

# Project Release Planning (Jerma)

- [Overview](#)
- [Scope](#)
  - [Requirements](#)
- [Release Artifacts](#)
- [Architecture](#)
  - [High level architecture diagram](#)
  - [Internal Dependencies](#)
  - [External Dependencies](#)
- [Test and Verification](#)
- [Risks](#)

## Overview

<b>Project Name</b>	<b>Enter the name of the project</b>
Target Release Name	Jerma
Project Lifecycle State	Mature

## Scope

Release Engineering plans to achieve the following for the Jerma release:

- Stabilize Releng Verification Jobs
  - [RELENG-432](#) - Getting issue details... **STATUS**
- Execute on TAC Infra CI/CD [Recommendation](#)
  - [RELENG-433](#) - Getting issue details... **STATUS**
- Determine Status/Future of Subprojects
- Setup new OPNFV Build Machines
  - [RELENG-434](#) - Getting issue details... **STATUS**
    - Migrate OPNFV Build machines off UCS to LaaS VM
    - Standup new ARM builders as armbuild-3/4 are being taken offline

## Requirements

None.

## Release Artifacts

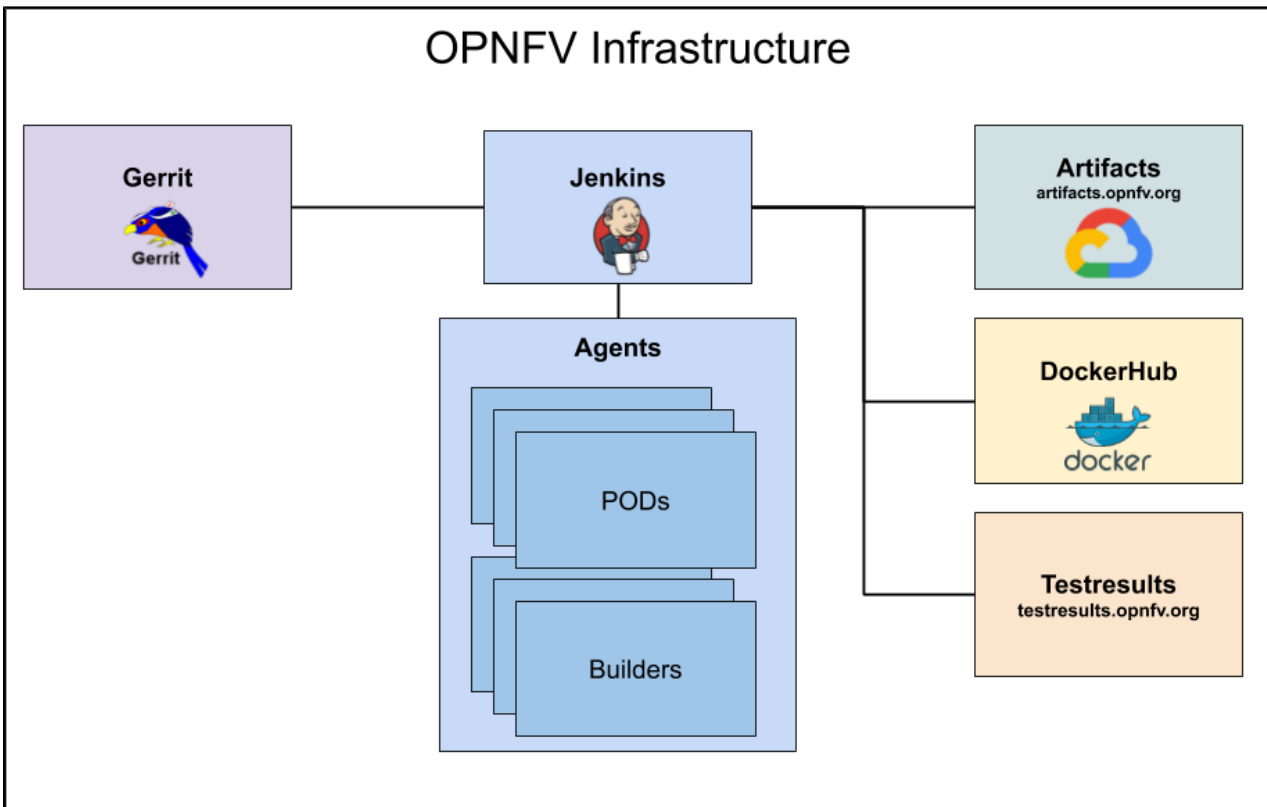
The Release Engineering project does not produce artifacts as the nature of the project does not allow it to be tied to specific versions, but on each release we tag the repository for easy reference when reviewing historical changes.

Indicate the work product (Executable, Source Code, Library, API description, Tool, Documentation, Release Note, etc) for this release.

<b>Name</b>	<b>Description</b>	<b>Format (Container, Compressed File, etc.)</b>
Releng Repository	Source Code	Git Tag

## Architecture

### High level architecture diagram



## Internal Dependencies

None

## External Dependencies

- [Jenkins Job Builder \(JJB\)](#)
- [global-jjb](#)

## Test and Verification

Jenkins jobs are tested manually using JJB from the command line and the OPNFV Jenkins [sandbox](#).

## Risks

List any risks and a plan to mitigate each risk.

Risk Description	Mitigation Plan