Tungsten Dashboard 1.0 Manual

Continuent Ltd

Tungsten Dashboard 1.0 Manual

Continuent Ltd

Abstract

This manual documents Tungsten Dashboard, a simple graphical user interface allowing you to manage all of your Tungsten Clusters in one single place.

This manual includes information for 1.0, up to and including 1.0.17.

Build date: 2026-03-16 (ac746a88)

Up to date builds of this document: Tungsten Dashboard 1.0 Manual (Online), Tungsten Dashboard 1.0 Manual (PDF)

Table of Contents

List of Figures

Chapter 1. Overview

A simple GUI management tool for Tungsten Cluster v5.3.x and above.

> **Important**
>
> Read this entire document before attempting installation.

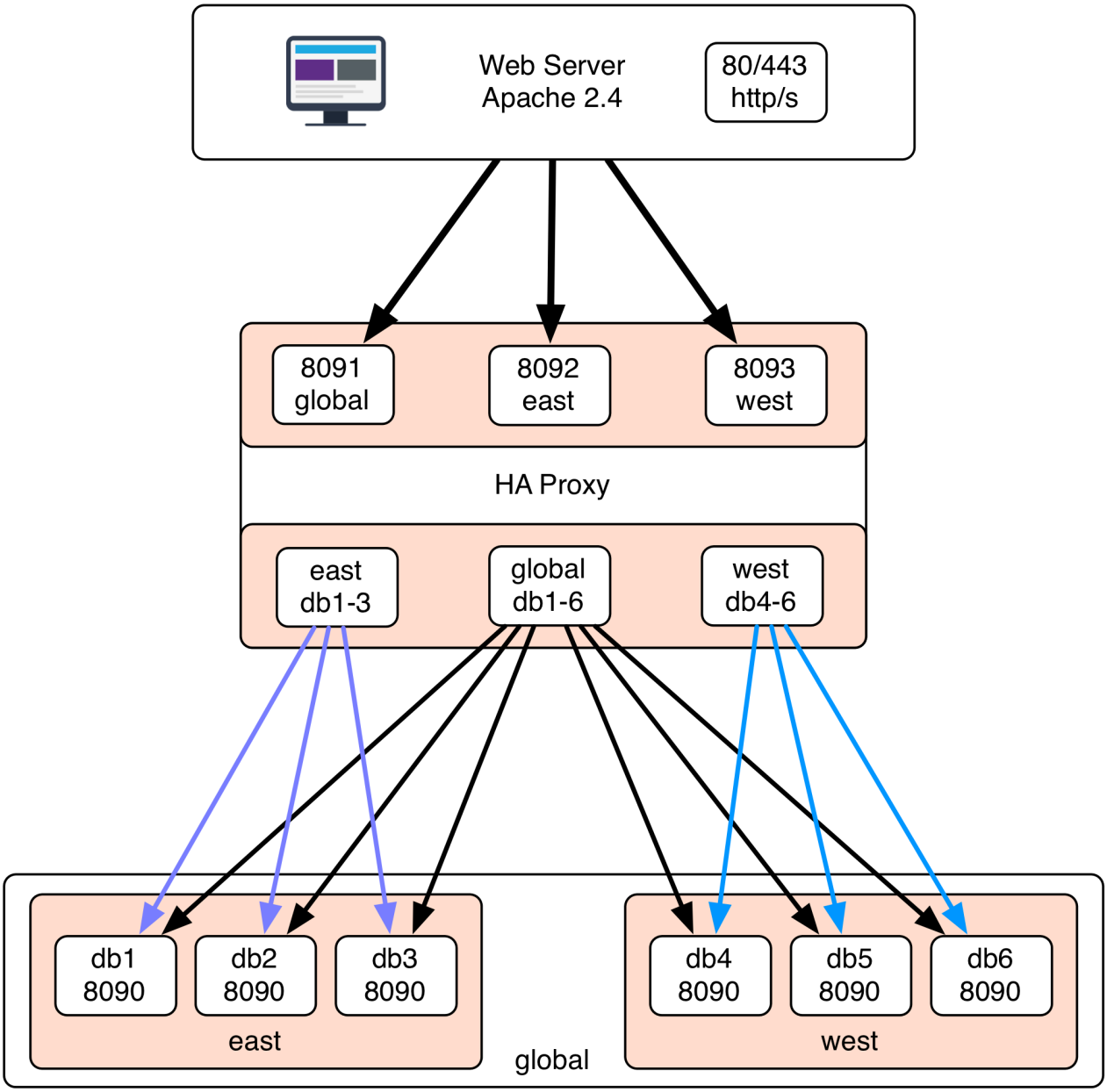
This application was written using PHP, jQuery and Bootstrap and uses HA Proxy to distribute API requests. The Apache 2 web server was used for the examples in this document.

The Dashboard is usually installed on a standalone web server with HA Proxy installed.

HA Proxy routes API requests to the various database nodes running the manager API listener on port 8090. There is one frontend per cluster. Each backend contains all db nodes for that cluster.

The architecture diagram below displays how things would look when using a 6-node Composite cluster named `global`, with two member clusters, named `east` and `west`.

Figure 1.1. Tungsten Dashboard Architecture



Chapter 2. Prerequisites

Continuent Tungsten Dashboard needs the following prerequisites to function:

- Continuent Tungsten Clustering v5.3.0 and above or v6.0.1 and above *only*.
- Web server with PHP7 support [sample configs provided for Apache 2.2 and 2.4]

  Please note that more recent versions of Lunux and Apache use php-fpm to run PHP code.

  - Below is a sample command line session on Amazon Linux 2 to get PHP 7.4 installed with all of the needed dependencies:

    ```
    shell> sudo -i
    root shell> amazon-linux-extras enable php7.4
    root shell> yum clean metadata
    root shell> yum install php php-common php-cli php-pdo php-fpm php-json php-mysqlnd php-mbstring php-xml
    root shell> systemctl start php-fpm
    root shell> systemctl enable php-fpm
    ```

  - Next, tweak the default Apache PHP configuration, otherwise PHP-FPM gets called for files that don't exist. So, edit the file:

    ```
    shell> sudo -i
    root shell> vi /etc/httpd/conf.d/php.conf
    ```

  - Change the SetHandler around line 27 from OLD to NEW:

    ```
    OLD:
    <FilesMatch \.php$>
    SetHandler "proxy:unix:/run/php-fpm/www.sock|fcgi://localhost"
    </FilesMatch>
    ```

    ```
    NEW:
    <FilesMatch \.php$>
    <If "-f %{SCRIPT_FILENAME}">
      SetHandler "proxy:unix:/run/php-fpm/www.sock|fcgi://localhost"
    </If>
    </FilesMatch>
    ```

  - Then check and restart Apache:

    ```
    shell> sudo apachectl configtest
    shell> sudo systemctl restart httpd
    ```

- PHP Curl Support - install `php-common` or `php-curl` and restart your web server
- HA Proxy - Chapter 9, *Install and Configure HA Proxy*.
- Make sure to open ALL of the appropriate firewall ports to ensure access.
  - The default architecture would require TCP port 8090 open between the web server running the Dashboard and all cluster nodes in all clusters that are to be administered by the GUI application.
  - By default, port 80 will need to be open from the client browser to the web server running the Tingsten Dashboard. If HTTPS has been implemented, TCP 443 must be opened in addition to port 80.

Chapter 3. Security Limitations

Continuent Tungsten Dashboard has the following security limitations:

Warning

THERE IS NO API SECURITY in versions less than v7.0.0 - If you enable the API on the Manager, anyone may connect to it. Use your firewall to block port 8090 from non-essential hosts.

Warning

SSL (https) is not supported on the Manager API endpoint in versions less than v7.0.0

Warning

Please use Apache Basic Auth to lock down access to the Tungsten Dashboard GUI.

Warning

SSL (https) configuration for the Tungsten Dashboard is possible, but is beyond the scope of this document.

Warning

Locking only works on a single web server host, so if you have installed the Tungsten Dashboard on more than one host, the lock is not shared and is therefore ineffective.

Chapter 4. Configure the Tungsten Cluster Manager API

The configuration will vary based upon the version of Tungsten software you are running. For versions less than 7.0.0, you will need to configure APIv1. For Tungsten v7.0.0 and higher, please configure for APIv2.

4.1. Configure the Tungsten Cluster APIv1 for Dashboard

Configure the Tungsten Cluster APIv1 for Dashboard - Needed for Tungsten software versions LESS THAN v7.0.0

Add the following to `/etc/tungsten/tungsten.ini` under the `[defaults]` section:

```
mgr-api-port=8090
mgr-api=true
mgr-api-address=0.0.0.0
mgr-api-full-access=true
```

> **Warning**
>
> Either SHUN the individual nodes one at a time and WELCOME each one after running the update, or set policy to *MAINTENANCE* via cctrl and update all nodes, then set the policy back to *AUTOMATIC* when all nodes have been completed.

Inform the running manager of the changed configuration:

```
shell> tpm update
```

> **Important**
>
> You may need to restart the manager.

Verify that the port is listening:

```
shell> sudo netstat -pan | grep 8090
```

4.2. Configure the Tungsten Cluster APIv2 for Dashboard

Configure the Tungsten Cluster APIv2 for Dashboard - Needed for Tungsten software versions v7.0.0 and higher

Version 1.0.10. This feature was first introduced in Tungsten Dashboard version 1.0.10-125

4.2.1. Remove the Existing Tungsten Cluster APIv1 Configuration

> **Important**
>
> Do these steps only if you have already enabled APIv1 and are upgrading to v7.0.0 or higher.

Remove the following APIv1-specific options from `/etc/tungsten/tungsten.ini` under the `[defaults]` section:

```
mgr-api-port=8090
mgr-api=true
mgr-api-address=0.0.0.0
mgr-api-full-access=true
```

4.2.2. Configure the Tungsten Cluster APIv2 INI Entries

> **Important**
>
> Perform these steps BEFORE installing or upgrading to v7 from a version less than 7.

For NON-secure deployments, check for the following in `/etc/tungsten/tungsten.ini` under the `[defaults]` section:

```
disable-security-controls=true
enable-connector-ssl=false
datasource-enable-ssl=false
```

If you located the above options, then your deployment is NON-Secure. ADD the following to `/etc/tungsten/tungsten.ini` under the `[defaults]` section:

```
rest-api-admin-user={desiredLoginHere}
```

```
rest-api-admin-pass={desiredPasswordHere}
connector-rest-api-address=0.0.0.0
connector-rest-api-authentication=true
connector-rest-api-port=8096
connector-rest-api-ssl=false
manager-rest-api-address=0.0.0.0
manager-rest-api-authentication=true
manager-rest-api-full-access=true
manager-rest-api-port=8090
manager-rest-api-ssl=false
replicator-rest-api-address=0.0.0.0
replicator-rest-api-authentication=true
replicator-rest-api-port=8097
replicator-rest-api-ssl=false
```

For Secure deployments, check for the following in `/etc/tungsten/tungsten.ini` under the `[defaults]` section:

```
disable-security-controls=false
enable-connector-ssl=true
datasource-enable-ssl=true
```

If you located the above options, then your deployment is SECURE. ADD the following to `/etc/tungsten/tungsten.ini` under the `[defaults]` section:

```
rest-api-admin-user={desiredLoginHere}
rest-api-admin-pass={desiredPasswordHere}
connector-rest-api-address=0.0.0.0
connector-rest-api-authentication=true
connector-rest-api-port=8096
connector-rest-api-ssl=true
manager-rest-api-address=0.0.0.0
manager-rest-api-authentication=true
manager-rest-api-full-access=true
manager-rest-api-port=8090
manager-rest-api-ssl=true
replicator-rest-api-address=0.0.0.0
replicator-rest-api-authentication=true
replicator-rest-api-port=8097
replicator-rest-api-ssl=true
```

Warning

Either SHUN the individual nodes one at a time and WELCOME each one after running the update, or set policy to *MAINTENANCE* via cctrl and update all nodes, then set the policy back to *AUTOMATIC* when all nodes have been completed.

Inform the running manager of the changed configuration:

```
shell> tpm update
```

Important

You may need to restart the Manager, Connector and Replicator.

Verify that the three ports (Manager, Connector and Replicator) are listening:

```
shell> sudo netstat -pan | grep 8090
shell> sudo netstat -pan | grep 8096
shell> sudo netstat -pan | grep 8097
```

Chapter 5. Test Connectivity to the Tungsten Manager API Directly

Test connectivity to the Tungsten Manager API directly using curl:

```
shell> curl -s http://db1:8090/manager/status/east/
shell> curl -s http://db4:8090/manager/status/west/
shell> curl -s -X POST http://db4:8090/manager/control/west/heartbeat
```

5.1. Test Connectivity to the Tungsten APIv2 Directly

Test Connectivity to the Tungsten APIv2 Directly

Version 1.0.10. This feature was first introduced in Tungsten Dashboard version 1.0.10-125

Test connectivity to the Tungsten APIv2 directly using curl for Secure deployments:

```
shell> /usr/bin/curl --user tungsten:demo -k --request GET 'https://db1:8090/api/v2/manager/status'

shell> /usr/bin/curl --user tungsten:demo -k --request POST 'https://db4:8090/api/v2/manager/control/service/south/heartbeat'
```

Test connectivity to the Tungsten APIv2 directly for Secure deployments using the tapi command on a Tungsten node:

> **Important**
>
> Please note that the tapi command exists only on Tungsten database nodes, because it is included with the Tungsten Clustering software, not the Dashboard.

```
shell> tapi -M -r status
{
 "managerPID" : 3234,
 "dataSourceName" : "db10-demo.continuent.com",
 "parentPID" : 3213,
 "dataServiceName" : "south",
 "isCoordinator" : true,
 "state" : "ONLINE",
 "uptimeSeconds" : 3428,
 "isWitness" : false,
 "policyMode" : "AUTOMATIC",
 "coordinator" : "db10-demo.continuent.com",
 "startTime" : "2022-03-14T20:41:41.427 UTC"
}

shell> tapi -M -r heartbeat
{
 "taskId" : "1e52cc32-ada5-4546-beea-32eb26846629",
 "operation" : "HeartbeatTask",
 "state" : "in_progress"
}
```

Test connectivity to the Tungsten APIv2 directly using curl for NON-Secure deployments:

```
shell> /usr/bin/curl --user tungsten:demo --request GET 'http://db1:8090/api/v2/manager/status'

shell> /usr/bin/curl --user tungsten:demo --request POST 'http://db4:8090/api/v2/manager/control/service/south/heartbeat'
```

Test connectivity to the Tungsten APIv2 directly for NON-Secure deployments using the tapi command on a Tungsten node:

> **Important**
>
> Please note that the tapi command exists only on Tungsten database nodes, because it is included with the Tungsten Clustering software, not the Dashboard.

```
shell> tapi --http -M -r status
WARN: Using insecure Non-SSL http connection instead of the default https
{
 "managerPID" : 3234,
 "dataSourceName" : "db10-demo.continuent.com",
 "parentPID" : 3213,
 "dataServiceName" : "south",
```

```
 "isCoordinator" : true,
 "state" : "ONLINE",
 "uptimeSeconds" : 3428,
 "isWitness" : false,
 "policyMode" : "AUTOMATIC",
 "coordinator" : "db10-demo.continuent.com",
 "startTime" : "2022-03-14T20:41:41.427 UTC"
}

shell> tapi --http -M -r heartbeat
WARN: Using insecure Non-SSL http connection instead of the default https
{
 "taskId" : "1e52cc32-ada5-4546-beea-32eb26846629",
 "operation" : "HeartbeatTask",
 "state" : "in_progress"
}
```

Chapter 6. Install the Tungsten Dashboard

There are two ways to install the Tungsten Dashboard - the standard way using a Linux server and manual configuration of Apache 2 and HAProxy, or via a Docker container.

6.1. Install the Tungsten Dashboard - Standard Method

This section describes the standard method of installing the Tungsten Dashboard when using a Linux server with Apache 2, PHP and optionally HAProxy installed locally.

> **Important**
>
> Please change the example values below to match your specific environment.

For example, create a new user called `tungsten`, group `tungsten`, homedir `/home/tungsten`:

```
shell> sudo useradd -m -d /home/tungsten -s /bin/bash -c "Tungsten Dashboard" -U tungsten
```

Note: Later on you will need to add the `apache` user to the `tungsten` group and restart apache.

Now create the Tungsten Dashboard web root directory and all needed subdirectories:

```
shell> sudo mkdir /volumes/data/www/tungsten
shell> sudo chown -R tungsten: /volumes/data/www/tungsten

shell> sudo su - tungsten
shell> cd /volumes/data/www/tungsten
shell> mkdir etc logs
shell> chmod 2775 logs
shell> chmod 2755 etc
```

Still as user tungsten, download the software using the temporary URL provided by Continuent, or login to the web download portal to obtain the software [https://www.continuent.com/downloads/], then copy to the web root directory for use in the next step:

```
shell> cd
shell> wget -O tungsten-dashboard-1.0.0-123.tar.gz 'TEMP_URL_PROVIDED_BY_CONTINUENT'
shell> tar xvzf tungsten-dashboard-1.0.0-123.tar.gz
shell> cd tungsten-dashboard-1.0.0-123
shell> rsync -a html/ /volumes/data/www/tungsten/html/
shell> chmod 2775 /volumes/data/www/tungsten/html
```

If not present, create the `html/locks` directory and set the permissions:

```
shell> mkdir /volumes/data/www/tungsten/html/locks
shell> chmod 2775 /volumes/data/www/tungsten/html/locks
```

6.2. Install the Tungsten Dashboard - Docker Method

This section describes the docker method of installing the Tungsten Dashboard using a pre-defined Docker container provided by Continuent containing Apache, PHP, HAProxy and the Dashboard.

6.2.1. Dashboard Docker Install - Quick Start

Below are the key steps needed to install the Tungsten Dashboard via Docker.

- Copy the downloaded software to the target Linux host:

  ```
  desktop> scp tungsten-dashboard-docker-1.0.15-13.tar.gz tungsten@db1:
  ```

- SSH to the target host:

  ```
  desktop> ssh tungsten@db1
  ```

- Extract the tarball to a temporary location like your home directory on that host:

  ```
  shell> tar xvzf tungsten-dashboard-docker-1.0.15-13.tar.gz
  tungsten-dashboard-docker/
  tungsten-dashboard-docker/README
  tungsten-dashboard-docker/config.php
  tungsten-dashboard-docker/docker-compose.yml
  tungsten-dashboard-docker/dshell
  tungsten-dashboard-docker/haproxy/
  tungsten-dashboard-docker/haproxy/haproxy.cfg
  ```

```
tungsten-dashboard-docker/hastat
tungsten-dashboard-docker/settings.d/
tungsten-dashboard-docker/settings.d/apiAuth.json
tungsten-dashboard-docker/settings.d/apiPassword.json
tungsten-dashboard-docker/settings.d/apiSSL.json
tungsten-dashboard-docker/settings.d/apiUser.json
tungsten-dashboard-docker/settings.d/apiVersion.json
tungsten-dashboard-docker/settings.d/startExpanded.json
tungsten-dashboard-docker/settings.d/useHAProxy.json
tungsten-dashboard-docker/tungsten_generate_haproxy.pl
tungsten-dashboard-docker/tungsten-dashboard-docker-saved-1.0.15-13.tar
```

- Proceed to the extracted directory:

```
shell> cd tungsten-dashboard-docker
```

- Inform the Docker server of the new Dashboard image to install:

```
shell> sudo docker load  --input tungsten-dashboard-docker-saved-1.0.15-13.tar
```

- Create Dashboard login credentials, i.e. user tungsten with password secret:

```
shell> (cd etc; htpasswd -c .htpasswd tungsten)
```

- Add explicit etc/hosts entries under the extra_hosts sections for both services, `haproxy` and `dashboard`:

```
shell> vi docker-compose.yml
```

- Create cluster-specific HAProxy entries:

  Note the following will only work when a valid Tungsten install and `/etc/tungsten/tungsten.ini` file exists:

```
shell > tpm generate-haproxy-for-api --port 8201 >> haproxy/haproxy.cfg
```

- Launch the containers:

```
shell> sudo docker-compose up -d
```

- Validate that everything is running properly:

```
shell> sudo docker ps
shell> sudo docker logs haproxy
shell> sudo docker logs dashboard
```

- View the GUI in a browser:

```
Browse to http://localhost:8080
Click on "Please click here to auto-define an existing service (recommended)"
Add the new cluster using:
Host Name: haproxy
Port Number: 8201
```

6.2.2. Dashboard Docker Install - Details

- Create the `.htpasswd` file under the `etc/` folder with your desired login and password for the Dashboard:

```
shell> (cd etc; htpasswd -c .htpasswd tungsten)
```

- Edit the `config.json` file and add the login you just created to the `administrators` line. The `tungsten` user is pre-populated.
- Create the HAProxy frontend and backend entries from your existing INI by running either tpm generate-haproxy-for-api or the enclosed ./tungsten_generate_haproxy.pl - for example, to append the results to the `haproxy/haproxy.cfg` file:

```
shell> ./tungsten_generate_haproxy.pl >> haproxy/haproxy.cfg
```

- Populate `/etc/hosts` inside the container[s] by adding indented lines under the `extra_hosts:` directive:

```
shell> vi docker-compose.yml
...
services:
dashboard:
  extra_hosts:
    db1: 10.0.0.101
    db2: 10.0.0.102
    db3: 10.0.0.103
    db4: 10.0.0.104
    db5: 10.0.0.105
    db6: 10.0.0.106
```

```
...
haproxy:
  extra_hosts:
    db1: 10.0.0.101
    db2: 10.0.0.102
    db3: 10.0.0.103
    db4: 10.0.0.104
    db5: 10.0.0.105
    db6: 10.0.0.106
...
```

- Run the two containers via the `sudo docker-compose up -d` command.
- Validate the containers via the `sudo docker ps` command.
- Point your browser to `http://localhost:8080`
- Click on "Please click here to auto-define an existing service (recommended)"
- Add the new cluster using hostname `haproxy` and the frontend port number(s) created above, i.e. `8201`

NOTES

- The Dashboard sees HAProxy under the hostname `haproxy` on port 8090-809X (depending on the haproxy config).
- The `dashboard` and `haproxy` containers are on an internal network that can see each other with these hostnames.
- The netcat command nc is required to use the included hastat script:

```
shell> yum -y install nc
```

- Use the enclosed dshell script to ssh to the container by providing the CONTAINER ID listed with docker ps as the only argument:

```
shell> sudo docker ps
shell> ./dshell {container name or containerid}
```

For example:

```
shell> ./dshell dashboard
shell> ./dshell haproxy
```

6.2.3. Docker Compose Quick Reference Guide

Docker-compose Install Summary

> **Note**
>
> NOTE: To install a different version of Compose, substitute 1.29.2 below with the version of Compose you want to use...

```
shell> sudo curl -L "https://github.com/docker/compose/releases/download/1.29.2/docker-compose-$(uname -s)-$(uname -m)" -o /usr/local/bin/docker-compose
shell> sudo chmod +x /usr/local/bin/docker-compose
shell> sudo ln -s /usr/local/bin/docker-compose /usr/bin/docker-compose
shell> docker-compose --version
```

Click here to see the Docker documentation for more information about docker-compose.

Docker-compose Key Tasks Summary

```
== Start in foreground mode:
shell> sudo docker-compose up

== Start as a daemon:
shell> sudo docker-compose up -d

== Stop all containers:
shell> sudo docker-compose down
```

6.2.4. Docker Quick Reference Guide

```
== View the logs from a container:
shell> sudo docker logs {ID|name}

== Login to the shell on a container:
```

```
shell> sudo docker exec -it {ID|name} /bin/bash
~or~
shell> ./dshell {ID|name}

== List all containers:
shell> sudo docker ps

== Restart a container:
shell> sudo docker restart {ID}
```

6.2.5. Dashboard Docker Install - Troubleshooting

- To install tools for checking connectivity on a container:

```
shell> ./dshell {container name or containerid}
shell> apt update
shell> apt install iputils-ping telnet netcat-openbsd curl jq
```

- To check the HAProxy status:

```
shell> ./dshell haproxy
root@f57d517b7acf:/# echo "show stat" | nc -U /var/lib/haproxy-stats-socket
```

- To check the Manager reachability:

```
shell> ./dshell haproxy
root@f57d517b7acf:/# /usr/bin/curl -s  --insecure --user tungsten:secret  --request GET 'https://db1-demo.continuent.com:8090/api/v2/manager/cluster/status/' | jq
```

- ERROR:

  Cannot raise FD limit to 8066, limit is 1024

  CAUSE:

  /etc/sysconfig/docker has incorrect options configured

  SOLUTION:

```
shell> sudo docker-compose down
shell> vi /etc/sysconfig/docker

==> CHANGE FROM:
OPTIONS="--default-ulimit nofile=1024:4096"
==> TO:
#OPTIONS="--default-ulimit nofile=1024:4096"
OPTIONS=""

shell> sudo service docker restart
shell> sudo docker-compose up -d
```

  ERROR DETAILS:

```
shell> sudo docker-compose up

...
haproxy      | [NOTICE] 161/185553 (1) : haproxy version is 2.3.10-4764f0e
haproxy      | [NOTICE] 161/185553 (1) : path to executable is /usr/local/sbin/haproxy
haproxy      | [ALERT] 161/185553 (1) : [haproxy.main()] Cannot raise FD limit to 8066, limit is 1024.
haproxy exited with code 1
...

shell> ps -ef | grep docker
root      2681     1  0 12:57 ?        00:00:10 /usr/bin/dockerd --default-ulimit nofile=1024:4096
```

- ERROR:

  Dashboard shows "Internal Server Error" in a browser; logs show "Could not open password file: /var/www/html/etc/.htpasswd"

  CAUSE:

  Forgot to create the .htpasswd file

  SOLUTION:

```
shell> cd tungsten-dashboard-docker
shell> sudo htpasswd -c etc/.htpasswd tungsten
```

  ERROR DETAILS:

```
shell> sudo docker logs dashboard

[Tue Jun 15 19:37:23.074342 2021] [authn_file:error] [pid 20] (2)No such file or directory: [client 222.33.44.55:63370] AH01620: Could not open password
```

- COMMAND:

  sudo docker-compose up

  ERROR:

  docker.errors.DockerException: Error while fetching server API version: ['Connection aborted.', FileNotFoundError[2, 'No such file or directory']]

  CAUSE:

  Docker server process not running

  SOLUTION:

  Start the Docker server process

6.3. Dashboard Initial JSON Configuration

Place under `settings.d/` as needed:

```
apiAuth.json
{
  "apiAuth": 1
}

apiPassword.json
{
  "apiPassword": "default"
}

apiSSL.json
{
  "apiSSL": 1
}

apiUser.json
{
  "apiUser": "default"
}

apiVersion.json
{
  "apiVersion": 2
}

useHAProxy.json
{
  "useHAProxy": 1
}
```

Chapter 7. Enabling Dashboard Security

The Dashboard relies upon the Basic Authentication feature of the web server to provide login security. Additionally, Role-Based Access Control (RBAC) uses that login string to provide additional functionality within the Dashboard. Without Basic Authentication in the web server, RBAC will not work in the Dashboard.

