

Azure Replicable CI / CD Locally

GitHub Project Repo: [beamerAzureCiCdLocal](#)

Author: Athanasios Garyfalos

Date: December 16, 2021

Introduction

Why to Replicate CI / CD
Locally

Software Development

Overall Problems
Simple Solution

Implementation

CI-CD Local

CI-CD Kubernetes

Demo

Demo CI / CD

Summary

Key points repetition

Bibliography

Introduction

Why to Replicate CI / CD Locally

Software Development

Implementation

Demo

Summary

Bibliography

High level description of the problem

- ▶ It works locally why not remotely?
- ▶ It works on my computer why not on yours?
- ▶ Common Wrong Assumptions:
 - ▶ Running same package(s) version
 - ▶ Using the same OS
 - ▶ Using the same hardware
 - ▶ No difference between local and remote systems
- ▶ Is there a solution?

Not replicable

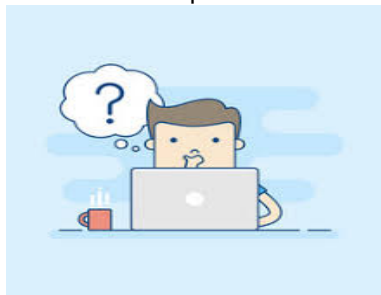


Figure 1: Problem Overview

High level description of the problem

- ▶ It works locally why not remotely?
- ▶ It works on my computer why not on yours?
- ▶ **Common Wrong Assumptions:**
 - ▶ Running same package(s) version
 - ▶ No difference on installation process
 - ▶ No difference between Operating Systems
- ▶ Is there a solution?

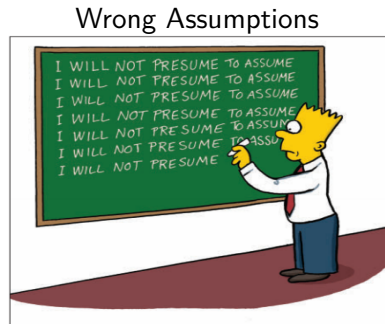


Figure 1: Problem Overview

High level description of the problem

- ▶ It works locally why not remotely?
- ▶ It works on my computer why not on yours?
- ▶ Common Wrong Assumptions:
 - ▶ Running same package(s) version
 - ▶ No difference on installation process
 - ▶ No difference between Operating Systems
- ▶ Is there a solution?



Figure 1: Problem Overview

High level description of the problem

- ▶ It works locally why not remotely?
- ▶ It works on my computer why not on yours?
- ▶ Common Wrong Assumptions:
 - ▶ Running same package(s) version
 - ▶ No difference on installation process
 - ▶ No difference between Operating Systems
- ▶ Is there a solution?

Is there a difference in compilers?



Figure 1: Problem Overview

High level description of the problem

- ▶ It works locally why not remotely?
- ▶ It works on my computer why not on yours?
- ▶ Common Wrong Assumptions:
 - ▶ Running same package(s) version
 - ▶ No difference on installation process
 - ▶ **No difference between Operating Systems**
- ▶ Is there a solution?



Figure 1: Problem Overview

High level description of the problem

- ▶ It works locally why not remotely?
- ▶ It works on my computer why not on yours?
- ▶ Common Wrong Assumptions:
 - ▶ Running same package(s) version
 - ▶ No difference on installation process
 - ▶ No difference between Operating Systems



Figure 1: Problem Overview

- ▶ Is there a solution?

Introduction

Software Development

Overall Problems

Simple Solution

Implementation

Demo

Summary

Bibliography

Cross OS Programming Languages

Python



Figure 2: Problem Overview

▶ Python:

▶ Python2

▶ Python3

▶ Node.js

▶ Vending

▶ Camel

▶ Python 2.7

▶ Matplotlib + LTP

▶ Java:

▶ Oracle

▶ OpenJDK

▶ AdoptOpenJDK

▶ Cross Platform OS

Cross OS Programming Languages

Python2 Deprecated



Figure 2: Problem Overview

▶ Python:

▶ Python2

▶ Python3

▶ Node.js

▶ Windows

▶ Linux

▶ MacOS

▶ Raspberry Pi

▶ Docker

▶ Java:

▶ Windows

▶ Linux

▶ MacOS

▶ Raspberry Pi

▶ Cross Platform OS

Cross OS Programming Languages

Python3 Active



Figure 2: Problem Overview

- ▶ Python:
 - ▶ Python2
 - ▶ Python3
- ▶ Node.js
 - ▶ Pending
 - ▶ C++
 - ▶ C
 - ▶ Rust
 - ▶ Haskell
 - ▶ Go
- ▶ Java:
 - ▶ C#
 - ▶ Objective-C
 - ▶ Swift
 - ▶ Kotlin
 - ▶ Kotlin Multiplatform
- ▶ Cross Platform OS

