

Architecture

Project Name : Cloud SDN Automation Reuse

Version : 0.1

Date : 31st March 2016

Notes:

Juniper Project Management Methodology

**Document Control**

Document Information

|  |  |
| --- | --- |
| **©** | **Information** |
| Document Owner / Author | Simon Green & Ashley Burston |
| Issue Date |  |
| File Name | Cloud Automation-Reuse-Architecture |

Document History

|  |  |  |  |
| --- | --- | --- | --- |
| **Version** | **Issue Date** | **Author/s** | **Changes** |
| 0.1 |  |  |  |

Document Approvals

|  |  |  |  |
| --- | --- | --- | --- |
| **Role** | **Name** | **Signature for Acceptance of content** | **Date** |
| Project Managers |  |  |  |
| Project Director |  |  |  |
| Project Stakeholders |  |  |  |

**Table of Contents**

[1 Introduction 3](#_Toc447282217)

[1.1 Purpose of document 3](#_Toc447282218)

[1.2 The Vision 3](#_Toc447282219)

[1.3 The Objective 3](#_Toc447282220)

[1.4 Problem Statement 3](#_Toc447282221)

[1.5 How the Openstack Community is addressing these problems 4](#_Toc447282222)

[1.6 Extending OpenStack Kolla to deploy Software Defined Networking into Containers 4](#_Toc447282223)

[1.7 Using Ansible to automate SDN evaluation and test Architectures 4](#_Toc447282224)

[1.8 Competitors 5](#_Toc447282225)

[1.9 Scope 5](#_Toc447282226)

[In scope 5](#_Toc447282227)

[Out of scope 5](#_Toc447282228)

[1.10 Risks 5](#_Toc447282229)

[1.11 Dependencies 6](#_Toc447282230)

[1.12 Existing assets 6](#_Toc447282231)

[2 Appendix 6](#_Toc447282232)

# Introduction

## Purpose of document

This document describes the Cloud SDN automation re-use project architecture version 1. It provides a high level view of the technical aspects of the project and how all the various components interact to meet the project objectives. It contains a clear delineation of the project scope including what will be included in this release of the complete set of technical deliverables. Additionally, it provides a clear list of assumptions, dependencies and limitations that are understood to be part of this release.

## The Vision

The vision is to provide Juniper Consultants and their customers tools which can deploy small, fast, light test and evaluation environments for Contrail SDN and OpenStack. Lowering the bar for evaluation of SDN architectures from the Data Centre to the desktop.

We envisage that over time these tools will be merged with the existing underlay automation project to provide a fully automated deployment of test architectures within the underlay (vQFX, vMX, vSRX) ,as well as within the software defined overlay (Contrail SDN and CSO NFV).

These tools will enable Consultants and customers to evaluate the benefits of Contrail SDN without specialised expertise.

These tools and their generated environments will be deployable on low footprint hardware; whether that be laptops, virtual machines or bare metal servers.

We expect that these tools will over time be extended to provide alternative architectures to OpenStack, such as Kubernetics; as well as full NFVs such as CSO.

The hope being That these tools will allow Juniper PS to differentiate ourselves from out competitors by providing customers an easy entrance point to evaluate SDN.

## Why we are excited

After a short evaluation of Kolla we created a complete OpenStack with each process running within a Docker container built from the Kolla images. All running within one Virtual Machine on a two core laptop.

Using locally written scripts we could then destroy and redeploy the entire OpenStack cluster, all subsystems, onto a different network within 15 minutes.

With real development its easy to see how this technology would allow us to build automated OpenStack+Contrail environments in minutes; deploy them wherever Docker can run.

## The Objective

The goal of this project is to move contrail test environments from specialised hardware intensive bare-metal servers and virtual machines, to an automated and less specialised container based model.

