MSIS 2628: The Business of Cloud Computing - Project #2

By-Anurag Gate, Sahil Sawhney

Business:

Note: Please find references at the end.

Amazon Web Services

AWS has already built a powerful global network to provide a virtual host for some of the world's most complex IT environments. Its data centers are fiber linked and arranged all over the world. In AWS the payments are scheduled according to the services you use down to the millisecond of compute time. In a nutshell, AWS is a fast and relatively easy way to migrate your DevOps to the cloud.

Google Cloud Platform

Google Cloud Platform is made up of a lot of different services and solutions to utilize the same software and hardware infrastructure that Google uses for its own products (like YouTube and Gmail).

Some of the main GCP's benefits are that it is one of the largest and most advanced computer networks, and it gives you an access to the numerous tools to help you focus on building your application. Stackdriver Monitoring, Stackdriver debugger, Stackdriver Logging, security scanner service (App Engine) and many more. You can use them all immediately as part of your application lifecycle pipeline.

Mapping AWS services to Google Cloud Platform products

Service Category	Service	AWS	Google Cloud Platform
Compute	IaaS	Amazon Elastic Compute Cloud	Compute Engine
	PaaS	AWS Elastic Beanstalk	App Engine
	Containers	Amazon Elastic Container Service	Google Kubernetes Engine
	Serverless Functions	AWS Lambda	Cloud Functions
	Managed Batch Computing	AWS Batch	N/A
Network	Virtual Networks	Amazon Virtual Private Cloud	Virtual Private Cloud
	Load Balancer	Elastic Load Balancer	Cloud Load Balancing
	Dedicated Interconnect	Direct Connect	Cloud Interconnect
	Domains and DNS	Amazon Route 53	Google Domains, Cloud DNS
	CDN	Amazon CloudFront	Cloud CDN
Storage	Object Storage	Amazon Simple Storage Service	Cloud Storage
	Block Storage	Amazon Elastic Block Store	Persistent Disk
	Reduced-availability Storage	Amazon S3 Standard-Infrequent Access, Amazon S3 One Zone-Infrequent Access	Cloud Storage Nearline
	Archival Storage	Amazon Glacier	Cloud Storage Coldline
	File Storage	Amazon Elastic File System	Cloud Filestore (beta)
Database	RDBMS	Amazon Relational Database Service, Amazon Aurora	Cloud SQL, Cloud Spanner
	NoSQL: Key-value	Amazon DynamoDB	Cloud Firestore, Cloud Bigtable
	NoSQL: Indexed	Amazon SimpleDB	Cloud Firestore

AWS Compute:

- Elastic Compute Cloud: Amazon's flagship compute service is Elastic Compute Cloud, or EC2. Amazon describes EC2 as "a web service that provides secure, resizable compute capacity in the cloud." EC2 offers a wide variety of options, including a huge assortment of instances, support for both Windows and Linux, bare metal instances, GPU instances, high-performance computing, auto scaling and more. AWS also offers a free tier for EC2 that includes 750 hours per month for up to twelve months.
- Container services: Within the compute category, Amazon's various container services are increasing in popularity, and it has options that support Docker, Kubernetes, and its own Fargate service that automates server and cluster management when using containers. It also offers a virtual private cloud option known as Lightsail, Batch for batch computing jobs, Elastic Beanstalk for running and scaling Web applications, as well as a few other services.

Google Compute:

- **Compute Engine:** By comparison, Google's catalog of compute services is somewhat shorter than its competitors'. Its primary service is called Compute Engine, which boasts both custom and predefined machine types, per-second billing, Linux and Windows support, automatic discounts and carbon-neutral infrastructure that uses half the energy of typical data centers. It offers a free tier that includes one f1-micro instance per month for up to 12 months.

- **Focus on Kubernetes:** Google also offers a Kubernetes Engine for organizations interested in deploying containers. Like all of the leading cloud vendors, it's set up to offer containers and microservices. And it's worth noting that Google has been heavily involved in the Kubernetes project, giving it extra expertise in this area.

AWS Storage:

- **SSS to EFS:** AWS offers a long list of storage services that includes its Simple Storage Service (S3) for object storage, Elastic Block Storage (EBS) for persistent block storage for use with EC2, and Elastic File System (EFS) for file storage. Some of its more innovative storage products include the Storage Gateway, which enables a hybrid storage environment, and Snowball, which is a physical hardware device that organizations can use to transfer petabytes of data in situations where Internet transfer isn't practical.

- **Database and archiving** On the database side, Amazon has a SQL-compatible database called Aurora, Relational Database Service (RDS), DynamoDB NoSQL database, ElastiCache in-memory data store, Redshift data warehouse, Neptune graph database and a Database Migration Service. Amazon offers Glacier, which is designed for long-term archival storage at very low rates. In addition, its Storage Gateway can be used to easily set up backup and archive processes.

Google Storage:

- **Unified Storage and more:** As with compute, GCP has a smaller menu of storage services available. Cloud Storage is its unified object storage service, and it also has a Persistent Disk option. It offers a Transfer Appliance similar to AWS Snowball, as well as online transfer services.

- **SQL and NoSQL** When it comes to databases, GCP has the SQL-based Cloud SQL and a relational database called Cloud Spanner that is designed for mission-critical workloads. It also has two NoSQL options: Cloud Bigtable and Cloud Datastore. It does not have backup and archive services .

Google Cloud Platform vs AWS Technical Comparison table

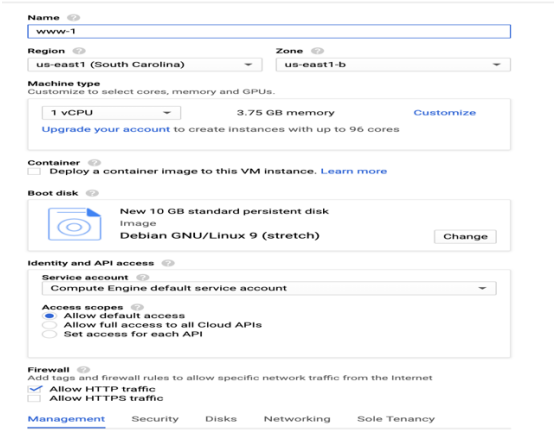
Category	AWS	GCP
Compute	You can equip AWS EC2 instances with up to 128 vCPUs and 3,904 GB of RAM.	You can equip Google Compute Engine instances with up to 96 vCPUs and 624 GB of RAM
Storage/Disk	General with volume sizes from 1GB to 16TB, and Provisioned IOPS SSD from 4GB to 16 TB	SSD, volume sizes from 1 GB to 64 TB
Network	Amazon EC2 instances have a maximum bandwidth of 25 Gbps, however, this is only on the largest instance sizes. Standard instances max out at 10 Gbps/second.	Each core is subject to a 2 Gbits/second (Gbps) cap for peak performance. Each additional core increases the network cap, up to a theoretical maximum of 16 Gbps for each virtual machine

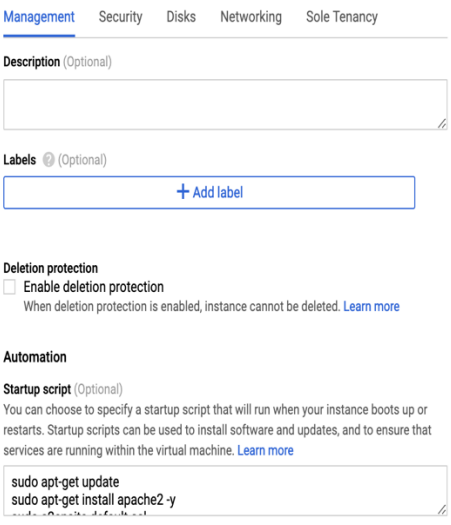
Deploy XAMPP on two instances in GCP

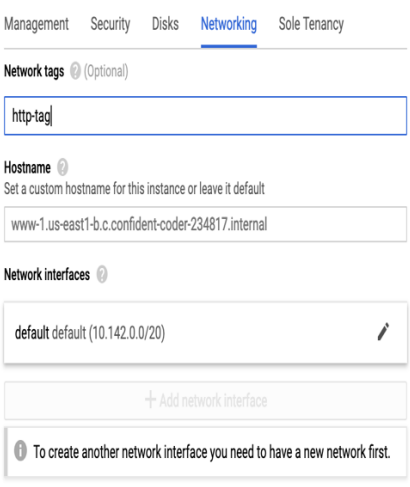
Create instances:

Name	Zone	Recommendation	In use by	Internal IP	External IP	Connect
inst3	us-east1-b		instance-group-5	10.142.0.4 (nic0)	35.227.55.36	SSH
inst4	us-east1-b			10.142.0.5 (nic0)	35.243.135.90	SSH
instance-5	us-east1-b		instance-group-5	10.142.0.6 (nic0)	35.231.69.171	SSH
instance-test	us-east1-b			10.142.0.3 (nic0)	35.231.23.23	SSH
sahilinstance	us-east1-b			10.142.0.2 (nic0)	35.196.218.225	SSH
ww-1	us-east1-b	Save $18 / mo	us-resources-w	10.142.0.7 (nic0)	35.231.26.71	SSH
ww-2	us-east1-b	Save $18 / mo	us-resources-w	10.142.0.8 (nic0)	35.229.95.171	SSH
ww-3	europe-west1-b	Save $20 / mo	europe-resources-w	10.132.0.2 (nic0)	35.195.128.146	SSH
ww-4	us-east1-b	Save $18 / mo		10.142.0.9 (nic0)	34.73.15.151	SSH
ww-5	europe-west1-b	Save $20 / mo	europe-resources-w	10.132.0.3 (nic0)	35.187.31.10	SSH

