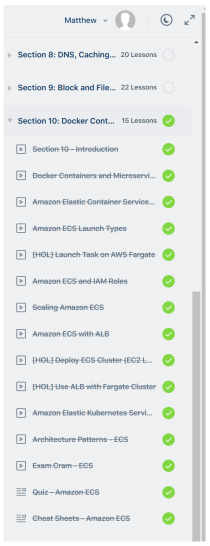
Matthew Pacey
CS 561

Assignment 4

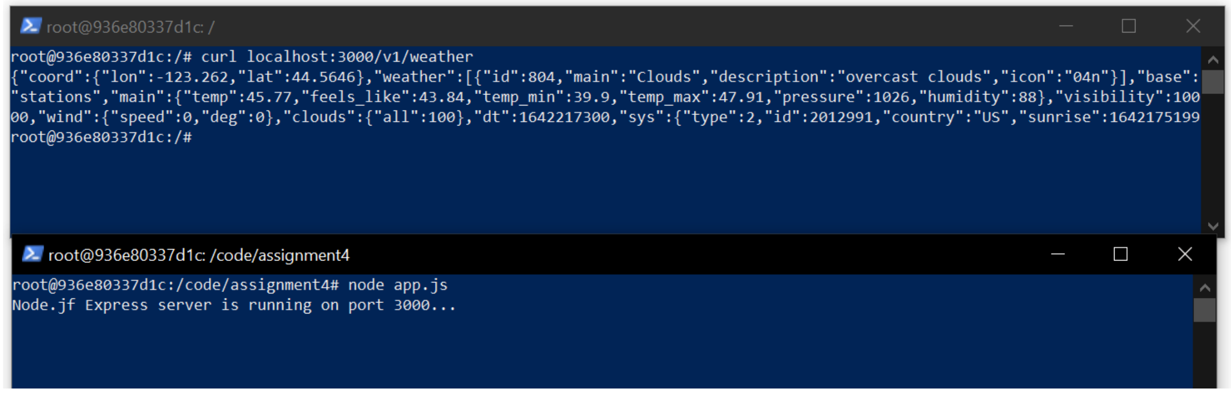
1. AWS Training #10



Matthew

Section 8: DNS, Caching... 20 Lessons
Section 9: Block and File... 22 Lessons
Section 10: Docker Cont... 15 Lessons ✓
~~Section 10 - Introduction~~ ✓
~~Docker Containers and Microservi...~~ ✓
~~Amazon Elastic Container Service...~~ ✓
~~Amazon ECS Launch Types~~ ✓
~~[HOL] Launch Task on AWS Fargate~~ ✓
~~Amazon ECS and IAM Roles~~ ✓
~~Scaling Amazon ECS~~ ✓
~~Amazon ECS with ALB~~ ✓
~~[HOL] Deploy ECS Cluster (EC2 L...~~ ✓
~~[HOL] Use ALB with Fargate Cluster~~ ✓
~~Amazon Elastic Kubernetes Servi...~~ ✓
~~Architecture Patterns - ECS~~ ✓
~~Exam Cram - ECS~~ ✓
~~Quiz - Amazon ECS~~ ✓
~~Cheat Sheets - Amazon ECS~~ ✓

2. Done
3. 4 Docker Configurations
    a. Server and client in same docker container



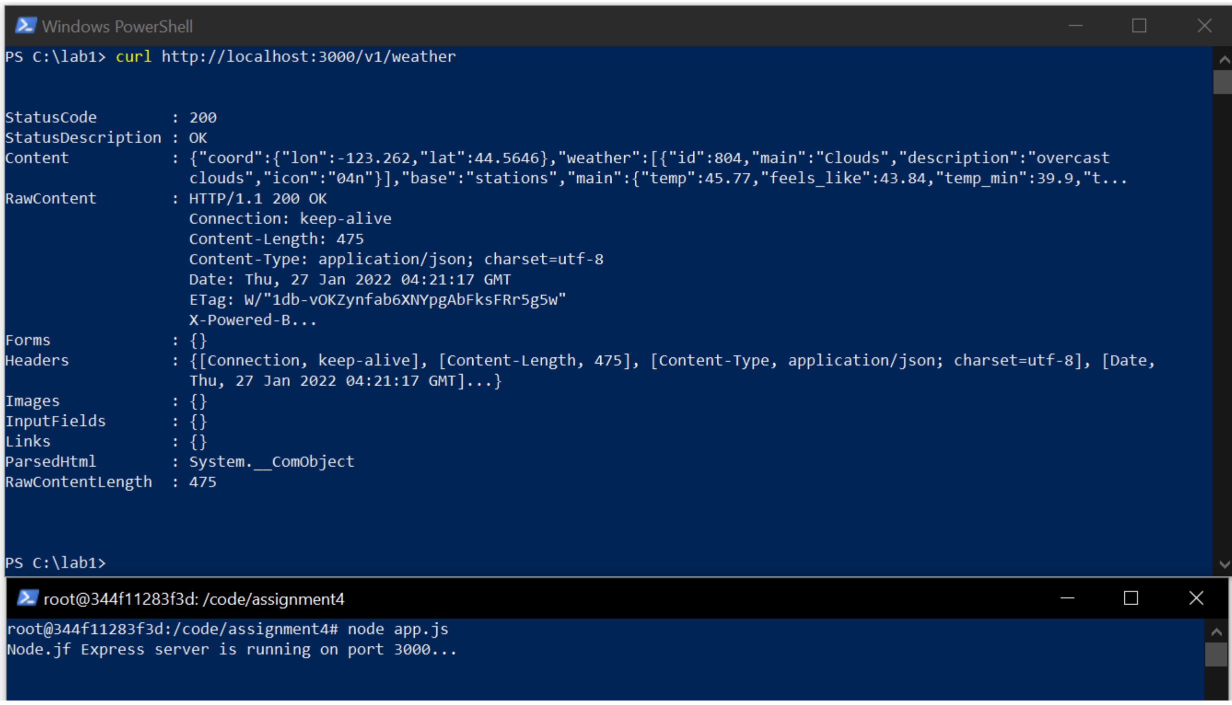
root@936e80337d1c: /

```
root@936e80337d1c:/# curl localhost:3000/v1/weather
{"coord":{"lon":-123.262,"lat":44.5646},"weather":[{"id":804,"main":"Clouds","description":"overcast clouds","icon":"04n"}],"base":
"stations","main":{"temp":45.77,"feels_like":43.84,"temp_min":39.9,"temp_max":47.91,"pressure":1026,"humidity":88},"visibility":100
00,"wind":{"speed":0,"deg":0},"clouds":{"all":100},"dt":1642217300,"sys":{"type":2,"id":2012991,"country":"US","sunrise":1642175199
root@936e80337d1c:/#
```

