冗長化構成 Gfarm 監視機能

導入・設定マニュアル

第 4.3 版

作成日：2021 年 10 月 4 日

変更履歴

版数	日付	変更内容	作成者
1.0	2012/03/16	初版。	SRA
1.0.1	2012/10/10	以下のファイル名の誤記を修正した。 ・userparameter_gfarm_redundant.conf ・register.php	SRA
1.0.2	2012/10/12	・ 「ファイル一覧」の表を、Zabbix 公式サイト提供分と gfarm_zabbix パッケージ提供分で分割した。 ・ 「zabbix_gfarm2.zip」という表記を、「gfarm_zabbix パッケージ」に改めた。 ・ 「Gfarm_zabbix 監視項目一覧.xls」というファイル名の誤記を修正した。 ・ 細かな誤植を修正した。	SRA
1.0.3	2012/11/21	・gfarm_zabbix 用インストールスクリプトを用意したので、本書もそのスクリプトを用いた手順を記述した。	SRA
	2012/12/06	・2012/11/2 の変更で、zabbix_server の起動に関する記述を削除してしまったため、直した。	SRA
1.1	2013/03/22	・CentOS 6 に対応した。 ・章立てを変更した。	SRA
	2014/05/23	・マクロ MONITOR_GFMD_DIRECTORY, MONITOR_GFSD_DIRECTORY についての記述を追加した。	SRA
2.0	2014/08/27	・gfarm_zabbix 2.0 用に改訂した。	SRA
2.1	2014/11/29	・gfarm_zabbix 2.1 用に改訂した。 ・CentOS 7 向けの記述（systemd 対応、firewalld 対応、MariaDB 採用）を追加した。	SRA
	2015/03/26	・ホストマクロ名｛$GFSD_HOSTNAMES｝を誤って｛$GFSD_HOST_NAMES｝と表記していたので修正した。 ・フェイルオーバスクリプトの仕組みについて、記載した。 ・ファイアウォールの設定を記した。	SRA
2.2	2015/07/09	・gfarm_zabbix 2.2 用に改訂した。	SRA

		・フェイルオーバスクリプトの挙動を詳述した。	
3.0	2015/09/30	・gfarm_zabbix 3.0 用に改訂した。	SRA
3.0.1	2015/12/22	・gfarm_conf.inc のパス名を訂正した。 ・gfuser コマンドの実行権限に関する記述を改訂した。 ・gfkey コマンドのオプションを改訂し、有効期限に関する記述を追加した。 ・zabbix_agentd.conf に ServerActive 設定に関する記述が欠けていたので追加した。 ・表 5-3 ホスト設定の項目「名前」は、Zabbix 2.0 以降では「ホスト名」となることを記載した。 ・表 5-3 ホスト設定の項目「IP アドレス」の記述を訂正した。	SRA
3.0.2	2016/03/16	「マクロ」欄の設定に関する注意を追加した。	SRA
3.1	2016/04/13	Web UI パスワード変更に関する注意を追加した。 iiabbix_get コマンド実行時に用いられるソース IP アドレスに関する注意を追加した。 「テンプレートのインポート」ボタンは、Zabbix 2.0 以降「インポート」となることを記載した。 表 5-3 ホスト名設定に、Zabbix 2.0 以降に存在する「表示名」欄を追加した。 Gfarm_gfmd_failover.conf の綴り誤りを訂正した。 Gfarm_gfmd_failover.pl の出力の解説を加えた。 Gfarm_zabbix 3.1 用に下記の改訂を行った。 ・Template_Gfarm_linux の位置づけを、これまでの Template OS Linux との排他利用から、共用に変更した。 ・これまで Template_Gfarm_linux に含まれていた監視項目のうち、Template OS Linux と排他利用になるものは Template_Gfarm_linux_alt に移した。 ・Template OS Linux と排他利用となる Template_Gfarm_gfmd_linux および Template_Gfarm_gfsd_linux を、Template_Gfarm_gfmd_linux_alt および	SRA

		Template_Gfarm_gfsd_linux へ改名した。	
4.0	2016/12/27	・仮に 3.1 としていたバージョン番号を 4.0 に変更した。 ・フェイルオーバ実行機能の設定に、「Zabbix エージェントの追加設定」の節を追加し、「Web インターフェース上での設定」の節に項目記述を追加した。 ・その他の注意点に、「メール通知設定」の節を追加した。	SRA
4.1	2017/03/15	・Gfarm_zabbix-4.0.1 用に改訂した。 ・install.conf に ZABBIX_PREFIX 設定を追加した。	SRA
4.1.1	2017/04/01	・Gfarm_zabbix-4.1.0 用に改訂した。 ・install.conf に GFMD_CONFIG_PREFIX 設定および POSTGRES_USER 設定を追加した。 ・sudoers に postgres ユーザーへの sudo 設定を追加した。	SRA
4.1.2	2018/11/02	・2.2 節「Gfarm 構成」において、サーバ監視には代表クライアント監視設定が必要であることを明記した。 ・表 5-4 および表 5-5「リンクするテンプレート」を、共通部分と選択分の 2 つの表として表記するのではなく、選択肢毎の 2 つの表として表記するよう変更した。 ・表 5-6 ホストマクロ設定において、MULTIPLE_EVENTS_TIMEOUT マクロおよび NODATA_TIMEOUT マクロの記述を追加した。また 3.1 版の Template_*_linux_alt テンプレートへの分割を反映していなかった箇所を修正した。	SRA
4.1.3	2019/09/09	・Gfarm_zabbix-4.1.0 用に failover_type として availability を設定した場合の記述を追加した。 ・{$TIME_DIFF_THRESHOLD}を 5 分から 30 秒に変更した。	SRA
4.2	2020/09/15	・Gfarm_zabbix-4.2.0 用に改訂した。 ・PostgreSQL の PID ファイルを指定する intall.conf の POSTGRES_PID_FILE 設定の記述	SRA

		を追加した。 ・フェイルオーバ実行機能が gfarm-2.7.17 以降のメタデータ read_only 機能を要求すること、および read_only 機能を利用する仕組みと設定を記載した。	
4.3	2021/10/04	・Zabbix 5.0 LTS に対応した。	

目次

1. はじめに

本ドキュメントは、Gfarm ファイルシステム（以降、Gfarm とする）における統合監視ソフトウェア Zabbix（http://www.zabbix.com/）で構成された障害監視システム(以降、gfarm_zabbix）を導入する際の、手順および設定について記載したものである。

Zabbix による障害監視システムの導入から、Gfarm の障害監視を行うための Zabbix の初期設定までを対象とする。導入後の管理・利用方法等については、「冗長化構成 Gfarm 監視機能 管理・利用マニュアル」を参照のこと。

なお、チケット管理システムのインストールに関しては、別途「異常時チケット登録機能」の「導入・設定マニュアル」を、運用に関しては「冗長化構成 Gfarm 監視機能 管理・利用マニュアル」を参照されたい。

本ドキュメントは、gfarm_zabbix バージョン 4.3 に対応している。

本ドキュメントの「7 フェイルオーバ実行機能の設定」に記載されているフェイルオーバ機能を利用するには gfarm-2.7.17 以降を必要とする。

本ドキュメントにおいて、root 権限でシェルに与えるコマンドはプロンプトとして「#」、一般ユーザー権限でシェルに与えるコマンドはプロンプトとして「$」を表記している。

1.1. gfarm_zabbix パッケージの構成

gfarm_zabbix パッケージは、以下の内容で構成されている。

- **Zabbix 向け Gfarm 監視モジュール**
  監視モジュールはさらに、Zabbix 向けの監視テンプレート、監視用の外部スクリプトおよびフェイルオーバスクリプトで構成される。これらを導入することで、Gfarm の稼働状況を Zabbix で監視できるようになる。また、フェイルオーバスクリプトを用いることで、Gfarm に障害が発生した祭、メタデータサーバを自動的にフェイルオーバできるようになる。
- **クライアント設定ファイル編集機能**
  Gfarm クライアントの設定ファイル（gfarm2.conf）を Web 上で編集する機能。
- **ドキュメント**
  Zabbix 向け Gfarm 監視モジュールや、クライアント設定ファイル編集機能に関するドキュメント。

1.2. 動作環境

gfarm_zabbix の動作環境は、次のものを対象としている。

表 1-1 動作環境の要件一覧

区分	要件
OS	Red Hat Enterprise Linux 6 または CentOS 6 または 7
Zabbix	1.8 系または 2.2 系または 5.0 LTS 系
監視対象とする Gfarm	2.5.8.13 以上

本書の解説は主に Zabbix 1.8 系を対象に行なっている。スクリーンショット等もすべて 1.8 系のものである。2.2 系および 5.0 LTS 系において操作手順が異なる場合、必要な箇所ではその旨を説明してあるが、多くは本書の読者が独力で容易に対処できる部分であり、特に説明はしていない。

2. システム構成

Gfarm に Zabbix を導入するにあたり、Zabbix の基本構成と Gfarm 上での構成について説明する。

2.1. Zabbix 基本構成

Zabbix は以下の要素により構成されている。

- **Zabbix サーバ**
  監視項目や収集した監視データを管理し、障害の検出や通知等を行う。監視項目や、収集した監視データを、データベース上に保存する。
- **Zabbix エージェント**
  監視対象ノード上で動作し、監視データの収集および Zabbix サーバへの送信を行う。
- **Zabbix Web インターフェース**
  監視項目の設定や監視データの閲覧等を行うための Web インターフェースを提供する。

以下に構成図を示す。

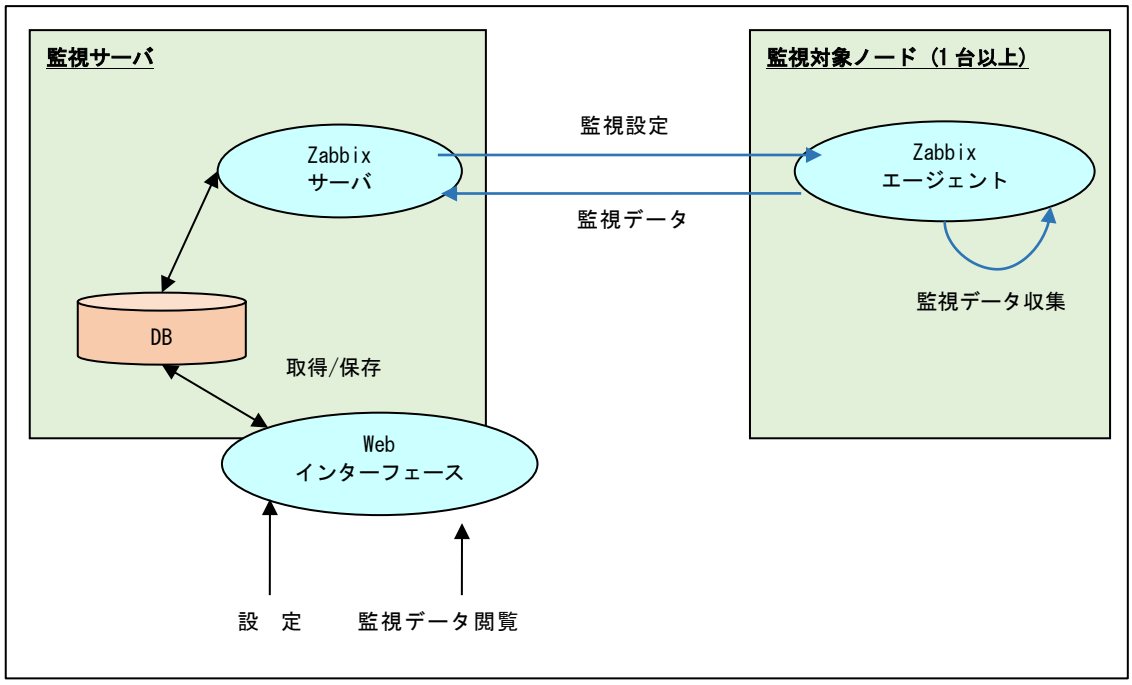


図 2-1 Zabbix 基本構成

2.2. Gfarm 構成

gfarm_zabbix では、Gfarm のサーバおよびクライアントを Zabbix の監視対象ノードとすることで Gfarm の監視を行う。具体的には、Gfarm 側は次のような構成となる。

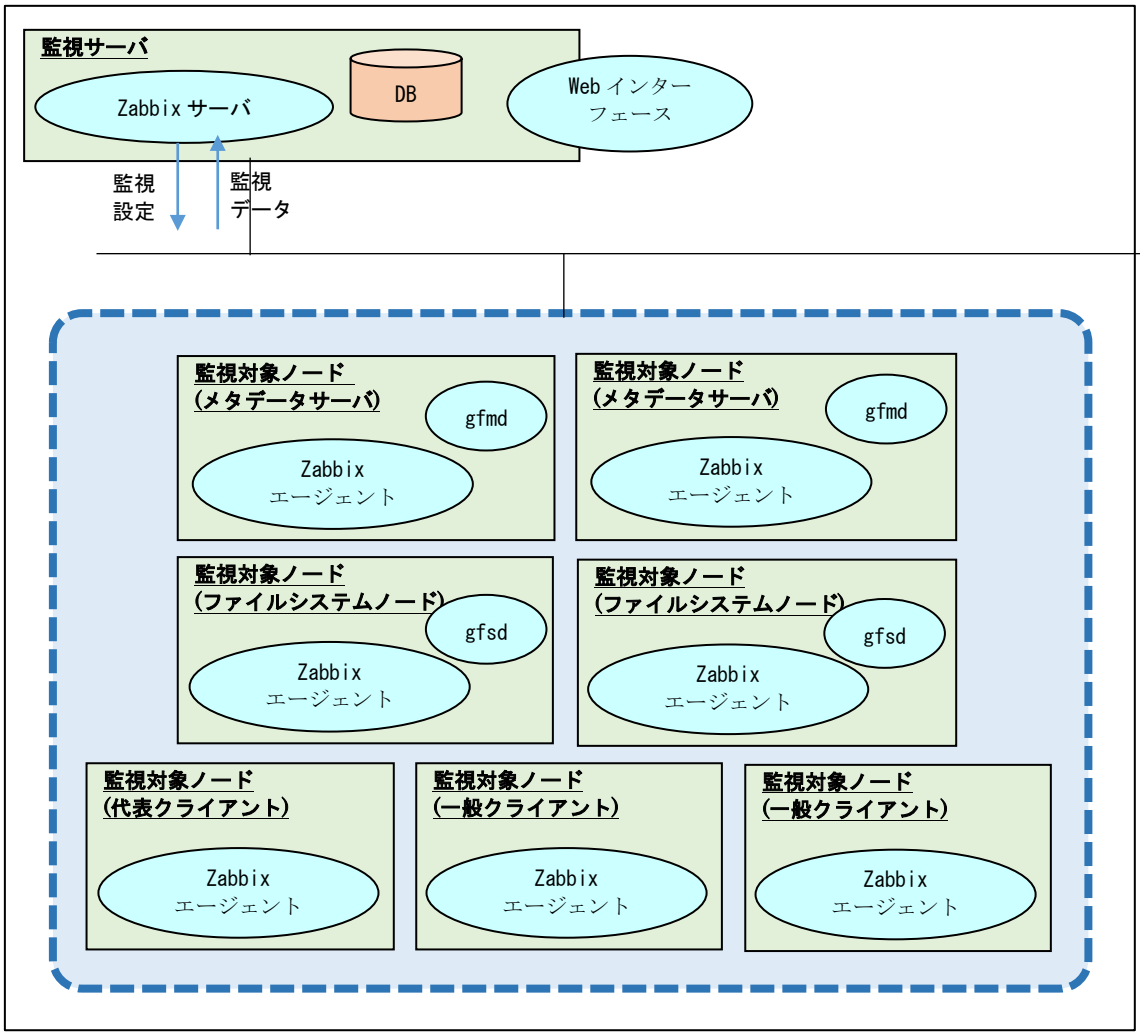


図 2-2 Gfarm 監視対象ノード構成

gfarm_zabbix では「代表クライアント」と「一般クライアント」の 2 種類のクライアント分類がある。代表クライアント機能は、クライアント監視の目的だけではなく、サーバの動作状況の監視の目的も兼ねているため、たとえ一般クライアントの監視を省略する場合でも、代表クライアントの監視については実施する必要がある。Gfarm のクライアントとして稼働しているホストの中から 1 台を代表クライアントとして選び、その他は一般クライアントという位置付けにする。あるいはクライアントのうちの 1 台ではなく、Zabbix サーバ、メタデータサーバ、ファイルシステムノードのうちのいずれか 1 台を代表クライアントとして兼任させても良い。

メタデータサーバやファイルシステムノードがアクセスするデータは、ルートディレクトリ直下の専用ディレクトリに置いて動作検証を実施している。それ以外の場所を選んだ場合、SELinux 関係で本書に記載のない追加設定が必要となる可能性がある。

また、Linux のバージョンの違い、Zabbix のバージョンの違い等が原因で SELinux の追加設定が必要となる場合もある。

SELinux 関係の追加設定に関しては「8.1 SELinux 環境での問題」を参照。

2.3. Zabbix 分散監視構成

gfarm_zabbix は、Zabbix による分散監視構成に対応している。この構成では、親子関係にある監視サーバ間（マスターノードー子ノード間）で相互監視を行うことで、監視サーバ自体の障害監視も行うことが可能である。以下に、ホスト構成を示す。

