AWS Certified Security Specialty

○試験メモ
・「イメージを作成」と「インスタンスからテンプレートを作成」の違い
イメージを作成；ボリュームやネットワーキング設定の状態を保存
インスタンスからテンプレートを作成：イメージに加えて、インスタンスタイプや起動スクリプトも含めて作成
・SCPはPrincipalとNotPrincipal、NotResourceをサポートしていない
・Secrets Managerは機密情報が高いパラメータの保管および自動ローテーションをサポートしているが、利用料金が高い
Lambda関数から複数のパラメータを呼び込むためには、Systems ManagerのParameter Storeを使った方が良い。
・**アクセスキーの使用はCloudTrailのAPI利用履歴でわかる**
・**CloudTrailが無効になったことを検知するのはGuardDutyでできる（Stealth:IAMUser/CloudTrailLoggingDisabled）**
・kms:EncryptionContextは暗号文の改竄を防ぐために追加認証データを使用して、IAMポリシーを定義する際に条件として追加することができる
・ALBとEC2インスタン間の通信はHTTPSに設定できる
・CloudTrailにより、DynamoDBを呼び出したソースIPを特定することができる（EC2インスタンスへのアクセスはキャプチャしない.）
・KMSキーのベストプラクティスはキーを自動ローテーションする設定にしておくこと。
・Configでは、パブリックアクセスのあるS3バケットが作成された場合に監視とアラートを行うことができる。
・アクセスキー使用の検出は、CloudTrail→EventBridge→SNS
・Amazon Sheildはレイヤー3,4のDDoS攻撃をブロックする。Shield Advancedはレイヤー3,4,7の攻撃をブロックできる
・sts:assume roleはアクセス元のIAMロールで設定する。
・**CloudTrailの検知はほぼリアルタイム(15分)**
・カスタムConfigルールはLambda関数により定義できる（VPCフローログの適用有無を調べるなど）
・CloudWatchとQuickSiteは直接連携できない
・S3 Vault LockはGlacierのみ。標準ストレージには対応していない
・**インポートされたKMSキーは手動でのみローテーションできる**
・GuardDutyの信頼できるIPリストはリージョン固有であり、またパブリックIPにのみ有効
・**AWS管理のKMSキーは、キーポリシーの設定ができない（そのため、 IAMユーザー側の読み取り権限なども設定不要）**
・アクセスキーの侵害は、GuardDutyで検出できる
・

○**AWS Trusted Advisor**
・コスト最適化：活用されていないEBSボリューム、Lambda関数の過剰なタイムアウト、アイドル状態のRDSインスタンス、関連づけられていないElastic IP等
・パフォーマンス：EC2のコンピューティング使用量、EBSのスループットとレイテンシー分析、CloudFront設定など

・セキュリティ：RDSのセキュリティグループのアクセスリスク（ソース0.0.0.0のポート21は赤警告、22は黄警告など）、漏洩したアクセスキー、不要なS3バケット許可など
・耐障害性：AZの無効化、Route53のヘルスチェック削除、RDSのバックアップ無効化の調査など
・サービスクォータ：AWSアカウントに作成できるリソース最大量

○**AWS Inspector**
・パッケージの脆弱性や意図しないネットワーク露出領域を継続的なスキャンで検出する脆弱性管理サービス
・検出できるタイプ
・パッケージの脆弱性：検出されたEC2、ECrコンテナイメージ、Lambda関数のソフトウェアパッケージをスキャンして脆弱性に関するCVEを示す
・ネットワーク到達生：EC2への許可されたネットワークパスがあるかどうかを示す。インターネットゲートウェイ、LB、VPCピアリング接続、VPWを介したVPNなどのVPCから到達可能かスキャン
・EC2スキャン - パッケージの脆弱性
・Systems Manager Agentによって、EC2のソフトウェアパッケージの情報を収集し、Inspectorがスキャンを実施
・ECRスキャン
リポジトリごとにオンプッシュスキャンと連続スキャンの2種類のスキャン方法を設定可能
・オンプッシュスキャン：イメージがリポジトリにプッシュされた場合のみスキャン
・連続スキャン：オンプッシュスキャンに加えて、InspectorがCVE情報をデータベースに追加するたびにスキャン
（期間は、Lifetime(デフォルト)、180日、30日から選択可能）
・Lambda関数スキャン
・Lambda関数コード内、Lambda Layerで使用されているアプリケーションパッケージの脆弱性を検出する。
・以下のパッケージでスキャン可能
・Amazon InspectorがLambda関数を検出したとき
・新しいLambda関数をデプロイした時
・既存のLambda関数を更新した時
・Amazon Inspectorが新しい脆弱性をデータベースに追加したとき
・EC2スキャン - ネットワーク到達性
・24時間ごとにネットワーク到達性をスキャン
・IGW、VGW、ピアリングVPCがあるかないかのチェック

