

# samba-operator The Next Phase

sambaXP 2021

John Mulligan  
Red Hat

Michael Adam  
Red Hat / Samba Team



# What we'll discuss today

## Project Updates

- ▶ Look Back
- ▶ Use Cases
- ▶ Organization

## Operator

- ▶ Components
- ▶ Design
- ▶ Resources
- ▶ Demo

## Future

- ▶ Short Term
- ▶ Long Term
- ▶ Working Together

## At sambaXP 2020

A prototype operator was introduced and demonstrated

- ▶ The prototype operator
  - <https://github.com/obnoxxx/samba-operator>
- ▶ Created Samba based instances with a static share and user
- ▶ Two Custom Resources (CRs):
  - `SmbService` - directly created a server instance
  - `SmbPvc` - created a `SmbService` and matching PVC

# Kubernetes Terminology

## A Refresher

- ▶ Pod, container
- ▶ Deployment
- ▶ PersistentVolume (PV), PersistentVolumeClaim (PVC)
- ▶ Provisioner / StorageClass (SC)
- ▶ Operator, Controller
- ▶ CustomResourceDefinition (CRD), CustomResource (CR)
- ▶ Service
- ▶ Secret

# Our Intended Use-Cases

The reasons we're still 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.

## Fast-Forward to Today

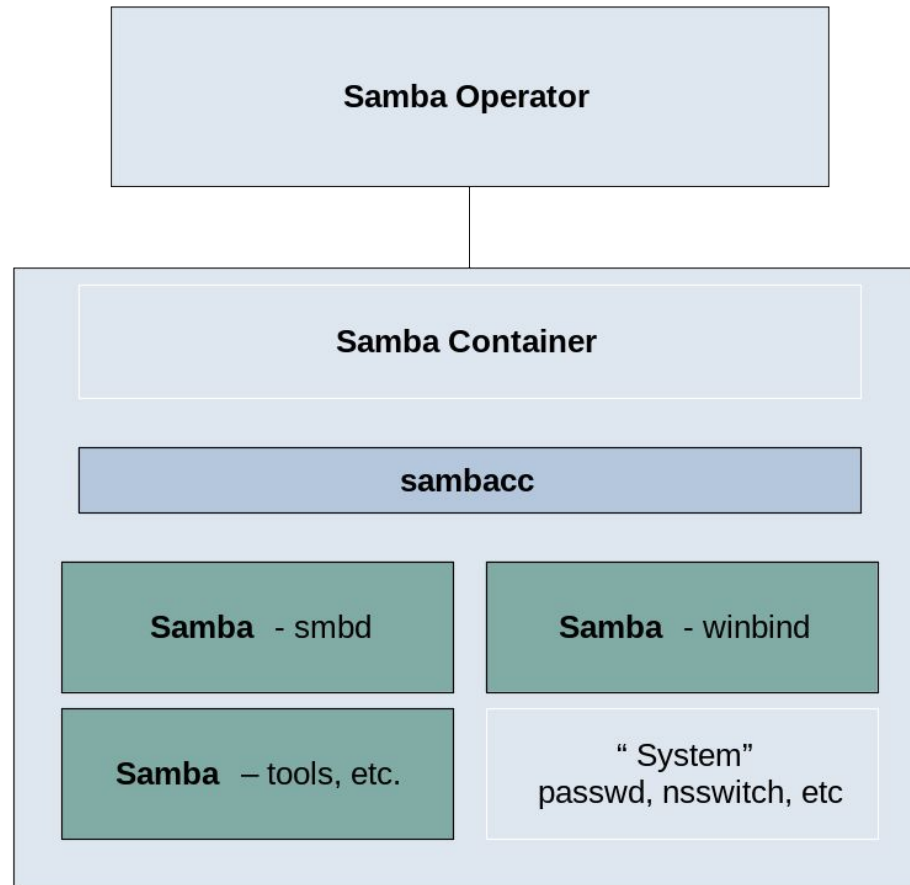
- ▶ Focusing on the NAS style use case first (but the others too...)
- ▶ Targeted the low hanging fruit
  - Improving the basic containers
  - Focusing on a declarative workflow
  - Supporting basic customizations - custom share names, etc
  - Active Directory support
- ▶ Adding people to the team working on the operator and supporting projects
- ▶ Establishing a proper github organization and ecosystem
  - <https://github.com/samba-in-kubernetes> (SINK)
  - <https://quay.io/organization/samba.org> (container registry)
  - PR → test (in k8s) → merge → build image → publish in quay.io

## How We've Organized Our Projects

- ▶ Samba Operator
  - Orchestrates resources within a Kubernetes environment
  - <https://github.com/samba-in-kubernetes/samba-operator/>
- ▶ Samba Container
  - Create OCI images containing Samba binaries and tools
  - File Server, Client, and AD DC Server (currently test only)
  - <https://github.com/samba-in-kubernetes/samba-container/>
- ▶ sambacc
  - Glue library and "samba-container" script
  - Abstract and unify aspects of managing samba in a container
  - <https://github.com/samba-in-kubernetes/sambacc/>

