



Etherem BlockChain Template Hands-On Lab

July, 2018

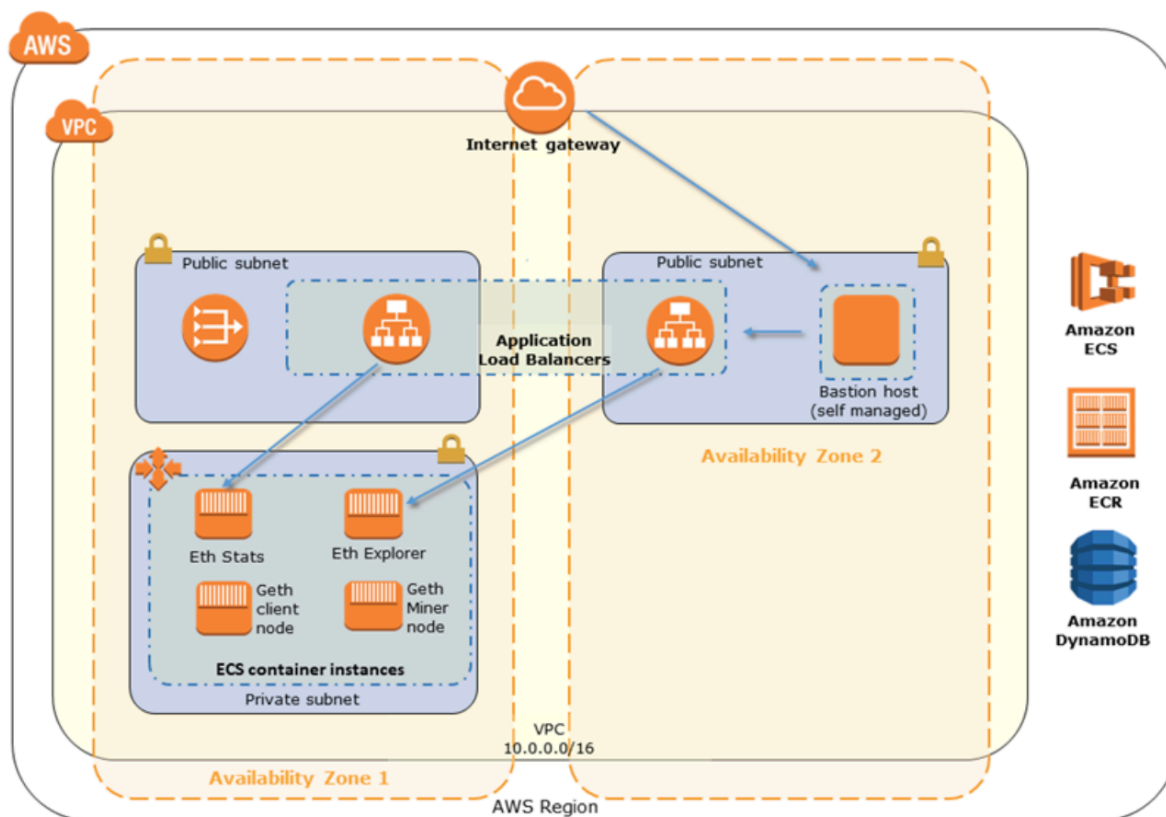
Amazon Web Services

Table of Contents

Overview	Error! Bookmark not defined.
Set Up Basic.....	3
Set Up Prerequisites.....	4
Create the VPC and Subnets	4
Create the Security Groups	8
Create IAM User Role.....	9
Create a Bastion Host	12
Create the Ethereum Network	12
Connect to the Ethereum Network Using the Bastion.....	Error! Bookmark not defined.

