HPC Application Support

HPC component access, debug, update, and rollbacks. *Revision 2.0*

Table of Contents

Revision History

Version	Name	Reason For Changes	Date
1.0	*Larry Johnson*	*Initial Revision*	*2/18/20*
2.0	*Larry Johnson*	*Close-Out*	*2/24/20*

Approved By

Approvals should be obtained for project manager, and all developers working on the project.

Name	Signature	Department	Date

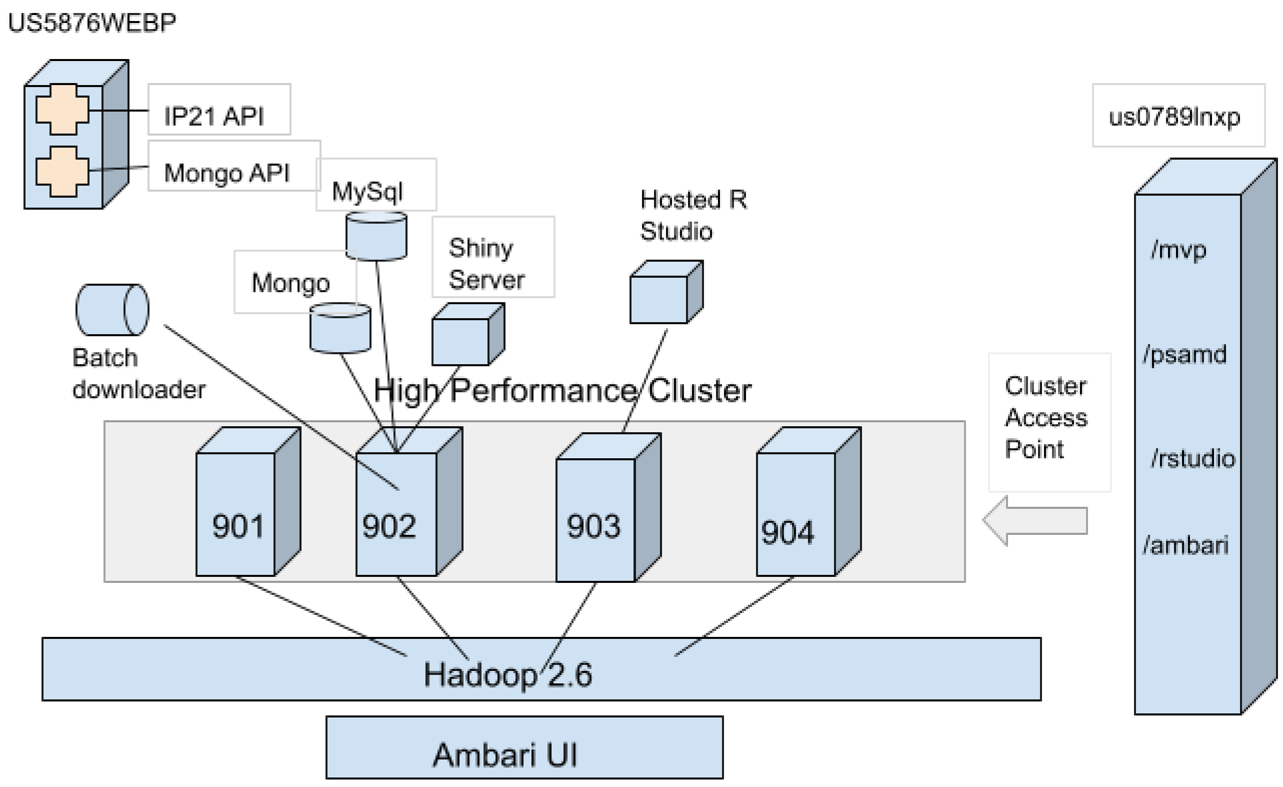
1. Introduction

1.1 Purpose

In this document we'll break down the System Overview (fig. 1.2) into its various components and features. Component and Feature will be used interchangeably going forward. For each component we'll document:

- How to access the component
- How to determine the current status
- How to stop, start, or restart
- How to review any logs, if available
- How decompose the component and troubleshoot it's various parts

1.2 System Overview



.

1.3 List of Components or Features

Below is a list of features from figure(1.2):

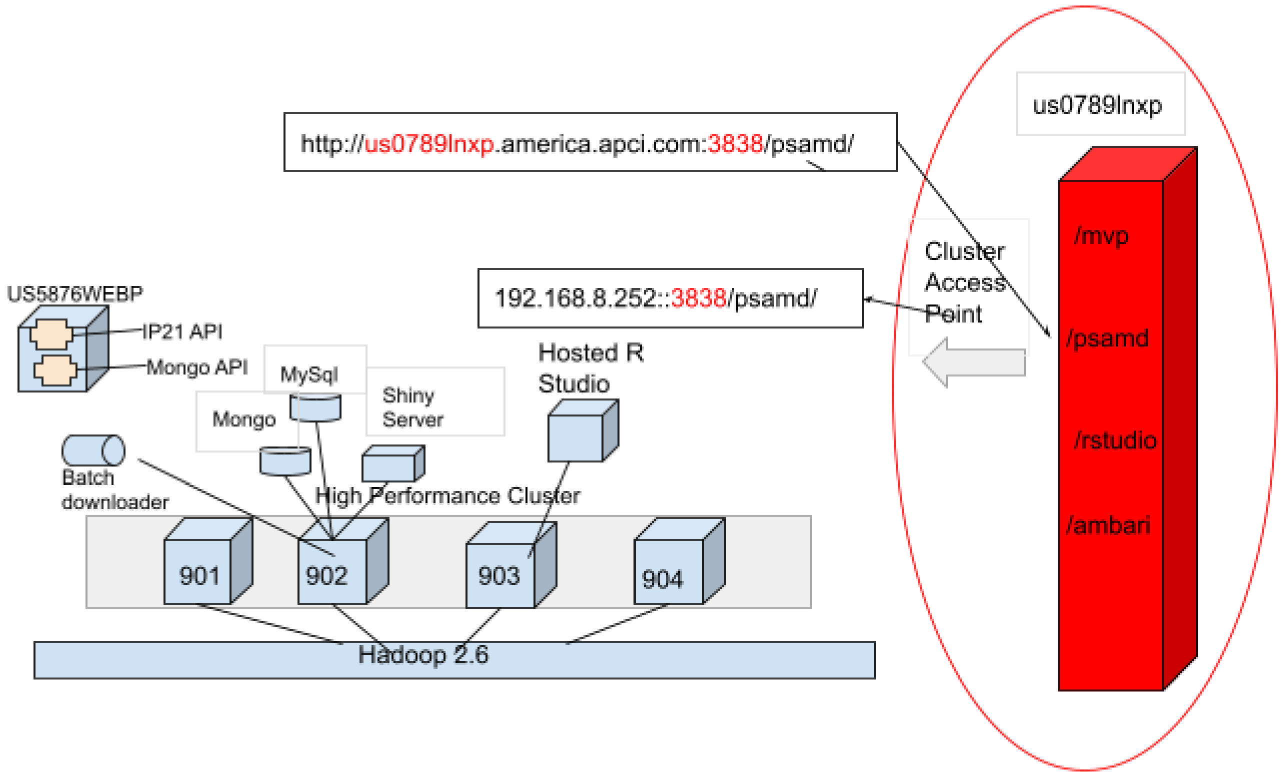
- **Port Forwards** - It's not shown very well but if you look at box us0789lnxp.america.apci.com those "/" directories are only accessible due to a series of port forwards to the relevant compute90* box which contains the software that needs to be accessed. A port forward from us0789lnxp will allow you access to any node.
- **us0789lnxp.america.apci.com -** Is our "jump" node, meaning it allows us to access the HPC Cluster (the 4 compute nodes). To access the cluster you first have to access the jump node.(use can use an ssh client like PUTTY to do that.
- **HPC Cluster**- consist of 4 Intel(R) Xeon(R) CPU E5-2683 v4 @ 2.10GHz servers. I set them up as an Hadoop cluster with Ambari for the UI. The servers, running RedHat 7.2, have 64 cpu's for each box and 256 GB of memory per box, but that's actually 32 hyperthreaded cores.
- **Hadoop -** The HPC cluster is running Hadoop 2.6, you can use the Ambari UI , use the "Host" link in the top navigation bar to see what is running on each box. The details on how to log onto Ambari and navigate through the services are in the Ambari UI section.
- **Ambari UI** - As I indicated above this UI is for configuring and maintaining your HPC cluster. There is a tutorial for Ambari below.