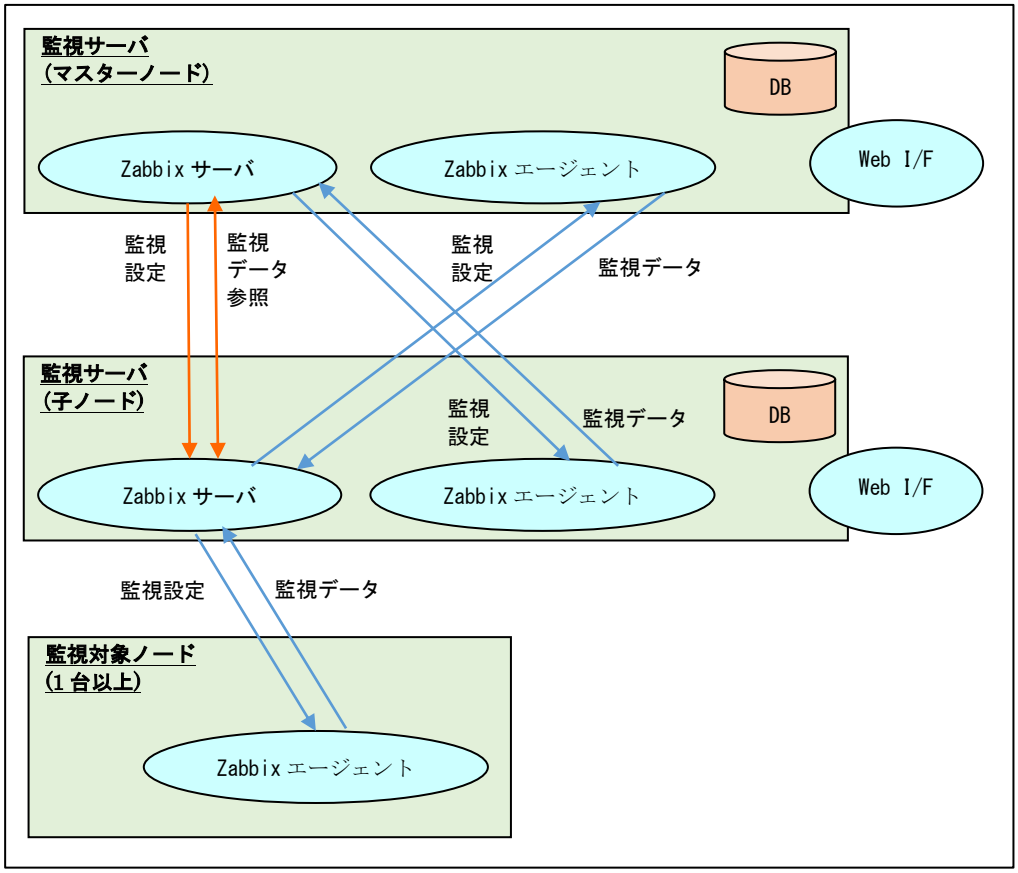


図 2-3 Zabbix 分散監視構成

監視サーバは、それぞれ以下のような役割を持つ。

- マスターノード
  全ての子ノードの監視設定および、監視データの参照を行う。
- 子ノード
  自身の配下の監視対象ノードの監視設定と、監視データを管理する。単独で動作可能であり、障害通知も行う。子ノード側の監視設定は、マスターノードでも閲覧や変更が可能であり、定期的に同期している

3. インストール

本章では、Zabbix サーバ、Zabbix エージェント、gfarm_zabbix パッケージのインストール手順について記載する。本ドキュメントで例示する Zabbix のバージョンは、1.8 系は 1.8.20、2.2 系は 2.2.5、5.0 LTS 系は 5.0.15 とする。また、Zabbix サーバおよび Zabbix Web インターフェースで利用するデータベースは、Zabbix で推奨されている MySQL (CentOS 7 では MariaDB) を利用するものとする。Web サーバーソフトウェアは Apache httpd を想定して記載している。

本章では最初にファイアウォールの設定について説明した後、インストールする必要のあるアプリケーションのインストール作業手順を記述してある。本章は確認する程度に読み進め、実際の作業は「4 各ノードの設定」に進み、その内容に応じて本章を参照して作業を行うこと。

3.1. ファイアウォールの設定

Zabbix および gfarm-zabbix パッケージを動作させるためには、次のようなネットワーク通信を許可する必要がある。なお宛先側のポート番号は、標準設定のものを載せているので、変更している場合は適宜読み替えること。

表 3-1 インストールするソフトウェア一覧

差出側アドレス		宛先側アドレス		用途
IP アドレス	ポート	IP アドレス	ポート	
Zabbix サーバ	ephemeral	Zabbix エージェント	10050/tcp	Zabbix エージェントへのアクセス
Zabbix エージェント	ephemeral	Zabbix サーバ	10051/tcp	Zabbix サーバへのアクセス
Zabbix サーバ	ephemeral	Zabbix サーバ	10051/tcp	分散監視のデータ送信（分散監視構成のみ)
Web I/F 利用クライアント	ephemeral	Web サーバ	80/tcp 443/tcp	HTTP、HTTPS による Web アクセス
Zabbix サーバ	ephemeral	gfmd	22/tcp	SSH（フェイルオーバスクリプト実行時のみ)

またこれとは別に、gfmd、gfsd、クライアント（代表クライアント、一般クライアント）間の通信がある。そちらの設定については、Gfarm 付属の SETUP.ja ファイルを参照のこと。

3.2. インストール対象とインストールするソフトウェア

インストールする必要のあるソフトウェアは、ノードの種類毎に異なる。

表 3-2 インストールするソフトウェア一覧

ノード	インストールするソフトウェア
監視サーバ	Zabbix サーバ Zabbix Web インターフェース Zabbix エージェント（分散監視構成の場合） gfarm_zabbix パッケージ
Gfarm メタデータサーバ	Zabbix エージェント gfarm_zabbix パッケージ
Gfarm ファイルシステムノード	Zabbix エージェント gfarm_zabbix パッケージ
代表クライアント	Zabbix エージェント gfarm_zabbix パッケージ
一般クライアント	Zabbix エージェント gfarm_zabbix パッケージ

3.3. Zabbix サーバのインストール

監視サーバ各機に対して、Zabbix サーバをインストールする手順を示す。なお、手順は全て root ユーザで実行する。

1. yum リポジトリ登録用 RPM を取得する。
   https://www.zabbix.com/download の「Choose your platform」で対象となるバージョンを選択し、表示された「Install Zabbix repository」項に従い実行する。以下にいくつかの CentOS バージョンおよび Zabbix バージョンでの例を挙げるが、バージョンの枝番も更新されている可能性があるので、インストール時点で確認する。
   [CentOS 6, Zabbix 1.8]:

```
# wget http://repo.zabbix.com/zabbix/1.8/rhel/6/x86_64/zabbix-release-1.8-1.el6.noarch.rpm
```

   [CentOS 7, Zabbix 2.2]:

```
# wget https://repo.zabbix.com/zabbix/2.2/rhel/7/x86_64/zabbix-release-2.2-1.el7.noarch.rpm
```

   [CentOS 7, Zabbix 5.0 LTS]:

```
# wget https://repo.zabbix.com/zabbix/5.0/rhel/7/x86_64/zabbix-release-5.0-1.el7.noarch.rpm
```

2. yum リポジトリ登録用 RPM をインストールする。
   RPM は、実際のファイル名に読み替えること。
   [CentOS 6, Zabbix 1.8]:

```
# rpm -ivh zabbix-release-1.8-1.el6.noarch.rpm
# yum clean all
```

[CentOS 7, Zabbix 2.2]:

```
# rpm -ivh zabbix-release-2.2-1.el7.noarch.rpm
# yum clean all
```

[CentOS 7, Zabbix 5.0 LTS]:

```
# rpm -ivh zabbix-release-5.0-1.el7.noarch.rpm
# yum clean all
```

3. MySQL / MariaDB サーバをインストールする。

[CentOS 6]:

```
# yum –y install mysql-server
```

[CentOS 7]:

```
# yum –y install mariadb-server
```

4. Apache HTTP Server、Zabbix サーバ、Zabbix Get パッケージをインストールする。

```
# yum -y install zabbix zabbix-server zabbix-server-mysql zabbix-get
```

5. Zabbix Web インターフェースをインストールする。

[Zabbix 2.0/2.2]

```
# yum -y install zabbix-web zabbix-web-mysql
```

[Zabbix 5.0 LTS]

```
# yum -y install yum-utils centos-release-scl
# yum-config-manager --enable zabbix-frontend
# yum -y install zabbix-web-mysql-scl zabbix-apache-conf-scl
```

6. /etc/my.cnf の変更を行う。(赤字の部分を追加する。)

```
[mysqld]
datadir=/var/lib/mysql
socket=/var/lib/mysql/mysql.sock
user=mysql
# Disabling symbolic-links is recommended to prevent assorted security risks
symbolic-links=0

character-set-server=utf8
skip-character-set-client-handshake

[mysqld_safe]
log-error=/var/log/mysqld.log
pid-file=/var/run/mysqld/mysqld.pid
```

7. MySQL / MariaDB サーバを起動する。

[CentOS 6]:

```
# service mysqld start
```

[CentOS 7]:

```
# systemctl start mariadb.service
```

8. Zabbix データベースと接続ユーザ zabbix を作成する。

[Zabbix 2.0/2.2]:

```
# mysqladmin create zabbix --default-character-set=utf8
# mysql -uroot
mysql> grant all privileges on zabbix.* to zabbix@localhost identified by 'zabbix';
mysql> flush privileges;
mysql> quit
```

[Zabbix 5.0 LTS]:

```
# mysql -uroot
MariaDB [(none)]> create database zabbix character set utf8 collate utf8_bin;
MariaDB [(none)]> grant all privileges on zabbix.* to zabbix@localhost identified by
'zabbix';
MariaDB [(none)]> flush privileges;
MariaDB [(none)]> quit
```

9. Zabbix の初期データをインポートする。

インストールした Zabbix サーバのバージョンに応じて、SQL ファイルのパスは適宜修正する。

[Zabbix 1.8]:

```
$ mysql -uroot zabbix ¥
  < /usr/share/doc/zabbix-server-mysql-1.8.20/create/schema/mysql.sql
$ mysql -uroot zabbix ¥
  < /usr/share/doc/zabbix-server-mysql-1.8.20/create/data/data.sql
$ mysql -uroot zabbix ¥
  < /usr/share/doc/zabbix-server-mysql-1.8.20/create/data/images_mysql.sql
```

[Zabbix 2.2]:

```
$ mysql -uroot zabbix ¥
  < /usr/share/doc/zabbix-server-mysql-2.2.5/create/schema.sql
$ mysql -uroot zabbix ¥
  < /usr/share/doc/zabbix-server-mysql-2.2.5/create/images.sql
$ mysql -uroot zabbix ¥
  < /usr/share/doc/zabbix-server-mysql-2.2.5/create/data.sql
```

[Zabbix 5.0 LTS]:

```
# zcat /usr/share/doc/zabbix-server-mysql*/create.sql.gz | ¥
  mysql -uzabbix -p zabbix
```

10. /etc/zabbix/zabbix_server.conf ファイルを編集し、データベース接続ユーザー zabbix のパスワードを設定する

```
DBPassword=zabbix
```

11. php の date.timezone を設定する。

[Zabbix 1.8/2.2]:

/etc/php.ini ファイルを編集し、[Date] セクション中に赤字で記した行を置く。

```
[Date]
; Defines the default timezone used by the date functions
; http://www.php.net/manual/en/datetime.configuration.php#ini.date.timezone
date.timezone = Asia/Tokyo
```

[Zabbix 5.0 LTS]:

/etc/zabbix/zabbix_server.conf ファイルを編集し、[zabbix] セクション中に赤字で記した行を置く。

```
php_value[date.timezone] = Asia/Tokyo
```

12. SELinux を有効にしている場合、gfarm-zabbix の配布に含まれる src/etc/zabbix-server-centos7.te を以下の手順で有効化する。

なおこのファイルは、OS や Zabbix のバージョンの違いに応じてユーザーがカスタマイズすることを想定し、バイナリ形式ではなくソース形式 *.te で提供している。カスタマイズの詳細は「8.1SELinux 環境での問題」を参照。

[CentOS 7, Zabbix 5.0 LTS]:

```
$ sudo yum -y install policycoreutils-python
$ cd src/etc
$ checkmodule -M -m -o zabbix-server-centos7.mod zabbix-server-centos7.te
$ semodule_package -o zabbix-server-centos7.pp -m zabbix-server-centos7.mod
$ sudo semodule -i zabbix-server-centos7.pp
```

13. IP パケットフィルタで HTTP、HTTPS のアクセスを制限している場合は、許可するように変更する。

[CentOS 6 (iptables を使用している場合)]:

iptables の設定ファイル /etc/sysconfig/iptables に下記（赤字の部分）を、他の “-A INPUT” 行よりも前に追加する。

```
# Firewall configuration written by system-config-firewall
```

```
# Manual customization of this file is not recommended.
*filter
:INPUT ACCEPT [0:0]
:FORWARD ACCEPT [0:0]
:OUTPUT ACCEPT [0:0]
-A INPUT -p tcp -m tcp --dport 443 -j ACCEPT
-A INPUT -p tcp -m tcp --dport 80 -j ACCEPT
-A INPUT -m state --state ESTABLISHED,RELATED -j ACCEPT
    (略)
COMMIT
```

下記コマンドで iptables を再起動する。

```
# service iptables restart
```

[CentOS 7 (firewalld を使用している場合)]:

```
# firewall-cmd --add-service=http –permanent
# firewall-cmd --add-service=https --permanent
```

14. サーバーを起動する。

[CentOS 6, Zabbix 1.8/2.2]:

```
# service httpd start
```

[CentOS 7, Zabbix 1.8/2.2]:

```
# systemctl start httpd.service
```

[CentOS 7, Zabbix 5.0 LTS]:

```
# systemctl restart httpd.service
```

15. ブラウザで下記 URI にアクセスする。

```
http://Zabbix サーバのホスト名/zabbix/
```

16. 「Introduction」画面が表示されるので、「Next」ボタンを押下する。

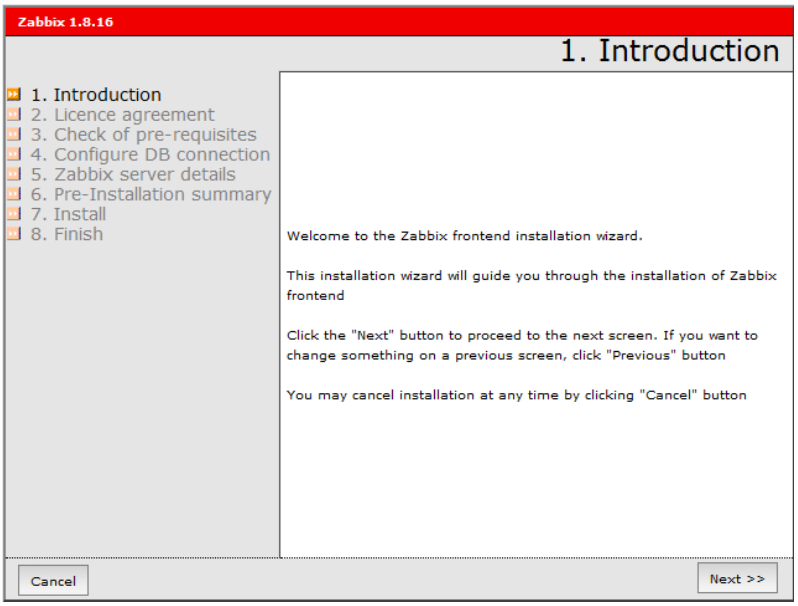


図 3-1 Introduction 画面

17. 「Licence agreement」画面が表示される。「I agree」をチェックし、「Next」ボタンを押下する。

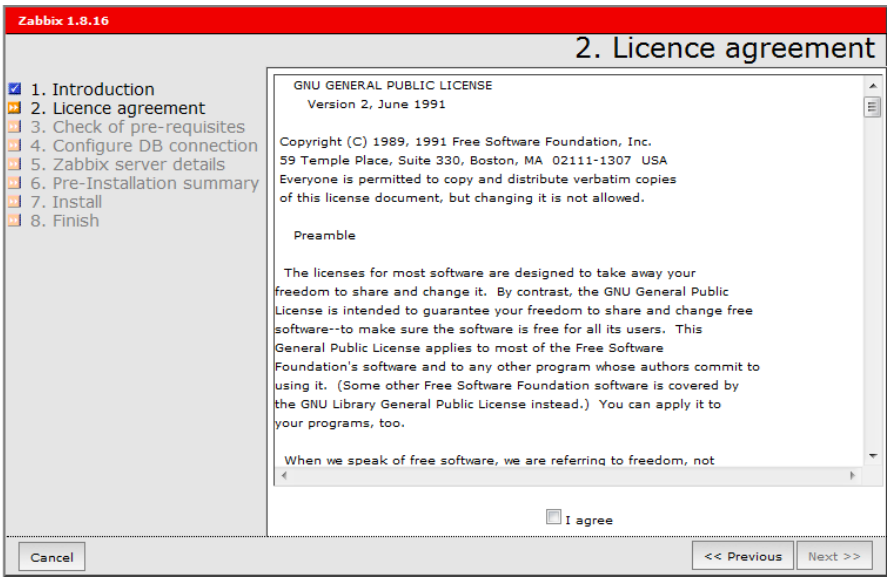


図 3-2 Licence agreement 画面

18. 「Check of pre-requisites」画面が表示される。「Next」ボタンを押下する。

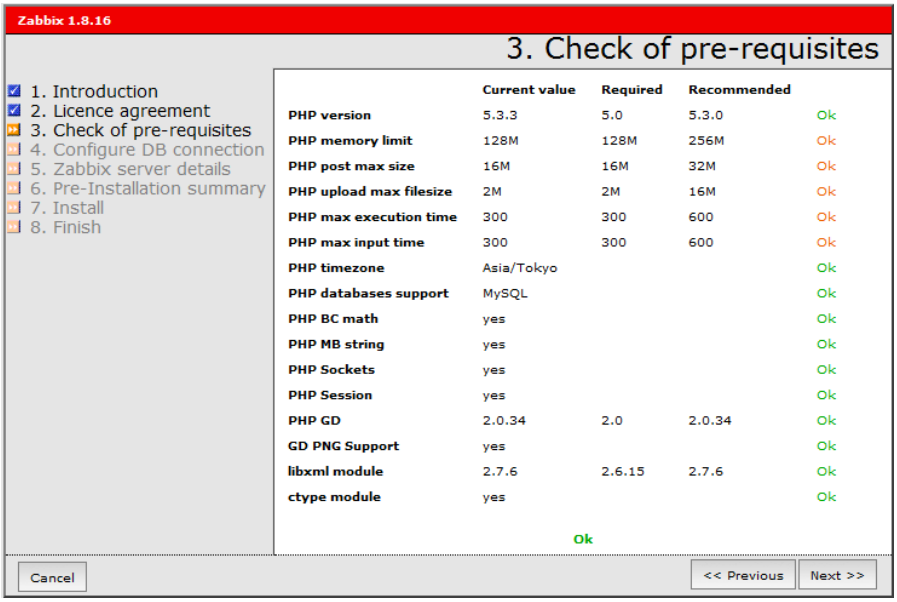


図 3-3 Check of pre-requisites 画面

19. 「Configure DB connection」画面が表示される。下記表の値を入力し、「Test connection」を押下し、「OK」が表示された後「Next」ボタンを押下する。

