



A true story about Infrastructure as Code



Sinisa Sokolic

Chief Technology Officer

Wakes up at 5 AM for the gym, grows his beard
instead of his hair
... knows how to use a camera



Benedikt Brauneck

Developer

IAFH (Infrastructure automator from hell)
Never sleeps, drinks too much energy drinks
... builds synthesizers for fun

CUSTOMER PROJECT

Expectations...

- Update XenDesktop 7.6 FP3 to current release
- Improve general VDI performance (GPU)
- Update software delivery to something that really works
- Make the infrastructure cloud-ready
- Implement change management and versioning



IMPROVEMENT? ACCOMPLISHED!

	Before	After
Installation Time	6:30 hours	1:30 hours
Manual adjustments	3-4 hours	none
Deployment	PVS (6 minutes boot time per vm)	MCS (100 VDIs ready in 15 minutes)
Redeployment	No clue if ever possible	1:30 hours (complete infrastructure)
Staging	none	DEV/QA/PRD

Cloud Native Landscape

v0.9.9

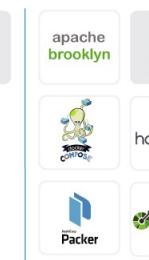
Database & Data Analytics



Streaming



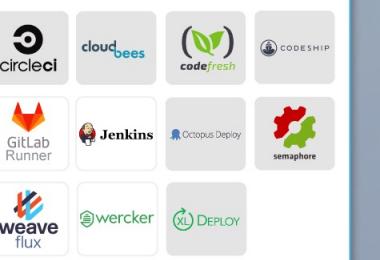
