Reference Design & Best Practice
Pivotal Ready Architecture

Executive Summary

Pivotal Ready Architecture (PRA) is an integrated development platform based on cloud computing and storage. It offers high performance potential, convenience, and addresses the challenges of creating, testing, and updating applications in a consolidated production environment.

Key Benefits

PRA enables you to quickly and reliably create an integrated development platform and provides a variety of deployment options to meet your needs. It is a tested PCF deployment that uses the scalable, hyper-converged Dell EMC VxRail Appliance and VMware vSphere with VMware vSAN and NSX-T. With PRA, you can deploy a production-ready PAS environment or PKS Kubernetes clusters more quickly and more reliably than when using a do-it-yourself approach.

The PAS and PKS deployments are designed to meet application operational needs. Dell EMC, VMware, and Pivotal Services can help with installing and testing PRA in your IT environment. IT professionals tasked with planning compute, network, and storage capacity can work with Dell EMC Services to determine the sizing requirements for these resources.

PRA enables developers to publish, run, and scale legacy, COTS, and cloud-native applications. The PRA platform is based on an integrated infrastructure that enables IT operators to easily manage and scale the development environment with tools such as VxRail Manager and PCF Operations Manager.

With PRA, you benefit from:

- Reference architectures for a variety of highly available Pivotal software deployments
- Private cloud and infrastructure services with simplified deployment
- Highly available platform on which to develop, and quickly deploy applications, reducing the time for application delivery
- Modern developer platforms, that boost developer productivity by combining application services, service discovery, container management, and orchestration within an ecosystem of developer tool plug-ins
- Consistent cloud solution, providing increased reliability, portability, and agility
Versions and Bill of Materials

Validated Versions And Interoperability For This Release

This recommended architecture includes VMware vSphere and NSX-T, a software-defined network virtualization that runs on VMware ESXi virtual hosts and combines routing, firewall, NAT/SNAT, and load balancing.

<table>
<thead>
<tr>
<th>Supported Software Components For Interoperability</th>
</tr>
</thead>
<tbody>
<tr>
<td>VxRail HCI System</td>
</tr>
<tr>
<td>VMware vSphere ESXi</td>
</tr>
<tr>
<td>VMware vSphere vCenter (VCSA)</td>
</tr>
<tr>
<td>VMware vSAN</td>
</tr>
<tr>
<td>VMware NSX-T</td>
</tr>
<tr>
<td>Pivotal Ops Manager</td>
</tr>
<tr>
<td>Pivotal PAS</td>
</tr>
<tr>
<td>Pivotal/VMware PKS</td>
</tr>
<tr>
<td>VMware NSX-T Container Plug-in for PCF</td>
</tr>
<tr>
<td>Dell EMC ECS</td>
</tr>
</tbody>
</table>

To use all the features listed here, NSX-T requires at least Advanced licensing from VMware when used with PAS. The equivalent of that licensing is included with the PKS product.

The following Bill of Materials was used to validate this interoperability matrix. These systems were selected as broadly representative of the common customer installations and do not represent an exhaustive list of possibilities.

<table>
<thead>
<tr>
<th>Lab Validated Components BOM</th>
</tr>
</thead>
<tbody>
<tr>
<td>VxRail Node Type</td>
</tr>
<tr>
<td>VxRail HCI System Software</td>
</tr>
<tr>
<td>VxRail BIOS</td>
</tr>
</tbody>
</table>
Minimum Viable Platform

Starting small is a great way to get started with container-ready infrastructure. You can read more about each of the key components in subsequent chapters but at this point we introduce the smallest individual configuration recommended for getting started. This configuration is considered stable and functional for trials and small application deployments (requiring only host level fault tolerance) and remote/branch office operation.
PKS Starter Kit:
This configuration includes the following key components:

1. VxRail hyper-converged server in a quad deployment
2. VxRail HCI System deployed for server and vSphere provisioning
3. Leaf switching in a redundant configuration to connected to a spine switch
4. vSAN storage configured for full system use
5. NSX-T software-defined networking for either PAS, PKS or both
6. App container platform of choice (PAS and/or PKS from Pivotal)
7. Host groups as needed to organize components for simple HA

There are a few trade-offs made at this low level of platform worth considering, including:

1. Overall capacity for apps is shared with the infrastructure
2. Management and operational components are kept to a minimum to reduce waste and also reduce recovery after a failure event
3. Storage is at the minimum viable configuration for stable operation during service or unplanned outages

Opportunities for growth will be discussed in subsequent chapters.

