CBTS Cloud Technical Buyer's Guide: Red Hat OpenShift

White paper

cbts

Consult Build Transform Support

CBTS Cloud Technical Buyers Guide: Red Hat OpenShift

The new reality is that achieving exponential growth is no longer about the big beating the small, it is about the fast beating the slow.

Speed wins in the modern application cloud era. It trumps efficiency. No one will dominate their industry by being the best at running a physical data center any longer. You will dominate over your competition by getting to market faster with a new product capability, getting unique insights from your data faster than anyone else does, and especially by changing direction and trying new things faster.

Modern application development is the rapid creation, maintenance, and management of capabilities that can run across complex hybrid cloud environments that are fully automated and scale on demand.

This approach lets organizations get the most from innovative technologies like containers and micro-services and game-changing practices like agile, DevOps, and continuous integration and deployment (CI/CD).

- Cloud Native describes the patterns of high performing organizations delivering software faster, consistently and reliably at scale.
- Continuous delivery, DevOps, and Micro-services label the why, how and what of the application cloud natives.
- In the most advanced state of CI/CD and DevOps, the automation and orchestration of the applications and infrastructure become so intertwined, they reach a point of being inseparable.

Application Modernization and Transformation

At the infrastructure layer, containers enable software to run reliably when moved from one computing environment to another and greatly assist in creating an efficient DevOps environment. However, to truly realize the benefits of next generation delivery we need to look at the application layer and ensure it is built for cloud native operations.

Applications that are built net-new, or have been refactored to be cloud native, have capabilities that are designed for resiliency and scale, and can be deployed on public cloud for a very nimble and efficient delivery model. This can greatly accelerate application deployment through a true DevOps experience leveraging containers. Developers can cut the time it takes to deliver a release or update functionality already in production. This can also include adoption of agile and DevOps processes.

Application Refactoring is Ideal for:

- Businesses with older technology that needs to be upgraded or migrated from on-premises to a public, private, or hybrid cloud solution.
- Companies that have completed a “lift and shift” of legacy apps to the cloud but are not getting the benefits they hoped for.
- Enterprises with a multitude of applications — from dozens to hundreds — that need a multi-year program for migrating each of them to the cloud.
- IT leaders who are tired of continually buying new servers and storage for line-of-business applications.

How it Works:

CBTS balances continuous application development, innovation and improvement with a rigorous, rapid development approach that:

- Aligns your business objectives to a technology application plan and schedule.
- Nurtures constant feedback loops on performance, functionality, and user behavior.
- Develops a transparent road-map to show how applications migrate, when services go live, and which features become available.
- Delivers reliable code at a rapid pace, resulting from our well-defined agile-development process.
- Supports monitoring and managing apps in the cloud on any platform, derived from many years of application support experience.

