Technical Writing Samples
Katherine Tothill
February 18, 2021

Table of Contents

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 best practices for any type of documentation.

Internal
Bash Script
Location: Support Playbook, under Account Reports

Introduction
These Bash scripts generate DNS query reports for a single account or a list of accounts. These scripts create and aggregate daily query reports to produce a single report for a given date range. These reports are used by Account Managers and should not be provided to customers.

Generating Account Query Reports
The following account data is required to generate reports:

Parameter	Example Value
Account ID	12345
Date Range	01/01/2020-12/31/2020
Monthly Query Limit	1,000,000
Network	Managed DNS (represented here as network 0)

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

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

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

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

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

Aggregated Report
```
declare -a aid_queries=(
"8156,4600000000"
"9742,3500000000"
"3332,60000000000"
"1401,500000000"
"8724,15000000000"
"6014,1500000000"
"12530,3000000000"
)

for aid_query in ${aid_queries[@]}; do aid=$(awk -F',' '{print $1}' <<< ${aid_query}); query=$(awk -F',' '{print $2}' <<< ${aid_query}); query-logs growth-report --daily-report "DAILY_REPORT_OUTPUT_${aid}" --query-limit "${query}" --networks 0 "GROWTH_REPORT_OUTPUT_${aid}"; done`
```

How-To Article
Location: Engineering Playbook, under Troubleshooting

Running Traceroutes
Introduction
This article describes how to run traceroutes from an OurDNS Point of Presence (POP) to a given host. Traceroutes are often requested by

support analysts, as only Engineering has access to OurDNS POPs.

 To run a traceroute, do the following:

1. SSH into the POP you want to connect to, represented here as hostname `<pop>.ourdns.com`:
        `username@host:~$ SSH <pop>.ourdns.com`

> **Note:**
> For assistance configuring SSH access, see the ["*Getting Started*"](#getting-started) section of the Engineering Playbook.

2. Run the traceroute command, where `<hostname>` defines the name of the host that our POP will attempt to connect to:
```
        username@<pop>:~$ traceroute <hostname>
        traceroute to <hostname> (23.227.38.65), 64 hops max, 52 byte packets

        1  192.168.200.1 (192.168.200.1)  1.570 ms  1.378 ms  1.835 ms
        2  96.120.70.57 (96.120.70.57)  9.733 ms  10.056 ms  9.529 ms
        3  162.151.171.65 (162.151.171.65)  10.584 ms  10.306 ms  9.698 ms
        4  be-315-ar01.needham.ma.boston.comcast.net (96.108.46.117)  13.884 ms  11.823 ms  14.000 ms
        5  76.96.121.242 (76.96.121.242)  17.615 ms  14.208 ms *
        6  <hostname> (23.227.38.65)  13.962 ms  14.765 ms  25.479 ms
```
3. Log out of the POP by typing `exit` and pressing `enter`.
        `username@<pop>:~$ exit`

4. Provide traceroute output to support.

Incident Jira Ticket
Location: Engineering Playbook, under Incident Response
Introduction
All incidents on our platform require the creation of a Jira ticket. Incident impact can be internal (such as failed Jira logins) or external (such as failed account logins). An incident’s Jira ticket allows users across teams to review the specifics of the incident at any time.

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

Incident Start Time: *07/24/2019 07:09 UTC*
Incident Resolved: *07/31/2019 15:24 UTC* (Duration: 176:15:00)
Incident Severity: Customer Impacting

Attack Vector: Not Applicable; internal infrastructure instability

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

How did we mitigate the incident?
Ops disabled the impacted statistics endpoints by connecting to the following POPs:
* AMS
* BOS
* JFK
* LAX

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

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

How do we prevent this from occurring in the future?
We are in the process of building out a new pipeline and attempting to increase available resources. We have also increased the alert sensitivity for relevant POPs to allow immediate detection of anomalies at least until the new pipeline is built.

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

Incident Communication
Location: Engineering Playbook, under Incident Response
Introduction
An Incident Communication (IC) is an executive summary of an incident that Support writes for Senior Management. Support updates the IC throughout the course of the incident, detailing the steps engineers have taken to identify, mitigate, and resolve the issue. ICs are written for both internal (*e.g.,* failed Jira logins) and external (*e.g.,* failed account logins) incidents. The actual document is strictly internal and should never be provided to customers. Although an IC is more general than the incident’s Jira ticket, the Jira can be used to quickly gather the pertinent details. The following example details an IC:

INC-123 Statistics Subsystem Availability
Severity: Customer Impacting
Product: Statistics Subsystem

Customers Reported:
* Acme, Acme, and Acme Law Firm
* Acme Advertising
* Acme Security

Description: We experienced an issue with our statistics subsystem. Portal and API statistics, and zone- and record-level query statistics were unavailable for all customers. Account-level data, 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 timeine 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

Case Studies
Introduction
The following case studies detail practical applications of these topics:
* Situation
* Task
* Action
* Result

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

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

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

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

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

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

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

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

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

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

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

Required Parameters	
key	Your API key. **Note:** To view an existing key or generate a new key, see our [*API Keys*](#api-keys) Help Guide.
zone	Name of the zone you are retrieving hosts and records for.
Optional Parameters	
FQDN	Fully Qualified Domain Name. If specified, the records returned are limited to records found at or beneath this host.

