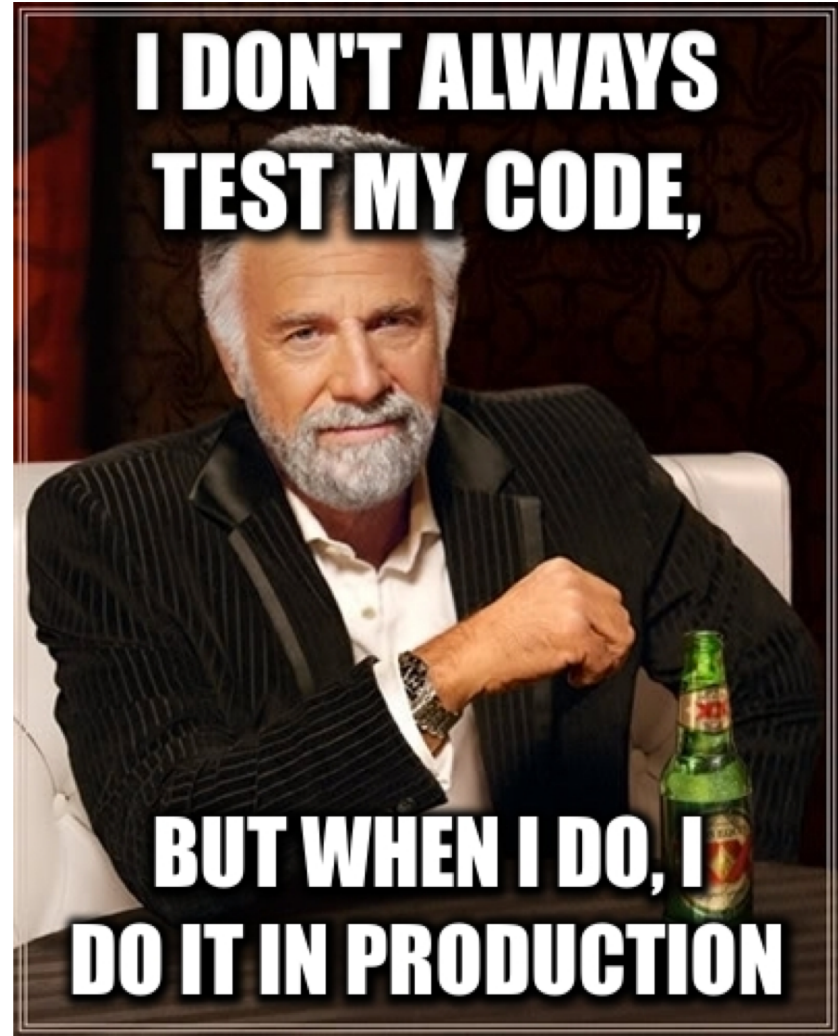


Network change management and continuous integration in the age of NetDevOps

MATYÁŠ PROKOP



Agenda

- Blah Blah Blah Infrastructure as Code Blah Blah Blah
- Live Demo

What is the Infrastructure as Code?

Infrastructure as code (IaC) is the process of managing and provisioning computer data centers (or campus – *Matyas 10/2019*) through machine-readable definition files, rather than physical hardware configuration or interactive configuration tools.

Source: Wiki

Process inspired by modern software development practices

- Version control provides higher quality and change control tracking
- Automatic testing and deployment
- Fast deployment
- Well established process
- You can put fancy stickers on your laptop

Infrastructure as code

- Change and source control
 - GitLab, GitHub, BitBucket
- Automation tools
 - Ansible, Terraform, Chef, Puppet
- APIs
 - Lots of APIs!
- Staging or testing environment
- Continuous Integration and Delivery (CI/CD) Pipeline
 - GitLab CI/CD, Jenkins

Change and source control



- GitLab will be your single source of truth
- Storing any configuration preferably in format like YAML
- Plan, code, test and deploy the code from single repository
- CI/CD Pipeline integration
- Alternatives: GitHub, BitBucket