- **Shiny Server -** We use Shiny to publish our apps. You only need to create a new directory and transfer your Shiny App to that directory and the server will pick up the change.
- **MySQL -** We use MySql as a datastore for data pulled from IP21
- **Mongo -** Mongo is used as a configuration database for each plant. It also holds the current state of data pulled from each plant.
- **Batch Downloader -** There is a nightly cron job that backs up the mysql db and pulls down the previous days delta from each plant.
- **US5876WEBP -** This is a windows server we use to host our Node JS API's.

1.4 Overview

This section provides an overview of the structural and functional decomposition of the system. Focuses on the support details of the particular components. Including information on the major responsibilities and roles the component must play.

1.5 US0789LNXP



1.6 US0789LNXP Support Details

- **Role**

    - us0789lnxp is used to allow access to the cluster and related applications
- **Basic Functionality**
    - In order to allow access to a port on 901-4 we use ssh to tunnel through us0789lnx using port forwards
- **Access**
    - All users able to login into us0789lnxp can set port forwards
- **Troubleshooting**
    - If you lose access to a url (http://us0789lnxp.america.apci.com:3838/psamd/) or API or database,
    - You can check that its not in the list of port forwards below, if so you can check its status and restart if necessary
- **Status**
    - on us0789lnxp you can run the process list command and grep (search) for your port
    - for example, to check the psamd app listed at the url above
    - Run: ps -ef | grep 3838
        - You'll see the following results:
    - johnsol3 48986   1 0 2017 ?   00:00:37 ssh -nfNT -L us0789lnxp.america.apci.com:3838:192.168.8.252:3838 root@192.168.8.252
    - johnsol3 75215   1 0 Jan11 ?   00:00:00 ssh -nfNT -L us0789lnxp.america.apci.com:8600:192.168.8.253:3838 root@192.168.8.253
    - You see there is an running tunnel from us0789lnxp:3838 to 192.168.8.252 (902)
    - The other is Rstudio being hosted on 903 I believe
- **Restart**
    - To restart just run the command as you see listed below

Apache Yarn is the successor to map-reduce which allows you to break up a job into smaller jobs and submit those jobs to separate CPU's for parallel computation.

Port Forwards

First you want to log into us0789lnxp and check that all services are running:
ps -ef|grep fNT

HBase REST Server (optional)
ssh -nNT -L us0789lnxp.america.apci.com:8082:192.168.8.251:8080 root@192.168.8.251

Job History Service - Both Ambari Yarn - MapReduce use this port
ssh -nfNT -L us0789lnxp.america.apci.com:19888:192.168.8.252:19888 root@192.168.8.252

NodeManager Webapp Address- Ambari Yarn
ssh -nfNT -L us0789lnxp.america.apci.com:8042:192.168.8.252:8042 root@192.168.8.252

NodeManager Webapp Address- Ambari Yarn
ssh -nfNT -L us0789lnxp.america.apci.com:18042:192.168.8.250:8042 root@192.168.8.250

Ambari UI

```
 ssh -fnNT -L us0789lnxp.america.apci.com:8180:192.168.8.253:8180 root@192.168.8.253
```

Submit job to cluster /Join cluster

```
ssh -nNT -L us0789lnxp.america.apci.com:7077:192.168.8.253:7077 root@192.168.8.253
```

NodeManager Webapp Address

```
ssh -nNT -L us0789lnxp.america.apci.com:8042:192.168.8.252:8042 root@192.168.8.252
```

Shiny Server

```
ssh -nfNT -L us0789lnxp.america.apci.com:3838:192.168.8.252:3838 root@192.168.8.252
```

Rstudio online host

```
ssh -nfNT -L us0789lnxp.america.apci.com:8787:192.168.8.251:8787 root@192.168.8.251
```

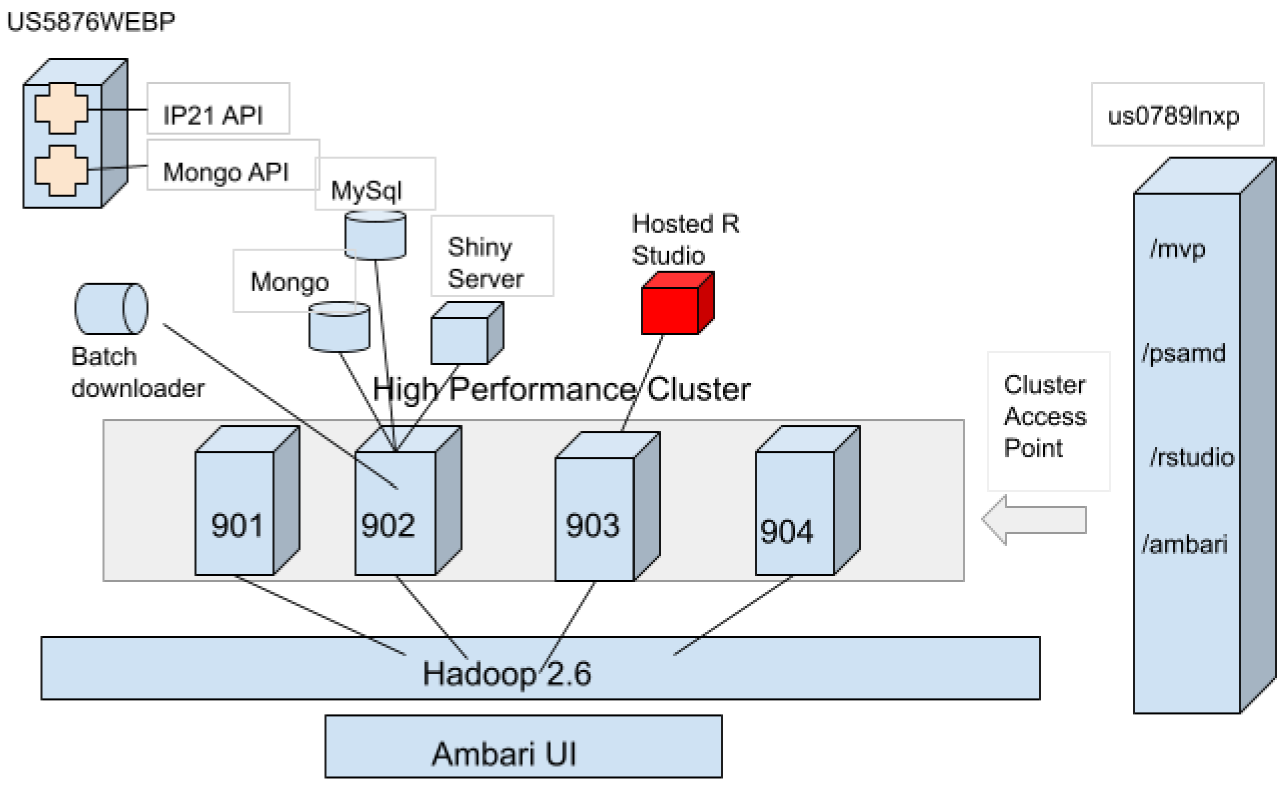
```
ssh -nfNT -L us0789lnxp.america.apci.com:8500:192.168.8.250:8500 root@192.168.8.250
```

Shiny Server on 253

```
ssh -nfNT -L us0789lnxp.america.apci.com:8600:192.168.8.253:3838 root@192.168.8.253
```

