**STEPS for the Patching process for Linux servers and Windows servers**:

**Step1:**

Makes sure that all the instances are shown up in the **Managed instances** in the AWS different accounts.

**Managed Instances**: EC2 Systems Manager uses a light-weight agent installed on your EC2 **instances** and on-premises servers that communicates securely with the Systems Manager service and executes management tasks.

Systems Manager Prerequisite: All EC2 instances are installed with SSM Agent as well as having an IAM machine role. EC2 instances meeting these requirements will show up in Managed Instances.

AWS Documentation link 🡪 http://docs.aws.amazon.com/systems-manager/latest/userguide/systems-manager-setting-up.html

**Troubleshoot Managed Instances:**

**Windows EC2 Instance** – RDP onto the instance. Check Task Manager 🡪 Services to see if the Amazon SSM Agent and EC2 Config Services are running. If so, restart the services and check once again in AWS Managed Instances.

If not shown, download and install EC2Config. Then check to see that the EC2 Config and SSM Agent services are showing as running in the Task Manager.

AWS Documentation link for SSM Agent 🡪 http://docs.aws.amazon.com/systems-manager/latest/userguide/systems-manager-setting-up.html

**LINUX EC2 Instance**- Putty onto the instance.

Type the following commands (redhat or Amazon linux)- **status amazon-ssm-agent**, if you see the SSM agent running please do a restart – **restart amazon-ssm-agent.**

To Install the SSM Agent (REDHAT and Amazon Linux)- **yum install -y https://s3.amazonaws.com/ec2-downloads-windows/SSMAgent/latest/linux\_amd64/amazon-ssm-agent.rpm**

**SUSE**: ; zypper install -y https://s3.amazonaws.com/ec2-downloads-windows/SSMAgent/latest/linux\_amd64/amazon-ssm-agent.rpm; zypper info amazon-ssm-agent

**EC2 Instance Tag for Patching Activities**

See Wiki Link to Patching 🡪 https://wiki.cardinalhealth.net/AWS\_Patching

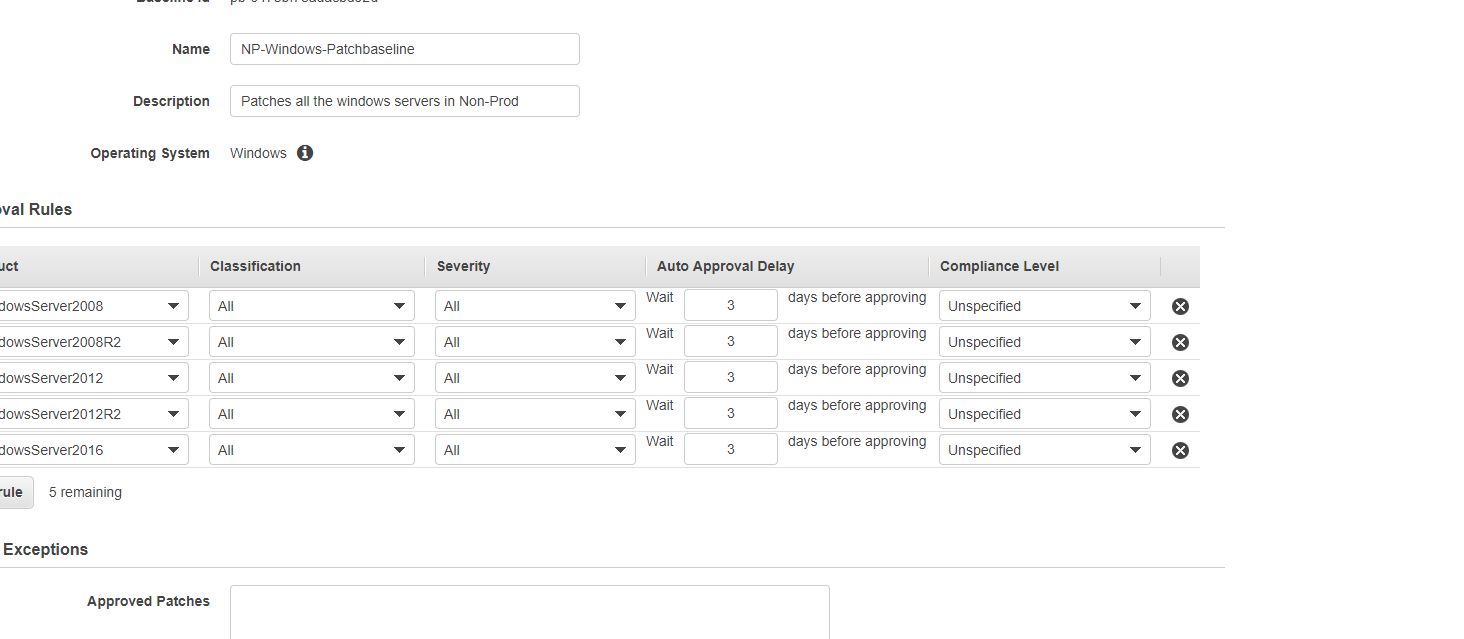
**PatchBaseline walk through**:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | Repository | | Scheduler | |
| Windows | AWS Patch Baselines | | AWS SSM - Maint Windows | |
|  |  |  |  |  |
| Amazon Linux | AWS Patch Baselines | | AWS SSM - Maint Windows | |

