m 02s
Connection to ec2-                .us-west-2.compute.a
Spark standalone cluster started at http://ec2-
Ganglia started at http://ec2-                .us-west
Done!
./spark-ec2-branch-2.0/spark_ec2.py:1564: Resourc
REAM, proto=6, laddr=('10.0.2.15', 56912), raddr=
  real_main()
```



Spark-ec2

用完服務，記得將ec2節點刪除，服務是以小時計費

方法1: Web - ec2 Dashboard，Action ➔ Instance State ➔ Terminate



方法2: CMD

./spark-ec2-branch-2.0/spark-ec2 destroy test(Cluster名稱)

Spark-ec2

用完服務，記得將ec2節點刪除，服務是以小時計費

Launch Instance | Connect | Actions

Filter by tags and attributes or search by keyword

Name	Instance ID	Instance Type	Availability Zone	Instance State
test-master-i-0...	i-04129b306a8503033	t2.micro	us-west-2a	terminated
test-slave-i-0e...	i-0e6a310dfb2152bbd	t2.micro	us-west-2a	terminated
test-slave-i-04...	i-045e6e8fbc074c9a8	t2.micro	us-west-2a	terminated

左圖畫面
約略在**terminate 10~20**分鐘後
才會淨空

Resources

You are using the following Amazon EC2 resources in the US West (Oregon) region:

- 0 Running Instances
- 0 Dedicated Hosts
- 0 Volumes
- 1 Key Pairs
- 0 Placement Groups
- 0 Elastic IPs
- 0 Snapshots
- 0 Load Balancers
- 3 Security Groups

刪除Cluster成功

雲端運算範例

Gradient-Boosted Trees (GBTs) Classification

https://spark.apache.org/docs/2.1.0/mllib-ensembles.html

Code : https://github.com/apache/spark/blob/master/examples/src/main/python/mllib/gradient_boosting_classification_example.py

Data : https://github.com/apache/spark/blob/master/data/mllib/sample_libsvm_data.txt

Step 1 . 將程式檔上傳至Master主機 root帳號之家目錄

```
# scp -i /home/user/Demo.pem  gradient_boosting_classification_example.py
  root@ec2-12-34-56-83.us-west-2.compute.amazonaws.com:/root/.
```

Step 2 . 連線至Master主機 準備執行運算工作

```
# ssh -i /home/user/Demo.pem  root@ec2-12-34-56-83.us-west-2.compute.amazonaws.com
```

```
root@ip-172-31-40-71 ~]$ ls
ephemeral-hdfs                          hadoop-2.4.0.tar.gz.1  mapreduce        scala  spark-ec2        tachyon
gradient_boosting_classification_example.py  hadoop-native          persistent-hdfs  spark  spark-warehouse
```

雲端運算範例

Step 3．在master中啟動Cluster運算工作 (前面+ time，可以得知執行時間)

如果要存取S3檔案，一定要在master先export Access key

```
#  export AWS_ACCESS_KEY_ID=ABCDEFGHIJKLMNOPQRST
#  export AWS_SECRET_ACCESS_KEY=rANDomrANDomrANDomrANDomrANDom
# time ~/spark/bin/spark-submit
  --master spark://ec2-12-34-56-83.us-west-2.compute.amazonaws.com:7077
  gradient_boosng_classification_example.py
```

```
17/05/23 12:14:06 INFO DAGScheduler: ResultStage 51 (count at /root/gradient_boosting_classification_example.py:48)
17/05/23 12:14:06 INFO DAGScheduler: Job 32 finished: count at /root/gradient_boosting_classification_example.py:48
Test Error = 0.0333333333333
Learned classification GBT model:
TreeEnsembleModel classifier with 5 trees

  Tree 0:
    If (feature 406 <= 20.0)
     If (feature 100 <= 165.0)
      Predict: -1.0
     Else (feature 100 > 165.0)
      Predict: 1.0
    Else (feature 406 > 20.0)
     Predict: 1.0
  Tree 1:
    If (feature 434 <= 0.0)
     If (feature 568 <= 253.0)
```

雲端運算範例

補充 : 讀取S3檔案寫法 (記得先export Access key)

