



Google Summer of Code Project Proposal



Project:- Nodecloud

Partik Singh

Mentor:- Mohit Bhat

Project Introduction

NodeCloud is an API allowing developers to interact with multiple cloud providers using a unified interface. This abstraction layer helps to hide the implementation details of different cloud providers, making it easier for developers to use cloud services without having to deal with the specifics of each provider. The use of cloud platforms is highly dependent on the use case, and different cloud providers offer different tradeoffs, such as performance, scalability, availability, cost, and more.

NodeCloud simplifies the process of developing cloud-based applications by providing a common interface for multiple cloud providers, making it easier for developers to build scalable, reliable, and cost-effective applications.

A developer often needs to combine the services of different cloud providers while developing a project; in that case, it becomes tough for a developer to switch between the services of various cloud providers.

Currently, Nodecloud comes with the plugins for AWS, Google Cloud Platform, Microsoft Azure, Digital Ocean, and Aliyun and also with an automatic class generator that uses **code parsers** (which reads the SDK files and returns the class declaration node of the Abstract Syntax Tree), **data extraction functions** (which extracts all the data required for class generation), **transformers** (most important part of class generation which generates the final class). The automatic class generator leverages the Typescript Compiler API to automate the process of class generation for nodecloud for various cloud plugins.

The aim of this project is to extend the plugin functionality of Nodecloud to include a couple of emerging cloud providers, Linode and Oracle Cloud.

Oracle Cloud:-

Oracle Cloud is a cloud computing service that provides businesses and organizations with access to a wide range of cloud-based resources, including servers, storage, network, applications, and services. Oracle Cloud offers a range of services, including infrastructure as a service (IaaS), platform as a service (PaaS), and software as a service (SaaS). This means that developers can choose the level of service that best suits

their needs, from simple cloud storage and hosting to fully managed applications and platforms.

Linode:-

Linode is a cloud hosting provider that provides Linux-powered virtual machines to support a wide range of applications. Given its roots, many experts do not see it as beginner-friendly. Linode server locations are optimized for serving a global audience in terms of reducing latency and improving UX. With constant innovation, they aim to become the top choice for developers.

Under this project, I aim to implement the core functionalities of the Oracle cloud and Linode cloud providers and also extend the functionality of the existing cloud plugins to support newly introduced services.

Project Goals

The end goal of this project is to implement the Linode and Oracle cloud plugins into Nodecloud for abstracting cloud services provided by both cloud providers. While creating the plugins for both the cloud providers the code generation module must be taken into consideration as it is already present for the existing modules. This would help maintain uniformity for all the plugins and will also remove the need to manually write code for each plugin.

The Oracle Cloud Provider comes up with a Typescript SDK which can be used to implement the Oracle Cloud Plugin. This SDK has defined each service via a class based client and the type definition for every client is given hence we can use this SDK for this project as with this SDK we can leverage the automatic code generation module. Also, the SDK lacks a detailed documentation, but after going through the Documentation of the Cloud API and the code of the SDK and the type definition files, I came down to the fact that this SDK can be used in this project. The Oracle cloud SDK/API comes up with an abundance of services from different domains like Artificial Intelligence, Blockchain, Computation, Storage, Database, etc. After going through the documentation of the services and also the services implemented in the already existing cloud provider in Nodecloud, I have selected the services given in the following table to be implemented during the course of this project.

Product	Service	Oracle Cloud Offering
Compute	Compute Instances	Compute Instance Agent
	Kubernetes	Container Engine for Kubernetes
	Container	Container Instances
Storage	Object Storage	Object Storage Service
	File Storage	File Storage Service
	Archive Storage	Archive Storage Service
Databases	NO SQL Database	NO SQL Database Service
	SQL Database	MYSQL Database Service
	Database	Database Service
Networking	Load Balancer	Load Balancing Service
	DNS	DNS Service
Artificial Intelligence	Artificial Intelligence Speech	OCI Speech Service
	Artificial Intelligence Vision	OCI Vision Service
	Artificial Intelligence Language	OCI Language Service
Other	Notifications	OCI Notifications Service

Table 1

The Linode Cloud Provider also comes up with a Typescript SDK. The SDK for Linode comes up with a detailed documentation with comments provided for every function, which helps understand and implement the SDK; hence it is really easy to understand the working of the SDK, and also the detailed documentation provided for the Linode Cloud API also makes it easy to understand the SDK, also the type definition files are also defined and hence this SDK can be used to implement all the services that we wish to include in the Linode plugin. The table for the services that will be implemented in this project is as follows:-

Product	Service	Linode Cloud Offering
Compute	Compute Instances	Linode Instance
	Kubernetes	Linode Kubernetes Engine
Storage	Object Storage	Object Storage Service
	Block Storage	Block Storage
Database	NO SQL Database	Database Mongo DB
	SQL Database	Database MY SQL
		Database Postgresql
Networking	Load Balancers	Node Balancers
	DNS	DNS Manager

Table 2

Below is the list of all the goals that I wish to achieve during the course of Gsoc'23:-

1. **Oracle Cloud Provider Plugin and Linode Cloud Provide Plugin:-**

Oracle Cloud Provider and Linode Cloud provider plugin is to be implemented in the Nodecloud package with all the services mentioned above in Table 1 and Table 2, respectively. The sub-goals for implementing Cloud Providers are:-

a.) Building an Automatic Code Generation module for Oracle Cloud Plugin and Linode Cloud Plugin. Below are the sub-tasks that need to be done in order to build the module.

