

Organize Services and Elements

In this module, you will work with Elements and service monitors, and create Element Groups and Service Groups for them, respectively.

On a practical level, doing these exercises will get you accustomed to using the up.time interface to create different types of groups, and assign service monitors to Elements. At a conceptual level, you will learn how inheritance occurs in groups in up.time, and gain an understanding about how you can focus on structuring your monitored inventory, while up.time takes care of tracking Element-level relationships.

This module consists of the following exercises:

Exercise	Description	Time required
Create a Service Monitor and Service Group	Learn about service monitors in up.time. Create one as the foundation to a Service Group. Learn how Service Groups work by linking one to all of your Elements in one step.	1 slice
Create Element Groups	Begin organizing your monitored inventory by creating an Element Group and a pair of child Element Groups.	1 slice
Learn About Inheritance	Create a new Service Group (including a service monitor) and assign it to a top-level Element Group. Examine the services of an Element in a child group to learn about inheritance.	1 slice



Before You Begin

In the first module, if you followed all three tracks, your inventory should now include a VMware vCenter Server (along with its inventory of ESX hosts and VMs), a pair of physical servers (one agent-based Linux system, and a WMI-based Windows system), and an SNMP network device. The screenshots used in this module reflect this.

For the exercises in this module, the two physical servers are used as examples. If you did not add physical servers to your inventory, you can either just follow along, or consider [adding physical servers](#) so that you can do every exercise.

Create a Service Monitor and Service Group

As a default way to report server uptime, for every server-type Element that is added to up.time's inventory, a Ping service monitor is also created and assigned to it, in a one-to-one relationship. In this exercise, we will replicate this functionality, but instead using a single service monitor. We will be able to create a one-to-many relationship between a service monitor and all of your Elements using a *Service Group*. A Service Group is a group of service monitors that can be assigned to Elements or groups of Elements.

Add a Service Monitor

1. Click **Services**, then click **Add Service Monitor** in the left pane.

2. In the **Add Service Monitor** pop-up, select the **Ping** service monitor, and click **Continue**.

Add Service - Page 1 of 2
Select a service monitor to start: Want more? Search for monitors.

Operating System Monitors

- ☐ File System Capacity
- ☐ NFS Mount Performance Monitor
- ☐ Performance Check
- ☐ Process Count Check
- ☐ Windows Event Log Scanner
- ☐ Windows Service Check

Related Monitors:

- ☐ Live Splunk Listener
- ☒ Ping Monitor the responsiveness of your servers and network devices.
- ☐ VM Instance Performance
- ☐ vSphere
- ☐ Splunk Query

Applications - All Types

Applications - Databases

Applications - Email

Applications - General

Applications - Web Services

Domain Services

End User Experience Monitors

Network Device Monitors

Network Service Monitors

Storage Monitors

VMware Monitors

Advanced and Script Monitors

Cancel Continue...

3. Provide a **Service Name** of pingTest, and make it **Unassigned**.

Add Service - Ping - Page 2 of 2

Service Name:

Description:

Host:

- ☐ Single System
- ☐ Service Group
- ☒ Unassigned

Ping Settings

Number to send:

Save All for Graphing ☐

Note that this service monitor, as it's being created, can also be assigned to an individual Element (that is, a single system), or a Service Group; let's leave it assigned to nothing for now.

4. For the sake of completion, let's configure a pair of alert thresholds for the service monitor's **Average Round Trip Time**:
- warning > 30 ms
 - critical > 50 ms

Ping Settings

Save All for Graphing ☐

Number to send:

Average Round Trip Time Save for Graphing ☐

Warning: ms

Critical: ms

Percent Loss Save for Graphing ☐

Warning: %

Critical: %

Response time Save for Graphing ☐

Warning: ms

Critical: ms

Timing Settings

5. Click **Finish**, then click **Close Window**.

In the main UI window, the **Info** page for your newly created service monitor is displayed. Note there is no value in the **Host** field, showing that it is not assigned to any Element.

