HCL Software Academy
for Digital Solutions

Domino V12 Certificate Management

Daniel Nashed

HCL Lifetime Ambassador

HCL SOFTWARE

Agenda

- Theory
  - Introduction / Motivation
  - certstore.nsf & CertMgr
  - Manual Flows
  - Let's encrypt / ACME Support
  - New TLS Credential Cache
- **Live Demo** & Questions
- Troubleshooting Slides
- Q&A
- Bonus: Build your own Lab

Domino V12 Design Goal

- **Simplify** Domino certificate management
- **No external tools** like OpenSSL command-line to create keys and convert certificates needed!
- Replace difficult to handle ***.kyr** files with standard ***.pem** format
- Full **Let's Encrypt®** / ACME CA integration
- Simplified flows for external certificate authorities
- Domain wide secure and automated deployment for “TLS Credentials”
- Automated update of certificates including automatic cache update in internet server tasks
- Support modern standards like **ECDSA** in addition to RSA

Technology used for CertMgr

- Native Servertask & DSAPI Filter (C/C++)

- Leverages existing and new Notes security APIs

- Implements Let's Encrypt uses ACME protocol V2 (RFC 8555)
  - ACME = **A**utomatic **C**ertificate **M**anagement **E**nvironment
  - Own HCL implementation leveraging standards like
    - JSON, LibCurl, JWS, Notes crypto including PEM, RSA and ECDSA keys, (OpenSSL) …

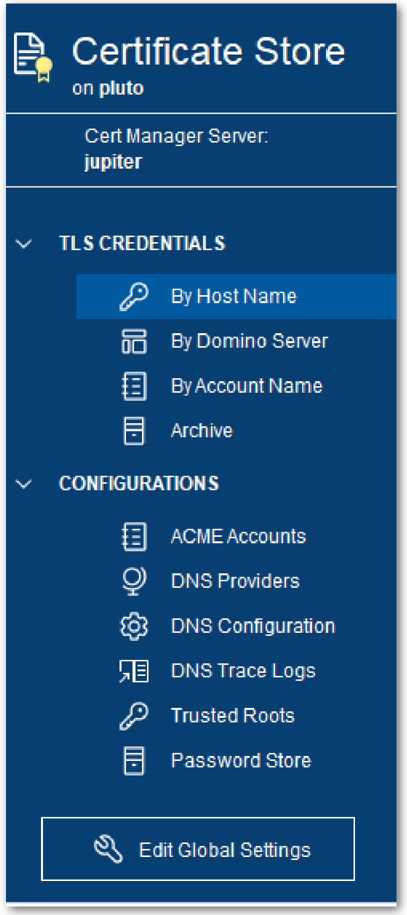
- Designed for "automation"

Domain wide CertStore Database & CertMgr

Privates Keys, Certificates,Trusted Roots

New - certstore.nsf

- Domain wide database managed by **CertMgr** task
- **Secure**, automated deployment for TLS Credentials and trusted roots
- Private keys are **encrypted** with CertMgr server
  and the server specified in the field “**Servers with access:**”
  - Special designed Vault style encryption with new API
- Easy to use with modern interface
- CertMgr servertask is only supported on **W64** and **Linux64**
  - **AIX** and **OS400** can still leverage certstore.nsf and the new TLS Cache
    - Create replica manually

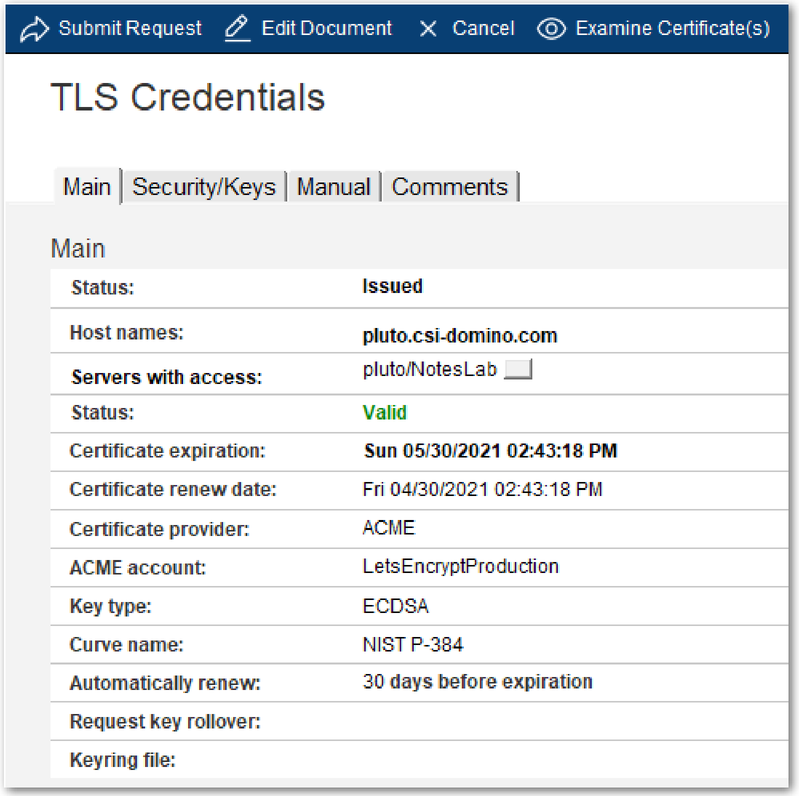


Create certstore.nsf on CertMgr Server

- <u>First</u> server in domain starting the “**certmgr**” servertask is setup as the CertMgr Server
  - Checks the Domino **directory profile** on **admin server** for an existing CertMgr server
  - If no server exists automatically creates the domain wider **certstore.nsf** database
  - Updates the directory profile on admin server to propagate the CertMgr server in the domain
- Starting the certmgr servertask on any **additional** server in the domain creates a replica
  - Each additional server acts like a “**CertMgr client**” and will just replicate the database every 2 minutes
  - Keeping the CertMgr servertask loaded is an <u>optional</u> convenience step
  - Any type of replication setup which ensures a short replication cycle can be used as well

certstore.nsf – TLS Credentials

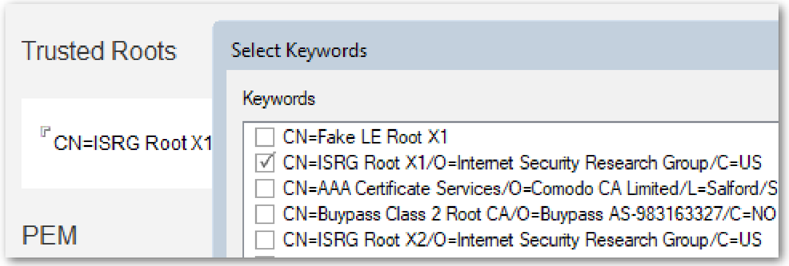
- TLS Credential = **private key** + **leaf certificate** + **chain** (intermediates) + **trusted root**
- Replaces "***.kyr** files"
  - Stored in **PEM** format (text with base64 encoded data)
- Can be created via
  - Manual flows including import
  - **ACME** protocol (Let's Encrypt & others)
- Specify trusted roots used for client certificate verification
- Used to be hidden in kyr-file and was difficult to manage



TLS Credentials

Main

Status:	Issued
Host names:	pluto.csi-domino.com
Servers with access:	pluto/NotesLab
Status:	Valid
Certificate expiration:	Sun 05/30/2021 02:43:18 PM
Certificate renew date:	Fri 04/30/2021 02:43:18 PM
Certificate provider:	ACME
ACME account:	LetsEncryptProduction
Key type:	ECDSA
Curve name:	NIST P-384
Automatically renew:	30 days before expiration
Request key rollover:	
Keyring file:	



Trusted Roots

CN=ISRG Root X1

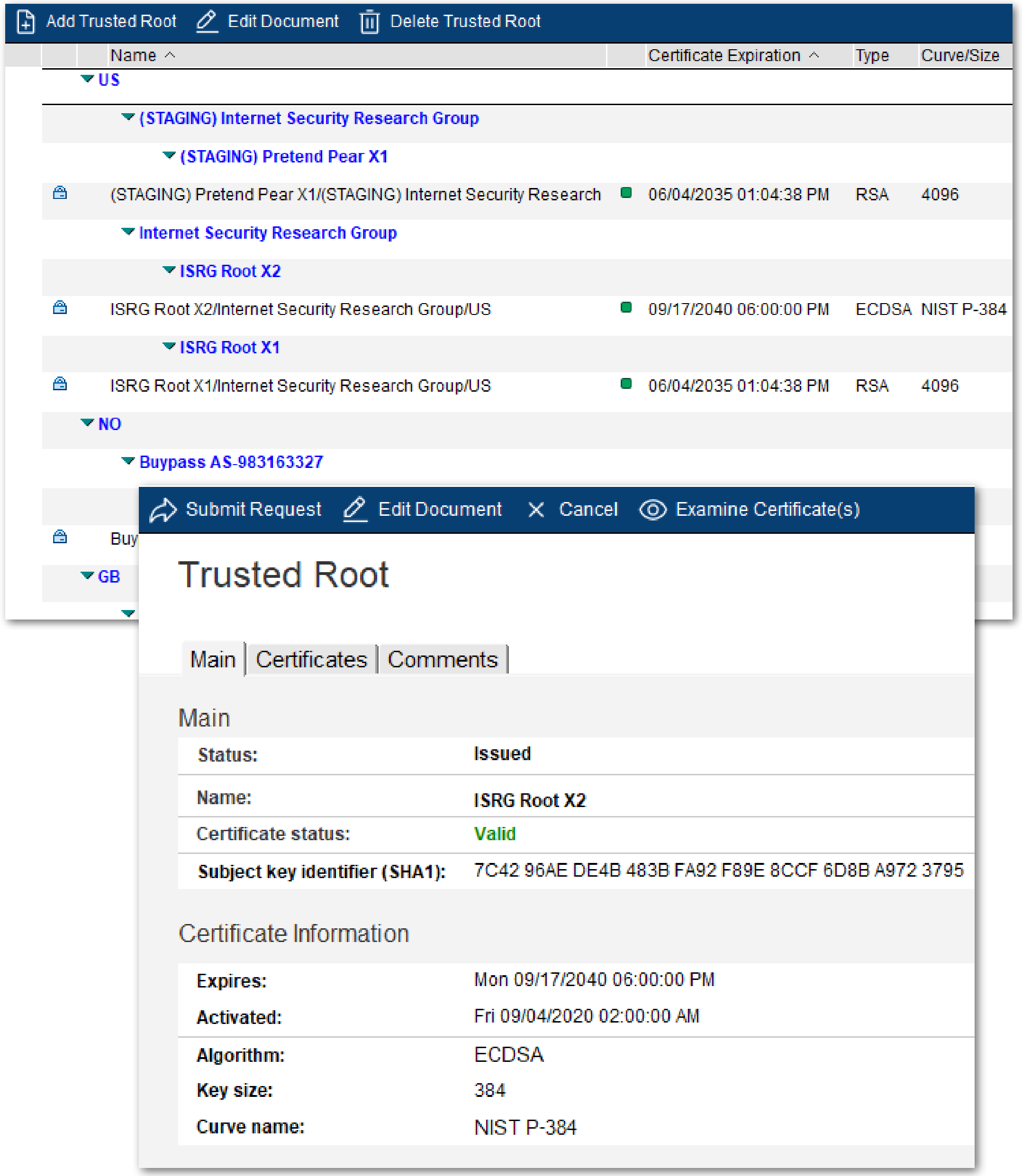
PEM

- [ ] CN=Fake LE Root X1
- [x] CN=ISRG Root X1/O=Internet Security Research Group/C=US
- [ ] CN=AAA Certificate Services/O=Comodo CA Limited/L=Salford/S
- [ ] CN=Buypass Class 2 Root CA/O=Buypass AS-983163327/C=NO
- [ ] CN=ISRG Root X2/O=Internet Security Research Group/C=US

HCL Software Academy for Digital Solutions HCL SOFTWARE

certstore.nsf – Trusted Root

- Stored in trusted, secured certstore.nsf
  - Replicated domain wide
- Used for client cert verification
- And auto complete certificate chains
  - ACME and manual flows
- Certificate chains are automatically sorted & completed
  - **Private Key** → matching **leaf certificate**
    → **intermediate certs** in the right order → **trusted root**
- Tip: you can import intermediate certificates as "Trusted Root" to be used to auto complete chains
  - Just listed with warning, that they are no root



Add Trusted Root | Edit Document | Delete Trusted Root

	Name ^		Certificate Expiration ^	Type	Curve/Size
	US				
	(STAGING) Internet Security Research Group				
	(STAGING) Pretend Pear X1				
🔒	(STAGING) Pretend Pear X1/(STAGING) Internet Security Research	🟩	06/04/2035 01:04:38 PM	RSA	4096
	Internet Security Research Group				
	ISRG Root X2				
🔒	ISRG Root X2/Internet Security Research Group/US	🟩	09/17/2040 06:00:00 PM	ECDSA	NIST P-384
	ISRG Root X1				
🔒	ISRG Root X1/Internet Security Research Group/US	🟩	06/04/2035 01:04:38 PM	RSA	4096
	NO				
	Buypass AS-983163327				
🔒	Buy				
	GB				

Submit Request | Edit Document | Cancel | Examine Certificate(s)

Trusted Root

Main | Certificates | Comments

Main

Status:	Issued
Name:	ISRG Root X2
Certificate status:	Valid
Subject key identifier (SHA1):	7C42 96AE DE4B 483B FA92 F89E 8CCF 6D8B A972 3795

Certificate Information

Expires:	Mon 09/17/2040 06:00:00 PM
Activated:	Fri 09/04/2020 02:00:00 AM
Algorithm:	ECDSA
Key size:	384
Curve name:	NIST P-384

HCL Software Academy for Digital Solutions   HCL SOFTWARE

New - Support for Elliptic Curves – “ECDSA Keys”

