| 問題 | 解答 |
| --- | --- |
| **①クラウドコンセプト** | |
| AWS6つのメリット | ・固定費を変動費に変える  ・スケールメリットによるコストメリット  ・キャパシティ予測が不要  ・速度と俊敏性が向上  ・データセンタの運用が不要  ・世界中に数分でデプロイ |
| クラウドサービスモデル3種類 | ・IaaS  ・PaaS  ・SaaS |
| AWSを操作する3つの方法 | ・マネジメントコンソール  ・コマンドラインインターフェース  ・ソフトウェア開発キット |
| **②請求とコスト** | |
| AWSの料金モデル3つ | ・時間単位制  使用した時間によって料金が変わります。  秒単位だったり、ミリ単位だったり・・・  ・データ容量  保存したデータ量や、確保したデータ量によって料金が変わります。基本的にGB単位です  ・データ送信  送信データ量に対してお金がかかります。  受診データは基本的に無料です。  送信データ量は、サービスごとではなくすべて集約してトータルの送信量で請求されます |
| 使いたい分だけ使うことの名前（必要な分だけ使用する） | オンデマンド |
| 使った分だけお金がかかることの名前 | 従量課金制 |
| たくさん使うと安くなることの名前 | ボリュームディスカウント |
| 予約して使う支払方法3つ | 全額前払い（AURI）  一部前払い（PURI）  前払いなし（NURI） |
| 無料で使えるサービス5種類 | AmazonVPC  Elastic　Beanstalk  Auto Scaling  Cloud Fomation  AWS IAM |
| オンプレ⇒クラウドに移行するときの料金見積もりサービス名称 | TOC計算ツール |
| 一括請求（アカウント統合）サービスの名称 | AWS　organizations |
| 3か月分の見積もりを出すサービス名称 | Cost　Explorer |
| AWSのサポート名称4種類 | ・ベーシックプラン  ・開発者プラン  ・ビジネスプラン  ・エンタープライズプラン |
| **③グローバルインフラストラクチャ** | |
| グローバルインフラストラクチャ3種類と特徴 | ・リージョン  ・アベイラビリティゾーン  ・POP |
| POPには何があるか | エッジロケーションとリージョン別エッジキャッシュ |
| **④セキュリティ** | |
| 責任共有モデルそれぞれ  ・AWSの責任  ・御客様の責任 | AWSの責任は、クラウドのセキュリティ  AWSが提供しているサービス自体のセキュリティに責任がある  お客様のセキュリティ　クラウドにおけるセキュリティ  AWSを使用してお客様が実装したものに責任がある |
| IAMの正式名称 | Identity　And　Management  認証と認可という意味があります |
| IAMの特徴 | IAMはグローバルサービス！  IAMで許可された権限はリージョンが変わっても継続します！ |
| IAMのコンポーネント4つ | ・ユーザ  ・グローバル  ・ポリシー  ・ロール |
| ポリシーとロールの違い | ポリシーは永続的な許可、ロールは一時的な許可です |
| 2段階認証などを用いる認証の名前 | 多要素認証　またはMFA |
| ルートユーザ作成後にやるべきこと | ・ルートユーザのアクセスキーを無効（削除）にする  ・ルートユーザの認証情報を安全な所へ移す　等 |
| アクセスログを監視するサービス名称 | Cloud　Trail |
| 鍵の作成・管理するサービス名称 | KMS |
| 他のSNSでログインできるサービス名称 | Amazon　Cognito |
| リソースの変更履歴を持つサービス名称 | AAWS　Config |
| コンプライアンスレポートをダウンロードできるサービス名称 | AWS　Artifact |
| **⑤ネットワーク** | |
| VPCの正式名称 | Virtual Private Cloud  最後のCはクラウドのCです！ |
| AWSで設定できるサブネットの有効範囲 | /16～/28まで |
| AWSで使用できないIPアドレスの数 | 全部で5つ  10.0.0.0/24の場合  10.0.0.0　ネットワークアドレス  10.0.0.1　内部通信  10.0.0.2　DNS用  10.0.0.3　今後利用予定  10.0.0.255　ブロードキャストアドレス  ちなみに、使えないIPを予約IPアドレスと言います |
| ・VPC内からインターネットにアクセスする為のゲートウェイの名称 | インターネットゲートウェイ  VPCにアタッチします |
