Web site technical requirements for hosting:

Sites must be developed in versions compatible with the technologies listed below supported exclusively on UNIX/Linux operating system:

1. Apache 2.4.6 Web Server with PHP

   (Versions 7.3.11, 7.2.10, 5.6.25)

   JavaScript (NodeJS 12.10.0, npm 6.10.3)

   Memcache 1.4.15

   Redis 3.2.4

2. Web Server Apache 2.4.6 with Tomcat 7.0.54 Java 1.8.5

   Database:

   MariaDB 10.3.27 (MySQL 8.0)

   PostgreSQL 9.4.6

   Postfix 2.10.1 (only localhost relay)

   ModSecurity 2.9.0 (enabled)

   Selinux 3.13.1 (enabled)

For design, optimization and performance criteria, aimed at maintaining functional compatibility, sites whose pages are based on JAVA technology are separated from those using PHP on different servers.

The content management software (CMS) used by customers to develop their sites must be in correspondence with the versions of programming languages supported.

In all cases, it is always advisable to refer to the compatibility report announced by the manufacturer. In this aspect it is important to keep in mind that, in general, in Open Source and Free Software technologies, compatibility is only guaranteed between different releases within the same kernel version. For example, the manufacturer guarantees compatibility between all releases within PHP5, but not between PHP4, PHP5 or PHP7 since they are different kernel versions. The same applies to MySQL or PostgreSQL.

On the shared web hosting platform PHP runs in secure mode, so some functions that can compromise the security of the site and allow attackers to inject commands or execute scripts from your web sites to the server are disabled. The disabled functions are mostly those recommended in this link https://www.php.net/manual/es/features.safemode.functions.php.

The ModSecurity security module and the SELinux tool in enforcing mode, as well as an AntiDOS module to secure the service and sites against malicious attacks and intrusions run on Apache web servers. The sites need to be designed with the established requirements so that they can function properly in a secure environment.

Customers with PHP Linux sites are advised not to use insecure PHP functions, use the apache rewrite module to enable clean URLs (friendly) and keep their sites up to date to avoid problems with the security module as it denies dirty URLs that it finds suspicious.

It is recommended to enable the use of memcached or redis in the site configuration to improve performance. Some types of CMS, LMS or Framework have a built-in option to enable the use of these features.

About the JavaScript NodeJS application:

Due to the limitations of serving in the shared environment in terms of TCP port mapping per application on sites, the JavaScript application will be reachable only in the local environment via TCP localhost:port. Remote access would be via the site URL http://www.nombresitio.[nat][co].cu/node/<mynodejsapp_dir>/. To avoid the need to assign more than one TCP port, you must implement all the JavaScript NodeJS functionalities required by your site in a single application, complying with the conditions listed below. In addition, you must always maintain the same working directory and initial loading file of the application so that you do not have to modify the initial enabled configuration.

To enable the application in NodeJS, you must observe the following conditions that must be met.

- ✓ The limit on the number of JavaScript NodeJS applications per site is one. It must be implemented in such a way that it includes all the necessary functionalities for the site.
- ✓ In order to have the JavaScript NodeJS application running, a TCP port, which is mandatory will be assigned to avoid conflicts with other applications. Note that this is a shared environment.
- ✓ Configure the application's family parameter for IPv4.
- ✓ You must place your JavaScript application directory inside the htdocs/node/ directory structure as a ProxyPass /node/ type redirection is defined in the site configuration pointing to localhost by the TCP port we assign to it. For example, suppose your application is located in the mynodejsapp_dir/ directory. This directory must be located in htdocs/node/ <mynodejsapp_dir>/.
- ✓ You must inform the technical support team of the initial loading file, e.g. index.js, server.js, app.js, etc.
- ✓ Once you place your JavaScript application in a directory within htdocs/node/, and we have verified that it has been configured for the TCP port we assigned to it, we would compile the application and enable it on the web platform. Make sure that your application does not have other ports configured in different files.
- ✓ Whenever you make changes into your JavaScript application, you must inform the technical support team so that they can stop the service and proceed to re-compile and re-load the application again.
- ✓ Preferably, keep the JavaScript application always in the same directory.
- ✓ If you change the name of the directory containing the application or the initial loading file you must inform us as we must update it in the configuration of your javascript nodejs service, re-compile it and restart it.
- ✓ Implement all the JavaScript NodeJS functionalities required by your site in a single application.
- ✓ Remote TCP protocol access to the JavaScript application will not be allowed.