Automation tools



ANSIBLE

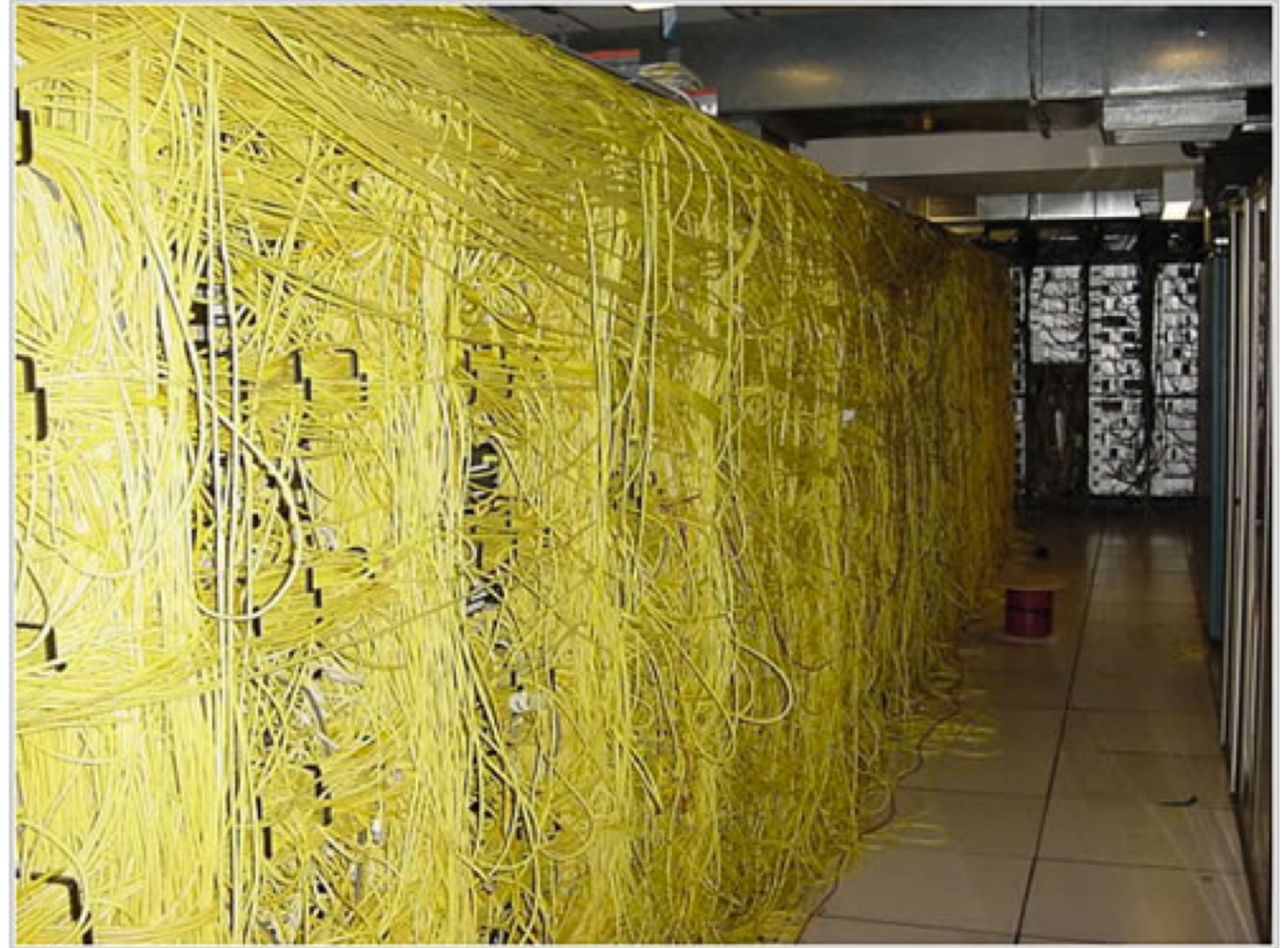
- Ansible will allow consistent configuration
- Ansible modules provides more flexibility
- Ansible is agentless
- Strong community

