# **AWS Services Provisioned by CloudFormation Template**

This table provides a detailed overview of the AWS services configured by your CloudFormation template, highlighting their purpose, hardware specifications, and interdependencies within the AI-SPM platform.

| **AWS Service** | **Hardware Configuration** | **Reason for Usage** | **Interacting Services** |
| --- | --- | --- | --- |
| **VPC** | CidrBlock: 10.0.0.0/16 | Provides an isolated network environment for the AI-SPM platform, ensuring secure communication between components and isolation from other AWS resources. | Internet Gateway, Public Subnets, Private Subnets, NAT Gateways, Security Groups, RDS, ECS, ALB |
| **Internet Gateway** | N/A | Enables internet access for public subnets, essential for NAT Gateways, the Application Load Balancer (ALB), and CloudFront distribution. | VPC, Public Subnets, Public Route Table |
| **Public Subnets (x2)** | CidrBlock: 10.0.1.0/24, 10.0.2.0/24 | Hosts public-facing resources like NAT Gateways and the Application Load Balancer (ALB) for external access. | VPC, Internet Gateway, Public Route Table, NAT Gateway, ALB |
| **Private Subnets (x2)** | CidrBlock: 10.0.3.0/24, 10.0.4.0/24 | Hosts private resources such as the RDS database and ECS Fargate tasks, ensuring they are not directly accessible from the internet. | VPC, NAT Gateway, Private Route Tables, RDS, ECS Fargate tasks |
| **Elastic IPs (EIPs) (x2)** | N/A | Provides static public IP addresses for the NAT Gateways, ensuring consistent outbound connectivity from private subnets. | NAT Gateway |
| **NAT Gateways (x2)** | N/A | Allows instances in private subnets to initiate outbound connections to the internet (e.g., for updates, external APIs) while preventing inbound connections. | Private Subnets, EIP, Private Route Tables |
| **Public Route Table** | N/A | Defines routing rules for public subnets, directing internet-bound traffic through the Internet Gateway. | Public Subnets, Internet Gateway |
| **Private Route Tables (x2)** | N/A | Define routing rules for private subnets, directing internet-bound traffic through the NAT Gateways. | Private Subnets, NAT Gateway |
| **ECSTaskExecutionRole (IAM Role)** | N/A | Grants Amazon ECS permissions to perform actions on your behalf, such as pulling container images from ECR and writing container logs to CloudWatch Logs. | ECS Cluster, ECR Repositories, CloudWatch Log Groups |
| **AppServerRole (IAM Role)** | N/A | Provides runtime permissions for the application containers (Node.js API Gateway, Python Microservices) to access other AWS services like Secrets Manager. | ECS Task Definitions, Secrets Manager |
| **AppServerInstanceProfile (IAM Instance Profile)** | N/A | A container for the AppServerRole, primarily included for potential future EC2-based deployments (currently unused with Fargate). | AppServerRole |
| **Secrets Manager** | N/A | Securely stores and manages sensitive configuration data, including database credentials, JWT secrets, API keys, and notification channel credentials. | ECS Task Definitions (via ValueFrom for environment variables), AppServerRole |
| **RDS PostgreSQL Instance** | DBInstanceClass: db.t3.micro (scalable), AllocatedStorage: 20Gi (scalable), Engine: postgres:14 | Provides a managed, highly available PostgreSQL database for all platform data (user accounts, AI assets, vulnerabilities, audit logs). | ECS Task Definitions (Node.js API Gateway, Python Microservices), RDSSecurityGroup |
| **ECR Repository (Node.js)** | Lifecycle Policy: Keep last 10 images, ScanOnPush: true | Stores Docker images for the Node.js API Gateway, with automated vulnerability scanning on push. | APIGatewayTaskDefinition |
| **ECR Repository (Python)** | Lifecycle Policy: Keep last 10 images, ScanOnPush: true | Stores Docker images for the Python Microservices, with automated vulnerability scanning on push. | MicroservicesTaskDefinition |
| **ECS Cluster** | N/A | A logical grouping of containerized services, providing the foundation for scalable container management using AWS Fargate. | ECS Services, ECS Task Definitions |
| **APIGatewayTaskDefinition (ECS Task Definition)** | Cpu: 512 (0.5 vCPU), Memory: 1024 (1 GB RAM) | Defines the container configuration, resource requirements, and environment variables for the Node.js API Gateway service. | APIGatewayService, ECR Repository (Node.js), Secrets Manager, CloudWatch Log Group (APIGatewayLogGroup), RDS PostgreSQL |
| **MicroservicesTaskDefinition (ECS Task Definition)** | Cpu: 1024 (1 vCPU), Memory: 2048 (2 GB RAM) | Defines the container configuration and resources for the Python Microservices (AI Scanner, Data Integrity, Wiz Integration, Compliance Engine). | MicroservicesService, ECR Repository (Python), Secrets Manager, CloudWatch Log Group (MicroservicesLogGroup), RDS PostgreSQL |
| **APIGatewayLogGroup (CloudWatch Log Group)** | RetentionInDays: 14 | Centralized storage for logs generated by the Node.js API Gateway, aiding in debugging and operational monitoring. | APIGatewayTaskDefinition |
| **MicroservicesLogGroup (CloudWatch Log Group)** | RetentionInDays: 14 | Centralized storage for logs generated by the Python Microservices. | MicroservicesTaskDefinition |
