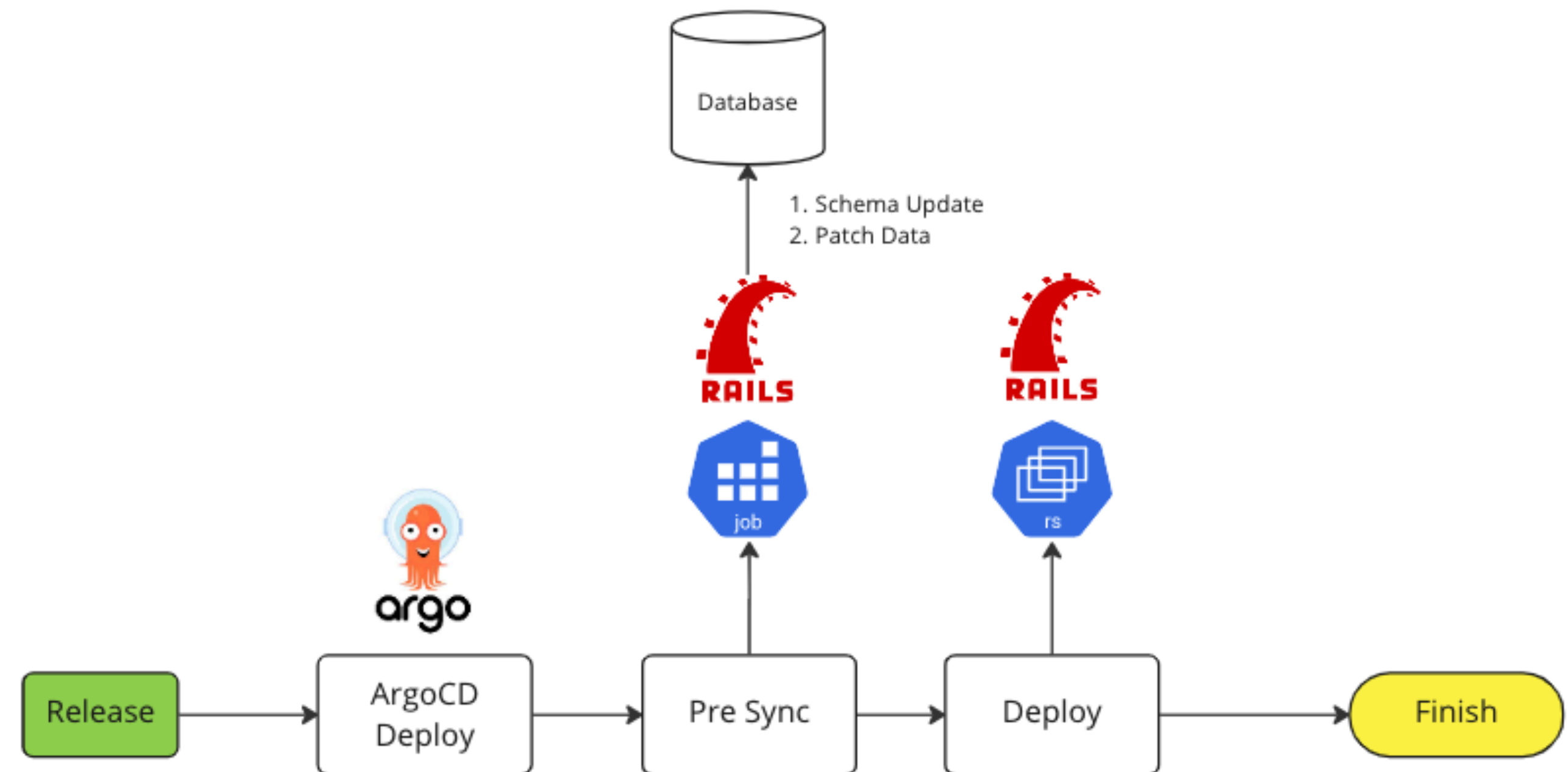


# Run Rails data-migrate at two phases in deployment

# Background About Deployment Cycle

## At Amazingtalker

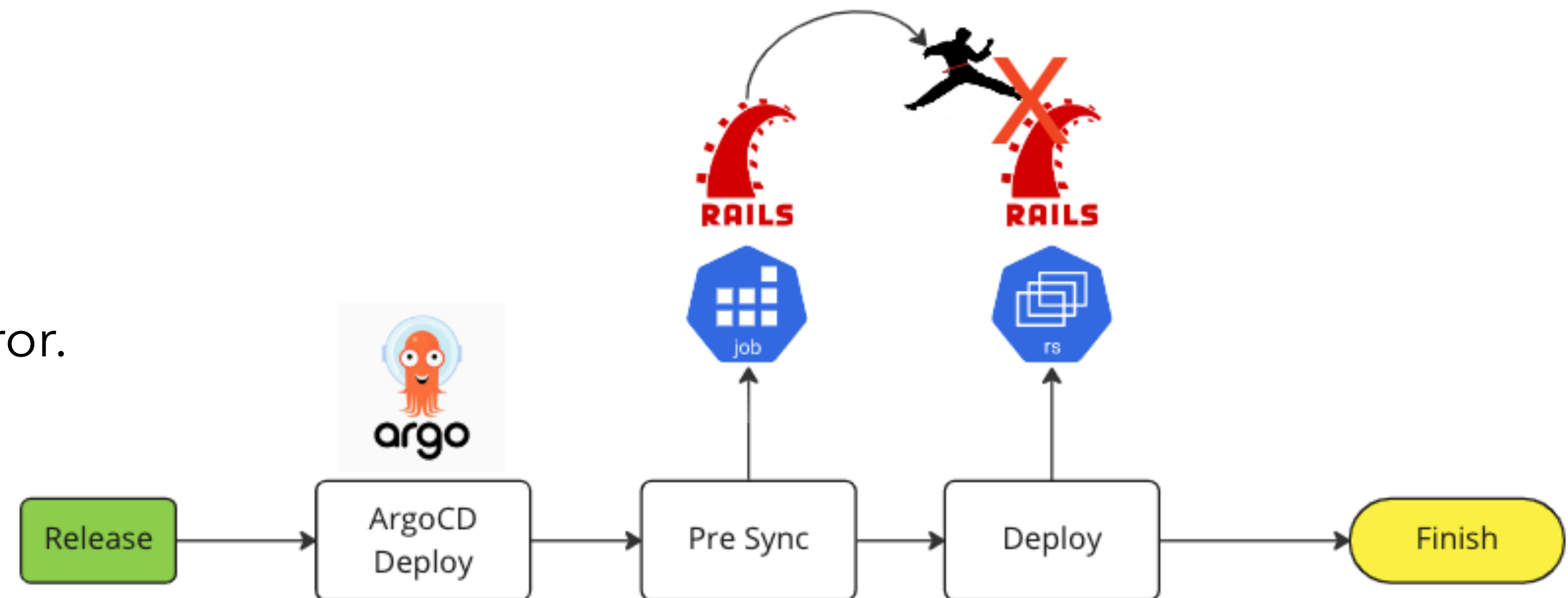
- Operate on K8s (AWS EKS)
- GitOps by ArgoCD
- Pre sync job before deploy
  - DB schema change
  - Patch dirty data



