HCLSoftware

Fueling the **DIGITAL+** economy.

HCLSoftware

Unleash the Power of OTS (One Touch Setup)

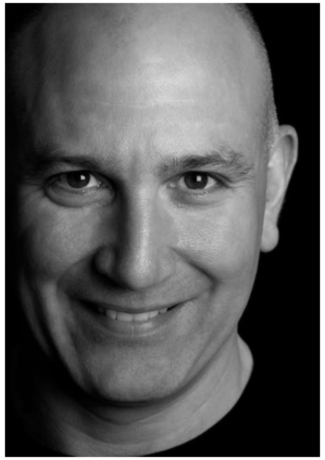
OpenNTF, 26. September 2024

HCLSoftware
HCL Ambassador
LIFETIME

Daniel Nashed -- https://blog.nashcom.de

Why do I speak about OTS?

- **Maintainer of Domino Start Script on GitHub**
  - The Domino Start Script provides full OTS integration
- **Maintainer of Domino Container Community project on GitHub**
  - The Domino container supports OTS since first code drop
- Using OTS for automation testing since available in first code drops
- One of the maintainers for the HCL OTS setup GitHub community project
  - https://github.com/HCL-TECH-SOFTWARE/domino-one-touch-setup
- Administrator, Developer, Architect & Troubleshooter
  - Providing “glue” to make OTS even more flexible



HCLSoftware
HCL Ambassador
LIFETIME

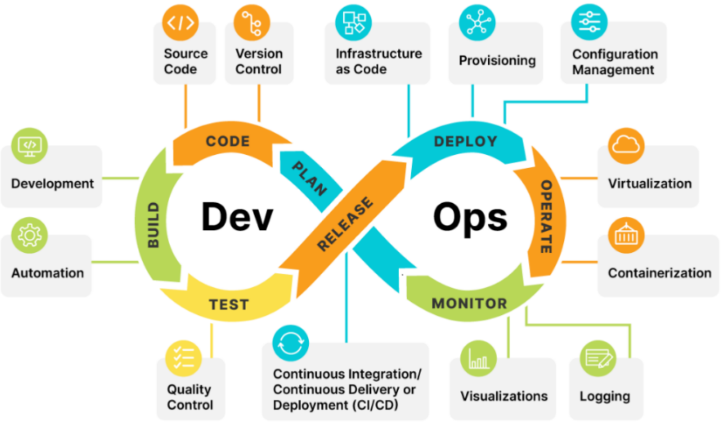
Nash!Com
Communication Systems

OTS in 90 minutes with 20 slides?

- **Understand OTS basics**
  - One Touch Setup - Environment variables vs. JSON file
  - How to invoke OTS, etc.
- **How to use OTS in real life**
  - Setup first and additional servers
  - One Touch “Templating”
- **Advanced Scenarios**
  - Configure complete servers
  - Running agents & more
- **Demos & Details**
  - More examples, useful resources and a surprise …

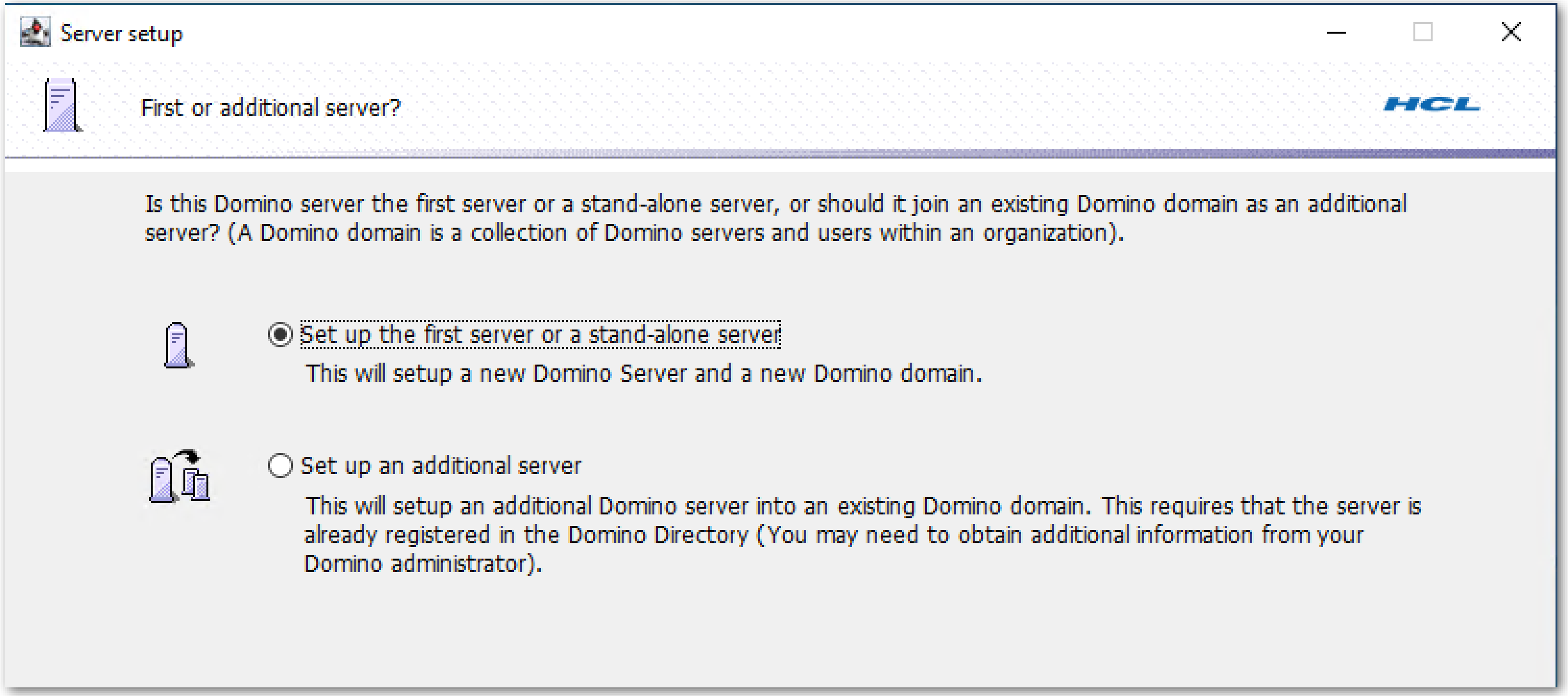
Why One Touch Setup?

