WSO2 White Paper

Cloud Native Advantage: Multi-Tenant, Shared Container PaaS

WSO2

lean • enterprise • middleware

Version 1.1 (June 19, 2012)

Table of Contents

Our clients are considering running middleware as a service instead of deploying traditional middleware silos. For example, running ESB-as-a-Service across multiple tenants. The clients are interested in potential cost savings and vendor alternatives. Multi-tenant shared container PaaS will deliver a significant advantage when compared with single tenant, dedicated container PaaS.

A tenant is an isolated or personalized run-time environment context that cannot be shared across PaaS consumers. Tenant specific personalization can occur across multiple dimensions:

- Information access privileges
- Information aggregation and composition
- Business processes and rules
- Service levels and Quality of Service
- Security policies, subscriber entitlements, and social network access privileges
- Monetization rates

Personalization may require loading tenant specific code, configuration files, or data. Tenant specific resources must be isolated and not shared across tenants. Security managers, code deployers, and tenant-aware load balancing components built into a Platform as a Service (PaaS) environment will influence whether tenants require dedicated containers to achieve expected context separation and resource isolation. Not all PaaS platforms contain these components at the application platform level, and their ability to isolate tenants while sharing resources will vary based on container design and tenancy encapsulation level (e.g. server hardware, virtual machine, application platform container, application). When multi-tenancy is supported within the application container (i.e. Java Virtual Machine [JVM], .NET Common Language Runtime [CLR]), the environment optimizes resource sharing and delivers a more cost efficient PaaS.

A PaaS container is a standalone, Internet addressable node offering application platform services (e.g. Web application hosting, API management, integration endpoint hosting, ESB mediation, registry services, identity management, and relational database). Containers host tenant resources and context. Containers may serve a single tenant during a specific timeframe (dedicated), or serve multiple, concurrent tenants (shared).

PaaS Container Partitioning Strategies

A PaaS controller component defines partitions and tunes container sharing, service resource allocation, QoS, and utilization. Containers may be assigned into service-specific or tenant specific partitions. When a PaaS host cannot simultaneously host multiple tenants, the PaaS controller partitions distinct and separate container resource pools per tenant. Resources allocated to a single, flat namespace resource partition may be shared more widely than resources divided across multiple partitions. Figure 1 and Figure 2 illustrate the different partitioning schemes.

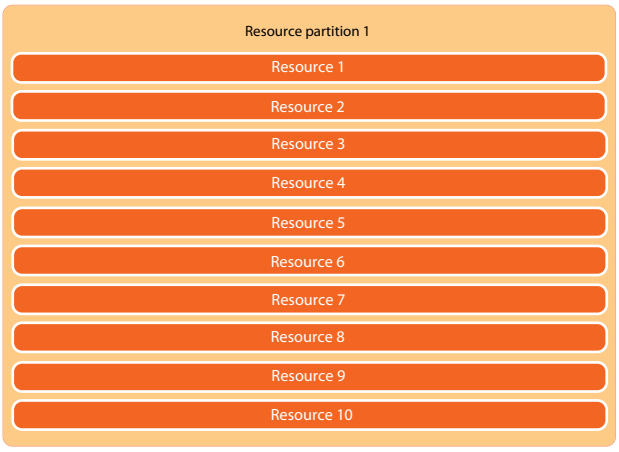

Figure 1: Resources Partitioned into a Single, Flat Namespace

Using a single, flat namespace-partitioning scheme, all tenant traffic may be serviced by any available resource. Associating the available resource pool across the largest number of tenants maximizes resource sharing.

