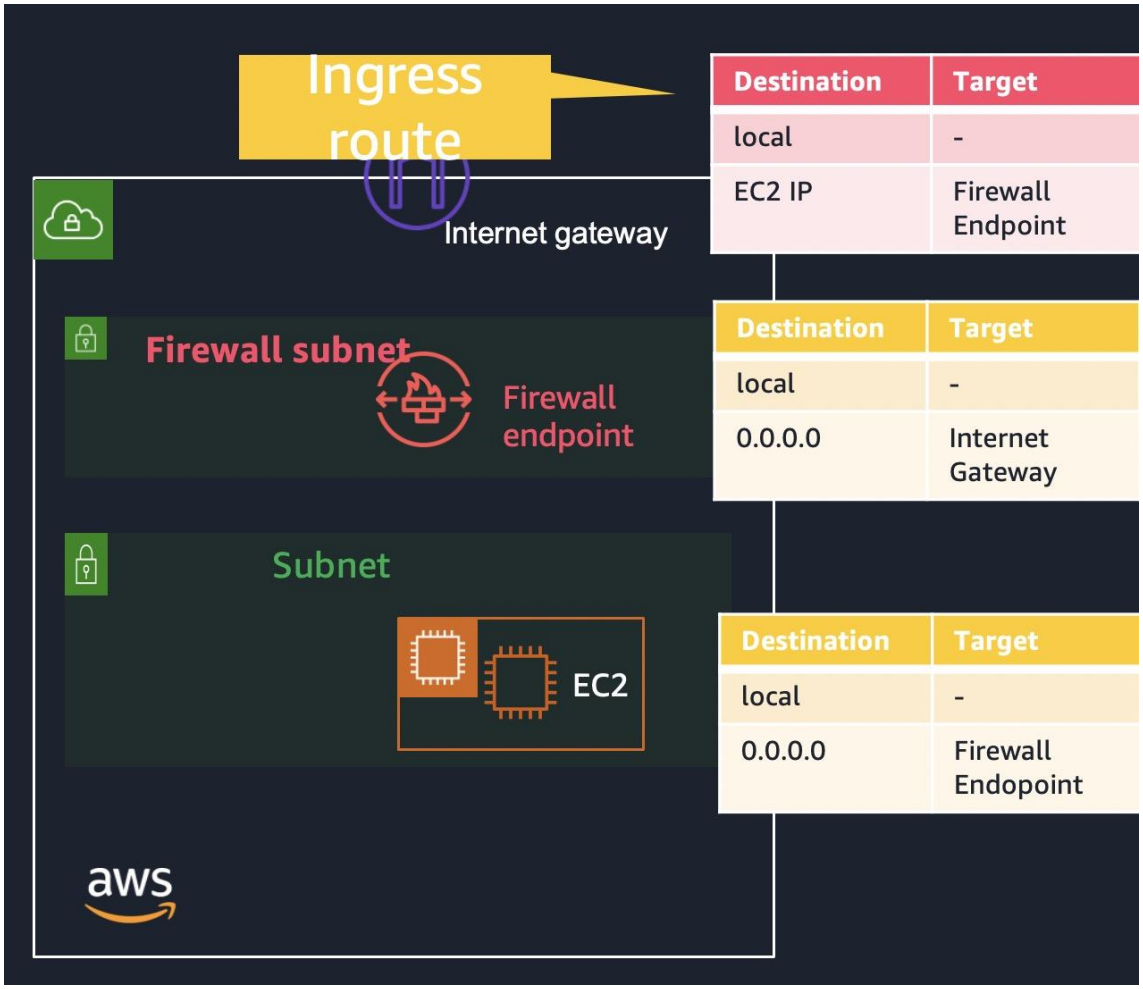
AWS Network Firewall ハンズオン

2021/06/09

シニアエバンジェリスト
亀田

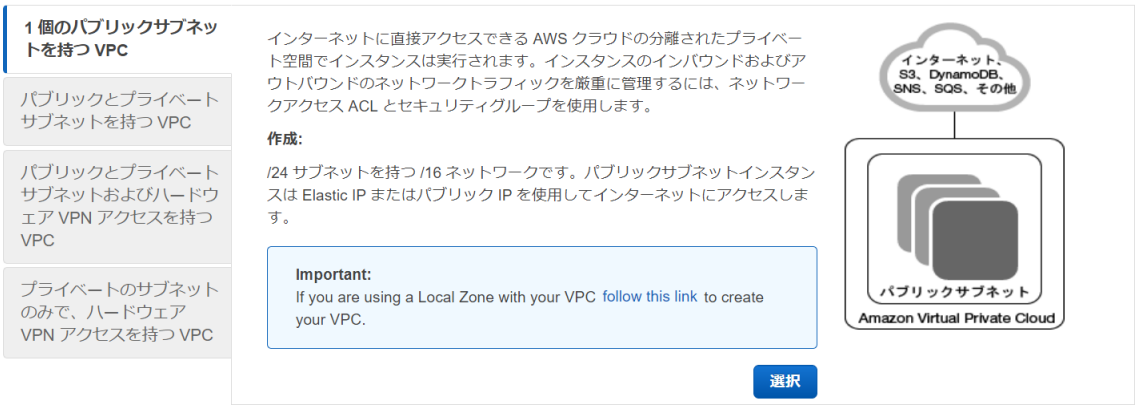
[はじめに]このハンズオンでは多くの設定パラメータを取り扱います。それらを管理する memo.txt が用意されていますので、手順の指示に従い memo.txt に設定項目を書き込んで下さい。
また VPC の少し複雑な構成を構築するため、作業途中で混乱してしまった場合は以下を常に参考してください。




## VPC と Public サブネット、EC2 の起動

1. VPC ウィザードを起動します


2. [1 個のパブリックサブネットを持つ VPC]を選びます



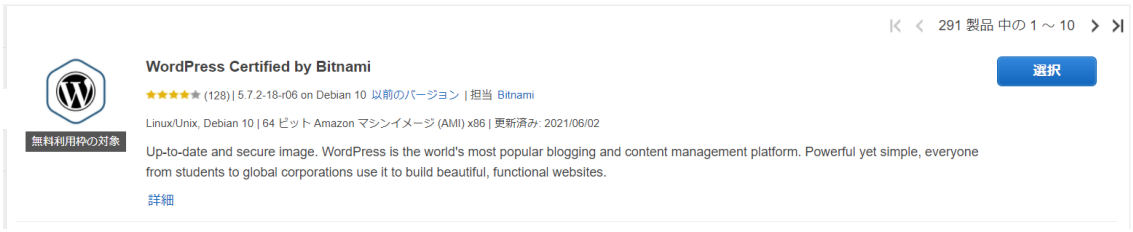

3. memo.txt の内容を参考にしながら値を入力し、[VPC の作成]を押します


4. 作成されたら画面左ペイン、[サブネット]を選択し、作成されている Public という名のサブネットの AZ 情報を memo.txt の[Public Subnet AZ]に記載しておきます。


5. 同様にインターネットゲートウェイの ID を memo.txt の[Internet Gateway]に記載します
6. EC2 の画面から[インスタンスを起動]を押します


