PRD: Definitive bhyve VM manager

Introduction

Mid 2023, Greg Wallace at FreeBSD Foundation received inputs from enterprise stakeholders about gaps in FreeBSD's enterprise capabilities. This led to the formation of the Enterprise Working Group.

Within this group and out of the original feedback, one work stream emerged for specifying and building a “definitive bhyve+jail management toolset” to be added to FreeBSD. This may be accomplished in different ways - this work stream intends to evaluate options and identify a way forward that is feasible, delivers results reasonably fast and is compatible with FreeBSD’s design.

The main pain points mentioned in initial communications:

- High manual effort required to get jails running
- Lack of resources in maintaining ports providing utilities (i.e. run by one person) and some of them even not maintained and developed further at all
- Insufficient oversight and management of needs and requirements, lack of progress
- Low visibility and insufficient marketing of available technology options to enterprise customers

Customer needs, market situation

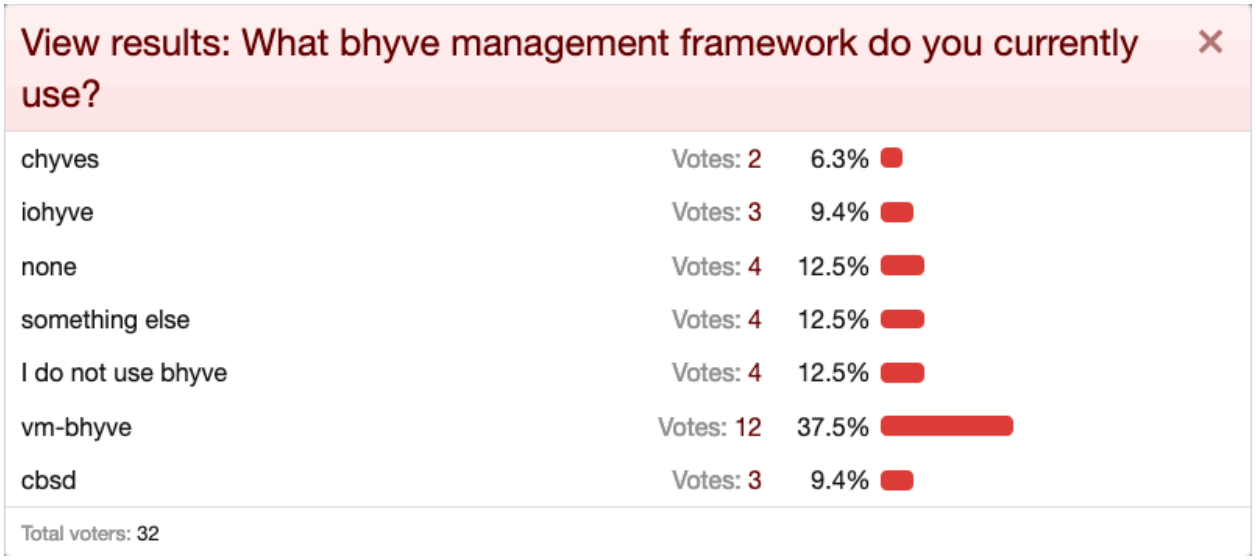
Existing BHYVE managers

Review of existing ports confirm a particular pain point: development has either ceased or only a single committer is supporting further development and addressing bugs.

Port Name	Active?	Commits 2023	# of committers in 2023
VM-Bhyve	Low	3	3
CBSD	Busy	>10	1
RunHyve	Abandoned 2022	0	2
Iohyve	Abandoned 2017	0	1
Web administration	Medium	15	1

Port Name	Active?	Commits 2023	# of committers in 2023
VM-Bhyve	Low	3	3
CBSD	Busy	>10	1
RunHyve	Abandoned 2022	0	2
BVM	Abandoned 2022	0	1
Chyves	Abandoned 2017	0	0
AppJail (jails)	High	>100	1

- Virt-manager?



Source: https://forums.freebsd.org/threads/bhyve-management-poll.59692/

Overview of ongoing activities

	Coverage Area		Ready for Base?	Development Lifecycle Stage							
Activity Name	VM	Jail		Prep	Req	Design	Dev	Test	Port	Rev	Merge
bhyve Management Toolset											
bhyve & Jails State Engine	○	○	Y								
bhyve multi-host management	○		Y								

bhyve Capabilities											
9p file system	●		Y								
tarfs support storage backend		○	Y								
Snapshot / motion, IP mobility https://reviews.freebsd.org/D30954	○	○	Y								
iSCSI storage access	○										
NFS storage / libnfs	○										
Boot root NFS - https://github.com/stblassitude/boot_root_nfs	●		Y								
Security Enhancements											
Jailing bhyve	●		Y								
Additional Existing Port Development											
OpenStack	●		N								
LibVirt	●		N								
Podman		●	N								
XC - https://github.com/michael-yuji/xc		●	N								
Jailer	○	●	Y								