- **In modern IT environments automatic installation & configuration plays an important role**
  - **DevOps** is a methodology in the software development and IT industry.
    Used as a set of practices and tools, DevOps **integrates** and **automates** the work of software **development** and IT **operations** as a means for improving and shortening the systems development life cycle.
- OTS helps with **production**, **test** and especially **automation test environments**
  - **Silent install** is used to install servers native or in containers
  - **OTS** is used to setup the installed server
- For HCL OTS plays an important role in automation testing
- It can also simplify normal deployments
- Helps to ensure to always have a standardized and reproducible setup
  - This isn't limited to container or K8s deployments and works with Windows as well!



Classical Domino Setup

- Java based GUI to configure Domino
- Allows PDS file based customization
  - Very limited configuration options. Not well documented
  - Used by the early IBM Domino Container image originally initiated by Thomas Hampel



First or additional server?

Is this Domino server the first server or a stand-alone server, or should it join an existing Domino domain as an additional server? (A Domino domain is a collection of Domino servers and users within an organization).

● Set up the first server or a stand-alone server

This will setup a new Domino Server and a new Domino domain.

○ Set up an additional server

This will setup an additional Domino server into an existing Domino domain. This requires that the server is already registered in the Domino Directory (You may need to obtain additional information from your Domino administrator).

Domino 12+ One Touch Setup

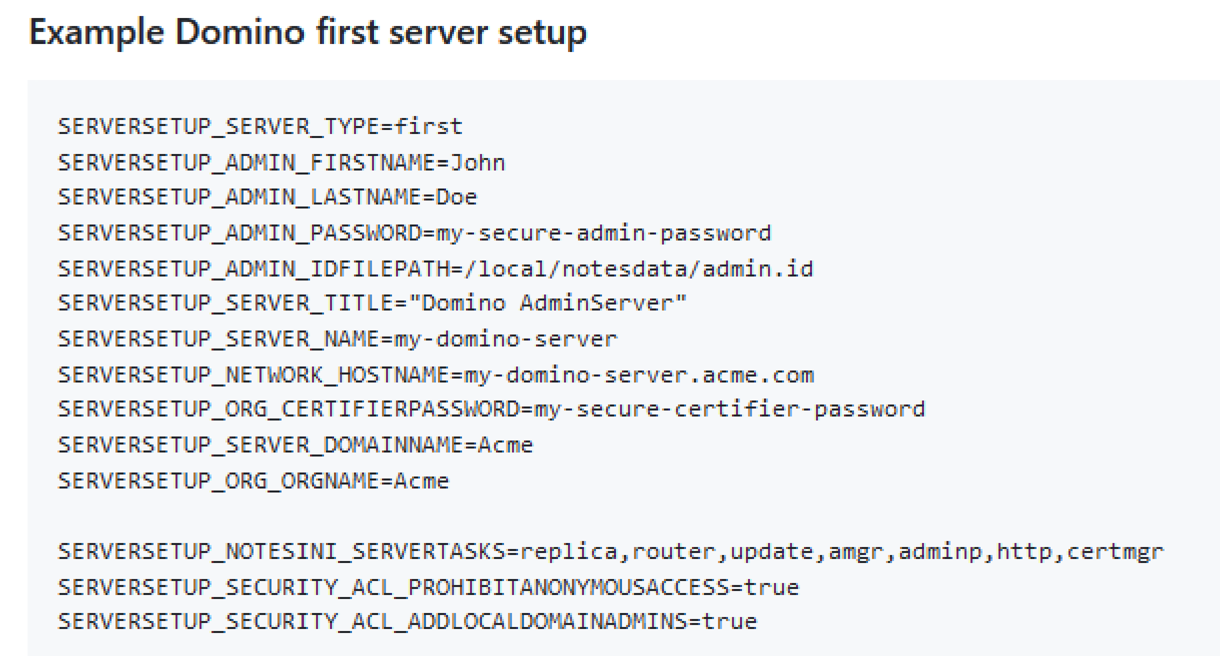
- **New way in Domino 12+ to automatically configure Domino**
  - Idea is based on the early Domino container image
- **Cross platform approach**
  - Implemented on top of the existing setup code
  - Available for Domino Containers and normal servers
- **Provides additional configuration options (only available via OTS)**
  - TLS Setup
  - Create an ID Vault
  - And more ..
- **First & Additional server setup**
- **Extended in every release**

One Touch Setup – Environment Variables

HCL Domino

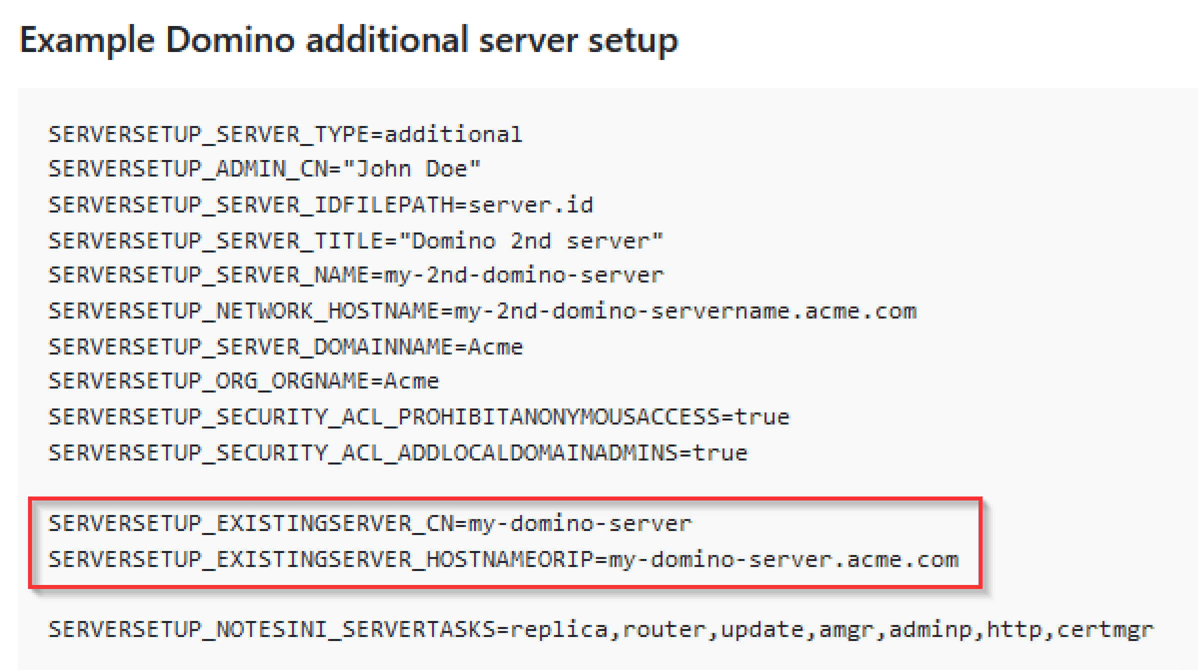
- **Simple configuration with environment variables**
  - Everything you know from the classical setup
  - Many additional parameters

