

Vilniaus Gedimino technikos universitetas

Elektronikos fakultetas

Kompiuterijos ir ryšių technologijų katedra

Virtualizacija duomenų centruose

Modulis ELKRM17303

Laboratorinio darbo nr. 7 (+8) ataskaita

Atliko: TETfm-20 grupės magistrantas
Saulius Krasuckas

Tikrino: lekt. dr. Liudas Duoba

Laboratorinis darbas nr. 7 (+8)

Virtualizacija duomenų centruose

Darbo tikslas

Artimiau susipažinti su virtualizacija ir patyrinėti pasirinktą *Type-1* hipervizorių.

Darbo užduotis

- Parenkite trumpą ataskaitą, kurioje:
 - bus pateiktas Jūsų pasirinkto **I** -tipo hipervizoriaus pristatymas (pagrindinės funkcijos ir pan.)
 - laikysitės principo: *Less is more*; svarbu kokybė.
 - išlaikykite vienodą stilių.
- Parengtą ataskaitą PDF formatu įkelkite į **Moodle** sistemą

Darbo eiga

Apžvalgai pasirenku dedikuotą **SmartOS** hipervizorių, minimą laboratorinio darbo apraše.

Kas yra SmartOS ?

Kalbant tiksliau, tai konverguota hipervizorinė operacijų sistema (OS).

Tai distribucija ne populiarios Linux OS, bet mažiau žinomos "*illumOS*" sistemos (2010 m. kilusios iš Solaris OS), ir yra klasikinė Unix OS. Lyginant su Linux, ji turi gerokai stabilesnę architektūrą.

Du hipervizoriai kartu

Virtualių mašinų (toliau: VMų) valdymui SmartOS siūlo du skirtingus hipervizorius pasirinktinai:

- **KVM** [21] psl. 5 + 14
(gana senos Linux 2.6.34 versijos VMM + senos *Qemu* 0.14.1 versijos *Device* emulatoriaus portas, 2011 m.);
- **bhyve** [24] psl. 6
(gana naujos FreeBSD 11.1 versijos VMM portas, 2017 m.)

Tiesą sakant, jie abu nėra gryni *Type I* hipervizoriai. [1]

Jie abu startuoja ne „plikame metale“, bet jau veikiančioje įprastinėje OS (Linux, FreeBSD, SmartOS). Ir jie abu tą OS paverčia *Type I* hipervizoriumi, suspecializuoja.

Tačiau kadangi tokia OS tinka ir bendros paskirties naudojimui, pagal apibrėžimą tokį hipervizorių priimtina laikyti ir *Type II* hipervizoriumi.

Dėl aiškumo juos abu priskiriu hibridinei *Type 1.5* (dar vadinamai *Type 3*) kategorijai. [2]

Abiejų hipervizorių palyginimas

- **KVM:**
 - Gana paplitęs VM mechanizmas / interfeisas;
 - SmartOS KVM implementacija neleidžia plėsti VM resursų: diskų, CPU, RAM; (tenka VM perkurti iš naujo ir perkelti aplikacijos duomenis)

- palaiko iki 256 vCPU / VM; [25]
- palaiko iki 2 TiB RAM / VM;
- su *illumOS* išeities kodu nesuderinama GPL licencija; [29]
(Todėl SmartOS KVM vystomas kaip atskiras projektas)
- Linux branduolys ir KVM yra „judantis taikiny“ su kitokia architektūra nei *illumOS* sistema.
Į pastarąją pastoviai perkelti naujus pakeitimus būtų itin daug darbo.
Todėl SmartOS KVM implementacija nenaujinama.
- **bhyve:**
 - *bhyve* kilo irgi kitoje OS, *FreeBSD*; [8]
Naujos funkcijos ten atsiranda anksčiau / greičiau.
 - palaiko šiek mažiau OSų / OS atvaizdų;
 - palaiko tik 16 vCPU / VM; [26]
 - implementuoja platforminį UEFI interfeisą:
 - palaikomi tik su UEFI suderinami *Guest OS*; [28]
 - nepalaiko tiesioginės, neemuliuotos *Legacy BIOS* / Boot-ROM kelties; [24] psl. 12
 - todėl kai kuriems OS tenka turėti/sukurti po skirtingą atvaizdą kiekvienam hipervizoriui;
 - modernus kodas, mažas *Overhead*
 - didesnis našumas (angl. *Performance*);
 - stipresnis pajėgumų dauginamumas (angl. *Scalability*);
 - integracija su ZFS :
 - akimirksniniai duomenų (FSų) klonai
 - duomenų šifravimas (jei reikia)
 - ir jų deduplikacija (jei reikia)
 - palaiko *labai* įvairius Storage backend-us
 - *PCI pass-through* galimybė; [13] psl. 10
 - integruotas *Net-antispoofing* mechanizmas; [9] psl. 11
 - su *illumOS* išeities kodu ("CDDL") tiesiogiai suderinama licencija ("Simplified BSD License") leidžia integruoti *bhyve* kodą tiesiogiai į branduolį;
 - Dabar SmartOS sistemoje *bhyve* naudojamas pagal nutylėjimą.

Kadangi apie populiarų KVM tikėtinais rašys kiti grupiokai (o be to, SmartOS naudoja senoką KVM versiją), toliau akcentuosiu SmartOS + *bhyve* VMM kombinaciją.

Ką SmartOS siūlo ?

Pats SmartOS hostas yra specializuota OS — skirta tik Guest OS startavimui / valdymui, ir niekam kitam.

SmartOS komponentai [13]

(Bent jau FreeBSD atveju pavadinimai tikslūs, SmartOS atveju galimi smulkūs pakitimai)

- *vmm.ko* — branduolio draiveris (tvarkyklė);
- *bhyve* — VMų / Guest OS apdorojimo programa, pats VMM (*Userland*);

- `bhyveload` ir `grub2-bhyve` — Guest OS įkrovikliai;
- `bhyectl` — VM egzempliorių (angl. *instances*) valdiklis, pvz. jų išjungimui.

