

# EdgeX Grid Bot 修正版 (Koyeb)

## 1. 設置方法

### 1.1 Koyebを開く

Koyeb - service newを開く

必ず Worker かつ Github を選択すること

The screenshot shows the Koyeb interface for creating a new service. On the left, there's a sidebar with 'Create service' and 'Deploy one-click app' sections. Under 'Create service', 'Worker' is selected and highlighted with a grey background. Under 'Deploy one-click app', 'GitHub' is also highlighted. The main area is titled 'Let's deploy a new service' and contains two options: 'Create worker service' (selected) and 'Docker'. The 'Create worker service' section is described as 'For long running processes like background processing and task execution'. It shows icons for GitHub and Docker, with detailed descriptions: 'GitHub' for connecting, building, and deploying code from a GitHub repository, and 'Docker' for deploying Docker images from any public or private registries.

Public Github Repositoryに以下のURLを入れる

<https://github.com/zaramcei/edgex-grid>

## Create a new service

Import the GitHub repository you want to deploy or use an existing public repository

Import project 1 of 4

Search GitHub repository

Refresh

Missing a GitHub repository? [Edit GitHub app permissions](#)

### Public GitHub repository

Use a public repository by entering the URL below. Features like auto-deploy are not available when using a public repository.

<https://github.com/zaramcei/edgem-grid>

Import

Back

必ず以前稼働していたリージョンと同じ場所、同じCPUを選択する

## Create a new service

Pick the instance and regions to deploy

Choose instance and regions 3 of 4

CPU Eco

CPU Standard

GPU

Nano

0.25 vCPU 256 MB RAM 2 GB Disk

\$0.0036/h

\$2.68/month

Micro

0.5 vCPU 512 MB RAM 5 GB Disk

\$0.0072/h

\$5.36/month

Deploy in regions

Metropolitan Continental



Frankfurt

751ms



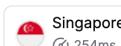
Paris

721ms



San Francisco

323ms



Singapore

254ms



Tokyo

23ms



Washington, D.C.

506ms

Small

1 vCPU 1 GB RAM 10 GB Disk

\$0.0144/h

\$10.71/month

Medium

\$0.0288/h

Back

Next

Dockerfileを選択

## Create a new service

Choose whether to build your code with a Dockerfile or a Buildpack

Build options: 2 of 4

<input type="radio"/> <b>Buildpack</b> Detect your project's language, install dependencies, and package the environment into a runnable image.	<input checked="" type="radio"/> <b>Dockerfile</b> Build a runnable container image for your project. Requires your repository to contain a Dockerfile.
--	--

> Customize Dockerfile settings

[Back](#) [Next](#)

# 1.2 環境変数を以下の解説に従って設定する

The screenshot shows the configuration page for the 'edgex-grid' service. The top navigation bar includes 'Overview', 'Metrics', 'Console', and 'Settings' (which is selected). The main configuration area is divided into several sections:

- Service type:** Worker
- Source:** zaramcei/edgex-grid (branch: main)
- Builder:** Dockerfile
- Environment variables and files:** 15 variables, no files
- Instance:** Small (1 vCPU, 1GB RAM, 10GB Disk) | Standard | Paris
- Scaling:** Fixed (1 instance/region)
- Volumes:** No volumes configured
- Service name:** edgex-grid

At the bottom, there are two buttons: 'Save and deploy' (green) and 'Save' (grey).

**Pause service**: Once this operation is performed, your service will be paused and all active deployments stopped. You can wake-up your service by clicking the "Resume" button below.

Buttons: Resume (grey), Pause (orange).

