



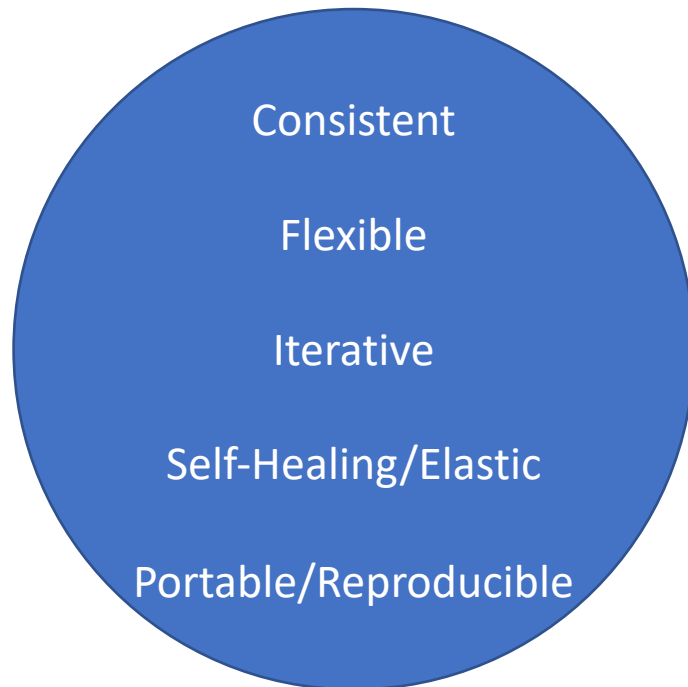
Why CI/CD?

Let me tell you a story...

Imaging 'The Business' coming with an idea for the company, and they request the developers to create a tool or an app. After some meetings, the developers spend days or months coding on their laptops, and finally, they deliver an application to the 'IT Operations guys' to deploy it in Production. IT tells 'Support' that there is a new tool they need to provide support to the customer but no more information or instructions. Support tests in Production and requests changes because some features don't work. IT must communicate with the developers, and they spend some time investigating because 'it works for them'. Finally, the definitive version reaches 'Security', and sadly, the new app doesn't meet with some of the security policies of the company...

We have spent maybe one year and significant money trying to create an application in the old way.

It sounds absurd, BUT it happens more than you could imaging. To avoid this, we would have to aim for the 'ideal system', and it looks something like this:



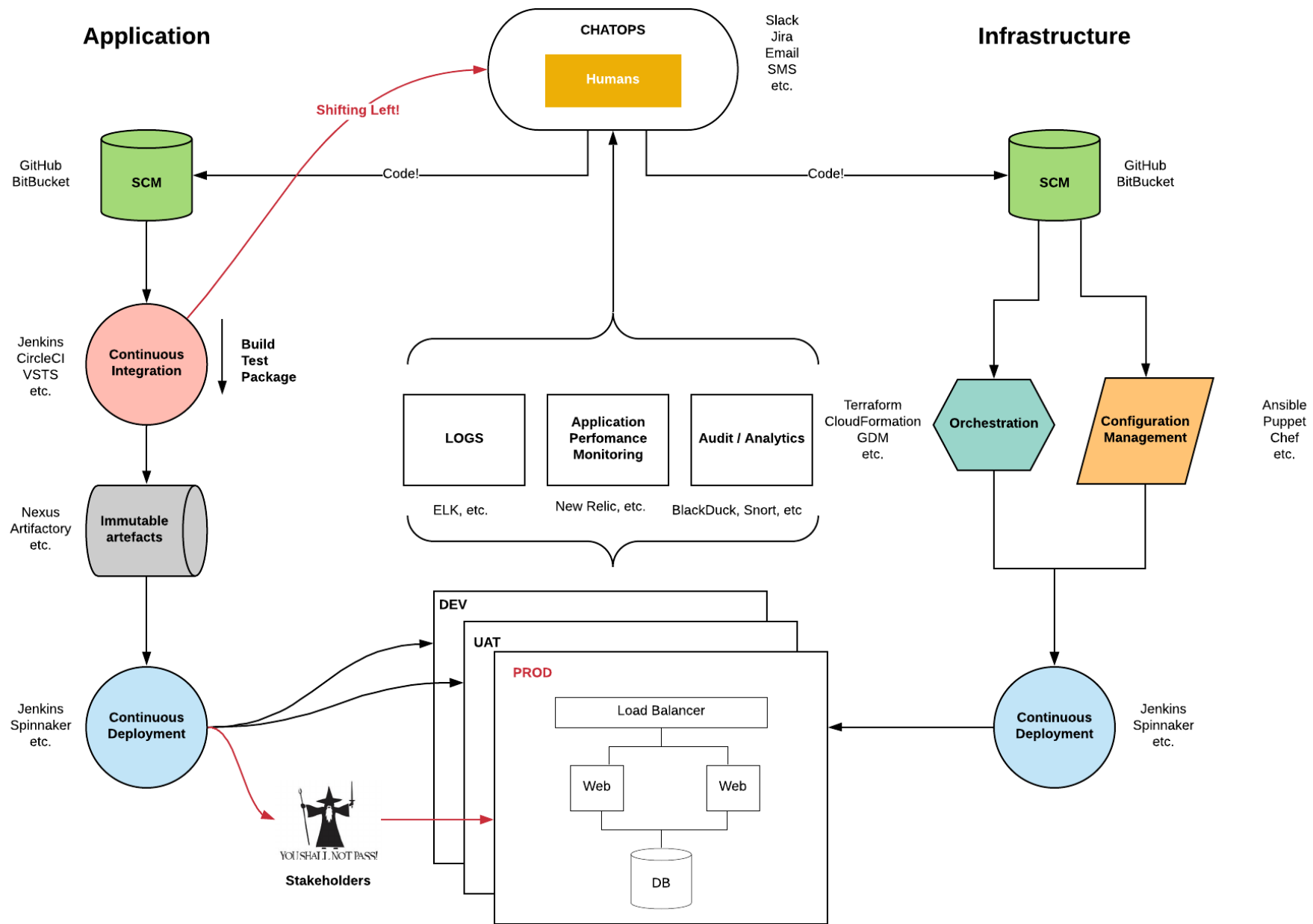
Consistent: All the env should be the same, but also the processes.

Flexible: If the business requires a change, the system should be able to adapt to the changes without lot of pain.

Iterative: You should be able to add new features easily in loops to keep customers happy on their requests.

Self-Healing / Elastic: The system should be able to recover from errors and also grow or shrink with the demand.

Portable / Reproducible: The system must run in a laptop or in a serverless architecture without pain.



Infrastructure

- The code to create the resources needed for the infrastructure is stored in a SCM (like GIT or Bitbucket)
- Using Orchestration (like Terraform or Cloudformation), we deploy that code and with a Configuration Management (Ansible or Puppet) we can configure the infrastructure.
- Using Continuous Deployment software (Jenkins or Spinnaker), we deploy the code into the desired environments (dev, uat, prod...)

Application

- Same as for the infrastructure, the code for the application is store in a SCM (like GIT or Bitbucket)
- With Continues Integration we can build, test and create a package of the code (Immutable artefact that moves across the different environments)
- Using Continuous Deployment software (Jenkins or Spinnaker, we deploy the application into the environments automatically (usually except for Production which probably need approval from stakeholders)

Benefits

- Using SCM we create a collaborative environment.
- CI tools help you compiling and testing your code much faster and preventing bugs when releasing to Production, avoiding time from developers and costs from downtime of the application
- Using CD tools, we can automate the infrastructure creation, avoiding time costs, and possible errors. Also, these deployments are much faster