Architektūros diagramos

- Virtualaus ūkio iš šešių Guestų pavyzdys — 2 Linux VM + 4 SmartOS konteineriai: [38]
 - (Linux) Elk
 - (Linux) OCS Inventory NG
 - DHCP
 - Samba
 - powerDNS
 - MySQL PerconaDB

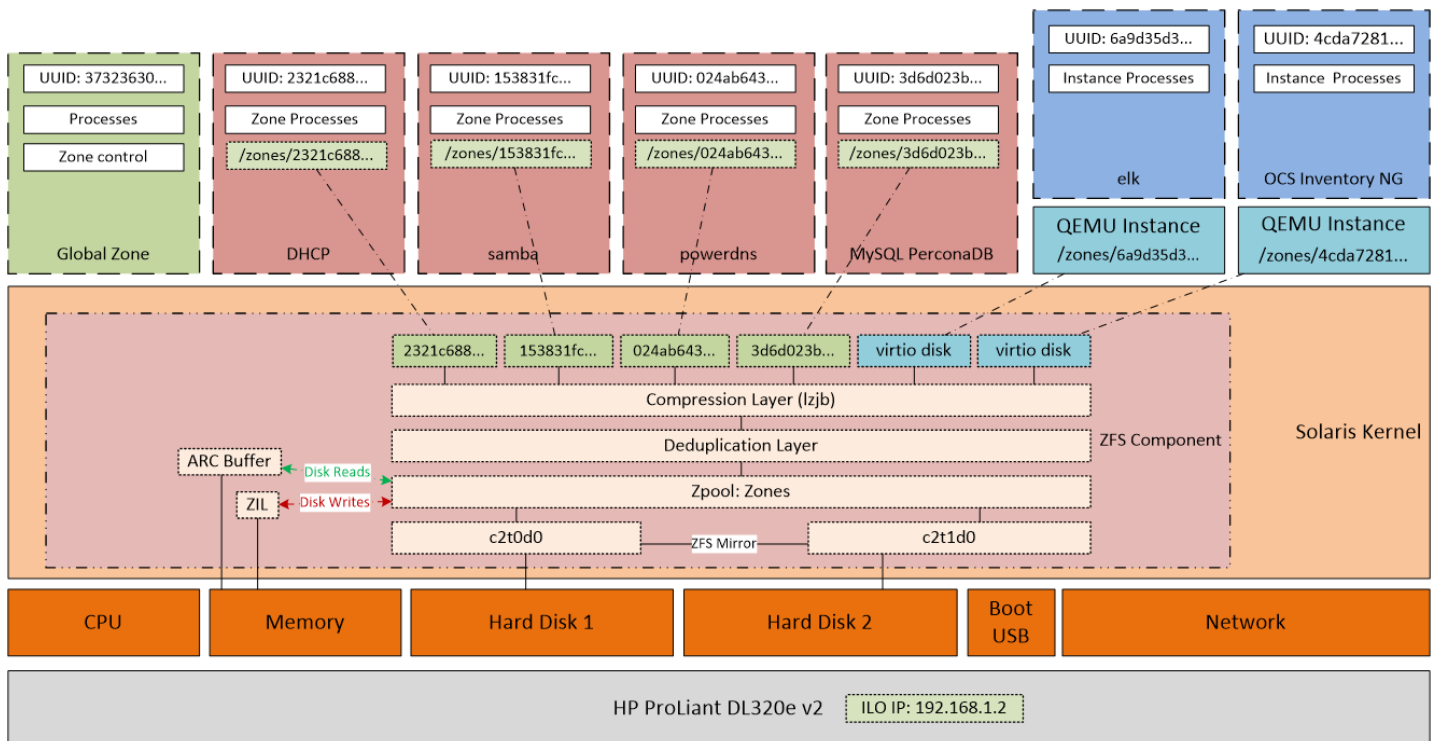


Figure 1. Hipervizoriaus pjūvis per Storage

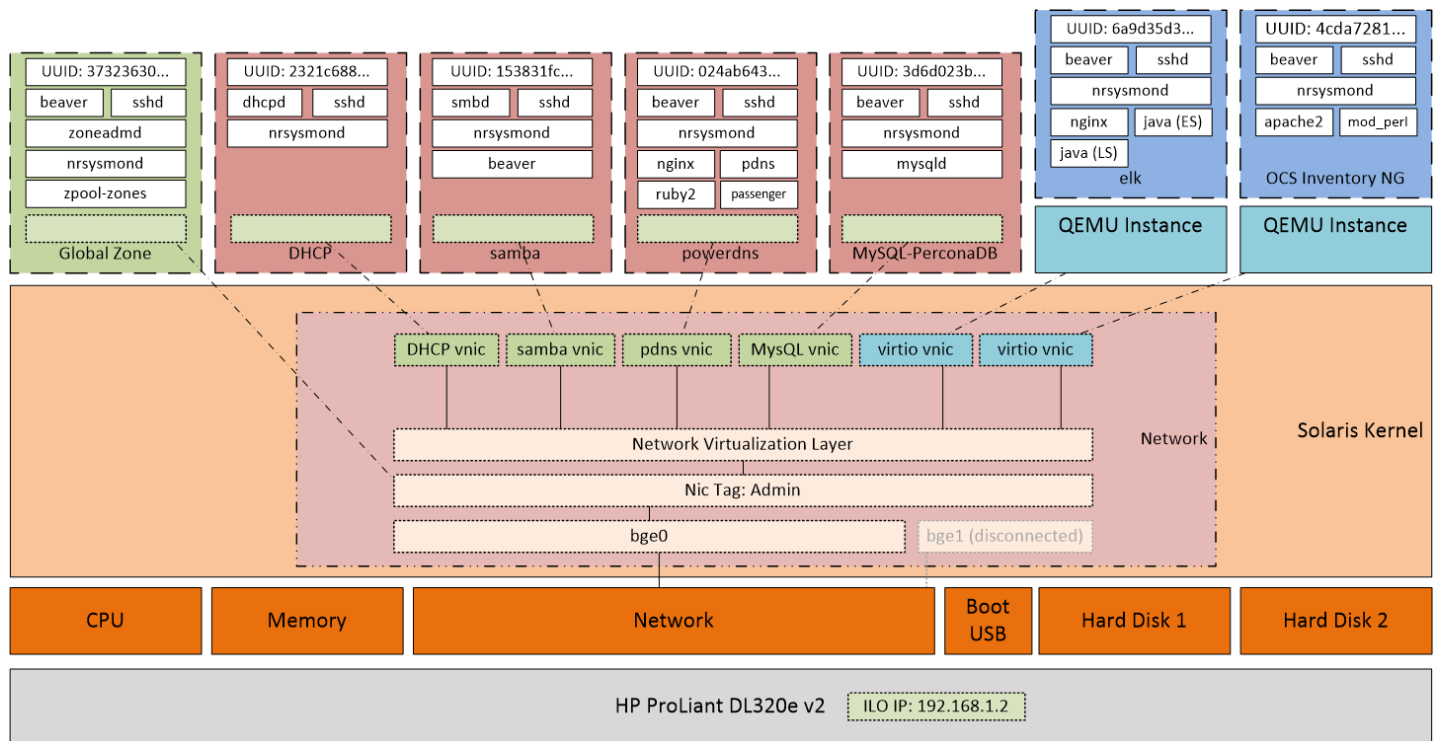


Figure 2. Hipervizoriaus pjūvis per Network

- Ankstesnės FreeBSD prezentacijos (komponentų diagramos). [12]
- Diagramos iš dokumentacijos: [39]

