12.FEB.2018

MeetUp Hannover

Azure IaaC - Deployment Automation for SAP on Azure

Jens Gerecke
Azure Cloud Solutions Architect

Microsoft Deutschland GmbH
Jens.Gerecke@Microsoft.com
+49 (160) 5892 156

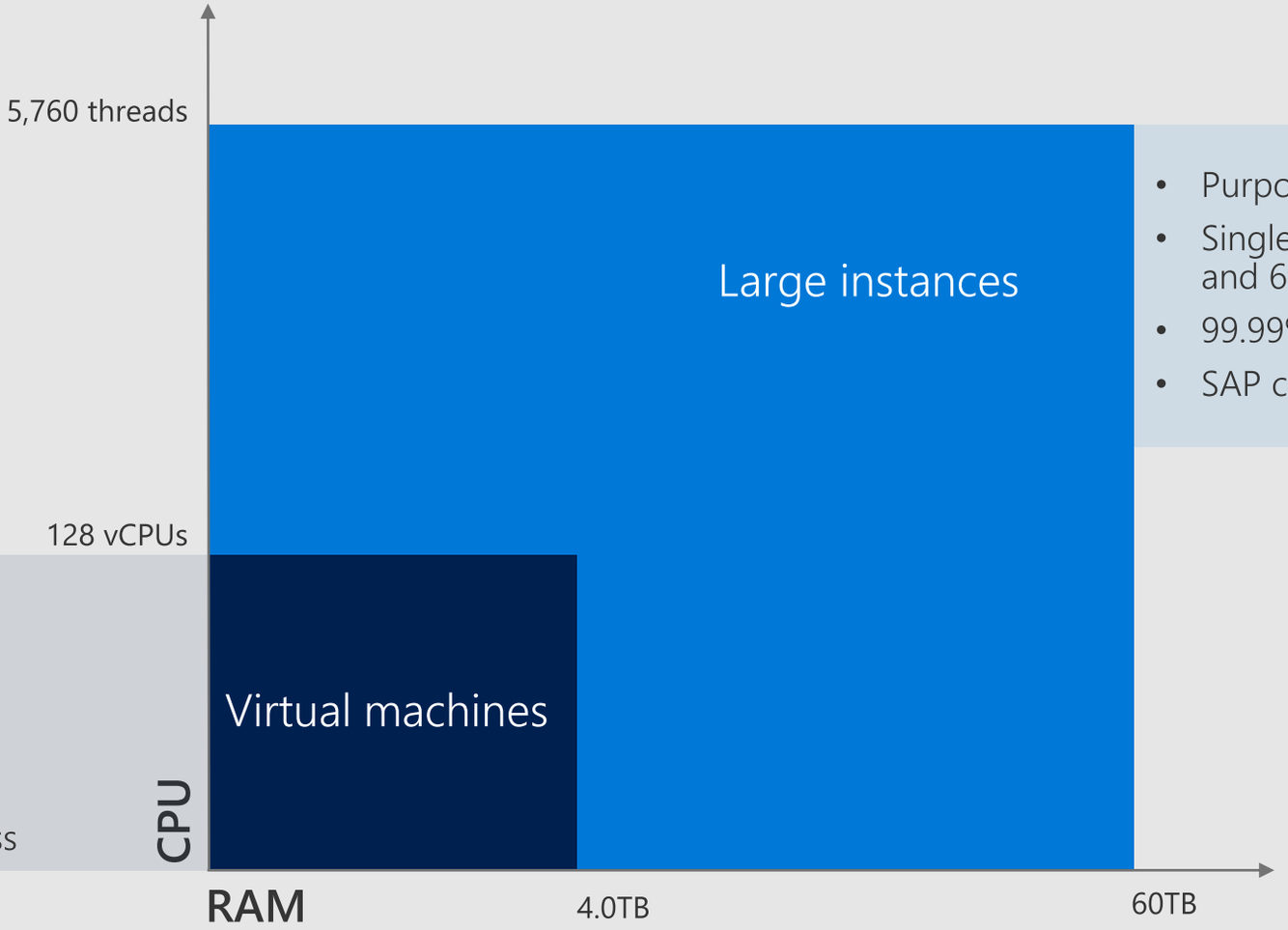
Microsoft

Agenda

```
{
    "18:30": "Begrüßung & Pizza",

    "19:00": "Infrastructure as Code - Teil 1",
    "parameters": {
        "SpeakerName": "Simon Schwingel - Cloud Solution Architect @ Microsoft ",
        "Topics": "Einführung Infrastructure as Code & Erfahrungen aus aktuellen Projekten"
    },

    ## Pause

    "20:00": "Infrastructure as Code - Teil 2",
    "parameters": {
        "SpeakerName": "Jens Gerecke - Cloud Solution Architect @ Microsoft ",
        "Topics": "Azure Deployment Automation & Projektbeispiel SAP on Azure"
    }
}
```

Azure - Most powerful and scalable cloud for SAP HANA

A combination of VMs and purpose-built large instances provides the largest scale and widest range for SAP HANA of any hyperscale cloud



- Purpose-built infrastructure for SAP HANA
- Single node supports up to 20TB for OLTP, and 60TB OLAP in scale-out mode
- 99.99% SLA (High availability)
- SAP certified

- On-demand infrastructure
- Single VM up to 4.0TB
- SLA: 99.9% (Single VM) or 99.95% (high availability)
- Per second billing
- SAP certification in progress

M-Series Virtual Machines

Massive memory, CPU, storage, and scale

Hyper-threaded support

Premium storage support

Based on Intel® Xeon® Processor E7-8890 v3

High performance DDR4 memory

Support Nested Virtualization with Windows Server 2016

SAP certified for Netweaver on AnyDB

HANA certification by April CY18

VM size	vCPU's	Memory: TB	Local SSD: TB	Persistent Data Disks Max	Network bandwidth	Availability
M64s	64	1.0	2	64	Extremely high	GA in U.S. and EU
M64ms	64	1.75	2	64	Extremely high	GA in U.S. and EU
M128s	128	2.0	4	64	Extremely high	GA in U.S. and EU
M128ms	128	4.0	4	64	Extremely high	GA in U.S. and EU

