Etherem BlockChain Template
Hands-On Lab

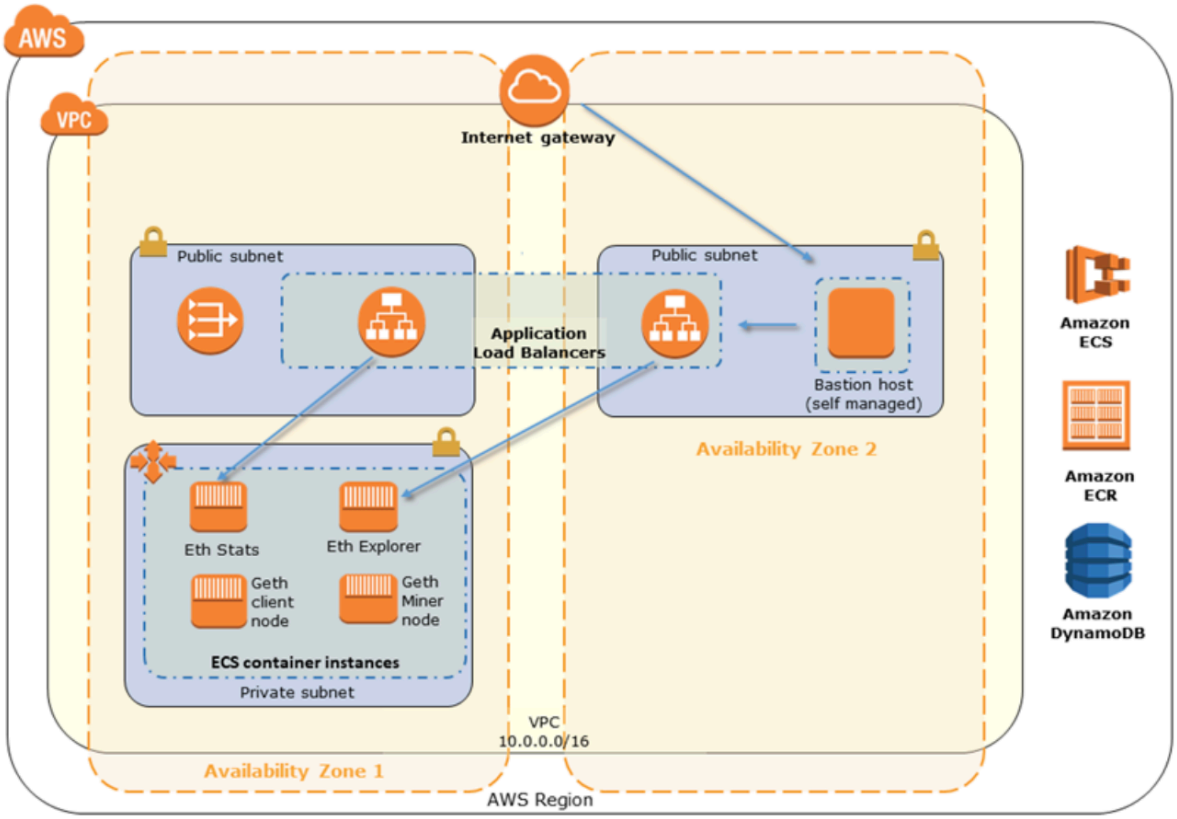
July, 2018

Amazon Web Services

Table of Contents

Overview

you create your Ethereum network on an ECS cluster composed of multiple EC2 instances, with an Application Load Balancer and related resources. The following diagram depicts a Ethereum network created using the template with the ECS container platform option:



0 Set Up Basic

Before you start with AWS Blockchain Templates, complete the following tasks:

Sign Up for AWS
Create an IAM User
Create a Key Pair

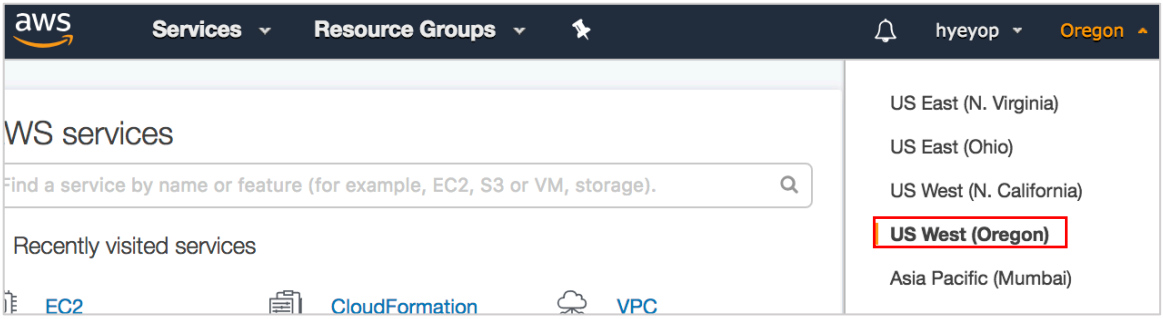
1 Set Up Prerequisites

※ For more information, see the Getting Started with AWS Blockchain Templates.