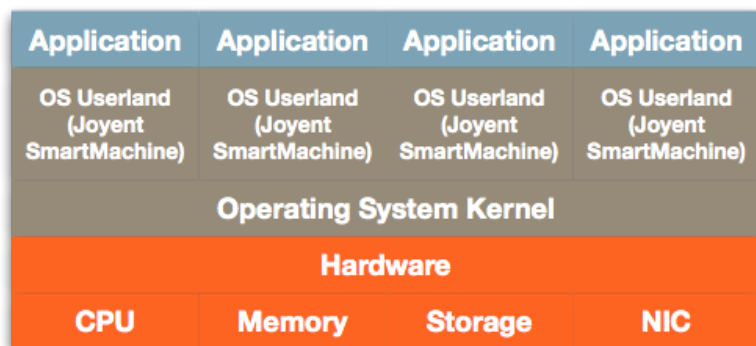


Figure 3. OS lygio virtualizacija (specializuoti SmartOS konteineriai)

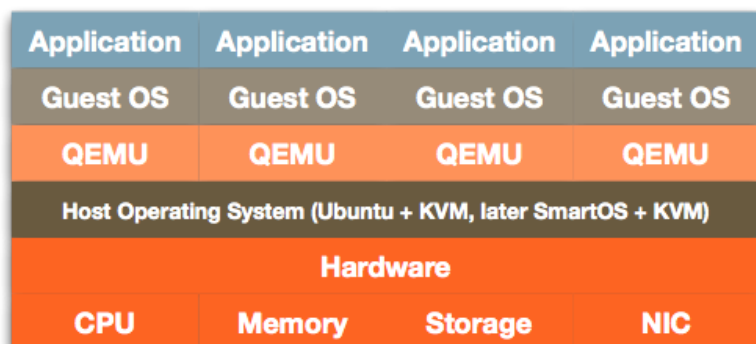


Figure 4. HVM (Type-1.5) virtualizacija (KVM + bhyve)

Savybės:

1. Užima mažai vietos (kelis šimtus MiB):

- ⇒ sparti keltis (įkrova), ~30s
- ⇒ sumažintas atakų perimetras
- Nodų konfigūravimui rekomenduojama naudoti *Chef* (gamintojo rekomenduojamiausia), *Ansible* ar kt. *Configuration Management* sistemą. [3]

2. Skirta kelčiai iš R/O atvaizdo (*Live image*) ar bent jau ne iš kietųjų diskų:

- per PXE (LAN tinklu)
- iš USB Flash Disk (UFD)
- iš CD-ROM
- *Host OS* nenaudoja lokalių diskų
 - ⇒ padidintas atsparumas sisteminio disko gedimams
 - ⇒ supaprastėja „diegimas“, ypač turint daug mašinų (tėra vos keli nustatymai) [4]
 - ⇒ spartus atnaujinimas (pakanka tinklu nusikopijuoti bendrą OS atvaizdą + *Reboot*)
 - kaskart per PXE
 - vieną sykį į UFD per SSH
 - ⇒ daugiau vietos diskuose lieka *Guest OS* egzemplioriams
- 2020-09 įgyventinta keltis ir iš diskų (iš *ZFS pool* grupės); [35]
Toks įdiegimas įmanomas ir be fizinės prieigos prie serverio (iš Linux OS) [36]

3. *in-memory / Live OS*:

- *root* failų sistema (FS) laikoma RAME
- FS pakeitimai arba efemeriški (*/etc*), arba neįmanomi (R/O) [5]
(išskyrus keletą svarbių direktorijų ir failų, atvirų rašymui; gal pvz.: ?)
 - ⇒ neveikia naujų vartotojų kūrimas
 - ⇒ primontuota **R/O**: */usr* (pagrindinė OS failų vieta)
 - ⇒ primontuota R/W: */usbkey* (OS įdiegimo duomenys)
 - ⇒ primontuota R/W: */zones* (Guest OS duomenys)
 - ⇒ primontuota R/W: */var* (logai ir būsenos failai)
 - ⇒ primontuota R/W: */opt* (saugo esminius OS pakeitimus + naujus paketus ir jų konfigūracijas)
 - ⇒ dar labiau sumažintas atakų perimetras

4. Visi serverio lokalūs diskai apjungiami į vieną grupę (*pool*):

- apjungimas *programinis*
- apjungimo metodas RAIDZ
- panašus į RAID 5/6/7
 - ⇒ padidinta ir diskų I/O sparta, ir patikimumas
- tinkliniai diskai nenaudojami

5. Siūlo iškart du virtualizacijos lygius:

- OS lygis, labai efektyvus — konteineriai: tiesiog *zonos*, dar vadinamos OS VM.

Šiuo atveju Guest OS bus tokio paties tipo, kaip ir Host OS.

Tačiau *illumOS* turi "LX" mechanizmą, kuris transliuoja Linux *syscalls* į *illumOS syscalls*.

Jis konteineriuose įgalina ir Linux branduolio interfeisą. [37]

Tad gamintojas siūlo tokius Guest OS: [30]

- Linux distribucijų atvaizdai
 - ⇒ CentOS 7: 20180323
 - ⇒ CentOS 6: 20170407
 - ⇒ Alpine 3: 20170303
 - ⇒ Debian 9: 20180404
 - ⇒ Debian 8: 20170214
 - ⇒ Debian 7: 20161213
 - ⇒ Ubuntu 16.04: 20170403
 - ⇒ Ubuntu 14.04: 20170403
- SmartOS Guest atvaizdai (*developer-friendly*).
 - ⇒ Base (švari aplinka, DIY)
 - ⇒ Apache
 - ⇒ Brocade Virtual Traffic Manager (vTM)
 - ⇒ Cassandra
 - ⇒ Ghost
 - ⇒ Java
 - ⇒ Minimal
 - ⇒ MongoDB
 - ⇒ MySQL Cluster
 - ⇒ Nginx
 - ⇒ Node.js
 - ⇒ Pkgbuild
 - ⇒ Percona
 - ⇒ Percona Cluster
 - ⇒ PostgreSQL
 - ⇒ Redis
 - ⇒ Standard (įvairių *Web-development* įrankių pakuotė)
- „geležies“ lygis, labai izoliuotas — virtualios mašinos: tiesiog VM, dar vadinamos HVM. Palaikomos Guest OS: [31] (beveik visos populiariausios)
 - ⇒ Windows *desktop* ir Windows Server versijos;
 - ⇒ Linux distribucijos;
 - ⇒ *BSD distribucijos;
 - ⇒ illumOS distribucijos (SmartOS, OmniOS);
 - ⇒ Plan9

