# Dynamic Auto Scaling by means of Infrastructure as a code in Cloud Environment

Aayushi Chaudhari[1], Prof. Gayatri S. Pandi (Jain)[2]

[1]*M.E Student, Dept. of Computer Engineering, L.J College of Eng. & Tech., Ahmedabad, Gujarat, India*
[2]*Head of the department, Computer Engineering Department, L.J College of Eng. & Tech. Ahmedabad, Gujarat, India*

**ABSTRACT**

*Cloud Computing is an emerging technology nowadays. Scalability is an important aspect in cloud computing especially, when the demands of user changes abruptly. Auto-scaling is the strategy that has ability to adjust the available resources to meet the user demands. As the popularity and usage of cloud computing has increased, it leads to highly unpredictable demands of users. One major issue about providing automatic scalability is that it considers user-defined threshold, which cannot respond to real time internet traffic loads. In this paper, a model is proposed which includes scaling of resources by means of infrastructure as a code. This mechanism will dynamically create the high level language code to automate provisioning of resources based on availability of existing resources and user demands. Based on the code generated, the virtual machines will shrink or expand.*

**Keywords:** - *Infrastructure as a Code, Auto-Scaling, Cloud Watch*

## 1. Introduction

Cloud computing offers small and big organizations, the opportunity to scale their computing resources. It is done by either increasing or decreasing the required resources. Cloud computing provides delivery of resources on demand over the internet. It also provides the users to store and access the data stored by them on cloud. It provides metered service, so that users are asked to pay only for what they use. Cloud provides elasticity by scaling up as computing needs increase and then scaling down again as demands decrease.


**Fig-1: Cloud Computing**

Amazon Cloud Watch Monitor is a monitoring service for AWS cloud resources and the applications you run on AWS. Amazon Cloud Watch collects and monitor log files, set alarms, and automatically react to changes in AWS resources. Amazon Cloud Watch Monitor (CWM) is used to monitor resource utilization and application performance.

Infrastructure as Code (IAC) is a type of IT infrastructure that operations teams can automatically manage and provision through code, rather than using a less flexible or manual process. Infrastructure as Code is sometimes referred to as programmable infrastructure. It does configuration through machine-processable definition files, rather than physical hardware configuration or the use of interactive configuration tools.

## 2. Related Work

[1] In first paper, Abul Bashar proposed Bayesian Networks based predictive modeling framework that captures the historical behavior of the system involving various performance metrics. He used two modules, namely, the Datacenter Management System (DMS) and the Decision Support System (DSS) to create the model to make the decisions and also to predict future behavior by estimating the desired metrics of interest. Then this learned model is transformed to ID by adding decision node to it i.e. Scalability_Control with two actions Scale_Down and Scale_Up and a utility node named Reward. So the ID is responsible to make the decision to scale up or scale down based on the Response_time. So, if the ID makes correct decision then it is rewarded and if it makes wrong decision then it gets the penalty.

[2] In second paper, Marco A. S. Netto, Carlos Cardonha, Renato L. F. Cunha, Marcos D. Assuncao evaluates various autoscaling strategies uses Auto-scaling Demand Index (ADI) that reports utilization level, if the auto-scaling demand Index is above the target utilization then QoS is penalized and if the auto-scaling demand Index is below the target utilization. So, the main goal is to reduce the gap between actual utilization and the target interval. Here, the two main challenges are executed they are: first one decision on when to trigger auto-scaling operations and second is the step size that is used to expand or shrink the resource pool. It included three strategies for triggering auto-scaling operations that are reactive, conservative and predictive and two strategies for setting auto-scaling step sizes are fixed and adaptive. Reactive strategy had an excellent performance as it always reacts immediately to any deviation from the desired utilisation interval.Different trace logs publically available from production data center cluster by Google, this are the records of user jobs submitted to the cluster, their CPU usage and duration. Based on that tree of the strategies are applied along with step sizes to check out the observations.

