Securing HTCondor Flocking

Kevin Hrpcek
UW-Madison Space Science and Engineering Center

SSEC

- Earth Atmospheric Research
  - Weather, climate, numerical weather prediction
  - CIMSS, SIPS, SDS, McIDAS
  - Collaboration with NOAA,NASA,NWS
- Ice
  - Ice core drilling
  - Antarctica weather stations
- Engineering
  - S-HIS Sounder
  - High speed photometer on Hubble - Removed to fix optics
- Off earth atmosphere

SSEC

WISCONSIN
UNIVERSITY OF WISCONSIN-MADISON

Satellite data processing

- High throughput satellite data processing
- Polar Orbiters
  - MODIS (Terra 1999, Aqua 2002)
  - VIIRS (SNPP 2011, NOAA20 2017)
  - CrIS (SNPP 2011, NOAA20 2017)
- GEO - experimental
  - ABI (GOES 16)
  - AHI (Himawari 8/9)
- Forward Stream Processing for Polar Orbiters
  - Uses ~20% of cluster day to day
- Periodic mission reprocessing
  - Days to weeks of processing

Flocking

- Bidirectional sharing of compute resources among HTCondor clusters
- On UW campus
  - CHTC, SSEC, WID, HEP, IceCube, Physics, DoIT, BioStat, BioChem
- Bidirectional isn’t necessary
- Jobs need to be architected to work over internet or wan
  - This is what keeps my team from flocking out
- Runs like normal condor job but as nobody user

SSEC

WISCONSIN
UNIVERSITY OF WISCONSIN-MADISON

Network

- Unrouted private network for resources
- Few hosts such as condor submitter have multiple network connections so they can be routed to from outside private network
- Compute needs many resources on private network
  - Ceph, NFS, Database

Flocking Security Problems

- Condor provides some security
  - Nobody user
- Not really secure…
  - Probe network resources
  - Break out of working directory
  - Download anything onto compute nodes
  - Primarily relying on linux user security

Possible Solutions

- Lots of firewall rules?
- Don’t flock?
- Let it be and hope for the best?
- Virtual Machines?
- Docker?
- Something else?

Docker

- Start from clean container with each restart
  - Something breaks? Restart it
- Can provide network isolation by specifying NIC to use
- Less overhead than VM
- Easily modifiable
  - Building images is easy
- Doesn’t require overhauling my infrastructure

Flocking+Docker Theory

- Create a new vlan and trunk it to the all switch ports for compute and condor submitter
- HTCondor submitter acts as the flocking vlan gateway to the internet
  - Default route for this vlan
  - NAT
- HTCondor submitter acts as a firewall between flocking and SIPS networks
  - Very important
- Each compute node runs docker and a CentOS 7 based container that is running condor_master
- Management script controls the regular startd and flocking startd

The Docker Image

```
# Build Instructions
# docker build -t centos7-flock .
# Upload to docker repo
# docker push sipsdev.sips:5000/centos7-flock

FROM centos:latest

MAINTAINER SSEC SIPS

ADD htcondor-stable-rhel7.repo /etc/yum.repos.d/htcondor-stable-rhel7.repo
RUN rpm --import http://research.cs.wisc.edu/htcondor/yum/RPM-GPG-KEY-HTCondor && yum -y upg
rade && yum -y install condor && yum clean all
ADD condor/ /etc/condor/
ADD master.sh /root/master.sh
ADD inventory_shm.sh /usr/libexec/condor/inventory_shm.sh
~
~
~
~
```

SSEC

WISCONSIN
UNIVERSITY OF WISCONSIN-MADISON

Docker Network

- Need to have container run on a specific vlan with no access to system routes or other network interfaces
- Macvlan driver
  - Directly connects a host's 'physical' interface to a running container

