ISSN 2278-3091
Volume 9, No.5, September - October 2020
International Journal of Advanced Trends in Computer Science and Engineering
Available Online at http://www.warse.org/IJATCSE/static/pdf/file/ijatcse298952020.pdf
https://doi.org/10.30534/ijatcse/2020/298952020

Key Decisions to Design a Secure Cluster in GKE Cloud

Dr. Vijaya Sri Kompalli[1], Udaya Sri Kompalli[2]
[1]Department of Computer Science and Engineering, Koneru Lakshmaiah Education Foundation, India,
kompallivsri@gmail.com
[2]Department of Computer Science and Engineering, NRI Institute of Technology, India,
kudayasri@nrigroupofcolleges.com

ABSTRACT

Virtualization technologies and secure cloud on-demand services escalate the need of cloud orchestration. Numerous Cloud Service Providers enhance the orchestration of cloud services based on well-configured projects. Several factors make customers cross-fingered to host resources on cloud in terms of virtualization, security, cost, load balancing, resource deployment, utilization and more. High demand to on-cloud services tend towards security and policies. Google Kubernetes Engine migrates the resources from cloud to GKE and supports security policies. The major view of this study is to focus on the key issue in the design of the cluster and decisions towards network set up to secure user projects. Use of subnets and cluster design policies secure the cloud applications to a greater extent. The Kubernetes cluster architecture and design considerations are applied and implemented on GKE in GCP and the results are shown.

Key words : configuration, deployment, devops, GKE, orchestration, security, virtualization

1. INTRODUCTION

Computing on cloud become essential in order to address the situations of data dulge. Use of data generating tools grow exponentially during the situations of COVID-19. Physical transactions were on hold due to pandemic situations and almost for almost past eight months in India, all the business transactions run online. Many businesses started to run on cloud due to which concern towards security in cloud has become a mere concern [21-24]. Cloud security at various levels of application design in view of Iaas, Paas, Saas have different security concerns and design details to deal with [1]. In recent technological evolutions, cloud computing has numerous cloud orchestration frameworks at horizontal and vertical levels. Even the multi-cloud infrastructures have become the trend of technology usage [2]. Hybrid deployment models combine public and private clouds. Cloud services are provided by web host applications directly. This enhances the resource mobility [3]. Cloud orchestrations using containers are lighter than the virtual machines. On-premise cloud orchestration using containers is another framework of cloud service. Set of container images are tested to be away from risk using image scanning tools [4]. Secure orchestration in hybrid cloud environment using OpenStack is another category of work in which the orchestration performed following few security policies within a data center [5]. Configuring a secure cloud is another complex task. Model-driven orchestrations whose concern is to design a secure-aware model in cloud [6]. Various cloud architectures and applications using cloud technologies were researched using various algorithms and techniques to deal with security issues [7]-[20].

a section that you want to designate with a certain style, then select the appropriate name on the style menu. The style will adjust your fonts and line spacing. **Do not change the font sizes or line spacing to squeeze more text into a limited number of pages.**

1.1 Final Stage

When you submit your final version, after your paper has been accepted, prepare it in two-column format, including figures and tables.

1.2 Figures

As said, to insert images in *Word*, position the cursor at the insertion point and either use Insert | Picture | From File or copy the image to the Windows clipboard and then Edit | Paste Special | Picture (with “Float over text” unchecked).

The authors of the accepted manuscripts will be given a copyright form and the form should accompany your final submission.

2. GKE KUBERNETES SERVER

GKE maintains some elements in the cluster and the container takes care of some other elements. User has a choice of options of elements from these two. Cluster Master runs the Kubernetes process. Master’s life cycle is managed by GKE during the modifications in the cluster. Master is responsible deciding what runs on each work nodes and for containerized applications and managing life cycle of work nodes in the

cluster. In GKE, user can specify the minimum and maximum scale of node pool. Master and the nodes communicate using Kubernetes API. Kubernetes API calls directly via http or grcp or indirectly from CLI. Master runs the API Kubernetes server to process request. Clusters master API server is the hub for all the communications in the cluster. APIM provide management, monitoring and authentication services. Also, container registry is a private container image. It is a single place where users can maintain Docker Images. It provides fine-grained access control. Most of the applications, docker hub is used as container registry for storing public docker images. But, for control access to images there is a need to use private registry like container registry. Container registry can be accessed through secured http end-points. It helps to push, pull or manage images from any system, VM instance or hardware

.

