Managed Servers

Managed Servers

Nanaged Servers								
Managed Servers List Refree	sh Table $ \mathcal{G} $			items per p	age 20 💌	Search by name	e	٩
↓↑ Server Name	Requests (2 min)	Monitor	Backup	DNS	DHCP		State	
Centos DHCP	4 requests	v0.23	-	-	IPv4: v0.22	- IPv6: v0.22	Running	¢
RHEL Backup	0 requests	-	-	2	-		Stopped	¢
RHEL8 6cDNS Server	0 requests	-		С.	5 <u>0</u>		Stopped	¢
Ubuntu DNS	4 requests	v0.23		v0.21	IPv4: v0.13	- IPv6: v0.13	Running	¢
		Displaying 1 to	o 4 of 4 items.					
		Previous	1 Next					

The Managed Servers tab allows for DNS, DHCP, and Backup VM servers to be created, installed, managed, and monitored from within ProVision.

Managed Servers is handled via two main areas of the GUI: the **Managed Servers** Tab and in Admin **Admin Settings - Managed Servers** subtab. Managed servers actions are also available via APIv2 and the command line.

Requirements

Note: Python3, Docker, and Rabbit MQ services are used to support and run Managed Servers. Local customers planning on utilizing managed servers will need to either install these services, or contact support@6connect.com for assistance to accommodate specific network needs.

Required installations are already included for hosted customers.

- Managed Servers
 - Overview
 - Features
 - Permissions
 - Administration
 - Managed Servers API
 - Managed Servers Flow Chart
 - Managed Servers Initial Setup and Installation
 - Install RabbitMQ
 - Enter Managed Servers Settings
 - 1) Enter Monitoring Settings
 - 2) Enter Managed Backup Settings
 - 3) Enter Update Settings
 - Update Settings Docker Update Mode (Internet)
 - Update Settings Image Download (Local)
 - 4) RabbitMQ Settings
 - Install Python 3 / Additional Services
 - Create Managed Servers
 - Working with Managed Servers
 - Additional Information

Overview

ProVision's Managed Servers tab allows for DNS (Knot-based), DHCP(Kea-based), and Backup VMs to be created, managed and monitored from within ProVision, either from the GUI or command line.

Features

Managed Server Features include:

- VM creation, management, and monitoring from within ProVision
- Minimal requirements Managed Servers only requires Python3 and Docker as dependencies
- OS options include CentOS and Ubuntu
- Supports both remote and manual setups
- Supports both root and non-root installations (for the latter, some root steps may be required)
- Installation and upgrade may be performed either via internet connection or locally:
 - ° Remote internet install/upgrade connects to 6connect's servers to get new versions
 - o Local installs/upgrades will require users to upload the new version locally into their ProVision instance
 - No kernel updates or other system updates are required for local updates only the Docker container will be updated.

Monitoring Features include:

- DNS Monitoring includes hit count for given period for each host/resource, total hit count, and last access time
- System monitoring Includes CPU load, Disk usage, connection information, swap memory, and virtual memory monitoring.
- Monitoring is available via GUI or API.
- Both "Push" and "Pull" options for statistics data are available
- Heartbeat service independent of Docker to provide monitoring data

Managed Backup Features include:

- Dedicated database and web server.
- Requests and stores the backups.
- Verifies backups checks for file corruption, data integrity, encryption/keys.
- Backups are encrypted and the backup server itself doesn't have the password/key to decrypt them.
- Customizable retention settings.
- Monitors its own disk space.
- Separate access management/users.

Permissions

Managed Servers uses a "6c servers" user group, under which automatic or manual "server users" are created. The "6c servers" user group gives permissions to all 6c server resources.

The "server users" are necessary to allow access and permissions for the managed server(s). When new server is created with an empty user (instead of a user-provided existing user) a random new user is automatically created under that group. Only that server user has permission to update the associated managed server.

Administration

The "Managed Servers" sub-tab under Admin Settings includes modules for global setup of Monitoring Settings, Update Settings, Monitoring Services, RabbitMQ settings, and Managed Backups Settings.

These settings provide the "default" values for servers created from the Managed Servers Tab.

