

## Helpful terms

- **Failure Domains** - know your abstractions

### Avoid:

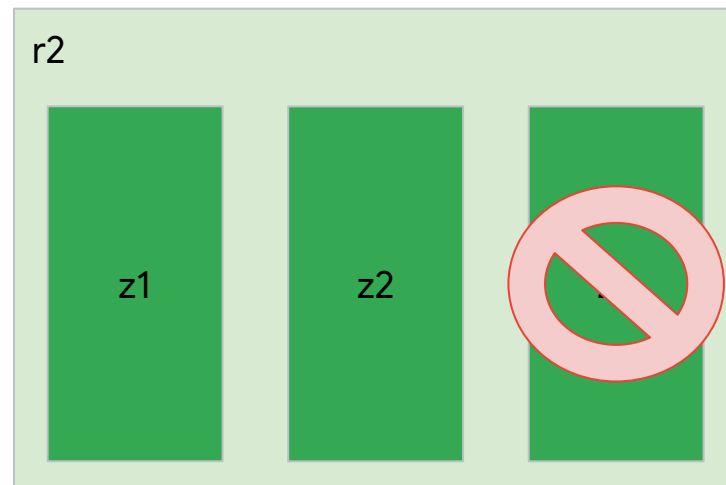
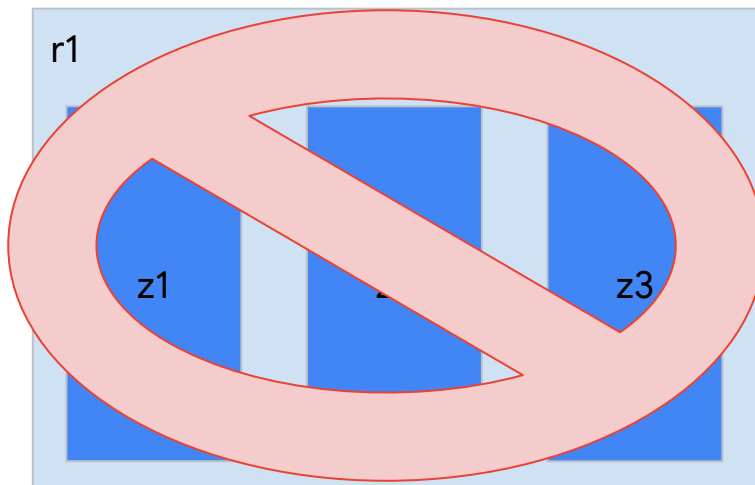
- **Coordinated Failure** - isolated change
- **Cascading Failure** - plan for containment

### Via:

- **Gradual Change** - fail early, fail small

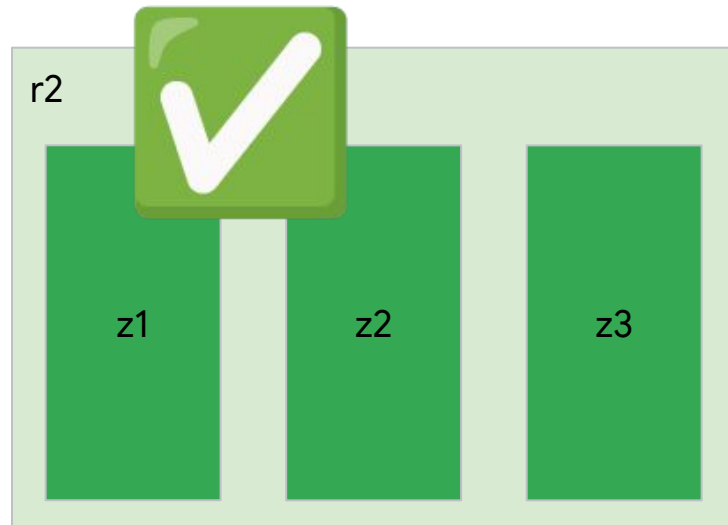
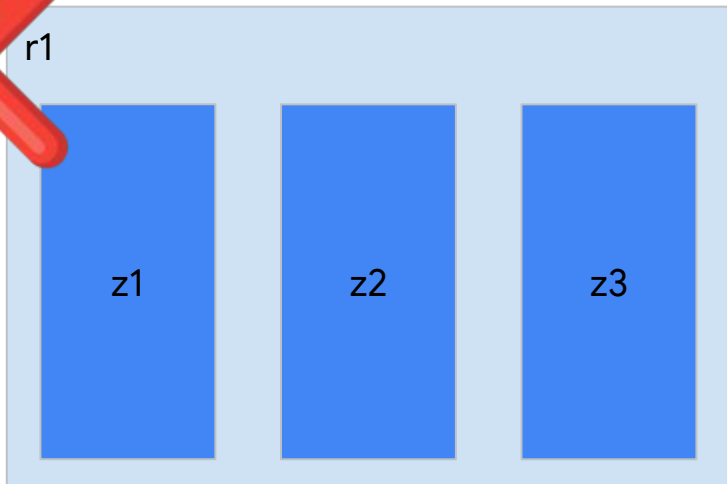
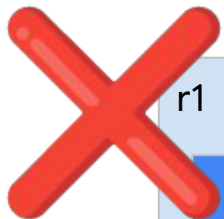
## Helpful terms