Use the `GET` API call and endpoint below to view the details for any zone and its corresponding records within your account:

`$ curl -X GET -H "APIKEY:$key" ‘https://api.ourdns.com/accounts/details/zone/FQDN’`

Sample Response:


```
{
  "id": "310422af9f792d37dffb528b",
  "hostmaster": "hostmaster@example.com",
  "ttl": 3600,
  "nx_ttl": 3600,
  "retry": 7200,
  "zone": "example.com",
  "refresh": 43200,
  "expiry": 1209600,
  "dns_servers": [
    "ns1.p01.ourdns.com",
    "ns2.p01.ourdns.com",
    "ns3.p01.ourdns.com",
    "ns4.p01.ourdns.com"
  ],
  "networks": [0],
  "network_pools": ["p01"],
  "primary": {
    "enabled": false,
    "secondaries": []
  },
```

```
  "records": [
    {
      "id": "310022af9f782d37dffb1790",
      "type": "NS",

      "ttl": 3600,
  "short_records": [
    "ns1.p01.ourdns.com",
    "ns2.p01.ourdns.com",
    "ns3.p01.ourdns.com",
    "ns4.p01.ourdns.com"
      ],
      "domain": "example.com"
    },
    {
      "id": "310512509f782d58bb1df419",
      "type": "A",
      "ttl": 3600,
      "short_records": [
        "1.2.3.4"
      ],
      "domain": "www.example.com"
    }
  ],
  "meta": {}
{
```

Sub-Delegation Help Guide
Location: https://help.ourdns.com/advanced-dns
Introduction
This OurDNS Help Guide details the steps to separate zone ownership between a parent zone and child zone, often referred to as sub-delegation. A common use case for sub-delegation would be to delegate responsibility for a segment of the DNS name to a subset of users or another DNS provider. For example, the domain `ourdns.com` includes `help.ourdns.com` as a child zone whose management is restricted to the team managing this Help Guide. A child zone is separate from the parent, but is technically a subdomain of the parent zone.

For technical specifications, please see [*RFC 1034 Domain Concepts and Facilities*]{#https://tools.ietf.org/html/rfc1034)

Creating a Child Zone Within Our Platform
When creating a child zone within our platform, create an NS record for the child zone within the parent zone. This will assign both the parent and child zones to our platform.

For assistance updating permissions regarding the child zone, see our

[*Managing User and Team Permissions Help Guide*](#perms).

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

Creating a Child Zone With Another DNS Provider
When creating a child zone with another DNS provider, begin by adding the following NS records to the child zone on *our* platform; this assigns the child zone to the other provider:

Zone	Host	Record Type	Records
example.com	child.example.com	NS	ns1.p01.otherdns.com
			ns2.p01.otherdns.com
			ns3.p01.otherdns.com
			ns4.p01.otherdns.com

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

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

An example request for hosts at `child.example.com` would be answered as follows:
`Root Name Servers -> .com Name Servers -> example.com Name Servers (Us) -> child.example.com Name Servers (Other DNS Provider) -> Requested Host(s) and Record(s)`

Testing Sub-Delegation

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

; <<>> DiG 9.10.6 <<>> child.example.com +trace
;; global options: +cmd
.                       85384   IN      NS      a.root-servers.net.
.                       85384   IN      NS      b.root-servers.net.
.                       85384   IN      NS      c.root-servers.net.
.                       85384   IN      NS      d.root-servers.net.
.                       85384   IN      NS      e.root-servers.net.
.                       85384   IN      NS      f.root-servers.net.
.                       85384   IN      NS      g.root-servers.net.
.                       85384   IN      NS      h.root-servers.net.
```