2.1 Virtualization and Shared Houses

In shared machines are not isolated. System utilizes common library, applications and OS. Virtual machines are isolated. In Containerization, isolation is preserved. In a containerized machine we need to specify a node machine type and OS image. Machine type is a set of virtualized hardware resources required by the user that is available to virtual machine instance including things like memory size, virtualized CPU. Each node is a standard compute engine machine type like the default one is N1-1 with one virtual CPU and 3.75 GB of memory. User can define their required machine type. Using GKE user can create their own cluster based on suitable budget requirements. A compute engine supplies an optimized OS images with docker installed. Users can create singe zone, multi-zone or regional cluster. Cluster type selection impacts the cluster's availability.

3. GKE WORKLOADS

In Kubernetes a job is termed as a Pod. It is a basic unit of deployment in a container. A worker node run pods with container. A pod represents a single instance in a running cluster. Pods can contain one or more docker containers. Kubernetes volumes allows containers and pods to share data. Pods can be moved to their nodes without reconfiguring. GKE uses IP tables to make nodes communicate directly due to which load balancers will not have failure issues. After creating .yaml file of a pod, it is uploaded to master. Then master decides which node to run further. If a pod stops running and restarts then the pod needs to define and deploy a replica specifying how many pods are run. In such case It is not a pod it is the deployment node that has meta data as deployment name, replica set, no. of pods to run, which pods to run in deployment and pods labels and containers. Health check of labeled nodes is done using http or gcp port to check on pod.

4. SECURE GKE CLUSTER

Users and administrators can use security measures to enable running container help another container in the same cluster. Nodes in cluster run and GC helps to enable the user projects. Privileges or containerized processes need to have some threshold in order to maintain security in a Kubernetes cluster. Few settings allow to change security settings of processes like user and group, available Linux capabilities and escalating privileges. Project creation need to have much focus on cluster design, based on application needs. Application deployment covers how to build and deploy images securely. Kubernetes Engines allows standard Google Cloud services and standards like projects, IAMs and VPNs etc.

4.1 Key Points to maintain for Cluster Security:

1. Communication needs of pods: Give firewall rules to allow connections to only those pods that need communication. Otherwise, block everything and set of white lists to identify the connectivity required.
2. Conventions to deploy cloud networks: Check existing network security connections and then make new one's compatible.
3. Which networks need to be given access to operate with Kubernetes control plane: Configure only the smallest set of authorized networks to do job securely.
4. IP mask enabling to all nodes in a cluster: IP mask reading has no drawbacks. It should be made by default enablement.
5. Limit level of access to workload: Create service accounts specific to applications to access Kubernetes Engine to access when there is a need. Best practice is to keep the number low as much as possible. It should follow custom limit service accounts. Application should not use service accounts for resource access. Need to have Google Account or Google Cloud account. Credential rotations are also best practices.

4.2 Key Decisions and Constraints to secure network:

1. Subnets span multiple zones but cannot span multiple regions.
2. Create firewall rules to allow traffic in between subnets.
3. Use tags to properly tag Kubernetes Engine rules and firewall rules from which facts need to be taken. Also, need to take care about public and private access.
4. Least privileges are best practice.
5. Meta data contains credentials and could be exploited if the pod compromises.
6. Binary Authorization on GCP for Software Supply Chain for software integrity and quality before deployment.

5. EXPERIMENTAL RESULTS

GKE procedures to scale and manage containers with required cluster setup.
Span regions within a single cloud environment. Challenges include Mixed resources, limited reachability, availability etc. Current deployment of heterogenous methods helps to resolve these challenges by multi-cloud deployments, fronting on-premise data, continuous integration and delivery.

