

Installing and Configuring Icinga 2

Imaging Team @ Roche Tissue Diagnostics



Getting Started	3
Icinga	3
Graphite	4
Grafana	4
Installations	5
Installing Child: Windows Servers	5
Connecting to Server	5
Running Installer	5
Generating Ticket	6
Configuring Parent	7
Installing Child: CentOS Servers	9
Generating Ticket	9
Installing Icinga	9
Configuring Parent	11
Configurations	12
Icinga	12
Services	12
Commands	13
Emailing	13
Grafana	13
Panels	13
Dashboards	14
Playlists	15
References	16
LinuxTechi Tutorials	16
Icinga2 Documentation	16
Terminologies	17
Zones.conf Parameters	18
Services.conf Parameters	19
Commands.conf Parameters	20
Users.conf Parameters	21

Getting Started

Icinga

Icinga is an open source computer system and network monitoring application. It was originally created as a fork of the Nagios system monitoring application in 2009. Icinga is the chosen system driving our network monitoring system at the moment.

Icinga replaced our old system, a Nagios web interface, as a more modern and versatile approach to monitoring that could scale easily as our systems grow.

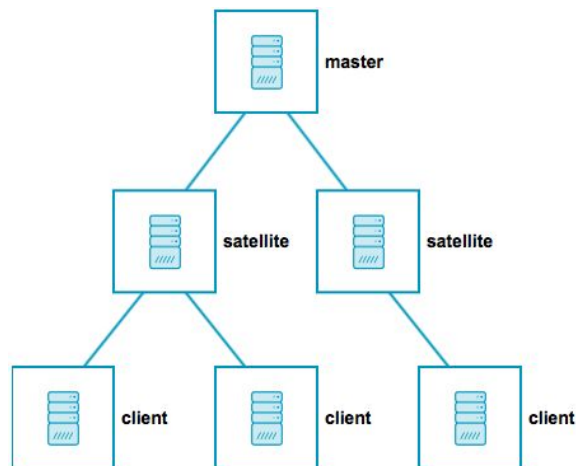


FIGURE 1: Icinga's *Top Down Command Endpoint* system

Our **master** instance is hosted on **sunusingbld05.sun.roche.com**. Configuration files can be found via SSH on that server in the **/etc/icinga2/** folder. Plugins can be found in the **/usr/lib64/nagios/plugins/** folder.

Icinga offers a useful web interface, viewable at **<http://sunusingbld05.sun.roche.com/icingaweb2>**. The username is "**icinga**" and the password is the same as the Ventana account on sunusingbld05.sun.roche.com.

More information about Icinga can be found on their site's documentation pages: **<http://www.icinga.com/>**

Graphite

Graphite is a free open-source software tool that monitors and graphs numeric time-series data such as the performance of computer systems. Graphite produces our graphs from data pulled by Icinga.



Though it is not necessary to view graphs via Graphite directly, it is possible by navigating to **<http://sunusimgbld05.sun.roche.com/>**

More information about Graphite can be found on their site's documentation pages:
<http://graphiteapp.org/>

Grafana

Grafana is a dashboard solution that ties together Icinga's data and Graphites graphs into a user-friendly and interactive interface. The Grafana instance can be viewed at **<http://sunusimgbld05.sun.roche.com:3000>** with the username "**admin**" and password "**admin**". The main playlist that runs on our monitoring television can be played from **<http://sunusimgbld05.sun.roche.com:3000/playlists/play/1>**.