7. wordpress と検索し、[AWS Marketplace]をクリックします

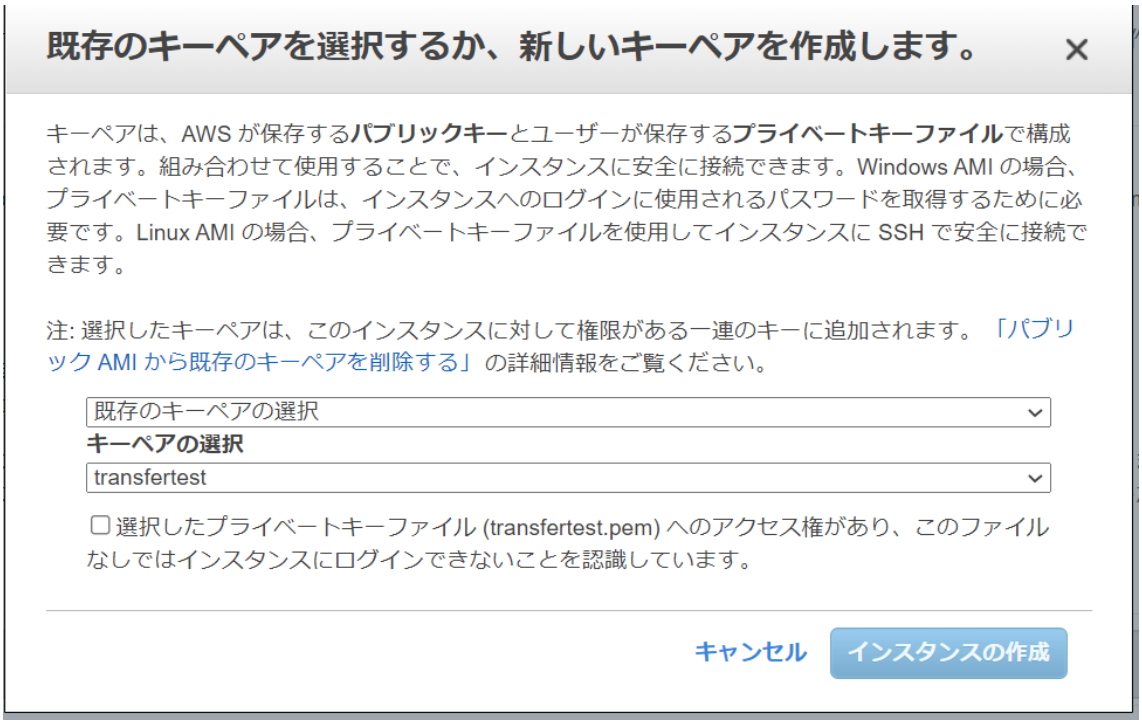

8. [WordPress Certified by Bitnami]を選びます。（ブラウザでアクセス可能な EC2 を起動したいだけなので何でも大丈夫です）



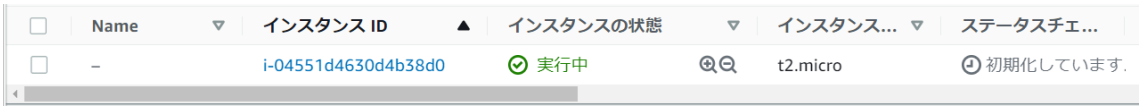

9. 画面一番下の[Continue]を押します
10. [t2.micro]を選び、[次のステップ]をおします


11. ネットワーク設定を先程作成したサブネットに配置するように指定します。[自動割り当てパブリック IP]を有効にしてください

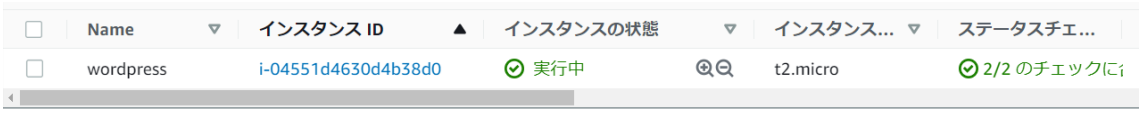

12. その他全てデフォルトで[確認と作成]をおします。次の画面で再度[起動]をおします
13. [キーペアなしで続行]を選び[インスタンスの作成]をおします。(SSH の FW による制御などをテストされたい方は正しい鍵を設定してください。このハンズオンでは割愛します)




14. EC2 が起動中で順次ステータスが以下のように変更していくので待ちます



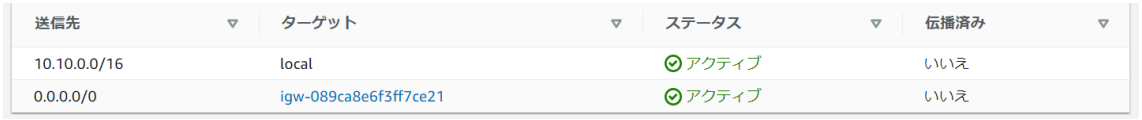

待っている間に、Name に[wordpress]と設定しておきます。インスタンス ID を memo.txt の[EC2 ID]にコピペしておきます
ステータスチェックが緑色で 2/2 と表示されればアクセスが可能です




左のチェックボックスにチェックをつけ、パブリック IPv4 DNS の値を、memo.txt の[EC2 PublicDNS]にコピペした後、ブラウザでアクセスしてみてください

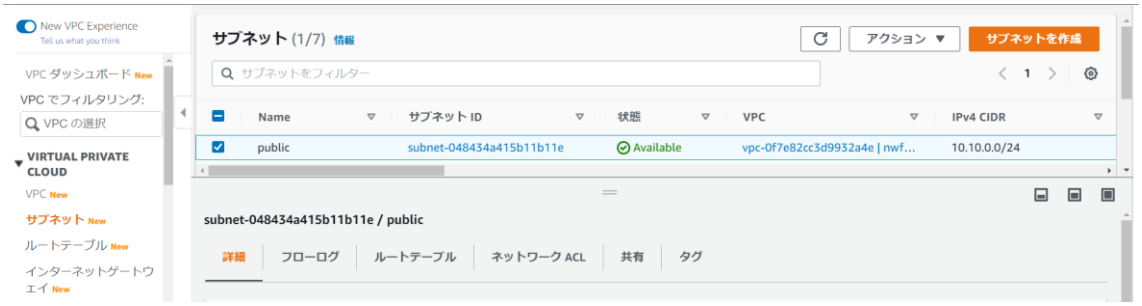
15. wordpress の画面が表示されれば完了です。記事を書くわけでありませんので、表示される内容はなんでも大丈夫です。
16. VPC 画面左ペインからルートテーブルをクリックします。EC2 を起動したサブネットに紐ついているルートテーブルを特定し、その ID を memo.txt の[Route Table ID]にコピペします。また、Name を[public]に変更しておきます。

