

この度は、AWSを利用したWordpressの作成方法の動画 (https://youtu.be/JQS5libJqDw)ならびに当資料を ダウンロード頂き誠にありがとうございます。

ネットビジョンシステムズ株式会社のMiyabiです。

当PDFは、動画内で使用されたコマンドならびに手順を解説しております。 AWS(Amazon Web Sercvice)のご利用のお役に立てれば光栄です。

また、弊社は次世代のネットワークエンジニア、サーバーエンジニアを多数輩出する企業 でもございます。

> 未経験者からエンジニアになりたい! 自分にもできる、将来性のある仕事をしたい! 手に職をつけたい。 自分になにが出来るのかまだ分からない

> > そんな方々をお待ちしております。

【ネットビジョンシステムズ株式会社採用サイト】

http://www.netvisionsystems.jp/

パソコン苦手だしな。。。。エンジニアってなんか難しそう。。。

そんな悩みを解決すべく、本年2016年8月

無料で通えるエンジニア育成学校を設立致しました。

【ネットビジョンアカデミー】

http://www.netvisionacademy.jp/

ご興味のある方 お話だけでも聞きたい方 すぐにでもエンジニアを目指して、時代の波に乗りたい方 どしどしご連絡ください。

【ネットビジョンシステムズ株式会社】

http://www.netvisionsystems.biz/

[Facebook]

https://www.facebook.com/netvisionsystems/

AWS クラウド 無料利用枠

アマゾン ウェブ サービス (AWS) 無料利用枠では、実際の環境の AWS クラウドサービスを無料でお使いいただけます。 AWS 無料利用枠には、AWS にサインアップした日から 12 ヶ月間お使いいただける無料利用枠が付いたサービスと、12 ヶ月間の無料利用期間終了後にも自動的に期限切れにならない追加サービスが提供されています。

AWS アカウントを作成すると、使用制限の範囲内で次に示す製品およびサービスをすべて無料でご利用いただけます。 以下の手順に従って今すぐ始めると、自動的に AWS 無料利用枠をご利用いただけます。 AWS アカウントにサインアップします。

<u>請求先住所とクレジットカード情報を入力します。無料利用枠の上限を超えない限り、料金が請求されることはありません。</u> 詳細はこちら→ https://aws.amazon.com/jp/free/

Amazon web serviceより引用

AWS 各サービス概要説明

EC2 (Amazon Elastic Compute Cloud)

クラウド上で、インスタンスという仮想コンピューティング環境(自由に拡張と縮小ができる)を利用することができるWEBサーバ、DBサーバ、キャッシュサーバーなど様々

EBS (Amazon Elastic Block Store)

EC2からマウントして利用出来るストレージサービス 高い耐久性と可用性

S3 (Amazon Simple Storage Service)

データの格納と取り出しに特化したデータストレージサービス スケーラビリティに優れ、高い安全性と耐久性

VPC (Amazon Virtual Private Cloud)

安全に構築・運用できるAWSアカウント専用の仮想ネットワーク サブネット単位・ホスト単位での柔軟なアクセス制御ができ、パブリック・プライベートサブネットの切り分けが可能

RDS (Amazon Relational Database Service)

クラウド上で利用できるリレーショナルデータベースサービス Amazon Aurora、Oracle、Microsoft SQL Server、PostgreSQL、MySQL、MariaDBから選ぶことが可能

Route53

可用性と拡張性に優れ、世界的に分散されているDNSウェブサービス 通常のDNSとは違い4つのネームサーバーで構成される

ELB (Elastic Load Balancing)

アプリケーションへのトラフィックを複数のEC2インスタンスに自動分散するクラウドロードバランサーロードバランサーとコントローラーサービスの2つのコンポーネントで構成される

IAM (Identity and Access Management)