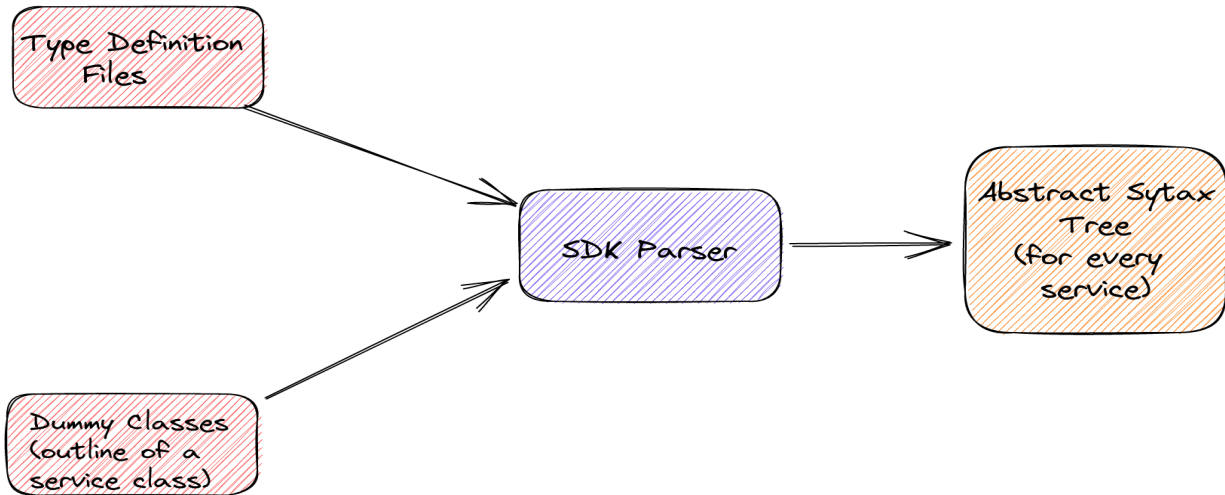
1. **Code parser:**-A code parser needs to be created for the Oracle Cloud SDK and Linode Cloud SDK which can be used to read the SDK files and convert them into AST(Abstract Syntax Tree) and return the class declaration node of the AST for each service class that needs to be added.
2. **Data Extraction Functions:**-Some Data Extraction functions are needed to be implemented in the generator of Oracle and Linode Cloud SDKs which can be used to extract all the required data from the AST node.
3. **Transformer :-** Separate transformer for Oracle and Linode Cloud SDKs needs to be implemented which could add functions, parameters and comments for the corresponding functions in the auto-generated classes.

b.) Implement Linode and Oracle Cloud as a plugin in Nodecloud, implement the plugin with Lerna and also publish the plugin as a separate NPM package.