    このルートテーブルは、以下のように VPC 内部の通信は local、それ以外の通信(つまり VPC 外部への通信)は全て Internet Gateway へルーティングする、となっています。後ほど Network Firewall の構築を行ったあと、VPC 外部への通信は Internet Gateway ではなく、Network Firewall のエンドポイントを指定します。

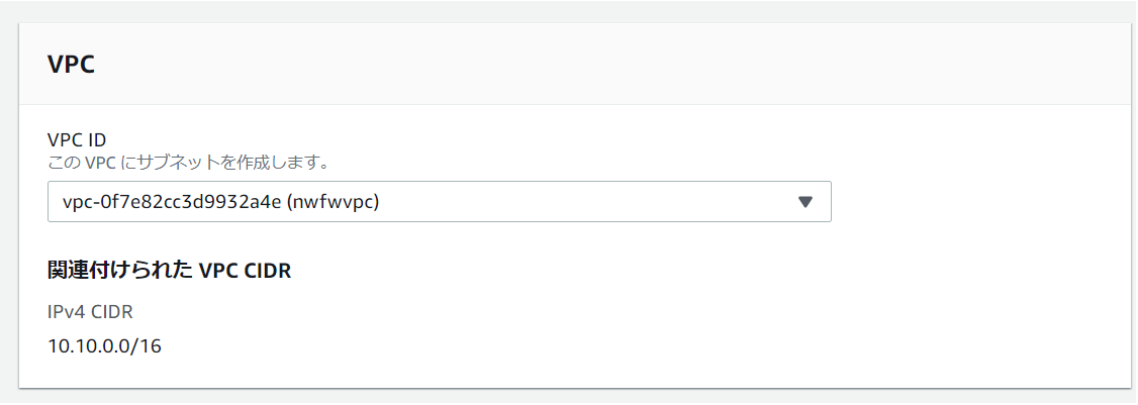



**Network Firewall 用ネットワークの構築**

17. 今までの手順で、Network Firewall で保護すべき環境ができました。これから、Network Firewall 専用サブネットを同じ VPC に作成していきます。
    VPC 画面左ペインから、サブネットをクリックし、[サブネットを作成]をおします



