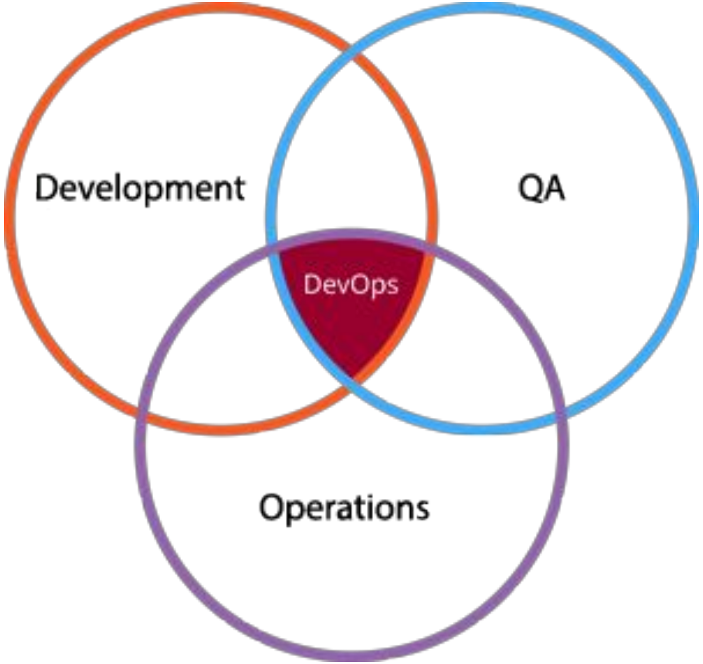
CI/CD

Common principles and planning

Agenda

- About me :)
- Who is DevOps
- What is CI/CD
- Infrastructure
- Instruments

Who is DevOps



- Devops: PPP
  - People
  - Process
  - Product

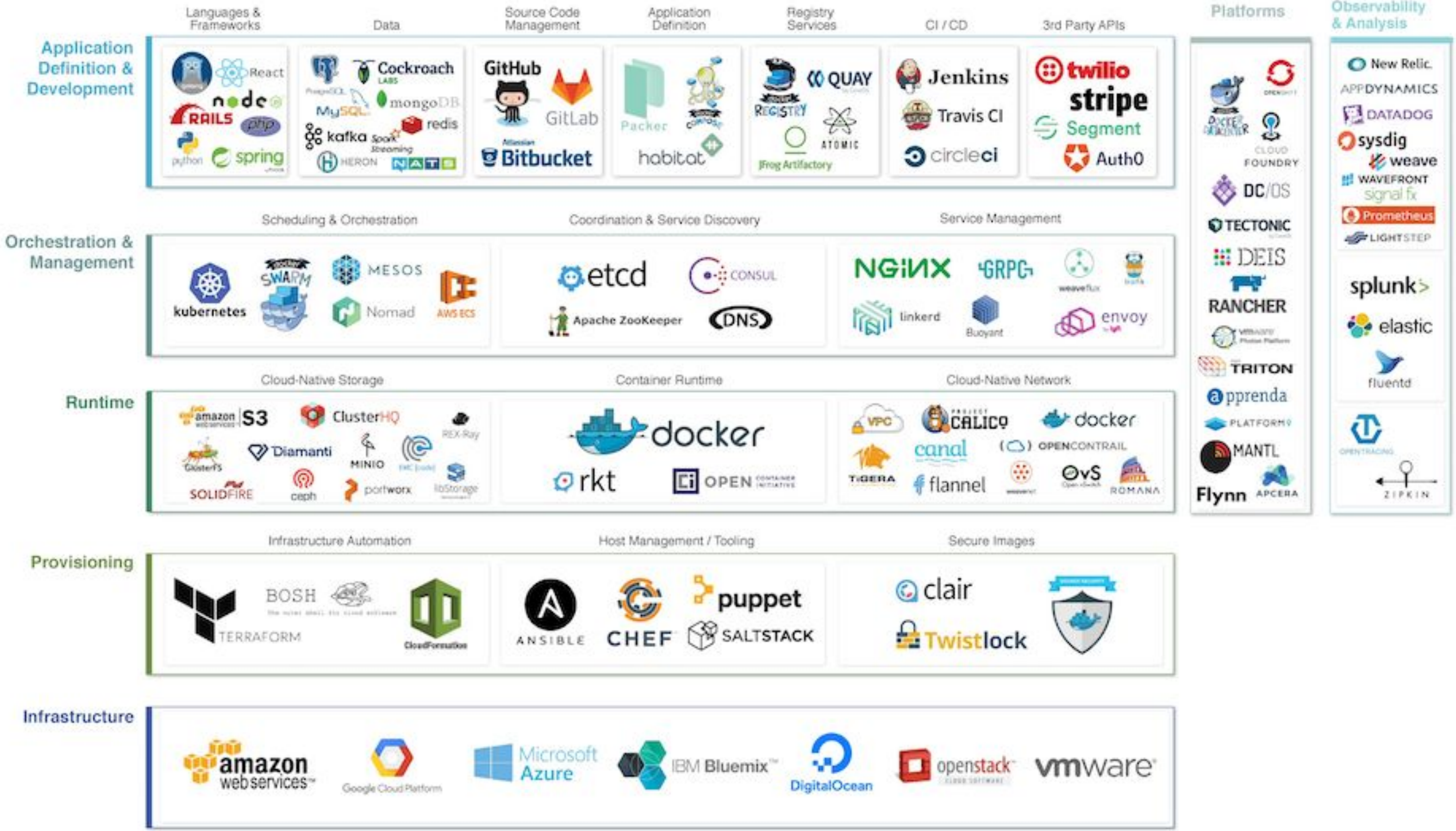
DevOps does not exist :)

