This document describes the usage of the Alya Base Configuration for administrators.

Version

Version	Date	Author	Description
1.0	23.02.2021	Konrad Brunner (Alya)	Initial creation

Contents

1. What is the Base Configuration 1
2. Where do I get it from 2
3. The first step 2
4. Checkout the repository to a new device 3
5. Build your own starter package 6
6. The different directories 7
7. Information structure 8
   7.1. The hub site configuration script 9
8. Any further questions 9

1. What is the Base Configuration

The Alya Base Configuration is a collection of hundreds of PowerShell scripts for Microsoft Cloud services, which provides a standardized setup. The whole thing is open source according to the GPL 3.0 license, which means it is freely accessible and your own changes and additions are possible and welcome at any time. In the user community, mutual give and take should result in everyone benefiting from one another and thus reaching their goal faster.

The collection presents best practices from past projects from a wide variety of areas, for example Identity Management, Application Management, MFA, Security, Data Protection, Governance, Tenant and Subscription Management, Logging, Auditing and Reporting, Azure, Teams, SharePoint, Exchange, Network, Telephony, DevOps, Intune, Windows Virtual Desktop, etc.

The script collection of currently over 300 scripts is a compendium for all possible configuration tasks in the Microsoft Cloud and a common framework for everything else. It is managed with Git and is based on the latest modules and libraries. The multi-layer structure provides flexibility, supports as many application areas as possible and ensures easy updates between the forks.

A tried-and-tested naming concept across all services is easy to configure, facilitates collaboration and prevents confusion.

2. Where do I get it from

The most actual source of the Alya Base Configuration is hosted on GitHub and Azure DevOps:
On Azure DevOps: https://dev.azure.com/alyaconsulting/_git/ALYADO-ADM-Public
On GitHub: https://github.com/AlyaKoni/ALYADO-ADM-Public

We suggest hosting your own copy or fork of the project in your own GitHub or Azure DevOps. Please download a zip of the project, extract it on your local disk and check it in to your own git repository or just clone the project to your local disk and add a remote to your own git repository to commit it there.

3. The first step

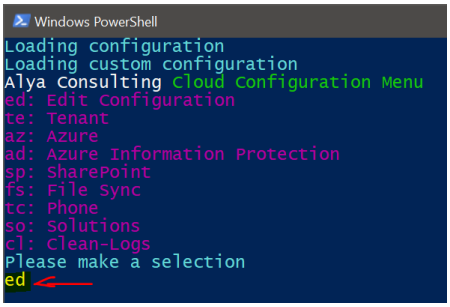
The very first step is to configure the constants for your environment.

Open a command prompt as administrator and launch 00_SetExecutionPolicy.cmd. This sets the PowerShell execution policy for the actual user to RemoteSigned which is required for the other PowerShell scripts.

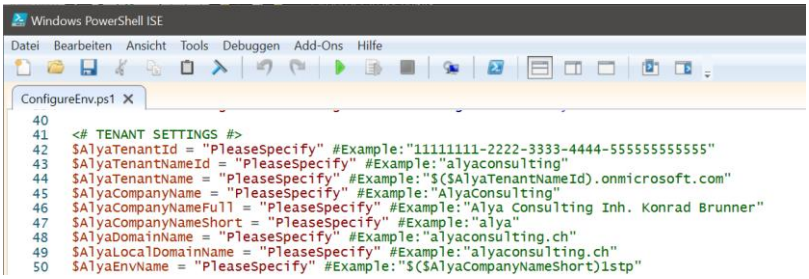
Right click the 03_StartMenu.ps1 script and select “Execute with PowerShell” or open a PowerShell window and type:

```
sl "X:\PathTo\XXXDO-ADM-BaseConfiguration "
.\03_StartMenu.ps1
```

Select the menu option “ed” and hit enter:



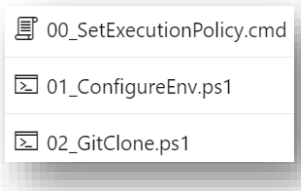
The PowerShell ISE opens, where you must specify at least your tenant settings:



Save the file, commit, and push your changes to git.