Create a Service Group

Let's now create a Service Group that includes the service monitor, and is assigned to the **My Infrastructure** Element Group.

1. In the left pane, click **Add Service Group**.
2. In the **Add Service Group** pop-up window, confirm the group type is **Regular**, then click **Continue**.
3. Give the Service Group a name. In this example, we'll call this group `pingable`:

Service Monitors

Systems

Service Group

Up.time gives you the ability to group multiple service monitors in to a single Service Group making it easier to assign this group of services to many systems.

Add Regular Service Group - Page 2 of 2

Name of Service Group:

Description:

Available Services:

Selected Services:

Available Element Groups:

Selected Element Groups: ☒ Include subgroups

Available Elements:

Selected Elements:

Buttons: Add >, < Remove, Add All >>, << Remove All

Buttons: Add >, < Remove, Add All >>, << Remove All

Buttons: Add >, < Remove, Add All >>, << Remove All

Buttons: Cancel, Finish

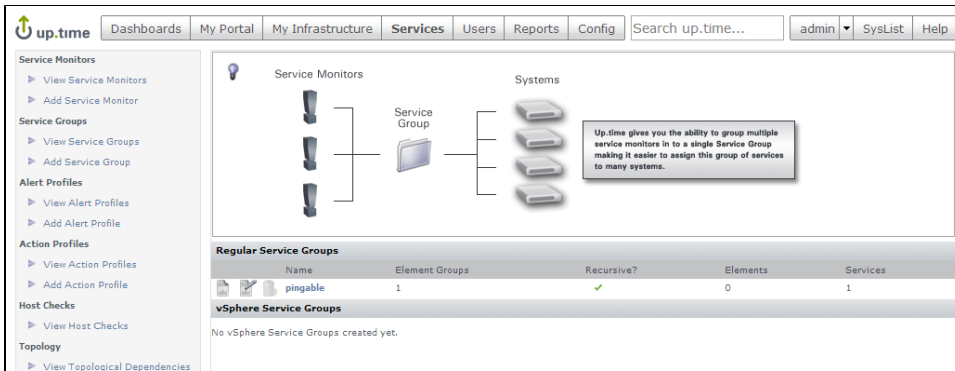
4. Configure the Service Group to include the `pingTest` service monitor you created earlier in this exercise, and associate it with the **My Infrastructure** Element Group.



FYI

My Infrastructure is always available as an Element Group, as it represents all of your monitored inventory (or, Elements), as shown when you view it in the **My Infrastructure** panel. The other two groups shown in this example (**Discovered Hosts**, and **Discovered Virtual Machines**) were automatically created when you add a VMware vCenter Server as an Element.

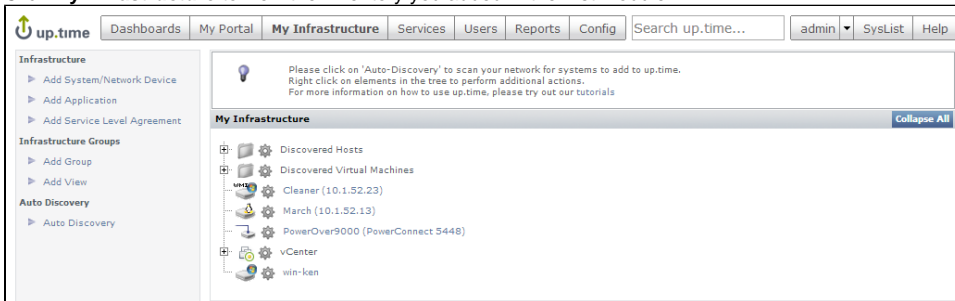
5. Click **Finish**, then click **Close Window**.
In the main UI window, you are taken to your list of Service Groups, of which you now have one:



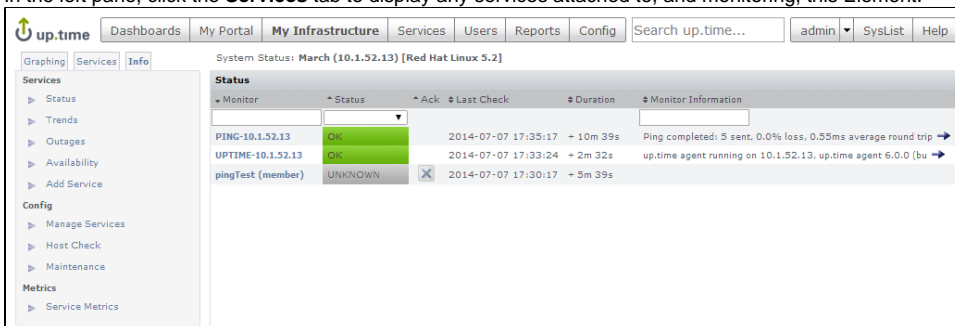
The list provides a summary of each Service Group. In our case, our group is associated with a single Element Group (and any of its subgroups); it is not directly assigned to any individual Elements; it contains a single service monitor.

Validation

1. Click **My Infrastructure** to view the inventory you added in the first module:



2. Click any Element's gear icon, then in the pop-up menu, click **View** to display its **General Information** page. Since you have associated the Service Group to all Elements (by way of the **My Infrastructure** Element Group), the service monitor will be attached to any Element.
3. In the left pane, click the **Services** tab to display any services attached to, and monitoring, this Element.



You can see that by creating a Service Group (**pingable**), and linking it to an Element Group to which this Element belongs (**My Infrastructure**), this Element now uses this service monitor. Some things to note using the above screenshot:

- the **pingTest** service monitor has a (**member**) suffix to indicate it is "attached" to this Element via a Service Group, meaning the properties of this service monitor can only be edited at the Service Group level (it is possible to "detach" a service monitor to make it editable at an Element level, but this is not a best practice)
- the **pingTest** service monitor status may be shown as **UNKNOWN** because it has recently been added, and a full monitoring interval has not yet passed
- to view how long a monitoring interval is for a particular Element, click its **Info** tab, then look in the **Data Collection** section of the **General Information** page

Bonus Knowledge

While we are viewing this Element's **Status** page (if you have clicked the **Info** tab, return to the **Status** page by clicking the **Services** tab), let's learn a few more things about service monitors and Elements using the example screenshot above, which is of a server-type Element with an up.time agent installed on it:

- as with server-type Elements, upon addition to the up.time inventory, a Ping service monitor was created and assigned to the Element called **PING-`<hostname>`**
- server-type Elements that have the up.time agent installed will also have an **UPTIME-`<hostname>`** service monitor assigned to it
- if you click **Host Check** in the left pane, you can see which service up.time is using to monitor that particular Element's status (which applies to most Elements except for VMware vCenter Servers)
- if you click **Manage Services**, you will see other service monitors attached to the Element that aren't necessarily related to status
 - **Configuration Update Gatherer**: collects any configuration changes for the monitored Element every 24 hours
 - **Platform Performance Gatherer**: collects basic performance metrics from the monitored Element using the up.time Agent or WMI, and feeds them into up.time; its metrics can be used with a performance-related service monitor such as Performance Check

Now that you've gone through a rudimentary exercise of creating a service monitor, a Service Group, and assigning them to the **My Infrastructure** Element group, we'll learn more about their properties.

Create Element Groups

As you add Elements to up.time, by default, they end up at the top of the **My Infrastructure** hierarchy. Unless you are monitoring a small number of Elements, it's best practice to keep your monitored inventory well organized. Doing so helps both administrators and end users, which we will see in this exercise.