SCM



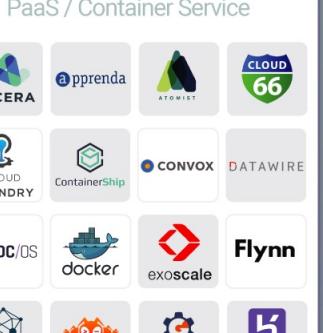
Application Definition



CI/CD



Platforms



Observability & Analysis

Monitoring



Scheduling & Orchestration

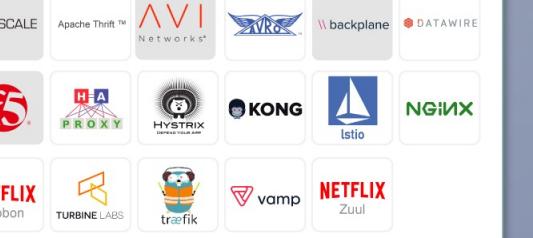
Orchestration & Management



Coordination & Service Discovery

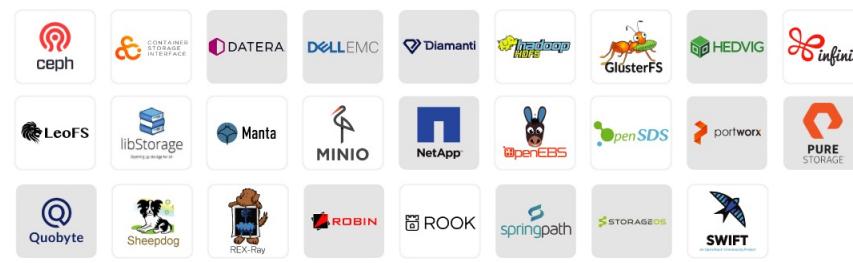


Service Management



Cloud-Native Storage

Runtime



Container Runtime

CNI

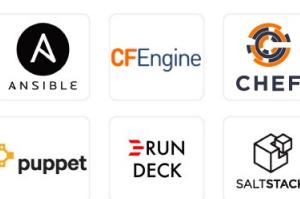


Cloud-Native Network



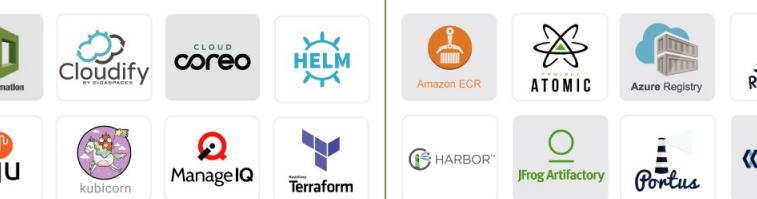
Host Management / Tooling

Provisioning



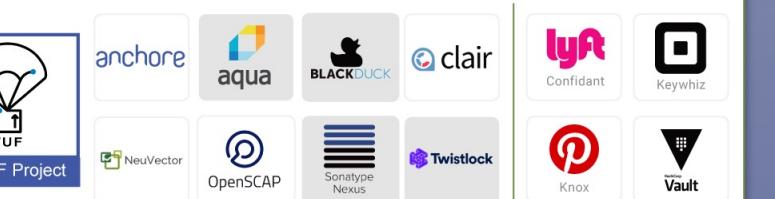
Infrastructure Automation

Container Registries



Secure Images

Key Management

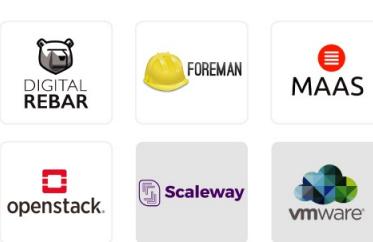


Public

Cloud



Private

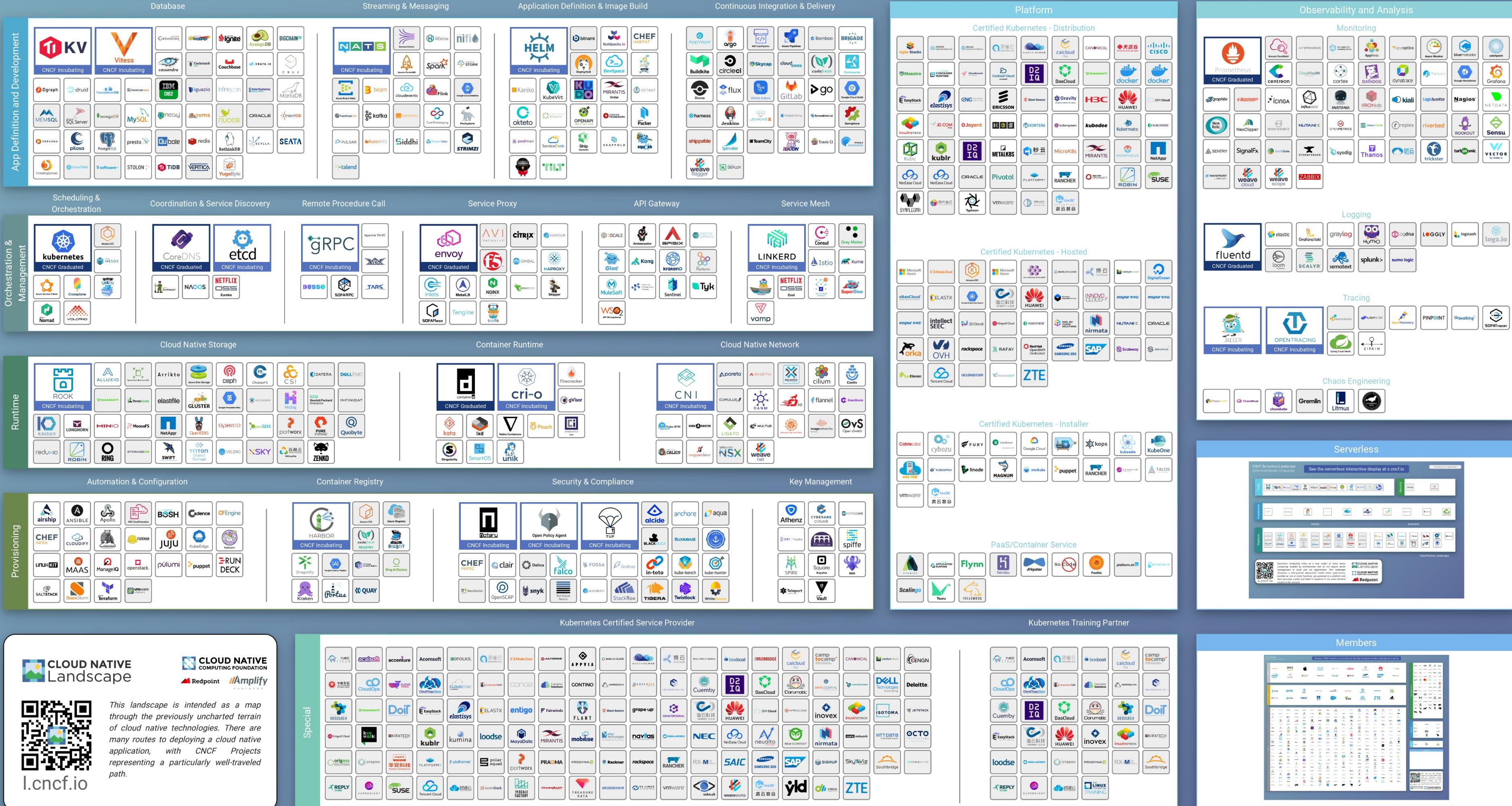


Greyed logos are not open source

This landscape is intended as a map through the previously uncharted terrain of cloud native technologies. There are many routes to deploying a cloud native application, with CNCF Projects representing a particularly well-traveled path.

