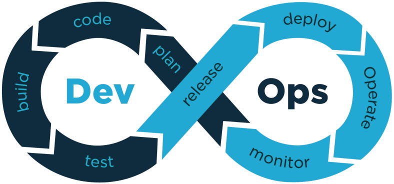


Azure Replicable CI / CD Locally

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
  - No difference on installation process
  - No difference between Operating Systems
- Is there a solution?

Local Vs Remote

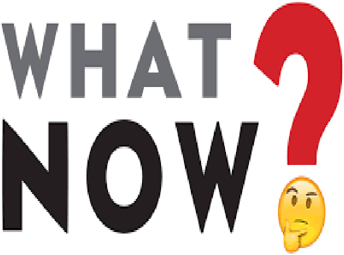


Figure 1: Problem Overview

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?

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?

Wrong Assumptions

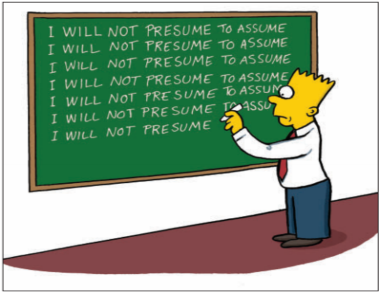

I WILL NOT PRESUME TO ASSUME
I WILL NOT PRESUME TO ASSUME
I WILL NOT PRESUME TO ASSUME
I WILL NOT PRESUME TO ASSUME
I WILL NOT PRESUME TO ASSUM
I WILL NOT PRESUME TO ASSU
I WILL NOT PRESUME

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?

Major / Minor Version

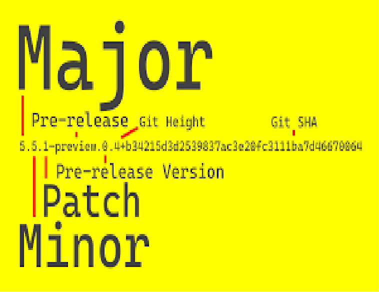


Figure 1: Problem Overview

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?

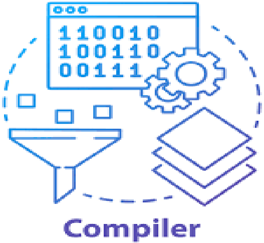


Figure 1: Problem Overview

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?

Windows vs Mac vs Linux

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?

Maybe...

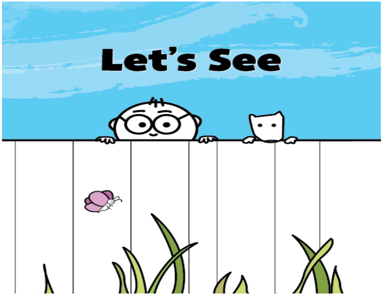


Figure 1: Problem Overview

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
  - Pending
  - Current
  - Active LTS
  - Maintenance LTS
- Java:
  - Oracle
  - OpenJDK
  - AdoptOpenJDK
- Cross Platform OS

Cross OS Programming Languages

Python2 Deprecated

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

Python3 Active

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

Node JS

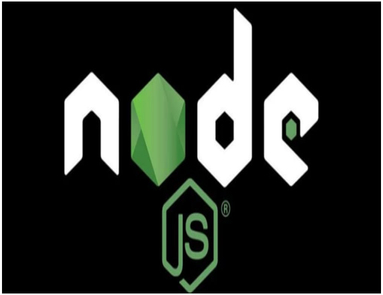
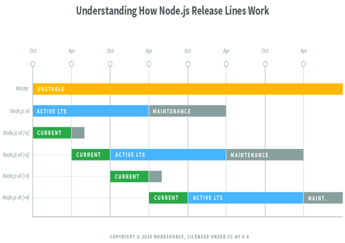


Figure 2: Problem Overview

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

Cross OS Programming Languages

Under Development



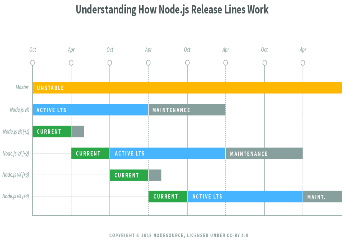
COPYRIGHT © 2018 NODESOURCE, LICENSED UNDER CC-BY 4.0

Figure 2: Problem Overview

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

Cross OS Programming Languages

Current Latest Release



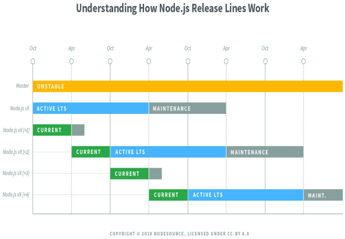
COPYRIGHT © 2018 NODESOURCE, LICENSED UNDER CC-BY 4.0

