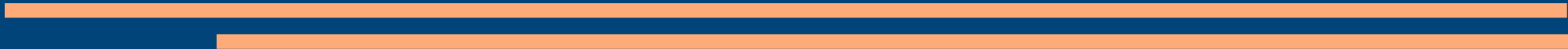
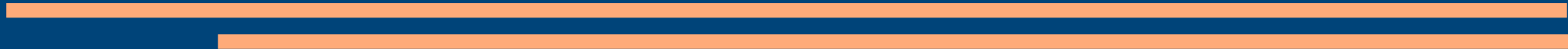


System Monitoring With Nagios

Monitoring Concepts and Nagios Configuration
Tutorial



Why Monitor?

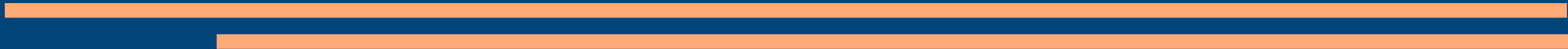


Part I: Monitoring Principles



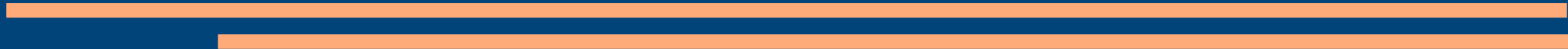
Types of Monitoring

- Environmental
- Network Performance
- Application Performance
- Network Device Status
- Server / System Status



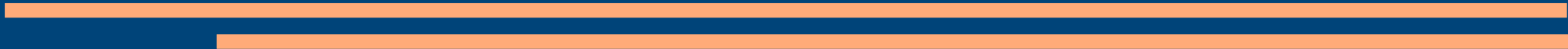
Monitoring Models

- Polling
 - Actively query devices to determine status
 - Schedule queries to minimize time between a failure and you knowing about the failure
- Listening
 - Devices tell you when something is wrong
- Hybrid



Thresholds

- Levels of Severity
 - Normal Operation
 - Warning
 - Critical
 - Off-line



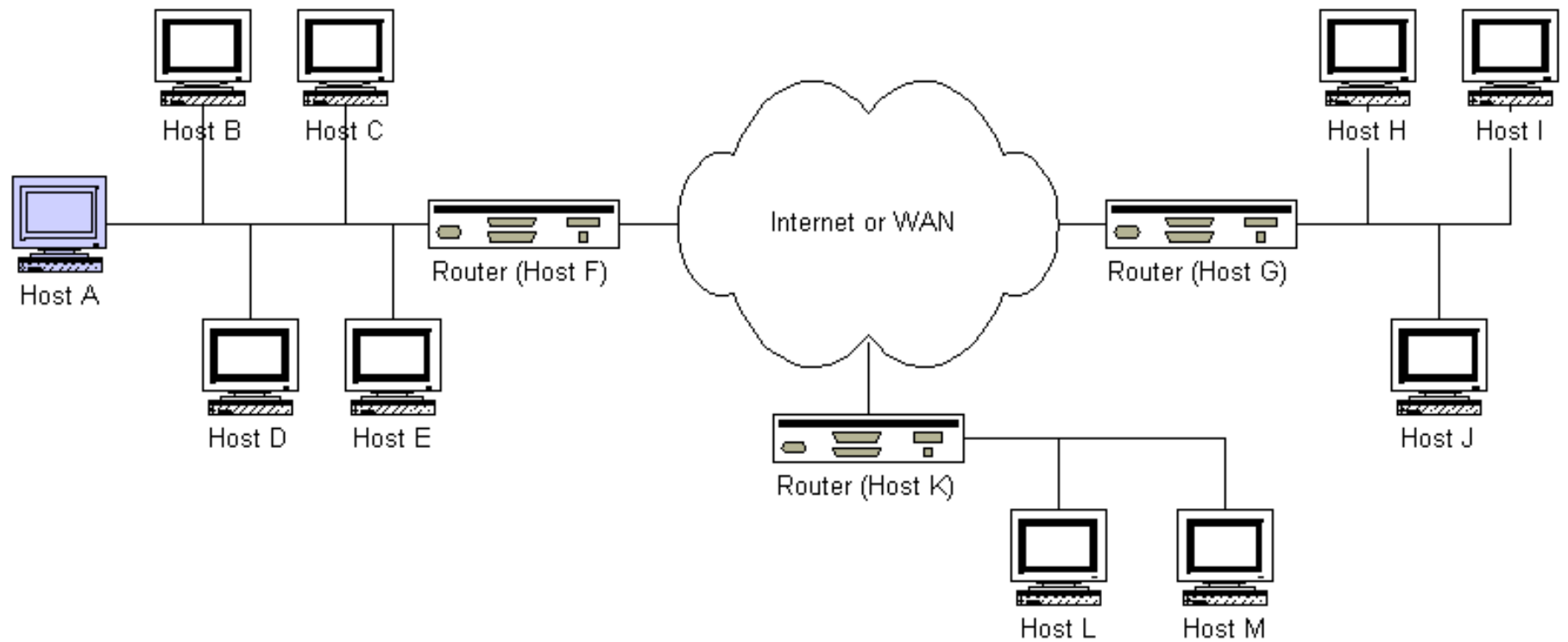
Intervals

- How many times do we try before declaring a host or service “dead”?
 - How often do we re-check the dead service?
 - How often do we check a normally-operating host or service?
 - How often do we send out notifications after a problem has occurred?
-
-