When RBAC is enabled (requires web server Basic Auth to be working fully), there are just two roles currently:

- administrator - which gives read-write access to everything to any valid login listed in the `"administrators":[]` option in the `WEBROOT/html/config.json` file.
- operator - which is read-only and is the role given to anyone with a valid login

To enable login and password security for the Dashboard along with Role-Based Access Control (RBAC), be sure to do the following:

- Deploy the correct Apache config to enable Basic Authentication pointing to the `WEBROOT/etc/.htpasswd` file.

  Please see Section 8.2.2, "Create the Dashboard-specific Web Server Configuration File" and Section 8.2.3, "Configure Web Server Boot and Restart Process".

- Ensure that the `WEBROOT/etc/.htpasswd` file contains one or more login/password pairs using the htpasswd command.

  Please see Section 8.2.4, "Populate Logins Using htpasswd".

- Configure the Dashboard RBAC via the `WEBROOT/html/config.json` file to add logins from the `WEBROOT/etc/.htpasswd` file to the `administrators` JSON array.

  Please see Section 8.2.5, "Enable RBAC via `config.json`".

Chapter 8. Configure the Apache 2 Web Server

> **Important**
>
> Please change the example values below to match your specific environment.

8.1. Example: Web Server on Ubuntu

This sequence of steps to install and configure Apache 2 and HAProxy on Ubuntu was derived from a session with a customer. YMMV as always.

- Generate the custom frontend and backend definitions for HAProxy, and prepare the cluster for updates to the configuration of the cluster nodes, if you have not done them already:

  On a single database node per cluster:

```
tungsten@db1 shell> tpm generate-haproxy-for-api
tungsten@db1 shell> echo 'set policy maintenance' | cctrl
```

- Update the cluster configuration to support the REST APIv2.

  On all database nodes:

```
tungsten@dbN shell> vi /etc/tungsten/tungsten.ini
==> Ensure that the rest api settings have been added to the above!
tungsten@dbN shell> tpm update
tungsten@dbN shell> tapi ping
==> Create the REST API admin user if you did not do so at install time:
tungsten@dbN shell> tapi --create --create-user tungsten --create-password secret
```

- AFTER all tpm updates have been completed, return the cluster to AUTOMATIC mode.:

  On a single database node per cluster:

```
tungsten@db1 shell> echo 'set policy automatic' | cctrl
```

- Update the `/etc/hosts` file to ensure all nodes are reachable.

  On the Dashboard web server host, perform the following steps:

```
shell> nslookup dashboard.customer.org
shell> sudo vi /etc/hosts
==> Ensure this server's hostname exists in the hosts file
==> Ensure that all database nodes exist in the hosts file
```

- Install Apache 2 and all other needed software on the Dashboard web server:

```
shell# apt update
shell# apt install apache2
shell# systemctl start apache2
shell# systemctl enable apache2
shell# systemctl status apache2
shell# apache2 -V
shell# apt install php php-curl libapache2-mod-php jq socat haproxy
shell# vi /etc/php/7.4/apache2/php.ini
==>    enable extension=php_curl by removing the leading semi-colon (;)
shell# systemctl restart apache2
```

- In this example, the customer placed the Tungsten Dashboard web root directory onto an NFS mount, so we needed to create that set of directories before installing the Dashboard package:

```
shell# mkdir -p /nfs/tungsten/html /nfs/tungsten/etc /nfs/tungsten/logs
shell# chown -R www-data: /nfs/tungsten
shell# chmod -R ug+rw /nfs/tungsten
```

- Create the `.htpassword` file to provide Basic Authorization functionality.

```
shell# htpasswd -c /nfs/tungsten/etc/.htpasswd tungsten
secret
secret
shell# cat /nfs/tungsten/etc/.htpasswd
```

- Install the Tungsten Dashboard software package from your home directory into the web root directory, on NFS in this case:

```
shell# cd
```

```
shell# tar xvzf tungsten-dashboard-1.0.15-13.tar.gz
shell# cd tungsten-dashboard-1.0.15-13
shell# cp html/config.php.sample html/config.php
shell# cp html/config.json.sample html/config.json
shell# rsync -a html/ /nfs/tungsten/html/
shell# vi /nfs/tungsten/html/config.json
==> Update the administrators entry if needed:
"administrators":[ tungsten ],
==> Update the enableRBAC entry to 1:
"enableRBAC":1,
```

- Configure the Dashboard virtualhost in Apache2:

  For example Apache 2 conf file entries, please see the above section Section 8.2, “Example: Web Server on Amazon Linux 2”.

```
shell# less /etc/apache2/envvars
shell# less /etc/apache2/apache2.conf
shell# vi /etc/apache2/sites-enabled/000-default.conf
==> Edit the existing section to add the needed items
shell# apachectl configtest
shell# systemctl restart apache2
shell# systemctl status apache2
shell# journalctl -xe
shell# cat /var/log/apache2/error.log
```

- Validate that the Dashboard web server host is able to reach all the nodes:

```
shell> for host in db1 db2 db3 db4 db5 db6; do
  ping $host
  #telnet $host 8090
  /usr/bin/curl -s --insecure --user tungsten:secret 'http://${host}:8090/api/v2/manager/status' | jq .
done
```

- Configure and test HAProxy:

```
shell# systemctl enable haproxy
shell# systemctl start haproxy
shell# systemctl status haproxy
shell# vi /etc/haproxy/haproxy.cfg
shell# systemctl restart haproxy
shell# systemctl status haproxy
shell# socat stdio /var/run/haproxy.sock | grep -i stat

shell> telnet localhost 8201
shell> telnet localhost 8202
shell> telnet localhost 8203
shell> /usr/bin/curl -s --insecure --user tungsten:secret 'http://localhost:8201/api/v2/manager/status' | jq .
shell> /usr/bin/curl -s --insecure --user tungsten:secret 'http://localhost:8202/api/v2/manager/status' | jq .
shell> /usr/bin/curl -s --insecure --user tungsten:secret 'http://localhost:8203/api/v2/manager/status' | jq .
```

8.2. Example: Web Server on Amazon Linux 2

8.2.1. Add `apache` user to `tungsten` group

Add the `apache` user to the `tungsten` group:

```
shell> sudo usermod -a -G tungsten apache
```

8.2.2. Create the Dashboard-specific Web Server Configuration File

Create the `apache` configuration file for the web service:

```
shell> sudo vim /etc/httpd/conf.d/z01-tungsten-dashboard.conf
```

> **Important**
>
> Be sure to check the configuration and correct it until the configtest passes:
>
> ```
> shell> sudo apachectl configtest
> ```

Select one of the examples below to populate the web server config file.

For Apache version 2.2 with no authentication or Role-Based Access Control (RBAC):

```
<VirtualHost *:80>
ServerName dashboard.yourdomain.com
```

```
DocumentRoot /volumes/data/www/tungsten/html
DirectoryIndex index.php
ServerAdmin dashboard.apache.admin@yourdomain.com

  Header set Access-Control-Allow-Origin *

  ErrorLog "| /usr/sbin/rotatelogs /volumes/data/www/tungsten/logs/errors.log 86400"
  CustomLog "| /usr/sbin/rotatelogs /volumes/data/www/tungsten/logs/access.log 86400" combined

<Directory "/volumes/data/www/tungsten/html">
  AllowOverride All
  Options +FollowSymLinks +ExecCGI -Indexes
  Order allow,deny
  Allow from all
</Directory>

</VirtualHost>
```

For Apache version 2.2 with auth and RBAC using Basic Auth with an htpasswd-generated file:

```
<VirtualHost *:80>
ServerName dashboard.yourdomain.com

DocumentRoot /volumes/data/www/tungsten/html
DirectoryIndex index.php
ServerAdmin dashboard.apache.admin@yourdomain.com

  Header set Access-Control-Allow-Origin *

  ErrorLog "| /usr/sbin/rotatelogs /volumes/data/www/tungsten/logs/errors.log 86400"
  CustomLog "| /usr/sbin/rotatelogs /volumes/data/www/tungsten/logs/access.log 86400" combined

<Directory "/volumes/data/www/tungsten/html">
  AllowOverride All
  Options +FollowSymLinks +ExecCGI -Indexes
  Order allow,deny
  Allow from all
  AuthType Basic
  AuthName "Tungsten Dashboard - RESTRICTED"
  AuthUserFile /volumes/data/www/tungsten/etc/.htpasswd
  Require valid-user
</Directory>

</VirtualHost>
```

For Apache version 2.4 with no authentication or Role-Based Access Control (RBAC):

```
<VirtualHost *:80>
ServerName dashboard.yourdomain.com

DocumentRoot /volumes/data/www/tungsten/html
DirectoryIndex index.php
ServerAdmin dashboard.apache.admin@yourdomain.com

  ErrorLog "| /usr/sbin/rotatelogs /volumes/data/www/tungsten/logs/errors.log 86400"
  CustomLog "| /usr/sbin/rotatelogs /volumes/data/www/tungsten/logs/access.log 86400" combined

<Directory "/volumes/data/www/tungsten/html">
  AllowOverride All
  Options +FollowSymLinks +ExecCGI -Indexes
  Order allow,deny
  Allow from all
  Require all granted
</Directory>

</VirtualHost>
```

For Apache version 2.4 with auth and RBAC using Basic Auth with an htpasswd-generated file:

```
<VirtualHost *:80>
ServerName dashboard.yourdomain.com

DocumentRoot /volumes/data/www/tungsten/html
DirectoryIndex index.php
ServerAdmin dashboard.apache.admin@yourdomain.com

  ErrorLog "| /usr/sbin/rotatelogs /volumes/data/www/tungsten/logs/errors.log 86400"
  CustomLog "| /usr/sbin/rotatelogs /volumes/data/www/tungsten/logs/access.log 86400" combined
```

```
<Directory "/volumes/data/www/tungsten/html">
  AllowOverride All
  Options +FollowSymLinks +ExecCGI -Indexes
  Order allow,deny
  Allow from all
  #Require all granted
  <RequireAll>
    AuthType Basic
    AuthName "Tungsten Dashboard - RESTRICTED"
    AuthUserFile /volumes/data/www/tungsten/etc/.htpasswd
    Require valid-user
  </RequireAll>

</Directory>

</VirtualHost>
```

For Apache version 2.4 with auth and RBAC using Basic Auth via LDAP:

```
shell> sudo yum install -y mod_ldap
```

```
<VirtualHost *:80>
ServerName dashboard.yourdomain.com

DocumentRoot /volumes/data/www/tungsten/html
DirectoryIndex index.php
ServerAdmin dashboard.apache.admin@yourdomain.com

  ErrorLog "| /usr/sbin/rotatelogs /volumes/data/www/tungsten/logs/errors.log 86400"
  CustomLog "| /usr/sbin/rotatelogs /volumes/data/www/tungsten/logs/access.log 86400" combined

<Directory "/volumes/data/www/tungsten/html">
  AllowOverride All
  Options +FollowSymLinks +ExecCGI -Indexes
  Order allow,deny
  Allow from all
  #Require all granted
  <RequireAll>
                AuthType Basic
                AuthName "Tungsten Dashboard - RESTRICTED"
                AuthBasicProvider ldap
                AuthLDAPURL ldap://ldap.ad.demo.com:XXX/DC=ad,DC=demo,DC=com?sAMAccountName?sub
                AuthLDAPBindDN ldapuser@ad.demo.com
                AuthLDAPBindPassword abcdef123456
                Require ldap-group CN=DataServicesAdmins,OU=SQL,OU=Groups,OU=London,OU=NewYork,OU=United States,OU=North America,DC=ad,DC=demo,DC=com
                Require valid-user
  </RequireAll>
</Directory>

</VirtualHost>
```

8.2.3. Configure Web Server Boot and Restart Process

Configure start-at-boot and restart the web server:

```
shell> sudo chkconfig httpd on
shell> sudo service httpd restart
shell> sudo service httpd status
~OR~
shell> sudo systemctl enable httpd
shell> sudo systemctl restart httpd
shell> sudo systemctl status httpd
```

8.2.4. Populate Logins Using htpasswd

Ensure that the `WEBROOT/etc/.htpasswd` file contains one or more login/password pairs using the htpasswd command.

```
shell> htpasswd -c /volumes/data/www/tungsten/etc/.htpasswd {desiredlogin}
```

8.2.5. Enable RBAC via `config.json`

To enable RBAC security, the `$WEBROOT/html/config.json` file will need to be updated with two settings: `"enableRBAC": 1` and `"administrators": []`, for example:

```
{
  "clusters": {
```

```
  },
  "menus": {
  },
  "settings": {
      "administrators": [ "tungsten","admin","root" ],
      "enableRBAC":1
  }
}
```

When RBAC is enabled (requires web server Basic Auth to be working fully), there are just two roles currently:

- `administrator` - which gives read-write access to everything to any valid login listed in the `"administrators":[]` option in the config file.
- `operator` - which is read-only and is the role given to anyone with a valid login. There is NO explicit entry for "operators" in the config file.

8.2.6. Configure SELinux for Apache

Warning

There are additional steps to take when SELinux is enabled.

To check if SELinux is enabled:

```
shell> getenforce
shell> sestatus
```

These are example extra steps to take if SELinux is enabled:

```
shell> sudo -i
shell> chcon -R -t httpd_sys_rw_content_t /volumes/data/www/tungsten/html
shell> chcon -R -t httpd_sys_rw_content_t /volumes/data/www/tungsten/logs
shell> semanage fcontext -a -t httpd_sys_rw_content_t "/volumes/data/www/tungsten/html(/.*)?"
shell> semanage fcontext -a -t httpd_sys_rw_content_t "/volumes/data/www/tungsten/logs(/.*)?"
shell> restorecon -Rv /volumes/data/www/tungsten/*
shell> semanage port -a -t http_port_t -p tcp 8090
shell> setsebool -P httpd_can_network_connect 1
shell> systemctl restart httpd
shell> systemctl restart php-fpm
```

Be sure to check in the `audit.log` for any `denied` messages containing `http` or `php`.

Here are two example commands to run to help troubleshoot selinux and httpd:

```
shell> ausearch -m avc -c httpd
shell> grep httpd /var/log/audit/audit.log
```

Chapter 9. Install and Configure HA Proxy

The Tungsten Cluster Manager listens on port 8090 for API calls, so we configure the HA Proxy listener ports to not conflict with that.

As of v1.0.11-2, the default listener port for HAProxy is 8201, changed from 8091 to prevent port conflicts with Prometheus exporters when installed on a Tungsten v7+ cluster node.

There must be one frontend per cluster, so the first cluster is assigned the default listener port number 8201.

In the examples below, we assign frontend port 8201 to the composite global, frontend port 8202 to the cluster east and frontend port 8203 to the cluster west.

It is imperative that there be one backend per cluster containing all nodes in that cluster. In the case of a composite, the backend should contain all nodes from all member clusters.

In the below examples, backend east contains member nodes db1-3, backend west contains nodes db4-6 and backend global contains nodes db1-6.

NOTE: See `haproxy.cfg` in the `examples/` directory for a more complete sample config to be used locally on a web server or jump host.

9.1. Install and Prepare HA Proxy

Install and prepare the HA Proxy deployment:

```
shell> sudo -i
shell> yum install haproxy
shell> cd /etc/haproxy/
shell> cp haproxy.cfg haproxy.cfg.orig
```

9.2. Generate the Frontend and Backend Definitions

Generate the custom frontend and backend definitions for HAProxy from the `/etc/tungsten/tungsten.ini` file.

Important

The following will only work on a host where Tungsten Clustering is installed and a valid `/etc/tungsten/tungsten.ini` file exists.

Create cluster-specific HAProxy entries - for example, perform this command on a single database node per cluster:

```
shell > tpm generate-haproxy-for-api --port 8201 >> haproxy/haproxy.cfg
```

9.3. Modify the HAProxy Configuration File

Edit `/etc/haproxy/haproxy.cfg` and define the global options, defaults, frontend listeners, backend services and associated hosts using the provided defaults below and the output from above:

```
shell> vim /etc/haproxy/haproxy.cfg
```

```
global
  chroot      /var/lib/haproxy
  pidfile     /var/run/haproxy.pid
  maxconn     4000
  user        haproxy
  group       haproxy
  daemon

  # turn on stats unix socket
  stats socket /var/lib/haproxy/stats

defaults
  mode                    tcp
  log                     global
  option                  tcplog
  option                  dontlognull
  option                  redispatch
  retries                 3
  timeout queue           1m
  timeout connect         10s
  timeout client          1m
```

```
  timeout server          1m
  timeout check           10s
  maxconn                 3000

frontend world
  bind *:8201
  default_backend             global

frontend east
  bind *:8202
  default_backend             east

frontend west
  bind *:8203
  default_backend             west

backend world
  balance     roundrobin
  server  db1 db1.yourdomain.com:8090 check
  server  db2 db2.yourdomain.com:8090 check
  server  db3 db3.yourdomain.com:8090 check
  server  db4 db4.yourdomain.com:8090 check
  server  db5 db5.yourdomain.com:8090 check
  server  db6 db6.yourdomain.com:8090 check

backend east
  balance     roundrobin
  server  db1 db1.yourdomain.com:8090 check
  server  db2 db2.yourdomain.com:8090 check
  server  db3 db3.yourdomain.com:8090 check

backend west
  balance     roundrobin
  server  db4 db4.yourdomain.com:8090 check
  server  db5 db5.yourdomain.com:8090 check
  server  db6 db6.yourdomain.com:8090 check
```

9.4. Ensure HAProxy Starts at Boot

Configure start at boot:

```
shell> sudo chkconfig haproxy on
~OR~
shell> sudo systemctl enable haproxy
```

9.5. Restart HAProxy

Restart the HA Proxy service:

```
shell> sudo service haproxy restart
~OR~
shell> sudo systemctl restart haproxy
```

9.6. Verify HAProxy Started

Verify that HAProxy has started properly:

```
shell> sudo service haproxy status
~OR~
shell> sudo systemctl status haproxy

shell> sudo socat stdio /var/run/haproxy.sock | grep -i stat
shell> telnet localhost 8201
shell> telnet localhost 8202
shell> telnet localhost 8203
```

9.7. Configure SELinux for HAProxy

Warning

There are additional steps to take when SELinux is enabled.

To check if SELinux is enabled:

```
shell> getenforce
```

```
shell> sestatus
```

These are example extra steps to take if SELinux is enabled:

```
shell> sudo setsebool -P httpd_can_network_connect 1
shell> sudo setsebool -P haproxy_connect_any 1
shell> sudo systemctl restart haproxy
```

Be sure to check in the `audit.log` for any `denied` messages containing `haproxy`.

Here are two example commands to run to help troubleshoot selinux and haproxy:

```
shell> ausearch -m avc -c haproxy
shell> grep haproxy /var/log/audit/audit.log
```

For more information about HAProxy, please visit http://www.haproxy.org

Chapter 10. Test Connectivity to the Tungsten Manager via HAProxy

10.1. Test Connectivity to APIv1 via HAProxy

Test connectivity to the Tungsten Manager API Version 1 via HAProxy using curl:

```
shell> curl -s http://localhost:8201/manager/status/global/
shell> curl -s http://localhost:8202/manager/status/east/
shell> curl -s http://localhost:8203/manager/status/west/
shell> curl -s -X POST http://localhost:8203/manager/control/west/heartbeat
```

Please note that APIv1 does not support authentication or SSL.

10.2. Test Connectivity to APIv2 via HAProxy

Version 1.0.10. This feature was first introduced in Tungsten Dashboard version 1.0.10-125

Test connectivity to the Tungsten APIv2 via HAProxy using curl for Secure (SSL-enabled) deployments:

```
shell> /usr/bin/curl --user tungsten:demo -k --request GET 'https://localhost:8201/api/v2/manager/status'
shell> /usr/bin/curl --user tungsten:demo -k --request POST 'https://localhost:8201/api/v2/manager/control/service/south/heartbeat'

shell> /usr/bin/curl -s --insecure --user tungsten:secret 'https://localhost:8201/api/v2/manager/status' | jq .
shell> /usr/bin/curl -s --insecure --user tungsten:secret 'https://localhost:8202/api/v2/manager/status' | jq .
shell> /usr/bin/curl -s --insecure --user tungsten:secret 'https://localhost:8203/api/v2/manager/status' | jq .
```

Test connectivity to the Tungsten APIv2 via HAProxy using curl for NON-Secure (No SSL) deployments:

```
shell> /usr/bin/curl --user tungsten:demo --request GET 'http://localhost:8201/api/v2/manager/status'
shell> /usr/bin/curl --user tungsten:demo --request POST 'http://localhost:8201/api/v2/manager/control/service/south/heartbeat'
```

Chapter 11. Configure the Tungsten Dashboard

11.1. Configure the Required APIv2 Admin User for Tungsten Cluster

> **Important**
>
> If you did NOT configure APIv2 before installing or upgrading v7, you will need to perform the below Create Admin User Procedure to create and enable the required APIv2 Admin user on all database nodes!

APIv2 Security Basics

Authentication is handled using username/password pairs, and an initial admin user account MUST be created on every node before the API can be used on that node.

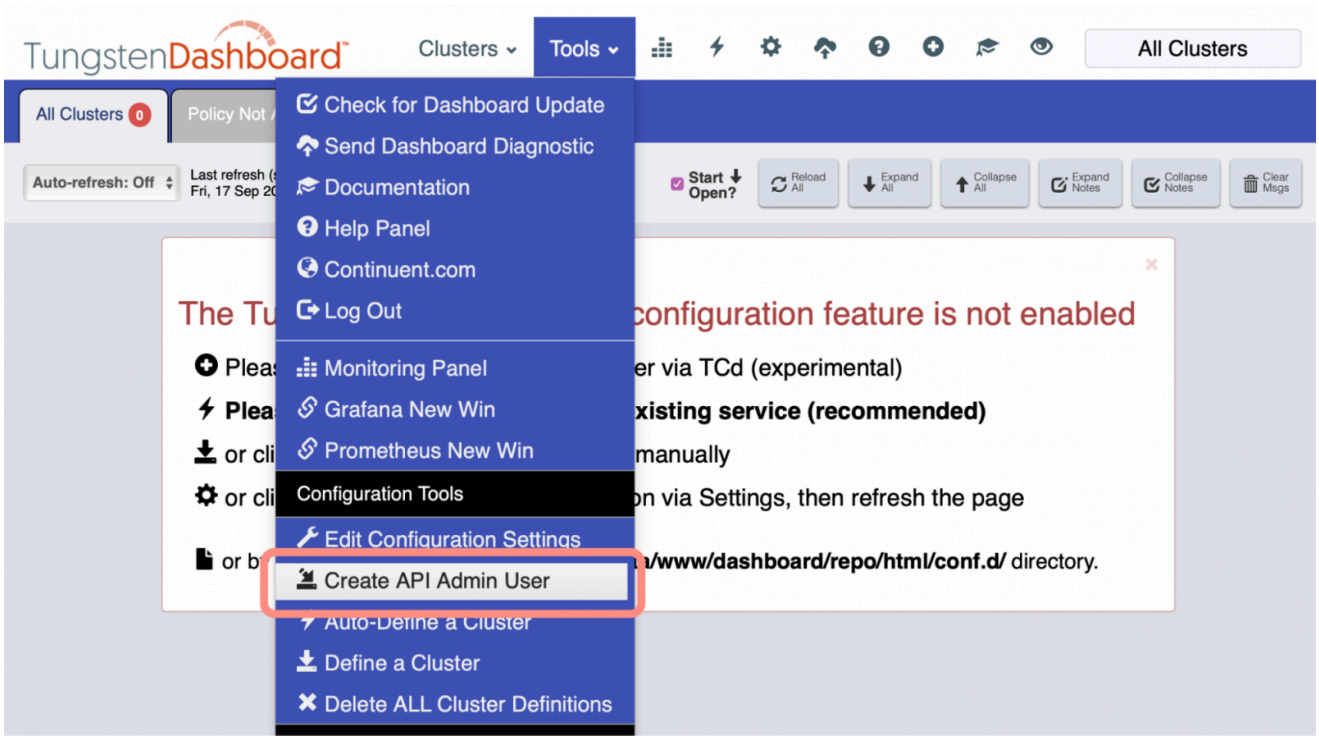
If no admin user has been created, only the ping and createAdminUser API calls will be available.

Users created through the API are host-specific only, and will have to be re-created on each host of the cluster. At install time, the tpm command will handle populating the API user and password on all cluster nodes when you specify the `rest-api-admin-user` and `rest-api-admin-pass` options.

