



IT Automation

サーバ分散型HA構成

インストールマニュアル

※本書では「Exastro IT Automation」を「ITA」として

Exastro IT Automation Version 1.8

Exastro developer

はじめに

1. 本資料について

- ・本書ではITAのHA構成構築のための手動インストールの手順について説明します。
- ・本書は以下のデプロイ例を構築することを想定して作成しております。

https://exastro-suite.github.io/it-automation-docs/install_ja.html

Exastro IT Automation ハイアベイラビリティ構成

- ・目的の環境構築のために、本書で参照するマニュアルの一覧は以下になります。環境を構築する際は、「構築する順番」の示す通りにマニュアルを参照してください。
※ 本手順書による環境構築はrootユーザで実行してください。

構築する順番	構築環境	マニュアル名
1	AWS	AWS設定
2	backyard	HA構成(backyard)
3	Ansible	HA構成(Ansible)
4	Web/AP	HA構成(Web・AP)

2. 動作要件

- ・ITAの動作要件については、以下のドキュメントを参照してください。

https://exastro-suite.github.io/it-automation-docs/asset/Documents_ja/Exastro-ITAシステム構成/環境構築ガイド基本編.pdf

IT Automation BASE システム構成/環境構築ガイド 基本編

1.1 サーバ動作要件

- ・連携ドライバの動作要件については下記のドキュメントをそれぞれご確認ください。

https://exastro-suite.github.io/it-automation-docs/asset/Documents_ja/Exastro-ITAシステム構成/環境構築ガイドAnsible-driver編.pdf

Ansible-driver システム構成/環境構築ガイド Ansible-driver編

3. システム要件

3. ポート開放

- ITAでは、各システム(サーバ)間の通信に以下のドキュメントの一覧にあるポートを使用するので、そちらを参照して必要なポートを開放してください。

[https://exastro-suite.github.io/it-automation-docs/asset/Documents ja/Exastro-ITA システム構成/環境構築ガイド 基本編.pdf](https://exastro-suite.github.io/it-automation-docs/asset/Documents%20ja/Exastro-ITA%20システム構成%20環境構築ガイド%20基本編.pdf)

IT Automation BASE システム構成/環境構築ガイド 基本編

2.2 システムの通信要件

- クラスタ構成サーバを同一ネットワーク上に構成し、以下の通信ポートが利用可能であること。

通信種別	ポート番号
ssh(scp)	22/tcp
MYSQL/Aurora(RDS)	3306/tcp
NFS	2409/tcp
Pacemaker	2224/tcp
Corosync	5405/udp

※Pacemaker、Corosyncポート参考URL:

https://access.redhat.com/documentation/ja-jp/red_hat_enterprise_linux/7/html/high_availability_add-on_reference/s1-firewalls-haar

4. 作業前提

- 各種ソフトウェアのインストールはオンライン環境での手順を記載しています。オフラインで実行する場合はソフトウェアの依存関係を構築サーバと合致させた上で、事前のライブラリ収集を行ってください。
- 構築手順の前に、以下の基本的なOS設定を実施して下さい。なお、特に断らない限り、OS基本設定は両方のサーバで同一に揃える必要があります。

- SELinuxの無効化
SELinuxが有効な状態でHAクラスタを正常に運用するには、SELinuxに関する、極めて高度な知識と経験が必要になります。本手順ではSELinuxを無効での検証となっておりますが、無効化はユーザの自己責任で対応ください。

本資料では標準インストールの機能有効化を前提としています。

ita_answers.txtの内容

ita_base:yes
createparam:yes
hostgroup:yes
ansible_driver:yes
cobbler_driver:no
terraform_driver:yes
cicd_for_iac:no

ITAの起動サービスは、ITAのバージョン、インストールする機能によって変動します。

本資料は、ITA ver1.8.xの標準インストールを想定している為、

上記の想定と異なるバージョンの利用と機能のインストールを行う場合は、

本手順の下記の箇所にて対象サービス内容について、適宜修正が必要となります。

- 構築手順「3.ITAサービス停止設定」の対象サービス

AWS設定手順書

ここでは事前準備として、AWSの設定手順を説明していきます。
IPアドレスや設定値は任意で変更をお願いいたします。

今回の利用するサービスは下記のとおりです。

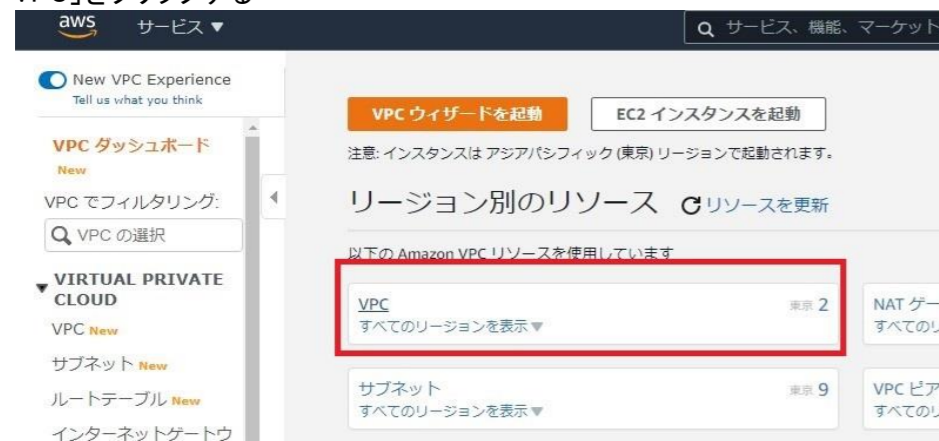
利用サービス一覧
1.VPC
2.サブネット
3.インターネットゲートウェイ
4.ルートテーブル
5.セキュリティグループ
6.IAM
7.RDS
8.EFS
9.Route53
10.EC2
11.ElasticIP
12.Loadbarancer

1.VPC

1.コンソールにログインし、左上の「サービス」から「VPC」をクリックする



2.「VPC」をクリックする



AWS設定手順書

3.「VPCを作成」をクリックする



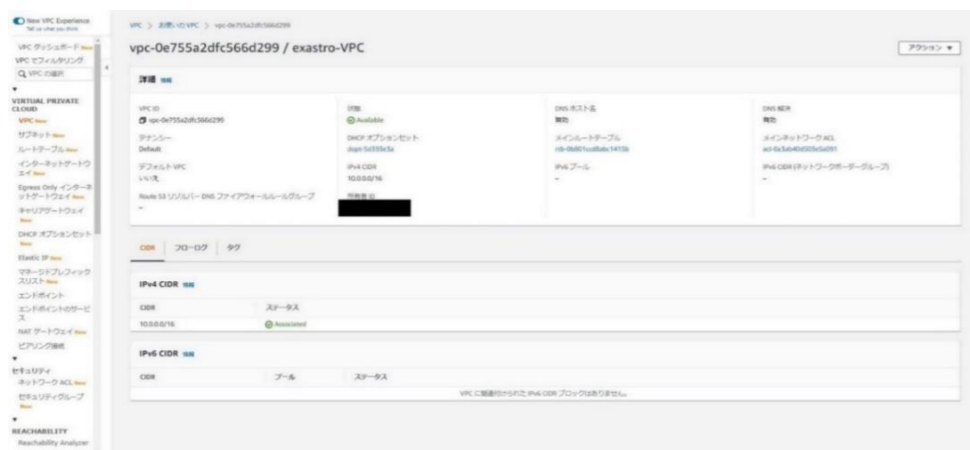
4.VPCの情報を入力する



入力例)

名前タグ	IPv4 CIDRブロック
exastro-VPC	10.0.0.0/16

5.VPCが作成できたことを確認する



AWS設定手順書

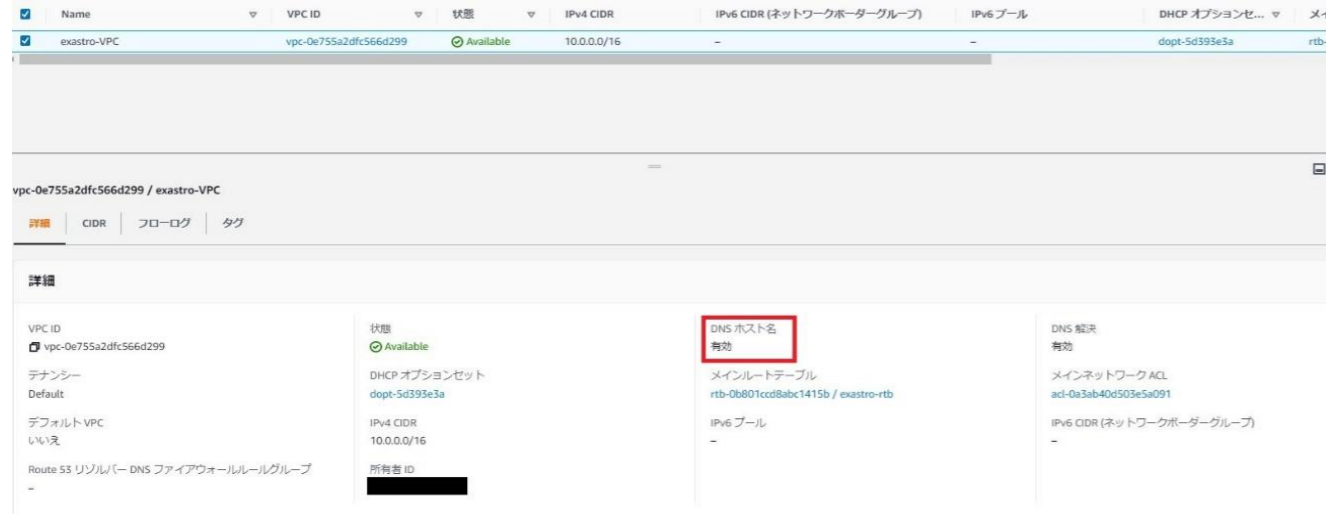
6.2の画面に戻りアクションから「DNSホスト名を編集」をクリックする



7.DNSホスト名を有効化し、「変更を保存」をクリックする



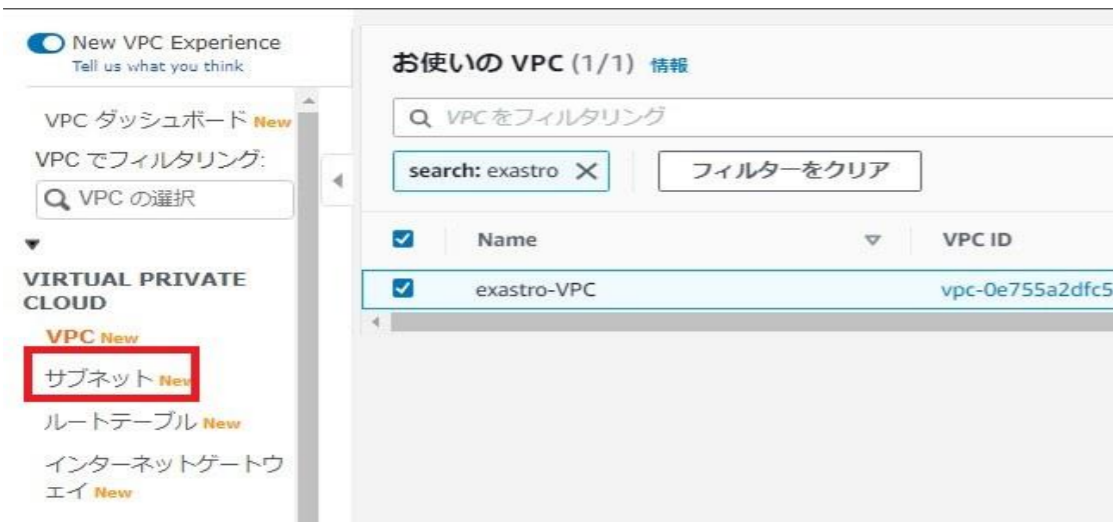
8.DNSホスト名が有効になっていることを確認する



AWS設定手順書

2.サブネット

1.左側のメニューから「サブネット」をクリックする



2.「サブネットを作成」をクリックする



AWS設定手順書

3.先ほど作成したVPCをクリックし、サブネットの情報を入力する ※合計で6つのサブネットを作成する

入力例)

サブネット名	アベイラビリティゾーン	IPv4 CIDRブロック
public-a	アジアパシフィック (東京)/ap-northeast-1a	10.0.0.0/24
public-c	アジアパシフィック (東京)/ap-northeast-1c	10.0.1.0/24
private-a	アジアパシフィック (東京)/ap-northeast-1a	10.0.100.0/24
private-c	アジアパシフィック (東京)/ap-northeast-1c	10.0.101.0/24
protect-rds-a	アジアパシフィック (東京)/ap-northeast-1a	10.0.192.0/21
protect-rds-c	アジアパシフィック (東京)/ap-northeast-1c	10.0.200.0/21

4.作成後、状態がAvailableになっていることを確認する

<input type="checkbox"/>	protect-rds-a	subnet-0feaff149f4c3cd2b	✔ Available	vpc-09dbe5341dd5c28c0 exastro-VPC	10.0.192.0/21
<input type="checkbox"/>	private-c	subnet-03493ada234c0e87b	✔ Available	vpc-09dbe5341dd5c28c0 exastro-VPC	10.0.101.0/24
<input type="checkbox"/>	public-a	subnet-031fc186b7c1e20f5	✔ Available	vpc-09dbe5341dd5c28c0 exastro-VPC	10.0.0.0/24
<input type="checkbox"/>	private-a	subnet-0c0466242e820bad7	✔ Available	vpc-09dbe5341dd5c28c0 exastro-VPC	10.0.100.0/24
<input type="checkbox"/>	protect-rds-c	subnet-059de53dabed42c42	✔ Available	vpc-09dbe5341dd5c28c0 exastro-VPC	10.0.200.0/21
<input type="checkbox"/>	public-c	subnet-02cc4aebf47e8f190	✔ Available	vpc-09dbe5341dd5c28c0 exastro-VPC	10.0.1.0/24

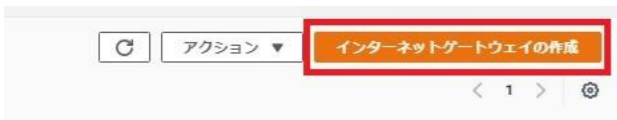
AWS設定手順書

3.インターネットゲートウェイ

1.左側のメニューから「インターネットゲートウェイ」をクリックする



2.「インターネットゲートウェイの作成」をクリックする



3.インターネットゲートウェイの情報を入力する

VPC > インターネットゲートウェイ > インターネットゲートウェイの作成

インターネットゲートウェイの作成 情報

インターネットゲートウェイは、VPCをインターネットに接続する仮想ルーターです。新しいインターネットゲートウェイを作成するには、ゲートウェイの名前を以下から指定します。

インターネットゲートウェイの設定

名前タグ
[Name] というキーと、既定した値を使用し、タグを作成します。

タグ - オプション

タグは、AWS リソースに割り当てられるラベルです。各タグはキーとオプションの値で構成されています。タグを使用してリソースを検索およびフィルタリングしたり、AWS のコストを追跡したりできます。

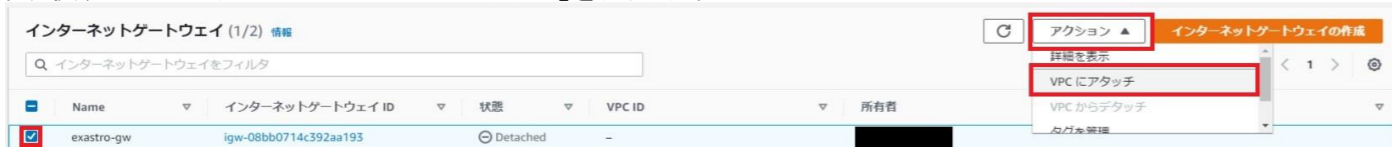
キー × 値 - オプション ×

さらに 49 個のタグ、を追加できます。

入力例)
名前タグ exastro-gw

AWS設定手順書

4.作成後、左上のアクションから「VPCにアタッチ」をクリックする



5.作成したVPCを選択し、「インターネットゲートウェイのアタッチ」をクリックする



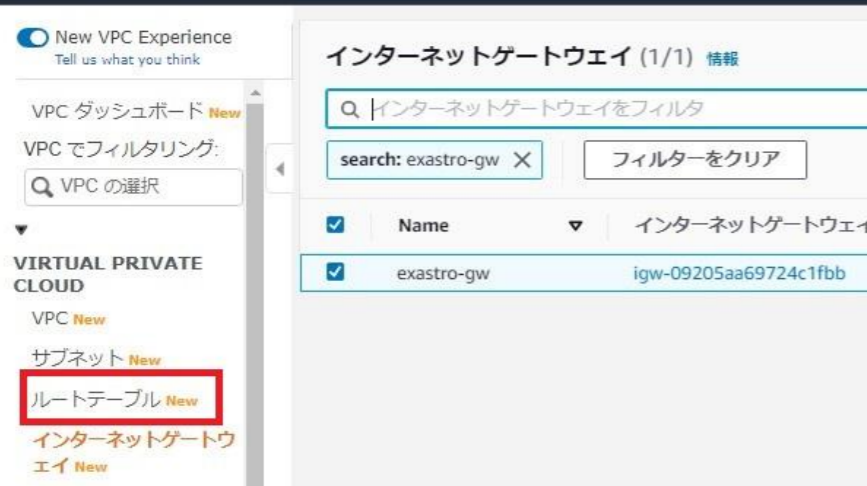
6.状態がAvailableになっていることを確認する



AWS設定手順書

4. ルートテーブル

1. 左側のメニューから「ルートテーブル」をクリックする



2. 作成したVPCのIDで検索ボックスから検索する



3. 対象のルートテーブルの下の「ルート」タブをクリックする



AWS設定手順書

4. ルートの情報を追加する

VPC > ルートテーブル > rtb-0b801ccd8abc1415b > Edit routes

Edit routes

Destination	Target	Status	Propagated
10.0.0.0/16	local	Active	いいえ
0.0.0.0/0	igw-09205aa69724c1fbb (exastro-gw)	Active	いいえ