# Background About Deployment Cycle - Problem

## At Amazingtalker

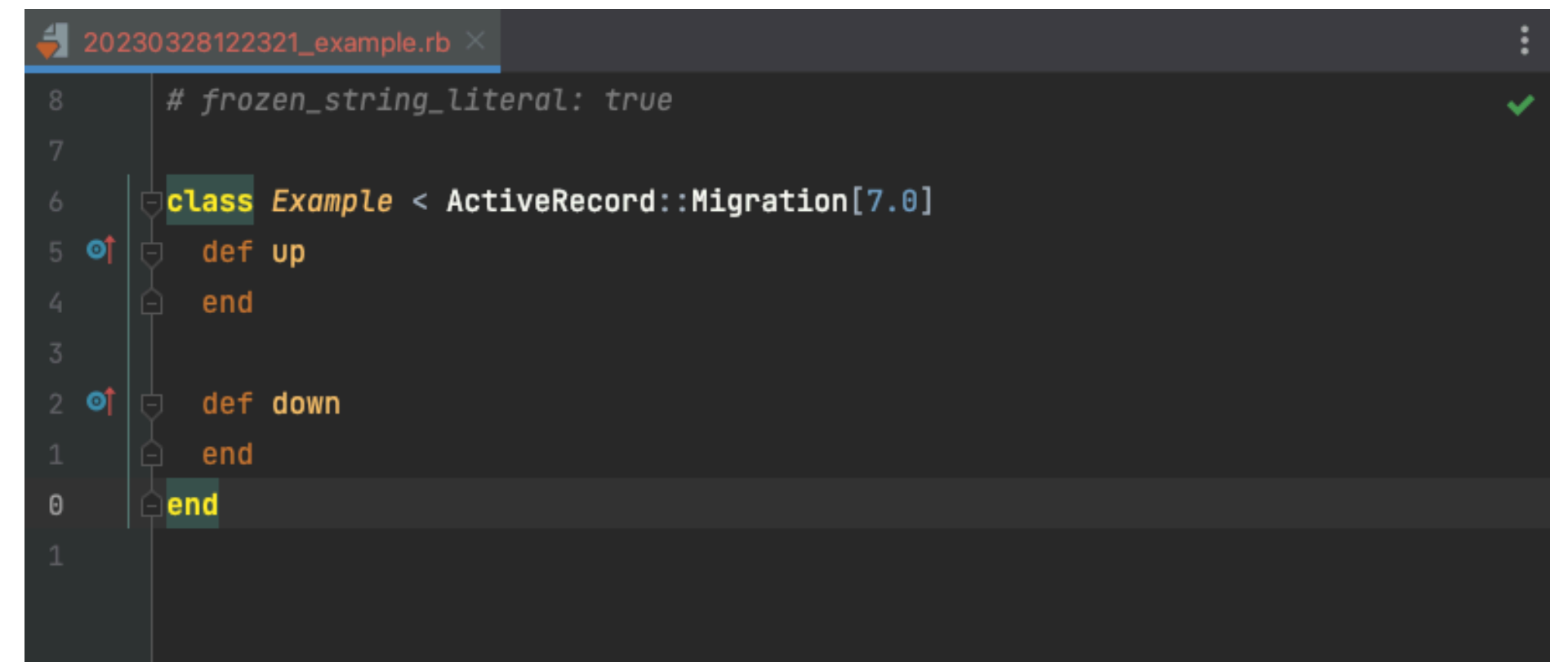
- Missing Part: some operation needs to be trigger after deploy
- Trigger new Sidekiq Job
  - Trigger at Pre Sync
    - Sidekiq Server not upload yet, throw error.
  - Deploy at next cycle.
    - Bad DX