Create Admin User Procedure

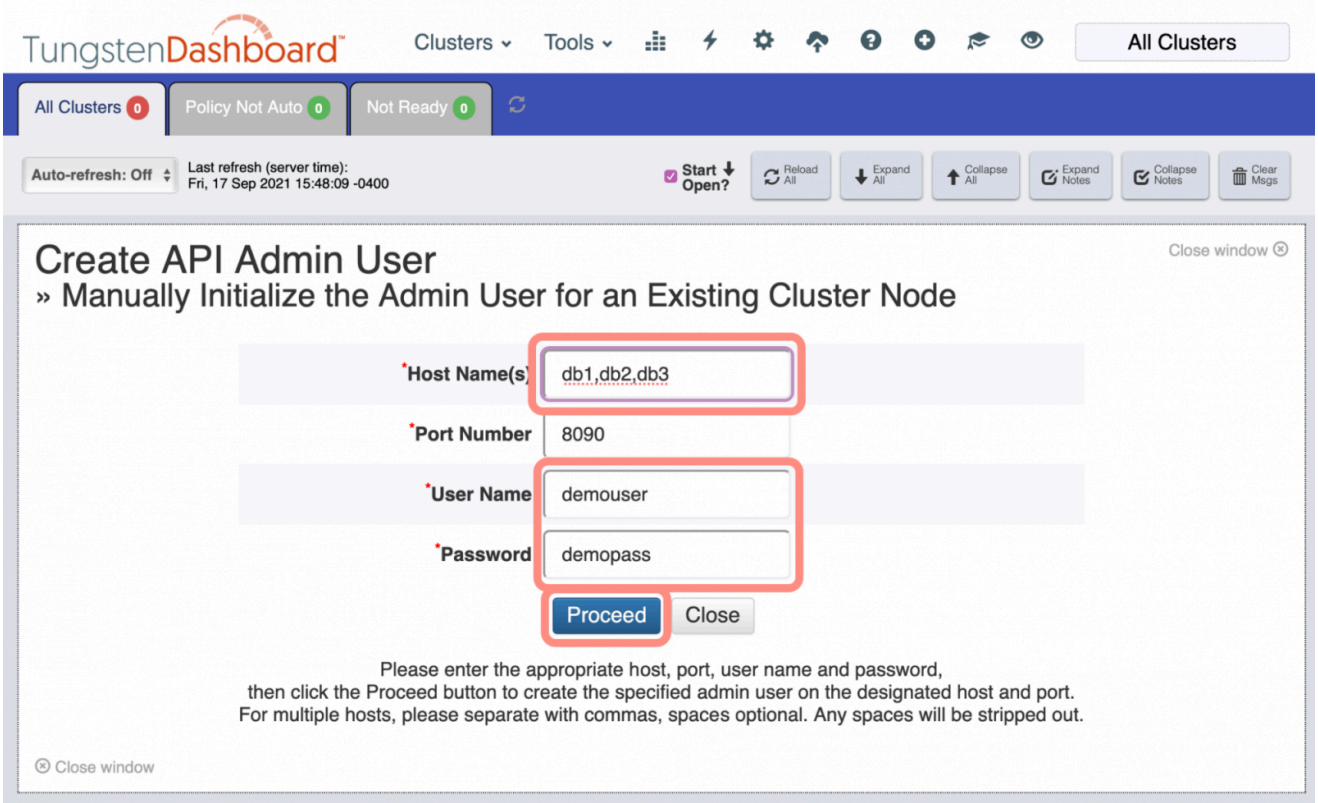
From the Dashboard GUI, select the Tools menu, then option "Create API Admin User".

Figure 11.1. Tungsten Dashboard Create APIv2 Admin User Menu Option



Please enter the appropriate host or hosts, port, user name and password, then click the Proceed button to create the specified admin user on the designated host[s] and port. For multiple hosts, please separate with commas, spaces optional. Any spaces will be stripped out.

Figure 11.2. Tungsten Dashboard Create APIv2 Admin User Form



Once the above has completed successfully, record the default API User Name and API Password in the Dashboard Configuration panel, if needed, or set the user and password on a per-cluster basis.

11.2. Tungsten Dashboard Initial Configuration Example

Create and edit the `config.json` file to set the `administrators` and `enableRBAC` values to match your Basic Auth security setup, or additionally create the file `config.json` and populate it with the `$jsonConfig` data structure and appropriate values:

```
shell> sudo su - tungsten
shell> cd /volumes/data/www/tungsten/html/
shell> cp config.php.sample config.php
shell> cp config.json.sample config.json
shell> vim config.json
{
  "clusters": {
  },
  "menus": {
      "tools": {
          "<span class=\"glyphicon glyphicon-link\"></span> Add your links here":"http://docs.continuent.com/tungsten-dashboard-1.0/tungsten-dashboard-con
          "Archive Mode Docs":"http://docs.continuent.com/tungsten-clustering-7.1/operations-status-changingstates.html#operations-status-changingstates-a
          "Cluster Heartbeat Docs":"http://docs.continuent.com/tungsten-clustering7.1/cmdline-tools-cctrl-commands.html#cmdline-tools-cctrl-commands-clust
      }
  },
  "settings": {
      "administrators": [ "tungsten","newuser1","testuser123" ],
      "dashboardMaintenanceScreen":0,
      "enableRBAC":1,
      "enableUpdates":1,
      "sortByConfigOrderNotAlpha":1
  }
}
```

11.3. Tungsten Dashboard Configuration Best Practices

> **Important**
>
> *** There is a one-to-one relationship between Tungsten services and haproxy ports. See examples following this section. ***

- Host and port are required for all clusters.
- A cluster is marked as a composite parent if it has the "children" array, even if the array is empty.
- A cluster is marked as a composite child if it has the "memberOf" key defined.
- All Composite member (child) clusters require their own definitions so we know about the host and port for each.
- All cluster service names MUST be unique. If you have clusters in different environments that have the same names, they will conflict.
- Added in v1.0.7: To solve the above limitation that all cluster service names must be unique, add the sub-key `actualName` pointing to the "real" name of the service, and change the top-level cluster service name to some alias that you understand.

  For example, you have two clusters named "east", one in prod and the other in staging:

```
"clusters": {
       "east_prod": {
               "host":"localhost",
               "port":"8091",
               "actualName":"east"
       },
       "east_staging": {
               "host":"localhost",
               "port":"8092",
               "actualName":"east"
       }
},
```

> **Important**
>
> When using composite clusters, the `children` key (for the composite service) and the `memberOf` key (for the member cluster services) must point to the "alias" names, not the `actualName` value. For example:

```
"clusters": {
       "global_prod": {
               "host":"localhost",
               "port":"8091",
               "children": [ "east_prod","west_prod" ],
               "actualName":"global"
       },
       "east_prod": {
               "host":"localhost",
               "port":"8092",
               "memberOf":"global_prod"
               "actualName":"east"
       },
       "west_prod": {
               "host":"localhost",
               "port":"8092",
               "memberOf":"global_prod"
               "actualName":"west"
       }
},
```

- Please note that the `host: localhost`should remain localhost because this tells the app to call the haproxy server on the GUI server node, which will then handle routing to the appropriate manager/database node.
- You may add your own custom menu options to the tools menu by editing the menus->tools section in the json configuration.
- By default the Auto-refresh feature is disabled (i.e. set to zero). You may enable `autoRefreshDelay` by setting it it to one of the Auto-Refresh time interval values.
- By default, the list of Auto-Refresh time intervals is defined as 5, 10, 30, 60, 120 or 300 seconds. You may change that by using the `autoRefreshList` setting, i.e.:

```
 "autoRefreshList": [3,5,10,30,60,120,300,600]
```

> **Important**
>
> PLEASE NOTE: `autoRefreshList` values less than 3 seconds are strongly discouraged.

- Under normal circumstances, you should not need to get a lock, since all operations automatically attempt to obtain a lock for efficiency purposes. *This has the side-effect of leaving your session in a locked state.*

  There are two settings that help address this situation, `autoUnlockHeartbeat` and `autoUnlockAll`.

  You may set `autoUnlockHeartbeat` to 1 to automatically unlock after issuing a heartbeat command.

  You may set `autoUnlockAll` to 1 to automatically unlock after issuing any command.

- You may set `dashboardMaintenanceScreen` to 1 to display a Maintenance-In-Progress message.

- The default Tab Badge update rate is 30 seconds. You may disable it by setting `tabUpdateRate` to zero (0). You may change the refresh rate in seconds by specifying a non-zero value.

```
"tabUpdateRate":60
```

- Use `lockBaseDir` to change the location of the temporary lock files. The default `lockBaseDir` is `{WEBROOT}`, making the default lock directory `{WEB-ROOT}/locks/`, (i.e. a `lockBaseDir` of `/tmp` will yield a lock directory of `/tmp/locks`).

```
"lockBaseDir":"/tmp"
```

- Added in v1.0.7: Use `msgFadeOutTimer` to automatically close messages after the defined timeout in seconds. The default is `60` seconds.

```
"msgFadeOutTimer":60
```

- Added in v1.0.8: Set `disableConfigDisplay` to 1 to prevent the menu choice for Tools -> Display Configuration from appearing.

```
"disableTooltips":1
```

- Added in v1.0.8: Set `disableTooltips` to 1 to prevent the formatted hover-over help tooltips from appearing.

```
"disableTooltips":1
```

- Added in v1.0.10: Use `enableGrafana` to display a button which opens Grafana in an iframe.

```
"enableGrafana":1
```

- Added in v1.0.10: Use `enablePrometheus` to display a button which opens Prometheus in an iframe.

```
"enablePrometheus":1
```

- Added in v1.0.8: Use `windowTitle` to change the browser window title from the default of "Tungsten Dashboard".

```
"windowTitle":"Prod Env | Tungsten Dashboard"
```

- Added in v1.0.8: The `sortByConfigOrderNotAlpha` controls the Cluster View sort. By default the list of cluster services is sorted alphabetically. Set `sortByConfigOrderNotAlpha` to 1 for the cluster services to be displayed in the order listed in the config.php file.

```
"sortByConfigOrderNotAlpha":1
```

- Added in v1.0.8: The `enableRBAC` setting controls the Role-Based Access Control (RBAC) feature. Disabled by default, set it to one and populate the list of read-write users via the sibling `administrators` setting.

  There are two roles:

  - Administrator - Full access
  - Operator - Read-only access

  When `enableRBAC` is set to one, all users not listed in the `administrators` setting are read-only Operators.

  When enabled, the user's current role will be displayed in the footer. Refresh the page to activate any changes to `config.php`.

  > This feature requires Basic Auth to be properly configured on the Web server.

```
"enableRBAC":1
```

  Use the `administrators` setting to list the users with admin privs:

```
"administrators": [ "adminUser1","adminUser2" ]
```

11.4. Tungsten Dashboard Configuration Settings Reference

Variable Name	Default	Type	Description	Recommended Value	Via GUI
administrators	`[]`	ARRAY of STRINGS	You may set `administrators` to a list of usernames matching those used by Basic Auth (i.e. via htpasswd). Any users not listed are considered to be read-only Operators. Requires that the `enableRBAC` setting be enabled (set to `1`) and that Basic Auth in the web server has been properly configured. Added in v1.0.8	`[ "user1", "user2", "user3" ],`	N
apiAuth	`0`	BOOLEAN	Enabling API Authentication adds the Basic Auth user and password to every API call. You must define and activate the API auth in the Tungsten INI. Added in v1.0.10	1	Y
apiVersion	`1`	INTEGER	Define the API Version to use (1 or 2). Added in v1.0.10	2	Y
apiSSL	`0`	BOOLEAN	Enabling API SSL uses the https protocol instead of http to make API calls. Added in v1.0.10	1	Y
apiPassword	`None`	STRING	The API Password is required when API Authentication is enabled. Added in v1.0.10	{api password}	Y
apiUser	`None`	STRING	The API User is required when API Authentication is enabled. Added in v1.0.10	{api user name}	Y
auditDir	`{WEBROOT}/audit.d`	STRING	Use `auditDir` to change the location of the temporary audit files. Added in v1.0.10	`{WEBROOT}/audit.d`	N
autoLockAll	`1`	BOOLEAN	Automatically lock the Dashboard during ANY non-read-only action to prevent other users from performing any non-read-only actions on this cluster.	`1`	Y
autoLockHeartbeat	`0`	BOOLEAN	Automatically lock the Dashboard during heartbeat actions to prevent other users from performing any non-read-only actions on this cluster. Overridden by `autoLockAll`	`0` if `$autoLockAll=1`, otherwise `1`	Y
autoRefreshDelay	`0`	INTEGER Seconds	Controls the Auto-Refresh feature. By default the Auto-refresh feature is disabled (i.e. set to zero seconds). To have the Auto-Refresh feature enabled upon initial page load, set `autoRefreshDelay` to one of the Auto-Refresh time interval values (see `autoRefreshList`).	`0`	Y
autoRefreshList	`[5, 10, 30, 60, 120, 300]`	Seconds as an array of INTEGERs	The list of time intervals in seconds shown on the Auto-Refresh dropdown menu. `autoRefreshList` values less than 3 seconds are strongly discouraged.	`[5, 7, 10, 15, 20, 30, 60, 120, 300, 600]`	N
autoUnlockAll	`0`	BOOLEAN	Under normal circumstances, you should not need to get a lock, since all operations automatically attempt to obtain a lock for efficiency purposes. This has the side-effect of leaving your session in a locked state. You may set `autoUnlockAll` to 1 to automatically unlock the Dashboard after issuing any command.	`1`	Y
autoUnlockHeartbeat	`0`	BOOLEAN	Under normal circumstances, you should not need to get a lock, since all operations automatically attempt to obtain a lock for efficiency purposes. This has the side-effect of leaving your session in a locked state. You may set `autoUnlockHeartbeat` to 1 to automatically unlock the Dashboard after issuing a heartbeat command.	`1`	Y
connectorPort	`8096`	INTEGER	The `connectorPort` value is used to determine what port to communicate with the Connector upon when performing auto-configuration and auto-define, as well as populating form fields in other places. Only change this if you have changed the API listener port for the Connector as well. Added in v1.0.10	`8096`	N
curlTimeoutGET	`10`	INTEGER Seconds	The timeout used when curl connects to the manager for a GET-specific API call, in seconds. Added in v1.0.10	10	Y
curlTimeoutPOST	`60`	INTEGER Seconds	The timeout used when curl connects to the manager for a POST-specific API call, in seconds. Added in v1.0.10	60	Y

Variable Name	Default	Type	Description	Recommended Value	Via GUI
customerEmail	None	STRING	Use the customerEmail to pre-populate the Tungsten Dashboard diagnostic upload form customer email field. Added in v1.0.9	{Your customer email address}	Y
dashboardMaintenanceScreen	0	BOOLEAN	You may set dashboardMaintenanceScreen to 1 to display a "Maintenance-In-Progress" message.	0	N
disableConfigDisplay	0	BOOLEAN	Set disableConfigDisplay to 1 to prevent the menu choice for Tools -> Display Configuration from appearing. Added in v1.0.8	0, 1if you do not wish read-only access to the configuration via the browser.	Y
disableSettingsEdit	0	BOOLEAN	You can disable the editing of settings in the browser by changing the value of disableSettingsEdit to 1 in the config.php file, in the "settings": { } stanza. Added in v1.0.9	0, 1 if you do not wish to allow settings to be changed via the browser interface.	N
disableTooltips	0	BOOLEAN	Set disableConfigDisplay to 1 to prevent the formatted hover-over help tooltips from appearing. Added in v1.0.8	0, 1 if you do not wish read-only access to the configuration via the browser.	Y
downloadAccessKey	{Obtain from Support}	STRING	Use downloadAccessKey to enable the Tungsten Dashboard self-update feature. Requires downloadSecretKey.	{Assigned Download Key}	N
downloadSecretKey	{Obtain from Support}	STRING	Use downloadSecretKey to enable the Tungsten Dashboard self-update feature. Requires downloadAccessKey.	{Assigned Download Secret}	N
enableAudit	0	BOOLEAN	Enables the admin command audit trail feature. Added in v1.0.10	1	Y
enableAutoConfiguration	0	BOOLEAN	Use enableAutoConfiguration to enable the Tungsten Dashboard auto-configuration feature. Attempts to connect to the Manager on localhost to determine the cluster to display. Related options are managerPort and useHAProxy. Added in v1.0.9	1	Y
enableDebug	0	BOOLEAN	Set enableDebug to 1 to get additional logging information and use the debug software versions when checking for an available update. Added in v1.0.9	0	Y
enableExpertMode	0	BOOLEAN	enableExpertMode disables both confirmation prompts when Deleting All Definitions. Added in v1.0.9	0	Y
enableGrafana	0	BOOLEAN	You may set enableGrafana to 1 to display a "Grafana" button in the top menu, which when clicked will open an iframe to "http://{Dashboard_Server_Hostname}:3000". Added in v1.0.10	0, 1 if you have Grafana available	Y
enablePrometheus	0	BOOLEAN	You may set enablePrometheus to 1 to display a "Prometheus" button in the top menu, which when clicked will open an iframe to "http://{Dashboard_Server_Hostname}:9090". Added in v1.0.10	0, 1 if you have Prometheus available	Y
enableRBAC	0	BOOLEAN	You may set enableRBAC to 1 to enable the use of Role-Based Access Control. Requires that the administrators setting be populated and that Basic Auth in the web server has been enabled or no actions will be allowed. Added in v1.0.8	1	N
enableNotes	0	BOOLEAN	You may set enableNotes to 1 to allow text notes to be saved on a per-node basis Added in v1.0.10	1	Y

Variable Name	Default	Type	Description	Recommended Value	Via GUI
enableUpdates	`1`	BOOLEAN	You may set `enableUpdates` to `0` in the `config.php` file, in the `"settings": { }` stanza to disable the Dashboard self-update feature. When enableUpdates is set to `1` (enabled, default), two other values are needed, `downloadAccessKey` and `downloadSecretKey`. Added in v1.0.9	`1`	N
enableURLDisplay	`0`	BOOLEAN	You may set `enableURLDisplay` to 1 to enable the display of the backend API calls for transparency and learning. Added in v1.0.10	`1`	Y
jumpToTopOnMsg	`1`	BOOLEAN	The Dashboard places all messages at the top of the center scroll window (for now). If you have scrolled down to view information below the fold, and execute a command, it is possible the message will be obscured. When `jumpToTopOnMsg` is set to the default of 1, the center portal will auto-scroll to the top so the message is visible. Set `jumpToTopOnMsg` to 0 to disable this behavior and leave the window where it is after a command is selected.	`0` if you prefer to scroll on your own, when you are ready to do so	Y
flagOnLagColor	`warning`	STRING	Use `flagOnLagColor` to define the row background color when flagOnLagDelay is active. One of: danger, warning or info Requires `flagOnLagDelay > 0`. Added in v1.0.10	`info`	Y
flagOnLagDelay	`0`	INTEGER Seconds	The value used when comparing the replica's seqno to the primary's seqno. If the difference is larger than this value, then set the row background to the danger color. Set to zero (0) to disable. Added in v1.0.10	60	Y
lockBaseDir	`{WEBROOT}`	STRING	Use `lockBaseDir` to change the location of the temporary lock files. The default `lockBaseDir` is `{WEBROOT}`, making the default lock directory `{WEBROOT}/locks/`, (i.e. a `lockBaseDir` of `/tmp` will yield a lock directory of `/tmp/locks`).	`{WEBROOT}`	N
managerPort	`8090`	INTEGER	The `managerPort` value is used to determine what port to communicate with the Manager upon when performing auto-configuration and auto-define, as well as populating form fields in other places. Only change this if you have changed the API listener port for the Manager as well. Added in v1.0.9	`8090`	N
msgFadeOutTimer	`60`	INTEGER Seconds	Use `msgFadeOutTimer` to automatically close messages after the defined timeout in seconds. Added in v1.0.7	`60`	Y
navButtonFormat	`icon`	STRING	Use `navButtonFormat` to control the Cluster View control buttons style on the third navigation bar. You may specify one of: "`icon`", "`text`", "`text,icon`" or "`icon,text`"	`icon,text` until the icons become familiar, then the default value of `icon`	Y
noteGlyphicon	`comment`	STRING	Use `noteGlyphicon` to specify the note-per-node Glyphicon. Added in v1.0.10	`comment`	N
notesDir	`{WEBROOT}/notes.d`	STRING	Use `notesDir` to change the location of the temporary notes files. Added in v1.0.10	`{WEBROOT}/notes.d`	N
replicatorPort	`8097`	INTEGER	The `replicatorPort` value is used to determine what port to communicate with the Replicator upon when performing auto-configuration and auto-define, as well as populating form fields in other places. Only change this if you have changed the API listener port for the Replicator as well. Added in v1.0.10	`8097`	N
sortByConfigOrderNotAlpha	`0`	BOOLEAN	The `sortByConfigOrderNotAlpha` controls the Cluster View sort. By default the list of cluster services is sorted alphabetically. Set `sortByConfigOrderNotAlpha` to 1 for the cluster services to be displayed in the order listed in the config.php file. Added in v1.0.8	`1` if you prefer the clusters to display in the order listed in the `config.php` file.	N
startExpanded	`0`	BOOLEAN	Use `startExpanded` to control the initial display of the cluster nodes. The default of 0 hides the cluster nodes. Set this option to 1 for all nodes to be visible upon initial load.	`1`	Y

Variable Name	Default	Type	Description	Recommended Value	Via GUI
tabUpdateRate	`30`	INTEGER Seconds	The default Tab Badge update rate is 30 seconds. The Tab Bar and associated badges are located in navigation bar two. You may disable it by setting `tabUpdateRate` to zero (0). You may change the refresh rate in seconds by specifying a non-zero value.	`30`	Y
tmpDir	`/tmp`	STRING	Use `tmpDir` to specify where downloaded software packages are saved to by the Tungsten Dashboard self-update feature. Added in v1.0.9	`/tmp`	N
uploadAccessKey	`{Shipped Upload Key}`	STRING	Use `uploadAccessKey` to enable the Tungsten Dashboard self-diagnostic feature. Requires `uploadSecretKey`. Added in v1.0.9	Same as default	N
uploadSecretKey	`{Shipped Upload Secret}`	STRING	Use `uploadSecretKey` to enable the Tungsten Dashboard self-diagnostic feature. Requires `uploadAccessKey`. Added in v1.0.9	Same as default	N
useHAProxy	`0`	BOOLEAN	The `useHAProxy` value is used to determine how to calculate ports when performing auto-configuration and auto-define. Set the value to `1` to both set the base managerPort to 8201 [as of v1.0.11-2, previous value was 8091], and determine the manager port number automatically during various operations based on calulations using the base managerPort. Set the value to `0` (default) to use the base managerPort of 8090 with no attempt to auto-define the port. You can enable this option by changing the "Using HA Proxy?" setting via the Dashboard settings page in the browser, or changing the value in the `config.php` file, in the `"settings": { }` stanza. Added in v1.0.9	`1`	Y
windowTitle	`Tungsten Dashboard`	STRING	Use `windowTitle` to change the browser window title from the default of "Tungsten Dashboard". Added in v1.0.8	`Tungsten Dashboard | {Your Env Here}`	Y
yamlDir	`{WEB-ROOT}/yaml.d`	STRING	Use `yamlDir` to change the location of the temporary yaml files. Added in v1.0.10	`{WEB-ROOT}/yaml.d`	N

11.5. Tungsten Dashboard Configuration Settings GUI Panel

Configure the Dashboard Settings via the Browser

Version 1.0.9. This feature was first introduced in Tungsten Dashboard version 1.0.9-61

Figure 11.3. Tungsten Dashboard Edit Settings Menu Option

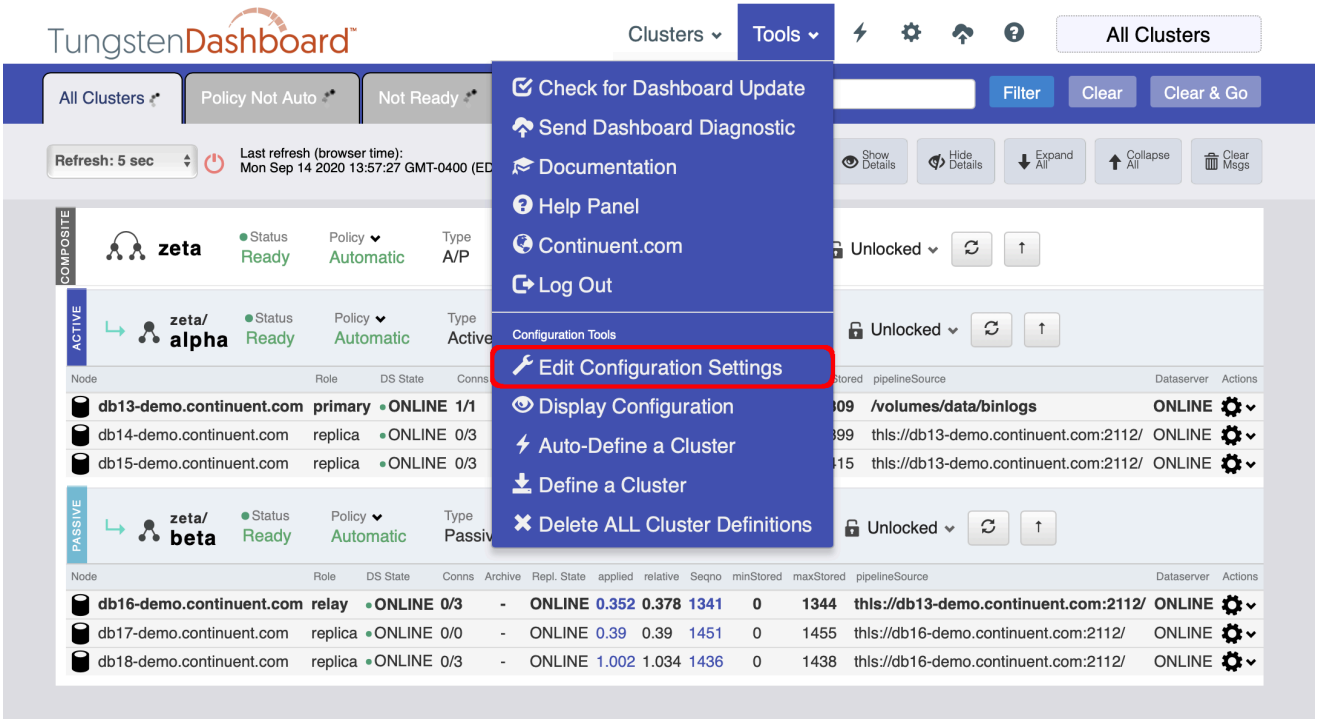
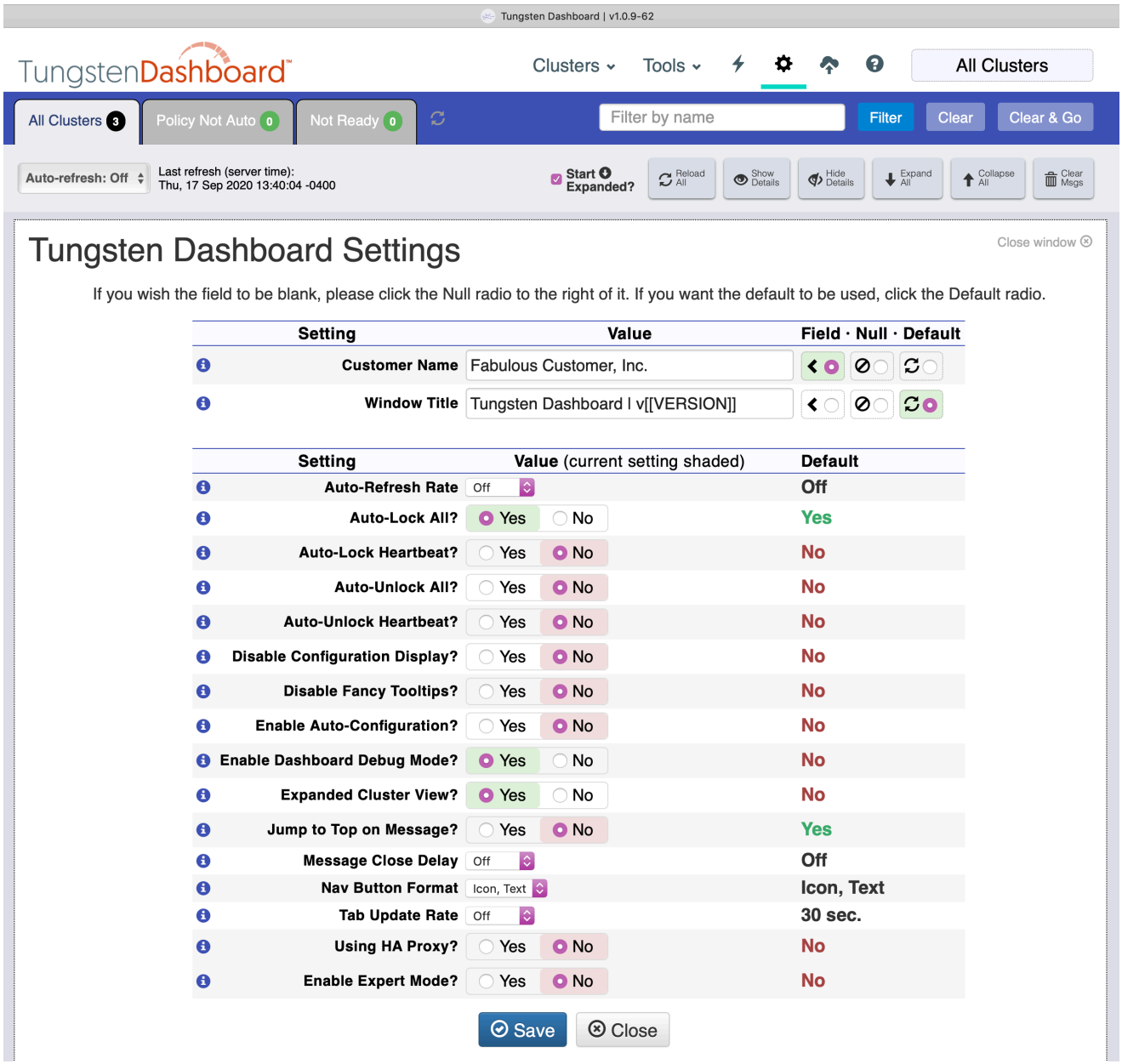


Figure 11.4. Tungsten Dashboard Edit Settings Form



- Update the desired settings then click the "Save" button to update the values on disk.
- Values are stored as single JSON text files in the `{WEBROOT}/settings.d/` subdirectory.
- You can manually edit the files in the `{WEBROOT}/settings.d/` subdirectory. The page will reflect the changes on disk when refreshed.
- Settings Order of Precedence:
  1. Files in `settings.d`
  2. Settings in `config.json`
  3. Coded defaults
- The Edit Settings panel is also available by clicking the appropriate icon in the top tool bar.