root@936e80337d1c: /code/assignment4

```
root@936e80337d1c:/code/assignment4# node app.js
Node.jf Express server is running on port 3000...
```

b. Server in docker, client on host



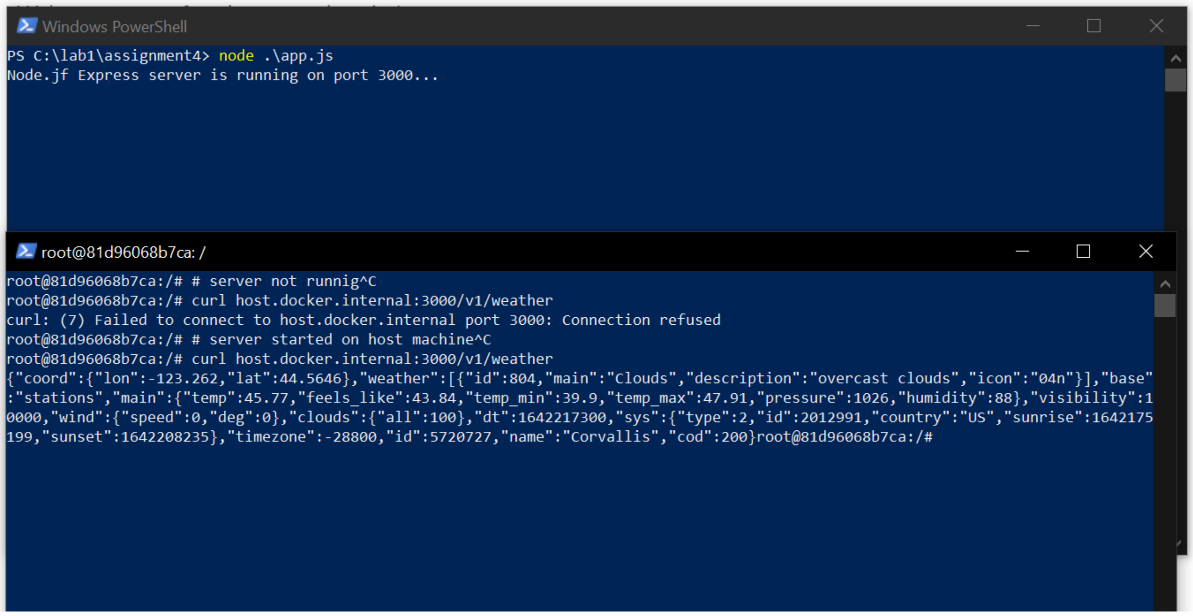
Windows PowerShell

```
PS C:\lab1> curl http://localhost:3000/v1/weather


StatusCode        : 200
StatusDescription : OK
Content           : {"coord":{"lon":-123.262,"lat":44.5646},"weather":[{"id":804,"main":"Clouds","description":"overcast
                    clouds","icon":"04n"}],"base":"stations","main":{"temp":45.77,"feels_like":43.84,"temp_min":39.9,"t...
RawContent        : HTTP/1.1 200 OK
                    Connection: keep-alive
                    Content-Length: 475
                    Content-Type: application/json; charset=utf-8
                    Date: Thu, 27 Jan 2022 04:21:17 GMT
                    ETag: W/"1db-vOKZynfab6XNYpgAbFksFRr5g5w"
                    X-Powered-B...
Forms             : {}
Headers           : {[Connection, keep-alive], [Content-Length, 475], [Content-Type, application/json; charset=utf-8], [Date,
                    Thu, 27 Jan 2022 04:21:17 GMT]...}
Images            : {}
InputFields       : {}
Links             : {}
ParsedHtml        : System.__ComObject
RawContentLength  : 475



PS C:\lab1>
```

root@344f11283f3d: /code/assignment4

```
root@344f11283f3d:/code/assignment4# node app.js
Node.jf Express server is running on port 3000...
```

c. Server on host, client in docker



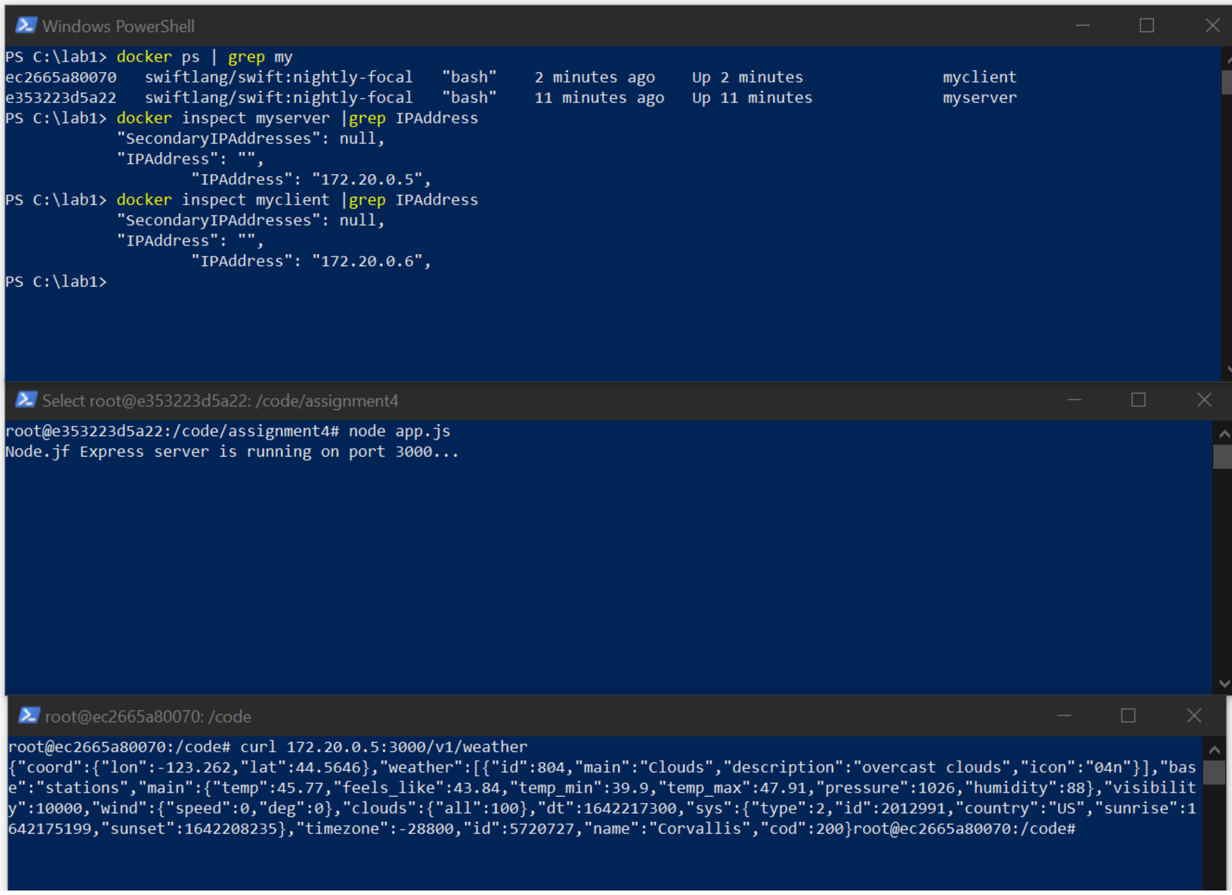
Windows PowerShell

```
PS C:\lab1\assignment4> node .\app.js
Node.jf Express server is running on port 3000...
```

