



enStratus Security Architecture

Design of the enStratus System and General enStratus Security Policies

enStratus enables you to live up to your existing IT policies and procedures in a cloud environment. Leveraging enStratus as your cloud broker, you take full control over the aspects of your cloud infrastructure that could result in the compromise of your data. This white paper describes the basic security architecture of the enStratus system, our policies for maintaining that infrastructure, and mechanisms for leveraging enStratus to secure your own public cloud systems.

Philosophy

We have built enStratus on the foundation of separation of roles in an IT infrastructure. Through the combination of role separation with the wide use of encryption and proper key management, you can construct a cloud-based infrastructure that will tolerate failures at multiple levels without damaging the overall integrity of your data or your ability to recover from disaster.

Separation of roles starts at the organizational level:

- You (or your integration partner) control the operation of your applications and databases
- enStratus manages the provisioning of systems, key management, and user management
- Your cloud provider manages the physical resources on which everything operates

In short, your data sits with your cloud provider. Your keys sit with us. And you manage how they work together.

enStratus helps you enforce separation of roles in two ways:

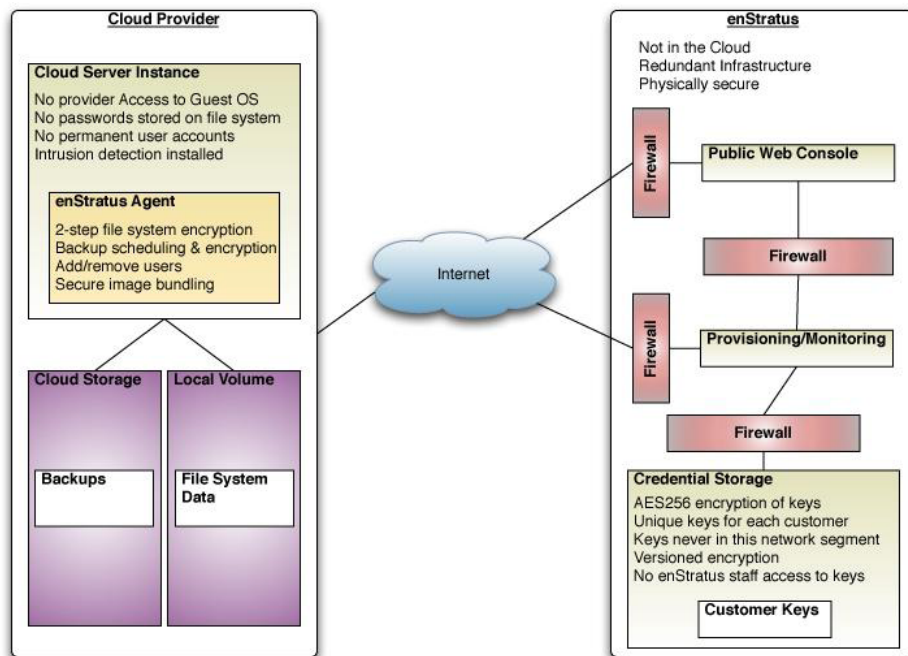
- How you set up and manage users to control the infrastructure
- How you set up clusters, so the compromise of any single virtual server does not threaten the integrity of the entire infrastructure

The Architecture

enStratus operates outside of any of the clouds we broker, without any operational elements in any public cloud. We store all customer data in our own data center—on our own servers.

All enStratus systems operate inside a state-of-the-art data center in downtown Minneapolis, running on redundant virtualized environments located on dedicated physical servers. We operate 3 VLANs, only two of which are directly accessible from the public internet: the VLAN housing our web console and web services, and the VLAN housing our provisioning system. The third VLAN, which houses customer credentials, is not addressable from the public Internet and has limited connectivity from within our own infrastructure.

All file systems, attached to all virtual machines in this infrastructure, are encrypted using SHA256 encryption.



The patent-pending security architecture under which enStratus operates.

Web Console and Web Services

All user interaction with enStratus—whether through our web application or through our web services API—occurs through the console zone.