キャンセル Preview **変更を保存**

5. 追加されていることを確認する

Updated routes for rtb-0b801ccd8abc1415b / exastro-rtb successfully

VPC > ルートテーブル > rtb-0b801ccd8abc1415b

rtb-0b801ccd8abc1415b / exastro-rtb

詳細 情報

ルートテーブル ID
rtb-0b801ccd8abc1415b

VPC
vpc-0e755a2dfc566d299 | exastro-VPC

メイン
はい

所有者 ID
[REDACTED]

明示的なサブネットの関連付け
-

ルート | サブネットの関連付け | Edge の関連付け | ルート伝播 | タグ

ルート (2)

ルートのフィルタリング

送信先	ターゲット	ステータス
10.0.0.0/16	local	Active
0.0.0.0/0	igw-09205aa69724c1fbb	Active

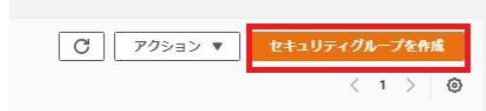
AWS設定手順書

5.セキュリティグループ

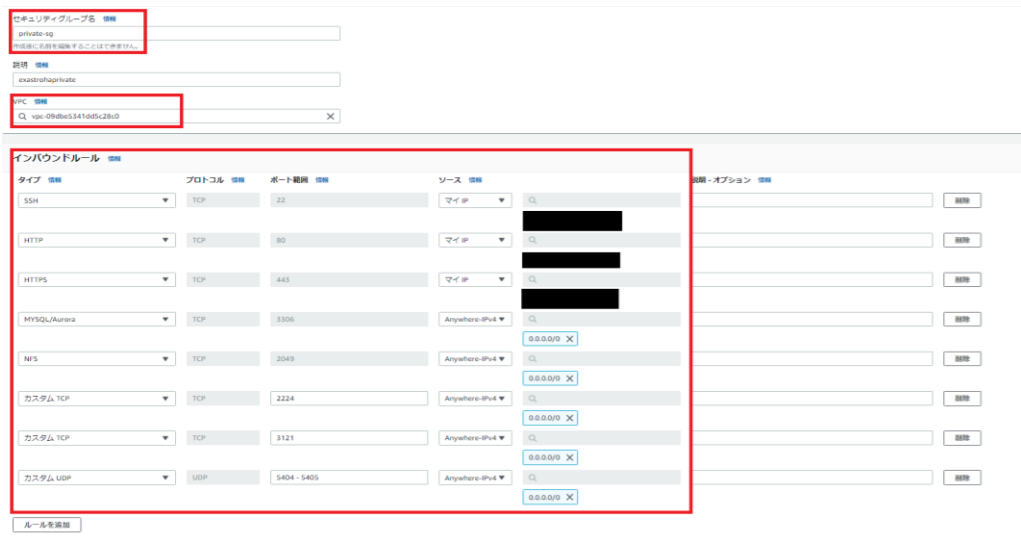
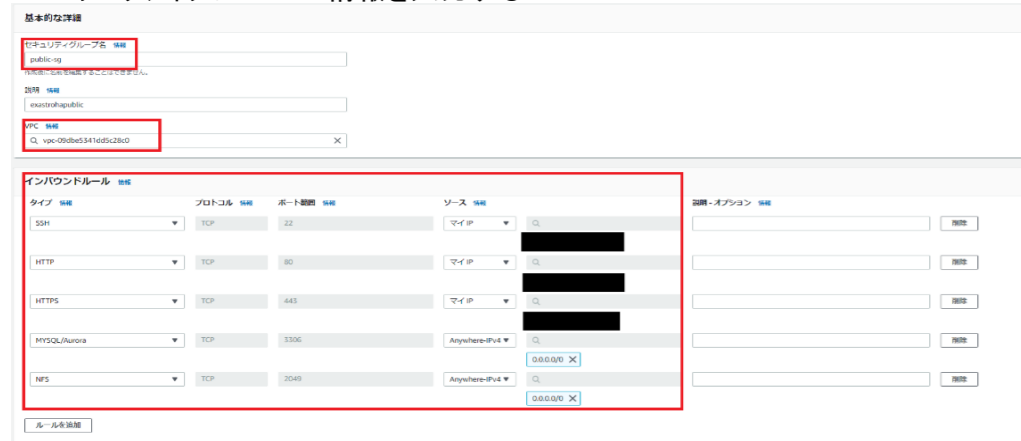
1.左側のメニューから「セキュリティグループ」をクリックする



2.「セキュリティグループを作成」をクリックする



3.セキュリティグループの情報を入力する



入力例)

セキュリティグループ名	public-sg
VPC	exastro-vpc

インバウンドルール

タイプ	プロトコル	ポート範囲	ソース
SSH	TCP	22	マイIP
HTTP	TCP	80	マイIP
HTTPS	TCP	443	マイIP

アウトバウンドルール

タイプ	プロトコル	ポート範囲	ソース
すべてのトラフィック	すべて	すべて	0.0.0.0/0

入力例)

セキュリティグループ名	private-sg
VPC	exastro-vpc

インバウンドルール

タイプ	プロトコル	ポート範囲	ソース
SSH	TCP	22	マイIP
HTTP	TCP	80	マイIP
HTTPS	TCP	443	マイIP
カスタムUDP	UDP	5404-5405	10.0.0.0/16 ※
カスタムTCP	TCP	3121	10.0.0.0/16 ※
カスタムTCP	TCP	2224	10.0.0.0/16 ※

※ VPCのNWアドレス

アウトバウンドルール

タイプ	プロトコル	ポート範囲	ソース
すべてのトラフィック	すべて	すべて	0.0.0.0/0

5.EFS、RDS用のセキュリティグループを作成する

The screenshot shows the AWS IAM console interface for creating security groups. It is divided into two sections: one for EFS and one for RDS. Each section has a 'Basic Information' tab and a 'Rules' tab. In the EFS section, the security group name is 'exastro-efs' and the VPC is 'exastro-vpc'. The inbound rule is for NFS (port 2049) from source 10.0.0.0/16. The outbound rule is for all traffic to destination 0.0.0.0/0. In the RDS section, the security group name is 'exastro-rds' and the VPC is 'exastro-vpc'. The inbound rule is for MySQL/Aurora (port 3306) from source 10.0.0.0/16. The outbound rule is for all traffic to destination 0.0.0.0/0.

入力例)

セキュリティグループ名	exastro-efs
VPC	exastro-vpc

インバウンドルール

タイプ	プロトコル	ポート範囲	ソース
NFS	TCP	2049	10.0.0.0/16 ※ ※ VPCのNWアドレス

アウトバウンドルール

タイプ	プロトコル	ポート範囲	ソース
すべてのトラフィック	すべて	すべて	0.0.0.0/0

入力例)

セキュリティグループ名	exastro-rds
VPC	exastro-vpc

インバウンドルール

タイプ	プロトコル	ポート範囲	ソース
MYSQL/Aurora	TCP	3306	10.0.0.0/16 ※ ※ VPCのNWアドレス

アウトバウンドルール

タイプ	プロトコル	ポート範囲	ソース
すべてのトラフィック	すべて	すべて	0.0.0.0/0

AWS設定手順書

6.IAM

1.左上の「サービス」から「IAM」をクリックする



2.左側のメニューの「ポリシー」をクリックする

- ダッシュボード
- ▼ アクセス管理
 - ユーザーグループ
 - ユーザー
 - ロール
 - ポリシー**
 - ID プロバイダー
 - アカウント設定

3.「ポリシーの作成」をクリックする



AWS設定手順書

4.JSONタブを開いて、ポリシーを入力する

ポリシーの作成

1 2 3

ポリシーにより、ユーザー、グループ、またはロールに割り当てることができる AWS アクセス権限が定義されます。ビジュアルエディタで JSON を使用してポリシーを作成または編集できます。 [詳細はこちら](#)

ビジュアルエディタ **JSON** 管理ポリシーのインポート

```
1 {
2   "Version": "2012-10-17",
3   "Statement": [
4     {
5       "Effect": "Allow",
6       "Action": [
7         "ec2:Describe*",
8         "ec2:DisassociateAddress",
9         "ec2:AssociateAddress"
10      ],
11      "Resource": [
12        "*"
13      ]
14    }
15  ]
16 }
```

セキュリティ: 0 エラー: 0 警告: 0 提案: 0

文字数: 150 / 6,144

キャンセル **次のステップ: タグ**

入力例)