More information about Grafana can be found on their site's documentation pages:
<http://grafana.com/>

Installations

Installing Child: Windows Servers

Connecting to Server

Connecting to a given Windows Server can be done through Windows **Remote Desktop Connection**. The “Computer” will be some “name.sun.roche.com”, and the “Username” should be whichever you have permissions to sign in with:

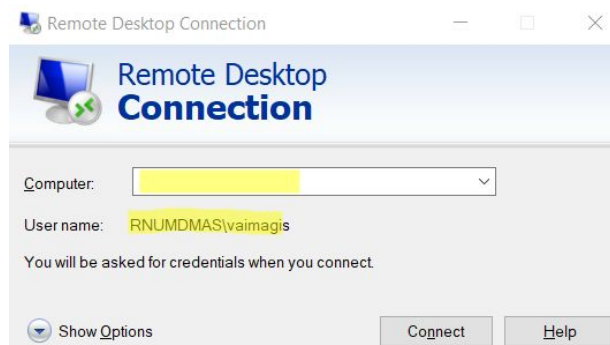


FIGURE 2: The launch screen of Remote Desktop

Once remotely connected to the server, proceed with the following:

Running Installer

The .MSI (Microsoft Installer) executable for Icinga’s GUI installation wizard can be found at **packages.icinga.com/windows**. Version 2.8.4 is preferred. This version can be downloaded directly by clicking **here**.

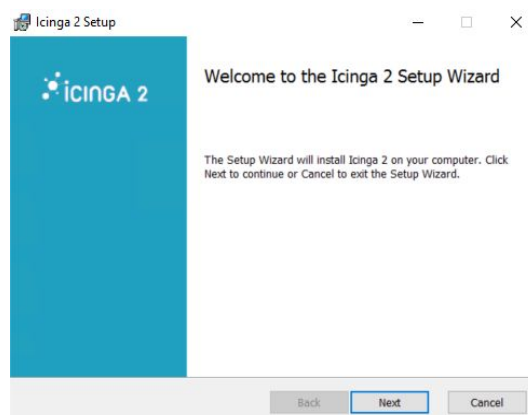


FIGURE 3: The launch screen of Icinga 2

Continue with default settings to install Icinga 2 and run the setup wizard. The “FQDN” should match your server’s domain name.

FIGURE 4: The blank setup wizard

Generating Ticket

To fill in the “Setup Ticket”, you need to generate one from your parent server (normally ‘SUNUSimgBLD05’):

```
[root@PARENT /]# icinga2 pki ticket --cn 'CHILD.sun.roche.com'
```

“CHILD.sun.roche.com” should be replaced with the FQDN of your new child server. In this case, we are setting up Icinga2 on 'SUNUSIMGBLD06.sun.roche.com'. This command will result in a long hexadecimal value, such as:

```
83bbbe4a3f8f1582ae85fa0bcd0ee6c0fb6589ee
```

Copy and paste this value somewhere whilst continuing with the installation of Icinga2. Plugging in these values, the setup wizard should look like this:

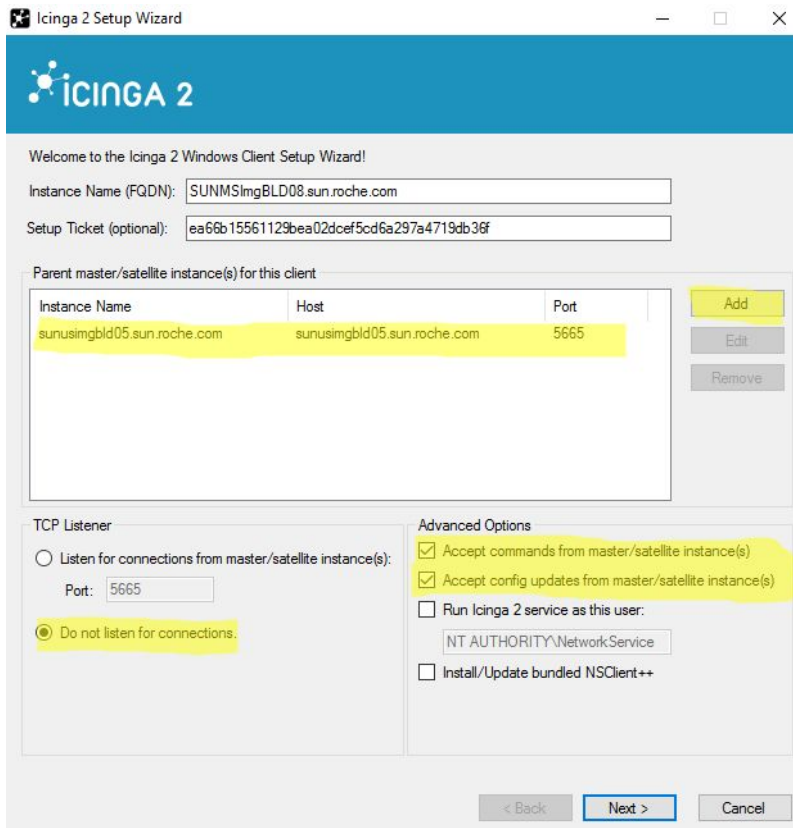


FIGURE 5: The filled-out setup wizard

Clicking “Add” to add the Host’s FQDN (or IP) and port (always 5665). Also, click the checkboxes for “Accept commands from master/satellite instances” and “Accept config updates from master/satellite instances”. After everything has been filled out, the window should look like the highlighted one on the right above.

Clicking through with default values, the program should finish successfully. If it crashes beyond this point, it is likely the Icinga2 server was not restarted with the second command displayed in the command line blurb above.

Configuring Parent

The last step is to let the Parent know there is a new Child to watch for:

```
[root@PARENT /]# nano /etc/icinga2/zones.conf
```

Copy and paste **this block of text** at the bottom of the document, but replacing the FQDN (e.g. SUNUSimgBLD07.sun.roche) and IP (e.g. 141.167.70.245) with your current server’s FQDN and IP.

```
[root@PARENT /]# systemctl restart icinga2
```

After a short warm-up period, all services running off of icinga's core (e.g. Icingaweb2, Grafana, Graphite) should have access to the new server's data.



A screenshot of the Icinga2 Web interface showing a list of services in a 'PENDING' state. The services are listed in a table with four rows, each showing the service name, the host, and the service type. The service names are SUNUSIMGBLD07.sun.roche.com, and the service types are disk, ping4, http, and disk /.

PENDING	SUNUSIMGBLD07.sun.roche.com (PENDING): disk
PENDING	SUNUSIMGBLD07.sun.roche.com (PENDING): ping4
PENDING	SUNUSIMGBLD07.sun.roche.com (PENDING): http
PENDING	SUNUSIMGBLD07.sun.roche.com (PENDING): disk /

FIGURE 6: Services starting on Icinga2 Web

Accessing data can be done visually through the Icinga2 web interface or interactively via Graphite on the Grafana Dashboard:

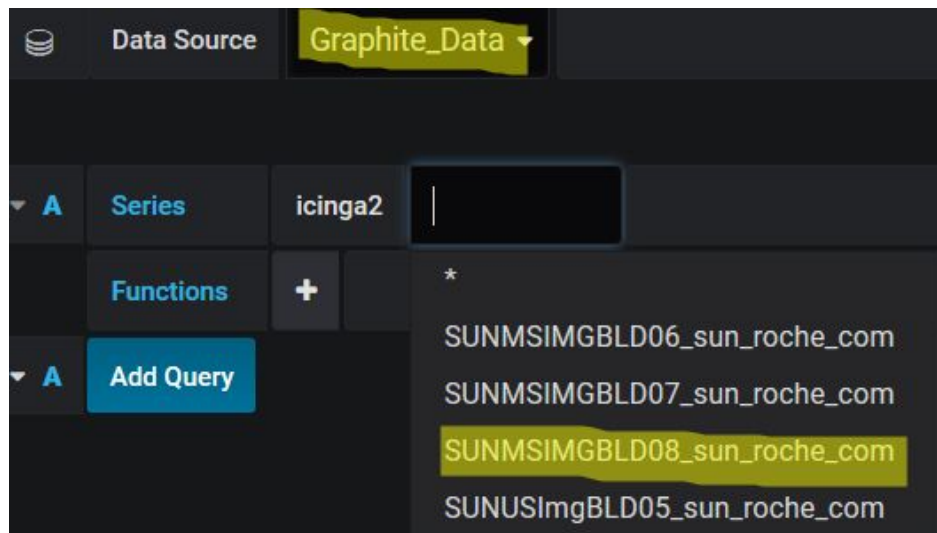


FIGURE 7: Accessing Graphite data from Grafana



Installing Child: CentOS Servers

Generating Ticket

To fill in the “Setup Ticket”, you need to generate one from your parent server (normally ‘SUNUSImgBLD05’):

```
[root@PARENT /]# icinga2 pki ticket --cn 'CHILD.sun.roche.com'
```

