

# **# Technical Writing Samples**

**##### Katherine Tothill**

**##### January 25, 2021**

## **# Table of Contents**

\* [General Introduction] (#general-introduction)

\* [Internal] (#internal)

\* [Bash Script] (#bash)

\* [How To Article] (#how-to)

\* [Incident Jira Ticket] (#jira)

\* [Incident Communication] (#incident)

\* [Case Studies] (#case-studies)

\* [Managed DNS] (#managed)

\* [Email Delivery] (#email)

\* [External] (#external)

\* [API Help Guide] (#api)

\* [Sub-Delegation Help Guide] (#sub)

\* [Status Page] (#status)

\* [Conclusion] (#conclusion)

## **##### <a name="general-introduction"></a>General Introduction**

The following writing samples comprise my professional technical writing portfolio. These samples are designed to demonstrate my ability to write internal and external technical documentation. The internal examples are intended for Domain Name System (DNS) engineers and support analysts. The external examples are intended for customers that have DNS service with the fictional company, "OurDNS". This document does not define

23 best practices for any type of documentation.

24

25

26 **### <a name="internal"></a>Internal**

27 **#### <a name="bash"></a>Bash Script**

28 *\*Location: Support Playbook, under Account Reports\**

29

30 **##### Introduction**

31 These Bash scripts generate DNS query reports for

31 a single account or a list of accounts. These

31 scripts create and aggregate daily query reports

31 to produce a single report for a given date range.

31 These reports are used by Account Managers and

31 should not be provided to customers.

32

33 **##### Generating Account Query Reports**

34 The following account data is required to generate  
34 reports:

35 | Parameter | Example Value

35 |

36 |-----|-----

36 |-----|

37 | Account ID | 12345

37 |

38 | Date Range | 01/01/2020-12/31/2020

38 |

39 | Monthly Query Limit | 1,000,000

39 |

40 | Network | Managed DNS (represented

40 here as network 0) |

41

42 **##### Single Account**

43 Use the following scripts when generating a query  
43 report for a single account:

44

45 **#### Daily Reports**

46 ``query-logs daily-report --account 12345 --start`  
46 `01/01/2020 --end 12/31/2020 DAILY_REPORT_12345``

47

48 **#### Aggregated Report**

49 ``query-logs growth-report --daily-report`  
49 `"DAILY_REPORT_OUTPUT_12345" --query-limit`  
49 `"1,000,000" --networks 0`  
49 `"GROWTH_REPORT_OUTPUT_12345"``

50

51 **#### Account List**

52 Use the following script when generating query  
52 reports for a list of accounts, as each account ID  
52 and query limit are unique. In this example, the  
52 account IDs are listed in a file called  
52 `account_ids.txt`.

53

54 **#### Daily Reports**

55 `for aid in $(cat account_ids.txt); do query-logs`  
55 `daily-report --account ${aid} --start 01/01/2020`  
55 `--end 12/31/2020 DAILY_REPORT_${aid}; done`

56

57 **#### Aggregated Report**

58 ```

59 `declare -a aid_queries=(`  
60 `"8156,46000000000"`  
61 `"9742,35000000000"`  
62 `"3332,60000000000"`  
63 `"1401,5000000000"`

```
64 "8724,15000000000"
65 "6014,15000000000"
66 "12530,30000000000"
67 )
68
69 for aid_query in ${aid_queries[@]}; do aid=$(awk
69 -F',' '{print $1}' <<< ${aid_query}); query=$(awk
69 -F',' '{print $2}' <<< ${aid_query}); query-logs
69 growth-report --daily-report
69 "DAILY_REPORT_OUTPUT_${aid}" --query-limit
69 "${query}" --networks 0
69 "GROWTH_REPORT_OUTPUT_${aid}"; done`
70 ``
71
72
73 ##### <a name="how-to"></a>How-To Article
74 *Location: Engineering Playbook, under
74 Troubleshooting*
75
76 ##### Running Traceroutes
77 ##### Introduction
78 This article describes how to run traceroutes from
78 an OurDNS Point of Presence (POP) to a given host.
78 Traceroutes are often requested by support
78 analysts, as only Engineering has access to OurDNS
78 POPs.
79
80 To run a traceroute, do the following:
81
82 1. SSH into the POP you want to connect to,
82 represented here as hostname `<pop>.ourdns.com`:
83     `username@host:~$ SSH <pop>.ourdns.com`
```