| プライベートサブネットからインターネットにアクセスする為のゲートウェイの名称 | NATゲートウェイ  パブリックサブネットに配置するので注意！ |
| インスタンスに対するFWの名称 | セキュリティグループ  何かと使います！ホワイトリストで、ステートフル！ |
| サブネットに対するFWの名称 | ネットワークACL  用途は少なめ。ブラックリストでステートレス！ |
| DNSの役割を持つサービスの名称 | Route53  ドメイン名とIPアドレスを対応させるサービスです。PoPに存在します。  DNSの仕組みも確認しましょう！ |
| CDNの役割を持つサービスの名称 | Cloud　Front  キャッシュを保持するサービスです。  PoPに存在します！CDNとは何かも確認しましょう！ |
| **⑥コンピューティング** | |
| EC2の正式名称 | Elastic compute cloud  Cが2つでEC2です |
| EC2とは？ | 仮想マシンです！  EC2という仮想マシンをどう使うかによって、○○サーバという役割ができます。  例：WEBサイトをのっける⇒webサーバ  　　データを扱う⇒DBサーバ |
| EC2作成時に最初に設定するもの | AMI（Amazon　Machine　Image）  主にOSを選びます！ |
| インスタンスタイプとは何か？ | 一言でいうと、仮想マシンのスペックです。  CPUやメモリの性能を選ぶことができます |
| パブリックIPを固定するためのIPアドレス | Elastic　IPアドレス  通常、EC2を停止してまた立ち上げるとIPアドレスが変わってしまいますが、ElasticIPアドレスを使うと、EC2を停止しても、IPアドレスが変わりません。 |
| EC2の料金モデルと特徴 | 下記の通り |
| ①使った分だけ | オンデマンドインスタンス 使った分だけお金がかかります。  使用用途は、突発的なキャパシティの増加（予測できないもの）に使われます |
| ②予約して使う | リザーブドインスタンス  事前に予約して、キャパシティを確保できます。  予約時に、使用期間が1年または3年を選ぶことになり、途中でやめても、使用期間分のお金はかかります。  予測できる業務や、常に動かす必要があるものは、リザーブドインスタンスを使ったほうがお得です。 |
| ③入札して使う | スポットインスタンス  AWS内の空いているインスタンスを低価格で使用できます。需要に合わせて、スポット価格が変わっていくので、入札金額を設定して、入札額よりもスポット価格が下回っていれば、使用可能です。  途中で需要が高まり、スポット価格があがると途中で停止する可能性があります。  使用用途は、途中で停止しても問題ない業務（テスト等）です |
| ④⑤自分専用のハードウェアを使用する | ④ハードウェア専有インスタンス  ⑤専有ホスト  どちらも他のインスタンスが、自分が使用しているサーバに介入しない（同居しない）ようにできます。 自分しか使わないサーバ（ハードウェア）にできますが、④はインスタンスを停止⇒再開するとハードウェアが変わり、⑤はいかなる時も固定のハードウェアを使用することができます。  ⑤のような状況が必要なケースは、基盤やメーカを指定しないと実現できない業務がある場合等です。 |
| Lambdaとは | 一言でいうと、関数です。イベント・スケジュールに応じてプログラムが実行されます。 |
| Lambdaの特徴 | サーバレスコンピューティングです  Lambdaが事項されるサーバを管理することなく、プログラムを実行できます |
| Lambdaの使用用途 | 何かのイベントが起きた時や、事前にスケジュールを設定してLambdaを起動することができます。  例：業務開始時に自動的にEC2を立ちあげ、業務終了時に自動的にEC2を停止させる　等  実用的に使われているのは、アレクサや、ルンバなどのロボットです。声やイベントに応じて、プログラムが作動します。 |
| コンテナとは | 仮想化　という言葉がでてくればＯＫです。  正確には、OSの仮想化技術ですが、通常の仮想マシンよりも軽量で、スピードも速いです |
| コンテナに関するそれぞれの特徴 | 以下の通りです |
| Docker | コンテナの仕組みを作るサービスです。AWS外のサービスです。 |
| ECS | Dockerで作ったコンテナを管理してくれるサービスです。AWSで使用できます。 |
| Kubernetes | Dockerで作ったコンテナを管理してくれるサービスです。OSSで作られたAWS外のサービスです。 |
| EKS | Kubernetesを使っている人が、AWSとの互換性を求めたときに使用するサービスです。 |