```
#  export AWS_ACCESS_KEY_ID=ABCDEFGHIJKLMNOPQRST
#  export AWS_SECRET_ACCESS_KEY=rANDomrANDomrANDomrANDomrANDom
```

Test.py

data = sparkContext.textFile("s3n://demo17may/sample_libsvm_data.txt")

如果檔案很多，也可以使用萬用字元

data = sparkContext.textFile("s3n://demo17may/*.txt")

運算結果也可以存回S3上…

雲端運算範例

補充：運算結果儲存

程式中任何儲存動作，都會將檔案存放到Cluster HDFS中

Ex. model.save(sc, "target/tmp/myGradientBoostingClassificationModel")

➔ hdfs://ec2-12-34-56-83.us-west-2.compute.amazonaws.com:9000/user/root/target/tmp/myGra…

Ex. results.saveAsTextFile(“Results")

➔hdfs://ec2-12-34-56-83.us-west-2.compute.amazonaws.com:9000/user/root/Results

從Cluster 之HDFS取出檔案到master的電腦上

在master的環境下… (/user/root/是hdfs的預設家目錄，可打可不打)

~/ephemeral-hdfs/bin/hadoop fs -get /user/root/target

或 # ~/ephemeral-hdfs/bin/hadoop fs -get target

~/ephemeral-hdfs/bin/hadoop fs -get /user/root/Results

ls ~ (就會看到檔案存至master的家目錄)

確定檔案存到家目錄後，就將HDFS中的結果刪除，避免重新執行運算時，遇到覆寫權限問題

~/ephemeral-hdfs/bin/hadoop fs -rm -r /user/root/target

左邊指令都在master上執行

雲端運算範例

將Master上的結果，存回自己的本機(桌電/筆電)

```
# sftp -i /root/Demo.pem  root@ec2-12-34-56-83.us-west-2.compute.amazonaws.com
```

左邊指令在本機(筆電)上執行

