darwinbox

Darwinbox

Security White Paper

July 2020

Table of Contents

INTRODUCTION

Darwinbox is a certified secure and compliant HRIS platform with SOC-2 Type 2, ISO 27001 and GDPR.

We accomplish this by performing state-of-the-art application designs, security products, and cloud platform facilities, and are thus able to deliver the same level of security in our SaaS offering as any of the largest and best providers. We use the same cloud capabilities, the same security products and algorithms, the same blueprints, and the same operation procedures. We even use the same methodology for representing these capabilities.

The standard way that security, privacy, reliability, availability, integrity, scalability, and recoverability of the solution are represented is through compliance. Like requirements and implementation guides, compliance profiles have a list of "controls." We compare our SaaS solution -- including its technology and operations -- against these controls. The checklist covers everything from technology design and deployment to the day-to-day operational aspects of policies and procedures.

GDPR Compliant

As mentioned, these checklists are provided by industry and standards organisations such as CSA and BSI. These two are more focused on technology. There are other profiles more focused on operations. One popular one for SaaS offerings is from the industry organisation called Shared Assurances, and the compliance profile is called the Standard Information Gathering ("SIG"). O. They bring us the SOC compliance profile (Service Organization Controls). Another popular one comes from the International Standards Organization (ISO), called ISO 27001.

Nowadays, when enterprises want to understand a SaaS offering's assurances around security, privacy, reliability, availability, integrity, scalability, and recoverability, they will ask for an "attestation" or "certification" to one or more of the above compliance profiles.

This paper sets out the Darwinbox directions on compliance and explains our technical and operations approach in many important areas of security, privacy, reliability, availability, integrity, scalability, and recoverability.

Your Data 

Darwinbox employs a multi-tier, multi-datacenter data-ingestion pipeline to process log data securely. The Darwinbox ingestion pipeline includes network devices that are able to process data from multiple inputs on both TCP (encrypted) and UDP as well as via active collection from third-party repositories such as S3 buckets

Once data is collected, it is tagged to identify the specific customer and then processed with additional metadata. The data is then indexed into a dedicated data-store that is able to scale and grow to fit any needed capacity of customers data. All data is universally available across multiple data centers and is backed up to ensure data availability.

Your data does not leave our application and never leaves the cloud; it is only accessed through the application by a properly credentialed and entitled user. Data as it is stored in our system is always stored in a cryptographically encrypted form, as will be explained below

The security, privacy, reliability, availability, integrity, scalability, and recoverability of your data therefore only depends on the Darwinbox.com application as well as the cloud platform. Let us consider both of them

The AWS Cloud Platform

Darwinbox runs on Amazon Web Services (AWS). Cloud security is one of AWS' highest priorities. As an AWS customer, we – and therefore our enterprise customers - benefit from a data center and network architecture built to meet the requirements of the most security-sensitive organizations. AWS hosts many solutions, applications, and websites around the world. They have a massive security team and state-of-the-art security software and hardware.

One of the key features of AWS is that it enables us to implement business continuity with replication between applications and data across multiple data centers in the same region using availability zones. While doing so, we also retain complete control and ownership over the region in which our data is physically located, making it easy to meet regional compliance and data residency requirements.

The AWS infrastructure puts strong safeguards in place to help protect customer privacy. All data is stored in highly secure AWS data centers. AWS manages dozens of compliance programs in its infrastructure. This means that all of the datacenter related portions of our compliance profiles have already been completed. This allows us to maintain the highest standard of security without having to manage our own facility.

Finally, the AWS security design is such that security scales with our AWS cloud usage. No matter the size of our business, the AWS infrastructure is designed to keep data safe.

AWS provides several security capabilities and services to increase privacy and control network access. These include:

1. Network firewalls built into Amazon VPC and web application firewall capabilities in AWS WAF let us create private networks and control access to instances and applications
2. Encryption in transit with TLS across all services

Availability is of paramount importance in the cloud. AWS customers benefit from AWS services and technologies built from the ground up to provide better resilience in the face of DDoS attacks. We use a combination of AWS services to implement a defensive in-depth strategy and thwart attacks. AWS offers a range of tools to allow us to move fast while still ensuring that our cloud resources comply with organizational standards and best practices.

AWS services include:

1.Two security assessment services: Amazon Inspector and Amazon Trusted Advisor. Inspector focuses on servers while Advisor focuses on networks, storage, and user access. These tools automatically assess

2.Deployments for vulnerabilities or deviations from best practices including impacted networks, OS, and attached storage Deployment tools to manage the creation and decommissioning of AWS resources according to organization standards

3.Inventory & config. management tools, including AWS Config, that identify AWS resources and then track and manage changes to those resources over time.

4.Template definition and management tools, including AWS Kubernetes Service and AWS CloudFormation, to create standard, pre-configured environments.

AWS provides tools and features that enable us to see exactly what is happening in our AWS environment. This includes:

1. Deep visibility into API calls through AWS CloudTrail, including who, what, and from where calls were made

2. Alert notifications through Amazon CloudWatch when specific events occur, or thresholds are exceeded

3. Log aggregation options, investigation streamlining, and compliance reporting

These tools and features give us the visibility we need to spot issues before they impact our application. In fact, we use our own application for log analysis!