4. Checkout the repository to a new device

Please download first following 3 files from your repository to a newly created folder with the name XXXDO-ADM-BaseConfiguration where XXX is an abbreviation for your company:



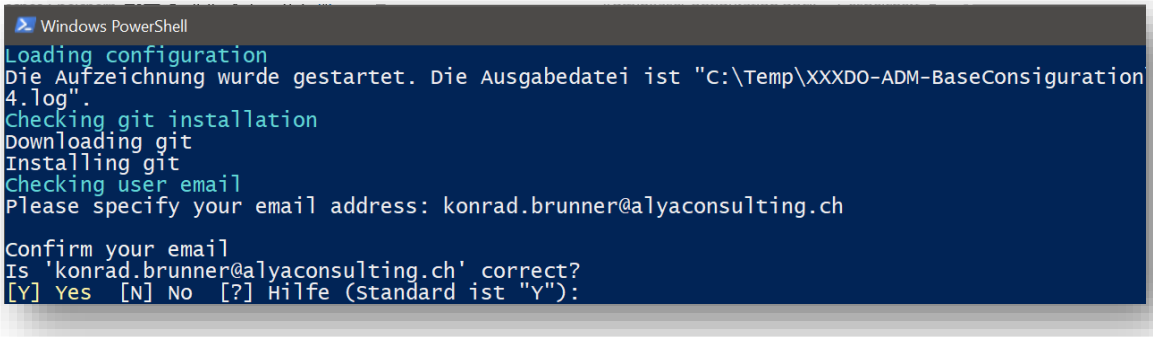
Right click all 3 files with right mouse button and select properties in the context menu. Tick the Unblock checkbox to allow access to the 3 files.

Right click 00_SetExecutionPolicy.cmd and select “Run as administrator” or launch it in an administrator command prompt. This sets the PowerShell execution policy for the actual user to RemoteSigned which is required for the other PowerShell scripts.

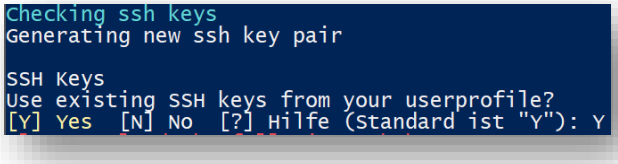
Now start 02_GitClone.ps1 by right click “Run with PowerShell” or open a PowerShell 64bit window and type:

```
sl "X:\PathTo\XXXDO-ADM-BaseConfiguration"
.\02_GitClone.ps1
```

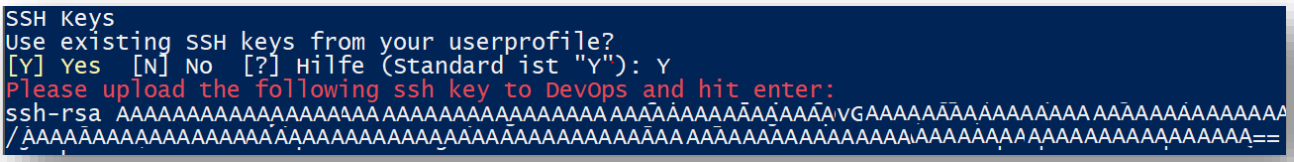
The script starts downloading and unpacking the git command line tool. When done it asks you about your email address:



Next the script checks the ssh key pair. If you do not have an existing key pair, it will generate one for you. If you have already an existing key pair in your user profile, it will ask to use that one:

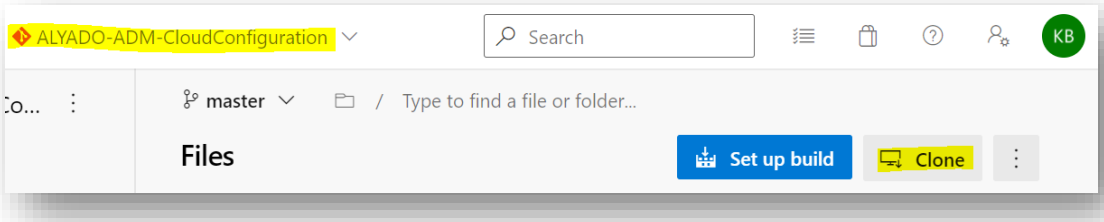


Now the script shows you your public rsa key:

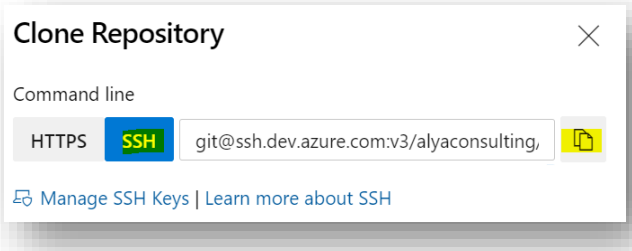


Copy this key from ssh-rsa up to the two equal signs at the end.

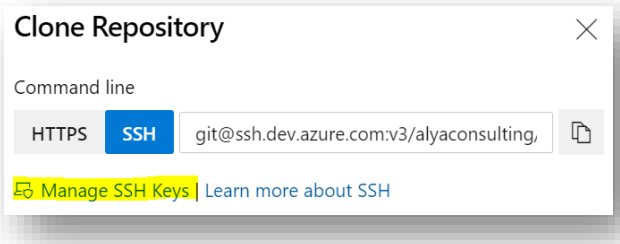
Open your own repository in a browser and select “Clone”:



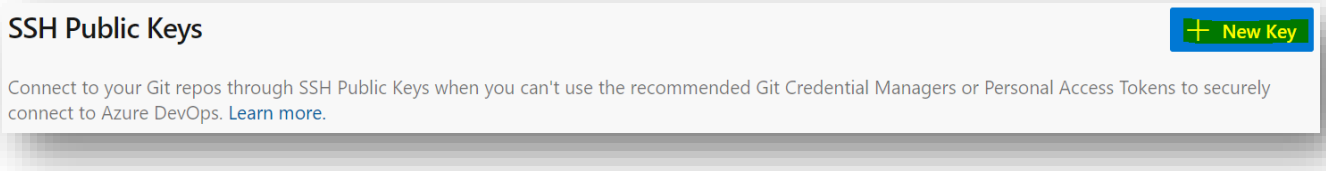
Select “SSH”:



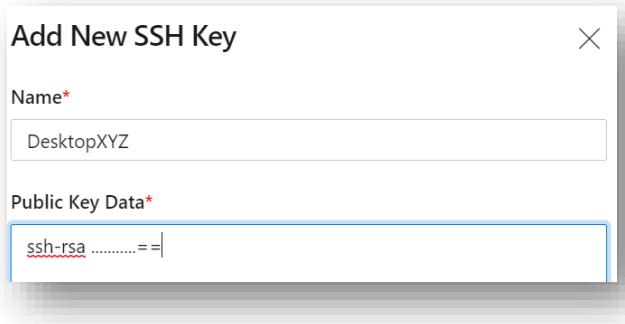
Select “Manage SSH Keys”:



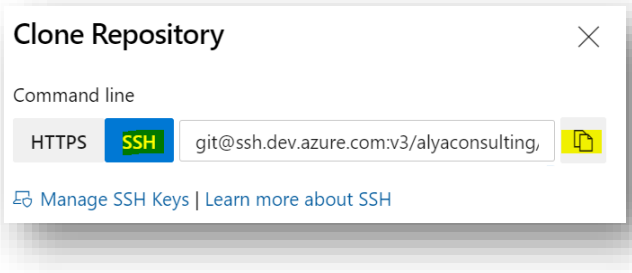
Select “+ New Key”:



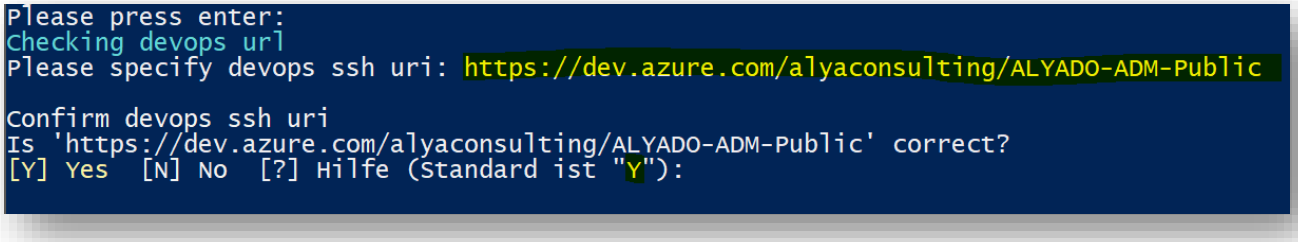
Give a name and add the copied key to “Public Key Data”:



Copy your “SSH” url:



Press in the PowerShell window enter, paste and confirm the SSH url to your own repository:



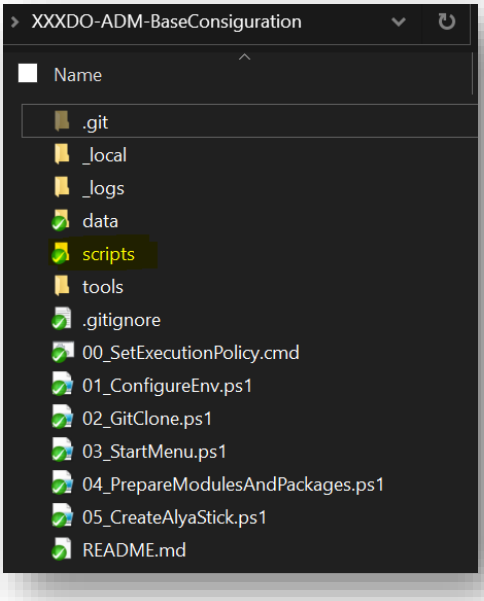
The script now connects to your repository and checks out the latest version:

```
Checking .ssh content
Checking connection
True
Fetching repository
connecting actual directory to git repository
Repository is not yet connected. Connecting...
Initialized empty Git repository in C:/Temp/XXXDO-ADM-BaseConsiguration/.git/
To: git@ssh.dev.azure.com:v3/alyaconsulting/ALYADO-ADM-Public/ALYADO-ADM-Public
Fetching changes
Checking out
Branch 'master' set up to track remote branch 'master' from 'origin'.
On branch master
Your branch is up to date with 'origin/master'.

Untracked files:
  (use "git add <file>..." to include in what will be committed)
    data/GlobalConfig.json

nothing added to commit but untracked files present (use "git add" to track)
Repository now connected!
```

Check if you have the scripts directory checked out. If it exists, all was working fine:



5. Build your own starter package

If you like to use the Base Configuration scripts on different servers or clients, you can create a minimum starter package with only a few files. This starter package you can then copy to a new device, to download the actual version from your Alya Base Configuration to work with the scripts and check in back actual changes.

To create your own starter package, put following files into a zip file you can take with you to new clients:

- 00_SetExecutionPolicy.cmd
- 01_ConfigureEnv.ps1
- 02_GitClone.ps1
- data\GlobalConfig.json

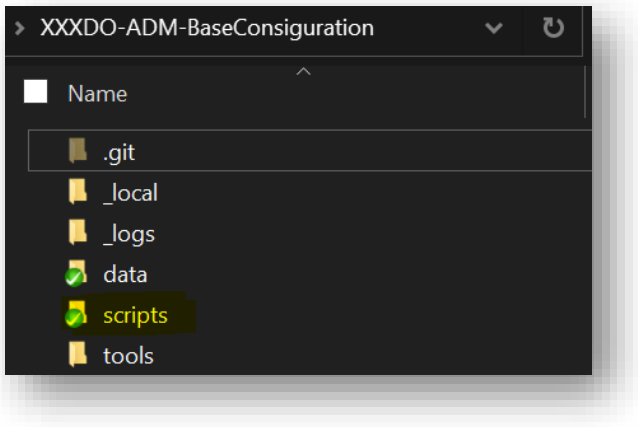
More is in general not needed. If you don't like to create on each new device a new SSH key pair to be configured in your repository, you can include your key pair in the zip file. May you have then to take care about the zip because each person having access to the zip will have access your repository with the provided keys.

