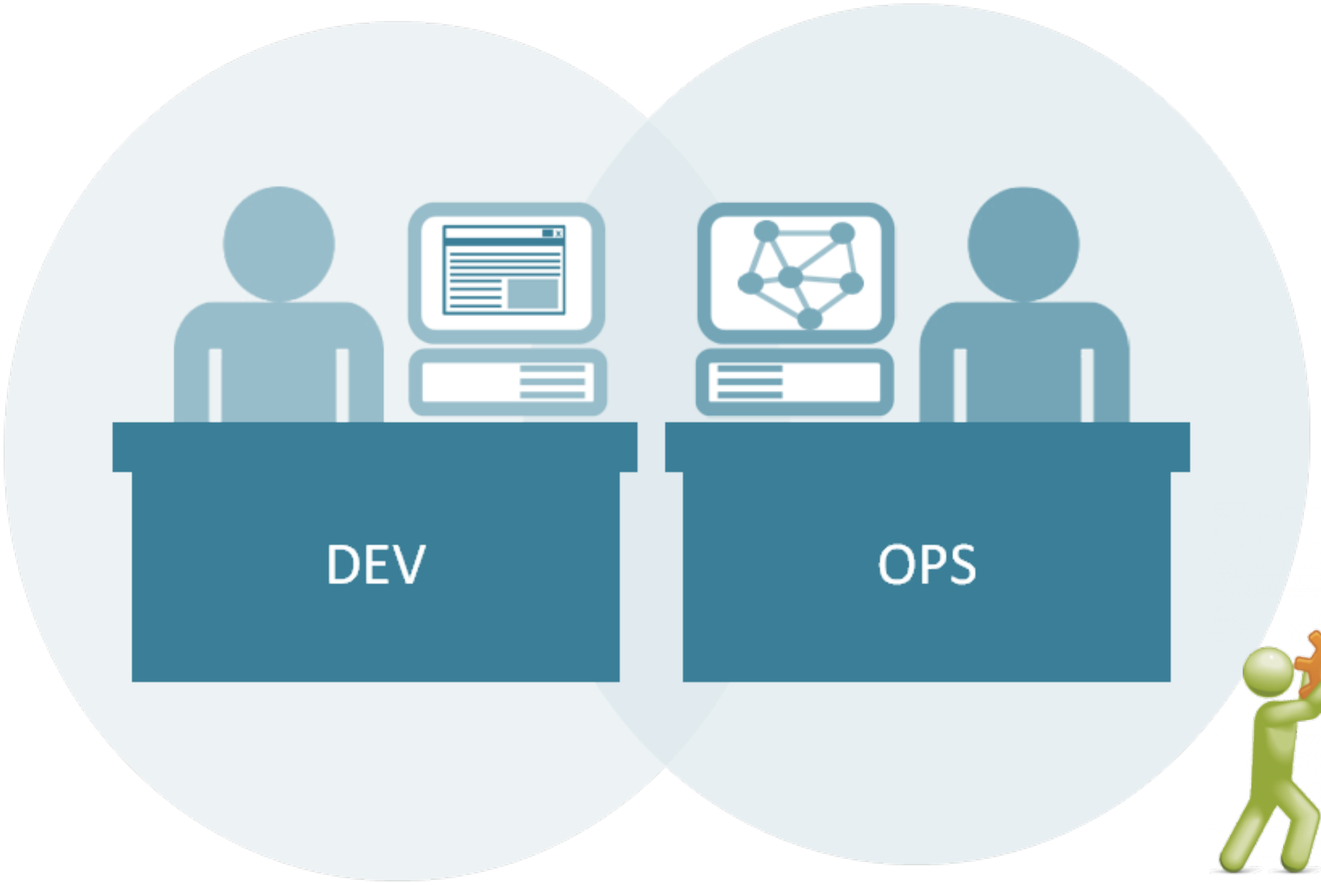
Software Operations

CS480 Software Engineering

Dr. Yu Sun
http://yusun.io
yusun@cpp.edu

CAL POLY POMONA



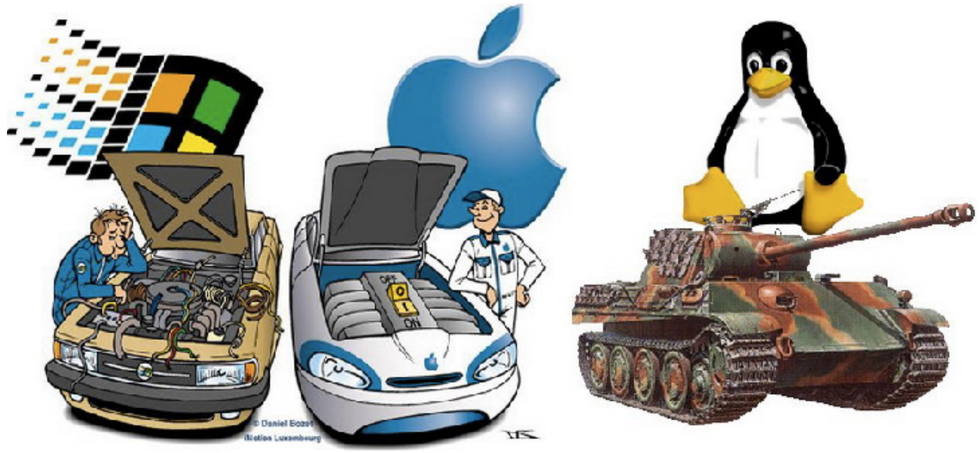
Developers VS Operators

Linux Basics

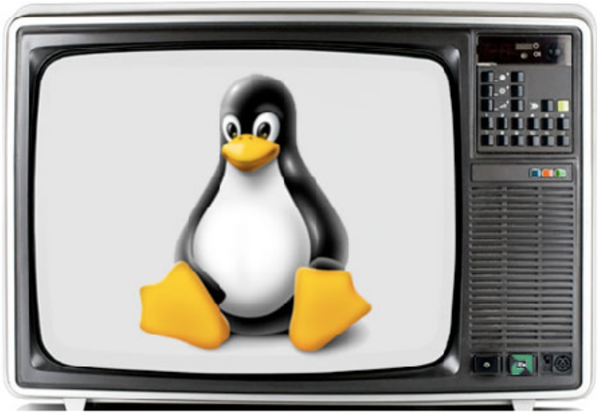
MICROSOFT
Microsoft
Microsoft
Microsoft Windows/386
Version 2.1
Microsoft Windows
for Workgroups
Version 3.1
Microsoft Windows NT
Version 3.1
Microsoft Windows
for Workgroups
Version 3.11
Microsoft Windows NT
Workstation
Version 3.5
Microsoft
Windows 95
Microsoft Internet Explorer
Microsoft
Windows NT
Workstation
Microsoft
Windows 98
Microsoft
Windows 98
Second Edition
Microsoft
Windows 2000
Professional
Microsoft
Windows Me
Millennium
Edition
Microsoft
Windows Server 2003
Standard Edition
Windows Vista
Ultimate
Windows 7
Home Premium

```
From: wschlich <wschlich@prometheus.bla.fasel.org>
To: wschlich@prometheus.bla.fasel.org
Subject: Messages from example.sh running on prometheus.bla.fasel.org
Date: Wed, 27 May 2009 21:37:34 +0200 (CEST)

[2009-05-27 21:37:33 +02:00] [____DEBUG] {example.lib.sh:49}, __init(): command line argument: -a
[2009-05-27 21:37:33 +02:00] [____DEBUG] {bashinator.lib.0.sh:151}, __prepare(): successfully created temporary script subcommand logfile '/tmp
[2009-05-27 21:37:33 +02:00] [____DEBUG] {bashinator.lib.0.sh:159}, __prepare(): script subcommand logfile: '/tmp/example.log.8hGzmB'
[2009-05-27 21:37:33 +02:00] [____DEBUG] {example.lib.sh:76}, __main(): this is a debug test
[2009-05-27 21:37:33 +02:00] [_____INFO] {example.lib.sh:76}, __main(): this is a info test
[2009-05-27 21:37:33 +02:00] [___NOTICE] {example.lib.sh:76}, __main(): this is a notice test
[2009-05-27 21:37:33 +02:00] [__WARNING] {example.lib.sh:76}, __main(): this is a warning test
[2009-05-27 21:37:33 +02:00] [____ERROR] {example.lib.sh:76}, __main(): this is a err test
[2009-05-27 21:37:33 +02:00] [_CRITICAL] {example.lib.sh:76}, __main(): this is a crit test
[2009-05-27 21:37:33 +02:00] [____ALERT] {example.lib.sh:76}, __main(): this is a alert test
