

DiCloud Deliverable 2f – May 15, 2010

Amazon CloudWatch to monitor cloud resource usage

Michael Zink, Prashant Shenoy, David Irwin and Emmanuel Cecchet
{zink, shenoy, irwin, cecchet}@cs.umass.edu

University of Massachusetts, Amherst
140 Governors Drive
Amherst, MA 01003-9264

Amazon offers the CloudWatch service to monitor EC2 instances. As shown on Figure 1, CloudWatch captures raw data from Amazon Web Services (AWS) and convert them into metrics. This service was primarily designed for use with Amazon Elastic Load Balancing and Auto Scaling that looks at cpu usage across instances and automatically create new instances when the load increases to prevent overload situations.

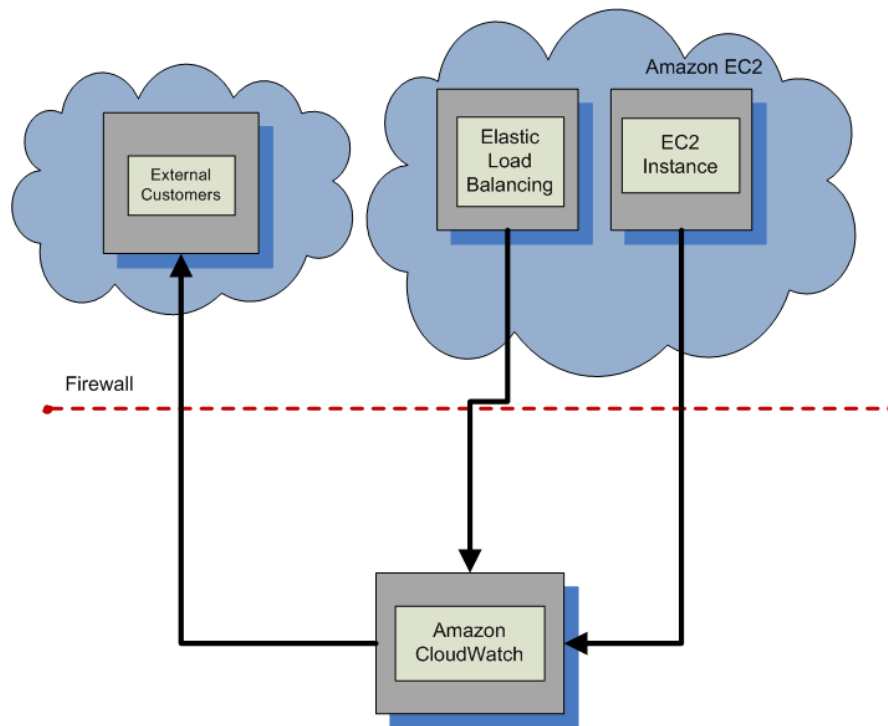


Figure 1. Amazon CloudWatch overview¹

In this report, we investigate the use of Amazon CloudWatch to track resource usage in EC2 instances and monitor the cost involved in using these resources so that a user does not use more than what has been allocated.

¹ Picture from <http://docs.amazonwebservices.com/AmazonCloudWatch/latest/DeveloperGuide/index.html>

The metrics that can be reported by CloudWatch depends on the instance that is deployed. The minimum set of metrics that can be obtained is described below:

```
manu@ubuntu-vm:~/ec2/CloudWatch-1.0.2.3/bin$ ./mon-list-metrics
DiskWriteOps    AWS/EC2
NetworkIn       AWS/EC2
NetworkIn       AWS/EC2    { InstanceId=i-e9d38c82 }
NetworkOut      AWS/EC2    { InstanceType=m1.small }
NetworkOut      AWS/EC2
NetworkOut      AWS/EC2    { ImageId=ami-0022c769 }
```

Independently of the platform that is used (Linux or Windows), these metrics can be reported by CloudWatch. This has the advantage of not requiring any specific monitoring tool to be installed and maintained in the deployed images and prevent the user from tampering with the monitoring system.

The monitoring information can also be visualized in the AWS management console in a graphical form as shown on Figure 2.