root@81d96068b7ca: /

```
root@81d96068b7ca:/# # server not runnig^C
root@81d96068b7ca:/# curl host.docker.internal:3000/v1/weather
curl: (7) Failed to connect to host.docker.internal port 3000: Connection refused
root@81d96068b7ca:/# # server started on host machine^C
root@81d96068b7ca:/# curl host.docker.internal:3000/v1/weather
{"coord":{"lon":-123.262,"lat":44.5646},"weather":[{"id":804,"main":"Clouds","description":"overcast clouds","icon":"04n"}],"base"
:"stations","main":{"temp":45.77,"feels_like":43.84,"temp_min":39.9,"temp_max":47.91,"pressure":1026,"humidity":88},"visibility":1
0000,"wind":{"speed":0,"deg":0},"clouds":{"all":100},"dt":1642217300,"sys":{"type":2,"id":2012991,"country":"US","sunrise":1642175
199,"sunset":1642208235},"timezone":-28800,"id":5720727,"name":"Corvallis","cod":200}root@81d96068b7ca:/#
```

d. Server in docker, client in another docker container



Windows PowerShell

```
PS C:\lab1> docker ps | grep my
ec2665a80070   swiftlang/swift:nightly-focal   "bash"    2 minutes ago    Up 2 minutes                  myclient
e353223d5a22   swiftlang/swift:nightly-focal   "bash"    11 minutes ago   Up 11 minutes                 myserver
PS C:\lab1> docker inspect myserver |grep IPAddress
            "SecondaryIPAddresses": null,
            "IPAddress": "",
                    "IPAddress": "172.20.0.5",
PS C:\lab1> docker inspect myclient |grep IPAddress
            "SecondaryIPAddresses": null,
            "IPAddress": "",
                    "IPAddress": "172.20.0.6",
PS C:\lab1>
```

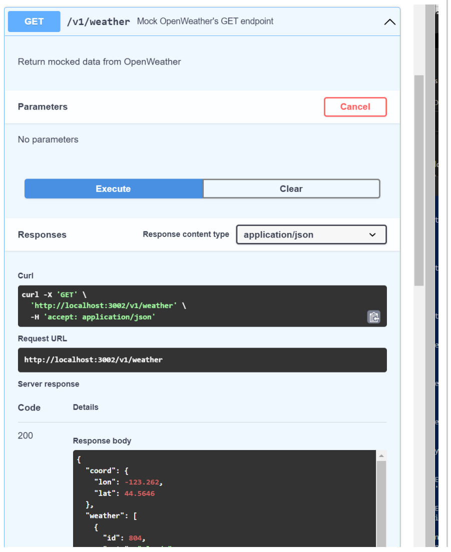
Select root@e353223d5a22: /code/assignment4

```
root@e353223d5a22:/code/assignment4# node app.js
Node.jf Express server is running on port 3000...
```

root@ec2665a80070: /code

```
root@ec2665a80070:/code# curl 172.20.0.5:3000/v1/weather
{"coord":{"lon":-123.262,"lat":44.5646},"weather":[{"id":804,"main":"Clouds","description":"overcast clouds","icon":"04n"}],"bas
e":"stations","main":{"temp":45.77,"feels_like":43.84,"temp_min":39.9,"temp_max":47.91,"pressure":1026,"humidity":88},"visibilit
y":10000,"wind":{"speed":0,"deg":0},"clouds":{"all":100},"dt":1642217300,"sys":{"type":2,"id":2012991,"country":"US","sunrise":1
642175199,"sunset":1642208235},"timezone":-28800,"id":5720727,"name":"Corvallis","cod":200}root@ec2665a80070:/code#
```

4. Source code https://github.com/osu-mp/cs561-assignments/blob/main/4/swagger.yaml


GET /v1/weather Mock OpenWeather's GET endpoint

Return mocked data from OpenWeather

Parameters

No parameters

Responses — Response content type: application/json

Curl

```
curl -X 'GET' \
  'http://localhost:3002/v1/weather' \
  -H 'accept: application/json'
```

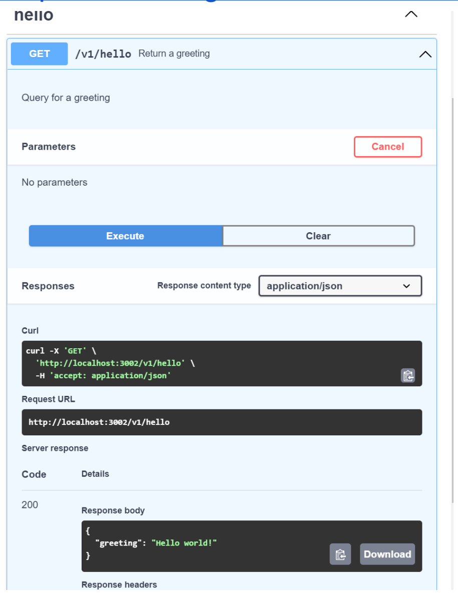
Request URL

```
http://localhost:3002/v1/weather
```