DoEverything DoEt :)

- Deployment of release in production
- Standardization of development environment
- The ability to understand features of the developed application and configure the infrastructure for the normal functioning of the software
- Integration of development processes into delivery
- Detection and fix of various problems
- Setting up the environment for periodic changes
- Process automation

Know Everything

- Soft skills and communication
- Source control systems
- Continuous integration
- Infrastructure automation tools
- Clouds
- Security
- Testing
- Collaboration
- Big picture thinking


Application Definition & Development
Languages & Frameworks
React
node
RAILS
php
python
spring
Data
Cockroach LABS
PostgreSQL
MySQL
mongoDB
kafka
Spark Streaming
redis
HERON
NATS
Source Code Management
GitHub
GitLab
Atlassian Bitbucket
Application Definition
Packer
compose
habitat
Registry Services
REGISTRY
QUAY
ATOMIC
JFrog Artifactory
CI / CD
Jenkins
Travis CI
circleci
3rd Party APIs
twilio
stripe
Segment
Auth0
Orchestration & Management
Scheduling & Orchestration
kubernetes
SWARM
MESOS
Nomad
AWS ECS
Coordination & Service Discovery
etcd
CONSUL
Apache ZooKeeper
DNS
Service Management
NGINX
GRPC
weaveflux
linkerd
Buoyant
envoy
Runtime
Cloud-Native Storage
amazon web services S3
ClusterHQ
REX-Ray
GlusterFS
Diamanti
MINIO
EMC {code}
libStorage
SOLIDFIRE
ceph
portworx
Container Runtime
docker
rkt
OPEN CONTAINER INITIATIVE
Cloud-Native Network
VPC
PROJECT CALICO
docker
TIGERA
canal
OPENCONTRAIL
flannel
weavenet
OvS
ROMANA
Provisioning
Infrastructure Automation
TERRAFORM
BOSH
CloudFormation
Host Management / Tooling
ANSIBLE
CHEF
puppet
SALTSTACK
Secure Images
clair
Twistlock
Infrastructure
amazon web services
Google Cloud Platform
Microsoft Azure
IBM Bluemix
DigitalOcean
openstack
vmware
Platforms
DOCKER DATACENTER
OPENSHIFT
CLOUD FOUNDRY
DC/OS
TECTONIC
DEIS
RANCHER
vmware Photon Platform
TRITON
apprenda
PLATFORM9
MANTL
Flynn
APCERA
Observability & Analysis
New Relic
APPDYNAMICS
DATADOG
sysdig
weave
WAVEFRONT
signal fx
Prometheus
LIGHTSTEP
Monitoring
splunk>
elastic
fluentd
Logging
OPENTRACING
ZIPKIN
Tracing

What will you have

- Each day research
- Sleepless nights
- English
- Traveling experience ???
- Salary
    - https://salaries.dev.by/
    - https://www.glassdoor.nl/Salarissen/amsterdam-devops-engineer-salarissen-SRCH_IL.0,9_IM1112_KO10,25.htm?countryRedirect=true

