**OVERVIEW**

* Các loại cloud:
  + Private cloud: sử dụng riêng tư bởi một tổ chức, thường được tổ chức sử dụng để bảo mật về thông tin. Ví dụ cho mô hình này là Rackspace
  + Public cloud: Sở hữu bởi bên thứ ba, và có tới 6 lợi thế khi sử dụng
  + Hybrid cloud: là sự kết hợp giữa private và public, có thể control các vấn đề nhạy cảm tối ưu về tiền
* 5 tính chất của cloud computing
  + On-demand: toàn bộ tài nguyên mà người dùng sử dụng sẽ không bị aws can thiệp
  + Broad network access: có thể được định nghĩa tùy thích bởi người dùng
  + Multi tenancy and resource pooling: có thể share nhiều người dùng với cùng một tài nguyên và ứng dụng nhưng vẫn đảm bảo về bảo mật và các chính sách
  + Rapid elasticity and scalability: khả năng tương thích và scale cao
  + Measured service: tính toán chính xác số tiền sử dụng của người dùng
* Các loại cloud computing:
  + IaaS: cung cấp các khối về IT, về network, data storage và người dùng có thể configure, dễ dàng chuyển đổi. Ví dụ các cloud hiện nay
    - Được configure về OS, middleware, runtime, data, applications
    - Ví dụ: EC2,
  + PaaS: Không thể quản lý về infrastructure, nhưng vẫn được quản lý về các dữ liệu, và ứng dụng,… Ví dụ heroku.
    - Được quản lý data và cũng như applications
    - Ví dụ Beansstalk
  + SaaS: thường là sản phẩm và mình chỉ quản lý sản phẩm đó
    - Không được configure gì thêm chỉ thừa hưởng và sử dụng application
    - Ví dụ: Rekognitiion for ML
* **Region** là một cluster về data center, khi đã khởi tạo dữ liệu tại một region thì không thể chuyển trực tiếp sang một region khác mà phải thực hiện khởi tạo lại tại region mới. Tuy nhiên không phải mỗi region đều có chung service có chỗ có này có chỗ có kia.
* **Availability zones** được hiểu như là một tập con của Region. Mỗi AZ là một hoặc nhiều data centers. Mỗi data center như là một server có đầy đủ mọi thứ và được connect private với nhau. 1 region có tối đa 6 AZ, và thấp nhất 2 AZ và thấp nhất là 2 AZ
* **Edge:**
* Dịch vụ Global: IAM, route 53, CloudFront, WAF, dịch vụ chỉ theo vùng: EC2 Elastic Beanstalk, Lambda,…

**IAM SECTION**

* Users:
* Groups: chứa nhiều users trong cùng một policies
* Policies: các quyền của users hoặc groups được định nghĩa thông qua file json
* Permission:
  + Policy describe some resource that user or role can be accessed
  + Using priciple: define the service that user need to use, don’t allow all
  + Pricipal: account/user/role to which this policy applied to
* Roles:
* Security: MFA + Password Policies
* AWS CLI: cần phải create access key và chỉ sử dụng với IAM user
* AWS SDK: quản lý AWS service thông qua một ngôn ngữ, ví dụ Pytnon thì có module boto3
* Access keys: sử dụng cho AWS CLI và SDK
* Audit: Dùng để giám sát users sử dụng các tài nguyên gì trong khoảng thời gian, với IAM Credential Reports sẽ dùng để xem các chỉnh sửa về mọi thứ của user còn IAM Access Advisor sẽ theo dõi các service mà user truy cập.
* IAM security tool is IAM Credentials Report