Figure 2: Problem Overview

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

Cross OS Programming Languages

Active Long Term Support


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:
  - Oracle
  - OpenJDK
  - AdoptOpenJDK
- Cross Platform OS

Cross OS Programming Languages

Maintenance Long Term Support

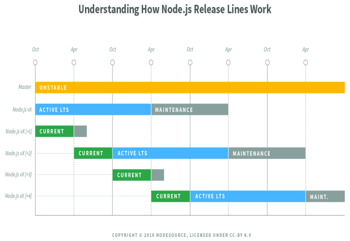


Figure 2: Problem Overview

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

Java Oracle

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

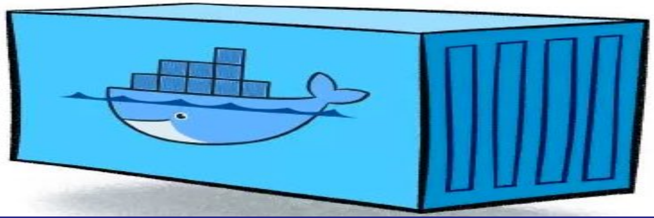
- Everyone needs to have exactly same OS.
- Everyone needs to install packages the same way.
- Everyone must update the OS and package at the same time.
- To what extent this is possible and practical?

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



Introduction

Software Development

Implementation

CI-CD Local

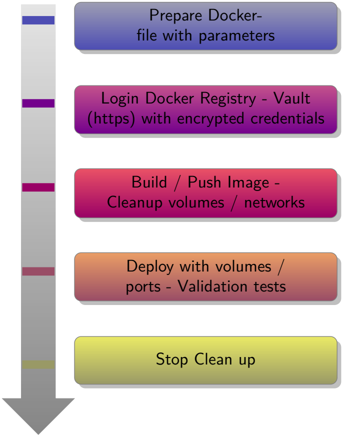
CI-CD Kubernetes

Demo

Summary

Bibliography

Local Procedure



Prepare Dockerfile with parameters

Login Docker Registry - Vault (https) with encrypted credentials

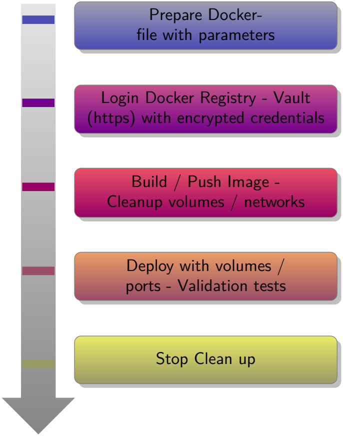
Build / Push Image - Cleanup volumes / networks

Deploy with volumes / ports - Validation tests

Stop Clean up

- 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



Prepare Dockerfile with parameters

Login Docker Registry - Vault (https) with encrypted credentials

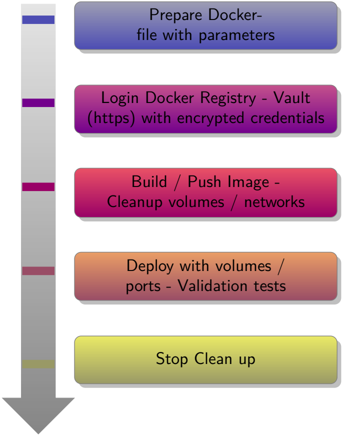
Build / Push Image - Cleanup volumes / networks

Deploy with volumes / ports - Validation tests

Stop Clean up

- 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


Prepare Dockerfile with parameters

Login Docker Registry - Vault (https) with encrypted credentials

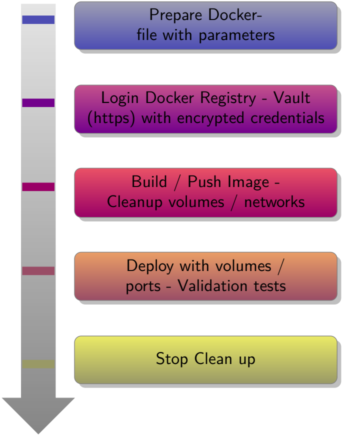
Build / Push Image - Cleanup volumes / networks

