

Washington DC | June 27, 2024

aws SUMMIT



© 2024, Amazon Web Services, Inc. or its affiliates. All rights reserved.

DEV206

Create DEV, QA, and UAT environments using unique domain names

Angelo Mandato

(he/him)

AWS Community Builder / AWS Community Leader – Columbus, Ohio

Founder / CTO

Painless Analytics



© 2024, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Unique domain names for lower environments

- 01 Problem
- 02 Unique Top level domains to the rescue
- 03 Benefits
- 04 Costs
- 05 Example
- 06 Wrap up what we learned
- 07 Bonus! – local development

What is a Domain Name?

example.com - Domain Name (root hostname)

com – Top-Level Domain (TLD)

example – Second-level Domain

example.com



The diagram shows the domain name 'example.com' with two white curly braces underneath. The first brace is under 'example' and the second is under '.com'. Below the first brace is the text 'second-level domain (SLD)' and below the second brace is the text 'top-level domain (TLD)'.

second-level domain (SLD) top-level domain (TLD)

https://en.wikipedia.org/wiki/Domain_name

Problem



Sub Domains for lower environments

- beta.example.com - BETA
- uat.example.com - UAT
- 241-staging.example.com - STAGING
- pr-334-dev.example.com - DEV
- qa.example.com - QA
- www.example.com - PRODUCTION

This only works for a monolithic app!

Production Sub Domains with environments

- dashboard.example.com - PRODUCTION
- uat-dashboard.example.com - UAT
- dev-dashboard.example.com - DEV
- pr-2324-dashboard.example.com - DEV PULL REQUEST
- qa-dashboard.example.com - QA

What's wrong with this?

Sub Domains introduce problems

- Session cookies that use the root hostname
- Environments sharing root hostname
- Hard to manage root hostname in multiple Accounts*
- 3rd Party services mixed in, Google Tag Manager, Analytics, etc..
- CORS rules are complicated
- Security concerns
- **WHAT IF PROD FORMS SUBMIT TO THE QA ENVIRONMENT???**

THERE IS A BETTER WAY!

Unique Top Level Domains (TLD) to the rescue

Unique top-level domain for PRODUCTION

- `www.example.com`
- `account.example.com`
- `api.example.com`
- `mail.example.com`

`anything.example.com` is PRODUCTION

Unique top-level domain name for DEV

- `www.example.dev`
- `account.example.dev`
- `api.example.dev`
- `mail.example.dev`

`anything.example.dev` is DEV

Select TLD's for each environment

Root hostname for each environment:

- example.**com** - PRODUCTION
- example.**dev** - DEV
- example.**biz** - UAT
- example.**net** - QA
- example.**be** - BETA

If internal VPN, you can make up your own TLD **.extension!**

Benefits



Benefits of using domain names with unique TLD's



IaC

CloudFormation /
CDK environments
configured by root
hostname



Security

DEV and QA
environments less
strict than
UAT/PROD



Privacy Rights

Easy to maintain
lower environment
data separate from
production



Development

Development
never found on
production
hostnames

You are a Super Hero!



Costs



Costs for TLD's

.us - \$15/year

.cc - \$12/year

.net - \$15/year

.org - \$14/year

.biz - \$19/year

.ca - \$13/year

.bet - \$24/year

.site - \$25/year

.io - \$71/year

.link - \$5/year

.be - \$9/year

.de - \$9/year

.name - \$9/year



Route 53 TLD Registration Pricing

Internal VPC's?

.whatever **FREE!!!**

SOLD



Example

Example: Laravel / WordPress Site

www.painlessanalytics.com - PROD

www.painlessanalytics.dev - DEV

www.painlessanalytics.be - BETA

Example: ReactJS Dashboard

dashboard.painlessanalytics.com - PROD

dashboard.painlessanalytics.dev - DEV

dashboard.painlessanalytics.be - BETA

Example: Root Hostname SSL Cert

Create SSL Certificate
(environment specific)

