The SINK Report

Updates on Samba in Containers and Kubernetes

sambaXP 2022

John Mulligan
Red Hat

SAMBA Red Hat

An Overview

Recap

- The state of the SINK
- Use Cases

What's New

- Recent developments
- Work in Progress

Future

- To Be Continued
- Scattered Thoughts

Previously, on Adventures in sambaXP

- The samba-operator project was announced in 2020 by Michael Adam
- Why containers? Easier to orchestrate and automate
- Why an operator? Improve the orchestration workflow (on Kubernetes)
- In 2021 we demonstrated the state of the operator, with:
  - New Share oriented resource definitions
  - AD support for containers
- We defined a suite of project repos under a github org:
  - https://github.com/samba-in-kubernetes
- We created a quay.io organization to host our container images:
  - https://quay.io/organization/samba.org

Our Intended Use-Cases

The reasons we're working on this

Windows Virtual Machines

Windows VMs running within an existing cluster (kubevirt). SMB is the natural choice for attaching shared storage to these systems. Dynamically provisioned storage along with dynamically provisioned VMs.

Windows Workers

Worker nodes based on the Windows platform running Windows containers. SMB is the natural choice for Read-Write-Many file storage for volumes backing these applications. Dynamically provisioned storage.

NAS Users

Traditional file-share workflows on the client side. Clients outside kubernetes! Kubernetes as the future "base-OS". Common declarative management workflows for administrators.

Recent Developments

Focus on Availability

- CTDB Enabled containers
- Cluster-backed Shares in StatefulSets

Focus on Testing

- Improved test environment using CentosCI
- Improving test coverage

Focus on Platforms

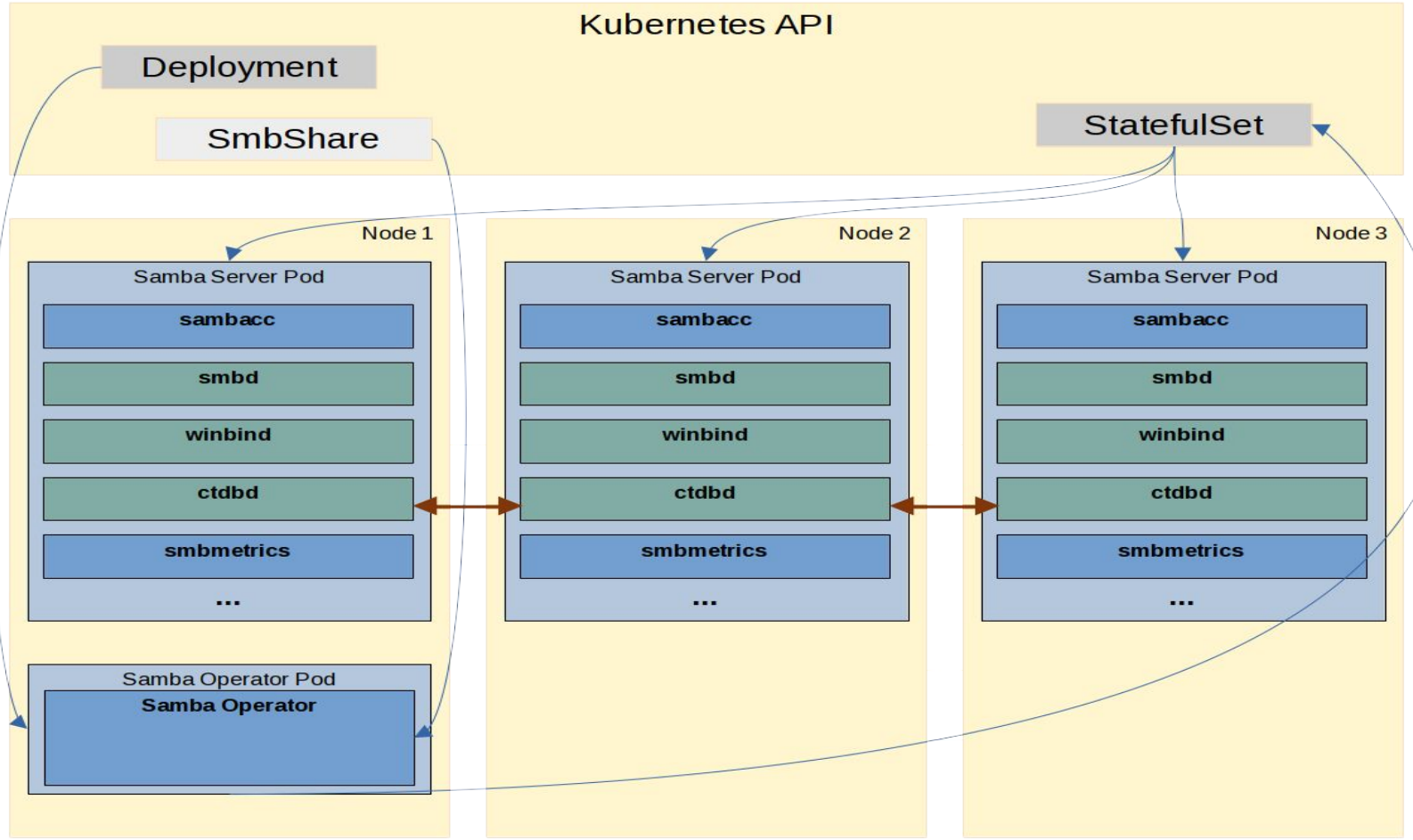
- Standard Kubernetes
- minikube
- OpenShift