Compliance

Hosting our SaaS infrastructure on AWS creates a shared responsibility model between Darwinbox.com and AWS. This shared model reduces our operational burden as AWS operates, manages, and controls the components from the host operating system and virtualisation layer down to the physical security of the facilities in which the services operate.

Because the AWS cloud infrastructure comes with so many built-in security features, we can simply focus on the security of our guest operating system (including updates and security patches), our application software as well as configuration of the AWS-provided security features such as firewalls and configuration monitors.

Here are more details on the 4 compliance profiles (SOC 1, CSA STAR, SOC 2, and ISO 27001) we support.

SOC 1/SSAE 16/ISAE 3402 (formerly SAS 70)

These specifications are physical datacenter related; therefore, they are implemented by Amazon AWS and we automatically inherit them.

Datacenter SOC 1 / SSAE 16 / ISAE 3402

Increasingly, businesses outsource basic functions such as data storage and access to applications to cloud service providers (CSPs) and other service organizations. In response, the American Institute of Certified Public Accountants (AICPA) has developed the Service Organization Controls (SOC) framework, a standard for controls that safeguard the confidentiality and privacy of information stored and processed in the cloud. This aligns with the International Standard on Assurance Engagements (ISAE), the reporting standard for international service organizations.

Service audits based on the SOC framework fall into two categories—SOC 1 and SOC 2—that apply to Darwinbox cloud services. SOC 2 is covered later in this paper; at first consider SOC 1.

A SOC 1 audit evaluates the effectiveness of a CSP's internal controls that affect the financial reports of a customer using the provider's cloud services. The Statement on Standards for Attestation Engagements (SSAE 16) and the International Standards for Assurance Engagements No. 3402 (ISAE 3402) are the standards under which the audit is performed and is the basis of the SOC 1 report. The controls for a SOC 1 audit are entirely focused on traditional datacenter types of concerns, that is, controls around data center physical security, availability of operational systems such as power and air conditioning, and so on. As such, by hosting on AWS, Darwinbox enjoys the SOC 1 compliance of the AWS datacenters.

AWS's data centers are state of the art, utilizing innovative architectural and engineering approaches. Amazon has many years of experience in designing, constructing, and operating large-scale data centers. This experience has been applied to the AWS platform and infrastructure. AWS data centers are housed in nondescript facilities. Physical access is strictly controlled both at the perimeter and at building ingress points by professional security staff utilizing video

AWS provides data center access and information only to employees and contractors who have a legitimate business need for such privileges. When an employee no longer has a business need for these privileges, his or her access is immediately revoked, even if they continue to be an employee of Amazon or Amazon Web Services. All physical access to data centers by AWS employees is logged and audited routinely. Employees with physical access do not have logical access. Authorized staff must pass two-factor authentication a minimum of two times to access data center floors. All visitors and contractors are required to present identification and are signed in and continually escorted by authorized staff. AWS production network is segregated from the Amazon corporate network and requires a separate set of credentials for logical access. The Amazon corporate network relies on user IDs, passwords, and Kerberos, while the AWS production network requires SSH public-key authentication through a bastion host. AWS developers and administrators on the Amazon corporate network must explicitly request access through the AWS access management system to access AWS cloud components. All requests are reviewed by appropriate personnel prior to approval.

AWS data center's electrical power systems are designed to be fully redundant and maintainable without impact to operations, 24 hours a day, and seven days a week. Uninterruptible Power Supply (UPS) units provide back-up power in the event of an electrical failure for critical and essential loads in the facility. Data centers use generators to provide back-up power for the entire facility.

AWS monitors electrical, mechanical, and life support systems and equipment so that any issues are immediately identified. Preventative maintenance is performed to maintain the continued operability of equipment. The Amazon incident management team employs industry-standard diagnostic procedures to drive resolution during business-impacting events. Staff operators provide 24 x 7

When a storage device has reached the end of its useful life, AWS procedures include a decommissioning procedure that is designed to prevent customer data from being exposed to unauthorized individuals. AWS uses the techniques detailed in DoD 5220.22-M ('National Industrial Security Program Operating Manual") or NIST 800-88 ("Guidelines for Media Sanitization") to destroy data as part of the decommissioning process. All decommissioned magnetic storage devices are degaussed and physically destroyed in accordance with industry-standard practices. Based on the shared responsibility model, here are some examples of inherited, hybrid, and shared controls. AWS is responsible for all inherited controls. Darwinbox needs to build on AWS' offerings and do its part for meeting hybrid and shared controls

Inherited Controls – Examples include:

1. Media Protection	2. Maintenance	3. Physical & Environmental Control

Compliance

Hybrid Controls – Examples include:

- Account Management where the AWS IAM service enables customers to securely control access to AWS services and resources
- Resiliency (business continuity / disaster recovery), where AWS provides the architecture (regions and availability zones) and services which can be used to recover systems during an event.

