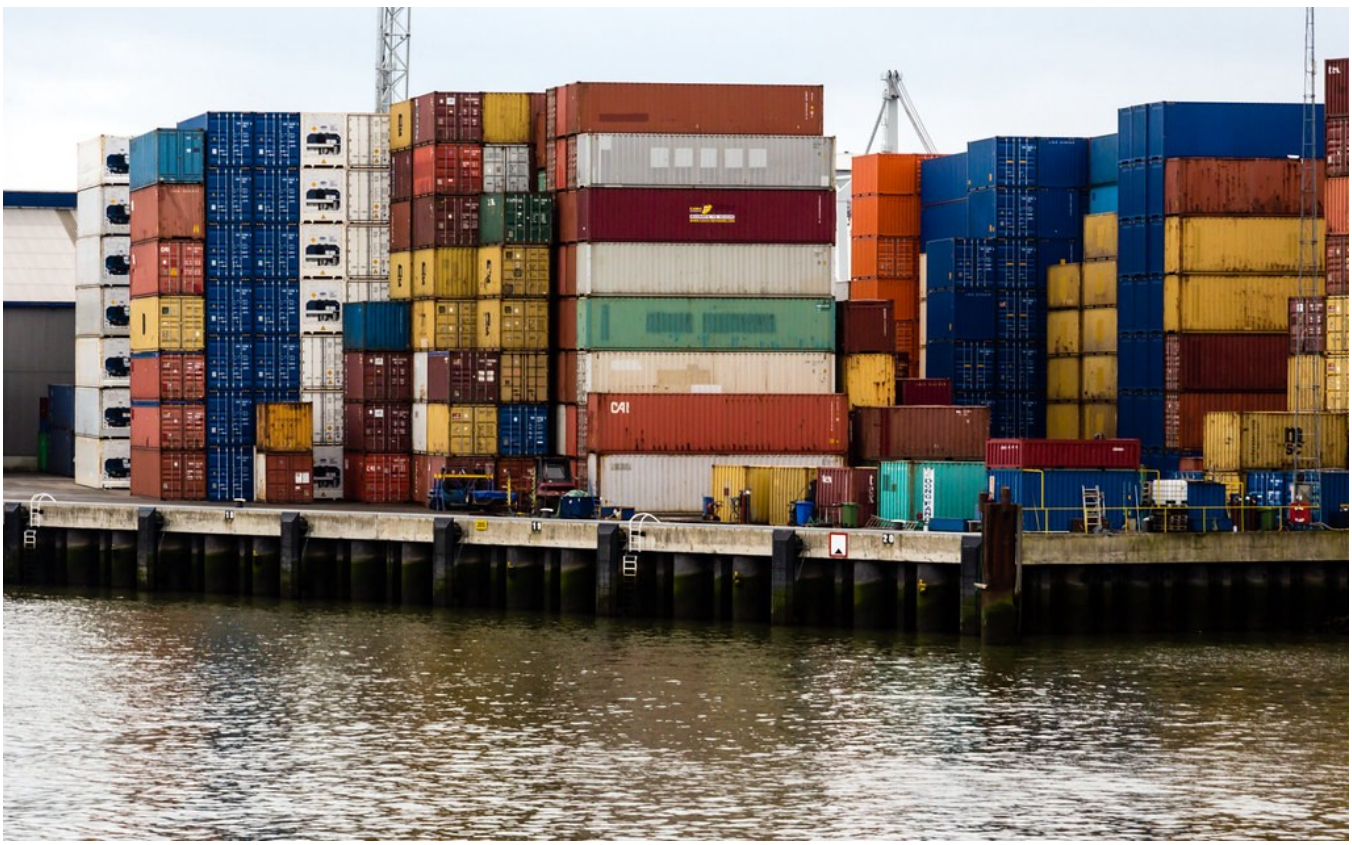
How Can You Automate Your DevOps Pipeline With Blue Green Deployments