SAP on Azure

Quick-start guide: Manual installation of single-instance SAP HANA on Azure VMs

https://docs.microsoft.com/en-us/azure/virtual-machines/workloads/sap/hana-get-started

SAP HANA (large instances) overview and architecture on Azure

https://docs.microsoft.com/en-us/azure/virtual-machines/workloads/sap/hana-overview-architecture

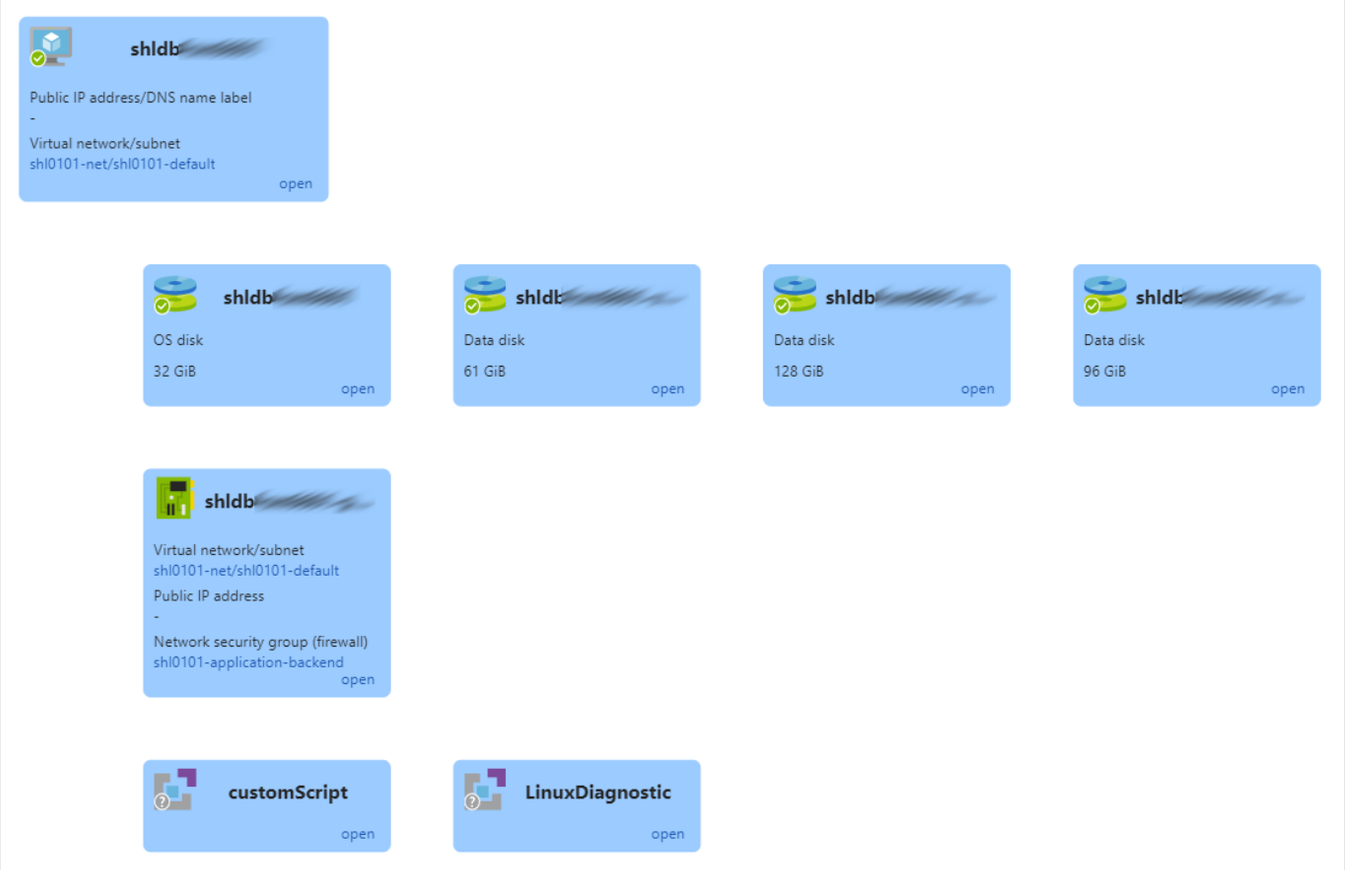
How to install and configure SAP HANA (large instances) on Azure

https://docs.microsoft.com/en-us/azure/virtual-machines/workloads/sap/hana-installation

