

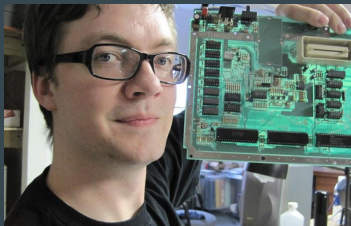
# Automating FreeBSD Jails



Using IOcage and SaltStack

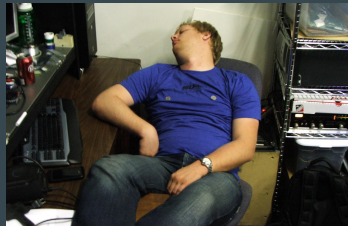
# Who?

d3c4f



- Systems Admin / DevOps
- Software Developer
- PortAParty Co-Creator
- theTransistor Founder
- OpenWest Core Team
- BlackHat Volunteer
- Caffeine Addict
- Electronic Badge Designer

Yukaia



- Systems Administrator
- Automobile Enthusiast
- PortAParty Co-Creator
- theTransistor Founder
- OpenWest Core Team
- Blackhat Volunteer
- Dr.Pepper Connoisseur
- Hardware / Server Nerd



## Contact Info>

d3c4f@sausage.land

yukaia@sausage.land



# What will we be talking about today?

- FreeBSD Jails (via iocage)
- SaltStack Basics
- Using SaltStack with FreeBSD Jails



# FreeBSD Jails - What are these things?



# FreeBSD Jails - What are these things?

FreeBSD Jails are an implementation of OS-Level Virtualization.

They serve the need to establish separation between services, mainly for security and ease of administration.



# FreeBSD Jails - What are these things?

FreeBSD Jails are an implementation of OS-Level Virtualization.

They serve the need to establish separation between services, mainly for security and ease of administration.

- ***Virtualization***
  - Each jail is a virtual environment with its own files, processes, and user accounts.
- ***Security***
  - Each jail provides controls to separate it from the host system and any other jails.
- ***Ease of Delegation***
  - Jails allow delegation of tasks that require SU access without handing over complete control of the system.



# FreeBSD Jails - What is iocage?





# FreeBSD Jails - What is iocage?

“iocage is a zero dependency drop in jail/container manager amalgamating some of the best features and technologies FreeBSD operating system has to offer. It is geared for ease of use with a simple and easy to understand command syntax.”



# FreeBSD Jails - What is iocage? (Features)

- Utilizes ZFS for Jail Datasets and Snapshot Management
- Templates, Clones, BaseJails, and Fully Independent Jails
- Virtual networking stacks (vnet)
- Shared IP based jails (non-vnet)
- Resource Limits (CPU, RAM, FileSystem Quotas, etc)
- Binary Updates
- Export and Import



# FreeBSD Jails - Manual Setup



# FreeBSD Jails - Manual Setup

1. Install FreeBSD with ZFS for Jails (We'll use ZFS on root)



# FreeBSD Jails - Manual Setup

FreeBSD Installer

---

Welcome

Welcome to FreeBSD! Would you  
like to begin an installation  
or use the live CD?

<Install> <Shell> <Live CD>



# FreeBSD Jails - Manual Setup

FreeBSD Installer

Partitioning

How would you like to partition your disk?

Auto (UFS)	Guided Disk Setup
Manual	Manual Disk Setup (experts)
Shell	Open a shell and partition by hand
Auto (ZFS)	Guided Root-on-ZFS

