

How to Build a CI/CD Pipeline

Mohamed Radwan
DevOps Engineer

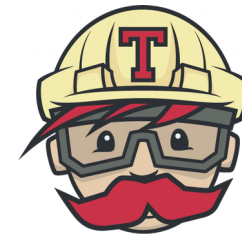
Overview

- Introduction to Gitlab
- Gitlab CI Configuration file
- (CI/CD) Pipeline
- Immutable AWS Deployments
- Creating Image
- Testing Image
- Configuration Management Tools
- Deploy Image
- Demo

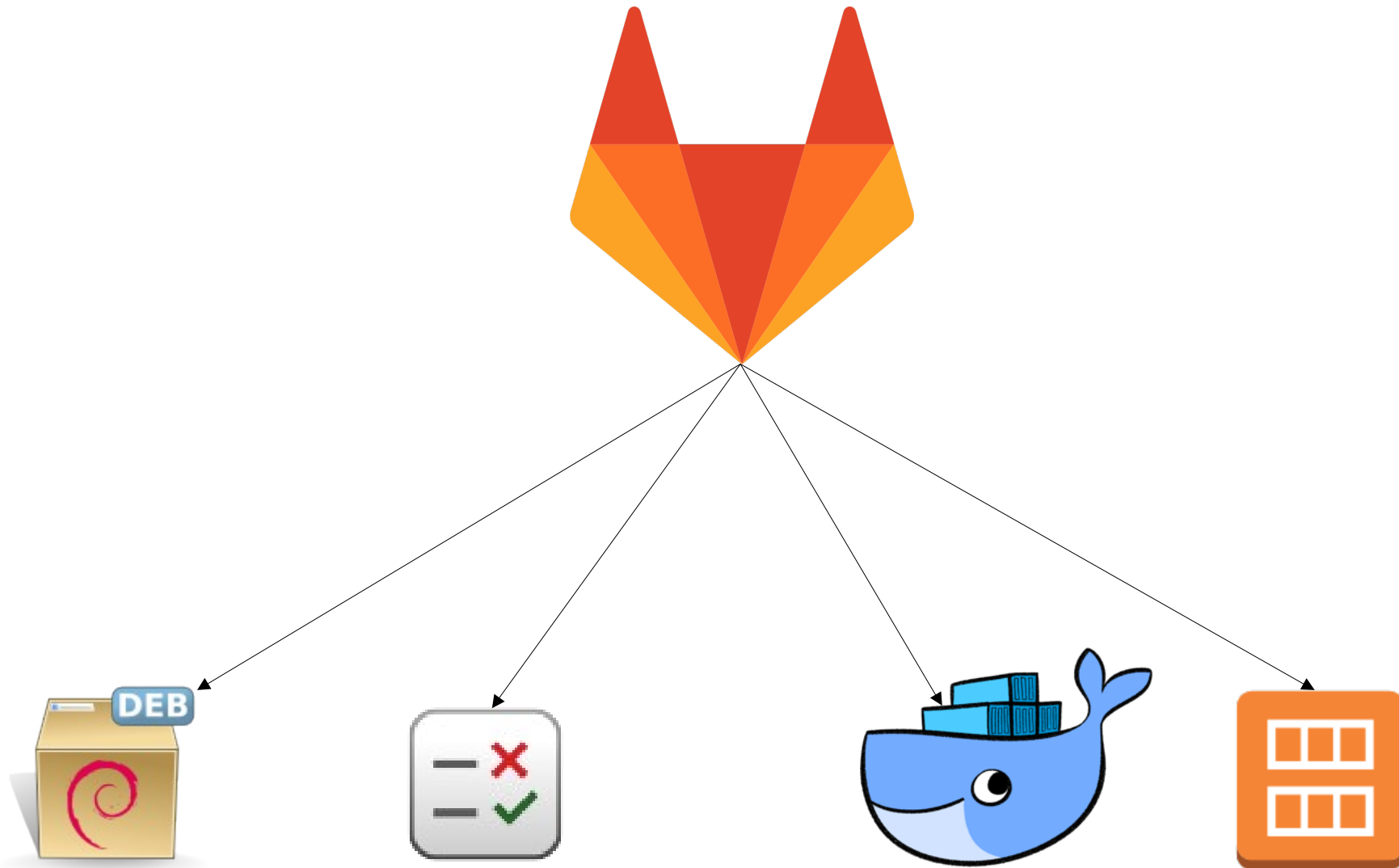


Introduction to Gitlab

- A git-based code like github.
- A (CI/CD) platform like Jenkins , Travis and others.
- Gitlab components:
 - 1- Gitlab server : hosting service, CI build system management
 - 2- Gitlab runners: User-space that execute builds.
 - Highly configurable you can have multiple runners per repo.
 - It can run anywhere: laptop , AWS ..etc