We aim to automate the deployment and configuration of the containers, OpenStack and Contrail SDN, in a repeatable documented way. That allows Juniper consultants and customers to select and deploy Cloud architectures running OpenStack and Contrail SDN, within self contained environments for the purpose of evaluation and training.

The project leverages recent developments within the OpenStack community around the use of containers to simplify Cloud datacenter deployments. Enabling smaller, faster, portable and maintainable test and evaluation environments for customers and Juniper engineers interested in moving network services into a software defined overlay.

The complete package will be downloadable and provided along with a user guide & developers guide. The user guide will provide details of the installation, setup and usability. The developer’s guide will detail the technical steps required to create new or modify existing, templates & tests.

Consultants can then choose from a large library of existing cloud architectures, to utilise internally and for customer engagements.

## Problem Statement

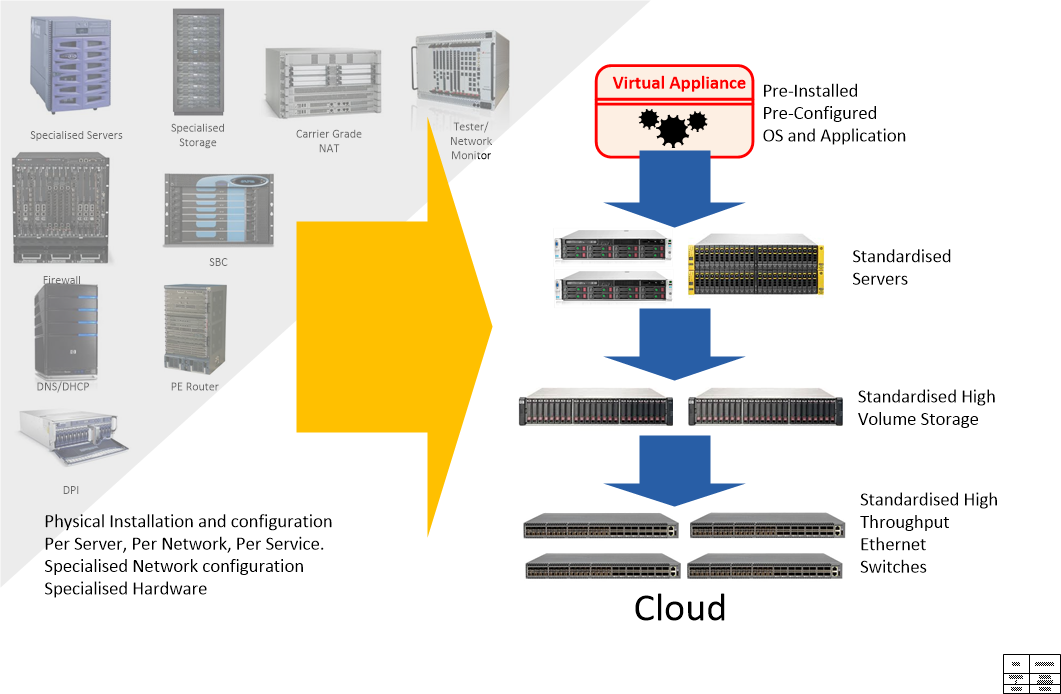


Figure 1: The Dream: Networks and Services running on commodity hardware, deployed in seconds..

The 2015 OpenStack user survey shows that 7% of all deployments globally are running Contrail SDN as a replacement for Neutron. The benefits of Contrail SDN are obviously well established. What is also clear is that these customers are not engaging Juniper Professional Services during their journey towards the Next Generation Data Centre.

Beyond the simplest all-in-one architecture software technologies such as OpenStack, Contrail SDN, Contrail Cloud Orchestration, require increasing amounts of dedicated hardware as well as specialised skill sets in order to deploy and run. This presents customers with a dilemma; while they see benefits in the flexibility of SDN within their Data Centres, they struggle to justify the effort required to deploy and evaluate next generation architectures for themselves. Tending to settle for the familiarity of classical underlay network on physical boxes.

