PharmaSUG 2021 - Paper EP-178

Portable, Dynamic, and Powerful: Virtualization Requirements in Regulatory Review

Eli Miller, Atorus Research

ABSTRACT

Virtualization is being explored by many industry groups as it presents a solution to several challenges that some organizations have struggled with. Container runtimes allow for system architects to design reproducible, immutable environments that can be portable, easily qualified, and hosted on a variety of different hosts. Despite these benefits, many organizations do not take advantage of the developing use of containers. This paper will discuss certain aspects of containerization that solve unique challenges the industry faces and explore aspects of containerization that may present challenges for their use in regulatory submissions.

INTRODUCTION

As the use of virtualization becomes more mature in the pharmaceutical industry, there is a need of scripts for qualifying containers, as well as requirements for how containers should be constructed. Many industry groups are exploring the use of containers; however, this paper will be focused on issues that affect the qualification of the environment and issues that might arise in regulatory review. Starting with why, this paper will briefly discuss several benefits containerization can offer analysis systems for clinical research. Three flows for qualification will be discussed along with their pros and cons. Finally, how data is mounted, copied, and qualified with different methods of storage will be explored.

Each of these are critical components that must be resolved before dynamic review documents can make an impact on regulatory review. Virtualization and containerization are technically complex, and these efforts may not work if the setup or use of the containers requires skills that are not already held by reviewers and scientists. Certain pieces of the qualification process will be technical in nature; however, the end goal would be a solution that is usable by a reviewer or scientist with little knowledge of the containerization process.

THE PROBLEMS CONTAINERS SOLVE

PORTABILITY

Containerization allows system designers to focus on a single runtime rather than designing environments that are interoperable between different OS types as the runtime will guarantee a consistent interface for the system to sit on. Any image can be used by an organization for analysis, then reliably used by a different reviewer, on a different machine, with minimal setup or changes to the existing systems. While the movement of images across analysis systems is a great feature, another benefit is the flexibility to switch between images within a single system. Certain studies may have different analysis requirements that can conflict with one another, by virtualizing these environments scientists can switch between them seamlessly.

REPRODUCABILITY

Containers remove a great deal of the setup non-virtualized systems require. Containers can be started in seconds where regular environments can take hours to configure. Along with the ability to go from a stored image to running container, images can be stored with tags that allow for versioning of environments to set checkpoints for study milestones or points in time. As containers can be packaged up with minimal effort, the cost of storing them becomes negligible. Instead of storing physical infrastructure with outdated systems, reproducibility requirements can be met by freezing the analysis system in an image registry.

ENVIRONMENT REQUIREMENTS

Analysis environments can be a challenge to create due to the complexity of dependencies that most systems require. Running multiple environments on the same host can be a challenge if there are conflicts between environment dependencies. Containers solve this challenge by being flexible enough to allow for multiple environments to be constructed on top of one another. Images are built in layers meaning base images can be extended to make constructing and maintaining them easier. A good example are images created by the R community. Generally, they are started with a base OS image, then extended to include a version of R built from source or binaries, then extended again to include any packages, dependencies, or data needed in the package.

SYSTEM QUALIFICATION APPROACHES

There are several different stages of the image lifecycle where qualification could be performed, and while the tests are the same in these situations, the results can mean slightly different things. These flows are not mutually exclusive, in fact all three of these can be utilized for portions of the qualification process if an organization finds this suits their needs best.

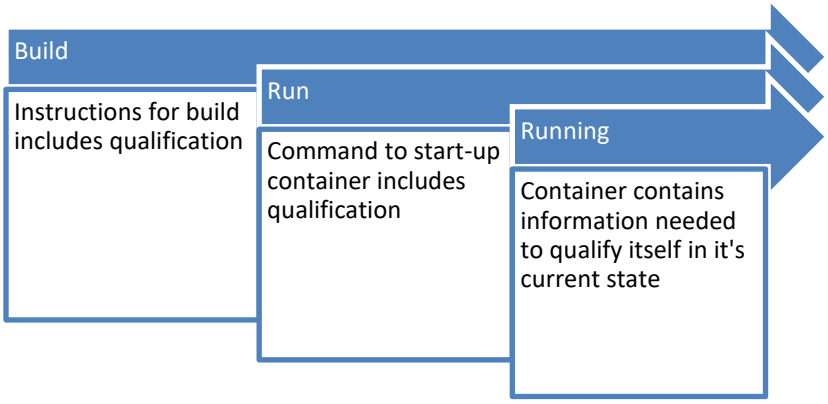

Figure 1. Sample Qualification Stages

ON-RUN QUALIFICATION

In this framework when a container is started, the first process that is executed is a qualification script that ensures the environment is working as expected. In this case, the qualification could include resources like external storage, which is discussed below. This method would prevent a container that is not passing all qualification tests from being used, however the qualification tests would be run every time the analysis environment is started which may not be necessary and would be time consuming. Tests that rely on mounted storage could be ran in this context to verify expected data is present and ready for analysis.