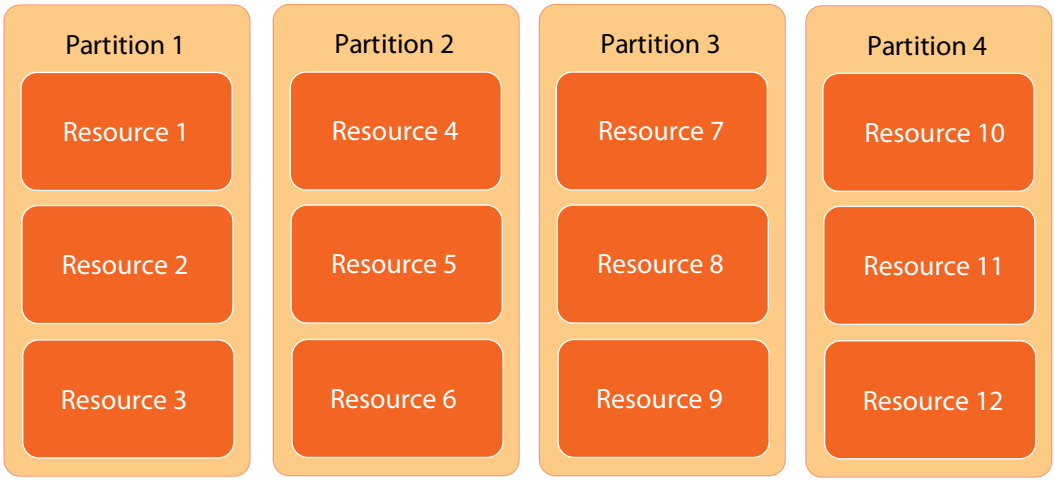

Figure 2: Resources Partitioned across Multiple Namespaces

Using the multiple namespace resource-partitioning scheme, tenant traffic is serviced by specific tenant resource partitions. For example, Partition 1 will service tenants with organization name starting in A-F, and Partition 2 will service tenants with organization name starting G-M. With multiple namespace partitioning, the environment resources may only service a subset of the total tenant traffic. When the environment uses a multiple namespace partitioning scheme, there is a higher probability that resources will be unevenly utilized. A wide body of queuing theory research demonstrates sub-optimal performance when traffic is fragmented across multiple namespace-partitioned environments.

Container Tenancy

Application PaaS (aPaaS) containers host application resources (i.e. database connection pools, resource bundles, mediations, and session resources). Based on architecture capabilities (i.e. security management, code deployment, administration, and tenant-aware load balancing), a container may support servicing multiple concurrent tenants without application design modifications, or be restricted to supporting a single tenant until application design is refactored.

When performing multi-tenant application design, an architect will often specify a factory pattern to deliver appropriate resources to worker processes or threads. For example, a factory binds a database connection or resource to a tenant context, the application code stores the context in a tenant session, and the tenant-specific resources are accessed during subsequent tenant workload execution. Software-as-a-Service (SaaS) applications often employ the factory pattern to bind tenant specific resources to tenant context and tenant worker threads. Tenant-aware aPaaS containers deliver application resources based on tenant context, and eliminate development tasks required to manual code tenant specific personalization or tenant specific isolation boundaries (e.g. a database connection to a specific database instance).

PaaS offerings based on non-tenant aware Java Virtual Machines require isolating JVM instances into tenant-specific partitions. The non-tenant aware PaaS must dedicate a JVM pool per tenant, and not

share JVM instances across multiple concurrent tenants. Most PaaS offerings (e.g. RedHat OpenShift, VMWare CloudFoundry, Jelastic PaaS, or Amazon BeanStalk) require a dedicated container pool per tenant organization or tenant application. For example, a tenant application hosted on Jelastic PaaS will be bound to a dedicated number of Tomcat servers. As a pre-requisite to hosting an application, the developer will provision a specific number of application platform instances and a well-defined topology (e.g. 4 Tomcat server instances, 1 mySQL database instance). While the number of application servers may grow or shrink based on user demand, multiple tenants will not simultaneously co-exist on application server instances. Figure 3 and Figure 4 demonstrate tenant-container topology and tenant resource partitions based on application platform architecture constraints.