Notifications

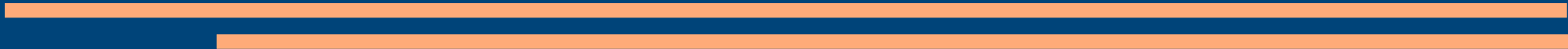
- Who gets notified?
 - How do they get notified?
 - Pager / SMS
 - Email
 - Phone call
 - Escalation
 - Send a message to somebody else if the problem isn't resolved
 - Automatic submission to trouble ticket system
-
-

Dependencies

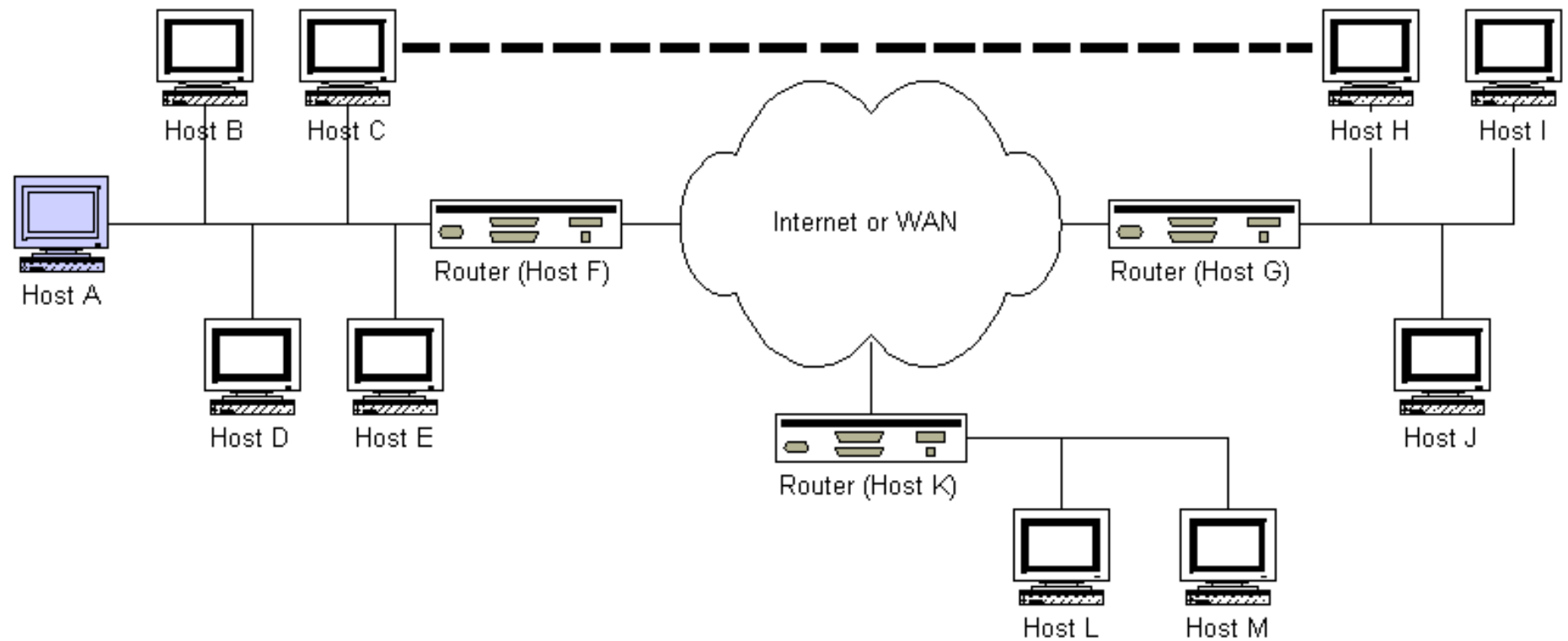


Dependencies

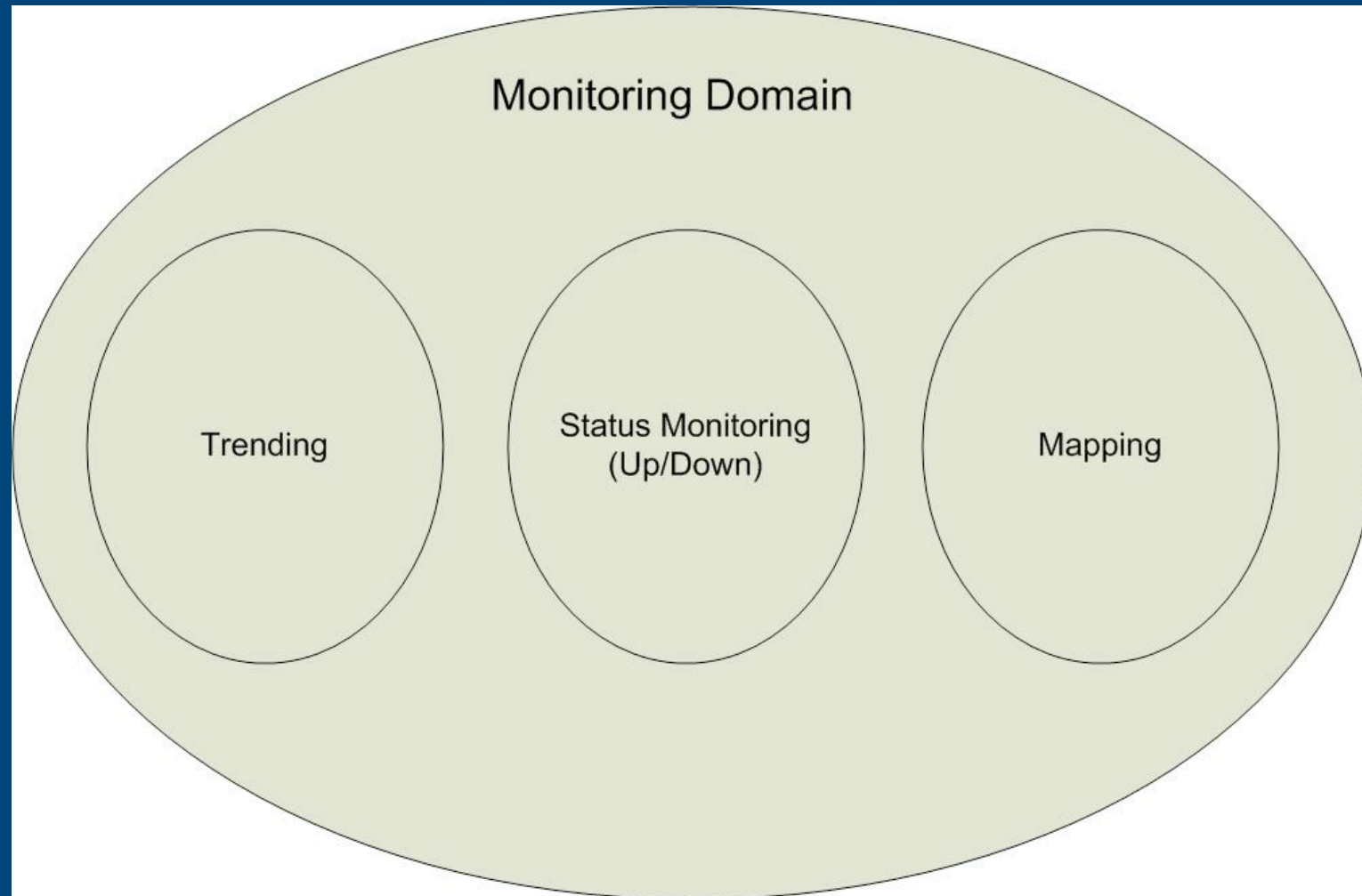
- Two Types
 - Reachability
 - One host's up/down status affects all communication with another host
 - Service Checking
 - A process or daemon on one machine uses the resources of another process on itself or on a different machine