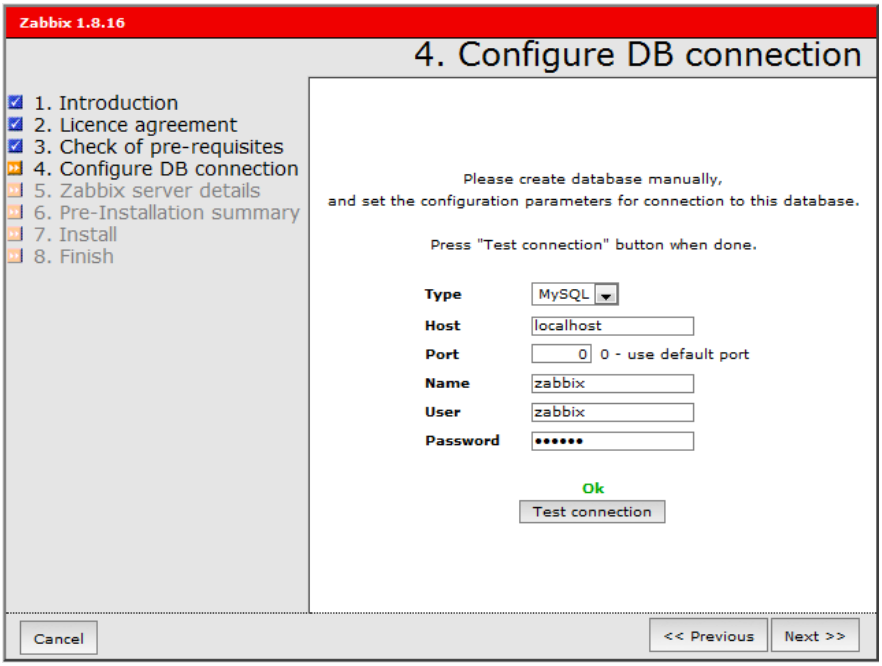


図 3-4 Configure DB connection 画面

各項目に入力すべき値は、次の通り。

表 3-3 DB 接続設定

設定項目	設定値

Type	MySQL
Host	localhost
Port	0
Name	zabbix
User	zabbix
Password	zabbix

20. 「Zabbix server details」画面が表示される。下記表の値を入力し、「Next」ボタンを押下する。

図 3-5 Zabbix server details 画面

各項目に入力すべき値は、次の通り。

表 3-4 Zabbix server 設定

設定項目	設定値
Host	localhost
Port	10051
Name	*監視サーバ名*

21. 「Pre-Installation Summary」画面が表示される。入力内容に間違いがないことを確認し、「Next」ボタンを押下する。

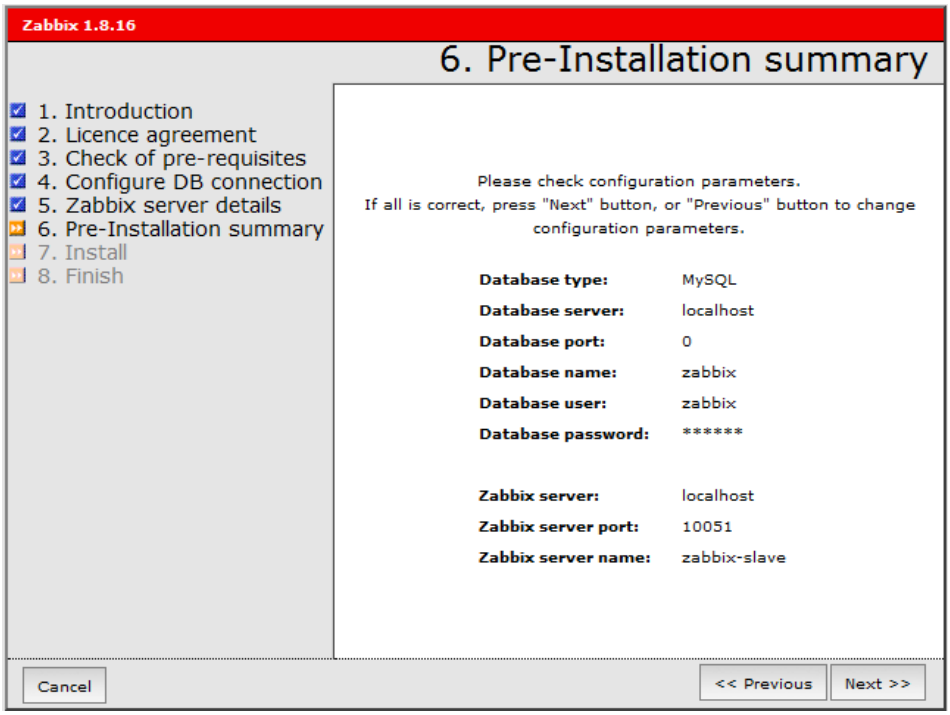


図 3-6 Pre-Installation summary 画面

SELinux を有効にしている場合はファイルを自動的に保存できないため、設定ファイルの配置に失敗したことを通知する「Install」画面が表示される。

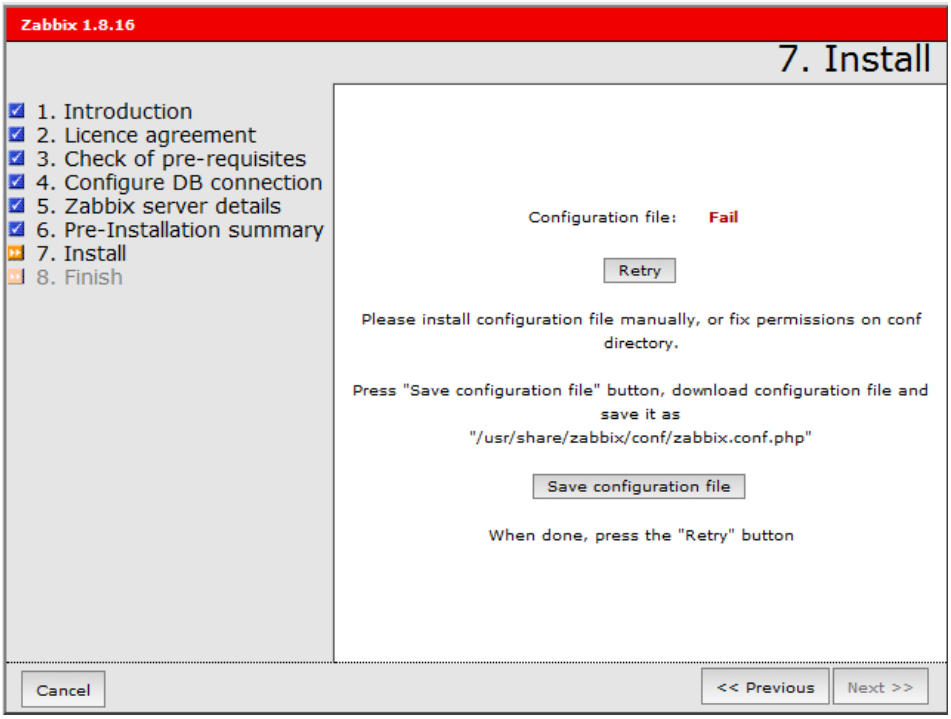


図 3-7 Install 画面

ファイルが自動的に保存された場合は、手順 24 に進む。

22. 「Save configuration file」ボタンを押下し、設定ファイルを任意の場所に保存する。
23. 設定ファイルを cp コマンドで配置し、「Retry」ボタンを押下する。

```
# cp zabbix.conf.php /etc/zabbix
```

24. 設定ファイル配置に成功した「Install」画面が表示される。「Next」ボタンを押下する。

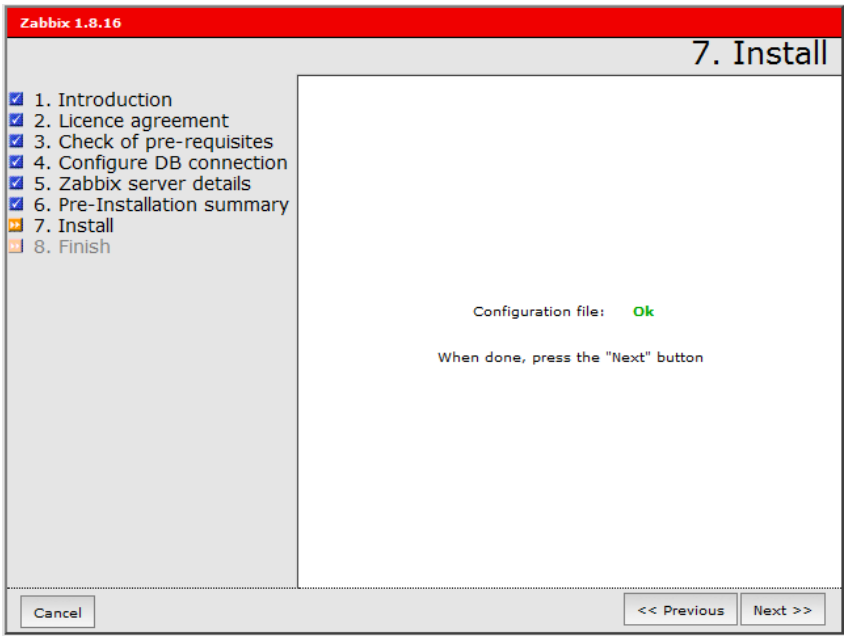


図 3-8 Install 画面

25. 「Finish」画面が表示される。「Next」ボタンを押下する。

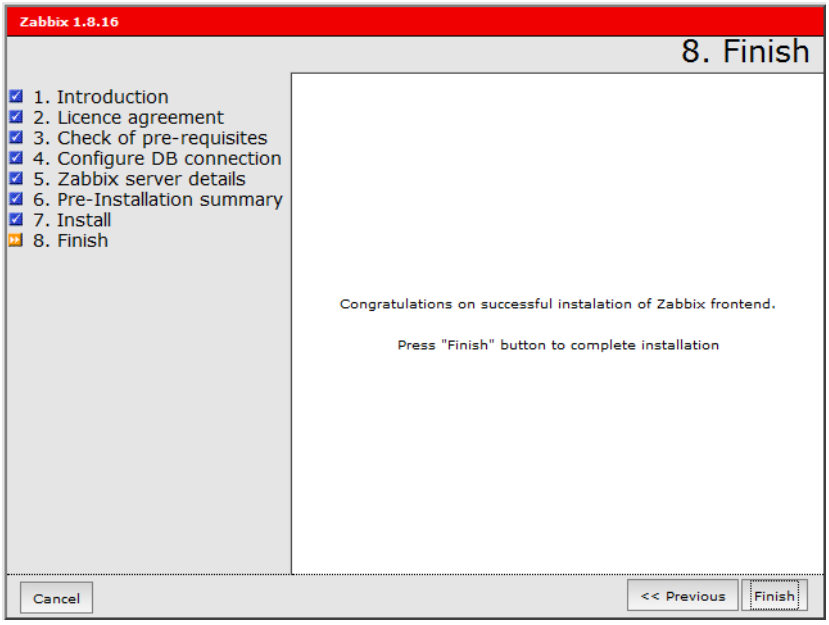


図 3-9 Finish 画面

26. Zabbix にログインできるかどうか確認する。

ブラウザでログイン画面:

http://*Zabbix サーバのホスト名*/zabbix/

にアクセスし、ユーザ「Admin」、パスワード「zabbix」でログインする。

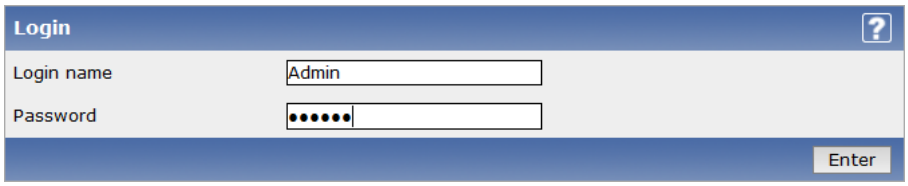


図 3-10 ログイン画面

ログインに成功すると、次のような画面が表示される。

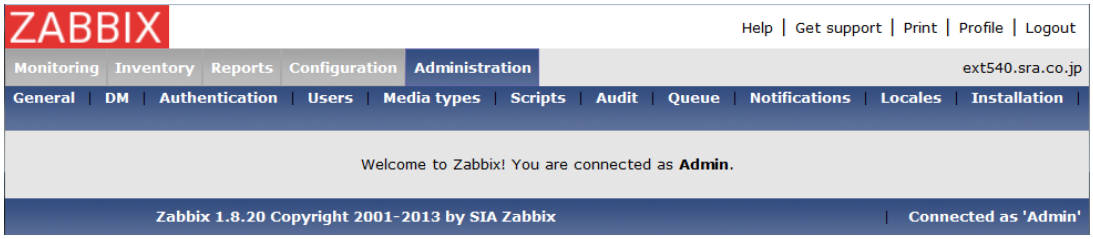


図 3-11 ログイン成功通知

27. パスワードを変更し、言語設定を日本語に変える。(省略可)

メニューの「Administration」－「Users」を選択すると、ユーザグループまたはユーザの一覧が表示される。右上の「Users」「User groups」を切り替えるプルダウンメニューがあるので、「Uesrs」を選択する。ユーザ一覧が表示されたら、Alias 欄の「Admin」をクリックする。

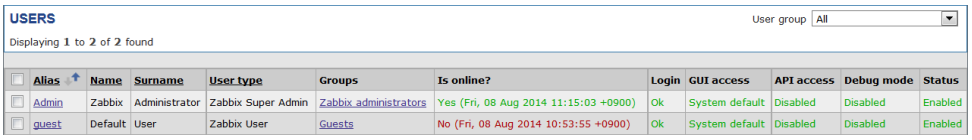


図 3-12 ユーザ一覧

Password 欄の「Change password」欄を選択し、必ずパスワードを変更する。また Admin 権限でのアクセスは、https プロトコルに限定すべきである。

Language に「Japanese (JP)」を選択し、「Save」を押下する。画面を再読み込みすると、言語が日本語に変わる。なお本書では以降、画面に使われている語句は、言語設定を日本語にしたときのもので表記する。

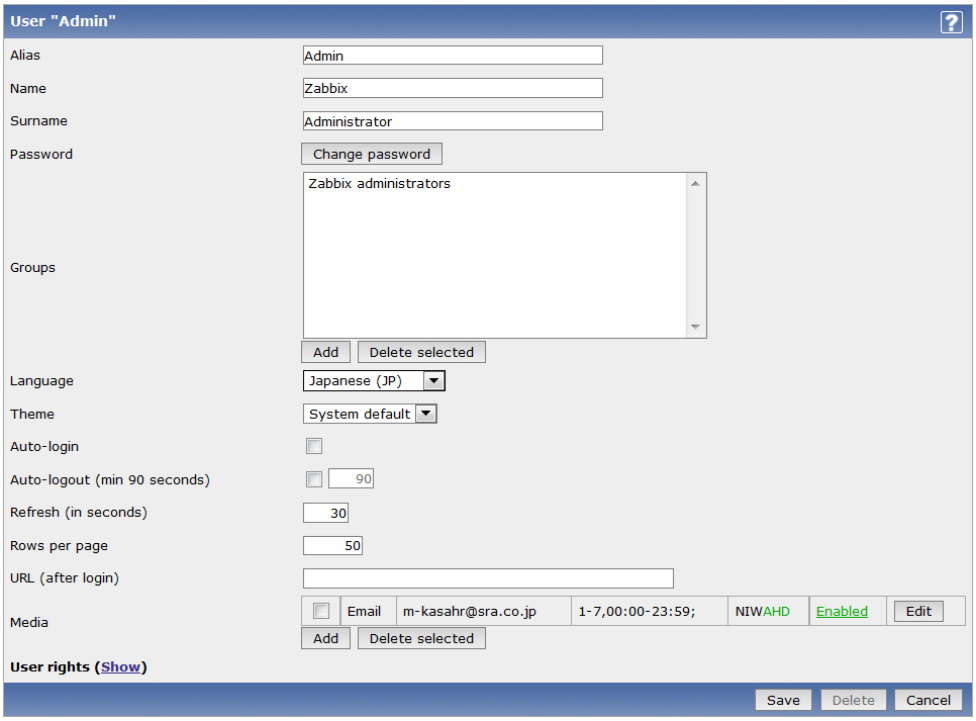


図 3-13 ユーザ設定画面

28. OS 起動時に MySQL / MariaDB サーバ、Apache HTTP Server が起動するように設定する。

[CentOS 6 Zabbix 1.8/2.2]:

```
# chkconfig --level 345 mysqld on
# chkconfig --level 345 httpd on
```