Craig Godden-Payne
Mar 8 · 3 min read ★



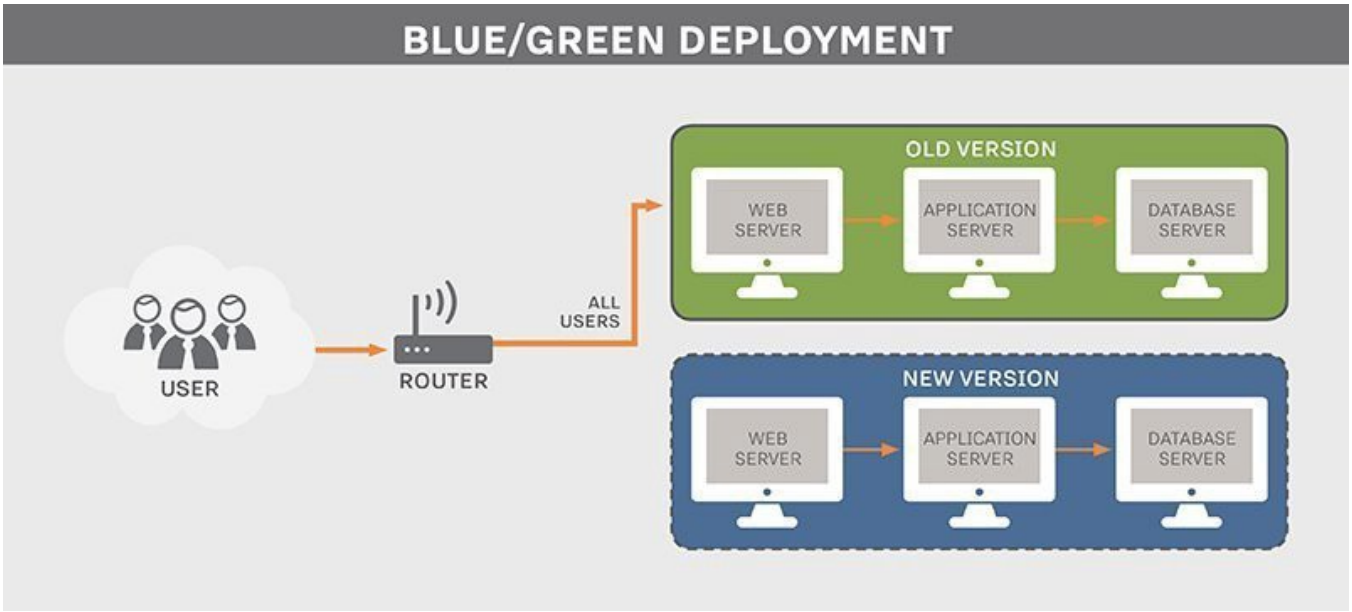
There are several challenges when automating deployments, the main being managing the switch over from version 1, to version 2 without any downtime.

If software is running in a production environment, it is just bad if you have to show a maintenance page, or even have a short downtime. There are techniques to help mitigate these issues, and one being a blue green deployment.

What is blue green?

The idea behind blue green is by having two production environments, which are as identical to each other as can be.

The two environments can be switched between each other with minimum effort.



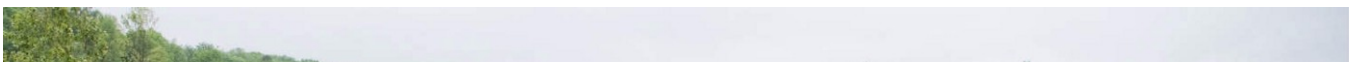
In the example above, the green environment is live, and we can deploy our new version 2 of the software to the blue environment. We can perform any tests needed on this environment before we switch the router to point to Blue.

Rollbacks

Say you have an issue with the new deployed blue environment, maybe you have seen a traffic drop, or an increase in 500 errors, it is in theory as simple to roll back the deployment.

