Cloud Summary

!Note: This is just a summary, so it has just the important information.

<< صل علي محمد >>

# Lecture 2 (AWS module 2 student guide)

– Cloud architecture : Is the process of applying cloud characteristics in a solution that uses cloud services and features.

– Cloud architect jobs:
1- Engages with decision makers to identify business goals and capabilities that need improvements
2- Works with delivery team that implement the solution to ensure that the technology features are appropriate
3- Ensure alignment between technology deliverables and business goals.

– Pillars of AWS well architected framework:
- دي اعمدة (خصائص) تعمل حسابك عليها (انك تطبقها) وانت بتعمل ال design solution بتاعك
- تقدر تقول عليها دليل التصميم الكويس

1- Operational excellence => The ability to run and monitor systems
=> You must be aware of how the system will be deployed, updated and operated.

2- Security => Apply security at all layers
=> Apply traceability, identity foundation, risk assessment and mitigation

3- Reliability => Recover quickly from infrastructure or service disruption

4- Performance efficiency => choose efficient resources and maintain their efficiency as demand change

5- Cost optimization => Eliminate unneeded expense

6- Sustainability => Understand your impact (it is the last pillar and it’s made to help organizations lower their environmental impact)
=> Maximize utilization of resources

Design best practices : -

1- Enable scalability: Ensure that your architecture handle changes in demand.
– Note: AWS cloudwatch helps you monitor to detect whether the total load across your fleet of servers has reached a specified threshold.

- بتراقب حالة ال resources بتاعتك عشان تاخد قرار اول لما توصل لل full capacity مثلا ممكن تنبهك وانت تاخد قرار وطبعا ظا مش احسن حاجة عشان كدا احنا مستغلناش طبيعة ال automicity بتاع ال cloud
- الافضل مثلا انك تخلي السيستم ي scale up or down automatically

2- Automate your environment: automate provisioning(توفير, بناء), termination(انهاء), and configuration of resources.

3- Treat resources as disposable

ودا ببساطة الي كنا بنتكلم عليه فوق ان انت تحاول تخلي كل حاجة dynamic علي اد متقدر

4- Use loosely coupled components

متخليش ال components بتاع حضرتك مرتبطة ببعض اوي زي ال full mesh

To apply this we use a service called Elastic load balancing

5- Design services not servers

استخدم الخدمات الي امازون بتوفرهالك ووفر وقت ومجهود علي نفسك

6- Choose the right database solution

7- Avoid single points of failure

بتبقا حاجة في ال design بتاعك لو حصلها مشكلة هتوقف السيستم كله

ممكن تتفادي كدا بان انت ت replicate الحاجة دي بحيث انها لو وقعت تروح علي النسخة الاحتياطية والشغل ميقفش

8- Optimize for cost

اشتغل علي ادك عشان توفر مش مال سايب هوا

9- Use caching (Amazon cloudFront)

10- Secure your entire infrastructure: Build security into every layer of your architecture

# Lecture 3 (AWS module 3 student guide)

– Amazon S3 (simple storage service):

ركز في الاختصارات انت عارف الي فيها كويس

** Object storage
** Data files (objects) stored in a bucket
** Maximum file size of a single object is 5TB
** All objects have a REST-accessible unique URL
** All objects has a key, version id, value ,meta data, subresources
** We identify an object from the key and version id
** value => content that you store
** metadata => user metadata (by you)
=> system metadata (by Amazon)

– S3 benefits:

1- Durability (ensures data is not lost)
(99.999999999) 11 nines (9 nines) after the dot
2- Availability (can access data when needed)
(99.99) 4 nines
3- Scalability
4- Security
5- Performance

– S3 use cases:

1- Store and distribute web content and media
2- Host static website
3- Datastore for communication and analytics
4- Backup and archive for critical data

– How to protect your S3 ?

— Newly created S3 are private and protected by default
— This considerations because of a lot of security or if you want to handle the access to your s3

1- Block public access
2- IAM policies
3- Bucket policies
4- Access control list (ACL)
5- S3 access points
6- Presigned URLs
7- AWS trusted advisor

- Encryption: encodes data with a secret key
- Types of encryption: 1- Server-side (The default option and managed be AWS)
2- Client-side (you take the responsibility of managing the process)

- Versioning:

ـلما بترفع فايل كذا مره كل مره بياخد version واخر version هو الي بيتعرض
ـطبعا دا بيحمي الفايلات بتاعتك لو مثلا مسحتها بالغلط او حصل اي مشكلة
ـوبيخليك تتابع ال versions لو مثلا حبيت ترجع ل version معين او حبيت تعرف حاجة انت كنت عملها زمان
– لو عملت enable of versioning مش هتعرف تعمله disable تاني بس تقدر تعمله suspend

S3 Storage classes:

1- S3 standard => Frequently access data
2- S3 standard IA => Infrequently access data (lower cost solution) (several zones in the same region)
3- S3 one zone IA => the difference is the term (one zone) means that it is in just one availability zone
4- S3 glacier or deep archive => rarely accessed data

# To manage this lifecycle to be automatically configured use Amazon lifecycle Policy

# S3 Pricing: The following is not free:
1- GBs (Gigabytes) of object store
2- Transfer out to other region or to the internet
3- PUT, COPY, LIST, GET, POST, SELECT, lifecycle transition data retrieval requests

