Developing Docker and Docker-Compose Specifications: A Developers' Survey

Reis, David et al.

MINNA FENG, SANDY NGUYEN, BEATRICE DUVAL

Agenda

- Introduction
- Related Work
- Goals of the Study
- Methodology
- Sampling and Recruitment
- Analysis of Results
- Threats to Validity and Limitations
- Conclusions
- Future Research

Introduction

- Study current usage.
- Understand how people create and deploy Docker-based systems.
- Find better ways to address challenges.
- Show how ancillary tools make development process better.

Related Work

Available tools lack support for quality-focused maintenance and evolution.

Challenging categories on Stack Overflow: application development, configuration, networking, basic concepts and debugging.

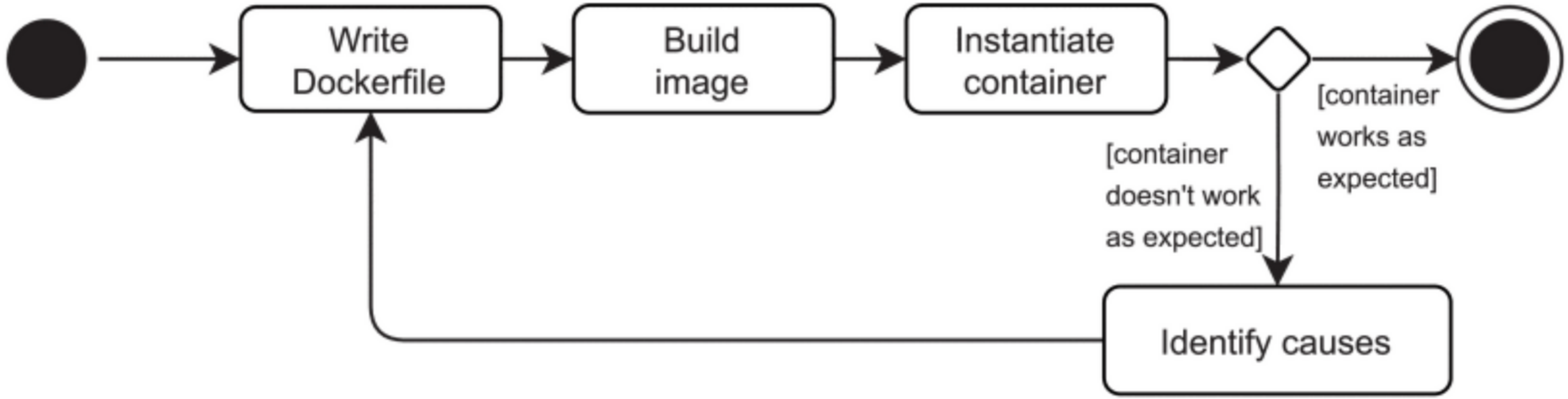
Over ⅓ of Dockerfiles couldn't be built error-free, and many had quality issues.

½ of the IaC specifications in their sample contained syntax and configuration-related defects at one of their revisions.

Over ¼ of the projects needlessly used Docker-Compose, multi-component specifications mostly used basic features.

Opportunities for improving the efficiency of developing Docker specifications and for tools to reduce the time spent in development.

Example of a workflow when developing a Dockerfile



Goals of the Study

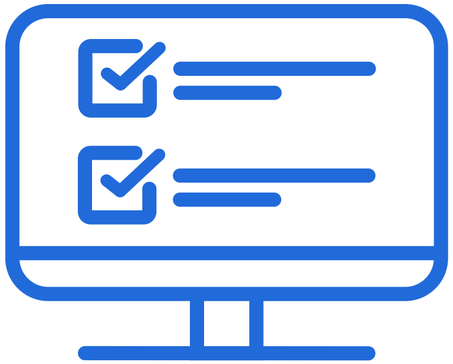
RQ1 Which activities are regarded as time-consuming?

RQ2 Which approaches are used to diagnose and correct problems?

RQ3 What role is played by ancillary software tools?

OUR STRATEGY

Methodology



Online Surveys

Target forums

PEOPLE AND QUESTIONNAIRE

Sampling

Sampling stategy:

- Convenience
- Referral-chain sampling

To Prevent Oversampling:

- Communities gathering
- Social Networking
- Participants of the XP Conderence

To persuade participation:

- Offered first-hand access to results
- Showed a short promotional video

PEOPLE AND QUESTIONNAIRE

Questionnaire

- Online Form
- Response time: below 5 mins
- Mix of open and closed-ended questions
- Either focus on Dockerfiles or docker.compose.yml

RESULTS

Preliminary run with students

- Most perceive they spend much time understanding why a Docker container is not running as intended.
- Few use ancillary tools in Docker-compose.

RESULTS

Demographics

120
Participants

24
Different countries

90%
Work in the industry

12%
Work in academia

87%
Role in development

54%
Role in operations

RESULTS

Experience Level with Docker Technologies

EXPERIENCED GROUP

54

more than 3 years

INEXPERIENCED GROUP

65

3 years or less

RESULTS

RQ1 – Which activities are time consuming?

- 1) Dockerfile and 2) docker-compose
- Development activities breakdown
- Attitude towards a considerable amount of time spent on these activities
- Impact of experience level

RQ1 - WHICH ACTIVITIES ARE TIME CONSUMING?

