

## A highly available and resilient platform for running containers with lower management effort

As Scale Computing sees more and more adoption across distributed enterprise organizations, we are seeing accelerated adoption of containers and the use of API-driven automation such as infrastructure-as-code. Large organizations often have workloads that need to run within containers. Containerization is motivated by many factors: modern applications are often developed to be “container first”, microservices are increasingly being deployed, and hybrid cloud deployments demand portable workloads.

### Seamless, Programmatic Deployment

Scale Computing HyperCore (SC//HyperCore) allows for seamless, programmatic deployment of containers. To run containers on SC//HyperCore, users simply deploy a container-optimized operating system with a container runtime of choice (often Docker or in a Kubernetes environment, containerd or CRI-O). This can be accomplished manually, however, container use cases often require a high degree of automation. Therefore, our REST-APIs and support of cloud-Init radically improve how users can run containerized workloads by automating the installation of the operating system, container runtime and workload containers.

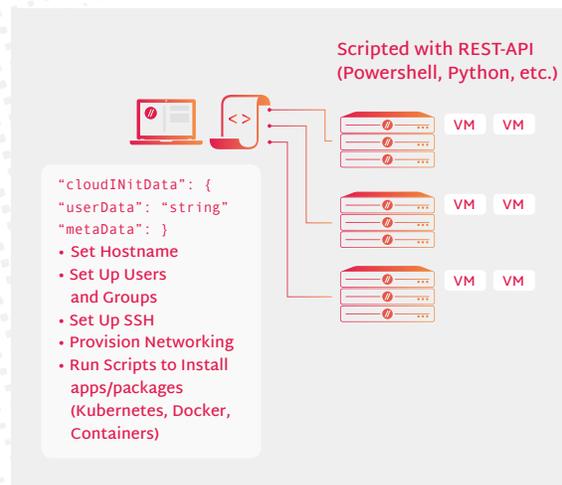
### Cloud-init Customization with REST-APIs

SC//HyperCore includes cloud-init customization via REST-APIs to enable infrastructure as code so developers and administrators can automate otherwise very manual processes.

Cloud-init in SC//HyperCore allows easy, mass provisioning of customized VMs.

#### Benefits:

- Save time automating manual steps in site and application setup
- Reduce human error from manual setup
- Ensure deployment consistency across sites
- Enable consistent change control and more reliable update through standardization

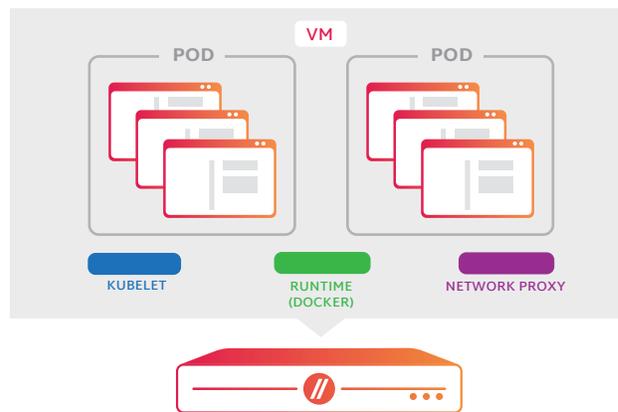


Scale Computing understands that SC//HyperCore must work flawlessly in supporting both VMs and containers. It must be easy to manage and deploy containers programmatically, no matter the implementation. Recent investments in REST-APIs radically improve the speed and ease of mass-container deployment on SC//HyperCore. As such, organizations can save management overhead by writing scripts to handle daily management tasks needed at scale, particularly when they are deploying containers. Perhaps the single most important REST-API endpoint for deploying containers on SC//HyperCore is CloudInitData. Cloud-init is a powerful open-source technology that allows users to automatically configure VMs on their first boot. Specifically, cloud-init is a package available to most Linux distributions that allows users to take a cloud image (essentially a vanilla Linux image equipped with cloud-init) and provide both the VM with meta and user data via a script during the initial boot. The configurations available are quite powerful. Users can set up SSH keys, create users, write files, and even run entire commands automatically.

Run Docker - or any other container runtime - commands right after VM boot to create containers within a VM without the need for manual user intervention. Running containers within a VM provides many benefits, especially when deploying containers without a container cluster management tool like Kubernetes. SC//HyperCore overcomes the challenges of implementing high availability inherent to any container deployment.

---

While SC//HyperCore is already incredibly powerful for this use case, Scale Computing continues to add features that will make containers even easier to deploy and manage. For example, integrations with various third-party tools such as IBM Edge Application Manager, Avassa, and Portainer can also help users centrally manage, deploy, and update containers. We will continue to invest in the future of application deployment.



## Kubernetes Distribution

Today, nearly any Kubernetes distribution can easily be deployed within VMs running on SC//HyperCore. Our REST-APIs and cloud-init support improve the speed and efficiency of deploying new K8S clusters and applications through infrastructure-as-code. All Kubernetes deployments (and any container deployments) make use of our redundant storage and compute architecture to ensure application uptime. For this reason, even single-node Kubernetes clusters can be deployed on SC//HyperCore and provide full resiliency for stateful applications and data as well as the Kubernetes control plane / API server itself, greatly reducing the complexity and resource requirements for deploying Kubernetes to edge locations. If a multi-node Kubernetes cluster is required to share stateful application data across the K8S cluster, customers can make use of Container Attached Storage by turning to Kubernetes storage extensions, like OpenEBS. Additionally, Kubernetes can connect to an external storage device like a NAS/SAN or run a virtual NAS within a resilient VM on Scale Computing to serve NFS or object storage.