11.6. Define a Cluster

Version 1.0.9. This feature was first introduced in Tungsten Dashboard version 1.0.9-61

Prior to version 1.0.9, all cluster definitions had to be located in the `config.php` file in the `clusters` stanza.

You may now configure cluster definitions by hand or via the GUI.

> **Important**
>
> Using the GUI to define clusters is the recommended best practice as of version 1.0.9

A new configuration path `{WEBROOT}/conf.d` can be populated with plain text files in JSON format, one per cluster.

Entries which functioned properly in the `config.php` file will work in the `{WEBROOT}/conf.d` files.

Both the Auto-Define Cluster and Define Cluster GUI worklflows create files in the `{WEBROOT}/conf.d` subdirectory.

11.6.1. Auto-Define a Cluster

Automatically Create the Configuration Definition Files for an Existing Cluster

Version 1.0.9. This feature was first introduced in Tungsten Dashboard version 1.0.9-61

Figure 11.5. Tungsten Dashboard Auto-Define Menu Option

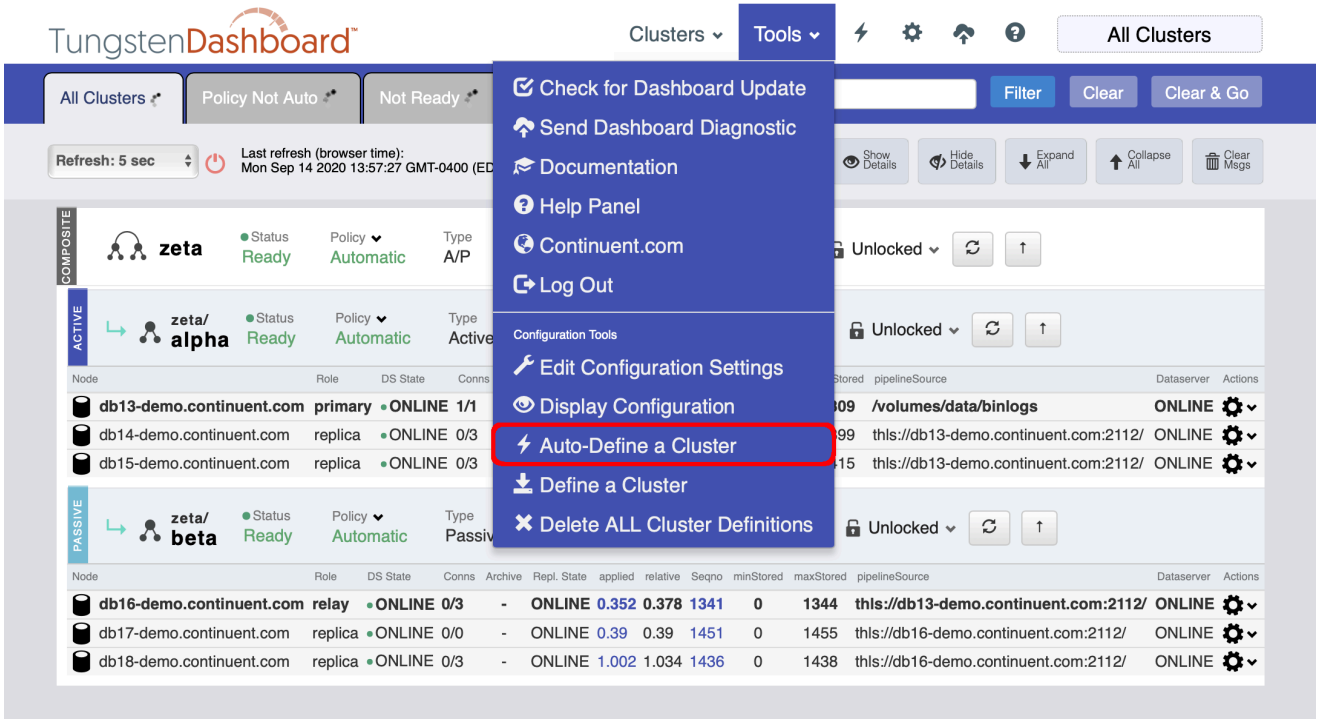
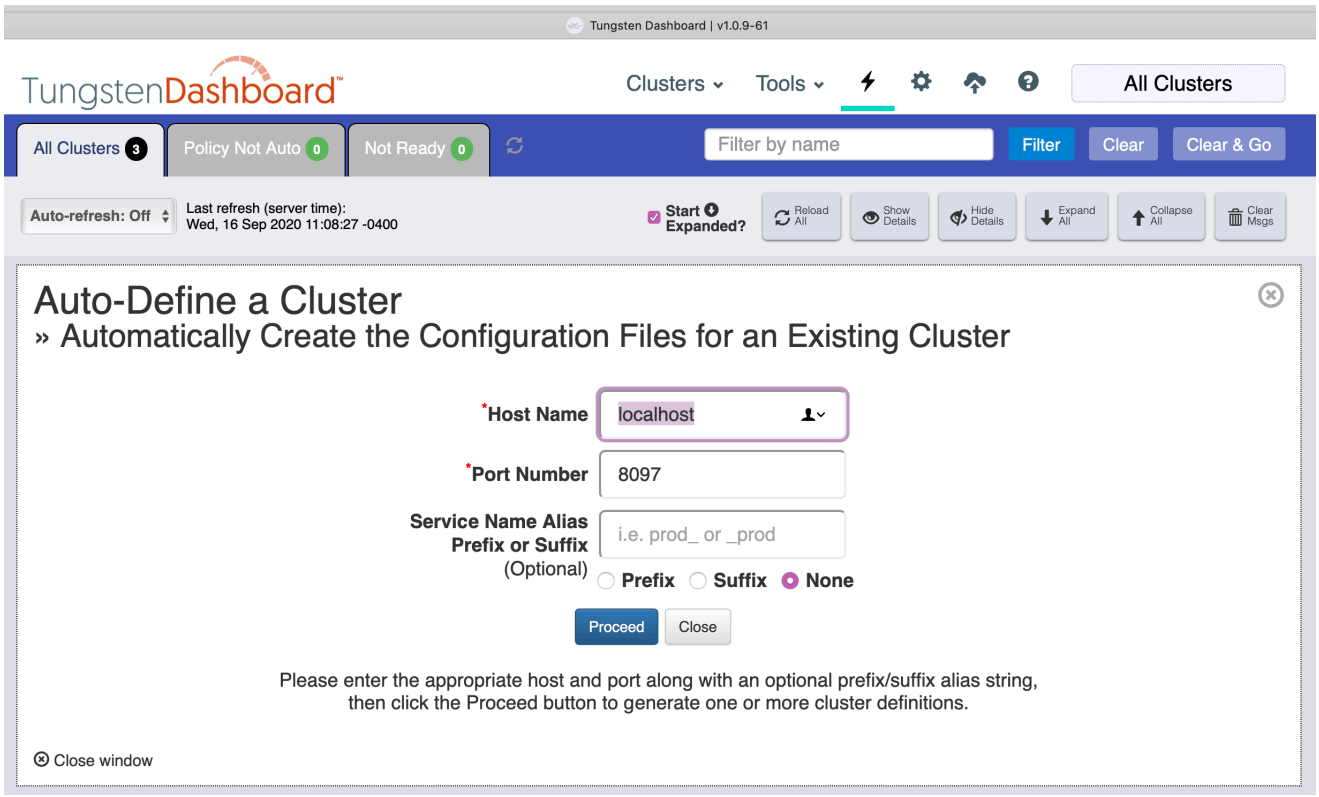


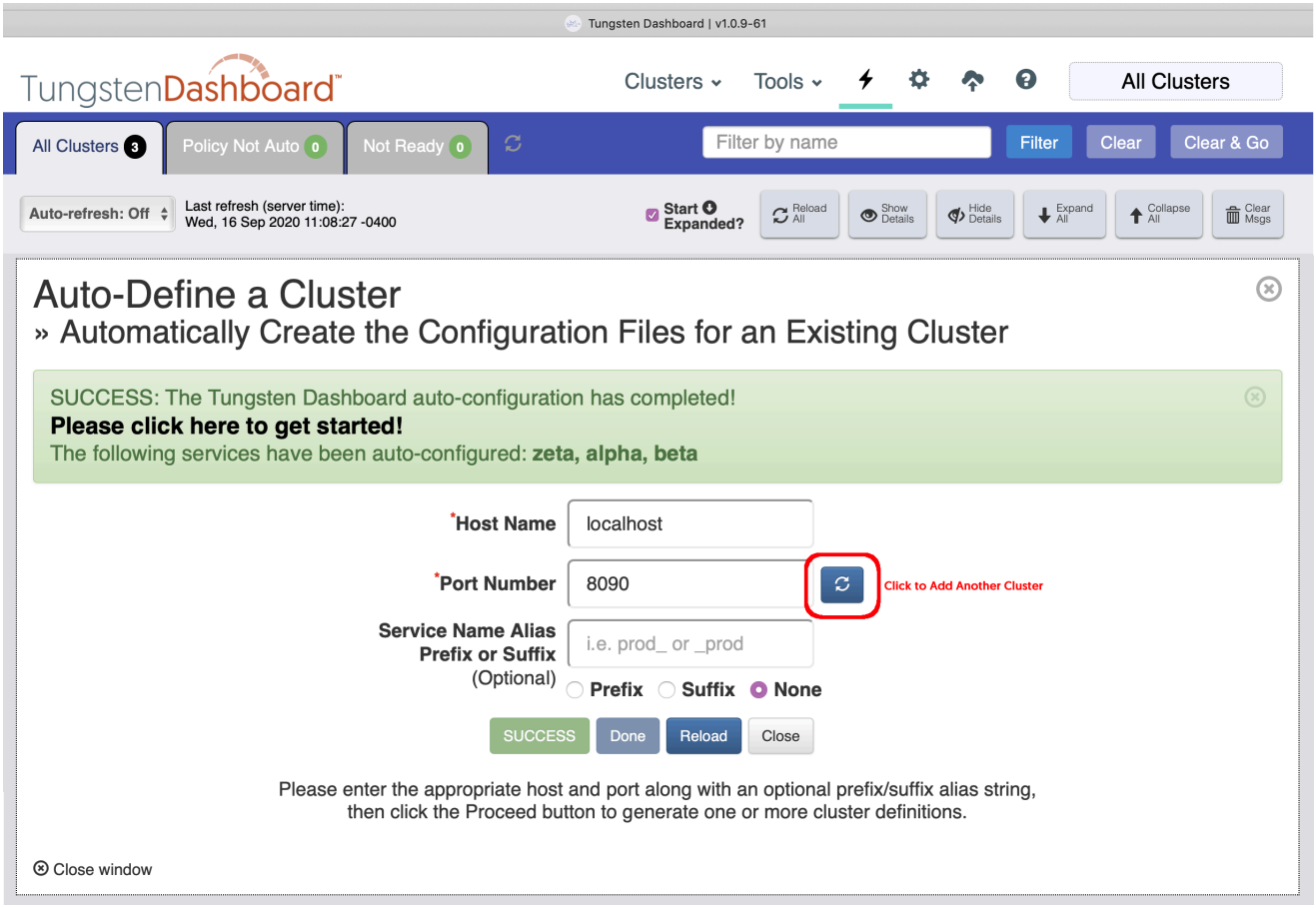
Figure 11.6. Tungsten Dashboard Auto-Define a Cluster Form



- Enter the appropriate host and port along with an optional prefix/suffix alias string, then click the Proceed button to generate one or more cluster definitions.
- When you enter the "Service Name Alias Prefix or Suffix", this value gets added to the beginning or end of the actual service name found.

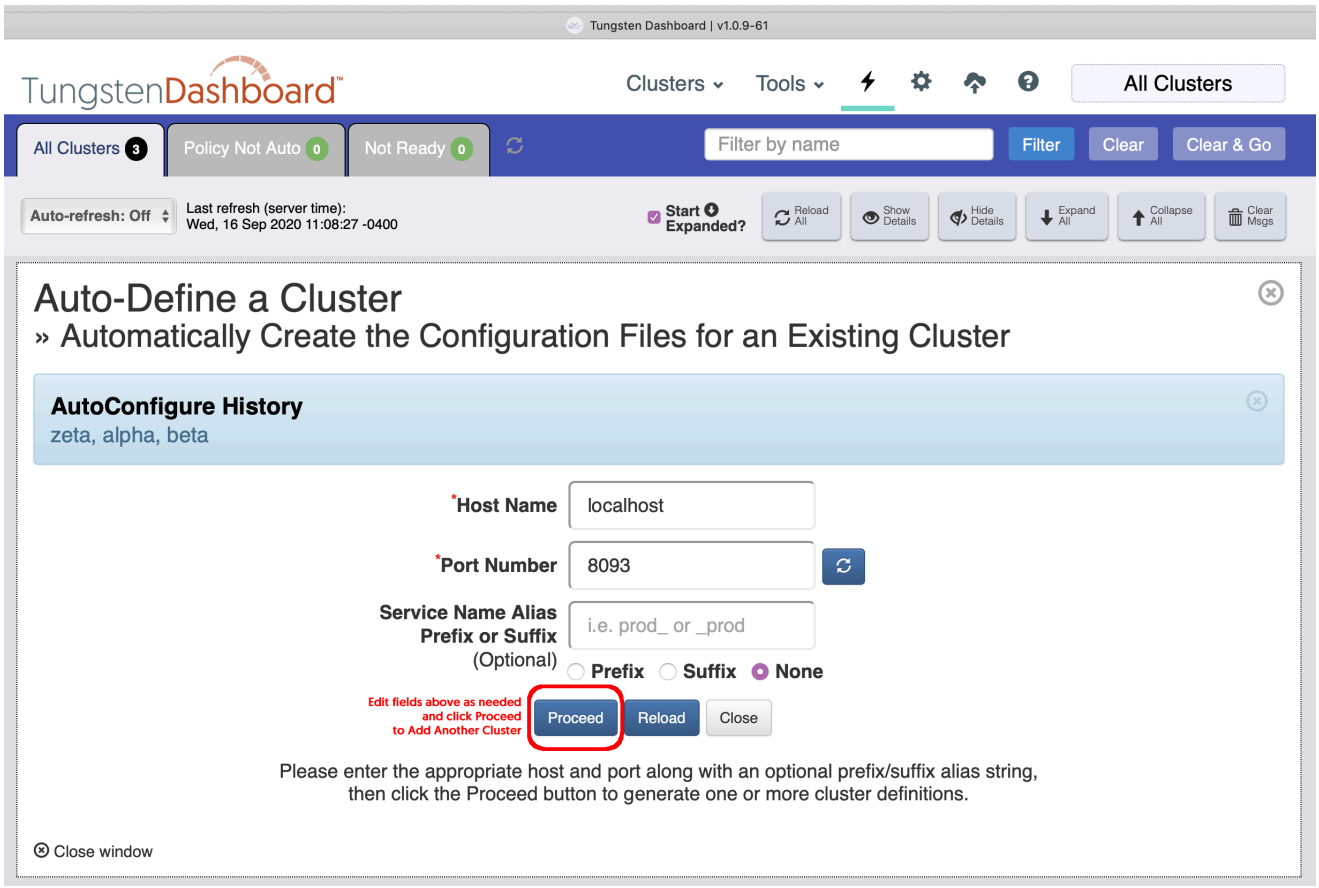
  This is useful if you have more than one cluster defined with the same service name. The Dashboard requires a unique identifier in that case. The addition of a prefix_ or _suffix is usually enough to uniquely identify that cluster.

Figure 11.7. Tungsten Dashboard Auto-Define a Cluster Form Completed



- Once the auto-define finds and configures a cluster, the results will be displayed in green and the Proceed control button will be disabled.
- To easily add another cluster without leaving the form, simply click on the Refresh button to prime the form for another run.

Figure 11.8. Tungsten Dashboard Auto-Define a Cluster Form after the Refresh button has been clicked

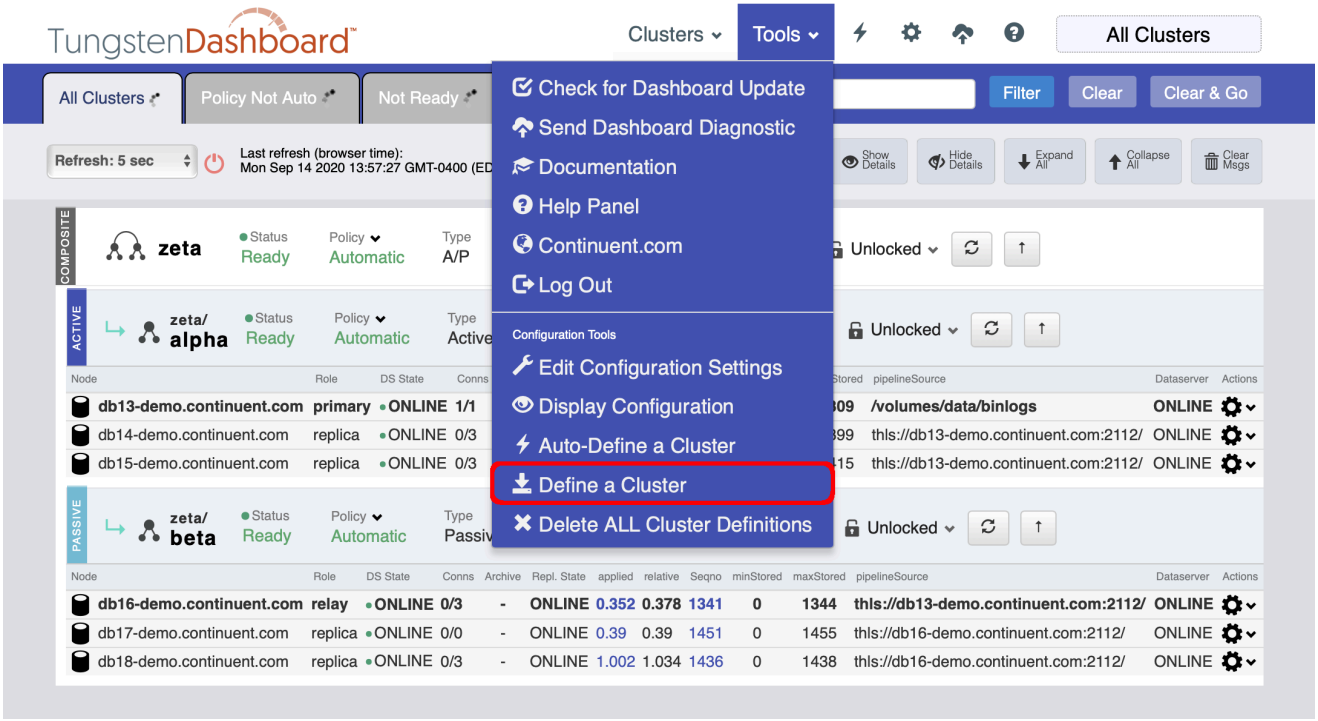


- After clicking the refresh button, the previously-created cluster will be moved into the activity history and the port will be advanced to the next available port.
- Edit the fields as needed, then click the enabled Proceed button again.
- Click Reload to finish and see the resulting clusters!

11.6.2. Define a Cluster via GUI

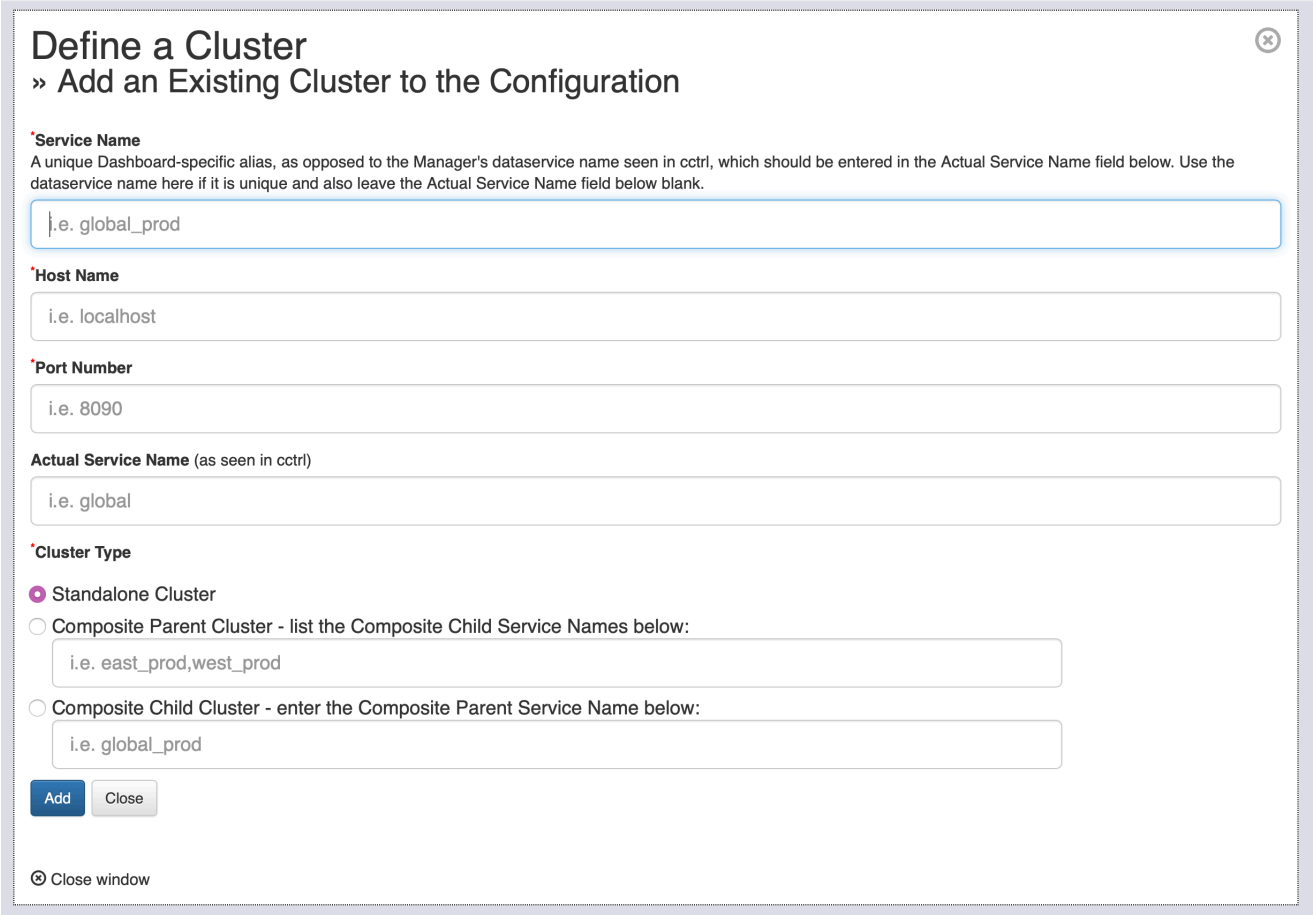
Version 1.0.9. This feature was first introduced in Tungsten Dashboard version 1.0.9-61

Figure 11.9. Define a Cluster Menu Option



- You may choose to define a cluster manually.
- This form will allow you to create cluster definitions in the `{WEBROOT}/conf.d/` subdirectory, one JSON text file per cluster defined.
- You must complete this form once for each cluster to add if you are not using auto-define. If you have a composite cluster with two member clusters, that would imply a total of three definition files, one for the parent composite cluster, and one for each of the member clusters.

Figure 11.10. Define a Cluster Form



- Service Name

  A unique Dashboard-specific alias, as opposed to the Manager's dataservice name seen in cctrl, which should be entered in the Actual Service Name field below. Use the dataservice name here if it is unique and also leave the Actual Service Name field blank.