Shared Controls – Examples include:

1. Patch Management in which AWS is responsible for patching and fixing flaws within the infrastructure, but customers are responsible for patching their guest OS and applications
2. Awareness & Training in which AWS trains AWS employees, but customers must train their own employees.
3. Configuration Management in which AWS maintains the configuration of its infrastructure devices, but customers are responsible for configuring their own guest operating systems, databases, and applications.

Configuration Management in which AWS maintains the configuration of its infrastructure devices, but customers are responsible for configuring their own guest operating systems, databases, and applications.

Configuration Management in which AWS maintains the configuration of its infrastructure devices, but customers are responsible for configuring their own guest operating systems, databases, and applications.

Cloud Security Alliance (CLOUD CONTROLS MATRIX)

The Cloud Security Alliance Cloud Controls Matrix (CCM) is specifically designed to provide fundamental security principles to guide cloud vendors and to assist prospective cloud customers in assessing the overall security risk of a cloud provider.

CCM™
Cloud Controls Matrix

The Cloud Security Alliance (CSA) is the world's leading organization dedicated to defining and raising awareness of best practices to help ensure a secure cloud computing environment.

The CSA CCM provides a controls framework that gives detailed understanding of security concepts and principles that are aligned to the Cloud Security Alliance guidance in 13 domains. The foundations of the Cloud Security Alliance Controls Matrix rest on its customized relationship to other industry-accepted security standards, regulations, and controls frameworks such as the ISO 27001/27002, ISACA COBIT, PCI, NIST, Jericho Forum and NERC CIP and will augment or provide internal control direction for service organization control reports attestations provided by cloud providers. As a framework, the CSA CCM provides organizations with the needed structure, detail and clarity relating to information security tailored to the cloud industry. The CSA CCM strengthens existing information security control environments.

As a control's framework, the CSA CCM provides organizations with the needed structure, detail and clarity relating to information security tailored to cloud computing. See more about CCM at https://cloudsecurityalliance.org/research/ccm/. Darwinbox uses the CCM as the central control set for our

InfoSec program. These controls are called "Matrix" because they are mapped to many other compliance program control frameworks

The CCM is mapped to:

- ✓ AICPA SOC 2 2014 Trust Services Criteria
- ✓ Canada PIPEDA (Personal Information Protection Electronic Documents Act)
- ✓ COBIT 5.0
- ✓ COPPA (Children's Online Privacy Protection Act)
- ✓ CSA Enterprise Architecture
- ✓ ENISA (European Network Information and Security Agency) Information Assurance
- ✓ Framework
- ✓ European Union Data Protection Directive 95/36/EC
- ✓ FERPA (Family Education and Rights Privacy Act)
- ✓ HIPAA/HITECH act and the Omnibus Rule
- ✓ ISO/IEC 27001:2013
- ✓ ITAR (International Traffic in Arms Regulation)
- ✓ Mexico – Federal Law on Protection of Personal Data Held by Private Parties
- ✓ NIST SP 800-53 Rev 3 Appendix J
- ✓ NZISM (New Zealand Information Security Manual)
- ✓ ODCA (Open Data Center Alliance) Usage Model PAAS Interoperability Rev. 2.0
- ✓ PCI DSS v3

As indicated in bold, the compliance frameworks discussed in detail in this whitepaper, which are of primary focus to Darwinbox, can easily be rationalized as needed in the future, because of our use of the CSA CCM. Darwinbox will be publishing our complete responses, based on the CCM Controls, in the form of the CSA CAIQ, by officially registering them with the CSA STAR program.

SOC 2

A SOC 2 audit gauges the effectiveness of a CSP's system, based on the AICPA Trust Service Principles and Criteria. SOC 2 reports specifically address one or more of the following five key system domains:

- **Security** – The system is protected against both physical and logical unauthorized access
- **Availability** – The system is available for operation and use as committed or agreed
- **Processing integrity** – System processing is complete, accurate, timely and authorized
- **Confidentiality** – Information designated as confidential is protected as committed or agreed

These reports are intended to meet the needs of a broad range of users that need to understand internal control at a service organization as it relates to security, availability, processing integrity, confidentiality, and privacy.

- **Privacy** – Personal information is collected, used, retained, disclosed, and destroyed in conformity with the commitments in the entity's privacy notice and with criteria set forth in generally accepted privacy principles (GAPP) issued by the AICPA.

SOC 2 involves the same types of technical and operational controls that the above compliance profiles do, however the SOC 2 process includes a very formal requirement around "corporate" policies and procedures. More specifically, SOC 2 lays out requirements for service organizations around having documented policies and procedures in place, specifically that of information security and operational specific policies. While Darwinbox has all of these in place, a SOC 2 audit requires a detailed checklist that for each paragraph of those policies and procedures, the company show evidence of adhering to them, in change logs, meeting minutes, all the way to design documents and bug reports. And it must be shown that these processes have been followed for some time, usually 6 months.

Darwinbox is SOC 2 Type 2 certified.

ISO 27001 / 22301

ISO 27001 is a security management standard that specifies security management best practices and comprehensive security controls following the ISO 27002 best practice guidance. This is a widely recognized international security standard. ISO 22301 is a business continuity plan which adds to these requirements.