Rest APIs



- Using of Ansible is preferred but APIs are sometimes needed
- Calling APIs via Python scripts

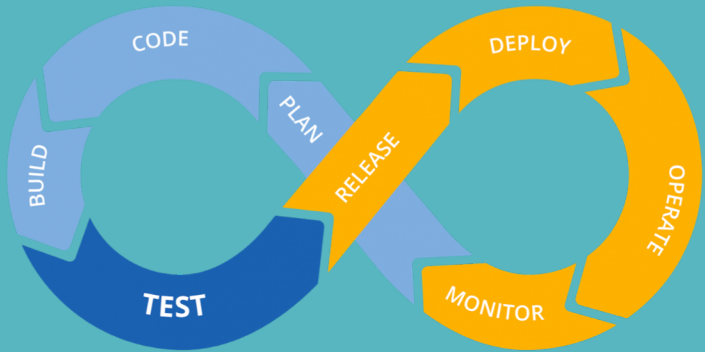
Staging environment



Staging environment

- Hardware staging environment
- Virtual staging environment
 - GNS3, VIRL, Eve-ng
 - Vagrant
- Staging or testing tenant environment
 - ACI Tenant

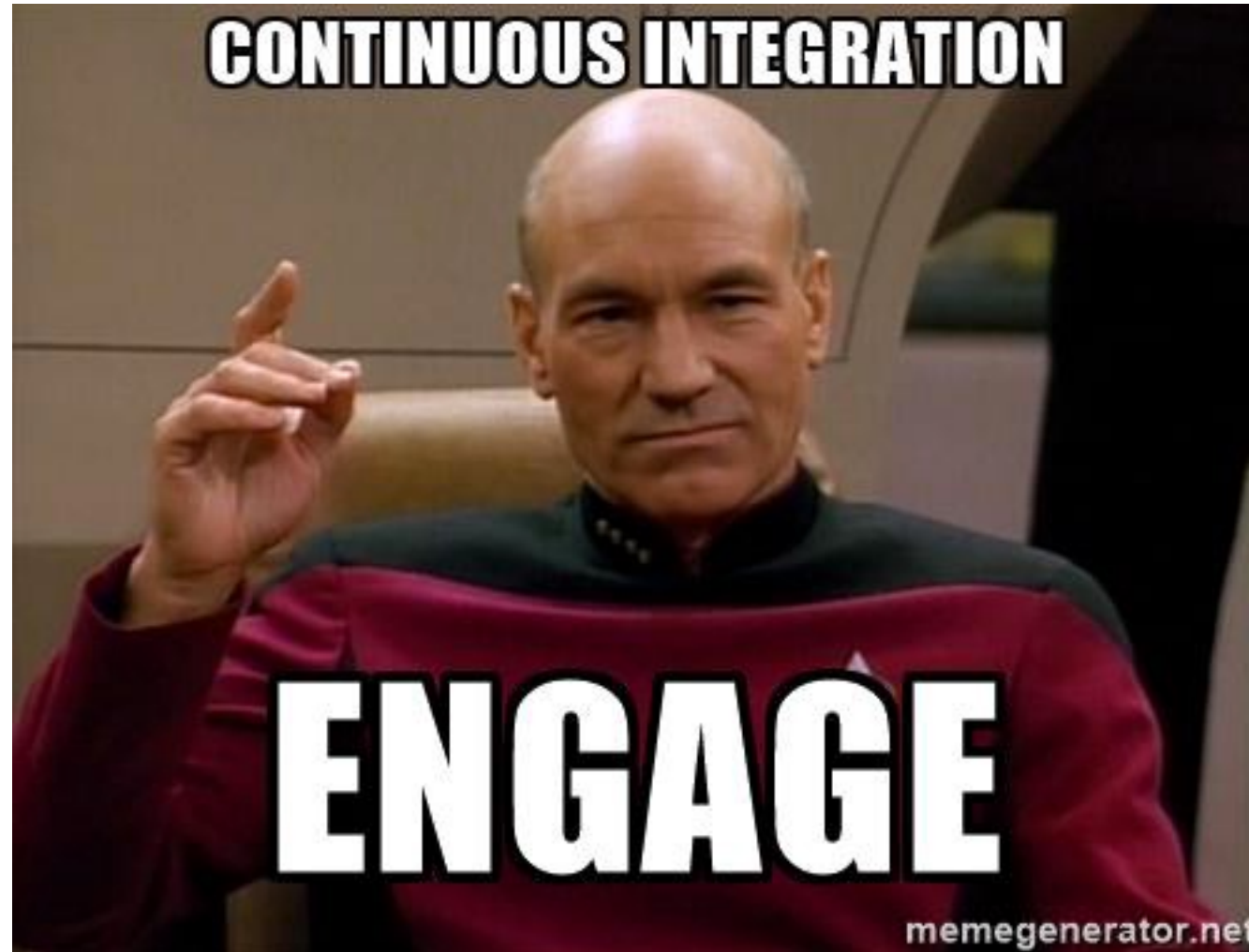
CI/CD Pipeline



- Essential part
- Only way to merge any changes into repository
- Increase the speed and quality of changes
- Pushing the configuration from the repository to the staging and production environment