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](#)

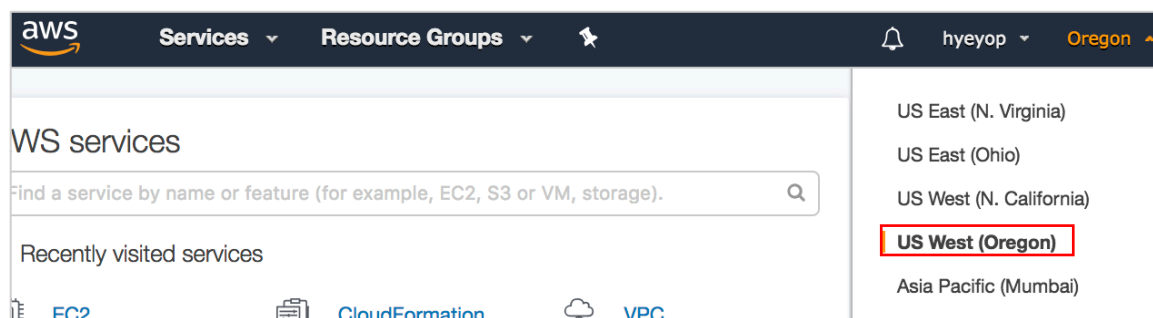
1 Set Up Prerequisites

※ For more information, see the [Getting Started with AWS Blockchain Templates](#).

1. Create a VPC and Subnets

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**.

 A screenshot of the 'Create VPC' dialog box. The dialog has a title bar 'Create VPC' with a close button. Below the title bar is a descriptive text: 'A VPC is an isolated portion of the AWS cloud populated by AWS objects, such as Amazon EC2 instances. You must specify an IPv4 address range for your VPC. Specify the IPv4 address range as a Classless Inter-Domain Routing (CIDR) block; for example, 10.0.0.0/16. You cannot specify an IPv4 CIDR block larger than /16. You can optionally associate an Amazon-provided IPv6 CIDR block with the VPC.' Below this text are three input fields: 'Name tag' with the value 'EthereumVPC', 'IPv4 CIDR block*' with the value '10.0.0.0/16', and 'IPv6 CIDR block*' with the radio button 'No IPv6 CIDR Block' selected. There is also a 'Tenancy' dropdown set to 'Default'. At the bottom right are 'Cancel' and 'Yes, Create' buttons.

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

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

Create subnet

Specify your subnet's IP address block in CIDR format; for example, 10.0.0.0/24. IPv4 block sizes must be between a /16 netmask and /28 netmask, and can be the same size as your VPC. An IPv6 CIDR block must be a /64 CIDR block.

Name tag

VPC*

vpc-0386198781c76a6ef

VPC CIDRs

CIDR	Status	Status Reason
10.0.0.0/16	associated	

Availability Zone

us-west-2a

IPv4 CIDR block*

* Required

Cancel

Create

Total 3 subnets are described below:

Public Subnet1	<ul style="list-style-type: none"> Name tag : EthereumPubSub1 VPC : (Ethereum VPC 선택) Availability Zone : us-west-2a IPv4 CIDR Zone : 10.0.0.0/24
Public Subnet2	<ul style="list-style-type: none"> Name tag : EthereumPubSub2 VPC : (Ethereum VPC 선택) Availability Zone : us-west-2b IPv4 CIDR Zone : 10.0.1.0/24
Private Subnet	<ul style="list-style-type: none"> 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**.

Create Route Table ✕

A route table specifies how packets are forwarded between the subnets within your VPC, the Internet, and your VPN connection.

Name tag ?

VPC ✓

Cancel Yes, Create

- Name tag : EthereumPub
- VPC : EthereumVPC

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

Summary **Routes** Subnet Associations Route Propagation Tags

Cancel Save

View: All rules

Destination	Target	Status	Propagated	Remove
10.0.0.0/16	local	Active	No	
<input data-bbox="220 1137 494 1176" type="text" value="0.0.0.0/0"/>	<input data-bbox="630 1176 1093 1220" type="text" value="igw-0c5579990a377fd3e EthereumIGW"/>		No	✕

Add another route

- Destination : 0.0.0.0/0
- VPC : EthereumIGW

7. Click **Subnet Associations** 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**.

Create NAT Gateway

Create a NAT gateway and assign it an Elastic IP address. [Learn more.](#)

Subnet*

Elastic IP Allocation ID*

Subnet ID **VPC**

subnet-e8a627b2	vpc-b75e2ece default
subnet-023c244627b1c5d38	vpc-0386198781c76a6ef EthereumVPC
subnet-0ba96e40	vpc-b75e2ece default
subnet-0596995fc96664de	vpc-0386198781c76a6ef EthereumVPC
subnet-2734915e	vpc-b75e2ece default
subnet-0b0d3bf078a130328	vpc-0386198781c76a6ef EthereumVPC