AWSへのアクセスを安全に制御できる権限管理のサービス 1つのアカウントの中で複数の役割を持ったアカウントを作ることができる

Amazon CloudWatch

AWSモニタリングサービス

AWS上のリソース、カスタムメトリックス、アプリケーションが生成するあらゆるログファイル等をモニタリング可能

VPC領域の作成

0【AWS新規登録】



https://aws.amazon.com/jp/

{AWS TOP画面}→{サインアップ}→{登録用アドレス入力}→{私は新規ユーザー}→すべて入力に登録を完了させる。 ※下記URL参照

https://aws.amazon.com/jp/register-flow/

1【サインイン】

https://aws.amazon.com/ip/

{AWS TOP画面}→{サインアップ}→{サインイン(登録済みは既存を選択)}

2【リージョン(設置地域)設定】

右上のサポートの左の{地域/国]をクリック]→{東京}を選択 ※この設定をリージョン設定という。(下の画像では既に東京になってます) ※使いたい国のサーバーを選ぶ事ができる。

■ AWS - サービス - 編集 · honda - 東京 - サポート -

3【VPC領域の作成】

※安全に構築・運用できるAWSアカウント専用の仮想ネットワークホーム(左上のオレンジの四角い箱の列にある){サービス}→{ネットワーキング}→{VPC}を選択



左側VPCダッシュボードー覧より{VPC}を選択 ↓ 画面中央にある{VPCの作成}を選択

{ネームタグ}にVPC領域に付ける名前を入力(ここでは{VPC領域}とする)
{CIDR ブロック} は使用するIPアドレス範囲を表しており{10.0.0.0/16}を指定しテナンシーは{デフォルト}を選択
※下記画像参照

VPC 0)作成		×
		タンスなどの AWS オブジェクトによって	
		ルーティング (CIDR) のブロック形式を使 」ます。/16 より大きい VPC を作成するこ	
	0.0.0.0/16)を指定し	よす。/16 より大きい VPC を作成するこ	とはできません。

→{作成}でVPC領域が作られる。

4【パグリックサブネットを作る】

左側VPCダッシュボードー覧の{サブネット}→{サブネットの作成}を選択



{ネームタグ}を設定する(ここでは{パブリックサブネット}とする) {VPC}は3で作ったものをタブで指定、アベイラビリティーゾーンはそのまま {CIDRブロック}を(10.0.1.0/24)で設定

1



→{作成}

5【インターネットゲートウェイを作る】

※VPC 内のリソースと インターネットとの通信に使われるコンポーネント

左側VPCダッシュボードー覧の{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}を選択→{任意の名前を入力(ここでは空欄)}→{作成}を選択でインターネットゲートウェイが作成される。

5で作ったインターネットゲートウェイに[チェックを入れ]→上段の {VPCにアタッチ}を選択 結びつける領域(3で作ったVPCを選択)→「アタッチ



※これでインターネットゲートウェイが結び付けられる。

6【ルートテーブルを確認】

左側VPCダッシュボードー覧の{サブネット}→4で作成した{パブリックサブネット}を選択 →下段にある概要右側に記載の{ルートテーブル:****}を確認



7【ルートテーブルの設定値を確認】

左側VPCダッシュボード一覧より{ルートテーブル}→6で確認した{ルートテーブル:***}と同じ(サブネットID)を選択し設定
→下段{ルート}のタブを選択し設定値



8【パブリックルートテーブルの作成】

7の画面のまま上段にある{ルートテーブルの作成}を選択

ルートテーブルの名前を入力する(ここでは{パブリックルートテーブル}とする)

作成先の{VPCを選択}(3で作成したVPC領域を選択)



→{作成}

9【ルートテーブルをサブネットに割り当てる】

8で作成した{パブリックルートテーブル}を選択し、下段の{サブネットの関連付け}→{編集}を選択サブネットー覧が表示されるので、割り当てたいサブネット(4で作成{パブリックサブネット))を選択し→{保存}



10【デフォルトゲートウェイをインターネットゲートウェイに設定する】

10の状態のまま下段タブの要約の隣{ルート}→{編集)を選択→{別ルートの追加} →送信先に{0.0.0.0/0}、ターゲットに{インターネットゲートウェイ}を設定(ターゲット欄をマウスでクリックすると下に出てく るので選択)→{保存}