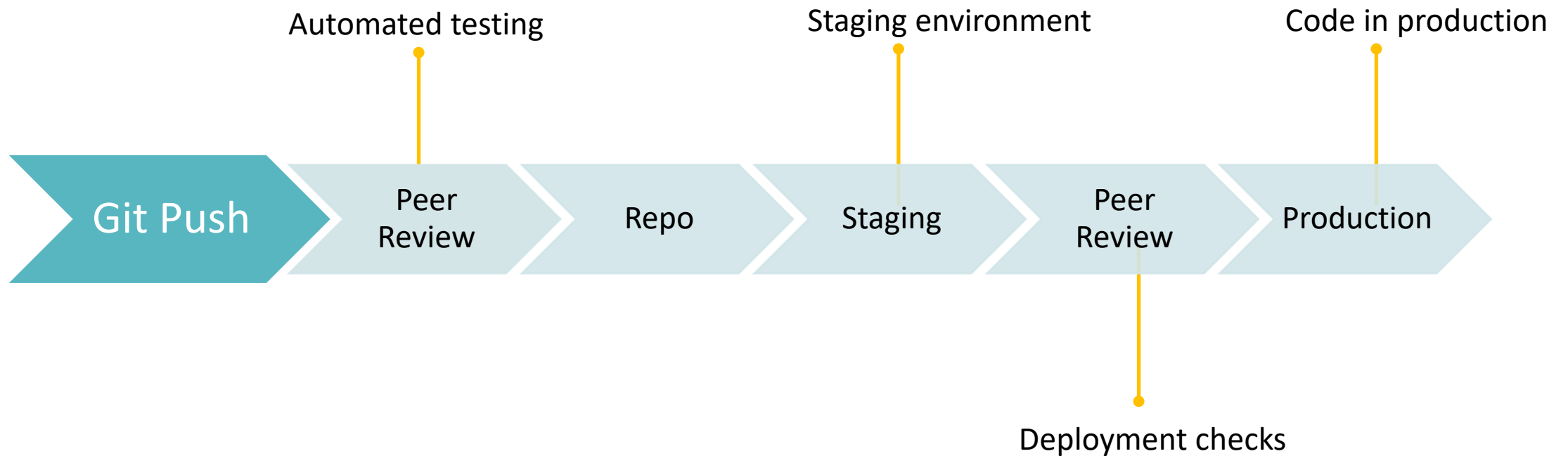
CONTINUOUS INTEGRATION



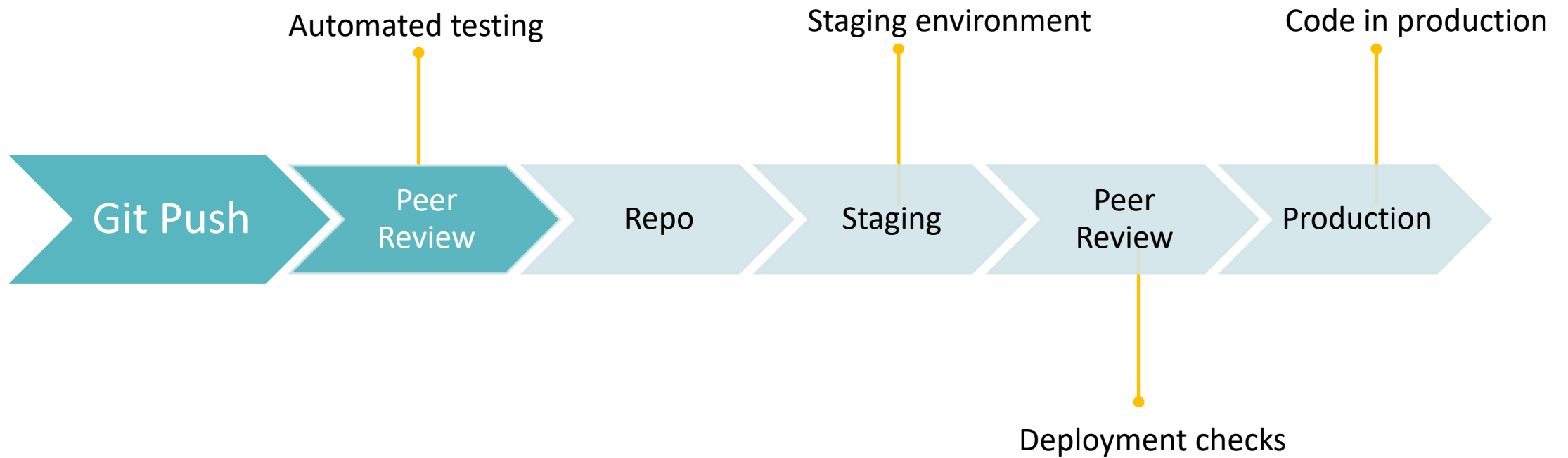
