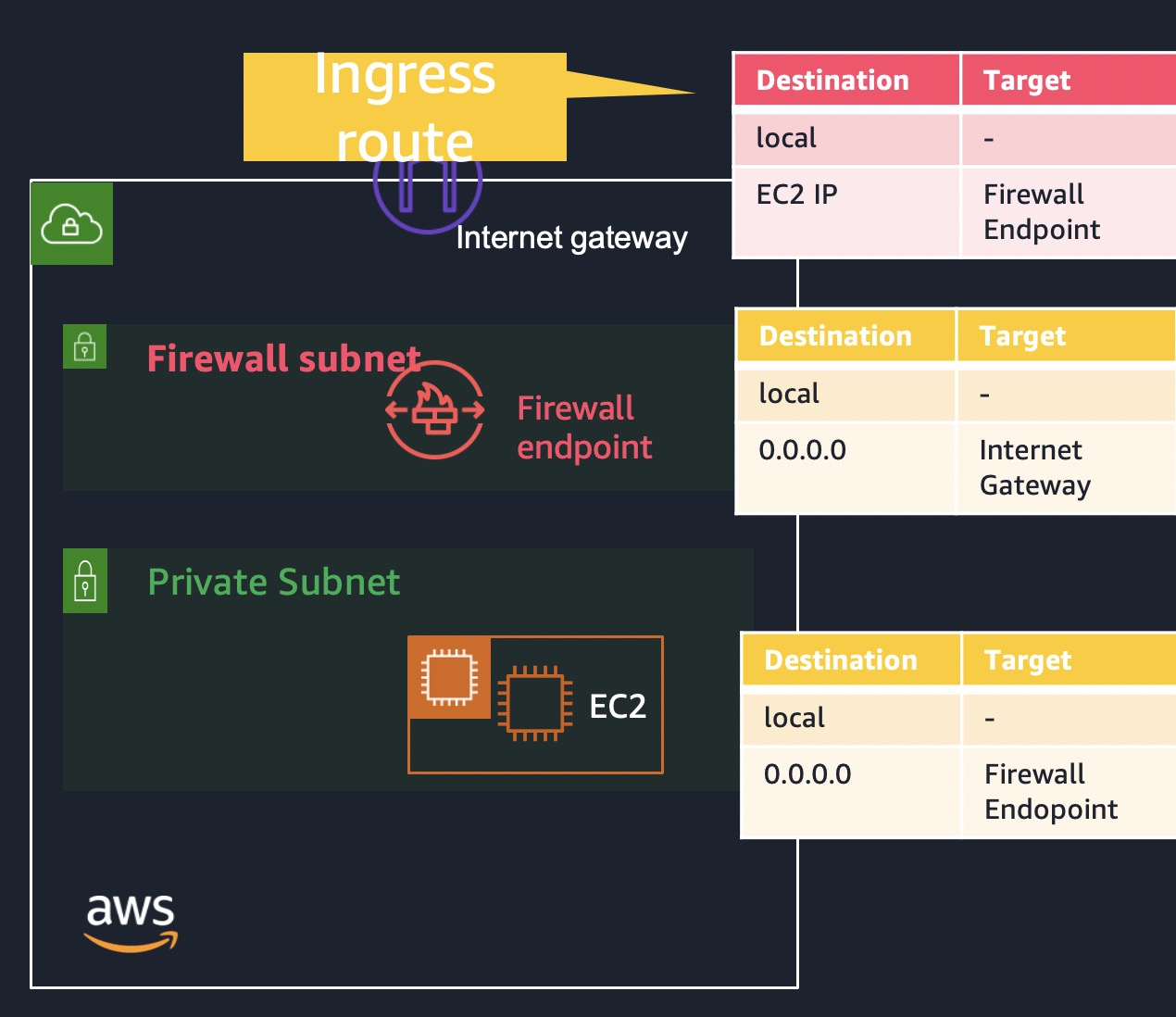
AWS Network Firewall ハンズオン

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。

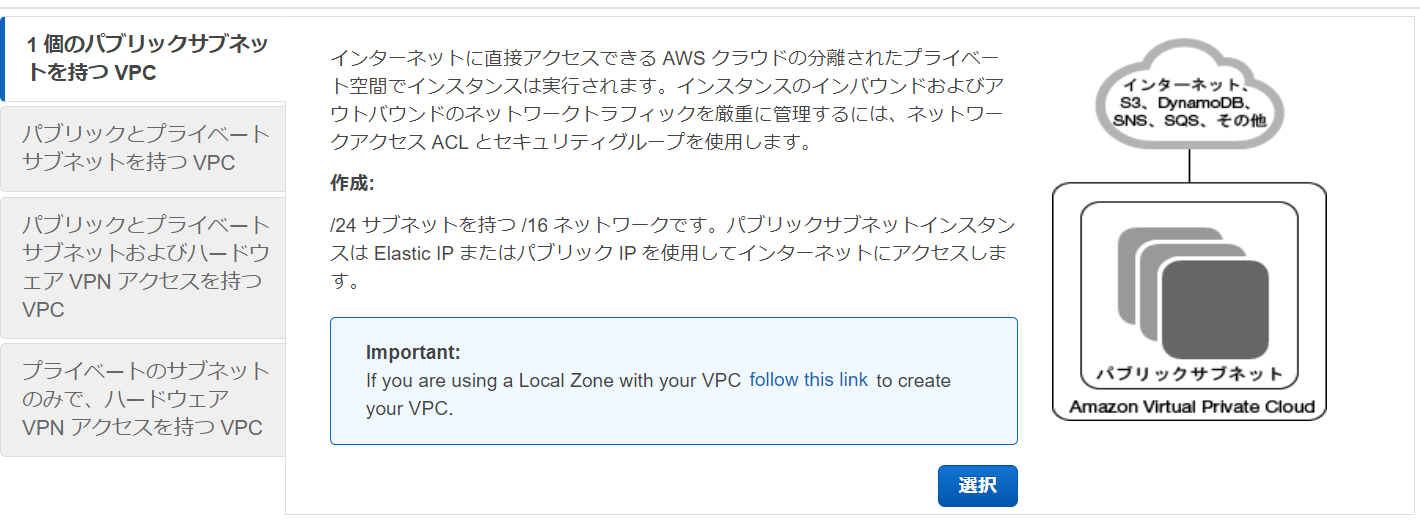
またVPCの少し複雑な構成を構築するため、作業途中で混乱してしまった場合は以下を常に参考してください。