[3] In third paper, Ali Yadavar, Nikravesh Samuel, A. Ajila, Chung-Horng Lung, conducted experiment on Amazon EC2 infrastructure using Hidden Markov Model (HMM). CPU utilization, throughput, and response time are being considered as performance metrics in this experiment. There are two broad categories of auto-scaling systems: reactive and proactive (predictive). Reactive auto-scaling approaches react to the system changes but do not anticipate future changes. Proactive scaling approaches try to predict future system behavior and adjust application resources in advance to meet the future needs. To generate historical data, they have used TPC-W benchmark as load generator for 5 hours. After collecting the data it is divided into training and testing data. It considers different metrics such as Mean Absolute Percentage Error (MAPE), Root Mean Square Error (RMSE). It compared scaling decisions of cloud watch and Hidden Markov Model.

[4] In fourth paper, Ali Yadavar Nikravesh, Samuel A. Ajila, Chung-Horng Lung proposed a model to increase the prediction accuracy of auto-scaling systems by choosing an appropriate time-series prediction algorithm based on the performance pattern. Workload is considered as the performance metric. Support Vector Machine (SVM) and Neural Networks (NN) were utilized as time-series prediction techniques. An autonomic element will regularly sense sources of change by using “sensors” or “reasons” about the current system situation for possible adaptation and action for the future. Autonomic manager applies the domain specific knowledge linked to the cloud workload pattern and apply the

appropriate predictor algorithm. Then, predictor and workload interact to implement the chosen algorithm and decide whether to scale up or scale down.

[5] In fifth paper, Wen-Hwa Liao, Ssu-Chi Kuai and Yu-Ren Leau, proposed dynamic threshold adjustment strategy that can expedite the creation of virtual machines according to workload demands. Their main goal is to reduce the web application response time and error rate when the system is under a heavy workload. It can expedite the release of virtual machines to reduce virtual machine running time when the system is under a light workload. Firstly, Resource usage and CPU utilization values are monitored using the Cloud Watch service of AWS. Then this values are transferred to a dynamic threshold controller for analysis. The controller transmits a timely policy and threshold to the virtual machines for providing computing resources. The upper threshold of CPU resource utilization should be set to approximately 50%–75% and the dynamic **lower threshold** of CPU utilization was designed in the range 5%–30%.

## 3. Proposed Work

In this section, we have described proposed model for management of resources by auto scaling using an infrastructure as a code mechanism. This model will be useful for improving the utilization of resources at granular level as the configuration of virtual machine will be done based on the user requests i.e. either the user needs large number of resources or micro level of resources. First the cloud user will request for resources and this request will be directed to the AWS i.e. Amazon Web Service. Load balancer will handle the request of user by firstly seeking out if the resources are available or not.

