Team IRIS

Systems Team Recruitment 2024

26/02/2024

Instructions

1. More than one task can be attempted. For the bonus task to be considered, at least one other task must be completed.
2. Log all the steps you take while completing the tasks and take screenshots wherever possible.
3. Create a Gitlab repository, and add all the configuration files and screenshots of the task there. Also, explain the steps taken in the README. In the case of multiple tasks, create a separate branch for each task. Document all your tasks and hyperlink all the branches in your main branch.
4. The deadline for submission of the tasks is **17/03/2024 11:59 PM IST.**
5. Fill this form before the deadline to submit your tasks.
6. For any help regarding the tasks, or task submission, contact respective **POCs**.
7. **Feel free to submit any amount of work you've done for the tasks.**

Tasks

Deploy a Rails application available here using docker containers.
The following tasks are to be done:

1. Pack the rails application in a docker container image.
2. Launch the application in a docker container. Launch a separate container for the database and ensure that the two containers are able to connect.
    a. The DB port should not be exposed to the host or external network. It must be internal to the docker network only.
    b. Expose the application port to the host machine at port 8080. So you should be able to access the app at "localhost:8080".
3. Launch an Nginx container, and configure it as a reverse proxy to the rails application. Expose it at port 8080 on localhost. So now the rails application shouldn't be accessed directly. All requests will go through Nginx.

4. Now launch two more containers of the rails application. All three containers should be able to connect to a single database container. Configure Nginx container to load balance incoming requests between the three containers.
5. Enable persistence for the DB data and Nginx config files so that they are available even when the containers go down.
6. Use docker-compose to easily bring these containers up together with a single command.
7. Add requests rate limit to Nginx to limit the number of HTTP requests to the application in a given period of time.
8. Write a Daemon to take timely backups of the data and code.

Bonus Tasks

- Create a local Kubernetes cluster using minikube/kind.
    - Launch the rails application on the cluster.
    - Connect to the database running outside the cluster.
    - Design a solution to assign static IP addresses to each container/pod running in the Kubernetes cluster, ensuring accessibility.
- Implement a solution using nfs/smb/cif to share /storage folder on a separate VM/container amongst all rails instances.
- Write a Gitlab CI/CD pipeline that creates a docker image and uploads to the docker hub registry.

Submission

Form: https://iris.nitk.ac.in/form/sys-recs-2024

Resources

1. Gitlab Repo of the application
2. Docker Setup and Dockerize an Application
3. Overview of Docker Compose
4. Nginx Reverse Proxy for a Rails App with Docker
5. Use bind mounts
6. NGINX Rate Limiting
7. Kubernetes Tutorial

POC - Devaansh Kumar (+91 8050093839)
Vinit Puranik (+91 7406035232)
Whatsapp Group link