# dotCMS Security Guide

## Security for dotCMS Enterprise & dotCMS Cloud

dotcms

# Contents

This document describes how dotCMS Enterprise and dotCMS Cloud deliver a secure hosted environment for digital experience platforms.

# What is dotCMS Cloud?

dotCMS Cloud is the dotCMS Platform-as-a-Service offering for web content management. Along with access to the dotCMS Enterprise Edition, dotCMS Cloud offers on-going support and maintenance as part of the subscription. Deployed in a three tiered web architecture, dotCMS Cloud is functionally separated into:

- The delivery tier - which has the dotCMS delivery tier application
- The repository - where all content, metadata, user and workfow data are stored
- The authoring (content management) tier - CMS web application to edit and publish

# How does dotCMS Cloud Handle Security?

dotCMS offers out-of-the-box authentication and authorization services, but also uses an extensible security mechanism allowing for very flexible and secure integrations with external authentication and authorization services.

The sections below cover the security capabilities of the dotCMS Enterprise Edition as well the dotCMS Cloud platform starting with data center and network security and moving on to application, user and data security before touching on security procedures and audits.

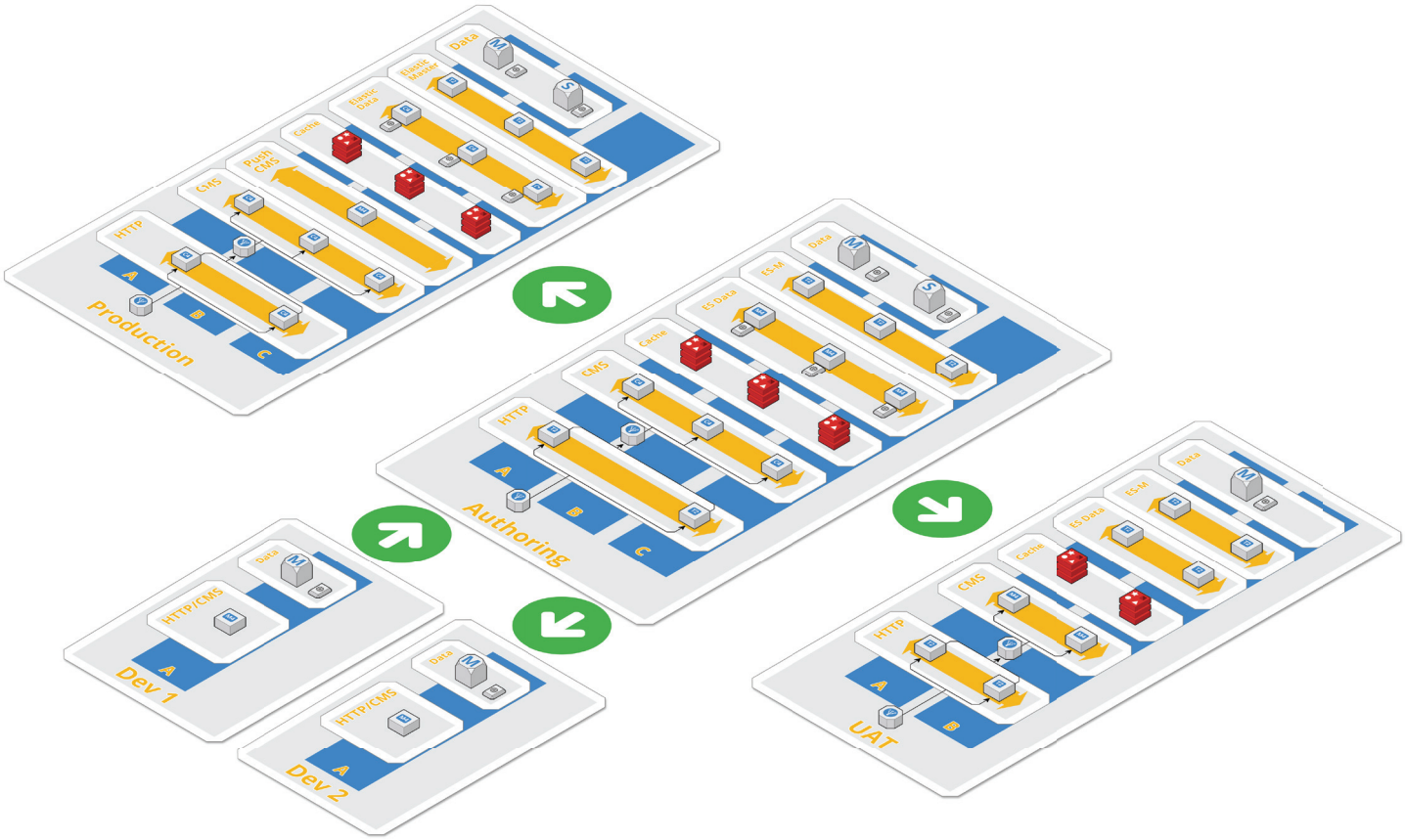
# Physical Security

## Hosting and Infrastructure

For the delivery of dotCMS Cloud, dotCMS works with Amazon Web Services (AWS), which is tier-5 global cloud infrastructure provider that meets the highest standards in availability and security.

All data centers have redundant internet connectivity. Along with a clustered production environment, dotCMS Cloud offers a Test environment and a Staging/Acceptance environment as part of its standard offering giving customers full control for continuous development (Test) and integration tests (Acceptance) before deploying to a Production environment.