Cross OS Programming Languages

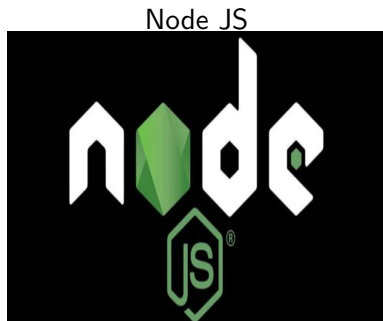


Figure 2: Problem Overview

► Python:

- Python2
- Python3

► Node.js

- Pending
- Current
- Active LTS
- Maintenance LTS

► Java:

- Oracle JDK
- OpenJDK
- AdoptOpenJDK

► Cross Platform OS

Cross OS Programming Languages

Under Development

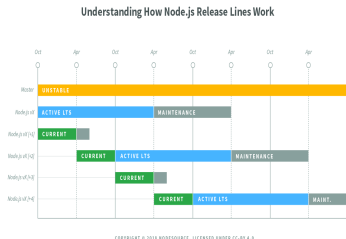


Figure 2: Problem Overview

- ▶ Python:
 - ▶ Python2
 - ▶ Python3

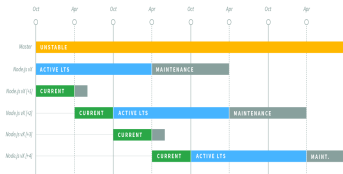
- ▶ Node.js
 - ▶ Pending
 - ▶ Current
 - ▶ Active LTS
 - ▶ Maintenance LTS

- ▶ Java:
 - ▶ Oracle JDK
 - ▶ OpenJDK
 - ▶ AdoptOpenJDK
 - ▶ Amazon Corretto
- ▶ Cross Platform OS

Cross OS Programming Languages

Current Latest Release

Understanding How Node.js Release Lines Work



COPYRIGHT © 2018 NODESOURCE, LICENSED UNDER CC-BY 4.0

Figure 2: Problem Overview

- ▶ Python:
 - ▶ Python2
 - ▶ Python3
- ▶ Node.js
 - ▶ Pending
 - ▶ **Current**
 - ▶ Active LTS
 - ▶ Maintenance LTS
- ▶ Java:
 - ▶ Java 8
 - ▶ Java 9
 - ▶ Java 10
 - ▶ Java 11
 - ▶ Java 12
 - ▶ Java 13
 - ▶ Java 14
 - ▶ Java 15
 - ▶ Java 16
 - ▶ Java 17
 - ▶ Java 18
 - ▶ Java 19
 - ▶ Java 20
 - ▶ Java 21
 - ▶ Java 22
 - ▶ Java 23
 - ▶ Java 24
 - ▶ Java 25
 - ▶ Java 26
 - ▶ Java 27
 - ▶ Java 28
 - ▶ Java 29
 - ▶ Java 30
 - ▶ Java 31
 - ▶ Java 32
 - ▶ Java 33
 - ▶ Java 34
 - ▶ Java 35
 - ▶ Java 36
 - ▶ Java 37
 - ▶ Java 38
 - ▶ Java 39
 - ▶ Java 40
 - ▶ Java 41
 - ▶ Java 42
 - ▶ Java 43
 - ▶ Java 44
 - ▶ Java 45
 - ▶ Java 46
 - ▶ Java 47
 - ▶ Java 48
 - ▶ Java 49
 - ▶ Java 50
 - ▶ Java 51
 - ▶ Java 52
 - ▶ Java 53
 - ▶ Java 54
 - ▶ Java 55
 - ▶ Java 56
 - ▶ Java 57
 - ▶ Java 58
 - ▶ Java 59
 - ▶ Java 60
 - ▶ Java 61
 - ▶ Java 62
 - ▶ Java 63
 - ▶ Java 64
 - ▶ Java 65
 - ▶ Java 66
 - ▶ Java 67
 - ▶ Java 68
 - ▶ Java 69
 - ▶ Java 70
 - ▶ Java 71
 - ▶ Java 72
 - ▶ Java 73
 - ▶ Java 74
 - ▶ Java 75
 - ▶ Java 76
 - ▶ Java 77
 - ▶ Java 78
 - ▶ Java 79
 - ▶ Java 80
 - ▶ Java 81
 - ▶ Java 82
 - ▶ Java 83
 - ▶ Java 84
 - ▶ Java 85
 - ▶ Java 86
 - ▶ Java 87
 - ▶ Java 88
 - ▶ Java 89
 - ▶ Java 90
 - ▶ Java 91
 - ▶ Java 92
 - ▶ Java 93
 - ▶ Java 94
 - ▶ Java 95
 - ▶ Java 96
 - ▶ Java 97
 - ▶ Java 98
 - ▶ Java 99
 - ▶ Java 100
- ▶ Cross Platform OS