ポリシー内容

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "ec2:Describe*",
        "ec2:DisassociateAddress",
        "ec2:AssociateAddress"
      ],
      "Resource": [
        "*"
      ]
    }
  ]
}
```

5.「次のステップ: 確認」をクリックする

キャンセル 戻る **次のステップ: 確認**

AWS設定手順書

6.ポリシー名を入力し、「ポリシーの作成」をクリックする

ポリシーの作成

1 2 3

ポリシーの確認

名前 exastro-policy

英数字と「+、@、_」を使用します。最大 128 文字。

説明

最大 1000 文字。英数字と「+、@、_」を使用します。

概要

Q フィルター

サービス	アクセスレベル	リソース	リクエスト条件
許可 (284 サービス中 1) 残りの 283 を表示			
EC2	制限 リスト、読み込み、書き込み	すべてのリソース	なし

タグ

キー

値

リソースに関連付けられたタグはありません。

入力例)

名前

exastro-policy

* 必須

キャンセル

戻る

ポリシーの作成

7.作成されたことを確認する

exastro-policy が作成されました。

ポリシーの作成

ポリシーアクション

ポリシーのフィルタ

Q exastro

ポリシー名	タイプ	次として使用	説明
exastro-policy	ユーザーによる管理	なし	

8.左側のメニューの「ロール」をクリックする

ダッシュボード

▼ アクセス管理

ユーザーグループ

ユーザー

ロール

ポリシー

ID プロバイダー

アカウント設定

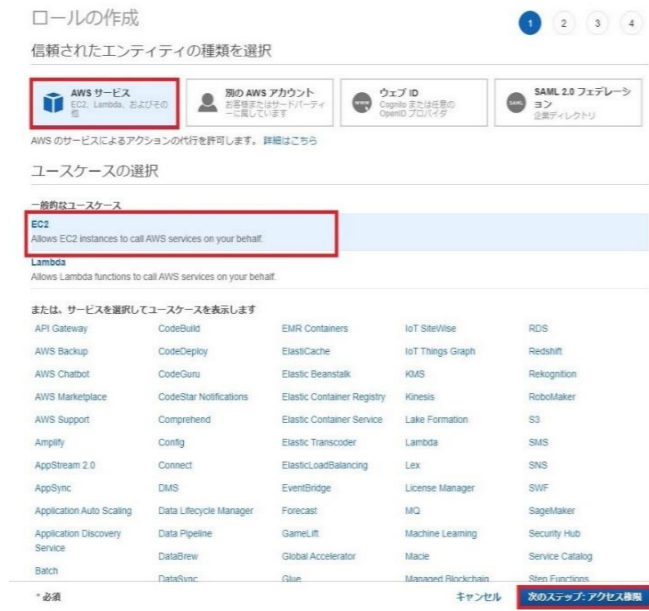
9.「ロールの作成」をクリックする

ロールの作成

ロールの削除

AWS設定手順書

10.「AWSサービス」の「EC2」を選択し「次のステップ:アクセス権限」をクリックする



11.先ほど作成したポリシーにチェックをし「次のステップ:タグ」をクリックする



AWS設定手順書

12.「次のステップ: 確認」をクリックする

ロールの作成

タグの追加 (オプション)

IAM タグは、ロールに追加できるキーと値のペアです。タグには、E メールアドレスなどのユーザー情報を含めるか、役職などの説明文とすることができます。タグを使用して、このロールのアクセスを整理、追跡、制御できます。詳細はこちら

キー	値 (オプション)	削除
<input type="text" value="新しいキーを追加"/>	<input type="text"/>	<input type="button" value="削除"/>

さらに 50 個のタグを追加できます。

キャンセル 戻る **次のステップ: 確認**

13 ロール名を入力し、先ほど選択したポリシーになっていることを確認し「ロールの作成」をクリックする

ロールの作成

確認

以下に必要な情報を指定してこのロールを見直し、作成してください。

ロール名:
英数字と「+、@、_」を使用します。最大 64 文字。

ロールの説明:
最大 1000 文字。英数字と「+、@、_」を使用します。

登録されたエンティティ: AWS のサービス、ec2.amazonaws.com

ポリシー:

アクセス権限の境界: アクセス権限の境界が設定されていません。

追加されたタグはありません。

* 必須

キャンセル 戻る **ロールの作成**

入力例)

ロール名	exastro-role
ポリシー	exastro-policy

14.ロールができていることを確認する

ロールの作成 ロールの削除

exastro

ロール名

exastro-role

AWS設定手順書

15.左側のメニューの「ユーザー」をクリックする

ダッシュボード

▼ アクセス管理

ユーザーグループ

ユーザー

ロール

ポリシー

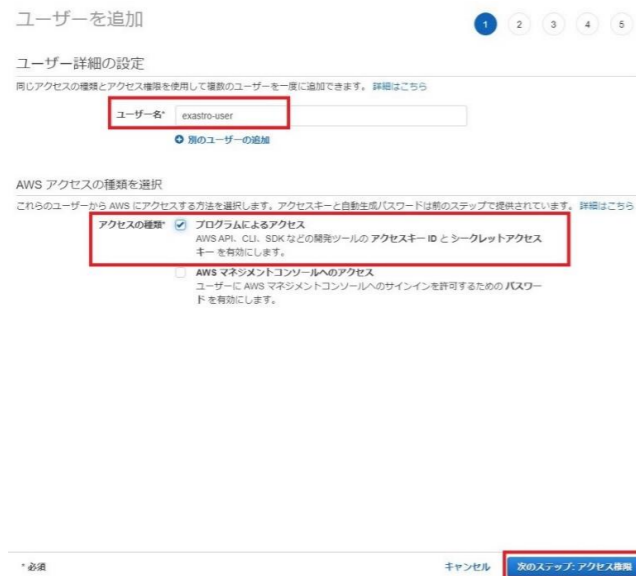
ID プロバイダー

アカウント設定

16.「ユーザーの追加」をクリックする



17.ユーザー名とプログラムによるアクセスにチェックを入れ、「次のステップ:アクセス権限」をクリックする



入力例)

ユーザー名	exastro-user
アクセスの種類	プログラムによるアクセス

AWS設定手順書

18.既存のポリシーを直接アタッチを選択し、先ほど作成したポリシーをチェックして「次のステップ:タグ」をクリックする

ユーザーを追加 1 2 3 4 5

▼ アクセス許可の設定

ユーザーをグループに追加 アクセス権限を既存のユーザーからコピー **既存のポリシーを直接アタッチ**

ポリシーの作成 🔄

ポリシーのフィルタ 1件の結果を表示中

ポリシー名	タイプ	次として使用
<input checked="" type="checkbox"/> exastro-policy	ユーザーによる管理	Permissions policy (1)

▼ アクセス権限の境界の設定

キャンセル 戻る **次のステップ:タグ**

19.「次のステップ:確認」をクリックする

キャンセル 戻る **次のステップ:確認**

AWS設定手順書

20.内容を確認し「ユーザーの作成」をクリックする

ユーザーを追加

1 2 3 4 5

確認

選択内容を確認します。ユーザーを作成した後で、自動生成パスワードとアクセスキーを確認してダウンロードできます。

ユーザー詳細

ユーザー名	exastro-user
AWS アクセスの種類	プログラムによるアクセス・アクセスキーを使用
アクセス権限の境界	アクセス権限の境界が設定されていません

アクセス権限の概要

次のポリシー例は、上記のユーザーにアタッチされます。

タイプ	名前
管理ポリシー	exastro-policy

タグ

追加されたタグはありません。

キャンセル

戻る

ユーザーの作成

21.csvのダウンロードを実施する ※サーバ設定時に使用する

ユーザーを追加

1 2 3 4 5

成功

以下に示すユーザーを正常に作成しました。ユーザーのセキュリティ認証情報を確認してダウンロードできます。AWS マネジメントコンソールへのサインイン手順を E メールでユーザーに送信することもできます。今回が、これらの認証情報をダウンロードできる最後の機会です。ただし、新しい認証情報はいつでも作成できます。

AWS マネジメントコンソールへのアクセス権を持つユーザーは「<https://943001175239.signin.aws.amazon.com/console>」でサインインできます

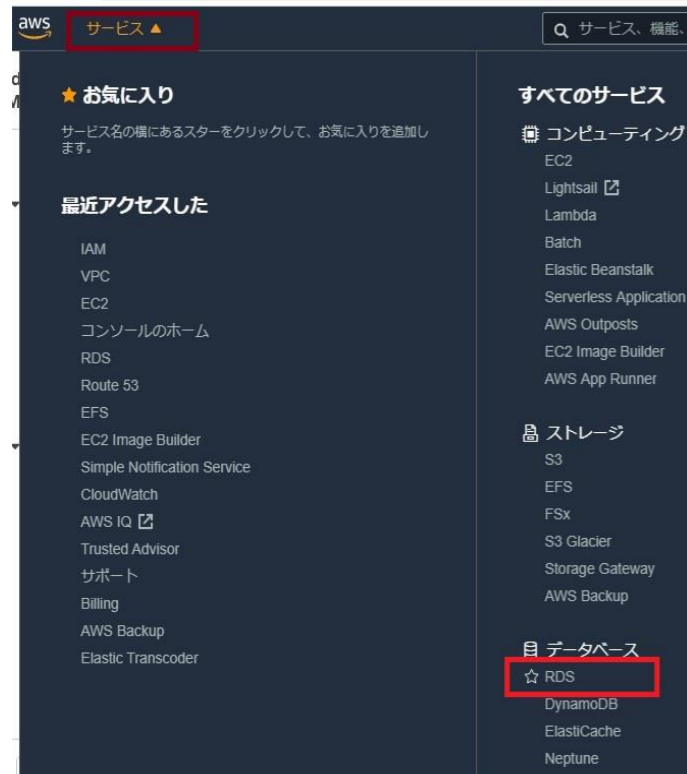
📄 .csv のダウンロード

ユーザー	アクセスキー ID	シークレットアクセスキー
▶ exastro-user	██████████	***** 表示

AWS設定手順書

7.RDS

1.左上のサービスからRDSをクリックする



2.左側のメニューから「サブネットグループ」をクリックする



3.「DBサブネットグループ作成」をクリックする



AWS設定手順書

4.DBサブネットグループの情報を入力し、「作成」をクリックする

RDS > Subnet groups > DBサブネットグループを作成

DBサブネットグループを作成

新しいサブネットグループを作成するには、名前と説明を入力し、既存のVPCを選択します。その後、そのVPCに関連するサブネットを追加できます。

サブネットグループの詳細

名前
サブネットグループの作成後に名前を変更することはできません。
exastrodb-subnet
1~255文字にする必要があります。英数字、スペース、ハイフン、アンダースコア、ピリオドを使用できます。

説明
rds-subnet

VPC
DBサブネットグループに使用するサブネットに対応するVPCを選択します。サブネットグループが作成された後、別のVPCを選択することはできません。
exastro-VPC (vpc-0e755a2dfc566d299)

サブネットを追加

アベイラビリティゾーン
追加するサブネットを含むアベイラビリティゾーンを選択します。
アベイラビリティゾーンを選択
ap-northeast-1a X ap-northeast-1c X

サブネット
追加するサブネットを選択します。リストには、選択したアベイラビリティゾーンのサブネットが含まれます。
サブネットを選択
subnet-097fba3cefd69f30 (10.0.192.0/21) X
subnet-061dbdda51760b21d (10.0.200.0/21) X

アベイラビリティゾーン	サブネットID	CIDRブロック
ap-northeast-1a	subnet-097fba3cefd69f30	10.0.192.0/21
ap-northeast-1c	subnet-061dbdda51760b21d	10.0.200.0/21

キャンセル 作成

入力例)

名前	exastrodb-subnet
説明	rds-subnet
VPC	exastro-VPC
アベイラビリティゾーン	ap-northeast-1a ap-northeast-1c
サブネット	subnet-097fba3cefd69f30 (10.0.192.0/21) subnet-061dbdda51760b21d (10.0.200.0/21)

5.左側のパラメータグループをクリックし、「パラメータグループの作成」をクリックする

パラメータグループアクション ▼ **パラメータグループの作成**

< 1 > ⓘ

6.パラメーターグループファミリーを選択し、グループ名を記載して「作成」をクリックする ※説明は任意です

RDS > パラメータグループ > パラメータグループの作成

パラメータグループの作成

パラメータグループの詳細
パラメータグループを作成するには、パラメータグループファミリーを選択し、パラメータグループの名前と説明を入力します

パラメータグループファミリー
このDBパラメータグループが適用されるDBファミリー
mariadb10.5

グループ名
DBパラメータグループの識別子
exastro-param

説明
DBパラメータグループの説明
exastrodb param

キャンセル 作成

入力例)

パラメーターグループファミリー	mariadb10.5
グループ名	exastro-param

AWS設定手順書

7.作成されたことを確認する



8.作成したパラメータグループをクリックして、パラメーターの設定を実施する ※一つ設定するにつき「変更の保存」を必ずクリックすること



AWS設定手順書

パラメータ

query_cache_type

名前	値	許可された値	変更可能	送信元	適用タイプ	データ型	説明
query_cache_type	1	0-2	true	engine-default	static	integer	For query results either don't cache (-OFF), cache except for NO_CACHE (-ON), or only CACHE (-DEMAND)

パラメータ

tmp_table_size

名前	値	許可された値	変更可能	送信元	適用タイプ	データ型	説明
tmp_table_size	67108864	0-4294967295	true	engine-default	dynamic	integer	If an in-memory temporary table exceeds the limit, MySQL automatically converts it to an on-disk MyISAM table. Increased value can improve perf for many advanced GROUP BY queries.

パラメータ

mrr_buffer_size

名前	値	許可された値	変更可能	送信元	適用タイプ	データ型	説明
mrr_buffer_size	67108864	8192-2147483647	true	engine-default	dynamic	integer	Size of buffer to use when using MRR with range access.

パラメータ

max_heap_table_size

名前	値	許可された値	変更可能	送信元	適用タイプ	データ型	説明
max_heap_table_size	67108864	16384-1844674407370954752	true	engine-default	dynamic	integer	Maximum size to which MEMORY tables are allowed to grow.

パラメータ

character_set_server

名前	値	許可された値	変更可能	送信元	適用タイプ	データ型	説明
character_set_server	utf8	big5, dec8, cp850, hp8, ko8r, latin1, latin2, swe7, ascii, ujis, sjis, hebrew, tis620, euckr, ko8rs, gb2312, greek, cp1250, gbk, latin5, armSCII8, utf8, uc2, cp866, koi8r2, macce, macroman, cp932, latin7, utf8mb4, cp1251, utf16, cp1256, cp1257, utf32, binary, geostd8, cp932, eucjms	true	engine-default	dynamic	string	The server's default character set.

パラメータ

max_connections

名前	値	許可された値	変更可能	送信元	適用タイプ	データ型	説明
extra_max_connections		1-100000	false	engine-default	dynamic	integer	The number of connections on the extra_port.
max_connections	256	1-100000	true	system	dynamic	integer	The number of simultaneous client connections allowed.

パラメーター	値
explicit_defaults_for_timestamp	1
tx_isolation	READ-COMMITTED
innodb_buffer_pool_size	536870912
innodb_log_buffer_size	67108864
min_examined_row_limit	100
join_buffer_size	134217728
query_cache_size	536870912
query_cache_type	1
max_heap_table_size	67108864
tmp_table_size	67108864
mrr_buffer_size	67108864
max_connections	256
character_set_server	utf8

9.保存後、左側メニューのダッシュボードから「データベースの作成」をクリックする



10.RDSを作成する

RDS > データベースの作成

データベースの作成

データベース作成方法を選択 [情報](#)

標準作成
可用性、セキュリティ、バックアップ、メンテナンスといったすべての設定オプションを設定します。

簡潔作成
推奨されるベストプラクティス設定を使用します。一部の設定オプションは、データベースの作成後に変更できます。

エンジンのオプション

エンジンのタイプ [情報](#)

Amazon Aurora

MySQL

MariaDB

PostgreSQL

Oracle

Microsoft SQL Server

バージョン
MariaDB 10.5.8

テンプレート
お客様のユースケースに合わせてサンプルテンプレートを選択します。

本番稼働用
高い可用性と、高速で安定したパフォーマンスのために、デフォルト値を使用します。

開発/テスト
このインスタンスは本番稼働環境ではない開発で使用します。

無料利用枠
RDS 無料利用枠を利用すると、新しいアプリケーションの開発、既存のアプリケーションのテスト、Amazon RDS の実装経験の蓄積が可能です。 [情報](#)

設定

DB インスタンス識別子 [情報](#)
DB インスタンスの名前を入力します。この名前は、AWS アカウントが現在の AWS リージョンで所有しているすべての DB インスタンスにおいて一意である必要があります。

exastro-rds

DB インスタンス識別子は大文字と小文字を区別しませんが、すべて小文字で保存されます (例: "mydbinstance")。制約として、使用できるのは 1~60 文字以内で英数字またはハイフンのみです。1 文字目は英数字でなければなりません。また、ハイフンを連続で 2 つ使ったり、最後の文字をハイフンにしたりすることはできません。

▼ 認証情報の設定

マスターユーザー名 [情報](#)
DB インスタンスのマスターユーザーのログイン ID を入力します。

admin

1~16 文字の英数字。1 文字目は文字である必要があります

パスワードの自動生成
Amazon RDS がパスワードを生成するか、お客様が自分でパスワードを設定することができます

マスターパスワード [情報](#)

制約事項: 表示可能な ASCII 文字で 8 文字以上で入力してください。次の文字を含めることはできません: / (スラッシュ)、@ (単一引用符)、* (二重引用符)、および @ (アットマーク)。

パスワードを確認 [情報](#)

DB インスタンスサイズ

DB インスタンスクラス [情報](#)
処理能力とメモリの要件に合った DB インスタンスクラスを選択します。以下の DB インスタンスクラスオプションは、上記で選択したエンジンでサポートされているものに制限されます。

標準クラス (m クラスを含む)

メモリ最適化クラス (r クラスと x クラスを含む)

パースト可能クラス (t クラスを含む)

db.m6g.large
2 vCPUs 8 GB RAM Network: 4,750 Mbps

以前の世代のクラスを含める

※入力例の記載が無いものはデフォルト値です(パスワードは任意です)

入力例)

作成方法	標準作成
エンジンのオプション	MariaDB
バージョン	MariaDB 10.5.8
テンプレート	本番稼働用
DBインスタンス識別子	exastro-rds
マスターユーザー名	admin
可用性と耐久性	スタンバイインスタンスを作成する
VPC	exastro-VPC
サブネットグループ	exastrodb-subnet
パブリックアクセス可能	なし
VPCセキュリティグループ	既存の選択 exastro-rds
データベースポート	3306
DBパラメータグループ	exastro-param
オプショングループ	defaultmariadb-10.5
バックアップ	自動バックアップの有効化
モニタリング	拡張モニタリングの有効化
ログのエクスポート	監視ログ エラーログ 全般ログ スロークエリログ
選択ウィンドウ	土曜日 03:00 0.5時間

AWS設定手順書

ストレージ

ストレージタイプ [情報](#)
プロビジョンド IOPS (SSD)

ストレージ割り当て
100 GiB
最小: 100 GiB, 最大: 65,536 GiB

プロビジョンド IOPS [情報](#)
3000 IOPS
Minimum: 1,000 IOPS, Maximum: 80,000 IOPS

ストレージの自動スケーリング [情報](#)
アプリケーションのニーズに基づいて、データベースのストレージに対する動的なスケーリングのサポートを提供します。

ストレージの自動スケーリングを有効にする
この機能を有効にすると、指定したしきい値を超えた場合にストレージを増やすことができます。

最大ストレージしきい値 [情報](#)
データベースが指定されたしきい値に自動スケールされると、料金が適用されます。
1000 GiB
最小: 101 GiB, 最大: 65,536 GiB

可用性と耐久性

マルチ AZ 配置 [情報](#)

スタンバイインスタンスを作成する (本稼働環境向けに推奨)
データの可用性を確保し、RTO を短縮し、システムバックアップの間のレイテンシーの急上昇を最小限に抑えるために、別のアベイラビリティゾーン (AZ) にスタンバイを作成します。

スタンバイインスタンスを作成しないでください

AWS設定手順書

接続

Virtual Private Cloud (VPC) 情報
この EC2 インスタンスの接続ネットワークを選択する VPC。
exastro-VPC (vpc-0e755a2dfc566d299)