- Triggered via **SetupAutoConfigure=1**



Example Domino first server setup

```
SERVERSETUP_SERVER_TYPE=first
SERVERSETUP_ADMIN_FIRSTNAME=John
SERVERSETUP_ADMIN_LASTNAME=Doe
SERVERSETUP_ADMIN_PASSWORD=my-secure-admin-password
SERVERSETUP_ADMIN_IDFILEPATH=/local/notesdata/admin.id
SERVERSETUP_SERVER_TITLE="Domino AdminServer"
SERVERSETUP_SERVER_NAME=my-domino-server
SERVERSETUP_NETWORK_HOSTNAME=my-domino-server.acme.com
SERVERSETUP_ORG_CERTIFIERPASSWORD=my-secure-certifier-password
SERVERSETUP_SERVER_DOMAINNAME=Acme
SERVERSETUP_ORG_ORGNAME=Acme

SERVERSETUP_NOTESINI_SERVERTASKS=replica,router,update,amgr,adminp,http,certmgr
SERVERSETUP_SECURITY_ACL_PROHIBITANONYMOUSACCESS=true
SERVERSETUP_SECURITY_ACL_ADDLOCALDOMAINADMINS=true
```

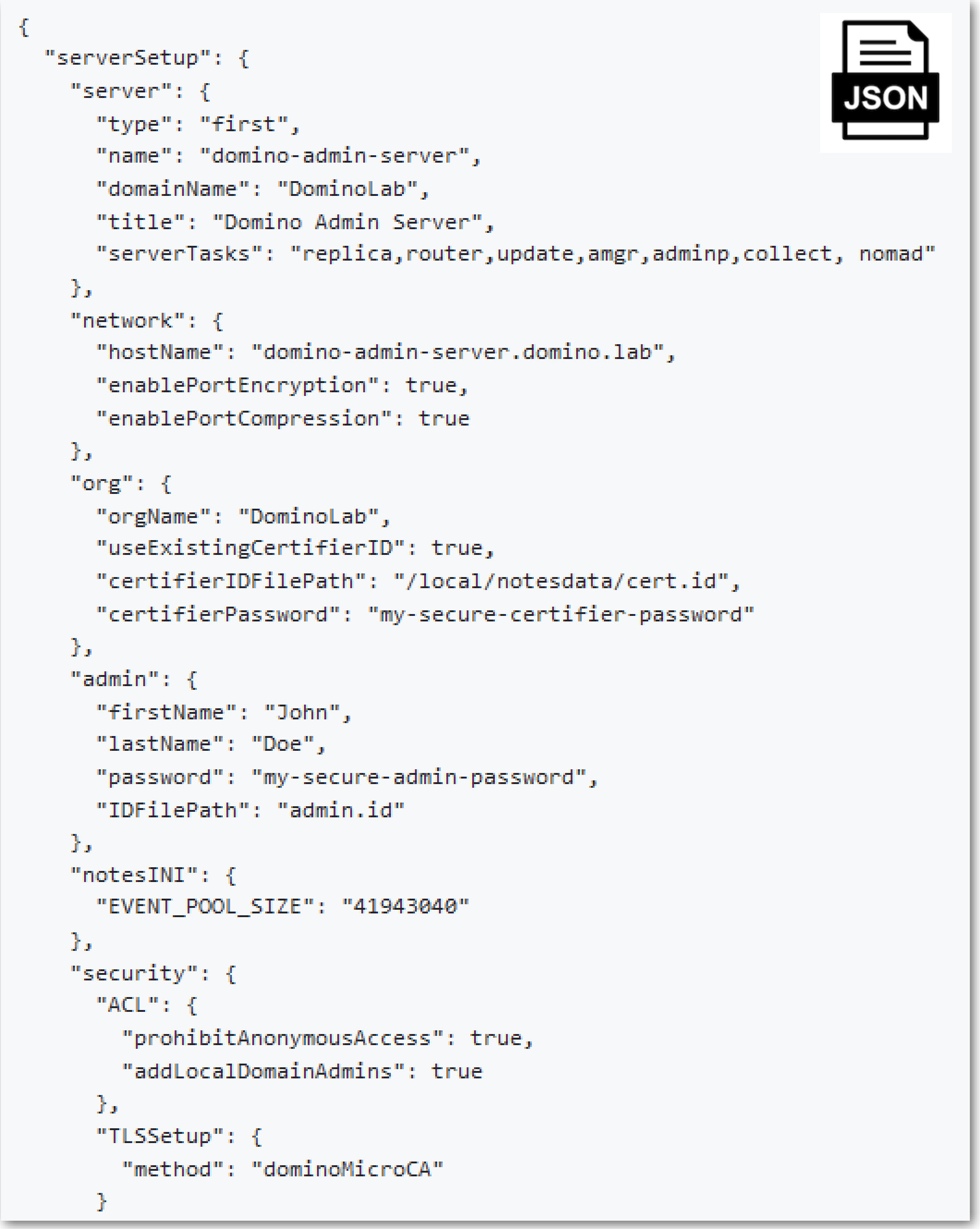


Example Domino additional server setup

```
SERVERSETUP_SERVER_TYPE=additional
SERVERSETUP_ADMIN_CN="John Doe"
SERVERSETUP_SERVER_IDFILEPATH=server.id
SERVERSETUP_SERVER_TITLE="Domino 2nd server"
SERVERSETUP_SERVER_NAME=my-2nd-domino-server
SERVERSETUP_NETWORK_HOSTNAME=my-2nd-domino-servername.acme.com
SERVERSETUP_SERVER_DOMAINNAME=Acme
SERVERSETUP_ORG_ORGNAME=Acme
SERVERSETUP_SECURITY_ACL_PROHIBITANONYMOUSACCESS=true
SERVERSETUP_SECURITY_ACL_ADDLOCALDOMAINADMINS=true

SERVERSETUP_EXISTINGSERVER_CN=my-domino-server
SERVERSETUP_EXISTINGSERVER_HOSTNAMEORIP=my-domino-server.acme.com

SERVERSETUP_NOTESINI_SERVERTASKS=replica,router,update,amgr,adminp,http,certmgr
```

