

Integration workshop [16,17,18,19,20] September, 2014

Thirumal Ravula, Virtual Labs Engineering and Architecture Division (VLEAD)

2014-09-16

Contents

1	Integration	2
1.1	Overview	2
1.2	Different View	2
1.3	Current Status with Versioning of Labs.	3
1.3.1	Total Labs and their type	3
1.3.2	Status of sources of 171 simulation labs	3
2	Transformation of a Lab from Source to Application	3
2.1	Sources on the github	3
2.2	Creation of container	3
2.3	Automate this process	3
2.4	Deploy the lab	4
2.5	Pictorial overview of the deployment process.	4
3	Advantages of autodeployment	4
3.1	At a glance	4
3.2	Services at Play	4

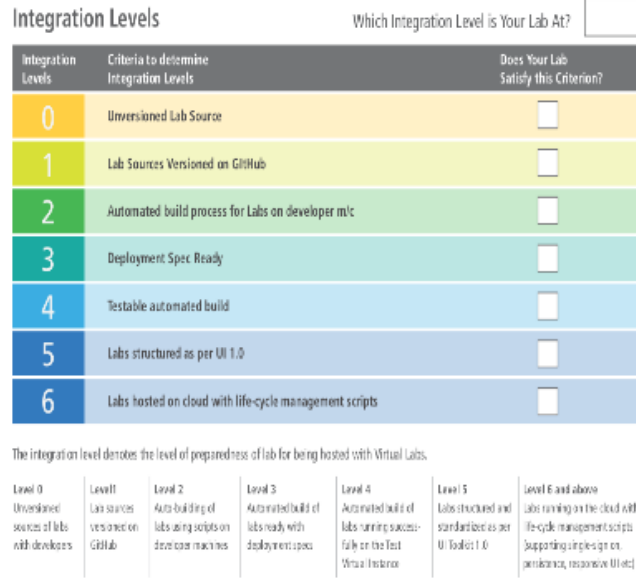


Figure 1: Integration Levels

1 Integration

1.1 Overview

1.2 Different View

Level	capability
Level 0	Unversioned - Lab sources with the developer.
Level 1	Versioned - Lab sources versioned in github.
Level 2	Lab is built manually and the lab is deployed in the developer environment
Level 3	Lab is built automatically and the lab is deployed in the developer environment
Level 4	Lab deployment specification is developed and the lab is auto deployed in the developer environment
Level 5	Labs structured and standarized as per UI Toolkit 1.0 and auto deployed in the developer environment.
Level 6 & above	Lab deployed on the cloud lifecycle scripts, with analytics, single sign on, persistence, responsive UI etc.

1.3 Current Status with Versioning of Labs.

1.3.1 Total Labs and their type

Lab type	No.
Remote Triggered	22
Simulation	171
Total	193

1.3.2 Status of sources of 171 simulation labs

Versioning status	No.
versioned with VLEAD	79
yet to be versioned with VLEAD	92
Total	171

2 Transformation of a Lab from Source to Application

2.1 Sources on the github

- Create a github handle.
- Create a repository and clone it on your machine
- Copy all the lab sources to this repository and push them.

2.2 Creation of container

- Create a container and start it.
- Deploy the lab on this container manually.

2.3 Automate this process

- Restructure your lab into src, scripts, README.md folders.
- Write makefile that builds the lab nad moves all files that are required to run the lab into build folder.
- Write other scripts if necessary to automate without any manual intervention.
- Create a labspec.json in scripts folder and put all the steps into this.

- Push all your changes to your lab repository on github.

2.4 Deploy the lab

- Clone OVPL from gitHub.
- Run **make** from the src folder. This will ensure all the services are running.
- Provide the lab id and url of the lab as inputs to the webpage at <http://localhost:8080>
- Follow the logs and ensure the lab is successfully deployed.

2.5 Pictorial overview of the deployment process.

3 Advantages of autodeployment

3.1 At a glance

Decoupling Development of a lab is decoupled from hosting a lab.

Scaling A lab can scaled when required.

Continous Integration Provision to set many gate keepers before a lab can be deployed.

3.2 Services at Play

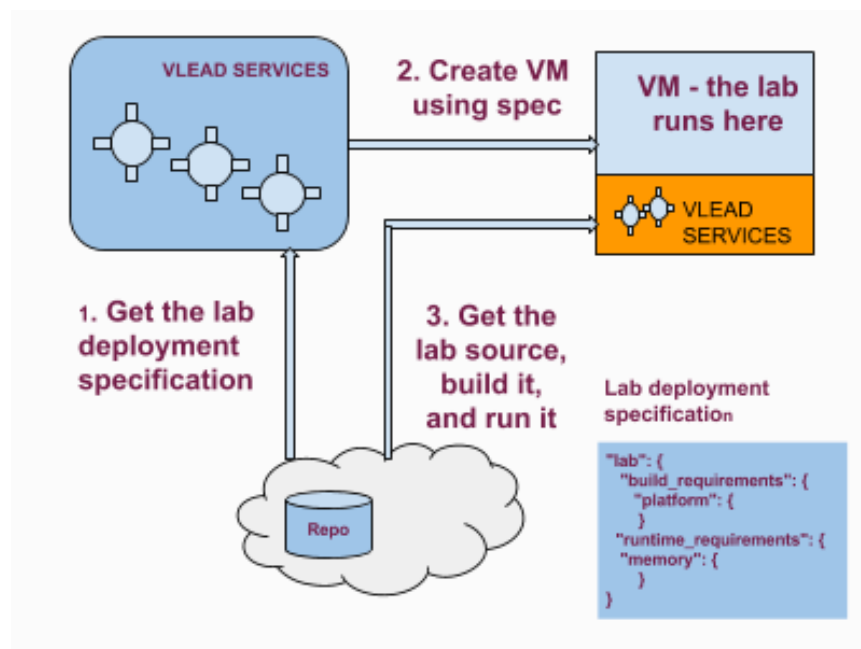


Figure 2: overview of deployment process Services at Play