**AWS Services Checklist**

IAM Checklist:

🗹Avoid the use of "root" account.

🗹Ensure MFA is enabled for all the IAM users that have a console password.

🗹Ensure credentials unused for 90 days or greater are disabled.

🗹Ensure access keys are rotated every 90 days or less.

🗹Ensure IAM password policy requires at least one uppercase letter.

🗹Ensure IAM password policy requires at least one lowercase letter.

🗹Ensure IAM password policy requires at least one symbol.

🗹Ensure IAM password policy requires at least one number.

🗹Ensure IAM password policy requires a minimum length of 14 or greater.

🗹Ensure IAM password policy prevents password reuse.

🗹Ensure IAM password policy expires passwords within 90 days or less.

🗹Ensure no root account access key exists.

🗹Ensure MFA is enabled for the "root" account.

🗹Ensure hardware MFA is enabled for the "root" account.

🗹Ensure security questions are registered in the AWS account.

🗹Ensure IAM policies are attached only to groups or roles.

🗹Do not setup access keys during the initial user setup for all IAM users that have a console password.

🗹Ensure IAM policies that allow full ":" administrative privileges are not created.

🗹Ensure an IAM group for administration purposes is created.

🗹Follow the principle of least privilege while assigning permissions to users.

🗹If a user from another AWS account needs access, grant permission using roles.

EC2 Checklist:

🗹Use the latest AMI provided by Amazon AWS as they are more secure.

🗹Use strong and unique SSH key pairs to connect to your instances.

🗹Restrict SSH access by allowing connections only from trusted IP addresses or IP ranges.

🗹Disable password-based authentication for SSH.

🗹Apply the principle of least privilege by allowing only necessary ports and protocols.

🗹Regularly review and update security group rules.

🗹Enable multi-factor authentication.

🗹Avoid using root credentials for day-to-day tasks.

🗹Use VPC Flow Logs to monitor the traffic that reaches your instances.

🗹Restrict access to your instances using security groups.

🗹Use private subnets for instances that should not be accessed directly from the internet.

🗹Ensure data and disk volumes in EBS are encrypted with AES-256.

🗹Restrict inbound access to SSH, FTP, SMTP, MySQL, PostgreSQL, MongoDB, MSSQL, CIFS, etc., to required entities only.

🗹Use IAM roles to grant access to EC2, instead of access keys for temporary requirements.

🗹Make sure that no VPC endpoints are exposed by checking the principal value in the policy.

🗹Ensure no ACLs allow unrestricted inbound or outbound access.

S3 Checklist:

🗹Ensure that your Amazon S3 buckets use the correct policies and are not publicly accessible.

🗹Make use of object-level or bucket-level permissions in addition to IAM Policies to grant access to resources.

🗹Enable MFA Delete to prevent accidental deletion of buckets.

🗹Consider encryption of stored data, both server-side and client-side.

🗹Enable encryption of inbound and outbound data traffic through SSL endpoints.

🗹Configure S3 lifecycle management and versioning to deal with accidental deletions.

🗹Ensure S3 access logging is enabled.

🗹Constantly audit and monitor S3 buckets using CloudWatch metrics.

🗹Implement least privilege access.

🗹Consider using VPC endpoints for Amazon S3 access.

RDS Checklist:

🗹Configure appropriate database subnet groups.

🗹Disable public accessibility to your RDS instances.

🗹Configure network access control lists (ACLs).

🗹Configure security groups.

🗹Run the database on non-default ports.

🗹Enable encryption at rest for RDS instances.

🗹Implement appropriate database authentications.

🗹Restrict access to cloud users on Amazon RDS using IAM policies.

🗹Implement additional logging and monitoring.

🗹Enable automatic minor version upgrades.

🗹Follow client access guidelines.

🗹Implement encryption in transit using SSL/TLS.

🗹Implement least privilege model for DB users.

🗹Restrict operations using account resource limits.

🗹Record all activities on the database using audit logging.

ELB (Elastic Load Balancer) Checklist:

🗹Use HTTP/SSL.

🗹Configure ELB security policies to support strong cipher suites and protocols.

🗹Utilize AWS Web Application Firewall (WAF) in conjunction with the ELB for protection against common web attacks.

🗹Enable access logging on the ELB.

🗹Consider using AWS Shield Advanced for enhanced DDoS protection and real-time monitoring.

🗹Conduct periodic security assessments and reviews of your ELB configuration.

Lambda Checklist:

🗹Record all activities on the database using audit logging.

🗹Implement logging and audit trails for AWS Lambda.

🗹Use temporary AWS credentials.

🗹Define and categorize assets.

🗹Design an information security management system to protect assets.

🗹Configure security groups and network ACLs for Lambda functions.

🗹Monitor and analyze CloudWatch Logs for Lambda functions.

🗹Implement least privilege access control for Lambda functions.

🗹Restrict operations using account resource limits.

🗹ECS (Elastic Container Service) Checklist:

🗹Review and configure the cloud control panel settings, including access control and container images.

🗹Choose the right public or private Elastic Container Registry (ECR) for images.

🗹Configure Amazon ECR to scan images on push using basic or enhanced scanning.

🗹Use VPC Endpoint or security with known sources only.

🗹Implement compliance standards as per business requirements.

🗹Use Amazon Organization Service Control Policies (SCPs) to manage ECS tasks, creation, and region lock.

🗹Enable AWS CloudTrail and monitor for malicious ECS behavior by an identity in AWS.

🗹Enable AWS Config rules in the account and region of ECS.