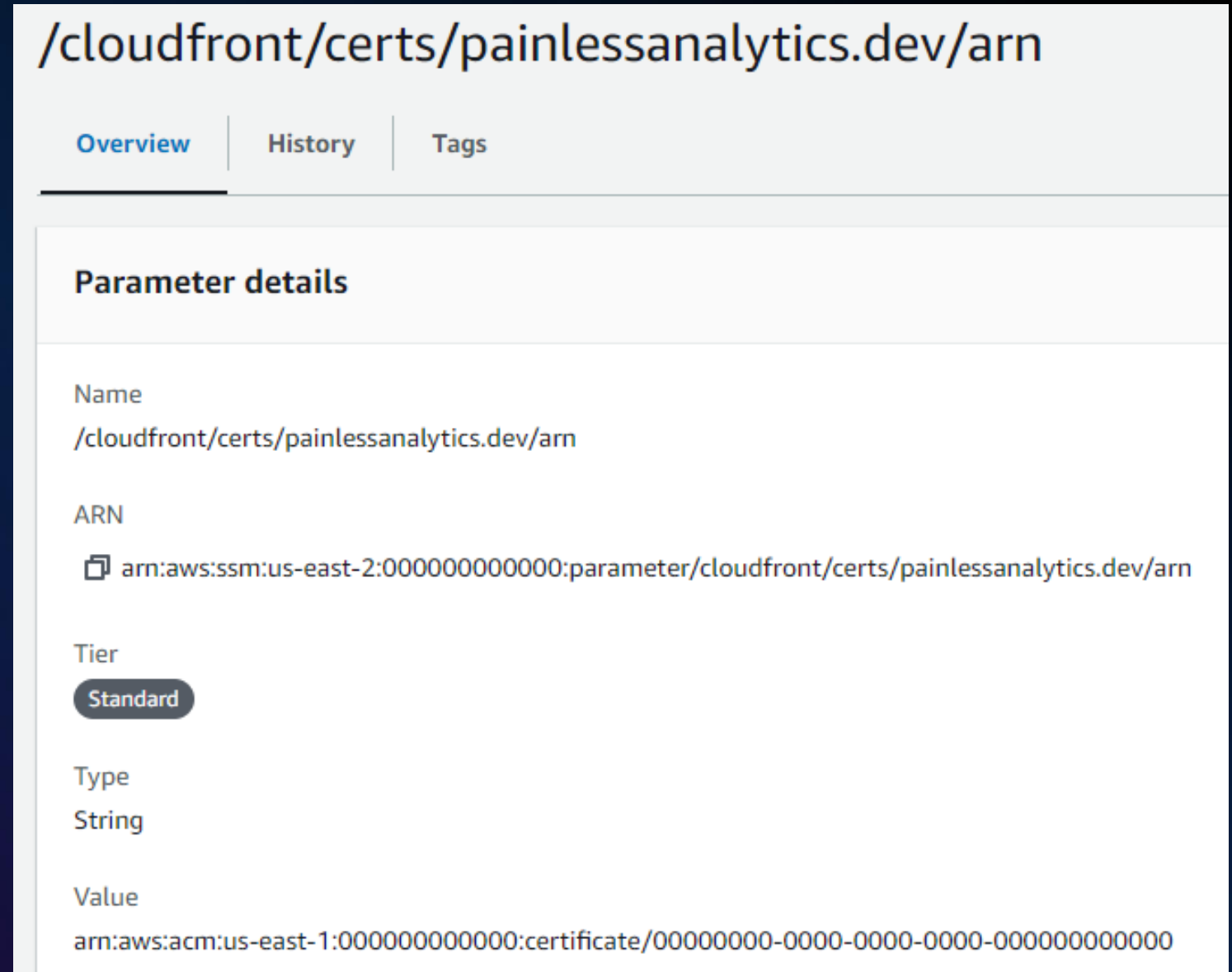
```
PS C:\projects> aws acm request-certificate --domain-name painlessanalytics.com  
--subject-alternative-names *.painlessanalytics.com --validation-method DNS  
-profile prod --region us-east-1
```

Go to Certificate Manager (ACM)

Create CNAME name records to validate certificate

Example: Setup Parameter Store

Set SSL Cert **arn**
Parameter in
Systems Manager
Parameter Store



The screenshot displays the AWS Systems Manager Parameter Store console. At the top, the parameter path `/cloudfront/certs/painlessanalytics.dev/arn` is shown. Below this, there are three tabs: **Overview**, **History**, and **Tags**. The **Overview** tab is selected, showing the **Parameter details** section. This section includes the following information:

- Name:** `/cloudfront/certs/painlessanalytics.dev/arn`
- ARN:** `arn:aws:ssm:us-east-2:000000000000:parameter/cloudfront/certs/painlessanalytics.dev/arn`
- Tier:** **Standard**
- Type:** **String**
- Value:** `arn:aws:acm:us-east-1:000000000000:certificate/00000000-0000-0000-0000-000000000000`

Example: Setup Parameter Store

Set root **Hostname**
Parameter in
Systems Manager
Parameter Store

/service/root/hostname

[Overview](#)[History](#)[Tags](#)

Parameter details

Name

/service/root/hostname

ARN

 arn:aws:ssm:us-east-2:000000000000:parameter/service/root/hostname

Tier

Standard

Type

String

Value

painlessanalytics.dev

Example: CloudFormation using root hostname

Get Parameter from
Parameter Store

Use parameter to
Get the SSL cert
arn and set the
alias hostname

```
5 Parameters:
6
7   ServiceRootHostName:
8     Type: AWS::SSM::Parameter::Value<String>
9     Default: /service/root/hostname
10    Description: "Root Hostname for service, e.g. painlessanalytics.com."
11
```

```
68  WwWCDN:
69    Type: AWS::CloudFront::Distribution
70    DeletionPolicy: Retain
71    Properties:
72      DistributionConfig:
73        Enabled: true
74        # DefaultRootObject: index.html
75        Aliases:
76          - !Sub "www.${ServiceRootHostName}"
77        ViewerCertificate:
78          AcmCertificateArn: !Sub "${resolve:ssm:/cloudfront/certs/${ServiceRootHostName}/arn}"
79          MinimumProtocolVersion: TLSv1.2_2021
80          SslSupportMethod: sni-only
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99
100
101
102
103
104
105  Tags:
106    - Key: Name
107      Value: !Sub "account.${ServiceRootHostName}"
```

Root hostname CloudFormation Advantages



Account / Lower
Environment uses a
unique root hostname

prefix.example.com

Hostname prefixes
remain consistent



Parameter Store
utilization minimized

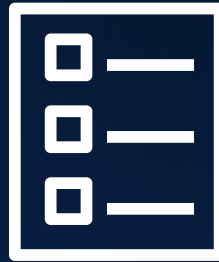


Wildcard SSL certs
Account specific

Benefits



Simple to understand,
document, and
standardize



Reproducible
AWS CloudFormation
templates



Accessible Publicly
(if not in a
private VPN)



Private Network – make
up your own TLD's!



SES can remain in
"**sandbox**" for
lower environments

Configuration Settings

Systems Manager
Parameter Store

.env & config files

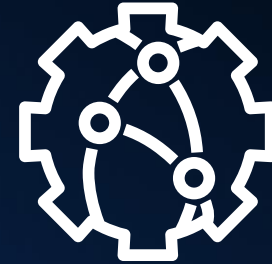
CI/CD pipelines

Hide lower environments from public?



VPN

private network is
ideal but in some
cases not possible



CloudFront Function

Block access,
allow if a cookie or
header matches a
condition

CloudFront Function: Allow Exceptions

```
1  /*
2  * Deny access to CloudFront distro unless a magic cookie is set or path in exception regex
3  * By Angelo Mandato (https://angelo.mandato.com)
4  */
5
6  const secretWord = 'sesame'; // Secret word
7  const cookieValue = '465d06d6-2e9e-4b5c-b716-e2b6d199fd02'; // Random value that if set, allows entry
8  const exceptionRegex = /^\/(api|webhook)/; // Exception regular expression
9
10 function handler(event) {
11     const req = event.request;
12
13     if( exceptionRegex && req.uri.match(exceptionRegex) ) {
14         return req; // Exception list, let them through!
15     }
16 }
```

