

### [IAM ロールの作成]

1. IAM のマネージメントコンソールからロールを選択し、[ロールの作成]ボタンをおします

#### Identity and Access Management (IAM)

---

##### 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

ロールの作成

ロールの削除

2. [EC2]を選んで、[次のステップ：アクセス権限]をおします

#### 一般的なユースケース

##### EC2

Allows EC2 instances to call AWS services on your behalf.

##### Lambda

Allows Lambda functions to call AWS services on your behalf.

または、サービスを選択してユースケースを表示します

API Gateway	CodeBuild	EMR	IoT SiteWise	RDS
AWS Backup	CodeDeploy	EMR Containers	IoT Things Graph	Redshift
AWS Chatbot	CodeGuru	ElastiCache	KMS	Rekognition
AWS Marketplace	CodeStar Notifications	Elastic Beanstalk	Kinesis	RoboMaker
AWS Support	Comprehend	Elastic Container Registry	Lake Formation	S3
Amplify	Config	Elastic Container Service	Lambda	SMS
AppStream 2.0	Connect	Elastic Transcoder	Lex	SNS
AppSync	DMS	ElasticLoadBalancing	License Manager	SWF
Application Auto Scaling	Data Lifecycle Manager	EventBridge	MQ	SageMaker
Application Discovery Service	Data Pipeline	Forecast	Machine Learning	Security Hub
Batch	DataBrew	GameLift	Macie	Service Catalog
Braket	DataSync	Global Accelerator	Managed Blockchain	Step Functions
	DeepLens	Glue	MediaConvert	Storage Gateway

\* 必須

キャンセル

次のステップ: アクセス権限

3. [ポリシーの作成]ボタンをおします

### ▼ Attach アクセス権限ポリシー

新しいロールにアタッチするポリシーを 1 つ以上選択します。

ポリシーの作成

4. [JSON]タブを選んで、[rolepolicy.txt]の中身をコピーし、(もともと IAM ロールあるデフォルト文字列は削除します) [次のステップ: タグ]をおします

## 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

```
74 {
75   "Sid": "AllowFISExperimentRoleSSMSendCommand",
76   "Effect": "Allow",
77   "Action": [
78     "ssm:SendCommand"
79   ],
80   "Resource": [
81     "arn:aws:ec2:*:*:instance/*",
82     "arn:aws:ssm:*:*:document/*"
83   ],
84 },
85 {
86   "Sid": "AllowFISExperimentRoleSSMCancelCommand",
87   "Effect": "Allow",
88   "Action": [
89     "ssm:CancelCommand"
90   ],
91   "Resource": "*"
92 },
93 }
```

セキュリティ: 0 エラー: 0 警告: 0 提案: 0

文字数: 1,568 / 6,144.

キャンセル 次のステップ: タグ

5. タグは空欄のまま、[次のステップ：確認]をおし、名前に[fispolicy]と入力し、[ポリシーの作成]をおします

## ポリシーの作成

1 2 3

### ポリシーの確認

名前\*

fispolicy

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

このポリシーはアクセス許可を提供しないいくつかのアクション、リソース、条件が定義されています。アクセス権限を付与するには、ポリシーに該当するリソースまたは条件を持つアクションがある必要があります。詳細については、[残りの表示](#)。 [詳細はこちら](#) を選択します。

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 (282 サービス中 7) 残りの 275 を表示			
EC2	制限: リスト, 書き込み	複数	なし
EKS	制限: 読み込み	すべてのリソース	なし
Elastic Container Service	制限: リスト, 読み込み, 書き込み	複数	なし

\* 必須

キャンセル 戻る **ポリシーの作成**

6. ロール作成の画面に、ブラウザの別タブで戻り、先ほど作成したポリシーを選び、[次のステップ: タグ]をおします。(ポリシーが表示されない場合は、画面右の丸井→マークをおしてみてください)

## ロールの作成


1 2 3 4

### ▼ Attach アクセス権限ポリシー

新しいロールにアタッチするポリシーを1つ以上選択します。

ポリシーの作成 

ポリシーのフィルタ  2件の結果を表示中

	ポリシー名 ▼	次として使用
<input type="checkbox"/>	 AmazonFISServiceRolePolicy	Permissions policy (1)
<input checked="" type="checkbox"/>	fispolicy	なし