PoC Environment for SAP on Azure

Customer is using SLES for SAP Linux VMs

SAP Database VM



Jumphost VM

Simple json templates

VS.Code

SLES VM – challenge with WALinuxAgent

After successful deployment customer found warnings in logs

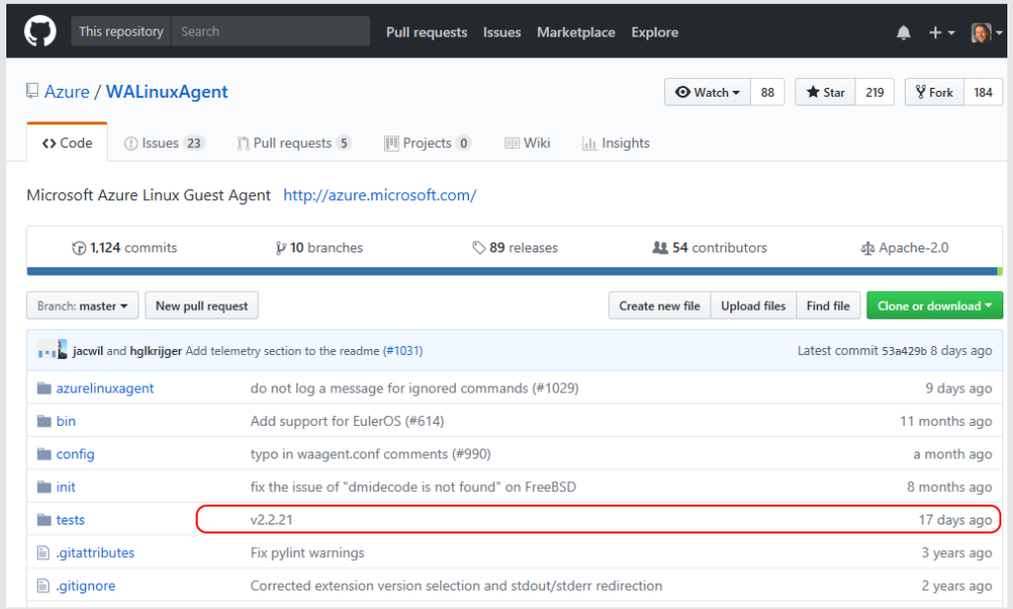
```
2018-01-29T09:39:00.947741+00:00 shlhd python[2702]: 2018/01/29 09:39:00.947549 INFO [HTTP Delay] Delay 15 seconds for Status Code 503
2018-01-29T09:39:04.039452+00:00 shlhd python[2702]: 2018/01/29 09:39:04.039316 INFO [HTTP Delay] Delay 15 seconds for Status Code 503
2018-01-29T09:39:04.041400+00:00 shlhd python[2702]: 2018/01/29 09:39:04.041288 INFO [HTTP Retry] Attempt 2 of 3: [HTTP Retry] HTTP GET Status Code 503
2018-01-29T09:39:19.078633+00:00 shlhd python[2702]: 2018/01/29 09:39:19.078499 INFO [HTTP Delay] Delay 15 seconds for Status Code 503
2018-01-29T09:39:19.079599+00:00 shlhd python[2702]: 2018/01/29 09:39:19.079424 INFO [HTTP Retry] Attempt 3 of 3: [HTTP Retry] HTTP GET Status Code 503
2018-01-29T09:39:31.224775+00:00 shlhd python[2702]: 2018/01/29 09:39:31.224587 INFO Event: name=WALinuxAgent, op=HeartBeat, message=, duration=0
2018-01-29T09:39:34.116029+00:00 shlhd python[2702]: 2018/01/29 09:39:34.115837 INFO [HTTP Delay] Delay 15 seconds for Status Code 503
2018-01-29T09:39:34.153484+00:00 shlhd python[2702]: 2018/01/29 09:39:34.153374 INFO [HTTP Delay] Delay 15 seconds for Status Code 503
2018-01-29T09:39:34.155845+00:00 shlhd python[2702]: 2018/01/29 09:39:34.155741 INFO [HTTP Retry] Attempt 2 of 3: [HTTP Retry] HTTP GET Status Code 503
2018-01-29T09:39:49.188602+00:00 shlhd python[2702]: 2018/01/29 09:39:49.188456 INFO [HTTP Delay] Delay 15 seconds for Status Code 503
2018-01-29T09:39:49.189992+00:00 shlhd python[2702]: 2018/01/29 09:39:49.189873 INFO [HTTP Retry] Attempt 3 of 3: [HTTP Retry] HTTP GET Status Code 503
```

```
2018-01-29T10:02:40.416656+00:00 shlco python[1557]: 2018/01/29 10:02:40.398334 WARNING Storage service is temporarily unavailable.
2018-01-29T10:02:40.417052+00:00 shlco python[1557]: 2018/01/29 10:02:40.417003 INFO Will retry in 15 seconds.
2018-01-29T10:02:55.486924+00:00 shlco python[1557]: 2018/01/29 10:02:55.486862 WARNING Storage service is temporarily unavailable.
2018-01-29T10:02:55.506482+00:00 shlco python[1557]: 2018/01/29 10:02:55.506414 INFO Will retry in 15 seconds.
2018-01-29T10:03:13.595787+00:00 shlco python[1557]: 2018/01/29 10:03:13.595727 WARNING Storage service is temporarily unavailable.
2018-01-29T10:03:13.616647+00:00 shlco python[1557]: 2018/01/29 10:03:13.616589 INFO Will retry in 15 seconds.
```

Solution for WALinuxAgent – Agent Version!

```
linux@jgdemovm:~> zypper info python-azure-agent
Loading repository data...
Reading installed packages...


Information for package python-azure-agent:
-------------------------------------------
Repository     : @System
Name           : python-azure-agent
Version        : 2.2.18-34.11.1
Arch           : noarch
Vendor         : SUSE LLC <https://www.suse.com/>
Support Level  : unknown
Installed Size : 1.8 MiB
Installed      : Yes
Status         : up-to-date
Source package : python-azure-agent-2.2.18-34.11.1.src
Summary        : Microsoft Azure Linux Agent
Description    :
    The azure-agent supports the provisioning and running of Linux
    VMs in the Microsoft Azure Public Cloud and Microsoft Azure Stack private
    cloud. This package should be installed on Linux disk
    images that are built to run withing the Microsoft Azure or
    Microsoft Azure Stack framework.
```

"… To answer the question about upgrades – we depend on the distribution vendor to update the packages in their repo. @D…. may have some idea when this will happen next for SUSE. …"



https://docs.microsoft.com/en-us/azure/virtual-machines/linux/update-agent

https://github.com/Azure/WALinuxAgent

SLES VM – challenge with eth0 & eth1

„… Wenn wir zwei Interfaces bauen (via json template), dann fehlt die ifcfg-eth1 Konfig in der VM. Lege ich sie manuell an und starte den Netzwerk-Dienst durch, dann bekommt der Server auch seine zugewiesene IP. …“

```
jgdemovm:/etc/sysconfig/network # ip a | grep eth
2: eth0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast state UP group default qlen 1000
    link/ether 00:0d:3a:22:1c:76 brd ff:ff:ff:ff:ff:ff
    inet 192.168.10.4/24 brd 192.168.10.255 scope global eth0
3: eth1: <BROADCAST,MULTICAST> mtu 1500 qdisc noop state DOWN group default qlen 1000
    link/ether 00:0d:3a:23:6d:a4 brd ff:ff:ff:ff:ff:ff

jgdemovm:/etc/sysconfig/network # ls -lart ifcfg-eth*
-rw-r--r-- 1 root root 85 May 31  2016 ifcfg-eth0
```