Deploy with volumes / ports - Validation tests

Stop Clean up

- 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



Prepare Dockerfile with parameters

Login Docker Registry - Vault (https) with encrypted credentials

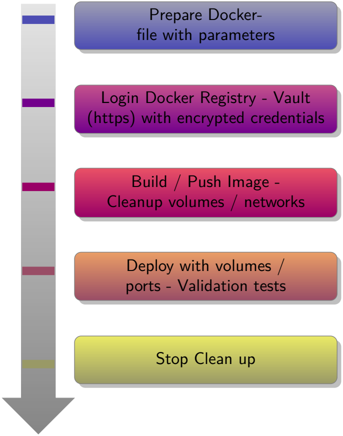
Build / Push Image - Cleanup volumes / networks

Deploy with volumes / ports - Validation tests

Stop Clean up

- 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



Prepare Dockerfile with parameters

Login Docker Registry - Vault (https) with encrypted credentials

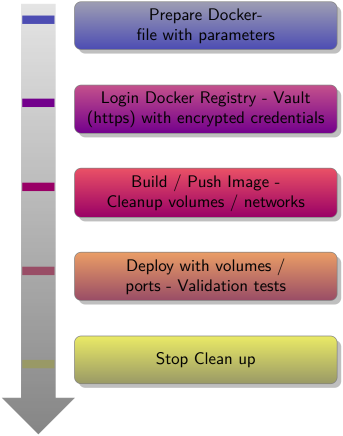
Build / Push Image - Cleanup volumes / networks

Deploy with volumes / ports - Validation tests

Stop Clean up

- 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



Prepare Dockerfile with parameters

Login Docker Registry - Vault (https) with encrypted credentials

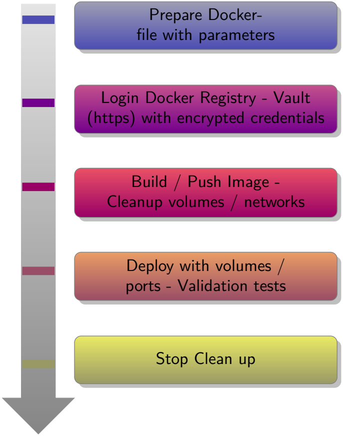
Build / Push Image - Cleanup volumes / networks

Deploy with volumes / ports - Validation tests

Stop Clean up

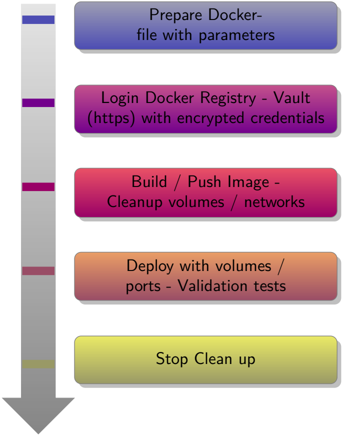
- Dockerfile (template).
- Vault (https).
- Any socket.
  - Azzure Registry.
  - Build Dockerfile.
  - Push Image.
  - Logout Azzure.
  - Prune everything.
  - Raise error (if).
- Deployment (volume).
- Validation (tests).
- Stop, Cleanup.

Local Procedure


Prepare Dockerfile with parameters
Login Docker Registry - Vault (https) with encrypted credentials
Build / Push Image - Cleanup volumes / networks
Deploy with volumes / ports - Validation tests
Stop Clean up

- 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


Prepare Dockerfile with parameters

Login Docker Registry - Vault (https) with encrypted credentials

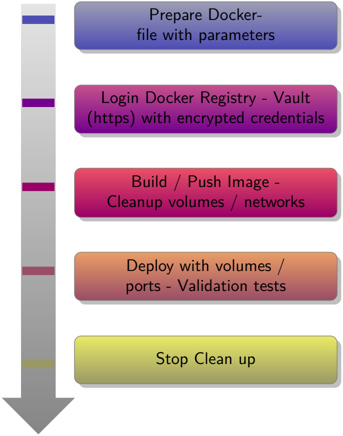
Build / Push Image - Cleanup volumes / networks

Deploy with volumes / ports - Validation tests

Stop Clean up

- 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.