ISO 27001 (formally known as ISO/IEC 27001:2005) is a specification for an information security management system (ISMS). An ISMS is a framework of policies and procedures that includes all legal, physical, and technical controls involved in an organization's information risk management processes.

Much like SOC 2, ISO 27001 requires the same types of technical and operational controls that the above compliance profiles do. However, the ISO 27001 process includes a very formal requirement around "corporate" policies and procedures and around "continuous improvement." It specifies that companies cover a self-evaluation process which judges and improves the suitability and adequacy of the information security management system (ISMS) as well as its effectiveness. Most companies use a traditional Plan-Do-Check-Act (PDCA) approach for continual improvement, but other methodologies such as Lean, Kaizen, and Six Sigma can now also be used. ISO 27001 requires that this be in place and one "cycle" of improvement be demonstrated before certification can be granted. As such, it takes the longest of all the compliance profiles to achieve.

Darwinbox is ISO 27001 audited and certified.

ISO 22301 is a related specification which Darwinbox considers quite important. ISO 22301 requires us to plan, establish, implement, operate, monitor, review, maintain and continually improve a documented management system to protect against, reduce the likelihood of occurrence, prepare for, respond to, and recover from disruptive incidents when they arise. In other words, it is a formalized and deep Business

Continuity Plan (BCP). Darwinbox is ISO 27001 compliant. In conjunction with the Darwinbox ISO 27001 program, we are including an ISO 22301 program.

Security and Risk Management 

We have just reviewed in detail the Darwinbox compliance programs. These programs use standardized compliance profiles because these are the ways which the industry has developed to express assurances around software-based services. These compliance profiles require a certain amount of formality and rigor around how we work, how we develop and deploy our solutions, and how we operationally run our service.

Behind this formality, we have taken specific actions in the delivery of our services to fulfill these requirements**. The following sections describe, in "non-formal terms," some of the actions we have taken to ensure the following:**

- Our systems are designed with security capabilities. They will withstand security attacks of several forms, from denial of service to vulnerability exploits to malware.
- We build them with professional, documented processes. Important decisions are made carefully and are documented. Mistakes can be easily detected and reversed.
- We respect our customers' data in our stewardship of it as well as in their privacy. In no case will we lose their hard-earned work.
- Our systems will be universally available, so they can count on using them when they need them. If there are unforeseen problems such as hardware or network failures, our platform will respond and take actions to continue working.
- Our people are of the highest integrity; everyone knows their job and knows how to expedite processes through the company as needed to deliver our services.
- We regularly monitor and test our systems against attacks and poor performance and to make sure preventative mechanisms are working.

Secure Software Design Lifecycle

The Security Development Lifecycle (SDL) is a security assurance process that is focused on software development. As a company-wide initiative and a mandatory policy, we "design in" security from the start. At Darwinbox, the SDL is based on three core concepts—education, continuous process improvement, and accountability.

Each member of a project team must be educated in security concepts and development. Additionally, in the interests of staying lean, engineers and testers performing security-related tasks or SDL-related tasks should acquire relevant security knowledge prior to performing the tasks on the sprint. In this case, relevant is defined as security concepts that are pertinent to the features developed or tested during the sprint.

Web-based applications

- Cross-site scripting (XSS) vulnerabilities
- SQL injection vulnerabilities

Database applications

- SQL injection vulnerabilities

Programming

- ✓ Buffer overflows
- ✓ Integer overflows
- ✓ Input validation
- ✓ Language-specific issues (PHP, Java, C#)

Cryptographic code

- Common cryptographic errors

Access Management, Entitlements Control

Access control definitions are mostly geared towards restricting access to information and information processing facilities. Their main objective is to ensure authorized user access and to prevent unauthorized access to systems, applications, and services.

We guard our AWS infrastructure very carefully. To illustrate, our AWS "account" is the account that we created when we first signed up for AWS. The email address and password specified during the signup process were used to create our AWS account, which represents a business relationship between us and AWS. This account, a.k.a. root account, provides us with full access to all resources in our AWS account.

Identity and access management is one key requirement that is included in most information security and compliance standards. Although most of these standards have similar control objectives (or definitions) for identity and access management, it may be referred to as identity management, user management, access controls, or authentication.

However, following AWS recommendations, we are NOT to not use this account for day-to-day operations. We store the AWS root account credentials in a safe place and instead use AWS Identity and Access Management (IAM) user credentials for day-to-day interaction with AWS. To add an extra layer of security, we use Multi-Factor Authentication (MFA) on the root account

Continuing on Accounts, we create multiple AWS accounts for our organizations -- separate accounts, for example, for production resources and backup resources. This separation allows us to cleanly separate different types of resources and provide excellent security benefits.

