DIMS Training Manual Documentation

Release 0.3.2

Dave Dittrich

Nov 29, 2017

Contents

This is the DIMS Training Manual (version 0.3.2).

CHAPTER 1

Introduction

1.1 Introduction

This chapter introduces ...

CHAPTER 2

Referenced documents

1. Contract HSHQDC-13-C-B0013, “From Local to Gobal Awareness: A Distributed Incident Management System,” Section C - Statement of Work
2. DIMS User Manual v 0.2.1
3. DIMS Job Descriptions v 2.9.0

CHAPTER 3

Using the Dashboard

This section will introduce basic usage of the DIMS Dashboard.

Currently this section contains a demo of the dashboard with commentary. The following sub-sections will go through:

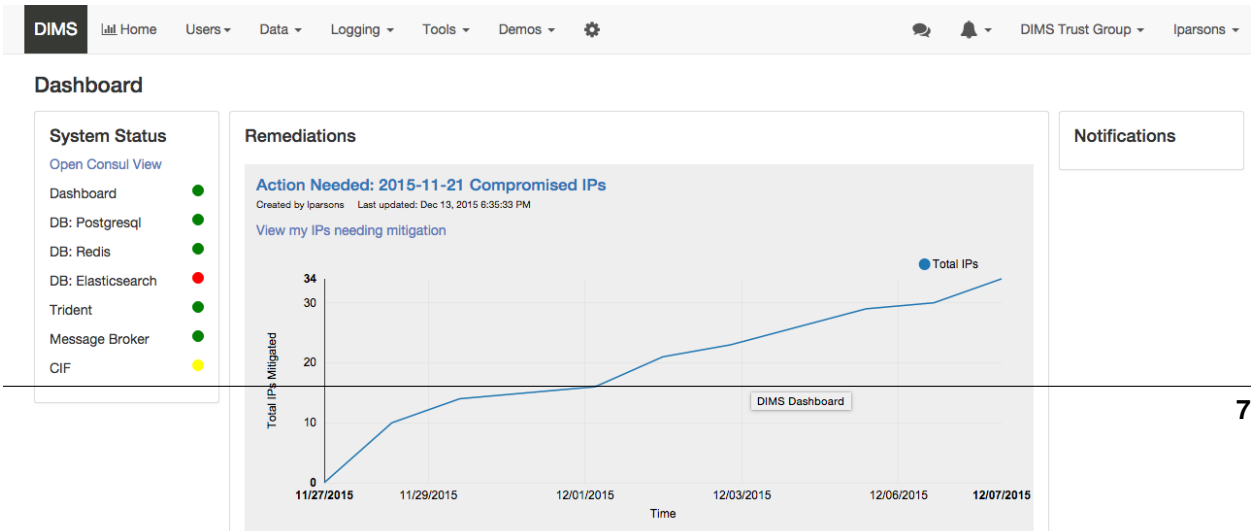
1. Mitigation Scenario
2. System health
3. Live log streaming
4. Chat
5. User display and trust group info - show users by trust group

The demo application is not currently using https, so you won't need to worry about certificates when logging in. Make sure you are logged out of the dashboard (if you are already logged in) and reload the login page if you are already on that page (to make sure your client has the latest version).

3.1 Mitigation Scenario

Go to `demo.prisem.washington.edu` and log in using your `ops-trust` username and password.

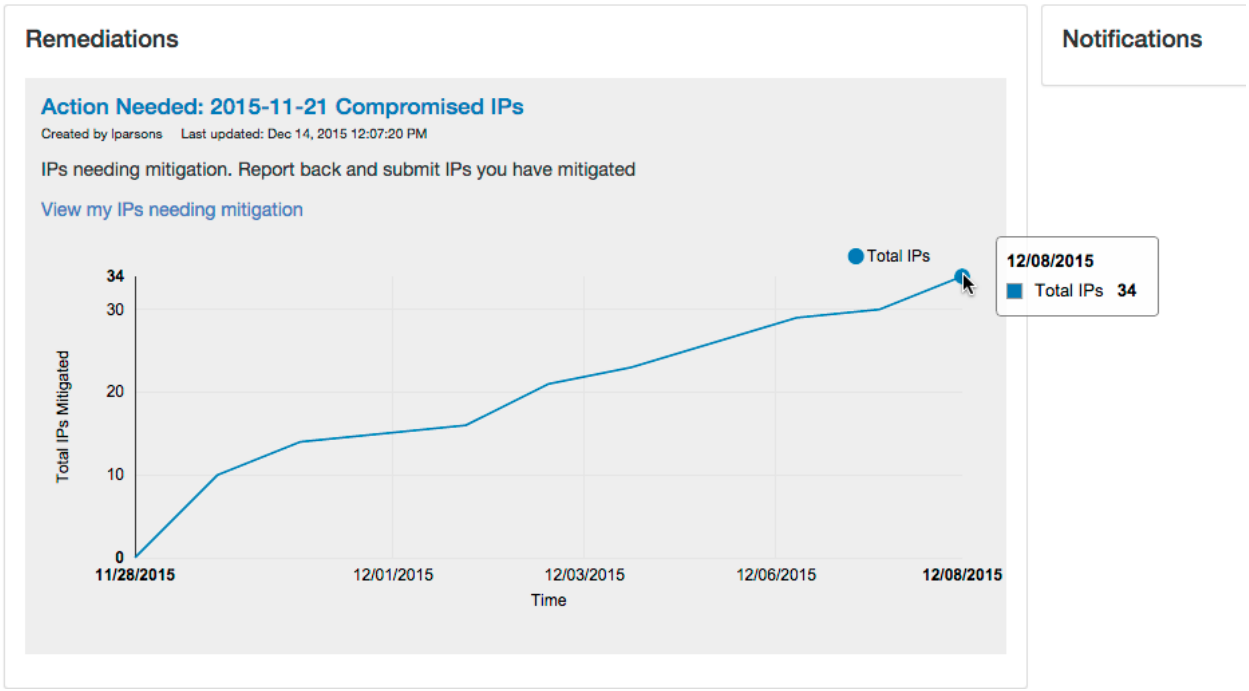
The main dashboard will display.



Remediations are mitigation activities

in the system where the logged in user (lparsons in the screenshots) has IPs that are compromised and need to be remediated. Currently, the system contains one of these "Mitigation scenario" activities which was bootstrapped programmatically.

Hover over the graph to display data points. Note the number of mitigated IPs at the most recent data point (the dates may differ than that in the screenshot):



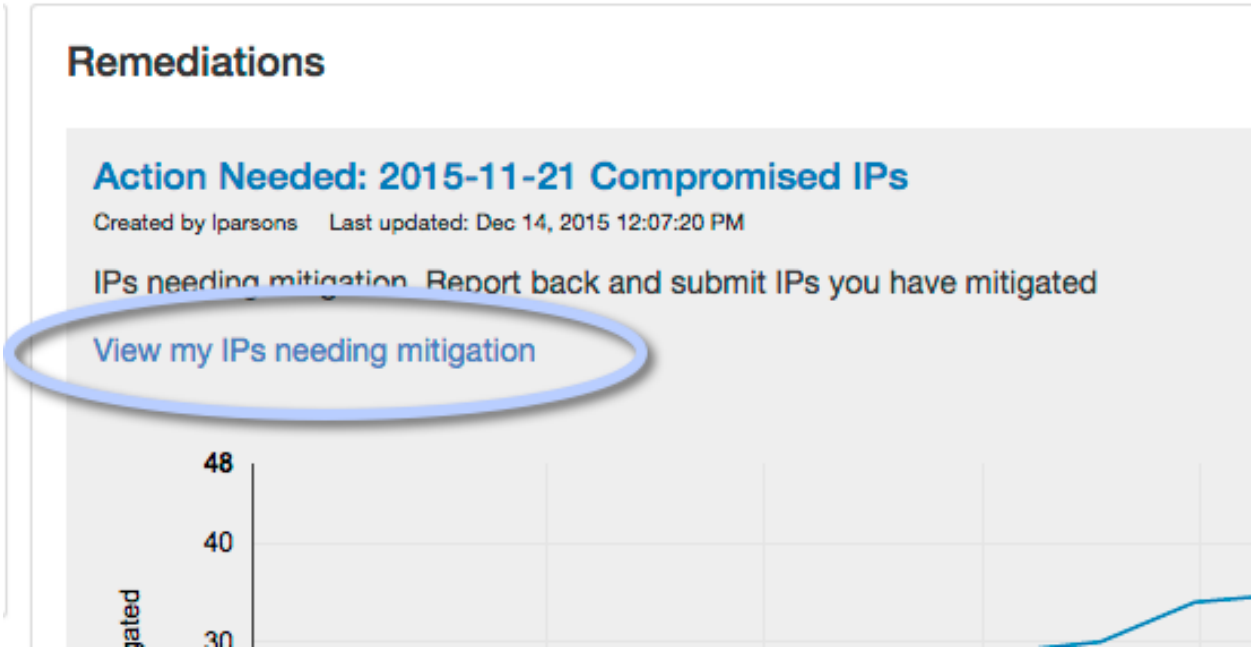
Click **View my IPs needing mitigation** link to display a modal window where you can submit IPs that have been mitigated. Right now, the UI for this consists of the modal displaying all remaining IPs you need to address.

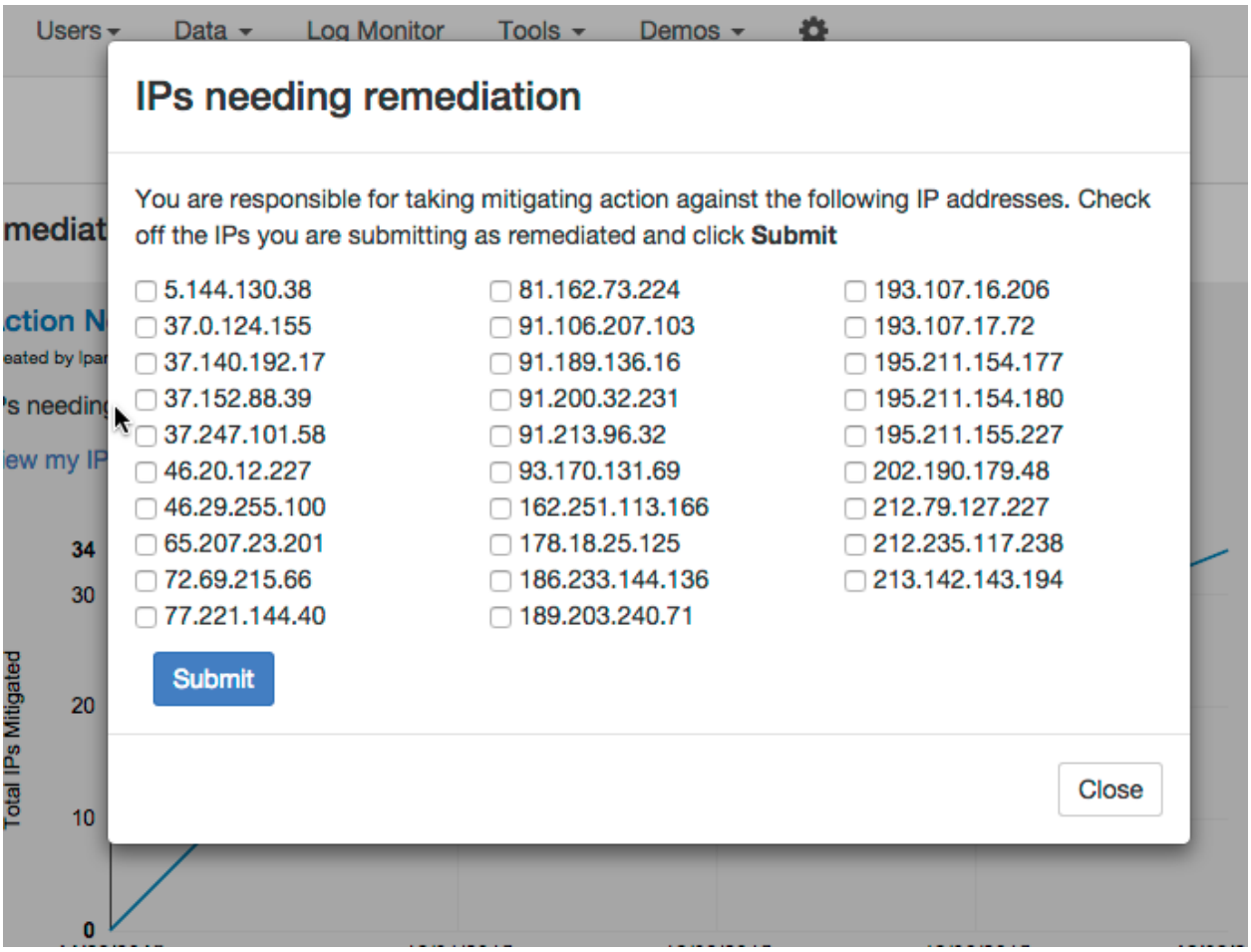
This mitigation activity has IPs that need to be remediated for the users dittrich, lparsons, mboggess, and swarner. So your IPs will look different than those in this figure.