Cross OS Programming Languages

Active Long Term Support

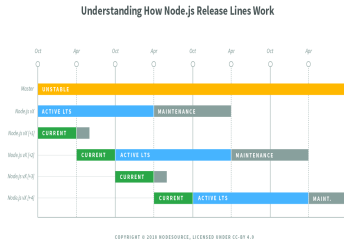


Figure 2: Problem Overview

- ▶ Python:
 - ▶ Python2
 - ▶ Python3
- ▶ Node.js
 - ▶ Pending
 - ▶ Current
 - ▶ **Active LTS**
 - ▶ Maintenance LTS
- ▶ Java:

Cross OS Programming Languages

Maintenance Long Term Support

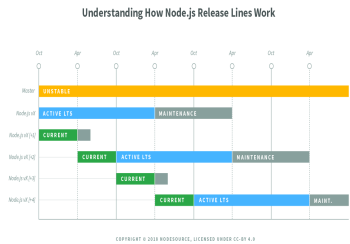


Figure 2: Problem Overview

- ▶ Python:
 - ▶ Python2
 - ▶ Python3
- ▶ Node.js
 - ▶ Pending
 - ▶ Current
 - ▶ Active LTS
 - ▶ Maintenance LTS

- ▶ Java:
 - ▶ Oracle
 - ▶ IBM
 - ▶ Red Hat
- ▶ Cross Platform OS

Cross OS Programming Languages

Tons of versions



Figure 2: Problem Overview

- ▶ Python:
 - ▶ Python2
 - ▶ Python3
- ▶ Node.js
 - ▶ Pending
 - ▶ Current
 - ▶ Active LTS
 - ▶ Maintenance LTS
- ▶ Java:
 - ▶ Oracle
 - ▶ OpenJDK
 - ▶ AdoptOpenJDK
- ▶ Cross Platform OS

Cross OS Programming Languages



Figure 2: Problem Overview

- ▶ Python:
 - ▶ Python2
 - ▶ Python3
- ▶ Node.js
 - ▶ Pending
 - ▶ Current
 - ▶ Active LTS
 - ▶ Maintenance LTS
- ▶ Java:
 - ▶ Oracle
 - ▶ OpenJDK
 - ▶ AdoptOpenJDK
- ▶ Cross Platform OS

Cross OS Programming Languages

Java OpenJDK



Figure 2: Problem Overview

- ▶ Python:
 - ▶ Python2
 - ▶ Python3
- ▶ Node.js
 - ▶ Pending
 - ▶ Current
 - ▶ Active LTS
 - ▶ Maintenance LTS
- ▶ Java:
 - ▶ Oracle
 - ▶ **OpenJDK**
 - ▶ AdoptOpenJDK
- ▶ Cross Platform OS

Cross OS Programming Languages

Maintenance Long Term
Support (LTS)



Figure 2: Problem Overview

- ▶ Python:
 - ▶ Python2
 - ▶ Python3
- ▶ Node.js
 - ▶ Pending
 - ▶ Current
 - ▶ Active LTS
 - ▶ Maintenance LTS
- ▶ Java:
 - ▶ Oracle
 - ▶ OpenJDK
 - ▶ AdoptOpenJDK

▶ Cross Platform OS

Cross OS Programming Languages

Different Installation Process

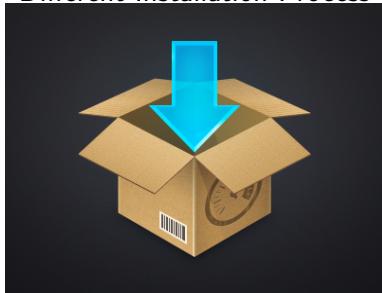


Figure 2: Problem Overview

- ▶ Python:
 - ▶ Python2
 - ▶ Python3
- ▶ Node.js
 - ▶ Pending
 - ▶ Current
 - ▶ Active LTS
 - ▶ Maintenance LTS
- ▶ Java:
 - ▶ Oracle
 - ▶ OpenJDK
 - ▶ AdoptOpenJDK
- ▶ Cross Platform OS

Cross OS Programming Languages

Bundle package



Figure 2: Problem Overview

- ▶ Python:
 - ▶ Python2
 - ▶ Python3
- ▶ Node.js
 - ▶ Pending
 - ▶ Current
 - ▶ Active LTS
 - ▶ Maintenance LTS
- ▶ Java:
 - ▶ Oracle
 - ▶ OpenJDK
 - ▶ AdoptOpenJDK
- ▶ Cross Platform OS

Cross OS Programming Languages

Source Code



Figure 2: Problem Overview

- ▶ Python:
 - ▶ Python2
 - ▶ Python3
- ▶ Node.js
 - ▶ Pending
 - ▶ Current
 - ▶ Active LTS
 - ▶ Maintenance LTS
- ▶ Java:
 - ▶ Oracle
 - ▶ OpenJDK
 - ▶ AdoptOpenJDK
- ▶ Cross Platform OS

Introduction

Software Development

Overall Problems

Simple Solution

Implementation

Demo

Summary

Bibliography

Impossible vs Possible

Impossible Solution

- ▶ Everyone needs to have exactly same OS.
- ▶ Everyone needs to install packages the same way.
- ▶ Everyone must update the OS and package at the same time.
- ▶ To what extent this is possible and practical?