データベースの作成後に、VPC の選択を変更することはできません。

サブネットグループ 情報
選択した VPC で DB インスタンスが使用できるサブネットと IP 範囲を定義する DB サブネットグループ。
exastrodb-subnet

パブリックアクセス可能 情報
 あり
VPC 外部の Amazon EC2 インスタンスとデバイスが異なる種類のデータベースに接続できます。データベースに接続できる VPC 内の EC2 インスタンスおよびデバイスから特定のポート/トラフィックが許可されていることを確認します (セキュリティグループはパブリックにアクセス可能なデータベースに必要です)。
 なし
RDS はパブリック IP アドレスをデータベースに割り当てません。VPC 内部の Amazon EC2 インスタンスとデバイスのみを異なる種類のデータベースに接続できます。

VPC セキュリティグループ
RDS セキュリティグループを 1 つ以上選択し、データベースへのアクセスを許可します。セキュリティグループのルールで EC2 インスタンスと VPC 内のデバイスからの特定のポート/トラフィックが許可されていることを確認します (セキュリティグループはパブリックにアクセス可能なデータベースに必要です)。
 既存の選択
既存の VPC セキュリティグループの選択
 新規作成
新しい VPC セキュリティグループの作成

既存の VPC セキュリティグループ
VPC セキュリティグループを選択します
exastro-rds

追加の接続設定
データベースポート 情報
データベースがアプリケーションの接続に使用する TCP/IP ポート。
3306

追加設定

データベースオプション、暗号化、バックアップ、バックアップが無効、Performance Insights が有効、監視モニタリングが有効、メンテナンス、CloudWatch Logs、削除保護 が有効

データベースの選択肢

最初のデータベース名 情報
データベース名を指定しないと、Amazon RDS はデータベースを作成しません。

DB パラメータグループ 情報
exastro-param

オプショングループ 情報
defaultmariadb-10.5

バックアップ
データベースのポイントインタイムスナップショットを作成します。
 自動バックアップの有効化
データベースのポイントインタイムスナップショットを作成します。

バックアップ保持期間 情報
このインスタンスの自動バックアップを RDS が保存する日数を選択します。
7 日間

バックアップウィンドウ 情報
Amazon RDS によって作成されるデータベースの自動バックアップの期間を選択します。
 選択ウィンドウ
 設定なし

開始時間
05 : 00 UTC 期間 0.5 時間

スナップショットにタグをコピー

暗号化
 暗号を有効化
選択した対象のインスタンスを暗号化します。マスターキー ID とエイリアスは、Key Management Service (KMS) コンソールを使用して作成した後に、リストに表示されます。 情報

AWS設定手順書

The screenshot shows the AWS RDS console configuration page for Performance Insights, Monitoring, and Maintenance. The 'Monitoring' section is highlighted with a red box, showing the 'Enable Monitoring' checkbox checked, a retention period of 7 days, and a monitoring interval of 60 seconds. The 'Logs Export' section is also highlighted, showing checkboxes for 'Audit Log', 'Error Log', 'Slow Query Log', and 'Slow Query Log' all checked. The 'Maintenance' section shows the 'Enable Automatic Minor Version Upgrade' checkbox unchecked, and the 'Maintenance Window' section is highlighted with a red box, showing a start time of 03:00 UTC and a duration of 0.5 hours.

11.RDSが作成され、利用可能になることを確認する ※少し時間がかかります

The screenshot shows the AWS RDS console instance list. The instance 'esatdb-rs' is highlighted with a red box, and its status is '利用可能' (Available). The table below shows the instance details:

ID	DB 識別子	ロール	エンジン	リージョンとAZ	サイズ	ステータス	CPU	開始のアクティビティ	メンテナンス	VPC	マルチAZ
esatdb-rs	esatdb-rs	インスタンス	MySQL	ap-northeast-1	db.m5g.large	利用可能	100%	0%	vpk-0a75a2d8-5664299		あり

AWS設定手順書

8.EFS

1.左上のサービスからEFSをクリックする



2.「ファイルシステムの作成」をクリックする



3.名前を入力し、作成したVPCを選択、リージョンを選択し「カスタマイズ」をクリックする

ファイルシステムの作成

サービスの推奨設定を使用して EFS ファイルシステムを作成します。 [詳細はこちら](#)

名前 - オプション
ファイルシステムに名前を付けます。

名前は 256 文字より長くすることはできません。使用できるのは文字、数字、および一部の記号 (+ = . _ : /) のみです。

Virtual Private Cloud (VPC)
EC2 インスタンスをファイルシステムに接続する VPC を選択します。 [詳細はこちら](#)

exastro-VPC

可用性と耐久性
リージョンのストレージクラスを使用してファイルシステムを作成するには、リージョンを選択します (推奨)。One Zone ストレージクラスを使用してファイルシステムを作成するには、One Zone を選択します。 [詳細はこちら](#)

リージョン
複数の AZ にデータを冗長的に保存

1ゾーン
単一の AZ 内にデータを冗長的に保存

キャンセル

入力例)

名前-オプション	exastro-efs
VPC	exastro-VPC
可用性と耐久性	リージョン

AWS設定手順書

4.自動バックアップを選択し、ネットワークは先ほどのVPCから選択し、EFS用に作成したセキュリティグループを追加し「次へ」をクリックする

入力例)

ネットワーク	exastro-VPC
セキュリティグループ	exastro-efs

5.「次へ」をクリックする

AWS設定手順書

6.内容を確認し「作成」をクリックする

確認して作成する

ステップ 1: ファイルシステムの設定 編集

ファイルシステム

フィールド	値	編集可能ですか?
名前	exastro-efs	あり
パフォーマンスモード	汎用	なし
スループットモード	バースト	あり
圧縮	あり	なし
KMS キー ID	-	なし
ライフサイクルポリシー	最後のアクセスから 30 日	あり
自動バックアップ	あり	あり
VPC ID	vpc-0c755a2b3c566d299 (exastro-VPC)	あり
Availability Zone	リージョン	なし

タグ

タグキー ▼ タグ値 ▼

このリソースに適用可能なタグはありません。

タグを管理

ステップ 2: ネットワークアクセス

編集

マウントターゲット

Availability Zone	サブネット	IP アドレス	セキュリティグループ
ap-northeast-1a	subnet-097ba3c0f8f9f30	-	sg-0488a06043c1211
ap-northeast-1c	subnet-0c1b0a5d1792b27d	-	sg-0488a06043c1211

ステップ 3: ファイルシステムポリシー

編集

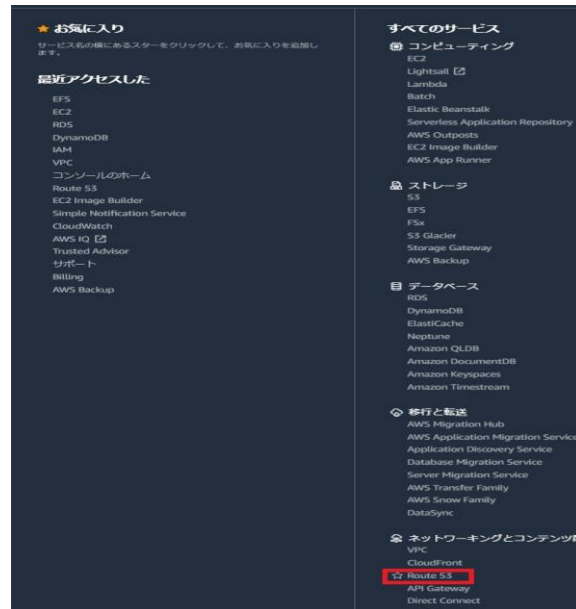
▼ ファイルシステムポリシー

キャンセル 戻る 作成

AWS設定手順書

9.Route53

1.左上のサービスからRoute53をクリックする



2.「ホストゾーンの作成」をクリックする



AWS設定手順書

3.ドメイン名を入力し、プライベートホストゾーンにチェックし、東京リージョンとVPCを選択し「ホストゾーンの作成」をクリックする

ホストゾーン設定

ドメイン名 Info
これは、トラフィックをルーティングするドメインの名前です。
exastro-aws.local

説明 - オプション Info
この場で、同じ名前のホストゾーンを区別できます。
ホストゾーンは次の目的で使用されます。

タイプ Info
このタイプは、インターネットまたは Amazon VPC でトラフィックをルーティングするかどうかを示します。
 パブリックホストゾーン
 プライベートホストゾーン

ホストゾーンに関連付ける VPC Info
このホストゾーンを参照して 1 つ以上の VPC の DNS クエリを解決するには、当該の VPC を選択します。別の AWS アカウントで作成された VPC をホストゾーンに関連付けるには、AWS CLI などのプログラム的な方法を用いる必要があります。

プライベートホストゾーンに関連付ける各 VPC に対して、Amazon VPC 設定 enableDnsHostnames および enableDnsSupport を true に設定する必要があります。

リージョン Info: アジアパシフィック (東京) [ap-nor-1]
VPC ID Info: vpc-0e755a2dfc566d299

タグ Info
このリソースに関連付けられたタグがありません。

キャンセル **ホストゾーンの作成**

入力例)

ドメイン名	exastro-aws.local
タイプ	プライベートホストゾーン
リージョン	アジアパシフィック(東京)[ap-northeast-1]
VPC ID	vpc-0e755a2dfc566d299

4.「レコードの作成」をクリックする

レコード (2) | ホストゾーンのタグ (0)

レコード (2) Info
Automatic モードは最新のファイルが確実に最新化された現在の構成動作です。モードを変更するには、設定に移動します。

レコードを削除 | ソーンファイルをインポート | **レコードを作成**

レコード名	タイプ	ルート	重み	ルーティングポリシー
exastro-aws.local	NS	シンプル	-	ns-1536.awsdns-00.co.uk. ns-0.awsdns-00.com. ns-1024.awsdns-00.org. ns-512.awsdns-00.net.
exastro-aws.local	SOA	シンプル	-	ns-1536.awsdns-00.co.uk. awsdns-hostmaster.amazon.com. 1 7200 900 1209600 86400

AWS設定手順書

5.レコード名、CNAMEを選択しRDSのエンドポイントを値に入力し、「レコードを作成」をクリックする

Route 53 > ホストゾーン > exastro-aws.local > レコードを作成

レコードのクイック作成 Info ウィザードに切り替える 別のレコードを追加

▼ レコード 1 削除

レコード名 Info .exastro-aws.local 有効な文字: a-z, 0-9, ! * # \$ % & ' () * +, - / : ; < = > ? @ [\] ^ _ ' { } . ~

レコードタイプ Info CNAME - 別のドメイン名および一部の AWS ...

値 Info エイリアス 複数の値を個別の行に入力します。

TTL (秒) Info 推奨値: 60~172800 (2 日間)

ルーティングポリシー Info

入力例)

レコード名	rds
レコードタイプ	CNAME
値	RDSのエンドポイント

6.レコード名、CNAMEを選択しEFSのエンドポイントを値に入力し、「レコードを作成」をクリックする

Route 53 > ホストゾーン > exastro-aws.local > レコードを作成

レコードのクイック作成 Info ウィザードに切り替える 別のレコードを追加

▼ レコード 1 削除

レコード名 Info .exastro-aws.local 有効な文字: a-z, 0-9, ! * # \$ % & ' () * +, - / : ; < = > ? @ [\] ^ _ ' { } . ~

レコードタイプ Info CNAME - 別のドメイン名および一部の AWS ...

値 Info エイリアス 複数の値を個別の行に入力します。

TTL (秒) Info 推奨値: 60~172800 (2 日間)

ルーティングポリシー Info

入力例)

レコード名	efs
レコードタイプ	CNAME
値	EFSのエンドポイント

AWS設定手順書

10.EC2

1.左上のサービスからEC2をクリックする



2.「インスタンスを起動」をクリックする



3.OSを選択する

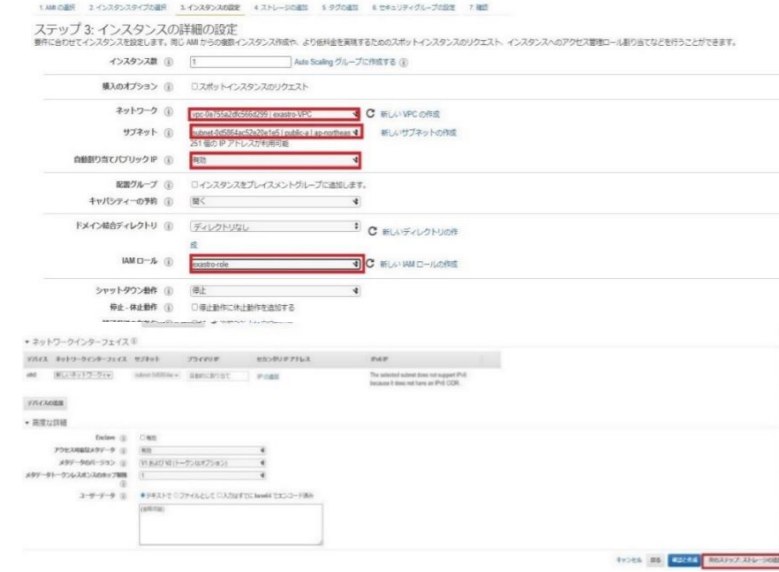


AWS設定手順書

4. サイズを選択する ※Exastrolに必要な最低限のスペックにて実施



5. 作成したVPC、対象のサブネット、自動割り当てパブリックIP、作成したIAMロールを選択し「次のステップ: ストレージの追加」をクリックする



6. 「次のステップ: タグの追加」をクリックする



AWS設定手順書

7.「次のステップ: セキュリティグループ」をクリックする



8.作成したセキュリティグループを選択し、「確認と作成」をクリックする

ステップ 6: セキュリティグループの設定
 セキュリティグループは、インスタンスのトラフィックを制御するファイアウォールのルールセットです。このページで、特定のトラフィックに対してインスタンスへの到達を許可するルールを追加できます。たとえば、ウェブサーバーをセットアップして、インターネットトラフィックにインスタンスへの到達を許可する場合、HTTP および HTTPS ポートに無制限のアクセス権限を与えます。新しいセキュリティグループを作成するか、次の既存のセキュリティグループから選択することができます。詳細はこちら。



9.確認し、問題なければ「起動」をクリックする

※publicサブネットの各リージョンに1台ずつ、privateサブネットの各リージョンに2台ずつ作成する



入力例)

webap-sv-a	
ネットワーク	exastro-VPC
サブネット	public-c
自動割り当てパブリックIP	有効
IAMロール	exastro-role
セキュリティグループ	public-sg

入力例)

webap-sv-c	
ネットワーク	exastro-VPC
サブネット	public-c
自動割り当てパブリックIP	有効
IAMロール	exastro-role
セキュリティグループ	public-sg

入力例)

backyard-sv-a	
ネットワーク	exastro-VPC
サブネット	private-a
自動割り当てパブリックIP	有効
IAMロール	exastro-role
セキュリティグループ	private-sg

入力例)

backyard-sv-c	
ネットワーク	exastro-VPC
サブネット	private-c
自動割り当てパブリックIP	有効
IAMロール	exastro-role
セキュリティグループ	private-sg

入力例)

ansible-sv-a	
ネットワーク	exastro-VPC
サブネット	private-c
自動割り当てパブリックIP	有効
IAMロール	exastro-role
セキュリティグループ	private-sg

入力例)

ansible-sv-c	
ネットワーク	exastro-VPC
サブネット	private-c
自動割り当てパブリックIP	有効
IAMロール	exastro-role
セキュリティグループ	private-sg

AWS設定手順書

10.SSHに必要な鍵を作成する場合はこちらで作成する

既存のキーペアを選択するか、新しいキーペアを作成します。 ×

キーペアは、AWS が保存するパブリックキーとユーザーが保存するプライベートキーファイルで構成されます。組み合わせて使用することで、インスタンスに安全に接続できます。Windows AMI の場合、プライベートキーファイルは、インスタンスへのログインに使用される/パスワードを取得するために必要です。Linux AMI の場合、プライベートキーファイルを使用してインスタンスに SSH で安全に接続できます。

注 選択したキーペアは、このインスタンスに対して権限がある一連のキーに追加されます。「パブリック AMI から既存のキーペアを削除する」の詳細情報をご覧ください。

新しいキーペアの作成

キーペア名
exastro-key

キーペアのダウンロード

… 続行するには、事前にプライベートキーファイル (*.pem ファイル) をダウンロードする必要があります。それを、安全でアクセス可能な場所に保存します。一度作成されたファイルは再度ダウンロードすることはできません。

キャンセル インスタンスの作成

11.作成後アクション→ネットワーキング→ソース宛先/変更をクリックする

The screenshot shows the AWS Management Console interface for instance management. A table lists several instances in a 'running' state. An 'Actions' menu is open for one of the instances, showing options like 'Connect', 'Show details', and 'Networking'. The 'Networking' option is highlighted with a red box, and its sub-menu is visible, with 'Source/destination checks' also highlighted with a red box.

Name	インスタンス ID	インスタンスの状態	インスタンス...	ステータスチェック	アラームの状態	アベイラビリティ...
backyard-sv-c	i-0c5a71d5c191888bd	実行中	t2.medium	初期化して	ネットワークインターフェイスをアタッチ	
ansible-sv-c	i-0c6bd96647b05e50a	実行中	t2.medium	2/2 のチェッ	ネットワークインターフェイスをデタッチ	
backyard-sv-a	i-06bbb332458eff146	実行中	t2.medium	2/2 のチェッ	ソース/宛先チェックを変更	
ansible-sv-a	i-0efa4a53fa42b0fbb	実行中	t2.medium	2/2 のチェッ	Elastic IP アドレスの関連付けの解除	

12.停止にチェックを入れ、「保存」をクリックする ※privateサブネットに属しているEC2全てに対して実施する

The screenshot shows the 'Source/destination checks' configuration page in the AWS console. The 'Stop' checkbox is checked and highlighted with a red box. A warning message states that for NAT instances, stopping source/destination checks is required. At the bottom, the 'Save' button is highlighted with a red box.

送信元/送信先チェック

各 EC2 インスタンスはデフォルトで送信元と送信先のチェックを実行します。インスタンスは、送受信するすべてのトラフィックの送信元または送信先である必要があります。

インスタンス ID
i-064df3b317782f2d (ta-ha-sv02)

ネットワークインターフェイス
eni-0b0e4030a10ebda1b

送信元/送信先チェック中

停止

これが NAT インスタンスの場合は、送信元/送信先のチェックを停止する必要があります。NAT インスタンスは、送信元または送信先がそのものではない場合、トラフィックを送受信できる必要があります。

AWS CLI コマンド