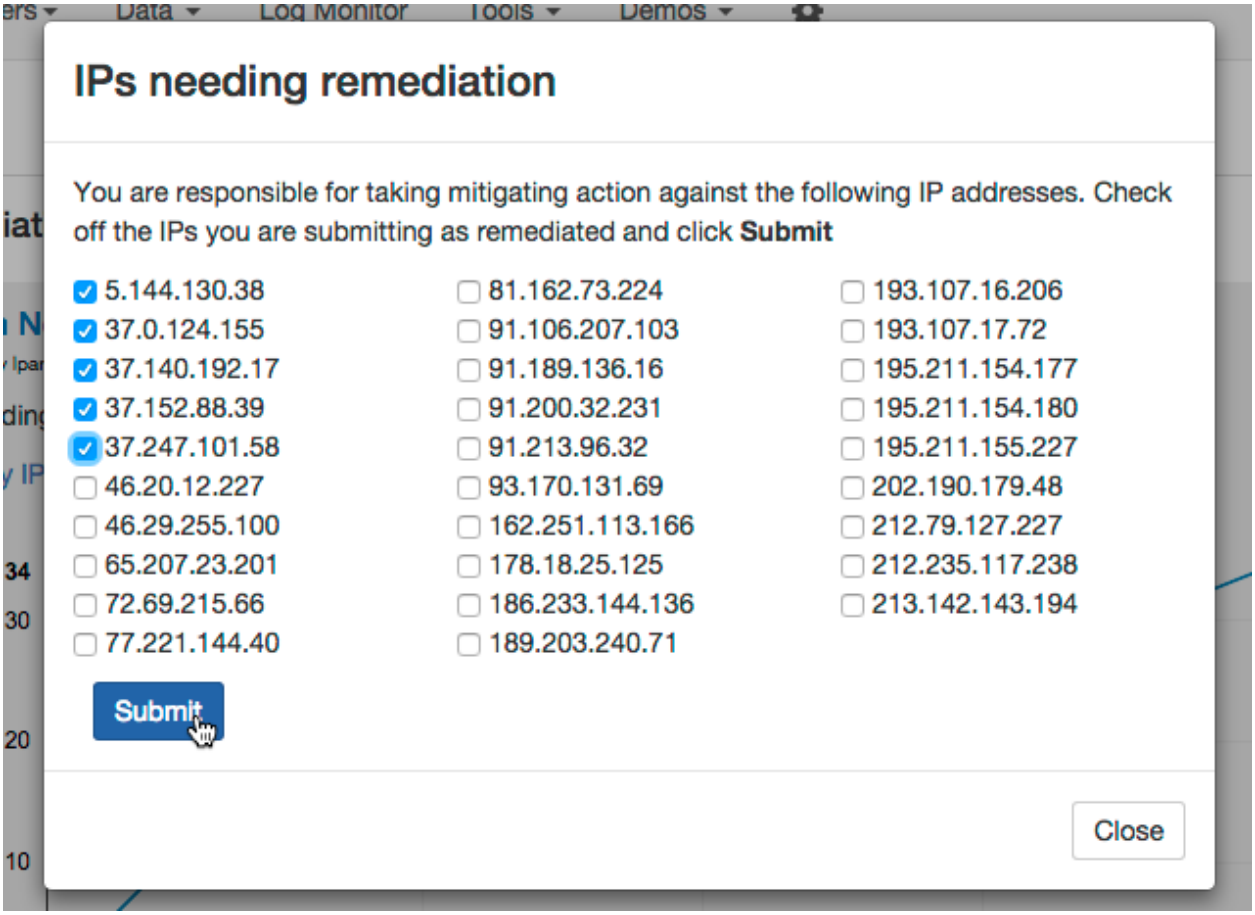
Check off some IPs indicating that they have been mitigated and click *Submit*.

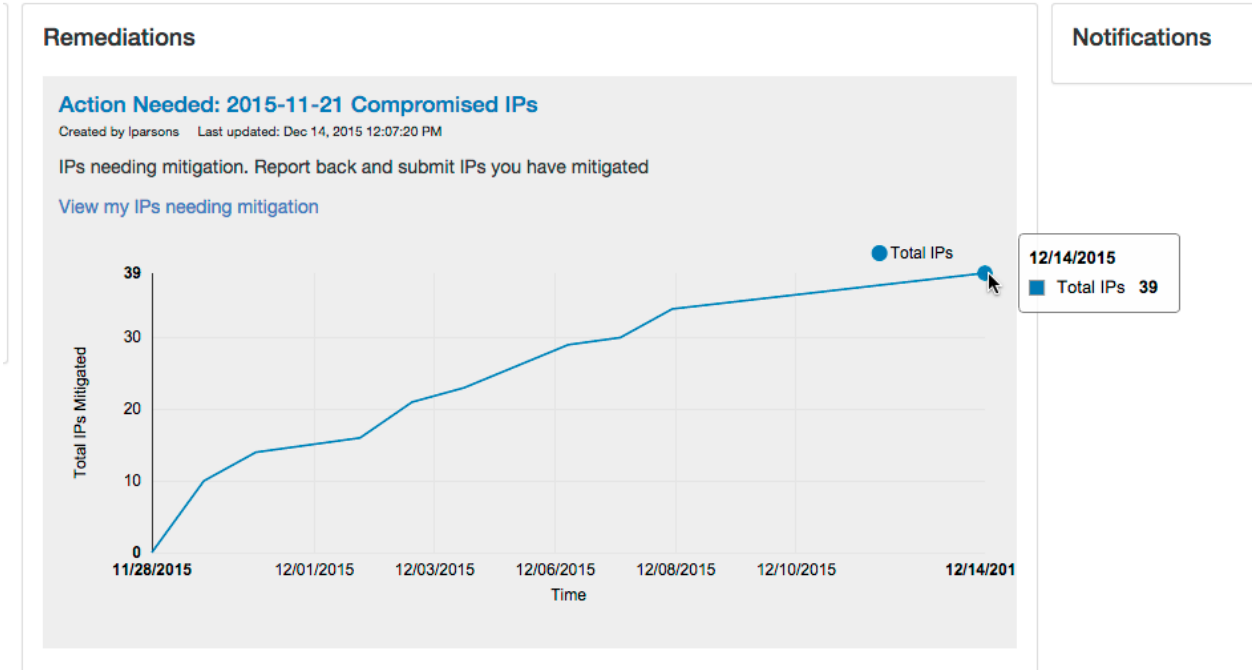
The modal window will close and the graph will be updated. Hover over the last data point to verify. For this user, the total IPs mitigated is now 39.

Note: Currently, to start a new mitigation activity, a user will do so via the Dashboard (UI not available yet), using a


Remediations
Action Needed: 2015-11-21 Compromised IPs
Created by lparsons Last updated: Dec 14, 2015 12:07:20 PM
IPs needing mitigation. Report back and submit IPs you have mitigated
View my IPs needing mitigation
48
40
30

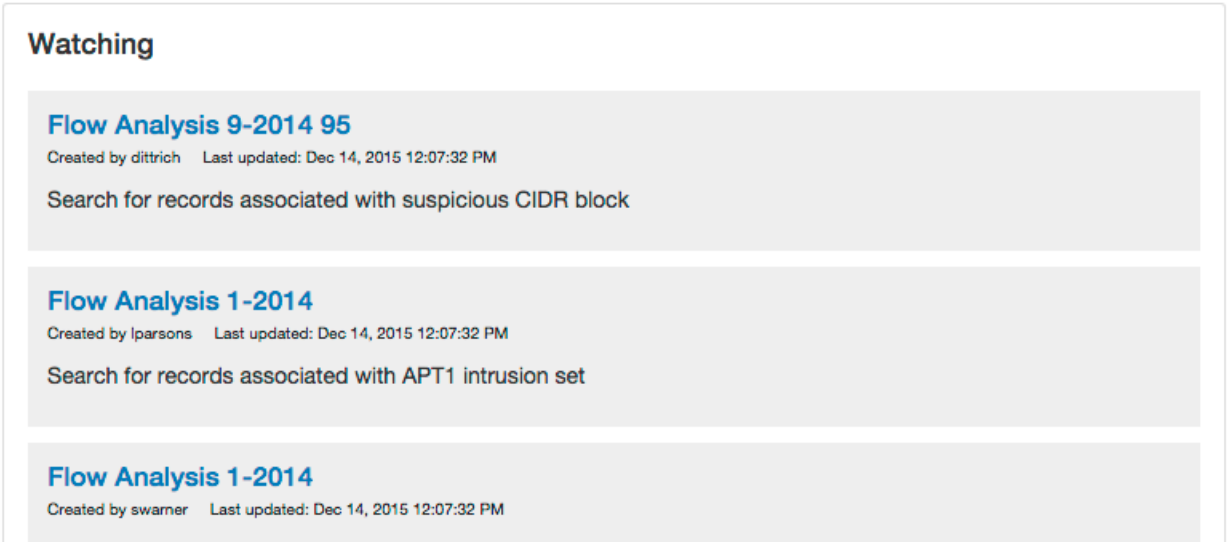

Users
Data
Log Monitor
Tools
Demos
IPs needing remediation
You are responsible for taking mitigating action against the following IP addresses. Check off the IPs you are submitting as remediated and click Submit
5.144.130.38
37.0.124.155
37.140.192.17
37.152.88.39
37.247.101.58
46.20.12.227
46.29.255.100
65.207.23.201
72.69.215.66
77.221.144.40
81.162.73.224
91.106.207.103
91.189.136.16
91.200.32.231
91.213.96.32
93.170.131.69
162.251.113.166
178.18.25.125
186.233.144.136
189.203.240.71
193.107.16.206
193.107.17.72
195.211.154.177
195.211.154.180
195.211.155.227
202.190.179.48
212.79.127.227
212.235.117.238
213.142.143.194
Submit
Close
Total IPs Mitigated
34
30
20
10
0


Data
Log Monitor
Tools
Demos
IPs needing remediation
You are responsible for taking mitigating action against the following IP addresses. Check off the IPs you are submitting as remediated and click Submit
5.144.130.38
37.0.124.155
37.140.192.17
37.152.88.39
37.247.101.58
46.20.12.227
46.29.255.100
65.207.23.201
72.69.215.66
77.221.144.40
81.162.73.224
91.106.207.103
91.189.136.16
91.200.32.231
91.213.96.32
93.170.131.69
162.251.113.166
178.18.25.125
186.233.144.136
189.203.240.71
193.107.16.206
193.107.17.72
195.211.154.177
195.211.154.180
195.211.155.227
202.190.179.48
212.79.127.227
212.235.117.238
213.142.143.194
Submit
Close


Remediations
Notifications
Action Needed: 2015-11-21 Compromised IPs
Created by lparsons Last updated: Dec 14, 2015 12:07:20 PM
IPs needing mitigation. Report back and submit IPs you have mitigated
View my IPs needing mitigation
Total IPs
12/14/2015
Total IPs 39
Total IPs Mitigated
39
30
20
10
0
11/28/2015
12/01/2015
12/03/2015
12/06/2015
12/08/2015
12/10/2015
12/14/201
Time

form to submit the suspect IPs that the user probably received on a Trident email list. The system then automatically parses the list and bins the IPs according to attributes belonging to users, creating a new activity that will appear in the Remediations list for those users that are affected. There will also be some sort of notification. (In the future this creation would be automated by a service that can process emails that come into the system.)