Environment variable and files を開く

以下の画像はあくまで設定例なのでこの画像の通りに設定せず、指示されたパラメータで稼働させてください。

[Environment variables](#) [Files](#)

Configure manually or drag and drop file with multiple variables

Name	Value ⓘ	
EDGE_X_ACCOUNT_ID		
EDGE_X_ASSET_LOSSCUT_PERCENTAGE	3	
EDGE_X_ASSET_TAKE_PROFIT_PERCENTAGE	1	
EDGE_X_BALANCE_RECOVERY_ENABLED	true	
EDGE_X_BASE_URL	https://pro.edgex.exchange	
EDGE_X_CONTRACT_ID	10000001	
EDGE_X_GRID_FIRST_OFFSET_USD	10	
EDGE_X_GRID_LEVELS_PER_SIDE	10	
EDGE_X_GRID_OP_SPACING_SEC	0.1	
EDGE_X_GRID_SIZE	0.002	
EDGE_X_GRID_STEP_USD	10	
EDGE_X_INITIAL_BALANCE_USD	560	
EDGE_X_LEVERAGE	100	
EDGE_X_RECOVERY_ENFORCE_LEVEL_USD	5	
EDGE_X_STARK_PRIVATE_KEY		

設定したら Save and deploy を押して起動する

## 2. 設定値の解説

### 2.1 基本設定（必須）

- EDGE\_X\_ACCOUNT\_ID : あなたのEdgeXアカウントID（数値）
- EDGE\_X\_STARK\_PRIVATE\_KEY : あなたの秘密鍵
- EDGE\_X\_BASE\_URL : EdgeX APIのベースURL（デフォルト: https://pro.edgex.exchange）
- EDGE\_X\_CONTRACT\_ID : 取引する銘柄のID（例: BTC-PERPは 10000001）
- EDGE\_X\_LEVERAGE : 取引レバレッジ倍率（例: 100） - 損益率の計算に使用

### 2.2 グリッド設定

- EDGE\_X\_GRID\_STEP\_USD : グリッド幅（USD） - 価格レベル間の間隔（例: 10, 50, 100）

- EDGEX\_GRID\_FIRST\_OFFSET\_USD : 中央価格からの初回オフセット (USD) (例: 10 , 50 )
- EDGEX\_GRID\_LEVELS\_PER\_SIDE : 片側 (買い/売り) のグリッド本数 (例: 5 , 10 )
- EDGEX\_GRID\_SIZE : 1本あたりの注文数量 (BTC) (例: 0.001 , 0.002 )
- EDGEX\_GRID\_OP\_SPACING\_SEC : 注文間の待機時間 (秒) - レート制限回避用 (デフォルト: 0.1 )

## 2.3 ポジションベースのロスカット/利確設定 (設定が難しくロスカットが頻発するので非推奨)

基本的には2.4のアセットベースのロスカットの方を設定してください。

保有中のポジションの未実現損益 (レバレッジを考慮) が指定の閾値に達したら、そのポジションを成行注文で自動的にクローズする機能です。

### 拳動:

- ポジション単位で個別に損益を監視
- 損益率 = (未実現損益 / ポジション価値) × レバレッジ倍率
- 閾値に達した瞬間に成行でポジションクローズ
- クローズ後、すぐに通常のグリッド取引を再開

### 設定:

- EDGEX\_POSITION\_LOSSCUT\_PERCENTAGE : ポジションの未実現損益が指定%以下になったら自動ロスカット (例: 10 = -10%でロスカット)
  - null または未設定の場合はロスカットを実行しない
- EDGEX\_POSITION\_TAKE\_PROFIT\_PERCENTAGE : ポジションの未実現損益が指定%以上になったら自動利確 (例: 10 = +10%で利確)
  - null または未設定の場合は利確を実行しない

## 2.4 資産ベースのロスカット/利確設定

口座全体の総資産 (残高+未実現損益) を監視し、ポジション開始時の資産を基準にして指定%の変動があった場合に全ポジションをクローズする機能です。ポジション単位ではなく口座全体の資産を保護します。