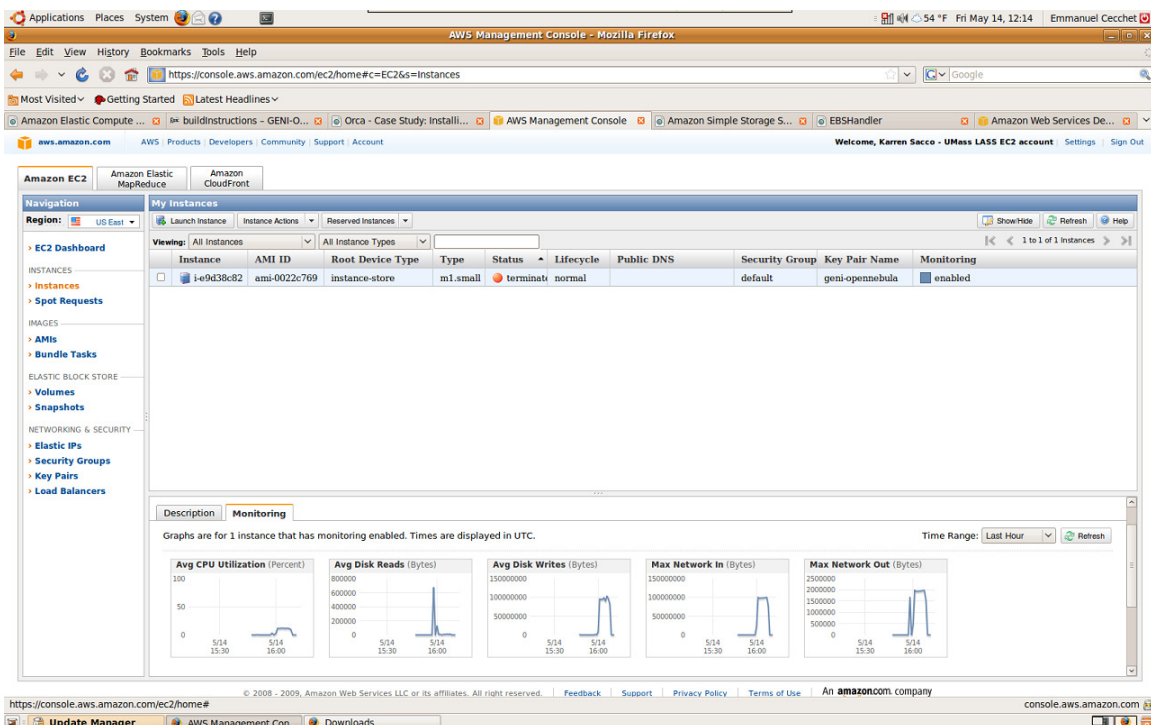


Figure 2. CloudWatch monitoring in the AWS management console

In the following sections, we investigate how we can measure the cost of running an EC2 instance which consist of: the running time of the instance, its network usage and its disk usage (for EBS). Note that S3 usage is only through a put/get interface that can be tracked by the proxy.

1. Monitoring EC2 instance running time

Amazon EC2 instances are billed by the hour and the running time of an instance can easily be obtained by simply querying the status of the instance. This does not require CloudWatch to be enabled on the instance.

The following code snippet shows the status of an instance and its start time (in bold):

```
manu@ubuntu-vm:~/ec2/CloudWatch-1.0.2.3/bin$ ec2-describe-instances
INSTANCE      i-e9d38c82      ami-0022c769    ec2-184-73-74-196.compute-1.amazonaws.com
               domU-12-31-39-05-1D-24.compute-1.internal    running geni-opennebula      0
               ml.small      2010-05-14T15:46:05+0000    us-east-1a
               monitoring-enabled    184.73.74.196    10.241.30.210      instance-store
```

The running time is simply obtained by calculating the difference between the starting time and the current time. Each started hour is billed fully and prices vary according to the EC2 region it is run in.