6. Naudoja ir dubliuotą virtualizaciją:

- Guest OSą palaikantis VMM procesas (tiek `qemu-kvm`, tiek `bhyve`) veikia tik konteineryje; (*Double hulled virtualization* patentas [6])
 - ⇒ dar labiau sumažintas atakos perimetras; [9] psl. 9
 - ⇒ preciziškas Guest OSų valdymas — konteineris pilnai užtikrina QoS, resursų valdymą, I/O ribojimą (angl. *throttling*), apskaitą, kitą instrumentuotę; [21] psl. 22
 - ⇒ apjungtas konteinerių bei VMų valdymas (komanda `vmadm`)
- Abu hipervizoriai, `KVM` ir `bhyve` geba veikti kartu — vienu metu viename hoste aptarnauti savo atskirus VMus; [24] psl. 8

7. Host OS turi po atskirą įrankį:

- `piadm` valdyti Host OS ZFS atvaizdams (*Platform Image*)
- `imgadm` valdyti Guest OS atvaizdams
- `vmadm` valdyti Guest OS egzemplioriams
 - Guest aprašymui naudojamas JSON formatas;
- `dladm` valdyti OS tinklo interfeisams (L2, įskaitant jų virtualizavimą)
- `ipadm` valdyti OS potinkliams (L3)
- `fwadm` valdyti OS ugniasienei (L2 - L4)

8. Naudoja tarpplatforminius paketų valdiklius `pkgsrc` ir `pkgin` (kilusius NetBSD sistemoje).

Pirmos keturios savybės išplaukia iš **griežtai lokali** Host nodų **talpyklinės architektūros**. [23]

Tai reiškia, kad kiekviename node VMai saugomi tik lokaliuose diskuose ir startuoja ne iš NAS ar SAN tinklo.

Tokia architektūra lemia neitin tipinį hipervizoriaus panaudojimo scenarijų — mažiau kompleksišką ir labiau decentralizuotą nei pvz. rinkos lyderis VMware *ESXi*:

- atkrinta *Storage* tinklo įnešamas vėlinimas;
- išauga nodų I/O nepriklausomumas;
- talpykliniai diskai keičiami kiekvienam Hostui atskirai;
- *High-availability* (HA) / *Fault tolerance* (FT) tenka projektuoti *Application* lygmenyje; [27], [32]
- *Live Migration* sunkiau įgyvendinamas;
- *Live Migration* bus lėtesnis.

(Duomenų suvienodinimas tarp lokalių talpyklų „suvalgys“ dalį LAN pralaidumo).

Kaip paminėta 6-oje savybėje, SmartOS hipervizorius Guest OSą talpina į VMą, o ją aptarnaujantį *userland* procesą į OS konteinerį.

Tai įgalino konverguoti Guest valdymą — ir HVMus, ir OS VMus (konteinerius) valdyti viena komanda. Bet to, padeda saugotis nuo *Host-to-Guest escape* atakų.

Kokius DC/IT uždavinius SmartOS sprendžia ?

SmartOS sprendžia ir įprastus virtualizavimo uždavinius, ir keletą naujų:

- Kadangi SmartOS HVM virtualizaciją kombinuoja su savo efektyvia konteinerių architektūra (angl. *Cloud-native*), tai bent anksčiau, apie 2013 m. ji leisdavo SmartOS pagrindu veikiantiems „debesims“ drąsiai **atlaikyti staigų apkrovų šuolį per kelias dešimtis tūkstančių KAV** (kasdienių aktyvių vartotojų) tiek pagal CPU, tiek pagal RAM resursus.

Kai tuo metu kitos debesinės architektūros buvo priverstos naudoti tiesiog 50% *Overprovisioning*.

Scenarijus aktualus aptarnaujant pvz. mobiliųjų žaidimų ar socialinio tinklo žaidimų projektus. [33]

- SmartOS palaiko **I/O pralaidumo paskirstymą tarp VMų** ir (šuolių atveju) gelbsti jau paruošto *Overprovisioning* sumažinimui, VM tankio bei Host I/O išnaudojamumo padidinimui. [34]

Įprasti virtualizavimo uždaviniai

- ar VM migruojami tarp hostų klasteryje?

Taip: tam reikalingas CLI komandų vykdymas ir tinklo (LAN) pralaidumas.

- migracija be Downtime ar su?

Su trupu prastova: dar vadinama *Warm migration*. Tai visai dera su konteinerizacija ir *Microservices* architektūra.

- ar yra VMų snapšotai?

Egzistuoja tik VM diskinės talpos snapšotai.

- ar yra globalus resursų ribojimas (pvz. užtikrinant DR rezervą)

Mano rasta duomenimis, globalaus resursų ribojimo *SmartOS* neturi.

- ar veikia Oversubscription?

Dėl puikios OS architektūros lygiagretingumo *Oversubscription* veikia puikiai ir leidžia atlaikyti itin aukštus apkrovų šuolius (tinklo, CPU, I/O).

- kaip skeiliasi?

Vertikalia ir horizontalia ašimi skeiliasi puikiai ir tinka *Microservices* strategijai: lengva kurti atskirus Guestus atskiriems servisams (ypač kadangi palaikomas Docker), taip pat itin paprasta klonuoti esamus Guestus.

- kaip atliekamas DR?

DR atliekamas rankiniu būdu: [40]

- identifikuojame GRUB (angl. *Bootloader*) sritį ir *ZFS Boot pool* diskinę erdvę;
- padarome atsarginę jų kopiją;
- fizinės nelaimės atveju atstatome šiuos duomenis + aplikacijos duomenis iš atsarginių kopijų naujame serveryje.
- startuojame naują serverį (Host OS).

Ar SmartOS valdomas tiesiogiai, ar klasteriniu būdu ?

Kaip pavienis įrankis, SmartOS valdomas tiesiogiai.

Tačiau daugianodžių SmartOS debesų valdymui siūlomas atskiras, irgi atvirojo kodo įrankis (angl. *Cloud management platform*) **Triton DataCenter** (TDC) / **Triton Compute Service** (TCS), į kurio funkcijas irgi trumpai atsižvelgsiu. [7]

Jei norime kelti OSus tinklu, šiame įrankyje verta PXE mechanizmui dedikuoti atskirą hostą, vad. Head Node (HN). Kiti pakeltieji hostai jau vadinsis Compute Node (CN).

