

Entwicklung einer flexibel konfigurierbaren Build-und Integrationstest-Pipeline
der SAP HANA Development Infrastruktur

Kelvin Glaß
Freie Universität Berlin

9. Februar 2015

Grundbegriffe

Aufgabenstellung

Darstellung der aktuellen Infrastruktur

Probleme

Anforderungen an eine Problemlösung

Ergebnisse

Darstellung der neuen Infrastruktur

Fazit

Bewertung der Ergebnisse

Ausblick

Definition

Ein **Pipelineschritt** ist ein Prozess zur Erstellung, Qualitätssicherung oder Verteilung einer Software.

Example

Erstellungsschritt: Maven-build

Schritt zur Qualitätssicherung: Unittest-Ausführung/Auswertung

Definition

Eine **Pipeline** ist eine Verkettung von Pipelineschritten.

Example

Maven-Build → JUnit-Test → Integrationstest → Deploy

Grundbegriffe

Aufgabenstellung

Darstellung der aktuellen Infrastruktur

Probleme

Anforderungen an eine Problemlösung

Ergebnisse

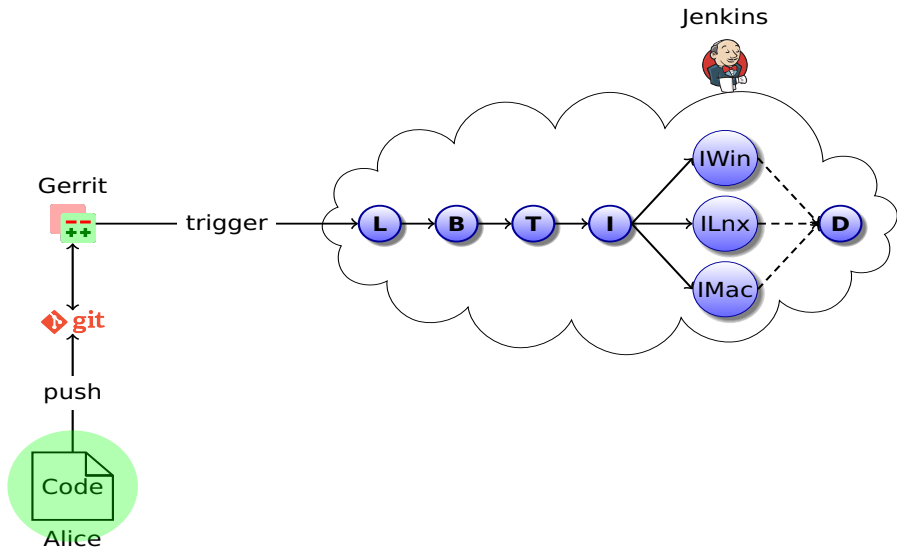
Darstellung der neuen Infrastruktur

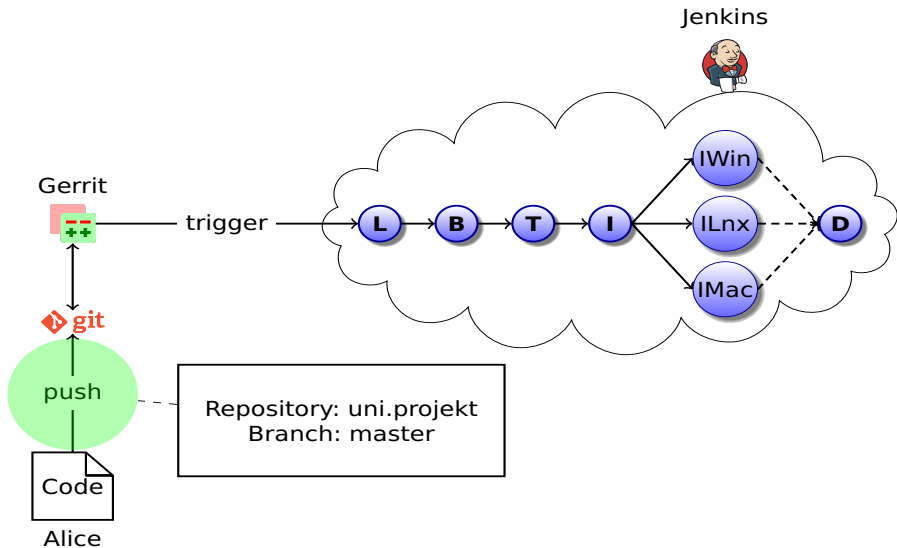
Fazit

Bewertung der Ergebnisse

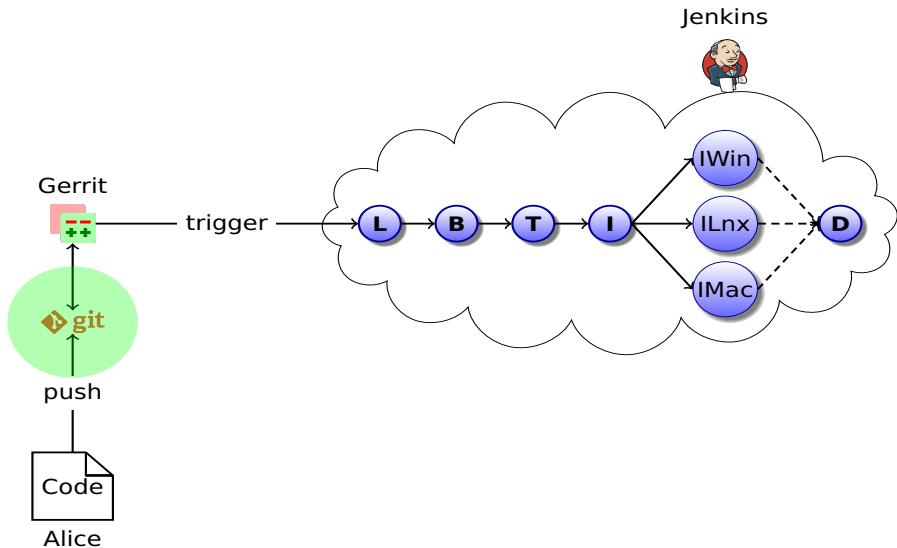
Ausblick

Darstellung der aktuellen Infrastruktur

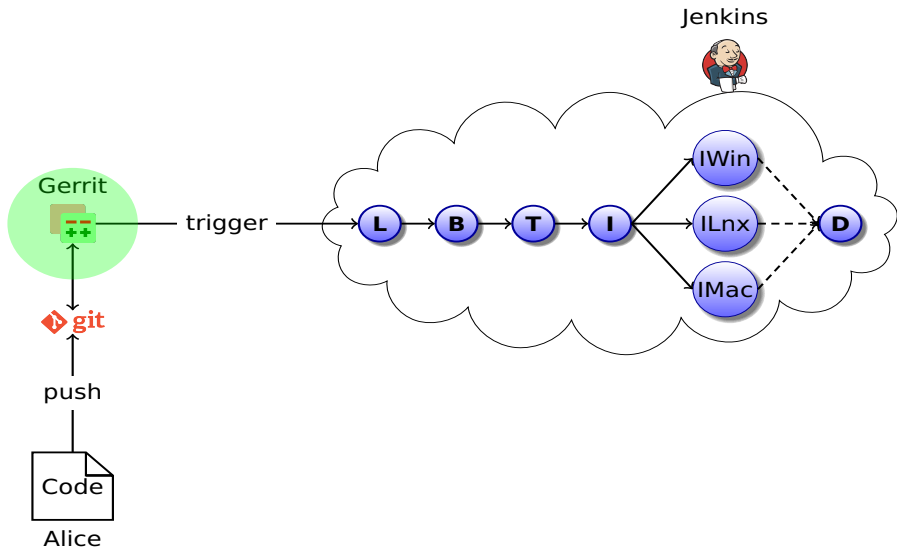




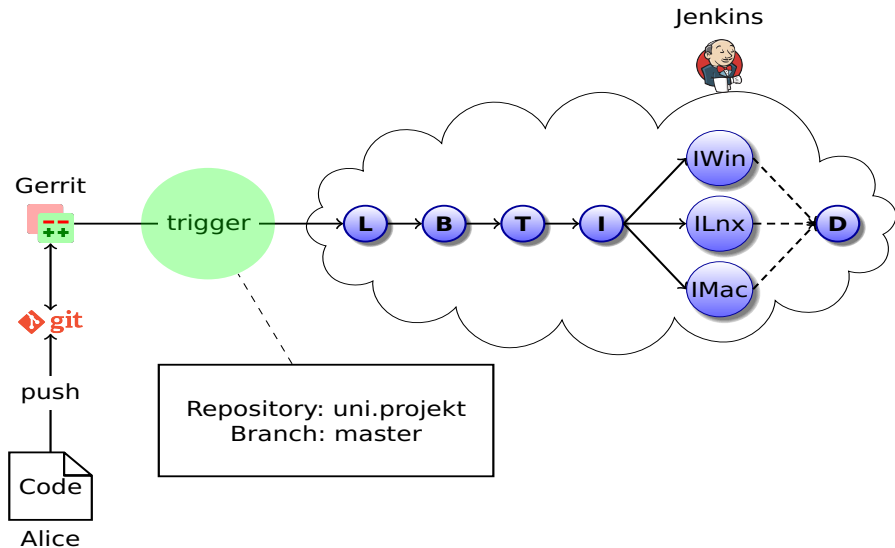
Darstellung der aktuellen Infrastruktur



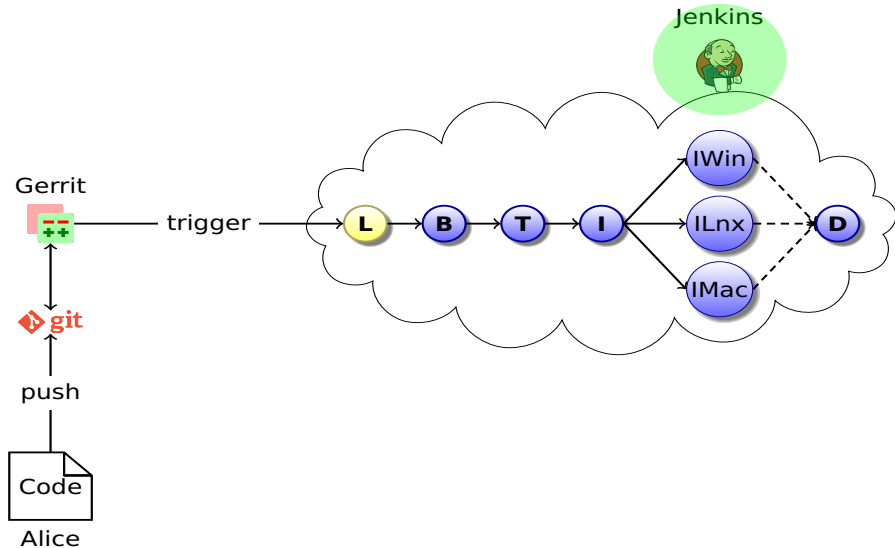
Darstellung der aktuellen Infrastruktur



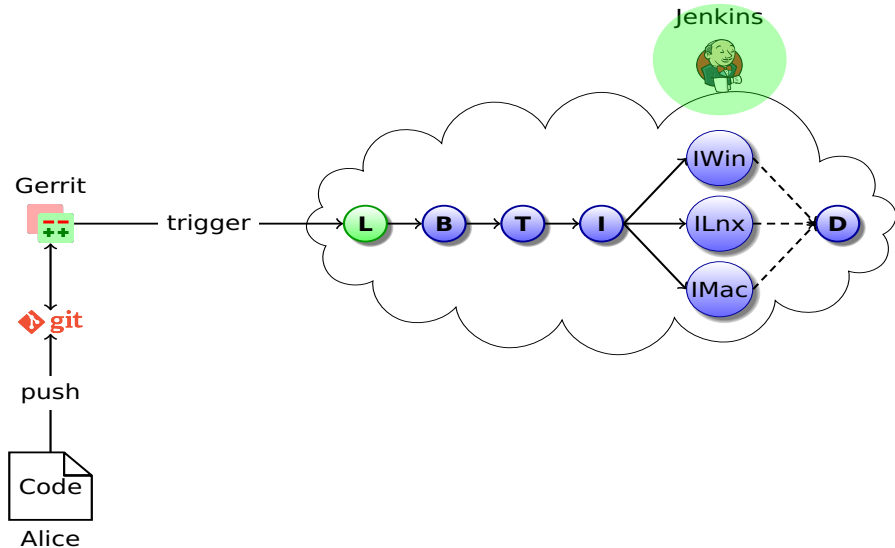
Darstellung der aktuellen Infrastruktur



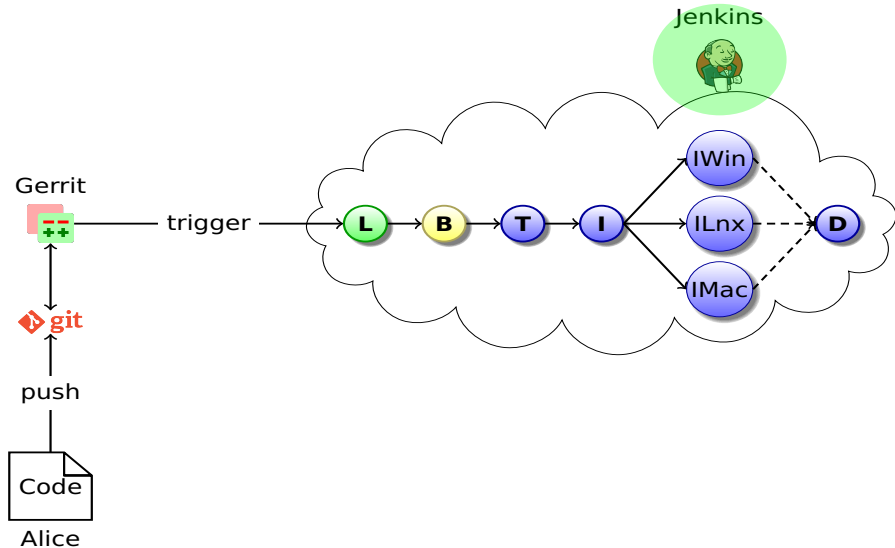
Darstellung der aktuellen Infrastruktur



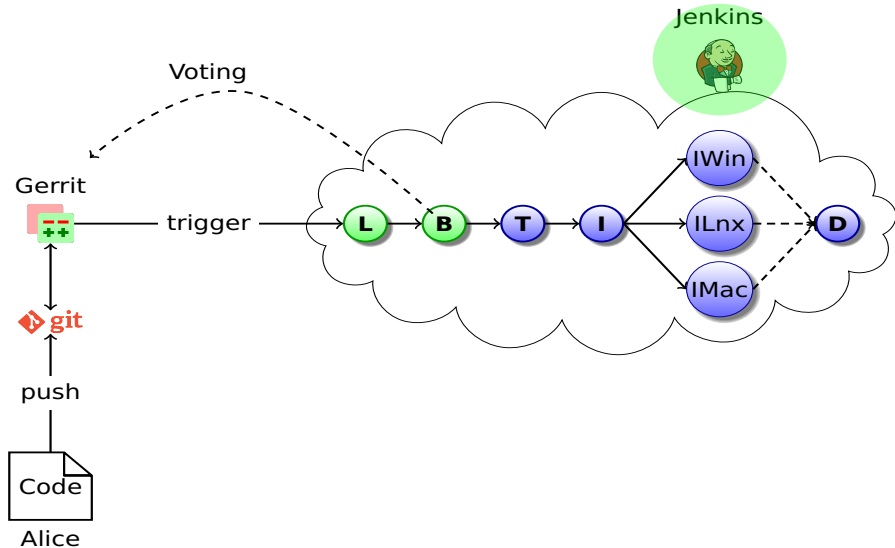
Darstellung der aktuellen Infrastruktur

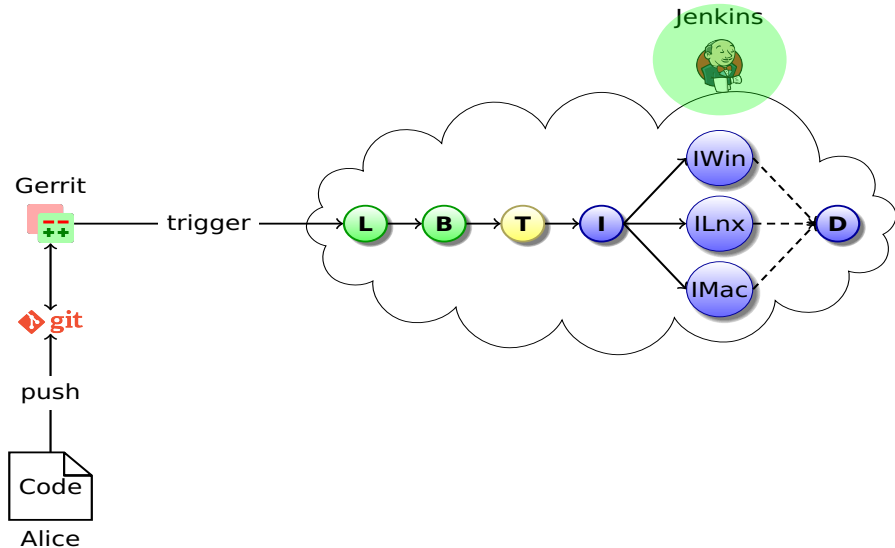


Darstellung der aktuellen Infrastruktur

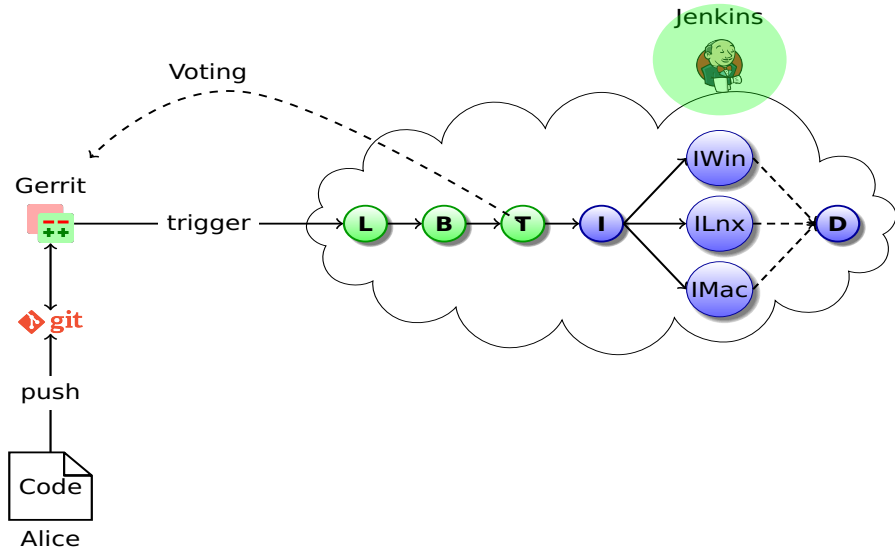


Darstellung der aktuellen Infrastruktur

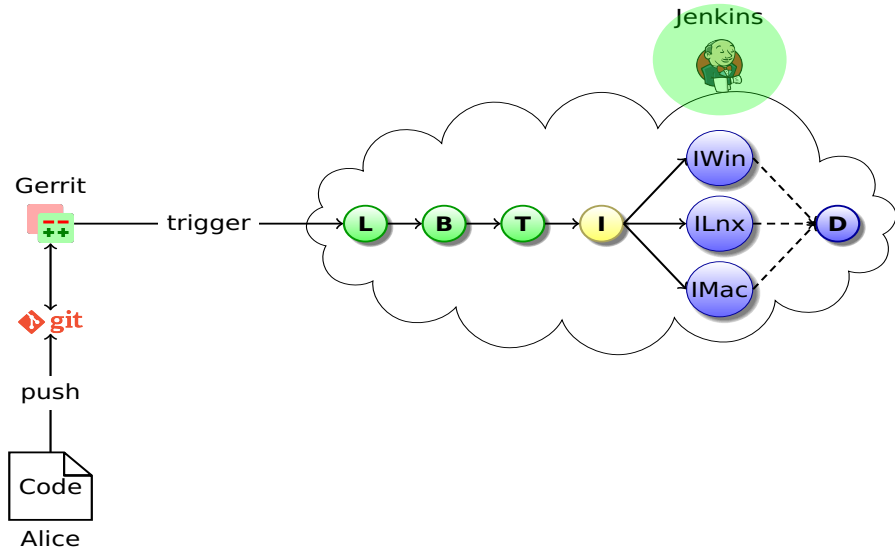




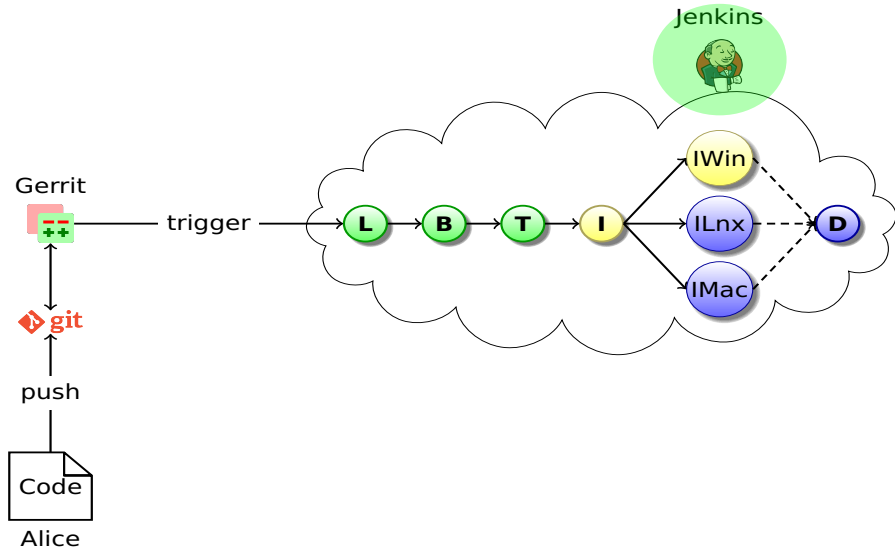