Server response

Code	Details
200	Response body

```
{
  "coord": {
    "lon": -123.262,
    "lat": 44.5646
  },
  "weather": [
    {
      "id": 804,
```


hello

GET /v1/hello Return a greeting

Query for a greeting

Parameters

No parameters

Responses — Response content type: application/json

Curl

```
curl -X 'GET' \
  'http://localhost:3002/v1/hello' \
  -H 'accept: application/json'
```

Request URL

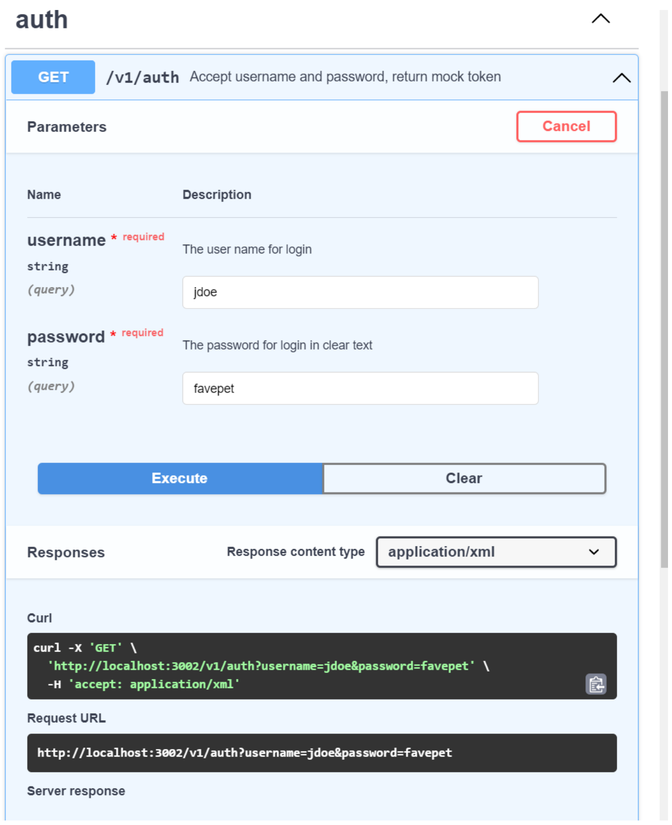
```
http://localhost:3002/v1/hello
```

Server response

Code	Details
200	Response body

```
{
  "greeting": "Hello world!"
}
```

Response headers


auth

GET /v1/auth Accept username and password, return mock token

Parameters

Name	Description
username * required string (query)	The user name for login jdoe
password * required string (query)	The password for login in clear text favepet

Responses — Response content type: application/xml

Curl

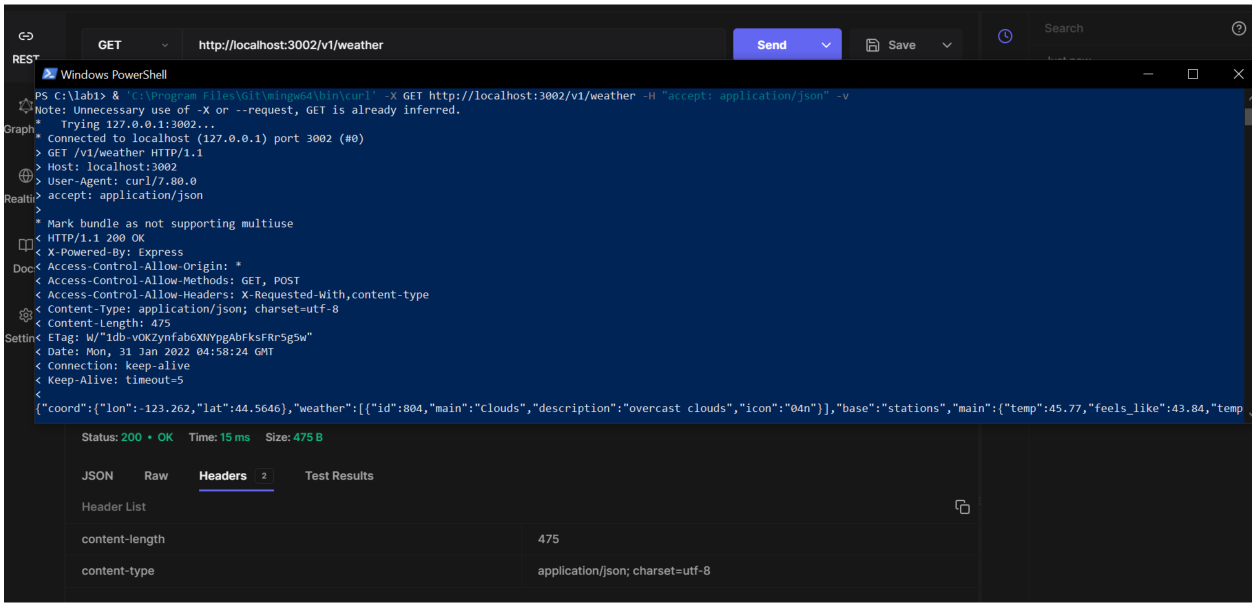
```
curl -X 'GET' \
  'http://localhost:3002/v1/auth?username=jdoe&password=favepet' \
  -H 'accept: application/xml'
```

Request URL

```
http://localhost:3002/v1/auth?username=jdoe&password=favepet
```

Server response

5. **Weather**
The hopscotch headers only had content-length and content type, whereas the curl headers included the access-control headers that were allowed in app.js

