Image not found or type unknown

Installation and configuration guide

1. WHMCS setup(install/update)

MinIO S3 module **WHMCS**

Order now | Download | FAQ

Module is coded ionCube v13

Supported php version:

- php 7.4 WHMCS 8.11.0 -
- php 8.1 WHMCS 8.11.0 +
- php 8.2 WHMCS 8.11.0 +

To install and update a module, you must perform one and the same action.

1. Download the latest version of the module.

PHP 8.2

```
wget http://download.puqcloud.com/WHMCS/servers/PUQ_WHMCS-MinIO-S3/php82/PUQ_WHMCS-MinIO-S3-
latest.zip
```

PHP 8.1

```
wget http://download.puqcloud.com/WHMCS/servers/PUQ_WHMCS-MinIO-S3/php81/PUQ_WHMCS-MinIO-S3-
latest.zip
```

PHP 7.4

```
wget http://download.puqcloud.com/WHMCS/servers/PUQ_WHMCS-MinIO-S3/php74/PUQ_WHMCS-MinIO-S3-latest.zip
```

All versions are available via link: http://download.puqcloud.com/WHMCS/servers/PUQ_WHMCS-MinIO-S3/

2. Unzip the archive with the module.

```
unzip PUQ_WHMCS-MinIO-S3-latest.zip
```

3. Copy and Replace "puqMinIOS3" from "PUQ_WHMCS-MinIO-S3" to "WHMCS_WEB_DIR/modules/servers/"

Setup guide: MinIO S3 setup

MinIO S3 module WHMCS

Order now | Download | FAQ

There are many ways to install MinIO. Below we will introduce the installation method from binaries. In the following description, we will provide additional steps beyond the basic installation to set up the service properly. The description will include, among others, setting up the service, nginx proxy and SSL certificates.

In the current example, we will use the Debian 10 operating system.

1 - Installing and configuring the MinIO server

If you haven't updated the package database recently, update it:

```
sudo apt update
```

Then download the Minio server binary from the official website:

```
wget https://dl.min.io/server/minio/release/linux-amd64/minio
```

Output

```
# wget https://dl.min.io/server/minio/release/linux-amd64/minio
--2022-08-10 10:01:59-- https://dl.min.io/server/minio/release/linux-amd64/minio
Resolving dl.min.io (dl.min.io)... 178.128.69.202, 138.68.11.125
Connecting to dl.min.io (dl.min.io)|178.128.69.202|:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 96968704 (92M) [application/octet-stream]
Saving to: ‘minio’

minio
100%[=====================================================
92,48M 16,7MB/s in 6,8s

2022-08-10 10:02:07 (13,6 MB/s) - ‘minio’ saved [96968704/96968704]
```

Once the download is complete, a file called minio will be in your working directory. Use the following command to get the executable:

```
sudo chmod +x minio
```

Now move the file to the `/usr/local/bin` directory, where the Minio systemd startup script expects to find it:

```
sudo mv minio /usr/local/bin
```

This will allow us to write a service unit file in the next steps of this tutorial to ensure that Minio starts up automatically on system boot.

For security reasons, it is recommended to avoid running the Minio server as root. This will limit the damage that can be done to the system in the event of a security breach. Because the systemd script you'll use in step 2 is looking for an account and group called minio-user, create a new user with that name:

```
sudo useradd -r minio-user -s /sbin/nologin
```

In this command, you used the -s flag to set up `/sbin/nologin` as the shell for minio-user. This is a shell that does not allow the user to log in, which is not necessary for minio-user.

Next, transfer ownership of the Minio binary to **minio-use**r:

```
sudo chown minio-user:minio-user /usr/local/bin/minio
```

Next, you need to create a directory where Minio will store the files. This location will be where you store the buckets that you will use later to organize the objects you store on your Minio server.This tutorial will use the **minio** directory name:

```
sudo mkdir /usr/local/share/minio
```

Give **minio-user** ownership of this directory:

```
sudo chown minio-user:minio-user /usr/local/share/minio
```

Most server configuration files are stored in the `/etc` directory, so this is where you need to create your configuration file:

```
sudo mkdir /etc/minio
```

Give **minio-user** ownership of this directory:

```
sudo chown minio-user:minio-user /etc/minio
```

Use **Nano** or your favorite text editor to create the environment file needed to change the default configuration:

```
sudo nano /etc/default/minio
```

After opening the file, add the following lines to set a few important environment variables in the environment file:

> MINIO_ACCESS_KEY="minio"
> MINIO_VOLUMES="/usr/local/share/minio/"
> MINIO_OPTS="-C /etc/minio --address :9000 --console-address :9001"
> MINIO_SECRET_KEY="miniostorage"

Let's take a look at these variables and the values you have set:

- **MINIO_ACCESS_KEY**: This variable specifies the access key you will use to access the Minio browser user interface.
- **MINIO_SECRET_KEY**: This variable specifies the private key you will use to pass login credentials to the Minio interface. In this tutorial, we'll use the miniostorage value, but we recommend choosing a different, more complex password to keep your server secure.
- **MINIO_VOLUMES**: This variable specifies the storage directory you have created for your buckets.
- **MINIO_OPTS**: This variable determines where and how the server serves the data. The -C flag tells Minio the configuration directory to use, and the --address flag specifies the IP

address and port to bind to. If no IP address is specified, Minio will bind to whatever address is set on the server, including localhost and any Docker-related IP addresses, so we recommend that you directly specify the IP address here. You can change the default port 9000 if you like.
Save and close the environment file after making changes.

You have installed Minio and set a number of important environment variables. Next, you need to configure the server to run as a system service.

2 - Installing the Systemd MinIO startup script

In this step, you will set up the Minio server to manage it as a systemd service.

Create a file `/etc/systemd/system/minio.service`

```
sudo nano /etc/systemd/system/minio.service
```

File contents:

```
[Unit]
Description=MinIO
Documentation=https://docs.min.io
Wants=network-online.target
After=network-online.target
AssertFileIsExecutable=/usr/local/bin/minio

[Service]
WorkingDirectory=/usr/local/

User=minio-user
Group=minio-user

EnvironmentFile=/etc/default/minio
ExecStartPre=/bin/bash -c "if [ -z \"${MINIO_VOLUMES}\" ]; then echo \"Variable MINIO_VOLUMES not set in /etc/default/minio\"; exit 1; fi"

ExecStart=/usr/local/bin/minio server $MINIO_OPTS $MINIO_VOLUMES
```

```
# Let systemd restart this service always
Restart=always

# Specifies the maximum file descriptor number that can be opened by this process
LimitNOFILE=65536

# Disable timeout logic and wait until process is stopped
TimeoutStopSec=infinity
SendSIGKILL=no

[Install]
WantedBy=multi-user.target

# Built for ${project.name}-${project.version} (${project.name})
```

Then run the following command to reload all **systemd units**:

```
sudo systemctl daemon-reload
sudo systemctl enable minio
```

Now that the systemd script is installed and configured, it's time to start the server.

3 - Starting the MinIO Server

In this step, you will start the server and change the firewall settings to allow access through the browser interface.

Start Minio server:

```
sudo systemctl start minio
```

Then check the Minio's status, the IP address it's bound to, memory usage, and more with the following command:

```
sudo systemctl status minio
```

The result will look like this:

4 - Securing Access to MinIO

Server with Let’s Encrypt SSL/TLS Certificate

You need to replace **yourdomain.com** with your own domain

Certbot is a console based certificate generation tool for Let's Encrypt.

In this recipe, we will generate a Let's Encypt certificate using Certbot. This certificate will then be deployed for use in the MinIO server.

Install Certbot

```
sudo apt update
sudo apt install certbot nginx python3-certbot-nginx -y
```

Set up Nginx proxy with MinIO Server

Proxy all requests

```
rm /etc/nginx/sites-enabled/default
nano /etc/nginx/sites-enabled/minio
```

```
server {
        listen 80 default_server;
        server_name yourdomain.com;
        return 301 https://$host$request_uri;
}



server {
        listen 443 ssl http2;
        server_name yourdomain.com;

        ssl_certificate /etc/letsencrypt/live/yourdomain.com/fullchain.pem;
        ssl_certificate_key /etc/letsencrypt/live/yourdomain.com/privkey.pem;
        ssl_trusted_certificate /etc/letsencrypt/live/yourdomain.com/cert.pem;
        ssl_session_timeout 20m;
        ssl_ciphers ECDHE-RSA-AES128-GCM-
SHA256:ECDHE:ECDH:AES:HIGH:!NULL:!aNULL:!MD5:!ADH:!RC4;
        ssl_protocols TLSv1 TLSv1.1 TLSv1.2;
        ssl_prefer_server_ciphers on;
        ssl_verify_client off;

        ignore_invalid_headers off;

        client_max_body_size 0;

        proxy_buffering off;

        location / {
                proxy_set_header X-Real-IP $remote_addr;
                proxy_set_header X-Forwarded-For $proxy_add_x_forwarded_for;
                proxy_set_header X-Forwarded-Proto $scheme;
                proxy_set_header Host $http_host;

                proxy_connect_timeout 300;
                proxy_http_version 1.1;
                proxy_set_header Connection "";
                chunked_transfer_encoding off;

                proxy_pass http://localhost:9001;
 }
}
```

