SRE Take Home Challenge

The Assignee needs to complete one of the following:

# Assignment 1 (1 week)

The assignment is divided into 3 parts.

The assignee is expected to complete each part according to the definition of done of that part

And if he/she is not able to complete anything, then describe the challenge he/she faced in completing it.

1. Build a custom Jenkins Docker image from the official base

Install role based auth strategy plugin if not there https://plugins.jenkins.io/role-strategy

Enable the following roles:

1. Admin
2. Deployer
3. Developer
4. Prod-deployer

Give relevant access to the above roles

Save image back on Docker hub

**Definition of done**:

The resultant image in Docker hub

Rationale applied in giving access to each of the roles.

1. Create a pipeline to automate and publish the above image after testing the relevant parts.

**Definition of Done:**

* 1. Pipeline in chosen tool
  2. Test cases.

1. Run the above image on a hosted Docker service.

**Definition of done:**

1. URL of the running service
2. Rationale behind choosing the hosted service if any.

# Assignment 2 (1 Week)

The assignment is divided into 2 parts.

The assignee is expected to complete each part according to the definition of done of that part

And if he/she is not able to complete anything, then describe the challenge he/she faced in completing it.

1. Create a script (whatever scripting language is preferred) to scrape logs for given events. The events will be provided in a text/json file. The events format would look like :
   * [timestamp] [log\_level] [thread] This is a data event showing [ID]
   * [timestamp] [log\_level] [thread] This is an error event for client [$CLIENT ID]

Where [] mark the fields in the log, the value for which is to be collected and sent to ELK stack , a data event needs to go to an index named “data” and an error event needs to go to an index “error”

[$] marks protected fields for which the value is to be masked before sending to the ELK stack.

You are free to change the format to suit your script, though make sure to clearly state the assumptions made and things supported as part of your scraper.

**Definition of Done:**

* + The script with a proper readme file.
  + If a hosted ELK could be used to show the data then good, else only printing a valid payload for ELK in a text file or a json dump for ELK would suffice.

1. Come up with a valid trigger for the script.

**Definition of Done:**

The detailed description behind choosing the trigger which should include the following:

1. Implementation feasibility
2. Performance considerations
3. Maximum Data lag between an event occurrence and reporting