One Touch Setup – JSON Files

- **More flexible and powerful functionality**
  - Everything you know from Environment Setup
  - More complex, but very powerful and flexible
- **Additional Application configuration**
  - Create databases
  - Add or Update documents
  - And much more!
- **JSON based approach**
  - A file containing all configuration specified at startup



```
{
  "serverSetup": {
    "server": {
      "type": "first",
      "name": "domino-admin-server",
      "domainName": "DominoLab",
      "title": "Domino Admin Server",
      "serverTasks": "replica,router,update,amgr,adminp,collect, nomad"
    },
    "network": {
      "hostName": "domino-admin-server.domino.lab",
      "enablePortEncryption": true,
      "enablePortCompression": true
    },
    "org": {
      "orgName": "DominoLab",
      "useExistingCertifierID": true,
      "certifierIDFilePath": "/local/notesdata/cert.id",
      "certifierPassword": "my-secure-certifier-password"
    },
    "admin": {
      "firstName": "John",
      "lastName": "Doe",
      "password": "my-secure-admin-password",
      "IDFilePath": "admin.id"
    },
    "notesINI": {
      "EVENT_POOL_SIZE": "41943040"
    },
    "security": {
      "ACL": {
        "prohibitAnonymousAccess": true,
        "addLocalDomainAdmins": true
      },
      "TLSSetup": {
        "method": "dominoMicroCA"
      }
```

{JSON} – A standard you should know

- https://www.json.org
- Most configuration files today are in JSON
  - It replaces XML in many cases
- **UTF-8 format** without a byte order mark (BOM)
  - Note: ASCII is a subset of UTF-8
- **JQ - The most popular parser on Linux & Windows**
  - Very powerful and included in most Linux distributions
  - https://stedolan.github.io/jq/
- **RapidJSON** is a popular C++ JSON implementation also used by Notes/Domino
  - https://rapidjson.org/

jq is a lightweight and flexible command-line JSON processor.

How to invoke OTS JSON Setup

- **Two different methods**
  - Depends on your use case

- **Invoke OTS JSON via command-line parameters**
  - Often used from command-line

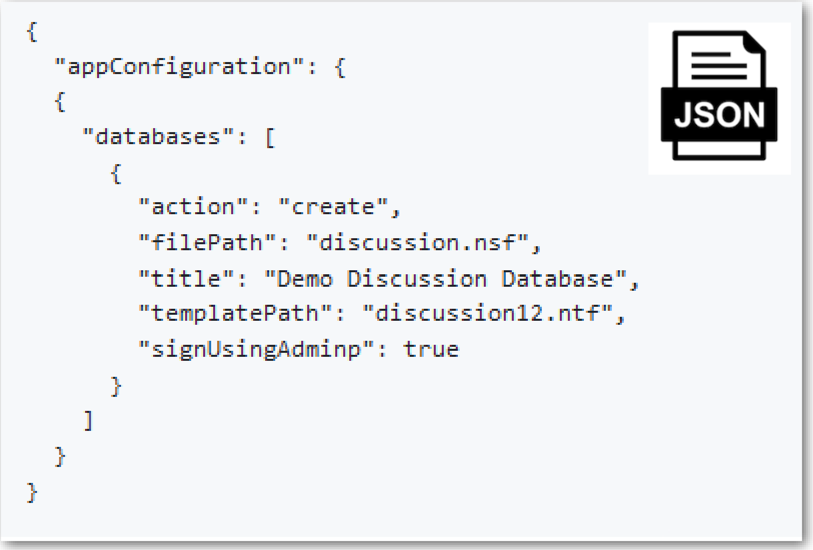
```
server -autoconf setup.json
```

- **Invoke OTS JSON via environment variables**
  - Specially interesting for but not limited to container environments

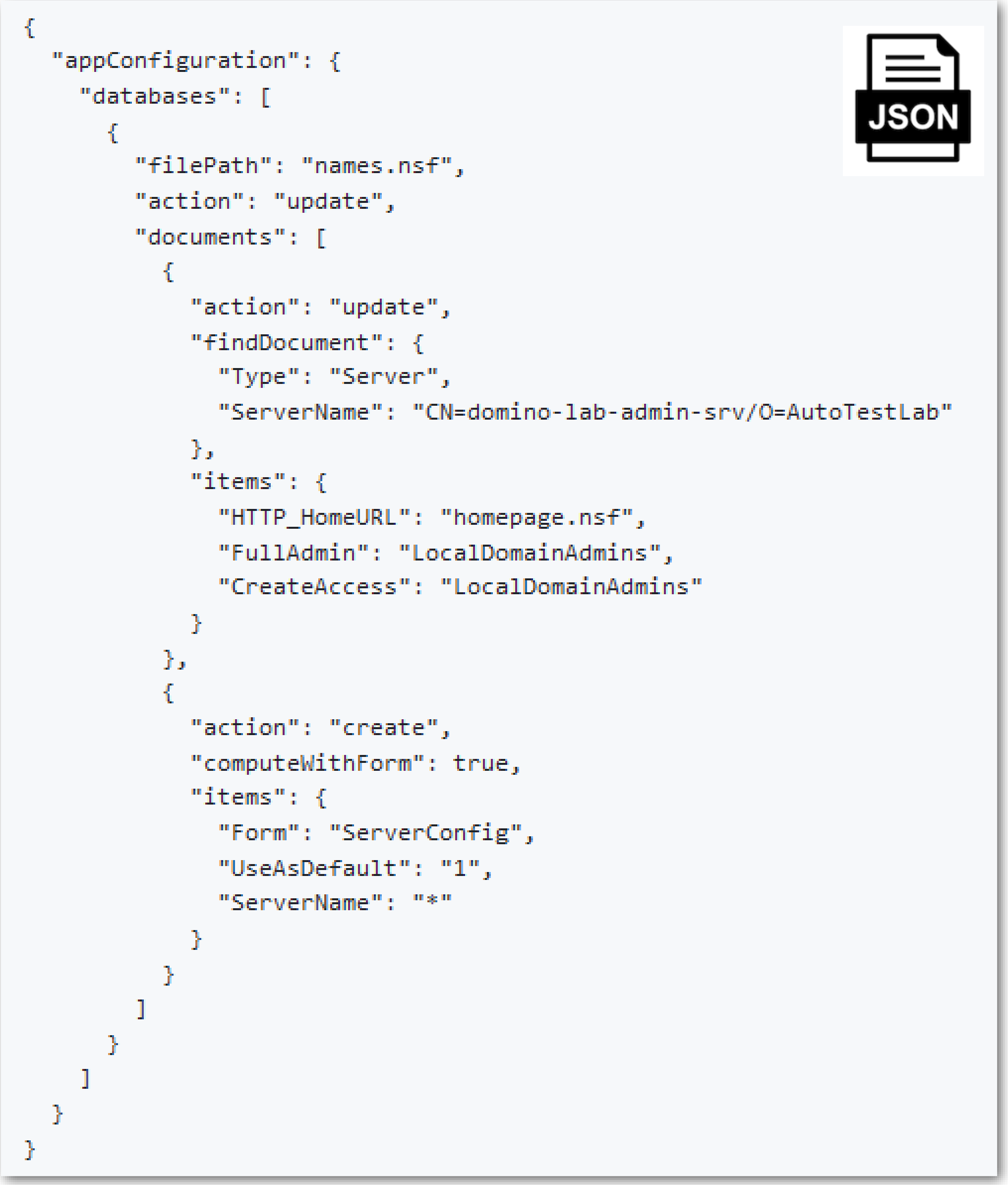
```
SetupAutoConfigureParams=/tmp/auto-config.json

SetupAutoConfigure=1
```