Allocation ID **Elastic IP**

eipalloc-04ef47fdf4adea8e4	54.71.30.89
-----------------------------------	--------------------

* Required

[Cancel](#) [Create a NAT Gateway](#)

- Subnet : EthereumPubSub1
- VPC : EIP

9. Choose Edit route tables.

Create NAT Gateway

✓ Your NAT gateway has been created.
Note: In order to use your NAT gateway, ensure that you [edit your route tables](#) to include a route with the following NAT gateway.
[Find out more.](#)

NAT Gateway ID nat-0fc74a20266c56845

[Edit route tables](#) [Close](#)

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

[Summary](#) **[Routes](#)** [Subnet Associations](#) [Route Propagation](#) [Tags](#)

[Cancel](#) [Save](#)

View: All rules

Destination	Target	Status	Propagated	Remove
10.0.0.0/16	local	Active	No	
0.0.0.0/0	<input type="text" value="nat-0ee582a319285a375"/>	No	No	✕

[Add another route](#)

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

- Click **Subnet Associations** tab and **Edit**.
Select **EthereumPvtSub1** and click **Save**.

2. Create the Security Groups

To create two security groups

- Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
- 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

- 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)
-

- 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**.

Create role

1 2 3

Select type of trusted entity

AWS service
EC2, Lambda and others

Another AWS account
Belonging to you or 3rd party

Web identity
Cognito or any OpenID provider

SAML 2.0 federation
Your corporate directory

Allows AWS services to perform actions on your behalf. [Learn more](#)

Choose the service that will use this role

EC2
Allows EC2 instances to call AWS services on your behalf.

Lambda
Allows Lambda functions to call AWS services on your behalf.

API Gateway	CodeDeploy	EMR	IoT	Rekognition
AWS Support	Config	ElastiCache	Kinesis	S3
AppSync	DMS	Elastic Beanstalk	Lambda	SMS
Application Auto Scaling	Data Pipeline	Elastic Container Service	Lex	SNS
Auto Scaling	DeepLens	Elastic Transcoder	Machine Learning	SWF
Batch	Directory Service	ElasticLoadBalancing	Macie	SageMaker
CloudFormation	DynamoDB	Glue	MediaConvert	Service Catalog
CloudHSM	EC2	Greengrass	OpsWorks	Step Functions
CloudWatch Events	EC2 - Fleet	GuardDuty	RDS	Storage Gateway
CodeBuild	EKS	Inspector	Redshift	Trusted Advisor

Select your use case

EC2 Role for Elastic Container Service
Allows EC2 instances in an ECS cluster to access ECS.

Elastic Container Service
Allows ECS to create and manage AWS resources on your behalf.

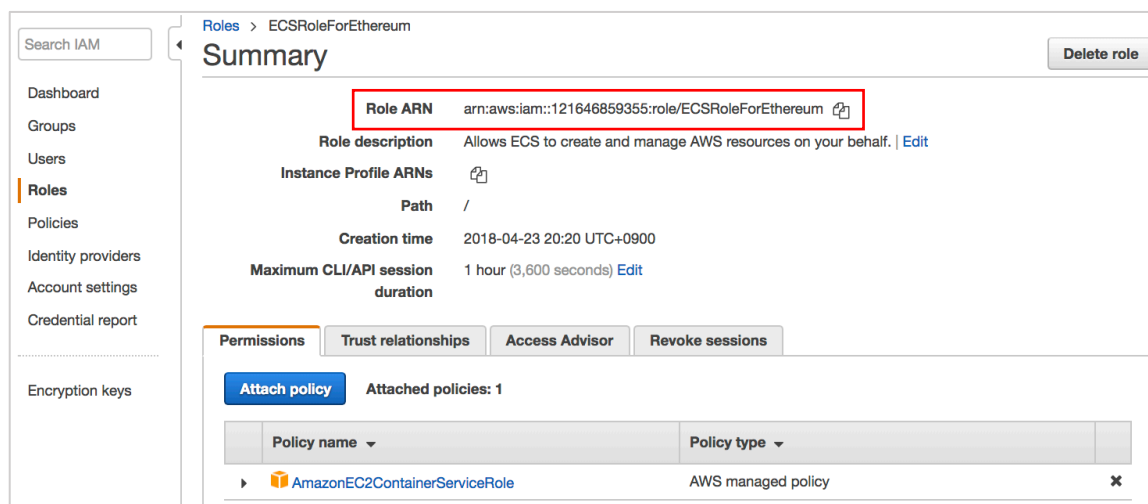
Elastic Container Service Autoscale
Allows Auto Scaling to access and update ECS services.