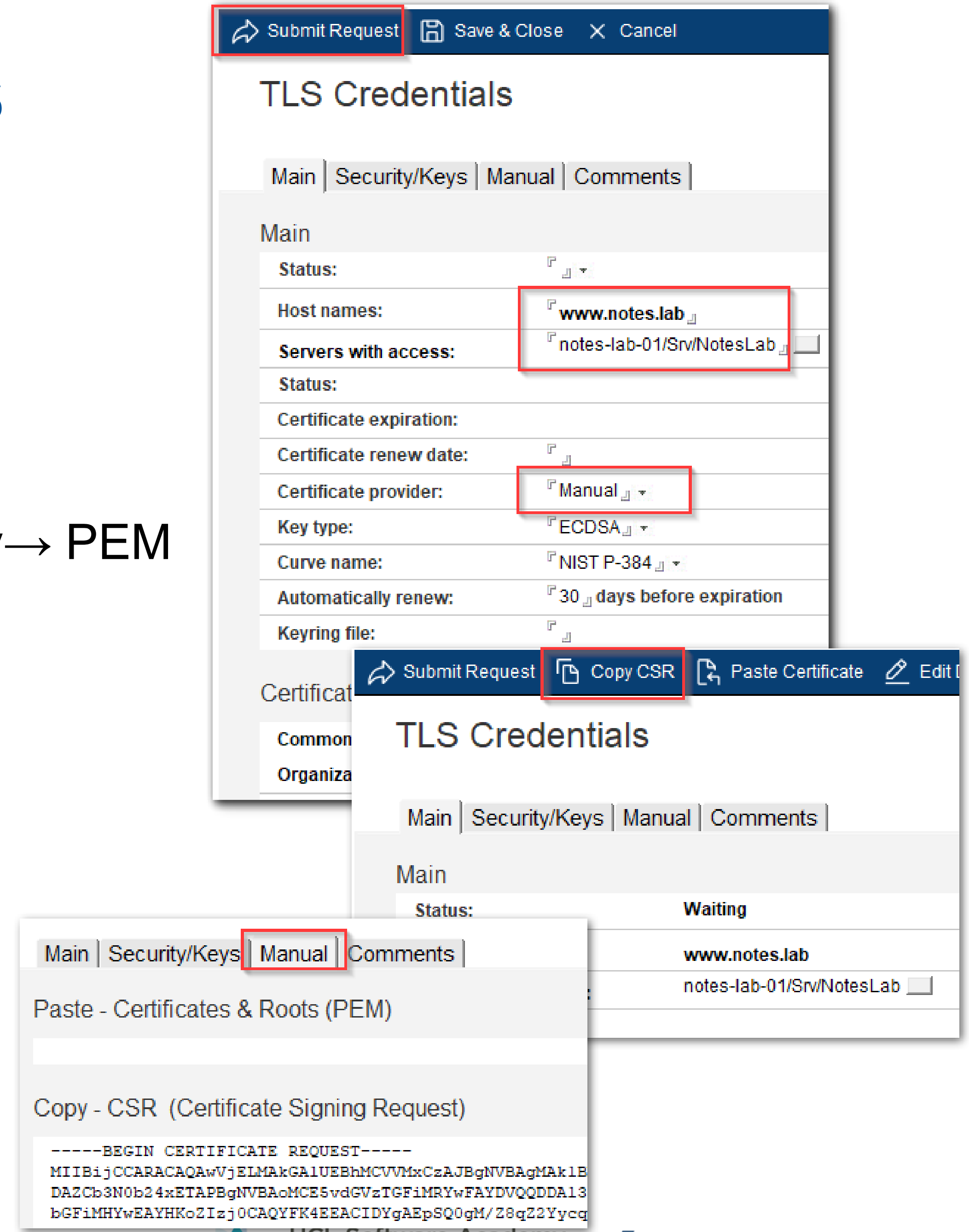
- ECDSA is the more modern, more secure standard with less overhead
  - **256 bit** (NIST P-256) ECDSA key → **3072 bit** RSA key or a **128 bit** AES key.
  - **384 bit** (NIST P-384) ECDSA key → **7680 bit** RSA key or a **192 bit** AES key.
  - **512+ bit** ECDSA key (NIST P-521) → **15360 bit** RSA key or a **256 bit** AES key.
- Fully supported in the Domino V12 TLS/SSL stack
  - Support for RSA and ECDSA key types in parallel
- With **ECDSA** the following ciphers are automatically used instead of the cipher config
  - **TLS_ECDHE_ECDSA_WITH_AES_128_GCM_SHA256 (0xC02B)**
  - **TLS_ECDHE_ECDSA_WITH_AES_256_GCM_SHA384 (0xC02C)**
- Background Information ECDSA
  - https://blog.cloudflare.com/ecdsa-the-digital-signature-algorithm-of-a-better-internet/

Support for two important TLS 1.2 Curves – X25519 & X448

- Strong security with improved performance
  - Details: https://en.wikipedia.org/wiki/Curve25519
- The new order of curves
  - **Curve X25519**
  - Curve NIST P-256
  - **Curve X448**
  - Curve NIST P-384
  - Curve NIST P-521
- Notes.ini parameter per curve
  - All curves enabled by default
  - Current best practice
- If really needed disable individual curves
  - SSL_DISABLE_CURVE_X25519=1
  - SSL_DISABLE_CURVE_P256=1
  - SSL_DISABLE_CURVE_X448=1
  - SSL_DISABLE_CURVE_P384=1
  - SSL_DISABLE_CURVE_521=1

Manual Certificate Operations

- **1.** CertMgr processes submitted requests and creates
  - Private key (RSA or ECDSA)
    - Saved locally encrypted for assigned servers
- CSR (**C**ertificate **S**igning **R**equest) signed by private key→ PEM
- **2.** Admin copies CSR to CA
- **3.** Admin imports certificate & chain (PEM) back
- Paste full chain in <u>any</u> order and <u>submits</u> the form again
- Duplicate certs are ignored
- Missing intermediate certs and root are automatically added from “Trusted Roots” in certstore.nsf



Status:	
Host names:	www.notes.lab
Servers with access:	notes-lab-01/Srv/NotesLab
Status:	
Certificate expiration:	
Certificate renew date:	
Certificate provider:	Manual
Key type:	ECDSA
Curve name:	NIST P-384
Automatically renew:	30 days before expiration
Keyring file:	

Status:	Waiting
	www.notes.lab
	notes-lab-01/Srv/NotesLab

Paste - Certificates & Roots (PEM)

Copy - CSR (Certificate Signing Request)

```
-----BEGIN CERTIFICATE REQUEST-----
MIIBijCCARACAQAwVjELMAkGA1UEBhMCVVMxCzAJBgNVBAgMAk1B
DAZCb3N0b24xETAPBgNVBAoMCE5vdGVzTGFiMRYwFAYDVQQDDA13
bGFiMHYwEAYHKoZIzj0CAQYFK4EEACIDYgAEpSQ0gM/Z8qZ2Yycq
```

HCL Software Academy for Digital Solutions

HCL SOFTWARE

Certificate Health Check & Inspecting Certificate Chains

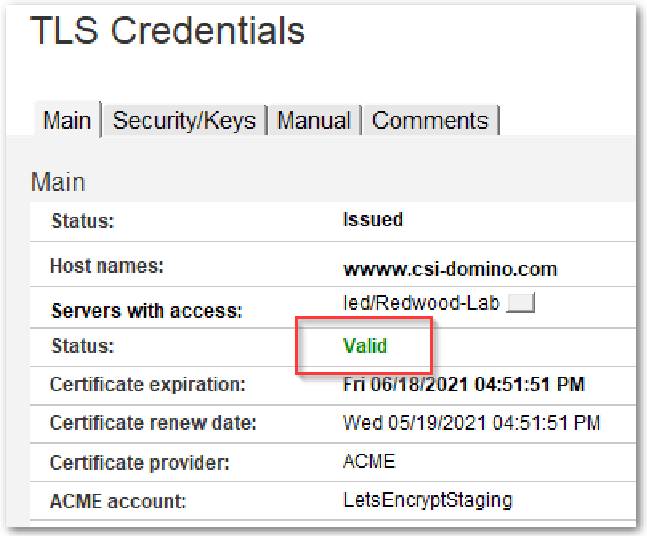
- All certificate operations check if the certificate is valid
  - Status: Valid, Warnings, Errors
  - Detailed warning and error messages

- Most common warning:

  "Last cert in chain NOT self signed – No root found"

  - **Not an error** - Just means that there is no trusted root
  - Trusted roots can be imported separately and are added to the chain if present

- Updates CertMgr statistics to reflect the current health status

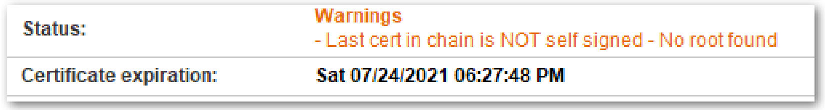


TLS Credentials

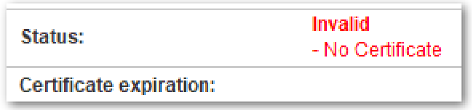
Main | Security/Keys | Manual | Comments |

Main

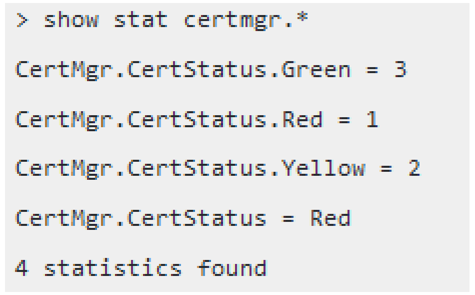
Status:	Issued
Host names:	wwww.csi-domino.com
Servers with access:	led/Redwood-Lab
Status:	Valid
Certificate expiration:	Fri 06/18/2021 04:51:51 PM
Certificate renew date:	Wed 05/19/2021 04:51:51 PM
Certificate provider:	ACME
ACME account:	LetsEncryptStaging



Status:	Warnings - Last cert in chain is NOT self signed - No root found
Certificate expiration:	Sat 07/24/2021 06:27:48 PM



Status:	Invalid - No Certificate
Certificate expiration:	

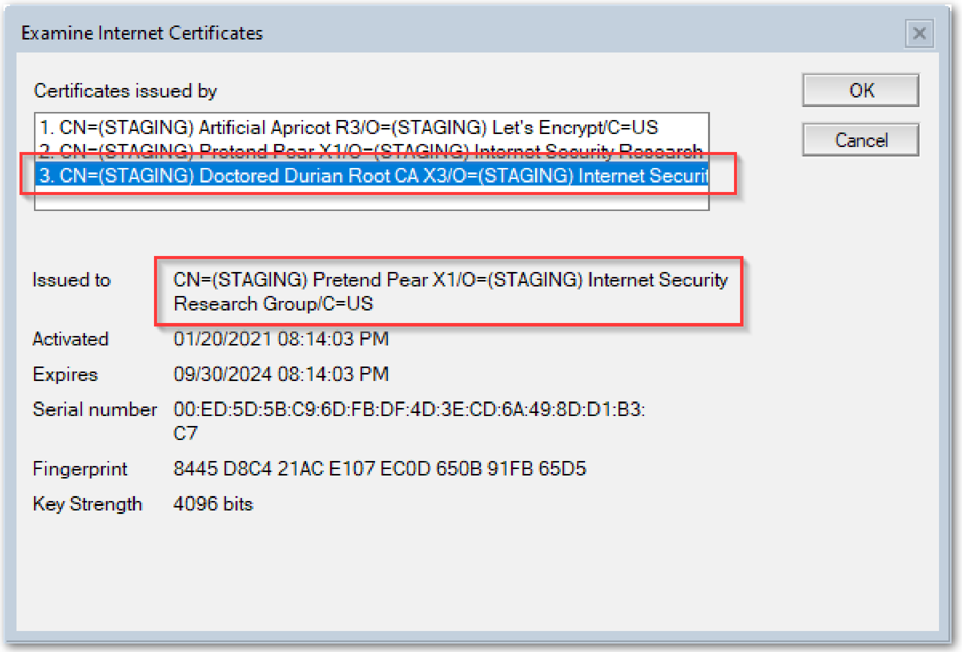


```
> show stat certmgr.*
CertMgr.CertStatus.Green = 3
CertMgr.CertStatus.Red = 1
CertMgr.CertStatus.Yellow = 2
CertMgr.CertStatus = Red
4 statistics found
```

HCL Software Academy for Digital Solutions   HCL SOFTWARE

Certificate Details

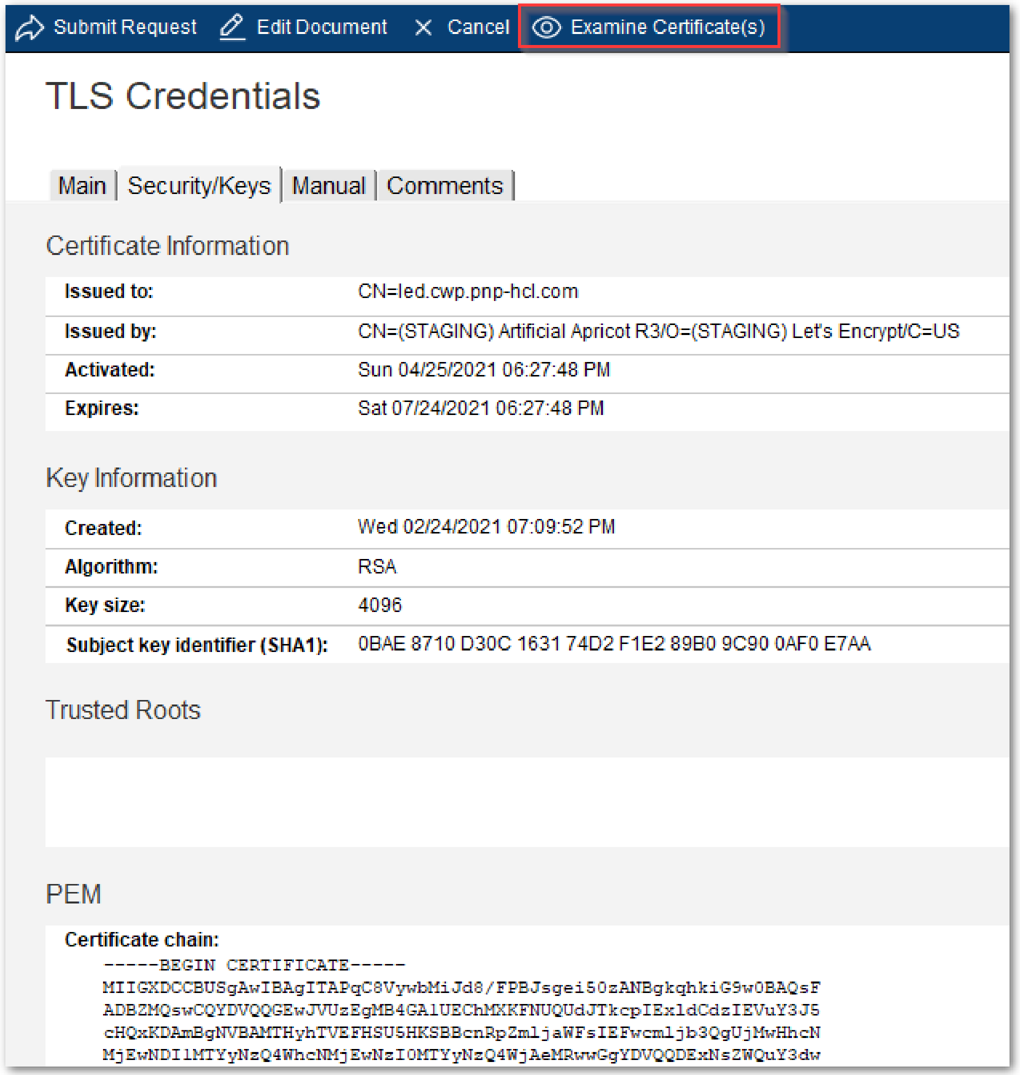
- Examine Certificate(s) Dialog
- Copy Certificate chain to check with external tools
  - e.g. openssl x509 -in my.pem -text –noout
  - Certificate information