| **SecurityLogGroup (CloudWatch Log Group)** | RetentionInDays: !Ref LogsRetentionDays (default 30), KmsKeyId: !Ref LogEncryptionKey | Stores security events (authentication, authorization, threat detection) for analysis and auditing, with encryption at rest. | KMS Key |
| **AuditLogGroup (CloudWatch Log Group)** | RetentionInDays: 2555 (7 years), KmsKeyId: !Ref LogEncryptionKey | Stores immutable audit trails for compliance purposes (e.g., SOX, HIPAA, GDPR), with strong encryption and long retention. | KMS Key |
| **PerformanceLogGroup (CloudWatch Log Group)** | RetentionInDays: !Ref LogsRetentionDays (default 30) | Stores performance metrics (e.g., API response times, database query performance) for application and infrastructure optimization. | N/A (application components push to it) |
| **AgenticLogGroup (CloudWatch Log Group)** | RetentionInDays: !Ref LogsRetentionDays (default 30), KmsKeyId: !Ref LogEncryptionKey | Stores logs related to AI agent behavior, decision tracking, and anomaly detection, with encryption at rest due to potential sensitive data. | KMS Key |
| **MonitoringLogGroup (CloudWatch Log Group)** | RetentionInDays: !Ref LogsRetentionDays (default 30) | Stores health checks, metrics collection, and alerting events for overall system observability. | N/A (application components push to it) |
| **LogEncryptionKey (KMS Key)** | N/A | A Customer Master Key (CMK) used to encrypt sensitive log data at rest within CloudWatch Log Groups, enhancing data protection. | SecurityLogGroup, AuditLogGroup, AgenticLogGroup |
| **LogEncryptionKeyAlias (KMS Alias)** | N/A | Provides a human-readable alias for the KMS encryption key, simplifying key management and referencing. | LogEncryptionKey |
| **APIGatewayService (ECS Service)** | DesiredCount: 2 | Manages and maintains the desired number of running tasks for the Node.js API Gateway, ensuring high availability and scalability. | ECS Cluster, APIGatewayTaskDefinition, ALB Target Group, Service Discovery |
| **MicroservicesService (ECS Service)** | DesiredCount: 1 | Manages and maintains the desired number of running tasks for the Python Microservices, ensuring their availability and scalability. | ECS Cluster, MicroservicesTaskDefinition, Service Discovery |
| **ServiceDiscoveryNamespace** | N/A | A private DNS namespace within the VPC, enabling internal service discovery for microservices (e.g., ai-scanner.ai-spm-microservices.local). | Service Discovery Services (api-gateway, ai-scanner, etc.) |
| **Service Discovery Services (x5)** | N/A | Registers each microservice (API Gateway, AI Scanner, Data Integrity, Wiz Integration, Compliance Engine) within the private DNS namespace for internal resolution. | ECS Services, ServiceDiscoveryNamespace |
| **S3 Frontend Bucket** | Private Access, Versioning Enabled | Securely hosts static frontend assets (HTML, CSS, JavaScript) for the web application, with public access blocked and versioning for recovery. | CloudFront Distribution, CloudFront Origin Access Control |
| **CloudFront Origin Access Control (OAC)** | N/A | Provides a secure way for CloudFront to access content in your S3 bucket without making the bucket publicly accessible. | S3 Frontend Bucket, CloudFront Distribution |
| **CloudFront Distribution** | N/A | Acts as a Content Delivery Network (CDN) to serve frontend assets globally with low latency and routes API requests to the ALB, providing HTTPS termination and security headers. | S3 Frontend Bucket, CloudFront Origin Access Control, ALB (for API routing), CloudFrontResponseHeadersPolicy |
| **CloudFrontResponseHeadersPolicy** | Strict-Transport-Security: 31536000s, X-Content-Type-Options: nosniff, X-Frame-Options: DENY | Applies essential security headers to all responses served via CloudFront, enhancing browser security against common web vulnerabilities. | CloudFront Distribution |
| **ALB Security Group** | Ingress: TCP 80/443 from 0.0.0.0/0 | Controls inbound traffic to the Application Load Balancer, allowing HTTP and HTTPS access from the internet. | Application Load Balancer |
| **Application Load Balancer (ALB)** | N/A | Distributes incoming application traffic across multiple targets (ECS tasks), providing high availability and fault tolerance. | Public Subnets, ALB Security Group, ALB Listeners, ALB Target Group |
| **ALB Target Group** | Protocol: HTTP, Port: 5000, HealthCheckPath: /api/health | A logical grouping of targets (ECS tasks) that the ALB routes requests to, with defined health checks to ensure traffic is sent only to healthy instances. | Application Load Balancer, APIGatewayService |
| **ALB Listener HTTP** | Port: 80, Protocol: HTTP | Listens for incoming HTTP traffic on port 80 and redirects it to HTTPS (port 443) if an SSL certificate is configured. | Application Load Balancer, ALB Target Group |
| **ALB Listener HTTPS** | Port: 443, Protocol: HTTPS, SSL Policy: ELBSecurityPolicy-2016-08 | Listens for incoming HTTPS traffic on port 443, using the provided SSL certificate for encryption, and forwards requests to the ALB Target Group. | Application Load Balancer, ALB Target Group, SSLCertificateArn |