If you like to add your key pair to the zip, add following files:

- _local\LocalConfig.json
- _local\ssh\id_rsa
- _local\ssh\id_rsa.pub

You have now to copy your zip to a new client, extract it and run first 00_SetExecutionPolicy.cmd and then 02_GitClone.ps1.

6. The different directories



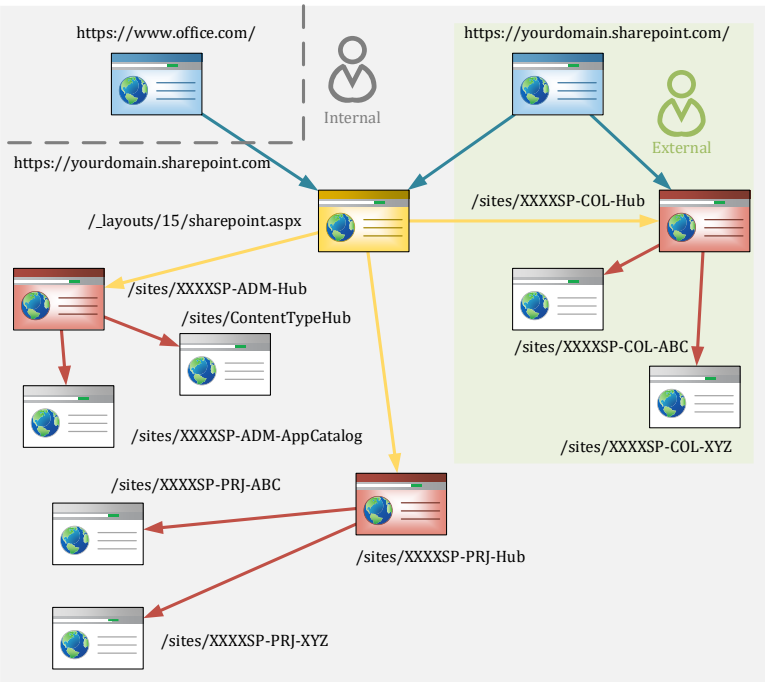
- .git
  - This directory contains your actual git index. I general you have nothing to do with this directory. Only change something on that directory, if you know what you do 😉.
- _local
  - In this directory we store things which should kept in general local. Things we never check in into git.
- _logs
  - Each run of any script generates a transcript in this directory.
- data
  - This directory holds you own data. Exports, own scripts, copies of templates and so on.
- scripts
  - This directory holds all the provided scripts. These scripts should be held general. Please do not add in this directory any script specific to your environment. The idea is, that this

directory can be copied from environment to environment without doing any change in it. So put your environment specific stuff in data or an own directory of your choice.

- tools
  - This directory holds tools the scripts download from the internet.

7. Information structure

The information structure built by the base configuration looks as follows:



7.1. The hub site configuration script

The hub sites at TraiLa are generated with a script. Please use this script, to create a new hub:

scripts\sharepoint\Configure-HubSites.ps1

This script uses a hub site configuration located here:

data\sharepoint\HubSitesConfiguration-en-US.ps1

To create a new hub, define it first in the script HubSitesConfiguration-en-US.ps1 by adding a new hub site configuration to the $hubSites variable:

```
$hubSites = @(
    @{
        short = "ADM"
        title = "$prefix-ADM-Hub"
        url = "$prefix-ADM-Hub"
        template = "TeamSite"
        locale4Creation = 1033 #TODO from config
        description = "Hub site for the administrative sites"
        siteScriptDescription = "Assigns the ADM site design"
        siteScript = $defaultSiteScript
        subSiteScript = $defaultSiteScript
        headerLayout = "Compact"   # Standard, Compact
        headerEmphasis = "None"  # None, Neutral, Soft, Strong
        siteLogoUrl = $AlyaLogoUrlQuad
        homePageTemplate = $homePageTemplateGenHub
    },
    @{
        short = "COL"
```

Run then the Configure-HubSites.ps1 script to create the newly defined hub site.

8. Any further questions

In case of any question about the Alya Base Configuration, please call our support:

Mo–Fr 8.00 to 12.00 and 13.00 to 17.00 under

support@alyaconsulting.ch

+41 62 562 04 63