

4th March 2013

Testing Amazon EC2 network speed

While working with Amazon many aspects remain behind the scenes, which is a benefit for most of users, who require a working service, while having no interest in its implementation. However, this can be a problem for Amazon solution architects. Some of the internal aspects can be learned from Amazon support, yet in most cases deeper understanding requires various tests and experiments.

Think, for instance, of network performance. Does Amazon guarantee a certain bandwidth for each machine? What is the relation between network performance and server resources, region or time of day? I should mention that Amazon support strongly recommends using large machine shapes whenever network speed is critical, while the maximum speed is 1 Gb/sec. However, we would better see for ourselves.

[\[https://www.blogger.com/u/4/blogger.g?blogID=6256672819220323884\]](https://www.blogger.com/u/4/blogger.g?blogID=6256672819220323884)

1. Test Conditions and Site Preparation

The goal of this test is to find out the pure maximum network bandwidth, ideally independent from operating system and software. This is why we picked **iperf** as a testing tool and **Ubuntu 12.04** as a platform.

Naturally, running all machines, setting up required software and launching the tests manually is not an option for us.

We have had nearly every operation automated using the following tools:

1. **AMI** including a launch script that receives all necessary data from user-data, [described here](#) [\[http://epamcloud.blogspot.com/2013/02/automatic-configuration-of-virtual.html\]](http://epamcloud.blogspot.com/2013/02/automatic-configuration-of-virtual.html)
2. **Chef** executing a recipe for installation, configuration and launch of iperf
3. **Cloud Formation** launching stacks of required virtual machines as scheduled

The things left unautomated are creation of Cloud Formation templates, display of statistics and charts drawing. However, all of them can also be easily automated, in case there is a need to run these tests regularly.

Machines are launched in the server-client pairs: a pair for each shape, availability zone and region.

User-data pass Chef server address, a role for Chef client, recipe attributes and a unique tag for each machine pair:

```
chefserver=\«chefserver:4000\»;chefrole=\«iperf\»;chefattributes=\«iperf.role=client\»;tag=\«us1a-to-us1a-t1micro\»
```

Double quotes on the example above are enclosed so that the Cloud Formation template can pass validation. Without Cloud Formation there is no need to do that.

The **iperf.role** attribute contains a machine role: iperf in server mode or iperf in client mode. This tag and role combination is used to create a unique identifier for each machine:

```
tag = GetValue("#{node[:userdata]}", "#{node[:userdata]}", "tag")
node.override['iperf']['hostid'] = "#{tag}_#{node.iperf.role}"
```

The server simply launches iperf:

```

execute "iperf-server-run" do
  command "/usr/bin/iperf -s&"
  action :run
end

```

Client simply looks for a host with the same tag and server role, retrieves its public_hostname, launches testing and emails its results. All of the above is specified using attributes:

```

server = search(:node, "hostid:#{node['iperf']['tag']}_server").first[:ec2]
[:public_hostname]
Chef::Log.info("Server: #{server}")

if server.any?
  execute "iperf-client-run" do
    command "/usr/bin/iperf -t #{node.iperf.time} -c #{server} | mail #
{node['iperf']['email']} -s #{node['iperf']['region']}#{node['ipe
rf']['shape']}_#{node['iperf']['role']}"
    action :run
  end
else
  Chef::Log.info("iperf server not found, wait.")
end

```

If the client fails to find a server with the required tag, it repeats the search after a specified interval.