Access to the console is currently controlled via a user name/password system. enStratus has purchased an extended validation (EV) certificate for the console to prevent phishing attacks. The entire user session is encrypted using this EV certificate. Users should verify that the site they believe to be <https://cloud.enstratus.com> is in fact our console by looking for the EV marker. Most browsers indicate the use of an EV certificate by showing the color green around the company name, in this case enStratus Networks, Inc.

We enable customers to define the security profile they want for authentication, including:

- SAML Federation
- Multi-factor authentication
- Open ID with trusted providers

If a user has access to multiple accounts, enStratus will always force them to authenticate with the strongest authentication system configured for the accounts to which they have access. enStratus implements an increasing delay in authentication upon the entry of incorrect authentication credentials. After three attempts, enStratus will lock the account out for 30 minutes.

Systems administrators may set up any number of users to manage their infrastructure and associate different users with different roles. Similarly, an administrator may set up multiple accounts and specify different roles for a given user in each account.

Any time any change is made to a user account—whether it is a change of role, password, or something else—enStratus sends an email notification to the user, alerting them to the change. Users should pay attention to these alerts and verify that the change was not a result of an attack on the system.

Within the enStratus console, sensitive data like encryption keys and passwords are “write only”. In other words, users with the appropriate permissions may create or update credentials, but there is no mechanism for them to view those credentials after saving them. The entire console zone has no access to the credentials zone, where the credentials are stored, and the provisioning zone provides no programmatic means for the interface systems, like the console and the web services API, to retrieve credentials.

User logins timeout after a certain period of inactivity.

Provisioning Systems

The user interaction systems in the console zone communicate with the provisioning systems in the provisioning zone via the web services API. There is no database access across the zones. All web services communication occurs over SSL using an SSL certificate signed by GeoTrust, VeriSign, or GlobalSign.

Each web services request uses a customer-specific authentication token. enStratus automatically generates this token when a user account is created. The provisioning system performs both authentication and access checks for each web services call using this custom token.

The provisioning systems perform the following functions:

- Monitoring of virtual servers in customer cloud infrastructures for various events
- Orchestrating the launch and configuration of systems operating in the cloud
- Processing requests by servers operating in the cloud
- Alerting appropriate individuals about key events happening in the cloud (for example, server failures)
- Tracking the encryption systems used for various resources in the cloud

Two systems talk to the provisioning systems:

- The web console and web services system
- The cloud agents (programs running on each server in the cloud)

The provisioning system does not store sensitive customer keys or authentication credentials. It instead relies on the credential system to store that data. Encryption keys for the data in the credentials system are calculated and maintained encrypted in memory in the provisioning system.

enStratus logs all events that result in a change of system state to an “insert-only” audit database. We also provide services that enable customers to log events into our audit system for their own compliance requirements.

Credentials Management

The credential management system operates inside its own network segment and is accessible only from the provisioning system. All communication between the provisioning system and the credentials system occurs over SSL web services, using an SSL certificate signed by GeoTrust, VeriSign, or GlobalSign.

All customer encryption and authentication credentials are stored in an AES256-encrypted database. No encryption keys exist in the credentials management zone. Each row in the credentials database is encrypted and decrypted on the provisioning server, using an encryption key unique to each customer account. No global key exists and no key is ever stored on any file system within our infrastructure.

enStratus does not store any data in the credentials system that identifies who the row belongs to or what purpose the encrypted data serves. We also provide versioning of our encryption systems so that we can implement new, more powerful encryption schemes in the future without the risk of making old data inaccessible.

Server Agents

Customers’ virtual servers in the cloud may operate either with or without an enStratus agent.

The enStratus agent is a Tomcat web services application that listens on port 2003. It also listens on the local host Tomcat management port of 12003 for shutdown requests. enStratus will automatically establish firewall rules in the cloud, allowing port 2003 access from the enStratus provisioning server.

On startup, the enStratus agent contacts the enStratus provisioning system over SSL to execute an authentication handshake. The agent establishes its identity with the provisioning server and provides a session-key for authentication between the two parties as well as further encrypting data transmitted from the provisioning server to the client. Communication from the enStratus agent to the provisioning system occurs exclusively over SSL, using SSL certificates signed by GeoTrust, VeriSign, or GlobalSign.