---

```
84
85 > **Note:**
86 > For assistance configuring SSH access, see the
86 ["*Getting Started*"](#getting-started) section of
86 the Engineering Playbook.
87
88 2. Run the traceroute command, where `<hostname>`
88 defines the name of the host that our POP will
88 attempt to connect to:
89 ```
90     username@<pop>:~$ traceroute <hostname>
91     traceroute to <hostname> (23.227.38.65),
91 64 hops max, 52 byte packets
92
93         1  192.168.200.1 (192.168.200.1)  1.570 ms
93 1.378 ms  1.835 ms
94         2  96.120.70.57 (96.120.70.57)   9.733 ms
94 10.056 ms  9.529 ms
95         3  162.151.171.65 (162.151.171.65) 10.584
95 ms 10.306 ms  9.698 ms
96         4
96 be-315-ar01.needham.ma.boston.comcast.net
96 (96.108.46.117) 13.884 ms 11.823 ms 14.000 ms
97         5  76.96.121.242 (76.96.121.242) 17.615
97 ms 14.208 ms *
98         6  <hostname> (23.227.38.65) 13.962 ms
98 14.765 ms 25.479 ms
99 ```
100 3. Log out of the POP by typing `exit` and
100 pressing `enter`.
101     `username@<pop>:~$ exit`
102
```

103 4. Provide traceroute output to support.

104

105

106 ##### <a name="jira"></a>Incident Jira Ticket

107 *\*Location: Engineering Playbook, under Incident*

107 *Response\**

108 ##### Introduction

109 All incidents on our platform require the creation  
109 of a Jira ticket. Incident impact can be internal  
109 (such as failed Jira logins) or external (such as  
109 failed account logins). An incident's Jira ticket  
109 allows users across teams to review the specifics  
109 of the incident at any time.

110

111 Additionally, Support uses an incident's Jira  
111 ticket to generate an Incident Communication (IC,  
111 see [*\*Incident Communication\**] (#incident)). The  
111 following is an example of an incident Jira ticket:

112

113 **\*\*Incident Start Time:\*\*** *\*07/24/2019 07:09 UTC\**

114 **\*\*Incident Resolved:\*\*** *\*07/31/2019 15:24 UTC\**

114 (Duration: 176:15:00)

115 **\*\*Incident Severity:\*\*** Customer Impacting

116 **\*\*Attack Vector:\*\*** Not Applicable; internal  
116 infrastructure instability

117

118 **\*\*Incident Description:\*\*** Increased traffic caused  
118 our aging data pipeline to become unable to handle  
118 this increase. This caused the statistics  
118 subsystem to become unstable. Resultingly,  
118 customer statistical data was unavailable.

119

120 **\*\*How did we mitigate the incident?\*\***

121 Ops disabled the impacted statistics endpoints by  
121 connecting to the following POPs:

122 \* AMS

123 \* BOS

124 \* JFK

125 \* LAX

126

127 Billing stats were then paused from  
127 OurStatsProgram, while we addressed supporting the  
127 current pipeline.

128

129 **\*\*Root Cause:\*\*** Data pipeline unable to handle  
129 increased traffic, as it has not been scaled to  
129 meet increased demand.

130

131 **\*\*How do we prevent this from occurring in the  
131 future?\*\***

132 We are in the process of building out a new  
132 pipeline and attempting to increase available  
132 resources. We have also increased the alert  
132 sensitivity for relevant POPs to allow immediate  
132 detection of anomalies at least until the new  
132 pipeline is built.

133

134 To date, we have experienced three incidents as a  
134 result of the current pipeline's less performant  
134 infrastructure. Discussions are ongoing within the  
134 #ops channel of our messaging system regarding the  
134 progress of the new pipeline.

135

136

137 **#### <a name="incident"></a>Incident Communication**  
138 *\*Location: Engineering Playbook, under Incident*  
138 *Response\**  
139 **#### Introduction**  
140 An Incident Communication (IC) is an executive  
140 summary of an incident that Support writes for  
140 Senior Management. Support updates the IC  
140 throughout the course of the incident, detailing  
140 the steps engineers have taken to identify,  
140 mitigate, and resolve the issue. ICs are written  
140 for both internal (*\*e.g.,\* failed Jira logins*) and  
140 external (*\*e.g.,\* failed account logins*)  
140 incidents. The actual document is strictly  
140 internal and should never be provided to  
140 customers. Although an IC is more general than the  
140 incident's Jira ticket, the Jira can be used to  
140 quickly gather the pertinent details. The  
140 following example details an IC:  
141  
142 **\*\*INC-123 Statistics Subsystem Availability\*\***  
143 **\*\*Severity:\*\*** Customer Impacting  
144 **\*\*Product:\*\*** Statistics Subsystem  
145 **\*\*Customers Reported:\*\***  
146 *\* Acme, Acme, and Acme Law Firm*  
147 *\* Acme Advertising*  
148 *\* Acme Security*  
149  
150 **\*\*Description:\*\*** We experienced an issue with our  
150 statistics subsystem. Portal and API statistics,  
150 and zone- and record-level query statistics were  
150 unavailable for all customers. Account-level data,  
150 including network-level statistics, continued to