| 上記の内、AWSのサービスは？ | 上記の通り、ECSとEKSが該当します。 |
| プラットフォームがすでに用意されていて、コードをアップロードするだけで簡単に、アプリケーションをデプロイできるAWSのサービスは？ | Elastic　Beanstalk  アップロードしたい言語を指定すると、その言語を扱うことができるプラットフォームを自動的に立ち上げ、管理をしてくれます。  結局裏で出来上がっているのは、EC2等なので、自分で1から構築するか、時間を掛けず全部用意してもらいたいか、で使用するかどうか選びましょう。 |
| **⑦ストレージ** | |
| EC2と組みあわせる永続的なストレージ名称 | EBS |
| 上記サービスの特徴（○○ストレージ） | ブロックストレージ  オブジェクトストレージは、一部変更しても、全体を更新しますが、ブロックストレージは、変更箇所のみ更新（更新を局所化）できるので、更新が早いです！ |
| EC2を停止すると消える揮発性のストレージ名称 | インスタンスストア  EBSと比較で出てくるので覚えてね |
| S3の正式名称 | Simple　Storage service  Sが3つでS3です！ |
| S3はどの位置に属するサービスか ①リージョン②ＡＺ③サブネット | リージョンサービスです。  S3作った時、リージョンだけ選びましたよね！ |
| アーカイブ（長期保存）を目的とするデータ保存に適切なストレージ名称 | S3　Glacier  Glacierは氷河という意味があります |
| ＡＷＳで使用するファイルストレージ名称 | EFS（最後のSはストアのS） |
| 上記サービスの特徴 | 通常EBSは1つのインスタンスからしかアクセスできませんが、EFSは複数のインスタンスからアクセスできるのが特徴です！ |
| **⑧データベース** | |
| マネージドとアンマネージドの違い | 一概に言えるわけではありませんが、マネージドはバックアップ機能、冗長化、アップデート等を自動で行ってくれるものです。管理が楽なので、マネージドを優先して使うほうが良いですね。  たいして、アンマネージドは上記の内容を自分で行わなければなりません。細かいカスタマイズをしたい場合には、アンマネージドを選んだほうが良いですね。  EC2はアンマネージド、RDSはマネージドサービスです |
| RDSが扱うデータベースエンジン1つ | ・MySQL  ・Oracle  ・Amazon　Aurora  ・Microsoft SQLserver  ・PostgreSQL  ・Maria DB  この中の内1つ言えていればOK! |
| RDSのRと、その意味 | Rはリレーショナルという言葉で、2次元の表をあらわまします。  Excelのような表形式でデータを保持する場合は、リレーショナルDBを扱います。 |
| 自分自身でSQLサーバを立てず（EC2にデータベースエンジンを導入せず）、RDSを使用するメリットは？ | 5/14に話しましたね！  RDSはマネージドなサービスなので管理が楽です。EC2をDBサーバにしたいのであれば、カスタマイズを自分で行い時に利用しましょう。 |
| マルチAZ配置とは何か | 複数のAZに配置することです。（そのままですね）  単一のAZに障害が発生したとしても、別のAZで動いているサービスがあれば、耐障害性を備えられます。  逆に言えば、単一のAZだけで動かすようなことは、本番環境ならあり得ません！ |
| DynamoDBはどのようなDBか（RDSと比較して） | SQLを扱わないデータの保持の仕方を行います。列（項目）に縛られず、自由に情報を保持したい場合は、DynamoDBを扱います。  例えば、商品の商品情報は商品によって持たせたい項目が変わりますよね。  洗剤であれば、商品の価格以外に内容量だったり、成分だったり・・・  ぬいぐるみであれば大きさだったり、材質だったり・・・  商品によって持たせたい情報が異なる時、RDSで保持するのは不便な場合があります。そんなときのDynamoDBです。 |
| 上記のようなDBは何というか | NoSQLといい、SQLを扱わずにデータを扱うことができます |
| RDSとDynamoDBを比較した時のそれぞれの特徴（最低2つ） | ・RDSは正確なデータのやり取りが必要な時に利用します。対して、DynamoDBは、処理の速さが求められるときに使用します  ・RDSは複雑な処理（抽出）が得意です。たいしてDynamoDBは抽出程度の簡単な処理しかできません  ・RDSに向いている処理はATM等のデータを扱う場合です。たいしてDyanamoDBはネットショップの買い物かご（その人しか扱わないデータ）を保持するのに扱われます。  他にもこんな答えは！というものがあったら、聞いてくださいね。 |