Impossible vs Possible

Impossible Solution

- ▶ Everyone needs to have exactly same **OS**.
- ▶ Everyone needs to install packages the same way.
- ▶ Everyone must update the OS and package at the same time.
- ▶ To what extent this is possible and practical?

Impossible vs Possible

Impossible Solution

- ▶ Everyone needs to have exactly same **OS**.
- ▶ Everyone needs to install packages **the same way**.
- ▶ **Everyone must update the OS and package at the same time.**
- ▶ To what extent this is possible and practical?

Impossible vs Possible

Impossible Solution

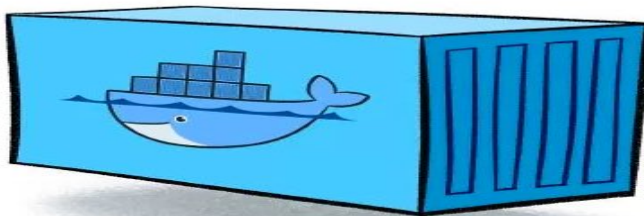
- ▶ Everyone needs to have exactly same **OS**.
- ▶ Everyone needs to install packages **the same way**.
- ▶ Everyone must update the OS and package **at the same time**.
- ▶ **To what extent this is possible and practical?**

Impossible vs Possible

Impossible Solution

- ▶ Everyone needs to have exactly same **OS**.
- ▶ Everyone needs to install packages **the same way**.
- ▶ Everyone must update the OS and package **at the same time**.
- ▶ To what extent this is **possible and practical**?

Containers again and again



CI-CD Kubernetes

Introduction

Demo

Software Development

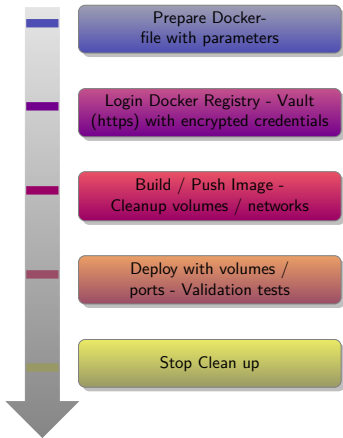
Summary

Implementation

Bibliography

CI-CD Local

Local Procedure



► Dockerfile (template).

► Vault (https).

► Any socket.

► Dockerfile (template).

► Dockerfile (template).

► Dockerfile (template).

► Dockerfile (template).

► Dockerfile (template).

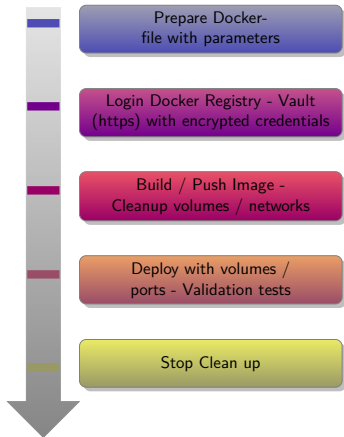
► Dockerfile (template).

► Deployment (volume).

► Validation (tests).

► Stop, Cleanup.

Local Procedure



► Dockerfile (template).

► Vault (<https>).

► Any socket.

► Azure Registry.

► Build Docker image.

► Push image.

► Login Docker.

► Pull image.

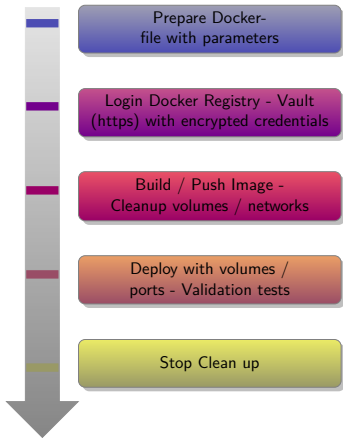
► Build image.

► Deployment (volume).

► Validation (tests).

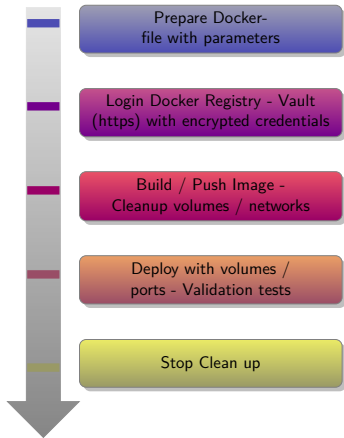
► Stop, Cleanup.