ON-DEMAND QUALIFICATION

If the running container changes during qualification, for example if the qualification depends on mounted storage, it may be necessary to qualify the container after it is stared. This method ensures the container can be ready for qualification at any time, and the qualification is done at the most convenient time for the user. This offers greater flexibility over on-run qualification and will allow a programmer to write their own analysis, add their own tests, and qualify their contributions in the analysis environment directly.

ON-BUILD QUALIFICATION

In this process the qualification is done as one of the final steps of the construction of the image. This way the qualification artifacts are available in each container when they are started. This results in the qualification documents being built into image itself so they cannot be lost and are always the environment for anyone with access to review them. In this case, mounted storage would not be available during the qualification process, which may require certain portions to be requalified.

DATA VOLUMNES AND CONTAINER PORTABILITY

Containers running R are generally large due to the dependencies that R requires. When packages and their dependencies are included the images, they can become larger still. Table 1 below gives a brief view of different R containers and how additional dependencies result in larger container sizes.

Repository/Image:Tag	Description	Size (compressed)
rocker/r-ver:4.0.0	Minimal Base R Image	310MB
rocker/rstudio:4.0.0	RStudio Environment	558MB
rocker/shiny:4.0.0	Shiny Server Application	508MB
rocker/shiny-verse:4.0.0	Shiny Server Application with tidyverse library	661MB
rstudio/r-session-complete:bionic-1.4.1103-4	Container to run jobs from RStudio Server	3.04GB
alpine:3.12.4	Minimal Linux Server	2.67MB

Table 1. Community R Container Sizes

DATA IN-IMAGE

Files can be copied directly into an image when it is built. This would be the most straight-forward way to include data as it is now a part of the image itself. While this is a simple solution, this results in the image becoming much larger and harder to pull onto new hosts. As seen in Table 1, an analysis environment that depends on several different languages and libraries can reach the size of gigabytes and adding study data on top of that can make the images become hard to manage and pull from registries.

DATA VOLUMES

File systems present on the host can be mounted into a running container. This would be the preferred way if the container is being run on a host that already has the data present on it. This method could be used in situations like the eCTD where the data is in a known location and has the benefit of not duplicating the data and keeping the container as lightweight as possible. For this configuration, a directory on the host can be specified and mounted in a container which can allow for analysis environments to be reused for different study data.

EXTERNAL FILE SYSTEMS

External filesystems are sources of data where the host is not storing the actual files, instead the container is mounting storage over a network. This has the benefit of a lightweight image, neither the container nor the host needs to contain any data, instead it is retrieved as the environment needs it. Containers could leverage existing file systems or databases which most organizations already possess for hosting their data. This method adds another application that the container depends on which could fail and cause delays in analysis.

CONCLUSION

Challenges remain in the virtualization and containerization of study analysis. The nuances of different approaches of qualification can cause confusion and present a barrier for adoption in the industry. The benefits of virtualization can improve the clinical analysis workflows of most organizations and builds in reproducibility and portability to core of the analysis. To effectively realize these benefits developers, system architects, and statisticians must know the high-level details of how their environments are constructed and run, as well as the alternatives to their current environment.

Qualification is a critical component of any regulated study analysis must be available and its concepts well-documented for it to be accepted by organizations and regulatory reviewers. System qualification is not a trivial task; however, virtualization presents an incredible gain in efficiently for system qualification. The ability to perform a qualification at any point in the container lifecycle with minimal intervention from a user removes point of friction that can be painful depending on the workflow, while still giving organizations a high level of confidence in their systems.

Portability is a feature of images that can be lost easily if an organization is not careful about how they construct their environments. As images become larger the extraction and transmission can take considerably longer and can be inefficient if data present on the image is already present on the intended host machines. By mounting external filesystems, a user could easily pull down the analysis environment and load in the needed data without duplicating any information. Existing databases and eCTD structures present a useful drop in for mounting storage in a virtualized environment.

CONTACT INFORMATION

Your comments and questions are valued and encouraged. Contact the author at:

Eli Miller
Eli.Miller@AtorusResearch.com