1. Create a VPC and Subsets

To select the Region

1. Open the Amazon VPC console at https://console.aws.amazon.com/.
2. Choose **US West(Oregon) region**.

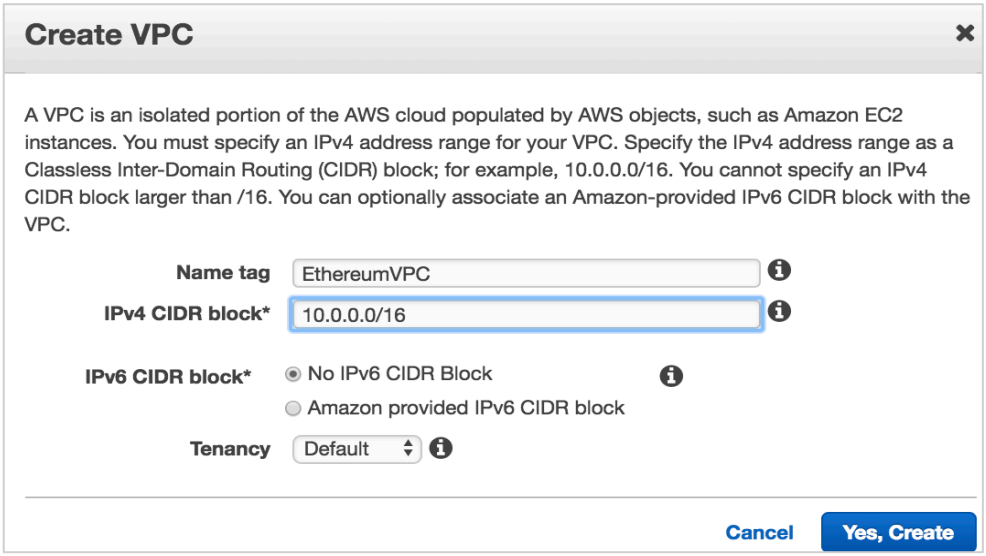


To create an Elastic IP address

1. Open the Amazon VPC console at https://console.aws.amazon.com/vpc/.
2. Choose **Elastic IPs**, **Allocate new address**, **Allocate**, **Close**.

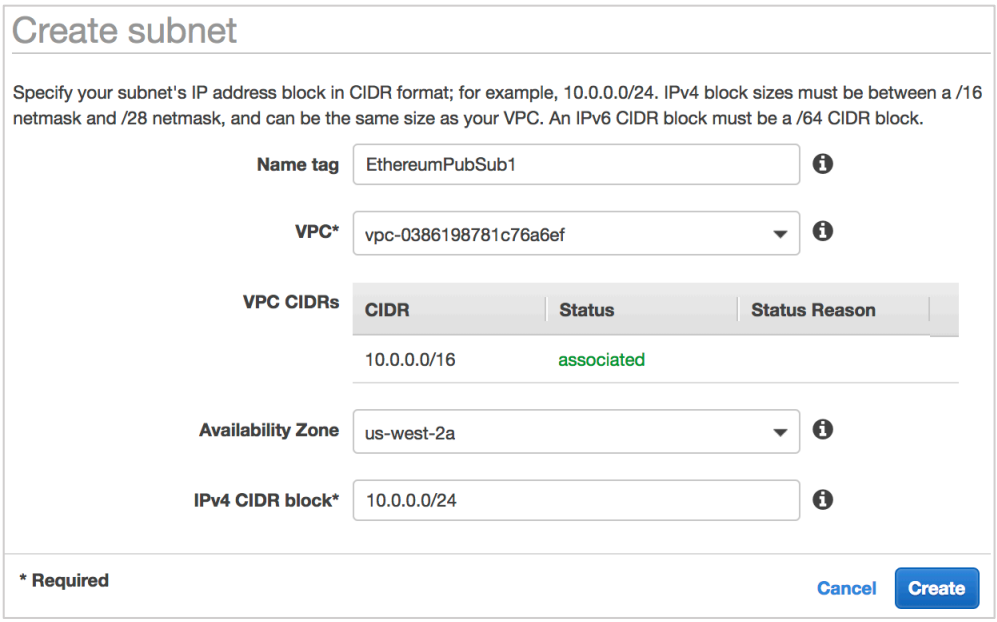
To create the VPC and Subnet

1. On the **Your VPCs**, click **Create the VPC**.



- **Name tag :** EthereumVPC
- **IPv4 CIDR:** 10.0.0.0/16

2. On **Subnets**, click **Create subnet**. (Total 3)



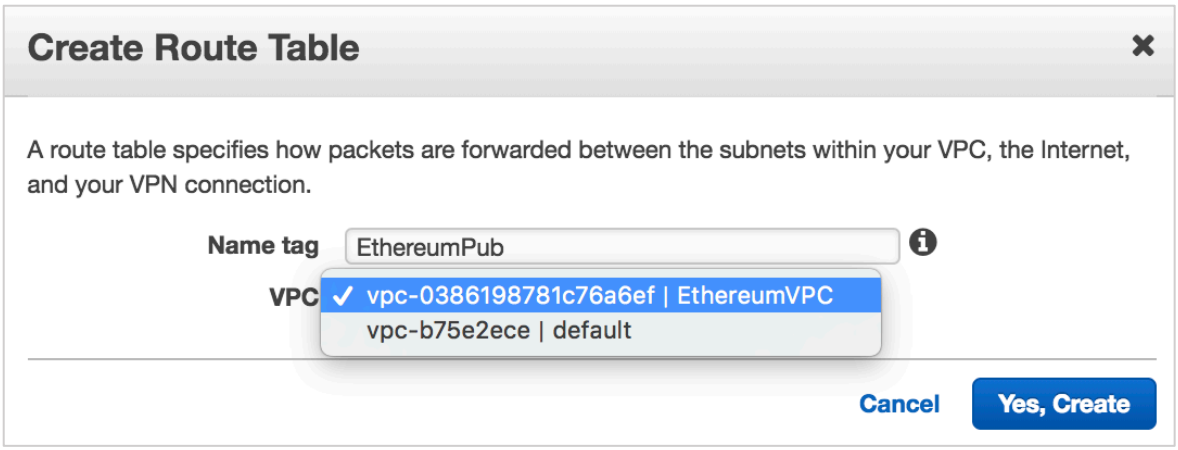
Total 3 subnets are described below:

Public Subnet1	▪ Name tag : EthereumPubSub1 ▪ VPC : (Ethereum VPC 선택) ▪ Availability Zone : us-west-2a ▪ IPv4 CIDR Zone : 10.0.0.0/24
Public Subnet2	▪ Name tag : EthereumPubSub2 ▪ VPC : (Ethereum VPC 선택) ▪ Availability Zone : us-west-2b ▪ IPv4 CIDR Zone : 10.0.1.0/24
Private Subnet	▪ Name tag : EthereumPvtSub1 ▪ VPC : (Ethereum VPC 선택) ▪ Availability Zone : us-west-2a ▪ IPv4 CIDR Zone : 10.0.2.0/24