Examine Internet Certificates

Certificates issued by

1. CN=(STAGING) Artificial Apricot R3/O=(STAGING) Let's Encrypt/C=US
2. CN=(STAGING) Pretend Pear X1/O=(STAGING) Internet Security Research
3. CN=(STAGING) Doctored Durian Root CA X3/O=(STAGING) Internet Securit

Issued to	CN=(STAGING) Pretend Pear X1/O=(STAGING) Internet Security Research Group/C=US
Activated	01/20/2021 08:14:03 PM
Expires	09/30/2024 08:14:03 PM
Serial number	00:ED:5D:5B:C9:6D:FB:DF:4D:3E:CD:6A:49:8D:D1:B3:C7
Fingerprint	8445 D8C4 21AC E107 EC0D 650B 91FB 65D5
Key Strength	4096 bits



TLS Credentials

Certificate Information

Issued to:	CN=led.cwp.pnp-hcl.com
Issued by:	CN=(STAGING) Artificial Apricot R3/O=(STAGING) Let's Encrypt/C=US
Activated:	Sun 04/25/2021 06:27:48 PM
Expires:	Sat 07/24/2021 06:27:48 PM

Key Information

Created:	Wed 02/24/2021 07:09:52 PM
Algorithm:	RSA
Key size:	4096
Subject key identifier (SHA1):	0BAE 8710 D30C 1631 74D2 F1E2 89B0 9C90 0AF0 E7AA

Trusted Roots

PEM

Certificate chain:

```
-----BEGIN CERTIFICATE-----
MIIGXDCCBUSgAwIBAgITAPqC8VywbMiJd8/FPBJsgei50zANBgkqhkiG9w0BAQsF
ADBZMQswCQYDVQQGEwJVUzEgMB4GA1UEChMXKFNUQUdJTkcpIExldCdzIEVuY3J5
cHQxKDAmBgNVBAMTHyhTVEFHSU5HKSBBcnRpZmljaWFsIEFwcmljb3QgUjMwHhcN
MjEwNDI1MTYyNzQ4WhcNMjEwNzI0MTYyNzQ4WjAeMRwwGgYDVQQDExNsZWQuY3dw
```

HCL Software Academy for Digital Solutions

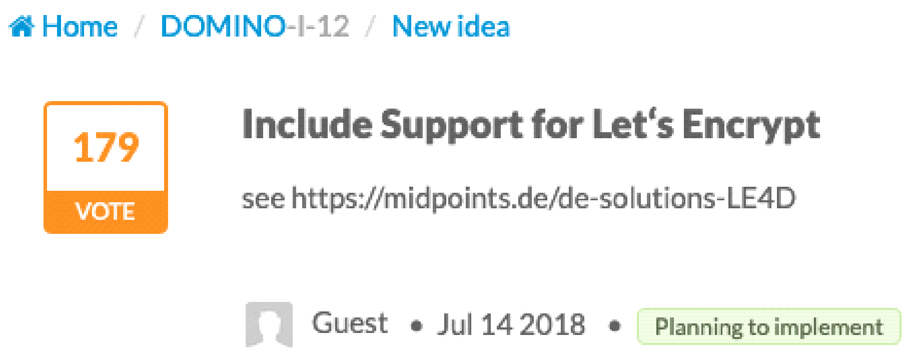
HCL SOFTWARE

Let's Encrypt / ACME Support

Introduction and HTTP-01 Challenges

Automated Certificate Management

- Support for Let's Encrypt
  - ACME protocol V2 (**RFC 8555**)
  - **A**utomatic **C**ertificate **M**anagement **E**nvironment
- Free of charge SSL/TLS certificates
- Fully integrated into **certstore.nsf** & **CertMgr**
- Easy to deploy
- Automatic certificate update (request) and deployment (reload on server)


Home / DOMINO-I-12 / New idea

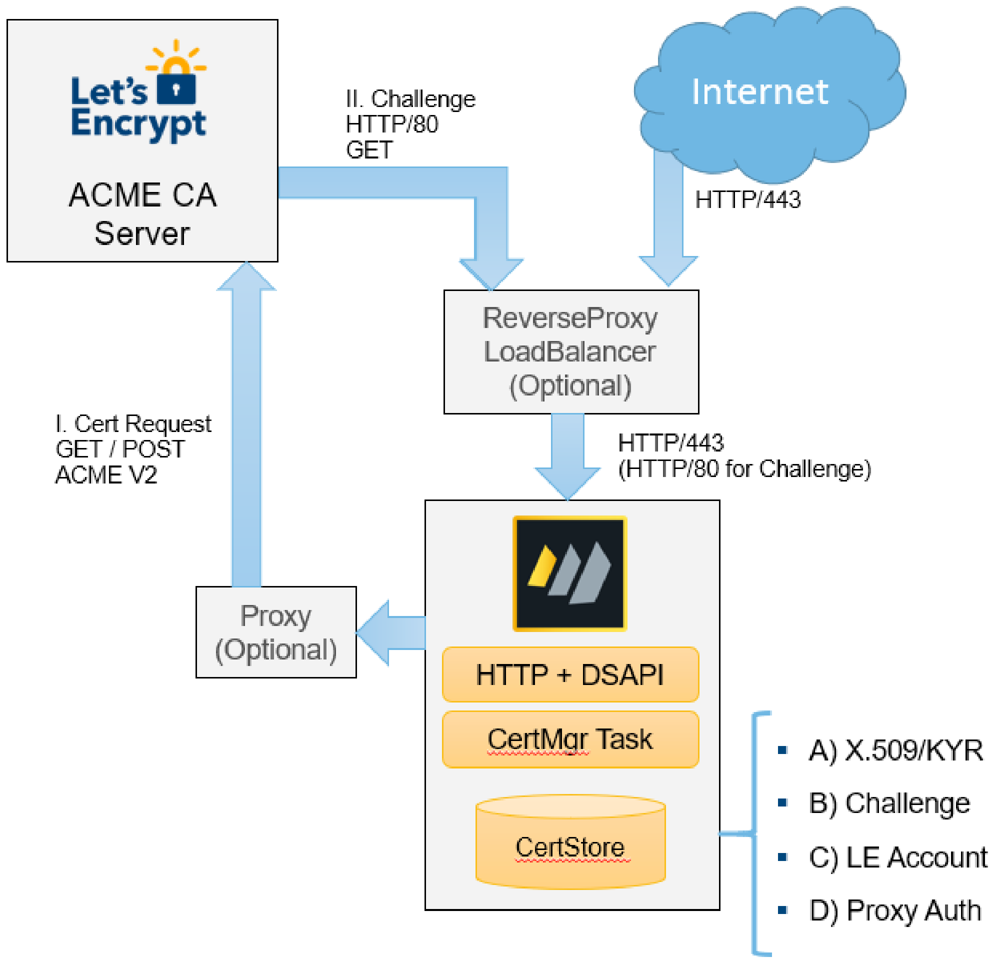
179 VOTE

Include Support for Let's Encrypt

see https://midpoints.de/de-solutions-LE4D

Guest • Jul 14 2018 • Planning to implement

Architecture Diagram



Components

*) Connection between Domino, LE and CertStore could be local or NRPC

Domino HTTP and LE could be on separate server and just need a common CertStore.

(A) X.509 today in kyr file
(B) Challenge needed to verify request
(C) LE used to authenticate with ACME CA
(D) Proxy account needed for outgoing communication

Flow

1. LE creates account (C) with ACME server
2. LE creates private key and writes it to CertStore (A)
3. LE creates CSR and sends it to ACME CA *)
4. LE puts received challenge (B) in CertStore
6. ACME server requests challenge on port 80 to verify
7. Domino HTTP replies with challenge (B) from CertStore
8. LE receives certificate including and writes it to CertStore (A)

HTTP (and INET tasks) read X.509 from CertStore (A)

*) Proxy communication uses Proxy user (D)

ACME HTTP-01 Challenges

- How it works
  1. ACME server sends a challenge to ACME client
  2. ACME server will ask via in-bound HTTP port 80 for the “secret” at a well-known URL
- DSAPI Filter “certmgrdsapi” needs to be enabled in server doc / internet site !!
  - Tip: **load certmgr -c** adds the DSAPI filter to server doc – Internet sites need to specify manually
- If server is configured to only allow authenticated connection configure public URL
  - Notes.ini: HTTPPUBLICURLS=/**.well-known/acme-challenge/***:/redir.nsf/*:/MFASetup*
- Again: **Inbound HTTP port 80 required**!
- If the server is not reachable by the ACME server (e.g. Let's Encrypt), the challenge fails !!!
  - Tip: Inbound connection can be a proxy connection

ACME Provider Let‘s Encrypt – included in template

- Let's Encrypt Staging
  - https://letsencrypt.org/docs/staging-environment
  - Should always be used for first steps testing connectivity
  - Provides the same functionality like Let's Encrypt production
  - Much higher limits for certificates and errors

- Let's Encrypt production
  - https://letsencrypt.org

Additional tested ACME Providers

- ACME is a standard supported by more providers
- New ACME provider can be added using their published directory URL
- ZeroSSL *)
  - https://zerossl.com
  - Requires registration + external account binding (EAB)
- BuyPass *)
  - https://buypass.com

On premise ACME CA

- SmallStep ACME CA
  - https://smallstep.com/docs/tutorials/acme-challenge
- CA with ACME functionality
- Can also operate as “Sub-CA” for an existing corporate CA
- Good choice for internal customer deployments
- Another tested environment
- Directory URL configured depends on your deployment
- Setup on Docker in 10 minutes!

ACME DNS-01 Challenges

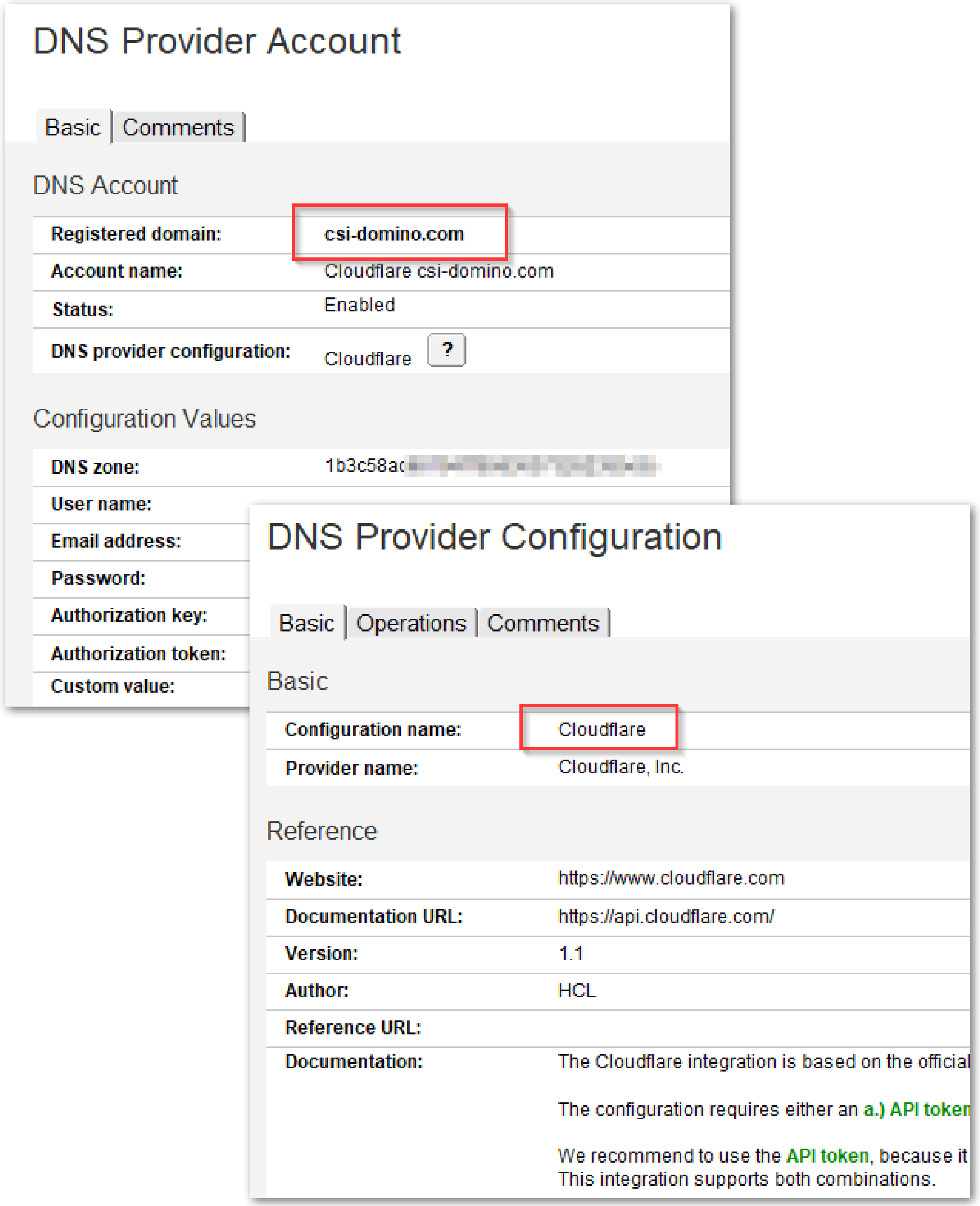
Background, Providers and Configuration

ACME DNS-01 Challenges & Wild Card Certificates

- Allows to request certificates without inbound internet connection
- ACME Challenge is stored in **DNS TXT** record
- Supports wild card certificates! e. g. ***.acme.com**
- Requires DNS provider API with outgoing HTTPS connection to DNS provider
  - No inbound connection needed
  - Can leverage outgoing proxy connections
- CertMgr server can request certificates for any server in the DNS domain