AWS Identity and Access Management (IAM) is a web service that helps us securely control access to AWS resources for our users. We use IAM to control who can use our AWS resources (authentication) and what resources they can use and in what ways (authorization). We use IAM to create additional users and assign permissions to these users following the least privilege principle. With IAM, we can securely control access to AWS services and resources for users in our AWS account. For example, as we require administrator-level permissions, we can create an IAM user, grant that user full access, and then use those credentials to interact with AWS. Later, as we need to revoke or modify permissions, we can delete or modify any policies that are associated with that IAM user. Additionally, if we have multiple users that require access to our AWS account, we can create unique credentials for each user and define who has access to which resources

In addition to manually creating individual passwords for our IAM users, we follow a password policy that applies to all IAM user passwords in our AWS account.

Penetration and Vulnerability Testing

A penetration test (or pen test) is a set of procedures designed to bypass the security controls of an IT system in order to test that system's resistance to attack. Essentially it is a form of ethical hacking to identify and assess vulnerabilities in your computer systems before they are found by the wrong people.

We perform regular pen testing to provide management with confidence that Darwinbox systems are secure from attack and provide reassurance to customers that their data is adequately protected. A pen test is a key requirement in determining whether security policies are effective and is also essential for compliance.

AWS Trusted Advisor and AWS Inspector

AWS Trusted Advisor is an application that draws upon best practices learned from AWS' aggregated operational history of serving hundreds of thousands of AWS customers. Trusted Advisor inspects our AWS environment and makes recommendations for saving money, improving system performance, or closing security gaps. Trusted Advisor is available in the AWS Management Console.

Amongst other things, Trusted Advisor includes a list of checks of interest in this document:

- Security – identification of security settings that could make our AWS solution less secure.
- Fault Tolerance – recommendations that help increase the resiliency of your AWS solution by highlighting redundancy shortfalls, current service limits, and over utilized resources.

Trusted Advisor checks security groups for rules that allow unrestricted access (0.0.0.0/0) to specific ports or to a resource. Unrestricted access increases opportunities for malicious activity (hacking, denial-of-service attacks, and loss of data). The ports with highest risk are flagged red, and those with less risk are flagged yellow. Ports flagged green are typically used by applications that require unrestricted access, such as HTTP and SMTP.

Trusted Advisor also checks for our use of AWS IAM and also checks the root account and warns if MFA is not enabled.

For Amazon S3, Trusted Advisor checks buckets that have open access permissions. Bucket permissions that grant upload/delete access to everyone create potential security vulnerabilities by allowing anyone to add, modify, or remove items in a bucket. This check examines explicit bucket permissions, but it does not examine associated bucket policies that might override the bucket permissions.

Trusted Advisor also checks the password policy for our account and warns when a password policy is not enabled, or if password content requirements have not been enabled.

Password content requirements have not been enabled. Password content requirements increase the overall security of our AWS environment by enforcing the creation of strong user passwords. When we create or change a password policy, the change is enforced immediately for new users but does not require existing users to change their passwords.

Amazon Inspector analyzes an application's configuration and activity, looking for a wide spectrum of possible vulnerabilities across Amazon EC2 instances, and collecting information such as how the application communicates with other AWS services, whether it uses secure channels, and the network traffic between instances.

Amazon Inspector compares this information against AWS's extensive rules packages, which represent thousands of potential security vulnerabilities that AWS continuously updates with the latest threat intelligence. Once an assessment of the application's Amazon EC2 environment is completed, we can view the findings, along with detailed recommendations for remediation, in the Amazon Inspector console.

External Network Penetration Test

A Network Pentest (also known as an External Network Security Assessment) is an external penetration test that identifies the vulnerabilities of computer systems through their exposure to the Internet. **We perform the following Network Penetration Tests on the following devices and systems:**

- FTP Servers
- HTTP/HTTPS Servers
- Mail Servers
- Intranet/Extranet Servers
- DNS Servers
- Internet Routers Firewalls
- IDS/IPS
- VPN Servers

Overall Web Application Penetration Test

We also test the various Darwinbox websites, extranets, and intranets for application layer vulnerabilities. Our applications are tested for:

- Information disclosure
- Privilege escalation
- SQL injection
- Cross-site scripting
- Cross-site request forgery
- Access control issues
- Other issues

Data Encryption, Data Privacy

First of all, AWS has strategically placed a limited number of access points in the cloud to allow for a more comprehensive monitoring of inbound and outbound communications and network traffic. These customer access points are called API endpoints, and they allow secure HTTP access (HTTPS), which allows us to establish a secure communication session with our storage or compute instances within AWS.

AWS has implemented network devices that are dedicated to managing interfacing communications with internet service providers (ISPs). AWS employs a redundant connection to more than one communication service at each internet-facing edge of the AWS network. These connections each have dedicated network devices. We can connect to an AWS access point via SSL, a cryptographic protocol that is designed to protect against eavesdropping, tampering, and message forgery.

Data in Transit

Services from AWS provide support for both IPsec and SSL/TLS for protection of data in transit. IPsec is a protocol that extends the IP protocol stack, often in network infrastructure, and allows applications on upper layers to communicate securely without modification. SSL/TLS, on the other hand, operates at the session layer, and while there are third-party SSL/TLS wrappers, it often requires support at the application layer as well.