※インターネットゲートウェイの名称「igw-xxxxxx」(xはランダムの値)下記画像参照



左側VPCダッシュボード一覧の{サブネット}→{パブリックサブネット}(4で作成したもの)を選択 →下段の{ルートテーブル}に新しいルートテーブル(10で作成したもの)が表示されているかどうかを確認



ホーム(左上のオレンジの四角い箱の列にある){サービス}を選択→[コンピューティング]→{EC2} リージョンを確認(2を参照)し、左側EC2ダッシュボードのメニューから{インスタンス}を選択→{インスタンスの作成}を選択

インスタンスを起動する際に用いるイメージファイル「AMI」(Amazon Linux AMI 2016.03.3 (HVM), SSD Volume Type)を選択 (下記画像参照) XAMI=Amazon Machine Image

www.marketplace に摘動されているもので選択できます。独自の AMI のいすれかで選択することもできます。 クイックスタート | < 25個のAMI中1~25 > マイ AMI Amazon Linux AMI 2016.03.3 (HVM), SSD Volume Type - ami-374db956 Amazon Linux The Amazon Linux AMI is an EBS-backed, AWS-supported image. The default image includes AWS command line tools, 64ビット AWS Marketplace 無料利用枠の対象 Python, Ruby, Perl, and Java. The repositories include Docker, PHP, MySQL, PostgreSQL, and other packages. ルートデバイスタイプ: ebs 仮想化タイプ: hvm コミュニティ AMI

> 次の画面では{t2.micro}(無料枠)を選択→右下にあるインスタンスの詳細へ進む ※インスタンスタイプ=仮想マシンのスペック

•{ネットワーク}

→VPC領域を選択(VPC作成3で作ったもの) ・{サブネット}を選択→一覧から{パブリックサブネット}を選択(自動で選ばれる)

・{自動割り当てパブリックIP}

→ 「有効」を選択。 ※インターネットからインスタンスにアクセスできるようにするためパブリックIPアドレスを付与する ・{IAMロール}、{シャットダウン動作}、チェック項目変更なし

> ・下段の{ネットワークインターフェイス} →プライマリIPに[10.0.1.10]を入力→[次の手順]を選択

インスタンス数 ①	1 Auto Scaling グループに作成する (i)	プザオフ / / 類片 (J)
購入のオプション (i)	□スポットインスタンスのリクエスト	和鮮保護の有効化 ① 目読った利料から保護します
ネットワーク ①	vpc-10dcc675 (10.0.0.0/16) VPC領域 * subnet-a13de3d7(10.0.1.0/24) パブリックサブネット・* 新しいサブネットの作成	
サブネット ①	Numeral rates ary (10.0.1.024) 「ハンケッグランネッド・・」 利用可能な 251 IP アドレス	専有デナンシーには途域外会が適用されます。
自動割り当てパブリックIP (i)	有効化 ▼	▼ネットワークインターフェイス ®
IAM ロール (i)	なし ▼ C 新しいIAM ロールの作成	デバス キがワーケンターフェイス サブキか プライヤリ P せかがり Pアドレス
シャットダウン動作 (i)	停止 ▼	eth0 新ルキッパフーウイン: submet-a15de3d * 10.0.1.10 Pの論tt
削除保護の有効化 (□誤った削除から保護します	
モニタリング ①	■ CloudWatch 詳細モニタリングを有効化 迫力解す金が適用されます。	

ストレージを設定する(デフォルトのまま) {次の手順}インスタンスタグ付け

インスタンスに名前を設定(ここではwebサーバー)

キー (最大 127 文字)	値 (最大 255 文字)	
Name	WEBサーバー	8
1		
{次の手順:セキュリティグループ	を設定]を選択	
\downarrow		