Hbase Rest Server on 252

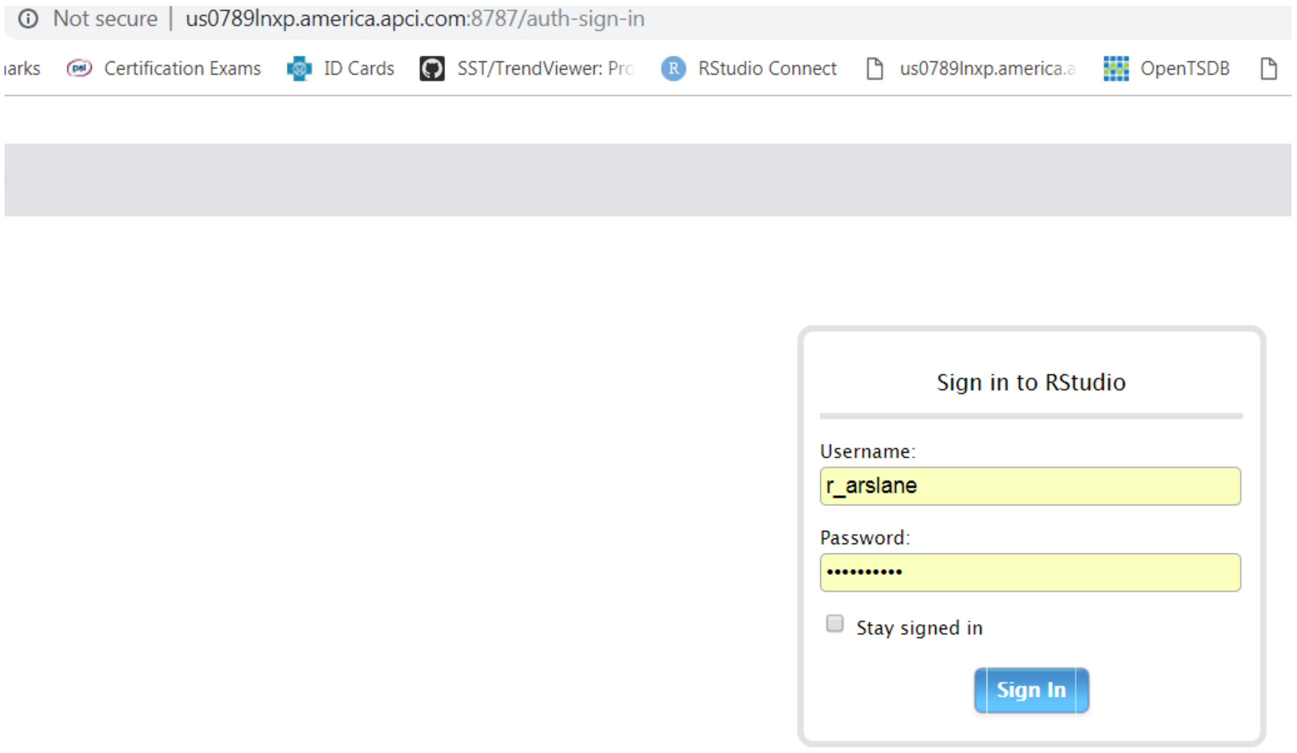
```
ssh -nfNT -L us0789lnxp.america.apci.com:8083:192.168.8.252:8080 root@192.168.8.252
```



2. Hosted R Studio Support

There is a version of RStudio running on box 903 that allows users to be able to develop in R to a cloud like environment, making development, publishing and sharing of apps easier.

The server can be accessed via http://us0789lnxp.america.apci.com:8787



You will have the same IDE as you would in your local R Studio

- The only catch here is if you install a new library you have to install it on the server.
    - log onto 903 run R
    - Then identify the correct library, "libPaths()" will give you a list of any R libraries.
    - I like to cd to each lib to see which is active.
    - Then install the package "install.packages("ggplot2", lib="/data/Rpackages/")"

3. Ambari Support

In the system we will review the Apache Ambari which we use as an UI. The project is aimed at making Hadoop management simpler by developing software for provisioning, managing, and monitoring Apache Hadoop clusters. Ambari provides an intuitive, easy-to-use Hadoop management web UI backed by its RESTful APIs.

Ambari enables System Administrators to:

- Provision a Hadoop Cluster
    - Ambari provides a step-by-step wizard for installing Hadoop services across any number of hosts.
    - Ambari handles configuration of Hadoop services for the cluster.
- Manage a Hadoop Cluster
    - Ambari provides central management for starting, stopping, and reconfiguring Hadoop services across the entire cluster.
- Monitor a Hadoop Cluster
    - Ambari provides a dashboard for monitoring health and status of the Hadoop cluster.
    - Ambari leverages Ambari Metrics System for metrics collection.
    - Ambari leverages Ambari Alert Framework for system alerting and will notify you when your attention is needed (e.g., a node goes down, remaining disk space is low, etc).

Ambari enables Application Developers and System Integrators to easily integrate Hadoop provisioning, management, and monitoring capabilities to their own applications with the Ambari REST APIs.

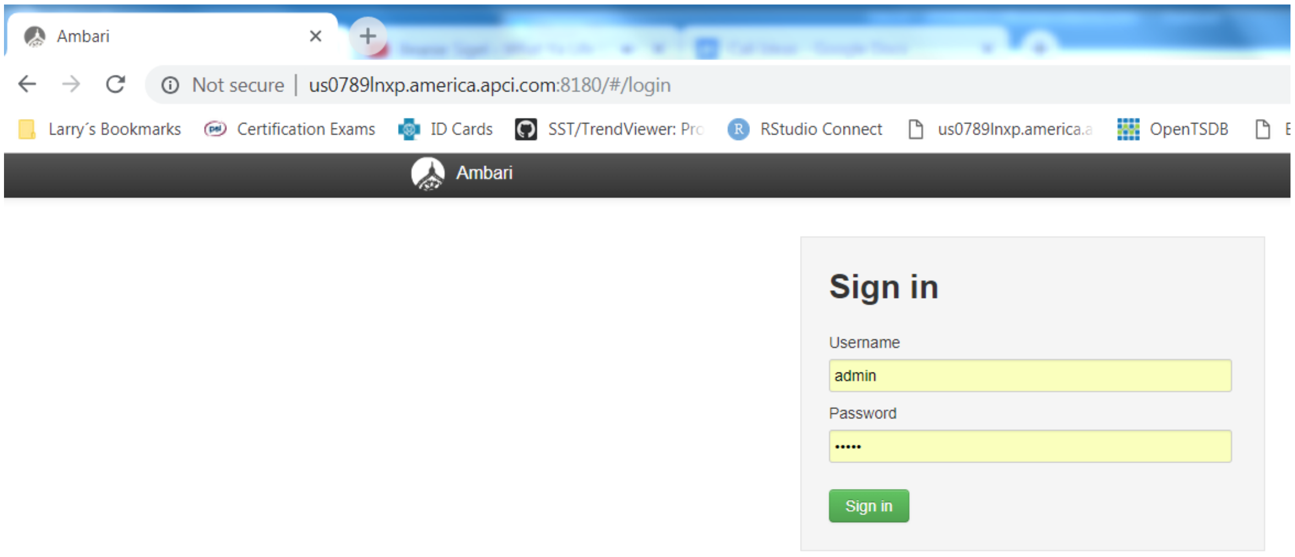
Let's review the critical systems that can crash the system, but can be restored through Ambari.
Here we are assuming we don't have access to the system or it's not functioning correctly.
The Shiny UI may be up but none of the inputs are working.

For Ambari you need to make sure the following are in green status:

- HDFS Service
- Zoo Keeper
- Spark2
- Yarn

You login using admin/admin:

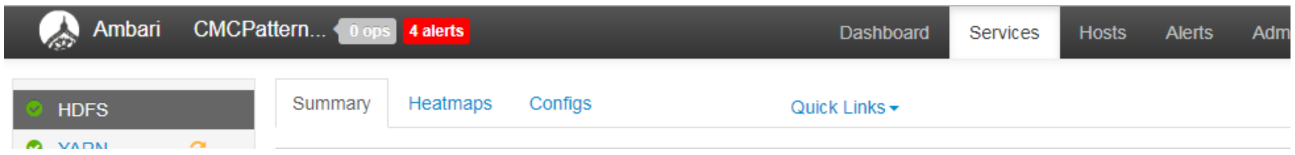
http://us0789lnxp.america.apci.com:8180/#/login



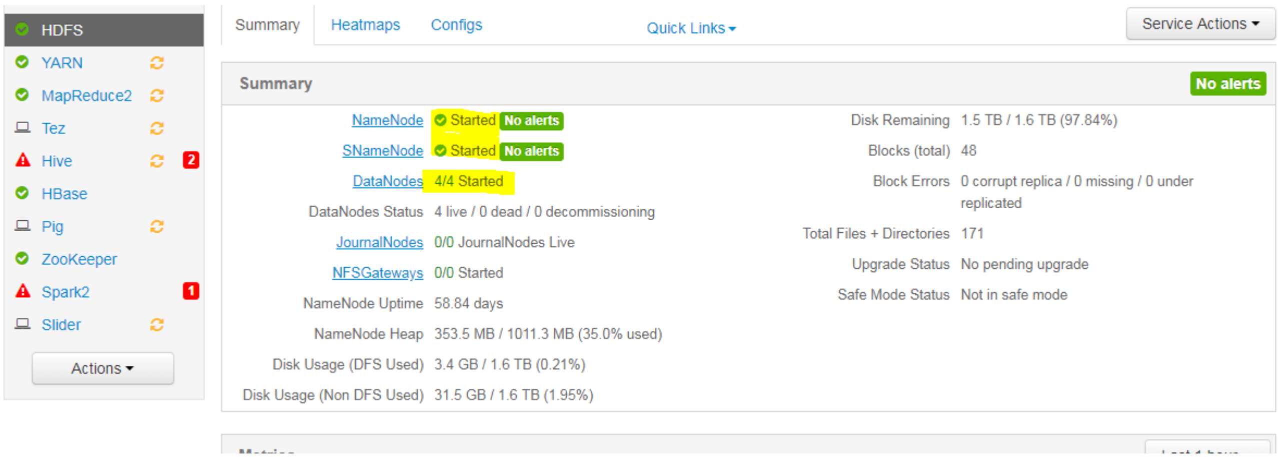
Ambari HDFS

facilitate using a network of many computers to solve problems involving massive amounts of data and computation. It provides a software framework for distributed storage and processing of big data using the MapReduce programming model.

