Services

Stephen James

Clients vs Servers

- Clients consume services
- Servers provide services
- However, there will typically be services running on both clients and servers

What are protocols?

- Rules that define a common "language" for exchanging data
- These can be layered
  - You've already heard about TCP and UDP, which both build on top of IP
- Allow multiple implementations of services that can communicate with each other, and can use the same clients

So… what is a service anyway?

- In short, a service is a set of one or more functionalities provided by software
- Many services run as daemons (background processes)
- Many services will be set to automatically start once a system boots
- Some types of services that are commonly accessed over the network will have standard ports
  - These ports can usually be changed
  - Some services of the same type will use different ports since they are ports will vary

Common services

Database management systems

- Provide a way to store, manage, and access data
- No "standard" ports, DBMSs have their own communication protocols
  - Usually have their own clients to interact with them
- Popular examples:
  - MariaDB/MySQL: 3306/tcp
  - Microsoft SQL Server (MSSQL): 1433/tcp
  - MongoDB: 27017/tcp
  - PostgreSQL: 5432/tcp
  - Redis: 6379/tcp

Domain Name System

- Hierarchical and decentralized naming system for computers
- Allow use of domain names instead of IP address (e.g. A and AAAA records)
  - Numbers tend to be harder to remember and express
- Allow pointing domain name to another domain name (e.g. CNAME records)
  - Setting up canonical name records effectively creates aliases
- Allow find domain names for IP address (e.g. PTR records)
  - Reverse DNS lookup
- "Forwarder" vs "resolver"
  - Forwarders only forward incoming requests to other DNS servers to be handled
  - Resolvers can respond with local records, in addition to forwarding

Domain Name System

Standard ports:

- 53/tcp
- 53/udp

Popular examples:

- BIND
- Dnsmasq
- PowerDNS

Useful utilities:

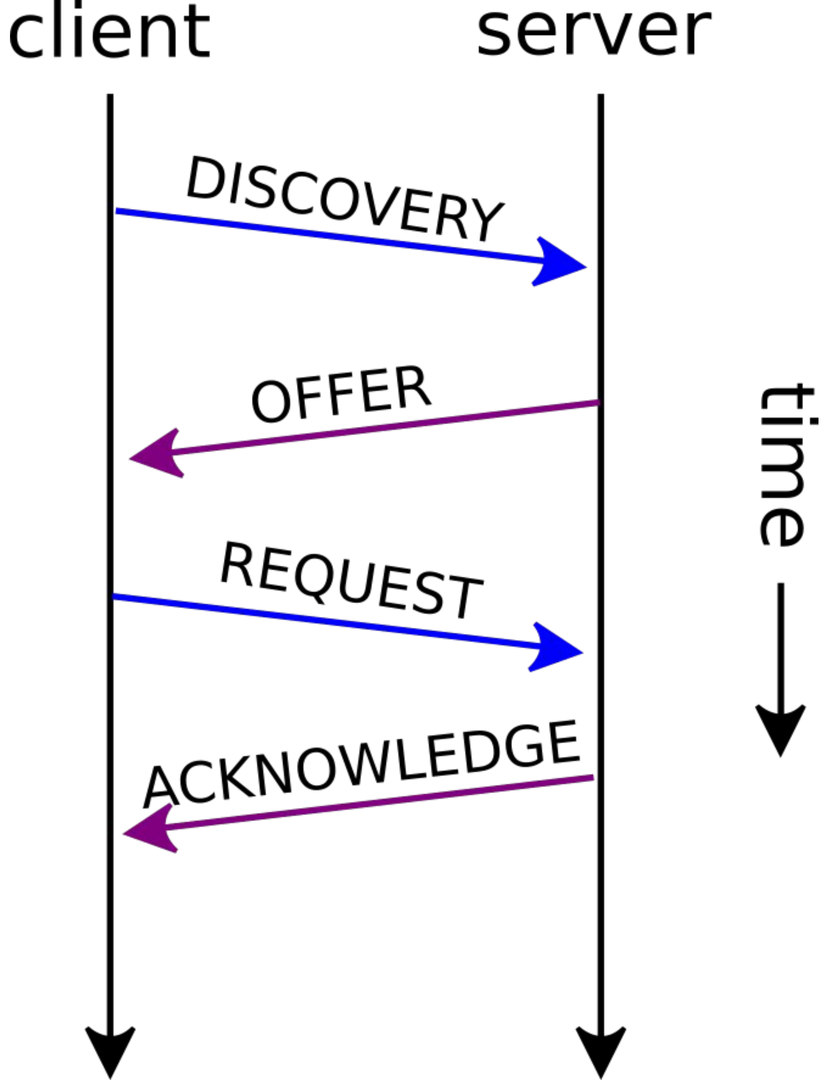
- dig (domain information groper)
- host
- nslookup

Dynamic Host Configuration Protocol

- Allows us to easily get and centrally manage network configuration
  - Can give us IP addresses, gateways, subnet masks, DNS servers, etc.
  - Eliminates the need to statically assign network configuration to all machines
- "DHCP pool" refers to a range of IP addresses available for
- Many routers offer this, but it can also be installed through things like:
  - Dnsmasq
  - FreeRADIUS
  - DHCP server role on Windows Server
- Standard ports:
  - Server: 67/udp
  - Client: 68/udp

DHCP steps

1. Client tries to find available DHCP servers
   a. Will use Automatic Private IP Addressing (APIPA) if no response
2. Servers respond, offering a lease for an IP address
3. Client accepts the first offer by requesting the offered address
4. Server sends an acknowledgement (or a negative acknowledgement if the address is unavailable)