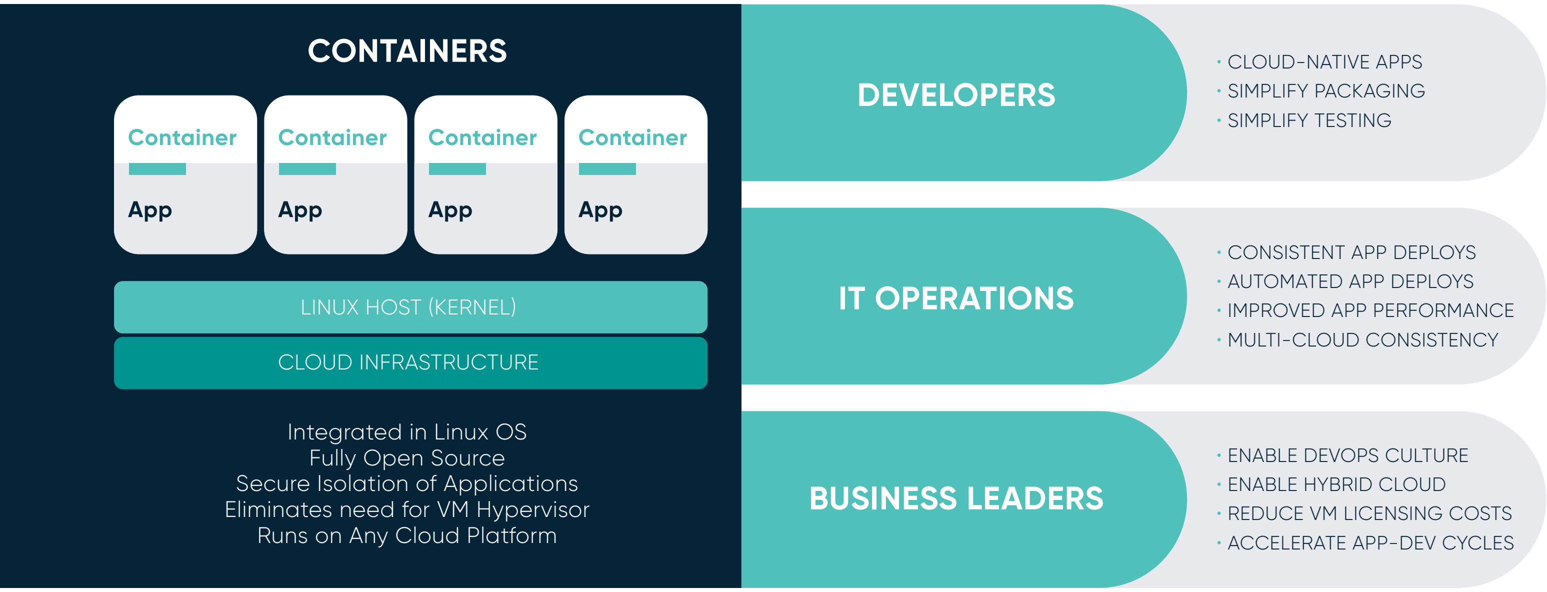
Red Hat OpenShift Container Environment

Containers enable software to run reliably when moved from one computing environment to another and greatly assist in creating an efficient DevOps environment.

Containers package an application, plus all of its dependencies (libraries, configuration files) bundled into one portable image.

What are Containers?

Consistent Tools for Both Developers and IT Operations



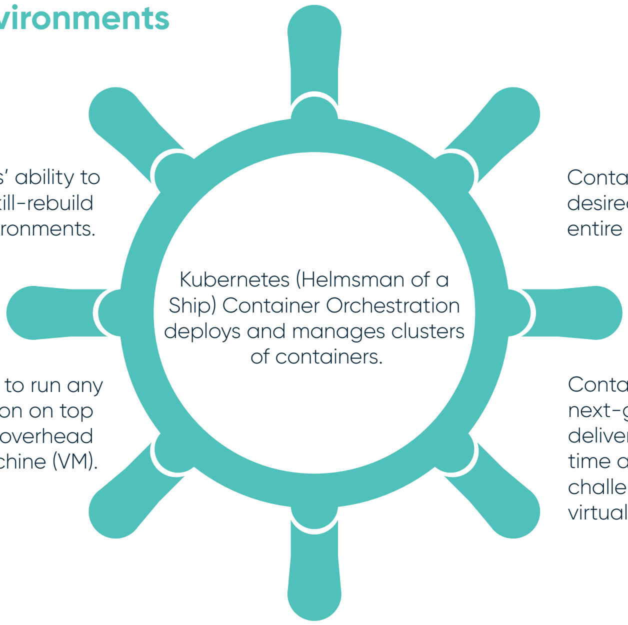
Why Containerized Environments

Enabling DevOps

.

Greatly improves developers' ability to deploy and configure/kill-rebuild production application environments.

