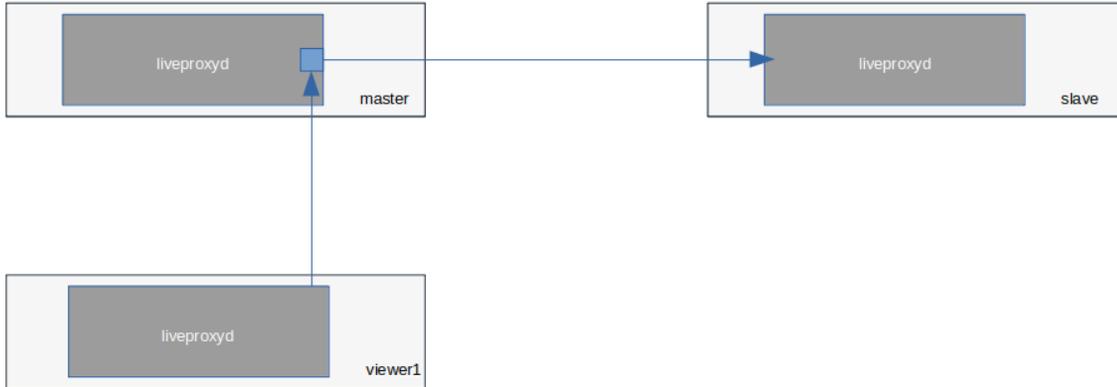


# Cascading Livestatus

1. It is now possible to cascade livestatus proxy configurations. This comes in handy to build cascaded distributed GUI (status GUI) setups.

The feature has been build for a scenario like this:

- A distributed setup where you have remote sites that are not directly reachable
- These remote sites are only reachable through a single "master" site
- You use the "masters" for configuration of all the related sites
- The central site is only used as central operating site (overview, reporting) and not for configuration
- The viewer site should be an independent checkmk site (can be on the same machine)



## Step-by-step guide

Scenario: We have a slave which is monitoring some hosts. But this slave site is not directly reachable. Therefore, we need a so-called viewer Site. This viewer Site will get all data for our slave Site via the master Site

1. Create a distributed Setup by following our [manual](#)
2. Create one or more viewer sites. For this example I will create viewer1 and viewer2
3. **On the master:**
  - a. Go to the Master Site WATO - CONFIGURATION and edit the connection slave
  - b. Go to 'Use Livestatus Proxy Daemon' and enable 'Allow access via TCP'.
    - TCP port: Type in one free TCP port. In my example I use 6560
    - Restrict access to IP addresses: If you want to restrict, type in the addresses
    - Encrypt communication: Is not working at the moment. We have created an internal Ticket for that

i

## Edit site connection slave

---

Alias .....

**▼ STATUS CONNECTION**

**Connection** .....

Connect via TCP (IPv4) ▼

TCP address to connect to

Host:  Port:

**Use Livestatus Proxy Daemon** .....

Use Livestatus Proxy Daemon ▼

**Parameters** .....

Use global connection parameters ▼

Use the global parameters for this connection

Allow access via TCP

TCP port

Restrict access to IP addresses

Encrypt communication

**Connect timeout** .....

Seconds

**Persistent Connection** .....

Use persistent connections

**URL prefix** .....

**Status host** .....

No status host ▼

**Disable in status GUI** .....

Temporarily disable this connection

4. Create a new viewer [Site](#)
5. Configuration of cascading Livestatus on Site viewer1

+ New connection

- Go to *WATO - CONFIGURATION Distributed Monitoring*
  - BASIC SETTINGS

!
Site ID should be the name of the master site where the slave is connected

i

## Edit site connection master

---

No changes
Main Menu
All Sites
Status encryption

**▼ BASIC SETTINGS**

**Site ID** .....

master

**Alias** .....

- STATUS CONNECTION

Use here the Port of Step 3 (Allow access via TCP Setting of Site Slave configured on Master Site)

Encryption should be 'Plain text'

If you want to connect a second viewer, you need to enable 'Allow access via TCP' again! Please use a different port  
This only optional

- Leave the remaining options at their defaults
6. Configuration of cascading Livestatus on Site viewer2



- Go to *WATO - CONFIGURATION Distributed Monitoring*
- BASIC SETTINGS

- STATUS CONNECTION

**▼ STATUS CONNECTION**

**Connection** ..... Connect via TCP (IPv4) ▼  
TCP address to connect to  
Host: 192.168.2.161 Port: 6561

**Encryption**  
Plain text (Unencrypted) ▼  
Use plain text, unencrypted transport

**Use Livestatus Proxy Daemon** ..... Use Livestatus Proxy Daemon ▼

**Parameters**  
Use global connection parameters ▼  
Use the [global parameters](#) for this connection

Allow access via TCP

**Connect timeout** ..... 2 Seconds

Use persistent connections

**Persistent Connection** .....

**URL prefix** .....

**Status host** ..... No status host ▼

Disable in status GUI .....  Temporarily disable this connection

**► CONFIGURATION CONNECTION**

Use here the Port we configured in Step 5 (Allow access via TCP Setting configured on viewer1)

Encryption should be 'Plain text'

If you want to connect a second viewer, you need to enable 'Allow access via TCP' again! Please use a different port  
This only optional

Summary: Now you're able to monitor all hosts from Site 'slave' via the Site 'master'!

 Related Werk to this topic: [#5970](#)

## Related articles

- [Livestatus Performance](#)
- [Livestatus queries](#)
- [Cascading Livestatus](#)
- [Livestatus connection in state Unknown](#)
- [cmk.cee.liveproxy.Client.ClientRequestTimeoutException](#)