**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



1. [１個のパブリックサブネットを持つVPC]を選びます



1. 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



1. 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。



1. 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
2. EC2の画面から[インスタンスを起動]を押します



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



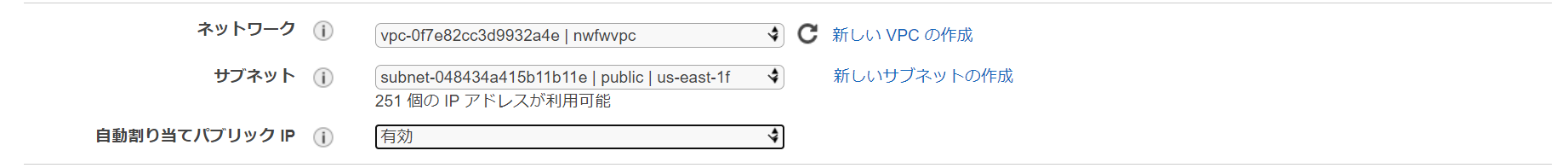
1. [WordPress Certified by Bitnami]を選びます。（ブラウザでアクセス可能なEC2を起動したいだけなので何でも大丈夫です）



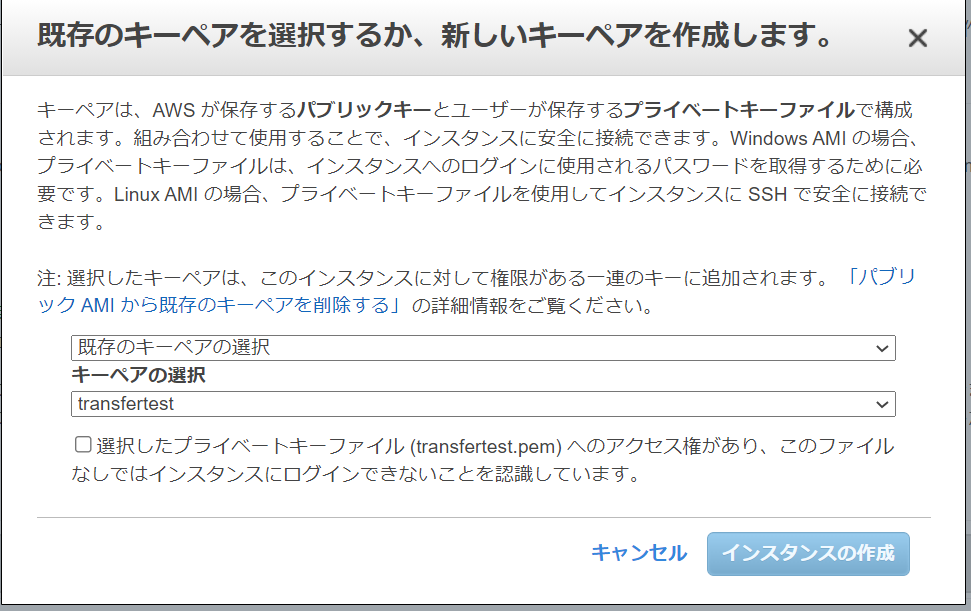
1. 画面一番下の[Continue]を押します
2. [t2.micro]を選び、[次のステップ]をおします



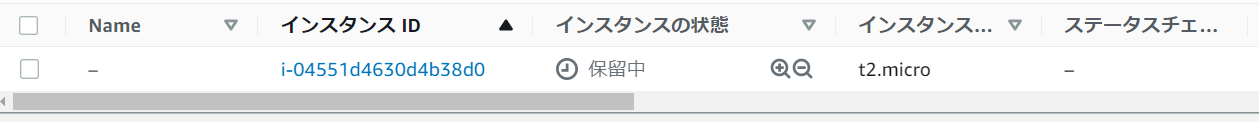
1. ネットワーク設定を先程作成したサブネットに配置するように指定します。[自動割り当てパブリックIP]を有効にしてください



1. 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
2. [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）



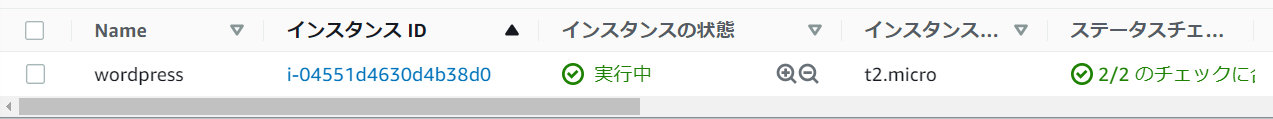
1. EC2が起動中で順次ステータスが以下のように変更していくので待ちます





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

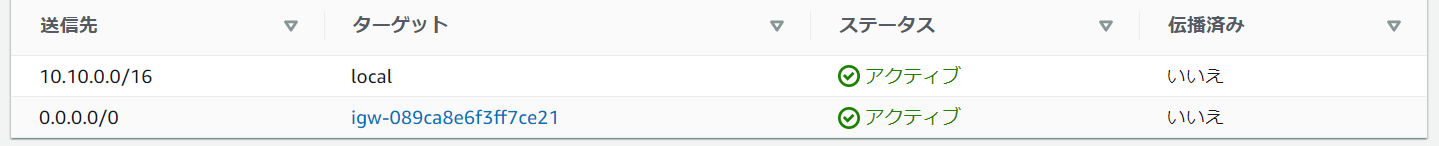
ステータスチェックが緑色で2/2と表示されればアクセスが可能です



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

1. wordpressの画面が表示されば完了です。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。
2. 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。