| AWSのデータウェアハウスの名称 | Amazon　Redshift  大量のデータを保存し、分析します |
| Amazon　Auroraの説明と特徴 | MySQLは、既に存在しているサービスですが、AWS独自のDBエンジンとして、Auroraというものがあります。  MySQLとPostgreSQLに互換性があり、MySQLの約5倍の速度で処理され、マネージドより1歩上をいく、フルマネージドサービス（完全自動化）を実現しています。  とにかく、ほかのSQL使うよりも早くて、性能もよくて、耐障害性も優れています！ |
| **⑨クラウドアーキテクチャ** | |
| フレームワーク5つの柱をそれぞれ答える | ・運用上の優秀性  ・セキュリティ  ・信頼性  ・パフォーマンス効率  ・コスト最適化 |
| 以下に該当する説明が何の柱か答えること | 以下の通り |
| ①トレーサビリティの実現 | セキュリティ  監視できる状況を作りましょう！ |
| ②キャパシティを推測しない | 信頼性  リソースの追加や削除は人間が考えず、自動化しよう！ |
| ③運用をコードとして実行する | 運用上の優秀性  ヒューマンエラーをなくし、自動化しましょう！ |
| ④定期的に元に戻せるよう変更を適用する | 運用上の優秀性  運用時トラブルに対応できるような備えをしましょう |
| ⑤最新テクノロジーの標準化 | パフォーマンス  テクノロジーをサービスとして利用しましょう  （経済格差なく最新のものを利用できるのがクラウド） |
| ⑥サーバレスアーキテクチャを使用 | パフォーマンス  管理することなく、自動化させましょう！ |
| ⑦費用を分析し帰結させる | コスト最適化  システムの使用状況とコストを特定しましょう |
| ⑧水平スケールでシステムの可用性を高める | 信頼性  耐障害性を備え、故障が起きにくい設計にしましょう |
| 信頼性について説明 | 必要な機能を、必要な時に提供できるシステムの能力 |
| 高可用性について説明 | 正常に稼働することに加え、維持はもちろん劣化にもたえらえる能力 |
| 耐障害性について説明 | 一部に障害が発生しても、稼働できるようなシステム構成の仕組み |
| Trusted Advisorの5つを答える | ・コスト最適化  ・パフォーマンス  ・セキュリティ  ・耐障害性（フォールトトレラント）  ・サービス制限 |
| **⑩Auto　Scalingとモニタリング** | |
| Load　Balancing　3種類と特徴 | **・Application　Load Balancer**  ⇒HTTPやHTTPSのトラフィックを負荷分散  　複雑なルーティングに対応  **・Network　Load　Balancer**  ⇒TCP、UDP、TLSのトラフィックを負荷分散  　きわめて高いパフォーマンスが要求されるルーティングに対応（大量アクセス等）  **・Classic　Load　Balancer**  ⇒旧世代のロードバランサー  　互換性用に残ったままになっている  ※ちなみに最近は、ゲートウェイロードバランサーも追加されました！セキュリティ要素が高めです。 |
| Amazon Cloud　Watchとは | モニタリングのサービス  CPU等のメトリクス収集により追跡する  何かを自動化させるなら、cloud watch必須！ |
| アラームの通知に使用するサービス名 | Amazon SNS |
| EC2 Auto Scalingとは | コンピュータの性能などを自動で増減させるためのサービス。単なる増減だけでなく、異常を検知した時の置き換えにも対応 |
| Auto Scalingグループで設定するもの（3つ） | 最小サイズ  最大サイズ  必要キャパシティ |
| **その他** | |
| サーバ等の台数を増やすことをなんというか | スケールアウト |
| サーバ等の台数を減らすことをなんというか | スケールイン |
| サーバ等の性能を上げることをなんというか | スケールアップ |
| サーバ等の性能を下げることをなんというか | スケールダウン |
| 水平スケーリングとは | スケールアウトのこと、台数を増やすこと |
| 垂直スケーリングとは | スケールアップのこと、性能を上げること |
| レイテンシーとは | 遅延のこと　低レイテンシー⇒遅延なく高速 |
| デプロイとは | リリース、展開のこと。システムをデプロイ（リリース）等と使用します |
| プロビジョニングとは | 準備のこと、リソースのプロビジョニング等 |