- Actual Service Name

  The configured service name as seen in cctrl and trepctl.

- Composite Parent Cluster

  Enter the Composite Child Service Names. Use the Dashboard unique alias, not the actual service name.

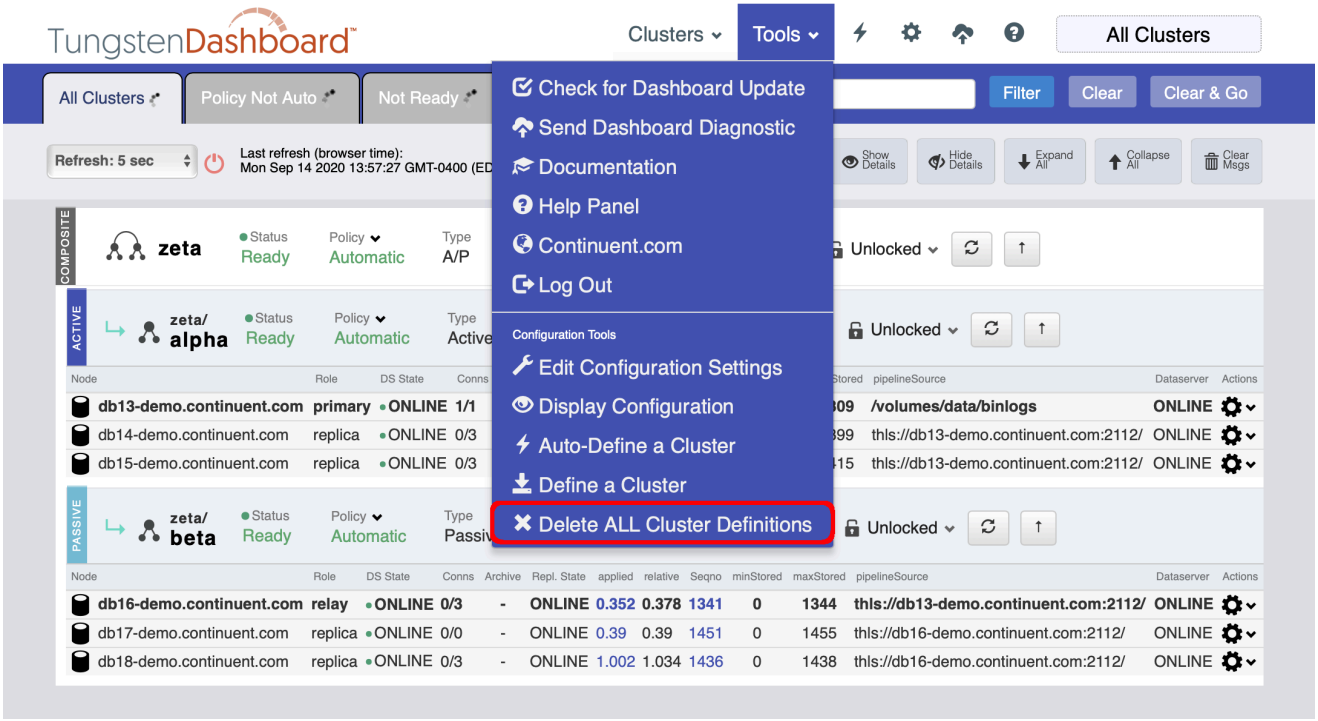
- Composite Child Cluster

  Enter the Composite Parent Service Name. Use the Dashboard unique alias, not the actual service name.

11.6.3. Delete All Cluster Definitions

Version 1.0.9. This feature was first introduced in Tungsten Dashboard version 1.0.9-61

Figure 11.11. Tungsten Dashboard Delete All Cluster Definitions Menu Option



- There are two confirmation prompts before the definitions are removed.
- Once removed, this action cannot be undone. You will have to re-create the definitions.
- Only definitions that are stored as single JSON files in the `{WEBROOT}/conf.d/` subdirectory will be deleted. Clusters defined in the `config.php` file will not be removed.

Figure 11.12. Tungsten Dashboard Delete All Cluster Definitions First Confirmation Prompt

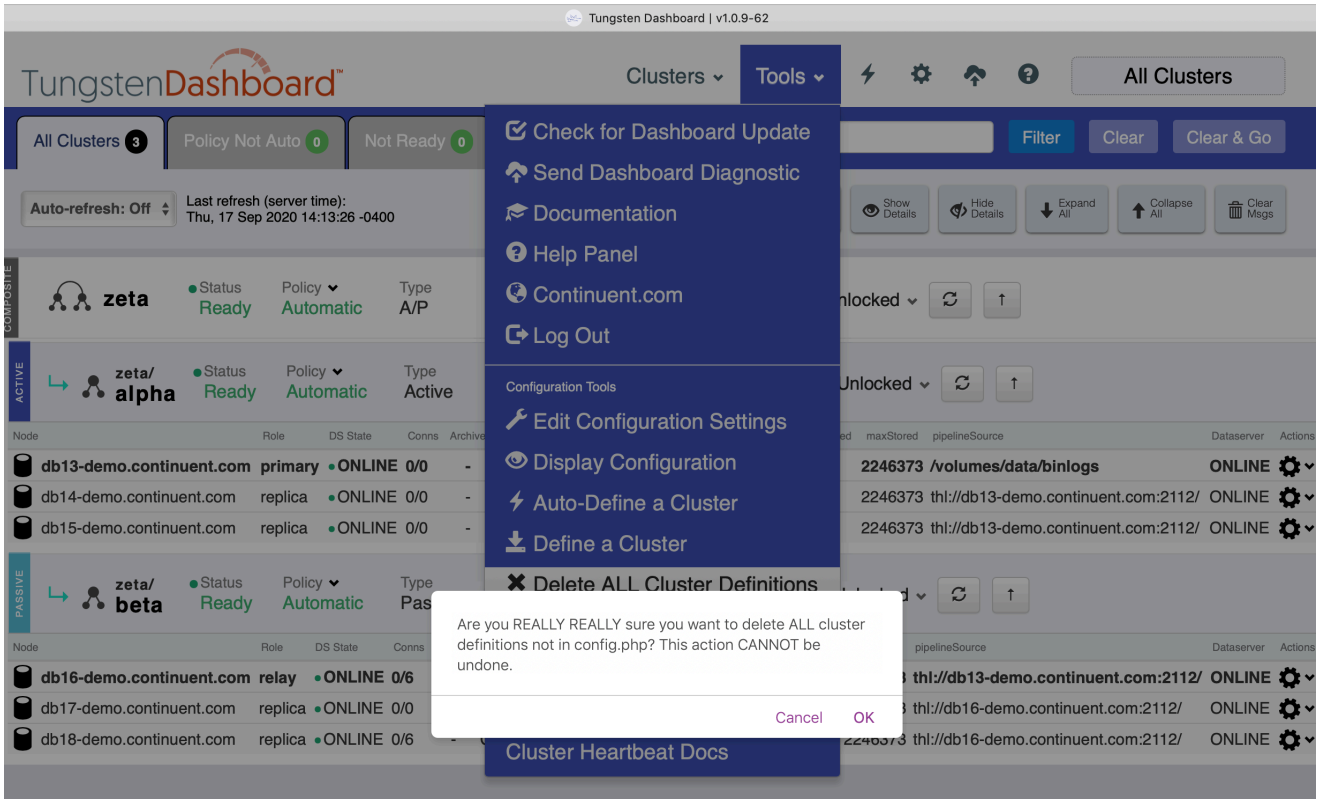


Figure 11.13. Tungsten Dashboard Delete All Cluster Definitions Second Confirmation Prompt

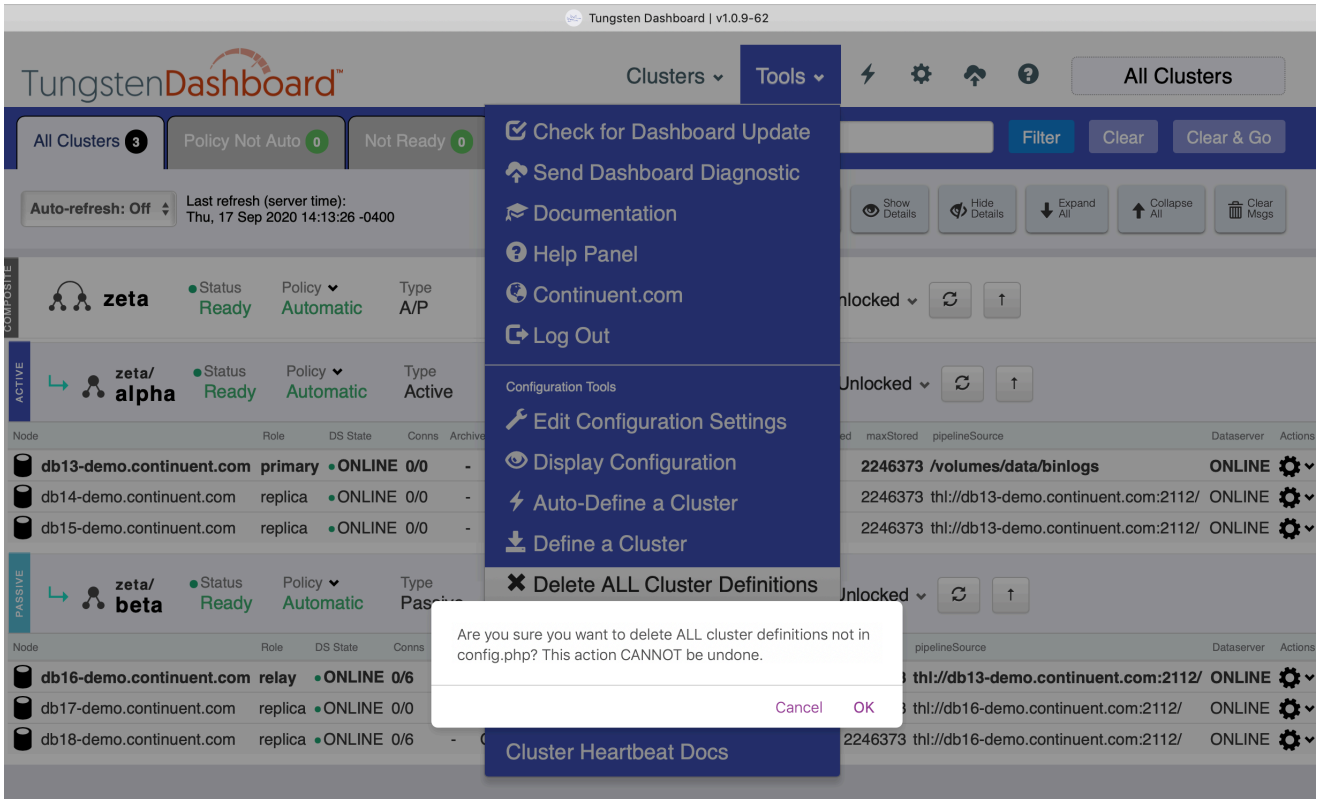
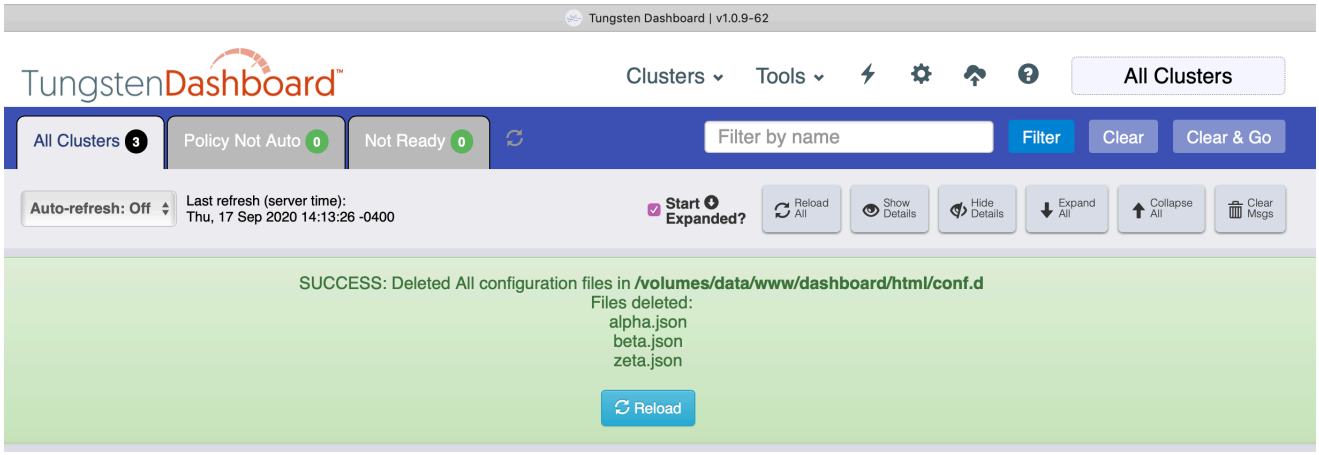


Figure 11.14. Tungsten Dashboard Delete All Cluster Definitions Success



11.6.4. Cluster Definition Configuration Examples

This is a sample standalone cluster configuration from `config.json`:

```
"clusters": {
  "north": {
      "host":"localhost",
      "port":"8203"
  }
},
```

This is a sample composite cluster configuration from `config.json` (either active/passive or active/active):

```
"clusters": {
  "global": {
      "host":"localhost",
      "port":8201,
      "children": [ "west","east" ]
  },
  "east": {
      "host":"localhost",
      "port":"8202",
      "memberOf":"global"
  },
  "west": {
      "host":"localhost",
      "port":"8203",
      "memberOf":"global"
  }
},
```

Chapter 12. Access the Tungsten Dashboard GUI via a browser

Access the Tungsten Dashboard GUI via a browser:

```
Browser URL: http://dashboard.yourdomain.com/
```

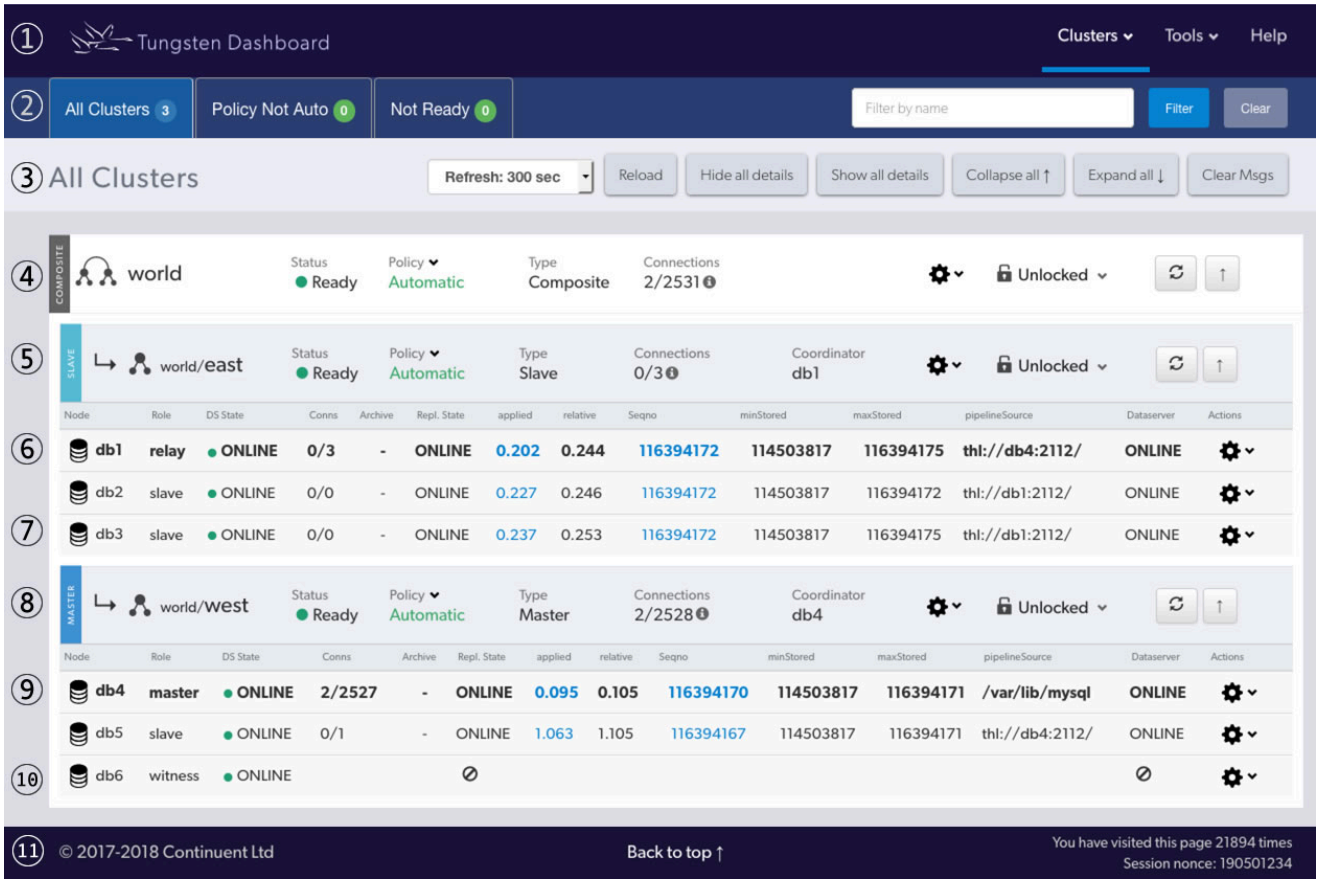
Chapter 13. Tungsten Dashboard User Interface

This section describes all of the features and functionality available in our browser-based Graphical User Interface.

13.1. Tungsten Dashboard User Interface Overview

Below is a sample of how the Dashboard would look for a Composite cluster called `world` with two 3-node member clusters, called `east` and `west`:

Figure 13.1. Tungsten Dashboard User Interface



1. Navigation Bar One
2. Navigation Bar Two
3. Navigation Bar Three
4. Example Composite cluster parent `world` summary row with controls
5. Example Composite cluster member `east` summary row with controls
6. Example cluster *relay* node `db1` summary row with controls
7. Example cluster *Replica* node `db3` summary row with controls
8. Example Composite cluster member `west` summary row with controls
9. Example cluster *Primary* node `db4` summary row with controls
10. Example cluster *witness* node `db6` summary row with controls
11. Footer with copyright, back-to-top link, visit count and session id

13.2. Dashboard Navigation Bar One

Nav Bar One is the first horizontal bar across the top of the window.

Figure 13.2. Example Navigation Bar One

1. Logo and site title - click either to return to the home page (full page load)
2. Clusters menu - All clusters configured will be displayed in a hierachical view. Click on any onw to limit the view to that cluster. If you select a Composite cluster, the parent and all member clusters will show.
3. Tools menu - various links to outside resources. Custom links may be added here via the `config.json` file in the web root directory.
4. Help feature - click to reveal helpful information.

13.3. Dashboard Navigation Bar Two

Nav Bar Two is the second horizontal bar across the top of the window.

Figure 13.3. Example Navigation Bar Two

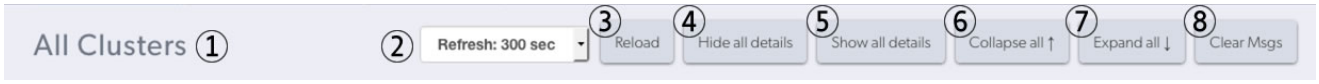
The badges for "Policy Not Auto" and "Not Ready" tabs are auto-updated via AJAX every 30 seconds independently of the Auto-Refresh setting on Navigation Bar Two.

1. All Clusters Tab - click to see all available clusters, same as clicking logo and site title (full page load)
2. Policy Not Auto Tab - click to see all only those clusters where the policy is set to other than *`AUTOMATIC`*
3. Not Ready Tab - click to see only clusters that are not in the Ready state
4. Filtering feature - enter a value to search for in the cluster name. The search is case in-sensitive and has automatic wildcards on both sides of the string. Click on the Clear button to empty out the filter field.

13.4. Dashboard Navigation Bar Three

Nav Bar Three is the third horizontal bar across the top of the window.

Figure 13.4. Example Navigation Bar Three

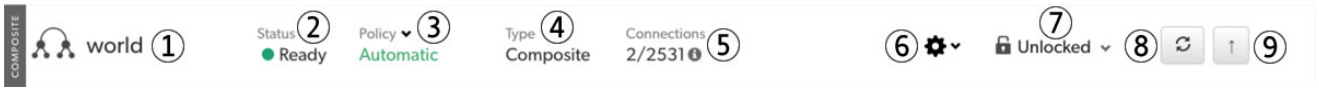


1. Content title - shows current view or filter in use
2. Auto-refresh feature - select a refresh rate of 0 (off), 5, 10, 30, 60, 120 or 300 seconds. This will enable AJAX-based reloads of the clusters in the content section without reloading the entire page. Look for the spinner in the refresh button per cluster when the refresh is triggered.
3. Reload button - same as clicking the top logo (full page load)
4. Hide All Details button - each database node is expandable to display all available details. This button closes them all.
5. Show All Details button - each database node is expandable to display all available details. This button opens them all.
6. Collapse All button - each Composite cluster is expandable to display all available node rows. This button closes them all.
7. Expand All button - each Composite cluster is expandable to display all available node rows. This button opens them all.
8. Clear Messages button - dismiss all messages that are showing at the top of the screen.

13.5. Dashboard Composite Parent Row

A composite Parent row contains controls for the entire Composite cluster.

Figure 13.5. Example Composite Parent Row



1. Cluster type `composite` vertical tag, resource icon and parent cluster name
2. Composite cluster status. The color will change based on the status. Status will be one of: Ready, Warning, or Error
3. Cluster Policy. One of: *`AUTOMATIC`*, *`MAINTENANCE`* or *`MIXED`*. There is a state-sensitive dropdown menu to allow the Policy to be changed.
4. Cluster type - one of: Standalone, Composite Active/Active (CAA) or Composite Active/Passive (CAP). Standalone has no composite child clusters. This field is a duplicate of the vertical tag at the start of field (1), above.
5. Connections - display the total number of active connections from all Connectors to all nodes in this specific cluster. If you hover over the info icon, you can see the full breakdown by node.
6. Composite actions dropdown menu - these are the same commands available when using cctrl -multi followed by use {composite_service_name_here}, i.e.:

```
shell> cctrl -multi
[LOGICAL] / > use world
[LOGICAL] /world > {your_selected_command_here}
```

   - Heartbeat (actually cluster heartbeat)
   - Recover
   - Switch - only available for CMS clusters
   - Failover - only available for CMS clusters
7. Locking status text and icon with dropdown menu to allow lock control.

   Important

   Under normal circumstances, you should not need to get a lock, since all operations automatically attempt to obtain a lock for efficiency purposes.
8. Refresh button - triggers an AJAX refresh of the parent cluster and all member clusters including all node rows. (no page load)
9. Collapse all in Composite cluster - hide node rows for all member clusters in this Composite.

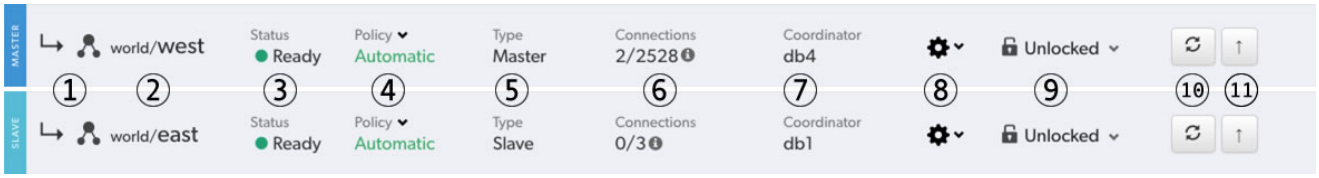
13.6. Dashboard Composite Member Rows

A composite member row contains controls for all nodes in the member cluster.

Member clusters may have either the Active or Passive role.

There will be only one Active member cluster and any number of Passive member clusters.

Figure 13.6. Example Composite Member Rows



1. Cluster type vertical tag (`Primary` or `Replica`), member cluster right-arrow indicator and cluster resource icon
2. Cluster parent service name followed by the cluster service name

3. Composite member cluster status. The color will change based on the status. Status will be one of: Ready, Warning, or Error
4. Cluster Policy. One of: *AUTOMATIC*, *MAINTENANCE* or *MIXED*. There is a state-sensitive dropdown menu to allow the Policy to be changed.
5. Cluster type - one of: Standalone, Composite, Active or Passive. Active and Passive both imply Composite membership. Standalone has no composite membership. This field is a duplicate of the vertical tag at the start of field [1], above.
6. Connections - display the total number of active connections from all Connectors to all nodes in the entire Composite cluster. If you hover over the info icon, you can see the full breakdown by node.
7. Coordinator - display the host which currently has the `coordinator` role for the member cluster. Every cluster designates one of the Tungsten Managers in the cluster as the coordinator and it is this Manager that will be responsible for taking action, if action is required, to recover the cluster's database resources to the most highly available state possible.
8. Cluster actions dropdown menu - there are three distinct types of choices in this dropdown menu
   - UI-Specific
     - Toggle Details - show or hide the node details for all nodes in the member cluster
   - Cluster-level commands

     These are the same commands available when using cctrl, i.e.:

     ```
     shell> cctrl
     [LOGICAL] /east > {your_selected_command_here}
     ```

     > **Note**
     >
     > The cluster service name displayed will be the service name of the node you are logged into.

     - Heartbeat
     - Recover
     - Switch
     - Failover
   - Composite datasource-level commands

     These are the same commands available when using cctrl -multi followed by use {composite_service_name_here}, i.e.:

     ```
     shell> cctrl -multi
     [LOGICAL] / > use world
     [LOGICAL] /world > datasource {cluster_member_service_here} {your_selected_command_here}
     ```

     Here are some individual examples:

     ```
     [LOGICAL] /world > datasource east recover
     [LOGICAL] /world > datasource west fail
     [LOGICAL] /world > switch to west
     ```

     - Recover
     - Welcome
     - Online
     - Offline
     - Shun
     - Promote - this is the same as doing a switch to {cluster_member_service_here}
     - Fail
9. Locking status text and icon with dropdown menu to allow lock control.

   > **Important**
   >
   > *Under normal circumstances, you should not need to get a lock, since all operations automatically attempt to obtain a lock for efficiency purposes.*

10. Refresh - triggers an AJAX refresh of that member cluster only (no page load)
11. Collapse - hide the node rows for that member cluster only

13.7. Dashboard Composite Member Node Rows

A node row contains controls for that one specific cluster node.

