報道各位

2020 年 8 月 26 日
株式会社インプレスR&D
https://nextpublishing.jp/

Terraform 初心者のためのチュートリアル！

『はじめての人のための Terraform for AWS』発行

技術の泉シリーズ、8月の新刊

インプレスグループで電子出版事業を手がける株式会社インプレス R&D は、『はじめての人のための Terraform for AWS』(著者:morudara、buchios)を発行いたします。

最新の知見を発信する『技術の泉シリーズ』は、「技術書典」や「技術書同人誌博覧会」をはじめとした各種即売会や、勉強会・LT 会などで頒布された技術同人誌を底本とした商業書籍を刊行し、技術同人誌の普及と発展に貢献することを目指します。

『はじめての人のためのTerraform for AWS』
https://nextpublishing.jp/isbn/9784844378990

著者:morudara、buchios
小売希望価格:電子書籍版　1600 円(税別)／印刷書籍版　2000 円(税別)
電子書籍版フォーマット:EPUB3／Kindle Format8
印刷書籍版仕様:B5 判／カラー／本文 126 ページ
ISBN:978-4-8443-7899-0
発行:インプレス R&D

＜＜発行主旨・内容紹介＞＞

AWS(Amazon Web Service)の環境構築をTerraformで行う、初心者向けの解説本です。はじめての方でもわかりやすく理解できるよう、チュートリアル形式で紹介しています。

本書を読むことで、AWS の主要なサービスと Terraform についての基礎的な知識や Terraform を用いた AWS 環境の構築技術について学ぶことができます。

(本書は、次世代出版メソッド「NextPublishing」を使用し、出版されています。)

Terraform をつかった AWS 環境構築の準備を基本から解説

図 1.6: タグの追加

図 1.7: ユーザーの作成

図 1.8: 作成確認と csv のダウンロード

1.2　Terraformを準備しよう

Terraformのセットアップ

tfenvの利用

Terraformはバージョンの変化が早いツールです。作業端末に直接Terraformをインストールしてしまうとバージョンの変化に対応しにくいため、tfenvを利用します。tfenvはTerraformのバージョン管理を簡単にできるツールです。

tfenvのインストール作業は、PCにバイナリインストールする方法とHomebrewを利用する方法があります。本書では、Homebrewでインストールします。

brewコマンドでインストールをしましょう。なお、先にTerraformがインストールされているとインストールができませんので、アンインストールしておきます。

```
$ brew uninstall terraform
$ brew install tfenv
```