{セキュリティーグループ名}に任意名前を入力(ここではWEB-SGとする)、送信元はセキュリティーの為、マイIPを指定する。 他の項目はデフォルトのまま

セキュリティグループの割り当て: ●新しいセキュリティグループを作成する ○既存のセキュリティグループを選択する セキュリティグループ名: WEB-SG 説明: launch-wizard-1 created 2016-08-22T13:58:57.928+09:00 ポート範囲 () 送信元 () SSH 任意の場所 ▼ 0.0.0.0/0 ルールの資加 {確認と作成}を選択 1 {作成}を選択 1 {新しいキーペアの作成}を選択し任意の名前(ここではmykey)を入力 [キーペアのダウンロード]を選択しダウンロード(後ほど使うので大切に保管) 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。 キーベアは、AWS が保存するパプリックキーとューザーが保存するプライベーキキーファイルで開航されます。 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 マイルは、インスタンへのログインに使用れる1シワートを設するために必要です。Linux AMI の場合、ブ 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。 : 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 既存のキーペアを削除するの詳細情報をご覧ください。 キーペアのダウンロード 送行するには、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な 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。一度作成されると、ファイルを 再度ダウンロードすることはできなくなります。 キャンセル インスタンスの作成 →{インスタンスの作成}

12【インスタンスの確認】

ホーム(左上のオレンジの四角い箱の列にある){サービス}→{コンピューティング}→{EC2}
リソースの文字の下に[〇個実行中のインスタンス]を選択し以下の状態確認
{インスタンスの状態が「Pending(起動中)」から「running(起動)」になるか確認
その横にある「ステータスチェック」が2/2になれば起動成功



※止め方は、上記画像のrunningの所で{右クリック}→{インスタンス状態}→{停止}が選択できる

13【パブリックIPアドレスを調べる】

12の画面のまま、「パブリック IP」のところに書いてあるパブリックIPをメモ(下の画像とは数字は違う)



14【Tera termで接続する】

Tera termを下記よりインストールし解凍する。 http://forest.watch.impress.co.jp/library/software/utf8teraterm/download_10868.html

起動し「ホスト」部分に13でメモした{パブリックIP}を入力、サービスは{SSH}で{OK}→{続行} ।

ユーザ名(ec2-user)と入力→「RSA/DSA/ECDSA/ED25519鍵を使う」を 選択し→[秘密鍵]を押す。→キーペアファイル(11でダウンロードしたファイルを指定→[OK]

※入力作業が遅いと起動しないので、起動しない場合は再度17からまた、右上の画像のようにすべてのファイルに変更しないと鍵が見つからない

↓ 接続完了



15【サーバーにApacheをインストール】

\$ sudo yum -y install httpd ※コピペは可能だがTeraTermへ貼り付けの際マウスは右クリックで貼り付け可能 また\$は既に入力されているので入力不要

16【Apacheの起動】

上記同様に以下コマンド入力 \$ sudo service httpd start

<u>バーを再起動するとApacheは停止し</u> ようので自動で起動する。

\$ sudo chkconfig httpd on ※chkconfig=自動起動の「設定(on)」「解除(off)」「設定確認(--list)」

18【自動起動(3)がonになっているかの確認】

\$ sudo chkconfig --list httpd

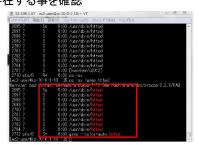
[ec2-user@ip-10-0-1-10 ~]\$ sudo chkconfig --list httpd 6:off httpd 0:off 1:off 2:on 3:on 5:on

※上記画像で3がonであれば自動起動が有効です。

19【Apache実行中のプロセスを確認】

20【ネットワークの待ち受け状態を確認する】

\$ sudo Isof -I -n -P \$ps -ax | grep httpd ※httpdが存在する事を確認



ホーム(左上のオレンジの四角い箱の列にある){サービス}→{すべてのAWSサービス} →{コンピューティング}→{EC2}→{左側EC2ダッシュボード}一覧より{セキュリティーグループ}を選択→{WEB-SG}を選択(11で作成し たもの)