<https://myapp.example.dev/api/version>

<https://myapp.example.dev/webhook/stripe>

CloudFront Function: Open Sesame

```
37
38 // Query string, we are looking for open=XXX
39 ✓ if( req.querystring["open"] && req.querystring["open"].value == secretWord ) {
40     var e = new Date();
41     e.setDate(e.getDate() + 1); // 1 day cookie
42
43     // Let them pass, set the cookie
44 ✓     const resp = {
45         statusCode: 200,
46         statusDescription: 'Ok',
47 ✓         cookies: {
48 ✓             "magic_cookie": {
49                 "value" : cookieValue,
50                 "attributes": "Secure; HttpOnly; Expires=" + e.toUTCString()
51             }
52         },
53 ✓         body: {
54             "encoding": "text",
55             "data": "<!DOCTYPE html><html><head><title>Welcome!</title></head><body>" +
56                 "<p>Welcome to the party! <a href=\"\"+ req.uri +\"\">Continue</a></p></body></html>"
57         }
58     }
59     return resp;
60 }
61
```

<https://myapp.example.dev/?open=sesame>

CloudFront Function: Close Sesame

```
16
17 // Allow them to delete the cookie
18 if( req.querystring["close"] && req.querystring["close"].value == secretWord ) {
19     // Let them pass, set the cookie
20     const resp = {
21         statusCode: 200,
22         statusDescription: 'Ok',
23         cookies: {
24             "magic_cookie": {
25                 "value" : "",
26                 "attributes": "Secure; HttpOnly; Expires=Thu, 01 Jan 1970 00:00:00 GMT"
27             }
28         },
29         body: {
30             "encoding": "text",
31             "data": "<!DOCTYPE html><html><head><title>Good bye!</title></head><body>"+
32                 "<p>Thanks for all the fish!</p></body></html>"
33         }
34     }
35     return resp;
36 }
37
```

<https://myapp.example.dev/?close=sesame>

CloudFront Function: Block site if no cookie set

```
61
62 // If no cookie or the cookie is not the value we are looking for
63 if( !req.cookies["magic_cookie"] || req.cookies["magic_cookie"].value != cookieValue ) {
64     // Access denied!
65     const resp = {
66         statusCode: 503,
67         statusDescription: 'Service Unavailable',
68         body: {
69             encoding: "text",
70             data: "<!DOCTYPE html><html><head><title>Sorry!</title>"+
71                 "<meta name=\"robots\" content=\"noindex, nofollow\" />"+
72                 "</head><body><p>No soup for you!</p></body></html>"
73         }
74     };
75     return resp;
76 }
77
78 return req; // Let them have cake!!!
79 }
80 // eof
```

Wrap up what we learned

Unique Domains for lower environments



Unique Domains for lower environments

- No more confusing sub domains
e.g. psycho-db-devbeta.example.com
- No more session / cookie collisions with prod and lower environments
- Secure lower environments with private networks
- Prevent access to lower environments with CloudFront functions
- SES Sandbox mode is a feature for non-prod environments
- IaC / CloudFormation attributes use consistent prefixes
- Switching from one environment to the other is as easy as replacing ".com" with ".dev" and hitting ENTER.

Bonus! – local development

Use a TLD for local development!

With your favorite editor open your local hosts file

MacOS: `/private/etc/hosts`

Linux: `/etc/hosts`

Windows: `C:\Windows\System32\drivers\etc\hosts`
(run editor as administrator)

Tip: Check out Windows PowerToys' **Hosts File Editor**

411 on hosts file

Find your inner self, **127.0.0.1** of course!

```
127.0.0.1 www.painlessanalytics.local painlessanalytics.local
127.0.0.1 account.painlessanalytics.local
127.0.0.1 api.painlessanalytics.local
127.0.0.1 dashboard.painlessanalytics.local
127.0.0.1 dashboard-dev.painlessanalytics.local

# IPv6 flavor
::1 www.painlessanalytics.local painlessanalytics.local
::1 account.painlessanalytics.local
::1 api.painlessanalytics.local
::1 dashboard.painlessanalytics.local
::1 dashboard-dev.painlessanalytics.local
```

Use a real TLD for local development

Use case:

- OAuth2 (services like Google will not use .local hostnames)

```
# Temporary for testing Google OAuth2
127.0.0.1 account.painlessanalytics.net
127.0.0.1 dashboard.painlessanalytics.net
127.0.0.1 dashboard-dev.painlessanalytics.net
127.0.0.1 api.painlessanalytics.net
```