インストールが正常に完了したかどうかを確認しましょう。以下のようにhelpの結果が返ってくれば問題ありません。

Web サーバーや DB サーバーなどを作りながら、実際の機能ごとに解説

図 4.2: パラメーターグループ_作成後

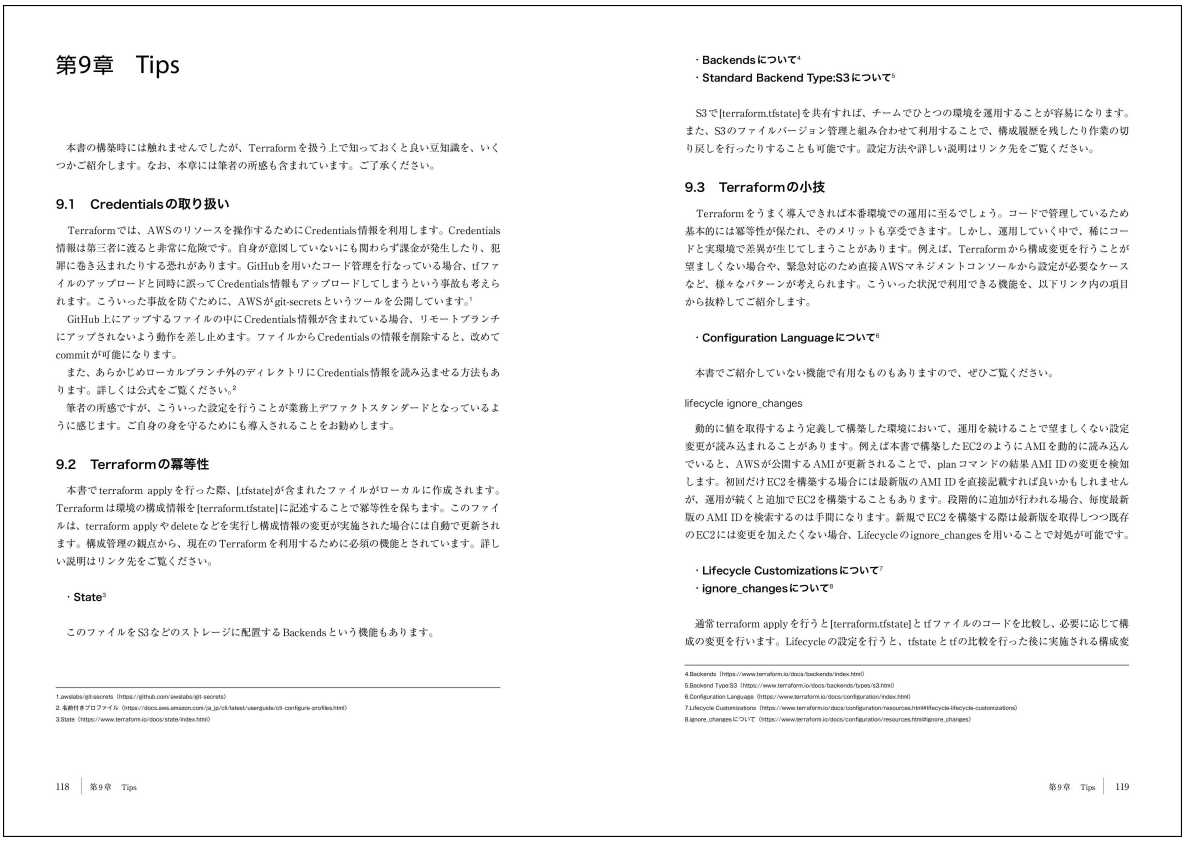
4.6　RDS用セキュリティグループの構築

まずはセキュリティグループを構築します。作成済みの[vpc_sg.tf]に追記してください。

vpc_sg.tf

```
################################################################################
# RDSがAPサーバーから3306ポートを利用した通信を受け入れるSG設定
################################################################################

# RDSがAPサーバーから3306ポートを利用した通信を受け付けるSGの構築
resource "aws_security_group" "rds_sg" {

  # セキュリティグループ名を設定
  name   = "rds-sg"

  # セキュリティグループを構築するVPCのIDを設定
  vpc_id = aws_vpc.vpc.id

  # タグを設定
  tags = {
    Name = "rds-sg"
  }

}

# 出て行く通信の設定
resource "aws_security_group_rule" "egress_rds_sg" {

  # このリソースが通信の出て行く先を設定することを定義
  # egressを設定
  type              = "egress"

  # ポートの範囲設定
  # 全てのトラフィックを許可する場合いずれも0で設定
  from_port         = 0
  to_port           = 0

  # プロトコル設定
  # 以下は全てのIPv4トラフィックを許容する設定
  protocol          = "-1"

  # 許可するIPの範囲を設定
  # 以下は全てのIPv4トラフィックを許容する設定
  cidr_blocks       = ["0.0.0.0/0"]

  # このルールを付与するセキュリティグループを設定
  security_group_id = aws_security_group.rds_sg.id
}

# 3306ポートを受け入れる設定
resource "aws_security_group_rule" "ingress_rds_3306" {

  # このリソースが通信を受け入れる設定であることを定義
  # ingressを設定
  type              = "ingress"

  # ポートの範囲設定
  # 今回利用するAmazon Aurora MySQLはデフォルトで3306
  # 3306のみ利用するよう、from_portとto_portに記述
  from_port         = "3306"
  to_port           = "3306"

  # プロトコルはtcpを設定
  protocol          = "tcp"

  # 許可するIPの範囲を設定
  # Webサーバーを配置しているサブネットのCIDRを設定
  cidr_blocks       = ["10.0.2.0/24"]

  # このルールを付与するセキュリティグループを設定
  security_group_id = aws_security_group.rds_sg.id
```