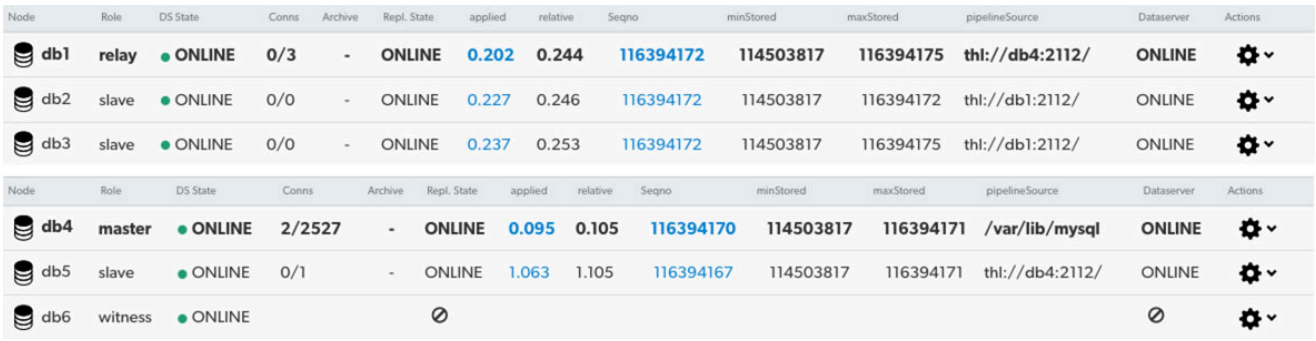
Cluster nodes may have one of the following roles: Primary, Replica, Witness or Standby. Composite member cluster nodes may also have the Relay role.

For any cluster, there will be only one Primary/Relay cluster node and any number of Replica nodes.

A Cluster Primary node is assigned the special role of Relay when it is part of a Composite Passive cluster.

Active witness nodes do not have a database and therefore do not run a replicator. Passive witness nodes do not appear because they have no Manager process running.

Figure 13.7. Example Composite Member Node Rows



- Node - the hostname of the server
- Role - one of Primary, Relay, Replica, Standby or Witness
- DS State - DataSource state can be `ONLINE`, `OFFLINE`, SHUNNED or FAILED. There may be other, less-used values.
- Conns - number of active connections / total number of connections created since last restart
- Archive - has Archive mode been enabled? See Mark a Datasource as Archive for more information.
- Repl. State - the state of the Replicator process, one of: `ONLINE`, `OFFLINE` or ERROR
- applied - the *`appliedLatency`* value, which is how long it took to actually get the event either extracted from the Primaries binary logs or applied into the Replicas target database
- relative - the *`relativeLatency`* value, which is how long it has been since we performed an action
- Seqno - the *`appliedLastSeqno`* value
- minStored - the *`minimumStoredSeqNo`* value, which is the sequence number of the oldest event stored in the THL
- maxStored - the *`maximumStoredSeqNo`* value, which is the sequence number of the latest event to be stored in the THL
- *`pipelineSource`* - the protocol, host and port where the replicator is pulling THL from
- Dataserver - the state of the database server, one of `ONLINE`, `OFFLINE` or UNKNOWN
- Actions - the node-specifc commands dropdown menu. There are four distinct types of choices in this dropdown menu.
  - UI-Specific
    - For all nodes that have a running Replicator, the installed Tungsten version will be the first item visible.
    - Toggle Details - show or hide the node details for that specific node
  - DataSource (Node-level) Commands

These are the same commands available when using cctrl, i.e.:

```
shell> cctrl
[LOGICAL] /east > datasource {node_hostname_here} {your_selected_command_here}
```

> **Note**
>
> The cluster service name displayed will be the service name of the node you are logged into.

- Recover
- Welcome
- Offline - only appears if the DataSource is in the `ONLINE` state
- Online - only appears if the DataSource is in the `OFFLINE` state
- Fail

- Replicator-specific DataSource (Node-level) Commands

  These are the same commands available when using cctrl, i.e.:

  ```
  shell> cctrl
  [LOGICAL] /east > replicator {node_hostname_here} {your_selected_command_here}
  ```

  Here are some individual examples:

  ```
  [LOGICAL] /world > replicator db1 online
  [LOGICAL] /world > replicator db3 offline
  ```

  - Offline - only appears if the Replicator is in the `ONLINE` state
  - Online - only appears if the Replicator is in the `OFFLINE` state

- Replica-specific DataSource (Node-level) Commands

  > **Important**
  >
  > These are commands are ONLY available on a node with the Replica or Standby roles. Nodes with either Primary, Relay or Witness roles will not display the Replica-specific menu options.

  These are the same commands available when using cctrl, i.e.:

  ```
  shell> cctrl
  [LOGICAL] /east > datasource {node_hostname_here} {your_selected_command_here}
  ```

  Here are some individual examples:

  ```
  [LOGICAL] /world > datasource db1 shun
  [LOGICAL] /world > datasource db3 recover
  [LOGICAL] /world > switch to db2
  ```

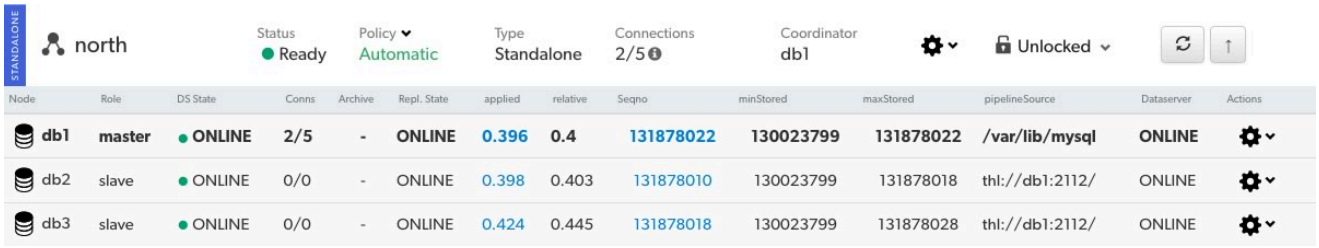
  - Backup
  - Restore
  - Shun
  - Enable Standby
  - Disable Standby
  - Promote - this is the same as doing a switch to {node_hostname_here}

13.8. Dashboard Standalone Cluster

All of the controls and information are the same for Standalone clusters and nodes as they are for Composite with the following exceptions:

- A Standalone Cluster is not part of a Composite.
- There will be no Composite commands in the service-level dropdown menu.

Figure 13.8. Example Standalone Cluster



Chapter 14. Send a Dashboard Diagnostic to Support

Upload a Dashboard Diagnostic Package to Continuent Support

Version 1.0.9. This feature was first introduced in Tungsten Dashboard version 1.0.9-61

Figure 14.1. Tungsten Dashboard Send Diagnostic Menu Option

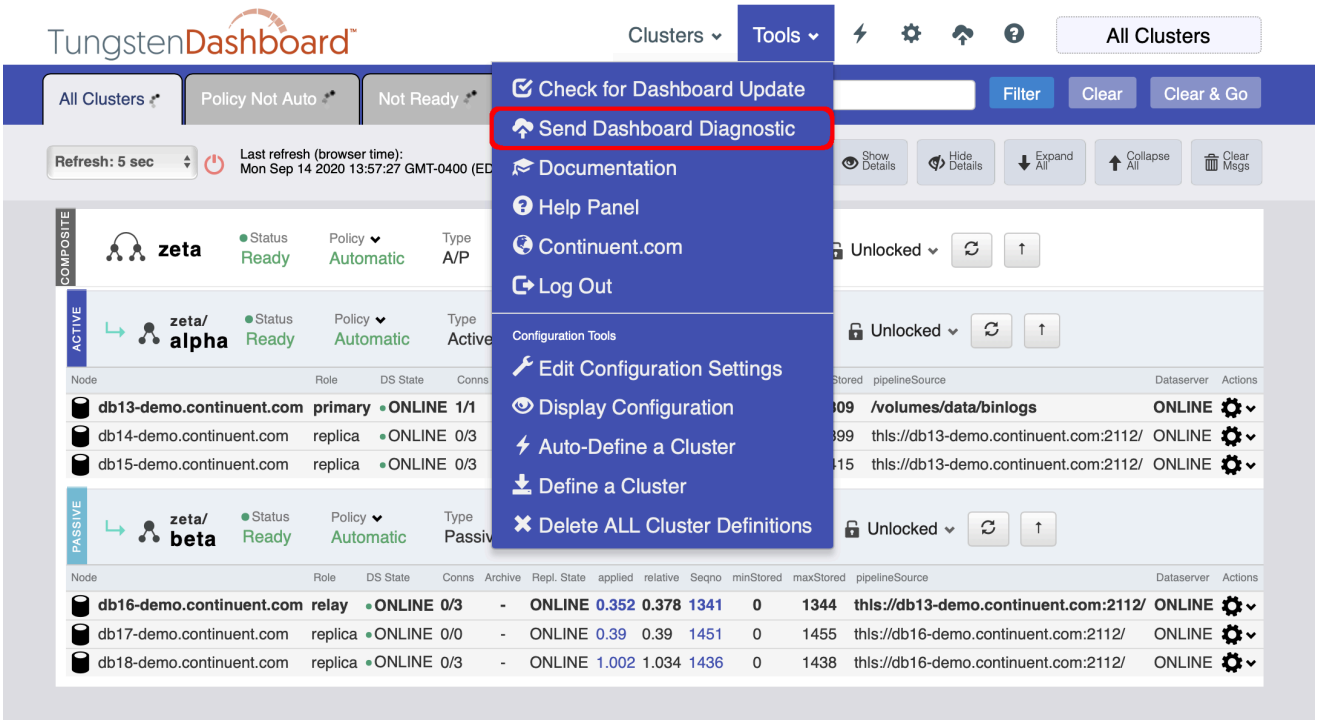
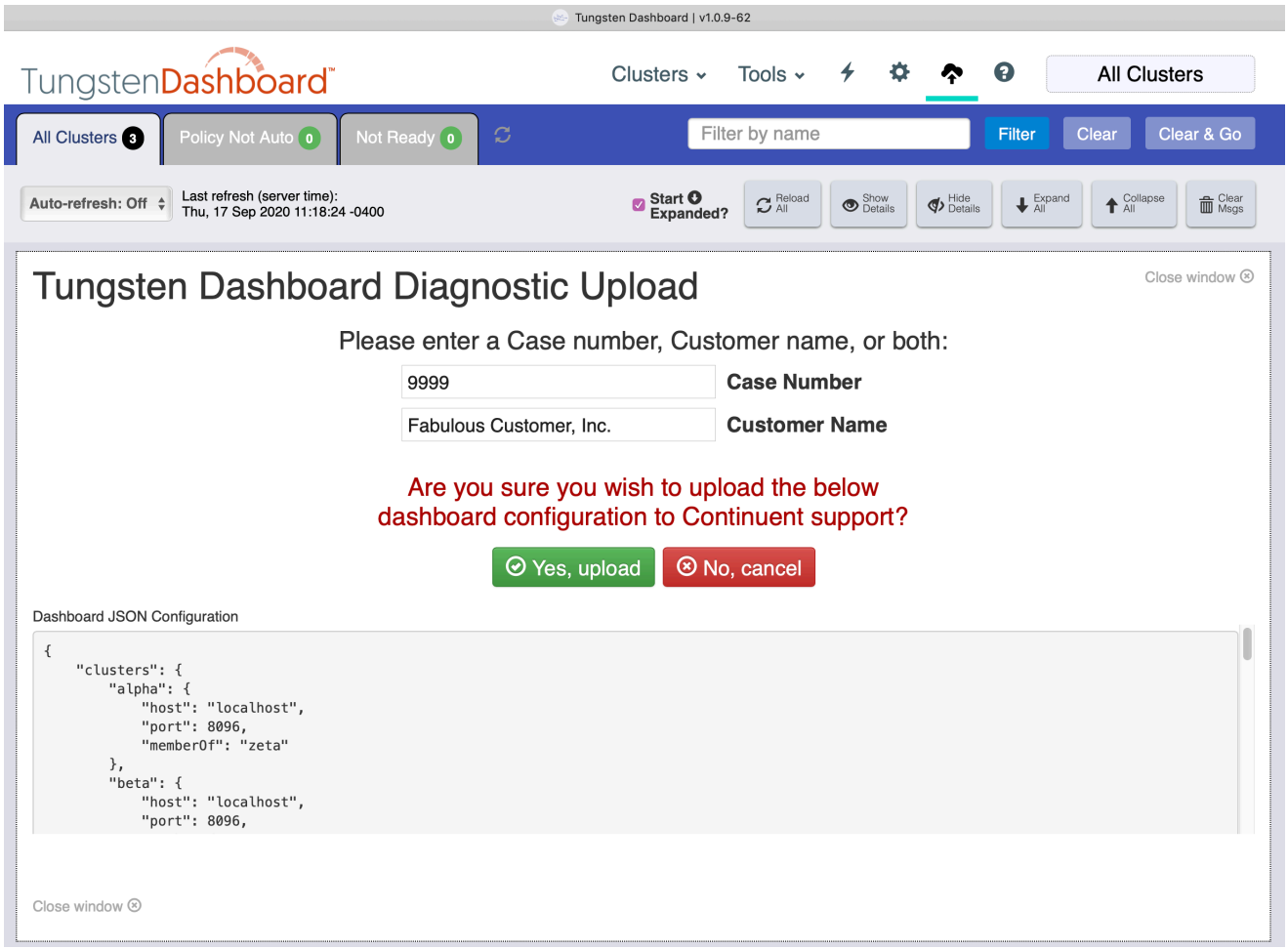


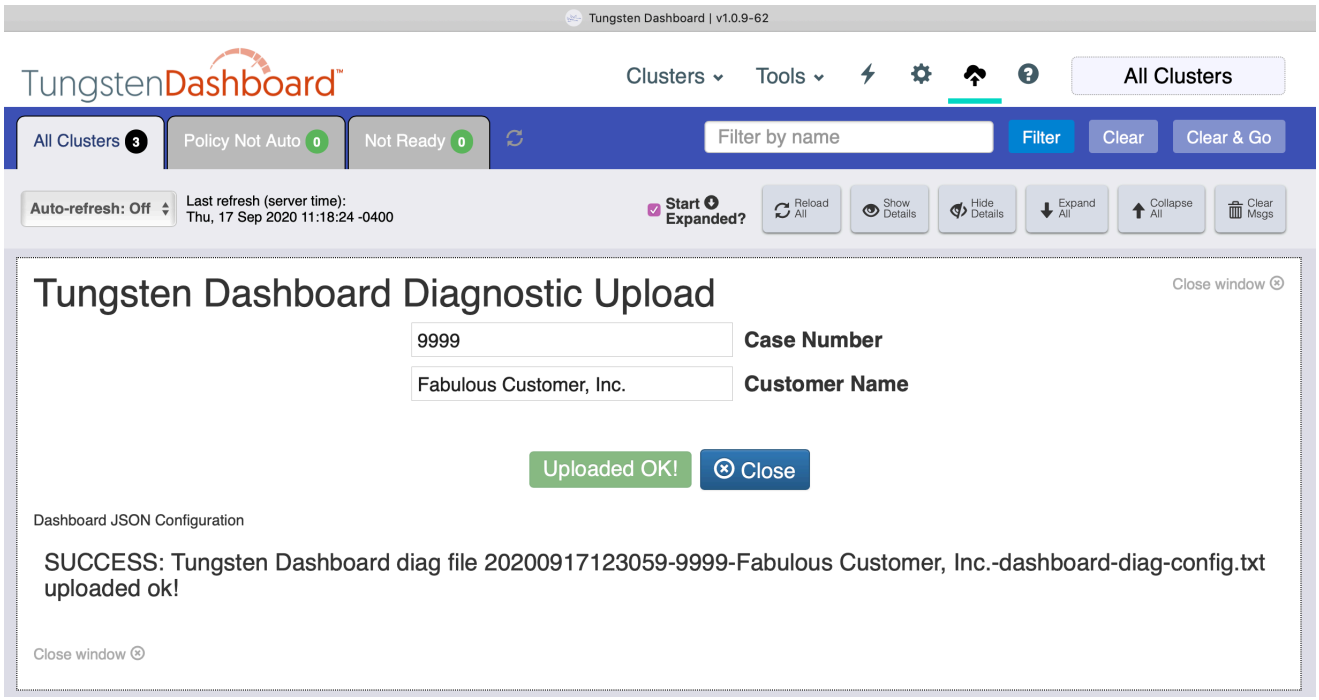
Figure 14.2. Tungsten Dashboard Send Diagnostic Form



- Enter the appropriate case number and customer name then click the "Yes, Upload" button to send the diagnostic to Support.
- The diagnostic contains the JSON configuration of the dashboard.

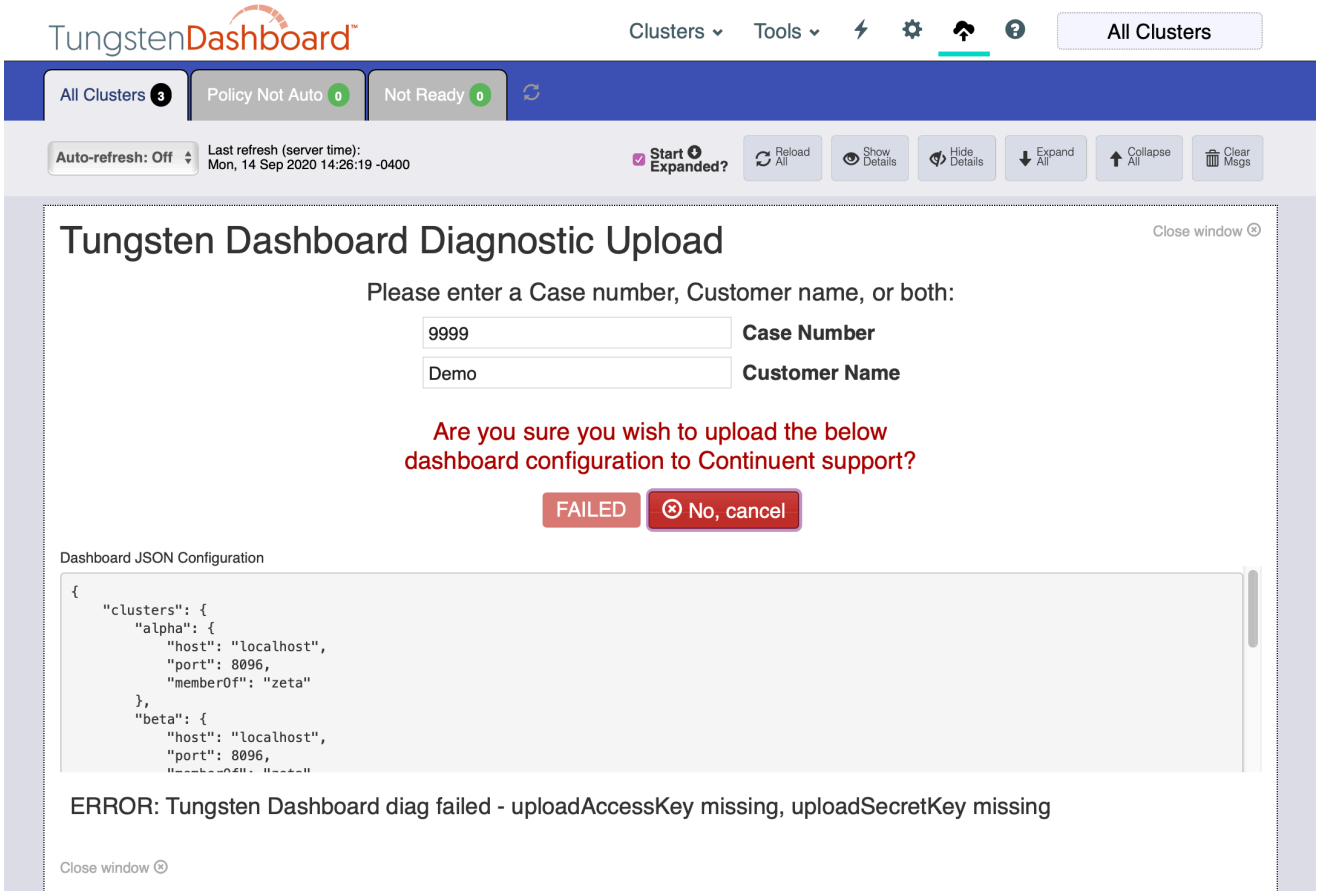
  This JSON text is uploaded to Continuent Support's protected AWS bucket. No other customer has access to this location, it is upload-only.
- The Diagnostic feature is also available by clicking the appropriate icon in the top tool bar.

Figure 14.3. Tungsten Dashboard Send Diagnostic Success



If you do not have the proper upload-specific access and secret keys configured, you would see an error like this:

Figure 14.4. Tungsten Dashboard Send Diagnostic Failure Due to Missing Keys



Chapter 15. Monitoring Tungsten Clusters Using Prometheus and Grafana

Tungsten Dashboard has introduced basic support for using Prometheus and Grafana to monitor Tungsten Clusters. As of Tungsten Clustering software v6.1.4, key Prometheus exporters have been added to the distribution. These exporters will allow a Prometheus server to gather metrics for:

- the MySQL server and the underlying node "hardware" using external binaries added to the distribution
- the Tungsten Manager, Replicator and Connector using new built-in functionality

A new script has been included to assist with the management and testing of the exporters called tmonitor.

To learn more about the tmonitor command and the included exporters, please visit Monitoring Status Using Prometheus Exporters for more information.

> IMPORTANT: To get the most benefit out of the exporters along with ensuring both ease of configuration and security, Continuent requires that both the Prometheus and Grafana servers be installed onto the same instance hosting the Dashboard web server when using the Prometheus and Grafana integration with Dashboard.

15.1. Monitoring Tungsten Clusters Using Prometheus

The below example procedure is designed to help you get Prometheus installed and working with the goal of monitoring Tungsten Clusters through the Dashboard.

This section of the documentation is a summary guide for how to install an external software product, Prometheus. The usual caveats apply, and as always, your mileage may vary.

For more information about getting started with Prometheus, please visit the Prometheus website at https://prometheus.io/docs/introduction/first_steps/

15.1.1. Example Prometheus Installation Procedure

First, download the tarball from https://prometheus.io/download/

Next, go to the install directory, normally `/usr/local`, and extract the tarball. Complete the Prometheus software installation by creating a symbolic link for convenience when upgrading.

```
shell> cd /usr/local
shell> sudo tar xvzf {tarball_fullpath_here}
shell> sudo ln -s {extracted_dir} prometheus
```

In this step, create the Prometheus user and directories you will need, along with setting proper ownership. Be sure to modify the examples to match your environment.

```
shell> sudo useradd -rs /bin/false prometheus
shell> sudo mkdir -p ~prometheus/data/
shell> sudo mkdir -p ~prometheus/var/
shell> sudo touch ~prometheus/var/prometheus.log
shell> sudo chown -R prometheus: /usr/local/prometheus* ~prometheus
```

15.1.2. Example Prometheus Configuration Procedure

The below example shows three (3) 3-node clusters for a total of nine (9) nodes.

Each node has 5 available exporters (name:port) - node:9400, mysqld:9404, replicator:8091, manager:8092 and connector:8093.

Create anew or edit the existing Prometheus configuration file, normally `/usr/local/prometheus/prometheus.yml`, and adjust the file to match your specific needs.

```
shell> sudo vi /usr/local/prometheus/prometheus.yml

# sample config for monitoring Tungsten Clusters
global:
  scrape_interval:     15s # Set the scrape interval to every 15 seconds. Default is every 1 minute.
  evaluation_interval: 15s # Evaluate rules every 15 seconds. The default is every 1 minute.
  # scrape_timeout is set to the global default (10s).
```

```
# Alertmanager configuration
alerting:
  alertmanagers:
  - static_configs:
    - targets:
      # - alertmanager:9093

# Load rules once and periodically evaluate them according to the global 'evaluation_interval'.
rule_files:
  # - "first_rules.yml"
  # - "second_rules.yml"

# A scrape configuration containing exactly one endpoint to scrape:
# Here it's Prometheus itself.
scrape_configs:
# The job name is added as a label `job=<job_name>` to any timeseries scraped from this config.
- job_name: 'prometheus'

  # metrics_path defaults to '/metrics'
  # scheme defaults to 'http'.

  static_configs:
  - targets: ['localhost:9090']

- job_name: 'node'
  scrape_interval: 5s
  static_configs:
    - targets: ['db1:9100','db2:9100','db3:9100','db4:9100','db5:9100','db6:9100','db7:9100','db8:9100','db9:9100']

- job_name: 'mysqld'
  scrape_interval: 5s
  static_configs:
    - targets: ['db1:9104','db2:9104','db3:9104','db4:9104','db5:9104','db6:9104','db7:9104','db8:9104','db9:9104']


- job_name: 'tungsten_replicator'
  scrape_interval: 5s
  static_configs:
  - targets: ["db1:8091", "db2:8091", "db3:8091", "db4:8091", "db5:8091", "db6:8091",'db7:8091','db8:8091','db9:8091']

- job_name: 'tungsten_manager'
  scrape_interval: 5s
  static_configs:
  - targets: ["db1:8092", "db2:8092", "db3:8092", "db4:8092", "db5:8092", "db6:8092",'db7:8092','db8:8092','db9:8092']

- job_name: 'tungsten_connector'
  scrape_interval: 5s
  static_configs:
  - targets: ["db1:8093", "db2:8093", "db3:8093", "db4:8093", "db5:8093", "db6:8093",'db7:8093','db8:8093','db9:8093']
```

15.1.3. Example Prometheus Boot Configuration Procedures

- init.d-based procedure

  Create the `prometheus` boot script for init.d:

```
shell> sudo vi /etc/init.d/prometheus

#!/bin/bash
#
# /etc/rc.d/init.d/prometheus
#
# Prometheus monitoring server
#
#  chkconfig: 2345 20 80 Read
#  description: Prometheus monitoring server
#  processname: prometheus

# Source function library.
. /etc/rc.d/init.d/functions

PROGNAME=prometheus
RETENTION=3d
HOMEDIR="/home"
INSTALLDIR="/usr/local"

PROG=$INSTALLDIR/$PROGNAME/$PROGNAME
CONFIG_FILE=$INSTALLDIR/$PROGNAME/$PROGNAME.yml

USER=$PROGNAME
```

```
DATADIR=$HOMEDIR/$USER/data
LOGFILE=$HOMEDIR/$USER/var/$PROGNAME.log
LOCKFILE=$HOMEDIR/$USER/var/$PROGNAME.pid

start() {
  echo -n "Starting $PROGNAME: "
  daemon --user $USER --pidfile="$LOCKFILE" "$PROG --config.file=$CONFIG_FILE --storage.tsdb.path=$DATADIR --storage.tsdb.retention=$RETENTION --web.enable-admin-a
  echo $(pidofproc $PROGNAME) >$LOCKFILE
  echo
}

stop() {
  echo -n "Shutting down $PROGNAME: "
  killproc $PROGNAME
  rm -f $LOCKFILE
  echo
}


case "$1" in
  start)
  start
  ;;
  stop)
  stop
  ;;
  status)
  status $PROGNAME
  ;;
  restart)
  stop
  start
  ;;
  reload)
  echo "Sending SIGHUP to $PROGNAME"
  kill -SIGHUP $(pidofproc $PROGNAME)
  ;;
  *)
      echo "Usage: <servicename> {start|stop|status|reload|restart}"
      exit 1
  ;;
esac
```

Enable the `prometheus` service to start at boot time via chkconfig, and then start it using service:

```
shell> sudo chkconfig --add prometheus
shell> sudo chkconfig --list | grep prometheus
shell> sudo service prometheus start
shell> sudo service prometheus status
```

- systemd-based procedure

  Create the `prometheus.service` boot script for systemd:

```
shell> sudo vi /etc/systemd/system/prometheus.service

[Unit]
Description=Prometheus
After=network.target

[Service]
User=prometheus
Group=prometheus
Type=simple
ExecStart=/usr/local/bin/prometheus \
  --config.file=/usr/local/prometheus/prometheus.yml \
  --storage.tsdb.path=/home/prometheus/data \
  --web.enable-admin-api

[Install]
WantedBy=multi-user.target
```

  Use systemctl to reload the boot config, enable the `prometheus` service to start at boot time, and then start Prometheus:

```
shell> sudo systemctl daemon-reload
shell> sudo systemctl enable prometheus
shell> sudo systemctl start prometheus
```

15.1.4. Example Prometheus Test Procedure

Once the Prometheus server has been started, you may test that it is running via browser URL `http://{yourServer}:9090/graph`

Prometheus may now be enabled in the Tungsten Dashboard one of two ways, either via the browser Dashboard settings panel or manually by editing the `config.json` file in the Dashboard WEBROOT directory. Add the configuration option `"enablePrometheus":1` and refresh the Dashboard page in the browser to see the additional button in the top navigation bar.