Primary Technical Concepts

Dell EMC VxRail and VMware vSphere

PRA is based on VxRail hyper-converged infrastructure (HCI) hardware from DellEMC. A hyper-converged solution is made up of servers, network switches and management software to create a complete set of data center resources in a single, collapsed system. There is no need to separate storage systems from network systems from compute systems. The VxRail solution includes tools to organize servers into clusters of virtualized compute capacity, establish and format a shared storage target from individual drives in servers and align the networking ports to the needed jobs, all thru automation.

Further, VxRail establishes a lifecycle management system to keep the clusters patched and current with VMware and releases through further automation.

Overall, the purview of VxRail HCI System Software is to take servers in a raw state and deliver a fully paved vSphere installation, complete with vCenter Server managing cluster(s), vSAN shared storage and related constructs.

VMware NSX-T Software Defined Networking

Virtualized, software-defined networking is a core concept adapted to the PRA solution. This important infrastructure layer brings maximum flexibility and growth potential for PRA from small to large systems. NSX-T brings a few key technologies into play that benefit the solution:

1. Tunneled networks of near-infinite capacity for Orgs, Namespaces and Apps to use
2. Bridged interfaces to the datacenter network core that can be expanded and clustered for speed, capacity and redundancy
3. Routing interfaces that are defined and deployed logically for a variety of needs to meet Org, Namespace and App needs, including routing directly to the app container
4. Load balancing logical instances for Apps and Namespaces on-demand

The PKS software has native support for NSX-T built into the product and requires an NSX-T installation to be complete prior to use. PAS software is also compatible with NSX-T but does not strictly require its presence.

Architecture Review

The overall architecture to PRA is as follows (bottom up view):

- A VxRail HCI cluster, or series of clusters, are organized using VxRail Manager
- vSphere is paved onto the cluster(s) via VxRail Manager
- A vSAN shared storage target is organized from local disks in each server per cluster
- NSX-T is layered on and a routed interface to the datacenter is established
- The container platform of choice is layered over the payload cluster(s) and interfaced with NSX-T for dynamic deployment of networks, routers and load balancers

All of these components build on each other to establish a horizontally-scalable - via host addition to a new or existing cluster - and vertically-scalable - via Memory/Disk/Network addition to existing hosts - container-ready system that is self-contained and self-healing.

Growth Beyond Minimum Viable Platform 1

Considerations and Growth Vectors

Organic growth from MVP1 is expected and encouraged. The next sections will offer considerations on growing from a lower to a higher level of fault tolerant architecture progressively.
Key items that change are:

1. Two more racks are added to the original, making three racks
2. Each rack represents a vSAN fault domain and vSphere Host Group, equivalent to a PCF availability zone
3. Management functions do not change location
4. The maximum number of hosts per installation is limited to 64 due to vSAN Cluster limitation. Migration from the original rack to using the added racks is possible with PCF by modifying vSphere host group definition and performing a `bosh recreate`. The PaaS will deploy all components that can be deployed as multiples will be into equal distributions across both AZs (as long as multiples selected are evenly divisible by the number of AZs used).

Repaving is a good choice at this stage, as it will place the singleton and multiple element components in the correct locations.

Multiple Clusters Design

This design is considered production-ready with ample high availability and redundancy capacity for full-time enterprise use. This size is appropriate for combined PKS and PAS deployments combined, if desired, using Resource Pools to organize components of each into the clusters.

Key items that change at this point in growth:

1. A total of four clusters are deployed: one for management of the whole system and three for payloads generated by the system.
2. All management functions are focused onto a single cluster of six hosts with vSAN. The remaining three clusters are AZs for PCF product(s) and for payload (app) use only.
3. All PaaS components that deploy in multiples are deployed in triples to take advantage of the multiple availability zones. HA features of both the IaaS and PaaS are brought to bear on the AZs.

Migration from the starter kit may be a challenge, as all management VMs from vSphere, NSX-T and Pivotal will be migrated to a dedicated cluster and not blended in with any of the app containers created by PKS and/or PAS.

A fresh install of the PaaS layer makes this model the easiest to install.

Repave is a good choice at this stage, as it will give the opportunity to place management components in the proper cluster and evacuate any pre-existing clusters of anything other than payload (app) components.
PKS Multiple Cluster Design:
Storage

PAS Storage

Pivotal recommends the following capacity allocation for PAS installations:

- For production use, at least 8 TB of data storage, either as one 8 TB store or a number of smaller volumes adding up to 8 TB. High churn or heavily used development foundations may require significantly more storage to accommodate new code and buildpacks.

- For small installations without many tiles, 4-6 TB may suffice.