Als OS habe ich hier:

```
"publisher": "SUSE",
"offer": "SLES-Priority",
"sku": "12-SP2",
"version": "latest"
```

SLES VM – solution for eth0 & eth1 issue

Microsoft.Network/virtualNetworks template reference - https://docs.microsoft.com/en-us/azure/templates/microsoft.network/virtualnetworks
"apiVersion": "2017-10-01"

Microsoft.Network/publicIPAddresses template reference - https://docs.microsoft.com/en-us/azure/templates/microsoft.network/publicipaddresses
"apiVersion": "2017-10-01"

Microsoft.Network/networkInterfaces template reference - https://docs.microsoft.com/en-us/azure/templates/microsoft.network/networkinterfaces
"apiVersion": "2017-10-01"

Microsoft.Network/networkSecurityGroups/securityRules template reference - https://docs.microsoft.com/en-us/azure/templates/microsoft.network/networksecuritygroups/securityrules
"apiVersion": "2017-10-01"

Microsoft.Storage/storageAccounts template reference - https://docs.microsoft.com/en-us/azure/templates/microsoft.storage/storageaccounts
"apiVersion": "2017-10-01"

Microsoft.Compute/virtualMachines template reference - https://docs.microsoft.com/en-us/azure/templates/microsoft.compute/virtualmachines
"apiVersion": "2017-12-01" - Achtung! Hier sagt die Docu "2017-12-01" aber bei meinem Test habe ich bemerkt, dass aktuell "2017-03-30" verwendet wird.

Microsoft.Compute/virtualMachines/extensions template reference - https://docs.microsoft.com/en-us/azure/templates/microsoft.compute/virtualmachines/extensions
"apiVersion": "2017-12-01" - Achtung! Hier sagt die Docu "2017-12-01" aber bei meinem Test habe ich bemerkt, dass aktuell "2017-03-30" verwendet wird.

Mit der Anpassung der apiVersion in den json templates läuft jetzt das Deployment und beide eth config sind wie erwartet richtig angelegt.

```
linux@jgdemovm:~> cd /etc/sysconfig/network
linux@jgdemovm:/etc/sysconfig/network> ls -lart ifcfg-eth*
-rw-r--r-- 1 root root 114 Oct 30 15:52 ifcfg-eth0
-rw-r--r-- 1 root root 186 Feb  1 10:02 ifcfg-eth1
```

SLES VM – challenge with data disc's

Unexpected data disc order in SLES VM

```
"resources": [
    {
        "apiVersion": "[parameters('apiVersion')]",
        "comments": "Deployment of multiple managed disc's premium",
        "copy": {
            "name": "managedDiscCopy",
            "count": "[length(parameters('managedDiscList'))]"
          },
        "type": "Microsoft.Compute/disks",
        "name": "[toLower(concat(parameters('vmName'), '-', parameters('managedDiscList')[copyIndex()].name))]",
        "location": "[resourceGroup().location]",
        "properties": {
            "creationData": {
                "createOption": "[parameters('managedDiscList')[copyIndex()].createOption]",
                "sourceUri": "[parameters('managedDiscList')[copyIndex()].sourceUri]"
            },
            "accountType": "[parameters('managedDiscList')[copyIndex()].accountType]",
            "diskSizeGB": "[parameters('managedDiscList')[copyIndex()].diskSizeGB]",
            "osType": "[parameters('managedDiscList')[copyIndex()].osType]"
        },
        "dependsOn": [],
        "tags": {
          "Department": "[parameters('tagDepartment')]",
          "Project": "[parameters('tagProject')]",
          "Customer": "[parameters('tagCustomer')]",
          "displayName": "[parameters('managedDiscList')[copyIndex()].name]"
        }
    }
]
```

```
"dataDisks": [
  {
    "lun": 0,
    "name": "[concat(parameters('vmName'), '-disc0_10')]",
    "createOption": "Attach",
    "caching": "None",
    "managedDisk": {
        "id": "[variables('datadisc0_10Id')]"
    }
  },
  {
    "lun": 1,
    "name": "[concat(parameters('vmName'),'-disc1_20')]",
    "createOption": "Attach",
    "caching": "None",
    "managedDisk": {
        "id": "[variables('datadisc1_20Id')]"
    }
  },
  {
    "lun": 2,
    "name": "[concat(parameters('vmName'),'-disc2_30')]",
    "createOption": "Attach",
    "caching": "None",
    "managedDisk": {
        "id": "[variables('datadisc2_30Id')]"
    }
  },
  {
    "lun": 3,
    "name": "[concat(parameters('vmName'),'-disc3_40')]",
    "createOption": "Attach",
    "caching": "None",
    "managedDisk": {
        "id": "[variables('datadisc3_40Id')]"
    }
  },
  {
    "lun": 4,
    "name": "[concat(parameters('vmName'),'-disc4_50')]",
    "createOption": "Attach",
    "caching": "None",
    "managedDisk": {
      "id": "[variables('datadisc4_50Id')]"
    }
  }
]
```

```
jgdemovm:~ # lsblk | grep sd
…
sdc       8:32   0   32G  0 disk
sdd       8:48   0  512G  0 disk
sde       8:64   0    1T  0 disk
sdf       8:80   0  256G  0 disk
sdg       8:96   0  512G  0 disk
```

<- expected 32GB disc
<- expected 1TB disc
<- expected 256GB disc
<- expected 512GB disc
<- expected 512GB disc

Linux VMs – solution for disc order

You cannot predict the order how the devices get mapped. I've experienced this a load of time with ARM automation. ..... Normally you can fix this on a bios level (in grub) .... but grub is not available during boot of Azure VMs.