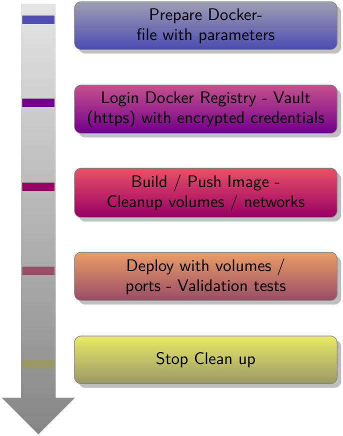
Local Procedure



Prepare Dockerfile with parameters
Login Docker Registry - Vault (https) with encrypted credentials
Build / Push Image - Cleanup volumes / networks
Deploy with volumes / ports - Validation tests
Stop Clean up

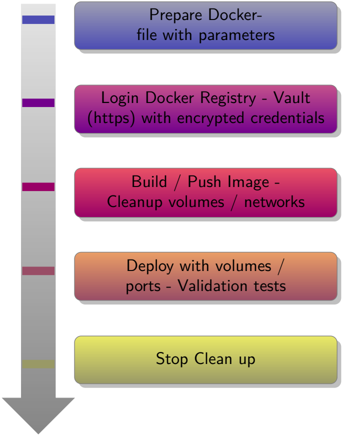
- Dockerfile (template).
- Vault (https).
- Any socket.
  - Azzure Registry.
  - Build Dockerfile.
  - Push Image.
  - Logout Azzure.
  - Prune everything.
  - Raise error (if).
- Deployment (volume).
- Validation (tests).
- Stop, Cleanup.

Local Procedure


Prepare Dockerfile with parameters
Login Docker Registry - Vault (https) with encrypted credentials
Build / Push Image - Cleanup volumes / networks
Deploy with volumes / ports - Validation tests
Stop Clean up

- 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



Prepare Dockerfile with parameters

Login Docker Registry - Vault (https) with encrypted credentials

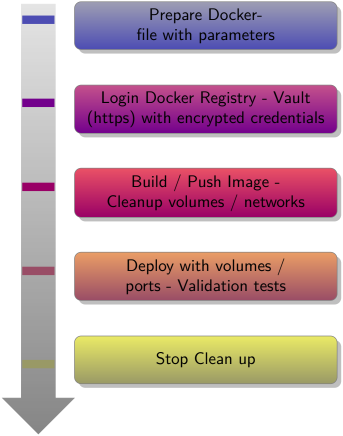
Build / Push Image - Cleanup volumes / networks

Deploy with volumes / ports - Validation tests

Stop Clean up

- 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


Prepare Dockerfile with parameters
Login Docker Registry - Vault (https) with encrypted credentials
Build / Push Image - Cleanup volumes / networks
Deploy with volumes / ports - Validation tests
Stop Clean up

- 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.

Introduction

Software Development

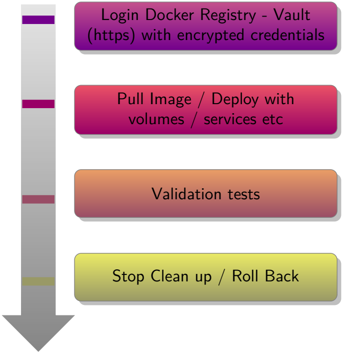
Implementation
CI-CD Local
CI-CD Kubernetes

Demo

Summary

Bibliography

Local k8s Deployment Procedure



Login Docker Registry - Vault (https) with encrypted credentials

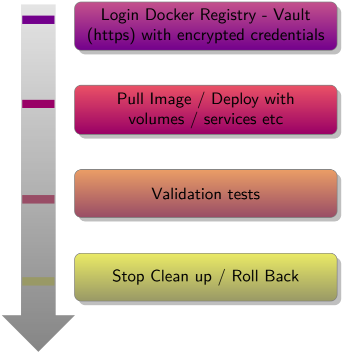
Pull Image / Deploy with volumes / services etc

Validation tests

Stop Clean up / Roll Back

- 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



Login Docker Registry - Vault (https) with encrypted credentials

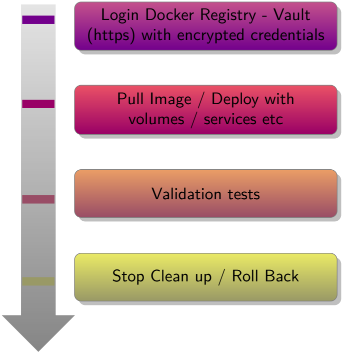
Pull Image / Deploy with volumes / services etc

Validation tests

Stop Clean up / Roll Back

- 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



Login Docker Registry - Vault (https) with encrypted credentials

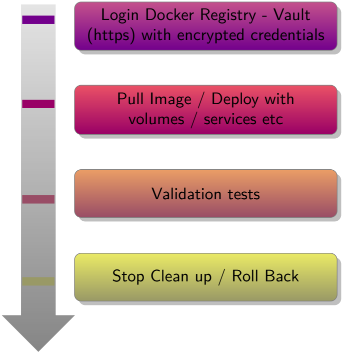
Pull Image / Deploy with volumes / services etc