In the code, different stages of secure cloud cluster shown from Figure 1-12, starting from Cloud setup and configuring a project with the given user authorization is done.

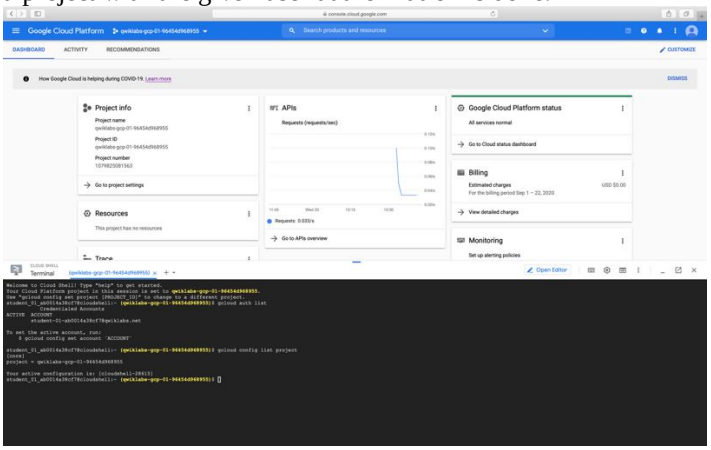

Figure 1: Configure a project

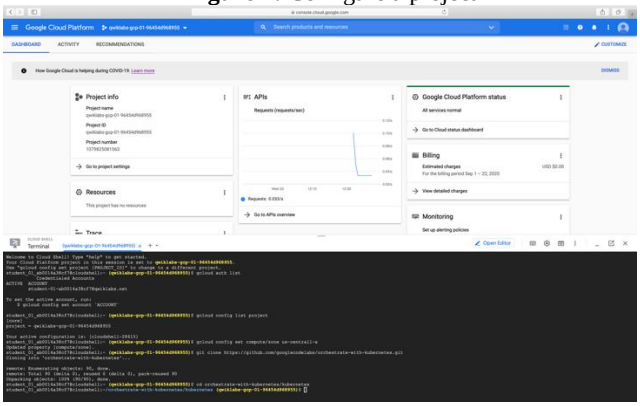

Figure 2: Zone setting and orchestration with GKE

.