Example of a Cloud formation template:

```

{
  "AWSTemplateFormatVersion" : "2010-09-09",
  "Parameters" : {
    "InstanceSecurityGroup" : {
      "Description" : "Name of an existing security group",
      "Default" : "iperf",
      "Type" : "String"
    }
  },
  "Resources" : {
    "USlatoUS1aT1MicroServer" : {
      "Type" : "AWS::EC2::Instance",
      "Properties" : {
        "AvailabilityZone" : "us-east-1a",
        "KeyName" : "test",
        "SecurityGroups" : [{ "Ref" : "InstanceSecurityGroup" }],
        "ImageId" : "ami-31308c58",
        "InstanceType" : "t1.micro",
        "UserData" : { "Fn::Base64" : { "Fn::Join" : ["", [

```

```

"chefserver=\"chefserver:4000\";chefrole=\"iperf\";chefattributes=\"iperf.role=client
\";tag=\"us1a-to-us1a-t1micro\"
    ]]]}
  }
},
  "US1atoUS1aT1MicroClient" : {
    "Type" : "AWS::EC2::Instance",
    "Properties" : {
      "AvailabilityZone" : "us-east-1a",
      "KeyName" : "test",
      "SecurityGroups" : [{ "Ref" : "InstanceSecurityGroup" }],
      "ImageId" : "ami-31308c58",
      "InstanceType" : "t1.micro",
      "UserData" : { "Fn::Base64" : { "Fn::Join" : [",", [
"chefserver=\"chefserver:4000\";chefrole=\"iperf\";chefattributes=\"iperf.role=server
\";tag=\"us1a-to-us1a-t1micro\"
    ]]]}
  }
}
}
}

```

All required regions must have required AMI images and keys provided. Creation of security groups can be described right in the template.

Below is the example of Cloud Formation stack launch by cron:

```

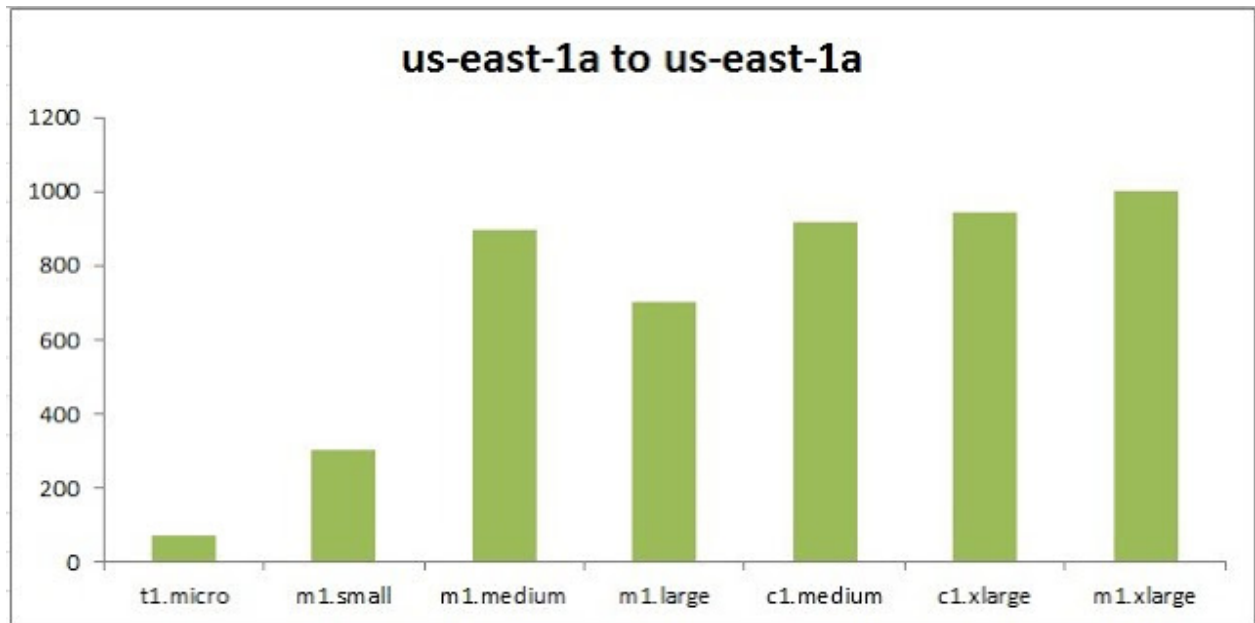
05 00 * * * cfn-create-stack --template-file=iperf_us-east-1a-to-us-east-1a.template
--stack-name iperf-us-east-1a-to-us-east-1a --region us-east-1
50 00 * * * cfn-delete-stack iperf-us-east-1a-to-us-east-1a --region us-east-1 --
force

```

Each stack stays launched for up to one hour for cost effectiveness reasons.

