Confluence Documentation

Application Architecture

November 16, 2021

Nikki Tebaldi (ntebaldi@umass.edu)

TABLE OF CONTENTS

TABLE OF FIGURES

CLASS DIAGRAM

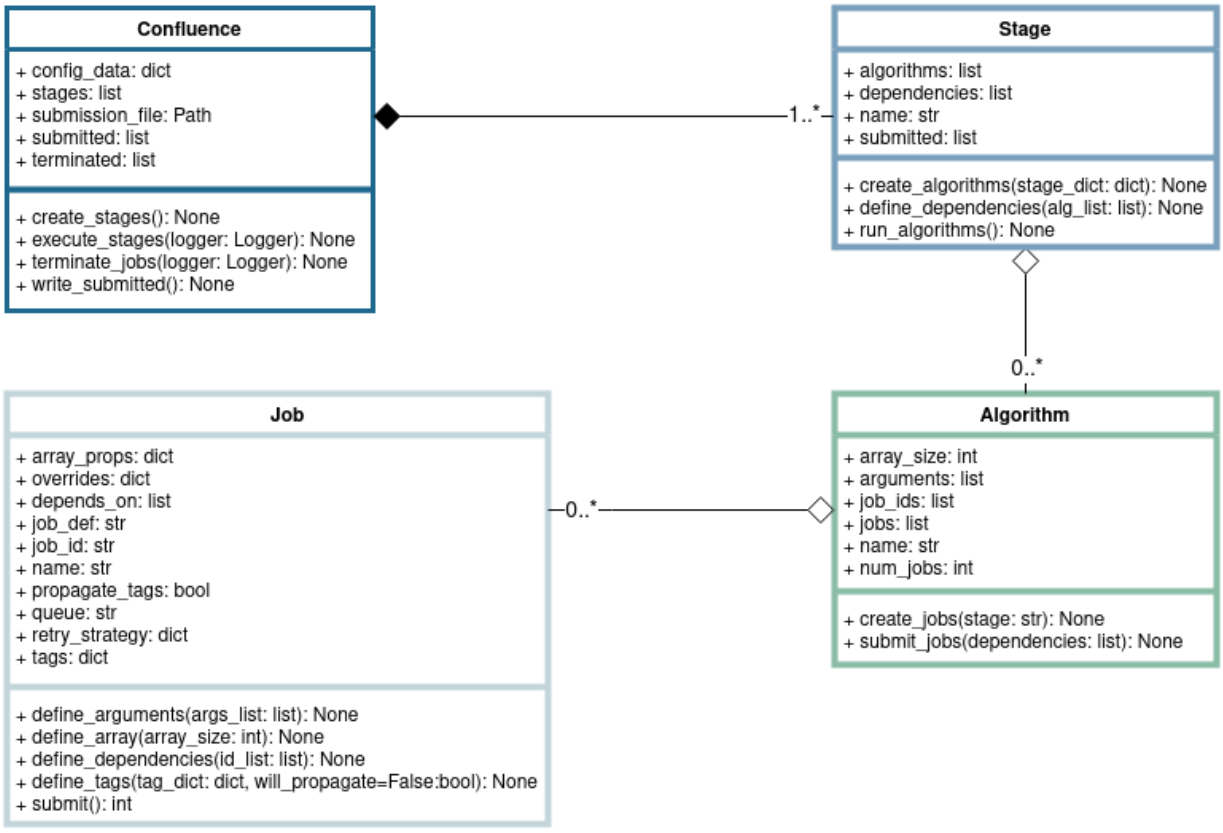

Figure 1 confluence-aws Class Diagram

Code Repository: https://github.com/SWOT-Confluence/confluence-aws

Confluence as code is a Python program that uses Boto3, the AWS SDK. It determines and submits jobs required for each stage of Confluence to AWS Batch.

COMPONENT DESCRIPTION

The **Confluence** class is composed of Stage objects. This class orchestrates the components required to submit AWS Batch jobs. Confluence objects create and execute Stage objects which store Algorithm objects that invoke Job objects to submit jobs to AWS Batch. This class also terminates jobs if a job submission fails. Confluence logs the jobs that are submitted in a CSV file and logs status and errors to the console.

The **Stage** class aggregates Algorithm objects and represents a stage in the Confluence workflow. The Stage class creates the required Algorithm objects that are needed to submit jobs to AWS Batch for the

entire stage. The Stage class also tracks job dependencies to maintain the order of execution between stages.

The **Algorithm** class aggregates Job objects and represents an algorithm that requires execution in the Confluence workflow. An algorithm is often submitted to AWS Batch as a set of jobs. An Algorithm class creates Job objects to handle the submission operation. The Algorithm class also keeps track of the numeric job identifiers for all AWS Batch jobs that have been submitted.

The **Job** class represents a job in AWS Batch. A job has many properties, and the Job class allows for customization of many of these properties to meet the needs of whatever function is building a job submission. The Job class uses the Boto3 AWS SDK to submit jobs and returns the numeric job identifier upon submission.

CONFLUENCE JOB CONFIGURATION

Sample confluence.yaml file: https://github.com/SWOT-Confluence/confluence-aws/blob/main/confluence.yaml

The configuration and submission of jobs is handled by a YAML file titled: "confluence.yaml". This file allows the user to set a 'log_file' and 'submission_file' destination. It also allows the users to tell Confluence what modules to run in each stage and define the number of jobs, array size, and arguments associated with each module.

The organization of the "stages" in the yaml file starts with a stage name and then provides the name of any modules that should be run. Note that some stages like the 'input' stage only have one module and it is also named 'input' while other stages have multiple modules like the 'flpe' stage with 'geobam', 'hivdi', 'metroman', 'momma', 'sad', and 'sic4dvar' modules. It is important to retain the names of the stages and modules as they are in the YAML file.

Each module contains several properties that allow the user to define job submission properties:

"**num_jobs**" or number of jobs indicates to the Algorithm class how many Job objects should be created for each module in a stage.

"**array_size**" is used by the Job class to submit an Array Job (https://docs.aws.amazon.com/batch/latest/userguide/array_jobs.html).

"**arguments**" indicates the command line arguments that should be submitted with each job and passed to the containers that are created by AWS Batch to execute the job. The arguments are submitted as a list by the Job class.

LIST OF ACRONYMS

AWS: Amazon Web Services

EC2: Amazon Elastic Compute Cloud

ECR: Amazon Elastic Container Registry

EFS: Amazon Elastic File System

FLPE: Flow Law Parameter Estimation

IAM: AWS Identity and Access Management (IAM)

JPL: Jet Propulsion Laboratory

PO.DAAC: Physical Oceanography Distributed Active Archive Center

S3: Amazon Simple Storage Service

SDS: Science Data System

SoS: SWORD of Science

SWORD: SWOT River Database

SWOT: Surface Water and Ocean Topography

VPC: Amazon Virtual Private Cloud