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18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。  ・フェイルオーバスクリプトの挙動を詳述した。 | SRA |
| 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。  abbix\_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  Template\_Gfarm\_gfsd\_linux へ改名した。 | SRA |
| 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。 | SRA |
| 4.3 | 2021/10/18 | ・Zabbix 5.0 LTS に対応した。  ・read\_write\_failover\_configのデフォルト値を「read\_only disable」へと変更し、昇格するメタデータサーバが最新のメタデータを保持している場合はデフォルトでメタデータ更新可能状態で立ち上がるようにした |  |

目次

[1. はじめに 1](#_Toc51061513)

[1.1. gfarm\_zabbixパッケージの構成 1](#_Toc51061514)

[1.2. 動作環境 2](#_Toc51061515)

[2. システム構成 3](#_Toc51061516)

[2.1. Zabbix基本構成 3](#_Toc51061517)

[2.2. Gfarm構成 4](#_Toc51061518)

[2.3. Zabbix分散監視構成 6](#_Toc51061519)

[3. インストール 1](#_Toc51061520)

[3.1. ファイアウォールの設定 1](#_Toc51061521)

[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 1](#_Toc51061522)

[3.3. Zabbixサーバのインストール 2](#_Toc51061523)

[3.4. Zabbixエージェントのインストール 13](#_Toc51061524)

[3.5. gfarm\_zabbixパッケージのインストール 15](#_Toc51061525)

[3.5.1. install.confの編集 15](#_Toc51061526)

[3.5.2. Zabbixエージェント用ファイルのインストール 17](#_Toc51061527)

[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 19](#_Toc51061528)

[4. 各ノードの設定 21](#_Toc51061529)

[4.1. zabbixユーザの登録と共通認証鍵の作成 21](#_Toc51061530)

[4.2. 監視サーバの設定 21](#_Toc51061531)

[4.2.1. Zabbixサーバの設定 21](#_Toc51061532)

[4.2.2. Zabbixエージェントの設定 (分散監視構成の場合) 23](#_Toc51061533)

[4.2.3. クライアント設定ファイル編集機能の設定 24](#_Toc51061534)

[4.3. 監視サーバ以外の設定 24](#_Toc51061535)

[4.3.1. Zabbixエージェントの設定 24](#_Toc51061536)

[4.3.2. gfarm\_zabbixスクリプトの設定 26](#_Toc51061537)

[4.4. zabbix\_getによる動作確認 26](#_Toc51061538)

[5. 監視設定 30](#_Toc51061539)

[5.1. 監視項目の設定 30](#_Toc51061540)

[5.1.1. Gfarm監視用テンプレートの導入 30](#_Toc51061541)

[5.1.2. ホストグループの設定 32](#_Toc51061542)

[5.1.3. ホストの追加 34](#_Toc51061543)

[6. 分散監視構成設定 41](#_Toc51061544)

[6.1. 分散監視設定の準備 41](#_Toc51061545)

[6.2. マスターノードの分散監視設定 42](#_Toc51061546)

[6.3. 子ノードの分散監視設定 44](#_Toc51061547)

[6.4. 相互監視構成設定 46](#_Toc51061548)

[7. フェイルオーバ実行機能の設定 47](#_Toc51061549)

[7.1. フェイルオーバ実行機能の動作 47](#_Toc51061550)

[7.2. SSH公開鍵の生成と配布 50](#_Toc51061551)

[7.3. zabbixユーザのsudo権限の設定 52](#_Toc51061552)

[7.4. フェイルオーバスクリプトの設定ファイルの編集 52](#_Toc51061553)

[7.5. Zabbix エージェントの追加設定 57](#_Toc51061554)

[7.6. Webインターフェース上での設定 57](#_Toc51061555)

[8. その他の注意点 59](#_Toc51061556)

[8.1. SELinux環境での問題 59](#_Toc51061557)

[8.2. メール通知設定 60](#_Toc51061558)

[9. gfarm\_zabibx旧バージョンからのアップグレード 61](#_Toc51061559)