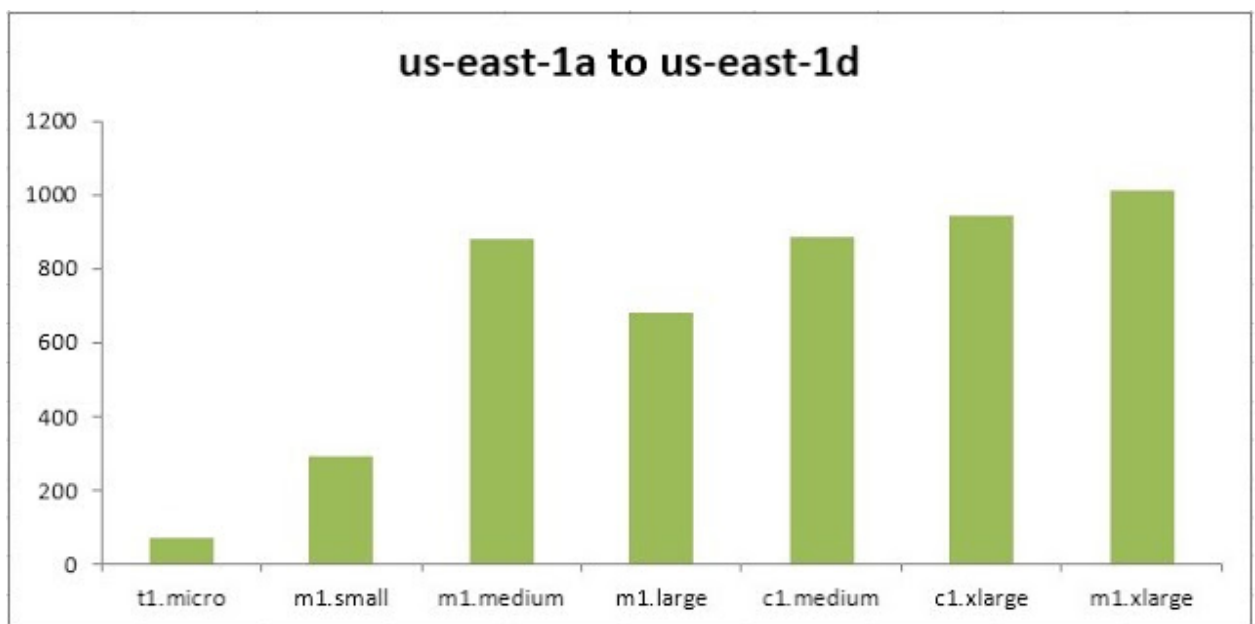
2. Test results.

Each test was run several times to avoid inadvertent distortion. We discarded individual results, markedly different from the general picture and averaged out the rest of the data.



[<http://1.bp.blogspot.com/-KfXsNuF2k3w/UTSdWDHZIMI/AAAAAAAAAHA/2sX5mMR0tQQ/s1600/1.jpg>]

