|  |  |  |
| --- | --- | --- |
| Image | |  |
| **Advance Development Factory**  Requirement & Design Specification | | |
|  | DevSecOps  V1.0  UMGC Capstone | |

|  |
| --- |
|  |
| Table of Contents  [Introduction 4](#_Toc51324756)  [Document Conventions 4](#_Toc51324757)  [Intended Audience 4](#_Toc51324758)  [Definitions, Acronyms, & Abbreviations 5](#_Toc51324759)  [Scope 5](#_Toc51324760)  [Overall Description 5](#_Toc51324761)  [Product Perspective & Concept 5](#_Toc51324762)  [Product Features 5](#_Toc51324763)  [User Classes & Characteristics 6](#_Toc51324764)  [Operating Environment 6](#_Toc51324765)  [Design and Implementation Constraints 6](#_Toc51324766)  [Assumptions and Dependencies 7](#_Toc51324767)  [System Features 7](#_Toc51324768)  [1) Feature: Isolated Operating Environment 7](#_Toc51324769)  [2) Feature: Application Build 7](#_Toc51324770)  [3) Feature: Application Test 8](#_Toc51324771)  [4) Feature: Application Deployment 8](#_Toc51324772)  [External Interface Requirements 8](#_Toc51324773)  [User Interface Overview 8](#_Toc51324774)  [Hardware Interfaces 8](#_Toc51324775)  [Software Interface 8](#_Toc51324776)  [System Features/Modules 9](#_Toc51324777)  [1) Isolated Operating Environment 9](#_Toc51324778)  [2) Application Build 10](#_Toc51324779)  [3) Application Test 11](#_Toc51324780)  [4) Application Deployment 12](#_Toc51324781) |
|  |

|  |
| --- |
| Introduction As part of a diverse ecosystem of applications and technologies within software development, a need has been identified that standardizes the building, testing, and deployment of an organization's application suite. This document outlines the Advance Development Factory design and requirements specification that sets the standard for developers to build, test, and deployment of a diverse set of applications under a common framework that mitigates clients operating environment dependencies. Document Conventions This document follows a modified IEEE 830 Recommended Practice for Software Requirements Specifications. This document contains five main sections with support sections for details. Section one of the introduction provides a summary of what this document is and who it is intended. Section two presents the overall description of the product and its features. Section three provides the high-level system features for release 1.0. Section four provides the external interface requirements from the users' perspective. Section five provides the system features/modules which expand upon the system feature from section three. Intended Audience This document is an internal document intended for all current and future project teams and is not meant to facilitate customer negotiations. This document is intended to be read in its entirety then used as a reference manual. Definitions, Acronyms, & Abbreviations This document will contain acronyms and vocabulary specific to the ADF application. This section defines ADF's vocabulary.  ADF – Advance Development Factory  UMGC – University of Maryland Global Campus  OE – Operating environment  OS – Operating System  CLI – Command Line Interface Scope The scope of the ADF is limited to the following development activities of UMCG Capstone applications:   * Building * Testing * Deploying |

# Overall Description

## Product Perspective & Concept

The Advance Development Factory intends to provide all UMGC projects with a standard mechanism for building, testing and deploying applications whilst limiting users operating environment dependencies. ADF is able to accomplish a low operating environment dependency by building the environment in which users can build, test, and deploy applications. ADF is a bootstrap application context that leverages Docker to isolate user's environments.

## Product Features

The following are the core features of ADF release 1.0. Some of the core features of ADF are characterized by indirect features in that they provide users with a facultative environment context that is not best defined by a stimulus and response format; thus, section three will outline ADF's use cases in a hybrid stimulus, response, format or facultative format.

* Isolated OE
* Application build
* Application test
* Application deployment

## User Classes & Characteristics

Developer:

Developers are the only user classes as ADF enables a development environment framework capable of software development lifecycle activities.