[2009-05-27 21:37:33 +02:00] [EMERGENCY] {example.lib.sh:76}, __main(): this is a emerg test
[2009-05-27 21:37:33 +02:00] [____DEBUG] {example.lib.sh:117}, exampleFunction(): fooArgument: foo
[2009-05-27 21:37:33 +02:00] [____DEBUG] {example.lib.sh:120}, exampleFunction(): barArgument: bar
[2009-05-27 21:37:33 +02:00] [_____INFO] {example.lib.sh:124}, exampleFunction(): this is an example function
[2009-05-27 21:37:33 +02:00] [____ALERT] {example.lib.sh:138}, bazFunction(): FATAL: dying for test purposes
[2009-05-27 21:37:34 +02:00] [____ALERT] {example.lib.sh:138}, bazFunction(): function call stack (most recent last):
[2009-05-27 21:37:34 +02:00] [____ALERT] {example.lib.sh:138}, bazFunction(): --> __dispatch('-a') called in 'example.sh' on line 71
[2009-05-27 21:37:34 +02:00] [____ALERT] {example.lib.sh:138}, bazFunction(): --> __main() called in 'bashinator.lib.0.sh' on line 115
[2009-05-27 21:37:34 +02:00] [____ALERT] {example.lib.sh:138}, bazFunction(): --> fooFunction('fooArgs') called in 'example.lib.sh' on line 85
[2009-05-27 21:37:34 +02:00] [____ALERT] {example.lib.sh:138}, bazFunction(): --> barFunction('barArgs') called in 'example.lib.sh' on line 130
[2009-05-27 21:37:34 +02:00] [____ALERT] {example.lib.sh:138}, bazFunction(): --> bazFunction('bazArgs') called in 'example.lib.sh' on line 134
[2009-05-27 21:37:34 +02:00] [____ALERT] {example.lib.sh:138}, bazFunction(): please check script subcommand log '/tmp/example.log.8hGzmB' for
[2009-05-27 21:37:34 +02:00] [____ALERT] {example.lib.sh:138}, bazFunction(): terminating script

--8<--[ start of script subcommand log (/tmp/example.log.8hGzmB) ]--8<--
rm: cannot remove `/does/not/exist': No such file or directory
mkdir: cannot create directory `/does/not/exist': No such file or directory
ls: cannot access /does/not/exist: No such file or directory
--8<--[ end of script subcommand log ]--8<--

--
sent by example.sh running on prometheus.bla.fasel.org at 2009-05-27 21:37:34
```