[CentOS 7 Zabbix 1.8/2.2]:

```
# systemctl enable mariadb.service
# systemctl enable httpd.service
```

[CentOS 7 Zabbix 5.0 LTS]:

```
# systemctl enable mariadb.service
# systemctl enable httpd.service
```

3.4. Zabbix エージェントのインストール

Zabbix エージェントのインストールは、監視対象ノード各機で行う。分散監視構成の場合は、加えて監視サーバ各機にもインストールする。以下に、インストール手順を示す。手順は全て root ユーザで実行する。

1. yum リポジトリ登録用 RPM を取得する。(同じホストに、先に Zabbix サーバをインストールしている場合、本工程は実施済みなので実行しなくて良い。)。

https://www.zabbix.com/download の「Choose your platform」で対象となるバージョンを選択し、表示された「Install Zabbix repository」項に従い実行する。以下にいくつかの CentOS バージョンおよび Zabbix バージョンでの例を挙げるが、バージョンの枝番も更新されている可能性があるので、インストール時点で確認する。

[CentOS 6, Zabbix 1.8]:

```
# wget http://repo.zabbix.com/zabbix/1.8/rhel/6/x86_64/zabbix-release-1.8-1.el6.noarch.rpm
```

[CentOS 7, Zabbix 2.2]:

```
# wget https://repo.zabbix.com/zabbix/2.2/rhel/7/x86_64/zabbix-release-2.2-1.el7.noarch.rpm
```

[CentOS 7, Zabbix 5.0 LTS]:

```
# wget https://repo.zabbix.com/zabbix/5.0/rhel/7/x86_64/zabbix-release-5.0-1.el7.noarchrpm
```

2. yum リポジトリ登録用 RPM をインストールする。(同じホストに、先に Zabbix サーバをインストールしている場合、本工程は実施済みなので実行しなくて良い。)
   RPM は、実際のファイル名に読み替えること。

   [CentOS 6, Zabbix 1.8]:

   ```
   # rpm -ivh zabbix-release-1.8-1.el6.noarch.rpm
   # yum clean all
   ```

   [CentOS 7, Zabbix 2.2]:

   ```
   # rpm -ivh zabbix-release-2.2-1.el7.noarch.rpm
   # yum clean all
   ```

   [CentOS 7, Zabbix 5.0 LTS]:

   ```
   # rpm -ivh zabbix-release-5.0-1.el7.noarch.rpm
   # yum clean all
   ```

3. Zabbix エージェントをインストールする。

   [CentOS 6, Zabbix 1.8] / [CentOS 7, Zabbix 2.2]:

   ```
   # yum -y install zabbix zabbix-agent
   ```

   [CentOS 7, Zabbix 5.0 LTS]:

   ```
   # yum -y install zabbix-agent
   ```

4. SELinux を有効にしている場合、gfarm-zabbix の配布に含まれる src/etc/zabbix-agent-gfarm-centos7.te を以下の手順で有効化する)。
   なおこのファイルは、OS や Zabbix のバージョンの違いに応じてユーザーがカスタマイズすることを想定し、バイナリ形式ではなくソース形式 *.te で提供している。カスタマイズの詳細は「8.1 SELinux 環境での問題」を参照。

   [CentOS 7, Zabbix 5.0 LTS]:

```
$ sudo yum -y install policycoreutils-python
$ cd src/etc
$ checkmodule -M -m -o zabbix-aent-gfarm-centos7.mod ¥
  zabbix-agent-gfarm-centos7.te
$ semodule_package -o zabbix-agent-gfarm-centos7.pp ¥
  -m zabbix-agent-gfarm-centos7.mod
$ sudo semodule -i zabbix-agent-gfarm-centos7.pp
```

6. zabbix ユーザの設定を変更する。ホームディレクトリを「/etc/zabbix」、シェルを「/bin/bash」、任意のパスワードを設定する。

```
# usermod -d /etc/zabbix -s /bin/bash -p password zabbix
```

7. /etc/zabbix の所有者を変更する。

```
# chown zabbix:zabbix /etc/zabbix
```

8. visudo コマンドで、zabbix ユーザの sudo 権限を設定する。

```
# visudo
```

/etc/sudoers ファイルに以下の 2 行を追加する。

```
Defaults:zabbix    !requiretty
zabbix   ALL=(_gfarmfs,_gfarmmd,postgres)  NOPASSWD: ALL
```

3.5. gfarm_zabbix パッケージのインストール

gfarm_zabbix パッケージ（“gfarm_zabbix-*バージョン番号*.tar.gz”というファイル）のインストールは、監視サーバ各機および監視対象ノード各機で行う。以下に、インストール手順を記す。なお、あらかじめ当該ホストでは、Gfarm ファイルシステムとしてのインストールおよび設定は完了しているものとする。

3.5.1. install.conf の編集

gfarm_zabbix パッケージを展開すると、src ディレクトリの下に install.conf というファイルがあるので、このファイルをエディタで編集する。このファイルは、シェルスクリプトとして解釈されるので、注意すること。このため、たとえば「=」の前後に空白を入れるとエラーになる。

```
# Gfarm のコマンド類（例 gfhost）が置かれているディレクトリ
GFARM_BINDIR=/usr/local/gfarm/bin

# config-gfarm コマンドの –prefix で指定した、gfmd データのトップディレクトリ
GFMD_CONFIG_PREFIX=/
```

```
# gfmd の設定ファイル
GFMD_CONF_FILE=$GFMD_CONFIG_PREFIX/gfmd.conf

# PostgreSQL デーモンの PID ファイル
POSTGRES_PID_FILE=$GFMD_CONFIG_PREFIX/var/gfarm-pgsql/postmaster.pid

# PostgreSQL デーモンを実行する UNIX ユーザー権限
POSTGRES_USER=postgres

# gfmd、gfsd、gfarm2fs のログメッセージが記録される syslog ファイル
GFARM_SYSLOG_FILE=/var/log/messages



# Zabbix がインストールされているディレクトリ。
# RPM からインストールした場合は /usr となる。
# zabbix_agent コマンドが ${ZABBIX_PREFIX}/sbin/zabbix_agent として存在すること。
ZABBIX_PREFIX=/usr

# Zabbix サーバの設定ファイルが置かれるディレクトリ
# 'zabbix' ユーザのホームディレクトリと同じでなくてはならない。
ZABBIX_CONFDIR=/etc/zabbix

# gfarm_zabbix が syslog にエラーを出力する際のファシリティ
ZABBIX_SYSLOG_FACILITY=local0

#########################################################
# クライアント設定ファイル編集機能向けの設定
#########################################################
# クライアント設定ファイル編集機能のインストールディレクトリ
EDITOR_HTMLDIR=/var/www/html/gfarm2-conf-editor

# $EDITOR_HTMLDIR ディレクトリの所有ユーザとグループ
EDITOR_HTMLDIR_USER=apache
EDITOR_HTMLDIR_GROUP=apache
```

3.5.2. Zabbix エージェント用ファイルのインストール

install.conf ファイルの置かれたディレクトリをカレントディレクトリとして、root 権限で以下のコマンドを実行する。

1. インストールを行う。

```
# ./install-agentd.sh
```

スクリプトの実行結果として、次のようなメッセージが出力される。

```
Install the file: /etc/zabbix/zabbix_agentd.d/userparameter_gfarm.conf
Install the file: /etc/zabbix/externalscripts/gfarm_generic_client_gfhost.sh
Install the file: /etc/zabbix/externalscripts/gfarm_generic_client_gfmdhost.sh
Install the file: /etc/zabbix/externalscripts/gfarm_gfmd_failover.pl
Install the file: /etc/zabbix/externalscripts/gfarm_gfmd_failover_agentd.pl
Install the file: /etc/zabbix/externalscripts/gfarm_gfmd_failover_common.pl
Install the file: /etc/zabbix/externalscripts/gfarm_gfmd_gfhost.sh
Install the file: /etc/zabbix/externalscripts/gfarm_gfmd_postgresql.sh
Install the file: /etc/zabbix/externalscripts/gfarm_gfmd_postgresql_alive.sh
Install the file: /etc/zabbix/externalscripts/gfarm_gfsd_gfhost.sh
Install the file: /etc/zabbix/externalscripts/gfarm_gfsd_gfsched.sh
Install the file: /etc/zabbix/externalscripts/gfarm_represent_client_gfhost.sh
Install the file: /etc/zabbix/externalscripts/gfarm_represent_client_gfmdhost.sh
Install the file: /etc/zabbix/externalscripts/gfarm_represent_client_gfmdhost2.sh
Install the file: /etc/zabbix/externalscripts/gfarm_utils.inc
Install the file: /etc/zabbix/externalscripts/gfarm_conf.inc

Set mode (= 0644) of the file: /var/log/messages
```

上記のメッセージにある通り、インストールスクリプトは syslog ファイル（上記では /var/log/messeges）のパーミッションを 0644 に変更する。これは zabbix-agentd が syslog ファイルから gfmd や gfsd の出力したログを検出するために、パーミッションを緩めている（ログファイルのローテーション時も、このパーミッションが維持される)。0644 では緩すぎるという場合は、適宜変更すること。ただし、zabbix ユーザの権限で読めるようにする必要がある。

「7 フェイルオーバ実行機能の設定」に記載されているフェイルオーバ機能は、gfarm-2.7.17 以降で提供されている設定ファイル gfmd.failover.conf および gfmd.failover.agent.conf の機能に依存している。もしメタデータ冗長化設定を行っているのにこれら 2 つの設定ファイルが存在しない場合、インストールスクリプトは警

告を発するので、メッセージに従い「config-gfarm-update --update」コマンドを実行することにより、この 2 つのファイルを作成する。また gfmd.failover.conf はその内容に「include gfmd.failover.agent.conf」を含む必要がある。もし既存の gfmd.failover.conf にこの行が含まれていない場合もインストールスクリプトは警告を発するので、メッセージに従いこの内容の行を追加する。

インストールスクリプトは、当該ノード上の $GFMD_CONF_FILE（この値は install.conf で指定）ファイルの内容を読み取って、PostgreSQL へのアクセス情報を $ZABBIX_CONFDIR/externalscripts/ ディレクトリ（同上）の下の gfarm_conf.inc というファイルに転記する。PostgreSQL のアクセス情報が正しく書き込まれたか、念のためファイルを確認すること。

```
# config-gfarm コマンドの –prefix で指定した、gfmd データのトップディレクトリ
GFMD_CONFIG_PREFIX=/

# syslog ファシリティ
SYSLOG_FACILITY=local0

# PostgreSQL デーモンの PID ファイル
POSTGRES_PID_FILE=

# PostgreSQL デーモンを実行する UNIX ユーザー権限
POSTGRES_USER=

# PostgreSQL サーバのホスト名
PGHOST=mds-master

# PostgrSQL サーバの TCP ポート番号
PGPORT=10602

# PostgreSQL データベース名
PGDATABASE=gfarm

# PostgreSQL へ接続する際のユーザ名
PGUSER=gfarm
```

```
# PostgreSQL へ接続する際のパスワード
PGPASSWORD="MycWlXdJpvyhV52NpMxQzZX3QiJdP=GRCzv2MJCXQBH"
```

なお、本ファイルの設定項目は、インストール後に手で編集しても支障なく、その時点からその設定が有効になる。PostgreSQL の接続情報を変えた場合は、忘れずに更新すること。

3.5.3. クライアント設定ファイル編集機能のインストール

クライアント設定ファイル編集機能は、必要な場合のみインストールを行う。クライアント設定ファイル編集機能に関しての詳細は「冗長化構成 Gfarm 監視機能 管理・利用マニュアル」を参照のこと。

クライアント設定ファイル編集機能は、監視サーバだけにインストールする。分散構成では、子ノードのほうの監視サーバにインストールする。install.conf ファイルを編集した後、install.conf ファイルの置かれたディレクトリをカレントディレクトリとして、root 権限で以下のコマンドを実行する。

1. インストールを行う。

```
# ./install-editor.sh
```

スクリプトの実行結果として下記が出力されるので、出力された内容に従い zabbix ユーザの crontab ファイルまたは/etc/cron.d/ディレクトリ下のファイルに /etc/zabbix/gfmdlist.sh を定期的に実行する設定を行う。

```
Install the file: /var/www/html/gfarm2-conf-editor/common.php
Install the file: /var/www/html/gfarm2-conf-editor/download.php
Install the file: /var/www/html/gfarm2-conf-editor/edit.php
Install the file: /var/www/html/gfarm2-conf-editor/index.php
Install the file: /var/www/html/gfarm2-conf-editor/save.php
Install the file: /etc/zabbix/gfmdlist.sh
Please add the following lines to a crontab file of user 'zabbix':

    # Run 'gfmdhost -l' every five minutes.
    */5 * * * * /etc/zabbix/gfmdlist.sh

or add the following lines to a file under /etc/cron.d/:

    # Run 'gfmdhost -l' every five minutes.
*/5 * * * * zabbix /etc/zabbix/gfmdlist.sh
```

4. 各ノードの設定

本章では、各ノードの設定を記載する。

4.1. zabbix ユーザの登録と共通認証鍵の作成

Gfarm 上に zabbix ユーザを登録する。

以下の手順は、Gfarm クライアントとして動作している任意のホスト 1 台を選び、Gfarm の gfarmadm グループ権限を持つユーザで実行する。

1. Gfarm 上に zabbix ユーザを作成する。

```
$ gfuser -c zabbix zabbix "/home/zabbix" ""
```

zabbix ユーザが作成されたことを確認する。

```
$ gfuser -l zabbix
zabbix:zabbix:/home/zabbix:
```

2. Gfarm 共有認証鍵を生成する。

zabbix ユーザの Gfarm 共有認証鍵の作成を行う。Gfarm クライアントとして動作している任意のホスト上にて、zabbix ユーザで実行する。

-p オプションでは、鍵の有効期限を秒数で指定する。期限が切れると無効になるため、鍵を再生成し、すべての監視サーバーおよび監視対象ノードにコピーし直す必要が生じるため、保守予定日を勘案し、適切な秒数を指定する。

```
$ gfkey -f -p 31536000
```

3. 指定した有効期限で Gfarm 共有認証鍵が生成されたことを確認する

```
$ gfkey –e
expiration time is Fri May 10 06:09:14 2013
```

4. 作成した Gfarm 共有認証鍵を、すべての監視サーバおよび監視対象ノードにコピーする。

```
$ scp -p /etc/zabbix/.gfarm_shared_key zabbix@ホスト名:/etc/zabbix
```

4.2. 監視サーバの設定

監視サーバの設定を行う。

4.2.1. Zabbix サーバの設定

Zabbix サーバの設定を行う。下記の手順は全て root ユーザで実行する。

1. Zabbix サーバの設定ファイル/etc/zabbix/zabbix_server.conf を編集する。(赤字の箇所は、注意して設定する必要がある。) なお、NodeID のデフォルト値は 0 なので、仕様に従えば設定を省略できる筈だが、Zabbix 1.8.20 を利用している場合は再起動時にエラー

が発生するため、NodeID は設定しておくことを薦める。

```
NodeID=0
LogFile=/var/log/zabbix/zabbix_server.log
LogFileSize=0
PidFile=/var/run/zabbix/zabbix_server.pid
DBName=zabbix
DBUser=zabbix
DBPassword=zabbix
Timeout=30
AlertScriptsPath=/etc/zabbix/alertscripts
ExternalScripts=/etc/zabbix/externalscripts
```

2. Zabbix サーバを起動する。

[CentOS 6]:

```
# service zabbix-server start
```

[CentOS 7]:

```
# systemctl resstart zabbix-server.service
```

3. IP パケットフィルタで Zabbix サーバのアクセスを制限している場合は、許可するよう変更する。

[CentOS 6 (iptables を使用している場合)]:

iptables の設定ファイル /etc/sysconfig/iptables の下記（赤字の部分）を、他の “-A INPUT” 行よりも前に追加する。

```
# Firewall configuration written by system-config-firewall
# Manual customization of this file is not recommended.
*filter
:INPUT ACCEPT [0:0]
:FORWARD ACCEPT [0:0]
:OUTPUT ACCEPT [0:0]
-A INPUT -p tcp -m tcp --dport 10051 -j ACCEPT
-A INPUT -p tcp -m tcp --dport 443 -j ACCEPT
    (略)
COMMIT
```

iptables を再起動する。

```
# service iptables restart
```

[CentoS 7 (firewalld を使用している場合)]:

```
# firewall-cmd --add-port=10050/tcp –permanent
# firewall-cmd --add-port=10051/tcp –permanent
```

```
# firewall-cmd --reload
```

4. OS 起動時に Zabbix サーバが起動するように設定する。

   [CentOS 6]:

```
# chkconfig --level 345 zabbix-server on
```

   [CentOS 7]:

```
# systemctl enable zabbix-server.service
```

5. SELinux 環境では、httpd プロセスからのネットワーク接続を許可する。

```
# setsebool -P httpd_can_network_connect 1
```

4.2.2. Zabbix エージェントの設定 (分散監視構成の場合)

分散監視構成では監視サーバ上でも Zabbix エージェント動作させることになるので、その設定を行う。下記の手順は全て root ユーザで実行する。

1. Zabbix エージェントの設定ファイル/etc/zabbix/zabbix_agentd.conf を編集する。(赤字の箇所は、注意して設定する必要がある。)

```
PidFile=/var/run/zabbix/zabbix_agentd.pid
LogFile=/var/log/zabbix/zabbix_agentd.log
LogFileSize=0
Server=192.168.0.1,192.168.0.2      ← 各監視サーバの IP アドレス
Hostname=zabbix-master              ← この監視サーバの GUI 表示ホスト名
ListenIP=0.0.0.0
Timeout=30
Include= /etc/zabbix/zabbix_agentd.d/
```

