System Shell Tools

In this article

This section describes **shell tools** required for the deployment management of the system. All these tools are part of the JeraSoft Billing distributive. In order to use the tools you will need either SSH or direct access to the server console. Some of the tools require *root permissions* to run.

Important!

Please use these tools **only if you have a clear understanding** of what you are doing. Misuse of the tools may cause improper functioning of the system.

Attention

Please note, for simplification we introduced **<APP_PATH>** variable that refers to the JeraSoft Billing application files location. This location may differ, but typically it is **"/opt/jerasoft/vcs"**. So whenever you see an example with path like *"<APP_PATH>/bin/system/cluster"* it means *"/opt/jerasoft/vcs/bin/system/cluster"*.

Requirements Checker

The tool is aimed to check **minimal requirements** of your server before installation.

Usage

```
<APP_PATH>/bin/system/setup-checker
```

The tool requires ***root*** permissions and takes no arguments. It should be executed before installation of the system in order to check minimal hardware and software requirements.

Attention

The tool checks only minimal requirements. **Real hardware requirements** highly depend on your traffic and deployment model.

Safety Checker

The tool is aimed to check configuration of the main server settings after installation.

Usage

```
<APP_PATH>/bin/system/security-checks
```

The tool takes no arguments. When executed it performs numerous checks for the correctness of the network and server configuration.

Services Manager

The tool is used for **management of System Services**. It allows to correctly start, stop and perform other actions over various JeraSoft Billing Services such as RADIUS Server, SIP Server, Calculator, etc.

Usage

```
<APP_PATH>/bin/system/service <COMMAND> [<service-name>] [<options>]
```

The tool should be run under ***root*** or ***vcs*** user. The tool typically takes 2 arguments – action to perform and related system service. Actions prefixed with *"all-*"* do not require service name and operate over all services.

Command	Description
`start`	Start System Service Takes *"--wait"* option in order to wait and exit only when service finishes its execution.
`stop`	Stop System Service
`restart`	Stop and then start System Service
`reload`	Send reload (HUP) signal to the System Service (forces reload of settings, connections, etc)
`status`	Show current status of the System Service
`all-start`	Start all required System Services (list of services varies depending on the role of the current node in the cluster)
`all-stop`	Stop all running System Services
`all-status`	Show status of all System Services on the current node

Usage Examples

Restart RADIUS Server

```
<APP_PATH>/bin/system/service restart bbradiusd
```

Start Files Downloader

```
<APP_PATH>/bin/system/service start files_downloader
```

Start all required System Services

```
<APP_PATH>/bin/system/service all-start
```

Cluster Manager

The tool is used to **manage nodes in the cluster deployment**. It allows to initialize the cluster, add a new node, promote redundancy to master, etc.

Usage

```
<APP_PATH>/bin/system/cluster <COMMAND> [<options>]
```

The tool requires ***root*** permissions. The list of arguments and other requirements depend on the command used. Please refer to the below table for a summary and respective sections for details.

Command	Description	Nodes	Root Required
`status`	Show status of the cluster	Any node	No
`init-master`	Init Master Node configuration	Master	Yes
`init-slave`	Init Slave Node configuration	Master	Yes
`promote`	Promote current node to Master	Redundancy	Yes
`sync-files`	Sync files from Master	Redundancy, Reporting, Processing	No
`remove-node`	Remove Node from the Cluster	Master	Yes

Cluster Status

The command shows Cluster Status, including all nodes with their roles, IP addresses, current lag to Master, and overall status.

Bash

```
<APP_PATH>/bin/system/cluster status
```

The command can be executed **on the Master** to get the most detailed information about the cluster:

Role	Node ID	IP Address	Status	Receive Lag	Replay Lag	Replication
> Master	node-01	172.17.172.17	[Current]	N/A	N/A	Master
Redundancy	node-02	172.17.172.101	Connected	0.00 MB	0.00 MB	Streaming
Processing	node-03	172.17.172.102	Connected	0.00 MB	0.00 MB	Logical

Alternatively, the command can be executed at any other node - in this case, only the status of the connection between this particular node and the Master will be shown.

If any node failed and has been disconnected from the cluster, it will be shown like this:

Role	Node ID	IP Address	Status	Receive Lag	Replay Lag	Replication
> Master	node-01	172.17.172.17	[Current]	N/A	N/A	Master
Redundancy	node-02	172.17.172.101	Disconnected	Unknown	Unknown	Streaming
Processing	node-03	172.17.172.102	Connected	0.00 MB	0.00 MB	Logical

In this case, you have to re-check failed node, fix it and then return to the cluster using the *"init-slave"* command.

Init Master

The command is used for the initial configuration of the Master Node.

Bash

```
<APP_PATH>/bin/system/cluster init-master <IP-ADDRESS> [<options>]
```

Command has to be executed **on the Master** node and requires **root** permissions. The following options are supported:

Option	Description	Default
`<IP-ADDRESS>`	IP Address of the Master server **(required)**	
`--ssh-port=<port>`	SSH Port as the master node	`22`
`--pg-data=<path>`	Path to PostgreSQL data directory	`autodetect`

Init Slave

The command is used to add a node to the cluster. There are different contexts when it is required:

- First-time deployment of the cluster
- Addition of a new slave node to the cluster
- Addition of the old master to work as a slave after failover

Bash

```
<APP_PATH>/bin/system/cluster init-slave <IP-ADDRESS> [<options>]
```

Command has to be executed **on the Master** node and requires **root** permissions. The following options are supported:

Option	Description	Default
`<IP-ADDRESS>`	IP Address of the Slave server **(required)**	
`--role=<role>`	Role of the new node: • **redundancy** - fully-featured redundancy, that acts as a hot standby and can be promoted to the Master at any time (may be used for redundancy and load balancing at the same time) • **reporting** - a node that receives most of the requests for the reports, holds a full snapshot of the database, however, it might be delayed from Master depending on the current load and requests (might be used for failover, as a last resort) • **processing** - lightweight node for processing of real-time requests (authentication, authorization, and routing), can not be used for failover as it does not hold any statistical data • **calculation** - a node to help calculate huge amounts if the Master struggles to process solely	`redundancy`
`--ssh-port=<port>`	SSH Port at the remote node	`22`
`--ssh-user=<user>`	SSH User at the remote node	`jerasupport`
`--pg-data=<path>`	Path to PostgreSQL data directory at the remove node	`autodetect`

Promote to Master

The command is used to **promote the Redundancy node to Master**.

Bash

```
<APP_PATH>/bin/system/cluster promote
```

Command has to be executed **on the Redundancy node** and requires **root** permissions. Reporting node can be used as a last resort if there are no Redundancy node alive. There are no options required.

After the promotion is performed, all required System Services will be started on the current node (new master). After you fix the old *Master,* you may add it as a new *Slave* using the *"init-slave"* command.

> ⚠ **Attention**
>
> In case when you have more than 2 nodes in the cluster, you need to **re-init all other** nodes from this new *Master*.

Sync Files

The command is used to sync data and application files from the Master.

Bash

```
<APP_PATH>/bin/system/cluster sync-files
```

Command has to be executed **on the Redundancy node** and by default, it is added to the crontab for automatic synchronization.

Remove Node

The command is used to remove a node from the cluster.

Bash

```
<APP_PATH>/bin/system/cluster remove-node <IP-ADDRESS>
```

Command has to be executed **on the Master** node. The node in question shouldn't have any active database replication. The following options are supported:

Option	Description	Default
`<IP-ADDRESS>`	IP Address of the remote node **(required)**	