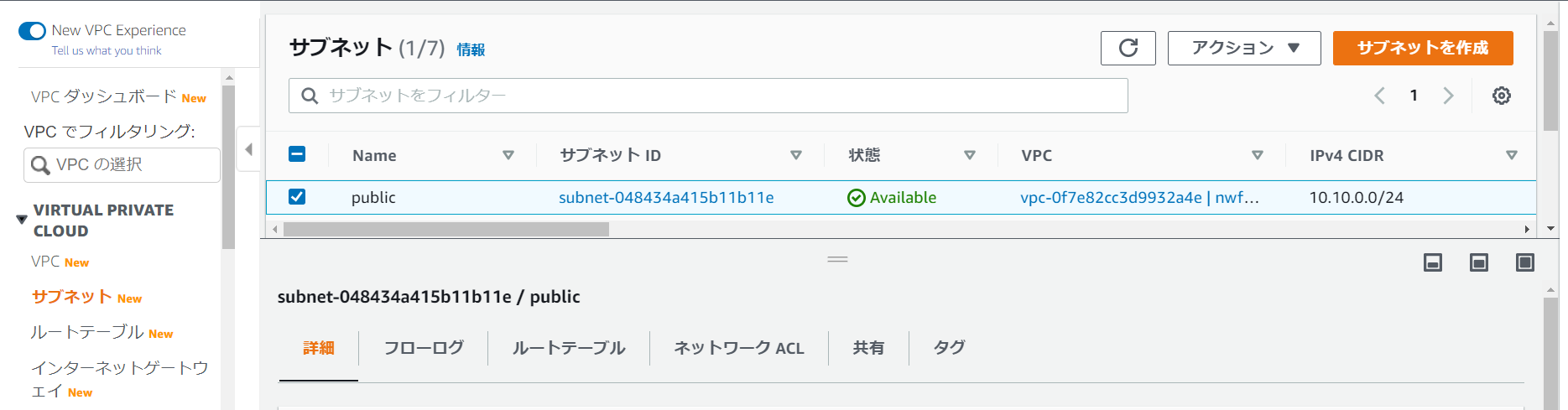
このルートテーブルは、以下のようにVPC内部の通信はlocal、それ以外の通信(つまりVPC外部への通信)は全てInternet Gatewayへルーティングする、となっています。後ほどNetwork Firewallの構築を行ったあと、VPC外部への通信はInternet Gatewayではなく、Network Firewallのエンドポイントを指定します。