Instances

[+ Create Instance](#)

Showing 1 - 4 of 4 entries

Data Center: US-SW-1

[actions](#)[export.csv](#)[columns](#)

<input type="checkbox"/>	Name	IP	Image	Firewall	Created	Status
<input type="checkbox"/>	server-1	64.30.129.97	ubuntu-14.04/20170403	Disabled	2018-02-13 11:05 (EST)	✓ running
<input type="checkbox"/>	server2	165.225.159.21	ubuntu-14.04/20170403	Disabled	2018-01-09 17:57 (EST)	✓ running
<input type="checkbox"/>	1009d417	165.225.159.151	standard/15.1.1	Disabled	2018-01-07 20:30 (EST)	✓ running
<input type="checkbox"/>	3a1f3346	165.225.156.33	base-64/17.3.0	Disabled	2018-01-04 19:54 (EST)	✓ running

Figure 5. TDC Guest OS valdymo GUI

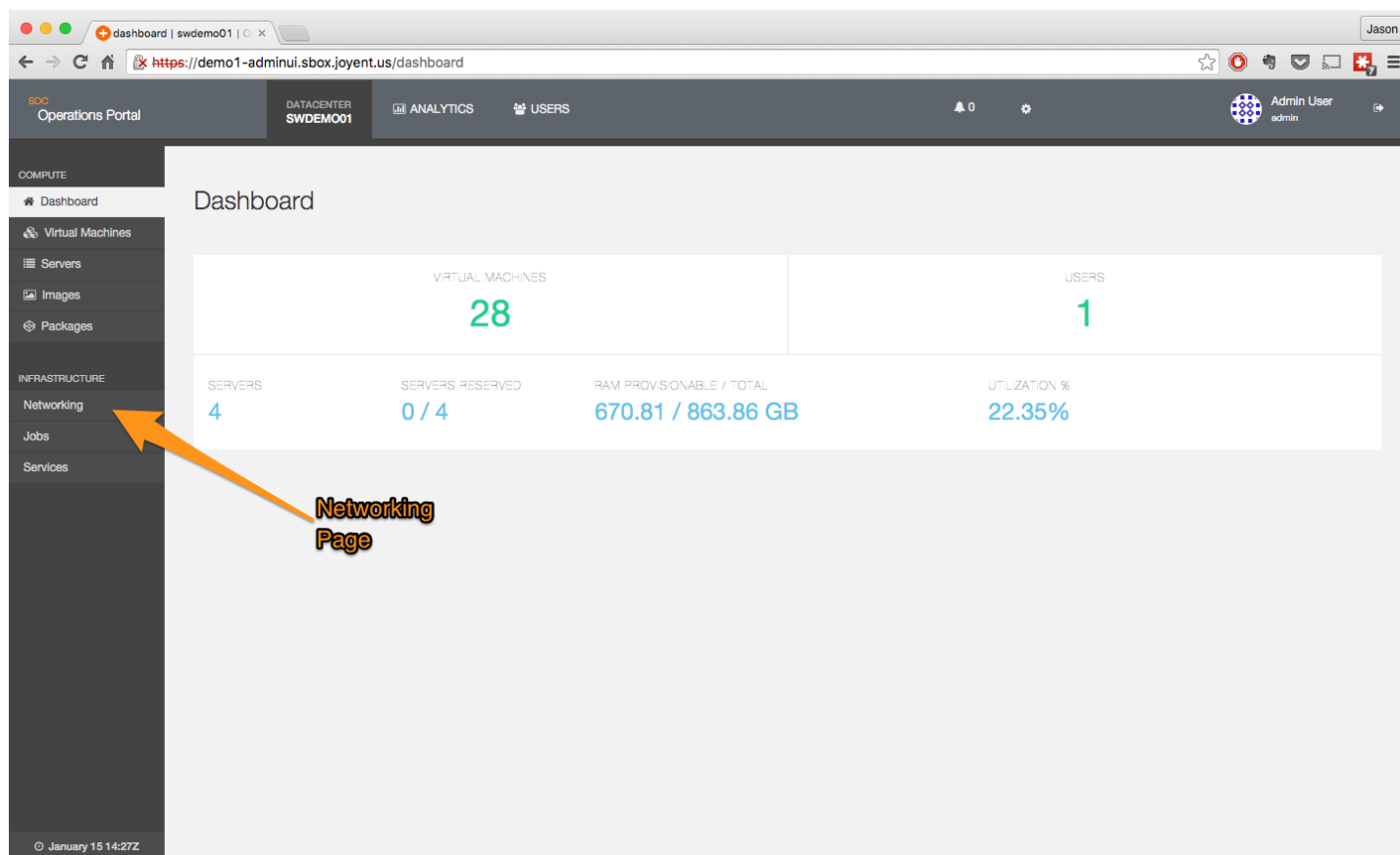


Figure 6. Vieno TDC duomenų centro hostų apžvalga

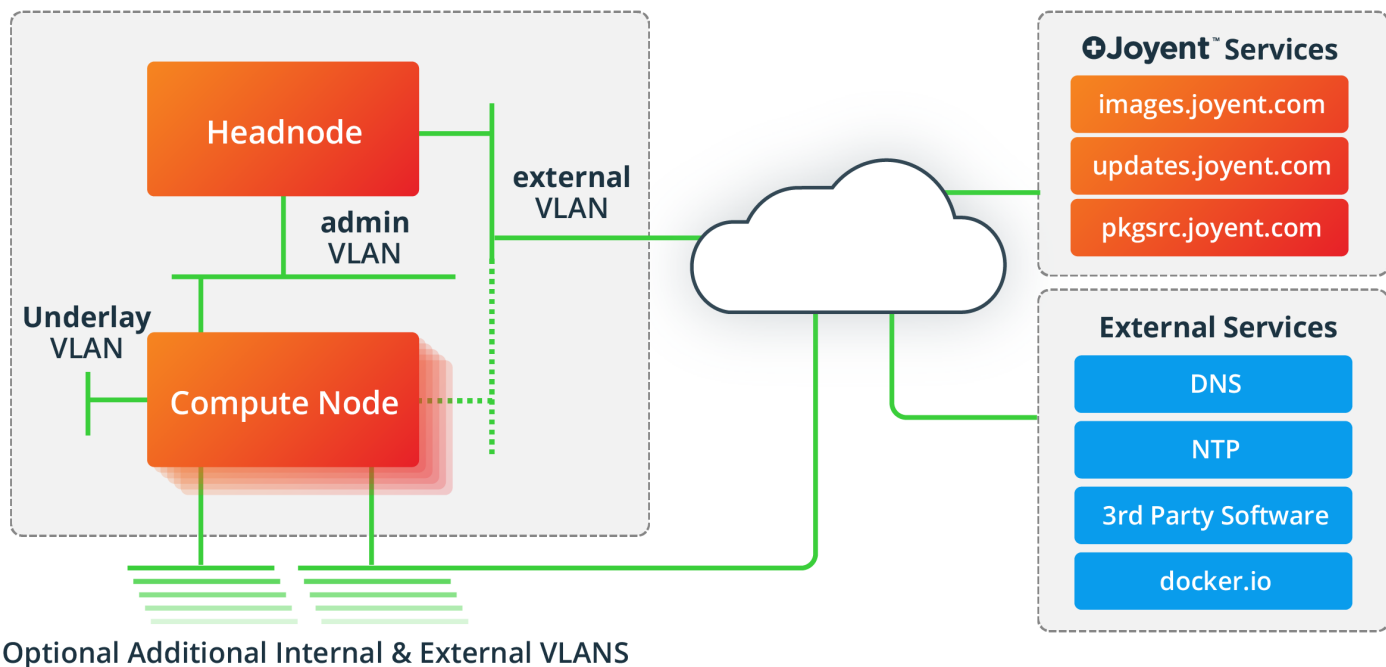


Figure 7. TDC HN (headnode) ir CN (compute node) tinklai duomenų centre ir sujungimai su išore