・まとめ
・Inspectorはネットワーク到達性やソフトウェアパッケージの脆弱性をニアリアルタイムに検出できる脆弱性管理システム
・Organizationで一元管理できる、またSecurityHubに統合できる
・EventBridge→SNS（Lambda）のような構成もできる
Systems Managerとの統合も可能
・環境内のリソースを自動で認識することでスケーラブルな監視が可能
・数クリックで簡単に有効化して即座に利用できる

・ソフトウェアのインストール、新しい脆弱性が発見された場合などにスキャンが自動で行われ、リアルタイムで脆弱性情報を確認できる

○**Amazon GuardDuty**

・サービスログの監視

・AWS Organizationで複数アカウントを管理できる。

（管理アカウントは、メンバーの追加削除、抑止ルール、信頼IPリスト、脅威リストを設定できる）

・監視対象ログ（各種ログを直接pullする）

・CloudTrail Event Logs

・VPC Flow Logs

・DNS Logs

・Optional Features(S3 Logs、EBS Volumes、RDS&Aurora Login Activity、Lambda Network Activityなど)

・EventBridgeルールと関連づけれる

・EventBridgeはSNSとLambdaと紐付けれる

（LambdaからWAFやNetword Firewallのルールを自動更新する）

・暗号通貨DDoS攻撃を防げる。

・FindingTypes

・EC2 Finding Types

・IAM Finding Types

・Kubernetes Audit Logs Finding Types

・Malware Protection Finding Types

・RDS Protection Finding Types

・S3 Finding Types

・GuardDuty EC2の検出結果タイプ

・Backdoor:EC2/C&CActivity.B

・Backdoor:EC2/C&CActivity.B!DNS

・Backdoor:EC2/DenialOfService.Dns

・Backdoor:EC2/DenialOfService.Tcp

・Backdoor:EC2/DenialOfService.Udp

・Backdoor:EC2/DenialOfService.UdpOnTcpPorts

・Backdoor:EC2/DenialOfService.UnusualProtocol

・Backdoor:EC2/Spambot

・Behavior:EC2/NetworkPortUnusual

・Behavior:EC2/TrafficVolumeUnusual

・CryptoCurrency:EC2/BitcoinTool.B

・CryptoCurrency:EC2/BitcoinTool.B!DNS

・DefenseEvasion:EC2/UnusualDNSResolver

・DefenseEvasion:EC2/UnusualDoHActivity

・DefenseEvasion:EC2/UnusualDoTActivity

・Impact:EC2/AbusedDomainRequest.Reputation

・Impact:EC2/BitcoinDomainRequest.Reputation

・Impact:EC2/MaliciousDomainRequest.Reputation

・Impact:EC2/PortSweep

・Impact:EC2/SuspiciousDomainRequest.Reputation
・Impact:EC2/WinRMBruteForce
・Recon:EC2/PortProbeEMRUnprotectedPort
・Recon:EC2/PortProbeUnprotectedPort
・Recon:EC2/Portscan
・Trojan:EC2/BlackholeTraffic
・Trojan:EC2/BlackholeTraffic!DNS
・Trojan:EC2/DGADomainRequest.B
・Trojan:EC2/DGADomainRequest.C!DNS
・Trojan:EC2/DNSDataExfiltration
・Trojan:EC2/DriveBySourceTraffic!DNS
・Trojan:EC2/DropPoint
・Trojan:EC2/DropPoint!DNS
・Trojan:EC2/PhishingDomainRequest!DNS
・UnauthorizedAccess:EC2/MaliciousIPCaller.Custom
・UnauthorizedAccess:EC2/MetadataDNSRebind
・UnauthorizedAccess:EC2/RDPBruteForce
・UnauthorizedAccess:EC2/SSHBruteForce
・UnauthorizedAccess:EC2/TorClient
・UnauthorizedAccess:EC2/TorRelay