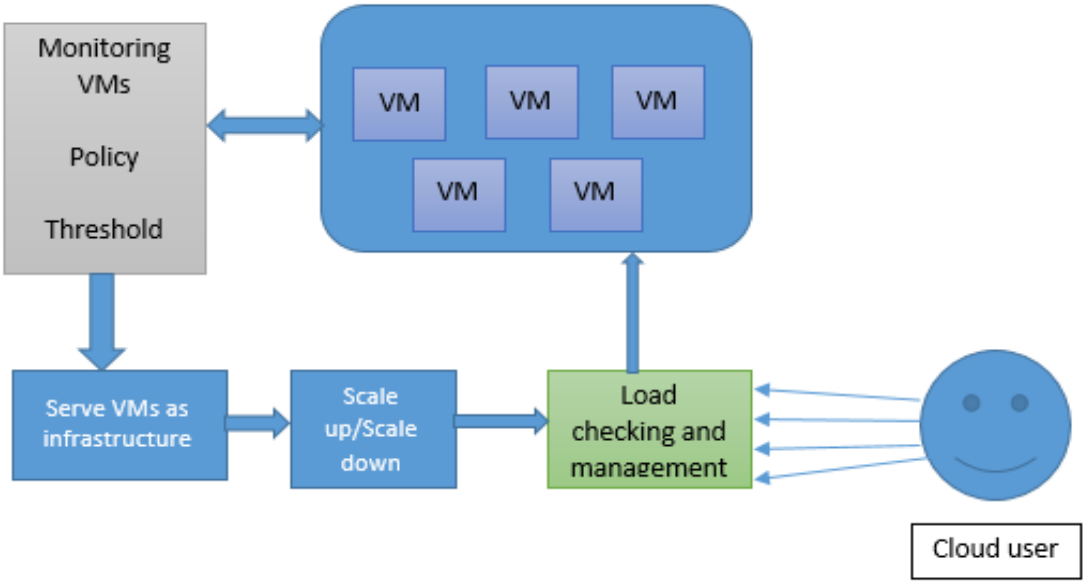


**Fig-2: Proposed Methodology**

If resources are available, it places the request. Load balancer acts as a proxy to handle the traffic across the servers. If the resources are not available then scale up needs to be done. Cloud Watch Monitor (CWM) continuously monitors the usage of resources, does resource accounting and maintain the statistics. Based on the information collected by CWM, threshold is updated and every time the threshold will be changed taking current conditions into account by dynamic algorithm. After the dynamic threshold is updated the dynamic code is automatically created by infrastructure as a code module. This code will decide whether to scale up or scale down. Every time this code will vary based on the arrived request and condition of the resource usage.

Then this decision is passed to infrastructure manager, will decide to increase or decrease the VMs. Here the novelty factor is that this system will provide extraordinary flexibility by creating dynamic code every time based on current situation. And also, provide the dynamism in threshold.

## 4. Result Analysis

We are calculating the CPU utilization of different virtual machines and based on the utilization the incoming request will be allocated for processing to that particular CPU and if required additional server is added to it for handling the request.

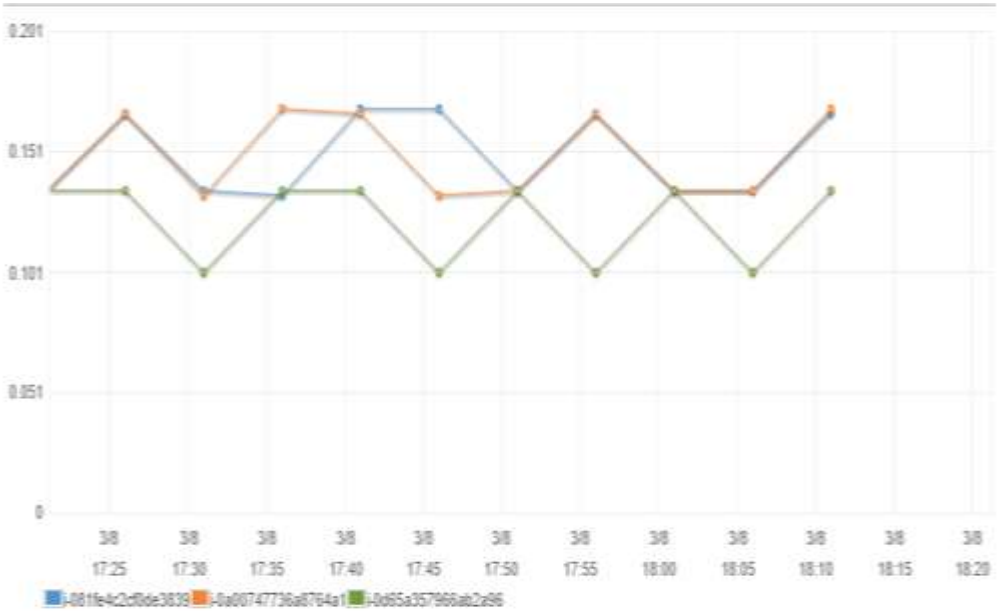


**Fig-3: CPU utilization for three VMs**

Three servers are in working state that shows the incoming request that is being handled on the different servers based on the load on particular server at an instance of time. If required the server will be added to handle user requests.