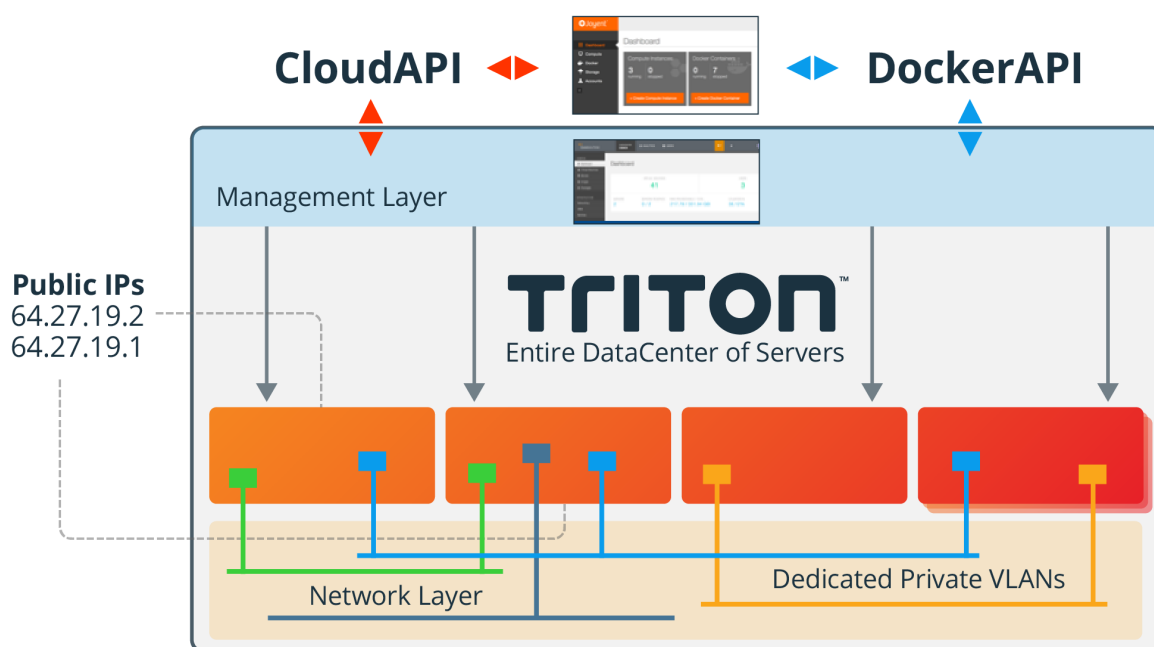


Figure 8. Virtualios TDC infrastruktūros valdymas REST API pagalba

TDC įgalina centralizuotai:

- atlikti *Firewalling* (OS vidinio mechanizmo IPFilter dėka): <https://docs.joyent.com/public-cloud/network/firewall>
- valdyti tinklus L2 (Fabric, VLAN, VNIC, per-container TCP/IP stack) ir L3 (IP subnets, VXLAN, antispoofing, routing, NAT) lygmenyse programiškai (SDN): <https://docs.joyent.com/public-cloud/network/sdn>

Daugiau apie Triton DataCenter:

- Konteinerių ir VMų Combo diagramos: [14]
 - Detalesnė Triton DC sudėtis: [15]
-

SmartOS pranašumai:

- *Solaris / illumOS* projektuotas didesniai saugumui (apskritai TODO patikimumui) nei Linux, ir naudojantis tai juntama tiesiogiai
- palaikymo *VirtIO* — efektyvų ir vieną populiariausių paravirtualizacijos (PV) interfeisų; [13] psl. 10
- *cloud-init* standarto palaikymas: [19], [20]
 - Įgalina debesų (egzempliorių) inicializavimą nepriklausomai nuo platformos, pvz.:
 - OS vartotojų paskyrų sukūrimą
 - programinių paketų sudiegimą
 - Git repozitorijų nuklonavimą
 - apskritai kone bet kuriuos OS administravimo veiksmus.
 - Naudoja YAML sintaksę
 - tenka ją suderinti su SmartOS `vmadm` naudojama JSON sintakse.
 - Palaikomas:
 - tiek visų didžiųjų viešos debesijos tiekėjų,
 - tiek atliekant OS provizijavimą ir privačiuose debesyse,
 - ir „plikoje geležy“ (angl. *Bare metal*).

SmartOS trūkumai:

- Kiek vėlokai žengė į rinką (2011-2013 m.), todėl kol kas užima mažoką jos dalį; [10] 13m 41s
- Kol kas palaiko tik *x86* architektūrą (*no ARM*); [10] 15m 7s
- Kol kas neveikia VM *Live Migration* (dar tik kuriama).
Veikia tik VM *Warm / Cold Migration*; [11]
- Guest OS VGA išvestis ribota, pasiekama tik VNC protokolu;
- Hostas valdomas per CLI, JSON ir truputį YAML (sąlyginis trūkumas).
Norint GUI, reiktų naudoti papildomą įrankį:
 - pilnavertę debesijos valdymo platformą *Triton DataCenter* arba
 - debesies valdiklį ir orkestratorių *Project FiFo* (kai mažesnis ūkis ir vengiama dedikuoto HNo, *Head-node*).
- Rinkodaros strategija dar tik kuriama, ji kinta;
(todėl stipresnės kitų hipervizorių adminų ir jūzerių bendruomenės)

Ar SmartOS yra nišinis hipervizorius ?

Pagal rinkos dydį ir kai kuriuos trūkumus SmartOS gal ir tikėtų vadinti nišiniu sprendimu.

Tačiau pagal siūlomą architektūrą SmartOS vis dėlto yra inovatyvus, genialiai paprastas, efektyvus ir nemokamas atvirojo kodo sprendimas.

Ar tai hipervizorių daro labiau nišiniu, ar mažinau nišiniu — požiūrio ir susitarimo klausimas.