○ feasible ● available

Live migration activities

ID	Title	Status	Created	Last Update	Reviewers
D26387	bhyve - Snapshot Save and Restore multiple devices	Needs Review, landed in a different review	Sep 10 2020, 10:21 AM	Jun 21 2023, 6:15 AM	jhb

ID	Title	Status	Created	Last Update	Reviewers
D28270	Warm Migration feature for bhyve	Needs Review	Jan 21 2021, 2:19 PM	Jun 30 2021, 9:30 AM	jhb
D29262	Bhyve - Using JSON format for saving and restoring the state	Needs Review	Mar 14 2021, 5:37 PM	Jun 21 2023, 6:15 AM	jhb
D29538	bhyve: Move the gdb_active check to gdb_cpu_suspend().	Closed/Accepted	Apr 1 2021, 4:47 PM	Apr 4 2021, 8:53 AM	-
D30471	Bhyve - Capsicum integration	Needs Review, landed in different revision	May 26 2021, 3:48 PM	Jun 21 2023, 6:15 AM	jhb
D30954	Live Migration feature for bhyve	Needs Review	Jun 30 2021, 10:31 AM	Jul 26 2021, 5:45 PM	jhb
D33431	bhyve snapshot fix for AMD CPUs	Closed/~~Denied~~ Accepted	Dec 14 2021, 9:22 AM	Jun 30 2022, 11:15 PM	-
D34547	bhyve - snapshot capsicum integration[Part 1]	Needs Review	Mar 14 2022, 1:41 PM	Jun 21 2023, 6:15 AM	rew

ID	Title	Status	Created	Last Update	Reviewers
D34717	Warm Migration feature for bhyve [Part 1]	Closed/Accepted	Mar 31 2022, 6:36 AM.	Jun 19 2023, 6:55 AM	rew, imp, corvink
D34718	Warm Migration feature for bhyve [Part 2]	Needs Revision	Mar 31 2022, 6:47 AM	Tue, Nov 7, 7:59 PM	(no:)afedorov, markj, jhb, rew
D34719	Warm Migration feature for bhyve [Part 3]	Needs Review, Changes Requested	Mar 31 2022, 6:50 AM	Jun 20 2023, 11:42 AM	afedorov, (no:)gusev dot vitaliy at gmail.com
D34720	Warm Migration feature for bhyve [Part 4]	Needs Review	Mar 31 2022, 6:53 AM	Jun 23 2023, 5:34 AM	corvink
D34721	Warm Migration feature for bhyve [Part 5]	Needs Review	Mar 31 2022, 6:56 AM	Jun 14 2023, 3:29 PM	corvink
D34722	Live Migration feature for bhyve [Part 1]	Needs Review	Mar 31 2022, 6:57 AM	Jun 14 2023, 3:30 PM	corvink
D34811	Live Migration feature for bhyve [Part 2]	Needs Review	Apr 6 2022, 11:52 PM	Jun 14 2023, 3:30 PM	corvink

ID	Title	Status	Created	Last Update	Reviewers
D34813	Live Migration feature for bhyve [Part 3]	Abandoned	Apr 6 2022, 11:53 PM	May 18 2023, 10:08 PM	None

Known customers and customer requests

1. Bhyve customers?
2. Michael Osipov at Siemens:
   Definitive managers of bhyve VM (e.g., vm-bhyve) and jails (e.g., Bastille): This is a pain for us in the enterprise as well as the entire, broader community. While FreeBSD has exceptional jails with a lot of possibilities, everything has to be done manually, jail managers arise, often run by a single person, as we have seen with ezjail, then there was iocage and others, now promising Bastille, but again, still the person-free-time problem (busload of tickets reported).

   I wish there would be a very decent jail manager packaged with FreeBSD base. To some extent this applies to bhyve as well: We now rely on vm-bhyve, requires whole compiler for UEFI guests (should be as slim as possible ideally), but here as well more or less one maintainer. There should be a manager in base as well.

   I know that no tool can cover 100%, but 80% would be more than enough for most and then the rest can be contributed by the community.

   I care about VNET jails and VM via TAP over a bridge only since they have their own VLAN here. Here I must admit that the Linux community has created much better high level tools, while low level with Docker daemon it is just an inferior solution compared to in-kernel jails.

   See Production User Action Items for Michael Dexter's Minutes of 2023/10/20 call w/ Michael Osipov.

