IOSR Journal of Computer Engineering (IOSR-JCE)
e-ISSN: 2278-0661, p- ISSN: 2278-8727Volume 16, Issue 4, Ver. I (Jul-Aug. 2014), PP 39-46
www.iosrjournals.org

Auto-Scaling, Load Balancing and Monitoring As service in public cloud

Gopal Dhaker[#1], Dr. Savita Shiwani[#2]
[#1]*(Computer Science& Engineering/Suresh Gyan Vihar University, Jaipur 302025, India)*
[#2]*(Associate Professor, Suresh Gyan Vihar University, Jaipur,India)*

Abstract: *Cloud computing[1][2][3] word has changed the classical computing environment in IT industries. it is most emerging & popular technology in IT & research field because of its great feature such as virtualization and on-demand resource allocating (dynamic) . Now - a – days because of increased use of internet the associate resource are increasing rapidly resulting generation of high work load. To provide the reliable service to client with QOS[5] the load balancing mechanism is necessary in cloud environment, to prevent system from overloading and crash an autoscaling mechanism must also be provided according to the application and incoming user traffic. load balancing mechanism provides the distribution of load among one or more nodes of cloud system, for efficient service model autoscaling feature also enabled with the load balancer to handle the excess load. Auto scaling scaled-up and scaled down the platform dynamically according to the clients incoming traffic this save money and physical resources. Latency based routing is the new concept in cloud computing which provide the load balancing based on DNS latency to global client by mapping domain name system (DNS) [10)through the different hosted zone .provide the load balancing based on geographical service region. To achieve above mentioned we use the public cloud services such as amazons'EC2. ELB. This research is divided in four part such as: i) load balancing ii) auto scaling iii)latency based routing iv) resource monitoring . while discussing each topic in detailed we will implement the individual service and test while providing load from external software tool putty we will produce result for efficient load balancing.*

Keywords: *load balancing, autoscaling, latency based routing, resource monitoring, cloud computing.*

I. Introduction

Cloud computing g is the most emerging technology which drew the attention of all the technocrat in the field of computer science .Cloud computing is the technique which represents both cloud and the application (services).it is basically referred as to accessing computing service (resources & application) over the internet [3][4][6].

cloud service provider handles the date from the remote location about that client is unaware but an individuals can access his data from anywhere simply by a system with internet connection. cloud computing has changed the classical computing environment in IT industry with cloud computing many corporate is migrating their business from traditional computing to cloud computing in order to meet their business requirement. Cloud computing is been considered as the most revolutionary technology in the IT industry for example we can assume whole internet as a single cloud in which people share space and resource from the pool of virtual space . the most important thing provides by cloud computing is the virtualization of resources .NITS[4][7] give the standard definition of cloud computing as " it is the framework which enable the user or client in the computing environment to access on demand services and shared pool of resources such as server , network , application. For example when we save image over the internet or send some files using internet we are using cloud computing. many website is running on cloud computing because of its elasticity and auto scaling feature . cloud computing can be used as three mode and 4 deployment model, three models[9] are

Cloud provides different application in different ways and we can group them in three categories. The word service is a concept of reusability of cloud component across the service provider's network. Cloud computing services are "*as a service*" model. **Software as service (SAAS)** deployed as application on cloud and client can access it through internet. User does not need to install on local system neither pay for license of software. In this model user need not to worry about the software maintenance, its version, all these handle by cloud service provider . user get ready to use application in customized format in a web browser and can utilize high resource required for the on demand service and has to pay only for time application used. Main benefit of SAAS is the cost reduction and eliminating need of high capacity server which save up front cost . example of SAAS are

- Online graphics designing
- Video calling & conferencing
- Google docs
- Microsoft Office 365

- Lync Online

Platform as a Service (Paas) Platform as service is second service delivering model after software as service model. In this service deployment model all the required resources such as operating system, database, and programming language that is required to built and test application or services from internet . Customer is free to build application of his choice. Paas includes the designing, deployment and testing of user made application. service developed on one cloud service provider will not work on other cloud provider's network. Example of Paas is ***Google App Engine*** is Google cloud platform in which the web application is developed and hoisted .

Infrastructure as a Service (IaaS) It is a hardware as a service because of this service providing model provides hardware and network resources to user client customize operating system and develop its own software and application. this service is used by the large organization to create their own computational (software & application) model . this base layer deals with network resources, virtual machine, and the servers. Iaas is basically hardware on rent which includes service level agreement, load balancing , fault tolerance, firewall and network capabilities. User have to pay for the how much & how long resource he is using. so autoscaling is an essential part of Iaas. Example of infrastructure as service is

- Amazon web service,
- go grid, and
- 3 tera.