Application Configuration

- **Very flexible LEGO brick approach**
  - Create and update databases & documents
  - Can be used for system databases and applications



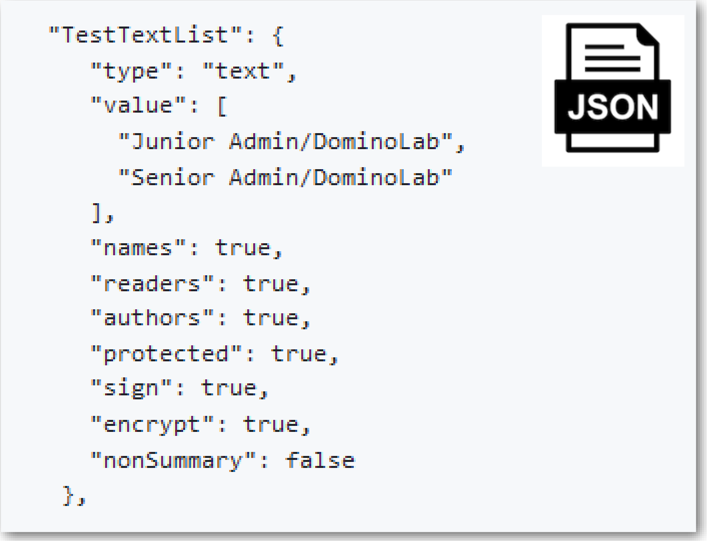
```
{
  "appConfiguration": {
  {
    "databases": [
      {
        "action": "create",
        "filePath": "discussion.nsf",
        "title": "Demo Discussion Database",
        "templatePath": "discussion12.ntf",
        "signUsingAdminp": true
      }
    ]
  }
}
```



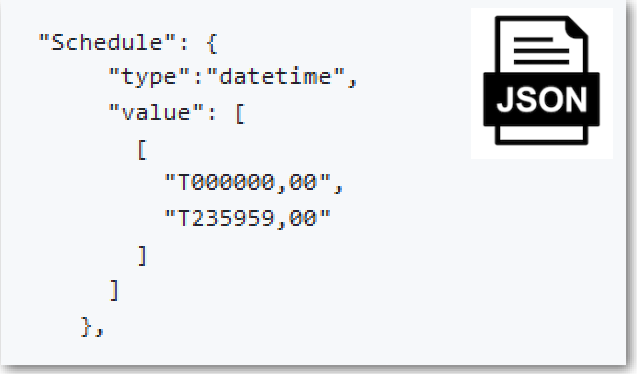
```
{
  "appConfiguration": {
    "databases": [
      {
        "filePath": "names.nsf",
        "action": "update",
        "documents": [
          {
            "action": "update",
            "findDocument": {
              "Type": "Server",
              "ServerName": "CN=domino-lab-admin-srv/O=AutoTestLab"
            },
            "items": {
              "HTTP_HomeURL": "homepage.nsf",
              "FullAdmin": "LocalDomainAdmins",
              "CreateAccess": "LocalDomainAdmins"
            }
          },
          {
            "action": "create",
            "computeWithForm": true,
            "items": {
              "Form": "ServerConfig",
              "UseAsDefault": "1",
              "ServerName": "*"
            }
          }
        ]
      }
    ]
  }
}
```

Application Configuration

- **Create and update documents**
  - Find documents with simple search syntax
  - Formula based search (V14.0+)
  - Response document support (V14.0+, important for Internet sites)
- **Items and item flags**
  - Support for all basic items including text-lists and item flags
  - Support for date ranges (V14.0+, important for connection docs)



```
"TestTextList": {
   "type": "text",
   "value": [
     "Junior Admin/DominoLab",
     "Senior Admin/DominoLab"
   ],
   "names": true,
   "readers": true,
   "authors": true,
   "protected": true,
   "sign": true,
   "encrypt": true,
   "nonSummary": false
 },
```



```
"Schedule": {
     "type":"datetime",
     "value": [
       [
         "T000000,00",
         "T235959,00"
       ]
     ]
   },
```

Vaildating OTS JSON

- Basic validation can be done with any JSON tool like **JQ**
  - To make ensure JSON content is OK, validate it against the schema

- Domino ships
  - **dominoOneTouchSetup.schema.json** a schema file (Domino binary directory)
  - **validjson** a Domino binary
  - **Admin client:** Configuration > Tools > Server > Validate JSON

- Validate a ots.json file against the schema via
  - `validjson valid.json -default`

- **Tip:** A separate tool without Domino dependency written in C++ with RapidJSON
  - https://github.com/HCL-TECH-SOFTWARE/domino-container/tree/main/tools/checkjson
  - Useful on Linux when you don't have Domino installed (e.g. in a container environment)

Troubleshooting failed installs

- **Validating OTS JSON is an important best practices**
  - This avoids many possible issues (and round trips when testing)
  - Still not all failures can be prevented

- **OTS writes an error log**
  - **IBM_TECHNICAL_SUPPORT/autoconfigure.log**
  - Not all error messages are speaking messages in the right context yet
  - But it gives you a good idea of what is going wrong
  - Some errors are fatal, some are warnings
  - Specially for additional server setups there can be connection issues
  - ID Vault operations are a post install operation
    - For example uploading registered users to ID Vault

Domino 14.0 Enhancements

- **Domino 14.0**
  - Certificate Authority
  - Create a full-text index
  - Autoregister servers with specific names
  - An additional server can be created using files in a "seed" directory if the first server is unavailable
    - Offline for example on a ship
  - Create replicas on additional servers
  - Create response documents
  - **FindDocument** criteria can be specified by using a **formula**
  - Support for date ranges