How you organize your Elements may depend on the policies that determine other aspects of your IT infrastructure, such as naming conventions for hosts. You may want to divide your inventory primarily by platform (for example, you could have Linux, Solaris, and Windows top-level groups); you may want to divide by geographical location; or you could organize first by Element function (for example, Production versus QA/UAT). In our example, we'll organize our Elements by function, then by platform.

1. Click **My Infrastructure**, then click **Add Group** in the left pane.
2. Provide a **Group Name** of `Production`, leaving the other configuration options empty, or at the default value:

With up.time, you can create nested groups. This allows better categorizing of systems by geography or function. For example, you can create a group called Datacenters and then two subgroups called Production and Disaster Recovery. Each group then can be assigned all the elements that belong to it, along with the appropriate up.time users who are allowed to view those systems.

Groups

Group Name

Production

Group Description

Parent Group

My Infrastructure

Available Groups

Discovered Hosts
Discovered Virtual Machines

Add >

< Remove

Add All >>

<< Remove All

Selected Groups

Available Elements

build-seleniummw (build-seleniummw.rd.local)
build-sol10-x86
build-sonar
build-vmstudio-26 (build-vmstudio-26.rd.local)
Cleaner (10.1.52.23)
dev-sles112-x64
exch-ad01 (exch-ad01.rd1.local)
exch-ad02 (EXCH-AD02.rd1.local)
lab-novell65
lab-novell65ssl
linux-rh63-vm

Add >

< Remove

Add All >>

<< Remove All

Selected Elements

Available User Groups

SysAdmin User Group

Add >

< Remove

Add All >>

<< Remove All


Selected User Groups

Cancel

Save

3. Click **Save**, then click **Add Another**.

4. Provide a **Group Name** of `Windows Servers`, and make the **Parent Group** the previously created **Production** group:

 With up.time, you can create nested groups. This allows better categorizing of systems by geography or function. For example, you can create a group called `Datacenters` and then two subgroups called `Production` and `Disaster Recovery`. Each group then can be assigned all the elements that belong to it, along with the appropriate up.time users who are allowed to view those systems.

Groups

Group Name

Group Description

Parent Group

Available Groups

Discovered Hosts
Discovered Virtual Machines
Production

Selected Groups

Available Elements

build-seleniumw (build-seleniumw.rd.local)
build-sol10-x86
build-sonar
build-vmstudio-26 (build-vmstudio-26.rd.local)
dev-sles112-x64
exch-ad01 (exch-ad01.rd1.local)
exch-ad02 (EXCH-AD02.rd1.local)
lab-novell65
lab-novell65ssl
linux-rh63-vm
March (build-bamboo)

Selected Elements

Cleaner (10.1.52.23)

Available User Groups

SysAdmin User Group

Selected User Groups

5. From the **Available Elements** list, locate and add your Windows Element.



Note

As mentioned at the beginning of this module, it is assumed you have Windows and Linux server Elements in your inventory. If you don't, you can either follow along, substitute these examples with something similar in your current test inventory, or go back and add these types of Elements.

6. Click **Save**, then click **Add Another**.

7. Provide a **Group Name** of Linux Servers, and again make **Production** the **Parent Group**:

With up.time, you can create nested groups. This allows better categorizing of systems by geography or function. For example, you can create a group called Datacenters and then two subgroups called Production and Disaster Recovery. Each group then can be assigned all the elements that belong to it, along with the appropriate up.time users who are allowed to view those systems.

Groups

Group Name:

Group Description:

Parent Group:

Available Groups

- Discovered Hosts
- Discovered Virtual Machines
- Production
- Windows Servers

Selected Groups

Available Elements

- build-seleniummw (build-seleniummw.rd.local)
- build-sol10-x86
- build-sonar
- build-vmstudio-26 (build-vmstudio-26.rd.local)
- Cleaner (10.1.52.23)
- dev-sles112-x64
- exch-ad01 (exch-ad01.rd1.local)
- exch-ad02 (EXCH-AD02.rd1.local)
- lab-novell65
- lab-novell65ssl
- linux-rh63-vm

Selected Elements

Available User Groups

- SysAdmin User Group

Selected User Groups

Buttons: Add >, < Remove, Add All >>, << Remove All

Buttons: Add >, < Remove, Add All >>, << Remove All

Buttons: Add >, < Remove, Add All >>, << Remove All

Buttons: Cancel, Save

8. Click **Save**, then click **Close Window**.

Notice how we did not add your Linux server to the **Linux Servers** Element group as we were creating it. You can also assign an Element to an Element Group from the perspective of the Element itself.