File Transfer Protocol

- Used for file transfer over a network
- FTP transmits data (including credentials) in plaintext
- FTPS adds support for TLS
- Standard ports:
  - FTP: 21/tcp
  - FTPS: 990/tcp
- Popular examples:
  - IIS
  - PureFTPd
  - vsftpd

Logging

Mail

Secure Shell

- Provides a way to securely communicating over an unsecured network
  - Typically used to access a shell (via the command line) or to remotely execute a command
  - Among other things, it can also be used to copy files (e.g. SCP and SFTP)
- Standard port: 22/tcp
- OpenSSH is, by far, the most common SSH server

Web

- Web servers process incoming requests from clients for web resources over HTTP and related protocols
  - Web resources are identified by a Uniform Resource Locator (URL)
  - Might perform additional processing while handling the request
- HTTP is unencrypted; data is transmitted in plaintext
  - Anyone on any of the networks on a path from you to the server can see this data
- HTTPS is an extension of HTTP that is encrypted using TLS, or previously, SSL
  - Client is also able to authenticate the server (using the server's certificate)

Web

Ports:

- HTTP: 80/tcp
- HTTPS: 443/tcp

Popular software:

- Apache HTTP Server (httpd)
- Apache Tomcat
- Internet Information Services (IIS)
- lighttpd
- Nginx

Useful client tools:

- Web browsers
- cURL
- GNU Wget

Many services work together to make network communication work as it does today!

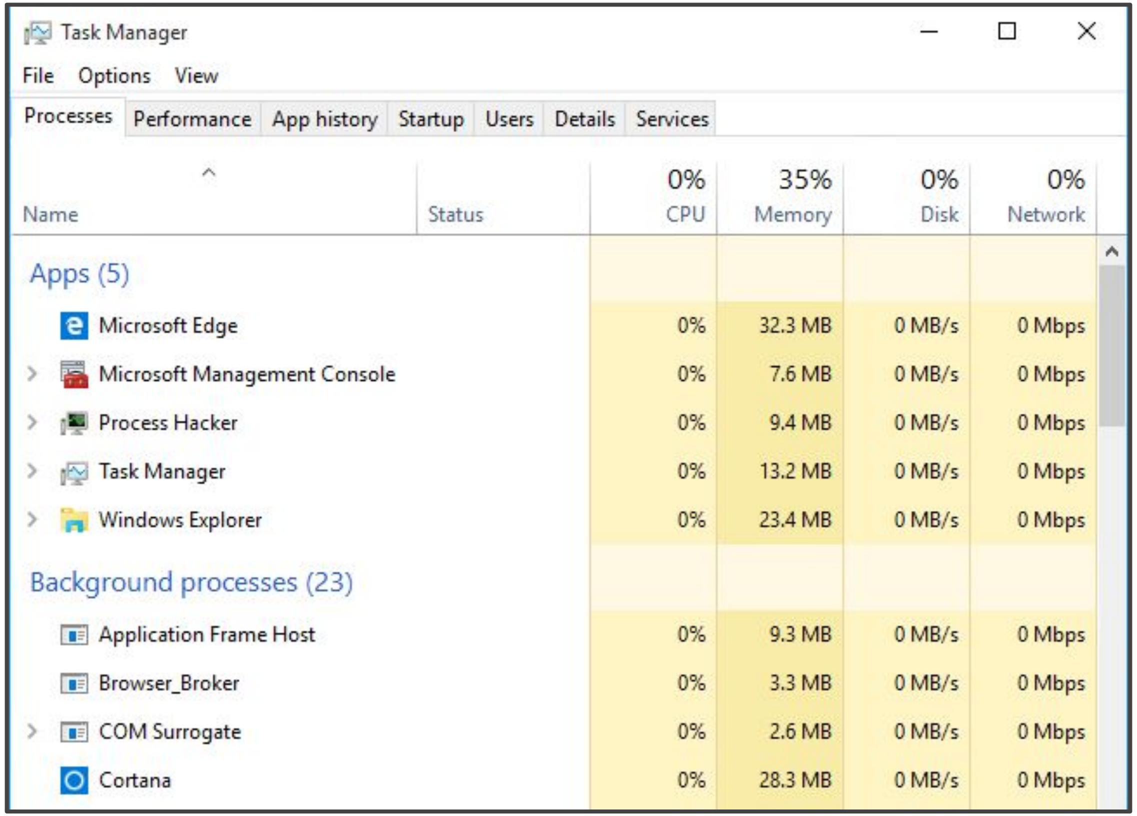
How we get to `https://ubnetdef.org/`

1. Get an IP address, gateway, etc.
   a. Either via DHCP or static IP configuration
2. Resolve "ubnetdef.org" to an IP address
   a. Ask a DNS server for the A (of using IPv4) or AAAA (if using IPv6) records for "ubnetdef.org"
   b. DNS server should respond with "128.205.44.157"
3. Send an HTTP GET request to 128.205.44.157 asking for host ubnetdef.org and path "/"
   a. TCP handshake starts, and public keys etc. are exchanged (since we're using HTTPS)
   b. Client (browsers etc.) will do
   c. Web server processes request then responds

Note that the above steps are simplified: a lot more happens!