For more information about next steps with Prometheus, please visit the Prometheus website at https://prometheus.io/docs/introduction/first_steps/

15.2. Monitoring Tungsten Clusters Using Grafana

The below example procedure is designed to help you get Grafana installed and working with the goal of monitoring Tungsten Clusters through the Dashboard.

This section of the documentation is a summary guide for how to install an external software product, Grafana. The usual caveats apply, and as always, your mileage may vary.

For more information about getting started with Grafana, please visit the Grafana website at https://grafana.com/docs/grafana/latest/guides/getting_started/

15.2.1. Example Grafana Installation Procedure

This procedure example uses the YUM-based method. For other ways to install Grafana, please visit the Grafana install page at https://grafana.com/docs/grafana/latest/installation/

First, create the YUM repository configuration file for Grafana:

```
shell> sudo vi /etc/yum.repos.d/grafana.repo

[grafana]
name=grafana
baseurl=https://packages.grafana.com/oss/rpm
repo_gpgcheck=1
enabled=1
gpgcheck=1
gpgkey=https://packages.grafana.com/gpg.key
sslverify=1
sslcacert=/etc/pki/tls/certs/ca-bundle.crt
```

Install Grafana using yum:

```
shell> sudo yum install grafana
```

15.2.2. Example Grafana Configuration Procedure

REQUIRED STEP - Configure Embedding

In order to use the Grafana integration with the Tungsten Dashboard, one line needs to be added to the `[security]` stanza in the Grafana configuration file (normally `/etc/grafana/grafana.ini`). This setting is usually commented out and set to false, so just add a new line under the commented one:

```
shell> sudo vi /etc/grafana/grafana.ini

...
[security]
;allow_embedding = false
allow_embedding = true
...
```

OPTIONAL STEP - Configure Anonymous Auth

The embedded Grafana panel will require a login and password. To disable this requirement, and allow the panels to be shown without authentication via limited read-only access, add two lines under the `[auth.anonymous]` stanza in the Grafana configuration file (normally `/etc/grafana/grafana.ini`). These settings are usually commented out, so just add the new lines under the commented ones:

```
shell> sudo vi /etc/grafana/grafana.ini

...
[auth.anonymous]
;enabled = false
enabled = true

;org_name = Main Org.
org_name = {Your Exact Organization Name In Grafana}
...
```

OPTIONAL STEP - Configure HTTPS

The embedded Grafana panel will be called on whatever transport is used to load the main Tungsten Dashboard page, i.e. http or https. To configure Grafana to use https, add four lines to the `[server]` stanza in the Grafana configuration file (normally `/etc/grafana/grafana.ini`). These settings are usually commented out, so just add the new lines under the commented ones:

```
shell> sudo vi /etc/grafana/grafana.ini

...
[server]
;protocol = http
protocol = https

;domain = localhost
domain = dashboard.yourdomain.com

;cert_file =
cert_file = /etc/letsencrypt/archive/dashboard.yourdomain.com/fullchain1.pem

;cert_key =
cert_key = /etc/letsencrypt/archive/dashboard.yourdomain.com/privkey1.pem
...
```

Important

It is critical that the `domain =` value be the same FQDN as the one the Tungsten Dashboard web service answers to, and that the same keys are in use.

Our example shows Let's Encrypt certificates that are shared with the Dashboard web server instance. For this to work, the permissions of the key files must allow for Grafana to access them. Velow is an example of how you could allow access to the needed certificate files:

```
shell> sudo chgrp grafana /etc/letsencrypt/archive/dashboard.yourdomain.com/privkey1.pem
shell> sudo chmod g+r /etc/letsencrypt/archive/dashboard.yourdomain.com/privkey1.pem
```

Please remember to restart Grafana when the certificates expire and get renewed!

Below is a sample Let's Encrypt command to get a cert for dashboard.yourdomain.com, assuming that there is real DNS for that domain, that it resolves for the world, and that web server is reachable by the world:

```
shell> certbot certonly \
--webroot \
--renew-by-default \
--agree-tos \
-v \
--debug \
--email you@yourdomain.com \
-w /volumes/data/tungsten/html \
-d dashboard.yourdomain.com \
--dry-run
```

Please remember to remove the `--dry-run` argument at the end and re-run to get the real certs!

All examples provided for convenience. As always, YMMV, and supporting Grafana and/or certificates is outside Continuent's scope.

15.2.3. Example Grafana Boot Configuration Procedure

The YUM-based install automatically creates the `grafana` user, along with the `systemd` and `init.d` boot scripts. This means you do not have to create the boot scripts by hand!

- init.d-based procedure

  Enable the `grafana-server` service to start at boot time via chkconfig, and then start it using service:

  ```
  shell> sudo chkconfig --add grafana-server
  shell> sudo chkconfig --list | grep grafana-server
  shell> sudo service grafana-server start
  shell> sudo service grafana-server status
  ```

- systemd-based procedure

  Use systemctl to reload the boot config, enable the `grafana-server` service to start at boot time, and then start Grafana:

  ```
  shell> sudo systemctl daemon-reload
  shell> sudo systemctl enable grafana-server
  shell> sudo systemctl start grafana-server
  ```

15.2.4. Example Grafana Test Procedure

Once the Grafana server has been started, you may test that it is running via browser URL `http://{yourServer}:3000`

Login as user `admin` with a password of `admin`, and please change the admin password when prompted to do so.

Grafana may now be added to the Dashboard via the `config.json` file in the Dashboard WEBROOT directory. Add the configuration option `"enableGrafana":1` and refresh the Dashboard page in the browser to see the additional button in the top navigation bar.

For more information about next steps with Grafana, please visit the Grafana website at https://grafana.com/docs/grafana/latest/guides/getting_started/

15.2.5. Example Grafana Setup and Usage

Once logged into the Grafana server as admin, you may configure a data source and import the dashboards.

- Create a data source using Prometheus

  Click the Configuration cog on the left nav bar, then click "Add data source", Choose prometheus, then add 'http://localhost:9090' to the HTTP URL field, then click Save & Test at the bottom. This should create a new data source named 'Prometheus for use a few steps below.

- Optional Step: Import the Included Prometheus and Grafana Dashboards

  If you want to use the built-in metrics for Prometheus and/or Grafana, import the included dashboards as desired.

  Click on the Dashboards Tab in the center window to the right of the Settings Tab, then click on the blue Import button for each of Prometheus Stats, Prometheus 2.0 Stats and Grafana metrics.

- Import the Continuent Tungsten Dashboard

  Hover over the Dashboards icon in the left nav bar, then select Manage from the sub-menu. Click the Import link to the right of the green New Dashboard button.

  In the Grafana.com Dashboard field, enter `12760` for the Continuent Tungsten dashboard, then click Load.

  Select the Prometheus data source, then click the green Import button.

  If you have Prometheus setup correctly and running, you should see results instantly.

  Save this Dashboard by clicking the 3.5 inch floppy icon in the upper-right corner, then click the green Save button.

  Click the star in the upper-right corner to make this dashboard a favorite. This makes finding the dashboard MUCH easier.

- Import the Node Exporter Full Dashboard

  Hover over the Dashboards icon in the left nav bar, then select Manage from the sub-menu. Click the Import link to the right of the green New Dashboard button.

  In the Grafana.com Dashboard field, enter `1860` for the Node Exporter Full dashboard, then click Load.

  Select the Prometheus data source, then click the green Import button.

  If you have Prometheus setup correctly and running, you should see results instantly.

  Save this Dashboard by clicking the 3.5 inch floppy icon in the upper-right corner, then click the green Save button.

  Click the star in the upper-right corner to make this dashboard a favorite. This makes finding the dashboard MUCH easier.

- Import the Percona MySQL Dashboard

  Hover over the Dashboards icon in the left nav bar, then select Manage from the sub-menu. Click the Import link to the right of the green New Dashboard button.

  In the Grafana.com Dashboard field, enter `7362` for the Percona MySQL dashboard, then click Load.

  Select the Prometheus data source, then click the green Import button.

  If you have Prometheus setup correctly and running, you should see results instantly.

  Save this Dashboard by clicking the 3.5 inch floppy icon in the upper-right corner, then click the green Save button.

  Click the star in the upper-right corner to make this dashboard a favorite. This makes finding the dashboard MUCH easier.

For more information about next steps with Grafana, please visit the Grafana website at https://grafana.com/docs/grafana/latest/guides/getting_started/

Appendix A. Dashboard Frequently Asked Questions (FAQ)

The following details information should be considered when using the Tungsten Dashboard:

- A DS state of `ONLINE` when the node role is Witness means that the manager is online only. An Active Witness node will never be a live DataSource because it has no database and no replicator.
- Passive Witness nodes will NOT appear because they have no running Manager/API.
- The Tab Menu Badges for Policy Not Auto and Not Ready auto-refresh via AJAX every 30 seconds independently of the main Auto-refresh Setting.
- The Show All Details button is useful when used with the native browser search.
- All operations will attempt to obtain a lock automatically.
- An auto-lock request will fail if the resource is already locked.
- Composite and Cluster Status may be one of: Ready, Warning or Error.
- For a Composite to be other than Ready, a Member cluster must be `OFFLINE` or `FAILED` from the Composite view. A single failed node will NOT change the Composite Status.
- There is no impact on the Manager API if security is enabled via `--disable-security-controls=false`.
- The Manager API calls are not encrypted with SSL by default.
- Filtering is only available with more than one cluster.
- Filtering is case-insensitive with automatic wildcards on both ends.

Appendix B. Release Notes

B.1. Tungsten Dashboard 1.0.15 GA (14 February 2024)

Version End of Life. 14 February 2025

Tungsten Dashboard provides a web-based UI for monitoring and managing Tungsten Clustering deployments.

Tungsten Dashboard v1.0.15 is a feature and bugfix release.

- New Feature - Dashboard now checks to see if the pipelinesource on a replica is not pointing to the current primary and highlights the pipelineSource with bold red if mismatched.
- New Feature - now able to change the Auto-Configure host and port via Settings
- Improvement - Dashboard now has Connector polling disabled by default to reduce manager load. Enable via the Settings panel.

  Technical Details - automatically adds the `?includeRouters=false` flag to the end of calls to `/api/v2/manager/cluster/status` and `/api/v2/manager/status/service/{$service}`
- Improved Docker support.
- Improved support for Distributed Data Groups (DDG), including an added setting for enableDDGNodeColors.
- Improvement - updated various wording to differentiate between the Auto-Define and Auto-Confgigure features. updates all copyrights to 2024
- Improvement - Updates all copyrights to 2024.
- BugFix - Primary definition corrected to exclude shunned or failed masters.
- BugFix - Settings panel help broken when fancy tooltips were not enabled.
- BugFix - Auto-Define now gets the proper security defaults when none are specified in the form.

B.2. Tungsten Dashboard 1.0.14 GA (11 April 2023)

Version End of Life. 11 April 2024

Tungsten Dashboard provides a web-based UI for monitoring and managing Tungsten Clustering deployments.

Tungsten Dashboard v1.0.14 mainly provides support for jQuery 3.6.4 along with a new "Client Request Tracking" feature.

- Upgraded to jQuery 3.6.4
- Added new feature "Client Request Tracking" to log one line per client request call.
- Improved formatting for the diagnostic upload result message.

B.3. Tungsten Dashboard 1.0.13 GA (31 January 2023)

Version End of Life. 31 January 2024

Tungsten Dashboard provides a web-based UI for monitoring and managing Tungsten Clustering deployments.

Tungsten Dashboard v1.0.13 is a release for a single bugfix.

- Fixed an auto-configure regression for composite clusters.

B.4. Tungsten Dashboard 1.0.12 GA (14 December 2022)

Version End of Life. 14 December 2023

Tungsten Dashboard provides a web-based UI for monitoring and managing Tungsten Clustering deployments.

Tungsten Dashboard v1.0.12 mainly provides support for the new datasource drain feature in Tungsten Clustering v7.0.2, along with a few improvements and bugfixes.

- The new datasource drain feature in Tungsten Clustering v7.0.2 is now supported in the Tungsten Dashboard node menus

  There is a new,related setting called `drainTimeout` which controls the length of time to wait before closing the connection to a database node from a Connector

- Changed managerPort setting default value to 8201 from 8091
- Updated the auto-refresh browser-specific timestamp display to fit better on the navbar
- Corrected a page reload issue

B.5. Tungsten Dashboard 1.0.11 GA (8 November 2022)

Version End of Life. 8 November 2023

Tungsten Dashboard provides a web-based UI for monitoring and managing Tungsten Clustering deployments.

Tungsten Dashboard v1.0.11 provides a number of new features, improvements and bugfixes.

Cluster Configuration File Changes

- The `config.php` file has been updated to include a `config.json` file in the same subdirectory. If the `config.json` exists, the contents will be used in place of any json configuration defined manually in the `config.php` file.
- The new best practice is to use only `config.json`, not `config.php` any longer.
- Since there is a new `config.php` file with this version, a new migration tool called `upgrade_config.php` has been included to easily install the new `config.php` file while still maintaining all of your existing settings.

  The `upgrade_config.php` command will create a new file called `config.json` containing all of the settings that used to be inside the `config.php` file.

  > **Important**
  >
  > The `upgrade_config.php` tool should be NOT be RUN if the `config.json` file already exists!
  >
  > The `upgrade_config.php` tool should be RUN ONLY ONCE!

```
COMPLETE ALL THE STANDARD UPGRADE STEPS FIRST!

shell> cd {DASHBOARD_WEBROOT}/
For example:
shell> cd /volumes/data/www/tungsten/html/

shell> ./upgrade_config.php `pwd`
```

> **Important**
>
> The `upgrade_config.php` tool should be NOT be RUN if the `config.json` file already exists!
>
> The `upgrade_config.php` tool should be RUN ONLY ONCE!

New! Cluster Tagging and Filtering Feature

- You are now able to specify one or more tags per cluster in all cluster definition forms.
- You are now able to filter the cluster display by a tag from the top search bar.

Support for both v6/APIv1 and v7/APIv2 clusters at the same time!

Support for per-cluster API User and Password!

- You are now able to specify the API settings on a per-cluster basis (apiVersion, apiAuth, apiUser, apiPassword and apiSSL), allowing for mixed APIv1 and APIv2 clusters in the same Dashboard session, and clusters with different admin user/password pairs.
- In previous versions of the Tungsten Dashboard, the API Version, API Authentication/User/Password and API Encryption settings were global only. This meant that a version 6 cluster running APIv1 and a version 7 cluster running APIv2 would not work at the same time in the same Dashboard sessioa, nor would clusters with different API user/password pairs.
- The new configuration fields have been added to the display, add and edit cluster definition forms in all areas of the dashboard.
- The API settings for each cluster are now displayed upon hover over the cluster service name. Each option will also show if the value is derived from the global default, or from a cluster-specific setting.

- Expert mode now allows one-click toggling of global API authentication and SSL Encryption options in addition to the existing global API version toggle.

New Display Options for the REACT Frontend GUI

- Two new settings have been added to the Tungsten Dashboard to support an improved GUI experience.
- Both of the new settings are available via the GUI (config.php options in parens):
  - Display Style (displayStyle)

    Specify which horizontal display method to use in the new REACT frontend (Fill Space or Compressed)
  - Home Page (homepage)

    Specify which page to display first in the new REACT frontend (Cluster, Dashboard or Metrics)

New Cluster Rename Behavior

- When changing a composite cluster parent name, the children are now updated with the new parent name. Previously, this was a manual operation.
- when changing a composite cluster child name, the parent is now updated with the new child name. Previously, this was a manual operation.

New Tab Bar Behaviors for Improved Performance

- A new setting (enableTabs) has been added to the Tungsten Dashboard to control the Tab bar behavior.
- While getting the information to populate the Tab bar is a lightweight AJAX call, the actual data gathering in the backend is quite heavy, and requires an API call to every cluster. This backend processing can cause slowdowns in response time for the Dashboard and is why the new default is to have "dumb" Tabs and better performance.
- Now, by default, the Tab bar will operate in "dumb" mode, and will NOT display the quantity of nodes that are not in `AUTOMATIC` mode, nor the quantity of nodes that are in the not ready state. Also, the Tab refresh button will not appear either.
- Enabling Tabs in the Settings Panel restores the original behavior of displaying the counts and the refresh button.

Additional New Features of Note

- The Tungsten Dashboard now fully supports CAA clusters in v6!
- Added the ability to close windows using the Escape key, enabled by default. Disable in the Settings panel.
- Added a Copy To Clipboard button to all cluster configuration display windows.
- Added a refresh button to the missing cluster display box for convenience - no longer need to refresh the entire page
- In expert mode, there is a new heartbeat trigger button per cluster in old frontend
- Cleaned up error_log calls to make the log file as quiet as possible, and added new setting (enableVerbose) and the aassociated verbose_log() function.
- When RBAC is enabled, a logout link is now visible at the bottom center of the footer. Also improved footer messaging when RBAC is disabled and/or basic auth is disabled.
- For both auto and manual cluster definition, if you create a Dashboard Service ID (service name) with hyphens, they will be converted to underscores upon save due to the way Javascript handles id's with hyphens.
- When useHAProxy is enabled, the Manager Port will now be set to 8201 instead of 8091 to avoid port conflicts when installed directly on a cluster node.
- The browser-specific date and time display for auto-refresh has been shortened to better fit on the navbar.

B.6. Tungsten Dashboard 1.0.10 GA (7 March 2022)

Version End of Life. 6 March 2023

Tungsten Dashboard provides a web-based UI for monitoring and managing Tungsten Clustering deployments.

Tungsten Dashboard v1.0.10 provides a number of new features, improvements and bugfixes.

Prometheus and Grafana Integration

- Two new settings have been added to the Tungsten Dashboard to support Prometheus and Grafana integration.
- Both of the new settings are available via the GUI (config.php options in parens):
  - Enable Prometheus Integration (enablePrometheus)

    Enable integration with Prometheus to display data in a new window.
  - Enable Grafana Integration (enableGrafana)

    Enable integration with Grafana to display graphs inside the Dashboard.

Configurable CURL Timeouts

- Two new settings have been added to the Tungsten Dashboard to help compensate for slow environmemnts where API calls may take longer to complete.
- Both of the new settings are available via the GUI (config.php options in parens):
  - CURL GET Timeout (curlTimeoutGET)

    The timeout used when curl connects to the Manager for a GET-specific API call, in seconds.
  - CURL POST Timeout (curlTimeoutPOST)

    The timeout used when curl connects to the Manager for a POST-specific API call, in seconds.

New Audit Trail Feature

- The ability to track all write API calls made has been added.
- There will be one audit file per day created in the auditDir (default: {WEBROOT}/audit.d).
- One line per write containing: timestamp, ipaddr, user, role and msg/action
- The new setting is partially available via the GUI (config.php options in parens):
  - Enable Audit Trail (enableAudit)

    Capture all POST API calls to a file in the {WEBROOT}/audit.d subdir like audit-{YYMMDD}.log
  - Audit subdirectory name (auditDir) (NOT available via GUI)

    The directory used to store the audit files (default: {WEBROOT}/audit.d)

New Notes-Per-Node Feature

- You can now store text on a per-node basis.
- There will be one note file per node created in the notesDir (default: {WEBROOT}/notes.d).
- The new setting is partially available via the GUI (config.php options in parens):
  - Enable Per-Node Notes (enableNotes)

    Turn on the notes per node feature to capture text to a file in the {WEBROOT}/notes.d subdir, named like {SERVICE}-{FQDN}.txt
  - Note Icon (noteGlyphicon) (NOT available via GUI)

    Use noteGlyphicon to specify the note-per-node Glyphicon (default: comment).
  - Notes subdirectory name (notesDir) (NOT available via GUI)

    The directory used to store the note files (default: {WEBROOT}/notes.d)

New API URL Display Feature

- You can now enable the display of the back-end API call URLs for transparency and learning.
- The new setting is available via the GUI (config.php options in parens):
  - Enable API URL Display (enableURLDisplay)

    Display the API call URL for each command run.

New Flag-On-Lag Feature

- You can now automatically highlight node rows where the Replicator is lagging by a specified number of seconds.
- The new setting is available via the GUI (config.php options in parens):
  - Flag-On-Lag Delay (flagOnLagDelay) (in seconds)

    Set this option to a non-zero value to enable node row highlighting when the Replica is more than the specified number of seconds behind the Primary.
  - Flag-On-Lag Color (flagOnLagColor)

    The background color to use when marking a node row as too far behind. One of Info (blue), Warning (yellow) or Danger (red).

B.7. Tungsten Dashboard 1.0.9 GA (12 August 2020)

Version End of Life. 11 August 2021

Tungsten Dashboard provides a web-based UI for monitoring and managing Tungsten Clustering deployments.

Tungsten Dashboard v1.0.9 provides a number of new features, improvements and bugfixes.

Dashboard Configuration

- Now able to configure Dashboard settings via the browser

  You can disable the editing of settings in the browser by changing the value of `disableSettingsEdit` to `1` in the `config.php` file, in the `"settings": { }` stanza:

  ```
  "disableSettingsEdit": 1
  ```
- All settings configured via the browser page are stored in the `{webroot}/settings.d/` directory as individual JSON text files named for the setting. Please ensure it exists and is writable by the web server user.
- You may edit or delete any of the files in the `{webroot}/settings.d/` directory. The setting will revert to the default if deleted. you may also choose to configure settings in this way as opposed to using the config.php file. Your choice.
- Refactored all options and created centralized defaults

Software Update

- Now able to self-update the Dashboard software via the browser

  There are four related settings, `enableUpdates`, `tmpDir`, `downloadAccessKey` and `downloadSecretKey`.

  All four must be located in the `config.php` file, in the `"settings": { }` stanza. They are not accessible from the browser settings page.

  You can disable the Dashboard self-update feature by changing the value of `enableUpdates` to `0` in `config.php` (default: 1):

  ```
  "enableUpdates": 1
  ```

  The `tmpDir` value is used to determine where downloaded software packages are saved to:

  ```
  "tmpDir":"/tmp"
  ```

  The other two (`downloadAccessKey` and `downloadSecretKey`) need to be obtained from Continuent support and typicially ship with the Dashboard installation package.

Cluster Definitions

- Now able to manually create and save cluster definitions in the `conf.d` subdirectory. Originally, a cluster could only be defined in the `"clusters": { }` stanza.
- Now able to create and save cluster definitions to the `conf.d` subdirectory via a browser workflow
- Added Display, Edit and Remove Cluster Definition menu choices for each cluster
- Now able to automatically define cluster definitions in `conf.d` just by providing a hostname and port number in a browser workflow
- Now able to automatically define cluster definitions in `conf.d` at Dashboard startup

  There are three related settings, `enableAutoConfiguration`, `managerPort` and `useHAProxy`.

You can enable the Dashboard auto-configuration feature by changing the "Enable Auto-Configuration?" setting via the Dashboard settings page in the browser, or changing the value of `enableAutoConfiguration` to `1` in `config.php` (default: 0) or via the Dashboard settings page in the browser:

```
"enableAutoConfiguration": 1
```

The `managerPort` value is used to determine what port to communicate with the manager upon when performing auto-configuration and auto-define, as well as populating form fields in other places. Only change this if you have change the API listener port for the Manager as well.

```
"managerPort": 8090
```

The `useHAProxy` value is used to determine how to calculate ports when performing auto-configuration and auto-define.

Set the value to `1` to determine the manager port number automatically during various operations based on calulations using the base managerPort.

Set the value to `0` (default) to use the base managerPort with no attempt to auto-define the port.

You can enable the manager port auto-configuration feature by changing the "Using HA Proxy?" setting via the Dashboard settings page in the browser, or changing the value in the `config.php` file.

```
"useHAProxy": 1
```

UI/UX

- Role name cleaning (Master is now Primary, and Slave is now Replica for nodes; Master is now Active, and Slave is now Passive for clusters)
- Improve error handling for JSON responses to AJAX calls
- Bug fixes in service alias support
- Many footer improvements, including a link to check for an available Dashboard software update
- Stop providing tabInfo during intitial page load, instead do it as AJAX call after load to save initial page load time

Dashboard Diagnostics

- Now able to upload a Dashboard Diagnostic containing the JSON configuration to Continuent Support's protected AWS bucket. No other customer has access to this location, it is upload-only.