Obtain the SSL/TLS Certificate

```
sudo certbot --nginx -d yourdomain.com
```

Restart **nginx** wer server

```
sudo service nginx restart
```

In order for the certificate to be updated automatically, you must add to the crontab

```
crontab -e
```

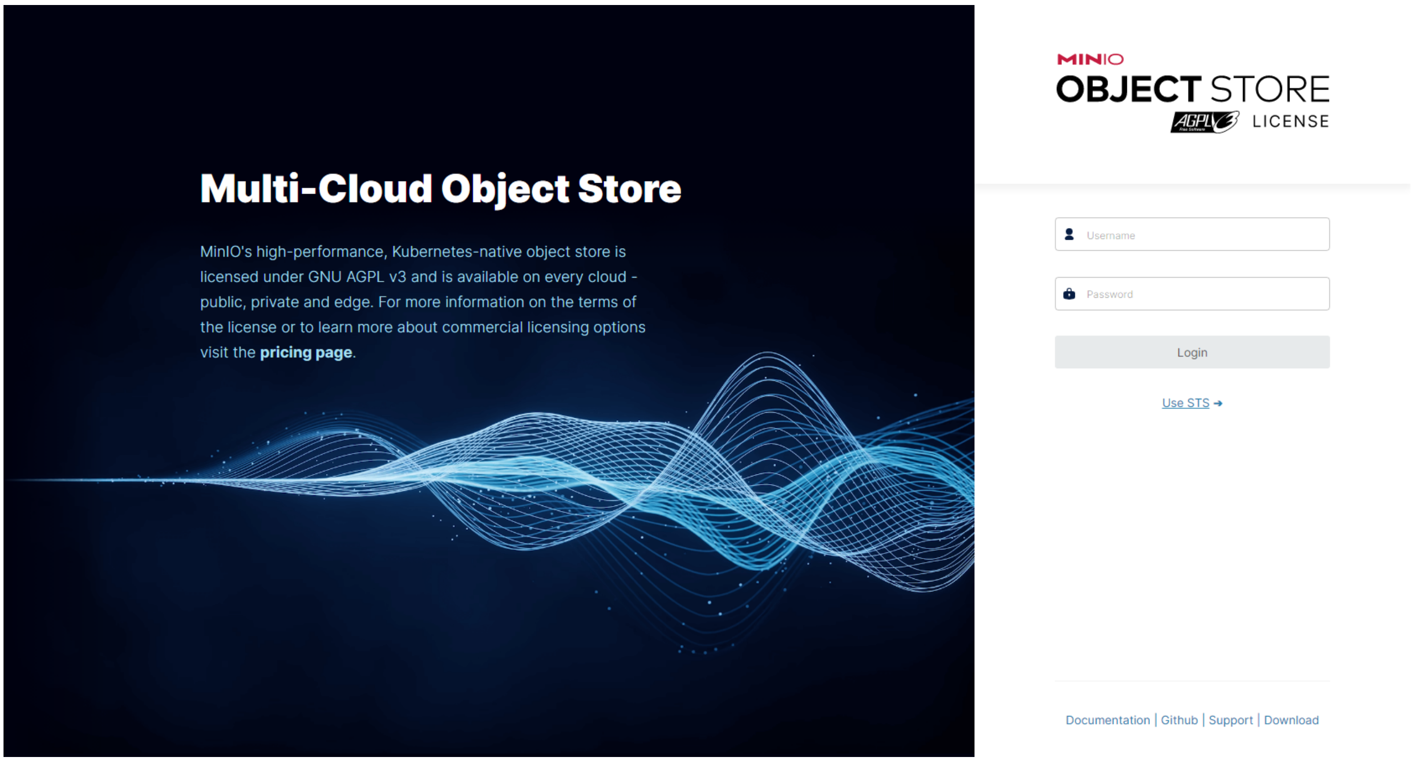
```
0 12 * * * /usr/bin/certbot renew --quiet
```

The configuration is now complete.
Login to the server

url: https://yourdomain.com/

For authorization, use the data that was written in the file `/etc/default/minio`

Username: minio
Password: miniostorage


Multi-Cloud Object Store

MinIO's high-performance, Kubernetes-native object store is licensed under GNU AGPL v3 and is available on every cloud - public, private and edge. For more information on the terms of the license or to learn more about commercial licensing options visit the **pricing page**.

Setup guide: WHMCS setup

MinIO S3 module **WHMCS**

Order now | Download | FAQ

Module is coded ionCube v13

Supported php version:

- php 7.4 WHMCS 8.11.0 -
- php 8.1 WHMCS 8.11.0 +
- php 8.2 WHMCS 8.11.0 +

To install and update a module, you must perform one and the same action.