・GuardDutyのIAM検出ログタイプ
・CredentialAccess:IAMUser/AnomalousBehavior
（AWS 環境へアクセスを取得するために使用された API が異常な方法で呼び出されました。）
・DefenseEvasion:IAMUser/AnomalousBehavior
（防御対策を回避するために使用された API が異常な方法で呼び出されました。）
・Discovery:IAMUser/AnomalousBehavior
（リソースの検出に一般的に使用される API が、異常な方法で呼び出されました。）
・Exfiltration:IAMUser/AnomalousBehavior
（AWS 環境からデータを収集するために一般的に使用される API は、異常な方法で呼び出されました。）
・Impact:IAMUser/AnomalousBehavior
（ある AWS 環境でデータやプロセスを改ざんするために、一般的に使用される API が、異常な方法で呼び出されました。）
・InitialAccess:IAMUser/AnomalousBehavior
（ある AWS 環境への不正アクセスを取得するために一般的に使用される API が、異常な方法で呼び出されました。）
・PenTest:IAMUser/KaliLinux
（API が Kali Linux EC2 マシンから呼び出されました。）
・PenTest:IAMUser/ParrotLinux
（API が Parrot Security Linux マシンから呼び出されました。）
・PenTest:IAMUser/PentooLinux
（API が Pentoo Linux マシンから呼び出されました。）
・Persistence:IAMUser/AnomalousBehavior
（ある AWS 環境への不正アクセスを維持するために一般的に使用される API が、

異常な方法で呼び出されました。）
・Policy:IAMUser/RootCredentialUsage
（API がルートユーザーサインイン認証情報を使用して呼び出されました。）
・PrivilegeEscalation:IAMUser/AnomalousBehavior
（通常ある AWS 環境への高レベルの許可を取得するために使用される API が異常な方法で呼び出されました。）
・Recon:IAMUser/MaliciousIPCaller
（API が悪意のある既知の IP アドレスから呼び出されました。）
・Recon:IAMUser/MaliciousIPCaller.Custom
（API が悪意のある既知の IP アドレスから呼び出されました。）
・Recon:IAMUser/TorIPCaller
（API が Tor 出口ノードの IP アドレスから呼び出されました。）
・Stealth:IAMUser/CloudTrailLoggingDisabled
（AWS CloudTrail ログ記録は無効です。）
・Stealth:IAMUser/PasswordPolicyChange
（アカウントのパスワードポリシーが弱化されています。）
・UnauthorizedAccess:IAMUser/ConsoleLoginSuccess.B
（世界中でコンソールに対する複数の正常なログインが確認されました。）
・UnauthorizedAccess:IAMUser/InstanceCredentialExfiltration.InsideAWS
（インスタンス起動ロールを通じて EC2 インスタンス専用に作成された認証情報は、AWS 内の別のアカウントから使用されています。）
・UnauthorizedAccess:IAMUser/InstanceCredentialExfiltration.OutsideAWS
（インスタンス作成ロールで EC2 インスタンス専用に作成された認証情報が外部 IP アドレスから使用されています。）
・UnauthorizedAccess:IAMUser/MaliciousIPCaller
（API が悪意のある既知の IP アドレスから呼び出されました。）
・UnauthorizedAccess:IAMUser/MaliciousIPCaller.Custom
（API がカスタム脅威リストにある IP アドレスから呼び出されました。）
・UnauthorizedAccess:IAMUser/TorIPCaller
（API が Tor 出口ノードの IP アドレスから呼び出されました。）

・GuardDuty RDS Protection の検出結果タイプ
・CredentialAccess:RDS/AnomalousBehavior.SuccessfulLogin
・CredentialAccess:RDS/AnomalousBehavior.FailedLogin
・CredentialAccess:RDS/AnomalousBehavior.SuccessfulBruteForce
・CredentialAccess:RDS/MaliciousIPCaller.SuccessfulLogin
・CredentialAccess:RDS/MaliciousIPCaller.FailedLogin
・Discovery:RDS/MaliciousIPCaller
・CredentialAccess:RDS/TorIPCaller.SuccessfulLogin
・CredentialAccess:RDS/TorIPCaller.FailedLogin
・Discovery:RDS/TorIPCaller