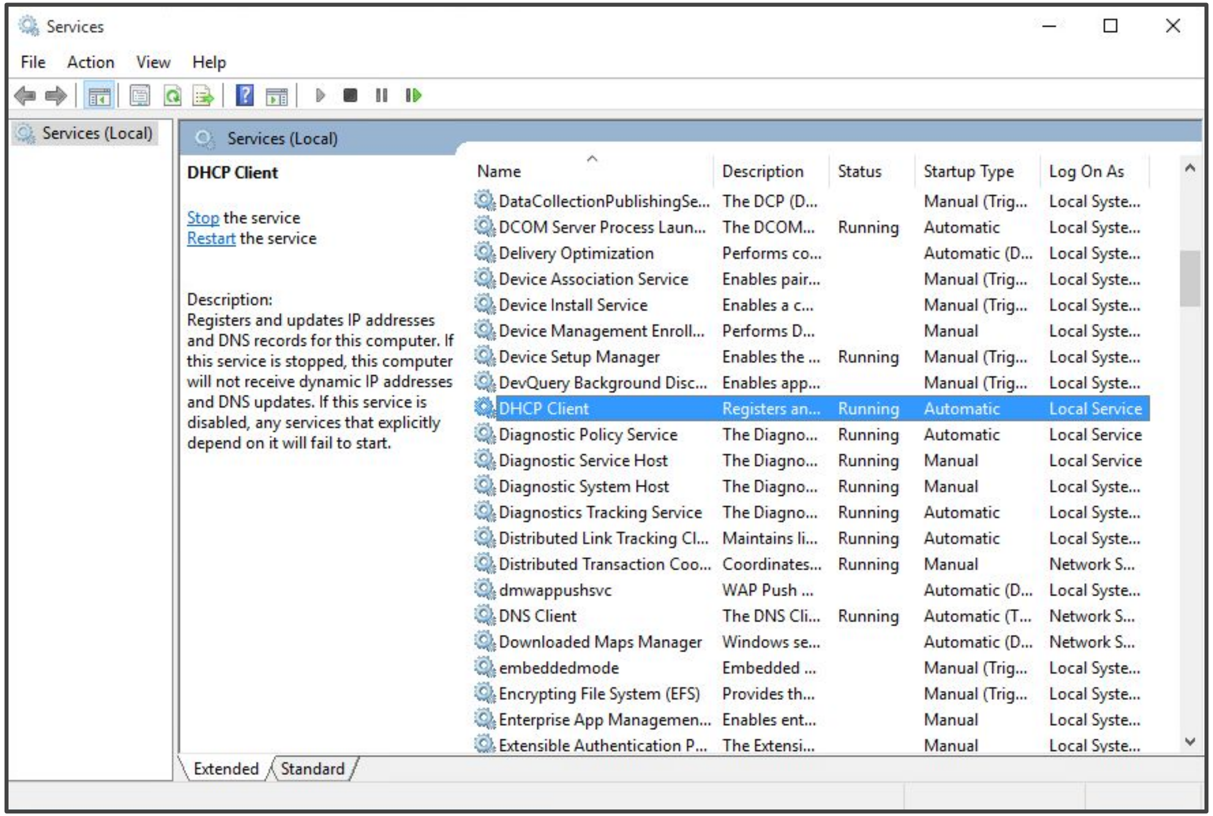
Managing services

Task manager (Windows)


Task Manager
File Options View
Processes Performance App history Startup Users Details Services

Name	Status	0% CPU	35% Memory	0% Disk	0% Network
Apps (5)					
Microsoft Edge		0%	32.3 MB	0 MB/s	0 Mbps
Microsoft Management Console		0%	7.6 MB	0 MB/s	0 Mbps
Process Hacker		0%	9.4 MB	0 MB/s	0 Mbps
Task Manager		0%	13.2 MB	0 MB/s	0 Mbps
Windows Explorer		0%	23.4 MB	0 MB/s	0 Mbps
Background processes (23)					
Application Frame Host		0%	9.3 MB	0 MB/s	0 Mbps
Browser_Broker		0%	3.3 MB	0 MB/s	0 Mbps
COM Surrogate		0%	2.6 MB	0 MB/s	0 Mbps
Cortana		0%	28.3 MB	0 MB/s	0 Mbps

services.msc (Windows)



