



**OASE** Operation Autonomy  
Support Engine

**Prometheus連携**

※本書では「Operation Autonomy Support Engine」を「OASE」として記載します。

# 目次

## 1. はじめに

[1.1 Prometheus連携【実習】について](#)

## 2. シナリオ説明

[2.1 本書のシナリオ](#)

## 3. 事前設定

[3.1 事前設定](#)

[3.2 監視アダプタ設定](#)

## 4. 作業実行

[4.1 ディジションテーブルファイルの作成](#)

[4.2 ディジションテーブルファイルのアップロード](#)

[4.3 テストリクエスト](#)

[4.4 プロダクション適用](#)

[4.5 アラート発火](#)

[4.6 アクション実行結果の確認](#)

# 1. はじめに

# 1.1 Prometheus連携【実習】について (1/2)

## 本書について

本書では Exastro OASE と監視ソフトウェア「Prometheus」の連携について解説しています。

OASE はいくつかのソフトウェアと連携が可能ですが、本書では「Prometheus アダプタ」および「メールドライバ」との連携をします。包括的な内容は、OASE の公式マニュアル集である [OASE docs](#) をご参照ください。

## 前提

OASE および以下の設定が完了していること

- 「① OASE インストール」～「③ メール送信確認」については、[OASE クイックスタート](#)の「1.1 クイックスタートについて (2/3)」を参照
- 「④ Prometheusアダプタインストール」については、[環境構築マニュアル -ドライバインストール編-](#) を参照
- 「⑤ Prometheusのインストール」～「⑥ Prometheusの設定」については別途公式ドキュメントなどを参照

