# Condor Users Guide


**6.1 Edition**

**Condor Users Guide :**
6.1 Edition
Published Nov 27 2012

# Table of Contents

# List of Tables

# Preface

The Rocks Condor Roll uses the latest stable Condor Release to provide High Throughput Computing environment for Rocks clusters. The Rocks Condor Roll builds on the very good work by the Condor team, to seamlessly install the *de facto* standard grid middleware on Rocks Clusters.

Please visit the Condor Project Homepage[1] to learn more about Condor and the software releases.

A local copy of their online manual pages is available  on this cluster [2].

# Notes

1. http://www.cs.wisc.edu/condor/
2. condor-Manual

# Chapter 1. Overview

**Table 1-1. Summary**

| | |
|---|---|
| Name | condor |
| Version | 6.1 |
| Maintained By | Rocks Group |
| Architecture | i386, x86_64 |
| Compatible with Rocks® | 6.1 |

The condor roll has the following requirements of other rolls. Compatability with all known rolls is assured, and all known conflicts are listed. There is no assurance of compatiblity with third-party rolls.

**Table 1-2. Compatibility**

| Requires | Conflicts |
|---|---|
| **Base** | |
| **Kernel** | |
| **OS** | |

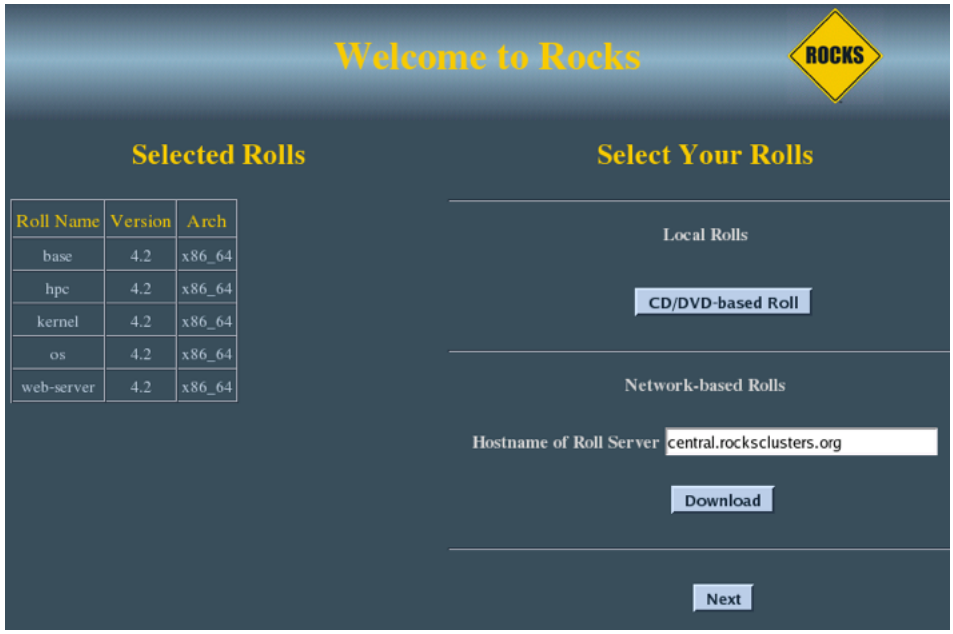

This roll has been released independent of the corresponding Rocks® release. It therefore requires the complete **OS** roll and will not function correctly if using only the **Jumbo** or incomplete set of **OS** CDROMs.

# Chapter 2. Installing the Condor Roll

## 2.1. Adding the Roll

The Condor Roll can be installed during the Frontend installation or added to a running frontend. In either case, client nodes must be (re)installed.

The Condor Roll is added to a Frontend installation in exactly the same manner as the required base Roll. Simply select the Condor roll as you would any network- or CD-based roll.




Once the Condor Roll is loaded the installer will continue installation automatically with no further user input. The Frontend will be configured as a condor pool manager and a submit node. Nodes of appliance type "compute" will will be configured as submit/execute nodes. Other appliances can be made part of the execution pool through setting an appliance attribute and reinstalling those nodes.

