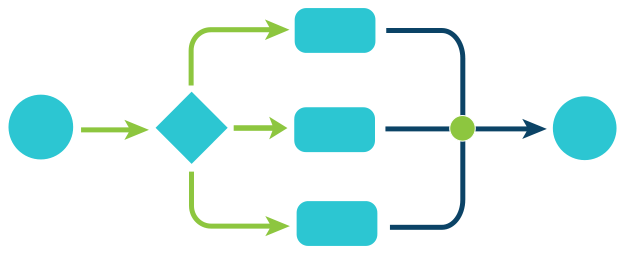
## 



BioFlows Framework

0.1

**─**

Mohamed Fawzy

Cairo , Egypt

# 

# 

# Overview

BioFlows is a distributed pipeline framework for expressing , designing and running scalable reproducible and distributed computational bioinformatics workflows in cloud containers.

BioFlows Framework consists of software tools and cloud microservices that communicate together to achieve a highly distributed , highly coordinated and fault tolerant environment to run parallel bioinformatics pipelines onto cloud containers and cloud servers. BioFlows also has BioFlows Description Language (BDL) which is an imperative and declarative standard for describing and expressing computational bioinformatics tools and pipelines, BDL is flexible , easy to use and a human readable language that enables researchers to design reproducible and scalable computational pipelines. The language is based entirely on Yet Another Markup Language (YAML).

# Design Goals

## Portability

BioFlows Framework enables researchers to run massively parallel bioinformatics pipelines on multiple Cloud Platforms , Operating Systems and different Cloud Containers.

Extensibility

BioFlows Framework is highly Extensible Framework which allows researchers to extend its functionalities by providing alternative implementation to its different layered components, in addition , BioFlows has very rich set of APIs and documentations which allows the community to develop software tools that take advantage of BioFlows Framework.

Fault Tolerance

BioFlows Framework ensures a seamless and highly consistent execution environment for bioinformatics tools.

Tools execution in BioFlows is managed by two highly coordinated microservices, The Cluster Manager and JobManager , Cluster Manager ensures that JobManager is always running on the cluster Node while JobManager ensures the integrity of the running job, if the job failed, the Job Manager restarts the job locally and if it failed for n times which defaults to 3 times, the cluster manager will distribute this particular job onto another cluster Node which satisfies the current job computing specifications or wait until available suitable resource becomes free to use.

Reproducibility

BioFlows Framework enables researchers to write self-contained computational pipelines that are sufficient to run on different Environments with zero configuration because it contains all execution and configuration parameters without any modification, thus, The framework allows different institutions to rapidly reproduce results.

# BioFlows Architecture Components

There are two major tiers that makeup BioFlows Framework, this section describes these two major tiers and outlines their structure, layers, and tools, in addition, we describe the methods and protocols used in establishing bidirectional communication among different microservices.

## (Cluster/Server) Tier

This tier consists of all tools and different daemonized microservices which makes up the runtime and the distributed execution environment of BioFlows Framework and ensures consistent and scalable cloud environment where pipelines run in a fully distributed, resilient and fault tolerant environment, this tier consists of many layers which communicate among each other to provide consistency, durability, fault tolerance and speed for running different computational pipelines. The following figure (1-1) explains the structure of this component, its layers and then we move deeper into intricate details of how all these layers and tools communicate to achieve fully distributed and resilient execution cloud environment.

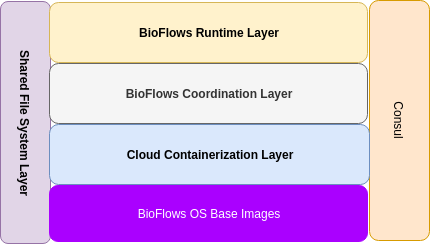


Figure : (1-1)

As Shown in figure (1-1), The cluster/server tier is composed of multiple layers stacked together to form a solid distributed, resilient, fault tolerant and scalable cloud runtime environment where different pipelines run to achieve speed, consistency, and reliability.

The image shows the level of abstraction for each layer starting from BioFlows OS Base Images way up to Bioflows Runtime Layer, the following section describes each layer in detail.

BioFlows OS Images

BioFlows Framework allows computational pipelines to run onto different cloud containers, each step or tool in a pipeline, when scheduled by Cluster Scheduler, will run on its own container, these containers run on a homogenous unified Operating system image where they share all basic computational tools and dependencies, moreover , the job executor on each container makes sure that each tool has its own dependencies resolved and installed in the container before the tool run , otherwise, the job executor reports back any errors or failure to the job manager which spawned the container, we will discuss the coordination logic later in this paper, in section [....]

Cloud Containerization Layer

This layer consists of the underlying Unix/Linux Containerization technology, which is used by the Job manager to spawn new containers on a host node, BioFlows framework supports Docker and Singularity Containers, in addition, BioFlows was built with flexibility and extensibility in mind and thus can accommodate other containerization technology.

BioFlows Coordination Layer

BioFlows Framework is a scalable distributed Ecosystem for managing massively parallel computational pipelines over a cluster of host machines, Successively, BioFlows has a distributed queue management system, network protocols like AMQP, RPC (Remote procedure calls) used for direct microservice-to-microservice communications, intelligent resource allocation and leader election algorithms implemented in the cluster manager microservice, in addition, BioFlows has a shared File System where intermediate and final results from different running tools are stored, this file system is shared among containers instances and mounted automatically for each container during the spawning process by the job manager.