The four deployment model are : Cloud can be deployed and hosted in different style according to need of use their business model similar to P/I/saas, organization can choose public, private or hybrid clouds mode to deploy services. So far there was a tendency organization to use internal solution (private cloud)
to handle the highly requested data. Now the organization is using hybrid solution to deploy application on cloud. There are four categorized of deployment models in cloud environment. **Public cloud** this type of cloud is managed and owned by a third organization and the resources and services is accessed via the public internet. This is also known as external cloud. It is a standard computational model in which resources such as storage , application , and hardware, server is rendered by user on public network, and service provider bills the user on a pay –as-u-go model Example of public cloud model is Amazon's EC2, sun cloud model , Google app engine, azure service platform. **Private cloud** Private cloud are owned and managed by a single organization in the organization's infrastructure itself. This model of cloud are made only for a single enterprise . this model gives a great security to data and control over cloud resources. Only the specific client who is authorize to access can access the cloud service . it is like the past LAN network for individuals with virtualization advantage. **Hybrid cloud** it is the combination of public and private cloud modes . exact definition of hybrid cloud is the use of virtual cloud server and physical hardware together to provide a common service. It is also known as combined cloud. In this combine environment some resources is provided in-house and other is rendered externally. **Community cloud** is nothing but a private cloud which is used by a group of organization. This model is local infrastructure restricted.. Service is managed by a third party service provider

Characteristics of cloud computing [6] [2] – cloud's characteristics made it most demanding technology in IT field The major characteristics are on demand service, pay as u go , distributive service made it popular . and apart from this pooled resource, real network use and elasticity made it easier to use .

II. Auto Scaling And Load Balancing

1) Load balancing is the technique of redistributing the total load among the individual nodes of the cloud collective system to improve the response time and utilization of resources. Simultaneously avoiding condition which include some overloaded nodes while other nodes of system is free or under loaded. Thus it improve the overall performance of the cloud network, load balancing is a challenging task in cloud computing because it ensure the stability of system & user satisfaction. It includes fault tolerance, throughput, and reliability. Load balancing can be initiated by either sender our receiver. A load rebalancing is a generic expression of distributing high processing load among the different nodes

Objective of load balancing- Load balancing is very crucial in cloud system as the incoming load is unpredictable and variable also depends on different factor. A good load balancer must meet following requirements.

- Improve the operation performance significantly
- It must have fault tolerance capability and provides the backup path when system fails due to high load
- Most importantly it should always maintain the stability of the system and perform steady operations.

2) Autoscaling Elasticity is the main attribute in the context of cloud computing and scalability is the key benefits provided by cloud. In big organization the enterprise application that use cloud network need a quality of service bond between client and cloud service provider. Backup path, scalability & availability of service is necessity for deployment of any application on cloud. Autoscaling technique ensures the QOS agreement between client & service provider. One can define autoscaling as the technique used to full fill requirement of on demand resource allocation when incoming traffic is increased and remove extra free resource when application is not have peak load means scaling up & down the system.

3) Resource Monitoring Nowadays no of cloud services in cloud network is increasing and infrastructure to run the application is also rapidly increased it create complexity between different infrastructure and application deployed on it. Efficient Cloud monitoring become necessity to handle and operate this complex cloud infrastructure & applications effectively

Need of monitoring in cloud

Cloud monitoring is very important task for cloud client and provider. As it manage & control the hardware and software service provider side and provides the information about performances of platform and services at user side. Cloud monitoring is essential to maintain the service level agreement (availability, delay). Also indicate workload generated by the cloud to user and provider. Cloud monitoring allow user to implement the mechanism for the fault tolerance and recovery path .

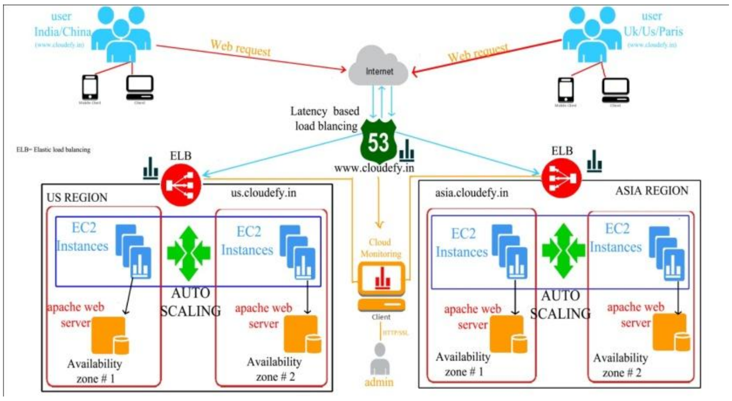
III. Problem Formulate

Cloud computing is new and evolving technology conceptually a elastic & distributed model of distributing the resource over the network (cloud computing). System component must be cooperate to handle the request from cloud computing thus intercommunication between different resource hardware and software components is required to fulfill the client request. If the no of client requests become large it will create a bottleneck problem in the system, resulting the imbalance system in which some node is overloaded and other either don’t have any work load or have only light load. this is called imbalanced system. to handle this issue we need a efficient load balancing technique. As the use of cloud computing increasing it demands the efficient resource provisioning. Lots of technique is proposed to ensure the proper load balancing but following issue has not been resolved full.