Pain points

- Existing ports are either understaffed or not maintained anymore. Bugs and issues are not addressed quickly enough for enterprise needs.

- Maintaining bhyve VMs is often not only about managing the VM but also about managing its (remote) storage at the same time; in practice, vm storage could be provisioned via NFS, iSCSI, local disks etc. which all act and are maintained differently.
- Managing bhyve VM state is complex; i.e. return codes and different ways a bhyve process may get terminated. This makes correct and full clean up a challenge.
- There is no canonical way (i.e. through a library) to interact with bhyve or to react to relevant events around a vm (i.e. devd-like events on startup, shutdown, crash, etc.).
- There is no out-of-the-box way to manage multi-node bhyve hypervisors (i.e. across multiple machines).
- Bhyve does not (yet) support easily moving virtual machines from one host to another (neither with shared storage or other media, i.e. zfs). The same applies for jails.
- The current situation requires every user to implement home-grown solutions; this means that coding is done again and again for the same functionality, which increases load on already loaded resources and does not generate value for the community at large. That energy and coding could create value elsewhere instead.

Expected results

The goal is that FreeBSD gets an officially supported bhyve VM and jail management tool set that helps enterprise adoption and use. We want that tool set to be open and interoperable so it can be expanded and built upon further by the community.

Assumptions and Constraints

- Separate tooling for bhyve and jails
- Jail manager and bhyve may require a daemon or kernel module to better manage jail and vm states (does that follow KISS?); they might make use of the same state engine but would likely have some differences in their state set.
- Tooling in base - limits languages and frameworks to C, Shell and Lua
- Vale, netmap and netgraph are out of scope
- MVP: there’s a feasible minimum viable product we can agree on
- We intend to develop iteratively, which helps establish a reasonable pace at which the project team perceives and achieves measurable progress
- While this project’s decision making process is straightforward and fast, it operates in the FreeBSD ecosystem in general; that means, while its deliverables and designs can start out without major review cycles, they still need to be communicated and reviewed following common governance processes before they merge into base. Therefore, we intend to always be open and transparent on our work and welcoming to continuous feedback.

- We need to expect that base constraints apply: only what is necessary/required for all will go into base; for anything beyond we will have to consider ports.

Alternatives

We went into this work stream assuming that the solution is to build some tooling into base. To contrast this idea - what are the alternatives to this approach?

- Increase support and staffing for a select port(s)
- Improve documentation
- Simplify existing tooling
- Merge codebase of an existing port into base (if language compatible)
- Define a set of sane defaults for existing tools and configuration elements/files
- Improving bhyve integration with existing tools like virt-manager

Risks

- Coding stuck in review limbo before getting adopted into base
- Lack of development resources during/after implementation and publication
- Implementation too simple for enterprise use cases
- Implementation not addressing key pain points
- Development of base components does not improve support capacity and capabilities

(Product) Marketing and Communication

Key metrics

-
- …
- …

Separation of concerns

There is some overlap with jails management; for usability purposes the toolset should be kept separate in the end - even if it do;les share some code base.

Terms, wording, and copy

- Bhyve

- Virtual machine
- UEFI
- CSM
- bios
- …

Personas

Current and future FreeBSD users who wish to create, deploy, run, and/or orchestrate virtual machines on FreeBSD.

Functional Requirements

Description of key functional aspects on a very high level, for example Epic level. Detailed user stories will go into your ticketing system (Such as Jira, Github, Trello etc.). You can also link (some of) them here.

- Manage and handle different bhyve states - including failure states and unexpected terminations (i.e. crashes)
    - Keep security boundaries and ACLs in place so state management only allows authorized users to modify or retrieve VM states
- Manage different storage backends
    - Local disks or files - ZFS, UFS
    - iSCSI?
    - Block files over NFS? Libnfs use as storage backend integrated into bhyve
    - Allow users to securely store and use access credentials?
    - Diskless boot - https://github.com/stblassitude/boot_root_nfs
    - the 9p client out of Juniper
- Network configuration
    - Handling multiple interfaces and
    - Different VLANs
    - Connecting jails to bhyve VMs via bridge
- Multi node management
    - motion/transfer VMs
    - Know what is running where

  John, as a sysadmin of a multi-node bhyve environment, wants to easily move workloads across nodes in the cluster to keep downtimes at a minimum.

John, as a sysadmin of a multi-node bhyve environment, wants to have a full overview of what workloads are running on which node, so he can better understand the health of the cluster and whether any VMs should be moved.