propagate via the Portal and API. There was no impact to DNS resolution.

**\*\*Incident Start Time:\*\*** \*07/24/2019 7:09 UTC\*

**\*\*Incident Resolved:\*\*** \*07/31/2019 15:24 UTC\*

(Duration: 176:15:00)

**\*\*Customer Impact:\*\*** All customers experienced unavailability of statistics and billing data.

**\*\*Resolution:\*\*** Engineers disabled relevant statistical endpoints to mitigate impact while they deployed a fix which recovered the statistics subsystem for our Portal and API.

Engineers worked in parallel to do the following:

- \* Restore statistics availability on the existing data pipeline.
- \* Build a new data pipeline that will stabilize delivery of query metrics to the Portal and API.

\* **\*\*Note:\*\*** No specific timeline exists for the creation of the new pipeline; monitor the #ops channel in our messaging system for the latest updates.

**\*\*Jira Ticket:\*\***

<https://ourdns.atlassian.net/browse/INC-123>

**\*\*External Status Page:\*\***

<https://ourdns.com/#!/incident/123>

**#### <a name="case-studies"></a>Case Studies**

**##### Introduction**

The following case studies detail practical applications of these topics:

168 \* Situation

169 \* Task

170 \* Action

171 \* Result

172

173 ##### <a name="managed"></a>Managed DNS

174 ##### Situation

175 Customer experienced resolution delays of as much  
175 as 15-20 minutes when provisioning a new hostname  
175 via the API. Queries were prevented from receiving  
175 a correct response until the minimum Time to Live  
175 (TTL) elapsed. Until TTL expiration, the name  
175 server returned a Non-Existent Domain (NXDOMAIN)  
175 response.

176

177 ##### Task

178 Identify and mitigate customer's resolution delays  
178 and NXDOMAIN responses when provisioning a new  
178 hostname via the API.

179

180 ##### Action

181 Review of sample publish call confirmed that the  
181 change had propagated to the edge and was  
181 available. Review of the customer's provided code  
181 identified an incomplete task status value. The  
181 task status value identifies what phase the  
181 publish task is in. Once in a completed status,  
181 the hostname becomes live on the edge.

182

183 ##### Result

184 Once the task status value was in a completed  
184 status, propagation was immediate.

185

186 ##### &lt;a name="email"&gt;&lt;/a&gt;Email Delivery

187 ##### Situation

188 Customer received the following Simple  
188 Authentication Security Layer (SASL) error message  
188 while configuring a new server to send through the  
188 OurDNS email delivery system via Postfix: "SASL  
188 authentication failed". SASL provides mechanisms  
188 for authentication, data integrity-checking, and  
188 encryption for application development. This error  
188 message indicated a failed authentication attempt  
188 when connecting to the OurDNS email server to send  
188 via Postfix.

189

190 ##### Task

191 Identify and resolve authentication error message,  
191 so customer can connect to the OurDNS email server  
191 and send via Postfix.

192

193 ##### Action

194 Reviewed customer configuration for both servers.  
194 Tested locally and determined that the "SASL  
194 authentication failed" error message was in  
194 response to the `libplain.so` or `liblogin.so`  
194 modules not installed in the `/usr/lib/sasl2`  
194 directory.

195

196 Locally tested a rehash of the SASL password for  
196 two servers. Determined that SASL passwords are  
196 required to be rehashed for each server and cannot  
196 be copied between them.

197

## 198 ##### Result

199 Customer rehashed SASL and updated libraries for  
199 their OS and sent via Postfix successfully.

200

201

202 ### <a name="external"></a>External

203 ##### <a name="api"></a>API Help Guide

204 \*Location: <https://help.ourdns.com/API>\*\*

205 ##### Introduction

206 The following instructions describe how to review  
206 your zone details with a single API call.  
206 Previously, retrieving complete zone details via  
206 the OurDNS API required multiple calls. Should you  
206 still wish to utilize multiple API calls to gather  
206 zone details, see our ["\*(Legacy) Get Zone  
206 Details\*"](#legacy-zone) Help Guide.