You can select the HDFS service either through the menu on the left or the services navigation bar:



For HDFS the config needs to look like this:

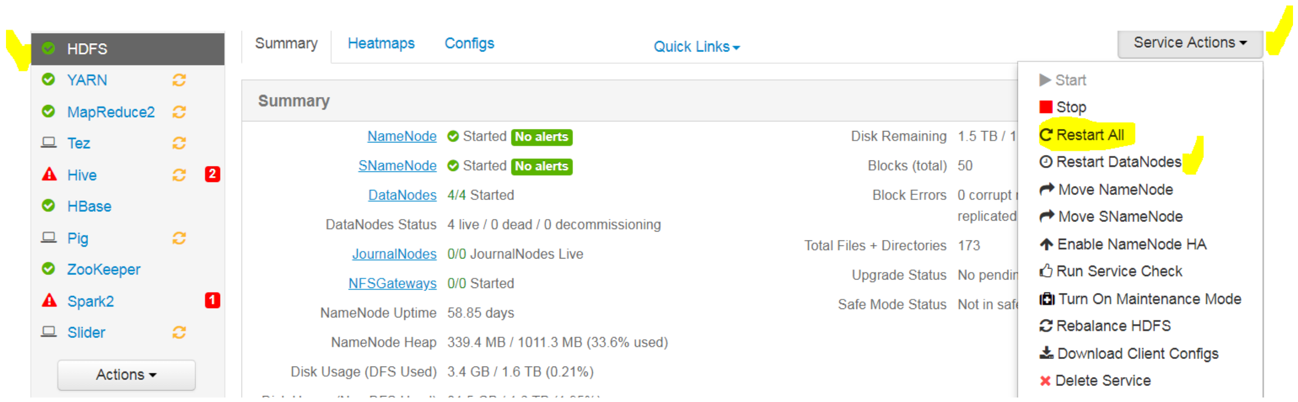


Summary — No alerts

NameNode: Started — No alerts
SNameNode: Started — No alerts
DataNodes: 4/4 Started
DataNodes Status: 4 live / 0 dead / 0 decommissioning
JournalNodes: 0/0 JournalNodes Live
NFSGateways: 0/0 Started
NameNode Uptime: 58.84 days
NameNode Heap: 353.5 MB / 1011.3 MB (35.0% used)
Disk Usage (DFS Used): 3.4 GB / 1.6 TB (0.21%)
Disk Usage (Non DFS Used): 31.5 GB / 1.6 TB (1.95%)
Disk Remaining: 1.5 TB / 1.6 TB (97.84%)
Blocks (total): 48
Block Errors: 0 corrupt replica / 0 missing / 0 under replicated
Total Files + Directories: 171
Upgrade Status: No pending upgrade
Safe Mode Status: Not in safe mode

- The NameNode needs to be started and green (it will be red if there is a problem)
- The SNameNode Needs to be started and green)
- DataNodes needs to be 4/4 started

If your services are showing RED, the solution is to restart the services, using the Service Actions menu on the far right side.



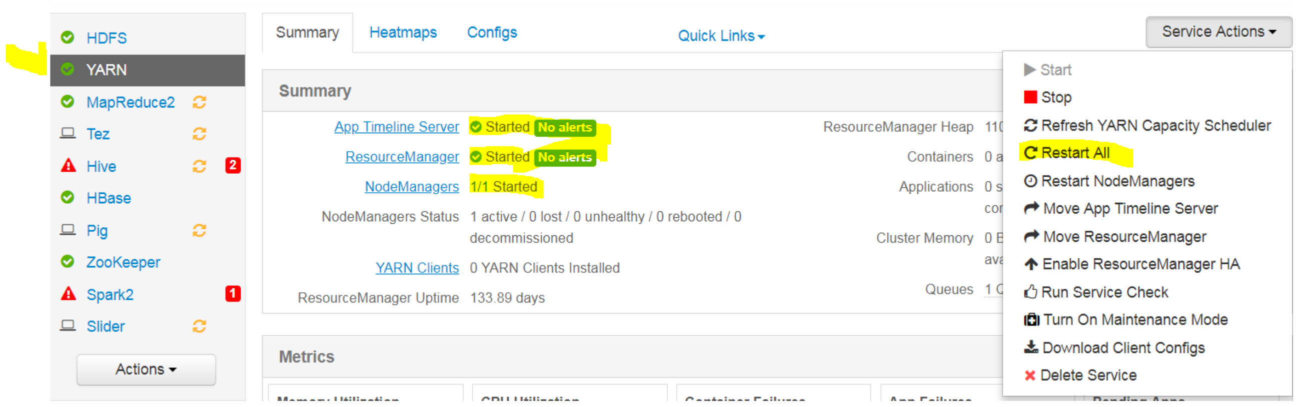
Summary

NameNode: Started — No alerts
SNameNode: Started — No alerts
DataNodes: 4/4 Started
DataNodes Status: 4 live / 0 dead / 0 decommissioning
JournalNodes: 0/0 JournalNodes Live
NFSGateways: 0/0 Started
NameNode Uptime: 58.85 days
NameNode Heap: 339.4 MB / 1011.3 MB (33.6% used)
Disk Usage (DFS Used): 3.4 GB / 1.6 TB (0.21%)
Disk Remaining: 1.5 TB / 1
Blocks (total): 50
Block Errors: 0 corrupt replicated
Total Files + Directories: 173
Upgrade Status: No pendi
Safe Mode Status: Not in saf

I placed a yellow check mark on the Service Action menu on the right side. Each Service Action is different depending on which page you are on. As you can see I placed a check mark on the left side to emphasize we are on the HDFS page. So the context of the Service Action menu on the left has to do with HDFS DataNodes. You can just restart the DataNodes (I placed a check there) or you can Restart All. I highlighted that as the go to option. Restarting all should get you back in green if not a pop-up will indicate what is wrong.

Ambari Yarn

In effect Yarn has replaced mapreduce which splits a task per cpu and then aggregate all the work from each core and produces the final answer.



Summary

App Timeline Server: Started No alerts
ResourceManager: Started No alerts
NodeManagers: 1/1 Started
NodeManagers Status: 1 active / 0 lost / 0 unhealthy / 0 rebooted / 0 decommissioned
YARN Clients: 0 YARN Clients Installed
ResourceManager Uptime: 133.89 days
ResourceManager Heap: 11
Containers: 0 a
Applications: 0 s co
Cluster Memory: 0 E ava
Queues: 1 C

Your Yarn should show green in the Summary. If not you can use the service action to Restart All.

Ambari Hbase

The same technique used for AMBARI HDFS and AMBARI YARN will work for AMBARI HBASE