**EC2**

* Khi build image in EC2 builder thì nên chọn manual để có thể cài đặt thêm một số gói cần thiết và cũng như các thứ tự cài đặt và các thứ tự cần kiểm trả máy hoạt động vd như test reboot,…
* Khi tạo image builder cần phải có tạo role EC2 để có thể tạo được
* User data is just run once times, when the ec2 instance start, after that it’s never to be done again
* ***EC2 instance type***: example m5.2xlarge
  + m: instance class
  + 5: generation of instance (AWS improves them over time)
  + 2xlarge: size within the instance class
* Types – General Purpose:
  + Workload such as web servers or code repositories
  + Balance between compute, memory, networking
* Type – Compute Optimized
  + Require high perfformance process
  + Case using: Media transcoding, web server high perfomance, machine learning, dedicated gaming server
* Type – Memory Optimizeed
  + Fast perfomance for workloads that process large data in memory
  + Use caszse: high perfomance, relational / non relational database
  + Distributed web scale cacche stores, application performing real time processing of big unstructured data
* Type: Storage Optimized
  + Using in case require high, sequential read and write access to large data on local storage
  + Use cases: high frequency online transaction processing systems, relational & NoSQL database, cache for in memory database (Redis), data warehousing,
* ***Security Groups: firewall of ec2 instance***
* Inbound traffic that means the traffic from outside into the EC2 instance, and the Outbound traffic is opposite.
* If your application is **not accessible (time out)** 🡪 **security group issue**
* If your application gives a **“connection refused” error**, 🡪 **application error** or it’s not launched
* By default, all inbound will be blocked and outbound will be authorised
* One ec2 can be attached many security groups
* SSH – 22, FTP – 21, SFTP – 22, HTTP – 80, HTTPS – 443, RDP – 3389 – login into a windows instance
* When we need attach IAM role on console, we need to create role for that and then click button actions 🡪 securtiy 🡪 modify IAM roles.
* ***Instance Purschasing Options***
  + On-demand 🡪 pay be second
    - Linux or Win billing per second, after the first minute
    - All other OS – billing per hour
    - Highest cost, no long term commitment
    - Use case: short term and un interrupted workloads
  + Reverse (1 – 3 years) 🡪 long workload
    - 72%T discount compared to on-demand
    - Reverse Period – 1 year or 3 year ( that means how many resource using in that time)
    - When don’t use it can buy or sell in Reserve Instance Marketplace
  + Saving plans (1-3 years) 🡪 long workload, commitment amount of usage
    - Commit usage ($10/hour for 1 or 3 years)
  + Spot instances 🡪 short workloads, cheap, can lose instance (less reliable)
    - The most cost efficient instance
    - Useful for workload that are resilient to failure
    - Not suitalbe for critical jobs or database
  + Dedicated host 🡪 book an entire physical server, can be controlled instance placement
    - Allow address compliance requirements and use your existing server bound software licenses (per socket, per core, perVM sofware licesens)
    - Purchasing options: on demand, reserved
    - The most expensive options
    - Bring your own license
    - Companies that have strong regulatory or compliance needs
  + Dedicated instances 🡪 no other customers will share your hardware
    - Instance run on hardware that’s dedicated to you
    - May share hardware with other instance in same account
    - No control over instance placement
  + Capacity Reservation
    - Reserve on demand capacity in specific AZ
    - Always have acccess to ec2 capacity
    - No time commitment, bo billing discount
    - Combine with regional reserved instance and saving plans
    - Suitable for short term, uninterrupted workloads that needs to be in a specific AZ
* ***Spot Instance Request:***
* We need to create the max price that ec2 instance can be followed and then terminate the spot instance when the compute in ec2 instance over the threshold
* Request about the maximum price and maximum spot instance, AMI, type….
* Two type requests: one time request and persistent request. Persistance request 🡪 spot instance just follow the reuqest in the first time and then recreate new spot instance like that. One time request just using in first time and then closed
* Only cancel spot instance request: open, active, disable
* Canceling a Spot request does not terminate instances
* You must first cancel a Spot Rqeuest, and then terminate the associated Spot instances
* ***Spot Fleet = set of Spot Instance + (optional) On-demand Instances***
* The spot fleet will try to meet the target capacity with price constraints
* Stategies to allocate Spot Instances
  + lowestPrice: cost optimization, short workload
  + diversified: great for availability, long workload
  + capacityOptimized: optimal capacity for the number of instance
* Spot Fleets allows to automatically request Spot Instance with the lowest price
* ***Solutions Architect Associate Level:***
* Elastic IP that’s means public IPv4 that cannot delete. You can only have 5 Elastic IP in your account (if you want more, contact AWS). If you don’t want to use Elastic IP, you can use random IP public and then register a DNS name to it. Or another way, can use Load Balancer and don’t care about the public IP
* *Placement Group:*
  + Cluster: packs instances closse together inside a Availability Zone, help to communicate between node to node is low latency
    - Pros: great network
    - Cons: if the rack fails, all instance fails at the same time
    - Use case: big data job, application needs low latency and high network throughput
  + Partition: spreads your instance across logical partitions about the hardware,…
  + Spread: strictly places a small instance
    - Pros: span across AZ, reduce risk, instance are on different physical hardware
    - Cons: limit to 7 instance per AZ per placement group
    - Use case: application need to maximize high availability or critical application
* **EC2 Nitro** that is the new virtualization technology
  + Allow performance better: Networking (enhanced networking, HPC, IPv6), high speed EBS (64K IOPS, non Nitro has max 32K IOPS)
  + Better underlying security
  + Instance types examples: virtualized: c5, c5a, c6g,… bare mental: a2.metal, c5.metal,…
* vCPU: for example have 4 CPU and 2 threads per CPU 🡪 8 vCPU