### 拳動:

- 総資産 = 現在の残高 + 全ポジションの未実現損益

- 基準資産 (initial\_asset) はポジションを持っていない時の残高
- ポジション保有中に総資産が基準資産から指定%変動したら発動
- 発動時の処理:
  - i. 全ポジションを成行で即座にクローズ
  - ii. 全ての未約定注文をキャンセル
  - iii. クローズ完了を確認後、基準資産を現在の残高に更新
  - iv. 30秒のクールダウン後、通常のグリッド取引を再開
- 次回ポジションは新しい基準資産から損益を計算

## 設定:

- EDGEX\_ASSET\_LOSSCUT\_PERCENTAGE : 総資産（残高+未実現損益）がポジション開始時の資産から指定の%減少したら全ポジションクローズ（デフォルト: 3.0 = -3%）
  - 例: 0.05 に設定すると、総資産が-0.05% (約-0.2 USD at 400 USD) 減少時にロスカット
  - ポジション単位ではなく、口座全体の資産を保護する機能
- EDGEX\_ASSET\_TAKE\_PROFIT\_PERCENTAGE : 総資産が初期資産から指定%増加したら全ポジションクローズ（デフォルト: 5.0 = +5%）
  - 0 または未設定の場合は利確を実行しない

## 2.5 全戻し全決済による資産リカバリー設定

資産が減少した後のポジションで含み益が出ているケースのうち、未実現損益と現在の（減少した）資産価格の合計値が初期資産額に到達したらポジションを全部クローズする機能に関する設定

- EDGEX\_BALANCE\_RECOVERY\_ENABLED : バランスリカバリー機能の有効/無効 (true or false)
  - 残高が初期残高まで回復したら全ポジションをクローズする機能
- EDGEX\_INITIAL\_BALANCE\_USD : 初期残高 (USD) - リカバリーの基準値 (例: 400.0)
- EDGEX\_RECOVERY\_ENFORCE\_LEVEL\_USD : リカバリーモードを有効にする最小損失額 (USD) (デフォルト: 3.0)
  - 初期残高から指定額以上の損失が出た場合のみリカバリーモードが発動

## 2.6 稼働スケジュール設定

リモートのJSONファイルから稼働スケジュールを取得し、指定された時間帯のみBotを稼働させる機能です。

- スケジュールURL: <https://zaramcei.github.io/edgex-grid/schedule/schedule.json>
- スケジュールが取得できない場合でも、既存のスケジュール情報で動作を継続

## スケジュール機能の有効/無効:

- `EDGE_USE_SCHEDULE` : スケジュール機能の有効/無効 (デフォルト: `true`)
  - 未設定または空欄: スケジュール機能が有効
  - `false`, `0`, `no`: スケジュール機能が無効 (常時稼働)

## スケジュールタイプの選択:

- `EDGE_USE_SCHEDULE_TYPE` : 使用するスケジュールタイプ (デフォルト: `normal`)
  - `normal`: 通常稼働 (短めの稼働時間)
  - `aggressive`: 攻め稼働 (長めの稼働時間)
  - スケジュールJSONに定義された任意のタイプ名を指定可能

## 設定例:

```
EDGE_USE_SCHEDULE_TYPE=normal      # 通常稼働スケジュールを使用  
EDGE_USE_SCHEDULE_TYPE=aggressive # 攻め稼働スケジュールを使用
```

## 挙動:

- 5分ごとにリモートからスケジュールJSONを取得・更新
- スケジュール時間外の場合、グリッド注文は発注されず待機状態になる
- `lot_coefficient`: グリッド注文のサイズに掛ける係数 (例: `0.5` なら通常の0.5倍のサイズで注文)

## スケジュール外への移行時の動作:

稼働時間内から時間外に移行した際の動作を環境変数で制御できます。

- `EDGE_OUT_OF_SCHEDULE_ACTION` : スケジュール外に出た時の動作 (デフォルト: `auto`)
  - `nothing`: 全指値注文をキャンセルし、ポジションはそのまま維持して終了。裁量でクローズしたい場合に使用
  - `auto`: 現在価格から5ドル有利な価格で指値クローズ注文を出し、1分以内に約定しなければ成行でクローズすることで注文手数料の削減とリスク管理を両立
  - `immediately`: 即座に成行でポジションをクローズ、注文手数料はTAKERだが損失拡大リスクは即座に無くなる