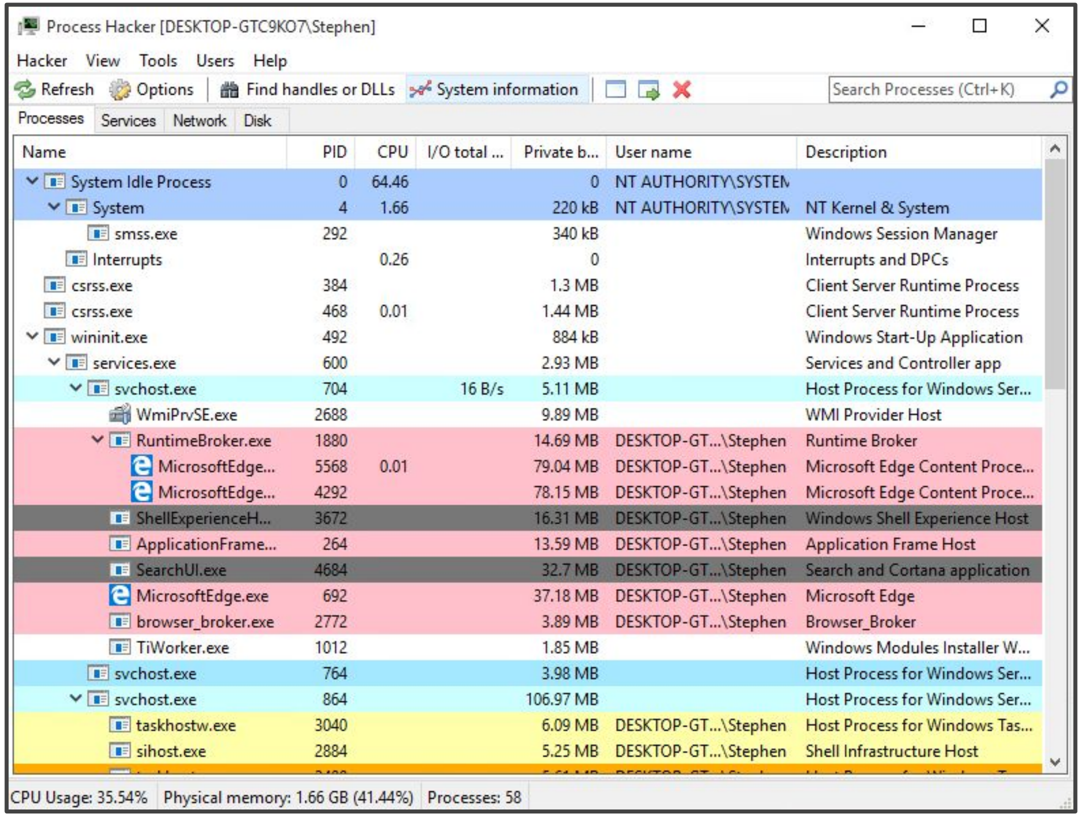
DHCP Client

Stop the service
Restart the service

Description:
Registers and updates IP addresses and DNS records for this computer. If this service is stopped, this computer will not receive dynamic IP addresses and DNS updates. If this service is disabled, any services that explicitly depend on it will fail to start.

Name	Description	Status	Startup Type	Log On As
DataCollectionPublishingSe...	The DCP (D...		Manual (Trig...	Local Syste...
DCOM Server Process Laun...	The DCOM...	Running	Automatic	Local Syste...
Delivery Optimization	Performs co...		Automatic (D...	Local Syste...
Device Association Service	Enables pair...		Manual (Trig...	Local Syste...
Device Install Service	Enables a c...		Manual (Trig...	Local Syste...
Device Management Enroll...	Performs D...		Manual	Local Syste...
Device Setup Manager	Enables the ...	Running	Manual (Trig...	Local Syste...
DevQuery Background Disc...	Enables app...		Manual (Trig...	Local Syste...
DHCP Client	Registers an...	Running	Automatic	Local Service
Diagnostic Policy Service	The Diagno...	Running	Automatic	Local Service
Diagnostic Service Host	The Diagno...	Running	Manual	Local Service
Diagnostic System Host	The Diagno...	Running	Manual	Local Syste...
Diagnostics Tracking Service	The Diagno...	Running	Automatic	Local Syste...
Distributed Link Tracking Cl...	Maintains li...	Running	Automatic	Local Syste...
Distributed Transaction Coo...	Coordinates...	Running	Manual	Network S...
dmwappushsvc	WAP Push ...		Automatic (D...	Local Syste...
DNS Client	The DNS Cli...	Running	Automatic (T...	Network S...
Downloaded Maps Manager	Windows se...		Automatic (D...	Network S...
embeddedmode	Embedded ...		Manual (Trig...	Local Syste...
Encrypting File System (EFS)	Provides th...		Manual (Trig...	Local Syste...
Enterprise App Managemen...	Enables ent...		Manual	Local Syste...
Extensible Authentication P...	The Extensi...		Manual	Local Syste...

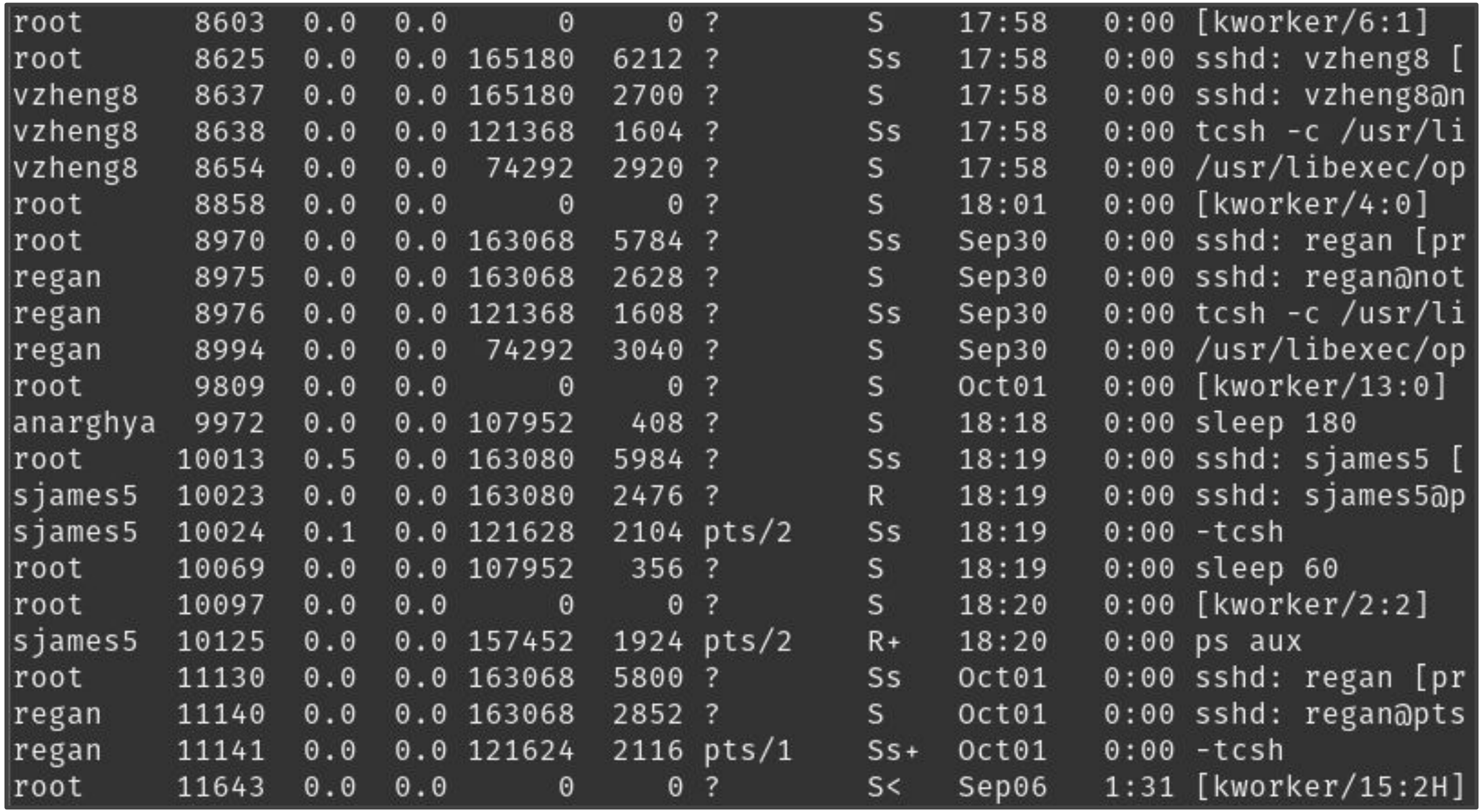
Process Hacker (Windows)