“CHILD.sun.roche.com” should be replaced with the FQDN of your new child server. In this case, we are setting up Icinga2 on 'SUNUSIMGBLD06.sun.roche.com'. This command will result in a long hexadecimal value, such as:

```
83bbbe4a3f8f1582ae85fa0bcd0ee6c0fb6589ee
```

Copy and paste this value (the one you generated) somewhere whilst continuing with the installation of Icinga2. It will be used in the setup wizard in a few steps.

Installing Icinga

First, to make things simpler, enter superuser mode:

```
[user@CHILD ~]$ sudo su
```

Next, download and install Icinga2:

```
[root@CHILD /]# yum install
```


```
https://packages.icinga.com/epel/icinga-rpm-release-7-latest.noarch.rpm
```

```
[root@CHILD /]# yum install icinga2 -y
```

Then, start Icinga2 and run the setup wizard:

```
[root@CHILD /]# systemctl start icinga2 && systemctl enable icinga2
```

```
[root@CHILD /]# icinga2 node wizard
```

```

Welcome to the Icinga 2 Setup Wizard!

We will guide you through all required configuration details.

Please specify if this is a satellite/client setup ('n' installs a master setup) [Y/n]: y
Starting the Client/Satellite setup routine...

Please specify the common name (CN) [SUNUSIMGBLD09.sun.roche.com]:

Please specify the parent endpoint(s) (master or satellite) where this node should connect to:
Master/Satellite Common Name (CN from your master/satellite node): SUNUSImgbLD05.sun.roche.com

Do you want to establish a connection to the parent node from this node? [Y/n]: y
Please specify the master/satellite connection information:
Master/Satellite endpoint host (IP address or FQDN): SUNUSImgbLD05.sun.roche.com
Master/Satellite endpoint port [5665]:

Add more master/satellite endpoints? [y/N]: N
Parent certificate information:

Subject:      CN = SUNUSImgbLD05.sun.roche.com
Issuer:       CN = Icinga CA
Valid From:   May 24 18:24:58 2018 GMT
Valid Until:  May 20 18:24:58 2033 GMT
Fingerprint: 01 4A 05 D7 5B 9D F4 39 2B C4 D9 7A 9C 2E 8B 88 D1 A7 71 55

Is this information correct? [y/N]: Y

Please specify the request ticket generated on your Icinga 2 master (optional).
(Hint: # icinga2 pki ticket --cn 'SUNUSIMGBLD09.sun.roche.com'): d9aa619e74f3fe68dce759b40538b1efa57ee996

Please specify the API bind host/port (optional):
Bind Host []:
Bind Port []:

Accept config from parent node? [y/N]: Y
Accept commands from parent node? [y/N]: Y

Reconfiguring Icinga...
Disabling feature notification. Make sure to restart Icinga 2 for these changes to take effect.
Enabling feature api. Make sure to restart Icinga 2 for these changes to take effect.

Done.

Now restart your Icinga 2 daemon to finish the installation!

```

FIGURE 8: Icinga2's Linux setup wizard

All responses needed from Wizard:

- Y
- [enter]
- SUNUSImgbLD05.sun.roche.com
- Y
- SUNUSImgbLD05.sun.roche.com
- [enter]
- N
- Y
- [Ticket generated in **Generating Ticket**]
- [enter]
- [enter]
- Y
- Y

Lastly, restart the local Icinga2 instance:

```
[root@CHILD /]# systemctl restart icinga2
```

Configuring Parent

The last step is to let the Parent know there is a new Child to watch for:

```
[root@PARENT /]# nano /etc/icinga2/zones.conf
```

Copy and paste **this block of text** at the bottom of the document, but replacing the FQDN (e.g. SUNUSImgBLD07.sun.roche) and IP (e.g. 141.167.70.245) with your current server's FQDN and IP.

```
[root@PARENT /]# systemctl restart icinga2
```

After a short warm-up period, all services running off of icinga's core (e.g. Icingaweb2, Grafana, Graphite) should have access to the new server's data.