```
sftp> ls   (確定一下有看到檔案或資料夾)
sftp> mget -r  target   (抓整個資料夾)
sftp> mget  part-0000*
sftp> exit
```

此時已連線到master上了

如果是存預算結果，通常Spark會用part-00000 part-00002 part-00003 … (文字檔)來儲存
用 # more part-00000 就可以看到運算結果

雲端運算範例
Port 8080

Master … http://ec2-12-34-56-83.us-west-2.compute.amazonaws.com:8080 Master 資源管理畫面

Spark Master at spark://ip-172-31-40-71.us-west-2.compute.internal:7077

URL: spark://ip-172-31-40-71.us-west-2.compute.internal:7077
REST URL: spark://ip-172-31-40-71.us-west-2.compute.internal:6066 *(cluster mode)*
Alive Workers: 2
Cores in use: 2 Total, 0 Used
Memory in use: 2.0 GB Total, 0.0 B Used
Applications: 0 Running, 8 Completed
Drivers: 0 Running, 0 Completed
Status: ALIVE

Workers

Worker Id	Address	State	Cores	Memory
worker-20170523110240-172.31.37.143-42336	172.31.37.143:42336	ALIVE	1 (0 Used)	1024.0 MB (0.0 B Used)
worker-20170523110240-172.31.46.122-55131	172.31.46.122:55131	ALIVE	1 (0 Used)	1024.0 MB (0.0 B Used)

Running Applications

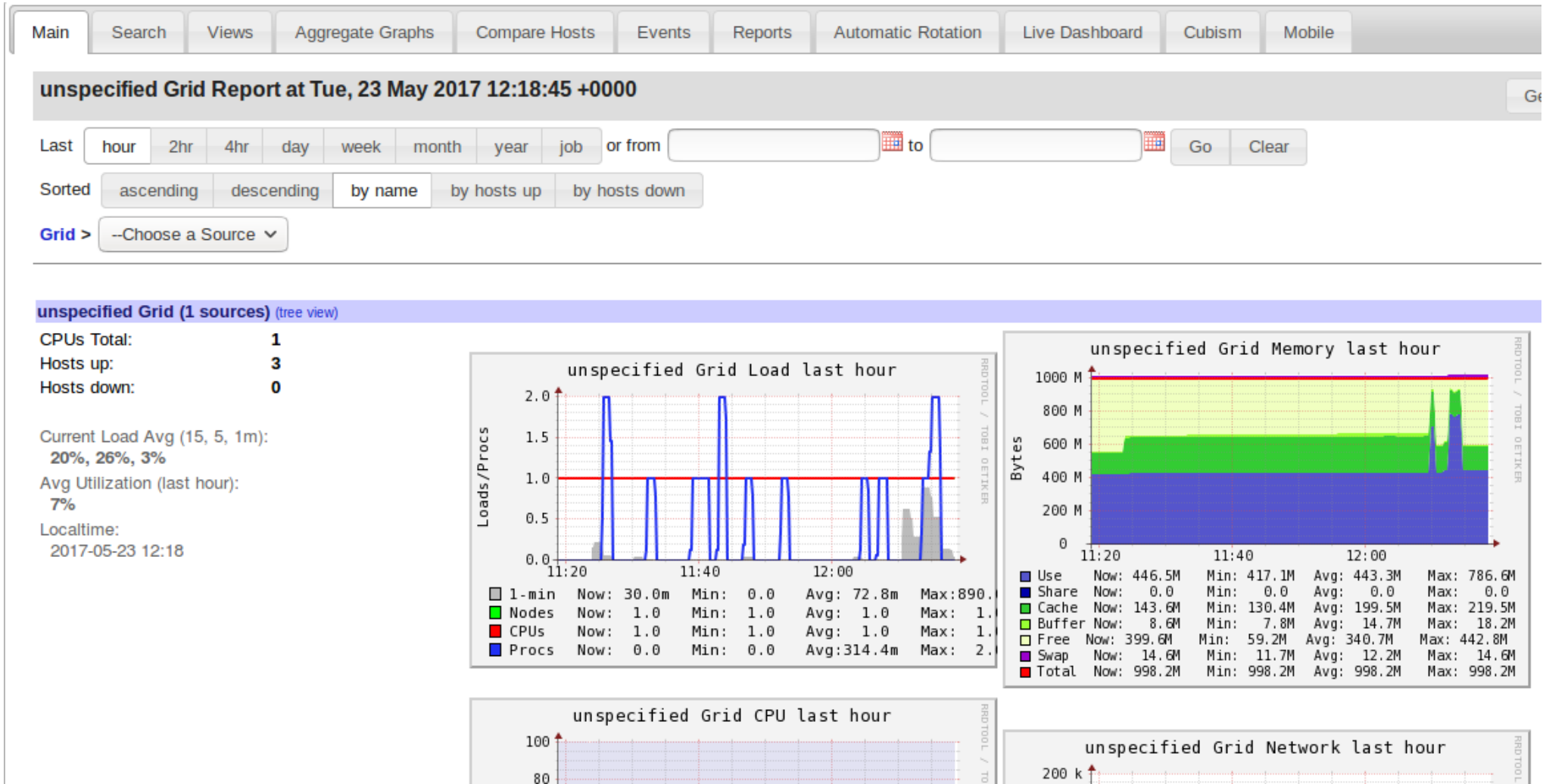
Application ID	Name	Cores	Memory per Node	Submitted Time	User	State	Durati

Completed Applications

Application ID	Name	Cores	Memory per Node	Submitted Time	User	State
app-20170523121347-0007	PythonGradientBoostedTreesClassificationExample	2	512.0 MB	2017/05/23 12:13:47	root	FINISHED
app-20170523121320-0006	PythonGradientBoostedTreesClassificationExample	2	512.0 MB	2017/05/23 12:13:20	root	FINISHED

雲端運算範例
Port 5080

Master … http://ec2-12-34-56-83.us-west-2.compute.amazonaws.com:5080/ganglia/ Cluster硬體資源監控



雲端運算範例
Port 4040

Master … http://ec2-12-34-56-83.us-west-2.compute.amazonaws.com:4040/jobs/ 運算階段，運算執行完後就無法連結