Process Hacker [DESKTOP-GTC9KO7\Stephen]
Hacker View Tools Users Help
Refresh
Options
Find handles or DLLs
System information
Search Processes (Ctrl+K)
Processes
Services
Network
Disk

Name	PID	CPU	I/O total ...	Private b...	User name	Description
System Idle Process	0	64.46		0	NT AUTHORITY\SYSTEM	
System	4	1.66		220 kB	NT AUTHORITY\SYSTEM	NT Kernel & System
smss.exe	292			340 kB		Windows Session Manager
Interrupts		0.26		0		Interrupts and DPCs
csrss.exe	384			1.3 MB		Client Server Runtime Process
csrss.exe	468	0.01		1.44 MB		Client Server Runtime Process
wininit.exe	492			884 kB		Windows Start-Up Application
services.exe	600			2.93 MB		Services and Controller app
svchost.exe	704		16 B/s	5.11 MB		Host Process for Windows Ser...
WmiPrvSE.exe	2688			9.89 MB		WMI Provider Host
RuntimeBroker.exe	1880			14.69 MB	DESKTOP-GT...\Stephen	Runtime Broker
MicrosoftEdge...	5568	0.01		79.04 MB	DESKTOP-GT...\Stephen	Microsoft Edge Content Proce...
MicrosoftEdge...	4292			78.15 MB	DESKTOP-GT...\Stephen	Microsoft Edge Content Proce...
ShellExperienceH...	3672			16.31 MB	DESKTOP-GT...\Stephen	Windows Shell Experience Host
ApplicationFrame...	264			13.59 MB	DESKTOP-GT...\Stephen	Application Frame Host
SearchUI.exe	4684			32.7 MB	DESKTOP-GT...\Stephen	Search and Cortana application
MicrosoftEdge.exe	692			37.18 MB	DESKTOP-GT...\Stephen	Microsoft Edge
browser_broker.exe	2772			3.89 MB	DESKTOP-GT...\Stephen	Browser_Broker
TiWorker.exe	1012			1.85 MB		Windows Modules Installer W...
svchost.exe	764			3.98 MB		Host Process for Windows Ser...
svchost.exe	864			106.97 MB		Host Process for Windows Ser...
taskhostw.exe	3040			6.09 MB	DESKTOP-GT...\Stephen	Host Process for Windows Tas...
sihost.exe	2884			5.25 MB	DESKTOP-GT...\Stephen	Shell Infrastructure Host

