# **SVM**

### **Project Category : AI Model Training/ Execution**

### **Project Name : Data Foundation Cloud (DCloud) Environment**

### **Team Members:**

### **Shlok Bansal : 2021201046**

### **Vishal Pandey : 2021201070**

### **Mayank Mukundam : 2021201057**

### **Project Overview:**

1. Docker nodes with defined dataset and constrained resource access (GPU/CPU, storage and memory)
2. Python and NVIDIA based AI/ ML setup for TensorFlow and PyTorch

# **Software Requirements**

## **1. Introduction**

The system provides an efficient solution for building a containerised ecosystem. Users can easily configure these ecosystems according to their requirements but within constraints of limited CPUs, GPUs, memory. User will be provided with a friendly interface to manage all sorts of configurations. This streamlined process reduces the time and effort required to manage the ecosystem.

## **2. System Features**

* Configurations management eg no of gpus, ram, automatic software installation support
* Docker container management according to requirements
* Container monitoring
* Real Time Status Updates
* handle concurrent requests

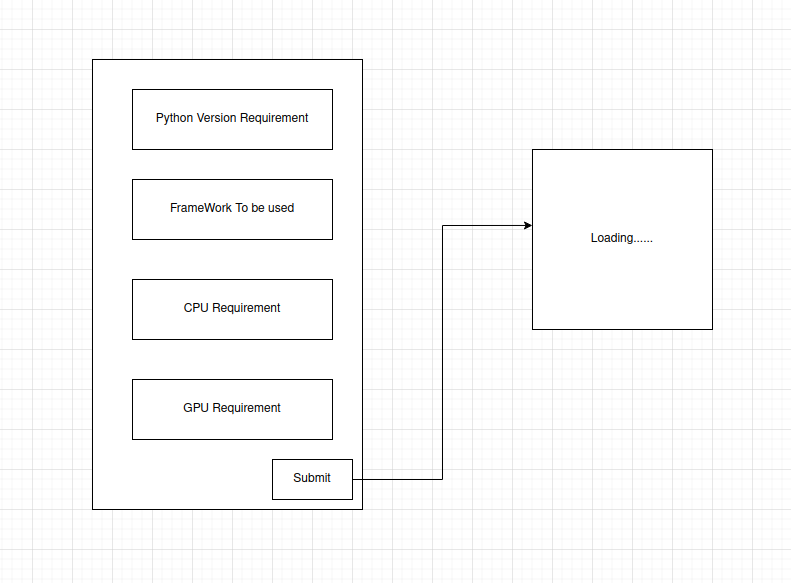
## **3. Non Functional Requirements**

* Performance: The system should be able to handle high and perform efficiently, with quick response time.
* High Scalability
* High Availability
* Interoperability: The system should be able to integrate with DFS platform
* User-friendly: The system should be easy to use, with a user-friendly interface and intuitive functionality.
* Reliability: The system should be reliable, with minimal downtime and a high degree of stability.
* Compatibility: The system should be compatible with various data formats, allowing users to upload their own ecosystem config files.
* Maintainability: The system should be easy to maintain and upgrade, with clear documentation and a robust architecture.

## **4. Project Deliverable**

The project will be delivered in phases:

* Prototype phase: Building a simple and efficient prototype to demonstrate the workflow for container ecosystem deployment
* Concurrency support: This phase will mostly focus on making the abover prototype more rustic and handle traffic efficiently
* UI Integration phase This phase deals with the Integration of whole backend with a UI Interface so that, it can be easily used by the end user.



As shown in the above diagram, the UI will appear in the as a form wherein user will get to choose the python version user will work on, framework user wants to get support of in their Ecosystem, Computing Requirements for the Ecosystem. All the frameworks & python resource requirements will be a dropdown which shows the list of resources that are available to create an ecosystem. On clicking the Submit button a progress bar will be shown which also shows success/failure status at the end of loading.

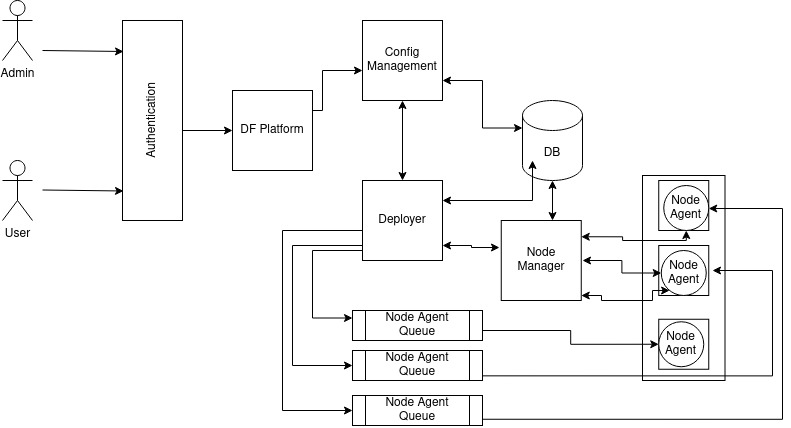
* Logging and Fault Tolerance

## 

## 

## 

## **Architecture Overview**



## **Configuration Contract**