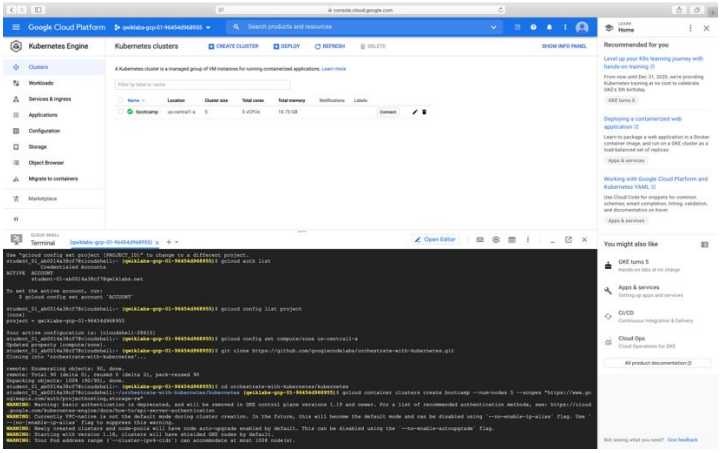

Figure 3:A bootcamp cluster is created

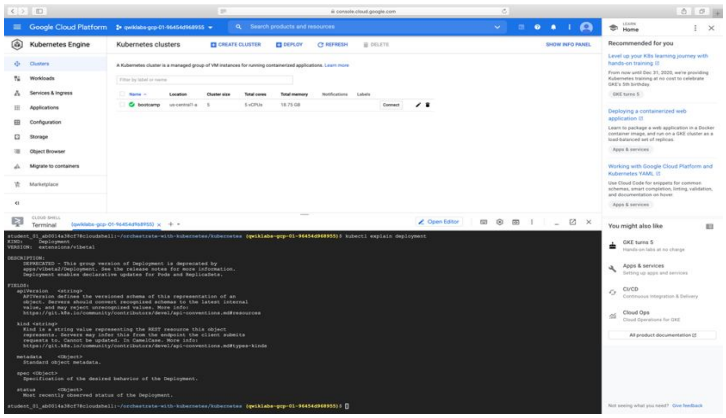

Figure 4: Version based deployment is done

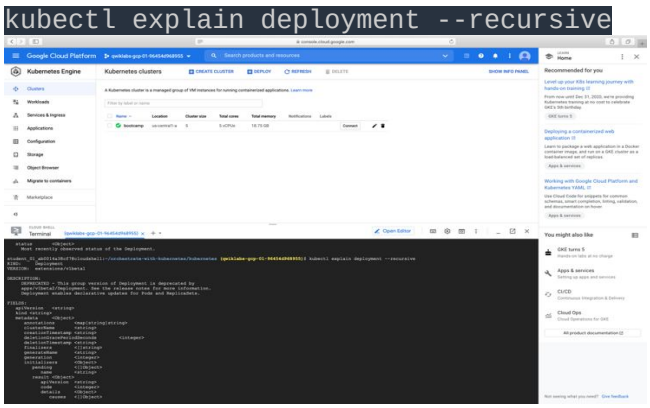

Figure 5: Recursive deployment is done

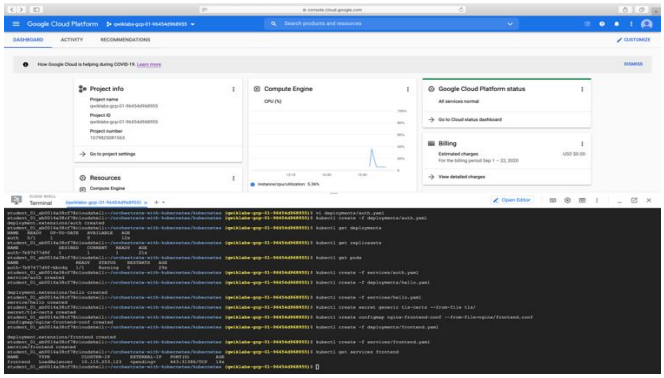

Figure 6: Create deployment and get replicas, pods, services

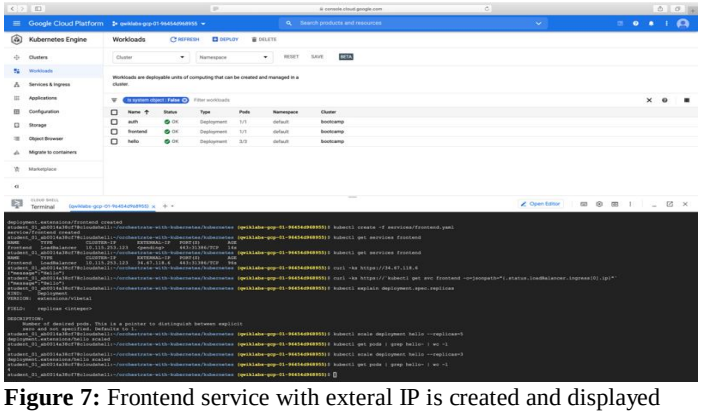

Figure 7: Frontend service with exteral IP is created and displayed communication.