207

208 This table details the parameters to use when  
208 configuring the API call:

209

210 | Required Parameters |

210

210 | |

211 |-----|-----

211 |-----|-----

211 |-----|

212 | key | Your API key. **\*\*Note:\*\***

212 To view an existing key or generate a new key, see  
212 our [\*API Keys\*](#api-keys) Help Guide. |

213 | zone | Name of the zone you are  
213 retrieving hosts and records for.

213 | |

```
214 | Optional Parameters |
214
214 |
215 | FQDN | Fully Qualified Domain
215 Name. If specified, the records returned are
215 limited to records found at or beneath this host. |
216
217 Use the `GET` API call and endpoint below to view
217 the details for any zone and its corresponding
217 records within your account:
218
219 ` $ curl -X GET -H "APIKEY:$key"
219 `https://api.ourdns.com/accounts/details/zone/FQDN
219 ` `
220
221 Sample Response:
222 ```
223 {
224   "id": "310422af9f792d37dffb528b",
225   "hostmaster": "hostmaster@example.com",
226   "ttl": 3600,
227   "nx_ttl": 3600,
228   "retry": 7200,
229   "zone": "example.com",
230   "refresh": 43200,
231   "expiry": 1209600,
232   "dns_servers": [
233     "ns1.p01.ourdns.com",
234     "ns2.p01.ourdns.com",
235     "ns3.p01.ourdns.com",
236     "ns4.p01.ourdns.com"
237   ],
```

```
238     "networks": [0],
239     "network_pools": ["p01"],
240     "primary": {
241         "enabled": false,
242         "secondaries": []
243     },
244     "records": [
245         {
246             "id": "310022af9f782d37dfffb1790",
247             "type": "NS",
248
249             "ttl": 3600,
250     "short_records": [
251         "ns1.p01.ourdns.com",
252         "ns2.p01.ourdns.com",
253         "ns3.p01.ourdns.com",
254         "ns4.p01.ourdns.com"
255     ],
256         "domain": "example.com"
257     },
258     {
259         "id": "310512509f782d58bb1df419",
260         "type": "A",
261         "ttl": 3600,
262         "short_records": [
263             "1.2.3.4"
264         ],
265         "domain": "www.example.com"
266     }
267 ],
268     "meta": {}
269 {
```

270     ` `` `

271

272

273     **#### <a name="sub"></a>Sub-Delegation Help Guide**

274     *\*Location: <https://help.ourdns.com/advanced-dns>\**

275     **##### Introduction**

276     This OurDNS Help Guide details the steps to

276     separate zone ownership between a parent zone and

276     child zone, often referred to as sub-delegation. A

276     common use case for sub-delegation would be to

276     delegate responsibility for a segment of the DNS

276     name to a subset of users or another DNS provider.

276     For example, the domain ``ourdns.com`` includes

276     ``help.ourdns.com`` as a child zone whose management

276     is restricted to the team managing this Help

276     Guide. A child zone is separate from the parent,

276     but is technically a subdomain of the parent zone.

277

278     For technical specifications, please see [*\*RFC*

278     1034 Domain Concepts and

278     Facilities\*]{#<https://tools.ietf.org/html/rfc1034>)

279

280     **##### Creating a Child Zone Within Our Platform**

281     When creating a child zone within our platform,

281     create an NS record for the child zone within the

281     parent zone. This will assign both the parent and

281     child zones to our platform.

282

283     For assistance updating permissions regarding the

283     child zone, see our [*\*Managing User and Team*

283     Permissions Help Guide\*](#perms).

284

285 For assistance configuring DNSSEC with parent and  
285 child zones, see our [[\\*Enabling DNSSEC On A](#)  
285 [Sub-Delegation Help Guide\\*](#)] ([#dnssec-sub](#)).

286

## 287 ##### Creating a Child Zone With Another DNS 287 Provider

288 When creating a child zone with another DNS  
288 provider, begin by adding the following NS records  
288 to the child zone on *\*our\** platform; this assigns  
288 the child zone to the other provider:

289

290	Zone	Host	Record Type	Records
291	:---	:---	:-----	:-----
292	example.com	child.example.com	NS	
292	ns1.p01.otherdns.com			
293	ns2.p01.otherdns.com			
294	ns3.p01.otherdns.com			
295	ns4.p01.otherdns.com			

296

297 Once the above records have been added to the  
297 child zone on our platform, create  
297 ``child.example.com`` at your other DNS provider as  
297 a parent zone with your desired records as usual.