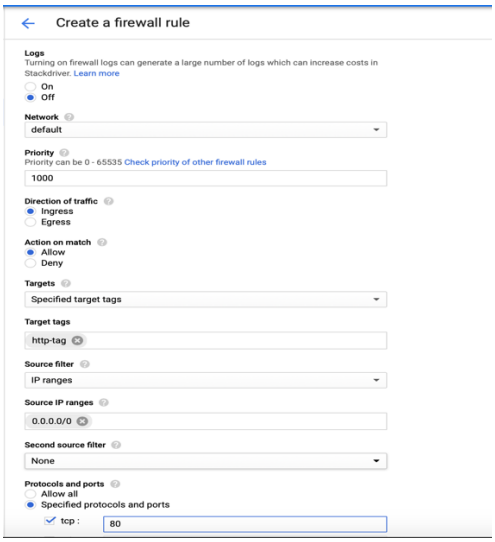
Configuring instances:



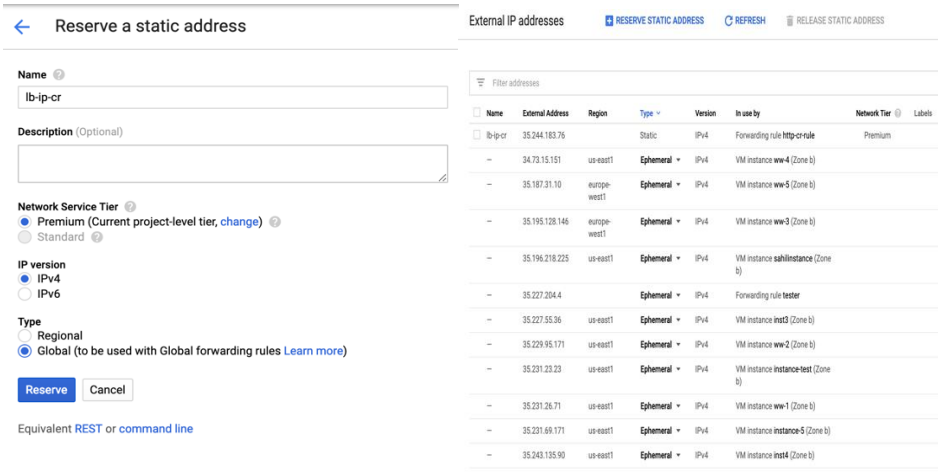




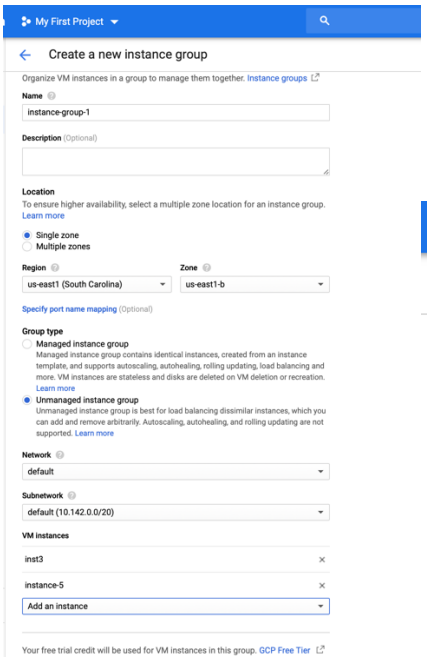
Create firewall rule:

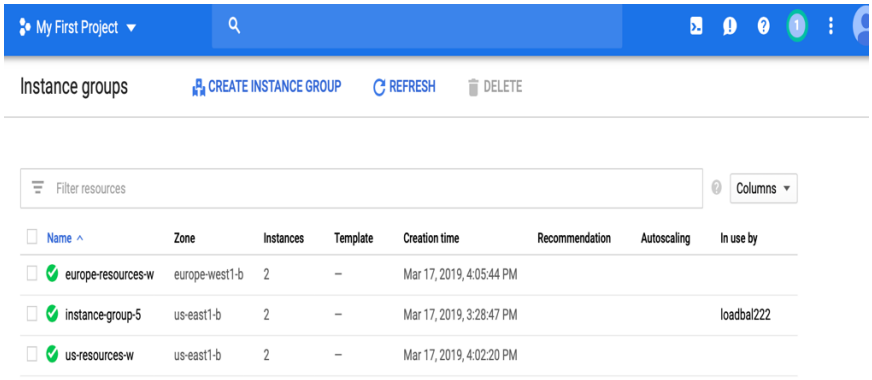


Create IPv4 and IPv6 global static external IP addresses for your load balancer:

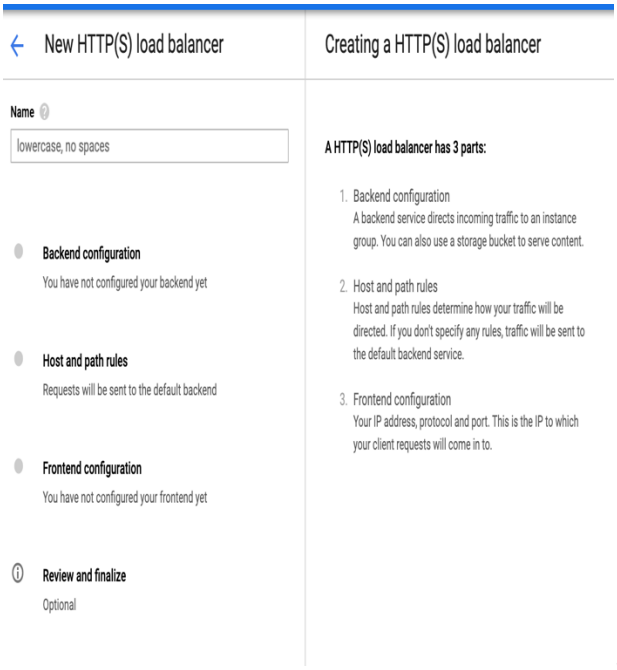


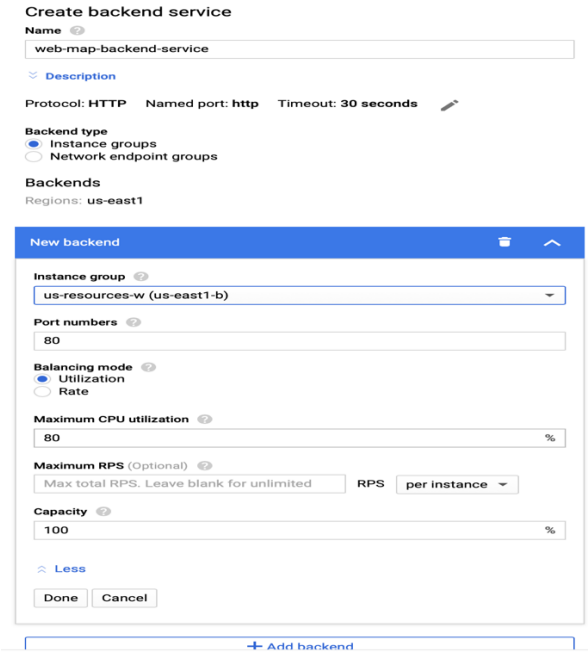
Create an instance group for each of your zones:

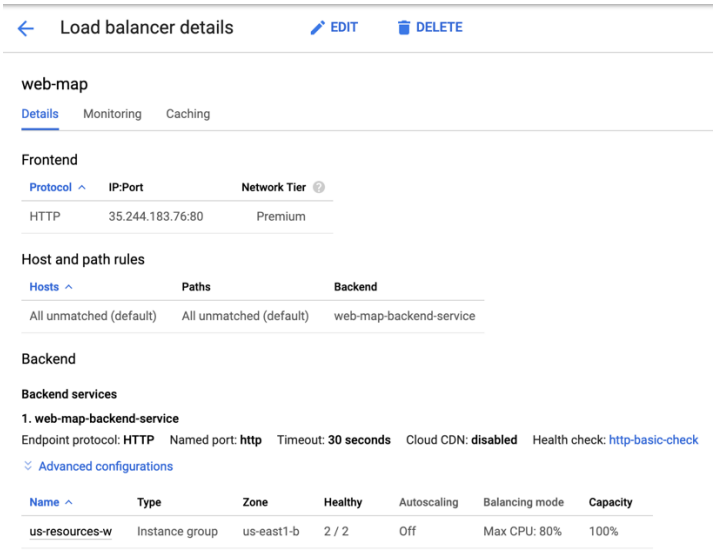




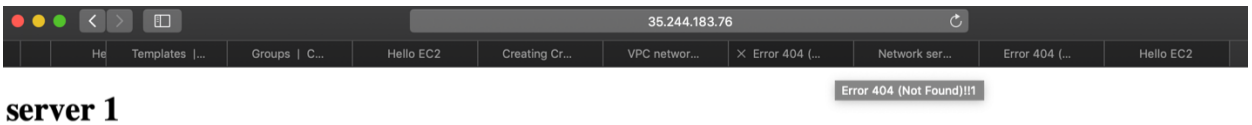
Configuring the load balancing service:







Accessing IP: Port



server 1

server 2

High level Services:

At a high level, cloud platforms begin by providing a set of baseline services: compute, storage, networking, and database services. AWS's baseline services include:

- Compute: Amazon Elastic Compute Cloud (EC2)
- Storage: Amazon Simple Storage Service (S3) and Amazon Elastic Block Store (EBS)
- Networking: Amazon Virtual Private Cloud (VPC)
- Databases: Amazon Relational Database Service (RDS) and Amazon DynamoDB