I. Data availability issue
II. Ensuring proper scalability to handle the excess load
III. Ensuring the proper authentication, auditing , authorization
IV. Ensure system stability and steady operation
V. Proper monitoring of load

IV. Proposed System

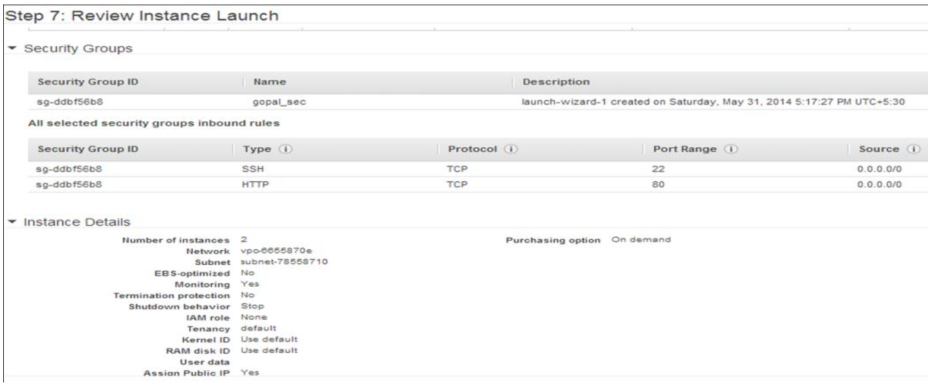
System Architecture



The system architecture is show in above picture. This proposed system aimed for public cloud in which large no of nodes are connected in distributed manner. This system divide load balancing into 3 step to provide efficient solution to above mentioned issue this can be done using public cloud services .we will use Amazon web service’s networking services to balance the load in the proposed system this services individually does not provide any beneficial solution for cloud computing but if used together we can configure a smart load balancing system. This research will present an analysis of Amazon web service’s networking services. Load balancing will be handle in three part i) latency based load balancing ii) local regional load balancing iii) autoscaling to handle excess load

V. Implementation

Step -1 creating instances – In the proposed model we have created two instances per service region in different availability zone. Cloud instances can be lanced in AWS cloud environment by using the EC2 service .we have to select the EC2 from the AWS console . after selecting the basic operating system , instance type is selected we have selected General purpose small instances, now need to security group security group is nothing but the configured protocol for which instances is configured. After reviewing the instance is launched and it is show in instances tab under EC2. We have selected the US and ASIA region for the experiment. An apche server is installed on these instances by the command "***SUDO APT-GET APACHE2"*** this will install the apache web server on instances. Following images will show the creation of instances. We can connect to our instances by using either DNS address or IP address.



Step 7: Review Instance Launch

▾ Security Groups

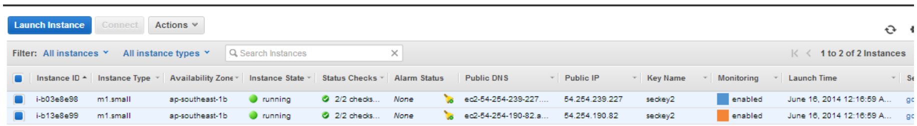
Security Group ID	Name	Description
sg-ddbf56b8	gopal_sec	launch-wizard-1 created on Saturday, May 31, 2014 5:17:27 PM UTC+5:30

All selected security groups inbound rules

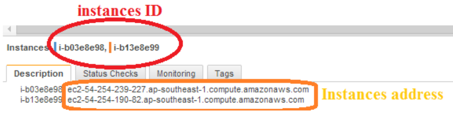
Security Group ID	Type	Protocol	Port Range	Source
sg-ddbf56b8	SSH	TCP	22	0.0.0.0/0
sg-ddbf56b8	HTTP	TCP	80	0.0.0.0/0

▾ Instance Details

Number of instances	2
Network	vpc-6655870e
Subnet	subnet-78558710
EBS-optimized	No
Monitoring	Yes
Termination protection	No
Shutdown behavior	Stop
IAM role	None
Tenancy	default
Kernel ID	Use default
RAM disk ID	Use default
User data	
Assign Public IP	Yes
Purchasing option	On demand



☐	Instance ID	Instance Type	Availability Zone	Instance State	Status Checks	Alarm Status	Public DNS	Public IP	Key Name	Monitoring	Launch Time
☐	i-b03e8e98	m1.small	ap-southeast-1b	● running	✓ 2/2 checks...	None	ec2-54-254-239-227....	54.254.239.227	seckey2	enabled	June 16, 2014 12:16:59 A...
☐	i-b13e8e99	m1.small	ap-southeast-1b	● running	✓ 2/2 checks...	None	ec2-54-254-190-82.a...	54.254.190.82	seckey2	enabled	June 16, 2014 12:16:59 A...