For security and logistical reasons, it is impossible to use a signed SSL certificate to encrypt communications from the enStratus provisioning server to the enStratus agent. Though we use SSL to encrypt the communications between the provisioning server and agent, those communications are at-risk of a “man-in-the-middle” attack. To eliminate the risks associated with such an attack, all sensitive data transmitted from provisioning server to agent is further encrypted using an encryption key provided by the agent to the provisioning server during the handshake phase. Because of this second layer of communication with the keys transmitted during a trusted out-of-bound session, any success in implementing a man-in-the-middle attack will result only in access to encrypted data.

The agent handles all sensitive security operations on behalf of enStratus in the cloud, including:

- System backups and backup encryption
- File system encryption
- SSL configuration
- Machine image bundling
- Shell/Remote desktop user management
- Integration with intrusion detection systems

The provisioning server may ask the agent to perform certain functions from time to time that require the use of encryption and/or authentication credentials that normally reside outside the cloud. The agent takes care of the following concerns relating to these operations:

- Ensuring the encryption of the credentials in transit in a manner closed to a man-in-the-middle attack
- Decrypting the credentials only to perform the operation in question
- Immediately wiping any memory locations in which the unencrypted credentials were stored
- If writing the unencrypted credentials to disk was absolutely necessary, securely deleting the unencrypted credentials from the file system
- Handling any errors that might otherwise accidentally leave unencrypted credentials stored on disk or retained in memory

File System Encryption

enStratus will optionally automate the encryption of file systems in the public cloud for clustered environments.

Each clustered enStratus server minimally has three volumes:

- The root file system
- An ephemeral or scratch file system
- A persistent file system

Because enStratus generally does not have access to the boot process of a cloud server, we never encrypt the root file system. Consequently, enStratus never stores any sensitive data on the root file system and recommends all customers avoid storing sensitive data on the root file system.

If a customer has configured enStratus to encrypt their cluster file systems, enStratus will first generate a random, one-time encryption key to encrypt the ephemeral file system. Due to the nature of some file system encryption tools, it is sometimes necessary to temporarily write this key out to the unencrypted root file system to create and mount the ephemeral volume. The key is subsequently securely deleted and never again used for any of the customer's systems.

Once the ephemeral store is encrypted, enStratus will send over permanent encryption keys for encrypting/decrypting the persistent storage. In this case, any need to write the key to a file system will be written to the encrypted ephemeral file system, instead of the root file system. Once the volume is mounted, the key is then securely deleted.

Backup Encryption

If a customer has configured enStratus to encrypt backups, enStratus will encrypt all backups before uploading them to the cloud storage environment or sending them over to the customer's backup cloud.

Shell/Remote Desktop Access

enStratus enables you to create machine images/server templates without having interactive user accounts pre-defined. Accounts are added and removed from the console on-demand as the operational system requires. The ability to add and remove shell access is governed by a user's role on the console.

On Unix environments, enStratus enforces key-based access to the operating system shell via SSH. No secret authentication keys are ever stored on a Unix server—not even temporarily. The enStratus console further provides a mechanism for adding users on-demand rather than having user accounts reside permanently on a server. As a result, customers can enforce a kind of multi-factor authentication for access into shell environments.

Though account access can be built on-demand (as with Unix servers), Windows servers require the temporary storage of passwords to support user authentication. Because both factors are password-based (for the console and RDP access), this is not true two-factor authentication. Nevertheless, the limited time frame in which accounts must be active on a Windows server managed by an enStratus agent make brute-force attacks against a Windows server extraordinarily difficult.

In a future release, enStratus will implement one-time, auto-expiring passwords for the Windows environment. When a user needs access to a Windows server, they will request access via the console and receive a one-time password. That account will then automatically expire after a pre-defined period of time.

enStratus Security Policies

Our objective is to support processes and procedures that will enable our customers to meet their internal security requirements, as well as support relevant standards, certifications, and regulations. This list of security policies is designed to provide a general overview of how enStratus approaches our internal security and does not represent our complete set of security policies and procedures. We are happy to work with individual customers to help address specific security concerns. Additionally, we are happy to share our security policies with customers.