GCP's baseline services include:

- Compute: Google Compute Engine and Google App Engine
- Storage: Google Cloud Storage
- Networking: Google Virtual Private Cloud
- Databases: Google Cloud SQL, Google Cloud Firestore, and Google Cloud Bigtable

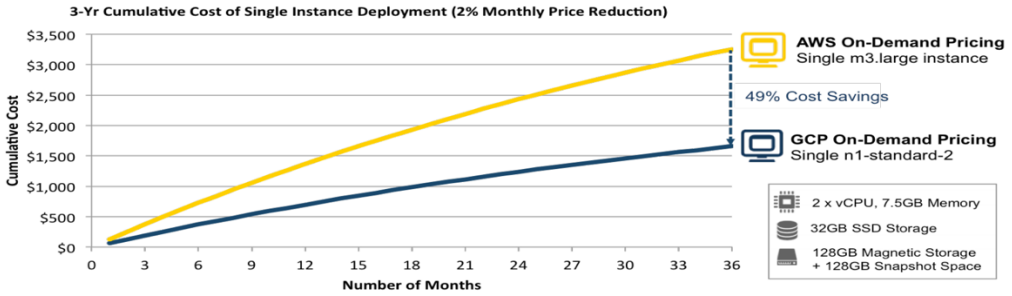
Cost Benefit Analysis

The Google and Amazon online pricing calculators were used to generate the cost of a single, medium-sized instance comprised of two vCPUs,

7.5GB of memory, 32GB of SSD storage, 128GB of magnetic storage, and 128GB of snapshot storage space. A 2% monthly

price reduction was included in the model based on an expected 25% annual cost reduction, as historically seen by cloud providers. In reality, cost reductions are unpredictable and would happen only a few times per year, but for simplicity of modeling we averaged the expected annual reduction across the entire year. As shown in Figure 3, the three-year cost of operating the on-demand GCP instance at 100% utilization (24 x 7) was 49% lower than the cost of operating the AWS on demand instance. ESG Lab also validated that the GCP on-demand pricing advantage was relatively consistent across other comparably sized instances as well as between the number of cores per instance

FIGURE 3. **Cost Comparison: Single Instance Deployment**



Source: Enterprise Strategy Group, 20

Why This Matters

So, why does this matter to you? Google offers a very simple, on-demand, real-time pricing structure, and works to automatically save you money. Customers know that Google will automatically discount prices based on sustained usage and will pass on any price reductions. AWS pricing is complex, and ESG Lab analysis comparing GCP and AWS on-demand pricing demonstrated a 49% cost advantage for equivalent VMs. Google delivers lower costs and less pricing complexity.

Unique offering of GCP:

Pre-emptible VMs

Hot on the heels of sustained usage discounts comes the preemptible VMs capability of GCE. It is designed for the same reason as sustained usage discounts – to drive maximum utilization of infrastructure. Unlike EC2 spot instances, customers need not bid for unused capacity. It removes the pain involved in complex bidding and gambling on fluctuating market movements. Any Google Compute Engine VM can be launched in preemptive mode

Custom VM Sizes:

Google Compute Engine offers custom VM sizes where customers can precisely choose the number of CPU cores and the amount of memory required for their workload. Depending on the zone where the VM is launched, customers can choose anywhere between 1 core to 32 cores of CPU and up to 6.5GB of RAM per vCPU. The VMs can run popular flavors of Linux or Windows operating systems.

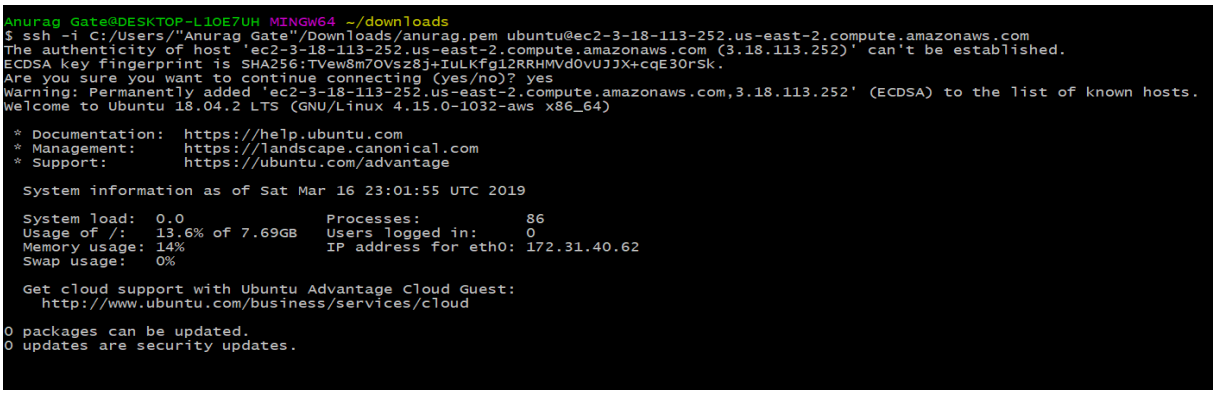
TECHNICAL PART 1

Creation of the 1st instance with image id ‘ami-0c55b159cbfafe1f0’.

Command: ‘aws ec2 run-instances --image-id ami-0c55b159cbfafe1f0 --count 1 --instance-type t2.micro --key-name Amazon_Key_HP --security-group-ids sg-0dd78dac14232eb6e --subnet-id subnet-d124f09d’

SSH to instance A using the public DNS of created 1st instance.

Command: ‘ssh -i 'C:\Users\HP\.ssh\Amazon_Key_HP.pem' ubuntu@ec2-13-59-1-118.us-east-2.compute.amazonaws.com’

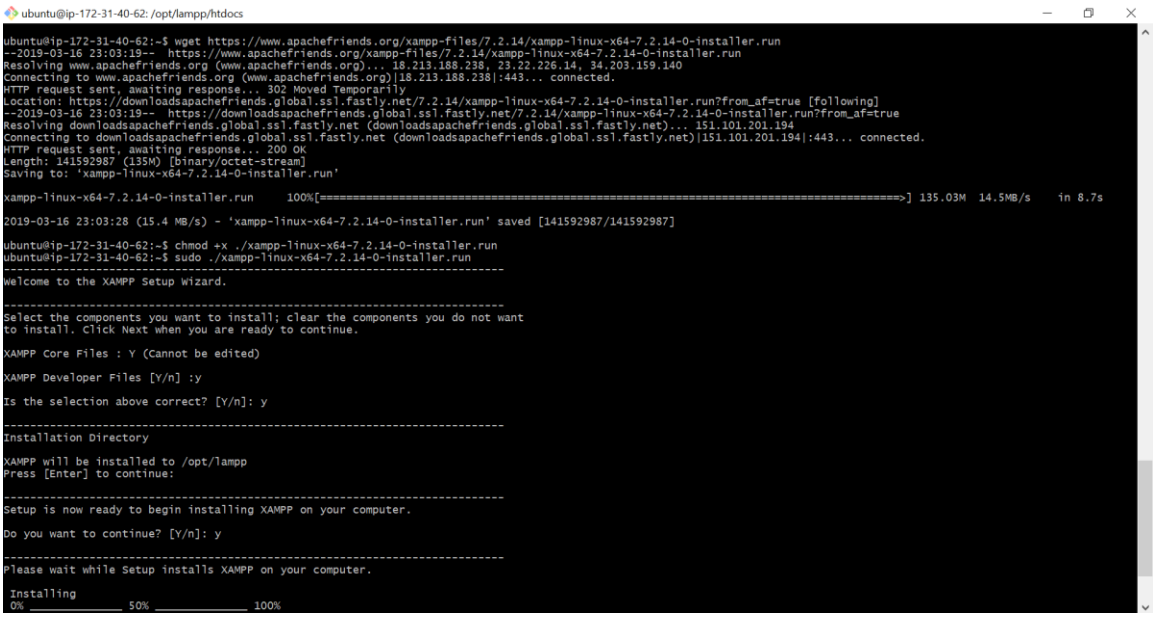


Install the XAMPP server on the instance.

Commands: ‘wget https://www.apachefriends.org/xampp-files/7.2.14/xampp-linux-x64-7.2.14-0-installer.run’

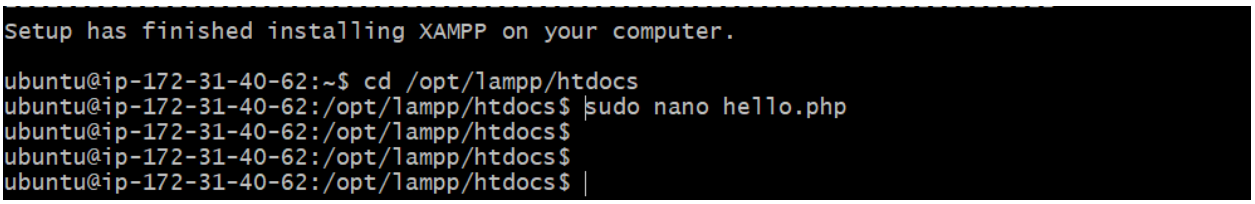
chmod +x ./xampp-linux-x64-7.2.14-0-installer.run

sudo ./xampp-linux-x64-7.2.14-0-installer.run



Enter the xampp folder and create the php file for the reference. (hello.php)

Commands: $ cd /opt/lampp/htdocs $ sudo nano hello.php



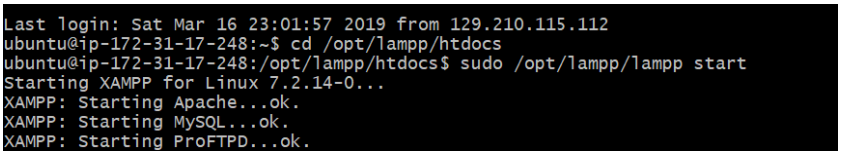
We then created the image of instance 1

Command: ‘aws ec2 create-image --instance-id i-07e50a14e896c66eb --name "myinstanceimage" --no-reboot’

We then created another instance with image id we got from instance 1.