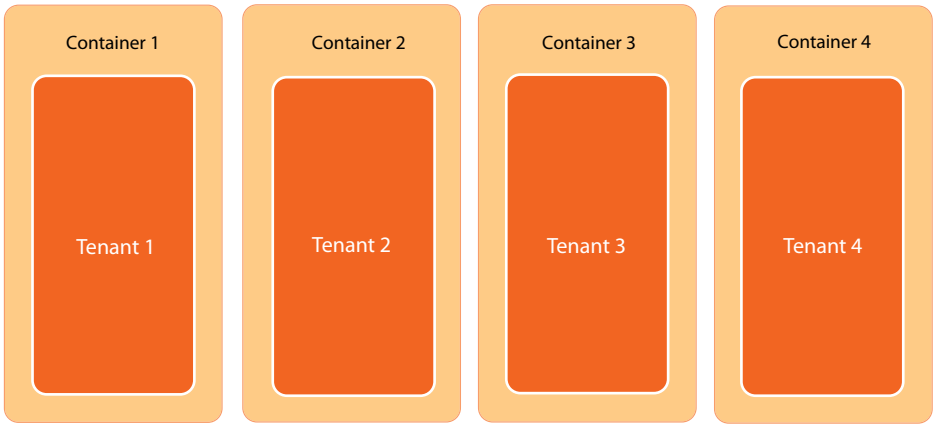

Figure 3: Single Tenant/Application Per Container Association

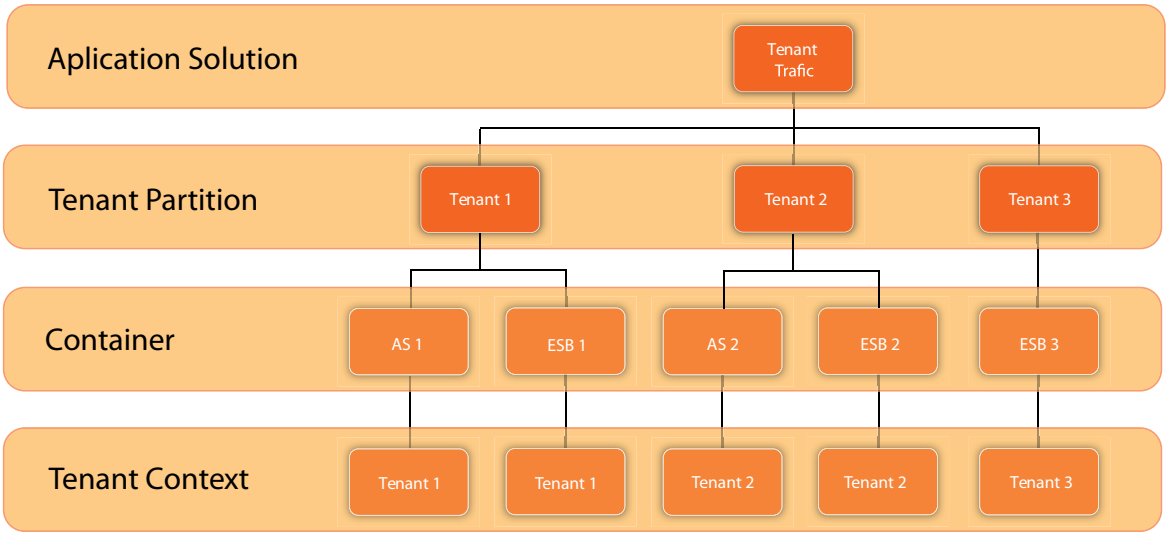

Figure 4: Single Tenant, Dedicated Container Topology Example

In the example use case, tenant 1 and tenant 2 require web application hosting and ESB services. Tenant 3 requires only ESB services. In the Figure 4 illustration, application server (AS) instances and Enterprise Service Bus (ESB) instances are dedicated per tenant. A three-tenant solution on the single, dedicated container PaaS requires deploying five PaaS container instances. The topology is obtained by the following calculation:

- One partition per tenant
- At least 1 platform container instance per tenant-specific application service
- An IaaS node per (N) platform container instances

Traditional application servers can host multiple applications, but resource isolation requirements often restrict sharing across multiple tenants. Common resource isolation requirements apply to security, application resource loading/pooling, administration, monitoring, management, code deployment, and load balancing. PaaS offerings based on traditional application servers often enforce multi-tenancy at the virtual machine level. Each application server instance is hosted within a dedicated virtual machine, and the entire application environment stack runs within a dedicated partition (i.e. dedicated DNS hostname, dedicated virtual machine, dedicated application server license, dedicated application server instance, and dedicated application code execution space).