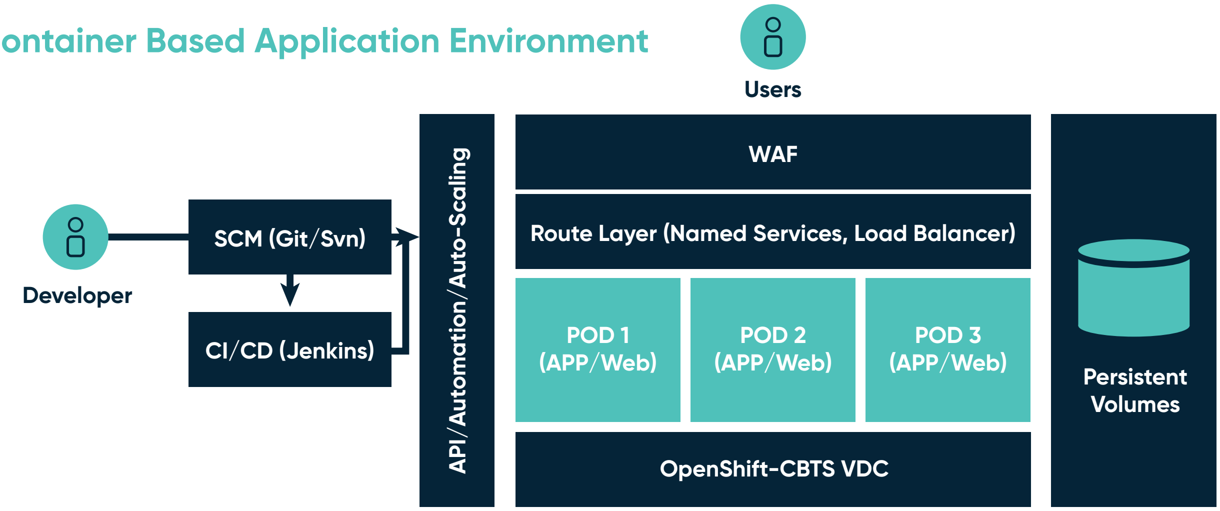
Containerization allows you to scale only the desired functions without impacting the entire application.


Kubernetes (Helmsman of a Ship) Container Orchestration deploys and manages clusters of containers.

Containerization also allows you to run any platform with its own configuration on top of your infrastructure without the overhead of a virtual machine (VM).

Containers allow developers to achieve next-generation efficiency in software delivery or allow product managers to save time and resources by settling many of the challenges that they face with traditional virtualization.

Container Based Application Environment



Speed/Operational Excellence

- Automation - CI/CD pipeline
- Automation - (terraform)
- Config Management
- Network Automation
- Access to Standard Container Registries

Secure

- Aggregated Logging
- Separation of Duties (DEV/QA vs Production)
- Network Segmentation

Reliability

- Load Balancer
- DB Replication (Cross Region)
- Backup (Static Data Only)
- Backup Container Registry