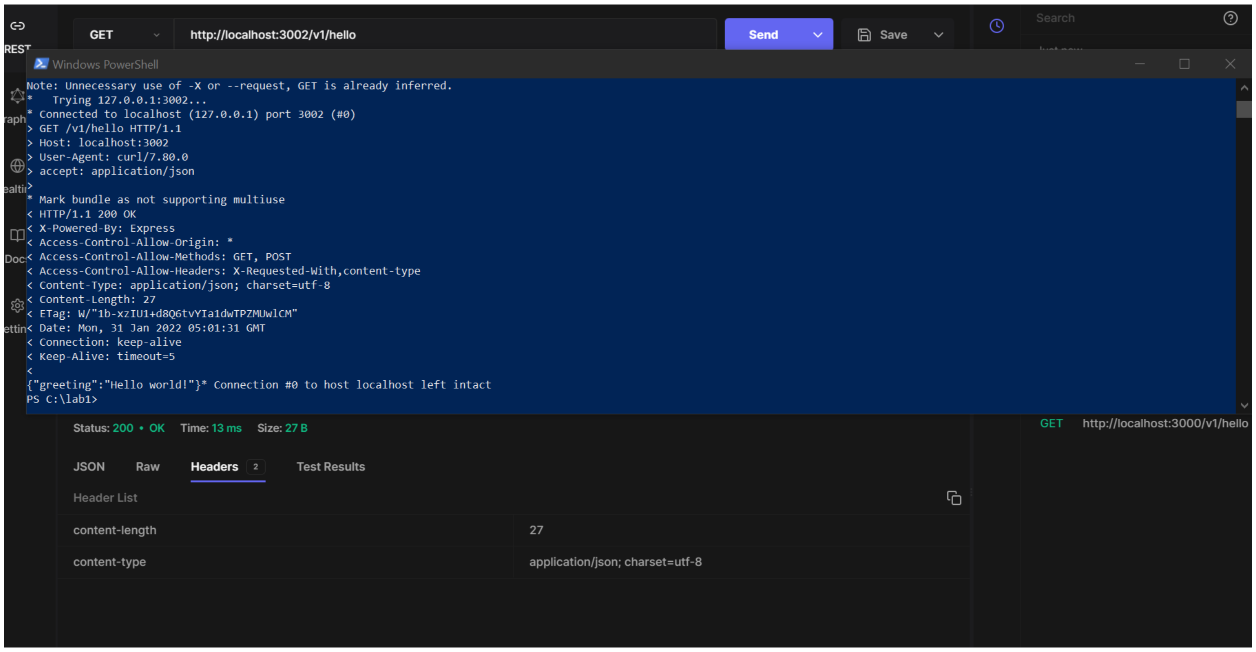


```
PS C:\lab1> & 'C:\Program Files\Git\mingw64\bin\curl' -X GET http://localhost:3002/v1/weather -H "accept: application/json" -v
Note: Unnecessary use of -X or --request, GET is already inferred.
*   Trying 127.0.0.1:3002...
* Connected to localhost (127.0.0.1) port 3002 (#0)
> GET /v1/weather HTTP/1.1
> Host: localhost:3002
> User-Agent: curl/7.80.0
> accept: application/json
>
* Mark bundle as not supporting multiuse
< HTTP/1.1 200 OK
< X-Powered-By: Express
< Access-Control-Allow-Origin: *
< Access-Control-Allow-Methods: GET, POST
< Access-Control-Allow-Headers: X-Requested-With,content-type
< Content-Type: application/json; charset=utf-8
< Content-Length: 475
< ETag: W/"1db-vOKZynfab6XNYpgAbFksFRr5g5w"
< Date: Mon, 31 Jan 2022 04:58:24 GMT
< Connection: keep-alive
< Keep-Alive: timeout=5
<
{"coord":{"lon":-123.262,"lat":44.5646},"weather":[{"id":804,"main":"Clouds","description":"overcast clouds","icon":"04n"}],"base":"stations","main":{"temp":45.77,"feels_like":43.84,"temp
```

Status: 200 • OK Time: 15 ms Size: 475 B

content-length	475
content-type	application/json; charset=utf-8

Hello

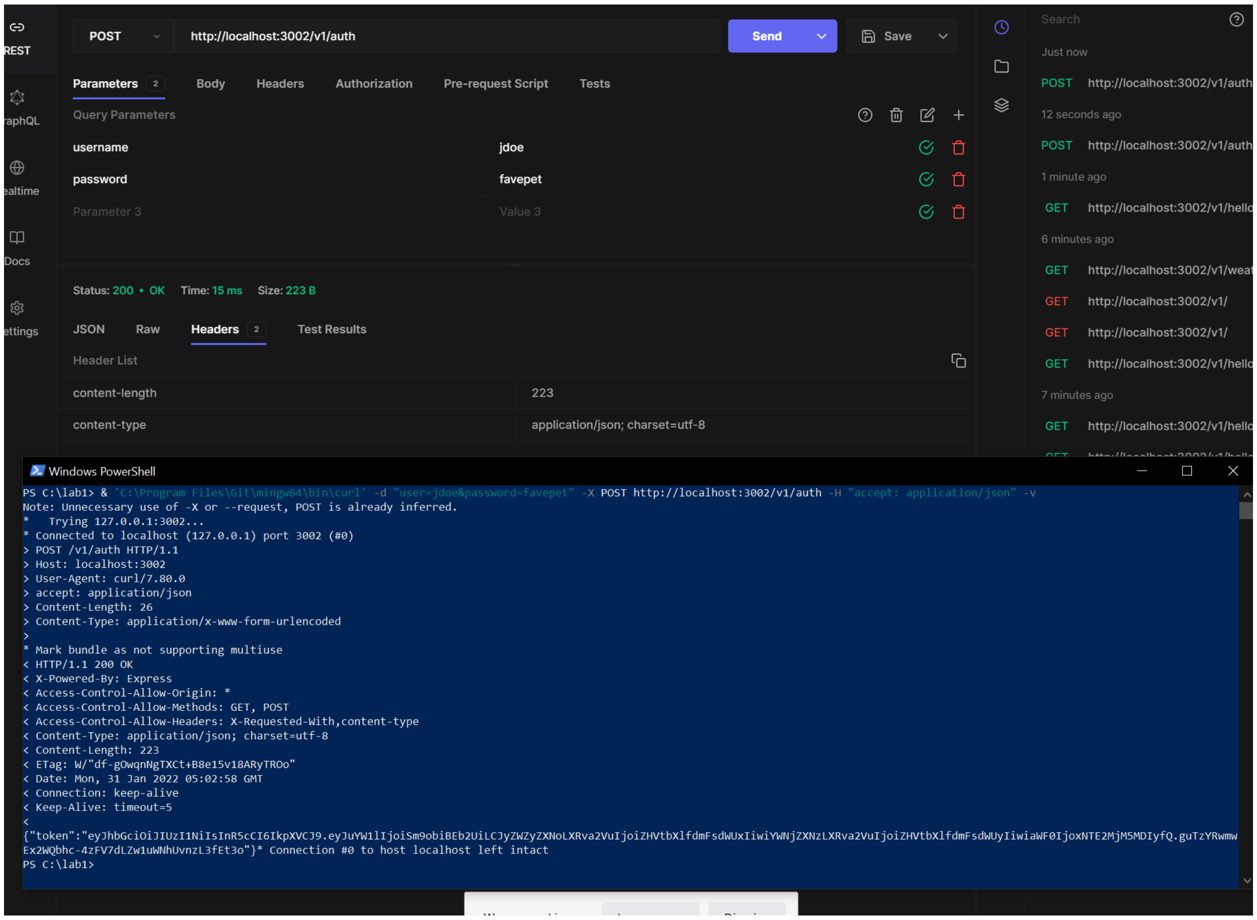


```
Note: Unnecessary use of -X or --request, GET is already inferred.
*   Trying 127.0.0.1:3002...
* Connected to localhost (127.0.0.1) port 3002 (#0)
> GET /v1/hello HTTP/1.1
> Host: localhost:3002
> User-Agent: curl/7.80.0
> accept: application/json
>
* Mark bundle as not supporting multiuse
< HTTP/1.1 200 OK
< X-Powered-By: Express
< Access-Control-Allow-Origin: *
< Access-Control-Allow-Methods: GET, POST
< Access-Control-Allow-Headers: X-Requested-With,content-type
< Content-Type: application/json; charset=utf-8
< Content-Length: 27
< ETag: W/"1b-xzIU1+d8Q6tvYIa1dwTPZMUwlCM"
< Date: Mon, 31 Jan 2022 05:01:31 GMT
< Connection: keep-alive
< Keep-Alive: timeout=5
<
{"greeting":"Hello world!"}* Connection #0 to host localhost left intact
PS C:\lab1>
```