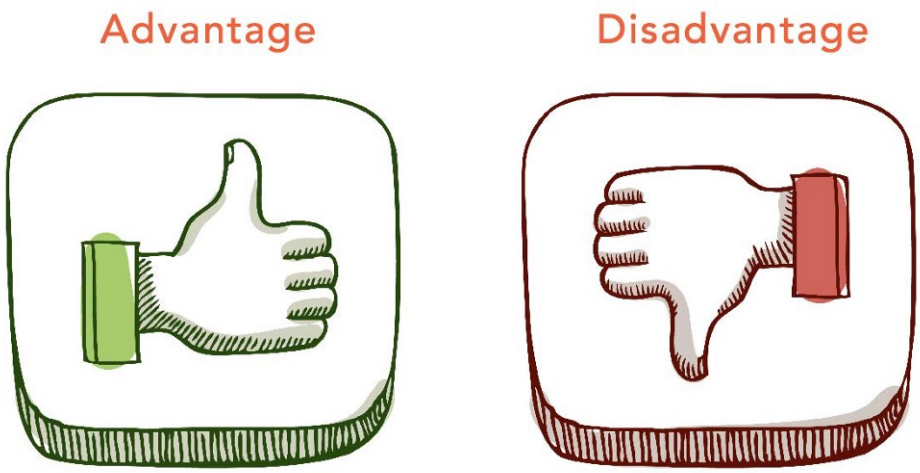
You just switch the router from Green to Blue.

- There are some caveats to this being as simple to rollback, you may have to replay any missing transactions that were applied to the Blue environment, whilst the Green environment was live.



What are the main advantages of using Blue Green?

- You are able to deploy to a live like environment and test before actually making the deployment live to the public. This can be a massive advantage if you do not have the confidence that the change may break something in production.
- You have a standby working environment at anytime. Whether this is for disaster recovery purposes, or for rolling back. A standby environment ready to go has huge advantages.