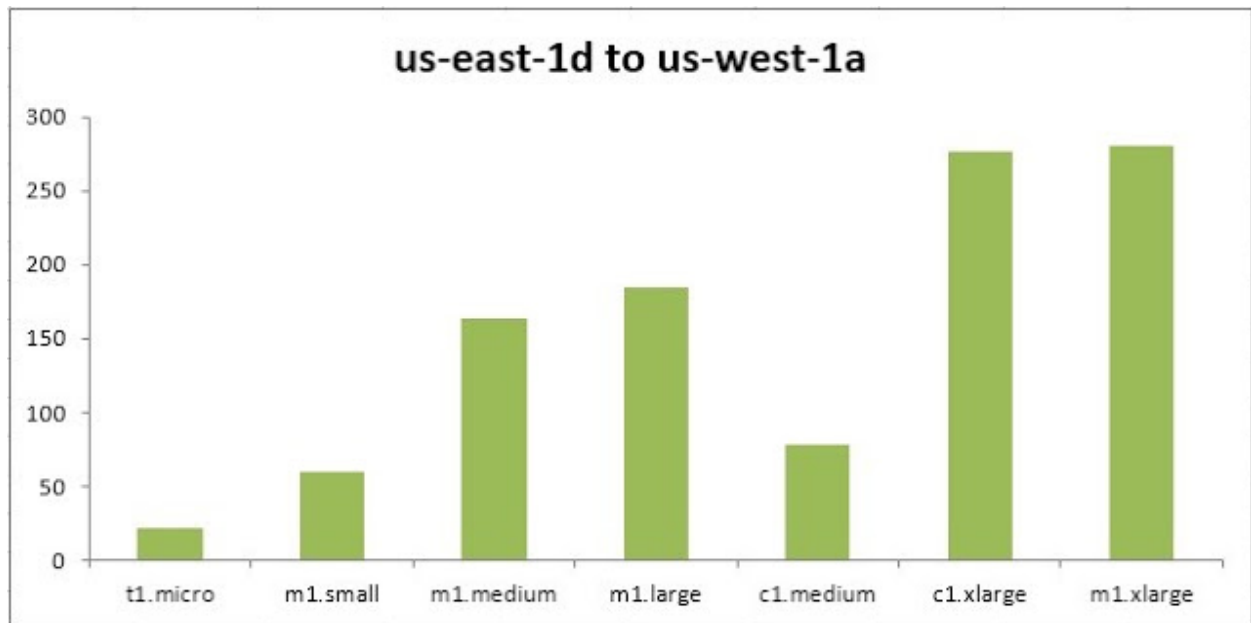
Figure 1 - Network speed within one availability zone Mb/sec



[<http://4.bp.blogspot.com/-fA3E9hUT6bU/UTSdWBgcAhl/AAAAAAAAGs/7VV6HI978Do/s1600/2.jpg>]

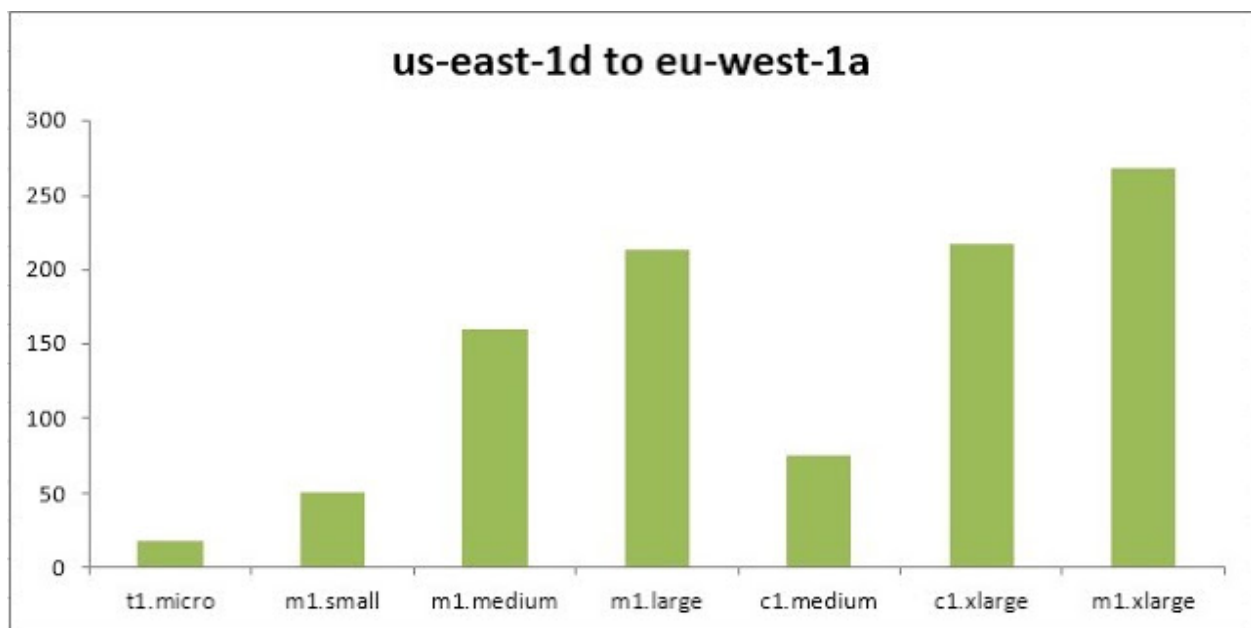
Figure 2 - Network speed for different availability zones within a single region, Mb/sec

