i2 Private Cloud

OpenStack Reference Architecture

White Paper

Authored:
Richard Haigh - Head of Delivery Enablement
Steven Armstrong - Principal Automation Engineer

Reviewed:
Thomas Andrew - Senior Delivery Manager
Stephen Lowe - Director of Technology
Cathal Sheridan - Director of Technology
Paul Cutter - Chief Technology Officer

Published on www.betsandbits.com

V1.0 - OCT 2016 (C)

Contents

Abstract

This white paper documents the creation, purpose and benefits of Paddy Power Betfair's Private OpenStack Cloud. It is aimed at those looking to adopt a similar approach to an extensible, scalable and performant platform for their own production estate.

The paper documents the programme approach and presents the rationale for each individual technology supplier. It also explains the high level design around each technology. As such, a 'reference stack' is presented, which the paper hopes to promote.

The audience for this paper would include engineers involved in an OpenStack project and managers looking to assess the feasibility of such a project, or improvements to a current project.

The authors believe that an Open Source approach to infrastructure automation and tooling offers tangible advantage in terms of innovation and adaptability.

For any comments or feedback please contact richard.haigh@paddypowerbetfair.com.

Introduction to Paddy Power Betfair

Paddy Power Betfair plc was formed in 2016 from the merger of two of the fastest-growing online betting operators in the world; Paddy Power plc and Betfair Group plc.

Paddy Power was an international multi-channel betting and gaming group widely seen as one of the most distinctive consumer brands in Europe.

Betfair was an innovative online betting and gaming operator which pioneered the betting exchange in 2000, changing the landscape of the sports betting industry.

Paddy Power Betfair has a market-leading presence in the UK, Ireland, Australia and the USA, as well as a range of B2C and B2B operations across Europe.

In the context of this paper - the focus is on Betfair's infrastructure (which, importantly, includes the betting exchange). For context, performance requirements for the infrastructure estate underpinning the betting exchange platform are summarised as:

- 135M daily transactions
- 3.7Bn daily API calls
- 2.5TB daily log data output
- 120,000 time-series monitoring points per second
- 500 deployments per week

Whilst the OpenStack solution will be applicable for both estates - it was designed primarily to cope with the loads generated by the betting exchange platform and its associated services.

The Overarching Problem

Modern technology infrastructure needs to meet the demands of today's fast-moving digital businesses. The pace of change in application development has grown exponentially over the past few years, driven by the need to deliver new products and features to customers at an ever increasing rate. Those customers expect to use their favourite digital products around the clock, wherever they are. Balancing availability with the need to carry out maintenance is a key challenge. This, combined with improvements to the automation of application deployment, means that infrastructure teams can quickly become a bottleneck.

Paddy Power Betfair's existing infrastructure had evolved to keep pace with a rapidly growing company. To keep pace with future business growth, the organisation needed to make a fundamental change to the way it viewed infrastructure. Building on the learnings and experience gained from driving more agility in our development teams, and using the latest technology available, we created i2 - our next generation infrastructure. The name simply refers to the 2nd generation of infrastructure (i2).

An Introduction to i2 as the Solution

The i2 programme has delivered a new network and hosting infrastructure based on Infrastructure as a Service (IaaS) principles. Ultimately this will host all of Paddy Power Betfair's development, test and production applications and the associated tooling. The new infrastructure uses software defined networking (SDN) to reduce the reliance on physical network equipment and simplify network operations. This allows software applications to run independently in multiple data centres and balance customer traffic between them. Wherever possible applications run active-active across data centres to provide a high level of resilience.

This infrastructure was designed following an in-depth selection process, ultimately selecting the following technologies and vendors:

- Commodity x86 servers for ease of support and future scalability from HP
- Flash-based storage from Pure
- Leaf/Spine networking from Arista
- Routing and SSL-Offload from Citrix
- KVM Virtualisation and OpenStack tooling from Red Hat
- Software-defined networking from Nuage
- An open-source delivery pipeline allowing a high level of integration

I2 is designed with pre-production environments that closely match production environments to allow like-for-like testing, assuring delivery quality. It also provides performance and infrastructure test environments to allow destructive testing without impacting production.

Key Benefits

The new infrastructure will deliver the following benefits:

Faster, self-service environment provisioning, leveraging OpenStack APIs as the infrastructure middleware, reducing delays in creating environments and expediting the delivery of code to market.

An **API driven** approach to infrastructure so resources can be programmatically requested.

A **standard delivery tooling** approach, enabling economies of scale in support and mobility of staff between teams - further allowing easy future change or extended integration where appropriate.

Full traceability of firewall policies, mapped one-to-one to each micro-service applications.

A **Horizontally scalable** model allowing rapid growth as more compute resources are added.

Low latency network using leaf spine topology, increasing application performance.

A **standard monitoring stack**, enabling faster recovery from incidents.

High level of underlying infrastructure **availability** with reduced time to repair.

An infrastructure designed with **security risk control baked-in**.

Our Approach

Initial selection process with pitches from competing vendors. We ran an internal assessment of possible vendors and then invited a handful of suppliers into a request for proposal (RFP) process. The responses from this narrowed our selection. After further refinement to the RFP process we took two vendor groups through to a final selection stage. The final decision was made on a combination of RFP response, feedback from workshops and a final pitch for the project. This process took around 3 months to complete.

Proof of Concept to validate the architecture and vendor claims. The successful vendors were tasked with creating a Proof of Concept (PoC) implementation of a basic OpenStack environment with enough functionality for us to run functional and performance testing. This was designed to validate not only the core architecture and design, but also to validate RFP claims. This process took 6 weeks to complete.