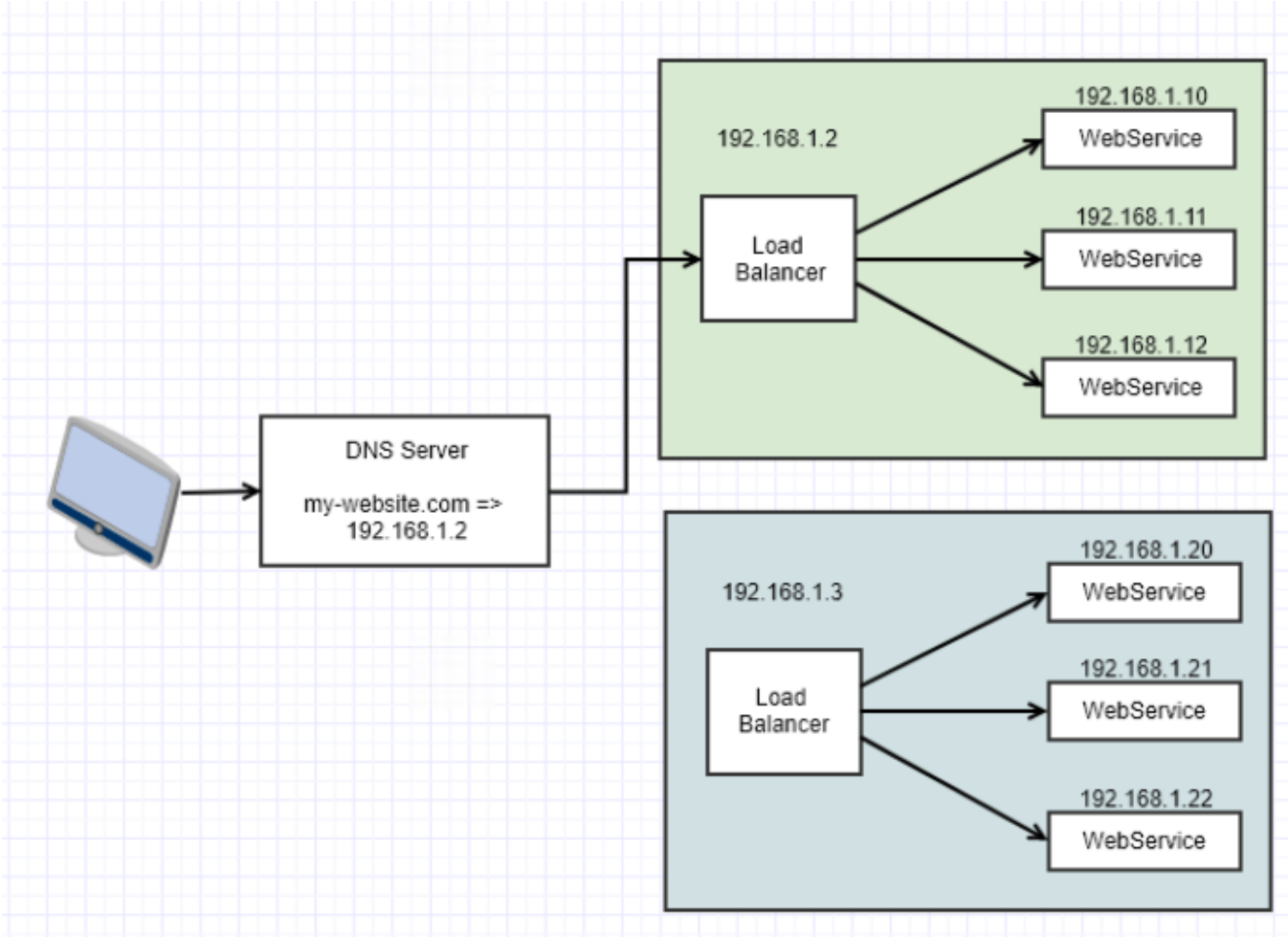


What are the downsides?

- Managing database schema changes can be complicated, as it is much more complex to rollback a database schema change than to forward fix manually. To mitigate this, it is sometimes to treat the database as a separate entity, which has a forward changing schema, no edits etc. although this is not always the most practical.

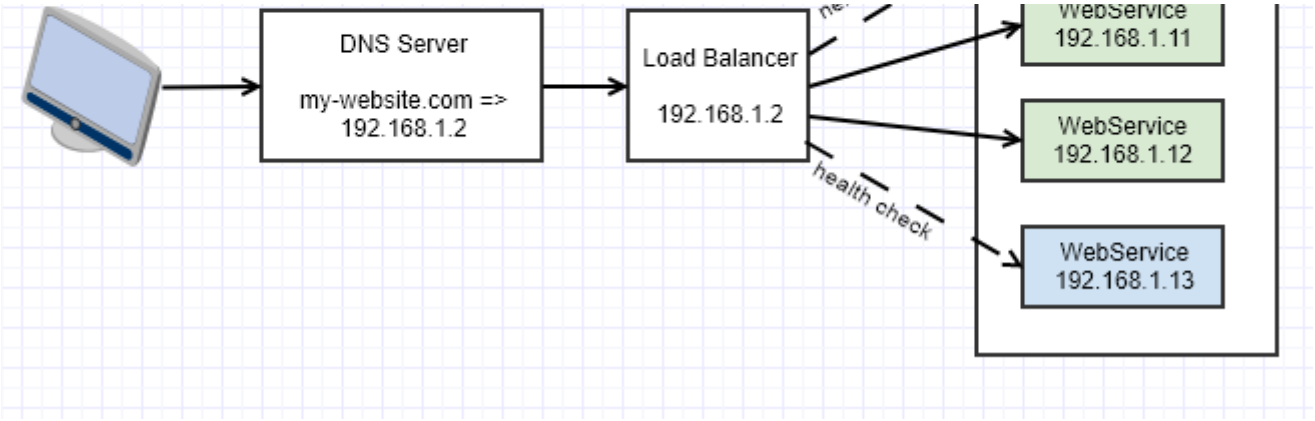
But how can we do a blue green deployment?