2. Zabbix エージェントを起動する。

   [CentOS 6]:

```
# service zabbix-agent start
```

   [CentOS 7]:

```
# systemctl start zabbix-agent.service
```

3. IP パケットフィルタで Zabbix エージェントのアクセスを制限している場合は、許可するよう変更する。

   [CentOS 6 (iptables を使用している場合)]:

   /etc/sysconfig/iptables の下記（赤字の部分）を、他の “-A INPUT” 行よりも前に追加する。

```
# Firewall configuration written by system-config-firewall
# Manual customization of this file is not recommended.
*filter
```

```
:INPUT ACCEPT [0:0]
:FORWARD ACCEPT [0:0]
:OUTPUT ACCEPT [0:0]
-A INPUT -p tcp -m tcp --dport 10050 -j ACCEPT
-A INPUT -p tcp -m tcp --dport 10051 -j ACCEPT
      (略)
COMMIT
```

iptables を再起動する。

```
# service iptables restart
```

[CentOS 7 (firewalld を使用している場合)]:

```
# firewall-cmd --add-port=10050/tcp –permanent
# firewall-cmd --add-port=10051/tcp –permanent
# firewall-cmd --reload
```

4. OS 起動時に Zabbix エージェントが起動するように設定する。

[CentOS 6]:

```
# chkconfig --level 345 zabbix-agent on
```

[CentOS 7]:

```
# systemctl enable zabbix-agent.service
```

4.2.3. クライアント設定ファイル編集機能の設定

クライアント設定ファイル編集機能を使用する場合は、監視サーバ（分散監視構成の場合は子ノードのほう）で下記の設定を行う。

1. apache ユーザが、端末を持たない状態でも sudo で任意のコマンドを管理者権限で実行できるよう、visudo で設定する。

```
# visudo
```

/etc/sudoers ファイルに以下の 2 行を追加する。

```
Defaults:apache      !requiretty
apache   ALL=(ALL)   NOPASSWD: ALL
```

4.3. 監視サーバ以外の設定

Gfarm メタデータサーバ、Zabbix ファイルシステムノード、代表クライアント、一般クライアント各機に対して、本節の設定を行う。

4.3.1. Zabbix エージェントの設定

Zabbix サーバの設定を行う。下記の手順は全て root ユーザで実行する。

1. Zabbix エージェントの設定ファイル/etc/zabbix/zabbix_agentd.conf を編集する。
   (赤字の箇所は注意して設定する必要がある。)

```
PidFile=/var/run/zabbix/zabbix_agentd.pid
LogFile=/var/log/zabbix/zabbix_agentd.log
LogFileSize=0
Server=192.168.0.2          ← 監視サーバの IP アドレス
                              (分散監視構成では子ノードのほうを指定)
ServerActive=192.168.0.2    ← 監視サーバの IP アドレス
                              (分散監視構成では子ノードのほうを指定)
Hostname=fsn1               ←このホストの表示ホスト名
ListenIP=0.0.0.0
Timeout=30
Include= /etc/zabbix/zabbix_agentd.d/
```

2. Zabbix エージェントを起動する。
   [CentOS 6]:

```
# service zabbix-agent start
```

   [CentOS 7]:

```
# systemctl o¥mstart zabbix-agent.service
```

3. IP パケットフィルタで Zabbix エージェントのアクセスを制限している場合は、許可するよう変更する。
   [CentOS 6 (iptables を使用している場合)]:
   /etc/sysconfig/iptables の下記（赤字の部分）を、他の “-A INPUT” 行よりも前に追加する。

```
# Firewall configuration written by system-config-firewall
# Manual customization of this file is not recommended.
*filter
:INPUT ACCEPT [0:0]
:FORWARD ACCEPT [0:0]
:OUTPUT ACCEPT [0:0]
-A INPUT -p tcp -m tcp --dport 10050 -j ACCEPT
-A INPUT -p tcp -m tcp --dport 10051 -j ACCEPT
     (略)
COMMIT
```

   iptables を再起動する。

```
# service iptables restart
```

[CenOS 7 (firewalld を使用している場合)]:

```
# firewall-cmd --add-port=10050/tcp –permanent
# firewall-cmd --add-port=10051/tcp –permanent
# firewall-cmd --reload
```

4. OS 起動時に Zabbix エージェントが起動するように設定する。

[CentOS 6]:

```
# chkconfig --level 345 zabbix-agent on
```

[CentOS 7]:

```
# systemctl ddate zabbix-agent.service
```

4.3.2. gfarm_zabbix スクリプトの設定

必要に応じて、監視サーバおよび監視対象ノード各機の /etc/zabbix/externalscripts/gfarm_conf.inc ファイルを編集する。特に、メタデータサーバ機では PostgreSQL のアクセス情報がこのファイルに転記されているので、正しい情報が記載されているか確認すること。

```
# syslog ファシリティ
SYSLOG_FACILITY=local0

# PostgreSQL サーバのホスト名
PGHOST=mds-master

# PostgrSQL サーバの TCP ポート番号
PGPORT=10602

# PostgreSQL データベース名
PGDATABASE=gfarm

# PostgreSQL へ接続する際のユーザ名
PGUSER=gfarm

# PostgreSQL へ接続する際のパスワード
PGPASSWORD="MycWlXdJpvyhV52NpMxQzZX3QiJdP=GRCzv2MJCXQBH"
```

4.4. zabbix_get による動作確認

ここまでの設定が正しいかどうかを確認するには、zabbix_get コマンドを用いると便利

である。zabbix_get コマンドは、Zabbix サーバ（分散監視構成の場合は、子ノードのほう）上で実行する必要がある。また Zabbix サーバに複数の IP アドレスが付与されている場合には、zabbix_agentd.conf の「Server=」設定にある IP アドレスをソース IP アドレスに用いて問い合わせを行う必要があるため、zabbix_get コマンドの –s オプションや –I オプションを適切に指定する。SELinux を有効にしている環境で、どうしてもうまくいかない場合は、「8.1 SELinux 環境での問題」を参照のこと。

1. zabbix_get コマンドを実行する。

$ zabbix_get -s *監視対象ノード* -p 10050 -k *監視アイテム名*

ここで「監視対象ノード」には、ノードのホスト名もしくは IP アドレスを指定する。監視対象ノード上では、Zabbix エージェントが動作していなければならない。「監視アイテム名」には、様々なものが指定できるが、代表的なものを挙げておく。

表 4-1 代表的な監視アイテム

監視アイテム名	監視対象ノード種別
	説明
gfarm.gfmd.gfhost	メタデータサーバ
	監視対象ノード上にて、_gfarmmd ユーザで gfhost コマンドを実行して成功するかどうかを確認する。成功すると “ok” が、失敗すると失敗理由がそれぞれ表示される。 監視対象ノード上において、ユーザ zabbix で gfarm_gfmd_gfhost.sh を実行するのと同じである。
gfarm.gfmd.postgresql.alive	メタデータサーバ
	監視対象ノード上にて、postgresql ユーザ権限で PostgreSQL デーモンのプロセス ID を指定してシグナルを送ることにより、デーモンプロセスが存在するかどうかを確認する。成功すると “ok”が、失敗すると失敗理由がそれぞれ表示される。 監視対象ノード上において、ユーザーzabbix で gfarm_gfmd_postgresql_alive.sh を実行するのと同じである。
gfarm.gfsd.gfhost	ファイルシステムノード
	監視対象ノード上にて、_gfarmfs ユーザで

	“gfhost-lv” コマンドを実行して成功するかどうかを確認する。成功すると “ok” が、失敗すると失敗理由がそれぞれ表示される。 監視対象ノード上において、ユーザ zabbix で gfarm_gfsd_gfhost.sh を実行するのと同じである。
gfarm.represent_client.gfmdhost	代表クライアント
	監視対象ノード上にて、“gfmdhost -l” コマンドを実行して成功するかどうかを確認する。成功すると “ok” が、失敗すると失敗理由がそれぞれ表示される。 監視対象ノード上におて、ユーザ zabbix で gfarm_represent_client_gfmdhost.sh を実行するのと同じである。
gfarm.represent_client.gfmdhost2	代表クライアント
	監視対象ノード上にて、“gfmdhost –N -1 –D *メタデータサーバホスト*” コマンドを各メタデータサーバに対してそれぞれ実行する。マスターメタデータサーバが 2 台以上立ち上がっていなければ成功となり、“ok” が表示される。2 台以上立ち上がっている場合は、エラーメッセージが表示される。 監視対象ノード上におて、ユーザ zabbix で gfarm_represent_client_gfmdhost2.sh を実行するのと同じである。
gfarm.generic_client.gfhost	一般クライアント
	監視対象ノード上にて、“gfhost -lv” コマンドを実行して、認証が通るかどうかを確認する。成功すると “ok” が、失敗すると失敗理由がそれぞれ表示される。 監視対象ノード上におて、ユーザ zabbix で gfarm_generic_client_gfhost.sh を実行するのと同じである。
proc.num[*プロセス名*]	全ノード種別
	監視対象ノード上にて、動作中の「*プロセス名*」の個数を表示する。たとえば

	“proc.num[gfmd]” とすれば、動作中の gfmd のプロセス数が表示される。

5. 監視設定

本節では、Zabbix の監視設定について記載する。Zabbix サーバの設定、Zabbix エージェントの設定および、監視項目の設定方法について記載する。分散監視構成の場合は、次章「6 分散監視構成設定」の設定も合わせて行うこと。

本章の記載内容は、初期導入時向けである。設定の変更や監視項目の追加を行う場合には、別途ドキュメント「冗長化構成 Gfarm 監視機能 管理・利用マニュアル」を参照のこと。

5.1. 監視項目の設定

Zabbix での監視項目の設定は、全て Web インターフェース上で行う。Zabbix では、以下の項目を設定することにより監視を行う。

表 5-1 設定項目一覧

設定項目	説明
ホスト	監視対象の設定。 Gfarm 監視では、Gfarm メタデータサーバ、Gfarm ファイルシステムノード、Gfarm クライアントノード、監視サーバ(相互監視用)をホストとして設定。
ホストグループ	監視対象(ホスト)をグループ化する設定。 Gfarm 監視では、Gfarm ファイルシステムを 1 ホストグループとして設定。
アイテム	監視項目の設定。 Zabbix サーバが各監視対象から収集する監視情報を設定。
トリガー	収集した監視情報に対して、障害検知する際の閾値の設定。
アクション	障害発生時の障害通知やスクリプト実行等の設定。

Zabbix では、監視項目をテンプレート化して管理する機能を有している。テンプレートには、各種アイテム/トリガーの設定が記述してある。gfarm_zabbix パッケージでは Gfarm 監視用のテンプレートを用意しており、本書ではこのテンプレートを利用した設定手順について記載する。

次節より、Gfarm 監視設定における各手順について説明する。

5.1.1. Gfarm 監視用テンプレートの導入

Gfarm 監視用テンプレートの導入手順を以下に示す。

1. Web インターフェースへのログイン
   (冗長構成の場合は、子ノードのほうの) 監視サーバの Web インターフェースにアクセスし、Admin ユーザでログインする。
2. テンプレート設定画面
   メニューの「設定」－「テンプレート」からテンプレート設定画面を表示する。

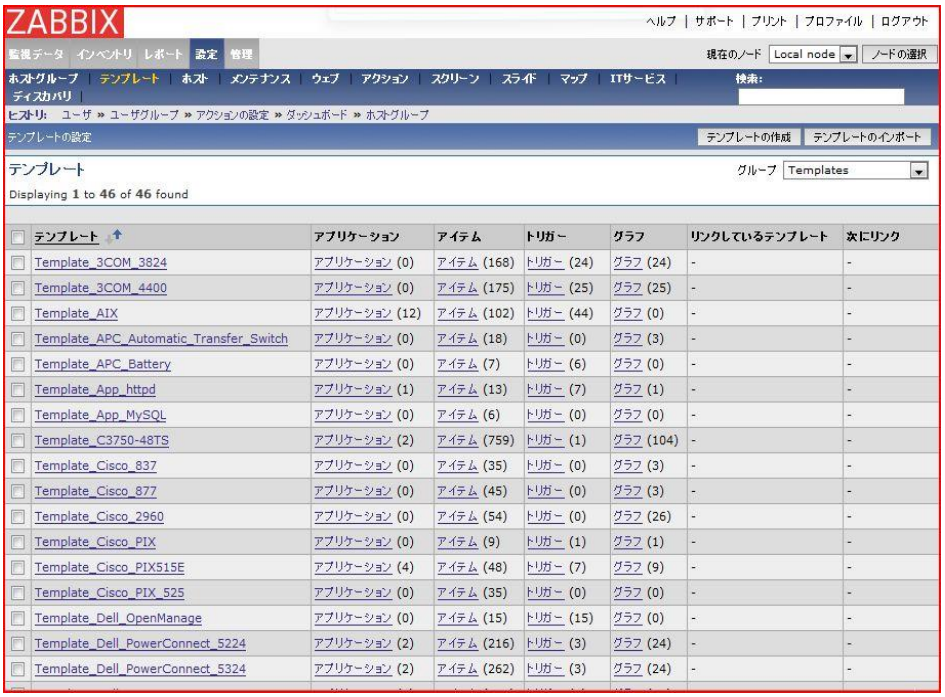


図 5-1 テンプレート設定画面

3. インポート画面
   「テンプレートのインポート」ボタン（Zabbix 2.0 以降では「インポート」ボタン）を押下し、インポート画面を表示する。

図 5-2 インポート画面

4. gfarm_zabbix パッケージを展開したディレクトリ下にあるファイル src/templates/Template_Gfarm_exported_all.xml を選択し、「インポート」ボタンを押下する。成功メッセージが表示されることを確認する。

5.1.2. ホストグループの設定

ホストグループの設定手順を以下に示す。

1. ログイン
   (冗長構成の場合は、子ノードのほうの) 監視サーバの Web インターフェースにアクセスし、Admin ユーザでログインする。
2. ホストグループ設定画面
   「設定」－「ホストグループ」からホストグループ設定画面を表示する。

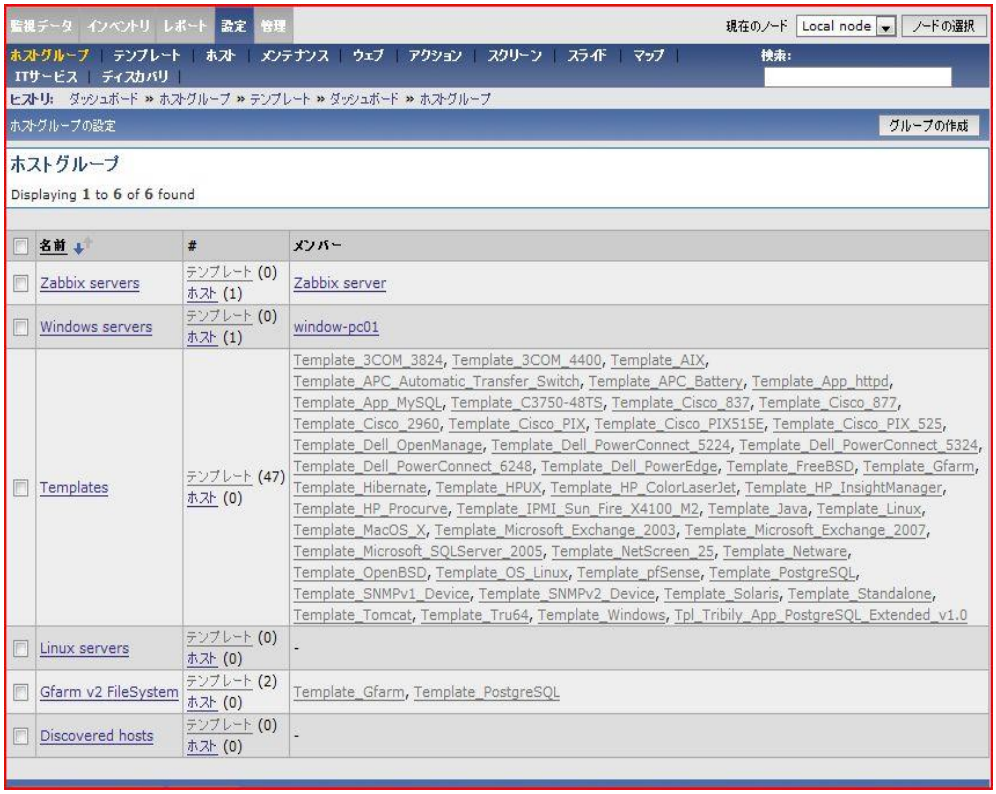


図 5-3 ホストグループ設定画面

3. ホストグループ作成画面

「グループの作成」ボタン(Zabbix 5.0 LTS では「ホストグループの作成」ボタン)を押下し、ホストグループ設定画面を表示する。

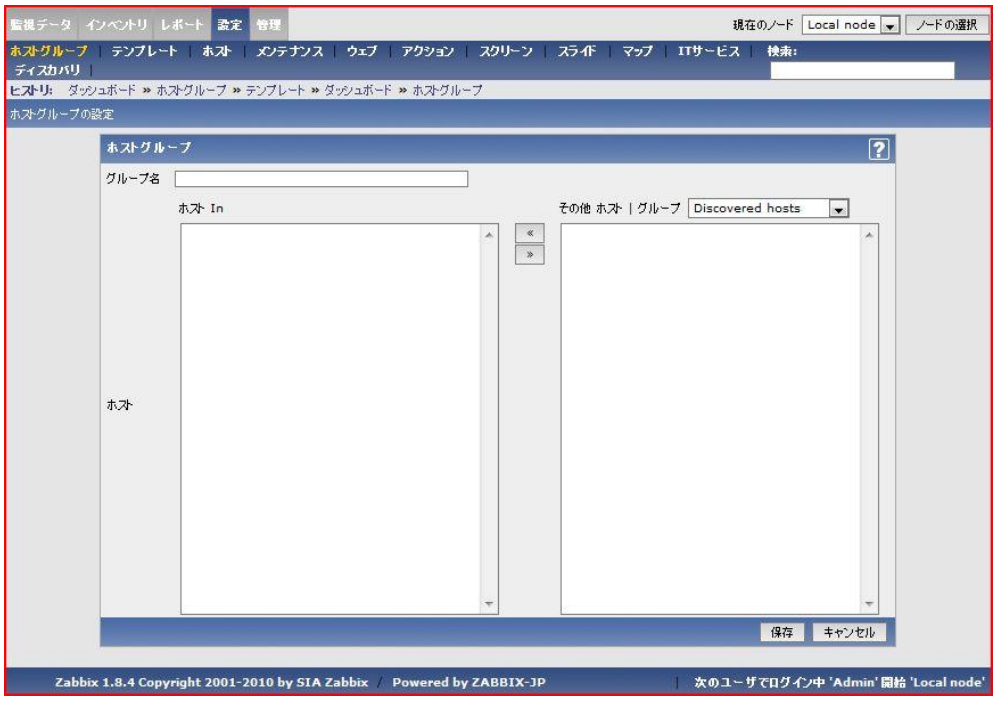


図 5-4 ホストグループ作成画面

4. ホストグループ作成

下記情報を入力後、「保存」ボタン(Zabbix 5.0 LTS では「グループ名」のみを入力して「追加」ボタン)を押下する。

表 5-2 ホストグループ設定

入力項目	設定値
グループ名	Gfarm Filesystem
ホスト	なし

「保存」ボタン押下後、成功メッセージが表示され、一覧に追加されていることを確認する。

以上で、テンプレートのインポートおよび、ホストグループの設定が完了となる。次節以降は、監視対象となるサーバ、ノードの追加を行う。

5.1.3. ホストの追加

監視対象ノードを追加するには、Zabbix にそのノードを「ホスト」としてそれぞれ追加することになる。追加手順は以下の通りである。なお、分散監視構成においては、監視サーバ自身もホストとして追加する。