🗹Use access control (roles or IAM users) for ECS clusters, services, and tasks.

🗹Use the most recent version of the ECR agent daemon on EC2.

🗹Do not store secrets/passwords in ECS Task Definitions.

🗹Use trusted container images only from ECR with no high/critical vulnerabilities.

🗹Limit the ability to SSH into EC2 containers to read-only file systems.

🗹Use API or GitOps to pull information for troubleshooting.

🗹Use Amazon CloudWatch to monitor malicious ECS configuration changes.

🗹Only use authorized container images.

EKS (Elastic Kubernetes Service) Checklist:

🗹Isolate Kubernetes nodes.

🗹Strengthen authentication and authorization.

🗹Take advantage of Kubernetes Roles-Based Access Control (RBAC).

🗹Avoid keeping secrets in an environment variable.

🗹Don't run containers in privileged mode.

🗹Don't share the host's IPC or network namespaces.

🗹Disable NET\_RAW.

🗹Check unsafe/proc mount.

🗹Don't use the root file system for container security.

🗹Build a rolling update strategy.

AWS CodeBuild Checklist:

🗹Utilize AWS Identity and Access Management (IAM) to control access to CodeBuild resources.

🗹Follow the principle of least privilege by granting only the necessary permissions to users and roles.

🗹Enable multi-factor authentication (MFA) for IAM users with access to CodeBuild.

🗹Store your source code in a secure and version-controlled repository, such as AWS CodeCommit or a trusted external source control system.

🗹Regularly update and patch the operating systems and software packages in your build environments.

🗹Use dedicated build environments for different projects to minimize the risk of code contamination.

🗹Restrict network access to build environments by placing them in private subnets and configuring security groups and network ACLs accordingly.

🗹Use separate build environments for different stages of the software development lifecycle to minimize the impact of security breaches.

🗹Avoid hard-coding sensitive information, such as API keys or credentials, directly in build specifications or source code.

🗹Enable logging for CodeBuild builds and configure logs to be stored centrally in services like Amazon CloudWatch Logs or Amazon S3.

🗹Stay up to date with the latest features, patches, and updates provided by AWS for CodeBuild.

🗹Implement build-time checks and tests for security-related requirements, such as secure coding practices and adherence to security policies.

🗹Encrypt environment variables that contain sensitive data in transit and at rest.

🗹Utilize code analysis tools and security scanners to identify vulnerabilities or potential security issues in your code.

🗹Stay up to date with the latest features, patches, and updates provided by AWS for CodeBuild.

AWS CodePipeline Checklist:

🗹Ensure IAM privileges are least privileged in CI/CD.

🗹Ensure the security of the CI/CD pipeline.

🗹Ensure the production environment does not allow manual changes or SSH access.

🗹Set up proper policies in GitHub for GitHub Actions.

🗹Allow GitHub Actions to create and approve pull requests.

🗹Require approval for every pull request before merging into the main branch.

DynamoDB Checklist:

🗹Enable encryption at rest for DynamoDB.

🗹Use IAM roles to authenticate access to DynamoDB.

🗹Use IAM policies for DynamoDB-based authorization.

🗹Use IAM policy conditions for fine-grained access control.

🗹Use a VPC endpoint and policies to access DynamoDB.

🗹Consider client-side encryption.

🗹Use AWS CloudTrail to monitor AWS managed KMS key usage.

🗹Tag your DynamoDB resources for identification and automation.

CloudFront Checklist:

🗹Ensure CloudFront distributions enforce HTTPS-only access for viewers.

🗹Verify that all traffic from CloudFront distributions to the origin server is encrypted.

🗹Enable Server Name Indication (SNI) for CloudFront distributions.

🗹Enable failover for CloudFront distributions.

🗹Use an Origin Access Identity (OAI) for CloudFront distributions.

🗹Do not allow deprecated SSL/TLS protocols.

🗹Configure a default root object for CloudFront distributions.

🗹Use a custom SSL/TLS certificate for CloudFront distributions.

🗹Associate CloudFront distributions with an AWS Web Application Firewall (WAF) web ACL.

🗹Enable access logs for CloudFront distributions.

EBS (Elastic Block Store) Checklist:

🗹Encrypt your EBS volumes using AWS Key Management Service (KMS).

🗹Implement AWS managed keys or bring your own key (BYOK) for encryption.

🗹Use encrypted Amazon Machine Images (AMIs) for launching instances that require encrypted EBS volumes.

🗹Utilize AWS Identity and Access Management (IAM) to control access to EBS resources.

🗹Follow the principle of least privilege by granting only necessary permissions to users and roles.

🗹Enable network-level encryption for data in transit using Amazon VPC traffic encryption.

🗹Utilize Amazon CloudWatch to monitor EBS volume metrics.

🗹Configure security groups to control inbound and outbound traffic to instances and EBS volumes.

🗹Limit access to only necessary ports and protocols required for your applications.

🗹Regularly review and update security group rules based on changing requirements.

VPC (Virtual Private Cloud) Checklist:

🗹Use AWS Identity and Access Management (IAM) to control access to your VPC and associated resources.

🗹Design your VPC architecture with appropriate network segmentation using subnets, route tables, and security groups.

🗹Implement the principle of least privilege by granting only necessary permissions to users and roles.

🗹Enable VPC Flow Logs to capture information about IP traffic flowing in and out of your VPC.

🗹Utilize security groups and network ACLs to control traffic flow between subnets and restrict access to VPC resources.

🗹Use secure protocols like SSL/TLS for traffic between clients and resources in the VPC.