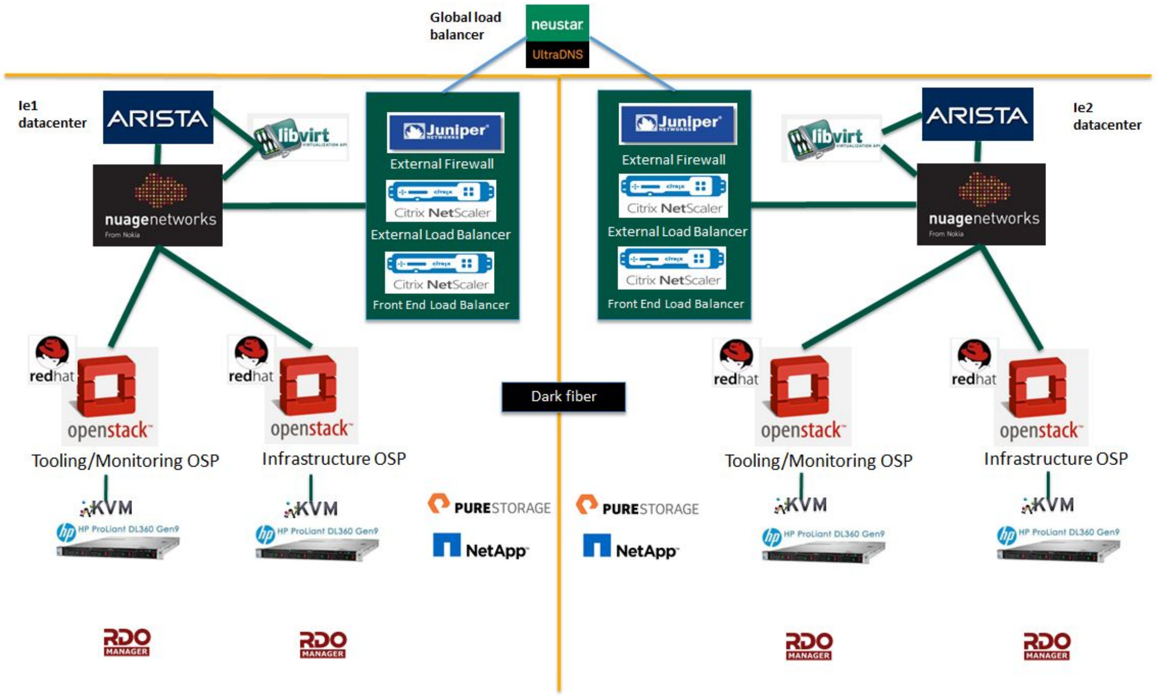
Production pilot project to build initial infrastructure, tooling and process. Following a successful PoC a project was initiated to build the first stage of the production estate. This would be a minimum viable product in terms of features but would include a lock-step build in two data centres, a fully resilient implementation within each data centre and a complete delivery tooling build. Two applications would be selected and each migrated to the new estate to handle 100% of their production traffic on the new estate. The pilot estate would contain around 100 hypervisors in total along with associated networking, storage and routing solutions. This process took 8 months to complete.

Migration of production components to new estate with changes to delivery process. This project commenced with an initial handful of 10 applications (Phase 1). A process of onboarding was designed for this phase. From this, learnings were taken to plan the migration of the remaining 150 applications. New applications (approximately 50 expected) were designed on the new estate from the start. Deprecated applications expired on the previous estate and therefore did not require migration. The Phase 1 process started in parallel with the Pilot project and lasted 10 weeks. The remaining migrations were expected to run for 12 to 18 months.

Continuing development of the OpenStack Estate. In parallel to the migration project the underlying OpenStack estate and its associated tooling was iteratively improved. Examples included purpose built test-lab environments and additional monitoring capability. A core team focused on this work. Whilst this effort started shortly after the PoC, it was expected to be a perpetual part of the OpenStack life-cycle and as such the timescales would be ongoing.

Decommissioning of existing equipment. Running in parallel to the migration project, this phase decommissioned the previous estate as its workload moved to the new estate. Ultimately this removed physical hardware from the data centres. The timescale for this ran alongside the migration project and was expected to also last around 18 months.

Reference Stack



Paddy Power Betfair runs an active-active data center configuration. This utilises UltraDNS to route traffic through an external Juniper firewall to two tiers of Netscalers. Tier 1 (External Load balancing) is used for SSL offload, while Tier 2 (Front End Load Balancing) is used to route traffic to the various applications. This design is mirrored across both data centers.

There are two OpenStack instances within each data center, hosting:

- **Tooling and Monitoring** - runs the delivery tooling and monitoring.
- **Infrastructure** - runs test environments and production workloads.

The network is governed by the Nuage Software Defined Network (SDN), whilst each of the four OpenStack clouds (two per data center) is deployed using Red Hat Director (Red Hat's version of the upstream Triple-O project).

Nuage's SDN platform (VSP) provides segregation between layer 3 domains - isolating test and production environments. These are further segregated using Nuage firewalls. The Nuage SDN is tightly coupled with an Arista driven leaf-spine architecture to minimise network latency. One Nuage SDN instance is present per data center to govern two OpenStack clouds and their associated layer-3 domains. The data centres are connected by dark fibre and both are direct mirrors of each other.