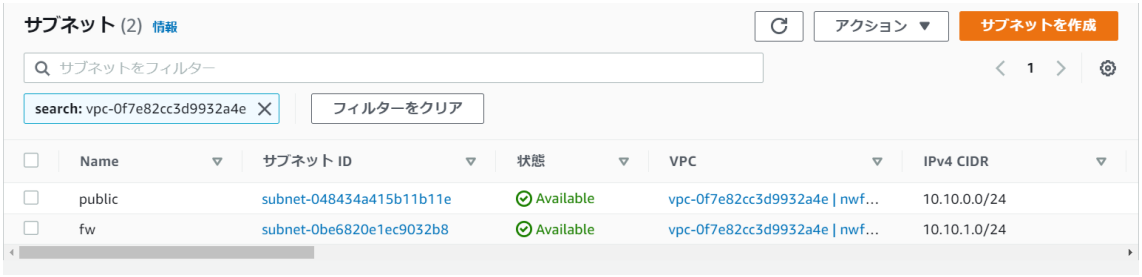
18. 先程 EC2 を起動した VPC を選びます




19. 以下のように設定して[サブネットを作成]をおします


AZ は必ず EC2 と同じにしてください。その他の値は memo.txt を参考にして入力してください

20. 現在 VPC には Subnet が 2 つ存在しています。　新しくできたサブネット ID を memo.txt にコピペしておきます




21. VPC 画面左ペインから、ファイアウォールを選んでクリックします

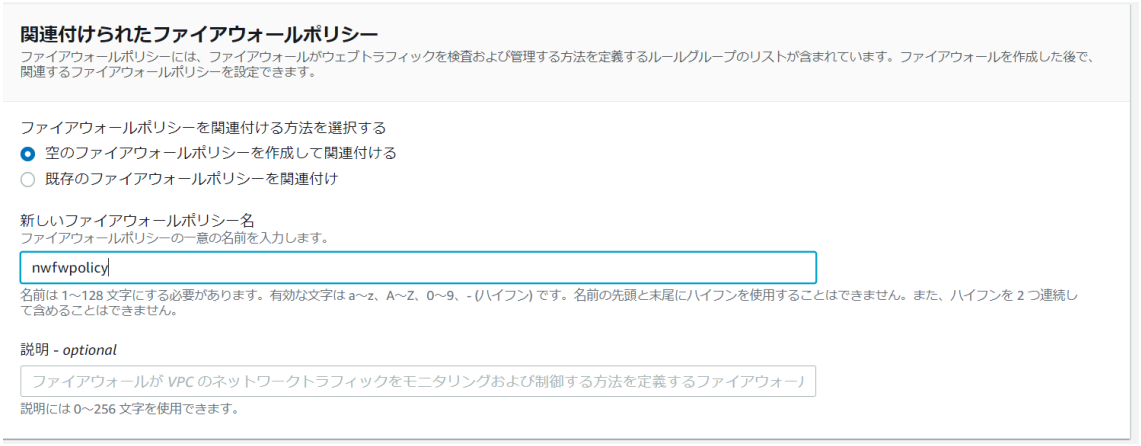

22. [ファイアウォールの作成]をおします


23. 名前を適当につけます


24. VPC は EC2 を起動した VPC を指定します。AZ も同様に同じです。（NetworkFirewall は AZ 単位で動作するため、必ず揃えてください）。サブネットは EC2 が起動している public ではなく、fw を指定します。つまり以下のような経路を作ります。

    Internet ⇔ Internet Gateway ⇔ fw subnet（Network Firewall）⇔ public subnet


25. ポリシーは後で設定しますが、何か必要なため、[空の…]を選んで適当な名前を付けます。




26. [ファイアウォールを作成]をおします
27. ステータスが[プロビジョニング]になるので、しばらく待ちます


28. 待っている間に、fw subnet 用ルートテーブルを作成します。VPC 画面左ペインからサブネットをクリックし、[ルートテーブルを作成]をおします。名前に[fwrt]とつけ、VPC は EC2 を起動したものを選択します


29. [ルートテーブルを作成]をおします
30. 作成されたルートテーブル ID を memo.txt の[FW Route Table ID]にコピペします

31. ブラウザでネットワークファイアウォールのタブに戻ります。以下のように、準備完了と表示されていれば起動は完了。fw subnet で Network Firewall が起動されました


32. 画面左ペインからルートテーブルをクリックし、[public]の方をクリックします


現在 VPC には 3 つのルートテーブルが存在しています。

public:明示的に public subnet に紐付けられている

fwrt: Network Firewall ようだがまだ fw subnet に紐付けられていない（後で紐付け作業を行います）

メイン: デフォルトで存在しているもの。明示的に紐付けられたルートテーブルを持たないサブネットが利用する。(現時点だと fw subnet は fwrt に紐付けられていないため、こちらが使われる)


これらのルートテーブルを変更しながら

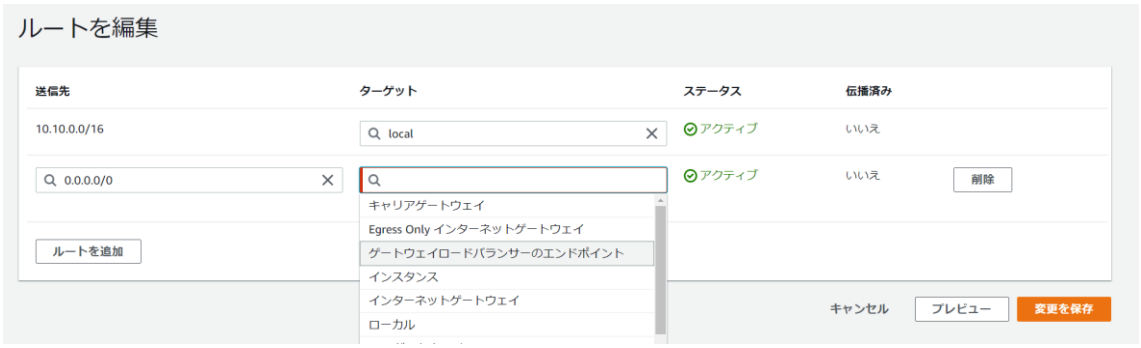
Internet ⇔ Internet Gateway ⇔ fw subnet (Network Firewall) ⇔ public subnet