Apparently, 'm1.medium' machines perform better compared to 'm1.large'. We can assume that instances shaped from 't1.micro' to 'm1.medium' are launched on less powerful physical servers, which is why 'm1.medium' can get all available bandwidth. At the same time, 'm1.large' are launched on more powerful yet heavily loaded servers, resulting in lower network speed.



[<http://4.bp.blogspot.com/-vulk0RGh8MA/UTSdWKJZgel/AAAAAAAAAGw/R4fPioTL26U/s1600/3.jpg>]

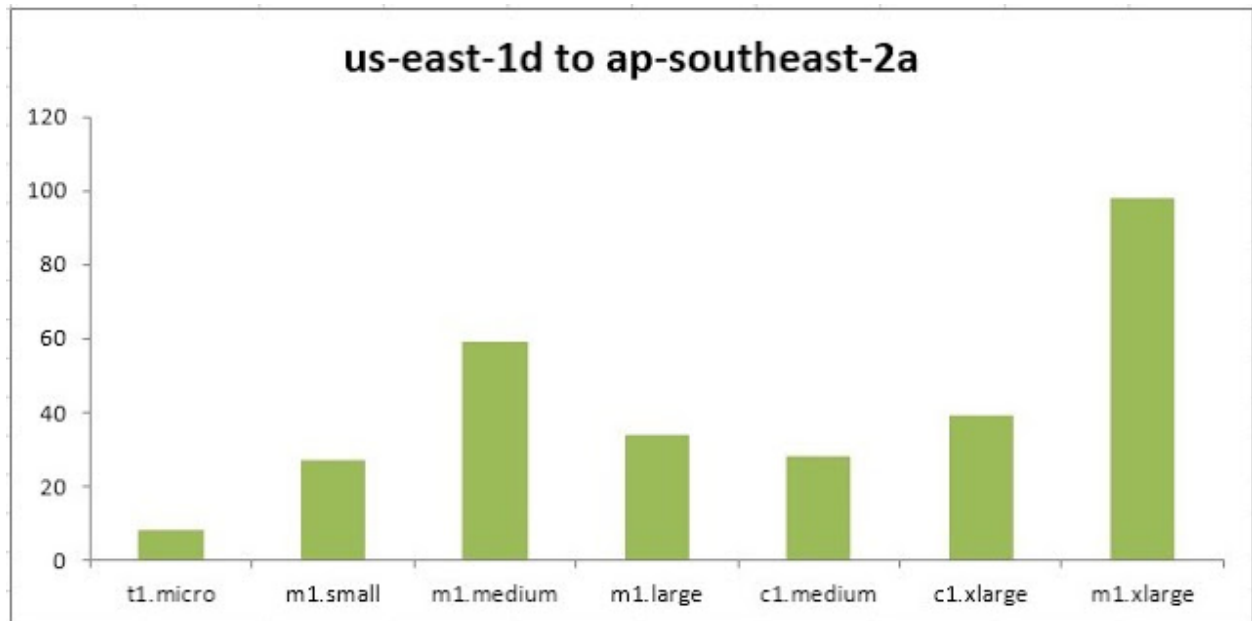
Figure 3 - Different regions within a single continent, Mb/sec