298

299 After sub-delegation is configured successfully,  
299 requests for ``example.com`` and its associated  
299 hosts would still be answered by us; requests for  
299 ``child.example.com`` and its associated hosts would  
299 be answered by the name servers of your other DNS  
299 provider.

300

301 An example request for hosts at



```
301 `child.example.com` would be answered as follows:
302 `Root Name Servers -> .com Name Servers ->
302 example.com Name Servers (Us) -> child.example.com
302 Name Servers (Other DNS Provider) -> Requested
302 Host(s) and Record(s)`
```

```
303
```

#### 304 ##### Testing Sub-Delegation

```
305
```

```
306 Should you wish to test this configuration, we
306 recommend using `dig` as described in our [*Using
306 Dig to Confirm DNS Changes*](#dig) Help Guide. A
306 example of a successful `dig` is shown below:
```

```
307 ```
```

```
308 username@host:~$ dig child.example.com +trace
```

```
309
```

```
310 ; <<>> DiG 9.10.6 <<>> child.example.com +trace
```

```
311 ;; global options: +cmd
```

```
312 .                85384      IN      NS     a.root-servers.net.
```

```
313 .                85384      IN      NS     b.root-servers.net.
```

```
314 .                85384      IN      NS     c.root-servers.net.
```

```
315 .                85384      IN      NS     d.root-servers.net.
```

```
316 .                85384      IN      NS     e.root-servers.net.
```

```
317 .                85384      IN      NS     f.root-servers.net.
```

```
318 .                85384      IN      NS     g.root-servers.net.
```

```
319 .                85384      IN      NS     h.root-servers.net.
```

```
320 .                85384      IN      NS     i.root-servers.net.
```

```
321 .                85384      IN      NS     j.root-servers.net.
```

```
322 .                85384      IN      NS     k.root-servers.net.
```

```
323 .                85384      IN      NS     l.root-servers.net.
```

```
324 .                85384      IN      NS     m.root-servers.net.
```

```
325 ;; Received 239 bytes from 8.8.8.8#53(8.8.8.8) in
```

```
325 46 ms
```

326

327	com.	172800	IN	NS	e.gtld-servers.net.
328	com.	172800	IN	NS	g.gtld-servers.net.
329	com.	172800	IN	NS	l.gtld-servers.net.
330	com.	172800	IN	NS	b.gtld-servers.net.
331	com.	172800	IN	NS	d.gtld-servers.net.
332	com.	172800	IN	NS	a.gtld-servers.net.
333	com.	172800	IN	NS	i.gtld-servers.net.
334	com.	172800	IN	NS	m.gtld-servers.net.
335	com.	172800	IN	NS	c.gtld-servers.net.
336	com.	172800	IN	NS	k.gtld-servers.net.
337	com.	172800	IN	NS	h.gtld-servers.net.
338	com.	172800	IN	NS	j.gtld-servers.net.
339	com.	172800	IN	NS	f.gtld-servers.net.

340 ;; Received 844 bytes from

340 192.36.148.17#53(i.root-servers.net) in 27 ms

341

342	example.com.	172800	IN	NS	ns1.p01.ourdns.com.
343	example.com.	172800	IN	NS	ns2.p01.ourdns.com.
344	example.com.	172800	IN	NS	ns3.p01.ourdns.com.
345	example.com.	172800	IN	NS	ns4.p01.ourdns.com.