Physical Security

The operational infrastructure for enStratus operates out of a hosting facility in Minneapolis built to telecommunication industry standards with redundant fiber vaults. Physical access to the data center is restricted to personnel with a business need to access the infrastructure and our servers are further locked within a cage inside the infrastructure. There is 24x7 video surveillance of all entrance/exits, lobbies, ancillary rooms, and equipment.

Access control is via two independent systems:

- A swipe card to gain access to the facility
- Proximity cards with PIN codes to gain entrance to the data center

In addition to these general security controls, the environment has fire suppression, on-site generators and other appropriate operational controls.

Application Security

We have touched on bits of the enStratus architecture built around the objective of establishing a secure trust relationship between virtual machines which may be operating in a public cloud and other systems. This is critical because a good cloud security architecture has no pre-bundled user accounts or credentials for external systems like centralized logging systems, IDS servers, or directory services. Because the images cannot be trusted to have that data at launch, enStratus has a very strong mechanism for establishing a trust relationship between the VM and enStratus and extending it to third-party/custom tools. This trust relationship enables an enStratus customer to do many things not otherwise available to administrators.

Based on this trust and the key management described above, enStratus can hand encryption keys and other sensitive data to virtual machines in the cloud for a discreet period to perform a specific function. enStratus will securely remove the specified keys from the VM once the operation is complete, with error checks to ensure removal.

enStratus integrates a plug-in architecture for interacting with intrusion detection systems on virtual machines in the cloud. The virtual machine IDS will forward all events to enStratus and enStratus will alert/forward these events to the proper targets and maintain an audit history.

enStratus can automatically encrypt block storage devices like AWS EBS and/or configure multiple volumes into a software-based RAID. The encryption credentials are maintained in the enStratus key management system. They exist on the VM only for the time period needed to set up encryption and are then removed.

As noted above, you never want pre-defined interactive user accounts established on your images/templates. enStratus enables you to manage shell/RDP access post-launch and even define access rights across an entire group of machines. In other words, a DBA will have shell access to database servers, when they need it, without manually defining that right for each server). When a user is removed from enStratus (or your LDAP/AD directory), all shell access to VMs in the cloud is also removed.

Product Security

Security is not an afterthought at enStratus: it's core to the entire architecture. Our first customer brought us a standard information security assessment that had to be met both for enStratus and the applications we were managing in their environment. That belief in and focus on the importance of security has been present ever since.

We follow three core principles:

1. There is no inherent trust in the cloud or its resources. Any trust must be earned.
2. Everything must be resources monitored, and all changes must be recorded.
3. When in doubt, treat all data as if it is sensitive data.

enStratus has a Chief Security Architect responsible for the security of the platform and our operations. All new features are reviewed by the CSA in accordance with our product security policies, and any security-related issues with the component design are first brought up at that point. Examples of items in our policies are: a requirement that all database interactions occur through prepared statements or stored procedures; sensitive data must be encrypted at rest, in transit, and in memory except when actively in use; all service endpoints must be signed by a trust authority or there must be another mechanism of establishing trust available.

At each point during the development process, we are actively looking for ways in which misfeasance or malfeasance by internal users, the cloud provider, or hostile agents might impact the system in development. And, our SaaS environment routinely undergoes a variety of compliance assessments, including penetration testing and customer reviews.

enStratus takes every reasonable precaution to ensure that its application is as secure as possible. For example, the core of the enStratus application uses end-to-end SSL for network transit, all data volumes are encrypted, the application is coded using Java to prevent buffer overflows, and all SQL queries are parameterized to prevent SQL injection attacks. Similarly, we have mechanisms in place to enable input/output validation and to leverage anti-XSS tokens. All cookies are set to both secure and http only. enStratus uses Qualys for both application and infrastructure security scans and is in the process of evaluating products such as Whitehat Sentinel and web-application firewalls to further enhance its security stance.

enStratus has well defined secure coding standards and all security relevant source code goes under source code review. enStratus is in the process of evaluating vendors for regular third party assessments.

