LIGATUS

From On-Premise to Containers

Our CI/CD Journey into the Cloud

Teodor Stoev

Senior Software Developer

Email: teodor.stoev@ligatus.com

- **Solid experience with Java and web technologies**
- **Focus on**
  - Continuous Delivery
  - Innovation in software architecture
  - Research, evaluation and integration of new software technologies

Agenda

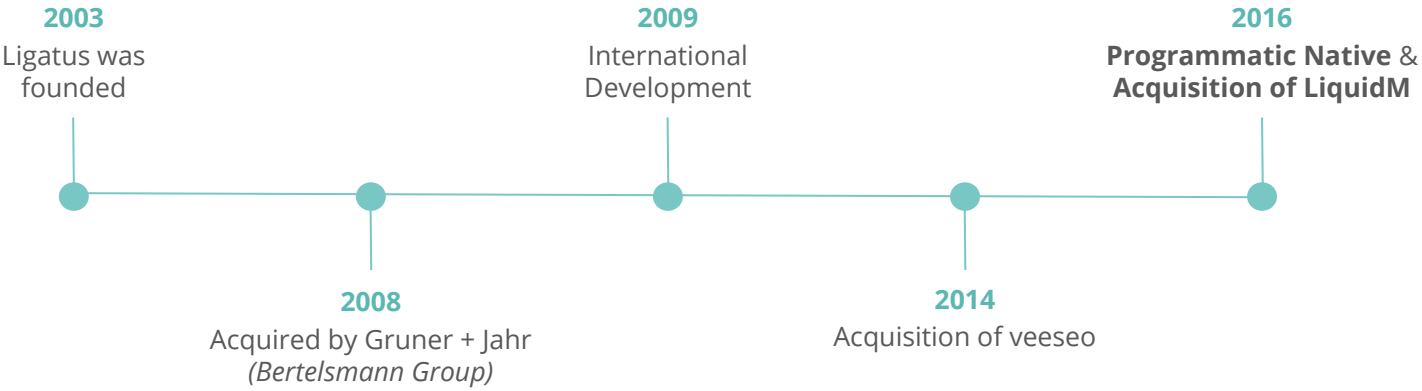
Who we are

What we left behind

Where we currently are

What is still ahead of us

Ligatus, Europe's leading Native Advertising and Performance Marketing Network



Ligatus, Europe's leading Native Advertising and Performance Marketing Network

FOR PUBLISHERS:

A full revenue and service solution for premium publishers:

- Content Recommendation
- Anti-Adblock Tools
- Traffic Exchange Solutions (exchange traffic between publishers)

FOR ADVERTISERS:

Tailored Native Advertising solutions to help advertisers achieve their marketing goals:

- **Content Promotion:** for awareness & engagement objectives
- **Direct Response:** for acquisition objectives

OUR NETWORK:

- Premium selected media publishers
- Multi-device
- International Reach: **9 countries**
  - **+31 billion** ad impressions
  - **+1200 premium** partners
- Programmatic Native