OpenStack - Red Hat

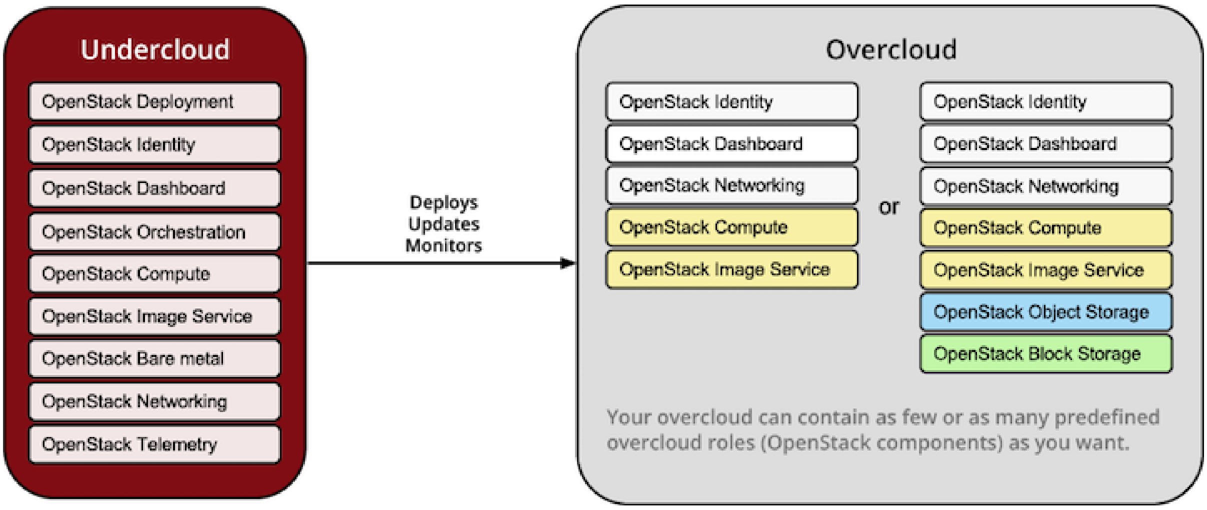
Requirements

- Ability to provision virtual and physical machines.
- Open APIs to use in our self-service workflow development.
- A highly available and redundant solution (both within and across data centres).
- Professional services to help initiate the project.
- A train/skill-up model enabling us to self-support.
- Avoid vendor lock-in.
- The freedom to select the software/hardware stack ourselves.

Technical Details Of Implementation

Each of the four OpenStack clouds is deployed using Red Hat's OpenStack Platform Director (OSP Director). This consists of a single installer instance (running virtualised), hosted in a small libvirt environment. These libvirt servers also host core services such as NTP, DNS and LDAP. This external libvirt environment is necessary so that it can control the instantiation of the OpenStack environment.

Each OSP Director VM has a one-to-one mapping with one of the four OpenStack clouds across the two data centers. OSP Director splits OpenStack into two tiers - called the overcloud and the undercloud. It uses a small undercloud version of OpenStack to deploy the OpenStack overcloud. The overcloud deployment comprises both OpenStack "Tooling and Monitoring" and "Infrastructure" clouds. The overcloud is split into the overlay and underlay networks.



The overcloud underlay consists of three OpenStack controllers, which host the OpenStack services. These are deployed on three bare metal servers. Each set of controllers are deployed in a high-availability (HA) fashion using pacemaker, running quorum, and deployed under the default Keystone version 3 domain. The controllers are deployed using heavily customised Heat templates by OSP Director. The Nuage plugin is installed on all controllers and neutron services are switched off.

KVM Hypervisors (compute nodes) are scaled out on each OpenStack cloud using the undercloud. They also use customised Heat templates that install the Nuage plug-in on each controller and the Nuage VRS component on each compute node. These compute nodes sit on the underlay network and are not accessible from the external API. The Nuage VRS is used to control the firewall in the overlay tenant network as opposed to using neutron security groups.

A Single OpenStack Controller is made up of the following services:

- **Horizon** is the OpenStack dashboard that users connect to view the status of VM or bare metal servers.
- **HAProxy** is a highly available proxy to the horizon dashboard.
- **Keystone** is the identity service for OpenStack allowing user access.
- **Glance** is the image service for OpenStack storing all VM templates.
- **Cinder** is the block storage service for OpenStack allowing Pure Storage volumes to be provisioned and attached to VMs or bare metal servers.
- **Nova** is the compute service for OpenStack used for provisioning VMs in the tenant network.
- **RabbitMQ** is the message queue used to queue requests for new VM's from self-service pipelines.
- **Galera** is the database used to store all OpenStack data in the Nova (compute) and Neutron (networking) databases; holding VM, port and subnet information.
- **Swift** is the object storage service for OpenStack (the backend for Glance).
- **Neutron** is the OpenStack networking service. When using Nuage the Neutron layer-2 and layer-3 agents are switched off and replaced by the Nuage plugin.
- **Nuage plugin** is the replacement for Neutron and integrates services with the Nuage VSD.
- **Ceilometer** is the OpenStack telemetry service into which Sensu monitoring integrates.
- **Ironic** is the OpenStack bare metal provisioning service.
- **Manila** is the OpenStack file share as a service project that allows management of Netapp NFS mounts and shares.

Each of the industry standard OpenStack APIs are available for all services. This allows delivery teams to interact with the infrastructure using self-service workflow actions from deployment pipelines.