・GuardDuty Malware Protection
・マルウェアファイル検知を有効、
・コンテナ対応

○**AWS Security Hub**
・中央でAWSアカウントを管理して、セキュリティチェックを行う
・ダッシュボードが統合されている
・Config
・GuardDuty
・Macie
・Inspector
・IAM Access Analyzer
・AWS Systems Manager
・AWS Health
・AWS Firewall Manager
・3rd Party製品（aqua, 3CORE SECなど）
・上記のサービスから検知された内容は、以下のサービスと連携できる
・Audit Manager
・AWS Chatbot
・Amazon Detective
・Trusted Advisor
・SSM Explorer
・異なるアカウントのSecurity HubもAWS Organizationで統合することができる
・AWS Configは Enableになっている必要がある。
・５分以内に検知が送信される

○**Amazon Detective**
・ログの調査できるサービス
・機械学習を用いて、各種サービスの検知について、より詳細な分析ができる
（GuardDuty, Macie、Security Hubなどの検知に基づく）
・ログの自動収集、分析の自動化、視覚化
・アラートのトリアージ
・アラートがお客さん環境にとってTrue,Positive化を素早く分析
（例：送信データのサイズは？この通信は常時発生しているか？インシデントの前に何が起きたか？APIコールの失敗は異常なことか？）
・False Positiveなら、不要な調査を回避
・True Positiveと判断、またはFalse Positiveと判断できない場合は、優先順位をつけて、インシデント調査へ
・インシデント調査
・インシデントの根本原因、被害の影響を調査
・他のデータと相関させた深い操作
・分析例
・このIPと通信しているEC2はあるか？、他に悪用されたPrincipal IDはあるか？
・スレットハンティング
・内外のIndicator of Compromise(侵害の痕跡）を元に自組織にも影響があるかないか。
（例：脅威レポートで報告されたIPアドレスが過去1年間にEC2インスタンスと通

信したか？）
・CloudTrailの情報を内部で収集しているので、CloudTrailを無効にされても情報収集は可能
・前提サービス
GuardDutyを有効化した48時間後に、Amazon Detectiveのコンソールに移動して有効化
（有効化後は24時間程度まつ。）
・収集もと
・CloudTrail
・GuardDuty
・VPC Flow Logs
・開始後2週間は機械学習のトレーニング期間
・Detectiveを有効化するとデータを自動収集
・収集のために特別な設定は必要ない
・GuardDutyから取り込まれる検出結果は一部の結果タイプ
・Detectiveのマルチアカウント（マスターアカウント、メンバーアカウント）はOrganizationsとは別の管理
・マルチアカウントでもGuardDutyとSecurity Hubとの統合も可能
GuardDutyとSecurity Hubのマスターアカウントと、Detectiveのマスターアカウントは同一を推奨

○**Amazon Macie**
・機械学習とパターンマッチングを組み合わせて機微情報(PII(Personal Indicator Information)など)を検出する。
・更新間隔はデフォルトで15分（15分以下にはできない）
・Organizationsとの連携も可能）

・機能
・S3バケットの利用状況の可視化、格納されている大量のオブジェクトの可視化
・設定に基づき、指定されたバケット内の機微情報の評価・検出を効率的に実行（AWSマネージド定義とカスタム定義の両方を利用可能）
・評価検出結果の参照、他サービスへの連携

・ジョブを指定してスキャンする（毎日、毎週、毎月）
・SSE-S3、AWS-KMS、SSE-KMSによる暗号化が行われている場合、Macieはバケットをスキャンすることが可能
・クライアントサイド暗号化、カスタマー提供型のサーバーサイド暗号化（SSE-C）の場合には、復号する手段が無いため、スキャンできない

○**ペネトレーションテストとは？**
・テスト対象の企業/組織に応じて様々なサイバー攻撃手法を講じて、システムなどへの

侵入を試みることでセキュリティレベルを評価する取り組み
・以下の８サービスは事前承認なしで、貫通テストできる
・EC2。WAF、NATゲートウェイ、ELB
・RDS
・Lambda and Lambda Edge Function
・Amazon Lightsail resources
・Amazon Elastic Beanstalk
・Amazon Aurora
・Amazon CloudFront
・Amazon API Gateways
・AWS AppSync
・Amazon Elastic Beanstalk
・AWS Fargate
・Amazon Elastisearch
・Amazon FSx
・Amazon Transit Gateway

○**DDoS Simulation テスト**
AWS DDoSテストパートナーに承認を得る必要がある。

○**Compromised EC2インスタンス（侵害されたEC2）**
対応手順
１.インスタンスメタデータを取得
２.終了保護をする
３.インスタンスを孤立させる（SGから切り離す）
４.ELBからEC2を切り離す
５.EBSのスナップショットを撮る。