[<http://3.bp.blogspot.com/-u6sAaG1VpUc/UTSdWV8NyII/AAAAAAAAAG0/CrHIRXpJZF0/s1600/4.jpg>]

Figure 4 - Inter-regional in US-EAST-1 and EU-WEST-1, Mb/sec

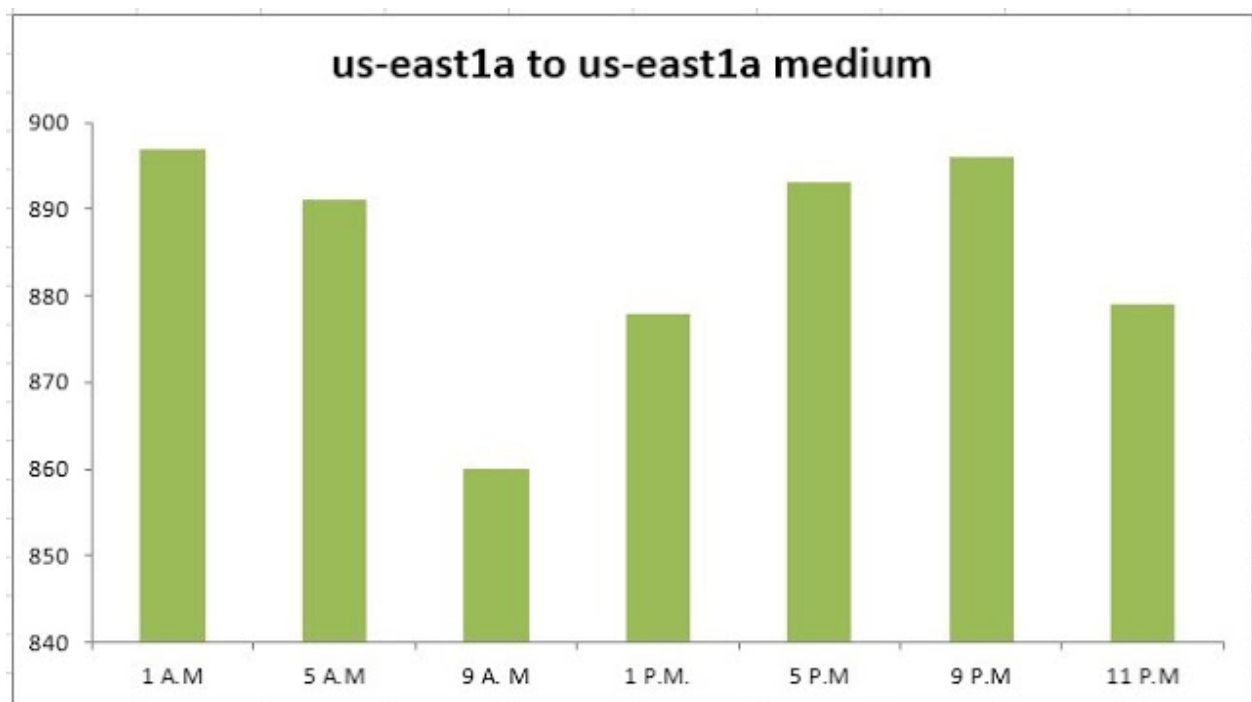
This one shows that the difference in network speed for various shapes is not the same even between regions, however machines optimized by memory (m1) display directly proportional dependency between network speed and machine shapes. We suspect Amazon forcedly limits the speed in this case on hardware level.