- タグは空欄のまま[次のステップ:確認]をおします
- ロール名に、[fisrole]と入力し[ロールの作成]をおします

## ロールの作成

1 2 3 4

### 確認

以下に必要な情報を指定してこのロールを見直してから、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。

信頼されたエンティティ AWS のサービス: ec2.amazonaws.com

ポリシー [fispolicy](#) 

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

- 作成されたロールを再度クリックし、[信頼関係]のタブを選びます

アクセス権限 信頼関係 タグ アクセスアドバイザー セッションの無効化

ロールと、ロールのアクセス条件を引き受けることができる信頼されたエンティティを表示できます。 [ポリシードキュメントの表示](#)

**信頼関係の編集**

**信頼されたエンティティ**

以下の信頼されたエンティティでは、このロールを引き受けることができます。

**条件**

以下の条件では、信頼されたエンティティがロールを引き受ける方法とタイミングを定義します。  
このロールに関連付けられている条件はありません。

**信頼されたエンティティ**

ID プロバイダー ec2.amazonaws.com

10. [信頼関係の編集]を押し、[relation.txt]の中身に置換し、[信頼ポリシーの更新]をおします

### 信頼関係の編集

以下のアクセスコントロールポリシードキュメントを編集して、信頼関係をカスタマイズできます。

ポリシードキュメント

```
1 {
2   "Version": "2012-10-17",
3   "Statement": [
4     {
5       "Effect": "Allow",
6       "Principal": {
7         "Service": "fis.amazonaws.com"
8       },
9       "Action": "sts:AssumeRole",
10      "Condition": {}
11    }
12  ]
13 }
```

キャンセル 信頼ポリシーの更新

11. 信頼されたエンティティが ec2 から fis に代わっていれば成功です

### 信頼されたエンティティ

ID プロバイダー fis.amazonaws.com

### [テスト用 EC2 の作成]

本シナリオでは、EC2 へのログイン、EC2 への通信確認などは行わず、生存確認のみを行います。このため、ネットワーク設定 (VPC, Subnet, Security Group, IP アドレス等)、SSH に用いる鍵、インスタンスタイプなどは全て適当に設定をお願いします。

12. EC2 インスタンスを二つ起動します。

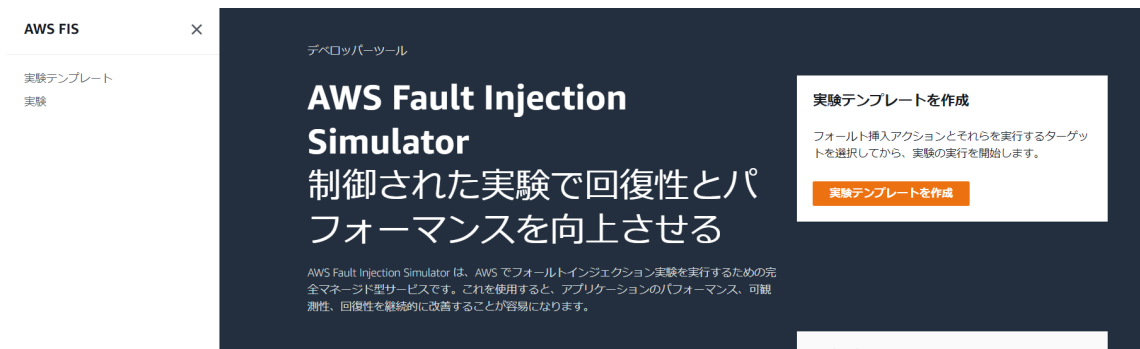
1 台は Amazon Linux2 を起動し、Name タグに[Linux]と設定し、もう 1 台は Windows を起動し、Name タグに[Windows]と設定してください。

以下の様に Windows と Linux が 1 台筒起動する状態になれば OK です。

<input type="checkbox"/>	Windows	i-01bb269402a613373	✔ 実行中	🔍	t2.micro	✔ 2/2 のチェックに: アラーム...	+	ap-northeast-1a
<input type="checkbox"/>	Linux	i-067cdeca684998ff6	✔ 実行中	🔍	t2.micro	✔ 2/2 のチェックに: アラーム...	+	ap-northeast-1a

## [FIS の実験設定と実験開始]

13. ブラウザの別タブで FIS のマネージメントコンソールにアクセスします



14. [実験テンプレートを作成]をおし、説明に[linuxshutdown]と入力し、先程作成した IAM ロールを選びます

### 実験テンプレートを作成 情報

#### 説明とアクセス許可

**説明**

実験の説明を追加します。

最大 512 文字の説明を入力します。

**IAM ロール**

IAM ロールを選択して、実験を実行するためのアクセス許可を付与します。 [詳細はこちら](#)