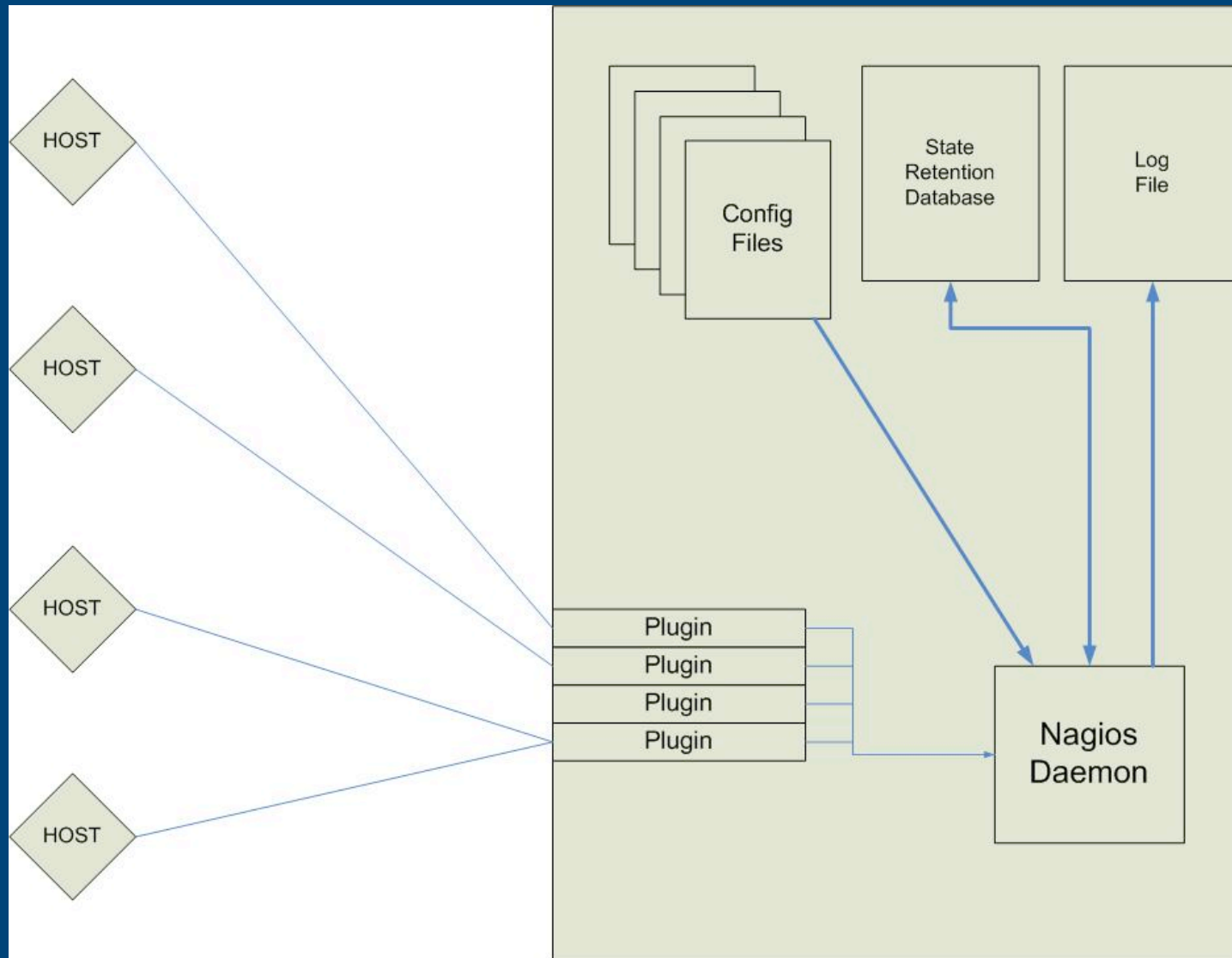
Dependencies



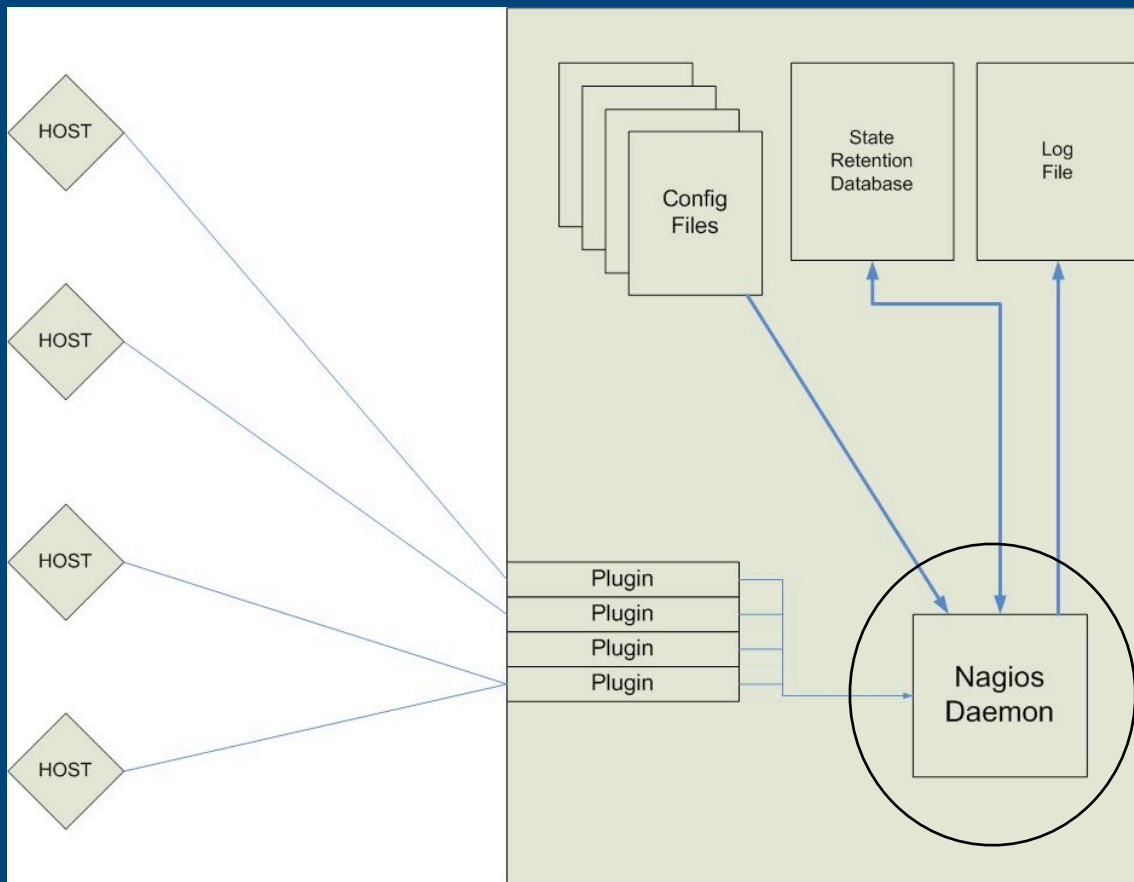
The “Monitoring Domain”