Stability



Simplicity



Security

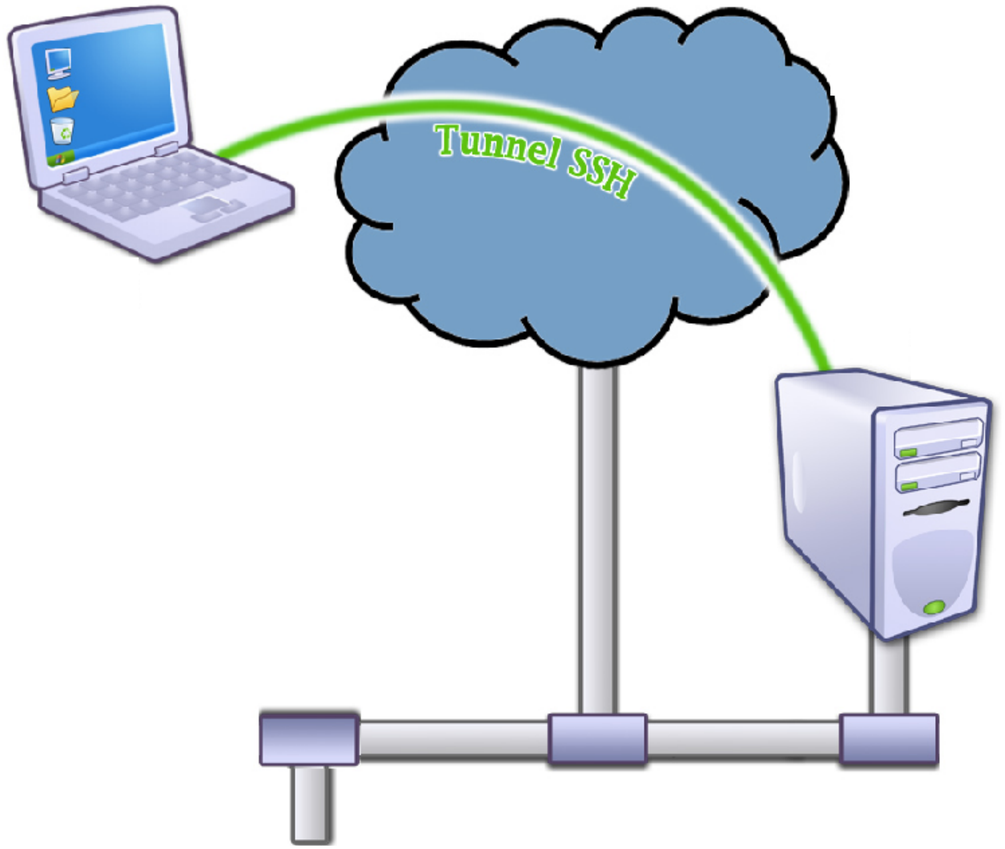
Free

learn

with Google™

SSH

access remote machine
copy files to remote machine



Run Process