実際に活用する際の Tips も掲載



第9章　Tips

本書の構築時には触れませんでしたが、Terraformを扱う上で知っておくと良い豆知識を、いくつかご紹介します。なお、本章には筆者の所感も含まれています。ご了承ください。

9.1　Credentialsの取り扱い

Terraformでは、AWSのリソースを操作するためにCredentials情報を利用します。Credentials情報は第三者に渡ると非常に危険です。自身が意図していないにも関わらず課金が発生したり、犯罪に巻き込まれたりする恐れがあります。GitHubを用いたコード管理を行なっている場合、tfファイルのアップロードと同時に誤ってCredentials情報もアップロードしてしまうという事故も考えられます。こういった事故を防ぐために、AWSがgit-secretsというツールを公開しています。[1]

GitHub上にアップするファイルの中にCredentials情報が含まれている場合、リモートブランチにアップされないよう動作を差し止めます。ファイルからCredentialsの情報を削除すると、改めてcommitが可能になります。

また、あらかじめローカルブランチ外のディレクトリにCredentials情報を読み込ませる方法もあります。詳しくは公式をご覧ください。[2]

筆者の所感ですが、こういった設定を行うことが業務上デファクトスタンダードとなっているように感じます。ご自身の身を守るためにも導入されることをお勧めします。

9.2　Terraformの冪等性

本書でterraform applyを行った際、[.tfstate]が含まれたファイルがローカルに作成されます。Terraformは環境の構成情報を[terraform.tfstate]に記述することで冪等性を保ちます。このファイルは、terraform applyやdeleteなどを実行し構成情報の変更が実施された場合には自動で更新されます。構成管理の観点から、現在のTerraformを利用するために必須の機能とされています。詳しい説明はリンク先をご覧ください。

・State[3]

このファイルをS3などのストレージに配置するBackendsという機能もあります。

・Backendsについて[4]
・Standard Backend Type:S3について[5]

S3で[terraform.tfstate]を共有すれば、チームでひとつの環境を運用することが容易になります。また、S3のファイルバージョン管理と組み合わせて利用することで、構成履歴を残したり作業の切り戻しを行ったりすることも可能です。設定方法や詳しい説明はリンク先をご覧ください。

9.3　Terraformの小技

Terraformをうまく導入できれば本番環境での運用に至るでしょう。コードで管理しているため基本的には冪等性が保たれ、そのメリットも享受できます。しかし、運用していく中で、稀にコードと実環境で差異が生じてしまうことがあります。例えば、Terraformから構成変更を行うことが望ましくない場合や、緊急対応のため直接AWSマネジメントコンソールから設定が必要なケースなど、様々なパターンが考えられます。こういった状況で利用できる機能を、以下リンク内の項目から抜粋してご紹介します。

・Configuration Languageについて[6]

本書でご紹介していない機能で有用なものもありますので、ぜひご覧ください。

lifecycle ignore_changes

動的に値を取得するよう定義して構築した環境において、運用を続けることで望ましくない設定変更が読み込まれることがあります。例えば本書で構築したEC2のようにAMIを動的に読み込んでいると、AWSが公開するAMIが更新されることで、planコマンドの結果AMI IDの変更を検知します。初回だけEC2を構築する場合には最新版のAMI IDを直接記載すれば良いかもしれませんが、運用が続くと追加でEC2を構築することもあります。段階的に追加が行われる場合、毎度最新版のAMI IDを検索するのは手間になります。新規でEC2を構築する際は最新版を取得しつつ既存のEC2には変更を加えたくない場合、Lifecycleのignore_changesを用いることで対処が可能です。