We started the XAMPP server on the instance 2 as well.

Command: ‘sudo /opt/lampp/lampp start ‘



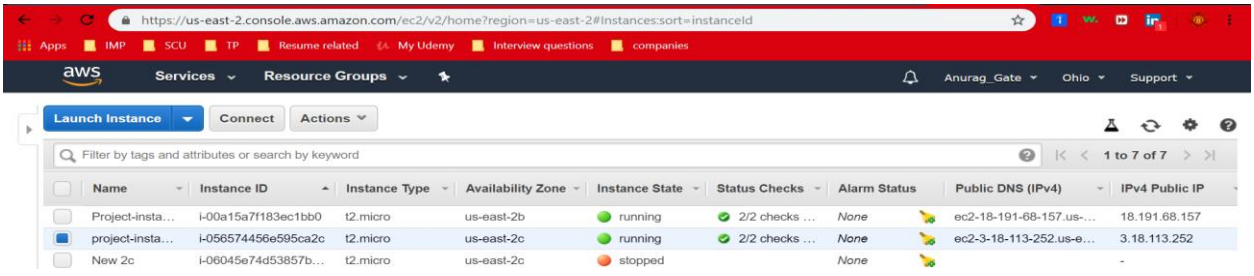
Below we can see the creation of the load balancer.

```
MINGW64:/

Anurag Gate@DESKTOP-L1OE7UH MINGW64 /
$ aws elbv2 create-load-balancer --name LB_Anurag --subnets subnet-9e09ded2 subnet-6427481e --security-groups sg-08108ad1a9f744bf5

An error occurred (ValidationError) when calling the CreateLoadBalancer operation: The load balancer name 'LB_Anurag' can only contain characters that are alphanumeric char
acters and hyphens(-)

Anurag Gate@DESKTOP-L1OE7UH MINGW64 /
$ aws elbv2 create-load-balancer --name lb-Anurag --subnets subnet-9e09ded2 subnet-6427481e --security-groups sg-08108ad1a9f744bf5
{
    "LoadBalancers": [
        {
            "LoadBalancerArn": "arn:aws:elasticloadbalancing:us-east-2:469776056688:loadbalancer/app/lb-Anurag/6edd016a075d6f14",
            "DNSName": "lb-Anurag-1093066015.us-east-2.elb.amazonaws.com",
            "CanonicalHostedZoneId": "Z3AADJGX6KTTL2",
            "CreatedTime": "2019-03-17T00:07:40.320Z",
            "LoadBalancerName": "lb-Anurag",
            "Scheme": "internet-facing",
            "VpcId": "vpc-a2f9f7ca",
            "State": {
                "Code": "provisioning"
            },
            "Type": "application",
            "AvailabilityZones": [
                {
                    "ZoneName": "us-east-2b",
                    "SubnetId": "subnet-6427481e"
                },
                {
                    "ZoneName": "us-east-2c",
                    "SubnetId": "subnet-9e09ded2"
                }
            ],
            "SecurityGroups": [
                "sg-08108ad1a9f744bf5"
            ],
            "IpAddressType": "ipv4"
        }
    ]
}
```

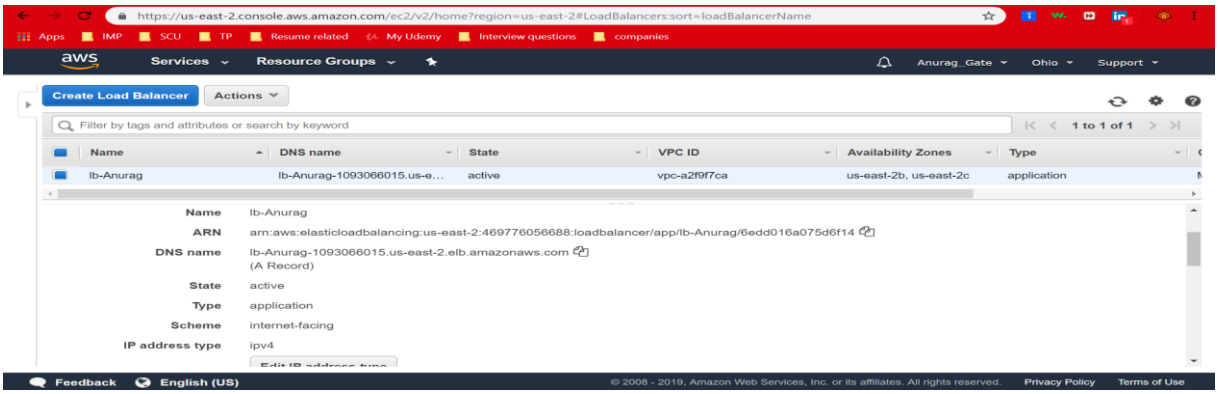
```
Anurag Gate@DESKTOP-L1OE7UH MINGW64 /
$ aws elbv2 create-target-group --name my-targetsanurag --protocol HTTP --port 80 --vpc-id vpc-a2f9f7ca
{
    "TargetGroups": [
        {
            "TargetGroupArn": "arn:aws:elasticloadbalancing:us-east-2:469776056688:targetgroup/my-targetsanurag/cbb37117ab646bc7",
            "TargetGroupName": "my-targetsanurag",
            "Protocol": "HTTP",
            "Port": 80,
            "VpcId": "vpc-a2f9f7ca",
            "HealthCheckProtocol": "HTTP",
            "HealthCheckPort": "traffic-port",
            "HealthCheckEnabled": true,
            "HealthCheckIntervalSeconds": 30,
            "HealthCheckTimeoutSeconds": 5,
            "HealthyThresholdCount": 5,
            "UnhealthyThresholdCount": 2,
            "HealthCheckPath": "/",
            "Matcher": {
                "HttpCode": "200"
            },
            "TargetType": "instance"
        }
    ]
}
```

```
Anurag Gate@DESKTOP-L1OE7UH MINGW64 /
$ aws elbv2 register-targets --target-group-arn arn:aws:elasticloadbalancing:us-east-2:469776056688:loadbalancer/app/lb-Anurag/6edd016a075d6f14 --targets Id=i-056574456e595
ca2c Id=i-00a15a7f183ec1bb0

An error occurred (ValidationError) when calling the RegisterTargets operation: 'arn:aws:elasticloadbalancing:us-east-2:469776056688:loadbalancer/app/lb-Anurag/6edd016a075d
6f14' is not a valid target group ARN

Anurag Gate@DESKTOP-L1OE7UH MINGW64 /
$ aws elbv2 register-targets --target-group-arn arn:aws:elasticloadbalancing:us-east-2:469776056688:targetgroup/my-targetsanurag/cbb37117ab646bc7 --targets Id=i-056574456e5
95ca2c Id=i-00a15a7f183ec1bb0

Anurag Gate@DESKTOP-L1OE7UH MINGW64 /
$ aws elbv2 create-listener --load-balancer-arn arn:aws:elasticloadbalancing:us-east-2:469776056688:loadbalancer/app/lb-Anurag/6edd016a075d6f14 --protocol HTTP --port 80 --
default-actions Type=forward,TargetGroupArn=arn:aws:elasticloadbalancing:us-east-2:469776056688:targetgroup/my-targetsanurag/cbb37117ab646bc7
{
    "Listeners": [
        {
            "ListenerArn": "arn:aws:elasticloadbalancing:us-east-2:469776056688:listener/app/lb-Anurag/6edd016a075d6f14/57151dd62a353a6a",
            "LoadBalancerArn": "arn:aws:elasticloadbalancing:us-east-2:469776056688:loadbalancer/app/lb-Anurag/6edd016a075d6f14",
            "Port": 80,
            "Protocol": "HTTP",
            "DefaultActions": [
                {
                    "Type": "forward",
                    "TargetGroupArn": "arn:aws:elasticloadbalancing:us-east-2:469776056688:targetgroup/my-targetsanurag/cbb37117ab646bc7"
                }
            ]
        }
    ]
}
```

We can see 2 instances running in Ohio region



Here is the snapshot of Load balancer we created named ‘lb-anruag’



TECHNICAL PART 2

First, we create a folder in our machine and cd to that folder.