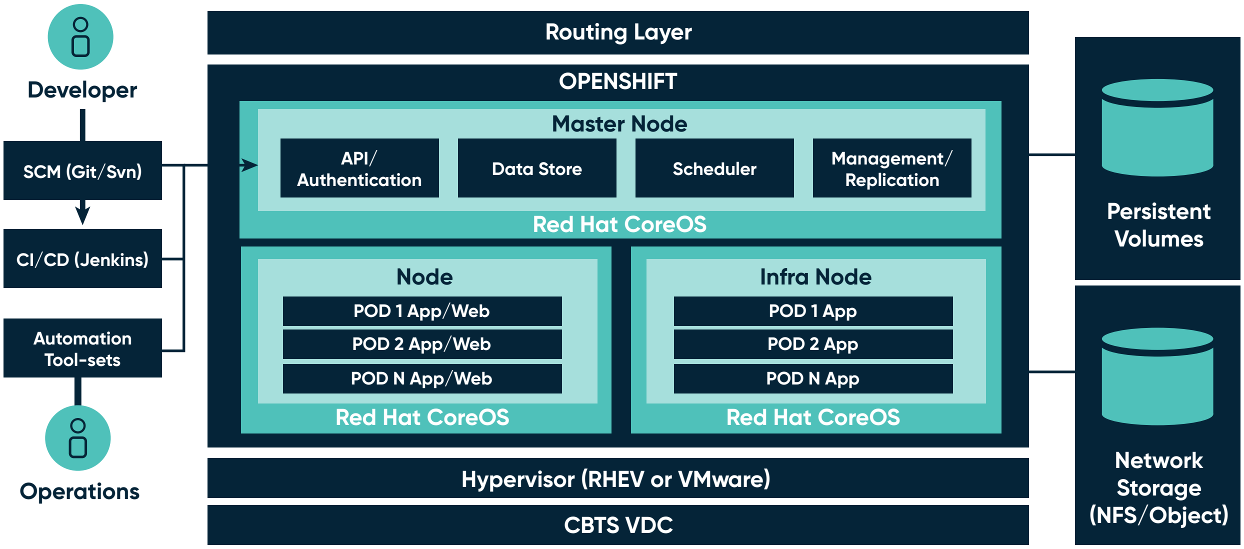
Cost Optimized

- Elasticity (Scale in/out)
- Right Storage for the Right Task
- Only Backup Static Data and Database

Performance

- Monitoring
- APM
- Auto-Scaling

OpenShift Implementation Overview



Multiple OpenShift Clusters

- DEV/QA/Staging (Geographically Separate, enabled for DR)
- Production
- Multi-tenancy managed via OpenShift SDN

Platform: RHEV

- RHEV as the underlying virtual machine tier
- The OpenShift platform nodes run RH CoreOS

Capacity

- vCPU=59, RAM=336GB
- Minimal Growth Unit: 12vcpu, 64GB RAM

Storage

- iSCSI for block/File storage (Persistent Volumes)
- Object Storage - Not in Scope, Not all Pods will require external IPs. Standard Persistent Volumes preferred.
- Persistent Volumes backed up via platform

Automation

- Terraform used to manage containers
- Monitoring: EM7, Metrics Server
- Auto-Scaling
- Networking/Routing Layer (Named Services/HA Proxy/F5)
- Container Registry
- Log Aggregation Services

Multi-Tiered Approach for Ultimate Flexibility

Tier 1 Entry: OpenShift – CBTS public cloud/On-Prem private cloud

- Project Based Multi-tenancy
- Auto-Scaling
- Load Balancer
- EM7 Basic Monitoring
- Terraform for Automation
- Configuration Management
- Data Backup is Additional

Tier 2 Advanced: BASIC offering plus the following capabilities

- Persistent Volume Backups
- Centralized Logging
- Multi-Region
- Application performance monitoring with New Relic

Tier 3 Enterprise: Advanced offering plus the following capabilities

- OpenShift SDN
- WAF available
- API for vulnerability scanning
- Scheduled monthly rebuild for middleware updates

Cloud Automation with Terraform

Create container pods using Terraform to implement Infrastructure as code. This allows your application design to be portable to any container based IaaS and allows you to seamlessly use the lowest cost environments for your needs. Using Terraform also allows you to standardize your automation stack across multiple Cloud Environments. The more portable your environment, the cheaper its total cost of ownership over the long term.

PV Backup using Persistent Volume Snapshots

Each OpenShift project will include a backup pod responsible for taking snapshots of the claimed persistent volumes. The snapshots will be set up for short term retention, but can be configured to be archived for a longer term retention or off-site copy.

Standard Templates Available:

- Load Balancer
- Replicated/Multi-Master databases
- Common Middleware/messaging layers
- Centralized logging
- Auto-Scaling
- Common Application Stacks, including LAMP, Git/Jenkins CI/CD
- Web Application Firewall

Monitoring of pods included:

- EM7

Standard Security tools:

- Centralized Logging
- Configuration Management
- Network Segregation
- Firewall Management
- WAF Management

Seamless Application Patching

Eliminate the struggle to keep up to date with middleware patches forever. Containers will be automatically updated to the latest versions on a weekly basis. Dev/QA will always be updated a week before production. Automatic CI/CD testing will be triggered as part of the upgrades.

API for Vulnerability Scanning

API for vulnerability scanning available for your use in your CI/CD testing chain.

Multi-Tenant

OpenShift SDN can be used to create isolated networks to support a Multi-Tenant environment.

CBTS is your trusted partner to help you get to the next generation of cloud native application and infrastructure delivery so that you can vastly improve your time to market, eliminate downtime, and leverage automation for ease of ongoing management.