Ambari Zookeeper

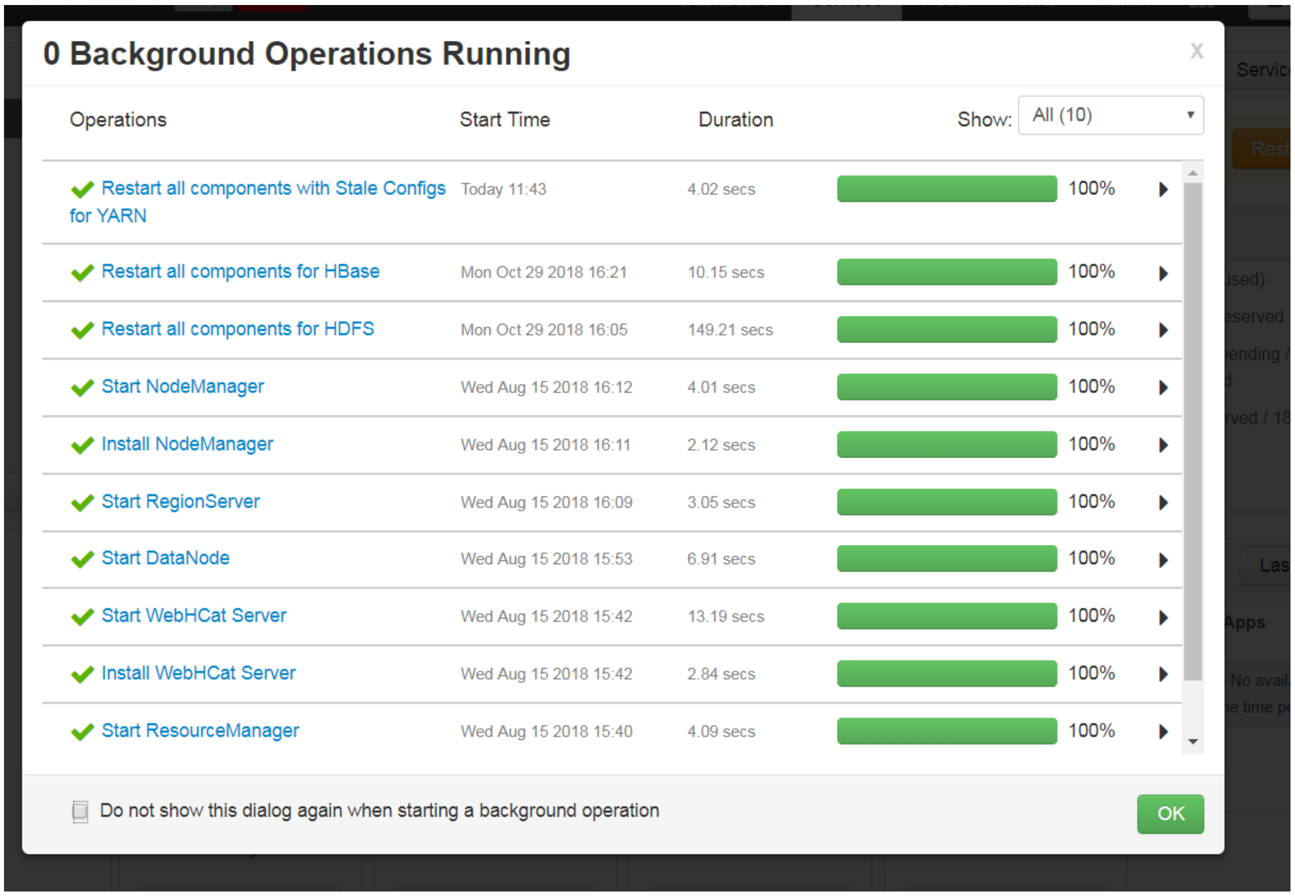
The same technique used for AMBARI HDFS and AMBARI YARN will work for AMBARI ZooKeeper

Ambari Spark

The same technique used for AMBARI HDFS and AMBARI YARN will work for AMBARI SPARK

Ambari Errors

When you do a restart of a component you will see a pop-up like this:

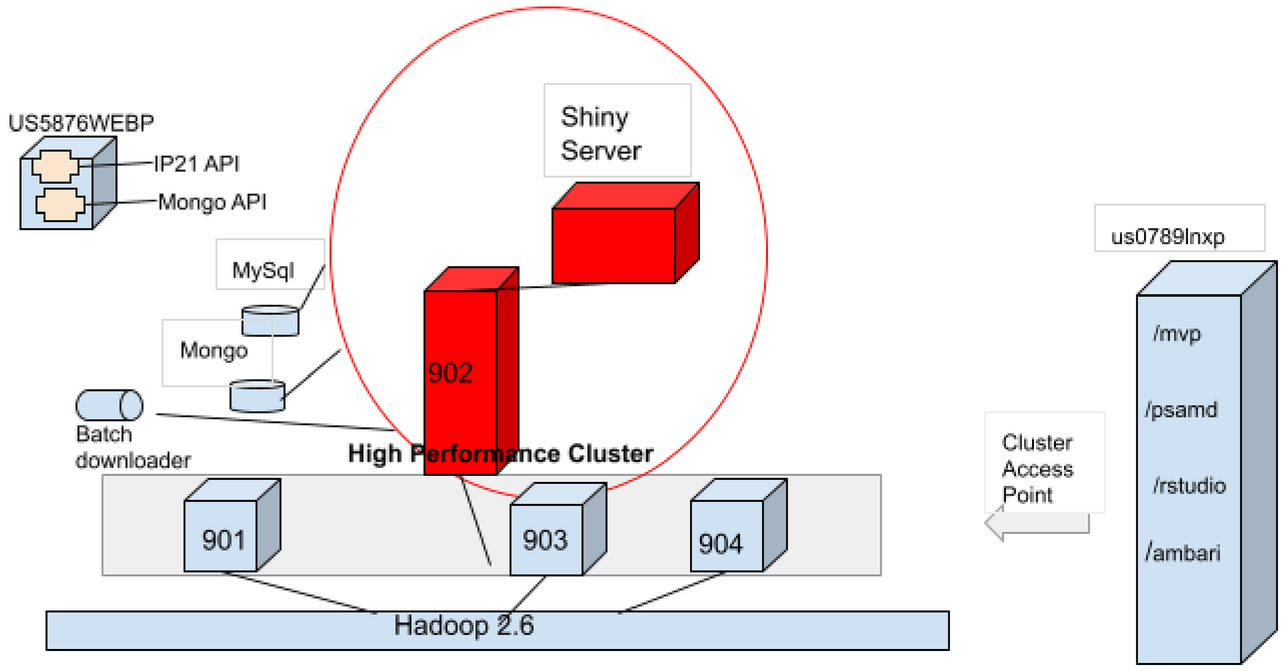


0 Background Operations Running

Operations	Start Time	Duration	Show: All (10)
✓ Restart all components with Stale Configs for YARN	Today 11:43	4.02 secs	100%
✓ Restart all components for HBase	Mon Oct 29 2018 16:21	10.15 secs	100%
✓ Restart all components for HDFS	Mon Oct 29 2018 16:05	149.21 secs	100%
✓ Start NodeManager	Wed Aug 15 2018 16:12	4.01 secs	100%
✓ Install NodeManager	Wed Aug 15 2018 16:11	2.12 secs	100%
✓ Start RegionServer	Wed Aug 15 2018 16:09	3.05 secs	100%
✓ Start DataNode	Wed Aug 15 2018 15:53	6.91 secs	100%
✓ Start WebHCat Server	Wed Aug 15 2018 15:42	13.19 secs	100%
✓ Install WebHCat Server	Wed Aug 15 2018 15:42	2.84 secs	100%
✓ Start ResourceManager	Wed Aug 15 2018 15:40	4.09 secs	100%

Do not show this dialog again when starting a background operation

We are only concerned with the top one; the rest are historical. If the operation does not go to 100% you can click on it and get the std output and std error that occurred. From this you can diagnose the problem.