Vulnerabilities are an unfortunate reality for every vendor and enStratus takes them extremely seriously. All potential security vulnerabilities are triaged by the CTO and Chief Security Architect and are prioritized by the severity of the vulnerability, the number of impacted customers and the likelihood of exploitation. On the basis of that prioritization, configuration changes, code fixes or other compensating controls will be applied as appropriate.

Customer Data Management

enStratus does not have any direct access to most sensitive customer data:

- We do not have access (nor programmatic mechanisms for providing ourselves access) to customer servers in the cloud, unless the customer specifically grants such access
- All credential encryption keys are calculated and stored in memory, i.e. enStratus personnel do not have mechanisms for decrypting or otherwise accessing unencrypted customer keys
- Customer credit card data for billing is stored with our PCI level I billing provider, Aria. We do not have any access to that credit card data

It is enStratus policy not to develop mechanisms through which we can access this data.

Whenever we do have access to customer data—either through the customer granting enStratus personnel access to assist in consulting or other data not part of our general operations—enStratus personnel are directed to treat that data as if it were our own. enStratus policies dictate that physical media of customer data (paper, storage devices, etc.) are never to be left in view to passersby and are to be disposed of securely in secure disposal repositories on enStratus premises.

Except where a consulting engagement absolutely requires it, electronic customer data should never leave the cloud or appear on enStratus systems. Where the consulting engagement requires it, electronic customer data is never to be placed on laptops.

All hardware is to be securely wiped before being disposed of.

Systems Access

enStratus requires redundancy in all administrative roles with strict separation of administrative roles. enStratus does not use generic administrative accounts on any systems. Instead, each user has their own access credentials with a separate set of credentials for privilege escalation.

Users are granted administrative access only as their job function dictates. All access—administrative and otherwise—is removed within 24 hours after the individual no longer requires that access to perform a job function. In the event of a non-voluntary termination, all efforts are made to remove all access concurrent with the termination.

Personnel

enStratus requires a background check of all new hires prior to their first day of work. The degree of the background check is appropriate to the job function in question.

enStratus also requires all personnel to become familiar with enStratus policies and procedures and provides regular training on the subject.

Encryption and Key Management

enStratus encrypts all file systems attached to our virtual servers and we further encrypt all backups before offloading them to our backup staging environment. The backups remain encrypted in offsite storage.

All network transactions are also encrypted using encryption appropriate to the operation.

enStratus personnel perform remote maintenance on enStratus servers via SSH using SSH keys for authentication. Employees are directed to protect their private keys with strong pass-phrases and not to share those private keys with anyone else.

No operational environment has access to any of the various private encryption keys for decrypting backups. They have access only to the public keys for the purposes of encryption. The private keys are housed outside our hosting environment on two separate systems accessible by redundant administrators. Hard copies of the encryption keys are further stored securely off-premises, unrelated to our backup offsite storage. Private keys and data are never to be in the same physical or virtual space, except in the event of executing enStratus disaster recovery procedures.

Availability and Disaster Recovery

enStratus operates across redundant hardware on VMware-based virtualized servers. The environment can tolerate the full loss of any component. We have multiple backbone connectivity and our data center is fully climate controlled with generator backup and fire suppression.

In the event of a disaster resulting in full loss of our data center, we have recovery point and recovery time objectives of 24 hours supported by daily offsite backups. In the event of any enStratus outages, customer cloud environments will continue to operate normally.

To further enhance our disaster recovery/business continuity plans, we are in the process of building out a second geographically diverse datacenter. This will enable faster recovery times, especially in the case of large scale issues beyond our local datacenter.

Compliance

Compliance is a reality for most organizations. enStratus's SaaS environment is PCI compliant. enStratus's architecture can be easily built on-premise and made compliant with whatever needs our customer has. Our services team is ready to assist on a broad range of compliance matters, such as SOX, HIPAA/HITECH and GLBA. enStratus's strong enablement of encryption, automation and governance allows customers the ability to enhance the compliance posture of the systems it is managing. For example, enStratus reports can be generated to demonstrate patch levels, access control and authentication status. For more information on how enStratus helps with compliance concern, please see the forthcoming white paper on compliance control with enStratus.