The dotCMS Cloud environment is set up based on dotCMS's best-practices with regards to performance and security. Each environment is made up of multiple layers: Load balance layer, Web proxy layer, Application layer, and Database layer. Each virtual machine in each layer has its own host based firewall rules. And, because a typical environment contains multiple instances (nodes) of the site application server and the CMS application server, it ensures delivering high performance and availability.



# Access to Data Centers

dotCMS is hosted on an AWS infrastructure and the perimeter layer[1] is enforced with regards to access to their data centers.

## ACCESS IS SCRUTINIZED

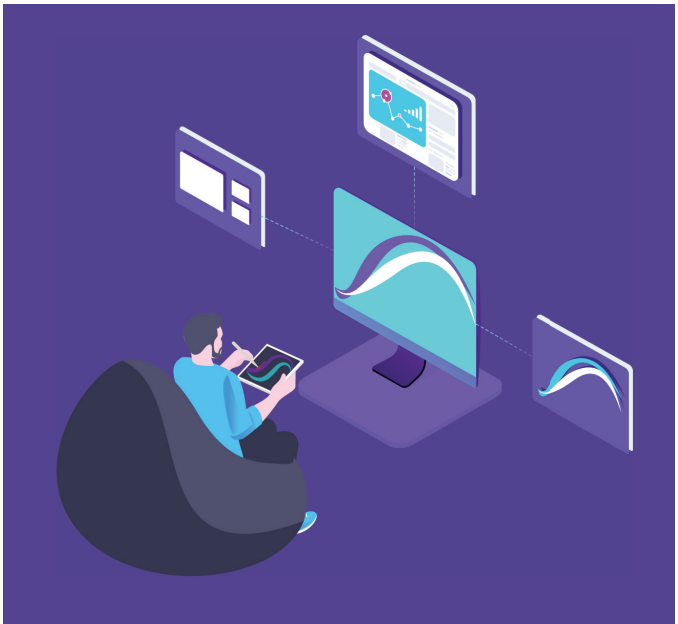
AWS restricts physical access to people who need to be at a location for a justified business reason. Employees and vendors who have a need to be present at a data center must first apply for access and provide a valid business justification. The request is reviewed by specially designated personnel, including an area access manager. If access is granted, it is revoked once necessary work is completed.

## ENTRY IS CONTROLLED AND MONITORED

Entering the Perimeter Layer is a controlled process. AWS staffs entry gates with security officers and employs supervisors who monitor officers and visitors via security cameras. When approved individuals are on site, they are given a badge that requires multi-factor authentication and limits access to pre-approved areas.

## AWS DATA CENTER WORKERS ARE SCRUTINIZED, TOO

AWS employees who routinely need access to a data center are given permissions to relevant areas of the facility based on job function. But their access is regularly scrutinized, too. Staff lists are routinely reviewed by an area access manager to ensure each employee's authorization is still necessary. If an employee doesn't have an ongoing business need to be at a data center, they have to go through the visitor process.



AWS staffs entry gates with security officers and employs supervisors who monitor officers and visitors via security cameras.

## MONITORING FOR UNAUTHORIZED ENTRY

We are continuously watching for unauthorized entry on our property, using video surveillance, intrusion detection, and access log monitoring systems. Entrances are secured with devices that sound alarms if a door is forced or held open.