1. Download the latest version of the module.

PHP 8.2

```
wget http://download.puqcloud.com/WHMCS/servers/PUQ_WHMCS-MinIO-S3/php82/PUQ_WHMCS-MinIO-S3-latest.zip
```

PHP 8.1

```
wget http://download.puqcloud.com/WHMCS/servers/PUQ_WHMCS-MinIO-S3/php81/PUQ_WHMCS-MinIO-S3-latest.zip
```

PHP 7.4

```
wget http://download.puqcloud.com/WHMCS/servers/PUQ_WHMCS-MinIO-S3/php74/PUQ_WHMCS-MinIO-S3-latest.zip
```

All versions are available via link:
http://download.puqcloud.com/WHMCS/servers/PUQ_WHMCS-MinIO-S3/

2. Unzip the archive with the module.

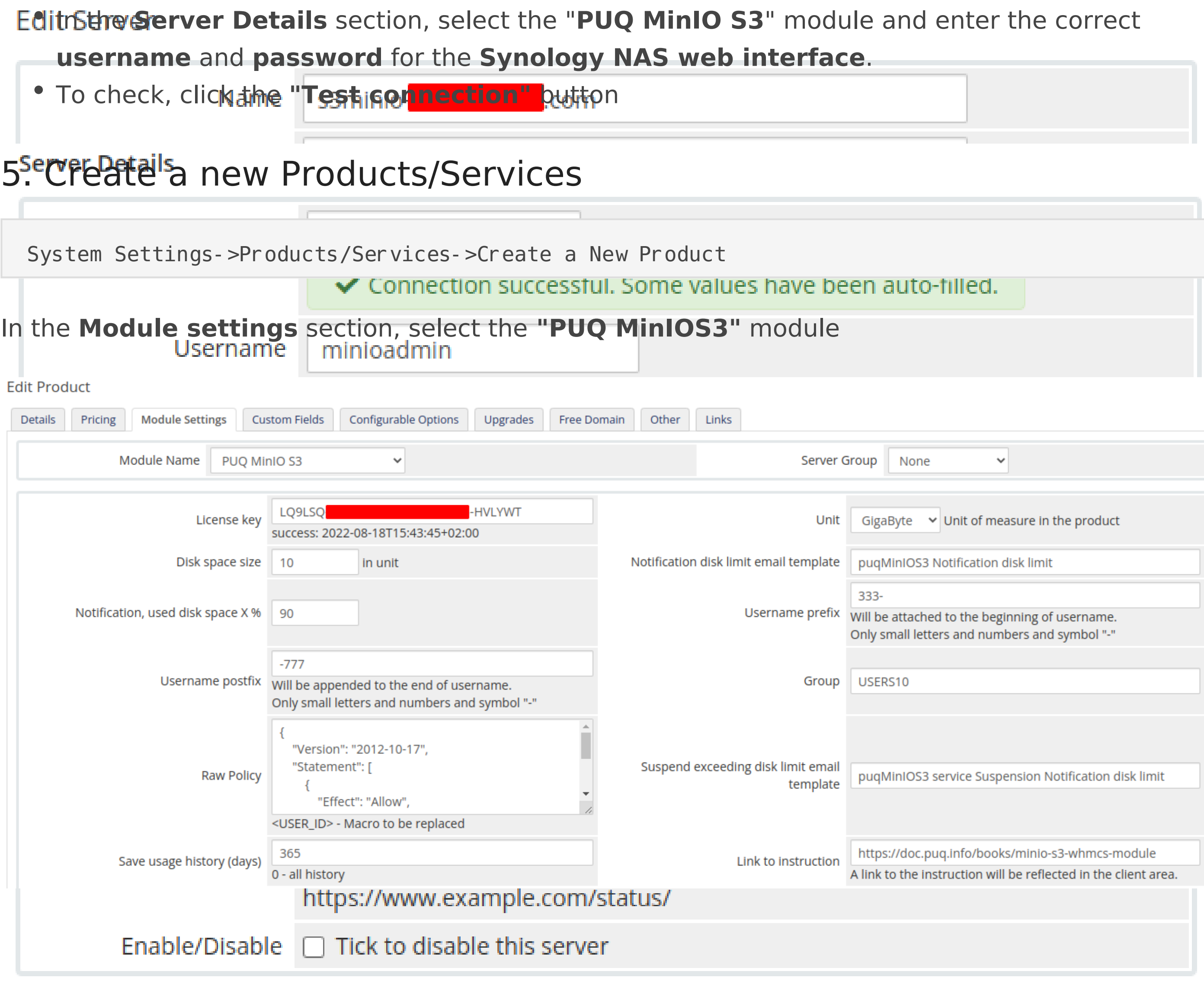
```
unzip PUQ_WHMCS-MinIO-S3-latest.zip
```

3. Copy and Replace "puqMinIOS3" from "PUQ_WHMCS-MinIO-S3" to "WHMCS_WEB_DIR/modules/servers/"

4. Create new server MinIO in WHMCS (System Settings->Products/Services->Servers)

```
System Settings->Servers->Add New Server
```

- Enter the correct **Name** and **Hostname**
- In the **Server Details** section, select the "**PUQ MinIO S3**" module and enter the correct **username** and **password** for the **Synology NAS web interface**.
- To check, click the **"Test connection"** button



Edit Server

Name s3minio[redacted].com

Server Details

Connection successful. Some values have been auto-filled.