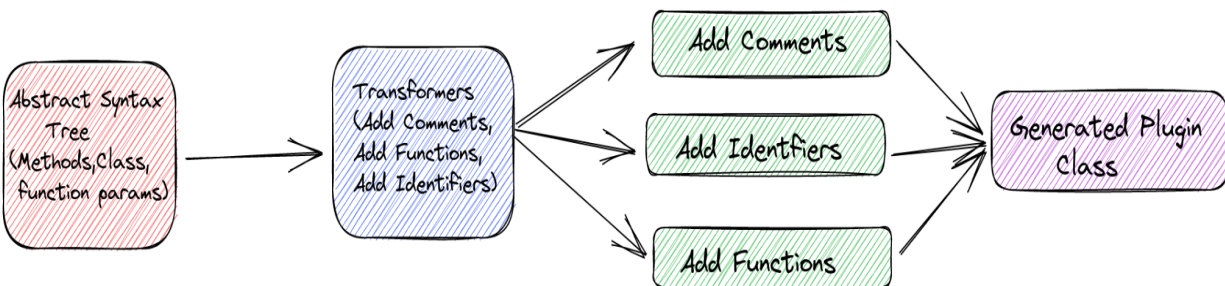
Implementation

Oracle Cloud Plugin

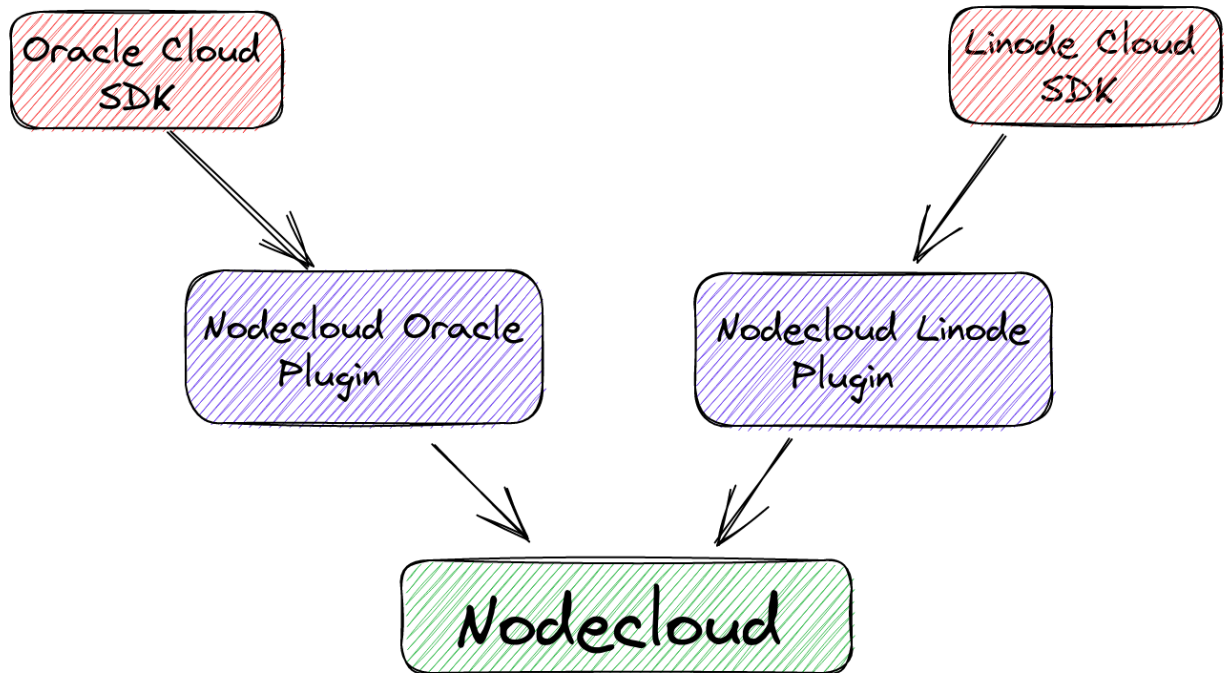
1. The Oracle Cloud plugin will be implemented using the Typescript SDK that is provided by the Oracle Platform itself (the SDK can be found [here](#)). And similarly the Linode Platform also provides an official Typescript SDK for the API v4(the SDK can be found [here](#)).
2. The Oracle Cloud SDK contains a separate module or SDK generated for each service and every service has its own type definition file for the service client hence we can use those files in this project to leverage the powers of the automatic code generation module provided by the Typescript compiler API.
3. Below is the list of all the Services along with the functions I wish to implement in the Oracle Plugin and in the Linode Plugin respectively, these functions are mainly selected based on my understanding of the requirements of a user, functions can be added or subtracted based on the review of the mentor.
4. The Oracle and Linode Cloud plugin implementation have to be implemented in two steps in order to maintain the structure of the code and leverage the benefits of automatic class generation. Below are the two steps:-
 - a. Automatic generation of Nodecloud classes for both plugins
 - b. Using the generated classes into a mono repo and publishing the plugin as a separate NPM package as well.
4. For leveraging the automatic class generation a separate module for the Oracle and Linode Cloud SDKs should be implemented. For that purpose:-
 - a. Adding new services and their corresponding functions to be implemented in the node-cloud.yml file.
 - b. Implementing the SDK parser which will generate relevant AST(Abstract Syntax Tree) for every service to be implemented.



- c. Implementing relevant data extraction functions which will extract all the required information to generate a new class. These have to be implemented in the generator module of the Oracle and Linode Cloud SDKs.
- d. Implementation of a transformer for the SDKs which will contain the addFunctions, addIdentifier, and addComments functions to finally generate the required class for each service.
- e. Write all the relevant tests needed to check the functionality of the whole class generator module.
- f. Manually check the generator module by applying it to the SDKs and checking the implementation for each service



5. After this, I have to use the generated classes for the implementation of the plugin using a mono repo structure and publish it as a separate NPM package along with the appropriate examples, samples, and vast documentation.