< **OK** >      <Cancel>



# FreeBSD Jails - Manual Setup

```
root@ubug:~ #
```



# FreeBSD Jails - Manual Setup

1. Install FreeBSD with ZFS for Jails (We'll use ZFS on root)
2. Install iocage





# FreeBSD Jails - Manual Setup

```
root@ubug:~ # pkg install iocage
The package management tool is not yet installed on your system.
Do you want to fetch and install it now? [y/N]: y
Bootstrapping pkg from pkg+http://pkg.FreeBSD.org/FreeBSD:11:amd64/quarterly, please wait...
Verifying signature with trusted certificate pkg.freebsd.org.2013102301... done
Installing pkg-1.8.7_1...
Extracting pkg-1.8.7_1: 100%
Updating FreeBSD repository catalogue...
Fetching meta.txz: 100%    940 B    0.9kB/s    00:01
Fetching packagesite.txz: 100%    5 MiB    5.8MB/s    00:01
Processing entries: 100%
FreeBSD repository update completed. 25262 packages processed.
Updating database digests format: 100%
The following 1 package(s) will be affected (of 0 checked):

New packages to be INSTALLED:
    iocage: 1.7.5

Number of packages to be installed: 1

45 KiB to be downloaded.

Proceed with this action? [y/N]: █
```



# FreeBSD Jails - Manual Setup

1. Install FreeBSD with ZFS for Jails (We'll use ZFS on root)
2. Install iocage
3. Fetch a release



# FreeBSD Jails - Manual Setup

```
root@ubug:~ # iocage fetch
Setting up zpool [zroot] for iocage usage...
If you wish to change zpool, use 'iocage activate'
INFO: Creating zroot/iocage
INFO: Creating zroot/iocage/jails
INFO: Creating zroot/iocage/.defaults
INFO: Creating zroot/iocage/download
INFO: Creating zroot/iocage/releases
Supported releases are:
10.2-RELEASE
9.3-RELEASE
Please select a release [-]: █
```



# FreeBSD Jails - Manual Setup

1. Install FreeBSD with ZFS for Jails (We'll use ZFS on root)
2. Install iocage
3. Fetch a release
4. Create a Jail



# FreeBSD Jails - Manual Setup

```
root@ubug:~ # iocage create tag=sausages
```



# FreeBSD Jails - Manual Setup

```
** rlimits=off
** boot=off
** notes=none
** owner=root
** priority=99
** last_started=none
** type=jail
** hostid=9c6adf42-22a3-4c48-ad23-65d0e736beb8
** cpuset=off
** jail_zfs=off
** jail_zfs_dataset=iocage/jails/af12c54e-6f06-11e6-b549-08002774b728/root/data
** release=10.2-RELEASE
** hack88=0
** sync_target=none
** sync_tgt_zpool=none
** gitlocation=https://github.com
** compression=lz4
** quota=none
** dedup=off
** reservation=none
root@ubug:~ # iocage list
JID  UUID                                BOOT  STATE  TAG
P
-    af12c54e-6f06-11e6-b549-08002774b728  off   down   sausages  none
root@ubug:~ #
```



# FreeBSD Jails - Manual Setup

1. Install FreeBSD with ZFS for Jails (We'll use ZFS on root)
2. Install iocage
3. Fetch a release
4. Create a Jail
5. Start a Jail



# FreeBSD Jails - Manual Setup

```
root@ubug:~ # iocage start sausages
```





# FreeBSD Jails - Manual Setup

```
root@ubug:~ # iocage start sausages
* Starting af12c54e-6f06-11e6-b549-08002774b728 (sausages)
  + Started (shared IP mode) OK
  + Starting services          OK
root@ubug:~ # █
```



# FreeBSD Jails - Manual Setup

1. Install FreeBSD with ZFS for Jails (We'll use ZFS on root)
2. Install iocage
3. Fetch a release
4. Create a Jail
5. Start a Jail
6. Autostart Jails
  - a. Add "iocage\_enable=YES" to /etc/rc.conf
  - b. Run: iocage set boot=on UUID|TAG
    - i. Ex: iocage set boot=on sausages
7. Console into a Jail



# FreeBSD Jails - Manual Setup

```
root@ubug:~ # iocage console sausages
FreeBSD 11.0-RC2 (GENERIC) #0 r304729: Wed Aug 24 06:59:03 UTC 2016

Welcome to FreeBSD!

Release Notes, Errata: https://www.FreeBSD.org/releases/
Security Advisories:  https://www.FreeBSD.org/security/
FreeBSD Handbook:     https://www.FreeBSD.org/handbook/
FreeBSD FAQ:          https://www.FreeBSD.org/faq/
Questions List: https://lists.FreeBSD.org/mailman/listinfo/freebsd-questions/
FreeBSD Forums:       https://forums.FreeBSD.org/

Documents installed with the system are in the /usr/local/share/doc/freebsd/
directory, or can be installed later with:  pkg install en-freebsd-doc
For other languages, replace "en" with a language code like de or fr.

Show the version of FreeBSD installed:  freebsd-version ; uname -a
Please include that output and any error messages when posting questions.
Introduction to manual pages:  man man
FreeBSD directory layout:      man hier

Edit /etc/motd to change this login announcement.
root@af12c54e-6f06-11e6-b549-08002774b728:~ #
```