Status: 200 • OK Time: 13 ms Size: 27 B

content-length	27
content-type	application/json; charset=utf-8

Auth



POST http://localhost:3002/v1/auth

username	jdoe
password	favepet
Parameter 3	Value 3

Status: 200 • OK   Time: 15 ms   Size: 223 B

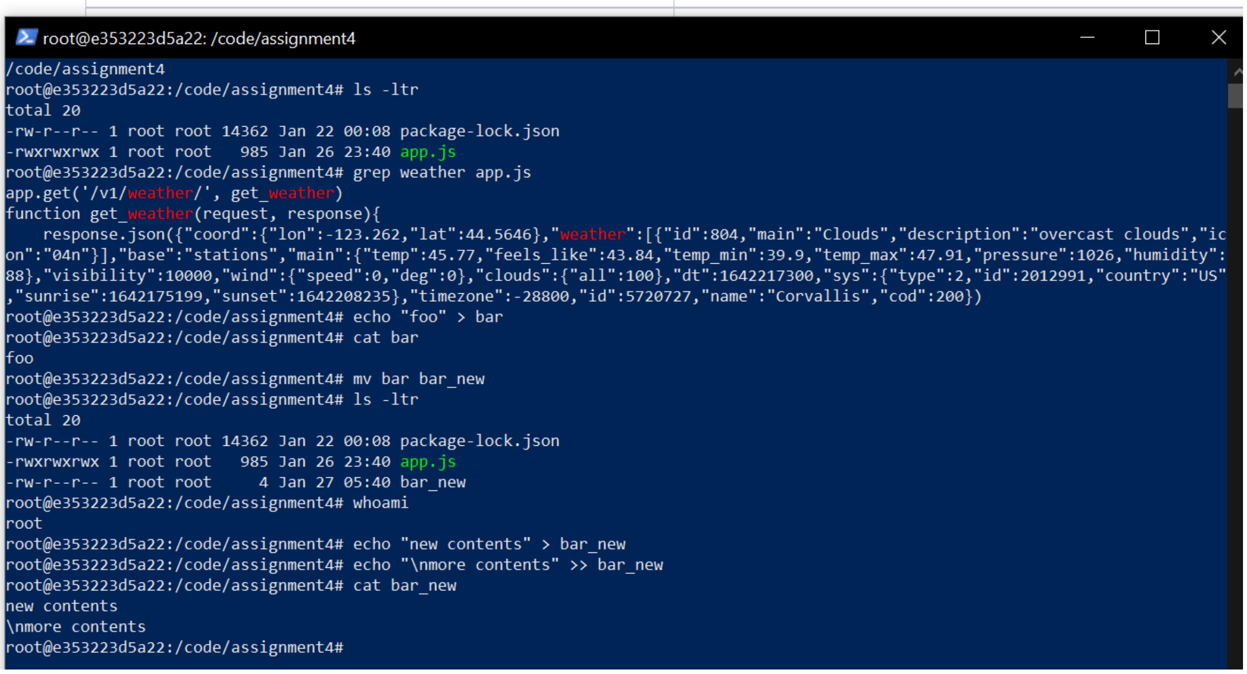
content-length	223
content-type	application/json; charset=utf-8

Just now	
POST	http://localhost:3002/v1/auth
12 seconds ago	
POST	http://localhost:3002/v1/auth
1 minute ago	
GET	http://localhost:3002/v1/hello
6 minutes ago	
GET	http://localhost:3002/v1/wea
GET	http://localhost:3002/v1/
GET	http://localhost:3002/v1/
GET	http://localhost:3002/v1/hello
7 minutes ago	
GET	http://localhost:3002/v1/hello