CLOUD NATIVE COMPUTING FOUNDATION

Redpoint | **Amplify**





PACKER

CREATE IMAGES



TERRAFORM

CREATE ENVIRONMENT



POWERSHELL DSC

CONFIGURE ENVIRONMENT



PSPM

USER ENVIRONMENT



PACKER

Image Management made easy

Current Version 1.4.4

www.packer.io

Open Source – please contribute

Extend if you want/need to

brew install packer

choco install packer

WHY PACKER?

BECAUSE...

1

Create fully updated OS images

From ISO or cloud based templates

2

Re-usability of base images

Saves time and costs, same image in DEV/ACC/PRD, immutable infra, standardization

3

Provision on multiple targets

One pipeline creates images On-Premises and in the Cloud

4

Speed through automation

Speed, speed, speed.... daily, weekly, monthly ... Continuous Delivery

5

Use cloud based images

No necessary bootstrap (AutoUnattend, etc.)

```
1 {"builders": [ {
2     "type": "vmware-iso",
3     "tools_upload_flavor": "windows",
4     "remote_username": "root",
5     "remote_password": "██████████",
6     "remote_type": "esx5",
7     "remote_host": "████████████",
8     "remote_datastore": "████████",
9     "headless": false,
10    "version": 13,
11    "keep_registered": true,
12    "skip_export": true,
13    "skip_compaction": true,
14    "vm_name": "SRV-16_1908211",
15    "guest_os_type": "windows9srv-64",
16
17    "communicator": "winrm",
18    "winrm_username": "Administrator",
19    "winrm_password": "████████",
20    "winrm_timeout": "15m",
21    "shutdown_command": "shutdown /s /t 0 /f",
22    "shutdown_timeout": "2h",
23    "boot_wait": "0s",
24    "boot_command": "<down><enter><enter><enter>",
25    "vnc_disable_password": true,
26    "cpus": 8,
27    "disk_size": 122880,
28    "network_adapter_type": "e1000e",
29    "memory": 16384,
30    "disk_adapter_type": "lsisas1068",
31    "disk_type_id": "thin",
32
33    "iso_url": "./ISO/SW_DVD9_Win_Server_STD_CORE_2016_64Bit_English_-4_DC_STD_MLF_X21-70526.ISO",
34    "iso_checksum": "████████████████████████████████",
35    "iso_checksum_type": "SHA1",
36    "floppy_files": [
37        "./Build/SERVER_2016_STD/Autounattend.xml",
38        "./Config/Builder/winrm.ps1"
39    ],
40    "vmx_data": {
41        "ethernet0.present": "TRUE",
42        "ethernet0.startConnected": "TRUE",
43        "ethernet0.networkName": "1845_local",
44        "uefi.secureBoot.enabled": "FALSE",
45        "firmware": "efi",
46        "bios.bootorder": "cdrom,hdd",
47
48        "sched.mem.pin": "TRUE",
49        "sched.mem.min": "16384"
```

Packer – On-Premises

- Highly complex
- Different for every customer and environment
- Complexity grows with different Hypervisors

```
1  [
2      "builders": [
3          {
4              "type": "amazon-ebs",
5              "region": "eu-central-1",
6              "instance_type": "t2.medium",
7              "ssh_username": "ec2-user",
8              "ami_name" : "Windows_SRV2K19_Base_{{timestamp}}",
9              "ami_description": "Customized W2k19_base",
10             "associate_public_ip_address": "true",
11             "source_ami_filter": {
12                 "filters": [
13                     {
14                         "virtualization-type": "hvm",
15                         "name": "Windows_Server-2019-English-Full-Base-*",
16                         "root-device-type": "ebs"
17                     },
18                     {
19                         "owners": ["amazon"],
20                         "most_recent": true
21                     }
22                 ],
23                 "user_data_file": "./scripts/SetUpWinRM.ps1",
24                 "communicator": "winrm",
25                 "winrm_username": "Administrator",
26                 "winrm_insecure": true,
27                 "winrm_use_ssl": false,
28                 "vpc_id": "██████████",
29                 "subnet_id": "██████████"
30             }
31         }
32     ]
33 ]
```