# FreeBSD Jails - Manual Setup

1. Install FreeBSD with ZFS for Jails (We'll use ZFS on root)
2. Install iocage
3. Fetch a release
4. Create a Jail
5. Start a Jail
6. Autostart Jails
  - a. Add "iocage\_enable=YES" to /etc/rc.conf
  - b. Run: iocage set boot=on UUID|TAG
    - i. Ex: iocage set boot=on sausages
7. Console into a Jail
8. Destroy a Jail



# FreeBSD Jails - Manual Setup

```
root@ubug:~ # iocage destroy sausages
```

```
WARNING: this will destroy jail af12c54e-6f06-11e6-b549-08002774b728  
Dataset: zroot/iocage/jails/af12c54e-6f06-11e6-b549-08002774b728
```

```
Are you sure ? y[N]: █
```





# SaltStack - What is SaltStack?



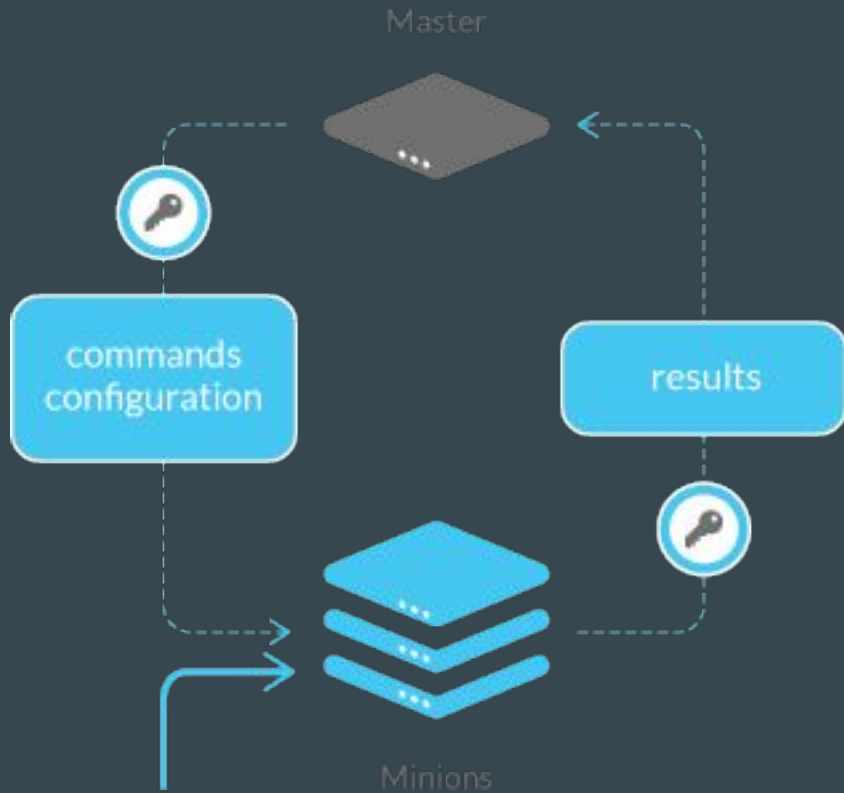
# SaltStack - What is SaltStack?

“a Python-based open-source configuration management software and remote execution engine”





# SaltStack - What is SaltStack?



# SaltStack - Who is using SaltStack?



# SaltStack - Who is using SaltStack?

- Rackspace - <https://developer.rackspace.com/blog/why-i-use-saltstack/>
- EMC - <https://youtu.be/UalzIq40xMk>
- Google Kubernetes
- Zillow
- Photobucket
- CloudFlare
- LinkedIn
- Hulu
- HP Cloud Services
- DoD
- ...