# 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。

## 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。

## 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。

# 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。

## 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。

**監視対象ノード (1台以上)**

**監視サーバ**

設定意

監視データ閲覧意

監視データ収集

DB

取得/保存

監視設定意

監視データ意

図 2‑1 Zabbix基本構成

## 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。

**監視サーバ**

**監視対象ノード**

**(メタデータサーバ)**

DB

監視

設定

監視

データ

**監視対象ノード**

**(メタデータサーバ)**

**監視対象ノード**

**(ファイルシステムノード)**

**監視対象ノード**

**(ファイルシステムノード)**

**監視対象ノード**

**(代表クライアント)**

**監視対象ノード**

**(一般クライアント)**

**監視対象ノード**

**(一般クライアント)**

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。

## 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－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。

**監視サーバ**

**(マスターノード)**

**監視対象ノード**

**(1台以上)**

DB

監視

設定

監視

データ

参照

**監視サーバ**

**(子ノード)**

DB

監視データ

監視

設定

監視データ

監視

設定

監視設定

監視データ

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

# 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。

## 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。

## 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 |

## 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> |

1. 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 |

1. 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 |

1. 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 |

1. 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 |

1. /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 |

1. MySQL / MariaDBサーバを起動する。

[CentOS 6]:

|  |
| --- |
| # service mysqld start |

[CentOS 7]:

|  |
| --- |
| # systemctl start mariadb.service |

1. 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 |

1. 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 |

1. /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 |

1. 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 |

1. 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 |

1. 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 |

1. 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 |

1. 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/ |

1. 「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

1. 「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

1. 「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

1. 「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 |

1. 「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* |

1. 「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。

1. 「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。
2. 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 |

1. 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

1. 「Finish」画面が表示される。「Next」ボタンを押下する。



図 3‑9 Finish画面

1. 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

1. 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

1. 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 |

## 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 |

1. 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 |

1. 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 |

1. 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)
2. 。

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 |

1. 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 |

1. /etc/zabbixの所有者を変更する。

|  |
| --- |
| # chown zabbix:zabbix /etc/zabbix |

1. 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 |
|  |

## 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。

### 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 |

### 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。

### 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 |

# 各ノードの設定

本章では、各ノードの設定を記載する。

## 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: |

1. 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 |

1. 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 |

1. 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 |

## 監視サーバの設定

監視サーバの設定を行う。

### 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 |

1. Zabbixサーバを起動する。

[CentOS 6]:

|  |
| --- |
| # service zabbix-server start |

[CentOS 7]:

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

1. 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 |

1. 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 |

1. 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 |

### 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/ |

1. 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 |

1. 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 |

1. 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 |

### 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 |

## 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。

### 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/ |

1. 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 |

1. 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 |

1. 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 |

### 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" |

## 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。 |

# 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。

## 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。

### 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。

1. 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

1. 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

1. 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。

### 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。

1. 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

1. 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

1. 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。

### 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」としてぞ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。

1. 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

1. 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

1. 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。

1. 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)
2. 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。

# 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。

## 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

1. 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 |

1. 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。

## 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。

1. 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

1. 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。

## 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。

1. 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

1. 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。

## 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。

# 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。

## 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。

**Zabbixサーバ**

**フェイルオーバ**

**スクリプト**

**情報収集**

**スクリプト**

**メタデータ**

**サーバ**

**情報収集**

**スクリプト**

**メタデータ**

**サーバ**

実行

実行 (SSH)

実行 (SSH)

**Zabbixサーバ：**

**メタデータサーバ：**

**メタデータサーバ：**

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。

1. 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。

1. 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。
2. 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。
3. 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作業を行う必要がある。

* 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。

## 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 |

1. 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* |

## 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 |

## 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=consistency  #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「read\_only disable」。 |
| 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。 |

1. 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 |

## 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 |

## 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。

1. 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。
2. 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
3. 「アクションの作成」ボタンを押下する。
4. 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。

# 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。

## 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 |

## 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。

# 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。

1. 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。。



1. 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。

1. 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。
2. 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。
3. 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 |

1. 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。

1. 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。

1. 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。

1. 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。