PACKER IN A CODEBUILD PIPELINE ON AWS (JSON)

```
1 ---  
2 version: 0.2  
3  
4 phases:  
5   install:  
6     runtime-versions:  
7       python: 3.7  
8   pre_build:  
9     commands:  
10    - curl -qL -o packer.zip https://releases.hashicorp.com/packer/1.4.3/packer\_1.4.3\_linux\_amd64.zip && unzip packer.zip  
11    - ./packer version  
12    - curl -qL -o packer-provisioner-windows-update-linux.tgz https://github.com/rgl/packer-provisioner-windows-update/releases/download/v0.7.1/packer-provisioner-windows-update-linux.tgz  
13    - chmod a+x packer-provisioner-windows-update  
14    - echo 'Validate json'  
15    - ./packer validate ./packer_builds/w2k19_base.json  
16   build:  
17     commands:  
18       - echo "Building HashiCorp Packer template, w2k19.json"  
19       - ./packer build ./packer_builds/w2k19_base.json  
20  
21   post_build:  
22     commands:  
23       - echo "HashiCorp Packer build completed on `date`"  
24
```

PACKER IN A CODEBUILD PIPELINE ON AWS (YAML)

- [Dashboard](#)
- [Image Management](#)
- [Infrastructure Management](#)
- [Configuration Management](#)
- [Profile Management](#)

Home / Dashboard

Modules

Image Management with Packer 3 Enabled	Infrastructure Management with Terraform 2 Enabled	Configuration Management with PowerShell DSC 20 Enabled	Profile Management with PowerShell Profile Management 3 Enabled
Analytics & Monitoring with Grafana Disabled	Citrix Infrastructure Templates with Terraform Disabled	DSC Infrastructure Templates with PowerShell DSC Disabled	Code and Infrastructure Tests with Pester Disabled
Microsoft FSLogix Disabled	Chocolatey Open Source Packages Disabled	AWS <small>Preview</small> Workspaces support Not Connected	Microsoft Application Virtualization Not Connected

Infrastructures

Microsoft Azure Not Connected	Amazon <small>AWS</small> Connected	Google Cloud Not Connected	Alibaba <small>Preview</small> Cloud Not Connected
--	--	---	---

W2k19_base

[Edit ▾](#)[Delete build project](#)[Start build](#)

Configuration

Source provider
Amazon S3

Primary repository
[infrax0-artifacts/](#)

Artifacts upload location
-

Build badge
Disabled

[Build history](#)[Build details](#)[Build triggers](#)[Metrics](#)

Build history

[Stop build](#)[View artifacts](#)[View logs](#)[Delete builds](#)[Retry build](#)< 1 2 3 >

<input type="checkbox"/>	Build run	Status	Source version	Submitter	Duration	Completed
<input type="checkbox"/>	W2k19_base:0feabafb-f74c-47b2-8475-f41fa754f78f	Succeeded	infrax0-artifacts/	portal-frontend-user	15 minutes 10 seconds	17 hours ago

16 MINUTES BUILD TIME

COSTS 0,16 \$

19:00 BUILD TIME!
0,19 \$ BUILD COSTS!

14



Terraform

Infrastructure creation made easy

Current Version 0.12.12

www.terraform.io

Open Source – please contribute

Extend if you want/need to

brew install terraform

choco install terraform

WHY TERRAFORM?

BECAUSE...

1

Create your infrastructure from scratch
On-premises and in the Cloud

2

Use your Packer images
Trigger redeployments through hooks and triggers

3

All major Cloud and On-Premises vendors
Build once, deploy multiple

4

Live the DevOps Life!
Everything is code, so everyone can contribute

5

Don't need to code for one solution
Abstract values and everything else into variables and modules