Windows PowerShell

```
PS C:\lab1> & 'C:\Program Files\Git\mingw64\bin\curl' -d "user=jdoe&password=favepet" -X POST http://localhost:3002/v1/auth -H "accept: application/json" -v
Note: Unnecessary use of -X or --request, POST is already inferred.
*   Trying 127.0.0.1:3002...
* Connected to localhost (127.0.0.1) port 3002 (#0)
> POST /v1/auth HTTP/1.1
> Host: localhost:3002
> User-Agent: curl/7.80.0
> accept: application/json
> Content-Length: 26
> Content-Type: application/x-www-form-urlencoded
>
* Mark bundle as not supporting multiuse
< HTTP/1.1 200 OK
< X-Powered-By: Express
< Access-Control-Allow-Origin: *
< Access-Control-Allow-Methods: GET, POST
< Access-Control-Allow-Headers: X-Requested-With,content-type
< Content-Type: application/json; charset=utf-8
< Content-Length: 223
< ETag: W/"df-gOwqnNgTXCt+B8e15v18ARyTROo"
< Date: Mon, 31 Jan 2022 05:02:58 GMT
< Connection: keep-alive
< Keep-Alive: timeout=5
<
{"token":"eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJuYW1lIjoiSm9obiBEb2UiLCJyZWZyZXNoLXRva2VuIjoiZHVtbXlfdmFsdWUxIiwiYWNjZXNzLXRva2VuIjoiZHVtbXlfdmFsdWUyIiwiaWF0IjoxNTE2MjM5MDIyfQ.guTzYRwmw
Ex2WQbhc-4zFV7dLZw1uWNhUvnzL3fEt3o"}* Connection #0 to host localhost left intact
PS C:\lab1>
```

6. Unix commands practice

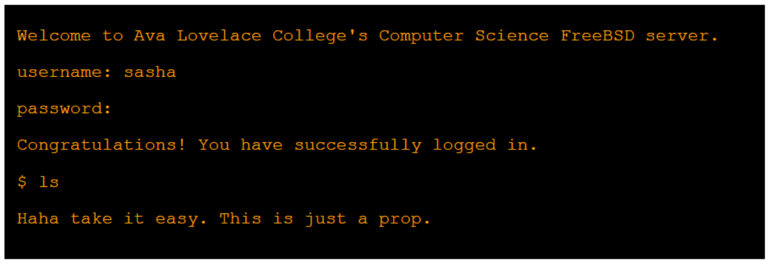


root@e353223d5a22: /code/assignment4

```
/code/assignment4
root@e353223d5a22:/code/assignment4# ls -ltr
total 20
-rw-r--r-- 1 root root 14362 Jan 22 00:08 package-lock.json
-rwxrwxrwx 1 root root   985 Jan 26 23:40 app.js
root@e353223d5a22:/code/assignment4# grep weather app.js
app.get('/v1/weather/', get_weather)
function get_weather(request, response){
    response.json({"coord":{"lon":-123.262,"lat":44.5646},"weather":[{"id":804,"main":"Clouds","description":"overcast clouds","ic
on":"04n"}],"base":"stations","main":{"temp":45.77,"feels_like":43.84,"temp_min":39.9,"temp_max":47.91,"pressure":1026,"humidity":
88},"visibility":10000,"wind":{"speed":0,"deg":0},"clouds":{"all":100},"dt":1642217300,"sys":{"type":2,"id":2012991,"country":"US"
,"sunrise":1642175199,"sunset":1642208235},"timezone":-28800,"id":5720727,"name":"Corvallis","cod":200})
root@e353223d5a22:/code/assignment4# echo "foo" > bar
root@e353223d5a22:/code/assignment4# cat bar
foo
root@e353223d5a22:/code/assignment4# mv bar bar_new
root@e353223d5a22:/code/assignment4# ls -ltr
total 20
-rw-r--r-- 1 root root 14362 Jan 22 00:08 package-lock.json
-rwxrwxrwx 1 root root   985 Jan 26 23:40 app.js
-rw-r--r-- 1 root root     4 Jan 27 05:40 bar_new
root@e353223d5a22:/code/assignment4# whoami
root
root@e353223d5a22:/code/assignment4# echo "new contents" > bar_new
root@e353223d5a22:/code/assignment4# echo "\nmore contents" >> bar_new
root@e353223d5a22:/code/assignment4# cat bar_new
new contents
\nmore contents
root@e353223d5a22:/code/assignment4#
```