Admin Settings Applicati	ion Authentication DNS DHCP	Peering Templates Scanning	& Discovery Managed Servers
Monitoring Settings	*	Update Settings	· · · · · · · · · · · · · · · · · · ·
Monitoring API settings - the server possible they will not be able to res	rs will make requests to this IP as it is highly olve ProVision's host	Update settings and how managed	servers are updated.
Default Monitoring API IP	2607:fae0:a000:6::200	API timeout:	300
Vorifie CA			Timeout for the API calls in miniseconds.
Verity CA	() OFF	Update Mode:	Docker Update (Internet)
Communication type	Push-based •		Specify the source for updates.
		Update username:	Isemane for the undates
Monitoring intervals settings			
Monitoring of the system:	ON O System load - CPU, RAM, etc.	DNS update token:	Token for DNS repository.
System monitoring interval:	5 System monitoring interval in seconds.	DHCF update token.	Token for DHCP repository.
Monitoring of the DNS		Monitoring update token	
monitoring of the Bito.	Request/response count, etc.		Token for Monitoring repository.
DNS monitoring interval:	60	Backup update token	
	DNS data will be collected and send (seconds).		Token for Monitoring repository.
Monitoring data rotantian			Test Tokens Save Changes

After making any settings updates, be sure to click "Save Changes" under the updated module.

For additional details, see Managed Servers Settings.

Managed Servers API

APIv2 endpoints are available for Managed Servers and Managed Server Monitoring.

To view these endpoints from your instance, go to the API Tab APIv2 Swagger Documentation, then click the Swagger links for the family categories "PVServers" and "Monitoring". For information on using the API, see APIv2.

Managed Servers Endpoints:

default	
GET	/pvservers Get pvserver resources
POST	/pvservers/addDHCPService Add DHCP Service to Managed Server
POST	/pvservers/addDNSService Add DNS Service to Managed Server
GET	/pvservers/config/{id} GET setup bundle for PVserver
GET	/pvservers/downloaddebug/{id} Download debug bundle from PVserver
GET	/pvservers/imgversions List the versions of given docker image
POST	/pvservers/manageService Manage service of PVserver
POST	/pvservers/pushAllZones Push all DNS zones to PVserver
POST	/pvservers/pushconfig Push configuration and install PVserver
POST	/pvservers/pushupdate Execute update on PVServer
POST	/pvservers/runDiagnostics Run diagnostics on PVserver
POST	/pvservers/testdockercred Test Docker Credentials

Monitoring Endpoints:



Managed Backups Endpoints:

defaul	t	
POST	/backups Create new backup	

Managed Servers Flow Chart



Managed Servers - Initial Setup and Installation

Install necessary packages (RabbitMQ, Python 3.6+, Docker), if not already installed with your ProVision instance.

Python 3.6+ and Docker should be installed on the target VM for the managed server in advance, prior to starting the managed server installation.

If an internet connection is established, Docker will be automatically installed by ProVision scripts.

Install RabbitMQ

If RabbitMQ has not already been installed on your ProVision instance / VM, you will need to install it by running the included shell script (Note: Installing RabbitMQ requires internet access.):

1) Run install_rabbitmq.sh.