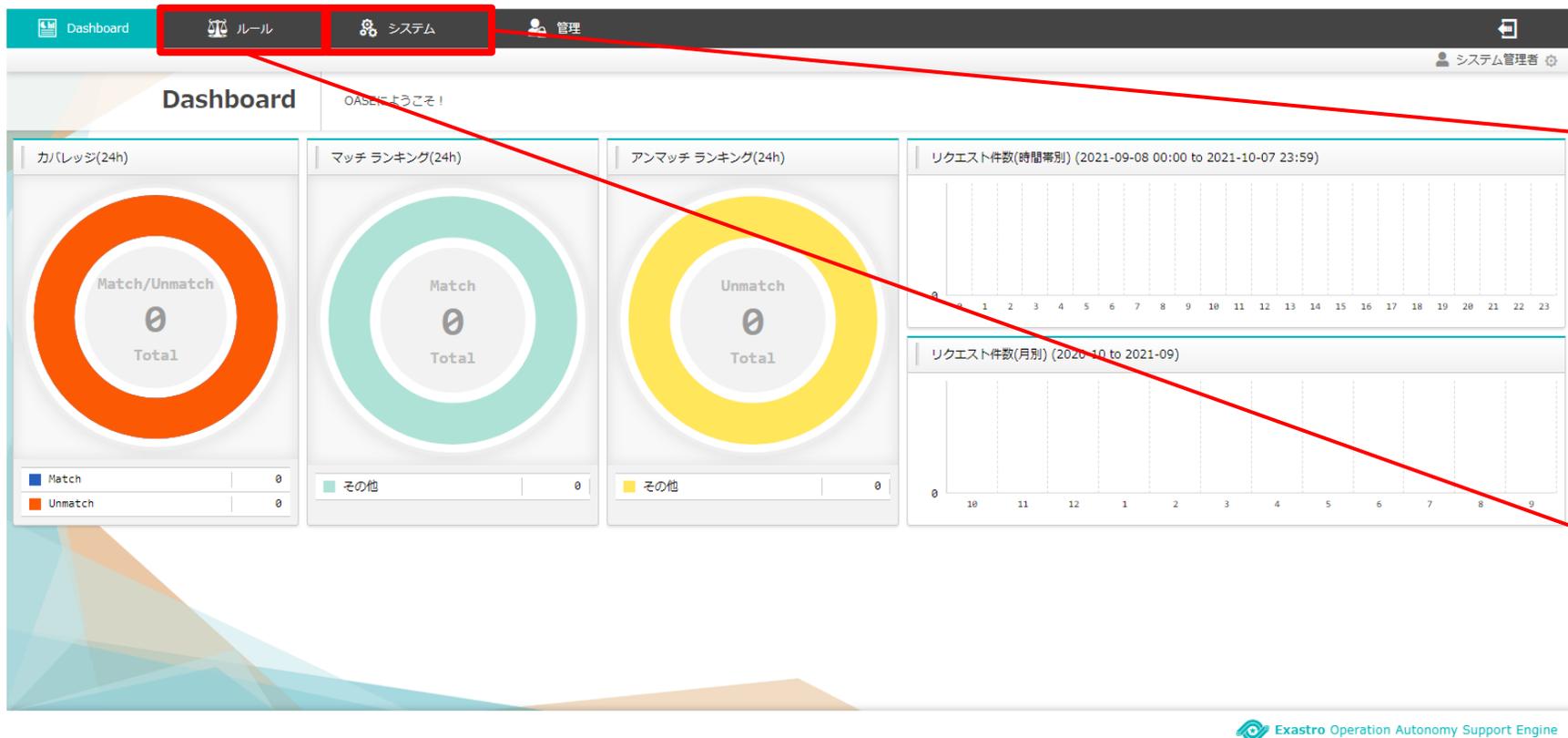


# 1.1 Prometheus連携【実習】について (2/2)

## Prometheus連携メニュー概要

Prometheusとの連携では、下記のメニューを操作します。

それぞれのメニュー機能についての説明は、Exastro OASEの公式マニュアル集である [OASE docs](#) をご参照ください。



カテゴリ：システム

画面名称

アクション設定

監視アダプタ

カテゴリ：ルール

画面名称

ディジションテーブル

トークン払い出し

ルール

リクエスト履歴

アクション履歴

## 2. シナリオ説明

## 2.1 本書のシナリオ (1/3)

### シナリオ

下図は本書で実行するシナリオのイメージです。

監視対象のインスタンスが Down することでアラートが発生し、OASE が連動してアクションを実行します。

※実際には、監視対象のインスタンスにある Node Exporter のコンテナを停止させる。

①インスタンスDown  
(Node Exporter を停止)



監視対象サーバ



②インスタンスDownの  
検知とアラート発報



Prometheusサーバ



③アラートリストを収集し、  
ルールとマッチさせる



OASEサーバ

メールドライバの  
実行



## 2.1 本書のシナリオ (2/3)

### OASEの事前設定～作業実行

#### 【事前設定】

OASEの各種設定

アクション設定 ※メールドライバ

トークンの払い出し

ディビジョンテーブル作成

監視アダプタ ※Prometheusアダプタ

#### 【作業実行】

ルールの作成・登録

ルールマッチング  
およびアクション実行

ディビジョンテーブルファイル作成 ※エクセル操作

ディビジョンテーブルファイルアップロード

テストリクエスト

プロダクション適用

実行

アクション実行結果の確認

## 2.1 本書のシナリオ (3/3)

### シナリオ

本シナリオでは、「インスタンスダウン状態が5分続いた場合に Critical としてアラート発火」するという監視ルールのYAML ファイルを使用します。

alert\_rules.yml

```
groups:
- name: example
  rules:
  #ダウンの状態が5分間継続していたらアラートを出す
  - alert: instance_down
    expr: up == 0
    for: 5m
    labels:
      severity: critical
  #通知内容
  annotations:
    summary: "Instance {{ $labels.instance }} down"
    description: "{{ $labels.instance }} of job {{ $labels.job }} has been down for more than 5 minutes."
```