## 2.2. Frontend Boot

On first boot, the Condor roll will complete its configuration of the Condor daemons.

## 2.3. Install on Running System

The Condor Roll can be installed on a running system. The following assumes that roll is available in .iso form and is called "condor.iso".

```
$ su - root
# rocks add roll condor.iso
# rocks enable roll condor
# rocks create distro
# rocks run roll condor | bash
```

```
# shutdown -r now
```

# Chapter 3. Using the Condor Roll

## 3.1. Using Condor

This section explains the Condor pool configuration on Rocks clusters, and provides link to further documentation.

Machines in the Condor pool can serve a variety of roles:

- *Central manager* machine is the collector of information, and the negotiator between resources and resource requests. In any Condor pool one machine must be the Central Manager.
- *Submit machine* allows Condor jobs submission.
- *Execute machine* provides resources for Condor jobs execution.

The Frontend is configured as the pool's Central Manager, and the Submit machine. The rest of the nodes are configured as Submit/Execute machines.

The configuration of condor is done during the install, the resulting configuration files are located in /opt/condor/etc. To reconfigure condor pool use /opt/condor/sbin/CondorConf command, and then restart condor daemons so the new configuration takes an effect. CondorConf takes *-h* flag that gives the usage explanation on stdout. Usually, the reconfiguration is only needed when you have multiple network interfaces on your nodes, and the default configuration is not using your desired interface.

By default, on the frontend condor daemons are running on *eth1* interface, and on the compute nodes on *eth0* interface.

Here is en axample command for configuring condor as the pool manager and the submit machine on the frontend, assuming the forntend has IP 198.202.89.155:

```
/opt/condor/sbin/CondorConf -n frontend -t sm -m 198.202.89.155
```

The following command will configure compute nodes as Submit/Execute machines:

```
/opt/condor/sbin/CondorConf -n compute -t se -m 198.202.89.155
```

To find information about administrering and using Condor Pools please see the original Condor manual at Condor manuals[1]

## 3.2. Testing the Condor Roll

1. First, make sure condor daemons are running by executing:

   ```
   # ps -ef | grep condor
   ```

   On the frontend, the output should be similar to following:

   ```
   condor    2623     1  0 Apr19 ?        00:04:26 /opt/condor/sbin/condor_master
   condor    2646  2623  0 Apr19 ?        00:20:25 condor_collector -f
   condor    2647  2623  0 Apr19 ?        00:04:56 condor_negotiator -f
   condor    2649  2623  0 Apr19 ?        00:00:02 condor_schedd -f
   ```

And on the compute nodes, the output should be similar to following:

```
condor   17007     1  0 Apr19 ?        00:01:09 /opt/condor/sbin/condor_master
condor   17009 17007  0 Apr19 ?        00:00:02 condor_schedd -f
condor   17010 17007  0 Apr19 ?        00:09:09 condor_startd -f
```

2. Try a test job submission. If you don't already have a non-root user account, you'll need to create one:

```
# useradd testuser
# rocks sync users
```

Then, to run a test job submission, execute:

```
# su - testuser
$ cp /opt/condor/tests/hello.* .
$ condor_submit hello.sub
```

3. Check if jobs are submitted by executing:

```
$ condor_q
```

The output should be similar to:

```
-- Submitter: jeebs.rocksclusters.org : <172.19.119.241:44626> : jeebs.rocksclusters.org
 ID      OWNER            SUBMITTED     RUN_TIME ST PRI SIZE CMD
   2.0   testuser       12/12 09:38   0+00:00:01 R  0   0.0  hello.sh

1 jobs; 0 idle, 1 running, 0 held
```

*R* in status column(ST) means running. *I* means idling.