DNS Provider Interface

- Triggered by configured “**registered domain**”
  - Choose the right provider
  - Specify provider specific information
- **Build-in** support to integrate DNS providers
  - Easy to integrate REST interface (@Formulas)
    - Recommended interface!
    - Works for most providers
    - “Low code approach”
- Notes Agent
- Command-Line Integration



DNS Provider Account

Basic | Comments

DNS Account

Registered domain:	csi-domino.com
Account name:	Cloudflare csi-domino.com
Status:	Enabled
DNS provider configuration:	Cloudflare

Configuration Values

DNS zone:	1b3c58ac[redacted]
User name:	
Email address:	
Password:	
Authorization key:	
Authorization token:	
Custom value:	

DNS Provider Configuration

Basic | Operations | Comments

Basic

Configuration name:	Cloudflare
Provider name:	Cloudflare, Inc.

Reference

Website:	https://www.cloudflare.com
Documentation URL:	https://api.cloudflare.com/
Version:	1.1
Author:	HCL
Reference URL:	
Documentation:	The Cloudflare integration is based on the official The configuration requires either an a.) API token We recommend to use the API token, because it This integration supports both combinations.

HCL Software Academy for Digital Solutions HCL SOFTWARE

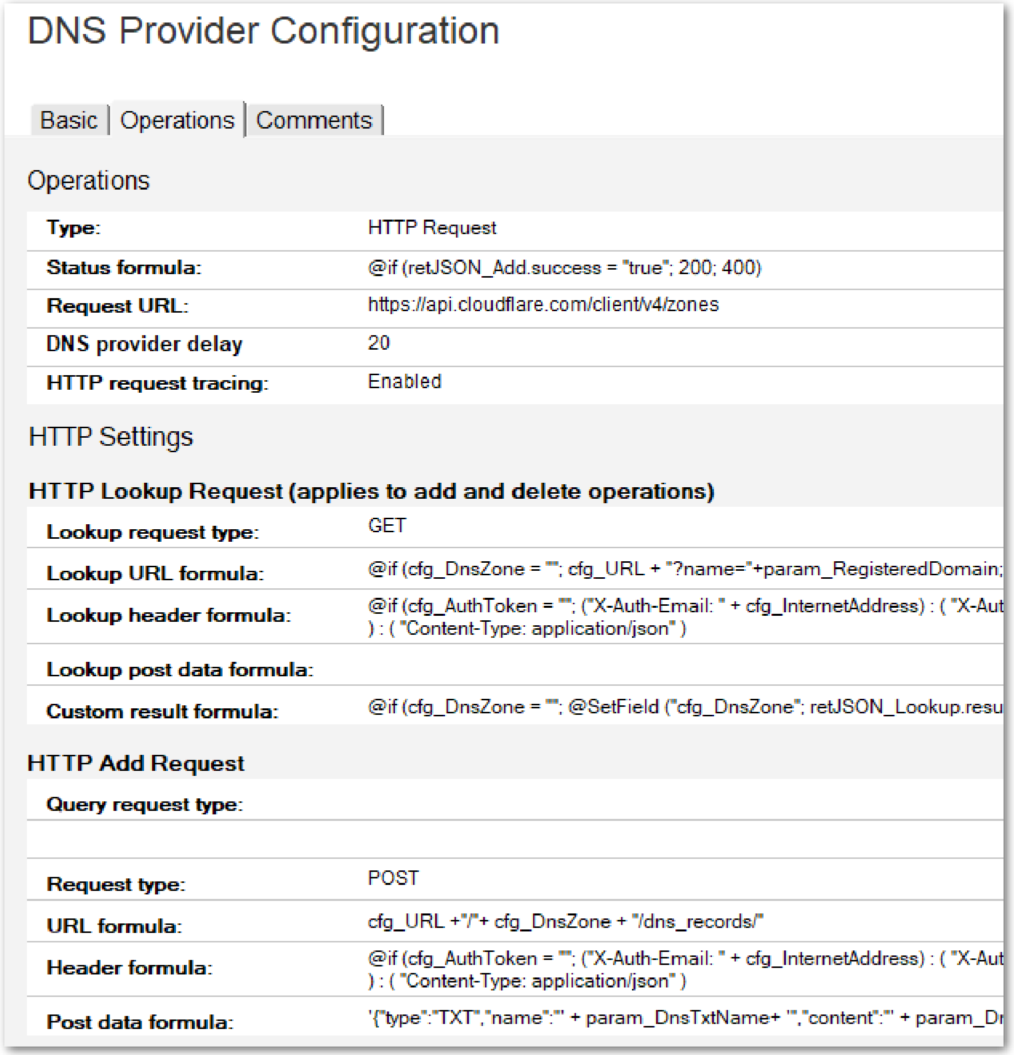
Available DNS TXT Integration

- REST API
  - Cloudflare, Inc
  - Digital Ocean, LLC
  - Hetzner Online GmbH (Germany)
  - ACME DNS
  - Let's Encrypt Pebble test server

- Command-Line
  - AWS Route 53 DNS

DNS Provider Interface – Import DXL

- DNS TXT API DXL files are not included in certstore.ntf
- Provider interfaces can be imported via DXL files
  - Database action: **Import DXL**
- REST interface is based on @formulas
  - Low code approach
  - Can parse JSON responses
  - Helper buttons for inserting fields & testing formulas
  - Trace results useful for troubleshooting and development


DNS Provider Configuration

Operations

Type:	HTTP Request
Status formula:	@if (retJSON_Add.success = "true"; 200; 400)
Request URL:	https://api.cloudflare.com/client/v4/zones
DNS provider delay	20
HTTP request tracing:	Enabled

HTTP Settings

HTTP Lookup Request (applies to add and delete operations)

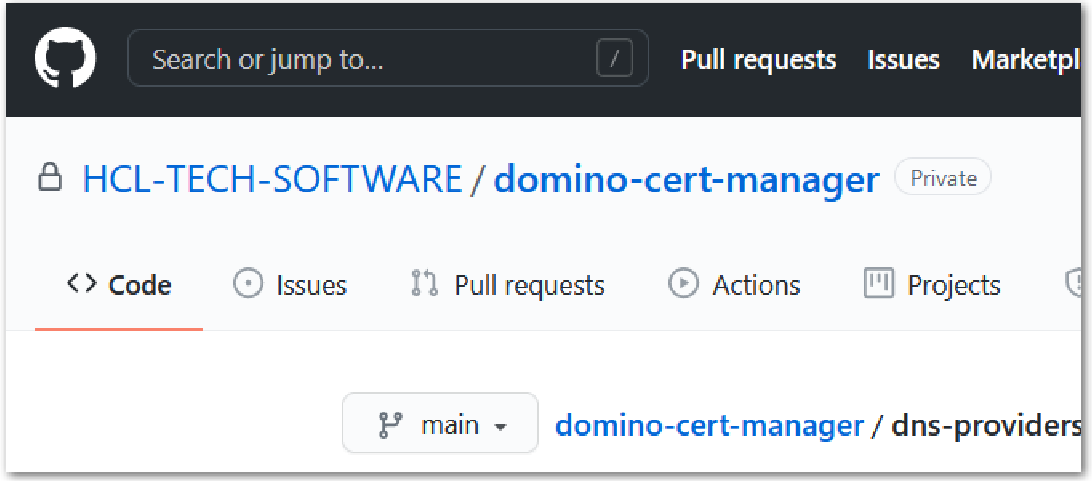
Lookup request type:	GET
Lookup URL formula:	@if (cfg_DnsZone = ""; cfg_URL + "?name="+param_RegisteredDomain;
Lookup header formula:	@if (cfg_AuthToken = ""; ("X-Auth-Email: " + cfg_InternetAddress) : ("X-Aut) : ("Content-Type: application/json")
Lookup post data formula:	
Custom result formula:	@if (cfg_DnsZone = ""; @SetField ("cfg_DnsZone"; retJSON_Lookup.resu

HTTP Add Request

Query request type:	
Request type:	POST
URL formula:	cfg_URL +"/"+ cfg_DnsZone + "/dns_records/"
Header formula:	@if (cfg_AuthToken = ""; ("X-Auth-Email: " + cfg_InternetAddress) : ("X-Aut) : ("Content-Type: application/json")
Post data formula:	'{"type":"TXT","name":"' + param_DnsTxtName+ '","content":"' + param_Dr

HCL Github repository for CertMgr

- Available as of today
- Brand new git repository for DNS TXT provider integration
- Contains ready to import and use DXL files and scripts
- Intendent to build, share and collaborate DNS provider configurations


Search or jump to...
Pull requests Issues Marketpl

HCL-TECH-SOFTWARE / domino-cert-manager Private

Code Issues Pull requests Actions Projects

main domino-cert-manager / dns-providers

HCL Software Academy for Digital Solutions HCL SOFTWARE

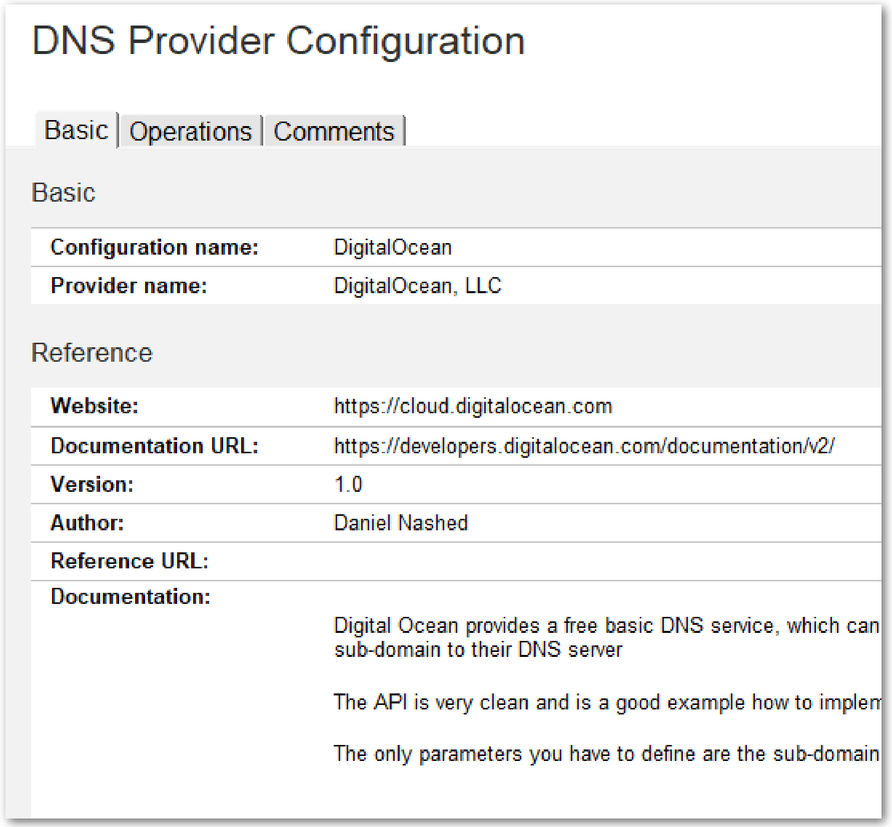
Building DNS TXT API Interfaces

Low code approach with REST API and JSON

- Many DNS providers offer a **REST** base interface to manage DNS records
- Modern interfaces with **JSON** payloads
- CertMgr preferred integration option: HTTP request with @Formulas
- Technology used
  - **@Formulas**
  - **HTTP/HTTPs** requests (via **Curl** build into the servertask)
  - **JSON** parsing results (native in the servertask) and make results available to @forumulas
- @Formulas for different steps of the operation including lookups
  - Most flows should need to be that complicated flows
- **Add/Delete** operation for **DNS TXT** flows

DNS Provider Configuration

- Designed to be shared
- Code & documentation in one document
- DXL Export/Import to share
- Meta information for references, version author



