Arrikto

Rok: Decentralized storage for the cloud native world

Cloud native applications and containers are becoming more and more popular, as enterprises realize their benefits: containers are great at providing a portable, isolated, and lightweight execution environment for applications. However, things start to get very tricky when stateful applications enter the game, because they depend on storage to persist their state. And portability of storage is a much harder problem to solve, by far.

We argue that in the cloud native world there should be a unified data management/ distribution layer that provides all the necessary data services (clones, snapshots), and is totally independent from the persistence layer, which is where the data reside. Moreover, this data management/distribution layer should not depend on the environment that the application runs.

IT admins demand performance, flexibility, and high availability for their stateful apps. Their constant dilemma is whether they should deploy their workloads on shared or local storage. On the one hand, shared storage provides flexibility, and high availability, but it is expensive and lacks in performance. On the other hand, local storage delivers high IOPS with extremely low latency, but it misses the flexibility feature. In the case of cloud native applications, high availability at the storage layer is not a real issue, since the design of cloud native apps caters for high availability from the very start. However, operational flexibility is a must have!

What you really need is running your application over fast, local NVMe SSDs, while keeping the flexibility of shared storage at the same time.

Arrikto builds Rok:
Decentralized storage for the cloud native world. Rok is the global data distribution layer that makes your apps truly performant and at the same time truly portable, in a fraction of the cost of a shared storage solution.

Rok allows you to run your stateful containers over fast, local NVMe storage on-prem or on the cloud, and still be able to snapshot the whole application, along with its data and distribute it efficiently: across machines of the same cluster, or across distinct locations and administrative domains over a decentralized network.

	Local storage	Shared storage	Local storage & Rok
Performace	✓		✓
Cost	✓		✓
Flexibility		✓	✓
High Availability*		✓	

Figure 1: Comparing the available storage solutions

* cloud native apps are highly available by design

We think performance and flexibility shouldn't be mutually exclusive anymore. One should have both. Everywhere.

Rok is fully integrated with Kubernetes, whether it runs on AWS, GCP, VMware, bare metal on-prem, and with Docker, so you can even run it on your laptop.

	Shared Storage	NVMe + Rok	Comparison
Storage Type EBS	EBS (io1)	Local NVMe	-
Storage Capacity (raw)	100 TB	100 TB	-
Instance Type	c4.4xlarge	i3.4xlarge	-
Number of instances	27	27	-
Total vCPUs	432	432	same
Total GB of RAM	810	3,294	4x better
Nominal aggregate write IOPS	432 K	9,720 K	22x better
Nominal aggregate read IOPS	432 K	22,275 K	51x better
Cost per month	$50,838	$18,016	64% cheaper

Figure 2: Comparison of running a Cassandra cluster on AWS over shared storage (EBS), and local NVMe-backed instances along with Rok. [Read the article here](#)

Key Features & Benefits

Performance

Rok allows you to run your applications over local NVMe SSDs inside your data center, and NVMe-backed instances on the cloud. This way, you enjoy the blazing fast storage that NVMe SSDs offer:

• extremely high IOPS
• I/O latency in the order of μs
• massive scale-out, with excellent scalability as the cluster grows
• significant cost savings compared to running over shared storage

Flexibility
Rok gives you the flexibility of shared storage:

• fast node migrations
• local backups
• offsite backups

Your applications may run over local NVMe SSDs, but you keep the flexibility feature, as if they were running over shared storage. Rok provides a data management layer that makes it possible for you to instantly snapshot your containers for local and offsite backups, and keep these backups in an immutable backup store, e.g. S3. Combined with Rok's instant cloning, you can recover from hardware failure in minutes, regardless of the volume's capacity.
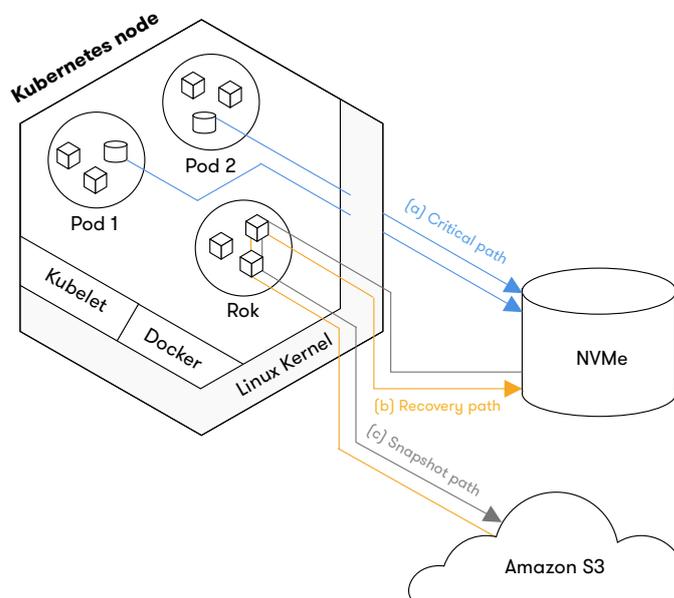

Figure 3: Architecture: A single Kubernetes node that uses NVMe SSD for the critical path, and Rok on the side for the data services

(a) critical path: goes directly through the Linux kernel to NVMe, without any Rok involvement. Minimal latency.
(b) recovery path: Rok intervenes to produce a ready-to-use, thinly cloned volume from snapshot data. This is completely transparent to the cloud native application. In the background, Rok continues to fetch data from the immutable backup store to the local NVMe. When this is done, operation reverts to (a).
(c) snapshot path: Rok sits on the side, tracking the set of blocks that the application has touched on the local NVMe. Periodically, Rok produces a new application snapshot by incorporating these changes into a new deduplicated snapshot, and pushes it to the snapshot store. Users set snapshot policies per application, per container, or per individual volume.