Host Network

```
[root@p205 ~]# ifconfig
docker0: flags=4099<UP,BROADCAST,MULTICAST>  mtu 1500
        inet 172.17.0.1  netmask 255.255.0.0  broadcast 0.0.0.0
        ether 02:42:dd:1f:5b:7b  txqueuelen 0  (Ethernet)
        RX packets 0  bytes 0 (0.0 B)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 0  bytes 0 (0.0 B)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

em1: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 10.1.2.5  netmask 255.255.0.0  broadcast 10.1.255.255
        inet6 fe80::a9e:1ff:fed8:8f34  prefixlen 64  scopeid 0x20<link>
        ether 08:9e:01:d8:8f:34  txqueuelen 1000  (Ethernet)
        RX packets 295881629664  bytes 439174759193654 (399.4 TiB)
        RX errors 0  dropped 803445  overruns 0  frame 0
        TX packets 45877875957  bytes 18994316862589 (17.2 TiB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

em1.2512: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet6 fe80::a9e:1ff:fed8:8f34  prefixlen 64  scopeid 0x20<link>
        ether 08:9e:01:d8:8f:34  txqueuelen 1000  (Ethernet)
        RX packets 8911506  bytes 22700797393 (21.1 GiB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 3761449  bytes 323156505 (308.1 MiB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

lo: flags=73<UP,LOOPBACK,RUNNING>  mtu 65536
        inet 127.0.0.1  netmask 255.0.0.0
        inet6 ::1  prefixlen 128  scopeid 0x10<host>
        loop  txqueuelen 1  (Local Loopback)
        RX packets 10049501  bytes 1718767102 (1.6 GiB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 10049501  bytes 1718767102 (1.6 GiB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0
```

SSEC

WISCONSIN
UNIVERSITY OF WISCONSIN-MADISON

```
[root@p205 ~]# docker network ls
NETWORK ID          NAME                DRIVER              SCOPE
dd29147dd165        bridge              bridge              local
ed8aa1164555        host                host                local
aed6aefa22b4        macvlan2512         macvlan             local
2462669ab377        none                null                local
[root@p205 ~]# docker network inspect macvlan2512
[
    {
        "Name": "macvlan2512",
        "Id": "aed6aefa22b49ccaa44d7bff028aca33aec08c16eb226438d246cfbadf742fb7",
        "Created": "2017-05-25T18:42:58.185032061Z",
        "Scope": "local",
        "Driver": "macvlan",
        "EnableIPv6": false,
        "IPAM": {
            "Driver": "default",
            "Options": {},
            "Config": [
                {
                    "Subnet": "10.27.0.0/16",
                    "Gateway": "10.27.0.1"
                }
            ]
        },
        "Internal": false,
        "Attachable": false,
        "Ingress": false,
        "Containers": {},
        "Options": {
            "parent": "em1.2512"
        },
        "Labels": {}
    }
]
[root@p205 ~]#
```

SSEC

WISCONSIN
UNIVERSITY OF WISCONSIN-MADISON

Container Network

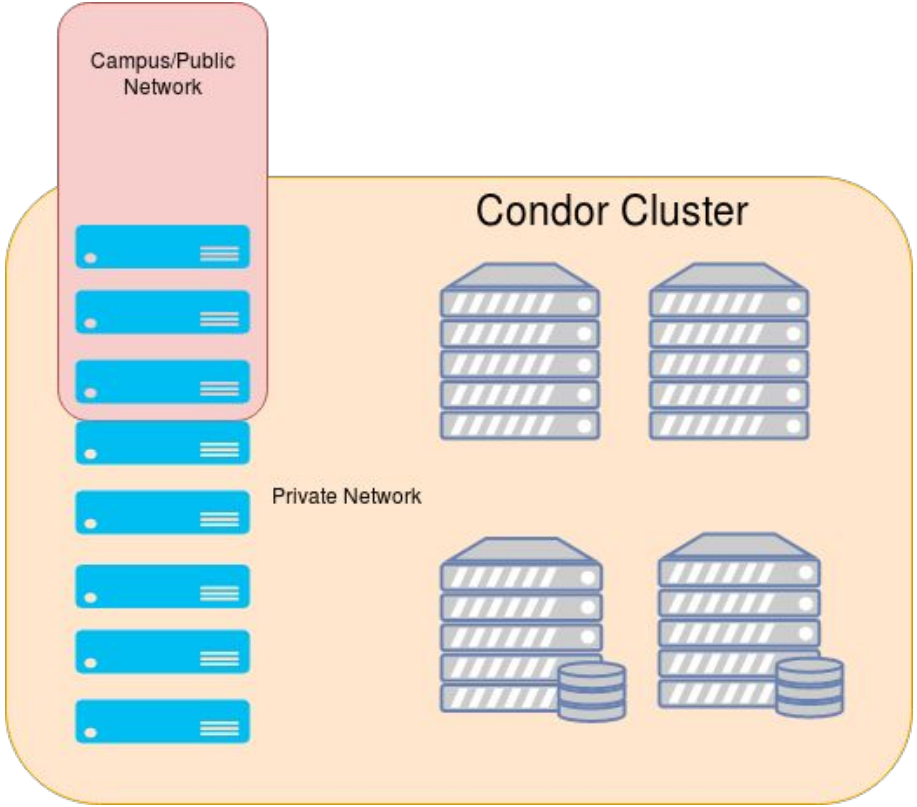
docker run --hostname f205.sips --name flocking_startd --network macvlan2512 --ip=10.27.2.5 --dns=8.8.8.8 -it -v /dev/shm --tmpfs /dev/shm:rw,nosuid,nodev,exec,size=64g sipsdev.sips:5000/centos7-flock

```
[root@p205 ~]# docker run --hostname f205.sips --name flocking_startd --network macvlan251
v /dev/shm --tmpfs /dev/shm:rw,nosuid,nodev,exec,size=64g sipsdev.sips:5000/centos7-flock
[root@f205 /]# ifconfig
bash: ifconfig: command not found
[root@f205 /]# ip a
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN qlen 1
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
       valid_lft forever preferred_lft forever
15: eth0@if4: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc noqueue state UNKNOWN
    link/ether 02:42:0a:1b:02:05 brd ff:ff:ff:ff:ff:ff link-netnsid 0
    inet 10.27.2.5/16 scope global eth0
       valid_lft forever preferred_lft forever
```

