

# Ancillary Servers

November 5, 2019 / Gavin Jackson

development

inbucket

monitoring

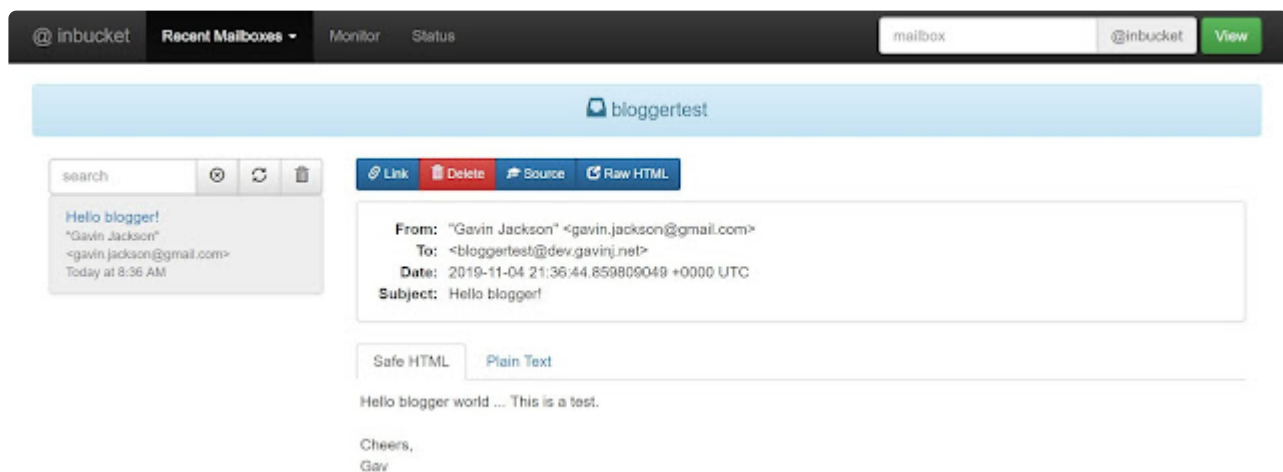
open source

pastebin

This post looks at handy ancillary servers that I find myself using to support my day-to-day software development tasks.

What do I mean by *ancillary server* In this context, it's an internet-connected server that is used to help me in my day to-day work. These typically run on a cheap VPS server (think Linode or Digital Ocean). The utilities are all open source software.

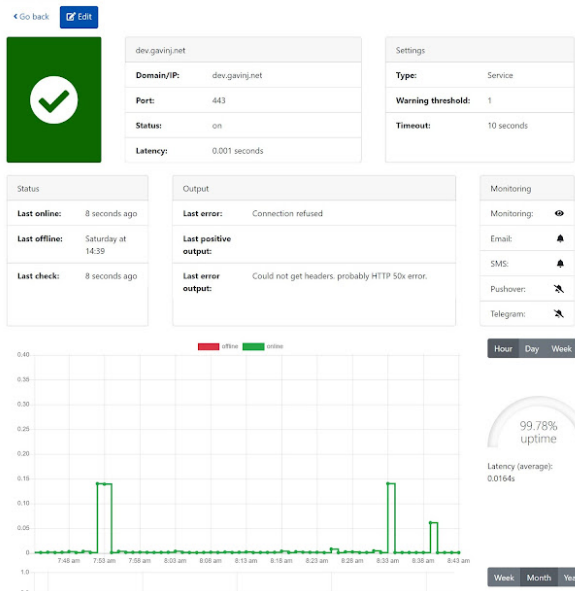
1. **InBucket** - this is a mailinator-like service for testing email services, it's basically an SMTP server listening on port 25 that accepts email for all email addresses and provides a web interface that allows you to check the email - a REST API allows you to write unit tests. I find this handy for a lot of reasons, obviously this is great for testing software (we use an internal instance for testing low environments), but it's also great for creating throw-away emails. <https://www.inbucket.org/>



2. **PHPServerMonitor** - need a simple service to check if your servers are running This is a nice, simple one that does just that. It integrates with a number of notification subsystems (email, SMS gateways etc) you can easily define the frequency of the checks - I have them running every minute and it send me an sms if any of my servers go down. Quick and easy to set up.