## AWS SECURITY OPERATIONS CENTERS MONITORS GLOBAL SECURITY

AWS Security Operations Centers are located around the world and are responsible for monitoring, triaging, and executing security programs for our data centers. They oversee physical access management and intrusion detection response while also providing global, 24/7 support to the on-site data center security teams. In short, they support our security with continuous monitoring activities such as tracking access activities, revoking access permissions, and being available to respond to and analyze a potential security incident.

[1] https://aws.amazon.com/compliance/data-center/perimeter-layer/

# Network Security

dotCMS insulates the dotCMS Cloud platform from inappropriate or malicious internet traffic. To accomplish this, dotCMS employs multiple network defenses, from firewalls and network intrusion detection to 24/7/365 network surveillance and incident response.

# Software Security

The dotCMS software runs on a secure enterprise stack[2] of operating systems, application servers, and database servers. Multiple server pairs (CMS units) make up the dotCMS Cloud platform. Each customer is granted exclusive access to their own content management environment and database instance. A combination of Web, database, and application security methods and practices insulate customers both from each other and from external attack.

## WEB-BROWSER SECURITY

To access the CMS Web interface, the customer's browser must have JavaScript and session cookies enabled. Cookies used by the CMS application do not contain any user credentials or session data. In other words, dotCMS does not store any sensitive information on the user's system.

## DATABASE SECURITY

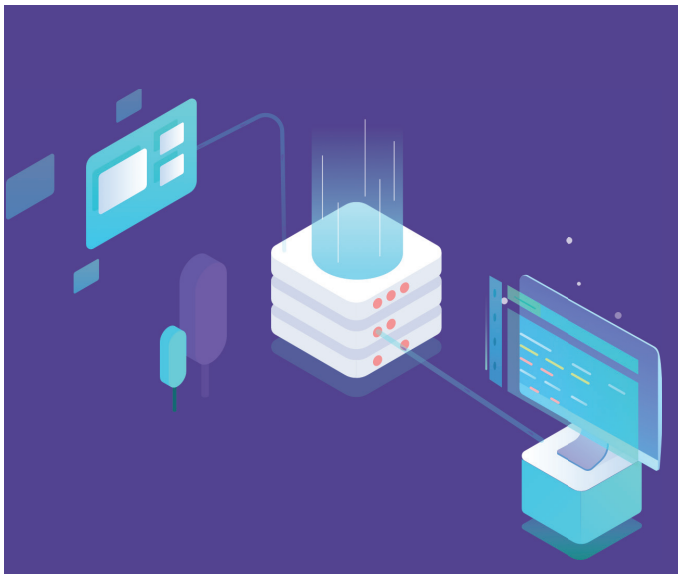
Each customer is given their own separate database instance on the PostgreSQL database cluster. Access to that database instance is protected by an auto-generated strong password, unique to each customer. In addition, each database can only be accessed from the CMS Web server to which that customer has been assigned. Together, these database access controls protect the privacy and integrity of each customer's managed content.

## APPLICATION SECURITY

During the application development on dotCMS, content management security guidelines are used to avoid introducing application vulnerabilities that might otherwise be exploited to attack the dotCMS Cloud platform or gain unauthorized access. The dotCMS content management architecture and its underlying frameworks prevent SQL injection attacks by default.

## CROSS-SITE REQUEST FORGERY PROTECTION

For added security against CSRF type attacks, the dotCMS CSRF filter plugin[3] can be deployed as a strong preventative measure against "Cross-site Request Forgery". The plugin forces validation of the browser header "referer" and validates the referring host against the list of hosts being served in dotCMS. A configuration property can be used to add additional hosts to the list or additional aliases can be added to each host while using the dotCMS backend site editing tools. This filter will only run via OSGi[4] in dotCMS running under the Tomcat servlet container. If you are running dotCMS in another app server, you will need to copy the logic of this plugin and provide it as a "static"[5] plugin.



Each customer has a separate database instance for enhanced security.