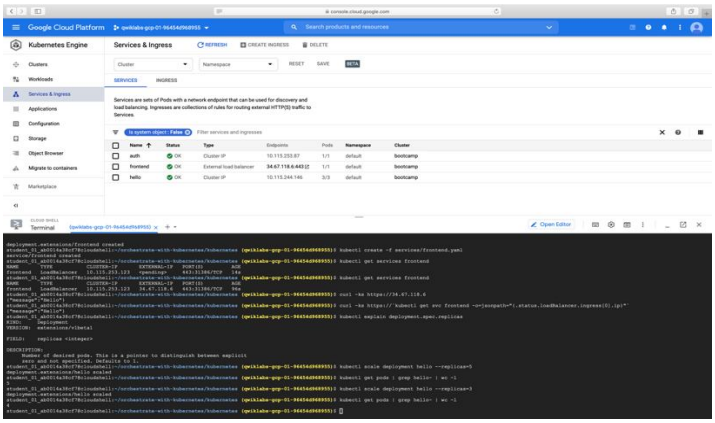

Figure 8: Scaling of deployment to 5 nodes

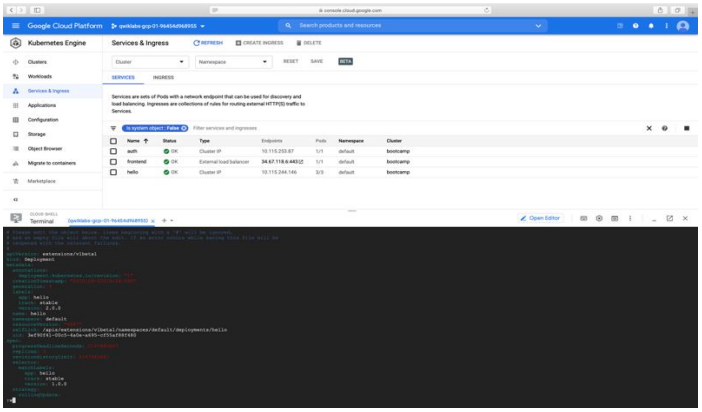

Figure 9: Rollout history of deployments

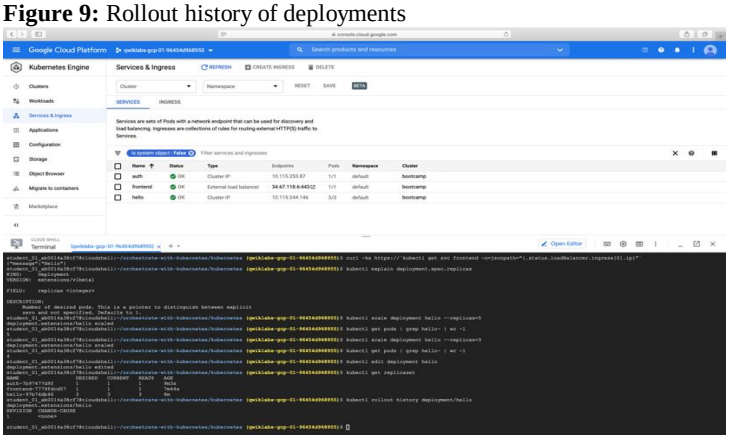

Figure 10: Paused the rolled out history

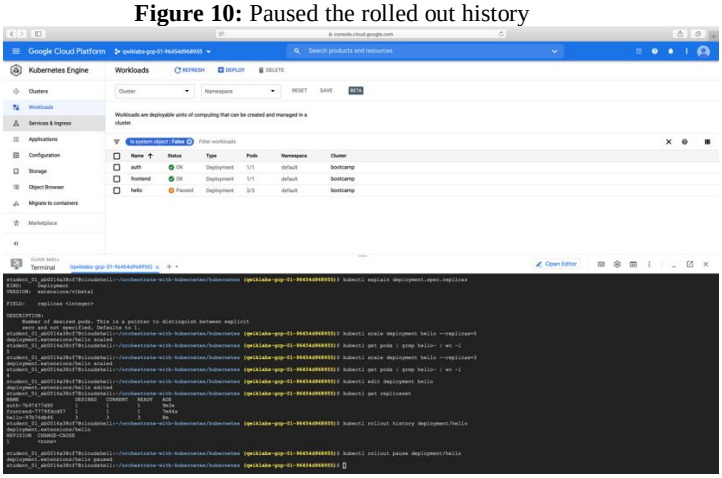

Figure 11: Final rollout of history

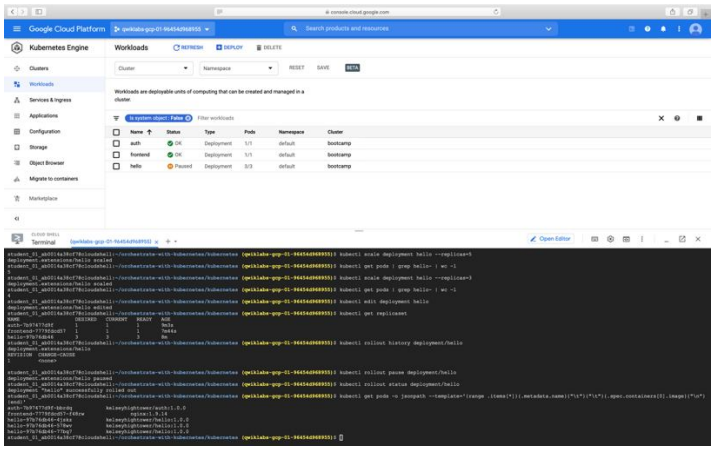

Figure 12: Secure Cluster Detail

5. CONCLUSION

Devops needs orchestrating cloud resources and to manage a secure cluster in Google Kubernetes Engine. Most of the server to client environments are communicated based on secure http: services. Security of cloud is based on the cluster design based on user application demand services. GKE security has some of the best practices that can enhance security in cloud. In the current study, implementation of user project using GKE server with required cluster design was orchestrated for efficient and secure cloud usage. By following the specified key points and applying the decisions in networking issues improves the quality and security of cloud resource utilization with GKE clusters.

.

REFERENCES