を設定していきます。

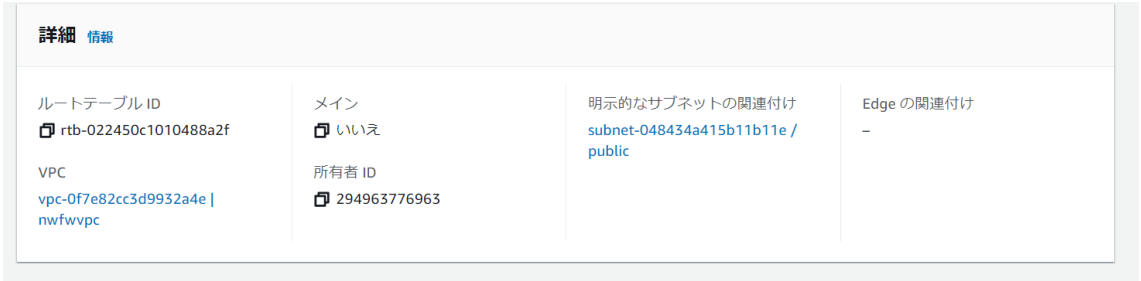
33. まず public ルートテーブルと public サブネットの紐付けを一度外します。これは本来設計上必要ないのですが、そうしないと設定変更作業が行えません。このため、一度外した後、設定変更を行い、再度紐付けを行います。[アクション]から[サブネット

の関連付けを編集]を選んでチェックを外せば、紐付けが解除されます。


34. ルートの編集ボタンを押し[0.0.0.0/0]の設定内容がインターネットゲートウェイになっているものをゲートウェイロードバランサーのエンドポイントに変更し、[変更を保存]をおします




35. 再度[アクション]からサブネットの関連付けを編集を選び、public subnet と関連付けます。正しく作業されれば、以下のようになるはずです。

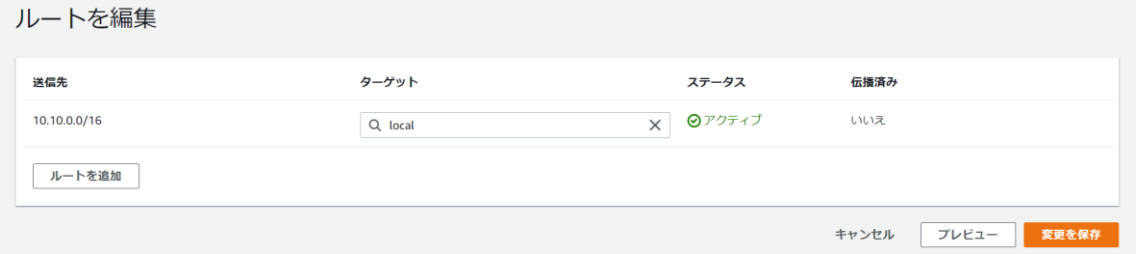




これで EC2 とインターネットとの通信は、Network Firewall 経由となります。EC 2

から直接 Internet Gateway へアクセスできなくなりました。

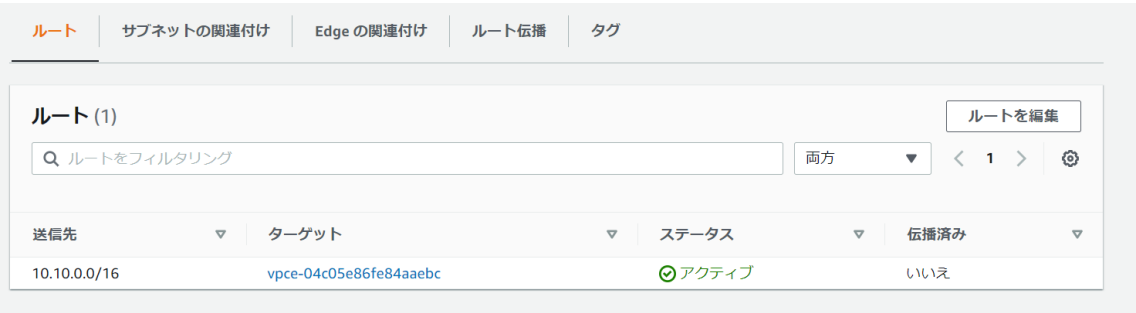
36. 続いて fw subnet 用の fwrt を編集します。[ルートの編集]をおしてください