[2] https://dotcms.com/docs/latest/dotcms-technology-requirements

[3] https://dotcms.com/docs/latest/csrf-filter-plugin

[4] https://dotcms.com/docs/latest/osgi-plugins

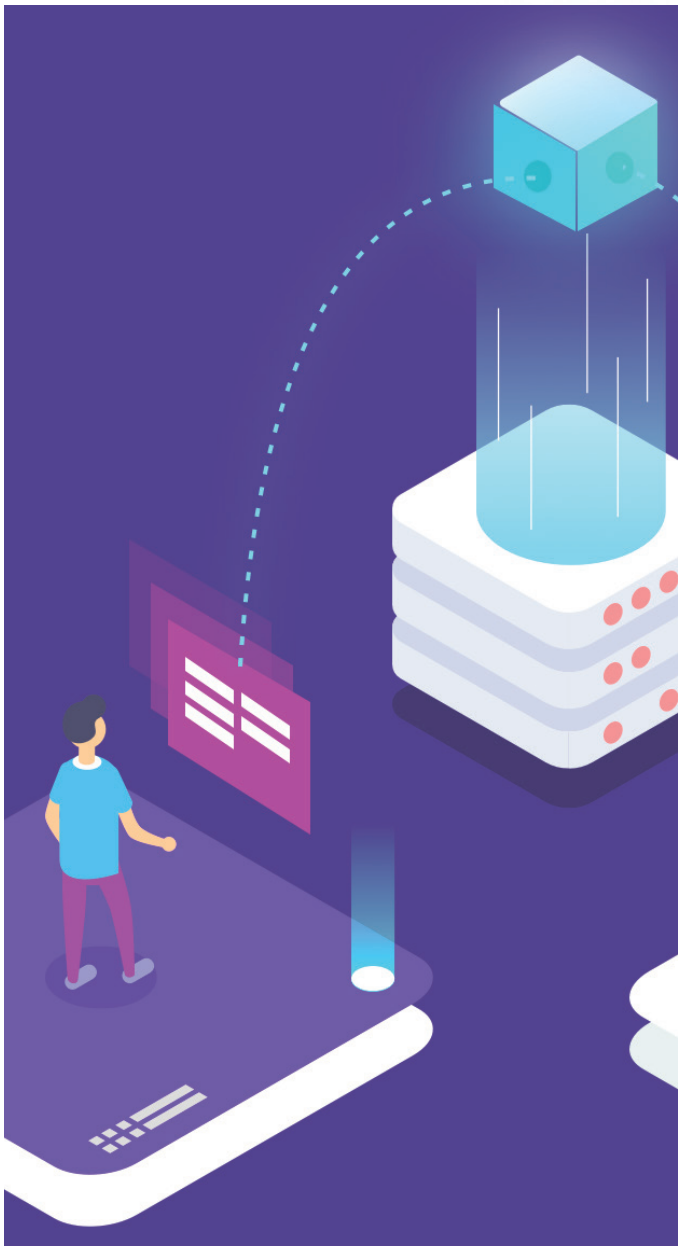
[5] https://dotcms.com/docs/latest/static-plugins

# Application Access and User Permissions

dotCMS has an extensive security model that limit access on repository level. By default, applications use a single (password) authentication and authorization mechanism. If required, multi-factor authentication can be added by configuration (not customization). Also, the complexity of the password can be configured and tailored to customer specific needs. Passwords can expire on a configurable interval and this policy holds for all users.

## User Management

User management is a very important aspect of any type of enterprise software. For content management systems, this aspect needs to be split into user and permission management for site visitors and for CMS users. Combined, they control which content a site visitor can see, or which content a CMS user can see or edit. dotCMS uses by default an internally developed authentication and authorization solution but also provides the option to integrate with LDAP and SAML servers. Certified options for LDAP and SAML include: Microsoft Active Directory, Oracle LDAP, and OpenLDAP.



## Permissions

dotCMS offers a granular permission module[6]. Key distinctions with regards to permissions are on roles and objects. It allows enterprises to set-up very granular authorization model that fits basically any need.

To simplify the use of permissions, dotCMS allows you to implement permission inheritance[7]. Child objects may be configured to automatically inherit the permissions of their parent objects, so any new content created in a particular folder automatically receives appropriate permissions.

Using permission inheritance, you can configure your site to automatically assign appropriate permissions to new content. By avoiding the need to permission each object individually, you can allow your content

[6] https://dotcms.com/docs/latest/permissions

[7] https://dotcms.com/docs/latest/permission-inheritance

contributors to create content without being concerned about (or aware of) permissions.

## AUTHENTICATION: PASSWORD MANAGEMENT AND ACCESS CONTROLS

By default, dotCMS only uses password authentication but a multi-factor authentication can be configured as well. dotCMS also supports IP based access control lists (ACL) so that only people coming from a customer specified set of IP addresses can access the CMS environment. Password complexity can be configured and tailored to match customer specific security policies. Empty passwords are not allowed and password expiration is enabled by default with configurable time frames to match customer specific security policies. A forgotten password feature can be configured in the client-specific project. Admin users can reset a user's password but passwords can also be reset through the dotCMS Support Desk. After a configurable number of attempts, a CAPTCHA must be filled in to prevent brute forcing passwords.