Agenda

Who we are

What we left behind

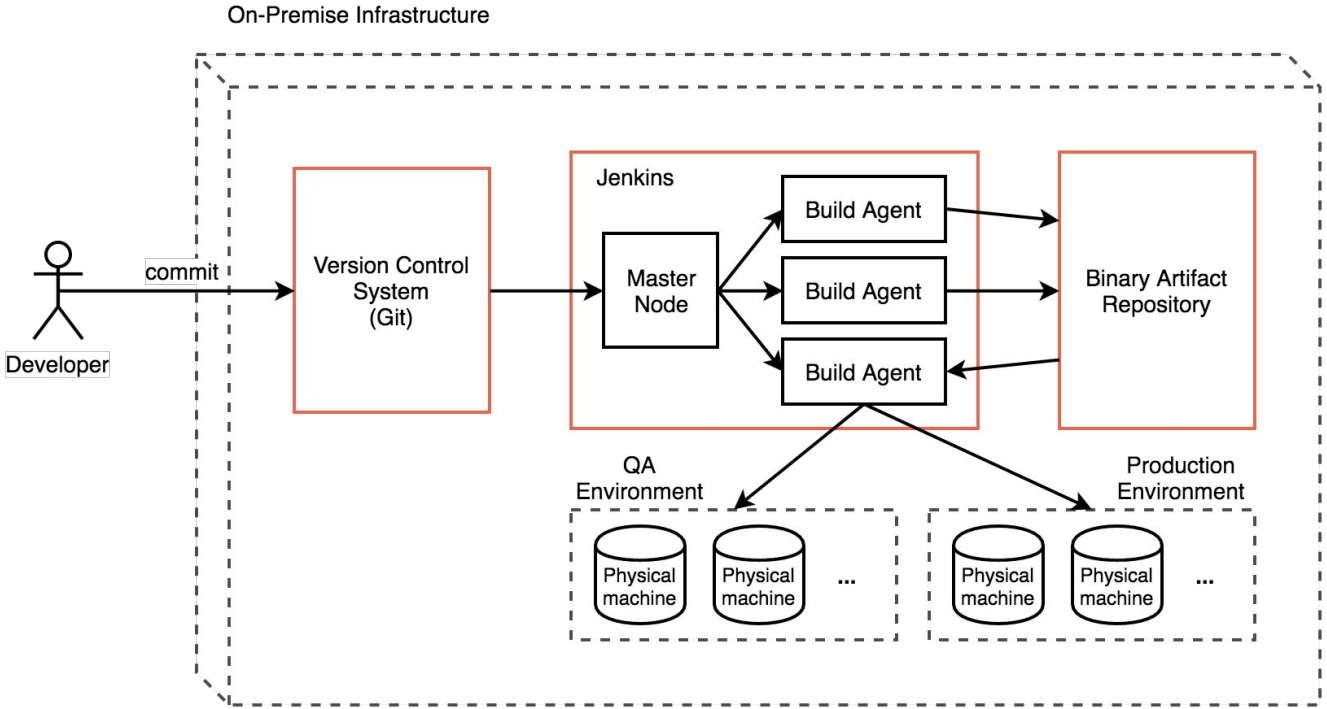
Where we currently are

What is still ahead of us

The old world

- **On-premise infrastructure for**
  - Build and deployment system
  - QA and Production system
- **Deployment on self-managed physical machines**
- **Classical CI server: Jenkins**
  - Hard to maintain: many interconnected dependencies
  - Hard to scale for distributed teams with multiple projects
  - Build agents are reused for the next job in the line

The old world

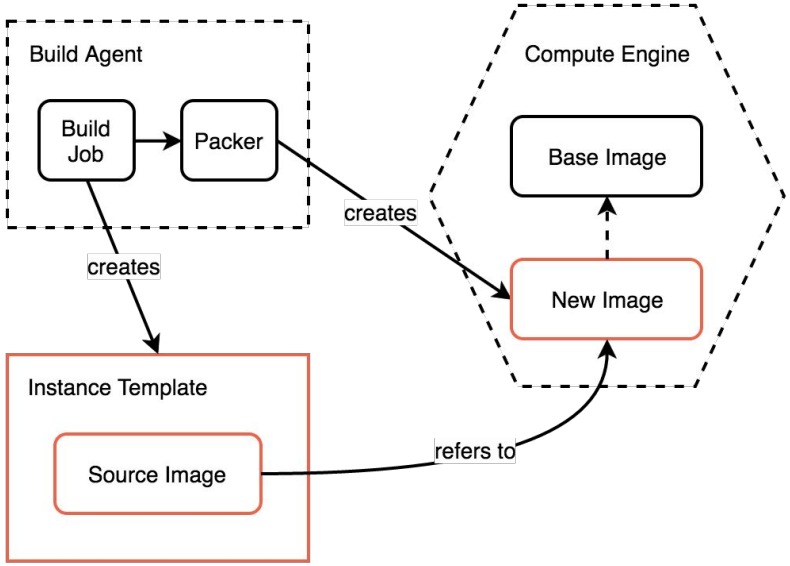


Why not good enough?