1. ログイン

(冗長構成の場合は、子ノードのほうの) 監視サーバの Web インターフェースにアクセスし、Admin ユーザでログインする。

2. ホスト一覧画面の表示

メニューの「設定」－「ホスト」からホスト一覧画面を表示する。

図 5-5 ホスト一覧画面

3. ホスト作成画面の表示

「ホストの作成」ボタンを押下し、ホスト作成画面を表示する。

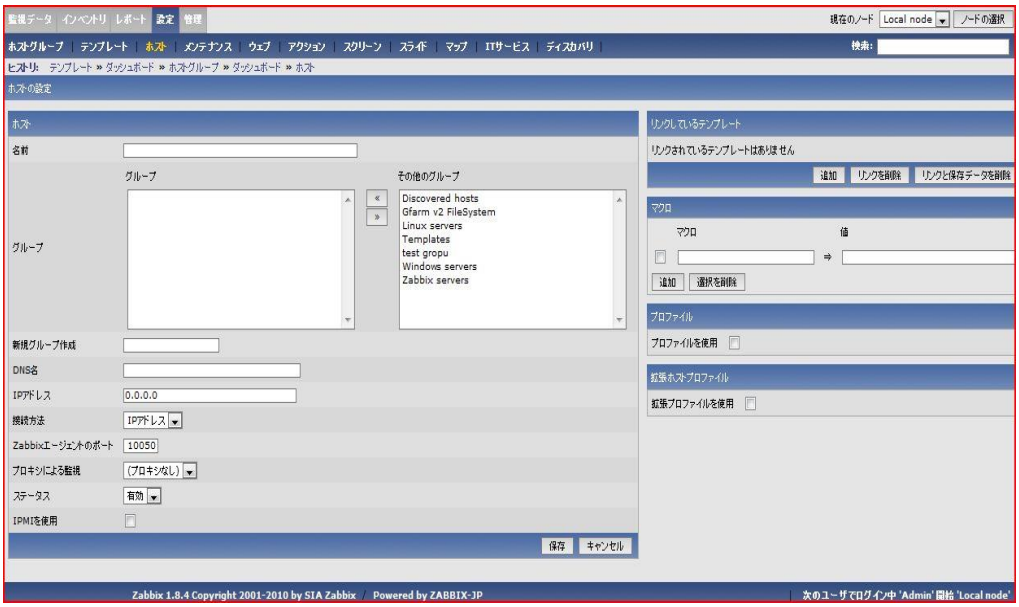


図 5-6 ホスト作成画面

4. ホストの作成

下記情報を入力後、「保存」ボタンを押下し、ホストを作成する。「リンクしているテンプレート」は「追加」ボタンを押下すると、テンプレートの一覧が表示されるので、一覧から選択する。

表 5-3 ホスト設定

設定項目	設定値
名前 (Zabbix 2.0 以降ではホスト名)	監視対象ノードの名前。 ※ホスト上の zabbix_agentd.conf ファイルの Hostname で設定した名前と必ず一致させること。
表示名 (Zabbix 2.0 以降に存在)	Web 画面で表示される監視対象ノードの名前
グループ	Gfarm Filesystem ※「5.1.2 ホストグループの設定」で追加したグループを選択する。
新規グループ作成	空欄
DNS 名	空欄
IP アドレス	監視対象ノードの IP アドレス
接続方法	IP アドレス
Zabbix エージェントのポート	10050
プロキシによる監視	(プロキシなし)
ステータス	有効
IPMI を使用	チェックなし
リンクするテンプレート	監視対象ノードの種別によって異なる。詳しくは後述。
マクロ	監視対象ノードの種別によって異なる。詳しくは後述。
プロファイル	チェックなし ※チェックすると、項目が表示されるので必要に応じて入力
拡張ホストプロファイル	チェックなし ※チェックすると、項目が表示されるので必要に応じて入力

「リンクするテンプレート」欄は、監視対象ノードに応じて、次のように選択する。ここで、監視対象ノードの CPU やファイルシステムといったシステムの資源に関する監視には、次の 2 通りの方法がある。

A) Zabbix 付属の Template OS Linux テンプレートを利用(Zabbix のバージョンによっては Template Linux あるいは Template OS Linux by Zabbix agent という名称の場合がある)

B)　gfarm_zabbix 提供の Template_*_linux_alt テンプレートを利用

このうち B) Template_*_linux_alt テンプレートは、なんからの理由で Template OS Linux テンプレートを利用しない場合のために残してあるもので、特に理由がなければ A) Zabbix 付属の Template OS Linux の利用を推奨する。

表 5-4 A) Template OS Linux 利用時の「リンクするテンプレート」一覧

監視対象ノード種別	リンクするテンプレート
メタデータサーバ	Template_Gfarm_gfmd Template_Gfarm_linux Template OS Linux
ファイルシステムノード	Template_Gfarm_gfsd Template_Gfarm_linux Template OS Linux
代表クライアント	Template_Gfarm_represent_client Template_Gfarm_linux Template OS Linux
一般クライアント	Template_Gfarm_generic_client Template_Gfarm_linux Template OS Linux

表 5-5 B) Template_*_linux_alt 利用時の「リンクするテンプレート」一覧

監視対象ノード種別	リンクするテンプレート
メタデータサーバ	Template_Gfarm_gfmd Template_Gfarm_linux Template_Gfarm_gfmd_linux_alt Template_Gfarm_linux_alt
ファイルシステムノード	Template_Gfarm_gfsd Template_Gfarm_linux Template_Gfarm_gfsd_linux_alt Template_Gfarm_linux_alt
代表クライアント	Template_Gfarm_represent_client Template_Gfarm_linux Template_Gfarm_linux_alt
一般クライアント	Template_Gfarm_generic_client Template_Gfarm_linux

	Template_Gfarm_linux_alt

監視対象ノード 1 台が複数のノード種別を兼任している場合、上記のテンプレートをそれぞれリンクする。
1 台のノード上で複数台の gfsd を動作させている場合は、Template_Gfarm_gfsd をリンクし、後述するホストマクロ {$GFSD_HOSTNAMES} に各ファイルシステムノードのホスト名を記す。

さらに選択したテンプレートに応じて「マクロ」欄を設定する。次表で、マクロのデフォルト値と異なる値をセットする場合は、その「マクロ」欄に設定すること。デフォルト通りで良い項目については、「マクロ」欄を設定する必要はない。「マクロ」欄を設定した場合には、対応する「値」の欄を空のまま放置せず、適切に設定すること。

表 5-6 ホストマクロ設定

共通:	
マクロ	説明
{$NODATA_TIMEOUT}	/var/log/messages のログ監視の「障害イベントを連続して生成(Multiple PROBLEM events): NO」設定の監視対象メッセージについて、そのメッセージがどれくらいの期間発生しなくなったらトリガーを中止するかを設定する。単位は秒。 デフォルト値は 10800 (3 時間)。
{$MULTIPLE_EVENTS_TIMEOUT}	/var/log/messages のログ監視の「障害イベントを連続して生成(Multiple PROBLEM events): YES」設定の監視対象メッセージについて、そのメッセージがどれくらいの期間発生しなくなったらトリガーを中止するかを設定する。単位は秒。 デフォルト値は 15。
Template_Gfarm_gfmd:	
マクロ	説明
{$GFMD_LOGFILE}	gfmd のログメッセージを記録している syslog ファイルのパス。 デフォルト値は/var/log/messages。
Template_Gfarm_gfsd:	

マクロ	説明
{$GFSD_LOGFILE}	gfsd のログメッセージを記録している syslog ファイルのパス。 デフォルト値は/var/log/messages。
{$GFSD_HOSTNAMES}	ファイルシステムノードのホスト名。gfhost –c でファイルシステムノードを登録した際のホスト名でなければならない。ホスト上で複数の gfsd を動作させている場合は、ホスト名を空白で区切って並べること。本マクロの値を- (ハイフン・マイナス) に設定すると、ファイルシステムノード上で hostname –f を実行した際に得られるホスト名を指定したものと看做される。 デフォルト値は、- (ハイフン・マイナス)。
Template_Gfarm_linux:	
マクロ	説明
{$KERNEL_LOGFILE}	カーネルのログメッセージを記録している syslog ファイルのパス。 デフォルト値は/var/log/messages。
{$MEM_FREE_THRESHOLD}	メモリの空き率が本マクロの値を下回ると、トリガーを上げる。 デフォルト値は 20(%)。
{$TIME_DIFF_THRESHOLD}	Zabbix サーバと監視対象ノードの間の時刻のずれが、本マクロの値よりも大きくなるとトリガーを上げる。数値だけ書くと「秒」の意味となり、数値の後ろに “m” を付けると「分」を表す。 デフォルト値は 30 (=30 秒)。
Template_Gfarm_gfmd_linux_alt:	
マクロ	説明
{$GFMD_PGDATA_DIR}	PostgreSQL のデータ領域ディレクトリへのパス。 デフォルト値は/var/gfarm-pgsql。
{$GFMD_PGDATA_DIR_FREE_THRESHOLD}	前項 {$GFMD_PGDATA_DIR} ディレクトリの属するファイルシステムの空き容量率が、本マクロの値を下回るとトリガーを上げる。 デフォルト値は 30(%)。
Template_Gfarm_gfsd_linux_alt:	
マクロ	説明
{$GFSD_SPOOL_DIR}	gfsd のスプールディレクトリへのパス。

	デフォルト値は/var/gfarm-spool。
{$GFSD_SPOOL_DIR_FREE_THRESHOLD}	前項｛$GFMD_PGDATA_DIR｝ディレクトリの属するファイルシステムの空き容量率が、本マクロの値を下回るとトリガーを上げる。 デフォルト値は 30(%)。
Template_Gfarm_linux_alt:	
マクロ	説明
{$MONITOR_DIR}	このディレクトリの属するファイルシステムの空き容量率を監視する。 デフォルト値は/ (ルートディレクトリ)。
{$MONITOR_DIR_FREE_THRESHOLD}	前項｛$MONITOR_DIR｝ディレクトリの属するファイルシステムの空き容量率が、本マクロの値を下回るとトリガーを上げる。 デフォルト値は 30(%)。
{$LOADAVG_THRESHOLD}	直近 1 分の平均 CPU 負荷（÷CPU コア数）が、本マクロの値を上回るとトリガーを上げる。 デフォルト値は 5。

「保存」ボタン押下後、ホスト一覧が表示され、作成したホストが追加されていることを確認する。

6. 分散監視構成設定

分散監視構成の場合は、前章に続いて本章の設定も合わせて行う必要がある。なお説明上は子ノードが 1 つだけであることを想定しているが、同じ要領で設定を行うことで、2 つ目以降の設定も設定可能である。

6.1. 分散監視設定の準備

Zabbix での分散監視では、各 Zabbix サーバに対し識別子であるノード ID を割り振る必要がある。

1. ノード ID の決定

   各 Zabbix サーバに対しノード ID を割り振る。割り振るノード ID は任意の値で問題ないが、重複しないように注意すること。分散監視時のノード ID として指定可能な範囲は、1～999 である。以降の手順では、以下のノード ID を割り振ったものとして説明する。

   - マスターノード：1
   - 子ノード　　：2

2. ノード ID の設定

   手順 1 で設定したノード ID を、各監視サーバ上の設定ファイル /etc/zabbix/zabbix_server.conf に設定する。以下の赤字箇所を追加する（下記例は、マスターノードの場合)。

```
### Option: NodeID
#           Unique NodeID in distributed setup.
#           0 - standalone server
#
# Mandatory: no
# Range: 0-999
# Default:
# NodeID=0
NodeID=1
```

3. データベースの変換

   分散監視を行う際には、データベースのデータを分散監視用に変換する必要があるので、以下を実施する。-n オプションでノード ID を指定する。Zabbix サーバが起動中の場合は、停止してから実施すること。(下記例はマスターノードの場合)

```
# /usr/sbin/zabbix_server -c /etc/zabbix/zabbix_server.conf -n 1
Converting tables ..........................................................done.
Conversion completed.
```

マスターノード側、子ノード側双方で上記手順を実施後、Zabbix サーバの起動を行い、Web インターフェースより分散監視設定を行う。

6.2. マスターノードの分散監視設定

まず、マスターノード側の分散監視設定を行う。

1. Web インターフェースへのログイン
   マスターノードの Web インターフェースにアクセスし、Admin ユーザでログインする。
2. 分散監視管理画面の表示
   メニューの「管理」－「分散監視」から分散監視管理画面を表示する。初期状態では、自分自身が Local Node として登録されている。

図 6-1 分散監視管理画面

3. 子ノードの追加
   右側のプルダウンメニューから「ノード」を選択し、「新規ノード」ボタンを押下で設定画面が表示されるので、下記情報を設定する。

表 6-1 子ノード設定

設定項目	設定値
名前	任意の名称
ID	2 ※子ノードのノード ID を指定
タイプ	子
マスターノード	Local node ※自分自身を指定
タイムゾーン	GMT+09:00
IP アドレス	追加する子ノードの IP アドレス
ポート	10051

ヒストリの保存期間(日)	90
トレンドの保存期間(日)	365

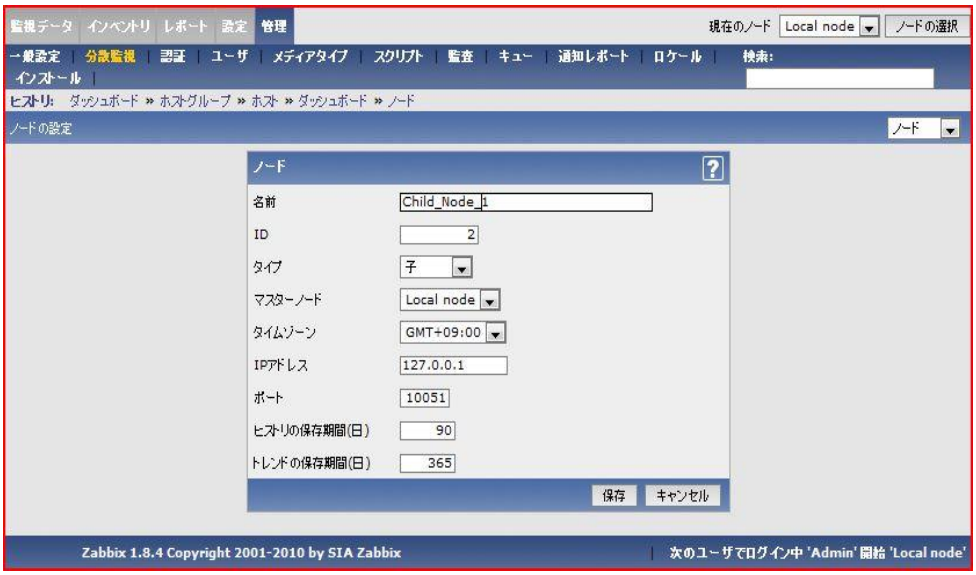


図 6-2 ノード作成画面

ノード情報を入力後、「保存」ボタンを押下すると、下記画面が表示され、Local node/の配下に、子ノードが追加されていることを確認する。

図 6-3 分散監視管理画面(子ノード追加後)