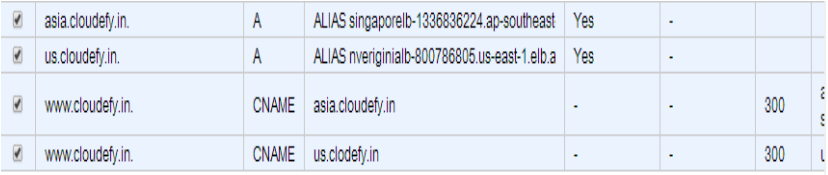
instances ID

Instances: i-b03e8e98, i-b13e8e99

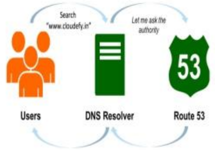
i-b03e8e98 ec2-54-254-239-227.ap-southeast-1.compute.amazonaws.com
i-b13e8e99 ec2-54-254-190-82.ap-southeast-1.compute.amazonaws.com

Instances address

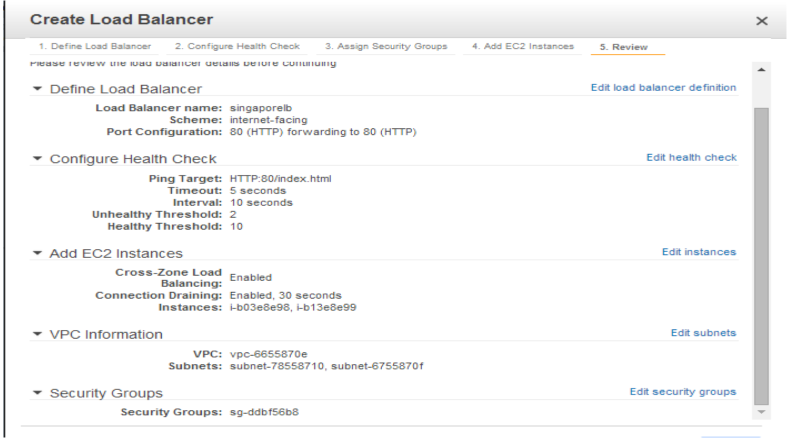
- **Step 2 – Creating main load balancer (latency based)-** main load balancer which is software load balancer based on the latency in which load is distribute among the different service region based upon the location of request this is done by DNS resolver and created hosted zone. Latency based routing choose latent region instead choosing the any random service region to forward load for the process. This man load balance will forward the traffic to the regional load balancer. in this proposed work we have created one hosted zone **"www.cloudefy.in"** and alias as **us.cloudefy.in & asia.cloudefy,in** is created as A name and CNAME for each service region . the request coming for " www.cloudefy.in" will be first resolved by DNS resolver then will be sent to the more. We can also implement weighted rule in latency based routing to improve the load handling.



☑	asia.cloudefy.in.	A	ALIAS singaporelb-1336836224.ap-southeast	Yes	-		
☑	us.cloudefy.in.	A	ALIAS nvQerginialb-800786805.us-east-1.elb.a	Yes	-		
☑	www.cloudefy.in.	CNAME	asia.cloudefy.in	-	-	300	a s
☑	www.cloudefy.in.	CNAME	us.clodefy.in	-	-	300	u



- **Step 3- creation of regional load balancer (Round robin based)-** the instances set behind this load balancer is A no of secondary load balancer can be created this load balancer balanced the load between the instances in different availability zone. These load balancer can balance load in same availability in the same service region. It cant balance load between multiple service region . load balancing is done according to Round Robin algorithm. Load balancer is create using the Elastic load balancing service, selecting the launch load balancer under this we choose the load balancer id and the region in which it is being configured. Add the instances to this instances, set the ping path for health check , in proposed work we have created two load balancer one each for US & ASIA region. We configured this load balancer for web page request. Ping path also set on HTTP protocol on instance and we use security group as SSL protocol. A load balancer is shown in fig



Create Load Balancer

1. Define Load Balancer 2. Configure Health Check 3. Assign Security Groups 4. Add EC2 Instances 5. Review

Please review the load balancer details before continuing

Define Load Balancer — Edit load balancer definition

Load Balancer name: singaporelb
Scheme: internet-facing
Port Configuration: 80 (HTTP) forwarding to 80 (HTTP)