9. In the main UI window, locate your Linux server in **My Infrastructure**.

10. Click its gear icon, then click **Edit**.

11. In the pop-up window, change the Element's **Parent Group** from **My Infrastructure**, to **Linux Servers**:

Parent Group:

Custom Field 1:

Custom Field 2:

Custom Field 3:

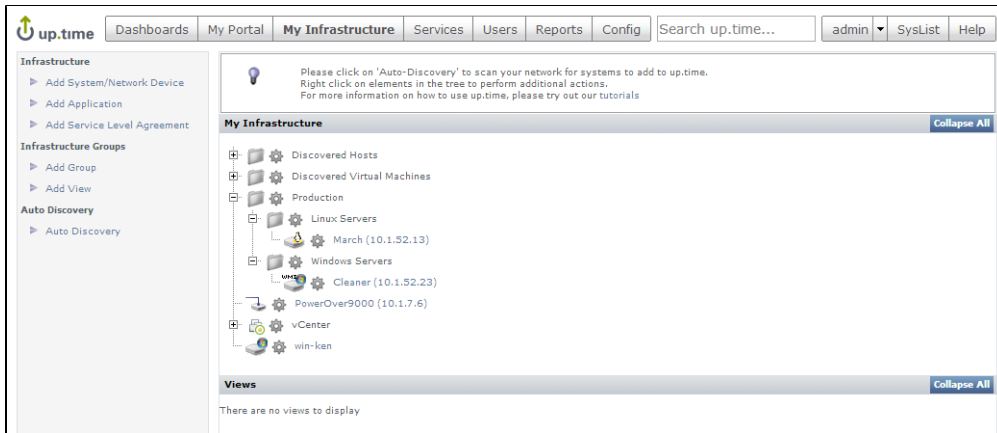
Custom Field 4:

Buttons: Add >, < Remove, Add All >>, << Remove All

12. Click **Save**.

Validation (My Infrastructure)

After adding the Element Groups, in the main UI window, **My Infrastructure** is displayed. Click to expand the **Production** group, and the **Windows Servers** and **Linux Servers** child groups. Each of these child groups contain one Element.



Structuring your monitored inventory in **My Infrastructure** not only facilitates Element management for system administrators, but, as the structures are reflected in dashboards, can also be useful for end users.

Pro Tip

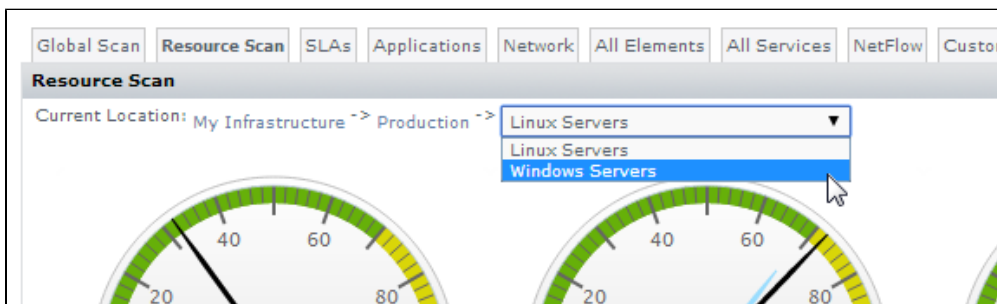
Elements are not just servers, network devices, and their virtual instances; they can also be Applications and SLAs. Although based on metrics retrieved from monitored hardware (whether physical or virtual), Applications and SLAs provide health and performance information from a business perspective. Nonetheless, they are also found under **My Infrastructure**, and can be placed in their own Element Groups. Refer to the product documentation for more information about [SLAs](#), and [Applications](#) and the [Applications dashboard](#).