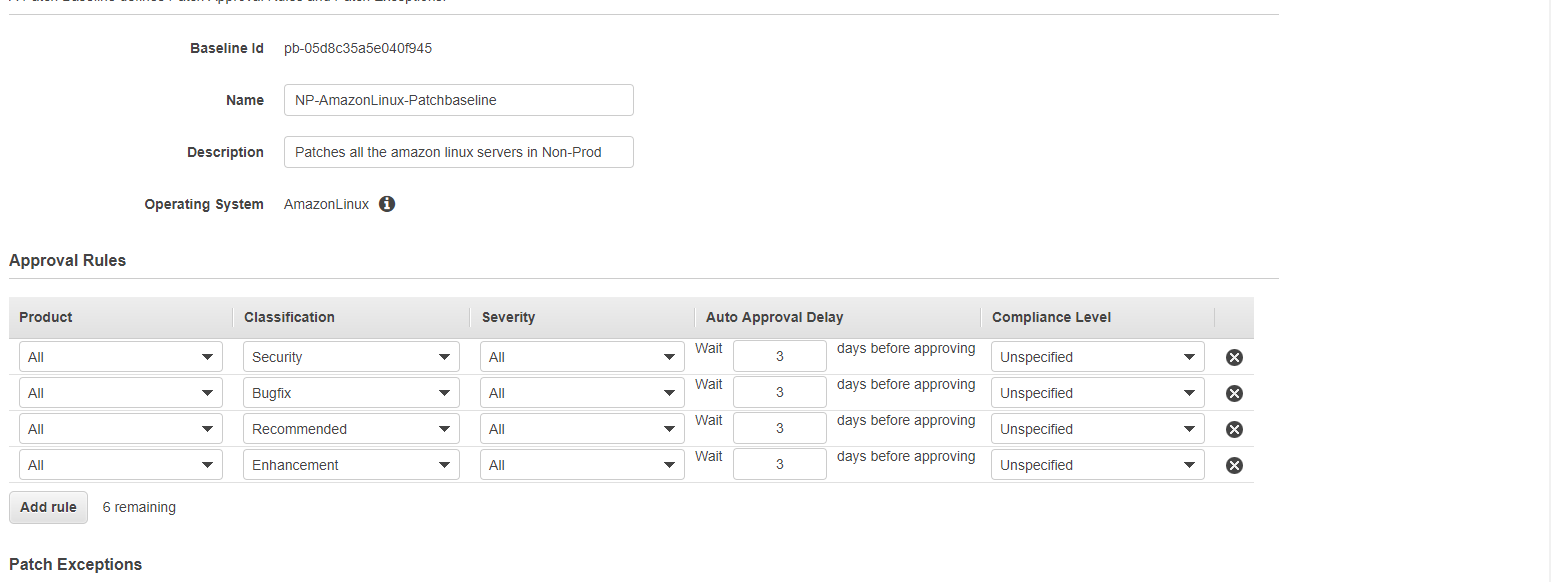
**For creating a patchbaseline please follow this link:**

<http://docs.aws.amazon.com/systems-manager/latest/userguide/sysman-patch-walkthrough.html#sysman-patch-consolewalk>