Darstellung der aktuellen Infrastruktur



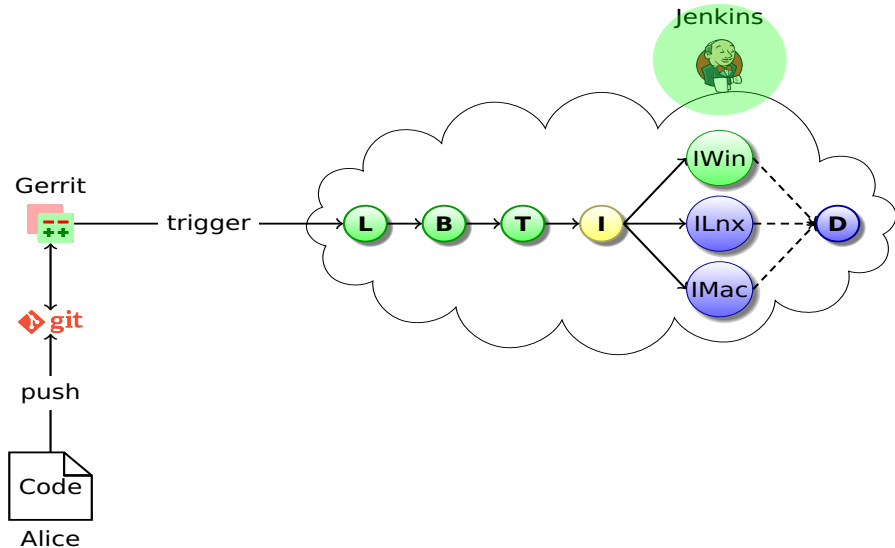
Darstellung der aktuellen Infrastruktur



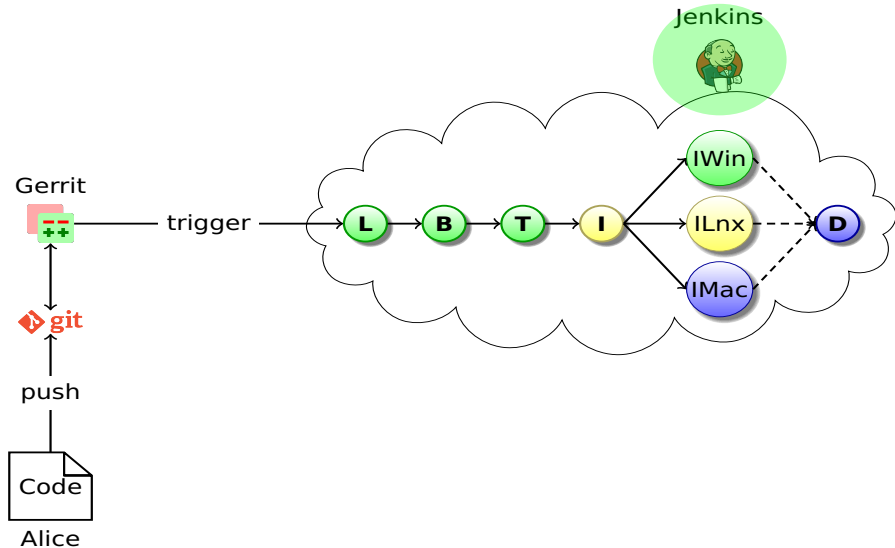
Darstellung der aktuellen Infrastruktur



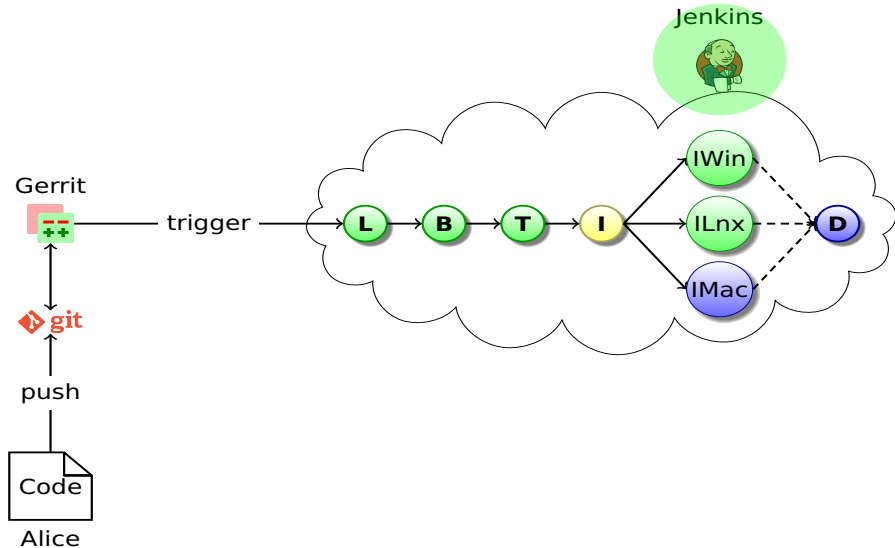
Darstellung der aktuellen Infrastruktur



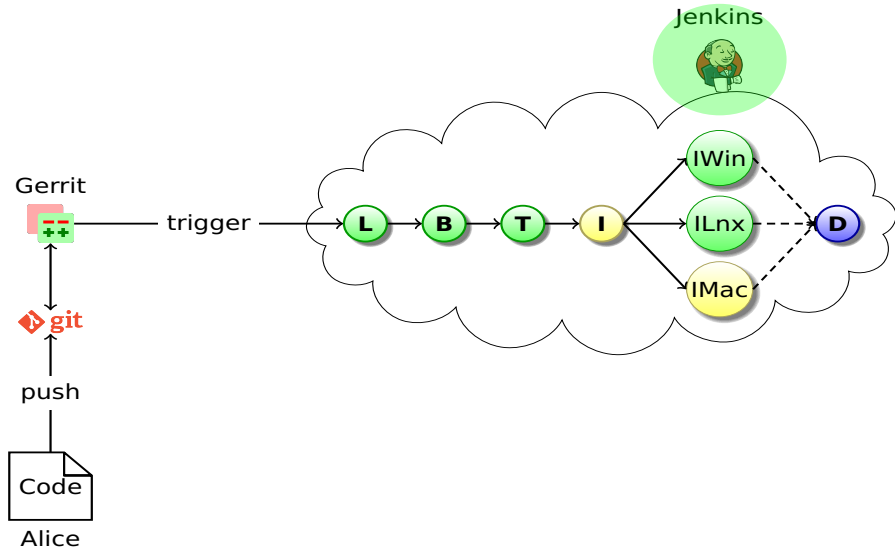
Darstellung der aktuellen Infrastruktur



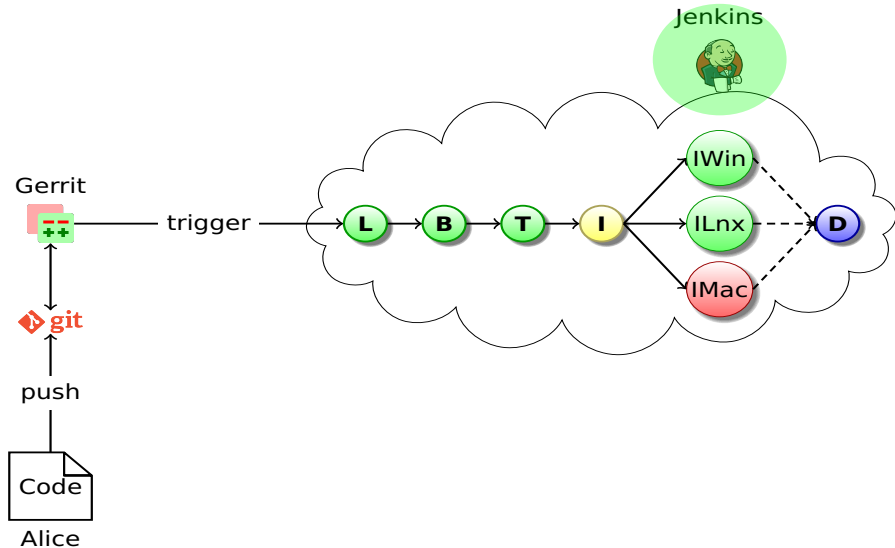
Darstellung der aktuellen Infrastruktur



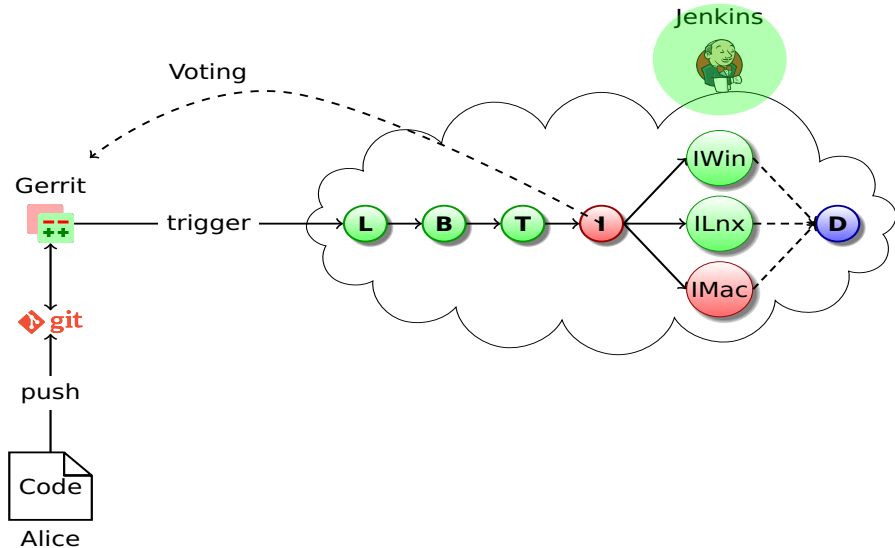
Darstellung der aktuellen Infrastruktur



Darstellung der aktuellen Infrastruktur



Darstellung der aktuellen Infrastruktur



- ▶ Alle Pipelines sind fest implementiert
 - ▶ manuelle Implementierung für jede Pipeline(änderung)
 - ▶ Der Pipelinenutzer kann keine Änderungshistorie einsehen
 - ▶ trotz generischer Jobs

Example

Maven-Build-Job nutzbar für alle Maven-Projekte in Java

- ▶ Kein partieller Pipelinedurchlauf zur Beendigung eines gescheiterten Durchlaufs möglich

Anforderungen an eine Problemlösung

1. Pipelinekonfigurationen müssen:
 - ▶ persistent hinterlegt werden,
 - ▶ modifizierbar sein,
 - ▶ als Änderungshistorie einzusehen sein
2. Pipeline muss retriggered werden können (In jedem Pipelineschritt)
 - ▶ Informationen eines Schrittdurchlaufs speichern
3. Das System muss eine hohe Wartbarkeit besitzen

Outline

Grundbegriffe

Aufgabenstellung

Darstellung der aktuellen Infrastruktur

Probleme

Anforderungen an eine Problemlösung

Ergebnisse

Darstellung der neuen Infrastruktur

Fazit

Bewertung der Ergebnisse

Ausblick

1. Konfiguration persistieren



- ▶ abstrakte Konfiguration zur Abbildung einer Pipeline

PipeConfig.yaml

```
1 Version: 1.0
2 Pipeline:
3   name: "Alice-pipeline for uni-project"
4   product:
5     name: "Uni-Projekt"
6     language: "java"
7     repository: "uni.projekt"
8     branches:
9     - "master"
```

Steps:

```