## AUTHORIZATION: ROLES, GROUPS, AND PERMISSIONS

dotCMS allows for very fine authorization controls[8], which are fully configurable. By default the roles author, editor, and admin are defined. dotCMS uses Context Aware Role Based Access Control (CA-RBAC). Roles can be assigned to only parts of the system (features and content). Typically, the users of the CMS are split into groups, where each group has their own set of access rights. These groups, as well as the actual users and their login credentials are stored in the repository. Next to storing this information in the repository, it is also possible to perform authentication against external systems. This allows for instance the reuse of an external LDAP or Active Directory system to authenticate users, removing the need to create and maintain a copy of all user information in the CMS.

Actions taken to each CMS deployment are limited by network and system access controls as needed by the customer administrator (for user accounts) or dotCMS (for administrator accounts). Any CMS session that deviates from the previous 30 day parole for that user in at least three ways results will trigger a security alert. An email message is also sent to the customer administrator to warn of potential user account compromise. User permissions are further needed and enforced at three points:

- Each user account is associated with defined Access Control Lists.
- Each user account can also be granted specific CMS File/Folder permissions.
- Each user account must be assigned one or more CMS Work ow permissions that determine whether that user can create, edit, approve, or publish CMS-managed content.

## SECURITY EXTENSIONS / INTEGRATIONS

dotCMS has been developed to align with Enterprise security policies. In addition to the out-of-the-box authentication and authorization solutions, dotCMS also fully integrates with SAML and LDAP servers including Microsoft Active Directory, Open LDAP, Oracle LDAP, and other LDAP compliant directory services[9]. This also allows for the use of single-sign-on mechanisms. Integration with other identity management systems or single-sign on mechanism is available via dotCMS's open and extensible system. In case SSO solution is preferred, HTTP(s) or another reverse proxy is configured and used to redirect browser clients to a central Enterprise SSO server for authentication. After authentication, the user and his valid security token are then redirected back. Alternatively, the CMS and Site application can authenticate users using Form Authentication, JAAS or String Security Integration, or using a custom implementation. dotCMS comes with a standard set of security providers to connect to several types of external systems, but also allows exibility to create custom security providers.

[8] https://dotcms.com/docs/latest/role-permissions

[9] https://dotcms.com/docs/latest/ldap-configuration

# Data Security and Backups

The most important part of a dotCMS application is the database (repository). This repository contains the settings of the application, as well as the content that is shown and managed by the CMS. Keeping this data safe and secure is key.

## DATA OWNERSHIP

All content, configuration, and targeting data belongs to the customer and can be entered through the dotCMS interface. This includes, click-path, and web-visitor information for the Personalization / Content Targeting module which is stored in a separate NoSQL database (ElasticSearch).

## APPLICATION DATA ACCESS

This diagram shows the different layers of a dotCMS application. Applications are built using dotCMS's tested and secure application delivery framework. This framework enforces a security session and is always present, making it possible to restrict access up to field level on content objects. dotCMS is functionally separated into the authoring tier, the repository, and the delivery tier, but also logically separated into load balance layer, web proxy layer, application layer, and a database layer. Each virtual machine in each layer has its own host based firewall rules. Data lives less than seconds in the web layer as it's only passed through by the proxies, unless (memory or disk) caching is enabled in the proxy layer.

## ENCRYPTION, SSL & CERTIFICATES

dotCMS will store important and sensitive data (such as user passwords) in an encrypted format using the (Java) SHA-1 hash/algorithm with salting (size 8). For the dotCMS Cloud platform, traffic from the web server to the client is encrypted over an https connection. The data between the primary and secondary data center is transported over a private line. dotCMS supports SSL. For the dotCMS Cloud service, certificates (encryption keys) are provided by the clients and installed on the dotCMS servers. HTTPS is used by all processes that require secure communication (like password validation).