Compute Services	
Container Instance Service (Container Instances)	getContainer
	updateContainer
	listContainers
	createContainerInstance
	deleteContainerInstance
	getContainerInstance
	restartContainerInstance
	startContainerInstance
	stopContainerInstance
	updateContainerInstance

```

1
2 import common = require("oci-common");
3 import * as requests from "./request";
4 import * as responses from "./response";
5 declare class LoadBalancerClient{
6     constructor(params: common.AuthParams, clientConfiguration?: common.ClientConfiguration)
7     createLoadBalancer(
8         createLoadBalancerRequest: requests.CreateLoadBalancerRequest
9     ): Promise<responses.CreateLoadBalancerResponse>
10
11     deleteLoadBalancer(
12         deleteLoadBalancerRequest: requests.DeleteLoadBalancerRequest
13     ): Promise<responses.DeleteLoadBalancerResponse>
14     getLoadBalancer(
15         getLoadBalancerRequest: requests.GetLoadBalancerRequest
16     ): Promise<responses.GetLoadBalancerResponse>
17     updateLoadBalancer(
18         updateLoadBalancerRequest: requests.UpdateLoadBalancerRequest
19     ): Promise<responses.UpdateLoadBalancerResponse>
20     listLoadBalancers(
21         listLoadBalancersRequest: requests.ListLoadBalancersRequest
22     ): Promise<responses.ListLoadBalancersResponse>
23 }

```

Example Type Definition File For Oracle Cloud SDK

Kubernetes Cluster Service (Container Engine for Kubernetes)	createCluster
	createKubeConfig
	deleteCluster
	getCluster
	updateCluster
	listClusters
	createNodePool
	deleteNode
	deleteNodePool
	getNodePool

```

1  class ClassName {
2      /**
3       *
4       * @param {module} do Oracle SDK
5       * @param {object} options SDK options
6       */
7      constructor(ocisdsk, params, clientConfiguration) {
8          this._oci = ocisdsk;
9          this._sdkclassName = this._oci.SDKClassName(params, clientConfiguration);
10     }
11
12     function() {
13         return new Promise((resolve, reject) => {
14             this._sdkclassName
15                 .SDKFunctionName()
16                 .then(data => resolve(data))
17                 .catch(err => reject(err));
18         });
19     }
20 }
21
22 module.exports = ClassName;
23

```

Dummy Class for Oracle Cloud SDK

Storage	
Object Storage Service (Object Storage)	createBucket
	deleteBucket
	getBucket
	listBuckets
	reencryptBucket

Node Cloud File changes for the Oracle Cloud SDK Conatiner Engine

```


1  import * as fs from 'fs';
2  import * as path from 'path';
3  import { createSourceFile, ScriptTarget, SyntaxKind } from 'typescript';
4
5  export function getAST(sdkFileName) {
6      return new Promise(async (resolve, reject) => {
7          try {
8              const file = path.join(
9                  __dirname,
10                 `../../../../../node_modules/oci-${sdkFileName.toLowerCase()}/lib/`
11             );
12             const ast = createSourceFile(
13                 file,
14                 fs.readFileSync(file).toString(),
15                 ScriptTarget.Latest,
16                 true
17             );
18
19             let cloned = null;
20
21             await ast.forEachChild(child => {
22                 if (SyntaxKind[child.kind] === 'ClassDeclaration') {
23                     cloned = Object.assign({}, child);
24                 }
25             });
26
27             if (!cloned) {
28                 reject(new Error('Class not found!'));
29             } else {
30                 resolve(cloned);
31             }
32         } catch (error) {
33             if (error.code === 'ENOENT') {
34                 reject(new Error('File not found!'));
35             } else {
36                 reject(error);
37             }
38         }
39     });
40 }
41

```

Node Cloud Parser for Oracle Cloud SDK

Database	
MYSQL Database Service (MYSQL)	createDBSystem
	deleteDBSystem

	getDBSystem
	restartDBSystem
	startDBSystem
	stopDBSystem
	updateDBSystem



```

1  Oracle:
2    create: client.d.ts createContainerInstance
3    delete: client.d.ts deleteContainerInstance
4    start: client.d.ts startContainerInstance
5    stop: client.d.ts stopContainerInstance
6    describe: client.d.ts getContainer
7    list: client.d.ts listContainers

```

Node Cloud File changes for the Oracle Cloud SDK Conatiner Engine

NO SQL Database Service (NO SQL)	createTable
	deleteTable
	getTable
	listTables
	updateTable
	deleteRow
	getRow
	updateRow



```
1 Oracle:
2   create: client.d.ts createCluster
3   delete: client.d.ts deleteCluster
4   listClusters: client.d.ts listClusters
5   describeCluster: client.d.ts getCluster
6   createNodePool: client.d.ts createNodePool
7   deleteNodePool: client.d.ts deleteNodePool
8   deleteNode: client.d.ts deleteNode
9   describeNodePool: client.d.ts getNodePool
```