/usr/bin/sh /path_to_provision/tools/install_rabbitmq_{OS}.sh

Ubuntu RabbitMQ installation script:

```
install rabbitmg ubuntu.sh
#!/usr/bin/sh
# https://www.rabbitmq.com/install-debian.html#apt-quick-start-cloudsmith
sudo apt-get install curl gnupg apt-transport-https -y
## Team RabbitMQ's main signing key
curl -lsLf "https://keys.openpgp.org/vks/vl/by-fingerprint/0A9AF2115F4687BD29803A206B73A36E6026DFCA" | sudo
gpg --dearmor | sudo tee /usr/share/keyrings/com.rabbitmq.team.gpg > /dev/null
## Launchpad PPA that provides modern Erlang releases
curl -lsLf "https://keyserver.ubuntu.com/pks/lookup?op=get&search=0xf77fleda57ebblcc" | sudo gpg --dearmor |
sudo tee /usr/share/keyrings/net.launchpad.ppa.rabbitmq.erlang.gpg > /dev/null
## PackageCloud RabbitMQ repository
curl -lsLf "https://packagecloud.io/rabbitmq/rabbitmq-server/gpgkey" | sudo gpg --dearmor | sudo tee /usr
/share/keyrings/io.packagecloud.rabbitmq.gpg > /dev/null
## Add apt repositories maintained by Team RabbitMQ
sudo tee /etc/apt/sources.list.d/rabbitmq.list <<EOF</pre>
## Provides modern Erlang/OTP releases
##
## "bionic" as distribution name should work for any reasonably recent Ubuntu or Debian release.
## See the release to distribution mapping table in RabbitMQ doc guides to learn more.
deb [signed-by=/usr/share/keyrings/net.launchpad.ppa.rabbitmq.erlang.gpg] http://ppa.launchpad.net/rabbitmq
/rabbitmq-erlang/ubuntu focal main
deb-src [signed-by=/usr/share/keyrings/net.launchpad.ppa.rabbitmq.erlang.gpg] http://ppa.launchpad.net
/rabbitmq/rabbitmq-erlang/ubuntu focal main
## Provides RabbitMQ
##
## "bionic" as distribution name should work for any reasonably recent Ubuntu or Debian release.
## See the release to distribution mapping table in RabbitMQ doc guides to learn more.
deb [signed-by=/usr/share/keyrings/io.packagecloud.rabbitmq.gpg] https://packagecloud.io/rabbitmq/rabbitmq-
server/ubuntu/ focal main
deb-src [signed-by=/usr/share/keyrings/io.packagecloud.rabbitmq.gpg] https://packagecloud.io/rabbitmq
/rabbitmq-server/ubuntu/ focal main
EOF
## Update package indices
sudo apt-get update -y
## Install Erlang packages
sudo apt-get install -y erlang-base \setminus
                                erlang-asn1 erlang-crypto erlang-eldap erlang-ftp erlang-inets \
                                                        erlang-mnesia erlang-os-mon erlang-parsetools erlang-
public-key \
                                                                                 erlang-runtime-tools erlang-
snmp erlang-ssl \
erlang-syntax-tools erlang-tftp erlang-tools erlang-xmerl
## Install rabbitmq-server and its dependencies
sudo apt-get install rabbitmq-server -y --fix-missing
sudo rabbitmq-plugins enable rabbitmq_management
sudo rabbitmq-server start -detached
sleep 5
sudo rabbitmqctl add_user pv_rabbitmq S3curePassW0rd!
sudo rabbitmqctl set_user_tags pv_rabbitmq administrator
sudo rabbitmqctl set_permissions -p / pv_rabbitmq ".*" ".*"
sudo systemctl enable rabbitmq-server
echo "RabbitMQ user is: pv_rabbitmq"
echo "RabbitMQ password is: S3curePassW0rd!"
echo "RabbitMQ port is: 5672"
echo "RabbitMQ management port is: 15672"
```

CentOS RabbitMQ installation script:

install_rabbitmq_centos.sh #!/usr/bin/sh # https://www.rabbitmg.com/install-debian.html#apt-quick-start-cloudsmith sudo yum install curl gnupg apt-transport-https -y ## Team RabbitMQ's main signing key curl -lsLf "https://keys.openpgp.org/vks/v1/by-fingerprint/0A9AF2115F4687BD29803A206B73A36E6026DFCA" | sudo gpg --dearmor | sudo tee /usr/share/keyrings/com.rabbitmq.team.gpg > /dev/null ## Launchpad PPA that provides modern Erlang releases curl -lsLf "https://keyserver.ubuntu.com/pks/lookup?op=get&search=0xf77fleda57ebblcc" | sudo gpg --dearmor | sudo tee /usr/share/keyrings/net.launchpad.ppa.rabbitmq.erlang.gpg > /dev/null ## PackageCloud RabbitMQ repository curl -lsLf "https://packagecloud.io/rabbitmq/rabbitmq-server/gpgkey" | sudo gpg --dearmor | sudo tee /usr /share/keyrings/io.packagecloud.rabbitmq.gpg > /dev/null ## Add apt repositories maintained by Team RabbitMQ #sudo tee /etc/apt/sources.list.d/rabbitmg.list <<EOF</pre> ## Provides modern Erlang/OTP releases ## ## "bionic" as distribution name should work for any reasonably recent Ubuntu or Debian release. ## See the release to distribution mapping table in RabbitMQ doc guides to learn more. #deb [signed-by=/usr/share/keyrings/net.launchpad.ppa.rabbitmq.erlang.gpg] http://ppa.launchpad.net/rabbitmq /rabbitmq-erlang/ubuntu focal main #deb-src [signed-by=/usr/share/keyrings/net.launchpad.ppa.rabbitmq.erlang.gpg] http://ppa.launchpad.net /rabbitmq/rabbitmq-erlang/ubuntu focal main ## Provides RabbitMO ## ## "bionic" as distribution name should work for any reasonably recent Ubuntu or Debian release. ## See the release to distribution mapping table in RabbitMQ doc guides to learn more. #deb [signed-by=/usr/share/keyrings/io.packagecloud.rabbitmq.gpg] https://packagecloud.io/rabbitmq/rabbitmqserver/ubuntu/ focal main #deb-src [signed-by=/usr/share/keyrings/io.packagecloud.rabbitmq.gpg] https://packagecloud.io/rabbitmq /rabbitmg-server/ubuntu/ focal main #EOF ## Update package indices sudo yum update -y ## Install Erlang packages sudo yum install -y erlang-base erlang-asn1 erlang-crypto erlang-eldap erlang-ftp erlang-inets erlang-mnesia erlang-os-mon erlang-parsetools erlang-public-key erlang-runtime-tools erlang-snmp erlang-ssl erlang-syntaxtools erlang-tftp erlang-tools erlang-xmerl ## Install rabbitmq-server and its dependencies sudo yum install rabbitmq-server -y sudo rabbitmg-plugins enable rabbitmg management sudo rabbitmg-server start -detached sleep 5 sudo rabbitmqctl add_user pv_rabbitmq S3curePassW0rd! sudo rabbitmqctl set_user_tags pv_rabbitmq administrator sudo rabbitmqctl set_permissions -p / pv_rabbitmq ".*" ".*" sudo systemctl enable rabbitmq-server echo "RabbitMQ user is: pv_rabbitmq" echo "RabbitMQ password is: S3curePassW0rd!" echo "RabbitMQ port is: 5672" echo "RabbitMQ management port is: 15672"