11 - name: 'Build'
12   build:
13     type: maven
14   post:
15     Gerrit:
16       voting:
17         verified:
18           failure:
19             value: -1
20             text: "build of 'Uni Projekt' failed"
21           success:
22             value: +1
23             text: "successfully built 'Uni Projekt'"
24
```

Beispielkonfiguration

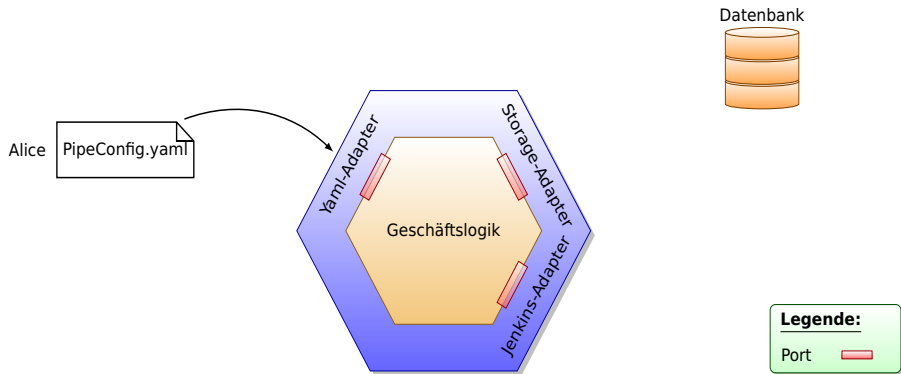
```
25 - name: 'Test'  
26   preconditions:  
27     - result: SUCCESS  
28     stepname: 'Build'  
29   test:  
30     testtype: JUnit  
31   post:  
32     gerrit:  
33       voting:  
34         review:  
35           failure:  
36             value: -2  
37             text: "test of 'Uni Projekt' failed"  
38           success:  
39             value: +2  
40             text: "successfully tested 'Uni Projekt'"
```

Beispielkonfiguration

```
41 - name: 'IMac'
42   ...
43 - name: 'ILnx'
44   ...
45 - name: 'IWin'
46   preconditions:
47     - result: SUCCESS
48     stepname: 'Test'
49   ...
50 - name: 'Deploy'
51   preconditions:
52     - result: SUCCESS
53     stepname: 'IMac'
54     - result: SUCCESS
55     stepname: 'ILnx'
56     - result: SUCCESS
57     stepname: 'IWin'
58   ...
```

1. Pipelinekonfiguration persistieren

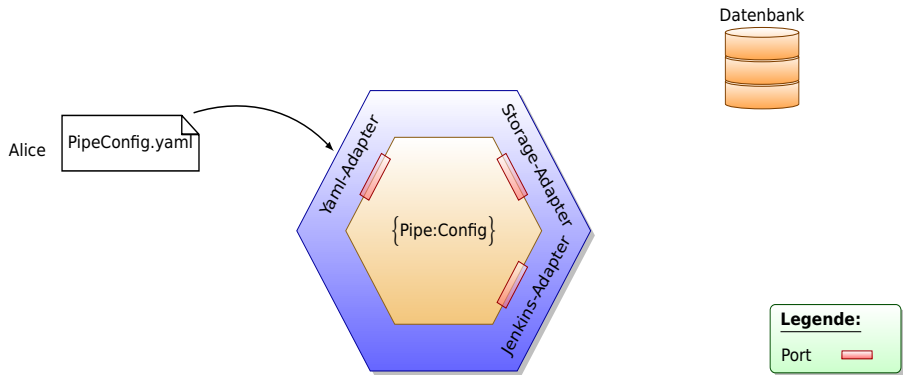
- ▶ Nur über die Geschäftslogik
- ▶ Yaml \rightarrow HashMap
- ▶ HashMap \rightarrow SQL
- ▶ SQL wird abgesetzt



Abarbeitungsweise der neuen Infrastruktur

1. Pipelinekonfiguration persistieren

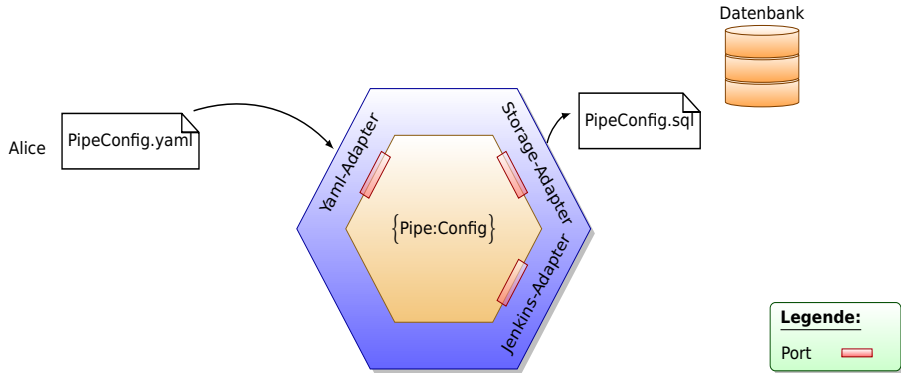
- ▶ Nur über die Geschäftslogik
- ▶ **Yaml** \rightarrow **HashMap**
- ▶ HashMap \rightarrow SQL
- ▶ SQL wird abgesetzt



Abarbeitungsweise der neuen Infrastruktur

1. Pipelinekonfiguration persistieren

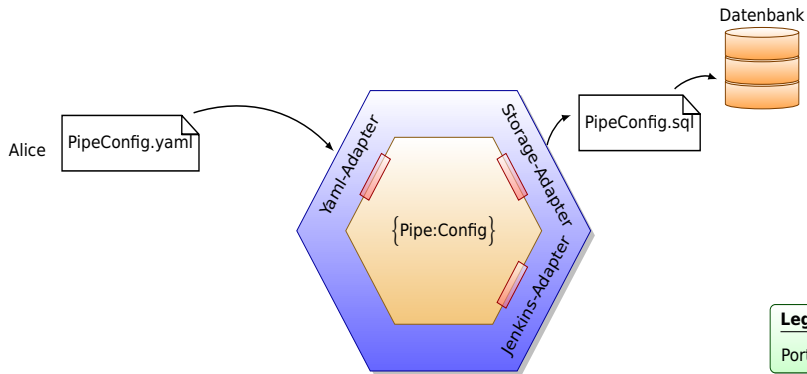
- ▶ Nur über die Geschäftslogik
- ▶ Yaml \rightarrow HashMap
- ▶ **HashMap \rightarrow SQL**
- ▶ SQL wird abgesetzt