・Lifecycle Customizationsについて[7]
・ignore_changesについて[8]

通常terraform applyを行うと[terraform.tfstate]とtfファイルのコードを比較し、必要に応じて構成の変更を行います。Lifecycleの設定を行うと、tfstateとtfの比較を行った後に実施される構成変

1.awslabs/git-secrets（https://github.com/awslabs/git-secrets）
2.名前付きプロファイル（https://docs.aws.amazon.com/ja_jp/cli/latest/userguide/cli-configure-profiles.html）
3.State（https://www.terraform.io/docs/state/index.html）
4.Backends（https://www.terraform.io/docs/backends/index.html）
5.Backend Type:S3（https://www.terraform.io/docs/backends/types/s3.html）
6.Configuration Language（https://www.terraform.io/docs/configuration/index.html）
7.Lifecycle Customizations（https://www.terraform.io/docs/configuration/resources.html#lifecycle-lifecycle-customizations）
8.ignore_changesについて（https://www.terraform.io/docs/configuration/resources.html#ignore_changes）

118 | 第9章　Tips

第9章　Tips | 119

＜＜目次＞＞

＜＜著者紹介＞＞

morudara

企画からフロントとサーバーサイドとインフラと幅広く担当するエンジニア。

最近はReactとRuby on Railsでの開発業務がメイン。クラウドはAWSとGCPとOpenStackの業務経験があり、ネットワーク/サーバー設計から構築運用まで幅広く手がけた経験も有る。

Twitter @morudara

buchios

AWSとIaCが得意なインフラエンジニア。最近はKubernetesの業務がメイン。

オンプレとクラウドの両方で、サーバー設計から構築運用まで幅広く手がけた経験も有る。
Twitter @buchios

＜＜販売ストア＞＞

電子書籍:

Amazon Kindle ストア、楽天 kobo イーブックストア、Apple Books、紀伊國屋書店 Kinoppy、Google Play Store、honto 電子書籍ストア、Sony Reader Store、BookLive!、BOOK☆WALKER

印刷書籍:

Amazon.co.jp、三省堂書店オンデマンド、honto ネットストア、楽天ブックス

※　各ストアでの販売は準備が整いしだい開始されます。
※　全国の一般書店からもご注文いただけます。

【インプレス R&D】 https://nextpublishing.jp/

株式会社インプレスR&D（本社:東京都千代田区、代表取締役社長:井芹昌信）は、デジタルファーストの次世代型電子出版プラットフォーム「NextPublishing」を運営する企業です。また自らも、NextPublishing を使った「インターネット白書」の出版など IT 関連メディア事業を展開しています。

※NextPublishing は、インプレス R&D が開発した電子出版プラットフォーム（またはメソッド）の名称です。電子書籍と印刷書籍の同時制作、プリント・オンデマンド（POD）による品切れ解消などの伝統的出版の課題を解決しています。これにより、伝統的出版では経済的に困難な多品種少部数の出版を可能にし、優秀な個人や組織が持つ多様な知の流通を目指しています。

【インプレスグループ】 https://www.impressholdings.com/

株式会社インプレスホールディングス（本社:東京都千代田区、代表取締役:松本大輔、証券コード:東証 1 部 9479）を持株会社とするメディアグループ。「IT」「音楽」「デザイン」「山岳・自然」「モバイルサービス」「学術・理工学」「旅・鉄道」を主要テーマに専門性の高いメディア&サービスおよびソリューション事業を展開しています。さらに、コンテンツビジネスのプラットフォーム開発・運営も手がけています。

【お問い合わせ先】

株式会社インプレス R&D NextPublishing センター
TEL 03-6837-4820
電子メール: np-info@impress.co.jp