This script is taken from Installing on Debian and Ubuntu — RabbitMQ and may need an update in future.

2) You may verify the installation is working by opening http://your.host:15672

3) Once confirmed working, go to ProVision Admin Settings Managed Server Settings. Then, enter the RabbitMQ IP, port, username, and password into the RabbitMQ Settings section. When done, click "Save Changes".

Enter Managed Servers Settings

In ProVision, navigate to Admin Settings Managed Servers and enter or verify settings in the following areas:

1) Enter Monitoring Settings

Under Monitoring Settings, select the "Default Monitoring API IP", and enter any additional setting options as follows:

Monitoring Settings	~
Monitoring API settings - the serve possible they will not be able to res	rs will make requests to this IP as it is highly solve ProVision's host
Default Monitoring API IP	•
Verify CA	OFF
Communication type	Push-based -
Monitoring intervals settings	
Monitoring of the system:	System load - CPU, RAM, etc.
System monitoring interval:	5
	System monitoring interval in seconds.
Monitoring of the DNS:	OFF Request/response count, etc.
DNS monitoring interval:	ex: 60
	DNS data will be collected and send (seconds).
Monitoring data retention	
Raw data retention:	1440
	The period in minutes to keep the raw/detailed monitoring data.
Aggregated data retention:	30
	The period in days to keep the aggregated monitoring data.
	Save Changes

• Default Monitoring API IP: The default IP to be used for API calls (heartbeat, activation, etc.) for managed server monitoring and to send API requests, as multiple IPs may be available depending on the Provision set up.

 Note: The Default Monitoring API IP can be set to a default value in this area, but it may also be set individually per managed server as an override.

- Verify CA: Toggle to "on" if you wish to enable the CA check.
- Communication type: Push (when the server pushes data to ProVision) and pull (when ProVision connects to the server and pulls the data). Push is preferred, but Pull-based may be desirable in cases where a firewall would not allow inbound connections to ProVision.
- Monitoring of the system: Toggle to "on" to enable monitoring of system statistics.
- System monitoring interval: How often (in seconds) the system will be checked. The smaller the interval, the more data there will be to store.
- Monitoring of the DNS: Toggle to "on" to enable gathering statistics from the DNS server.
- DNS monitoring interval: How often (in seconds) the DNS server will be monitored. Data will be aggregated for the set period of time and the aggregated results will be sent.
- Raw data retention: The number of minutes to retain raw data. By default, this is set to 1440 minutes (1 day). Raw data requires sizable storage needs.
- Aggregated data retention: The number of days to retain aggregated data.

2) Enter Managed Backup Settings

If in use, enter the password for Managed Backup Servers and click "save changes".

Managed Backup Settings		~
Managed backup settings		
Password	The password is hidden.	
		Save Changes

3) Enter Update Settings

Enter Update settings to use for Managed Servers, starting with API timeout (in milliseconds) and Update Mode.