In this blog by Ivan Mc Kinley ivmckinl@microsoft.com first paragraph outlines the scenario
https://access.redhat.com/documentation/en-us/red_hat_enterprise_linux/7/html/storage_administration_guide/persistent_naming

Have a look at this solution
https://github.com/hsirtl/sap-2-tier-on-oracle-linux/blob/master/setup_appsvr.sh
This script for an SAP-app server VM. The script provides a function "prepare_and_mount_lun" that allows you to specify the lun (that you specify via ARM remplate) and the mountpoint.

CLI 2.0 – automated logon via service principal

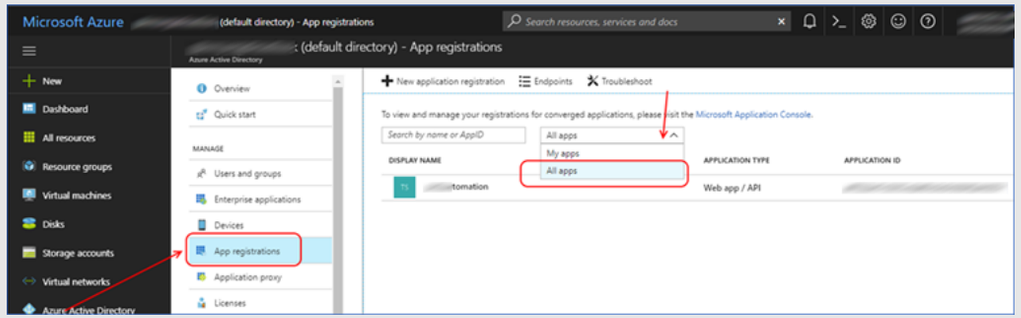
https://docs.microsoft.com/de-de/cli/azure/create-an-azure-service-principal-azure-cli?view=azure-cli-latest

Step 1:

```
az ad sp create-for-rbac --name spcliautomation --create-cert --years 1 --verbose
```

Das ist der erwartete Output. Die wichtigen Elemente habe ich markiert:

```
Retrying role assignment creation: 1/36
Please copy /home/<USER>/tmpj3fphfkk.pem to a safe place. When run 'az login' provide the
file path to the --password argument
{
  "appId": "b0723bc7-aaaa-bbbb-cccc-ddddfa4c351c",
  "displayName": “spcliautomation",
  "fileWithCertAndPrivateKey": "/home/<USER>/tmp1234xyz.pem",
  "name": "http://spcliautomation",
  "password": null,
  "tenant": „<YOUR_AAD_TENANT_ID>"
}
```



Step 2:

Überprüfen des erzeugten Service Principals im Azure Portal

Step 3:

Umbennen des erzeugten Cert-Files

```
mv tmp1234xyz.pem spcliautomation_cert.pem
```

Step 4:

Login in AZ CLI von einer beliebigen bash shell (Linux Maschine – nicht aus der Cloud Shell!!!)

Das Cert-File **spcliautomation_cert.pem** auf diese Linux Maschine kopieren

Und dann login into CLI via service principal & cert

```
az login --service-principal -u http://spcliautomation --password ~/spcliautomation_cert.pem --tenant <YOUR_AAD_TENANT_ID>
```