1. Yan, Xiaowei & Zhang, Xiaosong & Chen, Ting & Zhao, Hongtian & Li, Xiaoshan. (2012). **The Research and Design of Cloud Computing Security Framework.** Lecture Notes in Computer Science. 121. 757-763. 10.1007/978-3-642-25541-0_95.
2. Nicole Paladi,Antonis Michalas, Hai-Van Dang, **"Towards secure cloud orchestration for Multi-Cloud Deployments",** CrossCloud'18: Proceedings of the 5th Workshop on CrossCloud Infrastructures & Platforms, April 2018 Article No.: 4 Pages 1–6, doi.org/10.1145/3195870.3195874
3. S. Na, J. Park and E. Huh, **"Personal Cloud Computing Security Framework,"** 2010 IEEE Asia-Pacific Services Computing Conference, Hangzhou, 2010, pp. 671-675, doi: 10.1109/APSCC.2010.117.
4. S. Bhowmik, S. M. Saira Bhanu and B. Rajendran, **"Container Based On-Premises Cloud Security Framework,"** 2020 International Conference on Inventive Computation Technologies (ICICT), Coimbatore, India, 2020, pp. 773-778, doi: 10.1109/ICICT48043.2020.9112561.
5. D. Sitaram, S. Harwalkar, N. Ashwin and S. K. Ajmal, **"Secure Orchestration Based Federation in Hybrid Cloud Environments,"** 2015 International Conference

on Information Technology (ICIT), Bhubaneswar, 2015, pp. 13-19, doi: 10.1109/ICIT.2015.35.