Validation (Resource Scan)

Click **Dashboards**, then click the **Resource Scan** tab. This dashboard summarizes resource usage for server-type Elements from various points in your hierarchy of Elements: you can click an individual server to show its usage data, or Element Groups to show an average of all its Elements. Your **Production** group is here, showing an average of the data for the servers you added to its platform-specific child groups. Click the parent group to drill down and display the **Windows Servers** and **Linux Servers** child groups. Each child group will now show usage data for its respective Element. Click either child group to display its contents, which in our case is a single server that you added:

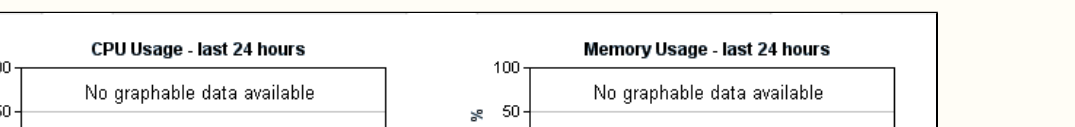


When you are viewing data for one child group, you can move laterally to display a sibling child group by using the drop-down at the end of the breadcrumb trail at the top of the dashboard:



Note

In the sample images above, the gauges and performance graphs are showing more information than you may be seeing on your own Monitoring Station. A data-collection interval is required to have passed before any information is shown; until then, you may see empty graphs such as the following:



The image displays four empty performance graphs arranged in a 2x2 grid. Each graph has a title, a y-axis labeled with a percentage sign and values 0, 50, and 100, and a central text box that reads 'No graphable data available'. The titles are: 'CPU Usage - last 24 hours', 'Memory Usage - last 24 hours', 'Disk Busy - last 24 hours', and 'Disk Usage - last 24 hours'.

In the first exercise, you created a basic Service Group and assigned it to all of your Elements via the **My Infrastructure** Element Group. Let's create another Service Group that demonstrates inheritance. To do this, we will make use of the **Production** parent Element Group we created in the previous exercise. But first, let's create a service monitor that will be a part of the Service Group. Since it will be assigned to Elements in the **Production** Element group, this service monitor should be something appropriate for all servers in a production environment.

- Add Service - Page 1 of 2

Select a service monitor to start:

Want more? Search for monitors.

Operating System Monitors

☐ File System Capacity

☐ NFS Mount Performance Monitor

☒ Performance Check

Monitor key OS performance metrics like CPU (per process or total), Memory, Network I/O and Disk I/O.

Windows Service Check

Related Monitors:

☐ Live Splunk Listener

☐ Ping

☐ Splunk Query

☐ VM Instance Performance

☐ vSphere ESX Server Performance

Applications - All Types

Applications - Databases

Applications - Email

Applications - General

Applications - Web Services

Domain Services

End User Experience Monitors

Network Device Monitors

Network Service Monitors

Storage Monitors

VMware Monitors

Advanced and Script Monitors

Cancel

Continue...

3. Provide a **Service Name** of Server Performance (Prod), and leave it **Unassigned** (we are going to create a Service Group for it):

Add Service - Performance Check - Page 2 of 2

Service Name:

Description:

Host: ☐ Single System ☐ Service Group ☒ Unassigned

Performance Check Settings Save All for Graphing ☐

Time Interval

Sustained Average Time Interval: min

CPU Check

CPU Value:

CPU Warning Threshold: %

CPU Critical Threshold: %

Run Queue Check (run queue length per CPU)

Run Queue Warning Threshold: num/cpu

Run Queue Critical Threshold: num/cpu

Memory Check (percent memory used greater than)