Foundations of Continuous

- Continuous Integration
    - integrate -> test -> deployment
- Continuous Delivery
    - Continuous Integration -> release decision -> delivery
- Continuous Deployment
    - continuous deliver to any environment

Continuous Integration

Structure:

- VCS (Version control systems)
- Storages
    - Images, packages, credentials ...
- CI orchestrator
- Feedback tools

Steps (automated and infinitive looped) :

Push code -> CI orchestrator - > Pipeline -> Release

Continuous Integration. VCS

Everything under control:

- All see what is all doing (commits history)
- Access control (protect branches/PR)
- Approving control (who is code reviewers)

Most popular and famous platforms:

- github
- gitlab
- bitbucket

Continuous Integration. Storages

Binaries files:

- Central storages (NFS, S3, SMB, SFTP ...)
- Packages (yum/apt repos, NEXUS ...)
- Images (dockerhub, vagrant cloud ...)

CI/CD code:

- VCS

Continuous Integration. Orchestrator

- Jenkins
- Bambo (Atlassian apps)
- TeamCity
- Gitlab/Github

Docker/k8s:

- Kubeapplier
- Flux CD
- Spinnaker
- Jenkins X

Continuous Integration. Feedback tools

- Email
- Slack/flowdock/telegram
- Ticket system (Jira)
- Wiki

Continuous Integration

Requirements:

- All code inside the VCS
  - Include code of CI/CD pipeline
- Collective work
  - standups, synks, meetings, ticket system, shared documentation
- Pipeline should work everywhere, outside of local sandboxes
- All steps are automated

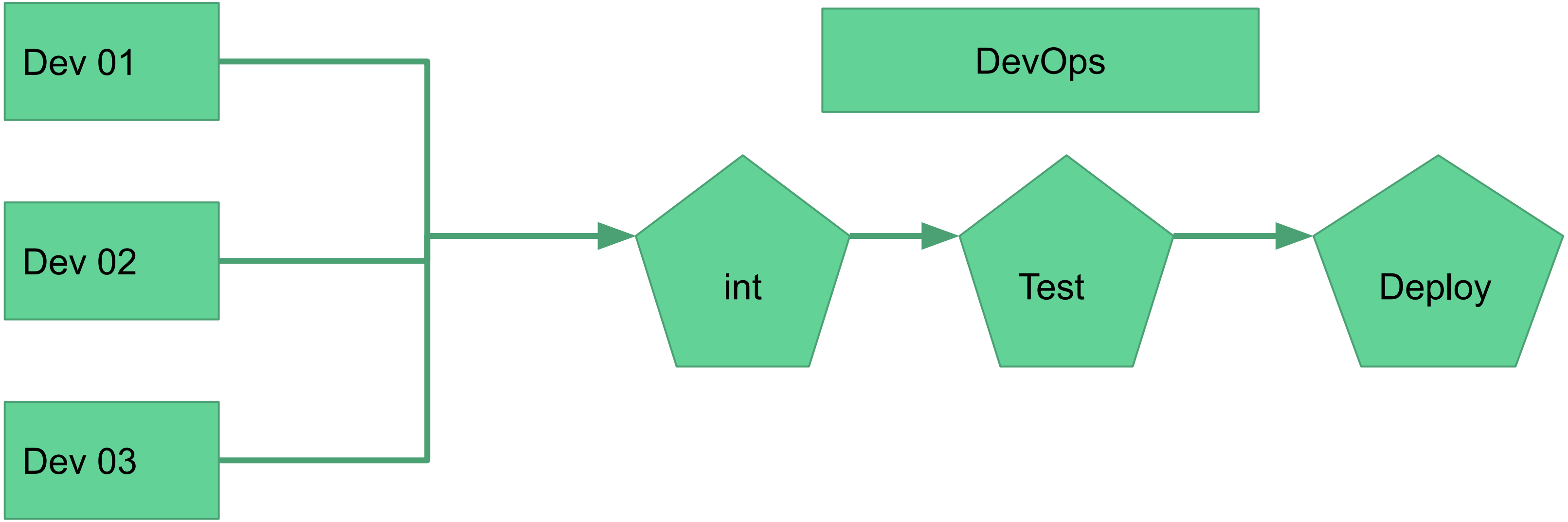
Continuous Delivery

- CI+ deployment into production environment
- Should have human trigger
- All automated, except previous point

Infrastructure providers

- Bare metal
  - Any kind of virtualization, k8s, openstack ...
- Remote cloud
  - AWS, Google Cloud, Microsoft Azure ...