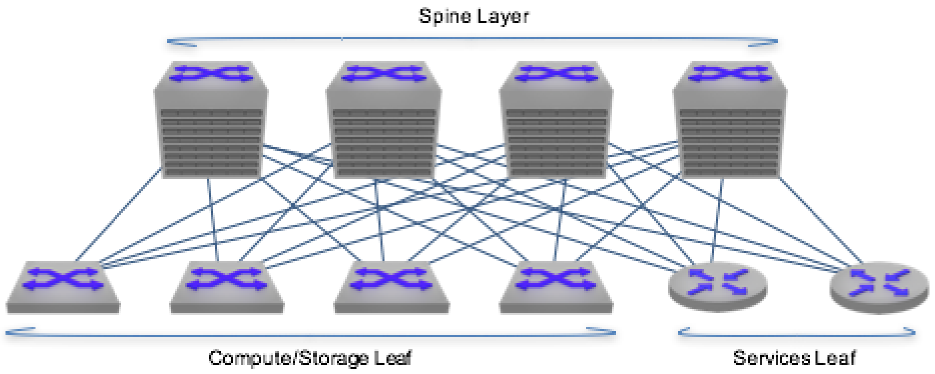
Switching - Arista

Requirements

- No vendor lock-in for our SDN or future networking solution.
- Full risk mitigation in case the primary SDN solution failed (back to layer-2).
- A deep-buffer option in spine switches to support mixed workloads.

Technical Details of Implementation

The i2 network design uses a Clos (https://en.wikipedia.org/wiki/Clos_network) leaf-spine architecture. Central to this is a series of leaf switches which form the access layer. These switches are fully meshed to a series of spine switches. The mesh ensures that access-layer switches are no more than two hops away from each other. This approach minimises latency and the likelihood of bottlenecks between access-layer switches and gives the ability to select and alter the appropriate switch architecture based on each new use-case.



The data center infrastructure is based on the following principles and uses Arista to create the leaf-spine architecture. This gives the ability to scale out by simply adding further racks.

General specification:

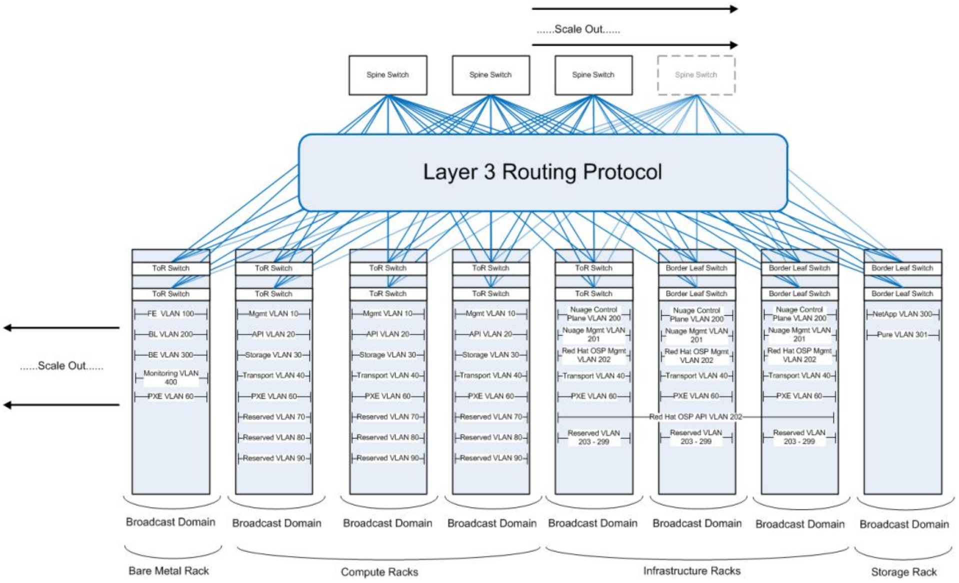
- Deep buffer Arista 7500 series spine switches are used to best support mixed workloads, future developments, and provide a path to higher capacities including 100G and beyond.
- Each rack contains 2 Arista 7050X leaf switches which are configured for redundancy using MLAG (https://eos.arista.com/mlag-basic-configuration/).
- Layer-2 domain segmentation is at rack level

Several types of racks are used, hosting various components:

- Bare Metal rack: hosts single-use physical compute.
- Compute rack: hosts Hypervisors to support application / micro services instances.
- Storage rack: hosts Pure Storage.

There are also specific infrastructure rack designs hosting:

- Control elements (VSD HA cluster, pair of VSC, OpenStack controllers).
- External connectivity outside DC (pair of 7850 VSG) – North-South traffic.
- Connectivity with Bare Metal servers within DC (pair of 7850 VSG) – East-West traffic.



Arista switches are updated using Arista ZTP (Zero Touch Provisioning) once they are racked and cabled in the data center. These pull configuration from the Arista's CloudVision Exchange (CVX) platform which contains all routing information for the Arista switches. Arista's CVX platform provides single point abstraction of the physical infrastructure to the Nuage SDN controller.

Software Defined Networking - Nuage Networks

Requirements

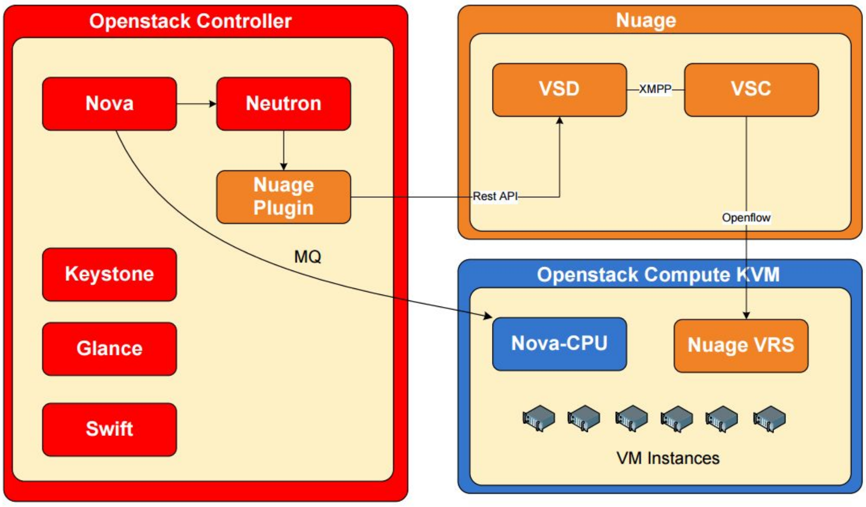
- Ability to horizontally scale the network.
- A highly available and redundant solution.
- Ability to bridge the i2 environment with the existing network.
- Ability for developers to self-serve firewall changes.
- Supplier consulting team to initiate the project.
- A train/skill-up model to leave us self-supporting.