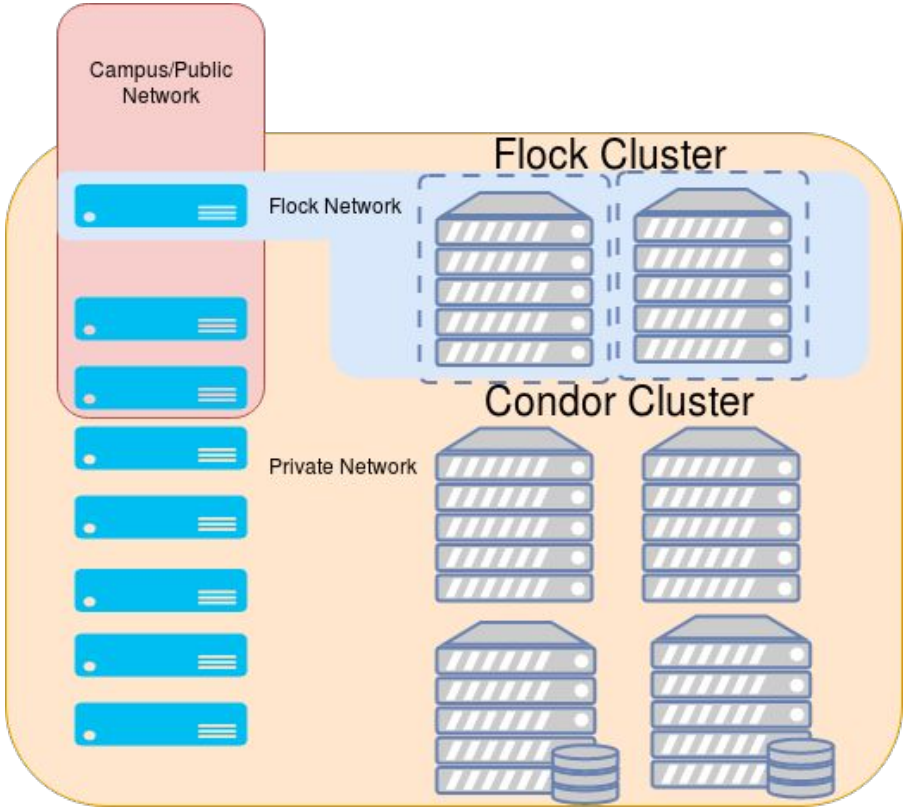
SSEC

WISCONSIN
UNIVERSITY OF WISCONSIN-MADISON

Old Network



New Network


Campus/Public Network
Flock Cluster
Flock Network
Condor Cluster
Private Network

Monitoring from HTCondor

- Regular startd hosts start with ‘p’
- Flocking containers start with ‘f’
- All show up on the condor master

```
[root@condor ~]# condor_status f220.sips
Name               OpSys      Arch   State     Activity LoadAv Mem     ActvtyTime

slot1@f220.sips    LINUX      X86_64 Unclaimed Idle      0.010 128742  6+09:45:31
                     Machines Owner Claimed Unclaimed Matched Preempting

       X86_64/LINUX        1     0       0         1       0          0

              Total        1     0       0         1       0          0
[root@condor ~]# condor_status p220.sips
[root@condor ~]# condor_status p215.sips
Name               OpSys      Arch   State     Activity LoadAv Mem     ActvtyTime

slot1@p215.sips    LINUX      X86_64 Unclaimed Idle      0.270 126694 71+18:20:02
slot1_1@p215.sips  LINUX      X86_64 Claimed   Busy      0.120 1024   3+02:45:27
slot1_2@p215.sips  LINUX      X86_64 Claimed   Busy      0.170 1024   3+02:45:21
                     Machines Owner Claimed Unclaimed Matched Preempting

       X86_64/LINUX        3     0       2         1       0          0

              Total        3     0       2         1       0          0
```