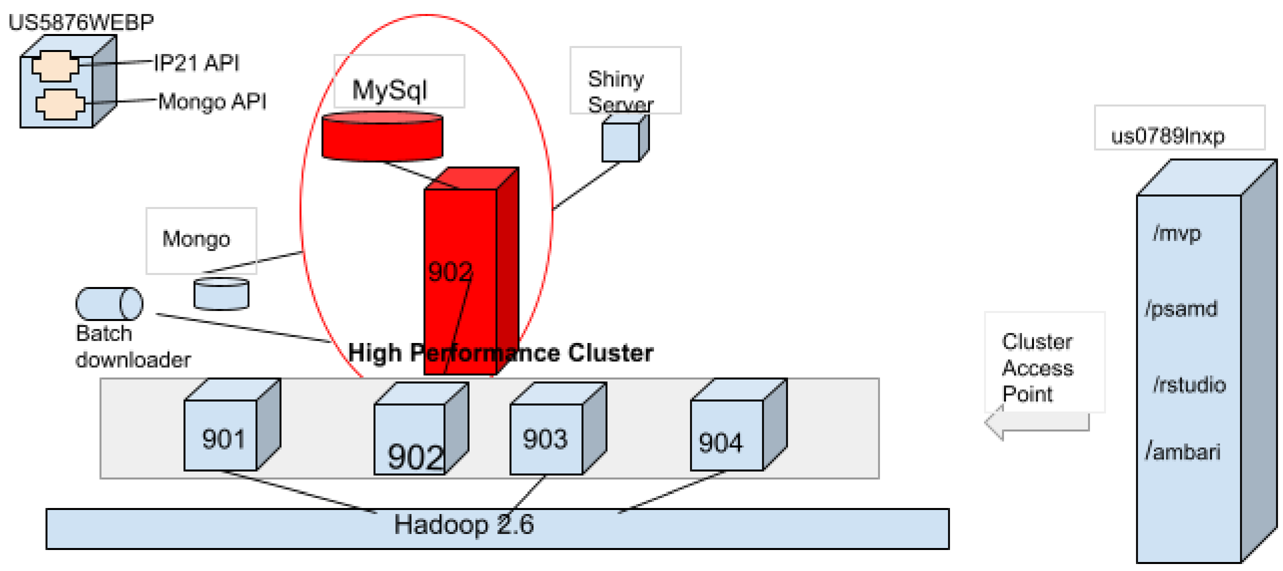


3.1 Shiny Server Support Details

- **Role**
  - Currently the shiny server host most front-end apps on the HPC
- **Basic Functionality**
  - The Shiny Server runs on box 902 on port :3838
  - The port is externally accessible by tunneling through us0789lnxp (see above)
- **Access**
  - To access the shiny-server you'll need to logon to 902 as root.
  - The hosted directories are located at /srv/shiny-server/
- **New Apps**
  - When you create a new locally on your box.
  - To transfer to Shiny you first have to make a new directory for your app.
  - To create a new directory logon to server 902
  - cd to directory /srv/shiny-server "cd /srv/shiny-server"
  - make a new director 'mkdir /myappName
  - The app directory will be owned by root initially.
  - If you want to write back to you app you'll need to change the owner
  - Shiny runs as "hdfs' user so you'll need to run 'chown hdfs <yourappname>
  - Files also have a group they belong to, you can leave it as root or change it.
  - To change it run "chgrp hdfs <yourappname>
- **Troubleshooting**
  - If you lose access to any of the hosted apps on 902 (http://us0789lnxp.america.apci.com:3838/psamd/)
  - You can check the status of the shiny server and restart if necessary
  - The config file is located at /etc/shiny-server/shiny-server.conf
  - The log files are located at /var/log/shiny-server/
- **Status**

    - To check the status logon to 902 and run the “service” command
    - [root@compute902 ~]# service shiny-server status
    - You should see something similar to : Active: active (running) since Thu 2018-03-01 14:17:46 EST; 2 weeks 0 days ago
- **Restart**
    - To restart just run the command as you see listed below
    - [root@compute902 ~]# service shiny-server restart

3.2 MySql Server

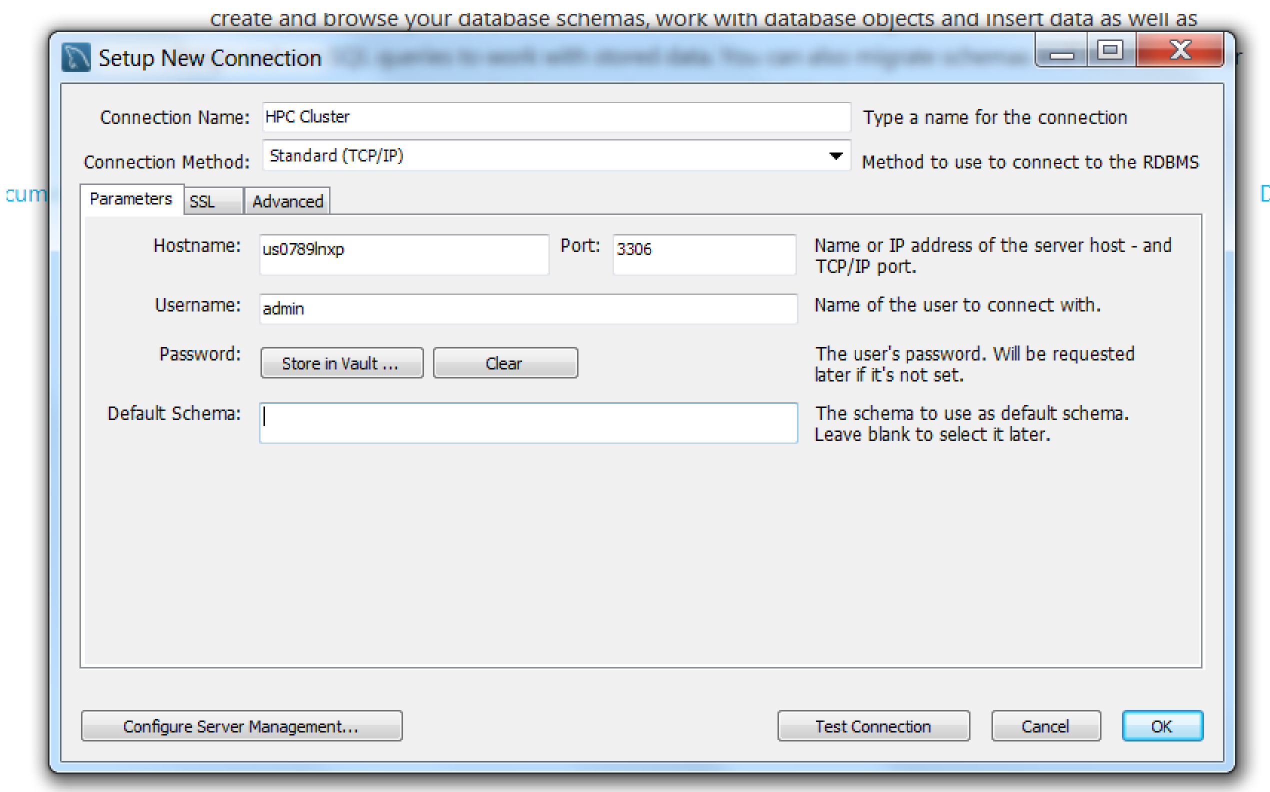


3.3 MySql Server Support Details

- **Role**
    - MySql is the current prod database
- **Basic Functionality**
    - The MySql Server runs on box 902 on port :3306
    - The port is externally accessible by tunneling through us0789lnxp (see above)
    - Since the Mysql Server is running on the same box as the Shiny apps, it currently configured to access 127.0.0.1 (localhost).
    - Next release should have access through a NodeJS API using the 3306 tunneled port

- **Access**
    - To access the MySql Server you’ll need to logon to 902 as root.
    - The following command will give you command prompt access to prod db
    - mysql -uroot -pCl1920_my -Dpsamd
    - Running select min and max “Time” from the tables can give you a quick snapshot of the data
    - “select min(Time) from baytown”
    - “select max(Time) from baytown”
    - You can also download the MySQl workbench

- https://dev.mysql.com/downloads/workbench/
- The username and password are”admin” and “DB01_mysql”



Setup New Connection

Connection Name: HPC Cluster — Type a name for the connection

Connection Method: Standard (TCP/IP) — Method to use to connect to the RDBMS

Hostname: us0789lnxp Port: 3306 — Name or IP address of the server host - and TCP/IP port.

Username: admin — Name of the user to connect with.

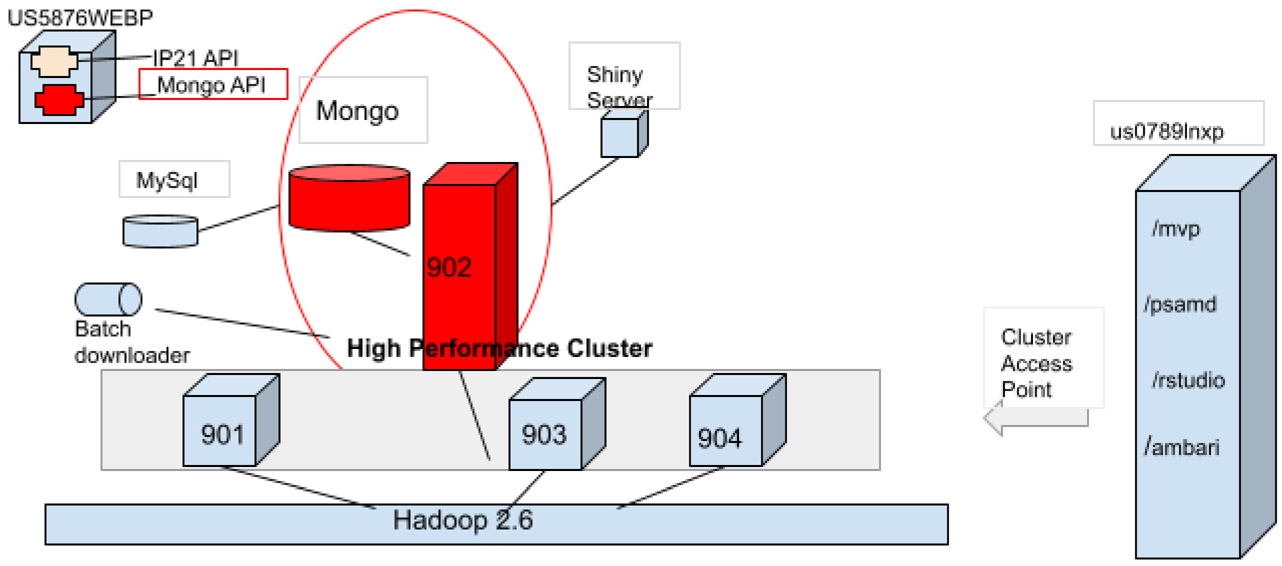
Password: Store in Vault ... Clear — The user's password. Will be requested later if it's not set.