- **Jenkins only designed for CI, not for CD**
  - No native support for CD - (buggy) plugins necessary
  - No native support for autoscaling build agents - static ones cannot handle the load that CD puts on them
- **Manual updates for every CI component**
- **Lack of flexibility to set up new hardware or reuse old one**
  - Any change requires significant manual effort
  - Cannot autoscale to meet dynamic requirements or save costs

First step: Deployment in the Cloud

- **A new GCE Image built on each pipeline run**
- **The base image**
  - runs a chosen OS and a puppet agent
  - is regularly updated by a separate pipeline
- **A new instance template replaces the old one**



Next step: Redesign CI

- **Start from scratch!**
  - Give up on-premise
  - Give up Jenkins
  - Do everything in the cloud
- **The investment**
  - 4 software engineers
  - 2 months effort
  - 1 room

Mob programming

Source: http://team-coder.com/mob-programming/

Agenda

Who we are

What we left behind

Where we currently are

What is still ahead of us

The goal we set

- **Host the entire system in the Google Cloud**
  - Use Google OAuth instead of self-managed credentials
  - Do not use any VPN (easy access to the system)
- **Fully automate**
  - provisioning and scaling of the CI environment
  - updates of all software components
  - replacement of certificates and keys
  - backup and recovery process

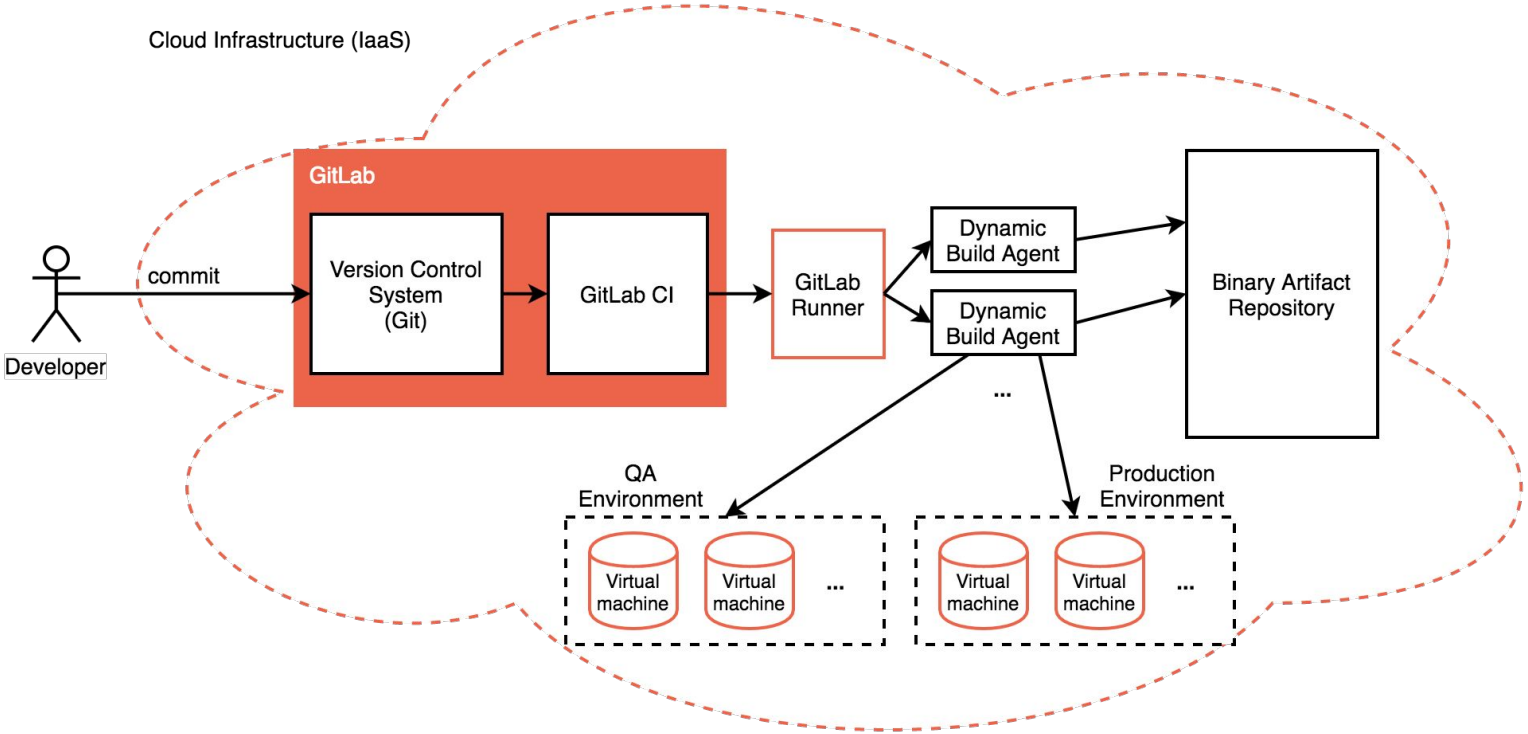
The equipment of choice: GitLab

- **Features**
  - Git repository management
  - GitLab CI for CI and CD
  - In-built Docker registry for build artifacts
  - Code reviews, issue tracking, activity feeds, wikis
- **Open Source (MIT license)**
- **Can be used as SaaS or self-hosted**
- **Highly scalable**

The new world

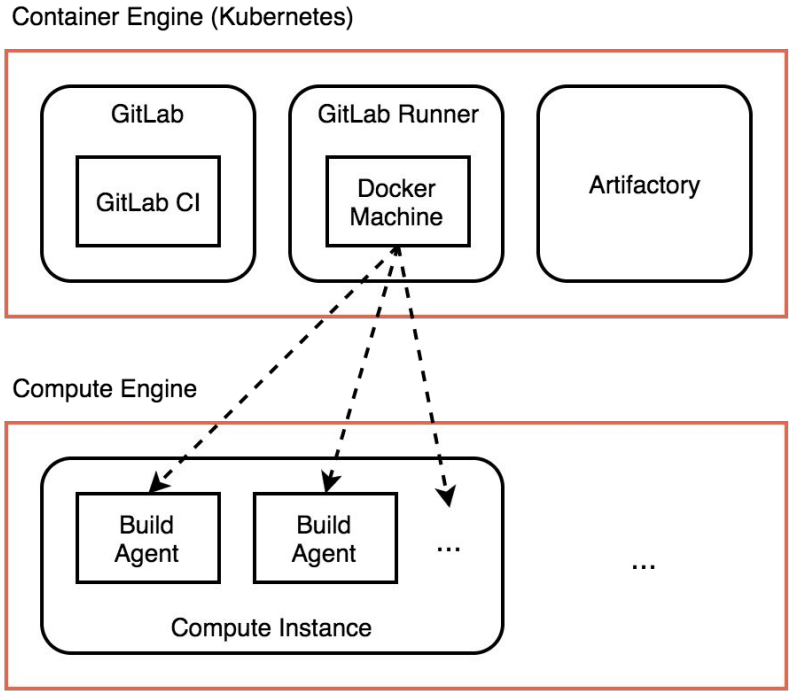
- **Cloud infrastructure (IaaS) for both**
  - Build and deployment system
  - QA and Production system
- **Deployment on cloud-based virtual machines**
- **Gitlab CI**
  - Fully integrated with the version control system (Git)
  - Native support for CD: simple configuration as a YAML script
  - Autoscaling build agents as containers with Docker machine

The new world



Under the hood

- **Google-managed Kubernetes cluster**
- **Main CI components deployed on Kubernetes**
- **Autoscaling build agents**
  - Build agents are short-lived Docker containers
  - Compute instances are started on-demand