SSEC

WISCONSIN
UNIVERSITY OF WISCONSIN-MADISON

Shepherd

- Python program that manages the flock
- Runs on condor master
- Uses python bindings to keep track of everything
- Turns regular and flocking startd on and off as necessary
- /tmp/flockoff override
- Always prefers local work to flocking
- Leave ~25% of cluster to not flock
- Run with circus or systemd

Shepherd Script Logic

- If /tmp/flockoff: ensure all flocking disabled; else
- Get status of all hosts, regular and flock, and store it
- Check condor queue
- If idle queue < 600 and not all hosts are flocking
  - Condor_off $x number of regular startd (p220) condor_on flock container on that physical host (f220)
  - Disable startd process monitoring in Icinga2
- Elif idle queue > 600 and there is active flocking
  - Condor_off $y flocking startd, condor_on corresponding physical condor startd
  - Enable startd process monitoring in Icinga2
- Sleep 5 min and repeat

Shepherd Status

- Prints current status of all shepherd managed hosts

```
[root@vultur ~]# /opt/shepherd/shepherd.py -s
Active Regular Compute Node: 4
['p-7-17.sips', 'p212.sips', 'p302.sips', 'p306.sips']


Active Flock Compute Nodes: 71
['f-7-1.sips', 'f-7-10.sips', 'f-7-11.sips', 'f-7-12.sips', 'f-7-13.sips', 'f-7-14.sips', 'f-7-15.sips', 'f-7-16.sips', 'f-7-18.sips', 'f-7-19.sips'
, 'f-7-2.sips', 'f-7-20.sips', 'f-7-21.sips', 'f-7-22.sips', 'f-7-23.sips', 'f-7-24.sips', 'f-7-25.sips', 'f-7-26.sips', 'f-7-27.sips', 'f-7-28.sips
', 'f-7-29.sips', 'f-7-3.sips', 'f-7-30.sips', 'f-7-31.sips', 'f-7-32.sips', 'f-7-33.sips', 'f-7-34.sips', 'f-7-35.sips', 'f-7-36.sips', 'f-7-37.sip
s', 'f-7-38.sips', 'f-7-39.sips', 'f-7-4.sips', 'f-7-40.sips', 'f-7-41.sips', 'f-7-42.sips', 'f-7-5.sips', 'f-7-6.sips', 'f-7-7.sips', 'f-7-8.sips',
 'f-7-9.sips', 'f203.sips', 'f204.sips', 'f213.sips', 'f214.sips', 'f215.sips', 'f216.sips', 'f217.sips', 'f218.sips', 'f220.sips', 'f221.sips', 'f2
22.sips', 'f223.sips', 'f224.sips', 'f225.sips', 'f226.sips', 'f227.sips', 'f228.sips', 'f229.sips', 'f230.sips', 'f231.sips', 'f232.sips', 'f233.si
ps', 'f234.sips', 'f235.sips', 'f236.sips', 'f237.sips', 'f238.sips', 'f239.sips', 'f240.sips', 'f305.sips']
[root@vultur ~]#
```

SSEC

WISCONSIN
UNIVERSITY OF WISCONSIN-MADISON

Puppet

- Install docker
- Set up em1.2512 host interface
- Set up macvlan2512 docker network
- Install systemd service to manage flocking container