### 3. 事前設定

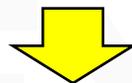
## 3.1 事前設定

### 事前設定

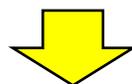
ルールマッチングした場合メール通知するために以下の設定が必要となります。

Zabbix 実習編同様、事前設定が必要となりますので [Exastro OASE Zabbix連携（実習）5.事前設定](#) を参照し以下の設定を行ってください。

アクション設定 メールドライバの追加



トークンの払い出し  
※テストリクエスト送信時に利用



ディビジョンテーブルファイルの作成

#### POINT

事前に[環境構築マニュアル-ドライバインストール編](#)を参照のうえメールドライバをインストールしてください。

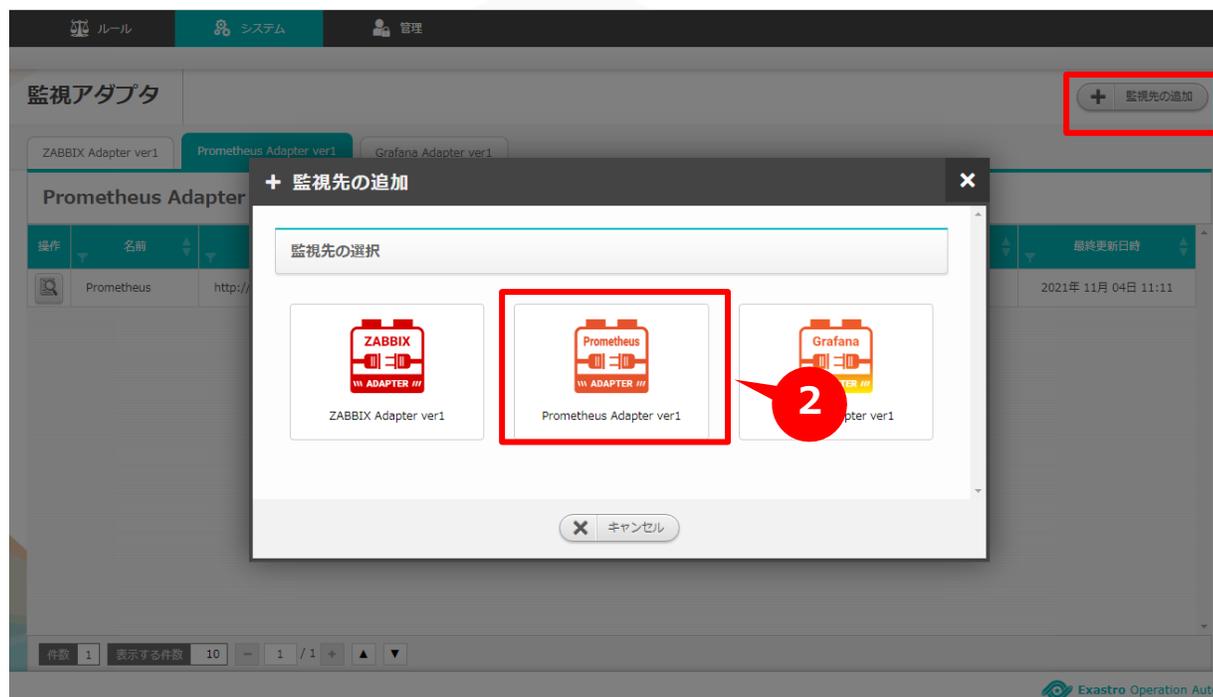
## 3.2 監視アダプタ設定 (1/3)