Local Procedure



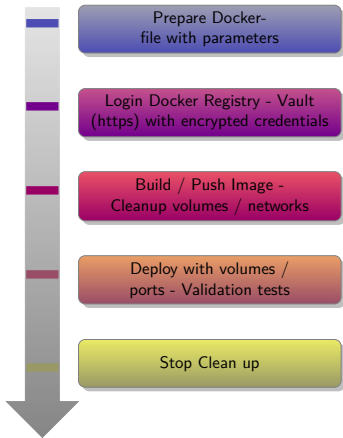
- ▶ Dockerfile (template).
- ▶ Vault (https).
- ▶ Any socket.
 - ▶ Azzure Registry.
 - ▶ Build Dockerfile.
 - ▶ Push Image.
 - ▶ Logout Azzure.
 - ▶ Prune everything.
 - ▶ Raise error (if).
- ▶ Deployment (volume).
- ▶ Validation (tests).
- ▶ Stop, Cleanup.

Local Procedure



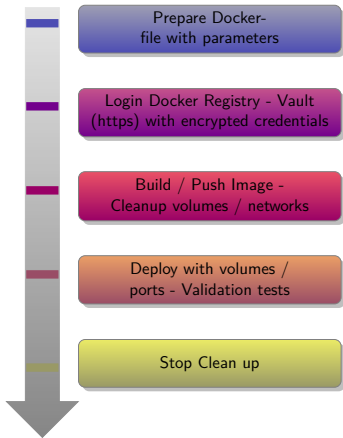
- ▶ Dockerfile (template).
- ▶ Vault (https).
- ▶ Any socket.
 - ▶ **Azzure Registry.**
 - ▶ Build Dockerfile.
 - ▶ Push Image.
 - ▶ Logout Azzure.
 - ▶ Prune everything.
 - ▶ Raise error (if).
- ▶ Deployment (volume).
- ▶ Validation (tests).
- ▶ Stop, Cleanup.

Local Procedure



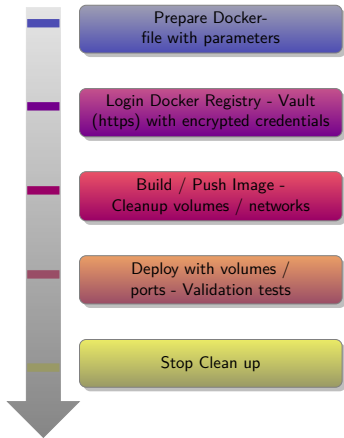
- ▶ Dockerfile (template).
- ▶ Vault (https).
- ▶ Any socket.
 - ▶ Azure Registry.
 - ▶ **Build Dockerfile.**
 - ▶ Push Image.
 - ▶ Logout Azure.
 - ▶ Prune everything.
 - ▶ Raise error (if).
- ▶ Deployment (volume).
- ▶ Validation (tests).
- ▶ Stop, Cleanup.

Local Procedure



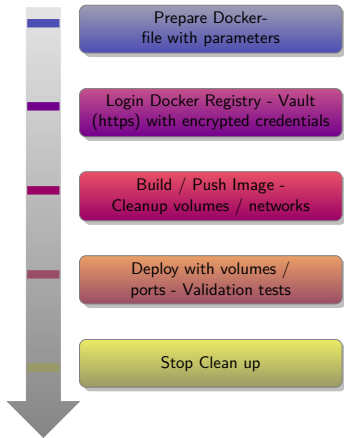
- ▶ Dockerfile (template).
- ▶ Vault (https).
- ▶ Any socket.
 - ▶ Azzure Registry.
 - ▶ Build Dockerfile.
 - ▶ **Push Image.**
 - ▶ Logout Azzure.
 - ▶ Prune everything.
 - ▶ Raise error (if).
- ▶ Deployment (volume).
- ▶ Validation (tests).
- ▶ Stop, Cleanup.

Local Procedure



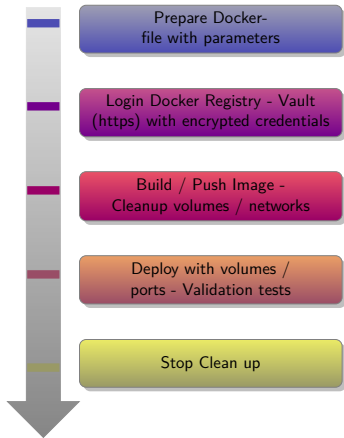
- ▶ Dockerfile (template).
- ▶ Vault (https).
- ▶ Any socket.
 - ▶ Azzure Registry.
 - ▶ Build Dockerfile.
 - ▶ Push Image.
 - ▶ Logout Azzure.
 - ▶ Prune everything.
 - ▶ Raise error (if).
- ▶ Deployment (volume).
- ▶ Validation (tests).
- ▶ Stop, Cleanup.