Node Cloud File changes for the Oracle Cloud SDK Kubernetes Engine

Load Balancing Service (Load Balancing)	createLoadBalancer
	deleteLoadBalancer
	getLoadBalancer
	updateLoadBalancer
	listLoadBalancers

```

1  constructor(params: common.AuthParams, clientConfiguration?: common.ClientConfiguration) {
2      const requestSigner = params.authenticationDetailsProvider
3        ? new common.DefaultRequestSigner(params.authenticationDetailsProvider)
4        : null;
5      if (clientConfiguration) {
6          this._clientConfiguration = clientConfiguration;
7          this._circuitBreaker = clientConfiguration.circuitBreaker
8            ? clientConfiguration.circuitBreaker!.circuit
9            : null;
10         this._httpOptions = clientConfiguration.httpOptions
11           ? clientConfiguration.httpOptions
12           : undefined;
13     }
14     // if circuit breaker is not created, check if circuit breaker system is enabled to use default circuit breaker
15     const specCircuitBreakerEnabled = true;
16     if (
17         !this._circuitBreaker &&
18         common.utils.isCircuitBreakerSystemEnabled(clientConfiguration!) &&
19         (specCircuitBreakerEnabled || common.CircuitBreaker.DefaultCircuitBreakerOverriden)
20     ) {
21         this._circuitBreaker = new common.CircuitBreaker().circuit;
22     }
23     this._httpClient =
24       params.httpClient ||
25       new common.FetchHttpClient(requestSigner, this._circuitBreaker, this._httpOptions);
26
27     if (
28       params.authenticationDetailsProvider &&
29       common.isRegionProvider(params.authenticationDetailsProvider)
30     ) {
31         const provider: common.RegionProvider = params.authenticationDetailsProvider;
32         if (provider.getRegion()) {
33             this.region = provider.getRegion();
34         }
35     }
36 }

```

Client Constructor Function for Each Service of Oracle Cloud SDK

Artificial Intelligence	
Artificial Intelligence Speech Service (AI Speech)	createTranscriptionJob
	getTranscriptionJob
	listTranscriptionJobs
	updateTranscriptionJob
	cancelTranscriptionTask
	getTranscriptionTask

Artificial Intelligence Vision Service (AI Vision)	createDocumentJob
	getDocumentJob
	cancelDocumentJob
	getImageJob
	createImageJob
	cancelImageJob
	createProject
	deleteProject
	getProject
	updateProject

Linode Cloud Plugin

Compute	
Kubernetes Cluster Service (Kubernetes Service)	getKubernetesCluster
	createKubernetesCluster
	updateKubernetesCluster
	deleteKubernetesCluster
	getKubeConfig
	resetKubeConfig
	getNodePool
	createNodePool
	updateNodePool
	deleteNodePool


```

1  class ClassName {
2      /**
3       *
4       * @param {module} do Linode SDK
5       * @param {object} options SDK options
6       */
7      constructor(linodeSdk, linodeToken) {
8          this._linode = linodeSdk;
9          this._linodeToken = linodeToken;
10     }
11
12     function() {
13         this._linode.setToken(this._linodeToken);
14         return new Promise((resolve, reject) => {
15             this._linode
16                 .SDKFunctionName()
17                 .then(data => resolve(data))
18                 .catch(err => reject(err));
19         });
20     }
21 }
22
23 module.exports = ClassName;
24

```

Example Dummy Class for Linode

Linode Instance Service (Linode Instances)	getLinodes
	createLinode
	updateLinode
	deleteLinode
	cloneLinode
	getLinodeConfig

	createLinodeConfig
	deleteLinodeConfig
	updateLinodeConfig

Storage	
Object Storage Service (Object Storage Service)	getBucket
	getBuckets
	createBucket
	deleteBucket
	getBucketAccess
	updateBucketAccess

Database	
Linode Database Service (Database engines:-MongoDB,Postgresql,MYSql) (Database Service)	getDatabases
	getDatabaseTypes
	getDatabaseType
	getDatabaseEngines
	createDatabase
	updateDatabase
	deleteDatabase
	getEngineDatabases
	getEngineDatabase

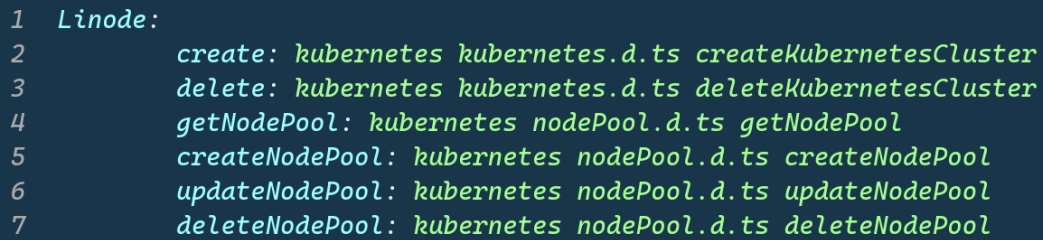
Network	
Linode Nodebalancer Service (Nodebalancer)	getNodeBalancers
	getNodeBalancer
	updateNodeBalancer
	createNodeBalancer
	deleteNodeBalancer
	getNodeBalancerConfig
	createNodeBalancerConfig
	updateNodeBalancerConfig

```

1  Linode:
2      create: nodebalancers nodebalancers.d.ts createNodeBalancer
3      delete: nodebalancers nodebalancers.d.ts deleteNodeBalancer
4      list: nodebalancers nodebalancers.d.ts getNodeBalancers
5      update: nodebalancers nodebalancers.d.ts updateNodeBalancer
6      describe: nodebalancers nodebalancers.d.ts getNodeBalancer