Then we run the command ‘eb init’ to initialize the elastic beanstalk. We then have to select the region in which we want to create the beanstalk. We then create a new application and name it.

```
C:\Users\Anurag Gate\Desktop>cd cloud-project-anurag

C:\Users\Anurag Gate\Desktop\cloud-project-anurag>eb init

Select a default region
1) us-east-1 : US East (N. Virginia)
2) us-west-1 : US West (N. California)
3) us-west-2 : US West (Oregon)
4) eu-west-1 : EU (Ireland)
5) eu-central-1 : EU (Frankfurt)
6) ap-south-1 : Asia Pacific (Mumbai)
7) ap-southeast-1 : Asia Pacific (Singapore)
8) ap-southeast-2 : Asia Pacific (Sydney)
9) ap-northeast-1 : Asia Pacific (Tokyo)
10) ap-northeast-2 : Asia Pacific (Seoul)
11) sa-east-1 : South America (Sao Paulo)
12) cn-north-1 : China (Beijing)
13) cn-northwest-1 : China (Ningxia)
14) us-east-2 : US East (Ohio)
15) ca-central-1 : Canada (Central)
16) eu-west-2 : EU (London)
17) eu-west-3 : EU (Paris)
18) eu-north-1 : EU (Stockholm)
(default is 3):

Select an application to use
1) cloud-project-anurag
2) [ Create new Application ]
(default is 1): 2
```

We then selected our platform as PHP and selected the default version

```
Enter Application Name
(default is "cloud-project-anurag2"): anurag-cloud-eb
Application anurag-cloud-eb has been created.

Select a platform.
1) Node.js
2) PHP
3) Python
4) Ruby
5) Tomcat
6) IIS
7) Docker
8) Multi-container Docker
9) GlassFish
10) Go
11) Java
12) Packer
(default is 1): 2

Select a platform version.
1) PHP 7.2
2) PHP 7.1
3) PHP 7.0
4) PHP 5.6
5) PHP 5.5
6) PHP 5.4
(default is 1):
Cannot setup CodeCommit because there is no Source Control setup, continuing with initialization
Do you want to set up SSH for your instances?
(Y/n): n
```

We have then created a new environment using the command ‘eb create - -scale 2’ and select the environment name and the DNS name.

We have selected the classic load balancer.

```
C:\Users\Anurag Gate\Desktop\cloud-project-anurag>eb create --scale 2
Enter Environment Name
(default is anurag-cloud-eb-dev):
Enter DNS CNAME prefix
(default is anurag-cloud-eb-dev):

Select a load balancer type
1) classic
2) application
3) network
(default is 2): 1
NOTE: The current directory does not contain any source code. Elastic Beanstalk is launching the sample application instead.
Do you want to download the sample application into the current directory?
(Y/n): n
Environment details for: anurag-cloud-eb-dev
  Application name: anurag-cloud-eb
  Region: us-west-2
  Deployed Version: Sample Application
  Environment ID: e-zseehtzmrk
  Platform: arn:aws:elasticbeanstalk:us-west-2::platform/PHP 7.2 running on 64bit Amazon Linux/2.8.7
  Tier: WebServer-Standard-1.0
  CNAME: anurag-cloud-eb-dev.us-west-2.elasticbeanstalk.com
  Updated: 2019-03-18 02:07:56.546000+00:00
Printing Status:
2019-03-18 02:07:55    INFO    createEnvironment is starting.
```

We have created a new bucket in our region using the cli command ‘aws s3 mb s3://anurag-cloud-eb --region us-west-2’.

The bucket name should be unique and it should be in the same region as the region of the beanstalk.

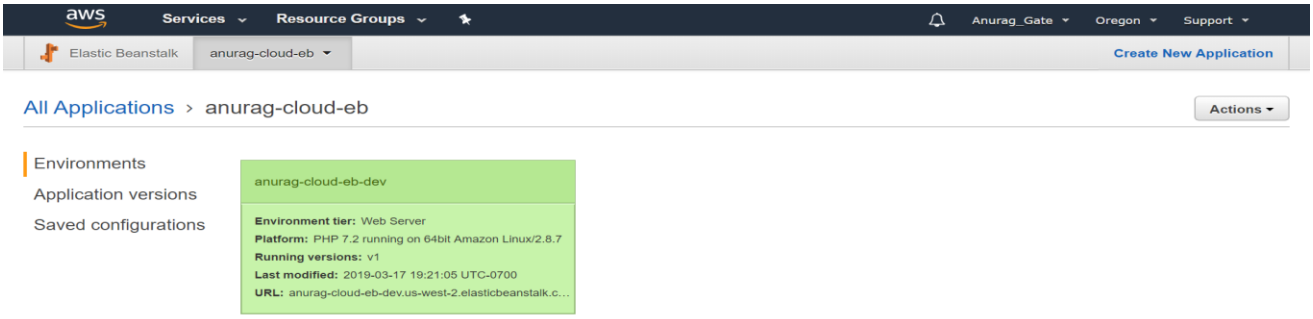
We have then copied the zip file containing my .php file in the bucket using the command ‘aws s3 cp "C:/Users/Anurag Gate/Downloads/php-code.zip" s3://anurag-cloud-eb’.

Then we have added our application to the elastic beanstalk using the following command :

‘aws elasticbeanstalk create-application-version --application-name anurag-cloud-eb --version-label v1 --source-bundle S3Bucket=anurag-cloud-eb,S3Key=php-code.zip –region us-west-2’

```
C:\Users\Anurag Gate\Desktop\cloud-project-anurag>aws s3 mb s3://anurag-cloud-eb --region us-west-2
make_bucket: anurag-cloud-eb

C:\Users\Anurag Gate\Desktop\cloud-project-anurag>aws s3 cp "C:/Users/Anurag Gate/Downloads/php-code.zip" s3://anurag-cloud-eb
upload: ..\..\Downloads\php-code.zip to s3://anurag-cloud-eb/php-code.zip

C:\Users\Anurag Gate\Desktop\cloud-project-anurag>aws elasticbeanstalk create-application-version --application-name anurag-cloud-eb --version-label v1 --source-bundle S3Bu
cket=anurag-cloud-eb,S3Key=php-code.zip

An error occurred (InvalidParameterCombination) when calling the CreateApplicationVersion operation: Unable to download from S3 location (Bucket: anurag-cloud-eb  Key: php-
code.zip). Reason: The bucket is in this region: us-west-2. Please use this region to retry the request

C:\Users\Anurag Gate\Desktop\cloud-project-anurag>aws elasticbeanstalk create-application-version --application-name anurag-cloud-eb --version-label v1 --source-bundle S3Bu
cket=anurag-cloud-eb,S3Key=php-code.zip --region us-west-2
{
    "ApplicationVersion": {
        "ApplicationVersionArn": "arn:aws:elasticbeanstalk:us-west-2:469776056688:applicationversion/anurag-cloud-eb/v1",
        "ApplicationName": "anurag-cloud-eb",
        "VersionLabel": "v1",
        "SourceBundle": {
            "S3Bucket": "anurag-cloud-eb",
            "S3Key": "php-code.zip"
        },
        "DateCreated": "2019-03-18T02:12:12.834Z",
        "DateUpdated": "2019-03-18T02:12:12.834Z",
        "Status": "UNPROCESSED"
    }
}

C:\Users\Anurag Gate\Desktop\cloud-project-anurag>
```

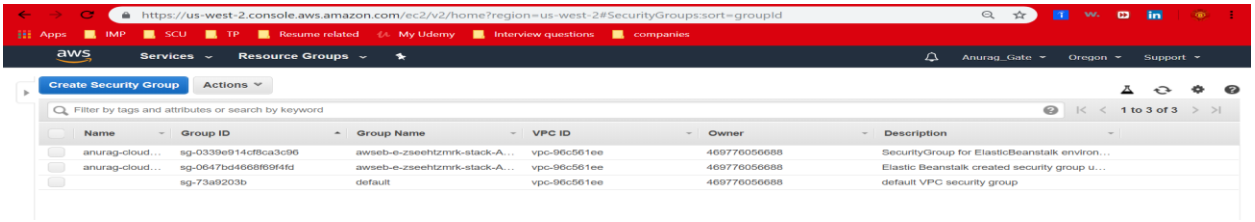
Here we can see the details of our application in the beanstalk with environment type, platform, version, modified date and time details and the URL to access it.



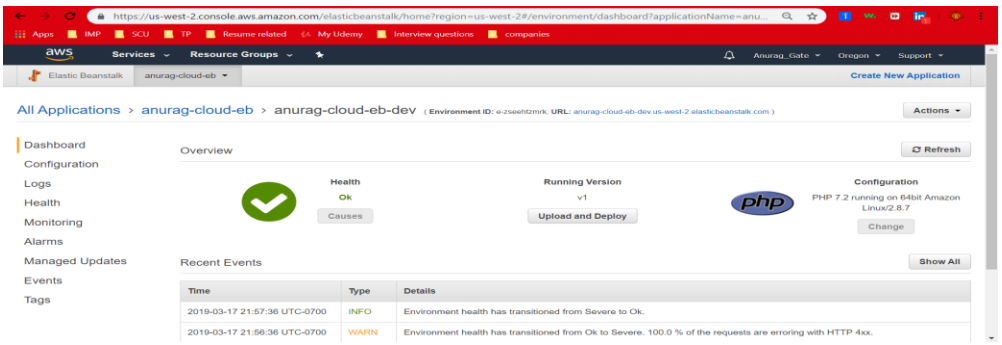
Here is the snapshot of the output of our .php file when we run the URL of our elastic beanstalk.

Hello World

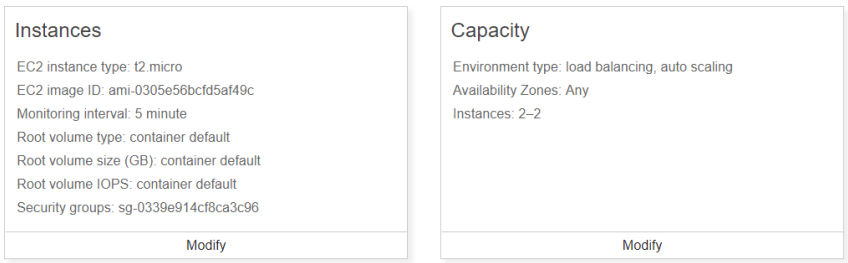
Here are the 2 security groups for our elastic beanstalk.