Deployment types. Old way

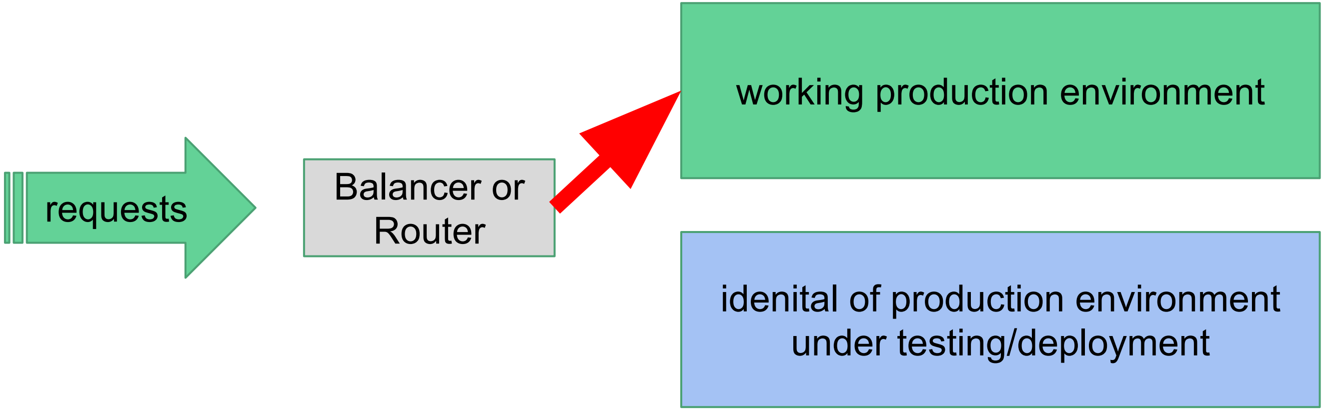


Deployment types

- Colored:
  - Green/Blue
  - Greenfield/Brownfield
- Canary
- Rolling deployment
- etc...

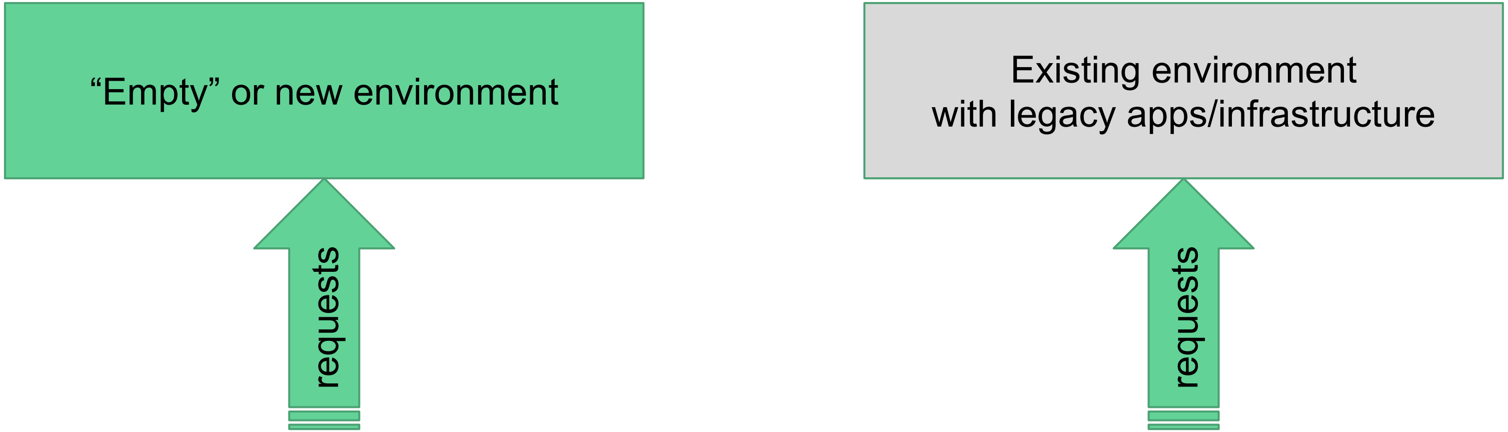
Deployment types. Green/Blue

- Two identical production environments
  - Green - “working” environment
  - Blue - “staging” environment


requests
Balancer or Router
working production environment
idenital of production environment under testing/deployment