下段の{インバウンド}→{編集}

[ルールの追加]→から[カスタムTCPルール]を選択

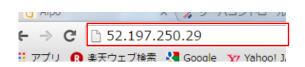
{ポート範囲}に「80」と入力

{送信元}は{マイ IP}を指定→{保存} ※送信元は、下の画像ではカスタム(0.0.0.0/0)になっていますが、マイIPを指定する。



22【APACHE疎通確認】

左側EC2ダッシュボードー覧よりインスタンス→WEBサーバーを選択→パブリックIPをコピー
→そのIPをネット上段に入れ検索(画像左下)すると以下(画像右下)が現れる
※この時のパブリックIPは自身のパブリックIPなので下の画像とIPとは異なる





DBサーバーの構築

23【アベイラビリティーゾーンを確認する。】

ホーム(左上のオレンジの四角い箱の列にある){サービス}→{すべてのAWSサービス}→{ネットワーキング}→{VPC}→左側VPCダッシュボードー覧から{サブネット}を選択→{パブリックサブネット}選択→下段の{概要}タブを選択し{アベイラビリティーゾーン}の文字を確認、メモ

※アベイラビリティゾーン=リージョンを論理的にグループ化したもの



24【プライベートサブネットを作る】

25の画面のまま

上段にある{サブネットの作成}を選択 ネームタグには「プライベートサブネット」を入力 VPCは最初に作ったVPC領域(3参照)を指定。アベイラビリティゾーンは25のものを選択。 CIDRブロックには「10.0.2.0/24」と入力



→{作成}

25【プライベートサブネットにサーバーを構築する。】

ホーム(左上のオレンジの四角い箱の列にある){サービス}→{すべてのAWSサービス} →{コンピューティング}→{EC2}

左側EC2ダッシュボードー覧より{インスタンス}→{インスタンスの作成} {Amazon Linux AMI 2016.03.3 (HVM), SSD Volume Type - ami-374db956]を選択 →{無料利用枠の対象}を選択し{次の手順}

ネットワークでは「VPC領域」を選択し、サブネットには「プライベートサブネット」を設定

作成するインスタンスはインターネットから直接接続させないので 自動割当パブリックIPを「無効」、その他項目はそのまま

下段(ネットワークインターフェイス)の(プライマリIP)は(10.0.2.10)を割り当てる {ストレージの追加は変更なし}



キ ー (最大 127 文字)	値 (最大 255 文字)
Name	DBサーバー
ALCO BET	

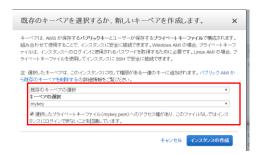
セキュリティーグループは新規、名前、説明は{DB-SG}

セキュリティグループの割り当て:	●新しいセキュリティグループを作成する	
	●既存のセキュリティグループを選択する	
セキュリティグルーブ名:	DB-SG	
説明:	DB-SG	

MySQLの通信を許す場合は通信ポートに「3306」と設定する

{送信元}には「任意の場所」 以下画像のように、{MYSQL/Aurora} を追加し{確認と作成へ}→{作成}→{既存にチェック}





→{インスタンス作成}

※セキュリティグループ=インスタンスに対するファイアウォールの設定

26【pingコマンドで疎通確認できるようにする。】

DBサーバーに対してICMPプロトコルが通るように構成するホーム(左上のオレンジの四角い箱の列にある){サービス}→{すべてのAWSサービス}→{ネットワーキング}→{VPC}→VPCダッシュ

ボードー覧より{セキュリティーグループ}→{DB-SG}を選択 下段メニューより{インバウンドルール}→{編集}

{別のルール追加}

{すべての ICMP}を選択し送信元には{0.0.0.0/0}を指定して保存

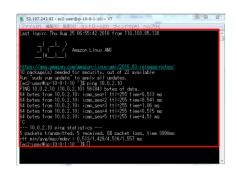


27【疎通を確認する(TeraTerm)】

ホーム(左上のオレンジの四角い箱の列にある){サービス}→{すべてのAWSサービス}
→{コンピューティング}→{EC2}→左側EC2ダッシュボードー覧より{インスタンス}を選択→{WEBサーバー}を選択→画面下段の{説明}
のタブより{パブリックIP}の確認、コピーをする。
コピーしたIPでteraterm|CSSHでログイン(14参照)