# Post Sync Expectation

Run data-migrate again in another phase

- Generate migration script template
  - `$ bin/rails generate db_migration example`
- Execution
  - `$ bin/bundle exec rake data:migrate`
  - Prevent same task run several times
- Isolation

A screenshot of a code editor showing a Ruby migration script. The file is named '20230328122321\_example.rb'. The code defines a class 'Example' that inherits from 'ActiveRecord::Migration[7.0]'. It includes two methods: 'up' and 'down', both of which are currently empty. The 'up' method is on line 5 and the 'down' method is on line 2. The script ends with an 'end' statement on line 0. A green checkmark is visible in the top right corner of the editor window.

```
8 # frozen_string_literal: true
7
6 class Example < ActiveRecord::Migration[7.0]
5   def up
4     end
3
2   def down
1     end
0   end
1
```

# Post Sync - Generator

Run data-migrate again in another phase

- Rails Generator can generate template file dynamic




```
class InitializerGenerator < Rails::Generators::NamedBase
  source_root File.expand_path('templates', __dir__)

  def copy_initializer_file
    copy_file "initializer.rb", "config/initializers/#
{file_name}.rb"
  end
end
```

Copy

And let's execute our generator:



```
$ bin/rails generate initializer core_extensions
```

Copy

# Post Sync - Generator

Run data-migrate again in another phase

- Generator can trigger other Generator
  - Get params from CLI
- \$ rails generate post\_sync xxxx\_xxx

```
# lib/generators/post_sync_generator.rb

class post_syncGenerator < Rails::Generators::NamedBase
  def copy_initializer_file
    DataMigrate.configure do |config|
      config.data_migrations_path = POST_SYNC_PATH
    end
    Rails::Generators.invoke("data_migration", [file_name])
  end
end
```

# Post Sync - Execution

Run data-migrate again in another phase

- Rake task execution and dependency tool
  - Trigger other rake task
- \$ rails generate post\_sync xxxx\_xxx

```
# lib/tasks/post_sync.rake

namespace :post_sync do
  task :migrate do
    DataMigrate.configure do |config|
      config.data_migrations_path = POST_SYNC_PATH
    end

    Rake::Task['data:migrate'].invoke
  end
end
```



# Post Sync - Check

Run data-migrate again in another phase

- Store records in the same DB table.
- Check file names in CI

```
class PostSyncChecker
  CHECK_FOLDER = [
    DataMigrate.config.data_migrations_path,
    POST_SYNC_PATH
  ]

  class << self
    def is_collision?
      (pre_sync_versions, post_sync_versions) = CHECK_FOLDER.map do |path|
        list_migration_versions(path: File.join(Rails.root, path))
      end

      puts pre_sync_versions
      (pre_sync_versions & post_sync_versions).present?
    end

    def list_migration_versions(path:)
      Dir.entries(path)
        .map { |filename| parse_version(filename: filename) }
        .compact
    end

    def parse_version(filename:)
      result = /\d{14}_(.+)\.rb$/.match(filename)
      return unless result.present?
      result[1]
    end
  end
end
```

[Follow link \(cmd + click\)](https://github.com/ilyakatz/data-migrate/blob/2dd90c495b4d57e4dc700e3f9be149c7e2b93b57/lib/data_migrate/data_migrator_five.rb#L44)