346 ;; Received 296 bytes from

346 192.5.6.30#53(a.gtld-servers.net) in 22 ms

347

348	child.example.com.	172800	IN	NS	
348	ns1.p01.otherdns.com.				
349	child.example.com.	172800	IN	NS	
349	ns2.p01.otherdns.com.				
350	child.example.com.	172800	IN	NS	
350	ns3.p01.otherdns.com.				
351	child.example.com.	172800	IN	NS	
351	ns4.p01.otherdns.com.				

```
352 ;; Received 256 bytes from 198.51.45.8#53
352 (ns2.p01.ourdns.com) in 27 ms
353
354 child.example.com. 3600 IN A 23.227.38.65
355 ;; Received 64 bytes from
355 198.51.43.8#53(ns2.p01.otherdns.com) in 21 ms
356 ` ` `
357
358 > Note:
359 > If you are attempting to delegate an existing
359 child zone to another DNS provider, you may need
359 to contact `support@ourdns.com` to confirm that
359 these changes will not adversely impact DNS
359 resolution. If you are creating a brand-new child
359 zone, no extra steps are required.
360
361
362 #### <a name="status"></a> Status Page
363 *Location: https://status.ourdns.com*
364 ##### Introduction
365 Visit the OurDNS Status Page to follow incidents,
365 past or in progress, as well as any upcoming
365 maintenance. If you have not already done so, you
365 can sign up for email or SMS notifications
365 whenever OurDNS creates or updates an incident.
365 See the example status post below, so that you
365 know what to expect in your inbox or via SMS:
366
367 ##### Portal and API Statistics Performance Impact
368 *07/24/2019 07:09 UTC*
369 We are currently experiencing an issue with our
369 statistics subsystem. Some Portal and API
```

369 statistics will be intermittently unavailable for  
369 some customers. DNS Resolution is unaffected by  
369 this issue and is currently operating as expected.  
369 Customers with further questions are encouraged to  
369 contact ``support@ourdns.com.``

370

371 *\*07/24/2019 10:14 UTC\**

372 Our Engineering team continues to actively  
372 investigate the issue. There remains no impact to  
372 DNS. Customers with further questions are  
372 encouraged to contact ``support@ourdns.com.``

373

374 *\*07/25/2019 17:13 UTC\**

375 Our Engineers are working in parallel tracks to  
375 accelerate resolution of the statistics subsystem  
375 issue as follows:

376 *\* We are building a new data pipeline that will  
376 more permanently stabilize delivery of query  
376 metrics to the API/portal. We have accelerated  
376 transitioning statistics to this new pipeline to  
376 address the ongoing metrics issue.*

377 *\* Engineers are attempting to restore statistics  
377 availability on the existing data pipeline. This  
377 work has been ongoing since this issue first  
377 surfaced.*

378

379 We are continuing to devote significant resources  
379 to these efforts and expect that one of these  
379 workstreams will lead to resolution soon. Updates  
379 will be provided as they are available. Customers  
379 with further questions are encouraged to contact  
379 ``support@ourdns.com.``

380

381 *\*07/26/2019 16:18 UTC\**

382 Our Engineers have deployed a fix to restore  
382 statistics availability for the Portal and API.  
382 Account-level data, including network-level  
382 statistics, continue to propagate via the Portal  
382 and API, however, zone- and record-level query  
382 statistics remain unavailable.

383

384 We are progressing with the creation of a new  
384 statistics subsystem. This subsystem will more  
384 permanently stabilize delivery of query metrics to  
384 the Portal and API and fully restore zone- and  
384 record-level statistics. We are continuing to  
384 address this issue with the utmost priority and  
384 will provide updates as they become available.  
384 Customers with questions are encouraged to contact  
384 ``support@ourdns.com.``

385

386 *\*07/30/2019 19:02 UTC\**

387 Our Engineers have successfully migrated zone- and  
387 record-level statistics to the new data pipeline  
387 and expect statistics to become available via the  
387 Portal and API in the next few hours.

388

389 *\*07/31/2019 0:15 UTC\**

390 Our Engineers have successfully restored zone- and  
390 record-level statistics in the Portal and API.  
390 Please note that historical API data as well as  
390 some search/aggregation features in the Portal are  
390 currently unavailable but will be gradually  
390 restored.

391  
392 We are actively monitoring this deployment.  
392 Customers with questions are encouraged to contact  
392 ``support@ourdns.com.``  
393  
394 **\*\*Incident Resolved\*\***  
395 *\*07/31/2019 15:24 UTC\**  
396 We are pleased to report that our Engineers have  
396 successfully restored zone- and record-level  
396 statistics in the Portal and API. Please note that  
396 historical API data as well as some  
396 search/aggregation features in the Portal are  
396 currently unavailable but will be gradually  
396 restored.  
397  
398 We have monitored this deployment for a few hours  
398 and are now considering this resolved.  
399 Duration: 176:15:00  
400  
401  
402 **#### `<a name="conclusion"></a>`Conclusion**  
403 Thank you for taking the time to review the  
403 provided samples. I hope they provided valuable  
403 insight into some of my skills and abilities. Have  
403 a nice day!  
404  
405  
406 Composed in `<oXygen/>` XML Editor 23.0, build  
406 2020121712