```
aws ec2 modify-instance-attribute --instance-id i-064df3b317782f2d --no-source-destination-check
```

キャンセル 保存

AWS設定手順書

11.Elastic IP

1.左側のメニューのElastic IPをクリックする



2.「IPアドレスの割り当て」をクリックする

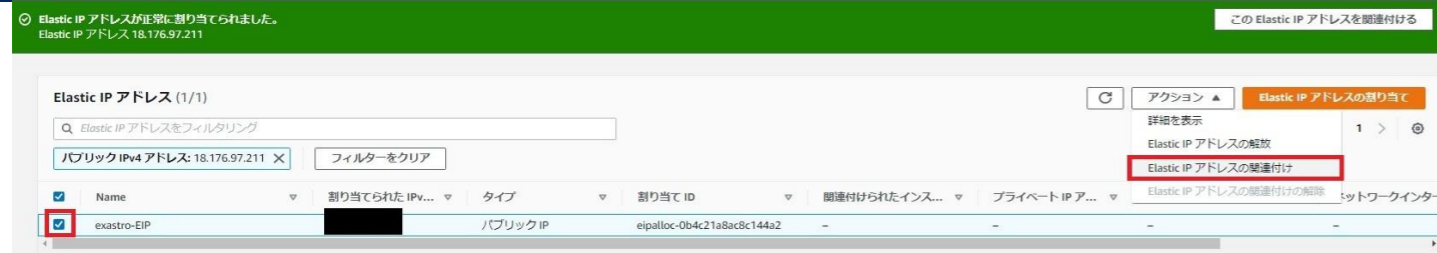


3.ネットワークボーダーグループが東京リージョンになっていることを確認し、「割り当て」をクリックする



4.IPが生成されたらアクション→Elastic IPアドレスの関連付けをクリックする

AWS設定手順書



5. インスタンスを選択し (ansible-sv-a で動作しているインスタンス)、プライベート IP アドレスを入力し (候補に出るのでそれを選択) 「関連付ける」をクリックする



AWS設定手順書

12.LoadBalancer

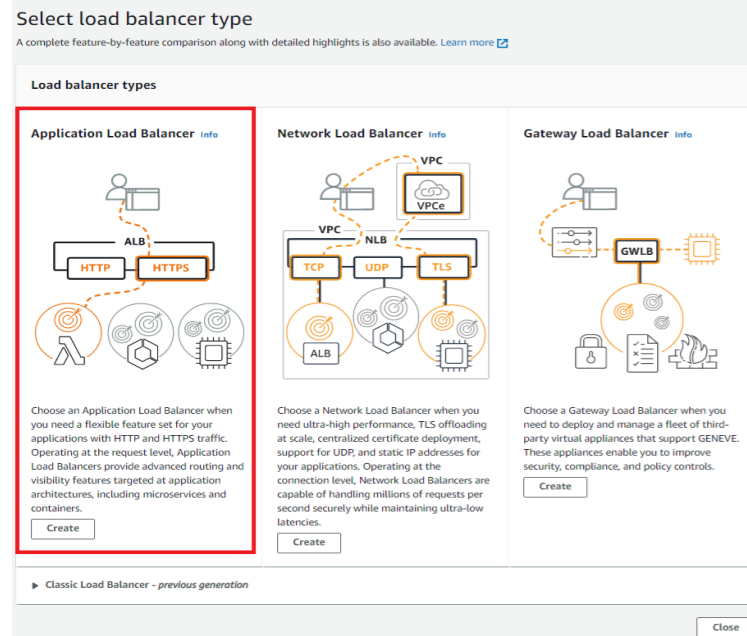
1.左のメニューの「ロードバランサー」をクリックする



2.「ロードバランサーの作成」をクリックする



3.「Application Load Balancer」を選択する ※この手順書ではApplication Load Balancerでの設定になります



AWS設定手順書

4.各種設定内容を入力する

Create Application Load Balancer Info

The Application Load Balancer distributes incoming HTTP and HTTPS traffic across multiple Targets such as Amazon EC2 instances, microservices, and containers, based on request attributes. When the load balancer receives a connection request, it evaluates the listener rules in priority order to determine which rule to apply, and if applicable, it selects a target from the target group for the rule action.

▶ How Application Load Balancers work

Basic configuration

Load balancer name Must be at least 3 characters within your AWS account and cannot be changed after the load balancer is created.

exastro-lb

A maximum of 32 alphanumeric characters including hyphens are allowed, but the name must not begin or end with a hyphen.

Scheme Info

Internet-facing An internet-facing load balancer routes requests from clients over the internet to targets. Requires a public subnet. [Learn more](#)

Internal An internal load balancer routes requests from clients to targets using private IP addresses.

IP address type Info

IPv4 Recommended for internal load balancers.

Dualstack Includes IPv4 and IPv6 addresses.

Network mapping Info

The load balancer routes traffic to targets in the selected subnets, and in accordance with your IP address settings.

VPC Info

Select the virtual private cloud (VPC) for your targets. Only VPCs with an internet gateway are enabled for selection. The selected VPC cannot be changed after the load balancer is created. To select a different VPC for your targets, click [View available VPCs](#).

exastro-VPC
vpc-09bbe5341dd5c28cd
Pub: 10.0.0.0/16

Mappings Info

Select at least two Availability Zones and one subnet per zone. The load balancer routes traffic to targets in these Availability Zones only. Availability Zones that are not supported by the load balancer or the VPC are not available for selection. Subnets cannot be removed after the load balancer is created, but additional subnets can be added.

ap-northeast-1a

Subnet
subnet-031f186b7c1c20f5 public-a

IPv4 settings
Assigned by AWS

ap-northeast-1c

Subnet
subnet-02c4a6b47d8f190 public-c

IPv4 settings
Assigned by AWS

5.途中まで入力したら「Create Target group」をクリックする

Security groups Info

A security group is a set of firewall rules that control the traffic to your load balancer.

Security groups

Select security groups

Create new security group

public-sg sg-0c54a54ade368bd53
VPC: vpc-09bbe5341dd5c28cd

Listeners and routing Info

A listener is a process that checks for connection requests, using the protocol and port you configure. Traffic received by the listener is then routed per your specification. You can specify multiple rules and multiple certificates per listener after the load balancer is created.

▼ Listener HTTP:80 Remove

Protocol: HTTP Port: 80 Default action Info

1-65535 Forward to: Select a target group

Create target group

Add listener

AWS設定手順書

6.Target groupの設定をする

※HTTPで設定をしておりますが、HTTPSでの設定も可能です

Basic configuration
Settings in this section cannot be changed after the target group is created.

Choose a target type

- Listeners**
Supports load balancing to instances within a specific VPC.
- IP addresses
Supports load balancing to VPC and on-premises resources.
Facilitates routing to multiple IP addresses and network interfaces on the same instance.
Offers flexibility with multi-region based architectures, simplifying cross-application communication.
- Lambda function
Facilitates routing to a single Lambda function.
Accessible to Application Load Balancers only.
- Application Load Balancer
Offers the flexibility for a Network Load Balancer to accept and route TCP requests within a specific VPC.
Facilitates using stateful addresses and protocols with an Application Load Balancer.

Target group name
awscli-04
A maximum of 25 alphanumeric characters including hyphens are allowed, but the name must not begin or end with a hyphen.

Protocol: HTTP | Port: 80

VPC
awscli-vpc
The VPC that you want to include in the target group.

Send requests to targets using
 HTTP
Send requests to targets using HTTP/1.1. Supported when the request protocol is HTTP/1.1 or HTTPS.
 HTTPS
Send requests to targets using HTTPS. Supported when the request protocol is HTTPS or gRPC, but gRPC specific headers are not available.
 gRPC
Send requests to targets using gRPC. Supported when the request protocol is gRPC.

Health checks
The associated load balancer periodically sends requests, per the settings below, to the registered targets to test their status.

Health check protocol
HTTP

Health check path
Use the default path with a * to allow the root, or specify a custom path if configured.
/common/common_auth.php?login&grp=&no=

Advanced health check settings

7.対象のインスタンスを選択する

Register targets
This is an optional step to create a target group. However, to ensure that your load balancer routes traffic to this target group you must register your targets.

Available instances (2/17)
Filter instances by priority or status

Instance ID	Name	State	Security groups	Zone	Subnet ID
i-01a73d86b0b27f3	webapp-a	running	sg-9a18783c	ap-northeast-1a	subnet-026a6f7d87f80
i-081a2895d78a3f	webapp-a	running	sg-9a18783c	ap-northeast-1a	subnet-026a6f7d87f80

2 selected
Points for the selected instances
From the available points to the selected instances:
80
100000 instances include points with a maximum of 100000 instances include points with a maximum of 100000
Include as pending below

Review targets

Targets (0)
Filter instances by priority or status

Instance ID	Name	Port	State	Security groups	Zone	Subnet ID
-------------	------	------	-------	-----------------	------	-----------

No instances added yet
Specify instance IDs, or leave this group empty if you prefer to add targets later.

0 pending

8.先ほど入力していたロードバランサーの画面に戻り作成したTarget groupを設定し作成する

Listeners and routing
A listener is a process that checks for connection requests, using the protocol and port you configure. Traffic received by the listener is then routed per your configuration. You can specify multiple listeners and routes to your target group when the load balancer is created.

Listener: HTTP:80

Protocol: HTTP | Port: 80 | Default action: Target Group: awscli-04

Tags - optional
Customizing tags for your load balancer. These enable you to categorize your AWS resources so you can more easily manage them. The tag is required, but 'Value' is optional. For example, you can use 'tag:Environment:Prod' or 'tag:Environment:Prod:Prod'.

Summary

Basic configuration Instance ID: awscli-04 • Instance facing • IPvt	Security groups • sg-9a18783c	Network mapping VPC: awscli-vpc • sg-9a18783c • sg-9a18783c • sg-9a18783c	Listeners and routing • HTTP:80
---	---	--	---

Attributes
Certain default attributes will be applied to your load balancer. You can view and edit them after creating the load balancer.

手順名	必要なパッケージ一覧は以下を参照してください。
HA構成(backyard)	https://exastro-suite.github.io/it-automation-docs/asset/Learn_ja/ITA-online-install_ja.pdf 3.11 環境構築(7/7)のライブラリ概要<Mariadb、httpd、Ansible>以外に含まれているもの

※ 設定・備考にあるバージョン(x.x.x)は、インストールするITAバージョンに読み替えてください。
また、(ITAインストール先ディレクトリ)、(Ansibleサーバのホスト名)などの文字はインストールする環境に合わせて適宜読み替えてください。

No.	目的	Master	Slave	設定	備考
ITAのインストール					
1	ITAのインストール資材を展開する	●	●	以下URLのオールインワン構成のインストールマニュアル参照 https://exastro-suite.github.io/it-automation-docs/learn_ja.html#deploy	
2				ITAインストールディレクトリを設定します。 <code>find it-automation-x.x.x -type f xargs -I{} sed -i -e "s:%%%%ITA_DIRECTORY%%%%/(ITAインストール先ディレクトリ):g" {}</code>	赤字は、環境に合わせて読み替えること
Apache削除設定					
3	Apacheサービス停止・無効化し削除する	●	●	<code>systemctl disable httpd</code> <code>systemctl stop httpd</code> <code>yum remove httpd.x86_64 httpd-filesystem.noarch httpd-tools.x86_64</code>	

手順名	必要なパッケージ一覧は以下を参照してください。
HA構成(backyard)	https://exastro-suite.github.io/it-automation-docs/asset/Learn_ia/ITA-online-install_ia.pdf 3.11 環境構築 (7/7) のライブラリ概要 <Mariadb, httpd, Ansible> 以外に含まれているもの

※ 設定・備考にあるバージョン(x.x.x)は、インストールするITAバージョンに読み替えてください。
また、(ITAインストール先ディレクトリ)、(Ansibleサーバのホスト名)などの文字はインストールする環境に合わせて適宜読み替えてください。

No.	目的	Master	Slave	設定	備考
ITAサービス停止設定					
4	ky-serviceを停止する	●	●	<pre>systemctl stop ky_activatedirectory_roleuser_replication-workflow.service systemctl stop ky_ansible_execute-workflow.service systemctl stop ky_ansible_towermasterSync-workflow.service systemctl stop ky_bulk_excel-workflow.service systemctl stop ky_change_col_to_row.service systemctl stop ky_cmdbmenuanalysis-workflow.service systemctl stop ky_create_er-workflow.service systemctl stop ky_create_param_menu_execute.service systemctl stop ky_data_portability_execute-workflow.service systemctl stop ky_hostgroup_check_loop.service systemctl stop ky_hostgroup_split.service systemctl stop ky_legacy_role_valautostup-workflow.service systemctl stop ky_legacy_role_varsautolistup-workflow.service systemctl stop ky_legacy_valautostup-workflow.service systemctl stop ky_legacy_varsautolistup-workflow.service systemctl stop ky_mail.service systemctl stop ky_pioneer_valautostup-workflow.service systemctl stop ky_pioneer_varsautolistup-workflow.service systemctl stop ky_std_checkcondition-linklist.service systemctl stop ky_std_synchronize-Collector.service systemctl stop ky_std_synchronize-Conductor.service systemctl stop ky_std_synchronize-regularly2.service systemctl stop ky_std_synchronize-regularly.service systemctl stop ky_std_synchronize-symphony.service systemctl stop ky_terraform_execute-workflow.service systemctl stop ky_terraform_checkcondition-workflow.service systemctl stop ky_terraform_varsautolistup-workflow.service systemctl stop ky_terraform_valautosetup-workflow.service</pre>	ITAのバージョンによって、存在しないサービスがあります。 その場合は、サービス停止不要です。
4	ky-service 自動起動設定を無効にする	●	●	<pre>systemctl disable ky_activatedirectory_roleuser_replication-workflow.service systemctl disable ky_ansible_execute-workflow.service systemctl disable ky_ansible_towermasterSync-workflow.service systemctl disable ky_bulk_excel-workflow.service systemctl disable ky_change_col_to_row.service systemctl disable ky_cmdbmenuanalysis-workflow.service systemctl disable ky_create_er-workflow.service systemctl disable ky_create_param_menu_execute.service systemctl disable ky_data_portability_execute-workflow.service systemctl disable ky_hostgroup_check_loop.service systemctl disable ky_hostgroup_split.service systemctl disable ky_legacy_role_valautostup-workflow.service systemctl disable ky_legacy_role_varsautolistup-workflow.service systemctl disable ky_legacy_valautostup-workflow.service systemctl disable ky_legacy_varsautolistup-workflow.service systemctl disable ky_mail.service systemctl disable ky_pioneer_valautostup-workflow.service systemctl disable ky_pioneer_varsautolistup-workflow.service systemctl disable ky_std_checkcondition-linklist.service systemctl disable ky_std_synchronize-Collector.service systemctl disable ky_std_synchronize-Conductor.service systemctl disable ky_std_synchronize-regularly2.service systemctl disable ky_std_synchronize-regularly.service systemctl disable ky_std_synchronize-symphony.service systemctl disable ky_terraform_execute-workflow.service systemctl disable ky_terraform_checkcondition-workflow.service systemctl disable ky_terraform_varsautolistup-workflow.service systemctl disable ky_terraform_valautosetup-workflow.service</pre>	ITAのバージョンによって、存在しないサービスがあります。 その場合は、設定の無効は不要です。

手順名	必要なパッケージ一覧は以下を参照してください。
HA構成(backyard)	https://exastro-suite.github.io/it-automation-docs/asset/Learn_ia/ITA-online-install_ia.pdf 3.11 環境構築 (7/7) のライブラリ概要 <Mariadb, httpd, Ansible> 以外に含まれているもの

※ 設定・備考にあるバージョン(x.x.x)は、インストールするITAバージョンに読み替えてください。
また、(ITAインストール先ディレクトリ)、(Ansibleサーバのホスト名)などの文字はインストールする環境に合わせて適宜読み替えてください。

No.	目的	Master	Slave	設定	備考
EFS設定					
5	東京に時刻設定をする	●		timedatectl set-timezone Asia/Tokyo	
6	EFSの使用準備をする	●		cd /tmp yum -y install git git clone https://github.com/aws/efs-utils yum -y install make rpm-build cd efs-utils make rpm yum -y install ./build/amazon-efs-utils*rpm	オンラインの場合は以下のコマンドでインストールしてください。 pear install HTML_AJAX-beta
7	EFSのディレクトリ設定をする	●		mount -t efs -o tls [Route53に登録したEFSの名前解決設定の情報]:/ /mnt cd /mnt mkdir -p ./((ITAインストール先ディレクトリ))/data_relay_storage/symphony mkdir -p ./((ITAインストール先ディレクトリ))/data_relay_storage/conductor mkdir -p ./((ITAインストール先ディレクトリ))/data_relay_storage/ansible_driver mkdir -p ./((ITAインストール先ディレクトリ))/ita_sessions mkdir -p ./((ITAインストール先ディレクトリ))/ita-root/webconfs/sheets mkdir -p ./((ITAインストール先ディレクトリ))/ita-root/webconfs/users mkdir -p ./((ITAインストール先ディレクトリ))/ita-root/temp mkdir -p ./((ITAインストール先ディレクトリ))/ita-root/uploadfiles mkdir -p ./((ITAインストール先ディレクトリ))/ita-root/webroot/uploadfiles mkdir -p ./((ITAインストール先ディレクトリ))/ita-root/webroot/menus/sheets mkdir -p ./((ITAインストール先ディレクトリ))/ita-root/webroot/menus/users chmod 777 ./((ITAインストール先ディレクトリ))/data_relay_storage/symphony chmod 777 ./((ITAインストール先ディレクトリ))/data_relay_storage/conductor chmod 777 ./((ITAインストール先ディレクトリ))/data_relay_storage/ansible_driver chmod 777 ./((ITAインストール先ディレクトリ))/ita_sessions chmod 755 ./((ITAインストール先ディレクトリ))/ita-root/webconfs/sheets chmod 755 ./((ITAインストール先ディレクトリ))/ita-root/webconfs/users chmod 777 ./((ITAインストール先ディレクトリ))/ita-root/temp chmod 777 ./((ITAインストール先ディレクトリ))/ita-root/uploadfiles chmod 777 ./((ITAインストール先ディレクトリ))/ita-root/webroot/uploadfiles chmod 755 ./((ITAインストール先ディレクトリ))/ita-root/webroot/menus/sheets chmod 755 ./((ITAインストール先ディレクトリ))/ita-root/webroot/menus/users	赤字は、環境に合わせて読み替えること AWSコンソールからEFSを作成してから本手順を実施する