## 2.7 ポジションサイズ制限 (REDUCE\_MODE)

ポジションサイズが一定の閾値を超えた場合、ポジションを積み増す方向の注文をスキップし、ポジションを減らす方向の注文のみを許可する機能です。

BTCとRATIOの2種類から**どちらか一方のみ**を設定する必要があります。両方設定した場合や両方未設定の場合、LIMITよりREDUCE\_ONLYの方が大きい（矛盾していて意味のない設定）場合は起動時にエラーで終了します。

### BTC絶対値による制限

ポジションサイズ (BTC) の絶対値で制限します。

- EDGEX\_POSITION\_SIZE\_LIMIT\_BTC : この値以上で REDUCE\_MODE に突入 (例: 0.1 )
- EDGEX\_POSITION\_SIZE\_REDUCE\_ONLY\_BTC : この値を下回るまで積み増し禁止 (例: 0.05 )

例:

```
EDGEX_POSITION_SIZE_LIMIT_BTC=0.1  
EDGEX_POSITION_SIZE_REDUCE_ONLY_BTC=0.05
```

ポジション: 0.08 BTC → 通常モード (BUY/SELL両方OK)  
ポジション: 0.10 BTC → REDUCE\_MODE突入 (積み増し禁止)  
ポジション: 0.06 BTC → REDUCE\_MODE継続  
ポジション: 0.04 BTC → REDUCE\_MODE解除 (BUY/SELL両方OK)

### RATIO (総資産比率) による制限

ポジション価値が総資産に対して何%かで制限します。

動作確認時は

- EDGEX\_POSITION\_SIZE\_LIMIT\_RATIO : 0.5
- EDGEX\_POSITION\_SIZE\_REDUCE\_ONLY\_RATIO : 0.25

の設定がおすすめです。

計算式:

```
current_ratio = (現在BTC価格 × ポジションサイズ) / 総資産(initial_asset)
```

- EDGEX\_POSITION\_SIZE\_LIMIT\_RATIO : この割合(%)以上で REDUCE\_MODE に突入 (例: 50.0 )
- EDGEX\_POSITION\_SIZE\_REDUCE\_ONLY\_RATIO : この割合(%)を下回るまで積み増し禁止 (例: 30.0 )

例:

```
EDGEX_POSITION_SIZE_LIMIT_RATIO=50.0  
EDGEX_POSITION_SIZE_REDUCE_ONLY_RATIO=30.0
```

総資産: 10,000 USD / BTC価格: 100,000 USD

ポジション: 0.04 BTC → ratio=40% → 通常モード  
ポジション: 0.05 BTC → ratio=50% → REDUCE\_MODE突入  
ポジション: 0.035 BTC → ratio=35% → REDUCE\_MODE継続  
ポジション: 0.025 BTC → ratio=25% → REDUCE\_MODE解除

挙動:

1. ポジションサイズが LIMIT 値以上になると REDUCE\_MODE に突入
2. REDUCE\_MODE 中はポジションを積み増す方向の注文をスキップ
  - LONG保持中: BUY注文をスキップ、SELL注文のみ許可
  - SHORT保持中: SELL注文をスキップ、BUY注文のみ許可
3. ポジションサイズが REDUCE\_ONLY 値を下回ると REDUCE\_MODE を解除
4. 通常モードに戻り、両方向の注文を再開

設定方法:

バリデーション:

- BTCとRATIOの両方を設定 → エラー終了
- BTCもRATIOも未設定 → エラー終了
- LIMITのみ設定してREDUCE\_ONLY未設定 → エラー終了
- REDUCE\_ONLY >= LIMIT → エラー終了 (REDUCE\_ONLYはLIMITより小さい必要あり)