Username minioadmin

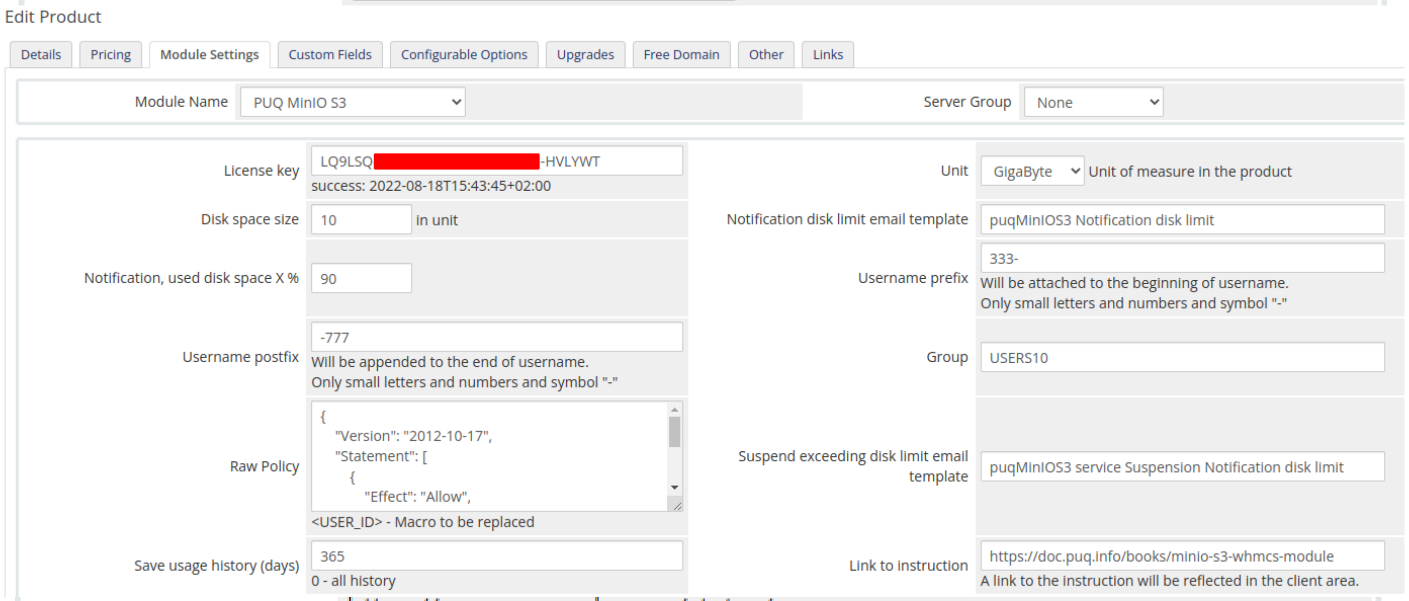
https://www.example.com/status/

Enable/Disable ☐ Tick to disable this server

5. Create a new Products/Services

```
System Settings->Products/Services->Create a New Product
```

In the **Module settings** section, select the **"PUQ MinIOS3"** module



Edit Product

Details | Pricing | Module Settings | Custom Fields | Configurable Options | Upgrades | Free Domain | Other | Links

Module Name: PUQ MinIO S3 — Server Group: None

License key	LQ9LSQ[redacted]-HVLYWT success: 2022-08-18T15:43:45+02:00	Unit	GigaByte Unit of measure in the product
Disk space size	10 in unit	Notification disk limit email template	puqMinIOS3 Notification disk limit
Notification, used disk space X %	90	Username prefix	333- Will be attached to the beginning of username. Only small letters and numbers and symbol "-"
Username postfix	-777 Will be appended to the end of username. Only small letters and numbers and symbol "-"	Group	USERS10
Raw Policy	{ "Version": "2012-10-17", "Statement": [{ "Effect": "Allow", <USER_ID> - Macro to be replaced	Suspend exceeding disk limit email template	puqMinIOS3 service Suspension Notification disk limit
Save usage history (days)	365 0 - all history	Link to instruction	https://doc.puq.info/books/minio-s3-whmcs-module A link to the instruction will be reflected in the client area.

