Shared Automation Initiative

I believe that nothing and no one should be left out of this new path, for that reason we want all to be part of the change…

Learn, Test, Automate and Share

Jonathan Prado

Cloud Architect Lead. Nerd & Rider all the time.

Ninja skils:

- Bash
- Python

Ninja tools:

- Docker & compose
- Kubernetes
- Terraform
- Chaos Engineering

***AWS**:*

EC2, RDS, EKS, ELB, S3, R53, EBS

Objectives

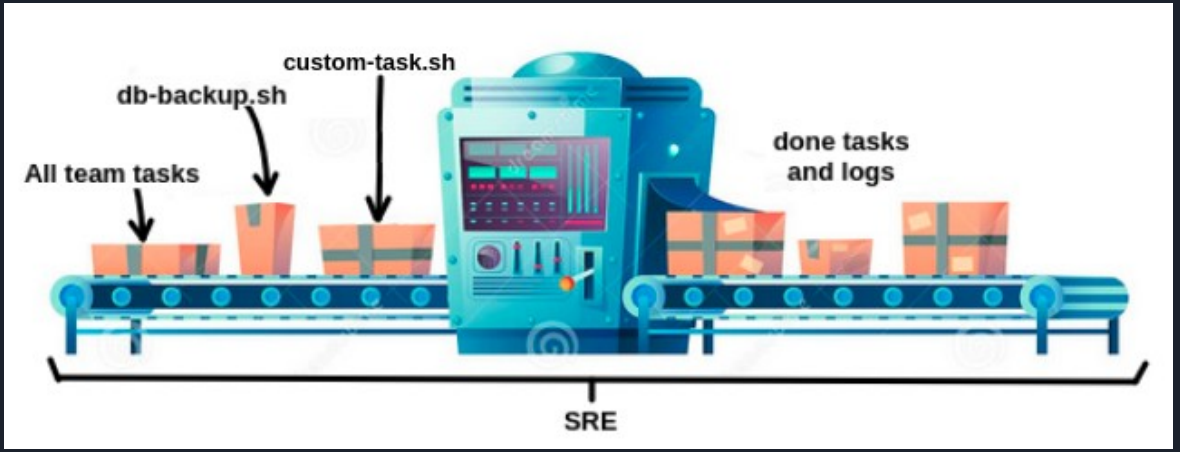
Our objective is **Improve the “Customer Lifetime Value”**.
To do it we need some actions:

- Reduce manual intervention.
- Automate as many manual tasks as possible.
- Share knowledge between areas.

Common Problems:

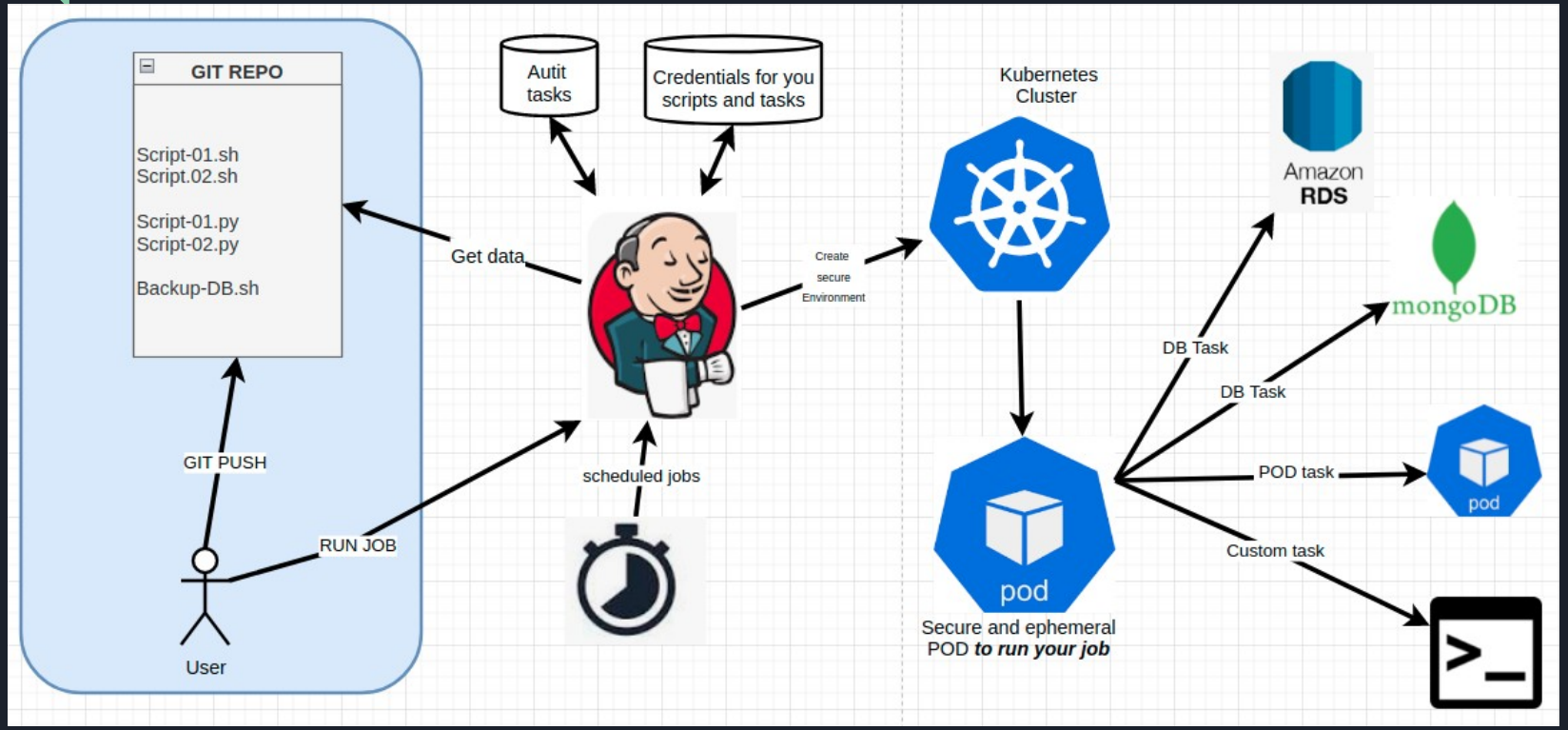
- **Security risks:** we have a lot users and access points to secure and maintain, this goes against the security best practices *(Least privilege).*
- **Administrative tasks overhead:** We saw a lot of same and recurrently jobs, scripts executions that are executed by humans.
- **Auditory and traceability:** We can see or follow all execution flows.
- **Knowledge is not shared:** We depend on a particular human to do some things, this is a big problem because if it person can't continue doing something no one else can.

Our proposal



- **Shared automaton:** sharing information everyone can participate in the improvement of the platform.
- **Centralized jobs:** Each team can create and maintain our own scripts in a centralized gitlab repo.
- **Security risks:** Create a only one access point to do all tasks.
- **Administrative task overload:** separating the tasks, there will be no dependencies between teams (we will use a new concept: "***put it here and execute it where and when you want***").
- **Auditory and traceability:** Centralizing logins we can eliminate risks of shared credentials and delete. it when someone leave, and we can generate auditory of all access and all activities.
- **Knowledge is not public:** We don't have secrets, our code is shared with all and it can be maintained by all
- **AaaC:** All as a code, all jobs or tasks need to be code, because it can be improved, managed and save & restore easily and can be checked in a control version platform (like git)

Continuous Automation - High level design



Demo

Doubts or questions