```

Example Node Cloud File Changes for Linode Parser for Nodebalancer



```
1 Linode:
2   create: kubernetes kubernetes.d.ts createKubernetesCluster
3   delete: kubernetes kubernetes.d.ts deleteKubernetesCluster
4   getNodePool: kubernetes nodePool.d.ts getNodePool
5   createNodePool: kubernetes nodePool.d.ts createNodePool
6   updateNodePool: kubernetes nodePool.d.ts updateNodePool
7   deleteNodePool: kubernetes nodePool.d.ts deleteNodePool
```

Example Node Cloud File Changes for Linode Parser for Kubernetes

Note:- I have also started the implementation of the project on my own in my forked repo of Nodecloud in different branches and created the pull requests for both cloud plugins. The links of both the pull requests can be found below:-

- Oracle:-<https://github.com/partik03/nodecloud/pull/2>
- Linode:-<https://github.com/partik03/nodecloud/pull/1>

Project Timeline

<u>Timeline</u>	<u>Milestone</u>
May 4 - May 28	<ol style="list-style-type: none"> 1. Fine tuning with the mentor 2. Finalizing the project requirements 3. Finalizing the project approach by discussing with the mentor
May 29 - June 4	<ol style="list-style-type: none"> 1. Initialise the generator module for Oracle and Linode 2. Add dummy classes for both the cloud providers 3. Finalize all the services and their required functions to be implemented
June 5-June 11	<ol style="list-style-type: none"> 1. Implement the SDK parser for generating AST for Oracle Cloud provider
June 12 - June 18	<ol style="list-style-type: none"> 1. Implement the SDK parser for generating AST for Linode Cloud provider
June 19 - June 25	<ol style="list-style-type: none"> 1. Implement the Data Extractor functions for Oracle cloud provider
June 26- July 2	<ol style="list-style-type: none"> 1. Implement the Data Extractor functions for Linode cloud provider
July 3 - July 9	<ol style="list-style-type: none"> 1. Implement Transformer for Oracle cloud provider
July 10 - July 16 (July 10 -July 14 {Midterm Evaluation})	<ol style="list-style-type: none"> 1. Bug fixes + Backlog completion for <u>Midterm Evaluation</u> 2. Implement Transformer for Linode Cloud Provider
July 17 - July 23	<ol style="list-style-type: none"> 1. Write unit and integration tests for the automatic class generation module for

	<p>Linode and Oracle cloud provide</p> <ol style="list-style-type: none"> 2. Manually check the generator module by applying it to the SDKs and checking the implementation for each service
July 24 - July 30	<ol style="list-style-type: none"> 1. Make use of the generated classes for Oracle to implement a plugin using a mono repo structure with Lerna 2. Make use of the generated classes for Linode to implement a plugin using a mono repo structure with Lerna
July 31- August 6	<ol style="list-style-type: none"> 1. Testing the plugins via the Nodecloud package 2. Bug fixes 3. Completing backlogs(if any)
July 31 - August 6	<ol style="list-style-type: none"> 1. Bug fixed 2. Completing backlogs(if any) 3. Testing the plugins via the Nodecloud package
August 7 -August 13	<ol style="list-style-type: none"> 1. Testing the plugins via the Nodecloud package 2. Adding examples
August 14 - August 20	<ol style="list-style-type: none"> 1. Bug fixes(if any) 2. Writing documentation 3. Getting mentor's feedback and suggestions
August 21 -August 28	<ol style="list-style-type: none"> 1. Finalising the project based on mentor's suggestions 2. Project Presentation

Score Contributions

I have been involved in the Score Community since December last year. I have contributed to a number of projects namely Codelabz, OpenMF, Scan8, and Cloudlibz and I have thoroughly gone through the code of Nodecloud, and I am familiar with the codebase. I have been a consistent contributor in the community making pull requests, filing issues, and interacting on the Gitter channel of Codelabz.

Below are some of my contributions

Issues

<u>Issue</u>	<u>Project</u>
1. #508	Codelabz
2. #511	Codelabz
3. #531	Codelabz
4. #563	Codelabz
5. #659	Codelabz
6. #705	Codelabz
7. #707	Codelabz
8. #308	OpenMF
9. #112	Com-Dictionary
10. #113	Com-Dictionary
11. #141	ImageLab

Pull requests

<u>Pull Request</u>	<u>Project</u>
1. #508	Codelabz
2. #512	Codelabz
3. #532	Codelabz
4. #562	Codelabz
5. #664	Codelabz
6. #706	Codelabz
7. #708	Codelabz
8. #309	OpenMF
9. #91	Scan8
10. #46	Cloudlibz
11. #110	Com-Dictionary
12. #111	Com-Dictionary
13. #114	Com-Dictionary
14. #142	ImageLab

Personal Information

Name: Partik Singh Bumrah

Phone: +91 8146046906

Email: partikbumrah13508@gmail.com,

College Email: partik.sbumrah.cse21@itbhu.ac.in,

Github: [partik03](#),

Time Zone: Kolkata, India (GMT+5:30)

I am Partik Singh and I am a second-year Computer Science and Engineering undergraduate at Indian Institute of Technology, Varanasi. I am a Cloud Native and Web3 enthusiast, having participated in cohorts such as Tezos Buildercamp and boot camps such as Alchemy University, where I gained fundamental knowledge in Blockchain and Web3.