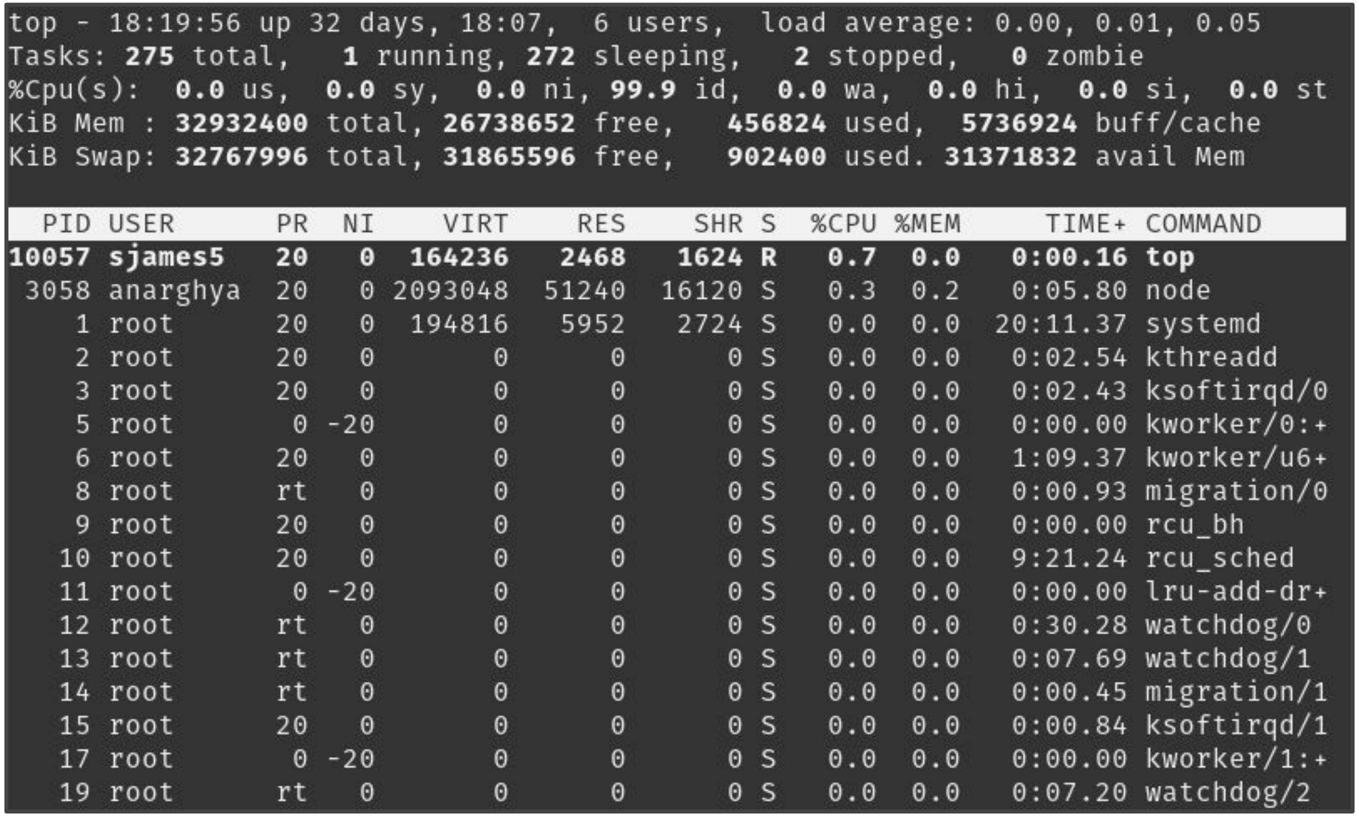
CPU Usage: 35.54% Physical memory: 1.66 GB (41.44%) Processes: 58

ps (Unix)