To create Internet G/W, Public Route Table

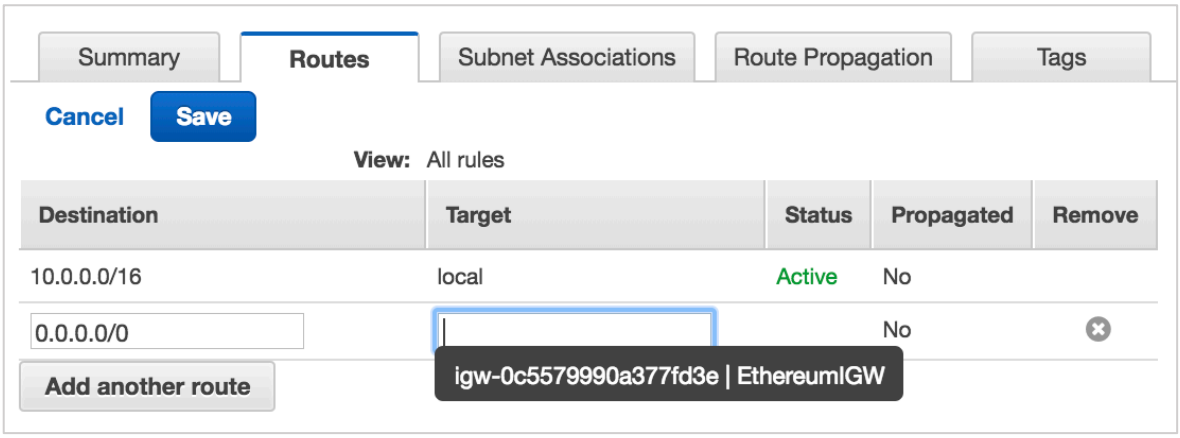
3. On **Internet Gateways**, click **Create Internet gateway**.
   - **Name tag :** EthereumIGW

4. Choose EthereumIGW on list and click **Actions**, **Attach to VPC, Attach** with EthereumVPC.
5. Click **Route Tables**, click **Create Route Table**.



- **Name tag :** EthereumPub
- **VPC :** EthereumVPC

6. Click **Routes** tab, **Edit, Add another route.**

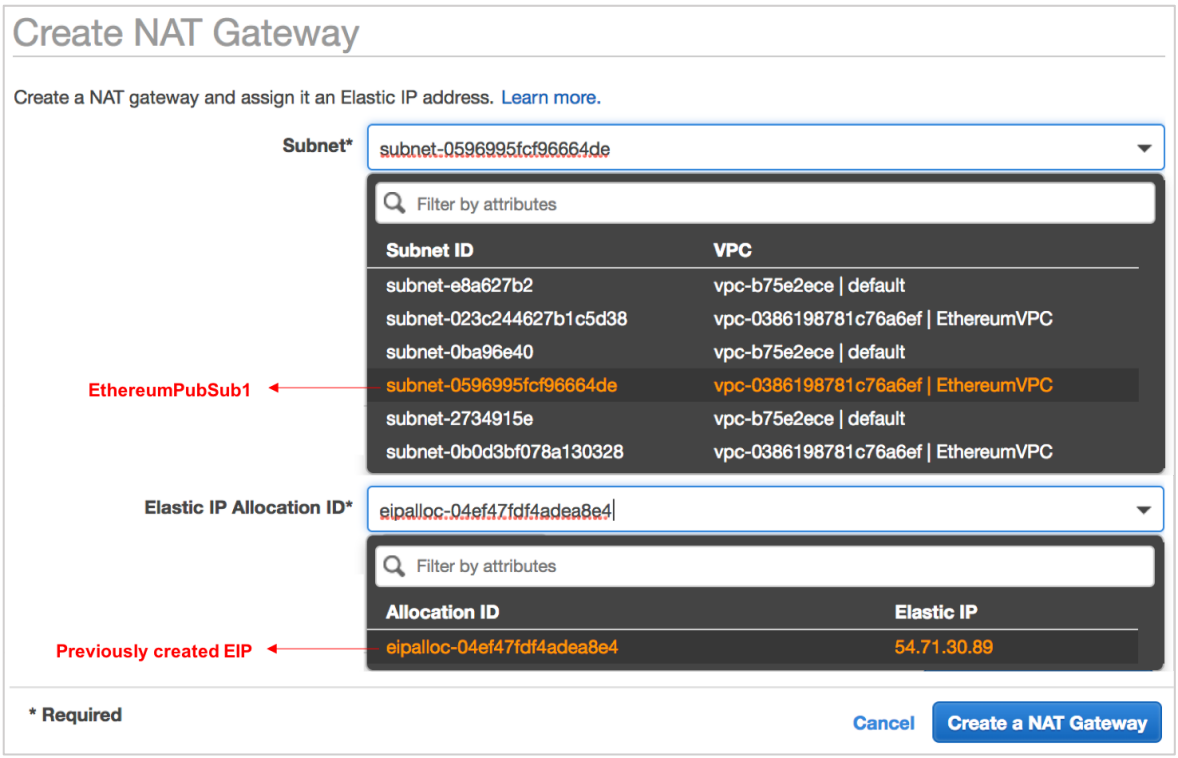


- **Destination :** 0.0.0.0/0
- **VPC :** EthereumIGW

7. Click **Subnet Assoiciations** tab and **Edit**.
   Select **EthereumPubSub1, EthereumPubSub2** and click **Save**.

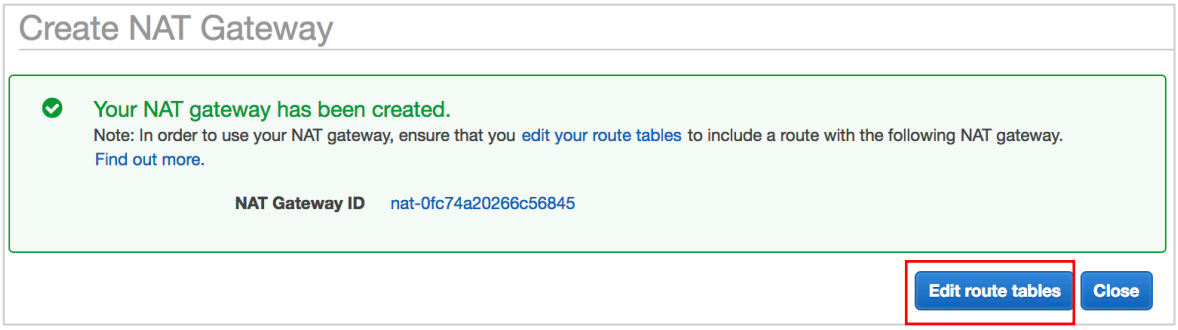
To create NAT Gateways, Private Route Table

8. On **NAT Gateways**. click **Create a NAT Gateway.**
   click **Create a Nat Gateway** With Previously created EthereumPubSub1 and EIP.