- **Failure Domains** – know your abstractions



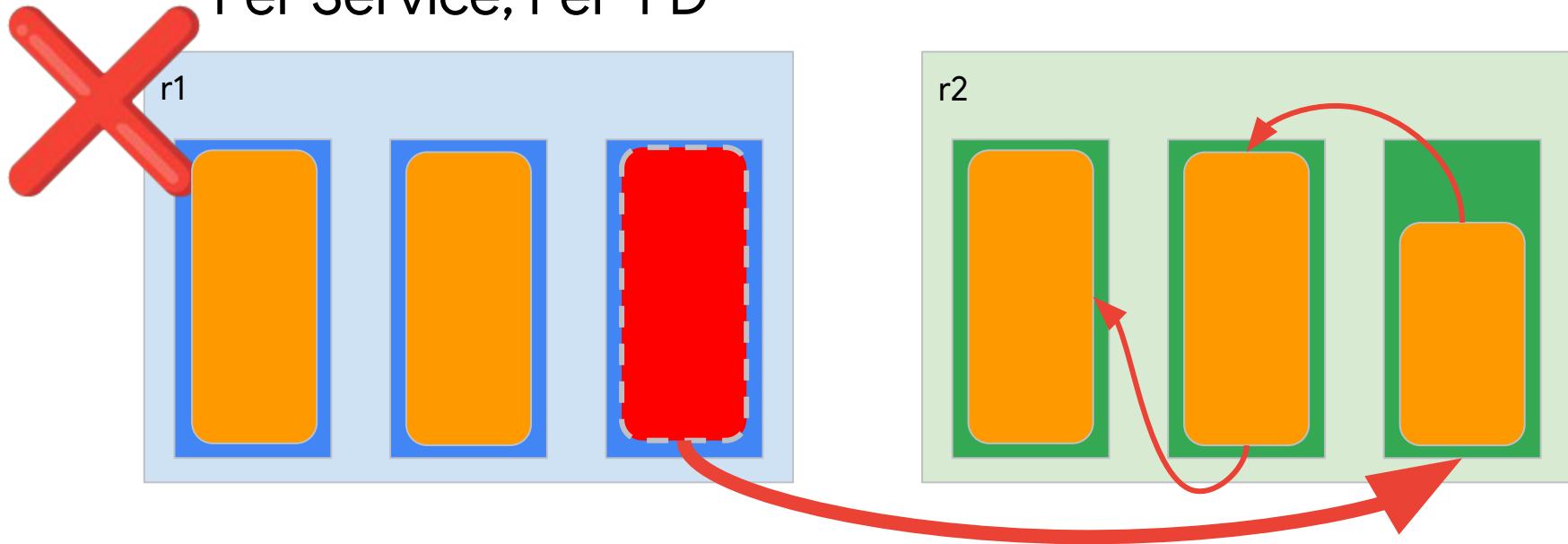
## Helpful terms

Avoid **Coordinated Failure** – isolate changes *by FD*



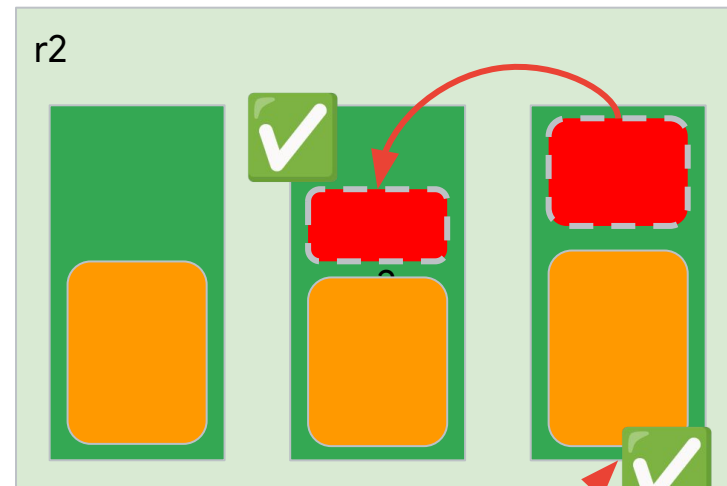
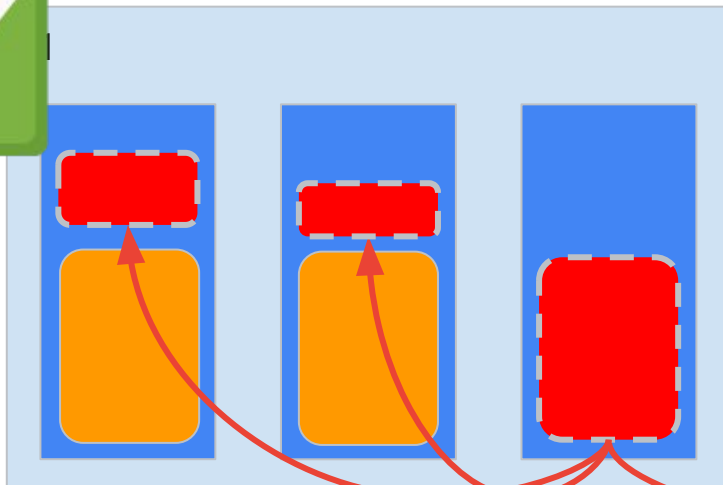