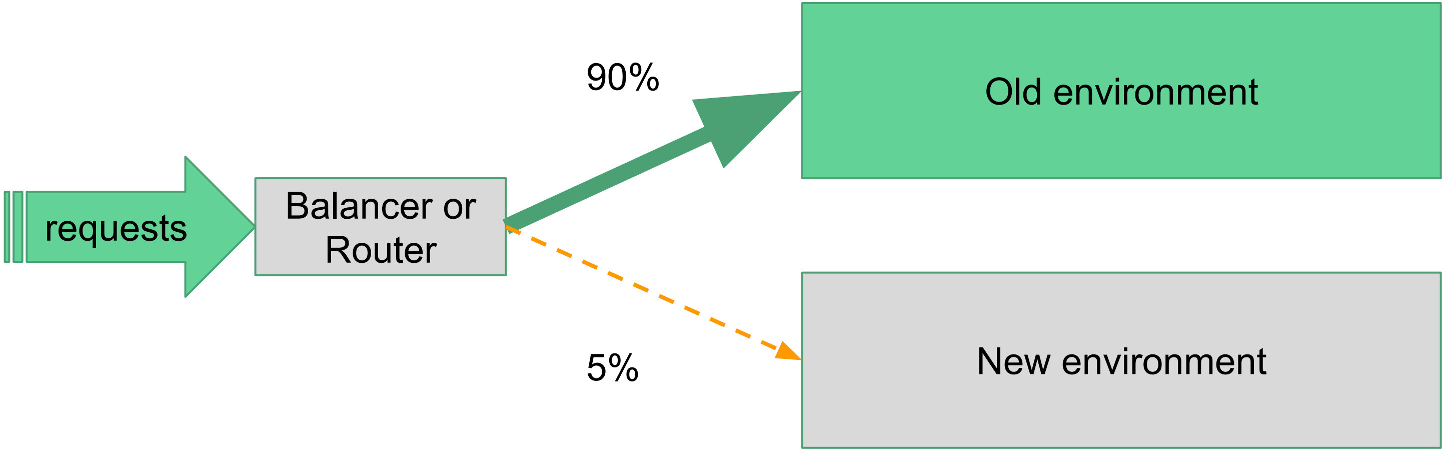
Deployment types. Greenfield/Brownfield

- Greenfield - new deployment
- Brownfield - deployment/upgrade on existing environment