DNS Provider Configuration

Basic

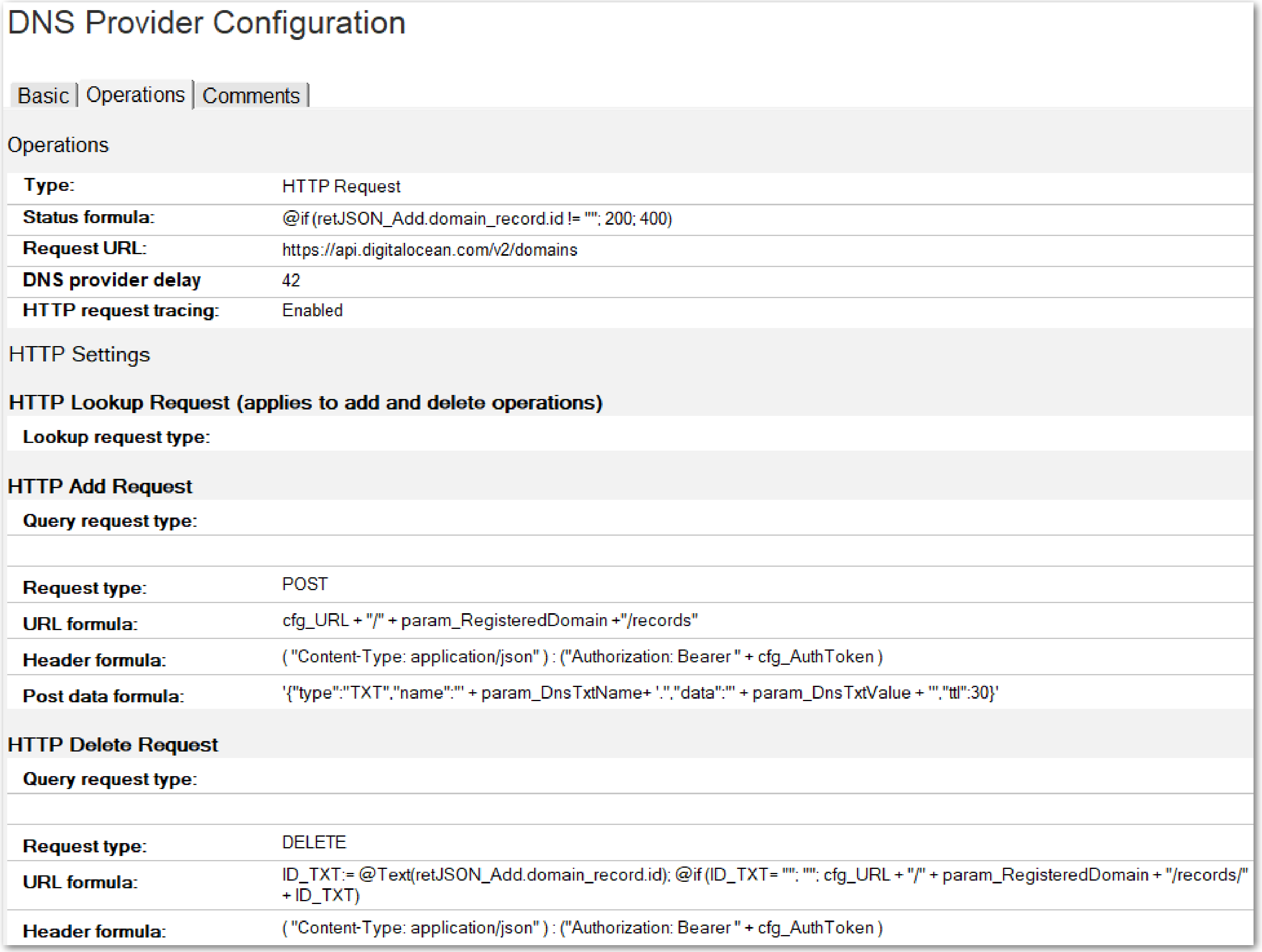
Configuration name:	DigitalOcean
Provider name:	DigitalOcean, LLC

Reference

Website:	https://cloud.digitalocean.com
Documentation URL:	https://developers.digitalocean.com/documentation/v2/
Version:	1.0
Author:	Daniel Nashed
Reference URL:	
Documentation:	Digital Ocean provides a free basic DNS service, which can sub-domain to their DNS server The API is very clean and is a good example how to implem The only parameters you have to define are the sub-domain

Example: Digital Ocean

- **Add** and **delete** operation share the same data in one result document
- Define HTTP request types, URL, Header and Post data via @formula
- Use fields from configuration and also parameters passed by CertMgr
- Results from JSON output can be used in standard fields
- Status formula to check if request was successful
  - Uses HTTP status code syntax:
  - 2xx = OK



DNS Provider Configuration

Basic | Operations | Comments |

Operations

Type:	HTTP Request
Status formula:	@if (retJSON_Add.domain_record.id != ""; 200; 400)
Request URL:	https://api.digitalocean.com/v2/domains
DNS provider delay	42
HTTP request tracing:	Enabled

HTTP Settings

HTTP Lookup Request (applies to add and delete operations)

Lookup request type:	

HTTP Add Request

Query request type:	
Request type:	POST
URL formula:	cfg_URL + "/" + param_RegisteredDomain +"/records"
Header formula:	("Content-Type: application/json") : ("Authorization: Bearer " + cfg_AuthToken)
Post data formula:	'{"type":"TXT","name":"' + param_DnsTxtName+ '.","data":"' + param_DnsTxtValue + '","ttl":30}'

HTTP Delete Request

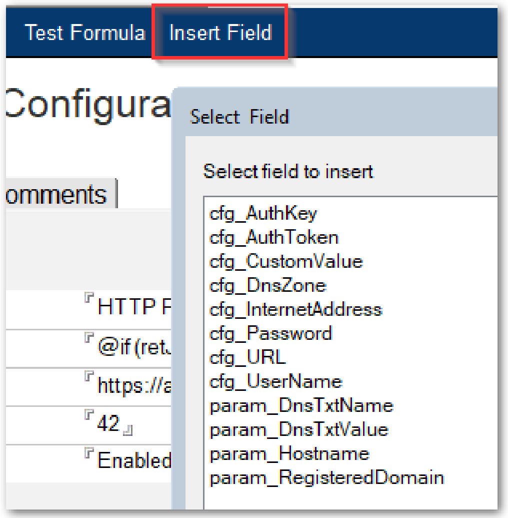
Query request type:	
Request type:	DELETE
URL formula:	ID_TXT:= @Text(retJSON_Add.domain_record.id); @if (ID_TXT= ""; ""; cfg_URL + "/" + param_RegisteredDomain + "/records/" + ID_TXT)
Header formula:	("Content-Type: application/json") : ("Authorization: Bearer " + cfg_AuthToken)

HCL Software Academy for Digital Solutions

HCL SOFTWARE

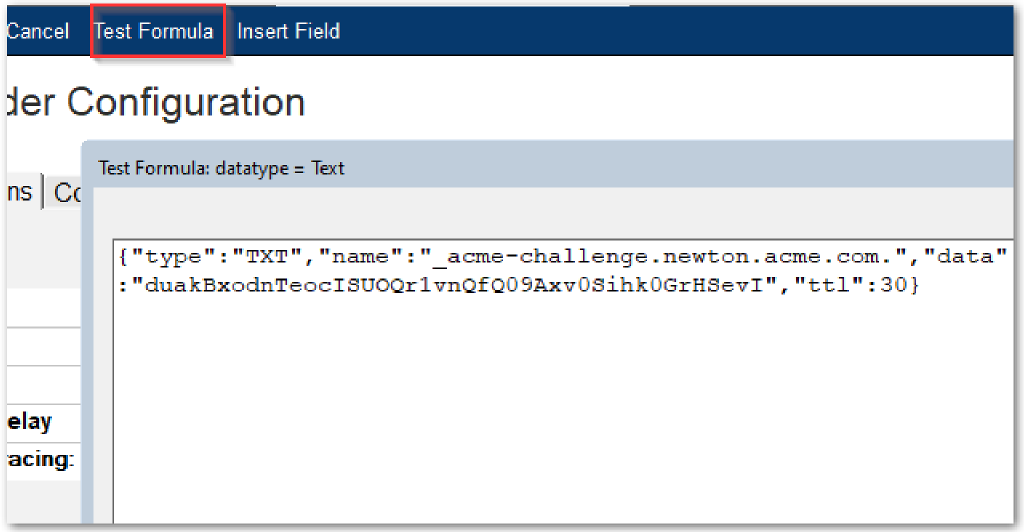
Build-in Developer Support

- Insert configuration fields and parameters into formulas
- Test formulas with sample data
  - Results can be copied and modified
- Should make it a lot easier to implement your own formulas
- Request Trace helps during implementation



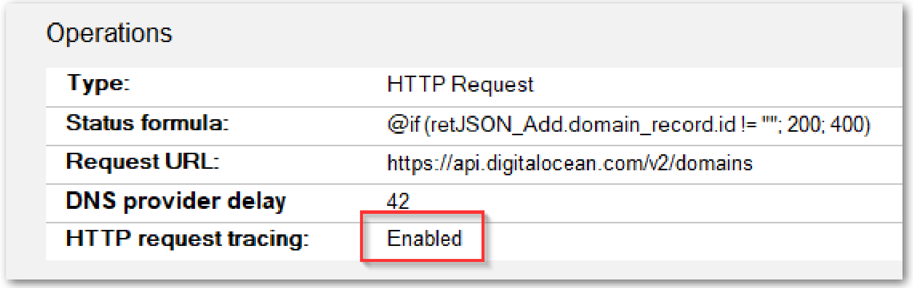
Select field to insert

cfg_AuthKey
cfg_AuthToken
cfg_CustomValue
cfg_DnsZone
cfg_InternetAddress
cfg_Password
cfg_URL
cfg_UserName
param_DnsTxtName
param_DnsTxtValue
param_Hostname
param_RegisteredDomain



Test Formula: datatype = Text

```
{"type":"TXT","name":"_acme-challenge.newton.acme.com.","data"
:"duakBxodnTeocISUOQr1vnQfQ09Axv0Sihk0GrHSevI","ttl":30}
```



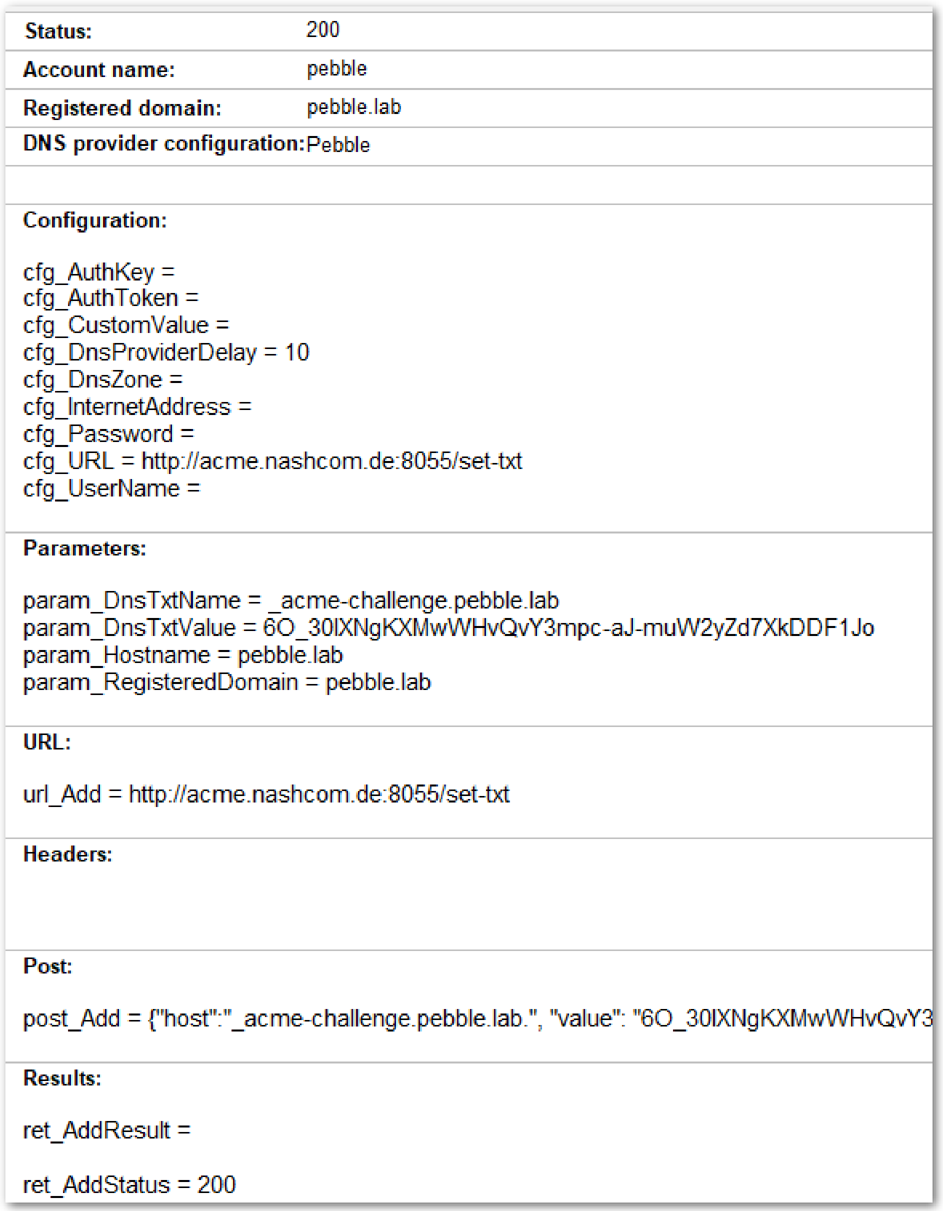
Operations

Type:	HTTP Request
Status formula:	@if (retJSON_Add.domain_record.id != ""; 200; 400)
Request URL:	https://api.digitalocean.com/v2/domains
DNS provider delay	42
HTTP request tracing:	Enabled

HCL Software Academy for Digital Solutions HCL SOFTWARE

Build-in Request Tracing

- Enabled either for all operations or **just on error**
- Shows all details about requests and results
- Documents are stored in certstore.nsf / DNS Trace Logs
- JSON fields and other results can be copied back into formulas



Status:	200
Account name:	pebble
Registered domain:	pebble.lab
DNS provider configuration:	Pebble

Configuration:

```
cfg_AuthKey =
cfg_AuthToken =
cfg_CustomValue =
cfg_DnsProviderDelay = 10
cfg_DnsZone =
cfg_InternetAddress =
cfg_Password =
cfg_URL = http://acme.nashcom.de:8055/set-txt
cfg_UserName =
```

Parameters:

```
param_DnsTxtName = _acme-challenge.pebble.lab
param_DnsTxtValue = 6O_30lXNgKXMwWHvQvY3mpc-aJ-muW2yZd7XkDDF1Jo
param_Hostname = pebble.lab
param_RegisteredDomain = pebble.lab
```

URL:

```
url_Add = http://acme.nashcom.de:8055/set-txt
```

Headers:

Post:

```
post_Add = {"host":"_acme-challenge.pebble.lab.", "value": "6O_30lXNgKXMwWHvQvY3
```

