

Installation Of The First Windows-Client

Assumptions

1. Master has just been set up and is able to execute local checks; Displays these in IcingaWeb2.
2. `icinga2 node wizard` has been run at the master. Mode has been given as Master, CN has been chosen to be the machines FQDN.
3. In that document, Master has Hostname and Endpoint-Name `debian85.local`, as well as the IP `192.168.200.6`
4. The Windows Client is named `LAPTOP-AUQ5DGU2`.
5. We want the Client to connect to the Master, not vice versa.

Modifications At The Master

zones.conf master

```
object Endpoint NodeName {  
}
```

```
object Endpoint "LAPTOP-AUQ5DGU2"{  
}
```

```
object Zone ZoneName {  
    endpoints = [ NodeName ]  
}
```

```
object Zone "LAPTOP-AUQ5DGU2" {  
    endpoints= [ "LAPTOP-AUQ5DGU2" ]  
    parent= ZoneName  
}
```

```
object Zone "global-templates" {  
    global = true  
}
```

Run The Below Commands

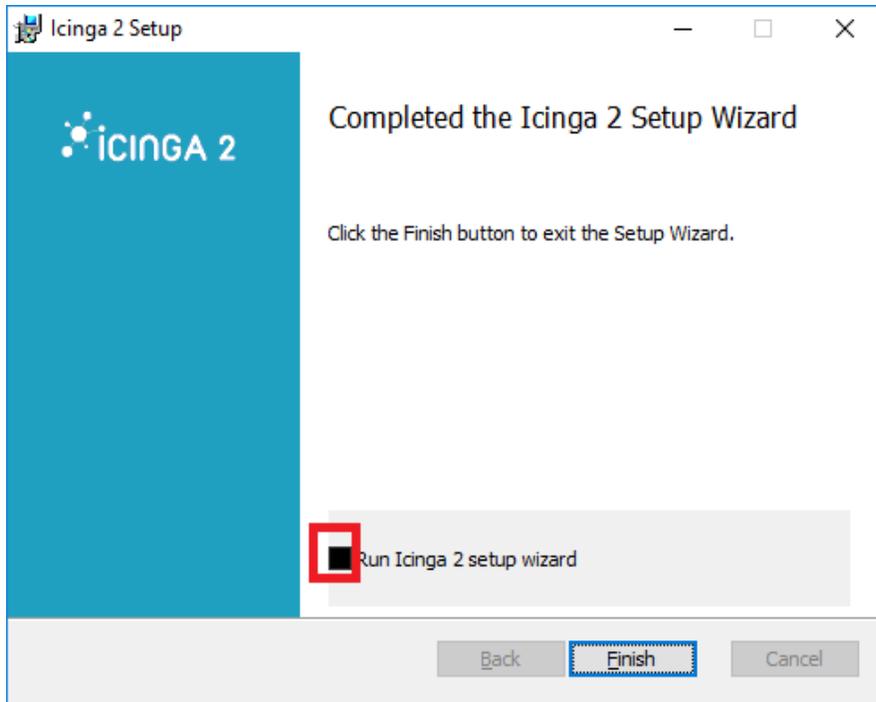
```
root@debian85:/etc/icinga2/zones.d# mkdir /etc/icinga2/zone.d/global-templates  
root@debian85:/etc/icinga2/zones.d# mkdir /etc/icinga2/zone.d/LAPTOP-AUQ5DGU2
```

```
root@debian85:/etc/icinga2/zones.d# icinga2 pki ticket --cn 'LAPTOP-AUQ5DGU2'  
2483cf6f158c06f362b2f2a7ea29b72b25d14d17
```