* Required

Cancel

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.



Search IAM

Roles > ECSRoleForEthereum

Summary Delete role

Role ARN `arn:aws:iam::121646859355:role/ECSRoleForEthereum`

Role description Allows ECS to create and manage AWS resources on your behalf. [Edit](#)

Instance Profile ARNs [Add](#)


Path /

Creation time 2018-04-23 20:20 UTC+0900

Maximum CLI/API session duration 1 hour (3,600 seconds) [Edit](#)

Permissions **Trust relationships** **Access Advisor** **Revoke sessions**

[Attach policy](#) Attached policies: 1

Policy name	Policy type
 AmazonEC2ContainerServiceRole	AWS managed policy

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.

The screenshot shows the AWS IAM console interface. On the left is a navigation menu with options like Dashboard, Groups, Users, Roles, Policies, Identity providers, Account settings, Credential report, and Encryption keys. The main area displays the 'Summary' for the role 'EC2RoleForEthereum'. Key details include:

- Role ARN:** arn:aws:iam::121646859355:role/EC2RoleForEthereum
- Role description:** Allows EC2 instances to call AWS services on your behalf.
- Instance Profile ARNs:** arn:aws:iam::121646859355:instance-profile/EC2RoleForEthereum (highlighted with a red box)
- Path:** /
- Creation time:** 2018-04-23 14:47 UTC+0900
- Maximum CLI/API session duration:** 1 hour (3,600 seconds)

Below the summary, there are tabs for 'Permissions', 'Trust relationships', 'Access Advisor', and 'Revoke sessions'. The 'Permissions' tab is active, showing an 'Attach policy' button and a list of 'Attached policies: 1'. The policy listed is 'EthereumPolicyForEC2' of type 'Managed policy'.

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 : <input checked="" type="radio"/> 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 :

	<ul style="list-style-type: none"> ▪ EC2 Security Group : EthereumEC2-SG ▪ IAM Role : ▪ EC2 Instance Profile ARN : ▪ Application Load Balancer Security Group : EthereumALB-SG
ECS cluster configuration	default
EthStats	<ul style="list-style-type: none"> ▪ Deploy EthStats : true ▪ EthStats Connection Secret : (at least 6 character)

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

CloudFormation ▾ **Stacks** > Stack Detail

EthereumNet

Other Actions ▾ Update Stack

Stack name: EthereumNet

Stack ID: arn:aws:cloudformation:us-west-2:121646859355:stack/EthereumNet/eff5a10-7c99-11e8-8acf-062d9ae1b752

Status: CREATE_IN_PROGRESS

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

CloudFormation ▾ **Stacks**

Create Stack ▾ Actions ▾ Design template

Filter: Active ▾ By Stack Name

	Stack Name	Created Time	Status
<input type="checkbox"/>	EthereumNet-EthereumCom... NESTED	2018-07-01 04:37:07 UTC+0900	CREATE_COMPLETE
<input type="checkbox"/>	EthereumNet-EthereumCom... NESTED	2018-07-01 04:36:58 UTC+0900	CREATE_COMPLETE
<input type="checkbox"/>	EthereumNet-EthereumCom... NESTED	2018-07-01 04:36:53 UTC+0900	CREATE_COMPLETE
<input checked="" type="checkbox"/>	EthereumNet	2018-07-01 04:36:47 UTC+0900	CREATE_COMPLETE

Overview **Outputs** Resources Events Template Parameters Tags Stack Policy Change S

Key	Value	Description
EthStatsURL	http://internal-Ether-LoadB-1WG517MRZ9219-1804360435.us-west-2.elb.amazonaws.com	Visit this URL to see the status of yo...
EthExplorerURL	http://internal-Ether-LoadB-1WG517MRZ9219-1804360435.us-west-2.elb.amazonaws.com:8080	Visit this URL to view transactions o...
EthJsonRPCURL	http://internal-Ether-LoadB-1WG517MRZ9219-1804360435.us-west-2.elb.amazonaws.com:8545	Use this URL to access the Geth JS...