37. 以下のように[0.0.0.0/0]に対してインターネットゲートウェイをセットします


これで、fw subnet に存在しているノード（Network Firewall）がインターネットゲートウェイへ通信をルーティングできるようになりました

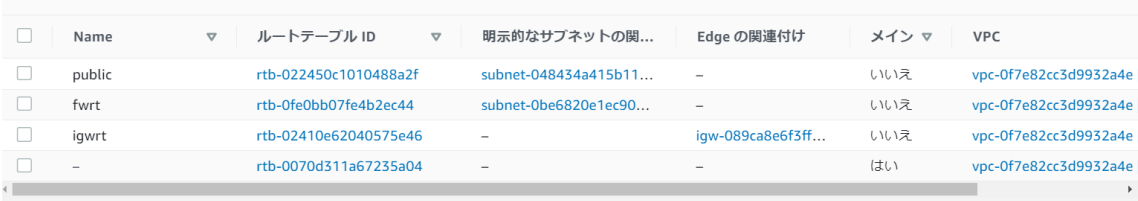
38. もう一つ、画面左ペイン、ルートテーブルをクリックして[ルートテーブルを作成]をおします。これは、subnet 用ではなく、Internet Gateway 用のルートテーブルです。外部から受ける Internet Gateway の通信を全て、Network Firewall へ振り向けるためのルートテーブルです。名前を[igwrt]にし VPC は EC2 と同じものを指定します。その後ルートを編集し以下のようにします。