Introduction to Gitlab

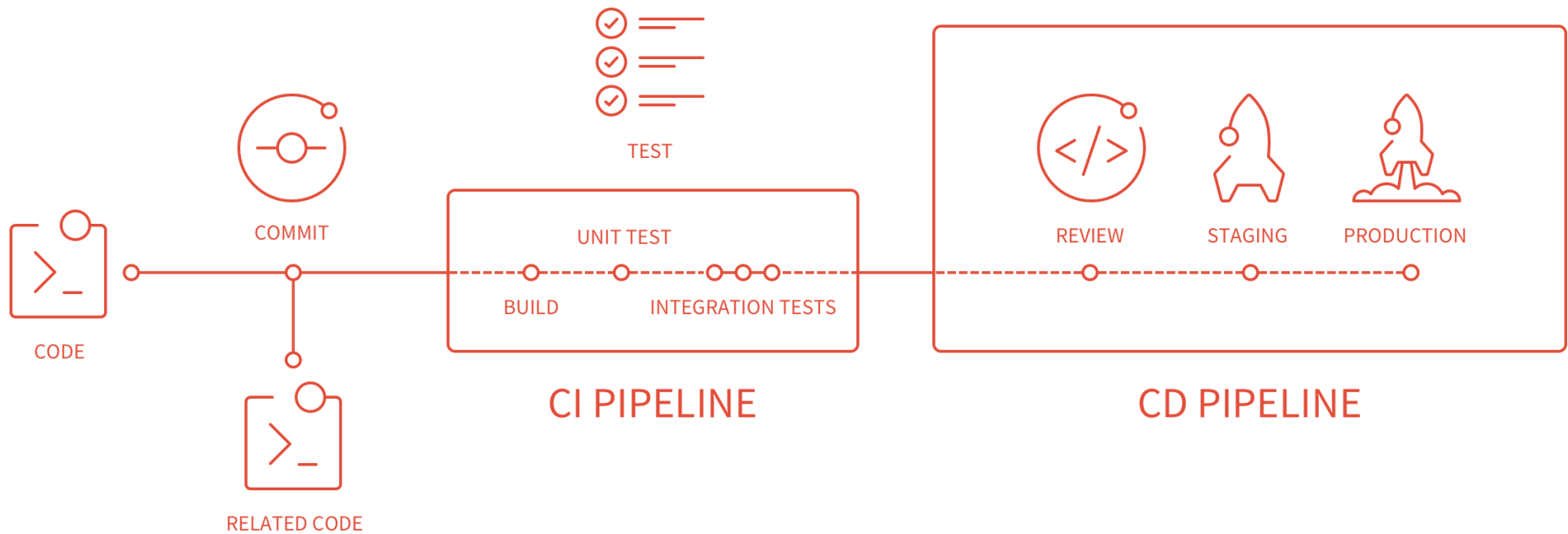


Gitlab CI configuration file

- Standard YAML
 - Very human-friendly
 - .gitlab-ci.yml, in the top directory of your git repository
 - Describes pipelines which consist of stages
 - Each stage has a specific function: build, test, deploy...
 - Each stage can have its own tags.
 - Each stage can produce artifacts re-use from other stages
 - Stages can run in parallel

