National College of Ireland

Configuration Manual

MSc Research Project
Cloud Computing

Animesh Tewari
Student ID:22247963

School of Computing
National College of Ireland

Supervisor: Prof. Sean Heeney

National College of Ireland
Project Submission Sheet
School of Computing

National College of Ireland

Student Name:	Animesh Tewari
Student ID:	22247963
Programme:	Cloud Computing
Year:	2024
Module:	MSc Research Project
Supervisor:	Prof. Sean Heeney
Submission Due Date:	12/08/2024
Project Title:	Configuration Manual
Word Count:	888
Page Count:	4

I hereby certify that the information contained in this (my submission) is information pertaining to research I conducted for this project. All information other than my own contribution will be fully referenced and listed in the relevant bibliography section at the rear of the project.

ALL internet material must be referenced in the bibliography section. Students are required to use the Referencing Standard specified in the report template. To use other author's written or electronic work is illegal (plagiarism) and may result in disciplinary action.

Signature:	
Date:	12th August 2024

PLEASE READ THE FOLLOWING INSTRUCTIONS AND CHECKLIST:

Attach a completed copy of this sheet to each project (including multiple copies).	☐
Attach a Moodle submission receipt of the online project submission, to each project (including multiple copies).	☐
You must ensure that you retain a HARD COPY of the project, both for your own reference and in case a project is lost or mislaid. It is not sufficient to keep a copy on computer.	☐

Assignments that are submitted to the Programme Coordinator office must be placed into the assignment box located outside the office.

Office Use Only	
Signature:	
Date:	
Penalty Applied (if applicable):	

Configuration Manual

Animesh Tewari
22247963

1 Interacting with the smart contract

To run the project, you would require to setup node.js[1] and Truffle[2], which is a testing framework, to interact with the smart contracts locally. To install these kindly follow below steps:

1.1 Install Node.js and npm

Node.js is a JavaScript runtime that is required for the project, and npm called the Node Package Manager to handle all the package related to node , similar to how brew is the package manager in Mac.

1. **Download and Install Node.js:**
   - Use this link `https://nodejs.org/` and download the LTS (Long Term Support) version of Node.js suitable as per the operating system.
   - Follow the installation instructions provided on the website.
2. **Verify the Installation:**
   - Open a terminal.
   - Run the following commands to verify that Node.js and npm are installed correctly:

```
-$ node -v
-$ npm -v
```

   - If it is successfully installed these commands would show versions of these.

1.2 Install Truffle

As the Binance Smart Chain is compatible with Ethereum Virtual Machines, the deployed smart contracts are written in Solidity and deployed on the Binance Smart Chain Testnet. To interact with the smart contract, Truffle would be required. To install Truffle kindly follow the below steps:

1. **Install Truffle:**

[1]https://nodejs.org/en/learn/getting-started/introduction-to-nodejs
[2]https://archive.trufflesuite.com/docs/truffle/how-to/debug-test/test-your-contracts/

- Run the following command in terminal to install Truffle on the system:

```
-$ npm install truffle
```

2. **Verify the Installation:**

   - After installation, check that Truffle is installed correctly by running:

```
-$ truffle version
```

   - This command will display the version of Truffle installed.

For more details, on how to install kindly refer [3]

1.3 Interact with deployed smart contracts

Now, to interact with the deployed smart contracts, you can use the `truffle console` command with the network flag set to `bsctestnet`, as the network settings have already been added in the `truffle-config.js`. When at the root level of the project folder, run the command below to enter the Truffle console:

```
-$ truffle console --network bsctestnet
```

The console will open like this 1

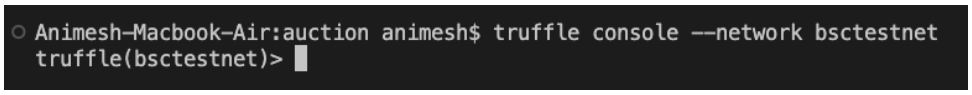

Figure 1: Truffle console

It's important to make sure that the .env file, in the root directory, of the folder is provided with the private key from Metamask for the Binance Smart Chain Test Network as can be seen in 2