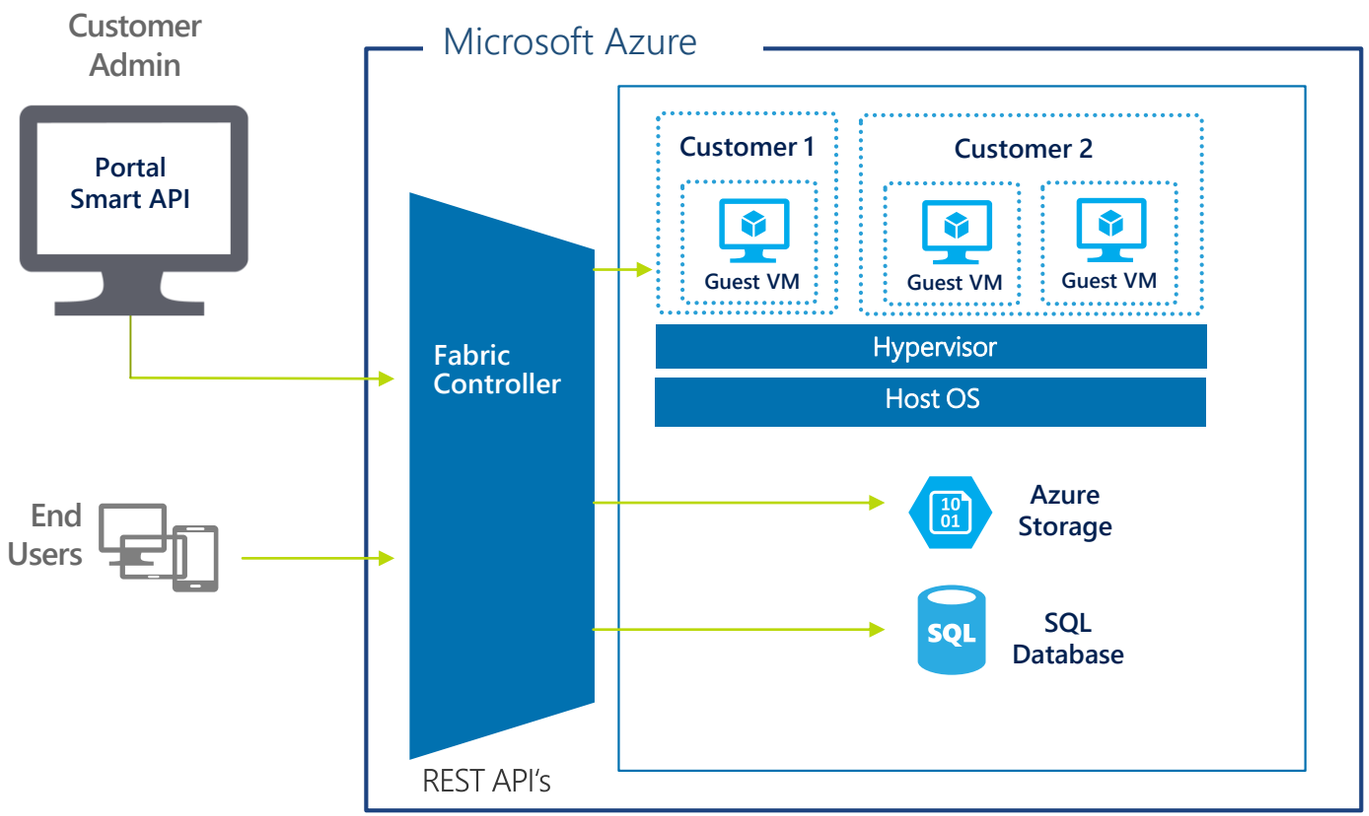
Using native Azure REST APIs

No Independencies for

- local client OS
- programing language or SDK
- Tools

Customer is using existing tools for automation based on REST APIs.

The “Magic” Azure Fabric Controller



- ✓ Isolates customer environments using the Fabric Controller
- ✓ Runs a configuration-hardened version of Windows Server as the Host OS
- ✓ Uses Hyper-V – a battle tested and enterprise proven hypervisor

Azure REST APIs

Components of a Azure REST API request/response

https://docs.microsoft.com/en-us/rest/api/

- request URI, request message header, request message body
- response message header, response message body

Authentication

POST https://login.microsoftonline.com/<YOUR_AAD_TENANT_ID>/oauth2/token

Subscription Management

GET https://management.azure.com/subscriptions?api-version=2016-06-01

Resourcegroup Management

GET https://management.azure.com/subscriptions/<YOUR_SUBSCRIPTION_ID>/resourcegroups?api-version=2017-05-10

Resource Management

https://management.azure.com/subscriptions/<YOU_SUBSCRIPTION_ID>/resourcegroups/<RESOURCEGROUP>/providers/Microsoft.Storage/storageAccounts/jgdemostorage20180130?api-version=2017-06-01

Using the Azure Resource Manager REST API

https://blogs.msdn.microsoft.com/cloud_solution_architect/2016/02/20/using-the-azure-resource-manager-rest-api/

Authentication – Get Bearer Token

The ARM REST API uses oauth2 authentication which requires that a bearer token be sent as a request header with each operation. This bearer token expires so must be periodically refreshed. Client libraries can support the automatic refresh of the bearer token but this must be done manually when the REST API is used.

The Get Bearer Token operation uses an Azure Active Directory (AAD) service principal to retrieve a bearer token. The creation of the service principal and its addition to an appropriate RBAC role is documented here and described in this post.

oAuth2 Authentication with Service Principal

Create a service principal with a password
https://docs.microsoft.com/en-us/cli/azure/create-an-azure-service-principal-azure-cli?toc=%2Fazure%2Fazure-resource-manager%2Ftoc.json&view=azure-cli-latest

Manage Azure Active Directory service principals for automation authentication
https://docs.microsoft.com/en-us/cli/azure/ad/sp?view=azure-cli-latest#create-for-rbac

https://docs.microsoft.com/en-us/cli/azure/ad/sp?view=azure-cli-latest#az_ad_sp_reset_credentials

HowTo – Create & Use Service Principal

Using CLI 2.0 to create a service principal:

```
az ad sp create-for-rbac --name rest_api_automation --years 1 --verbose
```

Result:

```
{
  "appId": "babababa-xxxx-yyyy-zzzz-6b9d9c022789",
  "displayName": "rest_api_automation",
  "name": "http://rest_api_automation",
  "password": „acacacac-0000-1111-2222-33f0daa2cf4c",
  "tenant": „<YOUR_AAD_TENANT_ID>"
}
```

Login into cli via service principal & Password

```
az login --service-principal -u http://rest_api_automation --password acacacac-0000-1111-2222-33f0daa2cf4c --tenant <YOUR_AAD_TENANT_ID>

az ad sp show --id http://rest_api_automation
```