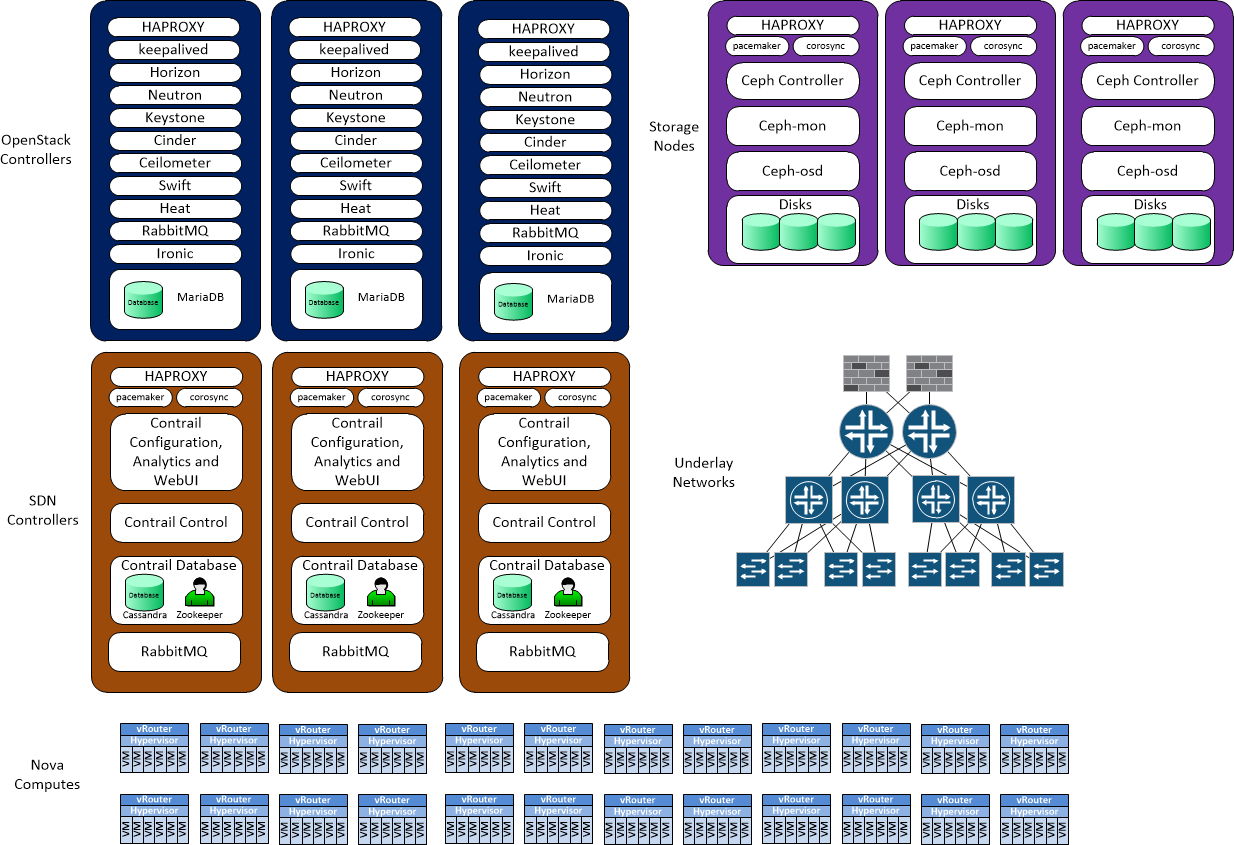


Figure 2: The Reality of a Cloud Data Centre Running OpenStack and SDN

Within Juniper, engineers face the circular argument of not being able to position next generation products such as Contrail SDN with their customers, without they themselves first gaining hands on experience.

Additionally, the demarcation between the software defined and the physical world requires customers to build classical underlay networks with Leaf, Spine, Router and Firewalls. Then to additionally engage a different skill set to deploy the Software Defined Overlay into the Cloud.