4. Once the queue is empty (above command shows no jobs) can see the history of jobs execution with:

```
$ condor_history
```

To see all the nodes in the condor pool do:

```
$ condor_status
```

The output should be similar to:

```
Name           OpSys       Arch   State      Activity   LoadAv Mem   ActvtyTime

vm1@compute-0 LINUX       INTEL  Unclaimed  Idle       0.000   506  0+00:40:04
vm2@compute-0 LINUX       INTEL  Unclaimed  Idle       0.000   506  0+00:45:05
vm3@compute-0 LINUX       INTEL  Unclaimed  Idle       0.000   506  0+00:45:06
vm4@compute-0 LINUX       INTEL  Unclaimed  Idle       0.000   506  0+00:45:07
vm1@compute-0 LINUX       INTEL  Unclaimed  Idle       0.000   506  0+00:35:04
vm2@compute-0 LINUX       INTEL  Unclaimed  Idle       0.000   506  0+00:40:05
vm3@compute-0 LINUX       INTEL  Unclaimed  Idle       0.000   506  0+00:40:06
vm4@compute-0 LINUX       INTEL  Unclaimed  Idle       0.000   506  0+00:40:07
vm1@compute-0 LINUX       INTEL  Unclaimed  Idle       0.000   506  0+00:25:04
vm2@compute-0 LINUX       INTEL  Unclaimed  Idle       0.000   506  0+00:30:05
vm3@compute-0 LINUX       INTEL  Unclaimed  Idle       0.000   506  0+00:30:06
vm4@compute-0 LINUX       INTEL  Unclaimed  Idle       0.000   506  0+00:30:07
vm1@compute-0 LINUX       INTEL  Unclaimed  Idle       0.000   506  0+00:15:05
vm2@compute-0 LINUX       INTEL  Unclaimed  Idle       0.000   506  0+00:20:06
vm3@compute-0 LINUX       INTEL  Unclaimed  Idle       0.000   506  0+00:20:07
vm4@compute-0 LINUX       INTEL  Unclaimed  Idle       0.000   506  0+00:20:08
vm1@compute-0 LINUX       INTEL  Unclaimed  Idle       0.000   506  0+00:10:04
vm2@compute-0 LINUX       INTEL  Unclaimed  Idle       0.000   506  0+00:15:05
vm3@compute-0 LINUX       INTEL  Unclaimed  Idle       0.000   506  0+00:15:06
vm4@compute-0 LINUX       INTEL  Unclaimed  Idle       0.000   506  0+00:15:07
vm1@compute-0 LINUX       INTEL  Unclaimed  Idle       0.000   506  0+00:00:04
vm2@compute-0 LINUX       INTEL  Unclaimed  Idle       0.000   506  0+00:05:05
vm3@compute-0 LINUX       INTEL  Unclaimed  Idle       0.000   506  0+00:05:06
```

```
vm4@compute-0 LINUX       INTEL  Unclaimed  Idle       0.000   506  0+00:05:07
vm1@compute-0 LINUX       INTEL  Owner      Idle       0.860   506  0+00:00:09
vm2@compute-0 LINUX       INTEL  Unclaimed  Idle       0.000   506  0+00:00:05
vm3@compute-0 LINUX       INTEL  Unclaimed  Idle       0.000   506  0+00:00:06
vm4@compute-0 LINUX       INTEL  Unclaimed  Idle       0.000   506  0+00:00:07

                     Machines Owner Claimed Unclaimed Matched Preempting

          INTEL/LINUX       28     1       0        27       0          0

                Total       28     1       0        27       0          0
```

# Notes

1. http://www.cs.wisc.edu/condor/manual

# Chapter 4. Customizing the Condor Roll

## 4.1. Customizing the Condor Roll

This section describes the default Condor configuration and some simple cusomizations that can be applied in Rocks with version >= 5.4

By default, Condor execution daemons are only installed on Rocks *compute* appliances. For Rocks 5.2 and newer, the Condor roll makes use of *attributes* to enable Condor to be installed on any appliance. This may be particularly useful to groups who are including the Xen roll and would like Condor to install on VM Container appliances.