Technical Details Of Implementation

The Nuage Virtualised Services Platform (VSP) consists of three different components: VSD (Director), VSC (Controller) and VRS (Service). To power a single OpenStack cloud we run three OpenStack controllers. These run in quorum using Pacemaker. A Nuage plugin is installed on each of the three OpenStack controllers. These are used to power each of the four OpenStack clouds across each of the two data centres. By using the Nuage networking plugin there is no need for an OpenStack network node with a Neutron L3 agent. Therefore, the Neutron agents are switched in favour of the Nuage plug-in.

The Nuage solution distributes the computation of the firewall rules to the controller and the hypervisor compute layer. This allows it to scale massively when compared to Neutron's approach of using a centralised model. A further advantage of this distributed firewall model is that no equivalent centralised firewall solution is required - such as an OpenStack Network Node or other commercial firewall solution.

The Nuage plugin integrates with the OpenStack controllers and KVM compute using the following workflow:



When a Neutron command is issued to OpenStack, the Nuage plugin uses REST calls to communicate with the Nuage VSD and vice-versa. This then communicates with the VSC using XMPP (https://xmpp.org/).

Openflow (which allows network controllers to determine packet path) allows the Arista CVX, Nuage VSC and VRS (Openvswitch) to communicate. The Nuage VRS is deployed on each KVM hypervisor compute node.

The Nuage OpenStack plug-in can support two modes of operation:

1. OpenStack-managed mode.
2. VSD managed mode.

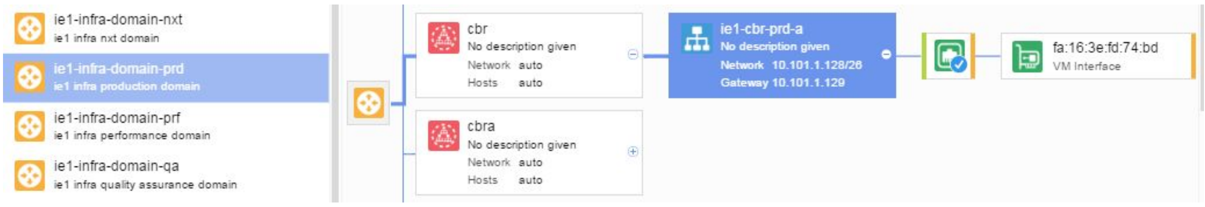
OpenStack-managed mode requires no direct provisioning on the VSD, all commands are issued via Neutron, however functionality is limited to those commands that Neutron supports.

Paddy Power Betfair uses the VSD managed mode, which allows use of the rich network feature set provided within the Nuage VSP platform. All networking commands are provisioned directly via the VSD using the Python SDK. All networks that are created in Nuage are replicated in OpenStack with a one-to-one mapping. ACL rules are governed using Nuage in VSD managed mode as opposed to OpenStack Security Groups.

Nuage creates its software based object model with the following constructs:

- **Organisation**: Governs all layer-3 domains.
- **Layer-3 domain**: Segments different environments so teams cannot hop from test environments to production. A GRT (https://www.aaai.org/ojs/index.php/aimagazine/article/viewFile/1573/1472) Hub domain is used to connect the i2 network to the Paddy Power Betfair native network. Each layer-3 domain by default inherits from a Nuage layer-3 domain template with an implicit drop. As such, when a child zone is created from a layer-3 domain, the zone has a Deny-All at egress and ingress level by default. This is then propagated down to the layer-3 subnets of the zone.
- **Zones**: A zone segments firewall policies at the application level, so each micro-service application has its own unique zone per layer-3 Domain.
- **Layer-3 Subnet:** VMs or bare metal servers deployed in OpenStack reside in these subnets.

In this example we see the Nuage object model with the layer-3 domain: ie1-infra-domain-prd, the child zone: cbr, one subnet: ie1-cbr-prd-a and one VM deployed on that subnet.



ie1-infra-domain-nxt
ie1 infra nxt domain

ie1-infra-domain-prd
ie1 infra production domain

ie1-infra-domain-prf
ie1 infra performance domain

ie1-infra-domain-qa
ie1 infra quality assurance domain

cbr
No description given
Network auto
Hosts auto

cbra
No description given
Network auto
Hosts auto

ie1-cbr-prd-a
No description given
Network 10.101.1.128/26
Gateway 10.101.1.129

fa:16:3e:fd:74:bd
VM Interface

When delivery teams on-board to i2 they are allocated two /26 network subnets per environment. This allows them to deploy A/B immutable subnets, with each release tied to a unique subnet that alternates. When a release is complete, the previous VM's and subnets and ACL rules are automatically cleaned down. This keeps the environment clean over time.

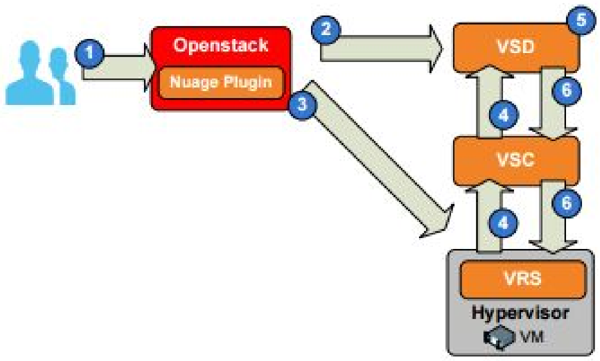
Subnets are mapped one-to-one between OpenStack Neutron and Nuage so appear in both systems. Nuage is the master.