Focus on Team

Bringing new Contributors on board:

Sachin Prabhu, Anoop CS, & Shachar Sharon

Clustered Operation - Block Diagram



Really, it's all about orchestration

- Using declarative configuration in the software stack
- Bringing multiple tools together, but with isolation
- Providing interfaces and utilities that coordinate between components:
  - svcwatch - monitors k8s network resources and helps transfer externally determined IP addresses to AD DNS
  - smbmetrics - extract metrics from Samba and share them with Prometheus monitoring stack
  - CSI Provisioner - dynamically create & delete Shares for applications in the "Cloud Native" way

Highlighting the smbmetrics project

- Prometheus is a de-facto standard in the Kubernetes & Cloud space
- Our Samba based applications can now export metrics like many other apps in a Kubernetes cluster
- However, neither are container or Kubernetes specific
- It is “just” a Go binary that executes standard Samba CLI tools
- Sound interesting? Drop by & say Hi:
  - https://github.com/samba-in-kubernetes/smbmetrics

The Road to Release

- We're working on our first releases!
- We're confident that the core functions are ready for more exposure
- Creating a release process
  - Tagging the git repos
  - Building and tagging the containers
  - Documentation
- Release version v0.2 completed for: svcwatch and sambacc
- Other project repositories in progress
- Watch for the announcement email

Longer Term Topics

- CSI Provisioning
- Full NT ACL support, Security & AD Domain improvements
- APIs - New and improved
- Additional CTDB integration improvements
- Keeping up with the Kube
- First class AD DC support?

Ramping Up: the CSI Provisioner

- The current Custom Resource Definitions (SmbShare, etc) are pretty great for a human to use (we think!)
- But many apps in a Kubernetes cluster want storage on demand
- This is typically done by creating PVCs (Persistent Volume Claims) and these are fulfilled by CSI provisioners
- A Windows worker node that wants a Read-Write-Many (shared access) volume? Give it an SMB share!
- A rough prototype was created in 2021
- It's time to dive back in and make dynamic provisioning a well-supported option:
  - https://github.com/samba-in-kubernetes/csi-provisioner-smb

Security and ACLs

- We want full NT ACL support - especially for “User Shares”/”Home folders”
- It’s important to us to keep containers isolation through the typical mechanisms
- The acl xattr MR:
  - We need NT ACL metadata when smbd does not possess CAP_SYS_ADMIN
  - https://gitlab.com/samba-team/samba/-/merge_requests/1908
  - What other implications does changing xattr namespace hold?
- We need automation for initializing the permissions/ACLs
- Other security automation: Offline Domain Join support
- Operating under tighter default security of OpenShift
- Anything we’re overlooking?

Having and Using Samba APIs

- All this automation makes APIs ever more valuable
- Recent work to add “smbconf” API wrappers for Python:
  - https://gitlab.com/samba-team/samba/-/merge_requests/2469
  - https://gitlab.com/samba-team/samba/-/merge_requests/2500
- Other Python APIs - Domain JOIN, etc.
- Added JSON support to various CLI commands
  - Machine parsable output is huge
  - Great when APIs are not available, or not using Python or C
- Better when Testing too!

CTDB Improvements

- Improving the behavior of failover
  - Currently use Kubernetes native load balancing (Service)
  - We're investigating combining Multus with CTDB native IP failover
  - Not for every cluster
  - Challenging to test so far
- Need to resolve issues with failover of cluster leader
- Node management in today's CTDB is "static"
  - Adding "nodes" isn't too bad; shrinking a cluster is a TODO item
  - We'd love a more dynamic way to manage CTDB nodes
  - DNS names, IPv6 addresses, etc?
- Having appropriate CTDB events trigger the right Kubernetes events

Keeping Up with the Kubernetes

- Users have varying requirements for pod placement, etc.
  - We want to give users a way to construct the Pods, Deployments, Stateful Sets the way they want, while still keeping share management simple
  - Different ways of integrating with networking too
- Kubernetes is changing all the time as well - can we make use of new features - and new developments in the wider ecosystem
- Should we offer other methods of deployment: Helm? Operator Hub?

AD Domain Controller Support

- We have an AD DC image:
  - Our team uses the image for testing
  - We have updated it to use JSON configuration like file server image
  - The new JSON config is not heavily utilized
  - Single DC instance
  - Based on Fedora Samba RPMs
- Should we be doing more with the AD DC image?
  - Should the operator be able to manage an AD stack too?
  - What about a single-use-only domain and skip “static” users & groups?
- Looking for community feedback and contributions!

Some Rather Scattered Thoughts

- Could we separate winbind to run outside the smbd pod
  - TCP sockets? DNS??
- Cluster support in SMB protocol
  - Tickle-ack feels like a workaround for limited environments
- UNIX Extensions - tell all our client OSes “just use SMB” :-)
- Is SMB over QUIC even more cloudy?
  - Would it work well with (future) k8s networking infrastructure the way HTTP(S) is well supported today?
- Networking, networking, networking - what are we missing?

Jump On In - The Water's Fine

- We experienced a small but noticeable increase in questions and comments
- We've been sending regular status updates to the Samba mailing lists
- But we need **more**!
- We would dearly like to see additional contributors and users
  - Kubernetes Operator
  - Container images
  - Anywhere in the stack that interests you
- Just send me an email telling us where/why we've done it all wrong ;-)

Questions?

Thank you very much!

John Mulligan

- phlogistonjohn@asynchrono.us
- jmulligan@redhat.com
- http://asynchrono.us

SAMBA Red Hat