↓

teratermで接続したら以下「ping」を実行する(コマンド入力) \$ ping 10.0.2.10 →データの疎通が確認できる。 ※ストップはctr+c



28【DNS名を付ける】

ホーム(左上のオレンジの四角い箱の列にある){サービス}→{すべてのAWSサービス}→{ネットワーキング}→{VPC}→左側VPCダッシュボードの{VPC]を選択→{VPC領域}を選択→{DNSホスト名の編集}→{はい}

※これを設定する事で、パブリックIPで接続することもこのDNS名(アドレスみたいな表記)で指定してログインすることも可能となる。

NATサーバーを構築する

NATサーバー=IPアドレスを変換するサーバーで、パブリックとプライベートの2つのネットワークインターフェイスを持つ

29【NATサーバーで用いるセキュリティグループを作る。】

ホーム(左上のオレンジの四角い箱の列にある){サービス}→{すべてのAWSサービス}→{コンピューティング}→{EC2}→ {EC2ダッシュボード}ー覧より{セキュリティーグループ}を選択

{セキュリティーグループの作成}を選択して新規に{セキュリティグループ]を作成するセキュリティー名に{NAT-SG}と入力、説明は{NAT Segurity Group}と入力、VPCは最初に作ったものを指定

画面下段、{インバウンド} タブから{ルールの追加}を選択してインバウンドのルールを設定する タイプをHTTP(80)とHTTPS(443)、SSH(22)を追加する、送信元は任意の場所 ※下の画像では、送信元はカスタムになっていますが任意の場所を指定

セキュリティグルーブ名 ()		NAT-SG			
	説明(NAT Security Gr	oup		
	VPC (i)	vpc-10dcc675 (10.0.0.0/16) VPC領域	¥	
キュリティグル・	ープのルール:				
Constitution of	and the state of				
インバウンド	アウトバウン	1-			
インハウンド タイプ ①	アクドハウン	プロトコル (i)	ポート範囲(i)	送信元(i)	
	▼ ▼		ポート範囲 (i)	送信元 (i) カスタム ▼ 10.0.2.0/24	6
タイプ(i)		プロトコル (i)			6

同様に{アウトバウンド}のルールを設定する 現在設定されているものは、すべての通信が許可されているので設定を[削除]し タイプをHTTP(80)とHTTPS(443)を追加する、送信元は任意の場所 {保存}ボタンをクリックして作成完了

セキュリティグループ名 (i) 説明 (i)		NAT-SG			
		NAT Security Gr	NAT Security Group		
	VPC (i)	vpc-10dcc675 (10.0.0.0/16) VPC領域	₩	
キュリティグル・	ーブのルール:				
インバウンド	アウトバウン	F			
イブ ①		プロトコル ()	ボー・範囲()	送信先 ①	
HTTP	*	TCP	80	任意の場所 ▼ 0.0.0.0/0	
				任意の場所 ▼ 0.0.0.0/0	

30【NATインスタンスを起動する。】

左側メニューより{インスタンス}を選択 →{インスタンスの作成}を選択

{コミュニティーAMI}タブから{ami-vpc-nat}を検索し、NATのAMIを選択する(最新のもの)

スナツノ T. Amazon マンフィメーン(AIVII)

AMI は、インスタンスの作成に必要なソフトウェア構成(OS、アプリケーションサーバー、アプリケーション)を含むテンプレートです。AMI は、AWS が提供するもの、ユーザーコミュニティが提供するもの、または AWS Marketplace に掲載されているものを選択できます。独自の AMI のいずれかを選択することもできます。



インスタンスの種類を選択しする。

次の画面はそのまま次へ このNATサーバーはパブリックサブネットに配置するため、

ネットワークの選択画面では「VPC領域」を選択し、サブネットとして「パブリックサブネット」を選択また、インターネットと通信するためにパブリックIPアドレスが必要なので{有効}にする