EXPLORER

citrix.tf X

citrix.tf > citrix_storefront_cluster

```
1 module "citrix_storefront_cluster" {
2     source = "./modules/citrix/storefront-cluster"
3
4     instance_count          = "${var.citrix_storefront_instance_count}"
5     instance_type            = "${var.citrix_storefront_instance_type}"
6     instance_image_name      = "${var.citrix_storefront_ami_name}"
7     instance_volume_size_in_gb = "${var.citrix_storefront_instance_volume_size_in_gb}"
8     instance_hostname         = "${var.citrix_storefront_hostname}"
9
10    domain_name    = "${var.domain_name}"
11    domain_zone_id = "${aws_route53_zone.infraxo_domain.zone_id}"
12
13    active_directory_domain_name      = "${var.ad_domain_name}"
14    active_directory_domain_controller_ip = "${var.ad_domain_controller_ip}"
15    active_directory_domain_admin_username = "${var.ad_domain_admin_username}"
16    active_directory_domain_admin_password = "${var.ad_domain_admin_password}"
17
18    dsc_configuration_name = ""           // The configuration name will be generated automatically by the
19    dsc_pullserver_url      = "${var.dsc_pullserver_url}"
20    dsc_registration_key    = "${var.dsc_registration_key}"
21
22    vpc_id                  = "${module.vpc.vpc_id}"
23    public_subnet_ids        = "${module.vpc.public_subnets}"
24    private_subnet_ids       = "${module.vpc.private_subnets}"
25
26    name_prefix              = "${local.name_prefix}"
27    environment              = "${local.environment}"
28    aws_account_id           = "${var.aws_account_id}"
29
30    tags = "${local.default_tags}"
31 }
32 }
```

```
3
4 aws_region = "eu-central-1"
5
6 aws_profile = "terraform-demolab" # aws configure --profile terraform-demolab => with access key of terraform-demolab user
7
8 # Terraform labels
9
10 company_fullname = "RIS Consulting GmbH"
11
12 app_fullname = [REDACTED]
13
14 company_name = [REDACTED]
15
16 app_name = [REDACTED]
17
18 # Network
19
20 network_address = "10.100.0.0/16"
21
22 subnet_count = 2
23
24 public_subnet_address_pool = ["10.100.101.0/24", "10.100.102.0/24", "10.100.103.0/24", "10.100.104.0/24", "10.100.105.0/24"]
25
26 private_subnet_address_pool = ["10.100.1.0/24", "10.100.2.0/24", "10.100.3.0/24", "10.100.4.0/24", "10.100.5.0/24"]
27
28 # Active Directory Domain
29
30 ad_domain_name = [REDACTED]
31
32 ad_domain_controller_ip = [REDACTED]
33
34 ad_domain_admin_username = [REDACTED]
35
36 ad_domain_admin_password = [REDACTED]
37
38 # Domain (Route53)
39
40 domain_name = [REDACTED]
41
42 # DSC
43 dsc_pullserver_url = [REDACTED]
44
45 dsc_registration_key = [REDACTED]
46
47 # GitHub
```



Serverless DSC Pull-Server

Configuration management made easy

Runs completely in AWS Lambda

Implemented in .NET Core 3.0

Available with WMF 5.1 or Windows 7 upwards

Ment for servers but we use it on clients as well.

WHY DSC?

BECAUSE...

1

Infrastructure as Code
On-premises and in/from the Cloud

2

Flexibility & Support
Domain and non-domain joined hosts, software deployment, get rid of GPOs, desired state, auditing,...

3

Connectivity through HTTPS
Remote configuration possible

4

Live the DevOps Life!
Everything is code, so everyone can contribute

5

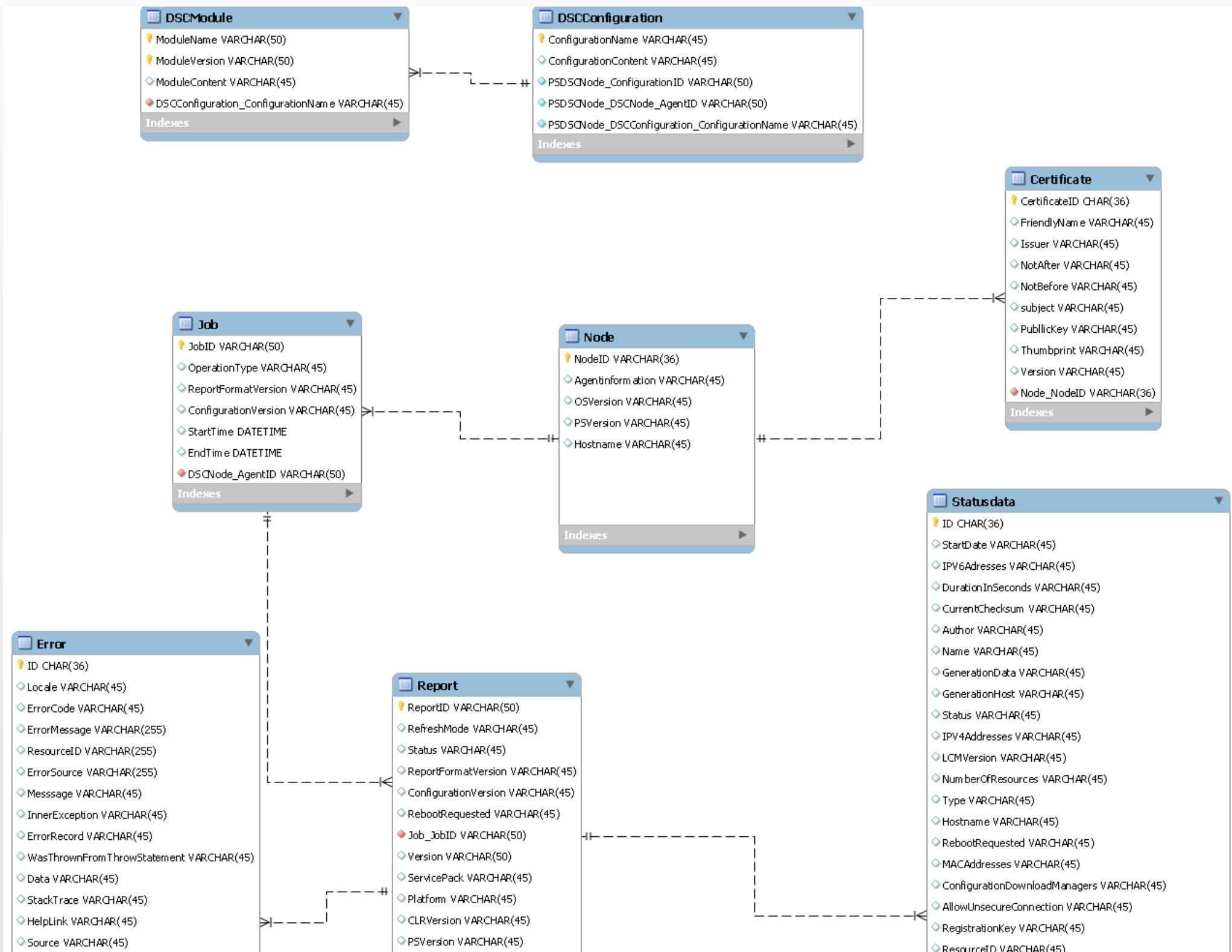
Extend with your own Modules
Abstract values and everything else into variables and modules