Customer Best Practices

enStratus aims to provide customers with the tools necessary to secure a public cloud infrastructure with as much automation as possible. In the end, however, responsibility for the security of customer systems is up to the customer. To that end, we recommend a number of best practices.

Strong Recommendations

Strong recommendations are items every enStratus customer should implement regardless of business need.

- Begin building machine images using minimalist, trusted public images. enStratus provides machine images for Ubuntu, CentOS, and Windows.
- Disable root logins. When launching servers in the Amazon cloud from images with the enStratus agent installed, do not launch those servers with AWS root keys. On Ubuntu systems, disable the ubuntu user.
- Do not build machine images from operational systems. Build images from pristine systems with minimal software.
- Never bundle interactive user accounts into machine images.
- Backup early; backup often.
- Leverage enStratus encryption for your backups.
- Store your backup encryption keys securely outside the cloud; enStratus has no access to recover those keys if you lose them.
- Implement an enStratus-supported intrusion detection system on your servers. enStratus currently supports OSSEC. Our architecture makes it easy to support other IDS systems, so please let us know if you prefer a different IDS tool.
- Pay attention to your intrusion detection alerts.
- Take advantage of enStratus firewall/security group support and limit firewall rules to specific IPs or IP ranges. Don't grant general security group to security group rules or open access rights.
- Exactly two people in your organization should have access to the cloud provider console (e.g. AWS console). All other users should manage the infrastructure through enStratus using individual user accounts. Administrators should access the cloud provider console only when enStratus is down or if the cloud provider console enables a function not available via enStratus.
- Follow all best practices related to building and maintaining applications on the Internet.
- Stay on top of operating system and software security patches.
- Grant/remove user shell/remote desktop accounts on-demand. Don't leave interactive accounts on servers in the cloud.
- Remove a user's console access when their job function no longer involves infrastructure management. Removing a user's console access also removes all shell/remote desktop access privileges.
- Leverage hardening tools like Bastille to harden your operating system before bundling a machine image.
- AWS S3 encryption isn't as useful as it appears.
- Implement a change detection tool, like tripwire.

General Recommendations

Our general recommendations are common best practices, but they may not be suitable to specific business needs.

- Use enStratus support for file system encryption. Use XFS + RAID0 to mitigate the performance hit.
- Implement multi-factor authentication when available. Do not use Open ID.
- Do not ask a single virtual server to perform multiple roles. A database server is a database server. A Tomcat server is a Tomcat server. A load balancer is a load balancer.
- Don't store any authentication credentials on the file system. If you absolutely must do so, make sure the file permissions are set so that only the application user can read them. Furthermore, make sure the file cannot be exposed to network services. Keep in mind that there is NO REASON WHATSOEVER your custom applications should rely on credentials in permanent configuration files. enStratus will pass in this data when a service starts up. You can therefore read the data and delete the file.
- enStratus will change the name of the default admin user on database engines that support it. Leverage this feature.
- Avoid making manual changes whenever possible. Leverage the automation tools supported by enStratus, such as Chef, Puppet or CFEngine.
- If you have to make manual changes, always use sudo.

enStratus™ is a cloud infrastructure management solution for deploying and managing enterprise-class applications in public, private and hybrid clouds. enStratus has a multi-cloud architecture that provides governance, automation and cloud independence.

We deliver governance for the cloud through a patent-pending security architecture and a powerful management console across all leading public and private clouds including Amazon Web Services, AT&T Synaptic Storage, Bluelock, Cloud Central, Cloud.com, CloudSigma, EMC Atmos, Eucalyptus, GoGrid, Google Storage, Nimbula, OpenStack, Rackspace, ServerExpress, Terremark, VMware and Windows Azure.

Based in Minneapolis, Minnesota, enStratus serves hundreds of organizations worldwide including Korea's largest telecom provider KT, Silanis, SAIC, and The Cloud Security Alliance.

<http://www.enstratus.com> | 612.746.3091 | contact@enstratus.com