ENGAGE

memegenerator.net

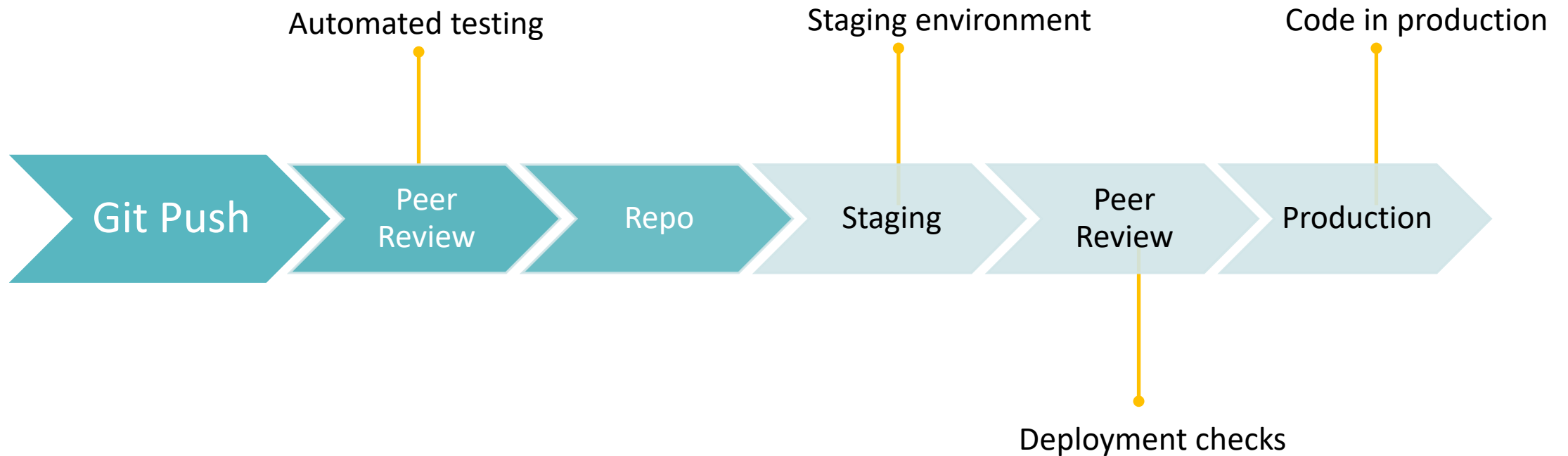
Pushing the code to the repository