このとき、Local node(マスターノード)の設定でタイムゾーンがデフォルト設定の GMT+00:00 になっているので、GMT+09:00 に変更しておくこと。

以上で、マスターノード側での子ノードの追加は完了となる。別の子ノードを追加する際には、同様の手順を実施する。

6.3. 子ノードの分散監視設定

次に、子ノード側の分散監視設定を行う。

1. Web インターフェースへのログイン
   子ノードの Web インターフェースにアクセスし、Admin ユーザでログインする。
2. 分散監視管理画面の表示
   メニューの「管理」－「分散監視」から分散監視管理画面を表示する。初期状態では、自分自身が Local Node として登録されている。

図 6-4 分散監視管理画面

3. マスターノードの追加
   右側のプルダウンメニューから「ノード」を選択し、「新規ノード」ボタンを押下で設定画面が表示されるので、下記情報を設定する。

表 6-2 マスターノード設定

設定項目	設定値
名前	任意の名称
ID	1 ※マスターノードのノード ID を指定
タイプ	マスター
タイムゾーン	GMT+09:00
IP アドレス	追加するマスターノードの IP アドレス
ポート	10051
ヒストリの保存期間(日)	90
トレンドの保存期間(日)	365

図 6-5 ノード作成画面

ノード情報を入力後、「保存」ボタンを押下すると、下記画面が表示され、自分自身が、追加したマスターノードの配下になっていることを確認する。

図 6-6 分散監視管理画面(マスターノード追加後)

このとき、Local node(子ノード)の設定でタイムゾーンがデフォルト設定の GMT+00:00 になっているので、GMT+09:00 に変更しておくこと。

以上で、子ノード側でのマスターノードの追加は完了となる。

6.4. 相互監視構成設定

相互監視によって、マスターノードで子ノードの監視、子ノードでマスターノードを監視する。相互監視の設定手順として以下を実施する。

1. Zabbiz サーバのホスト追加

   Zabbix サーバの監視設定手順は、「5.1.3 ホストの追加」と同様の手順を踏む。ホスト設定情報の内で、異なる部分を以下に示す。

表 6-3 Zabbix サーバ ホスト設定

設定項目	設定値
グループ	Zabbix Servers
DNS 名	Zabbix サーバの DNS 名
IP アドレス	Zabbix サーバの IP アドレス
リンクするテンプレート	Template App Zabbix Server (Zabbix のバージョンによっては、Template_Zabbix_Server)

相互監視を行うため、上記設定手順は、マスターノード、子ノードそれぞれで実施する。マスターノード側では、子ノードの情報。子ノード側では、マスターノードの情報を設定する必要があるので注意すること。

7. フェイルオーバ実行機能の設定

Gfarm でメタデータサーバが冗長化されている場合、マスターサーバの障害時に、フェイルオーバさせてスレーブサーバをマスターサーバに昇格させることが可能である。本機能では、Zabbix がマスターサーバの致命的な障害を検出した場合、自動的にマスターサーバを停止し、昇格可能なスレーブメタデータサーバをマスターに昇格させる自動フェイルオーバ実行機能を実現する。本機能は、初期状態では無効になっている。

7.1. フェイルオーバ実行機能の動作

フェイルオーバ実行機能は、Zabbix が障害を検知した際にフェイルオーバを行うスクリプトを実行することで実現している。Zabbix から起動されたフェイルオーバスクリプトは、さらに各メタデータサーバの情報収集を行うスクリプトを実行する。この情報収集を行うスクリプトは各サーバホスト上にインストールされており、フェイルオーバスクリプトは SSH を利用してホスト毎に情報収集を実行する。

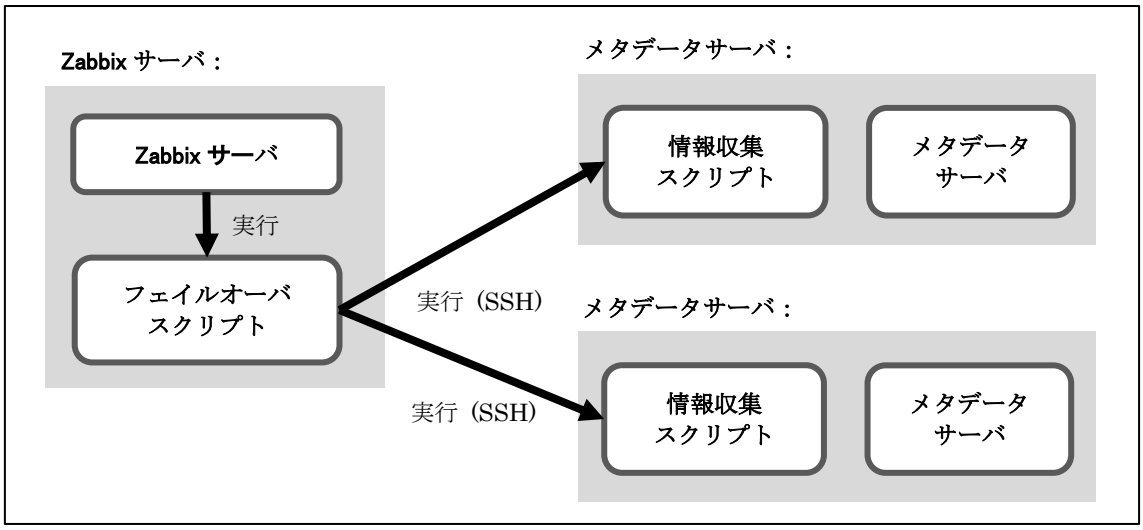


図 7-1 フェイルオーバスクリプトの動作

情報収集の結果、以下の条件をすべて満たしていれば、フェイルオーバスクリプトはフェイルオーバを試みる。条件を満たしていなければフェイルオーバを断念して、フェイルオーバスクリプトは実行を終了する。

- すべてのメタデータサーバホストから、正常に情報収集ができた。
- 動作しているスレーブメタデータサーバが保持するメタデータは最新である。
- マスターメタデータサーバ（クライアント向けにポートを listen しているメタデータサーバ）が動作していない。

ただし「failover_type=availability」と設定されている場合は、情報収集ができないメタ

データサーバホストが存在した場合でもフェイルオーバを実行する。さらに「failover_type=availability」かつ「allow_inconsistency=true」と設定されている場合は動作しているスレーブメタデータサーバが保持するメタデータが古いものであることが判明している場合もフェイルオーバを実行する。このように情報収集ができないメタデータサーバーホストが存在したりメタデータが古い場合には、メタデータの更新を禁止してフェイルオーバを行うので、古いメタデータを参照することによるアクセスエラーが発生する可能性があるものの、split brain によるメタデータ更新の衝突が発生することはない。このメタデータ更新禁止機能を利用するには gfarm-2.7.17 以降を必要とする。

フェイルオーバを実行するに当たって、マスターへ昇格させるスレーブメタデータは、以下の要領で選択する。

1. 情報収集の結果、現在動作中でないサーバは候補から除外する。
2. 情報収集の結果、ジャーナルファイルの最大シーケンス番号が他のサーバより小さいサーバは、候補から除外する。このとき、停止中のサーバについても、その番号は有効とする。言い換えれば、停止中のサーバだけが最新のシーケンス番号を持っていると、昇格対象のサーバが選定されない状態となる。ただし「failover_type=availability」かつ「allow_inconsistency=true」と設定されている場合にはそのような場合でも選定される。
3. 候補に残っているサーバの中で、フェイルオーバスクリプトの設定ファイル（詳しくは後述）で最も先頭近くに記述されているものを昇格対象とする。

昇格対象サーバを選定できた場合、フェイルオーバスクリプトはそのサーバに対して昇格を一度だけ試みる。昇格対象サーバを選定できなかったり、昇格を試みたものの成功しなかったりした場合も、リトライや昇格対象の選定し直しは行わない。また、フェイルオーバスクリプトは Zabbix のトリガーが上がった際のアクションとして起動されるので、その後トリガーが上がったままとなってしまうと、再実行される機会がないので注意が必要である。トリガーがいったん取り下げられれば、再びトリガーが上がったときには、フェイルオーバスクリプトが実行される。

「failover_type=availability」と設定されている場合で、情報収集ができないメタデータサーバが存在するかフェイルオーバ対象サーバのメタデータが古いことが判明している状況下でフェイルオーバを実行する場合、フェイルオーバスクリプトは上記に加えて以下も実行する。

1. 設定 hook_to_prepare_for_failover にあるコマンドを、コマンド引数に、新マスターの

ホスト名と、停止しているメタデータサーバのホスト名のリストを与えて監視サーバ上で実行する。旧マスターメタデータサーバのホスト名は、この停止しているホスト名のリスト中のうちの一つであるはずである。

このコマンドが失敗(すなわち 0 以外の終了コード) を返した場合には、フェイルオーバを中断する。

このコマンドでは、split brain 状態が発生するのを回避するための処理を行うことを想定している。たとえば障害の発生したメタデータサーバを IPMI 経由で停止するとか、あるいは障害の発生したメタデータサーバの接続されている Ethernet Switch のポートを down させるといった処理を記述する。

なお gfarm-2.7.17 以降を利用する場合、下記の read_only_failover_config に関する記述にあるように、このような状況で昇格したメタデータサーバではデフォルトでメタデータ更新が禁止されるので、hook_to_prepare_for_failover の設定がデフォルトのままでも split brain によってメタデータ更新の矛盾が発生する危険はない。

2. 昇格対象の gfmd の現時点のジャーナルファイルを、「元のファイル名.日付_時刻」という名称のファイルにバックアップする。バックアップに失敗した場合はフェイルオーバを中断する。

   これは万一の場合に、障害の発生したメタデータサーバのジャーナルファイルと内容を比較できるようにするために実施している。
3. 昇格対象の gfmd がフェイルオーバ時に読むファイル gfmd_failover_agent_conf_file の内容を、設定項目 read_only_failover_config の内容で置き換える。この設定項目のデフォルトは「read_only enable」であり、昇格後の gfmd はデフォルトでメタデータ更新が禁止され split brain を防止する。
4. 昇格対象の gfmd へ USR1 シグナルを送りマスターへと昇格させる。
5. 設定 hook_after_failover にあるコマンドを監視サーバ上で実行する。このコマンドにも hook_to_prepare_for_failover と同様に、新マスターのホスト名および停止しているメタデータサーバのホスト名のリストが引数として与えられる。

   このコマンドは元々、gfmd がアクセス可能になった後、安全のために各 gfsd を read only モードに移行するといった措置を可能にするために用意していたが、read_only_failover_config 機能が実装されたため、現在では特に設定の必要はない。このコマンド開始時には gfmd はまだ昇格処理中でアクセスできないので注意すること。

設定項目 read_only_failover_config が適用されメタデータ更新が禁止された状態でメタデータサーバが立ち上がった場合、以下のいずれかの方法で復旧する。

- 旧マスターを復旧させたのち、今回 read_only で昇格した新マスターを再起動してスレーブに戻す。
- 旧マスターが復旧できない場合は、新マスターでメタデータ更新を有効にする。まず

split brain が発生しないことを確認するため、マスターメタデータサーバが一つしか起動していないこと、および新旧の両マスターで「gfjournal –m ジャーナルファイル名」でジャーナルファイルのシーケンス番号を表示させそれが一致することが確認する。

- シーケンス番号が一致する場合は gfarmadm グループ権限を持つユーザでコマンド「gfstatus –Mm 'read_only disable'」を実行してメタデータの更新を有効にする。
- 一致しない場合には（どちらの方法も gfmd の再起動を伴うため時間はかかるが)、ジャーナルファイルを旧マスターから新マスターへコピーしてから新マスターgfmd を再起動する、あるいは gfdump.postgresql コマンドで旧マスターのバックエンドデータベースをダンプし新マスターへコピー後リストアするのどちらかの手段でメタデータを一致させる。この結果、新マスターはスレーブに戻って起動するので、この gfmd に対してシグナル USR1 を送りマスターへ昇格させる。

split brain が発生する危険がない場合、すなわち「failover_type=consistency」と設定されているか、あるいは「failover_type=availability」でもすべてのメタデータサーバから情報が取得でき、さらに昇格対象のサーバが保持するメタデータが最新のものだと判明している状況では、フェイルオーバスクリプトは昇格対象のサーバ上の gfmd_failover_agent_conf_file を read_only_failover_config ではなく、read_write_failover_config の内容で置き換える。

以下、フェイルオーバスクリプトの設定手順について説明する。前章までの記述にしたがって Zabbix サーバエージェントが導入済みであることを前提とする。

7.2. SSH 公開鍵の生成と配布

下記の手順を、代表クライアント上の zabbix ユーザで実行する。

この手順は代表クライアントから Gfarm メタデータサーバに ssh ログインできることを確認すると共に、ssh 先のメタデータサーバのエントリを known_hosts ファイルに登録するために行う。known_hosts ファイルにエントリが未登録の場合は、パスフレーズなしで SSH ログインができずフェイルオーバ実行が失敗するので、注意が必要である。また、known_hosts に登録するホスト名は、gfmdhost -l コマンドで表示されるものと一致している必要がある点も、合わせて注意すること。下記の手順は全て zabbix ユーザで実行する。この手順により、代表クライアントから各メタデータサーバに SSH でログインできることを可能にする。

1. zabbix ユーザの認証用の鍵を生成する。

```
$ ssh-keygen -N "" -t rsa
```

2. メタデータサーバ各機に、鍵をコピーする。

```
$ ssh-copy-id zabbix@メタデータサーバのホスト名
```

代表クライアントからメタデータサーバ各機に、パスフレーズ無しで ssh 接続ができることを確認する。

```
$ ssh メタデータサーバのホスト名
```

7.3. zabbix ユーザの sudo 権限の設定

メタデータサーバ各機で、下記の手順を root ユーザで実行する。

1. zabbix ユーザが、任意のコマンドを管理者権限で実行できるよう設定する。
   visudo コマンドを使用し、/etc/sudoers ファイルを編集する。

```
# visudo
```

   zabbix ユーザに関する設定行を以下のように修正する。(赤字が修正部分)

```
zabbix    ALL=(ALL)          NOPASSWD: ALL
```

7.4. フェイルオーバスクリプトの設定ファイルの編集

下記の手順を、代表クライアントの zabbix ユーザで実行する。

1. フェイルオーバスクリプトファイルの設定ファイルを編集する。
   エディタで、設定ファイル$ZABBIX_CONFDIR/externalscripts/gfarm_gfmd_failover.conf を開いて編集する。ただしここで$ZABBIX_CONFDIR は、gfarm_zabbix をインストールした際に install.conf ファイルの設定項目 ZABBIX_CONFDIR として指定した値（初期値は/etc/zabbix）である。したがって初期値のままなら、設定ファイルのパスは/etc/zabbix/externalscripts/gfarm_gfmd_failover.conf となる。

   設定ファイルは、INI ファイル形式である。以下の例のように、メタデータサーバ 1 台毎に [gfmd1]、[gfmd2]、....とセクションに分けて記述する。

```
ssh=ssh -i /etc/zabbix/.ssh/id_rsa
failover_type=availability

[gfmd1]                          ← Gfarm メタデータサーバ 1 台目の設定
host=mds-master                        ← ホスト名
gfmd_listen_port=10601                 ←listen している TCP ポート番号
gfmd_pid_file=/var/run/gfmd.pid        ←PID ファイルのパス
gfmd_journal_file=/var/gfarm-metadata/journal/0000000000.gmj ←ジャーナルのパス
gfmd_failover_agent_conf_file=/etc/gfmd.failover.agent.conf ←フェイルオーバースクリプトが上書きする gfmd 設定ファイルのパス
gfarm_bindir=/usr/bin                  ←Gfarm コマンドのディレクトリへのパス
```

```
[gfmd2]                          ← Gfarm メタデータサーバ 2 台目の設定（以下同様）
host=mds-slave
gfmd_listen_port=10601
gfmd_pid_file=/var/run/gfmd.pid
gfmd_journal_file=/var/gfarm-metadata/journal/0000000000.gmj
gfmd_failover_agent_conf_file=/etc/gfmd.failover.agent.conf
gfarm_bindir=/usr/bin
```

前述のように、フェイルオーバスクリプトが昇格対象サーバを選定する際、複数のサーバが候補として残った場合は、本設定ファイルの先頭に最も近いサーバが選ばれるため、記述順も意識すること。
記述可能な設定項目は、下記の表の通り。

表 7-1 フェイルオーバスクリプト設定項目一覧

設定項目	説明
セクション共通部	
log_to_syslog	syslog にメッセージを書き込むかどうかのフラグ。true ないし yes を指定すると、syslog への書き込みが行われる。 デフォルト値は true。
syslog_facility	syslog にメッセージを書き込む際に使用するファシリティ。 デフォルト値は user。
lock_file	フェイルオーバスクリプトの二重起動を防止するために使用する、ロックファイルのパス。 デフォルト値は/var/tmp/gfarm_gfmd_failover.lock。
inspection_timeout	情報収集を開始してから完了するまでの最大待ち時間で、単位は秒。これを越えると、時間切れになった旨のメッセージが出力され、フェイルオーバスクリプト自体がエラーとなって終了する。never を指定すると、無期限になる。 デフォルト値は 200。
promotion_timeout	フェイルオーバを開始してから完了 (クライアント向けにポートを listen しているのを確認できた状態) までの最大待ち時間で、単位は秒。これを越えると、