Configure Health Check — Edit health check

Ping Target: HTTP:80/index.html
Timeout: 5 seconds
Interval: 10 seconds
Unhealthy Threshold: 2
Healthy Threshold: 10

Add EC2 Instances — Edit instances

Cross-Zone Load Balancing: Enabled
Connection Draining: Enabled, 30 seconds
Instances: i-b03e8e98, i-b13e8e99

VPC Information — Edit subnets

VPC: vpc-6655870e
Subnets: subnet-78558710, subnet-6755870f

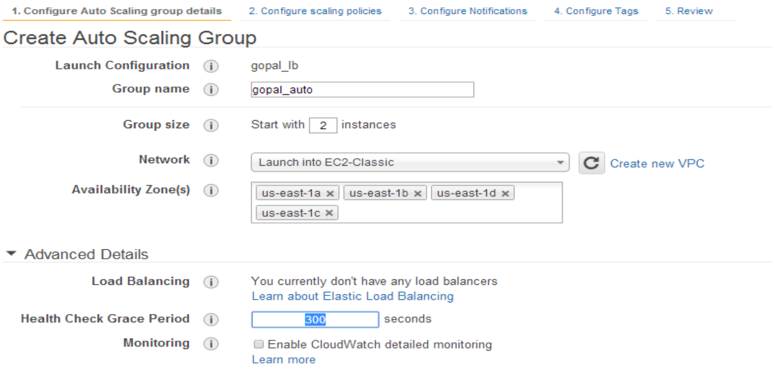
Security Groups — Edit security groups

Security Groups: sg-ddbf56b8

Step 4- creating auto scaling group[12] –Amazon’s auto scaling group creation is much similar to creating instances of ec2 . Amazon web service provides the facility to accommodate the dynamic change in the system. This is done by Auto scaling group. Autoscaling especially need for the application on which no of request is variable per hour per day. Some time peak load and some time completely free. Auto scaling groups defined as.

I) First he AMI is created for the scaling group as the new initialized instances must have same configuration as other instances in the group.

II) Now the maximum and minimum size of scaling group is defined by the client. Initial group size also defined here in the show picture we have created a scaling group named **gopal_auto** and set the initial group size to the 2. Added the availability zone .



1. Configure Auto Scaling group details 2. Configure scaling policies 3. Configure Notifications 4. Configure Tags 5. Review

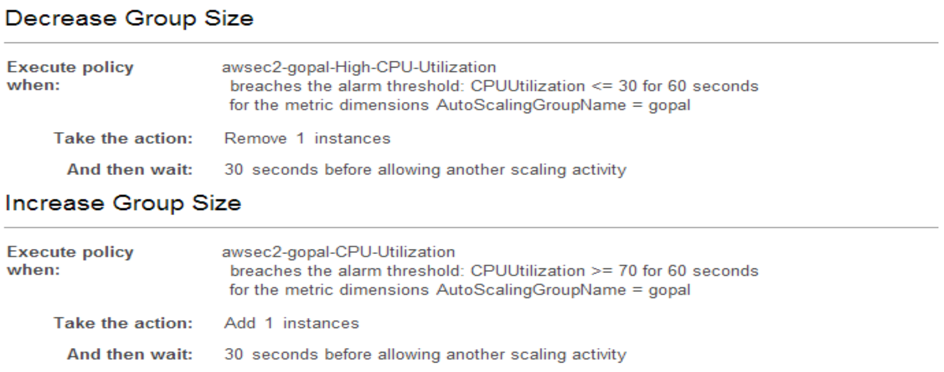
Create Auto Scaling Group

Launch Configuration: gopal_lb
Group name: gopal_auto
Group size: Start with 2 instances
Network: Launch into EC2-Classic — Create new VPC
Availability Zone(s): us-east-1a ×, us-east-1b ×, us-east-1d ×, us-east-1c ×

Advanced Details

Load Balancing: You currently don't have any load balancers
Learn about Elastic Load Balancing
Health Check Grace Period: 300 seconds
Monitoring: □ Enable CloudWatch detailed monitoring
Learn more

Creating policies for auto scaling Autoscaling properties is created on the basic of metric . these can be CPU utilization , disk read or write up, network traffic , request count. Alarm is configure for specific metric. Cloud watch observe each metric and rise alarm. and autoscaling group will perform specific action . load balancer get informed automatically to send or stop sending request to the newly created or terminated instances. In the shown picture we have created the policy for decrease and increase the group size based on CPU utilization and also defined the appropriate action .