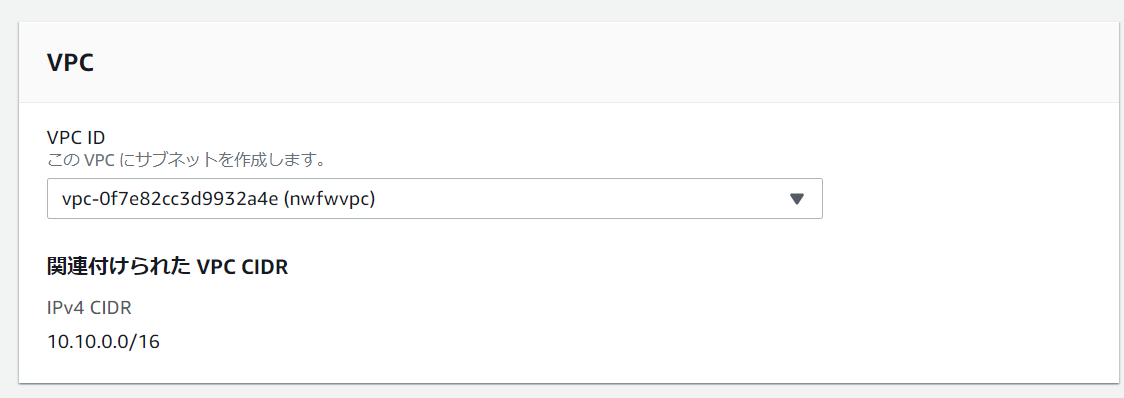
**Network Firewall 用ネットワークの構築**

1. 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



1. 先程EC2を起動したVPCを選びます

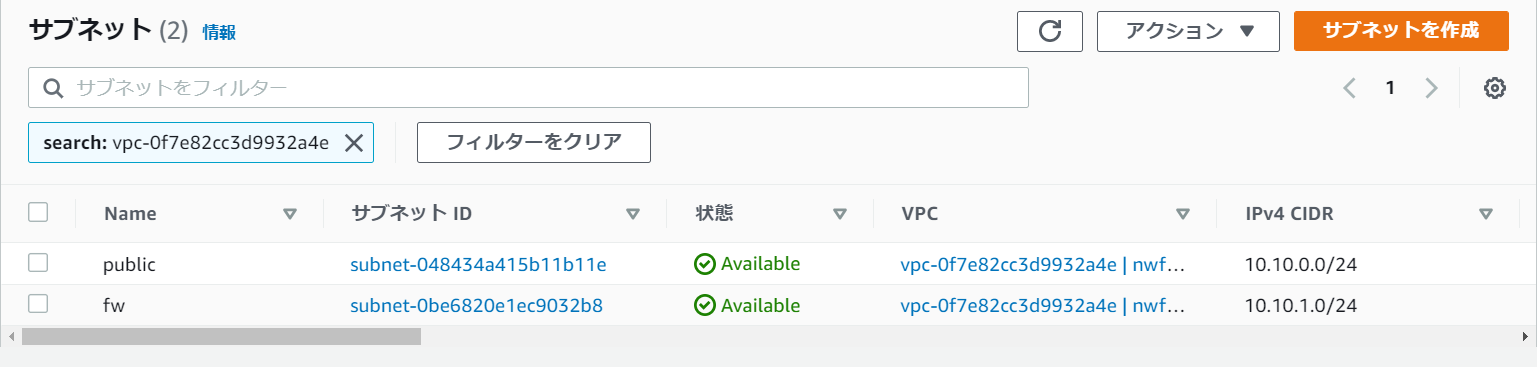


1. 以下のように設定して[サブネットを作成]をおします



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

1. 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



1. 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



1. [ファイアウォールの作成]をおします



1. 名前を適当につけます

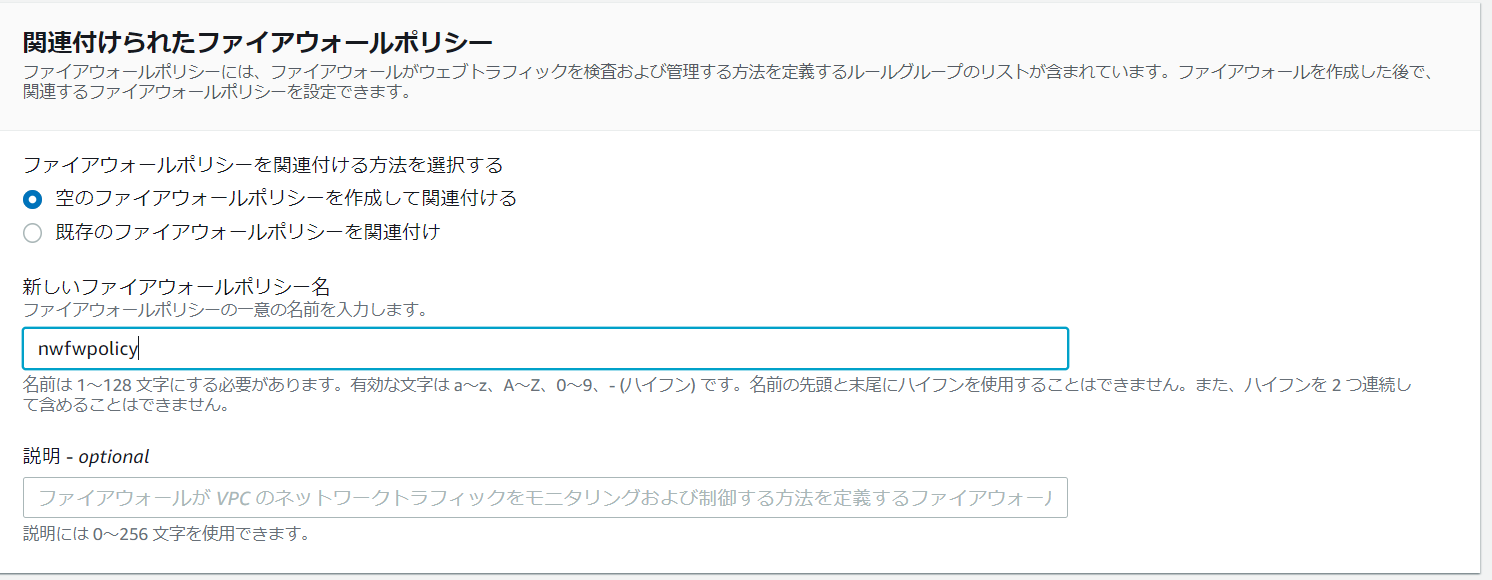


1. VPCはEC2を起動したVPCを指定します。AZも同様に同じです。（NetworkFirewallはAZ単位で動作するため、必ず揃えてください）。サブネットはEC2が起動しているpublicではなく、fwを指定します。つまり以下のような経路を作ります。

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



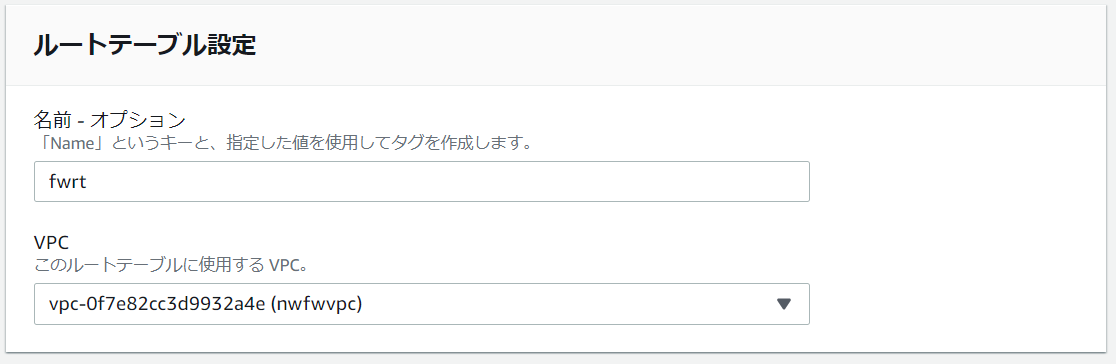
1. ポリシーは後で設定しますが、何か必要なため、[空の…]を選んで適当な名前を付けます。