Local Procedure



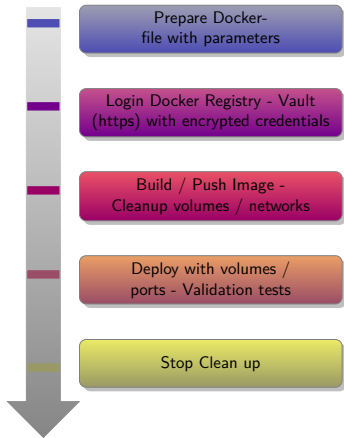
- ▶ Dockerfile (template).
- ▶ Vault (https).
- ▶ Any socket.
 - ▶ Azzure Registry.
 - ▶ Build Dockerfile.
 - ▶ Push Image.
 - ▶ Logout Azzure.
 - ▶ **Prune everything.**
 - ▶ Raise error (if).
- ▶ Deployment (volume).
- ▶ Validation (tests).
- ▶ Stop, Cleanup.

Local Procedure



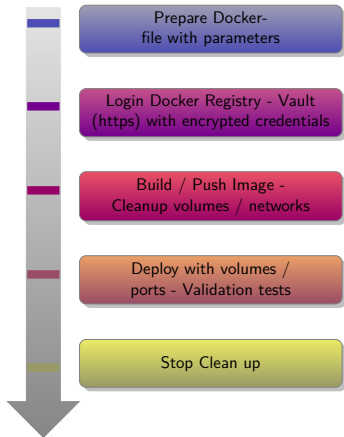
- ▶ Dockerfile (template).
- ▶ Vault (https).
- ▶ Any socket.
 - ▶ Azzure Registry.
 - ▶ Build Dockerfile.
 - ▶ Push Image.
 - ▶ Logout Azzure.
 - ▶ Prune everything.
 - ▶ Raise error (if).
- ▶ Deployment (volume).
- ▶ Validation (tests).
- ▶ Stop, Cleanup.

Local Procedure



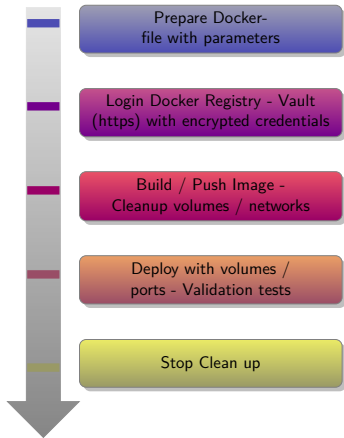
- ▶ Dockerfile (template).
- ▶ Vault (https).
- ▶ Any socket.
 - ▶ Azzure Registry.
 - ▶ Build Dockerfile.
 - ▶ Push Image.
 - ▶ Logout Azzure.
 - ▶ Prune everything.
 - ▶ Raise error (if).
- ▶ Deployment (volume).
- ▶ Validation (tests).
- ▶ Stop, Cleanup.

Local Procedure



- ▶ Dockerfile (template).
- ▶ Vault (https).
- ▶ Any socket.
 - ▶ Azzure Registry.
 - ▶ Build Dockerfile.
 - ▶ Push Image.
 - ▶ Logout Azzure.
 - ▶ Prune everything.
 - ▶ Raise error (if).
- ▶ Deployment (volume).
- ▶ Validation (tests).
- ▶ Stop, Cleanup.

Local Procedure



- ▶ Dockerfile (template).
- ▶ Vault (https).
- ▶ Any socket.
 - ▶ Azzure Registry.
 - ▶ Build Dockerfile.
 - ▶ Push Image.
 - ▶ Logout Azzure.
 - ▶ Prune everything.
 - ▶ Raise error (if).
- ▶ Deployment (volume).
- ▶ Validation (tests).
- ▶ Stop, Cleanup.

CI-CD Kubernetes

Introduction

Demo

Software Development

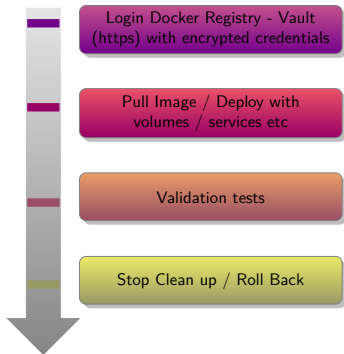
Summary

Implementation

Bibliography

CI-CD Local

Local k8s Deployment Procedure



► **Vault ([https](https://vaultproject.io/)).**

► Any socket.

► Azure Registry.

► Pull Image.

► Deployment (volume).

► Logout Azure.

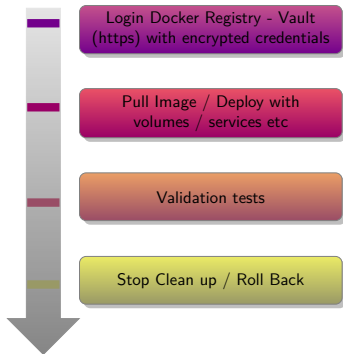
► Validation (tests).