The **Watching** section lists Activities that the user has subscribed to, either by subscribing to a public activity created by someone else or by creating a new activity.



Activities are collections of data, queries, etc. They can be public or private. If a user subscribes to a public activity, the user receives a notification when new data is added to the activity. This is a first cut at the UI, and most of the UI display/functions (creating, sharing, subscribing) are currently in progress and not online (server side API and associated modules exist). The only thing you can see right now in the UI is the list of activities.

3.2 System health

The status area on the left is mostly static at present. However, a link to open the consul UI in a new tab exists.

Click **Open Consul view**:

and the Consul UI will open in a new tab with the **NODES** tab selected.

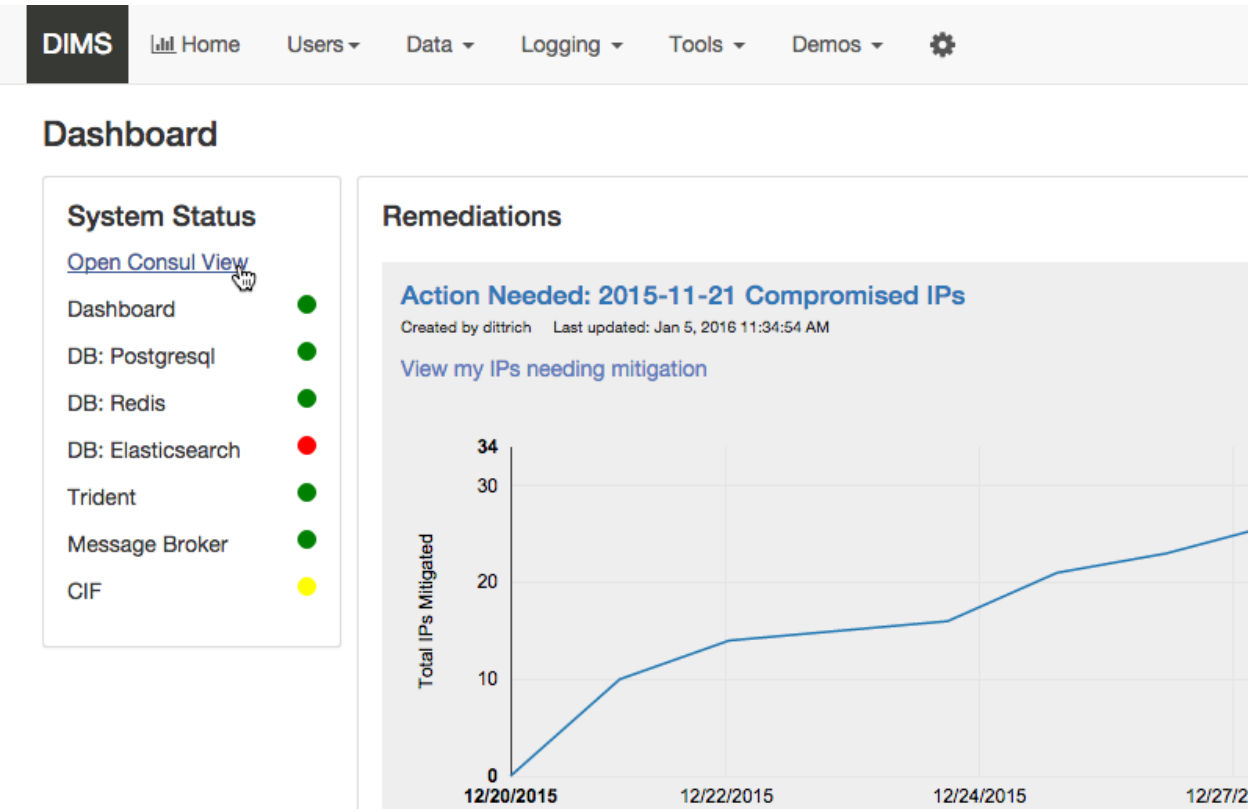
3.3 Live log streaming

The live log monitoring is now a popup panel so as to persist data across page views. That means the buffers won't be cleared if you go to a different section on the site (e.g. new page load).

1. Click **Logging** in the Navigation bar and select **Live log streaming**.

   The Live log streaming window anchored to the bottom of the browser window will display.

   There are tabs for the log exchanges the server monitors. Each tab has a button to turn on and turn off that particular log monitor. The user can clear the buffer using the *Clear* button. The user can hide the window by clicking the minimize button (down arrow in title bar), and then maximize it by clicking the maximize button (up arrow). Clicking the close button (X) turns off all monitors and closes the window. The window can also



be closed by clicking **Logging > Live log streaming** in the Nav bar. (This is a toggle - if the window is active, clicking it closes the window. If the window is closed, clicking the button opens the window.)

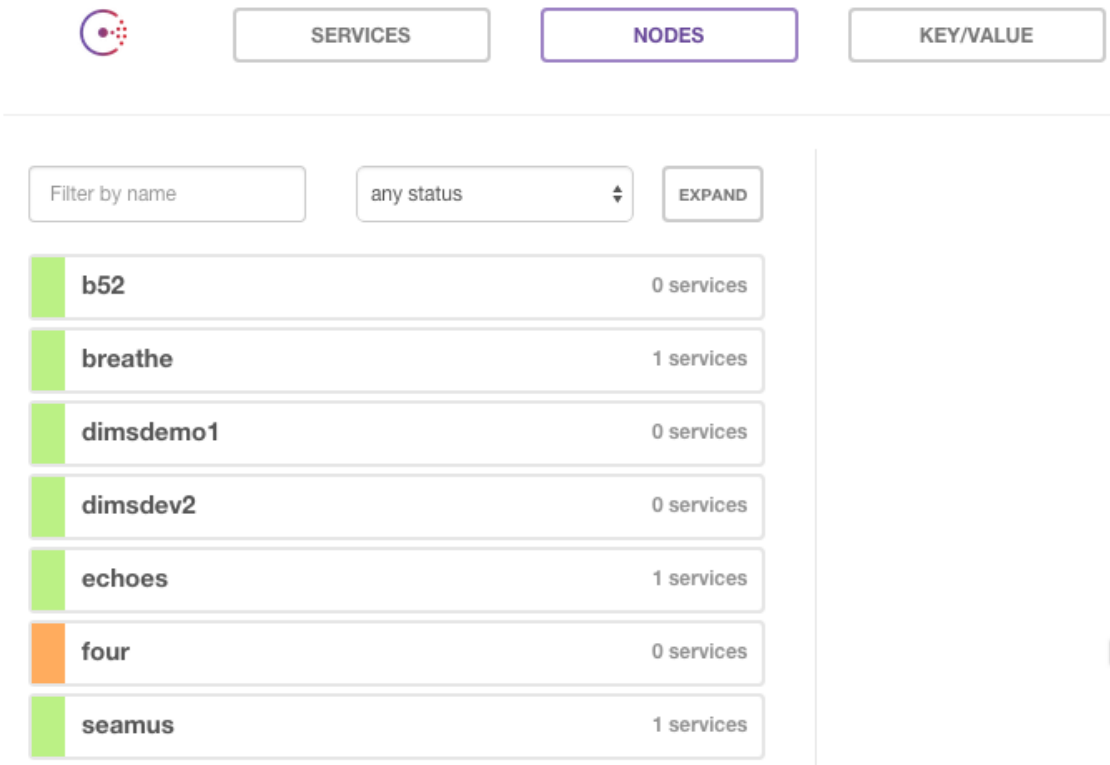
The *Live log streaming* window, like the *Chat* window, is independent of other page views. So it will remain active even if you go to a different view via a menu or navigation button.

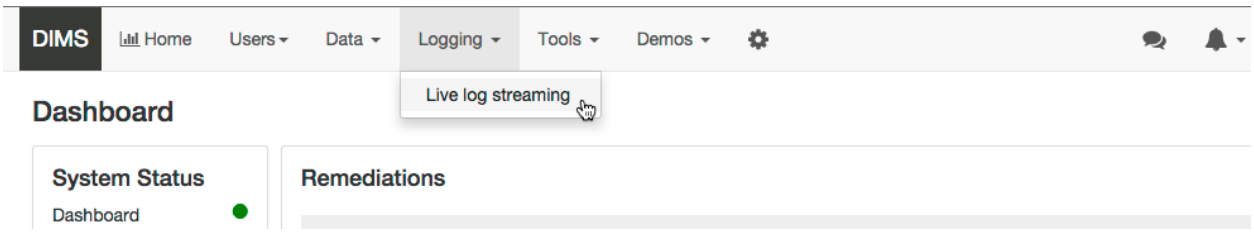
2. Click on **Devops** tab and click button **Turn on Devops**
3. Do the same for Health - click on Health tab and click button **Turn on Health**
4. You could start an activity that reports to devops via another program, or wait a couple minutes and you'll probably get info on Health:
5. Click the minimize button:

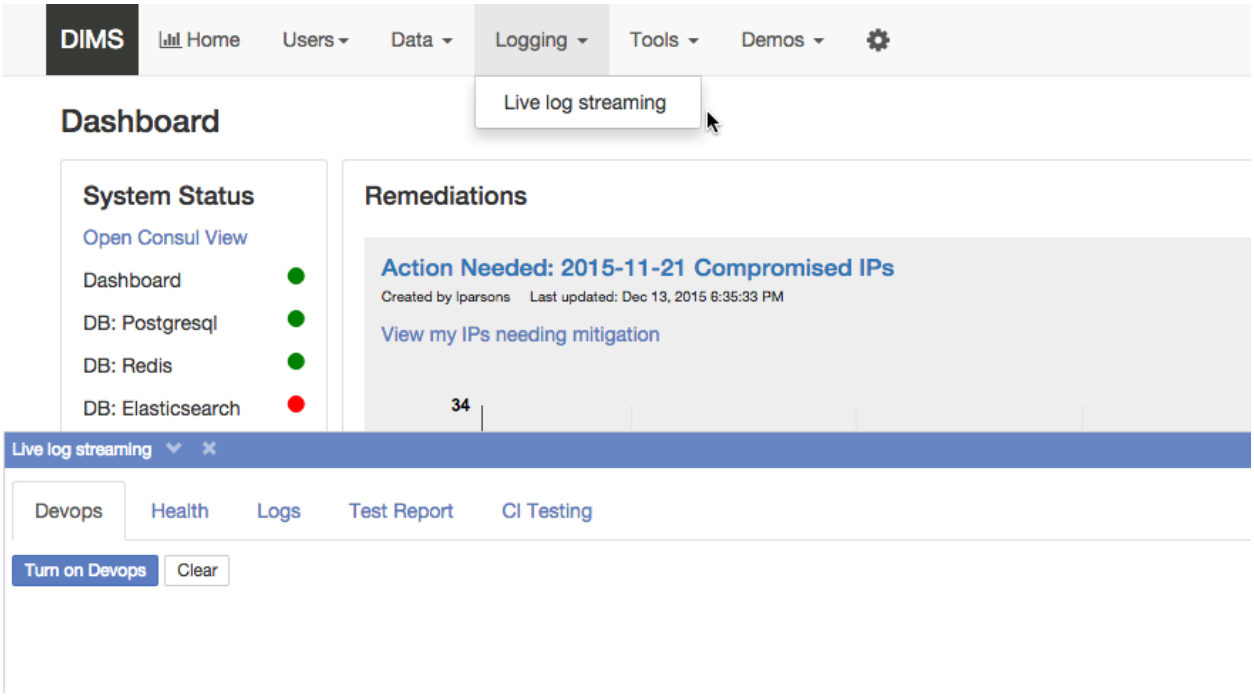
   and the logs will minimize to the bottom of the window.
6. Then click maximize to open it again.

