AMAZON-AWS CRONTAB

Version 1.0

User Manual

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **S. No** | **Document Version** | **Updated By** | **Reviewed By** | **Release Date** |
| 1 | 1.0.1 | Vishnu Moorthy K | Raja | 21-May-16 |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |

***Amazon-AWS Jar Deployment***

**Description:**

A Linux cron job directs the cron daemon to run one or more commands on a specified schedule. For example, suppose your stack supports a PHP e-commerce application. You can set up a cron job to have the server send you a sales report at a specified time every week. For more information about how to run a cron job directly on a Linux-based computer or instance, see [What are cron and crontab, and how do I use them?](https://kb.iu.edu/d/afiz) on the Indiana University knowledge base website.

Although you can manually set up cron jobs on individual Linux-based instances by connecting to them with SSH, and editing their crontab entries, a key advantage of AWS OpsWorks is that you can direct it to run the task across an entire layer of instances. The following procedure describes how to set up a cron job on a PHP App Server layer's instances, but you can use the same approach with any layer.

## Required Files:

## Create Base Folder under /home/ec2-user

## The Base Folder must contains the following files/folders

## “properties” folder – which contains what are the property files need for the corresponding jar (Database property, Log 4j property)

## “lib” folder – which contains the what are the dependency jars needed for the core

## “Executable Jar” – which performs the corresponding functionality using crontab

**Procedures:**

**Step -1:**

Open Putty and load the corresponding AWS instance

**Step -2:**

* Check the Current Directory using below command

**$ pwd**

* Use “**crontab**” command to create, view, edit & delete Cron Job.
* **$ crontab –e**  - This command is used to create a new Cron Job or edit the already existing cron job.
* **$ crontab –l** - This command is used to view the already existing Cron Job.

**Step -3:**

* Open cron job editor using **“$ crontab –e”** and type the require command lines.
* The corn job command line contains the 5 stars (\*) and the execution command.
* These 5 Stars represents
* **Minute**
* **Hour**
* **Day of Month**
* **Month**
* **Day of Week**
* For Ex : Here I create a Cron Job for executing a jar file for every 5 minutes.
  + **\*/5 \* \* \* \* java -jar /home/ec2-user/MagNet\_Documents/MagNet\_Mail\_Reader/hbsap-mail-reader-1.0.1.jar 2>&1 >> /home/ec2-user/MagNet\_Documents/MagNet\_Mail\_Reader/MailReaderCron.log &**
  + From this example, I can execute the jar with the corresponding location for every 5 minutes and write the jar’s console in the given log file.
* We can give the more than one Cron Jobs
  + **\*/5 \* \* \* \* java -jar /home/ec2-user/MagNet\_Documents/MagNet\_Mail\_Reader/hbsap-mail-reader-1.0.1.jar 2>&1 >> /home/ec2-user/MagNet\_Documents/MagNet\_Mail\_Reader/MailReaderCron.log &**
  + **\*/5 \* \* \* \* java -jar /home/ec2-user/MagNet\_Documents/MagNet\_Message\_Sender/hbsap-message-sender-1.0.1.jar 2>&1 >> /home/ec2-user/MagNet\_Documents/MagNet\_Message\_Sender/MailSenderCron.log &**