So in order to possession SDN with our customers we need to simplify the evaluation to the point where our consultants and our customers can deploy and play with real environments within minutes.

## How the Openstack Community is addressing these problems

Within the OpenStack community the complexity of deploying OpenStack commercially was formally recognised in 2015 when it adopted the Kolla project. The additional complexity of replacing OpenStack networking with SDN has never been addressed and is viewed as something for the experts.

Essentially OpenStack Cloud environments beyond the day one all-ion-one test setups quickly consumed tens of cores, multiple physical bare-metal servers with memory requirements to match. Deployment strategies for OpenStack range from DevOps do-it-yourself coding, to commercial platforms such as Mirantis and Redhat which take control of the whole Data Centre server infrastructure.

Project Kolla was initiated as a response, with a mandate to provides production-ready containers and deployment tools for operating OpenStack clouds. In practice project Kolla builds [Docker](http://docker.com/) containers and [Ansible](http://ansible.com/) playbooks which can then be used to deploy portable OpenStack architectures within minutes.

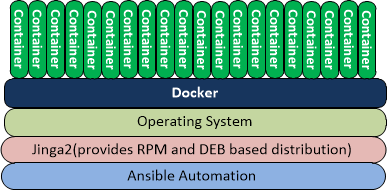


Figure 3: Kolla Stack

Project Kolla became a Big Tent project in the Juno release of OpenStack. While still in its infancy the project has reached a point of maturity where we see it as a useful platform for test and evaluation of OpenStack.

## Extending OpenStack Kolla to deploy Software Defined Networking into Containers

This project will extend the OpenStack Kolla project to deploy OpenStack and Contrail SDN fully integrated within containers. We see that the benefits of smaller, faster, automated deployments will help Juniper Consultants and their customers evaluate SDN as a technology that can move the network into the same position that OpenStack has moved bare metal Data Centres.

## Using Ansible to automate SDN evaluation and test Architectures

Juniper has existing teams within Professional Services who have already automated the deployment of the Data Centre underlay using Ansible playbooks.

The next logical step for professional services is to extend this level of automation into the software defined overlay.

Kolla already deploys OpenStack using Ansible Playbooks and Docker Containers. So is the platform of choice for extend into the Software Defined domain.