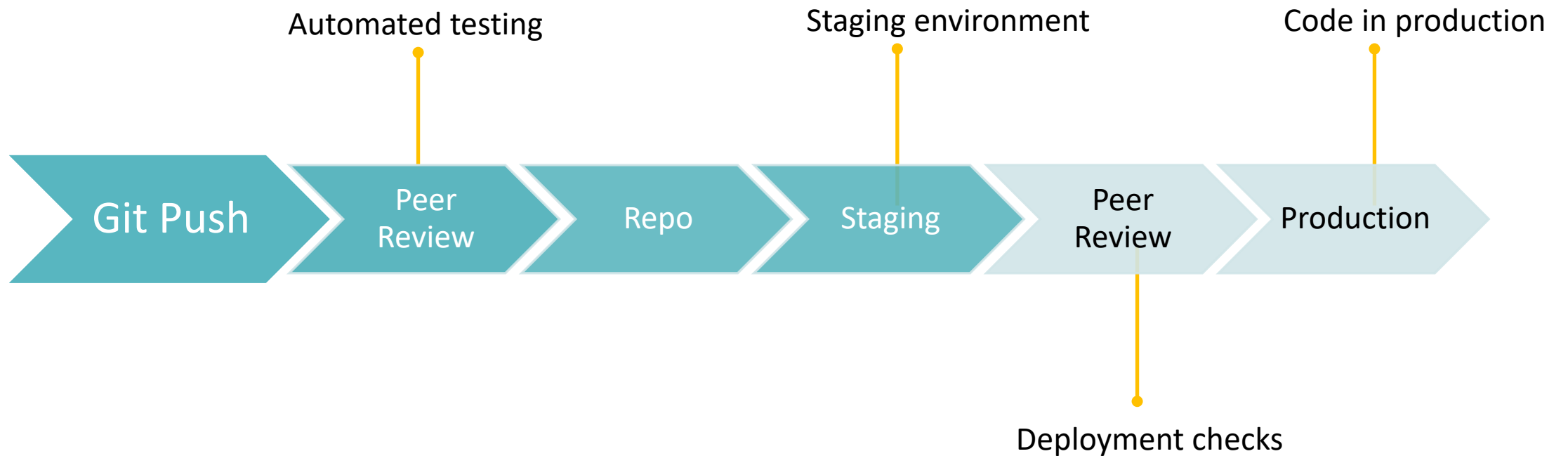
Code validation



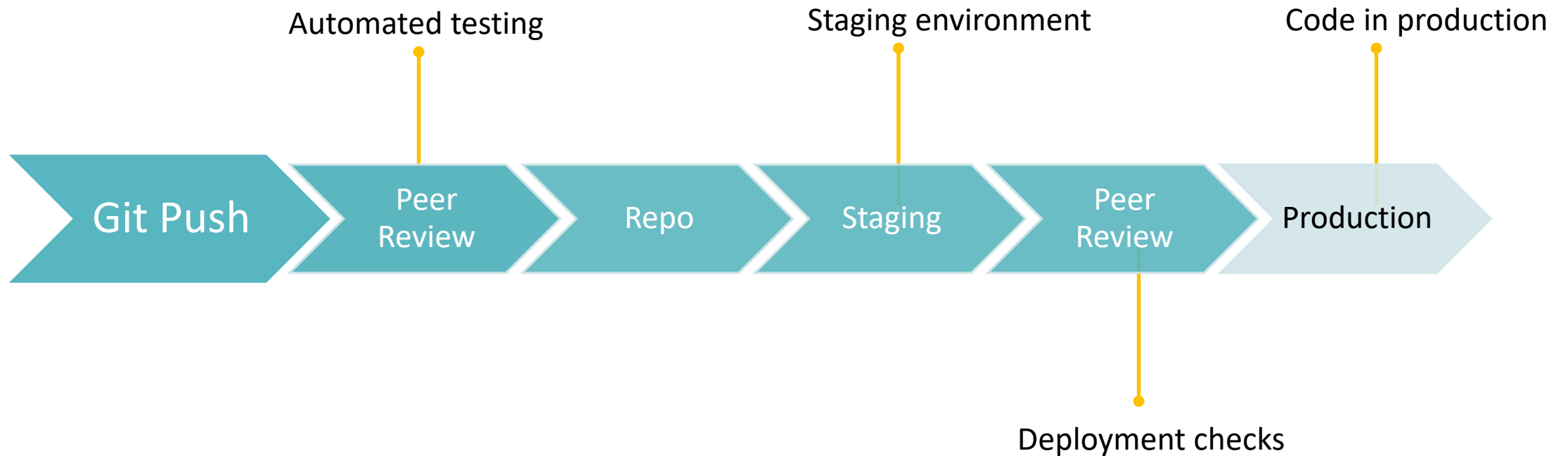
Code is stored in