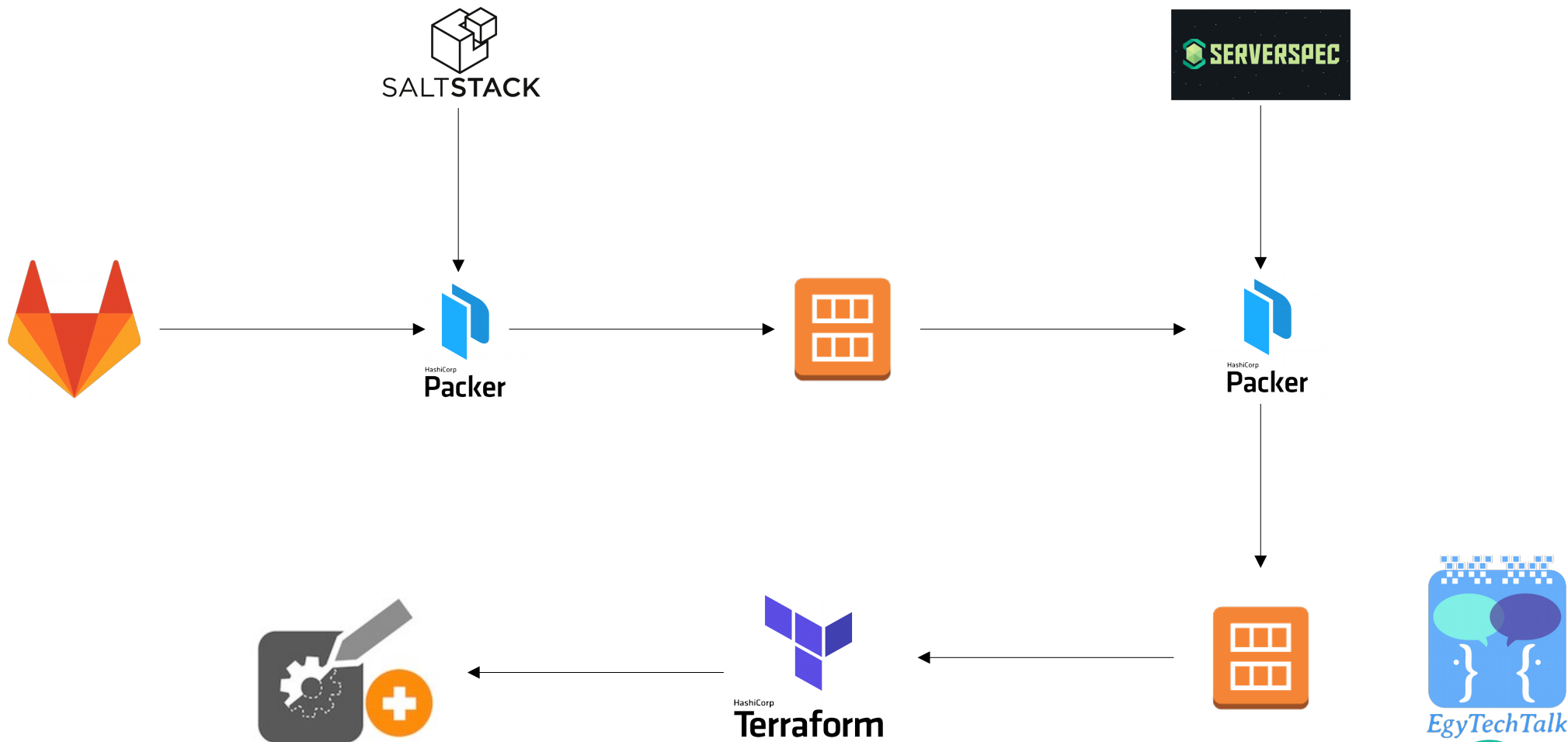


(CI/CD) Pipeline



Immutable AWS Deployments

- Automate the setup and deployment for every part and every layer of your infrastructure.
- Never change any part of your system once it is deployed. If you need to change it, deploy a new system.



Creating Image

- Packer is an open source tool for creating images for multiple platforms from a single source configuration.
- Packer does not replace configuration management like Chef , Puppet, SaltStack and Ansible .
- Packer is able to use tools like Chef or Puppet to install software onto the image.
- Machine image formats examples: Ami for EC2, VMDK/VMX files for VMware, OVF exports for VirtualBox



Testing Image

- Serverspec is the name of a Ruby tool which allows you to write simple tests, to validate that a server is correctly configured
- With Serverspec, you can write RSpec tests for checking your servers are configured correctly.

Resource Types:

bond | bridge | cgroup | command | cron | default_gateway | docker_container | docker_image | file | group | host | interface | ip6tables | ipfilter | ipnat | iptables | kernel_module | linux_audit_system | linux_kernel_parameter | lxc | mail_alias | mysql_config | package | php_config | port | ppa | process | routing_table | selinux | selinux_module | service | user | x509_certificate | x509_private_key

```
describe package('nginx') do
  it { should be_installed }
end
```

```
describe port(80) do
  it { should be_listening }
end
```

```
describe cron do
  it { should have_entry '* * * * * /usr/local/bin/foo' }
end
```



Configuration Management Tools

- Puppet, Chef, Ansible and SaltStack present different paths to achieve a common goal of managing large-scale server infrastructure .
- Configuration Management capabilities offers the following :
 - Orchestration
 - Automated provisioning
 - Configuration automation
 - Code management
 - Reporting



Deploy Image

- Terraform is a tool for building, changing, and versioning infrastructure safely and efficiently.
- Infrastructure as Code : can be shared and re-used.
- Change Automation : Complex changesets can be applied to your infrastructure with minimal human interaction.



DEMO

- Build AMI using Packer and SaltStack
- Testing AMI by Using serverspec and packer
- Deploy by using Terraform.

Q & A