Spark 2.0.0 | Jobs | Stages | Storage | Environment | Executors | PythonGradientBoostedTreesClassi... application UI

Spark Jobs (?)

User: root
Total Uptime: 16 s
Scheduling Mode: FIFO
Completed Jobs: 9

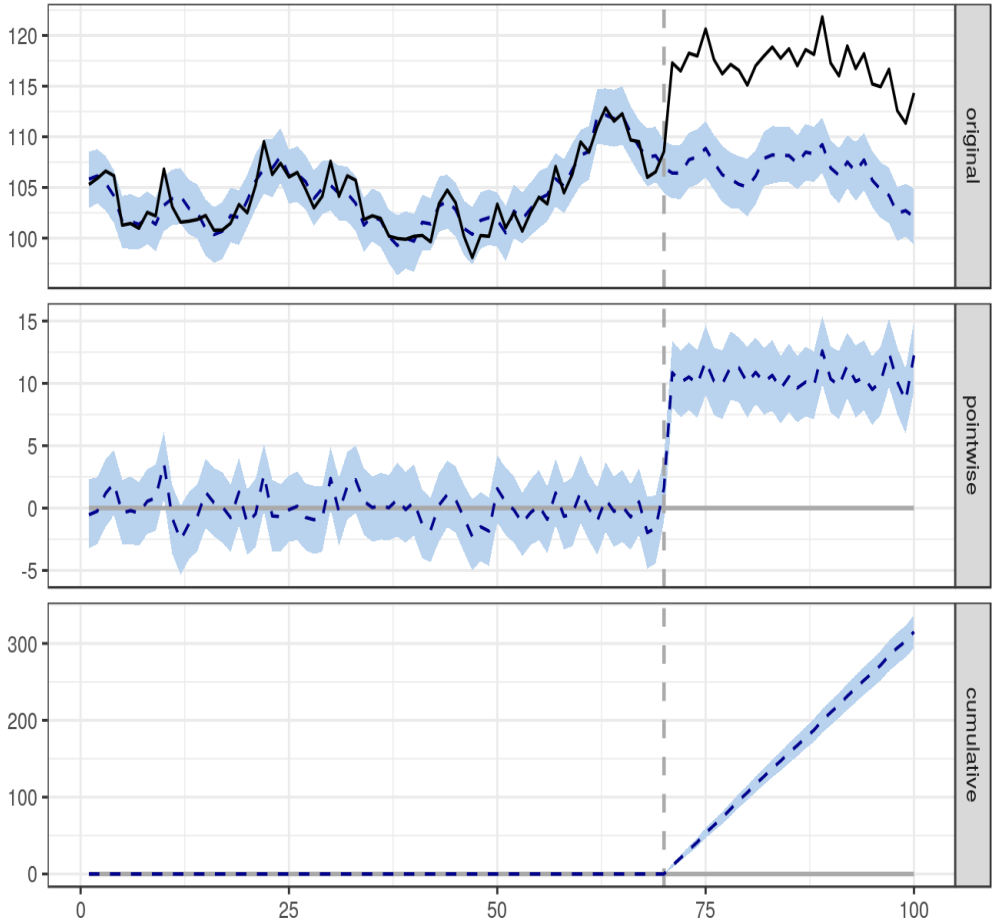
▸ Event Timeline

Completed Jobs (9)

Job Id	Description	Submitted	Duration	Stages: Succeeded/Total	Tasks (for all stages): Succeeded/Total
8	count at DecisionTreeMetadata.scala:116	2017/05/23 12:14:00	0.1 s	1/1	2/2
7	take at DecisionTreeMetadata.scala:112	2017/05/23 12:13:59	0.3 s	1/1	1/1
6	collectAsMap at RandomForest.scala:550	2017/05/23 12:13:59	0.3 s	2/2	4/4
5	collectAsMap at RandomForest.scala:550	2017/05/23 12:13:58	1 s	2/2	4/4
4	collectAsMap at RandomForest.scala:894	2017/05/23 12:13:55	3 s	2/2	4/4
3	count at DecisionTreeMetadata.scala:116	2017/05/23 12:13:54	0.2 s	1/1	2/2