## DATA INTEGRITY & BACKUPS

dotCMS makes full backups of all customer data on a daily basis. Since dotCMS/repository does not store all information in the database, backing up assets is also a critical step. The backups are transported to a second data center at a different location over a dedicated private line. From the backups the originals systems can be restored. dotCMS is located in two data centers per region (North-America and EMEA) and backups are copied from the primary data center to the secondary and vice versa. In addition, dotCMS can transfer a copy of the backup over a secure connection to a customer's server at an additional premium. dotCMS doesn't have backups in the primary location and we have access 24/7 to the backups in the secondary location. The backup and retention policies for dotCMS Cloud are as follows:

**Production environments:**

- A full backup is made every night between 1 am and 8 am Central European Time
- For the last seven days all backups are kept
- For the last month one backup per week is kept
- For the last twelve months one backup per month is kept

**Test and Acceptance (Staging) environments:**

- A full backup is made every night between 1 am and 8 am Central European Time
- Backups are kept for at least three days

It is very common to restore a Production backup in a Development or Testing environment for testing purposes during a project / new release. The dotCMS infrastructure team that manages the dotCMS Cloud platform tests the backup and restore procedures regularly.

[6] https://dotcms.com/docs/latest/permissions

[7] https://dotcms.com/docs/latest/permission-inheritance

# Compliance and Security Policies

## Standards & Security Audits

The AWS data centers and products dotCMS is contracting for the delivery of dotCMS Cloud are certified for all major commercial and government certification standard[10]. dotCMS and dotCMS clients regularly request external agencies to conduct security audits for dotCMS and dotCMS Cloud. These security audits ensure that the CMS authoring environment and Delivery Tier comply with the latest security standards to protect dotCMS implementations against attacks. To date, all projects comply with these security audits. As a company, dotCMS is responsible for ensuring the dotCMS Platform is aligned with the latest best practices in security. Additionally, dotCMS has built up and documented a series of best-practices to help prevent vulnerabilities such as cross site in delivery channels.

## Logging

All activity in the dotCMS authoring and run-time environment is registered and available for reporting. The audit log contains among others, logins, workflow actions, and any modifications to the system. System admins can access the log files through the authoring UI and inspect / export the log files for further auditing. The log files can also be accessed through the REST API in real-time and feed into Third Party Application Monitoring platforms.

## Monitoring

Security-related events are routinely monitored and logged by dotCMS's firewalls and servers. A monitoring daemon on each server also keeps an eye on operational events, including host resources and environmental factors. All alerts are relayed to dotCMS's Network Operations Center (NOC). In addition, priority 1 alerts are immediately escalated by paging dotCMS NOC staff. At the dotCMS NOC, trained network and system administrators monitor incoming alerts 24/7/365, verifying each new alert before initiating the appropriate response.

To investigate alerts, dotCMS NOC staff uses strongly authenticated, encrypted administrative interfaces to remotely query dotCMS On-Demand platform components. Specifically, all terminal server sessions are protected by Secure Shell (SSH) or Virtual Private Network (VPN) tunnels:

- For SSH administrative access, dotCMS requires SSH version 2, RSA digital certificate authentication. Password authentication over SSH is not allowed.

- For VPN administrative access, dotCMS requires an SSL VPN tunnel, protected with personal certificates, 160-bit HMAC-SHA1 for message integrity, and 128-bit Blow sh encryption.

These secure interfaces let dotCMS investigate alerts remotely, while preventing unauthorized access to the dotCMS Cloud platform or disclosure, modi cation, or replay of sensitive management messages.

Our clients typically have a Third Party Application Monitoring and/or Application Performance Management solution in place to monitor all their business applications to have end-to-end visibility on their entire digital experience platform. Solutions like AppDynamics, New Relic, and Dynatrace are well know solutions in this regard.


[10] https://aws.amazon.com/compliance/pci-data-privacy-protection-hipaa-soc-fedramp-faqs/

## Vulnerability Testing

dotCMS Cloud platform undergoes vulnerability assessments and penetration tests at regular intervals. In addition, clients of dotCMS conduct load and penetration tests periodically. Some dotCMS clients in government and cyber-security go even further by inspecting every single line of code on an annual basis. All security vulnerabilities are shared and resolved immediately in the core software if needed.

dotCMS's clients (particularly in financial services or government ) engage third parties to conduct penetration tests on the dotCMS Cloud platform. If non-compliances are found in either the core software of dotCMS or the Cloud platform, they are resolved with the highest priority.

### VULNERABILITY REMEDIATION / PATCH MANAGEMENT

To help eliminate vulnerabilities before they can possibly be exploited, dotCMS combines proactive patch management with periodic internal penetration tests.