Furthermore, BioFlows has consul which is a service mesh solution providing a full-featured control plane for service discovery, configuration and segmentation functionality in fully automated and distributed environment, moreover, consul is being used as orchestrator to build cluster topologies on the fly and to act as a distributed in-memory key value store, where transient configurations and data are being stored and shared among different container instances during execution, furthermore , BioFlows Framework uses consul as a distributed DNS Store where different microservices can discover other services for direct communication in a fully automated way.

BioFlows Runtime Layer

This layer consists of major BioFlows microservices and tools that coordinate between each others and make up BioFlows distributed system, these tools include Cluster Manager , Job Managers, and job executors. <More Info>

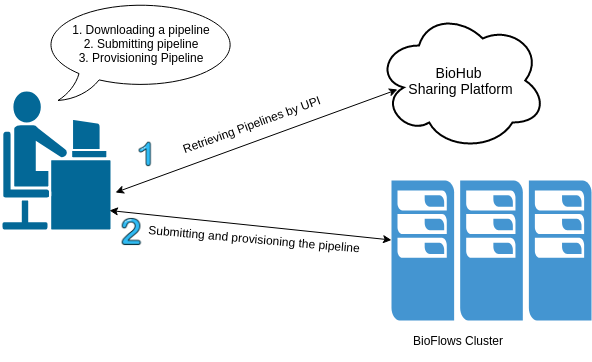
2. Client Tier - Client Toolkit

BioFlows also has client toolkit which consists of a set of different command line tools which can be used to author new computational pipelines , validate a pipeline, communicate with BioFlows Biohub to submit or retrieve pipelines and/or tools, communicate with BioFlows Cluster and many other functionalities, for more information, you can refer to BioFlows Client Toolkit Github repository [GIVE A NUMBER FROM REFERENCES] .

3. BioHub Component - BioFlows Sharing Platform

Reproducibility and zero configuration are of utmost importance in any scientific research [1], It has been a well known problem to reproduce the same results of computational bioinformatics analyses of other researchers.

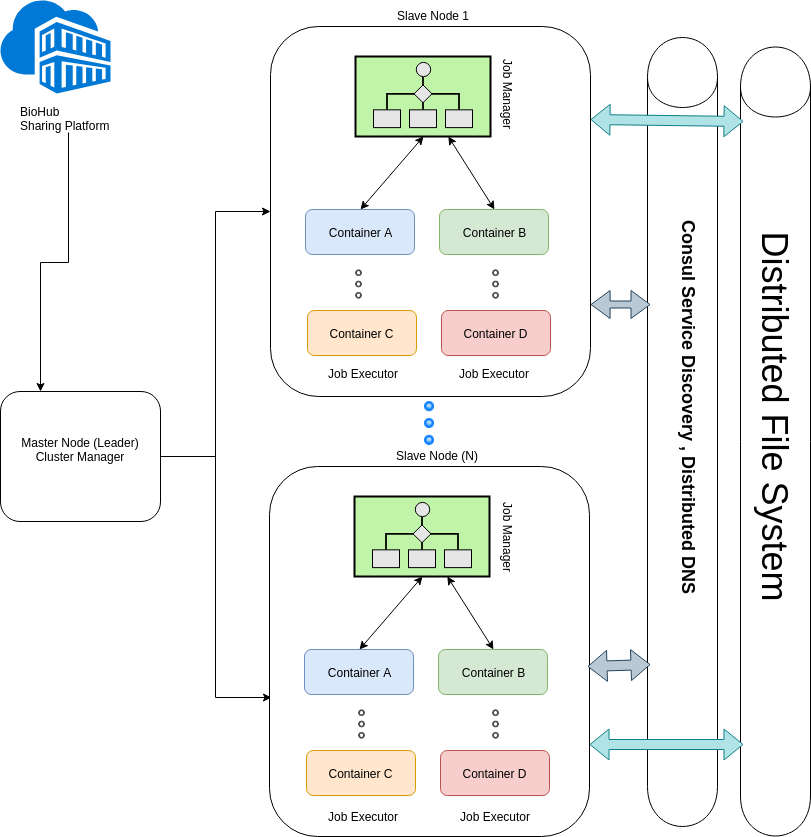
BioFlows Framework praises the concept of “Sharing is Caring” and for this, we have built BioHub which is a centralized cloud sharing platform of computational bioinformatics pipelines that require zero configuration to run , BioFlows Client Toolkit eases the task of searching, downloading, submitting and provisioning these pipelines onto BioFlows Distributed Cluster with zero configuration and no time, BioHub component introduces some interesting security challenges that we will discuss a bit later in this paper , in the security section of the framework and we will also discuss the ways BioFlows has for mitigating and counter measuring those challenges, moreover, BioFlows Framework is bridging the gap between deep technical skills , code development and infrastructure operations, since , by using flows framework researchers and biologists with no computer skills can still administer flows framework to obtain , submit and provision bioinformatics pipelines onto cluster of machines so easily with no prior knowledge for the underlying technologies used thus increasing their focus on the most important aspects of their research and to find answers to their research questions as in Figure (1-2).



*Figure : (1-2)*

In addition, researchers can author new computational pipelines from known tools registry or defining their own tools then piping them into a workflow, setting explicit parameters, defining execution constraints and other parameters for each tool and/or in the workflow itself, afterward, researchers should validate the workflow on their research data sets after making sure that the workflow is validated and it works correctly they can share it with other researchers who might be seeking answers to similar research questions.

Below, Figure (1-3) clearly shows BioFlows cloud Cluster Architecture and its major components in a clearer way



*Figure (1-3): BioFlows Framework Cloud Architecture*