A screenshot of the Icinga2 Web interface showing a list of pending services. The list is displayed in a table with four rows, each representing a different service type (disk, ping4, http, disk /). The status for all services is 'PENDING'. The FQDN 'SUNUSIMGBLD07.sun.roche.com' is repeated for each row.

PENDING	SUNUSIMGBLD07.sun.roche.com (PENDING): disk
PENDING	SUNUSIMGBLD07.sun.roche.com (PENDING): ping4
PENDING	SUNUSIMGBLD07.sun.roche.com (PENDING): http
PENDING	SUNUSIMGBLD07.sun.roche.com (PENDING): disk /

FIGURE 9: Services starting on Icinga2 Web

Accessing data can be done visually through the Icinga2 web interface or interactively via Graphite on the Grafana Dashboard:

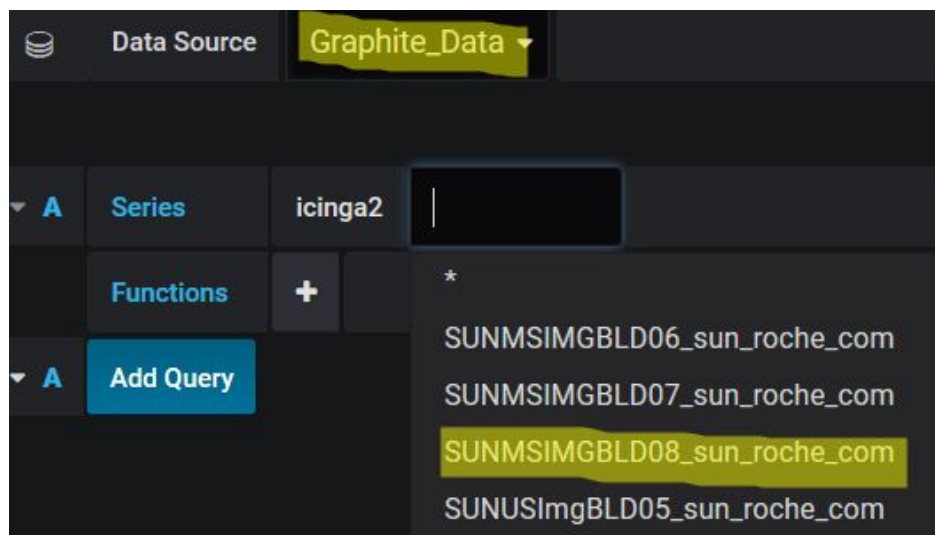


FIGURE 10: Accessing Graphite data from Grafana

Configurations

Adding new services to be monitored is relatively painless, but can require troubleshooting to understand what part of the multi-step process is not working.

Icinga

Services

Adding, removing, or modifying service monitoring can be done through the `Conf.d/Services.conf` file. To understand how to modify this file, check out the References section on **Service.conf Parameters**.

After any modifications of configurations files, Icinga2 needs to be reset:

```
[root@PARENT /]# systemctl restart icinga2
```

If it fails to restart, you can get a thorough error print out via the following:

```
[root@PARENT /]# icinga2 daemon -C
```

When calling commands on child servers, that child needs to have the proper plugins to execute the desired commands. If they don't have these plugins, you will get the following error in Icinga Web 2:

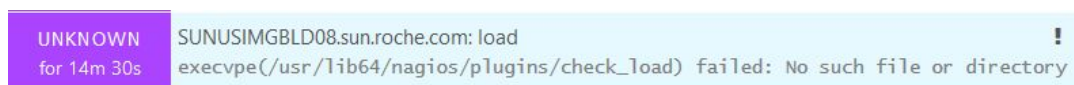


FIGURE 11: Error when a child server does not have a necessary plugin

To fix this, you need to install the necessary plugins on the child server:

```
[root@CHILD /]# yum install  
https://dl.fedoraproject.org/pub/epel/epel-release-latest-7.noarch.rpm  
[root@CHILD /]# yum install nagios-plugins-all
```