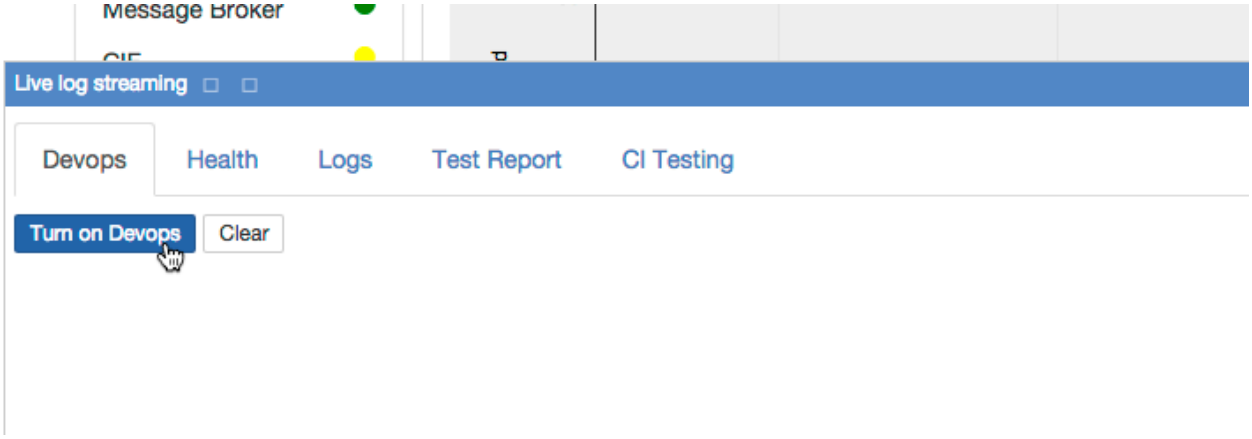
   The messages will still be there (maybe more).

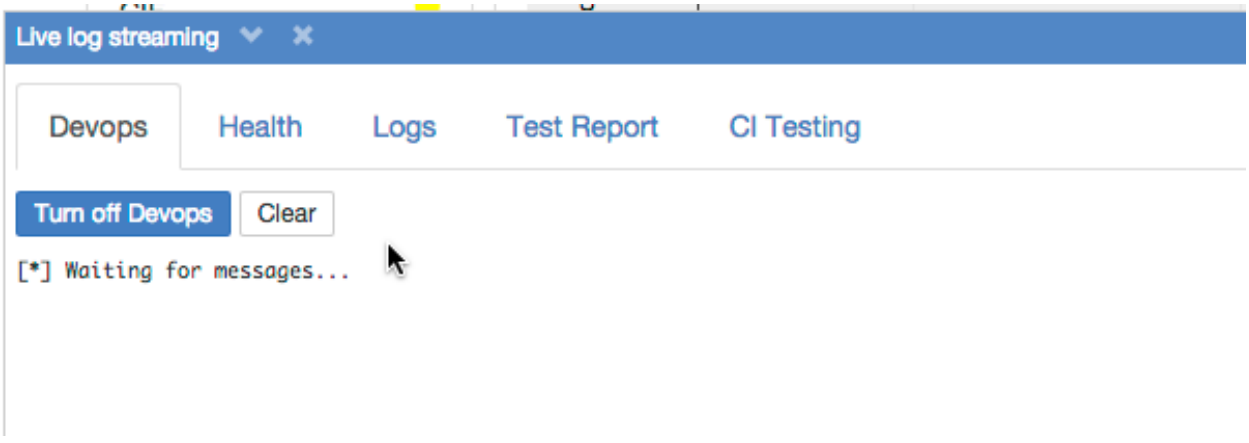
   You can go to different locations in the app without clearing the log buffers. So go to **Users > Find DIMS users** to display users in your current trust group. The users will display behind the streaming window. Minimize the streaming log display to view the users:
7. You can clear the log buffers individually by clicking **Clear** in a log tab. To clear all the buffers and close the display, click the **Log Monitor** link in the nav bar or just click the **X** in the monitor window title bar.


SERVICES
NODES
KEY/VALUE
Filter by name
any status
EXPAND
b52
0 services
breathe
1 services
dimsdemo1
0 services
dimsdev2
0 services
echoes
1 services
four
0 services
seamus
1 services

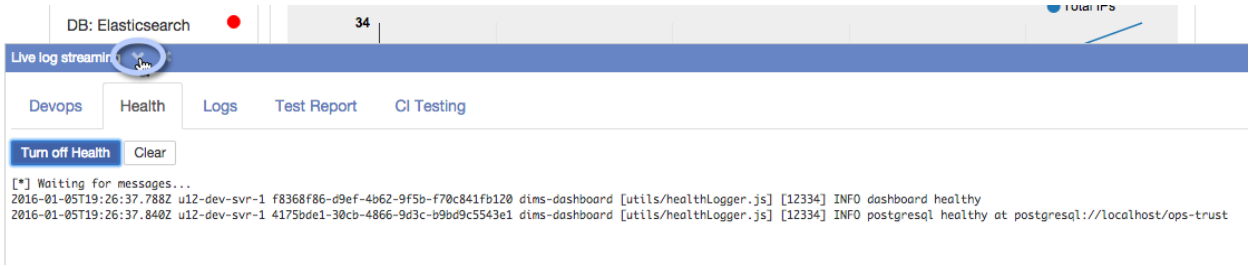

DIMS
Home
Users
Data
Logging
Tools
Demos
Live log streaming
Dashboard
System Status
Dashboard
Remediations


DIMS
Home
Users
Data
Logging
Tools
Demos
Live log streaming
Dashboard
System Status
Open Consul View
Dashboard
DB: Postgresql
DB: Redis
DB: Elasticsearch
Remediations
Action Needed: 2015-11-21 Compromised IPs
Created by lparsons Last updated: Dec 13, 2015 6:35:33 PM
View my IPs needing mitigation
34
Live log streaming
Devops
Health
Logs
Test Report
CI Testing
Turn on Devops
Clear

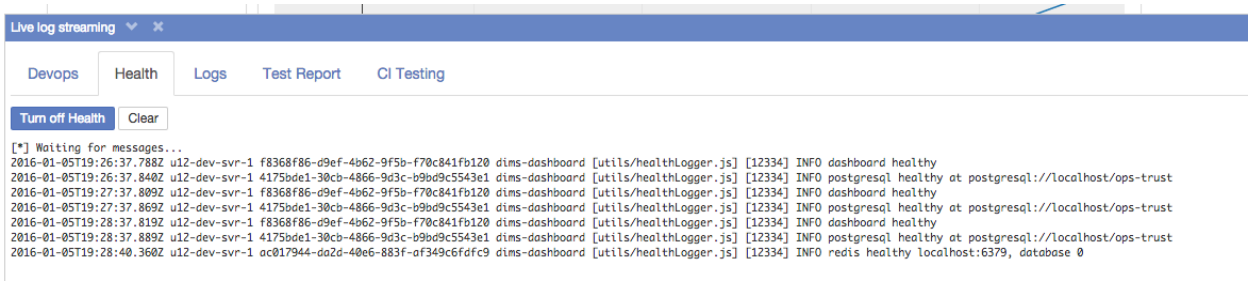

Message Broker
Live log streaming
Devops
Health
Logs
Test Report
CI Testing
Turn on Devops
Clear


Live log streaming
Devops
Health
Logs
Test Report
CI Testing
Turn off Devops
Clear
[*] Waiting for messages...


DB: Elasticsearch
Live log streaming
Devops
Health
Logs
Test Report
CI Testing
Turn off Health
Clear
[*] Waiting for messages...
2016-01-05T19:26:37.788Z u12-dev-svr-1 f8368f86-d9ef-4b62-9f5b-f70c841fb120 dims-dashboard [utils/healthLogger.js] [12334] INFO dashboard healthy
2016-01-05T19:26:37.840Z u12-dev-svr-1 4175bde1-30cb-4866-9d3c-b9bd9c5543e1 dims-dashboard [utils/healthLogger.js] [12334] INFO postgresql healthy at postgresql://localhost/ops-trust


DB: Elasticsearch
Live log streaming
Devops
Health
Logs
Test Report
CI Testing
Turn off Health
Clear
[*] Waiting for messages...
2016-01-05T19:26:37.788Z u12-dev-svr-1 f8368f86-d9ef-4b62-9f5b-f70c841fb120 dims-dashboard [utils/healthLogger.js] [12334] INFO dashboard healthy
2016-01-05T19:26:37.840Z u12-dev-svr-1 4175bde1-30cb-4866-9d3c-b9bd9c5543e1 dims-dashboard [utils/healthLogger.js] [12334] INFO postgresql healthy at postgresql://localhost/ops-trust

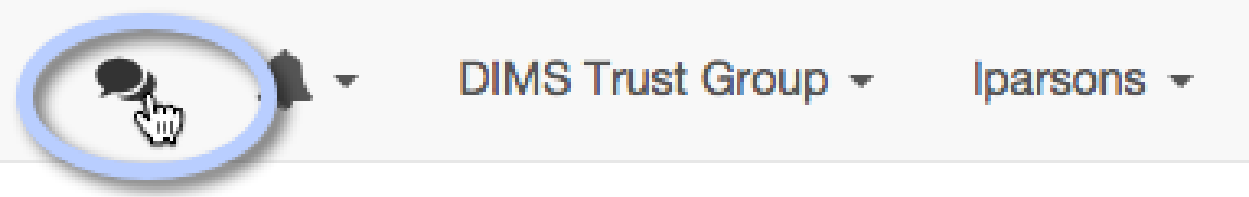

Live log streaming


Live log streaming
Devops
Health
Logs
Test Report
CI Testing
Turn off Health
Clear
[*] Waiting for messages...
2016-01-05T19:26:37.788Z u12-dev-svr-1 f8368f86-d9ef-4b62-9f5b-f70c841fb120 dims-dashboard [utils/healthLogger.js] [12334] INFO dashboard healthy
2016-01-05T19:26:37.840Z u12-dev-svr-1 4175bde1-30cb-4866-9d3c-b9bd9c5543e1 dims-dashboard [utils/healthLogger.js] [12334] INFO postgresql healthy at postgresql://localhost/ops-trust
2016-01-05T19:27:37.809Z u12-dev-svr-1 f8368f86-d9ef-4b62-9f5b-f70c841fb120 dims-dashboard [utils/healthLogger.js] [12334] INFO dashboard healthy
2016-01-05T19:27:37.869Z u12-dev-svr-1 4175bde1-30cb-4866-9d3c-b9bd9c5543e1 dims-dashboard [utils/healthLogger.js] [12334] INFO postgresql healthy at postgresql://localhost/ops-trust
2016-01-05T19:28:37.819Z u12-dev-svr-1 f8368f86-d9ef-4b62-9f5b-f70c841fb120 dims-dashboard [utils/healthLogger.js] [12334] INFO dashboard healthy
2016-01-05T19:28:37.889Z u12-dev-svr-1 4175bde1-30cb-4866-9d3c-b9bd9c5543e1 dims-dashboard [utils/healthLogger.js] [12334] INFO postgresql healthy at postgresql://localhost/ops-trust
2016-01-05T19:28:40.360Z u12-dev-svr-1 ac017944-da2d-40e6-883f-af349c6fdfc9 dims-dashboard [utils/healthLogger.js] [12334] INFO redis healthy localhost:6379, database 0