- **License key:** A pre-purchased license key for the **"PUQ MinIOS3"** module. For the module to work correctly, the key must be active
- **Unit:** Packet disk space units
- **Disk space size:** Disk size in this product

- **Notification disk limit email template:** Email template that will be sent when the disk quota is exceeded in %
- **Notification, used disk space X %:** Sets a percentage parameter, after exceeding this parameter a notification will be sent to the user
- **Username prefix/Username suffix:** Necessary in order to generate a username for the service, in the format: **prefix<client_id>-<service_id>suffix**
- **Group:** The group that will be assigned to the user on the server side of the **MinIO S3**
- **Raw Policy:** The policy that is assigned to the user during creation on the server

Example:

In the given policy example. The user has the right to create buckets with a name starting with the username.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "admin:Heal",
                "admin:SetBucketTarget",
                "admin:TopLocksInfo",
                "admin:DataUsageInfo",
                "admin:GetBucketQuota",
                "admin:GetBucketTarget"
            ],
            "Resource": [
                "arn:aws:s3:::<USER_ID>*"
            ]
        },
        {
            "Effect": "Allow",
            "Action": [
                "s3:*"
            ],
            "Resource": [
                "arn:aws:s3:::<USER_ID>*"
            ]
        }
    ]
}
```

- **Suspend exceeding disk limit email template:** The template of the letter that will be sent to the client if his disk limit is 100% or less 100%
- **Save usage history (days):** The number of days it takes to save user disk usage statistics
- **Link to instruction:** Link to the instruction, if filled out, it will be reflected in the client area

Email Template (puqMinIOS3 Notification disk limit)

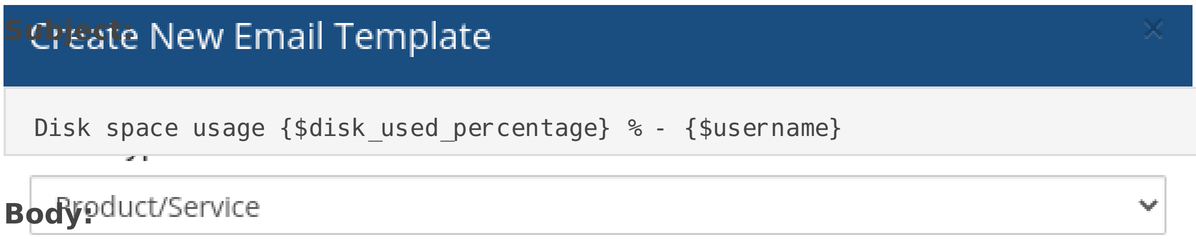
MinIO S3 module **WHMCS**

Order now | Download | FAQ

Create an email template for customer notifications.

```
System Settings->Email Templates->Create New Email Template
```

- **Email Type:** Product/service
- **Unique Name:** puqMinIOS3 Notification disk limit



Subject:

```
Disk space usage {$disk_used_percentage} % - {$username}
```

Body:

```
Dear {$client_name},


This letter informs you that the disk space usage limit is coming to an end.


Product/Service: {$service_product_name}
Due Date: {$service_next_due_date}


Username: {$username}


Disk limit: {$disk_limit_bytes*$unit_coefficient} {$unit}
Disk used: {$disk_used_unit} {$unit} ({$disk_used_percentage} %)
Disk free: {$disk_free_unit} {$unit} ({$disk_free_percentage} %)
```

```



{$signature}
```

Email Template (puqMinIOS3 service Suspension Notification disk limit)

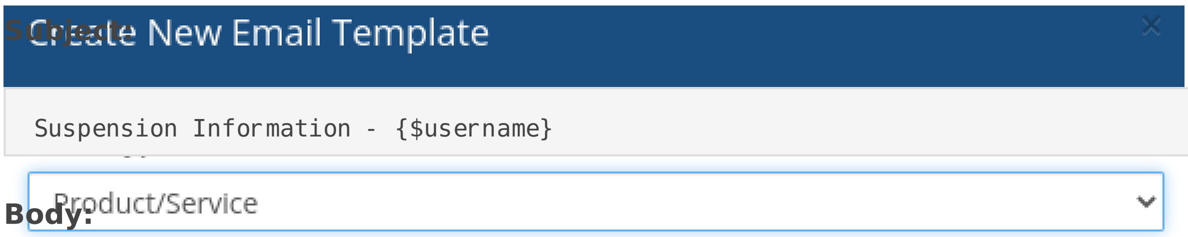
MinIO S3 module **WHMCS**

Order now | Download | FAQ

Create an email template for customer notifications.

```
System Settings->Email Templates->Create New Email Template
```

- **Email Type:** Product/service
- **Unique Name:** puqMinIOS3 service Suspension Notification disk limit



Subject:

```
Suspension Information - {$username}
```

Body:

```
Dear {$client_name},


This email informs you that the S3 account has been disabled due to running out of free space.