The AWS Management Console uses SSL/TLS between the client browser and console service endpoints to protect AWS service management traffic. Traffic is encrypted, data integrity is authenticated, and the client browser authenticates the identity of the console service endpoint by using an X.509 certificate. After an SSL/TLS session is established between the client browser and the console service endpoint, all subsequent HTTP traffic is protected within the SSL/TLS session.

We also use AWS APIs to manage services from AWS either directly from applications or third-party tools, or via SDKs, or via AWS command line tools. AWS APIs are web services (SOAP or REST) over HTTPS. SSL/TLS sessions are established between the client and the specific AWS service endpoint, depending on the APIs used, and all subsequent traffic, including the SOAP/REST envelope and user payload, is protected within the SSL/TLS session.

SSH is the preferred approach for establishing administrative connections to Linux servers. SSH is a protocol that, like SSL, provides a secure communications channel between the client and the server. In addition, SSH also supports tunneling, which we use when protecting the application session in transit.

As a security best practice, we use HTTPS protocol to protect our data intransit. The SSL encryption and decryption operations can be handled by our EC2 instances. We handle SSL termination using Elastic Load Balancing (ELB) which supports SSL termination. ELB can centrally manage our SSL certificates, and also encryption to back-end instances with optional public key authentication.

We load balance HTTP/HTTPS applications and sometimes use layer 7- specific features, such as X-Forwarded and sticky sessions. We also use strict layer 4 load balancing for applications that rely purely on the TCP protocol.

It is equally important to encrypt data "at-rest". This ensures the security of the data even if an unauthorized party gains access to it. Encrypting data at-rest makes it much more difficult for an attacker to compromise data, even if they are able to compromise an endpoint.

Data at Rest

Our source data comes from either systems we connect to or an Amazon EC2 instance. We can upload that data over a secure HTTPS connection to any of the AWS services that support automatic server-side encryption. The service endpoint handles the encryption and key management processes for us.

For example, Amazon S3 or EBS supports server-side encryption (SSE) of user data. Server-side encryption is transparent to us. Our database uses EBS storage so all database storage is encrypted in this way.

Encryption is one of the most widely used techniques for protecting data at rest. AWS server-side encryption encrypts data on our behalf "after" the API call is received by the service, leveraging AWS KMS. We do not have to worry about managing and rotating the keys as AWS automatically manages them for us.

The SSE-S3 or EBS process is where Amazon manages the encryption keys for us. In this option, AWS generates a unique encryption key for each object, and then encrypts the object using AES-256. The encryption key is then encrypted itself using AES-256-with a master key that is stored in a secure location. The master key is rotated on a regular basis.

Data Integrity

To ensure that data integrity is not compromised through deliberate or accidental modification, we use resource permissions to limit the scope of users who can modify the data. Even with resource permissions, accidental deletion by a privileged user is still a threat. We perform data integrity checks, such as Message Authentication Codes (SHA-1/SHA-2) or Hashed Message Authentication Codes (HMACs), digital signatures, or authenticated encryption (AES-GCM) to detect data integrity compromise on key pieces of data. If we detect data compromise, we restore the data from backup or restore to a previous version

Configuration / Change Control

To ensure implementation of consistent security controls for all our applications running in the AWS cloud, we build standardized, automated, and repeatable architectures that can be deployed for common use cases. Automation helps us easily meet the foundational requirements for building a secure application in the AWS cloud, while providing a level of uniformity that follows proven best practices. We use a technique known as "Infrastructure as Code."

Automating the deployment of "Infrastructure as Code" allows us to:

- Deploy identical infrastructure across all environments including development, QA, testing, and production setups. It also allows us to create production-like setups for testing the code during development and testing phases.
- Perform frequent testing of the code as the deployment of infrastructure is no longer a time-consuming, resource-intensive process. We can deploy and configure the infrastructure at the push of a button and perform our tests instantaneously. This allows us to fail early and fail fast in case of any discrepancies in the code and fix it without wasting more time and resources on the faulty code.
- Eliminate delays associated with setting up and configuring infrastructure. By automating infrastructure deployment, our developers do not have to wait (or depend) on the operations team to provision the infrastructure before they can test the code.

Automating architecture deployment in AWS helps simplify the process of auditing and accrediting deployed applications. Having a base configuration for components such as IAM and VPC controls ensures that workload owners are deploying architectures based on compliance standards.

Audit and Trace

Audit and Trace security principles identify the requirements to understand on an ongoing basis if the mechanisms are in place to continually protect data and the assets storing or processing that data. Audit and Trace apply both to system artifacts (configuration) as well as data. Audit and Trace allows Information Security to validate from the current time going backwards, what has been accessed and by whom, and when, and to compare these against the expected. Audit and Trace are also crucial for forensics should a security incident occur. Securing our assets is knowing what they are. We should not have to guess what our IT inventory consists of, who is accessing our resources, and what kind of actions did someone execute on our resources. Audit and Trace are tools for keeping track of and monitoring our AWS cloud resources, so we have instant visibility into our inventory as well as our user and application activity. For example, with Audit and Trace we can discover existing AWS resources, export a complete inventory of our AWS resources with all configuration details, and determine how a resource was configured at any point in time. These capabilities enable compliance auditing, security analysis, resource change tracking, and troubleshooting.