### システム > 監視アダプタ

#### 監視先の追加

Prometheusアダプタを追加します。

- ① 監視先の追加を押下する
- ② Prometheus Adapter ver1を押下する



※ドライバをインストールしていない場合、上記の画面は表示されません。

## 3.2 監視アダプタ設定 (2/3)

### システム > 監視用アダプタ

#### Prometheusで設定したアラートを使用する場合

- ① Prometheus Adapter ver1の必要な情報を入力する

名前 \* @ Prometheus

URI \* @ http://\*\*\*

ディシジョンテーブル名選択 warning\_test

監視情報 Prometheus項目

イベント発生日時 \* data.alerts[].activeAt

インスタンス情報 \* data.alerts[].labels.instance

条件名 Prometheus項目

アラート \* @ data.alerts[].labels.alertname

対象 \* @ data.alerts[].labels.instance

最終更新者 システム管理者

最終更新日時 2021年11月12日10:29

キャンセル 保存

項目	設定値
名前	(任意の文字列)
URI	http://xx.xxx.xxx.xxx:9090/api/v1/alerts (API(alerts)のエンドポイント)
ディシジョンテーブル名選択	warning_test (イベントリクエストの実施対象となるディシジョンテーブル名をプルダウン名から選択する)

### システム > 監視用アダプタ

#### Prometheusで設定したアラートを使用する場合

- ① と都合情報を入力する
- ② 保存を押下する

Prometheus側で設定したアラートを使用する場合の設定値は何か変わる

※設定値は対象環境によって変わりますのでご注意ください

Prometheus Adapter ver1

名前 *	Prometheus												
URI *	http://***												
ディビジョンテーブル名選択	***												
突合情報	<table><thead><tr><th>監視情報</th><th>Prometheus項目</th></tr></thead><tbody><tr><td>イベント発生日時 *</td><td>data.alerts[].activeAt</td></tr><tr><td>インスタンス情報 *</td><td>data.alerts[].labels.instance</td></tr></tbody></table> <table><thead><tr><th>条件名</th><th>Prometheus項目</th></tr></thead><tbody><tr><td>アラート *</td><td>data.alerts[].labels.alertname</td></tr><tr><td>対象 *</td><td>data.alerts[].labels.instance</td></tr></tbody></table>	監視情報	Prometheus項目	イベント発生日時 *	data.alerts[].activeAt	インスタンス情報 *	data.alerts[].labels.instance	条件名	Prometheus項目	アラート *	data.alerts[].labels.alertname	対象 *	data.alerts[].labels.instance
監視情報	Prometheus項目												
イベント発生日時 *	data.alerts[].activeAt												
インスタンス情報 *	data.alerts[].labels.instance												
条件名	Prometheus項目												
アラート *	data.alerts[].labels.alertname												
対象 *	data.alerts[].labels.instance												
最終更新者	システム管理者												
最終更新日時	2021年11月12日10:29												

キャンセル 保存

1

項目	設定値						
突合情報	<table><thead><tr><th>監視情報</th><th>Prometheus項目</th></tr></thead><tbody><tr><td>イベント発生日時</td><td></td></tr><tr><td>インスタンス情報</td><td></td></tr></tbody></table>	監視情報	Prometheus項目	イベント発生日時		インスタンス情報	
	監視情報	Prometheus項目					
	イベント発生日時						
	インスタンス情報						
<table><thead><tr><th>条件名</th><th>Prometheus項目</th></tr></thead><tbody><tr><td>アラート名</td><td></td></tr><tr><td>対象</td><td></td></tr></tbody></table>	条件名	Prometheus項目	アラート名		対象		
条件名	Prometheus項目						
アラート名							
対象							

## 4. 作業実行

# 4.1 ディジジョンテーブルファイル作成

## ルール > ディジジョンテーブル

### ディジジョンテーブルファイルのダウンロードおよび作成

- ① 事前設定で作成したディジジョンテーブルの「ダウンロードボタン」を押下しディジジョンテーブルファイルをダウンロードする
- ② 以下の内容を記述作成する(詳細については[Exastro OASE docs](#)をご参照ください。)