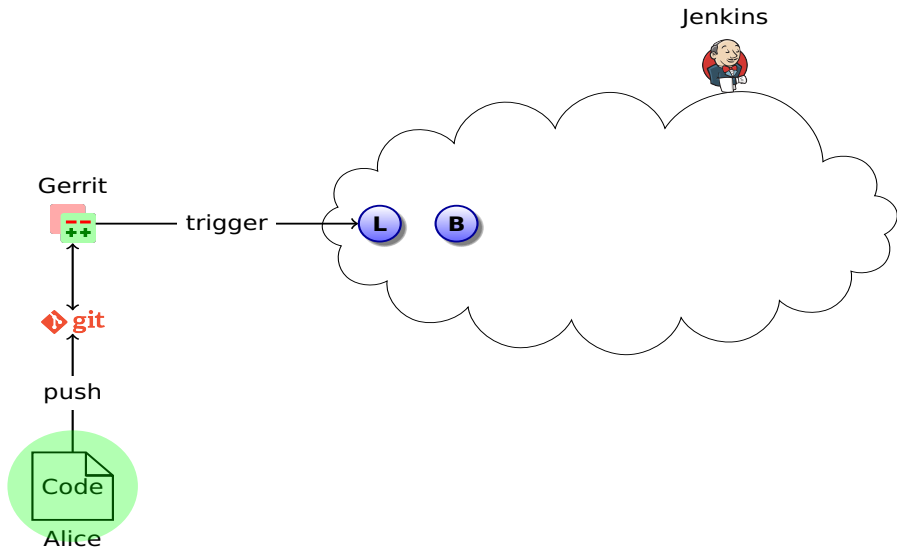


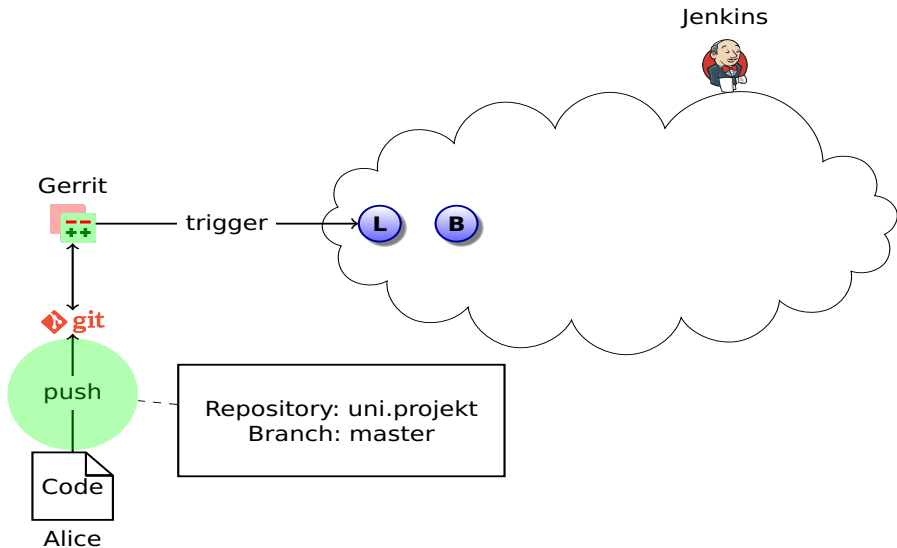
Abarbeitungsweise der neuen Infrastruktur

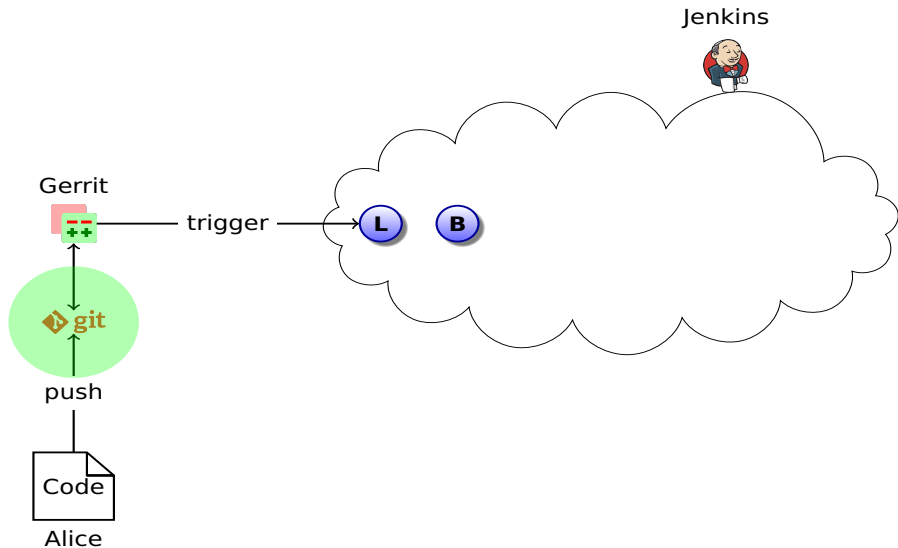
1. Pipelinekonfiguration persistieren

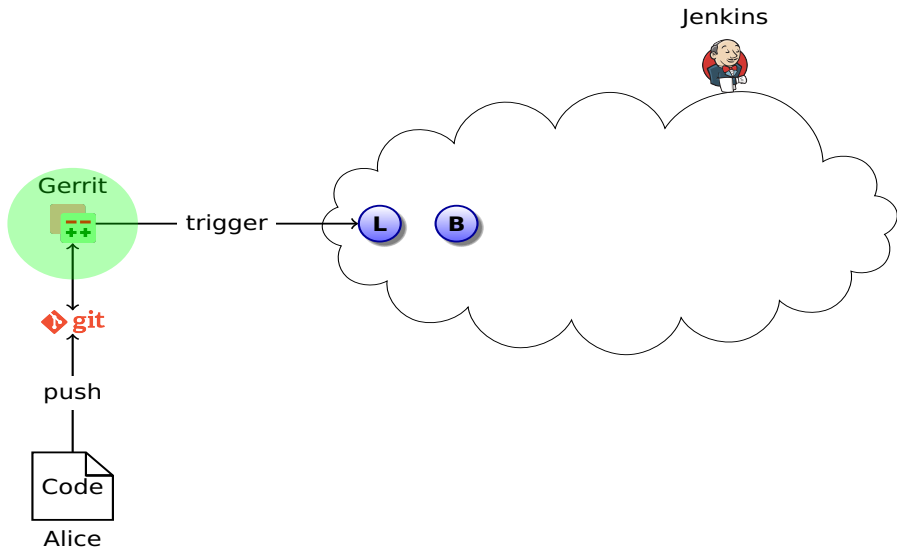
- ▶ Nur über die Geschäftslogik
- ▶ Yaml \rightarrow HashMap
- ▶ HashMap \rightarrow SQL
- ▶ **SQL wird abgesetzt**

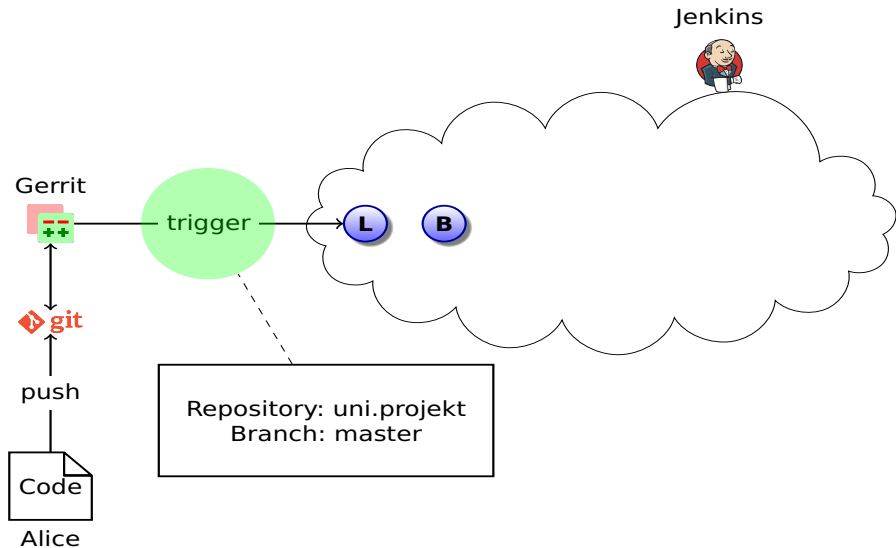


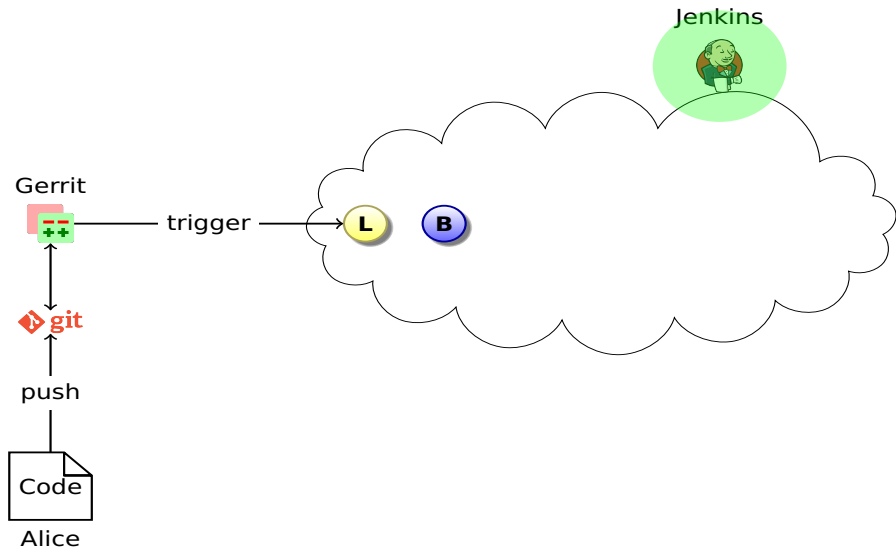




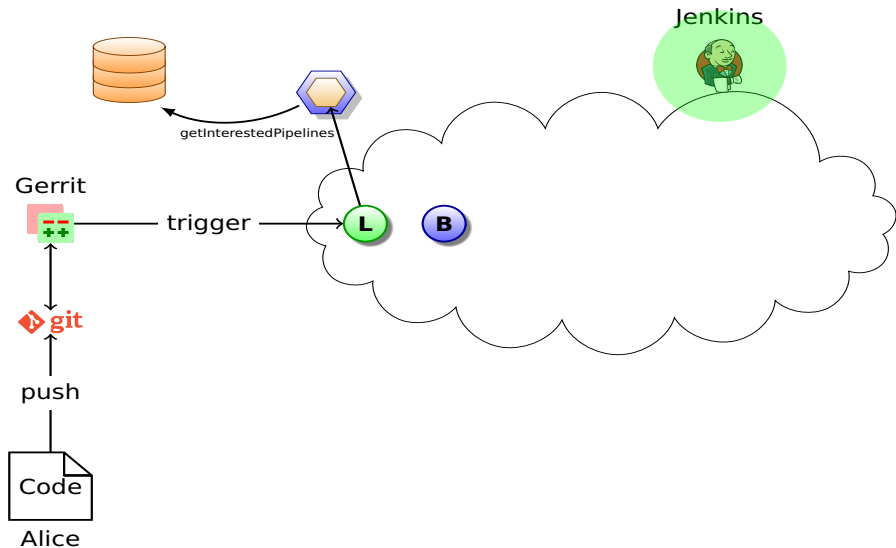




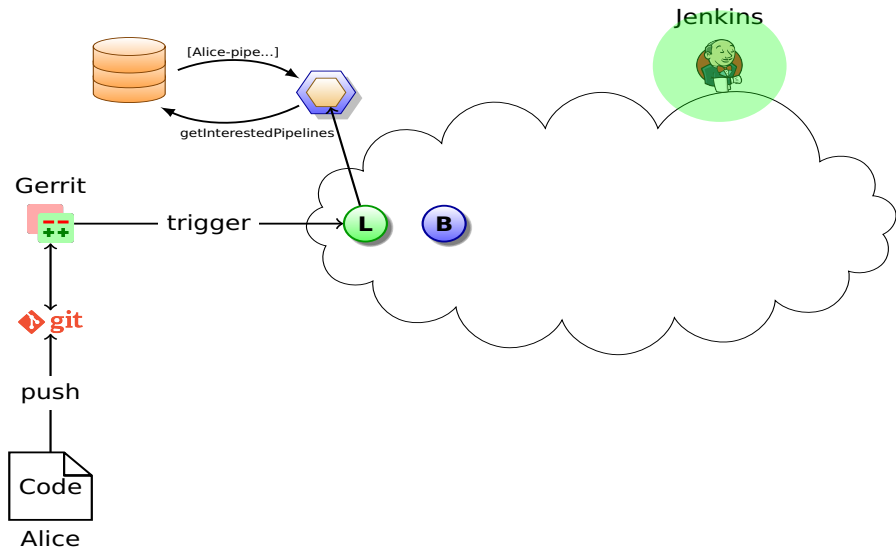




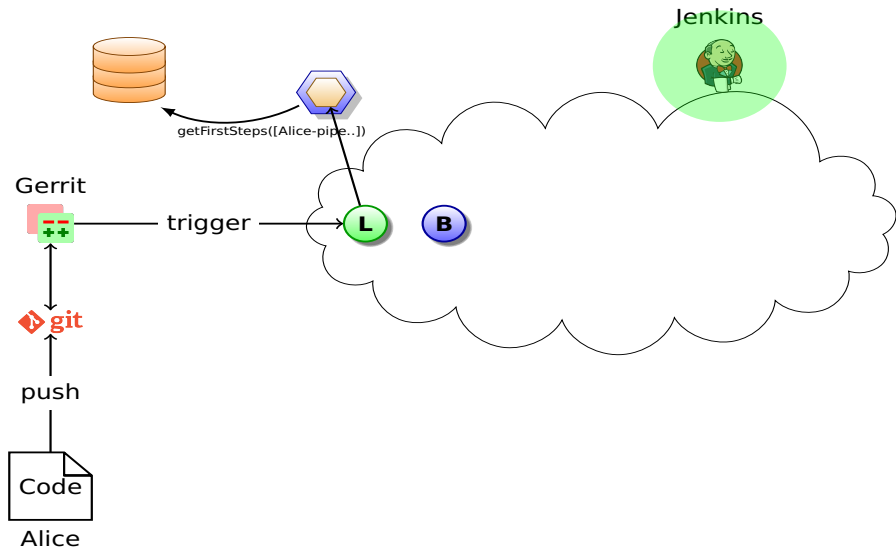
Abarbeitungsweise der neuen Infrastruktur