Devices	RegistrationData	StatusReport
<p>TargetName</p> <p>ConfigurationID</p> <p>ServerCheckSum</p> <p>TargetCheckSum</p> <p>NodeCompliant</p> <p>LastComplianceTime</p> <p>LastHeartbeatTime</p> <p>Dirty</p> <p>StatusCode</p>	<p>AgentId</p> <p>LCMVersion</p> <p>NodeName</p> <p>IPAddress</p> <p>ConfigurationNames</p>	<p>JobId</p> <p>Id</p> <p>OperationType</p> <p>RefreshMode</p> <p>Status</p> <p>LCMVersion</p> <p>ReportFormatVersion</p> <p>ConfigurationVersion</p> <p>NodeName</p> <p>IPAddress</p> <p>StartTime</p> <p>EndTime</p> <p>Errors</p> <p>StatusData</p> <p>RebootRequested</p> <p>AdditionalData</p>

GARBAGE...

Out-of-the-Box DSC database schema.

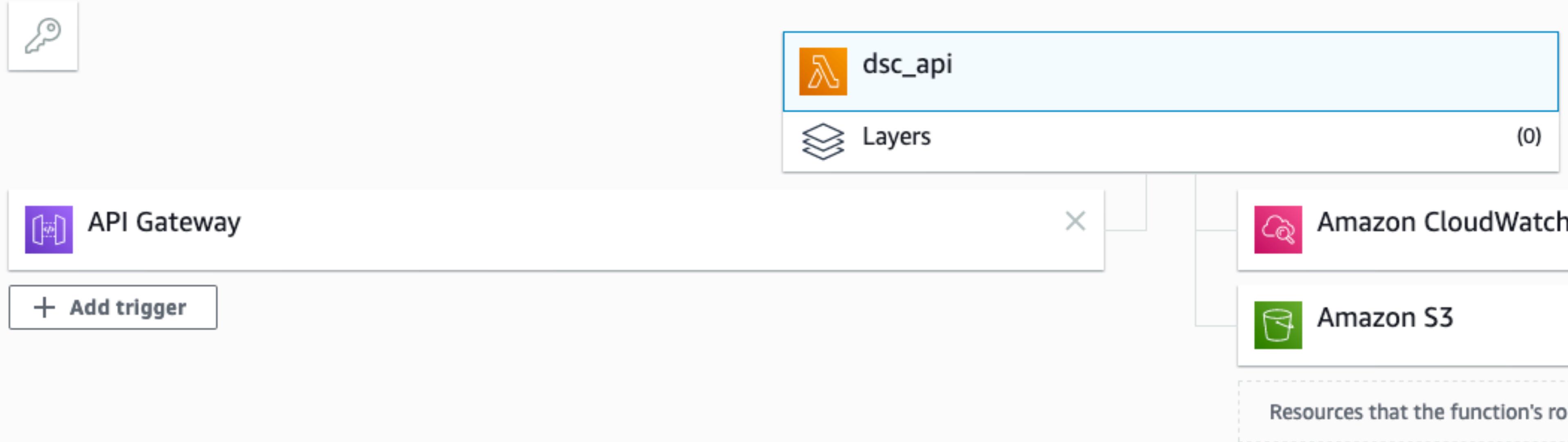
Can only use MS SQL or local JET DB



BETTER...

Reworked database schema.