3 Connect to the Ethereum Network using the Bastion

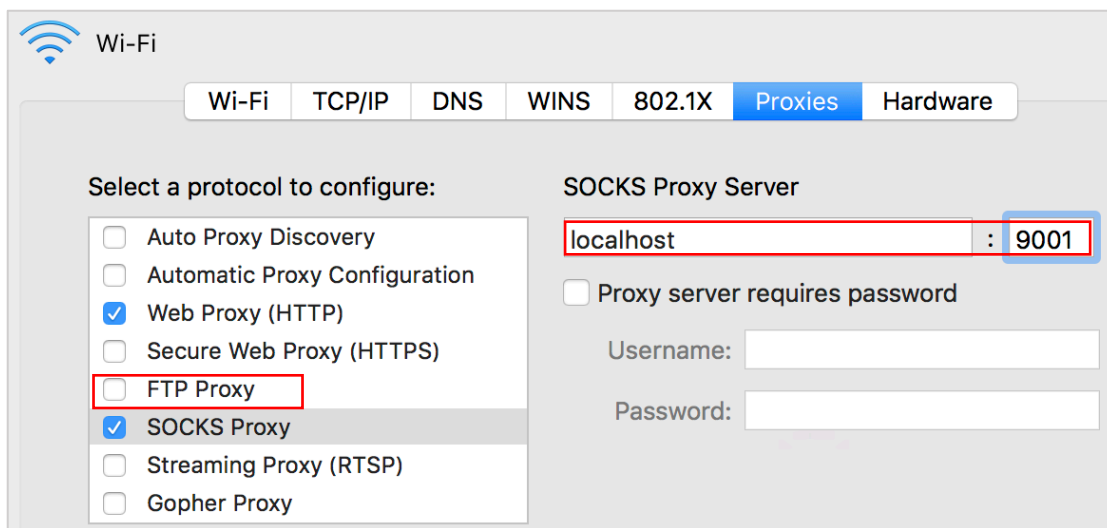
※ For more information, see the [Connect to Your Instance](#).

(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**.

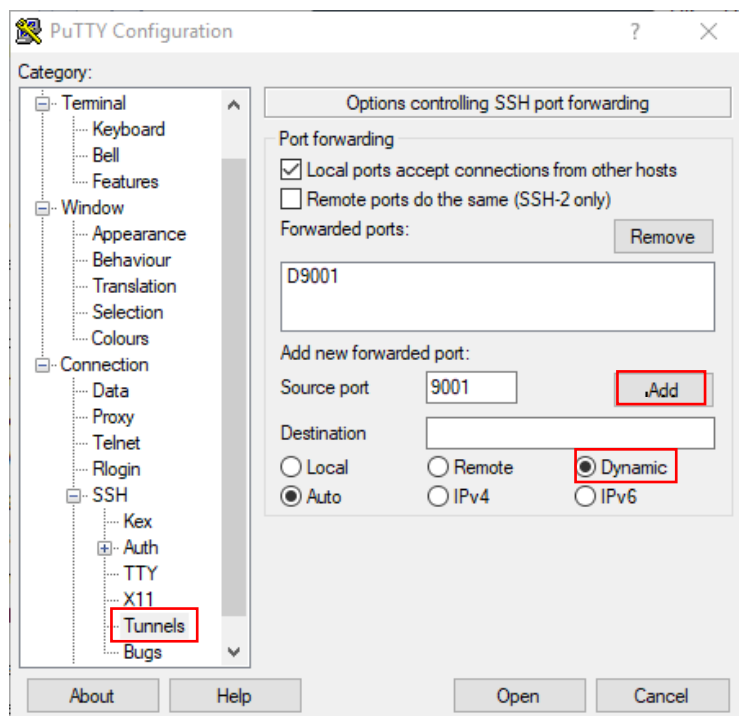


※Using FoxyProxy Standard with Chrome,
select **More Tools, Extensions**. Under **FoxyProxy Standard**, select **Details, Extension options, Add New Proxy**. Select **Manual Proxy Configuration**. For **Host or IP Address** type **localhost** and for **Port** type **9001**. Select **SOCKS proxy, Save**.