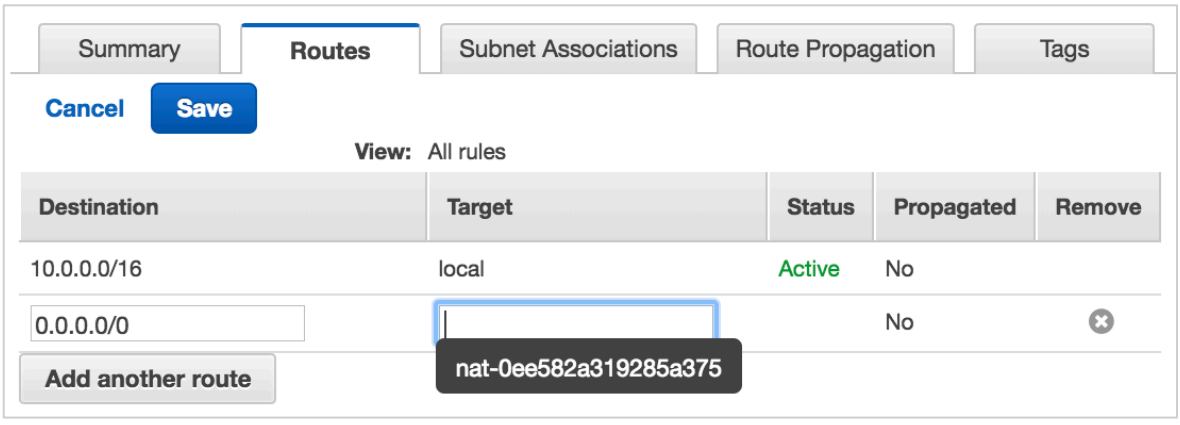


- **Subnet :** EthereumPubSub1
- **VPC :** EIP

9. Choose **Edit route tables**.



10. Click **Routes** tab, **Edit, Add another route, Save.**



- **Destination :** 0.0.0.0/0
- **VPC :** NAT Gateway (nat-xxxxxxxxx)

11. Click **Subnet Assoiciations** tab and **Edit**.
    Select **EthereumPvtSub1** and click **Save**.

2. Create the Security Groups

To create two security groups

1. Open the Amazon EC2 console at https://console.aws.amazon.com/ec2/.
2. In the navigation pane, choose **Security Groups**, **Create Security Group**, **Save.**

- **Security group name :** EthereumEC2-SG
- **VPC** : EthereumVPC

- **Security group name :** EthereumALB-SG
- **VPC** : EthereumVPC

Add inbound rules

1. Select **EthereumEC2-SG** from the list. On the **Inbound** tab, choose **Edit**.
   Total 2 inbound rules are described below :

- **Type :** All traffic
- **Source** : EthereumEC2-SG Group ID (sg-xxxxxx)

- **Type :** All traffic
- **Source** : EthereumALB-SG Group ID (sg-yyyyy)

2. Select **EthereumALB-SG** from the list. On the **Inbound** tab, choose **Edit**.
   Total 6 inbound rules are described below :

- **Type :** All traffic
- **Source** : EthereumEC2-SG Group ID (sg-xxxxxx)

- **Type :** All traffic
- **Source** : EthereumALB-SG Group ID (sg-yyyyy)

- **Type :** SSH
- **Source** : My IP

- **Type :** HTTP
- **Source** : 0.0.0.0/0

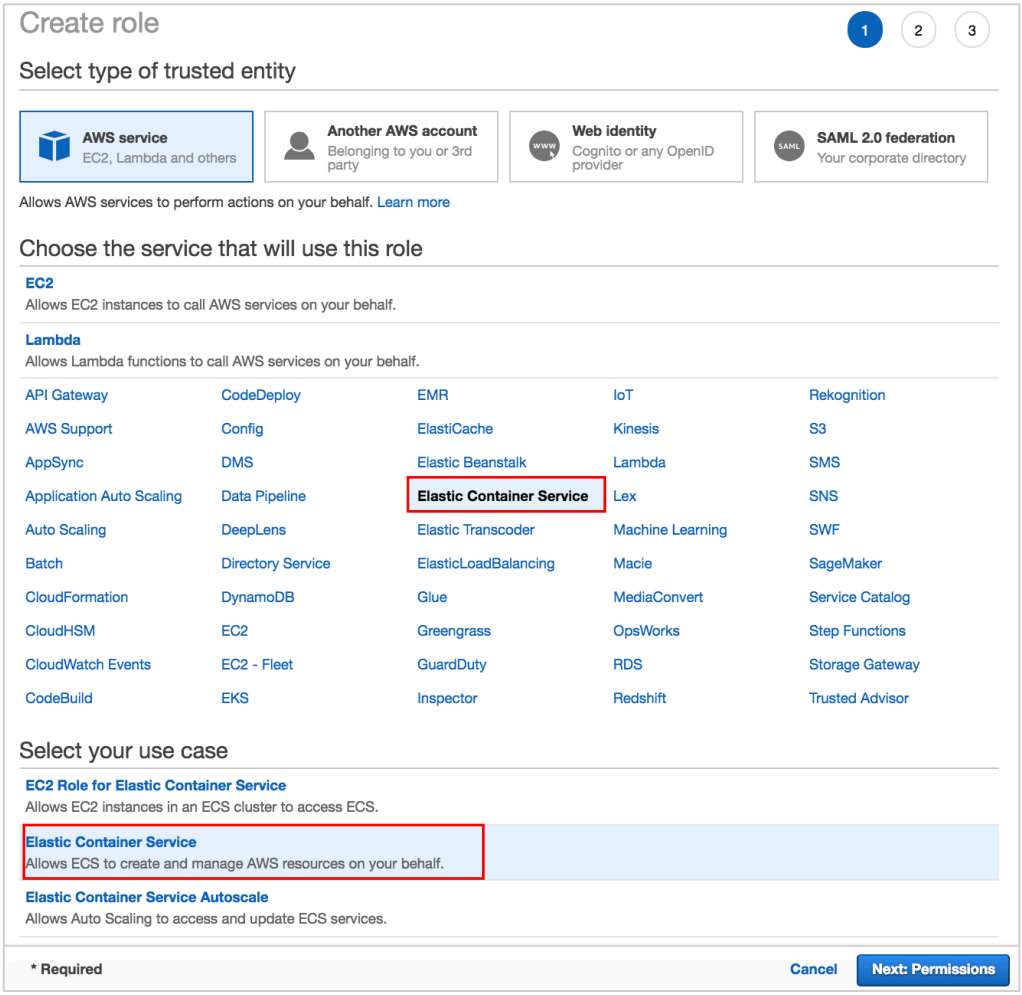
- **Type :** Custom TCP
- **Port Range** : 8080
- **Source** : My IP

