```
# parse
sns_result = aws.SNS(record['Sns'])

# download
s3_result = aws.S3Download(sns_result.file_info, sns_result.jmodel_runid)

cmdline_args = [os.path.join(os.getcwd(), 'lib', 'phyml'), ]
cmdline_args.extend(['-i', s3_result.local_file])
cmdline_args.extend(sns_result.payload.split())

trace_file = os.path.join(
        s3_result.tmp_folder,
        "{}_phyml_trace_{}.txt".format(
               sns_result.jmodel_runid,
               sns_result.jmodel_modelname)
)

logging.info("PhyML starting...")
phyml_start = timer()

with open(trace_file, "w") as file:
        result = subprocess.run(cmdline_args,
                                stdout=file,
                                stderr=subprocess.STDOUT)

phyml_duration = (timer() - phyml_start)
logging.warn("PhyML took {} secs".format(phyml_duration))

# bail out if phyml error'd
if result.returncode != 0:

        logging.critical("PhyML.ReturnCode={}".format(result.returncode))

        # log trace file
        with open(trace_file, 'r', encoding='UTF-8') as file_stream:
                file_contents = file_stream.read()
                logging.error(file_contents)

        raise subprocess.SubprocessError("Error calling PhyML")

# phyml succeeded, go ahead
result_files = [x for x in
                os.listdir(s3_result.tmp_folder)
                if x != sns_result.jmodel_runid]

logging.debug("Phyml produced = {}".format(result_files))

# upload
s3_up = aws.S3Upload(s3_result.tmp_folder, result_files, sns_result)

logging.info("Uploaded = {} to {}://{}/".format(
        list(s3_up.files.values()),
        sns_result.file_info['bucket'],
        s3_up.jmodel_runid)
)

# mark as done
aws.DynamoDB(sns_result.jmodel_runid, sns_result.jmodel_modelname)
```