If these default Nagios plugins do not include the plugin you are looking to utilize, you can manually place it in the `/usr/lib64/nagios/plugins` folder via SFTP. And if it isn't executable (or green-colored when ls-ing the folder), you can fix that via:

```
[root@CHILD /]# chmod +x check SOMETHING
```

Further issues can normally be resolved by simple Googling, or by a question on a forum like **Monitoring Portal**.

Commands

Adding, removing, or modifying the commands being a service can be done through the Conf.d/Commands.conf file. To understand how to modify this file, check out the References section on **Commands.conf Parameters**.

For most cases, however, modifying this file is not necessary, and is usually for implementing custom plugins.

Emailing

Controlling who receives emails can be controlled through the Conf.d/Users.conf:

```
[root@PARENT /]# nano /etc/icinga2/conf.d/users.conf
```

For instructions on how to create new users to be added to the email list, see **Users.conf**.

Grafana

Panels

Creating UI objects (called panels) in Grafana is intuitive and user-friendly. Clicking the “Add Panel” icon brings the following options up.

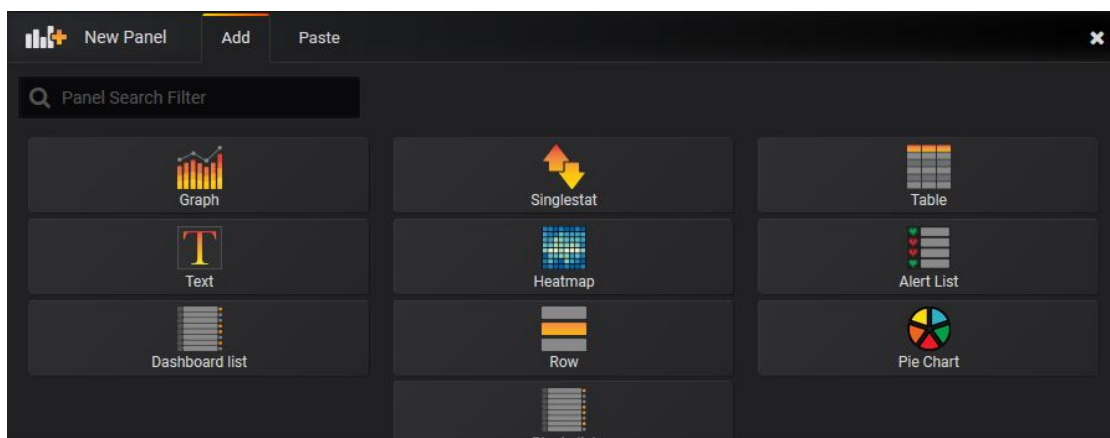


FIGURE 12: Options when adding a Grafana panel

After selecting a panel, you can edit it by clicking on the header of the panel. From here, you can manage the header's name, metrics, axes, etcetera. These modifications are straightforward, with the possible exception of the “Metrics” tab. It is important to select

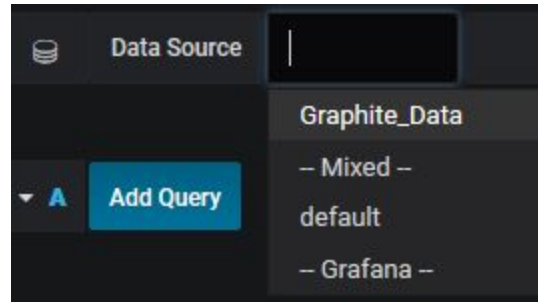


FIGURE 13: Setting the correct Data Source

Selecting Graphite_Data will allow Queries to pull graphs from Icinga's data:

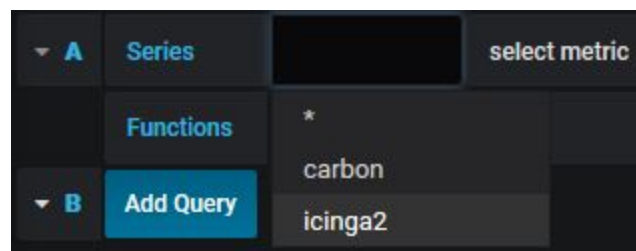


FIGURE 14: Adding an Icinga Query

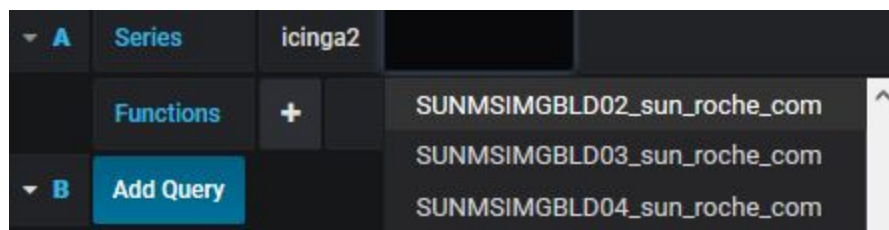


FIGURE 15: Selecting a host

The rest of the options are intuitive, selecting the service to monitor, its performance data or metadata, and the variable from that service to monitor.

Dashboards

Dashboards are the slides in the overview playlist, and can also be used as stand-alone monitoring interfaces to aggregate graphs and statistics into one concise page. They can be created easily through the Create menu (shown below), and populated with **Panels**.

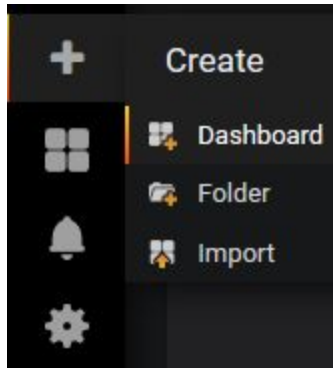


FIGURE 16: Creating a new Dashboard

Playlists

Playlists function as a slideshow of dashboards for easy display. They can be created and modified via the Dashboards menu. This menu allows the creation, modification, and deletion of playlists. Modification includes controlling which dashboards are displayed, and how long they are displayed for.

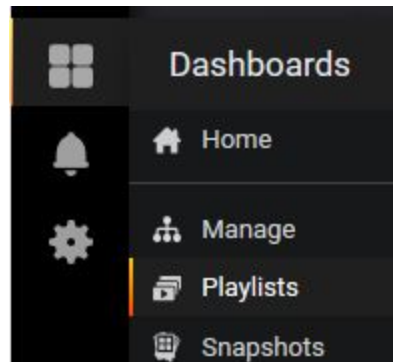


FIGURE 17: Navigating to the Playlist menu



References

LinuxTechi Tutorials

LinuxTechi two tutorials on Icinga were the main guide followed in initially configuring Icinga2. However, it should be noted that these tutorials were not perfect, and there were additional workarounds to get programs working properly. These are a good place to start though:

Installing Icinga Parent:

<https://www.linuxtechi.com/install-configure-icinga2-centos-7-rhel-7/>

Installing Icinga Child:

<https://www.linuxtechi.com/add-remote-linux-windows-host-icinga-2-monitoring/>

Icinga2 Documentation

Icinga2 has comprehensive documentation and covers many low- and high-level topics. However, the documentation is often clunky and suggested practices are not always as effective as the docs make them seem. For concepts and principles, it is a must read.

Official Documentation

<https://www.icinga.com/docs/icinga2/>



Terminologies

Icinga and other programs use specific wordings in regards to their installations and practices. These wordings should be noted to fully understand the processes described in this document. The following is a comprehensive (for the scope of this document) list of terminologies to watch for.

Distributed Monitoring is the name of the game. The following are key terms to know to best understand how the current system is configured. Read more **here**.