- **Type :** Custom TCP
- **Port Range** :8545
- **Source** : My IP

3. Create IAM Role

To create the IAM role for ECS

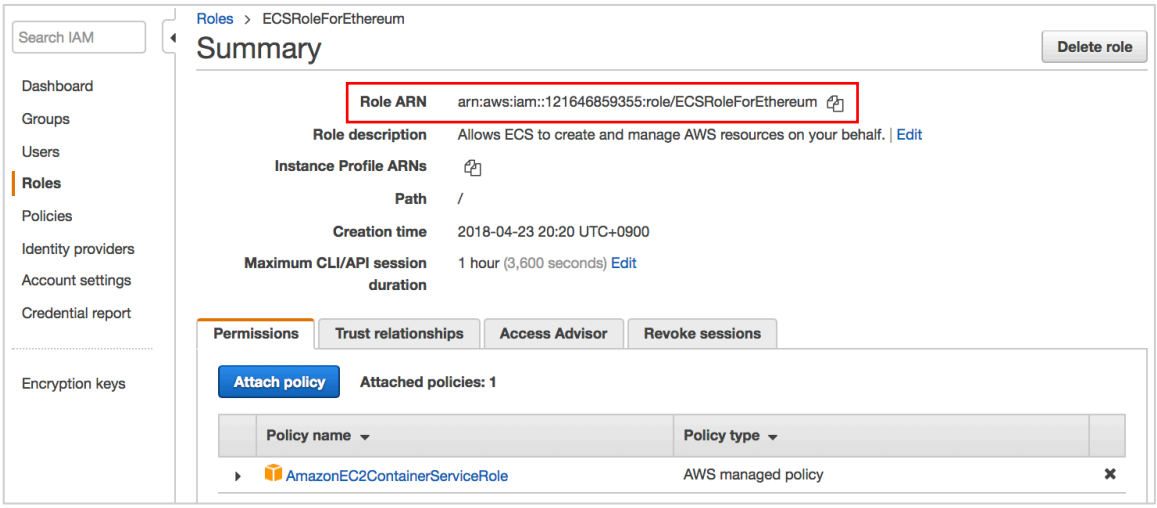
1. Open the IAM console at https://console.aws.amazon.com/iam/.
2. In the navigation pane, choose **Roles**, **Create Role**.
3. Under **Select type of trusted entity**, choose **AWS service**.
4. For **Choose the service that will use this role**, choose **Elastic Container Service**.
   Under **Select your use case**, choose **Elastic Container Service**, **Next:Permissions.**



5. For **Permissions policy**, leave the default policy (**AmazonEC2ContainerServiceRole**) selected, and choose **Next:Review**.
6. For **Role name**, enter a value that helps you identify the role. Choose **Create role**.
   - **Role Name :** ECSRoleForEthereum

7. Select **ECSRoleForEthereum** that you just created from the list. Copy the **Role ARN** value.

   ※ You need this **Role ARN** when you create the Ethereum network.



To create the EC2 instance profile

8. In the navigation pane, choose **Policies**, **Create policy**.
9. Choose **JSON** and replace the default policy statement with the following JSON policy:

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "ecs:CreateCluster",
        "ecs:DeregisterContainerInstance",
        "ecs:DiscoverPollEndpoint",
        "ecs:Poll",
        "ecs:RegisterContainerInstance",
        "ecs:StartTelemetrySession",
        "ecs:Submit*",
        "ecr:GetAuthorizationToken",
        "ecr:BatchCheckLayerAvailability",
        "ecr:GetDownloadUrlForLayer",
        "ecr:BatchGetImage",
        "logs:CreateLogStream",
        "logs:PutLogEvents",
```

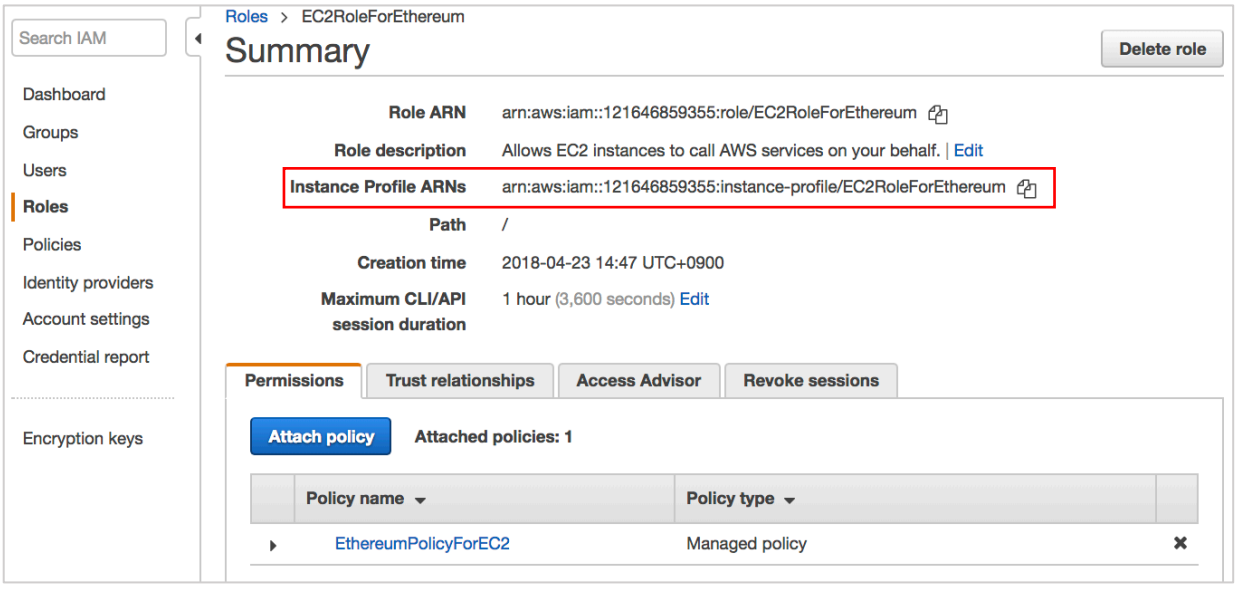
```
            "dynamodb:BatchGetItem",
            "dynamodb:BatchWriteItem",
            "dynamodb:PutItem",
            "dynamodb:DeleteItem",
            "dynamodb:GetItem",
            "dynamodb:Scan",
            "dynamodb:Query",
            "dynamodb:UpdateItem"
        ],
        "Resource": "*"
    }
  ]
}
```

10. Choose **Review policy**.
11. Enter Name, Description. Choose **Create policy**
    - **Name :** EthereumPolicyForEC2

12. Choose **Roles**, **Create role**.
13. Choose **EC2**, **Next: Permissions**.
14. In the **Search** field, enter EthereumPolicyForEC2.
15. Select it and choose **Next : Review**
16. enter Role Name, Description. Choose **Create role**.
    - **Role Name :** EC2RoleForEthereum

17. Select **ECSRoleForEthereum** that you just created from the list. Copy the **Instance Profile ARN** value.

    ※ You need **EC2 Instance Profile ARN** when you create the Ethereum network.



4. Create a Bastion Host

1. Open the IAM console at https://console.aws.amazon.com/ec2/.
2. Choose Launch Instance.

Step2	▪ **Type** : T2.micro
Step3	▪ **Network** : (EthereumVPC) ▪ **Subnet** : (EthereumPubSub1) ▪ **Auto-assign Public IP** : Enable