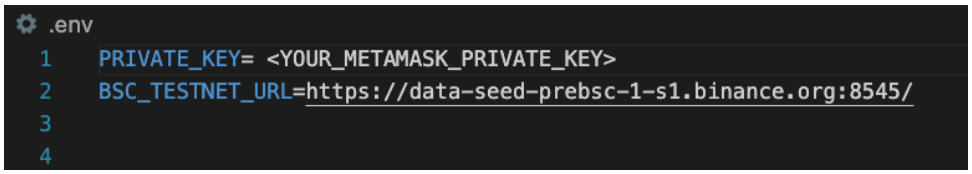

Figure 2: Env file with private key for interacting with Binance Smart Chain Testnet

2 Deploying the Poller and SMPC Containers on AWS Fargate

This section provides the essential steps for deploying the Poller and SMPC Docker containers on AWS Fargate using Elastic Container Service (ECS).

[3] https://archive.trufflesuite.com/docs/truffle/how-to/install/

2.1 Step 1: Authenticate Docker to AWS ECR

Authenticate Docker to your AWS ECR repository:

```
aws ecr get-login-password --region <region> | docker login --username AWS --password
```

2.2 Step 2: Tag and Push Docker Images to ECR

Tag and push your Docker images to ECR:

Tag the Poller Image:

```
docker tag poller:latest <account-id>.dkr.ecr.<region>.amazonaws.com/poller-service:l
```

Push the Poller Image:

```
docker push <account-id>.dkr.ecr.<region>.amazonaws.com/poller-service:latest
```

Tag the SMPC Image:

```
docker tag smpc:latest <account-id>.dkr.ecr.<region>.amazonaws.com/smpc-service:lates
```

Push the SMPC Image:

```
docker push <account-id>.dkr.ecr.<region>.amazonaws.com/smpc-service:latest
```

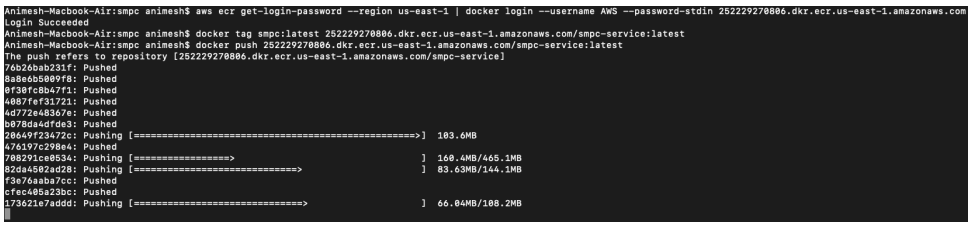

Figure 3: Sending locally created docker image to ECR

Kindly refer to this 3 on how the terminal would look.

2.3 Step 3: Create ECS Cluster

Create an ECS cluster using the AWS Management Console:

1. Go to `Elastic Container Service (ECS)`.
2. Click `Create Cluster`.
3. Choose `Networking only` (Fargate) and click `Create`.

2.4 Step 4: Define Task Definitions

Define task definitions for Poller and SMPC:

1. Go to `Task Definitions` in the ECS Console.
2. Click `Create new Task Definition` and select `Fargate`.
3. Configure the Poller container (e.g., `port 8000`) and SMPC containers (e.g., `ports 8001, 8002`).
4. Save and create the task definitions.

2.5 Step 5: Deploy Services on Fargate

Deploy the services:

1. Navigate to your ECS Cluster.
2. Click `Services` and `Create Service`.
3. Choose the Poller and SMPC task definitions and click `Create`. For the poller, 1 task run should be enough and for SMPC, minimum should 2.

These steps outline the deployment process for the Poller and SMPC containers on AWS Fargate.