I am also a part of our college's programming club The Club Of Programmers for a year now where I have had the opportunity to work on a number of existing as well as fresh incoming projects. I have an extensive background in a variety of technologies, such as C, C++, Python, JavaScript, Typescript, React.js, Next.js, Vue.js, Node.js, Solidity, and GoLang.

I have also contributed to open-source organizations such as Layer5 and Appwrite. I am also a silver medalist in the 11th Inter IIT Tech Meet, where our team worked on creating a Snap for Metamask. This Snap had improvements for Metamask and included features such as a Private Key Recovery System powered by Biometric Authentication, Notifications for Incoming Transactions and Notifications for Bids on NFTs.

I am also a proud community member of Layer5 and Appwrite. I am participating in GSOC'23 to learn a lot of new stuff and also for exposure to Open Source software development.

I consider myself to be perfect for working on the Nodecloud project as I am well versed with Nodejs and Typescript and have also been exploring cloud-native for some time. I think I have the ability to understand all the logic that goes into building a complete Javascript-based package. I have also gone through and understood the already implemented logic/code in the NodeCloud package and also understood the workings of the SDKs provided by the Cloud Providers and I believe that I also have the programming ability needed to turn this gathered knowledge into code. I aim to learn a lot while working on this project.

Reference

Project reference

1. <https://github.com/oracle/oci-typescript-sdk/tree/master/lib>
2. <https://docs.oracle.com/en-us/iaas/Content/services.htm>
3. <https://docs.oracle.com/en-us/iaas/api/#/>
4. <https://www.linode.com/docs/api/>
5. <https://github.com/linode/manager/tree/develop/packages/api-v4>
6. <https://developers.linode.com/libraries-tools/>

Questions

1. Are you a SCoRe contributor/ Have you contributed to SCoRe before?

-Yes, I have been contributing to SCoRe since December 2022. I have contributed to Codelabz, OpenMF, Scan8, and Cloudlibz.

2. How can we reach you (e.g.: email) if we have questions about your application?

-One can reach me via the following channels;_

Primary Email: partikbumrah13508@gmail.com

Secondary Email: partiksingh2287@gmail.com

Contact Number: +91 8146046906

Github: @partik03

Time Zone: Kolkata, India (GMT+5:30)

Also I am open to communication with the mentor on:-

- Discord (Partik Singh #9615)
- Google Meet
- Zoom
- MS Teams

3. What is your GitHub username(s):
- Github User: @partik03

Project Specific Questions

4. Which SCoRe GSoC project are you applying for (please submit separate applications for each project):
- I am going to apply for, #13: Nodecloud-Oracle and Linode Cloud Provider, from the GSoC'23 ideas list.
5. What do you plan to accomplish over this summer for this project?
- a. What project do you want to work on?
- I want to work on the Nodecloud project as I am very interested in Cloud and have also been exploring Cloud Native. I also have the skillset required for this project.
 - How you will approach that project portion (with your milestones))
 - I will approach all the milestones for the project in the following two steps:-
 - Ask the mentor for his view about the approach and implementation of the milestone and his idea of the final result after the milestone is achieved.
 - Ask the mentor for suggestions and feedback after the achievement of every milestone.
6. If you have your own project to propose, please describe it here:
- N/A
7. Project-related details.
- Yes, I have set up the project correctly. Honestly, I haven't had any difficulties while setting up the project, the documentation helped me a lot while setting up the project.
8. List down any plans you have during this summer(over the time period of GSoC, such as classes, job, vacation plans, thesis, etc.)
- I don't have any commitment other than GSOC during this summer as I am having vacations from my college. I will dedicate my time only working on this project to deliver the goals I listed above for this project.
9. Education:
- A. What year are you in school?

- I am a second-year Computer Science undergraduate student at the Indian Institute of Technology (BHU), Varanasi, India.
- B. What programming courses have you taken?
- Some of my programming courses are as follows:-
 1. Semester 1
 - a. Computer Programming (C programming)
 2. Semester 2
 - a. Data Structures and Algorithms
 - b. Information and Technology Workshop 1
(Introduction to Linux and Python)
 3. Semester 3
 - a. Computer System Organization
 - b. Information and Technology Workshop 2
(Introduction to Web Development using Python)
 4. Semester 4 (Ongoing)
 - a. Artificial Intelligence
 - b. Algorithms
 - c. Operating Systems
- C. What is your major?
- I am pursuing a Bachelor's(4year) in Computer Science and Engineering.
- D. Have you done group projects(programming or otherwise)?
- Yes, I have done a lot of group projects with my batchmates, in The Club of Programmers(COPS) IIT BHU, in The Business Club IIT BHU(BIZ), and with some of my college seniors as well.
 - Some of the group projects we did were:-
 - Recreating the COPS SDG site
 - Working on creating a study portal for IIT BHU in COPS
 - Building SNAPX snap with the Inter IIT Tech Team(won silver)
 - Building the Business Club's site and also a mock trading platform (both in progress)
 - Organising Hack It Out(a hackathon) in Technex(Annual Tech Fest of IIT BHU)