## Helpful terms

Avoid **Cascading Failure** - capacity plan for  $N+1$  ( $N+2$ )  
- Per Service, Per-FD



## Helpful terms

Avoid **Cascading Failure** - capacity plan for N+1 (N+2)  
Per Service, Per-FD

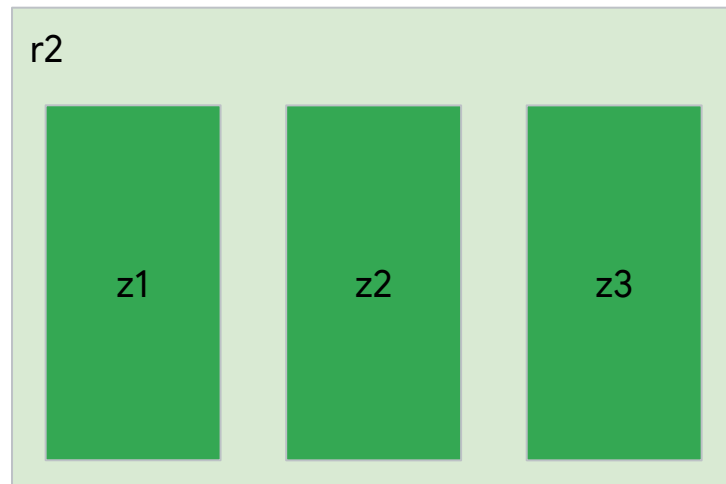
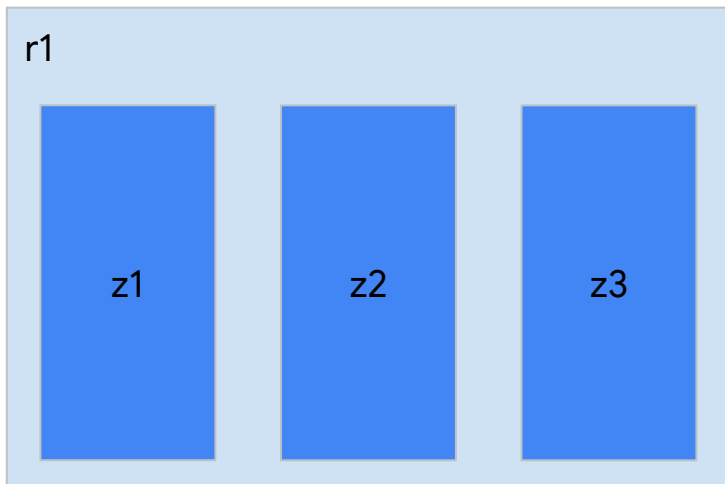


x-region "spill"