1. [ファイアウォールを作成]をおします
2. 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



1. 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



1. [ルートテーブルを作成]をおします
2. 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
3. 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



1. 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

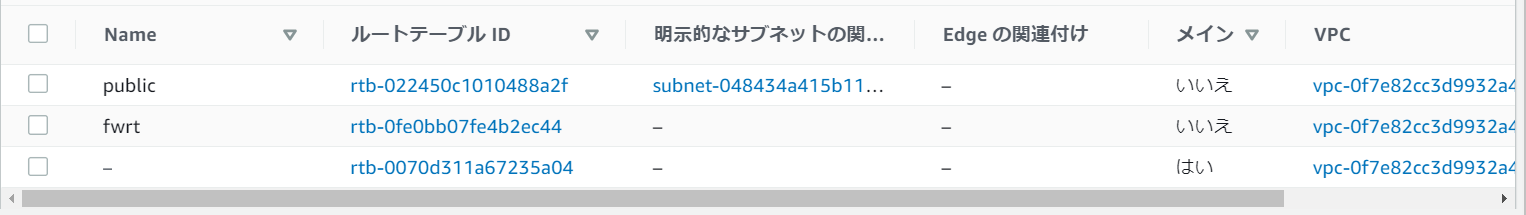


現在VPCには３つのルートテーブルが存在しています。

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

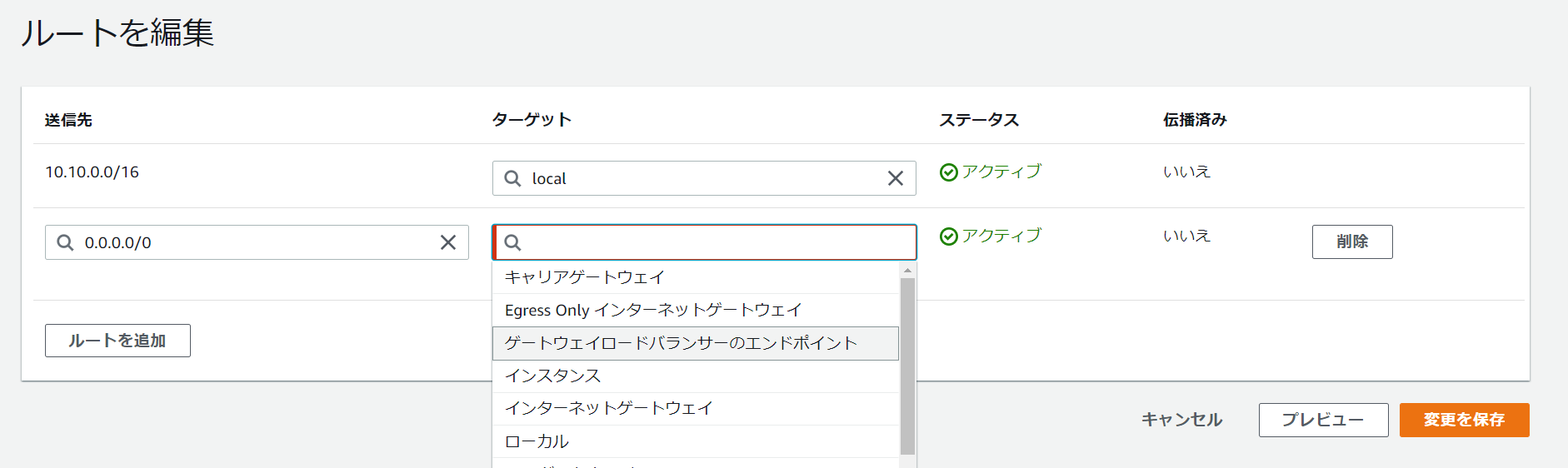
Internet ⇔ Internet Gateway ⇔ fw subnet (Network Firewall) ⇔ public subnet

を設定していきます。

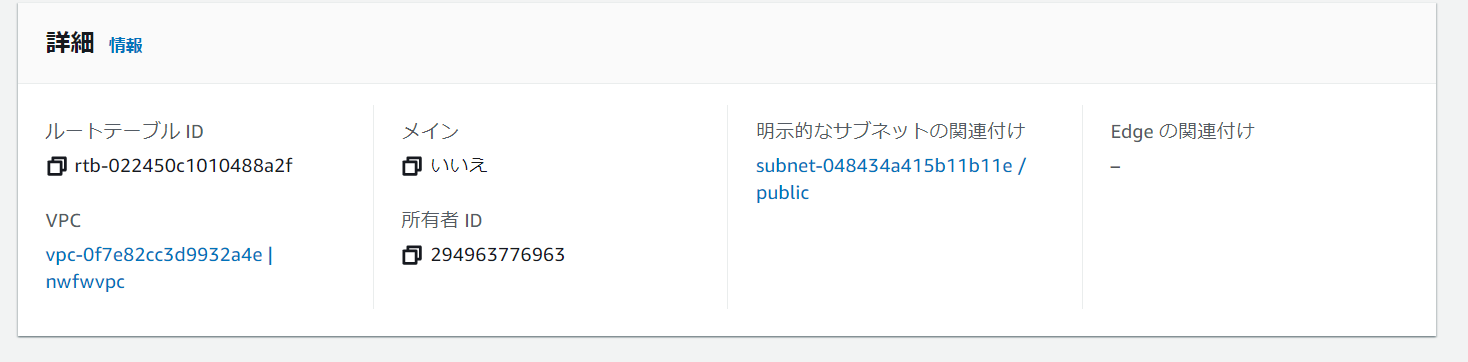
1. まずpublic ルートテーブルとpublic サブネットの紐付けを一度外します。これは本来設計上必要ないのですが、そうしないと設定変更作業が行えません。このため、一度外した後、設定変更を行い、再度紐付けを行います。[アクション]から[サブネットの関連付けを編集]を選んでチェックを外せば、紐付けが解除されます。