## Operating Environment

The following are the initial developer OE dependencies the enable the ADF bootstrapping process.

OE-1: GNU bash, version 4.2 or greater

OE-2: Docker, version 19.03 or greater

OE-3: GNU Make, version 3.82 or greater

## Design and Implementation Constraints

The following constraints ensure the integrity of ADF's core capabilities remain intact as ADF's features expand.

CO-1: The ADF shall be realized in the form of a Docker image.

CO-2: The ADF shall adopt Docker's best practices. [[1]](#footnote-1)

CO-3: The ADF shall support building Java SE 8-11 applications.

CO-4: The ADF shall support apache-maven as necessary to build Java SE 8-11 applications.

CO-5: The ADF shall provide an Azure CLI.

CO-6: The ADF shall provide a Docker CLI.

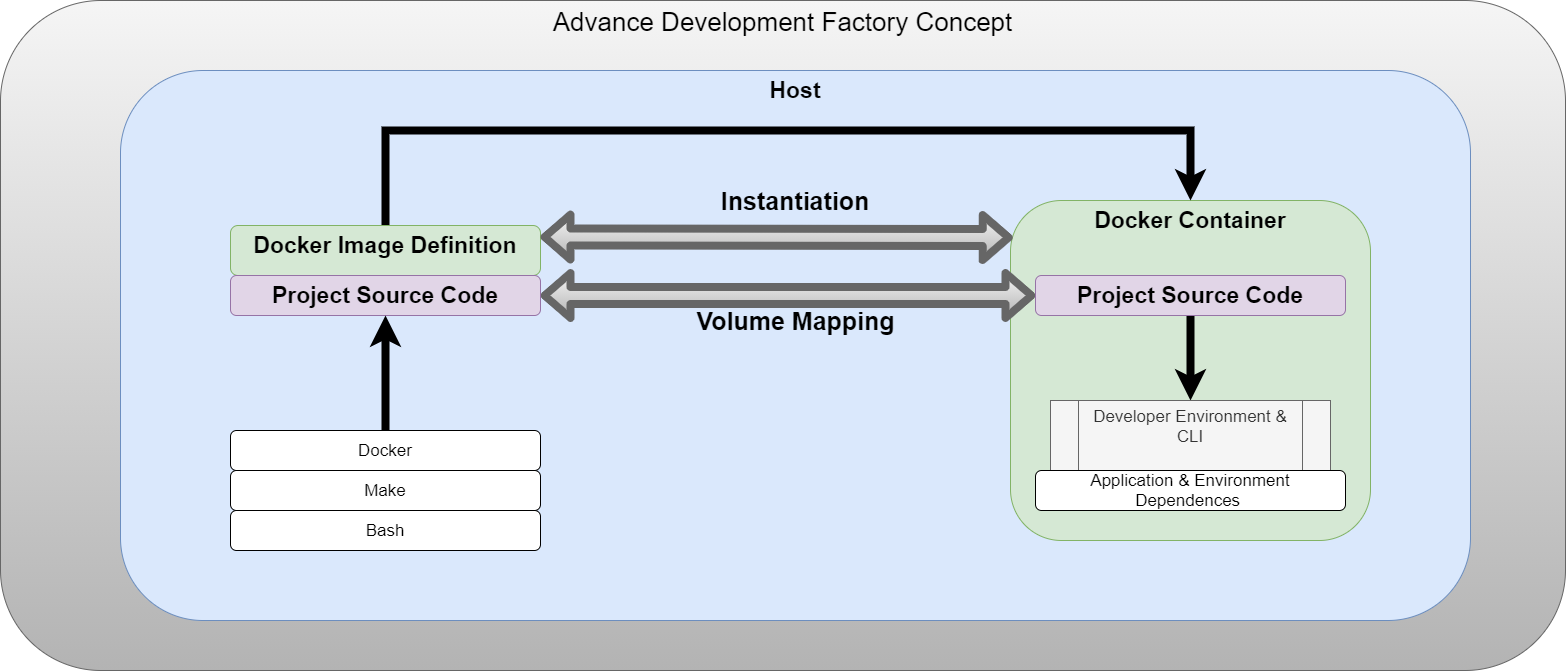
## Assumptions and Dependencies

The following assumptions and dependencies are assumed to be true.