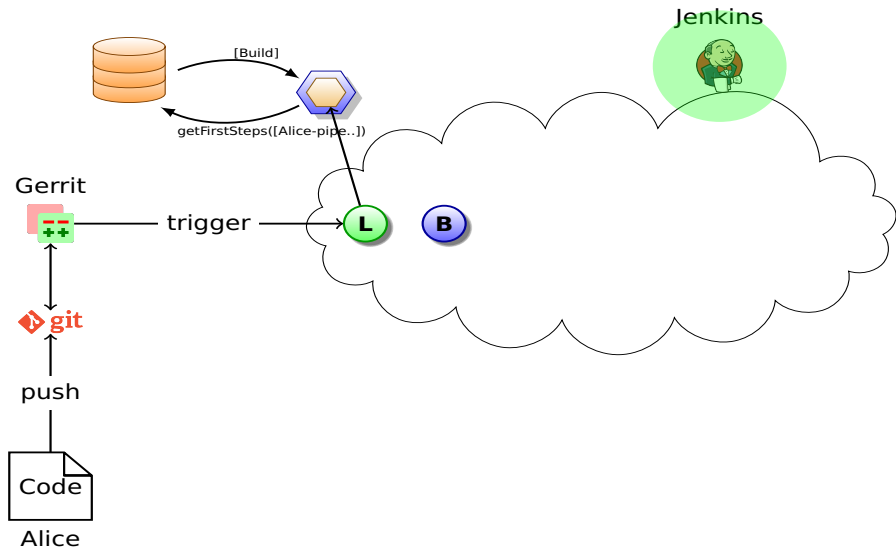
Abarbeitungsweise der neuen Infrastruktur



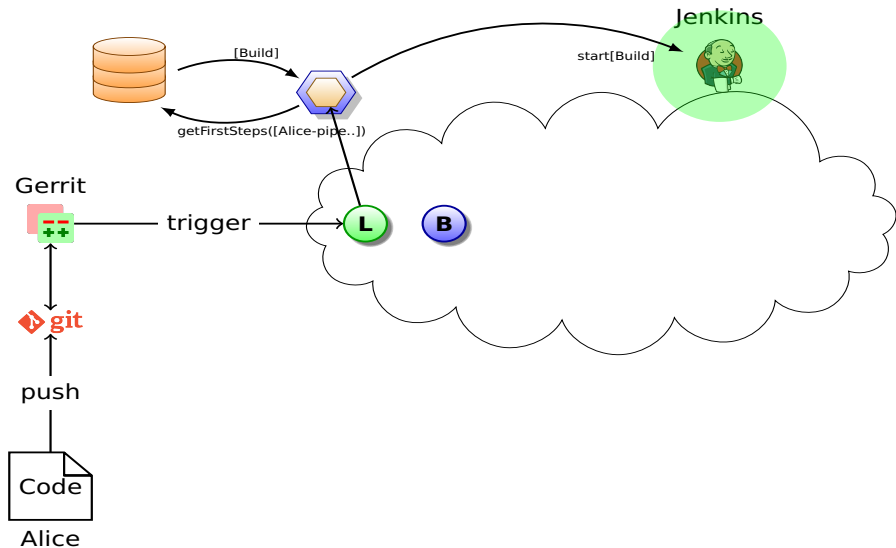
Abarbeitungsweise der neuen Infrastruktur



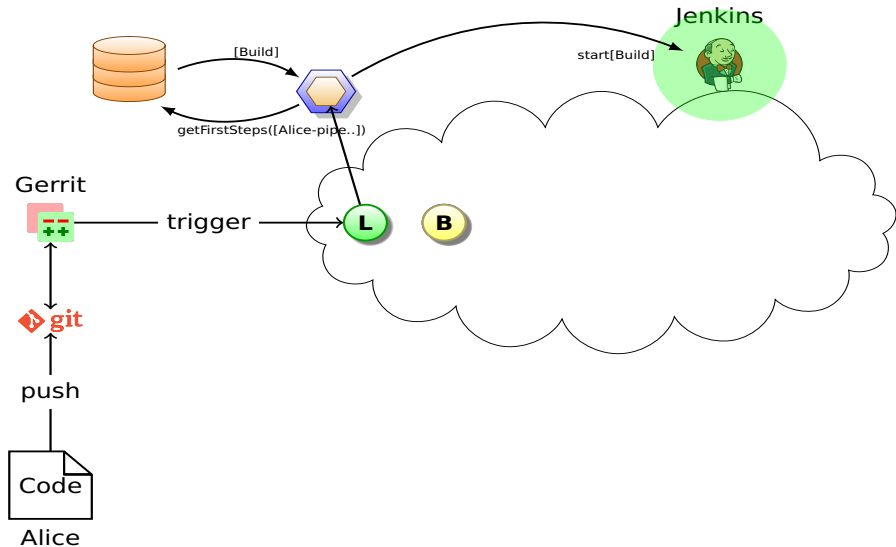
Abarbeitungsweise der neuen Infrastruktur

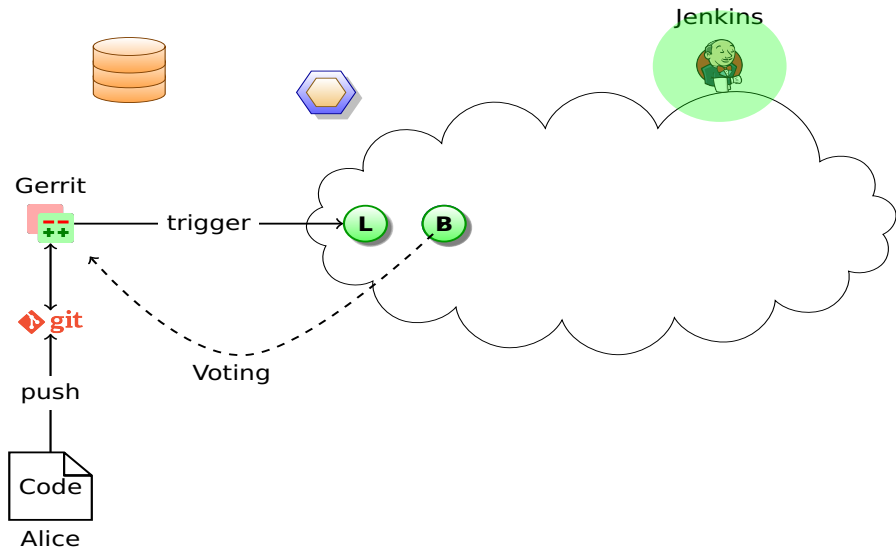


Abarbeitungsweise der neuen Infrastruktur

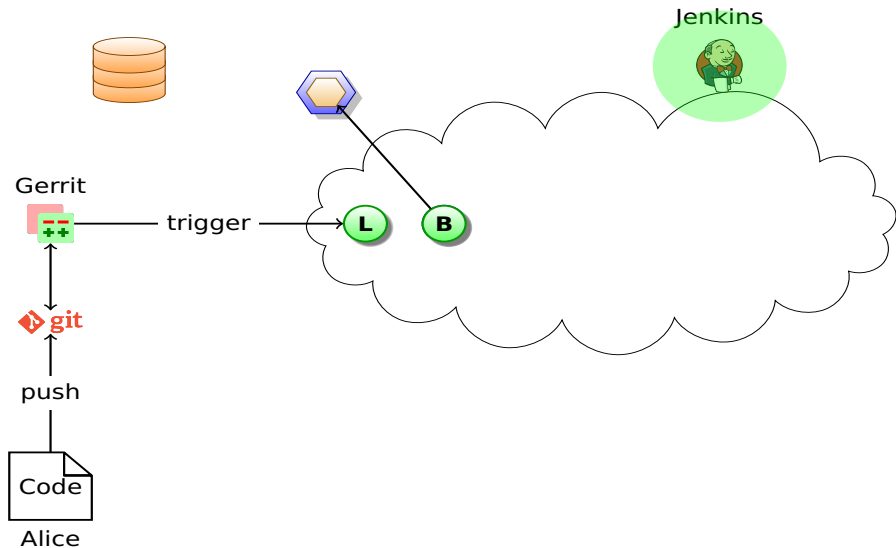


Abarbeitungsweise der neuen Infrastruktur

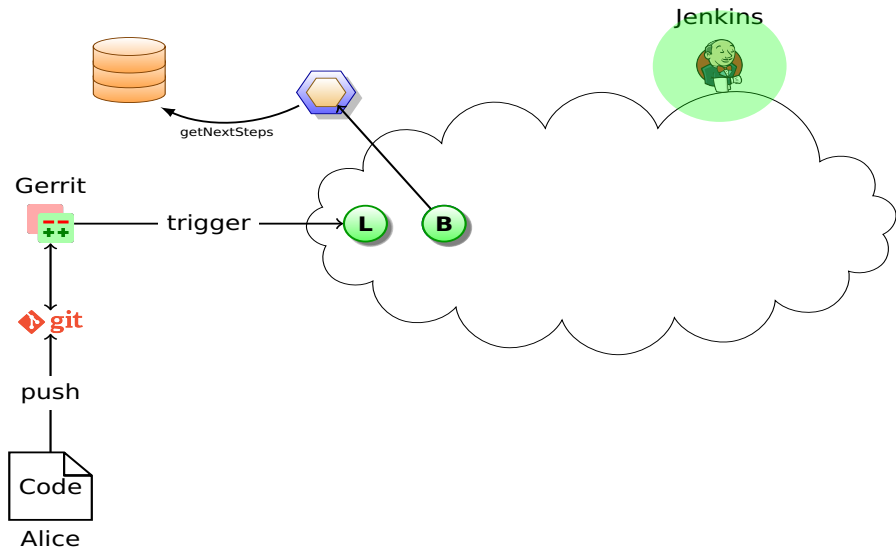




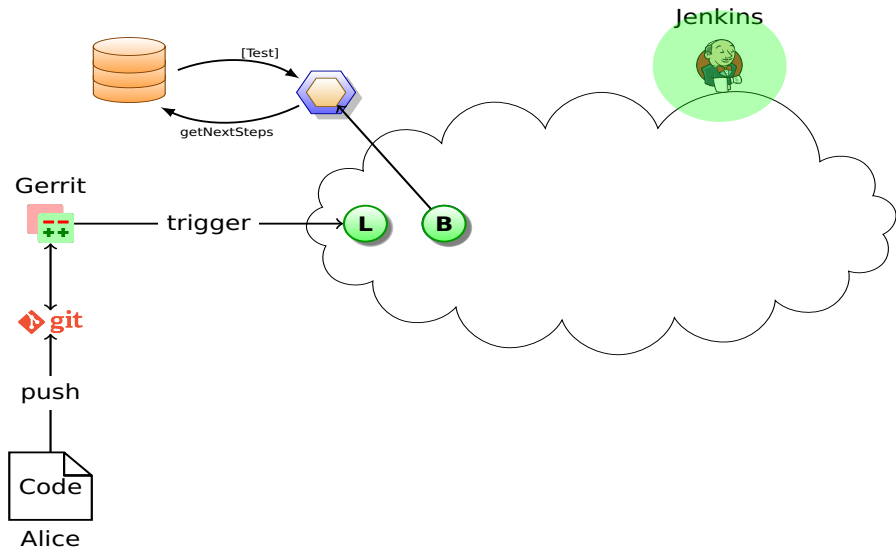
Abarbeitungsweise der neuen Infrastruktur



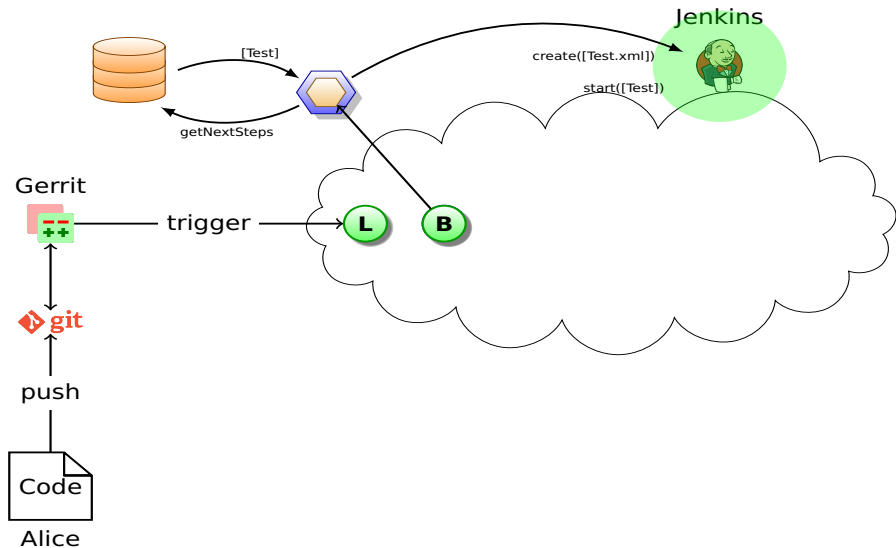
Abarbeitungsweise der neuen Infrastruktur