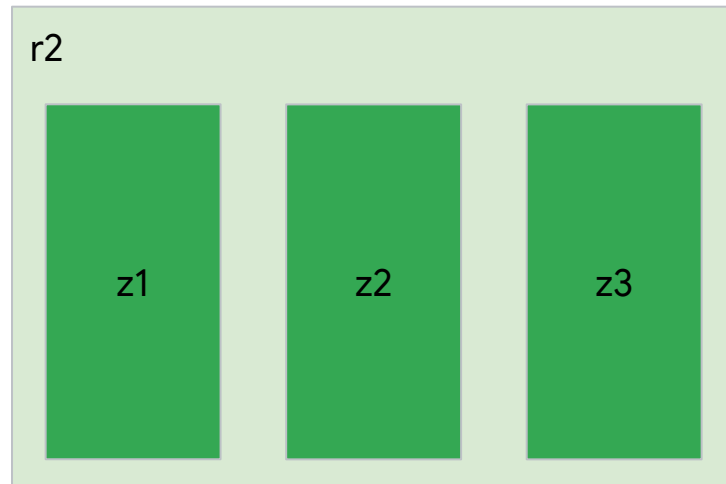
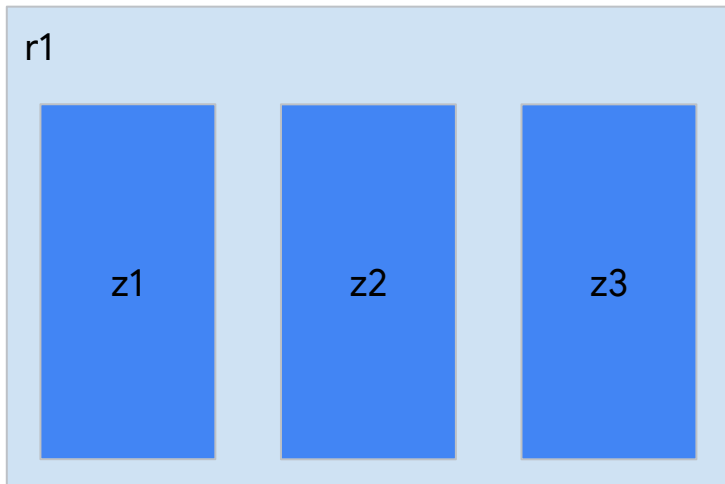
## Helpful terms

Via **Gradual Change** – fail early, fail small



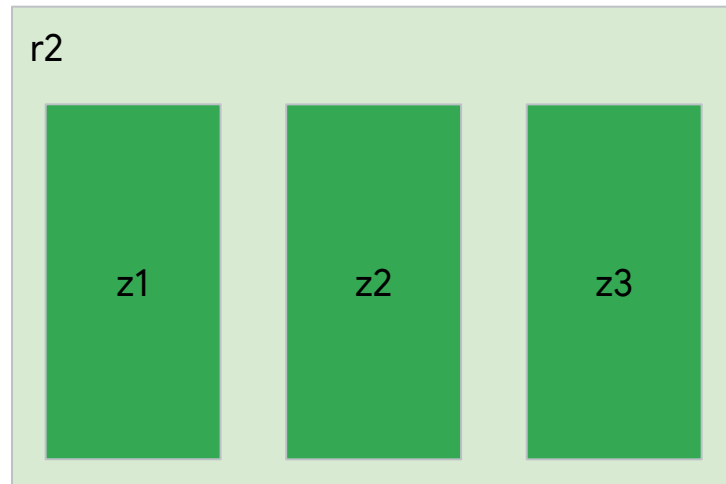
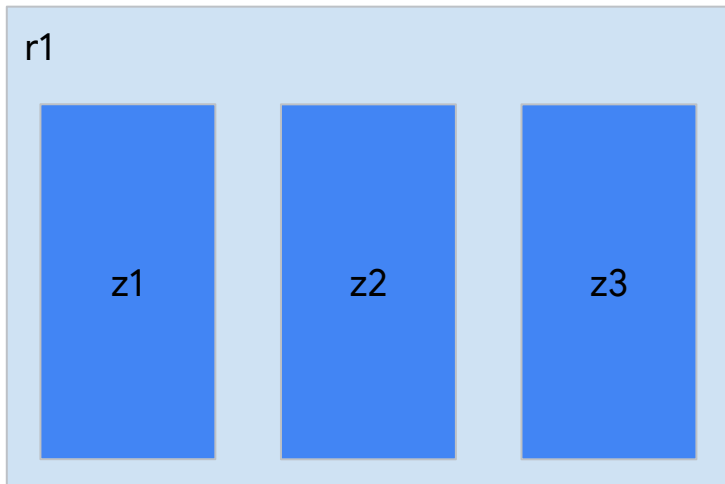
## Helpful terms