**WINDOWS PATCHBASLEINE :**



**LINUX PATCHBASELINE:**



**AutoApproval dates for Patch Baseline:**

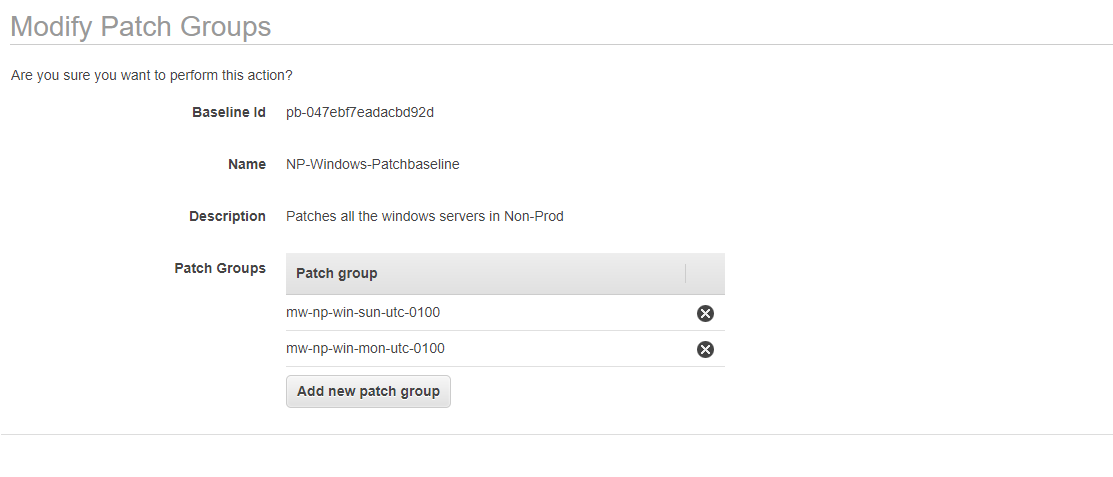
**Non-Prod-3 days**

**Stage- 6 days**

**Production- 14 days.**

**ADD Tags to Patchbaseline:**

**Under SystemManager Services -> PatchBaselines -> Actions ->Modify patch Groups -> add appropriate tags values.**



### Configuring Maintenance Windows:

### The EC2 instance tag, “Patch Group”, values are indicative of the maintenance window each EC2 instance tag value.

Key=Patch Group (yes there is a space here)

Value=*mw-os-day-time (Eg. mw-win-sun-2100)*

AWS Maintenance Task setup notes:

Create Maintenance Window

- Name: mw-win-<day>-<time> (eg. mw-win-sun-2100)

- Cron/Rate Expression: cron(0 01 ? \* 2 \*) (eg. this sets the job to run at 1am UTC EVERY Monday, which is 9PM Sunday. Tune this for the monthly runs)

<http://docs.aws.amazon.com/AmazonCloudWatch/latest/events/ScheduledEvents.html>

- Duration: 4 hours

- Stop Initiating tasks: 1 hour before window closes

**Register Targets**

- Owner Information = First.Last

- Specify Tags:

- Tag Name: Patch Group

- Tag Value: *mw-os-day-time (Eg. mw-win-sun-2100)*

**Register Tasks**

Register run task command for the following documents

Priority = 0 lnx-updatessm-rhel-azml-cent (Updates SSM Agent)

Priority = 1 lnx-bigfix-install (IBM patch compliance tool)

Priority = 2 lnx-crowdstrike-installlatest (Updates Anti-virus)

Priority = 3 AWS-RunPatchBaseline (Applies OS patch)

**Schedule Tasks (Windows)**

**Cloud-native Windows**

Register run task command for the following documents

Priority = 0 AWS-updateSSMAgent (Updates SSM Agent)

Priority = 5 Win-cloudstrike-install (Updates Anti-virus)

Priority = 6 Win-bigfix-install (BM patch compliance tool)

Priority = 9 AWS-RunPatchBaseline (Applies OS patch)

**Lift-and-shift Windows**

Register run task command for the following documents

Priority 1 win-wmf51-install

Priority 5 win-crowdstrike-install

Priority 6 win-bigfix-install

Priority 9 AWS-RunPatchBaseline

Priority 3 AWS-UpdateSSMAgent

Priority 0 win-dotnet45-install

Priority 4 win-awspstools-install

Priority 2 win-getuptime

**Step2(ON SCHEDULED PATCH DAY)**

8PM EST – Trigger off the Lambda functions “**StartPatch-amazl”** which start all the servers. The servers that are started by the Lambda function will create tags to the severs. The trigger is set for 8PM EST.