Decrease Group Size

Execute policy when: awsec2-gopal-High-CPU-Utilization
breaches the alarm threshold: CPUUtilization <= 30 for 60 seconds
for the metric dimensions AutoScalingGroupName = gopal

Take the action: Remove 1 instances

And then wait: 30 seconds before allowing another scaling activity

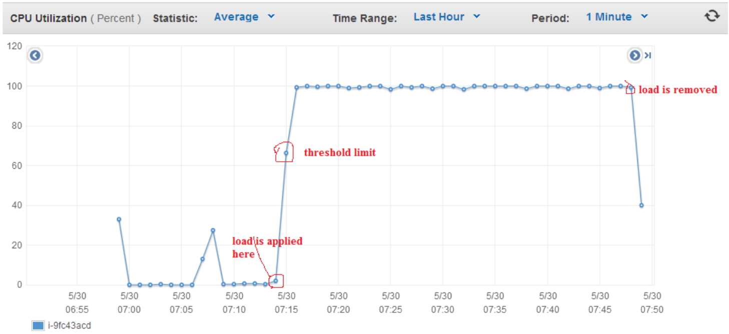
Increase Group Size

Execute policy when: awsec2-gopal-CPU-Utilization
breaches the alarm threshold: CPUUtilization >= 70 for 60 seconds
for the metric dimensions AutoScalingGroupName = gopal

Take the action: Add 1 instances

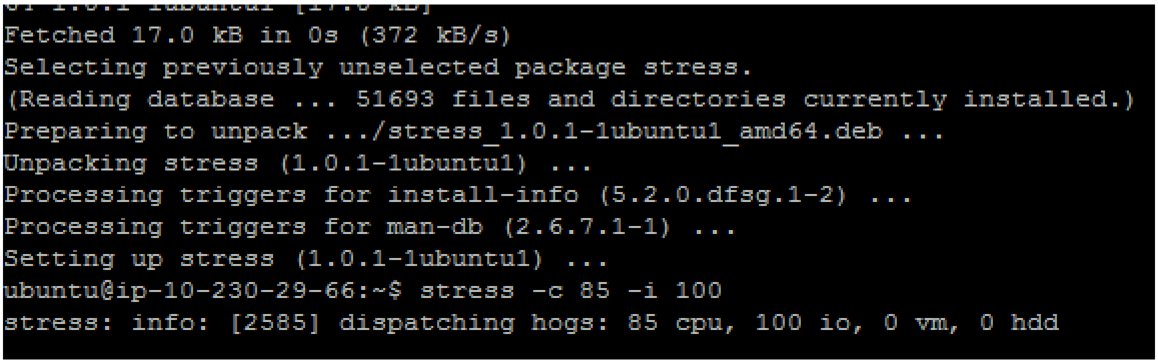
And then wait: 30 seconds before allowing another scaling activity

- **Step-5 Resource monitoring** to provide the efficient solution of load balancing issue. system must have a reliable and correct monitoring tool by which admin can monitor each and every component . Amazon provides the cloud watch exclusively for monitoring the cloud. It provides the cloud watch as monitoring tool which automatically monitor load balancer and instances for matrix for example latency and no of request . it provides the detailed information in graph form as per user selected time frame. User can select different parameter and matrix to control the system. In this proposed work we are using CPU utilization watch, memory watch, network pattern .it also provide a comparison between different graphs. instances health monitoring is done automatically in cloud watch. best part of this service is notified the admin for specific condition and able to execute user defined actions . the following picture is monitoring graph for an scaling group instance which is showing the maximum CPU utilization for instance i-9fc43acd. If we do detailed study of this graph them we will get to know that initial when load is not applied (part IV) maximum CPU utilization was become zero as the load is applied on the instances is gone over loaded . and autoscaling action is performed (part IV)



VI. Testing And Load Generation

Load generation – virtual load will be generation though putties the terminal emulation software on the different instances from the different PC in the lab. To generate the load in putty we must install stress component in apache server . this can be done by following command “***Sudo apt-get install stress”*** and the required load is generated by the “ ***stress –c 80 –m 50 -1= 1000***”. As shoen in fig .

