**1.** ჩამოთვალეთ ჩვენთვის კარგად ნაცნობი ვირტუალიზაციის ტიპები, იმსჯელეთ

რით განსხვავდებიან ერთმანეთისგან და რომელ ვირტუალიზაციის ტიპს

იყენებენ ქლაუდ პროვაიდერები? რა არის hypervisor?

ვირტუალიზაციის ტიპები და hypervisor

**ვირტუალიზაციის ტიპები:**

* **სერვერის ვირტუალიზაცია** – ერთი ფიზიკური სერვერი იყოფა მრავალ ვირტუალურ სერვერად.
* **სამუშაო გარემოს ვირტუალიზაცია** – მომხმარებელი მუშაობს ვირტუალურ დესკტოპზე, რომელიც შეიძლება იყოს ფიზიკურად სხვა ლოკაციაზე.
* **ქსელური ვირტუალიზაცია** – ქსელური რესურსები ლოგიკურად იყოფა და იმართება ეფექტურად.
* **შენახვის ვირტუალიზაცია** – სხვადასხვა ფიზიკური საცავი ერთიანდება ერთ ვირტუალურ გარემოში.

**Cloud Providers და ვირტუალიზაცია:**  
ქლაუდ პროვაიდერები (AWS, Azure, GCP) უმეტესად იყენებენ **სერვერის ვირტუალიზაციას**, რათა უზრუნველყონ რესურსების ოპტიმალური გამოყენება და მოქნილობა.

**Hypervisor:**  
Hypervisor არის პროგრამული უზრუნველყოფა, რომელიც მართავს ვირტუალურ მანქანებს (VMs). არსებობს ორი ტიპი:

* **Type 1 (Bare Metal)** – უშუალოდ მუშაობს ფიზიკურ აპარატურაზე (მაგ: VMware ESXi, Microsoft Hyper-V, Xen).
* **Type 2 (Hosted)** – ოპერაციულ სისტემაზე დაყენებული პროგრამა (მაგ: VirtualBox, VMware Workstation).

**2.** წარმოიდგინეთ რომ ხართ AWS-ს კონსულტანტი, როგორი ტიპის ქლაუდ

ინფრასტუქტურის გამოყენებას ურჩევდით ქართულ ფინტექ კომპანიას,

რომელიც ფუნქციონირებს ლოკალურ ბაზარზე და ამუშავებს მომხმარებლის

ტრანზაქციებსა და პირად მონაცემებს, ორგანიზაციის წარმომადგენლები ღელავენ

უსაფრთხოების საკითხებზე, რა მიდგომების გამოყენებას ურჩევდით

უსაფრთხოების უზრუნველსაყოფად.

**AWS ინფრასტრუქტურის რეკომენდაცია ქართულ ფინტექ კომპანიას**

**ინფრასტრუქტურის ტიპი:**

* **ჰიბრიდული ქლაუდი** – ფინტექ კომპანიისთვის ჰიბრიდული მოდელი კარგი არჩევანია, რადგან მისცემს მონაცემების ნაწილობრივ ლოკალურად შენახვის და AWS-ის სერვისების გამოყენების საშუალებას.
* **VPC (Virtual Private Cloud)** – მონაცემების იზოლაციისთვის და ქსელის უსაფრთხოების გასაუმჯობესებლად.
* **Multi-AZ Deployment** – მაღალი ხელმისაწვდომობისთვის მონაცემთა ბაზები უნდა განთავსდეს რამდენიმე ზონაში.

**უსაფრთხოების რეკომენდაციები:**

* **IAM (Identity and Access Management)** – მინიმალური პრივილეგიების (least privilege) პოლიტიკის გამოყენება.
* **AWS Shield და WAF (Web Application Firewall)** – DDoS შეტევების თავიდან აცილება.
* **Encryption (KMS, S3 Encryption, RDS Encryption)** – მონაცემების დაშიფვრა ტრანზიტსა და შენახვისას.
* **Security Groups და Network ACLs** – ქსელური უსაფრთხოების რეგულირება.
* **CloudTrail და GuardDuty** – აქტივობების მონიტორინგი და ანომალიების დაფიქსირება.

**3.** რა არის Autoscaling? რა შემთხვევაშია ვიყენებთ ამ სერვისს? დავუშვათ

ინსტანსების სამიზნე ჯგუფში ერთ-ერთი ინსტანსი გამოვიდა მწყობრიდან რას

მოიმოქმედებს autoscaling-ის მექანიზმი? რისი მანიშნებელია desired capacity?

