아이디어 정리

1. 로그인

* IAM 접근 코드를 사용자로부터 입력을 받고 그 후 아이디와 패스워드를 입력하도록 한다. 그 후 해당 아이디와 패스워드를 통해서만 SW 에 접근이 가능하도록 한다.
* 이때 얻어진 IAM 접근 코드를 통해서 실제로 aws 접근이 가능한 형태이다. 여기서 아이디와 패스워드는 해당 SW에 대한 접근이 가능하도록 한다.
* 오류방지를 위해 해당 IAM 코드를 정확히 입력했는지 확인해야 한다.

1. 화면 전체 UI

* 전체 스크린의 사이즈를 가지고 온다. 그리고 해당 사이즈를 기준으로 상,하,좌,우 에 대해서 밀착시킨다.

.

사이즈고정

사이즈고정

중앙

사이즈고정

* 상,하,좌,우 각 패널들은 바같부분을 기준으로 중앙쪽으로 확장되는 개념으로 윈도우 사이즈를 계산하여 배치한다.

1. 마스터 노드 생성

에디터를 통해서 IAM 접근 코드로 AWS 에 접속이 가능해지면, 그러한 상태부터 마스터 노드의 생성이 가능해진다. 마스터 노드의 삭제는 모든 에이전트 노드들이 삭제되고 작업이 없을 때만이 삭제할 수 있다. 그리고 에디터를 종료해도 마스터 노드는 삭제가 되지 않는다. 에디터를 새로 생성해서 마스터 노드의 생성 유무를 확인 할 수 있어야 한다.

1. 에디터에서의 shell script 에디팅

코드 에디팅 화면이 나타난다. 배경1을 위에

나타나는 코드 에디팅 화면

클릭했을때

Samtools

BWA

배경1

1. 초기 에디터가 생성하는 AWS 자원

* 마스터 노드, registry server, DB??
* 에디터가 생성한 AWS 자원의 경우 3번항목과 같은 생명 주기를 가진다.

1. AWS 노드 생성 지연 및 에러 처리
2. 노드 에디터

* 먼저 데스크탑용 에디터는 에디터 설치 시 저장된 파일(xml 또는 기타 파일, A 파일) 등을 가지고와서 NodeEditor 의 툴 선택창에 데이터를 집어 넣어준다. 이때, 넣어주는 데이터 즉, 저장된 파일(xml 또는 기타 파일, A 파일) 의 버전을 확인한다. 버전을 확인하는 방법은 사용자가 에디터의 로그인을 하였을 경우 원격 서버에 저장된 파일(xml 또는 기타파일, B 파일)의 버전과 비교하고 만약 A 파일 보다 B 파일의 버전이 최신일 경우는 해당 B 파일을 다운로드 하고 새롭게 리로딩 하는 과정을 거친다.
* 에디터를 통해서 작성된 파일은(C 파일) 사용자가 런 또는 드라이 런을 했을 경우, 서버로 전송이 된다. 전송된 파일은 서버에서 파싱? 이 되어 파일에서 잡들을 분리하고 기타 정보들을 분리하여 DB 에 저장한다.

1. 잡을 생성하는 주기(생명주기) 좀더 생각해보기…

* 사용자가 job 을 실행시키면 job 을 생성해주는 모듈(Job Creator) 는 생성된다.
* 사용자가 job 을 실행시키면 먼저 job 을 생성해주는 모듈이 살아 있는지 확인한다. 만약 살아있으면 전송된 파일(C 파일)을 파싱하고 정보를 DB에 저장한다.
* DB 에 저장된 job 과 그에 대한 정보들이 존재하는 한 job 을 생성해주는 모듈(Job Creator)을 살아있다. DB에 저장된 job 과 그에 대한 정보들이 job을 수행하여 점차 없어지고(보여주는 부분은 별도의 저장공간 또는 테이블에 넣어둠) 아무런 job 이 없다면 job 을 생성해주는 모듈(Job Creator)은 삭제된다.

1. 최종 데이터 보내기 및 에러 보내기
2. 여러 파일들을 보낼 때 처리 (프로토타입 보고 재설계)

사용자가 관리해야 할 파일은 대략적으로 4가지 유형이 있다.

1. 분석할 파일(input - 여러 개의 파일)
2. 분석에 필요한 파일(input – 공통적으로 사용할 파일)
3. 결과 파일 (output – 여러 개의 파일)
4. 기타 결과 파일(output – 하나의 파일)

파일들객체

BWA

1. 파일 입출력이 필요함.

* 프로젝트 파일, 서버 전송 파일 포멧 정리 필요