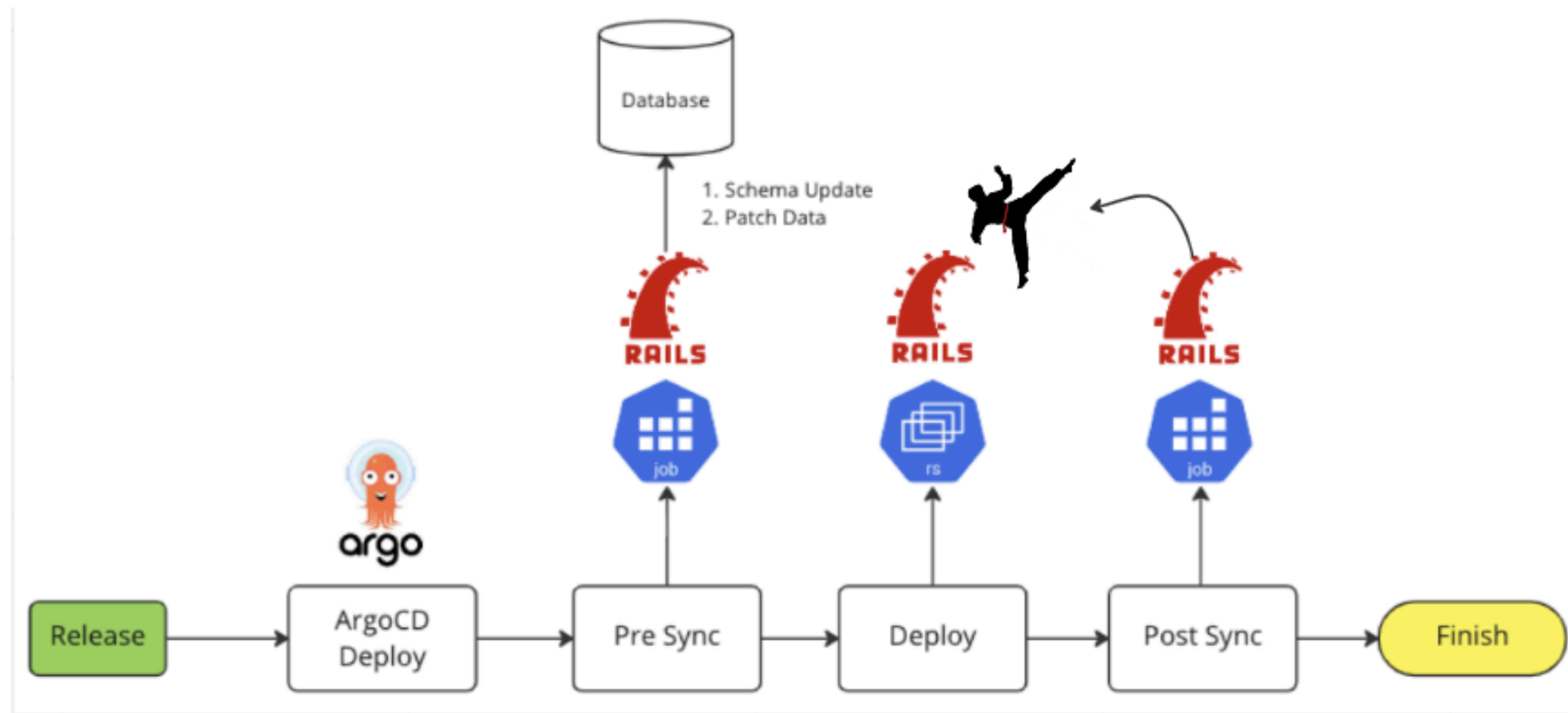
# ref: [https://github.com/ilyakatz/data-migrate/blob/2dd90c495b4d57e4dc700e3f9be149c7e2b93b57/lib/data\\_migrate/data\\_migrator\\_five.rb#L44](https://github.com/ilyakatz/data-migrate/blob/2dd90c495b4d57e4dc700e3f9be149c7e2b93b57/lib/data_migrate/data_migrator_five.rb#L44)



# Recap

- Remain db migration and data migration at Pre Sync Phase
- Trigger new Sidekiq Job at Post Sync Phase

└─ db/  
└─ data  
└─ schema  
└─ post\_sync



# Ref

- Data-migrate: <https://github.com/ilyakatz/data-migrate>
- Rails Generator: <https://guides.rubyonrails.org/generators.html>
- My blog post: <https://yuanchieh.page/posts/2023/2023-03-28-rails-%E9%83%A8%E7%BD%B2%E5%84%AA%E5%8C%96-%E9%AD%94%E6%94%B9-data-migration-%E5%AF%A6%E4%BD%9C-post-sync-%E6%A9%9F%E5%88%B6/>