Multi-tenant Shared Container

Single tenant, dedicated server container tenancy (e.g. Jelastic PaaS, Amazon Beanstalk, RedHat OpenShift, CloudBees) contrasts with shared container tenancy (e.g. WSO2 Stratos, Google AppEngine). In shared container tenancy, tenants simultaneously share a common pool of run-time language containers (i.e. JVM, CLR). The run-time language container instance is dynamically shared across multiple tenant applications. Multi-tenant application platform service instances can host multiple tenants with appropriate resource isolation. The container is purpose-built to overcome context isolation challenges, deployment tooling issues, and resource sharing impediments. Multi-tenant shared container platforms use OSGI class loading, custom security managers, Cloud aware code deployer, tenant-aware administration screens, and tenant-aware load balancing to create a safe, multi-tenant environment while maximizing resource sharing. Within a multi-tenant platform, a platform container (aka application server) can host multiple concurrent tenants (~7-100). Figure 5 and Figure 6 visually depict the container-tenant association and expected deployment topology.

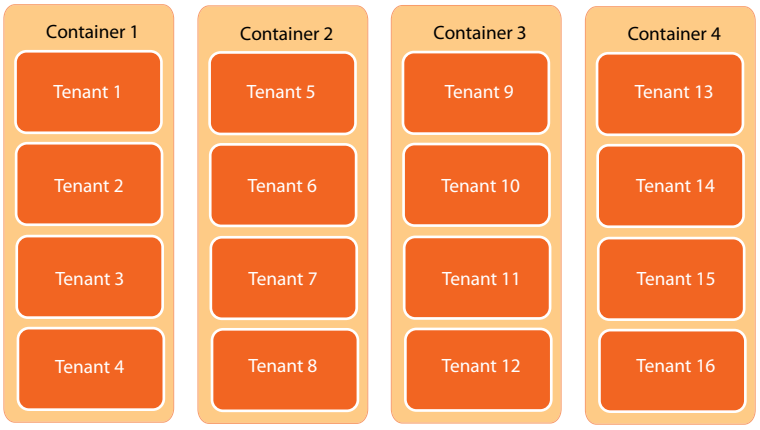

Figure 5: Multiple Tenants/Application Per Container Association

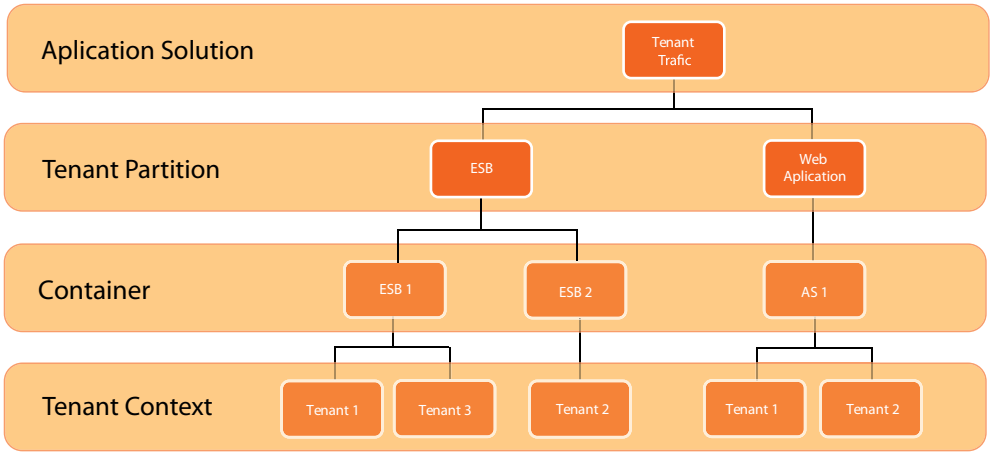

Figure 6: Multi-Tenant, Shared Container Topology Example

In the example use case, tenant 1 and tenant 2 require web application hosting and ESB services. Tenant 3 requires only ESB services. In the Figure 6 illustration, application server (AS) instances and Enterprise Service Bus (ESB) instances are shared across tenants. A three-tenant solution on the shared container PaaS requires deploying three PaaS container instances. The topology is obtained by the following calculation:

- One partition per application platform service (i.e. ESB, web application)
- At least 1 platform container instance per application platform server
- Two platform container instances to handle high ESB load
- Service containers shared across multiple tenant context
- An IaaS node per (N) platform container instances

Multi-Tenant Shared Container PaaS TCO Benefits

In single tenant, dedicated container PaaS, significantly more expense is required to run a PaaS environment compared with a multi-tenant, shared application container PaaS. Main technical attributes influencing total cost of ownership include:

- Container sharing and tenant isolation level
- Tenant Density per JVM or application platform server
- Container license cost

The proposed PaaS cost evaluation tool compares multi-tenant, shared application container PaaS with single tenant, dedicated container PaaS (i.e. traditional application server deployment in Cloud) across multiple tenant counts and application platform service combinations. The worksheet incorporates application platform license (or subscription) cost, PaaS Management service cost, infrastructure expense, and IT management overhead. Across all scenarios, the worksheet calculates cost when application platforms are deployed on Infrastructure as a Service (IaaS).

Since both a shared application container PaaS and single tenant PaaS can scale up/down on-demand, the worksheet excludes savings derived from on-demand provisioning and teardown. Instead, the worksheet calculates expense based on the maximum number of JVM instances required for steady-state workloads.

The worksheet does not depict intangible savings derived from faster time to market. Since the scenarios assume traditional web application architecture, container-level isolation via OSGI, and application clustering (instead of stateless message passing), application migration cost and development cost will be equal and are excluded from the calculation.

Cost Calculation Methodology

The worksheet reflects the following cost components:

- Application Platform Subscription (or license fee)
- PaaS Management Services (i.e. controllers, load balancer, metering, billing, and monitoring)
- Infrastructure as a Service (compute only)
- IT Administrative and Management

The Application Platform subscription component calculates the license, vendor maintenance, or subscription expense required to license access to application platform components. The private Cloud scenarios listed in the worksheet utilize a straight yearly subscription fee per JVM instance.

PaaS Management Services load balance tenants across application platform nodes, monitor PaaS node health, collect metering information, and provision tenant environments. As many 'Private PaaS' deployments replicate traditional application platform topology within the run-time environment, the worksheet removes PaaS Management Service cost from the traditional deployment template scenarios. If your single tenant, dedicated deployment requires run-time management services (e.g. Cloud Foundry), adjust the worksheet accordingly. The worksheet may understate the cost required to deploy Cloud Foundry and other single dedicated PaaS delivering on-demand, tenant-aware elastic load balancing.

Across all scenarios, the worksheet calculates cost when application platforms are deployed on Infrastructure as a Service (IaaS). The worksheet IaaS compute benchmark cost from Amazon AWS EC2 calculator (as of 11 May 2012). The projected spend is $659 per month for a High-Memory Double Extra Large Instance.
The IT Administrative and Management yearly cost item incorporates administration effort required to generate monthly chargeback/showback statements, configure tenants, monitor run-time environment per tenant, and up-front training investment.

Column B reflects the multi-tenant, shared container scenario. Column C reflects a single tenant deployment, which is used as input to traditional deployment scenarios (Column D-G). The traditional deployment scenarios replicate single, dedicated tenant silos in a PaaS environment. A dedicated tenant silo shares IaaS across JVMs, and dedicates running JVM instances to specific tenants.

Calculation Variables

The following variables may be changed to evaluate cost within your specific context:

- Number of tenants
- Tenant density per JVM [applied per Application Platform Service]
- License instances (i.e. Java Virtual Machine instances or servers) per Application Platform Service and PaaS Management Service
- Java Virtual Machine (JVM) density per IaaS Node
- Cost per IaaS Node

IaaS compute instance size (e.g. small, medium, large) and application platform footprint (i.e. 256MB, 2GB, 4GB) will influence JVM density per IaaS node. The worksheet assumes a 32GB IaaS compute instance node with 4GB allocated to each Java Virtual Machine. The worksheet projects an expected JVM density of 8 JVMs per IaaS node. The worksheet user may adjust the number of JVMs per IaaS node.