Collaboration at global scale

Running your applications over local storage together with Rok provides you with a feature that was never possible before, collaboration at global scale. This means that you can take a snapshot of a whole application, along with its data, and share it with another user of a completely distinct administrative domain, at a distinct location. This is a perfect match for test & dev use cases, analytics, or forensics.

Rok connects the whole of your global infrastructure. It runs on-prem, on any cloud, and on your laptop, and turns your silos into a decentralized storage network, enabling global, end user-driven collaboration workflows.
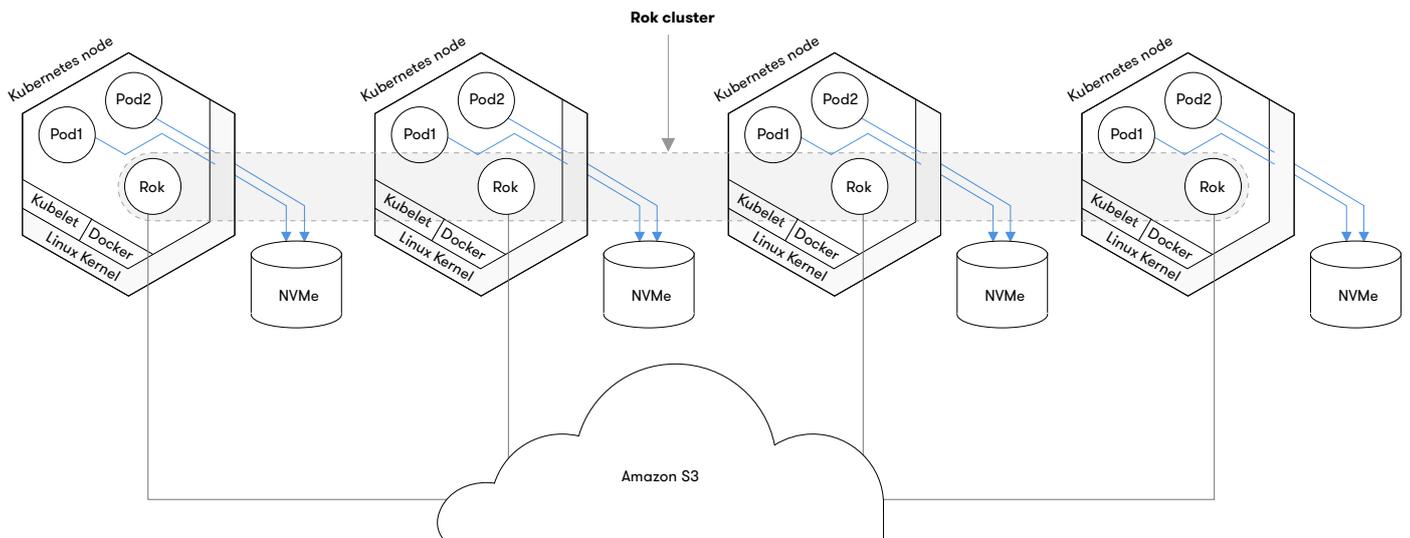

Figure 4: The Rok Cluster: Run your applications over blazing-fast NVMe SSDs, and use Rok on the side to enjoy shared-storage-like flexibility

Solving the 4 major problems stateful apps face today

Data Mobility - Multi-cloud Environments

It is out of the question that enterprises desire workload portability. Stateful applications should be fully portable, so that companies can take full advantage of the new multi-cloud world that is already here. Containers have already solved the problem of runtime portability. However, data is not portable yet, and this is a big obstacle to adopting containers in the enterprise world. An important reason why data portability does matter is "data gravity", which means that apps have to be physically close to the data they manage for reasons of efficiency.

Rok solves the problem of data mobility by providing a data management and distribution layer that makes data fully portable, so that they can move freely towards the app. All this, without making compromises regarding performance. Rok enables you to share an entire application stack, among different regions of the same public cloud, or even among different public or private clouds.

Spawning time and node migrations

Your workloads run directly over local storage with latency in the microseconds. Rok works on the side of your local storage stack, and provides all the required data services (efficient snapshots, clones). This enables instant container recovery on any other node. You can now spawn the container on any other node of the cluster from a previous

snapshot and be up and running instantly regardless of volume size.