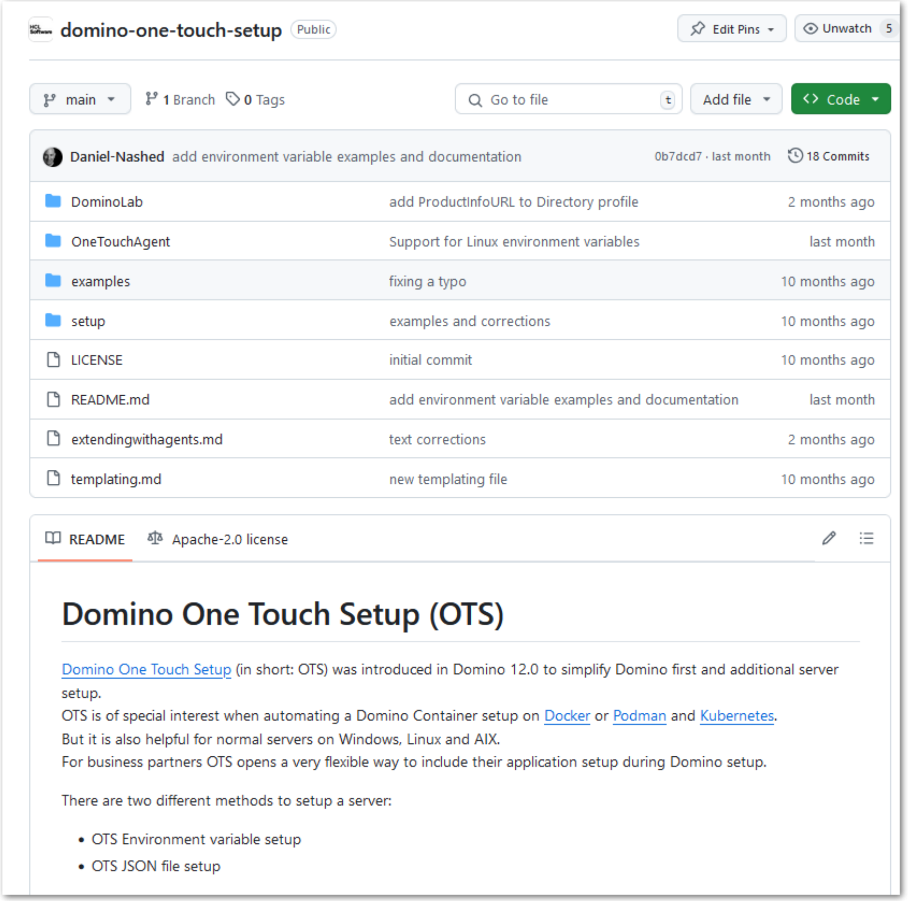
HCLSoftware

HCL One Touch GitHub Project

The best starting point

HCL One Touch GitHub Project

- https://github.com/HCL-TECH-SOFTWARE/domino-one-touch-setup
- Documentation for OTS JSON including examples
- One Touch Lotus Script agent
- Full examples
- Let’s have a look



Daniel-Nashed add environment variable examples and documentation 0b7dcd7 · last month 18 Commits

DominoLab	add ProductInfoURL to Directory profile	2 months ago
OneTouchAgent	Support for Linux environment variables	last month
examples	fixing a typo	10 months ago
setup	examples and corrections	10 months ago
LICENSE	initial commit	10 months ago
README.md	add environment variable examples and documentation	last month
extendingwithagents.md	text corrections	2 months ago
templating.md	new templating file	10 months ago

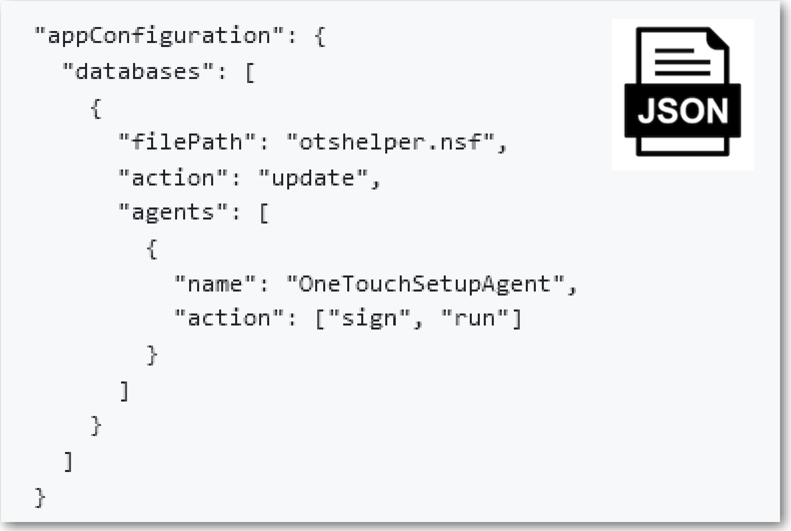
Domino One Touch Setup (OTS)

Domino One Touch Setup (in short: OTS) was introduced in Domino 12.0 to simplify Domino first and additional server setup.
OTS is of special interest when automating a Domino Container setup on Docker or Podman and Kubernetes.
But it is also helpful for normal servers on Windows, Linux and AIX.
For business partners OTS opens a very flexible way to include their application setup during Domino setup.

There are two different methods to setup a server:

- OTS Environment variable setup
- OTS JSON file setup

Running Agents inline



```
"appConfiguration": {
  "databases": [
    {
      "filePath": "otshelper.nsf",
      "action": "update",
      "agents": [
        {
          "name": "OneTouchSetupAgent",
          "action": ["sign", "run"]
        }
      ]
    }
  ]
}
```

- **Signing databases**
  - Databases are signed by **server.id** via **Adminp**
  - Asynchronous operation not helpful to run agents during setup
- But agent actions can also sign and run agents inline
  - Very useful for application setup
- **“If you only have a hammer, every problem looks like a nail”**
  - OTS is very flexible functionality – But not the solution to everything
  - Only intended for server setup
- Agents are a good way for reusable logic
  - Can be used in OTS or normal application setup logic for existing servers

OTS Helper Agent

- **Agents**
  - Not all functionality can be available in OTS
  - But Notes Agents can encapsulate logic and allow additional functionality

- **Current functionality**
  - Create clusters and add servers to clusters via Adminp request
  - Cross certify a **safe.id**
  - Configuring Domino 14 Auto Update & Install (requires updating Domino directory profile)