► Roll back (if).

► Stop, Cleanup.

► Roll Back.

Local k8s Deployment Procedure



▶ Vault (https).

▶ Any socket.

▶ Azzure Registry.

▶ Pull Image.

▶ Deployment (volume).

▶ Logout Azzure.

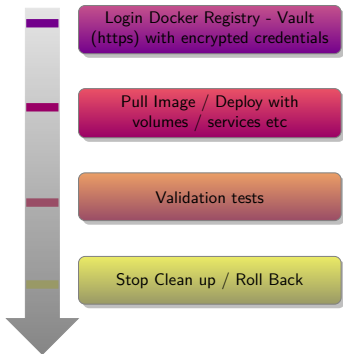
▶ Validation (tests).

▶ Roll back (if).

▶ Stop, Clean up.

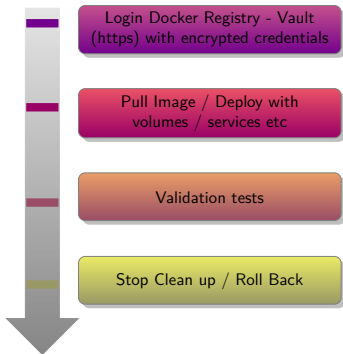
▶ Pull Image.

Local k8s Deployment Procedure



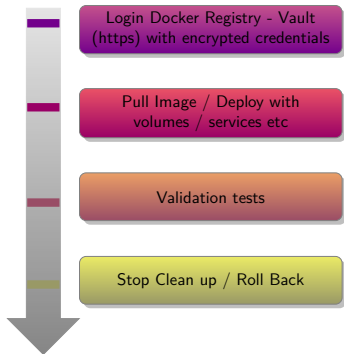
- ▶ Vault (https).
- ▶ Any socket.
 - ▶ Azzure Registry.
 - ▶ Pull Image.
- ▶ Deployment (volume).
- ▶ Logout Azzure.
- ▶ Validation (tests).

Local k8s Deployment Procedure



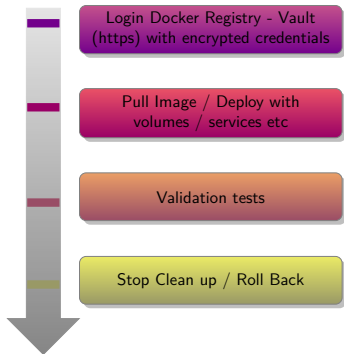
- ▶ Vault (https).
- ▶ Any socket.
 - ▶ Azzure Registry.
 - ▶ Pull Image.
- ▶ Deployment (volume).
- ▶ Logout Azzure.
- ▶ Validation (tests).

Local k8s Deployment Procedure



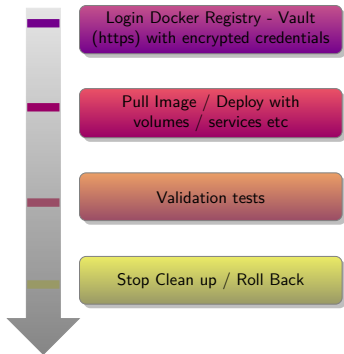
- ▶ Vault (https).
- ▶ Any socket.
 - ▶ Azzure Registry.
 - ▶ Pull Image.
- ▶ Deployment (volume).
- ▶ Logout Azzure.
- ▶ Validation (tests).

Local k8s Deployment Procedure



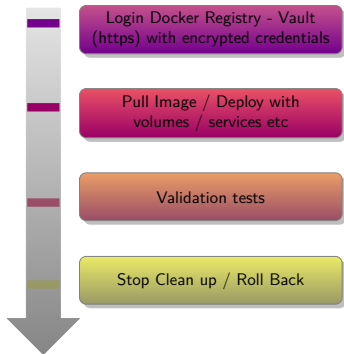
- ▶ Vault (https).
- ▶ Any socket.
 - ▶ Azzure Registry.
 - ▶ Pull Image.
- ▶ Deployment (volume).
- ▶ Logout Azzure.
- ▶ Validation (tests).
 - ▶ Raise error (if).
 - ▶ Stop Docker.
 - ▶ Stop k8s.

Local k8s Deployment Procedure



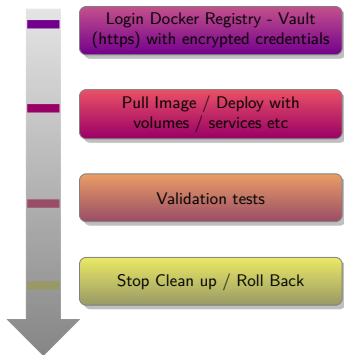
- ▶ Vault (https).
- ▶ Any socket.
 - ▶ Azzure Registry.
 - ▶ Pull Image.
- ▶ Deployment (volume).
- ▶ Logout Azzure.
- ▶ Validation (tests).
 - ▶ Raise error (if).
 - ▶ Stop, Cleanup.
 - ▶ Roll back.