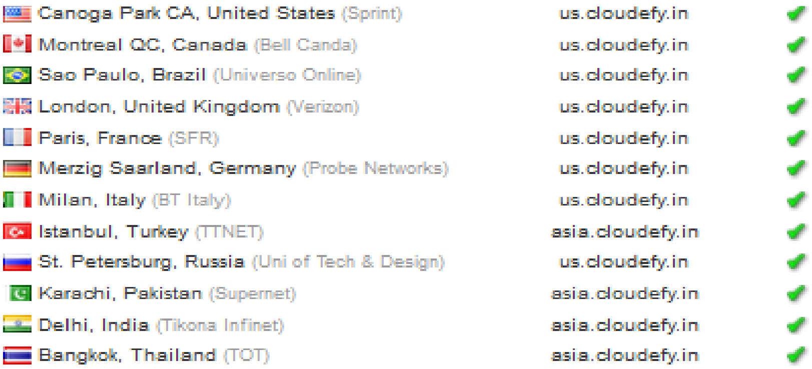


```
u1 1.0.1-1ubuntu1 [17.0 kB]
Fetched 17.0 kB in 0s (372 kB/s)
Selecting previously unselected package stress.
(Reading database ... 51693 files and directories currently installed.)
Preparing to unpack .../stress_1.0.1-1ubuntu1_amd64.deb ...
Unpacking stress (1.0.1-1ubuntu1) ...
Processing triggers for install-info (5.2.0.dfsg.1-2) ...
Processing triggers for man-db (2.6.7.1-1) ...
Setting up stress (1.0.1-1ubuntu1) ...
ubuntu@ip-10-230-29-66:~$ stress -c 85 -i 100
stress: info: [2585] dispatching hogs: 85 cpu, 100 io, 0 vm, 0 hdd
```

Testing :- for testing we have deployed simple web application on each instance differentiating the different availability zone .

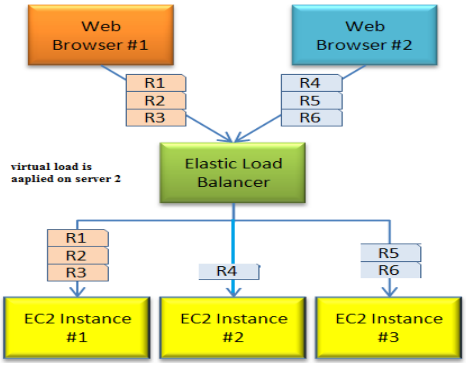
VII. Results

- We have searched www.cloudefy.in from the different geographical area with the help of proxy server and routed nearest hosting service as shown in fig this resolve our first issue of latency based routing latency based routing forward these request to regional load balancers to process the request, *us.clodefy.in* & *asia.cloudefy.in* representing the regional load balancer (step 2).

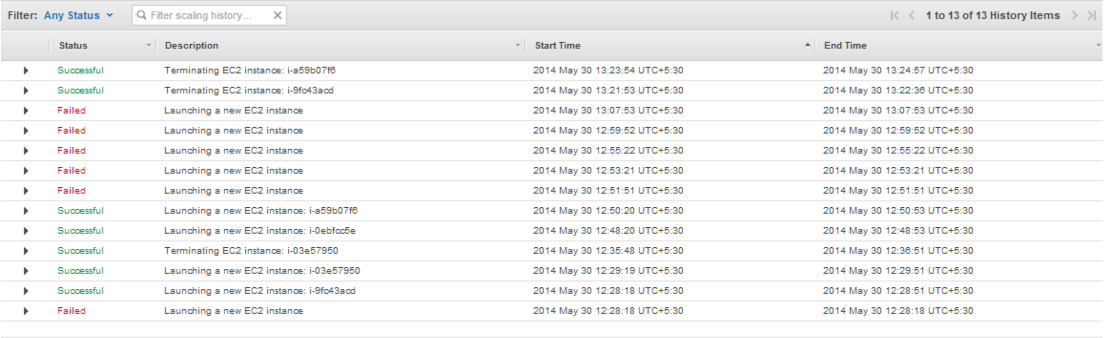


Canoga Park CA, United States (Sprint)	us.cloudefy.in	✓
Montreal QC, Canada (Bell Canda)	us.cloudefy.in	✓
Sao Paulo, Brazil (Universo Online)	us.cloudefy.in	✓
London, United Kingdom (Verizon)	us.cloudefy.in	✓
Paris, France (SFR)	us.cloudefy.in	✓
Merzig Saarland, Germany (Probe Networks)	us.cloudefy.in	✓
Milan, Italy (BT Italy)	us.cloudefy.in	✓
Istanbul, Turkey (TTNET)	asia.cloudefy.in	✓
St. Petersburg, Russia (Uni of Tech & Design)	us.cloudefy.in	✓
Karachi, Pakistan (Supernet)	asia.cloudefy.in	✓
Delhi, India (Tikona Infinet)	asia.cloudefy.in	✓
Bangkok, Thailand (TOT)	asia.cloudefy.in	✓

- At the regional load balancer the load balancer distributed the load in round robbing fashion . as shown in fig . here in the figure requested is generated randomly and thrown on the load balancer. some virtual load is also applied on the instance 2 as we can see from the picture load is distributed among the instances thus it sole our second proposed issue of load balancing in EC2



- Auto scaling result is collected from the summery of autoscaling group activity in the cloud watch . load pattern and the polices is already explained in step 5 and step 4 . these result is according to the load pattern sown in step5 . in starting as we defined the initial group size was 2 instances. But we also define that if maximum CPU utilizations less then 30% terminate one instance this is shown in the 4th line from bottom of figure. When we applied the load the auto scaling system automatically launched the new instances (5th and 6th line). Latter when load was remove autoscaling system terminated the newly created instances. (1st and 2nd line). Failed is system error of cloud network.