▼

15. [ターゲットの追加]をおします（アクションの追加が画面上先に表示されていますが、それは後で設定を行いますので、先にその下にあるターゲットを設定します）
16. 名前に[linux]、リソースタイプを[awaws:ec2:instance]、ターゲットメソッドを[リソース ID]とし、先程作成した Linux の EC2 インスタンス ID を選びます。選択モードは[すべて]を選び、[保存]をおします

ターゲットを追加

選択したアクションを実行するターゲットリソースを指定します。 [詳細はこちら](#)

名前

linux

リソースタイプ

aws:ec2:instance

ターゲットメソッド

☒ リソース ID

☐ リソースタグとフィルター

リソース ID

リソース ID を選択してください

Q

☐ i-0db3b29a746ea50cb

☐ i-01bb269402a613373

☒ i-067cdeca684998ff6

選択モード

すべて

キャンセル

保存

17. 以下のように実験を行うターゲットがセットされていれば成功です

ターゲット (1)

選択したアクションを実行するターゲットリソースを指定します。

▶ linux (aws:ec2:instance)

編集

削除

ターゲットを追加

18. 次に、ターゲットに対するアクションを設定するため、画面上の[アクションを追加]をおします
19. 名前に[linuxshutdown]、アクションタイプに[aws:ec2:stop-instances]、ターゲットに[linux]を選択します。アクションパラメータに[1]と入力します。(2分後にインスタンスを自動で再起動させます)

▼ 新しいアクション

保存

削除

名前

linuxshutdown

説明 - オプション

アクションタイプ

ターゲットで実行するアクションタイプを選択します。 [詳細はこちら](#)

aws:ec2:stop-instances ▼

次のあと開始 - オプション

このアクションの前に実行するアクションを選択します。それ以外の場合、このアクションは実験が開始するとすぐに実行されます。

アクションを選択 ▼

ターゲット

まだターゲットが存在しない場合は、このアクションのターゲットが自動的に作成されます。追加のターゲットは以下で作成できます。

linux ▼

アクションパラメータ

このアクションのパラメータ値を指定します。 [詳細はこちら](#)

startInstancesAfterDuration - オプション

The time to wait before restarting the instances (ISO 8601 duration).

20. [保存]をおします

アクション (1)

ターゲットリソースで実行する 1 つ以上のアクションを指定します。各アクションの実行時間 (分単位) と、実験中にアクションを開始するタイミングを決定します。 [詳細はこちら](#)

▶ linuxshutdown / aws:ec2:stop-instances

編集

削除

開始: 実験の開始時 / ターゲット: linux

アクションを追加

ターゲット (1)

選択したアクションを実行するターゲットリソースを指定します。

▶ linux (aws:ec2:instance)

編集

削除

アクション: linuxshutdown

ターゲットを追加

21. これで実験で行うアクションとそのターゲットが設定されました。画面下[実験テンプ



レートを作成]をおします

22. 確認が求められますので[作成]と入力し再度[実験テンプレートを作成]をおします。  
以下のような画面が表示されれば作成完了です。

AWS FIS > 実験テンプレート > EXT3smMqgZCyCn1N

EXT3smMqgZCyCn1N / linuxshutdown 情報 アクション ▼

詳細

実験テンプレート ID EXT3smMqgZCyCn1N	説明 linuxshutdown	IAM ロール fisrole <a href="#">🔗</a>
停止条件 -	作成時刻 May 13, 2021, 13:10:55 (UTC+09:00)	最終更新時刻 May 13, 2021, 13:12:02 (UTC+09:00)

23. 画面左のペインから[実験テンプレート]を選び、アクションのドロップダウンリストから[実験を開始]を選びます

## 実験を開始 EXT3smMqgZCyCn1N

現在の実験状態とテンプレートのスナップショットを使用して実験を開始します。実験が開始された後、この実験に影響を与えずに実験テンプレートに変更を加えることができます。

### タグを実験

この実験にタグを関連付けます。

このリソースに関連付けられたタグはありません

新しいタグを追加

さらに 50 の タグ を追加できます

キャンセル 実験を開始

24. ブラウザの別のタブで先程の EC2 が確認できるようにした状態で、[実験を開始]ボタンをおします。  
追加のダイアログで[確認]と入力し実験を開始します