2	take at DecisionTreeMetadata.scala:112	2017/05/23 12:13:54	0.4 s	1/1	1/1
1	runJob at PythonRDD.scala:441	2017/05/23 12:13:53	95 ms	1/1	1/1
0	loadLibSVMFile at /root/gradient_boosting_classification_example.py:34	2017/05/23 12:13:50	4 s	1/1	2/2

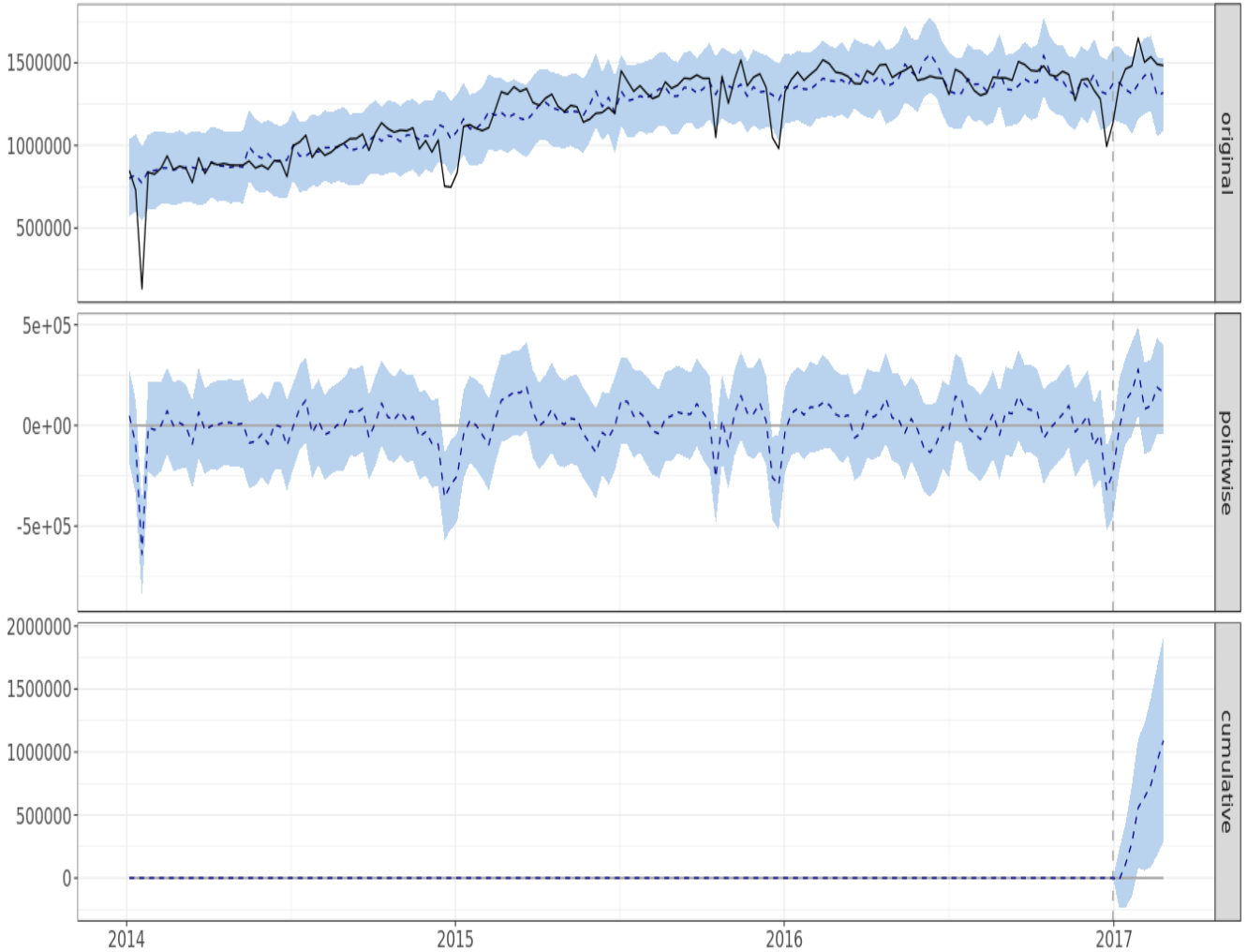
CausalImpact R

https://google.github.io/CausalImpact/CausalImpact.html



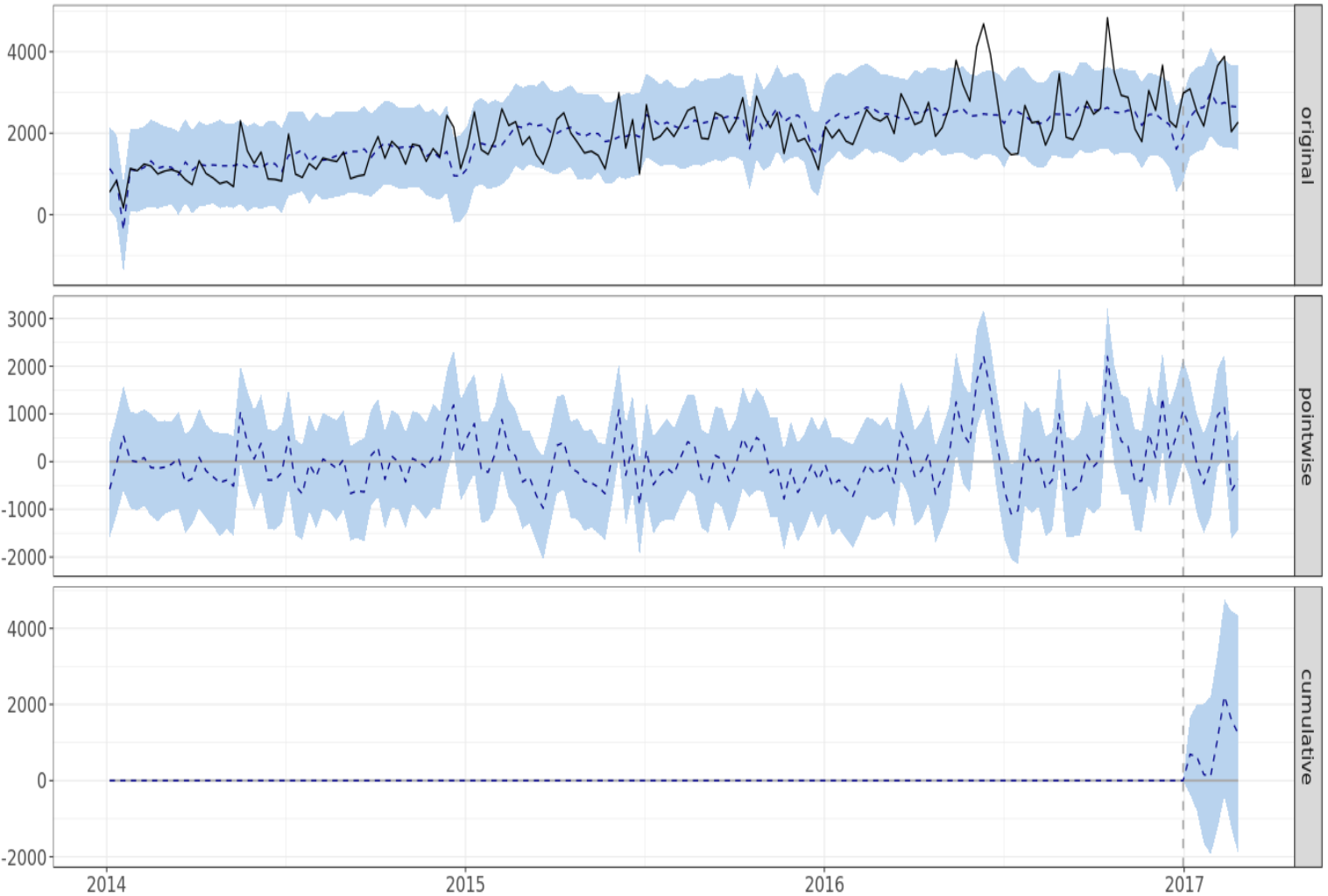
	A	B	C	D	E	F
1		Y	X1	X2	X3	X4
2	2014/1/5	58	33	37	7	52
3	2014/1/12	64	39	38	7	52
4	2014/1/19	60	36	38	9	50
5	2014/1/26	63	33	39	8	51
6	2014/2/2	64	36	39	9	52
7	2014/2/9	65	32	38	8	53
8	2014/2/16	66	36	38	7	54
9	2014/2/23	72	39	41	7	53
10	2014/3/2	78	36	44	9	51
11	2014/3/9	72	37	39	10	52
12	2014/3/16	70	34	40	8	53
13	2014/3/23	68	38	39	8	51
14	2014/3/30	68	34	38	7	53

CausalImpact R