手順名	必要なパッケージ一覧は以下を参照してください。
HA構成(backyard)	https://exastro-suite.github.io/it-automation-docs/asset/Learn_ia/ITA-online-install_ia.pdf 3.11 環境構築 (7/7) のライブラリ概要 <Mariadb, httpd, Ansible> 以外に含まれているもの

※ 設定・備考にあるバージョン(x.x.x)は、インストールするITAバージョンに読み替えてください。
また、(ITAインストール先ディレクトリ)、(Ansibleサーバのホスト名)などの文字はインストールする環境に合わせて適宜読み替えてください。

No.	目的	Master	Slave	設定	備考
8	共有ファイルのコピーを実施する	●		<pre> cp -rp /((ITAインストール先ディレクトリ))/data_relay_storage/symphony /mnt/((ITAインストール先ディレクトリ))/data_relay_storage/ cp -rp /((ITAインストール先ディレクトリ))/data_relay_storage/conductor /mnt/((ITAインストール先ディレクトリ))/data_relay_storage/ cp -rp /((ITAインストール先ディレクトリ))/data_relay_storage/ansible_driver /mnt/((ITAインストール先ディレクトリ))/data_relay_storage/ cp -rp /((ITAインストール先ディレクトリ))/ita_sessions /mnt/((ITAインストール先ディレクトリ))/ cp -rp /((ITAインストール先ディレクトリ))/ita-root/webconfs/sheets /mnt/((ITAインストール先ディレクトリ))/ita-root/webconfs/ cp -rp /((ITAインストール先ディレクトリ))/ita-root/webconfs/users /mnt/((ITAインストール先ディレクトリ))/ita-root/webconfs/ cp -rp /((ITAインストール先ディレクトリ))/ita-root/temp /mnt/((ITAインストール先ディレクトリ))/ita-root/ cp -rp /((ITAインストール先ディレクトリ))/ita-root/uploadfiles /mnt/((ITAインストール先ディレクトリ))/ita-root/ cp -rp /((ITAインストール先ディレクトリ))/ita-root/webroot/uploadfiles /mnt/((ITAインストール先ディレクトリ))/itaroot/webroot/ cp -rp /((ITAインストール先ディレクトリ))/ita-root/webroot/menus/sheets /mnt/((ITAインストール先ディレクトリ))/itaroot/webroot/menus/ cp -rp /((ITAインストール先ディレクトリ))/ita-root/webroot/menus/users /mnt/((ITAインストール先ディレクトリ))/itaroot/webroot/menus/ cd /tmp umount /mnt </pre>	赤字は、環境に合わせて読み替えること AWSコンソールからEFSを作成してから本手順を実施する
9	設定ファイルを作成する	●	●	<pre> vi /etc/systemd/system/mount-nfs-sequentially.service [Unit] Description=Workaround for mounting NFS file systems sequentially at boot time After=remote-fs.target [Service] Type=oneshot ExecStart=/bin/mount -avt nfs4 RemainAfterExit=yes [Install] WantedBy=multi-user.target </pre>	
10	上記設定を反映させる	●	●	<pre> systemctl daemon-reload systemctl enable mount-nfs-sequentially.service </pre>	

手順名 HA構成(backyard)	必要なパッケージ一覧は以下を参照してください。 https://exastro-suite.github.io/it-automation-docs/asset/Learn_ia/ITA-online-install_ia.pdf 3.11 環境構築 (7/7) のライブラリ概要 <Mariadb, httpd, Ansible> 以外に含まれているもの
-----------------------	--

※ 設定・備考にあるバージョン(x.x.x)は、インストールするITAバージョンに読み替えてください。
また、(ITAインストール先ディレクトリ)、(Ansibleサーバのホスト名)などの文字はインストールする環境に合わせて適宜読み替えてください。

No.	目的	Master	Slave	設定	備考
RDS設定					
11	RDSに接続する	●		mysql -h [DBのホスト名 (Route53にRDSの名前解決設定の情報)] -P 3306 -u admin -p<admin/パスワード>	赤字は、環境に合わせて読み替えること
12	ユーザとDBを作成する	●		CREATE USER '<ユーザ名>' IDENTIFIED BY '<パスワード>'; CREATE USER '<ユーザ名>'@'localhost' IDENTIFIED BY '<パスワード>'; CREATE DATABASE [DB名] CHARACTER SET utf8; GRANT SELECT, INSERT, UPDATE, DELETE, CREATE, DROP, REFERENCES, INDEX, ALTER, CREATE TEMPORARY TABLES, LOCK TABLES, EXECUTE, CREATE VIEW, SHOW VIEW, CREATE ROUTINE, ALTER ROUTINE, EVENT, TRIGGER ON [DB名].* TO '<ユーザ名>'@'%' WITH GRANT OPTION; GRANT SELECT, INSERT, UPDATE, DELETE, CREATE, DROP, REFERENCES, INDEX, ALTER, CREATE TEMPORARY TABLES, LOCK TABLES, EXECUTE, CREATE VIEW, SHOW VIEW, CREATE ROUTINE, ALTER ROUTINE, EVENT, TRIGGER ON [DB名].* TO '<ユーザ名>'@'localhost' WITH GRANT OPTION; exit	赤字は、環境に合わせて読み替えること
13	DBへのデータ流し込みを実施する	●		mysqldump --default-character-set=utf8 -u <ita_answer.txtで設定したDBユーザー名> -p<ita_answer.txtで設定したDB/パスワード> [ita_answer.txtで設定したDB名] > ITA_DB.sql sed -i 's/DEFINER=[^']*/*%/g' ITA_DB.sql mysql --default-character-set=utf8 -h [DBのホスト名 (Route53にEFSの名前解決設定の情報)] -P 3306 -u <RDSのマスターユーザ名> -p<RDSのマスターパスワード> [DB名] < ITA_DB.sql	赤字は、環境に合わせて読み替えること
14	MariaDB接続情報設定	●	●	①以下の文字列をbase64エンコード、rot13暗号を行った文字列を作成してください。 echo -ne "mysql:dbname=[DB名];host=[DBのホスト名 (Route53にRDSの名前解決設定の情報)]" base64 tr 'A-Za-z' '[N-ZA-Mn-za-m]' ②作成した文字列を以下のファイルに書き込んでください。※記載済みの値は削除してください vi /<ITAインストール先ディレクトリ>/ita-root/conf/commonconfs/db_connection_string.txt	例: ITAのDB名が「ITA_DB」、Route53に登録しているDNS名が rds.exastro-aws.localの場合、以下のコマンドで base64エンコード、rot13暗号を行った文字列を取得できます。 echo -ne "mysql:dbname=ITA_DB;host=rds.exastroaws.local" base64 tr '[A-Za-z]' '[N-ZA-Mn-za-m]' 赤字は、環境に合わせて読み替えること
15	MariaDBのユーザ情報設定	●	●	①MariaDBのユーザ名をbase64エンコード、rot13暗号を行った文字列を作成してください。 echo -ne "<ユーザ名>" base64 tr 'A-Za-z' '[N-ZA-Mn-za-m]' ②作成した文字列を以下のファイルに書き込んでください。※記載済みの値は削除してください。 vi /<ITAインストール先ディレクトリ>/ita-root/conf/commonconfs/db_username.txt	例: MariaDBのユーザ名が「ITA_USER」の場合、以下のコマンドで base64エンコード、rot13暗号を行った文字列を取得できます。 echo -ne "ITA_USER" base64 tr '[A-Za-z]' '[N-ZA-Mnza-m]' 赤字は、環境に合わせて読み替えること
16	MariaDBのパスワード情報設定	●	●	①MariaDBのパスワードをbase64エンコード、rot13暗号を行った文字列を作成してください。 echo -ne "<パスワード>" base64 tr 'A-Za-z' '[N-ZA-Mn-za-m]' ②作成した文字列を以下のファイルに書き込んでください。※記載済みの値は削除してください。 vi /<ITAインストール先ディレクトリ>/ita-root/conf/commonconfs/db_password.txt	例: MariaDBのパスワードが「ITA_PASSWD」の場合、 以下のコマンドでbase64エンコード、rot13暗号を行った文字列を取得できま す。 echo -ne "ITA_PASSWD" base64 tr '[A-Za-z]' '[N-ZAMn-za-m]' 赤字は、環境に合わせて読み替えること
17	MariaDBを停止する	●	●	systemctl disable mariadb systemctl stop mariadb	
18	MariaDBの削除を実施する ※本作業を実施するとmysqlコマンドによるRDSへのログインや管理が行えなくなるため、作業の実施については利用者にて判断	●	●	yum list installed grep -i mariadb yum remove mariadb.x86_64 mariadb-backup.x86_64 mariadb-common.x86_64 mariadb-connector-c.x86_64 mariadb-connector-c-config.noarch mariadb-errmsg.x86_64 mariadb-gssapi-server.x86_64 mariadbserver.x86_64 mariadb-server-utils.x86_64	

手順名	必要なパッケージ一覧は以下を参照してください。
HA構成(backyard)	https://exastro-suite.github.io/it-automation-docs/asset/Learn_ia/ITA-online-install_ia.pdf 3.11 環境構築 (7/7) のライブラリ概要 <Mariadb、httpd、Ansible> 以外に含まれているもの

※ 設定・備考にあるバージョン(x.x.x)は、インストールするITAバージョンに読み替えてください。
また、(ITAインストール先ディレクトリ)、(Ansibleサーバのホスト名)などの文字はインストールする環境に合わせて適宜読み替えてください。

No.	目的	Master	Slave	設定	備考
Pacemaker使用準備					
19	awscliをインストールする	●	●	yum -y install awscli	
20	ユーザー作成で作成したcredentials.csvを確認する	●	●	aws configure AWS Access Key ID [None]: ***** AWS Secret Access Key [None]: ***** Default region name [None]: ap-northeast-1 Default output format [None]: json	赤字は、環境に合わせて読み替えること IAM作成時にダウンロードしたcsvを参照すること
21	ホスト名とIPをhostsに登録する	●	●	uname -n vi /etc/hosts (backyardのMasterServerのローカルIP) (backyardのMasterServerのホスト名) (backyardのSlaveServerのローカルIP) (backyardのSlaveServerのホスト名) (AnsibleサーバのIPアドレス) (Ansibleサーバのホスト名)	赤字は、環境に合わせて読み替えること ※EIPを設定したAnsibleサーバのIP、ホスト名を登録すること
Pacemaker設定					
22	Pacemaker、Corosync、pcsをインストールする	●	●	dnf install --enablerepo=HighAvailability pacemaker corosync pcs	
23	pcs の管理アカウントとなるユーザーID「hacluster」のパスワードを設定する	●	●	passwd hacluster 新しいパスワード:xxxxxx 新しいパスワードを再入力してください:xxxxxx passwd: すべての認証トークンが正しく更新できました。	赤字は、環境に合わせて読み替えること
24	pcsdサービスを起動し、システムの起動時に pcsd が有効になるよう設定する	●	●	systemctl enable pcsd systemctl start pcsd systemctl status pcsd	
25	クラスターを認証する	●	●	pcs host auth <backyardのMasterServerホスト名> <backyardのSlaveServerのホスト名> Username: hacluster Password:passwd haclusterで設定したパスワード	赤字は、環境に合わせて読み替えること
26	クラスターを指定してスタートする	●	●	pcs cluster setup aws-cluster --start <backyardのMasterServerホスト名> <backyardのSlaveServerのホスト名> --force	赤字は、環境に合わせて読み替えること
27	クラスターを自動起動するように設定をする	●	●	pcs cluster enable --all	
28	STONITH (Shoot The Other Node In The Head) オプションを無効にする	●	●	pcs property set stonith-enabled=false	
29	クラスターが必要最低限の数に達していない場合でも特別な動作は行わず全ノードの管理を続行する	●	●	pcs property set no-quorum-policy=ignore	
30	属性待ち時間を設定する	●	●	pcs property set transition-delay="0s"	
30	自動フェイルバックを無効にする	●	●	pcs resource defaults resource-stickiness="INFINITY" migration-threshold="1"	

手順名	必要なパッケージ一覧は以下を参照してください。
HA構成(backyard)	https://exastro-suite.github.io/it-automation-docs/asset/Learn_ia/ITA-online-install_ia.pdf 3.11 環境構築 (7/7) のライブラリ概要 <Mariadb, httpd, Ansible> 以外に含まれているもの

※ 設定・備考にあるバージョン(x.x.x)は、インストールするITAバージョンに読み替えてください。
また、(ITAインストール先ディレクトリ)、(Ansibleサーバのホスト名)などの文字はインストールする環境に合わせて適宜読み替えてください。

No.	目的	Master	Slave	設定	備考
31	corosync設定をする	●	●	<pre>cp -p /usr/lib/systemd/system/corosync.service /etc/systemd/system/ sed -i 's/^#Restart=on.*Restart=on-failure/' /etc/systemd/system/corosync.service sed -i 's/^#RestartSec=.*RestartSec=70/' /etc/systemd/system/corosync.service</pre>	
32	Pacemakerの内部プロセスが異常になった場合もノード故障として取り扱う	●	●	<pre>sed -i 's/^#PCMK_fail_fast.*PCMK_fail_fast=yes/' /etc/sysconfig/pacemaker</pre>	
33	corosync設定をする	●	●	<pre>cp -p /usr/lib/systemd/system/pacemaker.service /etc/systemd/system sed -i "s/^#ExecStopPost=¥/bin¥/sh.*ExecStopPost=¥/bin¥/sh -c 'pidof crmd ¥ killall -TERM corosync'/" /etc/systemd/system/pacemaker.service</pre>	
34	EFSリソース設定をする	●		<pre>pcs resource create symphony Filesystem device="[Route53に登録したEFSの名前解決設定の情報]:/(ITAインストール先ディレクトリ)/data_relay_storage/symphony" directory="/(ITAインストール先ディレクトリ)/data_relay_storage/symphony" fstype="nfs" pcs resource create conductor Filesystem device="[Route53に登録したEFSの名前解決設定の情報]:/(ITAインストール先ディレクトリ)/data_relay_storage/conductor" directory="/(ITAインストール先ディレクトリ)/data_relay_storage/conductor" fstype="nfs" pcs resource create ansible_driver Filesystem device="[Route53に登録したEFSの名前解決設定の情報]:/(ITAインストール先ディレクトリ)/data_relay_storage/ansible_driver" directory="/(ITAインストール先ディレクトリ)/data_relay_storage/ansible_driver" fstype="nfs" pcs resource create webconfs_sheets Filesystem device="[Route53に登録したEFSの名前解決設定の情報]:/(ITAインストール先ディレクトリ)/ita-root/webconfs/sheets" directory="/(ITAインストール先ディレクトリ)/ita-root/webconfs/sheets" fstype="nfs" pcs resource create webconfs_users Filesystem device="[Route53に登録したEFSの名前解決設定の情報]:/(ITAインストール先ディレクトリ)/ita-root/webconfs/users" directory="/(ITAインストール先ディレクトリ)/ita-root/webconfs/users" fstype="nfs" pcs resource create temp Filesystem device="[Route53に登録したEFSの名前解決設定の情報]:/(ITAインストール先ディレクトリ)/ita-root/temp" directory="/(ITAインストール先ディレクトリ)/ita-root/temp" fstype="nfs" 7-15 pcs resource create uploadfiles Filesystem device="[Route53に登録したEFSの名前解決設定の情報]:/(ITAインストール先ディレクトリ)/ita-root/uploadfiles" directory="/(ITAインストール先ディレクトリ)/itaroot/uploadfiles" fstype="nfs" pcs resource create webroot_uploadfiles Filesystem device="[Route53に登録したEFSの名前解決設定の情報]:/(ITAインストール先ディレクトリ)/ita-root/webroot/uploadfiles" directory="/(ITAインストール先ディレクトリ)/ita-root/webroot/uploadfiles" fstype="nfs" pcs resource create menus_sheets Filesystem device="[Route53に登録したEFSの名前解決設定の情報]:/(ITAインストール先ディレクトリ)/ita-root/webroot/menus/sheets" directory="/(ITAインストール先ディレクトリ)/ita-root/webroot/menus/sheets" fstype="nfs" pcs resource create menus_users Filesystem device="[Route53に登録したEFSの名前解決設定の情報]:/(ITAインストール先ディレクトリ)/ita-root/webroot/menus/users" directory="/(ITAインストール先ディレ</pre>	赤字は、環境に合わせて読み替えること
35	EFSリソースグループ設定をする	●		<pre>pcs resource group add efsmount symphony conductor ansible_driver webconfs_sheets webconfs_users temp uploadfiles webroot_uploadfiles menus_sheets menus_users</pre>	

手順名	HA構成(backyard)
-----	----------------

必要なパッケージ一覧は以下を参照してください。
https://exastro-suite.github.io/it-automation-docs/asset/Learn_ia/ITA-online-install_ia.pdf
 3.11 環境構築 (7/7) のライブラリ概要 <Mariadb, httpd, Ansible> 以外に含まれているもの

※ 設定・備考にあるバージョン(x.x.x)は、インストールするITAバージョンに読み替えてください。
 また、(ITAインストール先ディレクトリ)、(Ansibleサーバのホスト名)などの文字はインストールする環境に合わせて適宜読み替えてください。

No.	目的			設定	備考
		Master	Slave		
36	各kyファイルリソース設定をする	●		<pre> pcs resource create ky_active_directory_roleuser_replication-workflow systemd:ky_ansible_execute-workflow ¥ op monitor interval=60 timeout=100 ¥ op start interval=0s timeout=100 ¥ op stop interval=0s timeout=100 pcs resource create ky_ansible_execute-workflow systemd:ky_ansible_execute-workflow ¥ op monitor interval=30 timeout=60 ¥ op start interval=0s timeout=60 ¥ op stop interval=0s timeout=60 pcs resource create ky_ansible_towermasterSync-workflow systemd:ky_ansible_towermasterSync-workflow ¥ op monitor interval=30 timeout=60 ¥ op start interval=0s timeout=60 ¥ op stop interval=0s timeout=60 pcs resource create ky_bulk_excel-workflow systemd:ky_bulk_excel-workflow ¥ op monitor interval=30 timeout=60 ¥ op start interval=0s timeout=60 ¥ op stop interval=0s timeout=60 pcs resource create ky_change_col_to_row systemd:ky_change_col_to_row ¥ op monitor interval=30 timeout=60 ¥ op start interval=0s timeout=60 ¥ op stop interval=0s timeout=60 pcs resource create ky_cmdbmenuanalysis-workflow systemd:ky_cmdbmenuanalysis-workflow ¥ op monitor interval=30 timeout=60 ¥ op start interval=0s timeout=60 ¥ op stop interval=0s timeout=60 </pre>	
37				<pre> pcs resource create ky_create_er-workflow systemd:ky_create_er-workflow ¥ op monitor interval=30 timeout=60 ¥ op start interval=0s timeout=60 ¥ op stop interval=0s timeout=60 pcs resource create ky_create_param_menu_execute systemd:ky_create_param_menu_execute ¥ op monitor interval=30 timeout=60 ¥ op start interval=0s timeout=60 ¥ op stop interval=0s timeout=60 pcs resource create ky_data_portability_execute-workflow systemd:ky_data_portability_execute-workflow ¥ op monitor interval=30 timeout=60 ¥ op start interval=0s timeout=60 ¥ op stop interval=0s timeout=60 pcs resource create ky_hostgroup_check_loop systemd:ky_hostgroup_check_loop ¥ op monitor interval=30 timeout=60 ¥ op start interval=0s timeout=60 ¥ op stop interval=0s timeout=60 pcs resource create ky_hostgroup_split systemd:ky_hostgroup_split ¥ op monitor interval=30 timeout=60 ¥ op start interval=0s timeout=60 ¥ op stop interval=0s timeout=60 pcs resource create ky_legacy_role_valautostup-workflow systemd:ky_legacy_role_valautostup-workflow ¥ op monitor interval=30 timeout=60 ¥ op start interval=0s timeout=60 ¥ op stop interval=0s timeout=60 </pre>	