6. E. Caron, A. Lefray and J. Rouzaud-Cornabas**, "Secured systems in Clouds with Model-Driven Orchestration,"** 2016 IEEE Conference on Communications and Network Security (CNS), Philadelphia, PA, 2016, pp. 498-506, doi: 10.1109/CNS.2016.7860541
7. Gavvala, Siva Kumar; Jatoth, Chandrashekar; Gangadharan, G. R.; Buyya, Rajkumar, **QoS-aware cloud service composition using eagle strategy, Future Generation Computer Systems**-The International Journal Of Escience, Jan 2019, 10.1016/j.future.2018.07.062
8. Lakum, Tarasvi; Rao, B. Thirumala, **A Key-Ordered Decisional Learning Parity with Noise (DLPN) Scheme for Public Key Encryption Scheme in Cloud Computing**, International Journal Of Advanced Computer Science And Applications, Nov-2019,
9. Balakrishna, G.; Rao, M. N., ESBL: **Design and Implement A Cloud Integrated Framework for IoT Load Balancing,** International Journal Of Computers Communications & Control, Aug 2019, 10.15837/ijccc.2019.4.3491
10. Dhote, B. L., & Krishna Mohan, G. (2019). **Trust and security to shared data in cloud computing: Open issues** doi:10.1007/978-981-13-2673-8_14
11. Gavvala, S. K., Jatoth, C., Gangadharan, G. R., & Buyya, R. (2019). **QoS-aware cloud service composition using eagle strategy.** Future Generation Computer Systems, 90, 273-290. doi:10.1016/j.future.2018.07.062
12. Kiranbabu, M. N. V., & Satyanarayana, K. V. V. (2019). **Inquisition the prospect of ranking cloud service provider using distinctive algorithms.** International Journal of Innovative Technology and Exploring Engineering, 8(4S), 171-176.
13. Kulkarni, N., Lalitha, S. V. N. L., & Deokar, S. A. (2019). **Real time control and monitoring of grid power systems using cloud computing.** International Journal of Electrical and Computer Engineering, 9(2), 941-949. doi:10.11591/ijece.v9i2.pp.941-949
14. Lavanya, K., Reddy, L. S. S., & Eswara Reddy, B. (2019). **Distributed based serial regression multiple imputation for high dimensional multivariate data in multicore environment of cloud**. International Journal of Ambient Computing and Intelligence, 10(2), 63-79. doi:10.4018/IJACI.2019040105
15. Mane, P. M., & Sheela Rani, C. M. (2019). **High data availability with effective data integrity and user revocation using abe scheme for cloud storage**. International Journal of Innovative Technology and Exploring Engineering, 8(4S2), 109-113.
16. Pallavi, L., Jagan, A., & Thirumala Rao, B. (2019). **BTS algorithm: An energy efficient mobility management in mobile cloud computing system for 5G heterogeneous networks.** Journal of Theoretical and Applied Information Technology, 97(1), 48-60.
17. Potluri, S., Subba Rao, K., & Vijaya Lakshmi, A. (2019). **Quality of service-based cloud models in manufacturing process automation** doi:10.1007/978-981-10-8201-6_26
18. Prasad, G. S., Praneetha, D. L., Srivalli, S., & Sukesh, B. V. (2019). **Information security in cloud by using enhanced triple-DES encryption algorithm.** International Journal of Innovative Technology and Exploring Engineering, 8(5), 679-682.
19. Sastry, J. K. R., & Basu, M. T. (2019). **Securing SAAS service under cloud computing based multi-tenancy systems.** Indonesian Journal of Electrical Engineering and Computer Science, 13(1), 65-71. doi:10.11591/ijeecs.v13.i1.pp65-71
20. Siva Thirdha, S., Satyanarayana, K. V. V., Tabassum, S. K., & Hima Vamsi, G. (2019). **An efficient pricing strategy based on utilization of the user in GPUACC (GPU-accelerated cloud computing).** International Journal of Innovative Technology and Exploring Engineering, 8(5), 185-188.
21. Usha. V., Sri Kompalli, V., **Resource Management and Simulation Tools in Fog Computing – A Comparitive Study**, International Journal of Advanced Trends in Computer Science and Engineering, 2020.
22. Kota, S.P., Sri Kompalli, V., **Automated Collection of Artifacts from a Live Windows System using e-Triage Tool**, International Journal of Advanced Trends in Computer Science and Engineering, 2020.
23. Karimunnisa, S., Kompalli VS, **Cloud Computing: Review of Recent Research Progress and Issues**, International Journal of Advanced Trends in Computer Science and Engineering, 2019.
24. Kishore K.R., Sri KV., **Disaster Preparedness: Veterinarian Perspective**, Springer Series in Geomechanics and Geo Engineering, 2019.