1. 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



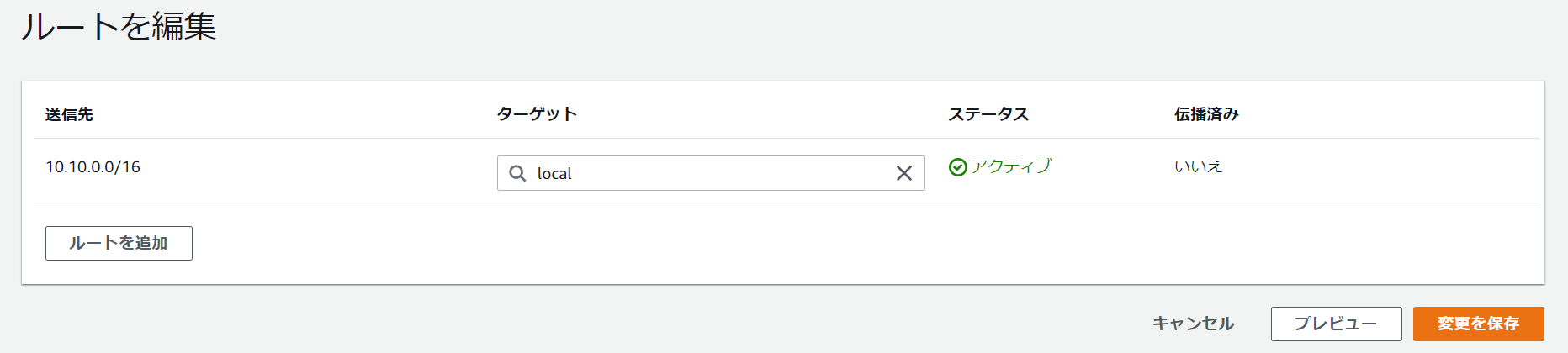
1. 再度[アクション]からサブネットの関連付けを編集を選び、public subnetと関連付けます。正しく作業されれば、以下のようになるはずです。





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。EC２から直接Internet Gatewayへアクセスできなくなりました。