ルール属性	アラート(任意)属性(可)	対象(完全一致)	ルール名(必須)	発生事象(必須) ※不要の場合は「X」を記載	対処概要(必須) ※不要の場合は「X」を記載	アクション種別(必須)	アクションパラメータ情報(必須) ※ダブルクォーテーションは使用不可	送信メールパラメータ情報(必須) ※ダブルクォーテーションは使用不可 ※不要の場合は「X」を記載	リトライ回数	リトライ間隔	抑制回数	抑制間隔	条件回数	条件期間	大グループ(必須) ※不要の場合は「X」を記載	小グループ(必須) ※不要の場合は「X」を記載	承認パラメータ情報	無効日	無効日
Prometheus連携	instance_down	node-exporter:9090	rule_warning	インスタンスダウン	アラート通知	mail(ver1)	MAIL_NAME=(メールドライバ名), MAIL_TO=,MAIL_CC=,MAIL_BCC=, MAIL_TEMPLATE=(メールテンプレート名)	X	1	1	1	1	X	X	X	X	X		



ルール説明	Prometheus連携	リトライ間隔	1
アラート	instance_down	リトライ回数	1
対象	node-exporter:9090	抑制間隔	1
ルール名	rule_warning	抑制回数	1
発生事象	インスタンスダウン	条件回数	X
対処概要	アラート通知	条件期間	X
アクション種別	mail(ver1)	大グループ	X
アクションパラメータ情報	MAIL_NAME= (メールドライバ名), MAIL_TO=,MAIL_CC=,MAIL_BCC=, MAIL_TEMPLATE=(メールテンプレート名)	優先順位	X
承認パラメータ情報	X	小グループ	X
		優先順位	X

ディジジョンテーブルファイル作成

ディジジョンテーブルファイル  
アップロード

テストリクエスト

プロダクション適用

アラート発火

アクション実行結果の確認

## 4.2 ディジジョンテーブルファイルのアップロード

### ルール > ルール

#### ディジジョンテーブルのアップロード

作成したファイルを選択しアップロードします。

- ① 「ファイル選択」を押下し作成したディジジョンテーブルファイルを選択する
- ② 「アップロード」を押下する
- ③ ステージング適用完了になっていることを確認する

Dashboard ルール システム 管理

システム管理者

ステージング適用ルール

テストリクエスト

ファイルを選択: id00000000011.xlsx

アップロード

過去を含め表示 OFF

操作	ディジジョンテーブル名	ルールファイル	適用ステータス	作業ステータス	最終更新者	最終更新日時
<input checked="" type="checkbox"/>	warning_test	id00000000011.xlsx	検証未実施	ステージング適用完了	システム管理者	2021年 11月 08日 16:59

## 4.3 テストリクエスト

### ルール > ルール

#### 登録したルールのテスト

- ① 「テストリクエスト」を押下する
- ② ディジジョンテーブルタブを押下し作成したディジジョンテーブルをプルダウン選択する
- ③ 設定タブを押下し作成したルールと合致する値を入力する
- ④ 実行ボタンを押下しルールに合致したことを確認する
- ⑤ 閉じるボタンを押下しステータスを検証完了にする

ディジジョンテーブルファイル作成

ディジジョンテーブルファイルアップロード

テストリクエスト

プロダクション適用

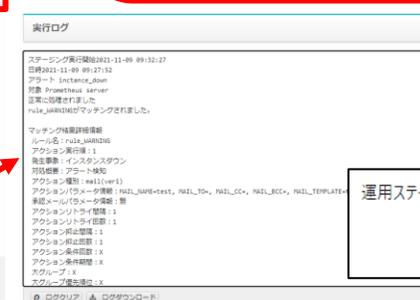
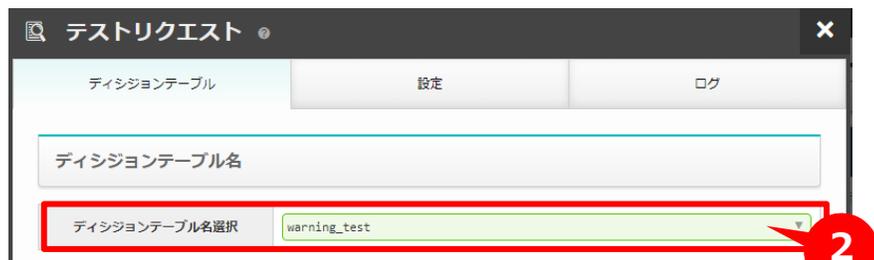
アラート発火