Default Schema: — The schema to use as default schema. Leave blank to select it later.

- **Troubleshooting**
  - If you start receiving “no data returned” errors or the data starts looking corrupted
  - You can check the status of the MySql server and restart if necessary
  - The log files are located at /var/log/mysql and /var/log/mysqld
  - If the data is corrupted someway you can also replace the current database with a twice weekly backup in /tmp
  - Or just delete all data before a previous date
- **Status**
  - To check the status logon to 902 and run the “service” command
  - [root@compute902 ~]# service mysqld status
  - You should see something similar to : Active: active (running) since Tue 2018-02-13 09:42:52 EST; 1 months 0 days ago
- **Restart**
  - To restart just run the command as you see listed below
  - [root@compute902 ~]# service myslqd restart
- **Import mysql dump file**
  - mysql -uusername -ppassword db_name < file.sql
- **Delete new windows**
  - a quick fix is just to remove all the new windows made by users
  - You can just delete up to the first window
  - DELETE from baytown where Time < '2016-12-22 00:02:00';

- **If you import new data or delete old data you will need to update the mongo db to reflect the new data.**
    - See next section

3.4 Mongo Document DB

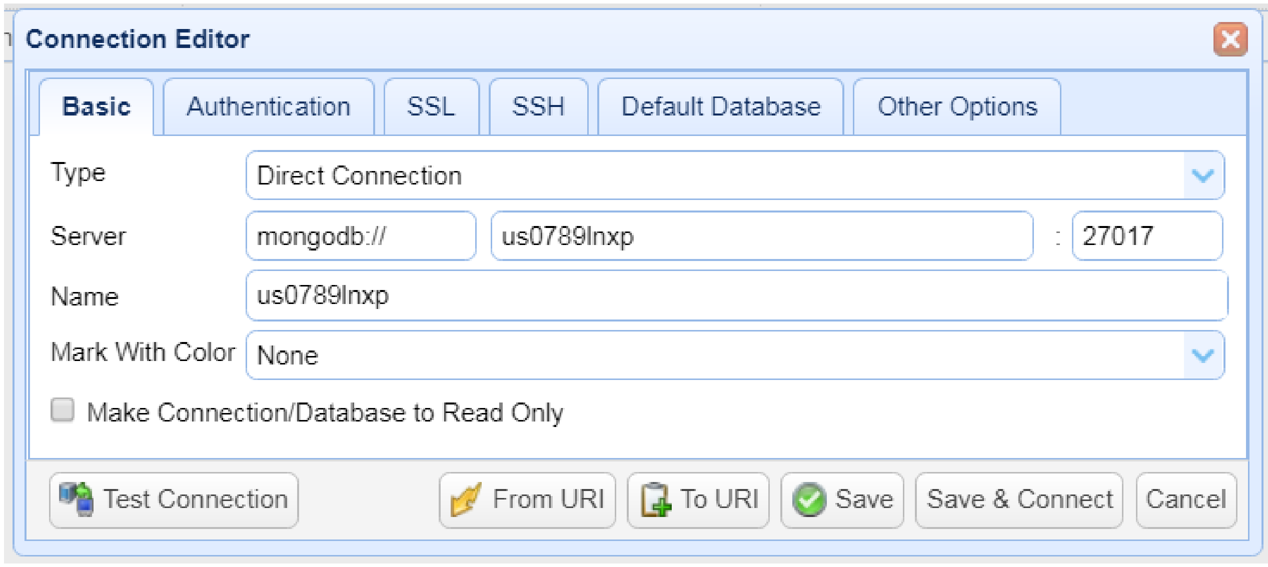


3.5 Mongo Support Details

- **Role**
    - Mongo tracks the current configuration of the prod database
    - It contains documents for each plant
    - It tracks modification to the data for each plant
- **Basic Functionality**
    - The mongodb is accessible through both and API and command line on 902
    - Any modifications to the data or plant information by the psamd applications also results in a call to the API to update the corresponding document in mongo.
    - Also a nightly batch downloader process (see next section) also updates the prod database and the corresponding documents in mongo
- **Access**
    - For support purposes the best way to access the mongodb is via command line.
    - Logon to 902 as Root.
    - Currently there is no password necessary for Root access.
    - Type “mongo” to access the mongodb shell commands.
    - Then type “use psamd” to switch to the prod database.
    - Typing “db.plants.find()” will pull every document in the db for the plants collection
- **Troubleshooting**
    - Any problems with the mongodb will cause the system to fail or malfunction
    - If you are seeing date errors or problem returning data or rendering the graph, check your mongo documents

- You will need to check the status of the mongdb, start or restart if necessary
- Check if the plant documents have been corrupted
- And that the API is up and running

- **Status**
  - To check the status of the mongo db run " service mongod status" on the command line
  - You should see something similar to : Active: active (running) since Fri 2017-10-13 18:01:14 EDT; 5 months 3 days ago
- **Restart**
  - To restart run "service mongod restart"
- **Corrupted Plants**
  - Logon to 902 as root and run the "mongo" command
  - Then run the "use psamd" to switch to prod db
  - The run "db.plants.find()" to view all the docs stored in mongo
  - The psamd runs off the "endDate" to keep track of the latest data pulled from IP21
  - The system assumes this is the latest data available, you cannot pull data after this date
  - The "sparse" section is a snapshot of the current data available in the system
  - Based off this section the system will decide whether it needs to make a call to IP21.
  - Corruption here may manifest in no data being returned because the mongo doc says we have the data, but then the actual pull to the prod db returns nothing, because the mongo doc is incorrect.
- **Restoring mongo data**
  - To restore the mongo data you need to update the json template located at: /cronjobs/plant.json
  - Set the "startDate" and "endDate" to valid dates in the db, generally one day including the max(Time) in the db.
  - Update the sparse section, generally by reducing the array down to one, the latest.
  - The use the following command to drop the current data import the json:
  - mongoimport --db psamd --collection plants --drop --file /cronjobs/plant.json --type json
- **Important Note:**
  - **Review the Mysql support section to ensure your mongo doc dates match the data in the prod database**
- **Checking the mongo API**
  - The mongo API is hosted on US5876WEBP
  - You'll need "firecall" authorization to access it: https://iapps.apci.com/firecall/
  - The mongo api should be running as service: Mongo Node 4040
  - You can check by going to start->"Component Services" -> "services Local"
  - The service is running the mong_api.js script located at c:\NodeJS
  - You could also stop the service, open a node JS prompt (type node into start search box).
  - At the prompt run "node mongo_api.js"
  - This will give you a real-time view of what mongo is receiving an returning.
  - You can modify mongo_api.js to "console.log("")" additional data.
  - Just remember to save the file, use control-c to stop the process in the cmd window and "node mongo_api.js" to restart the api
- **MongoDB Editor**
  - Download the editor noSQLBooster,
  - https://nosqlbooster.com/s3/download/releasesv5/nosqlbooster4mongo-5.0.3.exe
  -