Infrastructure as code

Google Deployment Manager

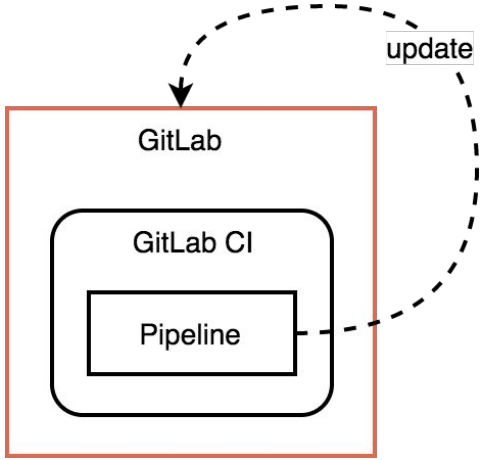
```
resources:
- name: k8s-dev-build-cluster
  type: container.v1.cluster
  metadata:
    dependsOn:
    - gitlab-conf-pd
    - gitlab-data-pd
    ...
  properties:
    zone: europe-west1-b
    cluster:
      name: k8s-dev-build-cluster
      initialNodeCount: 1
      nodeConfig:
        machine_type: n1-standard-4
        oauthScopes:
        - https://www.googleapis.com/auth/compute
        ...
```

Kubernetes configuration file

```
apiVersion: extensions/v1beta1
kind: Deployment
metadata:
  name: gitlab
  ...
spec:
  replicas: 1
  template:
    metadata:
      ...
    spec:
      containers:
      - name: gitlab-ce-container
        image: eu.gcr.io/ligatus/gitlab-ce:latest
        volumeMounts:
        - mountPath: /etc/gitlab
          name: gitlab-conf-pd
        ...
```

Software updates

- **Software updates per CD pipeline**
  - One pipeline per software component
  - Triggered automatically every day
  - Only a couple of minutes downtime
- **Backup and restore**
  - All data on persistent disks in GCE
  - Snapshot creation is part of the pipeline
  - A single script restores the entire system



Agenda

Who we are

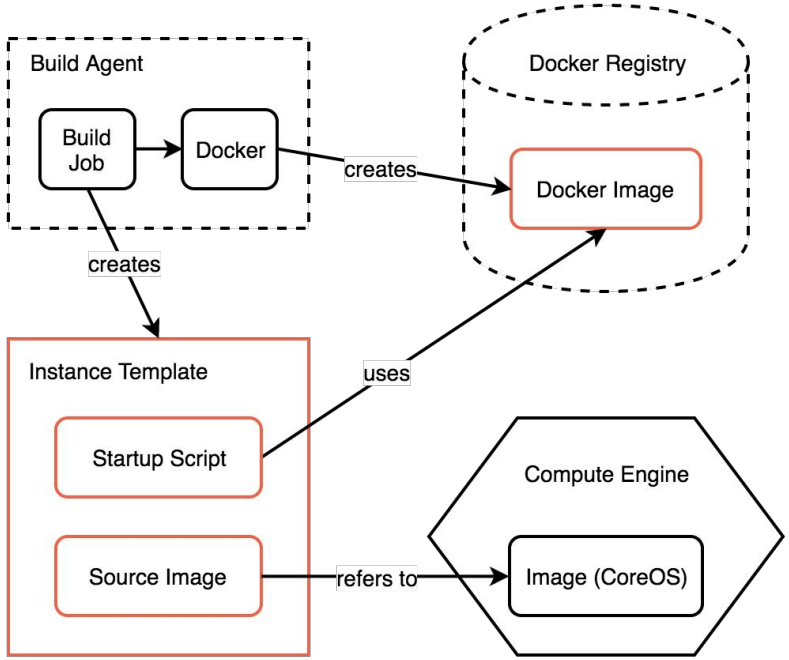
What we left behind

Where we currently are

What is still ahead of us

Docker in Production

- **The delivery pipeline only builds**
  - a Docker image
  - a GCE instance template
- **GCE instances recreated using an existing GCE image**
- **Faster and more efficient deployment**



Thank you!

Questions

?

Contacts

Assen Todorov
Head of Development

Email: assen.todorov@ligatus.com

Teodor Stoev
Senior Software Developer

Email: teodor.stoev@ligatus.com

Sascha Schlegel
Senior Software Developer

Email: sascha.schlegel@ligatus.com

Sebastian Wanka
Student Assistant

Email: sebastian.wanka@ligatus.com