- dotCMS monitors security lists for new exposures that may impact dotCMS Cloud
- As new security patches become available, they are first reviewed for relevance to dotCMS Cloud Platform.
- Relevant security patches are first verified on QA/ Staging servers, typically for two days before being applied to production servers.
- Routine vulnerability scans are also performed by dotCMS semi-annually.

### SECURITY INCIDENT MANAGEMENT

dotCMS has a dedicated and specific process around security issues and some issues are dealt with higher priority than other issues. During incident investigation, if NOC sta determines that an attack is underway or has occurred, actions will be taken to quarantine IP addresses and/or disconnect sessions as needed to contain the incident and prevent future damage. If necessary to mitigate the attack or protect customer content, staff may also temporarily disable CMS customer accounts and/or databases.The dotCMS Service Manager assigned to each affected customer account will contact the customer to review the incident, actions taken, and impact on that customer.

### SECURITY & PRIVACY POLICIES

dotCMS has implemented a set of corporate policies to take maximum security measure for our clients and our company. These policies are reviewed periodically (at a minimum once per year) as part of our business continuity process. dotCMS currently has the following security & privacy policies implemented:

- Security Policy
- Password Policy
- Privacy Policy
- Cookie Policy
- GDPR Policy

All policies are documented and available upon request.

### BUSINESS CONTINUITY AND DISASTER RECOVERY

Aside from backup and security protocols, dotCMS has an extensive business continuity and disaster recovery plan. For details, please refer to the Business Continuity Plan which can be provided as a separate document upon request.

# Security Best Practices

Regardless if your organization deploys dotCMS on your cloud / hosting environment or you build your applications on dotCMS Cloud, our engineering team has developed a number of best-practices that you to build and manage a secure platform with dotCMS. The best practices laid out in this document are all implemented for dotCMS Cloud customers. For on-premise deployment of dotCMS it is up to the customer's implementation team to follow these best-practices.

## Recommended Approaches

Although specific configurations and practices are necessary to implement strong security, it is often just as important to adopt an overall approach to security that ensures that, when resources are scarce or there are conflicting needs, appropriate choices can be made about which specific security practices to implement. The following are some approaches[11] we recommend to help you identify, implement, and manage your security practices for your dotCMS site:

- Continual improvement
- Layered security
- Whitelist access

## Server Configuration

The applications you run on your server, including dotCMS, can only be as secure as your server itself. If your server security is compromised, efforts to secure dotCMS can not protect your server and site from being compromised. Therefore it is vital that you ensure you configure your server for maximum security.The following are several steps you can use to increase the security of your server[12]:

- Run dotCMS on a dedicated server
- Disable all unused ports
- Disable or uninstall all unused services and applications
- Limit access points with a firewall
- Use and maintain anti-virus software on endpoints
- Restrict server permissions
- Remove all sensitive files from the ROOT folder

## Database Security

The database-driven applications, including dotCMS, are as secure as the database configuration and server themselves. If the database security is compromised, just like the server, efforts to secure dotCMS can not protect the database (server) and the applications that are supported. The following steps[13] are designed to increase the database security in the dotCMS application landscape:

- Limit access to database accounts, files, and folders
- Control database permissions
- Secure database configuration

[11] https://dotcms.com/docs/latest/security-best-practices#Approaches

[12] https://dotcms.com/docs/latest/security-best-practices#ServerConfiguration

[13] https://dotcms.com/docs/latest/security-best-practices#Database

## Latest Version & Patches

Operating System and software vendors regularly release updates which include security enhancements and fixes for potential security issues. Therefore it is crucial that you keep all your software up-to-date to ensure you have protection from known security vulnerabilities. This includes, but is not limited to, all of the following:

- dotCMS[14]
- Third Party Software:
  - Operating System
  - Application Server
  - Database
  - Java Virtual Machine[15]
- Web-browsers

## dotCMS Configuration for Maximum Security

The following topics outline some of the most common areas of dotCMS configuration that should be considered when implementing security for your dotCMS site. Not all of these configurations[16] will make sense for all sites; however you should consider each of these and understand the implications and potential vulnerabilities if you choose not to implement them:

- Increase login Security
- Configure SSL
- Require HTTPS Access
- Implement Secure Push Publishing
- Disable features with potential security risks

## Access to Static Endpoints

Customers using the enterprise feature, Push Publishing, should follow these guidelines.