・オフライン調査
インスタンスをシャットダウンする
・オンライン調査
スナップショットやネットワークトラフィックから調べる
・孤立手順はLambdaで自動化
・SSM Run Commandでメモリー取得を自動化

○**Compromised S3（侵害されたS3）**
・GuardDutyで侵害されたS3を特定する
・怪しい挙動をしたもとを検知し、CloudTrailやAmazon Detectiveを使って検査

○**Compromised ECS**
・GuardDutyで検知
・怪しい挙動を評価して切り離す。（Deny all ingress/egress traffic（セキュリティグループで））

○**Compromised RDS**

・GuardDutyで検知
・もし怪しい挙動が見られたら、
・セキュリティグループとACLでアクセスブロック
・DBユーザーのアクセスをアクセスを切り離す
・DBパスワードを変更する

○**EC2 Key Pair - Explained**
1.UserはEC2作成時に、プライベートキーをダウンロードする
2.EC2は~/.ssh/authorized_keysにパブリックキーを格納する
3.ユーザーはプライベートキーを使って、EC2にSSH接続できる。

○**EC2インスタンスConnect**
・ユーザーはEC2 Connect APIに接続する
・AWS IP Rangeをソースとして、SSH 22をセキュリティグループで許可する

○**AWS Systems Manager**
・Resource Groups
・EC2のtagに基づいて、グループに分けることができる
・リージョナルサービス
・Operation
・Shared Resources
・Documents
・JSONもしくはYAMLで記載
・パラメータを定義できる
・アクションを定義できる

・Change Management
・Automation
・EC2再起動、AMI作成、EBSスナップショットなどを自動実行できる
・Automation Runbookでアクション指定（AnsibleのPlaybookのようなもの？）
・コンソール、SDK、EventBridgeのスケジュール、メンテナンスウィンドウ、Configルール修正をトリガーとして起動できる
・Maintenance Windows
・Application Management
・Parameter Store
（例）Name：/my-app/prod/db-url、Type：String、Value：prod.database.pass）
・aws ssm get-parameters —names “パラメータ名”で取得できる
（KMSで暗号化されたSecret Valueは「aws ssm get-parameters —names “パラメータ名” —with-decryption」で取得できる
・設定、パスワードなどを保存できる
・KMSを使って暗号化できる
・サーバレス、スケーラブル、耐久性高い

・CloudFormationと連携
・EventBridgeを使って通知できる

・Node Management
・Inventory
・EC2のメタデータを取得できる（インストール済みソフトウェア、OSドライバ、設定、起動中サービスなどの情報）
・S3とAthenaで解析できる
・複数のアカウントとリージョンからクエリデータ取得可能
・カスタムメタデータによるメタデータの取得も可能

・Session Manager
・SSHアクセス、SSHキー、bastion hostsがいらない
・EC2、オンプレミスに安全なシェルがスタートできる
・CloudTrailはStartSessionで検知
・S3とCloudWatch Logsに出力できる
・IAMポリシーでインスタンスのタグを使って許可する
（"ssm:resourceTag/Environment":["Dev"]）

・Run Command
・documentを実行するもしくは、ただコマンドを実行する
・SSHは必要なし
・SNSで通知できる
・EventBridgeを呼び出せる
・S3もしくはCloudWatch Logsに結果を出力できる
・Patch Manager
・パッチを自動化
・脆弱性のあるパッチを検知できる
・S3にレポートを送れる
・Patch Baseline
・何をEC2にインストールすべきかを定義する
・リリースされたら自動でパッチが適用される
・Patch Group
・State Manager

○**Amazon EventBridge**
・スケジュール起動
・Eventパターンによる起動
・IAMルートユーザサインイン
・Lambdaによる起動
・サービスとの連携による起動
・Codeシリーズ
・S3イベント
・Trusted Advisor
・EC2

・CloudTrail
・Event Bus
複数のイベントソースからのイベントを一元的に管理する。
・デフォルトEvent Bus
・パートナーEvent Bus（DATADOG、zendeskなど）
・カスタムEvent Bus（）