Part II: Nagios Architecture



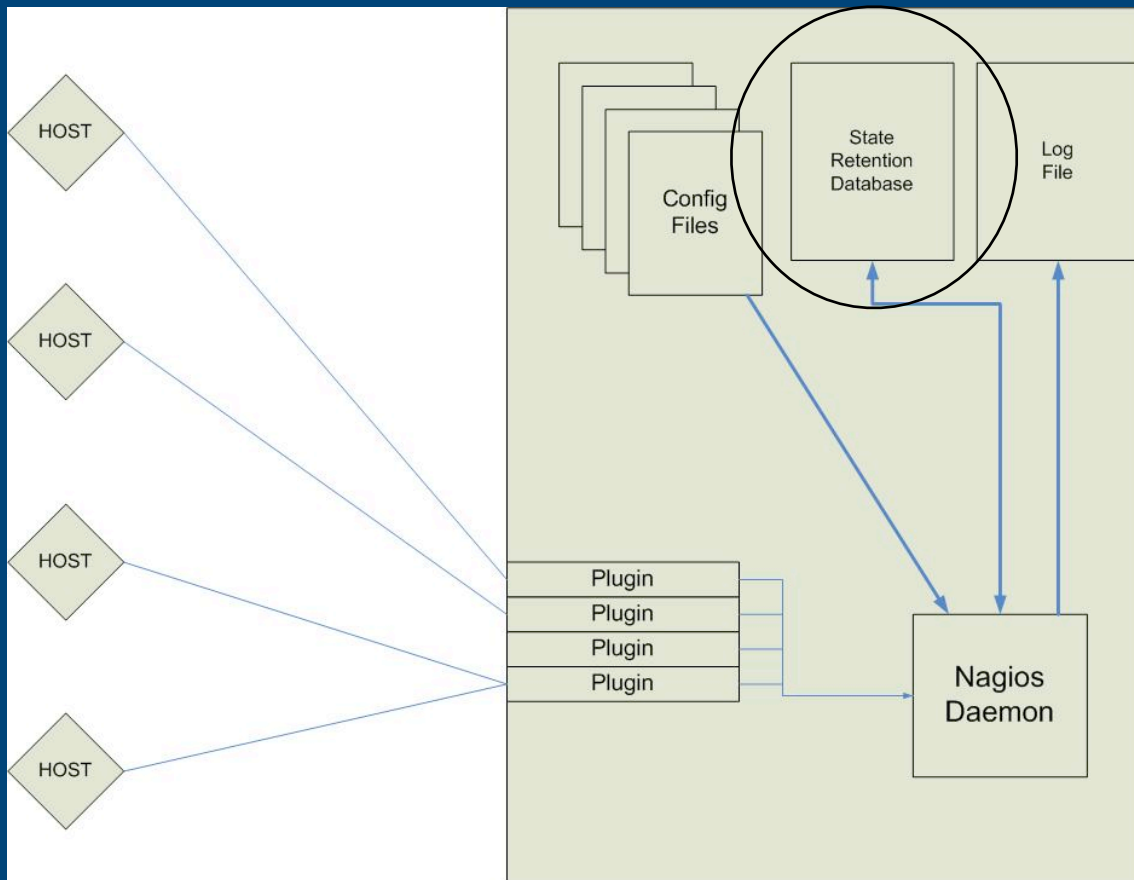
Nagios Architecture – Daemon



Schedules checks and processes results.

The daemon is separate from the service checks.

Nagios Architecture – State Retention



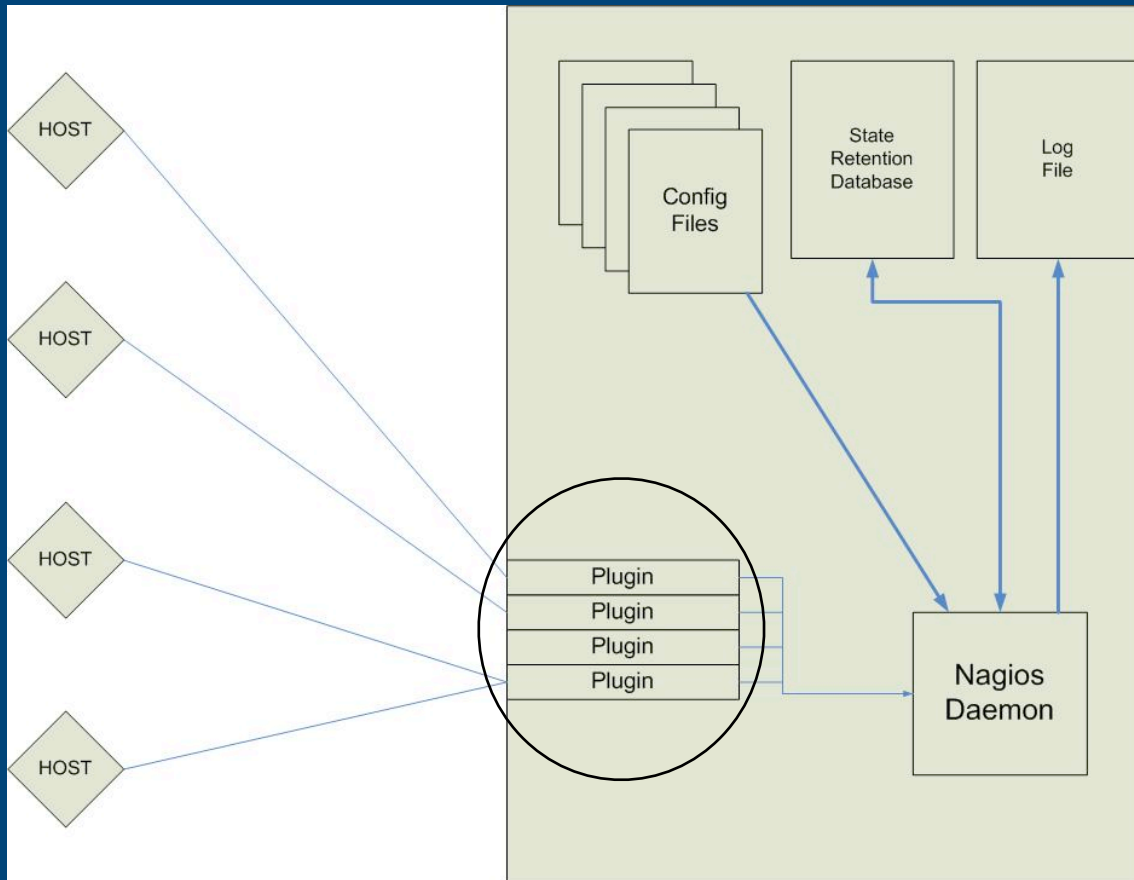
The daemon reports service and host states to the retention database when checks are executed.