**Autoscaling და მისი მექანიზმი**

**Autoscaling** არის AWS-ის მექანიზმი, რომელიც ავტომატურად ზრდის ან ამცირებს სერვერების (EC2 ინსტანსების) რაოდენობას დატვირთვის შესაბამისად.

**გამოყენების შემთხვევები:**

* **მოთხოვნის ზრდისას** – ტრეფიკის გაზრდის შემთხვევაში დამატებითი რესურსები ავტომატურად იწყება.
* **მოთხოვნის შემცირებისას** – არაეფექტური რესურსების თავიდან ასარიდებლად, ნაკლებად დატვირთული ინსტანსები ითიშება.
* **Fault Tolerance** – თუ ინსტანსი გამოვიდა მწყობრიდან, Autoscaling ქმნის ახალ ინსტანსს.

**თუ ინსტანსი გამოვიდა მწყობრიდან:**  
Autoscaling ჯგუფი ავტომატურად ჩაანაცვლებს ამ ინსტანსს ახალი, ჯანსაღი ინსტანსით, რათა დააბალანსოს მოთხოვნილი სიმძლავრე.

**Desired Capacity:**  
ეს არის ის რაოდენობა ინსტანსების, რომელიც ჯგუფს უნდა ჰქონდეს ნორმალურ პირობებში. Autoscaling ცდილობს ეს რაოდენობა მუდმივად შეინარჩუნოს.

**4.** ორგანიზაციამ გადაწყვიტა ახალმოსულ მონაცემთა ბაზების დეველოპერს მისცეს

aws cloud-ის გამოყენების უფლება, ახალი თანამშრომლისთვის ანგარიშის შექმნის

შემდგომ რა უნდა მოიმოქმედოს უფლებამოსილმა პირმა, რა IAM კომპონენტებს

უნდა მიაკუთვნოს ანგარიში? ასევე ჩამოთვალეთ IAM entities (კომპონენტები)

**IAM კომპონენტები და ახალი თანამშრომლისთვის კონფიგურაცია**

**მიმდინარე ნაბიჯები:**

* **ახალი IAM მომხმარებლის (IAM User) შექმნა**
* **მინიმალური საჭირო უფლებების მინიჭება (Policies, Roles)**
* **MFA (Multi-Factor Authentication) გააქტიურება**
* **Access Keys-ის გამოყენება CLI ან API წვდომისთვის (თუ საჭიროა)**

**IAM კომპონენტები (IAM Entities):**

* **Users** – ინდივიდუალური ანგარიშები AWS რესურსებზე წვდომისთვის.
* **Groups** – მომხმარებლების ჯგუფები, რომლებსაც ერთი და იგივე ნებართვები ენიჭებათ.
* **Roles** – გამოყენებულია სერვისებსა და ანგარიშებს შორის წვდომის გასაზრდელად.
* **Policies** – ნებართვების დოკუმენტები, რომლებიც განსაზღვრავს, რას შეუძლიათ IAM მომხმარებლებს.

ეს მიდგომები დაეხმარება კომპანიას, რომ უზრუნველყოს უსაფრთხოება და შესაბამისობა AWS პლატფორმაზე.

5. წარმოიდგინეთ, რომ კონსულტაციას უწევთ ჯანდაცვის ორგანიზაციას, რომელიც ქმნის ონლაინ პლატფორმას პაციენტთა მონაცემების დამუშავებისთვის, რომელიც ეტაპობრივად ინერგება სხვადასხვა კლინიკასა და ჰოსპიტალში. გუნდი გადაწყვეტს AWS-ის გამოყენებას, რადგან სურს დაბალი დანახარჯებით სწრაფი მასშტაბირება და საიმედოობა.

თქვენი ამოცანაა, რჩევები მისცეთ შემდეგ საკითხებთან დაკავშირებით:

**1. სერვისების შერჩევა და ქლაუდ ინფრასტრუქტურის რეკომენდაცია**

ჯანდაცვის სექტორში მონაცემების დაცულობა, მაღალი ხელმისაწვდომობა და მასშტაბირება კრიტიკულია. ამიტომ, გირჩევთ **ჰიბრიდული ან სრულად ქლაუდ ინფრასტრუქტურას**, რომელიც მოიცავს:

* **მონაცემთა დამუშავება და შენახვა**
  + **Amazon RDS (Aurora for Healthcare) ან DynamoDB** – თუ საჭიროა **სტრუქტურირებული მონაცემები და მაღალი ხელმისაწვდომობა**, RDS Aurora საუკეთესო არჩევანია (PostgreSQL ან MySQL). **თუ მოთხოვნაა დაბალი ლატენტურობა და სკალირებადობა**, მაშინ DynamoDB უკეთესი გამოსავალია.აღნიშნულ შემთხვევაში ჯობია Amazon RDS (Aurora for Healthcare).
  + **Amazon S3** – პაციენტთა დიდი მოცულობის ფაილების (სკანები, სურათები) უსაფრთხო შენახვისთვის.
  + **AWS Backup** – მონაცემთა უსაფრთხო და ავტომატური სარეზერვო ასლების შესაქმნელად.
* **ინფრასტრუქტურის მართვა და აპლიკაციის გაშვება**
  + **Amazon EC2 + Auto Scaling** – სერვერების ავტომატური მასშტაბირებისთვის.
  + **AWS Lambda** – მცირე API სერვისებისთვის, რომლებსაც არ სჭირდებათ მუდმივი სერვერი.
  + **Amazon ECS (Fargate) ან EKS (Kubernetes)** – კონტეინერიზაციისთვის.

**2. მაღალ ხელმისაწვდომობა და ინფრასტრუქტურის სიმარტივე**

მომხმარებლების ტრაფიკის მოულოდნელი ზრდის შემთხვევაში საჭიროა **ავტომატური მასშტაბირება და დატვირთვის გადანაწილება**:

* **Application Load Balancer (ALB) ან Network Load Balancer (NLB)** – ტრაფიკის სწორად გადანაწილებისთვის.
* **Auto Scaling Groups (ASG)** – EC2 ინსტანსების ავტომატური რეგულირება დატვირთვის მიხედვით.
* **Multi-AZ Deployment** – მონაცემთა ბაზების და სერვისების რამდენიმე Availability Zone-ში განთავსება.
* **Amazon CloudFront** – სწრაფი კონტენტისა და მედიის მიწოდებისთვის.

**3. უსაფრთხოება და მომხმარებელთა მონაცემთა დაცვა**

ჯანდაცვის მონაცემები დაცული უნდა იყოს **HIPAA და GDPR რეგულაციების შესაბამისად**. რეკომენდაციები:

* **IAM (Identity and Access Management)** – მინიმალური პრივილეგიების პოლიტიკა.
* **AWS KMS (Key Management Service)** – მონაცემების დაშიფვრა ტრანზიტსა და შენახვისას.
* **AWS Shield & AWS WAF** – DDoS თავდასხმებისგან დაცვისთვის.
* **Amazon GuardDuty & AWS Security Hub** – უსაფრთხოების ავტომატური მონიტორინგისთვის.
* **VPC (Virtual Private Cloud) და Security Groups** – ქსელური იზოლაციისა და წვდომის მართვისთვის.

**4. ხარჯების ოპტიმიზაცია**

AWS-ის ხარჯების კონტროლის გზები:

* **Spot Instances** – არაპრიორიტეტული სამუშაოებისთვის (მაგ. ანალიტიკა, მონაცემთა დამუშავება).
* **Reserved Instances (RI)** – გრძელვადიანი გამოყენების შემთხვევაში (მაგ. მონაცემთა ბაზები).
* **AWS Cost Explorer & Budgets** – ხარჯების ანალიზისა და კონტროლისთვის.
* **AWS Auto Scaling** – მხოლოდ საჭირო რესურსების გამოყენება.

**5. მონიტორინგი და ლოგირება**

პლატფორმის მონიტორინგისთვის და პრობლემების დროულად აღმოჩენისთვის:

* **Amazon CloudWatch** – აპლიკაციისა და სერვერების მონიტორინგი.
* **AWS X-Ray** – API ტრანზაქციების დეტალური ანალიზი.
* **AWS CloudTrail** – მომხმარებლების აქტივობების ლოგირება და აუდიტი.
* **Amazon SNS & AWS Lambda** – ავტომატური შეტყობინებები და რეაქციები პრობლემებზე.

ამ სერვისებით და მიდგომებით, პლატფორმა იქნება **მაღალი ხელმისაწვდომობის, უსაფრთხოების და ოპტიმიზაციის** დონეზე, რაც მისცემს კომპანიას საშუალებას, **მინიმალური დანახარჯებით სწრაფად გააფართოოს მომსახურება და შეინარჩუნოს მონაცემთა დაცულობა**.