Tenant isolation, partitioning strategies, and service performance will influence tenant density per JVM. The worksheet's projected Application Platform Service JVM count considers redundancy, failover, expected load, throughput, and service performance. The JVM count will typically vary based on hosted Application Platform Service type, throughput, and expected load. For example, an Identity Management service can often accommodate

a greater number of tenants compared with an Enterprise Service bus service. The tenant density per JVM will vary based on vendor product and workload profile. In the single tenant scenarios, a single tenant is mapped per dedicated JVM or JVM cluster. In the multi-tenant shared container scenario, a JVM instance uses OSGI partitioning, tenant specific resource context, and on-demand tenant artifact loading to share JVM resources while maintaining Quality of Service, performance, and security. In a multi-tenant shared JVM scenario, the worksheet specifies a maximum of seven (7) tenants per JVM in high load situations (e.g. ESB, Application Server, Business Process Management).

Tenant density significantly impacts the number of IaaS nodes and licensed application platform servers. In the default worksheet configuration, a single tenant per JVM yields a single tenant density of eight (8) tenants per IaaS node. In a multi-tenant shared JVM scenario, the worksheet specifies a maximum of seven (7) tenants per JVM and 42 tenants per IaaS node.

The worksheet uses a straight subscription support fee per JVM instance and IaaS compute benchmark cost from Amazon AWS EC2 calculator (as of 11 May 2012). Worksheet users should adjust this number to reflect vendor specific license expense, maintenance expense, subscription investment amounts, and internal cost to deliver IaaS compute nodes. As storage, network I/O, and external IP addresses will remain constant across all scenarios, the worksheet excludes these costs.

Modifying Cost Scenarios

To modify the cost scenarios and accurately model your expected cost savings from multi-tenant, shared container PaaS, you should collect the following load information:

- A specified number of tenant projects in Year 1,2,3
- Known tenant load (e.g. web requests, transactions, memory, CPU, latency)
- Known mapping between tenant load and service capacity

The information can be used to derive the following variable values:

- Expected number of tenants
- Number of Application Platform Service JVMs
- JVM density per IaaS node

Modify the worksheet to reflect your specific environment context. Cost variables are shown in ‘blue font’.

Bottom Line

In use case #1, small-scale, single application service deployments (i.e. 8 tenants and 5 services), multi-tenant shared container PaaS is 2.5 times (2.5X, 250%) more efficient than single container deployment when measuring number of JVM instances. Positive financial TCO is achieved after four (4) tenants subscribe to the environment. After all 8 tenants have subscribed, the multi-tenant shared container PaaS is 200% more cost efficient than single container deployments. Table 1 and Table 2 illustrate the details.

In use case #2, small-scale, single application service deployments (i.e. 8 tenants and 1 service), multi-tenant shared container PaaS is two times (1.7x, 170%) more efficient than single container deployment when measuring number of JVM instances. Positive financial TCO is achieved after five (5) tenants subscribe to the environment. After all eight (8) tenants have subscribed, the multi-tenant shared container PaaS is 144% more cost efficient than single container deployments. Table 3 and Table 4 illustrate the details.

In use case #3, large-scale, single application service deployments (i.e. 100 tenants and 3 services), multi-tenant shared container PaaS is 3.8 times (3.8X, 380%) more efficient than single container deployment when measuring number of JVM instances. Positive financial TCO is achieved after twenty-two (22) tenants subscribe to the environment. After all 100 tenants have subscribed, the multi-tenant shared container PaaS is 278% more cost efficient than single container deployments. Table 5 and Table 6 illustrate the details.

Use Case 1: Complex solution, Low Tenant Count

On-Premise Shared PaaS vs. On-Premise Single Tenant Hosting	Shared Container PaaS Scenario	Traditional Deployment Template	Traditional - Scenario 1	Traditional - Scenario 2	Traditional - Scenario 3	
Tenants and Partitioning						
Tenant Isolation Level	Shared PaaS Nodes	Dedicated Instances	Dedicated Instances	Dedicated Instances	Dedicated Instances	Dedicated Instances
Tenants	8	1	2	4	6	8
PaaS Management Services [# of JVMs]	8	0	0	0	0	0
Application Platform Services [# of JVMs]						
Application Server	2	2	4	8	12	16
Enterprise Service Bus	2	2	4	8	12	16
Governance Registry	2	1	2	4	6	8
Identity Server	1	1	2	4	6	8
Data Services Server	1	1	2	4	6	8
Load Balancer for Application Platform Service Clusters	2	0	0	0	0	0