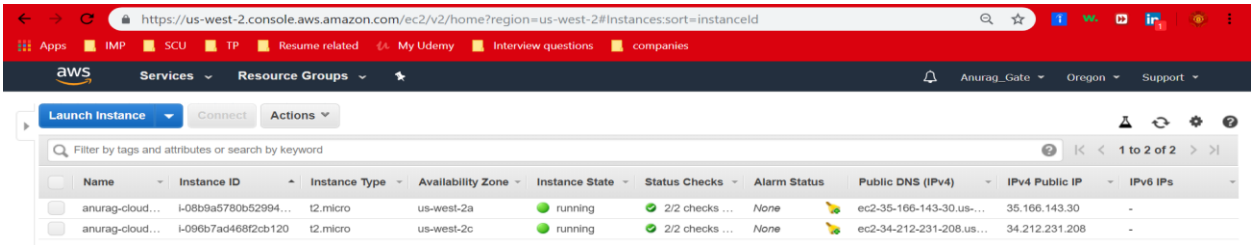
We can also check the health of our elastic beanstalk application.



Here this snapshot shows the number of running instances as 2 in our elastic beanstalk.



Here are the details of the instances we have for our application in the beanstalk.



Pros and Cons of using CLI over Console

Pros of CLI:

1. Command line interface is much faster than GUI or console as users only need to utilize the keyboard to navigate the interface
2. Command line can be used remotely with greater ease than GUIs as remote GUIs are bit sluggish.
3. For complex tasks, CLI can be scripted to automate things.

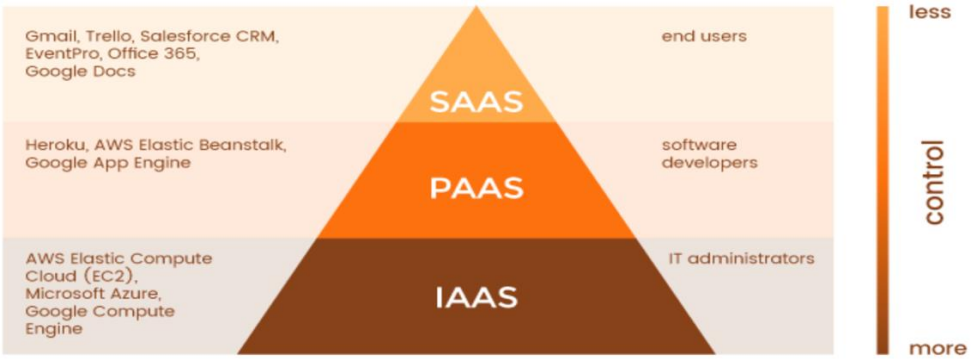
4. Using CLI over GUI takes a lot of less computer's system resources.

Cons of CLI:

1. It has a steeper learning curve for the beginners due to higher degree of memorization of commands and their complex syntax.
2. It is less user friendly than GUI for novice users.
3. CLI does not have the ability to show multiple things at once on 1 screen.
4. CLI is not a good option for certain tasks that involve graphical tasks or surfing the web or graphical applications.
5. Sometimes typing commands having long file names can be annoying without cut/paste support.

THEORY QUESTIONS

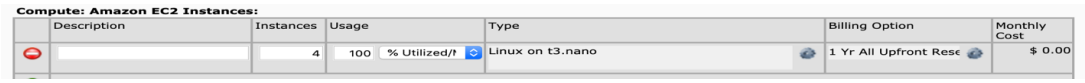
1) Breakdown of PaaS services between both providers:

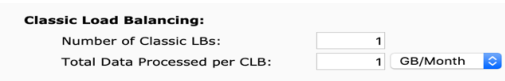


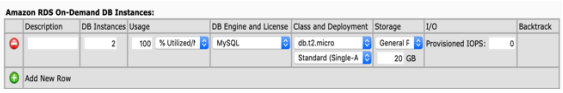
AWS Elastic Beanstalk (PaaS)	Google App Engine (PaaS)
we can deploy and manage applications in the AWS Cloud without having to learn about the infrastructure that runs those applications.	It's a platform to build apps and scale automatically. It supports popular languages. We can build our application in Node.js, Java,
Elastic Beanstalk supports applications developed in Go, Java, .NET, Node.js, PHP, Python, and Ruby. When you deploy your application, Elastic Beanstalk builds the selected supported platform version and provisions one or more AWS resources, such as Amazon EC2 instances, to run your application.	For monitoring, logging & diagnostics, Google Stackdriver gives you powerful application diagnostics to debug and monitor the health and performance of your app Application Versioning: to easily host different versions of your app, easily create development, test, staging, and production environments
Price: There is no additional charge for AWS Elastic Beanstalk. You pay for AWS resources (e.g. EC2 instances or S3 buckets) you create to store and run your application. You only pay for what you use, as you use it; there are no minimum fees and no upfront commitments.	Price: App Engine applications run as instances within the standard environment or the flexible environment. Instances within the standard environment have access to a daily limit of resource usage that is provided at no charge defined by a set of quotas. Beyond that level, applications will incur charges

2) As per the following configurations, the calculations have been made.

- **1 Load Balancer**
- **4 EC2 instances (We chose t3.nano as it was cheaper than the standard configuration)**
- **2 DB instances**

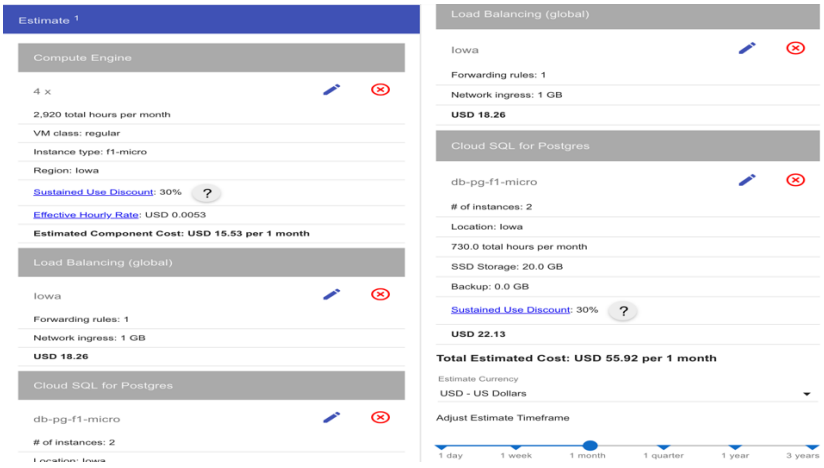






Cost of hosting for 1 year with these configurations (with free-tier discount) : $173.5

Even without free-tier discount AWS is still cheaper with cost of $573.72 than GCP's $671.04.



3) GCP Security Models:

All data is encrypted in transit between Google, the customers, and data centers by default; as well as the data in all of the Cloud Platform services. The data stored on persistent disks is encrypted under 256-bit AES and each encryption key is also encrypted with a set of regularly changed master keys.

Because Google has relationships with some of the biggest ISPs in the world, this helps improve the security of your data in transit as it means less hops across the public internet.

Google Cloud Identity and Access Management (Cloud IAM) was launched in September 2017 to provide predefined roles that give granular access to specific Google Cloud Platform resources and prevent unwanted access to other resources. This is similar to AWS IAM.

AWS Security Models:

Just like Google Cloud Platform, the AWS platform has a security model that has been improved upon for over a decade. Some of their security features include:

All data is encrypted in transit between AWS, the customers, and data centers; as well as the data in all of the AWS cloud. The data stored on EC2 instances is encrypted under 256-bit AES and each encryption key is also encrypted with a set of regularly changed master keys.

Network firewalls built into Amazon VPC, and web application firewall capabilities in AWS WAF let you create private networks, and control access to your instances and applications.

Flexible key management options, including AWS Key Management Service, allowing you to choose whether to have AWS manage the encryption keys or enable you to keep complete control over your keys.

Dedicated, hardware-based cryptographic key storage using AWS CloudHSM, allowing you to satisfy compliance requirements.

AWS Identity and Access Management (IAM), AWS Multi-Factor Authentication, and AWS Directory Services allow for defining, enforcing, and managing user access policies.

AWS has audit-friendly service features for PCI, ISO, HIPAA, SOC and other compliance standards.

Although GCP is a late entrant in the market, it pulls ahead in security as it encrypts all of its data and communication channels by default, including traffic between data centers.

AWS' Relational Database Services makes data encryption an option but it requires configuration if the service spans multiple availability zones. The traffic between data centers is also not encrypted by default by AWS.

4) GCP Docker Service: Google Kurbernetes Engine is a unique compute offering by GCP that is an abstraction over Compute Engine. GKE may be categorized more generally as Container as a Service (CaaS) offering, sometimes coined Kubernetes as a Service (KaaS), which allows customers to easily run their Docker containers in a fully managed Kubernetes environment.

Moreover, Cloud Build is a service that executes your builds on Google Cloud Platform infrastructure. Cloud Build can import source code from Google Cloud Storage, Cloud Source Repositories, GitHub, or Bitbucket, execute a build to your specifications, and produce artifacts such as Docker containers or Java archives. Cloud Build executes your build as a series of build steps, where each build step is run in a Docker container.

AWS: provides support for both Docker open-source and commercial solutions. There are a number of ways to run containers on AWS, including Amazon Elastic Container Service (ECS) is a highly scalable, high performance container management service. Amazon Elastic Container Service for Kubernetes (EKS) makes it easy for you to run Kubernetes on AWS. AWS Fargate is technology for Amazon ECS that lets you run containers without provisioning or managing servers. Amazon Elastic Container Registry (ECR) is a highly available and secure private container repository that makes it easy to store and manage your Docker container images, encrypting and compressing images at rest so they are fast to pull and secure. AWS Batch lets you run highly-scalable batch processing workloads using Docker containers

Pros for AWS:

- It has multiple ways to support Docker including Fargate which is a technology for Amazon ECS that lets you run containers in production without deploying or managing infrastructure.
- Docker EE provides a platform for enterprise container management and security that is optimized for Amazon EC2 environments. Moreover, it has more user-friendly interface than GCP.