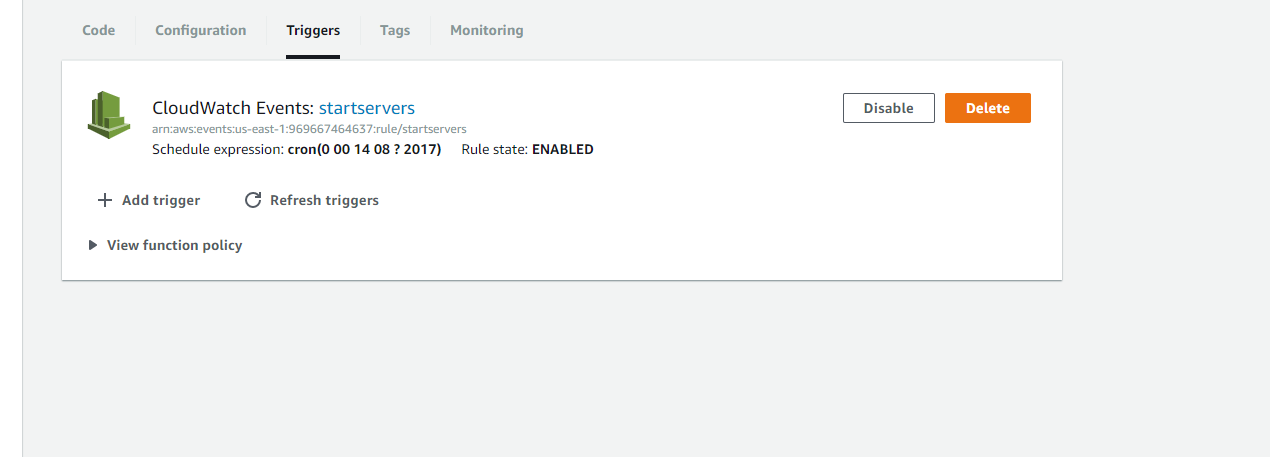
This are the tags the lambda function will leave on the instances that are started by the function

Key':'LinuxPatchingstatedtime','Value':'linuxlatestpatch

Key': 'WindowsPatchingstatedtime', 'Value': 'windowslatestpatch'

**AWS SERVICES->LAMBDA-FUNCTIONS**

Add the trigger to the Lambda function in the triggers tab.



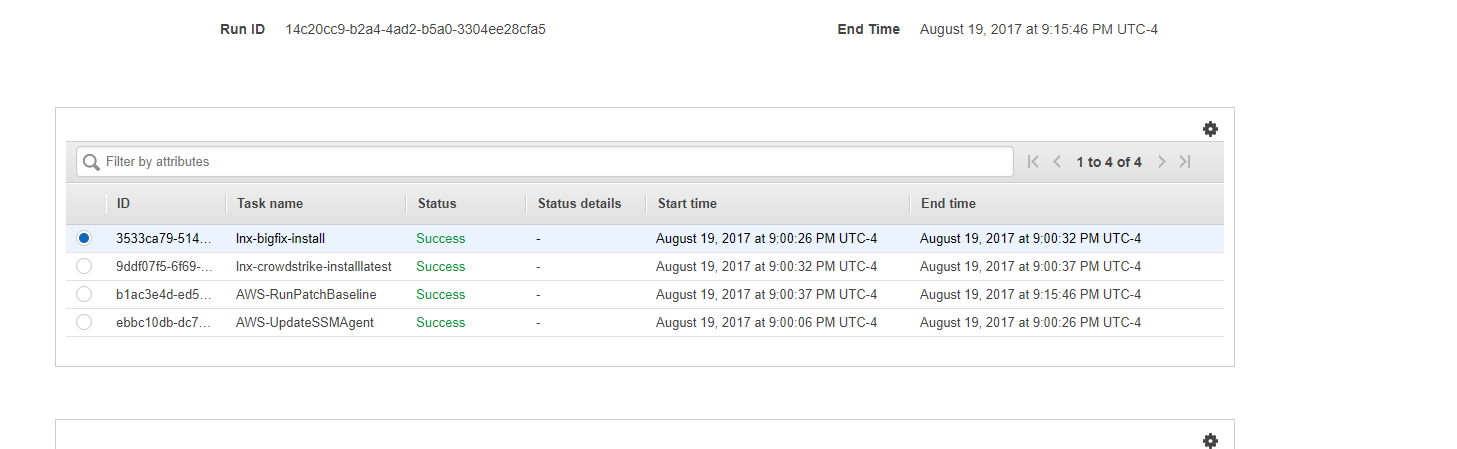
9PM EST – Systems Manager kicks off the Maintenance Windows as defined in the cron of the maintenance windows themselves.