手順名	HA構成(backyard)
-----	----------------

必要なパッケージ一覧は以下を参照してください。
https://exastro-suite.github.io/it-automation-docs/asset/Learn_ia/ITA-online-install_ia.pdf
 3.11 環境構築(7/7)のライブラリ概要<Mariadb、httpd、Ansible>以外に含まれているもの

※ 設定・備考にあるバージョン(x.x.x)は、インストールするITAバージョンに読み替えてください。
 また、(ITAインストール先ディレクトリ)、(Ansibleサーバのホスト名)などの文字はインストールする環境に合わせて適宜読み替えてください。

No.	目的	Master	Slave	設定	備考
	各kyファイルリソース設定をする	●		<pre> pcs resource create ky_legacy_role_varsautolistup-workflow systemd:ky_legacy_role_varsautolistup-workflow ¥ op monitor interval=30 timeout=60 ¥ op start interval=0s timeout=60 ¥ op stop interval=0s timeout=60 pcs resource create ky_legacy_valautostup-workflow systemd:ky_legacy_valautostup-workflow ¥ op monitor interval=30 timeout=60 ¥ op start interval=0s timeout=60 ¥ op stop interval=0s timeout=60 pcs resource create ky_legacy_varsautolistup-workflow systemd:ky_legacy_varsautolistup-workflow ¥ op monitor interval=30 timeout=60 ¥ op start interval=0s timeout=60 ¥ op stop interval=0s timeout=60 pcs resource create ky_pioneer_valautostup-workflow systemd:ky_pioneer_valautostup-workflow ¥ op monitor interval=30 timeout=60 ¥ op start interval=0s timeout=60 ¥ op stop interval=0s timeout=60 pcs resource create ky_pioneer_varsautolistup-workflow systemd:ky_pioneer_varsautolistup-workflow ¥ op monitor interval=30 timeout=60 ¥ op start interval=0s timeout=60 ¥ op stop interval=0s timeout=60 pcs resource create ky_std_checkcondition-linklist systemd:ky_std_checkcondition-linklist ¥ op monitor interval=30 timeout=60 ¥ op start interval=0s timeout=60 ¥ op stop interval=0s timeout=60 pcs resource create ky_std_synchronize-Conductor systemd:ky_std_synchronize-Conductor ¥ op monitor interval=30 timeout=60 ¥ op start interval=0s timeout=60 ¥ op stop interval=0s timeout=60 pcs resource create ky_std_synchronize-regularly systemd:ky_std_synchronize-regularly ¥ op monitor interval=30 timeout=60 ¥ op start interval=0s timeout=60 ¥ op stop interval=0s timeout=60 </pre>	

手順名	HA構成(backyard)
-----	----------------

必要なパッケージ一覧は以下を参照してください。
https://exastro-suite.github.io/it-automation-docs/asset/Learn_ja/ITA-online-install_ja.pdf
 3.11 環境構築(7/7)のライブラリ概要<Mariadb、httpd、Ansible>以外に含まれているもの

※ 設定・備考にあるバージョン(x.x.x)は、インストールするITAバージョンに読み替えてください。
 また、(ITAインストール先ディレクトリ)、(Ansibleサーバのホスト名)などの文字はインストールする環境に合わせて適宜読み替えてください。

No.	目的	Master	Slave	設定	備考
	各kyファイルリソース設定をする	●		<pre>pcs resource create ky_std_synchronize-regularly2 systemd:ky_std_synchronize-regularly2 ¥ op monitor interval=30 timeout=60 ¥ op start interval=0s timeout=60 ¥ op stop interval=0s timeout=60 pcs resource create ky_std_synchronize-symphony systemd:ky_std_synchronize-symphony ¥ op monitor interval=30 timeout=60 ¥ op start interval=0s timeout=60 ¥ op stop interval=0s timeout=60 pcs resource create ky_terraform_execute-workflow systemd:ky_terraform_execute-workflow ¥ op monitor interval=30 timeout=60 ¥ op start interval=0s timeout=60 ¥ op stop interval=0s timeout=60 pcs resource create ky_terraform_checkcondition-workflow systemd:ky_terraform_checkcondition-workflow ¥ op monitor interval=30 timeout=60 ¥ op start interval=0s timeout=60 ¥ op stop interval=0s timeout=60 pcs resource create ky_terraform_varsautolistup-workflow systemd:ky_terraform_varsautolistup-workflow ¥ op monitor interval=30 timeout=60 ¥ op start interval=0s timeout=60 ¥ op stop interval=0s timeout=60 pcs resource create ky_terraform_valautosetup-workflow systemd:ky_terraform_valautosetup-workflow ¥ op monitor interval=30 timeout=60 ¥ op start interval=0s timeout=60 ¥ op stop interval=0s timeout=60</pre>	
38	kyリソースのグループ化を実施する	●		<pre>pcs resource group add ky_services ky_activedirectory_roleuser_replication-workflow ky_ansible_execute-workflow ky_ansible_towermasterSync-workflow ky_bulk_excel-workflow ky_change_col_to_row ky_cmdbmenuanalysis-workflow ky_create_er-workflow ky_create_param_menu_execute ky_data_portability_execute-workflow ky_hostgroup_check_loop ky_hostgroup_split ky_legacy_role_valautostup-workflow ky_legacy_role_varsautolistup-workflow ky_legacy_valautostup-workflow ky_legacy_varsautolistup- workflow ky_pioneer_valautostup-workflow ky_pioneer_varsautolistup-workflow ky_std_checkcondition-linklist ky_std_synchronize- Conductor ky_std_synchronize-regularly ky_std_synchronize-regularly2 ky_std_synchronize-symphony ky_terraform_execute-workflow ky_terraform_checkcondition-workflow ky_terraform_varsautolistup-workflow ky_terraform_valautosetup-workflow</pre>	
39	これまでに作成したリソースを指定し、リソースグループ起動/停止の順序を指定する	●		<pre>pcs constraint order set efsmount ky_services</pre>	
40	これまでに設定したリソースが同ノードで起動する制限を指定する	●		<pre>pcs constraint colocation set efsmount ky_services</pre>	
41	HAソフトのサービスを開始する	●	●	<pre>systemctl enable corosync systemctl enable pacemaker systemctl start corosync systemctl start pacemaker</pre>	

手順名	必要なパッケージ一覧は以下を参照してください。
HA構成(Ansible)	https://exastro-suite.github.io/it-automation-docs/asset/Learn_ja/ITA-online-install_ja.pdf 3.11 環境構築(7/7)のライブラリ概要<Mariadb、httpd、Ansible>以外に含まれているもの

※ 設定・備考にあるバージョン(x.x.x)は、インストールするITAバージョンに読み替えてください。
また、(ITAインストール先ディレクトリ)、(Ansibleサーバのホスト名)などの文字はインストールする環境に合わせて適宜読み替えてください。

No.	目的	Master	Slave	設定	備考
ITAのインストール					
1	ITAのインストール資材を展開する	●	●	以下URLのオールインワン構成のインストールマニュアル参照 https://exastro-suite.github.io/it-automation-docs/learn_ja.html#deploy	
2				ITAインストールディレクトリを設定します。 <code>find it-automation-x.x.x -type f xargs -I{} sed -i -e "s:%%%%ITA_DIRECTORY%%%%/(ITAインストール先ディレクトリ):g" {}</code>	赤字は、環境に合わせて読み替えること
Apache停止・無効化設定					
3	Apacheサービス停止・無効化する	●	●	<code>systemctl disable httpd</code> <code>systemctl stop httpd</code>	

手順名	必要なパッケージ一覧は以下を参照してください。
HA構成(Ansible)	https://exastro-suite.github.io/it-automation-docs/asset/Learn_ia/ITA-online-install_ia.pdf 3.11 環境構築 (7/7) のライブラリ概要 <Mariadb, httpd, Ansible> 以外に含まれているもの

※ 設定・備考にあるバージョン(x.x.x)は、インストールするITAバージョンに読み替えてください。
また、(ITAインストール先ディレクトリ)、(Ansibleサーバのホスト名)などの文字はインストールする環境に合わせて適宜読み替えてください。

No.	目的	Master	Slave	設定	備考
ITAサービス停止設定					
4	ky-serviceを停止する	●	●	<pre>systemctl stop ky_activatedirectory_roleuser_replication-workflow.service systemctl stop ky_ansible_execute-workflow.service systemctl stop ky_ansible_towermasterSync-workflow.service systemctl stop ky_bulk_excel-workflow.service systemctl stop ky_change_col_to_row.service systemctl stop ky_cmdbmenuanalysis-workflow.service systemctl stop ky_create_er-workflow.service systemctl stop ky_create_param_menu_execute.service systemctl stop ky_data_portability_execute-workflow.service systemctl stop ky_hostgroup_check_loop.service systemctl stop ky_hostgroup_split.service systemctl stop ky_legacy_role_valautostup-workflow.service systemctl stop ky_legacy_role_varsautolistup-workflow.service systemctl stop ky_legacy_valautostup-workflow.service systemctl stop ky_legacy_varsautolistup-workflow.service systemctl stop ky_mail.service systemctl stop ky_pioneer_valautostup-workflow.service systemctl stop ky_pioneer_varsautolistup-workflow.service systemctl stop ky_std_checkcondition-linklist.service systemctl stop ky_std_synchronize-Collector.service systemctl stop ky_std_synchronize-Conductor.service systemctl stop ky_std_synchronize-regularly2.service systemctl stop ky_std_synchronize-regularly.service systemctl stop ky_std_synchronize-symphony.service systemctl stop ky_terraform_execute-workflow.service systemctl stop ky_terraform_checkcondition-workflow.service systemctl stop ky_terraform_varsautolistup-workflow.service systemctl stop ky_terraform_valautosetup-workflow.service</pre>	ITAのバージョンによって、存在しないサービスがあります。 その場合は、サービス停止不要です。
4	ky-service 自動起動設定を無効にする	●	●	<pre>systemctl disable ky_activatedirectory_roleuser_replication-workflow.service systemctl disable ky_ansible_execute-workflow.service systemctl disable ky_ansible_towermasterSync-workflow.service systemctl disable ky_bulk_excel-workflow.service systemctl disable ky_change_col_to_row.service systemctl disable ky_cmdbmenuanalysis-workflow.service systemctl disable ky_create_er-workflow.service systemctl disable ky_create_param_menu_execute.service systemctl disable ky_data_portability_execute-workflow.service systemctl disable ky_hostgroup_check_loop.service systemctl disable ky_hostgroup_split.service systemctl disable ky_legacy_role_valautostup-workflow.service systemctl disable ky_legacy_role_varsautolistup-workflow.service systemctl disable ky_legacy_valautostup-workflow.service systemctl disable ky_legacy_varsautolistup-workflow.service systemctl disable ky_mail.service systemctl disable ky_pioneer_valautostup-workflow.service systemctl disable ky_pioneer_varsautolistup-workflow.service systemctl disable ky_std_checkcondition-linklist.service systemctl disable ky_std_synchronize-Collector.service systemctl disable ky_std_synchronize-Conductor.service systemctl disable ky_std_synchronize-regularly2.service systemctl disable ky_std_synchronize-regularly.service systemctl disable ky_std_synchronize-symphony.service systemctl disable ky_terraform_execute-workflow.service systemctl disable ky_terraform_checkcondition-workflow.service systemctl disable ky_terraform_varsautolistup-workflow.service systemctl disable ky_terraform_valautosetup-workflow.service</pre>	ITAのバージョンによって、存在しないサービスがあります。 その場合は、設定の無効は不要です。

手順名	必要なパッケージ一覧は以下を参照してください。
HA構成(Ansible)	https://exastro-suite.github.io/it-automation-docs/asset/Learn_ia/ITA-online-install_ia.pdf 3.11 環境構築 (7/7) のライブラリ概要 < Mariadb, httpd, Ansible > 以外に含まれているもの

※ 設定・備考にあるバージョン(x.x.x)は、インストールするITAバージョンに読み替えてください。
また、(ITAインストール先ディレクトリ)、(Ansibleサーバのホスト名)などの文字はインストールする環境に合わせて適宜読み替えてください。

No.	目的	Master	Slave	設定	備考
EFS設定					
5	東京に時刻設定をする	●		timedatectl set-timezone Asia/Tokyo	
6	EFSの使用準備をする	●		cd /tmp yum -y install git git clone https://github.com/aws/efs-utils yum -y install make rpm-build cd efs-utils make rpm yum -y install ./build/amazon-efs-utils*rpm	オンラインの場合は以下のコマンドでインストールしてください。 pear install HTML_AJAX-beta
7	設定ファイルを作成する	●	●	vi /etc/systemd/system/mount-nfs-sequentially.service [Unit] Description=Workaround for mounting NFS file systems sequentially at boot time After=remote-fs.target [Service] Type=oneshot ExecStart=/bin/mount -avt nfs4 RemainAfterExit=yes [Install] WantedBy=multi-user.target	
8	上記設定を反映させる	●	●	systemctl daemon-reload systemctl enable mount-nfs-sequentially.service	

手順名	必要なパッケージ一覧は以下を参照してください。
HA構成(Ansible)	https://exastro-suite.github.io/it-automation-docs/asset/Learn_ja/ITA-online-install_ja.pdf 3.11 環境構築 (7/7) のライブラリ概要<Mariadb, httpd, Ansible> 以外に含まれているもの

※ 設定・備考にあるバージョン(x.x.x)は、インストールするITAバージョンに読み替えてください。
また、(ITAインストール先ディレクトリ)、(Ansibleサーバのホスト名)などの文字はインストールする環境に合わせて適宜読み替えてください。

No.	目的	Master	Slave	設定	備考
RDS設定					
9	MariaDB接続情報設定	●	●	<p>①以下の文字列をbase64エンコード、rot13暗号を行った文字列を作成してください。 echo -ne "mysql:dbname=[DB名];host=[DBのホスト名 (Route53にRDSの名前解決設定の情報)]" base64 tr '[-Za-z]' '[N-ZA-Mn-za-m]'</p> <p>②作成した文字列を以下のファイルに書き込んでください。※記載済みの値は削除してください vi /(ITAインストール先ディレクトリ)/ita-root/conf/commonconfs/db_connection_string.txt</p>	<p>例: ITAのDB名が「ITA_DB」、Route53に登録しているDNS名が rds.exastro-aws.localの場合、以下のコマンドで base64エンコード、rot13暗号を行った文字列を取得できます。</p> <p>echo -ne "mysql:dbname=ITA_DB;host=rds.exastroaws.local" base64 tr '[-Za-z]' '[N-ZA-Mn-za-m]'</p> <p>赤字は、環境に合わせて読み替えること</p>
10	MariaDBのユーザ情報設定	●	●	<p>①MariaDBのユーザ名をbase64エンコード、rot13暗号を行った文字列を作成してください。 echo -ne "<ユーザ名>" base64 tr '[-Za-z]' '[N-ZA-Mn-za-m]'</p> <p>②作成した文字列を以下のファイルに書き込んでください。※記載済みの値は削除してください。 vi /(ITAインストール先ディレクトリ)/ita-root/conf/commonconfs/db_username.txt</p>	<p>例: MariaDBのユーザ名が「ITA_USER」の場合、以下のコマンドで base64エンコード、rot13暗号を行った文字列を取得できます。</p> <p>echo -ne "ITA_USER" base64 tr '[-Za-z]' '[N-ZA-Mn-za-m]'</p> <p>赤字は、環境に合わせて読み替えること</p>
11	MariaDBのパスワード情報設定	●	●	<p>①MariaDBのパスワードをbase64エンコード、rot13暗号を行った文字列を作成してください。 echo -ne "<パスワード>" base64 tr '[-Za-z]' '[N-ZA-Mn-za-m]'</p> <p>②作成した文字列を以下のファイルに書き込んでください。※記載済みの値は削除してください。 vi /(ITAインストール先ディレクトリ)/ita-root/conf/commonconfs/db_password.txt</p>	<p>例: MariaDBのパスワードが「ITA_PASSWD」の場合、以下のコマンドでbase64エンコード、rot13暗号を行った文字列を取得できません。</p> <p>echo -ne "ITA_PASSWD" base64 tr '[-Za-z]' '[N-ZAMn-za-m]'</p> <p>赤字は、環境に合わせて読み替えること</p>
12	MariaDBを停止する	●	●	<pre>systemctl disable mariadb systemctl stop mariadb</pre>	
13	MariaDBの削除を実施する	●	●	<pre>yum list installed grep -i mariadb yum remove mariadb.x86_64 mariadb-backup.x86_64 mariadb-common.x86_64 mariadb-connector-c.x86_64 mariadb-connector-c-config.noarch mariadb-errmsg.x86_64 mariadb-gssapi-server.x86_64 mariadbserver.x86_64 mariadb-server- utils.x86_64</pre>	
Pacemaker使用準備					
14	awscliをインストールする	●	●	<pre>yum -y install awscli</pre>	
15	ユーザー作成で作成したcredentials.csvを確認する	●	●	<pre>aws configure AWS Access Key ID [None]: ***** AWS Secret Access Key [None]: ***** Default region name [None]: ap-northeast-1 Default output format [None]: json</pre>	<p>赤字は、環境に合わせて読み替えること IAM作成時にダウンロードしたcsvを参照すること</p>
16	ホスト名とIPをhostsに登録する	●	●	<pre>uname -n vi /etc/hosts (AnsibleサーバのMasterServerIPアドレス) (AnsibleサーバのMasterServerホスト名) (AnsibleサーバのSlaveServerIPアドレス) (AnsibleサーバのSlaveServerホスト名)</pre>	<p>赤字は、環境に合わせて読み替えること</p>

手順名	必要なパッケージ一覧は以下を参照してください。