Con for AWS:

- GCP has more through documentation to deploy Docker. By deploying containers on Compute Engine, you can simplify application deployment while controlling your VM infrastructure.
- Manage VMs that are running containers in the same way you would treat any other VM when configuring and managing your Compute Engine infrastructure.
- Create scalable services using managed instance groups running containers, which offer features like autoscaling, autohealing, rolling updates, multi-zone deployments and load balancing.

5) GCP: The Google Cloud SDK offers a variety of command line tools to interact with GCP, namely:

- gcloud — GCP's primary CLI
- gsutil — CLI to interact with Google Cloud Storage
- bq — CLI to interact with Google BigQuery
- kubectl — Kubernetes Engine's CLI

The interactive environment also supports standard bash features like:

- intermixing gcloud and standard bash commands
- running commands like cd and pwd, and set/use shell variables across command executions
- running and controlling background processes
- TAB-completing shell variables, and much more!

For example, you can assign the result of the command to a variable and later call this variable as an input to a different command:

The gcloud command-line interface is a tool that provides the primary CLI to Google Cloud Platform. You can use this tool to perform many common platform tasks either from the command-line or in scripts and other automations.

- You can use the gcloud CLI to create and manage: Google Compute Engine virtual machine instances and other resources, Google Cloud SQL instances, Google Kubernetes Engine clusters, Google Cloud Dataproc clusters and jobs among other services.
- In addition to running gcloud CLI commands from the command line, you can also run them from scripts or other automations — for example, when using Jenkins to drive automation of Google Cloud Platform tasks.

AWS: The AWS Command Line Interface (AWS CLI) is an open source tool that enables you to interact with AWS services using commands in your command-line shell. With minimal configuration, you can start using functionality equivalent to that provided by the browser-based AWS Management Console from the command prompt in your favorite terminal program:

Linux shells – Use common shell programs such as bash, zsh, and tsch to run commands in Linux, macOS, or Unix.

Windows command line – On Windows, run commands in PowerShell or at the Windows command prompt.

Remotely – Run commands on Amazon Elastic Compute Cloud (Amazon EC2) instances through a remote terminal such as Putty or SSH, or with AWS Systems Manager.

All IaaS (infrastructure as a service) AWS administration, management, and access functions in the AWS Management Console are available in the AWS API and CLI. New AWS IaaS features and services provide full AWS Management Console functionality through the API and CLI at launch or within 180 days of launch.

The AWS CLI provides direct access to the public APIs of AWS services. You can explore a service's capabilities with the AWS CLI, and develop shell scripts to manage your resources. Or, you can take what you learn to develop programs in other languages by using the AWS SDKs.

6) Answer: A cloud service level agreement (SLA) is the agreement provided by the cloud service provider to the customers which ensures a minimum level of service is maintained.

If cloud providers do not achieve and maintain the SLA, then user is eligible for a credit from providers based on downtime percentage. A downtime is the total number of minutes that are the part of maximum allotted minutes to have no VM connectivity in a specific region. The formula can be derived like:

Monthly uptime% = (Max available minutes – Downtime) / Max available minutes * 100

Amazon AWS: Amazon AWS offers SLA guaranteeing a monthly uptime of 99.99%. For less than 99.99% and greater than 99.0%, a user receives 10% credit. A downtime less than 99% and greater than 95%, a user receives a credit of 30% & a downtime for less than 95%, a user could receive 100% credit.

Monthly Uptime Percentage	Service Credit Percentage
Less than 99.99% but equal to or greater than 99.0%	10%
Less than 99.0% but equal to or greater than 95.0%	30%
Less than 95.0%	100%

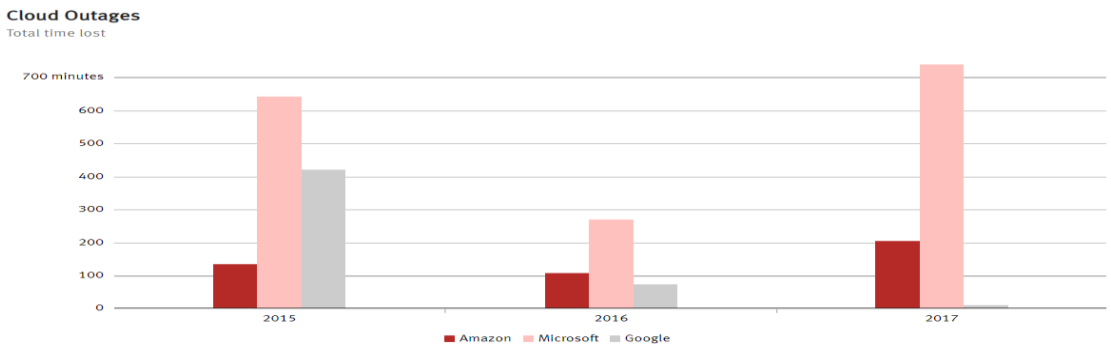
Microsoft Azure: Similarly, like Amazon AWS, Microsoft Azure offers SLA guaranteeing a monthly uptime of 99.99%. For less than 99.99% and greater than 99.0%, a user receives 10% credit. A downtime less than 99% and greater than 95%, a user receives a credit of 25% & a downtime for less than 95%, a user could receive 100% credit.

MONTHLY UPTIME PERCENTAGE	SERVICE CREDIT
< 99.99%	10%
< 99%	25%
< 95%	100%

Google Cloud: Google compute engine offers SLA guaranteeing a monthly uptime of 99.99%. For less than 99.99% and greater than 99.0%, a user receives 10% credit. A downtime less than 99% and greater than 95%, a user receives a credit of 25% & a downtime for less than 95%, a user receives 50% credit.

Monthly Uptime Percentage	Percentage of monthly bill for the respective Covered Service in the Region affected which did not meet SLO that will be credited to future monthly bills of Customer
99.00% - < 99.99%	10%
95.00% - < 99.00%	25%
< 95.00%	50%

Cloud downtime comparison of Amazon, Microsoft and Google in the years 2015,2016,2017.



7) Amazon EC2: It is able to both auto-scale and resize a virtual machine. Using AWS auto scaling, it is easy to setup an application scaling for multiple resources across multiple services. This task consists in keeping up a certain number of instances, starting and terminating on demand. Amazon auto-scaling is effective because it has an ability to maintain VMs healthy and restart them when

needed. In the case of Resizing, there are lot of things involved like virtualization, platform and networking issues so it is better to backup VM and create a new instance. AWS offers auto scaling at no additional charge based on scaling plans that the user defines for all the resources used by his applications.
Microsoft Azure: It's auto-scaling works through availability groups in which VMs can be added. A user can set the number of instances to scale up or down and also, he can choose a criterion i.e. CPU workload or queue size. All the instances in Azure can be easily resized by using Azure web-interface or PowerShell commands. Azure offers auto scaling per app or as part of platforms that manage groups of apps or groups of VMs.

Google CE: It provides horizontal auto-scaling by adding or removing new instances in a managed group of VMs. They are created from the same template of instance. In GCE, VM may adjust its capacity by custom criterion. In GCE, any instance can be resized using console, Gcloud or API butt a user must stop the machine first. GCP offers auto scaling only within the context of its managed instance groups platform. Autoscaling policies of Google include scaling based on CPU utilization, load balancing capacity and stack-driver monitoring metrics.

8) Feature comparison:

1. **Microsoft Azure** supports SSE which can be set up in Azure portal. It does not yet have the KMS to let the user create and manage the keys.
   Amazon S3 gives user an option to encrypt the data server using S3 managed SSE-S3 or SSE-KMS. Eventually the data is encrypted with AWS-256. Amazon also permits client-side encryption.
   On **Google cloud** the user's data Is encrypted using AES-256 and it remains in the same manner while it is at rest. While storing on google cloud, each file gets broken up into pieces and all pieces have their own encryption key. GC is the only service which encrypts by default. For others user has to turn on the encryption.
2. Talking about OS, a user can launch his own OS on **Azure** that is stored in VHD file. But when a user deletes his VM in Azure, the original VHD uploaded file also gets deleted.
   However, **Amazon** does not allow a user to launch his own VHD but allows to develop one with their own different OS offerings to save the image for the users. Its advantage is that if a user deletes the VM, he does not lose the image.

Cost comparison:

Microsoft Azure rates change on the basis of the amount of storage a user needs, where he is located and how often he wants to access the content.	Amazon S3 offers similar flexibility like Azure for scaling the storage. A user is charged for what he is using. There are no upfront or termination fees on S3. The difference is, Amazon S3 does not have multi-regional storage like Azure but it offers a middle-tier class between standard and archival storage.	Furthermore, Google cloud provides a good mixture of storage class options provided by Azure and Amazon making it the most scalable in the above cloud comparison. It also includes the multi-regional and regional options, a mid-range access option called 'nearline' and a glacier-like archival option.