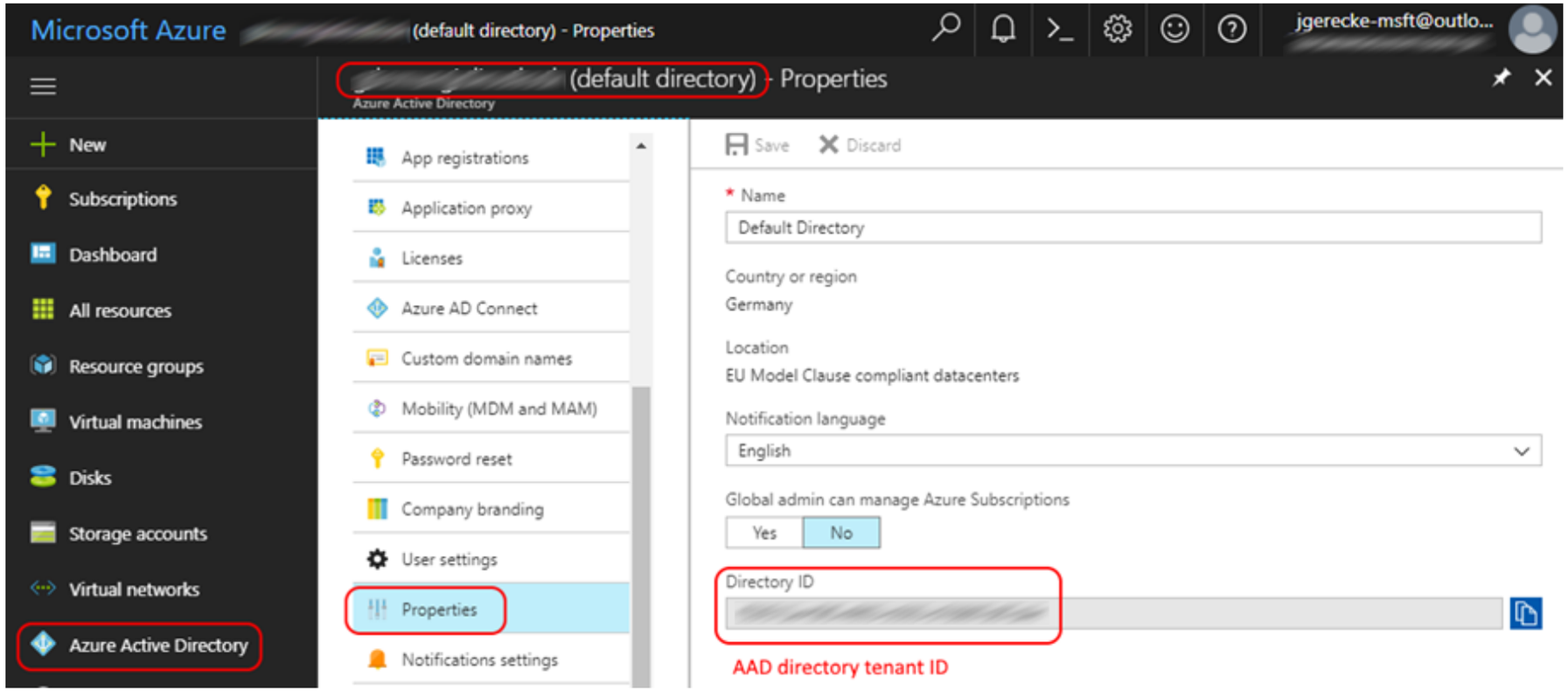
If required you can reset the Password

```
az ad sp reset-credentials --name http://rest_api_automation --years 1
```

HowTo – Use service principal in Postman (1)

- Postman - request oauth2 access token

- We need the AAD Tenant ID – Get it from Azure Portal



HowTo – Use service principal in Postman (2)

https://docs.microsoft.com/en-us/azure/active-directory/develop/active-directory-protocols-oauth-service-to-service#request-an-access-token

AAD Tenant ID = <YOUR_AAD_TENANT_ID>

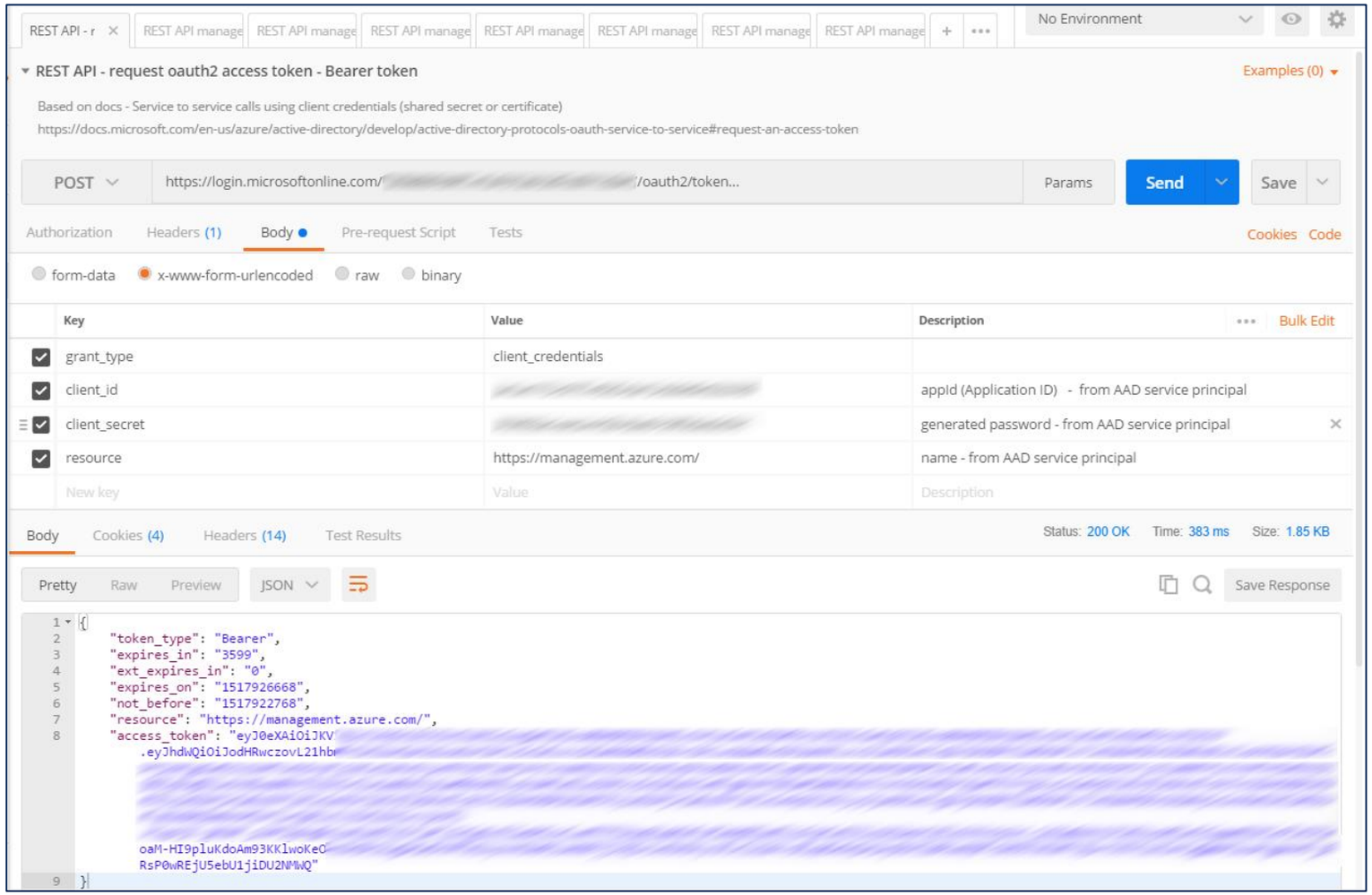
POST https://login.microsoftonline.com/<YOUR_AAD_TENANT_ID>/oauth2/token