It is also possible to upgrade to a package with more space.


Product/Service: {$service_product_name}
Due Date: {$service_next_due_date}
```

```
Username: {$username}


Disk limit: {$disk_limit_bytes*$unit_coefficient} {$unit}
Disk used: {$disk_used_unit} {$unit} ({$disk_used_percentage} %)
Disk free: {$disk_free_unit} {$unit} ({$disk_free_percentage} %)


{$signature}
```

Add server (MinIO S3)

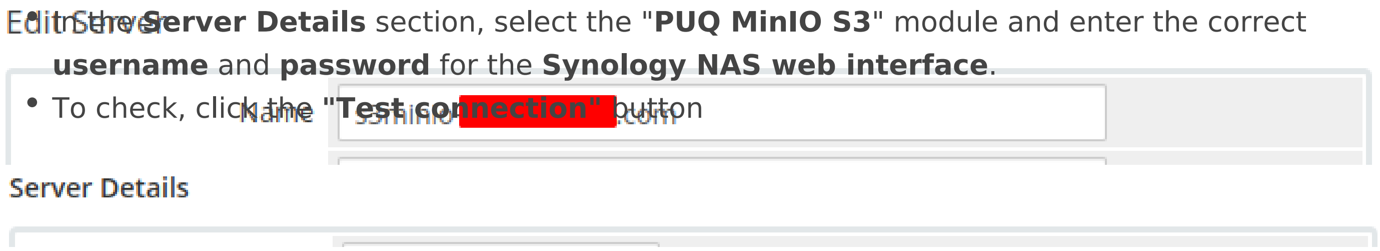
MinIO S3 module **WHMCS**

Order now | Download | FAQ

Add a new server to the system WHMCS.

```
System Settings->Servers->Add New Server
```

- Enter the correct **Name** and **Hostname**
- In the **Server Details** section, select the "**PUQ MinIO S3**" module and enter the correct **username** and **password** for the **Synology NAS web interface**.
- To check, click the **"Test connection"** button



Edit Server

Name s3minio[redacted].com

Server Details

Product Configuration

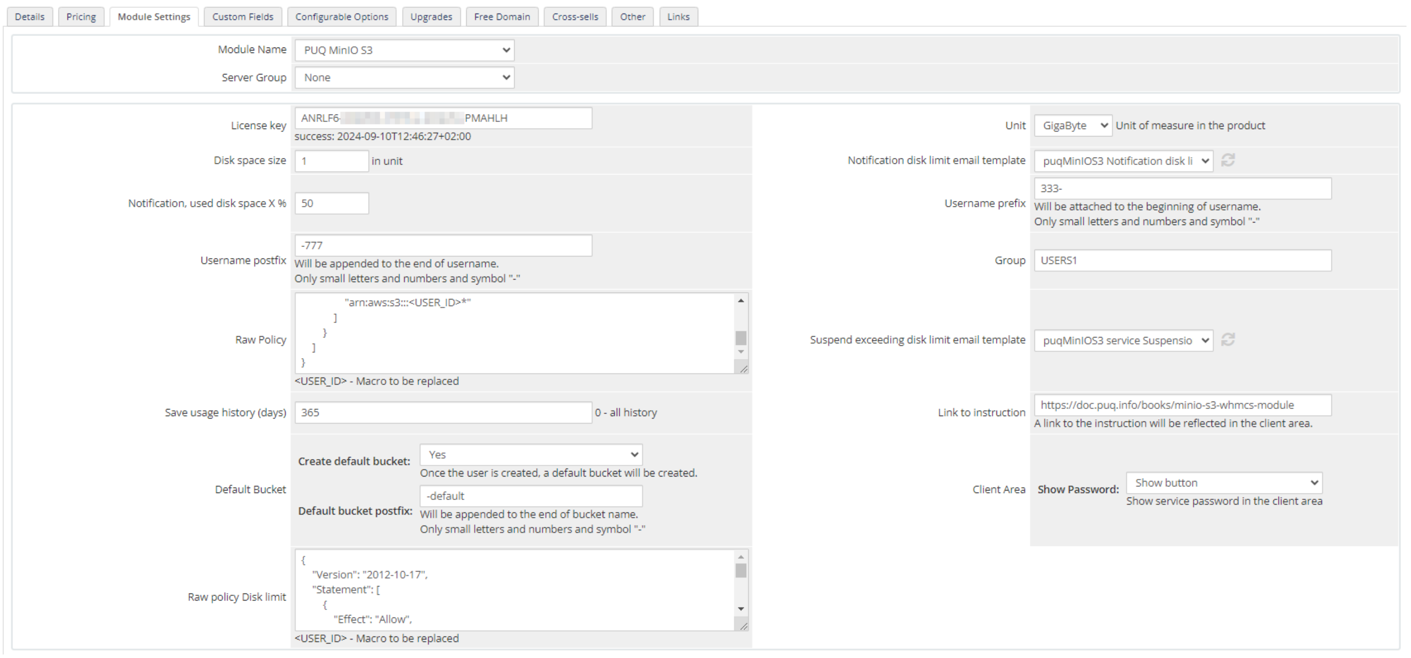
MinIO S3 module **WHMCS**

Order now | Download | FAQ

Add new product to WHMCS