○**Amazon CloudTrail**
・APIコールなどを記録（リアルタイムではない）
（APIコールは15分以内、ログファイルのS3送信は5分ごと）
・デフォルトで90日間保存される。
・SNSによる通知をするには、EventBridgeのイベントパターンと連携する
（CloudTrail → EventBridge → SNS）
・CloudTrail Events
・Management Events
・セキュリティ設定（IAM：AttachRole Policy）
・ルーティングルール（EC2 CreateSubnet）
・ログ設定(AWS CloudTrail CreateTrail)
・Data Events
・デフォルトでは無効化
・S3オブジェクトレベルアクティビティ（ex:GetObject、DeleteObject、PutObject）
・Lambda実行アクティビティ（the Invoke API）
・CloudTrail Insights Events
・不審なアクティビティを検知する
・サービス制限に達した
・IAMアクションのバーストに達した
・定期的なメンテナンスのギャップ
・Digest Files
・それぞれのログファイルにハッシュ化することで、ログファイルに変更が加えられていないかを検知できる
・SHA-256、RSAのSHA-256によるデジタル署名が使われる
・S3に保存されるので、S3をMFAやバケットポリシーで保護しておく
・AWS Organizationを管理アカウントで有効化することで、他アカウントも有効にできる。
（他アカウントはCloudTrailを編集できず、閲覧のみできる）

○**VPC Network Access Analyzer**
・2021年の新機能
・AWSないのネットワークインターフェース間のパスを分析して出力してくれる。

○**フェデレーションIAMロールとは**
・IAMにおいて、外部の信頼されたIDプロバイダ（Identity Provider）と連携してアクセス制御を行うためのロール

1.外部プロバイダとの信頼関係を設定する
2.ロールの作成により、フェデレーションユーザーがアクセスするリソースへの適切な権限を持つロールを定義する
3.ユーザーのフェデレーション設定。外部IDプロバイダでの認証後、フェデレーションユーザーに一時的なセキュリティトークンが発行される

〇AWS WAF
・レートベースのルール機能：5分以内に大量のHTTPリクエストを行う送信元IPアドレスを検出し、問題のある送信もとIPからのリクエストを自動的にブロックする
・geo matchステートメントにより特定の国をブロックすることができる。
・特定のIPを許可するには、WAF IP setステートメントを作成する。
・Baseline Rule Groups
・一般的なプロテクション（AWSManagedRulesCommonRuleSetなど）
・Use-case Specific Rule Groups
・AWSManagedRulesSQLiRuleSetなど
・IP Reputation Rule Groups
malicious IPのブロック（AWSManagedRulesAmazonReputationList）
・Bot Control Managed Rule Groups
・ボットからのアクセスをブロック（AWSManagedRulesBotControlRuleSet）
・Web ACL - Logging
・CloudWatch Logs log group - 5MB per Second
・S3 - 5 minutes interval
・Kinesis Data Firehouse - limited by Firehouse quotes
・CloudFront Origin Security
・AWS WAF → CloudFront(Custom HTTP Header) → AWS WAF（ALBへの直接アクセスを防ぐため） → AWS Secrets Manager → Lambda →CloudFrontのHTTP Headerを変更 → ALB

○AWS Shield
・DDoSを防ぐ
・Standard
・Free
・Layer3/ Layer4 を防ぐ、SYN/UDPフロードを防ぐ
・Advanced
・$3,000 per month
・Layer3/ Layer4/Layer7 を防ぐ、SYN/UDPフロードを防ぐ
・EC2, ELB, CloudFront , Global Accelerator , Route53の高度な攻撃を防ぐ
・24時間サポートセンターに連絡可能
・DDoS mitigationを自動作成
・CloudWatch Metrics
・DDosDetected
・DDoSAttackBitsPerSecond
・DDoSAttackPacketsPerSecond