39. [Edge の関連付け]タブを選び、[Edge の関連付けを編集]ボタンをおします


40. インターネットゲートウェイにチェックをつけ[変更を保存]をおします。以下のようになります

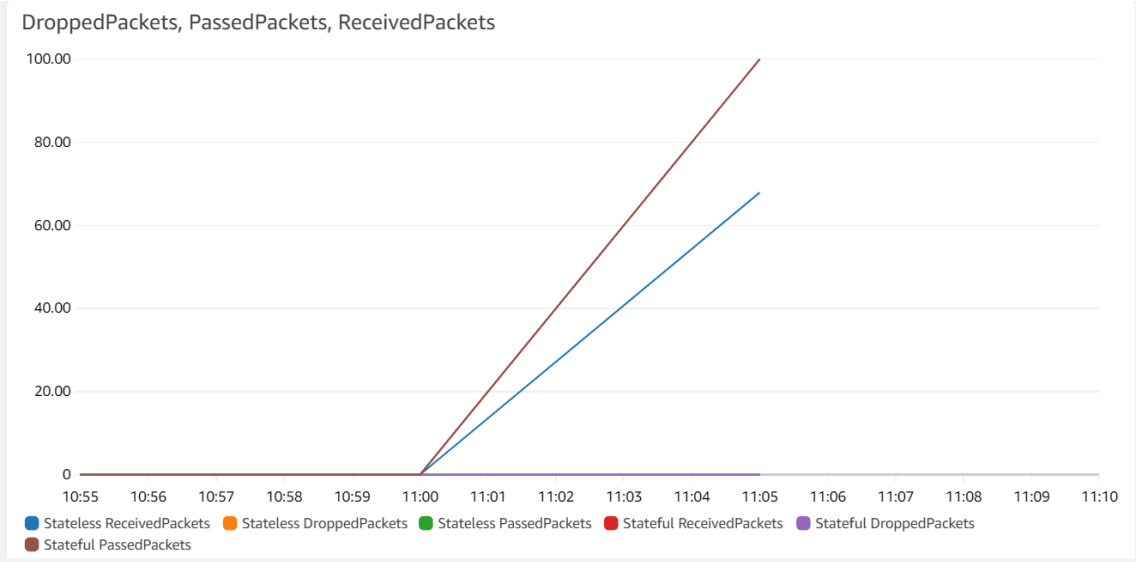

41. 最後に、ルートテーブルの fw を fw subnet に紐付けます。以下のようになります。




存在しているすべての Subnet には明示的にルートテーブルが紐付けられているた

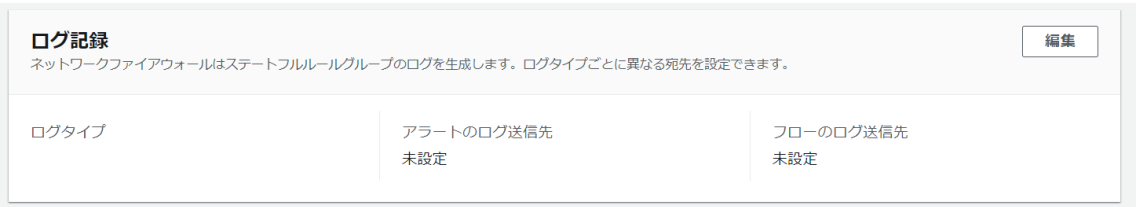
め、メインルートテーブルは使われていません。

42. この状態で memo.txt の[EC2 PublicDNS]にアクセスしてみてください
43. Wordpress が表示されたら、ブラウザで少し何度かリロードをして複数回アクセスを行った後、Network Firewall の[モニタリング]タブをクリックすると、通信が通っていることがわかります。




**ログの設定とセキュリティルール設定**

44. 今までの手順で、インターネットから EC2 への通信は Network Firewall を通る環境を作ることができました。これから、ログの設定、そしてセキュリティルールの設定を行っていきます。
    ファイアウォールの詳細タブから、ログ記録の編集をおします

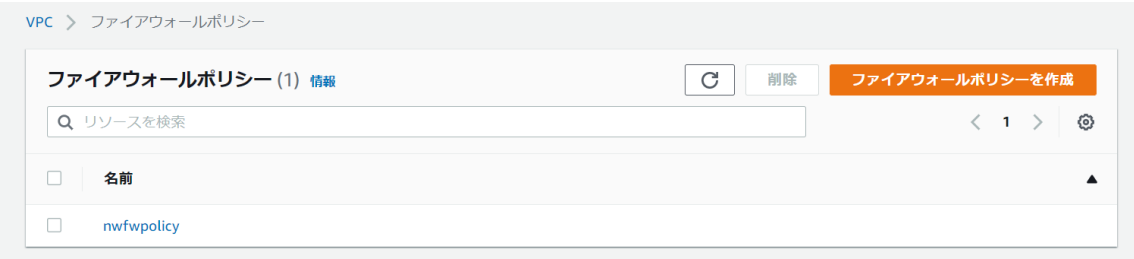



45. Alert にチェックをつけます
46. [CloudWatch log group]を選び、[ロググループを作成]をおします


47. [ロググループを作成]を再度押します


48. 適当な名前を付け[作成]をおします


49. ファイアウォールのログ設定画面に戻ると、先程指定したロググループが指定できるようになっていますので、指定をして[保存]をおします
50. 次に画面左ペインから、ファイアーウオールポリシーをクリックし先ほど作成した空のポリシーをクリックして選びます




51. ステートフルグループの[ルールグループを追加]から[新しい・・・]を選びます。**ステートレスグループではない**ので注意してください