# How Does It Fit Together?

A block diagram





# How to be Cloud Native?

# How to be Cloud Native?

- ▶ YAML
- ▶ YAML
- ▶ YAML
- ▶ Some More YAML to configure your YAML
- ▶ YAML
- ▶ A little JSON

## How to be Cloud Native?

- ▶ Focus on the End Result
- ▶ Take a declarative approach
  - Yes, often expressed in YAML
- ▶ Let code handle most of the complexities
- ▶ Take a few hints from the microservices trend:
  - Break up certain tasks by component
  - Isolate components (when reasonably possible)

# Our Redesigned CRs

Users of the Operator do not set up servers

## SmbShare

- ▶ What do you want to share?
- ▶ What storage do you want backing your share?
  - Request a PVC by name or embed a PVC specification
- ▶ How do you want to share it?
- ▶ Who can access your share →

Servers are inferred and created by the operator as needed to support shares.

## SmbSecurityConfig

- ▶ Predefined Users/Groups or Active Directory
- ▶ Users:
  - Pointer to a secret (a secure resource) holding a JSON representation of your desired users & groups
- ▶ Active Directory:
  - What domain (A.K.A. realm) to use
  - How to join to the domain
  - Pointers to a secret holding JSON representation of domain authentication info

## So What's All This Do, Again?

- ▶ Operator is informed when CRs are created, updated, or changed
- ▶ Makes changes to native Kubernetes resources
  - PVCs
  - Deployments → Pods
  - Services
- ▶ Pods execute containers - containers run processes:
  - Set up environment
  - Join AD - unattended if possible
  - Run smb
  - Run winbind

## Time for a Demo



## Demo Availability

- ▶ Most of what's shown in the demo is available today
- ▶ Support for Services with type of LoadBalancer & auto-registering AD DNS entries needs work
- ▶ Everything shown in the demo will be submitted in one form or another
- ▶ Demo code and scripts are available in the following locations:
  - <https://github.com/phlogistonjohn/samba-operator/tree/jjm-sxp>
  - <https://github.com/phlogistonjohn/samba-container/tree/jjm-sxp>
  - <https://github.com/phlogistonjohn/sambacc/tree/jjm-sxp>





## Short-Term Goals

- ▶ Submit patches for service management and AD DNS updates
- ▶ Documentation!
- ▶ Spit and polish
- ▶ Metrics and monitoring - smbstatus->prometheus
- ▶ Handling persistent metadata / make containers deterministic
- ▶ Investigate failover/clustering/scale-out with CTDB or alternatives
- ▶ Testing, testing, testing...

# CTDB, failover, etc...

What we're currently looking into...

- ▶ What do we need CTDB for?
  - HA / failover ⇒ How far does kubernetes HA get us?
    - Fencing etc...
  - Scaling out (supporting more concurrent users)
    - Many shares ⇒ covered by containers!
    - Many users per share ⇒ might need CTDB
- ▶ Reasons to avoid CTDB (just yet):
  - Because we can (for some parts) ⇒ reducing complexity
  - Static and somewhat delicate nodes file
  - ...
- ▶ ⇒ How much would CTDB need to change?

## Middle & Long Term Goals

- ▶ Offline Domain Join (ODJ) support (⇒ Günther's talk!)
- ▶ NTACL Support
- ▶ Experiment with Non-PVC backed storage: CephFS
  - How fast is the simple approach? Fast enough? ...
- ▶ Enhanced idmapping support
- ▶ "Vertical scaling" - hosting more than 1 share per-pod
- ▶ Additional kubernetes integration - kubernetes-based clients (windows workers, kubevirt vms, ...) & CSI provisioning
- ▶ Direct support for Samba AD DC - new CR/etc
- ▶ Possibly move under the "rook" umbrella?

## What Might We Need from Samba?

- ▶ NTACL Support without running as CAP\_SYS\_ADMIN
  - MR from Ralph:
  - [https://gitlab.com/samba-team/samba/-/merge\\_requests/1908](https://gitlab.com/samba-team/samba/-/merge_requests/1908)
- ▶ Reduced coupling between processes
- ▶ CTDB: more automated configuration ("cluster join"...)
- ▶ Dbwrap:
  - Separate config for volatile and persistent dbs?
  - Access serialization and intactness check for backend storage (like ctdb reclock)
- ▶ APIs? - In Python?

## How Can this Benefit Samba's Community?

- ▶ Integration into the "Cloud Native" Ecosystem
  - Possibly new users and contributors!
- ▶ Easier deployment & declarative configuration
  - New *kind* of users
- ▶ OS distribution agnostic deployments
- ▶ Chance for code cleanup and component segregation

# Questions?

Thank you very much!

Michael Adam

- ▶ [obnox@samba.org](mailto:obnox@samba.org)
- ▶ [obnox@redhat.com](mailto:obnox@redhat.com)

John Mulligan

- ▶ [phlogistonjohn@asynchrano.us](mailto:phlogistonjohn@asynchrano.us)
- ▶ [jmulligan@redhat.com](mailto:jmulligan@redhat.com)
- ▶ <http://asynchrano.us>