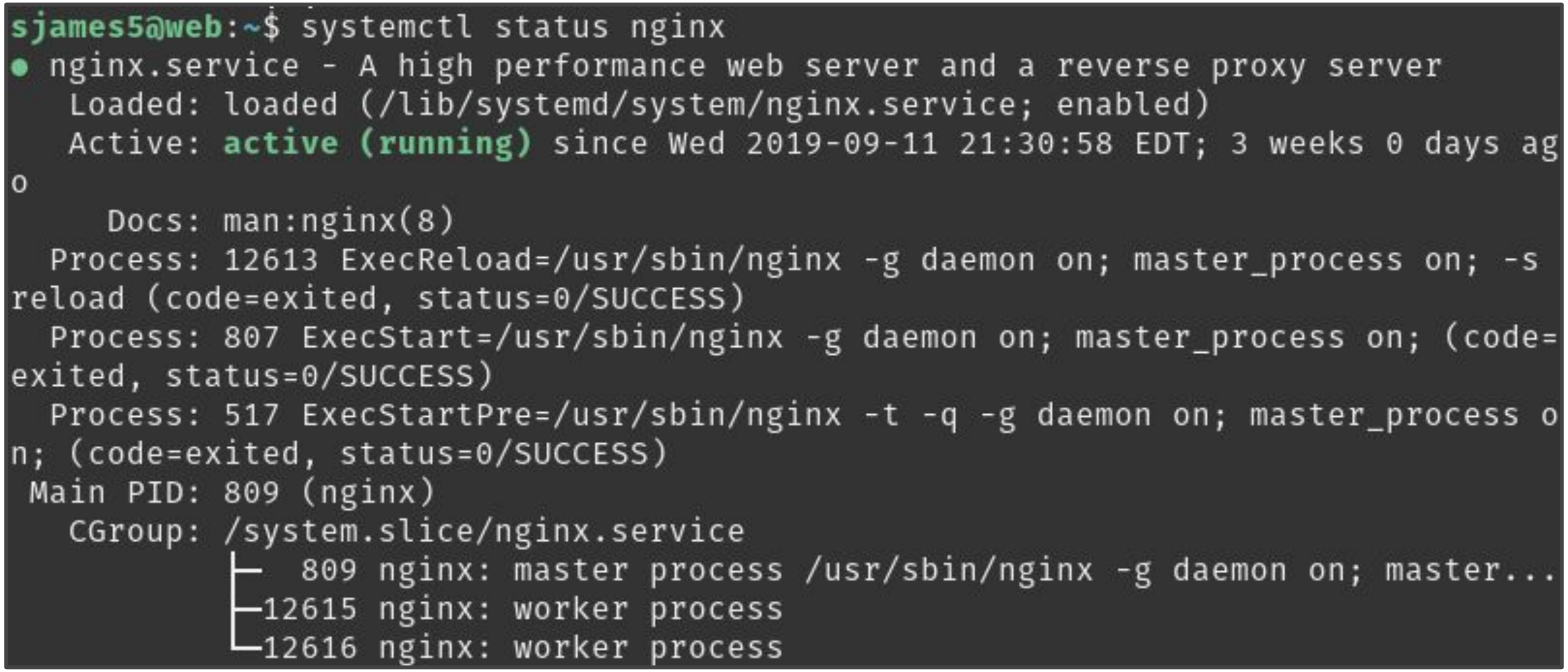
```
root       8603  0.0  0.0      0     0 ?        S    17:58   0:00 [kworker/6:1]
root       8625  0.0  0.0 165180  6212 ?        Ss   17:58   0:00 sshd: vzheng8 [
vzheng8    8637  0.0  0.0 165180  2700 ?        S    17:58   0:00 sshd: vzheng8@n
vzheng8    8638  0.0  0.0 121368  1604 ?        Ss   17:58   0:00 tcsh -c /usr/li
vzheng8    8654  0.0  0.0  74292  2920 ?        S    17:58   0:00 /usr/libexec/op
root       8858  0.0  0.0      0     0 ?        S    18:01   0:00 [kworker/4:0]
root       8970  0.0  0.0 163068  5784 ?        Ss   Sep30   0:00 sshd: regan [pr
regan      8975  0.0  0.0 163068  2628 ?        S    Sep30   0:00 sshd: regan@not
regan      8976  0.0  0.0 121368  1608 ?        Ss   Sep30   0:00 tcsh -c /usr/li
regan      8994  0.0  0.0  74292  3040 ?        S    Sep30   0:00 /usr/libexec/op
root       9809  0.0  0.0      0     0 ?        S    Oct01   0:00 [kworker/13:0]
anarghya   9972  0.0  0.0 107952   408 ?        S    18:18   0:00 sleep 180
root      10013  0.5  0.0 163080  5984 ?        Ss   18:19   0:00 sshd: sjames5 [
sjames5   10023  0.0  0.0 163080  2476 ?        R    18:19   0:00 sshd: sjames5@p
sjames5   10024  0.1  0.0 121628  2104 pts/2    Ss   18:19   0:00 -tcsh
root      10069  0.0  0.0 107952   356 ?        S    18:19   0:00 sleep 60
root      10097  0.0  0.0      0     0 ?        S    18:20   0:00 [kworker/2:2]
sjames5   10125  0.0  0.0 157452  1924 pts/2    R+   18:20   0:00 ps aux
root      11130  0.0  0.0 163068  5800 ?        Ss   Oct01   0:00 sshd: regan [pr
regan     11140  0.0  0.0 163068  2852 ?        S    Oct01   0:00 sshd: regan@pts
regan     11141  0.0  0.0 121624  2116 pts/1    Ss+  Oct01   0:00 -tcsh
root      11643  0.0  0.0      0     0 ?        S<   Sep06   1:31 [kworker/15:2H]
```

top (Unix)