- Secure console access
- Increase containerization/security of running VMs
    - wrap bhyve inside jail
    - Separate devfs rules?
    - Vnet interfaces?
- Provide a way to
    - template VMs
    - Snapshot VMs? Already should work! Needs testing

Network Handler

Why is there even an issue? Alternatives like kubernetes, docker, podman etc. are all maintaining some kind of network stack and make it "easy" to just plug in a new vm/guest that can connect to the host's network. `

There are some projects that are working on delivering docker, podman et. al onto FreeBSD - we should probably check: how are they addressing the space of networking and making things easy for sysadmins / users?

- Overall listing on the wiki, see [4]
- Podman see [5]
- Pot - can't say from command listing at [6]

There is a port implementing CNI on FreeBSD: net/containernetworking-plugins

Requirements

We certainly need to

- maintain virtual interfaces
    - tap interfaces for bhyve, including their MAC address
    - epairs for jails
    - would entail creation, renaming, destruction of such interfaces
    - tagging or associating them for use with a jail or vm or even a user-defined grouping
- connect interfaces together and to the outside world (including host)
    - bridges and assigned interfaces

    - vale (netmap) based switches and assigned interfaces
    - netgraph (?)
    - vxlan, wireguard, lots of gre tunnels, etc.
    - dispose of any bridges, switches etc. when no more users (jails, vms, ...)? reference counting?
- handle network addressing
    - for IPv4 as well as IPv6,
    - IP subnets,
    - available IP addresses,
    - IP assignments to vms or jails
    - We are unlikely to put any dhcpd server into base; those seem to be pretty complex
        - ISC has retired their ISC dhcpd and are now working on "kea"
        - source code for kea is quite big for "just" providing dhcp
        - complexity of dhcp can be seen on related wikipedia page
- route those IPs to the internet or NATing them, i.e. over `pf`
- want to provide hooks for lua or shell calls when state, assignments et al change; this would also allow us to add code for configuring dhcp servers, for example
- We need to fail gracefully in cooperation with existing base tools `bridge` and `ifconfig`, because we need to expect manual changes and need to be able to restore the system to working order when we detect changes
- We might need algorithms for doing algebra/calculus on IP subnets
    - understanding available number of IP addresses under different addressing schemas, identify what's the next free IP
        - using large subnets on a bridge - performance? simplicity
        - using `/30` subnets (host and guest IP pair) that are all routed - encapsulation
- Provide everything in an open, well documented library to be used by anyone besides the built tools by us
    - documented code
    - man pages

Functional considerations

- Management of interfaces, bridges, addresses and so on should probably be done under user-defined contexts, i.e. a namespace; this would allow us to group things together. I called it "tagging" before.
    - by vm or jail name
    - by a collection of jails that are sharing a particular VLAN
- Handling should consider multi host management; i.e. managing the distribution of an ~~IP~~ MAC address across multiple hypervisor hosts. ~~this might be obsolete if we handle tap~~

~~MAC addresses. It could be accomplished by using dhcp on the network instead.~~ I realized we are moving the problem statement from IP to MAC address if we are operating a dhcp server; we still need a multi host management - this will likely need some form of network communication across the hosts?

Non-Functional Requirements

This is often forgotten: things such as: how many users are expected to be supported at peak? Any latency expectations (page load in x ms) etc.?

- Support managing thousands of VMs concurrently

Design

Any high level UX/UI considerations to mention, but primarily link to any UX research or any prototypes as things progress.

State Handling / Process Supervision

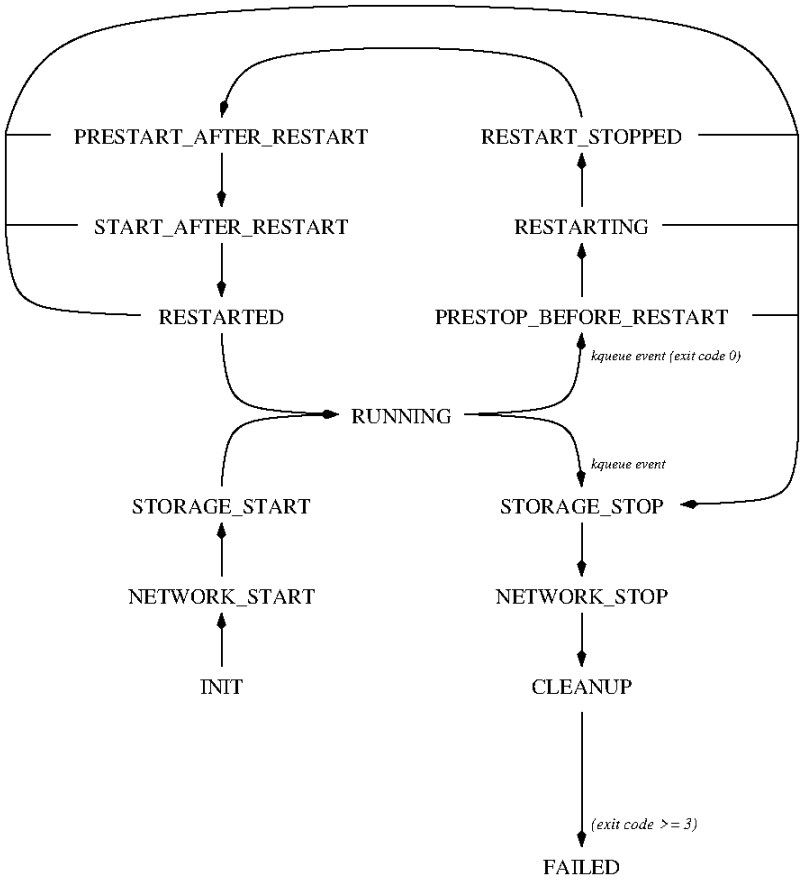
Question:

- do we know the expected states a bhyve vm would run through in its lifetime? For consistent error handling, we'd need to be aware of system elements we are setting up and tearing down?
- do we have two different kinds of failure states?
    - running failure
    - stopped failure

Actions in state nodes would have to be atomic - they either complete successfully or don't change the system at all (rollback). That (again?) puts the burden onto the sysadmin/user to create scripts that adhere to this rule. If it's broken, the problem space persists?
Solutions?

- delegate storage and network setup to "vmstated"?
- Assume stop_* is pure inverse of start_* able to handle failure cases?

- Need to stay generic enough to allow external tooling to access nmdm / console etc.



Technical Specifications

Anything that's relevant for/from engineering. Such as key technical challenges or risks.

Changelog

- 10/04/2023 - Initial version
- 10/06/2023 - cleaned up, reshared to include cm in change history
- 10/14/2023 - adding functional requirements, alternatives and assumptions, further research on existing ports, reference to wiki page
- 10/19/2023 - update after/during bhyve and dch call: 9p, movability of vms, governance
- 10/20/2023 - moving related documents to the end of the document, adding pain points coming out of call w/ Michael O and Johannes K
- 10/22/2023 - updated introduction text after receiving feedback on the forums, minor grammar fixes

- 11/1/2023 - adding technology roadmap for FreeBSD and spreadsheet for desired features in bhyve
- 11/2/2023 - added chyves; updated publicly accessible spreadsheet link
- 11/5/2023 - adding reference for bhyve life migration feature
- 11/9/2023 - added listing of ongoing and completed reviews on bhyve live migration
- 11/11/2023 - adding AppJail
- 11/22/2023 - adding network handling considerations
- 11/23/2023 - added CNI reference for network handling resource
- 11/24/2023 - moving changelog to the end of the document, updated hot migration review collection
- 12/20/2023 - adding state diagram

Related Documents

- https://wiki.freebsd.org/bhyve
- https://wiki.freebsd.org/Jails
- https://wiki.freebsd.org/EnterpriseWorkingGroup#bhyve_manageability
- https://www.sonatype.com/hubfs/9th-Annual-SSSC-Report.pdf
- https://github.com/michael-yuji/xc, https://www.youtube.com/watch?v=7__Iee8vmOw
- Dave Cottlehuber's and Michael Dexter's work on improving jails management: https://docs.google.com/spreadsheets/d/1IJ5MyIIZzHhakblAG0dM4mZkFKIXSpY3HskYYwuGG8w/edit#gid=0
- https://docs.google.com/document/d/1PFUmz6XpTVAGkq5dBe8uaBFV2Y4i-uR88AuiCLIRxIQ/edit
- https://www.youtube.com/@bhyvecon
- Bhyve diskless boot: https://github.com/stblassitude/boot_root_nfs
- Jailer - https://github.com/illuria/jailer/blob/main/README.md
- https://www.youtube.com/watch?v=RHLRW88AJLE - Hosting with FreeBSD jails
- Man pages?
- https://freebsdfoundation.org/blog/technology-roadmap/
- https://docs.google.com/spreadsheets/d/1g_zuadGsCsAxS-Luxb53i2ZUjU2tai-G16uN97cza88/edit?usp=sharing
- Migration feature under review - https://reviews.freebsd.org/D30954
- Commit for snapshot and requirements for using it: https://cgit.freebsd.org/src/commit/?id=483d953a86a
- https://developer.hashicorp.com/nomad/docs/networking/cni
- https://github.com/containernetworking/cni
-