#ASLI ENGINEERING

Blue Green Deployments

SWIPE

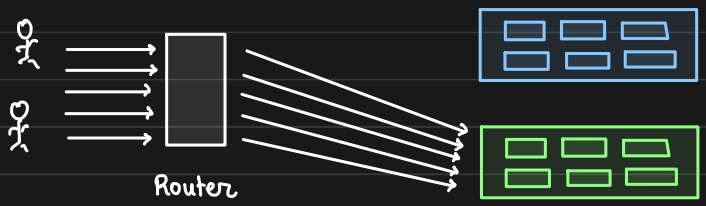
BY

ARPIT BHAYANI

Blue Green Deployments

Blue Green Deployment is a deployment pattern that reduces the downtime by running two identical production environments called Blue and Green

* at one time, only one of the environment is live



Green is live, while the Blue env is idle.

During an API server restart, the in-flight requests are hampered and the server does become unresponsive.

Server needs time to be fully functional → transient errors
micro-downtime 5xx

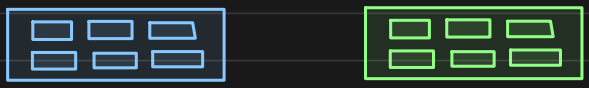
Three key advantages we get here are

- simple rollout
- quick rollback
- easy disaster recovery → expensive though

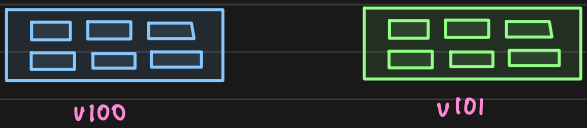
How Blue Green Deployments are implemented? Need of a proxy

In order to do a Blue Green Deployment, we

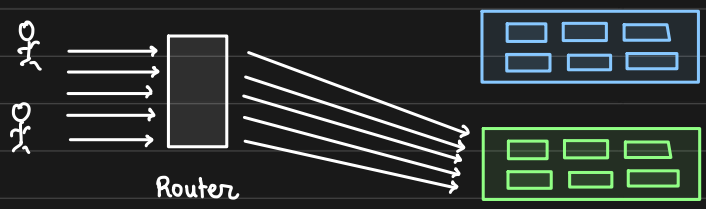
1. ensure the changes are forward and backward compatible
2. setup a parallel infrastructure (API server fleet)



3. deploy the new version to the new fleet "green"



4. validate the correctness of the new setup - QA, sanity, vitals
5. configure the proxy to switch 100% traffic to green



6. After everything validated, remove/stop the Blue infrastructure

Pros of having Blue Green Deployments

- Rollbacks are superfast and simple

Rolling back the changes to the previous state is just a route change

- downtime during deployments is minimized

The incoming traffic is instantly flipped from Blue to Green setup, leaving no scope of downtime.

* No period where no request could be served

- deployments are quick

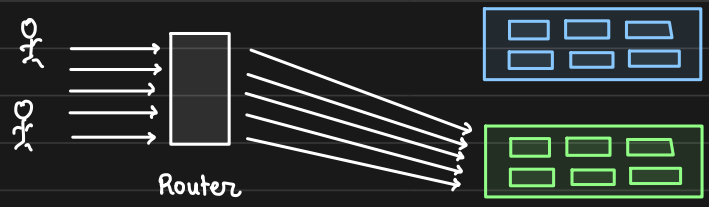
Just a flip of a switch, no need to wait for changes to happen from one server to another.

- disaster recovery plan is super simple
- deployments can happen in regular hours
- we can debug why a release failed

Possible challenges of doing a Blue Green Deployment

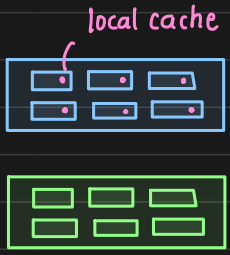
- during deployments the infra cost would be 2x

Separe fleet of infra during deployment + for some more time



- Stateful applications would take a hit

The information stored on instances like cached files, artifacts, will be lost when we switch from Blue to Green



* Everything would need to be rebuilt if it is something we cannot recreate, then we cannot use BG

- Database migrations

time for data copy ↖ ↗ cost involved

In Blue Green Deployment, the DB is not copied Hence we need to ensure forward and backward compatability in schema alterations

- Forward and Backward compatability for APIs

  Along with database migrations, we would need to ensure that the API responses across versions are compatible

  eg: new attribute added in response

  previous version of app should still work with new API response

- handling shared services across Blue Green setup

  we need to be extra careful on how shared services would behave across setup

- it takes time and efforts to have a Blue Green setup

When to use Blue Green Deployments?

- you wish to have no downtime deployment
- you can tolerate a 100% instant switch to new infra / version
- you can bear the cost of running 2x infra

Points to remember

- have a solid automation test suite
- ensure forward / backward compatability
- before switch validate the setup
- infra cost will shoot up, minimize the time for which both infra are up.