Taikymo pavyzdžiai:

- Docker konteinerių startavimas be *Triton* pagalbos. [16] (t. p. ir `fwadm` aprašymas)
- Asmeninio Docker registro naudojimas. [17]
- Orientavimasis į *Node.js* servisus. [18]
- "Deploying Kubernetes on SmartOS | Virtualization: How SmartOS Does it Differently". [41]
- "Cloud-on-a-Laptop" [42]
- "Thoughts on Kubernetes on SmartOS" [43]

Nuorodos

2020-11-17 **[1]** serverwatch.com , Christine Taylor, [What Is a Hypervisor Server?](#)

(<https://www.serverwatch.com/virtualization/hypervisor-server/#:~:text=Linux%20KVM%20and%20FreeBSD%20bhyve>)

2016-06-01 **[2]** marksei.com , Marksei, [What is a Virtual Machine? | Virtual Machine 102 – Hypervisor types](#)

(<https://www.marksei.com/what-is-virtual-machine/#:~:text=called%20Type%2D3%20or%20Type%2D1.5>)

2021-12-15 **[3]** wiki.smartos.org , (peržiūrėta) [Configuration Management on SmartOS](#)

(<https://wiki.smartos.org/configuration-management-on-smartos/>)

2012-04-13 **[4]** perkin.org.uk , Jonathan Perkin, [SmartOS global zone tweaks](#)

(<https://www.perkin.org.uk/posts/smartos-global-zone-tweaks.html>)

2012-11-23 **[5]** perkin.org.uk , Jonathan Perkin, [SmartOS and the global zone](#)

(<https://www.perkin.org.uk/posts/smartos-and-the-global-zone.html#:~:text=on%20running%20SmartOS.,So%20what%20can%20I%20do%3F,-Firstly%2C%20let%E2%80%99s%20look>)

2021-07-08 **[6]** joyent.com , Michael Zeller, [Reintroducing Bhyve](#)

(<https://www.joyent.com/blog/reintroducing-bhyve#:~:text=This%20is%20what%20we%20mean%20when%20we%20say%20double%2Dhulled%2Dvirtualization>)

2021-01-05 **[7]** docs.joyent.com , (redaguota) [Triton Operator Documentation](#) (<https://docs.joyent.com/private-cloud>)

2020-09-03 **[8]** klarasystems.com , Allan Jude, [bhyve | The FreeBSD Hypervisor](#)

(<https://klarasystems.com/articles/bhyve-the-freebsd-hypervisor/>)

2018-03-05 **[9]** bhyvecon.org , Mike Gerdts, [bhyve zones in SmartOS](#) (<https://bhyvecon.org/bhyvecon2018-Gwydir.pdf>)

2021-01-27 **[10]** youtube.com , Yaroslav Koisa, [FreeBSD's Bhyve Overview: Why it's better than other](#)

[hypervisors. At least for our use-case.](#) (<https://www.youtube.com/watch?v=uV61mVYsFM8&t=15m7s>)

2021-07-01 **[11]** docs.google.com , Alan Jude, [bhyve Weekly Call](#)

(<https://docs.google.com/document/d/1PFUmz6XpTVAGkq5dBe8uaBFV2Y4i-uR88AuiCLIRxIQ/>)

2011-05-13 **[12]** people.freebsd.org , Neel Natu | Peter Grehan, [BHyVe | BSD Hypervisor](#)

(https://people.freebsd.org/~neel/bhyve/bhyve_bsdcan_2011.pdf)

2014-05-07 **[13]** papers.freebsd.org , John Baldwin, [Introduction to bhyve](#)

(https://papers.freebsd.org/2014/baldwin-Introduction_to_bhyve.files/slides.pdf#page=6)

2021-01-05 **[14]** docs.joyent.com , (redaguota) [Triton End User Documentation > Containers and virtual machines >](#) (<https://docs.joyent.com/public-cloud/instances>)

2021-07-14 **[15]** github.com/joyent/triton , [Triton DataCenter | README](#)

(<https://github.com/joyent/triton/blob/master/README.md#overview>)

2021-06-11 **[16]** gaige.net , Gaige B. Paulsen, [Docker on SmartOS](#) (<https://www.gaige.net/docker-on-smartos.html>)

2018-02-11 **[17]** cyber-tec.org , Thomas Merkel, [Run Docker images on SmartOS](#)

(<https://www.cyber-tec.org/2018/02/11/run-docker-images-on-smartos/>)

2017-01-12 **[18]** joyent.com , Wyatt Preul, [Containers and microservices and Node.js! Oh, my!](#)

(<https://www.joyent.com/blog/microservices-containers-nodejs>)

2019-09-04 **[19]** [readthedocs.io](https://cloudinit.readthedocs.io/en/latest/topics/datasources/smartos.html) , (redaguota) [cloud-init » Docs » Datasources » SmartOS Datasource](#)
(<https://cloudinit.readthedocs.io/en/latest/topics/datasources/smartos.html>)

2019-01-16 **[20]** [shaner.life](https://shaner.life/using-cloud-init-with-smartos/) , Shaner, [Using cloud-init with SmartOS](#)
(<https://shaner.life/using-cloud-init-with-smartos/>)

2011-08-15 **[21]** [slideshare.net](https://www.slideshare.net/bcantrill/experiences-porting-kvm-to-smartos/22) , Bryan Cantrill, [Experiences porting KVM to SmartOS](#)
(<https://www.slideshare.net/bcantrill/experiences-porting-kvm-to-smartos/22>)

2012-06-29 **[22]** [github.com/joyent/illumos-kvm](https://github.com/joyent/illumos-kvm/blob/master/README.md#illumos-kvm-kvm-for-illumos) , (redaguota) [illumos-kvm: KVM for illumos](#)
(<https://github.com/joyent/illumos-kvm/blob/master/README.md#illumos-kvm-kvm-for-illumos>)

2014-06-06 **[23]** [wikipedia.org](https://en.wikipedia.org/wiki/SmartOS#:~:text=SmartOS%20follows%20a%20strict%20local%20node%20storage%20architecture) , (redaguota) Trentstersla, [SmartOS | "SmartOS follows a strict local node storage architecture"](#)
(<https://en.wikipedia.org/wiki/SmartOS#:~:text=SmartOS%20follows%20a%20strict%20local%20node%20storage%20architecture>)

2019-05-14 **[24]** [bhyvecon.org](https://bhyvecon.org/bhyveconOttawa2019-Patrick.pdf) , Patrick Mooney, [Porting bhyve to SmartOS](#)
(<https://bhyvecon.org/bhyveconOttawa2019-Patrick.pdf>)

2011-09-21 **[25]** [lwn.net](https://lwn.net/Articles/459754/) , Koen Vervloesem, [SmartOS: virtualization with ZFS and KVM](#)
(<https://lwn.net/Articles/459754/>)

2019-05-14 **[26]** [bhyvecon.org](https://bhyvecon.org/bhyveconOttawa2019-Rodney.pdf#page=3) , Rod Grimes, [bhyve VM_MAXCPU cleanup | VM_MAXCPU](#)
(<https://bhyvecon.org/bhyveconOttawa2019-Rodney.pdf#page=3>)