Microsoft Azure

	LRS - Cool	LRS - Hot	GRS - Cool	GRS - Hot
First 50 TB per month	$0.0152	$0.0208	$0.0334	$0.0458
Next 450 TB per month	$0.0152	$0.0200	$0.0334	$0.0440
Over 500 TB per month	$0.0152	$0.0192	$0.0334	$0.0422

Amazon S3

Tier	Standard per gigabyte	Standard - Infrequent access per gigabyte
First 50 TB / month	$0.023	$0.0125
Next 450 TB / month	$0.022	$0.0125
Over 500 TB / month	$0.021	$0.0125

Google Cloud

	Multi-regional per GB	Regional per GB	Nearline per GB	Coldline per GB
General pricing	$0.026	$0.020	$0.010	$0.007
Tokyo pricing	N/A	$0.023	$0.016	$0.010

Microsoft Azure bills usage for operations and data write-retrieval. These operations are billed at every 10k transactions. Write and retrieval are billed per gigabyte.	Studying the Amazon S3 usage, pricing is complex based on what the user is doing and what type of storage class he is working on. Uploads to amazon S3 are free.	Google cloud storage billing way makes it simpler than complex Amazon S3. Google refers to data writes/uploads as 'ingress' and data retrievals/downloads as 'egress'.

Microsoft Azure

	Hot (LRS):	Hot (GRS):	Cool (LRS):	Cool (GRS):	Archive (LRS):	Archive (GRS):
Read:	Free	Free	$0.01	$0.01	$0.02	$0.02
Write:	Free	Free	Free	Free	Free	Free

Amazon S3

First 1GB per month	Free
Up to 10TB per month	$0.090 per GB
Next 40 TB per month	$0.085 per GB
Next 100 TB per month	$0.070 per GB
Next 350 TB per month	$0.050 per GB

Google Cloud

	Network ingress	Network egress</strong	Network egress (China)</strong	Network egress (Australia) </strong
0-1 TB</strong	Free	$0.12 per GB	$0.23	$0.19
1-10 TB</strong	Free	$0.11 per GB	$0.22	$0.18
10+ TB</strong	Free	$0.08 per GB	$0.20	$0.15

Upload and Download Cost comparison

	Upload	Download
Azure	Free	Free
Amazon S3	Free	$0.09 per GB (below 10TB)
Google Cloud Storage	Free	$0.11 per GB (below 10TB)

Support cost comparison

	Developer support	Business support
Azure	$29	$300
Amazon S3	$29	$100
Google Cloud	$150	Starts at $400

Result: I think Looking at all the features and costs mentioned above Amazon S3 is the winner.

9)

Amazon EC2	Microsoft Azure VM
AWS EC2 users can configure their own VMs or select pre-configured images.	Azure users choose VHD which is like Machine instances to create VM. VHD is pre-configured by Microsoft or the user.
AWS offers temporary storage which is allocated at the creation of the instance and destroyed at termination. AWS also provides block storage which can be attached to an instance. It supports relational and NoSQL databases	Azure offers temporary storage via D drive & block storage via page blobs for VMs. Supports relational databases, NoSQL and Big Data.
Amazon EC2 offers Virtual Private Cloud (VPC) which allows users to create subnets, route tables, private IP address ranges and network gateways.	Microsoft Azure offers virtual network (VNET) which allows users to create subnets, route tables, private IP address ranges and network gateways.
Amazon has a good relationship with open source community so more open source integrations are available including Jenkins and GitHub.	If a user is already using windows tools like VBS, SQL, active directory then azure offers native integration. User can use same account to sign in to office 365. **Azure is good for .net developers.**
AWS has recently added machine learning tools and features to target IoT to allow customers to build mobile apps. It also offers support for docker.	Azure offers Hadoop support with Azure HDInsight.

Cost comparison:

AWS EC2: It has a pay-as-you-go model where they charge a user on a hourly basis. Instances can be purchased by On-demand model, Reserved model and spot model.

Azure VM: Microsoft's pricing is also pay-as-you-go but they charge per minute resulting into a more exact pricing model. Azure also provides short term commitments with the option between pre-paid or monthly charges.

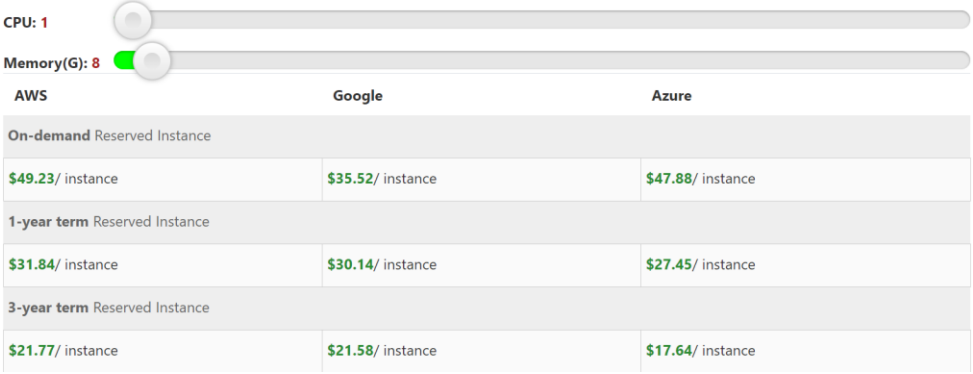
Amazon EC2	Google Compute engine
Technology used behind AWS EC2 VMs is Xen.	Technology used behind Google's cloud VM is KVM.
User can equip AWS EC2 instances with up to 128 vCPUs and 3904 GB of RAM	User can equip GC engine instances with up to 160 vCPUs and 3884 GB of RAM

EC2 instances are reserved within VPCs providing benefits of security groups and network ACLs. User can control incoming and outgoing traffic at instance level & work at subnet level to allow or deny specific IP addresses.	Google compute engine firewalls regulate the outgoing traffic from instances using the iptables. Google engine's Firewall comes under whitelist service.
ELB works with AWS's auto scaling and supports IPv4 and Ipv6, HTTP, TCP and logging.	Google compute engine also offers Load balancer in a similar manner but unlike AWS, it also allows balancing among regions, support content-based routing and does not need pre-warming.
EC2 provides Elastic Block storage for persistent storage which are offered in 3 types: Magnetic volumes, general purpose SSD and provisioned IOPS SSD volume.	Google compute engine offers persistent disk storage available as both standard (HDD) and solid-state (SDD).
EC2 offers SLA guaranteeing a monthly uptime of 99.95%. For less than 99% a user receives 30% service credit	Google compute engine offers SLA guaranteeing a monthly uptime of 99.95%. For anything less than 95% a user receives a 50% credit.

Cost comparison:

Amazon AWS EC2: It has a pay-as-you-go model where they charge a user on a hourly basis. Instances can be purchased by On-demand model, Reserved model and spot model.

Google Compute Engine: GCE machine types are charged for a minimum 10 minutes of use. After 10 minutes instances are charged in 1-minute increments. GCE also offers on-demand and sustained usage pricing models. For maximizing savings, GCE offers inferred instances like combined multiple, non-overlapping instances of the same type in the same time zone into single instance for billing.
Result: Looking at the above comparison of features and cost, according to me all of the 3 cloud providers support similar features and differ in cost with a very minimal change.
So, I would rank the VMs in following order: 1) Amazon EC2. 2) Microsoft Azure 3) Google compute engine



10) If you had to describe one really Cool feature of the Alternative Cloud Vendor what would that be and does AWS have anything like that or not?

Talking about some good feature, Google cloud offers a cloud machine learning engine, which helps ML engineers build models based on its open source TensorFlow deep learning library. Besides that, google also offers a whole host of off-the-shelf API for NLP, translation and computer vision.

In the similar manner, Amazon AWS launched the amazon machine learning service in 2015 to help developers create ML models.

References:

Business: https://cloud.google.com/files/esg-whitepaper.pdf

https://cloud.google.com/free/docs/map-aws-google-cloud-platform

https://www.datamation.com/cloud-computing/aws-vs-azure-vs-google-cloud-comparison.html

Q1. https://rubygarage.org/blog/iaas-vs-paas-vs-saas

https://cloud.google.com/appengine/docs/

Q2. https://calculator.s3.amazonaws.com/index.html

https://cloud.google.com/products/calculator/#id=d3b488ea-2295-4c23-94f1- 19fbad17fb6f

Q3. https://kinsta.com/blog/google-cloud-vs-aws/

https://www.crn.com/slide-shows/cloud/google-cloud-platform-vs-aws-which-provider-should-a-partner-pick-/3

Q4.https://medium.com/google-cloud/gcp-the-google-cloud-platform-compute-stack- explained-c4ebdccd299b

https://aws.amazon.com/docker/

Q5. https://cloud.google.com/blog/products/gcp/introducing-gcps-new-interactive-cli

Q6. https://docs.aws.amazon.com/cli/index.html

https://aws.amazon.com/compute/sla/

https://azure.microsoft.com/en-us/support/legal/sla/virtual-machines/v1_8/

https://cloud.google.com/terms/sla/

Q7. https://aws.amazon.com/autoscaling/

https://docs.microsoft.com/en-us/azure/cloud-services/cloud-services-how-to-scale-portal

https://cloud.google.com/compute/docs/autoscaler/

Q8. https://www.cloudberrylab.com/resources/blog/amazon-s3-azure-and-google-cloud-prices-compare/

https://cloud.netapp.com/blog/ebs-efs-amazons3-best-cloud-storage-system

Q9. https://www.infoworld.com/article/3237566/cloud-pricing-comparison-aws-vs-azure-vs-google-vs-ibm.html

https://www.computerworlduk.com/it-vendors/microsoft-azure-vs-amazon-aws-public-cloud-comparison-which-cloud-is-best-for-enterprise-3624848/

https://stackify.com/microsoft-azure-vs-amazon-web-services-vs-google-compute-comparison/