・DDoSAttackRequestsPerSecond

○**AWS Cognito**

APIベースで実装されるモバイルアプリやWebアプリにユーザー認証機能を提供するサービス

・ユーザプール

・独自のディレクトリでユーザーサインインやフェデレーションの情報に基づいてアプリへのアクセスに利用できるトークン（JWT）を提供

・IDプール

Cognitoユーザープールに加え、外部IDプロバイダでのログインに基づき、AWSにアクセスできるクレデンシャルを提供

○**AWS Managed Microsoft AD**

・Active DirectoryをAWS上に構築できるようになるサービス
（サービスの実態は、Microsoft AD）

・ログインを求めて、安全なユーザーがサービスを使うことを担保する

・ディレクトリサービスにより、ユーザやグループ、ITリソース、クライアント端末の情報を保存する

・複数アカウントや日次のスナップショットが可能

・信頼関係の設定（双方向）

AWS側の操作

1ディレクトリサービスの画面でディレクトリのセットアップを行う（AWS Managed Microsoft ADを選択）

2ディレクトリ情報（DNS名を入力する、アドミンパスワードなど）

3 VPCとサブネットを選択

4 ADのIPアドレスとDNS名が提供される

5 MSADを管理するEC2を起動する
（SSSManagedInstancecore,SSMDirectoryServiceAccessのロールをふよ）

6管理サーバーにツールをインストール

7セキュリティグループでオンプレミスのディレクトリに対するIPアドレスのアウトバウンドルールを追加する
（d-{ディレクトリID}_controller）

8 AWS MSADで信頼関係を追加する

オンプレミス側の設定

1ドメインコントローラー上でフォワーダーの設定を使う（他DNSサーバにクエリを条件付きで転送する。）

2信頼関係の設定

・ディレクトリのモニタリング

・ディレクトリのステータスを

・CloudWatch Logs

・ログインしたアカウント
・アクセスしたオブジェクト

○**AD Connector**
・既存のオンプレミスの Active Directory ユーザー情報で AWS リソース利用可能
・クラウド側でのユーザー管理が必要ない場合
・オンプレミスがわでユーザー管理のみを行うシンプルな構成が可能
・オンプレミス側 AD への問い合わせが発生する

○**Simple AD**
・低コスト
・AD と互換性のある Samba4 のサービスを提供
・5000 ユーザー以下でユーザー管理
・ユーザーとグループ管理だけで十分な場合
・オンプレミスとの信頼関係は使用不可

○**AWS Firewall Manager**
・AWS Organization で全てのアカウントのルールを管理できる
・Security Policy
・WAF rules（ALB、API Gateway、CloudFront）
・AWS Shield Advanced（ALB, CLB, NLB, Elastic IP, CloudFront）
・SG for EC2、ALB。ENI
・Network Firewall(VPC Level)
・Route 53 Resolver DNS Firewall
・Policies are created at the region level

○**CloudHSM**
・AWS が提供する暗号化ハードウェア
（KMS は AWS が提供する暗号化ソフトウェア）
・占有ハードウェア
・HSM デバイスは FIPS 140-2 Level 3 を満たす
・有料
・symmetric と asymmetric の両方をサポート
・CloudHSM Client Software を使う必要あり
・Redshift supports CloudHSM for database encryption and key management
・SSE-C 暗号化とともに使うのが良い選択肢
・IAM 許可
・CRUD an SHM Cluster
・CloudHSM Software
・Manage the Keys
・Manages the Users

・高可用性（Multi AZに配置される）

○**AWS KMS**
・Key Types
・Symmetric(AES-256)
・エンベロープ暗号化は必要ない
・Asymmetric(RSA&ECC key pairs)
・Public and Private Key pair
・Used for Encrypt/Decrypt , or Sign/Verify操作
・Use case：KMS APIを使えない外部ユーザーが暗号化をする場合

・Types of KMS Keys
・Customer Managed Keys
・Possibility of rotation policy
・Can add a Key Policy & Audit in CloudTrail
・Leverage for エンベロープ暗号化
・キーポリシーを変更できる
・AWS Managed Keys
・AWSサービスに使う
・AWSマネージド
・ビュー、トラック、audit（CloudTrail）できる
・キーポリシーは閲覧のみできる

・AWS Owned Keys
・AWSが作成および管理する
・複数のAWSアカウントに使える
・ビュー、トラック、auditできない
・キーポリシーは閲覧も変更もできない

・4KB以上はGenerate DataKey API

・KMS Key Policy
・Default KMS Key Policy
・ルートユーザーへのアクセス許可のみ
・IAMポリシーへのアクセス許可も追加する必要があり

・Custom KMS Key Policy
・Keyポリシーで明示的にユーザーの操作を許可している場合は、IAMポリシーの明示的許可は必要ない。
（別AWSアカウントの場合は、IAMポリシーにも許可が必要）
・KMS Grants
・一時的なアクセス許可付与のために使う
・AWSアカウント、IAMユーザー、ロールへのKMSへの許可を与える