2015-12-08 **[27]** [lists.smartos.org](https://www.mail-archive.com/smartos-discuss@lists.smartos.org.email.enqueue.archive.listbox.com/msg01707.html) , Bryan Horstmann-Allen, [Re: \[smartos-discuss\] High availability solutions with SmartOS](#)
(<https://www.mail-archive.com/smartos-discuss@lists.smartos.org.email.enqueue.archive.listbox.com/msg01707.html>)

2018-12-26 **[28]** [gist.github.com](https://gist.github.com/mgerdts/10376cdbc8f015f422d61664408db2aa#file-1-guest-images-md) , Mike Gerdts, [Bhyve Machine Images | UEFI and BIOS Emulation](#)
(<https://gist.github.com/mgerdts/10376cdbc8f015f422d61664408db2aa#file-1-guest-images-md>)

2010-08-09 **[29]** [github.com/joyent/illumos-kvm](https://github.com/joyent/illumos-kvm/blob/master/COPYING.linux) , [The KVM Project is derived from the Linux kernel](#)
(<https://github.com/joyent/illumos-kvm/blob/master/COPYING.linux>)

2021-04-28 **[30]** [docs.joyent.com](https://docs.joyent.com/public-cloud/instances/infrastructure/images#:~:text=Container%20images%3A%20Table%20of%20Contents) , (redaguota) [Container images: Table of Contents](#)
(<https://docs.joyent.com/public-cloud/instances/infrastructure/images#:~:text=Container%20images%3A%20Table%20of%20Contents>)

2021-12-10 **[31]** [wiki.freebsd.org](https://wiki.freebsd.org/bhyve#Q:_What_VM_operating_systems_does_bhyve_support?) , Christos Margiolis, [bhyve | Q: What VM operating systems does bhyve support?](#) (https://wiki.freebsd.org/bhyve#Q:_What_VM_operating_systems_does_bhyve_support.3F)

2010-08-13 **[32]** [hoffmanmarcom.com](https://www.hoffmanmarcom.com/dev/wp-content/docs/Cloud%20Architecture%20-%20Strategies%20and%20Configurations%20Guidebook%20Joyent.pdf#page=10) , (publikuota) [Cloud Based Application Architectures using Smart Computing | Tier 3: Web HA and database master/slave](#)
(<https://www.hoffmanmarcom.com/dev/wp-content/docs/Cloud%20Architecture%20-%20Strategies%20and%20Configurations%20Guidebook%20Joyent.pdf#page=10>)

2013-01-30 **[33]** [joyent.com](https://www.joyent.com/blog/joyent-takes-gaming-companies-to-the-next-level#:~:text=dramatically%20reducing%20the%20cost%20per%20DAU) , Rachel Balik, [Joyent Takes Gaming Companies to the Next Level](#)
(<https://www.joyent.com/blog/joyent-takes-gaming-companies-to-the-next-level#:~:text=dramatically%20reducing%20the%20cost%20per%20DAU>)

2012-10-11 **[34]** [joyent.com](https://www.joyent.com/blog/nodestack-is-nodejs-mongodb-and-smartos#:~:text=burstable%20IO%20sharing%20across%20VMs%20for%20less%20overprovisioning) , Ben Wen, VP at MongoLab, [Nodestack is Node.js, MongoDB, and SmartOS](#)
(<https://www.joyent.com/blog/nodestack-is-nodejs-mongodb-and-smartos#:~:text=burstable%20IO%20sharing%20across%20VMs%20for%20less%20overprovisioning>)

2020-09-01 **[35]** kebe.com , Daniel L. McDonald, [Now you can boot SmartOS off of a ZFS pool](https://kebe.com/blog/?p=504)
(<https://kebe.com/blog/?p=504>)

2021-11-16 **[36]** [hetzner.com](https://community.hetzner.com/tutorials/how-to-install-SmartOS) , Stefan Eestermans, [How to install SmartOS without the need for a USB flash drive](https://community.hetzner.com/tutorials/how-to-install-SmartOS) (<https://community.hetzner.com/tutorials/how-to-install-SmartOS>)

2015-05-26 **[37]** [brianewell.com](https://blog.brianewell.com/branded-lx-zones-on-smartos/) , Brian Ewell, [Branded \(LX\) Zones on SmartOS](https://blog.brianewell.com/branded-lx-zones-on-smartos/)
(<https://blog.brianewell.com/branded-lx-zones-on-smartos/>)

2014-07-27 **[38]** [svbtle.com](https://marclop.svbtle.com/smartos#SmartOS-Network#:~:text=42:41:58:2d:fa:8c) , Marc Lopez, [SmartOS overview](https://marclop.svbtle.com/smartos#SmartOS-Network#:~:text=42:41:58:2d:fa:8c)
(<https://marclop.svbtle.com/smartos#SmartOS-Network#:~:text=42:41:58:2d:fa:8c>)

2022-02-11 **[39]** [wiki.smartos.org](https://wiki.smartos.org/smartos-virtualization/) , (peržiūrėta) [SmartOS Virtualization | Virtualization: How SmartOS Does it Differently](https://wiki.smartos.org/smartos-virtualization/) (<https://wiki.smartos.org/smartos-virtualization/>)

2019-09-24 **[40]** [blog.jcea.es](https://blog.jcea.es/posts/20190924-dumping_boot_SmartOS.html#disaster-recovery) , Jesús Cea Avión [Dumping SmartOS boot zpool when booting from harddisk](https://blog.jcea.es/posts/20190924-dumping_boot_SmartOS.html#disaster-recovery)
(https://blog.jcea.es/posts/20190924-dumping_boot_SmartOS.html#disaster-recovery)

2019-02-12 **[41]** [youtube.com](https://www.youtube.com/watch?v=rA0pcmqpRx4) , ShanerLife [Deploying Kubernetes on SmartOS | Virtualization: How SmartOS Does it Differently](https://www.youtube.com/watch?v=rA0pcmqpRx4) (<https://www.youtube.com/watch?v=rA0pcmqpRx4>)

2021-03-19 **[42]** [github.com](https://github.com/joyent/triton/blob/master/docs/developer-guide/coal-setup.md) , Dan McDonald [Cloud on a Laptop \(CoaL\)](https://github.com/joyent/triton/blob/master/docs/developer-guide/coal-setup.md)
(<https://github.com/joyent/triton/blob/master/docs/developer-guide/coal-setup.md>)

2020-06-07 **[43]** [hanskruse.eu](https://hanskruse.eu/post/2020-06-07-thoughts_on_kubernetes_on_smartos/) , Hans Kruse [Thoughts on Kubernetes on SmartOS](https://hanskruse.eu/post/2020-06-07-thoughts_on_kubernetes_on_smartos/)
(https://hanskruse.eu/post/2020-06-07-thoughts_on_kubernetes_on_smartos/)