1. 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

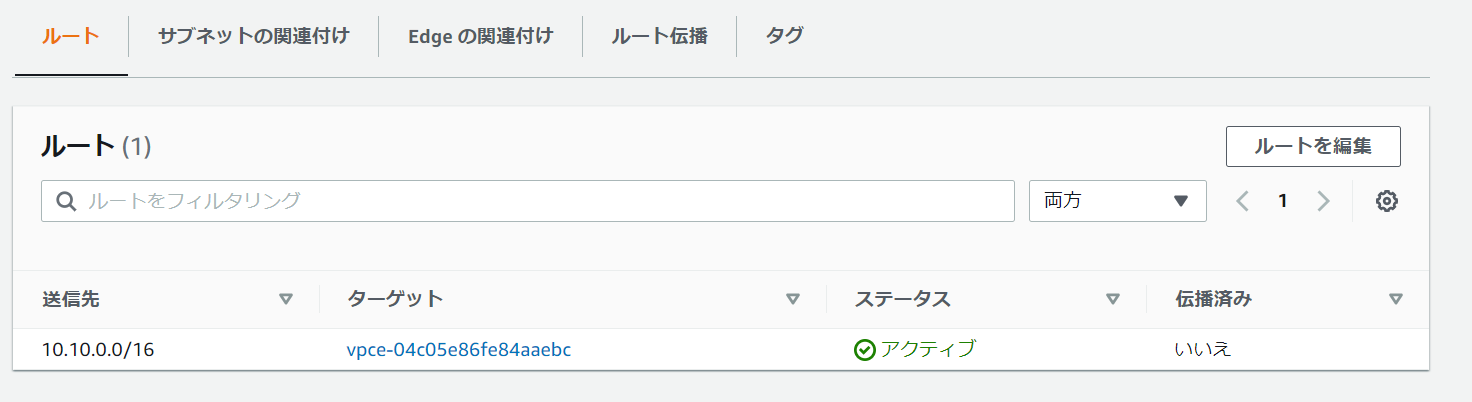


1. 以下のように[0.0.0.0/0]に対してインターネットゲートウェイをセットします



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

1. 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。



1. [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



1. 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

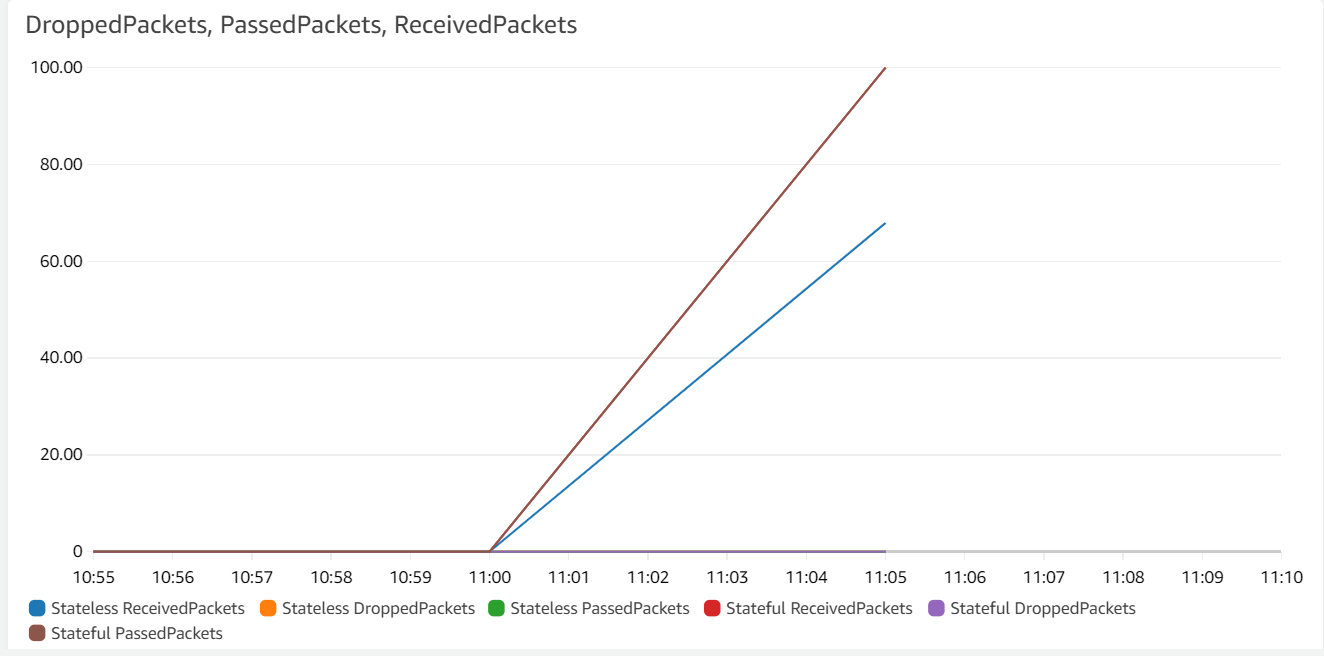


1. 最後に、ルートテーブルのfwをfw subnetに紐付けます。以下のようになります。



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。

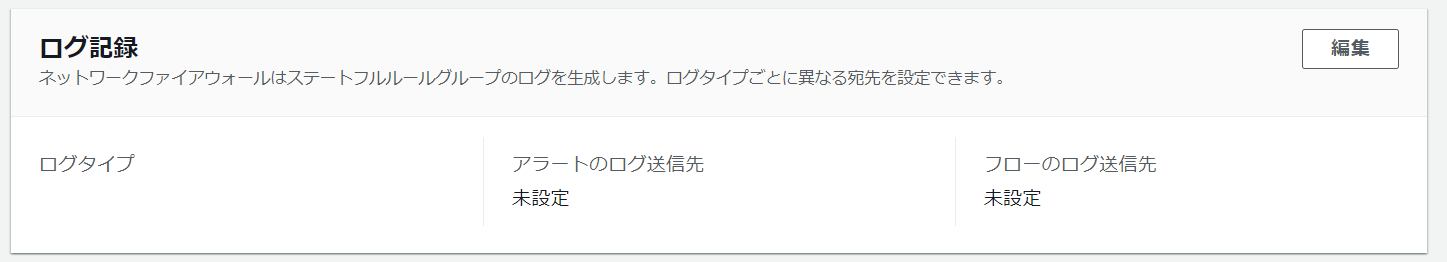
1. この状態でmemo.txtの[EC2 PublicDNS]にアクセスしてみてください
2. Wordpressが表示されたら、ブラウザで少し何度かリロードをして複数回アクセスを行った後、Network Firewallの[モニタリング]タブをクリックすると、通信が通っていることがわかります。