**EC2 INSTANCE STORAGE**

***EBS volume***

* EBS volume is a network drive that can attach to instance while they run
* It’s locked to an AZ (created in us-east-1a and will not be attached to us-east-1b, if want to move across AZ, will need to snapshot it and then attach to another instance in another AZ or region
* Have two type of capacity (in GBs and IOPS)
* By default the first EBS attach to instance will be a root volume, and will be deleted when the instance terminate, and second EBS attach to instance will be not deleted.
* With the snapshot EBS:
  + Should use EBS snapshot Archive to reduce charge about storing
  + Recycle bin for EBS snapshot (can specify retention 1 day to 1 year)
* EC2 instance maybe the first ebs attach, and will be lose if instance faced something wrong
* ***EBS volume types***
  + gp2/gp3 (SSD) General Purpose: balances price and performance for a wide variety of workloads
    - Cost effective, low latency
    - 1GB-16TB
    - gp3: 3000 IOPS and throughput 125MB/s; can upto 16000 IOPS and 1000MB /s
    - gp2: can burst IOPS 3000, max IOPS 16000
  + io1/io2 (SSD): higheset perform SSD for mission critical, low latency, high throughput workloads
    - Use case: critical business application, or application need more than 16K IOPS, or database workloads
    - io1/io2 (4GB-16TB) 🡪 max IOPS 64K for nitro EC2 and 32K for other
    - io2 Block Express (4GB – 64TB) 🡪 sub milisecond latency, max IOPS: 256K with IOPS/GB=1K/1
    - Supports EBS multi-attach
  + st1 (HDD): low cost HDD used for frequently accessed, throughput intensive workloads
    - Cannot be a boot volume
    - 125MB to 16TB
    - Use case: big data, data warehouse, log processing
    - Max throughpu 500 MB/s max IOPS 500
  + sc1 (HDD): lowest cost HDD used for less frequently accessed
    - For data that is infrequently accessed
    - Scenarios where lowest cost is important
    - Max throughput 250MB/s max IOPS 250
* Only gp2/gp3 and io1/io2 can be used as boot volumes 🡪 root volume
* ***EBS encyption***
* Encryption and decryption are handled transparently
* Minimal impact on latency
* Leverages key form KMS (AES-256)
* Copying unencrypted snapshot allows encryption
* Snapshots of encryphted volumes are encrypted
* ***EFS:***
* Managed NFS (network file system)
* EFS works with instances in multi AZ
* Highly available, scalable, expensive, (3xgp2), pay per use
* Use cases: content management, web serving, data sharing, Wordpress
* Uses NFSv4.1 protocol
* Use security group to control access to EFS
* Not use for windows
* Perfomance EFS:
  + Scale: 1000s concurrent NFS clients, 10GB+/s throughput; grow to petabyte scale
  + Performance mode: **general** (web server, CMS), **max I/O** (big data, media processing)
  + Throughput mode: **bursting** (1TB= 50MB/s + burst of up to 100MB/s), **provisioned** (1GB/s for 1TB storge)
* Storage class:
  + Storage Tiers (lifecycle management feature – move file after N days)
  + Availability and durability
  + Over 90% in cost saving
* Note: EC2 instance store is the good way to use as the cache in ec2 instance, and have 310K IOPS