Table 1: On-premise Shared PaaS Instance Count for Complex Solution, Low Tenant Count

Tenants and Partitioning	Shared Container PaaS Scenario	Traditional Deployment Template	Traditional - Scenario 1	Traditional - Scenario 2	Traditional - Scenario 3	
Tenant Isolation Level	Shared PaaS Nodes	Dedicated Instances	Dedicated Instances	Dedicated Instances	Dedicated Instances	Dedicated Instances
Tenants	8	1	2	4	6	8
Platform Footprint and Cost						
Number of JVM Instances	26	8	16	32	48	64
Subscription cost per JVM Instance [per year]	$8,000	$8,000	$8,000	$8,000	$8,000	$8,000
Total Platform Subscription Cost [per year]	$208,000	$64,000	$128,000	$256,000	$384,000	$512,000
Infrastructure as a Service						
JVM Density per IaaS node (32GB IaaS, 4GB JVM)	8	8	8	8	8	8
Calculated number of IaaS nodes	4	1	2	4	6	8
Cost per IaaS node [per year]	$7,908	$7,908	$7,908	$7,908	$7,908	$7,908
Total IaaS Cost [per year]	$31,632	$7,908	$15,816	$31,632	$47,448	$63,264
IT Administrative and Management Cost						
Total IT Administrative and Management Cost [per year]	$178,120	$32,400	$64,800	$129,600	$194,400	$259,200
Total Cost of Ownership						
Total Cost [Year 1]	$417,752	$104,308	$208,616	$417,232	$625,848	$834,464
Total Savings for PaaS / (Extra Expense)	$0	($313,444)	($209,136)	($520)	$208,096	$416,712
JVM Instance Efficiency		0.3	0.6	1.2	1.8	2.5
Cost Efficiency		25%	50%	100%	150%	200%

Table 2: On-premise Shared PaaS Cost for Complex Solution, Low Tenant Count

Use Case 2: ESB-as-a-Service, Low Tenant Count

On-Premise PaaS vs. On-Premise Hosting	Shared Container PaaS Scenario	Traditional Deployment Template	Traditional - Scenario 1	Traditional - Scenario 2		
Tenants and Partitioning						
Tenant Isolation Level	Shared PaaS Nodes	Dedicated Instances	Dedicated Instances	Dedicated Instances		Dedicated Instances
Tenants	8	1	2	4	5	8
PaaS Management Services [# of JVMs]	7	0	0	0	0	0
Application Platform Services [# of JVMs]						
Application Server	0	0	0	0	0	0
Gadget Server	0	0	0	0	0	0
Mashup Server	0	0	0	0	0	0
Enterprise Service Bus	2	2	4	8	10	16
Governance Registry	1	1	2	4	5	8
Load Balancer for Application Platform Service Clusters	1	0	0	0	0	0

Table 3: On-premise Shared PaaS Instance Count for ESB-as-a-Service, Low Tenant Count

Tenants and Partitioning	Shared Container PaaS Scenario	Traditional Deployment Template	Traditional - Scenario 1	Traditional - Scenario 2		
Tenant Isolation Level	Shared PaaS Nodes	Dedicated Instances	Dedicated Instances	Dedicated Instances		Dedicated Instances
Tenants	8	1	2	4	6	8
Platform Footprint and Cost						
Number of JVM Instances	19	4	8	16	21	32
Subscription cost per JVM Instance [per year]	$8,000	$8,000	$8,000	$8,000	$8,000	$8,000
Total Platform Subscription Cost [per year]	$152,000	$32,000	$64,000	$128,000		$256,000
Infrastructure as a Service						
JVM Density per IaaS node (32GB IaaS, 4GB JVM)	8	8	8	8	8	8
Calculated number of IaaS nodes	3	1	1	2	3	4
Cost per IaaS node [per year]	$7,908	$7,908	$7,908	$7,908	$7,908	$7,908
Total IaaS Cost [per year]	$23,724	$7,908	$7,908	$15,816	$23,724	$31,632
IT Administrative and Management Cost						
Total IT Administrative and Management Cost [per year]	$124,000	$18,960	$36,120	$72,240	$94,560	$144,480
Platform Subscription Cost Comparison (IT Admin & Mgt)	$0	($105,040)	($87,880)	($51,760)		$20,480
Total Cost of Ownership						
Total Cost [Year 1]	$299,724	$58,868	$108,028	$216,056		$432,112
Total Savings for PaaS / (Extra Expense)	$0	($240,856)	($191,696)	($83,668)		$132,388
JVM Instance Efficiency		0.2	0.4	0.8	1.1	1.7
Cost Efficiency		20%	36%	72%	96%	144%