**Configure your Push Publishing Static Endpoints[17] to restrict access to all of the following:**

- A specific AWS user account:
  - Create a separate AWS user account specifically for dotCMS.
  - Limit the user account to the minimum rights needed for dotCMS to publish static content (and, if necessary, create new buckets matching the AWS S3 bucket variables).
- A specific bucket or set of buckets:
  - Use a combination of prefixes and wildcards in the AWS S3 Access Control List (ACL) to limit the buckets the AWS user account has rights to.
- A specific IP address or range of IP addresses:
  - The AWS S3 ACL allows you to specify that content can only be written to a bucket from a specific source IP address or range of source IPs or specific HTTP referer[18]. Ensure that you restrict access so that only the IP address of your dotCMS authoring/UAT server can write content to your AWS S3 bucket(s).

[14] https://dotcms.com/docs/latest/security-best-practices#DotCMSVersion

[15] https://dotcms.com/docs/latest/security-best-practices#JVMSites

[16] https://dotcms.com/docs/latest/security-best-practices#DotCMSConfiguration

[17] https://dotcms.com/docs/latest/connecting-remote-servers#EndPointServers

[18] http://docs.aws.amazon.com/AmazonS3/latest/dev/example-bucket-policies.html#example-bucket-policies-use-case-4

## Secure User Management

In regards to dotCMS users, we recommend to follow the security guidelines with regards to user management[19] as laid out:

- Improve Password Security Requirements
- Deactivate Default Administrator Accounts
- Deactivate Unused User Accounts
- Limit User and Role Permissions

## Secure Coding Practices

Use secure coding practices to ensure that vulnerabilities are not introduced into your application through plugins, Velocity, or script code on your site. The following is a short list of important coding security practices. For a more complete list, please see the SANS SWAT (Securing Web Application Technologies) Checklist[20].

- Define security requirements.
- Educate developers about the security requirements.
- Conduct design reviews and code reviews.
- Don't Hardcode Credentials.
  - Never store credentials within the application code.
- Perform security testing.

**In addition, special attention should be paid to the following in all your application and plugin code:**

- Guard Against Malicious User Input[21]
- Sanitize SQL Queries
- Control the Use of Cookies[22]
- Limit the Potential for DoS/DDoS Attacks[23]

When a malicious or unauthorized user attempts to access or take malicious action against your system, they usually take actions which leave some kind of trace in the log files. Therefore it is important to review system logs as frequently as practical, both so you can recognize normal and abnormal log file messages, and detect any unauthorized attempts to access your system as quickly as possible so you can take additional precautions if necessary.

- Regularly review the Security log
- Periodically review other system log files
- Investigate unusual logging or user behavior
  - Unsuccessful login attempts
  - Unusual login locations
  - Attempts to login to individual's accounts while they're already in use

## Review Security Resources

The following is a list of some additional resources which you may find of value in implementing and evaluating the security of your dotCMS sites:

- dotCMS User Forum
- dotCMS Support (support@dotcms.com)
- Known Security Issues in dotCMS
- SANS SWAT (Securing Web Application Technologies) Checklist

[19] https://dotcms.com/docs/latest/security-best-practices#UserAccounts

[20] https://software-security.sans.org/resources/swat

[21] https://dotcms.com/docs/latest/security-best-practices#UserInput

[22] https://dotcms.com/docs/latest/security-best-practices#Cookies

[23] https://dotcms.com/docs/latest/security-best-practices#DDOS

# About dotCMS

dotCMS is a leading, open source content and customer experience management platform for companies that want innovation and performance driving their websites and other content-driven applications. Extensible and massively scalable, both small and large organizations can rapidly deliver personalized and engaging content across browsers, mobile devices, channels, second screens, and endpoints -- all from a single system.

Founded in 2003, dotCMS is a privately owned U.S. company with offices in Miami, Florida; Boston, Massachusetts and San Jose, Costa Rica. With a global network of certified development partners and an active open source community, dotCMS has generated more than a half-million downloads and thousands of implementations and integration projects worldwide. Notable dotCMS customers include: Telus, Standard & Poors, Hospital Corporation of America, Royal Bank of Canada, DirecTV, Thomson Reuters Foundation, China Mobile, Aon, and DriveTest Ontario.

**Web:** dotcms.com
**Phone:** +1-305-900-2001
**Email:** info@dotcms.com

**Miami**
3059 Grand Avenue
Miami, FL, 33133
U.S.A

**Boston**
200 Portland St
Boston, MA, 02114
U.S.A

**Heredia, Costa Rico**
Eurocenter
Primera Etapa, Piso 2
106  Heredia, Costa Rica