2. Monitoring EC2 Network usage

CloudWatch comes with command line tools that can report usage statistics. The examples below show the network input and output traffic during an scp session between a machine at UMass Amherst and EC2 small instance.

```
manu@ubuntu-vm:~/ec2/CloudWatch-1.0.2.3/bin$ mon-get-stats NetworkIn --namespace "AWS/EC2" --
statistics "Average,Sum,Maximum,Minimum" --headers
Time          Samples  Average      Sum           Minimum      Maximum      Unit
2010-05-14 15:47:00  1.0          0.0           0.0           0.0           0.0          Bytes
2010-05-14 15:48:00  1.0          0.0           0.0           0.0           0.0          Bytes
2010-05-14 15:49:00  1.0          0.0           0.0           0.0           0.0          Bytes
2010-05-14 15:50:00  1.0          0.0           0.0           0.0           0.0          Bytes
2010-05-14 15:51:00  1.0          0.0           0.0           0.0           0.0          Bytes
2010-05-14 15:52:00  1.0          0.0           0.0           0.0           0.0          Bytes
2010-05-14 15:53:00  1.0          0.0           0.0           0.0           0.0          Bytes
2010-05-14 15:54:00  1.0          0.0           0.0           0.0           0.0          Bytes
2010-05-14 15:55:00  1.0          0.0           0.0           0.0           0.0          Bytes
2010-05-14 15:56:00  1.0          0.0           0.0           0.0           0.0          Bytes
2010-05-14 15:57:00  1.0          0.0           0.0           0.0           0.0          Bytes
2010-05-14 15:58:00  1.0          17210.0       17210.0       17210.0       17210.0      Bytes
2010-05-14 15:59:00  1.0          5562.0        5562.0        5562.0        5562.0      Bytes
2010-05-14 16:00:00  1.0          2.3101877E7  2.3101877E7  2.3101877E7  2.3101877E7 Bytes
2010-05-14 16:01:00  1.0          1.00329492E8 1.00329492E8 1.00329492E8 1.00329492E8 Bytes
2010-05-14 16:02:00  1.0          9.7008116E7  9.7008116E7  9.7008116E7  9.7008116E7 Bytes
2010-05-14 16:03:00  1.0          9.8343474E7  9.8343474E7  9.8343474E7  9.8343474E7 Bytes
2010-05-14 16:04:00  1.0          9.801479E7   9.801479E7   9.801479E7   9.801479E7 Bytes
2010-05-14 16:05:00  1.0          9.8714448E7  9.8714448E7  9.8714448E7  9.8714448E7 Bytes
2010-05-14 16:06:00  1.0          9.939036E7   9.939036E7   9.939036E7   9.939036E7 Bytes
manu@ubuntu-vm:~/ec2/CloudWatch-1.0.2.3/bin$ mon-get-stats NetworkOut --namespace "AWS/EC2" --
statistics "Average,Sum,Maximum,Minimum" --headers
Time          Samples  Average      Sum           Minimum      Maximum      Unit
2010-05-14 15:47:00  1.0          0.0           0.0           0.0           0.0          Bytes
2010-05-14 15:48:00  1.0          0.0           0.0           0.0           0.0          Bytes
2010-05-14 15:49:00  1.0          0.0           0.0           0.0           0.0          Bytes
2010-05-14 15:50:00  1.0          0.0           0.0           0.0           0.0          Bytes
2010-05-14 15:51:00  1.0          0.0           0.0           0.0           0.0          Bytes
2010-05-14 15:52:00  1.0          0.0           0.0           0.0           0.0          Bytes
2010-05-14 15:53:00  1.0          0.0           0.0           0.0           0.0          Bytes
2010-05-14 15:54:00  1.0          0.0           0.0           0.0           0.0          Bytes
2010-05-14 15:55:00  1.0          0.0           0.0           0.0           0.0          Bytes
2010-05-14 15:56:00  1.0          0.0           0.0           0.0           0.0          Bytes
2010-05-14 15:57:00  1.0          0.0           0.0           0.0           0.0          Bytes
2010-05-14 15:58:00  1.0          1669268.0    1669268.0    1669268.0    1669268.0   Bytes
2010-05-14 15:59:00  1.0          4180.0       4180.0       4180.0       4180.0      Bytes
2010-05-14 16:00:00  1.0          465995.0    465995.0    465995.0    465995.0   Bytes
2010-05-14 16:01:00  1.0          1984624.0    1984624.0    1984624.0    1984624.0   Bytes
2010-05-14 16:02:00  1.0          1918948.0    1918948.0    1918948.0    1918948.0   Bytes
2010-05-14 16:03:00  1.0          1943752.0    1943752.0    1943752.0    1943752.0   Bytes
2010-05-14 16:04:00  1.0          1938924.0    1938924.0    1938924.0    1938924.0   Bytes
2010-05-14 16:05:00  1.0          1953856.0    1953856.0    1953856.0    1953856.0   Bytes
2010-05-14 16:06:00  1.0          1966600.0    1966600.0    1966600.0    1966600.0   Bytes
```

The minimum granularity of the statistics is 60 seconds. As the network bandwidth is limited to and from EC2 instances (from 2Mb/s on a small instance to about 20Mb/s on a large instance), the minute granularity should be enough to track network usage without significantly over-using the budget.

EC2 only charges for network traffic to and from the Internet but does not charge for network traffic within EC2 (between EC2 instances). CloudWatch does not make that difference and reports the aggregate network throughput. A conservative approach is to consider that all network traffic is Internet traffic and will be billed (about \$0.15/GB outbound and \$0.10/GB inbound).