```
System Settings->Products/Services->Create a New Product
```

In the **Module settings** section, select the **"PUQ MinIOS3"** module



Module Name	PUQ MinIO S3
Server Group	None

License key	ANRLF6-[redacted]PMAHLH success: 2024-09-10T12:46:27+02:00
Disk space size	1 in unit
Notification, used disk space X %	50
Username postfix	-777 Will be appended to the end of username. Only small letters and numbers and symbol "-"

Raw Policy	`"arn:aws:s3:::<USER_ID>*"`
	`]`
	`}`
	`]`
	`}`
	<USER_ID> - Macro to be replaced

Save usage history (days)	365 0 - all history
Default Bucket	**Create default bucket:** Yes Once the user is created, a default bucket will be created. **Default bucket postfix:** -default Will be appended to the end of bucket name. Only small letters and numbers and symbol "-"

Raw policy Disk limit	`{`
	`"Version": "2012-10-17",`
	`"Statement": [`
	`{`
	`"Effect": "Allow",`
	<USER_ID> - Macro to be replaced

Unit	GigaByte Unit of measure in the product
Notification disk limit email template	puqMinIOS3 Notification disk li
Username prefix	333- Will be attached to the beginning of username. Only small letters and numbers and symbol "-"
Group	USERS1
Suspend exceeding disk limit email template	puqMinIOS3 service Suspensio
Link to instruction	https://doc.puq.info/books/minio-s3-whmcs-module A link to the instruction will be reflected in the client area.
Client Area	**Show Password:** Show button Show service password in the client area

- **License key:** A pre-purchased license key for the **"PUQ MinIOS3"** module. For the module to work correctly, the key must be active
- **Unit:** Packet disk space units
- **Disk space size:** Disk size in this product
- **Notification disk limit email template:** Email template that will be sent when the disk quota is exceeded in %
- **Notification, used disk space X %:** Sets a percentage parameter, after exceeding this parameter a notification will be sent to the user
- **Username prefix/Username suffix:** Necessary in order to generate a username for the service, in the format: **prefix<client_id>-<service_id>suffix**
- **Group:** The group that will be assigned to the user on the server side of the **MinIO S3**
- **Raw Policy:** The policy that is assigned to the user during creation on the server

Example:

In the given policy example. The user has the right to create buckets with a name starting with the username.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "admin:Heal",
                "admin:SetBucketTarget",
                "admin:TopLocksInfo",
                "admin:DataUsageInfo",
                "admin:GetBucketQuota",
                "admin:GetBucketTarget"
            ],
            "Resource": [
                "arn:aws:s3:::<USER_ID>*"
            ]
        },
        {
            "Effect": "Allow",
            "Action": [
                "s3:*"
            ],
            "Resource": [
                "arn:aws:s3:::<USER_ID>*"
            ]
        }
    ]
}
```

- **Suspend exceeding disk limit email template:** The template of the letter that will be sent to the client if his disk limit is 100% or less 100%
- **Save usage history (days):** The number of days it takes to save user disk usage statistics
- **Link to instruction:** Link to the instruction, if filled out, it will be reflected in the client area
- **Default Bucket:** Options to enable or disable the creation of a default bucket, and also set its prefix
- **Client Area:** Client zone settings, show or not show the password in the client zone, type how to show the password.
- **Raw policy Disk limit:** The policy that will be applied to the client when the client runs out of space.

  Example:

  The user has the right to read and get, delete buckets whose name begins with the user's name.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "admin:Heal",
                "admin:SetBucketTarget",
                "admin:TopLocksInfo",
                "admin:DataUsageInfo",
                "admin:GetBucketQuota",
                "admin:GetBucketTarget"
            ],
            "Resource": [
                "arn:aws:s3:::<USER_ID>*"
            ]
        },
        {
            "Effect": "Allow",
            "Action": [
                "s3:GetObject",
                "s3:DeleteObject",
                "s3:ListBucket"
            ],
            "Resource": [
                "arn:aws:s3:::<USER_ID>*"
            ]
        }
    ]
}
```

Attention, policy recalculation occurs once a day during the collection of server statistics (UpdateServerUsage)