Logging and Monitoring

Logging and monitoring are key components in security and operational best practices as well as requirements for industry and regulatory compliance.

There are two aspects to consider when discussing logging:

- Collecting logs for all our resource-related activities
- Securing the collected logs

Understanding the changes made to our resources is a critical component of IT governance and security. It is equally important to prevent changes and unauthorized access to the log data.

Several regulatory requirements require us to maintain the integrity of the log management system in addition to collecting logs.

AWS CloudTrail is a web service that records AWS API calls made on our account and delivers log files to an Amazon S3 bucket that we specify. It records important

information about each API call, including the name of the API, the identity of the caller, the time of the API call, the request parameters, and the response elements returned by the AWS service. This information helps us to track changes made to our AWS resources and to troubleshoot operational issues. Near-real-time alerts to misconfigurations of logs detailing API calls or resource changes is critically important for effective IT governance and adherence to internal and external compliance requirements. Even from an operational perspective, it is imperative that logging is configured properly to give us the ability to oversee the activities of our users and resources.

Configuration Management

We use Puppet's configuration management solution Puppet makes it easy to track our resource's configuration. We can view continuously updated details of all configuration attributes associated with AWS and application specific resources and we are notified of every configuration change.

Puppet gives us access to resource configuration history. You can relate configuration changes with logging system events that possibly contributed to the change in configuration. This information provides you full visibility right into details such as "Who made the change?" and "From what IP address?" to the effect of this change on AWS and related resources. We can use this information to generate reports to aid auditing and assessing compliance over a period of time.

Threat Protection

We have a multi-layer strategy for threat protection.

Traditional Network Gateways and Firewalls

We create and attach an internet gateway (IGW), virtual private gateway (VGW), or both to establish external connectivity.

Using an IGW, we connect instances in our VPC to the internet. An IGW is a horizontally scaled, redundant, and highly available VPC component that allows communication between instances in our VPC and the Internet. It therefore imposes no availability risks or bandwidth constraints on our network traffic. An IGW serves two purposes: to provide a target in our VPC route tables for internet-routable traffic and to perform network address translation (NAT) for instances that have been assigned public IP addresses.

Once traffic is allowed through the IGW and NAT, it goes to an Elastic Load Balancer (ELB) and then to the EC2s as needed. On each EC2, we use AWS EC2-based firewalls in the form of security groups. A security group acts as a virtual firewall that controls the traffic for one or more instances. When we launch an instance, we associate one or more security groups with the instance. We add rules to each security group that allow traffic to or from its associated instances.

If a subnet's traffic is routed to an Internet gateway, the subnet is known as a public subnet. However, the Internet gateway route is not sufficient to provide Internet access to instances in the subnet. To enable an instance in your public subnet to communicate with the internet, it must have a public IP address or an Elastic IP address that's associated with a private IP address on your instance. Our instance is only aware of the private (internal) IP address space defined within the VPC and subnet. The internet gateway logically

provides the one-to-one NAT on behalf of our instance, so that when traffic leaves our VPC subnet and goes to the internet, the reply address field is set to the public IP address or Elastic IP address of our instance and not its private IP address. Conversely, traffic that's destined for public IP address or Elastic IP address of our instance has its destination address translated into the instance's private IP address before the traffic is delivered to the VPC. If a subnet doesn't have a route to the internet gateway, the subnet is known as a private subnet.

We use a public subnet for resources that must be connected to the internet and a private subnet for resources that won't be connected to the internet.

To summarize:

1. Public subnets have a routing table entry to an Internet gateway (IGW) for potential inbound/outbound access to the Internet
2. Private subnets have a routing table entry to a proxy/NAT instance for potential outbound-only Internet access
3. Protected subnets should have zero direct or indirect access to the Internet and are commonly used for regulated workloads (e.g., in-scope data)

Intrusion Detection/Prevention

Our primary Intrusion Detection/Prevention (IDS/IPS) protection is implemented by the extremely popular Open Source IPS solution Snort. Snorts sit in-line with our application traffic.

Once the traffic has been inspected and filtered, the Snort forwards traffic to the internal, backend load balancer which then distributes traffic across our application EC2 instances. This configuration allows the instances to scale and meet capacity demands without affecting the availability of our IPS Protection.

Incident Handling

Developing an incident-response strategy for our cloud-based environment is different from developing one for traditional, on-premise environments. When using the AWS cloud, we have a wealth of information in the form of logs and metrics that can be very helpful when responding to a security incident.

We consider ahead of time what information is provided to us by AWS and how it can be leveraged when responding to an incident. We can also use AWS partner solutions to analyze this data to effectively anticipate, deter, detect, respond, and recover during security incidents.

If some of our virtual machines get compromised in the AWS cloud, we could simply modify the security group attached to that instance to isolate the problem. We can take this one step further by creating a new subnet. Once we have isolated the issue, we can further investigate to identify the threat source, vulnerabilities in our configuration, and potential risks.