AS-1: Developers have a connection to the internet.

AS-2: Developers Docker settings enable container connectivity to the internet.

# System Features



### Feature: Isolated Operating Environment

This feature is realized by building and starting the ADF application. The ADF design and implementation constraints, coupled with the Docker technology, enable host OE isolation.

### Feature: Application Build

This feature is realized by the ADF application constraints that provide the required packages within the Docker image to build Java SE 8-11 applications.

### Feature: Application Test

This feature is realized by the ADF application constraints that provide the required packages within the Docker image to test Java SE 8-11 applications.

### Feature: Application Deployment

This feature is realized by the ADF application constraints that provide the required packages within the Docker image to deploy to an Azure instance via the Azure command-line interface.

# External Interface Requirements

## User Interface Overview

ADF is a command-line interface application, and no graphical user interfaces have been identified.

## Hardware Interfaces

No Hardware interfaces have been identified.

## Software Interface

The following software interface enabled ADF to operate as expected.

SI-1: Host Docker Configuration: The developer's host docker configuration enables host volume mapping

# System Features/Modules

### Isolated Operating Environment

Description:

A developer who has the initial ADF dependencies available defined by OE-1, OE-2, and OE-3, can start the ADF application.

Stimulus/Response Sequences:

Stimulus: A developer runs *make start-env* via BASH CLI.

Response: The system shall instantiate the ADF Docker image and provide the developer with an isolated environment.

Functional Requirements:

REQ-1.1: Upon running *make start-env* via BASH CLI, the system shall instantiate the ADF Docker image and provide the developer with an isolated environment.

REQ-1.2: When the ADF Docker image is instantiated, the system shall provide a volume map of the UMGC Capstone application source code into the ADF Docker container.

### Application Build

Description:

A developer who has the initial ADF dependencies available defined by OE-1, OE-2, and OE-3 and has started the ADF application can build UMGC Capstone applications.

Stimulus/Response Sequences:

Stimulus: A developer runs *make build* within the ADF context.

Response: The system shall build the UMGC Capstone application and produce an application executable.

Functional Requirements:

REQ-2.1: Upon running *make build* within the ADF context, the system shall build the UMGC Capstone application and produce an application executable.

REQ-2.2: During the building of a UMGC Capstone application, the system shall run developer-defined tests and only produce an application executable only if all testing is deemed successful.

REQ-2.3: When the UMGC Capstone application is successfully built, the system shall produce a Dockerized application from the previous application executable.

### Application Test

Description:

A developer who has the initial ADF dependencies available defined by OE-1, OE-2, and OE-3 and has started the ADF application can test UMGC Capstone applications.

Stimulus/Response Sequences:

Stimulus: A developer runs *make test* within the ADF context.

Response: The system shall run developer-defined tests.

Functional Requirements:

REQ-3.1: Upon running *make test* within the ADF context, the system shall run developer-defined tests.

### Application Deployment

Description:

A developer who has the initial ADF dependencies available defined by OE-1, OE-2, and OE-3 and has started the ADF application and built the UMGC application can deploy to an Azure resource.

Stimulus/Response Sequences:

Stimulus: A developer runs *make deploy* within the ADF context.

Response: The system shall deploy the application to an Azure resource.

Functional Requirements:

REQ-4.1: Upon running *make deploy* within the ADF context; the system shall deploy the application to an Azure resource.

REQ-4.2: They system shall make use of Azure templates to deploy UMGC Capstone applications.

1. https://docs.docker.com/develop/develop-images/dockerfile\_best-practices/ [↑](#footnote-ref-1)