Connection to external or third party services:

Connection to external or third party services is allowed, enabling going out to other services located outside the server where the site is hosted, which imply the interaction of this with other systems and applications external to the service, whether they are national services (Cuban) or on the Internet (outside Cuba), which meet certain characteristics, among which are:

- ✓ Access to XETID's payment gateway.
- ✓ Access to RSS channels. Content syndication.
- ✓ Registration through social network accounts.
- ✓ Access to update repository, extensions and CMS connectors and the most popular Frameworks.

Upon customers' demand, the inclusion of access to new APIs, RSS channels or update repositories will be assessed by the pertinent technical and commercial areas.

Access to external services is limited mainly for security reasons and it may be the case that access to the repository of some of the components is not authorized.

The types of access to authorized external elements do not contemplate all types of components, but if the customer sends to the 24-hour Web Hosting Assistance the FQDN or the access URL to the manufacturer's web page of each component to be installed, or to the API, the corresponding analysis will be carried out by the areas involved (technical, commercial and security area). If such request is feasible and does not conflict with the premises established for security and quality of service, online access from the web platform will be allowed.

It is important that customers know that authorizing the online access to the repository of web technologies, extensions and components, in cases where the component or extension, once deployed, depends on connections to third-party services for its operation. It does not guarantee its usability once deployed, since this access will not be authorized.

Example: A website could enable a plugin for anti-spam security on web forms, but this plugin performs this function using a connection to the manufacturer's site, or others, for control by querying blacklists of different security companies. This type of access would not be allowed, among other reasons because the web anti-spam functionality has many ways to be controlled by enabling captcha, honeypot techniques by timestamp, or certain types of components that do not rely on external links. Also, because the security module of the web platform and the IDS component of the Data Center firewall provide for this type of functionality. Adding components that perform this task depending on connections to external services would only increase the processing load and unnecessary traffic congestion in the service, which would ultimately result in slow loading of the sites, taking into account that it is a shared web environment.

In case of difficulties when installing or updating the site and its components online, we suggest that you first deploy the web site in a local environment and then upload it to the production server already running. Secondly, download the components and deploy them manually.

If enabling access to the repository of some type of component or plugin from a different manufacturer, not included in the official repository of the type of base technology of your website (CMS, LMS or FrameWork) is required, we suggest the customer to download the components and deploy them manually.

Other aspects to take into account:

- ✓ The allowed messaging is only SMTP relay from the local server where the site is hosted. For this purpose, you can use an email account under enet.cu, nauta.cu, or other domains managed by ETECSA's mail server as from of the messages sent from your web site. If you wish to use another mail from belonging to a domain not managed by ETECSA, you must enable an SPF record in such domain to authorize the ETECSA's Web Hosting Service mail servers to send the messages with that from. You must always inform the technical support team of the mail from you will use to authorize it in the site configuration.
- ✓ It is recommended to enable verification against non-human manipulation in all data capture forms, such as user registration or password reset forms, web forms, contact forms, node forms and comment forms. For this purpose, you can use CAPTCHA or an invisible system using honeypot methods per timestamp to prevent spambots from filling out forms on your site.
- ✓ Disk space quotas are measured by the actual space taken up by the site and its database once mounted on the server in production. So it should be noted that data volumes differ depending on the file system on which they are located. In our case the file system used is NFS over XFS on a UNIX/Linux platform, so the values obtained by the customer on a different file system, such as Windows NTFS, Linux ext3 or ext4 or other, during the development stage when calculating the site weight will not be the real space it would occupy once it is hosted on our servers. When the customer reaches the limit established for its

disk space quota, the customer will not be able to continue updating by any of the enabled ways (FTP, HTTP).

- ✓ Secure FTP access is offered to update sites only from national networks, with SSL authentication, so the FTP client program must support SSL protocol during the authentication. We recommend Filezilla using "FTPES -Ftp protocol over explicit TLS/SSL". It is suggested to enable HTTPS SSL and perform content updates from international networks via the web through this protocol.
- ✓ Site visit statistics available in the utility web service.
- ✓ Weekly backup to the site and database with 30-day retention.
- ✓ If your Web site has a content management interface to update information and administration via web and you wish to secure it by enabling HTTPS, you must contact technical support team and provide them with the URLs/Directories susceptible to be protected under this security mechanism. The certificate is self-signed by the platform, so it will only be used for administration, but if you have an official certificate for the site, generated by an international certifying entity, it is also accepted.

About the databases:

For sites with more than one database, these must be of the same technology and version and be associated with the type of platform contracted, according to the table with available technologies by platform on the first page of this document.