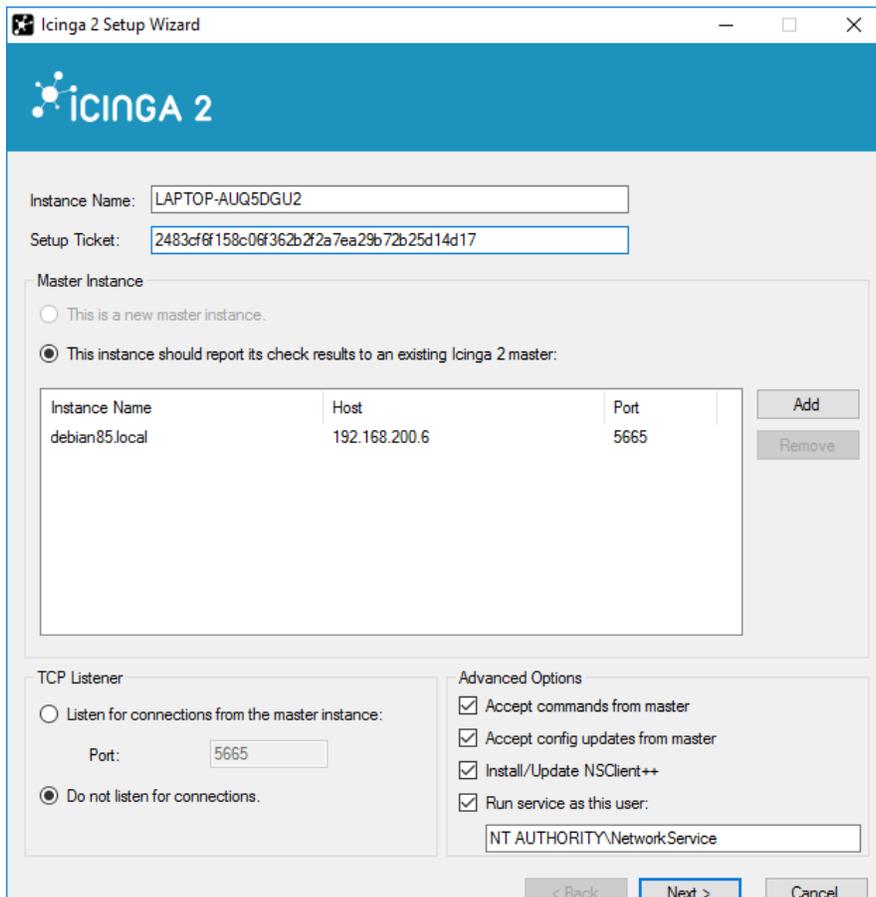
```
root@debian85:/etc/icinga2/zones.d# icinga2 feature list  
Disabled features: compatlog debuglog gelf graphite influxdb livestatus opentsdb perfddata  
statusdata syslog  
Enabled features: api checker command ido-mysql mainlog notification
```

Installation At The Windows Client

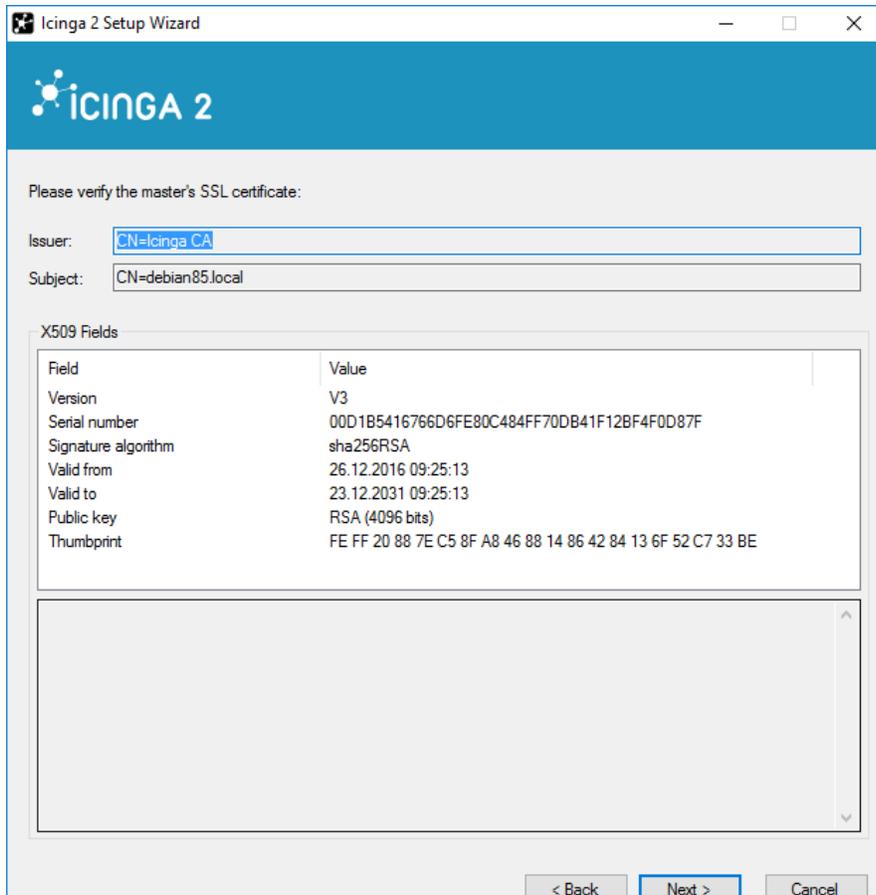
Get binary from http://packages.icinga.org/windows/icinga2-v2.6.0-x86_64.msi run and fill in according the screenshots below:



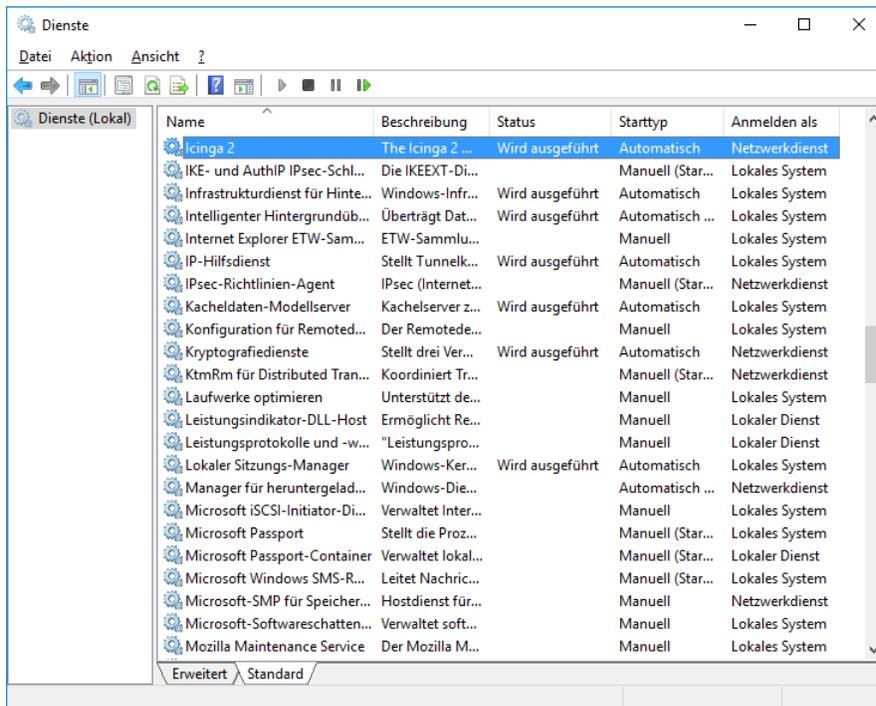
Screenshot1



Screenshot2



Screenshot3



Screenshot4

Modifications At The Windows Client

zones.conf client

Config Files are below C:\ProgramData\icinga2\etc\icinga2.

```
object Endpoint "debian85.local" {
    ### Next line results in the client connecting to the master, not vice versa.
    host = "192.168.200.6"
    port = "5665"
}

object Zone "master" {
    endpoints = [ "debian85.local" ]
}

### nodeName is a Constant in constants.conf
object Endpoint nodeName {
}

### ZoneName is a Constant in constants.conf
object Zone ZoneName {
    endpoints = [ nodeName ]
    parent = "master"
}

object Zone "global-templates" {
    global = true
}
```

Create An Intermim Windows Service At The *Master*, To Verify Top Down Replikation

File `/etc/icinga2/zone.d/LAPTOP-AUQ5DGU2/services.conf`

```
### we call the Service loadtmp, because load already exists.
apply Service "loadtmp" {
    import "generic-service"

    check_command = "load"
    enable_flapping = true
    /* Used by the ScheduledDowntime apply rule in `downtimes.conf`. */
    vars.backup_downtime = "02:00-03:00"

    assign where host.name == "LAPTOP-AUQ5DGU2"
}
```

Check At The Master

```
service icinga2 restart
icinga2 object list --type service --name loadtmp
```

The host object (*not* endpoint object !) LAPTOP-AUQ5DGU2 is not known at the master. So, the master is unable to create a service for that host. That is why neither the host nor the service show up in icingaweb2.