Fill in the remaining update settings as applicable to the selected update mode:

- Internet Docker Update (Recommended) This uses a Docker update directly from our docker registry. Internet connection is required for the managed server in order this mode to work.
 - When this is chosen, fields to enter tokens for the DNS / DHCP repository will displayed.

For assistance obtaining repository tokens, please contact support@6connect.com.

Update Settings - Docker Update Mode (Internet)

Update Settings	~
Update settings and how managed	d servers are updated.
API timeout:	300
	Timeout for the API calls in milliseconds.
Update Mode:	Docker Update (Internet)
	Specify the source for updates.
Update username:	readonly
	Username for the updates.
DNS update token:	••••••
	Token for DNS repository.
DHCP update token:	•••••
	Token for DHCP repository.
Monitoring update token	••••••
	Token for Monitoring repository.
Backup update token	
	Token for Monitoring repository.
	Test Tokens Save Changes

API timeout: Timeout for API calls in milliseconds.

Update Mode: Enter the source for updates, depending on internet accessibility. Two options are available:

• Image Download (Local) - This option is to update from ProVision, locally. No internet connection is required from the managed server, but an image file must be manually downloaded from a separate secure site and then uploaded into ProVision.

If "Image Download (Local)" is selected, you must download the image file from a secure 6connect site to then upload for this setting. Please contact support@6connect.com for credentials to access the image download area.

- Internet Docker Update This uses a Docker update directly from our docker registry. Internet connection is required for the managed server in order this mode to work. When this is chosen fields for tokens for our gitlab/registry appear to be filled in:
 - **Update username:** Enter the username for updates.
 - **DNS update token:** Enter the token for the DNS repository.
 - DHCP update token: Enter the token for the DHCP repository.
 - Monitoring update token: Enter the token for the Monitoring repository.
 - **Backup update token**: Enter the token for the Backup Monitoring repository.
- Test Tokens: Click to test the applied tokens for DNS, DHCP, and Monitor.

For assistance obtaining repository tokens, please contact support@6connect.com.

• ProVision Image Download - If Update Mode is set to "Image Download (Local)", then no internet connection is required from the managed server, but an image file must be manually downloaded from a separate secure site to then upload into ProVision.

Update Settings - Image Download (Local)

If Update Mode is set to "Image Download (Local)", the following settings will display to enter:

hanually.
thoose File No filosen Upload
s are updated.
0
out for the API calls in milliseconds.
age Download (Local)
ify the source for updates.
Test Tokens Save Changes

Image file: Attach the Image file for upload to ProVision.

If "Image Download (Local)" is selected, you must download the image file from a secure 6connect site to then upload for this setting. Please contact support@6connect.com for credentials to access the image download area.

API timeout: Timeout for API calls in milliseconds.

Update Mode: Enter the source for updates - Image Download (Local), or opt to change to Docker Update (Requires Internet).

Images history: Available only after saving changes. Once a history is available, it will display at the top of the module.

Update Settings	~
Images history	
pv_monitor-v0.21.tar	
pv_monitor-v0.20.tar	
pv monitor-v0.19.tar	
pv_monitor-v0.18.tar	
Upload the new version of the image	ge file manually.
Image file:	Choose File No filosen Upload
Update settings and how managed	servers are updated.
API timeout:	300
	Timeout for the API calls in milliseconds.
Update Mode:	Image Download (Local)
	Specify the source for updates.
	Test Tokens Save Changes

Test Tokens: Click to test the applied tokens for DNS, DHCP, and Monitor.

4) RabbitMQ Settings

Enter the RabbitMQ IP, port, username, and password into the RabbitMQ Settings section. To verify the settings, click "Test Credentials".

RabbitMQ Settings	~
RabbitMQ settings. It is used for con	nmunication between nodes and ProVision
RabbitMQ IP	
RabbitMQ Port	
Username	
Password	
	lest Credentials Save Changes

When done, click "Save Changes".

Install Python 3 / Additional Services

Setup the necessary services:

Run MQ workers	
<pre>python3 /path_to_provision/pvservers/setup/pv_install/setup_services.py</pre>	

Create Managed Servers

At this point, initial setup is complete and you may begin to create new managed servers in ProVision.

For instructions on creating a new managed server, see Working with Managed Servers.

Working with Managed Servers

For information on creating a new managed server, working with the managed server list, and viewing monitoring, see the following sections:

• Working with Managed Servers

Additional Information

For Admin-level managed server setting information, see:

Managed Servers Settings