The databases and the web site encoding must be UTF8, preferably using the utf8mb4 charset for table creation. In the case of MySQL databases, the database engine should be InnoDB and preferably enable the dynamic format. We suggest you to edit the script with the instructions for the creation of the database structure and change in the CREATE TABLE instructions,

ENGINE=MySAM to ENGINE=InnoDB, and add the parameter

ROW_FORMAT=DYNAMIC parameter to each table. Take into account the maximum length of varchar type data with respect to the selected charset. See these links https://mariadb.com/kb/en/troubleshooting-row-size-too-large-errors-with-innodb/, https://mariadb.com/kb/en/innodb-row-formats-overview/. Once the change is made, you will be able to introduce your tables into the production server.

The reason why the engine type or ENGINE InnoDB is mandatory is that its database is hosted on a distributed server scheme in high availability, in a MariaDB cluster, and this requirement allows guaranteeing the integrity of the tables in high availability schemes. The MyISAM engine is more lightweight, but does not allow the control mechanisms required in this scheme. See also the following links for a better understanding https://mariadb.com/kb/en/convertingtables-from-myisam-to-innodb/, https://mariadb.com/kb/en/choosing-the-right-storageengine/.

Another important aspect to keep in mind is that in the MariaDB cluster all nodes can write data to the tables, i.e. a multi-master replication is executed. Imagine a situation where all nodes in the cluster try to insert rows into the same table at the same time. The result could be duplicate values for any column using auto_increment. To avoid such conflicts, the cluster increments column values based on the number of nodes in the cluster. On member nodes, the 'wsrep_auto_increment_control' parameter to 'ON' is established to indicate to change the value of MariaDB's 'auto_increment_increment' and auto_increment_offset' automatically. This will prevent 'duplicate entry' errors. It is recommended not modify these variables to ensure the integrity of the multi-master replication. More information on this can be found at these links https://mariadb.org/auto-increments-in-galera/, https://galeracluster.com/library/kb/autoincrement-multiples.html.

Due to security policies, database administration is not allowed through the Web server where the site is hosted. For this purpose there is a Web manager (e.g. PhpMyAdmin) on a separate server with all the security requirements, with restricted access only for customers who have a database hosted. We ask customers to

avoid installing PhpMyAdmin, PGAdmin, Adminer or other web tools to manage their databases within the site structure.

Based on the above policy, the database hosting service is not provided as a service in itself, but must always be associated with a contracted website

Whenever the web application allows it, it is recommended to use privileged roles to access databases, such as db_datawriter, db_datareader and dbowner.

About enabling the HTTPS Protocol:

In the initial configuration, SSL is enabled to sites hosted on the web platform using a certificate self-signed by ETECSA only to access to the site administration directories. Since the certificate is self-signed, the customer will not be able to use it to enable HTTPS SSL to all site content. To do so, the customer must obtain an official certificate issued by a certifying entity and submit it through the service support team.

In order to comply with what is established in terms of guaranteeing the proper security and correct functioning of the sites, customers must implement the following requirements:

1. TLS protocol must be used, only versions v1.2 and v1.3 are allowed. TLS v1.3 will be established by default and as the main protocol.

2. Set up the SSL/TLS server in such a way that it properly selects the combination of cryptographic algorithms, which will be:

- ✓ ECDHE-RSA-AES256-GCM-SHA384
- ✓ DHE-RSA-AES256-GCM-SHA384
- ✓ ECDHE-RSA-AES128-GCM-SHA256
- ✓ DHE-RSA-AES128-GCM-SHA256

3. Authentication will be performed using the DiffieHellman Key agreement protocol, generated using OpenSSL v1.1.1b, with value equal to 4096bits.

4. Enable the Forward Secrecy option.

5. Disable TLS renegotiation mechanism, which can be initiated by the client.

6. Disable TLS compression.

7. Enable persistence on HTTP connections.

8. Enable OCSP Stapling.

9. Make use of a 301 redirect from HTTP to HTTPS, forcing the use (by default) of HTTPS.

10. Enable the TLS Fallback Signaling Cipher Suite Value (SCSV) option on the web server.

11. Enable the Secure directive in all the cookies used by the web application.

12. Enable HSTS (HTTP Strict Transport Security). Use a "max-age" value of at least 10886400 (18 weeks) and preferably 6 or 12 months ("31536000").

We also recommend the following:

The web application hosted on the UNIX/Linux web platform has the HTTPS secure protocol enabled by a full redirect to HTTPS.