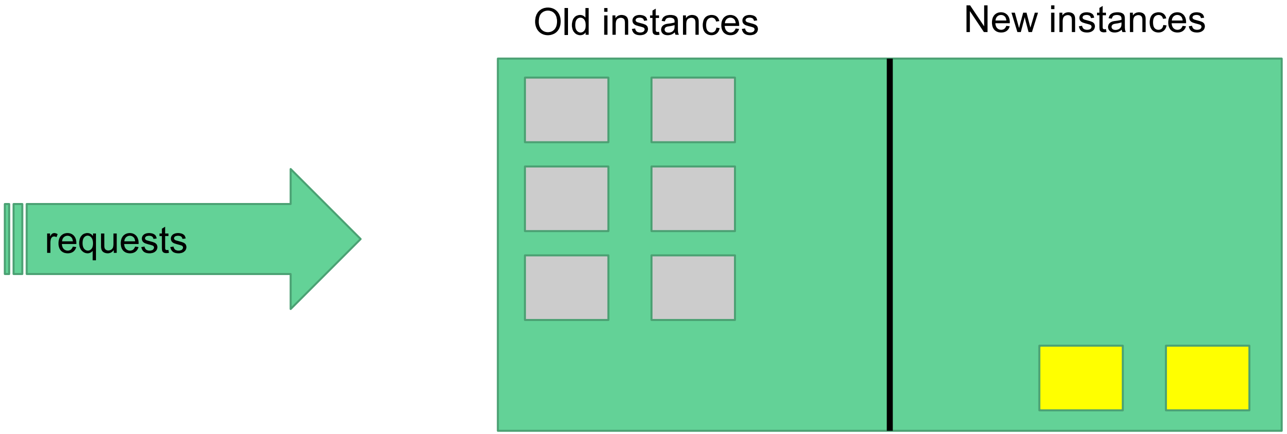
Deployment types. Canary

- Deploy new version
- Smooth flowing of the users to new environment



Deployment types. Rolling

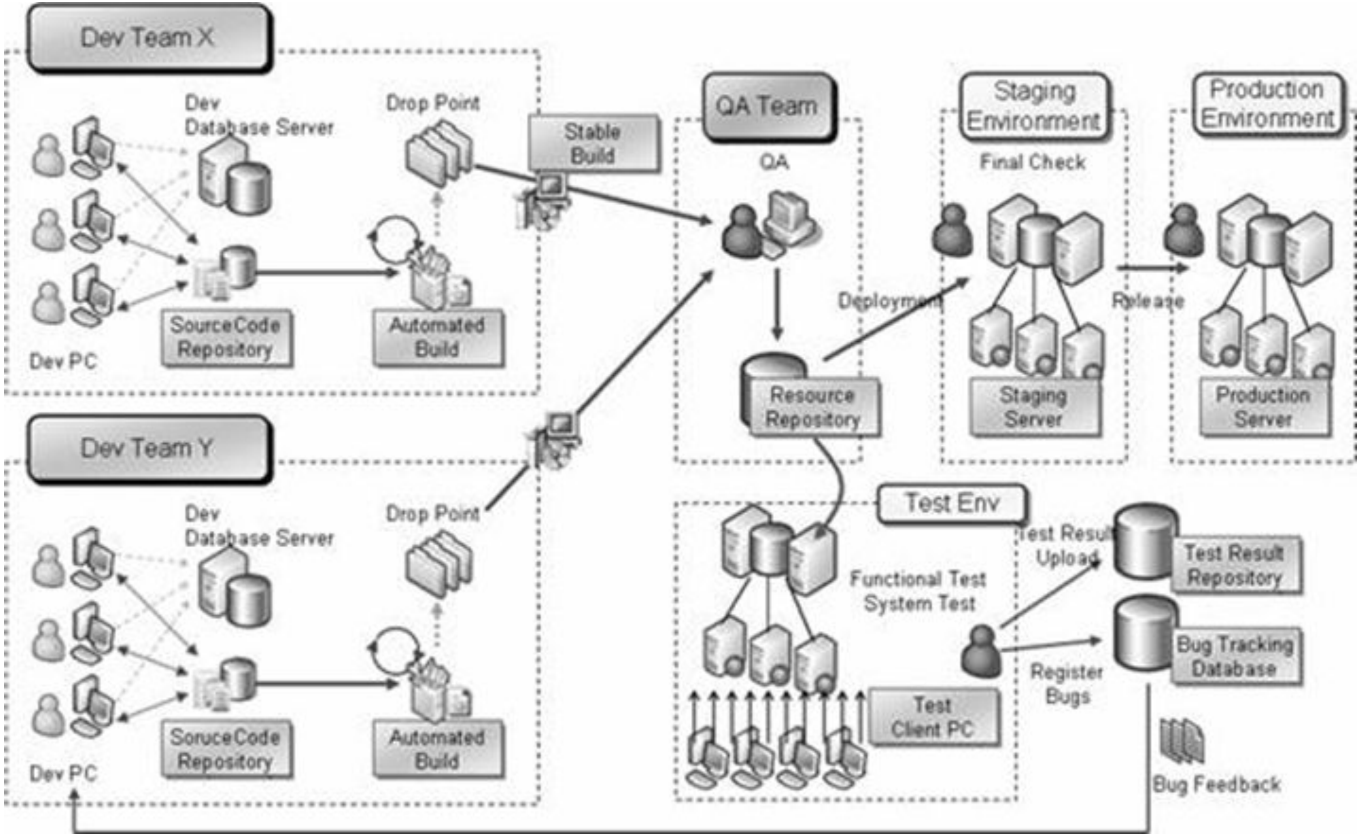
- Rolling new instances/images one by one



Environment structure

- Development
    - Local servers
- QA
    - Testing
- Integration
    - Lab/modules
- Production
    - Pre-production
    - Blue deployment
    - etc…

Environment structure



Home task

Registration on slack and join to the chanel

- https://github.com/
- https://gitlab.com/users/sign_in
- https://bitbucket.org
- https://hub.docker.com/
- https://app.vagrantup.com/boxes/search

Local installation:

- GIT: https://git-scm.com/downloads
- VirtualBox: https://www.virtualbox.org/wiki/Downloads
- Vagrant: https://www.vagrantup.com/downloads.html
- Visual Studio Code: https://code.visualstudio.com/download