**LOAD BALANCE (ALB)**

* Các thành phần cần khởi tạo khi tạo load balance
  + Chọn public Internet hoặc private (dựa vào phần Internet-facing: dùng cho việc sử dụng public Internet và Internal dành cho private)
  + Chọn version IP
  + Security groups – khởi tạo để chọn ra các traffic dùng cho việc connect tới service trên EC2: ví dụ như HTTP 0.0.0.0/0 for anywherer
  + Network mapping – dùng cho việc available service, khi có tác động vật lý gây hư hỏng đến az thì có thể thay đổi
  + Listeners and routing, create group instance and port: và giao thức.
* Việc sử dụng load balancer thì khi một request tới thì sẽ random sử dụng qua lại giữa các instances trong group instance đã tạo trước
* Sử dụng load balance có thể checking được trạng thái của instance mà không cần phải vào check ở service EC2

**RDS, Aurora, ElasticCache**

* RDS including: Postgres, MySQL, MariaDB, Oracle, SQL Server, Aurora
* RDS is managed serves:
  + Provisioned, OS patching
  + Continuous backups and restore to specific timestamp
  + Monitor dashboard
  + Read repliccas for imporved read perfomance
  + Muti AZ for DR
  + Maintenance windows for upgrade
  + Scaling capability
  + Storage backed by EBS (gp2 or io1)
* RDS backup:
  + Automated backups: daily full backup, transaction logs every 5 minutes, 7 days retention backup (can be 35 days max)
  + DB snapshot: manually triggered, retention of backup for your decision
* RDS replicas vs multi az:
  + Network cost will be charged when the read replicas is stored in another region with rds instance
  + But if the read replicas is the same region with rds instance, we don’t need to charge anything about netwroking
* RDS multi AZ – can call it RDS instance standby 🡪 just using for reading, the same way with rds replicas, but when master rds instance failover, traffic about writing will switch to rds standby
* How to transfer between single az to multi az. When we enable option multi az, RDS instance master will create db snapshot, and then move it to RDS standby, when finish to create, RDS standby will create sync with master
* RDS security:
  + If master is not encrypted the read replicas cannot be encrypted
  + Provide SSL certificate
  + To enforce SSL: Postgres: rds.force\_ssl=1, MySQL: GRANT USAGE ON \*.\* TO ‘mysqluser’@’%’REQUIRE SSL;

**SQS, SNS, Kinesis, Active MQ**

* Với Kinesis thì việc truyền dữ liệu sẽ được gán partition key theo từng Shards
* Với SQS thì sẽ theo quy luật First In First Out, có thể chạy dữ liệu multiple group, nhưng các group sẽ independence.

|  |  |  |
| --- | --- | --- |
| SQS | SNS | Kinesis |
| + pull data by requesting SQS queue  + Finish processed and then delete it, no other consumer can read ever again  +\_No need to provision throughput  + Only FIFO queues  + Individual message delay capability | + push data to many sub  + upto 12.500.000 sub  + data is not persisted  + Pub/sub  + Up to 100.000 topics  + No need to provision throughput  + Intergrates with SQS for fan out architecture pattern  + FIFO capability for SQS FIFO | + Standard: pull data, 2MB/shard  + Ehanced-fan out: push data 2MB/shard/consumer  + Possibility to replay data  + Meant for real-time big data, analystics and ETL  + Ordering at the shard level  + Data expires after X days  + Provisioned mode or on-demand capacity mode |

* Amazon MQ:
  + Managed Apache ActiveMQ
  + Doesn’t scale as much as SQS/SNS
  + Run on a dedicated machine (máy chuyên dụng), can run in HA with failover
  + Both queue feature (SQS) and topic features (SNS)
* MQ High availability:
  + Deploy MQ Broker in two region (one region using active and another using standby)
  + Connect MQ active to EFS (mounted on multiple Availability Zones)