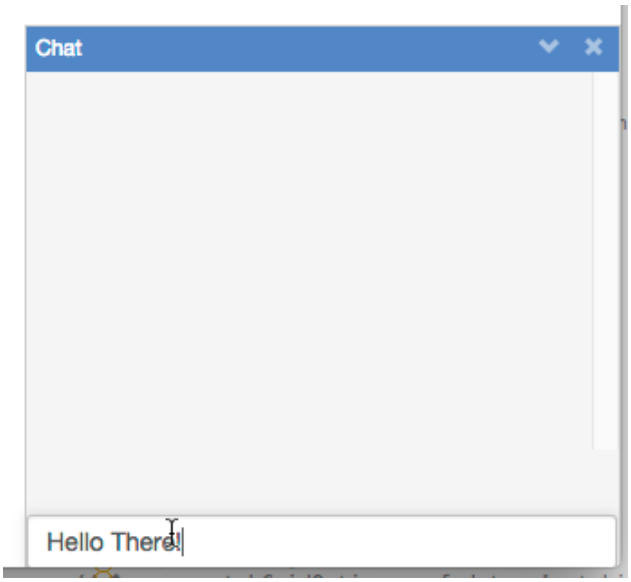

Live log streaming
Devops
Health
Logs
Test Report
CI Testing
Turn off Health
Clear
[*] Waiting for messages...
2016-01-05T19:26:37.788Z u12-dev-svr-1 f8368f86-d9ef-4b62-9f5b-f70c841fb120 dims-dashboard [utils/healthLogger.js] [12334] INFO dashboard
2016-01-05T19:26:37.840Z u12-dev-svr-1 4175bde1-30cb-4866-9d3c-b9bd9c5543e1 dims-dashboard [utils/healthLogger.js] [12334] INFO postgres
2016-01-05T19:27:37.809Z u12-dev-svr-1 f8368f86-d9ef-4b62-9f5b-f70c841fb120 dims-dashboard [utils/healthLogger.js] [12334] INFO dashboard
2016-01-05T19:27:37.869Z u12-dev-svr-1 4175bde1-30cb-4866-9d3c-b9bd9c5543e1 dims-dashboard [utils/healthLogger.js] [12334] INFO postgres
2016-01-05T19:28:37.819Z u12-dev-svr-1 f8368f86-d9ef-4b62-9f5b-f70c841fb120 dims-dashboard [utils/healthLogger.js] [12334] INFO dashboard
2016-01-05T19:28:37.889Z u12-dev-svr-1 4175bde1-30cb-4866-9d3c-b9bd9c5543e1 dims-dashboard [utils/healthLogger.js] [12334] INFO postgres
2016-01-05T19:28:40.360Z u12-dev-svr-1 ac017944-da2d-40e6-883f-af349c6fdfc9 dims-dashboard [utils/healthLogger.js] [12334] INFO redis he
2016-01-05T19:29:37.821Z u12-dev-svr-1 f8368f86-d9ef-4b62-9f5b-f70c841fb120 dims-dashboard [utils/healthLogger.js] [12334] INFO dashboard
2016-01-05T19:29:37.902Z u12-dev-svr-1 4175bde1-30cb-4866-9d3c-b9bd9c5543e1 dims-dashboard [utils/healthLogger.js] [12334] INFO postgres
2016-01-05T19:30:37.826Z u12-dev-svr-1 f8368f86-d9ef-4b62-9f5b-f70c841fb120 dims-dashboard [utils/healthLogger.js] [12334] INFO dashboard
2016-01-05T19:30:37.878Z u12-dev-svr-1 4175bde1-30cb-4866-9d3c-b9bd9c5543e1 dims-dashboard [utils/healthLogger.js] [12334] INFO postgres

3.4 Chat

1. Click the chat icon in the Nav bar to open the chat window:



Unless you're chatting with someone else who is logged in, there isn't much to see (you can send messages to yourself however). Enter a message in the message area of the chat box and press **Enter** key.



The message you sent will appear in your chat window:

3.5 User display and trust groups

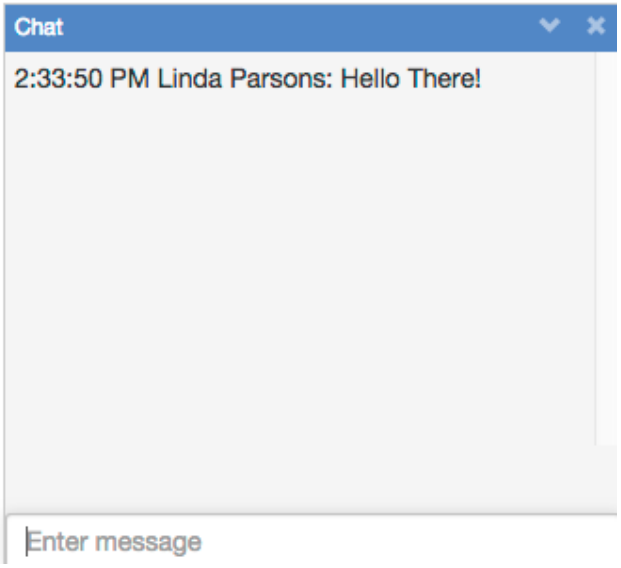
1. Note that the name of the trust group you are logged into displays on the menu bar:

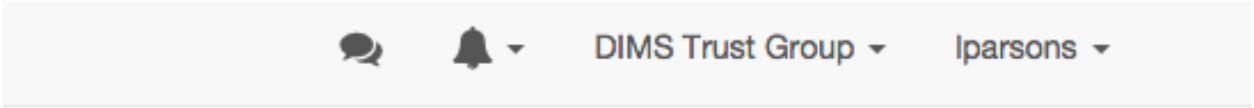
   The system remembers your last selection. If you have never selected a trust group, it will choose the first one in your list of trust groups when you first log in.

2. Display your profile information by selecting `lparsons > Profile` in the nav bar. Note that the trust group info now displays in the profile.

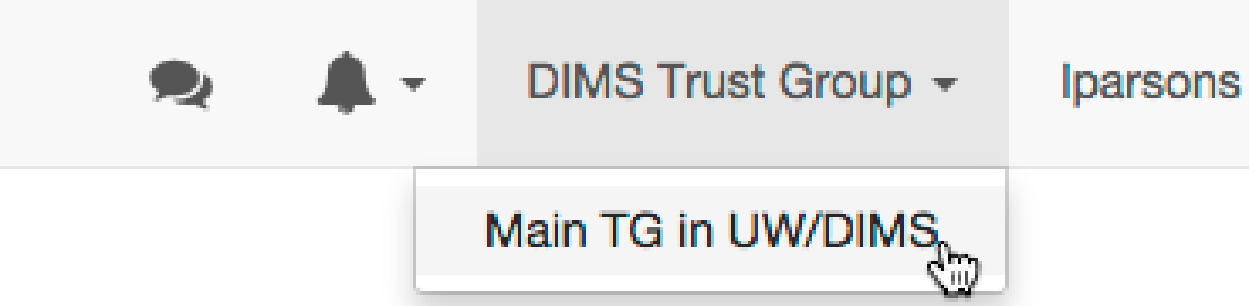
3. Change your trust group by clicking on the current trust group in the nav bar and selecting an option that displays in the menu. (If you are only in one trust group then no options will display.)

4. Note that the trust group listed in your profile will change to the current trust group:



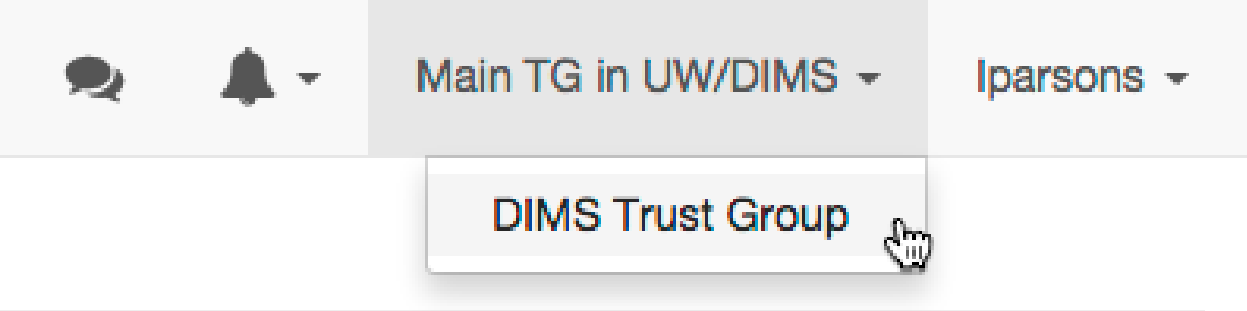


Admin	true
SysAdmin	true
Trust Group	DIMS Trust Group
Trust Group ID	dims
PGP Key Expiration	



Admin	true
SysAdmin	true
Trust Group	Main TG in UW/DIMS
Trust Group ID	main
PGP Key Expiration	

5. To see the users in your currently selected trust group, select `Users > Find DIMS Users` on the nav bar. The list of users in the current trust group will display.
6. Again, change the trust group via the trust group menu in the nav bar. The list of users will change to reflect the users in the new current trust group.



CHAPTER 4

Tickets

There are currently two types of tickets:

- Activity
- Mitigation

Activities are ad hoc ways for users to save information. An activity can have any number of topics associated with it, and users can create topics and add them to an activity.

Mitigation tickets are more structured in that a user can create one by making the appropriate api call and supplying a list of IPs, but the system automatically creates all associated topics, which have specific purposes in a mitigation activity.

4.1 General ticket structure

A *ticket* is described

- A Redis key/value pair where the value is a 1-level hash
- Zero or more associated "topics"

4.2 Topics

A topic is always associated with a parent ticket. The parent key can be derived from the topic key.

When a topic is added to a ticket, its key is added to the set of topic keys owned by the ticket (parent).

Topics are stored in redis as follows:

Table 4.1: Topic storage

Data	Key	Value
Metadata	Topic metadata key	(hash) JSON metadata

Metadata provided by calling method:

```
{
  datatype: (required) 'set' or 'string'
  name: (required) name of topic
  description: (optional) description of topic (default - '')
}
```

```
{
  createdTime: Unix epoch time when topic is created
  modifiedTime: Unix epoch time when topic modified
  num: Topic counter: (via topicCounterKey) - used to ensure uniqueness
}
```

CHAPTER 5

Ticket API

The dashboard server provides a REST API for working with *tickets*.

