

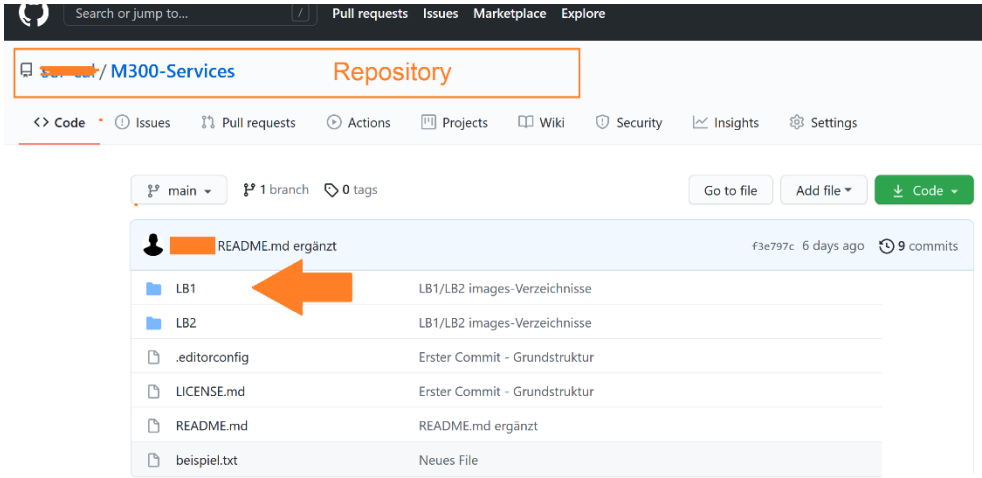
# 1 M300 LB1 (IaC) → **ST18a**

## (Infrastructure as Code – Hands-on Part1 )

### Handlungsziele:

- DVCS kennenlernen und anwenden
- Lab-Umgebungen / Services deklarativ aufsetzen und dokumentieren

### Formale Vorgaben

<b>Abgabetermin:</b>	<ul style="list-style-type: none"> <li>• <b>14. Juni 2021 10:00h.</b> (Danach nur noch Fernbewertung der Repos durch LP)</li> </ul>
<b>Speicherort und Abgabeobjekte</b>	<ul style="list-style-type: none"> <li>• Repository: <b>M300-Services</b> (Github)</li> <li>• Selbstdefinierte Ordner (Bilder nicht vergessen)</li> <li>• Inhaltsverzeichnis (siehe weiter unten)</li> <li>• Vagrantfile und entsprechende deklarativen Erklärungen</li> </ul> 
<b>Projektbeschreibung</b>	<ul style="list-style-type: none"> <li>• <b>Gem. Lernumgebung</b></li> </ul>
<b>Formalitäten</b>	<ul style="list-style-type: none"> <li>• <b>Eintrag des Repo-Links in Klassenliste → <a href="#">HIER</a></b> Projekt-Daten korrekt hinterlegt (Name, Git-Repository)</li> <li>• <b>Freigabe für Lehrperson auf Github</b></li> <li>• <b>Abgabe des ganzen Projektes als Zip-File <a href="#">HIER</a> (BSCW)</b></li> </ul>



## Auftrag – Infrastructure as Code (IaC, Kap. 10 / 20 / 25)

Sie erstellen - auf Basis von **VirtualBox/Vagrant** - ein selbst gewähltes **«Infrastructure as Code»** Projekt, indem sie einen Service oder Serverdienst **automatisieren**.

Teamarbeit ist erwünscht. Die Implementation des **IaC-Projekts** erfolgt hingegen **als Einzelarbeit**. Der erstellte Code sowie die gesamte Dokumentation wird versioniert, auf GitHub hinterlegt und der Lehrperson zugänglich gemacht (Lese-Rechte).

Das Internet ist eine zugelassene Ressource für solche Projekte. Entsprechend dürfen sie auch Codebeispiele aus dem Internet verwenden, sofern sie entsprechende Quellenangaben machen.

Der verwendete Code muss aber von ihnen vollständig dokumentiert sein. Das gilt auch für Code, welchen sie aus fremden Quellen verwenden.