Used to preserve status information across daemon restarts.

Nagios Architecture – The concept of “State”

- Soft State: Things might be broken, but we still need to make sure
 - Hard State: A host or service has been re-checked and is definitely dead.
 - Host States
 - Down
 - Unreachable
 - Recovery
 - Flapping
 - Service States
 - Warning
 - Unknown
 - Critical
 - Recovery
 - Flapping
-
-

Nagios Architecture – Plugins

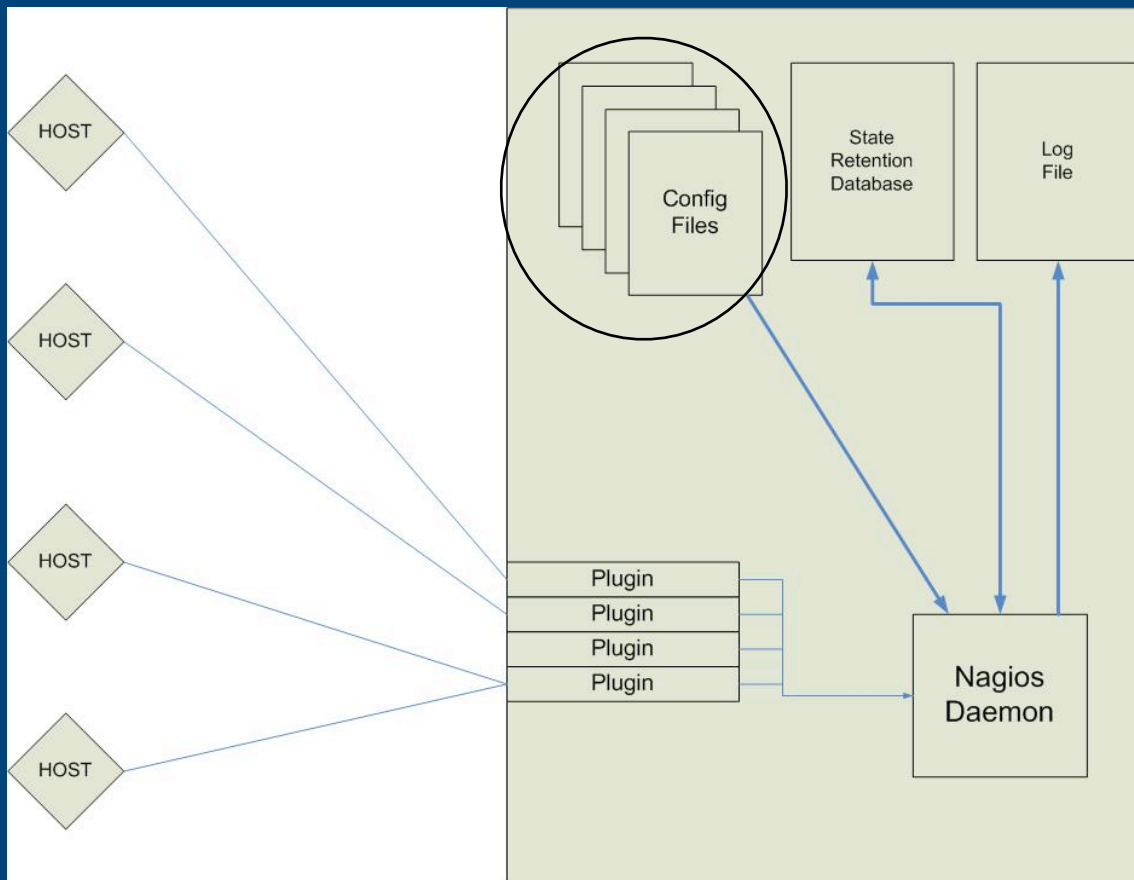


Performs the actual checks.

Can be any executable (script or compiled)

Compiled binaries provide the best performance.