**To view progress/failures of the patching process(Linux):**

Select the maintenance window, lower in pane, view the “History” tab. You can see the overall progress of the patching efforts and can drill down to any failures.

If one of the processes seems stuck - Example, more than 40-50 minutes, cancel that document as long as it is not executing the AWS-Runpatchbaseline doc.

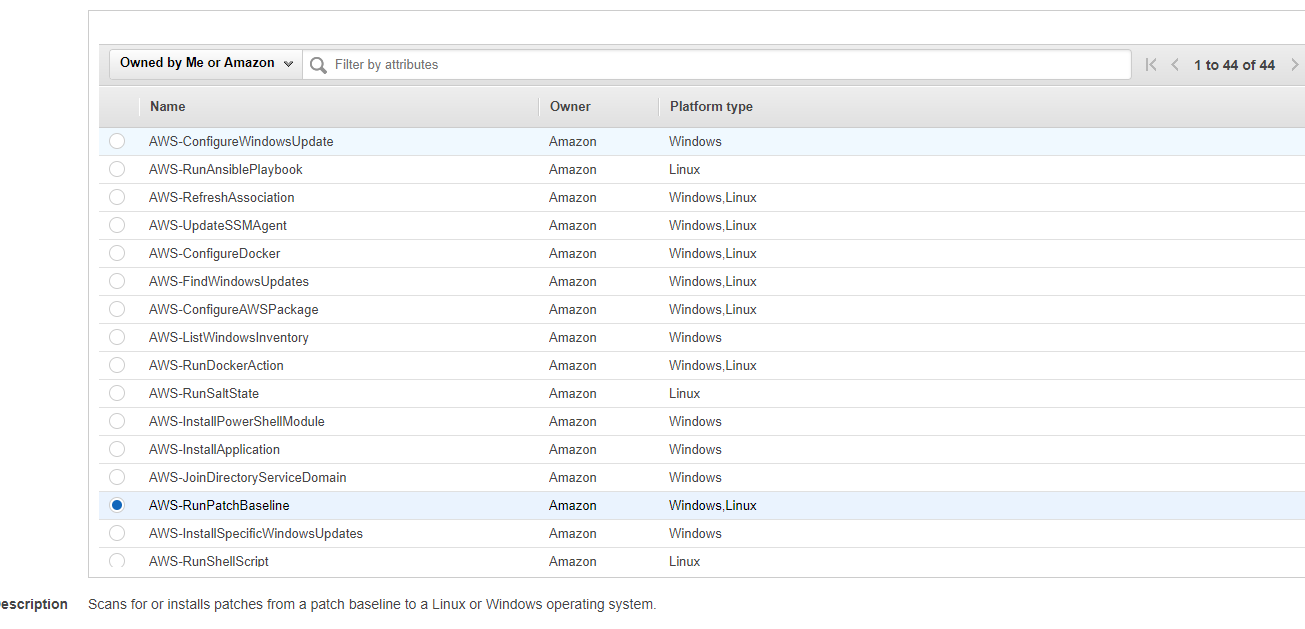
From History->Details->View Details



Then if you see a task executing for more time just click the task and go below for view details which will redirect you to run command and shows the process executing. Identify the task and Click->Action->Cancel document which cancel the task and let the below task to execute.