Step5	▪ **Key :** Name ▪ **Value :** EthereumBastion
Step6	▪ **Assign a security group :** ⦿ Select an **existing** security group (EthereumALB-SG)

2 Create the Ethereum Network

To create the Ethereum Network

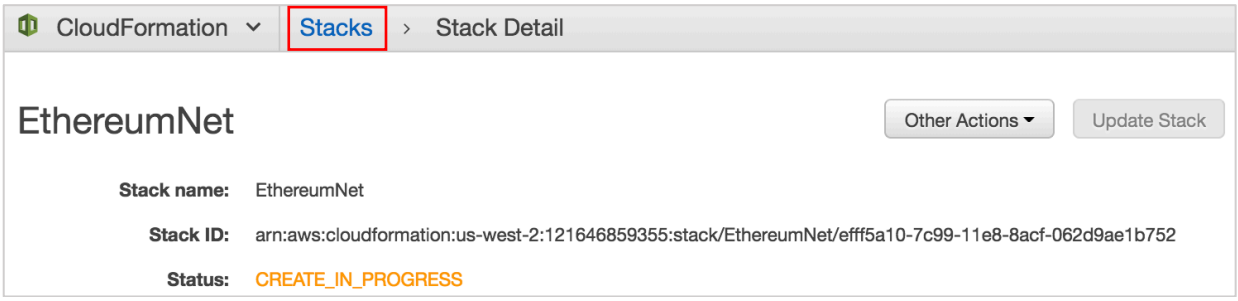
1. Open the latest AWS Blockchain Template for Ethereum in the AWS CloudFormation console
   ▪ Launch in US West (Oregon) region (us-west-2)

2. Enter values according to the following guidelines :

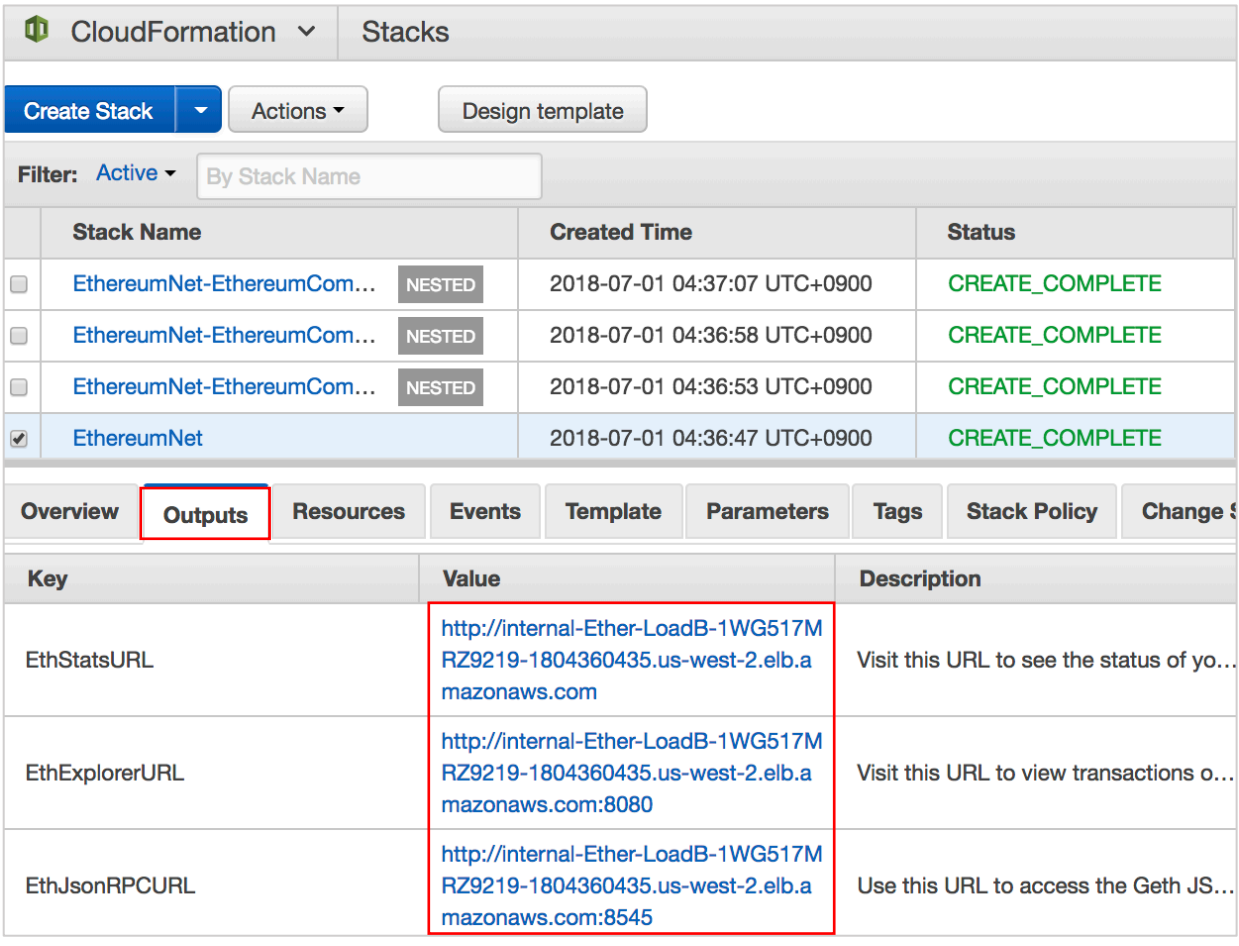
Details	▪ **Stack Name** : TestNet
Ethereum network parameters	default
Private Ethereum network parameters	default
Platform configuration	default
VPC network configuration	▪ **VPC ID** : EthereumVPC ▪ **List of VPC Subnets to use** : EthereumPvtSub1 ▪ **ALB Subnet IDs :** EthereumPutSub1, EthereumPubSub2
Security Configuration	▪ **EC2 Key Pair** :

	▪ **EC2 Security Group** : EthereumEC2-SG ▪ **IAM Role :** ▪ **EC2 Instance Profile ARN :** ▪ **Application Load Balancer Security Group** : EthereumALB-SG
ECS cluster configuration	default
EthStats	▪ **Deploy EthStats** : true ▪ **EthStats Connection Secret** : (at least 6 character)

3. Choose **Create**. Choose Stacks to monitor the progress of the stacks.



4. When all stacks show **CREATE_COMPLETE** for **Status**, click **Outputs** tab of the root stack.



3 Connect to the Ethereum Network using the Bastion

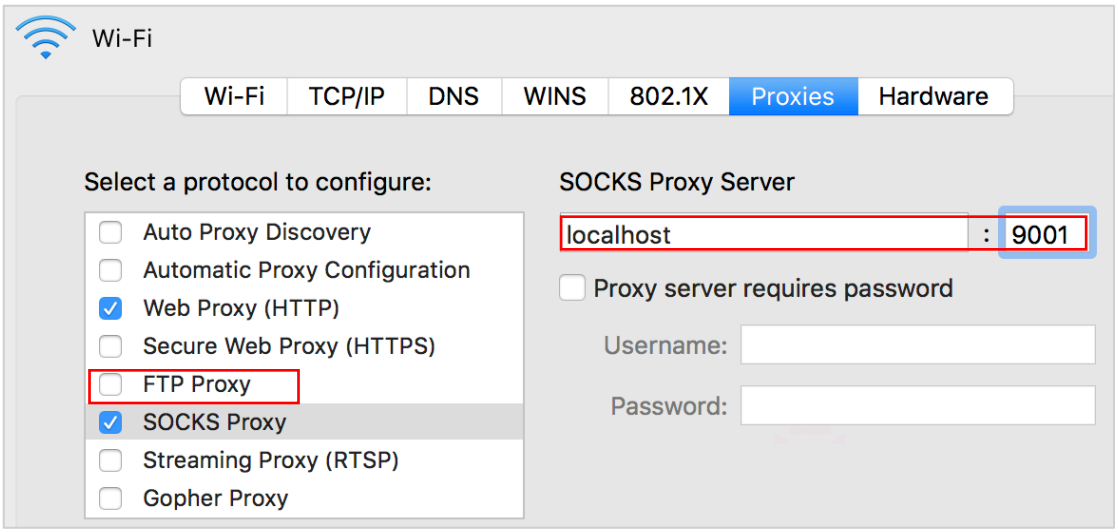
※ For more information, see the Connect to Your Instanace.

(Mac OSX) To connect to the bastion with SSH port forwarding using SSH

1. When connecting to your Bastion, add -D 9001 to SSH command.