Recheck At The Client

```
C:\Program Files\ICINGA2\sbin>icinga2.exe object list --type service --name loadtmp
```

Service is listed. We verified a successfull replication but are missing a successfull monitoring.

Modifications To Fix The Monitoring

Remove at the Client `conf.d\hosts.conf`

```
C:\ProgramData\icinga2\etc\icinga2\conf.d>ren hosts.conf hosts.conf.orig
C:\ProgramData\icinga2\etc\icinga2\conf.d>net stop icinga2
C:\ProgramData\icinga2\etc\icinga2\conf.d>net start icinga2
```

Create `/etc/icinga2/zone.d/LAPTOP-AUQ5DGU2/hosts.conf` At The Master

```
object Host "LAPTOP-AUQ5DGU2" {
    import "generic-host"

    address = "127.0.0.1"
    address6 = "::1"

    vars.os = "Windows"

    vars.disks["disk"] = {
        /* No parameters. */
    }
    vars.disks["disk C:"] = {
        disk_win_path = "C:"
    }

    vars.notification["mail"] = {
        groups = [ "icingaadmins" ]
    }
}
```

This results in the Host-Object being replicated from the Master to the Client. Now both, Master and Client, are aware of the host object, and Monitoring is working. If we had not removed the `hosts.conf` at the client, it's `icinga2` service would have complained while validating the replicated host object: "Already exists at my local `conf.d/hosts.conf`, Invalid, stop the service!!!"

Recheck At The Master

```
service icinga2 restart
icinga2 object list --type service --name loadtmp
```

Service is listed.

Recheck At The Client (2)

```
C:\Program Files\ICINGA2\sbin>icinga2.exe object list --type service --name loadtmp
```

Service is listed. We have a successful replication and a working Monitoring at the Master:

The screenshot shows the Icinga Web interface for a host named 'LAPTOP-AUQ5DGU2'. The service 'loadtmp' is shown as 'UP' since 9m 57s. The interface includes a navigation menu on the left with options like 'Übersicht', 'Hosts', 'Services', and 'Konfiguration'. The main content area displays the service status, a 'Jetzt prüfen' button, and sections for 'Ausgabe des Plugins' (LOAD OK 4.09816%), 'Problembehandlung' (with a 'Downtime planen' button), 'Performancedaten' (showing load at 4,09816% with 5% warning and 10% critical thresholds), and 'Benachrichtigungen' (with a 'Benachrichtigung senden' button).

Screenshot5

Further Operations

We would like to centrally manage the services.conf at the Master. If we change it here, it will be replicated to all zones (and the endpoints within these zones). The file contains `apply` rules, that add a set of default services to each host. First, we remove the file services.conf at the client, it will receive it back from the master:

```
C:\ProgramData\icinga2\etc\icinga2\conf.d>ren services.conf services.conf.orig
C:\ProgramData\icinga2\etc\icinga2\conf.d>net stop icinga2
C:\ProgramData\icinga2\etc\icinga2\conf.d>net start icinga2
```

At the master, we move the file services.conf into global-templates:

```
root@debian85:/etc/icinga2/conf.d# mv services.conf ../zones.d/global-templates/
```

and modify the service load from:

```
apply service "load" {
    import "generic-service"

    check_command = "load"
    enable_flapping = true
    /* Used by the ScheduledDowntime apply rule in `downtimes.conf`. */
    vars.backup_downtime = "02:00-03:00"

    assign where host.name == NodeName
}
```

to now read:

```
apply Service "load" {
  import "generic-service"

  check_command = "load"
  enable_flapping = true
  /* Used by the ScheduledDowntime apply rule in `downtimes.conf`. */
  vars.backup_downtime = "02:00-03:00"

  assign where host.name
}
```

The interrim service object *can* be removed (but it does not harm if you keep it...):

```
root@debian85:/etc/icinga2/conf.d# rm ../zones.d/LAPTOP-AUQ5DGU2/services.conf
root@debian85:/etc/icinga2/conf.d# service icinga2 restart
```

Recheck that the monitoring is still working. Service `loadtmp` has been replaced by `load`.

You should have learned by now:

- Objects that need to exist in all zones go to `zones.d/global-templates`.
- Objects that need to exist in a special zone only go to `zones.d/[zonename]`.
- Objects that exist in `conf.d` of *any* machine *may* conflict with these that exist at the master and thus are strictly to avoid.