Via **Gradual Change** – fail early, fail small



## Helpful terms

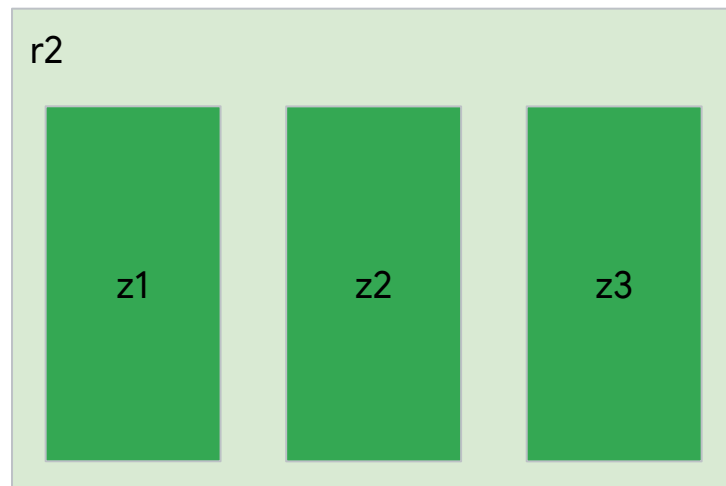
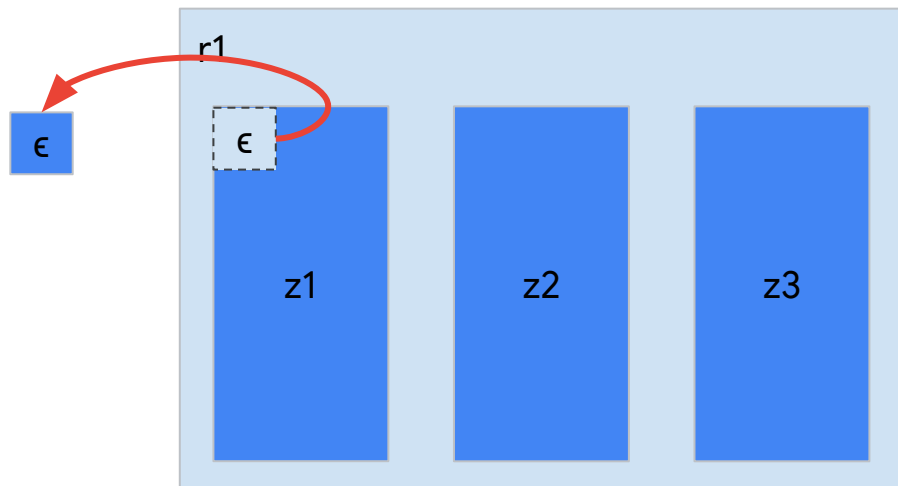
Via **Gradual Change** – fail early, fail small





## Helpful terms

Via **Gradual Change** – fail early, fail small



## Helpful terms

- **Failure Domains** - know your abstractions

### Avoid:

- **Coordinated Failure** - isolated change
- **Cascading Failure** - plan for containment

### Via:

- **Gradual Change** - fail early, fail small

# But Why?

- Accepting failure means you don't have to know everything ahead of the time!
- Embracing Failure (just keep it small?)
- Learning from Incidents
  - [learningfromincidents.io](https://learningfromincidents.io)
  - "why didn't we just ..." "so this will never happen again"
- **"Run from your problems"**
  - Don't expect to fix/prevent them immediately