**Brief introduction about company** [BalaSuresh Korumilli](mailto:balasuresh@argoid.com)**to me**

* What kind of service company provides ?
* Informed about the recommendation we give.
* Told about Alert’s message/mail we get from customers and its types.
* How to respond to those Alert’s we get via chat/mail.
* Gave tunneling access for 10.1.0.15 machine prod gateway.
* Introduced services in 10.1.0.15 machine which are Alert manager, Prometheus, Grafana.
* Explained a bit about Alert manager like in it how to differentiate b/w alerts and how to snooze those alerts as per time and tag.
* Got access to groups (argoid-infra, argoid-prod3-alert, argoid-prod-support, argoid-prod-support-team, argoid-team).
* Done with Unix commands.
* Now going through the Hadoop Ecosystem.

**Unix**

**Kafka** - its a messaging system used for processing big data in real time

**Zookeeper** - it maintains/manages a cluster (leader election, health report etc.)

**HDFS** - it is to store the big data and read/write it in real time

**Mapreduce** - it map and reduce the data (which is arrange and reduce size)

**Yarn** - it provides Resource manager(RM) and Node manager(NM), (RM provides resource to manage data/helps to get out the data from database),(NM manages nodes)

**Alert Manager -** it gets metrics from prometheus and presents it in form of alerts in its UI, we can thereby set our own custom alerts based on metrics we need

**Grafana -** its shows metrics in graphical dashboard, metrics is related to system/software/network

**Prometheus -** it stores metrics in files (edit\_log and fs\_image), which is then provided to alert manager

**Node Exporter -** it gets system/software/network metrics and provides to prometheus, it should be on each node

**Nifi -** it automate the flow of data between systems or can say is for dataflow, (eg.- it converts whole data into structured/useful data which can be stored like in HDFS)

**Airflow -** its used for scheduling, and monitoring batch-oriented workflows, it represents in its own UI where each square is a process which in it is a DAG

**Cassandra -** it is to store the big data and read/write faster, (eg.- data produced using kairosDB metrics)

**KairosDB -** it gets data from metrics which is from application so that data in cassandra

**SFTP -** its used to share file securely b/w any two machines or local to remote machines

**Zeppelin** - It is used to run spark jobs , it's an interpreter like pycharm, vs code. It is a web-based notebook that enables interactive data analytics.

**Redis** - its a temporary database(stores data where its not secure but we can read/write data really quick),(eg.- like in RAM which can’t be kept for long, not idle)

**Haproxy** - its same like load balancer but on CLI, so we have to make all configurations/entries manually

**Nginx** - it redirects the packet to the path defined while sending, further where it stores data there we implement HAProxy for LB

**Redash** - its shows metrics in graphical dashboard (like grafana) for which it request presto, metrics is related to database and its faster than its competitors

**Presto -** this service has strong code which help to get data from multiple sources using hive queries

**Hive -** it is used to write queries to fetch data from multiple resources(like HDFS), it like SQL, PostgresQL

**Azure/AWS -** cloud based services

**Docker -** it basically runs/stores any data/application isolatedly also we could pull image of data/application and get those anywhere we want

**Git -** here we can store software/OS and there versions which can later be used as backup

**Elastic search**

Lets encrypt - **cert bot**

**Bitbucket -**

**Jenkins -**

**Containers -**

**Kubernetes -**

**Python -**

**Terraform -**

**Maven -**

—--------------------------------------------------------------------------------------------------------------------------------------------------------------

Udemy credentials:

Email: safi@sentienz.com

Password: jDwgUx6EA4TS

PROD 3 GATEWAY :- ssh -vND 8888 ghalyan@20.198.101.3 , MACHINE :- ssh ghalyan@10.1.0.15

TEST GATEWAY :- ssh ghalyan@20.219.19.30 , MACHINE :- ssh 172.16.0.103

PROD 4 GATEWAY :- ssh -vND 8888 ghalyan@20.219.14.5 , MACHINE :- ssh ghalyan@10.2.0.150

—--------------------------------------------------------------------------------------------------------------------------------------------------------------

<https://eng.uber.com/hadoop-container-blog/>

**To know which folder is using how much space**

Sudo du -sch /\*

Alerts related to search-engine is very

**CRITICAL**

Please ensure that you are aware of the runbook created to start the service.

—--------------------------------------------------------------------------------------------------------------------------------------------------------------

**TO START VM FROM CLI ➖**

ssh ghalyan@20.198.101.3

[ghalyan@argoid-prod3-host-0074 ~]$ /opt/vm\_start\_stop.sh status 172.16.0.102

- name: cleanup logs, free disk space below 20%

sudo: yes

command: find /var -name "\*.log" \( \( -size +50M -mtime +7 \) -o -mtime +30 \) -exec truncate {} --size 0 \;

when: "item.mount == '/var' and ( item.size\_available < item.size\_total \* 0.2 )"

with\_items: ansible\_mounts

**This will truncate any \*.log files on the volume /var that are either older than 7 days and greater than 50M or older than 30 days if the free disk space falls below 20%.**

**100/100 good job!**

—--------------------------------------------------------------------------------------------------------------------------------------------------------------

**TO CREATE A VM ➖**

#az vm create --name argoid-test-0160 --resource-group argoid-prod4-resource-group-1 --location centralindia --image "OpenLogic:CentOS:7\_6-gen2:latest" --size Standard\_B1s --authentication-type ssh --admin-username ghalyan --ssh-key-values ~/.ssh/prod4.id\_rsa.pub --storage-sku Standard\_LRS --os-disk-size-gb 30 --vnet-name argoid-prod4-vpc --subnet argoid-prod4-vpc-subnet-1 --nsg argoid-prod4-nsg --private-ip-address 10.2.0.160 --public-ip-address "" --tags 'customer=none' 'env=prod4' 'instance=test-delete'

**AFTER CREATING VM {COMMANDS/ANSIBLE} TO SETUP ➖**

#ansible-playbook -i inventory/saas-prod4/common.ini sites/argoid-user-site.yml --limit=10.2.0.160

#ansible-playbook -i inventory/saas-prod4/common.ini sites/argoid-team-users.yml --limit=10.2.0.160

#ansible-playbook -i inventory/saas-prod4/common.ini sites/host-setup.yml --limit=10.2.0.160

—--------------------------------------------------------------------------------------------------------------------------------------------------------------

**Store local mac’s {Public key} in = {authorized\_keys} folder of Gateway**

**Than,**

**Store Gateway’s {Public and Private key} in = {ansible folder} of Gateway**

**OR**

**Store Gateway’s {Public key} in = {authorized\_keys} folder of VM’s inside Gateway**

—--------------------------------------------------------------------------------------------------------------------------------------------------------------