EXPqsoVrP7gPuaRgdF 情報 更新 アクション ▼

詳細

実験 ID EXPqsoVrP7gPuaRgdF	開始時刻 May 13, 2021, 13:21:42 (UTC+09:00)	状態 <a href="#">🔄 Initiating</a>	実験テンプレート ID EXT3smMqgZCyCn1N
作成時刻 May 13, 2021, 13:21:41 (UTC+09:00)	終了時刻 -	IAM ロール fisrole <a href="#">🔗</a>	停止条件 -

25. Linux EC2 インスタンスが停止され、2 分待つと復旧してきます

<input type="checkbox"/>	Windows	i-01bb269402a613373	🟢 実行中	🔍🔍	t2.micro
<input type="checkbox"/>	Linux	i-067cdeca684998ff6	🟡 停止中	🔍🔍	t2.micro

(2 分後)

<input type="checkbox"/>	Windows	i-01bb269402a613373	🟢 実行中	🔍🔍	t2.micro
<input type="checkbox"/>	Linux	i-067cdeca684998ff6	🟢 実行中	🔍🔍	t2.micro

26. 今度は同じ要領で、実験テンプレートを再度作成し[WindowsTerminate]という名前にします。今度は特定 EC2 インスタンスを設定せずタグベースで実験を行いますので、ターゲットを以下のように設定します

ターゲットを追加

×

選択したアクションを実行するターゲットリソースを指定します。 [詳細はこちら](#)

名前

リソースタイプ

Windowsterminate

aws:ec2:instance ▼

ターゲットメソッド

☐ リソース ID

☒ リソースタグとフィルター

選択モード

すべて ▼

リソースタグ

キー

値 - オプション

Name

Windows

削除

新しいタグを追加

リソースフィルター - オプション

指定した属性でリソースをフィルタリングします。 [詳細はこちら](#)

ターゲットに関連付けられたリソースフィルターはありません。

新しいフィルターを追加

キャンセル

保存

この設定では Name タグに Windows とついているインスタンス全てが対象となります。商用環境の場合例えば以下のように 10 台のうち 1 台のみを実験対象にするなどの制御が可能です

● リソースタグとフィルター

選択モード パーセンテージ (%)

パーセント 10

リソースタグ

キー 値 - オプション

Name Windows 削除

27. アクションでは先ほどと異なり `aws:ec2:terminate-instances` を指定します

▼ 新しいアクション 保存 削除

名前 説明 - オプション

Windowsterminate

アクションタイプ 次のあと開始 - オプション

ターゲットで実行するアクションタイプを選択します。 [詳細はこちら](#) このアクションの前に実行するアクションを選択します。それ以外の場合、このアクションは実験が開始するとすぐに実行されます。

aws:ec2:terminate-instances アクションを選択

ターゲット

まだターゲットが存在しない場合は、このアクションのターゲットが自動的に作成されます。追加のターゲットは以下で作成できます。

Windowsterminate

28. 作成されたテンプレートを用いて実験を開始すると今度は Windows インスタンスが終了されます

<input type="checkbox"/>	Windows	i-01bb269402a613373	終了済み	t2.micro
<input type="checkbox"/>	Linux	i-067cdeca684998ff6	実行中	t2.micro

[Auto Healing (静的構成の Auto Scaling 設定)]

次のステップでは、EC2 インスタンスに障害が発生した際に、自動で指定した台数まで EC2 インスタンス台数を復旧させる設定を行った後、FIS でインスタンスを終了し、正しく設定が動作しているかを確認します

29. 先程の EC2 linux インスタンスを選んで[アクション]→[イメージとテンプレート]→[インスタンスからテンプレートを作成]を指定します

30. 起動テンプレート名、テンプレートバージョンの説明に[autohealinglinux]と入力し、その他設定はデフォルトのままとし、画面下[起動テンプレートを作成]をおします

31. EC2 マネージメントコンソールの左ペイン一番下から[Auto Scaling グループ]を選びます



32. [Auto Scaling グループの作成]をおします。Auto Scaling グループ名に [linuxautohealing] と入力し、起動テンプレートに先程作成したものを指定し、[次へ]をおします

**起動テンプレートまたは起動設定を選択する** 情報

この Auto Scaling グループによって起動されたすべての EC2 インスタンスに共通の設定が含まれる起動テンプレートを指定します。現在起動設定を使用している場合は、起動テンプレートへの移行を検討してください。