Note: One API design decision was how *open* to make the API - that is, would the API automatically restrict data sent back for certain requests. For example, we have the concept of *public* and *private* tickets. When a GET request is made, do we want the API to return all tickets or a subset such as all public and all private belonging to the current user?

The API is currently restrictive - a GET request to /api/tickets would return all public activity tickets and all private tickets owned by the calling user.

The API is intended to require authentication. The plan is that the client would have obtained the token for the calling user and included it with the API call. The server will then look up the attributes for the user referenced by the token. These would be:

- username - user name (in Ops-trust) of the calling user
- trustgroup - trust group the user is logged into
- admin - is the user an admin in the trust group

This has not been implemented yet as we need to determine how to integrate this with Trident and their login tokens, as well as have a Trident instance running.

In the interim, the server currently authenticates users with the Dashboard via their Ops-trust (Trident) usernames and passwords and establishes a persistent login session for the user. This can be used to protect the API endpoints when accessing via the Dashboard client, but would prevent other clients from accessing. So the API is not protected by the current authentication mechanism in order to allow other clients access until we get the token authentication implemented.

Until token authentication is implemented however, requests from clients other than the Dashboard will not be able to retrieve user private tickets.

5.1 HTTP Verbs

```
GET     /api/ticket      list
POST    /api/ticket      create
GET     /api/ticket/:id  show
PUT     /api/ticket/:id  update
DELETE  /api/ticket/:id  delete
```

5.2 Responses

The API returns JSON. JSON responses follow the unofficial JSEND spec. See http://labs.omniti.com/labs/jsend/wiki for more information.

Successful requests will return JSON with a status of `success` and a `data` property with the JSON result.

```
{
   "data": <json>
   "status": "success"
}
```

The `data` property will generally be in the following form for one ticket:

```
"data": {
  "ticket": {
    <json describing ticket>
  }
}
```

or for multiple tickets:

```
"data": {
  "tickets": [ <array of json where each one describes a ticket>]
}
```

Since mitigations are a *special* form of ticket, we use the terms `mitigation` and `mitigations` as keys to their response:

```
"data": {
  "mitigation": {
    <json describing mitigation ticket>
  }
}
```

Requests that do not send back data (such as delete) will return with `data` set to `null`:

```
{
  "status": "success",
  "data": null
}
```

Unsuccessful requests will return JSON with an error `message` and a status of `error`:

```
{
    "message": "You do not have permission to access this ticket",
```

```
    "status": "error"
}
```

Requests that failed due to invalid data or parameters submitted may generate a `fail` response:

```
{
  "status": "fail",
  "data": <wrapper for reason request failed>
}
```

For example:

```
{
  "status": "fail",
  "data": {
    "name": "A name for the new ticket is required"
  }
}
```

Note: Currently most errors are reported as `error` rather than `fail`. We are working on refactoring so that errors that should be reported as `fail` are done so.

An HTTP status code is included in the response headers. For example, the following request returns with 400:

```
$ curl -k -I http://192.168.56.103/api/ticket
HTTP/1.1 400 Bad Request
Server: nginx/1.8.0
Date: Wed, 13 Jan 2016 16:44:48 GMT
Content-Type: text/plain; charset=utf-8
Content-Length: 96
Connection: keep-alive
X-Powered-By: Express
ETag: W/"60-kIP4LSNmFtWUQFvEw0Zo/g"
set-cookie: connect.sid=s%3Ah6h88KJrXfxT4Ycabe1dk5aTFY26SRx8.
↪o7d2T9y03YrbbR8ssnmF0tFEpfV9VNI6F7l9oJQEgAg; Path=/; HttpOnly
```

5.3 Retrieve a list of tickets

Returns list of tickets

5.3.1 No parameters

When no parameters are provided, the system defaults to the following parameters:

```
type: 'activity',
```

This will return all public activities plus any private activities belonging to the calling user. Invoked via GET http://dashboard_url/api/ticket/, returns HTTP status code and string reply.

Using curl:

```
curl -k https://dashboard_url/api/ticket/
```

Sample response:

```
{ "data": [
   "ticket:1",
   "ticket:2",
   "ticket:3" ]
}
```

5.3.2 With query parameters

```
private: boolean
type: string
ownedBy: string
open: boolean
```

`type` can be `mitigation` or `activity`. For mitigation tickets, no other parameters are needed, and any extra provided are ignored.

```
$ curl -k http://192.168.56.103/api/ticket?type=mitigation | python -m json.tool
  % Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
                                 Dload  Upload   Total   Spent    Left  Speed
100  1177  100  1177    0     0  34609      0 --:--:-- --:--:-- --:--:-- 35666
{
    "data": [
        {
            "data": [
                [
                    1450004398760,
                    0
                ],
                [
                    1450068277926,
                    6
                ],
                [
                    1450085780836,
                    13
                ],
                [
                    1452404888722,
                    329
                ],
                [
                    1452478109289,
                    343
                ]
            ],
            "ips": {
                "data": [],
                "user": null
            },
            "key": "dims:ticket:mitigation:1",
            "metadata": {
                "createdTime": 1452682798841,
```

```
                "creator": "lparsons",
                "description": "IPs needing mitigation. As you mitigate IPs, submit␣
↪them here.",
                "initialNum": 1408,
                "mitigatedNum": 343,
                "modifiedTime": 1452682798841,
                "name": "Action Needed: 12/12/2015 Compromised IPs",
                "num": 1,
                "open": true,
                "private": false,
                "type": "mitigation",
                "unknownNum": 864
            }
        }
    ],
    "status": "success"
}
```

5.4 Creating an activity

```
/**
 Returns a ticket: ticket key, ticket metadata, list of associated topic keys
 @return HTTP Status code and string reply.

 @example
 Example response:
        {"data": {
          "ticket":
            {"num":"1",
            "creator":"testuser",
            "type":"data",
            "createdTime":"1418060768120",
            "open":"true"},
          "key":"ticket:1",
          "topics":["ticket:1:data:cif:results:result1.txt",
                    "ticket:1:data:cif:results:result2.txt"]
          }
        }
 @example How to invoke
   GET https://dashboard_url/api/ticket/ticket:1

   Using curl:
     curl -k https://dashboard_url/api/ticket/ticket:1

 @param {string} id Ticket key in format ticket:<num>
/

/**
 Creates a new ticket
 @method create
 @return HTTP Status code and string reply.
        {"data": {
          "ticket":
            {"num":"2",
            "creator":"testuser",
```

```
            "type":"data",
            "createdTime":"1418060768120",
            "open":"true"},
          "key":"ticket:2"
          }
        }
 @example

   POST https://dashboard_url/api/ticket/
     body:
     {
       "type": "data",
       "creator": "testuser"
     }

    Using curl:
      curl --data "type=data&creator=testuser" -k https://dashboard_url/api/ticket

 @param {string} type Type of ticket being created
 @param {string} creator Username of user creating ticket (optional if user logged in,
                ignored if user logged in)
/

/**
 Adds a topic (metadata) to a ticket and saves the data (content)
 @method addTopic
 @return HTTP Status code and string reply.
 @example
 Sample json response:

   {"data":{
      "topic":{
        "parent":{
           "num":"12","creator":"testUser","type":"analysis","createdTime":
↪"1418131797522","open":"true"
         },
         "type":"analysis",
         "name":"namesearch:result2",
         "dataType":"hash"
         },
        "content":{"firstname":"bob","lastname":"johnson"},
        "key":"ticket:12:analysis:namesearch:result2"
       }
    }

 @example
 Example URI
     POST https://dashboard_url/api/ticket/ticket:27/topic
     body:
     {
       "name": "cif:results:1418060768120",
       "dataType": "string",
       "content": <string content>
     }
 Note that content in this example could be JSON that is stringified. Content could␣
↪also be content of a
 file, base64'd, as in
     POST https://dashboard_URL/api/ticket/ticket:28/topic
```

```
    body:
    {
      "name": "mal4s:result:resul1.png",
      "dataType": "string",
      "content": <base64 content of a .png file>
    }

Using curl with hash content (content is uri encoded):
     curl --data "name=namesearch:results&dataType=hash&content=%7B%22firstname%22:
↪%22bob%22,%22lastname%22:%22johnson%22%7D" -k https://dashboard_url/api/ticket/
↪ticket:12/topic

A successful response from the curl command might look like the following (line␣
↪feeds added for clarity - reponse is just a string):
     {"data":{
      "topic":{
        "parent":{"num":"12","creator":"testUser","type":"analysis","createdTime":
↪"1418131797522","open":"true"},
        "type":"analysis","name":"namesearch:result2","dataType":"hash"},
        "content":{"firstname":"bob","lastname":"johnson"},"key":
↪"ticket:12:analysis:namesearch:result2"}}

@param {string} id Ticket key in format ticket:<num>
@param {string} name Name of the topic - this represents the last part of the topic␣
↪key after
                     ticket:<num>:<ticket_type>:
@param {string} dataType Redis data structure to store the contents in - can be␣
↪string or hash
@param {string} content  Content to be stored

Note that content is optional if type is string. If no content is specified, then an␣
↪empty string
is stored at the topic key. You would use this if you want to use the contents of a␣
↪file as the
data to be stored. First create the topic with a type of string and no content. Then␣
↪you use the
returned topic key and do an update (PUT) of the topic with the uploaded file.

You cannot overwrite an existing topic with the same key. An error is returned if␣
↪the topic already
exists
/

/**
Retrieves a ticket topic's metadata and content. Invoked via GET

<pre>Sample response:

   { "data":{
     "topic":{
        "parent":{
          "num":"12","creator":"testUser","type":"analysis","createdTime":
↪"1418131797522","open":"true"
         },
        "type":"analysis",
        "name":"namesearch:result2",
        "dataType":"hash"
       },
```

```
        "content":{"firstname":"bob","lastname":"johnson"},
        "key":"ticket:12:analysis:namesearch:result2"
       }
     }</pre>

 @method showTopic

 @example

     GET https://dashboard_url/api/ticket/topic/ticket:27:analysis:namesearch:result2

    Using curl:

      curl -k https://dashboard_url/api/ticket/topic/
↪ticket:27:analysis:namesearch:result2

 @param {string} id Ticket topic key in format ticket:<num>:<type>:<topic_name>
 @return HTTP Status code and string reply.
/

/**
 Updates a ticket topic. You can only update content.
 @method updateTopic
 @return HTTP Status code and string reply.
   {"data":{
      "topic":{
        "parent":{
           "num":"12","creator":"testUser","type":"analysis","createdTime":
↪"1418131797522","open":"true"
         },
         "type":"analysis",
         "name":"namesearch:result2",
         "dataType":"hash"
         },
        "content":{"firstname":"john","lastname":"johnson"},
        "key":"ticket:12:analysis:namesearch:result2"
       }
    }
 @example

     PUT https://dashboard_url/api/ticket/ticket:27/topic
     body:
     {
       "content": <string content>
     }
   Note that content in this example could be JSON that is stringified. Content could␣
↪also be content of a
   file, base64'd, as in
     PUT https://dashboard_URL/api/ticket/ticket:28/topic
     body:
     {
       "content": <base64 content of a .png file>
     }

    Using curl with hash content (content is uri encoded):
      curl --data "content=%7B%22firstname%22:%22john%22,%22lastname%22:%22johnson%22
↪%7D" -k https://dashboard_url/api/ticket/ticket:12/topic
```

```
    A successful response from the curl command might look like the following (line␣
↪feeds added for clarity - reponse is just a string):
      {"data":{
       "topic":{
         "parent":{"num":"12","creator":"testUser","type":"analysis","createdTime":
↪"1418131797522","open":"true"},
         "type":"analysis","name":"namesearch:result2","dataType":"hash"},
         "content":{"firstname":"john","lastname":"johnson"},"key":
↪"ticket:12:analysis:namesearch:result2"}}

 @param {string} id Topic key in format ticket:<num>:<type>:<topic_name>
 @param {string} content  Content to be stored

/
```

CHAPTER 6

Diagnosing System Problems and Outages

This chapter covers using `dimscli` as a distributed shell for diagnosing problems throughout a DIMS deployment.