Check the http headers that could be sending the site in an unsecured way and solve this problem as soon as possible. At the same time we suggest you to check out the security recommendations provided by the following online verification tools: https://letsdebug.net/ (by Alex Zorin) https://check-your-website.server-daten.de/ (by Jürgen Auer) https://whynopadlock.com/ (by LexiConn) https://www.ssllabs.com/ssltest/ (by Qualys) https://securityheaders.com/ (by Probely)

ETECSA, during the service configuration, applies the 301 redirection from HTTP to HTTPS only for known directories and administration pages, according to the web content management technology used by the site, therefore, the customer is responsible for enabling it for the entire website.

About http/2.

In order to reinforce security and improve the technological performance of the HTTPS protocol, the http/2 protocol is enabled on the Shared Web Hosting platform.

To avoid errors that may occur when executing certain functionalities of the web pages when the browser used does not support this version of the http protocol, it is suggested to website owners to enable HTTPS, and to inform their users that they must have browsers supporting the http/2 protocol, such as Opera, Firefox or Chrome in their most updated versions.

About obtaining SSL Certificates.

The web hosting service offer does not include SSL certificate. Our company is not authorized to issue these certificates by the relevant authority in Cuba. Due to regulations imposed by the laws of the economic, commercial, technological and financial blockage of the United States against Cuba, there is no certifying entity in our country that can issue certificates recognized by the most popular web browsers.

The customer must obtain certificates from one of the international certifying entities. **Always** using a secure channel, customers must upload it to the platform and notify the 24-hour Web Hosting Assistance Service.

The pki/ directory has been enabled outside the root structure of the site for customers to deliver the certificate via TLS/SSL by ftp. **It should never be sent by email.**

For the SSL certificate generation for your site via third parties, the certification authority requires to verify the authenticity of the domain and its membership. There are several ways or challenges to verify the domain:

1. Sending an email to the domain in this case (@nombresitio.nat.cu).
2. DNS (CNAME), inserting a record in the domain record (usually inCPANNEL etc).
3. HTTP/HTTPS file upload, uploading a txt file to the root of the site.

The variant that we can guarantee and currently suggest is the third one, **HTTP/HTTPS file upload**, and it works perfectly, but for this variant to work, the customer must create in the site root a directory called ".well-known", or another name, something that can be done via ftp. But if the site has https enabled with an expired certificate or a self-signed one, or has IP filtering enabled so that it is visible only from a certain source that does not include the IP of the certifying entity, this verification will not work.

To perform certificate generation by the **HTTP/HTTPS file upload** method, three important conditions must be met:

1. The customer must create a hidden verification directory, generally named ".well-known", with the dot in front and with read permissions for the user running the site, in our case the customer's own ftp user, and place the files with the validation information that the certification authority requires during the domain validation.

2. The site must be accessible by the certification authority at the time of generation via http or https. It is important that the site does not have IP filtering enabled.

3. If verification occurs via https, the SSL certificate enabled on the site must be valid at the time of new certificate generation, otherwise, the customer must temporarily disable https. It is advisable that customers do not wait for the previous certificate to expire so they can do this via https.

As this is a third party service, we cannot be held responsible if there are any difficulties in obtaining the SSL certificate. Always check out to the provider's documentation.

Password change for both the FTP account and the Database user.

The service utilities portal site is enabled for password change, where you will be able to update it every time that you wish it.

In this site, if you know the previous password, you have 5 attempts to change it. If you exhaust the 5 attempts and fail to change the password, you can request to change your password to the 24-hour Web Hosting Assistance Service by emailing to: hosting@enet.cu.

The request will be accepted if it is made through the e-mail address belonging to the web site owner (contract holder), declared as part of the contact details required in the contracting process, providing, in the e-mail body, the following data, which are mandatory:

- ✓ The site name.
- ✓ Contract number.
- ✓ Name and Last name of the person responsible for the site (contract holder).

With this information the 24-hour Web Hosting Assistance Service will proceed to make the request and will inform you of the change made.

On the same utility site, you can update your contact information, telephone and e-mail, to ensure that you receive the password expiration notifications to the correct address.

Once the password expires and you have exhausted the 5 attempts to obtain a new one, or in case you forget it, you can request the change to the 24-hour Web Hosting Support Service by emailing to: hosting@enet.cu as described above.

Remember that, for greater security, you should change your password from time to time, or when you detect inappropriate use of your service by others.

Requirements for FTP account and Database user passwords.

The password must meet and contain the following requirements:

- Length between 8 and 15 characters.
- Lowercase and uppercase letters.
- Numerical characters and symbols.
- Not to repeat the last 5 passwords.

Expiration time.

The password is valid for 300 days. If the e-mail address provided by the customer is valid, a notification should be sent automatically when the password is about to expire, once expired, the notification that the password has expired is also sent to the customer so that the customer can update it through the service utilities portal site.