	時間切れになった旨のメッセージ が出力され、フェイルオーバスクリプト自体がエラーとなって終了する。never を指定すると、無期限になる。 デフォルト値は never。
failover_type	フェイルオーバの種類。availability と設定されている場合は、情報収集ができないメタデータサーバホストが存在した場合でもフェイルオーバを実行する。 デフォルトは consistency。
allow_inconsistency	failover_type=availability と設定されていて、動作しているスレーブサーバの保持するメタデータが古いことが判明している場合にもフェイルオーバを許すかを設定する。 その場合 gfmd_failover_agent_conf_file ファイルへ read_only_failover_config の設定値が書き込まれる。 デフォルトは true。
hook_to_prepare_for_failover	failover_type=availability と設定されており、かつ情報収集ができないメタデータサーバホストが存在する状況下でフェイルオーバを実施しようとする場合、フェイルオーバ準備として監視サーバ上で実行するコマンド。 このコマンドが失敗を返した場合はフェイルオーバを中断する。 引数として、新マスターのホスト名と、停止しているメタデータサーバのホスト名のリストが渡される。 デフォルトでは true コマンドが設定されている。
hook_after_failover	failover_type=availability と設定されており、かつ情報収集ができないメタデータサーバホストが存在する状況下でフェイルオーバ用シグナルを昇格対象 gfmd に送った後に監視サーバ上で実行するコマンド。 引数として、新マスターのホスト名と、停止しているメタデータサーバのホスト名のリストが渡される。 デフォルトは空文字列。
gfmd_failover_agent_conf_file	フェイルオーバ時に split brain 発生の可能性があるか否かの状況に応じ read_only_failover_config ないし read_write_failover_config の内容を書き込む昇格

	対象のサーバ上のファイル名。 デフォルトは/etc/gfmd.failover.agent.conf。
read_only_failover_config	split brain 発生の可能性がある状況、すなわち failover_type=availability と設定されており、かつ情報収集ができないメタデータサーバホストが存在する状況下でフェイルオーバを実施する場合に gfmd_failover_agent_conf_file ファイルに書き込まれる 1 行分の設定。 デフォルトでは「read_only enable」であり、この場合、メタデータ更新は禁止されるため split brain による矛盾は発生しない。 1 行ではなく複数行の設定を適用したい場合、この項目に「include gfmd.failover.read_only.conf」と設定したうえ、gfmd.failover.read_only.conf 側に「read_only enable」を含む複数行の記述をすればよい。
read_write_failover_config	split brain の可能性がない状況でフェイルオーバを実施する場合に gfmd_failover_agent_conf_file ファイルに書き込まれる 1 行分の設定。 デフォルトは空文字列。
各セクション	
host	gfmd の動作するホスト。 デフォルト値はセクション名。
zabbix_extscriptdir	host 上で、情報収集スクリプト（gfarm_gfmd_failover_agent.pl）やフェイルオーバスクリプト、およびその設定ファイル等がインストールされているディレクトリへのパス。 デフォルトは、(host 上ではなく）フェイルオーバスクリプトが実行されたホスト上での、当該ディレクトリのパス。
gfarm_bindir	host 上にインストールされた Gfarm の bindir（一般コマンドの置かれたディレクトリへのパス)。 デフォルト値は/usr/local/bin。
gfmd_journal_file	host 上の gfmd が読み書きするジャーナルファイルへのパス。 デフォルト値は /var/gfarm-

	metadata/journal/0000000000.gmj。
gfmd_pid_file	host 上の gfmd が作成する PID ファイルへのパス。 デフォルト値は/var/run/gfmd.pid。
gfmd_listen_address	host 上の gfmd が listen しているアドレス。 デフォルト値は 0.0.0.0。
gfmd_listen_port	host 上の gfmd が listen しているポートの番号。 デフォルト値は 601。
ssh	host に対して ssh で接続する際の ssh コマンド名およびオプション。フェイルオーバスクリプトは Zabbix から自動実行されるため、パスフレーズの入力無しで接続できるようになっている必要がある。 デフォルト値は ssh。
sudo	host に ssh で接続したとき、スーパユーザ権限でコマンドを実行する際に使用する sudo コマンドのコマンド名およびオプション。フェイルオーバスクリプトは Zabbix から自動実行されるため、パスワード入力無しで sudo が実行できるようになっている必要がある。 デフォルト値は sudo。

2. zabbix ユーザでフェイルオーバスクリプトをテスト実行する。
   設定ファイルの記述が終わったら、確認のためフェイルオーバスクリプトのテスト実行を行う。下記の手順は zabbix ユーザで実行する。全ての Gfarm メタデータサーバからの情報が表示されれば、正しく設定できていることを意味する。
   正しく動作している場合、LISTEN 欄はマスターgfmd のみ「yes」で他の gfmd は「-」となり、MAX_SEQNO 欄は全 gfmd で数字が表示される。

```
$ /etc/zabbix/externalscripts/gfarm_gfmd_failover.pl –t
RUN   LISTEN   MAX_SEQNO                HOST
yes   yes                 339   gfmd1 (mds-master:10601)
yes   -                   339   gfmd2 (mds-slave:10601)

master gfmd is running
```

7.5. Zabbix エージェントの追加設定

下記の手順を、代表クライアントとして割り当てたホストの root ユーザで実行する。

Zabbix エージェントの設定ファイル/etc/zabbix/zabbix_agentd.conf を編集し、下記の項目を追加設定する。

```
EnableRemoteCommands=1
LogRemoteCommands=1
```

Zabbix エージェントを再起動する。

[CentOS 6]:

```
# service zabbix-agent restart
```

[CentOS 7]:

```
# systemctl restart zabbix-agent.service
```

7.6. Web インターフェース上での設定

Zabbix 上でのフェイルオーバの設定は下記になる。下記の手順は全て Zabbix の Web インターフェース（分散監視構成の場合は、子ノードのほう）上で行う。

1. Web インターフェースへのログイン
   子ノードの Web インターフェースにアクセスし、Admin ユーザでログインする。
2. メニューの「設定」－「アクション」からアクション一覧画面を表示する。
3. Zabbix 2.0 以降の場合はイベントソースとして 「トリガー」 を選ぶ
4. 「アクションの作成」ボタンを押下する。
5. 次表の設定を行い、「保存」を押下する。

表 7-2 アクション設定

アクション	
設定項目	設定値
名前	フェイルオーバ実行
イベントソース (Zabbix 2.0 以降は存在せず)	トリガー
エスカレーションを有効 (Zabbix 2.0 以降は存在せず)	チェックなし
デフォルトの件名	変更なし(デフォルト値のまま)
デフォルトのメッセージ	変更なし(デフォルト値のまま)

リカバリメッセージ	チェックなし
ステータス	有効
アクションのコンディション	
設定項目	設定値
計算のタイプ	(A) and (B) and (C)
コンディション	(A)トリガーの値 = “障害” (B)メンテナンスの状態 　期間外 “メンテナンス” (C)トリガー = 　“Template_Gfarm_represent_client:Problem of gfmd ({ITEM.LASTVALUE})” ※(A)については、Zabbix 5.0 LTS 以降の場合、タイプとして「トリガーの深刻度」、オペレータとして「以上」、深刻度として「重度の障害」を選べばよい ※(C)については、Zabbix 5.0 LTS 以降の場合、[新規条件]の項で、[トリガー]、[等しい]を選び、トリガーの検索文字列として「Problem of gfmd」と入力すると候補として表れるので選択する。 それ以前のバージョンでは[新規条件]の項で、[トリガー]、[=]を選び[選択]ボタンを押下すると、選択ウィンドウが現れるので[Problem of gfmd (*不明*)]を選び、選択ボタンを押下する。
アクションのオペレーション(Zabbix 5.0 以降では「実行内容」)	
設定項目	設定値
オペレーションのタイプ	リモートコマンド
ターゲットリスト	現在のホスト
タイプ	カスタムスクリプト
次で実行	Zabbix エージェント
リモートコマンド	/etc/zabbix/externalscripts/gfarm_gfmd_failover.pl

以上で、フェイルオーバ実行機能の設定は完了である。

8. その他の注意点

Zabbix の導入・設定に関して、当該する章で書ききれなかった点をここで補足する。

8.1. SELinux 環境での問題

SELinux を有効にしている環境では､Zabbix エージェントが/etc/zabbix/externalscripts/の下にある外部スクリプトの実行に失敗したり、/var/log/messages 等のログファイルの読み込みに失敗したりすることがある。

/var/log/zabbix/zabbix_agentd.log に以下のようなメッセージが出力されていれば、この問題が起きている可能性が高い。

```
sh: /etc/zabbix/externalscripts/gfarm_gfmd_gfhost.sh: Permission denied
27376:20140826:113633.587 cannot open [/var/log/messages]: [13] Permission denied
```

問題を回避するには、SELinux の Zabbix 用セキュリティポリシーの定義を修正して、Zabbix エージェントによるこれらの処理が許可されるようにする。

あらかじめ policycoreutils-python パッケージをインストールしておく。

```
# yum -y install policycoreutils-python
```

このパッケージに付属する audit2allow コマンドの入力として /var/log/audit/audit.log を与えると、エラーを回避するのに必要となる追加定義が表示される。

```
# audit2allow </var/log/audit/audit.log
allow　zabbix_agent_t　タイプ 1:クラス 1　処理 1
allow　zabbix_agent_t　タイプ 2:クラス 2　{ 処理 2　処理 3}
```

これを gfarm-zabbix の配布に含まれる zabbix-agent-gfarm-centos7.te に以下のような形式で追加し

```
require {
    type タイプ 1;
    type タイプ 2;
    class クラス 1　処理 1;
    クラス 2　{ 処理 2　処理 3};
}
allow　zabbix_agent_t　タイプ 1:クラス 1　処理 1
allow　zabbix_agent_t　タイプ 2:クラス 2　{ 処理 2　処理 3}
```

「3.4 Zabbix エージェントのインストール」の手順に従ってバイナリ形式である zabbix-agent-gfarm-centos7.pp に変換後、設定する。

Zabbix 1.8 のように古いバージョンであり、Linux のディストリビューションも古いケ

ースでは、Zabbix 用のセキュリティポリシー定義を無効にすると解決する場合もある。無効にするには、Zabbix エージェントの動作しているホスト上で、root 権限で以下のコマンドを実行する。

```
# semodule -r zabbix
```

Zabbix エージェントが動作中であれば、無効にした後でいったん起動し直すこと。

[CentOS 6]:

```
# service zabbix-agent restart
```

[CentOS 7]:

```
# systemctl restart zabbix-agent.service
```

8.2. メール通知設定

ログファイル関係の障害通知は、一定の時間が経過すると Zabbix Web ユーザーインターフェースのダッシュボードから消去される。障害を見落とさないために「冗長化構成 Gfarm 監視機能 管理・利用マニュアル」の「6. メール通知設定」にある設定を行い、メールによる障害通知を有効にしておくことを強く推奨する。

9. gfarm_zabibx 旧バージョンからのアップグレード

現在 gfarm_zabbix パッケージを使用しているシステムで、新しいバージョンにアップグレードする手順について記す。ただし、このアップグレード手順では、これまでの監視データ（アイテムやトリガー）の履歴は引き継げず、消去されるので注意すること。

1. システム構成の把握

   gfarm_zabbix バージョン 3.0 までと 4.0 以降ではテンプレートの構成が異なるので、「2.2 Gfarm 構成」をまず参照し、どの監視ノードがどの役割（メタデータサーバ、ファイルシステムノード、代表クライアント、一般クライアント）に当たるのかを把握し、どのホストを代表クライアントにするかを決めてから、アップグレードを実行すること。

2. 旧バージョンの監視用テンプレートの削除

   Zabbix の Web インターフェースに Admin ユーザでログインし、「設定」→「テンプレート」メニューを選択する。

“Template_Gfarm_” で始まるテンプレートすべて（gfarm_zabbix バージョン 1 は 10 個、バージョン 2 は 11 個、バージョン 3.0 は 7 個）にチェックを入れる。

ウィンドウ下部にあるプルダウンメニューから「選択とリンクした要素も一緒に削除しますか？」を選び、実行を押下する。(プルダウンメニューの「選択を削除」では、テンプレートが存在しない状態で監視データだけ残ってしまうので、そちらは選択しないこと。）この操作で、履歴データも消去されるので注意すること。。

3. gfarm_zabbix パッケージのインストール

「3.5 gfarm_zabbix パッケージのインストール」にしたがって、監視対象ノード各機に gfarm_zabbix パッケージをインストールする。また、インストール後に、監視対象ノード各機にインストールされている、gfarm_zabbix-3.0 以前のバージョンからのアップデートの場合は、以下の設定ファイルを削除する。

```
$ZABBIX_CONFDIR/zabbix_agent.d/userparameter_redundant_gfarm.conf
$ZABBIX_CONFDIR/zabbix_agent.d/userparameter_postgresql.conf
```

ここで $ZABBIX_CONFDIR は、gfarm_zabbix パッケージインストール時に install.conf ファイルで指定したパスである（デフォルトは /etc/zabbix)。

なお、gfarm_zabbix-4.0.1 以前のバージョンから gfarm_zabbix-4.1 以降へのアップデートでは、以下の違いがあることに留意すること。

C) visudo コマンドで追加する /etc/sudoers ファイルの設定に、postgres ユーザーへの sudo 権限が追加されている。

D) install.conf ファイルに対する設定項目として、GFMD_CONFIG_PREFIX および POSTGRES_USER が追加されている。

4. zabbix_agentd の再起動

root 権限で以下のコマンドを実行して、監視対象ノード各機上で動作中の

zabbix_agentd を再起動する。

[CentOS 6]:

```
# service zabbix-agent restart
```

[CentOS 7]:

```
# systemctl restart zabbix-agent.service
```

5. gfarm_zabbix の動作確認

「4.4 zabbix_get による動作確認」の記述に沿って、gfarm_zabbix の動作確認を行う。

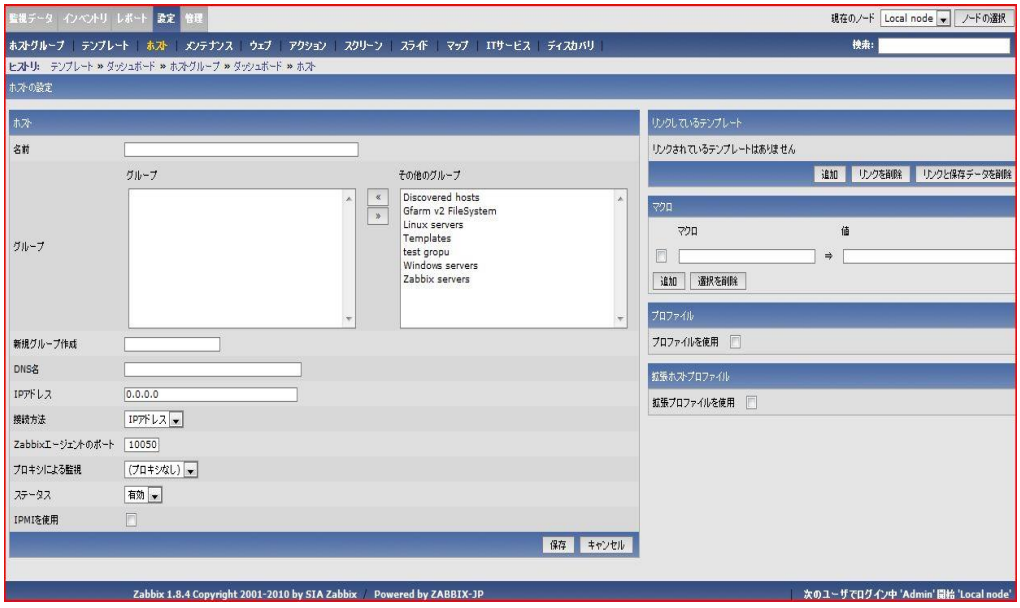
6. Gfarm 監視用テンプレートの再導入

「5.1.1 Gfarm 監視用テンプレートの導入」にしたがって、監視用テンプレートを導入し直す。

7. テンプレートへのリンクの再設定

各監視対象ノードに対して、テンプレートへのリンクを再設定する。Zabbix の Web インターフェースに Admin ユーザでログインし、「設定」→「ホスト」メニューを選択する。

監視対象ノードの「名前」部分をクリックする。そのホストの設定を行う画面が表示される。



「リンクするテンプレート」として、gfarm_zabbix パッケージで提供しているテンプレートを選択する。具体的にどのテンプレートをリンクさせるかについては、「5.1.3 ホ

ストの追加」にある「表 5-4 A) Template OS Linux 利用時の「リンクするテンプレート」一覧」「**エラー! 参照元が見つかりません。**」およびその前後の説明を参照すること。

分散監視を行っている場合は、「7.6 Web インターフェース上での設定」に従って Zabbix サーバの監視用テンプレートをリンクし直す必要がある。なお、旧バージョンの gfarm_zabbix で提供していた Zabbix サーバ監視用テンプレート (Template_Gfarm_zabbix および Template_Gfarm_Zabbix_Server) は gfarm_zabbix バージョン 3 以降では提供されていないので、Zabbix 付属のものを利用すること。

「マクロ」の設定欄についても同様に、「表 5-6 ホストマクロ設定」および前後の説明を読んだ上で、適切にマクロを定義すること。

「リンクするテンプレート」と「マクロ」の設定が両方とも終わったら、最後に「保存」ボタンを押下する。

このリンクの再設定は、監視ノード各機に対して行う。

8. フェイルオーバ実行機能の再設定

gfarm_zabbix-2.2 以前からのアップデートで、フェイルオーバ実行機能を利用している場合に行う。「7.6 Web インターフェース上での設定」で行ったアクションの設定で、コンディションの (C) トリガーとして指定している

Template_Gfarm_represent_client_nodep:Problem of gfmd ({ITEM.LASTVALUE})”

を、以下のようにテンプレート名を変更する。

Template_Gfarm_represent_client:Problem of gfmd ({ITEM.LASTVALUE})”

以上で、アップグレード作業は完了である。