Ansible has two primary CLI programs, `ansible` and `ansible-playbook`. Both of these programs are passed a set of hosts on which they are to operate using an Inventory.

Note: Read about Ansible and how it is used by the DIMS project in Section ansibleplaybooks:ansiblefundamentals of ansibleplaybooks:ansibleplaybooks.

```
[dimsenv] dittrich@dimsdemo1:~/dims/git/python-dimscli (develop*) $ cat complete_
↪inventory
[all]
floyd2-p.prisem.washington.edu
foswiki-int.prisem.washington.edu
git.prisem.washington.edu
hub.prisem.washington.edu
jenkins-int.prisem.washington.edu
jira-int.prisem.washington.edu
lapp-int.prisem.washington.edu
lapp.prisem.washington.edu
linda-vm1.prisem.washington.edu
rabbitmq.prisem.washington.edu
sso.prisem.washington.edu
time.prisem.washington.edu
u12-dev-svr-1.prisem.washington.edu
u12-dev-ws-1.prisem.washington.edu
wellington.prisem.washington.edu
```

Using this inventory, the modules `command` and `shell` can be used to run commands as needed to diagnose all of these hosts at once.

```
[dimsenv] dittrich@dimsdemo1:~/dims/git/python-dimscli (develop*) $ dimscli ansible␣
↪command --program "uptime" --inventory complete_inventory --remote-port 8422 --
↪remote-user dittrich
```

```
+-----------------------------------+--------+-----------------------------------
↪----------------------------------+
| Host                              | Status | Results                          ␣
↪                                  |
+-----------------------------------+--------+-----------------------------------
↪----------------------------------+
| rabbitmq.prisem.washington.edu    | GOOD   |  22:07:53 up 33 days,  4:32,  1 user,
↪  load average: 0.07, 0.13, 0.09   |
| wellington.prisem.washington.edu  | GOOD   |  22:07:57 up 159 days, 12:16,  1␣
↪user,  load average: 1.16, 0.86, 0.58  |
| linda-vm1.prisem.washington.edu   | GOOD   |  22:07:54 up 159 days, 12:03,  1␣
↪user,  load average: 0.00, 0.01, 0.05  |
| git.prisem.washington.edu         | GOOD   |  22:07:54 up 159 days, 12:03,  2␣
↪users,  load average: 0.00, 0.01, 0.05 |
| time.prisem.washington.edu        | GOOD   |  22:07:55 up 33 days,  4:33,  2␣
↪users,  load average: 0.01, 0.07, 0.12  |
| jenkins-int.prisem.washington.edu | GOOD   |  22:07:55 up 159 days, 12:03,  1␣
↪user,  load average: 0.00, 0.01, 0.05  |
| u12-dev-ws-1.prisem.washington.edu | GOOD   |  22:07:56 up 159 days, 12:03,  1␣
↪user,  load average: 0.00, 0.02, 0.05  |
| sso.prisem.washington.edu         | GOOD   |  22:07:56 up 159 days, 12:03,  1␣
↪user,  load average: 0.00, 0.01, 0.05  |
| lapp-int.prisem.washington.edu    | GOOD   |  22:07:54 up 159 days, 12:04,  2␣
↪users,  load average: 0.00, 0.01, 0.05 |
| foswiki-int.prisem.washington.edu | GOOD   |  22:07:55 up 159 days, 12:04,  1␣
↪user,  load average: 0.00, 0.01, 0.05  |
| u12-dev-svr-1.prisem.washington.edu | GOOD   |  22:07:59 up 155 days, 14:56,  1␣
↪user,  load average: 0.05, 0.08, 0.06  |
| hub.prisem.washington.edu         | GOOD   |  06:07:53 up 141 days, 12:19,  1␣
↪user,  load average: 0.08, 0.03, 0.05  |
| floyd2-p.prisem.washington.edu    | GOOD   |  22:07:53 up 33 days,  4:32,  1 user,
↪  load average: 0.00, 0.01, 0.05   |
| jira-int.prisem.washington.edu    | GOOD   |  22:07:54 up 159 days, 12:03,  2␣
↪users,  load average: 0.00, 0.01, 0.05 |
| lapp.prisem.washington.edu        | GOOD   |  22:07:54 up 159 days, 12:04,  2␣
↪users,  load average: 0.00, 0.01, 0.05 |
+-----------------------------------+--------+-----------------------------------
↪----------------------------------+
```

```
1  To: dims-devops@uw.ops-trust.net
2  From: Jenkins <dims@eclipse.prisem.washington.edu>
3  Subject: [dims devops] [Jenkins] [FAILURE] jenkins-update-cifbulk-server-develop-16
4  Date: Thu Jan 14 20:35:21 PST 2016
5  Message-ID: <20160115043521.C7D5E1C004F@jenkins>
6
7  Started by an SCM change
8  [EnvInject] - Loading node environment variables.
9  Building in workspace /var/lib/jenkins/jobs/update-cifbulk-server-develop/workspace
10
11 Deleting project workspace... done
12
13 [ssh-agent] Using credentials ansible (Ansible user ssh key - root)
14 [ssh-agent] Looking for ssh-agent implementation...
15 [ssh-agent]   Java/JNR ssh-agent
16 [ssh-agent] Started.
17
18  ...
19
```

```
TASK: [cifbulk-server | Make config change available and restart if updating existing] ***
<rabbitmq.prisem.washington.edu> REMOTE_MODULE command . /opt/dims/envs/dimsenv/bin/activate && supervisorctl -c /etc/supervisord.conf reread #USE_SHELL
failed: [rabbitmq.prisem.washington.edu] => (item=reread) => {"changed": true, "cmd": ". /opt/dims/envs/dimsenv/bin/activate && supervisorctl -c /etc/supervisord.conf reread", "delta": "0:00:00.229614", "end": "2016-01-14 20:34:49.409784", "item": "reread", "rc": 2, "start": "2016-01-14 20:34:49.180170"}
stderr: Error: could not find config file /etc/supervisord.conf
For help, use /usr/bin/supervisorctl -h
<rabbitmq.prisem.washington.edu> REMOTE_MODULE command . /opt/dims/envs/dimsenv/bin/activate && supervisorctl -c /etc/supervisord.conf update #USE_SHELL
failed: [rabbitmq.prisem.washington.edu] => (item=update) => {"changed": true, "cmd": ". /opt/dims/envs/dimsenv/bin/activate && supervisorctl -c /etc/supervisord.conf update", "delta": "0:00:00.235882", "end": "2016-01-14 20:34:50.097224", "item": "update", "rc": 2, "start": "2016-01-14 20:34:49.861342"}
stderr: Error: could not find config file /etc/supervisord.conf
For help, use /usr/bin/supervisorctl -h

FATAL: all hosts have already failed -- aborting

PLAY RECAP ********************************************************************
           to retry, use: --limit @/var/lib/jenkins/cifbulk-server-configure.retry

rabbitmq.prisem.washington.edu : ok=11   changed=4    unreachable=0    failed=1

Build step 'Execute shell' marked build as failure
[ssh-agent] Stopped.
Warning: you have no plugins providing access control for builds, so falling back to legacy behavior of permitting any downstream builds to be triggered
Finished: FAILURE
--
[[ UW/DIMS ]]: All message content remains the property of the author
and must not be forwarded or redistributed without explicit permission.
```

```
[dimsenv] dittrich@dimsdemo1:~/dims/git/ansible-playbooks (develop*) $ grep -r supervisord.conf
roles/supervisor-install/tasks/main.yml:  template: "src=supervisord.conf.j2 dest={{ dims_supervisord_conf }} owner=root group=root"
roles/supervisor-install/tasks/main.yml:  file: path=/etc/dims-supervisord.conf state=absent
roles/supervisor-install/templates/supervisor.j2:DAEMON_OPTS="-c {{ dims_supervisord_conf }} $DAEMON_OPTS"
roles/cifbulk-server/tasks/main.yml:  shell: ". {{ dimsenv_activate }} && supervisorctl -c {{ dims_supervisord_conf }} {{ item }}"
roles/cifbulk-server/tasks/main.yml:  shell: ". {{ dimsenv_activate }} && supervisorctl -c {{ dims_supervisord_conf }} start {{ name_base }}:"
roles/prisem-scripts-deploy/tasks/main.yml:  shell: ". {{ dimsenv_activate }} && supervisorctl -c {{ dims_supervisord_conf }} restart {{ item }}:"
roles/anon-server/tasks/main.yml:  shell: ". {{ dimsenv_activate }} && supervisorctl -c {{ dims_supervisord_conf }} {{ item }}"
roles/anon-server/tasks/main.yml:  shell: ". {{ dimsenv_activate }} && supervisorctl -c {{ dims_supervisord_conf }} start {{ name_base }}:"
roles/consul-install/tasks/main.yml:  shell: ". {{ dimsenv_activate }} && supervisorctl -c {{ dims_supervisord_conf }} remove {{ consul_basename }}"
roles/consul-install/tasks/main.yml:  shell: ". {{ dimsenv_activate }} && supervisorctl -c {{ dims_supervisord_conf }} {{ item }}"
roles/consul-install/tasks/main.yml:  shell: ". {{ dimsenv_activate }} && supervisorctl -c {{ dims_supervisord_conf }} start {{ consul_basename }}:"
```

```
roles/crosscor-server/tasks/main.yml:  shell: ". {{ dimsenv_activate }} &&␣
↪supervisorctl -c {{ dims_supervisord_conf }} {{ item }}"
roles/crosscor-server/tasks/main.yml:  shell: ". {{ dimsenv_activate }} &&␣
↪supervisorctl -c {{ dims_supervisord_conf }} start {{ name_base }}:"
group_vars/all:dims_supervisord_conf: '/etc/supervisord.conf'
```

```
[dimsenv] dittrich@dimsdemo1:~/dims/git/python-dimscli (develop*) $ dimscli ansible␣
↪shell --program "find /etc -name supervisord.conf" --inventory complete_inventory --
↪remote-port 8422 --remote-u
ser dittrich
+-------------------------------------+--------+---------------------------------+
| Host                                | Status | Results                         |
+-------------------------------------+--------+---------------------------------+
| rabbitmq.prisem.washington.edu      | GOOD   | /etc/supervisor/supervisord.conf |
| wellington.prisem.washington.edu    | GOOD   |                                 |
| hub.prisem.washington.edu           | GOOD   |                                 |
| git.prisem.washington.edu           | GOOD   | /etc/supervisor/supervisord.conf |
| u12-dev-ws-1.prisem.washington.edu  | GOOD   |                                 |
| sso.prisem.washington.edu           | GOOD   |                                 |
| jenkins-int.prisem.washington.edu   | GOOD   | /etc/supervisor/supervisord.conf |
| foswiki-int.prisem.washington.edu   | GOOD   |                                 |
| lapp-int.prisem.washington.edu      | GOOD   |                                 |
| u12-dev-svr-1.prisem.washington.edu | GOOD   | /etc/supervisor/supervisord.conf |
| linda-vm1.prisem.washington.edu     | GOOD   |                                 |
| lapp.prisem.washington.edu          | GOOD   |                                 |
| floyd2-p.prisem.washington.edu      | GOOD   |                                 |
| jira-int.prisem.washington.edu      | GOOD   | /etc/supervisor/supervisord.conf |
| time.prisem.washington.edu          | GOOD   |                                 |
+-------------------------------------+--------+---------------------------------+
```

```
[dimsenv] dittrich@dimsdemo1:~/dims/git/python-dimscli (develop*) $ dimscli ansible␣
↪shell --program "find /etc -name '*supervisor'*" --inventory complete_inventory --
↪remote-port 8422 --remote-use
r dittrich
+-------------------------------------+--------+-------------------------------------
↪-----------+
| Host                                | Status | Results                          ␣
↪           |
+-------------------------------------+--------+-------------------------------------
↪-----------+
| rabbitmq.prisem.washington.edu      | GOOD   | /etc/rc0.d/K20supervisor         ␣
↪           |
|                                     |        | /etc/rc3.d/S20supervisor         ␣
↪           |
|                                     |        | /etc/rc1.d/K20supervisor         ␣
↪           |
|                                     |        | /etc/default/supervisor          ␣
↪           |
|                                     |        | /etc/rc2.d/S20supervisor         ␣
↪           |
|                                     |        | /etc/rc6.d/K20supervisor         ␣
↪           |
|                                     |        | /etc/supervisor                  ␣
↪           |
|                                     |        | /etc/supervisor/supervisord.conf.
↪20140214204135 |
|                                     |        | /etc/supervisor/supervisord.conf.
↪20140214200547 |
```

```
|                                          |         | /etc/supervisor/supervisord.conf.
↪20140616162335 |
|                                          |         | /etc/supervisor/supervisord.conf.
↪20140814132409 |
|                                          |         | /etc/supervisor/supervisord.conf.
↪20140616162451 |
|                                          |         | /etc/supervisor/supervisord.conf.
↪20140616162248 |
|                                          |         | /etc/supervisor/supervisord.conf.
↪20140131230939 |
|                                          |         | /etc/supervisor/supervisord.conf.
↪20140222154901 |
|                                          |         | /etc/supervisor/supervisord.conf.
↪20140214194415 |
|                                          |         | /etc/supervisor/supervisord.conf.
↪20140222155042 |
|                                          |         | /etc/supervisor/supervisord.conf.
↪20150208174308 |
|                                          |         | /etc/supervisor/supervisord.conf.
↪20140814132717 |
|                                          |         | /etc/supervisor/supervisord.conf.
↪20140215134451 |
|                                          |         | /etc/supervisor/supervisord.conf.
↪20150208174742 |
|                                          |         | /etc/supervisor/supervisord.conf.
↪20140911193305 |
|                                          |         | /etc/supervisor/supervisord.conf.
↪20140219200951 |
|                                          |         | /etc/supervisor/supervisord.conf.
↪20140911202633 |
|                                          |         | /etc/supervisor/supervisord.conf
↪           |
|                                          |         | /etc/supervisor/supervisord.conf.
↪20140222154751 |
|                                          |         | /etc/supervisor/supervisord.conf.
↪20150208174403 |
|                                          |         | /etc/supervisor/supervisord.conf.
↪20140814132351 |
|                                          |         | /etc/supervisor/supervisord.conf.
↪20140814132759 |
|                                          |         | /etc/rc4.d/S20supervisor
↪           |
|                                          |         | /etc/init.d/supervisor
↪           |
|                                          |         | /etc/rc5.d/S20supervisor
↪           |
| wellington.prisem.washington.edu         | GOOD    |
↪           |
| linda-vm1.prisem.washington.edu          | GOOD    | /etc/rc0.d/K20supervisor
↪           |
|                                          |         | /etc/rc3.d/S20supervisor
↪           |
|                                          |         | /etc/rc1.d/K20supervisor
↪           |
|                                          |         | /etc/rc2.d/S20supervisor
↪           |
|                                          |         | /etc/rc6.d/K20supervisor
↪           |
```

```
|                                        |         | /etc/supervisor                                  ␣
↪            |
|                                        |         | /etc/rc4.d/S20supervisor                         ␣
↪            |
|                                        |         | /etc/dims-supervisord.conf                       ␣
↪            |
|                                        |         | /etc/init.d/supervisor                           ␣
↪            |
|                                        |         | /etc/rc5.d/S20supervisor                         ␣
↪            |
| git.prisem.washington.edu              | GOOD    | /etc/rc0.d/K20supervisor                         ␣
↪            |
|                                        |         | /etc/rc3.d/S20supervisor                         ␣
↪            |
|                                        |         | /etc/rc1.d/K20supervisor                         ␣
↪            |
|                                        |         | /etc/default/supervisor                          ␣
↪            |
|                                        |         | /etc/rc2.d/S20supervisor                         ␣
↪            |
|                                        |         | /etc/rc6.d/K20supervisor                         ␣
↪            |
|                                        |         | /etc/supervisor                                  ␣
↪            |
|                                        |         | /etc/supervisor/supervisord.conf                 ␣
↪            |
|                                        |         | /etc/rc4.d/S20supervisor                         ␣
↪            |
|                                        |         | /etc/init.d/supervisor                           ␣
↪            |
|                                        |         | /etc/rc5.d/S20supervisor                         ␣
↪            |
| time.prisem.washington.edu             | GOOD    |                                                  ␣
↪            |
| jenkins-int.prisem.washington.edu      | GOOD    | /etc/rc0.d/K20supervisor                         ␣
↪            |
|                                        |         | /etc/rc3.d/S20supervisor                         ␣
↪            |
|                                        |         | /etc/rc1.d/K20supervisor                         ␣
↪            |
|                                        |         | /etc/default/supervisor                          ␣
↪            |
|                                        |         | /etc/rc2.d/S20supervisor                         ␣
↪            |
|                                        |         | /etc/rc6.d/K20supervisor                         ␣
↪            |
|                                        |         | /etc/supervisor                                  ␣
↪            |
|                                        |         | /etc/supervisor/supervisord.conf                 ␣
↪            |
|                                        |         | /etc/rc4.d/S20supervisor                         ␣
↪            |
|                                        |         | /etc/init.d/supervisor                           ␣
↪            |
|                                        |         | /etc/rc5.d/S20supervisor                         ␣
↪            |
| u12-dev-ws-1.prisem.washington.edu     | GOOD    |                                                  ␣
↪            |
```

```
| sso.prisem.washington.edu            | GOOD   |                                                ␣
↪            |
| lapp-int.prisem.washington.edu       | GOOD   |                                                ␣
↪            |
| foswiki-int.prisem.washington.edu    | GOOD   |                                                ␣
↪            |
| u12-dev-svr-1.prisem.washington.edu | GOOD   | /etc/rc2.d/S20supervisor                       ␣
↪            |
|                                      |        | /etc/rc4.d/S20supervisor                       ␣
↪            |
|                                      |        | /etc/init.d/supervisor                         ␣
↪            |
|                                      |        | /etc/rc5.d/S20supervisor                       ␣
↪            |
|                                      |        | /etc/rc3.d/S20supervisor                       ␣
↪            |
|                                      |        | /etc/supervisor                                ␣
↪            |
|                                      |        | /etc/supervisor/supervisord.conf               ␣
↪            |
|                                      |        | /etc/rc6.d/K20supervisor                       ␣
↪            |
|                                      |        | /etc/rc1.d/K20supervisor                       ␣
↪            |
|                                      |        | /etc/rc0.d/K20supervisor                       ␣
↪            |
| hub.prisem.washington.edu            | GOOD   |                                                ␣
↪            |
| floyd2-p.prisem.washington.edu       | GOOD   |                                                ␣
↪            |
| jira-int.prisem.washington.edu       | GOOD   | /etc/rc0.d/K20supervisor                       ␣
↪            |
|                                      |        | /etc/rc3.d/S20supervisor                       ␣
↪            |
|                                      |        | /etc/rc1.d/K20supervisor                       ␣
↪            |
|                                      |        | /etc/default/supervisor                        ␣
↪            |
|                                      |        | /etc/rc2.d/S20supervisor                       ␣
↪            |
|                                      |        | /etc/rc6.d/K20supervisor                       ␣
↪            |
|                                      |        | /etc/supervisor                                ␣
↪            |
|                                      |        | /etc/supervisor/supervisord.conf               ␣
↪            |
|                                      |        | /etc/rc4.d/S20supervisor                       ␣
↪            |
|                                      |        | /etc/init.d/supervisor                         ␣
↪            |
|                                      |        | /etc/rc5.d/S20supervisor                       ␣
↪            |
| lapp.prisem.washington.edu           | GOOD   |                                                ␣
↪            |
+--------------------------------------+--------+------------------------------------------
↪------------+
```