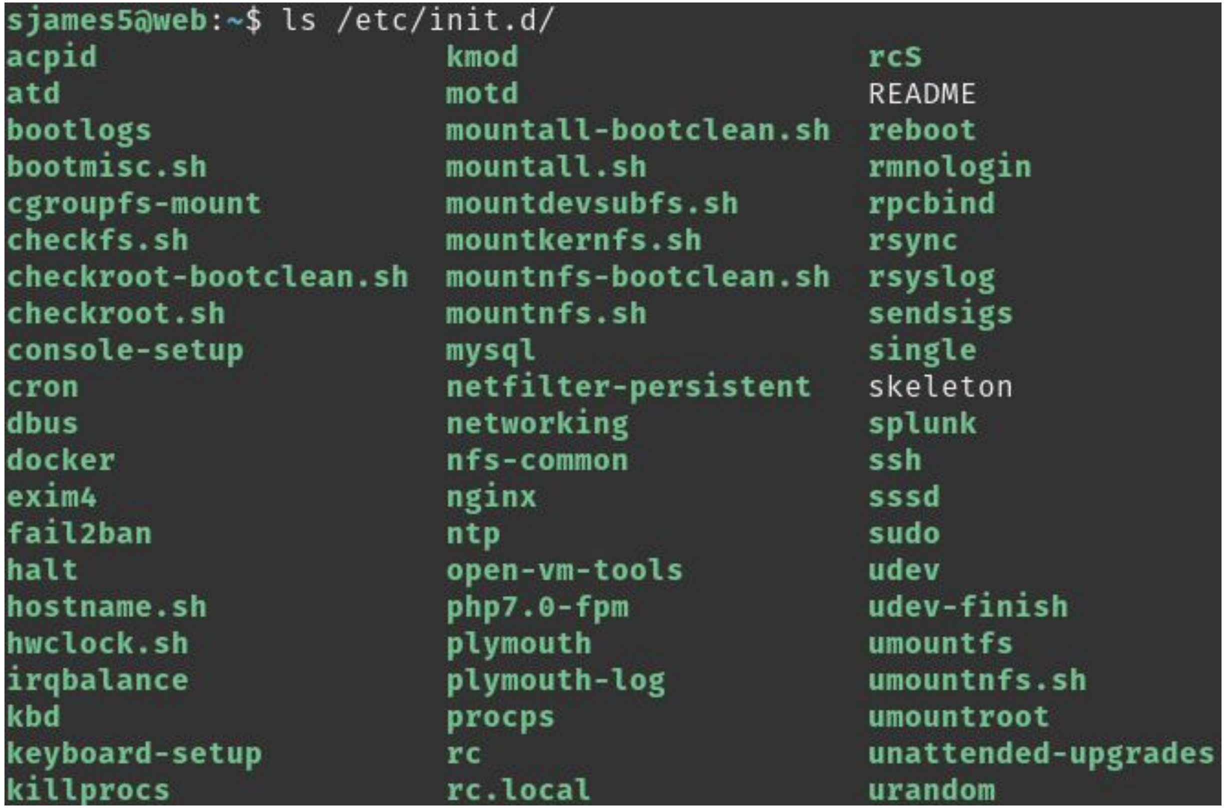
```
top - 18:19:56 up 32 days, 18:07,  6 users,  load average: 0.00, 0.01, 0.05
Tasks: 275 total,   1 running, 272 sleeping,   2 stopped,   0 zombie
%Cpu(s):  0.0 us,  0.0 sy,  0.0 ni, 99.9 id,  0.0 wa,  0.0 hi,  0.0 si,  0.0 st
KiB Mem : 32932400 total, 26738652 free,   456824 used,  5736924 buff/cache
KiB Swap: 32767996 total, 31865596 free,   902400 used. 31371832 avail Mem

  PID USER      PR  NI    VIRT    RES    SHR S  %CPU %MEM     TIME+ COMMAND
10057 sjames5   20   0  164236   2468   1624 R   0.7  0.0   0:00.16 top
 3058 anarghya  20   0 2093048  51240  16120 S   0.3  0.2   0:05.80 node
    1 root      20   0  194816   5952   2724 S   0.0  0.0  20:11.37 systemd
    2 root      20   0       0      0      0 S   0.0  0.0   0:02.54 kthreadd
    3 root      20   0       0      0      0 S   0.0  0.0   0:02.43 ksoftirqd/0
    5 root       0 -20       0      0      0 S   0.0  0.0   0:00.00 kworker/0:+
    6 root      20   0       0      0      0 S   0.0  0.0   1:09.37 kworker/u6+
    8 root      rt   0       0      0      0 S   0.0  0.0   0:00.93 migration/0
    9 root      20   0       0      0      0 S   0.0  0.0   0:00.00 rcu_bh
   10 root      20   0       0      0      0 S   0.0  0.0   9:21.24 rcu_sched
   11 root       0 -20       0      0      0 S   0.0  0.0   0:00.00 lru-add-dr+
   12 root      rt   0       0      0      0 S   0.0  0.0   0:30.28 watchdog/0
   13 root      rt   0       0      0      0 S   0.0  0.0   0:07.69 watchdog/1
   14 root      rt   0       0      0      0 S   0.0  0.0   0:00.45 migration/1
   15 root      20   0       0      0      0 S   0.0  0.0   0:00.84 ksoftirqd/1
   17 root       0 -20       0      0      0 S   0.0  0.0   0:00.00 kworker/1:+
   19 root      rt   0       0      0      0 S   0.0  0.0   0:07.20 watchdog/2
```

systemd (Linux)



```
sjames5@web:~$ systemctl status nginx
● nginx.service - A high performance web server and a reverse proxy server
   Loaded: loaded (/lib/systemd/system/nginx.service; enabled)
   Active: active (running) since Wed 2019-09-11 21:30:58 EDT; 3 weeks 0 days ag
o
     Docs: man:nginx(8)
  Process: 12613 ExecReload=/usr/sbin/nginx -g daemon on; master_process on; -s
reload (code=exited, status=0/SUCCESS)
  Process: 807 ExecStart=/usr/sbin/nginx -g daemon on; master_process on; (code=
exited, status=0/SUCCESS)
  Process: 517 ExecStartPre=/usr/sbin/nginx -t -q -g daemon on; master_process o
n; (code=exited, status=0/SUCCESS)
 Main PID: 809 (nginx)
   CGroup: /system.slice/nginx.service
           ├─  809 nginx: master process /usr/sbin/nginx -g daemon on; master...
           ├─12615 nginx: worker process
           └─12616 nginx: worker process
```

/etc/init.d (Unix)



```
sjames5@web:~$ ls /etc/init.d/
acpid                    kmod                     rcS
atd                      motd                     README
bootlogs                 mountall-bootclean.sh    reboot
bootmisc.sh              mountall.sh              rmnologin
cgroupfs-mount           mountdevsubfs.sh         rpcbind
checkfs.sh               mountkernfs.sh           rsync
checkroot-bootclean.sh   mountnfs-bootclean.sh    rsyslog
checkroot.sh             mountnfs.sh              sendsigs
console-setup            mysql                    single
cron                     netfilter-persistent     skeleton
dbus                     networking               splunk
docker                   nfs-common               ssh
exim4                    nginx                    sssd
fail2ban                 ntp                      sudo
halt                     open-vm-tools            udev
hostname.sh              php7.0-fpm               udev-finish
hwclock.sh               plymouth                 umountfs
irqbalance               plymouth-log             umountnfs.sh
kbd                      procps                   umountroot
keyboard-setup           rc                       unattended-upgrades
killprocs                rc.local                 urandom
```

Additional tools

- kill
- pstree
-

How to know about your services

Scan your network/hosts

- Network/host scans can expose ports that are open/closed/filtered
- Knowing what ports are open can help with determining what services are running, but tools like `nmap` can often check what specific services (including versions) are installed

See what services are running

- Using tools described earlier
- Check configuration files
- Check logs (log files, journalctl, etc.)