Local k8s Deployment Procedure



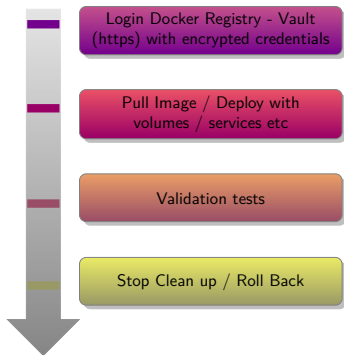
- ▶ Vault (https).
- ▶ Any socket.
 - ▶ Azzure Registry.
 - ▶ Pull Image.
- ▶ Deployment (volume).
- ▶ Logout Azzure.
- ▶ Validation (tests).
 - ▶ Raise error (if).
 - ▶ Stop, Cleanup.
 - ▶ Roll back.

Local k8s Deployment Procedure



- ▶ Vault (https).
- ▶ Any socket.
 - ▶ Azzure Registry.
 - ▶ Pull Image.
- ▶ Deployment (volume).
- ▶ Logout Azzure.
- ▶ Validation (tests).
 - ▶ Raise error (if).
 - ▶ **Stop, Cleanup.**
 - ▶ Roll back.

Local k8s Deployment Procedure



- ▶ Vault (https).
- ▶ Any socket.
 - ▶ Azzure Registry.
 - ▶ Pull Image.
- ▶ Deployment (volume).
- ▶ Logout Azzure.
- ▶ Validation (tests).
 - ▶ Raise error (if).
 - ▶ Stop, Cleanup.
 - ▶ Roll back.

Demo

Demo CI / CD

Introduction

Software Development

Implementation

Summary

Bibliography

Coming up: Demo

Demo

Introduction

Summary

Key points repetition

Software Development

Implementation

Bibliography

Summary

- ▶ In section “Introduction” page “2” Common issues, non replicable.
- ▶ In section “Software Development” page “3” Variety of programming languages and the problems.
- ▶ In section “Implementation” page “4” CI / CD Flow Build / Deploy / Test.

Extra Notes

- ▶ Demo on CI / CD build / deploy / validation and error handling cases.
- ▶ Both the CI / CD and Kubernetes project are provided as open source contribution. The Presentation was written in \LaTeX

Summary

- ▶ In section “Introduction” page “2” Common issues, non replicable.
- ▶ In section “Software Development” page “3” Variety of programming languages and the problems.
- ▶ In section “Implementation” page “4” CI / CD Flow Build / Deploy / Test.

Extra Notes

- ▶ Demo on CI / CD build / deploy / validation and error handling cases.
- ▶ Both the CI / CD and Kubernetes project are provided as open source contribution. The Presentation was written in **L^AT_EX**

Summary

- ▶ In section “Introduction” page “2” Common issues, non replicable.
- ▶ In section “Software Development” page “3” Variety of programming languages and the problems.
- ▶ In section “Implementation” page “4” CI / CD Flow Build / Deploy / Test.

Extra Notes

- ▶ Demo on CI / CD build / deploy / validation and error handling cases.
- ▶ Both the CI / CD and Kubernetes project are provided as open source contribution. The Presentation was written in \LaTeX

Summary

- ▶ In section “Introduction” page “2” Common issues, non replicable.
- ▶ In section “Software Development” page “3” Variety of programming languages and the problems.
- ▶ In section “Implementation” page “4” CI / CD Flow Build / Deploy / Test.

Extra Notes

- ▶ Demo on CI / CD build / deploy / validation and error handling cases.
- ▶ Both the CI / CD and Kubernetes project are provided as open source contribution. The Presentation was written in **L^AT_EX**

Summary

- ▶ In section “Introduction” page “2” Common issues, non replicable.
- ▶ In section “Software Development” page “3” Variety of programming languages and the problems.
- ▶ In section “Implementation” page “4” CI / CD Flow Build / Deploy / Test.

Extra Notes

- ▶ Demo on CI / CD build / deploy / validation and error handling cases.
- ▶ Both the CI / CD and Kubernetes project are provided as open source contribution. The Presentation was written in **L^AT_EX**

Summary

- ▶ In section “Introduction” page “2” Common issues, non replicable.
- ▶ In section “Software Development” page “3” Variety of programming languages and the problems.
- ▶ In section “Implementation” page “4” CI / CD Flow Build / Deploy / Test.

Extra Notes

- ▶ Demo on CI / CD build / deploy / validation and error handling cases.
- ▶ Both the CI / CD and Kubernetes project are provided as open source contribution. The Presentation was written in **L^AT_EX**

Coming up: Q & A

References I



GNU LESSER GENERAL PUBLIC LICENSE

GNU Operating System

available at <https://www.gnu.org/licenses/lgpl.html>.