While the concept of putting a list of host names into a file with a label is simple to understand, it is not very flexible

or scalable. Ansible supports a concept called a Dynamic Inventory. Rather than passing a hosts file using `-i` or `--inventory`, you can pass a Python script that produces a special JSON object.

What is not very widely known is that you can also trigger creation of a dynamic inventory within `ansible` or `ansible-playbook` by passing a *list* for the `-i` or `--inventory` option. Rather than creating a temporary file with `[all]` at the top, followed by a list of three host names, then passing that file with `-i` or `--inventory`, just pass a comma-separated list instead:

```
[dimsenv] dittrich@dimsdemo1:~/dims/git/python-dimscli (develop*) $ dimscli ansible␣
↪shell --program "find /etc -name supervisord.conf" --inventory rabbitmq.prisem.
↪washington.edu,time.prisem.washi
ngton.edu,u12-dev-svr-1.prisem.washington.edu --remote-port 8422 --remote-user␣
↪dittrich
+-------------------------------------+--------+---------------------------------+
| Host                                | Status | Results                         |
+-------------------------------------+--------+---------------------------------+
| rabbitmq.prisem.washington.edu      | GOOD   | /etc/supervisor/supervisord.conf |
| time.prisem.washington.edu          | GOOD   |                                 |
| u12-dev-svr-1.prisem.washington.edu | GOOD   | /etc/supervisor/supervisord.conf |
+-------------------------------------+--------+---------------------------------+
```

There is a subtle trick for passing just a single host, and that is to pass the name with a trailing comma (`,`), as seen here:

```
[dimsenv] dittrich@dimsdemo1:~/dims/git/python-dimscli (develop*) $ dimscli ansible␣
↪shell --program "find /etc -name supervisord.conf" --inventory rabbitmq.prisem.
↪washington.edu, --remote-port 84
22 --remote-user dittrich
+--------------------------------+--------+---------------------------------+
| Host                           | Status | Results                         |
+--------------------------------+--------+---------------------------------+
| rabbitmq.prisem.washington.edu | GOOD   | /etc/supervisor/supervisord.conf |
+--------------------------------+--------+---------------------------------+
```

CHAPTER 7

License

```
Berkeley Three Clause License
=============================

Copyright (c) 2014, 2015 University of Washington. All rights reserved.

Redistribution and use in source and binary forms, with or without
modification, are permitted provided that the following conditions are met:

1. Redistributions of source code must retain the above copyright notice, this
list of conditions and the following disclaimer.

2. Redistributions in binary form must reproduce the above copyright notice,
this list of conditions and the following disclaimer in the documentation
and/or other materials provided with the distribution.

3. Neither the name of the copyright holder nor the names of its contributors
may be used to endorse or promote products derived from this software without
specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND
ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED
WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE
DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT HOLDER OR CONTRIBUTORS BE LIABLE
FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL
DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR
SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER
CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY,
OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE
OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.
```

Section author: Dave Dittrich *dittrich@u.washington.edu*