

Crucible

[Go to General Topics](#)

The Objective of Crucible

Crucible provides a controlled, reproducible, and portable approach for constructing, hardening, assessing, transferring, deploying, and maintaining software and infrastructure environments. Crucible uses [Declarative Descriptions](#), composable baselines, [Machine Images](#), [Infrastructure as Code \(IaC\)](#), dependency capture, and [Compliance Evidence](#) to support [Accredited Software Factory](#) objectives across [Connected](#), [Disconnected](#), and [Air-Gapped Environments](#). The architecture separates provider-independent intent from provider-specific implementation, preserves [Traceability](#) and provenance across lifecycle stages, and supports [Reproducibility](#) and [Portability](#) without depending on a single cloud provider, infrastructure platform, or assessment tool.

Table of Contents

- [Table of Contents](#)
- [1. Introduction](#)
- [2. Operational Need and Mission](#)
- [3. Operational Environment](#)
- [4. Actors and Responsibilities](#)
- [5. Operational Concept](#)
- [6. Connected Operations](#)
- [7. Disconnected and Air-Gapped Operations](#)
- [8. Compliance Operations](#)
- [9. Crucible Interfaces](#)
- [Annexes](#)

—

© 2026 Dido Solutions, Inc. and Jackrabbit Consulting, Inc.

Table of Contents

- [1 Introduction](#)
 - [1.1 Purpose](#)
 - [1.2 Scope](#)
 - [1.3 Intended Audience](#)
 - [1.4 How to Use This Wiki](#)
 - [1.5 Document Conventions](#)
- 2. [2 Business and Executive Concepts](#)
 - [2.1 What Is Crucible?](#)

- 2.2 Accredited Software Factory as a Commodity
- 2.3 Business Value
- 2.4 Stakeholders
- 2.5 Current Status
- 3. 3 Environments and Connectivity
 - 3.1 Operational Environments
 - 3.2 Connected Operations
 - 3.3 Disconnected Operations
 - 3.4 Air-Gap Transfer
- 4. 4 Platforms and Providers
 - 4.1 Platform Concepts
 - 4.2 Cloud Agnosticism
 - 4.3 Provider Architecture
 - 4.4 Supported Providers
 - 4.4.1 Oracle Cloud Infrastructure
 - 4.4.2 Amazon Web Services
 - 4.4.3 OpenStack
 - 4.4.4 VMware vSphere
 - 4.4.5 libvirt
- 5. 5 Descriptions, Composition, and Baselines
 - 5.1 Crucible Description
 - 5.2 Baseline Concepts
 - 5.3 Baseline Composition
 - 5.4 Pipeline Workspace
 - 5.5 Version Pinning and Pull Lock
- 6. 6 Machine Images and Image Layers
 - 6.1 Machine Image Concepts
 - 6.2 Image Layering
 - 6.3 Image Build
 - 6.4 Supported Operating Systems
 - 6.4.1 Red Hat Enterprise Linux
 - 6.4.2 Ubuntu
 - 6.4.3 Windows Server
 - 5. 6.5 Supported Processor Architectures
 - 6. 6.6 Image Catalog and Records
 - 7. 6.7 Image Requirements and Resolution
 - 8. 6.8 Image Lifecycle
- 7. 7 Infrastructure and Deployment
 - 7.1 Infrastructure as Code
 - 7.2 Platform Initialization
 - 7.3 Infrastructure Deployment
 - 7.4 Provisioning
 - 7.5 Orchestration
 - 7.6 Teardown
- 8. 8 Dependencies and Air-Gap Operations
 - 8.1 Dependency Concepts
 - 8.2 Dependency Capture
 - 8.3 Dependency Store
 - 8.4 Transfer Bundle
 - 8.5 Offline Repositories and Mirrors
 - 8.5.1 Offline Repository