Table 4: On-premise Shared PaaS Instance Count for ESB-as-a-Service, Low Tenant Count

Use Case 3: ESB-as-a-Service, High Tenant Count

On-Premise PaaS vs. On-Premise Hosting	Shared Container PaaS Scenario	Traditional Deployment Template	Traditional Deployment	Traditional Deployment	Traditional Deployment	Traditional Deployment
Tenants and Partitioning						
Tenant Isolation Level	Shared PaaS Nodes	Dedicated Instances	Dedicated Instances	Dedicated Instances	Dedicated Instances	Dedicated Instances
Tenants	100	1	10	22	50	100
PaaS Management Services [# of JVMs]	10	0	0	0	0	0
Application Platform Services [# of JVMs]						
Application Server	0	0	0	0	0	0
Gadget Server	0	0	0	0	0	0
Mashup Server	0	0	0	0	0	0
Enterprise Service Bus	14	2	20	44	100	200
Governance Registry	2	1	10	22	50	100
Identity Server	1	1	10	22	50	100
Load Balancer for Service Clusters	3	0	0	0	0	0

Table 5: On-premise Shared PaaS Instance Count for ESB-as-a-Service, High Tenant Count

Tenants and Partitioning	Shared Container PaaS Scenario	Traditional Deployment Template	Traditional - Scenario 1	Traditional - Scenario 2	Traditional - Scenario 3	Traditional - Scenario 4
Tenant Isolation Level	Shared PaaS Nodes	Dedicated Instances	Dedicated Instances	Dedicated Instances	Dedicated Instances	Dedicated Instances
Tenants	100	1	10	22	50	100
Platform Footprint and Cost						
Number of JVM Instances	130	5	50	110	250	500
Cost Per JVM Instance	$8,000	$8,000	$8,000	$8,000	$8,000	$8,000
Total Platform Subscription Cost	$1,040,000	$40,000	$400,000	$880,000	$2,000,000	$4,000,000
Infrastructure as a Service						
JVM Density per IaaS node (32GB IaaS, 4GB JVM)	8	8	8	8	8	8
Calculated number of IaaS nodes	17	1	7	14	32	63
Cost per IaaS Year	$7,908	$7,908	$7,908	$7,908	$7,908	$7,908
Total IaaS Cost	$134,436	$7,908	$55,356	$110,712	$253,056	$498,204
IT Administrative and Management Cost						
Total IT Administrative and Management Cost	$1,224,400	$22,320	$217,800	$476,640	$1,083,600	$2,165,400
Total Cost of Ownership						
Total Cost	$2,398,836	$70,228	$673,156	$1,467,352	$3,336,656	$6,663,604
Total Savings / (Extra Expense)	$0		($1,725,680)	($931,484)	$937,820	$4,264,768
JVM Instance Efficiency		0.0	0.4	0.8	1.9	3.8
Cost Efficiency		3%	28%	61%	139%	278%

Table 6: On-premise Shared PaaS Instance Count for ESB-as-a-Service, High Tenant Count

About WSO2

WSO2 is the lean enterprise middleware company. It delivers the only complete open source enterprise SOA middleware stack purpose-built as an integrated platform to support today's heterogeneous enterprise environments—internally and in the cloud. WSO2's service and support team is led by technical experts who have proven success in deploying enterprise SOAs and contribute to the technology standards that enable them.

Check out more **WSO2 Whitepapers** and **WSO2 Case Studies**.