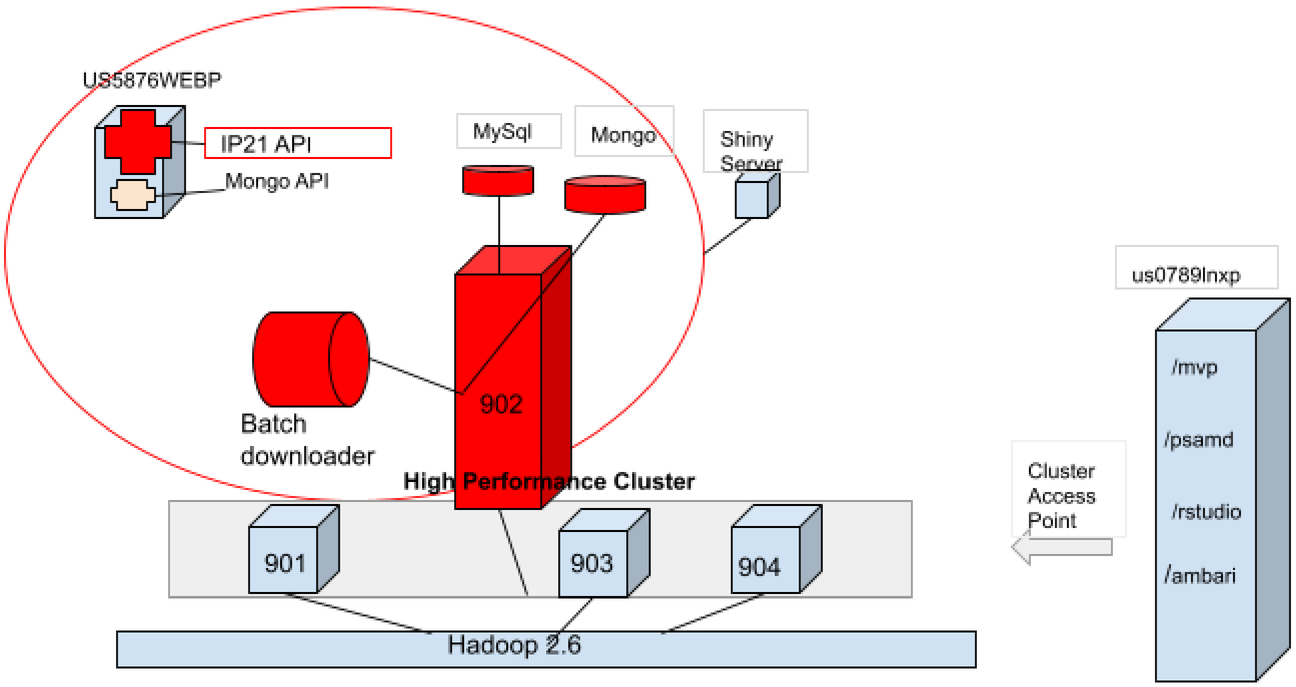
Connection Editor

Basic | Authentication | SSL | SSH | Default Database | Other Options

Type	Direct Connection
Server	mongodb:// us0789lnxp : 27017
Name	us0789lnxp
Mark With Color	None

☐ Make Connection/Database to Read Only

3.6 Batch Downloader

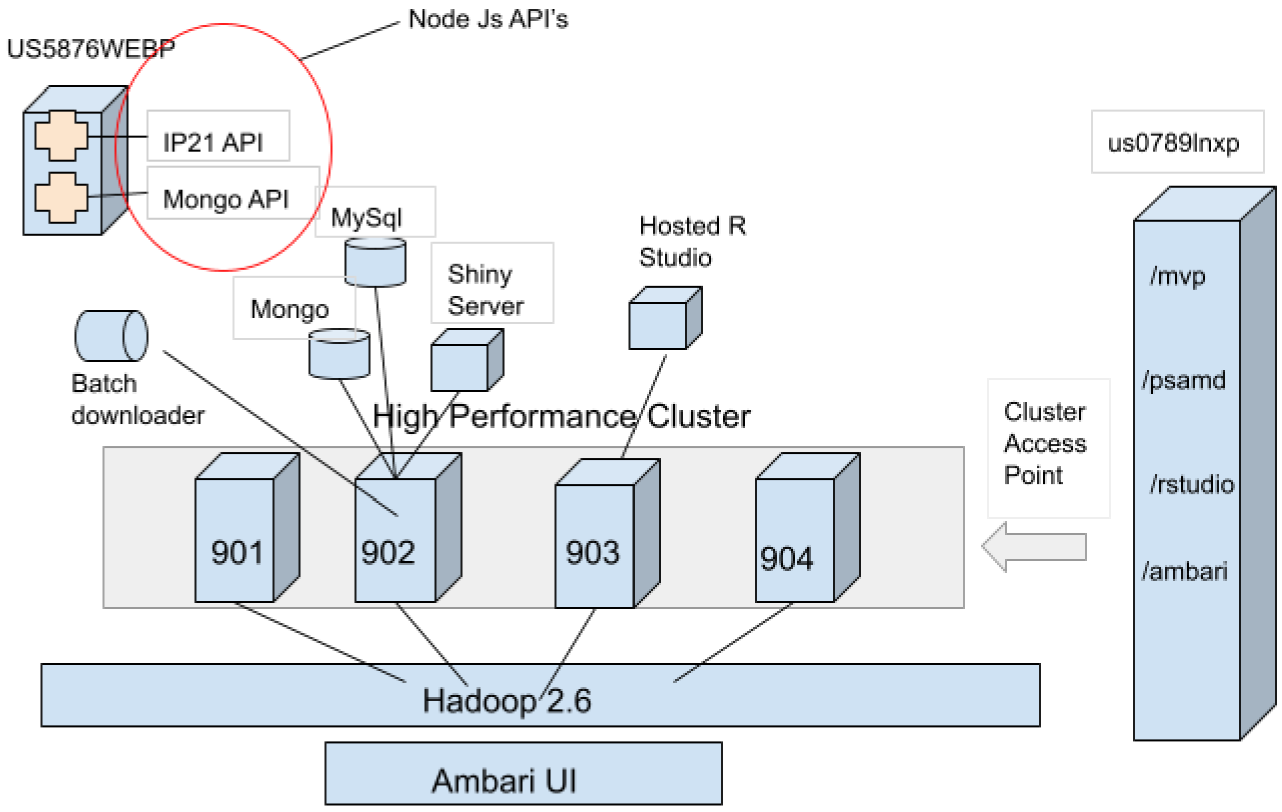


3.7 Batch Downloader Support Details

- **Role**
  - The batch downloader is runs nightly and updates prod db with the last days data from IP21
- **Basic Functionality**
  - The batch downloader is an R script the runs as a batch job scheduled as a linux cronjob.

- The file resides in /cronjobs
  - crontab -l will list the job
  - [root@compute902 ~]# crontab -l
  - 1 1 * * * /opt/continuum/anaconda/bin/R CMD BATCH /cronjobs/psamd_batch_prod.r
  - To edit the job use "crontab -e"
  - The batch downloader pulls tag info from mongo and pulls the relevant info from IP21 for each plant.
- **Access**
  - To access the job use the "crontab -e" to edit the job, or edit the R file in /cronjobs
- **Troubleshooting**
  - If the previous days data is not showing as the default option in the psamd date window
  - or your seeing wide ranging system failures or malfunction (see mongo trouble shooting).
  - You can check if the the cronjob itself had any problems
  - You may need to restore mongo and mysql based on the previous sections
  - And check that the batch download had access to both the mongo and IP21 AP

3.8 Node JS



- **Role**
  - Allows API access to different software platform
- **Basic Functionality**

    - download installer from https://nodejs.org/dist/v10.15.0/node-v10.15.0-x64.msi
    - After that you will have a nodeJS command prompt in your start window
    - This will give you a command called node <file.js>, that will run any JS server file.
    - To make a new server file you can copy what's in server.js on the windows jump box.
    - But basically you need to use a command called "npm" to add modules to the JS file.
    - The main one you need is express. npm install express
    - This gives you a nodeJS server.
    - After that you need to set up the server, it's only like 3 commands:

      ```
      var server = app.listen(4000, function () {

        var host = server.address().address
        var port = server.address().port
        console.log("listening at http://%s:%s", host, port)

      })
      ```

    - After this you set the paths that you want people to GET, POST, PUT, DELETE to.

      ```
      app.get('/getPlants', function (req, res, next) {
      //you can take anything off query line
                  req.query.body;
                  req.query.name;

                  //And you can send back whatever is needed
                                              res.send(send_data_str);
                                              res.end


      })
      ```

- **Access**
    - Access to the box is controlled through firewall you'll need to gain permission from Peter Verderame
- **Troubleshooting**
    - You start your server by typing "node <fileJS> into the node js prompt
    - You server should be running at this point, if not any js errors will be shown
    - Now if you close the node js prompt you server will stop
    - So you can add your server as a windows service

      ```
      var Service = require('node-windows').Service;

      // Create a new service object
      var svc = new Service({
        name:'OPENTSDB_API_Wrappers_4242',
        description: 'NodeJs Server on port 4242 to Access Opentsdb API through wrappers',
        script: 'C:\\NodeJs\\opentsdb.js'
      });

      // Listen for the "install" event, which indicates the
      // process is available as a service.
      svc.on('install',function(){
      ```

```
  svc.start();
});

svc.install();
```

- Now if you set your windows service to automatic it will restart every time windows restarts.

4. Database Schema

4.1 Tables, Fields and Relationships

Provide a description of any new tables, fields and relationships that need to be created for the design.

4.1.1 Databases

BPData production, BPDataPCMSTest for development and testing..

4.1.2 New Tables

List any new tables that will be needed, for each one including table name, table description, and related tables.

4.1.3 New Fields(s)

List any new tables that will be needed, for each one including table name, table description, and related tables.

Table Name	Field Name	Data Type	Allow Nulls	Field Description

4.1.4 Fields Change(s)

For each field change (such as data types, required/not required, or renaming), please complete a row of the following table. (Insert additional rows as needed.)

Table Name	Field Name	What to change?

4.1.5 All Other Changes

If any other changes are requested (stored procedures, indexes, relationships, security settings, DTS packages, maintenance plans, etc), please describe what is needed here.

4.2 Data Migration

(Optional) - Provide a description of how existing data should be migrated to new tables and fields.

Appendix A: Project Timeline