▼ Designer



LAMBDA

DSC implementation in AWS

APIs

Demodata

dsc-api-gateway

Resources

Stages

Authorizers

Gateway Responses

Models

Resource Policy

Documentation

Dashboard

Settings

Usage Plans

API Keys

Custom Domain Names

Resources

Actions ▾

◀ Method Execution

/api/{dsc+} - ANY - Integration Request

Provide information about the target backend that this method will call and whether the incoming request data should be modified.

Integration type Lambda Function [i](#)

HTTP [i](#)

Mock [i](#)

AWS Service [i](#)

VPC Link [i](#)

Use Lambda Proxy integration [i](#)

Lambda Region eu-central-1 [e](#)

Lambda Function dsc_api [e](#)

Execution role [e](#)

Invoke with caller credentials [i](#)

Credentials cache Do not add caller credentials to cache key [e](#)

Use Default Timeout [i](#)

API PROXY

Because Microsoft uses "(" and ")", in its API request URLs.

DSC ENDPOINTS

RegisterAgent

GetConfiguration

GetModule

SendStatusreport

Additional custom Endpoints to extend the solution (eg. list all nodes)

Typ / Endpoint	Variable LCM	Type	Default values (in hours)	Call count per run	Call count / day (average)	Call count / month
GetDscAction	RefreshFrequencyMins	recurring	6	1	4	120
GetDscConfiguration	After configuration change	dynamic	-	1	2	60
GetModule	After configuration change	dynamic	-		2	60
SendStatusreport	ConfigurationModeFrequencyMins	recurring	3	3	24	720
RegisterDscAgent	-	one time	-	1	1	30
			Sum	6	33	990

Node count	calls per month
10	9.900
100	99.000
200	198.000
400	396.000
800	792.000
1.000	990.000 (1010 = 999.900)
2.000	1.980.000
4.000	3.960.000
8.000	7.920.000
10.000	9.900.000

DEFAULT
VALUES

SERVERLESS POWERSHELL DSC



*„The first time you can save money
running your workloads in the
cloud.“*



PowerShell Profile Management

Profile Management made easy

Current Version 1.2.7

Written completely in PowerShell

Deployment via msi or PowerShell DSC

Configurations stored as JSON

WHY?

- PowerShell DSC runs in SYSTEM context
- PowerShell runs everywhere
- Extendable
- Runs at Logon and Logoff
- No infrastructure needed

PowerShell Profile Management Framework

```
1 "ID": "e382ab0b-63e5-47be-8494-2a0cb32ddbca",
2 "Version": 18,
3 "Creator": "SinisaSokolic",
4 "Conditions": [],
5 "LogonActions": [
6   {
7     "ID": "e48f81c6-3b74-492b-b7f9-cbae3038a95e",
8     "Name": "Set-RegistryKey",
9     "Parameters": [
10       {
11         "Key": "HKEY_CURRENT_USER\\Software\\Microsoft\\Windows\\CurrentVersion\\Explorer\\Advanced",
12       },
13       {
14         "ValueName": "LaunchTo"
15       },
16       {
17         "TargetValue": "1"
18       },
19       {
20         "Type": "DWord"
21       }
22     ],
23   },
24   "ActionCommand": "Set-RegistryKey -Key \"HKEY_CURRENT_USER\\Software\\Microsoft\\Windows\\CurrentVersion\\Explorer\\Advanced\"",
25   "Context": "User",
26   "IsOneTimeAction": "False"
27 },
28 {
29   "ID": "ddf75b97-936e-45d5-996b-2d4f0cea33af",
30   "Name": "Set-RegistryKey",
31   "Parameters": [
32     {
33       "Key": "HKEY_CURRENT_USER\\Software\\Microsoft\\Windows\\CurrentVersion\\Explorer\\AutoplayHandlers",
34     },
35     {
36       "ValueName": "DisableAutoplay"
37     },
38     {
39       "TargetValue": "1"
40     },
41     {
42       "Type": "DWord"
43     }
44   ],
45   "ActionCommand": "Set-RegistryKey -Key \"HKEY_CURRENT_USER\\Software\\Microsoft\\Windows\\CurrentVersion\\Explorer\\AutoplayHandlers\"",
46   "Context": "User",
47   "IsOneTimeAction": "False"
48 },
49 {
50   "ID": "06f330d3-b9be-4bb6-97ac-7a5799b01f00",
51   "Name": "Set-RegistryKey",
52   "Parameters": [
53     {
54       "Key": "HKEY_CURRENT_USER\\Software\\Microsoft\\Siuf\\Rules",
55     },
56     {
57       "ValueName": "NumberOfSIUFInPeriod"
58     },
59   ],
60   "ActionCommand": "Set-RegistryKey -Key \"HKEY_CURRENT_USER\\Software\\Microsoft\\Siuf\\Rules\"",
61   "Context": "User",
62   "IsOneTimeAction": "False"
63 }
```