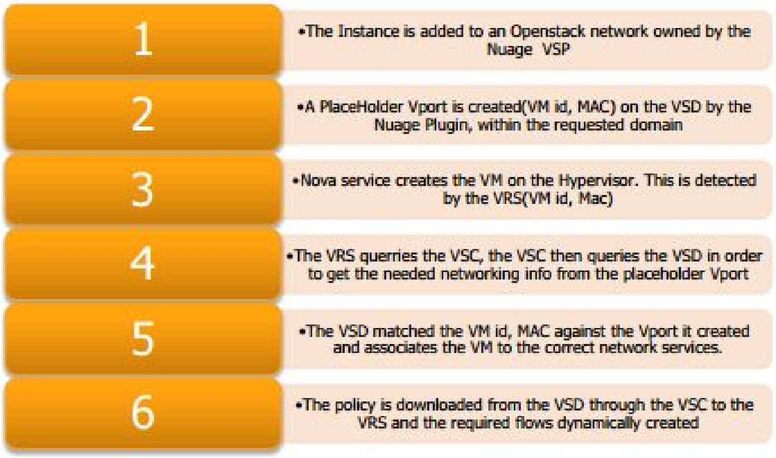
In this design, external connectivity uses a Nuage Networks hub domain. This is used as a common exit-domain across all other layer-3 domains (in our case - domains for QA, Integration, Pre-Production and Production environments) and across both OpenStack deployments per site.

Firewall rules are applied to a unique policy entity for each micro-service application per layer-3 domain. These are divided into two categories:

- Common ACL Rules: Currently governed by the Infrastructure Automation team, allowing all machines access to pre-requisites such as LDAP, NTP, DNS and Citrix Netscalers. These are examples of rules that do not need to be set by developers.
- Application Specific ACL Rules: These open up the bare minimum of ACL rules required for the application to operate, opening only the required ports between applications. This allows the Developer teams to only worry about the connectivity their specific application requires - and to not have to consider core pre-requisites.

The firewall policies are enforced at each KVM hypervisor using the Nuage VRS. When a new VM is created on a Nuage managed subnet the following process flow is invoked:





1	•The Instance is added to an Openstack network owned by the Nuage VSP
2	•A PlaceHolder Vport is created(VM id, MAC) on the VSD by the Nuage Plugin, within the requested domain
3	•Nova service creates the VM on the Hypervisor. This is detected by the VRS(VM id, Mac)
4	•The VRS querries the VSC, the VSC then queries the VSD in order to get the needed networking info from the placeholder Vport
5	•The VSD matched the VM id, MAC against the Vport it created and associates the VM to the correct network services.
6	•The policy is downloaded from the VSD through the VSC to the VRS and the required flows dynamically created

Nuage uses a distributed firewall model allowing it to scale massively. This approach is not seen in the model adopted by native Openstack Neutron and many SDN vendors, which rely on a centralised ruleset computation and centralised layer-3 agents.

In these centralised models, additional layer-3 agents can be introduced to avoid bottlenecks but it is architecturally cumbersome. The new Openstack Dragonflow project has been created to meet similar scale challenges, but at the time of writing, this is still in its infancy. Nuage has reference clients that scale beyond 10,000 compute nodes using one Openstack installation so scale is not a concern.

As part of the Paddy Power Betfair's design for ACLs in Nuage, a DENY policy is applied in the layer-3 domain templates. As a result of this rule, applications sitting in a zone in i2 cannot access subnets outside the layer-3 domain unless explicitly allowed.

Nuage uses a term called Network Macros to refer to external networks that can be described with an IP prefix. In order for applications to communicate with external networks or with other applications that have not yet been on-boarded to i2, they can specify a Network Macro to the specific ranges that the application requires.

Routing/Load Balancing – Citrix Netscalers

Requirements

- SSL offload
- “Production like” pre-production environment
- Easy migration path from current Netscaler estate (complex Layer 7 content switching)

Technical Details Of Implementation

A single Netscaler solution with one MPX and SDX (https://www.citrix.com/products/netscaler-adc/platforms.html) is used for pre-production in each of the two data centres. This is segmented into QA, Integration and Performance testing environments. Two Netscaler MPX and SDX appliances are used for the production environment to give redundancy.

These are organised into two tiers:

- External Load Balancer (Tier 1) MPX on the Netscalers for dedicated SSL offloading.
- Front End Load Balancer (Tier 2) SDX for context switching and routing.

The single pre-production MPX and SDX is segregated into the various testing environments (QA, Integration etc) with separate GSLB managing two clusters of VPX HA pairs. These run in a master/slave configuration per environment. Each of these environments are configured with the same settings as production, although VPX clusters are confined to one SDX.

Production environments use two SDX appliances at Tier 2. Each instance of the VPX HA pair clusters are split across different SDXs for redundancy. Each of the two VPX HA pair clusters are managed using GSLB at tier 2.

These load balancers are integrated using the Nitro Python SDK. Paddy Power Betfair has written self-service workflow actions to manage all Netscaler operations; to meet developers' load balancing needs, to create VIPs, scale up services, bring new releases into service and set-up unique health monitors for the micro-service applications.