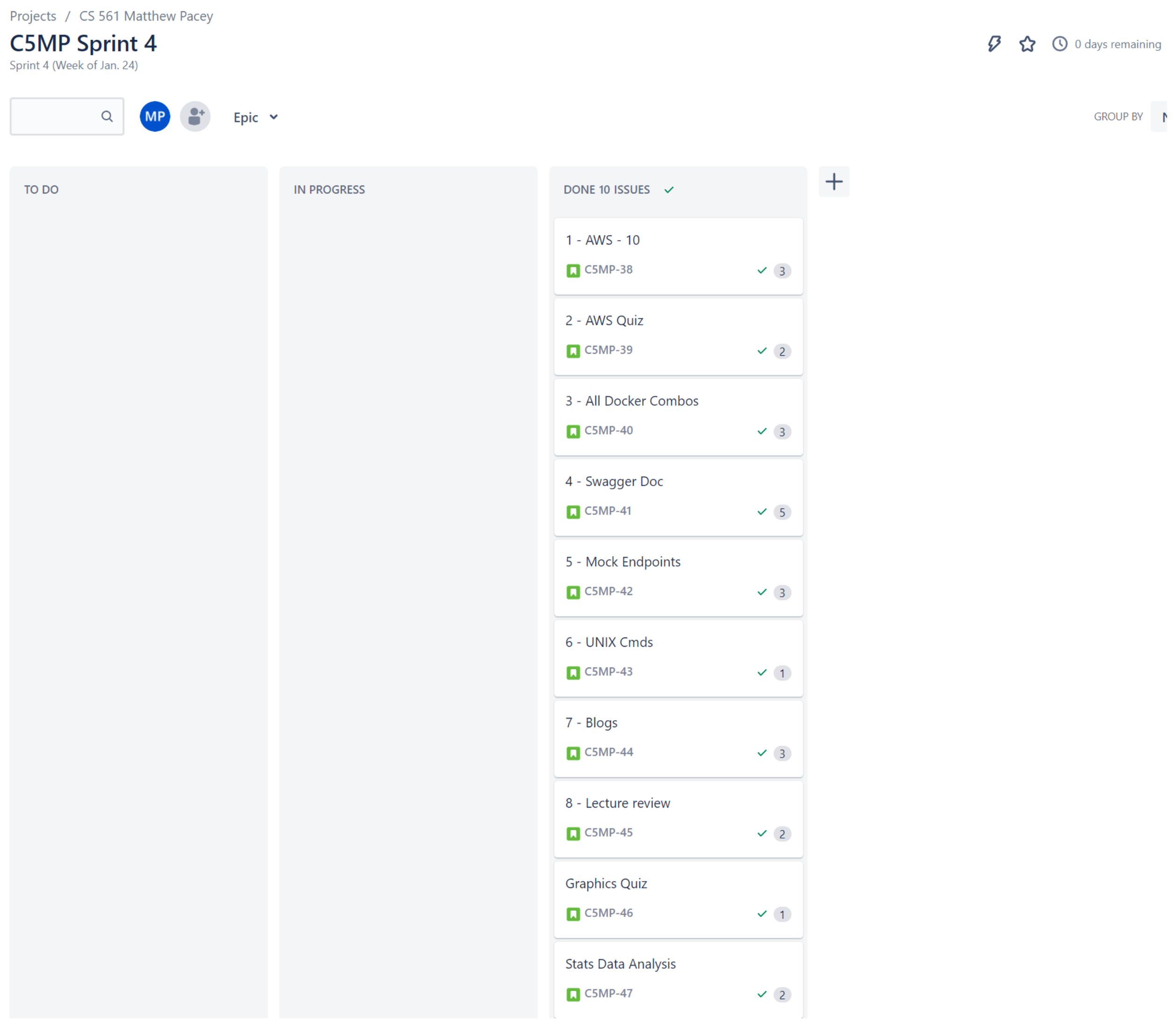
7. Cracking Hack-Along

The orange-and-black terminal window below will probably not show up on a smartphone. (Have not tried on a tablet.) So you might have to use a laptop (e.g. your macbook) or desktop for this one.



```
Welcome to Ava Lovelace College's Computer Science FreeBSD server.
username: sasha
password:
Congratulations! You have successfully logged in.
$ ls
Haha take it easy. This is just a prop.
```

Jira Board



Projects / CS 561 Matthew Pacey

C5MP Sprint 4

Sprint 4 (Week of Jan. 24)

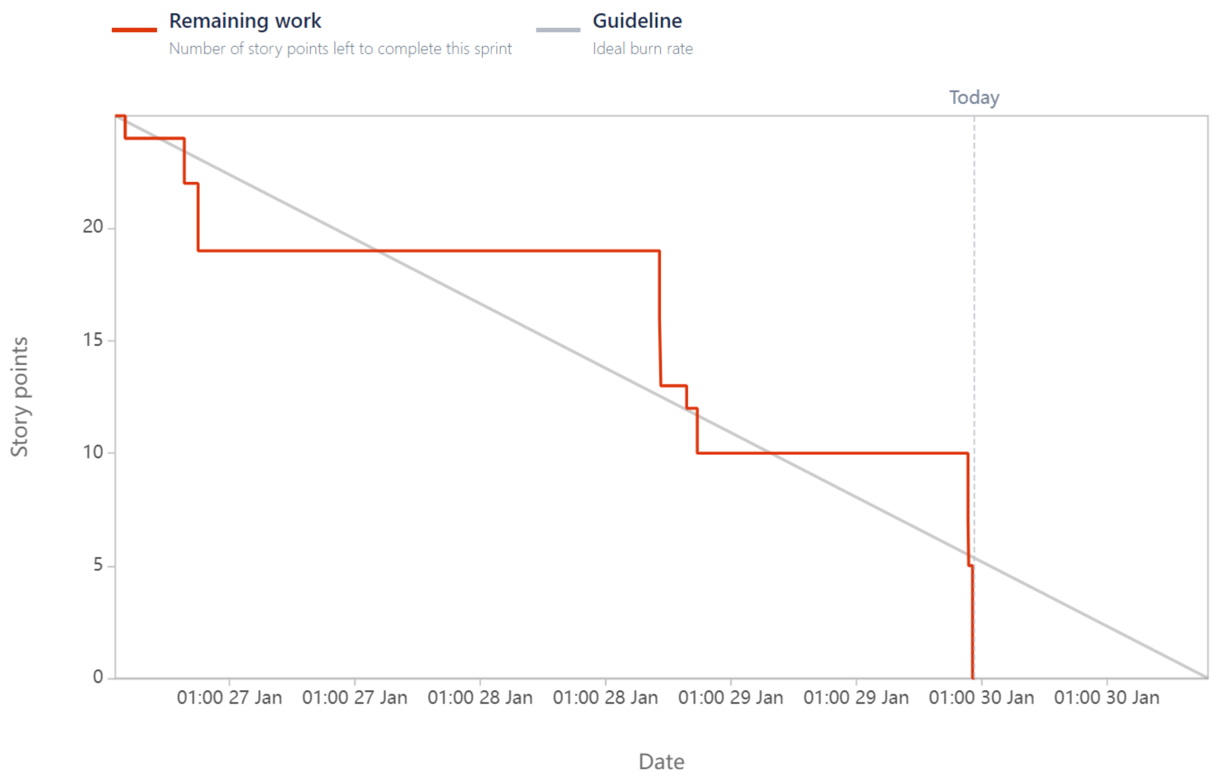
0 days remaining

Issue	Key	Count
1 - AWS - 10	C5MP-38	3
2 - AWS Quiz	C5MP-39	2
3 - All Docker Combos	C5MP-40	3
4 - Swagger Doc	C5MP-41	5
5 - Mock Endpoints	C5MP-42	3
6 - UNIX Cmds	C5MP-43	1
7 - Blogs	C5MP-44	3
8 - Lecture review	C5MP-45	2
Graphics Quiz	C5MP-46	1
Stats Data Analysis	C5MP-47	2

Burndown

Date - January 26, 2022 - January 30, 2022

Sprint goal - Sprint 4 (Week of Jan. 24)


Remaining work
Number of story points left to complete this sprint
Guideline
Ideal burn rate
Today
Story points
01:00 27 Jan
01:00 27 Jan
01:00 28 Jan
01:00 28 Jan
01:00 29 Jan
01:00 29 Jan
01:00 30 Jan
01:00 30 Jan
Date