プライベートIP(プライマリID)としては「10.0.1.20」を割り当てる

▼ ネットワークインターフェイス ①

デバイス	ネットワークインターフェイス	サブネット	プライマリ IP	セカンダリ IP アドレス
eth0	新しいネットワークイン:▼	subnet-a13de3d⁻ ▼	10.0.1.20	IPの追加

↓次へ ストレージはデフォルトのまま 値は「NATサーバー」

キー (最大 127 文字)	値 (最大 255 文字)
Name	NATサーバー
タグの作成 (最大 50 タグ)	

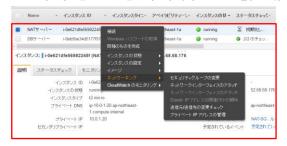
既存を選択→セキュリティグループに「NAT-SG」を選択→{確認と作成}→{作成}→{キーは既存}を選択→{作成}





NATサーバーのインスタンス(送信元/送信先チェック)を無効にする

送信元/送信先チェックを無効にするため、左側EC2メニューより{インスタンス}からNATサーバーのインスタンスIDの上で{右クリック}→{ネットワーク}→{送信元/送信先}の変更チェックを無効}にする



32【NATサーバーのインスタンスID確認する。】

画面中央の{インスタンス}→NATサーバーに設定されたインスタンスIDを確認しておく



33【ルートテーブルを構成する。】

ルートテーブルを構成する為、ホーム(左上のオレンジの四角い箱の列にある){サービス}→{ネットワーキング}→{VPC} →{左側VPCダッシュボード}ー覧より{ルートテーブル}を選択

メインルートテーブルのデフォルトゲートウェイをNATサーバーに変更する為 ※下の画像(黄色いマーク)の様にメインが(はい)の方を選択



画面下段{ルート}タブ→編集→送信元「0.0.0.0/0」にし ターゲットは上記のNATサーバーのインスタンスIDを選択(ターゲットの空欄を→クリック)して保存を選択

NATサーバー設定完了

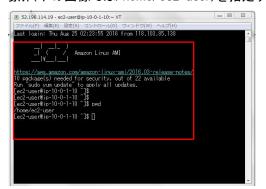


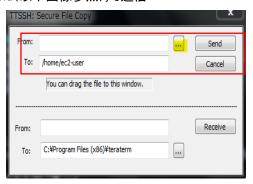
Wordpressのインストールと設定

34【mykeyをWEBサーバー上に置く。】

TeraTermにSSHで接続(14参照)し、pwdコマンドを入力し現在の自身の位置を確認する。下の画像では/home/ec2userにいることがわかる。

左上の→{ファイル}→{SSH SCP}を選択{Fromの横のボタン}からmykeyファイルを選択しToにはpwdで確認した自分の場所(下の画像では/home/ec2-user)を指定する。→{send(以下画像参照)}で送信





35【Isコマンドを入力しファイルがあるか確認する。】

※上記で転送したmykey.pemが存在していることがわかる

36【mykeyを自分しか読めないように変更する。】

\$chmod 400 mykey.pem

37【WEBサーバーからDBサーバーにログインする。】

<u>※重要(DBに直接アクセスできないのでWEBサーバーからコマンド入力でログインが可能になる、以下コマンドは36</u> の状態のまま行う)

\$ ssh -i mykey.pem ec2-user@10.0.2.10

\$yes

38【yumコマンドを使ってインストール。】

\$sudo yum -y install mysql-server

<u>39【MySQLを起動する。】</u>

\$ sudo service mysqld start

<u>40【管理者パスワードの設定する。】</u>

\$ mysqladmin -u root password

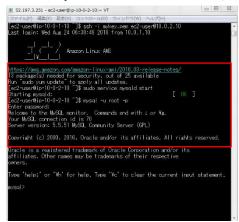
New password:設定したいパスワードを入力(ここではwordpress) 入力しても画面には文字は現れない

次に、confirm new password:と出るので、上と同じパスを入力

41【DBサーバーのMYSQLに入る。】

Enter password: パスワードを入力(ここでもwordpress)