know how to make configurations
run in background
keep logs
nohup

Manage Process

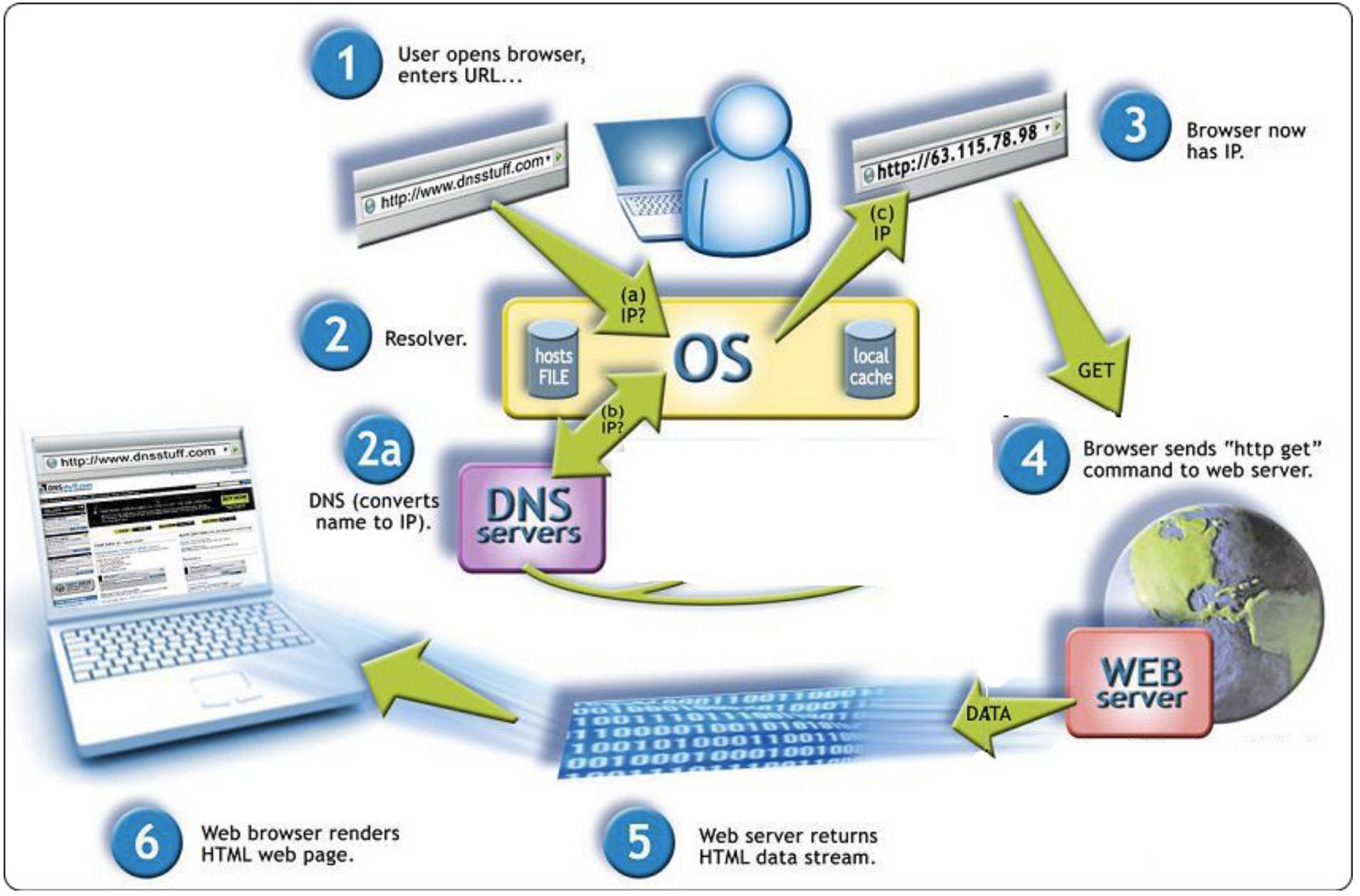
show running processes
terminate running processes
redirect port