All VMs or bare metal servers in OpenStack are added as services bound to LBVServers (load balancing virtual services) at the bottom of the Tier 2 front end load balancers. Each service has a unique health check monitor to immediately mark the service as down if it's health check fails. Above the LBVServers are a set of CSVServers (content switching virtual services) which perform the layer-7 content switching and routing of requests to the appropriate LBVServer. Each CSVServer is bound to a GSLBVServer (global server load balancing virtual services) which allows redundancy at Tier 2.

Tier 1 maps an external IP address to tier 2 via the CSVServer context switches. The Citrix Netscalers are an external network to Nuage so routable on allowed ports.

Central Storage - Pure

Requirements

- Performance
- Ease of management

Technical Details Of Implementation

Pure storage is integrated to OpenStack using the available Cinder driver. This allows Pure storage volumes to be managed using the OpenStack APIs allowing us to attach volumes to VMs for applications that have high performance needs. Pure Storage is managed using Cinder's common set of APIs as opposed to going directly to the Pure storage API. This is our preference as it abstracts the infrastructure through OpenStack allowing us increased future portability of technology.

In our implementation Pure storage volumes are used to host production Betting Exchange databases. Using Pure storage "snap volume clone" production database volumes can be cloned, imported into OpenStack Cinder from database standbys, data sets anonymised, then attached to VMs to create database test environments. This allows test databases to be frequently refreshed allowing developers to test against production-like anonymised data sets.

Pure storage Cinder volumes are also being utilised to virtualise performance heavy micro-service applications. These were previously only deemed capable of running on bare metal servers. Instead of hosting these applications on local disk, volumes are attached to VMs using Cinder and mounted. This gives the applications the performance benefits of an all-flash solution, aided by the portability of a virtualisation solution, while at the same time completely governing it by the rich OpenStack API.

Compute - HPE ProLiant

Requirements

- Commodity x86 based.
- 1U ‘pizza-box’ for ease of future maintenance when compared to blade enclosures.
- A rich API to allow automation of set-up activities.

Technical Details Of Implementation

Paddy Power Betfair currently implements ProLiant DL360 Gen9 1U servers for all of its compute. These are managed using HP Oneview to configure RAID configuration and server templates. The main driver for initially choosing HPE was their early commitment to the development of the TripleO upstream project that became Red Hat OSP Director. This had been proven to work at scale in the HPE Helion OpenStack public cloud and multiple private cloud implementations.

Red Hat OSP Director utilises Ironic (https://wiki.openstack.org/wiki/Ironic) to deploy the undercloud and all hypervisors. We have engineered an automated scale-out process based on the ability for the hardware to be programmatically inspected and securely deployed using Swift as a backend for Glance.

Ironic features allow:

- PXE-less deploy with virtual media.
- Automatic detection of current boot mode.
- Automatic setting of the required boot mode, if UEFI boot mode is requested by the Nova flavour’s extra spec.
- Booting the instance from virtual media (netboot) as well as booting locally from disk.
- UEFI Boot Support and UEFI Secure Boot Support.
- Passing management information via a secure, encrypted management network.
- Support for out-of-band cleaning operations.
- Support for out-of-band hardware inspection.
- HTTP(S) Based Deployment.

Utilisation of bare metal servers in the overcloud tenant network is planned for phase 2 of the project. Bare metal will be secured by Nuage layer-3 domains and ACL rules.

Delivery Tooling

Requirements

- Best suited tools at each stage.
- Consolidation of current tooling.
- Modular to allow easy future tool replacement.
- Open Source where possible for community effect.

Technical Details Of Implementation

The OpenStack Shade library maintains the compatibility of the OpenStack APIs so self-service modules are backwardly compatible.

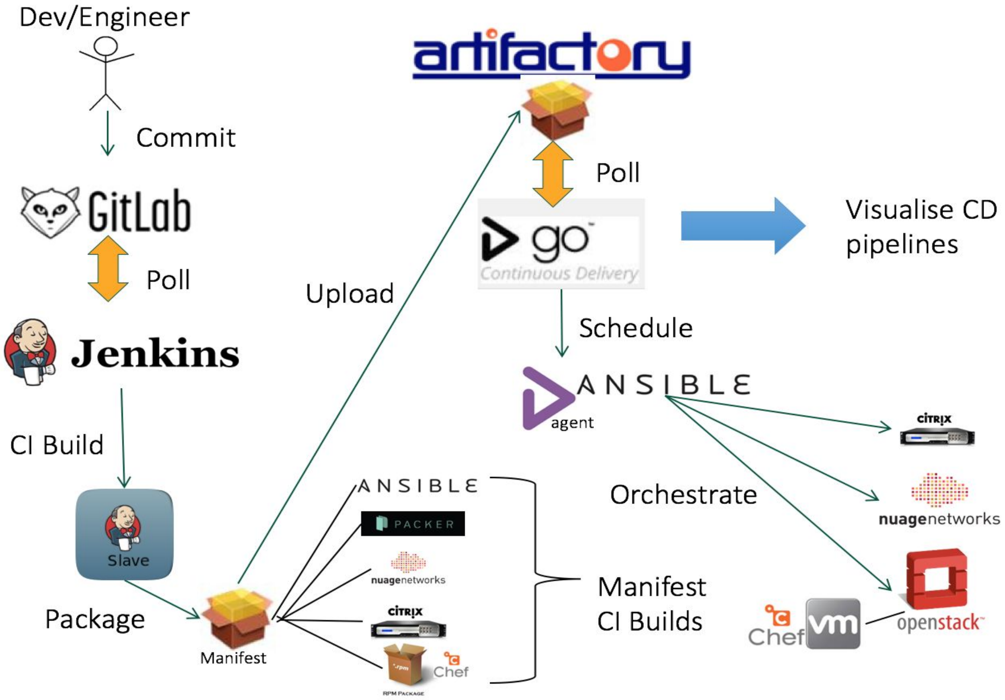
All OpenStack Ansible 2.x modules use OpenStack Shade as a superclass to maintain inter-cloud operability. Shade works with any OpenStack distribution and, by proxy, so does Ansible - thus removing vendor lock-in.