- 8.5.2 Package Repository
- 8.5.3 Container Mirror
- 8.5.4 Provider Mirror
- 6. 8.6 Offline Reconstitution
- 7. 8.7 Supply-Chain Capture and Integrity
- 9. 9 Compliance and Security
 - 9.1 Compliance by Design
 - 9.2 Compliance Baseline and Posture
 - 9.3 Compliance as Code
 - 9.4 Security Technical Implementation Guides
 - 9.5 Compliance Provider Architecture
 - 9.6 Scan, Normalize, and Remediate
 - 9.6.1 Compliance Scan
 - 9.6.2 Compliance Finding
 - 9.6.3 Normalized Finding
 - 9.6.4 Findings Model
 - 9.6.5 Remediation
 - 9.6.6 Post-Remediation Verification
- 7. 9.7 Compliance Evidence
- 8. 9.8 Evidence Outputs
 - 9.8.1 Security Controls Traceability Matrix
 - 9.8.2 Risk Management Framework Evidence
 - 9.8.3 Authority to Operate Evidence
- 9. 9.9 Security of Crucible
- 10. 10 Reproducibility, Provenance, and Traceability
 - 10.1 Reproducibility
 - 10.2 Provenance
 - 10.3 Traceability
 - 10.4 Controlled Inputs
 - 10.5 Auditability
- 11. 11 Architecture and Extensibility
 - 11.1 Architecture Overview
 - 11.2 Design Principles
 - 11.3 Subsystems
 - 11.3.1 Forge CLI
 - 11.3.2 Baseline Workspace
 - 11.3.3 Image Builder
 - 11.3.4 Infrastructure Engine
 - 11.3.5 Infrastructure Registry
 - 11.3.6 Infrastructure Core
 - 11.3.7 Platform Deployer
 - 11.3.8 Dependency Manager
 - 11.3.9 Compliance Engine
 - 11.3.10 Requirements Verifier
 - 4. 11.4 External Toolchain
 - 5. 11.5 Extension Model
 - 6. 11.6 Architecture Decision Records
- 12. 12 Operational Lifecycle
 - 12.1 Lifecycle Overview
 - 12.2 Lifecycle Stages
 - 12.2.1 Declare

- [12.2.2 Pull](#)
 - [12.2.3 Build](#)
 - [12.2.4 Harden](#)
 - [12.2.5 Scan](#)
 - [12.2.6 Upload](#)
 - [12.2.7 Initialize](#)
 - [12.2.8 Deploy](#)
 - [12.2.9 Verify](#)
 - [12.2.10 Audit](#)
 - [12.2.11 Maintain](#)
 - [12.2.12 Promote](#)
 - [12.2.13 Teardown](#)
 - [12.2.14 Reproduce](#)
3. [12.3 Build Pipeline](#)
13. [13 Use Cases](#)
- [13.1 STIG-Hardened Kubernetes on OCI](#)
 - [13.2 Air-Gapped Enclave Delivery](#)
 - [13.3 Accredited Software Factory](#)
 - [13.4 Reproducible Compliant Baseline](#)
 - [13.5 AWS Kubernetes Platform](#)
 - [13.6 Image-Only Delivery](#)
 - [13.7 Infrastructure-Only Delivery](#)
14. [14 Getting Started](#)
- [14.1 Prerequisites](#)
 - [14.2 Install Forge](#)
 - [14.3 Verify the Installation](#)
 - [14.4 Understand a Crucible Description](#)
 - [14.5 First Build](#)
 - [14.6 First Deployment](#)
 - [14.7 Capture Dependencies](#)
 - [14.8 Generate Compliance Evidence](#)
 - [14.9 Teardown the Environment](#)
 - [14.10 Where to Go Next](#)
15. [15 Command Reference](#)
- [15.1 Command Structure](#)
 - [15.2 Global Options](#)
 - [15.3 Pull](#)
 - [15.4 Build](#)
 - [15.5 Upload](#)
 - [15.6 Initialize Platform](#)
 - [15.7 Deploy](#)
 - [15.8 Build and Deploy](#)
 - [15.9 Destroy Platform](#)
 - [15.10 Dependency Commands](#)
 - [15.11 Compliance Commands](#)
 - [15.12 Workspace Commands](#)
 - [15.13 Requirements Commands](#)
 - [15.14 Exit Codes](#)
 - [15.15 Logging](#)
16. [16 Implementation Status and Roadmap](#)
- [16.1 Status Model](#)

- [16.2 Current Phase 1 Status](#)
- [16.3 OCI Status](#)
- [16.4 AWS Status](#)
- [16.5 Air-Gap Status](#)
- [16.6 Compliance Status](#)
- [16.7 Planned Providers](#)
- [16.8 Phase 2 Management Application](#)
- [16.9 Known Limitations](#)

17. [17 Annexes](#)

- [Annex A Acronyms and Abbreviations](#)
- [Annex B Terms and Definitions](#)
- [Annex C Supported Operating Systems and Architectures](#)
- [Annex D Provider Capability Matrix](#)
- [Annex E Compliance Artifact Matrix](#)
- [Annex F External Toolchain](#)
- [Annex G Requirements Traceability](#)
- [Annex H Architecture Decision Record Index](#)
- [Annex I Example Crucible Description](#)
- [Annex J Example End-to-End Workflow](#)
- [Annex K Repository and Workspace Structure](#)
- [Annex L References](#)

-
- [Annexes](#)
-

© 2026 Dido Solutions, Inc. and Jackrabbit Consulting, Inc.

From:
<https://wiki.didosolutions.com/> - **Dido Solutions, Inc Wiki**

Permanent link:
<https://wiki.didosolutions.com/dido/02-crusible/start?rev=1784403223>

Last update: **2026/07/18 12:33**