Monitor the System

check memory
check disk
check port

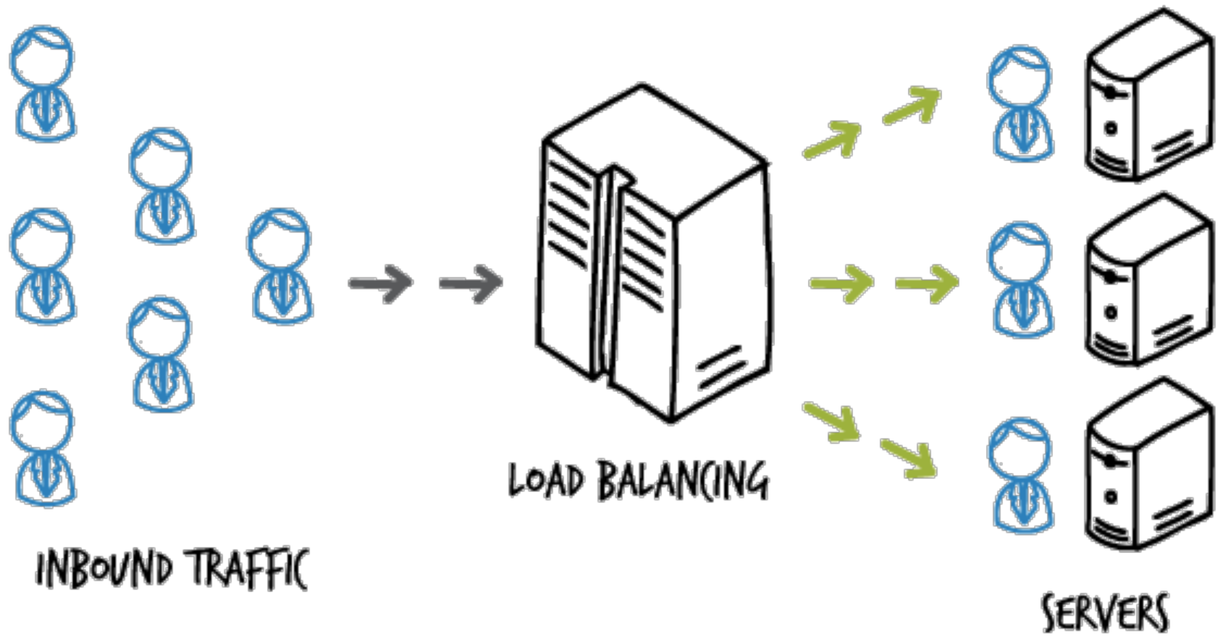
Domain Name

configure domain names (A, CNAME)
test domain names


1
User opens browser, enters URL...
http://www.dnsstuff.com
2
Resolver.
(a) IP?
hosts FILE
OS
local cache
(b) IP?
(c) IP
http://63.115.78.98
3
Browser now has IP.
GET
2a
DNS (converts name to IP).
DNS servers
4
Browser sends "http get" command to web server.
http://www.dnsstuff.com
WEB server
DATA
6
Web browser renders HTML web page.
5
Web server returns HTML data stream.

Load Balancer

configure load balancer



Something Cool or Something Useful?

LINUX