The primary consumer of storage is the NFS/WebDAV blobstore included in PAS. This job can be externalized to a hardware or emulated software S3-compatible blob store (see below).

PAS Blobstore Options:

**Elastic Cloud Storage**

Dell ECS object storage features a flexible software-defined architecture that provides the simplicity and low-cost benefits of the public cloud without the risk, compliance, and data sovereignty concerns.

PRA can use the ECS platform as its blobstore location to store application code, buildpacks, and applications. The blobstore uses the S3 protocol and can be managed by ECS nodes. The ECS platform provides software-defined object storage that is designed for modern cloud-scale storage requirements.

**Virtustream Storage Cloud**

Virtustream Storage Cloud is a hyper-scale, enterprise-class, public cloud object storage platform, built for resiliency and performance. Virtustream provides a unique value proposition with enterprise-class off-premises primary and backup storage.

**Pivotal Internal Blobstore**

Pivotal provides an internal WebDAV blobstore that PAS uses to store persistent data.

The PAS internal blobstore is deployed as a single virtual machine in PAS. It does not offer high availability but can be backed up using Pivotal BOSH Backup and Restore (BBR). It is designed for small, non-production deployments.

For PAS production-level deployments, Dell EMC and Pivotal recommend using an external S3-compatible blobstore.
PKS storage

PKS manages multiple Kubernetes clusters, while each cluster storage requirement depends on the size of the cluster and applications. [PKS Documentation](#) has a detailed guidance on PKS storage requirements.

PKS storage options to support stateful apps include:

- vSAN datastores
- NFS or VMFS or FC datastores from external sources

vSAN currently does not support sharing datastores across vSphere clusters. If fail-over across availability zones is desired in the multiple cluster architecture, deploying external shared storage across those availability zones (vSphere clusters) such as VMFS/NFS over iSCSI/FC for both static and dynamic persistent volume provisioning for stateful workloads is supported. Please refer to [PKS Documentation](#) for the implementation details of persistent volume.

Routing

### NSX-T Edge Type and Load Balancer capacity planning

The size of the NSX-T Edge and release of NSX-T determines the number of Load Balancers (varies by LB size) available to be used by the PKS Clusters and thus the number of PKS clusters that can be created.

<table>
<thead>
<tr>
<th>Edge Node</th>
<th>Max # of LB sized small</th>
<th>Max # of LB sized medium</th>
<th>Max # of LB sized large</th>
</tr>
</thead>
<tbody>
<tr>
<td>NSX-T release</td>
<td>2.2</td>
<td>2.3</td>
<td>2.2</td>
</tr>
<tr>
<td>Edge VM - Small</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>Edge VM - Medium</td>
<td>1</td>
<td>1</td>
<td>-</td>
</tr>
<tr>
<td>Edge VM - Large</td>
<td>40</td>
<td>40</td>
<td>1</td>
</tr>
<tr>
<td>Edge Bare Metal</td>
<td>750</td>
<td>750</td>
<td>75</td>
</tr>
</tbody>
</table>

It is recommended to go with a minimum of 4 Large Edge VMs (2 instances is very minimal and necessary for HA) to have a sufficient number of available Load Balancer instances. A production grade deployment should use medium load balancers. Even though there might be 40 small sized load balancers available on a large Edge VM, only the 4 medium sized load balancers are recommended for a production environment.

The size of the Load Balancer determines the number of Virtual Servers, Pools, and Pool Members per LB instance.
Note: Large LB is only available in Bare Metal Edge.

<table>
<thead>
<tr>
<th>LB Service</th>
<th>Small</th>
<th>Medium</th>
<th>Large (only with Bare Metal Edge)</th>
</tr>
</thead>
<tbody>
<tr>
<td>NSX-T release</td>
<td>2.2</td>
<td>2.3</td>
<td>2.2</td>
</tr>
<tr>
<td># of Virtual Servers per LB</td>
<td>10</td>
<td>10</td>
<td>1000</td>
</tr>
<tr>
<td># of Pools per LB</td>
<td>10</td>
<td>20</td>
<td>1000</td>
</tr>
<tr>
<td># of Pool Members per LB</td>
<td>30</td>
<td>200</td>
<td>3000</td>
</tr>
</tbody>
</table>

Important Note: Since PAS and PKS use active/standby Edge Node LB Service, one would always need to consider capacity as per Edge pair to derive the available number of Load Balancer Services. If there are 2 Edge servers, only the load balancer service capacity of one instance can be taken into consideration as the other is treated as stand-by.

The following table shows the capacity for 2 instances of large Edge VM as 4 for medium sized load balancers even though each Edge instance itself does have 4 load balance instance capacity.