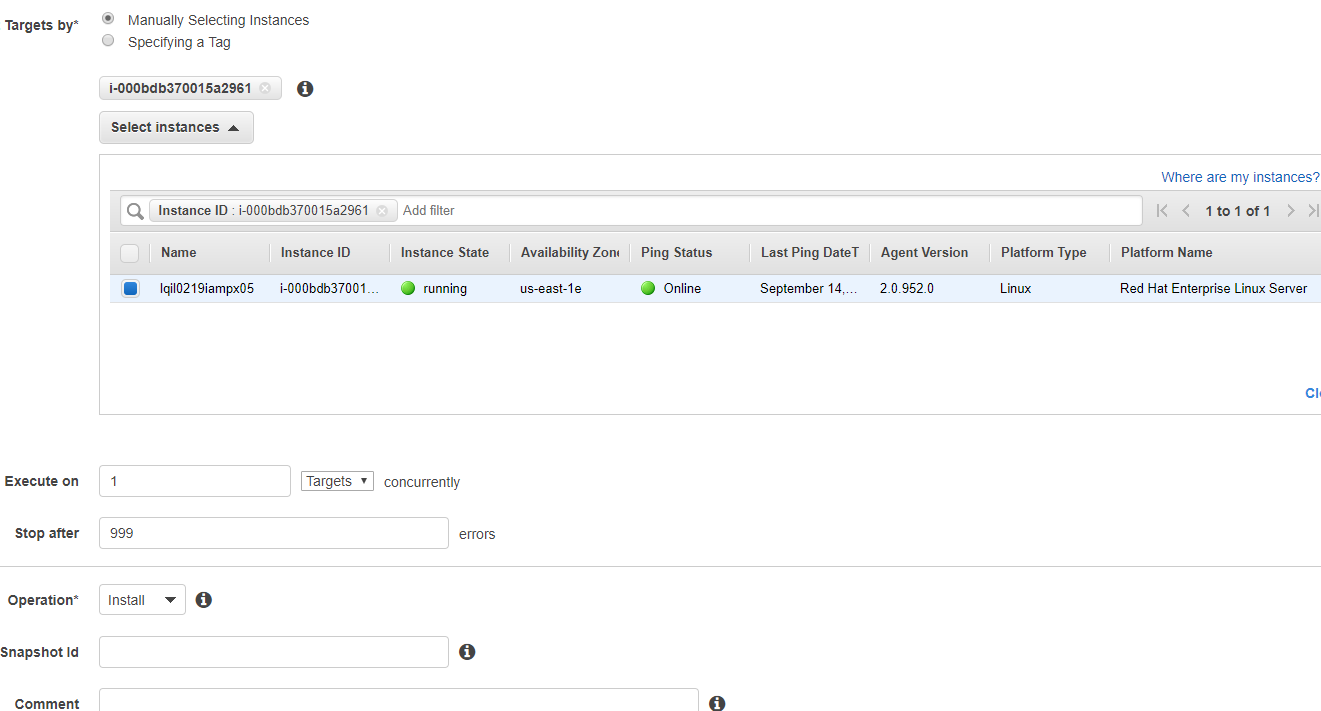
For the Instances failed the AWS-RUNPatchBaseline:

-Try to update them manually through run command. In AWS, under Systems Manager Services, click Run Command.



-Select the AWS-Runpatchbaseline document.

See the below example:



-If the patching through run command also fails, putty the instance and do a manual yum update.

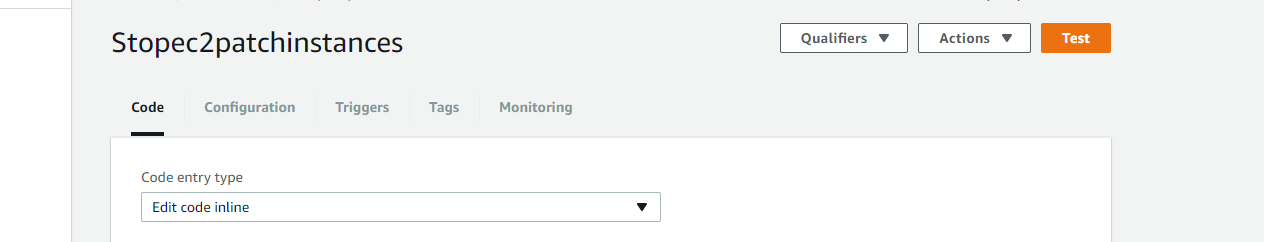
**Step5**:

**At 1am EST the Lambda functions again trigger to stop the instances which have the tags**

Key':'LinuxPatchingstatedtime','Value':'linuxlatestpatch

Key': 'WindowsPatchingstatedtime', 'Value': 'windowslatestpatch'.

-The cron time is set in the triggers as 1pm EST.



**WINDOWS TROUBLESHOOTING STEPS:**

1. Watch the output of each step, since all instances must finish or fail a step I ended up forcing some instances to fail.
2. Disk space!  I had quite a few instances that could not apply updates or run a step because the C drive was out of space.
3. If I failed a server on a step, I tried to go back and manually run the commands on the servers via ssm run command.
4. I had a few servers where windows update services were disabled, or set to never check.
   1. Re-enabled and started the service if it was not set
   2. For a few that I manually updated, opened windows update and enabled it, but did not allow it to download or install automatically.
   3. I don’t think I had to for las, but in some cases I have had to repair widows update itself or cleanup the cache files.
      1. Stop windows update service
      2. Delete files in C:\Windows\SoftwareDistribution\Download
      3. Start windows update service
5. Win 2003 servers cannot be patched via this method- manually patching should be done.