- Worked in open source organizations like Appwrite, Layer5, Meshery (also a community member of Layer5 and Appwrite)
- Done a Django project in my 3rd semester with my batchmate
- Participated in Hackout Hackathon'22 with four batchmates
- Developed the IIT BHU Seed Fund Website with one of my Batchmate

E. What was your primary contribution to/role in the group?

- In the groups in which I worked with my college seniors, I excessively worked on some critical portion of the project, I generally asked my seniors to assign me an important part of the project as I love solving complex problems. I also helped my fellow team members in any ability I could and also suggested ideas and recommendations to my seniors.
- The projects I did with my batchmates, in those projects, I always tried that everyone should contribute equally; hence we tried to divide the project into equally complex subproblems, and every team member handled a single sub-problem.
- In the open-source communities I worked, I generally raised issues, raised pull requests, did the work that the mentors assigned to me, offered my review on some pull requests, and made some feature requests to improve the project.

10. Do you have work experience in programming? Tell us about it.

- I have worked with code in my college for my course. We have built some projects in Django, Shell, and Python for some courses. Also, I have completed a bunch of programming assignments that I got from my college. Also, I am working on two projects for my fourth semester 1) First one is an Artificial Intelligence project, and 2) The second one is based on blockchain technology.
- Also, I have contributed to my college programming club COPS and The Business Club. I have contributed to many projects under COPS and the Business Club. Also, I have worked in a team for giving the

orientation of COPS to our college freshers in which we built a Telegram Bot for freshers providing them with all the information they require to survive in IIT BHU.

- I started Open Source with Hacktoberfest'22 and have contributed consistently to different organizations since then. I have worked with Python, Nextjs, and some GoLang during this work.
- I have also been a part of Web3 cohorts like Tezos Indian Buildercamp and Alchemy University, where I learned while coding the essentials of Web3 and interacted with my fellows during the cohorts.
- I have also been part of the Inter IIT Tech Team for the Consensys-Metamask Snap problem statement, for which we built a snap for recovering a lost private key for the user by verifying the user's biometric information. We also won the silver medal for this Snap.

11. Do you have previous open-source experience? Briefly describe what you have done.

- I am pretty new to open source. I learned about open source from my college seniors when I worked with them on some projects. I kickstarted my journey with open source with the celebrated Hacktoberfest, where I worked on individual projects and with organizations like Appwrite. I merged my four pull requests with two pull requests in the Appwrite Organisation.

Below are some of the pull requests I made during October:-

- <https://github.com/open-runtimes/examples/pull/29>
- <https://github.com/JasonFritsche/news-from-the-wormhole/pull/52>
- <https://github.com/COPS-IITBHU/sdg-site/pull/118>
- <https://github.com/open-runtimes/examples/pull/65>
- I also started contributing to Scorelabs organization during December last year and above are the Pull Requests and Issues I made till now in my Scorelabs journey.
- I also started exploring more Open Source during December and started contributing to the Meshery project of Layer5 organization, I am also a community member of Layer5 and also was the community member of Layer5Labs. During my journey, I made Pull Requests, filed Issues, and offered my review on some pull requests

in both Meshery and their Meshmap project. Some of my contributions are given below:-

- i. <https://github.com/meshery/meshery/pull/6651>
- ii. <https://github.com/meshery/meshery/pull/6657>
- iii. <https://github.com/meshery/meshery/pull/6678>
- iv. <https://github.com/meshery/meshery/pull/6800>
- v. <https://github.com/meshery/meshery/pull/6957>
- vi. <https://github.com/meshery/meshery/pull/6725>
- vii. <https://github.com/meshery/meshery/pull/6804>
- viii. <https://github.com/meshery/meshery/pull/6843>

12. Tell one interesting fact about yourself.

- Apart from programming and studies, I consider myself a gym freak. I go to the gym regularly and try my best to maintain my diet (although it's tough while staying in a hostel). I consistently hit the gym every day and try to do a bodyweight workout on Sundays while staying in the hostel.
- Gym plays an important role in my life. It makes me stay fit, be motivated, be disciplined, be consistent, be competitive, and be hardworking as well.
- Apart from that, I love sports. I played various sports during my school days, including cricket, football, badminton, and kickboxing. Although I enjoy playing all sports, I love cricket a lot, so I always try to find an opportunity to enjoy some time playing cricket in college.