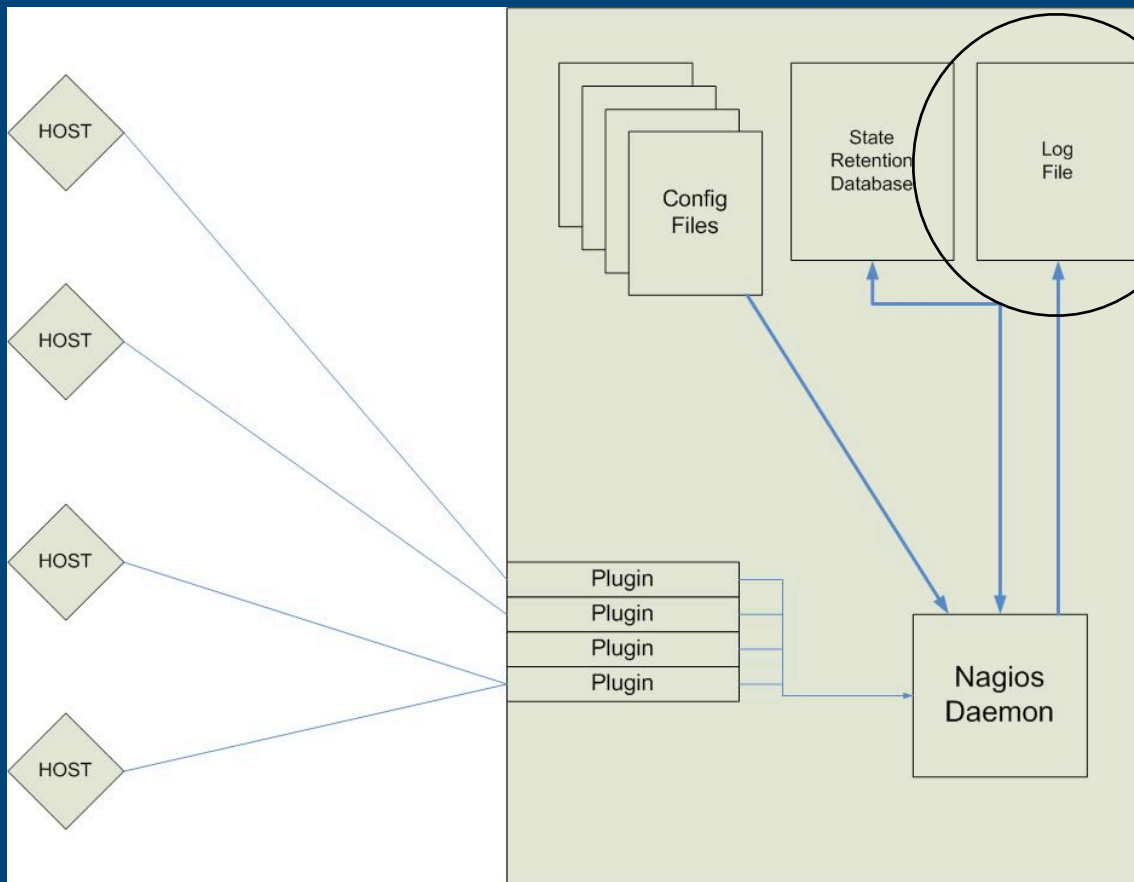
Nagios Architecture – Configuration Files



Plain text files

This is where we define what hosts and services will be checked.

Nagios Architecture – Logging



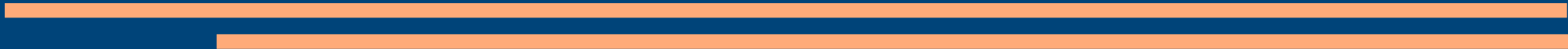
Results of checks are written here and to `/var/log/messages`.

`tail -f` this file to watch monitoring in real time.

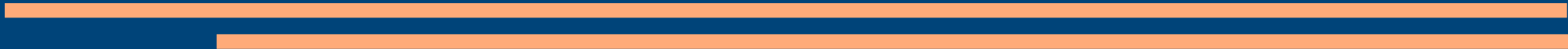
Nagios Architecture – Web Interface

Nagios can run without the web interface.

On most installations, the web interface is found at <http://yourserver/nagios/>

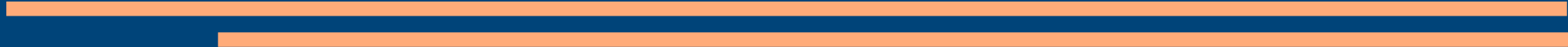


Part III: Nagios Configuration



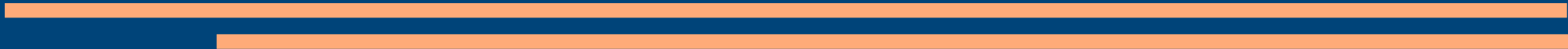
Configuration Files

- Required files
 - nagios.cfg – contains options for daemon behavior
 - cgi.cfg – controls the web interface
 - resource.cfg – tells nagios where to look for plugins
- Other files
 - Must be included in nagios.cfg using *cfg_file*
 - Can use as many as needed