"Host"	Any server running Icinga services. Should not be confused with "Parent" or "Master".
"Parent" (also "Master")	The Host that all Children point towards. Collects data from Children into one interface.
"Child" (also "Client")	A Host that points towards a specific Parent or Satellite. Collects data from server it is running on through Icinga daemons and relies diagnostics to Parent for analysis.
Top Down Command Endpoint	The principle behind how this Icinga system executes commands. Read more about this principle and how it affects configuring the system here .

Zones.conf Parameters

Highlighted pieces need to be adjusted to the new server's FQDN and IP address. SUNUSImgBLD05.sun.roche.com is the current host, so that FQDN stays the same unless the host has changed servers since when this documentation was written.

```
object Zone "FQDN.sun.roche.com" {
    endpoints = [ "FQDN.sun.roche.com" ]
    parent = "SUNUSImgBLD05.sun.roche.com"
}
object Endpoint "FQDN.sun.roche.com" {
    host = "CHILD.Server.IP.Address"
}
object Host "FQDN.sun.roche.com" {
    import "generic-host"
    address = "CHILD.Server.IP.Address"
    vars.http_vhosts["http"] = {
        http_uri = "/"
    }
    vars.disks["disk"] = {
    }
    vars.disks["disk /"] = {
        disk_partitions = "/"
    }
    vars.notification["mail"] = {
        groups = [ "icingaadmins" ]
    }
    vars.os = "Linux"
    vars.client_endpoint = "FQDN.sun.roche.com"
}
```

* vars.os is a variable all hosts will be assigned with. However, you may notice some hosts have other variables, such as 'vars.ims = "Running"' to call IMS-related services.

[Back to Windows Installation Walkthrough](#)

[Back to CentOS Installation Walkthrough](#)



Services.conf Parameters

The following are two example implementations of services (in this case, to monitor CPU load on Linux and Windows machines. Highlighted are important properties to take note of:

```
apply Service "load" {
    import "generic-service"
    check_command = "load"
    vars.backup_downtime = "02:00-03:00"
    command_endpoint = host.vars.client_endpoint
    assign where (host.address)
}
```

- * This line controls what plugin/check **command** will be called
- * This line is the implementation of the **Top Down Command Endpoint** principle.
- * This line applies the service to all **Hosts**.

```
apply Service "load-windows" {
    import "generic-service"
    check_command = "load-windows"
    command_endpoint = host.vars.client_endpoint
    assign where (host.address && host.vars.os == "Windows")
}
```

- * This line demonstrates using boolean logic to specify where services will run.

For a list of possible commands, check out Icinga's documentation **here**, and the links it recommends investigating (such as **this** for Windows monitoring commands). Googling for specific plugins is also effective, as Nagios offers many custom plugins that are easily implementable on Icinga.

Commands.conf Parameters

Commands allow for the implementation of custom plugins with unique attributes to be programmatically constructed and called on all desired servers. The following is the implementation of the **check_http** plugin with specific parameters as to be able to monitor Jenkins services on port 8080:

```
[root@PARENT /]# ./check http -H PARENT -u /api/xml?depth=0 -p 8080
```

```
object CheckCommand "jenkins-http" {  
    import "plugin-check-command"  
    command = [ PluginDir + "/check_http" ]  
    arguments = {  
        "-H" = {  
            value = "$host.name$"  
        }  
        "-u" = {  
            value = "/api/xml?depth=0"  
        }  
        "-p" = {  
            value = "8080"  
        }  
    }  
}
```

* This line shows how to call a specific executable plugin. `PluginDir` is a constant from `Constants.conf`.

* This line shows the use of variables when executing a command.

For most cases, making unique `CheckCommand` objects is not necessary, and when it is, these code snippets are normally provided when reading up on a specific plugin to be implemented.

Back to Configuration



Users.conf Parameters

The following is an example of adding a new User to Icinga for the main purpose of adding said User to the emailing list for warnings and errors of systems monitored by Icinga.

```
object User "username" {  
    import "generic-user"  
    display_name = "User Name"  
    groups = [ "icingaadmins" ]  
    email = "email@roche.com"  
}
```

- * This name needs to be unique to differentiate it from other Users.
- * Notifications will be sent to the following email address.

Back to Emailing