Validation tests

Stop Clean up / Roll Back

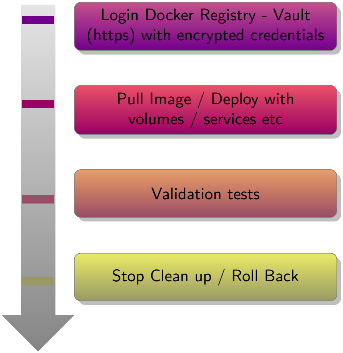
- Vault (https).
- Any socket.
  - Azzure Registry.
  - Pull Image.
- Deployment (volume).
- Logout Azzure.
- Validation (tests).
  - Raise error (if).
  - Stop, Cleanup.
  - Roll back.

Local k8s Deployment Procedure


Login Docker Registry - Vault (https) with encrypted credentials
Pull Image / Deploy with volumes / services etc
Validation tests
Stop Clean up / Roll Back

- 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



Login Docker Registry - Vault (https) with encrypted credentials

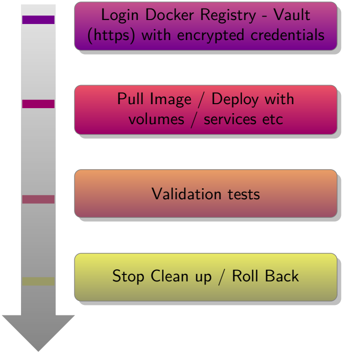
Pull Image / Deploy with volumes / services etc

Validation tests

Stop Clean up / Roll Back

- 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



Login Docker Registry - Vault (https) with encrypted credentials

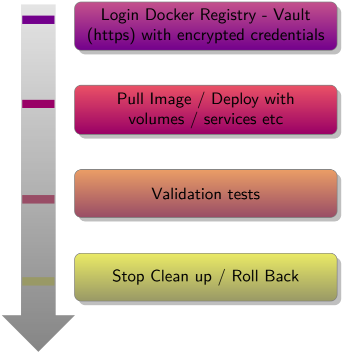
Pull Image / Deploy with volumes / services etc

Validation tests

Stop Clean up / Roll Back

- 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



Login Docker Registry - Vault (https) with encrypted credentials

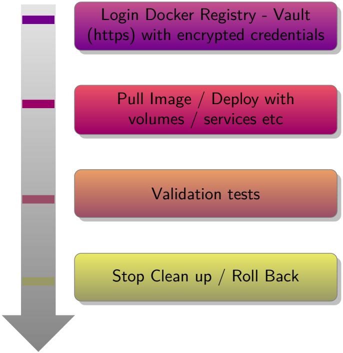
Pull Image / Deploy with volumes / services etc

Validation tests

Stop Clean up / Roll Back

- 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



Login Docker Registry - Vault (https) with encrypted credentials

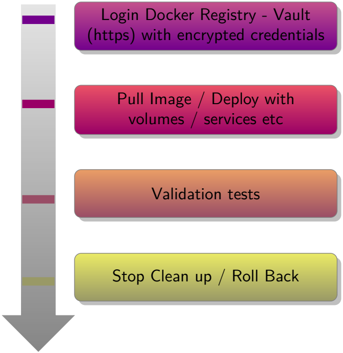
Pull Image / Deploy with volumes / services etc

Validation tests

Stop Clean up / Roll Back

- 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



Login Docker Registry - Vault (https) with encrypted credentials

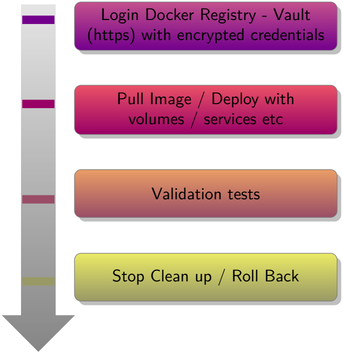
Pull Image / Deploy with volumes / services etc

Validation tests

Stop Clean up / Roll Back

- 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


Login Docker Registry - Vault (https) with encrypted credentials
Pull Image / Deploy with volumes / services etc
Validation tests
Stop Clean up / Roll Back

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

Introduction

Software Development

Implementation

Demo

Demo CI / CD

Summary

Bibliography

Coming up: Demo

Introduction

Software Development

Implementation

Demo

Summary

Key points repetition

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 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 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 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 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 LaTeX

Coming up: Q & A

References I

GNU LESSER GENERAL PUBLIC LICENSE
GNU Operating System
available at https://www.gnu.org/licenses/lgpl.html.