**Das bedeutet sie können über den verwendeten Code Auskunft geben.**

## Anforderungen

- Wiederholbar und konsistent ausführbar auf eigenem Rechner
- Die Entwicklungsschritte des Codes und der Dokumentation sind in der Git History durch **regelmässige und dokumentierte** Commits nachvollziehbar
- Service / Dienst startet mit 'vagrant up' (ohne User-Interaktion)
- Service / Dienst weist dokumentierte Sicherheitsmerkmale auf
- Die Projektdokumentation erfolgt in Markdown

## Bewertung der Praxisarbeit → Mass innerhalb der Kompetenzmatrix

### Komplexität, Umfang, Funktionalität, Umsetzung

- Gem. Kompetenzmatrix. Grad der Umsetzung (Volle Punktzahl bei ausgezeichneter Arbeit)
- Service-Funktion mit mehreren VMs oder Services (Bsp. Webserver mit DB-Backend)
- Keine Interaktion nötig nach dem Start, Service ist nach «Vagrant up» verfügbar
- Korrekte Netzwerkkonfiguration wie Portforwarding / Bridge-Konfig etc...
- Installation oder Konfiguration von zusätzlich benötigter SW
- Service / Dienst weist Sicherheitsmerkmale auf
- Der Fortschritt ist in der Git-History nachvollziehbar



## **Bewertung der Dokumentation → Mass innerhalb der Kompetenzmatrix**

### **Komplexität, Umfang, Funktionalität, Umsetzung**

- Es wird eine technische Dokumentation über ihr IaC Projekt erwartet. Sie beschreiben, wie der Service funktioniert, was die Rahmenbedingungen zum Service sind und wie sie den Service auf korrekte Funktion testen. Eine gut strukturierte und mit Grafiken ergänzte Dokumentation gibt Auskunft über verwendete Server, Applikationen, Netzwerkkonfigurationen.
- **Inhaltsverzeichnis (Beispiel)**
  - Inhaltsverzeichnis
  - Einleitung
  - Service-Aufbau, Setting (Struktur, Deklarativer Ablauf)
  - Umsetzung
  - Testing
  - Quellen
- **Markdown**
  - Verwendung von verschiedenen Markdown-Elementen (etwa Tabellen, Links Bilder etc...)
- **Darstellung**
  - Ansprechend
  - Nachvollziehbar
  - korrekt



# Bewertungsraster

## Ticketsystem

Abgabedatum:	Klasse:		Team:
Teilnehmende:	Nachname	Nachname:	Nachname:

Bewertung	Punkte
Setup Umgebung, Lernumgebung, Tools <span style="float: right;">★★</span> <b>1P</b> Virtualbx / <b>1P</b> Vagrant / <b>1P</b> Git-Setup / <b>1P</b> SSH-Keys / <b>1P</b> Repo-Setup / <b>1P</b> Editor	
Technische Doku (Struktur, Tiefe, Gestaltung, Formatierung, Nachvollziehbarkeit) <span style="float: right;">★★</span> <b>6P</b> Excellent / <b>5P</b> Sehr gut (überdurchschnittlich) / <b>3P</b> Gut / <b>2P</b> Ausreichend / <b>1P</b> Vorhanden	
Entwicklung des Repositories (Regelmässigkeit und Umfang der Updates/Commits) <span style="float: right;">★</span> Umfang Einträge + Commits des Repos → <b>3P</b> Regelm. / <b>2P</b> mehrmals wöchentl. / <b>1P</b> wöchentl.)	
Grund-Service, Funktionalität, Dokumentation (Eigen- oder Ergänzungsleistung!) <span style="float: right;">★</span> <b>3P</b> Excellent (Persist., deklarativ, viel Eigenleistung) / <b>2P</b> Gut (Niveau Vorlage) / <b>1P</b> Eigenl. vorh.	
Ergänzende Services, Funktionalität, Dokumentation <span style="float: right;">★</span> <b>3P</b> Excellent (Persist., deklarativ, viel Eigenleistung) / <b>2P</b> Gut (Niveau Vorlage) / <b>1P</b> Eigenl. vorh	
Engagement, Haltung, Professionalität, Kommunikation: (Max. 2 P.) <span style="float: right;">★★</span> <b>2P</b> Alle Punkte erfüllt	
Präsentation, Live-Demo & Quellenangaben: (Max. 2 P.) <span style="float: right;">★★</span> <b>2P</b> Projekt kundengerecht umgesetzt, keine erkennb. Fehler / <b>1P</b> Projekt umgesetzt, kleine Mängel	
<b>Total Punkte</b> (Max 25)	
<b>Note</b>	

Umrechnungshilfe Punkte – Note											
Punkte	0	2	4	8	11	13	14	16	19	22	25
Note	1	1,5	2	2,5	3	3,5	4	4,5	5	5,5	6

<b>Bemerkungen Expert/in:</b>
-------------------------------