[http://3.bp.blogspot.com/-sIL19zOMiSY/UTSdWh-Z_TI/AAAAAAAAAG8/L93fVlrQ-PA/s1600/5.jpg]

Figure 5 - Inter-regional in US-EAST and AP-SOUTHEAST-2, Mb/sec

This test revealed major speed fluctuations diversity launch to launch. This is most likely caused by strong influence of intermediary nodes and communication channels.



[<http://1.bp.blogspot.com/-yueK-MLOboA/UTSdWpxjzSI/AAAAAAAAAG4/am6NpE6bnWQ/s1600/6.jpg>]

Figure 6 - Depending on the time of the day, for m1.medium, Mb/sec, UTC

In order to check variations in network speed depending on the time of the day we chose 'm1.medium' machines, showing fair network speed for average machine shape. Considering that the same pair of machines could show 5-10 percent fluctuations, concluded that time of the day does not have a major influence on network workload

Peculiar facts, revealed during testing:

1. Approximately 5% of cases involved at least one machine from a stack failing health check and not launching correctly
2. Nearly 5% of cases involved the whole stack failing to launch and freezing in the 'creation in progress' state. We had to delete and re-launch them manually
3. Machines, optimized by CPU speed (c1) were starting twice as long, as the rest of machines. However, when they are known to launch just as fast as the others, when not using Cloud Formation

I hope this information was useful for you.

Posted 4th March 2013 by [EPAM Cloud](#)

Labels: [Amazon](#), [aws](#), [cloud](#), [EC2](#), [infrastructure](#)

11

[View comments](#)



Smriti Rajoria [December 11, 2013 at 3:25 AM](#)

nice post
[internet speed check](#)
[Reply](#)



Anonymous [December 27, 2013 at 5:36 AM](#)

nice post [download speed check](#)
[Reply](#)



Vishnu NUK [February 8, 2014 at 8:26 PM](#)

did you try open [speed test](#) ?
[Reply](#)



Tamer [February 11, 2014 at 1:35 PM](#)

Very nice post (and i'm not a spammer)
[Reply](#)



Anonymous [September 22, 2014 at 6:41 AM](#)

Very interseting post.

I would very much like to see this test repeated for the new m3 instance types.

[Reply](#)

▼ [Replies](#)



Rad W [October 30, 2014 at 11:37 AM](#)

+1



Unknown [November 10, 2015 at 6:44 AM](#)

+1

Reply



Addison Conroy [November 20, 2014 at 3:58 AM](#)

The issues involving [Cloud Virtual Servers](#) has been a popular topic amongst scholars for many years. Advancements in Cloud Virtual Servers can be linked to many areas. Cited by many as the single most important influence on postmodern micro eco compartmentalize, several of today most brilliant minds seem incapable of recognizing its increasing relevance to understanding future generations. Crossing many cultural barriers it still draws remarks such as 'I wouldn't touch it with a barge pole' and 'I'd rather eat wasps' from the easily lead, who form the last great hope for our civilization.

[Reply](#)



Anonymous [August 19, 2016 at 12:53 PM](#)

One of the biggest challenges that organizations face today is having inaccurate data and being unresponsive to the needs of the [Amazon AWS Users Email List](#) organization.

[Reply](#)



Parana Impact [November 5, 2016 at 8:57 AM](#)

Hello,
Thank you for the Blog.[Parana Impact](#) help you reach the right target customers to advertise your products and services.
[Amazon AWS Email List](#)

[Reply](#)



Abiya Carol [June 1, 2017 at 8:57 AM](#)

A nice article here with some useful tips for those who are not used-to comment that frequently. Thanks for this helpful information I agree with all points you have given to us. I will follow all of them.

[Dot Net Training in Chennai](#)

[Software Testing Training in Chennai](#)

[Reply](#)

Enter your comment...



Comment as: [Google Account](#) ▼

Publish

Preview