Used Memory Warning Threshold: %

Used Memory Critical Threshold: %

Swap Check (percent swap used greater than)

Used Swap Warning Threshold: %

Used Swap Critical Threshold: %

Disk I/O Check

Disk to Check: ☐ Individual disks ☒ Average across all disks

Disk Value:

Disk I/O Warning Threshold:

Disk I/O Critical Threshold:

Network I/O Check

Interface to Check: ☒ Individual interfaces ☐ Average across all interfaces

Network Value:

4. For the sake of the exercise, continue to configure the service monitor with some thresholds, similar to those shown above. For this module and guide, we won't be making use of any performance checks you define, but feel free to experiment. You can always enter some unrealistically strict thresholds to force alerts, just to see how they appear on the dashboards.
5. Click **Finish**, then click **Close Window**. Now that we have created a common performance check intended for all your production servers, let's do what we did in the first exercise of this module, and create a Service Group that will include it.
6. Click **Services**, then click **Add Service Group** in the left pane.
7. in the **Add Service Group** pop-up window, confirm the group type is **Regular**, then click **Continue**.

8. Configure the service group as follows:

Add Regular Service Group - Page 2 of 2

Name of Service Group:

Description:

Available Services

unassigned

pingTest

Selected Services

Server Performance (Prod)

Available Element Groups

Discovered Hosts
Discovered Virtual Machines
Linux Servers
My Infrastructure
Windows Servers

Selected Element Groups ☒ Include subgroups

Production

Available Elements

Discovered Hosts

vesxi51-00.rd.local
vesxi51-01.rd.local
vmh-rd10.rd.local
vmh-rd11.rd.local
vmh-rd12.rd.local
vmh-rd13.rd.local
vmh-rd14.rd.local
vmh-rd15.rd.local
vmh-rd3.rd.local
vmh-rd6.rd.local
vmh-rd7.rd.local

Selected Elements

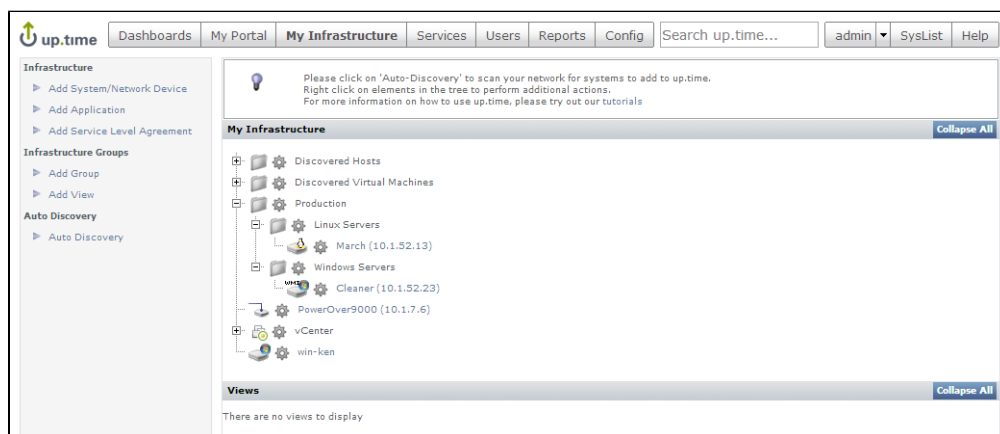
Cancel Finish

- Provide a logical group name such as **Server Availability/Performance (Prod)**
- From the **unassigned** group of **Available Services**, add the **Server Performance (Prod)** service monitor you created earlier in this exercise.
- Select, from the **Available Element Groups**, the **Production** Element Group that you created in the last exercise.
- Ensure the **include subgroups** check box has been selected.