An incident response strategy typically includes the following phases: anticipate, deter, detect, respond, and recover.

In order to anticipate and deter security incidents, we follow common security best practices. These include the AWS security best practices as taught by AWS in their security designs, operational best practices as outlined by our own IT organization, and regulatory practices as defined by governance and compliance standards.

For efficient detection of security incidents, we first need to have a good understanding of normal patterns in our environment. We can analyze logs to identify these normal patterns and also detect anomalies. AWS CloudTrail, Amazon CloudWatch, and AWS Config are some of our best tools for this phase. A comprehensive logging methodology helps us in every phase of our Incident Response strategy. This includes access logs, application logs, system logs, event logs, API call logs, and database logs.

We leverage various AWS tools and features to automatically collect and store logs from different components of our AWS resources. For example, AWS CloudTrail will capture all the API call logs while Amazon CloudWatch is used to capture application and system logs.

We have identified the critical logs for our IR management and ensured they are captured frequently and stored securely. For any of our sensitive data in S3 buckets, we turn on S3 bucket logs and include those logs in our overall log captures. S3 bucket logs will log all operations on all objects within that bucket. While AWS provides a wealth of logging information about the infrastructure, we use host-based logging to track the activity taking place inside our instances. For host-based logging, it is important to get the logs off of the host as soon as possible: we off-load the logs from the host on a daily basis.

Knowing our traffic under normal circumstances helps us easily identify abnormal patterns. We understand and benchmark our expected usage levels and use this data to identify abnormal levels or patterns. We are always looking for attackers who are either probing or testing our applications.

We are prepared to restore resources quickly. We keep our automation templates in sync with our infrastructure by avoiding manual patching. If we need to change a configuration, we use configuration management tools such as Chef. We are prepared to replace an instance on a short notice, and even deploy a new stack, if necessary.

We have a well-practiced plan of tactics and playbooks that are regularly rehearsed so that our Incident Response team can execute the plan rather than try to improvise under the stress of an attack.

Business Continuity / Disaster Recovery

Business Continuity (BC) ensures an organization's critical business functions continue to operate or recover quickly in case of an incident. It may also be referred to as High Availability (HA).

Disaster Recovery (DR) enables the recovery or continuation of vital technology infrastructure and systems following a natural or human-induced disaster. It may also be referred to as backup and recovery.

We use scaling as a defense tactic for security incident response management. In a similar fashion, scaling can also be used for high availability. We leverage scaling at different tiers in our AWS cloud architecture to ensure continuous availability of all our AWS resources and ultimately our service, application, and data. For example, we use auto-scaling and elastic load balancing to automatically scale our instances and distribute the load across instances. Additionally, we also use inherent features like standby replicas, read replicas, versioning, and snapshots to ensure high availability of our data.

Availability zones are distinct and isolated locations that are designed to be insulated from failures in other AZs. They are great for fault isolation. We launch instances in more than one AZ to prevent loss of service in the event of a failure that affects an entire AZ.

High availability on AWS for the relational database service, RDS, is accomplished by simply turning on a flag called multi-AZ which will create a standby replica of the database in another AZ. And, in case the master fails, the standby replica is automatically promoted as the master.

We use the following best practices in our DR strategy:

1. We make sure all critical systems, applications, and data are replicated.
2. We automate the deployment of our infrastructure in AWS using automation systems like Puppet and Ansible.
3. We ensure the retention policies on the data are set according to our organizational and regulatory requirements.
4. We test our DR environment regularly to ensure that services, applications, and data can be restored successfully within the defined SLAs.

Additional areas of consideration in our DR plan include:

- Our IT and business teams have developed a specific process to identify the organization's critical assets on AWS as part of the DR plan.
- We validate the completeness of our Amazon Machine Image (AMI) library. We validate that each AMI is complete and up-to-date with all relevant software updates, configuration changes and patches.
- We identify if automated provisioning of AWS resources is utilized. If not, we identify the process in place to regularly update and maintain these AMI's and ensure it is in alignment with our organization's patch management and configuration management policies.
- We validate our organization's required changes to the Domain Name System (DNS) at the time of a disaster and validate this process is documented in the DR plan
- We review Single Point of Failure (SPOF) analysis with our IT and business team.
- We monitor the cadence and types of DR testing executed across our AWS infrastructure.

Conclusions

This paper has covered in extensive detail ***Darwinbox's*** commitment to compliance as well as our technical approach to security.

ABOUT US

Darwinbox is India's fastest growing cloud based HCM platform that is enabling enterprises to achieve strategic HR goals faster and smarter. Over 350+ leading enterprises with 850k users manage their entire employee lifecycle on our unified platform.

One of the most preferred HCM solutions by enterprises in APAC
– Gartner

Arvind Fashioning Possibilities · kotak · adani wilmar · Myntra · NIVEA · Dr.Reddy's · ZILINGO · tokopedia · ZALORA · paytm · SWIGGY · HT Media

Know more about how 350+ Global Organisations enabled a #SmarterWorklife with Darwinbox

Explore Darwinbox