```
ssh -i path/key-pair.pem ec2-user@bastion-dns -D 9001
```

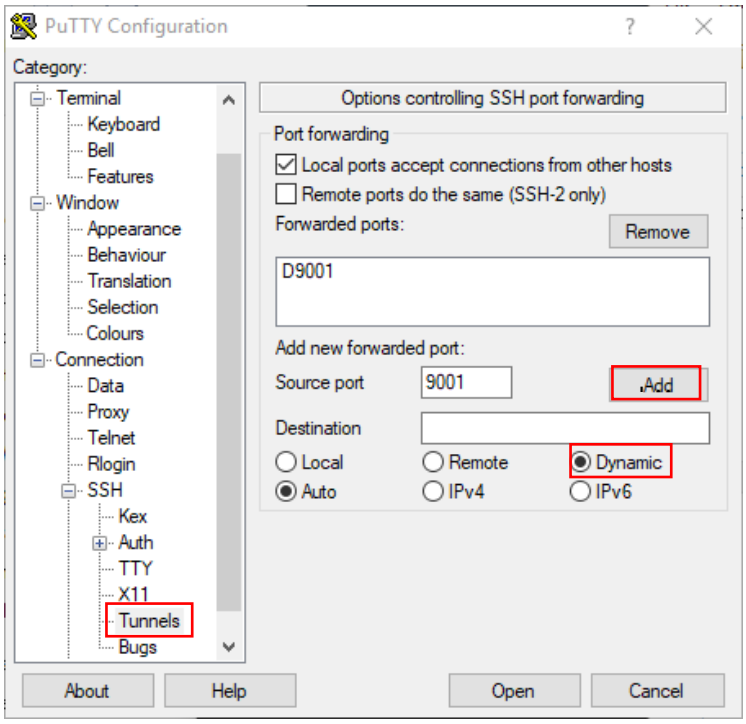
2. To configure your browser to use SOCKS proxy for localhost:9001 select System Preferences, Network, Advanced, select SOCKS proxy, and type **localhost:9001**.



※Using FoxyProxy Standard with Chrome,
select **More Tools**, **Extensions**. Under **FoxyProxy Standard**, select **Details**, **Extension options**, **Add New Proxy**. Select **Manual Proxy Configuration**. For **Host or IP Address** type **localhost** and for **Port** type **9001**. Select **SOCKS proxy, Save**.

(Windows) To connect to the bastion with SSH port forwarding using PuTTY

1. Starting a PuTTY session, add **Connection**, **SSH**, **Tunnels**.
2. For **Port forwarding**, choose **Local ports accept connections from other hosts**.
3. Under **Add new forwarded port :**



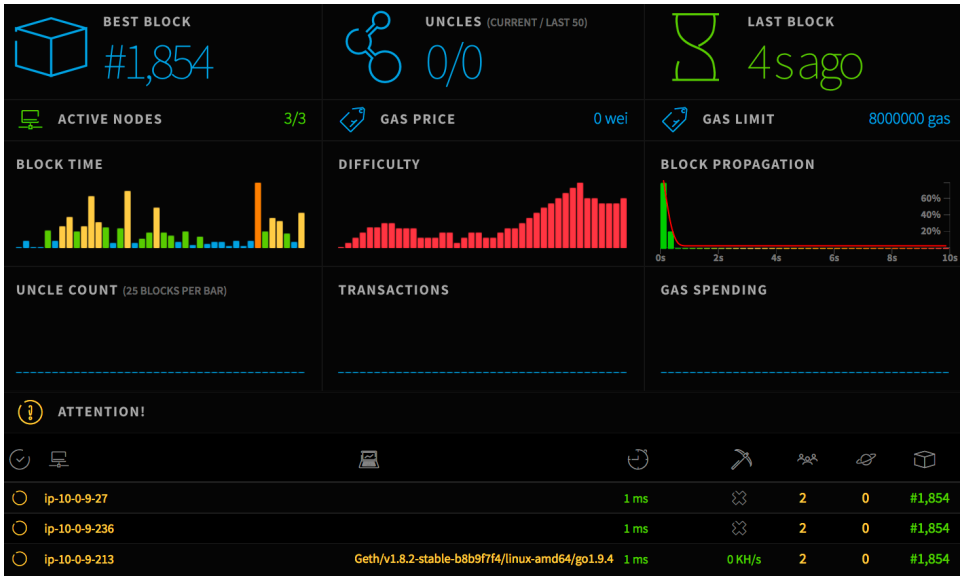
- **Source port**, enter **9001**.
- Leave **Destination** blank.
- Select **Dynamic**.
- Choose **Add**.

Choose **Open**.

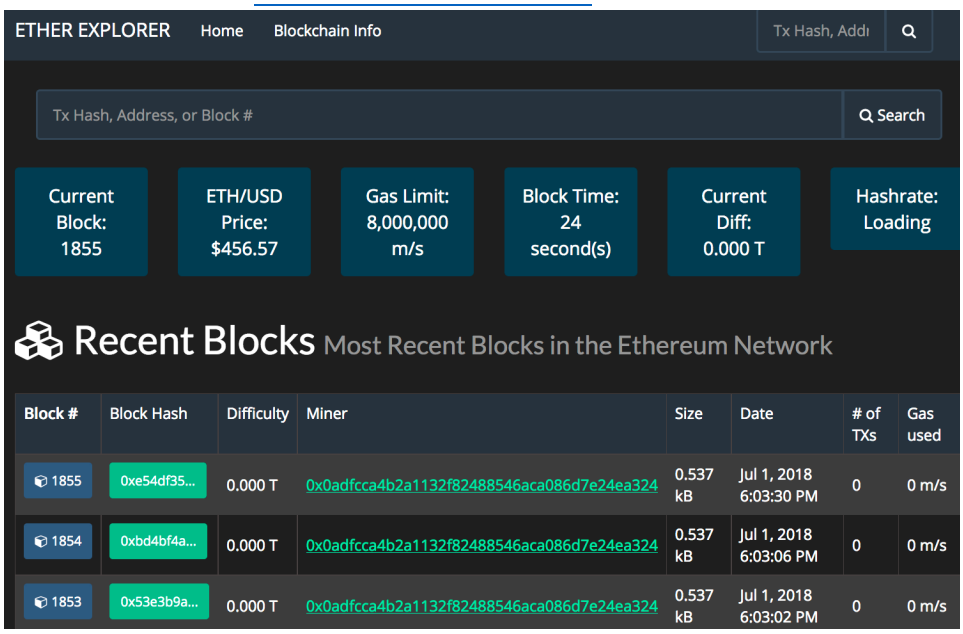
To check the results of the Ethereum Network

You are now able to connect to the Ethereum URLs, which are available on CloudFormation console using the **Outputs** tab of the root stack that you created with the template.

- **EthStats Web** : http://<ALB-DNS>



- **EthStats Web** : http://<ALB-DNS>:8080



- **EthJsonRPC Call** : http://<ALB-DNS>:8545

```
$ curl -X POST -d
'{"jsonrpc":"2.0","method":"eth_getBalance","params":["0x0ADfCCa4B2a1132F82488546AcA086D7E24EA324", "latest"],"id":1}' -H 'Content-Type:application/json' http://<ALB-DNS>:8545

{"jsonrpc":"2.0","id":1,"result":"0x282a92ca9849fdc0000"}

$ curl -X POST -d '{"jsonrpc":"2.0","method":"eth_blockNumber","params":[], "id":1}' -H 'Content-Type:application/json' http://<ALB-DNS>:8545

{"jsonrpc":"2.0","id":1,"result":"0x865"}
```