Filter: Any Status | Filter scaling history... | 1 to 13 of 13 History Items

Status	Description	Start Time	End Time
Successful	Terminating EC2 instance: i-a59b07f6	2014 May 30 13:23:54 UTC+5:30	2014 May 30 13:24:57 UTC+5:30
Successful	Terminating EC2 instance: i-9fc43acd	2014 May 30 13:21:53 UTC+5:30	2014 May 30 13:22:36 UTC+5:30
Failed	Launching a new EC2 instance	2014 May 30 13:07:53 UTC+5:30	2014 May 30 13:07:53 UTC+5:30
Failed	Launching a new EC2 instance	2014 May 30 12:59:52 UTC+5:30	2014 May 30 12:59:52 UTC+5:30
Failed	Launching a new EC2 instance	2014 May 30 12:55:22 UTC+5:30	2014 May 30 12:55:22 UTC+5:30
Failed	Launching a new EC2 instance	2014 May 30 12:53:21 UTC+5:30	2014 May 30 12:53:21 UTC+5:30
Failed	Launching a new EC2 instance	2014 May 30 12:51:51 UTC+5:30	2014 May 30 12:51:51 UTC+5:30
Successful	Launching a new EC2 instance: i-a59b07f6	2014 May 30 12:50:20 UTC+5:30	2014 May 30 12:50:53 UTC+5:30
Successful	Launching a new EC2 instance: i-0ebfcc5e	2014 May 30 12:48:20 UTC+5:30	2014 May 30 12:48:53 UTC+5:30
Successful	Terminating EC2 instance: i-03e57950	2014 May 30 12:35:48 UTC+5:30	2014 May 30 12:36:51 UTC+5:30
Successful	Launching a new EC2 instance: i-03e57950	2014 May 30 12:29:19 UTC+5:30	2014 May 30 12:29:51 UTC+5:30
Successful	Launching a new EC2 instance: i-9fc43acd	2014 May 30 12:28:18 UTC+5:30	2014 May 30 12:28:51 UTC+5:30
Failed	Launching a new EC2 instance	2014 May 30 12:28:18 UTC+5:30	2014 May 30 12:28:18 UTC+5:30

- step 3 – testing by searching www.cloudefy.in from normal and through proxy web server result as fallow

VIII. Conclusion

The current work aim to provide an detailed knowledge of importance of load balancing, autoscaling, resource monitoring and latency based load balancing for cloud environment. in this research we used Amazon cloud environment to develop a load efficient model if all the service is used as individual. It will not effective but if used together then an efficient load balancing. Autoscaling in cloud is depends on the their uses . autoscaling can be achieved in different way on different cloud platform. If we compare the services provided by the different cloud service provider the Amazon web services is more reliable and cost efficient,

Load balancing is very challenging issue and need to divide the load among the instances in distributive manner on this paper proposed a model using cloud services to provide the best solution to the load balancing issue

References

[1] Eddy Caron: Auto-Scaling, Load Balancing and Monitoring in Commercial and Open-Source Clouds

[2] Miss.Rudra Koteswaramma : Client-Side Load Balancing and Resource Monitoring in Cloud , ISSN: 2248-9622

[3] N. Ajith Singh, M. Hemalatha, “An approach on semi distributed load balancing algorithm for cloud computing systems” International Journal of Computer Applications Vol-56 No.12 2012.

[4] Nitika, Shaveta, Gaurav Raj, International Journal of advanced research in computer engineering and technology Vol-1 issue-3 May-2012

[5] Zenon Chaczko, Venkatesh Mahadevan, Shahrzad Aslanazadeh, and Christopher, IPCSIT Vol-14, IACSIT Press Singapore 2011

[6] Ali M. Alakeel, A Guide to Dynamic Load Balancing in Distributed

[7] Ali M. Alakeel, A Guide to Dynamic Load Balancing in Distributed Computer Systems, IJCSNS International Journal of Computer Science and Network Security, VOL.10 No.6, June 2010.

[8] http://www.amazon.com/gp/browse.html?node=201590011

[9] Amazon Elastic Compute Cloud http://aws.amazon.com/ec2/.

[10] Amazon web services cloud watch Web Site, November 2013.

[11] Aws elastic load balancing Web Site, November 2013

[12] R. Ranjan, A. Harwood, and R. Buyya. Peer-to-Peer Based Resource Discovery in Global Grids: A Tutorial. IEEE Communications Surveys and Tutorials, Volume 10, Issue 2, Pages 6-33, IEEE Communication Society, 2008.