For long running experiments, the exact cost can be adjusted by downloading the Amazon usage reports that are approximately updated every hour. Figure 3 shows an example of an EC2 usage report where the exact data transfer amounts are listed. Unfortunately, these values are not linked to an instance. If multiple users are to access resources and be billed separately, multiple AWS accounts should be used for accurate accounting.

A1	Service					G	H	I	J	K	L	M	N	O
323	AmazonEC2	CreateVolume	EBS:VolumeUsage	5/13/2010 21:00	5/13/2010 22:00	37580963840								
324	AmazonEC2	CreateVolume	EBS:VolumeUsage	5/13/2010 22:00	5/13/2010 23:00	26843545600								
325	AmazonEC2	CreateVolume	EBS:VolumeUsage	5/13/2010 23:00	5/14/2010 0:00	37580963840								
326	AmazonEC2	CreateVolume	EBS:VolumeUsage	5/14/2010 0:00	5/14/2010 1:00	32212254720								
327	AmazonEC2	CreateVolume	EBS:VolumeUsage	5/14/2010 1:00	5/14/2010 2:00	42949672960								
328	AmazonEC2	CreateVolume	EBS:VolumeUsage	5/14/2010 2:00	5/14/2010 3:00	48318382080								
329	AmazonEC2	CreateVolume	EBS:VolumeUsage	5/14/2010 3:00	5/14/2010 4:00	26843545600								
330	AmazonEC2	CreateVolume	EBS:VolumeUsage	5/14/2010 4:00	5/14/2010 5:00	42949672960								
331	AmazonEC2	CreateVolume	EBS:VolumeUsage	5/14/2010 5:00	5/14/2010 6:00	26843545600								
332	AmazonEC2	CreateVolume	EBS:VolumeUsage	5/14/2010 6:00	5/14/2010 7:00	53687091200								
333	AmazonEC2	CreateVolume	EBS:VolumeUsage	5/14/2010 7:00	5/14/2010 8:00	26843545600								
334	AmazonEC2	CreateVolume	EBS:VolumeUsage	5/14/2010 8:00	5/14/2010 9:00	32212254720								
335	AmazonEC2	CreateVolume	EBS:VolumeUsage	5/14/2010 9:00	5/14/2010 10:00	53687091200								
336	AmazonEC2	CreateVolume	EBS:VolumeUsage	5/14/2010 10:00	5/14/2010 11:00	37580963840								
337	AmazonEC2	CreateVolume	EBS:VolumeUsage	5/14/2010 11:00	5/14/2010 12:00	37580963840								
338	AmazonEC2	CreateVolume	EBS:VolumeUsage	5/14/2010 12:00	5/14/2010 13:00	26843545600								
339	AmazonEC2	CreateVolume	EBS:VolumeUsage	5/14/2010 13:00	5/14/2010 14:00	37580963840								
340	AmazonEC2	CreateVolume	EBS:VolumeUsage	5/14/2010 14:00	5/14/2010 15:00	42949672960								
341	AmazonEC2	CreateVolume	EBS:VolumeUsage	5/14/2010 15:00	5/14/2010 16:00	32212254720								
342	AmazonEC2	GetMetricStatistics	Calls	5/14/2010 15:00	5/14/2010 16:00	16								
343	AmazonEC2	ListMetrics	DataTransfer-Out-Bytes	5/14/2010 15:00	5/14/2010 16:00	2293								
344	AmazonEC2	RunInstances	BoxUsage	5/14/2010 15:00	5/14/2010 16:00	1								
345	AmazonEC2	GetMetricStatistics	DataTransfer-Out-Bytes	5/14/2010 15:00	5/14/2010 16:00	23094								
346	AmazonEC2	RunInstances	BoxMonitoringUsage	5/14/2010 15:00	5/14/2010 16:00	1								
347	AmazonEC2	ListMetrics	DataTransfer-In-Bytes	5/14/2010 15:00	5/14/2010 16:00	3923								
348	AmazonEC2	GetMetricStatistics	DataTransfer-In-Bytes	5/14/2010 15:00	5/14/2010 16:00	25660								
349	AmazonEC2	ListMetrics	Calls	5/14/2010 15:00	5/14/2010 16:00	1								
350	AmazonEC2	RunInstances	DataTransfer-Out-Bytes	5/14/2010 16:00	5/14/2010 17:00	17377102								
351	AmazonEC2	GetMetricStatistics	DataTransfer-Out-Bytes	5/14/2010 16:00	5/14/2010 17:00	191019								
352	AmazonEC2	RunInstances	DataTransfer-In-Bytes	5/14/2010 16:00	5/14/2010 17:00	785722182								
353	AmazonEC2	GetMetricStatistics	DataTransfer-In-Bytes	5/14/2010 16:00	5/14/2010 17:00	141128								
354	AmazonEC2	GetMetricStatistics	Calls	5/14/2010 16:00	5/14/2010 17:00	53								
355	AmazonEC2	CreateVolume	EBS:VolumeUsage	5/14/2010 16:00	5/14/2010 17:00	37580963840								
356	AmazonEC2	CreateVolume	EBS:VolumeUsage	5/14/2010 17:00	5/14/2010 18:00	48318382080								
357														