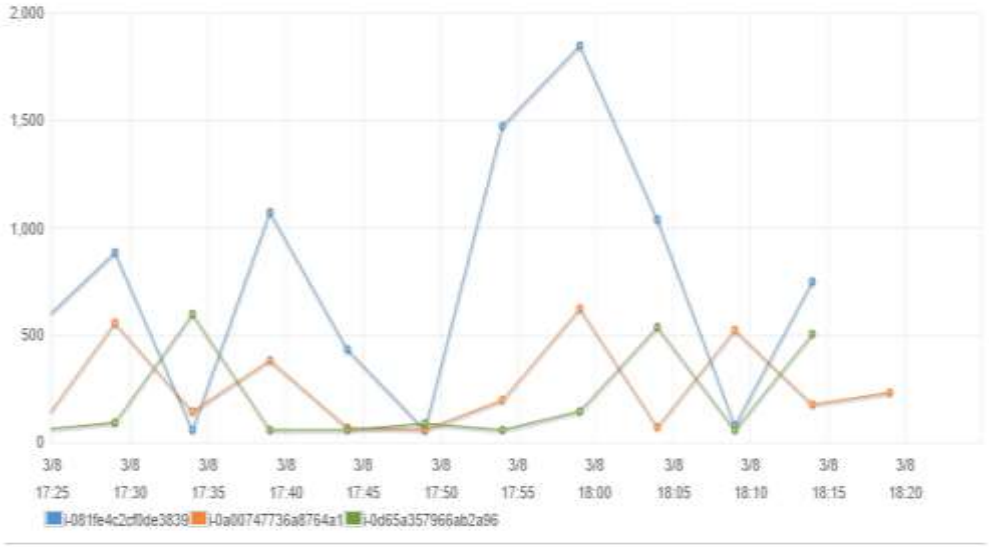


**Fig-4: Incoming request load coming to three servers**

## 5. Conclusion

Cloud computing is an emerging trend and as the resource demands of users are increasing there is need to provide efficient resources to them with ease. In our research, we will represent the novel approach for Auto-scaling using Infrastructure as a code which will generate a code for scaling up and scaling down of resources and Cloud watch Monitor will be used for keeping watch on utilization of resources and incoming requests from users with in Amazon web services for effective utilization of resources.

## 6. References

[1] Abul Bashar. "Autonomic Scaling of Cloud Computing Resources using BN-based Prediction Models Sensor Network" (2013) IEEE Conference Publications DOI: 10.1109/CloudNet.2013.6710578

[2] Marco A. S. Netto, Carlos Cardonha, Renato L. F. Cunha, Marcos D. Assuncao. "Evaluating Auto-scaling Strategies for Cloud Computing Environments ". (2014) IEEE Conference Publications DOI: 10.1109/MASCOTS.2014.32

[3] Ali Yadavar, Nikravesh Samuel, A. Ajila, Chung-Horng Lung. "Cloud Resource Autoscaling System based on Hidden Markov Model (HMM)"(2014) DOI: 10.1109/ICSC.2014.43IEEE Conference Publications

[4] Ali Yadavar Nikravesh, Samuel A. Ajila, Chung-Horng Lung. "Towards an Autonomic Auto-Scaling Prediction System for Cloud Resource Provisioning." (2015) 2015 10th International Symposium, DOI: 10.1109/SEAMS.2015.22

[5] Wen-Hwa Liao, Ssu-Chi Kuai and Yu-Ren Leau. " Auto Scaling Strategy for Amazon Web Services in Cloud Computing." (2015) 2015 IEEE International Conference. DOI: 10.1109/SmartCity.2015.209