Results:

```
ret_AddResult =

ret_AddStatus = 200
```

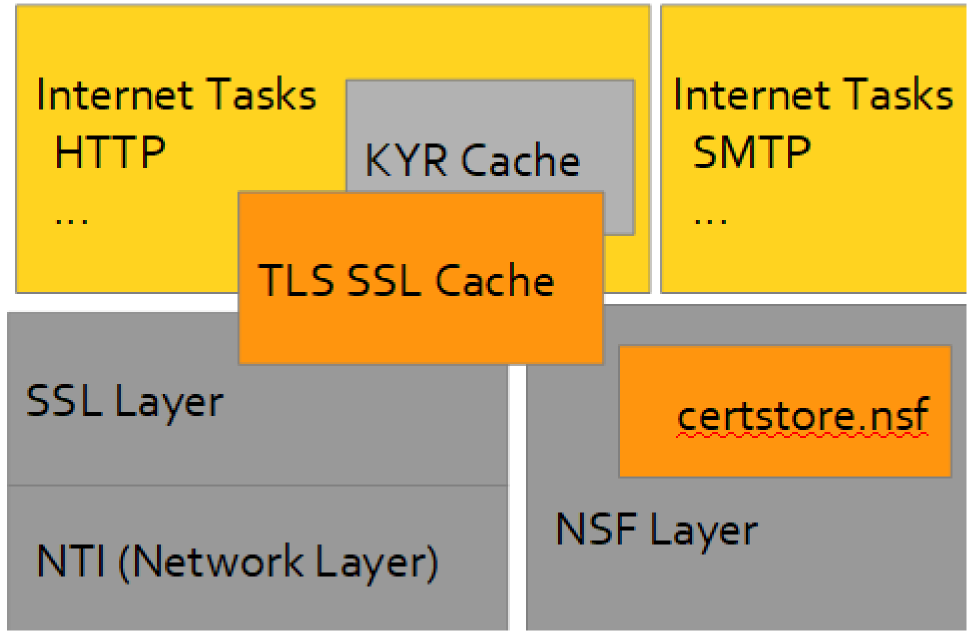
TLS Credentials Cache

New cache for RSA and ECDSA TLS Credentials

Mapping keyfiles for Internet Sites

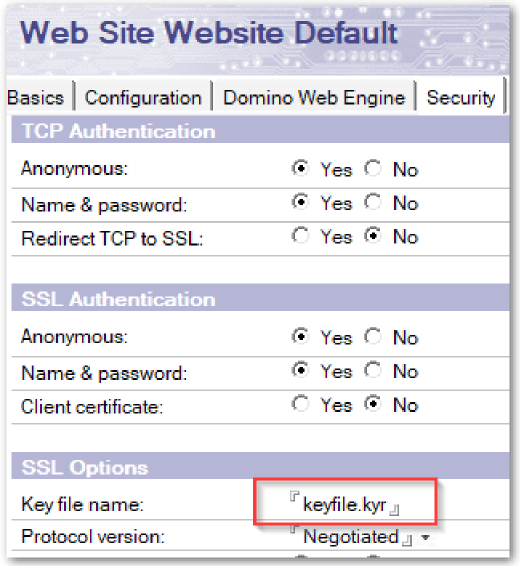
New TLS Cache

- ***.kyr** files have been managed by the KYR-Cache reading *.kyr files from **disk**
- New TLS Cache reads TLS Credentials directly from **certstore.nsf**
- TLS Cache sits in the SSL layer below internet protocols processes (e.g. HTTP/SMTP)
- Support for RSA and ECDSA keys in parallel
- Support for wildcard certificate lookups
- Automatic on the fly certificate reload
  - when added or updated
- Also manages trusted roots & OCSP cache



Keyfile name field is still very important!

- The keyfile name in server document and internet site is still triggering SSL
- Defines the default TLS Credential for the server
- Also used when server acts like a client (e.g. outgoing secure SMTP)
- Best practice:
  - Specify Domino server's host name you have a certificate for
  - Or specify **keyfile.kyr** in server document / internet site document
  - Have "**keyfile.kyr**" in the default TLS Credentials document tagging an RSA key as the default
  - Not only for HTTP -- Important for SMTP, LDAP, POP3, IMAP


Web Site Website Default

TCP Authentication

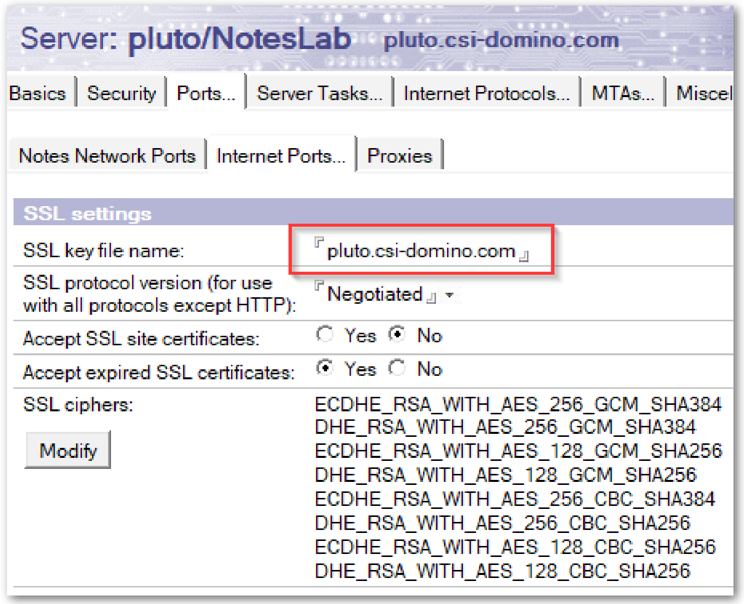
Anonymous:	[✓ Yes] No
Name & password:	[✓ Yes] No
Redirect TCP to SSL:	Yes [✓ No]

SSL Authentication

Anonymous:	[✓ Yes] No
Name & password:	[✓ Yes] No
Client certificate:	Yes [✓ No]

SSL Options

Key file name:	keyfile.kyr
Protocol version:	Negotiated


Server: pluto/NotesLab pluto.csi-domino.com

SSL settings

SSL key file name:	pluto.csi-domino.com
SSL protocol version (for use with all protocols except HTTP):	Negotiated
Accept SSL site certificates:	Yes [✓ No]
Accept expired SSL certificates:	[✓ Yes] No
SSL ciphers:	ECDHE_RSA_WITH_AES_256_GCM_SHA384 DHE_RSA_WITH_AES_256_GCM_SHA384 ECDHE_RSA_WITH_AES_128_GCM_SHA256 DHE_RSA_WITH_AES_128_GCM_SHA256 ECDHE_RSA_WITH_AES_256_CBC_SHA384 DHE_RSA_WITH_AES_256_CBC_SHA256 ECDHE_RSA_WITH_AES_128_CBC_SHA256 DHE_RSA_WITH_AES_128_CBC_SHA256

Demo Time

- First Steps
- Manual Flows
- ACME Flows
- Integrating with DNS TXT Providers

Known Limitations

Private Key Import and Export

TLS Credentials cannot be exported

- The private key of a TLS Credentials document is encrypted for security reasons
  - Only CertMgr and “Servers with access:” can decrypt the key
- There is currently no option to export the private key
  - A secure export is discussed for a future iteration
- Work-around
  - For other Domains
    - Copy server document into directory (or a DA directory) and encrypt for the server
    - Add the server to “Servers with access:”
- Create the key outside and import the key to be used for manual or ACME flows
  - As long as the key stays the same, the certificate can be merged with an existing key

Private keys cannot be imported via UI

- Most import operations would involve copy & paste or similar to transfer a private key
  - By design today the only options to import private keys are available via server console

- Work around : Current import / migration options

  - **load certmgr -importkyr keyring.kyr**
    - Import one KeyRing file (*.kyr) to CertStore

  - **load certmgr -importkyr all**
    - Import all KeyRing files referenced from server doc & internet sites to CertStore

  - **load certmgr -importpem <file>**
  - Import file containing PEM encoded certificate chain and keys to CertStore

Troubleshooting

Settings, Logging, Debug Options

Global Settings

- Mainly used to set defaults for important settings
  - For example: key type, key size, default ACME account and renewal interval

- Admin Server for CertMgr
  - Should <u>not</u> be changed in global configuration document!
  - Admin server is also stored in Directory profile “CertMgrServer” to publish in the domain

- Changing the admin server involves re-encryption of keys

- Migration via command-line option <u>only</u>!
  - Load certmgr **-MIGRATETOSERVER** server-name
  - Re-encrypts all private keys after checking all keys can be read

CertMgr Commands

- **Tell certmgr process**
  - Skips the wait time between requests

- Tip to reduce the interval for testing

  → notes.ini **CERTMGR_INTERVAL=2** (default: 30 seconds)

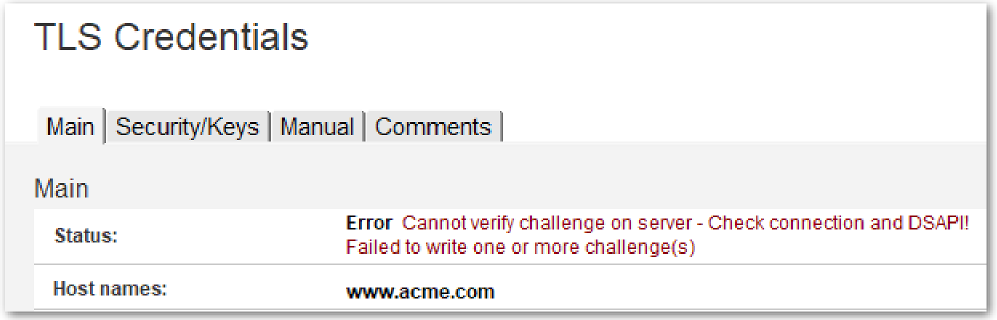
- **Tell certmgr shutdown**
  - Waits until a running request is terminated and stops cleanly
  - Recommended shutdown option
  - Usually not a problem because of the small volume of operations during a day

Additional Notes.ini Parameter

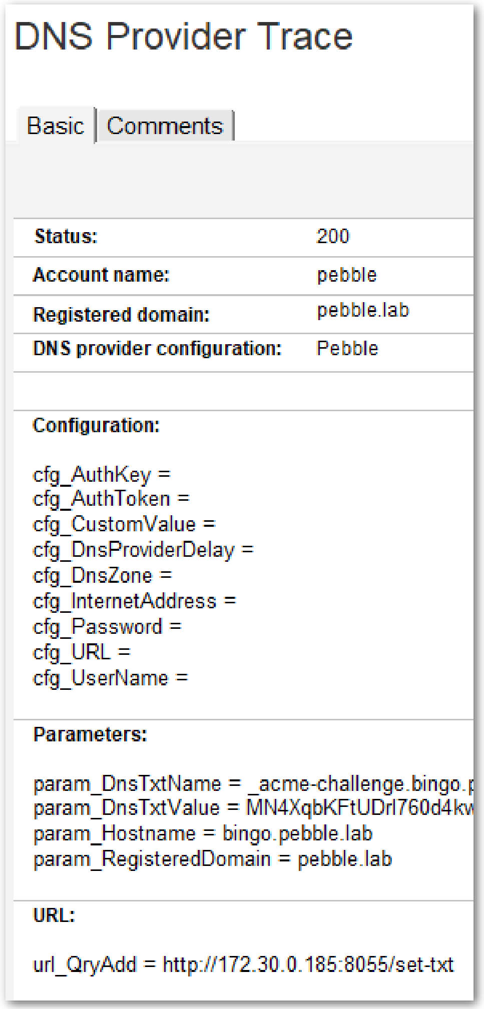
- **CertMgr_ReplicationInterval=n**
  - Default: 120 sec
  - Used for client mode
- **CertMgr_HealthCheckInterval=n**
  - Default 30 minutes
- **CertMgr_CompactFreeSpace=n**
  - Default: 50%, Compacts database when specified percentage is free
- **CertMgr_CompactDays=n**
  - Default: 30, Compacts database when not compacted since number of days
- **CertMgr_ACCEPT_TOU=1**
  - Same as command-line option to confirm ACME provider terms of use – useful for automation

Common Issues & Tracing

- ACME license terms not accepted
- DSAPI Filter not configured
  - Check server document / internet site
- **Port 80 cannot be reached – DNS or Firewall issue**
- Most errors are already visible in TLS Credentials document
  - More detailed information can be found in debug logs if enabled
- DNS-TXT provider cannot be reached or configuration problem
  - **DNS provider trace** should be set to error logging by default in provider config



Status:	Error Cannot verify challenge on server - Check connection and DSAPI! Failed to write one or more challenge(s)
Host names:	www.acme.com



Status:	200
Account name:	pebble
Registered domain:	pebble.lab
DNS provider configuration:	Pebble

Configuration:

```
cfg_AuthKey =
cfg_AuthToken =
cfg_CustomValue =
cfg_DnsProviderDelay =
cfg_DnsZone =
cfg_InternetAddress =
cfg_Password =
cfg_URL =
cfg_UserName =
```

Parameters:

```
param_DnsTxtName = _acme-challenge.bingo.p
param_DnsTxtValue = MN4XqbKFtUDrl760d4kw
param_Hostname = bingo.pebble.lab
param_RegisteredDomain = pebble.lab
```

URL:

```
url_QryAdd = http://172.30.0.185:8055/set-txt
```

Debugging and Troubleshooting Command Line

- **-v** = Verbose logging (log.nsf)
- **-d** = Debug mode
  - IBM_TECHNICAL_SUPPORT/certmgr_debug_[..].log
- **-l** = Log all Curl I/O to file
  - IBM_TECHNICAL_SUPPORT/certmgr_curl__[..].log
- **-z** = Connectivity test: Just get the ACME directory URLs and terminate
  - Useful for testing internet connectivity
- Example: load certmgr -d -l

TLS Cache Logging and Debugging

- **CERTSTORE_CACHELOG=1**
  - Recommended Starting point for all troubleshooting
  - Logs most important events only
- **CERTSTORE_CACHELOG=2**
  - Very detailed logging → Debug mode
- **DEBUG_SSL_TLSCACHE=1 *)**
  - Debug SSL side of TLS Cache
- **DEBUG_SSL_KYRCACHE=2 *)**
- Debug SSL for old KYR Cache