The basic customizations that can be applied without and scripting/programming by setting global, appliance, or host attributes. Please see the commands `rocks set attr help` and `rocks list attr help`

**Table 4-1. Attributes Used in Condor Roll**

| Attibute Name | Description |
|---|---|
| Condor_Client | Enable/Disable Condor Client Installation on any particular Appliance or Host |
| Condor_PasswordAuth | Use a shared pool password, instead of host-based authentication. Default: no. |
| Condor_HostAllow | Comma separates list of allowed readers/writers for Condor. Translates to HOSTALLOW directive in Condor Configuration file. |
| Condor_EnableMPI | Configure a local scheduler for MPI Universe Support |
| Condor_Master | Redefine the Condor Master that nodes use |
| Condor_Network | Define which network interface is used for Condor traffic. Default: frontends are set to public, clients are set to private. |
| Condor_PortHigh: | Upper Port range that Condor will use to communicate among daemons. Removal of this Attribute will result in removal of the HIGHPORT entry in condor_config.local after syncing the configuration. Default: 50000 |
| Condor_PortLow | Lower Port range that Condor will use to communicate among daemons. Removal of this Attribute will result in removal of the LOWPORT entry in condor_config.local after syncing the configuration. Default: 40000 |

## 4.2. Using a pool password to secure Condor Communications

The default Rocks configuration is to use host-based authentication. This is a good and simple choice for a

cluster with a private network. With the Rocks 5.4 version of the Condor Roll, it is straightforward to set up a "Pool Password" that utilizes a shared secret among pool members. This is especially useful when allowing remote systems to report directly to the Condor collector on your cluster. The EC2 Roll can utilize a pool password for a higher security.

The following, straightforward will create, copy and enable a system-wide shared-secret pool password.

1. Create a pool password. Use `rocks create condor password`
2. Enable pool password security. Use `rocks set attr Condor_Password yes`
3. Reconfigure Condor Daemons and copy new pool password. Use `rocks sync host condor syncpassword=yes localhost compute`

## 4.3. Examples of Condor Configuration

The following are short examples of how to customize Condor using Rocks commands.

- Enable Condor Client on all VM-Containers Appliances: `rocks add appliance attr vm-container Condor_Client true`
- Disable Condor on particular node: `rocks set host attr compute-0-0 Condor_Client false`
- Define a New Condor Master: `rocks set attr Condor_Master central-master.my.edu`
- Enable MPI/Dedicated Scheduler: `rocks set attr Condor_EnableMPI true`.

  Actively-running Condor daemons must be reconfigured for this attribute to take affect. This can be achieved dynamically on compute and frontend appliances using `rocks sync host condor frontend compute`.

  Reinstalled nodes will build the correct configuration.

## 4.4. Reconfiguring Condor after Installation

The configuration of Condor is done during the install, the resulting configuration files are located in /opt/condor/etc. To reconfigure Condor on a node, make appropriate attribute using the commands above and then run

```
# rocks sync host condor <hostname>
```

This will rewrite the condor_config.local on the file and then calls the Condor command `/opt/condor/sbin/condor_reconfigure`

To view the contents of the condor_config.local before making changes, use `rocks report host condor config <hostname>`

To find information about administrating and using Condor Pools please see the original Condor manual at Condor manuals[1] or locally [2].

## 4.5. Programatically changing the Contents of condor_config.local

Condor configuration is localized into /opt/condor/etc/condor_config.local. This file is generated programatically from the output of `rocks report host condor config <hostname>`.

The command `rocks report host condor config` is defined by the Condor roll and is written in Python. This report command is extensible through Rocks command plugins.

To see a sample Condor plugin, view the file in location `/opt/rocks/lib/python2.4/site-packages/rocks/commands/report/host/condor/config/plugin_sample.py`, which is reproduced here.

```
# $Id: customizing.sgml,v 1.8 2010/12/08 16:58:03 phil Exp $
import rocks.commands

class Plugin(rocks.commands.Plugin):

 def provides(self):
  return 'sample'

 def run(self, argv):
  # Argv contains the hostname and the in memory key-value store
        # that is eventually written to
  # /opt/condor/etc/condor_config.local
  # plugins can add/change/remove keys from the store

  # 1. Get the hostname and the key-value store, which
  #    is a python dictionary
  host, kvstore = argv

  # The following would add CONDOR_SAMPLE=Sample Plugin
  # the key = value dictionary (kvstore)  that is written out
  #
  # Example 1. Read an attribute from the database and set
  # the values
  value = self.db.getHostAttr(host, 'Condor_HostAllow')
  kvstore['CONDOR_SAMPLE'] = value

  # Example 2. Set the key CONDOR_SAMPLE to the hostname
  kvstore['CONDOR_SAMPLE'] = host

  # Example 3. Remove a key from the dictionary
  if 'CONDOR_SAMPLE' in kvstore:
   del kvstore['CONDOR_SAMPLE']

RollName = "condor"
```

Users/Roll Developers can add their own plugins for the "report host condor config" command to overwrite, add, and/or delete key,value pairs that are written into /opt/condor/etc/condor_config.local.