Actions

Show Notification	Sync File
Close Notification	Create Network Printer
Copy Directory	Create Shared Printer
Delete Directory	Delete Printer
Sync Directory	Run Process
Map Drive	Export Registry
Unmap Drive	Import Registry
Remove Environment Variable	Remove Registry Key
Set Environment Variable	Set Registry Key
Copy File	Execute Script
Delete File	Execute ScriptCommand

Conditions

Directory Exists	Application Is Installed
File Exists	Computername Is
Fileversion Is	User Is
User Is In Group	SystemVariable Is
Internet Is Available	Systemvariable Contains
Registry Key Exists	UserDomain Is
Registry Value Contains	Username Is
Registry Value Exists	

USE CASES

- Notifications
- Application Settings
- User Roaming
- User Environment Management
- OS Migrations
- User Experience customizing

Powershell DSC-Notification



Software update available!

For security and stability reasons, we
kindly ask you to restart your
computer as soon as possible.

Windows PowerShell

Restart now

DSC NOTIFICATIONS

- BurnToast Module
- <https://github.com/Windos/BurntToast>
- Because DSC cannot show user notifications
- DSC reboots are enabled or disabled, there is no in-between

CHALLENGES

- Many „People“ still can't code
- Escape reflex when customers see Visual Studio Code
- A lot of different technologies to learn
- Mindsets need to change, organizations as well
- Complexity of implementation in large environments
 - Big teams with a lot of dependencies
 - Pipeline definitions
 - Access Keys, credentials, versioning ...

INFRAXO
IT AUTOMATION

WHAT IS INFRA.XO?

- IT automation solution
- Abstracts the code and the complexity
- Modules
 - Image Management
 - Infrastructure Management
 - Configuration Management
 - Environment Management
- More to come (WebGUI assisted in the cloud and via Electron)
 - Chocolatey integration
 - FSlogix integration
 - Hardening policies vis DSC
 - GPO replacement via DSC and much more ...

- [Dashboard](#)
- [Image Management](#)
- [Infrastructure Management](#)
- [Configuration Management](#)
- [Profile Management](#)

Home / Dashboard

Modules

Image Management

with Packer

Targets 3

Enabled

Infrastructure Management

with Terraform

Stacks 2

Enabled

Configuration Management

with PowerShell DSC

Configurations 20

Enabled

Profile Management

with PowerShell Profile Management

Profiles 3

Enabled

Analytics & Monitoring

with Grafana

Disabled

Citrix Infrastructure Templates

with Terraform

Disabled

DSC Infrastructure Templates

with PowerShell DSC

Disabled

Code and Infrastructure Tests

with Pester

Disabled

Microsoft

FSLogix

Disabled

Chocolatey

Open Source Packages

Disabled

AWS Preview

Workspaces support

Not Connected

Microsoft

Application Virtualization

Not Connected

Infrastructures

Microsoft

Azure

Not Connected

Amazon

AWS

Connected

Google

Cloud

Not Connected

Alibaba Preview

Cloud

Not Connected

- [Dashboard](#)
- [Image Management](#)
- [Infrastructure Management](#)
- [Configuration Management](#)
- [Profile Management](#)

Home / Dashboard

Modules

Image Management with Packer 3 Enabled	Infrastructure Management with Terraform 2 Enabled	Configuration Management with PowerShell DSC 20 Enabled	Profile Management with PowerShell Profile Management 3 Enabled
Analytics & Monitoring with Grafana Disabled	Citrix Infrastructure Templates with Terraform Disabled	DSC Infrastructure Templates with PowerShell DSC Disabled	Code and Infrastructure Tests with Pester Disabled
Microsoft FSLogix Disabled	Chocolatey Open Source Packages Disabled	AWS <small>Preview</small> Workspaces support Not Connected	Microsoft Application Virtualization Not Connected

Infrastructures

Microsoft Azure Not Connected	Amazon AWS Connected	Google Cloud Not Connected	Alibaba <small>Preview</small> Cloud Not Connected
--	--	---	---

DOES IT FIT?



Where?

- On-Premises / Cloud
- Virtualized environments
- Fast changing environments

Reproducability?

- Everything is code
- Immutable infrastructure
- Desaster or failure? Just redeploy, reduce backups

Solutions

- Backend infrastructure deployment
- VDI and Hosted Apps with MCS or PVS
- Everything that needs to be agile

TELL US WHAT YOU THINK!

WANT BETA ACCESS? CONTACT:

S.SOKOLIC@RISCONSULTING.DE

@SINISASOKOLIC