```
.                       85384   IN      NS      i.root-servers.net.
.                       85384   IN      NS      j.root-servers.net.
.                       85384   IN      NS      k.root-servers.net.
.                       85384   IN      NS      l.root-servers.net.
.                       85384   IN      NS      m.root-servers.net.
;; Received 239 bytes from 8.8.8.8#53(8.8.8.8) in 46 ms

com.                    172800  IN      NS      e.gtld-servers.net.
com.                    172800  IN      NS      g.gtld-servers.net.
com.                    172800  IN      NS      l.gtld-servers.net.
com.                    172800  IN      NS      b.gtld-servers.net.
com.                    172800  IN      NS      d.gtld-servers.net.
com.                    172800  IN      NS      a.gtld-servers.net.
com.                    172800  IN      NS      i.gtld-servers.net.
com.                    172800  IN      NS      m.gtld-servers.net.
com.                    172800  IN      NS      c.gtld-servers.net.
com.                    172800  IN      NS      k.gtld-servers.net.
com.                    172800  IN      NS      h.gtld-servers.net.
com.                    172800  IN      NS      j.gtld-servers.net.
com.                    172800  IN      NS      f.gtld-servers.net.
;; Received 844 bytes from 192.36.148.17#53(i.root-servers.net) in 27
ms

example.com.    172800  IN      NS      ns1.p01.ourdns.com.
example.com.    172800  IN      NS      ns2.p01.ourdns.com.
example.com.    172800  IN      NS      ns3.p01.ourdns.com.
example.com.    172800  IN      NS      ns4.p01.ourdns.com.
;; Received 296 bytes from 192.5.6.30#53(a.gtld-servers.net) in 22 ms

child.example.com.      172800  IN      NS
ns1.p01.otherdns.com.
child.example.com.      172800  IN      NS
ns2.p01.otherdns.com.
child.example.com.      172800  IN      NS
ns3.p01.otherdns.com.
child.example.com.      172800  IN      NS
ns4.p01.otherdns.com.
;; Received 256 bytes from 198.51.45.8#53 (ns2.p01.ourdns.com) in 27
ms

child.example.com.      3600    IN      A       23.227.38.65
;; Received 64 bytes from 198.51.43.8#53(ns2.p01.otherdns.com) in 21
ms
```

> **Note:**
> If you are attempting to delegate an existing child zone to another DNS provider, you may need to contact `support@ourdns.com` to confirm that these changes will not adversely impact DNS resolution. If you are creating a brand-new child zone, no extra steps are required.

 Status Page
Location: https://status.ourdns.com
Introduction
Visit the OurDNS Status Page to follow incidents, past or in progress, as well as any upcoming maintenance. If you have not already done so, you can sign up for email or SMS notifications whenever OurDNS creates or updates an incident. See the example status post below, so that you know what to expect in your inbox or via SMS:

Portal and API Statistics Performance Impact
07/24/2019 07:09 UTC
We are currently experiencing an issue with our statistics subsystem. Some Portal and API statistics will be intermittently unavailable for some customers. DNS Resolution is unaffected by this issue and is currently operating as expected. Customers with further questions are encouraged to contact `support@ourdns.com.`

07/24/2019 10:14 UTC
Our Engineering team continues to actively investigate the issue. There remains no impact to DNS. Customers with further questions are encouraged to contact `support@ourdns.com.`

07/25/2019 17:13 UTC
Our Engineers are working in parallel tracks to accelerate resolution of the statistics subsystem issue as follows:
* We are building a new data pipeline that will more permanently stabilize delivery of query metrics to the API/portal. We have accelerated transitioning statistics to this new pipeline to address the ongoing metrics issue.
* Engineers are attempting to restore statistics availability on the existing data pipeline. This work has been ongoing since this issue first surfaced.

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

07/26/2019 16:18 UTC
Our Engineers have deployed a fix to restore statistics availability for the Portal and API. Account-level data, including network-level statistics, continue to propagate via the Portal and API, however, zone- and record-level query statistics remain unavailable.

We are progressing with the creation of a new statistics subsystem. This subsystem will more permanently stabilize delivery of query metrics to the Portal and API and fully restore zone- and record-level statistics. We are continuing to address this issue with the utmost

priority and will provide updates as they become available. Customers with questions are encouraged to contact `support@ourdns.com.`

07/30/2019 19:02 UTC
Our Engineers have successfully migrated zone- and record-level statistics to the new data pipeline and expect statistics to become available via the Portal and API in the next few hours.

07/31/2019 0:15 UTC
Our Engineers have successfully restored zone- and record-level statistics in the Portal and API. Please note that historical API data as well as some search/aggregation features in the Portal are currently unavailable but will be gradually restored.

We are actively monitoring this deployment. Customers with questions are encouraged to contact `support@ourdns.com.`

Incident Resolved
07/31/2019 15:24 UTC
We are pleased to report that our Engineers have successfully restored zone- and record-level statistics in the Portal and API. Please note that historical API data as well as some search/aggregation features in the Portal are currently unavailable but will be gradually restored.

We have monitored this deployment for a few hours and are now considering this resolved.
Duration: 176:15:00

Conclusion
Thank you for taking the time to review the provided samples. I hope they provided valuable insight into some of my skills and abilities. Have a nice day!

Composed in <oXygen/> XML Editor 23.0, build 2020121712