52. 適当な名前を付け、キャパシティに 100 と入力します


53. [Suricata compatible IPS rules]を選びます


54. memo.txt の中に入っている一番最後の文字列をコピペします
alert tcp any any -> any any (msg:"TCP traffic detected"; sid:200001; rev:1;)


55. [作成と追加]をおします。これでファイアウォールがすべての TCP 通信を Alert として CloudWatch Logs に出力するようになりました。ブラウザから wordpress に何度かアクセスしてログの出力を確認してください。(数分程度時間がかかります)


56. では、Alert ログの出力が確認出来たら、今度は先程のルールを編集します。画面左のペインから、ネットワークファイアウォールのルールグループをクリックして、表示されたルールをさらにクリックして、[ルールを編集]ボタンを押してください。


57. alert の部分を drop に変更して、保存します。数分まって再度ブラウザから wordpress にアクセスしてください。今度はアクセスできなくなっていることが確認できます。
Suricata のルールは、かなり複雑な設定が可能です。詳しくはこちらをご確認ください。
https://docs.aws.amazon.com/ja_jp/network-firewall/latest/developerguide/suricata-examples.html

おつかれさまでした！
削除ですが以下の手順で行ってください

1．ルートテーブルからルート編集でターゲットが vpce-となっている行を消す
2．ルートテーブルの Edge 及びサブネットの関連付けをはずす
3．EC2 を削除
4．Network Firewall のログ出力を停止
5．Network Firewall を削除

6．VPC を削除
7．ファイアーウオールポリシーの削除
8．ファイアーウォールのルールグループの削除
9．CloudWatch Logs のロググループを削除