・EC2 Image Builderが作成したKMSキーで暗号化されたフォレンジックAMIの複号

化の場合、
　　kms:Encrypt および kms:Decrypt の権限が必要。

○**KMS と HSM の違い**
・てナンシー：KMS はマルチテナント、HSM はシングルテナント
・Standard：KMS は FIPS140-2 Level 2、HSM は FIPS 140-2 Level 3
・Key Types：KMS は　Symmetric、Asymmetric、Digital Signing、HSM は Symmetric、Asymmetric、Digital Signing & Hashing
・Cryptographic Acceleration；KMS はなし、HSM は SSL/TLS Acceleration、Oracle TDE Acceleration
・Access Authentication：AWS IAM、 HSM はユーザーと管理の許可を作成
・Master Keys：KMS は AWS Owned Keys、AWS Managed Keys、 Customer Managed KMS Keys、HSM は Customer Managed CMK
・可用性：KMS は AWS マネージドサービス、HSM はマルチ AZ に配置される

○**AWS STS**
・AssumeRole
・AssumeRoleWithSAML
・AssumeRoleWithWebIdentity
・GetSessionToken

○**Cognito**
・Cognito User Pools(CUP)
　・ログイン：ユーザーネーム、password コンビネーション
　・e-mail、携帯番号認証
　・MFA
　・Federated 認証（Google, Facebook、SAML）
　・Login sends back a JSON Web Token(JWT)
　・データベースに保存したユーザーの認証
　・API Gateway と ALB と連携している。
　・
・Cognito Identity Pools(Federated Identities)
　・Get Identities for “user” so they obtain temporary AWS credentials
　・Your Identity Pool can include
　　・Public Providers(Google、Facebook、Apple)
　　・Users in an Amaozn Cognito User Pool
　　・OpenID Connect Providers & SAML Identity Provider
　　・Cognito Identity Pools allow for unauthorized User

○**AWS Verified Access**
・VPN なしで企業アプリケーションへの安全なネットワークアクセスを提供できる

○**Lambdaオーソライザー**
Lambdaを使用して実装されるカスタムな認可機構。
API Gatewayと組み合わせて使用され、APIリクエストの認可やアクセス制御を行うために利用される。
→3回認証に失敗したら、1時間使用不可になるようなロジック組み込み可能
・トークンベース
JWTやOAuthトークンなどのベアラートークンで発信者IDを受け取る
・リクエストパラメータ
ヘッダー、クエリ文字列パラメータ、stageVariablesおよび$contextの組み合わせで発信者IDを受け取る

○**AWS Managed Microsoft ADとAD Connectorの違いは？**
・AWS Managed Microsoft AD：
・AWSが管理・運用するフルマネージドなActive Directoryサービス
・ユーザーアカウントやグループ、ポリシーなどのユーザー管理機能がAWS Managed Microsoft ADで提供される。

・AD Connector
・オンプレミスのActive Directoryインフラストラクチャと AWSのリソースを連携させるためのプロキシサービス
・ユーザーはオンプレミスのADで管理される
・ ユーザーはオンプレミスのADに対して認証を行い、AD Connectorを介してAWSリソースにアクセスする。

○**Cognitoユーザープールと Cognito IDプール**
Cognitoユーザープール：ユーザーの認証と管理に特化しており、アプリケーションのユーザーアカウントを管理する。
Cognito IDプール：ユーザーの認証とアクセス制御に関連しており、AWSサービスへのアクセス権限を管理するための一意のIDを提供する。

○**Active Directoryと AWSの統合**
・Active Directoryを構成して、Active Directoryと AWSの間に証明書利用者信頼を追加する
・ActiveDirectoryに対する権限を持つActive Directoryのユーザーアカウントに対するIAMロールを作成する

○**IDプロバイダとアイデンティティストアの違い**
・IDプロバイダ：認証サービス
・アイデンティティストア：ユーザー情報を保管（AWS Managed Microsoft ADなど）

○**エンベロープ暗号化（Envelope Encryption）**
　・データの保護を目的として使用される暗号化の手法。
　・データ自体を直接暗号化するのではなく、データの暗号化に使用される鍵を暗号化して保護する。

○**AWS Black Belt**
・AWS Trusted Advisor【AWS Black Belt】
・Amazon Inspector 【AWS Black Belt】
・Amazon GuardDuty Malware Protection【AWS Black Belt】
・【AWS Black Belt Online Seminar】Amazon Detective
・【AWS Black Belt Online Seminar】Amazon Macie
・【AWS Black Belt Online Seminar】Amazon Cognito
・[AWS Black Belt Online Seminar] AWS Managed Microsoft AD
・【AWS Black Belt Online Seminar】AWSアカウント シングルサインオンの設計と運用