Rok only has to intervene during node recovery, which is transparent to the application. It does not invoke an application-wide data recovery and rebalancing operation, putting load on the whole cluster and impacting application responsiveness. Instead, it performs block-level recovery of this specific node from the Rok backup store, e.g., S3, with predictable performance. Let's illustrate this with an example: if you lose an entire Cassandra node, then your database will continue operating normally, as Rok in combination with Kubernetes will present another Cassandra node. This node will have the data of the latest snapshot that resides on the Rok snapshot store. In this case, Cassandra only has to recover the changed parts, but this is just a small fraction of the node data, and does not cause CPU load on the whole cluster.

Backup and Recovery
In the enterprise world, it is of vital importance to be able to back up and recover entire applications along with their data. Companies cannot afford data loss, while even the slightest outage time costs a lot of money. Rok ensures business continuity in the cloud native world in case of a failure. It enables you to back up a whole application by creating a group consistent snapshot of its containers in an app-independent way.

Moreover, you can have these snapshots available in all your locations. Therefore, on-site and off-site backup becomes a single operation, defined by the desired backup policies. Rok allows you to unify your whole infrastructure, and move your data around over a decentralized network. A location can be a data center, a public cloud, a branch office, or even a laptop, anywhere in the world.
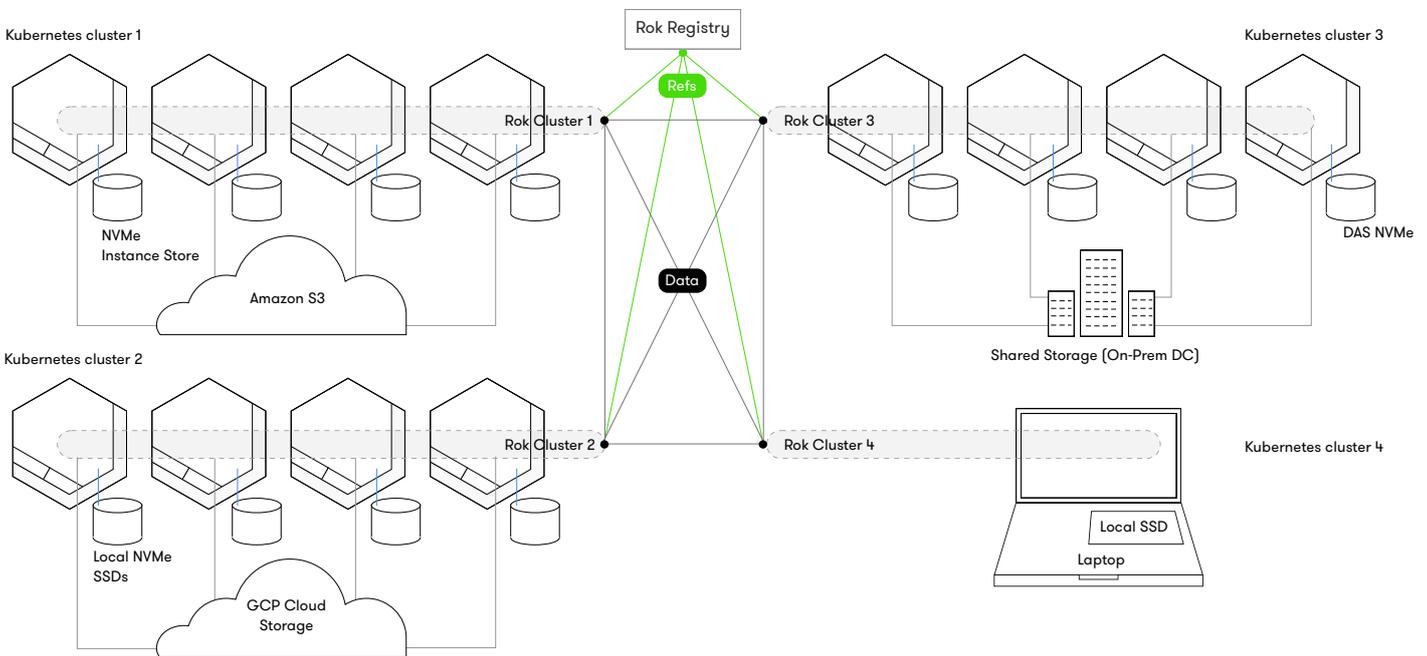

Figure 5: The Rok Network: Rok enables you to share your snapshots among locations over a decentralized network

Collaboration workflows with real data
Rok allows you to take a snapshot of an entire application along with its data, and share it with another user of a completely distinct administrative domain, at a distinct location. This is a perfect match for test & dev workflows, analytics, or forensics. CI/CD

test cases can now run using real data, in parallel, as Rok enables developers and CI/CD infrastructure to exchange data seamlessly.

Developers can share the whole application, as well as its state, with other developers, to ensure that all developers work with real data. Moreover, they can now run tests with real, production-derived data. You don't have to worry anymore, that something is going to break when deploying in production. Using Rok, the IT administrator can share a snapshot of the production database with the developers. The developers can thin-clone the snapshot on development and pre-production infrastructure.

Arrikto

rev. 20180206-2