(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** :



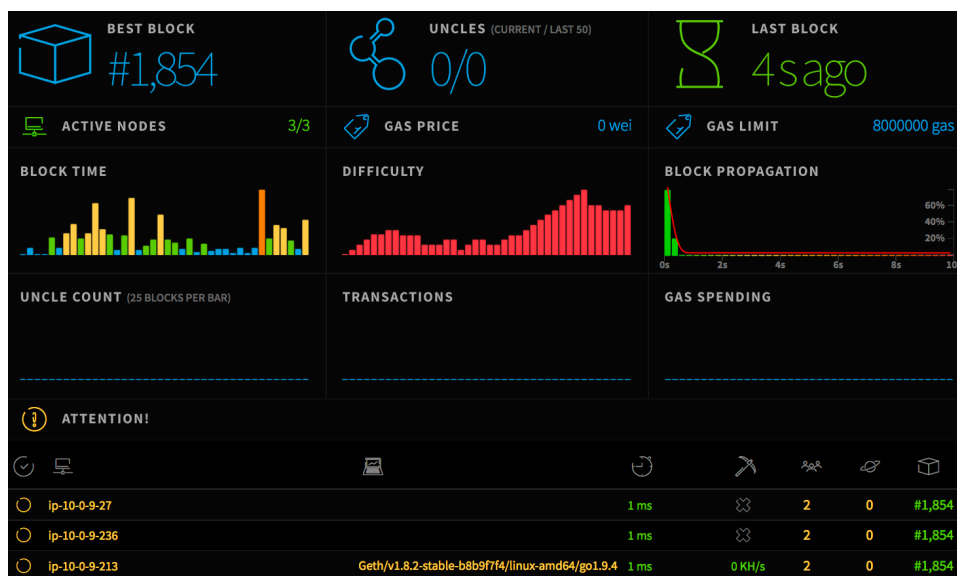
- **Source port**, enter **9001**.
- Leave **Destination** blank.
- Select **Dynamic**.
- Choose **Add**.

Choose **Open**.

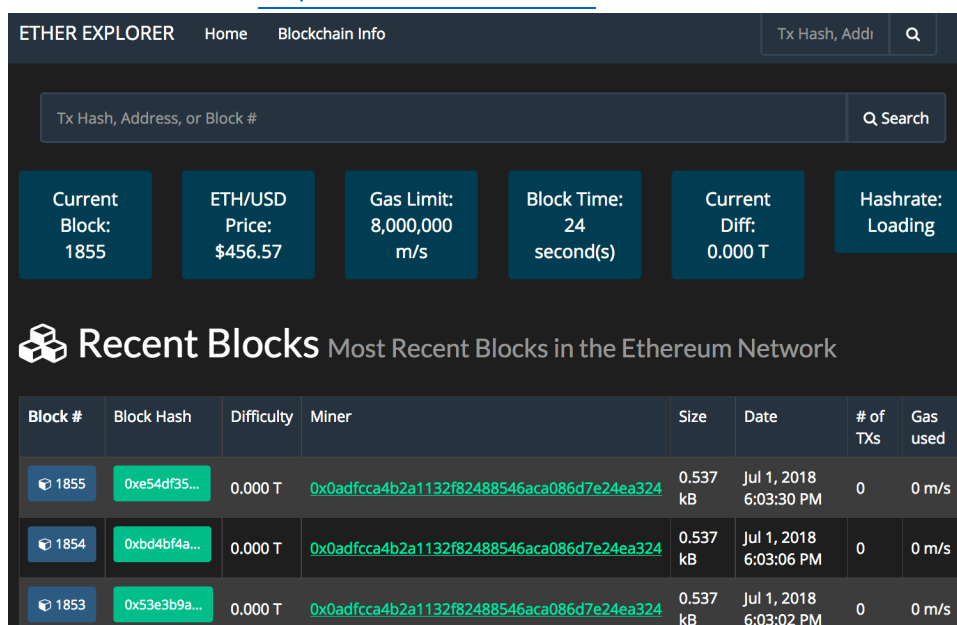
To check the results of the Ethereum Network

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"}
```