9. Click **Finish**, then click **Close Window**.

Validation

Click **My Infrastructure** to see your hierarchy:



Remember, referring to the structure presented in the example above, that you directly assigned the Service Group to the **Production** Element Group, and you chose to include subgroups. Click one of the Elements in either the Linux or Windows child Element Group. When that Element's **General Information** page is displayed, click the **Services** tab, then **Manage Services**.

up.time

Dashboards

My Portal

My Infrastructure

Services

Users

Reports

Config

Search up.time...

admin

SysList

Help

Graphing

Services

Info

Services

▶ Status

▶ Trends

▶ Outages

▶ Availability

Config

▶ Add Service

▶ Manage Services

▶ Host Check

▶ Maintenance

Graphing

▶ Service Metrics

Status: March [Red Hat Linux 5.3]

Status

▼ Monitor

▼

▲ Status

▲ Ack

▲ Last Check

▲ Duration

▲ Monitor Information

PING-build-bamboo

OK

2014-08-18 15:58:50

+ 10 days 23h

Ping completed: 5 sent, 0.0% loss, 0.41ms average r

pingTest (member)

OK

2014-08-18 15:56:09

+ 10 days 5h

Ping completed: 5 sent, 0.0% loss, 0.68ms average r

UPTIME-build-bamboo

OK

2014-08-18 15:59:02

+ 10 days 23h

up.time agent running on build-bamboo, up.time agent

Server Performance (Prod) (member)

UNKNOWN

+ 33s

Note that the "**Server Performance (Prod)**" service monitor is now attached to this Element by way of the Service Group. In the above example, we are showing the Linux server. The service monitor will also now be attached to the Windows server:

up.time

Dashboards

My Portal

My Infrastructure

Services

Users

Reports

Config

Search up.time...

admin

SysList

Help

Graphing

Services

Info

Services

▶ Status

▶ Trends

▶ Outages

▶ Availability

Config

▶ Add Service

▶ Manage Services

▶ Host Check

▶ Maintenance

Graphing

▶ Service Metrics

Status: Cleaner [Windows Server 2008 R2 Enterprise]

Status

▼ Monitor

▼

▲ Status

▲ Ack

▲ Last Check

▲ Duration

▲ Monitor Information

PING-build-w2k8-x64.rd.local

OK

2014-08-18 15:56:50

+ 10 days 23h

Ping completed: 5 sent, 0.0% loss, 0.49ms average r

pingTest (member)

OK

2014-08-18 15:53:49

+ 10 days 5h

Ping completed: 5 sent, 0.0% loss, 0.92ms average r

Server Performance (Prod) (member)

UNKNOWN

+ 50s

Additionally, whenever an Element is created in, or moved to, the **Production** Element Group (and, as configured, to either of its subgroups), that Element will inherit whichever service monitors are assigned to the group.

Extrapolating from this example, you could create a battery of service monitors that act as performance and health checks for all production servers. These service monitors would be added to the **Server Performance (Prod)** service group that is associated with the top-level **Production** Element Group. You could then create platform-specific health checks, and add them to appropriate Linux- and Windows-specific Service Groups (which you would need to create). These Service Groups could then respectively be associated with the existing Linux Servers and Windows Servers Element Groups.

Once these Service Group and Element Group relationships have been established, if you created more child Element Groups, the respective Elements they contain would inherit the appropriate service monitors. (For example, adding a "Solaris" Element Group as a child of the **Production** group, and sibling of the Linux and Windows groups means the Solaris servers will inherit all of the general health check service monitors; adding a "Databases" Element Group as a child of the **Windows Servers** group means the database servers will inherit all the Windows-specific health check monitors.)

The important thing to note is it's more efficient to manage not at the Element level, but at the object level, whether that object is a Service Group, Element Group, or other up.time construct you will learn about; if you focus on Element Groups and Service Groups, everything else lines up and falls into place. Well managed arrangements of Element Groups and Service Groups can result in powerful cascading configurations.

Back: [Tour the Interface](#)

Next: [Organize Users and Views](#)