```
> summary(impact)
Posterior inference {CausalImpact}

                          Average            Cumulative
Actual                    1.5e+06            1.2e+07
Prediction (s.d.)         1.4e+06 (52049)    1.1e+07 (416394)
95% CI                    [1.3e+06, 1.5e+06] [1.0e+07, 1.2e+07]

Absolute effect (s.d.)    136514 (52049)     1092116 (416394)
95% CI                    [38579, 245033]    [308633, 1960267]

Relative effect (s.d.)    10% (3.8%)         10% (3.8%)
95% CI                    [2.8%, 18%]        [2.8%, 18%]

Posterior tail-area probability p:   0.00512
Posterior prob. of a causal effect:  99.48823%
```

CausalImpact R

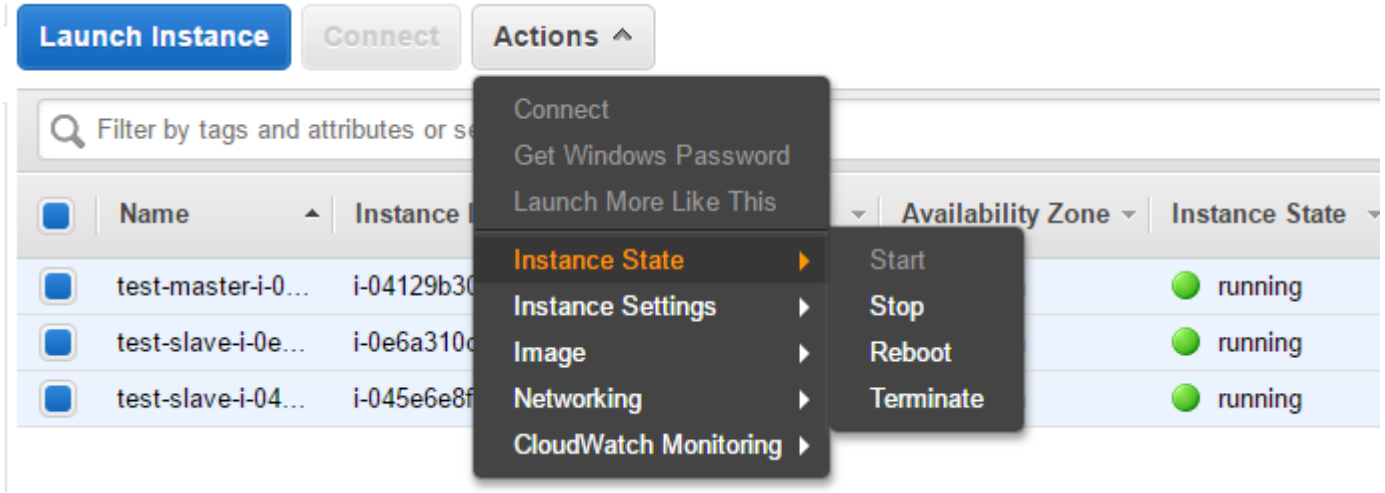



```
> summary(impact)
Posterior inference {CausalImpact}

                          Average        Cumulative
Actual                    2826           22608
Prediction (s.d.)         2672 (201)     21375 (1611)
95% CI                    [2283, 3067]   [18263, 24534]

Absolute effect (s.d.)    154 (201)      1233 (1611)
95% CI                    [-241, 543]    [-1926, 4345]

Relative effect (s.d.)    5.8% (7.5%)    5.8% (7.5%)
95% CI                    [-9%, 20%]     [-9%, 20%]

Posterior tail-area probability p:   0.224
Posterior prob. of a causal effect:  78%
```

再次提醒...

用完服務，記得將ec2節點刪除，服務是以小時計費

方法1: Web - ec2 Dashboard，Action ➔ Instance State ➔ Terminate



方法2: CMD

./spark-ec2-branch-2.0/spark-ec2 destroy test(Cluster名稱)

Q & A