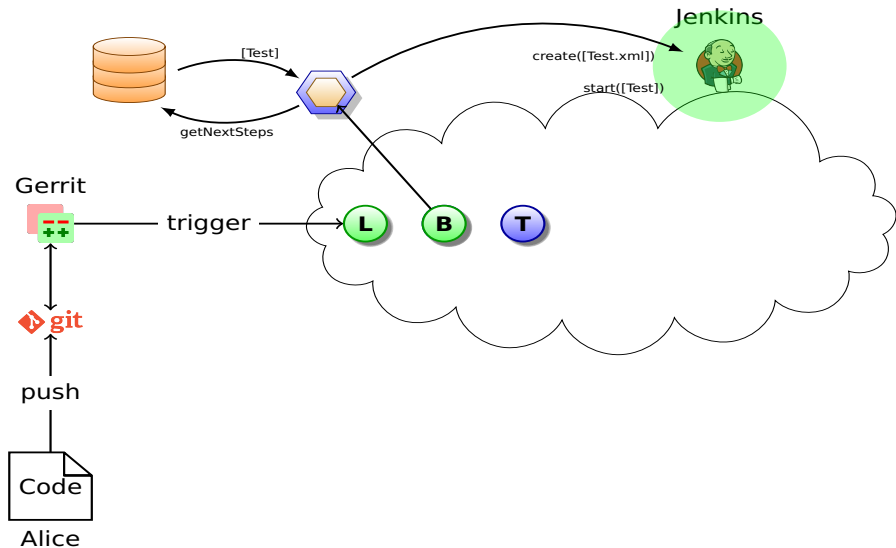
Abarbeitungsweise der neuen Infrastruktur



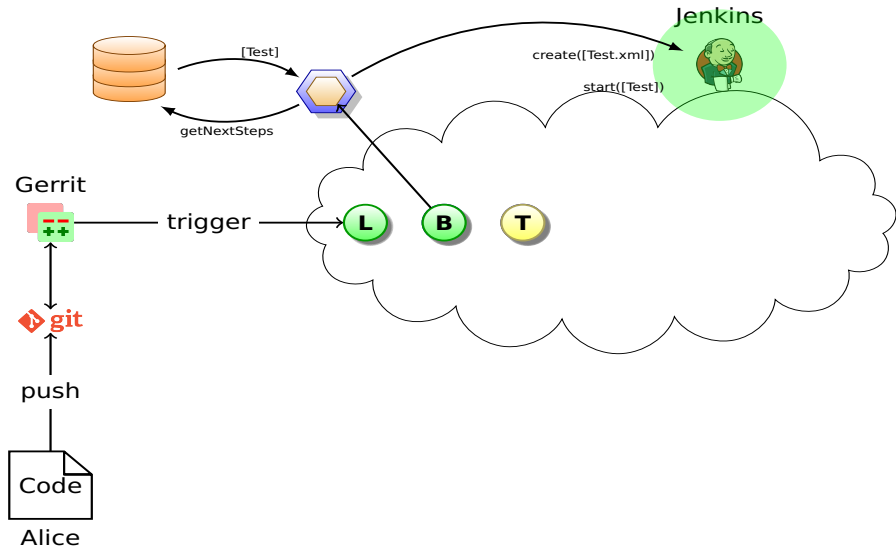
Abarbeitungsweise der neuen Infrastruktur



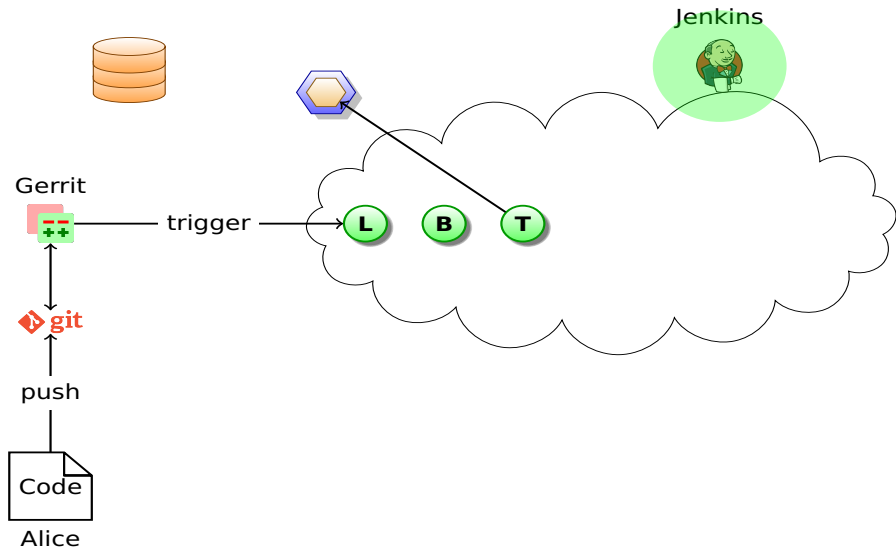
Abarbeitungsweise der neuen Infrastruktur



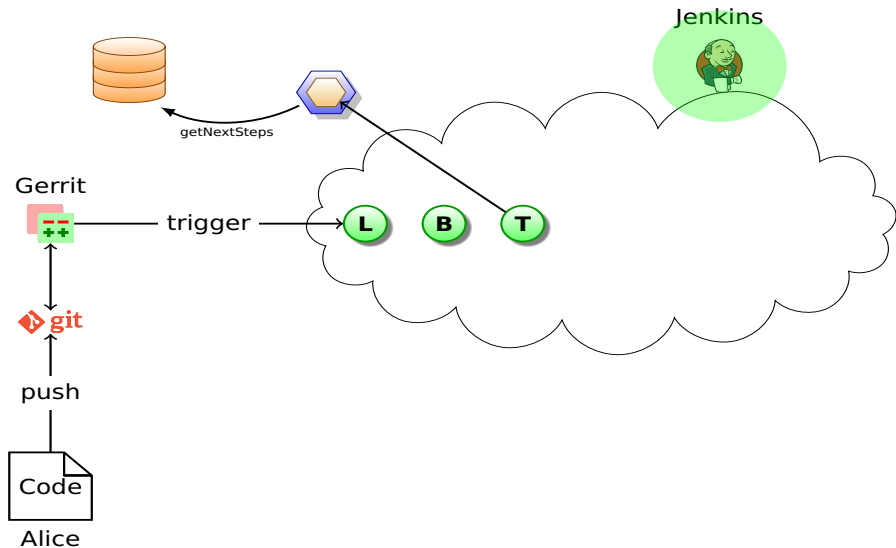
Abarbeitungsweise der neuen Infrastruktur



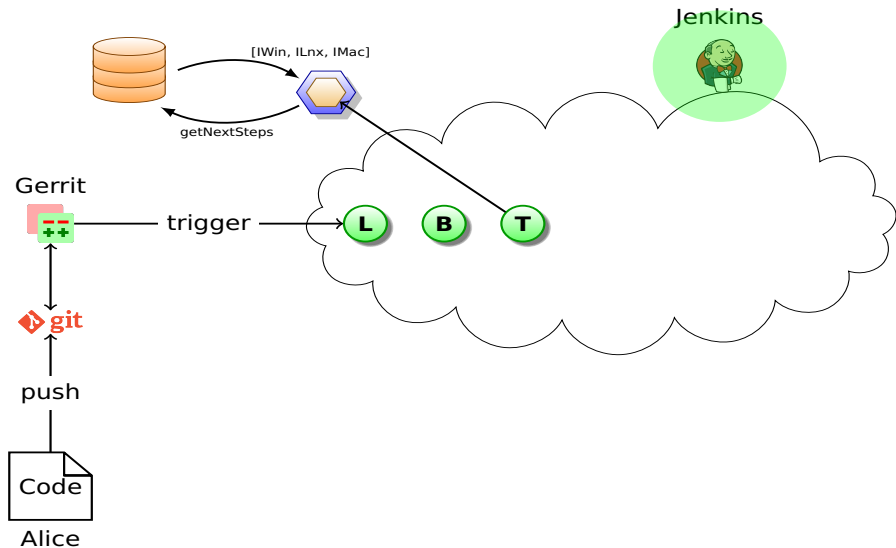
Abarbeitungsweise der neuen Infrastruktur



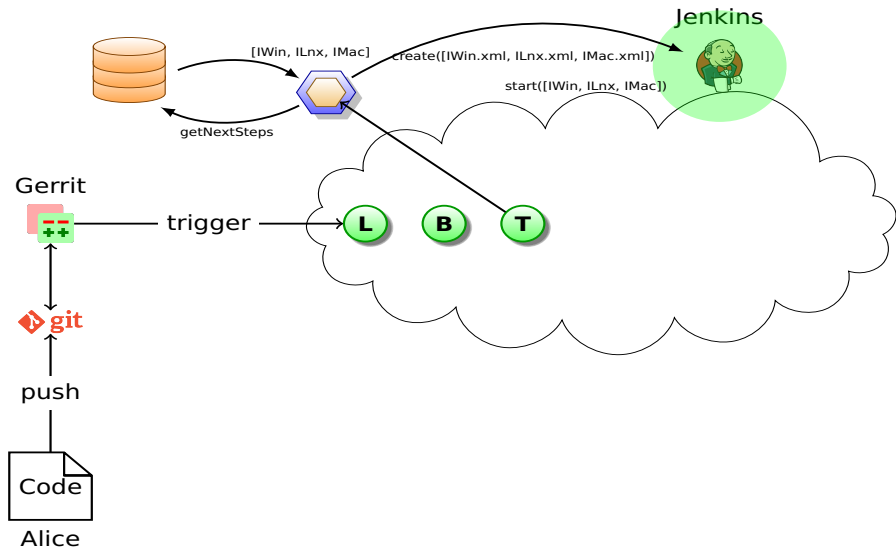
Abarbeitungsweise der neuen Infrastruktur



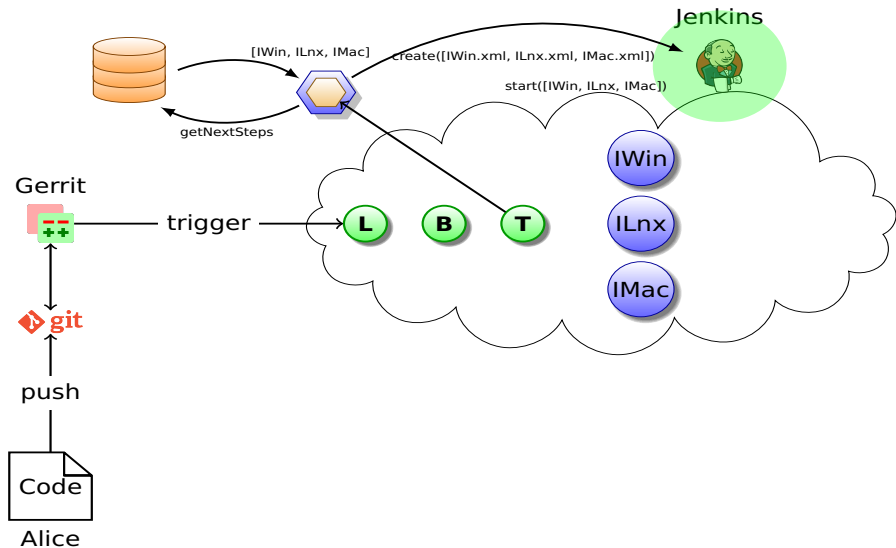
Abarbeitungsweise der neuen Infrastruktur



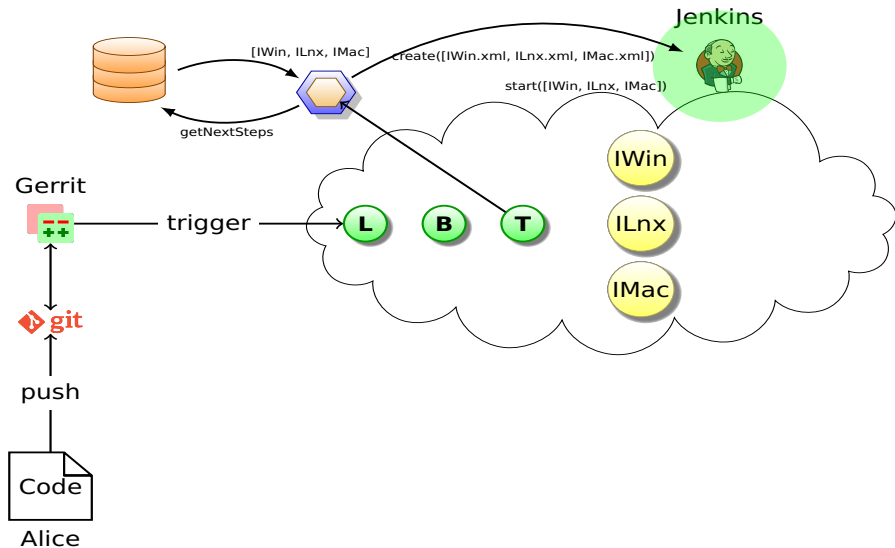
Abarbeitungsweise der neuen Infrastruktur