# SaltStack - Quick overview of the basic components



# SaltStack - Quick overview of the basic components

- Master
  - Used to send commands and configurations to the Minions.
  - Common to run with a single master, even in large environments.
  - Contains:
    - State Files
    - Top Files
    - Formulas
    - Pillars
    - Execution Modules
    - ... & more



# SaltStack - Quick overview of the basic components

- Master (sends commands, stores config stuffs)
- Minion
  - A system that is managed by the Master AND running the salt-minion service.
  - Receives commands and configurations from the Master.
  - Contains:
    - A running salt-minion service
    - Grains
      - Bits of information about the System that the salt-minion is running on
  - Returns information about requested command or configuration execution, or grain information.



# SaltStack - Quick overview of the basic components

- Master (sends commands, stores config stuffs)
- Minion (receives and executes commands, configurations)
- Modules
  - Small programs or scripts that are copied to the Minions from the Master
  - Distributed when “state.apply” is run, or when sync functions are executed.
  - Ran when requested by the master
  - Return information about execution in a (semi)standard format



# SaltStack - Quick overview of the basic components

- Master (sends commands, stores config stuffs)
- Minion (receives and executes commands, configurations)
- Modules (small programs, executed on the Minions)
- Formulas aka States
  - Actual written representation of a System's Configuration
  - Entry point begins with “top.sls”
  - Files end with “.sls”
  - Used to match Configurations to Minions in a variety of ways





# SaltStack - Quick overview of the basic components

- Master (sends commands, stores config stuffs)
- Minion (receives and executes commands, configurations)
- Modules (small programs, executed on the Minions)
- Formulas aka States (System Configurations)
- Grains
  - Information about each minion (OS, Memory, CPU, etc)
  - Many pre-defined Grains
  - You can create any custom entry for each system as well
  - Stored on the Minion



# SaltStack - Quick overview of the basic components

- Master (sends commands, stores config stuffs)
- Minion (receives and executes commands, configurations)
- Modules (small programs, executed on the Minions)
- Formulas aka States (System Configurations)
- Grains (Minion Information)
- Pillars
  - Stored Securely on the Master (not shared with Minions, unless explicitly asked)
  - User-Defined
  - Used to store information such as
    - Ports, File Paths, Configuration Parameters, Passwords, Keys, etc.



# SaltStack - Quick overview of the basic components

- Master (sends commands, stores config stuffs)
- Minion (receives and executes commands, configurations)
- Modules (small programs, executed on the Minions)
- Formulas aka States (System Configurations)
- Grains (Minion Information)
- Pillars (Secure Information)
- ... and lots more.



# SaltStack - Demo Time!



# SaltStack - Demo Time!

- Following Guide: <https://docs.saltstack.com/en/getstarted/fundamentals/>



# SaltStack - Demo Time!

- Following Guide: <https://docs.saltstack.com/en/getstarted/fundamentals/>
  - Spin up VirtualBox / Vagrant Demo



# SaltStack - Demo Time!

- Following Guide: <https://docs.saltstack.com/en/getstarted/fundamentals/>
  - Spin up VirtualBox / Vagrant Demo
  - View Keys



# SaltStack - Demo Time!

- Following Guide: <https://docs.saltstack.com/en/getstarted/fundamentals/>
  - Spin up VirtualBox / Vagrant Demo
  - View Keys
  - Send a command(s) and get data back
    - Get data back in misc formats (--out=[json|raw|txt|yaml|highstate|grains])





# SaltStack - Demo Time!

- Following Guide: <https://docs.saltstack.com/en/getstarted/fundamentals/>
  - Spin up VirtualBox / Vagrant Demo
  - View Keys
  - Send a command(s) and get data back
    - Get data back in misc formats (--out=[json|raw|txt|yaml|highstate|grains])
  - Targeting