اي حاجة غير ال delete , cancel عليها فلوس

– The following options are free:
1- Transfer in from the internet or in the same region
2- Transfer out to amazon cloudFront (cache service)
3- DELETE and CANCEL requests

# Techniques used to move objects to S3:

1- Multipart upload

ببساطة بتقسم الفايل لفايلات صغيرة وبعدين تجمعها تاني لما توصل

- Minimum 5MB , Maximum 5TB
- Benefits:

1- Increase throughput
2- Quick recovery from network issues
3- Ability to pause and resume uploads

2- Amazon S3 transfer acceleration

- Use cloudFront edge location

3- AWS SnowBall: peta-byte scale data transport option
4- AWS SnowMobile: exa-byte scale data transport option

– Services maybe not available in some regions, and it might not have the same cost

# Lecture 6 (Google cloud)

!Note: This section has the commands and some basic info from the lectures and the doctor video.

– Three models of cloud :
1- Infrastructure as a service (IAAS) => Google engine
=> More flexibility
=> flexibility comes with additional cost

2- Platform as a service (PAAS) => App engine
=> More scalability

3- Kubernetes (containers) => offers flexibility of IAAS and scalability of PAAS

# Containers are configurable, self-contained, and ultra-portable

– بص يمعلم ال container دا عبارة عن حاوية او صندوق بتحط فيه ال project بتاعك وتقدر ترنه من عليه كانك رافعه علي سيرفر بالظبط
– طبعا تقدر تحط اي نوع من الداتا مثلا ممكن تحط الداتابيز بتاعتك عليها او مجرد فايلات css , html , ...
– ال containers دي بنحطها في حاوية أكبر اسمها cluster وبكدا نكون خرجنا من كوكب docker ورحنا كوكب kubernetes الي بتخليك تقدر ت manage ال cluster

– تعالا بقا نبدا نعمل حاجة من الاول : 1- هتعمل فايل بايثون مثلا في اي حاجة
2- هتعمل docker file for configurations وهتوضح فيه:
1- OS image and version of Python
2- How to install Python
3- How to run app
### These files is on cloud build or docker (in the lecture they use cloud build)

– كدا احنا جاهزين نعمل ال container ودي الاوامر الي هنكتبها
– docker build -t py-server
-t => tag (name of the container)
– docker run -d py-server
-d => detached mode (run in the background)

–الف مبروك كدا عملنا الكونتينر بتاعنا وشغلناه

- عايزين بقا نعمل ال cluster الي هنحط فيه ال containers العملية دي اسمها bootstrapping

– gcloud container clusters create c1 --zone us-central1-a

– كدا عملنا ال cluster واسمه c1 وحددنا ال zone الي هيشغل فيها ودا optional

##When you deploy containers on nodes you use a wrapper called a Pod

– اعمل حسابك بقا طول محنا شغالين مع kubernetes الاوامر بتبدا ب kubectl

– احنا جربنا نرن ال container قبل منحطه في ال cluster دلوقتي هنرنه من خلال ال cluster

– ال container داخل ال cluster بتقدر تتحكم فيه من خلال Pod

– الاول محتاجين نعمل حاجة اسمها deployment ودي بتخليك تقدر تعمل كذا نسخة من ال Pods بتعتك

– kubectl create deployment dep0 --image nginx --replicas 3

كدا عملنا deployment من ال image الي اسمها nginx وخدنا ليها 3 نسخ ونقدر بعد كدا نرن

– kubectl run dep0 --image=nginx

– kubectl get pods => to get current pods

– دلوقتي احنا عايوين نخلي الناس تعرف تعمل access علي ال pod من خلال الانترنت فبنعمل حاجة اسمها loadbalancer

– kubectl expose deployments dep0 --port=80 --type=LoadBalancer

## Kubernetes creates a Service with a fixed IP for your Pods

## Service: is an abstraction which defines a logical set of Pods and a policy by which to access them.

– طبقة فوق ال pods عشان ت manage عملية ال access

– kubectl get services => to get the current services

دي شوية اوامر ابلعها معلش

– kubectl dep0 scale --replicas=3 => to scale a deployment

– kubectl dep0 autoscale dep0 --min=10 --max=15 --cpu=80

-- min = 10 => the minimum number of pods that should be running for the deployment or replicaset

-- max = 15 => the maximum ~ ~ ~ ~

-- cpu = 80 => maximum cpu utilization

– kubectl get pods -l "app=nginx" => get a config file

– kubectl get pod pod0 -o yaml => get a yaml file

– kubectl apply -f => to apply changes you made

– kubectl get replicasets

– kubectl get deployments

{{ Video commands }}

– docker run -p 8080:80 -d --name web0 nginx

– نغس الي عملناه فوق بس فيه كذا اوبشن زيادة

- p => port
- 8080 => our machine port
- 80 => nginx port
- name => define the name of the container

– docker ps => to show the containers

– docker executer -it web0 bash => open interactive(-it) bash
# in the interactive bash we write the following
– find / -name html => get the path of the html file we have

– docker build -t myweb0 **.** => this dot means all files in the folder / container

– gcloud container clusters get-credintials c1 --region --project gkebunk => connect the container to the cluster c1

<< صل علي محمد >>