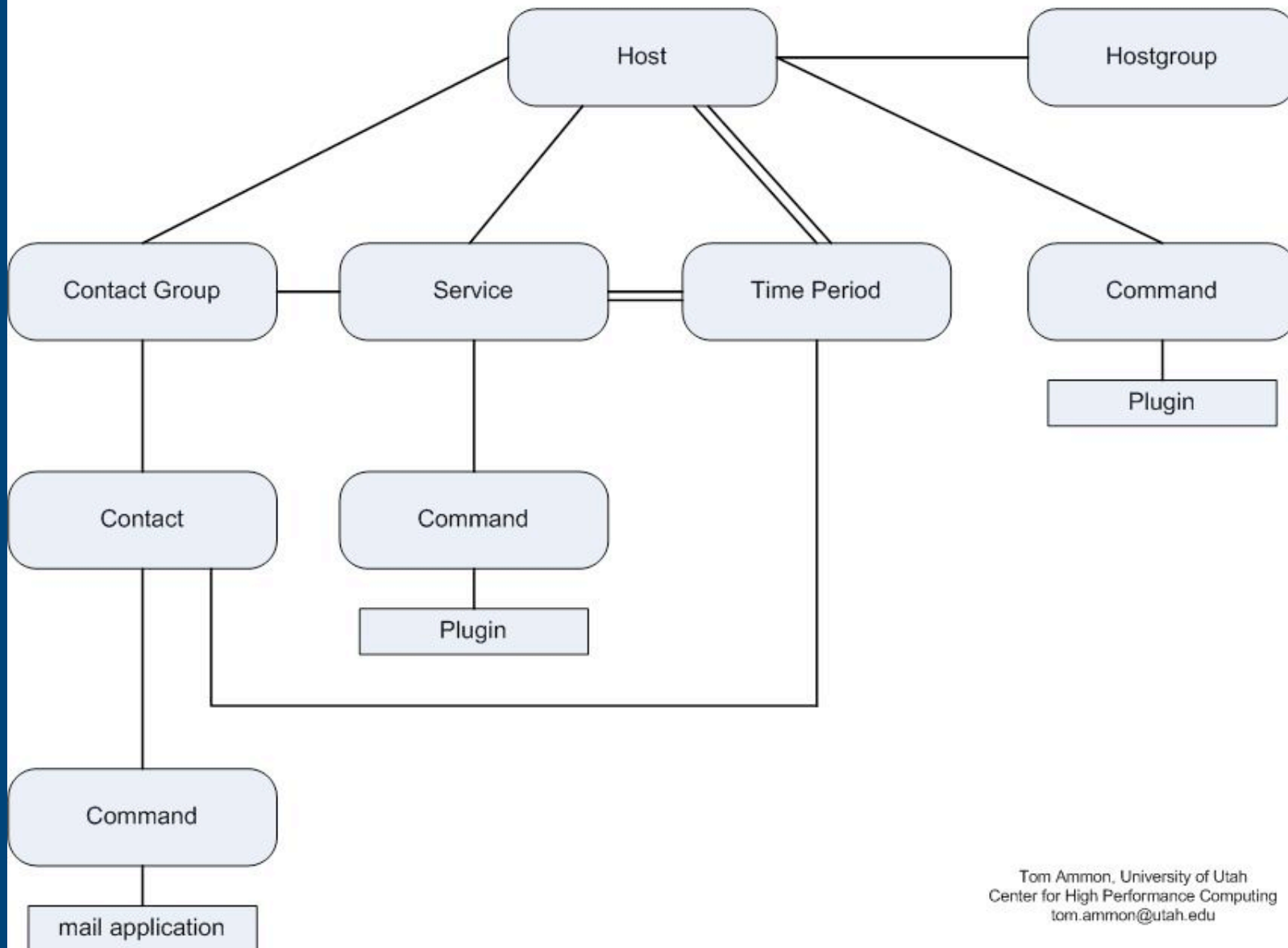


Structure and Syntax

- Required config file directives
- Object definition directives



Nagios Configuration Hierarchy

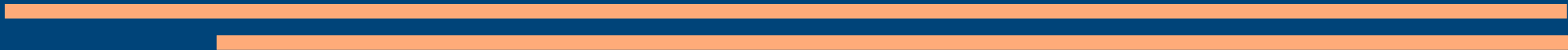


Hands-on: Configuring one host with one service

- Host and Service Object walk-through
 - Configuration steps:
 1. Create host
 2. Check configuration with *nagios -v nagios.cfg*
 3. Create the object that is missing
 4. Go back to #2 until no errors are reported
-
-

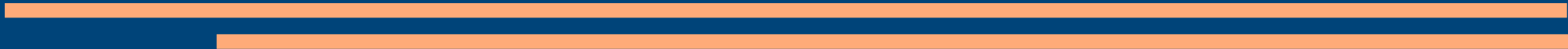
Make your life easier with Templates

- Templates look exactly like regular object definitions, with one exception:
 - The *register* directive

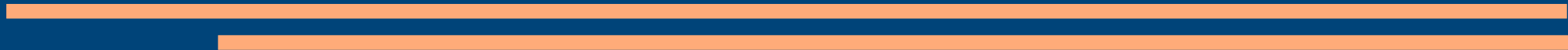


Hands-on: Configuration Using Templates

- Environmental
- Network Performance
- Application Performance
- Network Device Status
- Server / System Status



Part IV: Intro to Groundwork



Documentation

Installing Nagios

https://wiki.chpc.utah.edu/index.php/Nagios_Implementation

Migrating to Groundwork from Bare Nagios:

https://wiki.chpc.utah.edu/index.php/Groundwork_Configuration:_Building_CHPC_Monitoring_from_Nagios_CFG_Files

Nagios Web Site

<http://nagios.org>

Groundwork Web Site

<http://groundworkopensource.com>

Tom Ammon
Center for High Performance Computing

tom.ammon@utah.edu
801.674.9273