<https://www.phpservermonitor.org/>

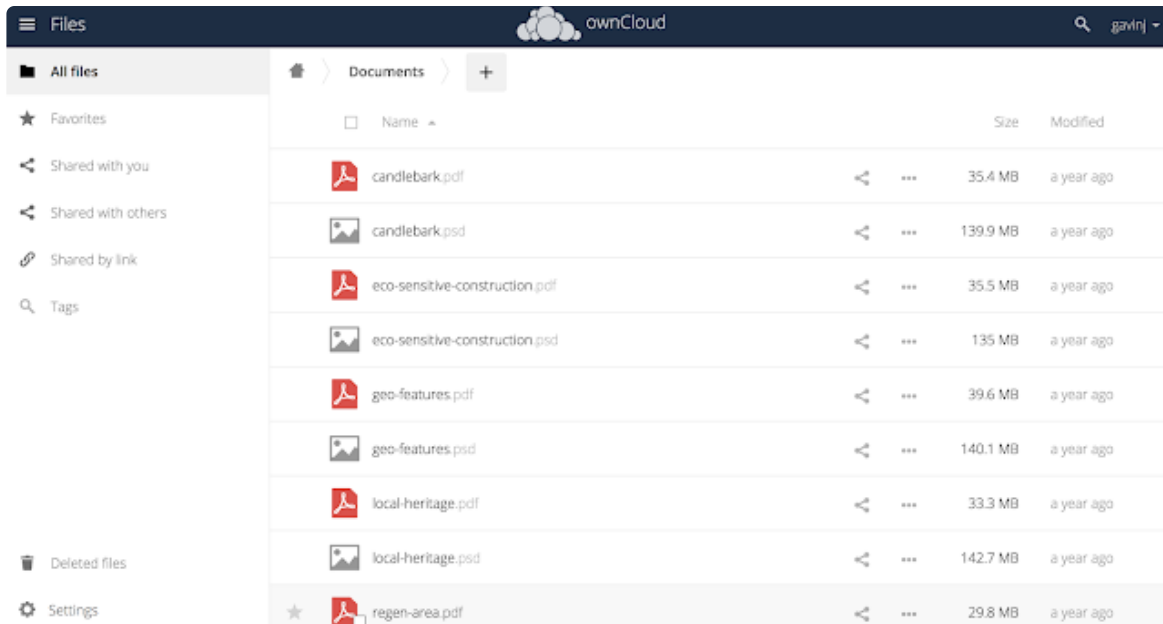
## Servers



3. **Stikked** - a personal pastebin that allows you to easily share snippets of code (or temporarily store them for yourself if you are jumping between networks etc). Simple rules for aging off notes, and very pretty syntax highlighting. <https://github.com/claudehohl/Stikked>

```
1. def parse(self, response):
2.     print('\n {0}'.format(self.properties[response.url.split('/')[1:][0]]))
3.     self.c.execute("insert into eco_property values (NULL, {0}, '{1}', '{2}';".format(self.eid,
self.properties[response.url.split('/')[1:][0]], response.url.split('/')[1:][0]))
4.     pid = self.c.lastrowid
5.     self.conn.commit()
6.     ws = self.ws.create_sheet(title="{0}".format(self.properties[response.url.split('/')[1:][0]))
7.     print("\n *#*#)
8.     attributes = {}
9.     rows = response.xpath("//[@id='ipage']/div[4]/table/tr")
10.    for index, row in enumerate(rows):
11.        if index > 0:
12.            print('== {0} =='.format(row.xpath('td[1]/text()').extract()[0]))
13.            self.c.execute("insert into eco_month values (NULL, {0}, {1}, '{2}';".format(self.eid, pid,
row.xpath('td[1]/text()').extract()[0]))
14.            mid = self.c.lastrowid
15.            self.conn.commit()
16.            ws.append([row.xpath('td[1]/text()').extract()[0]])
17.            print('AVAILABLE {0}'.format(row.css('.available').xpath('@title').extract()))
18.            for str_date in row.css('.available').xpath('@title').extract():
19.                from datetime import datetime
20.                date_object = datetime.strptime(str_date, '%a %d-%b-%Y')
21.                self.c.execute("insert into eco_day values (NULL, {0}, 'AVAILABLE', '{1}',
'{2}';".format(mid, str_date.split(' ')[0], date_object.strftime('%Y-%m-%d')))
22.                self.conn.commit()
23.                ws.append(['AVAILABLE'] + row.css('.available').xpath('@title').extract())
24.            print('BOOKED {0}'.format(row.css('.booked').xpath('@title').extract()))
25.            for str_date in row.css('.booked').xpath('@title').extract():
26.                from datetime import datetime
27.                date_object = datetime.strptime(str_date, '%a %d-%b-%Y')
28.                self.c.execute("insert into eco_day values (NULL, {0}, 'BOOKED', '{1}', '{2}';".format(mid,
str_date.split(' ')[0], date_object.strftime('%Y-%m-%d')))
29.                self.conn.commit()
30.                ws.append(['BOOKED'] + row.css('.booked').xpath('@title').extract())
31.            ws.append([' '])
32.        def closed(self, reason):
33.            self.ws.save(filename = "./output.xlsx")
34.            self.conn.commit()
35.            self.c.close()
```

4. **Owncloud** - ever wanted your own personal dropbox server Owncloud seems to fit the bill nicely. Written in PHP it's easy to set up, and also provides mobile apps that can connect to your own server! <https://github.com/owncloud>, <https://owncloud.org/>



What are your favorite ancillary servers Please leave a comment below!

Downloaded from <https://www.gavinj.net/post/ancillary-servers>

Generated July 9, 2026. Copyright Gavin Jackson. All rights reserved.