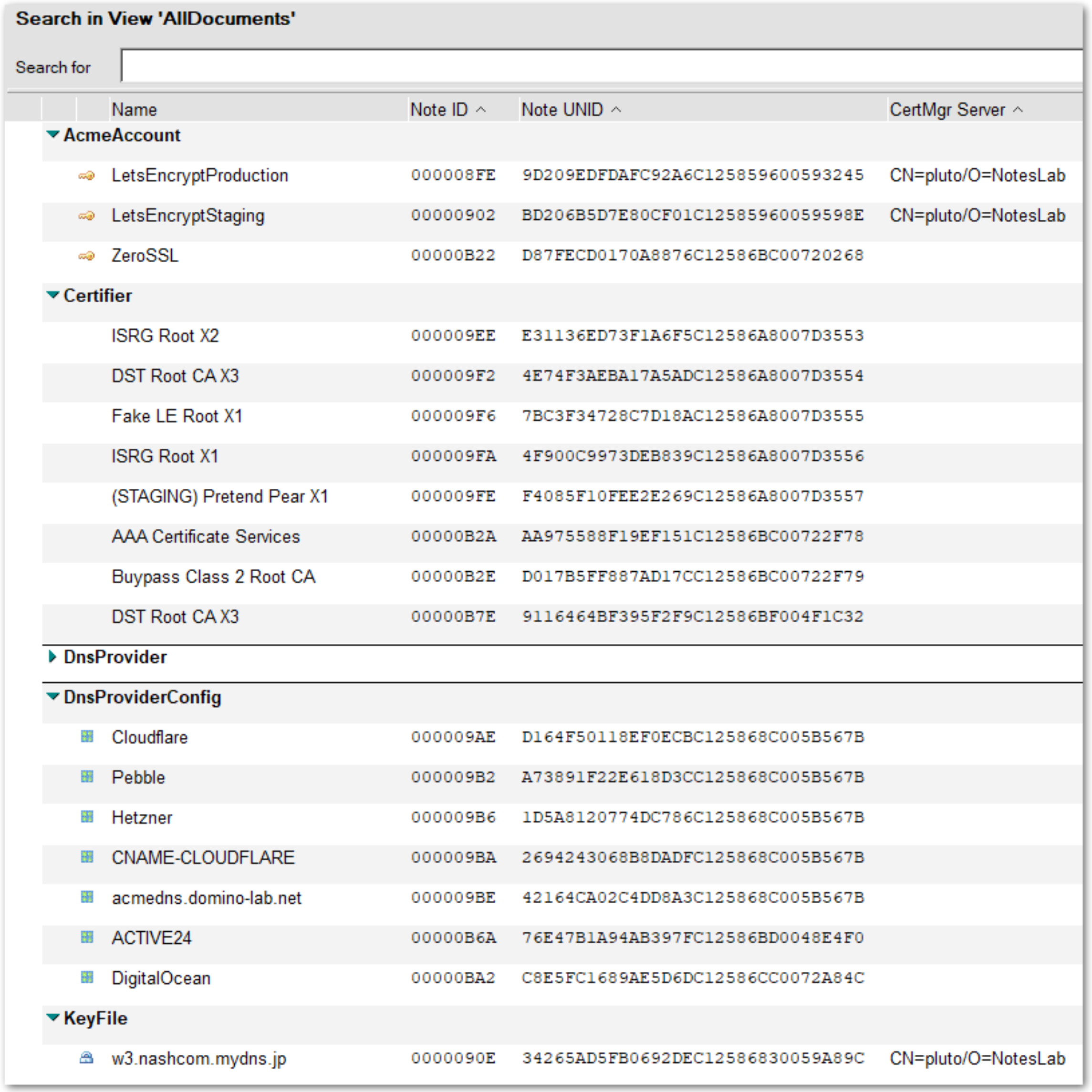
*) Task restart needed

DSAPI Debug Notes.ini Parameter

- DSAPI has no separate log file option
- Logging and debugging can be used to trace inbound challenge requests
- Notes.ini
  - **CERTMGR_DSAPIDEBUG=1**
  - **CERTMGR_DSAPIVERBOSE=1**
- Requires HTTP task restart (restart task http)

Hidden AllDocuments View

- Open via **CTRL+Shift → View → Goto**
- This view contains all documents by form
- CertMgr Server is listed for all documents encrypted
- Secondary sort column by NoteID and NoteUNID
  - Find documents listed in low level error messages



Search in View 'AllDocuments'

Search for

Name	Note ID ^	Note UNID ^	CertMgr Server ^
AcmeAccount			
LetsEncryptProduction	000008FE	9D209EDFDAFC92A6C125859600593245	CN=pluto/O=NotesLab
LetsEncryptStaging	00000902	BD206B5D7E80CF01C12585960059598E	CN=pluto/O=NotesLab
ZeroSSL	00000B22	D87FECD0170A8876C12586BC00720268	
Certifier			
ISRG Root X2	000009EE	E31136ED73F1A6F5C12586A8007D3553	
DST Root CA X3	000009F2	4E74F3AEBA17A5ADC12586A8007D3554	
Fake LE Root X1	000009F6	7BC3F34728C7D18AC12586A8007D3555	
ISRG Root X1	000009FA	4F900C9973DEB839C12586A8007D3556	
(STAGING) Pretend Pear X1	000009FE	F4085F10FEE2E269C12586A8007D3557	
AAA Certificate Services	00000B2A	AA975588F19EF151C12586BC00722F78	
Buypass Class 2 Root CA	00000B2E	D017B5FF887AD17CC12586BC00722F79	
DST Root CA X3	00000B7E	9116464BF395F2F9C12586BF004F1C32	
DnsProvider			
DnsProviderConfig			
Cloudflare	000009AE	D164F50118EF0ECBC125868C005B567B	
Pebble	000009B2	A73891F22E618D3CC125868C005B567B	
Hetzner	000009B6	1D5A8120774DC786C125868C005B567B	
CNAME-CLOUDFLARE	000009BA	2694243068B8DADFC125868C005B567B	
acmedns.domino-lab.net	000009BE	42164CA02C4DD8A3C125868C005B567B	
ACTIVE24	00000B6A	76E47B1A94AB397FC12586BD0048E4F0	
DigitalOcean	00000BA2	C8E5FC1689AE5D6DC12586CC0072A84C	
KeyFile			
w3.nashcom.mydns.jp	0000090E	34265AD5FB0692DEC12586830059A89C	CN=pluto/O=NotesLab

HCL Software Academy for Digital Solutions

HCL SOFTWARE

HCL Domino CertMgr & certstore.nsf

Q&A

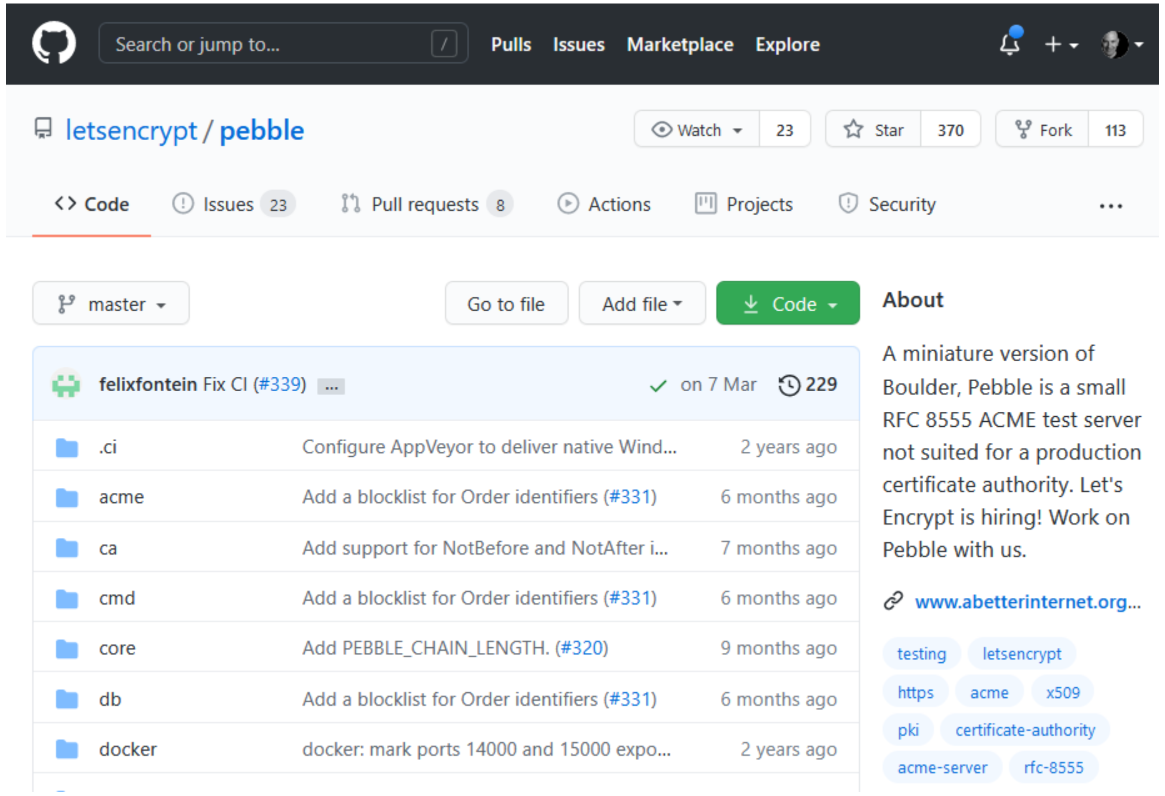
Building a Lab Environment

Domino on Docker with Let’s Encrypt Pebble

Build your own Lab Environment

- Challenging in internal lab environments
  - **HTTP-01** → Inbound Internet connections are difficult in internal test environments
  - **DNS-01** → Requires registered DNS domain and DNS TXT API integration

- Solution
  - Docker based lab environment
  - Very easy to setup via docker-compose
  - Supports HTTP-01 & DNS-01 challenges
  - Logging allows to trace and understand
  - We are using it also for automation testing



felixfontein Fix CI (#339) ... ✓ on 7 Mar 229

.ci	Configure AppVeyor to deliver native Wind...	2 years ago
acme	Add a blocklist for Order identifiers (#331)	6 months ago
ca	Add support for NotBefore and NotAfter i...	7 months ago
cmd	Add a blocklist for Order identifiers (#331)	6 months ago
core	Add PEBBLE_CHAIN_LENGTH. (#320)	9 months ago
db	Add a blocklist for Order identifiers (#331)	6 months ago
docker	docker: mark ports 14000 and 15000 expo...	2 years ago

About

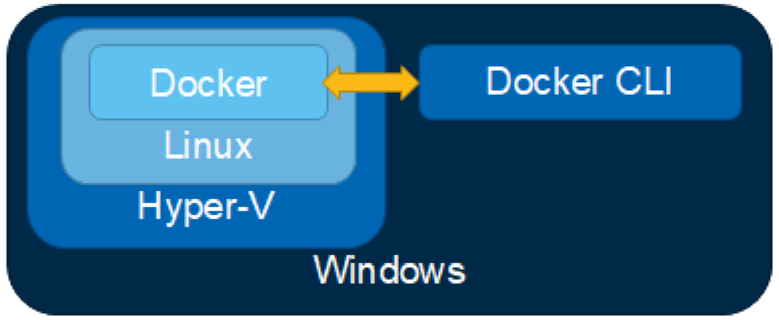
A miniature version of Boulder, Pebble is a small RFC 8555 ACME test server not suited for a production certificate authority. Let's Encrypt is hiring! Work on Pebble with us.

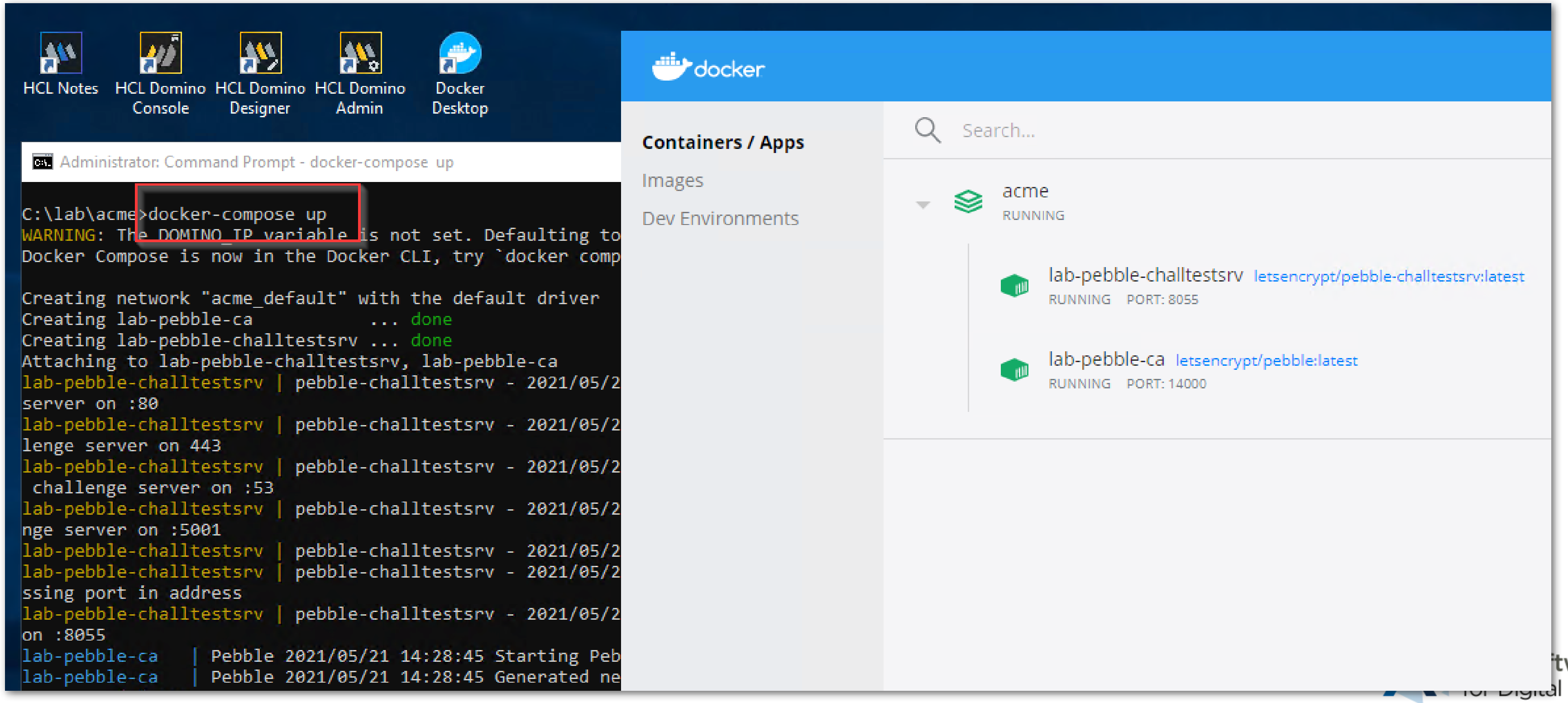
www.abetterinternet.org...