Dockerfile Development Process

A1 Reading Docker documentation.

A2 Finding out what are the right Dockerfile Commands that I need.

A3 Finding out what parent image is the most suitable.

A4 Finding Out What Are The Dependencies My System that must be added to the docker image.

A5 Confirming if the resulting container is working as intended.

A6 Trying to understand why the resulting container is not working as intended.

A7 Finding out which commands are responsible for the container misbehavior.

A8 Rebuilding the image and re-running the container to confirm that it is working as intended

RQ1 - WHICH ACTIVITIES ARE TIME CONSUMING?

Most Time Consuming Activities

A1 Reading Docker documentation.

A2 Finding out what are the right Dockerfile Commands that I need.

A3 Finding out what parent image is the most suitable.

→ **A4 Finding Out What Are The Dependencies My System that must be added to the docker image.**

→ **A5 Confirming if the resulting container is working as intended.**

→ **A6 Trying to understand why the resulting container is not working as intended.**

A7 Finding out which commands are responsible for the container misbehavior.

→ **A8 Rebuilding the image and re-running the container to confirm that it is working as intended**

RQ1 - WHICH ACTIVITIES ARE TIME CONSUMING?

Experience Level

A1 Reading Docker documentation.

A2 Finding out what are the right Dockerfile Commands that I need.

A3 Finding out what parent image is the most suitable.

→ **A4 Finding Out What Are The Dependencies My System that must be added to the docker image.**

→ **A5 Confirming if the resulting container is working as intended.**

A6 Trying to understand why the resulting container is not working as intended.

A7 Finding out which commands are responsible for the container misbehavior.

→ **A8 Rebuilding the image and re-running the container to confirm that it is working as intended**

RQ1 - WHICH ACTIVITIES ARE TIME CONSUMING?

Areas of Focus for Improvements

A1 Reading Docker documentation.

A2 Finding out what are the right Dockerfile Commands that I need.

A3 Finding out what parent image is the most suitable.

→ **A4 Finding Out What Are The Dependencies My System that must be added to the docker image.**

→ **A5 Confirming if the resulting container is working as intended.**

→ **A6 Trying to understand why the resulting container is not working as intended.**

A7 Finding out which commands are responsible for the container misbehavior.

→ **A8 Rebuilding the image and re-running the container to confirm that it is working as intended**

RQ1 - WHICH ACTIVITIES ARE TIME CONSUMING?

Docker-compose development process

W1 Reading Docker documentation;
W2 Finding out what are the keys that I need;
W3 Finding out what images are available;
W4 Trying to understand why the services are not working as intended;
W5 (Re)starting the services to confirm that they are working as intended;
W6 Configuring the properties of each service (e.g., port mapping, name, . . .);
...
R1 Trying to understand what the services are;
R2 Trying to understand the dependencies between services (e.g., depends_on);
R3 Trying to understand what volumes are used and how they are attached to the services;
R4 Trying to understand what networks are used and how they are connected to the services.

Docker-Compose

01 **Activities become easier with experience**

02 **Inexperienced developers would benefit from improvements in activities related to:**

- Configuring properties related to services
- Debugging
- Documentation

RESULTS

RQ2 - Which approaches are used to diagnose and correct problems?

Dockerfiles

- Trial-and-error
- Searching Web resources
- Entering the container to execute commands which may help to diagnose the issue manually
- Continuous integration pipelines
- End-to-end testing

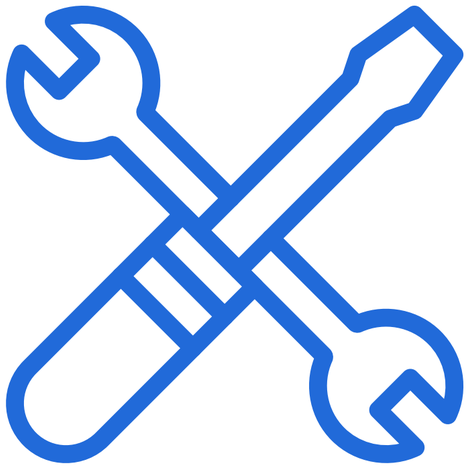
Docker-compose

- Analysis of the output logs
- Execution of commands within the running containers
- Isolating services individually, making sure they work as expected
- Test their dependencies

RESULTS

RQ3 – What role is played by ancillary software tools?

- Not widely used
- Need for tools that improve the process of writing specifications



Threats to Validity and Limitations

EFFORTS TO MITIGATE AND IDENTIFICATION OF THOSE WE CAN'T DISCARD

1

Questionnaire Design

2

Clarity of the questionnaire

3

Honesty of the respondents

4

Assessing developer experience

5

Representativeness of the sample

6

Generalization to to other IaC platforms

Conclusion

The Study

Try to reach insights on the development of **container and orchestration specficiations.**

The Focus

Focus on two IaC technologies: **Docker and Docker-Compose.**

The Results

- Dockerfile development does present challenging activities and developers tend to perceive most of them as **time-consuming**.
- Docker-compose results were **not relevant.**
- For both, the activities that would most benefit from improvement are debugging and finding dependencies.
- Developers do not use ancillary tools.

Future Research

Usability Evaluation:
Task analysis or cognitive walkthrough

New approaches or environments to develop Docker and Docker-Compose specifications