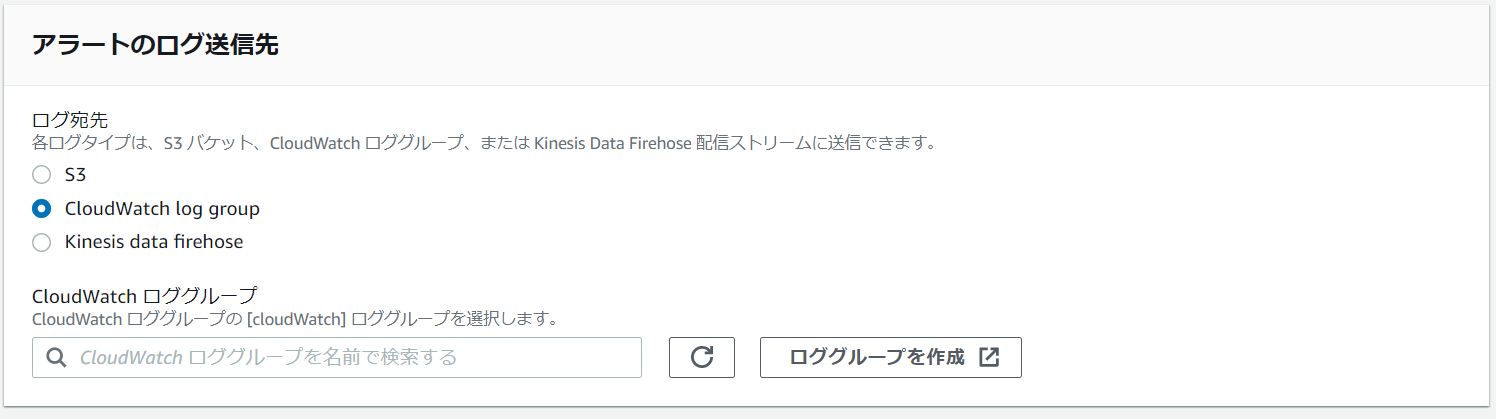
**ログの設定とセキュリティルール設定**

1. 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



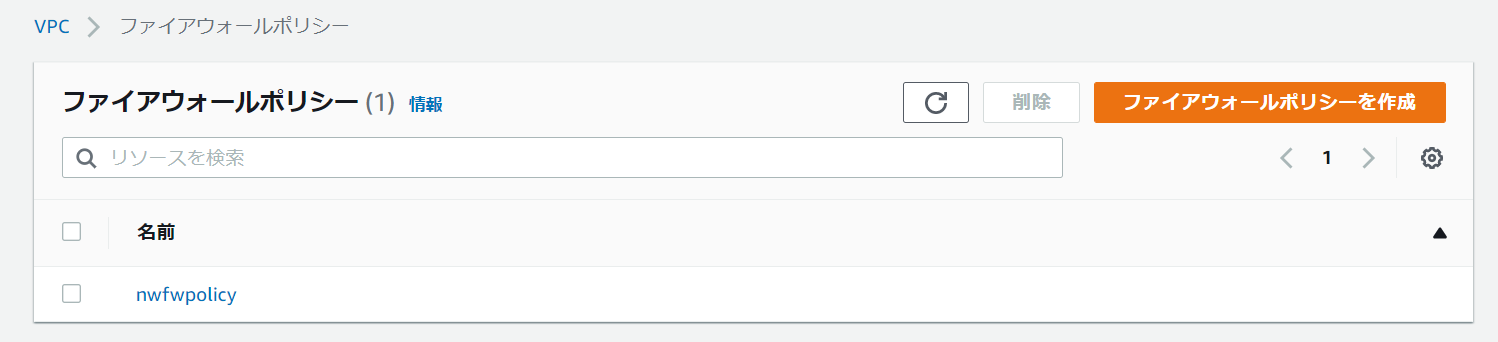
1. Alertにチェックをつけます
2. [CloudWatch log group]を選び、[ロググループを作成]をおします



1. [ロググループを作成]を再度押します
2. 適当な名前を付け[作成]をおします



1. 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
2. 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



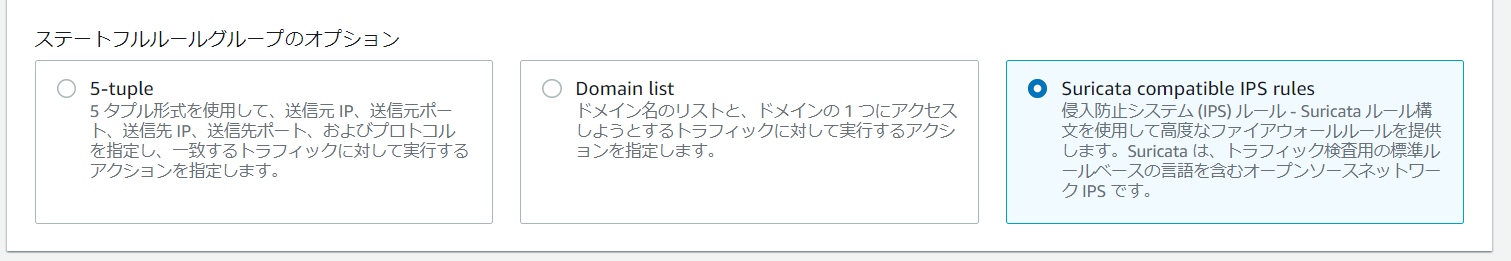
1. ステートフルグループの[ルールグループを追加]から[新しい・・・]を選びます。**ステートレスグループではない**ので注意してください



1. 適当な名前を付け、キャパシティに100と入力します

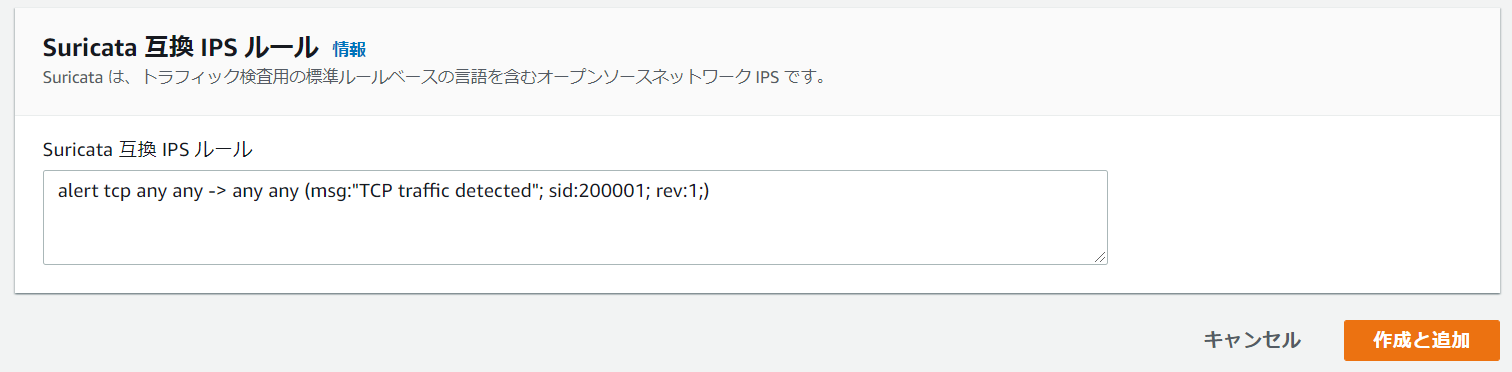


1. [Suricata compatible IPS rules]を選びます



1. 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;)



1. [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）



1. 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。



1. 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