アクション実行結果の確認



3 以下のように記述

アラート	instance_down
対象	node-exporter:9090 (環境によって読み替える必要有)



運用ステータスを「検証完了」にしてよろしいですか？

OK

5

## 4.4 プロダクション適用

### ルール > ルール

#### 検証完了したルールを本番環境で使用できるようにする

- ① チェックマークを押下しプロダクション適用を開始する
- ② 作業ステータスがプロダクション適用完了になっていることを確認する

ステージング適用ルール

✔ テストリクエスト    📁 ファイルを選択: ファイルが選択されていません。    📄 アップロード    🕒 過去を含め表示 OFF

操作	ディビジョンテーブル名	ルールファイル	運用ステータス	作業ステータス	最終更新者	最終更新日時
<input checked="" type="checkbox"/>	warning_test	id00000000011.xlsx	検証完了	ステージング適用完了	システム管理者	2021年 11月 09日 10:21

プロダクション適用します。よろしいですか?

OK    キャンセル

プロダクション適用ルール

🕒 過去を含め表示 OFF

操作	ディビジョンテーブル名	ルールファイル	運用ステータス	作業ステータス	最終更新者	最終更新日時
<input type="checkbox"/>	warning_test	id00000000011.xlsx	プロダクション適用	プロダクション適用完了	システム管理者	2021年 11月 09日 10:34

- ディビジョンテーブルファイル作成
- ディビジョンテーブルファイルアップロード
- テストリクエスト
- プロダクション適用**
- アラート発火
- アクション実行結果の確認

### node-exporter の起動と確認

#### ① node-exporter を起動する

```
# docker-compose up -d node-exporter
Starting node-exporter ... done
```

#### ② up になっていることを確認する

```
# docker-compose ps
```

Name	Command	State	Ports
node-exporter	/bin/node_exporter	Up	0.0.0.0:9100->9100/tcp, ::: 9100->9100/tcp
prometheus	/bin/prometheus --config.f	Up	0.0.0.0:9090->9090/tcp, ::: 9090->9090/tcp
...	...		

ディビジョンテーブルファイル作成

ディビジョンテーブルファイル  
アップロード

テストリクエスト

プロダクション適用

アラート発火

アクション実行結果の確認

### node-exporter の停止(アラート発火)

#### ③ node-exporter を停止する

```
# docker-compose stop node-exporter
Stopping node-exporter ... done
```

#### ④ Exit2 になっていることを確認する

```
# docker-compose ps
      Name                Command                                State          Ports
-----
node-exporter    /bin/node_exporter                    Exit2
prometheus      /bin/prometheus --config.f            Up             0.0.0.0:9090->9090/tcp, :::
...                                                     9090->9090/tcp
```

ディビジョンテーブルファイル作成

ディビジョンテーブルファイル  
アップロード

テストリクエスト

プロダクション適用

アラート発火

アクション実行結果の確認

## 4.6 アクション実行結果の確認

### 各種画面を確認する

OASE の「リクエスト履歴」と「アクション履歴」画面に該当のディシジョンテーブル名が上がっていることを確認する。

アクション設定（メールドライバ）で設定した件名・本文が届いていることを確認する。

ルールマッチ状況	リクエスト種別	ディシジョンテーブル名	リクエスト受信日時	イベント情報
✓	ステージング環境	warning_test	2021年11月11日17:24	{"EVENT_INFO":["instance_down","noc

ディシジョンテーブルファイル作成

ディシジョンテーブルファイル  
アップロード

テストリクエスト

プロダクション適用

アラート発火

アクション実行結果の確認



**Exastro**