ウェルカムメッセージ(下の画像)が表示され、MySQLのコマンドプロンプトが表示される



42【データベース名を決める。】

mysal > CREATE DATABASE wordpress DEFAULT CHARACTER SET utf8 COLLATE utf8 general ci:

<u>43【wordpressユーザーに今作成したwordpressデー</u>

mysql> grant all on wordpress.* to wordpress@"%" identified by 'wordpress'; この(mysql>)入力は不要

上記(45, 46)の権限を反映させるため、flush privilegegを実行する

mysql>flush privileges;

45【wordpressユーザーが登録されているか確認する。】 mysql>select user, host from mysql.user;

46 mysql>exitでlogoutしDBサーバーに切り替える。

47【DBサーバーが起動したとき、MySQLを自動起動する。】

\$ sudo /sbin/chkconfig mysqld on

48 \$ exitでlogoutしWEBサーバーに切り替える。

49【WordPressの実行に必要なライブラリをインストールする。】

\$ sudo yum -y install php php-mysgl php-mbstring

50【mysqlコマンドをインストールする。】 \$sudo vum -v install mysql

1【WebサーバーからDBサーバー上のMySQLへの疎通確認する。

\$ mysql -h 10.0.2.10 -u wordpress -p \$パスワード→wordpress \$exit.

<u>52【WordPressのダウンロードする。】</u> \$ cd ~

53 \$ wget http://wordpress.org/latest.tar.gz

<u>54</u>【展開する。】

55【wordpressディレクトリに移動する。】

\$ cd wordpress

※このディレクトリの中身がwordpressのプログラムー式となる

<u>56【「Apache」から見える場所にコピーする。】</u>

57【ファイルの所有者/グループをapache/apacheに変更する。】

\$ sudo chown apache:apache /var/www/html -R

<u>58【PHPを有効にする為再起動する。】</u>

59【WordPressの初期設定をする。】

EC2タブから{インスタンス}を開き、Webサーバーに割り当てられているパブリックIPアドレスを調べる インターネットで上記IPをURLに入れエンター→「言語を選択し続ける」→「さあはじめましょう」をクリック 設定画面が表示されるので、DBサーバーに作成した「Wordpressデータベース」の情報を入力

> データベース名、ユーザ名ともにwordpress パスワード: wordpress データベースホスト: 10.0.2.10、他はそのまま→{送信}

情報を入力し、「インストール実行」をクリック

サイトのタイトルや管理者のユーザ名、パスワードなどを入力する→[WordPressをインストール]

	聞インストールブロセスへようこそ ! 以 シング・ブラットフォームを使い始めるこ	下に情報を記入するだけで、世界一拡張性が高くパワ とかできます。
必要情報		
次の情報を入力してくださ	気い。ご心配なく、これらの情報は後から	いつでも変更できます。
サイトのタイトル	AWSdで学ぶネットワーク入門	
ユーザー名	admin	
	ユーザー名には、半角英数字、スペース、下額	l、ハイフン、ビリボ、アナマーク (@) のみが使用できます。
パスワード	THE REAL PROPERTY.	Ø2 18:4
	強力	
	重要:ログイン特にこのバスワードが必要	こなります。安全な場所に保管してください。
メールアトレス		
	次に迷む前にメールアドレスを65一度確認し	K Ear.
検索エンジンでの表示	□ 検索エンジンがサイトをインデッ	クスしないようにする
	このリクエストを尊重するかどうかは検索エンジ	ブンの設定によります。

60【初期設定完了】→ログイン(パスは64のものを利用します)



※下の画像は、ログイン後の画面です。→完成



<u>以下も重要な設定です。合わせて設定することをお勧めします。</u>

※インスタンス画面(現在3つが稼動中)でID上にて右クリック→インスタンス→停止で動作をとめることができる。 無料枠では使用時間が限られている為、使用しない時は停止することをお勧めします。



無料枠を超えない様にアラートを設定できる。以下参照

https://docs.aws.amazon.com/ja_jp/awsaccountbilling/latest/aboutv2/monitor-charges.html