HA構成(Ansible)	https://exastro-suite.github.io/it-automation-docs/asset/Learn_ja/ITA-online-install_ja.pdf 3.11 環境構築(7/7)のライブラリ概要<Mariadb、httpd、Ansible>以外に含まれているもの

※ 設定・備考にあるバージョン(x.x.x)は、インストールするITAバージョンに読み替えてください。
また、(ITAインストール先ディレクトリ)、(Ansibleサーバのホスト名)などの文字はインストールする環境に合わせて適宜読み替えてください。

No.	目的	Master	Slave	設定	備考
Pacemaker設定					
17	Pacemaker、Corosync、pcsをインストールする	●	●	<code>dnf install --enablerepo=HighAvailability pacemaker corosync pcs</code>	
18	pcs の管理アカウントとなるユーザーID「hacluster」のパスワードを設定する	●	●	<code>passwd hacluster</code> 新しいパスワード:xxxxxx 新しいパスワードを再入力してください:xxxxxx <code>passwd: すべての認証トークンが正しく更新できました。</code>	赤字は、環境に合わせて読み替えること
19	pcsdサービスを起動し、システムの起動時に pcsd が有効になるよう設定する	●	●	<code>systemctl enable pcsd</code> <code>systemctl start pcsd</code> <code>systemctl status pcsd</code>	
20	クラスターを認証する	●	●	<code>pcs host auth <AnsibleのMasterServerホスト名> <AnsibleのSlaveServerのホスト名></code> Username: hacluster Password:passwd haclusterで設定したパスワード	赤字は、環境に合わせて読み替えること
21	クラスターを指定してスタートする	●	●	<code>pcs cluster setup aws-cluster --start <AnsibleのMasterServerホスト名> <AnsibleのSlaveServerのホスト名> --force</code>	赤字は、環境に合わせて読み替えること
22	クラスターを自動起動するように設定をする	●	●	<code>pcs cluster enable --all</code>	
23	STONITH (Shoot The Other Node In The Head) オプションを無効にする	●	●	<code>pcs property set stonith-enabled=false</code>	
24	クラスターが必要最低限の数に達していない場合でも特別な動作は行わず全ノードの管理を続行する	●	●	<code>pcs property set no-quorum-policy=ignore</code>	
25	属性待ち時間を設定する	●	●	<code>pcs property set transition-delay="0s"</code>	
25	自動フェイルバックを無効にする	●	●	<code>pcs resource defaults resource-stickiness="INFINITY" migration-threshold="1"</code>	

手順名	必要なパッケージ一覧は以下を参照してください。
HA構成(Ansible)	https://exastro-suite.github.io/it-automation-docs/asset/Learn_ia/ITA-online-install_ia.pdf 3.11 環境構築 (7/7) のライブラリ概要 < Mariadb, httpd, Ansible > 以外に含まれているもの

※ 設定・備考にあるバージョン(x.x.x)は、インストールするITAバージョンに読み替えてください。
また、(ITAインストール先ディレクトリ)、(Ansibleサーバのホスト名)などの文字はインストールする環境に合わせて適宜読み替えてください。

No.	目的	Master	Slave	設定	備考
26	corosync設定をする	●	●	cp -p /usr/lib/systemd/system/corosync.service /etc/systemd/system/ sed -i 's/^#Restart=on.*Restart=on-failure/' /etc/systemd/system/corosync.service sed -i 's/^#RestartSec=.*RestartSec=70/' /etc/systemd/system/corosync.service	
27	Pacemakerの内部プロセスが異常になった場合もノード故障として取り扱う	●	●	sed -i 's/^#PCMK_fai¥_fail¥_fast.*PCMK_fai_fast=yes/' /etc/sysconfig/pacemaker	
28	corosync設定をする	●	●	cp -p /usr/lib/systemd/system/pacemaker.service /etc/systemd/system sed -i "s/^#¥ ExecStopPost=¥/bin¥/sh.*/ExecStopPost=¥/bin¥/sh -c 'pidof crmd ¥ killall -TERM corosync'/" /etc/systemd/system/pacemaker.service	
29	IP付け替えの設定をする	●	●	vi /etc/sysconfig/pacemaker /etc/sysconfig/pacemakerの中身 # Set the options to pass to valgrind, when valgrind is enabled. See # valgrind(1) man page for details. "--vgdb=no" is specified because # pacemaker-execd can lower privileges when executing commands, which would # otherwise leave a bunch of unremovable files in /tmp. VALGRIND_OPTS="--leak-check=full --trace-children=no --vgdb=no --num-callers=25 --logfile=/var/lib/pacemaker/valgrind-%p -- suppressions=/usr/share/pacemaker/tests/valgrindpcmk.suppressions --gen-suppressions=all" AWS_DEFAULT_REGION=ap-northeast-1 ←記載部分	赤字は、環境に合わせて読み替えること (リージョンが東京の場合は、末尾に AWS_DEFAULT_REGION=ap-northeast-1 と記載)
30	httpd設定をする	●		pcs resource create httpd systemd:httpd ¥ op monitor interval=60 timeout=100 ¥ op start interval=0s timeout=100 ¥ op stop interval=0s timeout=100	
31	eip設定をする	●		pcs resource create eip ocf:heartbeat:awsip ¥ elastic_ip="ElasticIPのIPアドレス" awscli="\$(which aws)" ¥ allocation_id=ElasticIPの割り当てID ¥ op start timeout="120s" interval="0s" on-fail="restart" ¥ op monitor timeout="20s" interval="30s" on-fail="restart" ¥ op stop timeout="120s" interval="0s" on-fail="block"	赤字は、環境に合わせて読み替えること
32	httpd, eipリソースグループ設定をする	●		pcs resource group add floatingip httpd eip	
33	EFSリソース設定をする	●		pcs resource create symphony Filesystem device="[Route53に登録したEFSの名前解決設定の情報]:/(ITA インストール先ディレクトリ)/data_relay_storage/symphony" directory="/(ITAインストール先ディレクト リ)/data_relay_storage/symphony" fstype='nfs' pcs resource create conductor Filesystem device="[Route53に登録したEFSの名前解決設定の情報]:/(ITA インストール先ディレクトリ)/data_relay_storage/conductor" directory="/(ITAインストール先ディレクト リ)/data_relay_storage/conductor" fstype='nfs' pcs resource create ansible_driver Filesystem device="[Route53に登録したEFSの名前解決設定の情 報]:/(ITAインストール先ディレクトリ)/data_relay_storage/ansible_driver" directory="/(ITAインストール先ディ レクトリ)/data_relay_storage/ansible_driver" fstype='nfs'	赤字は、環境に合わせて読み替えること
34	EFSリソースグループ設定をする	●		pcs resource group add efsmount symphony conductor ansible_drive	
35	これまでに作成したリソースを指定し、リソースグループ起動/停止の順序を指定する	●		pcs constraint order set efsmount floatingip	
36	これまでに設定したリソースが同ノードで起動する制限を指定する	●		pcs constraint colocation set efsmount floatingip	
37	HAソフトのサービスを開始する	●	●	systemctl enable corosync systemctl enable pacemaker systemctl start corosync systemctl start pacemaker	

手順名	必要なパッケージ一覧は以下を参照してください。
HA構成(Web・AP)	https://exastro-suite.github.io/it-automation-docs/asset/Learn_ja/ITA-online-install_ja.pdf 3.11 環境構築(7/7)のライブラリ概要<Mariadb, httpd, Ansible>以外に含まれているもの

※ 設定・備考にあるバージョン(x.x.x)は、インストールするITAバージョンに読み替えてください。
また、(ITAインストール先ディレクトリ)、(Ansibleサーバのホスト名)などの文字はインストールする環境に合わせて適宜読み替えてください。

No.	目的	設定	備考
ITAのインストール			
1	ITAのインストール資材を展開する	以下URLのオールインワン構成のインストールマニュアル参照 https://exastro-suite.github.io/it-automation-docs/learn_ja.html#deploy	
2		ITAインストールディレクトリを設定します。 <code>find it-automation-x.x.x -type f xargs -I{} sed -i -e "s:%%%%ITA_DIRECTORY%%%%/(ITAインストール先ディレクトリ):g" {}</code>	赤字は、環境に合わせて読み替えること
Apache停止設定			
3	Apacheサービスを停止する	<code>systemctl stop httpd</code>	

手順名 HA構成(Web・AP)	必要なパッケージ一覧は以下を参照してください。 https://exastro-suite.github.io/it-automation-docs/asset/Learn_ia/ITA-online-install_ia.pdf 3.11 環境構築(7/7)のライブラリ概要<Mariadb、httpd、Ansible>以外に含まれているもの
---------------------	--

※ 設定・備考にあるバージョン(x.x.x)は、インストールするITAバージョンに読み替えてください。
また、(ITAインストール先ディレクトリ)、(Ansibleサーバのホスト名)などの文字はインストールする環境に合わせて適宜読み替えてください。

No.	目的	設定	備考
ITAサービス停止設定			
4	ky-serviceを停止する	<pre>systemctl stop ky_activedirectory_roleuser_replication-workflow.service systemctl stop ky_ansible_execute-workflow.service systemctl stop ky_ansible_towermasterSync-workflow.service systemctl stop ky_bulk_excel-workflow.service systemctl stop ky_change_col_to_row.service systemctl stop ky_cmdbmenuanalysis-workflow.service systemctl stop ky_create_er-workflow.service systemctl stop ky_create_param_menu_execute.service systemctl stop ky_data_portability_execute-workflow.service systemctl stop ky_hostgroup_check_loop.service systemctl stop ky_hostgroup_split.service systemctl stop ky_legacy_role_valautostup-workflow.service systemctl stop ky_legacy_role_varsautolistup-workflow.service systemctl stop ky_legacy_valautostup-workflow.service systemctl stop ky_legacy_varsautolistup-workflow.service systemctl stop ky_mail.service systemctl stop ky_pioneer_valautostup-workflow.service systemctl stop ky_pioneer_varsautolistup-workflow.service systemctl stop ky_std_checkcondition-linklist.service systemctl stop ky_std_synchronize-Collector.service systemctl stop ky_std_synchronize-Conductor.service systemctl stop ky_std_synchronize-regularly2.service systemctl stop ky_std_synchronize-regularly.service systemctl stop ky_std_synchronize-symphony.service systemctl stop ky_terraform_execute-workflow.service systemctl stop ky_terraform_checkcondition-workflow.service systemctl stop ky_terraform_varsautolistup-workflow.service systemctl stop ky_terraform_valautosetup-workflow.service</pre>	ITAのバージョンによって、存在しないサービスがあります。 その場合は、サービス停止不要です。
4	ky-service 自動起動設定を無効にする	<pre>systemctl disable ky_activedirectory_roleuser_replication-workflow.service systemctl disable ky_ansible_execute-workflow.service systemctl disable ky_ansible_towermasterSync-workflow.service systemctl disable ky_bulk_excel-workflow.service systemctl disable ky_change_col_to_row.service systemctl disable ky_cmdbmenuanalysis-workflow.service systemctl disable ky_create_er-workflow.service systemctl disable ky_create_param_menu_execute.service systemctl disable ky_data_portability_execute-workflow.service systemctl disable ky_hostgroup_check_loop.service systemctl disable ky_hostgroup_split.service systemctl disable ky_legacy_role_valautostup-workflow.service systemctl disable ky_legacy_role_varsautolistup-workflow.service systemctl disable ky_legacy_valautostup-workflow.service systemctl disable ky_legacy_varsautolistup-workflow.service systemctl disable ky_mail.service systemctl disable ky_pioneer_valautostup-workflow.service systemctl disable ky_pioneer_varsautolistup-workflow.service systemctl disable ky_std_checkcondition-linklist.service systemctl disable ky_std_synchronize-Collector.service systemctl disable ky_std_synchronize-Conductor.service systemctl disable ky_std_synchronize-regularly2.service systemctl disable ky_std_synchronize-regularly.service systemctl disable ky_std_synchronize-symphony.service systemctl disable ky_terraform_execute-workflow.service systemctl disable ky_terraform_checkcondition-workflow.service systemctl disable ky_terraform_varsautolistup-workflow.service systemctl disable ky_terraform_valautosetup-workflow.service</pre>	ITAのバージョンによって、存在しないサービスがあります。 その場合は、設定の無効は不要です。

手順名 HA構成(Web・AP)	必要なパッケージ一覧は以下を参照してください。 https://exastro-suite.github.io/it-automation-docs/asset/Learn_ia/ITA-online-install_ia.pdf 3.11 環境構築(7/7)のライブラリ概要<Mariadb、httpd、Ansible>以外に含まれているもの
---------------------	--

※ 設定・備考にあるバージョン(x.x.x)は、インストールするITAバージョンに読み替えてください。
また、(ITAインストール先ディレクトリ)、(Ansibleサーバのホスト名)などの文字はインストールする環境に合わせて適宜読み替えてください。

No.	目的	設定	備考
EFS設定			
5	東京に時刻設定をする	timedatectl set-timezone Asia/Tokyo	
6	EFSの使用準備をする	cd /tmp yum -y install git git clone https://github.com/aws/efs-utils yum -y install make rpm-build cd efs-utils make rpm yum -y install ./build/amazon-efs-utils*rpm	
7	設定ファイルを作成する	vi /etc/fstab [Route53に登録したEFSの名前解決設定の情報]:/ (ITAインストール先ディレクトリ)/data_relay_storage/symphony efs defaults,_netdev 0 0 [Route53に登録したEFSの名前解決設定の情報]:/ (ITAインストール先ディレクトリ)/data_relay_storage/conductor efs defaults,_netdev 0 0 [Route53に登録したEFSの名前解決設定の情報]:/ (ITAインストール先ディレクトリ)/data_relay_storage/ansible_driver efs defaults,_netdev 0 0 [Route53に登録したEFSの名前解決設定の情報]:/ (ITAインストール先ディレクトリ)/ita_sessions efs defaults,_netdev 0 0 [Route53に登録したEFSの名前解決設定の情報]:/ (ITAインストール先ディレクトリ)/ita-root/webconfs/sheets efs defaults,_netdev 0 0 [Route53に登録したEFSの名前解決設定の情報]:/ (ITAインストール先ディレクトリ)/ita-root/webconfs/users efs defaults,_netdev 0 0 [Route53に登録したEFSの名前解決設定の情報]:/ (ITAインストール先ディレクトリ)/ita-root/temp efs defaults,_netdev 0 0 [Route53に登録したEFSの名前解決設定の情報]:/ (ITAインストール先ディレクトリ)/ita-root/uploadfiles efs defaults,_netdev 0 0 [Route53に登録したEFSの名前解決設定の情報]:/ (ITAインストール先ディレクトリ)/ita-root/webroot/uploadfiles efs defaults,_netdev 0 0 [Route53に登録したEFSの名前解決設定の情報]:/ (ITAインストール先ディレクトリ)/ita-root/webroot/menus/sheets efs defaults,_netdev 0 0 [Route53に登録したEFSの名前解決設定の情報]:/ (ITAインストール先ディレクトリ)/ita-root/webroot/menus/user efs defaults,_netdev 0 0	赤字は、環境に合わせて読み替えること
8	上記設定を反映させる	systemctl daemon-reload systemctl enable mount-nfs-sequentially.service mount -a	

手順名 HA構成(Web・AP)	必要なパッケージ一覧は以下を参照してください。 https://exastro-suite.github.io/it-automation-docs/asset/Learn_ia/ITA-online-install_ia.pdf 3.11 環境構築(7/7)のライブラリ概要<Mariadb、httpd、Ansible>以外に含まれているもの
---------------------	--

※ 設定・備考にあるバージョン(x.x.x)は、インストールするITAバージョンに読み替えてください。
また、(ITAインストール先ディレクトリ)、(Ansibleサーバのホスト名)などの文字はインストールする環境に合わせて適宜読み替えてください。

No.	目的	設定	備考
RDS設定			
9	MariaDB接続情報設定	<p>①以下の文字列をbase64エンコード、rot13暗号を行った文字列を作成してください。 echo -ne "mysql:dbname=[DB名];host=[DBのホスト名(Route53にRDSの名前解決設定の情報)]" base64 tr '[-Za-z]' '[N-ZA-Mn-za-m]'</p> <p>②作成した文字列を以下のファイルに書き込んでください。※記載済みの値は削除してください vi /(ITAインストール先ディレクトリ)/ita-root/confd/commonconfs/db_connection_string.txt</p>	<p>例: ITAのDB名が「ITA_DB」、Route53に登録しているDNS名が rds.exastro-aws.localの場合、以下のコマンドで base64エンコード、rot13暗号を行った文字列を取得できます。</p> <p>echo -ne "mysql:dbname=ITA_DB;host=rds.exastroaws.local" base64 tr '[-Za-z]' '[N-ZA-Mn-za-m]'</p> <p>赤字は、環境に合わせて読み替えること</p>
10	MariaDBのユーザ情報設定	<p>①MariaDBのユーザ名をbase64エンコード、rot13暗号を行った文字列を作成してください。 echo -ne "<ユーザ名>" base64 tr '[-Za-z]' '[N-ZA-Mn-za-m]'</p> <p>②作成した文字列を以下のファイルに書き込んでください。※記載済みの値は削除してください。 vi /(ITAインストール先ディレクトリ)/ita-root/confd/commonconfs/db_username.txt</p>	<p>例: MariaDBのユーザ名が「ITA_USER」の場合、以下のコマンドで base64エンコード、rot13暗号を行った文字列を取得できます。</p> <p>echo -ne "ITA_USER" base64 tr '[-Za-z]' '[N-ZA-Mn-za-m]'</p> <p>赤字は、環境に合わせて読み替えること</p>
11	MariaDBのパスワード情報設定	<p>①MariaDBのパスワードをbase64エンコード、rot13暗号を行った文字列を作成してください。 echo -ne "<パスワード>" base64 tr '[-Za-z]' '[N-ZA-Mn-za-m]'</p> <p>②作成した文字列を以下のファイルに書き込んでください。※記載済みの値は削除してください。 vi /(ITAインストール先ディレクトリ)/ita-root/confd/commonconfs/db_password.txt</p>	<p>例: MariaDBのパスワードが「ITA_PASSWD」の場合、以下のコマンドでbase64エンコード、rot13暗号を行った文字列を取得できます。</p> <p>echo -ne "ITA_PASSWD" base64 tr '[-Za-z]' '[N-ZAMn-za-m]'</p> <p>赤字は、環境に合わせて読み替えること</p>
12	MariaDBを停止する	systemctl disable mariadb systemctl stop mariadb	
13	MariaDBの削除を実施する	yum list installed grep -i mariadb yum remove mariadb.x86_64 mariadb-backup.x86_64 mariadb-common.x86_64 mariadb-connector-c.x86_64 mariadb-connector-c-config.noarch mariadb-errmsg.x86_64 mariadb-gssapi-server.x86_64 mariadbserver.x86_64 mariadb-server-utils.x86_64	
hosts設定			
14	ホスト名とIPをhostsに登録する	uname -n vi /etc/hosts 127.0.0.1 (Web・APサーバのホスト名) exastro-it-automation (AnsibleサーバのIPアドレス) (Ansibleサーバのホスト名)	赤字は、環境に合わせて読み替えること ※EIPを設定したAnsibleサーバのIP、ホスト名を登録すること
Pacemaker設定			
15	Apacheを起動する	systemctl start httpd	