- Domino container supports **safe.id** and **CustomNotesdataZip**

  - **`SafeIDFile=https://domino.lab/safe.id`**
  - **`CustomNotesdataZip=https://raw.githubusercontent.com/HCL-TECH-SOFTWARE/domino-one-touch-setup/main/OneTouchAgent/otshelper.zip`**

HCLSoftware

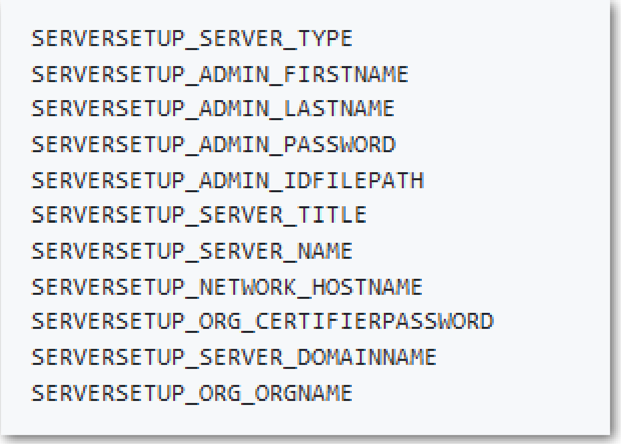
Combining Environment variables & JSON

Use the best of both worlds
native and in containers

“OTS Templating”

- **Combining the best of both worlds**
  - OTS Environment variables are a very simple way to setup servers
  - OTS JSON is a very flexible but more complex way
- Idea: OTS Template variables in Helm, Ansible style
  - Variables are mapped 1:1 from Environment setup to JSON

```
"name": "{{ SERVERSETUP_SERVER_NAME }}",
```



```
SERVERSETUP_SERVER_TYPE
SERVERSETUP_ADMIN_FIRSTNAME
SERVERSETUP_ADMIN_LASTNAME
SERVERSETUP_ADMIN_PASSWORD
SERVERSETUP_ADMIN_IDFILEPATH
SERVERSETUP_SERVER_TITLE
SERVERSETUP_SERVER_NAME
SERVERSETUP_NETWORK_HOSTNAME
SERVERSETUP_ORG_CERTIFIERPASSWORD
SERVERSETUP_SERVER_DOMAINNAME
SERVERSETUP_ORG_ORGNAME
```

Domino Container OTS Template Support

- Domino Start Script and Domino container project provide code to replace variables
  - **`/local/notesdata/DominoAutoConfigTemplate.json`**

    But there is more ..

- **Containers can download OTS templates via HTTPS**
  - Example: **`SetupAutoConfigureTemplateDownload=https://nashcom.lab/ots.json`**

- **Interactive command-line variable setup**
  - Supported by Domino Container run-time script **dominoctl** and Domino start script
  - Invoked via **`domino setup`** or **`dominoctl setup`**
  - Both projects come with a first and additional server pre-defined configuration
  - They can also consume any other JSON OTS file from local or remote
    - **`domino setup https://nashcom.lab/ots.json`**

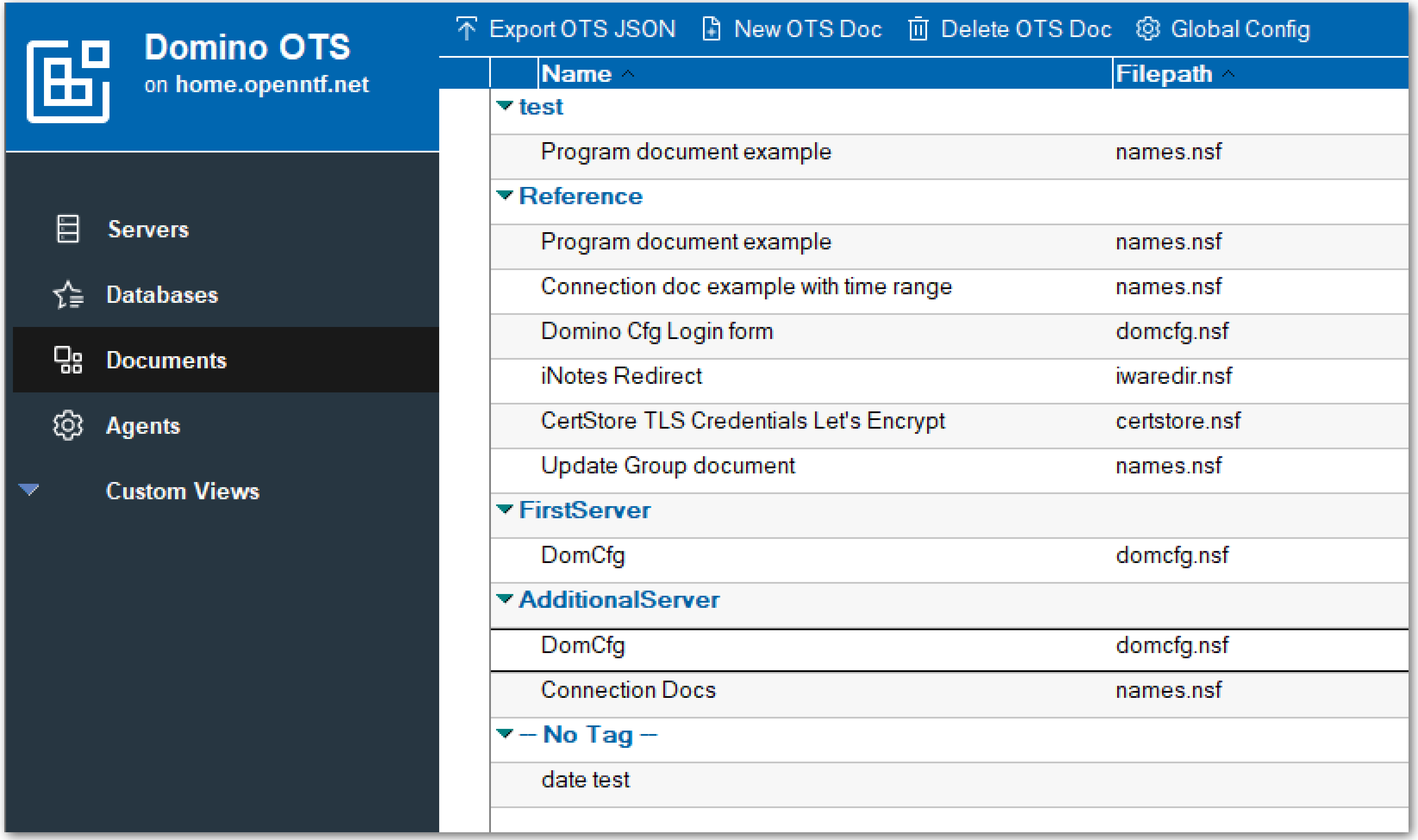
HCLSoftware

Demos & Examples

Generate JSON automatically?
Bring up 3 servers in one domain?