the repository



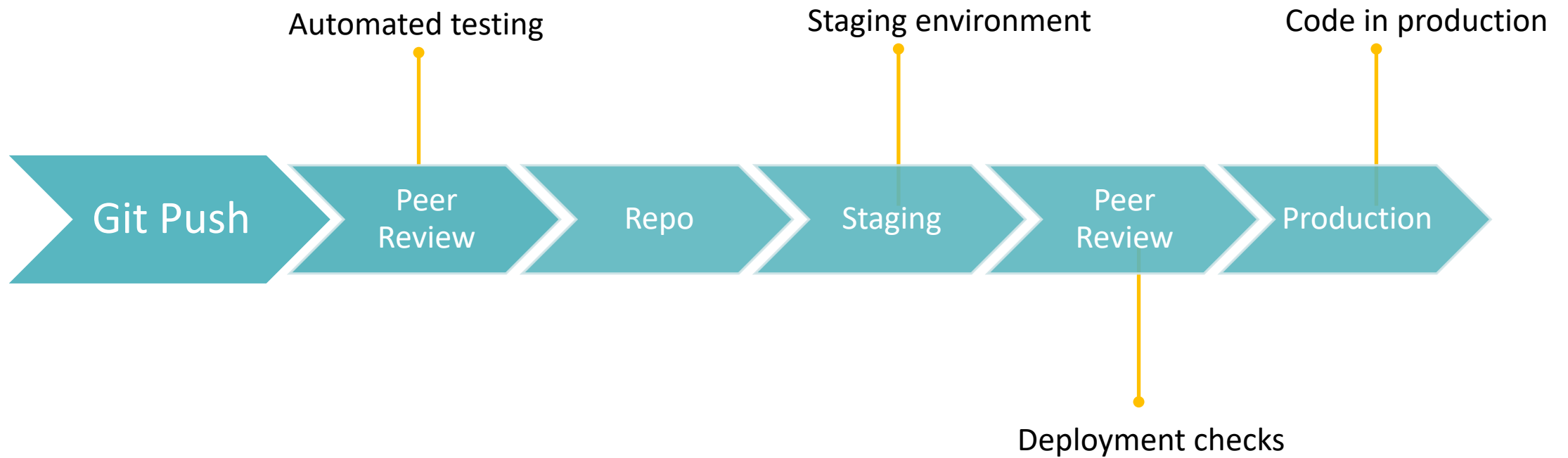
Running the code against the staging environment