<table>
<thead>
<tr>
<th>Edge Cluster</th>
<th># LB sized small</th>
<th># LB sized medium</th>
</tr>
</thead>
<tbody>
<tr>
<td>NSX-T release</td>
<td>2.2</td>
<td>2.3</td>
</tr>
<tr>
<td>2x Edge VM - Large</td>
<td>40</td>
<td>40</td>
</tr>
<tr>
<td>4x Edge VM - Large</td>
<td>80</td>
<td>80</td>
</tr>
<tr>
<td>8x Edge VM - Large</td>
<td>160</td>
<td>160</td>
</tr>
<tr>
<td>2x Edge Bare Metal</td>
<td>750</td>
<td>750</td>
</tr>
<tr>
<td>4x Edge Bare Metal</td>
<td>1500</td>
<td>1500</td>
</tr>
<tr>
<td>8x Edge Bare Metal</td>
<td>3000</td>
<td>3000</td>
</tr>
</tbody>
</table>

The number of available Load Balancer instances tied to the Edge instances is directly proportional (1:1) to the maximum number of Kubernetes clusters that can be supported. Each Kubernetes cluster would use minimum of one load balancer from an active Edge instance. Based on the type of load balancer and the Edge, the load balancer instances are fixed. As such, the resulting number of Kubernetes clusters created on the edge are also constrained by the number of free loadbalancer services on the active Edge.

Note: Maximum number of Edge nodes per Edge Cluster is 10 (as of NSX-T 2.3). If the number of Edge nodes is greater than 10, create additional Edge clusters. If the Load Balancer service capacity is fully utilized on a given Edge pair (one active, other standby), then install and bring up additional Edge VM instance pairs in the same Edge Cluster to handle requirements for additional load balancers (for existing or new PKS clusters).
Ingress Routing and Load Balancing for PKS

How you select ingress routing influences load balancing choices. You’re going to want both ingress routing and load balancing.

1. Ingress Routing - Layer 7
2. Service Type: LoadBalancer - Layer 4

Ingress Routing Layer 7
NSX-T native ingress router is included when deploying with NSX-T. Third party options include Istio or Nginx running as containers in the cluster. Wildcard DNS entries are needed to point at the ingress service in the style of go-routers in PAS. Domain info for ingress is defined in the manifest of the Kubernetes deployment. Here is an example.

apiVersion: extensions/v1beta1
class: Ingress
metadata:
  name: music-ingress
  namespace: music1
spec:
rules:
- host: music1.pks.domain.com
  http:
Service Type: Load Balancer - Layer 4

When pushing a Kubernetes deployment with type set to `LoadBalancer`, NSX-T automatically creates a new VIP for the deployment on the existing load balancer for that namespace. You will need to specify a listening and translation port in the service, along with a name for tagging. You will also specify a protocol to use. Here is an example.

```
apiVersion: v1
kind: Service
metadata:
  ...
spec:
  type: LoadBalancer
  ports:
  - port: 80
    targetPort: 8080
    protocol: TCP
    name: web
```

Summary And Conclusions

A fully meshed PKS and PAS installation based on best practices and reference design considerations will look as follows:
Common elements are the NSX T0 router and the associated T1 routers. This approach allows for any possible cross traffic between PKS and PAS apps to stay within the bounds of the T0 router and not exit the NSX overlay. This also provides a convenient, singular access point to the whole installation, making deployments of multiple, identical installations easier to automate.

AZs are aligned to vSphere clusters. PKS and PAS share the same Infrastructure and Services VLANs and are controlled by common components from the infrastructure plane.

Further Considerations

It continues to be a good design choice to use Resource Pools in vSphere Clusters as AZ constructs to stack different installations of PCF. As server capacity continues to increase, the efficiency of deploying independent clusters of a server just for one install of one product is low. As customers are commonly deploying servers approaching 1T of RAM, stacking many PCF installations in these clusters improves overall resource utilization.

Pay close attention to the maximum configuration maximums allowed per NSX-T installation. These targets change quickly as the product evolves, but you may find that a NSX-T installation per deployment potentially consumes all of the capacity of a NSX install (not a vSphere install). It's reasonable to consider a NSX-T installation per deployment to allow maximum capacity growth.

You may be tempted to split the PAS and PKS installations into separate network parent/child configurations, behind separate T0 routers. Before doing this, review VMware’s best practices for T0 to T0 routing efficiencies and weaknesses to ensure that approach meets your needs.

Please review the reference guidance from VMware for this solution, which is much more detailed in the IaaS constructs than shown here.

https://communities.vmware.com/docs/DOC-38609