**名前**

Auto Scaling グループ名  
グループを識別する名前を入力します。

現在のリージョンにあるこのアカウントに固有で、255 文字以内にする必要があります。

**起動テンプレート** 情報 起動設定に切り替える

起動テンプレート  
Amazon マシンイメージ (AMI)、インスタンスタイプ、キーペア、セキュリティグループなど、インスタンスレベルの設定を含む起動テンプレートを選択します。

▼ 

[起動テンプレートを作成する](#) 

33. ネットワークのサブネットを適当に選び、(今回は 1 台構成なのでどこでも問題ありません) [次へ]をおします
34. ヘルスチェックは[60]に変更し、[次へ]をおします

### ヘルスチェック - 省略可能

#### ヘルスチェックのタイプ 情報

EC2 Auto Scaling は、ヘルスチェックに合格しなかったインスタンスを自動的に置き換えます。ロードバランシングを有効にした場合、常に有効になっている EC2 ヘルスチェックに加えて、ELB ヘルスチェックを有効にすることができます。

☒ EC2 ☐ ELB

#### ヘルスチェックの猶予期間

新しいインスタンスの運用が開始されてから、EC2 Auto Scaling が最初のインスタンスのヘルスチェックを実行するまでの時間です。

秒

35. 次の画面では維持台数の設定が出ますが、デフォルトの 1 台構成のまま進めますので、[次へ]をおします。さらにその次の画面、そしてその次のタグ設定の画面でも[次へ]をおします。

### グループサイズ - 省略可能 情報

希望するキャパシティを変更して Auto Scaling グループのサイズを指定します。キャパシティの下限と上限を指定することもできます。希望するキャパシティは制限範囲内である必要があります。

希望する容量

最小キャパシティ

最大キャパシティ

### 通知を追加 情報

Amazon EC2 Auto Scaling が Auto Scaling グループ内の EC2 インスタンスを起動または終了するたびに、SNS トピックに通知を送信します。

### タグ (0)

50 残り

36. 最後の確認画面で[Auto Scaling グループを作成]をおします

37. 以下のように EC2 インスタンスが 3 台存在しています。Auto Scaling で 1 台構成が保持されている Linux、今日のハンズオンで起動した Linux、すでに FIS 経由で終了済の Windows です。Auto Scaling は 30 秒間隔で EC 2 を維持するため、FIS で 2 台の Linux を終了させると、30 秒後に 1 台復旧してきます。

<input type="checkbox"/>	Linux	i-06048bb27cc216e7c	🟢 実行中	🔍🔍	t2.micro	🟢 2/2 のチェックに	アラーム...	+	ap-northeast-1c
<input type="checkbox"/>	Windows	i-01bb269402a613373	🔴 終了済み	🔍🔍	t2.micro	-	アラーム...	+	ap-northeast-1a
<input type="checkbox"/>	Linux	i-067cdeca684998ff6	🟢 実行中	🔍🔍	t2.micro	🟢 2/2 のチェックに	アラーム...	+	ap-northeast-1a

38. 先程と同じ手順で今度は Name タグに Linux を持つ EC2 インスタンスを全て Terminate する実験テンプレートを作成し、実験を開始します。名前は [linuxterminate] で作成してください
39. 以下のようにシャットダウン状態になり、30 秒待つと 1 台構成が Auto Scaling により復旧してきます

<input type="checkbox"/>	Linux	i-06048bb27cc216e7c	🔴 シャットダウン	🔍🔍	t2.micro				
<input type="checkbox"/>	Windows	i-01bb269402a613373	🔴 終了済み	🔍🔍	t2.micro				
<input type="checkbox"/>	Linux	i-067cdeca684998ff6	🔴 シャットダウン	🔍🔍	t2.micro				

(30 秒後)

<input type="checkbox"/>	Linux	i-06048bb27cc216e7c	🔴 終了済み	🔍🔍	t2.micro				
<input type="checkbox"/>	Linux	i-0e8b513e3b3438a43	🟢 実行中	🔍🔍	t2.micro				
<input type="checkbox"/>	Windows	i-01bb269402a613373	🔴 終了済み	🔍🔍	t2.micro				
<input type="checkbox"/>	Linux	i-067cdeca684998ff6	🔴 終了済み	🔍🔍	t2.micro				

おつかれさまでした！以下を削除してください

Auto Scaling グループ

FIS 実験テンプレート（実験結果は削除できませんが課金に影響を与えません）

IAM ロールとポリシー

EC2 インスタンスは全て終了されているはずですが、念のため確認してください