Deploying the code in the production



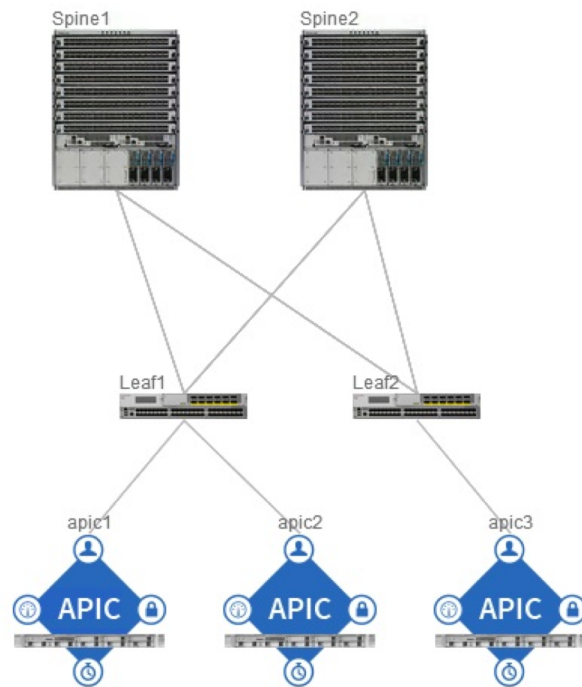
Code deployed



Live demo



Live demo



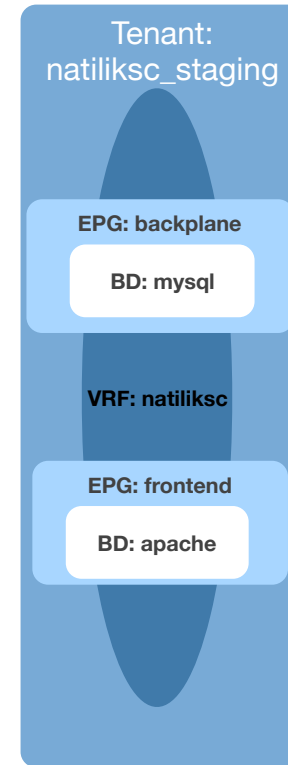
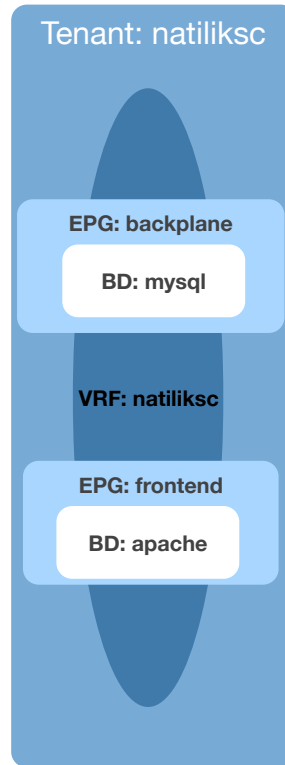
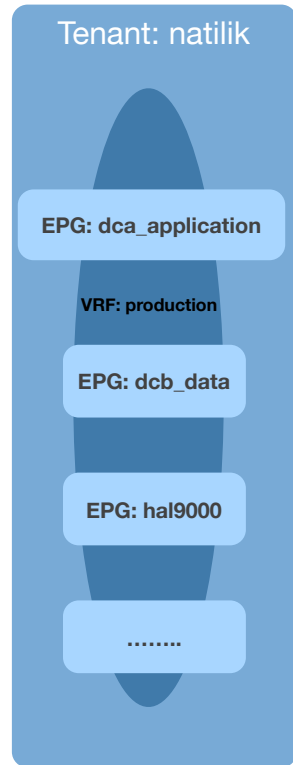
+



+



Live demo – ACI Logical Design



Legend

Tenant = virtual environment

EPG = End Point Group

VRF = Virtual Route Forwarding

Natilik ACI

Stay in touch

Matyas Prokop



matyas.prokop@gmail.com
email@matyasprokop.com



@maty0609



<https://git.matyasprokop.com>