Figure 3. Example of an EC2 usage report

3. Monitoring EC2 Disk usage

Similarly to the network usage, EC2 disk usage can be monitored with CloudWatch. If the instance is not using any EBS resource, it is not necessary to monitor disk usage as using local disks is free. However, if the instance is run from an EBS volume or mounts an EBS volume, it is necessary to monitor its usage as EBS charges for IO operations (\$0.10 per 1 million I/O requests on top of the \$0.10 per GB-month of provisioned storage).

As for the network traffic where CloudWatch reports both private and public traffic, CloudWatch does not distinguish between IOs on an EBS volume or on a local disk. The aggregate IOs are reported as shown below:

```
manu@ubuntu-vm:~/ec2/CloudWatch-1.0.2.3/bin$ ./mon-get-stats DiskWriteOps --namespace "AWS/EC2" --
statistics "Average,Sum,Maximum,Minimum" --headers
Time           Samples  Average  Sum      Minimum  Maximum  Unit
2010-05-14 15:47:00  1.0      291.0    291.0    291.0    291.0    Count
2010-05-14 15:48:00  1.0      0.0      0.0      0.0      0.0      Count
2010-05-14 15:49:00  1.0      0.0      0.0      0.0      0.0      Count
2010-05-14 15:50:00  1.0      0.0      0.0      0.0      0.0      Count
2010-05-14 15:51:00  1.0      0.0      0.0      0.0      0.0      Count
2010-05-14 15:52:00  1.0      0.0      0.0      0.0      0.0      Count
2010-05-14 15:53:00  1.0      0.0      0.0      0.0      0.0      Count
2010-05-14 15:54:00  1.0      0.0      0.0      0.0      0.0      Count
2010-05-14 15:55:00  1.0      0.0      0.0      0.0      0.0      Count
2010-05-14 15:56:00  1.0      0.0      0.0      0.0      0.0      Count
2010-05-14 15:57:00  1.0      0.0      0.0      0.0      0.0      Count
2010-05-14 15:58:00  1.0      372.0    372.0    372.0    372.0    Count
2010-05-14 15:59:00  1.0      31.0     31.0     31.0     31.0     Count
2010-05-14 16:00:00  1.0      1942.0   1942.0   1942.0   1942.0   Count
2010-05-14 16:01:00  1.0      23454.0  23454.0  23454.0  23454.0  Count
2010-05-14 16:02:00  1.0      22980.0  22980.0  22980.0  22980.0  Count
2010-05-14 16:03:00  1.0      23051.0  23051.0  23051.0  23051.0  Count
2010-05-14 16:04:00  1.0      24244.0  24244.0  24244.0  24244.0  Count
2010-05-14 16:05:00  1.0      21961.0  21961.0  21961.0  21961.0  Count
2010-05-14 16:06:00  1.0      25223.0  25223.0  25223.0  25223.0  Count
```

Similarly to the approach proposed for networking, all IOs can be considered EBS IOs and the cost can be adjusted hourly by downloading the Amazon usage report. The report (see Figure 3) does not indicate which EBS volume was used which will also require different AWS accounts for multiple users requiring separate billing.

Finally, CloudWatch allows to aggregate statistics from multiple instances which is convenient to monitor at once all the resources allocated by a user.

4. Summary

We believe that Amazon CloudWatch is a viable solution for a platform independent monitoring of Amazon resource usage. The minimum granularity of the resource usage (1 minute) should not allow an unanticipated resource usage exceeding the allocated budget. Conservative policies can be used to consider all network traffic as public traffic and all disk IOs as accessing EBS volumes. The cost can be re-adjusted hourly by downloading the actual resource usage from the usage reports providing for billing by Amazon.