testing letsencrypt https acme x509 pki certificate-authority acme-server rfc-8555

Docker Desktop Lab Environment on Windows

- Very simple to setup and use environment
  - Runs unmodified with default settings
  - Components are reachable via **127.0.0.1**
  - Just switch to the right directory and run “**docker-compose up**”





Administrator: Command Prompt - docker-compose up

```
C:\lab\acme>docker-compose up
WARNING: The DOMINO_IP variable is not set. Defaulting to
Docker Compose is now in the Docker CLI, try `docker comp

Creating network "acme_default" with the default driver
Creating lab-pebble-ca           ... done
Creating lab-pebble-challtestsrv ... done
Attaching to lab-pebble-challtestsrv, lab-pebble-ca
lab-pebble-challtestsrv | pebble-challtestsrv - 2021/05/2
server on :80
lab-pebble-challtestsrv | pebble-challtestsrv - 2021/05/2
lenge server on 443
lab-pebble-challtestsrv | pebble-challtestsrv - 2021/05/2
 challenge server on :53
lab-pebble-challtestsrv | pebble-challtestsrv - 2021/05/2
nge server on :5001
lab-pebble-challtestsrv | pebble-challtestsrv - 2021/05/2
lab-pebble-challtestsrv | pebble-challtestsrv - 2021/05/2
ssing port in address
lab-pebble-challtestsrv | pebble-challtestsrv - 2021/05/2
on :8055
lab-pebble-ca    | Pebble 2021/05/21 14:28:45 Starting Peb
lab-pebble-ca    | Pebble 2021/05/21 14:28:45 Generated ne
```

acme
RUNNING

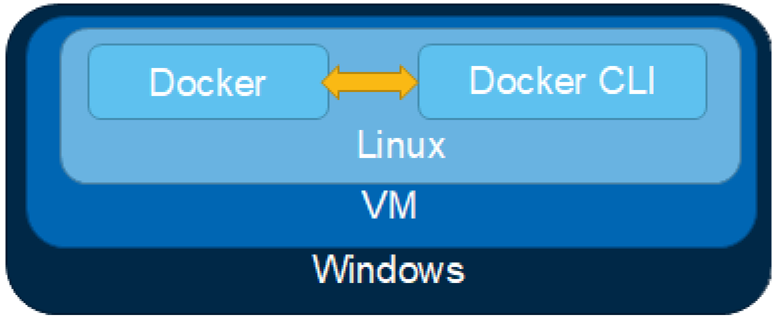
lab-pebble-challtestsrv letsencrypt/pebble-challtestsrv:latest
RUNNING PORT: 8055

lab-pebble-ca letsencrypt/pebble:latest
RUNNING PORT: 14000

Docker Server Lab Environment on Linux

- Docker is installed on a Linux host running in VM (Hyper-V or Virtualbox)

- More complex
  - No 127.0.0.1 addresses can be used
  - Needs configuration changes to map the right IP addresses inside VM / Docker
  - Networking on virtual machines can be complex and tricky
  - Virtualbox has some network limitations! → https://www.virtualbox.org/manual/ch06.html



- Starting fresh consider Docker Desktop
  - But if you have an existing environment
  - you are already experienced with VMs

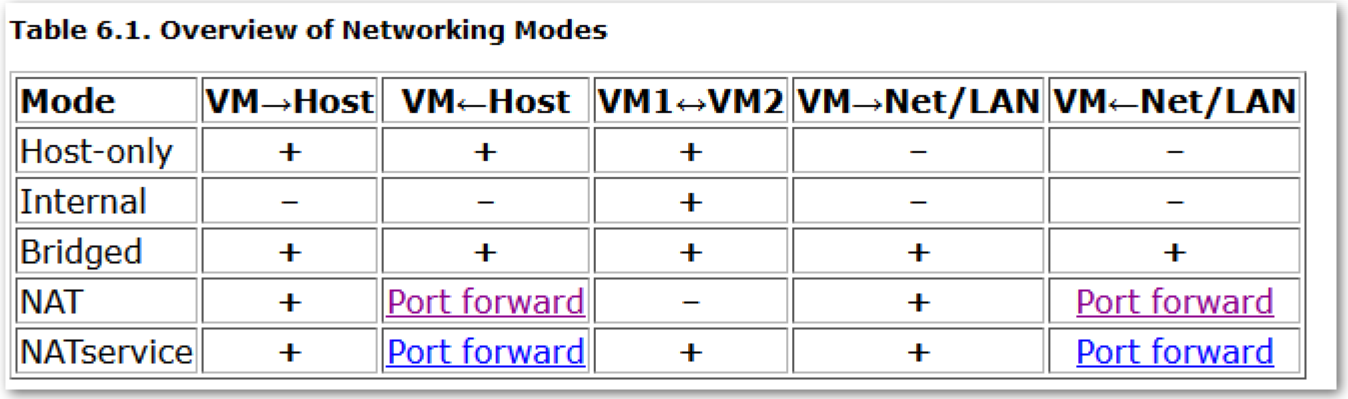
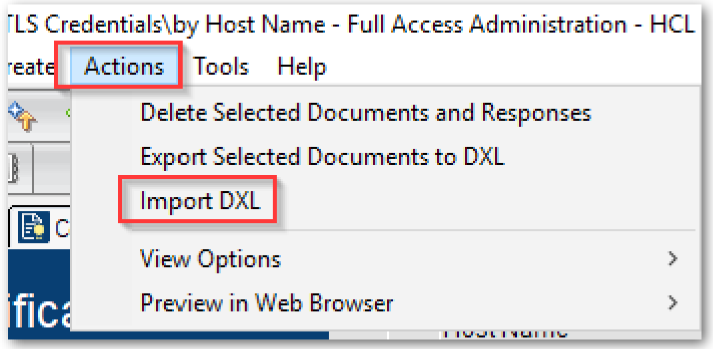


Table 6.1. Overview of Networking Modes

Mode	VM→Host	VM←Host	VM1↔VM2	VM→Net/LAN	VM←Net/LAN
Host-only	+	+	+	–	–
Internal	–	–	+	–	–
Bridged	+	+	+	+	+
NAT	+	Port forward	–	+	Port forward
NATservice	+	Port forward	+	+	Port forward

Import Lab Configuration

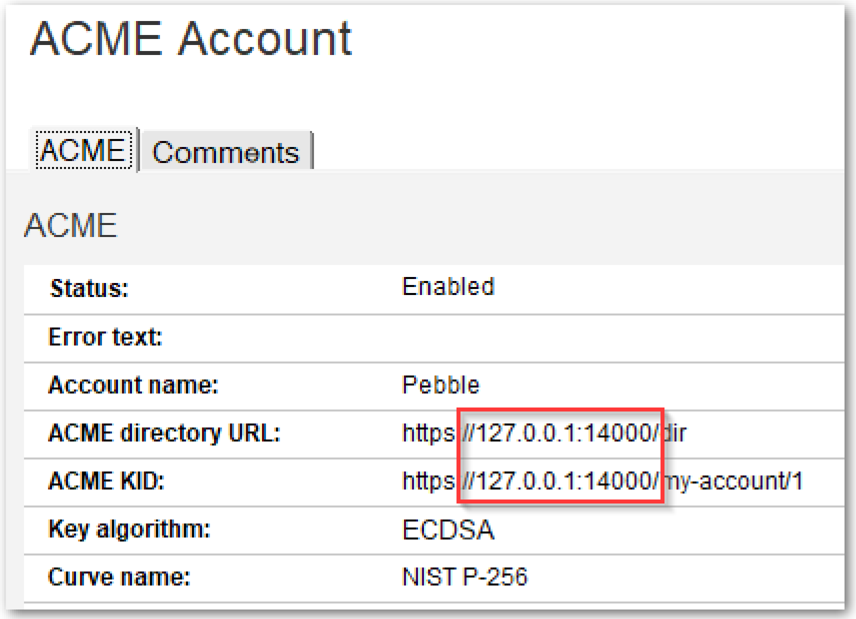
- The lab configuration is prepared in a DXL file
- You can just import it into the database
  - ACME Account
  - DNS Provider Configuration


TLS Credentials\by Host Name - Full Access Administration - HCL
reate
Actions
Tools
Help
Delete Selected Documents and Responses
Export Selected Documents to DXL
Import DXL
View Options
Preview in Web Browser

- The default will work for the Docker Desktop environment
- You need to change IP addresses matching the Docker IP address/hostname

Update IP Address in certstore.nsf

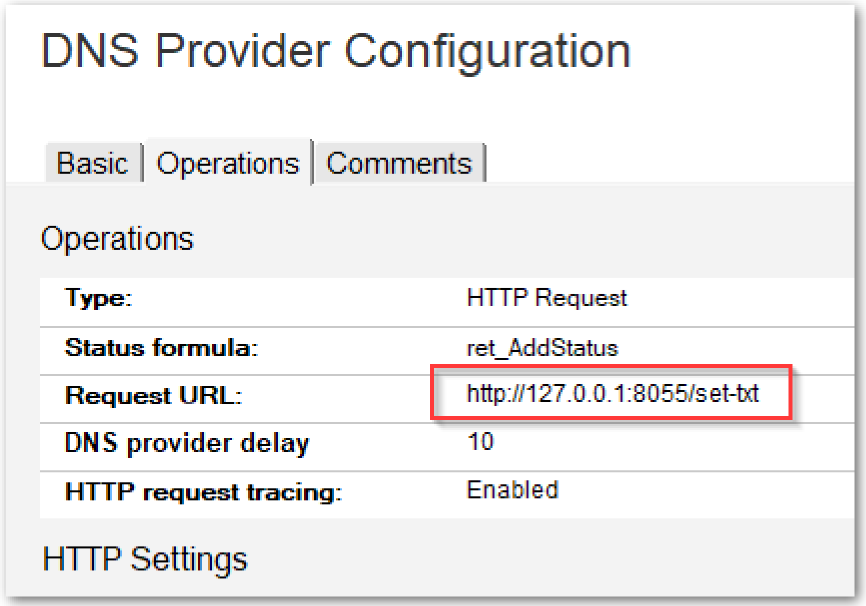
- The configuration is prepared for Docker Desktop
  - In case of a Linux Docker scenario you have to update the
  - IP address to reflect the correct address inside the VM



ACME Account

ACME

Status:	Enabled
Error text:	
Account name:	Pebble
ACME directory URL:	https://127.0.0.1:14000/dir
ACME KID:	https://127.0.0.1:14000/my-account/1
Key algorithm:	ECDSA
Curve name:	NIST P-256



DNS Provider Configuration

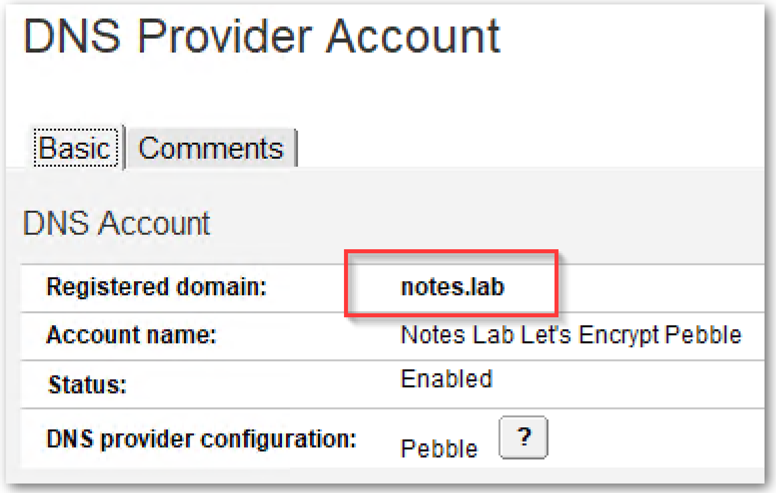
Operations

Type:	HTTP Request
Status formula:	ret_AddStatus
Request URL:	http://127.0.0.1:8055/set-txt
DNS provider delay	10
HTTP request tracing:	Enabled

HTTP Settings

DNS Provider Account Customization

- The DNS provider account is the trigger for DNS-01 challenges
- The lab environment allows to work with any domain
- The registered domain is used as a trigger
  - You can change the domain, because the challenge server allows accepts all DNS TXT records for this Pebble Lab server
- You can create multiple documents pointing to the same DNS provider configuration



Registered domain:	notes.lab
Account name:	Notes Lab Let's Encrypt Pebble
Status:	Enabled
DNS provider configuration:	Pebble

Pebble Tips

- Import trusted root from Pebble server
  - e.g **curl -k https://127.0.0.1:15000/roots/0**
- Pebble is designed for test and does not to store data permanently
  - Docker container has no volume
  - Root certificate and ACME account needs reset every time Pebble is restarted
- After restart you need to reset account
  - Remove “**ACME KID:**”
  - You will see error messages reminding you ;-)

HCL

Relationship™

BEYOND THE CONTRACT

$8.4 BILLION ENTERPRISE | **132,000** IDEAPRENEURS | **44** COUNTRIES

WATCH THE **FILM**