Summary:

We have created the CMT create and destroy sequence. While the created compute instance/virtual machine is up and running, we've displayed a temporary placeholder "Job Executing" phase in the Job Status UI as we were awaiting the details as now provided in this document.

Now we are finally able to replace that "Job Executing" placeholder with actually running the model!

Steps to be executed as soon as CMT returns "Created":

1. **STEP 1**: Obtain IP address of created virtual machine from CMT stackId

   SECTION REMOVED AS OF 4/1/2019
   We should have this from Aaron by end-of-day today.
   I will send it separately.

2. **STEP 2**: Go through the Data Ingest phase
   a. Step 2.1: Submit the DI JSON as currently created from user model config

      EXAMPLE HTTP POST of the DI JSON to the IP Address of the virtual machine:

```
curl --request POST --url http://13.59.16.58:5226/api/v1.0/di/q --header
'content-type: application/json' --data "{
   "q": {
    "queries": [
     {
       "name": "HRRR_PRS",
       "start_date": "2019-03-11T00:00:00-00:00",
       "end_date"  : "2019-03-11T01:00:00-00:00",
       "interval": 1
     }
    ]
   }
 }"
```

      Note: the bold IP Address is just an example IP address – this is obtained in Step 1.

   b. Step 2.2: Receive and parse the response message, making a list of the to-be-downloaded files.

      EXAMPLE HTTP Response from POST call above, showing the returned JSON documenting the initialization model files to be downloaded:

{"responseCode":0,"responseMsg":"Download initiated","statuses":[{"url":"http://www.ftp.ncep.noaa.gov/data/nccf/com/hrrr/prod/hrrr.20190311/conus/hrrr.t00z.wrfprsf00.grib2","state":"Downloading"},{"url":"http://www.ftp.ncep.noaa.gov/data/nccf/com/hrrr/prod/hrrr.20190311/conus/hrrr.t01z.wrfprsf00.grib2","state":"Downloading"}]}

Parse the "statuses" array and make a list of any/all links indicated with the string "Downloading" for state. I am not sure at this point if there are any other states we need to worry about. For now, forget about error handling. Just scan the "statuses" array for any URLs that are designated as "Downloading". Keep this list for the following execution steps.

c. Step 2.3: Update Job Status UI with "Data Ingest" section and subsections for file download progress

INSTEAD of the placeholder "Job Execution" list status item, add "Data Ingest" phase with sub-list items for each and every file being downloaded.
APRIL Update: I talked to Samm and we probably won't have download percentage. Just in progress or finished.

d. Step 2.4: Make periodic calls to check file download status, update UI, and scan for completion of DI, at which point we progress to JMT

THE FORMAT for making file download status checks to the DI is as follows:

- curl --request POST --url http://13.59.16.58:5226/api/v1.0/di/status --header 'content-type: application/json' --data '{ **urls**: ['"http://www.ftp.ncep.noaa.gov/data/nccf/com/hrrr/prod/hrrr.20190311/conus/hrrr.t00z.wrfprsf00.grib2"'] }'

POPULATE "**urls**" JSON array with the list of files being downloaded

THE RESPONSE MESSAGE format for these status checks is the same as above, exactly as returned when you first submit the DI JSON.

The way we will determine whether or not the DI is done is therefore by checking the "state" variable. Right now the logic on each status call will be: "if ALL urls states != "Downloaded", THEN done with DI and advance to JMT"

3. **STEP 3**: Go through the JMT Execution Phase
   a. Step 3.1: Submit the JMT JSON as currently created from user model config

EXAMPLE HTTP POST of the JMT JSON to the IP Address of the virtual machine:

curl --request POST --url http://13.59.16.58:7827/api/v1.0/jmt/job/create --header 'content-type: application/json' --data "{Replace with JMT JSON}"

Note: the bold IP Address is just an example IP address – this is obtained in Step 1.

b. Step 3.2: Receive and parse the response message, literally reading the "responseMsg' field as follows:

EXAMPLE HTTP Response from POST call above, showing the returned JSON from JMT:

- {"responseCode":0,"responseMsg":"WpsRunning","detailedMsg":[],"userId":1,"jobId":1,"jobResponseCode":3}

c. Step 3.3: Update Job Status UI with "JMT" section and subsections for JMT response message progress, and check for completion, while making periodic calls to check JMT states

THE FORMAT for making JMT status check is as follows:

- curl --request GET --url http://13.59.16.58:7827/api/v1.0/jmt/job/status/1

POPULATE the "1" after "status/" with the "jobId" returned by JMT.

WHEN "responseMsg" changes to something new:

1. Make a new line on the Job Status UI showing the new status

2. IF responseMsg = "Completed", then perform model run complete procedures, i.e., update UI and being to delete CMT.