  There are three related settings, `customerName`, `uploadAccessKey` and `uploadSecretKey`.

  The `customerName` value is used to pre-populate the diagnostic upload form.

  ```
  "customerName":"your customer name here"
  ```

  The other two (`uploadAccessKey` and `uploadSecretKey`) need to be located in `config.php`

  ```
  "uploadAccessKey":"AKIAIWDZPQUE5YL4SBDQ", ]
  "uploadSecretKey":"FQ0iVkTtH9biIZT2+IpwXwhqXvVwqMUqsZ4++N4K" ]
  ```

Misc Admin

- New Expert mode disables both confirmation prompts when Deleting All Definitions

  The default is `0` (disabled). Set `enableExpertMode` to 1 (one) to enable.

  ```
  "enableExpertMode": 1
  ```

- Use the `enableDebug` setting to get additional logging information and use the debug software versions when checking for an available update.

  ```
  "enableDebug": 1
  ```

B.8. Tungsten Dashboard 1.0.8 GA (4 June 2020)

Version End of Life. 3 June 2021

Tungsten Dashboard provides a web-based UI for monitoring and managing Tungsten Clustering deployments.

Tungsten Dashboard v1.0.8 provides a number of new features, improvements and bugfixes.

- Added basic Role-Based Access Control (RBAC). There are two roles, Administrator with full access and Operator with Read-Only access. This feature requires Basic Auth to be properly configured on the Web server.

  When enabled, the user's current role will be displayed in the footer. Refresh the page to activate any changes to `config.php`.

  The default is `0` (disabled). Set `enableRBAC` to 1 (one) to enable.

  ```
  "enableRBAC":1
  ```

  Use the `administrators` setting to list the users with admin privs:

  ```
  "administrators": [ "adminUser1","adminUser2" ]
  ```

- Improved page load performance via caching of API calls. This is especially helpful with Composite clusters that have multiple sites over a wide area.
- Added the ability to modify the browser window title using the new configuration option `windowTitle`
- Added the ability to change the cluster service sort order from the alpha default to as-written configuration order using the new configuration option `sortByConfigOrderNotAlpha`
- Site favicons along with the navigation bar logo and colors have been updated to promote a cleaner look. Additional icon replacements and color tweaks have been made throughout the tool.
- Added hover-based tooltips for all fields and buttons where possible. Set `disableTooltips` to 1 to prevent the tooltips from appearing.
- Significantly improved the Connector popover formatting, sorting and operation.
- Message handling is improved so that multiple actions and responses are tracked and messaged properly.
- Added the ability to view the json configuration in the browser via a menu link.
- Added the ability to check for Dashboard software updates.
- Added the ability to check for Clustering software updates on a per-node basis.

Tungsten Dashboard is compatible with both the Tungsten Clustering 5.3.x series and 6.x series.

B.9. Tungsten Dashboard 1.0.7 GA (26 November 2019)

Version End of Life. 26 November 2020

Tungsten Dashboard provides a web-based UI for monitoring and managing Tungsten Clustering deployments.

Tungsten Dashboard v1.0.7 provides a number of new features, improvements and bugfixes.

- Added the feature to allow for cluster service name aliases. You may now add the sub-key `actualName` pointing to the "real" name of the service, and change the top-level cluster service name to some alias that you understand.

  Previously, it was impossible to configure two or more clusters with the same service name. This could be required if clusters were installed into different environments like production, staging or development. While the best practice is to name the cluster services to match the environment (i.e. east_prod and east_staging), in some situations this may not be possible.

- Added a new feature to automatically fade out messages after a delay. The default is `60` seconds. Set `msgFadeOutTimer` to 0 (zero) to disable or to a positive integer to specify the delay in seconds.

  ```
  "msgFadeOutTimer":60
  ```

- Improved the look & feel of the overall layout, including display widths, the location of the timestamp marker and spacing.
- Fixed a bug where the controls to open and close a cluster were STILL not working.
- Fixed a bug where the datasource status details hover was not displaying properly

Tungsten Dashboard is compatible with both the Tungsten Clustering 5.3.x series and 6.x series.

B.10. Tungsten Dashboard 1.0.6 GA (3 September 2019)

Version End of Life. 3 September 2020

Tungsten Dashboard provides a web-based UI for monitoring and managing Tungsten Clustering deployments.

Tungsten Dashboard v1.0.6 is a bugfix and minor feature release.

- Fixed a bug where the controls to open and close a cluster were not working.
- When Auto-refresh is turned on, any issuence of a command will stop the auto-refresh. Simply re-select your desired refresh rate to turn it back on.

Tungsten Dashboard is compatible with both the Tungsten Clustering 5.3.x series and 6.x series.

B.11. Tungsten Dashboard 1.0.5 GA (28 June 2019)

Version End of Life. 28 June 2020

Tungsten Dashboard provides a web-based UI for monitoring and managing Tungsten Clustering deployments.

Tungsten Dashboard v1.0.5 is a bugfix release.

- Fixed CMM cluster bug where clusters other than the first do not show subservices.
- Tweaked cell alignment

Tungsten Dashboard is compatible with both the Tungsten Clustering 5.3.x series and 6.x series.

B.12. Tungsten Dashboard 1.0.4 GA (11 April 2019)

Version End of Life. 11 April 2020

Tungsten Dashboard provides a web-based UI for monitoring and managing Tungsten Clustering deployments.

Tungsten Dashboard v1.0.4 is a bugfix release.

- Fixed cluster-level open/close regression.
- Tweaked error text and reduced noise in the logs.

Tungsten Dashboard is compatible with both the Tungsten Clustering 5.3.x series and 6.x series.

B.13. Tungsten Dashboard 1.0.3 GA (22 March 2019)

Version End of Life. 22 March 2020

Tungsten Dashboard provides a web-based UI for monitoring and managing Tungsten Clustering deployments.

Tungsten Dashboard v1.0.3 is a feature release for better global controls and customization.

The default for `navButtonFormat` is `icon` if not specified.

- Added modal "Stop Auto-Refresh" button which will turn off the Auto-refresh feature. This button is only visible if auto-refresh is enabled.
- Added ability to set global buttons to icon, text or some combination. Use the setting `navButtonFormat` and specify one or more of `icon` or `text` as a comma-separated string, no spaces. Order counts.

```
$jsonConfig = <<<EOJ
{
 "settings": {
 "navButtonFormat":"icon",
...
EOJ;
```

Currently there are four (4) possible entries:

```
"navButtonFormat":"icon",
 "navButtonFormat":"text",
 "navButtonFormat":"icon,text",
 "navButtonFormat":"text,icon",
```

Tungsten Dashboard is compatible with both the Tungsten Clustering 5.3.x series and 6.x series.

B.14. Tungsten Dashboard 1.0.2 GA (20 September 2018)

Version End of Life. 20 September 2019

Tungsten Dashboard provides a web-based UI for monitoring and managing Tungsten Clustering deployments.

Tungsten Dashboard v1.0.2 is a bug fix release for better API error handling.

- Refactored API calls for better error handling.
- Better error reporting on the front-end.

Tungsten Dashboard is compatible with both the Tungsten Clustering 5.3.x series and 6.x series.

B.15. Tungsten Dashboard 1.0.1 GA (17 September 2018)

Version End of Life. 17 September 2019

Tungsten Dashboard provides a web-based UI for monitoring and managing Tungsten Clustering deployments.

Tungsten Dashboard v1.0.1 is a bug fix release that also contains a few improvements.

- Support for Composite Active/Active topology offered in Continuent Clustering v6.x (requires Continuent Clustering version 6.0.3)
- Improvements to the menu system layout and clarity
- Composite-level cluster commands have been relocated to a new menu to the right of the State field
- Composite clusters now display the actual composite state instead of the Ready/Warning/Error status indicators, and status indicator lights have been moved to the left of the State label
- Improvements to the locking system:
  - Auto-Lock and Auto-Unlock are now both configurable via config.php
  - Auto-Lock and Auto-Unlock setting are now both visible at the bottom of the cluster-level locking menu
  - Auto-Lock may be configured to attempt a lock for all actions, heartbeats only, or not at all
  - Auto-Unlock may be configured to attempt an unlock for all actions, heartbeats only, or not at all
- Additional formatting tweaks, including the reduction in height of the rows

Tungsten Dashboard is compatible with both the Tungsten Clustering 5.3.x series and 6.x series.

B.16. Tungsten Dashboard 1.0.0 GA (10 May 2018)

Version End of Life. 10 May 2019

Tungsten Dashboard provides a web-based UI for monitoring and managing Tungsten Clustering deployments.

It supports the following features:

- Full monitoring information on the status and progress of replication and the status of the cluster
- Monitor multiple clusters through a single page
- Perform switches and failovers
- Shun hosts
- Recover failed hosts

Tungsten Dashboard is compatible with the Tungsten Clustering 5.3.x series.

Appendix C. Upgrade the Tungsten Dashboard

C.1. Manually Updating the Tungsten Dashboard Software

Manually Download and Upgrade the Tungsten Dashboard Software

> **Important**
>
> Please change the example values below to match your specific environment.

As user tungsten, download the software using the temporary URL provided by Continuent, or login to the web download portal to obtain the software [https://www.continuent.com/downloads/], then copy the updated application files to the web root directory, overwriting the existing ones:

```
shell> sudo su - tungsten

## Set the WEBROOT env var for convenience
shell> WEBROOT={DASHBOARD_WEB_ROOT_DIR_HERE}
For example:
shell> WEBROOT=/volumes/data/www/tungsten/html

## Make a backup of current Dashboard directory
shell> tar cvzf backup.tar.gz $WEBROOT

## Obtain the software package and cd to extracted dir
shell> wget -O tungsten-dashboard-1.0.11-1.tar.gz 'TEMP_URL_PROVIDED_BY_CONTINUENT'
shell> tar xvzf tungsten-dashboard-1.0.11-1.tar.gz
shell> cd tungsten-dashboard-1.0.11-1

## CHeck what would be updated with:
shell> rsync -acvn html/ $WEBROOT/

## Perform the actual upgrade with:
shell> rsync -acv html/ $WEBROOT/
```

> **Note**
>
> Your `config.php` will NOT be overwritten. The software package contains only `config.php.sample`, so there is no risk of affecting your settings during an upgrade.

Version 1.0.11. There is a new `config.php` file with this version, and along with it, a migration tool called `upgrade_config.php` to easily install the new `config.php` file, while still maintaining all of your existing settings. The `upgrade_config.php` feature was first introduced in Tungsten Dashboard version 1.0.11-1 This version now uses the `config.json` file for configuration.

> **Important**
>
> To use the `upgrade_config.php` commaand, please complete all the above upgrade steps first!

```
shell> cd $WEBROOT
For example:
shell> cd /volumes/data/www/tungsten/html/

shell> ./upgrade_config.php `pwd`
```

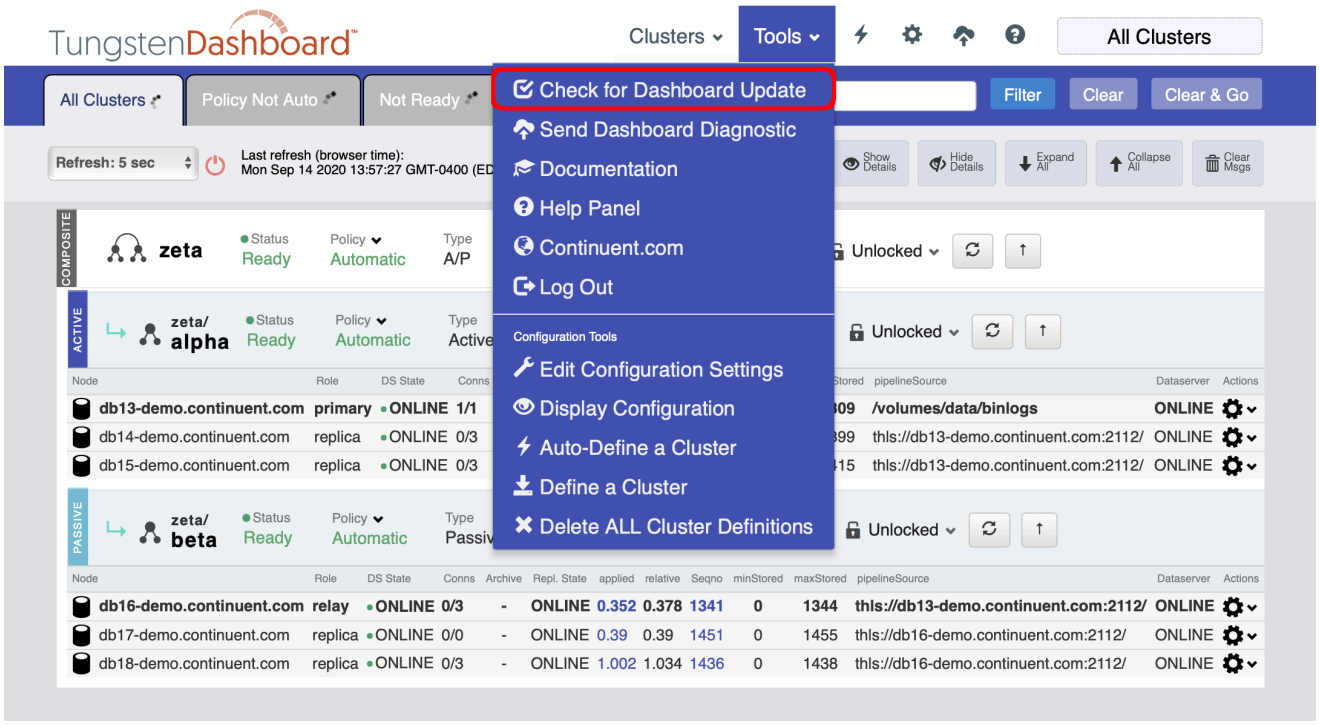
> **Important**
>
> The `upgrade_config.php` tool should be RUN ONLY ONCE!

C.2. Self-Updating the Tungsten Dashboard Software

Automatically Download and Upgrade the Tungsten Dashboard Software

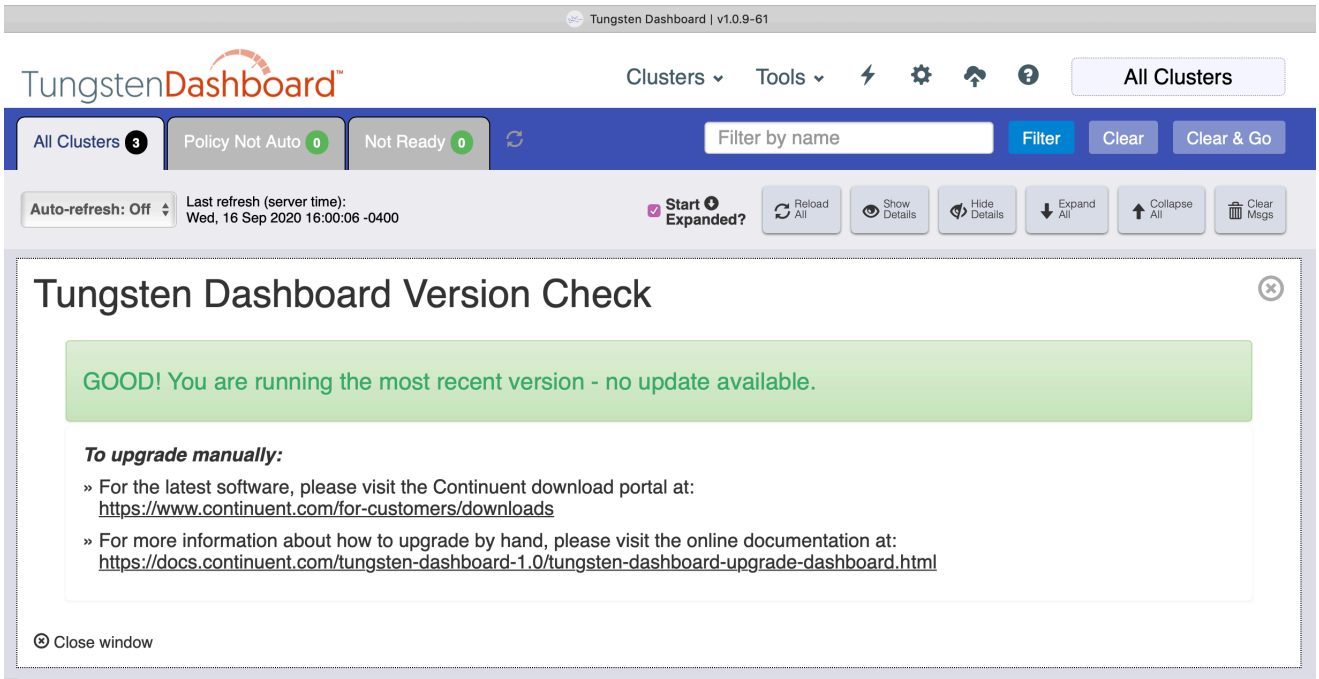
Version 1.0.9. This feature was first introduced in Tungsten Dashboard version 1.0.9-61

Figure C.1. Tungsten Dashboard Self-Update Menu Option



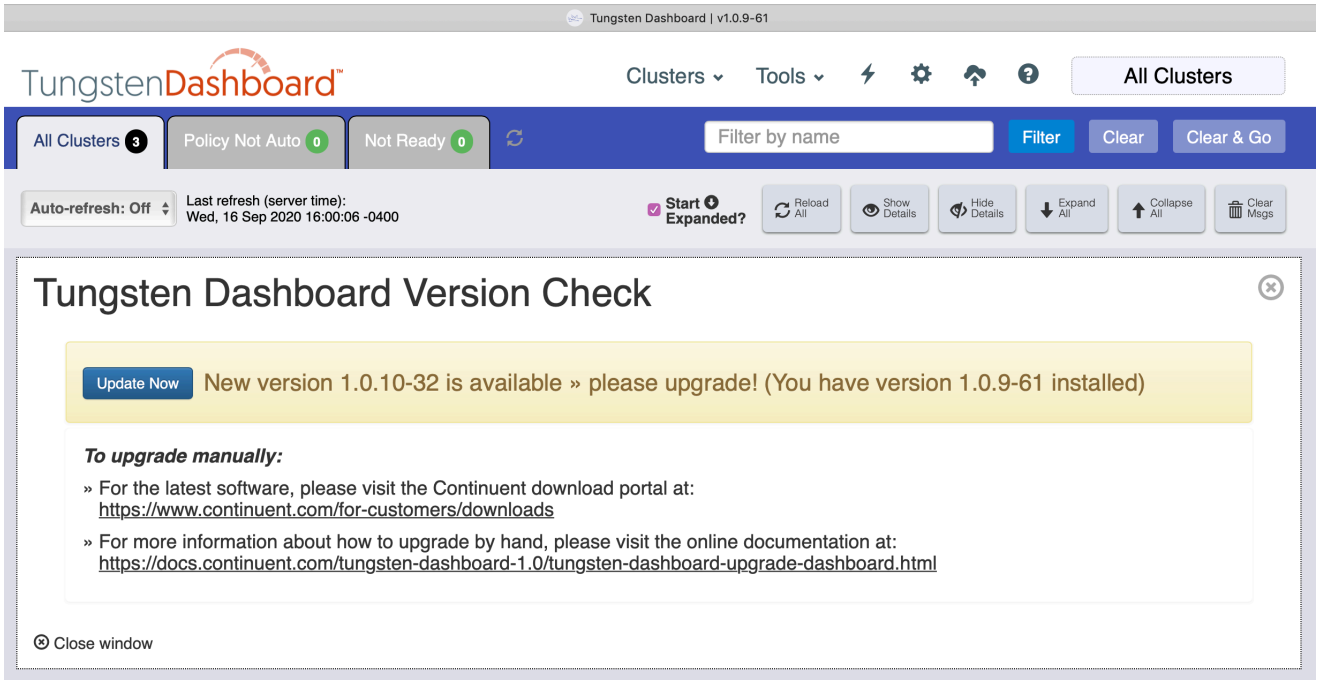
You may see a No Update Available message like this:

Figure C.2. Tungsten Dashboard No Update Available



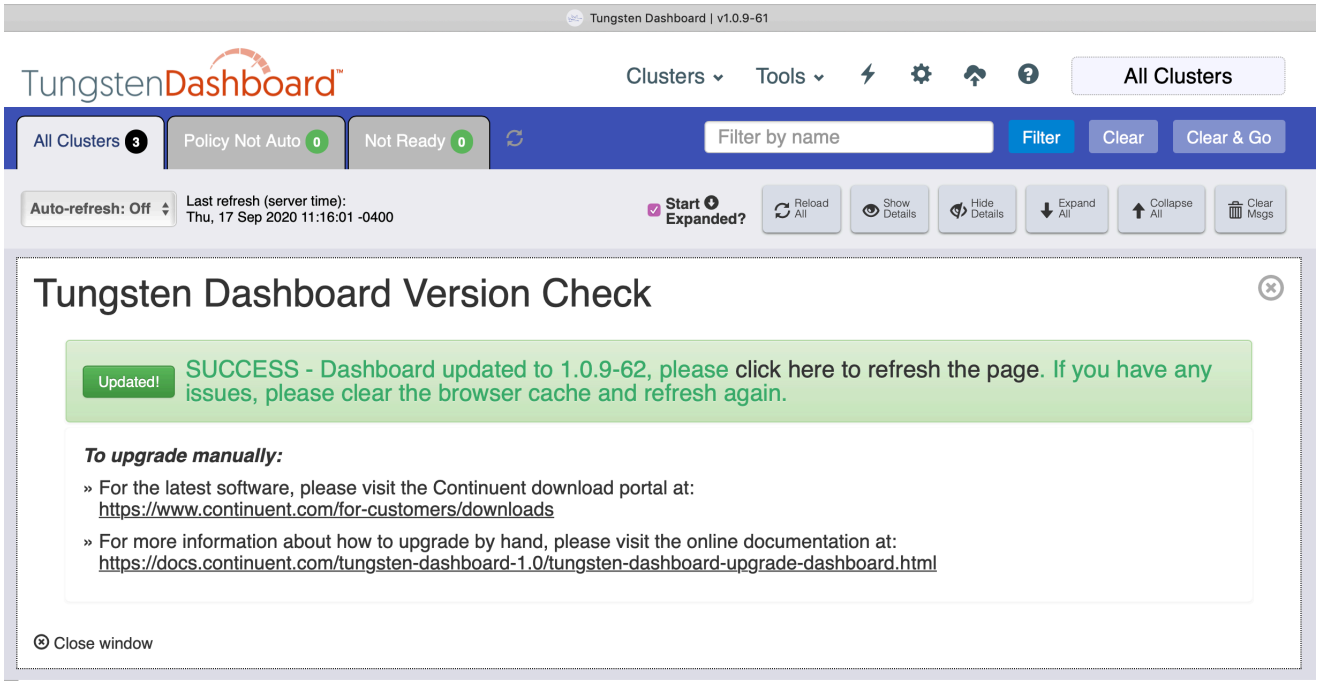
You may also see a New Version Available message like this:

Figure C.3. Tungsten Dashboard Self-Update Form



- Click the "Update Now" button to invoke the upgrade. There is no confirmation prompt, the upgrade begins immediately.
- When the upgrade is complete, simply refresh the page fully to get the new version.
- You may need to clear cache and refresh the page again to ensure the latest scripts and styles are loaded properly.

Figure C.4. Tungsten Dashboard Self-Update Success



Appendix D. UI Operational Scope Table

The following table describes the relationship between the UI elements on screen, their operation and scope, and the equivalent cctrl command that would be required to achieve the same operation.

UI Row Levels	Menu Label	Scope	cctrl Equivalent
Cluster	Composite Recover	Composite	use {composite_service}; datasource {composite_member} recover
Cluster	Composite Welcome	Composite	use {composite_service}; datasource {composite_member} welcome
Cluster	Composite Online	Composite	use {composite_service}; datasource {composite_member} online
Cluster	Composite Offline	Composite	use {composite_service}; datasource {composite_member} offline
Cluster	Composite Shun	Composite	use {composite_service}; datasource {composite_member} shun
Cluster	Composite Promote	Composite	use {composite_service}; switch to {composite_member}
Cluster	Composite Fail	Composite	use {composite_service}; datasource {composite_member} fail
Cluster	Heartbeat	Cluster	use {cluster_service}; cluster heartbeat
Cluster	Recover	Cluster	use {cluster_service}; recover
Cluster	Failover	Cluster	use {cluster_service}; failover
Cluster	Switch	Cluster	use {cluster_service}; switch
Composite	Heartbeat	Composite	use {composite_service}; cluster heartbeat
Composite	Recover	Composite	use {composite_service}; recover
Composite	Failover	Composite	use {composite_service}; failover
Composite	Switch	Composite	use {composite_service}; switch
Composite,Cluster	Set Policy to *AUTOMATIC*	Composite,Cluster	use {selected_service}; set policy automatic
Composite,Cluster	Set Policy to *MAINTENANCE*	Composite,Cluster	use {selected_service}; set policy maintenance
Node	Online	Node/Manager	use {cluster_service}; datasource {cluster_node} online
Node	Offline	Node/Manager	use {cluster_service}; datasource {cluster_node} offline
Node	Welcome	Node/Manager	use {cluster_service}; datasource {cluster_node} welcome
Node	Shun	Node/Manager	use {cluster_service}; datasource {cluster_node} shun
Node	Recover	Node/Manager	use {cluster_service}; datasource {cluster_node} recover
Node	Enable Archive	Node/Manager	use {cluster_service}; datasource {cluster_node} set archive
Node	Disable Archive	Node/Manager	use {cluster_service}; datasource {cluster_node} clear archive
Node	Backup	Node/Manager	use {cluster_service}; datasource {cluster_node} backup
Node	Promote	Node/Manager	use {cluster_service}; switch to {cluster_node}
Node	Fail	Node/Manager	use {cluster_service}; datasource {cluster_node} fail
Node	Restore	Node/Manager	use {cluster_service}; datasource {cluster_node} restore
Node	Enable Standby	Node/Manager	use {cluster_service}; datasource {cluster_node} standby
Node	Disable Standby	Node/Manager	use {cluster_service}; datasource {cluster_node} clear standby

UI Row Levels	Menu Label	Scope	cctrl Equivalent
Node	Online	Node/Replicator	use {cluster_service}; replicator {cluster_node} online
Node	Offline	Node/Replicator	use {cluster_service}; replicator {cluster_node} offline

Appendix E. Included External Packages In Use

Continuent Tungsten Dashboard includes the following software in the distribution package:

- bootstrap-3.3.7
- httpful-0.2.20
- jquery-1.12.4
- jsuri-1.3.1