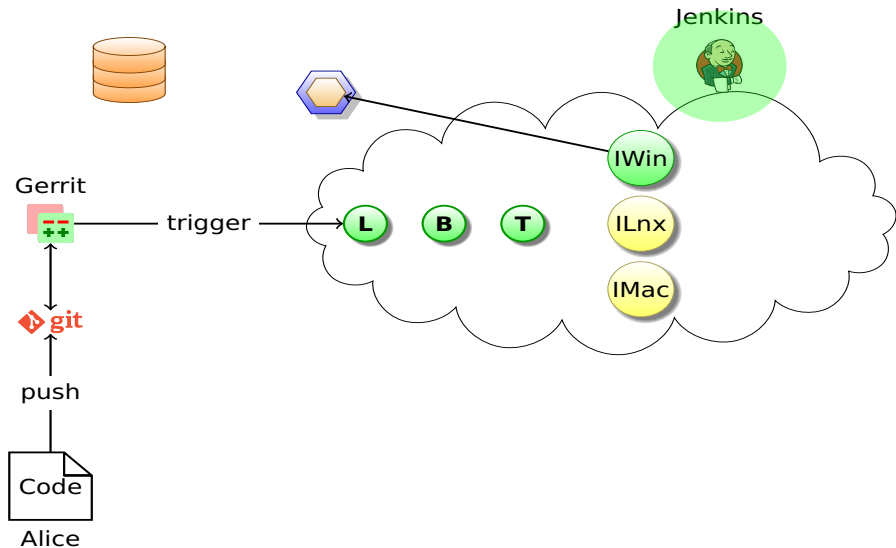
Abarbeitungsweise der neuen Infrastruktur



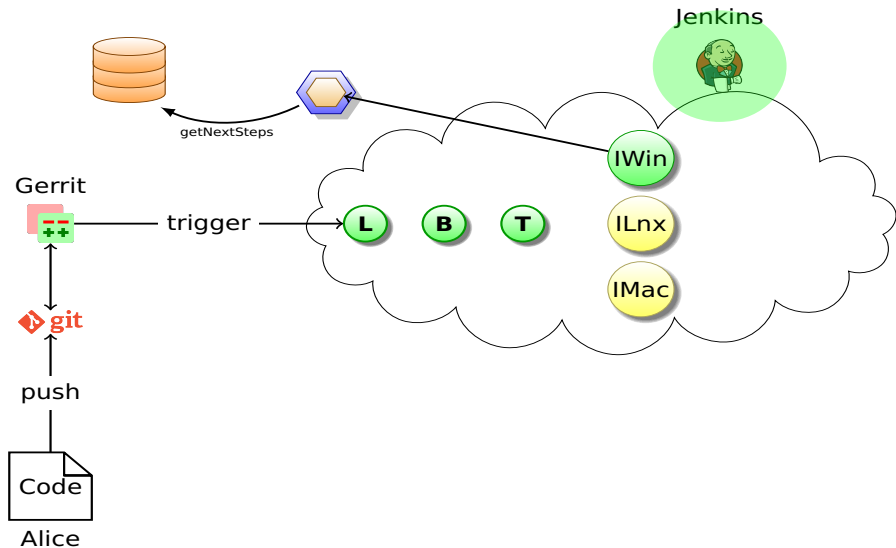
Abarbeitungsweise der neuen Infrastruktur



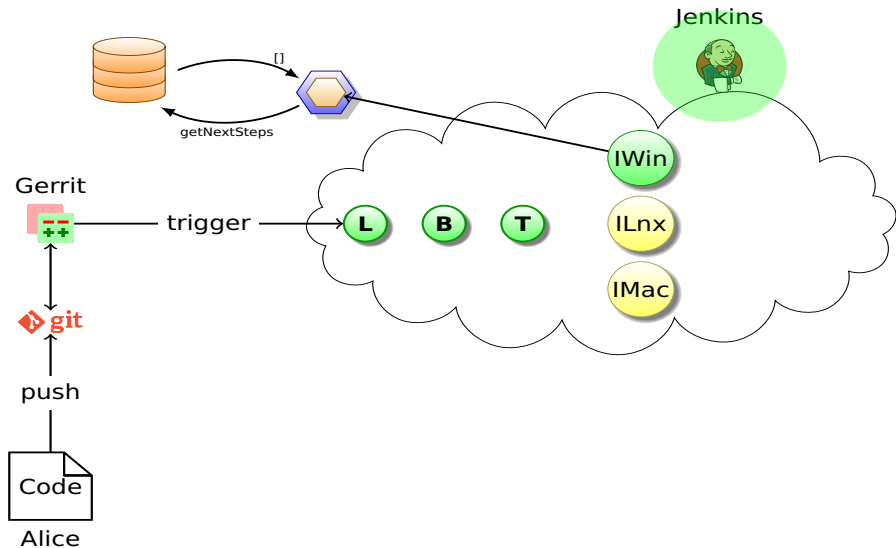
Abarbeitungsweise der neuen Infrastruktur



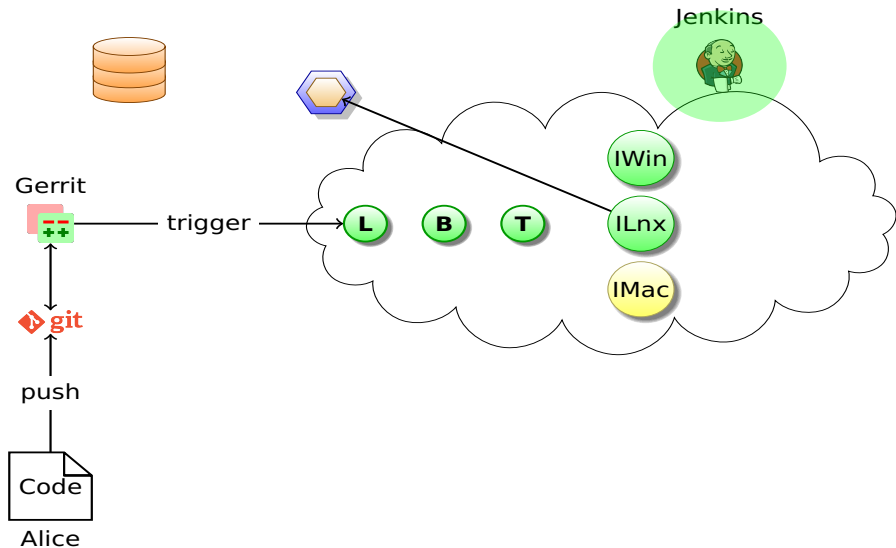
Abarbeitungsweise der neuen Infrastruktur



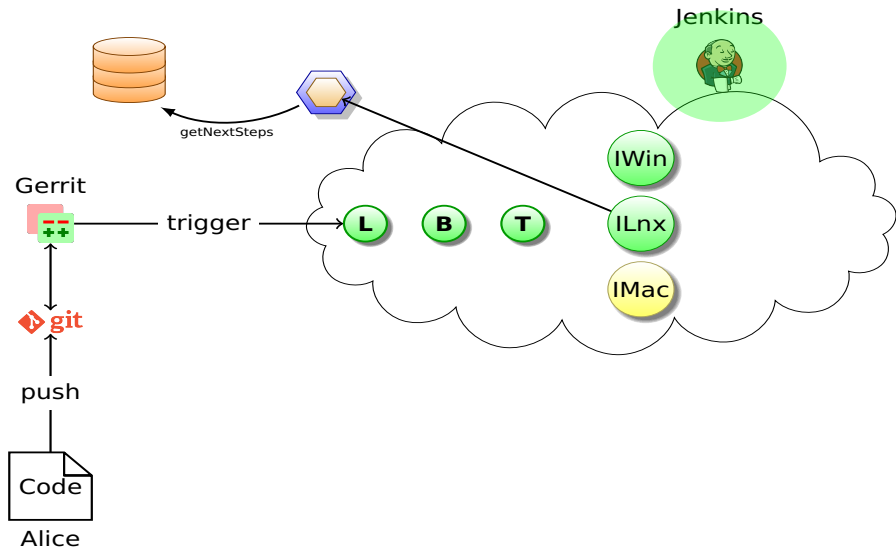
Abarbeitungsweise der neuen Infrastruktur



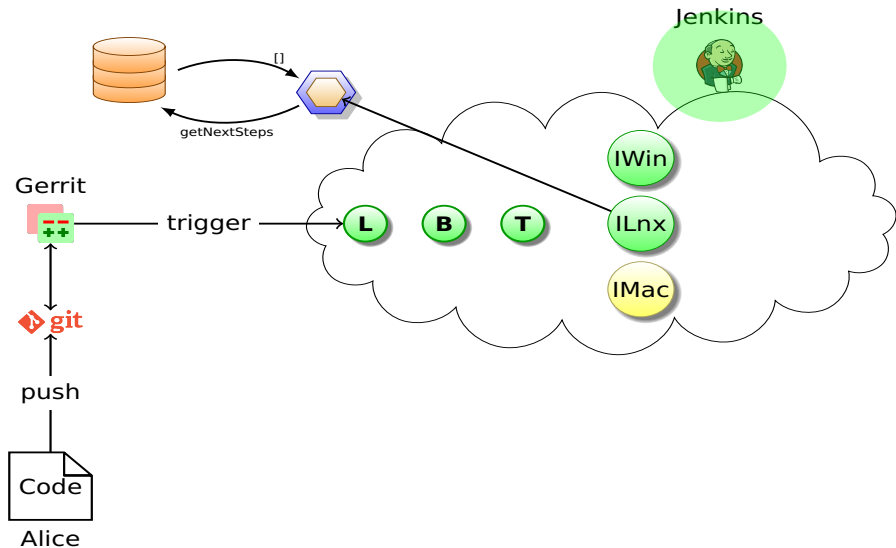
Abarbeitungsweise der neuen Infrastruktur



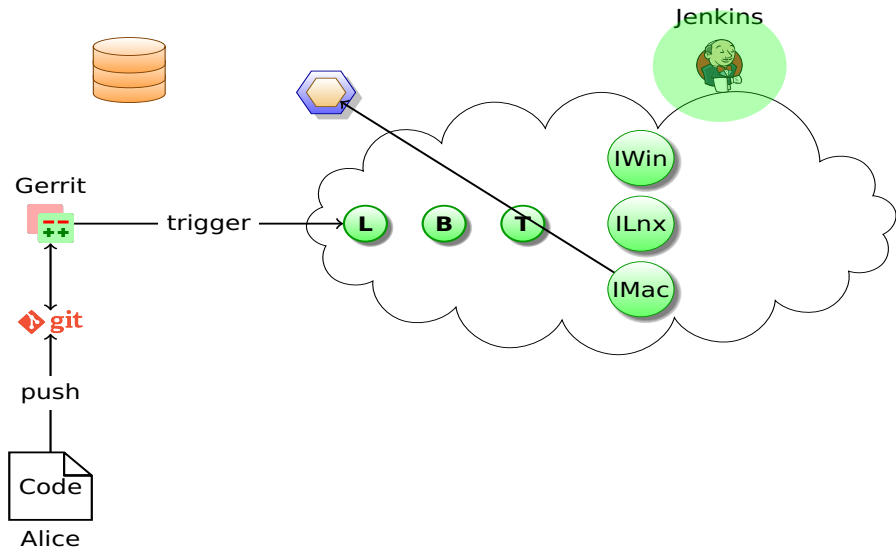
Abarbeitungsweise der neuen Infrastruktur



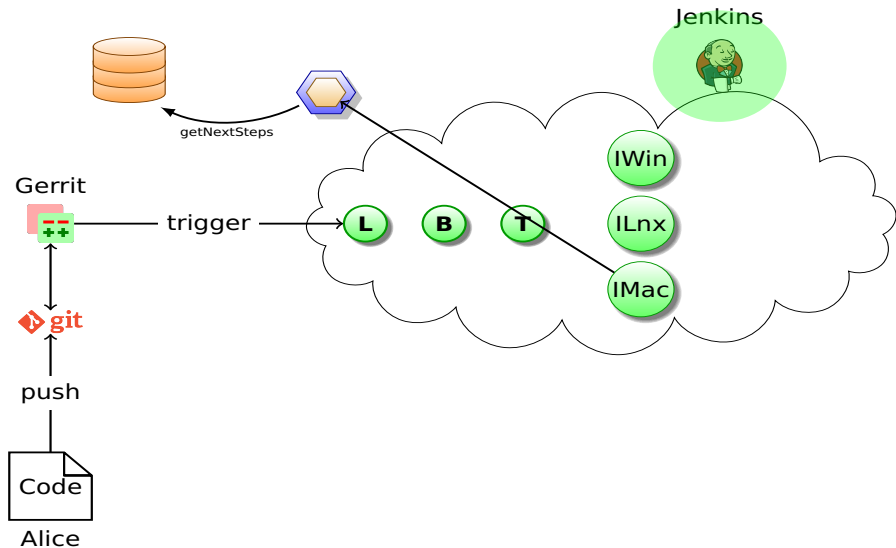
Abarbeitungsweise der neuen Infrastruktur



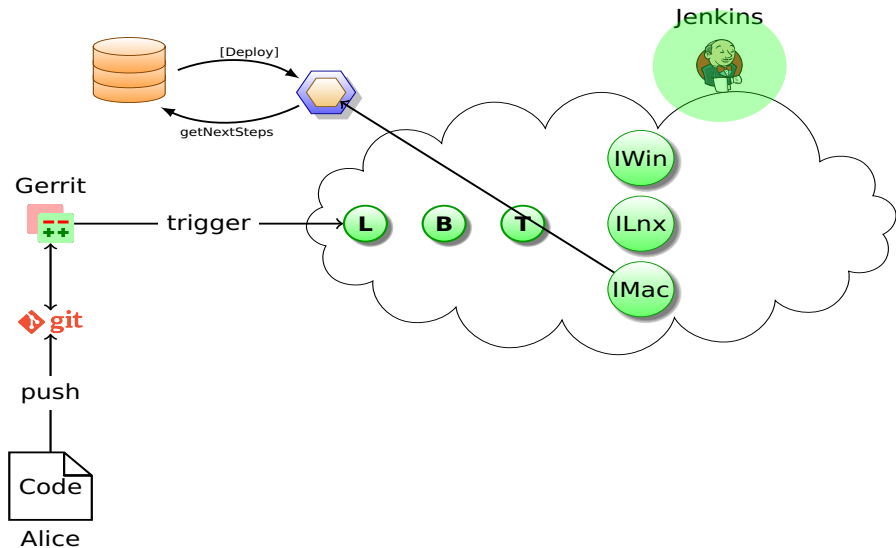
Abarbeitungsweise der neuen Infrastruktur