Ansible core modules are used in all pipelines and some custom OpenStack modules by extending Shade libraries.

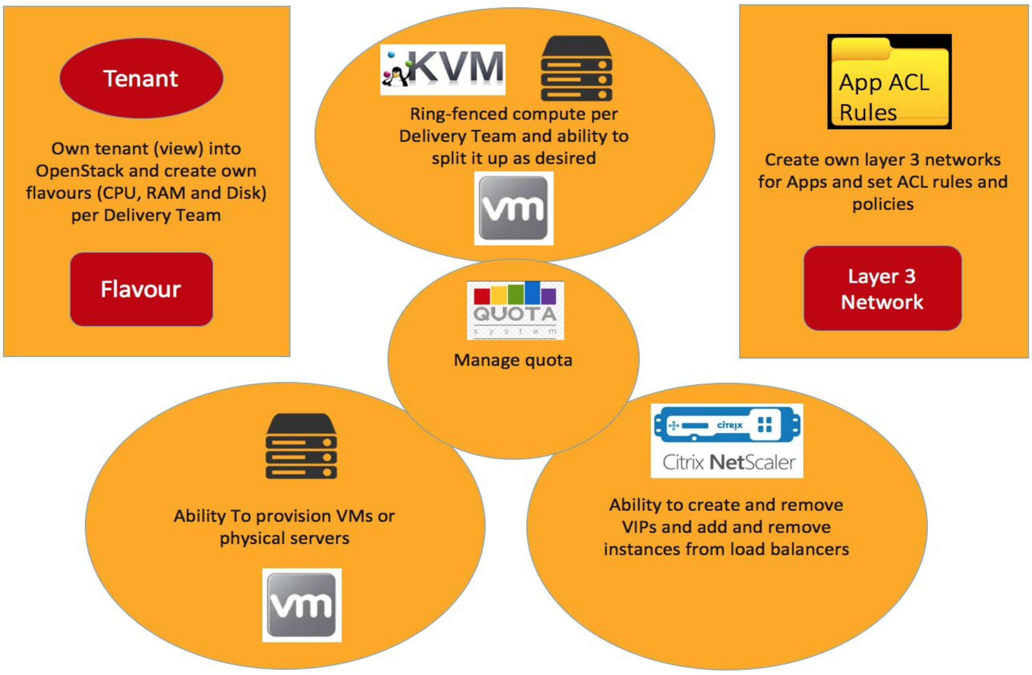
Nuage and Citrix Netscaler modules, although outside the core OpenStack API, have also been created in Ansible allowing a common automation approach..

Ansible playbooks are used to govern the available deployment pipeline operations. This, along with the common set of delivery tooling is used to give all development teams a consistent way to: spin up and create layer-3 subnets, launch VM's and bring releases into service on the load balancer. As such, Delivery teams can focus on developing the application as opposed to worrying about the infrastructure.

Each of the delivery tools used by Paddy Power Betfair in the i2 pipelines are displayed below:



Delivery teams, using the self-service workflows, have the ability to control the following variables in their pipelines:


Tenant
Own tenant (view) into OpenStack and create own flavours (CPU, RAM and Disk) per Delivery Team
Flavour
KVM
Ring-fenced compute per Delivery Team and ability to split it up as desired
vm
App ACL Rules
Create own layer 3 networks for Apps and set ACL rules and policies
Layer 3 Network
QUOTA system
Manage quota
Ability To provision VMs or physical servers
vm
Citrix NetScaler
Ability to create and remove VIPs and add and remove instances from load balancers

This combination of tooling and developer-centric capabilities allow true self-service of infrastructure consumption. This reduces our time to market and increases the quality of our product.

Future Roadmap

The i2 project, whilst now in production, has a long list of desirable future features.

1. **Immutable OpenStack clouds** for upgrades: Through Nuage we have the ability to have multiple OpenStack instances mapped to a single Nuage VSP installation, using net-partitions. This allows subnet A to exist on an old Openstack version and the new subnet release to exist on a new Openstack version. With a switch of API endpoints a team can migrate their applications to the new OpenStack version while still integrating with applications on the previous OpenStack release. Once all applications are migrated the old release of OpenStack will be decommissioned. Resources such as compute will then be recommissioned into the new OpenStack release. Cinder volumes can be imported in advance for stateful applications. This approach avoids the pain of OpenStack in-place upgrades and also allows Paddy Power Betfair to potentially skip OpenStack releases as breaking changes can be managed.

2. OpenStack **Freezer** for OpenStack Neutron and Nuage database synchronised backups, allowing for faster disaster recovery.

3. **Ironic** tenant network deployments secured by Nuage layer 3 policy.

4. **Sahara** project for managing Hadoop.

5. Utilisation of **Manila** for managing NFS Netapp workflow for VMs creating mounts and setting permissions.

6. **Load balancing** that doesn’t hairpin out to physical devices but allows east to west traffic to flow inside the tenant network in a fully OpenStack integrated fashion.

7. Utilisation of Nuage for **3rd party VPN** connectivity to layer 3 domains.

8. **Cinder replication** across OpenStack clouds and Datacenters.

For more information please contact richard.haigh@paddypowerbetfair.com, steven.armstrong@paddypowerbetfair.com or visit our engineering blog: www.betsandbits.com