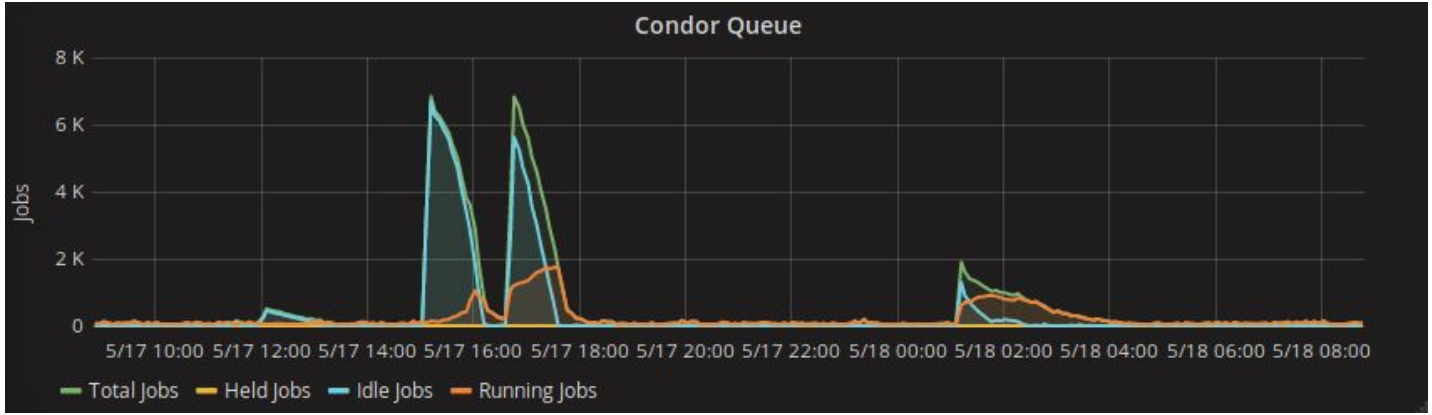
```
[Unit]
Description=Flocking Startd Container
After=docker.service
Requires=docker.service

[Service]
TimeoutStartSec=5min
Restart=always
ExecStartPre=/usr/bin/docker login -u asdf -p xxxxxxxxxxx  yyyyyyyyy.ssec.wisc.edu
ExecStart=/usr/bin/docker run --name flocking_startd --hostname f222.sips --network macvlan2512 --ip=10.27.2.22 --dns=8.8.8.8 --cap-drop=net_bind_service --cap-drop=net_raw
--cap-drop=mknod --cap-drop=sys_chroot -v /dev/shm --tmpfs /dev/shm:rw,nosuid,nodev,exec,size=64g yyyyyyyyy.ssec.wisc.edu/asdf/code/shepherd/centos7-flock /root/master.sh
ExecStartPost=/usr/bin/docker logout yyyyyyyyy.ssec.wisc.edu
ExecStop=/usr/bin/docker rm -f flocking_startd
```

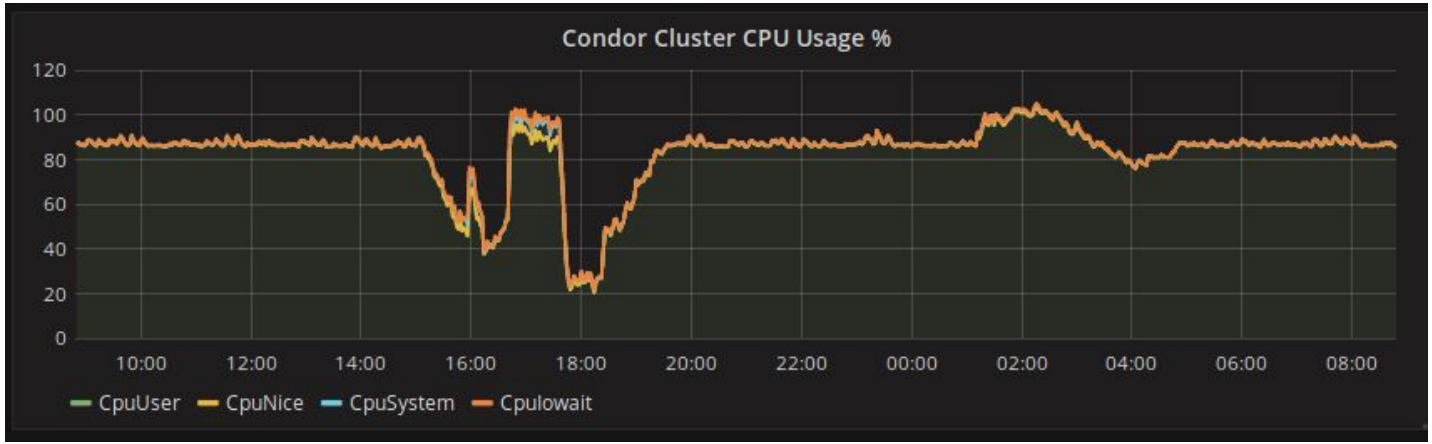
What does all this get me?

- Unprivileged user
- Unprivileged container
- Reduced Capabilities
- On a firewalled host
- On a firewalled vlan with no access to my private network

Risks

- Break out of container
- Keep kernel up to date to mitigate risks
- Only sharing /dev/shm to container
- A slip up in firewall rules could cause access to my network
- Other?





SSEC

WISCONSIN
UNIVERSITY OF WISCONSIN-MADISON

Questions?