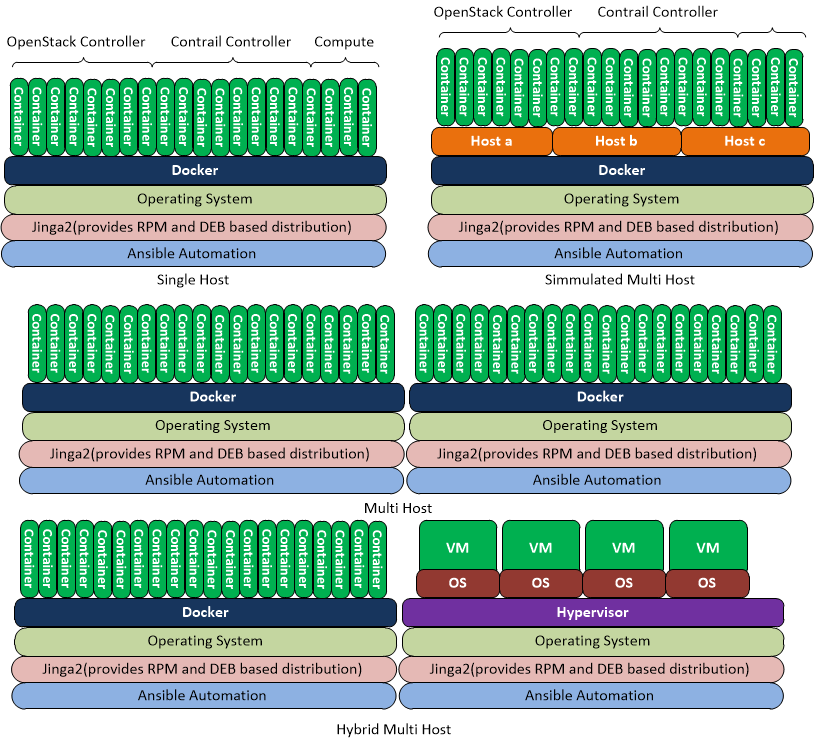


Figure 4: Ansible+Docker Automated SDN Deployments

## Competitors

While the concept of using containers for SDN evaluation may be unique to Juniper PS, our competitors are investing heavily in Kolla as an OpenStack deployment tool. Cisco for example have several developers dedicated to the Kolla project and use Kolla internally for testing OpenStack deployments.

<http://stackalytics.com/?user_id=&project_type=all&release=all&metric=all&company=cisco%20systems&module=kolla>

Junipers Contrail architecture may have an advantage over some of its competitors, in that all of the network components run as Linux user and kernel space code. So the Contrail architecture appears to lend itself appear well to containers. Nuage Networks for example run their controller as an emulated Virtual Machine, which limits their options in this space.

## Scope

### In scope

1. Build and deploy a repeatable design and test environment for development of container based SDN and OpenStack.
2. Extend OpenStack Kolla to deploy the Contrail management, Contrail controller and vRouter subsystems within Ansible playbooks and Docker containers.
3. Extend OpenStack Kolla to integrate the Contrail containers into an existing container based OpenStack.
4. Build tools to deploy fast, light weight, portable versions of the resulting tools and Docker images from a simple data set.
5. Extending the Kolla vagrant scripts to deploy Openstack and Contrail on VMWARE.
6. Provide the following:
   1. ARCH (Architecture) document
   2. Design document
   3. Software packages to Extend Kolla to deploy Contrail test environments.
   4. Playbooks to deploy defined test architectures.
   5. Software packages to generate a minimal environment to deploy the container based architectures.
7. Users Guide
8. Developers Guide

### Out of scope

While exciting we do not envisage containers as a method to deploy commercial Data Centres at this point in time. This project would be limited to deploying small, fast, light, repeatable test and evaluation environments. We do envisage multi-server architectures as being in scope.

CSO is today a series of micro services, each within its own super-resourced virtual machine. Running CSO within small test environments is currently impractical. Its clear that moving those micro services into Docker containers could make small CSO test environments a very feasible objective. However, CSO today runs Contrail SDN within it (Two instances), so this project plans to address the Chicken (Contrail SDN) and not the egg (CSO) at this point in time.

## Risks

1. While our initial analysis of Kolla demonstrates that OpenVSwitch can run within a fast, small, light containerized environment. We have no prior knowledge of SDN being moved into containers, so may find during the project investigation that its technically unfeasible at this point in time.
2. Kolla is a very new project and fast changing project. Maintaining develops within Kolla may become difficult to manage. To mitigate this risk we recommend that Kolla-Ansible is ran as a separate project which could in theory be merged back into OpenStack Kolla at some later date.
3. While Docker frequently cites the benefits of containerising micro services (light, fast, portable) verse virtualising services We are here pushing the boundaries of what’s possible within containers. We therefore cannot clearly evaluate which of these super-powers will in fact come true.

## Dependencies

1. Availability of Kolla on an Liberty OpenStack supporting all of the required software dependencies imposed by Contrail SDN.
2. Availability of Contrail release 3.

## Existing assets

The following assets were generated during the evaluation phase and are available to the project.

1. A virtualised portable development environment running a containerised OpenStack Liberty.
2. Procedures for building Kolla OpenStack from source
3. Architectural notes on Kolla’s architecture
4. Vagrant scripts for deploying Kolla on Virtualbox
5. Heat scripts capable of deploying test environments within OpenStack

# Appendix