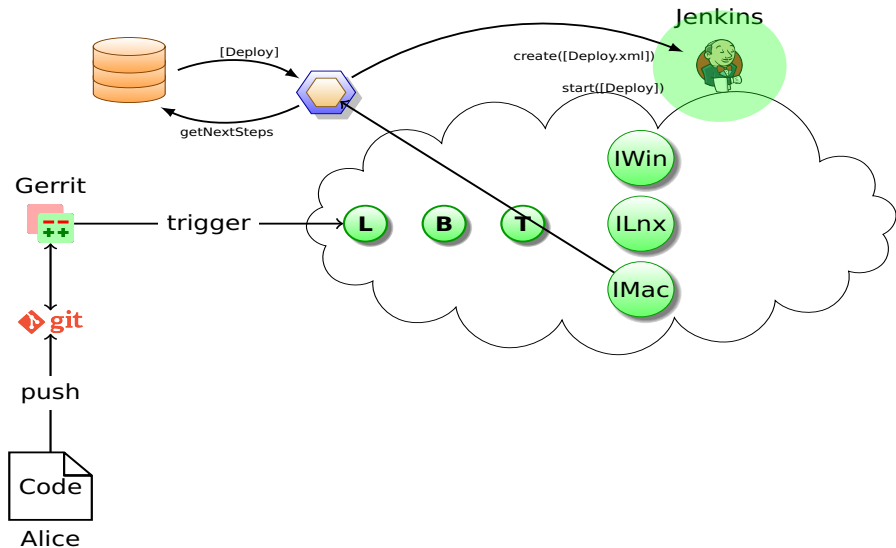
Abarbeitungsweise der neuen Infrastruktur



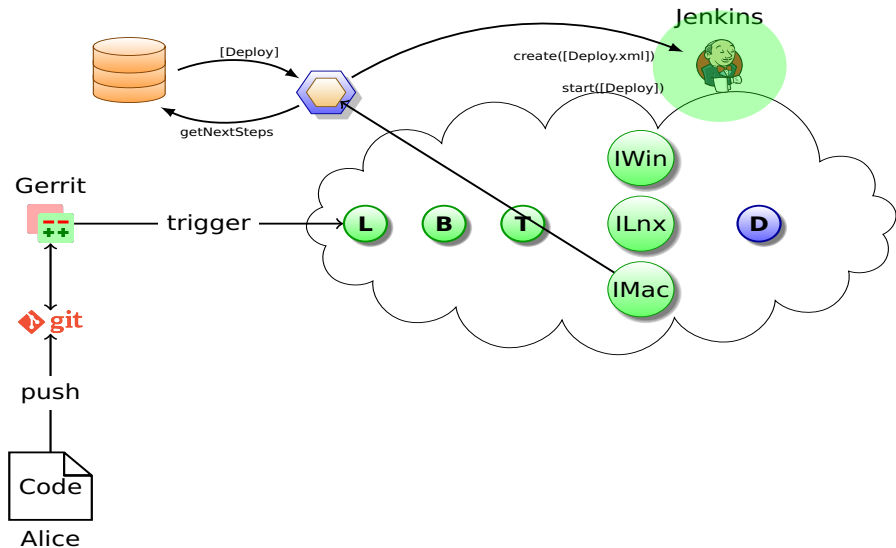
Abarbeitungsweise der neuen Infrastruktur



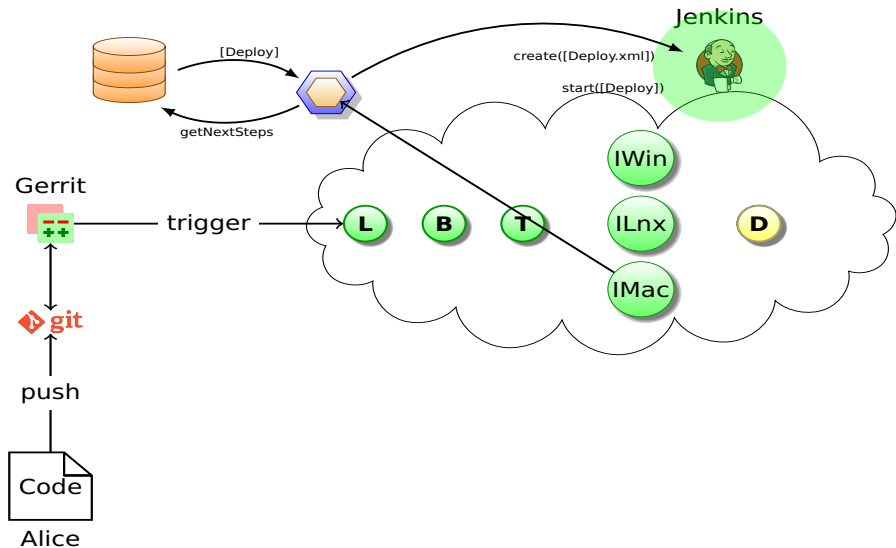
Abarbeitungsweise der neuen Infrastruktur



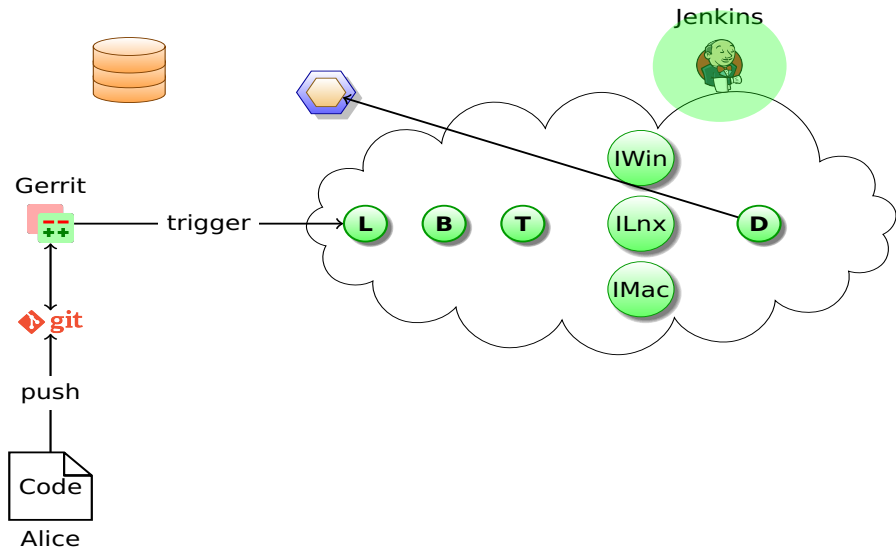
Abarbeitungsweise der neuen Infrastruktur



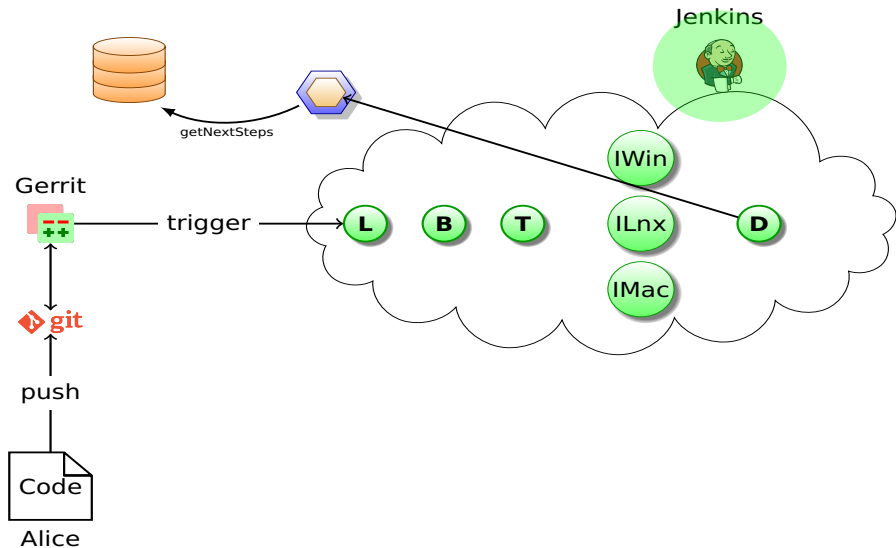
Abarbeitungsweise der neuen Infrastruktur



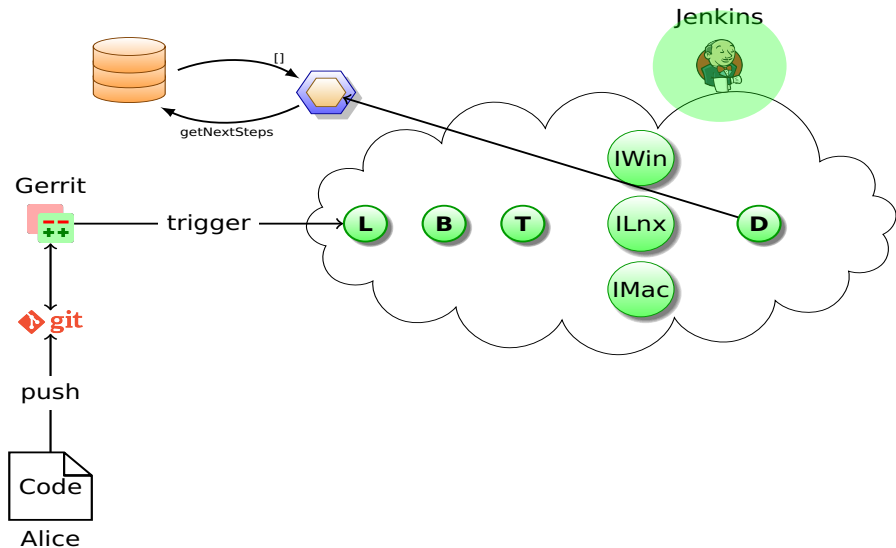
Abarbeitungsweise der neuen Infrastruktur

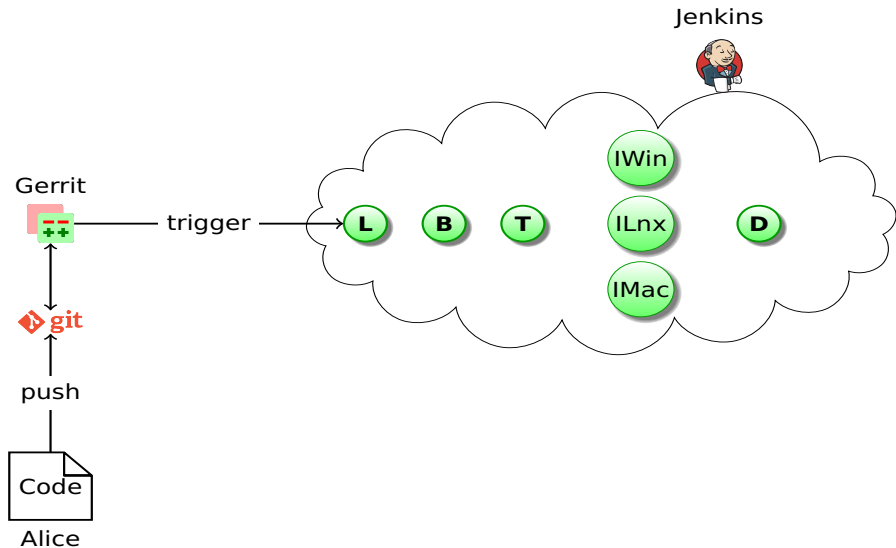


Abarbeitungsweise der neuen Infrastruktur



Abarbeitungsweise der neuen Infrastruktur





Outline

Grundbegriffe

Aufgabenstellung

Darstellung der aktuellen Infrastruktur

Probleme

Anforderungen an eine Problemlösung

Ergebnisse

Darstellung der neuen Infrastruktur

Fazit

Bewertung der Ergebnisse

Ausblick

- ▶ Entkopplung aller Systeme (Jenkins, DB, Konfigurationsmanager)
- ▶ Prinzipien für gute Wartbarkeit bestimmen
 - ▶ geringe Kopplung
 - ▶ hohe Kohäsion
 - ▶ konsistente Benennungen

Vor- und Nachteile der Lösung

- + leicht erweiterbar
- relativ unbekanntes Muster Hexagonale Architektur
- + zentraler Mechanismus zur Pipelineabarbeitung
- einmaliger Aufwand für die Einführung des Systems
- + leichte Einführung des Systems
 - ▶ bereits existierende Job-Templates können genutzt werden
- + Wartungsaufwand unabhängig von der Kundenanzahl

Outline

Grundbegriffe

Aufgabenstellung

Darstellung der aktuellen Infrastruktur

Probleme

Anforderungen an eine Problemlösung

Ergebnisse

Darstellung der neuen Infrastruktur

Fazit

Bewertung der Ergebnisse

Ausblick

- ▶ Entwicklung eines Dashboards
- ▶ Erweiterung der Vorbedingungslogik
- ▶ u.v.m.

- ▶ Jenkins-Logo
<https://wiki.jenkins-ci.org/download/attachments/2916393/logo.zip?version=3&modificationDate=1305324183000>
- ▶ Gerrit-Logo
<http://upload.wikimedia.org/wikipedia/mediawiki/a/a7/Gerrit.svg>
- ▶ Git-Logo
<http://git-scm.com/images/logos/downloads/Git-Logo-1788C.png>