DNS switching of load balancer



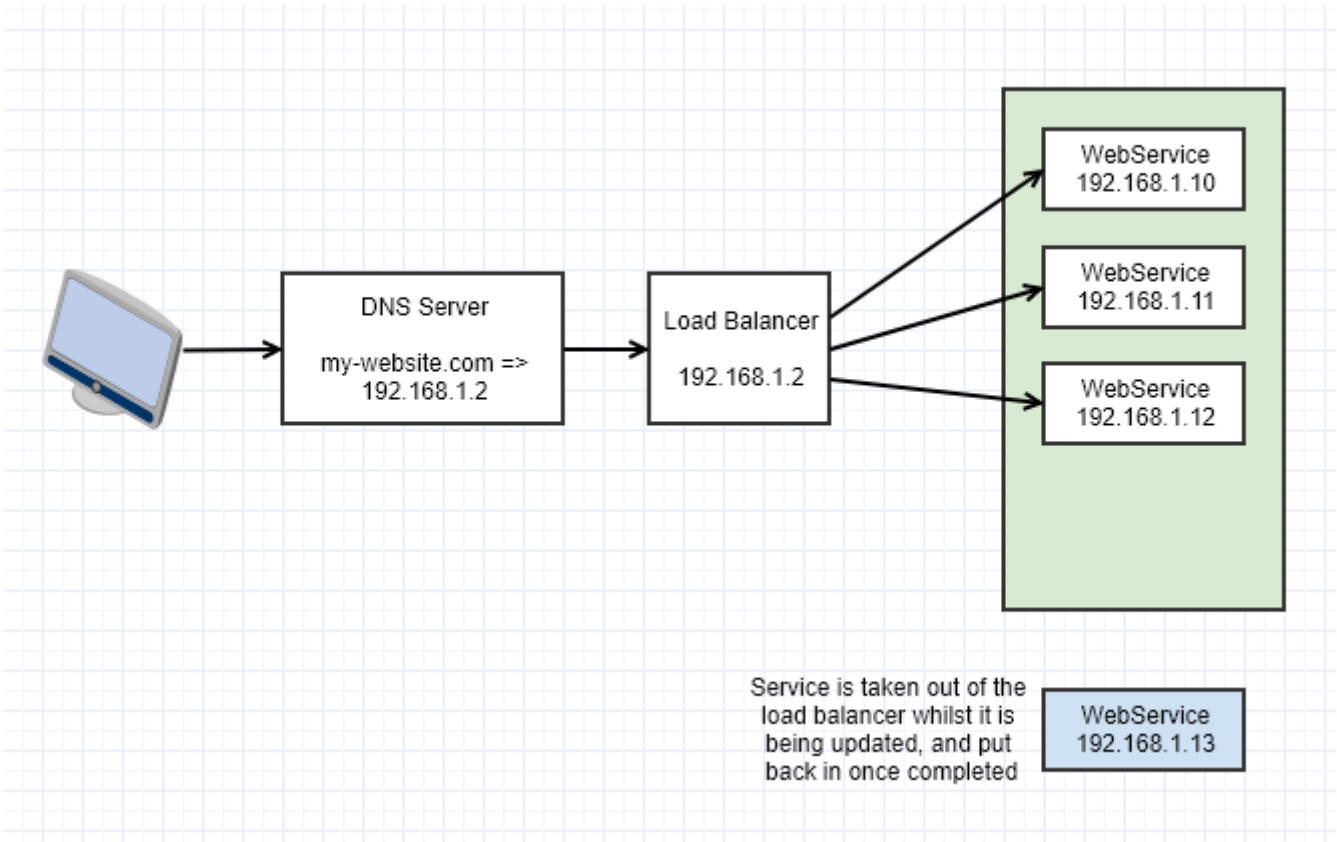
You can use DNS switching between the two environments

Load balancer switching



You can switch the target of the load balancer

Approach using physical / virtualised machines



This is more to show how this would have been done using legacy techniques.

Automation Deployment Blue Green Deployment Load Balancing DevOps


Download on the
App Store


GET IT ON
Google Play