		Key	Value	Description
Header		Content-Type	application/x-www-form-urlencoded	
Body		grant_type	client_credentials	
		client_id	babababa-xxxx-yyyy-zzzz-6b9d9c022789	appId (Application ID) - from AAD service principal
		client_secret	acacacac-0000-1111-2222-33f0daa2cf4c	generated password - from AAD service principal
		resource	https://management.azure.com/	Resource URL for which the Bearer Token will be valid

Postman – request oauth2 Bearer token

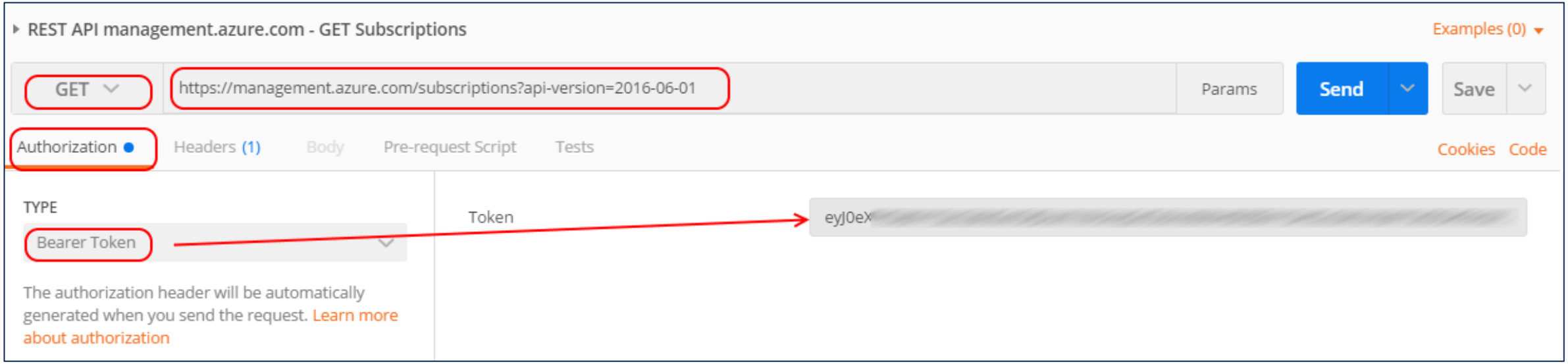
- Default lifetime for the token is 1 hour

- JWTs issued by Azure AD are signed, but not encrypted, you can easily inspect the contents of a JWT for debugging purposes. There are several tools available for doing so, such as [jwt.ms]



Postman – GET Subscriptions

https://docs.microsoft.com/en-us/rest/api/resources/subscriptions/get



Postman – CREATE or UPDATE resource group

https://docs.microsoft.com/en-us/rest/api/resources/resourcegroups/createorupdate

REST API management.azure.com - CREATE or UPDATE resource groups

Examples (0)

Based on Azure REST API docs - Resource Groups - Create Or Update https://docs.microsoft.com/en-us/rest/api/resources/resourcegroups/createorupdate

PUT https://management.azure.com/subscriptions/[redacted]/resourcegroups/jgdemo_rest_rg?api-version.. Params Send Save

Authorization Headers (2) Body Pre-request Script Tests Cookies Code

form-data x-www-form-urlencoded raw binary JSON (application/json)

```
{
    "location": "westeurope",
    "tags": {
        "Department": "TSI",
        "Project": "PoC SAPonAzure",
        "Customer": "xyz"
    }
}
```

Body Cookies Headers (11) Test Results Status: 201 Created Time: 893 ms Size: 768 B

Pretty Raw Preview JSON Save Response

```
{
    "id": "/subscriptions/[redacted]/resourceGroups/jgdemo_rest_rg",
    "name": "jgdemo_rest_rg",
    "location": "westeurope",
    "tags": {
        "Department": "TSI",
        "Project": "PoC SAPonAzure",
        "Customer": "xyz"
    },
    "properties": {
        "provisioningState": "Succeeded"
    }
}
```

Postman – CREATE or UPDATE deployment (via json template)

https://docs.microsoft.com/en-us/rest/api/resources/deployments/createorupdate

Requieres Azure Files URL with SAS token

REST API management.azure.com - CREATE or UPDATE deployment of resources via json template

Examples (0)

Based on Azure REST API docs - Deployments - Create Or Update
https://docs.microsoft.com/en-us/rest/api/resources/deployments/createorupdate
https://docs.microsoft.com/de-de/rest/api/resources/deployments
Deploys resources to a resource group. You can provide the template and parameters directly in the request or link to JSON files.

PUT https://management.azure.com/subscriptions/[redacted]/resourcegroups/jgdemo_rest_rg/providers/... Params Send Save

Authorization Headers (1) Body Pre-request Script Tests Cookies Code

form-data x-www-form-urlencoded raw binary JSON (application/json)

```
1  {
2    "properties": {
3      "mode": "Incremental",
4      "templateLink": {
5        "uri": "http://[redacted].file.core.windows.net/jsontemplates/azdeploy_storage.json?sv[redacted]",
6        "contentVersion": "1.0.0.0"
7      },
8      "parametersLink": {
9        "uri": "http://[redacted].file.core.windows.net/jsontemplates/azdeploy_storage.parameters.json?sv=[redacted]",
10       "contentVersion": "1.0.0.0"
11     },
12     "debugSetting": {
13       "detailLevel": "requestContent, responseContent"
14     }
15   }
16 }
```

URL for azure files for json template file - with SAS token

URL for azure files for json template parameters file - with SAS token

Questions?

Microsoft