Create OTS JSON via Lotus Script?

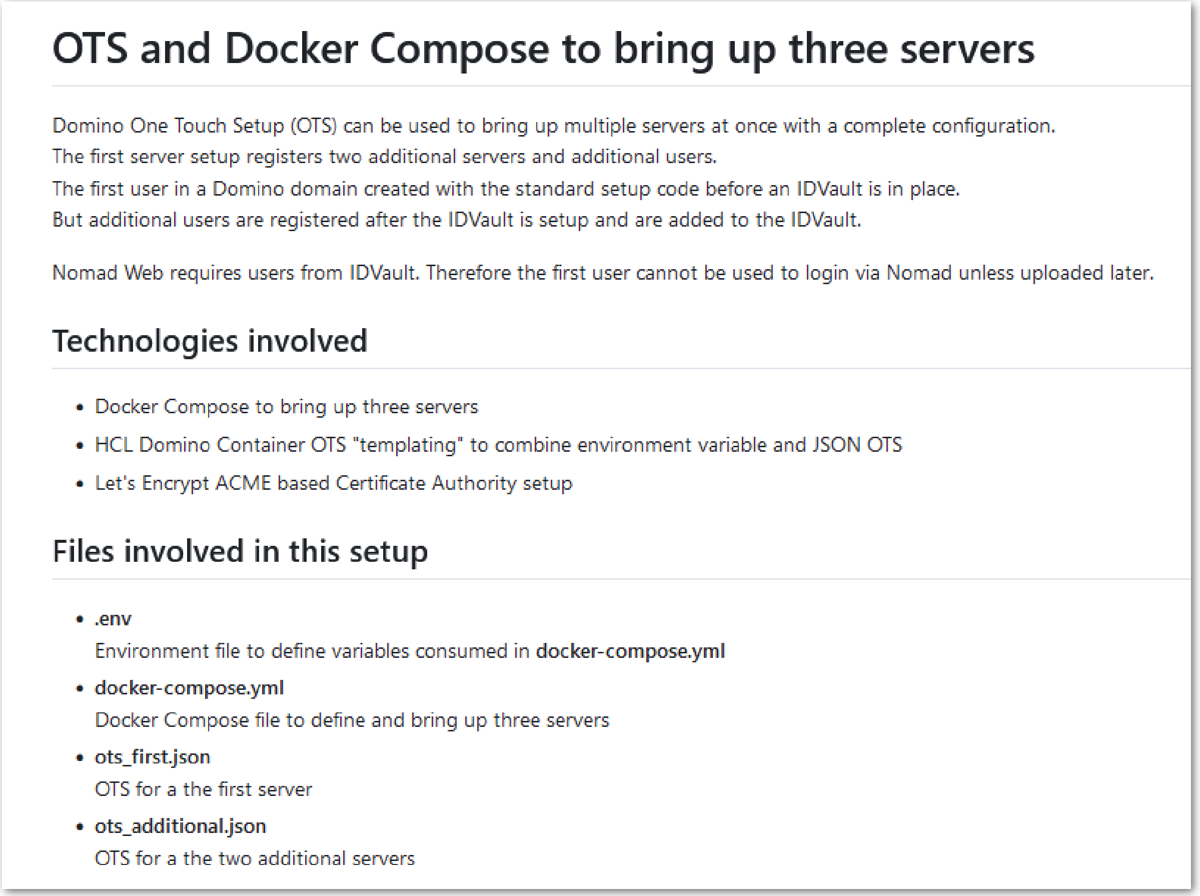
- Project to generate OTS JSON
- Available for free
- Provides
  - Server config
  - Database creation
  - Run agent configuration
  - Generate OTS JSON from existing documents
- **Generates full ots.json**
  - Export button
  - “ots.json” agent
- References & Examples



Name	Filepath
test	
Program document example	names.nsf
Reference	
Program document example	names.nsf
Connection doc example with time range	names.nsf
Domino Cfg Login form	domcfg.nsf
iNotes Redirect	iwaredir.nsf
CertStore TLS Credentials Let's Encrypt	certstore.nsf
Update Group document	names.nsf
FirstServer	
DomCfg	domcfg.nsf
AdditionalServer	
DomCfg	domcfg.nsf
Connection Docs	names.nsf
– No Tag –	
date test	

Bring up 3 servers end to end

- Example is available in OTS GitHub repo
- Full end to end example which can be cloned and extended
- Also good for learning
  - Let’s have a look



OTS and Docker Compose to bring up three servers

Domino One Touch Setup (OTS) can be used to bring up multiple servers at once with a complete configuration.
The first server setup registers two additional servers and additional users.
The first user in a Domino domain created with the standard setup code before an IDVault is in place.
But additional users are registered after the IDVault is setup and are added to the IDVault.

Nomad Web requires users from IDVault. Therefore the first user cannot be used to login via Nomad unless uploaded later.

Technologies involved

- Docker Compose to bring up three servers
- HCL Domino Container OTS "templating" to combine environment variable and JSON OTS
- Let's Encrypt ACME based Certificate Authority setup

Files involved in this setup

- **.env**
  Environment file to define variables consumed in **docker-compose.yml**
- **docker-compose.yml**
  Docker Compose file to define and bring up three servers
- **ots_first.json**
  OTS for a the first server
- **ots_additional.json**
  OTS for a the two additional servers

- https://github.com/HCL-TECH-SOFTWARE/domino-one-touch-setup/tree/main/DominoLab

HCL Documentation & Projects

- **Domino One Touch Setup**
  - https://help.hcl-software.com/domino/12.0.2/admin/wn_one-touch_domino_setup.html
- **HCL One Touch Setup Community GitHub repository**
  - https://github.com/HCL-TECH-SOFTWARE/domino-one-touch-setup
- **HCL Domino Container Project**
  - https://opensource.hcltechsw.com/domino-container/
- **Nash!Com Domino Start Script for Linux and AIX**
  - https://nashcom.github.io/domino-startscript/

Questions & Answers

- **Thank you** for your interest!

- **Open questions in chat?**
  - Presentation will be available for download from OpenNTF
  - There will be a Q&A summary on OpenNTF

- Additional information
  - https://blog.nashcom.de
  - **nsh@nashcom.de**