In the above code sample, the Condor report command driver passes the hostname and the dictionary of already defined key,value pairs (kvstore in the sample code). The sample code shows several different examples of changing the key 'CONDOR_SAMPLE'.

Plugins are written in Python, are called in random order, and must be named "plugin_<name>.py".

Plugins also enable any desired configurations to be properly applied with the command `rocks sync host condor config`.

# Notes

1. http://www.cs.wisc.edu/condor/manual
2. condor-Manual

# Appendix A. Rocks® Copyright

# Appendix B. Third Party Copyrights and Licenses

This section enumerates the licenses from all the third party software components of this Roll. A "best effort" attempt has been made to insure the complete and current licenses are listed. In the case of errors or ommisions please contact the maintainer of this Roll. For more information on the licenses of any components please consult with the original author(s) or see the Rocks® CVS repository[1].

## B.1. Condor

The following is the software license for the version of Condor included in this Roll.

```
Copyright 1990-2007 Condor Team, Computer Sciences Department,
University of Wisconsin-Madison, WI.

                                 Apache License
                           Version 2.0, January 2004
                        http://www.apache.org/licenses/

   TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

   1. Definitions.

      "License" shall mean the terms and conditions for use, reproduction,
      and distribution as defined by Sections 1 through 9 of this document.

      "Licensor" shall mean the copyright owner or entity authorized by
      the copyright owner that is granting the License.

      "Legal Entity" shall mean the union of the acting entity and all
      other entities that control, are controlled by, or are under common
      control with that entity. For the purposes of this definition,
      "control" means (i) the power, direct or indirect, to cause the
      direction or management of such entity, whether by contract or
      otherwise, or (ii) ownership of fifty percent (50%) or more of the
      outstanding shares, or (iii) beneficial ownership of such entity.

      "You" (or "Your") shall mean an individual or Legal Entity
      exercising permissions granted by this License.

      "Source" form shall mean the preferred form for making modifications,
      including but not limited to software source code, documentation
      source, and configuration files.

      "Object" form shall mean any form resulting from mechanical
      transformation or translation of a Source form, including but
      not limited to compiled object code, generated documentation,
      and conversions to other media types.

      "Work" shall mean the work of authorship, whether in Source or
      Object form, made available under the License, as indicated by a
      copyright notice that is included in or attached to the work
      (an example is provided in the Appendix below).
```

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

   (a) You must give any other recipients of the Work or Derivative Works a copy of this License; and

(b) You must cause any modified files to carry prominent notices stating that You changed the files; and

(c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions. Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

```
8. Limitation of Liability. In no event and under no legal theory,
   whether in tort (including negligence), contract, or otherwise,
   unless required by applicable law (such as deliberate and grossly
   negligent acts) or agreed to in writing, shall any Contributor be
   liable to You for damages, including any direct, indirect, special,
   incidental, or consequential damages of any character arising as a
   result of this License or out of the use or inability to use the
   Work (including but not limited to damages for loss of goodwill,
   work stoppage, computer failure or malfunction, or any and all
   other commercial damages or losses), even if such Contributor
   has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing
   the Work or Derivative Works thereof, You may choose to offer,
   and charge a fee for, acceptance of support, warranty, indemnity,
   or other liability obligations and/or rights consistent with this
   License. However, in accepting such obligations, You may act only
   on Your own behalf and on Your sole responsibility, not on behalf
   of any other Contributor, and only if You agree to indemnify,
   defend, and hold each Contributor harmless for any liability
   incurred by, or claims asserted against, such Contributor by reason
   of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS
```

# Notes

1. http://cvs.rocksclusters.org