# SaltStack - Demo Time!

- Following Guide: <https://docs.saltstack.com/en/getstarted/fundamentals/>
  - Spin up VirtualBox / Vagrant Demo
  - View Keys
  - Send a command(s) and get data back
    - Get data back in misc formats (--out=[json|raw|txt|yaml|highstate|grains])
  - Targeting
  - States



# SaltStack - Demo Time!

- Following Guide: <https://docs.saltstack.com/en/getstarted/fundamentals/>
  - Spin up VirtualBox / Vagrant Demo
  - View Keys
  - Send a command(s) and get data back
    - Get data back in misc formats (--out=[json|raw|txt|yaml|highstate|grains])
  - Targeting
  - States
  - Top File



# SaltStack - Demo Time!

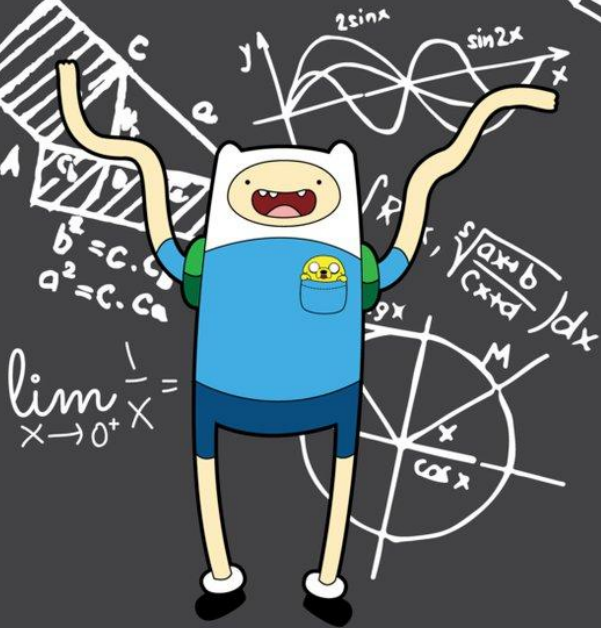
- Following Guide: <https://docs.saltstack.com/en/getstarted/fundamentals/>
  - Spin up VirtualBox / Vagrant Demo
  - View Keys
  - Send a command(s) and get data back
    - Get data back in misc formats (--out=[json|raw|txt|yaml|highstate|grains])
  - Targeting
  - States
  - Top File
- Another day
  - Functions, Pillars, Includes, Jinja, File Management, Salt-Cloud, etc.



**TIME TO COMBINE!**



# MATHEMATICAL!



iocage + salt : Ingredients



# iocage + salt : Ingredients

1. FreeBSD
  - a. ZFS
  - b. iocage
  - c. salt
2. Iocage Formula for SaltStack
  - a. <https://github.com/bougie/salt-iocage-formula>





# iocage + salt : Installation

- Install Salt and Git
  - `pkg install py27-salt git`
- Clone the salt-iocage-formula repository to /usr/local/etc/salt
  - `cd /usr/local/etc/salt/`
  - `git clone https://github.com/bougie/salt-iocage-formula.git`



# iocage + salt : Configuration

- Configure Salt-Master

- We'll use the Base System, but this could easily be another system, or even a jail.
- Set up the master config

- `cp /usr/local/etc/salt/master.sample /usr/local/etc/salt/master`

- `vim /usr/local/etc/salt/master`

- *Configure file\_roots, pillar\_roots, etc (point to cloned repo for this example)*

- Enable the SaltStack binary builds, edit /usr/local/etc/pkg/repos/saltstack.conf
  - See: <https://docs.saltstack.com/en/latest/topics/installation/freebsd.html>
- Enable the Master in /etc/rc.conf

- `sysrc salt_master_enable="yes"`

- Start the salt\_master service

- `service salt_master start`



# iocage + salt : Create a Jail

- `salt-call --local saltutils.sync_all`
- `salt-call --local iocage.fetch`
- `salt-call --local iocage.list_jails`
- `salt-call --local iocage.create full tag=sausages release=10.3-RELEASE`
- `salt-call --local iocage.start sausages`
- `salt-call --local iocage.stop sausages`
- `salt-call --local iocage.destroy sausages (NEED TO TYPE 'Y' + ENTER)`
- 

